Statements: Development.AbelianVarietyTheory (arithmetic geometry and Galois side)

59,300 statements from 7,401 files in part 1, proofs omitted, grouped by module.

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.CubicalPullbackLaw

theorem **CubicalPullLaw.ofHom** (h : CubicalPullLaw) : CubicalPullOfHomSt.{0}

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.GroupHomPullback

lemma **res_pullU** (f : Y ⟶ X) {U : X.Opens} {V : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (a : Γ(X, U)ˣ) : res h (pullU f U a) = pr f h a

lemma **res_res** {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (a : Γ(X, U)ˣ) : res h' (res h a) = res (h'.trans h) a

lemma **res_refl** {U : X.Opens} (a : Γ(X, U)ˣ) : res le_rfl a = a

lemma **res_pr** (f : Y ⟶ X) {U : X.Opens} {V W : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ V) (a : Γ(X, U)ˣ) : res h' (pr f h a) = pr f (h'.trans h) a

lemma **pr_res** (f : Y ⟶ X) {U U' : X.Opens} {V : Y.Opens} (h' : U ≤ U') (h : V ≤ f ⁻¹ᵁ U) (a : Γ(X, U')ˣ) : pr f h (res h' a) = pr f (h.trans fun _ hy => h' hy) a

lemma **pr_pr** (g : Y' ⟶ Y) (f : Y ⟶ X) {U : X.Opens} {V : Y.Opens} {W : Y'.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ g ⁻¹ᵁ V) (a : Γ(X, U)ˣ) : pr g h' (pr f h a) = pr (g ≫ f) (fun _ hy => h (h' hy)) a

lemma **pr_congr** {f g : Y ⟶ X} (e : f = g) {U : X.Opens} {V : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : V ≤ g ⁻¹ᵁ U) (a : Γ(X, U)ˣ) : pr f h a = pr g h' a

lemma **le_pre_inf** (M L : LB X) (φ : K → (Y ⟶ X)) (j : K → M.ι) (i : K → L.ι) (k : K) : LB.wchart M φ j ⊓ LB.wchart L φ i ≤ φ k ⁻¹ᵁ (M.U (j k) ⊓ L.U (i k))

lemma **res_linComb_g** (L : LB X) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (a b : K → L.ι) {W : Y.Opens} (h : W ≤ LB.wchart L φ a ⊓ LB.wchart L φ b) : res (U

lemma **res_prod_pr** (φ : K → (Y ⟶ X)) (ε : K → ℤ) {U : K → X.Opens} {V W : Y.Opens} (h : ∀ k, V ≤ φ k ⁻¹ᵁ U k) (h' : W ≤ V) (a : ∀ k, Γ(X, U k)ˣ) : res h' (∏ k, pr (φ k) (h k) (a k) ^ ε k) = ∏ k, pr (φ k) (h'.trans (h k)) (a k) ^ ε k

lemma **isoUnit_compat** {L M : LB X} {u : LB.Iso L M} (hu : u.IsGen) (j j' : M.ι) (i i' : L.ι) (f : Y ⟶ X) {W : Y.Opens} (h : W ≤ f ⁻¹ᵁ (M.U j ⊓ L.U i)) (h' : W ≤ f ⁻¹ᵁ (M.U j' ⊓ L.U i')) (hM : W ≤ f ⁻¹ᵁ (M.U j ⊓ M.U j')) (hL : W ≤ f ⁻¹ᵁ (L.U i ⊓ L.U i')) : pr f h (isoUnit u j i) = pr f hM (M.g j j') * (pr f hL (L.g i i'))⁻¹ * pr f h' (isoUnit u j' i')

theorem **isoLinComb_isGen** {L M : LB X} {u : LB.Iso L M} (hu : u.IsGen) (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (isoLinComb u φ ε).IsGen

theorem **isLB_linComb** {L : LB X} (hL : L.IsLB) (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (LB.LinComb L φ ε).IsLB where

lemma **wchart_square** (L : LB X) {f : X' ⟶ X} {φ : K → (Y ⟶ X)} {φ' : K → (Y' ⟶ X')} {h : Y' ⟶ Y} (sq : ∀ k, φ' k ≫ f = h ≫ φ k) (j : K → (L.pull f).ι) : LB.wchart (L.pull f) φ' j ≤ h ⁻¹ᵁ LB.wchart L φ (fun k => j k PUnit.unit)

theorem **genSquare_isGen** (L : LB X) {f : X' ⟶ X} {φ : K → (Y ⟶ X)} {φ' : K → (Y' ⟶ X')} {h : Y' ⟶ Y} (sq : ∀ k, φ' k ≫ f = h ≫ φ k) (ε : K → ℤ) {t : LB.Gen (LB.LinComb L φ ε)} (ht : t.IsGen) : (genSquare L sq ε t).IsGen

lemma **pt_psum** {T : Over S'} {n : ℕ} (x : Fin n → (T ⟶ G')) (I : Finset (Fin n)) : f.pt (psum G' x I) = psum G (fun i => f.pt (x i)) I

lemma **comp_psum** {T T' : Over S} (g : T' ⟶ T) {n : ℕ} (x : Fin n → (T ⟶ G)) (I : Finset (Fin n)) : g ≫ psum G x I = psum G (fun i => g ≫ x i) I

lemma **cube_sq** (I : thK.{u}) : thφ G' (x3 G') I ≫ f.tot = f.cube ≫ thφ G (x3 G) I

lemma **eSec_sq** : eSec G' ≫ f.tot = f.unitMap ≫ eSec G

theorem **pullCube_isGen** (L : LB G.left) {t : LB.Gen (Theta G L)} (ht : t.IsGen) : (f.pullCube L t).IsGen

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.MultiplicationEtaleTorsion

theorem **comp_mulN** (A : Over S) [GrpObj A] {T : Over S} (x : T ⟶ A) (n : ℕ) : x ≫ mulN A n = x ^ n

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.PolarizedAbelianVarieties

lemma **trivialLB_isLB** (X : Scheme.{0}) : (trivialLB X).IsLB where

lemma **isTrivialLB_of_g** {X : Scheme.{0}} (L : LB X) (h : ∀ i j, L.g i j = 1) : IsTrivialLB L

lemma **linComb_g** {X Y : Scheme.{0}} (L : LB X) (h : ∀ i j, L.g i j = 1) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (i j) : (L.LinComb φ ε).g i j = 1

lemma **isAmpleLB_of_isAffine** {X : Scheme.{0}} [IsAffine X] (L : LB X) (h : ∀ i j, L.g i j = 1) (hc : ⊔ i, L.U i = ⊤) : IsAmpleLB L

lemma **geometricallyConnected_id** (S : Scheme.{0}) : GeometricallyConnected (𝟙 S)

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.SectionsGroupRank

theorem **map_ev** (f : Y ⟶ X) {U : X.Opens} {V V' : Y.Opens} (e : V ≤ f ⁻¹ᵁ U) (i : V' ≤ V) (x : Γ(X, U)) : (Y.presheaf.map (homOfLE i).op).hom ((f.appLE U V e).hom x) = (f.appLE U V' (i.trans e)).hom x

theorem **ev_map** (f : Y ⟶ X) {U U' : X.Opens} {V : Y.Opens} (e : V ≤ f ⁻¹ᵁ U) (i : U ≤ U') (e' : V ≤ f ⁻¹ᵁ U') (x : Γ(X, U')) : (f.appLE U V e).hom ((X.presheaf.map (homOfLE i).op).hom x) = (f.appLE U' V e').hom x

theorem **ev_ev** (f : Z ⟶ Y) (g : Y ⟶ X) (k : Z ⟶ X) (hk : f ≫ g = k) {U : X.Opens} {V : Y.Opens} {W : Z.Opens} (e₁ : V ≤ g ⁻¹ᵁ U) (e₂ : W ≤ f ⁻¹ᵁ V) (e₃ : W ≤ k ⁻¹ᵁ U) (x : Γ(X, U)) : (f.appLE V W e₂).hom ((g.appLE U V e₁).hom x) = (k.appLE U W e₃).hom x

theorem **transition_pull** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (h : Z ⟶ Y) (φ' : K → (Z ⟶ X)) (hφ : ∀ k, h ≫ φ k = φ' k) (j j' : K → L.ι) (e : LB.wchart L φ' j ⊓ LB.wchart L φ' j' ≤ h ⁻¹ᵁ (LB.wchart L φ j ⊓ LB.wchart L φ j')) : (h.appLE _ _ e).hom (show Γ(Y, LB.wchart L φ j ⊓ LB.wchart L φ j') from ((LB.LinComb L φ ε).g j j').val) = (show Γ(Z, LB.wchart L φ' j ⊓ LB.wchart L φ' j') from ((LB.LinComb L φ' ε).g j j').val)

theorem **Secs.ext** {s t : Secs A L a T} (h : ∀ j, LB.Sec.f s j = LB.Sec.f t j) : s = t

theorem **liftFun_eq** {T : Over S} (W : (T ⊗ A).left.Opens) (r : Γ(T.left, ⊤)) : liftFun A W r = ((fst T A).left.appLE ⊤ W le_top).hom r

theorem **map_liftFun** {W W' : (T ⊗ A).left.Opens} (e : W' ≤ W) (r : Γ(T.left, ⊤)) : ((T ⊗ A).left.presheaf.map (homOfLE e).op).hom (liftFun A W r) = liftFun A W' r

theorem **zero_f** (j) : LB.Sec.f (0 : Secs A L a T) j = 0

theorem **add_f** (s t : Secs A L a T) (j) : LB.Sec.f (s + t) j = LB.Sec.f s j + LB.Sec.f t j

theorem **neg_f** (s : Secs A L a T) (j) : LB.Sec.f (-s) j = - LB.Sec.f s j

theorem **smul_f** (r : Γ(T.left, ⊤)) (s : Secs A L a T) (j) : LB.Sec.f (r • s) j = liftFun A _ r ∗ LB.Sec.f s j

theorem **appLE_liftFun** {T' : Over S} (H : T' ⊗ A ⟶ T ⊗ A) (u : T' ⟶ T) (hH : H ≫ fst T A = fst T' A ≫ u) (W : (T ⊗ A).left.Opens) (W' : (T' ⊗ A).left.Opens) (e : W' ≤ H.left ⁻¹ᵘ W) (r : Γ(T.left, ⊤)) : (H.left.appLE W W' e).hom (liftFun A W r) = liftFun A W' (u.left.appTop.hom r)

theorem **isPullSec_pullSec** {T' : Over S} (u : T' ⟶ T) (s : Secs A L a T) : IsPullSec A L a (u ▷ A) (whiskerRight_snd _ _) s (pullSec u s)

theorem **pullSec_zero** {T' : Over S} (u : T' ⟶ T) : pullSec u (0 : Secs A L a T) = 0

theorem **pullSec_add** {T' : Over S} (u : T' ⟶ T) (s t : Secs A L a T) : pullSec u (s + t) = pullSec u s + pullSec u t

theorem **pullSec_smul** {T' : Over S} (u : T' ⟶ T) (r : Γ(T.left, ⊤)) (s : Secs A L a T) : pullSec u (r • s) = u.left.appTop.hom r • pullSec u s

theorem **twist_comb** {R R' C : Type} [CommRing R] [CommRing R'] [CommRing C] (F : R →+* C) (G : R' →+* C) (c : C) {x' y' r' : R} {x y r : R'} (hx : F x' = c * G x) (hy : F y' = c * G y) (hr : F r' = G r) : F 0 = c * G 0 ∧ F (x' + y') = c * G (x + y) ∧ F (r' * x') = c * G (r * x)

theorem **isActOf_comb** (θ : ThetaGrp A L a T) {s t s' t' : Secs A L a T} (hs : IsActOf A L a θ s s') (ht : IsActOf A L a θ t t') (r : Γ(T.left, ⊤)) : IsActOf A L a θ (s + t) (s' + t') ∧ IsActOf A L a θ (r • s) (r • s')

theorem **isActOf_zero** (θ : ThetaGrp A L a T) : IsActOf A L a θ (0 : Secs A L a T) (0 : Secs A L a T)

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.SemiAbelianSchemes

theorem **isAbelianScheme_unit** (S : Scheme.{u}) : IsAbelianScheme (𝟙_ (Over S)) where

theorem **flat_splitTorusAff** : Flat (splitTorusAff ZU X).hom

theorem **finitePresentation_splitTorusAff** [Module.Finite ℤ X] : LocallyOfFinitePresentation (splitTorusAff ZU X).hom

theorem **isTorusExtension_splitTorus** (k : Type u) [Field k] [Module.Free ℤ X] [Module.Finite ℤ X] : IsTorusExtension (splitTorus (Spec (CommRingCat.of k)) X)

Development.AbelianVarietyTheory.BlockTheory.AbelianSchemeLaw.TrivialBiextension

theorem **LamBiext.nonempty** (h : LamBiext.{u}) : LamBiextNonempty.{u}

lemma **pt₃_pt** {T : Over S} (a b c : T ⟶ G) (p q : G3 G ⟶ G) : pt₃ G a b c ≫ pt G p q = pt G (lift a (lift b c) ≫ p) (lift a (lift b c) ≫ q)

lemma **pt₃_law₁φ** {T : Over S} (a a' b : T ⟶ G) : (fun k => pt₃ G a a' b ≫ law₁φ G (x3 G 0) (x3 G 1) (x3 G 2) k) = law₁φ G a a' b

lemma **pt₃_law₂φ** {T : Over S} (a b b' : T ⟶ G) : (fun k => pt₃ G a b b' ≫ law₂φ G (x3 G 0) (x3 G 1) (x3 G 2) k) = law₂φ G a b b'

lemma **sum_ulift_three** (f : ULift.{u} (Fin 3) → ℤ) : ∑ k, f k = f ⟨0⟩ + f ⟨1⟩ + f ⟨2⟩

lemma **compat_sameSum** {T : Over S} (a a' b b' : T ⟶ G) : LB.SameSum (Sum.elim (law₂φ G (padd G a a') b b') (Sum.elim (law₁φ G a a' b) (law₁φ G a a' b'))) (Sum.elim lawε (Sum.elim lawε lawε)) (Sum.elim (law₁φ G a a' (padd G b b')) (Sum.elim (law₂φ G a b b') (law₂φ G a' b b'))) (Sum.elim lawε (Sum.elim lawε lawε))

lemma **pull_one** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (h : Z ⟶ Y) (ht : ∀ j, t.s j = 1) (j) : (LB.pullGen L φ ε t h).s j = 1

lemma **mul_one'** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (ht : ∀ j, t.s j = 1) (ht' : ∀ j, t'.s j = 1) (j) : (LB.mulGen L φ ε φ' ε' t t').s j = 1

lemma **corr_one** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (ht : ∀ j, t.s j = 1) (ht' : ∀ j, t'.s j = 1) : LB.Gen.Corr L φ ε φ' ε' t t'

lemma **isGen_one** (φ : K → (Y ⟶ X)) (ε : K → ℤ) : LB.Gen.IsGen (L

Development.AbelianVarietyTheory.BlockTheory.AdelicCMFields.SupportedAtSplit

theorem **SplitRepresentatives.cm** (H : SplitRepresentatives) : SplitRepresentativesCM

Development.AbelianVarietyTheory.BlockTheory.AdelicCMFields.UnitaryHeckeCharacterIdele

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsCMExtension.isCMField** [h : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsCMExtension F E] : IsCMField E

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.embedding_conj** [_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsCMExtension F E] (φ : E →+* ℂ) (x : E) : φ (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.conj F E x) = starRingEnd ℂ (φ (x))

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.conj_conj** [_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsCMExtension F E] (x : E) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.conj F E (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.conj F E x) = x

theorem **conj_ne_one** [IsCMExtension F E] : conj F E ≠ 1

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.continuous_finiteCompletionMap** (w : HeightOneSpectrum (𝒪 L)) : Continuous (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.finiteCompletionMap K L w)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.finiteCompletionMap_coe** (w : HeightOneSpectrum (𝒪 L)) (x : K) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.finiteCompletionMap K L w (algebraMap K _ x) = algebraMap L (w.adicCompletion L) (algebraMap K L x)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.finiteCompletionMap_mapsTo** (w : HeightOneSpectrum (𝒪 L)) : Set.MapsTo (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.finiteCompletionMap K L w) ((w.under (𝒪 K)).adicCompletionIntegers K : Set _) (w.adicCompletionIntegers L : Set _)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.tendsto_under** : Tendsto (fun w : HeightOneSpectrum (𝒪 L) => w.under (𝒪 K)) cofinite cofinite

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.adeleMap_algebraMap** (x : K) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.adeleMap K L (algebraMap K _ x) = algebraMap L (AdeleRing (𝒪 L) L) (algebraMap K L x)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.continuous_adeleMap** : Continuous (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.adeleMap K L)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.continuous_ideleMap** : Continuous (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleMap K L)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleMap_principalIdele** (x : Kˣ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleMap K L (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.principalIdele K x) = _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.principalIdele L (Units.map (algebraMap K L).toMonoidHom x)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleConj_principalIdele** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsCMExtension F E] (x : Eˣ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleConj F E (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.principalIdele E x) = _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.principalIdele E (Units.map (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.conj F E).toRingEquiv.toRingHom.toMonoidHom x)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.exists_isUniformiser** (P : HeightOneSpectrum (𝒪 E)) : ∃ ϖ, _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsUniformiser P ϖ

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.valueAt_eq** (χ : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar E) {P : HeightOneSpectrum (𝒪 E)} (h : χ.IsUnramifiedAt P) {ϖ : (P.adicCompletion E)ˣ} (hϖ : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsUniformiser P ϖ) : χ.valueAt P = (χ.localComponent P ϖ : ℂ)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.valueAt_of_ramified** (χ : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar E) {P : HeightOneSpectrum (𝒪 E)} (h : ¬ χ.IsUnramifiedAt P) : χ.valueAt P = 0

theorem **norm_valueAt_le** (χ : UnitaryHeckeChar E) (P : HeightOneSpectrum (𝒪 E)) : ‖χ.valueAt P‖ ≤ 1

theorem **norm_term_le** (χ : UnitaryHeckeChar E) (P : HeightOneSpectrum (𝒪 E)) (s : ℂ) : ‖χ.valueAt P * (Ideal.absNorm P.asIdeal : ℂ) ^ (-s)‖ ≤ (Ideal.absNorm P.asIdeal : ℝ) ^ (-s.re)

theorem **one_sub_ne_zero** (χ : UnitaryHeckeChar E) (P : HeightOneSpectrum (𝒪 E)) {s : ℂ} (hs : 0 < s.re) : 1 - χ.valueAt P * (Ideal.absNorm P.asIdeal : ℂ) ^ (-s) ≠ 0

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.summable_absNorm_rpow** {σ : ℝ} (hσ : 1 < σ) : Summable fun P : HeightOneSpectrum (𝒪 E) => (Ideal.absNorm P.asIdeal : ℝ) ^ (-σ)

theorem **multipliable_eulerFactor** (χ : UnitaryHeckeChar E) {s : ℂ} (hs : 1 < s.re) : Multipliable fun P => χ.eulerFactor P s

theorem **hasProd_eulerFactor** (χ : UnitaryHeckeChar E) (s : {s : ℂ // 1 < s.re}) : HasProd (fun P => χ.eulerFactor P s) (χ.finiteL s)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.norm_eq_one_of_valued** {P : HeightOneSpectrum (𝒪 E)} {x : P.adicCompletion E} (h : Valued.v x = 1) : ‖x‖ = 1

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.finite_mulSupport** (x : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.Idele E) : (Function.mulSupport fun v : HeightOneSpectrum (𝒪 E) => ‖x.val.2 v‖).Finite

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleNorm_ne_zero** (x : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.Idele E) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleNorm x ≠ 0

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleNorm_principalIdele** (x : Eˣ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleNorm (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.principalIdele E x) = 1

theorem **continuous_finiteIdeleNorm** : Continuous (finiteIdeleNorm (E

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.continuous_ideleNorm** : Continuous (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleNorm (E

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.ideleNorm_localIdele** (P : HeightOneSpectrum (𝒪 E)) (u : (P.adicCompletion E)ˣ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.ideleNorm (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.localIdele P u) = ‖(u : P.adicCompletion E)‖

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.normPow_isUnramifiedAt** (t : ℝ) (P : HeightOneSpectrum (𝒪 E)) : (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.normPow E t).IsUnramifiedAt P

private theorem **norm_uniformiser_eq_absNorm_inv** {P : HeightOneSpectrum (𝒪 E)} {ϖ : (P.adicCompletion E)ˣ} (hϖ : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.IsUniformiser P ϖ) : ‖(ϖ : P.adicCompletion E)‖ = (Ideal.absNorm P.asIdeal : ℝ)⁻¹

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.normPow_valueAt** (t : ℝ) (P : HeightOneSpectrum (𝒪 E)) : (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHeckeCharacterIdele.UnitaryHeckeChar.normPow E t).valueAt P = Complex.exp ((-(t * Real.log (Ideal.absNorm P.asIdeal)) : ℝ) * Complex.I)

theorem **exists_normPow_valueAt_eq_neg_one** (P : HeightOneSpectrum (𝒪 E)) : ∃ t : ℝ, (normPow E t).valueAt P = -1

Development.AbelianVarietyTheory.BlockTheory.CMModuli.AlgebraLevelSetting

theorem **torus_w** {e : ℕ} (i : splitTorus (Lv I e) X ⟶ thickObj R I e G) : (i.left ≫ pr I e G) ≫ G.hom = (splitTorus (Lv I e) X).hom ≫ thick R I e

theorem **exists_three_or_seven** (V : Type) [CommRing V] [IsLocalRing V] : ∃ n : ℕ, (n = 3 ∨ n = 7) ∧ IsUnit (n : V)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.CoefficientFieldBaseChange

theorem **ze_prim** {n : ℕ} (hn : 0 < n) : IsPrimitiveRoot (ze n) n

theorem **ze_pow** {n m : ℕ} (hn : 0 < n) (hm : 0 < m) : ze (n * m) ^ m = ze n

theorem **rootZero_mem_adjoin** (l : Level H) : H.rt0 ∈ KA l

theorem **ze_mem_KA** (l : Level H) : ze l.A ∈ KA l

theorem **IsCoeffField.coeff_field_contains_root** {l : Level H} {F₀ : IntermediateField ℚ ℂ} (h : IsCoeffField l F₀) : H.rt0 ∈ F₀

theorem **IsCoeffField.ze_mem** {l : Level H} {F₀ : IntermediateField ℚ ℂ} (h : IsCoeffField l F₀) : ze l.A ∈ F₀

theorem **IsCoeffField.KA_le** {l : Level H} {F₀ : IntermediateField ℚ ℂ} (h : IsCoeffField l F₀) : KA l ≤ F₀

theorem **isPullback_baseChange** {l : Level H} {X : Scheme.{0}} (str : X ⟶ Sp (KA l)) {F₀ : IntermediateField ℚ ℂ} (h : IsCoeffField l F₀) : IsPullback (baseChangeFst str h) (baseChangeStr str h) str (spMap h.incl)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.CoveringAbelianLocusSmooth

theorem **IsChartAt.unif** {Y : Model Xo c β} (hY : Y.N3) {x : Y.X} {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] {ξ : Sp R ⟶ Y.X} (h : IsChartAt Y x R ξ) : ∃ (t : SAB H (Sp R)) (_ : SABMap t Y.t ξ), UniformisationConclusion t (ξ ⁻¹' abLocus Y.t.G)

theorem **goodPrime_iff** (l : Level H) {p : ℕ} (hp : p.Prime) : GoodPrime l p ↔ ¬ (p : ℤ) ∣ discr E ∧ ¬ p ∣ l.M'

Development.AbelianVarietyTheory.BlockTheory.CMModuli.CoveringAndSmoothness

theorem **IsUnifBase.out** {Y : Model Xo c β} (h3 : Y.N3) {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] {ξ : Sp R ⟶ Y.X} (h : IsUnifBase Y R ξ) : ∃ (t : SAB H (Sp R)) (_ : SABMap t Y.t ξ), UniformisationConclusion t (ξ ⁻¹' abLocus Y.t.G)

theorem **AbLocusSmooth.smooth** {Y : Model Xo c β} (h : AbLocusSmooth Y) (U : Y.X.Opens) (hU : (U : Set Y.X) = abLocus Y.t.G) : SmoothOfRelativeDimension D.dim (U.ι ≫ Y.s)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.CubicalSymmetricTriv

theorem **IsCubSymIso.isPullCubical** {m : CubeMap G' G} {L' : CubSym G'} {L : CubSym G} {w : LB.Iso L'.L.L (L.L.L.pull m.F)} (h : IsCubSymIso m L' L w) : m.IsPullCubical L'.cub L.cub

theorem **IsBiextTriv.axes** {L : RigLB G} {c : CubicalStr G L} {i : H ⟶ G} [IsMonHom i] {s : LB.Gen (lamOn i L.L)} (h : IsBiextTriv c i s) {T : Over S} {z : T.left} {J : (T.left ⟶ G.left) → L.L.ι} (hJ : Good L.L z J) (t : T ⟶ H) (x : T ⟶ G) : lamV s z J 1 x = 1 ∧ lamV s z J t 1 = 1

theorem **existsUnique_isPairing** (N : ℕ) {T : Over S} (x x' : T ⟶ G) (hx : x ^ N = 1) (hx' : x' ^ N = 1) : ∃! e, IsPairing c N x x' e

theorem **isPairing_pairing** (N : ℕ) {T : Over S} (x x' : T ⟶ G) (hx : x ^ N = 1) (hx' : x' ^ N = 1) : IsPairing c N x x' (pairing c N x x' hx hx')

theorem **IsTorusExt.isStable** {i : splitTorus S X ⟶ G} {q : G ⟶ B} (h : IsTorusExt i q) (V : B.left.Opens) : IsStable i (q.left ⁻¹ᵁ V)

theorem **trivGen_isGen** {Y Z : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Z ⟶ Y)) (ε : K → ℤ) : (trivGen φ ε).IsGen

theorem **val_trivGen** {Y Z W : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Z ⟶ Y)) (ε : K → ℤ) (r : W ⟶ Z) (z : W) (J) : val (trivGen φ ε) r z J = 1

Development.AbelianVarietyTheory.BlockTheory.CMModuli.CyclotomicRootsOfUnity

theorem **inv_natCast_mem** {W : ValuationSubring ℂ} {p : ℕ} (hp : p.Prime) (hW : LiesOver W p) {d : ℕ} (hd : ¬ p ∣ d) : (d : ℂ)⁻¹ ∈ W

theorem **zeta_mem_valuationSubring** (W : ValuationSubring ℂ) {M : ℕ} (hM : M ≠ 0) : zeta M ∈ W

theorem **cycloLocal_le** {W : ValuationSubring ℂ} {p : ℕ} (hp : p.Prime) (hW : LiesOver W p) {M : ℕ} (hM : M ≠ 0) {F : IntermediateField ℚ ℂ} (hF : zeta M ∈ F) : cycloLocal p M ≤ W.toSubring ⊓ F.toSubfield.toSubring

theorem **zeta_mem_of_dvd** {F : IntermediateField ℚ ℂ} {a N : ℕ} (hN : N ≠ 0) (h : a ∣ N) (hF : zeta N ∈ F) : zeta a ∈ F

theorem **zeta_lcm_mem** {F : IntermediateField ℚ ℂ} {a b : ℕ} (ha : a ≠ 0) (hb : b ≠ 0) (h₁ : zeta a ∈ F) (h₂ : zeta b ∈ F) : zeta (Nat.lcm a b) ∈ F

theorem **Idx.Le.refl** {p : ℕ} (i : Idx H p) : i.Le i

theorem **Idx.Le.trans** {p : ℕ} {i j k : Idx H p} (h : i.Le j) (h' : j.Le k) : i.Le k

theorem **zeta_mem_sup** (F : IntermediateField ℚ ℂ) (N : ℕ) : zeta N ∈ F ⊔ IntermediateField.adjoin ℚ {zeta N}

theorem **Idx.le_add** {p : ℕ} (i : Idx H p) {N : ℕ} (hN : N ≠ 0) : i.Le (i.add hN)

theorem **Idx.zeta_mem_add** {p : ℕ} (i : Idx H p) {N : ℕ} (hN : N ≠ 0) : zeta N ∈ (i.add hN).F

Development.AbelianVarietyTheory.BlockTheory.CMModuli.DVRSplitting

theorem **IsFinUnram.map_maximalIdeal** {V W : Type} [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] [CommRing W] [IsDomain W] [IsDiscreteValuationRing W] [Algebra V W] (h : IsFinUnram V W) : IsLocalHom (algebraMap V W) ∧ (maximalIdeal V).map (algebraMap V W) = maximalIdeal W

Development.AbelianVarietyTheory.BlockTheory.CMModuli.EllipticFibrePrincipalData

theorem **isKForm_kForm** (D : Datum H) (ε : ℤˣ) : IsKForm D ε (kForm D ε true)

theorem **isHomFam_tmul** {R : Type} [CommRing R] {ω : R} {C : Ell H R ω} {D : Datum H} {ε : ℤˣ} {T : PAT H (Sp R)} (h : IsTen C D (kForm D ε true) T) : IsHomFam C T.t (⊤ : Submodule (𝒪 E) D.La) ε true (fun a => h.ten.tmul a.1) where

theorem **isHolTo_slice** {g : ℕ} {X : Scheme.{0}} {e : (Fin g → ℂ) → (Sp ℂ ⟶ X)} (h : IsHolTo ((Set.univ : Set (Fin 0 → ℂ)) ×ˢ (Set.univ : Set (Fin g → ℂ))) fun bz => e bz.2) : IsHolTo Set.univ e

theorem **isExp_family** {D : Datum H} {l : Level H} {Xo : Moduli D l} {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (FU : FamilyUnif Xo σ ty) (i : Fin FU.n) {Pe : (FU.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (FU.Di i) ty Pe) : IsExp Xo.univ.t (FU.u i Pe) (FU.Di i) ty

Pe (FU.ex i Pe) where

theorem **im_mul_form_pos** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) {ρ : ℂ} (hρ : ρ ^ 2 = -(H.d : ℂ)) {a : D.La} (ha : a ∈ PureType.lat hP ρ) (h0 : a ≠ 0) : 0 < ρ.im * D.EF (H.w0 • a) a

theorem **isLatData_id** {D D₂ : Datum H} (h₁ : D.p = 1 ∧ D.q = 0) (h₂ : D₂.p = 0 ∧ D₂.q = 1) {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hty : IsTypeFn H 1 0 ty) (hP : IsPeriod D ty Pe) : IsLatData D D₂ D (p

Development.AbelianVarietyTheory.BlockTheory.CMModuli.ExponentialDifferential

theorem **exp_zero** : E.exp 0 = 1

theorem **mem_lat** (z : Fin g → ℂ) : z ∈ E.lat ↔ E.exp z = 1

theorem **exp_add_lat** (z : Fin g → ℂ) (la : E.lat) : E.exp (z + la) = E.exp z

theorem **exists_chart** (z : Fin g → ℂ) : ∃ i, z ∈ pre F.exp (L.U i)

theorem **IsSec.add_lat** {F : Unif A L g} {a : F.toExp.lat → (Fin g → ℂ) → ℂ} (ha : F.HasFactor a) {s : L.Sec} {θ : (Fin g → ℂ) → ℂ} (hs : F.IsSec s θ) (la : F.toExp.lat) (z : Fin g → ℂ) : θ (z + la) = a la z * θ z

Development.AbelianVarietyTheory.BlockTheory.CMModuli.FractionLevelCorrespondence

lemma **le_chart** {L' : LB X'} {L : LB X} {φ : X' ⟶ X} {a : K → (T ⟶ X')} {b : K → (T ⟶ X)} {ε : K → ℤ} (hb : ∀ k, b k = a k ≫ φ) {j' : K → L'.ι} {j : K → L.ι} {V : T.Opens} (h' : V ≤ (LB.LinComb L' a ε).U j') (h : V ≤ (LB.LinComb L b ε).U j) (k : K) : V ≤ a k ⁻¹ᵁ ((L.pull φ).U (fun _ => j k) ⊓ L'.U (j' k))

lemma **fracMap_algebraMap** (f : O' →+* R) (hf : Function.Injective f) (a : O') : fracMap f hf (algebraMap O' _ a) = algebraMap R _ (f a)

lemma **exists_isSubst** (J : Ideal O') [hJ : IsAdicComplete J O'] {π : O'} (hπ : π ∈ J) : ∃ ev, IsSubst π ev

lemma **IsSubst.unique** (J : Ideal O') [IsHausdorff J O'] {π : O'} (hπ : π ∈ J) {ev ev' : PowerSeries O' →+* O'} (h : IsSubst π ev) (h' : IsSubst π ev') : ev = ev'

lemma **two_mul_half** (x : ArcInput D c β O' b) (y : x.P.X) : 2 * (x.be y y / 2) = x.be y y

lemma **levelM_mul_div** (x : ArcInput D c β O' b) (y z : x.P.X) : (l.M : ℤ) * (x.be y z / (l.M : ℤ)) = x.be y z

lemma **levelA_mul_div** (x : ArcInput D c β O' b) (y z : x.P.X) : (l.A : ℤ) * (x.be y z / (l.A : ℤ)) = x.be y z

Development.AbelianVarietyTheory.BlockTheory.CMModuli.FullLevelModel

theorem **zeta_one** : zeta 1 = 1

theorem **oRing_le_KA** (l : Level H) {N : ℕ} (h : zeta N ∈ KA l) (c : ℕ) : oRing H N c ≤ (KA l).toSubring

theorem **zeta_mem_of_eq** (l : Level H) {N : ℕ} (h : l.A = N) : zeta N ∈ KA l

theorem **zeta_one_mem** (l : Level H) : zeta 1 ∈ KA l

theorem **oRing_le_mul** (N n e : ℕ) (he : 0 < e) : oRing H N n ≤ oRing H N (n * e)

theorem **oRing_one_le** (N c : ℕ) : oRing H 1 c ≤ oRing H N c

theorem **oRing_le_oLoc** (p N n : ℕ) (h : n.Coprime p) : oRing H N n ≤ oLoc H p N

theorem **ZP_le_oLoc** (p N : ℕ) : ZP p N ≤ oLoc H p N

Development.AbelianVarietyTheory.BlockTheory.CMModuli.HermitianTensorPowers

lemma **herm_add_left** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z z' w : Fin g → ℂ) : herm Hm (z + z') w = herm Hm z w + herm Hm z' w

lemma **herm_swap** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (h : Hm.IsHermitian) (v w : Fin g → ℂ) : herm Hm w v = starRingEnd ℂ (herm Hm v w)

lemma **shift_comm** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (h : Hm.IsHermitian) (k : ℕ) (v w : Fin g → ℂ) (θ : (Fin g → ℂ) → ℂ) (z : Fin g → ℂ) : shift Hm k v (shift Hm k w θ) z = commVal Hm k v w * shift Hm k w (shift Hm k v θ) z

lemma **shift_factor** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (h : Hm.IsHermitian) (k : ℕ) (c : ℂ) (v l : Fin g → ℂ) (a : ℤ) (ha : k * (herm Hm v l).im = a) (θ : (Fin g → ℂ) → ℂ) (hθ : ∀ z, θ (z + l) = factor Hm c l z ^ k * θ z) (z : Fin g → ℂ) : shift Hm k v θ (z + l) = factor Hm c l z ^ k * shift Hm k v θ z

Development.AbelianVarietyTheory.BlockTheory.CMModuli.HostConjugateIdeal

theorem **isCMField_complexConj_s_eq_neg** : IsCMField.complexConj E H.s = -H.s

theorem **mem_Psi_or** (τ : E →+* ℂ) : τ ∈ H.Psi v ∨ (starRingEnd ℂ).comp τ ∈ H.Psi

theorem **isIntegral_half** : IsIntegral ℤ ((1 + H.s) / 2)

theorem **w₀_add_conj** : H.w₀ + conjO E H.w₀ = 1

theorem **conj_mem_K** {x : E} (hx : x ∈ H.K) : IsCMField.complexConj E x ∈ H.K

theorem **Dvd.refl** (L : Level H) : L.Dvd L

theorem **Dvd.trans** {L L₂ L₃ : Level H} (h : L.Dvd L₂) (h' : L₂.Dvd L₃) : L.Dvd L₃

theorem **M'_pos** (L : Level H) : 0 < L.M'

theorem **M₀_coprime** (L : Level H) {p : ℕ} (hp : p.Prime) : Nat.Coprime p (L.M₀ p)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.LevelActionTransitions

theorem **mem_nLa** {n : ℕ} {x : D.La} : x ∈ nLa D n ↔ ∃ y, x = n • y

theorem **cls_eq** {n : ℕ+} {a b : D.La} : cls D n a = cls D n b ↔ ∃ y, a = b + (n : ℕ) • y

theorem **neg_mem_KMax** : (fun n => LinearEquiv.neg (𝒪 E)) ∈ KMax D

theorem **exists_sends** (l : Level H) (g : KMax D) (a : D.La) : ∃ b, Sends l g a b

theorem **isNormalisation_normalization** {U Y : Scheme.{0}} (f : U ⟶ Y) [QuasiCompact f] [QuasiSeparated f] : IsNormalisation f.toNormalization f.fromNormalization

theorem **IsNormalisation.isIntegralHom** {U Z Y : Scheme.{0}} {i : U ⟶ Z} {p : Z ⟶ Y} (h : IsNormalisation i p) : IsIntegralHom p

theorem **IsLevelAction.isIso** {m : Model univ h φ} {act lift act₀} (ρ : IsLevelAction m act lift act₀) (g : KMax D) : IsIso (act g)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.LieCharpolyRootPairs

lemma **origin_scale** (t : Spec (CommRingCat.of k) ⟶ S) (c : k) : origin t ≫ scale t c = origin t

lemma **IsRootPair.side** [CharZero k] {s : O} {δ : k} (h : IsRootPair s δ) (τ : O →+* k) : Xor' (τ s = δ) (τ s = -δ)

lemma **DetCond.swap** [IsAlgClosed k] [CharZero k] {ρ : OAct O G} {t : Spec (CommRingCat.of k) ⟶ S} {s : O} {δ : k} {p q : ℕ} (h : DetCond ρ t s δ p q) : DetCond ρ t s (-δ) q p

Development.AbelianVarietyTheory.BlockTheory.CMModuli.OpenModelSections

theorem **nonzeroOnFibre_iff** {V : X.Opens} (θ : ∀ i, Γ(G.left, G.hom ⁻¹ᵁ V ⊓ L.U i)) (x : X) [IsReduced (Scheme.Hom.fiber G.hom x)] : NonzeroOnFibre G L θ x ↔ ∃ (i : L.ι) (g : G.left), G.hom g = x ∧ g ∈ G.left.basicOpen (θ i)

theorem **incl_over** : m.incl ≫ m.s = baseChangeStr str h ≫ spMap φ

theorem **secMap_over** (σ : Over.mk (baseChangeFst str h) ⟶ univ.t.G) : m.secMap σ ≫ m.t.G.hom = m.incl

theorem **extends_one** (U : m.X.Opens) : m.Extends (toUnit _ ≫ MonObj.one) U

theorem **incl_mem_abLocus** (y : 𝟙(baseChange str h)) : m.incl y ∈ abLocus m.t.G

Development.AbelianVarietyTheory.BlockTheory.CMModuli.PeriodMapPresentation

theorem **domain_nonempty** (D : Datum H) : (domain D).Nonempty

theorem **IsPeriodMap.map_smul** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {P : D.La →+ (Fin g → ℂ)} (h : IsPeriodMap D ty P) (a : 𝒪 E) (x : D.La) : P (a • x) = tyAct ty a (P x)

theorem **Presents.unique** {D : Datum H} {g : ℕ} {ty ty' : Fin g → (E →+* ℂ)} {P : D.La →+ (Fin g → ℂ)} {J J' : ℝ ⊗[ℤ] D.La →ₗ[ℝ] ℝ ⊗[ℤ] D.La} (h : Presents D ty P J) (h' : Presents D ty' P J') : J = J'

Development.AbelianVarietyTheory.BlockTheory.CMModuli.PolarizationCorrespondence

@[simp] lemma **epsHom_fst** (f : R →+* R') (c : R') (x : DualNumber R) : (epsHom f c x).fst = f x.fst

@[simp] lemma **epsHom_snd** (f : R →+* R') (c : R') (x : DualNumber R) : (epsHom f c x).snd = c * f x.snd

lemma **epsZero_scale** (c : R) : epsZero R ≫ epsScale c = epsZero R

Development.AbelianVarietyTheory.BlockTheory.CMModuli.PositiveSignKForm

theorem **IsPosSign.unique** {D : Datum H} [Nontrivial D.La] {ε ε' : ℤˣ} (h : IsPosSign D ε) (h' : IsPosSign D ε') : ε = ε'

Development.AbelianVarietyTheory.BlockTheory.CMModuli.PreUniformization

theorem **pow_base_eq** {k : ℕ} (p : QkPt D F k) : p.g.b ^ k = toUnit _ ≫ D.c p.y

theorem **Ph.unique** {k : ℕ} {y : D.X} {g : GtPtF D F} {w} {s s₁ s₂ : RawFam D F k} (h₁ : Ph D F k y g w s s₁) (h₂ : Ph D F k y g w s s₂) : s₁ = s₂

theorem **gm_res_pull** {X Y Z : Scheme.{0}} (f : Y ⟶ X) (U : X.Opens) (a : Γ(X, U)ˣ) {V : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (r : Z ⟶ Y) (z : Z) (hz : r z ∈ V) : gm r z V (res h (pullU f U a)) = gm (r ≫ f) z U a

theorem **gm_of_res₂** {Y Z : Scheme.{0}} {U₁ U₂ : Y.Opens} {a₁ : Γ(Y, U₁)ˣ} {a₂ : Γ(Y, U₂)ˣ} (H : ∀ (V : Y.Opens) (h₁ : V ≤ U₁) (h₂ : V ≤ U₂), res h₁ a₁ = res h₂ a₂) (r : Z ⟶ Y) (z : Z) (m₁ : r z ∈ U₁) (m₂ : r z ∈ U₂) : gm r z U₁ a₁ = gm r z U₂ a₂

theorem **gm_of_res₃** {Y Z : Scheme.{0}} {U₁ U₂ U₃ : Y.Opens} {a₁ : Γ(Y, U₁)ˣ} {a₂ : Γ(Y, U₂)ˣ} {a₃ : Γ(Y, U₃)ˣ} (H : ∀ (V : Y.Opens) (h₁ : V ≤ U₁) (h₂ : V ≤ U₂) (h₃ : V ≤ U₃), res h₁ a₁ * res h₂ a₂ = res h₃ a₃) (r : Z ⟶ Y) (z : Z) (m₁ : r z ∈ U₁) (m₂ : r z ∈ U₂) (m₃ : r z ∈ U₃) : gm r z U₁ a₁ * gm r z U₂ a₂ = gm r z U₃ a₃

theorem **gm_of_res₄** {Y Z : Scheme.{0}} {U₁ U₂ U₃ U₄ : Y.Opens} {a₁ : Γ(Y, U₁)ˣ} {a₂ : Γ(Y, U₂)ˣ} {a₃ : Γ(Y, U₃)ˣ} {a₄ : Γ(Y, U₄)ˣ} (H : ∀ (V : Y.Opens) (h₁ : V ≤ U₁) (h₂ : V ≤ U₂) (h₃ : V ≤ U₃) (h₄ : V ≤ U₄), res h₁ a₁ * res h₂ a₂ = res h₃ a₃ * res h₄ a₄) (r : Z ⟶ Y) (z : Z) (m₁ : r z ∈ U₁) (m₂ : r z ∈ U₂) (m₃ : r z ∈ U₃) (m₄ : r z ∈ U₄) : gm r z U₁ a₁ * gm r z U₂ a₂ = gm r z U₃ a₃ * gm r z U₄ a₄

theorem **isoV_of** {X₁ X₂ Y Z : Scheme.{0}} {L₁ : LB X₁} {L₂ : LB X₂} {f₁ : Y ⟶ X₁} {f₂ : Y ⟶ X₂} (w : LB.Iso (L₁.pull f₁) (L₂.pull f₂)) (r : Z ⟶ Y) (z : Z) (J₁ : (Z ⟶ X₁) → L₁.ι) (J₂ : (Z ⟶ X₂) → L₂.ι) {a₁ a₂} (h₁ : r ≫ f₁ = a₁) (h₂ : r ≫ f₂ = a₂) : isoV w r z J₁ J₂ = gm r z ((L₂.pull f₂).U (fun _ => J₂ a₂) ⊓ (L₁.pull f₁).U (fun _ => J₁ a₁)) (w.s (fun _ => J₂ a₂, fun _ => J₁ a₁))

theorem **mem_iso** {X₁ X₂ Y Z : Scheme.{0}} {L₁ : LB X₁} {L₂ : LB X₂} {f₁ : Y ⟶ X₁} {f₂ : Y ⟶ X₂} (r : Z ⟶ Y) {z : Z} {J₁ : (Z ⟶ X₁) → L₁.ι} {J₂ : (Z ⟶ X₂) → L₂.ι} (g₁ : Good L₁ z J₁) (g₂ : Good L₂ z J₂) {a₁ a₂} (h₁ : r ≫ f₁ = a₁) (h₂ : r ≫ f₂ = a₂) : r z ∈ (L₂.pull f₂).U (fun _ => J₂ a₂) ⊓ (L₁.pull f₁).U (fun _ => J₁ a₁)

theorem **baseHom_left** {D : DegPre R} {e : ℕ} (g : GtCoord I D e G) {T : Over (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))} (x : T ⟶ thickObj R I e G) : (baseHom g T x).left ≫ x.left ≫ g.pG.left

theorem **red_one** (e : ℕ) : eSec (thickObj R I e G) ≫ (fstG I e G).left = thick R I e ≫ eSec G

theorem **red_inv** (e : ℕ) : (𝟙[thickObj R I e G]).left ≫ (fstG I e G).left = (fstG I e G).left ≫ (𝟙[G]).left

theorem **base_one** {D : DegPre R} {e : ℕ} (g : GtCoord I D e G) : eSec (thickObj R I e G) ≫ g.pG.left = thick R I e ≫ eSec D.P.A

theorem **base_inv** {D : DegPre R} {e : ℕ} (g : GtCoord I D e G) : (𝟙[thickObj R I e G]).left ≫ g.pG.left = g.pG.left ≫ (𝟙[D.P.A]).left

theorem **unifPre_rig** (e : ℕ) (Z : Scheme.{0}) (r : Z ⟶ Spec (CommRingCat.of (R ⧸ I ^ (e + 1)))) (z : Z) (JL) (JM) (hL : Good U.LG.L z JL) (hM : Good D.toPre.P.L z JM) : val (φ

theorem **unifPre_even** (e : ℕ) (Z : Scheme.{0}) (r : Z ⟶ (thickObj R I e G).left) (z : Z) (JL) (JM) (hL : Good U.LG.L z JL) (hM : Good D.toPre.P.L z JM) : val D.toPre.sym (r ≫ (U.gt e).pG.left) z JM * isoV (U.w e) r z JL JM = isoV (U.w e) (r ≫ (𝟙[thickObj R I e G]).left) z JL JM * val U.symG (r ≫ (fstG I e G).left) z JL

theorem **unifPre_compat** (e : ℕ) (Z : Scheme.{0}) (r : Z ⟶ (thickObj R I e G).left) (z : Z) (JL) (JM) (hL : Good U.LG.L z JL) (hM : Good D.toPre.P.L z JM) : isoV (U.w (e + 1)) (r ≫ thickMap R I e G) z JL JM = isoV (U.w e) r z JL JM

theorem **psum_three_comp** (e : ℕ) {H : Over (Spec (CommRingCat.of R))} [GrpObj H] [IsCommMonObj H] (p : Red I e G ⟶ H) {T : Over (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))} (β : (T ⟶ thickObj R I e G) →* (Over.mk (T.hom ≫ thick R I e) ⟶ H)) (hβ : ∀ x, (β x).left = x.left ≫ p.left) (x y u : T ⟶ thickObj R I e G) (k : thK) : (psum (thickObj R I e G) ![x, y, u] k.down).left ≫ p.left = pt3 I e p x y u ≫ thφ H (x3 H) k

theorem **unifPre_cub** (e : ℕ) (T : Over (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))) (x y u : T ⟶ thickObj R I e G) (z : T.left) (JL) (JM) (hL : Good U.LG.L z JL) (hM : Good D.toPre.P.L z JM) : val (L

Development.AbelianVarietyTheory.BlockTheory.CMModuli.PrimitiveExponentPullback

theorem **lamAt_fst_snd** (L : LB G.left) : lamAt G L (fst G G) (snd G G) = abelian₂ G L

theorem **lamPt_comp** {T T' : Over S} (h : T' ⟶ T) (x z : T ⟶ G) (I : LamK) : h.left ≫ lamPt G x z I = lamPt G (h ≫ x) (h ≫ z) I

theorem **sum_lamK** (f : LamK → ℤ) : ∑ I, f I = f (∅) + f ({0}) + f ({1}) + f ({0, 1})

theorem **lamPt_values** {T : Over S} (x z : T ⟶ G) : lamPt G x z (∅) = (1 : T ⟶ G).left ∧ lamPt G x z ({0}) = x.left ∧ lamPt G x z ({1}) = z.left ∧ lamPt G x z ({0, 1}) = (x ∗ z).left

theorem **sum_lam** {T : Over S} (x z : T ⟶ G) (h : x = 1 ∨ z = 1) (ψ : T.left ⟶ G.left) : (∑ I, if lamPt G x z I = ψ then lamExp I else 0) = 0

theorem **sameSum_adj** (a b : G ⟶ G) [IsMonHom a] [IsMonHom b] {T : Over S} (h : T ⟶ G ⊗ G) (hh : h ≫ fst G G = 1 ∨ h ≫ snd G G = 1) : LB.SameSum (fun k => h.left ≫ adjPt G a b k) adjExp (PEmpty.elim : PEmpty.{1} → (T.left ⟶ G.left)) PEmpty.elim

theorem **sameSum_prim** (w : G ⟶ G) : LB.SameSum (fun k => eSec G ≫ primPt G w k) primExp (fun _ : PUnit.{1} => eSec G) (fun _ => 1)

theorem **PullIso.toPull_surjective** {t' : Triple H S'} {t : Triple H S} {f : S' ⟶ S} : Function.Surjective (PullIso.toPull (t'

theorem **cubeMap_coord** {G' : Over S'} {G : Over S} [GrpObj G'] [IsCommMonObj G'] [GrpObj G] [IsCommMonObj G] (m : CubeMap G' G) (i : Fin 3) : m.F3 ≫ (x3 G i).left = (x3 G' i).left ≫ m.F

theorem **cubeMap_tripleProduct_unique** {G' : Over S'} {G : Over S} [GrpObj G'] [IsCommMonObj G'] [GrpObj G] [IsCommMonObj G] (m m' : CubeMap G' G) (h : m.F = m'.F) : m.F3 = m'.F3

theorem **res_eq_one** {X : Scheme.{0}} {U V : X.Opens} (h : V ≤ U) {x : Γ(X, U)ˣ} (hx : x = 1) : res h x = 1

theorem **pullU_eq_one** {X Y : Scheme.{0}} (f : Y ⟶ X) (U : X.Opens) {x : Γ(X, U)ˣ} (hx : x = 1) : pullU f U x = 1

theorem **mul_eq_one_of_both** {M : Type} [Monoid M] {x y : M} (hx : x = 1) (hy : y = 1) : x ∗ y = 1

theorem **zpow_eq_one_of** {M : Type} [Group M] {x : M} (hx : x = 1) (n : ℤ) : x ^ n = 1

theorem **isGen_one** {X : Scheme.{0}} (L : LB X) (hg : ∀ i j, L.g i j = 1) : ((fun _ => 1) : LB.Gen L).IsGen

theorem **linComb_g_one** {X Y : Scheme.{0}} (L : LB X) (hg : ∀ i j, L.g i j = 1) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (i j) : (LB.LinComb L φ ε).g i j = 1

theorem **linCombIso_one** {X Y : Scheme.{0}} (L M : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (p) : (LB.linCombIso (L

theorem **corr_one** {X Y : Scheme.{0}} (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' : LB.Gen (LB.LinComb L φ' ε')} (h : ∀ j, t.s j = 1) (h' : ∀ j, t'.s j = 1) : LB.Gen.Corr L φ ε φ' ε' t t'

Development.AbelianVarietyTheory.BlockTheory.CMModuli.ProjectiveSpaceOverSpec

theorem **isStrictHenselisation_real_complex** : IsStrictHenselisation (algebraMap ℝ ℂ)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.RelativeGroupHatForm

theorem **zeta_pow_div** {m n : ℕ} (hn : 0 < n) (h : m ∣ n) : zeta n ^ (n / m) = zeta m

theorem **KA_le** {l l₂ : Level H} (h : l.Dvd l₂) : KA l ≤ KA l₂

theorem **spMap_inclusion_comp** {A B C : IntermediateField ℚ ℂ} (h₁ : A ≤ B) (h₂ : B ≤ C) : spMap (IntermediateField.inclusion h₂).toRingHom ≫ spMap (IntermediateField.inclusion h₁).toRingHom = spMap (IntermediateField.inclusion (h₁.trans h₂)).toRingHom

Development.AbelianVarietyTheory.BlockTheory.CMModuli.SiegelFamilyZetaUnit

theorem **zetaElt_isPrimitiveRoot** (p : ℕ) {M : ℕ} (hM : M ≠ 0) : IsPrimitiveRoot (zetaElt p M) M

theorem **mem_intRing** {C : Type} [NormedField C] [IsUltrametricDist C] (x : C) : x ∈ intRing C ↔ ‖x‖ ≤ 1

Development.AbelianVarietyTheory.BlockTheory.CMModuli.SymplecticBasisLevel

theorem **exists_typeFn** (D : Datum H) : ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)), IsTypeFn D ty

theorem **incl_over** (l : Level H) : (KA l).val.toRingHom (baseKA l).ρ = H.rt0

theorem **SamePoint.refl** (D' : Datum H) (l : Level H) {g : ℕ} (P : D'.La →+ (Fin g → ℂ)) : SamePoint D' l P P

Development.AbelianVarietyTheory.BlockTheory.CMModuli.TateUniformizationInput

lemma **coeffRt_val** : ((coeffRt c : c.F) : ℂ) = H.rt0

lemma **coeffZeta_val** : ((coeffZeta c : c.F) : ℂ) = zeta l.A

lemma **isFracImage_V** (W : ValuationSubring ℂ) : IsFracImage (c.V W).subtype

lemma **IsFracImage.ext** {O K : Type} [CommRing O] [Field K] {β : O →+* c.F} (h : IsFracImage β) {gam gam' : c.F →+* K} (hgam : ∀ x, gam (β x) = gam' (β x)) : gam = gam'

lemma **gam_unique** (x x' : ArcInput D c β O' b) : x.gam = x'.gam

lemma **gam_rt** (x : ArcInput D c β O' b) (r : O) (hr : ((β r : c.F) : ℂ) = H.rt0) : x.gam (coeffRt c) = algebraMap O' (FractionRing O') (b r)

lemma **gam_zeta** (x : ArcInput D c β O' b) (z : O) (hz : ((β z : c.F) : ℂ) = zeta l.A) : x.gam (coeffZeta c) = algebraMap O' (FractionRing O') (b z)

lemma **charZero** (x : ArcInput D c β O' b) : CharZero (FractionRing O')

Development.AbelianVarietyTheory.BlockTheory.CMModuli.ThetaCosetLabels

theorem **CosetUnit.cosetCond** {n N' : ℕ} {R : Type} [CommRing R] {v : Lab g (N' ^ 2) → R} (h : CosetUnit n N' v) : CosetCond n N' v

theorem **CosetCond.cosetUnit** {n N' : ℕ} {R : Type} [CommRing R] [IsLocalRing R] {v : Lab g (N' ^ 2) → R} (h : CosetCond n N' v) : CosetUnit n N' v

Development.AbelianVarietyTheory.BlockTheory.CMModuli.ThetaTowerRecognition

theorem **qPt_unique** (ν : NumOf D C N) {k : ℕ} {p : QkPt D C k} {q q' : N.QT k} (h : ν.QPt p q) (h' : ν.QPt p q') : q = q'

theorem **ThetaSum.hasSum** {ν : NumOf D C N} {m : ℕ} {p : QkPt D C (m ^ 2)} {v : C} (h : ThetaSum ν m p v) {q : N.QT (m ^ 2)} (hq : ν.QPt p q) : Summable (N.txT m q) ∧ N.tx m q = v

theorem **ThetaSeriesSt.prime** (h : ThetaSeriesSt) : ThetaSeriesPrimeSt

Development.AbelianVarietyTheory.BlockTheory.CMModuli.TorusGraphMaps

lemma **PsiGenOn.mono** {D : DegPre R} {χ : D.X} {U U' : Set (Spec (CommRingCat.of R))} (h : PsiGenOn D χ U) (hU : U' ⊆ U) : PsiGenOn D χ U'

lemma **bar_bar** (H : Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

Development.AbelianVarietyTheory.BlockTheory.CMModuli.UniformTorsionIsomorphism

lemma **toGen_fst** {G : Over (Spec (CommRingCat.of R))} (x : TF (R

lemma **toGen_injective** {G : Over (Spec (CommRingCat.of R))} : Function.Injective (toGen (R

lemma **ptA_injective** : Function.Injective (u.ptA F)

Development.AbelianVarietyTheory.BlockTheory.CMModuli.UniformityLowerBounds

theorem **level_ne_zero** (S : ThetaSys o Ω) : (S.r * S.N) ^ 2 ≠ 0

theorem **intRing_injective** (K : IntermediateField ℚ_[p] (PadicAlgCl p)) : Function.Injective ((algebraMap K (PadicAlgCl p)).comp (intRing K).subtype)

Development.AbelianVarietyTheory.BlockTheory.ClassSums.ReadingWithSpecialisation

theorem **IsModulus.unique** {X : Data E r} {bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} {idx : Fin g ≃ X.Ψ × Fin r} {w : E} {v : TF n g} {ρ ρ' : ℝ} (h : IsModulus X bas idx w v ρ) (h' : IsModulus X bas idx w v ρ') {x : Frame X.Ψ r} (hx : x ∈ X.dom) (hu : IsUnitFr X x) {θ : (Fin g → ℂ) → ℂ} (h1 : IsTheta1 X bas idx w x X.Λ θ) (h2 : MeanOne X bas idx x X.Λ θ) {z : Fin n → Fin g → ℂ} (hz : thA X bas idx v x z ≠ 0) : ρ = ρ'

Development.AbelianVarietyTheory.BlockTheory.ClassSums.StabilizerKernelSums

theorem **mt_mul** (V : HSp D) (g h : GA D V) : mt V (g * h) = mt V g * mt V h

theorem **mt_one** (V : HSp D) : mt V 1 = 1

theorem **triv_iff** (V : HSp D) (a : AE D) (g : GA D V) : (∀ i j, a * (mt V g - 1) i j = 0) ↔ a • (mt V g - 1) = 0

theorem **smul_sub_one_eq_zero_of_left_inv** {R n : Type} [CommRing R] [Fintype n] [DecidableEq n] (a : R) (A B : Matrix n n R) (h : B * A = 1) (ha : a • (A - 1) = 0) : a • (B - 1) = 0

theorem **triv_inv** (V : HSp D) (a : AE D) (g : GA D V) (h : ∀ i j, a * (mt V g - 1) i j = 0) : ∀ i j, a * (mt V g⁻¹ - 1) i j = 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.AdaptedStandardBasis

theorem **ShortSpan.mono** {x : ℝ} {G : WTZ E} {S : Set (Rul Xo c)} {A A' B B' : ℝ} (h : ShortSpan U x G S A A') (hA : A ≤ B) (hA' : A' ≤ B') : ShortSpan U x G S B B'

theorem **GminWithDefinite.gmin** (h : GminWithDefinite) : Gmin

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.AdelicDegreeTameRamification

theorem **quadChar_ne_one** (H : Host E) : quadChar H ≠ 1

theorem **LFunction_one_ne_zero** (H : Host E) : (quadChar H).LFunction 1 ≠ 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.AdelicScalarMultiplier

theorem **simLevel_mono** {E : Type} [Field E] [NumberField E] {W : Type} [AddCommGroup W] [Module ℚ W] [Module E W] [IsScalarTower ℚ E W] [Module (𝒪 E) W] (h : LinearMap.BilinMap ℚ W E) (ξ : E) (Λ : Submodule (𝒪 E) W) {I J : Ideal (𝒪 E)} (hIJ : I ≤ J) : simLevel h ξ Λ I ≤ simLevel h ξ Λ J

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.BadPlacesGoodCoefficients

theorem **mem_badPlaces** (v : HeightOneSpectrum (𝒪 D.host.F)) : v ∈ badPlaces D ↔ ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal

theorem **rank_le_first_space_rank** (D : D7) : D.R.r ≤ (D.sp.Y 0).rank

theorem **r_le_rank_Wc** (D : D7) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : D.R.r ≤ (D.sp.W j).rank

theorem **rank_W_eq** (D : D7) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : (D.sp.W j).rank = (D.sp.Y 0).rank

theorem **DimParts.matched** (h : DimParts) : DimPartsMatched

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.BigBlockCharacters

theorem **isFrameOf_self** {u : Fin i → Fin d → C} (hu : IsSatFrame v u) : IsFrameOf v u u

theorem **v_pairDet_eq** {u w : Fin i → Fin d → C} (hu : IsSatFrame v u) (hw : IsFrameOf v u w) (l : Fin i → Fin d → C) : v (pairDet l w) = v (pairDet l u)

theorem **hasChars_iff** {G : LogFGL v d} {u : Fin i → Fin d → C} (hu : IsSatFrame v u) (r : EReal) : HasChars G u r ↔ ∃ l : Fin i → Fin d → C, (∀ b, IsCharLog G (l b)) ∧ toE (v (pairDet l u)) ≤ r

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.BlockEmbeddingFactors

theorem **blockMat_in** (b : Fin r → ℕ) (c : ℕ) (g : Matrix (Block b c) (Block b c) F) (i j : Block b c) : blockMat b c g i.1 j.1 = g i j

theorem **blockMat_out** (b : Fin r → ℕ) (c : ℕ) (g : Matrix (Block b c) (Block b c) F) (i j : Fin r) (h : ¬ (b i = c ∧ b j = c)) : blockMat b c g i j = (1 : Matrix (Fin r) (Fin r) F) i j

theorem **blockEmb_in** (b : Fin r → ℕ) (c : ℕ) {d : ℕ} (h : Fintype.card (Block b c) = d) (g : GL (Fin d) F) (i j : Fin d) : ((blockEmb b c h g : GL (Fin r) F) : Matrix (Fin r) (Fin r) F) (monoEquivOfFin (Block b c) h i).1 (monoEquivOfFin (Block b c) h j).1 = (g : Matrix (Fin d) (Fin d) F) i j

theorem **blockEmb_out** (b : Fin r → ℕ) (c : ℕ) {d : ℕ} (h : Fintype.card (Block b c) = d) (g : GL (Fin d) F) (i j : Fin r) (hij : ¬ (b i = c ∧ b j = c)) : ((blockEmb b c h g : GL (Fin r) F) : Matrix (Fin r) (Fin r) F) i j = (1 : Matrix (Fin r) (Fin r) F) i j

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.BlockProductIsomorphism

theorem **incl_transpose_mul** (e : Fin d ↪ Fin r) : (incl F e)ᵀ ∗ incl F e = 1

theorem **extendMat_apply** (e : Fin d ↪ Fin r) (g : Matrix (Fin d) (Fin d) F) (i j : Fin r) : extendMat e g i j = (1 : Matrix (Fin r) (Fin r) F) i j + ∑ a', ∑ a, (if i = e a then 1 else 0) ∗ (g − 1) a a') ∗ (if j = e a' then 1 else 0)

theorem **extendMat_in** (e : Fin d ↪ Fin r) (g : Matrix (Fin d) (Fin d) F) (a a' : Fin d) : extendMat e g (e a) (e a') = g a a'

theorem **extendMat_out** (e : Fin d ↪ Fin r) (g : Matrix (Fin d) (Fin d) F) (i j : Fin r) (h : i ∉ Set.range e ∨ j ∉ Set.range e) : extendMat e g i j = (1 : Matrix (Fin r) (Fin r) F) i j

theorem **blockEnum_mem** (b : Fin r → ℕ) (c : ℕ) (h : (block b c).card = d) (a : Fin d) : b (blockEnum b c h a) = c

theorem **isoGL_iff** {W W' : Type} [AddCommGroup W] [Module ℂ W] [AddCommGroup W'] [Module ℂ W'] (ρ : Representation ℂ (GL (Fin d) F) W) (ρ' : Representation ℂ (GL (Fin d) F) W') : IsoGL ρ ρ' ↔ Nonempty (Representation.Equiv ρ ρ')

theorem **isIsotypicOf_self** {G V : Type} [Group G] [AddCommGroup V] [Module ℂ V] (τ : Representation ℂ G V) : IsIsotypicOf τ τ

theorem **trivial_irreducible** (G : Type) [Group G] : (Representation.trivial ℂ G ℂ).IsIrreducible

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.BoundaryPolynomialFunctions

theorem **regularLocus_univ** (A : FinHolAtlas d X) : regularLocus A Set.univ d = Set.univ

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.BracketAbsoluteSums

theorem **finite_over** (D : D7) (p : ℕ) : {w : D.what₂.places.Pl | D.what₂.places.resChar w = p}.Finite

theorem **GamDvd.ne_zero** {D : D7} (h : GamDvd D) (u : D.what₂.Word) : (D.mem u).C.DGam ≠ 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ColumnDataLaws

theorem **σ_eq** : Y.σ = T.σ₀

theorem **phi_apply** (cc : Y.U.Comp → ℂ) (f : Rul (T.Q.Xo ((i : ℕ) + 1)) Y.c') (x : GA D (Wc D i)) : Y.phi cc f x = cc (Y.dict x).1 ∗ Y.mu x ∗ rdU (Y.U' Y.σ') f (Y.dict x).1 (Y.dict x).2

theorem **EquivFrameAt.equivAt** (h : Y.EquivFrameAt R) : Y.EquivAt R

theorem **LevelEquiv.smul** {cc : Y.U.Comp → ℂ} (h : Y.LevelEquiv R cc) (u : ℂ) : Y.LevelEquiv R fun k => u ∗ cc k

theorem **exists_dictDat** (h : LawsJ T) : ∃ d : DictDat Y.U, d.Is (Wc D i) (rank_Wc i) Y.dict Y.ι (T.x₀ ((i : ℕ) + 1) Y.σ)

theorem **EquivRow.bare** (h : EquivRow) : EquivRowBare

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.CompletedLStripBounds

theorem **IsCompletedL.unique** {χ : HChar E} {n : I → ℤ} {f g : ℂ → ℂ} (hf : IsCompletedL χ n f) (hg : IsCompletedL χ n g) : f = g

theorem **IsCompletedL.eq_mul** {χ : HChar E} {n : I → ℤ} {Λ L : ℂ → ℂ} (h : IsCompletedL χ n Λ) (hL : IsHeckeL χ L) {s : ℂ} (hs : 1 < s.re) : Λ s = archFactor n s ∗ L s

theorem **archFactor_two_mul** (m : I → ℤ) (s : ℂ) : archFactor (fun i => 2 ∗ m i) s = LInf m s

theorem **CompletedLBoundedInStrips.forall** (h : CompletedLBoundedInStrips) [IsTotallyComplex E] {φ : I → E →+∗ ℂ} (hφ : IsCMType φ) {χ : HChar E} {n : I → ℤ} (hu : IsUnitary χ) (hi : Formalism.L2.Host.HChar.HasInf χ φ n) (hn : ∃ i, n i ≠ 0) {Λ : ℂ → ℂ} (hΛ : IsCompletedL χ n Λ) : BoundedInStrips Λ

theorem **CompletedLBoundedInStrips.even** (h : CompletedLBoundedInStrips) : CompletedLBoundedInStripsEven

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ComponentMassSum

theorem **ComponentSum.deep** (h : ComponentSum) : ComponentSumDeep

theorem **finite_comp** : Finite (Comp Xo c)

theorem **partLat_le** (Z : Comp Xo c) (G : WTZ D.host.E) : partLat U Z G ≤ latW U G

theorem **isComplexData_of_laws** {T : Core D} (h : LawsM T) {j : ℕ} (hj : InR D j) : IsComplexData (T.U j) (T.V j) (T.dict j)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ConjugateLiftSL

theorem **sign_at_base_place_eq_embedding** : D.what₂.embLL = sg0 D A

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ConjugateOrderData

theorem **condExp_bdd** (ϱ : Ideal (OF E)) {l : ℕ} (hl : l ≠ 0) : BddAbove (expIn ϱ '' {v : HeightOneSpectrum (OF E) | (l : OF E) ∈ v.asIdeal})

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.CovariantDbarLeibniz

theorem **homCovDel_leibniz** (A : E → Matrix m m ℂ) (B : E → Matrix l l ℂ) {T : E → Matrix l m ℂ} {S : E → Matrix m k ℂ} {z : E} (hT : MContDiffAt 1 T z) (hS : MContDiffAt 1 S z) (ξ : E) : mdel (fun x => T x ∗ S x) z ξ + conn B z ξ ∗ (T z ∗ S z) = homCovDel A B T z ξ ∗ S z + T z ∗ (mdel S z ξ + conn A z ξ ∗ S z)

theorem **contDiff_tautPairGram** (a b : Unit ⊕ Unit) : ContDiff ℝ ∞ fun x => tautPairGram x a b

theorem **isPosFrame_tautPairGram** : IsPosFrame Set.univ tautPairGram

theorem **vertForm_tautPairGram_zero** (ξ η : ℂ) : vertForm tautPairGram 0 ξ η = kappa ∗ (ξ ∗ conj η)

theorem **lorentzPairGram_spec** : IsHermFrame (Metric.ball 0 1) lorentzPairGram ∧ (∀ z ∈ Metric.ball (0 : ℂ) 1, IsUnit (subField lorentzPairGram z).det) ∧ ¬ (lorentzPairGram 0).PosDef

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.CuspLatticeAnyLevel

theorem **exists_isGoodBasis** (A : FormsL LL Ks WL W G) (J0 : Subgroup G) (lam0 : J0 →∗ LLˣ) (eC : G → LL) [FiniteDimensional LL (A.Wgood J0 lam0 eC)] : ∃ b : Fin (Module.finrank LL (A.Wgood J0 lam0 eC)) → WL, IsGoodBasis A J0 lam0 eC b

theorem **MLat.of_anyLevel** (h : MLatAnyLevel) : MLat

theorem **isKappa_kaWt** (hK : DV.KapDef D) (hn : Even D.R.nV) {j : ℕ} (hj : InR D j) : IsKappa D j (DIV.Cons.kaWt D hK hn j)

theorem **carriesCusp_formsLOf** (T : Core D) (j : ℕ) (w : WeightFamilies.Wt D j) {LL : Type} [Field LL] [NumberField LL] [Algebra T.Q.c.F LL] {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (Comp : (LL →+* ℂ) → Type) (X : ∀ σ, Comp σ → ReciprocityTheory.ChartRestriction.G.FD.Data E r) (la : ∀ σ c, ReciprocityTheory.ChartRestriction.G.FD.Wt (X σ c).Ψ r) (rd : ∀ σ : LL →+* ℂ, Cusp T j w →ₛₗ[σ.comp (algebraMap T.Q.c.F LL)] FDSec (X σ)) (V : (LL →+* ℂ) → _) (dict) (mu) (hL2) (hRd) (h0) (hz) (ha) (hs) {G : Type} [Group G] (ρC : G →* (Cusp T j w →ₗ[T.Q.c.F] Cusp T j w)) (chiC : G →* (T.Q.c.F)ˣ) : CarriesCusp T j w (formsLOf (D

theorem **nrmOf_nonneg** {K : Type} [Field K] [NumberField K] (la : HeightOneSpectrum (𝒪 K)) (v : EReal) : 0 ≤ nrmOf la v

theorem **nrmOf_anti** {K : Type} [Field K] [NumberField K] (la : HeightOneSpectrum (𝒪 K)) {u v : EReal} (hu : u ≠ ⊥) (h : u ≤ v) : nrmOf la v ≤ nrmOf la u

theorem **nrmM_laws** (T : Core D) (j : ℕ) (lp : HeightOneSpectrum (𝒪 T.Q.c.F)) (y : T.PtM j lp) : (∀ f, 0 ≤ T.nrmM j lp y f) ∧ T.nrmM j lp y 0 = 0 ∧ (∀ f g, T.nrmM j lp y (f + g) ≤ max (T.nrmM j lp y f) (T.nrmM j lp y g)) ∧ ∀ (x : T.Q.c.F) f, T.nrmM j lp y (x • f) = FinitePlace.mk lp x * T.nrmM j lp y f

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.CuspValueComparison

theorem **labelChar_one_right** (ω : Fbar g →*₀ ℂ) (d : ℕ) (x : Fbar g) : labelChar ω d x 1 = 1

theorem **pb_ne_zero** (hT : IsTypeDatum gp pb) (i : Fin 2) : pb i ≠ 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.CuspidalInducedCharacter

theorem **finite_boundedLevelClassGroupFamily** {E : Type} [Field E] [NumberField E] (Pg : Finset (Pl E)) (r : ℕ) : Finite (G0 Pg r)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.CuspidalProductMultiplicity

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CuspidalProductMultiplicity.finite_kP** (PP : _root_.Formalism.L2.DIV.Pl E) : Finite (_root_.Formalism.L2.DIV.kP PP)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CuspidalProductMultiplicity.finite_boundedLevelClassGroupFamily** (Pg : Finset (_root_.Formalism.L2.DIV.Pl E)) (r : ℕ) : Finite (_root_.Formalism.L2.DIV.G0 Pg r)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CuspidalProductMultiplicity.nbar_pos** (Pg : Finset (_root_.Formalism.L2.DIV.Pl E)) (k : ℕ) : 0 < _root_.GlobalTheory.IndefiniteRowSideFullHecke.CuspidalProductMultiplicity.nbar Pg k

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.DegreeAdditiveFrobenius

theorem **IsFiniteOfDegree.unique** {C' C : Scheme.{0}} [IsIntegral C'] [IsIntegral C] {π : C' ⟶ C} {m n : ℕ} (h : IsFiniteOfDegree π m) (h' : IsFiniteOfDegree π n) : m = n

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.DescentConstants

theorem **ceilEight_spec** (n : ℕ) : 8 ∣ ceilEight n ∧ n ≤ ceilEight n ∧ ∀ a, 8 ∣ a → n ≤ a → ceilEight n ≤ a

theorem **expQ_spec** : bigR D ^ 7 * degStar D L ^ 2 ≤ (3 : ℝ) ^ (2 * expQ D L)

theorem **expC_spec** : bigR D ^ 8 * degStar D L ^ 2 ≤ (3 : ℝ) ^ expC D L ∧ max 4 (2 * expQ D L) ≤ expC D L

theorem **nU_eq** (j : ℕ) : nU D j = placeRankTwoOne.normU D j (wAn D j)

theorem **uAn_anchor** (j : ℕ) (τ : D.what₂.Tau) : uAn D j τ (D.R.k - 2) = 0

theorem **uOne_anchor** (j : ℕ) (τ : D.what₂.Tau) : uOne Q j τ (D.R.k - 2) = 0

theorem **numAt_hat** (j : ℕ) : (numAt Q j).hat = numAt Q j

theorem **strong_arith** {δ e A PP K fk b : ℝ} (hδ : 0 ≤ δ) (hδe : δ ≤ e) (hA : 0 ≤ A) (h16 : 16 * A ≤ PP) (hK : K ≤ 32 * fk * b) (hfk : 0 ≤ fk) (hb : 0 ≤ b) (h : 385 * PP * fk * e ≤ 1) : 24 * δ * A * K ≤ b / 8

theorem **Choice.x_ge_eps** {Q : ParamsT D L P} {j : ℕ} (h : Choice Q j) : 2 * L.sc.x ^ (L.sc.ez / 7) ≤ L.sc.x ^ (L.sc.eps / 6) ∧ 2 * (L.Ct + 4) * Real.log L.sc.x + 6 ≤ L.sc.x ^ ((L.sc.eps : ℝ) / 6)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.EpsilonPairingZeta

theorem **zeta_succ_pow** (ε : PreTilt O p) (n : ℕ) : zeta ε (n + 1) ^ p = zeta ε n

theorem **IsPrimitiveSystem.zeta_zero** {ε : PreTilt O p} (hε : IsPrimitiveSystem ε) : zeta ε 0 = 1

theorem **epsPairing_smul** [IsDomain O] {ε : PreTilt O p} (hε : IsPrimitiveSystem ε) (G : PDivGroup p O) (t : G.Tate) (χ : G.dual.Tate) : epsPairing hε G t χ • epsTate hε = pairing G O t χ

theorem **pairingO_toO** [IsDomain O] {ε : PreTilt O p} (hε : IsPrimitiveSystem ε) {G : PDivGroup p O} {h : ℕ} (e : G.Tate ≃ₗ[ℤ_[p]] (Fin h → ℤ_[p])) (t : G.Tate) (χ : G.dual.Tate) : pairingO hε e (toO e t) χ = padicToO p O (epsPairing hε G t χ)

theorem **mult_bb** (y : Fin 1 → O) : (mult ε).bb y = y

theorem **mult_Fil** : (mult ε).Fil = ⊥

theorem **muLie_mk** (m : Fin 1 → O) : muLie ε ((mult ε).Fil.mkQ m) (dlogT p O) = m 0

theorem **realizes_mu** (C : Type) [Field C] [Algebra O C] [IsIntegrallyClosedIn O C] [IsAlgClosed C] [CharZero C] : Realizes hε 1 1 (PDivGroup.mu p O) (padicAction _) (mult ε) (muBasis hε) (LinearEquiv.ofSubsingleton _ _) (muLie ε) where

theorem **norm_p** : ‖(p : C)‖ = (p : ℝ)⁻¹

theorem **IsIntegers.pow_dvd_iff** (H : IsIntegers O C) (n : ℕ) (a : O) : (p : O) ^ n ∣ a ↔ ‖algebraMap O C a‖ ≤ (p : ℝ)⁻¹ ^ n

theorem **IsIntegers.not_isUnit** (H : IsIntegers O C) : ¬IsUnit (p : O)

theorem **IsIntegers.isAdicComplete** [CompleteSpace C] (H : IsIntegers O C) : IsAdicComplete (span {(p : O)}) O

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.EqualVolumeMeanCorrection

theorem **IsSlope.unique** {B : ℝ} {u : D.what₂.Tau → Fin Y.N → ℝ} {c : Y.Comp} {a b : ℝ} (ha : IsSlope Y B u c a) (hb : IsSlope Y B u c b) : a = b

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ExtensionSingularValues

theorem **extZero_single** {n₀ n₁ : ℕ} (h : n₀ ≤ n₁) (s : Fin n₀) : extZero n₀ n₁ (Pi.single s 1) = Pi.single (Fin.castLE h s) 1

theorem **subst_mem** {n₀ n₁ : ℕ} (h : n₀ ≤ n₁) (x : Matrix (Fin n₁) (Fin n₀) ℂ) (ν : Fin n₀ → ℕ) {F : MvPolynomial (Fin n₀ × Fin n₀) ℂ} (hF : F ∈ dualWeyl ℂ n₀ n₀ ν) : subst h x F ∈ dualWeyl ℂ n₁ n₁ (rowNu (r

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.FibreFramesOnModels

theorem **isBaseRing_local** (c : Coeff l) (W : ValuationSubring ℂ) : IsBaseRing (c.V W).subtype

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.FiniteDegreeMorphisms

theorem **ScalesDeg.mul_le** {π : C' ⟶ C} {m : ℕ} (h : ScalesDeg k π m) {E : C.Modules} {r : ℕ} (hE : IsVB E r) {P P' : ℤ} (hP : IsMaxSubDeg k E P) (hP' : IsMaxSubDeg k ((Scheme.Modules.pullback π).obj E) P') : m * P ≤ P'

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.FrameConstantTwistFactor

theorem **IsFrameConst.unique** {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} {vac : WO S T → ℂ} {g : GA D T} {th : WO S T → ℂ} {c c' : ℂ} (h : IsFrameConst om vac g th c) (h' : IsFrameConst om vac g th c') : c = c'

theorem **frameVec_ne_zero** (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)) {vac : WO S T → ℂ} (hv : vac ≠ 0) (g : GA D T) : frameVec om vac g ≠ 0

theorem **isFrameConst_self** (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)) {vac : WO S T → ℂ} (hv : vac ≠ 0) (g : GA D T) : IsFrameConst om vac g (frameVec om vac g) 1

theorem **const_iff_frame** (om om' : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)) {vac : WO S T → ℂ} (hv : vac ≠ 0) (g : GA D T) (z a : ℂ) : (∀ th c, IsFrameConst om vac g th c → IsFrameConst om' vac g th (z * c * a)) ↔ frameVec om' vac g = (z * a) • frameVec om vac g

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.FramedFlatCoordinates

theorem **lieMult_iff** {k : Type} [Field k] {O : Type} [CommRing O] (act : OAct O G) (x : Sp k ⟶ S) (χ : O →+* k) (n : ℕ) : LieMult act x χ n ↔ ∃ v : Fin n → (epsAt x ⟶ G), IsEigenBasis act x χ v

theorem **exists_framed** (Y : Model Xo c β) (Xc : (Xo.XF c).Opens) (τ : E →+* ℂ) (hτ : τ (H.rt : E) = H.rt0) (k : Type) [Field k] [IsAlgClosed k] (j : c.F →+* k) (y : Sp k ⟶ Xo.XF c) (hy : y ≫ pullback.snd _ _ = spMap j) (hm : Set.range y ⊆ (Xc : Set (Xo.XF c))) : ∃ a : Framed Y Xc τ k j, a.x = y ≫ Y.jQ

theorem **Rule.add_apply** (s s' : Rule Y Xc τ) (R : Type) [CommRing R] (j : c.F →+* R) (a : Framed Y Xc τ R j) : (s + s') R j a = s R j a + s' R j a

theorem **Rule.zero_apply** (R : Type) [CommRing R] (j : c.F →+* R) (a : Framed Y Xc τ R j) : (0 : Rule Y Xc τ) R j a = 0

theorem **Rule.smul_apply** (t : c.F) (s : Rule Y Xc τ) (R : Type) [CommRing R] (j : c.F →+* R) (a : Framed Y Xc τ R j) : (t • s) R j a = j t * s R j a

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.GenericFiniteMaps

theorem **Statement.primeField** (h : Statement) : StatementPrimeField

theorem **IsProjSurface.finiteCoh** (hT : IsProjSurface k T) (h : IsVB E r) : FiniteCoh k E ∧ FiniteCoh k (dual E) ∧ FiniteCoh k (structureModule T)

theorem **degCh2_eq** (h : IsVB E r) : degCh2 k E = ((chi k E : ℚ) + chi k (dual E)) / 2 − r * (chi k (structureModule T) : ℚ)

theorem **degCh2_congr** (φ : E ≅ F) : degCh2 k E = degCh2 k F

theorem **dualLaw** : DualLaw k T

theorem **IsProjSurface.degCh2_toMod** (hT : IsProjSurface k T) {L : LB T} (hL : L.IsLB) : degCh2 k (toMod L) = (interNum k (structureModule T) ![L, L] : ℚ) / 2

theorem **isGenFinite_id** : IsGenFinite (𝟙 T) 1

theorem **isProjSurface_plane** (k : Type) [Field k] : IsProjSurface k (PSpace k 2)

theorem **degree_ch2_projective_plane_01** (k : Type) [Field k] : degCh2 k (toMod (On k 2 1)) = 1 / 2

theorem **hasFlag_biprod** {L M : LB T} (hL : L.IsLB) (hM : M.IsLB) : HasFlag (toMod L ⊞ toMod M) [M, L]

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.GenericMultiplicityOne

theorem **finite_res** (PP : Pl E) : Finite (kP PP)

theorem **finite_parInd** {F : Type} [Field F] [Finite F] {r : ℕ} {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (b : Fin r → ℕ) (σ : Representation ℂ (blockTri (κ

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.GoodFormsSpanInduced

theorem **actG0_add** (g : Formalism.L2.DIV.G0 L.Pg L.r) (F F' : GA D V → ℂ) (h : GA D V) : Lev.actG0 V β L g (F + F') h = Lev.actG0 V β L g F h + Lev.actG0 V β L g F' h

theorem **actG0_smul** (g : Formalism.L2.DIV.G0 L.Pg L.r) (a : ℂ) (F : GA D V → ℂ) (h : GA D V) : Lev.actG0 V β L g (a • F) h = a * Lev.actG0 V β L g F h

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.GoodIdempotentBadPlaces

theorem **goodTypes_finite** {pb : Fin 2 → 𝒪 D.host.E} (h : IsTypeDatum D.host.g pb) (om : ∀ i : Fin 2, (ZMod (D.host.g i))ˣ →* ℂˣ) (n : ℕ) : (goodTypes h om n).Finite

theorem **mem_badPl** (v : HeightOneSpectrum (𝒪 D.host.F)) : v ∈ badPl D ↔ ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal

theorem **heckePart_le** (c : EigSys D) : heckePart θ h om ψ j β M c ≤ goodForms θ h om ψ j β M

theorem **mem_heckePart** (c : EigSys D) (T : MMap M) : T ∈ heckePart θ h om ψ j β M c ↔ T ∈ goodForms θ h om ψ j β M ∧ ∀ P d R, Hecke₂ (D.sp.W j) (levP0 pb ψ j β) (chiP0 pb ψ j β) (badPl D) P d R → ∀ v, heckeLin (D.sp.W j) R (T v) = c P d • T v

theorem **heckePart_ne_bot** {c : EigSys D} (hc : c ∈ occurring θ h om ψ j β M) : heckePart θ h om ψ j β M c ≠ ⊥

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.GradedDifferentComparison

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedDifferentComparison.exists_generator_different** {v : Valuation C ℝ≥0} {O : Type} [CommRing O] [Algebra O C] (hv : v.Integers O) {E : Type} [Field E] [NumberField E] (τ : E →+* C) : ∃ δ : O, (differentIdeal ℤ (𝒪

E)).map (_root_.GlobalTheory.HTLat.embO hv τ) = Ideal.span {δ} ∧ v (algebraMap O C δ) = _root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedDifferentComparison.absDifferent v τ ∧ _root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedDifferentComparison.absDifferent v τ ≠ 0 ∧ _root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedDifferentComparison.absDifferent v τ ≤ 1

theorem **tower** (p : ℕ) [Fact p.Prime] : @IsScalarTower ℤ_[p] 𝒪_ℂ_[p] ℂ_[p] (algInt p).toSMul Algebra.toSMul (algField p).toSMul

theorem **fact_not_isUnit** (p : ℕ) [Fact p.Prime] : Fact ¬IsUnit (p : 𝒪_ℂ_[p])

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.HalfLatticeTypes

theorem **half_ne_bot** (l : Level H) : l.Mb ≠ ⊥

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.HermitianCurvatureForms

theorem **isPosFrame_tautPairGram** : IsPosFrame Set.univ tautPairGram

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.HermitianInterpolation

theorem **isCutoff_sq** : IsCutoff fun y => Real.smoothTransition (3 - y) ^ 2

theorem **isData_halfPlane** : IsData (ι

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.HodgePeriodGram

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.tlog_pos** {w : ℂ} (h0 : w ≠ 0) (h1 : ‖w‖ < 1) : 0 < _root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.tlog w

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.im_period** (ha : a ≤ n) (τ₀ : (Fin n → ℂ) → Matrix (Fin g) (Fin g) ℂ) (B : Fin a → Matrix (Fin g) (Fin g) ℤ) (z : Fin n → ℂ) (L : Fin a → ℂ) (hL : ∀ i, exp (L i) = z (Fin.castLE ha i)) : (_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.period τ₀ B z L).map Complex.im = _root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.imPeriod ha τ₀ B z

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.conn_hodgeGram** {rk : ℕ} (ha : a ≤ n) {τ₀ : (Fin n → ℂ) → Matrix (Fin g) (Fin g) ℂ} {B : Fin a → Matrix (Fin g) (Fin g) ℤ} (h : _root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.IsPeriodChart rk τ₀ B) {z : Fin n → ℂ} (hz : z ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.polydisc n 1) (ξ : Fin n → ℂ) : _root_.GlobalTheory.HermFrame.conn (_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.hodgeGram ha τ₀ B) z ξ = (_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.hodgeGram ha τ₀ B z)⁻¹ ∗ _root_.GlobalTheory.HermFrame.mdel (_root_.GlobalTheory.IndefiniteRowSideFullHecke.HodgePeriodGram.hodgeGram ha τ₀ B) z ξ

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.HodgeTateKernel

theorem **isNilpotent_dual** {R : Type} [CommRing R] {n : ℕ} (h : (p : R) ^ n = 0) : IsNilpotent (p : DualNumber R)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.HookProductDictionary

theorem **hookProduct_pos** (n : ℕ) (a : ℕ → ℕ) : 0 < hookProduct n a

theorem **finite_levelPlaces** (D : D7) : {v : HeightOneSpectrum (𝒪 D.host.F) | ((D.host.g 0 ∗ D.host.g 1 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal}.Finite

theorem **redV_surj** : Function.Surjective (redV R i)

theorem **exists_isReps** (i : Fin D.R.m) (hP : OMPins T R κ v₀ O i) : ∃ h, IsReps (O i) (v₀ i (sg0 D A)) h

theorem **IsAdelicScalar.unique** {i : Fin D.R.m} {c : EigSys D} {l l' : ℝ} (h : IsAdelicScalar R i c l) (h' : IsAdelicScalar R i c l') : l = l'

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.InfinitePlaceDepthSums

theorem **formFrame_spec** {K : Type} [Field K] [NumberField K] {n : ℕ} (H : Matrix (Fin n) (Fin n) K) (v : InfinitePlace K) (h : (H.map v.embedding).PosDef) : IsUnit (formFrame H v).det ∧ ((formFrame H v)⁻¹).conjTranspose ∗ (formFrame H v)⁻¹ = H.map v.embedding

theorem **occurring_nonempty** (hW : Df.WF) (hp : Df.Pos) : (occurring Df).Nonempty

theorem **le_lamMin** (hW : Df.WF) (S : IndefRows Df) (v : InfinitePlace K) (c : ℝ) (h : ∀ p, Df.Qp Bs.μ0 p ≠ ⊥ → c ≤ ∑ j, Real.log (S.lam j Bs.μ0 p v)) : c ≤ lamMin hW S v

theorem **numLa_eq** (E : Type) [Field E] [NumberField E] {x : ℝ} (ε : ℝ) (hx : 0 < x) : numLa E x ε = numL x ε + Real.log x + Real.log (rootDiscr E)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.IotaEmbeddingColumns

theorem **space_eq** {F : Face D A} (h : Ties T R κ v₀ F) (i : Fin D.R.m) : T.V ((i : ℕ) + 1) T.toCore.σ₀ = Wc D i

theorem **RowAt.norm** {i : Fin D.R.m} {hV : T.V ((i : ℕ) + 1) T.toCore.σ₀ = Wc D i} {Tup : Type} {tup : Tup → T0} {cc : (T.U ((i : ℕ) + 1) T.toCore.σ₀).Comp → ℂ} {ph : ℂ} (h : RowAt T R κ v₀ i hV tup cc ph) : (∀ k, ‖cc k‖ = 1) ∧ ‖ph‖ = 1

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.KappaLatticeEvaluation

theorem **mem_latIn** (C : Submodule K R) (Vs : HeightOneSpectrum (𝒪 K) → Set (R → EReal)) (hV : ∀ la, ∀ v ∈ Vs la, IsVal la v) (f : R) : f ∈ latIn C Vs hV ↔ f ∈ C ∧ ∀ la, ∀ v ∈ Vs la, 0 ≤ v f

theorem **Pinned.of_laws** {T : Core D} (h : LawsM T) : Pinned T

theorem **SupLatticeFull.laws** (h : SupLatticeFull) : SupLatticeFullLaws

theorem **mem_supLat** (T : Core D) (j : ℕ) (w : WeightFamilies.Wt D j) (f : T.R j) : f ∈ supLat T j w ↔ f ∈ cuspW T j w ∧ ∀ (la : HeightOneSpectrum (𝒪 T.Q.c.F)) (V : CDVR) (io : T.Q.c.F →+∗ V.Fr) (e : ℕ), OverLa la V.V io e → ∀ y : IntFr (T.Q.Xo j) T.Q.c V io, y.IsAlg → 0 ≤ nval V.V e (evI y f)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.LatticeNormTransport

theorem **nrm_ne_zero** {K : Type} [Field K] [NumberField K] (𝔞 : (FractionalIdeal (𝒪 K)⁰ K)ˣ) : nrm (𝔞 : FractionalIdeal (𝒪 K)⁰ K) ≠ 0

theorem **lamF_ne_zero** {E : Type} [Field E] [NumberField E] (s : ℕ) (χ : HChar E) (y : (Af E)ˣ) : lamF s χ y ≠ 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.LevelPlusSubgroup

theorem **posDef_dual** (τ : D.what₂.Tau) : (((Xp G).gram.map (tauE D τ))⁻¹).transpose.PosDef

theorem **mem_cols** (Ψ : Bridge.Gf (G.V (.top 0)) → (coefPlus G θ).W) : Ψ ∈ Cols G θ ↔ ∀ (γ : GQ (G.V (.top 0))) g (k : levPlus G), Ψ (Bridge.ratF (G.V (.top 0)) γ * g * k) = (coefPlus G θ).ρ γ (Ψ g)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.LieAlgebraMaps

theorem **isLieLinear_lieMap** {R : Type} [CommRing R] {T G : Over (Spec (CommRingCat.of R))} [GrpObj T] [GrpObj G] (u : T ⟶ G) [IsMonHom u] : IsLieLinear (lieMap u)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.LiftFormsMainTerm

theorem **conjRes_id** (χ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) : conjRes (RingHom.id ℂ) χ = χ

theorem **omegaP_triv** {a b : ℕ} (Pg : Finset (Pl D.host.E)) {r : ℕ} (hr : a = r) (χ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (y : ∀ PP : Pl D.host.E, Matrix (Fin b) (Fin a) (kP PP)) : omegaP (LevelP.triv Pg r ℂ) χ y = omegaK Pg y

theorem **isConjChr_id** (half : Bool) {c : (AE D)ˣ →* Circle} (hc : IsHeckeChar D c) : IsConjChr (RingHom.id ℂ) half c c

theorem **labE_base** (τ : D.what₂.Tau) : labE D.what₂.embLL τ = tauE D τ

theorem **lab_base** (ρ : D.what₂.Tau) : lab D.what₂.embLL ρ = ρ

theorem **isONBasis_onb** {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) (p : ℕ) (h : ∀ a, M.ε τ a = decide ((a : ℕ) < p)) : IsONBasis (gramAt D V τ) (cart M τ) (onb M τ) p

theorem **ofReal_is** {R : RealDat D A Ks} {F : Face D A} (h : R.Char F) (i : Fin D.R.m) (t : T) : (ofReal R i t).Is t (RingHom.id ℂ) R.triv where

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.LinearCoefficientData

theorem **idx_spec** {Pg : Finset (Pl D.host.E)} (h : IsRecPlaces D Pg) (PP : ↥Pg) : D.naut.plGw (idx h PP).1 (idx h PP).2 = PP.1

theorem **idx_mem** {Pg : Finset (Pl D.host.E)} (h : IsRecPlaces D Pg) (PP : ↥Pg) : ((D.host.g (idx h PP).1 : ℕ) : 𝒪 D.host.E) ∈ PP.1.asIdeal

theorem **idx_eq** {Pg : Finset (Pl D.host.E)} (h : IsRecPlaces D Pg) (PP : ↥Pg) (kv : Σ k : Fin 2, D.nd.PlG k) (e : D.naut.plGw kv.1 kv.2 = PP.1) : idx h PP = kv

theorem **rank_zeroth_component_eq** (D : D7) : (D.sp.Y 0).rank = D.R.r

theorem **DimPartsThree.dimParts** (h : DimPartsThree) : DimParts

theorem **exists_isRecPlaces** (D : D7) : ∃ Pg, IsRecPlaces D Pg

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ModelDdcForms

theorem **gradSqAt_nonneg** {e : ℕ} {X : Type} [TopologicalSpace X] (ψ : OpenPartialHomeomorph X (Fin e → ℂ)) (f : X → ℝ) (z w : Fin e → ℂ) : 0 ≤ gradSqAt ψ f z w

theorem **rComp_eq** {X : Type} [TopologicalSpace X] {m : ℕ} (L : Fin m → LocPot X) (s : Fin m → X → X → ℂ) (i : Fin m) (y : X) : rComp L s i y = Real.exp 1 - 2 * logNormSec (L i) (s i) y

theorem **rTot_zero** {X : Type} [TopologicalSpace X] (L : Fin 0 → LocPot X) (s : Fin 0 → X → X → ℂ) (y : X) : rTot L s y = Real.exp 1

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.MultiplicationKernelIdentity

theorem **mem_addValuationIntegerRing_iff** (v : AddValuation C (WithTop ℝ)) (x : C) : x ∈ Int0 v ↔ 0 ≤ v x

theorem **ev_lin** (a : S → O) (s t : S) : ev a s (lin a t) = a s - a t

theorem **v_unit** (v : AddValuation C (WithTop ℝ)) {u : Int0 v} (hu : IsUnit u) : v (u : C) = 0

theorem **RelLen.unique** (v : AddValuation C (WithTop ℝ)) {M : Type} [AddCommGroup M] [Module (Int0 v) M] {X' X : Submodule (Int0 v) M} {r r' : ℝ} (h : RelLen v X' X r) (h' : RelLen v X' X r') : r = r'

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.NormalisedBlowupCone

theorem **IsPolarisation.zero** {q : ℕ} {ψ : ConeFn H} (h : IsPolarisation q ψ) {X : Type} [AddCommGroup X] [Module (𝒪 E) X] (hX : IsCuspLattice E q X) {β : PosForm H X} (e : β.1 = 0) : ψ X β = 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.OmegaFactorization

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.latM_eq_dual** (𝔞 : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Frac F) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.latM 𝔞 = FractionalIdeal.dual ℤ ℚ (𝔞 : FractionalIdeal (𝒪 F)⁰ F)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.unit_inv_mul** (ε : (𝒪 F)ˣ) : (((ε⁻¹ : (𝒪 F)ˣ) : 𝒪 F) : F) * ((ε : 𝒪 F) : F) = 1

theorem **wirtWord_nil** {Φ : Type} [Fintype Φ] (g : (Φ → ℂ) → ℂ) : wirtWord [] g = g

theorem **wirtWord_cons** {Φ : Type} [Fintype Φ] (w : Φ × Bool) (ws : List (Φ × Bool)) (g : (Φ → ℂ) → ℂ) : wirtWord (w :: ws) g = wirt w.1 w.2 (wirtWord ws g)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.prod_emb** [IsTotallyReal F] (x : F) : ∏ τ : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Emb F, ((τ x : ℝ) : ℂ) = ((Algebra.norm ℚ x : ℚ) : ℂ)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.eisTerm_unit** [IsTotallyReal F] {k : ℕ} (hk : Even k) (σ : ℂ) (z : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Emb F → ℂ) (ε : (𝒪 F)ˣ) (a b : F) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.eisTerm k σ z (((ε : 𝒪 F) : F) * a, ((ε : 𝒪 F) : F) * b) = _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.eisTerm k σ z (a, b)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.eisClass_mk** [IsTotallyReal F] (𝔞 𝔟 : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Frac F) {k : ℕ} (hk : Even k) (σ : ℂ) (z : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Emb F → ℂ) (p : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.LatPair 𝔞 𝔟) :

root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.eisClass a b k σ z (Quotient.mk _ p) =
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.eisTerm k σ z p.1

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.qClass_mk** (a b :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Frac F) (k : ℕ) (z :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Emb F → ℂ) (p :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.QPair a b) :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.qClass a b k z (Quotient.mk _ p) =
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.qTerm a k z p.1

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.latPt_ne_zero** {z :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Emb F → ℂ} (hz : z ∈
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.upperHalf F) {p : F × F} (hp : p ≠ 0) (τ :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.Emb F) :
root.GlobalTheory.IndefiniteRowSideFullHecke.OmegaFactorization.latPt z p τ ≠ 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.OneMeasureSizes

theorem **probSq_rdU** (s : T.Q.c.F →+* ℂ) (f : T.R ((i : ℕ) + 1)) : probSq T i s (rdU (T.U ((i : ℕ) + 1) s) f) = petAt (T.U ((i : ℕ) + 1) s) (xiR D) (T.V ((i : ℕ) + 1) s) (T.dict ((i : ℕ) + 1) s) (muQ1 (T.V ((i : ℕ) + 1) s)) (expW D i) f ^ 2

theorem **adFun_rdU** {s : T.Q.c.F →+* ℂ} (hV : T.V ((i : ℕ) + 1) s = Wc D i) (f : T.R ((i : ℕ) + 1)) : adFun T i hV (rdU (T.U ((i : ℕ) + 1) s) f) = phiC T hV (wtA (kapTop D i)) f

theorem **OneMeasure.models** (h : OneMeasure D) {G : Type} [Group G] [Fintype G] {Pl0 Tu : Type} {A : FamIn D.what₂.LL G Pl0 Tu D.R.m} [Algebra T.Q.c.F D.what₂.LL] (R : RealDat D A T.Q.c.F) (hl : R.toLevCh.Is A.μ0) (hm : R.IsModels) (hT : LawsJ T) (ha : AlgTie T) (large₂ : SizeData₂ D) (hs : Sizes D large₂.sc.L) (n : ℝ) (hn : 1 ≤ n) : ∃ c : ℝ, 0 < c ∧ Real.log c ≤ 122 * (D.R.f * (D.R.r : ℝ) ^ 2 * large₂.sc.L) ∧ ∀ (σ : D.what₂.LL →+* ℂ) (hV : T.V ((i : ℕ) + 1) (gd T σ) = Wc D i) (x : (R.HW i σ).H), ∀ f ∈ scalarForms T i (gd T σ), Corr T i σ hV ((R.HW i σ).e x) f → n⁻¹ * ‖x‖ ^ 2 = c * probSq T i (gd T σ) f

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.OrthogonalNormSquareSum

theorem **fockB_self_nonneg** {σ : Type} (f : MvPolynomial σ ℝ) : 0 ≤ fockB f f

theorem **factor_sq** (lam : DomWeight N) (sg : DomWeight n) {T : ℝ} (hT : 0 < T) : factor hN lam sg T ^ 2 = T ^ (-Br.nBr lam sg) * (((Br.cBr hN lam sg : ℤ) : ℝ) ^ 2 * Br.kapBr ℝ hN lam sg)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.PaddedWittNormSquare

theorem **natW_eq_padZero** {a b : ℕ} (hab : a ≤ b) {lam ν : DomWeight a} {lam' : DomWeight b} {s' s : ℤ} (h : IsStepWeight lam ν lam' s' s) (hg : a < b ∨ lastEntry ν = 0) (i : Fin b) : ((lam'.natW i : ℕ) : ℤ) = padZero b ν i

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.PolarisationModuleIndex

@[simp] theorem **coe_latMod** (c : Ideal (OF E)) (y : (Af E)ˣ) : (latMod c y : Set E) = lattice c y

theorem **TrInt.zero** : TrInt (0 : E)

theorem **TrInt.add** {x y : E} (hx : TrInt x) (hy : TrInt y) : TrInt (x + y)

theorem **bar_algebraMap** (a : OF E) : bar (algebraMap (OF E) E a) = algebraMap (OF E) E a

theorem **polMod_mul_le** (L : Submodule (OF E) E) : polMod L * L ≤ dualLat L

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.PolarisedDualSystem

theorem **dualLab_dualLab** (d : List ℤ) : dualLab (dualLab d) = d

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.QuasiPlurisubharmonic

theorem **exp_one_le_rad** {D : ℕ} {Y : Type} [TopologicalSpace Y] {A : FinHolAtlas D Y} {L : LocPot Y} {s : Y → Y → ℂ} {m : ℕ} {W : Fin m → Set Y} (h : IsBoundary A L s W) (y : Y) : Real.exp 1 ≤ rad L s y

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.QuotientEigenDifference

theorem **stmtPadic_of_stmt** (h : Stmt) : PadicComplexThreeCases

theorem **mem_freeLattice_bool** (a : Bool → w.valuationSubring) (h : a true ≠ a false) {r : ℕ} (x : Bool → Fin r → C) : x ∈ freeLattice a r ↔ ∀ i, x false i ∈ w.valuationSubring ∧ (x true i - x false i) / ((a true : C) - a false) ∈ w.valuationSubring

theorem **resPair_apply** (a : Bool → w.valuationSubring) (r : ℕ) (x y : Bool → Fin r → C) : resPair a r x y = ∑ i, (x true i * y true i - x false i * y false i) / ((a true : C) - a false)

theorem **resPair_comm** (a : Bool → w.valuationSubring) (r : ℕ) (x y : Bool → Fin r → C) : resPair a r x y = resPair a r y x

theorem **resPair_perfect** (a : Bool → w.valuationSubring) (h : a true ≠ a false) (r : ℕ) (x : Bool → Fin r → C) : x ∈ freeLattice a r ↔ ∀ y ∈ freeLattice a r, resPair a r x y ∈ w.valuationSubring

theorem **coordFil_isFiltration** (a : Bool → w.valuationSubring) (κ : Bool → Fin 2) : IsFiltration a (coordFil a κ) where

theorem **single_mem_coordFil** (a : Bool → w.valuationSubring) (h : a true ≠ a false) (κ : Bool → Fin 2) (s : Bool) (i : Fin 2) (hi : i ≠ κ s) : (Pi.single s (Pi.single i ((a true : C) - a false)) : Bool → Fin 2 → C) ∈ coordFil a κ

theorem **coordFil_ann** (a : Bool → w.valuationSubring) (h : a true ≠ a false) (κ κ' : Bool → Fin 2) (hκ : ∀ s, κ s ≠ κ' s) (x : Bool → Fin 2 → C) (hx : x ∈ freeLattice a 2) : x ∈ coordFil a κ ↔ ∀ y ∈ coordFil a κ', resPair a 2 x y = 0

theorem **finrank_filSpace_coordFil** (a : Bool → w.valuationSubring) (h : a true ≠ a false) (κ κ' : Bool → Fin 2) (hκ : ∀ s, κ s ≠ κ' s) (s : Bool) : Module.finrank C (filSpace (coordFil a κ) s) = 1

theorem **exists_isDualDatum** (a : Bool → w.valuationSubring) (h : a true ≠ a false) : IsDualDatum a (resPair a (1 + 1)) (coordFil a fun s => if s then 0 else 1) (coordFil a fun s => if s then 1 else 0) ∧ ∀ s, Module.finrank C (filSpace (coordFil a fun s => if s then 0 else 1) s) = 1

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.QuotientNormSquareData

theorem **rowLat_fg** (t : T) : (D.rowLat t).FG

theorem **glued_fg** [Finite T] : D.glued.FG

theorem **RowData.slopeBound_iff** [StarRing L] {T : Type} [Finite T] {κ : T → Type} [∀ t, Fintype (κ t)] [∀ t, DecidableEq (κ t)] {τ ε C : Type} [Fintype ε] [DecidableEq ε] {m : ℕ} (D : RowData L T κ τ ε C m) (h0 : D.glued ≠ ⊥) (hh : ℝ) : D.SlopeBound hh ↔ Formalism.HN.muMaxDef (Pi.basisFun L ε) D.XE D.glued ≤ hh − D.logKap

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.RamifiedPlacesSplitEigen

theorem **lieMult_iff** {k : Type} [Field k] (act : OAct O G) (t : Sp k ⟶ S) (χ : O →+* k) (n : ℕ) : LieMult act t χ n ↔ ∃ v : Fin n → (epsAt t ⟶ G), IsEigBasis act t χ v

theorem **IsDetSection.smul** {Y : Model Xo c β} {τ : E →+* ℂ} {f : Rule Y} (hf : IsDetSection Y τ f) (a : c.F) : IsDetSection Y τ (fun R _ φ x v w => φ a * f R φ x v w)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ResidueInjectivityAt

theorem **ResInj.of_char** (h : ResInj) {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A) (hR : R.Char F) : ResInjAt R

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ResidueInjectivityChar

theorem **ResInj.of_char** (h : ResInj) (R : RealDat D A Ks) (F : Face D A) (hR : R.Char F) : ResInjAt R

theorem **ResInjAt.ext** {R : RealDat D A Ks} (h : ResInjAt R) (i : Fin D.R.m) {U U' : MMap (R.MW i (sg0 D A))} (hU : U ∈ R.Wall i (sg0 D A)) (hU' : U' ∈ R.Wall i (sg0 D A)) (e : ∀ v : ∀ ρ, lowT (R.MW i (sg0 D A)) ρ, U (fun ρ => v ρ) = U' (fun ρ => v ρ)) : U = U'

theorem **ResInjAt.good** {R : RealDat D A Ks} (h : ResInjAt R) (i : Fin D.R.m) {U : MMap (R.MW i (sg0 D A))} (hU : U ∈ R.Wat i (sg0 D A) R.triv) (e : VanishesOnLowest (R.MW i (sg0 D A)) U) : U = 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ResidueLevelMultiplicity

theorem **finite_res** (PP : Pl E) : Finite (kP PP)

theorem **finite_boundedLevelClassGroupFamily** (Pg : Finset (Pl E)) (r : ℕ) : Finite (G0 Pg r)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ScaledCoisometryNorms

theorem **sumGram_entries** {K : Type} [Zero K] {N n : ℕ} (hN : N = n + 1) (G' : Matrix (Fin n) (Fin n) K) (g : K) : (∀ a b : Fin n, sumGram N G' g (iC hN a) (iC hN b) = G' a b) ∧ (∀ a : Fin n, sumGram N G' g (iC hN a) (iL hN) = 0 ∧ sumGram N G' g (iL hN) (iC hN a) = 0) ∧ sumGram N G' g (iL hN) (iL hN) = g

theorem **brSq_eq** {ι : Type} [Fintype ι] {N n : ℕ} (hN : N = n + 1) (lam : ι → DomWeight N) (sg : ι → DomWeight n) {T : ℝ} (hT : 0 < T) : brSq hN lam sg T = T ^ (-∑ τ, nBr (lam τ) (sg τ)) * orthSq hN lam sg

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ScaledIntegerPeriodic

theorem **mem_scaledInt_iff** (ϖ : kˣ) (a : ℕ) (m : ℤ) (Z : Matrix (Fin a) (Fin a) k) : Z ∈ scaledInt ϖ a m ↔ Z ∈ matLat (Fin a) (ϖ ^ m • stdLat k (Fin a))

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.SiegelReframe

theorem **riem_eq_J** {g : ℕ} (x y : Fl g) : riem x y = x ·ᵥ (Matrix.J (Fin g) ℝ *ᵥ y)

theorem **thetaExp_add** {g : ℕ} {τ : Matrix (Fin g) (Fin g) ℂ} (hτ : τ.IsSymm) (l l' : Fl g) (z : Fin g → ℂ) : thetaExp τ (l + l') z = thetaExp τ l (z + per τ l') + thetaExp τ l' z + ∑ i, (l (Sum.inl i) : ℂ) * (l' (Sum.inr i) : ℂ)

theorem **theta_factor** {E : Type} [NormedAddCommGroup E] [NormedSpace ℂ E] {g : ℕ} {B : Set E} {τ : E → Matrix (Fin g) (Fin g) ℂ} (hτ : IsPeriodMap B τ) : IsFactor B τ (stdLat g) (fun l P => Complex.exp (2 * π * Complex.I * thetaExp (τ P.1) l P.2)) ∧ InducesPolarisation B τ (stdLat g) (fun l P => Complex.exp (2 * π * Complex.I * thetaExp (τ P.1) l P.2))

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.SpecialFibreUnits

theorem **PropZSlope.propZ** (h : PropZSlope) : PropZ

theorem **linComb_g_one** {Y Z : Scheme.{0}} (D : LB Z) (h1 : ∀ a b, D.g a b = 1) {K : Type} [Fintype K] (φ : K → (Y ⟶ Z)) (ε : K → ℤ) (i j : K → D.ι) : (LB.LinComb D φ ε).g i j = 1

theorem **normal_stalk** (A : Type) [CommRing A] [IsDomain A] [IsIntegrallyClosed A] (z : PrimeSpectrum A) : IsIntegrallyClosed ((Spec.structureSheaf A).presheaf.stalk z)

theorem **hyp_spec** (O : Type) [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] : Hyp (𝟙 (Spec (.of O))) (trivLB _)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.SplitQuadraticMinSums

theorem **not_isBig_iff** (k : ℕ) (f : Lt1 Dw) : ¬ IsBig k f ↔ IsSmall f ∨ IsSpread k f

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.SupremumLattice

theorem **supIn_full_iff** : ((supIn C Vs hV).FG ∧ Submodule.span K (supIn C Vs hV : Set R) = C) ↔ Submodule.IsLattice K (latOf C Vs hV)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TameInvariantLevelAction

theorem **chiDetG0_apply** {E : Type} [Field E] [NumberField E] {K : Type} [Field K] (Pg : Finset (Formalism.L2.DIV.Pl E)) (r : ℕ) (χ : ∀ w : ɪPg, (Formalism.L2.DIV.kP w.1)ˣ →* Kˣ) (g : Formalism.L2.DIV.G0 Pg r) : chiDetG0 Pg r χ g = ∏ w : ɪPg, χ w (Matrix.GeneralLinearGroup.det (g w))

theorem **isLift_one** (V : HSp D) {r : ℕ} (hr : V.rank = r) (β : GL (Fin r) (EAf D)) (Pg : Finset (Formalism.L2.DIV.Pl D.host.E)) : IsLift V hr β Pg 1 1

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TensorNormBound

theorem **IsTensorMap.unique** {lam : ι → DomWeight n} {lam' : ι → DomWeight n'} {A : ∀ τ, nablaSq ℂ n (lam τ) →ₗ[ℂ] nablaSq ℂ n' (lam' τ)} {T T' : VV ℂ lam →ₗ[ℂ] VV ℂ lam'} (h : IsTensorMap A T) (h' : IsTensorMap A T') : T = T'

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TensorPlaceNorm

theorem **fockF_eq_fockS** {σ : Type} (f : MvPolynomial σ ℂ) : Formalism.REAL.fockF f f = fockS f f

theorem **fockS_self_nonneg** {σ : Type} (f : MvPolynomial σ ℂ) : ∃ r : ℝ, 0 ≤ r ∧ fockS f f = r

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TopPlaceLogTheta

theorem **HostTie.host_heq** (h : HostTie N s Pr) : HEq N.H D.host.toHostAx

theorem **HostTie.rd_eq** (h : HostTie N s Pr) : N.rd = D.host.rdE

theorem **HostTie.cond_eq** (h : HostTie N s Pr) : N.cond = D.host.condN

theorem **HostTie.pr_cond** (h : HostTie N s Pr) : Pr.cond = D.host.condN

theorem **StepBound_sq.mono** {K : Type} [Field K] [NumberField K] [StarRing K] {Q : KerChain K} {i : ℕ} {v : InfinitePlace K} {t t' : ℝ} (h : StepBound_sq Q i v t) (ht : t ≤ t') (hpos : ∀ φ, 0 ≤ (famIpAt (Q.F i) (Q.hF i) v φ φ).re) : StepBound_sq Q i v t'

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TopRankPowers

theorem **add_sub_D** : X.D + (X.T - X.D) = X.T

theorem **sub_add** {a : ℕ} (ha : a ≤ X.a₀) : X.T - a + a = X.T

theorem **LocalHyp.of_meanHyp** {β κ : (F →+* ℂ) → ℝ} (h : X.MeanHyp β κ) : LocalHyp X β κ

theorem **Lifts.of_hyp** (h : Lifts) (hX : X.Hyp) {s : Fin X.D → Fin X.N → E} (hs : IsChosen X E s) : (∀ (τ : E →+* ℂ) (a : ℕ) (ha : a ≤ (X.baseChange E).a₀), ExtensionBound τ hs.good (X.T - a) ∧ LiftBound (X

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ToricSymbols

@[simp] theorem **coe_conjChar** {c : Ideal (OF E)} (g : ℂ ≃+* ℂ) (ψ : PicO c →* ℂˣ) (q : PicO c) : ((conjChar g ψ q : ℂˣ) : ℂ) = g (ψ q)

theorem **IsTwistedSum.unique** {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {c : Ideal (OF E)} {χ : HChar E} {ψ : PicO c →* ℂˣ} {T T' : ℂ} (h : IsTwistedSum φ s κ c χ ψ T) (h' : IsTwistedSum φ s κ c χ ψ T') : T = T'

theorem **isTwistedSum_iff_isToricSum** {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {c : Ideal (OF E)} {χ ξ : HChar E} {ψ : PicO c →* ℂˣ} (h : ∀ y, finPart ξ y = ψ (QuotientGroup.mk y)) (T : ℂ) : IsTwistedSum φ s κ c χ ψ T ↔ IsToricSum φ s κ c (χ * ξ) T

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TraceDualDifferent

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.mem_traceDual_iff** {M : FractionalIdeal (𝒪 F)⁰ F} (hM : M ≠ 0) (y : F) : y ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.traceDual M ↔ y ∈ FractionalIdeal.dual ℤ ℚ M

theorem **mem_traceDual_iff_different** {M : FractionalIdeal (𝒪 F)⁰ F} (hM : M ≠ 0) (y : F) : y ∈ traceDual M ↔ y ∈ ((differentIdeal ℤ (𝒪 F) : FractionalIdeal (𝒪 F)⁰ F))⁻¹ * M⁻¹

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.covol_pos** {M : FractionalIdeal (𝒪 F)⁰ F} (hM : M ≠ 0) : 0 < _root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.covol M

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.covol_eq_covolume** [IsTotallyReal F] (M : (FractionalIdeal (𝒪 F)⁰ F)ˣ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.covol (M : FractionalIdeal (𝒪 F)⁰ F) = ZLattice.covolume (mixedEmbedding.idealLattice F M)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.TraceDualDifferent.card_realEmb** [IsTotallyReal F] : Fintype.card (F →+* ℝ) = Module.finrank ℚ F

theorem **norm_add_ne_zero** {z : (F →+* ℝ) → ℂ} (hz : ∀ τ, (z τ).im ≠ 0) (a : F) : ∏ τ : F →+* ℝ, (z τ + ((τ a : ℝ) : ℂ)) ≠ 0

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TraceSubstitution

theorem **substTr_mem** {r r' n : ℕ} (ν : Fin n → ℕ) (x : Matrix (Fin r') (Fin r) ℂ) {f : MvPolynomial (Fin r × Fin n) ℂ} (hf : f ∈ dualWeyl ℂ r n ν) : substTr n x f ∈ dualWeyl ℂ r' n ν

theorem **substTr_mul** {r r' r'' n : ℕ} (x' : Matrix (Fin r'') (Fin r') ℂ) (x : Matrix (Fin r') (Fin r) ℂ) : substTr n (x' * x) = (substTr n x').comp (substTr n x)

theorem **substTr_vec** {r r' n : ℕ} (x : Matrix (Fin r') (Fin r) ℂ) (v : Fin r → ℂ) (s : Fin n) : substTr n x (∑ a, C (v a) * X (a, s)) = ∑ b, C ((x *ᵥ v) b) * X (b, s)

theorem **fockS_re_pos** {σ : Type} {f : MvPolynomial σ ℂ} (hf : f ≠ 0) : 0 < (fockS f f).re

theorem **invNorm_hw** {r n : ℕ} (hn : n ≤ r) (ν : Fin n → ℕ) (hν : Antitone ν) : invNorm hn ν (hw ℂ hn ν, hwMem ℂ r n hn ν hν) = 1

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TwistChoiceExample

theorem **W1_retuple** (X : CxData A) (cr : Crd A fc) (N : IndefNum Bs) (μ : HeightOneSpectrum (𝒪 K) → Bs.Tup) (Tw : Finset (HeightOneSpectrum (𝒪 K))) (w : HeightOneSpectrum (𝒪 K)) : (retuple X μ Tw).W1 cr N w ↔ X.W1 cr N w

theorem **W1_iff_fiveSet_or_fiveSet_conjPl** (X : CxData A) (cr : Crd A fc) (N : IndefNum Bs) (w : HeightOneSpectrum (𝒪 K)) : X.W1 cr N w ↔ FiveSet X cr N w ∨ FiveSet X cr N (conjPl w)

theorem **refInputs_of_links** (X : CxData A) (cr : Crd A fc) (wf : (lv (fc X.zs) false).WF) (hM : X.MLat false) (hK : X.KForm false) (hE : X.EWnorm false) (hp : X.H4pos) (h5 : X.H5 cr false) (hr : X.RATth cr false) : RefInputs X cr where

theorem **stdLat_isLattice** : Submodule.IsLattice K (stdLat K) where

theorem **lin_one_inj** : Function.Injective (lin (fun _ : Fin 1 => (1 : ℂ)))

theorem **exampleTwistChoice_w0C** (j : Unit) : (cx K v0).w0C false j = lin (fun _ : Fin 1 => (1 : ℂ))

theorem **tC_eq** (j : Unit) : (cx K v0).tC (crd K v0) false j false = lin (fun _ : Fin 1 => (1 : ℂ))

theorem **refInputs** : RefInputs (cx K v0) (crd K v0) where

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TwistedLevelValues

theorem **isGalConj_refl** {E : Type} [Field E] [NumberField E] (w : ℝ) (χ : Formalism.L2.Host.HChar E) : IsGalConj (RingEquiv.refl ℂ) w χ χ

theorem **isEmbOverK_base** : IsEmbOverK D D.what₂.embLL (RingEquiv.refl ℂ)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.TypedPartSplitPrimes

theorem **v_embX_le_one** (τ : E →+* ℂ) (a : 𝒪 E) : X.v (embX io X τ a) ≤ 1

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.UnitaryConjugation

theorem **mt_gram** (V : HSp D) (g : GA D V) : ((mt V g).map (cA D))ᵀ ∗ incM V.gram ∗ mt V g = incM V.gram

theorem **incM_smul** {m n : Type} (c : D.host.E) (J : Matrix m n D.host.E) : incM (c • J) = incA D c • incM J

theorem **continuous_conj** (D : D7) : Continuous (cA D)

theorem **continuous_conjU** (V : HSp D) {J : Matrix (Fin V.rank) (Fin V.rank) D.host.E} (c : D.host.E) (h : J = c • V.gram) : Continuous (conjU V c h)

theorem **blk_mul** {R : Type} [CommRing R] {p q d : ℕ} (e : Fin p ⊕ Fin q ≃ Fin d) (A A' : Matrix (Fin p) (Fin p) R) (B B' : Matrix (Fin q) (Fin q) R) : blk e A B ∗ blk e A' B' = blk e (A ∗ A') (B ∗ B')

theorem **blk_one** {R : Type} [CommRing R] {p q d : ℕ} (e : Fin p ⊕ Fin q ≃ Fin d) : blk e (1 : Matrix (Fin p) (Fin p) R) 1 = 1

theorem **blk_map** {R R' : Type} [Zero R] [Zero R'] {p q d : ℕ} (e : Fin p ⊕ Fin q ≃ Fin d) (A : Matrix (Fin p) (Fin p) R) (B : Matrix (Fin q) (Fin q) R) {f : R → R'} (h : f 0 = 0) : (blk e A B).map f = blk e (A.map f) (B.map f)

theorem **blk_transpose** {R : Type} [Zero R] {p q d : ℕ} (e : Fin p ⊕ Fin q ≃ Fin d) (A : Matrix (Fin p) (Fin p) R) (B : Matrix (Fin q) (Fin q) R) : (blk e A B)ᵀ = blk e Aᵀ Bᵀ

theorem **nonempty_num** {μ0 : T} (hL : L.Is μ0) (hi : i + 1 ≤ D.R.m) (hj : j + 1 ≤ D.R.m) : Nonempty (Num G i j)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.VpBounds

theorem **ToricDenominatorLevel.exact** (h : ToricDenominatorLevel) : ToricDenominator

theorem **expIn_le_two** {𝔭 : Ideal (OF E)} (h : IsBaseCond 𝔭) (v : HeightOneSpectrum (OF E)) : expIn 𝔭 v ≤ 2

theorem **expIn_le_maxConductorExponentOverPrime** {𝔭 : Ideal (OF E)} (h : IsBaseCond 𝔭) {l : ℕ} {v : HeightOneSpectrum (OF E)} (hv : (l : OF E) ∈ v.asIdeal) : expIn 𝔭 v ≤ a0 𝔭 l

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.WeightedNormSquareRows

theorem **mem_of_mem_divs** {A : Type} [CommRing A] [IsDedekindDomain A] {a : A} {P : HeightOneSpectrum A} (h : P ∈ divs a) : a ∈ P.asIdeal

theorem **evOk** (hX : X.Ok) (hPr : Pr.Ok) (P : LevFam X.E X.K R.r Pr.vGs) (u : Pr.Word) : (evGSD W Ch sel (mu0 X.F X.E X.K Sg) P u).Ok

theorem **card_eq_f** (Nd : NumDatum) (X : Fields) (pl : X.Tau ≃ RPl Nd.R) : X.f = Nd.R.f

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.WeilGramSignature

theorem **overK_base** : OverK (D

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.WeilTwistColumns

theorem **TwistFactorAt.norm** (hT : TwistFactorAt R i U dict ι e F ik φ th x₀ dδ U₂ dict₂ xi) : TwistFactorNormAt R i U dict ι e F ik φ th x₀ dδ U₂ dict₂ xi

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ZeroSectionBase

theorem **IsProjSurface.finiteCoh** (h : IsProjSurface k T) {E : T.Modules} {r : ℕ} (hE : IsVB E r) : FiniteCoh k E ∧ FiniteCoh k (VB.dual E) ∧ FiniteCoh k (structureModule T)

theorem **degCh2_congr** {E E' : T.Modules} (φ : E ≅ E') (r : ℕ) : degCh2 k E r = degCh2 k E' r

theorem **IsProjSurface.degCh2_rank_one** (h : IsProjSurface k T) {E : T.Modules} (hE : IsVB E 1) : degCh2 k E 1 = (topSelfIntersection k (det E 1) 2 : ℚ) / 2

theorem **isProjSurface_plane** (k : Type) [Field k] : letI : Algebra k Γ(PSpace k 2, ⊤)

theorem **zeroSec_comp** : zeroSec A ≫ A.hom = 𝟙 S

theorem **isVB_hodge** (hA : IsAbelianScheme A) (g : ℕ) [SmoothOfRelativeDimension g A.hom] : IsVB (hodge A) g

theorem **isPrincipalPol_of_polAV** (P : Out.PolAV S) : IsPrincipalPol P.A (lam (cls P.L P.isLB) (fst P.A P.A) (snd P.A P.A))

theorem **recordVanishes_of** (h : Ch2HodgeVanishes) {D : D7} {X : DivX D} (hX : RecordIsGeometric D X) : RecordVanishes D X

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ZetaPhiBranchOperator

theorem **exists_section** {K V W : Type} [Field K] [AddCommGroup V] [Module K V] [AddCommGroup W] [Module K W] (f : V →ₗ[K] W) : ∃ g : W →ₗ[K] V, ∀ v, f (g (f v)) = f v

theorem **pair_branchT** (lam : ι → DomWeight (n + 1)) (sg : ι → DomWeight n) (x : VV K lam) (y : VV K sg) : pairBK sg (branchT K lam sg x) y = pairB lam x.1 (branch K lam sg y)

theorem **mem_extLat** {K : Type} [Field K] [NumberField K] {n : ℕ} (M : Submodule (𝒪 K) (Fin n → K)) (t : K) (x : Fin (n + 1) → K) : x ∈ extLat M t ↔ (fun a : Fin n => x a.castSucc) ∈ M ∧ ∃ c : 𝒪 K, c • t = x (Fin.last n)

Development.AbelianVarietyTheory.BlockTheory.DivisorBase.ZetaValueIdeals

theorem **ZetaDenominatorsReal.cm** (h : ZetaDenominatorsReal) : ZetaDenominators

theorem **IsClassZeta.eqOn** {C : ClassGroup (𝒪 F)} {Z Z' : ℂ → ℂ} (h : IsClassZeta C Z) (h' : IsClassZeta C Z') : Set.EqOn Z Z' {1}ᶜ

theorem **IsZetaValues.unique** {k : ℕ} (hk : 1 ≤ k) {z z' : ClassGroup (𝒪 F) → ℚ} (h : IsZetaValues k z) (h' : IsZetaValues k z') : z = z'

theorem **mMin_mem** {l : ℕ} (hl : 1 ≤ l) (n : ℕ) : ∃ a : ℕ, 2 ≤ a ∧ a.Coprime l ∧ mMin l n = padicValNat l (a ^ n − 1)

theorem **LocalBound.mMin_eq** {k : ℕ} {z : ClassGroup (𝒪 F) → ℚ} {l : ℕ} (h : LocalBound k z l) : mMin l (finrank ℚ F ∗ k) = Formalism.PairingDivisionLocalBalance.mE l (finrank ℚ F ∗ k)

theorem **pir_rat** : IsPrincipalIdealRing (𝒪 ℚ)

theorem **absNorm_span_rat** (n : ℕ) : Ideal.absNorm (Ideal.span {(n : 𝒪 ℚ)}) = n

theorem **exists_span_rat** (I : Ideal (𝒪 ℚ)) : ∃ n : ℕ, I = Ideal.span {(n : 𝒪 ℚ)}

theorem **isClassZeta_rat** (C : ClassGroup (𝒪 ℚ)) : IsClassZeta C riemannZeta

Development.AbelianVarietyTheory.BlockTheory.EvidentLattices.LatticeLetterSources

theorem **srcAt_spec** (o : LamObj c nΨ n') (p : Σ τ : E →+* ℂ, Fin (blockSize H nΨ n' τ)) : (srcAt o p).chr = p.1 ∧ (srcAt o p).copy = (p.2 : ℕ)

theorem **srcAt_bijective** (o : LamObj c nΨ n') : Function.Bijective (srcAt o)

theorem **exists_isEvident** (o : LamObj c nΨ n') {f : ℕ} (h₁ : o.xΨ.P.g = f) (h₂ : o.x'.P.g = f) (h₃ : o.xB.P.g = f * (nΨ + n')) (U : Matrix (Fin o.xB.P.g) (Fin o.xB.P.g) o.K₁) : ∃ L : LamLetters o, L.m = blockSize H nΨ n' ∧ L.U = U ∧ L.IsEvident

theorem **frameOf_rfl** {r : ℕ} (Xx : DataX E r) (x : Frame Xx.Ψ r) : frameOf Xx rfl x = x

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.AdelicNormWeight

theorem **adConj_adConj** (x : AdF E ⊗[maximalRealSubfield E] E) : adConj E (adConj E x) = x

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.AdelicQuotientMeasure

theorem **cls_eq_iff** (g g' : H) : cls Γ g = cls Γ g' ↔ ∃ γ ∈ Γ, g = γ * g'

theorem **measurable_cls** : Measurable (cls Γ)

theorem **quotMeasure_apply** (μ : Measure H) (D : Set H) (hD : IsFundamentalDomain Γ D μ) {S : Set (LQuot Γ)} (hS : MeasurableSet S) : quotMeasure Γ μ D hD S = μ (cls Γ ⁻¹' S ∩ D)

theorem **quotMeasure_univ** (μ : Measure H) (D : Set H) (hD : IsFundamentalDomain Γ D μ) : quotMeasure Γ μ D hD Set.univ = μ D

theorem **toAf_cj** (x : E) : toAf E (cj E x) = cjAf E (toAf E x)

theorem **emb_cj** (τ : E →+* ℂ) (x : E) : τ (cj E x) = starRingEnd ℂ (τ x)

theorem **vol_ne_zero** (μ : Measure (UAd G)) [μ.IsHaarMeasure] (D : Set (UAd G)) (hD : IsFundamentalDomain (ratPts G) D μ) : μ D ≠ 0

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.AdmissibleHarmonicLattice

theorem **mem_stdMod** {x : Fin n → E} : x ∈ stdMod E n ↔ ∀ j, ∃ a : 𝒪 E, (a : E) = x j

theorem **single_mem_stdMod** (j : Fin n) : (Pi.single j 1 : Fin n → E) ∈ stdMod E n

theorem **std_selfDual** : (Lat.std E n).SelfDual

theorem **std_stdAt** (s : ℕ) : (Lat.std E n).StdAt s

theorem **std_admissible** {H : Host E} (l : Level H) : (Lat.std E n).Admissible l.M l.M

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.AgreementOfOrders

theorem **OrdersAgree.ne_top_iff** {h : GenFrEx Xo c} {Z : Comp Xo c} {ord : WTZ E → AC Z → ℤ} {q : QPt Xo c} (hq : OrdersAgree h Z ord q) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) : ordQ Xo c q f ≠ ⊤ ↔ evC h Z f ≠ 0

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.AlgebraicPointWeights

theorem **unitIndep_of** (h₁ : WeightOverFields Xo c) (h₂ : IntegralDiagUnit Xo c) : UnitIndep Xo c

theorem **intFrNe_of** (h₁ : AlgebraicPointOver Xo c) (h₂ : IntegralFrameAfterExt Xo c) : IntFrNe Xo c (fun _ _ y => y.IsAlg)

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.ArchimedeanCocycleCharacter

theorem **push_one** (c : Coeff l) (n : ℕ) : push (Dw

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.BaseChangeTensorProduct

theorem **bcBil_apply** (σ' : L →+* ℂ) (f : C →ₛₗ[σ] W) (t : L) (x : C) : bcBil σ' f t x = σ' t • f x

theorem **bcAdd_tmul** (hσ : ∀ a, σ' (algebraMap K L a) = σ a) (f : C →ₛₗ[σ] W) (t : L) (x : C) : bcAdd hσ f (t ⊗ₜ x) = σ' t • f x

theorem **rdGen_tmul** {σ' : KT T j N →+* ℂ} (hσ : IsExt σ') (w : WeightFamilies.Wt D j) (t : KT T j N) (F : Cf T j w) : rdGen hσ w (t ⊗ₜ F) = σ' t • rd T j w F

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.BlockProductLattices

theorem **mem_blockLat** (Λ₁ : GenLat E n₁) (Λ₂ : GenLat E n₂) (v : Fin (n₁ + n₂) → E) : v ∈ (blockLat Λ₁ Λ₂).La ↔ (fun j => v (Fin.castAdd n₂ j)) ∈ Λ₁.La ∧ (fun j => v (Fin.natAdd n₁ j)) ∈ Λ₂.La

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.CoefMemFlagSec

theorem **zAct_rmul** (X : Data E r) (cu : FJ.CuspFD X ρ) (t : Fin (nZ X ρ) → ℝ) (x b : Frame X.Ψ r) : zAct X cu t (rmul x b) = rmul (zAct X cu t x) b

theorem **isFlagSec_coefFn** (X : Data E r) (la : Wt X.Ψ r) (cu : FJ.CuspFD X ρ) (hd : ∀ t, ∀ x ∈ X.dom, zAct X cu t x ∈ X.dom) (ph : Fin (nZ X ρ) → ℝ) {P : Frame X.Ψ r → ℂ} (hP : IsFlagSec X la P) : IsFlagSec X la (coefFn (zAct X cu) ph P)

theorem **coef_mem_flagSec** {K : Type} [Field K] (σ : K →+* ℂ) (X : Data E r) (la : Wt X.Ψ r) (cu : FJ.CuspFD X ρ) (hd : ∀ t, ∀ x ∈ X.dom, zAct X cu t x ∈ X.dom) (ph : Fin (nZ X ρ) → ℝ) (P : OrbInt (zAct X cu) X.dom) (hP : IsFlagSec X la P) : (show FnVia σ (Frame X.Ψ r) from coefL (zAct X cu) X.dom ph P) ∈ flagSec σ X la

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.CoefficientSpaceWeights

theorem **part_anti** {Dw : Datum H} {a : (E →+* ℂ) × ℕ → ℕ} (ha : IsPartFam Dw a) (σ : E →+* ℂ) : Antitone (part Dw a σ)

theorem **coefAct_coe** (g : GL (Fin n) E) : coefAct Dw c n ha d (g : Matrix (Fin n) (Fin n) E) = coefRep Dw c n ha d g

theorem **natW_anti** {N : ℕ} (lam : DomWeight N) : Antitone lam.natW

theorem **ten_extremal_mem** {K ι : Type} [Field K] [Fintype ι] {N : ℕ} (lam : ι → DomWeight N) : ten (fun τ => extremal K N (lam τ)) ∈ VV K lam

theorem **LFTie.weil** (h : LFTie) : LFTieWeil

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.ComplexResidueRing

theorem **isComb_iff** (h : c.F ≤ c'.F) {m : ℕ} (b : Fin m → c'.F) (f : Fin m → Rul Xo c) (f' : Rul Xo c') : IsComb h b f f' ↔ f' = ∑ i, b i • ruleUp h (f i)

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.ConjugateFieldNorm

theorem **norm_embedding_hostFormConstant_eq** (τ : E →+* ℂ) : ‖τ Q.xi0‖ = Q.xiAbs

theorem **base_embedding_good** : GoodEmb Q.c Q.sg0

theorem **nrm_le_one_iff** {j : R.J} {la : HeightOneSpectrum (𝒪 Q.c.F)} (y : Q.Pt j la) (f : Q.Rules j) : Q.nrm y f ≤ 1 ↔ 0 ≤ Q.vAt_vw y f

theorem **xiAbsModels_of_pins** (h : Q.Pins) : Q.XiAbsModels

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.CuspFormLaws

theorem **autRt_mk** (V : HSpace E r) (h g : GA F V) : autRt F V h (autMk F V g) = autMk F V (g * h)

theorem **onQ_mk** (V : HSpace E r) (φ : GA F V → ℂ) (h : ∀ γ ∈ ratG F V, ∀ g, φ (γ * g) = φ g) (g : GA F V) : onQ F V φ (autMk F V g) = φ g

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.CuspMultiplicationClosed

theorem **mem_cusp** (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (hn : Even n) (f : Rul Xo c) : f ∈ Cusp U a n hn ↔ f ∈ sec Xo c a n ∧ ∀ σ : c.F →+* ℂ, GoodEmb c σ → IsHolCusp (U σ).X (laU (U σ) a n) (rdU (U σ) f)

theorem **cusp_le_sec** (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (hn : Even n) : Cusp U a n hn ≤ sec Xo c a n

theorem **laU_add** {σ : c.F →+* ℂ} (V : Unf Xo c σ r) (a a' : (E →+* ℂ) × ℕ → ℕ) {n n' : ℕ} (hn : Even n) (hn' : Even n') (k : V.Comp) : laU V (a + a') (n + n') k = laU V a n k + laU V a' n' k

theorem **rdU_mul** {σ : c.F →+* ℂ} (V : Unf Xo c σ r) (f g : Rul Xo c) (k : V.Comp) : rdU V (f * g) k = rdU V f k * rdU V g k

theorem **cuspMulClosed** (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) : CuspMulClosed U

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.CuspSectionNorms

theorem **ptNorm_add_le** {X : ι → Data E r} (la : FamWt X) (F G : Fn X) (p : FramePt X) : ptNorm la (F + G) p ≤ ptNorm la F p + ptNorm la G p

theorem **ptNorm_smul** {X : ι → Data E r} (la : FamWt X) (a : ℂ) (F : Fn X) (p : FramePt X) : ptNorm la (a • F) p = ‖a‖ * ptNorm la F p

theorem **isCuspSec_iff_of_finite** [Finite ι] {X : ι → Data E r} (la : FamWt X) (F : Fn X) : IsCuspSec la F ↔ ∀ c, F c ∈ (X c).SS (la c)

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.DeltaDatumCombinedCoefficient

theorem **conj_le_comap** : a.conj ≤ a.named.comap (H.bar : 𝒪 E →+* 𝒪 E)

theorem **KA_le** : KA l ≤ KA a.l₂

theorem **triv_pId** (l : Level H) : (triv l).pId = ⊤

theorem **phiDelta_transform** (δ : DeltaChar a F) (v : Vol D a) {n : ℕ} (k : D.La →ₗ[𝒪 E] D.La) (d : (𝒪 E ⧸ a.conj)ˣ) (hd : IsDetMod D a.conj k d) (u : Fin n → D.La) : phiDelta δ v (fun b => Submodule.Quotient.mk (u b)) = δ.loc (a.barQuot d) * phiDelta δ v (fun b => Submodule.Quotient.mk (k (u b)))

theorem **eq_inclusion** (a : IsAuxiliarySplitSquarefreeLevel l) (jK : KA l →+* KA a.l₂) (hj : ∀ x, ((jK x : KA a.l₂) : ℂ) = ((x : KA l) : ℂ)) : jK = (IntermediateField.inclusion a.KA_le).toRingHom

theorem **exists_bundleIso** {S S' : Scheme.{0}} {t' : SAB H S'} {t : SAB H S} {f : S' ⟶ S} (m : SABMap t' t f) : ∃ w, IsBundleIso m w

theorem **nonempty_cover** (h : ToroidalLevelCompatibility (H

theorem **Cover.triv_nabOver** (Xo : Moduli D l) (c : Coeff l) (nab : NabV Xo c) : (Cover.triv Xo c).NabOver @nab @nab

theorem **phiDelta_triv** (F : Type) [Field F] (v : Vol D (IsAuxiliarySplitSquarefreeLevel.triv l)) {n : ℕ} (x : Fin n → ConjCl D (IsAuxiliarySplitSquarefreeLevel.triv l).l₂) : phiDelta (DeltaChar.triv l F) v x = 1

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.EtaleProductPrimes

theorem **norm_thVal** (X : PT E) (s : (Fin 1 → Vsp X.dim) → ℂ) (a : Vsp X.dim) : ‖thVal X s a‖ = thRatio X s a

theorem **ProdForm.prodBound** {X : PT E} {s : (Fin 1 → Vsp X.dim) → ℂ} {r1 : ℕ} {Nc : ℝ} {i : ℕ} {y : ℝ} (h : ProdForm X s r1 Nc i y) : ProdBound X s r1 Nc i y

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.FlagSectionAction

theorem **isVal_values** (j : ℕ) (la : HeightOneSpectrum (𝒪 T.Q.c.F)) : ∀ v ∈ values T j la, IsVal la v

theorem **goodEmb_good** (σ : T.Q.c.F →+* ℂ) : GoodEmb T.Q.c (good T σ)

theorem **charLat_isLattice** (j : ℕ) (cu : Cusps T j) : Submodule.IsLattice D.host.E (charLat T j cu)

theorem **isFlagSec_rd** (h : LawsM T) {j : ℕ} (hj : InR D j) (w : WeightFamilies.Wt D j) (f : forms T j w) (k) : IsFlagSec ((T.U j T.σ₀).X k) (laU (T.U j T.σ₀) (wtA w) w.n k) (rdU (T.U j T.σ₀) f.1 k)

theorem **orbInt** (h : LawsCore T) {j : ℕ} (hj : InR D j) (w : WeightFamilies.Wt D j) : FJ.RdOrbInt T.σ₀ (T.X₀ j) (WeightFamilies.rho D j) ((rdU (T.U j T.σ₀)).comp (forms T j w).subtype)

theorem **petGood_eq** {j : ℕ} (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ) (hm : JMemL2 (T.V j σ) σ (rdC (T.U j) σ (wtA w) w.n) (T.dict j σ) (T.mu j σ)) {f : T.R j} (hf : f ∈ forms T j w) : petGood T j w σ f = l2C (T.U j) σ (T.V j σ) (T.dict j σ) (T.mu j σ) (xiR D) (wtA w) w.n hm f

theorem **isNormOn_supNorm** (h : LawsM T) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ) : IsNormOn (forms T j w) σ (supNorm T j w σ)

theorem **isNormOn_petNorm** (h : LawsM T) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ) : IsNormOn (forms T j w) σ (petNorm T j w σ)

theorem **petNorm_le_supNorm** (h : LawsM T) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ) : ∀ f ∈ forms T j w, petNorm T j w σ f ≤ supNorm T j w σ f

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.GeneralizedScalarsNumberField

theorem **pointNumberField_of_framed** (h : FramedPointNumberField Xo c) : PointNumberField Xo c

theorem **genScal_existsUnique** {S : Type} [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (h : y'.x = y.x) : ∃! a : S, GenScal a y y'

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.GeneratorValues

lemma **xK_eq** (D : DegPre R) (χ : D.X) : D.xK χ = jK R ≫ (D.c χ).left

theorem **exists_isPsiCoord** [IsLocalRing R] (D : DegPre R) (χ : D.X) : ∃ x, IsPsiCoord D χ x

lemma **res_res** {X : Scheme.{0}} {U V W : X.Opens} (h : U ≤ V) (h' : V ≤ W) (s : Γ(X, W)ˣ) : res h (res h' s) = res (h.trans h') s

lemma **pull_val** {X : Scheme.{0}} {S T : CommRingCat.{0}} (φ : S ⟶ T) (c : Spec S ⟶ X) (U : X.Opens) (g : Γ(X, U)ˣ) (h : ⊤ ≤ c ⁻¹ᵁ U) (h' : ⊤ ≤ (Spec.map φ ≫ c) ⁻¹ᵁ U) : Units.map (Scheme.ΓSpecIso T).hom.hom.toMonoidHom (res h' (pullU (Spec.map φ ≫ c) U g)) = Units.map φ.hom.toMonoidHom (Units.map (Scheme.ΓSpecIso S).hom.hom.toMonoidHom (res h (pullU c U g)))

theorem **IsPsiCoord.unique** {D : DegPre R} {χ : D.X} {x x' : (FractionRing R)ˣ} (hx : IsPsiCoord D χ x) (hx' : IsPsiCoord D χ x') : ∃ u : Rˣ, x = Units.map (algebraMap R (FractionRing R)).toMonoidHom u * x'

theorem **psiCoord_spec** [IsLocalRing R] (D : DegPre R) (χ : D.X) : IsPsiCoord D χ (psiCoord D χ)

theorem **psiVal_eq** {D : DegPre R} {χ : D.X} {x : (FractionRing R)ˣ} (hx : IsPsiCoord D χ x) : psiVal v D χ = v.toFun x

theorem **periodForm_symm** (D : DegPre R) (a b : D.X) : periodForm v D a b = periodForm v D b a

lemma **valuation_ne_zero** (x : (FractionRing R)ˣ) : (IsDiscreteValuationRing.maximalIdeal R).valuation (FractionRing R) x.val ≠ 0

theorem **ord_vw_uniformiser** {ϖ : R} (h0 : ϖ ≠ 0) (h : maximalIdeal R = Ideal.span {ϖ}) : (ord_vw R).toFun (toUnitK ϖ h0) = 1

theorem **ord_vw_pos** {r : R} (h0 : r ≠ 0) (h : r ∈ maximalIdeal R) : 0 < (ord_vw R).toFun (toUnitK r h0)

theorem **ramIdx_pos** {p : ℕ} (hp : IsOverPrime R p) : 0 < ramIdx hp

theorem **ord_vp_p** {p : ℕ} (hp : IsOverPrime R p) : (ord_vp hp).toFun (toUnitK (p : R) hp.ne_zero) = 1

theorem **periodForm_vp_eq** {p : ℕ} (hp : IsOverPrime R p) (D : DegPre R) (a b : D.X) : periodForm_vp hp D a b = (periodForm_vw D a b : ℚ) / (ramIdx hp : ℚ)

theorem **IsCentred.norm_unit** (hφ : IsCentred φ) (u : Rˣ) : ‖φ (algebraMap R (FractionRing R) u)‖ = 1

theorem **IsCentred.log_norm_p_neg** (hφ : IsCentred φ) {p : ℕ} (hp : IsOverPrime R p) : 0 < ‖(p : F)‖ ∧ Real.log ‖(p : F)‖ < 0

theorem **normVal_vp_p** (hφ : IsCentred φ) {p : ℕ} (hp : IsOverPrime R p) : (normVal_vp hφ hp).toFun (toUnitK (p : R) hp.ne_zero) = 1

theorem **isOverPrime_padicInt** (p : ℕ) [hp : Fact p.Prime] : IsOverPrime ℤ_[p] p where

theorem **isCentred_padic** (p : ℕ) [Fact p.Prime] : IsCentred (R

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.HalfDepthValuations

theorem **bar_eq_barE** (Q : Par D) (a : 𝒪 D.host.E) : Q.H.bar a = barE D a

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.HalfGeometricCovers

theorem **r_le_nV** (D : D7) : D.R.r ≤ D.R.nV

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.HalfLevelCoefficients

theorem **isHalf_levelG** (D : D7) : IsHalf D (MbOf D [D.host.g 0, D.host.g 1]) where

theorem **three_le_g** (D : D7) (k : Fin 2) : 3 ≤ D.host.g k

theorem **datAt_rank_pos** (D : D7) (j : ℕ) : 1 ≤ (datAt D j).p + (datAt D j).q

theorem **halfOfPrimes_iff** (D : D7) : HalfOfPrimes D ↔ ∀ pb : Fin 2 → HeightOneSpectrum (𝒪 D.host.K), PrimesOver D pb → ∃ l : Level (hostOfRecord D), l.M = D.host.g 0 * D.host.g 1 ∧ l.A = 1 ∧ l.Mb = halfOf D pb

theorem **isParAt_parWith** (D : D7) (hF : OFull D) (hz : OZeta D) (Mb : Ideal (𝒪 D.host.E)) (h : IsHalf D Mb) (Xo : ∀ j, Moduli (datAt D j) (levelAt D Mb h)) : IsParAt D Mb (parWith D hF hz Mb h Xo)

theorem **isParAt_parAt** (D : D7) (hF : OFull D) (hz : OZeta D) (hN : N1ab) (Mb : Ideal (𝒪 D.host.E)) (h : IsHalf D Mb) : IsParAt D Mb (parAt D hF hz hN Mb h)

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.HarmonicChartValues

theorem **monVal_natCast** (B : Germ.FB S d 0) (h : Germ.PS S 2 d) (w0 : 0) (m : ℕ) (f : Germ.PS S 1 d) {κ : Type} [Fintype κ] (u : κ → Fin d → S) (α : κ →₀ ℕ) : monVal B h w0 (m : ℤ) f u α = B.nabVal S h w0 m f u (MvPolynomial.monomial α 1)

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.JetChartLaws

theorem **jetVec_X** {σ : Type} (c : σ → S) (i : σ) : jetVec c (Ideal.Quotient.mk _ (MvPolynomial.X i)) = c i • (DualNumber.eps : DualNumber S)

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.LatticeEmbeddingHalf

theorem **latBase_le_latG_one** : latBase L 𝔄 ≤ latG F fr L 𝔄 1

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.LevelIndexFourierCharacter

theorem **bar_bar** (H : Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

theorem **bar_mem_of_mem_conjHalf** (l : Level H) {a : 𝒪 E} (ha : a ∈ conjHalf l) : H.bar a ∈ l.Mb

theorem **mul_mem_span_level** (l : Level H) {a b : 𝒪 E} (ha : a ∈ l.Mb) (hb : b ∈ conjHalf l) : ∃ z : 𝒪 E, a * b = (l.M : 𝒪 E) * z

theorem **exists_isSep** (l : Level H) : ∃ c, IsSep l c

theorem **sep_spec** (l : Level H) : IsSep l (sep l)

theorem **finite_quot** (I : Ideal (𝒪 E)) (hI : (l.M : 𝒪 E) ∈ I) : Finite (D.La ⧸ (I • (⊤ : Submodule (𝒪 E) D.La)))

theorem **level_mem_mul** : (l.M : 𝒪 E) ∈ l.Mb * conjHalf l

theorem **ef_smul_eq_zero** {a b : 𝒪 E} (ha : a ∈ l.Mb) (hb : b ∈ conjHalf l) (u v : D.La) : ((D.EF u ((a * b) • v) : ℤ) : ZMod l.M) = 0

theorem **ef_left_zero** {c : 𝒪 E} (hc : c ∈ conjHalf l) {u : D.La} (hu : u ∈ conjHalf l • (⊤ : Submodule (𝒪 E) D.La)) (v : D.La) : ((D.EF u (c • v) : ℤ) : ZMod l.M) = 0

theorem **ef_right_zero** {c : 𝒪 E} (hc : c ∈ conjHalf l) (u : D.La) {v : D.La} (hv : v ∈ l.Mb • (⊤ : Submodule (𝒪 E) D.La)) : ((D.EF u (c • v) : ℤ) : ZMod l.M) = 0

theorem **pairOne_mk** {c : 𝒪 E} (hc : IsSep l c) (u v : D.La) : pairOne D l (Submodule.Quotient.mk u) (Submodule.Quotient.mk v) = ((D.EF u (c • v) : ℤ) : ZMod l.M)

theorem **zeta_level_mem** (c : Coeff l) : zeta l.M ∈ c.F

theorem **zetaM_prim** (c : Coeff l) : IsPrimitiveRoot (zetaM c) l.M

theorem **levPtCl_mk** (Xo : Moduli D l) {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) (u : D.La) : levPtCl Xo x0 (Submodule.Quotient.mk u) = levPt Xo x0 u

theorem **indCoeff_eq** (c : Coeff l) {n : ℕ} (x₀ : Fin n → ConjCl D l) (t : Fin n → NamedCl D l) : indCoeff c x₀ t = (Nat.card (Fin n → NamedCl D l) : c.F)⁻¹ * (zetaM c ^ (pairing D l x₀ t).val)⁻¹

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.LevelPairAction

theorem **LevAct.congr** {gM gM' gA gA' : Dw.La →ₗ[𝒪 E] Dw.La} (hM : CongM l gM' gM) (hA : CongN l.A gA' gA) {S : Type} [CommRing S] [Algebra c.F S] {y y' : FrPt Xo c S} (h : LevAct gM gA y y') : LevAct gM' gA' y y'

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.ModelFamilies

theorem **perFr_add** (X : Data E r) (x : Frame X.Ψ r) (v w : Fin r → E) (τ : E →+* ℂ) (i : ℕ) : perFr X x (v + w) τ i = perFr X x v τ i + perFr X x w τ i

theorem **pmOf_apply** (M : ModelsOf F r) (k : Fin F.n) (x : Frame (M.X k).Ψ r) (a : (F.Di k).La) (j : Fin g) : pmOf M k x a j = perFr (M.X k) x (M.φ k a : Fin r → E) (relab c σ (e j).1) (e j).2

theorem **xiSign_of_kappa** (M : ModelsOf F r) (h : ∀ k, (M.X k).κ0 = 2 * (M.X k).ξ0) : XiSign M

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.NormFactorThetaValue

theorem **nsmul_single_mem_inter** {n M N : ℕ} {Λ : Lat E n} (h : Λ.Admissible M N) (j : Fin n) : N • (Pi.single j 1 : Fin n → E) ∈ Λ.inter

theorem **nsmul_single_mem** {n M N : ℕ} {Λ : Lat E n} (h : Λ.Admissible M N) (j : Fin n) : N • (Pi.single j 1 : Fin n → E) ∈ Λ.L

theorem **one_inZ** {n N : ℕ} {Λ : Lat E n} (h : Λ.Admissible l.M N) {T : Over Xo.X} : InZ Xo h (fun _ => (1 : T ⟶ Xo.univ.t.G))

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.PrimitiveGermFrames

theorem **sum_primε** : ∑ k, primε k = -1

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.PrincipalCongruenceGamma

theorem **GamEqAt.gamAt** {T : Core D} {j : ℕ} {σ : T.Q.c.F →+* ℂ} (h : GamEqAt T j σ) : GamAt T j σ

theorem **GamEq.gam** {T : Core D} (h : GamEq T) : Gam T

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.ReflexNormIndexIdeal

theorem **latIdeal_ne_bot** (Λ : Lat E n) : latIdeal Λ ≠ ⊥

theorem **latIdeal_std** : latIdeal (Lat.std E n) = ⊤

theorem **isBe_top** : IsBe Dw c ⊤ 1

theorem **beIdeal_spec** (hK : ReflexLe Dw c) (𝔞 : Ideal (𝒪 E)) (h𝔞 : 𝔞 ≠ ⊥) : IsBe Dw c 𝔞 (beIdeal hK 𝔞 h𝔞)

theorem **be_spec** (hK : ReflexLe Dw c) (Λ : Lat E n) : IsBeLat Dw c Λ (be hK Λ)

theorem **be_std** (hK : ReflexLe Dw c) : be hK (Lat.std E n) = 1

theorem **be_congr** (hK : ReflexLe Dw c) {n' : ℕ} {Λ : Lat E n} {Λ' : Lat E n'} (h : latIdeal Λ = latIdeal Λ') : be hK Λ = be hK Λ'

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.ShortRuleRepresentationChange

theorem **IsShortRule.unique** (h : CoverTot Δ) {n N : ℕ} {Λ : Lat E n} {β : c.F} {x₀ : Fin n → ConjCl Dw l} {a : (E →+* ℂ) × ℕ → ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hd : IsSeriesData Δ Λ N β a P) {f g : Rul Xo c} (hf : IsShortRule Δ Λ N β x₀ a P f) (hg : IsShortRule Δ Λ N β x₀ a P g) : f = g

private theorem **eqAtPrime_refl** {n : ℕ} (p : ℕ) (hp : p.Prime) (L : Submodule (𝒪 E) (Fin n → E)) : EqAtPrime p L L

theorem **std_mem_reps** (n : ℕ) (s : ℕ+) : Lat.std E n ∈ reps Δ n s

theorem **exists_inPsi** (H : Host E) : ∃ τ : E →+* ℂ, InPsi H τ

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.StandardFourierShortSpan

theorem **CuspLeSpan.genNeZero** (h : CuspLeSpan Δ hK nV cusp) : GenNeZero Δ hK nV cusp

theorem **CuspLeSpan.cuspEqSpan** (h : CuspLeSpan Δ hK nV cusp) (h' : ShortInCusp Δ hK nV cusp) : CuspEqSpan Δ hK nV cusp

theorem **SpanCanonical.neZero** (h : SpanCanonical Xo c) : NeZeroCanonical Xo c

theorem **exists_spanRange** (hr : 3 ≤ Dw.p + Dw.q) (hA : l.A = 1) : ∃ a n, SpanRange Dw l a n ∧ OnePos Dw a

theorem **StdFourier.unit** (h : StdFourier Xo c nab) : StdFourierUnit Xo c nab

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.SymmetricAdjointUnits

theorem **comp_lamφ** {T T' : Over X₀} (h : T' ⟶ T) (p : Fin 2 → (T ⟶ G)) (I : ULift.{0} (Finset (Fin 2))) : h.left ≫ lamφ G p I = lamφ G (fun k => h ≫ p k) I

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.TranslationFrames

theorem **comp_mulA** {T : Over S} (x : T ⟶ A) (N : ℕ) : x ≫ mulA A N = x ^ N

theorem **mulIsoT_le** (L : LB A.left) (N : ℕ) (T : Over S) (j : Fin 2 → L.ι) : LB.wchart L ![(snd T A).left, (snd T A ^ N).left] j ≤ (snd T A).left ⁻¹ᵁ LB.wchart L ![𝟙 A.left, (mulA A N).left] j

theorem **one_left** (T : Over S) : (1 : T ⟶ A).left = T.hom ≫ eSec A

theorem **originFrame_le** (L : LB A.left) (T : Over S) (j : Fin 1 → L.ι) : LB.wchart L (fun _ : Fin 1 => (1 : T ⟶ A).left) j ≤ LB.wchart L (fun _ : Fin 1 => T.hom ≫ eSec A) j

Development.AbelianVarietyTheory.BlockTheory.ExistenceTori.UnitaryHaarVolume

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHaarVolume.isClosed_Upq** (r p0 : ℕ) : IsClosed (_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHaarVolume.Upq r p0 : Set (GL (Fin r) ℂ))

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHaarVolume.baseFr_mem** : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHaarVolume.baseFr X ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryHaarVolume.OrthFr X

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.ActionCoordinates

theorem **herm_self** {P Q : ℕ} (x : Wsp P Q) : herm x x = (qf x : ℂ)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.AdelicUnitaryHaarMeasure

theorem **isUnimodularLC_of_commGroup** (G : Type) [CommGroup G] [TopologicalSpace G] [IsTopologicalGroup G] [T2Space G] [LocallyCompactSpace G] [SecondCountableTopology G] [MeasurableSpace G] [BorelSpace G] : IsUnimodularLC G

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.CharacterValuation

theorem **muAt_eq_muAtT** (h : MuqTie Z.Ch sel mu cst) (i : ℕ) (hi : i < Z.R.m) : muAt Z.Ch sel mu cst i = muAtT Z.Ch sel mu cst i

theorem **UnitAt.ne_zero** {K : Type} [Field K] [NumberField K] {w : HeightOneSpectrum (𝒪 K)} {x : K} (h : UnitAt w x) : x ≠ 0

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.ClassMeanPeriods

theorem **exists_curveIdeal_delta** (D : D7) : ∃ (cc : Ideal (𝒪 D.host.K)) (δ : ℂ), IsCurveIdeal D cc ∧ IsDiscriminantTwelfthRootOfIdealLattice D cc δ

theorem **classMean_iff** : ClassMean ↔ ∀ D : D7, ClassMeanAt D

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.ClassesDoubleCosets

theorem **CosetMass.unique** {G : Type} [Group G] {Γ K : Subgroup G} {m m' : ℝ} (h : CosetMass Γ K m) (h' : CosetMass Γ K m') : m = m'

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.ColumnStabiliser

theorem **mem_colStab** (D : D7) (Lv : LevelData D) (u : D.what₂.Word) : letI

theorem **map_colStab** (D : D7) (Lv : LevelData D) (u : D.what₂.Word) : letI

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.CosetGroupMembership

theorem **mem_cosetGrp** {G : Type} [Group G] (Γ K : Subgroup G) (y x : G) : x ∈ cosetGrp Γ K y ↔ x ∈ Γ ∧ y⁻¹ * x * y ∈ K

theorem **stabHat_eq** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] (hc : CMData F E) (S : HSpace E) (N : Submodule (𝒪 E) S.V) : stabHat hc S N = (levelAfL hc.sig S.gram (crdLat S N)).subgroupOf (UAf hc S)

theorem **CosetSum.pos** {G : Type} [Group G] {Γ K : Subgroup G} {m : ℝ} (h : CosetSum Γ K m) : 0 < m

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.EquidistributedConjugates

theorem **conjEmb_apply** (M : IntermediateField ℚ ℂ) (ρ : M →ₐ[ℚ] ℂ) (y : M) : conjEmb M ρ y = (starRingEnd ℂ) (ρ y)

theorem **exists_representatives** (D : D7) : ∃ r : ClassGroup (𝒪 D.host.K) → (Ideal (𝒪 D.host.K))⁰, ∀ C, ClassGroup.mk0 (r C) = C

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.FewBadPrimes

theorem **ArithData.ccK_eq** {C : FC} {s : Scale} {H : Host R} {q0 : ℕ} {bs : R.A → R.A → R.A} (h : ArithData C s H q0 bs) : H.ccK = ⊤

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.IndexExponentDegree

theorem **locScal_one_one** {K : Type} [Field K] (R : Ranks) (i q : ℕ) (w : Kˣ) : locScal (R.N i) (R.N (i + 1)) q w 1 1 = (q : K) ^ ((R.t + 1) / 2)

theorem **locScal_zero_zero** {K : Type} [Field K] (a b q : ℕ) (w : Kˣ) : locScal a b q w 0 0 = 1

theorem **even_N** (R : Ranks) (j : ℕ) : Even (R.N j)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.InfiniteCoordinateUnfold

theorem **IsStabEmb.inv_mul_fix** {P' Q' P Q : ℕ} {x₀ : Fin P ⊕ Fin Q → ℂ} {j : UpqT P' Q' →* UpqT P Q} (hj : IsStabEmb x₀ j) (g' : UpqT P' Q') (g : UpqT P Q) : UpqT.mat (j g' * g)⁻¹ *ᵥ x₀ = UpqT.mat g⁻¹ *ᵥ x₀

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.InterlacingEquivariance

theorem **brOne_rename** (K : Type) [Field K] {N n : ℕ} (hN : N = n + 1) (lam : DomWeight N) (sg : DomWeight n) (G : VV K fun _ : Unit => lam) : brOne K hN lam sg (rename Prod.snd (G : Pol K Unit N)) = rename Prod.snd ((Br.brStd hN (fun _ : Unit => lam) (fun _ : Unit => sg) G : VV K fun _ : Unit => sg) : Pol K Unit n)

theorem **interlace_iff** {N n : ℕ} (hN : N = n + 1) (lam : DomWeight N) (sg : DomWeight n) : Br.Interlace hN lam sg ↔ Formalism.MainNotions.PrimeDegreeVariant.Interlace hN.symm lam sg

theorem **brAt_eq** {K : Type} [Field K] {ι : Type} [Fintype ι] {N n : ℕ} (hN : N = n + 1) (lam : ι → DomWeight N) (sg : ι → DomWeight n) (t : ι → Kˣ) : Br.brAt hN lam sg (fun τ => (t τ : K)) = ((∏ τ, t τ ^ (-((∑ i, (lam τ).w i) - ∑ i, (sg τ).w i)) : Kˣ) : K) • Br.brStd hN lam sg

theorem **members_sameShift** {X : Fields} {Word : Type} {R : Ranks} (W : Wts X Word R) (hW : W.Ok) (u : Word) (τ : X.Tau) : Br.SameShift (W.lamTrue u R.m τ) (W.sgTrue τ)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.InvariantRadonFibre

@[simp] theorem **fibreIntegral_mk** (ν : Measure H) [ν.IsMulLeftInvariant] (f : G → ℝ≥0∞) (g : G) : fibreIntegral H ν f (g : G ⧸ H) = ∫⁻ h : H, f (g * h) ∂ν

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.InvariantRadonFibre.continuous_invCoset** : Continuous (_root_.GlobalTheory.IndefiniteRowSideFullHecke.InvariantRadonFibre.invCoset H)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.InvariantRadonFibre.IsInvariantRadon.regular** [LocallyCompactSpace G] [SecondCountableTopology G] [IsClosed (H : Set G)] [MeasurableSpace (G ⧸ H)] [BorelSpace (G ⧸ H)] {μ : Measure (G ⧸ H)} (h : _root_.GlobalTheory.IndefiniteRowSideFullHecke.InvariantRadonFibre.IsInvariantRadon H μ) : μ.Regular

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.JacobiThetaHalfCharacteristic

theorem **thetaChar_eq_tsum** (a b : Fin 2) (z τ : ℂ) : thetaChar a b z τ = ∑' n : ℤ, cexp ((Real.pi : ℂ) * I * ((n : ℂ) + ((a : ℕ) : ℂ) / 2) ^ 2 * τ + 2 * (Real.pi : ℂ) * I * ((n : ℂ) + ((a : ℕ) : ℂ) / 2) * (z + ((b : ℕ) : ℂ) / 2))

theorem **thetaChar_zero_zero** (z τ : ℂ) : thetaChar 0 0 z τ = jacobiTheta₂ z τ

theorem **jacobiTheta₂_half** (a b : Fin 2) (τ : ℂ) : jacobiTheta₂ ((((a : ℕ) : ℂ) * τ + ((b : ℕ) : ℂ)) / 2) τ = cexp (-((Real.pi : ℂ) * I * ((a : ℕ) : ℂ) ^ 2 * τ / 4) - (Real.pi : ℂ) * I * ((a : ℕ) : ℂ) * ((b : ℕ) : ℂ) / 2) * thetaChar a b 0 τ

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.JetSumTable

theorem **adj_single** (x : HomE E a b) (α : Fin a) (β : Fin b) : adj x (Pi.single β 1) α = star (x (Pi.single α 1) β)

theorem **InIntClosure.single** {H I : Type} {Sup : Set H} (tab : H → I → ℂ_[p]) (M : Set I) {x : H} (hx : x ∈ Sup) : InIntClosure Sup tab M (tab x)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.LatticeNormalFormUnipotent

theorem **isUnip_one** (X : Submodule F Y) (a : Fin c → Y) : IsUnip X a 1

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.LocalDensityEuler

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.LocalAdditiveCharacterDual.psiV_zero** (ψ : AddChar (_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.AF F) ℂ) (v : HeightOneSpectrum (𝒪 F)) : _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.psiV F ψ v 0 = 1

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.LocalAdditiveCharacterDual.psiV_add** (ψ : AddChar (_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.AF F) ℂ) (v : HeightOneSpectrum (𝒪 F)) (x y : _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) : _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.psiV F ψ v (x + y) = _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.psiV F ψ v x * _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.psiV F ψ v y

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.LocalAdditiveCharacterDual.star_eq_sigma** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [StarRing E] (σ : E →ₐ[F] E) (h : _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.IsCMExt F E σ) (hu : _root_.Formalism.HermitianLatticesTheory.IsCMUnram E) (x : E) : star x = σ x

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.LocalDualIndex

theorem **psiV_zero** (ψ : AddChar (AF F) ℂ) (v : HeightOneSpectrum (𝒪 F)) : psiV F ψ v 0 = 1

theorem **psiV_add** (ψ : AddChar (AF F) ℂ) (v : HeightOneSpectrum (𝒪 F)) (s t : Fv F v) : psiV F ψ v (s + t) = psiV F ψ v s * psiV F ψ v t

theorem **isScalingInf_of_eq** {ψ₀ ψ : AddChar (AF F) ℂ} (h₀ : IsStdChar F ψ₀) {a : F} (h : ∀ x, ψ x = ψ₀ (algebraMap F (AF F) a * x)) : IsScalingInf F ψ a

theorem **IsStdChar.isScalingInf_one** {ψ₀ : AddChar (AF F) ℂ} (h₀ : IsStdChar F ψ₀) : IsScalingInf F ψ₀ 1

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.MassFormulaPositivity

theorem **posDef_of_hasSignature** {n : Type} [Fintype n] [DecidableEq n] {J : Matrix n n ℂ} {p : ℕ} (h : Formalism.UG.HasSignature J p 0) : J.PosDef

theorem **trace_pos** {E : Type} [Field E] [NumberField E] [StarRing E] (hst : ∀ (φ : E →+* ℂ) (a : E), φ (star a) = starRingEnd ℂ (φ a)) (W : HSpace E) (hs : W.SigOK) {p : ℕ} (hp : ∀ τ : E →+* ℂ, W.sig τ = (p, 0)) (x : W.V) (hx : x ≠ 0) : 0 < Algebra.trace ℚ E (W.hT x x)

theorem **mass_formula₂** (D : D7) : letI

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.MeanPeriodLogarithms

theorem **logOmVal_eq** (D : D7) : logOmVal D = Real.eulerMascheroniConstant + Real.log (2 * Real.pi / D.host.rdE) - logDerivativeAtOne D

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.NormalFormInclusion

theorem **nf_one** (Pr : Primes X) (j : ℕ) (P : HeightOneSpectrum (𝒪 X.E)) (hP : P ∉ Pr.vGs) : NF Pr j P (1 : AutY X R j)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.OnePlaceBranchStatement

theorem **interlace_iff** {N n : ℕ} (hN : N = n + 1) (lam : DomWeight N) (sg : DomWeight n) : Br.Interlace hN lam sg ↔ Formalism.MainNotions.PrimeDegreeVariant.Interlace hN.symm lam sg

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.PeriodLatticeAgreement

theorem **mem_latt** {ω₁ τ w : ℂ} : w ∈ latt ω₁ τ ↔ ∃ m n : ℤ, w = ω₁ * ((m : ℂ) + (n : ℂ) * τ)

theorem **mem_scale** {c w : ℂ} {Λ : AddSubgroup ℂ} : w ∈ scale c Λ ↔ ∃ v ∈ Λ, c * v = w

theorem **scale_latt** (c ω₁ τ : ℂ) : scale c (latt ω₁ τ) = latt (c * ω₁) τ

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.PeriodPairingUnitLattice

theorem **IntegralUpTo.of_dvd** {n n' : ℕ} (h : IntegralUpTo emb Pl Λ Λ' per n) (hn : n ∣ n') : IntegralUpTo emb Pl Λ Λ' per n'

theorem **card_stabilizer_dvd₂** (Tw : TwForms D) (u : D.what₂.Word) (μ : D.what₂.Tup) (c : (Tw.fm u μ).clsP) : Nat.card ((Tw.fm u μ).stabilizer₂ c) ∣ (D.chTw u μ).DGam

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.PerpendicularEmbedding

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.herm_self** {P Q : ℕ} (x : _root_.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.Sp P Q) :

root.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.herm x x =
(_root_.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.qf x : ℂ)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.herm_eq_hform** {P Q : ℕ} (x y :
root.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.Sp P Q) :
root.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.herm x y = _root_.Formalism.UG.hform (starRingEnd ℂ)
(_root_.Formalism.UG.Jpq P Q) y x

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.cM_pos** (M : ℕ) : 0 <
root.GlobalTheory.IndefiniteRowSideFullHecke.PerpendicularEmbedding.cM M

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.RayPrincipalIdeals

theorem **count_span_eq_zero** {m : Ideal (𝒪 K)} {a : 𝒪 K} (ha : a ≠ 0) (h : IsCoprime (Ideal.span {a}) m) (p : HeightOneSpectrum (𝒪 K))
(hp : p.asIdeal ∣ m) : FractionalIdeal.count K p (FractionalIdeal.spanSingleton (𝒪 K)⁰ (a : K)) = 0

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SandwichNormalForm

theorem **mem_lamYat** {j : ℕ} {P : HeightOneSpectrum (𝒪 X.E)} (hP : P ∉ Pr.vGs) (y : Sp X.E (R.N j + cOf R j)) : y ∈ lamYat (Pr

theorem **nf_one** (j : ℕ) {P : HeightOneSpectrum (𝒪 X.E)} (hP : P ∉ Pr.vGs) : NF (R

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.ScalarCharacterTwist

theorem **ordE_pow** (w : HeightOneSpectrum (𝒪 K)) (k : K) (n : ℕ) : ordE w (k ^ n) = n * ordE w k

theorem **hasVal_vp_iff** (w : HeightOneSpectrum (𝒪 K)) (k : K) (r : ℚ) : HasVal_vp σ w (σ k) r ↔ k ≠ 0 ∧ (ordE w k : ℚ) = r * (eW w : ℚ)

theorem **unitAtW_iff** (w : HeightOneSpectrum (𝒪 K)) (k : K) : UnitAtW σ w (σ k) ↔ w.valuation K k = 1

theorem **modOne_iff** (k : K) : ModOne σ (σ k) ↔ ∀ φ : K →+* ℂ, ‖φ k‖ = 1

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SelfDualLatticeActions

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.act_one** (y :
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.Wv E v M) :
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.act v σ J 1 y = y

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.act_mul** (g g' :
root.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.HA
(_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) σ J) (y :
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.Wv E v M) :
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.act v σ J (g * g') y =
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.act v σ J g
(_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.act v σ J g' y)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.sigR_mem** {z :
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.Ev E v} (hz : z ∈
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.OEv E v) :
root.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR
(_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) σ z ∈
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualLatticeActions.OEv E v

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SelfDualLatticeComplements

theorem **mem_perp** (W : HSpace E) (x y : W.V) : y ∈ perp W x ↔ W.hD y x = 0

theorem **ForallPerpOver.exists** (h : ForallPerpOver E) : ExistsPerpOver E

theorem **injective_of_isIsomEmb** {W' W : HSpace E} {ι : W'.V →ₗ[E] W.V} (h : HSpace.IsIsomEmb W' W ι) : Function.Injective ι

theorem **UnitVectorPerpAll.unitVectorPerp** {D : D7} (h : UnitVectorPerpAll D) : UnitVectorPerp D

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SelfDualLatticeIndex

theorem **factorT_pos** (D : D7) (u : D.what₂.Word) : 0 < (((D.mem u).A.T : ℝ) ^ D.R.k - 1) ^ D.R.f

theorem **stabHat_le** (D : D7) (N : Submodule (𝒪 D.host.E) (Fin D.R.k → D.host.E)) : letI

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SelfDualUnramifiedPlaces

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.mem_maxOrder** (v : HeightOneSpectrum (𝒪 F)) (z :
root.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v ⊗[F] E) : z ∈
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.maxOrder F E v ↔
(_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.intToEv F E v).IsIntegralElem z

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.one_lt_resCard** (F : Type) [Field F] [NumberField F] (v
: HeightOneSpectrum (𝒪 F)) : 1 < _root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.resCard F v

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.density_factor_pos** (F E : Type) [Field F] [NumberField
F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) (M : ℕ) (hM : 1 ≤ M) : 0 < 1 -
root.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.eta F E v ^ M *
(_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelfDualUnramifiedPlaces.resCard F v ^ M)⁻¹

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SignatureFrame

theorem **SignFrame.reindex_J** (F : SignFrame ε P Q) : Matrix.reindex F.e F.e (Formalism.UG.Jpq P Q) = Jm ε

theorem **SignFrame.unitsMap_val** (F : SignFrame ε P Q) (g : GL (Fin P ⊕ Fin Q) ℂ) : (F.unitsMap g : Matrix (Fin n) (Fin n) ℂ) =
Matrix.reindex F.e F.e (g : Matrix (Fin P ⊕ Fin Q) (Fin P ⊕ Fin Q) ℂ)

theorem **SignFrame.inUJ_reindex** (F : SignFrame ε P Q) (m : Matrix (Fin P ⊕ Fin Q) (Fin P ⊕ Fin Q) ℂ) : inUJ ε (Matrix.reindex F.e F.e m)
↔ Formalism.UG.cT (starRingEnd ℂ) m * Formalism.UG.Jpq P Q * m = Formalism.UG.Jpq P Q

theorem **SignFrame.umat_toDiag** (F : SignFrame ε P Q) (g : UpqT P Q) : umat (F.toDiag g) = Matrix.reindex F.e F.e (UpqT.mat g)

theorem **SignFrame.continuous_toDiag** (F : SignFrame ε P Q) : Continuous F.toDiag

theorem **SignFrame.measurable_toDiag** (F : SignFrame ε P Q) : Measurable F.toDiag

theorem **SignFrame.toDiag_mul** (F : SignFrame ε P Q) (g h : UpqT P Q) : F.toDiag (g * h) = F.toDiag g * F.toDiag h

theorem **SignFrame.bijective_toDiag** (F : SignFrame ε P Q) : Function.Bijective F.toDiag

theorem **cdMeasure_apply** (c : CDHaar) (F : SignFrame ε P Q) {A : Set (UJ ε)} (hA : MeasurableSet A) : cdMeasure c F A = c.cd P Q (F.toDiag ⁻¹' A)

theorem **SignFrame.continuous_symm** (F : SignFrame ε P Q) : Continuous F.mulEquiv.symm

theorem **cdMeasure_isHaar** (c : CDHaar) (F : SignFrame ε P Q) : (cdMeasure c F).IsHaarMeasure

theorem **dimV_eq_fdegHC** {n : ℕ} {lam : Fin n → ℝ} (h : StrictAnti lam) : dimV lam = fdegHC lam

theorem **exists_signFrame** (ε : Fin n → Bool) : Nonempty (SignFrame ε (Fintype.card {a // ε a = true}) (Fintype.card {a // ¬ ε a = true}))

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SourceLatticeInvariance

theorem **intAt_zero** (P : HeightOneSpectrum (𝒪 E)) : IntAt P (0 : E)

theorem **intAt_add** {P : HeightOneSpectrum (𝒪 E)} {a b : E} (ha : IntAt P a) (hb : IntAt P b) : IntAt P (a + b)

theorem **intAt_mul** {P : HeightOneSpectrum (𝒪 E)} {a b : E} (ha : IntAt P a) (hb : IntAt P b) : IntAt P (a * b)

theorem **intAt_neg** {P : HeightOneSpectrum (𝒪 E)} {a : E} (ha : IntAt P a) : IntAt P (-a)

theorem **intAt_algebraMap** (P : HeightOneSpectrum (𝒪 E)) (r : 𝒪 E) : IntAt P (algebraMap (𝒪 E) E r)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.SplittingProjection

theorem **mem_primeOf_prP** {σ : E →ₐ[F] E} (hφ : IsSplitting σ ∨ φ) (a : intV E v) : a ∈ primeOf (prP φ) ↔ Valued.v (prP φ a) < 1

theorem **primeOf_prP_ne_top** {σ : E →ₐ[F] E} (hφ : IsSplitting σ ∨ φ) : primeOf (prP φ) ≠ ⊤

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.StabiliserCardinality

theorem **mem_stab** {G : Type} [Group G] (Γ K : Subgroup G) (y γ : G) : γ ∈ stab Γ K y ↔ γ ∈ Γ ∧ y⁻¹ * γ * y ∈ K

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.StabilizerCompletions

theorem **actVec_one** (y : Fin M → Af F ⊗[F] E) : actVec σ J 1 y = y

theorem **actVec_mul** (g k : HA (Af F) σ J) (y : Fin M → Af F ⊗[F] E) : actVec σ J (g * k) y = actVec σ J g (actVec σ J k y)

theorem **mem_stabVec** (x : Fin M → E) (k : HA (Af F) σ J) : k ∈ stabVec σ J x ↔ actVec σ J k (ratVec F x) = ratVec F x

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.StabilizerLattices

theorem **actV_eq** (σ : E →ₐ[F] E) {M : ℕ} (J : Matrix (Fin M) (Fin M) E) (h : HA R σ J) (y : Fin M → R ⊗[F] E) : actV R σ J h y = ((HA.mat R σ J h : GL (Fin M) (R ⊗[F] E)) : Matrix (Fin M) (Fin M) (R ⊗[F] E)).mulVec y

theorem **actV_one** (σ : E →ₐ[F] E) {M : ℕ} (J : Matrix (Fin M) (Fin M) E) (y : Fin M → R ⊗[F] E) : actV R σ J 1 y = y

theorem **actV_mul** (σ : E →ₐ[F] E) {M : ℕ} (J : Matrix (Fin M) (Fin M) E) (g h : HA R σ J) (y : Fin M → R ⊗[F] E) : actV R σ J (g * h) y = actV R σ J g (actV R σ J h y)

theorem **mem_stdLat** (E : Type) [Field E] (M : ℕ) (y : Fin M → E) : y ∈ stdLat E M ↔ ∀ i, IsIntegral ℤ (y i)

theorem **isSelfDualLat_std** {F E : Type} [Field F] [Field E] [Algebra F E] (σ : E →ₐ[F] E) (M : ℕ) : IsSelfDualLat σ (1 : Matrix (Fin M) (Fin M) E) (stdLat E M)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.StandardDeterminantUnit

theorem **splitNonConj_vT** {X : Fields} {Pr : Primes X} (hPr : Pr.Ok) (u : Pr.Word) : SplitNonConj (Pr.vT u)

theorem **representatives_of_split** (h : RepresentativesSplit) : Representatives

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.StandardKernelAtTorus

theorem **adj_single** (x : HomE E a b) (β : Fin b) (α : Fin a) : adj x (Pi.single β 1) α = star (x (Pi.single α 1) β)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.TameGeneratorWindow

theorem **TameDistinct.injective** {D : D7} (h : TameDistinct D) : TameInjective D

theorem **SmallPrimeSharpAt.weak** {D : D7} {L : SizeData₂ D} {C8 : ℝ} (h : SmallPrimeSharpAt L C8) : SmallPrimeAt L

theorem **lJ_den_pos** {D : D7} (L : SizeData₂ D) : 0 < (D.R.k : ℝ) * ((D.R.m : ℝ) + (D.R.t : ℝ)) * Real.log (3 * hWidth L)

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.ThetaConstantWeber

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.disc_ne_zero** (τ : ℍ) : (_root_.GlobalTheory.periodPairOfTau τ).g₂ ^ 3 - 27 * (_root_.GlobalTheory.periodPairOfTau τ).g₃ ^ 2 ≠ 0

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.jLat_eq** (τ : ℍ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.jLat τ = 1728 * (_root_.GlobalTheory.periodPairOfTau τ).g₂ ^ 3 / ((_root_.GlobalTheory.periodPairOfTau τ).g₂ ^ 3 - 27 * (_root_.GlobalTheory.periodPairOfTau τ).g₃ ^ 2)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.halfPeriods_notMem** (τ : ℍ) : (1 / 2 : ℂ) ∉ (_root_.GlobalTheory.periodPairOfTau τ).lattice ∧ (τ : ℂ) / 2 ∉ (_root_.GlobalTheory.periodPairOfTau τ).lattice ∧ (1 + (τ : ℂ)) / 2 ∉ (_root_.GlobalTheory.periodPairOfTau τ).lattice

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.ThetaWeber.pow_twelve** {τ : ℍ} (h : _root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.ThetaWeber τ) : (_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.thetaConst 0 0 τ / ModularForm.eta τ) ^ 12 = _root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.weberF τ ^ 24 ∧ (_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.thetaConst 0 1 τ / ModularForm.eta τ) ^ 12 = _root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.weberF₁ τ ^ 24 ∧ (_root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.thetaConst 1 0 τ / ModularForm.eta τ) ^ 12 = _root_.GlobalTheory.IndefiniteRowSideFullHecke.ThetaConstantWeber.weberF₂ τ ^ 24

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.TrivialColumnStabiliser

theorem **mem_stabGroup** (D : D7) (Lv : LevelData D) (u : D.what₂.Word) (y γ : GL (Fin D.R.k) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : γ ∈ stabGroup D Lv u y ↔ γ ∈ GRat D ∧ y⁻¹ * γ * y ∈ colU D Lv u

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.UnfoldingOperatorConditions

theorem **condO_of_ok** {X : Formalism.L2.Col.Fields} {SB : Finset (HeightOneSpectrum (𝒪 X.E))} {cst : LocCst X SB} (h : cst.Ok) (P : SB) : CondO P.1 (cst.psi P) (cst.D P)

theorem **condO_of_addChar** {K : Type} [Field K] (P : HeightOneSpectrum (𝒪 E)) (ψ : AddChar E K) (h1 : ∀ x, IntAt P x → ψ x = 1) (h2 : ψ ((qOf P : E)⁻¹) ≠ 1) (D : ℕ) : CondO P ψ D

theorem **exists_condO** (P : HeightOneSpectrum (𝒪 E)) : ∃ psi : E → ℂ, CondO P psi 1

Development.AbelianVarietyTheory.BlockTheory.GlobalSignature.UnitaryLatticeOrthogonal

private lemma **involution_apply_eq_tensorProductMap_restrictScalars** (a : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryLatticeOrthogonal.Ev F E v) : _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) σ a = ((Algebra.TensorProduct.map (AlgHom.id (_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) (_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v)) σ).restrictScalars (v.adicCompletionIntegers F)) a

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryLatticeOrthogonal.sigR_mem** {a : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryLatticeOrthogonal.Ev F E v} (ha : a ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryLatticeOrthogonal.OEv F E v) : _root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (_root_.Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) σ a ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryLatticeOrthogonal.OEv F E v

theorem **hR_one** {M : ℕ} (z y : Fin M → Ev F E v) : hR (Fv F v) σ (1 : Matrix (Fin M) (Fin M) E) z y = ∑ i, sigR (Fv F v) σ (z i) * y i

Development.AbelianVarietyTheory.BlockTheory.GoodPlaces.RamifiedPlaceAnyLevel

theorem **GoodPlaceAnyLevel.base** (hG : GoodPlaceAnyLevel) : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) [StarRing (𝒪 D.host.E)], (∀ a : 𝒪 D.host.E, star a = T.Q.H.bar a) → ∀ (Tc : CDVR) (io : T.Q.c.F →+* Tc.Fr) (yI : IntFr (T.Q.Xo j) T.Q.c Tc io) (e : ℕ) (la : HeightOneSpectrum (𝒪 T.Q.c.F)), OverLa la Tc.V io e → yI.IsAlg → (∀ p ∈ D.host.condN.primeFactors, ((p : 𝒪 T.Q.c.F) ∉ la.asIdeal)) → ∀ (C : IntegralChartA3 (T.Q.Xo j) T.Q.c Tc io yI) (k : ℕ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN (T.Q.Dw j) n) Tc.Fr), P ∈ FFV (Dw

theorem **RamPlaceAnyLevel.base** (hG : RamPlaceAnyLevel) : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) [StarRing (𝒪 D.host.E)], (∀ a : 𝒪 D.host.E, star a = T.Q.H.bar a) → ∀ (Tc : CDVR) (io : T.Q.c.F →+* Tc.Fr) (yI : IntFr (T.Q.Xo j) T.Q.c Tc io) (e : ℕ) (la : HeightOneSpectrum (𝒪 T.Q.c.F)), OverLa la Tc.V io e → yI.IsAlg → ∀ p ∈ D.host.condN.primeFactors, ((p : 𝒪 T.Q.c.F) ∈ la.asIdeal) → ∀ (C : IntegralChartA3 (T.Q.Xo j) T.Q.c Tc io yI) (k : ℕ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN (T.Q.Dw j) n) Tc.Fr), P ∈ FFV (Dw

theorem **GoodPlaceAnyLevel.cover** (hG : GoodPlaceAnyLevel) (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) [StarRing (𝒪 D.host.E)] (hs : ∀ a : 𝒪 D.host.E, star a = T.Q.H.bar a) (l₂ : Level T.Q.H) (Xo₂ : Moduli (T.Q.Dw j) l₂) (c₂ : Coeff l₂) (hF : c₂.F = T.Q.c.F) (Tc : CDVR) (io : c₂.F →+* Tc.Fr) (yI : IntFr Xo₂ c₂ Tc io) (e : ℕ) (la : HeightOneSpectrum (𝒪 T.Q.c.F)) (ho : OverLa la Tc.V (io.comp (IntermediateField.equivOfEq hF.symm : T.Q.c.F →+* c₂.F)) e) (hy : yI.IsAlg) (hp : ∀ p ∈ D.host.condN.primeFactors, ((p : 𝒪 T.Q.c.F) ∉ la.asIdeal)) (C : IntegralChartA3 Xo₂ c₂ Tc io yI) (k : ℕ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN (T.Q.Dw j) n) Tc.Fr) (hP : P ∈ FFV (Dw

theorem **RamPlaceAnyLevel.cover** (hG : RamPlaceAnyLevel) (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) [StarRing (𝒪 D.host.E)] (hs : ∀ a : 𝒪 D.host.E, star a = T.Q.H.bar a) (l₂ : Level T.Q.H) (Xo₂ : Moduli (T.Q.Dw j) l₂) (c₂ : Coeff l₂) (hF : c₂.F = T.Q.c.F) (Tc : CDVR) (io : c₂.F →+* Tc.Fr) (yI : IntFr Xo₂ c₂ Tc io) (e : ℕ) (la : HeightOneSpectrum (𝒪 T.Q.c.F)) (ho : OverLa la Tc.V (io.comp (IntermediateField.equivOfEq hF.symm : T.Q.c.F →+* c₂.F)) e) (hy : yI.IsAlg) (p : ℕ) (hp : p ∈ D.host.condN.primeFactors) (hpl : (p : 𝒪 T.Q.c.F) ∈ la.asIdeal) (C : IntegralChartA3 Xo₂ c₂ Tc io yI) (k : ℕ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN (T.Q.Dw j) n) Tc.Fr) (hP : P ∈ FFV (Dw

Development.AbelianVarietyTheory.BlockTheory.GoodPlaces.ShortAtLevelPrime

theorem **Oriented.ne_zero** {T : Full D} {i : Fin 2} {π : 𝒪 D.host.E} (h : Oriented T i π) : π ≠ 0 ∧ T.Q.H.bar π ≠ 0

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.AdelicUnitaryGroups

theorem **cT_map** {m p : Type} (σ : A →+* A) (σ' : B →+* B) (φ : A → B) (h : ∀ x, σ' (φ x) = φ (σ x)) (M : Matrix m p A) : cT σ' (M.map φ) = (cT σ M).map φ

theorem **glue_mul** (j : B →ₙ+* A) (a a' : Matrix n n A) (b b' : Matrix n n B) : glue j a b * glue j a' b' = glue j (a * a') (b * b')

theorem **glue_one** (j : B →ₙ+* A) : glue j (1 : Matrix n n A) 1 = 1

theorem **emb_mul** (M : Matrix m n A) (P : Matrix n m A) (hP : P * M = 1) (a b : Matrix n n A) : (1 + M * (a - 1) * P) * (1 + M * (b - 1) * P) = 1 + M * (a * b - 1) * P

@[simp] theorem **tensNU_tmul** (j : R →ₗ[F] R') (hj : ∀ a b, j (a * b) = j a * j b) (a : R) (e : E) : tensNU j hj (a ⊗ₜ[F] e) = j a ⊗ₜ[F] e

theorem **placeOf_isReal** (hc : CMData F E) (τ : E →+* ℂ) : (placeOf F τ).IsReal

theorem **ext_algebraMap** (hc : CMData F E) (τ : E →+* ℂ) (a : F) : InfinitePlace.Completion.extensionEmbedding (placeOf F τ) (algebraMap F (placeOf F τ).Completion a) = τ (algebraMap F E a)

@[simp] theorem **locMap_tmul** (hc : CMData F E) (τ : E →+* ℂ) (a : (placeOf F τ).Completion) (x : E) : locMap hc τ (a ⊗ₜ[F] x) = InfinitePlace.Completion.extensionEmbedding (placeOf F τ) a * τ x

theorem **locMap_sig** (hc : CMData F E) (τ : E →+* ℂ) (y : (placeOf F τ).Completion ⊗[F] E) : starRingEnd ℂ (locMap hc τ y) = locMap hc τ (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ hc.sig y)

theorem **locMap_bijective** (hc : CMData F E) (τ : E →+* ℂ) : Function.Bijective (locMap hc τ)

theorem **act_one** (y : Fin V.rank → R ⊗[F] E) : act hc V (1 : HA R hc.sig V.gram) y = y

theorem **act_mul** (g h : HA R hc.sig V.gram) (y : Fin V.rank → R ⊗[F] E) : act hc V (g * h) y = act hc V g (act hc V h y)

theorem **act_sub** (g : HA R hc.sig V.gram) (y z : Fin V.rank → R ⊗[F] E) : act hc V g (y − z) = act hc V g y − act hc V g z

theorem **act_inv** (g : HA R hc.sig V.gram) (y : Fin V.rank → R ⊗[F] E) : act hc V g⁻¹ (act hc V g y) = y

theorem **hT_coord** (V : HSpace E) (x y : V.V) : V.hT x y = ∑ i, ∑ j, star (V.frame.repr x i) * V.gram i j * V.frame.repr y j

theorem **gram_det_isUnit** (V : HSpace E) : IsUnit V.gram.det

theorem **cT_gram** (V : HSpace E) : cT (hc.sig : E →+* E) V.gram = V.gram

theorem **embMat_isom** (h : HSpace.IsIsomEmb B A ι) : cT (hc.sig : E →+* E) (embMat B A ι) * A.gram * embMat B A ι = B.gram

theorem **gram_projMat** : cT (hc.sig : E →+* E) (embMat B A ι) * A.gram = B.gram * projMat hc B A ι

theorem **projMat_embMat** (h : HSpace.IsIsomEmb B A ι) : projMat hc B A ι * embMat B A ι = 1

theorem **cT_projMat** : cT (hc.sig : E →+* E) (projMat hc B A ι) * B.gram = A.gram * embMat B A ι

theorem **ratPts_eq** : ratPts hc V = HFinIn (AF F) hc.sig V.gram

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.ArcFramePointTruncation

theorem **fracMap_algebraMap** (hg : Function.Injective g) (r : A) : fracMap g hg (algebraMap A _ r) = algebraMap A' _ (g r)

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.BesselDatum

theorem **period_sym** (d : BesselDatum A B ι) (ψ μ 𝓕 ν 𝒩 s φ F) : (sym d).period ψ μ 𝓕 ν 𝒩 s φ F = d.period ψ μ 𝓕 ν 𝒩 s φ F

theorem **toPos_sym** {A B : HSp D} {ι : A.V →ₗ[D.host.E] B.V} (d : BesselDatum A B ι) (ε m b) : d.toPos ε m (d.ρ + 1) b = b

theorem **embedding_xi_zero_ne_zero** (D : D7) : D.host.xi0 ≠ 0

theorem **hT_swap** {S : HSp D} {x y : S.V} {c : D.host.E} (h : S.hT x y = c) : S.hT y x = star c

theorem **odd_codim** {A B : HSp D} {ι : A.V →ₗ[D.host.E] B.V} (d : BesselDatum A B ι) : Odd (B.rank − A.rank)

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.CMCurveHermitianForms

theorem **cmHost_bar_halfIntegerGenerator** (H : Host E) : H.bar H.w0 = 1 − H.w0

theorem **herm_conj** (D : Datum H) (x y : D.La) : (herm D y x : 𝒪 E) = H.bar (herm D x y : 𝒪 E)

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.CuspFormNormData

theorem **embE_int_mem** (D : D7) (x : 𝒪 D.host.E) : D.embE (x : D.host.E) ∈ Kfld D

theorem **nK_pos** (D : D7) : 0 < nK D

theorem **nK_eq** (D : D7) : nK D = Module.finrank ℤ (OO D)

theorem **card_emb** (D : D7) : Fintype.card (Emb D) = nK D

theorem **vK_nonneg** {D : D7} (l : Place D) (a : OO D) : 0 ≤ vK l (a : KK D)

theorem **Conj.refl** : c.Conj c

theorem **nonempty_of_wittIndex** (h : wittIndex D j = rho j) : Nonempty (CuspGeom D j)

theorem **vAt_zero** {G : ThetaWeight D j} (y : Φ.Pt j) : Φ.vAt y (0 : Φ.Cusp j G) = ⊤

theorem **mul_mem_M** {G G' : ThetaWeight D j} {f : Φ.Cusp j G} {g : Φ.Cusp j G'} (hf : f ∈ Φ.M G) (hg : g ∈ Φ.M G') : Φ.mul f g ∈ Φ.M (G.add G')

theorem **mem_deepK** (G : ThetaWeight D j) (c₁ : ℝ) (f : Φ.Cusp j G) : f ∈ Φ.deepK G c₁ ↔ ∀ (a : Φ.MaxCusp j), ∀ χ ∈ (Φ.geom a).killed c₁, Φ.fjK a G χ f = 0

theorem **mem_deepKRec** (h : Φ.PositiveExponents) (G : ThetaWeight D j) (T : ℕ) (c : ℝ) (f : Φ.Cusp j G) : f ∈ Φ.deepKRec G T c ↔ ∀ (a : Φ.MaxCusp j) (χ : expGroup (Φ.geom a).X), (χ : Herm D.host (rho j)) ∈ shallowExponents D j T c (Φ.geom a).X → Φ.fjK a G χ f = 0

theorem **deepK_eq_top** (h : rho j = 0) (G : ThetaWeight D j) (c₁ : ℝ) : Φ.deepK G c₁ = ⊤

theorem **vK_mul** {D : D7} (l : Place D) (a b : KK D) : vK l (a * b) = vK l a + vK l b

theorem **vK_add** {D : D7} (l : Place D) (a b : KK D) : min (vK l a) (vK l b) ≤ vK l (a + b)

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.DegreeIdentityCharacter

theorem **hostChar_apply** (H : Host E) (n : ℤ) : hostChar H (n : ZMod H.d) = (jacobiSym n H.d : ℂ)

theorem **hostChar_ne_one** (H : Host E) : hostChar H ≠ 1

theorem **lFunction_one_ne_zero** (H : Host E) : haveI : NeZero H.d

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.DimensionConstantBounds

theorem **Z_eq** {D : D7} (L : SizeData₂ D) : Z L = zBase L

theorem **PP_eq_PPv** {D : D7} (L : SizeData₂ D) (CP : ℝ) : PP L CP = PPv D L CP 12

theorem **PP_eq_of_eP_eq_twelve** {D : D7} {L : SizeData₂ D} {P : ParameterChoice' D L} (T : ParamsT D L P) (h : T.eP = 12) : PP L T.CP = T.PP

theorem **RR_eq** (D : D7) : RR D = RRv D 7

theorem **f_pos** (D : D7) : 0 < D.R.f

theorem **lrd_eq** (D : D7) : lrd D = logdE D / (2 * (D.R.f : ℝ))

theorem **q_pos** (D : D7) : 0 < q D

theorem **dimFlag_pos** (D : D7) : 0 < dimFlag D

theorem **Dp_eq** {D : D7} {L : SizeData₂ D} {P : ParameterChoice' D L} (T : ParamsT D L P) : T.Dp = dimFlag D

theorem **half_f** (D : D7) : (D.R.f : ℝ) / 2 = D.R.n

theorem **fieldSize_O** {D : D7} (X : Prov D) : fieldSize X.KO = Real.log (degO X) + Real.log (rdO X)

theorem **fieldSize_E** (D : D7) : fieldSize D.host.E = Real.log (Module.finrank ℚ D.host.E : ℝ) + lrd D

theorem **logdE_nonneg** (D : D7) : 0 ≤ logdE D

theorem **Sizes.halves** {D : D7} {L : SizeData₂ D} {C : ℝ} (h : Sizes L C) : logdE D ≤ 4 * (D.R.f : ℝ) * L.sc.L ∧ |L.logDerivativeAtOne| ≤ 2 * L.sc.L

theorem **dimBound_le_dimBoundMax** (D : D7) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : n0 D j ≤ dimBoundMax D

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.EigenLatticeFibres

theorem **GenFib.exists_isVec** {t : SAB H (Sp (𝒪 K))} (F : GenFib t) (c : Fin F.g → K) : ∃ x, F.IsVec c x

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.EigenLatticeFrames

theorem **isEigenBasisAt_frPt** {Xo : Moduli Dw l} {c : Coeff l} {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (τ : E →+* ℂ) : IsEigenBasisAt Xo c y.x τ (y.u τ)

theorem **exists_basis_of_integral** {Xo : Moduli Dw l} {c : Coeff l} {T : CDVR} {io : c.F →+* T.Fr} (y : letI

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.FieldTwistConditions

theorem **zeta_mul_pow** (n k : ℕ) (hk : k ≠ 0) : zeta (n * k) ^ k = zeta n

theorem **curveOver_of_mul** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {F : IntermediateField ℚ ℂ} {n k : ℕ} (hk : k ≠ 0) (h : GlobalTheory.CompRat.CurveOver H F (n * k)) : GlobalTheory.CompRat.CurveOver H F n

theorem **FieldCondBoth.run** (h : FieldCondBoth D) {p : ℕ} (hp : p = D.host.pde ∨ p = D.host.pde') : FieldCond D p

theorem **bar_bar** (x : AfE D) : bar D (bar D x) = x

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.FinitePartTypes

theorem **HasTypeAt.of_mem** {S : SPar} {φ : GA D V → ℂ} (h : φ ∈ finPartOfType θ V τ S) : HasTypeAt θ V τ S φ

theorem **HasTypeAt.eq_zero_of_not_good** {S : SPar} (hS : ∀ p q, ¬ GoodPar V.rank p q S) {φ : GA D V → ℂ} (h : HasTypeAt θ V τ S φ) : φ = 0

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.FockOrthogonalDecomposition

theorem **infMat_one** (θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ) (V : HSp D) : infMat D.sp.cm.sig V.gram θ (1 : GA D V) = 1

theorem **infMat_mul** (θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ) (V : HSp D) (g h : GA D V) : infMat D.sp.cm.sig V.gram θ (g * h) = infMat D.sp.cm.sig V.gram θ g * infMat D.sp.cm.sig V.gram θ h

theorem **KfinTied.some** {Wl : WeilTied D X T} (h : KfinTied Wl) : KfinWeilCompatibleExists Wl

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.FrameNormRestrictedDensity

theorem **RestrDensity.genSup_pos** {h : LawsCore T} {j : ℕ} {hG : GenFrEx (T.Q.Xo j) T.Q.c} {fn : FrameNorms T j} {yP : GenericFrames T j} {ka : WeightFamilies.Wt D j} (H : RestrDensity h j hG fn yP ka) (F : (recB h).Cusp j ka) (hF : F ∈ (recB h).Mlat j ka) (Z : Comp (T.Q.Xo j) T.Q.c) (hZ : evC hG Z F.1 ≠ 0) (σ : T.Q.c.F →+* ℂ) : 0 < genSup hG fn yP ka F.1 Z σ

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.FullTranscendenceDegree

theorem **ptNorm_nonneg** (σ : T.Q.c.F →+* ℂ) (u : WeightFamilies.Wt D j) (F : CuspAt h j u) (p : FPt (T.U j σ).X) : 0 ≤ ptNorm h j σ u F p

theorem **ptNorm_le_l2Norm** (σ : T.Q.c.F →+* ℂ) {wL : WeightFamilies.Wt D j} {N : ℕ} (w : Fin N → CuspAt h j wL) (p : FPt (T.U j σ).X) (i : Fin N) : ptNorm h j σ wL (w i) p ≤ l2Norm h j σ w p

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.IsotypicSignedParameters

theorem **signedParameter_one_one_isOfSig** : signedParameterU11.IsOfSig 1 1

theorem **signedParameter_one_one_dimV_eq_one** : signedParameterU11.dimV = 1

theorem **IsFdegPar.pos** {n : ℕ} {P : SignedPar n} {d : ℝ} (h : IsFdegPar P d) : 0 < d

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.L2SquaredNorms

theorem **IsL2SquareIntegralProblem.nonneg** {P : ReciprocityTheory.ChartRestriction.G.Out.PolAV (Sp K₁)} {rig : LB.Gen (P.L.pull (eSec P.A))} {σ : K₁ →+* ℂ} {s : P.L.Sec} {v : ℝ} (h : IsL2SquareIntegralProblem P rig σ s v) : 0 ≤ v

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.LatticeModelCharts

theorem **mem_OC** {p : ℕ} (F : CF p) (x : F.C) : x ∈ OC F ↔ ‖x‖ ≤ 1

theorem **exists_smallField** {p : ℕ} (F : CF p) (h : #F.C ≤ 𝔠) : ∃ (F' : SmallField p) (e : F'.toCF.C ≃+* F.C), ∀ x, ‖e x‖ = ‖x‖

theorem **mk_le_continuum_of_denseRange** {A X : Type} [MetricSpace X] (f : A → X) (hf : DenseRange f) (hA : #A ≤ 𝔠) : #X ≤ 𝔠

theorem **mk_padicComplex_le** (p : ℕ) [Fact p.Prime] : #(PadicComplex p) ≤ 𝔠

theorem **nonempty_smallField** (p : ℕ) [Fact p.Prime] : Nonempty (SmallField p)

theorem **exists_evalU** (V : Type) [CommRing V] [IsLocalRing V] [IsAdicComplete (IsLocalRing.maximalIdeal V) V] (π : V) (hπ : π ∈ IsLocalRing.maximalIdeal V) : ∃ φ : V⟦X⟧ →+* V, (∀ a, φ (PowerSeries.C a) = a) ∧ φ PowerSeries.X = π

theorem **evalU_unique** (V : Type) [CommRing V] [IsLocalRing V] [IsAdicComplete (IsLocalRing.maximalIdeal V) V] (π : V) (hπ : π ∈ IsLocalRing.maximalIdeal V) (φ ψ : V⟦X⟧ →+* V) (hφ : ∀ a, φ (PowerSeries.C a) = a) (hφ' : φ PowerSeries.X = π) (hψ : ∀ a, ψ (PowerSeries.C a) = a) (hψ' : ψ PowerSeries.X = π) : φ = ψ

theorem **evalU_spec** (V : Type) [CommRing V] [IsLocalRing V] [IsAdicComplete (IsLocalRing.maximalIdeal V) V] (π : V) (hπ : π ∈ IsLocalRing.maximalIdeal V) : (∀ a, evalU V π hπ (PowerSeries.C a) = a) ∧ evalU V π hπ PowerSeries.X = π

theorem **DVRing.Unif.mem** {B : OLam p} {V : DVRing B} (π : V.Unif) : π.1 ∈ IsLocalRing.maximalIdeal V.V

theorem **isAbSch_trivial** (S : Scheme.{0}) : ReciprocityTheory.ChartRestriction.G.Semi.IsAbelianScheme (Grp.trivial (Over S)).X

theorem **OLam.p_ne_zero** (B : OLam p) : (p : B.O) ≠ 0

theorem **nonempty_pt** (p : ℕ) [Fact p.Prime] : Nonempty (Model.base (OLam.padic p)).Pt

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.LevelCuspCharacters

theorem **diag_mul_of_upper** {r : ℕ} {k : Type} [Field k] (A B : Matrix (Fin r) (Fin r) k) (hA : A.BlockTriangular id) (hB : B.BlockTriangular id) (i : Fin r) : (A * B) i i = A i i * B i i

theorem **toTorOne_bijective** : Function.Bijective (toTorOne p)

theorem **cycLen_pos** {r : ℕ} (b : Fin r → Fin r) (c : Cyc b) : 0 < cycLen b c

theorem **cycLen_dvd** {r : ℕ} (b : Fin r → Fin r) (c : Cyc b) : cycLen b c ∣ r.factorial

theorem **goodTypes_finite** (om : ∀ i : Fin 2, (ZMod (H.g i))ˣ →* Kˣ) : (goodTypes om).Finite

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.LieLiftsTangentReduction

theorem **IsGtTangent.isTangent** {R : Type} [CommRing R] [IsDomain R] {D : DegPre R} {ν : GtPt D (epsR R)} (h : IsGtTangent ν) : IsTangent ν.b

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.LinePeriodExponential

theorem **chi_ne_one** (H : Host E) : chi H ≠ 1

theorem **LFunction_ne_zero** (H : Host E) : (chi H).LFunction 1 ≠ 0 ∧ DifferentiableAt ℂ (chi H).LFunction 1

theorem **HasPeriod.unique** {M : Type} [Field M] [NumberField M] {ω : 𝒪 M} {C : Ell H (𝒪 M) ω} {x y : ℝ} (hx : HasPeriod C x) (hy : HasPeriod C y) : x = y

theorem **isPolFam_of_isTen** {C : Ell H R ω} {D : Datum H} {ε : ℤˣ} {h : D.La → D.La → OK H} (hh : IsKForm D ε h) {T : PAT H (Sp R)} (X : IsTen C D h T) : IsPolFam C T.t (⊤ : Submodule (𝒪 E) D.La) ε fun a => X.ten.tmul a.1

theorem **isHolTo_comp** {V W : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] [NormedAddCommGroup W] [NormedSpace ℂ W] {X : Scheme.{0}} {B : Set W} {y : W → (Sp ℂ ⟶ X)} {c : V → W} (hc : Differentiable ℂ c) (hB : ∀ z, c z ∈ B) (h : IsHolTo B y) : IsHolTo Set.univ fun z => y (c z)

theorem **isExp_of_familyUnif** {D : Datum H} {l : Level H} {Xo : Moduli D l} {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (FU : FamilyUnif Xo σ ty) (i : Fin FU.n) {Pe : (FU.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (FU.Di i) ty Pe) : IsExp Xo.univ.t (FU.u i Pe) (FU.Di i) ty Pe (FU.ex i Pe)

theorem **nonempty_latData** {D D' : Datum H} (hp : D.p = 1) (hq : D.q = 0) (h' : D'.p = 0 ∧ D'.q = 1) {g : ℕ} {ty : Fin g → (E →+* ℂ)} (hty : IsTypeFn H 1 0 ty) {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) : Nonempty (LatData hP 1)

theorem **deg_formula** (h : KeyDegreeSt) (x : KeyIn K) (oa od : HeightOneSpectrum (𝒪 K) → ℤ) (nth nη : (K →+* ℂ) → ℝ) (h₁ : ∀ p, ∃ a M, KeyPow.IsLocalModel x p a M ∧ p.valuation K a = WithZero.exp (-oa p) ∧ p.valuation K M.det = WithZero.exp (-od p)) (h₂ : ∀ σ : K →+* ℂ, ∃ x' : KeyIn ℂ, KeyPB x' x σ ∧ KeyPow.IsArchNorm x' (nth σ) (nη σ)) (ha : (Function.support oa).Finite) (hd : (Function.support od).Finite) : IsDegDe x ((1 / 2) * ((∑ᶠ p, (od p : ℝ) * Real.log (Ideal.absNorm p.asIdeal) - (1 / 2) * ∑ σ, Real.log (nη σ)) / Module.finrank ℚ K) + x.P.g / 4 * Real.log (2 * Real.pi))

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.OmegaUpDownForms

theorem **IsSplitting.symm** {A B : HSp D} {χA χB : (AE D)ˣ →* Circle} (h : IsSplitting A B χA χB) : IsSplitting B A χB χA

theorem **rank_Y_lt** (X : ThetaX D) {j : ℕ} (hj : j < D.R.m) : (D.sp.Y j).rank < (D.sp.Y (j + 1)).rank

theorem **kern_right** (f : Theta ψ A B) (g g' : GA D A) (h h' : GA D B) : kern om f (g * g') (h * h') = kern om (om (g', h') f) g h

theorem **kern_left** (hw : IsWeil ψ χA χB om) (f : Theta ψ A B) {γ : GA D A} (hγ : γ ∈ GQ A) {δ : GA D B} (hδ : δ ∈ GQ B) (g : GA D A) (h : GA D B) : kern om f (γ * g) (δ * h) = kern om f g h

theorem **kern_cont** (hw : IsWeil ψ χA χB om) (f : Theta ψ A B) : Continuous fun p : GA D A × GA D B => kern om f p.1 p.2

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.OrderValuation

theorem **NoCommonZero.at** {κ : Type} {w : κ → Rul Xo c} (h : NoCommonZero w) (la : HeightOneSpectrum (𝒪 c.F)) : NoCommonZeroAt la w

theorem **TransferAt.transfer** (h : TransferAt Xo c U) : Transfer Xo c U

theorem **ordVal_smul** {o : Rul Xo c → EReal} (ho : o ∈ ordVal Xo c) {a : c.F} (ha : a ≠ 0) (f : Rul Xo c) : o (a • f) = o f

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.SchurLatticeConjugates

theorem **conjDifferent_ne_bot** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] (σ : E →+* K) : conjDifferent σ ≠ ⊥

theorem **span_natCast_ne_bot** {K : Type} [Field K] [NumberField K] (n' : ℕ+) : Ideal.span {((n' : ℕ) : 𝒪 K)} ≠ ⊥

theorem **defectAt_vw_eq_mul** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] (σ : E →+* K) (n' : ℕ+) (a : ℕ) (la : HeightOneSpectrum (𝒪 K)) (e : ℝ) (he : 0 < e) : (defectAt_vw σ n' a la : ℝ) = e * a * max ((ord_vw la (conjDifferent σ) : ℝ) / e) ((ord_vw la (Ideal.span {((n' : ℕ) : 𝒪 K)}) : ℝ) / e)

theorem **defectAt_vw_le** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] (σ : E →+* K) (n' : ℕ+) (a : ℕ) (la : HeightOneSpectrum (𝒪 K)) : defectAt_vw σ n' a la ≤ defect_vw E n' a la

theorem **le_defectAt_vw_of_mem** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] (σ : E →+* K) (n' : ℕ+) (a : ℕ) (la : HeightOneSpectrum (𝒪 K)) (h : ((n' : ℕ) : 𝒪 K) ∈ la.asIdeal) : a ≤ defectAt_vw σ n' a la

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.SectionInputData

theorem **valZ_wtZ** (h : GenFrEx Xo c) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (f : Rul Xo c) : valZ h (wtZ a n) f = valC h a n f

theorem **valZ_mem** (h : GenFrEx Xo c) (G : WTZ E) (f : Rul Xo c) (Z : Comp Xo c) : valZ h G f Z ∈ AG Z G

theorem **isSecZ_wtZ** (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (f : Rul Xo c) : IsSecZ (wtZ a n) f ↔ f ∈ sec Xo c a n

theorem **isSecZ_one** : IsSecZ (0 : WTZ E) (1 : Rul Xo c)

theorem **IsSecZ.mul** {G G' : WTZ E} {f g : Rul Xo c} (hf : IsSecZ G f) (hg : IsSecZ G' g) : IsSecZ (G + G') (f * g)

theorem **IsSecZ.pow** {G : WTZ E} {f : Rul Xo c} (hf : IsSecZ G f) (k : ℕ) : IsSecZ (k • G) (f ^ k)

theorem **neOnComps_iff** (h : GenFrEx Xo c) (G : WTZ E) (f : Rul Xo c) : NeOnComps h f ↔ ∀ Z, valZ h G f Z ≠ 0

theorem **neOnComps_one** (h : GenFrEx Xo c) (ht : GenFromTest Xo c h) : NeOnComps h (1 : Rul Xo c)

theorem **toData_ka** : (I.toData h δ fn yP).ka = I.wka

theorem **toData_F₀** : (I.toData h δ fn yP).F0 = I.wF₀

theorem **toData_G₂** : (I.toData h δ fn yP).G2 = I.wG₂

theorem **toData_et_mem** (Z : Comp Xo c) : (I.toData h δ fn yP).et Z ∈ AG Z ((I.toData h δ fn yP).n1 • (I.toData h δ fn yP).wde)

theorem **toData_torsionSection_mem** (Z : Comp Xo c) : (I.toData h δ fn yP).ta1 Z ∈ AG Z ((2 : ℕ) • (I.toData h δ fn yP).G1)

theorem **toData_M_mem** (x : ∀ Z : Comp Xo c, AC Z) (hx : x ∈ (I.toData h δ fn yP).M) (Z : Comp Xo c) : x Z ∈ AG Z (I.toData h δ fn yP).ka

theorem **ofSection_weighted** (D T : ℕ) {L : WTZ E} {s : Rul Xo c} (hs : IsSecZ L s) : (ofSection D T L s).Weighted where

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.ShiftMapsUpDown

theorem **mapUp_add** (V W : HSp D) (x y : WA V W) : mapUp V W (x + y) = mapUp V W x + mapUp V W y

theorem **mapDown_add** (V W : HSp D) (x y : WA V W) : mapDown V W (x + y) = mapDown V W x + mapDown V W y

theorem **mapUp_neg** (V W : HSp D) (x : WA V W) : mapUp V W (-x) = -mapUp V W x

theorem **mapDown_neg** (V W : HSp D) (x : WA V W) : mapDown V W (-x) = -mapDown V W x

theorem **mapUp_zero** (V W : HSp D) : mapUp V W 0 = 0

theorem **mapDown_zero** (V W : HSp D) : mapDown V W 0 = 0

theorem **IsVacuum.mono** (Wl : WeilTied D X T) (V W : HSp D) {S S' : Finset (HeightOneSpectrum (𝒪 D.host.F))} (hS : S ⊆ S') {Φ : X.Sch V W} (h : IsVacuum Wl V W S Φ) : IsVacuum Wl V W S' Φ

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.SphereBlockLogMean

theorem **sphereMeasure_map_unitary** {N : ℕ} (e : EuclideanSpace ℂ (Fin (N + 1)) ≃ₗᵢ[ℂ] EuclideanSpace ℂ (Fin (N + 1))) : (sphereMeasure N).map (sphereMap e) = sphereMeasure N

theorem **finitePlace_eq_absNorm_zpow** (w : FinitePlace K) {c : K} (hc : c ≠ 0) : w c = (Ideal.absNorm w.maximalIdeal.asIdeal : ℝ) ^ (WithZero.log (w.maximalIdeal.valuation K c))

theorem **le_gaussAbsValue** {ι : Type} (w : FinitePlace K) (ω : MvPolynomial ι K) (d : ι →₀ ℕ) : w (coeff d ω) ≤ gaussAbsValue w ω

theorem **exists_gaussAbsValue_eq** {ι : Type} (w : FinitePlace K) {ω : MvPolynomial ι K} (h : ω ≠ 0) : ∃ d ∈ ω.support, gaussAbsValue w ω = w (coeff d ω)

theorem **gaussAbsValue_pos** {ι : Type} (w : FinitePlace K) {ω : MvPolynomial ι K} (h : ω ≠ 0) : 0 < gaussAbsValue w ω

theorem **finite_support_log_gaussAbsValue** {ι : Type} {ω : MvPolynomial ι K} (h : ω ≠ 0) : (Function.support fun w : FinitePlace K => Real.log (gaussAbsValue w ω)).Finite

theorem **heightZeroOnFractions_of_additive** (h : HeightZeroAdditive) : HeightZeroOnFractions

theorem **eval_hermitianLinearForm** {N : ℕ} (u z : EuclideanSpace ℂ (Fin (N + 1))) : eval (fun j => z j) (hermitianLinearForm u) = inner ℂ u z

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.SymplecticTensorMeasures

theorem **cE_cE** (x : D.host.E) : cE D (cE D x) = x

theorem **cA_inc** (x : D.host.E) : cA D (incA D x) = incA D (cE D x)

theorem **cA_cA** (x : AE D) : cA D (cA D x) = x

theorem **conj_map_mem_unitaryGroup** {R : Type} [CommRing R] {n : ℕ} (c : R →+* R) (J : Matrix (Fin n) (Fin n) R) (P g : GL (Fin n) R) (hP : Formalism.UG.cT c P.val * J * P.val = J.map c) (hg : g ∈ Formalism.UG.UJ c J) : (MulAut.conj P).toMonoidHom.comp (Matrix.GeneralLinearGroup.map c) g ∈ Formalism.UG.UJ c J

theorem **isoA** : Formalism.UG.cT (cA D) s.PA.val * V.gram.map (incA D) * s.PA.val = (V.gram.map (incA D)).map (cA D)

theorem **autGL_mem** (g : GL (Fin V.rank) (AE D)) (hg : g ∈ HR (AF D.host.F) D.sp.cm.sig V.gram) : s.autGL g ∈ HR (AF D.host.F) D.sp.cm.sig V.gram

theorem **mt_aut** (g : GA D V) : mt V (s.aut g) = s.PA.val * cM (mt V g) * (s.PA⁻¹).val

theorem **gram_herm** (V : HSp D) : Formalism.UG.cT (cE D) V.gram = V.gram

theorem **basConj_fix** {X : ThetaX D} {T : ThetaTied D X} {Wl : WeilTied D X T} (K : KfinTied Wl) (V : HSp D) (hV : V.SigOK) : (basConj K V hV).P.val * (K.bas V).map (cE D) = K.bas V

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.TensorObjectExistence

theorem **isExpAt_of_familyUnif** {D : Datum H} {l : Level H} {Xo : Moduli D l} {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (U : FamilyUnif Xo σ ty) (i : Fin U.n) (Pe : (U.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (U.Di i) ty Pe) : IsExpAt Xo.univ.t (U.u i Pe) (U.Di i) ty Pe (U.ex i Pe)

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.ThetaWeightHeightConstants

theorem **pSig_pos** {R : Ranks} (j : Lift R) : 0 < pSig j

theorem **pSig_add_qSig** {R : Ranks} (j : Lift R) : pSig j + qSig j = R.r

theorem **wittIndexOf_isGreatest** {E : Type} [Field E] [NumberField E] [StarRing E] (S : HSpace E) : IsGreatest (isotropicDims S) (wittIndexOf S)

theorem **rho_eq_qSig** {R : Ranks} (j : Lift R) : rho j = qSig j

theorem **dPrime_le** {R : Ranks} (j : Lift R) : dPrime j ≤ 2 * R.f * R.m

theorem **PP_eq_params** {D : D7} {L : SizeData₂ D} {P : ParameterChoice' D L} (T : ParamsT D L P) : PP L T.CP T.eP = T.PP

theorem **log_b** {D : D7} (L : SizeData₂ D) : Real.log L.sc.b = L.sc.L

theorem **size_add** (G G' : ThetaWeight D j) : (G.add G').size = G.size + G'.size - 1

theorem **size_nsmul_le** (N : ℕ) (hN : 1 ≤ N) (G : ThetaWeight D j) : (G.nsmul N).size ≤ N * G.size

theorem **one_le_size** {G : ThetaWeight D j} (h : G.Dominant) : 1 ≤ G.size

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.TotallyPositiveExponents

theorem **tau_conj** (H : Host R) (τ : Pl R) (x : H.E) : tau H τ (H.conj x) = starRingEnd ℂ (tau H τ x)

theorem **conj_conj** (H : Host R) (x : H.E) : H.conj (H.conj x) = x

theorem **smul_dot** (a : 𝒪 H.E) (y x : V H n) : (a • y) ·ᵥ x = y ·ᵥ (a • x)

theorem **conj_val** (χ : Herm H n) (y y' : V H n) : H.conj (val χ y y') = val χ y' y

theorem **level_pos** (H : Host R) : 0 < level H

theorem **toR_smul** (a : 𝒪 H.E) (x : V H n) : toR H n (a • x) = act (a : H.E) (toR H n x)

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.UnramifiedMeasureTuples

theorem **inv_lower_mem** {O : Type} [CommRing O] (I : Ideal O) (r d : ℕ) (h : GL (Fin (r + d)) O) (hh : ∀ i k : Fin (r + d), r ≤ i.1 → k.1 < r → (h : Matrix _ _ O) i k ∈ I) : ∀ i k : Fin (r + d), r ≤ i.1 → k.1 < r → ((h⁻¹ : GL (Fin (r + d)) O) : Matrix _ _ O) i k ∈ I

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.WeilIdempotents

theorem **herm_add_left** {A B : HSp D} (x y z : WA A B) : herm (x + y) z = herm x z + herm y z

theorem **adj_add** {A B : HSp D} (x y : WA A B) : adj (x + y) = adj x + adj y

theorem **herm_add_right** {A B : HSp D} (x y z : WA A B) : herm x (y + z) = herm x y + herm x z

theorem **herm_smul_left** {A B : HSp D} (c : AE D) (x y : WA A B) : herm (c • x) y = c ∗ herm x y

theorem **symp_add_left** {A B : HSp D} (x y z : WA A B) : symp (x + y) z = symp x z + symp y z

theorem **symp_add_right** {A B : HSp D} (x y z : WA A B) : symp x (y + z) = symp x y + symp x z

theorem **symp_zero_left** {A B : HSp D} (x : WA A B) : symp 0 x = 0

theorem **symp_zero_right** {A B : HSp D} (x : WA A B) : symp x 0 = 0

theorem **symp_neg_left** {A B : HSp D} (x y : WA A B) : symp (−x) y = −symp x y

theorem **gMat_one** {V : HSp D} : gMat (1 : GA D V) = 1

theorem **gMat_mul** {V : HSp D} (g g' : GA D V) : gMat (g ∗ g') = gMat g ∗ gMat g'

theorem **act_one** {A B : HSp D} (x : WA A B) : act 1 x = x

theorem **act_mul** {A B : HSp D} (p q : GA D A × GA D B) (x : WA A B) : act (p ∗ q) x = act p (act q x)

theorem **rho_apply** {A B : HSp D} (ψ : AddChar (AF D.host.F) ℂ) (u : WA A B) (f : WA A B → ℂ) (w : WA A B) : rho ψ u f w = (ψ (hsymp w u) : ℂ) ∗ f (w + u)

theorem **rho_mul** {A B : HSp D} (ψ : AddChar (AF D.host.F) ℂ) (u u' : WA A B) : rho ψ u ∗ rho ψ u' = (ψ (hsymp u u') : ℂ) • rho ψ (u + u')

Development.AbelianVarietyTheory.BlockTheory.HeightDefect.WordDerivatives

theorem **isSmoothInf_const** (V : HSp D) (e : LieInf V → GA D V) (c : ℂ) : IsSmoothInf V e fun _ => c

theorem **eventually_finNorm_le_one** (V : HSp D) (g : GA D V) : ∀ᶠ v in Filter.cofinite, finNorm V v g ≤ 1

theorem **finHt_mulSupport_finite** (V : HSp D) (g : GA D V) : (Function.mulSupport fun v => finHt V v g).Finite

theorem **one_le_height** (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (V : HSp D) (g : GA D V) : 1 ≤ height θ V g

theorem **wordDer_const** (V : HSp D) (e : LieInf V → GA D V) (l : List (LieInf V)) (c : ℂ) : ∃ c' : ℂ, wordDer V e l (fun _ => c) = fun _ => c'

Development.AbelianVarietyTheory.BlockTheory.IrreducibleCharacters.CuspidalCharacters

theorem **isIrreducible_iff** {G : Type*} [Group G] {V : Type*} [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) : ρ.IsIrreducible ↔ Nontrivial V ∧ ∀ U : Submodule ℂ V, (∀ g : G, ∀ v ∈ U, ρ g v ∈ U) → U = ⊥ ∨ U = ⊤

theorem **InUpperRightBlock.mul_eq_zero** {F : Type*} [Field F] {d a : ℕ} {M N : Matrix (Fin d) (Fin d) F} (hM : InUpperRightBlock a M) (hN : InUpperRightBlock a N) : M ∗ N = 0

Development.AbelianVarietyTheory.BlockTheory.JetGauges.BalancedFrameGauge

theorem **isBalancedGauge_of_canon** (B : FB R d O) (h : PS k 2 d) (hh : B.IsCanonGauge k h) : IsBalancedGauge k B h

theorem **eval_addSeries_two** (a : Fin 2 → Pt k n g) (ha : ∀ p : Fin 2 × Fin g, constantCoeff (a p.1 p.2) = 0) (i : Fin g) : ev a (addSeries k g i) = a 0 i + a 1 i

theorem **eval_const_mul_X** (a : Fin 1 → Pt k n g) (ha : ∀ p : Fin 1 × Fin g, constantCoeff (a p.1 p.2) = 0) (c : k) (i : Fin g) : ev a (C c ∗ X ((0 : Fin 1), i) : PS k 1 g) = C c ∗ a 0 i

theorem **addOf_addSeries** (a b : Pt k n g) (ha : ∀ i, constantCoeff (a i) = 0) (hb : ∀ i, constantCoeff (b i) = 0) : addOf (addSeries k g) a b = a + b

theorem **expS_zero** [Algebra ℚ k] : FB.expS k (0 : PS k n g) = 1

theorem **cc_pt** (j : Fin n) (i : Fin g) : constantCoeff (pt k g j i) = 0

theorem **cc_diag** (χ : Fin g → (O →+* k)) (a : O) (i : Fin g) : constantCoeff (diagSeries χ a i) = 0

theorem **comp_diagonalSeries** (χ : Fin g → (O →+* k)) (a : O) (b : Pt k n g) (hb : ∀ i, constantCoeff (b i) = 0) : comp1 (diagSeries χ a) b = fun i => C (χ i a) ∗ b i

Development.AbelianVarietyTheory.BlockTheory.JetGauges.BalancedHermitianSpace

theorem **IsBalancedSpace.eq** {S : Submodule ℝ (Bil Y)} (h : IsBalancedSpace E S) : S = balancedSpace E Y

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.BalancedHermitianSpace.isIntegralFn_evalSum** (S : Submodule ℝ (_root_.GlobalTheory.IndefiniteRowSideFullHecke.BalancedHermitianSpace.Bil Y)) {k : ℕ} (x x' : Fin k → Y) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.BalancedHermitianSpace.IsIntegralFn S (_root_.GlobalTheory.IndefiniteRowSideFullHecke.BalancedHermitianSpace.evalSum S x x')

Development.AbelianVarietyTheory.BlockTheory.JetGauges.BarTorsionGenerators

theorem **gens_subset_thGen** (Xo : Moduli Dw l) (c : Coeff l) (nab : NabV Xo c) (α : 𝒪 E) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : gens Xo c nab α a n ⊆ thGen Xo c nab a n

theorem **gens_eq_thGen** (Xo : Moduli Dw l) (c : Coeff l) (nab : NabV Xo c) {α : 𝒪 E} (hα : l.Mb = Ideal.span {H.bar α}) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : gens Xo c nab α a n = thGen Xo c nab a n

theorem **NoCommonZero.rd** {Xo : Moduli Dw l} {c : Coeff l} {σ : c.F →+* ℂ} {s : Set (Rul Xo c)} (h : NoCommonZero Xo c σ s) {r : ℕ} (U : Unf Xo c σ r) (k : U.Comp) (x) (hx : x ∈ (U.X k).dom) : ∃ f ∈ s, rdU U f k x ≠ 0

theorem **LongNoCommonZero.full** (hA : LongNoCommonZero) (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j) (α : 𝒪 D.host.E) (hn : D.R.r ≤ w.n) (hα : IsJetLevel T.Q.l α T.Q.l.M) (hD : 4 * WeightAlgebra.DF w + 4 ≤ T.Q.l.M) (σ : T.Q.c.F →+* ℂ) (hσ : GoodEmb T.Q.c σ) (k) (x) (hx : x ∈ ((T.U j σ).X k).dom) : ∃ f ∈ T.th j w, rdU (T.U j σ) f k x ≠ 0

Development.AbelianVarietyTheory.BlockTheory.JetGauges.BiadditiveBalancedSeries

theorem **IsBiaddBalanced.of_canon** {g : PS k 2 d} (h : B.IsCanonGauge k g) : IsBiaddBalanced B k g

theorem **cc_pt** (j : Fin n) (i : Fin d) : constantCoeff (pt k d j i) = 0

theorem **addOf_addLaw** (a b : Pt k n d) (ha : ∀ i, constantCoeff (a i) = 0) (hb : ∀ i, constantCoeff (b i) = 0) : addOf (addLaw k d).F a b = a + b

theorem **ev_XX** (a b : Pt k n d) (ha : ∀ i, constantCoeff (a i) = 0) (hb : ∀ i, constantCoeff (b i) = 0) (i j : Fin d) : ev ![a, b] (X ((0 : Fin 2), i) * X ((1 : Fin 2), j) : PS k 2 d) = a i * b j

theorem **comp_diagonalAction** (ty : Fin d → (O →+* k)) (a : O) (b : Pt k n d) (hb : ∀ i, constantCoeff (b i) = 0) : comp1 (diagAct ty a) b = fun i => C (ty i a) * b i

theorem **expS_zero** : FB.expS k (0 : PS k n d) = 1

theorem **isBiaddBalanced_XX** (ty : Fin d → (O →+* k)) (i j : Fin d) (hij : ∀ a, ty i a = ty j (star a)) : IsBiaddBalanced (FB.triv (addLaw k d) (diagOAct ty)) k (X ((0 : Fin 2), i) * X ((1 : Fin 2), j))

theorem **XX_not_symm** (i j : Fin d) (hij : i ≠ j) : ev (n

theorem **isHarmonic_X** [IsCMField E] (r : (E →+* k) → ℕ) {n : ℕ} (p : (Σ τ : E →+* k, Fin (r τ)) × Fin n) : IsHarmonic r (MvPolynomial.X p : MvPolynomial _ k)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.CMObjBundlePull

theorem **mem_psiBar_of_notMem_psi** (H : Host E) (τ : E →+* ℂ) (h : τ ∉ psi H) : τ ∈ psiBar H

Development.AbelianVarietyTheory.BlockTheory.JetGauges.CanonicalHarmonicGerms

@[simp] theorem **column_apply** {n : ℕ} (m : LtN Dw n →₀ ℕ) (b : Fin n) (ℓ : Lt1 Dw) : column m b ℓ = m (ℓ.1, ℓ.2, b)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.ConjugateKernelCount

theorem **torsionFactsStatement_iff** : TorsionFactsStatement ↔ ConjKernelCountStatement ∧ FixingLiftsStatement ∧ PairingSeparatesStatement ∧ FiniteOrderLiftsStatement ∧ PullFrameNuZeroStatement

Development.AbelianVarietyTheory.BlockTheory.JetGauges.ConjugatePlaceEigen

theorem **valuation_coe_eq_norm** (x : ℂ_[p]) : ((Valued.v x : NNReal) : ℝ) = ‖x‖

theorem **norm_p** : ‖((p : ℕ) : ℂ_[p])‖ = (p : ℝ)⁻¹

theorem **inv_p_lt_one** : (p : ℝ)⁻¹ < 1

theorem **mem_iff_norm** (x : ℂ_[p]) : x ∈ 𝒪_ℂ_[p] ↔ ‖x‖ ≤ 1

theorem **smodEq_iff** (n : ℕ) (x y : 𝒪_ℂ_[p]) : x ≡ y [SMOD ((span {((p : ℕ) : 𝒪_ℂ_[p])}) ^ n • ⊤ : Submodule 𝒪_ℂ_[p] 𝒪_ℂ_[p])] ↔ ‖(x : ℂ_[p]) - (y : ℂ_[p])‖ ≤ (p : ℝ)⁻¹ ^ n

theorem **tendsto_inv_p_pow** : Filter.Tendsto (fun n : ℕ => (p : ℝ)⁻¹ ^ n) Filter.atTop (nhds 0)

theorem **isHausdorff** : IsHausdorff (span {((p : ℕ) : 𝒪_ℂ_[p])}) 𝒪_ℂ_[p]

theorem **isPrecomplete** : IsPrecomplete (span {((p : ℕ) : 𝒪_ℂ_[p])}) 𝒪_ℂ_[p]

theorem **isAdicComplete** : IsAdicComplete (span {((p : ℕ) : 𝒪_ℂ_[p])}) 𝒪_ℂ_[p]

theorem **setting_padicComplex** : HidaSetting p (PadicComplex.valued p).v 𝒪_ℂ_[p] where

Development.AbelianVarietyTheory.BlockTheory.JetGauges.DecreasingNodeGaps

theorem **card_eq** (r k : ℕ) (hk : 1 ≤ k) : 2 * r + k + 1 = (k - 1) + (2 * r + 2)

@[simp] theorem **param_cloud** (r k : ℕ) (hk : 1 ≤ k) (v : Fin (k - 1) → ℝ) (b : Fin (2 * r + 2) → ℝ) (j : Fin (k - 1)) : param r k hk v b (cloudIdx r k j) = v j

@[simp] theorem **param_node** (r k : ℕ) (hk : 1 ≤ k) (v : Fin (k - 1) → ℝ) (b : Fin (2 * r + 2) → ℝ) (i : Fin (2 * r + 2)) : param r k hk v b (nodeIdx r k hk i) = -b i

@[simp] theorem **sign_cloud** (r k : ℕ) (hk : 1 ≤ k) (j : Fin (k - 1)) : sign r k hk (cloudIdx r k j) = true

@[simp] theorem **sign_node** (r k : ℕ) (hk : 1 ≤ k) (i : Fin (2 * r + 2)) : sign r k hk (nodeIdx r k hk i) = decide (Even i.val)

theorem **parity_of_isShiftedIntegral** (r k : ℕ) (lam : Fin (2 * r + k + 1) → ℝ) (u : Fin k → ℝ) (hl : IsShiftedIntegral lam) (hu : IsShiftedIntegral u) : Parity r k lam u

theorem **nodeGapGe_of_noncompactGapGe** (r k : ℕ) (hk : 1 ≤ k) (v : Fin (k - 1) → ℝ) (b : Fin (2 * r + 2) → ℝ) (h : ℝ) (hd : IsDecreasing r k hk v b) (hg : NoncompactGapGe r k hk (param r k hk v b) h) : NodeGapGe r b h

Development.AbelianVarietyTheory.BlockTheory.JetGauges.DiagonalPowerBalanced

theorem **diagOnB_iff** {D' D : DegPre R} {n : ℕ} (p : IsPowDeg D' D n) (F : D'.P.A ⟶ D'.P.A) (f : D.P.A ⟶ D.P.A) : DiagOnB p F f ↔ F = diagB p f

theorem **diagOnX_iff** {D' D : DegPre R} {n : ℕ} (p : IsPowDeg D' D n) (A : D'.X → D'.X) (a : D.X → D.X) : DiagOnX p A a ↔ A = diagX p a

theorem **CLinear.pow** {D' D : DegPre R} {n : ℕ} (p : IsPowDeg D' D n) {F : D'.P.A ⟶ D'.P.A} {f : D.P.A ⟶ D.P.A} {A : D'.X ⟶ D'.X} {a : D.X ⟶ D.X} (hF : DiagOnB p F f) (hA : DiagOnX p A a) (h : CLinear D f a) : CLinear D' F A

theorem **PowTauBalanced.act** (h : PowTauBalanced) : PowTauBalancedAct

Development.AbelianVarietyTheory.BlockTheory.JetGauges.EigenLatticeRankLie

theorem **mem_eigenLattice_iff** (hx : Chart.IsUnivChart G n x) {O : Type} [CommRing O] (act : OAct O G) (τ : O →+* R) (c : Fin n → R) : c ∈ eigenLattice hx act τ ↔ ∀ a, Chart.tvec (x

Development.AbelianVarietyTheory.BlockTheory.JetGauges.EndomorphismPullbackRigid

lemma **Lands.le_wchart** {L : LB A.left} {W Z : Over X} {x : Z ⟶ A} {p : W ⟶ Z} {v : W ⟶ A} (hv : p ≫ x = v) {i : L.ι} (h : Lands L v i) : ⊤ ≤ p.left ⁻¹ᵁ LB.wchart L (K

lemma **origin_comp** {T W : Over X} (t : W ⟶ T) (φ : A ⟶ A) [IsMonHom φ] : lift t (1 : W ⟶ A) ≫ snd T A ≫ φ = 1

lemma **lift_transl** {T W : Over X} (b : T ⟶ A) (t : W ⟶ T) (v : W ⟶ A) : lift t v ≫ transl b = v ∗ (t ≫ b)

lemma **lift_neg** {T W : Over X} (t : W ⟶ T) (v : W ⟶ A) : lift t v ≫ snd T A ≫ ι[A] = v⁻¹

lemma **linComb_trivL_g** {Y Z : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Z ⟶ Y)) (ε : K → ℤ) (j j') : (LB.LinComb (trivL Y) φ ε).g j j' = 1

lemma **trivIso_isGen** {T : Over X} (f : T ⊗ A ⟶ A) (a b : ℤ) : (trivIso f a b).IsGen

lemma **isoVal_trivIso** {T W : Over X} (f : T ⊗ A ⟶ A) (a b : ℤ) (t : W ⟶ T) (v u : W ⟶ A) (hu : lift t v ≫ f = u) (i j) (hi) (hj) : isoVal (trivIso f a b) t v u hu i j hi hj = 1

lemma **secVal_trivOne** {T W : Over X} (k : ℤ) (t : W ⟶ T) (v : W ⟶ A) (i) (hi) : secVal (trivOne k T) t v i hi = 1

lemma **rigVal_trivRig** (W : Over X) (i) (hi) : rigVal (trivRig A) W i hi = 1

theorem **isTranslatedPull_triv** (φ : A ⟶ A) [IsMonHom φ] {N : ℕ} (hN : Odd N) {T : Over X} (b : T ⟶ A) (hb : b ^ N = 1) : IsTranslatedPull (trivRig A) φ N b (trivOne 1 T) (trivOne N T)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.FormalDegreeHaarQuotient

theorem **col_of_fix** {h : Mat k r} {j : Fin (2 ∗ r + k + 1)} (hj : h ∗ᵥ Formalism.QuantumDeterminants.bvec j = Formalism.QuantumDeterminants.bvec j) (i : Fin (2 ∗ r + k + 1)) : h i j = if i = j then 1 else 0

theorem **col_out** {z : Fin (2 ∗ r + k + 1) → ℂ} {h : Mat k r} (hh : h ∈ UWset k r z) {j : Fin (2 ∗ r + k + 1)} (hj : ¬ (r ≤ j.val ∧ j.val ≤ r + k)) (i : Fin (2 ∗ r + k + 1)) : h i j = if i = j then 1 else 0

theorem **Hpk_row** {i i' : Fin (2 ∗ r + k + 1)} (hi : ¬ (r ≤ i.val ∧ i.val ≤ r + k)) (e : i.val + i'.val = 2 ∗ r + k) (l : Fin (2 ∗ r + k + 1)) : Hpk k r i' l = if l = i then 1 else 0

theorem **row_out** {z : Fin (2 ∗ r + k + 1) → ℂ} {h : Mat k r} (hh : h ∈ UWset k r z) {i : Fin (2 ∗ r + k + 1)} (hi : ¬ (r ≤ i.val ∧ i.val ≤ r + k)) (j : Fin (2 ∗ r + k + 1)) : h i j = if i = j then 1 else 0

theorem **leviW_mul** {z : Fin (2 ∗ r + k + 1) → ℂ} {h n : Mat k r} (hh : h ∈ UWset k r z) (hn : n ∈ Nset k r) : leviW k r (h ∗ n) = h

theorem **isUnit_det_of_mem** {h : Mat k r} (hh : h ∈ UH (Hpk k r)) (hH : IsUnit (Hpk k r).det) : IsUnit h.det

theorem **twist_mul** {Wp : Type∗} [NormedAddCommGroup Wp] [InnerProductSpace ℂ Wp] {z : Fin (2 ∗ r + k + 1) → ℂ} (c : Fin r → ℂ) (π : Mat k r → (Wp →L[ℂ] Wp)) {h n : Mat k r} (hh : h ∈ UWset k r z) (hn : n ∈ Nset k r) (hd : IsUnit h.det) : twist k r z c π (h ∗ n) = psiGen k r z c n • π h ∧ twistInv k r z c π (h ∗ n) = (psiGen k r z c n)⁻¹ • π h⁻¹

theorem **measurable_lmulM** (a : Mat k r) : Measurable (lmulM a : Mt k r → Mt k r)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.HarishChandraParameters

theorem **grpK_const** (k : ℕ) : grpK (fun _ : Fin k => true) = (Matrix.unitaryGroup (Fin k) ℂ : Set (Matrix (Fin k) (Fin k) ℂ))

Development.AbelianVarietyTheory.BlockTheory.JetGauges.HarmonicValueIntegrality

theorem **isIntHarm_of_mem_FFV** (T : CDVR) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN Dw n) T.Fr) (hP : P ∈ FFV (Dw

Development.AbelianVarietyTheory.BlockTheory.JetGauges.HodgeTateCharacterSystems

lemma **unitSec_hom** : unitSec G ≫ G.hom = 𝟙 _

lemma **spec_augMap** : Spec.map (augMap G) ≫ fromLoc G = unitSec G

lemma **spec_strMap** : Spec.map (strMap G) = fromLoc G ≫ G.hom

lemma **strMap_augMap** : strMap G ≫ augMap G = 𝟙 _

lemma **sub_aug_mem** (a : LocRing G) : a − algebraMap O (LocRing G) (aug G a) ∈ augIdeal G

lemma **dAt_add** (a b : LocRing G) : dAt G (a + b) = dAt G a + dAt G b

lemma **dAt_algebraMap** (r : O) : dAt G (algebraMap O (LocRing G) r) = 0

lemma **dAt_mul** (a b : LocRing G) : dAt G (a ∗ b) = aug G a • dAt G b + aug G b • dAt G a

lemma **readFn_tangentPt** {R : Type u} [CommRing R] [Algebra O R] (v : Cotangent G →ₗ[O] R) (s : Γ(G.left, ⊤)) : readFn ((tangentPt v).left.appTop.hom s) = tangentAlg v (G.left.presheaf.germ ⊤ (origin G) trivial s)

lemma **comp_mulN** {T : Over S} (x : T ⟶ G) (N : ℕ) : x ≫ mulN G N = x ^ N

lemma **torsionLift_ι** {N : ℕ} {T : Over S} (x : T ⟶ G) (hx : x ^ N = 1) : torsionLift x hx ≫ torsion.ι G N = x

lemma **torsion_ι_pow** (N : ℕ) : torsion.ι G N ^ N = 1

theorem **IsPadicCompleteSetting.exists_isPrimitiveRoot** {O : ValuationSubring C} {p : ℕ} (h : IsPadicCompleteSetting O p) : ∃ ζ : O, IsPrimitiveRoot ζ p

Development.AbelianVarietyTheory.BlockTheory.JetGauges.IntegralPartDecomposition

theorem **evalAt_tmul** (σ : E →+* C) (a : E) (x : C) : evalAt σ (a ⊗ₜ[ℚ] x) = σ a ∗ x

theorem **ratBasis_apply** (b : Basis ι ℤ (𝒪 E)) (i : ι) : ratBasis b i = ((b i : 𝒪 E) : E)

theorem **integralMap_tmul** (R : Subring C) (a : 𝒪 E) (x : R) : integralMap E R (a ⊗ₜ[ℤ] x) = (a : E) ⊗ₜ[ℚ] (x : C)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.LieMatrixVectors

lemma **Tz_negVec** (k r : ℕ) : Tz k r (negVec k r) = 1

lemma **isFrameFor_negVec** (k r : ℕ) : inWb k r (negVec k r) ∧ 0 < Tz k r (negVec k r) ∧ IsFrameFor k r (negVec k r) 1

lemma **isHCTypeU_trivial** (k : ℕ) : IsHCTypeU k (PI.UpqRep.rhoC (PI.UpqRep.allPos k)) (fun _ => (1 : ℂ →L[ℂ] ℂ))

Development.AbelianVarietyTheory.BlockTheory.JetGauges.MomentCombSplitting

theorem **IntMomCombSplit.intMomSplit** (h : IntMomCombSplit) : IntMomSplit

theorem **ne_zero_of_coprime** {p N₀ : ℕ} [hp : Fact p.Prime] (h : Nat.Coprime N₀ p) : N₀ ≠ 0

theorem **vecOf_nsmul** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {N N' : ℕ} (n : ℕ) (A : Matrix (Fin N') (Fin N) D.host.E) : vecOf D cc δ ((n : D.host.E) • A) = n • vecOf D cc δ A

theorem **goodBasis_smul** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {p : ℕ} [Fact p.Prime] {N N' r N₀ : ℕ} (L : Submodule (𝒪 D.host.E) (Fin N → D.host.E)) (L' : Submodule (𝒪 D.host.E) (Fin N' → D.host.E)) (h₀ : Nat.Coprime N₀ p) (b : Fin r → Matrix (Fin N') (Fin N) D.host.E) (hb : NNBasis D (tor0 D cc δ) δ p L L' b) : GoodBasis D cc δ L L' p N₀ (fun i => (N₀ : D.host.E) • b i)

theorem **goodLift_one** (D : D7) {p : ℕ} [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {N N' : ℕ} (L : Submodule (𝒪 D.host.E) (Fin N → D.host.E)) (L' : Submodule (𝒪 D.host.E) (Fin N' → D.host.E)) (n : ℕ) (A : Matrix (Fin N') (Fin N) D.host.E) : GoodLift D lam cc δ L L' 1 n A ↔ AdmLift D lam cc δ L L' n A

theorem **goodLift_smul** (D : D7) {p : ℕ} [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {N N' : ℕ} (L : Submodule (𝒪 D.host.E) (Fin N → D.host.E)) (L' : Submodule (𝒪 D.host.E) (Fin N' → D.host.E)) (N₀ s n : ℕ) (A : Matrix (Fin N') (Fin N) D.host.E) (h : AdmLift D lam cc δ L L' n A) : GoodLift D lam cc δ L L' N₀ n (((N₀ * s : ℕ) : D.host.E) • A)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.MonomialNorms

theorem **realized_of** (h : ExistsMemberSt) (hs : SupportSt) : ExistsMemberOfCuspData

Development.AbelianVarietyTheory.BlockTheory.JetGauges.NodalSeriesPeriods

theorem **shortCurve_Δ** {K : Type} [Field K] [CharZero K] (a b : K) : (shortCurve a b).Δ = a ^ 3 − 27 * b ^ 2

theorem **curveAt_Δ** {p : ℕ} [Fact p.Prime] (Λ : AddSubgroup ℂ) (h₂ : g₂ Λ ∈ algebraicClosure ℚ ℂ) (h₃ : g₃ Λ ∈ algebraicClosure ℚ ℂ) (lam : Place p) (h : disc Λ = 1) : (curveAt Λ h₂ h₃ lam).Δ = 1

private lemma **nodal_map_sub_eq_X_mul_sub_mul** (u v : A⟦X⟧) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W u − _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W v = X * (u − v) * (C W.a₁ + C W.a₂ * X + C W.a₃ * X ^ 2 * (u + v) + C W.a₄ * X ^ 3 * (u + v) + C W.a₆ * X ^ 5 * (u ^ 2 + u * v + v ^ 2))

private lemma **X_pow_succ_dvd_sub_of_X_pow_dvd** {k : ℕ} {u v : A⟦X⟧} (h : X ^ k ∣ u − v) : X ^ (k + 1) ∣ _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W u − _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W v

private lemma **X_pow_dvd_iterate_sub_iterate** (n m : ℕ) : X ^ n ∣ (_root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W)^[n + m] 0 − (_root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W)^[n] 0

private lemma **vSer_isFixedPt** : _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.vSer W = _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.step W (_root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.vSer W)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer_eq** : _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer W = X ^ 3 + C W.a₁ * X * _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer W + C W.a₂ * X ^ 2 * _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer W + C W.a₃ * _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer W ^ 2 + C W.a₄ * X * _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer W ^ 2 + C W.a₆ * _root_.GlobalTheory.IndefiniteRowSideFullHecke.NodalSeriesPeriods.wSer W ^ 3

theorem **eisSum_eq_G** (L : PeriodPair) {k : ℕ} (hk : k ≠ 0) : eisSum L.lattice.toAddSubgroup k = L.G k

theorem **g₂_eq** (L : PeriodPair) : g₂ L.lattice.toAddSubgroup = L.g₂

theorem **g₃_eq** (L : PeriodPair) : g₃ L.lattice.toAddSubgroup = L.g₃

Development.AbelianVarietyTheory.BlockTheory.JetGauges.OriginChartMultiplication

theorem **mulSeries_zero** {R : Type} [CommRing R] {d : ℕ} (F : FGLaw R d) : mulSeries F 0 = 0

theorem **mulSeries_succ** {R : Type} [CommRing R] {d : ℕ} (F : FGLaw R d) (n : ℕ) : mulSeries F (n + 1) = F.add (mulSeries F n) (pt R d 0)

theorem **mulSeries_one** {R : Type} [CommRing R] {d : ℕ} (F : FGLaw R d) : mulSeries F 1 = pt R d 0

theorem **constantCoeff_mulSeries** {R : Type} [CommRing R] {d : ℕ} (F : FGLaw R d) (n : ℕ) (i : Fin d) : MvPowerSeries.constantCoeff (mulSeries F n i) = 0

Development.AbelianVarietyTheory.BlockTheory.JetGauges.ParabolicPointMembership

theorem **wt_apply** {r : ℕ} (f g : C(Fin r → ℤ_[p], ℂ_[p])) (μ : Meas p r) : wt f μ g = μ (f * g)

theorem **continuous_zc** : Continuous (zc : ℤ_[p] → ℂ_[p])

Development.AbelianVarietyTheory.BlockTheory.JetGauges.PerfectPairings

theorem **mulNat_succ** {R : Type} [CommRing R] {d : ℕ} (G : FGLaw R d) (n : ℕ) : mulNat G (n + 1) = G.add (mulNat G n) (pt R d 0)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.PointDataQ

theorem **charZero_closure** (V : Type) [CommRing V] [IsDomain V] [CharZero V] : CharZero (AlgebraicClosure (FractionRing V))

theorem **colValOf_monomial** (B : FB R d 0) (k : Type) [CommRing k] [Algebra ℚ k] [Algebra R k] (h : PS k 2 d) (w0 : 0) (m : ℕ) {κ : Type} [Fintype κ] (u : κ → Fin d → k) {n : ℕ} (a : κ × Fin n →₀ ℕ) (c : k) (f : Fin n → PS k 1 d) : colValOf B k h w0 m u (MvPolynomial.monomial a c) f = c * ∏ i, B.nabVal k h w0 m (f i) u (MvPolynomial.monomial (colExp a i) 1)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.PullbackGroupAction

theorem **pullGrp_comm** (f : S' ⟶ S) (G : Over S) [GrpObj G] [IsCommMonObj G] : IsCommMonObj (pullGrp f G)

theorem **pullAct_fst** {O : Type} [CommRing O] (f : S' ⟶ S) {G : Over S} [GrpObj G] (A : OAct O G) (a : O) : ((pullAct f A).act a).left ≫ pullFst f G = pullFst f G ≫ (A.act a).left

theorem **eSec_pullFst** (f : S' ⟶ S) (G : Over S) [GrpObj G] : eSec (pullGrp f G) ≫ pullFst f G = (show (𝟙_ (Over S')).left ⟶ (𝟙_ (Over S)).left from f) ≫ eSec G

Development.AbelianVarietyTheory.BlockTheory.JetGauges.RelativeDifferentValues

theorem **finite_of_ne_bot** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) {I : Ideal (𝒪 E)} (hI : I ≠ ⊥) : FiniteMultiplicity P.asIdeal I

theorem **finite_different** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) : FiniteMultiplicity P.asIdeal (differentIdeal ℤ (𝒪 E))

theorem **finite_span** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) {p : ℕ} (hp : p.Prime) : FiniteMultiplicity P.asIdeal (Ideal.span {(p : 𝒪 E)})

theorem **multiplicity_span_pos** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) {p : ℕ} (h : (p : 𝒪 E) ∈ P.asIdeal) : 0 < multiplicity P.asIdeal (Ideal.span {(p : 𝒪 E)})

theorem **idealVal_vp_self** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) {p : ℕ} (h : (p : 𝒪 E) ∈ P.asIdeal) : idealVal_vp p P (Ideal.span {(p : 𝒪 E)}) = 1

theorem **isIntegral_I_mul_sqrt** (m : ℕ) : IsIntegral ℚ (Complex.I * (Real.sqrt m : ℂ))

theorem **isHostField_hostField** (e q₁ d : ℕ) : IsHostField (hostField e q₁ d) e q₁ d

theorem **hostNumbers_example** : HostNumbers 2 11 7

Development.AbelianVarietyTheory.BlockTheory.JetGauges.RigidNormaliserGenerators

theorem **cmp_snd** : cmp G F ≫ (snd (TF (R

Development.AbelianVarietyTheory.BlockTheory.JetGauges.RootValueClassBasis

theorem **Psi_one** (rk : (E ≃ₐ[ℚ] E) → ℕ) : Psi rk (1 : E) = 1

theorem **Psi_mul** (rk : (E ≃ₐ[ℚ] E) → ℕ) (x y : E) : Psi rk (x * y) = Psi rk x * Psi rk y

theorem **Psi_ne_zero** (rk : (E ≃ₐ[ℚ] E) → ℕ) {x : E} (hx : x ≠ 0) : Psi rk x ≠ 0

theorem **ClassBasis.h_pos** (B : ClassBasis E) (i : Fin B.k) : 0 < B.h i

theorem **Roots.ψ_ne_zero** (R : Roots rk B Ω) (i : Fin B.k) : R.ψ i ≠ 0

theorem **Roots.IsValue.mem** {R : Roots rk B Ω} {π : (Ideal (𝒪 E))⁰} {b : Ω} (h : R.IsValue π b) : b ∈ R.field

theorem **Roots.IsValue.ne_zero** {R : Roots rk B Ω} {π : (Ideal (𝒪 E))⁰} {b : Ω} (h : R.IsValue π b) : b ≠ 0

theorem **Roots.isValue_one** (R : Roots rk B Ω) : R.IsValue 1 1

private lemma **prod_mulSingle_zpow_eq_symm** {ι : Type} [Fintype ι] [DecidableEq ι] {G : Type} [CommGroup G] {n : ι → ℕ} (e : G ≃* ((i : ι) → Multiplicative (ZMod (n i)))) (b : ι → ℤ) : ∏ j, e.symm (Pi.mulSingle j (Multiplicative.ofAdd 1)) ^ b j = e.symm (fun j => Multiplicative.ofAdd (b j : ZMod (n j)))

theorem **exists_basis** (G : Type) [CommGroup G] [Finite G] : ∃ (k : ℕ) (g : Fin k → G), (∀ x, ∃ a : Fin k → ℤ, x = ∏ i, g i ^ a i) ∧ ∀ a : Fin k → ℤ, ∏ i, g i ^ a i = 1 → ∀ i, g i ^ a i = 1

theorem **exists_gen** {I : (FractionalIdeal (𝒪 E)⁰ E)ˣ} (h : ClassGroup.mk E I = 1) : ∃ γ : Eˣ, I = toPrincipalIdeal (𝒪 E) E γ

theorem **Roots.exists_isValue** (R : Roots rk B Ω) (π : (Ideal (𝒪 E))⁰) : ∃ b, R.IsValue π b

Development.AbelianVarietyTheory.BlockTheory.JetGauges.SemiAbelianAxisTangent

theorem **vecEquiv_mul** {X : Scheme.{0}} {d : ℕ} {T : Over X} (x y : T ⟶ vecObj X d) : vecEquiv T (x * y) = vecEquiv T x * vecEquiv T y

theorem **vecEquiv_one** {X : Scheme.{0}} {d : ℕ} {T : Over X} : vecEquiv T (1 : T ⟶ vecObj X d) = 1

theorem **homAt_vecPt** {A B : Type} [CommRing A] [Algebra S A] [CommRing B] [Algebra S B] (f : A →ₐ[S] B) (v : Fin d → A) : homAt x0 f ≫ vecPt x0 d v = vecPt x0 d fun j => f (v j)

theorem **vecPt_add** {A : Type} [CommRing A] [Algebra S A] (v w : Fin d → A) : vecPt x0 d (v + w) = vecPt x0 d v * vecPt x0 d w

theorem **vecPt_zero** {A : Type} [CommRing A] [Algebra S A] : vecPt x0 d (0 : Fin d → A) = 1

theorem **vecPt_injective** {A : Type} [CommRing A] [Algebra S A] : Function.Injective (vecPt x0 d : (Fin d → A) → _)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.SideNormalisedGap

theorem **measurable_mulP** {k r : ℕ} : Measurable (mulP (k

Development.AbelianVarietyTheory.BlockTheory.JetGauges.SmoothDecompositionForms

theorem **flat_dotProduct** (A : Matrix ι ι ℂ) (y a : ι → ℂ) : flat A y ⬝ᵥ a = herm A a y

theorem **tensForm_pure** (A : Matrix ι ι ℂ) {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] (a b : ι → ℂ) (v w : H) : tensForm A (pureT a v) (pureT b w) = herm A a b * ⟪w, v⟫_ℂ

theorem **contr_pure** (A : Matrix ι ι ℂ) (ω a : ι → ℂ) {E : Type} [AddCommGroup E] [Module ℂ E] (v : E) : contr A ω (pureT a v) = herm A a ω • v

theorem **tensAct_pure** (ρ : Mx k r → Matrix ι ι ℂ) (R : HRep k r) (g : Mx k r) (a : ι → ℂ) (v : R.H) : tensAct ρ R g (pureT a v) = pureT (ρ g *ᵥ a) (R.σ g v)

theorem **herm_dualGram_flat** [DecidableEq ι] {A : Matrix ι ι ℂ} (hA : Aᴴ = A) (h2 : A * A = 1) (y y' : ι → ℂ) : herm (dualGram A) (flat A y) (flat A y') = herm A y' y

theorem **norm_det_of_mem** {g : Mx k r} (hg : g ∈ UH (Hpk k r)) : ‖g.det‖ = 1

theorem **isFormRep_std** (k r : ℕ) : IsFormRep k r (stdRep k r) (Hpk k r) 1 where

theorem **dsigW_zero** (R : HRep k r) (w : List (Mx k r)) : dsigW (k

theorem **zero_mem_smooth** (R : HRep k r) : (0 : R.H) ∈ R.smooth

theorem **isSignedDecomp_trivial** [DecidableEq ι] (R : HRep k r) (ep : ι → ℝ) (hep : ∀ c, ep c = 1 ∨ ep c = -1) : IsSignedDecomp (fun _ => (1 : Matrix ι ι ℂ)) (Matrix.diagonal fun c => (ep c : ℂ)) R (fun _ : ι => R) (fun j v => Pi.single j v) (fun j x => x j) ep

Development.AbelianVarietyTheory.BlockTheory.JetGauges.SpansOverField

private theorem **bar_bar_jet_gauges** (a : 𝒪 E) : bar E (bar E a) = a

private theorem **eq_sum_of_support_subset** (s : X →ₗ[𝒪 E] X →₀ 𝒪 E) (hs : Finsupp.linearCombination (𝒪 E) id ∘ₗ s = LinearMap.id) (S : Finset X) (x : X) (hx : (s x).support ⊆ S) : x = ∑ m ∈ S, s x m • m

private theorem **sum_B_eq_sum_bar_outer** (B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) (hB : IsBalanced E (fun a c => B a c)) (s : X →ₗ[𝒪 E] X →₀ 𝒪 E) (hs : Finsupp.linearCombination (𝒪 E) id ∘ₗ s = LinearMap.id) (S : Finset X) {n : ℕ} (x x' : Fin n → X) (hx : ∀ i, (s (x i)).support ⊆ S) (hx' : ∀ i, (s (x' i)).support ⊆ S) : ∑ i, B (x i) (x' i) = ∑ m ∈ S, ∑ p ∈ S, B m (bar E (outer E x x' ((Finsupp.lapply m).comp s) ((Finsupp.lapply p).comp s)) • p)

theorem **pairing_eq_of_outer_eq** [Module.Finite (𝒪 E) X] [Module.Projective (𝒪 E) X] (B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) (hB : IsBalanced E (fun a c => B a c)) {n m : ℕ} (x x' : Fin n → X) (u u' : Fin m → X) (h : outer E x x' = outer E u u') : pairing (fun a c => B a c) x x' = pairing (fun a c => B a c) u u'

Development.AbelianVarietyTheory.BlockTheory.JetGauges.SplitUnramifiedPlaces

@[simp] theorem **bar_tmul** [StarRing OE] (z : ℤ_[p]) (a : OE) : bar p OE (z ⊗ₜ a) = z ⊗ₜ star a

theorem **bar_bar** [StarRing OE] (x : Op p OE) : bar p OE (bar p OE x) = x

@[simp] theorem **extend_tmul** {O : Type} [CommRing O] [Algebra ℤ_[p] O] (σ : OE →+* O) (z : ℤ_[p]) (a : OE) : extend σ (z ⊗ₜ a : Op p OE) = algebraMap ℤ_[p] O z * σ a

theorem **conj_conj** [StarRing OE] (w : Place p OE) : w.conj.conj = w

theorem **mul_eq_zero_of_ne** {w u : Place p OE} (h : w ≠ u) : w.e * u.e = 0

theorem **conj_induces_iff** [StarRing OE] (w : Place p OE) {O : Type} [CommRing O] [Algebra ℤ_[p] O] (σ : OE →+* O) : w.conj.Induces σ ↔ w.Induces (σ.comp (starRingEnd OE))

theorem **gaussEmb_sqrtd** (s : ℤ_[p]) (hs : s * s = -1) : gaussEmb s hs Zsqrtd.sqrtd = s

theorem **extend_gaussIdem** (s h : ℤ_[p]) (hs : s * s = -1) (hh : 2 * h = 1) : extend (gaussEmb s hs) (gaussIdem s h) = 1

theorem **extend_bar_gaussIdem** (s h : ℤ_[p]) (hs : s * s = -1) : extend (gaussEmb s hs) (bar p GaussianInt (gaussIdem s h)) = 0

theorem **mul_gaussIdem** (s h : ℤ_[p]) (hs : s * s = -1) (x : Op p GaussianInt) : x * gaussIdem s h = algebraMap ℤ_[p] _ (extend (gaussEmb s hs) x) * gaussIdem s h

theorem **exists_sqrt_neg_one_five** : ∃ s h : ℤ_[5], s * s = -1 ∧ 2 * h = 1

theorem **nonempty_splitPlace_five** : Nonempty (SplitPlace 5 GaussianInt)

theorem **nonempty_unrPlace_five** : Nonempty (UnrPlace 5 GaussianInt)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.ThetaSpanAdjoint

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryAdjointCongruence.pushLet_letB** {n : ℕ} (γ : Matrix (Fin n) (Fin n) E) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) (b : Fin n → E) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitaryAdjointCongruence.pushLet c γ (_root_.Formalism.L2.Pj.letB c σ i b) = _root_.Formalism.L2.Pj.letB c σ i (γ.mulVec b)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.ThetaWeightedNorm

theorem **be_weight** : 2 • ((I.i * I.g0h) • I.de2) + I.n1 • (2 • I.L - I.de2) = (2 * I.n1) • I.L

theorem **constant_lt_t_mul_a** : I.c1 < I.t * I.a1

Development.AbelianVarietyTheory.BlockTheory.JetGauges.TraceFormDualSystem

theorem **bar_bar** (a : 𝒪 E) : bar E (bar E a) = a

theorem **Exponent.smul_right** (l : Exponent E Y) (a : 𝒪 E) (y y' : Module.Dual (𝒪 E) Y) : l.form y (a • y') = bar E a * l.form y y'

theorem **nonempty_dualSystem** [Module.Finite (𝒪 E) Y] [Module.Projective (𝒪 E) Y] : Nonempty (DualSystem E Y)

theorem **DualSystem.apply_eq** (D : DualSystem E Y) (β : balancedForms E Y) (x x' : Y) : β.1 x x' = ∑ j, ∑ k, β.1 ((D.y j x * bar E (D.y k x')) • D.e j) (D.e k)

theorem **DualSystem.pairing_of_sum** (D : DualSystem E Y) (l : Exponent E Y) (β : balancedForms E Y) {ι : Type} [Fintype ι] (x x' : ι → Y) (h : ∀ y y', l.form y y' = ∑ i, y (x i) * bar E (y' (x' i))) : D.pairing l β = (∑ i, β.1 (x i) (x' i)) / 2

theorem **DualSystem.dual_expand** (D : DualSystem E Y) (y : Module.Dual (𝒪 E) Y) : y = ∑ j, y (D.e j) • D.y j

theorem **DualSystem.form_eq** (D : DualSystem E Y) (l : Exponent E Y) (y y' : Module.Dual (𝒪 E) Y) : l.form y y' = ∑ p : Fin D.n × Fin D.n, y (l.form (D.y p.1) (D.y p.2) • D.e p.1) * bar E (y' (D.e p.2))

theorem **DualSystem.pairing_eq** (D D' : DualSystem E Y) (l : Exponent E Y) (β : balancedForms E Y) : D.pairing l β = D'.pairing l β

theorem **pairing_eq_dualSystem** (D : DualSystem E Y) (l : Exponent E Y) (β : balancedForms E Y) : pairing l β = D.pairing l β

theorem **pairing_of_sum** (l : Exponent E Y) (β : balancedForms E Y) {ι : Type} [Fintype ι] (x x' : ι → Y) (h : ∀ y y', l.form y y' = ∑ i, y (x i) * bar E (y' (x' i))) : pairing l β = (∑ i, β.1 (x i) (x' i)) / 2

theorem **IsPhiC.phiC_eq** {β : balancedForms E Y} (hβ : IsPosSemidef β) {r : ℝ} (h : IsPhiC β r) : phiC β hβ = r

theorem **pairing_ofFamily** (β : balancedForms E Y) {ι : Type} [Fintype ι] (x : ι → Y) (hx : ∀ y : Module.Dual (𝒪 E) Y, (∀ i, y (x i) = 0) → y = 0) : pairing (Exponent.ofFamily x hx) β = ∑ i, quad β (x i)

theorem **pairingValues_nonempty** (β : balancedForms E Y) : (pairingValues β).Nonempty

theorem **isPhiC_zero** : IsPhiC (0 : balancedForms E Y) 0

theorem **exists_sum_of_add_bar_eq_one** (w : 𝒪 E) (hw : w + bar E w = 1) (l : Exponent E Y) : ∃ (n : ℕ) (u v : Fin n × Fin n → Y), (∀ y y', l.form y y' = ∑ p, (y (u p) * bar E (y' (v p)) + y (v p) * bar E (y' (u p)))) ∧ ∀ β : balancedForms E Y, pairing l β = ∑ p, (quad β (u p + v p) - quad β (u p) - quad β (v p))

theorem **traceForm_one_one** : (traceForm E).1 1 1 = Module.finrank ℚ E

theorem **traceForm_posSemidef** : IsPosSemidef (traceForm E)

theorem **half_finrank_mem_pairingValues** : ((Module.finrank ℚ E : ℝ) / 2) ∈ pairingValues (traceForm E)

Development.AbelianVarietyTheory.BlockTheory.JetGauges.UniformizationClosedSplit

theorem **unifAt_of** (h₁ : ModelAt Xo c) (h₂ : UnifOfModel Xo c) : UnifAt Xo c

theorem **PrimeToLevel.not_mem_M** {la : HeightOneSpectrum (𝒪 c.F)} (h : PrimeToLevel c la) : ((l.M : ℕ) : 𝒪 c.F) ∉ la.asIdeal

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.AdmissibleCharUnramified

theorem **twCoef_chi_one** (g : G) (m : O ⊗[O] X) : twCoef (Γ.Wchi 1) Γ.χ₁ X g m = m

theorem **cocycle_add** (z : ₁(Z1 (twCoef (Γ.Wchi 1) Γ.χ₁ X) ⊤)) (g h : G) : (z : G → O ⊗[O] X) (g * h) = (z : G → O ⊗[O] X) g + (z : G → O ⊗[O] X) h

theorem **homModInertia_eq_unr** (I : Subgroup G) : homModInertia Γ X I = (unr (twCoef (Γ.Wchi 1) Γ.χ₁ X) I : Set _)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.AdmissibleCharacterHyp

theorem **twCoef_Wchi** {G : Type} [Group G] [TopologicalSpace G] {Qp LL O : Type} [Field Qp] [Field LL] [Algebra Qp LL] [CommRing O] [Algebra O LL] [TopologicalSpace LL] (Γ : GalData G Qp LL O) (c : ℕ) (X : Type) [AddCommGroup X] [Module O X] (g : G) (y : O ⊗[O] X) : twCoef (Γ.Wchi c) Γ.χ₁ X g y = (((Γ.ε g) ^ (-((c : ℤ) - 1)) : Oˣ) : O) • y

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.AdmissibleSignatures

theorem **add_phi_mem_Y** {y' m : Vec K a ι} (hy : y' ∈ D.Y') (hm : m ∈ D.Minv) : y' + D.phi m ∈ D.Y

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.BaseChangeInertUnramified

theorem **atRecord_iff_guarded** (D : D7) (h : ∀ j, 1 ≤ j → j ≤ D.R.m → Standing D j) : AtRecord D ↔ AtRecordGuarded D

theorem **ubc_map_inv** (n : ℕ) (α : Multiset ℂ) : (ubc n α).map (·⁻¹) = ubc n α

theorem **card_ubc** (n : ℕ) (α : Multiset ℂ) (h : Multiset.card α = n / 2) : Multiset.card (ubc n α) = n

theorem **isBCV_of_isBaseChange** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} {A : NAut N} (B : NBC A) (i : N.Loc) (Q : A.AV i) (P : List (Part A.CG)) (h : IsBaseChange B (N.j i) (B.embAV i Q) P) : IsBCV N A i Q P

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.BaseSizes

theorem **zetaP_eq** (D : D7) (i : D.nd.Loc) (P : HeightOneSpectrum (𝒪 D.host.E)) (hP : P ∈ D.nd.Q i) : zetaP D i P = D.nd.zeta i P

theorem **lStar_pos** {D : D7} (L : SizeData₂ D) : 0 < LevelStarExponents.lStar L

theorem **vO_l** (D : D7) (i : D.nd.Loc) : vO (D.nd.O i) (D.nd.l i) (D.nd.l i : D.nd.O i) = 1

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.BlockCharactersCrystalline

theorem **chiRatio_eq** (D : D7) (i : D.nd.Loc) (c : ℕ) (h1 : 1 ≤ c) (h2 : c ≤ D.R.t) : chiRatio D i c = (epsU D.nd i) ^ (-((c : ℤ) - 1))

theorem **chiRatO_apply** (D : D7) (i : D.nd.Loc) (c : ℕ) (g : GE D.nd) : chiRatO D.nd i c g 0 0 = ((chiRatio D i c g : (D.nd.O i)ˣ) : D.nd.O i)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.BlockComplementLattice

theorem **blockRep_apply** {a : ℕ} (ρ : Γ →* GL (Fin a) O) (χ : ℕ → (Γ →* Oˣ)) (t : ℕ) (g : Γ) (v : (Fin a → FractionRing O) × (Fin t → FractionRing O)) : blockRep ρ χ t g v = ((((ρ g : GL (Fin a) O) : Matrix (Fin a) (Fin a) O).map (algebraMap O (FractionRing O))).mulVec v.1, fun c : Fin t => algebraMap O (FractionRing O) ((χ (c.1 + 1) g : Oˣ) : O) * v.2 c)

theorem **IsoBlock.rank_eq** {n a t : ℕ} {r : Γ →* GL (Fin n) O} {ρ : Γ →* GL (Fin a) O} {χ : ℕ → (Γ →* Oˣ)} (h : IsoBlock r ρ χ t) : n = a + t

theorem **Complement_at.family** {D : D7} (h : Complement_at D) (i : D.nd.Loc) (r : D.nd.X i → (G D.host.E D.nd.Om →* GL (Fin D.R.nV) (D.nd.O i))) (hr : GoodR D.nd i r) : ∃ ρ, IsComplementFamily D.nd i r ρ

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.BlockHodgeTateJumps

theorem **count_blockJumps** (t : ℕ) (a : ℤ) : (blockJumps t).count a = if 0 ≤ a ∧ a < (t : ℤ) then 1 else 0

theorem **BlockJumpsAt.mult** {π : N.X i} {w : HeightOneSpectrum (𝒪 H.E)} {hw : w ∈ placesOver H.E (N.l i)} {τ : Br.Kw nbr i w hw →ₐ[ℚ_[N.l i]] Br.LL nbr i w hw} (h : BlockJumpsAt i r ρ L nbr π w hw τ) (a : ℤ) : multV (Br.PD L nbr i w hw) (matRep (Br.loc nbr i w hw (rTwO N i r π))) τ a = multV (Br.PD L nbr i w hw) (matRep (Br.loc nbr i w hw (WpO N i ρ π))) τ a + if 0 ≤ a ∧ a < (R.t : ℤ) then 1 else 0

theorem **closed_of_all** (h : ∀ D : D7, BlockJumps_at D) (𝒞 : FamClass) : BlockJumpsClosed 𝒞

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.BlockJumpCount

theorem **IsJumpMultiset.unique** {ρ : Representation LL G V} {τ : K →ₐ[Qp] LL} {J J' : Multiset ℤ} (h : IsJumpMultiset PD ρ τ J) (h' : IsJumpMultiset PD ρ τ J') : J = J'

theorem **count_blockJumps** (t : ℕ) (a : ℤ) : (blockJumps t).count a = if 0 ≤ a ∧ a < t then 1 else 0

theorem **blockJumps_at.mult** {D : D7} (h : blockJumps_at D) (i : D.nd.Loc) (r : D.nd.X i → (GE D.nd →* GL (Fin D.R.nV) (D.nd.O i))) (ρ : D.nd.X i → (GE D.nd →* GL (Fin D.R.r) (D.nd.O i))) (π : D.nd.X i) (hπ : π ∈ D.nd.Z i) (hr : IsHeckeGalois D.nd i π (r π)) (hs : IsBlockSum D.nd i r ρ π) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : w ∈ placesOver D.host.E (D.nd.l i)) (τ : Br.Kw D.nbr i w hw →ₐ[ℚ_[D.nd.l i]] Br.LL D.nbr i w hw) (a : ℤ) : multV (Br.PD D.loc D.nbr i w hw) (matRep (Br.loc D.nbr i w hw (rTwO D.nd i r π))) τ a = multV (Br.PD D.loc D.nbr i w hw) (matRep (Br.loc D.nbr i w hw (WpO D.nd i ρ π))) τ a + (if 0 ≤ a ∧ a < D.R.t then 1 else 0)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.CMTypeEmbeddings

theorem **card_hcSigma** {R : Ranks} (P : HCParam R) : Multiset.card (hcSigma P) = R.r

theorem **hcSigma_int** {R : Ranks} (P : HCParam R) : ∀ z ∈ hcSigma P, ∃ n : ℤ, z = (n : ℂ)

theorem **IsCuspParam.unr** {D : D7} {S : GlobalTheory.HeightCounting.RowSide D} {σ : D.what₂.FT} {π : D.naut.CG D.R.r} (h : IsCuspParam D S σ π) (w : PlE D.host) (hw : ∀ k : Fin 2, w ∉ placesOver D.host.E (D.host.g k)) : D.naut.unr (D.naut.locG π w)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.CrystallineExtensions

theorem **IsExt.exists_cocycle** {ρ : Representation LL G V} {ρ' : Representation LL G W} {i : V →ₗ[LL] W} {p : W →ₗ[LL] LL} (h : IsExt ρ ρ' i p) : ∃ c : G → V, IsCocycleOf ρ' i p c

theorem **IsExt.cocycle_mul** {ρ : Representation LL G V} {ρ' : Representation LL G W} {i : V →ₗ[LL] W} {p : W →ₗ[LL] LL} (h : IsExt ρ ρ' i p) {c : G → V} (hc : IsCocycleOf ρ' i p c) (g k : G) : c (g * k) = c g + ρ g (c k)

theorem **IsExt.cocycle_sub_mem_coboundaries** {ρ : Representation LL G V} {ρ' : Representation LL G W} {i : V →ₗ[LL] W} {p : W →ₗ[LL] LL} (h : IsExt ρ ρ' i p) {c c' : G → V} (hc : IsCocycleOf ρ' i p c) (hc' : IsCocycleOf ρ' i p c') : c - c' ∈ B1 ρ

theorem **extRep_isExt** (ρ : Representation LL G V) (c : ↥(continuousCocycles ρ)) : IsExt ρ (extRep ρ c) (LinearMap.inl LL V LL) (LinearMap.snd LL V LL) ∧ IsCocycleOf (extRep ρ c) (LinearMap.inl LL V LL) (LinearMap.snd LL V LL) (c : G → V)

theorem **mk_mem_H1fSub** {V : Type} [AddCommGroup V] [Module LL V] [Module Qp V] [IsScalarTower Qp LL V] [TopologicalSpace V] [ContinuousAdd V] [ContinuousConstSMul LL V] (ρ : Representation LL G V) (c : ↥(continuousCocycles ρ)) : (Submodule.Quotient.mk c : H1V ρ) ∈ H1fSubmodule PD ρ ↔ (c : G → V) ∈ finiteCocycles PD ρ

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.CrystallineSubrepresentation

theorem **mem_Dcris** (ρ : Representation LL G V) (x : V ⊗[Qp] B) : x ∈ Dcris PD ρ ↔ ∀ g, actVB PD ρ g x = x

theorem **actVB_mulVB** (ρ : Representation LL G V) (g : G) (c : K₀) (x : V ⊗[Qp] B) : actVB PD ρ g (mulVB (K₀

theorem **comm_smul** (c : K₀) (x : B ⊗[Qp] V) : TensorProduct.comm Qp B V (c • x) = mulVB (K₀

theorem **comparison_tmul** (ρ : Representation LL G V) (b : B) (d : ↥(DcrisK₀ PD ρ)) : comparison PD ρ (b ⊗ₜ[K₀] d) = b • (d : B ⊗[Qp] V)

theorem **tensorB_mem** {ρ : Representation LL G V} {ρ' : Representation LL G V'} {f : V →ₗ[LL] V'} (hf : IsEquivariant ρ ρ' f) (x : V ⊗[Qp] B) (hx : x ∈ Dcris PD ρ) : tensorB (Qp

theorem **prodB_tmul** (v : V) (b : B) (v' : V') (b' : B) : prodB (LL

theorem **invariantsEq** : InvariantsEq PD

theorem **fontaineFormalism_iff** : FontaineFormalism l P ↔ ∀ K LL : FinExt l, FormalismFor (P.PD K) ↥LL.1

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.DefiniteFormInstances

theorem **ClosedAbsoluteShare.tied** {𝒞 : FamClass} (h : ClosedAbsoluteShare 𝒞) : ClosedFamilySpectrumCondition 𝒞

theorem **isBaseChange_embAV** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} {A : NAut N} (B : NBC A) (i : N.Loc) (P : A.AV i) (Pa : List (Part A.CG)) : IsBaseChange B (N.j i) (B.embAV i P) Pa ↔ Formalism.MainNotions.N.IsBCV N A i P Pa

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.DeterminantInTorus

theorem **isRx_iff** (D : D7) (i : D.nd.Loc) (r : D.nd.X i → (GE D.nd →* GL (Fin D.R.nV) (D.nd.O i))) : (∀ x, IsRx D i x (r x)) ↔ GoodR D.nd i r

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.DiagonalQuotientEigen

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.DiagonalQuotientEigen.qN_zero** (i : ZMod n) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.DiagonalQuotientEigen.qN W T i 0 = _root_.GlobalTheory.IndefiniteRowSideFullHecke.DiagonalQuotientEigen.NΔl W T i

theorem **IsNab.unique** {g : W.Gam} {a : LetA O n} {i : ZMod n} {j : ℤ} (h : Function.Injective fun x : W.DD T => (a * W.π₀) • x) {nb nb' : ↥(qN W T i j) →ₗ[O] ↥(qN W T i j)} (h₁ : IsNab W T g a i j nb) (h₂ : IsNab W T g a i j nb') : nb = nb'

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.GammaStableInverse

theorem **invSub_le** {a : A} {M M' : Submodule A D} (h : M ≤ invSub a M') : invSub a M ≤ invSub a M'

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.GoodPairSystems

theorem **samePair_conj** (w : D.what₂.places.Pl) : SamePair D w (D.what₂.places.conj w)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GoodPairSystems.twisted_conj** (w : D.what₂.places.Pl) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.GoodPairSystems.Twisted D (D.what₂.places.conj w) ↔ _root_.GlobalTheory.IndefiniteRowSideFullHecke.GoodPairSystems.Twisted D w

theorem **TwistedPlaceSettings.inW1** {twI : D.what₂.places.Pl → D.ntw.Tw} (h : TwistedPlaceSettings D twI) {w : D.what₂.places.Pl} (hw : Twisted D w) : D.ntw.NonzeroIndexPlace (twI w)

theorem **TwGoodAt.goodTw** {c : NChoice D} {xd : ThData D} {N0 : D.ntw.Tw → ℕ} (h : TwGoodAt c xd N0) {w : D.what₂.places.Pl} (hw : Twisted D w) : ∃ S, IntegralHeckeForms.GoodTw D.nexp (BlochKatoReduction.bk0 D) (expDat xd) N0 (c.twI w) (((D.ntw.l (c.twI w) : ℝ) ^ 2)⁻¹) S ∅ (c.Q w) (c.k' w) (BlochKatoReduction.Gen.psiAt (c.sd w) (c.psi' w))

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.GradedSeries

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedSeries.serMul_finite** (f : PowerSeries L) {u : ℤ → M} (hu : _root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedSeries.IsLaurent u) (k : ℤ) : (Function.support fun j : ℕ => PowerSeries.coeff j f • u (k - (j : ℤ))).Finite

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedSeries.serMul_one** (u : ℤ → M) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedSeries.serMul (1 : PowerSeries L) u = u

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedSeries.levN_mem_Nlev** (i : ZMod n) (ζ : LL) (x : ↥(W.N T)) : W.levN T i ζ x ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.GradedSeries.Nlev W T i ζ

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.ImagePolynomialAction

theorem **X_mul_polyR₁** (l e : ℕ) (U : Finset ℤ) (η : ℕ) : Polynomial.X * polyR₁ l e U η = polyR l e U η - Polynomial.C ((polyR l e U η).eval 0)

theorem **mem_shifts** {U : Finset ℤ} {η : ℕ} {n : ℤ} : n ∈ shifts U η ↔ ∃ i ∈ U, ∃ k : ℤ, (1 ≤ k ∧ k ≤ η) ∧ i + k = n

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.ImageStableDegreeBounds

theorem **mem_SHigh** {f : S O K e} : f ∈ SHigh O K l e ↔ (X : K⟦X⟧) ^ (l * e) ∣ (f : K⟦X⟧)

theorem **mem_mulL** {P : PhiMod O K l e a ι} {f : K⟦X⟧} {x : Vec K a ι} : x ∈ mulL P f ↔ ∃ y ∈ P.L, f • y = x

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.InvariantsVanishing

theorem **invariants_eq_bot_iff** : (resRep Γ).invariants = ⊥ ↔ ∀ x, (∀ g, resRep Γ g x = x) → x = 0

theorem **letters_ne_zero** {l : ℕ} [Fact l.Prime] {P : LocPrime l} {K LL : FinExt l} {n : ℕ} {emb : ZMod n → (ɪ(K0 K) →ₐ[ℚ_[l]] ɪLL.1)} {lam : ɪ(OL LL)} (h : Standing P K LL emb lam) : n ≠ 0

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.KernelCharacterDelta

theorem **binomPow_natCast** (k : Type) [Field k] (l : ℕ) [Fact l.Prime] [CharP k l] (m : ℕ) : binomPow k l (m : ℤ_[l]) = (1 + PowerSeries.X) ^ m

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.KernelDictionary

theorem **ne_two_of_good** {K LL : FinExt l} (h : Good K 1 LL) : l ≠ 2

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.LettersFreeOfRank

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.LetterBasisDecomposition.letter_smul** (i : ZMod n) (c : PowerSeries O) (x : ɪ(W.Nl T i)) : ((c • x : ɪ(W.Nl T i)) : W.DD T) = (Pi.single i c : _root_.Formalism.BasicNotions.LetA O n) • (x : W.DD T)

theorem **phi_letE** (i : ZMod n) : W.φA (letE (i - 1)) = letE i

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.LocalRingStructure

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.LocalRingStructure.letters_neZero** {l : ℕ} [Fact l.Prime] (K : _root_.Formalism.BasicNotions.Loc.FinExt l) : NeZero (Module.finrank ℚ_[l] ɪ(_root_.Formalism.BasicNotions.Loc.K0 K))

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.OnePlusPowerAssembly

theorem **LemmaPsi.modPG** (h : LemmaPsi) (l : ℕ) [Fact l.Prime] (k : Type) [Field k] [CharP k l] (hl : 5 ≤ l) (R : RingPG k l) (ι : k⦇X⦈ →ₐ[k] k⦇X⦈) (hι : IsCyclotomic R ι) : ∃ ψ : k⦇X⦈ →ₗ[k] k⦇X⦈, IsPsi l ι ψ ∧ PsiRing R ψ ∧ ∀ (n : ℕ) (V : Type) [AddCommGroup V] [Module k⦇X⦈ V] [Module k⟦X⟧ V] [Module k V] [IsScalarTower k⟦X⟧ k⦇X⦈ V] [IsScalarTower k k⦇X⦈ V] [IsScalarTower k k⟦X⟧ V] [FiniteDimensional k⦇X⦈ V] (D : ModPG R n V) (i : ZMod n), ∃ Ψ : V →ₗ[k] V, PsiMod l ψ (D.φ i) (D.γ (i - 1)) (D.γ i) Ψ

theorem **phiPi_vv** (k : Type) [Field k] (l : ℕ) [Fact l.Prime] : phiPi k l (vv k) = vv k ^ l

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.PlaceBelowAllPowers

theorem **samePlace_refl** (i : X.Tw) (b : Bool) : SamePlace i b i b

theorem **isReading_zpow** (i : X.Tw) (k : ℕ) (ψ : G D.host.E N.Om →* (X.Ok i k)ˣ) (hψ : ∀ g, ψ g ∈ Subgroup.zpowers (X.zk i k)) (a : ℤ) (ha : IsCoprime a (X.l i : ℤ)) : IsReading i i k ψ (ψ ^ a)

theorem **AllPowersGood.self_inv** {N0 : X.Tw → ℕ} {i : X.Tw} {eps' : ℝ} {S : ∀ (j : ℕ) (b : Bool), Finset (Y.Dat i j b)} {A Q : Finset (PlE D.host)} {k' : ℕ} {ψ' : X.RCG i Q →* (X.Ok i k')ˣ} (h : AllPowersGood Y bkP E N0 i eps' S A Q k' ψ') : GoodTw Y bkP E N0 i eps' S A Q k' ψ' ∧ GoodTw Y bkP E N0 i eps' S A Q k' ψ'⁻¹

theorem **AllConjugatesGood.allPowers** {D : Formalism.BasicNotions.D7} {bkP : Formalism.MainNotions.BlochKatoReduction.BK D} {E : XD D} {N0 : D.ntw.Tw → ℕ} (h : AllConjugatesGood D bkP E N0) : AllPowers D bkP E N0

theorem **AllPowers.aux₃** {D : Formalism.BasicNotions.D7} {bkP : Formalism.MainNotions.BlochKatoReduction.BK D} {E : XD D} {N0 : D.ntw.Tw → ℕ} (h : AllPowers D bkP E N0) : Formalism.MainNotions.IntegralHeckeForms.aux₃ D bkP E N0

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.PointRingCommRing

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.PointRingCommRing.letters_neZero** {l : ℕ} [Fact l.Prime] (K : _root_.Formalism.BasicNotions.Loc.FinExt l) : NeZero (Module.finrank ℚ_[l] ɪ(_root_.Formalism.BasicNotions.Loc.K0 K))

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.PrimeTypeRows

theorem **twGL_val** (r : Γ →* GL (Fin n) O) (χ : Γ →* Oˣ) (g : Γ) : ((twGL r χ g : GL (Fin n) O) : Matrix (Fin n) (Fin n) O) = ((χ g : Oˣ) : O) • (r g : Matrix (Fin n) (Fin n) O)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.RationalRayLevels

theorem **auxiliarySplitSquarefreeLevel_one** (M : ℕ) : IsAuxiliarySplitSquarefreeLevel D M 1

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.ReadingAllLemmas

theorem **GoodAll.goodTw** {N0 : X.Tw → ℕ} {i : X.Tw} {eps' : ℝ} {S : ∀ (j : ℕ) (b : Bool), Finset (Y.Dat i j b)} {A Q : Finset (PlE D.host)} {k' : ℕ} {ψ' : X.RCG i Q →* (X.Ok i k')ˣ} (h : GoodAll Y bkP E N0 i eps' S A Q k' ψ') : IntegralHeckeForms.GoodTw Y bkP E N0 i eps' S A Q k' ψ'

theorem **LemmaAll.aux₃** {D : Formalism.BasicNotions.D7} {bkP : BlochKatoReduction.BK D} {E : IntegralHeckeForms.XD D} {N0 : D.ntw.Tw → ℕ} (h : LemmaAll D bkP E N0) : IntegralHeckeForms.aux₃ D bkP E N0

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.RepresentationBounds

theorem **killsCoker_M_iff** {P : PhiMod O K l e a ι} {Q : PhiMod O K l e a κ} (F : Hom P Q) (n : ℕ) : KillsCoker F.app n P.M Q.M ↔ ∀ q : F.coker, n • q = 0

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.RepresentationOverSumChar

theorem **charpoly_blockSum** {r : ℕ} (t : ℕ) (ρ : Γ →* GL (Fin r) O) (χ : ℕ → (Γ →* Oˣ)) (g : Γ) : (blockSum t ρ χ g).charpoly = ((ρ g : GL (Fin r) O) : Matrix (Fin r) (Fin r) O).charpoly * ∏ c ∈ Finset.Icc 1 t, (X - C ((χ c g : Oˣ) : O))

theorem **continuous_blockSum_iff** [TopologicalSpace O] [TopologicalSpace Γ] {r : ℕ} (t : ℕ) (ρ : Γ →* GL (Fin r) O) (χ : ℕ → (Γ →* Oˣ)) (hχ : ∀ c, Continuous fun g => ((χ c g : Oˣ) : O)) : Continuous (fun g => blockSum t ρ χ g) ↔ Continuous (fun g => ((ρ g : GL (Fin r) O) : Matrix (Fin r) (Fin r) O))

theorem **isComplement_iff** (i : N.Loc) (π : N.X i) (ρ : GE N →* GL (Fin R.r) (N.O i)) : IsComplement N i π ρ ↔ IsRx N i π (sumChi N i ρ)

theorem **isRhoAt_iff** (i : N.Loc) (π : N.X i) (ρ : GE N →* GL (Fin R.r) (N.O i)) : IsRhoAt N i π ρ ↔ ∀ w : PlE H, SplitOut N i w → ∀ (B : ValuationSubring N.Om) (σ : GE N), IsPlaceOver (F

theorem **isRhoAt_of_splitsAs** (i : N.Loc) (π : N.X i) (rπ : GE N →* GL (Fin R.nV) (N.O i)) (ρ : GE N →* GL (Fin R.r) (N.O i)) (hr : FrobeniusMatrixCompatible N i π ((Units.coeHom _).comp rπ)) (hs : SplitsAs N i rπ ρ) : IsRhoAt N i π ρ

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.ResidualConstruction

theorem **isEnlargement_self** (rep : N.X i → (GE N →* GL (Fin R.nV) (N.O i))) (ρ : N.X i → (GE N →* GL (Fin R.r) (N.O i))) : IsEnlargement N i rep ρ (N.O i) (N.T i) rep ρ (N.chi i)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.ResidualLatticeSetting

theorem **maxLam_ne_top** (i : X.Tw) (M : ℕ) (ι : Type) : maxLam X i M ι ≠ ⊤

theorem **setting_of_auxiliaryPrimes** (i : X.Tw) (M : ℕ) (hM : 1 ≤ M) (A : Finset (PlE D.host)) {ι : Type} (P : ι → PlE D.host) (d : ι → (GE N →* Multiplicative (ZMod (X.l i ^ M)))) (Q : Finset (PlE D.host)) (h : SlopeTestRemainingChoices X i M A P d) (hQ : IsQ P Q) : HidaSetting X i M P Q d

theorem **selmerMatrixAuxiliaryPrimes_of_selMatrix** (h : SelMatrix bkP) : SelmerMatrixChosenPrimes bkP

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.SemisimplificationConstituents

theorem **blockSum_apply** {r t : ℕ} (ρ : Γ →* Matrix (Fin r) (Fin r) k) (χ : Fin t → (Γ →* kˣ)) (g : Γ) (u : (Fin r → k) × (Fin t → k)) : blockSum ρ χ g u = ((ρ g).mulVec u.1, fun c => ((χ c g : kˣ) : k) * u.2 c)

theorem **isSemisimplificationOf_self** {V : Type} [AddCommGroup V] [Module k V] (ρ : Representation k Γ V) (h : IsSS ρ) : IsSemisimplificationOf ρ ρ

theorem **repK_eq** (r : RepFam D i) (x : D.nd.X i) : repK D.nd i r x = matRep (matAlong D i (toK D.nd i) (r x))

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.SpehProductSatake

theorem **infString_one** (A : Multiset ℂ) : infString 1 A = A

theorem **satString_one** (q : ℕ) (B : Multiset ℂˣ) : satString q 1 B = B.map Units.val

theorem **infString_zero** (n : ℕ) : infString n {0} = Finset.univ.val.map fun k : Fin n => ((n : ℂ) - 1) / 2 - (k : ℕ)

theorem **satString_two** (q : ℕ) (hq : (q : ℂ) ≠ 0) : satString (q ^ 2) 2 {1} = (GlobalTheory.LGrp.bcParam 2 fun _ => Units.mk0 (q : ℂ) hq).map Units.val

theorem **isSatStringAt_one** (π : Submodule ℂ (UA E 1 → ℂ)) (w : HeightOneSpectrum (𝒪 E)) : IsSatStringAt π (a

theorem **isSpehBaseChange_const_one** : IsSpehBaseChange (Submodule.span ℂ {(fun _ => 1 : UA E 1 → ℂ)}) (a

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.TensorLatticeUnitPart

theorem **extBil_tmul** (b : M →ₗ[O] M' →ₗ[O] M'') (r r' : R) (x : M) (y : M') : extBil R b (r ⊗ₜ[O] x) (r' ⊗ₜ[O] y) = (r * r') ⊗ₜ[O] b x y

theorem **extForm_tmul** (b : M →ₗ[O] M' →ₗ[O] O) (r r' : R) (x : M) (y : M') : extForm R b (r ⊗ₜ[O] x) (r' ⊗ₜ[O] y) = b x y • (r * r')

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.TensorMultiplication

theorem **actVB_mulVB** (ρ : Representation LL G V) (g : G) (c : K₀) (x : V ⊗[Qp] B) : actVB PD ρ g (mulVB (K₀

theorem **comm_smul** [SMulCommClass Qp K₀ B] (c : K₀) (x : B ⊗[Qp] V) : TensorProduct.comm Qp B V (c • x) = mulVB (K₀

theorem **mulTens_tmul** (v : V) (v' : V') (b b' : B) : mulTens Qp B LL V V' ((v ⊗ₜ[Qp] b) ⊗ₜ[LL] (v' ⊗ₜ[Qp] b')) = (v ⊗ₜ[LL] v') ⊗ₜ[Qp] (b * b')

theorem **invB_holds** : InvB PD

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.TwistedGLEmbedding

theorem **pW_eq** : pW a = (R.k : ℚ) + a.1

theorem **qW_eq** : qW a = (R.m : ℚ) - a.1 - 1

theorem **nV_eq** (R : Ranks) : (R.nV : ℚ) = (R.k : ℚ) + R.m - 1 + R.t

theorem **lamWt_eq** : lamWt P a = (fullPar P a).map (· - 1 / 2)

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.UnitDualParts

theorem **extMul_tmul** {O : Type} [CommRing O] (R : Type) [CommRing R] [Algebra O R] {M M' M'' : Type} [AddCommGroup M] [Module O M] [AddCommGroup M'] [Module O M'] [AddCommGroup M''] [Module O M''] (m : M ⊗[O] M' →ₗ[O] M'') (r r' : R) (a : M) (a' : M') : extMul R m (r ⊗ₜ[O] a) (r' ⊗ₜ[O] a') = (r * r') ⊗ₜ[O] m (a ⊗ₜ[O] a')

theorem **extPair_tmul** {O : Type} [CommRing O] (R : Type) [CommRing R] [Algebra O R] {M M' : Type} [AddCommGroup M] [Module O M] [AddCommGroup M'] [Module O M'] (β : M →ₗ[O] M' →ₗ[O] O) (r r' : R) (a : M) (a' : M') : extPair R β (r ⊗ₜ[O] a) (r' ⊗ₜ[O] a') = β a a' • (r * r')

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.UnramifiedNoJump

theorem **conj_conj** (hc : IsCMConjOm E Ω c) (x : Ω) : c (c x) = x

theorem **conj_symm** (hc : IsCMConjOm E Ω c) (x : Ω) : c.symm x = c x

theorem **conjElt_apply** (hc : IsCMConjOm E Ω c) (g : G E Ω) (x : Ω) : conjElt hc g x = c (g (c.symm x))

theorem **scalGL_comm** (a : Aˣ) (M : GL (Fin n) A) : Commute (scalGL a) M

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.WachCrystallineComparison

theorem **WachCrisComparison.span** {l : ℕ} [Fact l.Prime] {P : LocPrime l} {AI : LocAinf l P} {H : LocHW l P} {F : LocFon l P AI H} {C : LocCmpT l P AI H F} (hc : WachCrisComparison l P AI H F C) : WachCrisComparisonSpan l P AI H F C

Development.AbelianVarietyTheory.BlockTheory.NonGaloisCusps.WachLettersFree

theorem **IsWachBasis.letter** {r : ℕ} {b : Fin r → W.DD T} (hb : IsWachBasis W T b) (i : ZMod n) : IsLetterBasis W T i (fun k => (letE i : LetA O n) • b k)

theorem **FrobBounds.pull** {M₀ h₀ : ℕ} (hF : FrobBounds W T M₀ h₀) : ∀ y ∈ W.φpull T, W.Q ^ M₀ • y ∈ W.N T

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.CurvatureFamilyNorm

theorem **weight_eta_sq_theta** : (2 : ℤ) • w.wEt + (w.n₁ : ℤ) • w.twoG₁ = (2 * (w.n₁ : ℤ)) • w.L

theorem **weight_product** : (2 * (w.n₁ : ℤ)) • (w.ka + w.G₂) + (2 * (w.n₁ : ℤ)) • w.twoG₁ + (4 : ℤ) • w.wEt = (2 * (w.n₁ : ℤ) * (w.T + nd X + 6)) • w.L

theorem **pairing_rem** (τ : X.Ψ) (a b : Fin r) (ha : (a : ℕ) = 0) (hb : (b : ℕ) + 1 = r) : 2 * (w.hF : ℤ) + 2 ≤ w.rem τ b - w.rem τ a

theorem **pairing_L** (τ : X.Ψ) (a b : Fin r) (ha : (a : ℕ) = 0) (hb : (b : ℕ) + 1 = r) : 38 * (w.T : ℤ) ≤ 10 * (w.L τ b - w.L τ a)

theorem **t_mul_r_le** : w.t * r ≤ 2 * w.T

theorem **nonempty_weights** (X : Data E r) (h : Shape X) : Nonempty (Weights X)

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.EtaDualPairing

theorem **isLocalRing_dualNumber** (A : Type u) [CommRing A] [IsLocalRing A] : IsLocalRing (DualNumber A)

theorem **red_comp_incl** (R : LAlg R₀) : (red R).comp (incl R) = LHom.id R

theorem **eta_mul_eta** (R : LAlg R₀) : eta R * eta R = 0

theorem **red_sqZero** (R : LAlg R₀) : (red R).SqZero

theorem **red_etaPt** (R : LAlg R₀) (v : B.gg ⊗[R₀] R) : B.map (red R) (etaPt B R v) = 0

theorem **red_constPt** (R : LAlg R₀) (y : B.G R) : B.map (red R) (constPt B R y) = y

theorem **nsmul_etaPt** {R : LAlg R₀} {N : ℕ} (hN : (N : R) = 0) (v : B.gg ⊗[R₀] R) : N • etaPt B R v = 0

theorem **nsmul_constPt** {R : LAlg R₀} {N : ℕ} {y : B.G R} (hy : N • y = 0) : N • constPt B R y = 0

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.ExhaustionFrames

theorem **zDet_pos** {z : V (nd X)} (hz : z ∈ Om X) (τ : X.Ψ) : 0 < zDet X z τ

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.FormalGroupTorsionCharts

theorem **evalTerm_finite** [Finite σ] (F : MvPowerSeries σ A) (x : σ → R) (hx : ∀ i, IsNilpotent (x i)) : (Function.support (evalTerm F x)).Finite

theorem **evalNil_eq_finsum** [Finite σ] (F : MvPowerSeries σ A) (x : σ → R) (hx : ∀ i, IsNilpotent (x i)) : evalNil F x hx = ∑ᶠ d, evalTerm F x d

theorem **map_mem_kerRes** (B : BX R₀) {R S : ResAlg k} (f : LHom R.alg S.alg) (hf : S.res.comp f = R.res) (x : kerRes B R) : B.map f x ∈ kerRes B S

theorem **tg_mem_kerRes** (B : BX R₀) (R : ResAlg k) {S : LAlg R₀} (π : LHom R.alg S) (hπ : π.SqZero) (w : B.gg ⊗[R₀] π.ker) : B.tg π hπ w ∈ kerRes B R

theorem **tateModule_ker** (B : BX R₀) (p : ℕ) (O : LAlg R₀) (hτ : τ ∈ tateModule B p O) : τ 0 = 0 ↔ ∃ τ' ∈ tateModule B p O, τ = p • τ'

theorem **classify_spec** (t : Tors B p O.alg) : (D.classify t).2.comp (D.str _) = LHom.id _ ∧ B.map (D.classify t).2 (D.y (D.classify t).1 : B.G (D.A _).alg) = (t : B.G O.alg)

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.FundamentalSetFrames

theorem **pairing_kapW** (X : Data E r) (h0 : 0 < X.p0) (h1 : X.p0 < r) (τ : X.Ψ) : pairing X h0 h1 (kapW X) τ = 2 * (r : ℤ) - 2

theorem **isInvWeight_const** (X : Data E r) (c : ℝ) : IsInvWeight X fun _ => (c : EReal)

theorem **continuous_dmin** (X : Data E r) (τ : X.Ψ) (i : Fin (r + 1)) : Continuous fun x : Frame X.Ψ r => dmin X.Gm x τ i

theorem **isInvWeight_log_norm** (X : Data E r) (L : Wt X.Ψ r) (s : Frame X.Ψ r → ℂ) (hs : X.IsBddSec L s) (hinv : ∀ γ ∈ X.Γ, ∀ x ∈ X.dom, ∀ b ∈ Bor X.Ψ r, X.pnorm L s (lact γ (rmul x b)) = X.pnorm L s x) : IsInvWeight X fun x => ENNReal.log (ENNReal.ofReal (X.pnorm L s x ^ 2))

theorem **zero_mem_ATilde** (X : Data E r) (x0 : Frame X.Ψ r) (bb : ℝ) (la : Wt X.Ψ r) (u : Frame X.Ψ r → EReal) : (fun _ => 0) ∈ ATilde X x0 bb la u

theorem **poin_eq** (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) (x : Frame X.Ψ r) : poin X la P x = ∑' γ, term X la P γ x

theorem **poin_mem_ASpace** (h : PoincareCuspidalStatement E) (X : Data E r) (x0 : Frame X.Ψ r) (hx0 : IsOrth X x0) (bb : ℝ) (hb : 4 * (X.Ψ.card : ℝ) * (r : ℝ) ^ 2 < bb) (la : Wt X.Ψ r) (hg : GapsNonneg X la) (hp : ∀ (h0 : 0 < X.p0) (h1 : X.p0 < r), ∃ τ : X.Ψ, 0 < pairing X h0 h1 la τ) (u : Frame X.Ψ r → EReal) (hu : IsInvWeight X u) (F : V (nd X) → ℂ) (hF : F ∈ ATilde X x0 bb la u) (P : Frame X.Ψ r → ℂ) (hP : IsExt X x0 la F P) (FF : Set (V (nd X))) (hFF : IsFundSet X x0 FF) : poin X la P ∈ ASpace X x0 la u FF

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.InterpolationPeaks

theorem **four_lt_bExp** (X : Data E r) (hf : 0 < X.Ψ.card) (hr : 0 < r) : 4 * (X.Ψ.card : ℝ) * (r : ℝ) ^ 2 < bExp X

theorem **factor_mul_firstMinimum** {X : Data E r} (Xa : ArchInM X) : tFac Xa * Xa.a1 = Xa.c1 + Xa.a1

theorem **tFac_range** {X : Data E r} (Xa : ArchInM X) : 0 ≤ tFac Xa ∧ tFac Xa * Xa.rr ≤ 2 * Xa.T

theorem **peak_of_interpGt** (h : InterpGtStatement E) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (hd : PropDisc X) (hm : ModeBound X) (hq : HasMargin X Xa.Fp Xa.L (fun x => tFac Xa * Xa.ps0 x) (bExp X)) (j : Fin Fl.δ) : ∃ P ∈ X.SSgt Xa.Fp Xa.c1, ∀ j', P (Fl.xs j') ≠ 0 ↔ j' = j

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.LowSpanCombinations

theorem **mem_lowIdx** {D T : ℕ} {al : Fin D → ℕ} : al ∈ lowIdx D T ↔ ∑ i, al i + D < T

theorem **nonempty_cmType** (X : Data E r) : Nonempty X.Ψ

theorem **tt_eq** (X : Data E r) (ξ : Fin r → E) (x : Frame X.Ψ r) (τ : X.Ψ) : X.tt ξ x τ = 1 / hf X ξ x τ

theorem **admissible_zero** (X : Data E r) : Admissible X (fun _ => 0) 0

theorem **cpt_zero** (X : Data E r) (x0 : Frame X.Ψ r) : cpt X x0 0 = x0

theorem **isFlat_rank_one** (X : Data E 1) (x0 : Frame X.Ψ 1) (h0 : x0 ∈ X.dom) : IsFlat X 0 (fun _ : Fin 1 => fun _ => 1) (fun _ => 1) (fun _ _ => 0) (fun _ : Fin 1 => x0)

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.PluriharmonicHolomorphic

theorem **cx_mul_fromRows** (γ : Matrix.symplecticGroup ι ℤ) (τ : Matrix ι ι ℂ) : cx γ.1 * fromRows τ (1 : Matrix ι ι ℂ) = fromRows (num γ τ) (denom γ τ)

theorem **cx_symplectic** (γ : Matrix.symplecticGroup ι ℤ) : (cx γ.1)ᵀ * J ι ℂ * cx γ.1 = J ι ℂ

theorem **cx_conjTranspose** {m n : Type} (M : Matrix m n ℤ) : (cx M)ᴴ = (cx M)ᵀ

theorem **fromRows_transpose_J_mul_fromRows** (P Q : Matrix ι ι ℂ) : (fromRows P Q)ᵀ * J ι ℂ * fromRows P Q = Qᵀ * P - Pᵀ * Q

theorem **form_H** (P Q : Matrix ι ι ℂ) : (fromRows P Q)ᴴ * J ι ℂ * fromRows P Q = Qᴴ * P - Pᴴ * Q

theorem **hecke_relation_image** (γ : Matrix.symplecticGroup ι ℤ) (τ : Matrix ι ι ℂ) : (denom γ τ)ᵀ * num γ τ - (num γ τ)ᵀ * denom γ τ = τ - τᵀ

theorem **rel_H** (γ : Matrix.symplecticGroup ι ℤ) (τ : Matrix ι ι ℂ) : (denom γ τ)ᴴ * num γ τ - (num γ τ)ᴴ * denom γ τ = τ - τᴴ

theorem **posDef_ofReal_iff** {Y : Matrix ι ι ℝ} (h : Y.IsSymm) : (Y.map Complex.ofReal).PosDef ↔ Y.PosDef

theorem **posDef_im_iff** {τ : Matrix ι ι ℂ} (h : τ.IsSymm) : (τ.map Complex.im).PosDef ↔ ((-Complex.I / 2) • (τ - τᴴ)).PosDef

theorem **isUnit_denom** (γ : Matrix.symplecticGroup ι ℤ) (τ : SiegelH ι) : IsUnit (denom γ τ.1)

theorem **act_mem** (γ : Matrix.symplecticGroup ι ℤ) (τ : SiegelH ι) : num γ τ.1 * (denom γ τ.1)⁻¹ ∈ SiegelH ι

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.SpanSubtraction

theorem **VanOrd.add** {X : Data E r} {Xa : ArchInM X} {Fl : FlatM X Xa} {a : ℕ} {f g : Frame X.Ψ r → ℂ} (hf : VanOrd Fl a f) (hg : VanOrd Fl a g) : VanOrd Fl a fun x => f x + g x

theorem **VanOrd.eq_zero** {X : Data E r} {Xa : ArchInM X} {Fl : FlatM X Xa} {a : ℕ} (ha : a ≠ 0) {f : Frame X.Ψ r → ℂ} (hf : VanOrd Fl a f) (j : Fin Fl.δ) (hs : ∀ k, Fl.s k (cpt X (Fl.xs j) 0) = 0) : f (cpt X (Fl.xs j) 0) = 0

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.TorusThetaNorms

theorem **exists_col_mem** (τ : D.Ψ) (i : Fin (T.X.dim τ.1)) : ∃ a : Fin r, (a : ℕ) = D.p0 + i

theorem **exists_col_conj** (τ : D.Ψ) (i : Fin (T.X.dim (ComplexEmbedding.conjugate τ.1))) : ∃ a : Fin r, (a : ℕ) + i + 1 = D.p0

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.TwoLevelLogarithm

theorem **epsBound** : EpsBound

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.UnitriangularCharts

theorem **card_below** (r : ℕ) : Fintype.card (Below r) = r * (r - 1) / 2

theorem **card_index** (X : Data E r) : Fintype.card (X.Ψ × Below r) = flagDim X

Development.AbelianVarietyTheory.BlockTheory.PoincareCuspForms.WordDerivativeBlocks

theorem **Block.isHomogeneous_entry** (B : Block dim n) (a b : Fin B.size) : (B.entry a b).IsHomogeneous 1

theorem **Block.isHomogeneous_poly** (B : Block dim n) : B.poly.IsHomogeneous (B.size * B.expo)

theorem **BlockSystem.isHomogeneous_poly** (S : BlockSystem dim n) : S.poly.IsHomogeneous S.degree

theorem **iteratedFDeriv_directions_comp_perm_of_analyticAt** (dir : Λ → W) {F : W → ℂ} {p : W} (hF : AnalyticAt ℂ F p) {k : ℕ} (u v : Fin k → Λ) (σ : Equiv.Perm (Fin k)) (h : u = v ∘ σ) : iteratedFDeriv ℂ k F p (fun j => dir (u j)) = iteratedFDeriv ℂ k F p (fun j => dir (v j))

theorem **wordDeriv_perm** (dir : Λ → W) {F : W → ℂ} {p : W} (hF : AnalyticAt ℂ F p) {l l' : List Λ} (h : l.Perm l') : wordDeriv dir l F p = wordDeriv dir l' F p

theorem **ph_nonneg** (f : ℕ) (rd : ℝ) (q : ℕ) (R₀ : ℝ) (la : Set.Ioi (0 : ℝ)) : 0 ≤ ph f rd q R₀ la

theorem **exp_pTwo** (f m : ℕ) : Real.exp (pTwo f m) = 2 * (3 * (dStar f m : ℝ)) ^ dStar f m

theorem **jetBound_pos** (la : Set.Ioi (0 : ℝ)) : 0 < P.jetBound la

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.ConjugacyTypeUnfolding

theorem **fix_of_fix** {I : AutInst} (h : I = instOf D) (x : Fin I.n → I.E) (g : GA D (D.sp.Y 0)) (hg : Formalism.L2.Theta.mt (D.sp.Y 0) g *ᵥ ratV (crdY h x) = ratV (crdY h x)) : (∀ v, uMat ((((Formalism.IndefR.RationalEmbedding.iota D).trans (castGA h.symm)) g).1 v) *ᵥ rv I v x = rv I v x) ∧ I.fmat (((Formalism.IndefR.RationalEmbedding.iota D).trans (castGA h.symm)) g).2 *ᵥ incl I x = incl I x

theorem **embA_fix** (hF : IsFrame X h u d o ζ) (a : GA D A) (x : Fin X.B.I.n → X.B.I.E) (hx : x ∈ fixSet X u) : Formalism.L2.Theta.mt (D.sp.Y 0) (embA d o a) *ᵥ ratV (crdY h x) = ratV (crdY h x)

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.DeterminantHeightFlags

theorem **mem_parFlag_iff** {C : Matrix (Fin n) (Fin j) E} {D : Matrix (Fin j) (Fin n) E} (hD : D * C = 1) (p : GAd σ J) : p ∈ parFlag σ J C (fun _ => 0) ↔ ∃ B : GL (Fin j) (AF F ⊗[F] E), (matHom _ σ J p).1 * adMat F C = adMat F C * B.1

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.FlatOmegaSeries

theorem **oneCoset_iff** (ψ : AddChar k Circle) (ν : Measure k) (μ : kˣ →* ℂˣ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (J : Subgroup (GL (n ⊕ Fin c) k)) : OneCoset ψ ν μ Φ J ↔ ∀ h, flat ψ ν μ Φ 1 h ≠ 0 → ∃ r ∈ R k n c 1, ∃ κ ∈ J, h = r * κ

theorem **Collapse.weak** (H : Collapse) : CollapseWeak

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.GramMatrixAdjoint

theorem **bddAbove_witt** {m : ℕ} (T : Matrix (Fin m) (Fin m) D.host.E) : BddAbove {r | ∃ W : Submodule D.host.E (Fin m → D.host.E), (∀ x ∈ W, ∀ y ∈ W, hE D.sp.cm.sig T x y = 0) ∧ Module.finrank D.host.E W = r}

theorem **upperLeft_blocks** {R : Type} {n : ℕ} (a b c d : Matrix (Fin n) (Fin n) R) : upperLeft (blocks a b c d) = a

theorem **mem_siegel_of_blocks** {n : ℕ} {g : GG D n} {a b d : Mx D n n} (h : mat g = blocks a b 0 d) : g ∈ siegel D n

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.LocalOrderMatrix

theorem **ordX_d** (R : Ranks) (j : ℕ) (i : Fin j) (l : Fin R.t) : (ordX R j (.inr (i, l))).val = R.k + j + (l.val + R.t * i.val)

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.QuasiFamilyPeriods

theorem **isQuasiFam_of_isThetaFam** {S L : HSp D} {Y : GA D S → WO S L → ℂ} (hY : IsThetaFam ψ S L Y) : IsQuasiFam ψ S L Y

theorem **PeriodLink.quasi** (H : PeriodLink D) : PeriodLinkQuasi D

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.SymmetricPatternGroup

theorem **mul_apply_of_no_path** {A : ι → ι → Prop} {g h : Matrix ι ι F} (hg : ∀ r s, ¬ A r s → g r s = (1 : Matrix ι ι F) r s) (hh : ∀ r s, ¬ A r s → h r s = (1 : Matrix ι ι F) r s) (hi : ∀ r, ¬ A r r) {r s : ι} (hp : ∀ l, A r l → ¬ A l s) : (g * h) r s = g r s + h r s − (1 : Matrix ι ι F) r s

theorem **mem_patGrp** {A : ι → ι → Prop} {hi ht} {g : GL ι F} : g ∈ patGrp F A hi ht ↔ ∀ r s, ¬ A r s → g.val r s = (1 : Matrix ι ι F) r s

theorem **symFree_irrefl** (r : Fin κ ⊕ Fin (2 * ρ + 1)) : ¬ SymFree κ ρ r r

theorem **symFree_trans** (r l s : Fin κ ⊕ Fin (2 * ρ + 1)) : SymFree κ ρ r l → SymFree κ ρ l s → SymFree κ ρ r s

theorem **mem_symGrp_iff** {g : GL (Fin κ ⊕ Fin (2 * ρ + 1)) F} : g ∈ symGrp F κ ρ ↔ ∃ n ∈ radN F κ ρ, ∃ u ∈ epsU F κ ρ, g = n * u

theorem **sumB_mul** {g h : GL (Fin κ ⊕ Fin (2 * ρ + 1)) F} (hg : g ∈ symGrp F κ ρ) (hh : h ∈ symGrp F κ ρ) : sumB (g * h) = sumB g + sumB h

theorem **sumEps_mul** {g h : GL (Fin κ ⊕ Fin (2 * ρ + 1)) F} (hg : g ∈ symGrp F κ ρ) (hh : h ∈ symGrp F κ ρ) : sumEps (g * h) = sumEps g + sumEps h

theorem **exists_isUnif** : ∃ ϖ : kˣ, IsUnif ϖ

theorem **psi_congr** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ϖ : kˣ} (hϖ : IsUnif ϖ) {a b : 𝒪[k]} (h : IsLocalRing.residue 𝒪[k] a = IsLocalRing.residue 𝒪[k] b) : ψ ((a : k) / ϖ) = ψ ((b : k) / ϖ)

theorem **psiZero_residue** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ϖ : kˣ} (hϖ : IsUnif ϖ) (u : 𝒪[k]) : psiZero hψ hϖ (IsLocalRing.residue 𝒪[k] u) = ψ ((u : k) / ϖ)

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.UnimodularTower

theorem **Fin2.Q_eq** (e : Fin2 N top) (P₁ P₂ φ g) (F : GA D T → ℂ) : e.t.Q φ (e.t.s.om (g, 1) (e.p₁.pt P₁)) (e.p₂.pt P₂) F = ∫ h' in e.t.vT.𝓕, F h' * e.inner P₁ P₂ φ g h' ∂e.t.vT.μ

theorem **Tower.Is.index** {T : HSp D} {top : RowOn T} {ρ j : ℕ} {N : Node c u ρ j} {t : Tower c u top ρ j N} (h : t.Is) : j + 2 * ρ + 2 = D.R.m

Development.AbelianVarietyTheory.BlockTheory.SiegelEisenstein.WeightedNormColumns

theorem **zCol_zero** (ρ : ℕ) : zCol n' ρ (fun _ : n => (0 : k)) = 0

theorem **flatE_zero** (χ : AddChar k Circle) (ν : Measure k) (ρ : ℕ) (Φ : Matrix (n ⊕ Fin (2 * ρ)) (n' ⊕ Fin (2 * ρ + 1)) k → ℂ) (x₀ : Matrix n n' k) : flatE χ ν ρ Φ (fromCols x₀ 0) = GlobalTheory.RootEx.opE χ ν (ρ + 1) Φ x₀

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.CenteredPerpendicularSets

theorem **inCen_zero** (s : A →+* A) (J : Matrix (Fin n) (Fin n) A) (W : Submodule A (Fin n → A)) : InCen s J W 0

theorem **normAt_zero** (x : I.GA) : normAt I x 0 = 0

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.ColumnDimensionRational

theorem **sig_incl** (x : E) : sigR R σ (inclR R x) = inclR R (σ x)

theorem **bc_mem** (γ : ɪ(Formalism.UG.UJ (σ : E →+* E) J)) : Formalism.UG.bc (inclR R) γ.1 ∈ HR R σ J

theorem **ratG_mem_Hp** (P : Matrix (Fin n) (Fin (r + 1)) E) (f : Matrix (Fin n) (Fin r) E) (γ : ɪ(HF σ J P f)) : ratG R σ J γ.1 ∈ Hp R σ J P f

theorem **act_one** (x : Fin n → R ⊗[F] E) : act R σ J 1 x = x

theorem **act_mul** (g h : HA R σ J) (x : Fin n → R ⊗[F] E) : act R σ J (g * h) x = act R σ J g (act R σ J h x)

theorem **act_sub** (g : HA R σ J) (x y : Fin n → R ⊗[F] E) : act R σ J g (x − y) = act R σ J g x − act R σ J g y

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.DefiniteFiniteCount

theorem **DefiniteHyp.rank_eq** {B : RowsAut} {d : InfinitePlace B.I.F → ℝ} {p : ℕ} (h : DefiniteHyp B d p) : B.I.n = p

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.DiscriminantKernelAutomorphisms

theorem **mem_orthModIsotropicLatticePerpNegative_iff** (D : D7) (u : D.what₂.Word) (F0 : FlagG (VS D) D.R.rS ((Tu D u : ℕ) : D.host.E)) (hz : F0.z = D.zmOf u) (Λ : LatV D) (w : (D.sp.X 0).V) : w ∈ flagPairDefinitePartLattice D F0 Λ ↔ (w, (0 : D.host.E)) ∈ latL D F0 Λ

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.FrameCoefficientData

theorem **invVol_self** {G : Type} [Group G] (K₀ : Subgroup G) : invVol K₀ K₀ = 1

theorem **isInnerLattice_std** (n : ℕ) (A : Frame F (Fin n → E) 0) (i₀ : Fin n) (hz : ∀ i, A.z i ∈ MaxOrd o E) (h1 : A.z i₀ = 1) (hs : ∀ x ∈ MaxOrd o E, star x ∈ MaxOrd o E) : A.IsInnerLattice (negStd E n) (Submodule.span (MaxOrd o E) (Set.range fun i : Fin n => (Pi.single i 1 : Fin n → E)))

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.FullySplitStabilizers

theorem **tF_herm** (r : ℕ) (T : F) (a b : Fin (r + 1)) : σ (TF r (algebraMap F E T) b a) = TF r (algebraMap F E T) a b

theorem **gramTwo_herm** (hσ : ∀ x, σ (σ x) = x) {r : ℕ} (T T' : F) (Z : Matrix (Fin (r + 1)) (Fin (r + 1)) E) (a b : Fin ((r + 1) + (r + 1))) : σ (gramTwo σ T T' Z b a) = gramTwo σ T T' Z a b

theorem **invol_of_cm** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.HermitianMassFormula

theorem **TFE_herm** (σ : E →ₐ[F] E) (r : ℕ) (T : 𝒪 F) (a b : Fin (r + 1)) : σ (TFE E r T b a) = TFE E r T a b

theorem **sig_sig** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

theorem **blockMatrix_hermitian** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (r : ℕ) (T T' : 𝒪 F) (Z : Matrix (Fin (r + 1)) (Fin (r + 1)) E) (a b : Fin ((r + 1) + (r + 1))) : σ (T2 σ r T T' Z b a) = T2 σ r T T' Z a b

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.NearNonzeroTerms

theorem **nearAbs_eq_zero** (d : InfinitePlace B.I.F → ℝ) (x : B.I.GA) (h : ∀ γ, IsNear B d (epN B) x γ → γ = 1) : nearAbs B d x = 0

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.OrbitStepSpace

theorem **stepSpace_zero** (ε : Fin n → Bool) (lam : Fin n → ℝ) (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)) : stepSpace ε lam σ 0 = lowKSub ε lam σ

theorem **stepSpace_mono** (ε : Fin n → Bool) (lam : Fin n → ℝ) (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)) : Monotone (stepSpace ε lam σ)

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.OrthonormalPlaceBasis

@[simp] theorem **uMat_expU** {n : ℕ} {ε : Fin n → Bool} (X : gLie ε) : uMat (expU X) = NormedSpace.exp (X : Matrix (Fin n) (Fin n) ℂ)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalPlaceBasis.stepSpan_mono** {P Q : ℕ} (h : P ≤ Q) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalPlaceBasis.VP B P ≤ _root_.GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalPlaceBasis.VP B Q

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalPlaceBasis.tens_w_mem** {d : InfinitePlace B.I.F → ℝ} (h : _root_.GlobalTheory.Count.IsDSData B d) (P : ℕ) : B.tens B.w ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalPlaceBasis.VP B P ∧ ‖B.tens B.w‖ = 1

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.SaturatedPerpDuality

theorem **perpIn_bot** (I : AutInst) (x : I.GA) : perpIn I x ⊥ = lamX I x

Development.AbelianVarietyTheory.BlockTheory.SiegelWeilLocal.SubquotientSizeInequality

theorem **norm_ptAt** (b : Basis ι K V) (X : InfinitePlace K → Matrix ι ι ℂ) (v : InfinitePlace K) (y : V) : ‖ptAt b X v y‖ = normAt b X v y

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.AgreementAtAlmostAllSplitPrimes

theorem **agreeAtAlmostAllSplit_iff** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} (A : NAut N) (χ χ' : A.CG 1) : AgreeAtAlmostAllSplit A χ χ' ↔ ∀ᶠ w in Filter.cofinite, IsSplitPl H w → A.locG χ w = A.locG χ' w

theorem **HeckeCharEqOfSplit.of_forall** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} {A : NAut N} (h : HeckeCharEqOfSplit N A) (χ χ' : A.CG 1) (hw : ∀ w, IsSplitPl H w → A.locG χ w = A.locG χ' w) : χ = χ'

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.BadPrimesPeriodBound

theorem **T_mem_badS** (D : D7) (u : D.what₂.Word) : (D.mem u).A.T ∈ badS D

theorem **small_mem_badS** (D : D7) (l : ℕ) (h : l ∈ D.host.small) : l ∈ badS D

theorem **g_mem_badS** (D : D7) (i : Fin 2) : D.host.g i ∈ badS D

theorem **recSeam_emb** (D : D7) (τ : D.what₂.Tau) (x : D.host.E) : (recSeam D).σ₀ ((recFields D).emb τ x) = TaylorLogMetric.embT D ((recSeam D).iT τ) ((recSeam D).iE x)

theorem **recWts_row** {D : D7} (EX : ErrX D) (u : D.what₂.Word) (τ : D.what₂.Tau) : RowPar.of ((recWts EX).hc u τ) = (recWts EX).row τ

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.BlockSumMultiplicities

theorem **SecondTwistCrystallineSmall.le** {L : SizeData₂ D} (h : SecondTwistCrystallineSmall L) : SecondTwistCrystallineAtMost L

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.ConjugacyRepresentativeSystems

theorem **SameCls.symm** {d d' : DecoSet E n ST SG} (h : SameCls d d') : SameCls d' d

theorem **SameCls.trans** {d d' d'' : DecoSet E n ST SG} (h : SameCls d d') (h' : SameCls d' d'') : SameCls d d''

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.CrystallineAtPlaces

theorem **embLL_sigmaOf** (g : D.R.GG) (x : D.host.E) : D.what₂.embLL (D.embE (D.host.sigmaOf g x)) = D.host.emb g x

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.FourierColumnIndicator

theorem **colFn_indicator** (P : HeightOneSpectrum (𝒪 E)) {n : ℕ} (y₀ : Sp E n) (h₀ : (qOf P : E) • y₀ ∈ stdAt E n P) : ColFn P (fun y => if y − y₀ ∈ stdAt E n P then (1 : K) else 0)

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.FrobeniusRankParts

theorem **twistRep_val** (ρ : Γ →* GL (Fin a) O) (ξ : Γ →* Oˣ) (g : Γ) : ((twistRep ρ ξ g : GL (Fin a) O) : Matrix (Fin a) (Fin a) O) = (ξ g : O) • ((ρ g : GL (Fin a) O) : Matrix (Fin a) (Fin a) O)

theorem **IsRankPart.charpoly_frob** {i : N.Loc} {x : N.X i} {ρ' : G H.E N.Om →* GL (Fin R.r) (N.O i)} (h : IsRankPart N i x ρ') {r : G H.E N.Om →* GL (Fin R.nV) (N.O i)} (hr : IsHeckeGalois N i x r) {w : PlE H} (hw : SplitOut N i w) {B : ValuationSubring N.Om} {σ : G H.E N.Om} (hB : IsPlaceOver (F

theorem **mem_ZgoodN_of_param** (A : NAut N) {i : N.Loc} {x : N.X i} {Sig : List (Part A.CG)} (hx : HasParamX N A i x Sig) (hg : goodSig N A Sig) : x ∈ ZgoodN N A i

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.GaloisRepresentations

theorem **GalFull.gal** (h : GalFull N A) : Gal N A

theorem **IsGalRep.crys** {i : N.Loc} {a : ℕ} {π : A.CG a} {r : Rep N i a} (h : IsGalRep N A i π r) (w : PlE H) (hw : w ∈ placesOver H.E (N.l i)) (hu : A.unr (A.locG π w)) : A.IsCrysW i r w

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.HeckeIntertwinersInert

theorem **IsInertOp.unique** {R : RealDat D A Ks} {F : Face D A} (hC : R.Char F) {t : T} {i : InertIx R t} {U U' : Module.End ℂ (R.HX t).H} (h : IsInertOp R t i U) (h' : IsInertOp R t i U') : U = U'

theorem **eigSum_comp** {V I J : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] {op : I → Module.End ℂ V} (hc : Col.PairComm op) (φ : J → I) (C : (J → ℂ) → Prop) : Col.eigSum op (fun x => C fun j => x (φ j)) = Col.eigSum (fun j => op (φ j)) C

theorem **HeckeRows.rowsAt** {R : RealDat D A Ks} (h : HeckeRows R) {I : T → Type} (lab : ∀ t, I t → InertIx R t) : RowsAt R lab

theorem **HeckeRows.split** {R : RealDat D A Ks} (h : HeckeRows R) : HeckeRowsSplit R

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.InertLocalIndices

theorem **locAt_one** (V : HSp D) (v : HeightOneSpectrum (𝒪 D.host.F)) : LocAt V v 1

theorem **hasResC_map** (p : Pl0 → D.what₂.LL[X]) (x : A.ok A.μ0 × ℕ → D.what₂.LL) : HasResC A p (fun i => D.what₂.embLL (x i)) ↔ (A.fam.R A.μ0).HasRes p x

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.InternalDirectSumOfParts

theorem **partsSum_iff_isInternal** (R : Ranks) {FT : Type u} [DecidableEq FT] (M : ℕ → Type v) [∀ j, AddCommGroup (M j)] [∀ j, Module ℂ (M j)] (part : ∀ j, FT → Submodule ℂ (M j)) : PartsSum R M part ↔ ∀ j : ℕ, 1 ≤ j → j ≤ R.m → DirectSum.IsInternal (part j)

theorem **partsSum_iff_partsSumIn_top** (R : Ranks) {FT : Type u} (M : ℕ → Type v) [∀ j, AddCommGroup (M j)] [∀ j, Module ℂ (M j)] (part : ∀ j, FT → Submodule ℂ (M j)) : PartsSum R M part ↔ PartsSumIn R M (fun _ => ⊤) part

theorem **partsSum_plane** (R : Ranks) : PartsSum R (fun _ => ℂ × ℂ) (fun _ (b : Bool) => cond b (LinearMap.range (LinearMap.inl ℂ ℂ ℂ)) (LinearMap.range (LinearMap.inr ℂ ℂ ℂ)))

theorem **not_partsSum_repeated** (R : Ranks) : ¬ PartsSum R (fun _ => ℂ) (fun _ (_ : Bool) => (⊤ : Submodule ℂ ℂ))

theorem **not_partsSum_zero** (R : Ranks) : ¬ PartsSum R (fun _ => ℂ) (fun _ (_ : Unit) => (⊥ : Submodule ℂ ℂ))

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.KernelAwayPrimes

theorem **valInt_zero** (Tw : TwForms D) {u : D.what₂.Word} {μ : D.what₂.Tup} (O : LiftObj D (D.chTw u μ)) {p : ℕ} [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (i : ℕ) : ValInt Tw O lam i 0

theorem **formLoss_of_valLoss** {Dx : DivX D} {Tw : TwForms D} (E : KerEnt D Dx Tw) (X : ErrX D) {u : D.what₂.Word} {μ : D.what₂.Tup} (O : LiftObj D (D.chTw u μ)) (ig : Fin 2) (w : D.what₂.places.Pl) (i : ℕ) (h : ValLoss E X O ig w i) (h0 : LatTie E O w i) (h1 : LatTie E O w (i + 1)) : FormLoss Tw X u μ ig w i

theorem **exp_le** (D : D7) (i : ℕ) (hi : i < D.R.m) : D.R.f * D.R.N i * D.R.N (i + 1) ≤ D.R.f * D.R.N D.R.m * (D.R.N D.R.m + D.R.t)

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.LocalInvariantsOfColumns

theorem **noInv_iff** {k Γ V : Type} [Field k] [Group Γ] [AddCommGroup V] [Module k V] (ρ : Representation k Γ V) (D : Subgroup Γ) : NoInv ρ D ↔ ∀ v : V, (∀ g ∈ D, ρ g v = v) → v = 0

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.LocalTransferIntertwiners

theorem **HasResC.zero** {j : Fin D.R.m} {Sg : Pl0 → D.what₂.LL[X]} {x : SphIx A j A.μ0 → ℂ} (h : HasResC A j Sg x) (P : A.ok A.μ0) : x (Sum.inl (P, 0)) = D.what₂.embLL ((Sg P * (A.fam.L j A.μ0).blockProd P).coeff 0)

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.NoLocalInvariants

theorem **matW_mul** (i : N.Loc) (ρ : G H.E N.Om →* GL (Fin R.r) (N.O i)) (g h : G H.E N.Om) : matW N i ρ (g * h) = matW N i ρ g * matW N i ρ h

theorem **matW_one** (i : N.Loc) (ρ : G H.E N.Om →* GL (Fin R.r) (N.O i)) : matW N i ρ 1 = 1

theorem **matT_transpose_mul_matW** (i : N.Loc) (ρ : G H.E N.Om →* GL (Fin R.r) (N.O i)) (g : G H.E N.Om) : (matT N i ρ g)ᵀ * matW N i ρ g = algebraMap (N.O i) (Kf N i) ((eps N i g : (N.O i)ˣ) : N.O i) • (1 : Matrix _ _ (Kf N i))

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.PeriodLowerBounds

theorem **stmtFam_of** (h : Stmt) {eps : ℝ} (Fm : FamP eps) (C1 : ℝ) : FamilyErrorExponentBound Fm C1

theorem **T_ne_g** {D : D7} (X : ErrX D) (u : D.what₂.Word) (i : Fin 2) : D.host.g i ≠ (D.mem u).A.T

theorem **expLe_of** {D : D7} {L : SizeData₂ D} {X : ErrX D} {lev : LevelData D} {K : Consts} {c : ErrC} {C1 : ℝ} (hF : L.FDef) (hE : ErrDefs D L X lev K c) (hC : 2 * C1 ≤ c.g) : ExpLe L X C1

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.PolynomialIntegralityJets

theorem **not_mem_S** {D : D7} {Dx : DivX D} {u : D.what₂.Word} {μ : D.what₂.Tup} {O : GL3Theta.LiftObj D (D.chTw u μ)} (h : LevelTie Dx O) (w : D.what₂.places.Pl) : LevFree Dx u μ w ↔ D.what₂.places.resChar w ∉ O.S

theorem **polyIntegral_applyAt0** {Lt : Type} [Fintype Lt] [DecidableEq Lt] (N0 N1 l : ℕ) [Fact l.Prime] (F : MvPowerSeries (Lt × Fin N0 × Fin N1) ℂ_[l]) (h : JetsIntegral F) (P : MvPolynomial (KM.XZ Lt N0 N1) ℤ) : PolyIntegral (KM.applyAt0 N0 N1 l F P)

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.SmallRowEigenvalues

theorem **one_lt_normV** {E : Type} [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) : 1 < normV w

theorem **rowEigSmall_of_le** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} (N : NData R H what₂) (A : NAut N) (h : ∀ (i : N.Loc) (σ : what₂.FT), what₂.isot (N.j i) σ ≠ ⊥ → ∀ w : PlE H, IsSplitPl H w → A.unr (A.locW i σ w) → ∀ α ∈ A.frobEig (A.locW i σ w), ‖α‖ ≤ ((normV

w : ℕ) : ℝ) ^ (((R.r : ℝ) - 1) / 2)) : RowEigSmall N A

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.TameEigenvaluesTypePrimes

theorem **exists_isFrobAtType** (k : Fin 2) (v : D.nd.PlG k) : ∃ φ, IsFrobAtType D k v φ

theorem **tameGenerator_of_strong** (h : TameAtTypePrimeStrong D) : TameGeneratorAtTypePrime D

theorem **root_ne_zero** (σ : D.what₂.FT) (hσ : D.nd.goodRow σ) (k : Fin 2) (v : D.nd.PlG k) : haveI : Fact (D.host.g k).Prime

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.ThetaEigenspaceImage

theorem **mjSig_le_mjSigOut** (i : N.Loc) (Sig : List (Part A.CG)) : MjSig N A i Sig ≤ MjSigOut N A i Sig

theorem **thetaTMapsIntoEig_of_out** (h : ThetaTMapsIntoEigOut N A) : ThetaTMapsIntoEig N A

theorem **thetaTMapsIntoEig_iff_map_le** : ThetaTMapsIntoEig N A ↔ ∀ (i : N.Loc) (x : N.X i) (Sig : List (Part A.CG)), x ∈ ZgoodN N A i → HasParamX N A i x Sig → (MjSig N A i Sig).map (what₂.thetaT (N.mu i) (N.j i)) ≤ eigC N A i x

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.ThetaGramMatrix

theorem **ThetaGramPositiveScalar.norm_sq** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} {A : NAut N} (h : ThetaGramPositiveScalar N A) (i : N.Loc) (σ : what₂.FT) (f : what₂.Mj (N.j i)) (hf : f ∈ what₂.isot (N.j i) σ) : ‖what₂.thetaT (N.mu i) (N.j i) f‖ ^ 2 = what₂.lamP (N.mu i) (N.j i) σ * ‖f‖ ^ 2

theorem **ThetaGramPositiveScalar.eq_zero** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} {A : NAut N} (h : ThetaGramPositiveScalar N A) (i : N.Loc) (σ : what₂.FT) (f : what₂.Mj (N.j i)) (hf : f ∈ what₂.isot (N.j i) σ) (h0 : what₂.thetaT (N.mu i) (N.j i) f = 0) : f = 0

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.ThetaLiftUniqueness

theorem **eachRowHasThetaLift_iff** : EachRowHasThetaLift N A ↔ ∀ (i : N.Loc) (σ : what₂.FT), (∃ f ∈ what₂.isot (N.j i) σ, f ≠ 0) → ∃ P : A.AV i, (∃ f ∈ what₂.isot (N.j i) σ, what₂.thetaT (N.mu i) (N.j i) f ≠ 0) ∧ ∀ f ∈ what₂.isot (N.j i) σ, what₂.thetaT (N.mu i) (N.j i) f ∈ A.typeV i P

Development.AbelianVarietyTheory.BlockTheory.ThetaLiftGeometry.WeightedChainData

theorem **cpy_self** (E : Type) [Field E] (a : ℕ) (x : Sp E a) : cpy E a a x = x

theorem **hasWt_zero** (lam : D.what₂.Tau → Fin N → ℤ) : HasWt D frm lam 0

theorem **unfTie_of_liftTie** (Tr : TaylorLogMetric.Tor D) (e : TaylorLogMetric.FamE D) (h : ∀ i < D.R.m, GL3Theta.liftTie D Tr e T.O i) : T.UnfTie

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.BlochKatoGeneratorBound

@[simp] theorem **coe_latAct** (L : Lattice (O

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GaloisLatticeModuleActions.mk_mem_H0Mod** (L : _root_.Formalism.WeaklyAdmissibleFiltrations.Lattice (O

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GaloisLatticeModuleActions.finite_latMod** (L : _root_.Formalism.WeaklyAdmissibleFiltrations.Lattice (O

theorem **GeneratorBound.le** {PD : PeriodData G Qp K₀ K B BdR} {T : LocLat G Qp LL O} {ϖ : O} (h : GeneratorBound PD T ϖ) : GeneratedByAtMost (latticeFiniteCohomology PD T.L) ((Module.finrank Qp K + 1) * Module.finrank O ↥T.L.T)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.BlockRootsCuspidal

theorem **IsBlockRoots.le_one** {F : Type} [Monoid F] {g : ℕ} {B : Multiset (ℕ × ℕ)} {Rt : Multiset F} (h : IsBlockRoots g B Rt) (hR : Rt.Nodup) : ∀ p ∈ B, 0 < p.1 → p.2 ≤ 1

theorem **blockRootsAt_of_cuspidal** (D : Formalism.BasicNotions.D7) (σ : D.what₂.FT) (k : Fin 2) (v : D.nd.PlG k) (h : D.ntypes.cuspidal σ) : BlockRootsAt D σ k v

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.BottomMinor

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CornerMinors.nextPart_le** {l : ℕ} {a : Fin l → ℕ} (ha : Antitone a) (i : Fin l) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.CornerMinors.nextPart a i ≤ a i

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CornerMinors.hookDen_pos** {l : ℕ} {a : Fin l → ℕ} (ha : Antitone a) : 0 < _root_.GlobalTheory.IndefiniteRowSideFullHecke.CornerMinors.hookDen a

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.CartanCoshPower

theorem **pOf_herm** {p q : ℕ} (B : Matrix (Fin p) (Fin q) ℂ) : (pOf B).IsHermitian

theorem **cartanB_eq** {p q : ℕ} {g : UpqT p q} {B : Matrix (Fin p) (Fin q) ℂ} (h : IsCartan g B) : cartanB g = B

theorem **isCartan_cartanB** {p q : ℕ} (g : UpqT p q) : IsCartan g (cartanB g)

theorem **isCartan_iff** {p q : ℕ} (g : UpqT p q) (B : Matrix (Fin p) (Fin q) ℂ) : IsCartan g B ↔ cartanB g = B

theorem **coshPow_pos** {p q : ℕ} (c : ℝ) (g : UpqT p q) : 0 < coshPow c g

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.DoubleCosetDecomposition

theorem **finMat_one** : finMat σ J (1 : HA (AF F) σ J) = 1

theorem **finMat_mul** (g h : HA (AF F) σ J) : finMat σ J (g * h) = finMat σ J g * finMat σ J h

theorem **mem_stabAt** (K : Subgroup (HA (AF F) σ J)) (g γ : HA (AF F) σ J) : γ ∈ stabAt σ J K g ↔ γ ∈ ratPts σ J ∧ g⁻¹ * γ * g ∈ K

theorem **IsDoubleCosetReps.pos** {K : Subgroup (HA (AF F) σ J)} {n : ℕ} {g : Fin n → HA (AF F) σ J} (h : IsDoubleCosetReps σ J K g) : 0 < n

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.FactorSeriesSupport

theorem **atFactor_single** {ι : Type} {G : ι → Type} [∀ i, Group (G i)] (i : ι) (γ : G i) (c : ℂ) : atFactor i (MonoidAlgebra.single γ c) = MonoidAlgebra.single (Pi.mulSingle i γ) c

theorem **atFactor_central** {ι : Type} {G : ι → Type} [∀ i, Group (G i)] (i : ι) (z : MonoidAlgebra ℂ (G i)) (hz : ∀ b, z * b = b * z) : atFactor i z ∈ Subalgebra.center ℂ (MonoidAlgebra ℂ (∀ i, G i))

theorem **IsPowLabel.unique** {g : ℕ} [Fact g.Prime] {a : ℤ} {f f₁ f₂ : (ZMod g)[X]} (h₁ : IsPowLabel a f f₁) (h₂ : IsPowLabel a f f₂) (m₁ : f₁.Monic) (m₂ : f₂.Monic) : f₁ = f₂

theorem **rootExp_ne_zero** (N g : ℕ) (hg : 2 ≤ g) : rootExp N g ≠ 0

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.FlatZetaIntegral

theorem **flatIntegrand_self** (f : X → ℂ) (a : X → ℝ) (s₀ : ℂ) : flatIntegrand f a s₀ s₀ = f

theorem **norm_flatIntegrand** (f : X → ℂ) {a : X → ℝ} (s₀ s : ℂ) {x : X} (hx : 0 < a x) : ‖flatIntegrand f a s₀ s x‖ = ‖f x‖ * a x ^ (s.re - s₀.re)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.FockSingularValues

theorem **fockAct1_lower** {p q n : ℕ} (a : Fin p) (b : Fin q) (Φ : Pol p q n 0) : fockAct1 p q n 0 (Eab (Sum.inr b) (Sum.inl a)) Φ = -lowerOp n a b Φ

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.GL2CharacterZeros

theorem **det_uMat** (ν : Matrix ν ρ k) (ν' : Matrix ρ ν k) (b : Matrix ρ ρ k) : (uMat ν ν' b).det = 1

theorem **glueCols_mul_uMat** (ξ : Matrix W ρ k) (x₀ : Matrix W ν k) (c : Matrix W ρ k) (ν : Matrix ν ρ k) (ν' : Matrix ρ ν k) (b : Matrix ρ ρ k) : glueCols ξ x₀ c * uMat ν ν' b = glueCols ξ (x₀ + ξ * ν') (c + ξ * b + x₀ * ν)

theorem **om_uGL** [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [Fintype W] [DecidableEq W] (μ : kˣ →* ℂˣ) (ν : Matrix ν ρ k) (ν' : Matrix ρ ν k) (b : Matrix ρ ρ k) (Φ : Matrix W (ρ ⊕ ν ⊕ ρ) k → ℂ) (x : Matrix W (ρ ⊕ ν ⊕ ρ) k) : om μ (uGL ν ν' b) 1 Φ x = Φ (x * uMat ν ν' b)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.GLVolumeIndex

theorem **isUnit_of_red** {n : Type} [Fintype n] [DecidableEq n] (A : Matrix n n O) (h : IsUnit (red A)) : IsUnit A

theorem **red_mul** {l m n : Type} [Fintype m] (A : Matrix l m O) (B : Matrix m n O) : red (A * B) = red A * red B

theorem **red_add** {m n : Type} (A B : Matrix m n O) : red (A + B) = red A + red B

theorem **red_eq_zero** {m n : Type} (A : Matrix m n O) : red A = 0 ↔ ∀ i j, A i j ∈ IsLocalRing.maximalIdeal O

theorem **exists_mem_of_red** {n : Type} [Fintype n] [DecidableEq n] {J : Subgroup (GL n O)} (hJ : princ O n ≤ J) {k : GL n O} (hk : k ∈ J) {A : Matrix n n O} (h : red A = red (k : Matrix n n O)) : ∃ k' ∈ J, (k' : Matrix n n O) = A

theorem **blocks_mul** {a t : Type} [Fintype a] [Fintype t] (g h : Matrix (a ⊕ t) (a ⊕ t) O) : (g * h).toBlocks₁₁ = g.toBlocks₁₁ * h.toBlocks₁₁ + g.toBlocks₁₂ * h.toBlocks₂₁ ∧ (g * h).toBlocks₂₁ = g.toBlocks₂₁ * h.toBlocks₁₁ + g.toBlocks₂₂ * h.toBlocks₂₁ ∧ (g * h).toBlocks₂₂ = g.toBlocks₂₁ * h.toBlocks₁₂ + g.toBlocks₂₂ * h.toBlocks₂₂

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.GraphHeightLattice

theorem **mem_latMoved** {g : GLoc σ J v} {x : Fin n → Fv F v ⊗[F] E} : x ∈ latMoved σ J L v g ↔ (matHom (Fv F v) σ J g⁻¹).1 *ᵥ x ∈ latLoc F L v

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.HermitianLattices

theorem **cj_mem** (L : LocE F E) {x : E} (hx : x ∈ L.O) : L.cj x ∈ L.O

theorem **isHermND_one** (L : LocE F E) (n : ℕ) : IsHermND L (1 : Matrix (Fin n) (Fin n) E)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.IndexAtLevel

theorem **boundFine_le_boundCoarse** (w : D.what₂.places.Pl) : boundFine_vp D u w ≤ boundCoarse_vp D u w

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.IntegralCoordinatesNorm

theorem **IntUnder.mono** {H H' : D7 → Prop} {A A' : ∀ D : D7, Set (Ideal (𝒪 D.host.K))} (hH : ∀ D, H' D → H D) (hA : ∀ D, A' D ⊆ A D) (h : IntUnder H A) : IntUnder H' A'

theorem **ArchUnder.mono** {H H' : D7 → Prop} (hH : ∀ D, H' D → H D) (h : ArchUnder H) : ArchUnder H'

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.IwahoriSubgroup

theorem **mem_iwahoriInt_iff** (F : Type) [Field F] [Valued F (WithZero (Multiplicative ℤ))] (r : ℕ) (g : GL (Fin r) (Int F)) : g ∈ iwahoriInt F r ↔ ∀ i j : Fin r, j < i → Valued.v (((g : Matrix (Fin r) (Fin r) (Int F)) i j : Int F) : F) < (1 : WithZero (Multiplicative ℤ))

theorem **self_mem_iwahoriDoubleCoset** {F : Type} [Field F] [Valued F (WithZero (Multiplicative ℤ))] {r : ℕ} (t : Matrix (Fin r) (Fin r) F) : t ∈ iwahoriDoubleCoset t

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.LatticeConditions

theorem **fracMap_alg** (c : O₀) : fracMap f hf (algebraMap O₀ (FractionRing O₀) c) = algebraMap O' (FractionRing O') (f c)

theorem **toOk_inj** (i : tw.Tw) (k : ℕ) : Function.Injective (toOk tw i k)

theorem **twO_inj** (a : N.Loc) : Function.Injective (tw.twist₂ a)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.LevelCarrying

theorem **ActionLaws.comap** (h : ActionLaws R) {G' : Type} [Group G'] (f : G' →* G) : ActionLaws (fun g => R (f g) : RelFam Xo c G')

theorem **ActionLaws.exists_mulAction** (h : ActionLaws R) (S : Type) [CommRing S] [Algebra c.F S] : ∃ _ : MulAction Gᵐᵒᵖ (FrPt Xo c S), ∀ (g : G) (y y' : FrPt Xo c S), R g S y y' ↔ y' = MulOpposite.op g • y

theorem **levelActionSt_of_principal** (h : PrincipalLevelAction) (D : D7) : LevelActionSt D

theorem **LevelActionAt.matrixGroup** {T : Core D} {j : ℕ} (h : LevelActionAt T j) (f : G0g D.nd →* LevelAut (T.Q.Dw j) T.Q.l) : ActionLaws (fun g => levelRel (T.Q.Xo j) T.Q.c (f g) : RelFam (T.Q.Xo j) T.Q.c (G0g D.nd)) ∧ ∀ c' : Coeff T.Q.l, T.Q.c.F ≤ c'.F → ActionLaws (fun g => levelRel (T.Q.Xo j) c' (f g) : RelFam (T.Q.Xo j) c' (G0g D.nd))

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.LocalIntegralGL

theorem **weilPair_one_right** (μ η : Fˣ →* ℂˣ) (nrm : AbsoluteValue F ℝ) (g : GL (Fin r) F) (Φ : Matrix (Fin r ⊕ Fin t) (Fin r) F → ℂ) : weilPair μ η nrm g 1 Φ = omegaS (μ * η) nrm g Φ

theorem **blockChar_apply** (μ : Fˣ →* ℂˣ) (P : Subgroup (GL (Fin r) F)) (lam : P →* ℂˣ) (k : P) : blockChar μ P lam k = lam k * (μ (GeneralLinearGroup.det (k : GL (Fin r) F)))⁻¹

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.PolarLatticePairs

theorem **mul_entry_mem** (D : D7) {N : ℕ} {A B : Matrix (Fin N) (Fin N) (AE D)} (hA : ∀ i j, A i j ∈ OhE D) (hB : ∀ i j, B i j ∈ OhE D) (i j : Fin N) : (A * B) i j ∈ OhE D

theorem **idHat_mul** (D : D7) {q : Ideal (𝒪 D.host.E)} {y r : AE D} (hy : y ∈ idHat D q) (hr : r ∈ OhE D) : y * r ∈ idHat D q

theorem **BlockMod.mul** (D : D7) {N : ℕ} {q : Ideal (𝒪 D.host.E)} {I : Finset (Fin N)} {A B : Matrix (Fin N) (Fin N) (AE D)} (hA : ∀ i j, A i j ∈ OhE D) (hB : ∀ i j, B i j ∈ OhE D) (h₁ : BlockMod D q I A) (h₂ : BlockMod D q I B) : BlockMod D q I (A * B)

theorem **BlockMod.one** (D : D7) {N : ℕ} (q : Ideal (𝒪 D.host.E)) (I : Finset (Fin N)) : BlockMod D q I 1

theorem **levelJ_le** (D : D7) {N : ℕ} (q : Ideal (𝒪 D.host.E)) (I : Finset (Fin N)) : levelJ D q I ≤ intPts D N

theorem **levelJ_top** (D : D7) {N : ℕ} (I : Finset (Fin N)) : levelJ D ⊤ I = intPts D N

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.RationalFormValues

theorem **ValuesIn.reps** {R : RealDat D A Ks} {i : Fin D.R.m} {t : T} {K' : Type} [Field K'] {σ' : K' →+* ℂ} {c' : VV K' (lamV D i) →ₗ[σ'] (R.cV i).W} (H : ValuesIn R i t c') {J : Subgroup (Bridge.Gf (Vc D i))} {Cl : Type} {y : Cl → Bridge.Gf (Vc D i)} (_hy : Bridge.IsReps (Bridge.ratF (Vc D i)) J y) (x : A.fam.amb.FL i t) : ∃ val : Cl → VV K' (lamV D i), ∀ k, c' (val k) = formFn R i t x (y k)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.ReductionClauses

theorem **ClauseA.cusp** {Sig : List (Part A.CG)} (h : ClauseA N A Sig) : IsCuspIso N A Sig

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.RootsOfUnityDivisibility

theorem **vw_one'** (w : Pl K) : vw w (1 : K) = 0

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.SelfDualLatticeMass

theorem **SelfDualMass.all_lattices** (h : SelfDualMass) (D : D7) : letI

theorem **stdForm_apply** (x y : Fin M → E) : stdForm E M x y = ∑ i, star (x i) * y i

theorem **mem_stdLat_iff** (x : Fin M → E) : x ∈ stdLat E M ↔ ∀ i, IsIntegral ℤ (x i)

theorem **mem_one_iff** (a : E) : a ∈ (1 : Submodule (𝒪 E) E) ↔ IsIntegral ℤ a

theorem **stdSpace_posDef** (hc : ∀ (τ : E →+* ℂ) (a : E), τ (star a) = starRingEnd ℂ (τ a)) : PosDefAtRealPlaces (stdSpace E M)

theorem **stdSpace_case** (D : D7) (M : ℕ) : letI

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.SelmerRankVanishing

theorem **SelmerBoundFromLocalInputs.full** {D : D7} (h : SelmerBoundFromLocalInputs D) : SelmerBoundWithInertiaCondition D

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.SelmerWindowSums

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelmerFiniteUnramified.finKs** (K : _root_.Formalism.BasicNotions.Loc.FinExt l) (s : ℕ) : FiniteDimensional ℚ_[l] ↥(_root_.Formalism.BasicNotions.Loc.Ks K s)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.SelmerFiniteUnramified.cyc_mem** (P : _root_.Formalism.BasicNotions.LocPrime l) (K LL : _root_.Formalism.BasicNotions.Loc.FinExt l) (g : ↥(_root_.Formalism.BasicNotions.Loc.GK K)) : ((_root_.Formalism.MainNotions.N.cycLL (P.PD K) ↥LL.1 g : (↥LL.1)ˣ) : ↥LL.1) ∈ _root_.Formalism.BasicNotions.Loc.OL LL

theorem **negTwist_mem** (P : LocPrime l) (K LL : FinExt l) (e : ℕ) (g : ↥(GK K)) : ((negTwist P K LL e g : (↥LL.1)ˣ) : ↥LL.1) ∈ OL LL

theorem **exists_pow_mul_pred_ge** (l : ℕ) [Fact l.Prime] (H : ℕ) : ∃ k : ℕ, H ≤ l ^ k * (l - 1)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.SingularValueFrame

theorem **one_le_coshProd** (X : Matrix (Fin n) (Fin n) ℂ) : 1 ≤ coshProd ε X

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.SplitColumnClasses

theorem **splitsCompletely_g** (D : D7) (i : Fin 2) : SplitsCompletely D.host.E (D.host.g i)

theorem **card_colClassSet** (D : D7) (L : LevelData D) (u : D.what₂.Word) : Nat.card (ColClassSet D L u) = Nat.card ((D.mem u).C.cls 0)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.StandardHermitianSpace

theorem **rank_eq** (i : Fin D.R.m) : (Vc D i).rank = D.R.nV

theorem **baseLat_eq** {T : Type} {L : LevCh D T} {μ0 : T} (hL : L.Is μ0) (i : Fin D.R.m) : (L.ΛV i : Set (Fin (Vc D i).rank → D.host.E)) = L0c D (Vc D i)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.SublevelSplitting

theorem **RowAt.plain** {R : RealDat D A Ks} (h : RowAt R) : RowPlainAt R

theorem **IsSubLev.vol_pos** {V : HSp D} {J J' : Subgroup (Bridge.Gf V)} {vol : ℝ} (h : IsSubLev V J J' vol) : 0 < vol ∧ vol ≤ 1

theorem **IsAdReps.pos** {V : HSp D} {Λ : Lat V} {J' : Subgroup (Bridge.Gf V)} {bad n : ℕ} {h : Fin n → Bridge.Gf V} {Lh : Fin n → Lat V} {gam : Fin n → ℕ} (H : IsAdReps V Λ J' bad h Lh gam) : (∀ a, 0 < gam a) ∧ 0 < n

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.TateLocalDualityTopDegree

theorem **norm_chi_le_one** (P : LocPrime l) (K : FinExt l) (g : ↥(GK K)) : ‖(((P.PD K).χ g : ℚ_[l]ˣ) : ℚ_[l])‖ ≤ 1

theorem **fracModIntTwist_mk** (P : LocPrime l) (K LL : FinExt l) (g : ↥(GK K)) (x : ↥LL.1) : fracModIntTwist P K LL g (FracModInt.mk LL x) = FracModInt.mk LL (algebraMap ℚ_[l] ↥LL.1 (((P.PD K).χ g : ℚ_[l]ˣ) : ℚ_[l]) * x)

theorem **TateLocalDualityDegreeTwoAll.tateDualityTop_of_all** {D : D7} (h : TateLocalDualityDegreeTwoAll D) : TateLocalDualityDegreeTwoGood D

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.ThetaLift

theorem **N_le_succ** (i : ℕ) : D.R.N i ≤ D.R.N (i + 1)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.ToricLatticeDenominators

theorem **vK_one** (l : ℕ) : vK R l 1 = 0

theorem **isIdx_self** (l : ℕ) (V : Submodule (FractionRing R) W) (L : Submodule R W) (h : IsLatIn V L) : IsIdx l V L L 0

theorem **thetaPsi_one** : ThetaPsi Y i j b O' k ∅ 1 = ThetaExt Y i j b O'

theorem **projectionDerivative_of_exportAt** (N0 : X.Tw → ℕ) (i : X.Tw) (h : exportAt bkP N0 i) : ProjectionDerivativeEstimate bkP N0 i

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.TrivialOfDensityOne

theorem **valueOneAt_of_trivAt** {χ : Ideles L →* ℂˣ} {P : HeightOneSpectrum (𝒪 L)} (h : TrivAt χ P) : ValueOneAt χ P

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.TwistCoverDesign

theorem **AA_ne_zero** (D : D7) : D.host.AA ≠ 0

theorem **IsCoverHalf.dvd** {H : ModHost D} {l l₂ : Level H} {Q : Finset (Place D)} {n : ℕ} {I : Ideal (𝒪 D.host.E)} (h : IsCoverHalf l Q n I l₂) : l.Dvd l₂

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.TwistedTriples

theorem **badS_eq** (a : D.nd.Loc) : badS D a = ↑(D.nd.S a)

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.VolumePerpendicular

theorem **IsVol.unique** (D : D7) : letI

theorem **volumeComparison_of_ratio** (D : D7) (h : VolumeRatio D) (ht : VolTie D) : Conclusion2 D

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.WeakIdealLattices

theorem **exists_pow_ge** (l : ℕ) [Fact l.Prime] (H₀ : ℕ) : ∃ n : ℕ, H₀ ≤ l ^ n ∗ (l − 1)

theorem **finiteDimensional_Ks** (K : FinExt l) (s : ℕ) : FiniteDimensional ℚ_[l] ↥(Ks K s)

theorem **inf_inf_eq_of_inf_eq** {α : Type} [SemilatticeInf α] {a b c d : α} (h : a ⊓ c = a ⊓ d) : a ⊓ b ⊓ c = a ⊓ b ⊓ d

theorem **isJumpAll_of_isJumpV** (P : LocPrime l) (T : Lat K LL) (τ : ↥K.1 →ₐ[ℚ_[l]] ↥LL.1) (u : ℤ) (h : IsJumpV (P.PD K) T.ρ τ u) : IsJumpAll P T u

theorem **mem_jumpsAt** {s : ℕ} {hK : K.1 ≤ Ks K s} {M₀ h₀ : ℕ} {T : Lat K LL} (hW : HypW AI s hK M₀ h₀ T) (τ : ↥K.1 →ₐ[ℚ_[l]] ↥LL.1) (u : ℤ) : u ∈ jumpsAt P T τ M₀ h₀ ↔ IsJumpV (P.PD K) T.ρ τ u

theorem **InvariantsKilledAllCoeff.good** {D : D7} (h : InvariantsKilledAllCoeff D) : InvariantsKilled D

theorem **nZero_eq_zero** (l : ℕ) [Fact l.Prime] (H₀ : ℕ) (h : H₀ ≤ l − 1) : nZero l H₀ = 0

theorem **piS_mem** {P : LocPrime l} (AI : LocAinf l P) (K : FinExt l) (s : ℕ) : piS AI s ∈ ringAs AI K s

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.WeightFlagWeyl

theorem **IsFrameExtremalUnit.extremal** {ε : Fin n → Bool} {lam : Fin n → ℝ} {σ : Matrix (Fin n) (Fin n) ℂ →ₗ[ℂ] H} {w : H} (h : IsFrameExtremalUnit ε lam σ w) : IsExtremalUnit ε lam σ w

Development.AbelianVarietyTheory.BlockTheory.TorsionBlocks.WittIndexParallel

theorem **nu_eq** (m j : ℕ) : nu m j = (m − j) / 2

theorem **parallelSig_of_sigOK** {D : D7} {V : HSp D} {p q : ℕ} (h : V.SigOK) (hs : ∀ τ, V.sig τ = (p, q)) : ParallelSig V p q

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.AdjointIsotropicDecomposition

theorem **isIsoSplit_three** (D : D7) : IsIsoSplit (spaceThree D) (spaceOne D) (r

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.AdjointRadical

theorem **mul_mem_lieRadical** {A X : Matrix (Fin r) (Fin r) k} (hA : A.BlockTriangular b) (hX : X ∈ lieRadical k b) : A ∗ X ∈ lieRadical k b

theorem **mem_lieRadical_mul** {A X : Matrix (Fin r) (Fin r) k} (hX : X ∈ lieRadical k b) (hA : A.BlockTriangular b) : X ∗ A ∈ lieRadical k b

theorem **blockTriangular_inv** {α : Type} [LinearOrder α] {c : Fin r → α} {g : GL (Fin r) k} (hg : (g : Matrix (Fin r) (Fin r) k).BlockTriangular c) : ((g⁻¹ : GL (Fin r) k) : Matrix (Fin r) (Fin r) k).BlockTriangular c

theorem **blockTriangular_of_sub_one** {A : Matrix (Fin r) (Fin r) k} (h : A − 1 ∈ lieRadical k b) : A.BlockTriangular b

theorem **levi_blockTriangular** (δ : levi k b) : ((δ : GL (Fin r) k) : Matrix (Fin r) (Fin r) k).BlockTriangular b

theorem **conj_mem_radical** (δ : levi k b) {n : GL (Fin r) k} (hn : n ∈ radical k b) : (δ : GL (Fin r) k)⁻¹ ∗ n ∗ δ ∈ radical k b

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.BarMatrix

theorem **reindex_form** {p q n : ℕ} {ε : Fin n → Bool} {e : Ix p q ≃ Fin n} (h : SignCompat ε e) : Matrix.reindex e e (Formalism.UG.Jpq p q) = Formalism.REAL.Jm ε

theorem **plainMat_mem** {p q n : ℕ} {ε : Fin n → Bool} {e : Ix p q ≃ Fin n} (h : SignCompat ε e) {g : Mat p q} (hg : gᴴ ∗ Formalism.UG.Jpq p q ∗ g = Formalism.UG.Jpq p q) : plainMat e g ∈ Formalism.MainNotions.S.UJset ε

theorem **barMat_mem** {p q n : ℕ} {ε : Fin n → Bool} {e : Ix p q ≃ Fin n} (h : SignCompat ε e) {g : Mat p q} (hg : gᴴ ∗ Formalism.UG.Jpq p q ∗ g = Formalism.UG.Jpq p q) : barMat e g ∈ Formalism.MainNotions.S.UJset ε

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.BoundaryVanishingPullback

theorem **continuous_move** (γ : GL (Fin a) k × GL (Fin b) k) : Continuous (move γ)

theorem **move_one** (x : Matrix (Fin b) (Fin a) k) : move 1 x = x

theorem **move_mul** (γ δ : GL (Fin a) k × GL (Fin b) k) (x : Matrix (Fin b) (Fin a) k) : move (γ * δ) x = move δ (move γ x)

theorem **rank_move** (γ : GL (Fin a) k × GL (Fin b) k) (x : Matrix (Fin b) (Fin a) k) : (move γ x).rank = x.rank

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.BoundedCommutantVacua

theorem **VacuumDense.model** (h : VacuumDense ψ V W μ ℱ) : VacuumDenseModel ψ V W μ ℱ

theorem **exists_isVac_ne_zero** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {sV : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ} {sW : D.what₂.Tau → Matrix (Fin W.rank) (Fin W.rank) ℂ} (hV : ∀ τ, IsCartanInv (gramAt D V τ) (sV τ)) (hW : ∀ τ, IsCartanInv (gramAt D W τ) (sW τ)) : ∃ f, IsVac ψ V W θ sV sW f ∧ f ≠ 0

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.CentralizerTensorForms

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.adj_eq_iff** (σ : A →+* A) {J : Matrix n n A} (hJ : IsUnit J.det) (l m : Matrix n n A) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.adj σ J l = m ↔ _root_.Formalism.UG.cT σ l * J = J * m

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.hform_eq** (σ : A →+* A) (J : Matrix n n A) (x y : n → A) : _root_.Formalism.UG.hform σ J x y = (fun i => σ (x i)) ·ᵥ (J *ᵥ y)

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.form_adj** (σ : A →+* A) {J : Matrix n n A} (hJ : IsUnit J.det) (l : Matrix n n A) (y y' : n → A) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.form σ J y (l *ᵥ y') = _root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.form σ J (_root_.GlobalTheory.IndefiniteRowSideFullHecke.CentralizerTensorForms.adj σ J l *ᵥ y) y'

theorem **isQuadInvol_complex** : IsQuadInvol ℝ ℂ (Complex.conjAe : ℂ →ₐ[ℝ] ℂ) where

theorem **isRegular_exElt** : IsRegular (Complex.conjAe : ℂ →ₐ[ℝ] ℂ) (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) ℂ) exElt where

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ChainMatrices

theorem **idx_lt** {j : ℕ} (hj : j < D.R.m) (a : Fin (D.sp.Y j).rank) : ((a : ℕ) < D.R.k → (a : ℕ) < (D.sp.X j).rank) ∧ (¬ (a : ℕ) < D.R.k + nAux D j → (a : ℕ) - nAux D j < (D.sp.X j).rank)

theorem **isAuxiliaryHyperbolicPart_initialChainSpace** : IsAuxiliaryHeckeCharacter D 0 (fun i => if h : i < D.R.rS then D.sp.hypE ⟨i, h⟩ else 0) (fun i => if h : i < D.R.rS then D.sp.hypF ⟨i, h⟩ else 0) D.sp.zm

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.CharacterPartsGaps

theorem **PartsMatch.char** (h : PartsMatch) : PartsMatchChar

theorem **mem_finS** (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) : v ∈ finS D ↔ ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ClassFunctionSumIntegral

theorem **map_conj_inv** {G H : Type} [Group G] [Group H] (f : G →* H) (a b : G) : f (a⁻¹ * b * a) = (f a)⁻¹ * f b * f a

theorem **mem_levelFin** (w : GA D V) : (finHom V).rangeRestrict w ∈ levelFin V K ↔ Lev.finE V w ∈ K

theorem **mem_classGroupAt** (z : GA D V) (γ : GQ V) : γ ∈ classGroupAt V K ((finHom V).rangeRestrict z) ↔ Lev.finE V (z⁻¹ * (γ : GA D V) * z) ∈ K

theorem **exists_lift** (x : Bridge.Gf V) : ∃ z : GA D V, (finHom V).rangeRestrict z = x

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.CommutingRationalGroups

theorem **algW_eC** (i : Fin D.R.m) : algW R i (RealDat.eC (A

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.CompatibleThetaRestrictions

theorem **isWittLower_iff** (B B' : HSp D) (t : ℕ) : IsWittLower B B' t ↔ ∃ ι C, IsWittStep B B' t ι C

theorem **exists_isWittStep_iff** (B : HSp D) (t : ℕ) : (∃ B' ι C, IsWittStep B B' t ι C ∧ B'.SigOK) ↔ t ≤ GlobalTheory.Cusp.wittIndex B

theorem **IsWittStep.isIsoFrame** {B B' : HSp D} {t : ℕ} {ι : B'.V →ₗ[D.host.E] B.V} {C : Matrix (Fin B.rank) (Fin t) D.host.E} (h : IsWittStep B B' t ι C) (ht : 1 ≤ t) : IsIsoFrame B C

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ConjugateAdjointDagger

theorem **cj_cj** {X Y : Type} (e : Y → X) (e' : X → Y) (h : ∀ x, e (e' x) = x) (f : X → ℂ) : cj e' (cj e f) = f

theorem **cjEnd_cj** {X Y : Type} (e : Y ≃ X) (L : Module.End ℂ (X → ℂ)) (f : X → ℂ) : cjEnd e L (cj e f) = cj e (L f)

theorem **dagger_dagger** (S T : HSp D) (x : WO S T) : dagger T S (dagger S T x) = x

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ContractionTensorModels

theorem **contrTensorModels_of** (h : ContrTensor) : ContrTensorModels

theorem **contrProps_of** {H₁ H₂ FF : Type} [NormedAddCommGroup H₁] [InnerProductSpace ℂ H₁] [CompleteSpace H₁] [NormedAddCommGroup H₂] [InnerProductSpace ℂ H₂] [CompleteSpace H₂] [NormedAddCommGroup FF] [InnerProductSpace ℂ FF] {G₁ G₂ : Type} [Group G₁] [Group G₂] {π₁ : G₁ →* (H₁ ≃ₗᵢ[ℂ] H₁)} {π₂ : G₂ →* (H₂ ≃ₗᵢ[ℂ] H₂)} {ρ : G₁ × G₂ →* (FF ≃ₗᵢ[ℂ] FF)} {Tm : H₁ →ₗ[ℂ] H₂ →ₗ[ℂ] FF} (hI : IsIsoBilin Tm) (hE : IsEquivBilin π₁ π₂ ρ Tm) {It : FF →ₗ[ℂ] H₂ →ₛₗ[starRingEnd ℂ] H₁} (hC : IsContr Tm It) : ContrProps π₁ π₂ ρ Tm It

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.CuspAlongSmoothRows

theorem **LiftsMult.exists_linear** {θ : InfFam D} {A B : Node D} {P : Type} {m : A.MeasDat} {K : P → GA D B.V → GA D A.V → ℂ} {It : P → SmT A.M → SmT B.M} (H : LiftsMult θ m K It) : ∃ io : A.Cusp θ →ₗ[ℂ] B.Cusp θ, ∀ (F : A.Cusp θ) p v h, IntegrableOn (fun g => K p h g * F.1 v g) m.fd m.ν ∧ ∫ g in m.fd, K p h g * F.1 v g ∂m.ν = (io F).1 (It p v) h

theorem **cuspAlong_iff_hasLifts** (G : GridPre R) : CuspAlong G ↔ G.HasLifts

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.CuspidalNoRadicalFixed

theorem **containsTypes_finRes** {H X W : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] [NormedAddCommGroup X] [InnerProductSpace ℂ X] [AddCommGroup W] [Module ℂ W] {σ : Representation ℂ (GfC V) W} {δ : GinfC V →* (X →L[ℂ] X)} {π : GA D V →* (H →L[ℂ] H)} {U : Submodule ℂ H} (h : IsCopyFinInf V σ δ π U) {T : TypeDataG V} (hT : ContainsTypes T σ) : ContainsTypes T (finRes π)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.DefiniteIndexDimensionCounts

theorem **GoodShare.ne_bot** {δ : ℝ} {b : Bool} {i : Fin D.R.m} (h : GoodShare R F δ b i) : R.Wat i (sg0 D A) (levB R F b) ≠ ⊥

theorem **exists_last** (D : D7) : ∃ i : Fin D.R.m, (i : ℕ) + 1 = D.R.m

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.DefiniteUnitaryAction

theorem **wt_ne_zero** (c : S.Cl) : (S.wt c : ℂ) ≠ 0

theorem **pairAt_smul_left** (a : K) (x y : S.V) : pairAt S σ (a • x) y = starRingEnd ℂ (σ a) * pairAt S σ x y

theorem **pairAt_smul_right** (a : K) (x y : S.V) : pairAt S σ x (a • y) = σ a * pairAt S σ x y

theorem **pairAt_eq** (h : ∀ a, σ (star a) = starRingEnd ℂ (σ a)) (x y : S.V) : pairAt S σ x y = σ (S.pair x y)

theorem **classSumAt_eq** (h : ∀ a, σ (star a) = starRingEnd ℂ (σ a)) (f f' : S.Forms) : classSumAt S σ f f' = σ (S.csum f f')

theorem **isUnitaryAt_iff** (h : ∀ a, σ (star a) = starRingEnd ℂ (σ a)) {G : Type} [Group G] (r : Representation K G S.Forms) : IsUnitaryAt S σ r ↔ ∀ (g : G) (f f' : S.Forms), S.csum (r g f : S.Cl → S.V) (r g f' : S.Cl → S.V) = S.csum (f : S.Cl → S.V) (f' : S.Cl → S.V)

theorem **IsUnitaryRep.tw** {G H : Type} [Group G] [NormedAddCommGroup H] [InnerProductSpace ℂ H] {ρ : Representation ℂ G H} (h : IsUnitaryRep ρ) (s : Bool) : IsUnitaryRep (Tw.rep s ρ)

theorem **DefiniteUnitary.csum** (h : DefiniteUnitary A) : (∀ t g (f f' : (A.MtildeCSD t).Forms), (A.MtildeCSD t).csum (A.leftGroupActionRight t g f : _ → _) (A.leftGroupActionRight t g f' : _ → _) = (A.MtildeCSD t).csum (f : _ → _) (f' : _ → _)) ∧ ∀ i t g (f f' : (A.MtCSD i t).Forms), (A.MtCSD i t).csum (A.leftGroupActionLift i t g f : _ → _) (A.leftGroupActionLift i t g f' : _ → _) = (A.MtCSD i t).csum (f : _ → _) (f' : _ → _)

theorem **rows_iff** (R : RealDat D A Ks) (F : Face D A) (h : R.Char F) : IsUnitaryRep R.gE ↔ ∀ (g : G), ∀ x ∈ R.EEat R.triv, ∀ y ∈ R.EEat R.triv, Formalism.IndefR.Mult.ipM (R.MY 0) R.ν0 R.fd0 (Formalism.IndefR.Mult.postOn _ (actL (Yc D 0) (R.βY 0) (R.dat0 R.triv) (R.eG g)) x) (Formalism.IndefR.Mult.postOn _ (actL (Yc D 0) (R.βY 0) (R.dat0 R.triv) (R.eG g)) y) = Formalism.IndefR.Mult.ipM (R.MY 0) R.ν0 R.fd0 x y

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.DepthZeroGoodLines

theorem **superSum_mul** (u v : unipB F (Fin n)) : superSum ((u * v : unipB F (Fin n)) : GL (Fin n) F).val = superSum (u : GL (Fin n) F).val + superSum (v : GL (Fin n) F).val

theorem **isDatum_detRep** (r : ℕ) {χ χW χV : kˣ →* ℂˣ} (h : IsUnram χ) (hu : IsUnitaryChar χ) (hW : IsUnitaryChar χW) (hV : IsUnitaryChar χV) (ht : IsTame χW) (hr : IsUnram (χW * χV⁻¹)) : IsDatum (detRep (Fin r) χ) χW χV

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.DownUpLambdaOperators

theorem **typeIdeal_ne** (D : D7) : Ideal.span {((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F)} ≠ 0

theorem **mem_typePlaces** (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) : v ∈ typePlaces D ↔ ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal

theorem **Concl.scalar_pos** {R : RealDat D A Ks} (h : Concl R) (t : T) (c : Eig D) : ∃ l : ℝ, 0 < l ∧ ∀ x ∈ partE R c, dnUp R t x = (l : ℂ) • x

theorem **Concl.of_match** {R : RealDat D A Ks} (h : Concl R) (hm : ∀ c, partE R c ≠ ⊥ → ∀ i, partW R i c ≠ ⊥) : ∃ κ : ℝ, 0 < κ ∧ ∀ lam, IsLam R lam → Lift.IsKap D.R.m (dnUp R) (partE R) lam κ

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.DualLiftInvariance

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.mem_lastKer** {L : Submodule 𝒪[k] (n'' → k)} {y : n'' ⊕ Fin 1 → k} : y ∈ _root_.GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.lastKer L ↔ y ∘ Sum.inl ∈ L ∧ y (Sum.inr 0) ∈ 𝒪[k]

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.lastKer_eq** (q : _root_.GlobalTheory.LatPair.Unif k) (L : Submodule 𝒪[k] (n'' → k)) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.lastKer L = _root_.GlobalTheory.LatPair.gluedLat q 1 2 L 0

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.isGlue_last** (q : _root_.GlobalTheory.LatPair.Unif k) (L : Submodule 𝒪[k] (n'' → k)) [L.IsLattice k] : _root_.GlobalTheory.LatPair.IsGlue (_root_.GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.lastKer L) (q.inv • _root_.GlobalTheory.LatPair.avec k n'' 1 1)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.EllipticPartTestData

theorem **eval_alpha** : eval₂ (algebraMap E (Fld P)) (alpha P) P.1 = 0

theorem **alpha_ne_zero** : alpha P ≠ 0

theorem **eval_inv_alpha** : eval₂ (((IsScalarTower.toAlgHom F E (Fld P)).comp σ : E →ₐ[F] Fld P) : E →+* Fld P) (alpha P)⁻¹ P.1 = 0

theorem **star'_alpha** : star' P (alpha P) = (alpha P)⁻¹

theorem **star'_algebraMap** (e : E) : star' P (algebraMap E (Fld P) e) = algebraMap E (Fld P) (σ e)

theorem **mulMat_mulVec** (φ : (k ⊗[F] Fld P) ≃ₗ[k ⊗[F] E] (Fin n → k ⊗[F] Fld P)) (z y : k ⊗[F] Fld P) : mulMat φ z *ᵥ φ y = φ (z * y)

theorem **torUnit_norm** (t : Tor P k) : (torUnit t).1 * sigR k (star' P) (torUnit t).1 = 1

theorem **unitary_of_hform** {A : Type} [CommRing A] (s : A →+* A) (G M : Matrix (Fin n) (Fin n) A) (h : ∀ a b, Formalism.UG.hform s G (M *ᵥ a) (M *ᵥ b) = Formalism.UG.hform s G a b) : Formalism.UG.cT s M * G * M = G

theorem **glChar_perm** (μ : Fin r → ℤ) (a : Fin r → ℂ) (s : Equiv.Perm (Fin r)) : glChar μ (a ∘ s) = glChar μ a

theorem **exists_roots** {Q : ℂ[X]} (hm : Q.Monic) (hd : Q.natDegree = r) : ∃ a : Fin r → ℂ, Q = ∏ i, (X - C (a i))

theorem **cxRoots_spec** (τ : E →+* ℂ) (P : PPol σ r) : P.1.map τ = ∏ i, (X - C (cxRoots τ P i))

theorem **exists_isTorusMeas** (P : PPol σ r) : ∃ dta dtf dt, IsTorusMeas P dta dtf dt

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ExponentSeparation

theorem **IsExp.unique** {r : ℕ} {V : Type} [AddCommGroup V] [Module ℂ V] {π : Representation ℂ (GL (Fin r) k) V} {E E' : Multiset ((Fin r → kˣ) →* ℂˣ)} (h : IsExp π E) (h' : IsExp π E') : E = E'

theorem **IsExp.unramified** {r : ℕ} {V : Type} [AddCommGroup V] [Module ℂ V] {π : Representation ℂ (GL (Fin r) k) V} {E : Multiset ((Fin r → kˣ) →* ℂˣ)} (h : IsExp π E) {χ : (Fin r → kˣ) →* ℂˣ} (hχ : χ ∈ E) {t : Fin r → kˣ} (ht : t ∈ torusO k (Fin r)) : χ t = 1

theorem **fixed_detRep_ne_bot** {r : ℕ} {χ : kˣ →* ℂˣ} (h : IsUnram χ) : fixed (detRep (Fin r) χ) (iwahori k (Fin r)) ≠ ⊥

theorem **isAdmissible_detRep** {r : ℕ} {χ : kˣ →* ℂˣ} (h : IsUnram χ) : IsAdmissible (detRep (Fin r) χ)

theorem **isExp_detRep** {r : ℕ} {χ : kˣ →* ℂˣ} (h : IsUnram χ) : IsExp (detRep (Fin r) χ) {torusChar fun _ => χ}

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.FourierJacobiAverage

theorem **app_coe** (l : SB X →ₗ[ℂ] V) (f : SB X) : app l (f : X → ℂ) = l f

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.FractionalBaseInclusion

theorem **algebraMap_ext_injective** (c : NChoice D) (w : D.what₂.places.Pl) : Function.Injective (algebraMap (D.ntw.O (c.twI w)) (c.Oe w))

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.GaussianBinomialClosedForm

theorem **secVal_mul** {q : ℕ} (hq : q ≠ 0) (χ : Lˣ →* ℂˣ) (t : ℂ) (a b : Lˣ) : secVal q χ t (a * b) = secVal q χ t a * secVal q χ t b

theorem **secVal_unit** (q : ℕ) {χ : Lˣ →* ℂˣ} (hχ : IsUnr χ) (t : ℂ) {a : Lˣ} (ha : Valued.v (a : L) = 1) : secVal q χ t a = 1

theorem **v_aD_le** {σ : L →+* L} {n : ℕ} (p : PD σ n) (hp : (p : GD σ n) ∈ KD σ n) : Valued.v (aD σ n p : L) ≤ 1

theorem **v_aD_eq** {σ : L →+* L} {n : ℕ} (p : PD σ n) (hp : (p : GD σ n) ∈ KD σ n) : Valued.v (aD σ n p : L) = 1

theorem **exists_isSphSec** (σ : L →+* L) {q : ℕ} (hq : q ≠ 0) (n : ℕ) {χ : Lˣ →* ℂˣ} (hχ : IsUnr χ) (s : ℂ) : ∃ Φ, IsSphSec σ q n χ s Φ

theorem **isZonal_of_isSatVecU** {ϖ : L} {σ : L →+* L} {q n : ℕ} (hU : IsUnramInv ϖ σ q) {V : Type} [AddCommGroup V] [Module ℂ V] {ρ : Representation ℂ (GU σ n) V} {x : V} {α : Fin (n / 2) → ℂˣ} (hx : GlobalTheory.LFun.IsSatVecU ϖ q ρ x α) (l : V →ₗ[ℂ] ℂ) (hl : ∀ k ∈ KU σ n, ∀ y, l (ρ k y) = l y) (h1 : l x = 1) : IsZonal ϖ q α fun g => l (ρ g x)

theorem **isZonal_one** [CompleteSpace L] {ϖ : L} {σ : L →+* L} {q : ℕ} (hU : IsUnramInv ϖ σ q) (n : ℕ) : ∃ α : Fin (n / 2) → ℂˣ, (∀ i, (α i : ℂ) = (q : ℂ) ^ ((n : ℤ) - 1 - 2 * (i.val : ℤ))) ∧ IsZonal (σ

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.GaussianBinomialFormula

@[simp] theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GaussianBinomialFormula.gaussBinom_zero_right** (q m : ℕ) : _root_.GlobalTheory.IndefiniteRowSideFullHecke.GaussianBinomialFormula.gaussBinom q m 0 = 1

@[simp] theorem **gaussBinom_zero_succ** (q j : ℕ) : gaussBinom q 0 (j + 1) = 0

theorem **gaussBinom_succ_succ** (q m j : ℕ) : gaussBinom q (m + 1) (j + 1) = gaussBinom q m (j + 1) + q ^ (m - j) * gaussBinom q m j

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.GaussianBinomialFormula.gaussBinom_eq_zero** (q : ℕ) : ∀ m j : ℕ, m < j → _root_.GlobalTheory.IndefiniteRowSideFullHecke.GaussianBinomialFormula.gaussBinom q m j = 0

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.GeneralDiagonalMatrix

theorem **stabLat_one** (V : HSp D) : StabLat V 1

theorem **dict_mem_dom** (h : GA D (T.V j σ)) : (T.dict j σ h).2 ∈ ((T.U j σ).X (T.dict j σ h).1).dom

theorem **rank_eq** : (T.V j σ).rank = D.R.r

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.GeometricConstantEquiv

theorem **IsGeo.exists_lift** {ρ : G →* Module.End T.Q.c.F (Cusp T i w)} (h : IsGeo T R i w ρ) (g : G) : ∃ κ, IsLift T R i g κ

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.GodementJacquetZeta

theorem **sum_blockSizes** (r t j : ℕ) : ∑ a, blockSizes r t j a = r + j * t

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.GoodLiftPhase

theorem **liftWeight_complexPoint_eq_twist** (R : RealDat D A Ks) (F : Face D A) (b : Bool) (j : Fin D.R.m) (y : Fin (Module.finrank D.what₂.LL (R.WL F b j)) → D.what₂.LL) : Tw.out (s

theorem **twMap_smul** {E E' : Type} [NormedAddCommGroup E] [InnerProductSpace ℂ E] [NormedAddCommGroup E'] [InnerProductSpace ℂ E'] (s : Bool) (c : ℂ) (f : E →ₗ[ℂ] E') : Tw.map s (c • f) = cj s c • Tw.map s f

theorem **cxPhase_eq** (R : RealDat D A Ks) (F : Face D A) (ph : Fin D.R.m → ℂ) : cxPhase R F ph = withLift (R.toCx F) fun j t => cj (twAt D.what₂.embLL A.v0) (ph j) • (R.toCx F).θ' j t

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.HalfPowerSpherical

theorem **act_mul** {ι : Type} [Fintype ι] [DecidableEq ι] (g h : GL ι k) (Λ : Submodule 𝒪[k] (ι → k)) : act (g * h) Λ = act g (act h Λ)

theorem **act_one** {ι : Type} [Fintype ι] [DecidableEq ι] (Λ : Submodule 𝒪[k] (ι → k)) : act 1 Λ = Λ

theorem **isChain_chainGen** (α α' : kˣ) (m : ℕ) : IsChain α α' m (chainGen α α' m)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.HeckePrimeDetection

theorem **_root_.GlobalTheory.IndefiniteRowSideFullHecke.HeckePrimeDetection.isHeckePrime_iff** {P : Ideal (𝒪 E)} {p : ℕ} : _root_.GlobalTheory.IndefiniteRowSideFullHecke.HeckePrimeDetection.IsHeckePrime l P p ↔ P.IsMaximal ∧ P ⊔ P.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤ ∧ p.Prime ∧ (p : 𝒪 E) ∈ P ∧ ¬ p ∣ l.M'

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.HeckeSupportLattices

theorem **two_mul_half** : 2 * (D.R.nV / 2) = D.R.nV

theorem **constPinned_of** (O : ∀ i, OMDat T.toCore R i Unit (fun _ => A.μ0)) (i : Fin D.R.m) (hp : OMPins T R κ v₀ O i) (hr : OneMapRow0 T R κ v₀ O i) (hc : CTie T R κ v₀ O i) : ConstPinned T R κ v₀ i

theorem **readAt0_of** (hj : JEvExists T R κ v₀) (fw : FWTie T R κ v₀) (hu : R.LUnif) (i : Fin D.R.m) (hc : ConstPinned T R κ v₀ i) : ReadAt0 T R κ v₀ i where

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.HeckeTransferRankOne

theorem **rank_Vc** (i : Fin D.R.m) : (Vc D i).rank = D.R.nV ∧ D.R.nV = D.R.r + D.R.t

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.HolomorphicLowestType

theorem **pW_add_qW** (i : Fin D.R.m) (τ : D.what₂.Tau) : pW i τ + qW i τ = (Wc D i).rank

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.IndicatorFunctionIdentity

private lemma **isOpen_stdLat** : IsOpen (stdLat k n : Set (n → k))

theorem **isOpen_lat** (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] : IsOpen (L : Set (n → k))

theorem **maps_std_iff** (x : Matrix n n' k) : (∀ y ∈ stdLat k n', x *ᵥ y ∈ stdLat k n) ↔ ∀ i j, x i j ∈ 𝒪[k]

theorem **maps_smul_iff** (g : GL n' k) (h : GL n k) (S : Submodule 𝒪[k] (n' → k)) (L : Submodule 𝒪[k] (n → k)) (x : Matrix n n' k) : (∀ y ∈ g • S, x *ᵥ y ∈ h • L) ↔ ∀ y ∈ S, ((h⁻¹ : GL n k).val * x * g.val) *ᵥ y ∈ L

private lemma **isCompact_isOpen_isClosed_integral_matrices** : IsCompact {x : Matrix n n' k | ∀ i j, x i j ∈ 𝒪[k]} ∧ IsOpen {x : Matrix n n' k | ∀ i j, x i j ∈ 𝒪[k]} ∧ IsClosed {x : Matrix n n' k | ∀ i j, x i j ∈ 𝒪[k]}

private lemma **exists_sideMul_preimage_integral** (S : Submodule 𝒪[k] (n' → k)) (L : Submodule 𝒪[k] (n → k)) [S.IsLattice k] [L.IsLattice k] : ∃ (g : GL n' k) (h : GL n k), {x : Matrix n n' k | ∀ y ∈ S, x *ᵥ y ∈ L} = (sideMul g h) ⁻¹' {x : Matrix n n' k | ∀ i j, x i j ∈ 𝒪[k]}

theorem **hom_lat** (S : Submodule 𝒪[k] (n' → k)) (L : Submodule 𝒪[k] (n → k)) [S.IsLattice k] [L.IsLattice k] : IsCompact {x : Matrix n n' k | ∀ y ∈ S, x *ᵥ y ∈ L} ∧ IsOpen {x : Matrix n n' k | ∀ y ∈ S, x *ᵥ y ∈ L}

private lemma **phiOne_congr_of_sub_mulVec_mem_ker** {P : Pair k n'} {Q : Pair k n} {x z : Matrix n n' k} (hz : ∀ y ∈ P.lat, (z − x) *ᵥ y ∈ Q.ker) : phiOne P Q z = phiOne P Q x

private lemma **isClosed_setOf_mulVec_mem** (S : Submodule 𝒪[k] (n' → k)) (L : Submodule 𝒪[k] (n → k)) [S.IsLattice k] [L.IsLattice k] : IsClosed {x : Matrix n n' k | ∀ y ∈ S, x *ᵥ y ∈ L}

theorem **phiOne_mem** (P : Pair k n') (Q : Pair k n) : phiOne P Q ∈ SB (Matrix n n' k)

private lemma **det_mem_of_integral_entries** {u : GL n k} (hu : ∀ i j, u.val i j ∈ 𝒪[k]) : u.val.det ∈ 𝒪[k]

theorem **valuation_det_of_fix** (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] {h : GL n k} (hh : h • L = L) : valuation k ((Matrix.GeneralLinearGroup.det h : kˣ) : k) = 1

theorem **theta_mem** (hμ : IsUnram μ) (hθ : IsEquivTheta μ π' π θ) (P₀ : Pair k n') (Q₀ : Pair k n) (v : V') : θ v (phiOneSB P₀ Q₀) ∈ levelVec π Q₀

theorem **K_apply** (hμ : IsUnram μ) (hθ : IsEquivTheta μ π' π θ) (P₀ : Pair k n') (Q₀ : Pair k n) (φ : EqForm π') (h : GL n k) : (K hμ hθ P₀ Q₀ φ).1 (h • Q₀) = π h (θ (φ.1 P₀) (phiOneSB P₀ Q₀))

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.IntegralMatrixTranslation

theorem **isIntMat_of_mem** {n : ℕ} {k : GL (Fin n) F} (hk : k ∈ intGL O F n) : IsIntMat O F (k : Matrix (Fin n) (Fin n) F)

theorem **IsIntMat.mul** {l m n : ℕ} {u : Matrix (Fin l) (Fin m) F} {v : Matrix (Fin m) (Fin n) F} (hu : IsIntMat O F u) (hv : IsIntMat O F v) : IsIntMat O F (u * v)

theorem **phiZero_left_invariant** {r n : ℕ} {k : GL (Fin r) F} (hk : k ∈ intGL O F r) (u : Matrix (Fin r) (Fin n) F) : phi0 O F R r n ((k : Matrix (Fin r) (Fin r) F) * u) = phi0 O F R r n u

theorem **phiZero_right_invariant** {r n : ℕ} {k : GL (Fin n) F} (hk : k ∈ intGL O F n) (u : Matrix (Fin r) (Fin n) F) : phi0 O F R r n (u * (k : Matrix (Fin n) (Fin n) F)) = phi0 O F R r n u

theorem **leftTr_mk** (r n : ℕ) (u : Matrix (Fin r) (Fin n) F) (g : GL (Fin r) F) : leftTr O F R r n u (QuotientGroup.mk g) = phi0 O F R r n (((g⁻¹ : GL (Fin r) F) : Matrix (Fin r) (Fin r) F) * u)

theorem **rightTr_mk** (r n : ℕ) (u : Matrix (Fin r) (Fin n) F) (h : GL (Fin n) F) : rightTr O F R r n u (QuotientGroup.mk h) = phi0 O F R r n (u * (h : Matrix (Fin n) (Fin n) F))

theorem **redRank_le** (O : Type) [CommRing O] [IsLocalRing O] {r n : ℕ} (u₀ : Matrix (Fin r) (Fin n) O) : redRank O u₀ ≤ r

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.IwahoriDoubleCosets

theorem **IsIntegralOp.unique** {μ : Measure G} {π : Representation ℂ G V} {f : G → ℂ} {A B : V →ₗ[ℂ] V} (hA : IsIntegralOp μ π f A) (hB : IsIntegralOp μ π f B) : A = B

theorem **HasTrace.unique** {μ : Measure G} {π : Representation ℂ G V} {f : G → ℂ} {c d : ℂ} (hc : HasTrace μ π f c) (hd : HasTrace μ π f d) : c = d

theorem **deltaB_pos** (t : n → kˣ) : 0 < deltaB t

theorem **deltaB_eq_modulus** (t : n → kˣ) : deltaB t = modulus k (fun _ : n => Unit) (leviTorus k n t)

theorem **IsExponents.unique** {π : Representation ℂ (GL n k) V} {E E' : Multiset ((n → kˣ) →* ℂˣ)} (h : IsExponents π E) (h' : IsExponents π E') : E = E'

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.KTypeExtension

theorem **maxCpt_subset** (M : DSModel V S) (τ : D.what₂.Tau) : maxCpt M τ ⊆ Formalism.QuantumDeterminants.UH (gramAt D V τ)

theorem **rep_mem_kFin** (M : DSModel V S) (τ : D.what₂.Tau) {k : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hk : k ∈ maxCpt M τ) {v : M.H τ} (hv : v ∈ kFin M τ) : M.rep τ k v ∈ kFin M τ

theorem **kFin_ne_bot** (M : DSModel V S) (τ : D.what₂.Tau) : kFin M τ ≠ ⊥

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.LabelSettingScalars

theorem **reached_zero** (t : T) (c : EigSys D) : reached R t c 0 = (rowPart R (badPl R t) c : Set (MMap (R.MY 0)))

theorem **reached_succ** (t : T) (c : EigSys D) (k : ℕ) : reached R t c (k + 1) = {x | ∃ u ∈ reached R t c k, IsUp R t k u x}

theorem **rowPart_mono** {Sb Sb' : Finset (HeightOneSpectrum (𝒪 D.host.F))} (h : Sb ⊆ Sb') (c : EigSys D) : rowPart R Sb c ≤ rowPart R Sb' c

theorem **GoodTypes.record** {R : RealDat D A Ks} (h : GoodTypes R) : R.grp_record

theorem **IsLValue.unique** {R : RealDat D A Ks} {t : T} {B : GlobalTheory.LFun.Fam D.host.E} {j : ℕ} {l l' : ℝ} (h : IsLValue R t B j l) (h' : IsLValue R t B j l') : l = l'

theorem **IsUpTop.unique** {R : RealDat D A Ks} {t : T} {u : MMap (R.MY (D.R.m - 1))} {x x' : (R.HX t).H} (v : (R.MY (D.R.m - 1)).Vec) (hv : ∏ τ, ‹(v τ : (R.MY (D.R.m - 1)).H τ), (v τ : (R.MY (D.R.m - 1)).H τ)›_ℂ ≠ 0) (h : IsUpTop R t u x) (h' : IsUpTop R t u x') : x = x'

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.LatticeCosetUniformizer

theorem **mem_latticeOf** (g : GL (Fin n) F) (x : Fin n → F) : x ∈ latticeOf 0 g ↔ ∃ v : Fin n → 0, (g : Matrix (Fin n) (Fin n) F) *ᵥ (fun i => algebraMap 0 F (v i)) = x

theorem **latticeOf_mul_le** (g k : GL (Fin n) F) (hk : k ∈ integralGL 0 F n) : latticeOf 0 (g * k) ≤ latticeOf 0 g

theorem **latticeOf_mul** (g k : GL (Fin n) F) (hk : k ∈ integralGL 0 F n) : latticeOf 0 (g * k) = latticeOf 0 g

@[simp] theorem **latticeOfCoset_mk** (g : GL (Fin n) F) : latticeOfCoset 0 (QuotientGroup.mk g : GL (Fin n) F ⧸ integralGL 0 F n) = latticeOf 0 g

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.LoweredSignatureVanishing

theorem **signCard_dualPar** (S : SPar) (b : Bool) : signCard (dualPar S) b = signCard S b

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.MinorOrderedEntries

theorem **span_orbit_le** {G V : Type} [Monoid G] [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (v : V) (g : G) : Submodule.span ℂ (Set.range fun h : G => ρ h v) ≤ (Submodule.span ℂ (Set.range fun h : G => ρ h v)).comap (ρ g)

theorem **weylSub_le** (n : ℕ) (lam : ι → ℕ → ℤ) (g : UG ι n) : weylSub n lam ≤ (weylSub n lam).comap (leftRep n g)

theorem **nC_succ** : S.nC S.m + 1 = S.nR S.m

theorem **length_block** (x : ℚ) (n : ℕ) : (block x n).length = n

theorem **length_psi** (Q : Entries S) : Q.psiPar.length = S.nR 0

theorem **length_big** (Q : Entries S) : Q.bigPar.length = S.nR S.m

theorem **length_base** (Q : Entries S) : Q.basePar.length = S.nC 0

theorem **length_small** (Q : Entries S) : Q.smallPar.length = S.nC S.m

theorem **hw_exact** (p : List ℚ) (h : ∀ x ∈ p, ∃ z : ℤ, x = z + ((p.length : ℚ) - 1) / 2) (i : ℕ) (hi : i < p.length) : (hw p i : ℚ) = p[i] - (((p.length : ℚ) - 1) / 2 - i)

theorem **block_odd** (c : ℤ) {n : ℕ} (hn : Odd n) : ∀ x ∈ block c n, ∃ z : ℤ, x = z

theorem **block_even** (c : ℤ) {n : ℕ} (hn : Even n) : ∀ x ∈ block c n, ∃ z : ℤ, x = z + 1 / 2

theorem **hw_of_half** (p : List ℚ) (hl : Even p.length) (h : ∀ x ∈ p, ∃ z : ℤ, x = z + 1 / 2) (i : ℕ) (hi : i < p.length) : (hw p i : ℚ) = p[i] - (((p.length : ℚ) - 1) / 2 - i)

theorem **hw_of_int** (p : List ℚ) (hl : Odd p.length) (h : ∀ x ∈ p, ∃ z : ℤ, x = z) (i : ℕ) (hi : i < p.length) : (hw p i : ℚ) = p[i] - (((p.length : ℚ) - 1) / 2 - i)

theorem **even_nR** (j : ℕ) : Even (S.nR j)

theorem **odd_nC** {i : ℕ} (hi : Even i) : Odd (S.nC i)

theorem **psi_half** (Q : Entries S) (hQ : Q.Good) : ∀ x ∈ Q.psiPar, ∃ z : ℤ, x = z + 1 / 2

theorem **base_int** (Q : Entries S) : ∀ x ∈ Q.basePar, ∃ z : ℤ, x = z

theorem **hw_psi** (Q : Entries S) (hQ : Q.Good) (i : ℕ) (hi : i < Q.psiPar.length) : (hw Q.psiPar i : ℚ) = Q.psiPar[i] - (((Q.psiPar.length : ℚ) - 1) / 2 - i)

theorem **hw_big** (Q : Entries S) (hQ : Q.Good) (i : ℕ) (hi : i < Q.bigPar.length) : (hw Q.bigPar i : ℚ) = Q.bigPar[i] - (((Q.bigPar.length : ℚ) - 1) / 2 - i)

theorem **hw_base** (Q : Entries S) (i : ℕ) (hi : i < Q.basePar.length) : (hw Q.basePar i : ℚ) = Q.basePar[i] - (((Q.basePar.length : ℚ) - 1) / 2 - i)

theorem **hw_small** (Q : Entries S) (i : ℕ) (hi : i < Q.smallPar.length) : (hw Q.smallPar i : ℚ) = Q.smallPar[i] - (((Q.smallPar.length : ℚ) - 1) / 2 - i)

theorem **exEntries_good** : exEntries.Good where

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ModerateGrowthSchwartz

theorem **half_add_half** : half D + half D = 1

theorem **fstCols_cols** (x y : WO A B) : fstCols A B (cols A B x y) = x

theorem **sndCols_cols** (x y : WO A B) : sndCols A B (cols A B x y) = y

theorem **coords** (x y : WO A B) : xOf A B (onX A B x) = x ∧ yOf A B (onX A B x) = 0 ∧ xOf A B (onY A B y) = 0 ∧ yOf A B (onY A B y) = y

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.NonFieldTypeVanishing

theorem **classKernel_mul** {Γ : Subgroup G} (γ : G) (f : G → ℂ) {g : G} (hg : g ∈ Γ) (x : G) : classKernel Γ γ f (g * x) = classKernel Γ γ f x

theorem **NonFieldTypeVanishing.zero** (h : NonFieldTypeVanishing) (hr : Odd V.rank) {S : D.what₂.Tau → SPar} {M : DSModel V S} {T : TypeDataG V} {U : AuxiliaryPlaceHermitianSpace V} (hH : HypH (k

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.NormalRepresentationLattice

theorem **nfBasis_val** (ϖ : kˣ) (g : GL n k) (e : Fin c → ℤ) (z : n → k) : (nfBasis ϖ g e z).val = fromBlocks g.val (Matrix.of fun r i => if i.val = 0 then z r * (ϖ : k) ^ e i else 0) 0 (diagonal fun i => (ϖ : k) ^ e i)

theorem **nfBasis_one** (ϖ : kˣ) (e : Fin c → ℤ) (z : n → k) : nfBasis ϖ 1 e z = normalBasis ϖ e z

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.OpenClosedSequences

theorem **indSnd_apply** (e : a ⊕ b ≃ n) (τ : Representation ℂ (Γ × (GL a k × GL b k)) W) (γ : Γ) (h x : GL n k) (f) : (indSnd e τ (γ, h) f).1.1 x = τ (γ, 1) (f.1.1 (x * h))

theorem **indSnd_inr** (e : a ⊕ b ≃ n) (τ : Representation ℂ (Γ × (GL a k × GL b k)) W) (h : GL n k) : indSnd e τ (1, h) = ind₂ e (τ.comp (MonoidHom.inr Γ _)) h

theorem **act_one** (x : Matrix b a k) : act x 1 = x

theorem **act_mul** (x : Matrix b a k) (γ δ : GL a k × GL b k) : act x (γ * δ) = act (act x γ) δ

theorem **continuous_act** (γ : GL a k × GL b k) : Continuous fun x : Matrix b a k => act x γ

theorem **rank_act** (x : Matrix b a k) (γ : GL a k × GL b k) : (act x γ).rank = x.rank

theorem **omega_eq_om** (μ : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (Φ : SB (Matrix b a k)) : (omega a b μ (g, h) Φ).1 = om μ⁻¹ g h Φ.1

theorem **omegaOn_apply** (μ : kˣ →* ℂˣ) (s : Set ℕ) (γ : GL a k × GL b k) (Φ : SB (RankIn k a b s)) (x : RankIn k a b s) : (omegaOn a b μ s γ Φ).1 x = (omChar a b μ γ : ℂ) * Φ.1 ⟨act x.1 γ, by rw [rank_act]; exact x.2⟩

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.OppositeColumnLifts

theorem **absF_one** : absF (1 : (AE D)ˣ) = 1

theorem **sum_divisible_eq_sum_mul** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) (e : A) {p q : ℕ} (u v : Fin p → A) (u' v' : Fin q → A) (h : ∑ i, u i * v i + ∑ i, u' i * v' i = 0) : ∑ i, φ (e * u i) * φ (e * v i) + ∑ i, φ (e * u' i) * φ (e * v' i) = 0

theorem **rowBlocks_mul_colBlocks** (e : AE D) {a a' d₁ d₂ : ℕ} (hd : a + a' = d₁ + d₂) (h : GL (Fin (d₁ + d₂)) (AE D)) : yLeft e hd h * xTop e hd h + yRight e hd h * xBot e hd h = 0

theorem **cTot_ne_zero** {c : T → ℕ → ℕ → ℂ} (h : IsSqConst G c) (t : T) : cTot D c t ≠ 0

theorem **rowVec_tuple** (t t' : T) (i : ℕ) (P : ∀ j, Pol (G.V i j) (G.V i (j + 1))) (n : ℕ) : rowVec G t i P n = rowVec G t' i P n

theorem **colLift_succ_int** (h : G.IsLifts) (t : T) {j n : ℕ} (hj : j ≤ D.R.m) (hn : n < D.R.m) (f : ∀ i, DX G i j) (hf : IsFinVec R.ψ R.θ true (G.V (n + 1) j) (G.V n j) (f n)) (P : ∀ i, Pol (G.V (i + 1) j) (G.V i j)) (F : G.CV 0 j) (v : SmT (G.V 0 j).M) : (G.colLift t j f (n + 1) F).1 (colVec G j P (n + 1) v) = (G.X n j).int (G.meas t n j) (fock R.ψ R.θ true (G.V (n + 1) j) (G.V n j) (f n) (P n)) ((G.colLift t j f n F).1 (colVec G j P n v))

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ParabolicLeviDecomposition

theorem **coord_apply** (x : GL (Fin r) k) : P.coord x = P.g⁻¹ * x * P.g

theorem **M_le_Q** : P.M ≤ P.Q

theorem **exists_mul** {q : GL (Fin r) k} (hq : q ∈ P.Q) : ∃ m ∈ P.M, ∃ n ∈ P.N, q = m * n

theorem **sub_one_mem_lieN** {u : GL (Fin r) k} (hu : u.1 - 1 ∈ P.lieN) : u.1 - 1 ∈ P.lieN

theorem **conj_mem_lieN** {q : GL (Fin r) k} (hq : q ∈ P.Q) {X : Matrix (Fin r) (Fin r) k} (hX : X ∈ P.lieN) : q.1 * X * (q⁻¹ : GL (Fin r) k).1 ∈ P.lieN

theorem **coe_adN** (q : P.M) (X : P.lieN) : (P.adN q X : Matrix (Fin r) (Fin r) k) = q.1.1 * X * (q.1⁻¹ : GL (Fin r) k).1

theorem **exists_isLeviPart** {K : Subgroup (GS F r S)} (h : ∀ x, ∃ q ∈ QS P, ∃ k ∈ K, x = q * k) : ∃ μ, IsLeviPart P K μ

theorem **measurable_subgroupOfEquivOfLe_symm** (h : T ≤ M) : Measurable (Subgroup.subgroupOfEquivOfLe h).symm

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.PartialLFunctionCharacter

theorem **IsPartialL.eq_lPartial** {S : Finset (HeightOneSpectrum (𝒪 D.host.F))} {χ : (AE D)ˣ →* Circle} {Λ : ℂ → ℂ} (h : IsPartialL S χ Λ) {s : ℂ} (hs : 1 < s.re) : Λ s = lPartial S χ s

theorem **IsFiniteL.eq_lEuler** {χ : (AE D)ˣ →* Circle} {Λ : ℂ → ℂ} (h : IsFiniteL χ Λ) {s : ℂ} (hs : 1 < s.re) : Λ s = Mi.lEuler χ s

theorem **eq_of_eqOn_re** {f g : ℂ → ℂ} (hf : Differentiable ℂ f) (hg : Differentiable ℂ g) (h : ∀ s : ℂ, 1 < s.re → f s = g s) : f = g

theorem **IsPartialL.unique** {S : Finset (HeightOneSpectrum (𝒪 D.host.F))} {χ : (AE D)ˣ →* Circle} {Λ Λ' : ℂ → ℂ} (h : IsPartialL S χ Λ) (h' : IsPartialL S χ Λ') : Λ = Λ'

theorem **IsFiniteL.unique** {χ : (AE D)ˣ →* Circle} {Λ Λ' : ℂ → ℂ} (h : IsFiniteL χ Λ) (h' : IsFiniteL χ Λ') : Λ = Λ'

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.PlusPolynomialInclusion

theorem **isUnit_root** (hP : IsPPol c r P) : IsUnit (AdjoinRoot.root P)

theorem **alpha_val** (hP : IsPPol c r P) : (alpha hP : AdjoinRoot P) = AdjoinRoot.root P

theorem **eval₂_conj_inv** (hP : IsPPol c r P) : P.eval₂ ((AdjoinRoot.of P).comp (c : E →+* E)) ((alpha hP)⁻¹ : (AdjoinRoot P)ˣ) = 0

theorem **starL_of** (hP : IsPPol c r P) (e : E) : starL hP (AdjoinRoot.of P e) = AdjoinRoot.of P (c e)

theorem **starL_root** (hP : IsPPol c r P) : starL hP (AdjoinRoot.root P) = ((alpha hP)⁻¹ : (AdjoinRoot P)ˣ)

theorem **starL_starL** (hP : IsPPol c r P) (hc : ∀ e, c (c e) = e) (y : AdjoinRoot P) : starL hP (starL hP y) = y

theorem **classParam_of_CM** (h : ClassParamCM) : ClassParam

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.PositiveRowsTuple

theorem **dimQ_eq** (b : Bool) : dimQ R F b = Module.finrank D.what₂.LL ((A.fam.R A.μ0).inV0 (F.lvl b) ((R.toCx F).Z A.μ0))

theorem **pos_iff** (b : Bool) : (lv (F.fc (R.toCx F).zs) b).Pos ↔ 0 < dimQ R F b

theorem **pos_iff_ne_bot** (b : Bool) : (lv (F.fc (R.toCx F).zs) b).Pos ↔ (A.fam.R A.μ0).inV0 (F.lvl b) ((R.toCx F).Z A.μ0) ≠ ⊥

theorem **AtScale.mono** {eps x0 x1 : ℝ} {D : D7} (h : AtScale eps x1 D) (hx : x0 ≤ x1) : AtScale eps x0 D

theorem **qPos_of** (h1 : RowsPos) (h2 : DimQEq) : QPos

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.PositiveTwistAverage

theorem **omTw_one** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (Φ : Matrix b a k → ℂ) : omTw μ 1 1 g h Φ = om μ g h Φ

theorem **isPosTw_one** (ψ : AddChar k Circle) (p : ℕ) (π : Representation ℂ (GL n k) P) (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (b : P →ₗ[ℂ] A →ₗ[ℂ] ℂ) : IsPosTw ψ 1 p π σ b ↔ IsPos ψ p π σ b

theorem **isUnramUnitary_one** : IsUnramUnitary (1 : kˣ →* ℂˣ)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ProjectiveLiftExistence

theorem **projLift_apply** (R : RealDat D A Ks) (t : T) (x : (R.HX t).H) : projLift R t x = postOn (R.MY 0) R.e0 (PiTensorProduct.lift.symm (R.thDn t x))

theorem **chain_Th_iff** (R : RealDat D A Ks) : R.chain_Th ↔ ∀ t, Represents R t (R.Th t)

theorem **Represents.unique** {R : RealDat D A Ks} (hE : R.models_HE) {t : T} {y y' : (R.HX t).H →ₗ[ℂ] R.HE.H} (h : Represents R t y) (h' : Represents R t y') : y = y'

theorem **exists_lift_iff** {X N : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup N] [Module ℂ N] {M : HMod X} {V : Submodule ℂ X} {ip : X → X → ℂ} (hM : M.Is V ip) (f : N →ₗ[ℂ] X) : (∃ y : N →ₗ[ℂ] M.H, ∀ x, M.e (y x) = f x) ↔ ∀ x, f x ∈ V

theorem **exists_represents_iff** {R : RealDat D A Ks} (hE : R.models_HE) (t : T) : (∃ y : (R.HX t).H →ₗ[ℂ] R.HE.H, Represents R t y) ↔ ∀ x, projLift R t x ∈ R.EEat R.triv

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.PseudoCoefficientSeparation

theorem **hasWideGaps_mul** (r : ℕ) : HasWideGaps fun i : Fin r => -((r : ℤ) * (i : ℕ))

theorem **exists_data_rank_one** (μ : Measure (UJ (allPos 1))) [μ.IsHaarMeasure] : HasWideGaps (fun _ : Fin 1 => (0 : ℤ)) ∧ ∃ hD : IsDiscreteWithInfChar (allPos 1) (fun _ => 0) fun _ => ContinuousLinearMap.id ℂ ℂ, ∃ f, IsPseudoCoeff (ujMat (allPos 1)) μ (ujRep hD.1.2.2.1) f ∧ ∃ hπ : IsIrrAdm (allPos 1) fun _ => ContinuousLinearMap.id ℂ ℂ, HasTrace (repOp (groupRep (allPos 1) hπ.2.1.1) μ f) 1

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.RapidDecayLift

theorem **LiftLaws.mono** {θ : InfFam D} {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} {ν : Measure (GA D W)} {𝓕 : Set (GA D W)} {cls cls' : (GA D W → ℂ) → Prop} (h : LiftLaws θ ψ ω ν 𝓕 cls) (hc : ∀ F, cls' F → cls F) : LiftLaws θ ψ ω ν 𝓕 cls' where

theorem **liftProps_of_rapid** (hP : GlobalTheory.Published.CuspRapidDecay) (h : LiftPropsRapid D) : LiftProps D

theorem **decayInt_le** {W : HSp D} (ν : Measure (GA D W)) (𝓕 : Set (GA D W)) (A : ℕ) (F : GA D W → ℂ) : decayInt ν 𝓕 A F ≤ decaySize W A F * ν 𝓕

theorem **LiftLaws.derT_add** {θ : InfFam D} {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} {ν : Measure (GA D W)} {𝓕 : Set (GA D W)} {cls : (GA D W → ℂ) → Prop} (h : LiftLaws θ ψ ω ν 𝓕 cls) {f : WO V W → ℂ} {F : GA D W → ℂ} (hf : IsThetaFunO ψ V W f) (hF : cls F) (τ : D.what₂.Tau) {Y : Matrix (Fin W.rank) (Fin W.rank) ℂ} (hY : Y ∈ lieAt W τ) (g : GA D V) : lift ω ν 𝓕 f (Rder θ W τ Y F) g + lift ω ν 𝓕 (dOmT θ ω τ Y f) F g = 0

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.RationalHeckeScalars

theorem **readAs_iff** (i : Fin D.R.m) (P : Pl D.host.E) (k : ℕ) (Rc : Finset (GA D (Wc D i))) (hL : (Fin (R.nW i) → D.what₂.LL) →ₗ[D.what₂.LL] (Fin (R.nW i) → D.what₂.LL)) : ReadAs R i P k Rc hL ↔ ∀ x, (R.HW i (sg0 D A)).e ((R.FW i).j (sg0 D A) (hL x)) = heckeScalar R i P k • heckeOn (R.MW i (sg0 D A)) Rc ((R.HW i (sg0 D A)).e ((R.FW i).j (sg0 D A) x))

theorem **RationalHeckeAt.row** {i : Fin D.R.m} {P : Pl D.host.E} {k : ℕ} (h : RationalHeckeAt R i P k) {α : GA D (Wc D i)} (hα : IsHeckeAt (Wc D i) P k α) {Rc : Finset (GA D (Wc D i))} (hR : Bridge.IsDC (levG (Wc D i) (R.βW i) (R.datWp i R.triv)) α Rc) : HeckeJ.HkJRow (sg0 D A) ((R.FW i).j (sg0 D A)) (R.HW i (sg0 D A)).e (heckeOn (R.MW i (sg0 D A)) Rc) (heckeScalar R i P k) (R.FW i).WK

theorem **heckeScalar_zero** (i : Fin D.R.m) (P : Pl D.host.E) : heckeScalar R i P 0 = 1

theorem **readAs_id** (i : Fin D.R.m) (P : Pl D.host.E) : ReadAs R i P 0 {1} LinearMap.id

theorem **RationalHeckeUnpinned.tie** (h : RationalHeckeUnpinned T R F κ v₀) : RationalHeckeTie T R F κ v₀

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.RationalRowTestKernel

theorem **test_tw_iff** {K E : Type} [Field K] [NormedAddCommGroup E] [InnerProductSpace ℂ E] (s : Bool) {σ σ' : K →+* ℂ} (h : ∀ a, σ' a = cj s (σ a)) {ι : Type} {X : ι → Type} (v : ∀ t, X t → E) : (∃ (n : ℕ) (ψ : Fin n → (Tw s E →ₗ[ℂ] ℂ)), (∀ e : Tw s E, (∀ a, ψ a e = 0) → e = 0) ∧ ∀ t x a, ∃ z : K, σ' z = ψ a (Tw.mk s (v t x))) ↔ (∃ (n : ℕ) (ψ : Fin n → (E →ₗ[ℂ] ℂ)), (∀ e : E, (∀ a, ψ a e = 0) → e = 0) ∧ ∀ t x a, ∃ z : K, σ z = ψ a (v t x))

theorem **rowTest_toCx_iff** (R : RealDat D A Ks) (F : Face D A) : RowTest (R.toCx F) ↔ RowTestModel R

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.RayFieldTheta

theorem **proj_clause** (i : Fin D.R.m) (ph : ℂ) (t : T0) (hG : R.thP_thG) (hinj : Function.Injective (R.HW i (sg0 D A)).e) (hp : RhoIs (sg0 D A) ((R.FW i).j (sg0 D A)) (R.HW i _).e (actW R i _) (R.FW i).chi (R.FW i).ρ) (x : (R.HV i t).H) (y : Fin (R.nW i) → D.what₂.LL) (h : (R.HW i (sg0 D A)).e ((R.FW i).j (sg0 D A) y) = ph • R.thF i t x) : ph • R.thG i t x = (R.FW i).j (sg0 D A) ((R.FW i).eOp (fun g => (idemOf A.X).coeff g) y)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.RealDirectionThetaNonzero

theorem **ThetaFacts.std** (h : ThetaFacts D) {ψ : AddChar (AF D.host.F) ℂ} {s : ℤ} (hψ : IsStdChar D ψ s) (S T : HSp D) : ThetaFactsAt ψ S T

theorem **symplOH_zero_left** (S T : HSp D) (y : WO S T) : symplOH S T 0 y = 0

theorem **symplOH_zero_right** (S T : HSp D) (x : WO S T) : symplOH S T x 0 = 0

theorem **heisO_apply_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ξ : WO S T) (f : WO S T → ℂ) : heisO ψ S T ξ f 0 = f ξ

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.RealPoints

theorem **thetaAt_eq** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)) (v : WO S T → ℂ) (g : GA D T) (w : WO S T) : thetaAt ψ om v g w = om (1, g) v w

theorem **perVec_lat** (Wt : UnfAt U k x) (a : (Wt.F.Di (Wt.ik k)).La) : perVec Wt (Wt.φ k a : Fin r → D.host.E) = Wt.Pm a

theorem **exists_witness** (hL : LawsM T.toCore) (hσ : GoodEmb T.Q.c σ) (h : GA D (Wc D i)) : ∃ (hx : (dictW T i σ hV h).2 ∈ ((T.U ((i : ℕ) + 1) σ).X (dictW T i σ hV h).1).dom) (Wt : UnfAt (T.U ((i : ℕ) + 1) σ) (dictW T i σ hV h).1 ((dictW T i σ hV h).2, hx)), Nonempty

(UnitAt Wt)

theorem **ReadsAt.unitary** {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} (H : ReadsAt T R i σ hV κ) {P : Formalism.L2.DIV.Pl D.host.E} {p k : ℕ} {α : GA D (Wc D i)} {Rc : Finset (GA D (Wc D i))} (hs : IsUnitaryHeckeOperatorSplitPrimeData T R i P p k α Rc) {f g : T.R ((i : ℕ) + 1)} (hf : f ∈ sec (T.Q.Xo ((i : ℕ) + 1)) T.Q.c (wtA κ) κ.n) (hg : g ∈ sec (T.Q.Xo ((i : ℕ) + 1)) T.Q.c (wtA κ) κ.n) (hr : GeoHecke T ((i : ℕ) + 1) P.asIdeal p k (hkScal T.Q.c 1 P.asIdeal k D.R.nV) f g) (h : GA D (Wc D i)) : phiW T i σ hV (wtA κ) g h = (muAt T R i P)⁻¹ ^ k * ∑ a ∈ Rc, phiW T i σ hV (wtA κ) f (h * a)

theorem **ReadsAt.modulus** {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} (H : ReadsAt T R i σ hV κ) {P : Formalism.L2.DIV.Pl D.host.E} {p k : ℕ} {α : GA D (Wc D i)} {Rc : Finset (GA D (Wc D i))} (hs : IsUnitaryHeckeOperatorSplitPrimeData T R i P p k α Rc) {f g : T.R ((i : ℕ) + 1)} (hf : f ∈ sec (T.Q.Xo ((i : ℕ) + 1)) T.Q.c (wtA κ) κ.n) (hg : g ∈ sec (T.Q.Xo ((i : ℕ) + 1)) T.Q.c (wtA κ) κ.n) (hr : GeoHecke T ((i : ℕ) + 1) P.asIdeal p k (hkScal T.Q.c ((Nat.card (kP P) : T.Q.c.F)⁻¹ ^ (k * D.R.nV / 2)) P.asIdeal k D.R.nV) f g) (h : GA D (Wc D i)) : phiW T i σ hV (wtA κ) g h = (muAt T R i P)⁻¹ ^ k * (Nat.card (kP P) : ℂ)⁻¹ ^ (k * D.R.nV / 2) * ∑ a ∈ Rc, phiW T i σ hV (wtA κ) f (h * a)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SatakeTrivialRangeBound

theorem **one_lt_resCard** : 1 < resCard k

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SchwartzBruhatMixedFamilies

theorem **MixExpansion.std** (h : MixExpansion D) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) {μ : Measure (Mx D T.rank ρ)} {ℱ : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ ℱ) : ExpansionS ψ d T μ ℱ

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SelfReciprocalLocalField

theorem **isSelfRecip_X_sub_C** {a : E} (ha : a * IsCMField.complexConj E a = 1) : IsSelfRecip E 1 (X - C a)

theorem **finrank_fld** : Module.finrank E (Fld P) = r

theorem **root_ne_zero** : AdjoinRoot.root P.1 ≠ 0

theorem **eval₂_conj_inv_root** : P.1.eval₂ ((AdjoinRoot.of P.1).comp (IsCMField.complexConj E : E →+* E)) (AdjoinRoot.root P.1)⁻¹ = 0

theorem **invol_algebraMap** (e : E) : invol P (algebraMap E (Fld P) e) = algebraMap E (Fld P) (IsCMField.complexConj E e)

theorem **invol_alpha** : invol P (alpha P) = (alpha P)⁻¹

theorem **invol_invol** (y : Fld P) : invol P (invol P y) = y

theorem **Isometric.of_norm** {G : Matrix (Fin r) (Fin r) E} {x n : (k ⊗[maximalRealSubfield E] plus P)ˣ} (hn : n ∈ norms P k) (h : Isometric P k G x) : Isometric P k G (x * n : (k ⊗[maximalRealSubfield E] plus P)ˣ)

theorem **norms_le** : norms P (maximalRealSubfield E) ≤ (norms P k).comap (Units.map (toLocalPlus P k).toMonoidHom)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SemilinearExtensionBases

theorem **IsExt.apply_coe** {Ks LL WL : Type} [Field Ks] [Field LL] [AddCommGroup WL] [Module LL WL] [Module Ks WL] {WK : Submodule Ks WL} {O K K'' : Type} [CommRing O] [Field K] [Algebra Ks K] [Field K''] [Algebra O K''] {Fm : Type} [AddCommGroup Fm] [Module K Fm] [Module O Fm] {emb : LL →+* K''} {eF : Fm ≃ₗ[K] K ⊗[Ks] WK} {e : WL →ₛₗ[emb] K'' ⊗[O] Fm} (h : IsExt O emb eF e) (f : WK) : e f = (1 : K'') ⊗ₜ[O] eF.symm ((1 : K) ⊗ₜ[Ks] f)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SimplePoleTwistedLift

theorem **SimplePole.rep** (h : SimplePole) (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (L : Submodule (𝒪 E) (Fin n → E)) {Vl : HeightOneSpectrum (𝒪 F) → Type} [∀ v, AddCommGroup (Vl v)] [∀ v, Module ℂ (Vl v)] (π : ∀ v, Representation ℂ (GLoc σ J v) (Vl v)) {S : Set (HeightOneSpectrum (𝒪 F))} (hS : S.Finite) {r t : ℕ} (ht : r + 5 ≤ t) {c : Fam E} (μ χ : (AF F ⊗[F] E)ˣ →* Circle) (hc : ∀ w ∉ above E S, (c w).card = r ∧ ∀ b ∈ c w, ‖(b : ℂ)‖ ≤ ((Ideal.absNorm w.asIdeal : ℕ) : ℝ) ^ (((r : ℝ) - 1) / 2)) (hπ : ∀ w ∉ above E S, paramFam σ J L π w = liftFam c (val μ) (val χ) t w) : ∃ Λ : ℂ → ℂ, MeromorphicOn Λ {s | ((r : ℝ) + 1) / 2 < s.re} ∧ (∀ s : ℂ, ((t : ℝ) + 1) / 2 < s.re → Λ s = LSRep σ J L S π (val χ) s) ∧ HasSimplePole Λ ((1 + (t : ℂ)) / 2)

theorem **twist_liftFam** (c : Fam E) (m x : HeightOneSpectrum (𝒪 E) → ℂˣ) (t : ℕ) : twist (liftFam c m x t) x⁻¹ = twist c m⁻¹ + segFam 1 t

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SkewGramHalfModule

theorem **starAdj_ofAdj** (S T : HSp D) (z : Mx D S.rank T.rank) : starAdj S T (ofAdj S T z) = z

theorem **halfMod_pos** {ρ : ℕ} (n : ℕ) (a : GL (Fin ρ) (AE D)) : 0 < halfMod n a

theorem **wittSmallProjection_wittNB_inv_mul_adjointInclusion** {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (b : Mx D ρ T₀.rank) (y : Mx D S.rank ρ) : d.p0 · (mt T (d.nB b)⁻¹ · d.incAdj S y) = ofAdj S T₀ (y · b)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SphericalZero

@[simp] theorem **resL_lagr** (w : WO A B) : resL A B (lagr A B w) = w

@[simp] theorem **resR_lagr** (w : WO A B) : resR A B (lagr A B w) = -w

theorem **delta_lagr** (f₁ f₂ : WO A B → ℂ) (w : WO A B) : delta A B f₁ f₂ (lagr A B w) = f₁ w * conj (f₂ w)

theorem **evZero_delta** (ν : Measure (WO A B)) (ℱ : Set (WO A B)) (f₁ f₂ : WO A B → ℂ) : evZero A B ν ℱ (delta A B f₁ f₂) = ip A B ν ℱ f₁ f₂

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SplitLinearModelNorm

theorem **absV_pos** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (y : (Fv F v)ˣ) : 0 < absV F v y

theorem **absFin_finite** (D : D7) (x : (AE D)ˣ) : (Function.mulSupport fun v => absV D.host.F v (normV D v x)).Finite

theorem **absFin_pos** (D : D7) (x : (AE D)ˣ) : 0 < absFin D x

theorem **modFac_pos** (S T : HSp D) (x y : (AE D)ˣ) : 0 < modFac S T x y

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SplitOffDetection

theorem **keepsCusp_iff_exists** (hE : HasTransform Xo c P p i cq a n) (hD : Detects Xo c a n) (hF : FinKer Xo c P p i) : KeepsCusp Xo c U P p i cq a n ↔ ∀ f ∈ cusp Xo c U a n, ∃ g ∈ cusp Xo c U a n, IsTransform Xo c P p i cq n f g

theorem **IsSplitOff.absNorm_ne_zero** (h : IsSplitOff l P p) : (Ideal.absNorm P : c.F) ≠ 0

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SplitSumStabLattice

theorem **pullPt_eq_actO** (S T : HSp D) (g : GA D S) (h : GA D T) (t : WO S T) : pullPt S T g h t = actO S T g⁻¹ h⁻¹ t

theorem **factor_eq** (χS χT : (AE D)ˣ →* Circle) (x y : (AE D)ˣ) : ((χT x : ℂ))⁻¹ * (χS y : ℂ) = (splitFacO χS χT x y : ℂ)

theorem **mulVec_mem_hatL** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} {β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} {Λ : Submodule (𝒪 E) (Fin n → E)} (hβ : Formalism.L2.DIV.IsFrame β Λ) {k : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} (hk : k ∈ Formalism.L2.DIV.St β) {y : Fin n → E ⊗[F] Formalism.L2.Herm.Af F} (hy : y ∈ Formalism.L2.Herm.hatL (F

theorem **mulVec_mem_hatL_iff** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} {β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} {Λ : Submodule (𝒪 E) (Fin n → E)} (hβ : Formalism.L2.DIV.IsFrame β Λ) {k : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} (hk : k ∈ Formalism.L2.DIV.St β) (y : Fin n → E ⊗[F] Formalism.L2.Herm.Af F) : y ∈ Formalism.L2.Herm.hatL (F

theorem **stabLat_of_mem_St** (V : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)) (hβ : Formalism.L2.DIV.IsFrame β Λ) (g : GA D V) (hg : Formalism.IndefR.Lev.finE V g ∈ Formalism.L2.DIV.St β) : StabLat V Λ g

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.StandardMaximalParabolic

theorem **mem_conjBy** {G : Type} [Group G] {g x : G} {H : Subgroup G} : x ∈ conjBy g H ↔ g⁻¹ * x * g ∈ H

theorem **mem_stdMaxParab** {R : Type} [CommRing R] {a b r : ℕ} (h : a + b = r) {g : GL (Fin r) R} : g ∈ stdMaxParab R h ↔ ∀ i j : Fin r, (j : ℕ) < a → a ≤ (i : ℕ) → (g : Matrix (Fin r) (Fin r) R) i j = 0

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.SupremumKeepingLattice

theorem **baseChange_mem** {k K' M : Type} [Field k] [Field K'] [Algebra k K'] [AddCommGroup M] [Module k M] (h : M →ₗ[k] M) (U : Submodule k M) (hU : ∀ x ∈ U, h x ∈ U) : ∀ x ∈ U.baseChange K', h.baseChange K' x ∈ U.baseChange K'

theorem **span_mem** {K' N : Type} [Field K'] [AddCommGroup N] [Module K' N] (O : ValuationSubring K') (h : N →ₗ[K'] N) (S : Set N) (hS : ∀ x ∈ S, h x ∈ S) : ∀ x ∈ Submodule.span O S, h x ∈ Submodule.span O S

theorem **partK_mem** (h : WL →ₗ[LL] WL) (hK : ∀ x ∈ X.WK, h x ∈ X.WK) (U : Submodule LL WL) (hU : ∀ x ∈ U, h x ∈ U) (K' : Type) [Field K'] [Algebra Ks K'] : ∀ x ∈ partK X U K', (onK X h hK).baseChange K' x ∈ partK X U K'

theorem **keepsSup_iff** {h : WL →ₗ[LL] WL} {hK : ∀ x ∈ X.WK, h x ∈ X.WK} {la : HeightOneSpectrum (𝒪 Ks)} : KeepsSup X h hK la ↔ (∀ f ∈ ball X la 1, onK X h hK f ∈ ball X la 1) ∧ ∀ g, h ∘ₗ X.ρ g = X.ρ g ∘ₗ h

theorem **keepsSup_id** (la : HeightOneSpectrum (𝒪 Ks)) : KeepsSup X LinearMap.id (fun _ hx => hx) la

theorem **liesOver_self** (la : HeightOneSpectrum (𝒪 Ks)) : LiesOver la (la.valuation Ks).valuationSubring

theorem **base_of_tie** (fw : FWTie T R κ v₀) : res T (sg0 D A) = T.σ₀ ∧ GoodEmb T.Q.c (res T (sg0 D A)) ∧ T.V ((i : ℕ) + 1) (res T (sg0 D A)) = Wc D i

theorem **ReadPinned.eval** (fw : FWTie T R κ v₀) (H : ReadPinned T R κ v₀ i) (hV : T.V ((i : ℕ) + 1) (res T (sg0 D A)) = Wc D i) (f : Cu T κ i) : evAt R i (sg0 D A) (v₀ i (sg0 D A)) (fw.jK i (sg0 D A) f) = phiC T hV (wtA (κ i)) (f : T.R ((i : ℕ) + 1))

theorem **readPinned_of_unif** (hu : R.LUnif) (hV : T.V ((i : ℕ) + 1) (res T (sg0 D A)) = Wc D i) (hf : FramePin T R i (res T (sg0 D A)) hV) (hj : JEvAt T R κ v₀ i (sg0 D A) hV) : ReadPinned T R κ v₀ i

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ThetaAbsoluteSum

theorem **rapidMeas_of_rapidDec** {G : GA D T → ℂ} (h : RapidDec T G) : RapidMeas T G

theorem **ThetaGrowth.at** (h : ThetaGrowth) (D : D7) : ThetaGrowthAt D

theorem **ThetaGrowthFull.thetaGrowth** (h : ThetaGrowthFull) : ThetaGrowth

theorem **PairBounds.rder_lift** (H : PairBounds A d ψ om θ μ ℱ) {τ : D.what₂.Tau} {X : Matrix (Fin S.rank) (Fin S.rank) ℂ} (hX : X ∈ lieAt S τ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {ν : Measure (GA D T)} {𝒢 : Set (GA D T)} (hν : ν.IsHaarMeasure) (h𝒢 : IsFundamentalDomain (GQ T) 𝒢 ν) {G : GA D T → ℂ} (hG : RapidMeas T G) : Rder θ S τ X (lift om ν 𝒢 f G) = lift om ν 𝒢 (dOm om θ τ X f) G

theorem **PairBounds.modGrowthExp_lift** (H : PairBounds A d ψ om θ μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {ν : Measure (GA D T)} {𝒢 : Set (GA D T)} (hν : ν.IsHaarMeasure) (h𝒢 : IsFundamentalDomain (GQ T) 𝒢 ν) {G : GA D T → ℂ} (hG : RapidMeas T G) : ModGrowthExp S A (lift om ν 𝒢 f G)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ThetaAdjointScalarParts

theorem **gap** {Rk : Ranks} (h : HCParam Rk) (j : Fin Rk.m) {n : ℕ} (hn : n < Rk.r) : ((Rk.t : ℚ) + 3) / 2 ≤ |Formalism.L2.NTheta.lamAt h n + h.e j|

theorem **IsScalarOnParts.mem** {R : RealDat D A Ks} {lam : Fin D.R.m → T → EigSys D → ℝ} (h : IsScalarOnParts R lam) (i : Fin D.R.m) (t : T) (c : EigSys D) {x} (hx : x ∈ goodPartOff R i (badPlaces R t) c) : R.thG i t (LinearMap.adjoint (R.thG i t) x) ∈ goodPartOff R i (badPlaces R t) c

theorem **offS_iff** (S : Finset (HeightOneSpectrum (𝒪 D.host.F))) (P : HeightOneSpectrum (𝒪 D.host.E)) : OffS D S P ↔ P.under (𝒪 D.host.F) ∉ S

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ThetaKernelGrowth

theorem **ThetaGrowthUnif.thetaGrowth** (H : ThetaGrowthUnif) (D : D7) : ThetaGrowth D

theorem **KernelBound.real** {D : D7} {V W : HSp D} {A d : ℕ} (H : KernelBound V W A d) (ψ : AddChar (AF D.host.F) ℂ) {θ : InfFam D} (hθ : IsInfFam θ) {U : AddSubgroup (WO V W)} (hU : IsOpen (U : Set (WO V W))) : ∃ C : ℝ, 0 ≤ C ∧ ∀ (χS χT : (AE D)ˣ →* Circle) (ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)), IsWeilDatumO ψ V W χS χT ω → ∀ (μ : Measure (WO V W)) (ℱ : Set (WO V W)), IsCovolOne (ratL W V) μ ℱ → ∀ f ∈ thetaLevel ψ V W U, sobO ψ θ μ ℱ d f ≠ ∞ ∧ ∀ (g : GA D V) (h : GA D W), ‖ω (g, h) f 0‖ ≤ C * (sobO ψ θ μ ℱ d f).toReal * (height V g * height W h) ^ A

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ThetaPrimeMultiplicity

theorem **IsThetaPrimeMult.integral** {R : RealDat D A Ks} {i : Fin D.R.m} {t : T} {y : (R.HV i t).H →ₗ[ℂ] MMap (R.MW i (sg0 D A))} (h : IsThetaPrimeMult R i t y) (x : (R.HV i t).H) (v : ∀ ρ, lowT (R.MW i (sg0 D A)) ρ) (g : GA D (Wc D i)) : IntegrableOn (fun u : GA D (Vc

D i) => R.ωP i t (u, g) (R.ptP i t (R.ioP i (PiTensorProduct.tprod ℂ v) (R.extV i t x u))) 0) (R.fdV i t) (R.νV i t) ∧ y x (fun ρ => (v ρ : (R.MW i (sg0 D A)).sm ρ)) g = ∫ u in R.fdV i t, R.ωP i t (u, g) (R.ptP i t (R.ioP i (PiTensorProduct.tprod ℂ v) (R.extV i t x u))) 0 ∂(R.νV i t)

theorem **isThetaPrimeMult_thF_iff** (R : RealDat D A Ks) : (∀ i t, IsThetaPrimeMult R i t (R.thF i t)) ↔ R.thP_thF_mem ∧ R.thP_thF

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ThetaSourceAction

theorem **isAdapted_model** {V : HSp D} {P : D.what₂.Tau → SPar} (M : DSModel V P) (τ : D.what₂.Tau) : IsAdapted (gramAt D V τ) (cart M τ) (onb M τ)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.ThetaTranslateRangeGap

theorem **range_eq_comap_iff** {X H M : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup H] [Module ℂ H] [AddCommGroup M] [Module ℂ M] (e : H →ₗ[ℂ] M) (f : X →ₗ[ℂ] H) (p : X →ₗ[ℂ] M) (U : Submodule ℂ M) (hinj : Function.Injective e) (hle : U ≤ LinearMap.range e) (hc : ∀ x, e (f x) = p x) : LinearMap.range f = U.comap e ↔ LinearMap.range p = U

theorem **goodRangeAt_iff** {R : RealDat D A Ks} (hW : R.models_HW) (hG : R.thP_thG) (i : Fin D.R.m) (t : T) : GoodRangeAt R i t ↔ LinearMap.range (postOn (R.MW i (sg0 D A)) (R.eWf i (sg0 D A)) ∘ₗ R.thF i t) = R.Wat i (sg0 D A) R.triv

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.TraceKernelDecomposition

theorem **TraceKernel.uncond** (h : TraceKernel) (hc : ∀ (D : D7) (θ : InfF D), IsInfFam θ → ∀ (V : HSp D) (μ : Measure (GA D V)) [μ.IsHaarMeasure] [μ.IsMulRightInvariant], CuspTraceClass θ V μ) : TraceKernelUncond

theorem **oneP_mem_GinfC** (θ : InfF D) (V : HSp D) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin V.rank) ℂ) (t : ℝ) : oneP θ V τ X t ∈ GinfC V

theorem **isCcInf_indicator** (θ : InfF D) (V : HSp D) {K : Subgroup (GA D V)} (ho : IsOpen (K : Set (GA D V))) (hc : IsCompact (K : Set (GA D V))) (hK : GinfC V ≤ K) : IsCcInf θ V ((K : Set (GA D V)).indicator fun _ => 1)

theorem **exists_isCcInf** (θ : InfF D) (V : HSp D) (hV : V.gram = 1) : ∃ f, IsCcInf θ V f ∧ f 1 = 1

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.TranslateRepresentations

theorem **IsTranslate.unique** {W : Type} [NormedAddCommGroup W] [InnerProductSpace ℂ W] {g₀ : G0 L.Pg L.r} {F F₁ F₂ : Bridge.Gf V → W} (h₁ : IsTranslate V β L g₀ F F₁) (h₂ : IsTranslate V β L g₀ F F₂) : F₁ = F₂

theorem **IsTransRep.unique** {W : Type} [NormedAddCommGroup W] [InnerProductSpace ℂ W] {G : Type} [Group G] {eG : G ≃* G0 L.Pg L.r} {M : HMod (Bridge.Gf V → W)} (hM : Function.Injective M.e) {ρ₁ ρ₂ : Representation ℂ G M.H} (h₁ : IsTransRep V β L eG M ρ₁) (h₂ : IsTransRep V β L eG M ρ₂) : ρ₁ = ρ₂

theorem **IsTransRep.eq_of_translate** {W : Type} [NormedAddCommGroup W] [InnerProductSpace ℂ W] {G : Type} [Group G] {eG : G ≃* G0 L.Pg L.r} {M : HMod (Bridge.Gf V → W)} (hM : Function.Injective M.e) {ρ : Representation ℂ G M.H} (h : IsTransRep V β L eG M ρ) {g : G} {x x' : M.H} (hx : IsTranslate V β L (eG g) (M.e x) (M.e x')) : x' = ρ g x

theorem **IsTranslate.eq_g0Act** {W : Type} [NormedAddCommGroup W] [InnerProductSpace ℂ W] {g₀ : G0 L.Pg L.r} {F F' : Bridge.Gf V → W} (h : IsTranslate V β L g₀ F F') (κ : (levEnv V β L).Q) (hκ : LiftsTo V β L κ g₀) : F' = Bridge.g0Act (levEnv V β L).ch κ F

theorem **GEquiv.th_of_translate** (h : GEquiv R) (t : T) (hinj : Function.Injective (R.HX t).e) (g : G) (x x' : (R.HX t).H) (hx : IsTranslate D.sp.X' R.βX (R.datXt t) (R.eG g) ((R.HX t).e x) ((R.HX t).e x')) : R.Th t x' = R.gE g (R.Th t x)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.TransvectionAction

theorem **IsTransv.unique** {S : HSp D} {w v : VA S} {g h : GA D S} (hg : IsTransv S w v g) (hh : IsTransv S w v h) : g = h

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.TrivialParameterBound

theorem **bound_eq** (q n : ℕ) : ((q : ℝ) ^ 2) ^ (((n : ℝ) - 1) / 2) = (q : ℝ) ^ ((n : ℝ) - 1)

theorem **trivial_hyp** {L : Type} [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] {ϖ : L} {σ : L →+* L} {q : ℕ} (hU : IsUnramInv ϖ σ q) (hq : q ≠ 0) (n : ℕ) : IsIrrSmooth (charRep (1 : GU σ n →* ℂˣ)) ∧ IsUnitarisable (charRep (1 : GU σ n →* ℂˣ)) ∧ HasSatU ϖ q (charRep (1 : GU σ n →* ℂˣ)) (trivParam q n hq)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.UniformModerateGrowthLevel

theorem **mem_mult** {M : DSModel W S} {J : Set (GA D W)} {χ : GA D W → ℂ} {T : MMap M} (h : IsMultMap θ M J χ T) : T ∈ Mult θ M J χ

theorem **isHolPar_parW** (κ : ∀ τ : D.what₂.Tau, Fin (W.sig (tauE D τ)).1 ⊕ Fin (W.sig (tauE D τ)).2 → ℤ) (h : ∀ τ, Formalism.L2.NTheta.HolOK _ _ (κ τ)) : IsHolPar (parW W κ)

theorem **single_mem_pMinus** (M : DSModel W S) (τ : D.what₂.Tau) {a b : Fin W.rank} (ha : M.ε τ a = false) (hb : M.ε τ b = true) : (M.fr τ)⁻¹ * Matrix.single a b (1 : ℂ) * M.fr τ ∈ pMinus M τ

theorem **isLevel_top** : IsLevel W (GfC W) (fun _ => 1)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.UnitaryAtAllRows

theorem **pairAt_smul** (a b : K) (x y : S.V) : pairAt S σ (a • x) (b • y) = starRingEnd ℂ (σ a) * σ b * pairAt S σ x y

theorem **csumAt_smul** (a b : K) (f f' : S.Cl → S.V) : csumAt S σ (a • f) (b • f') = starRingEnd ℂ (σ a) * σ b * csumAt S σ f f'

theorem **pairAt_eq** (h : ∀ a, σ (star a) = starRingEnd ℂ (σ a)) (x y : S.V) : pairAt S σ x y = σ (S.pair x y)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.UnramifiedCharacter

theorem **inStabAt_one** (V : HSp D) (L : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (v : HeightOneSpectrum (𝒪 D.host.F)) : InStabAt V L v 1

theorem **isLatVac_at_of_all** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {f : WO S T → ℂ} (hf : IsLatVac (heisO ψ S T) (latO S T LS LT fun _ => True) f) (v : HeightOneSpectrum (𝒪 D.host.F)) : IsLatVac (heisO ψ S T) (latO S T LS LT fun w => w = v) f

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.VacuumFockVectors

theorem **ptOf_one** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (M : DSModel V S) (M' : DSModel W S') (f : WO V W → ℂ) : ptOf ψ θ M M' f 1 = f

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.VectorLevelCharacters

theorem **principalLevelCharacter_eq_one** (R : RealDat D A Ks) (i : Fin D.R.m) (k : GA D (Wc D i)) : levChi (Wc D i) (R.βW i) (R.toLevCh.datWp i R.toLevCh.triv) k = 1

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.VolumeFlatSection

theorem **mem_parahoric** {g : GL (A ⊕ D) k} : g ∈ parahoric k A D ↔ g ∈ KO k (A ⊕ D) ∧ ∀ i j, valuation k (g.1 (.inr i) (.inl j)) < 1

theorem **InSupp.det_ne** {x : Matrix (A ⊕ D) (A ⊕ D') k} (h : InSupp x) : x.toBlocks₁₁.det ≠ 0

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.WeilDatumExistence

theorem **weilDatumExists_iff** (D : D7) : WeilDatumExists D ↔ ∀ (S T : HSp D) (ψ : AddChar (AF D.host.F) ℂ) (χS χT : (AE D)ˣ →* Circle), ((∃ om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ), IsWeilDatumO ψ S T χS χT om) ↔ IsStdChar D ψ (-1) ∧ IsHeckeChar D χS ∧ IsHeckeChar D χT ∧ ResEta D χS S.rank ∧ ResEta D χT T.rank)

Development.AbelianVarietyTheory.BlockTheory.TwistedDiscreteSeries.WeilDatumUniqueness

theorem **IsTildeDet.unique** {V : HSp D} {ξ : (AE D)ˣ →* Circle} {g : GA D V} {c c' : Circle} (h : IsTildeDet V ξ g c) (h' : IsTildeDet V ξ g c') : c = c'

theorem **IsTildeDet.at_one** {V : HSp D} {ξ : (AE D)ˣ →* Circle} {c : Circle} (h : IsTildeDet V ξ 1 c) : c = 1

theorem **WeilDatumTwist.unique** (H : WeilDatumTwist D) : WeilDatumUnique D

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.BaseChangeCharts

lemma **specMap_algebraMap_self** (K : Type) [CommRing K] : Spec.map (CommRingCat.ofHom (algebraMap K K)) = 𝟙 _

lemma **bcB_prB** : bcB D F ≫ DegPre.prB D (FractionRing R) = DegPre.prB D F

lemma **bcB_famφ** (α : D.X) (i : Fin 4) : (bcB D F).left ≫ DegPre.famφ D (FractionRing R) α i = DegPre.famφ D F α i

lemma **chart_le_bcB** (α : D.X) (j : Fin 4 → D.P.L.ι) : LB.wchart D.P.L (DegPre.famφ D F α) j ≤ (bcB D F).left ⁻¹ᵁ LB.wchart D.P.L (DegPre.famφ D (FractionRing R) α) j

lemma **toGK_gen** (U : UnifPre I D A G) : toGK F U ≫ U.gen.hom.left = (snd (TFK (R

lemma **cmpW_le_L** (U : UnifPre I D A G) (k : ℕ) (a : U.LG.L.ι) (b : A.L.ι) : cmpW F U a b ≤ (LPow A.A A.L (k : ℤ) (TFK (R

lemma **cmpW_le_LG** (U : UnifPre I D A G) (a : U.LG.L.ι) (b : A.L.ι) : cmpW F U a b ≤ toGK F U ⁻¹ᵁ LB.wchart U.LG.L (fun _ : Fin 1 => pullback.fst G.hom (jK R)) (fun _ => a)

theorem **IsTheImage.torsion** {U : Unif I D A G} {k : ℕ} {p : QkPt D.toPre F k} {x : TFK (R

theorem **isTheImage_of_forall** {U : Unif I D A G} {k : ℕ} (hk : 1 ≤ k) {p : QkPt D.toPre F k} {x : TFK (R

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.ConstantUnitPairing

theorem **ΓSpecIso_hom_constUnit** (R : Type) [CommRing R] [IsDomain R] (F : Type) [Field F] [Algebra (FractionRing R) F] (ζ : Fˣ) : (Scheme.ΓSpecIso (CommRingCat.of F)).hom.hom (constUnit R F ζ : Γ((TFK (R

theorem **constUnit_bijective** (R : Type) [CommRing R] [IsDomain R] (F : Type) [Field F] [Algebra (FractionRing R) F] : Function.Bijective (constUnit R F)

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.PowerGroupDatum

lemma **powPr_isMonHom** : (j : ℕ) → (i : Fin j) → IsMonHom (powPr G j i) | 0 => fun i => i.elim0 | j + 1 => fun i => by refine Fin.cases ?_ (fun k => ?_) i · exact inferInstanceAs (IsMonHom (fst G (powObj G j))) · have

lemma **powPr_bij** (T : Over S) : (j : ℕ) → Function.Bijective (fun (x : T ⟶ powObj G j) (i : Fin j) => x ≫ powPr G j i) | 0 => (fun x y _ => toUnit_unique (X

lemma **eSec_inv** : eSec G ≫ (GrpObj.inv (X

lemma **eSec_pr** (i : Fin j) : eSec G' ≫ (P.pr i).left = eSec G

lemma **inv_pr** (i : Fin j) : (GrpObj.inv (X

lemma **rig_le** (L : LB G.left) (a : PUnit.{1} → Fin j → L.ι) (i : Fin j) : LB.wchart (P.box L) (fun _ : PUnit.{1} => eSec G') a ≤ LB.wchart L (fun _ : PUnit.{1} => eSec G) (fun _ => a () i)

lemma **symm_le** (L : LB G.left) (p : (PUnit.{1} → Fin j → L.ι) × (Fin j → L.ι)) (i : Fin j) : LB.wchart (P.box L) (fun _ : PUnit.{1} => (GrpObj.inv (X

lemma **theta_le** (L : LB G.left) (a : thK.{0} → Fin j → L.ι) (i : Fin j) : LB.wchart (P.box L) (thφ G' (x3 G')) a ≤ (P.pr i ⊗ₘ P.pr i ⊗ₘ P.pr i).left ⁻¹ᵁ LB.wchart L (thφ G (x3 G)) (fun I => a I i)

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.RawDegreeForms

theorem **IsCompleteLocalNormal.pair** {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] (h : IsCompleteLocalNormal R) : IsAdicNormalPair R (IsLocalRing.maximalIdeal R)

theorem **isCompleteLocalNormal_padicInt** (p : ℕ) [Fact p.Prime] : IsCompleteLocalNormal ℤ_[p]

theorem **isAdicNormalPair_powerSeries** (O : Type) [CommRing O] [IsDomain O] [IsPrincipalIdealRing O] : IsAdicNormalPair (PowerSeries O) (Ideal.span {PowerSeries.X})

theorem **cT_add** (T : Over (base R)) (χ χ' : D.X) : D.cT T (χ + χ') = padd D.P.A (D.cT T χ) (D.cT T χ')

theorem **res_res** {Y : Scheme.{0}} {U V W : Y.Opens} (h : V ≤ U) (h' : W ≤ V) (u : Γ(Y, U)ˣ) : res h' (res h u) = res (h'.trans h) u

theorem **res_ψAt** {χ χ' : D.X} {i i' : D.P.L.ι} (e : χ = χ') (e' : i = i') {V V' W} (h h') (k : W ≤ V) (k' : W ≤ V') : res k (D.ψAt χ i V h) = res k' (D.ψAt χ' i' V' h')

theorem **res_rigAt** {i i' : D.P.L.ι} (e' : i = i') {V V' W} (h h') (k : W ≤ V) (k' : W ≤ V') : res k (D.rigAt i V h) = res k' (D.rigAt i' V' h')

theorem **quot_congr** {A : Type} [CommGroup A] {a b c d a' b' c' d' : A} (h0 : a = a') (h1 : b = c') (h2 : c = b') (h3 : d = d') : a * b⁻¹ * c⁻¹ * d⁻¹ = a' * b'⁻¹ * c'⁻¹ * d'⁻¹

theorem **τ_symm** (χ χ' : D.X) : LB.Gen.Corr D.P.L _ _ _ _ (D.τ χ χ') (D.τ χ' χ)

theorem **res_pull_res** {Y Z : Scheme.{0}} (f : Z ⟶ Y) {U V : Y.Opens} (h : V ≤ U) {W : Z.Opens} (k : W ≤ f ⁻¹ᵁ V) (k' : W ≤ f ⁻¹ᵁ U) (u : Γ(Y, U)ˣ) : res k (pullU f V (res h u)) = res k' (pullU f U u)

theorem **pull_pull** {Y Z Z' : Scheme.{0}} (f : Z ⟶ Y) (g : Z' ⟶ Z) (U : Y.Opens) (u : Γ(Y, U)ˣ) : pullU g _ (pullU f U u) = pullU (g ≫ f) U u

theorem **res_pull_res_pull** {Y Z Z' : Scheme.{0}} (f : Z ⟶ Y) (g : Z' ⟶ Z) (U : Y.Opens) {V : Z.Opens} {W : Z'.Opens} (h : V ≤ f ⁻¹ᵁ U) (k : W ≤ g ⁻¹ᵁ V) (k' : W ≤ (g ≫ f) ⁻¹ᵁ U) (u : Γ(Y, U)ˣ) : res k (pullU g V (res h (pullU f U u))) = res k' (pullU (g ≫ f) U u)

theorem **ψAt_trans** (χ : D.X) (i i' : D.P.L.ι) {V V' W} (h h') (k : W ≤ V) (k' : W ≤ V') (hW) : res k (D.ψAt χ i V h) = (D.gAt (D.cK χ).left i i' W hW)⁻¹ ∗ res k' (D.ψAt χ i' V' h')

theorem **rigAt_trans** (i i' : D.P.L.ι) {V V' W} (h h') (k : W ≤ V) (k' : W ≤ V') (hW) : res k (D.rigAt i V h) = D.gAt (pzero D.P.A (genPt R)).left i i' W hW ∗ res k' (D.rigAt i' V' h')

theorem **gAt_congr** {f f' : (genPt R).left ⟶ D.P.A.left} (e : f = f') (i i' W h h') : D.gAt f i i' W h = D.gAt f' i i' W h'

theorem **τ_isGen** (χ χ' : D.X) : (D.τ χ χ').IsGen

theorem **IsDeg.io_isPt** {D : RawDeg R} {I : Ideal R} (h : D.IsDeg I) (y : D.X) : (D.io y).IsPt

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.RigidPullbackIdentification

lemma **chart_le_target** (p' : Y ⟶ X') (p : Y ⟶ X) (hp : p' ≫ F = p) (j : L.ι) (j' : L'.ι) : p' ⁻¹ᵁ (LB.div (L.pull F) L').U (fun _ => j, j') ≤ LB.wchart L (fun _ : PUnit => p) (fun _ => j)

lemma **chart_le_source** (p' : Y ⟶ X') (j : L.ι) (j' : L'.ι) : p' ⁻¹ᵁ (LB.div (L.pull F) L').U (fun _ => j, j') ≤ LB.wchart L' (fun _ : PUnit => p') (fun _ => j')

lemma **ptK_comp** (φ : R →+* R') {B : Over (Spec (CommRingCat.of R))} {B' : Over (Spec (CommRingCat.of R'))} (F : B'.left ⟶ B.left) (x : 𝟙_ (Over (Spec (CommRingCat.of R))) ⟶ B) (x' : 𝟙_ (Over (Spec (CommRingCat.of R'))) ⟶ B') (hx : x'.left ≫ F = specOf φ ≫ x.left) (W : (Spec (CommRingCat.of R)).Opens) (g : (TK R').left ⟶ (TW W).left) (hg : g ≫ W.ι = jK R' ≫ specOf φ) : ptK x' ≫ F = g ≫ (toUnit (TW W) ≫ x).left

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.SeparatedSymmetricModel

@[simp] lemma **dn_left** {G : Over S} {T : Over S'} (y : T ⟶ (Over.pullback p).obj G) : (dn p y).left = y.left ≫ pullback.fst G.hom p

lemma **dn_map** {G G' : Over S} {T : Over S'} (y : T ⟶ (Over.pullback p).obj G) (f : G ⟶ G') : dn p (y ≫ (Over.pullback p).map f) = dn p y ≫ f

lemma **dn_padd** {T : Over S'} (a b : T ⟶ (Over.pullback p).obj G) : dn p (padd _ a b) = padd G (dn p a) (dn p b)

lemma **dn_pzero** (T : Over S') : dn p (pzero ((Over.pullback p).obj G) T) = pzero G _

lemma **dn_foldr** {T : Over S'} {n : ℕ} (x : Fin n → (T ⟶ (Over.pullback p).obj G)) (l : List (Fin n)) : dn p (l.foldr (fun k acc => padd _ (x k) acc) (pzero _ T)) = l.foldr (fun k acc => padd G (dn p (x k)) acc) (pzero G _)

lemma **downThree_comp_proj** (f : H ⟶ (Over.pullback p).obj G) (k : Fin 3) : dn3 p f ≫ x3 G k = dn p (x3 H k ≫ f)

lemma **inv_dn** (f : H ⟶ (Over.pullback p).obj G) [IsMonHom f] : (ι[H]).left ≫ (dn p f).left = (dn p f).left ≫ (ι[G]).left

lemma **symPull_le** (F : H.left ⟶ G.left) (hF : (ι[H]).left ≫ F = F ≫ (ι[G]).left) (L : LB G.left) (j : Fin 2 → PUnit → L.ι) : LB.wchart (L.pull F) ![(ι[H]).left, 𝟙 H.left] j ≤ F ⁻¹ᵁ LB.wchart L ![(ι[G]).left, 𝟙 G.left] (fun k => j k ⟨⟩)

lemma **isoPull_le** (h : Y' ⟶ Y) {a : Y ⟶ X} {b : Y ⟶ X'} {a' : Y' ⟶ X} {b' : Y' ⟶ X'} (ha : h ≫ a = a') (hb : h ≫ b = b') (j : (PUnit → M.ι) × (PUnit → L.ι)) : (LB.div (M.pull b') (L.pull a')).U j ≤ h ⁻¹ᵁ (LB.div (M.pull b) (L.pull a)).U j

lemma **toG_inv** (I : Ideal R) (e : ℕ) : (ι[thickObj R I e G]).left ≫ (toG I e G).F = (toG I e G).F ≫ (ι[G]).left

lemma **toB_inv** {e : ℕ} (j : CanonicalRigidImages.GtIso D I e G) : (ι[thickObj R I e G]).left ≫ (toB j).F = (toB j).F ≫ (ι[D.P.A]).left

lemma **thickMap_fst** (I : Ideal R) (e : ℕ) (Z : Over (Spec (CommRingCat.of R))) : thickMap R I e Z ≫ pullback.fst Z.hom (thick R I (e + 1)) = pullback.fst Z.hom (thick R I e)

lemma **thickMap_toG** (I : Ideal R) (e : ℕ) : thickMap R I e G ≫ (toG I (e + 1) G).F = (toG I e G).F

lemma **thickMap_toB** {e : ℕ} (j : CanonicalRigidImages.GtIso D I e G) (j' : CanonicalRigidImages.GtIso D I (e + 1) G) (h : thickMap R I e G ≫ j'.q.left = j.q.left ≫ thickMap R I e D.P.A) : thickMap R I e G ≫ (toB j').F = (toB j).F

Development.AbelianVarietyTheory.BlockTheory.UniformisationCharts.UniformizationBaseChange

lemma **pow_le_comap** {φ : R →+* R'} (hI : I ≤ I'.comap φ) (n : ℕ) : I ^ n ≤ (I' ^ n).comap φ

lemma **thickSpec_comp** {φ : R →+* R'} (hI : I ≤ I'.comap φ) (e : ℕ) : thick R' I' e ≫ specOf φ = thickSpec hI e ≫ thick R I e

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.ExistsDimOne

theorem **mulA_two** {S : Scheme.{0}} (A : Over S) [GrpObj A] : mulA A 2 = lift (𝟙 A) (𝟙 A) ≫ μ[A]

theorem **mulA_zero** {S : Scheme.{0}} (A : Over S) [GrpObj A] : mulA A 0 = toUnit A ≫ η[A]

theorem **comp_mulA** {S : Scheme.{0}} (A : Over S) [GrpObj A] {T : Over S} (x : T ⟶ A) (n : ℕ) : x ≫ mulA A n = x ^ n

theorem **kerPow** : KerPowSt

theorem **mulPull** : MulPullSt

theorem **mulN** : MulNSt

theorem **exists_dim_one** : ∃ P : PolAV (Spec (CommRingCat.of ℂ)), P.g = 1

theorem **separates** : ∃ (P : PolAV (Spec (CommRingCat.of ℂ))) (x : Over.mk (𝟙 (Spec (CommRingCat.of ℂ))) ⟶ P.A), InK P 2 x ∧ ¬ InK P 1 x ∧ ¬ InK P 3 x

theorem **mulA_two_ne_id** : ∃ P : PolAV (Spec (CommRingCat.of ℂ)), mulA P.A 2 ≠ 𝟙 P.A

theorem **summary** : KerPowSt ∧ MulPullSt ∧ MulNSt ∧ (∃ (P : PolAV (Spec (CommRingCat.of ℂ))) (x : Over.mk (𝟙 (Spec (CommRingCat.of ℂ))) ⟶ P.A), InK P 2 x ∧ ¬ InK P 1 x ∧ ¬ InK P 3 x) ∧ ∃ P : PolAV (Spec (CommRingCat.of ℂ)), mulA P.A 2 ≠ 𝟙 P.A

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.ExistsTwoTorsionWeilDegenerate

theorem **laws** : WeilSt ∧ WeilNondegSt ∧ CharPairSt

theorem **lifts** : ∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResι V ≫ x

theorem **skew** {S : Scheme.{0}} (P : PolAV S) (k : ℕ) {T : Over S} {x y : T ⟶ P.A} {c : Γ(T.left, ⊤)ˣ} (h : WeilIs P.A P.L k x y c) : WeilIs P.A P.L k y x c⁻¹

theorem **mul_right** {S : Scheme.{0}} [IsLocallyNoetherian S] (P : PolAV S) (k : ℕ) {T : Over S} {x y y' : T ⟶ P.A} {c c' : Γ(T.left, ⊤)ˣ} (h : WeilIs P.A P.L k x y c) (h' : WeilIs P.A P.L k x y' c') : WeilIs P.A P.L k x (y * y') (c * c')

theorem **weilIs_zero** {S : Scheme.{0}} (P : PolAV S) {T : Over S} (x y : T ⟶ P.A) : WeilIs P.A P.L ((0 : ℕ) : ℤ) x y 1

theorem **isElliptic_curve** (K : Type) [Field K] (h : (63 : K) ≠ 0) : (curve K).IsElliptic

theorem **exists_order_two** (K : Type) [Field K] (h : (63 : K) ≠ 0) : ∃ (P : PolAV (Spec (CommRingCat.of K))) (x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A), P.g = 1 ∧ x ≠ 1 ∧ x ^ 2 = 1

theorem **exists_value_ne_one** : ∃ (P : PolAV (Spec (CommRingCat.of ℂ))) (x y : baseOver (Spec (CommRingCat.of ℂ)) ⟶ P.A) (c : _), x ^ 2 = 1 ∧ y ^ 2 = 1 ∧ WeilIs P.A P.L ((2 : ℕ) : ℤ) x y c ∧ c ≠ 1 ∧ ¬ WeilIs P.A P.L ((2 : ℕ) : ℤ) x y 1

theorem **nondeg_fails_at_zero** : ∃ (P : PolAV (Spec (CommRingCat.of ℂ))) (x : baseOver (Spec (CommRingCat.of ℂ)) ⟶ P.A), x ^ 0 = 1 ∧ (∀ y : baseOver _ ⟶ P.A, y ^ 0 = 1 → WeilIs P.A P.L ((0 : ℕ) : ℤ) x y 1) ∧ x ≠ 1

theorem **value_pow** {S : Scheme.{0}} [IsLocallyNoetherian S] (P : PolAV S) (k : ℕ) {T : Over S} {x y : T ⟶ P.A} {c : Γ(T.left, ⊤)ˣ} (hx : x ^ k = 1) (hy : y ^ k = 1) (h : WeilIs P.A P.L k x y c) : c ^ k = 1

theorem **exists_twoTorsion_weil_degenerate** : ∃ (K : Type) (_ : Field K) (_ : IsAlgClosed K) (P : PolAV (Spec (CommRingCat.of K))) (x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A), x ^ 2 = 1 ∧ (∀ y : baseOver _ ⟶ P.A, y ^ 2 = 1 → WeilIs P.A P.L ((2 : ℕ) : ℤ) x y 1) ∧ x ≠ 1

theorem **isotropic_at_sq** {S : Scheme.{0}} [IsLocallyNoetherian S] (P : PolAV S) (m : ℕ) {T : Over S} (hm : IsUnit ((m * m : ℕ) : Γ(T.left, ⊤))) {x y : T ⟶ P.A} (hx : x ^ m = 1) (hy : y ^ m = 1) : WeilIs P.A P.L (m * m) x y 1

theorem **levels_differ** : ∃ (P : PolAV (Spec (CommRingCat.of ℂ))) (x y : baseOver (Spec (CommRingCat.of ℂ)) ⟶ P.A), x ^ 2 = 1 ∧ y ^ 2 = 1 ∧ WeilIs P.A P.L (((2 : ℕ) : ℤ) * ((2 : ℕ) : ℤ)) x y 1 ∧ ¬ WeilIs P.A P.L ((2 : ℕ) : ℤ) x y 1

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.GtBaseChangeSharp

theorem **cube_pull_sharp** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [IsCommMonObj G] [GrpObj G'] [IsCommMonObj G'] (m : CubeMap G' G) (L : RigLB G) (c : CubicalStr G L) : ∃ (hL : (L.L.pull m.F).IsLB) (hr : (m.rigPull L.L L.rig).IsGen) (c' : CubicalStr G' (m.rigLB L hL hr)), c'.t = m.thetaPull L.L c.t

theorem **both** : GtBaseChangeSt ∧ CubePullSt

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.HomExtendsOverDVRHolds

theorem **homExtendsOverDVR_holds** : GlobalTheory.IndefiniteRowSideFullHecke.HomExtensionOverDVR.HomExtendsOverDVR

theorem **homExtendsOverDVR_iff** : GlobalTheory.IndefiniteRowSideFullHecke.HomExtensionOverDVR.HomExtendsOverDVR ↔ GlobalTheory.LN.HomExtSt

theorem **homExtSt_holds** : GlobalTheory.LN.HomExtSt

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.MulGenNegRight

theorem **res_neg** {X : Scheme.{u}} {U V : X.Opens} (h : V ≤ U) (a : Γ(X, U)ˣ) : res h (-a) = -res h a

theorem **pullU_neg** {X Y : Scheme.{u}} (f : Y ⟶ X) (U : X.Opens) (a : Γ(X, U)ˣ) : pullU f U (-a) = -pullU f U a

theorem **negGen_isGen** {X : Scheme.{u}} {L : LB X} {t : LB.Gen L} (h : t.IsGen) : (negGen t).IsGen

theorem **negGen_negGen** (t : LB.Gen L) : negGen (negGen t) = t

theorem **pullGen_neg** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (h : Z ⟶ Y) : LB.pullGen L φ ε (negGen t) h = negGen (LB.pullGen L φ ε t h)

theorem **mulGen_neg_left** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) : LB.mulGen L φ ε φ' ε' (negGen t) t' = negGen (LB.mulGen L φ ε φ' ε' t t')

theorem **mulGen_neg_right** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) : LB.mulGen L φ ε φ' ε' t (negGen t') = negGen (LB.mulGen L φ ε φ' ε' t t')

theorem **corr_neg** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' : LB.Gen (LB.LinComb L φ' ε')} (h : LB.Gen.Corr L φ ε φ' ε' t t') : LB.Gen.Corr L φ ε φ' ε' (negGen t) (negGen t')

theorem **not_both** (h2 : IsUnit (2 : Γ(S, ⊤))) [Nonempty S] (L : RigLB G) (c : CubicalStr G L) (ψ : ULift.{u} (Fin 3) → ((G3 G).left ⟶ (G2 G).left)) (law : LB.Gen (LB.LinComb (abelian₂ G L.L) ψ lawε)) (h : LawIsCube G L c ψ law) : ¬ LawIsCube G L c ψ (negGen law)

theorem **linComb_g_one** {X Y : Scheme.{u}} (L : LB X) (hL : ∀ i j, L.g i j = 1) {K : Type u} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (i j) : (LB.LinComb L φ ε).g i j = 1

theorem **isGen_one'** {X : Scheme.{u}} (L : LB X) (hL : ∀ i j, L.g i j = 1) : LB.Gen.IsGen (L

theorem **isLB_of_g_one** {X : Scheme.{u}} (L : LB X) (hL : ∀ i j, L.g i j = 1) (hc : ⊔ i, L.U i = ⊤) : L.IsLB

theorem **lam_g_one** (i j) : (abelian₂ G (oneRig G).L).g i j = 1

theorem **main** (h2 : IsUnit (2 : Γ(S, ⊤))) [Nonempty S] : (LawIsCube G _ (oneCub G) _ (oneBiext G).law1 ∧ LawIsCube G _ (oneCub G) _ (oneBiext G).law2) ∧ ¬ LawIsCube G _ (oneCub G) _ (negBiext G (oneBiext G)).law1

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.NotRelAmpleOfNotIsProper

theorem **linComb_g_eq_one** {X Y : Scheme.{0}} {K : Type} [Fintype K] (L : LB X) (h : ∀ i j, L.g i j = 1) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (j j' : (LB.LinComb L φ ε).ι) : (LB.LinComb L φ ε).g j j' = 1

theorem **exists_gen_of_g_eq_one** {X : Scheme.{0}} (L : LB X) (h : ∀ i j, L.g i j = 1) : ∃ t : LB.Gen L, t.IsGen

theorem **rigidified_sym_triv** : Rigidified A (trivLB A.left) ∧ Out.Sym A (trivLB A.left)

theorem **rigidified_sym_noChart** : Rigidified A (PEmpty, PEmpty.elim, fun i => i.elim) ∧ Out.Sym A (PEmpty, PEmpty.elim, fun i => i.elim)

theorem **exists_polAV_dim_of_isEmpty** (S : Scheme.{0}) [IsEmpty S] (n : ℕ) : ∃ P : PolAV S, P.g = n

theorem **dim_eq_of_nonempty** {S : Scheme.{0}} (s : S) (P : PolAV S) {n : ℕ} (h : SmoothOfRelativeDimension n P.A.hom) : P.g = n

Development.AbelianVarietyTheory.Counterexamples.AbelianLawCounterexamples.WeilIsUniqueOfLocal

theorem **weilIs_noCharts** (k : ℤ) (x y : T ⟶ A) (c : Γ(T.left, ⊤)ˣ) : WeilIs A (noCharts A.left) k x y c

theorem **not_isLB_noCharts** (X : Scheme.{0}) [Nonempty X] : ¬ (noCharts X).IsLB

theorem **weilIs_unique_of_local** (p : T.left) (hp : ∀ V : T.left.Opens, p ∈ V → V = ⊤) (hL : L.IsLB) {x y : T ⟶ A} {c c' : Γ(T.left, ⊤)ˣ} (h : WeilIs A L k x y c) (h' : WeilIs A L k x y c') : unitRes (T

Development.AbelianVarietyTheory.Counterexamples.AdelicSplittingExamples.LfacDenominatorNeZero

theorem **sigma_invol** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

theorem **sigma_ne_id** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) : σ ≠ AlgHom.id F E

theorem **placeIdem_mul_self** {v : HeightOneSpectrum (𝒪 F)} {ε : AF F} (h : IsPlaceIdem F v ε) : ε * ε = ε

theorem **idem_mul_self** {σ : E →ₐ[F] E} {P : HeightOneSpectrum (𝒪 E)} {e f : AE F E} (h : IsPrimeIdem σ P e f) : e * e = e

theorem **isCMExt_of_isCMField** (K : Type) [Field K] [NumberField K] [IsCMField K] : IsCMExt (maximalRealSubfield K) K (IsCMField.complexConj K).toAlgHom

Development.AbelianVarietyTheory.Counterexamples.AdelicSplittingExamples.LocalConditionsOfGlobal

theorem **norm_one_tmul_eq** {F E K : Type} [Field F] [Field E] [Field K] [Algebra F E] [Algebra F K] [Module.Finite F E] (y : E) : Algebra.norm K ((1 : K) ⊗ₜ[F] y) = algebraMap F K (Algebra.norm F y)

theorem **isUnit_of_norm_eq** {F E K : Type} [Field F] [Field E] [Field K] [Algebra F E] [Algebra F K] [Module.Finite F E] {x : F} (hx : x ≠ 0) {y : K ⊗[F] E} (h : Algebra.norm K y = algebraMap F K x) : IsUnit y

theorem **norm_of_root** {F E K : Type} [Field F] [Field E] [Field K] [Algebra F E] [Algebra F K] [Module.Finite F E] {a r : K} (h : r ^ Module.finrank F E = a) : ∃ y : K ⊗[F] E, Algebra.norm K y = a

theorem **norm_complex** {F E : Type} [Field F] [Field E] [Algebra F E] [Algebra F ℂ] [Module.Finite F E] (a : ℂ) : ∃ y : ℂ ⊗[F] E, Algebra.norm ℂ y = a

theorem **norm_real_pos** {F E : Type} [Field F] [Field E] [Algebra F E] [Algebra F ℝ] [Module.Finite F E] {a : ℝ} (ha : 0 < a) : ∃ y : ℝ ⊗[F] E, Algebra.norm ℝ y = a

theorem **norm_pos_of_not_extends** {F E : Type} [Field F] [Field E] [Algebra F E] [CharZero F] [Module.Finite F E] (σ : F →+* ℝ) (hσ : ¬ ∃ τ : E →+* ℝ, τ.comp (algebraMap F E) = σ) {y : E} (hy : y ≠ 0) : 0 < σ (Algebra.norm F y)

theorem **local_conditions_of_global** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (x : F) (hx : x ≠ 0) (h : ∃ y : E, Algebra.norm F y = x) : (∀ v : HeightOneSpectrum (𝒪 F), ∃ y : TensorProduct F (v.adicCompletion F) E, Algebra.norm (v.adicCompletion F) y = algebraMap F (v.adicCompletion F) x) ∧ (∀ σ : F →+* ℝ, (¬ ∃ τ : E →+* ℝ, τ.comp (algebraMap F E) = σ) → 0 < σ x)

theorem **exists_norm_neg** {F E : Type} [Field F] [Field E] [Algebra F E] [Module.Finite F E] [IsGalois F E] (τ : E →+* ℝ) : ∃ y : E, τ (algebraMap F E (Algebra.norm F y)) < 0

theorem **norm_real_of_extends** {F E : Type} [Field F] [Field E] [Algebra F E] [Algebra F ℝ] [Module.Finite F E] [IsGalois F E] (τ : E →+* ℝ) (hτ : τ.comp (algebraMap F E) = algebraMap F ℝ) {a : ℝ} (ha : a ≠ 0) : ∃ y : ℝ ⊗[F] E, Algebra.norm ℝ y = a

theorem **local_norm_nonneg** {F E : Type} [Field F] [Field E] [Algebra F E] [Algebra F ℝ] [CharZero F] [Module.Finite F E] (hσ : ¬ ∃ τ : E →+* ℝ, τ.comp (algebraMap F E) = algebraMap F ℝ) (y : ℝ ⊗[F] E) : 0 ≤ Algebra.norm ℝ y

theorem **real_clause_iff** {F E : Type} [Field F] [Field E] [Algebra F E] [Algebra F ℝ] [CharZero F] [Module.Finite F E] [IsGalois F E] {x : F} (hx : x ≠ 0) : ((¬ ∃ τ : E →+* ℝ, τ.comp (algebraMap F E) = algebraMap F ℝ) → 0 < algebraMap F ℝ x) ↔ ∃ y : ℝ ⊗[F] E, Algebra.norm ℝ y = algebraMap F ℝ x

theorem **neg_one_not_norm** (E : Type) [Field E] [NumberField E] [IsCyclotomicExtension {4} ℚ E] : IsGalois ℚ E ∧ IsCyclic (E ≃ₐ[ℚ] E) ∧ Module.finrank ℚ E = 2 ∧ ¬ ∃ y : E, Algebra.norm ℚ y = -1

theorem **exists_gaussian_field** : ∃ (E : Type) (_ : Field E) (_ : NumberField E), IsCyclotomicExtension {4} ℚ E

Development.AbelianVarietyTheory.Counterexamples.AdelicSplittingExamples.TwoPrimesOfSplitPl

theorem **hermA_eq** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (x y : Fin n → RA F E) : hermA c G x y = x ⬝ᵥ (fm F G *ᵥ (sg (FiniteAdeleRing (𝒪 F) F) c ∘ y))

theorem **hermA_smul_left** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (a : RA F E) (x y : Fin n → RA F E) : hermA c G (a • x) y = a * hermA c G x y

theorem **isUnA_iff** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (g : Matrix (Fin n) (Fin n) (RA F E)) : IsUnA c G g ↔ gᵀ * fm F G * g.map (sg (FiniteAdeleRing (𝒪 F) F) c) = fm F G

theorem **det_norm_one** (c : E ≃ₐ[F] E) {n : ℕ} {G : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) {g : Matrix (Fin n) (Fin n) (RA F E)} (h : IsUnA c G g) : g.det * sg (FiniteAdeleRing (𝒪 F) F) c g.det = 1

theorem **isUnA_zero** (c : E ≃ₐ[F] E) {n : ℕ} (g : Matrix (Fin n) (Fin n) (RA F E)) : IsUnA c 0 g

theorem **congAtA_level_one** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (D : ℕ) (k : Matrix (Fin n) (Fin n) (RA F E)) : CongAtA F L 1 D k

theorem **congAtA_mod_zero** {n N : ℕ} (hN : N ≠ 1) (L : Submodule (𝒪 E) (Fin n → E)) {k : Matrix (Fin n) (Fin n) (RA F E)} (h : CongAtA F L N 0 k) : ∀ z ∈ hat F (L : Set (Fin n → E)), k *ᵥ z = z

theorem **congAtA_bot** {n N D : ℕ} (k : Matrix (Fin n) (Fin n) (RA F E)) : CongAtA F (⊥ : Submodule (𝒪 E) (Fin n → E)) N D k

theorem **congAtA_mul** {n N D : ℕ} {L : Submodule (𝒪 E) (Fin n → E)} {k k' : Matrix (Fin n) (Fin n) (RA F E)} (h : CongAtA F L N D k) (h' : CongAtA F L N D k') : CongAtA F L N D (k * k')

theorem **suppSplit_zero** (c : E ≃ₐ[F] E) {n : ℕ} {h : Matrix (Fin n) (Fin n) (RA F E)} (hh : SuppSplit c 0 h) : h = 1

theorem **suppSplit_kill** (c : E ≃ₐ[F] E) {n N : ℕ} {h : Matrix (Fin n) (Fin n) (RA F E)} (hh : SuppSplit c N h) (a : FiniteAdeleRing (𝒪 F) F) (ha : ∀ v, SplitPl c v → (N : 𝒪 F) ∉ v.asIdeal → a v = 0) : (a ⊗ₜ[F] (1 : E)) • (h - 1) = 0

theorem **suppSplit_mul** (c : E ≃ₐ[F] E) {n N : ℕ} {h h' : Matrix (Fin n) (Fin n) (RA F E)} (hh : SuppSplit c N h) (hh' : SuppSplit c N h') : SuppSplit c N (h * h')

theorem **suppSplit_inv** (c : E ≃ₐ[F] E) {n N : ℕ} {h : GL (Fin n) (RA F E)} (hh : SuppSplit c N h.1) : SuppSplit c N (h⁻¹).1

theorem **two_primes_of_splitPl** (c : E ≃ₐ[F] E) {v : HeightOneSpectrum (𝒪 F)} (h : SplitPl c v) : ∃ P Q : Ideal (𝒪 E), P.IsPrime ∧ Q.IsPrime ∧ P ≠ Q ∧ P.comap (algebraMap (𝒪 F) (𝒪 E)) = v.asIdeal ∧ Q.comap (algebraMap (𝒪 F) (𝒪 E)) = v.asIdeal

theorem **splitPl_of_two_primes** [NumberField E] (c : E ≃ₐ[F] E) (hc : c ≠ 1) (h2 : Module.finrank F E = 2) {v : HeightOneSpectrum (𝒪 F)} {P Q : Ideal (𝒪 E)} (hP : P.IsPrime) (hQ : Q.IsPrime) (hne : P ≠ Q) (h1 : P.comap (algebraMap (𝒪 F) (𝒪 E)) = v.asIdeal) (h1' : Q.comap (algebraMap (𝒪 F) (𝒪 E)) = v.asIdeal) : SplitPl c v

Development.AbelianVarietyTheory.Counterexamples.AnisotropicRankZero.WittSphericalSectionLeftInvariantRationalParabol

theorem **isIwasawa** (l : Lvl V) (hl : l.Good) (h : HasIw d l.K) : IsIwasawa (absDet d) l.K

theorem **absDet_rat** (q : stab d) (hq : (q : GA D V) ∈ GQ V) : absDet d q = 1

theorem **witt_spherical_section_left_invariant_rational_parabolic** (l : Lvl V) (hl : l.Good) (h : HasIw d l.K) (s : ℂ) : LeftInv (stab d ⊓ GQ V) (sphericalSectionIwasawa d l.K s)

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.ExistsCovPinIff

theorem **exists_trivial** (hr : 1 ≤ Dw.p + Dw.q) (hA : l.A = 1) (Tup : Type) : ∃ d : DeltaDat Xo c n, d.pd = 1 ∧ d.I = ⊤ ∧ d.l₂ = l ∧ d.cδ = 1 ∧ CovPin Xo c (fun _ : Tup => d)

theorem **exists_covPin_iff** (hr : 1 ≤ Dw.p + Dw.q) (Tup : Type) [Nonempty Tup] : (∃ dδ : Tup → DeltaDat Xo c n, CovPin Xo c dδ) ↔ l.A = 1

theorem **covPin_comp** {Tup Tup' : Type} (f : Tup' → Tup) (dδ : Tup → DeltaDat Xo c n) (h : CovPin Xo c dδ) : CovPin Xo c (fun t => dδ (f t))

theorem **covPin_smul** {Tup : Type} (dδ : Tup → DeltaDat Xo c n) (h : CovPin Xo c dδ) (a : ∀ t, (dδ t).c₂.F) (ha : ∀ t, a t ≠ 0) : CovPin Xo c (fun t => { dδ t with cδ

theorem **pd_mem** (d : DeltaDat Xo c n) (h : d.IsLev) {P : Ideal (𝒪 E)} (hI : d.I ≤ P) : (d.pd : 𝒪 E) ∈ P

theorem **not_both** {a b : ℕ} (hab : Nat.Coprime a b) {P : Ideal (𝒪 E)} (hP : P ≠ ⊤) (ha : (a : 𝒪 E) ∈ P) : (b : 𝒪 E) ∉ P

theorem **level_not_mem** (d : DeltaDat Xo c n) (h : d.IsLev) {P : Ideal (𝒪 E)} (hP : P ≠ ⊤) (hI : d.I ≤ P) : (l.M : 𝒪 E) ∉ P

theorem **split_below** (d : DeltaDat Xo c n) (h : d.IsLev) {P : Ideal (𝒪 E)} (hP : P ≠ ⊤) (hI : d.I ≤ P) : ∃ a ∈ Algebra.adjoin ℤ {H.w0}, a ∈ P ∧ H.bar a ∉ P

theorem **prime_under** (d : DeltaDat Xo c n) (h : d.IsLev) {P : Ideal (𝒪 E)} (hP : P ≠ ⊤) (hI : d.I ≤ P) {q : ℕ} (hq : q.Prime) (hqP : (q : 𝒪 E) ∈ P) : q ∣ d.pd ∧ zeta q ∈ c.F ∧ q − 1 ≤ Module.finrank ℚ c.F

theorem **finite_primes** {Tup : Type} (dδ : Tup → DeltaDat Xo c n) (h : CovPin Xo c dδ) : {q : ℕ | q.Prime ∧ ∃ t, ∃ P : Ideal (𝒪 E), P ≠ ⊤ ∧ (dδ t).I ≤ P ∧ (q : 𝒪 E) ∈ P}.Finite

theorem **pins_force** (O : OMDat C R i Tp tup) {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ ι x₀} (h : O.Pins κ v₀ ι x₀) (t : Tp) (PP : Formalism.L2.DIV.Pl D.host.E) (hPP : PP ∈ R.Q (tup t)) : ((C.Q.l.M : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal ∧ (∃ a ∈ Algebra.adjoin ℤ {C.Q.H.w0}, a ∈ PP.asIdeal ∧ C.Q.H.bar a ∉ PP.asIdeal) ∧ ∀ q : ℕ, q.Prime → (q : 𝒪 D.host.E) ∈ PP.asIdeal → zeta q ∈ O.c'.F ∧ q − 1 ≤ Module.finrank ℚ O.c'.F

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.ExistsStrictAnti

theorem **exists_strictAnti** (r : ℕ) : ∃ a : Fin r → ℤ, StrictAnti a

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.NotAdmissibleWithAmbientTop

theorem **atPlace_eq** (D : D7) (k : Fin 2) (v : D.nd.PlG k) (a) : atPlace D k v a = plHom D k v a

theorem **atPlace_central** (D : D7) (k : Fin 2) (v : D.nd.PlG k) (a : MonoidAlgebra ℂ (GL (Fin D.R.r) (ZMod (D.host.g k)))) (ha : ∀ b, a * b = b * a) : atPlace D k v a ∈ Subalgebra.center ℂ (MonoidAlgebra ℂ (G0g D.nd))

theorem **exists_acts** (D : D7) {V : Type} [AddCommGroup V] [Module ℂ V] (cen : Subalgebra.center ℂ (MonoidAlgebra ℂ (G0g D.nd)) →ₐ[ℂ] Module.End ℂ V) (k : Fin 2) (v : D.nd.PlG k) (F : V) (hF : F ≠ 0) : ∃ f, cen ⟨_, atPlace_central D k v _ (D.ntypes.eSer_central k f)⟩ F ≠ 0

theorem **exists_onMt** (D : D7) (μ j) (k : Fin 2) (v : D.nd.PlG k) (F : D.what₂.Mt μ j) (hF : F ≠ 0) : ∃ f, D.ntypes.onMt μ j (atPlace D k v (D.ntypes.eSer k f)) F ≠ 0

theorem **exists_onAmb** (D : D7) (μ) (k : Fin 2) (v : D.nd.PlG k) (F : D.what₂.Amb μ) (hF : F ≠ 0) : ∃ f, D.ntypes.onAmb μ (atPlace D k v (D.ntypes.eSer k f)) F ≠ 0

private theorem **exists_equiv_monicOf_of_perm** {K : Type} [Field K] (N : ℕ) (π : Equiv.Perm K[X]) (hπ : ∀ f, ((π f).Monic ∧ (π f).natDegree = N) ↔ (f.Monic ∧ f.natDegree = N)) : ∃ e : (Fin N → K) ≃ (Fin N → K), ∀ b, Formalism.CyclotomicNorms.monicOf (e b) = π (Formalism.CyclotomicNorms.monicOf b)

private theorem **exists_equiv_monicOf_sigma_places** (D : D7) [∀ k, Fact (D.host.g k).Prime] (π : ∀ k : Fin 2, Equiv.Perm (ZMod (D.host.g k))[X]) (mon : ∀ k f, ((π k f).Monic ∧ (π k f).natDegree = D.R.r) ↔ (f.Monic ∧ f.natDegree = D.R.r)) : ∃ e : (∀ kv : (Σ k : Fin 2, D.nd.PlG k), (Fin D.R.r → ZMod (D.host.g kv.1))) ≃ (∀ kv : (Σ k : Fin 2, D.nd.PlG k), (Fin D.R.r → ZMod (D.host.g kv.1))), ∀ a kv, Formalism.CyclotomicNorms.monicOf (e a kv) = π kv.1 (Formalism.CyclotomicNorms.monicOf (a kv))

theorem **relabel_types_e0** (D : D7) (P : Relabel D) : (relTypes D P).e0 = D.ntypes.e0

theorem **not_admissible_relabel** (D : D7) (P : Relabel D) (h : adm3 D) (μ j) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (σ : D.ntypes.Par) (F) (hF : F ∈ D.ntypes.MtPar μ j σ) (h0 : F ≠ 0) (k : Fin 2) (v : D.nd.PlG k) (hm : P.π k (D.ntypes.labPar k σ v) ≠ D.ntypes.labPar k σ v) : (rel D P).what₂ = D.what₂ ∧ (rel D P).ntypes.e0 = D.ntypes.e0 ∧ ¬ adm3 (rel D P)

theorem **galois_rel** (D : D7) (P : Relabel D) (h : adm3 D) (hc : ∀ τ k f, IsLabel D k f → P.π k (labAct D τ k f) = labAct D τ k (P.π k f)) (τ : ℂ ≃+* ℂ) (k : Fin 2) (f) (hf : IsLabel (rel D P) k f) : ((rel D P).ntypes.eSer k f).coeff.mapRange τ (map_zero τ) = ((rel D P).ntypes.eSer k (labAct (rel D P) τ k f)).coeff

theorem **deg_rev** {q : K[X]} (h : q.coeff 0 ≠ 0) : q.reverse.natDegree = q.natDegree

theorem **rev_rev** {q : K[X]} (h : q.coeff 0 ≠ 0) : q.reverse.reverse = q

theorem **coeff_of_dvd** {q f : K[X]} (h : q ∣ f) (hf : f.coeff 0 ≠ 0) : q.coeff 0 ≠ 0

theorem **rev_dvd** {q f : K[X]} (h : q ∣ f) : q.reverse ∣ f.reverse

theorem **unit_of_rev** {q : K[X]} (h : q.coeff 0 ≠ 0) (hu : IsUnit q.reverse) : IsUnit q

theorem **dl_lab** {N : ℕ} {f : K[X]} (h : f.Monic ∧ f.natDegree = N ∧ f.coeff 0 ≠ 0) : (dl f).Monic ∧ (dl f).natDegree = N ∧ (dl f).coeff 0 = (f.coeff 0)⁻¹

theorem **dl_dl** {N : ℕ} {f : K[X]} (h : f.Monic ∧ f.natDegree = N ∧ f.coeff 0 ≠ 0) : dl (dl f) = f

theorem **dvd_dl** {q f : K[X]} (h : q ∣ f) : q.reverse ∣ dl f

theorem **ptp_inv** (l : ℕ) {L : Type} [Field L] (x : L) : Formalism.CyclotomicNorms.primeToPart l x⁻¹ = (Formalism.CyclotomicNorms.primeToPart l x)⁻¹

theorem **root_dl** {g : ℕ} [Fact g.Prime] {N : ℕ} {f : (ZMod g)[X]} (h : f.Monic ∧ f.natDegree = N ∧ f.coeff 0 ≠ 0) {x : Formalism.CyclotomicNorms.Fbar g} (hx : x ∈ Formalism.CyclotomicNorms.rootsBar (dl f)) : x⁻¹ ∈ Formalism.CyclotomicNorms.rootsBar f

private theorem **irreducible_reverse_of_coeff_zero_ne_zero** {q : K[X]} (h0 : q.coeff 0 ≠ 0) (hq : Irreducible q) : Irreducible q.reverse

theorem **good_dl** {N g : ℕ} [Fact g.Prime] {PP J : Type} (f : PP → (ZMod g)[X]) (χ : PP → J → (ZMod g)ˣ) (h : Formalism.CyclotomicNorms.IsGood N g f χ) : Formalism.CyclotomicNorms.IsGood N g (fun v => dl (f v)) (fun v j => (χ v j)⁻¹)

theorem **label_dual** (D : D7) (k : Fin 2) {f} (h : IsLabel D k f) : IsLabel D k (dualLab D k f)

theorem **dualFun_invol** (D : D7) (k : Fin 2) : Function.Involutive (dualFun D k)

theorem **good_dualFun** (D : D7) (t : Tup2 D) (h : D.ntypes.goodTuple t) : D.ntypes.goodTuple (fun k v => dualFun D k (t k v))

theorem **not_admissible_dual_relabel** (D : D7) (h : adm3 D) (μ j) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (σ : D.ntypes.Par) (F) (hF : F ∈ D.ntypes.MtPar μ j σ) (h0 : F ≠ 0) (k : Fin 2) (v : D.nd.PlG k) (hl : IsLabel D k (D.ntypes.labPar k σ v)) (hm : dualLab D k (D.ntypes.labPar k σ v) ≠ D.ntypes.labPar k σ v) : (rel D (dualRel D)).what₂ = D.what₂ ∧ (rel D (dualRel D)).ntypes.e0 = D.ntypes.e0 ∧ ¬ adm3 (rel D (dualRel D))

theorem **not_admissible_with_ambient_top** (D : D7) (μ) (F : D.what₂.Amb μ) (h0 : F ≠ 0) (σ σ' : D.ntypes.Par) (k : Fin 2) (v : D.nd.PlG k) (hne : dualLab D k (D.ntypes.labPar k σ v) ≠ dualLab D k (D.ntypes.labPar k σ' v)) : ¬ adm3 (GlobalTheory.GoodProjectorTest.withAmb D fun _ _ => ⊤)

theorem **admissible_with_ambient_bot** (D : D7) (h : adm3 D) : adm3 (GlobalTheory.GoodProjectorTest.withAmb D fun _ _ => ⊥)

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.NotExistsExponentConditionCard

theorem **conj_conj'** {R : Ranks} (H : Host R) (x : H.E) : H.conj (H.conj x) = x

theorem **isConj_symm** {R : Ranks} {H : Host R} {P P' : HeightOneSpectrum (𝒪 H.E)} (h : IsConjPl H P P') : IsConjPl H P' P

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.NotFiniteIntegralClosure

theorem **not_finite_integralClosure** : ¬ Module.Finite ℤ (integralClosure ℤ ℂ)

theorem **lattice_eq_bot** {V : Type} [AddCommGroup V] [Module ℂ V] (MM : AddSubgroup V) (hs : ∀ a : ℂ, IsIntegral ℤ a → ∀ x ∈ MM, a • x ∈ MM) (hfg : Module.Finite ℤ MM) : MM = ⊥

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.NotVstdOfGramOne

theorem **mem_latC** (V : HSp D) (x : Fin V.rank → D.host.E) : x ∈ latC V ↔ x ∈ L0c D V

theorem **latC_coe** (V : HSp D) : (latC V : Set _) = L0c D V

theorem **hLat_one** (V : HSp D) : Bridge.hLat V 1 = latC V

theorem **vstd_iff** (i : Fin D.R.m) : VStd D i ↔ (Vc D i).gram = 1 ∧ ∀ x : Fin (Vc D i).rank → D.host.E, ∑ a, x a • (Vc D i).frame a ∈ (Vc D i).L0 ↔ ∀ a, IsIntegral ℤ (x a)

theorem **orthoSign** {E : Type} [Field E] [NumberField E] [StarRing E] (N : ℕ) (b : Bool) : Formalism.HermitianLatticesTheory.IsOrthoLattice (if b then -(diagForm (E

theorem **scaled_gram** {E : Type} [Field E] [NumberField E] [StarRing E] (N : ℕ) (u : Eˣ) (hu : star (u : E) * u = 1) : (scaled N u).gram = 1

theorem **scaled_sig** {E : Type} [Field E] [NumberField E] [StarRing E] (N : ℕ) (u : Eˣ) (hu : star (u : E) * u = 1) : (scaled N u).SigOK

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.PlaceExOfSurjective

theorem **exists_ratMat** (V : HSp D) (γ : GQ V) : ∃ γE, Bridge.IsRatMat V γ γE

theorem **ext_val** (V : HSp D) (C : Bridge.CoefDat D V) {S : CSD D.what₂.LL} {J : Subgroup (Bridge.Gf V)} {χ : J →* ℂˣ} (P : Bridge.ClassPres (Bridge.ratF V) S D.what₂.embLL C.ρ J χ) {lam : D.what₂.Tau → DomWeight V.rank} {c0 : VV D.what₂.LL lam →ₛₗ[D.what₂.embLL] C.W} {rho : GL (Fin V.rank) D.host.E →* Module.End D.what₂.LL (VV D.what₂.LL lam)} (hm : Bridge.CoefMapTie V C c0 rho) {pair nab del} (hc : Bridge.CoefTie V C P lam c0 pair nab del) (f : S.Cl → S.V) (g : Bridge.Gf V) : ∃ k v, P.ext f g = ((χ k : ℂˣ) : ℂ) • c0 v

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.PremisesModel

private theorem **isUnit_add_mul_of_mem_maximalIdeal** {O : Type*} [CommRing O] [IsLocalRing O] {ϖ : O} (hϖ : ϖ ∈ IsLocalRing.maximalIdeal O) {u : O} (hu : IsUnit u) (c : O) : IsUnit (u + ϖ * c)

private theorem **isUnit_diag_dvd_lower_mul** {O : Type*} [CommRing O] [IsLocalRing O] {ϖ : O} (hϖ : ϖ ∈ IsLocalRing.maximalIdeal O) {r : ℕ} (A B : Matrix (Fin r) (Fin r) O) (hA : ∀ i, IsUnit (A i i)) (hB : ∀ i, IsUnit (B i i)) (hA' : ∀ i j, j < i → ϖ ∣ A i j) (hB' : ∀ i j, j < i → ϖ ∣ B i j) : (∀ i, IsUnit ((A * B) i i)) ∧ ∀ i j, j < i → ϖ ∣ (A * B) i j

private theorem **isUnit_det_of_isUnit_diag_of_dvd_lower** {O : Type*} [CommRing O] [IsLocalRing O] {ϖ : O} (hϖ : ϖ ∈ IsLocalRing.maximalIdeal O) {ι : Type*} [Fintype ι] [DecidableEq ι] [LinearOrder ι] (N : Matrix ι ι O) (hd : ∀ i, IsUnit (N i i)) (hl : ∀ i j, j < i → ϖ ∣ N i j) : IsUnit N.det

private theorem **card_filter_sub_le_eq** (r k : ℕ) (hk : k ≤ r) : (univ.filter fun j : Fin r => r − k ≤ (j : ℕ)).card = k

private theorem **sum_filter_lt_sum_of_strictAnti** {r : ℕ} {a : Fin r → ℕ} (ha : StrictAnti a) (k : ℕ) (s : Finset (Fin r)) (hs : s.card = (univ.filter fun j : Fin r => r − k ≤ (j : ℕ)).card) (hne : s ≠ univ.filter fun j : Fin r => r − k ≤ (j : ℕ)) : ∑ j ∈ univ.filter (fun j : Fin r => r − k ≤ (j : ℕ)), a j < ∑ j ∈ s, a j

theorem **_root_.GlobalTheory.IwahoriNewtonTest.intersectionNumber_baseChange** : _root_.Formalism.PacketSums.IwahoriNewton

theorem **premises_model** : (2 : ℤ_[2]) ∈ IsLocalRing.maximalIdeal ℤ_[2] ∧ (2 : ℤ_[2]) ≠ 0 ∧ StrictAnti (![1, 0] : Fin 2 → ℕ) ∧ (∀ i, IsUnit ((!![1, 1; 2, 1] : Matrix (Fin 2) (Fin 2) ℤ_[2]) i i)) ∧ (∀ i j, j < i → (2 : ℤ_[2]) ∣ (!![1, 1; 2, 1] : Matrix (Fin 2) (Fin 2) ℤ_[2]) i j)

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.StatementOfTied

theorem **places_nonempty** (D : D7) : Nonempty D.what₂.Tau ∧ Nonempty D.what₂.Word

Development.AbelianVarietyTheory.Counterexamples.DegenerateExamples.TreeLeafParameterArithmeticOfBounds

theorem **degenerate_numbers** (O : FunData K Df) (C b : ℝ) : (degenerate O).PPF C b = 0 ∧ (degenerate O).Sig0 C b = 0 ∧ (degenerate O).cS = 0

theorem **degenerate_leaves** (O : FunData K Df) (C b : ℝ) (hC : 1 ≤ C) (hb : 1 ≤ b) : PrimePowerModuliSumBound (degenerate O) C b ∧ LeafCS (degenerate O) b ∧ LeafWt (degenerate O) C b

theorem **weight_free** (O : FunData K Df) (C b : ℝ) : ¬ LeafWt { O with Bw

theorem **exponent_free** (O : FunData K Df) (C x b : ℝ) (hx : 1 < x) : ∃ CH : ℝ, LeafPol O C x CH b

theorem **ga_mono** (O : FunData K Df) {C C' : ℝ} (h : C ≤ C') : O.ga C ≤ O.ga C'

theorem **le_ga** (O : FunData K Df) {C : ℝ} (hC : 0 ≤ C) : C * (O.N : ℝ) ^ 3 ≤ O.ga C

theorem **ga_gt** (O : FunData K Df) (hN : 1 ≤ O.N) (b C : ℝ) (hC : 0 ≤ C) (hb : b < C) : b < O.ga C

theorem **tree_leaf_parameter_arithmetic_of_bounds** (O : FunData K Df) (C b : ℝ) (hC : 1 ≤ C) (h1 : leafHypothesesConst O) (h7 : LatticeConstantBound O b) (hb : (O.m : ℝ) * O.f * (O.N : ℝ) ^ 2 ≤ b) (hN : 80 * O.r ^ 4 ≤ O.N) (hL : 2 * Real.log (2 * C) ≤ Real.log b) : DegreeAndSizeBounds O C b

theorem **leaf_bounds_of_log_sizes** (O : FunData K Df) [DecidableEq O.Sq] (C b : ℝ) (hL : 1 ≤ Real.log b) (hf : 1 ≤ O.f) (h0 : O.sv0 ∈ O.S0) (hq : ∀ q ∈ O.S0, IsPrimePow (O.qq q)) (hq0 : Real.log (O.qq O.sv0) ≤ Real.log b) (hS : ∑ q ∈ O.S0.erase O.sv0, Real.log (O.qq q) ≤ O.f * (20 * Real.log b + 2)) (hc : ∀ q ∈ O.S0, O.cc q ≤ 2) (hc0 : O.cc O.sv0 ≤ 1) (hP : 45 * O.f * Real.log b ≤ O.PPF C b) : PrimePowerModuliSumBound O C b ∧ LeafCS O b

Development.AbelianVarietyTheory.Counterexamples.HeckeCharacterExamples.ClusteredLetterwiseBoxes

theorem **nt_or_allThick** {n : ℕ} [NeZero n] (l d : ℕ) (Wd : NWord n) (hT : ∀ a, ThickN Wd.b Wd.η Wd.sz a ∨ Wd.sz a ≤ Wd.w) (hA : Rec.WordsAT l Wd d) : Rec.WordsNT Wd ∨ ∀ a, ThickN Wd.b Wd.η Wd.sz a

theorem **win_eq** {n : ℕ} (Wd : NWord n) (a : ZMod n) : (toDat Wd).win a = Wd.win a

theorem **hB_le** {D : D7} (L : SizeData₂ D) : Rec.hB L ≤ TwistedPlaces.hB L

theorem **recS_isTwistedDual_of_isTwistedDual** {w : HeightOneSpectrum (𝒪 D.host.E)} {hw : w ∈ placesOver D.host.E (D.nd.l i)} {Cx : TruncatedSeriesCountingIdentities.CoefExt D i w hw} {T : LatW i w hw Cx} (h : Formalism.MainNotions.N.IsTwistedDualOfGaloisRep i ρ π w hw Cx T) : RecS.IsTwistedDualOfGaloisRep i ρ π w hw Cx T

theorem **base_of** {w : HeightOneSpectrum (𝒪 D.host.E)} {hw : w ∈ placesOver D.host.E (D.nd.l i)} {Cx : TruncatedSeriesCountingIdentities.CoefExt D i w hw} {hG : Good (D.nbr i w hw).Kw 1 Cx.LL} {Wd : NWord (RecS.nLet i w hw)} (h : RecS.WordBase L HW i r ρ π w hw Cx hG Wd) : Formalism.MainNotions.N.WordBase L HW i r ρ π w hw Cx hG (toDat Wd) where

theorem **frozenB_of** (C₂ : ℝ) (h : RecS.JointB L HW C₂) (h1 : InertUnr D (D.nd.l i)) (h2 : Rec.IsRowB L (D.nd.l i)) (hP : RecS.PtIn i r ρ π) : LocalRepresentationBoundCondition L HW C₂ i r ρ π

theorem **frozenCD_of** (h : RecS.JointCD L HW) (h1 : InertUnr D (D.nd.l i)) (hP : RecS.PtIn i r ρ π) : LocalRepresentationCongruenceCondition L HW i r ρ π

theorem **inert_iff** (D : D7) (p : ℕ) : InertUnr D p ↔ TwistedPlaces.InertK D p

theorem **order_eq** (D : D7) (l : ℕ) [hl : Fact l.Prime] (hlN : ¬ l ∣ D.host.condN) : orderOf (D.R.hH (DV.frobG D l).2) = TwistedPlaces.ordH D l

theorem **frozenBCD_of** {D : D7} (L : SizeData₂ D) (HW : HWOf D.loc) (X : KillingConditions.HubData D) (R : KillingConditions.RowK) (hB : RecS.JointB L HW R.C₂) (hC : RecS.JointCD L HW) (hP : ∀ i, TwistedPlaces.TwLoc D i → TwistedPlaces.InertK D (D.nd.l i) → ∀ π ∈ D.nd.Zgood i, RecS.PtIn i (X.r i) (X.ρ i) π) : KillingConditions.JointParametersHubConditions L HW X R

Development.AbelianVarietyTheory.Counterexamples.HeckeCharacterExamples.InertPlacesGoodSet

theorem **inert_jacobi** {D : D7} {l : ℕ} (h : TwistedPlaces.InertK D l) : l ∉ D.host.small ∧ l ≠ D.host.q ∧ l ≠ D.host.d ∧ jacobiSym (-(D.host.d : ℤ)) l ≠ 1

theorem **eleven_le** {D : D7} {l : ℕ} (hl : l.Prime) (h : TwistedPlaces.InertK D l) (hs : 10 ≤ D.host.splitBound) : 11 ≤ l

theorem **rowL_of_wordsB** {C₂ : ℝ} (hA : WachCoordinates.Conclusion2 D C₂) (h11 : 11 ≤ D.nd.l i) {Wd : WordDat (nLet i w hw)} (hb : WordBase L HW i r ρ π w hw Cx hG Wd) (hB : WordsB L C₂ (D.nd.l i) Wd) : RowL i r ρ π w hw Cx (C₂ * ((nLet i w hw : ℝ) + 1) * D.R.r * ((WachCoordinates.hCor Wd.b Wd.w Wd.S Wd.η Wd.sz : ℝ) / ((D.nd.l i : ℝ) - 1) ^ (TwistedPlaces.jA L (D.nd.l i) + 1) + (((Wd.w + Wd.S + Wd.η : ℕ) : ℝ) + 1) * ((TwistedPlaces.jA L (D.nd.l i) : ℝ) + 1) + Real.logb (D.nd.l i) (WachCoordinates.hCor Wd.b Wd.w Wd.S Wd.η Wd.sz : ℝ)))

theorem **setting_of_wordsNT** (h2l : D.nd.l i ≠ 2) {Wd : WordDat (nLet i w hw)} (hb : WordBase L HW i r ρ π w hw Cx hG Wd) (hN : WordsNT Wd) (ft : Fintype (↥(D.nbr i w hw).Kw.1 →ₐ[ℚ_[D.nd.l i]] ↥Cx.LL.1)) (T₁ T₂ : LatW i w hw Cx) (F : LatHom T₁.L T₂.L) (h2 : IsTwistedDualOfGaloisRep i ρ π w hw Cx T₂) (h1 : IsSubOfTwistedDualSums i r w hw Cx T₁) (hF : F.Onto) : TruncatedSeriesCountingBounds.HidaSetting ((D.loc (D.nd.l i)).PD (D.nbr i w hw).Kw) (WW HW i w hw Cx hG) T₁ T₂ F Wd.b Wd.w Wd.S Wd.η Wd.m Wd.up Wd.sz Wd.Mi Wd.hi

theorem **setCS_of_wordsAT** (h11 : 11 ≤ D.nd.l i) {Wd : WordDat (nLet i w hw)} {d : ℕ} (hb : WordBase L HW i r ρ π w hw Cx hG Wd) (hC : WordBaseC (HW

Development.AbelianVarietyTheory.Counterexamples.HeckeCharacterExamples.PeriodBoundOfPlaceBound

theorem **vAdd_unit** (D : D7) (w : D.what₂.places.Pl) (s : 𝒪 D.what₂.LL) (hs : s ∉ (D.plEq w).asIdeal) : vAdd (D.plEq w) (s : D.what₂.LL) = 0

theorem **exists_place** (D : D7) (p : ℕ) (hp : p.Prime) : ∃ w : D.what₂.places.Pl, D.what₂.places.resChar w = p

Development.AbelianVarietyTheory.Counterexamples.HeckeCharacterExamples.SecondTwistCrystallineSmallOfJumpsOfCrys

theorem **isField_bdR** (D : D7) (l : ℕ) [Fact l.Prime] : IsField (D.loc l).R.BdR

theorem **card_small** : ((jT2 P j s).filter fun u : ℤ => |u| ≤ (n : ℤ)).card = (Finset.univ.filter fun i : Fin (R.k - 1) => |dj P j s ⌊P.v i⌋| ≤ (n : ℤ)).card + (Finset.univ.filter fun i : Fin R.m => |dj P j s (-⌊P.b i⌋)| ≤ (n : ℤ)).card

theorem **lows_small** (g : D.R.GG) (j : Fin D.R.m) (s : Bool) (i : Fin D.R.m) : |dj (hcAt D g) j s (-⌊(hcAt D g).b i⌋)| ≤ (etaHat L : ℤ)

theorem **cloud_large** (g : D.R.GG) (j : Fin D.R.m) (s : Bool) (h : (etaHat L : ℤ) < szT (hcAt D g) j s) (i : Fin (D.R.k - 1)) : ¬ |dj (hcAt D g) j s ⌊(hcAt D g).v i⌋| ≤ (etaHat L : ℤ)

theorem **count_eq** (g : D.R.GG) (j : Fin D.R.m) (s : Bool) (h : (etaHat L : ℤ) < szT (hcAt D g) j s) : ((jT2 (hcAt D g) j s).filter fun u : ℤ => |u| ≤ (etaHat L : ℤ)).card = D.R.m

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.BesselPeriodPositionOneEqZero

theorem **emb_isom** (X : ThetaX D) (t j : ℕ) (hj : j < D.R.m) : HSpace.IsIsomEmb (X.Xs t j) (X.Ys t j) (emb X t j)

theorem **emb_ne** (X : ThetaX D) (t j : ℕ) (hj : j < D.R.m) : ((X.Xs t j, X.Ys t j, emb X t j) : Tri D) ≠ tri0 D

theorem **cut_empty** (X : ThetaX D) (t j : ℕ) (hj : j < D.R.m) : IsEmpty ((cut X).Pos ((cut X).Xs t j) ((cut X).Ys t j) ((cut X).iXYs t j))

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.BlattnerShiftsSignedParameters

theorem **isEnum_unique** {n : ℕ} {S : SPar} {l l' : Fin n → ℝ} {e e' : Fin n → Bool} (h : IsEnum S l e) (h' : IsEnum S l' e') : l = l' ∧ e = e'

theorem **goodPar_add** {n p q : ℕ} {S : SPar} (h : GoodPar n p q S) : p + q = n

theorem **goodPar_ex** : GoodPar 2 1 1 {(1 / 2, true), (-1 / 2, false)}

theorem **lamSH_int** (R : Ranks) (P : HCParam R) (a : Fin (2 * R.rS + R.k + 1)) : ∃ m : ℤ, lamSH R P a - (((2 * R.rS + R.k + 1 : ℕ) : ℝ) - 1) / 2 = (m : ℝ)

theorem **isoDef_wrong_sign** {M : Type} [AddCommGroup M] [Module ℂ M] (π : Matrix (Fin 1) (Fin 1) ℂ → Module.End ℂ M) (x : ℚ) : isoDef 1 π {(x, false)} = ⊥

theorem **rho2C_two_one** : rhoOfIndex (p

theorem **blattner_add_rho2C** {p q : ℕ} {lam : Ix p q → ℚ} (h : IsParam lam) (a : Ix p q) : blattner lam a + rhoOfIndex a = lam a + (((Finset.univ.filter fun b => lam b < lam a).card : ℚ) - ((Finset.univ.filter fun b => lam a < lam b).card : ℚ)) / 2

theorem **blattner_inj** {p q : ℕ} {l l' : Ix p q → ℚ} (h : IsParam l) (h' : IsParam l') (e : blattner l = blattner l') : l = l'

theorem **blattner_sing** : blattner (fun _ : Ix 2 0 => (0 : ℚ)) = blattner rhoTwo

theorem **hasParam_sing** {T : GrpData 2 0} (R : DSRep T) : HasParam R (fun _ => 0) ↔ HasParam R rhoTwo

theorem **hasParam_zero** {T : GrpData 0 0} (R : DSRep T) (lam : Ix 0 0 → ℚ) : HasParam R lam

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ConjAtFixedIff

theorem **conj_fixed_iff** (hu : IsUnit H.det) (Z : Matrix (Fin n) (Fin n) ℂ) : -(H⁻¹ * Zᴴ * H) = Z ↔ Zᴴ * H + H * Z = 0

theorem **conjT_conj_mul** (hH : Hᴴ = H) (hu : IsUnit H.det) (Z : Matrix (Fin n) (Fin n) ℂ) : (-(H⁻¹ * Zᴴ * H))ᴴ * H = -(H * Z)

theorem **mul_conj** (hu : IsUnit H.det) (Z : Matrix (Fin n) (Fin n) ℂ) : H * (-(H⁻¹ * Zᴴ * H)) = -(Zᴴ * H)

theorem **re_mem** (hH : Hᴴ = H) (hu : IsUnit H.det) (Z : Matrix (Fin n) (Fin n) ℂ) : ((1 / 2 : ℂ) • (Z + -(H⁻¹ * Zᴴ * H)))ᴴ * H + H * ((1 / 2 : ℂ) • (Z + -(H⁻¹ * Zᴴ * H))) = 0

theorem **im_mem** (hH : Hᴴ = H) (hu : IsUnit H.det) (Z : Matrix (Fin n) (Fin n) ℂ) : ((1 / (2 * Complex.I)) • (Z - -(H⁻¹ * Zᴴ * H)))ᴴ * H + H * ((1 / (2 * Complex.I)) • (Z - -(H⁻¹ * Zᴴ * H))) = 0

theorem **exp_mem_UH** (hu : IsUnit H.det) {X : Matrix (Fin n) (Fin n) ℂ} (hX : Xᴴ * H + H * X = 0) (t : ℝ) : NormedSpace.exp (t • X) ∈ Formalism.QuantumDeterminants.UH H

theorem **gram_det_ne** : V.gram.det ≠ 0

theorem **gramAt_unit** (τ : D.what₂.Tau) : IsUnit (gramAt D V τ).det

theorem **gramAt_herm** (τ : D.what₂.Tau) : (gramAt D V τ)ᴴ = gramAt D V τ

theorem **conjAt_fixed_iff** (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : conjAt V τ Z = Z ↔ Z ∈ lieAt V τ

theorem **parts_mem** (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : (1 / 2 : ℂ) • (Z + conjAt V τ Z) ∈ lieAt V τ ∧ (1 / (2 * Complex.I)) • (Z - conjAt V τ Z) ∈ lieAt V τ

theorem **parts_sum** (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : (1 / 2 : ℂ) • (Z + conjAt V τ Z) + Complex.I • ((1 / (2 * Complex.I)) • (Z - conjAt V τ Z)) = Z

theorem **RderC_eq** (τ : D.what₂.Tau) {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) (φ : GA D V → ℂ) : RderC θ V τ X φ = Rder θ V τ X φ

theorem **oneP_spec** {X : ThetaX D} (T : ThetaTied D X) (hθ : IsInfFam (D

theorem **isAutFunSm_reSig** (s : (D.host.E →+* ℂ) → ℕ × ℕ) (φ : GA D V → ℂ) : IsAutFunSm θ (reSig V s) φ ↔ IsAutFunSm θ V φ

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.EmbRelExistsUnique

theorem **borel_NI** (D : D7) (V : HSp D) {r : ℕ} (C : Matrix (Fin V.rank) (Fin r) D.host.E) : BorelSpace (NI V C)

theorem **topGroup_GA** (D : D7) (V : HSp D) : IsTopologicalGroup (GA D V)

theorem **GA_ext** (V : HSp D) {g h : GA D V} (e : (matHom V g).1 = (matHom V h).1) : g = h

theorem **hT_coord** (V : HSp D) (x y : V.V) : V.hT x y = ∑ i, ∑ j, star (V.frame.repr x i) * V.gram i j * V.frame.repr y j

theorem **gram_pull** (A B : HSp D) (ι : B.V →ₗ[D.host.E] A.V) (hι : HSpace.IsIsomEmb B A ι) : cT (D.sp.cm.sig : D.host.E →+* D.host.E) (frameMat A B ι) * A.gram * frameMat A B ι = B.gram

theorem **gram_det_ne** (V : HSp D) : V.gram.det ≠ 0

theorem **split_gen** {E R : Type} [CommRing E] [CommRing R] (φ : E →+* R) {n r : ℕ} (P : Matrix (Fin n) (Fin r) E) (Q : Matrix (Fin r) (Fin n) E) (g₁ g₂ : Matrix (Fin n) (Fin n) R) (h₁ : g₁ * (1 - P * Q).map φ = g₂ * (1 - P * Q).map φ) (h₂ : g₁ * P.map φ = g₂ * P.map φ) : g₁ = g₂

theorem **tens_inj** {F E A B : Type} [Field F] [Field E] [Algebra F E] [CommRing A] [CommRing B] [Algebra F A] [Algebra F B] {n : ℕ} (m : Matrix (Fin n) (Fin n) ((A × B) ⊗[F] E)) (a : m.map (Algebra.TensorProduct.map (AlgHom.fst F A B) (AlgHom.id F E)) = 1) (b : m.map (Algebra.TensorProduct.map (AlgHom.snd F A B) (AlgHom.id F E)) = 1) : m = 1

theorem **embRel_unique** (A B : HSp D) (ι : B.V →ₗ[D.host.E] A.V) (hι : HSpace.IsIsomEmb B A ι) {g₁ g₂ : GA D A} {g' : GA D B} (h₁ : EmbRel A B ι g₁ g') (h₂ : EmbRel A B ι g₂ g') : g₁ = g₂

theorem **one_mem_KsphC** (V : HSp D) (S) : (1 : GA D V) ∈ KsphC V S

theorem **GfC_inf_GinfC** (V : HSp D) {g : GA D V} (h₁ : g ∈ GfC V) (h₂ : g ∈ GinfC V) : g = 1

theorem **ext_unitary** {R : Type} [CommRing R] (σ : R →+* R) {n r : ℕ} (J : Matrix (Fin n) (Fin n) R) (G : Matrix (Fin r) (Fin r) R) (P : Matrix (Fin n) (Fin r) R) (Q : Matrix (Fin r) (Fin n) R) (h1 : J * P = cT σ Q * G) (h2 : cT σ P * J = G * Q) (h3 : Q * P = 1) (h : Matrix (Fin r) (Fin r) R) (hh : cT σ h * G * h = G) : cT σ (1 + P * (h - 1) * Q) * J * (1 + P * (h - 1) * Q) = J

theorem **gram_herm** (V : HSp D) : cT (D.sp.cm.sig : D.host.E →+* D.host.E) V.gram = V.gram

theorem **cT_cT** {m p : ℕ} (M : Matrix (Fin m) (Fin p) D.host.E) : cT (D.sp.cm.sig : D.host.E →+* D.host.E) (cT (D.sp.cm.sig : D.host.E →+* D.host.E) M) = M

theorem **cT_map_gen** {F E R : Type} [Field F] [Field E] [Algebra F E] [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) {m p : ℕ} (M : Matrix (Fin m) (Fin p) E) : cT (sigR R σ) (M.map (inclR (F

theorem **ext_mul** {R : Type} [CommRing R] {n r : ℕ} (P : Matrix (Fin n) (Fin r) R) (Q : Matrix (Fin r) (Fin n) R) (h3 : Q * P = 1) (a b : Matrix (Fin r) (Fin r) R) (hab : a * b = 1) : (1 + P * (a - 1) * Q) * (1 + P * (b - 1) * Q) = 1

theorem **exists_gen** {F E R : Type} [Field F] [Field E] [Algebra F E] [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) {n r : ℕ} (J : Matrix (Fin n) (Fin n) E) (G : Matrix (Fin r) (Fin r) E) (P : Matrix (Fin n) (Fin r) E) (Gi : Matrix (Fin r) (Fin r) E) (hGi : G * Gi = 1) (hGi' : Gi * G = 1) (hG : cT (σ : E →+* E) P * J * P = G) (hJ : cT (σ : E →+* E) J = J) (hGh : cT (σ : E →+* E) G = G) (hP : cT (σ : E →+* E) (cT (σ : E →+* E) P) = P) (h : HR R σ G) : ∃ g : HR R σ J, g.1.1 * P.map (inclR (F

theorem **embRel_existsUnique** (A B : HSp D) (ι : B.V →ₗ[D.host.E] A.V) (hι : HSpace.IsIsomEmb B A ι) (g' : GA D B) : ∃! g : GA D A, EmbRel A B ι g g'

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.EntriesOfGaps

theorem **gapZ_nonneg** {p : ℕ} {l : Fin p → ℕ} (hl : Antitone l) (i : Fin p) : 0 ≤ WeightAlgebra.gapZ l i

theorem **isL_lWt** (ka : WeightFamilies.Wt D j) (T nW : ℕ) (hW : Even nW) (h : ∀ τ (i : Fin (D.R.sigW j).1), i.1 + 1 = (D.R.sigW j).1 → ((WeightFamilies.dimY D j : ℤ) + 2) * nW + 2 * D.R.r ≤ ka.aP τ i) : WeightAlgebra.IsL D ka (lWt D ka T nW hW) T nW

private lemma **ofGaps_eq_sub_mul** {p : ℕ} (a b g : Fin p → ℕ) (T : ℕ) (hg : ∀ k : Fin p, (g k : ℤ) = if (k : ℕ) + 1 < p then _root_.Formalism.MainNotions.WeightAlgebra.gapZ a k - T * _root_.Formalism.MainNotions.WeightAlgebra.gapZ b k else (a k : ℤ) - T * b k) (i : Fin p) : (_root_.Formalism.MainNotions.DIV.Cons.ofGaps g i : ℤ) = a i - T * b i

private lemma **mul_toNat_fdiv_add_two_le_of_add_two_le** {T K : ℕ} (hK : T + 2 ≤ K) {d : ℤ} (hd : 0 ≤ d) : (T : ℤ) * ((d + 2).fdiv K).toNat ≤ d

private lemma **gapZ_nonneg_of_antitone'** {p : ℕ} (a : Fin p → ℕ) (ha : Antitone a) (k : Fin p) : 0 ≤ _root_.Formalism.MainNotions.WeightAlgebra.gapZ a k

private lemma **mul_toNat_fdiv_le_of_le_of_nonneg** {T K : ℕ} (hK : T + 2 ≤ K) {y a : ℤ} (hy : y ≤ a) (ha : 0 ≤ a) : (T : ℤ) * (y.fdiv K).toNat ≤ a

theorem **_root_.GlobalTheory.RunWeightTest.isF'_fWt** (ka : _root_.Formalism.MainNotions.WeightFamilies.Wt D j) (T nW nP : ℕ) (hW : Even nW) (hP : Even nP) : _root_.Formalism.MainNotions.WeightAlgebra.IsF' ka (_root_.Formalism.MainNotions.DIV.Cons.lWt D ka T nW hW) (_root_.Formalism.MainNotions.DIV.Cons.fWt D ka (_root_.Formalism.MainNotions.DIV.Cons.lWt D ka T nW hW) T nP hP) T nP

theorem **main** (Tp : ParamsT D L P) (hK : DV.KapDef D) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : WeightAlgebra.IsL D (kaR Tp hK j) (lR Tp hK j) Tp.TO Tp.nWn ∧ WeightAlgebra.IsF' (kaR Tp hK j) (lR Tp hK j) (fR Tp hK j) Tp.TO Tp.nP ∧ kaR Tp hK j = WeightAlgebra.wadd (fR Tp hK j) (WeightAlgebra.wsmul Tp.TO (lR Tp hK j))

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ExistsExpoScalar

theorem **basis_exists** (X : Data E r) (cu : CuspFD X ρ) : ∃ bs, IsZBasis X cu bs

theorem **cover_one** (X : Data E r) (ξ : Fin r → E) (h : X.IsIso ξ) : ∃ cu : CuspFD X 1, ξ ∈ Submodule.span E (Set.range cu.e)

theorem **chi_zero** (X : Data E r) (γ : GL (Fin r) E) (hγ : γ ∈ X.Γ) : X.χ 0 γ = 1

theorem **isExpo_zero** (X : Data E r) (cu : CuspFD X ρ) : IsExpoFD X cu 0

theorem **killed_zero** (X : Data E r) (cu : CuspFD X (ρ + 1)) {c : ℝ} (hc : 0 ≤ c) : KilledFD X cu 0 c

theorem **exists_expo_scalar** (X : Data E r) (cu : CuspFD X ρ) : ∃ n : ℕ, 0 < n ∧ IsExpoFD X cu ((n : E) • (1 : Matrix (Fin ρ) (Fin ρ) E))

theorem **not_killed_scalar** (X : Data E r) (cu : CuspFD X ρ) {n : ℕ} (hn : 0 < n) {c : ℝ} (hc : c ≤ 0) : ¬ KilledFD X cu ((n : E) • (1 : Matrix (Fin ρ) (Fin ρ) E)) c

theorem **premises_model** : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (X : Data E 2), X.dom.Nonempty ∧ (∃ ξ, X.IsIso ξ) ∧ (∀ ξ, X.IsIso ξ → ∃ cu : CuspFD X 1, ξ ∈ Submodule.span E (Set.range cu.e)) ∧ (∀ cu : CuspFD X 1, ∃ bs, IsZBasis X cu bs) ∧ (∀ γ ∈ X.Γ, X.χ 0 γ = 1) ∧ ∃ (cu : CuspFD X 1) (χ χ' : Matrix (Fin 1) (Fin 1) E), IsExpoFD X cu χ ∧ KilledFD X cu χ 0 ∧ IsExpoFD X cu χ' ∧ ¬ KilledFD X cu χ' 0

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ExistsGoodFamily

theorem **jm_const** (n : ℕ) (s : Bool) : Formalism.REAL.Jm (fun _ : Fin n => s) = (if s then (1 : ℂ) else -1) • (1 : Matrix (Fin n) (Fin n) ℂ)

theorem **kset_const** (n : ℕ) (s : Bool) : Formalism.REAL.Kset (fun _ : Fin n => s) = UJset (fun _ : Fin n => s)

theorem **lowK_const** (n : ℕ) (s : Bool) (lam : Fin n → ℝ) (a : Fin n) : Formalism.QuantumDeterminants.lowK (fun _ : Fin n => s) lam a = lam a - (((n : ℝ) - 1) / 2 - (a : ℕ))

theorem **isDS_const_top** {n : ℕ} {s : Bool} {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] {lam : Fin n → ℝ} {σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)} (h : IsDS (fun _ => s) lam σ) : IsKTypeOf (fun _ => s) σ (Formalism.QuantumDeterminants.lowK (fun _ => s) lam) ⊤

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ExistsJunkLiftNotVanishing

theorem **hHom_add_left** (V W : HSp D) (x x' y : WA V W) : hHom V W (x + x') y = hHom V W x y + hHom V W x' y

theorem **hHom_add_right** (V W : HSp D) (x y y' : WA V W) : hHom V W x (y + y') = hHom V W x y + hHom V W x y'

theorem **symplH_add_left** (V W : HSp D) (x x' y : WA V W) : symplH V W (x + x') y = symplH V W x y + symplH V W x' y

theorem **symplH_add_right** (V W : HSp D) (x y y' : WA V W) : symplH V W x (y + y') = symplH V W x y + symplH V W x y'

theorem **heis_heis** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (w w' : WA V W) (f : WA V W → ℂ) : Heisenberg₂ ψ V W w (Heisenberg₂ ψ V W w' f) = ψ (symplH V W w w') • Heisenberg₂ ψ V W (w + w') f

theorem **heis_quasi** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (w₀ : WA V W) (f : WA V W → ℂ) (hf : ∀ γ w, f (ratW V W γ + w) = ψ (-(symplH V W (ratW V W γ) w)) * f w) (γ) (w : WA V W) : Heisenberg₂ ψ V W w₀ f (ratW V W γ + w) = ψ (-(symplH V W (ratW V W γ) w)) * Heisenberg₂ ψ V W w₀ f w

theorem **symplH_transpose** (V W : HSp D) (hne : V.rank ≠ W.rank) (x y : WA V W) : symplH W V x.transpose y.transpose = symplH V W x y

theorem **isThetaFun_transpose** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (hne : V.rank ≠ W.rank) (f : WA V W → ℂ) (hf : IsThetaFun ψ V W f) : IsThetaFun ψ W V (fun y : WA W V => f y.transpose)

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ExistsNotIntegralLie

theorem **spread_holds** : SpreadSt Xo c

theorem **all_hold** : IntFramesSt Xo c ∧ LieFreeSt Xo c ∧ SpreadSt Xo c

theorem **exists_framed** {S : Type} [Field S] [Algebra c.F S] (x : Sp S ⟶ Xo.X) (hx : x ≫ Xo.s = sOf c S) : ∃ y : FrPt Xo c S, y.x = x

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ExistsXiOKEvenRamified

theorem **conj_emb** (D : D7) (φ : D.host.E →+* ℂ) (y : D.host.E) : φ (GlobalTheory.Member.cF D y) = starRingEnd ℂ (φ y)

theorem **conj_conj** (D : D7) (y : D.host.E) : GlobalTheory.Member.cF D (GlobalTheory.Member.cF D y) = y

theorem **mem_congOne** (D : D7) (N : ℕ) (x : (AfE D)ˣ) : x ∈ congOne (GlobalTheory.Member.cF D) N ↔ CongOne N x ∧ NormOne x

theorem **fin_of_rat_tame** (D : D7) (N : ℕ) (hN : 0 < N) (xi : (AfE D)ˣ →* ℂˣ) (h1 : ∀ (x : (AfE D)ˣ) (a : D.host.E), (x : AfE D) = 1 ⊗ₜ a → NormOne x → xi x = 1) (h2 : ∀ x, NormOne x → CongOne N x → xi x = 1) : ∃ m : ℕ, 0 < m ∧ ∀ x, NormOne x → xi x ^ m = 1

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.FromTestIffOfValAtCx

theorem **evR_of_not_ft** {S : Type} [CommRing S] [Algebra c.F S] (f : Rul Xo c) (y : FrPt Xo c S) (hy : ¬ FT y) : evR Xo c f y = 0

theorem **evR_one** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (hy : FT y) : evR Xo c 1 y = 1

theorem **evR_map** (hWD : EvalWD (K

theorem **ex_zero** (i : Ix Dw) : ex Dw 0 0 i = 0

theorem **one_mem_sec** : (1 : Rul Xo c) ∈ sec Xo c 0 0

theorem **cabs_one** (σ : c.F →+* ℂ) : cabs (1 : CAt σ) = 1

theorem **cabs_zero** (σ : c.F →+* ℂ) : cabs (0 : CAt σ) = 0

theorem **fromTest_iff_of_valAtCx** (hcx : ValAtCx h yP) (Z : Comp Xo c) (P : Om Z →+* ℂ) : FT (genFr h Z) ↔ FT (yP Z P)

theorem **fromTest_of_nzSome** {f : Rul Xo c} (hN : NZSome h f) : ∃ Z, evC h Z f ≠ 0 ∧ FT (genFr h Z)

theorem **nzSome_one_iff** : NZSome h (1 : Rul Xo c) ↔ ∃ Z, FT (genFr h Z)

theorem **not_nzSome_zero** : ¬ NZSome h (0 : Rul Xo c)

theorem **laws_of_no_comp** (hne : IsEmpty (Comp Xo c)) (f : Rul Xo c) : PTNe Xo c ∧ ValAtCx h yP ∧ ¬ NZSome h f

theorem **supR_pos** [Fintype (Comp Xo c)] (hfn : FnNe Xo c fn) (hPT : PTNe Xo c) (hcx : ValAtCx h yP) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hS : f ∈ sec Xo c a n) (hB : SupBdd Xo c fn a n f) (hN : NZSome h f) (σ : c.F →+* ℂ) : 0 < supR Xo c fn σ a n f

theorem **pn_one_le** (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) : pn σ (fn σ) (ex Dw 0 0) 1 y ≤ 1

theorem **supBdd_one** : SupBdd Xo c fn 0 0 1

theorem **supR_one_le** (σ : c.F →+* ℂ) : supR Xo c fn σ 0 0 1 ≤ 1

theorem **pn_le_conj** (y : FrPt Xo c (CAt σ)) : pn σ (fn σ) (ex Dw a n) f y ≤ pn _ (fn _) (ex Dw a n) f (cj y)

theorem **bdd_of_conj** (hB : BddAt (K

theorem **supR_le_conj** (hB : BddAt (K

theorem **supBdd_of_fn_zero** (a : (E →+* ℂ) × ℕ → ℕ) {n : ℕ} (hn : n ≠ 0) (f : Rul Xo c) : SupBdd Xo c (fun _ _ _ => 0) a n f ∧ ∀ σ, supR Xo c (fun _ _ _ => 0) σ a n f = 0

theorem **exists_fn_small** (a : (E →+* ℂ) × ℕ → ℕ) {n : ℕ} (hn : n ≠ 0) (f : Rul Xo c) : ∃ fn : ∀ σ : c.F →+* ℂ, FrPt Xo c (CAt σ) → Ix Dw → ℝ, FnNe Xo c fn ∧ SupBdd Xo c fn a n f ∧ ∀ σ, supR Xo c fn σ a n f ≤ 1

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.ImaginaryRootConj

theorem **embOK_vl_ne** {T : CDVR} {V : VF C} {ι : T.Fr →+* C} (h : EmbOK T V ι) : vV T.V ((V.l : ℕ) : T.Fr) ≠ 0

theorem **vV_nonneg** (T : CDVR) (a : T.V) : 0 ≤ vV T.V (algebraMap T.V T.Fr a)

theorem **embOK_eOf** {T : CDVR} {V : VF C} {ι : T.Fr →+* C} (h : EmbOK T V ι) : 0 < eOf T V ∧ ((eOf T V : ℕ) : ℤ) = vV T.V ((V.l : ℕ) : T.Fr)

theorem **em_int_holds** (τ : E →+* ℂ) (x : 𝒪 E) (h : EmbOK T V ι) : V.Ge 0 (emA3 io ι τ x)

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.IsEmptyPlacesReferenceTuple

theorem **one_sub_pos** {q : ℝ} (hq : 1 < q) {i : ℕ} (hi : 1 ≤ i) : 0 < 1 - q ^ (-(i : ℤ))

theorem **prod_pos'** {q : ℝ} (hq : 1 < q) (n : ℕ) : 0 < ∏ i ∈ Finset.Icc 1 n, (1 - q ^ (-(i : ℤ)))

theorem **parIdx_pos** {q : ℝ} (hq : 1 < q) (r s : ℕ) : 0 < parIdx q r s

theorem **parIdx_zero** {q : ℝ} (hq : 1 < q) (r : ℕ) : parIdx q r 0 = 1

theorem **isEmpty_places_reference_tuple** (Z : DivX D) : IsEmpty (Z.QPl D.what₂.mu0)

theorem **myTied_setMY** (X : ThetaX D) (Z : DivX D) : ThetaDivisorCompatibility (setMY X Z) Z

theorem **mY_rep** (X : ThetaX D) (Z : DivX D) (n : ℕ) (μ : D.what₂.Tup) (j : ℕ) : (setMY X (rep Z n)).mY μ j = (setMY X Z).mY μ j ^ n

theorem **trivial_factor** (X : ThetaX D) (Z : DivX D) : ThetaDivisorCompatibility (setMY X (rep Z 0)) (rep Z 0) ∧ (rep Z 0).Qp = Z.Qp ∧ ∀ μ j, (setMY X (rep Z 0)).mY μ j = 1

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.IsTypeAtCentral

theorem **dRep_char** (m : ℤ) (θ : ℝ) (w : ℂ) : dRep (n

theorem **isTypeAt_zero** {τ : D.what₂.Tau} {S : SPar} : IsTypeAt θ V τ S 0 ↔ ∃ (H : Type) (_ : NormedAddCommGroup H) (_ : InnerProductSpace ℂ H) (σ : Matrix (Fin V.rank) (Fin V.rank) ℂ → (H →L[ℂ] H)) (lam : Fin V.rank → ℝ) (ε : Fin V.rank → Bool) (P : Matrix (Fin V.rank) (Fin V.rank) ℂ), IsEnum S lam ε ∧ IsDS ε lam σ ∧ IsUnit P ∧ Pᴴ * Formalism.REAL.Jm ε * P = gramAt D V τ

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.MatchingNotVacuous

theorem **genFibre_holds** : GenFibreSt Xo c

theorem **genFibre_tied** (T : CDVR) (O : Type) [CommRing O] (β : O →+* c.F) (Y : Model Xo c β) (yt : Sp T.V ⟶ Y.X) (z : Sp T.Fr ⟶ Xo.XF c) (hz : z ≫ Y.j = spMap (algebraMap T.V T.Fr) ≫ yt) (G : SAB H (Sp T.V)) (m : SABMap G Y.t yt) : ∃ (t' : SAB H (Sp T.Fr)) (m1 : SABMap t' Xo.univ.t (z ≫ pullback.fst _ _)) (m2 : SABMap t' G (spMap (algebraMap T.V T.Fr))) (m' : SABMap t' Y.oF.t z), m1.φ = m'.φ ≫ Y.m0.toSABMap.φ ∧ m2.φ ≫ m.φ = m'.φ ≫ Y.m.φ

theorem **matching_not_vacuous** {S : Type} [CommRing S] (x : Sp S ⟶ Xo.X) (t' : SAB H (Sp S)) (m1 : SABMap t' Xo.univ.t x) : (∃! g : t'.G.left ⟶ Gx x, g ≫ (snd (Over.mk x) Xo.univ.t.G).left = m1.φ ∧ g ≫ (fst (Over.mk x) Xo.univ.t.G).left = t'.G.hom) ∧ ∀ g : t'.G.left ⟶ Gx x, g ≫ (snd (Over.mk x) Xo.univ.t.G).left = m1.φ → ∀ j, (Xo.univ.t.PP.L.pull m1.φ).U j ≤ g ⁻¹ᵁ (PPx x).U (fun _ => j PUnit.unit)

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.NotHasInfExpThird

theorem **isVolMeasG_const** (D : D7) (W : HSp D) (P Q : ℕ) (μ : Measure (GA D W)) : IsVolMeasG D W (fun _ => P) (fun _ => Q) μ ↔ IsVolMeas D W P Q μ

theorem **volTie_of_volTieG** (D : D7) (h : VolTieG D) : VolTie D

theorem **gq_smul** (D : D7) (W : HSp D) (γ : GQ W) (g : GA D W) : γ • g = (γ : GA D W) * g

theorem **top_AE** (D : D7) : (inferInstance : TopologicalSpace (AE D)) = moduleTopology (AF D.host.F) (AE D)

theorem **not_isVolMeasG_zero** (D : D7) (W : HSp D) (P Q : D.what₂.Tau → ℕ) : ¬ IsVolMeasG D W P Q 0

theorem **congr_of_isVolMeasG** (D : D7) (W : HSp D) (P Q : D.what₂.Tau → ℕ) (μ : Measure (GA D W)) (h : IsVolMeasG D W P Q μ) (ρ : D.what₂.Tau) : ∃ A : Matrix (Fin (P ρ) ⊕ Fin (Q ρ)) (Fin W.rank) ℂ, Aᴴ * Formalism.UG.Jpq (P ρ) (Q ρ) * A = gramAt D W ρ

theorem **one_hecke** (D : D7) : IsHeckeChar D 1 ∧ UnramEverywhere D 1

theorem **not_resEta_one** (D : D7) : ¬ ResEta D 1 1

theorem **resEta_zero_iff** (D : D7) (χ : (AE D)ˣ →* Circle) : ResEta D χ 0 ↔ (∃ η, IsEta D η) ∧ ∀ a, χ (idF D a) = 1

theorem **not_hasInfExp_third** (D : D7) (χ : (AE D)ˣ →* Circle) (τ : D.what₂.Tau) : ¬ HasInfExp D χ τ (1 / 3)

theorem **isWittLower_self** (D : D7) (V : HSp D) : IsWittLower V V 0

theorem **isomEmb_inj** (D : D7) (V V' : HSp D) (ι : V'.V →ₗ[D.host.E] V.V) (h : HSpace.IsIsomEmb V' V ι) : Function.Injective ι

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.NotOverAndTriv

theorem **nonneg_int** (v : AddValuation Ω (WithTop ℤ)) (a : 𝒪 K) : 0 ≤ v (algebraMap K Ω (a : K))

theorem **zero_of_notMem** (ho : IsOver (K

theorem **zero_of_unit** (ho : IsOver (K

theorem **v_zpow** (hv : IsNormDiscrete v) {x : Ω} (hx : x ≠ 0) {m : ℤ} (hm : v x = m) (k : ℤ) : v (x ^ k) = ((k * m : ℤ) : WithTop ℤ)

theorem **exists_mul** (hv : IsNormDiscrete v) (ho : IsOver (K

theorem **restrict_eq** (hv : IsNormDiscrete v) (ho : IsOver (K

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.NusOffLevel

theorem **floor_le** (T d : ℕ) (c x : ℤ) (hc : 0 ≤ c) (hx : x ≤ c + 2) : (T : ℤ) * ((Int.fdiv x ((T + d + 2 : ℕ) : ℤ)).toNat : ℤ) ≤ c

theorem **gap_nonneg** {p : ℕ} {l : Fin p → ℕ} (h : Antitone l) (i : Fin p) : 0 ≤ WeightAlgebra.gapZ l i

theorem **of_gaps** {p : ℕ} (u a b : Fin p → ℕ) (T : ℤ) (h : ∀ i, WeightAlgebra.gapZ u i = WeightAlgebra.gapZ a i − T * WeightAlgebra.gapZ b i) : ∀ i, (u i : ℤ) = a i − T * b i

theorem **isF** (ka : WeightFamilies.Wt D j) (T nW nP : ℕ) (hW : Even nW) (hP : Even nP) : WeightAlgebra.IsF' ka (lWt D ka T nW hW) (fWt D ka (lWt D ka T nW hW) T nP hP) T nP

theorem **fdiv_toNat** (x : ℤ) (n : ℕ) (hx : 0 ≤ x) : (((Int.fdiv x n).toNat : ℕ) : ℤ) = Int.fdiv x n

theorem **isL** (ka : WeightFamilies.Wt D j) (T nW : ℕ) (hW : Even nW) (h : ∀ τ (i : Fin (D.R.sigW j).1), ¬ i.1 + 1 < (D.R.sigW j).1 → (WeightFamilies.dimY D j + 2) * nW + 2 * D.R.r ≤ ka.aP τ i) : WeightAlgebra.IsL D ka (lWt D ka T nW hW) T nW

theorem **choose_two_add** (p q : ℕ) : p * q + p * (p − 1) / 2 + q * (q − 1) / 2 = (p + q).choose 2

theorem **dim_eq** (T : ParamsT D L P) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : WeightFamilies.dimY D j = T.Dp

theorem **weights** (T : ParamsT D L P) (hK : DV.KapDef D) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : (kaR T hK j).n = D.R.nV ∧ D.R.nV = T.nP + T.TO * T.nWn ∧ WeightAlgebra.IsL D (kaR T hK j) (lR T hK j) T.TO T.nWn ∧ WeightAlgebra.IsF' (kaR T hK j) (lR T hK j) (fR T hK j) T.TO T.nP ∧ T.nWn = D.R.r + (T.nWn − D.R.r) ∧ D.R.r ≤ T.nWn − D.R.r ∧ (∀ τ (i : Fin (D.R.sigW j).1), i.1 + 1 = (D.R.sigW j).1 → 2 * T.TO ^ 2 ≤ (kaR T hK j).aP τ i) ∧ (∀ τ (i : Fin (D.R.sigW j).2), i.1 + 1 = (D.R.sigW j).2 → 2 * T.TO ^ 2 ≤ (kaR T hK j).aM τ i)

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.PageSupportRows

theorem **f_le_r** (D : D7) : D.R.f ≤ D.R.r

theorem **row_r** (D : D7) : 8 ≤ D.R.r

theorem **row_rd** (D : D7) : (D.R.f : ℝ) ^ 2 ≤ D.host.rdE

theorem **row_DF** {large₂ : SizeData₂ D} {X : PJ D B P j} {CK C1 A1 : ℝ} {ka wL wF : WeightFamilies.Wt D j} {T nP nW : ℕ} {h0 : ℝ} (h : SetC large₂ X CK C1 A1 ka wL wF T nP nW h0) : 3 ≤ DF wL

theorem **numeric_bounds_constants_pow_twelve** (f m r n Z C : ℝ) (hf : 1 ≤ f) (hm : 1 ≤ m) (hr : 1 ≤ r) (hn0 : 0 ≤ n) (hm4 : m ^ 4 ≤ f) (hZ : f * r ≤ Z) (hZ0 : 1000 ≤ Z) (hC : 1 ≤ C) (hn : n ≤ C * Z ^ 4) : 16200 * Real.pi * n * m ^ 4 * (2 * f * m) ^ 4 ≤ C * Z ^ 12 ∧ (2 * f * m) ^ 4 * n + 2 * (2 * f * m) ^ 3 ≤ C * Z ^ 12 ∧ r * (2 * f * m) ^ 3 ≤ C * Z ^ 12

theorem **page_bound_le_of_constants** {large₂ : SizeData₂ D} {K : KAdm} {H : KHyp} {Ch Ah Cq Aq C2 A2 CR : ℝ} {nW : ℕ} (hPg : PageAt large₂ C2 A2) (ht : TieC K H Ch Ah Cq Aq C2 A2 CR) (hd : DomAt D large₂ j nW H.C1) : flipOneThree D nW ≤ WeightFamilies.PPv D large₂ C2 A2

theorem **pageSupp_iff** {large₂ : SizeData₂ D} {X : PJ D B P j} {K : KAdm} {H : KHyp} {Ch Ah Cq Aq C2 A2 CR Bx CT : ℝ} {S : Scal B} {ka wL wF : WeightFamilies.Wt D j} {T nP nW : ℕ} {h0 : ℝ} {nus : HeightOneSpectrum (𝒪 B.KO) → ℝ} (hH : HypC large₂ X H C2 A2 CR Bx CT S ka wL wF T nP nW h0 nus) (ht : TieC K H Ch Ah Cq Aq C2 A2 CR) (hPg : PageAt large₂ C2 A2) : PageSupp D large₂ B C2 A2 nW S wL ↔ ∀ σ : B.KO →+* ℂ, (DF wL : ℝ) / 2 * Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ)) ≤ Real.log ‖σ (S.sW : B.KO)‖

theorem **not_pageSupp_min** {large₂ : SizeData₂ D} {C2 A2 : ℝ} {nW : ℕ} {S : Scal B} {wL : WeightFamilies.Wt D j} (hS : S.sW = ((qq D : ℕ) : 𝒪 B.KO) ^ DF wL) : (∀ σ : B.KO →+* ℂ, (DF wL : ℝ) / 2 * Real.log (qq D) ≤ Real.log ‖σ (S.sW : B.KO)‖) ∧ ¬ PageSupp D large₂ B C2 A2 nW S wL

theorem **rhoP_top** (hj : D.R.m ≤ j) (CR T c : ℝ) (ka : WeightFamilies.Wt D j) : WeightFamilies.rhoP B CR T c j ka = 0

theorem **conclAtM_iff** {large₂ : SizeData₂ D} {S : Scal B} {T : ℕ} {nus : HeightOneSpectrum (𝒪 B.KO) → ℝ} (hj : D.R.m ≤ j) (hD : DepthAtM X S wL nus) : ConclAtM large₂ X S ka wL wF T nus ↔ Concl X S ka wL wF T nus (X.rk ka (B.Mlat j ka) * D.R.f * D.R.m * large₂.sc.b * large₂.de2)

theorem **shareRho_top** (hj : D.R.m ≤ j) {large₂ : SizeData₂ D} {S : Scal B} {G : GlobalFamilyData X ka wL wF} {CRm c Ch Ah : ℝ} {T d : ℕ} (hQ : G.TiedQ large₂ S CRm T c Ch Ah) (h : ShareRhoT (D

theorem **conclAtM_of_local** (large₂ : SizeData₂ D) (X : Prov D B P) (K : KAdm) (H : KHyp) (Ch Ah Cq Aq C2 A2 CR CT : ℝ) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (S : Scal B) (ka wL wF : WeightFamilies.Wt D j) (T nP nW : ℕ) (h0 : ℝ) (nus : HeightOneSpectrum (𝒪 B.KO) → ℝ) (hH : HypC large₂ (X j h1 h2) H C2 A2 CR 0 CT S ka wL wF T nP nW h0 nus) (lA : LatAt B j ka) (sA : SupBoundByL2AtWeight large₂ B j ka K.CG K.AG) (mA : LAMwAt large₂ B j ka K.CL K.AL) (G : GlobalFamilyData (X j h1 h2) ka wL wF) (Gc PH Prw Pet Pd : ℝ) (tW : G.TiedW S Gc Prw) (tTh : G.TiedTh S Gc) (tZ : G.TiedZ S T Gc) (tQ : G.TiedQ large₂ S K.CRm T (cc D large₂ C2 A2 CR wL) Ch Ah) (dH : DeepHerm (X j h1 h2) ka) (dS : DeepSubMul (X j h1 h2)) (dL : DeepLogD (X j h1 h2) ka Pd) (pG : PageGd4 (X j h1 h2) ka large₂ K.CG K.AG Pd) (sRd : ShareRd (D

theorem **not_depthAtM_zero** {S : Scal B} (hS : S.sW = 0) {nus : HeightOneSpectrum (𝒪 B.KO) → ℝ} {la : HeightOneSpectrum (𝒪 B.KO)} (y : X.Pt la) : ¬ DepthAtM X S wL nus

theorem **combBound4_iet_zero** {G : GlobalFamilyData X ka wL wF} (hi : G.iet = 0) {Qf : ∀ la, LocalPointData G la} {S : Scal B} {c c' : ℝ} {nus : HeightOneSpectrum (𝒪 B.KO) → ℝ} (h : CombBound4 G Qf S c nus) : CombBound4 G Qf S c' nus

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.RegularityFirstClauseBot

theorem **vV_unit** (u : Vˣ) : vV V (algebraMap V F (u : V)) = 0

theorem **vV_eq_zero_of_triv** {T : CDVR} {ι : c.F →+* T.Fr} {y : IntFr Xo c T ι} (h : trivCls Xo c T ι y) {a : c.F} (ha : a ≠ 0) : vV T.V (ι a) = 0

theorem **not_overLa_of_triv** {T : CDVR} {ι : c.F →+* T.Fr} {y : IntFr Xo c T ι} (h : trivCls Xo c T ι y) (la : HeightOneSpectrum (𝒪 c.F)) (e : ℕ) : ¬ OverLa la T.V ι e

theorem **ordQ_smul** (q : QPt Xo c) {a : c.F} (ha : a ≠ 0) (f : Rul Xo c) : ordQ Xo c q (a • f) = ordQ Xo c q f

theorem **ord_nonneg_of_bddBelow** {C : ((E →+* ℂ) × ℕ → ℕ) → ℕ → Submodule c.F (Rul Xo c)} (h : RegularityR1Quotient Xo c C) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ C a n) (la : HeightOneSpectrum (𝒪 c.F)) (N : ℕ) (hb : ∀ v ∈ valSet Xo c (algCls Xo c) la, ((-(N : ℝ) : ℝ) : EReal) ≤ v f) (q : QPt Xo c) : 0 ≤ ordQ Xo c q f

theorem **ord_eq_top_of_no_point** {C : ((E →+* ℂ) × ℕ → ℕ) → ℕ → Submodule c.F (Rul Xo c)} (h : RegularityR2Quotient Xo c C) (la : HeightOneSpectrum (𝒪 c.F)) (h0 : valSet Xo c (algCls Xo c) la = ∅) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ C a n) (q : QPt Xo c) : ordQ Xo c q f = ⊤

theorem **premise_fails_of_empty** (hne : IntFrNe Xo c (algCls Xo c)) (ws : Fin 0 → Rul Xo c) (la : HeightOneSpectrum (𝒪 c.F)) : ¬ ∀ v ∈ valSet Xo c (algCls Xo c) la, ∃ i, v (ws i) ≠ ⊤

theorem **powerSeries_const_unit** (K : Type) [Field K] (a : K) (ha : a ≠ 0) : ∃ u : (PowerSeries K)ˣ, algebraMap (PowerSeries K) (FractionRing (PowerSeries K)) u = ((algebraMap (PowerSeries K) (FractionRing (PowerSeries K))).comp PowerSeries.C) a

theorem **trivCls_powerSeries** (y : IntFr Xo c (powerSeriesCDVR c.F) ((algebraMap (PowerSeries c.F) (FractionRing (PowerSeries c.F))).comp PowerSeries.C)) : trivCls Xo c _ _ y

theorem **regularity_first_clause_bot** : RegularityR1Quotient Xo c (fun _ _ => ⊥)

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.StarPurelyImaginaryEqNeg

theorem **cA_cA** (x : Formalism.L2.Theta.AE D) : cA D (cA D x) = x

theorem **cA_incA** (e : D.host.E) : cA D (incA D e) = incA D (star e)

theorem **gram_star** (V : HSp D) (i j) : star (V.gram i j) = V.gram j i

theorem **sum_small** (σ : A →+* A) (P : Matrix (Fin m) (Fin m) A) (Q : Matrix (Fin n) (Fin n) A) (x y : Matrix (Fin m) (Fin n) A) : ∑ i, ∑ j, ∑ k, ∑ l, σ (x i j) * P k i * Q j l * y k l = (x.map σ * Q * yᵀ * P).trace

theorem **sum_large** (σ : A →+* A) (P : Matrix (Fin m) (Fin m) A) (Q : Matrix (Fin n) (Fin n) A) (x y : Matrix (Fin m) (Fin n) A) : ∑ i, ∑ j, ∑ k, ∑ l, σ (x i j) * P i k * Q l j * y k l = ((x.map σ)ᵀ * P * y * Q).trace

theorem **inv_small** (σ : A →+* A) (hσ : ∀ a, σ (σ a) = a) (P g : Matrix (Fin m) (Fin m) A) (Q h : Matrix (Fin n) (Fin n) A) (hg : (g.map σ)ᵀ * P * g = P) (hh : (h.map σ)ᵀ * Q * h = Q) (x y : Matrix (Fin m) (Fin n) A) : ((g.map σ * x * hᵀ).map σ * Q * (g.map σ * y * hᵀ)ᵀ * P).trace = (x.map σ * Q * yᵀ * P).trace

theorem **inv_large** (σ : A →+* A) (hσ : ∀ a, σ (σ a) = a) (P g : Matrix (Fin m) (Fin m) A) (Q h : Matrix (Fin n) (Fin n) A) (hg : (g.map σ)ᵀ * P * g = P) (hh : (h.map σ)ᵀ * Q * h = Q) (x y : Matrix (Fin m) (Fin n) A) : (((g * x * (h.map σ)ᵀ).map σ)ᵀ * P * (g * y * (h.map σ)ᵀ) * Q).trace = ((x.map σ)ᵀ * P * y * Q).trace

theorem **herm_small** (σ : A →+* A) (hσ : ∀ a, σ (σ a) = a) (P : Fin m → Fin m → A) (Q : Fin n → Fin n → A) (hP : ∀ i k, σ (P i k) = P k i) (hQ : ∀ i k, σ (Q i k) = Q k i) (x y : Matrix (Fin m) (Fin n) A) : σ (∑ i, ∑ j, ∑ k, ∑ l, σ (x i j) * P k i * Q j l * y k l) = ∑ i, ∑ j, ∑ k, ∑ l, σ (y i j) * P k i * Q j l * x k l

theorem **mt_unit** (V : HSp D) (g : GA D V) : ((mt V g).map (cA D))ᵀ * V.gram.map (incA D) * mt V g = V.gram.map (incA D)

theorem **hHom_herm** (V W : HSp D) (x y : WA V W) : cA D (hHom V W x y) = hHom V W y x

theorem **split_ne** (e : Formalism.L2.Theta.AE D) (he : IsSplitIdem D e) : cA D e ≠ e

theorem **trace_cA** (x : Formalism.L2.Theta.AE D) : Algebra.trace (AF D.host.F) (Formalism.L2.Theta.AE D) (cA D x) = Algebra.trace (AF D.host.F) (Formalism.L2.Theta.AE D) x

theorem **star_purely_imaginary_eq_neg** (D : D7) : star D.host.xi0 = -D.host.xi0

theorem **sympl_skew** (V W : HSp D) (x y : WA V W) : sympl V W y x = -sympl V W x y

theorem **symplH_add_left** (V W : HSp D) (x x' y : WA V W) : symplH V W (x + x') y = symplH V W x y + symplH V W x' y

theorem **symplH_add_right** (V W : HSp D) (x y y' : WA V W) : symplH V W x (y + y') = symplH V W x y + symplH V W x y'

theorem **heis_heis** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (w w' : WA V W) (f : WA V W → ℂ) : Heisenberg₂ ψ V W w (Heisenberg₂ ψ V W w' f) = ψ (symplH V W w w') • Heisenberg₂ ψ V W (w + w') f

theorem **heis_quasi** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (w₀ : WA V W) (f : WA V W → ℂ) (hf : ∀ γ w, f (ratW V W γ + w) = ψ (-(symplH V W (ratW V W γ) w)) * f w) (γ w) : Heisenberg₂ ψ V W w₀ f (ratW V W γ + w) = ψ (-(symplH V W (ratW V W γ) w)) * Heisenberg₂ ψ V W w₀ f w

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.SymplHAddRight

theorem **hHom_add_left** (V W : HSp D) (x x' y : WA V W) : hHom V W (x + x') y = hHom V W x y + hHom V W x' y

theorem **hHom_add_right** (V W : HSp D) (x y y' : WA V W) : hHom V W x (y + y') = hHom V W x y + hHom V W x y'

theorem **symplH_add_left** (V W : HSp D) (x x' y : WA V W) : symplH V W (x + x') y = symplH V W x y + symplH V W x' y

theorem **symplH_add_right** (V W : HSp D) (x y y' : WA V W) : symplH V W x (y + y') = symplH V W x y + symplH V W x y'

theorem **heis_heis** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (u u' : WA V W) (f : WA V W → ℂ) (w : WA V W) : Heisenberg₂ ψ V W u (Heisenberg₂ ψ V W u' f) w = ψ (symplH V W u u') * Heisenberg₂ ψ V W (u + u') f w

theorem **eq_zero_of_fixed** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (L : AddSubgroup (WA V W)) {u u' : WA V W} (hu : u ∈ L) (hu' : u' ∈ L) (hne : ψ (symplH V W u u') ≠ 1) (f : WA V W → ℂ) (hf : ∀ x ∈ L, Heisenberg₂ ψ V W x f = f) : f = 0

theorem **unr_void** {X : ThetaX D} {T : ThetaTied D X} (Wl : WeilTied D X T) (V W : HSp D) (S : Finset (HeightOneSpectrum (𝒪 D.host.F))) {u u' : WA V W} (hu : u ∈ latVWS V W S) (hu' : u' ∈ latVWS V W S) (hne : Wl.ψ (symplH V W u u') ≠ 1) (Φ : X.Sch V W) (hΦ : ∀ x ∈ latVWS V W S, Heisenberg₂ Wl.ψ V W x (Wl.evS V W Φ) = Wl.evS V W Φ) : Φ = 0 ∧ ∀ χ χ' (p : GA D V × GA D W), Wl.om V W χ χ' p Φ = Φ

Development.AbelianVarietyTheory.Counterexamples.HeightSurvey.WindowsAllowOffCentre

theorem **last_plus** (hK : DV.KapDef D) (hn : Even D.R.nV) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (τ : D.what₂.Tau) (i : Fin (D.R.sigW j).1) (hi : i.1 + 1 = (D.R.sigW j).1) : (((Cons.kaWt D hK hn j).aP τ i : ℕ) : ℝ) = ((((D.hc τ).e ⟨j - 1, by omega⟩ - (D.hc τ).b ⟨j - 1, by omega⟩ : ℚ)) : ℝ) - ((D.R.t : ℝ) + 1) / 2

theorem **last_minus** (hK : DV.KapDef D) (hn : Even D.R.nV) (h1 : 1 ≤ j) (h2 : j < D.R.m) (τ : D.what₂.Tau) (i : Fin (D.R.sigW j).2) (hi : i.1 + 1 = (D.R.sigW j).2) : (((Cons.kaWt D hK hn j).aM τ i : ℕ) : ℝ) = ((((D.hc τ).b ⟨j, h2⟩ - (D.hc τ).e ⟨j - 1, by omega⟩ : ℚ)) : ℝ) - ((D.R.t : ℝ) + 1) / 2

theorem **clauses_at_shifted_gap** (T : ParamsT D L P) (hK : DV.KapDef D) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : 4 * (T.TO : ℝ) ^ 2 + D.R.t + 1 ≤ LevelStarExponents.h0 L - 4 ∧ (∀ τ (i : Fin (D.R.sigW j).1), i.1 + 1 = (D.R.sigW j).1 → (LevelStarExponents.h0 L - 4) / 2 - ((D.R.t : ℝ) + 1) / 2 ≤ ((Cons.kaR T hK j).aP τ i : ℝ)) ∧ (∀ τ (i : Fin (D.R.sigW j).2), i.1 + 1 = (D.R.sigW j).2 → (LevelStarExponents.h0 L - 4) / 2 - ((D.R.t : ℝ) + 1) / 2 ≤ ((Cons.kaR T hK j).aM τ i : ℝ))

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.BalancedPairs

theorem **emb_bar** (H : Host E) (τ : E →+* ℂ) (a : 𝒪 E) : τ ((H.bar a : 𝒪 E) : E) = ComplexEmbedding.conjugate τ (a : E)

theorem **balanced_pairs** (H : Host E) {l : Level H} (c : Coeff l) {S : Type} [CommRing S] [Algebra c.F S] (τ τ' : E →+* ℂ) (v : S) (hv : ∀ a : 𝒪 E, algebraMap c.F S (embK c τ a) * v = algebraMap c.F S (embK c τ' (H.bar a)) * v) (hne : τ ≠ ComplexEmbedding.conjugate τ') : v = 0

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.BilLActRight

theorem **cc_log** (G : FGLaw S d) (i : Fin d) : constantCoeff (logOf G i) = 0

theorem **additive_eq** (G : FGLaw S d) (g : PS S 1 d) (hg : ev (n

theorem **cc_pderiv** (f : PS S 1 d) (j : Fin d) : constantCoeff (pderiv S ⟨0 : Fin 1), j⟩ f) = coeff (Finsupp.single ⟨0 : Fin 1), j⟩ 1) f

theorem **log_act** {O : Type} [CommRing O] (G : FGLaw S d) (A : OAct O G) (a : O) (i : Fin d) : ev (n

theorem **ev_bilL** {n : ℕ} (G : FGLaw S d) (cc : Fin d → Fin d → S) {a : Fin 2 → Pt S n d} (ha : CZ a) : ev a (bilL G cc) = ∑ i, ∑ l, C (cc i l) * (ev ![a 0] (logOf G i) * ev ![a 1] (logOf G l))

theorem **log_act_at** {O : Type} [CommRing O] (G : FGLaw S d) (A : OAct O G) (a : O) (l : Fin 2) (i : Fin d) : ev (n

theorem **bilL_act_left** {O : Type} [CommRing O] (G : FGLaw S d) (A : OAct O G) (a : O) (cc : Fin d → Fin d → S) : ev (n

theorem **bilL_act_right** {O : Type} [CommRing O] (G : FGLaw S d) (A : OAct O G) (b : O) (cc : Fin d → Fin d → S) : ev (n

theorem **bilL_mixed** (G : FGLaw S d) (cc : Fin d → Fin d → S) (i l : Fin d) : constantCoeff (pderiv S ⟨0 : Fin 2), i⟩ (pderiv S ⟨1 : Fin 2), l⟩ (bilL G cc))) = cc i l

theorem **bilL_injective** (G : FGLaw S d) : Function.Injective (bilL G)

theorem **balanced_iff** {O : Type} [CommRing O] (G : FGLaw S d) (A : OAct O G) (a b : O) (cc : Fin d → Fin d → S) : (dLin (A.act a)).transpose * Matrix.of cc = Matrix.of cc * dLin (A.act b) ↔ ev (n

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ChartIndepTruncOfExpGauge

theorem **psiCover** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : PsiCoverSt H

theorem **exists_sq_of_degree_two** (m : Fin 1 × Fin d →₀ ℕ) (hm : m.degree = 2) : ∃ i j, i ≤ j ∧ m = sq2 i j

theorem **sq_monomial_injective** {i j a b : Fin d} (hij : i ≤ j) (hab : a ≤ b) (h : sq2 i j = sq2 a b) : i = a ∧ j = b

theorem **eq_sum_quadMat** (q : MvPowerSeries (Fin 1 × Fin d) S) (hq : ∀ m, coeff m q ≠ 0 → m.degree = 2) : q = ∑ p : Fin d × Fin d, C (quadMat q p.1 p.2) * X ⟨0, p.1⟩ * X ⟨0, p.2⟩

theorem **killAddVal** : KillAddValSt

theorem **expUniqTan** : ExpUniqTanSt

theorem **frTruncOfGerms** : FrTruncOfGermsSt

theorem **chartIndepTrunc_of_expGauge** (hG : ExpGaugeOfSt) : ChartIndepTruncSt

theorem **four_claims** : ExpUniqTanSt ∧ KillAddValSt ∧ FrTruncOfGermsSt ∧ (ExpGaugeOfSt → ChartIndepTruncSt)

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ConclusionNotEmpty

theorem **both_hold** : BoxRosatiSt ∧ BoxTauBalSt

theorem **conclusion_not_empty** {D' D : DegPre R} {n : ℕ} (pD : IsPowDeg D' D n) (f g : D.P.A ⟶ D.P.A) (f' g' : D'.P.A ⟶ D'.P.A) (hf : IsMonHom f) (hg : IsMonHom g) (ef : ∀ i, f' ≫ pD.pw.pr i = pD.pw.pr i ≫ f) (eg : ∀ i, g' ≫ pD.pw.pr i = pD.pw.pr i ≫ g) (hr : RosatiAdj D.P.A D.P.L f g) : ∃ u : LB.Gen (LB.LinComb D'.P.L (rosatiφ D'.P.A f' g') rosatiε), u.IsGen ∧ LB.Gen.Corr D'.P.L _ _ _ _ (LB.pullGen D'.P.L _ _ u (zero2 D'.P.A)) (LB.oneGen D'.P.L)

theorem **c_compatible** {D' D : DegPre R} {n : ℕ} (pD : IsPowDeg D' D n) (f : D.P.A ⟶ D.P.A) (f' : D'.P.A ⟶ D'.P.A) (a : D.X → D.X) (ef : ∀ i, f' ≫ pD.pw.pr i = pD.pw.pr i ≫ f) (ca : ∀ y, D.c (a y) = D.c y ≫ f) (x : D'.X) : D'.c (pD.xe.symm fun i => a (pD.xe x i)) = D'.c x ≫ f'

theorem **tauBal_id_iff** (D : DegPre R) (hr : RosatiAdj D.P.A D.P.L (𝟙 _) (𝟙 _)) (a a' : D.X → D.X) : D.TauBal (𝟙 _) (𝟙 _) a a' ↔ ∀ y y' z J, Good D.P.L z J → val (D.tau (a y) y') (𝟙 _) z J = val (D.tau y (a' y')) (𝟙 _) z J

theorem **pi_zpow_injective** {a b : ℤ} (h : pi0 ^ a = pi0 ^ b) : a = b

theorem **rosati_tate** : RosatiAdj tatePre.P.A tatePre.P.L (𝟙 _) (𝟙 _)

theorem **good_tate** (z : (TK InhTate.R).left) (J) : Good tatePre.P.L z J

theorem **val_tau_tate** (a b : ℤ) (z : (TK InhTate.R).left) (J) : val (tatePre.tau a b) (𝟙 _) z J = gm (𝟙 _) z ⊤ (pi0 ^ (2 * a * b))

theorem **gm_pi_zpow_injective** (z : (TK InhTate.R).left) {a b : ℤ} (h : gm (𝟙 _) z ⊤ (pi0 ^ a) = gm (𝟙 _) z ⊤ (pi0 ^ b)) : a = b

theorem **nonempty_TK** : Nonempty (TK InhTate.R).left

theorem **tauBal_tate_iff** (a a' : ℤ → ℤ) : tatePre.TauBal (𝟙 _) (𝟙 _) a a' ↔ ∀ y y', a y * y' = y * a' y'

theorem **premises_met** (m : ℤ) (n : ℕ) : ∃ (D' : DegPre InhTate.R) (pD : IsPowDeg D' tatePre n), (∀ i, 𝟙 _ ≫ pD.pw.pr i = pD.pw.pr i ≫ 𝟙 _) ∧ RosatiAdj tatePre.P.A tatePre.P.L (𝟙 _) (𝟙 _) ∧ (∀ y : ℤ, tatePre.c (m * y) = tatePre.c y ≫ 𝟙 _) ∧ tatePre.TauBal (𝟙 _) (𝟙 _) (fun y : ℤ => m * y) (fun y : ℤ => m * y)

theorem **sum_of_tauBal_pow** {D' : DegPre InhTate.R} {n : ℕ} (pD : IsPowDeg D' tatePre n) (a a' : ℤ → ℤ) (h : D'.TauBal (𝟙 _) (𝟙 _) (fun y => pD.xe.symm fun i => a (pD.xe y i)) (fun y => pD.xe.symm fun i => a' (pD.xe y i))) (y y' : Fin n → ℤ) : ∑ i, a (y i) * y' i = ∑ i, y i * a' (y' i)

theorem **conclusion_can_fail** {D' : DegPre InhTate.R} {n : ℕ} (pD : IsPowDeg D' tatePre (n + 1)) : ¬ D'.TauBal (𝟙 _) (𝟙 _) (fun y => pD.xe.symm fun i => (fun _ : ℤ => (0 : ℤ)) (pD.xe y i)) (fun y => pD.xe.symm fun i => (fun t : ℤ => t) (pD.xe y i))

theorem **rosati_iff** {R : Type} [CommRing R] [IsDomain R] {D' D : DegPre R} {n : ℕ} (pD : IsPowDeg D' D (n + 1)) (f g : D.P.A ⟶ D.P.A) (f' g' : D'.P.A ⟶ D'.P.A) (hf : IsMonHom f) (hg : IsMonHom g) (ef : ∀ i, f' ≫ pD.pw.pr i = pD.pw.pr i ≫ f) (eg : ∀ i, g' ≫ pD.pw.pr i = pD.pw.pr i ≫ g) : RosatiAdj D'.P.A D'.P.L f' g' ↔ RosatiAdj D.P.A D.P.L f g

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.CubicalPullBaseChange

theorem **rigPullIsGen** : RigPullIsGenSt.{0}

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.EqZeroOfPnorm

theorem **pnorm_nonneg** (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) (x : Frame X.Ψ r) : 0 ≤ X.pnorm la P x

theorem **pnorm_le_supN** (U : Unf Xo c σ r) (la) {P : FDSec U.X} (hP : P ∈ holSub U.X la) (p : FPt U.X) : (U.X p.1).pnorm (la p.1) (P p.1) p.2 ≤ supN U.X la P

theorem **eq_zero_of_pnorm** (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (h0 : X.pnorm la P x = 0) : P x = 0

theorem **good_conj_iff** (c : Coeff l) (σ : c.F →+* ℂ) : GoodEmb c (ComplexEmbedding.conjugate σ) ↔ ¬ GoodEmb c σ

theorem **goodOf_conj** (c : Coeff l) (σ : c.F →+* ℂ) : goodOf c (ComplexEmbedding.conjugate σ) = goodOf c σ

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsIsCayleyUIff

lemma **trace_Jm** {n : ℕ} (ε : Fin n → Bool) : (Jm ε).trace = 2 * ((univ.filter fun i => ε i = true).card : ℂ) − n

lemma **exists_perm** {n : ℕ} (ε δ : Fin n → Bool) (h : (univ.filter fun i => ε i = true).card = (univ.filter fun i => δ i = true).card) : ∃ e : Fin n ≃ Fin n, ∀ a, δ (e a) = ε a

theorem **exists_unitary** {n : ℕ} (A : Matrix (Fin n) (Fin n) ℂ) (hA : A.IsHermitian) (h2 : A * A = 1) (ε : Fin n → Bool) (ht : (Jm ε).trace = A.trace) : ∃ P ∈ Matrix.unitaryGroup (Fin n) ℂ, Pᴴ * Jm ε * P = A

lemma **trace_Hpk** (k r : ℕ) : (Hpk k r).trace = (k : ℂ) − 1

theorem **exists_isCayleyU_iff** (k r : ℕ) (ε : Fin (2 * r + k + 1) → Bool) : (∃ P, IsCayleyU k r ε P) ↔ (univ.filter fun i => ε i = true).card = k + r

theorem **transport** {k r : ℕ} {ε : Fin (2 * r + k + 1) → Bool} {P : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ} (hP : IsCayleyU k r ε P) (g : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ) : (P⁻¹ * g * P ∈ UH (Hpk k r) ↔ g ∈ UJset ε) ∧ (P⁻¹ * g * P ∈ Matrix.unitaryGroup (Fin (2 * r + k + 1)) ℂ ↔ g ∈ Matrix.unitaryGroup (Fin (2 * r + k + 1)) ℂ)

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsLatticePart

theorem **exists_move** (φ : P.toPre.P.A.left ⟶ a.D₀.P.A.left) (he : eSec P.toPre.P.A ≫ φ = spMap (PowerSeries.C (R

theorem **levelAgree_transition_congruent_mod_level** (φ₁ φ₂ : P.toPre.P.A.left ⟶ a.D₀.P.A.left) (he₁ : eSec P.toPre.P.A ≫ φ₁ = spMap (PowerSeries.C (R

theorem **exists_lattice_part** {R : Type} [CommRing R] [IsDomain R] (D : DegPre R) (hB : D.TauBimult) (F : Type) [Field F] [IsAlgClosed F] [Algebra (FractionRing R) F] (n : ℕ) (hn : 0 < n) (y : D.X) : ∃ p : QkPt D F n, p.y = y

theorem **exists_extension** {R : Type} [CommRing R] [IsDomain R] (D : DegPre R) (F F' : Type) [Field F] [Algebra (FractionRing R) F] [Field F'] [Algebra (FractionRing R) F'] (f : F →+* F') (hf : ∀ r : FractionRing R, f (algebraMap _ F r) = algebraMap _ F' r) (n : ℕ) (p : QkPt D F n) : ∃ p' : QkPt D F' n, QkExt f p p'

theorem **mem_closure_level** (F : Type) [Field F] [IsAlgClosed F] [Algebra (FractionRing k) F] (x : TF (R

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsPolPremises

theorem **both_hold** : QkPullExistsSt ∧ PolOActPullSt

theorem **injective_of_compat** (hF : ArcPull.Compat φ F) : Function.Injective φ

theorem **existsUnique_pair** {D : DegPre R} {D' : DegPre R'} (hD : IsPullDeg φ D D') (hF : ArcPull.Compat φ F) (k : ℕ) (p' : QkPt D' F k) : ∃! p : QkPt D F k, p'.y = hD.xe.symm p.y ∧ IsPullGtF φ hD F p.g p'.g

theorem **exists_premises** (hF : ArcPull.Compat φ F) (D : DegPre R) (hB : D.TauBimult) (k : ℕ) (p : QkPt D F k) : ∃ (D' : DegPre R') (hD : IsPullDeg φ D D') (p' : QkPt D' F k), p'.y = hD.xe.symm p.y ∧ IsPullGtF φ hD F p.g p'.g

theorem **halfPeriod_instance** : ∃ (D' : DegPre InhTate.R) (hD : IsPullDeg (RingHom.id _) multData.toPre D') (p' : QkPt D' KK 2), p'.y ≠ 0 ∧ ∀ p : QkPt multData.toPre KK 2, (p'.y = hD.xe.symm p.y ∧ IsPullGtF (RingHom.id _) hD KK p.g p'.g) ↔ p = halfPeriod

theorem **act_unique** {S S' : Scheme.{0}} {f : S' ⟶ S} {P : PolAV S} {P' : PolAV S'} {g : P'.A.left ⟶ P.A.left} (hg : IsGrpPullback P'.A P.A f g) {O : Type} [CommRing O] [StarRing O] (o : P.OAct O) (o₁ o₂ : P'.OAct O) (h₁ : ∀ a, (o₁.act a).left ≫ g = g ≫ (o.act a).left) (h₂ : ∀ a, (o₂.act a).left ≫ g = g ≫ (o.act a).left) : o₁ = o₂

theorem **exists_pol_premises** {S S' : Scheme.{0}} (f : S' ⟶ S) (P : PolAV S) : ∃ (P' : PolAV S') (g : P'.A.left ⟶ P.A.left), P'.g = P.g ∧ IsGrpPullback P'.A P.A f g ∧ LBIso (P.L.pull g) P'.L

theorem **elliptic_instance** {R : Type} [CommRing R] [IsNoetherianRing R] (W : _root_.WeierstrassCurve R) [W.IsElliptic] {S' : Scheme.{0}} (f : S' ⟶ Spec (CommRingCat.of R)) : ∃ (o : (WeierstrassCurve.ellPolAV W).OAct ℤ) (P' : PolAV S') (g : P'.A.left ⟶ (WeierstrassCurve.ellPolAV W).A.left), (∀ a : ℤ, o.act a = (𝟙 (WeierstrassCurve.ellPolAV W).A : _ ⟶ _) ^ a) ∧ P'.g = 1 ∧ IsGrpPullback P'.A (WeierstrassCurve.ellPolAV W).A f g ∧ LBIso ((WeierstrassCurve.ellPolAV W).L.pull g) P'.L ∧ ∃! o' : P'.OAct ℤ, ∀ a, (o'.act a).left ≫ g = g ≫ (o.act a).left

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsPowWithAction

theorem **exists_pow** (R : Type) [CommRing R] [IsDomain R] (I : Ideal R) (D : DegData R I) (n : ℕ) : ∃ D' : DegData R I, Nonempty (IsPowDeg D'.toPre D.toPre n)

theorem **isMonHom_of_coord** (pD : IsPowDeg D' D n) (f : D.P.A ⟶ D.P.A) [IsMonHom f] (f' : D'.P.A ⟶ D'.P.A) (h : ∀ i, f' ≫ pD.pw.pr i = pD.pw.pr i ≫ f) : IsMonHom f'

theorem **powOActSt_iff** : PowOActSt ↔ ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (t : SAB H (Sp V)) (P : PELUnif (maximalIdeal V) t) (n : ℕ) (D' : DegData V (maximalIdeal V)) (pD : IsPowDeg D'.toPre P.D.toPre n), ∃ (M : Module (𝒪 E) D'.toPre.X) (oD' : @DegPre.OAct V _ _ (𝒪 E) _ H.starRing D'.toPre M), (∀ (a : 𝒪 E) (x : D'.toPre.X) (i : Fin n), pD.xe (M.toDistribMulAction.toMulAction.toSMul.smul a x) i = P.mod.toDistribMulAction.toMulAction.toSMul.smul a (pD.xe x i)) ∧ ∀ (a : 𝒪 E) (i : Fin n), (letI

theorem **premises_met** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Δ : Datum H) (h : 1 ≤ Δ.p + Δ.q) (n : ℕ) : ∃ (t : SAB H (Sp (PowerSeries ℂ))) (P : PELUnif (maximalIdeal (PowerSeries ℂ)) t), 0 < P.A.g ∧ ∃ D' : DegData (PowerSeries ℂ) (maximalIdeal _), Nonempty (IsPowDeg D'.toPre P.D.toPre n)

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsPullbackPrescribed

theorem **lbIso_cube** {S : Scheme.{0}} {G T : Over S} [GrpObj G] [IsCommMonObj G] {n : ℕ} (x : Fin n → (T ⟶ G)) {L M : LB G.left} (h : LBIso L M) : LBIso (cubeLB G x L) (cubeLB G x M)

theorem **cube_isLB** {S : Scheme.{0}} {G T : Over S} [GrpObj G] [IsCommMonObj G] {n : ℕ} (x : Fin n → (T ⟶ G)) {L : LB G.left} (h : L.IsLB) : (cubeLB G x L).IsLB

theorem **exists_pullback_prescribed** {S S' : Scheme.{0}} (f : S' ⟶ S) (t : SAB H S) (PP' : RigLB ((Over.pullback f).obj t.G)) (hw : ∃ w : LB.Iso PP'.L (t.PP.L.pull (pullback.fst t.G.hom f)), w.IsGen) : ∃ (cub : CubicalStr _ PP') (sym : PP'.IsSymmetric) (adj : _) (prim : _) (m : SABMap ⟨(Over.pullback f).obj t.G, IsSemiAbelian.pullback f t.G t.semi, OAct.pullback f t.G t.act, PP', cub, sym, adj, prim⟩ t f), m.φ = pullback.fst t.G.hom f

theorem **exists_prescribed** {S S' : Scheme.{0}} (f : S' ⟶ S) (t : SAB H S) : ∃ PP' : RigLB ((Over.pullback f).obj t.G), ∃ w : LB.Iso PP'.L (t.PP.L.pull (pullback.fst t.G.hom f)), w.IsGen

theorem **isGrpPullback_inv** {S : Scheme.{0}} (G : Over S) [GrpObj G] [IsCommMonObj G] : IsGrpPullback G G (𝟙 S) (ι[G]).left

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsSecNeZero

theorem **span_holds** : TwistBCSpanStatement

theorem **span_any** (hk : 1 ≤ k) (α : D.X) : ∃ s : Fin (k ^ D.P.g) → D.SecB k α, ∀ (R' : Type) [CommRing R'] (f : R →+* R') (s' : LB.Sec (LB.LinComb D.P.L (fun i => (snd (Over.mk (Spec.map (CommRingCat.ofHom f))) D.P.A).left ≫ DegPre.famφB D α i) (DegPre.famε k))), ∃! r : Fin (k ^ D.P.g) → Γ(Spec (CommRingCat.of R'), ⊤), ∀ j, s'.f j = ∑ i, ((Over.mk (Spec.map (CommRingCat.ofHom f)) ⊗ D.P.A).left.presheaf.map (homOfLE le_top).op).hom ((fst (Over.mk (Spec.map (CommRingCat.ofHom f))) D.P.A).left.appTop.hom (r i)) * (pullSec D.P.L (DegPre.famφB D α) (DegPre.famε k) (s i) (snd (Over.mk (Spec.map (CommRingCat.ofHom f))) D.P.A).left).f j

theorem **exists_sec_ne_zero** (hk : 1 ≤ k) (α : D.X) (R' : Type) [CommRing R'] [Nontrivial R'] (f : R →+* R') : ∃ s' : LB.Sec (LB.LinComb D.P.L (fun i => (snd (Over.mk (Spec.map (CommRingCat.ofHom f))) D.P.A).left ≫ DegPre.famφB D α i) (DegPre.famε k)), ∃ j, s'.f j ≠ 0

theorem **exists_degPre_ell** (W : _root_.WeierstrassCurve R) [W.IsElliptic] (c : 𝟙_ (Over (Spec (CommRingCat.of R))) ⟶ (GlobalTheory.WeierstrassCurve.ellPolAV W).A) : ∃ (D : DegPre R) (α : D.X) (e : D.P = GlobalTheory.WeierstrassCurve.ellPolAV W), D.P.g = 1 ∧ e ▸ D.c α = c

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsUniquePairIff

theorem **isZero_toT** (g : GtPtF D F) : (toT g).IsZero ↔ g.IsZero

theorem **isPullF_iff_isPull** (x : Spec (CommRingCat.of F) ⟶ T.left) (g : GtPt D T) (g' : GtPtF D F) : GtPt.IsPullF F x g g' ↔ GtPt.IsPull (T'

theorem **over_of_isPullF** {x : Spec (CommRingCat.of F) ⟶ T.left} {g : GtPt D T} {g' : GtPtF D F} (H : GtPt.IsPullF F x g g') : x ≫ T.hom = jFK (R

theorem **maps_eq** {x : Spec (CommRingCat.of F) ⟶ T.left} {g : GtPt D T} {g' : GtPtF D F} (H : GtPt.IsPullF F x g g') (χ : D.X) (k : Fin 4) : DegPre.gtφ D F g'.b χ k = x ≫ DegPre.gtφT D T g.b χ k

theorem **isPullF_iff** (σ : TF (R

theorem **ext_valF** {g g' : GtPtF D F} (hb : g.b = g'.b) (hs : ∀ χ y J, Good D.P.L y J → val (g.s χ) (𝟙 _) y J = val (g'.s χ) (𝟙 _) y J) : g = g'

theorem **isPullF_unique** {x : Spec (CommRingCat.of F) ⟶ T.left} {g : GtPt D T} {a a' : GtPtF D F} (H : GtPt.IsPullF F x g a) (H' : GtPt.IsPullF F x g a') : a = a'

theorem **exists_isPullF** (σ : TF (R

theorem **formEq_zero** (χ : D.X) : FormEq (K'

theorem **isZero_restrict** {σ : TF (R

theorem **isAdd_restrict** {σ : TF (R

theorem **isNsmul_restrict** {σ : TF (R

theorem **isIota_restrict** {σ : TF (R

theorem **existsUnique_pair_iff** {k : ℕ} (x : Spec (CommRingCat.of F) ⟶ S.left) (p : QkPtW D W ψW k S) : (∃! q : QkPt D F k, QkPtW.IsPullF F x p q) ↔ x ≫ S.hom ≫ W.ι = jFK (R

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ExistsUniquePullExpands

theorem **exists_pull_pow** {V : Type} [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] {D : DegData V (maximalIdeal V)} {A : PolAV (Spec (CommRingCat.of (FractionRing V)))} {G : Over (Spec (CommRingCat.of V))} [GrpObj G] [IsCommMonObj G] (U : Unif (maximalIdeal V) D A G) (f : G ⟶ G) (k N : ℕ) (s : U.SecK k) (s' : U.SecK (k * N)) (h : IsPullByEnd U f k N s s') : ∃ c : ∀ y : (pullback G.hom (jK V) : Scheme.{0}), ((pullback G.hom (jK V) ⟶ G.left) → U.LG.L.ι) → (pullback G.hom (jK V) : Scheme.{0}).presheaf.stalk y, (∀ y J, Good U.LG.L y J → sval s' (𝟙 _) y J = sval s (endK f) y J * c y J) ∧ ∀ (m : ℕ) (t : U.SecK (m * k)), ∃ t' : U.SecK (m * k * N), IsPullByEnd U f (m * k) N t t' ∧ ∀ y J, Good U.LG.L y J → sval t' (𝟙 _) y J = sval t (endK f) y J * c y J ^ m

theorem **expands_pull_every_level** (hLaws : Laws) (P : Geo.Tor.PELUnif (maximalIdeal V) t) (a : NumberField.RingOfIntegers E) (N : ℕ) (hN : letI

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.FormAtOfClause

theorem **phiC_nonneg** (X : WeightAlgebra.PJ D B P j) (la : HeightOneSpectrum (𝒪 B.KO)) (y : X.Pt la) : 0 ≤ X.phiC la y

theorem **zero** (X : WeightAlgebra.PJ D B P j) (i : ℕ) : EtVal X i 0

theorem **mem_Mlat** (X : WeightAlgebra.PJ D B P j) (i : ℕ) (η : B.Cusp j (Formalism.L2.Pj.wEt D j i)) (h : EtVal X i η) : η ∈ B.Mlat j _

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.FrTruncFixIff

theorem **pairing_eq** {S : Type} [CommRing S] {d : ℕ} (M : Fin d → Fin d → S) (v w : Fin d → S) : ∑ i, ∑ j, M i j * v i * w j = v ⬝ᵥ (Matrix.of M *ᵥ w)

theorem **balanced_iff** (c : Coeff l) {S : Type} [CommRing S] [Algebra c.F S] {d : ℕ} (A : 𝒪 E → Matrix (Fin d) (Fin d) S) (u : Lt1 Dw → Fin d → S) (heig : ∀ (a : 𝒪 E) (ℓ : Lt1 Dw), A a *ᵥ u ℓ = algebraMap c.F S (embK c ℓ.1 a) • u ℓ) (hsp : ∀ v : Fin d → S, ∃ a : Lt1 Dw → S, ∀ j, ∑ ℓ, a ℓ * u ℓ j = v j) (M : Fin d → Fin d → S) : (∀ a, (A a)ᵀ * Matrix.of M = Matrix.of M * A (H.bar a)) ↔ ∀ ℓ ℓ' : Lt1 Dw, ∑ i, ∑ j, M i j * u ℓ i * u ℓ' j ≠ 0 → ℓ'.1 = ComplexEmbedding.conjugate ℓ.1

theorem **frTruncFix_iff** : FrTruncFixSt ↔ FrTruncFixBSt

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.FramePointsTrunc

theorem **linEig** : LinEigSt

theorem **expChartOf** : ExpChartOfSt

theorem **commonCoc** : CommonCocSt

theorem **three_hold** : ExpChartOfSt ∧ LinEigSt ∧ CommonCocSt

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.HtCharOfIsUnit

theorem **unitOf_one'** {R : Type} [CommRing R] [Algebra O R] : unitOf (1 : specO O R ⟶ GmO O) = 1

theorem **lie_iff** (w : specO O (DualNumber O) ⟶ GmO O) : ptMap (TrivSqZeroExt.fstHom O O O) ≫ w = toUnit (specO O O) ≫ η[GmO O] ↔ (unitOf w : DualNumber O).fst = 1

theorem **fst_eq** (x : LieG (GmO O)) : (unitOf x.1 : DualNumber O).fst = 1

theorem **isAl_taut** (p : ℕ) : IsAl (taut (O

theorem **dlogT_mem** : dlogT ∈ omega (GmO O)

theorem **dlogT_bijective** : Function.Bijective (dlogT (O

theorem **omega_le_span** : omega (GmO O) ≤ Submodule.span O {dlogT}

theorem **htSpan_gm** (p : ℕ) : htSpan (GmO O) p = omega (GmO O)

theorem **htChar_gm** (p : ℕ) {C : Type} [Field C] (v : Valuation C NNReal) [Algebra O C] [Nontrivial O] : HTChar p v (GmO O) ∧ omega (GmO O) ≠ ⊥

theorem **not_isAl_zero** (p : ℕ) (hp : ¬IsUnit (p : O)) : ¬IsAl (taut (O

theorem **htChar_of_isUnit** (p : ℕ) (hp : IsUnit (p : O)) {C : Type} [Field C] (v : Valuation C NNReal) [Algebra O C] (G : Over (Spec (CommRingCat.of O))) [GrpObj G] : HTChar p v G

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.InVVwIffOfNotTorsion

theorem **pow_eq_one_of_section** {A : Over S} [GrpObj A] {T T' : Over S} (u : T' ⟶ T) (v : T ⟶ T') (e : v ≫ u = 𝟙 T) (x : T ⟶ A) (N : ℕ) (h : (u ≫ x) ^ N = 1) : x ^ N = 1

theorem **exists_section_top** (T : Over S) : ∃ v : T ⟶ overRes (⊤ : T.left.Opens), v ≫ overResι ⊤ = 𝟙 T

theorem **inVVw_iff_of_not_torsion** (P : PolAV S) (N m : ℕ) {T : Over S} {I : Type} (h : I → (T ⟶ P.A)) (hh : ∀ i, h i ^ N ≠ 1) (χ : I → Γ(T.left, ⊤)ˣ) (s : LB.Sec (LPow P.A P.L N T)) : InVVw P N m h χ s ↔ Odd N ∧ IsUnit (N : Γ(T.left, ⊤)) ∧ InVV P N m s

theorem **exists_every_weight** : ∃ (P : PolAV (Spec (CommRingCat.of ℂ))) (T : Over (Spec (CommRingCat.of ℂ))) (x : T ⟶ P.A) (s : LB.Sec (LPow P.A P.L (3 ^ 2 : ℕ) T)), P.g = 1 ∧ Nonempty T.left ∧ s.f ≠ 0 ∧ (∀ χ : Γ(T.left, ⊤)ˣ, InVVw P (3 ^ 2) 3 (fun _ : Unit => x) (fun _ => χ) s) ∧ ∀ χ : Γ(T.left, ⊤)ˣ, ¬ InVVw' P (3 ^ 2) 3 (fun _ : Unit => x) (fun _ => χ) s

theorem **exists_symLift_top** (P : PolAV S) {N : ℕ} (hN : Odd N) {T : Over S} (x : T ⟶ P.A) (hx : x ^ N = 1) : ∃ U : ThetaGrp P.A P.L N T, SymLift P.A P.L N N x U

theorem **act_pull** (P : PolAV S) {N : ℕ} (hN : Odd N) {T T' : Over S} (u : T' ⟶ T) {x : T ⟶ P.A} {θ : ThetaGrp P.A P.L N T} {θ' : ThetaGrp P.A P.L N T'} (H : SymLift P.A P.L N N x θ) (H' : SymLift P.A P.L N N (u ≫ x) θ') {s t : LB.Sec (LPow P.A P.L N T)} (a : IsActOf P.A P.L N θ s t) : IsActOf P.A P.L N θ' (pullS P.A N u s) (pullS P.A N u t)

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.KFinDecayOfDecInf

theorem **gapOK_inhabited** (k r : ℕ) (hk : 1 ≤ k) : ∃ h : ℝ, GapOK k r h

theorem **thresholds** (k r : ℕ) (h : ℝ) (hg : GapOK k r h) : AG k r < (aOf r h − 1) / 2 ∧ AS k r < (aOf r h − 1) / 2 ∧ (2 ∗ Mexp r ∗ AS k r + 1) + AG k r + 1 ≤ aOf r h − 1

theorem **decay_range** (k r : ℕ) (h : ℝ) (h120 : 120 ∗ ((2 ∗ r + k + 1 : ℕ) : ℝ) ^ 3 ≤ h) : 120 ∗ ((2 ∗ r + k + 1 : ℕ) : ℝ) ∗ (((r : ℝ) + 1) ∗ Real.sqrt ((r : ℝ) + 1)) ≤ h

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.LatticeAndConstantUnique

theorem **comp_nonempty** {D : D7} (T : Core D) (j : ℕ) (σ : T.Q.c.F →+* ℂ) : Nonempty (T.U j σ).Comp

theorem **le_of_span_le** {E : Type} [Field E] [NumberField E] {r : ℕ} (A : FractionalIdeal (nonZeroDivisors (𝓞 E)) E) (hA : A ≠ 0) (L₁ L₂ : Submodule (𝓞 E) (Fin r → E)) (h : Submodule.span (𝓞 E) {v | ∃ a ∈ A, ∃ u ∈ L₁, v = a • u} ≤ Submodule.span (𝓞 E) {v | ∃ a ∈ A, ∃ u ∈ L₂, v = a • u}) : L₁ ≤ L₂

theorem **AA_ne_zero** (D : D7) : D.host.AA ≠ 0

theorem **lattice_and_constant_unique** {D : D7} (T : Core D) (j : ℕ) (σ : T.Q.c.F →+* ℂ) (k : (T.U j σ).Comp) (X Y : DataX D.host.E D.R.r) (hX : Fits T j σ k X) (hY : Fits T j σ k Y) : X.L' = Y.L' ∧ X.k0 = Y.k0

theorem **indices_unbounded** {D : D7} (T : Core D) (j : ℕ) (σ : T.Q.c.F →+* ℂ) (k : (T.U j σ).Comp) (X : DataX D.host.E D.R.r) (hX : Fits T j σ k X) (B : ℕ) : ∃ Y : DataX D.host.E D.R.r, Fits T j σ k Y ∧ B < Y.NA ∧ B < Y.NC

theorem **holds_by_value** (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ) (hj : InR D j) (σ : T.Q.c.F →+* ℂ) (hσ : GoodEmb T.Q.c σ) (k : (T.U j σ).Comp) : ∃ X : DataX D.host.E D.R.r, Fits T j σ k X ∧ X.k0 = 2 ∗ xiR D ∧ X.NA = Ideal.absNorm D.host.𝔟 ∧ X.NC = Ideal.absNorm D.host.ccK

theorem **log_absNorm_le** (D : D7) : Real.log (Ideal.absNorm D.host.𝔟 : ℕ) ≤ (fdeg D.host.E : ℝ) ∗ Real.log (ReciprocityTheory.ChartRestriction.G.FD.rdE D.host.E)

theorem **holds_without_rows** (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ) (hj : InR D j) (σ : T.Q.c.F →+* ℂ) (hσ : GoodEmb T.Q.c σ) (k : (T.U j σ).Comp) : ∃ X : DataX D.host.E D.R.r, Fits T j σ k X ∧ X.k0 = 2 ∗ xiR D ∧ Real.log X.NA ≤ (fdeg D.host.E : ℝ) ∗ Real.log (ReciprocityTheory.ChartRestriction.G.FD.rdE D.host.E) ∧ X.NC = Ideal.absNorm D.host.ccK

theorem **intersectionNumber_baseChange** (D : D7) (T : Top D) (j : ℕ) : Formalism.L2.Pj.QAt.DataXOfModelSt T j

theorem **indices_bounded** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : DataX E r) (a b : ℝ) (hA : Real.log X.NA ≤ a) (hC : Real.log X.NC ≤ b) : (X.NA : ℝ) ≤ Real.exp a ∧ (X.NC : ℝ) ≤ Real.exp b

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.LowKAddStrictAnti

theorem **rhoN_add_rhoC** {n : ℕ} (ε : Fin n → Bool) (a : Fin n) : rhoN ε a + rhoC ε a = ((n : ℝ) − 1) / 2 − a

theorem **lowK_add** {n : ℕ} (ε : Fin n → Bool) (lam : Fin n → ℝ) (a : Fin n) : (lowK ε lam + (2 : ℝ) • rhoC ε) a = lam a + (((n : ℝ) − 1) / 2 − a)

theorem **lowK_add_strictAnti** {n : ℕ} (ε : Fin n → Bool) {lam : Fin n → ℝ} (h : StrictAnti lam) : StrictAnti (lowK ε lam + (2 : ℝ) • rhoC ε)

theorem **torus_mem** {n : ℕ} (ε : Fin n → Bool) (θ : Fin n → ℝ) : Matrix.diagonal (fun a => Complex.exp ((θ a : ℂ) ∗ Complex.I)) ∈ Kset ε

theorem **wt_int** {n : ℕ} {H : Type∗} [NormedAddCommGroup H] [InnerProductSpace ℂ H] {ε : Fin n → Bool} {σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)} (h1 : σ 1 = 1) {μ : Fin n → ℝ} {v : H} (hv : v ≠ 0) (h : IsWtOn (Kset ε) σ μ v) (a : Fin n) : ∃ m : ℤ, μ a = m

theorem **isHCParam_of_isDS** {n : ℕ} {H : Type∗} [NormedAddCommGroup H] [InnerProductSpace ℂ H] {ε : Fin n → Bool} {lam : Fin n → ℝ} {σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)} (hl : StrictAnti lam) (h : IsDS ε lam σ) : IsHCParam lam

theorem **norm_entry** {ε : Fin 1 → Bool} {g : Matrix (Fin 1) (Fin 1) ℂ} (hg : inUJ ε g) : ‖g 0 0‖ = 1

theorem **wt_eq** {ε : Fin 1 → Bool} (m : ℤ) (μ : Fin 1 → ℝ) (v : ℂ) (hv : v ≠ 0) (h : IsWtOn (Kset ε) (chi m) μ v) : μ = fun _ => (m : ℝ)

theorem **lowK_one** (ε : Fin 1 → Bool) (lam : Fin 1 → ℝ) : lowK ε lam = lam

theorem **compact_one** (ε : Fin 1 → Bool) : CompactSpace (UJ ε)

theorem **isDS_chi_iff** (ε : Fin 1 → Bool) (m : ℤ) (lam : Fin 1 → ℝ) : IsDS ε lam (chi m) ↔ lam = fun _ => (m : ℝ)

theorem **order_needed** : lowK ![true, false] ![1 / 2, 7 / 2] = lowK ![false, true] ![5 / 2, 3 / 2] ∘ Equiv.swap 0 1

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.MatrixExponentialEntries

theorem **norm_entry** {ε : Fin 1 → Bool} {g : M₁} (hg : inUJ ε g) : ‖g 0 0‖ = 1

theorem **wt_eq** {ε : Fin 1 → Bool} (m : ℤ) (μ : Fin 1 → ℝ) (v : ℂ) (hv : v ≠ 0) (h : IsWtOn (Kset ε) (chi m) μ v) : μ = fun _ => (m : ℝ)

theorem **compact_one** (ε : Fin 1 → Bool) : CompactSpace (UJ ε)

theorem **isDS_chi** (ε : Fin 1 → Bool) (m : ℤ) : IsDS ε (fun _ => (m : ℝ)) (chi m)

theorem **exp_single_zero_zero_apply** (z : ℂ) : (NormedSpace.exp (z • (Matrix.single 0 0 1 : M₁))) 0 0 = Complex.exp z

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.NaturalOfChartIndep

theorem **frBCField** : FrBCFieldSt Xo c

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.NotIsProjOverAffineLine

theorem **all_six** : PullCubCompSt ∧ PullCubCancelSt ∧ SABCompSt ∧ SABCancelSt ∧ AbLocusOneSt ∧ ProjProperSt

theorem **base_eq** (m : CubeMap G' G) : m.e = eSec G' ≫ m.F ≫ G.hom

theorem **cubeMap_ext** (m n : CubeMap G' G) (h : m.F = n.F) : m = n

theorem **sabMap_ext** {t : SAB H S} {t' : SAB H S'} {f : S' ⟶ S} (m n : SABMap t' t f) (h : m.φ = n.φ) : m = n

theorem **existsUnique_cancel** {t : SAB H S} {t' : SAB H S'} {t'' : SAB H S''} {f : S' ⟶ S} {g : S'' ⟶ S'} (m : SABMap t' t f) (n : SABMap t'' t (g ≫ f)) : ∃! m' : SABMap t'' t' g, n.φ = m'.φ ≫ m.φ

theorem **abLocus_torus** (k : Type) [Field k] [IsAlgClosed k] [CharZero k] : IsSemiAbelian (splitTorus (Spec (CommRingCat.of k)) ℤ) ∧ abLocus (splitTorus (Spec (CommRingCat.of k)) ℤ) = ∅

theorem **not_isProjOver_affineLine** (R : Type) [CommRing R] [Nontrivial R] : ¬ IsProjOver (spMap (Polynomial.C : R →+* Polynomial R))

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.NotIsThetaOfNonunit

theorem **exists_nonzero_closed** (hS : SecBaseChangeSt) {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] [IsNoetherianRing R] (D : DegPre R) : ∃ th : D.SecB 1 0, ∃ x, D.P.A.hom x = closedPoint R ∧ x ∈ nonzeroLocus th

theorem **nonzero_and_spans** (hG : GlobalFnsSt) (hS : SecBaseChangeSt) [IsNoetherianRing R] (hI : I ≤ maximalIdeal R) (U : Unif I D A G) {θ : H0 U.LG.L 1} (hθ : IsTheta U θ) : (∃ g, G.hom g = closedPoint R ∧ g ∈ nonzeroLocus θ) ∧ ∀ σ : H0 U.LG.L 1, ∃ b : R, SecIsMul G.hom b σ θ

theorem **unit_mul_iff_nonzero** (hG : GlobalFnsSt) (hS : SecBaseChangeSt) [IsNoetherianRing R] (hI : I ≤ maximalIdeal R) (U : Unif I D A G) {θ : H0 U.LG.L 1} (hθ : IsTheta U θ) (σ : H0 U.LG.L 1) : (∃ u : Rˣ, SecIsMul G.hom (u : R) σ θ) ↔ NonzeroOnSpecial U σ

theorem **spans_iff_nonzero** (hG : GlobalFnsSt) (hS : SecBaseChangeSt) [IsNoetherianRing R] (hI : I ≤ maximalIdeal R) (U : Unif I D A G) {θ : H0 U.LG.L 1} (hθ : IsTheta U θ) (σ : H0 U.LG.L 1) : (∀ s : H0 U.LG.L 1, ∃ b : R, SecIsMul G.hom b s σ) ↔ NonzeroOnSpecial U σ

theorem **exists_isTheta_of_dvr** (hLaws : Laws) (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (D : DegData V (maximalIdeal V)) (A : PolAV (Spec (CommRingCat.of (FractionRing V)))) (G : Over (Spec (CommRingCat.of V))) [GrpObj G] [IsCommMonObj G] (U : Unif (maximalIdeal V) D A G) : ∃ θ : H0 U.LG.L 1, IsTheta U θ ∧ ∀ s : H0 U.LG.L 1, ∃ b : V, SecIsMul G.hom b s θ

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.NotJInjRdOfConst

theorem **jInjRd_of_laws** {D : D7} (T : Top D) (hM : LawsM T.toCore) {j : ℕ} (hj : InR D j) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) : JInjRd T j a n

theorem **sum_toLp_ae** {α : Type} [MeasurableSpace α] (μ : Measure α) {m : ℕ} (φ : Fin m → α → ℂ) (hφ : ∀ i, MemLp (φ i) 2 μ) (g : Fin m → ℂ) (s : Finset (Fin m)) : ((∑ i ∈ s, g i • (hφ i).toLp (φ i) : Lp ℂ 2 μ) : α → ℂ) =ᵐ[μ] ∑ i ∈ s, g i • φ i

theorem **sum_ae** {m : ℕ} (b : Fin m → C) (g : Fin m → ℂ) : ((∑ i, g i • uniformisationL2 V σ rd dict mu h (b i) : Lp ℂ 2 (muQ1 V)) : AutQ V → ℂ) =ᵐ[muQ1 V] onQ V (adel V dict mu (∑ i, g i • rd (b i)))

theorem **indepToC_of_jInj** (hJ : JInj V σ rd dict mu h) : IndepToC rd

theorem **jInjRd_of_jInjAt** {D : D7} {T : Top D} (h : LawsP T) (j : ℕ) (hj : InR D j) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (hd : Dom a n) (hJ : JInjAt h j hj a n hd) : JInjRd T j a n

theorem **jInjAt_of_aeDom** {D : D7} {T : Top D} (h : LawsP T) (j : ℕ) (hj : InR D j) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (hd : Dom a n) (hae : AeDom (T.U j) T.σ₀ (T.V j T.σ₀) (T.dict j T.σ₀) (T.mu j T.σ₀)) : JInjAt h j hj a n hd

theorem **jInjAt_of_aut** {D : D7} {T : Top D} (h : LawsP T) (j : ℕ) (hj : InR D j) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (hd : Dom a n) (hA : AdelAut (T.V j T.σ₀) (T.dict j T.σ₀) (T.mu j T.σ₀)) (hP : ∃ p : Measure (GA D (T.V j T.σ₀)) × Set (GA D (T.V j T.σ₀)), IsVolPair D (T.V j T.σ₀) p.1 p.2) : JInjAt h j hj a n hd

theorem **not_jInjRd_of_const** {D : D7} (T : Top D) (j : ℕ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (y : FrPt (T.Q.Xo j) T.Q.c (CAt T.σ₀)) (hy : ∀ k x, (T.U j T.σ₀).un k x = y) (b : Fin 2 → cusp (T.Q.Xo j) T.Q.c (T.U j) a n) (hb : LinearIndependent T.Q.c.F b) : ¬ JInjRd T j a n

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.NotSurjectiveSplitTorus

theorem **premises_met** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : IsAdicComplete (IsLocalRing.maximalIdeal ℤ_[3]) ℤ_[3] ∧ ∃ t : SAB H (Sp ℤ_[3]), IsProper ((Over.pullback (spMap (algebraMap ℤ_[3] (FractionRing ℤ_[3])))).obj t.G).hom ∧ ¬ IsIso t.G.hom

theorem **not_surjective_splitTorus** (S : Scheme.{0}) [Nonempty S] (X : Type) [AddCommGroup X] [Nontrivial X] : ¬ Function.Surjective (splitTorus S X).hom.appTop.hom

theorem **properness_needed** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (V : Type) [CommRing V] [IsDomain V] : ∃ t : SAB H (Sp V), ¬ Function.Bijective (powObj t.G 1).hom.appTop.hom

theorem **not_without_properness** : ¬ ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V], IsAdicComplete (IsLocalRing.maximalIdeal V) V → ∀ t : SAB H (Sp V), ∀ n : ℕ, Function.Bijective (powObj t.G n).hom.appTop.hom

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.OrientationExclusive

theorem **choose_two_add** (p q : ℕ) : (p + q).choose 2 = p * q + p.choose 2 + q.choose 2

theorem **dimY_eq** (Tp : ParamsT D L P) {j : ℕ} (hj : InR D j) : WeightFamilies.dimY D j = Tp.Dp

theorem **kappa_last** (Tp : ParamsT D L P) (hK : DV.KapDef D) {j : ℕ} (hj : InR D j) (τ : D.what₂.Tau) (i : Fin (D.R.sigW j).1) (hi : i.1 + 1 = (D.R.sigW j).1) : 2 * (Tp.TO : ℤ) ^ 2 ≤ ((kaR Tp hK j).aP τ i : ℤ)

theorem **lR_lastPos** (Tp : ParamsT D L P) (hK : DV.KapDef D) {j : ℕ} (hj : InR D j) : (lR Tp hK j).LastPos

theorem **one_le_DF** (Tp : ParamsT D L P) (hK : DV.KapDef D) {j : ℕ} (hj : InR D j) (τ : D.what₂.Tau) : 1 ≤ WeightAlgebra.DF (lR Tp hK j)

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.PairingIsoPullback

theorem **pullCongr_heq** {X X' : Scheme.{0}} {L : LB X} {L' : LB X'} {F F' : X' ⟶ X} (h : F = F') (w : LB.Iso L' (L.pull F)) : HEq (pullCongr h w) w

theorem **exists_comparison** : PairThetaSt

theorem **exists_local_gen** {X Y : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (y : Y) : ∃ V : Y.Opens, y ∈ V ∧ ∃ t : LB.Gen (LB.LinComb M (fun k => V.ι ⁻¹ᵖ φ k) ε), t.IsGen

theorem **exists_theta** {k : ℕ} {T : Over S} {x : T ⟶ A} (hx : x ^ k = 1) {K : Type} [Fintype K] {φ : K → (T.left ⟶ A.left)} {ε : K → ℤ} (v : LB.Gen (LB.LinComb L.L φ ε)) (hv : v.IsGen) (hF : ∀ (U : Type) [CommGroup U] (h : (T.left ⟶ A.left) → U), ∏ i, h (φ i) ^ ε i = h (1 : T ⟶ A).left ^ (k : ℤ) * h x.left ^ (-(k : ℤ))) : ∃ θ : ThetaGrp A L.L k T, θ.pt = x

theorem **exists_weil** {k : ℕ} {T : Over S} {x y : T ⟶ A} (hx : x ^ k = 1) (hy : y ^ k = 1) : ∃ w : Γ(T.left, ⊤)ˣ, ∀ (T' : Over S) (σ : T' ⟶ T) (z : T'.left) (J), Good L.L z J → gm σ.left z ⊤ w = cubN (cubV c z J) k 1 (σ ≫ y) (σ ≫ x) * (cubN (cubV c z J) k 1 (σ ≫ x) (σ ≫ y))⁻¹

theorem **thG_mul** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {k : ℕ} {T : Over S} (W W' : ThetaGrp A L.L k T) (hW : W.pt ^ k = 1) (hW' : W'.pt ^ k = 1) {T' : Over S} (σ : T' ⟶ T) (y : T' ⟶ A) {z : T'.left} {J} (hJ : Good L.L z J) : val (LB.mulGen L.L _ _ _ _ (thG W) (LB.pullGen L.L _ _ (thG W') (translT A W.pt).left)) (lift σ y).left z J = val (thG W) (lift σ 1).left z J ∗ cubN (cubV c z J) k 1 (σ ≫ W.pt) y ∗ (val (thG W') (lift σ 1).left z J ∗ cubN (cubV c z J) k 1 (σ ≫ W'.pt) ((σ ≫ W.pt) ∗ y))

theorem **comm_val** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {k : ℕ} {T : Over S} (θ θ' : ThetaGrp A L.L k T) (hx : θ.pt ^ k = 1) (hy : θ'.pt ^ k = 1) (e : Γ(T.left, ⊤)ˣ) {T' : Over S} (σ : T' ⟶ T) (y : T' ⟶ A) {z : T'.left} {J} (hJ : Good L.L z J) : val (LB.mulGen L.L _ _ _ _ (thG θ) (LB.pullGen L.L _ _ (thG θ') (translT A θ.pt).left)) (lift σ y).left z J = val (LB.mulGen L.L _ _ _ _ (LB.mulGen L.L _ _ _ _ (thG θ') (LB.pullGen L.L _ _ (thG θ) (translT A θ'.pt).left)) (constGen A L.L e)) (lift σ y).left z J ↔ gm σ.left z ⊤ e = cubN (cubV c z J) k 1 (σ ≫ θ'.pt) (σ ≫ θ.pt) ∗ (cubN (cubV c z J) k 1 (σ ≫ θ.pt) (σ ≫ θ'.pt))⁻¹

theorem **weilIs_iff** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {k : ℕ} {T : Over S} {x y : T ⟶ A} (hx : x ^ k = 1) (hy : y ^ k = 1) {w : Γ(T.left, ⊤)ˣ} (hw : ∀ (T' : Over S) (σ : T' ⟶ T) (z : T'.left) (J), Good L.L z J → gm σ.left z ⊤ w = cubN (cubV c z J) k 1 (σ ≫ y) (σ ≫ x) ∗ (cubN (cubV c z J) k 1 (σ ≫ x) (σ ≫ y))⁻¹) (e : Γ(T.left, ⊤)ˣ) : WeilIs A L.L k x y e ↔ e = w

theorem **ros_pair** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (f g : A ⟶ A) [IsMonHom f] [IsMonHom g] (u : LB.Gen (LB.LinComb L.L (rosatiφ A f g) rosatiε)) (hu : u.IsGen) (n : LB.Gen.Corr L.L _ _ _ _ (LB.pullGen L.L _ _ u (zero2 A)) (LB.oneGen L.L)) {T : Over S} {z : T.left} {J} (hJ : Good L.L z J) {k : ℕ} {x y : T ⟶ A} (hx : x ^ k = 1) (hy : y ^ k = 1) : cubN (cubV c z J) k 1 y (x ≫ f) ∗ (cubN (cubV c z J) k 1 (x ≫ f) y)⁻¹ = cubN (cubV c z J) k 1 (y ≫ g) x ∗ (cubN (cubV c z J) k 1 x (y ≫ g))⁻¹

theorem **weilAdjoint** (hG : GlobalFnsSt) (hC : CubeSt) : WeilAdjoint

theorem **pairing_weilMove** (hW : Out.WeilSt) {S S' X : Scheme.{0}} [IsLocallyNoetherian S'] (hX : Subsingleton X) {G : Over S} [GrpObj G] {L₀ : LB G.left} (hL₀ : L₀.IsLB) (P₁ : Out.PolAV S') {f : S' ⟶ S} {φ : P₁.A.left ⟶ G.left} (hp : IsGrpPullback P₁.A G f φ) (w : LB.Iso P₁.L (L₀.pull φ)) (hw : w.IsGen) {tS : X ⟶ S'} {t₀ : X ⟶ S} (ht : t₀ = tS ≫ f) (n : ℕ) (x y : Over.mk tS ⟶ P₁.A) (hx : x ^ n = 1) (hy : y ^ n = 1) (x₀ y₀ : Over.mk t₀ ⟶ G) (hx₀ : x₀.left = x.left ≫ φ) (hy₀ : y₀.left = y.left ≫ φ) (c : Γ(X, ⊤)ˣ) (hG : Out.WeilIs G L₀ n x₀ y₀ c) : Out.WeilIs P₁.A P₁.L n x y c

theorem **pairing_isoPullback** {S : Scheme.{0}} (B : Out.PolAV S) (θ : B.A ≅ B.A) (hθ : IsMonHom θ.hom) : IsGrpPullback B.A B.A (𝟙 S) θ.hom.left

theorem **pairing_weilTheta** (hW : Out.WeilSt) {S X : Scheme.{0}} [IsLocallyNoetherian S] (hX : Subsingleton X) (B : Out.PolAV S) (θ : B.A ≅ B.A) (hθ : IsMonHom θ.hom) (wθ : LB.Iso B.L (B.L.pull θ.hom.left)) (hwθ : wθ.IsGen) {t : X ⟶ S} (n : ℕ) (x y : Over.mk t ⟶ B.A) (hx : x ^ n = 1) (hy : y ^ n = 1) (c : Γ(X, ⊤)ˣ) (h : Out.WeilIs B.A B.L n (x ≫ θ.hom) (y ≫ θ.hom) c) : Out.WeilIs B.A B.L n x y c

theorem **pairing_weilInv** (hW : Out.WeilSt) {S : Scheme.{0}} [IsLocallyNoetherian S] (B : Out.PolAV S) {T : Over S} (n : ℕ) (p z : T ⟶ B.A) (hp : p ^ n = 1) (hz : z ^ n = 1) (c : Γ(T.left, ⊤)ˣ) (h : Out.WeilIs B.A B.L n p z c) : Out.WeilIs B.A B.L n p⁻¹ z c⁻¹

theorem **pairing_nondeg** (hW : Out.WeilSt) (hND : Out.WeilNondegSt) (hBC : Out.PolBaseChangeSt) {S : Scheme.{0}} (B : Out.PolAV S) (F : Type) [Field F] [IsAlgClosed F] (t : Spec (CommRingCat.of F) ⟶ S) (n : ℕ) (hn : (n : F) ≠ 0) (u : Over.mk t ⟶ B.A) (hu : u ^ n = 1) (h : ∀ z : Over.mk t ⟶ B.A, z ^ n = 1 → Out.WeilIs B.A B.L n u z 1) : u = 1

theorem **pairing_act_add** (z : T ⟶ B.A) (a b : O) : z ≫ o.act (a + b) = (z ≫ o.act a) ∗ (z ≫ o.act b)

theorem **pairing_act_zero** (z : T ⟶ B.A) : z ≫ o.act 0 = 1

theorem **pairing_act_nat** (z : T ⟶ B.A) (n : ℕ) : z ≫ o.act (n : O) = z ^ n

theorem **pairing_act_mul** (z : T ⟶ B.A) (a b : O) : z ≫ o.act (a ∗ b) = (z ≫ o.act b) ≫ o.act a

theorem **pairing_act_kill** (z : T ⟶ B.A) (n : ℕ) (hz : z ^ n = 1) (r : O) : z ≫ o.act (r ∗ n) = 1

theorem **pairing_tors** {S : Scheme.{0}} {G G' : Over S} [GrpObj G] [GrpObj G'] {T : Over S} (z : T ⟶ G) (f : G ⟶ G') (hf : IsMonHom f) (n : ℕ) (hz : z ^ n = 1) : (z ≫ f) ^ n = 1

theorem **halfToFullTorsion** (hR : WeilAdjoint) (hW : Out.WeilSt) (hND : Out.WeilNondegSt) (hBC : Out.PolBaseChangeSt) : HalfToFullTorsion

theorem **ext_of_schDom** {W X Y Z : Scheme.{0}} {f g : X ⟶ Y} (s : Y ⟶ Z) [IsSeparated s] (h : f ≫ s = g ≫ s) (ι : W ⟶ X) [IsSchemeTheoreticallyDominant ι] (hU : ι ≫ f = ι ≫ g) : f = g

theorem **schDom_spec_map** {A B : Type} [CommRing A] [CommRing B] [IsReduced A] (φ : A →+* B) (hφ : Function.Injective φ) : IsSchemeTheoreticallyDominant (Spec.map (CommRingCat.ofHom φ))

theorem **over_ext_of_points** {F : Type} [Field F] [IsAlgClosed F] {K Y : Over (Spec (CommRingCat.of F))} [LocallyOfFiniteType K.hom] [FormallyUnramified Y.hom] [LocallyOfFiniteType Y.hom] (u v : K ⟶ Y) (h : ∀ p : 𝟙_ (Over (Spec (CommRingCat.of F))) ⟶ K, p ≫ u = p ≫ v) : u = v

theorem **fix_torsion** (hMul : Out.MulNSt) {F : Type} [Field F] [IsAlgClosed F] (P : Out.PolAV (Spec (CommRingCat.of F))) (M : ℕ) (hM1 : 1 ≤ M) (hM : (M : F) ≠ 0) (σ : P.A ⟶ P.A) [IsMonHom σ] (hσ : ∀ x : 𝟙_ (Over (Spec (CommRingCat.of F))) ⟶ P.A, x ^ M = 1 → x ≫ σ = x) (T : Over (Spec (CommRingCat.of F))) (x : T ⟶ P.A) (hx : x ^ M = 1) : x ≫ σ = x

theorem **isoInv_isGen** {X : Scheme.{0}} {L M : LB X} {t : LB.Iso L M} (ht : t.IsGen) : (isoInv t).IsGen

theorem **pullGenLB_isGen** {X Y : Scheme.{0}} {L : LB X} {c : LB.Gen L} (hc : c.IsGen) (h : Y ⟶ X) : (pullGenLB c h).IsGen

theorem **exists_gen_congr** {X Y : Scheme.{0}} {L : LB X} {f f' g : Y ⟶ X} (h : f = f') {ε : Fin 2 → ℤ} (H : ∃ c : LB.Gen (LB.LinComb L (K

theorem **serreFromPoints** (hS : Out.SerreSt) (hMul : Out.MulNSt) (hBC : Out.PolBaseChangeSt) : SerreFromPoints

theorem **fixesHalfTorsion** : FixesHalfTorsion

theorem **all_four** : FixesHalfTorsion ∧ HalfToFullTorsion ∧ SerreFromPoints ∧ WeilAdjoint

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.PhiCOfNoIntegralForm

theorem **phiC_def** (b : SymB cj Y) : phiC cj b = sInf (vals b)

theorem **phiC_of_subsingleton** [Subsingleton Y] (b : SymB cj Y) : phiC cj b = 0

private lemma **nat_mul_mem_vals** {b : SymB cj Y} {x : ℝ} (hx : x ∈ vals b) (n : ℕ) (hn : 0 < n) : (n : ℝ) ∗ x ∈ vals b

theorem **phiC_eq_zero_of_neg** (b : SymB cj Y) {x : ℝ} (hx : x ∈ vals b) (hneg : x < 0) : phiC cj b = 0

theorem **symB_eq_bot_of_apply_one** (h : cj 1 = 0) : SymB cj Y = ⊥

private lemma **pos_mul_mem_vals_of_integral_eq_zero** (h : ∀ β : SymB cj Y, (∀ a c, ∃ z : ℤ, (β : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ) a c = z) → β = 0) {b : SymB cj Y} {x : ℝ} (hx : x ∈ vals b) {c : ℝ} (hc : 0 < c) : c ∗ x ∈ vals b

theorem **phiC_of_no_integral_form** (h : ∀ β : SymB cj Y, (∀ a c, ∃ z : ℤ, (β : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ) a c = z) → β = 0) (b : SymB cj Y) : phiC cj b = 0

private lemma **vals_eq_functionalValues** [IsCMField E] (b : SymB cj Y) : vals b = functionalValues b

theorem **one_le_phiC** (h : IsConj cj) (b : SymB cj Y) (hint : ∀ a c, ∃ z : ℤ, (b : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ) a c = z) (hpos : ∀ a, 0 ≤ (b : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ) a a) (h0 : b ≠ 0) : 1 ≤ phiC cj b

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.PremisesInhabited

theorem **integral_of_sum** (Y : Type) [AddCommGroup Y] (S : Submodule ℝ (Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ)) (l : S →ₗ[ℝ] ℝ) (m : ℕ) (x x' : Fin m → Y) (h : ∀ β : S, l β = ∑ i, (β : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ) (x i) (x' i)) (β : S) (hβ : ∀ a c, ∃ z : ℤ, (β : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ) a c = z) : ∃ z : ℤ, l β = z

theorem **trBil_apply** (x y : 𝒪 E) : trBil E x y = ∑ σ : E →+* ℂ, (σ (x : E) * starRingEnd ℂ (σ (y : E))).re

theorem **trBil_symm** (x y : 𝒪 E) : trBil E x y = trBil E y x

theorem **trBil_balanced** (c c' : 𝒪 E) (hc : (c' : E) = IsCMField.complexConj E c) (x y : 𝒪 E) : trBil E (c • x) y = trBil E x (c' • y)

theorem **trBil_posDef** : (Matrix.of fun i j => trBil E (RingOfIntegers.basis E i) (RingOfIntegers.basis E j)).PosDef

theorem **exists_half** : ∃ w : 𝒪 K₃, (w : K₃) + IsCMField.complexConj K₃ w = 1

theorem **premises_inhabited** : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (cj : E → E) (w : 𝒪 E) (Y : Type) (_ : AddCommGroup Y) (_ : Module (𝒪 E) Y) (_ : Module.Free ℤ Y) (_ : Module.Finite ℤ Y) (β : Y →ₗ[ℤ] Y →ₗ[ℤ] ℝ), (∀ (φ : E →+* ℂ) (z : E), φ (cj z) = star (φ z)) ∧ (∃ z, cj z ≠ z) ∧ (∀ c : 𝒪 E, ∃ c' : 𝒪 E, (c' : E) = cj c) ∧ (w : E) + cj w = 1 ∧ 0 < Module.finrank ℤ Y ∧ (∀ x y, β x y = β y x) ∧ (∀ c c' : 𝒪 E, (c' : E) = cj c → ∀ x y, β (c • x) y = β x (c' • y)) ∧ ∃ (ι : Type) (_ : Fintype ι) (b : Module.Basis ι ℤ Y), (Matrix.of fun i j => β (b i) (b j)).PosDef

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.PremisesMetRankOne

theorem **sum_six** (c : Pic A.left) (f g : A ⟶ A) {T : Over S} (x y : T ⟶ A) : ∑ i, rosatiε i • pull ((lift x y).left ≫ rosatiφ A f g i) c = pull ((x ≫ f) ∗ y).left c − pull (x ≫ f).left c − pull y.left c − pull (x ∗ (y ≫ g)).left c + pull x.left c + pull (y ≫ g).left c

theorem **adjCls_iff** {L : LB A.left} (hL : L.IsLB) (f g : A ⟶ A) : (∃ t : LB.Gen (LB.LinComb L (rosatiφ A f g) rosatiε), t.IsGen) ↔ AdjCls (cls L hL) f g

theorem **eq_one_of_cls** {L : LB A.left} (hL : L.IsLB) (hK : KTriv A L) {T : Over S} (p : T ⟶ A) (d : Pic T.left) (h : pull ((fst T A ≫ p) ∗ snd T A).left (cls L hL) − pull (snd T A).left (cls L hL) = pull (fst T A).left d) : p = 1

theorem **pointed_of_adjCls** {L : LB A.left} (hL : L.IsLB) (hK : KTriv A L) {f g : A ⟶ A} (h : AdjCls (cls L hL) f g) : η[A] ≫ f = η[A] ∧ η[A] ≫ g = η[A]

theorem **adjCls_factor** {n : ℕ} (p : IsPowGrp A' A n) (i₀ : Fin n) (c : Pic A.left) {f g : A ⟶ A} {F G : A' ⟶ A'} [IsMonHom f] [IsMonHom g] (hF : ∀ i, F ≫ p.pr i = p.pr i ≫ f) (hG : ∀ i, G ≫ p.pr i = p.pr i ≫ g) (h : AdjCls (∑ i, pull (p.pr i).left c) F G) : AdjCls c f g

theorem **hom_and_adjoint** {D D' : DegPre R} {n : ℕ} (p : IsPowDeg D' D n) (i₀ : Fin n) {f g : D.P.A ⟶ D.P.A} {F G : D'.P.A ⟶ D'.P.A} (hF : DiagOnB p F f) (hG : DiagOnB p G g) (u : LB.Gen (LB.LinComb D'.P.L (rosatiφ D'.P.A F G) rosatiε)) (hu : u.IsGen) : IsMonHom f ∧ IsMonHom g ∧ RosatiAdj D.P.A D.P.L f g

theorem **powTauBalanced** : PowTauBalanced

theorem **powTauBalancedAct** : PowTauBalancedAct

theorem **tauBal_id** (D : DegPre R) : D.TauBal (𝟙 _) (𝟙 _) id id

theorem **premises_met** (I : Ideal R) (D : DegData R I) (n : ℕ) : ∃ (D' : DegData R I) (p : IsPowDeg D'.toPre D.toPre n), DiagOnB p (𝟙 _) (𝟙 _) ∧ DiagOnX p id id ∧ CLinear D.toPre (𝟙 _) id ∧ D.toPre.TauBal (𝟙 _) (𝟙 _) id id

theorem **premises_met_rank_one** (n : ℕ) : ∃ (D' : DegData InhTate.R InhTate.I) (p : IsPowDeg D'.toPre InhTate.tateData.toPre n), DiagOnB p (𝟙 _) (𝟙 _) ∧ DiagOnX p id id ∧ CLinear InhTate.tateData.toPre (𝟙 _) id ∧ InhTate.tateData.toPre.TauBal (𝟙 _) (𝟙 _) id id

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.SumExtendByZeroLeSum

theorem **g_coprime** (D : D7) (k : Fin 2) (p : ℕ) (hp : p ∈ D.host.condN.primeFactors) : Nat.Coprime (D.host.g k) p

theorem **one_le_vLa** {K : Type} [Field K] [NumberField K] (la : HeightOneSpectrum (𝒪 K)) (p : ℕ) (hp : p ≠ 0) (hla : (p : 𝒪 K) ∈ la.asIdeal) : 1 ≤ vLa la (p : K)

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.ThetaMemLattice

theorem **choose_two_add** (p q : ℕ) : (p + q).choose 2 = p * q + p.choose 2 + q.choose 2

theorem **dimY_eq_flag** {j : ℕ} (hj : InR D j) : WeightFamilies.dimY D j = D.R.f * D.R.r.choose 2

theorem **two_sq_le_last** (Tp : ParamsT D L P) (hK : DV.KapDef D) {j : ℕ} (hj : InR D j) (τ : D.what₂.Tau) (i : Fin (D.R.sigW j).1) (hi : i.1 + 1 = (D.R.sigW j).1) : 2 * (Tp.TO : ℤ) ^ 2 ≤ ((kaR Tp hK j).aP τ i : ℤ)

theorem **lastPos_run** (Tp : ParamsT D L P) (hK : DV.KapDef D) {j : ℕ} (hj : InR D j) : (lR Tp hK j).LastPos

theorem **theta_mem_lattice** (Tp : ParamsT D L P) (hK : DV.KapDef D) {T : Top D} (h : LawsP T) {j : ℕ} (hj : InR D j) {f : T.R j} (hf : f ∈ T.th j (lR Tp hK j)) : ∃ F ∈ (recPJ h j hj).thAll (lR Tp hK j), F.1 = f

theorem **depth_scale** (s : 𝒪 B.KO) (hs : s ≠ 0) (w : WeightFamilies.Wt D j) (la : HeightOneSpectrum (𝒪 B.KO)) (y : X.Pt la) : X.depth s w la y = X.depth 1 w la y + (((vAdd la (s : B.KO) : ℤ) : ℝ) : EReal)

theorem **level_eq** {T : Top D} (h : LawsP T) : (T.Q.l.M : ℕ) = D.host.g 0 * D.host.g 1

theorem **main** (Tp : ParamsT D L P) (hK : DV.KapDef D) {T : Top D} (h : LawsP T) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (nuWW : ℕ → HeightOneSpectrum (𝒪 T.Q.c.F) → ℝ) {j : ℕ} (hj : InR D j) (la : HeightOneSpectrum (𝒪 T.Q.c.F)) (hg : ∀ i : Fin 2, ((D.host.g i : ℕ) : 𝒪 T.Q.c.F) ∉ la.asIdeal) (y : (recPJ h j hj).Pt la) : (recPJ h j hj).depth (runOf Tp hK (recB h.toLawsCore) hinj EX nuWW j).S.sW (lR Tp hK j) la y = (recPJ h j hj).depth 1 (lR Tp hK j) la y + (((ramOverZ la : ℝ) * clOf (lR Tp hK j) la : ℝ) : EReal) ∧ (∀ f ∈ T.th j (lR Tp hK j), ∃ F ∈ (recPJ h j hj).thAll (lR Tp hK j), F.1 = f) ∧ (lR Tp hK j).n = Tp.nWn ∧ (T.Q.l.M : ℕ) = D.host.g 0 * D.host.g 1

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.UnimodularStandardLattice

theorem **mem_stdLat** (v : Fin n → E) : v ∈ stdLat E n ↔ ∀ j, ∃ b : 𝒪 E, (b : E) = v j

private lemma **mem_range_algebraMap_iff_isIntegral_int** (x : E) : x ∈ (algebraMap (𝒪 E) E).range ↔ IsIntegral ℤ x

private lemma **complexConj_mem_integers_iff** (x : E) : IsCMField.complexConj E x ∈ (algebraMap (𝒪 E) E).range ↔ x ∈ (algebraMap (𝒪 E) E).range

theorem **_root_.GlobalTheory.Survey.UnimodularStandardLattice.unimod_std** : ((_root_.Formalism.L2.Pj.stdLat E n) : _root_.Formalism.L2.Pj.GenLat E n).Unimod

theorem **trivAt_std** (bad : ℕ) : ((stdLat E n) : GenLat E n).TrivAt bad

Development.AbelianVarietyTheory.Counterexamples.JetSurvey.WitnessIsFormalParam

theorem **tanBasis_of_lieBasis** {n : ℕ} (v : Fin n → (epsR S ⟶ (Over.pullback x0).obj t.G)) (hB : IsLieBasis v) : ∃ tg : Fin n → (Sp (DualNumber S) ⟶ t.G.left), IsTanBasisG t x0 tg

theorem **premises_met** [IsLocalRing S] : ∃ (n : ℕ) (tg : Fin n → (Sp (DualNumber S) ⟶ t.G.left)), IsTanBasisG t x0 tg

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.AdelEqZeroOfOffDom

theorem **rdU_eq_zero** (U : Unf Xo c σ r) (f : Rul Xo c) (h : ∀ k, ∀ x ∈ (U.X k).dom, rdU U f k x = 0) : rdU U f = 0

theorem **indep_one** {K M : Type} [Field K] [AddCommGroup M] [Module K M] {f : M} (h : f ≠ 0) : LinearIndependent K (fun _ : Fin 1 => f)

theorem **inj_of_indep** (U : Unf Xo c σ r) (h : RdIndep1 U) (a n) (hd : Dom a n) (f) (hf : f ∈ sec Xo c a n) (h0 : ∀ k, ∀ x ∈ (U.X k).dom, rdU U f k x = 0) : f = 0

theorem **domInj_of_indep** (U : Unf Xo c σ r) (h : RdIndep1 U) : RdDomInj1 U

theorem **domInj_iff** (U : Unf Xo c σ r) : RdDomInj1 U ↔ ∀ a n, Dom a n → ∀ f ∈ sec Xo c a n, rdU U f = 0 → f = 0

theorem **sec_eq_zero_of_isEmpty** (U : Unf Xo c σ r) [IsEmpty U.Comp] (h : RdDomInj1 U) (a n) (hd : Dom a n) (f) (hf : f ∈ sec Xo c a n) : f = 0

theorem **isUnf_copy** (U : Unf Xo c σ r) : Formalism.L2.PjFree.IsUnf (copy U) ↔ IsUnf U

theorem **indep_of_isUnf** (hS : Out.SecBaseChangeSt) (U : Unf Xo c σ r) (hU : IsUnf U) : RdIndep1 U ∧ RdDomInj1 U

theorem **invFun_relab_injective** (c : Coeff l) (σ : c.F →+* ℂ) : Function.Injective (Function.invFun (relab (E

theorem **conjEmb_inj** {τ τ' : E →+* ℂ} (h : conjEmb τ = conjEmb τ') : τ = τ'

theorem **dom_step** (ρ : E →+* ℂ) (m : ℕ) : Dom (step ρ m) 0

theorem **chi_eq_one** (U : Unf Xo c σ r) (h : ∀ a n, Dom a n → ∀ k, ∀ γ ∈ (U.X k).Γ, (U.X k).χ (laU U a n k) γ = 1) (k : U.Comp) (la : Wt (U.X k).Ψ r) (γ) (hγ : γ ∈ (U.X k).Γ) : (U.X k).χ la γ = 1

theorem **adelAut_of_dominant** (hχ : ∀ a n, Dom a n → ∀ k, ∀ γ ∈ (U.X k).Γ, (U.X k).χ (laU U a n k) γ = 1) (hd : DictIsUnif V dict Kf th hr) (hm : MuUnit V dict mu) (hc : Continuous mu) (hL : ∀ g₀ : GA D V, ∃ N ∈ 𝒩 g₀, ∃ (k : U.Comp) (x : GA D V → Frame (U.X k).Ψ r), ContinuousOn x N ∧ ∀ g ∈ N, x g ∈ (U.X k).dom ∧ ∃ γ ∈ (U.X k).Γ, dict g = (k, lact γ (x g))) : AdelAut V dict mu

theorem **meas_of_aut** (h : AdelAut V dict mu) : AdelMeas1 U V dict mu

theorem **aeZero_of_aut** (hA : AdelAut V dict mu) (hQ : QuotSupp V) (hmu : MuUnit V dict mu) (hd : DictIsUnif V dict Kf th hr) : AeDom1 U V dict mu

theorem **four_laws** (hS : Out.SecBaseChangeSt) (hU : IsUnf U) (hχ : ∀ a n, Dom a n → ∀ k, ∀ γ ∈ (U.X k).Γ, (U.X k).χ (laU U a n k) γ = 1) (hd : DictIsUnif V dict Kf th hr) (hm : MuUnit V dict mu) (hc : Continuous mu) (hL : ∀ g₀ : GA D V, ∃ N ∈ 𝒩 g₀, ∃ (k : U.Comp) (x : GA D V → Frame (U.X k).Ψ r), ContinuousOn x N ∧ ∀ g ∈ N, x g ∈ (U.X k).dom ∧ ∃ γ ∈ (U.X k).Γ, dict g = (k, lact γ (x g))) (hT : VolTieG D) (hs : V.SigOK) : RdIndep1 U ∧ RdDomInj1 U ∧ AdelMeas1 U V dict mu ∧ AeDom1 U V dict mu

theorem **meas_zero** : AdelMeas1 U V dict 0

theorem **aeZero_zero_iff** : AeDom1 U V dict 0 ↔ ∀ la, ∀ P ∈ holSub U.X la, ∀ k, ∀ x ∈ (U.X k).dom, P k x = 0

theorem **meas_off_dom** (h : AdelMeas1 U V dict mu) (P : FDSec U.X) (hP : ∀ k, ∀ x ∈ (U.X k).dom, P k x = 0) : AEStronglyMeasurable (onQ V (adel V dict mu P)) (muQ1 V)

theorem **adel_eq_zero_of_off_dom** (hd : ∀ g, (dict g).2 ∈ (U.X (dict g).1).dom) (P : FDSec U.X) (hP : ∀ k, ∀ x ∈ (U.X k).dom, P k x = 0) : adel V dict mu P = 0

theorem **zero_of_dict_off_dom** (hd : ∀ g, (dict g).2 ∉ (U.X (dict g).1).dom) (h : AeDom1 U V dict mu) (la) (f : Rul Xo c) (hf : rdU U f ∈ holSub U.X la) : rdU U f = 0

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.AdmissibleSmulIffOfCoprime

theorem **_root_.GlobalTheory.LevelCoverDatum.pd_pos** (d : _root_.Formalism.L2.Pj.DeltaDat Xo c n) (h : d.IsLev) : 0 < d.pd

theorem **Mb_ne_bot** (l : Level H) : l.Mb ≠ ⊥

theorem **_root_.GlobalTheory.LevelCoverDatum.half_mul_conj** (d : _root_.Formalism.L2.Pj.DeltaDat Xo c n) (h : d.IsLev) : d.I * d.I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(d.pd : 𝒪 E)}

theorem **half_unique** (d d' : DeltaDat Xo c n) (h : d.IsLev) (h' : d'.IsLev) (e : d.l₂ = d'.l₂) : d.I = d'.I ∧ d.pd = d'.pd

theorem **isLev_change** (d : DeltaDat Xo c n) (h : d.IsLev) (φ : (Fin n → LevQ Dw d.l₂) → c.F) (nab : NabV d.Xo₂ d.c₂) : DeltaDat.IsLev { d with φδ

theorem **_root_.GlobalTheory.LevelCoverDatum.isLev_smul** (d : _root_.Formalism.L2.Pj.DeltaDat Xo c n) (h : d.IsLev) (a : d.c₂.F) (ha : a ≠ 0) : _root_.Formalism.L2.Pj.DeltaDat.IsLev { d with cδ

theorem **_root_.GlobalTheory.LevelCoverDatum.zetaM_cover** (d : _root_.Formalism.L2.Pj.DeltaDat Xo c n) (h : d.l₂.M = l.M * d.pd) : _root_.Formalism.L2.Pj.zetaM d.c₂ ^ d.pd = d.eF (_root_.Formalism.L2.Pj.zetaM c)

theorem **_root_.GlobalTheory.LevelCoverDatum.noDelta_w** (nab : _root_.Formalism.L2.Pj.NabV Xo c) (m : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.SABMap Xo.univ.t Xo.univ.t (𝟙 Xo.X)) (x0 : Fin n → Dw.La) : (_root_.Formalism.L2.Pj.noDelta Xo c nab m n).w x0 = _root_.Formalism.L2.Pj.x0Dat c x0

theorem **_root_.GlobalTheory.LevelCoverDatum.noDelta_etaOK** (nab : _root_.Formalism.L2.Pj.NabV Xo c) (m : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.SABMap Xo.univ.t Xo.univ.t (𝟙 Xo.X)) : (_root_.Formalism.L2.Pj.noDelta Xo c nab m n).EtaOK ↔ (∀ a, (Xo.univ.eta a).left ≫ m.φ = (Xo.univ.eta a).left) ∧ ∀ a, (Xo.univ.etaA a).left ≫ m.φ = (Xo.univ.etaA a).left

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.ArchimedeanGluing

theorem **gam_iff** {D : D7} (T : Core D) : Gam T ↔ GlobalTheory.RunIn.GamAt T

theorem **root_eq_one_of_glue** {D : D7} (V : HSp D) (hglue : ArchGlue V) {r : ℕ} (hr : V.rank = r) (hr0 : 0 < r) {Comp : Type} (X : Comp → Data D.host.E r) (dict : GA D V → Σ c, Frame (X c).Ψ r) (mu : GA D V → ℂ) (Kf : Subgroup (GA D V)) (th : ∀ c, (X c).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (hth : ∀ k τ, Formalism.MainNotions.StandardLattice.IsInfEmb τ.1 (th k τ)) (hD : DictIsUnif V dict Kf th

hr) (hmu : MuUnit V dict mu) (M : ℕ) (hM : 3 ≤ M) (hK : IsLevelK V M Kf) (k : Comp) (γ γ' : GL (Fin r) D.host.E) (hγ' : γ' ∈ (X k).Γ) (hc : ∀ x ∈ (X k).Λ, ∃ y ∈ (X k).Λ, (γ : Matrix (Fin r) (Fin r) D.host.E) *ᵥ x - x = (M : D.host.E) • y) (u : D.host.E) (n : ℕ) (hn : 0 < n) (hu : u ^ n = 1) (he : (γ' : Matrix (Fin r) (Fin r) D.host.E) = u • (γ : Matrix (Fin r) (Fin r) D.host.E)) : u = 1

theorem **gam_of_lawsM** {D : D7} (hglue : ∀ V : HSp D, ArchGlue V) (T : Core D) (h : LawsM T) : Gam T

theorem **archGlue** {D : D7} (V : HSp D) : ArchGlue V

theorem **gam_of_laws** {D : D7} (T : Core D) (h : LawsM T) : Gam T

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.BaseClassMove

theorem **latAdm_unit** (l : Level H) (bad n : ℕ) : LatAdm l bad (unitLat E n)

theorem **shortValAt_beta** (nab : NabV Xo c) {n : ℕ} (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (β : c.F) (hβ : β ≠ 0) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S) : ShortValAt Xo c nab w Λ β N P y v ↔ ShortValAt Xo c nab w Λ 1 N P y (algebraMap c.F S β * v)

theorem **isBeta_unit** (n : ℕ) : IsBeta Dw c (unitLat E n) 1

theorem **zetaM_pow** : zetaM c ^ l.M = 1

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.BetaAtClosedPoint

theorem **psiIn_mono** (D : DegPre R) (χ : D.X) {I J : Ideal R} (hIJ : I ≤ J) (h : D.PsiIn χ I) : D.PsiIn χ J

theorem **ordPsi_of_pole** (D : DegPre R) (χ : D.X) (h : ¬ D.PsiIn χ ⊤) : ordPsi D χ = 0

theorem **psiIn_iff** {D : DegPre R} {χ : D.X} {r : R} (hr : PsiCoord D χ r) (J : Ideal R) : D.PsiIn χ J ↔ r ∈ J

theorem **iSup_pow_mem** {ϖ : R} (hϖ : Irreducible ϖ) (u : Rˣ) (n : ℕ) : (⨆ (k : ℕ) (_ : (u : R) * ϖ ^ n ∈ maximalIdeal R ^ k), (k : ℕ∞)) = n

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.BetaIsPrincipal

theorem **ln_std** (n : ℕ) : ((stdLat E n) : GenLat E n).Ln = stdLat E n

theorem **std_one** (n : ℕ) (β : c.F) (hβ : β = 1 ∨ β = -1) : β ≠ 0 ∧ BetaIs (Dw

theorem **betaH_congr** {n m : ℕ} (Λ : GenLat E n) (Λ' : GenLat E m) (h : ∀ β : c.F, (β ≠ 0 ∧ BetaIs (Dw

theorem **betaH_ne_zero** {n : ℕ} (Λ : GenLat E n) : betaH (Dw

theorem **detSet_scale** {n : ℕ} (M M' : Submodule (𝒪 E) (Fin n → E)) (a : E) (h : ∀ v, v ∈ M' ↔ ∃ w ∈ M, v = a • w) : detSet M' = (fun d => a ^ n * d) '' detSet M

theorem **betaIs_principal** {n : ℕ} (Λ : GenLat E n) (a : E) (h : ∀ v, v ∈ Λ.Ln ↔ ∃ w ∈ Λ.La, v = a • w) : BetaIs (Dw

theorem **betaIs_bot** {n : ℕ} (hn : 0 < n) (τ₀ : E →+* ℂ) (h0 : Dw.mult τ₀ ≠ 0) (β : c.F) : BetaIs (Dw

theorem **not_unimod_bot** {n : ℕ} (hn : 0 < n) : ¬ ((⊥) : GenLat E n).Unimod

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.CarriedGeneratorUnique

theorem **pinned_isIso** {S : Type} [CommRing S] {ρ : Xo.X ⟶ Xo.X} (ν : SABMap Xo.univ.t Xo.univ.t ρ) (x : Sp S ⟶ Xo.X) (pm : Gx x ⟶ Gx (x ≫ ρ)) (hf : pm ≫ (fst (Over.mk (x ≫ ρ)) Xo.univ.t.G).left = (fst (Over.mk x) Xo.univ.t.G).left) (hs : pm ≫ (snd (Over.mk (x ≫ ρ)) Xo.univ.t.G).left = (snd (Over.mk x) Xo.univ.t.G).left ≫ ν.φ) : IsIso pm

theorem **carried_generator_unique** {S : Type} [CommRing S] {ρ : Xo.X ⟶ Xo.X} (ν : SABMap Xo.univ.t Xo.univ.t ρ) (x : Sp S ⟶ Xo.X) (pm : Gx x ⟶ Gx (x ≫ ρ)) (hf : pm ≫ (fst (Over.mk (x ≫ ρ)) Xo.univ.t.G).left = (fst (Over.mk x) Xo.univ.t.G).left) (hs : pm ≫ (snd (Over.mk (x ≫ ρ)) Xo.univ.t.G).left = (snd (Over.mk x) Xo.univ.t.G).left ≫ ν.φ) (s' s'' : LB.Sec (PPx (x ≫ ρ))) (h : ∀ (i j : Xo.univ.t.PP.L.ι) (W : (Gx x).Opens), W = (snd (Over.mk x) Xo.univ.t.G).left ⁻¹ᵁ ((LB.div (Xo.univ.t.PP.L.pull ν.cm.F) Xo.univ.t.PP.L).U ((fun _ => i), j)) → ∀ (h₁ : W ≤ pm ⁻¹ᵁ (PPx (x ≫ ρ)).U (fun _ => i)), (pm.appLE ((PPx (x ≫ ρ)).U (fun _ => i)) W h₁).hom (s'.f (fun _ => i)) = (pm.appLE ((PPx (x ≫ ρ)).U (fun _ => i)) W h₁).hom (s''.f (fun _ => i))) (i : Xo.univ.t.PP.L.ι) : s'.f (fun _ => i) = s''.f (fun _ => i)

theorem **frMove_unique** {S : Type} [CommRing S] [Algebra c.F S] {ρ : Xo.X ⟶ Xo.X} (ν : SABMap Xo.univ.t Xo.univ.t ρ) (hw : ∀ w w', CubIso ν w → CubIso ν w' → w = w') (y y' y'' : FrPt Xo c S) (h' : FrMove ρ ν y y') (h'' : FrMove ρ ν y y'') : y' = y''

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.ChosenEqOfAllRules

theorem **ptNorm_conj** (σ : K →+* ℂ) (fn : pts (CAt σ) → ι → ℝ) (fn' : pts (CAt ((starRingEnd ℂ).comp σ)) → ι → ℝ) (e : ι → ℕ) (f : Rules pts) (y : pts (CAt σ)) (z : pts (CAt ((starRingEnd ℂ).comp σ))) (hv : evalAt pts mapR f z = conjAt σ (evalAt pts mapR f y)) (hn : ∀ i, |fn' z i| = |fn y i|) : ptNorm pts mapR _ fn' e f z = ptNorm pts mapR σ fn e f y

theorem **bddAt_of_fnorm_zero** (σ : K →+* ℂ) (e : ι → ℕ) (i : ι) (hi : e i ≠ 0) (f : Rules pts) : BddAt pts mapR σ (fun _ _ => 0) e f ∧ supAt pts mapR σ (fun _ _ => 0) e f = 0

theorem **exists_fnorm_small** (σ : K →+* ℂ) (e : ι → ℕ) (i : ι) (hi : e i ≠ 0) (f : Rules pts) : ∃ fnorm : pts (CAt σ) → ι → ℝ, (∀ y j, 0 < fnorm y j) ∧ BddAt pts mapR σ fnorm e f ∧ supAt pts mapR σ fnorm e f ≤ 1

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.CoefFnEqZeroOfNotPos

theorem **reading_mem_SS** (u : WeightFamilies.Wt D j) (F : (recB h.toLawsCore).Cusp j u) (k : (T.U j T.σ₀).Comp) : rdR h j F k ∈ ((T.U j T.σ₀).X k).SS (laU (T.U j T.σ₀) (wtA u) u.n k)

theorem **tieRead_iff** : TieReadSt h j ↔ ∀ u : WeightFamilies.Wt D j, JInjRd T j (wtA u) u.n

theorem **tieRead_holds** (hj : InR D j) : TieReadSt h j

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.DetNormOneGen

theorem **latHat_eq** (V : HSp D) : latHat V = GlobalTheory.AdLat.hat D.host.F (latCo V : Set (Fin V.rank → D.host.E))

theorem **latCo_fg** (V : HSp D) : (latCo V).FG

theorem **latCo_span** (V : HSp D) : Submodule.span D.host.E (latCo V : Set (Fin V.rank → D.host.E)) = ⊤

theorem **gram_det_ne_zero** (V : HSp D) : V.gram.det ≠ 0

theorem **det_norm_one_gen** {A B : Type} [CommRing A] [CommRing B] {n : ℕ} (σ : A →+* A) (τ : B → B) (φ : A →+* B) (hφ : ∀ x, τ (φ x) = φ (σ x)) (J g : Matrix (Fin n) (Fin n) A) (hJ : IsUnit J.det) (h : Formalism.UG.cT σ g * J * g = J) : (g.map φ).det * τ (g.map φ).det = 1

theorem **sg_fin** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (c : E ≃ₐ[F] E) (x : AF F ⊗[F] E) : GlobalTheory.AdLat.sg (Af F) c ((Algebra.TensorProduct.map (sndA F) (AlgHom.id F E)).toRingHom x) = (Algebra.TensorProduct.map (sndA F) (AlgHom.id F E)).toRingHom (sigR (AF F) (c : E →ₐ[F] E) x)

theorem **isUnit_det_JR** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (h : J.det ≠ 0) : IsUnit (JR (F

theorem **det_norm_one** (V : HSp D) (k : GA D V) : (finHom V k).1.det * GlobalTheory.AdLat.sg (Af D.host.F) (conjEquiv D) (finHom V k).1.det = 1

theorem **mulVec_mem** {V : HSp D} {N : ℕ} {k : GA D V} (hk : k ∈ levSet V N) : ∀ z ∈ latHat V, (finHom V k).1 *ᵥ z ∈ latHat V

theorem **inv_mulVec_mem** {V : HSp D} {k : GA D V} (hk : ∀ z ∈ latHat V, (finHom V k).1 *ᵥ z ∈ latHat V) : ∀ z ∈ latHat V, (finHom V k⁻¹).1 *ᵥ z ∈ latHat V

theorem **finSign_sq** (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] : finSign F E * finSign F E = 1

theorem **finSignGL_mem** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] {n : ℕ} (σ : E →ₐ[F] E) (J : Matrix (Fin n) (Fin n) E) : finSignGL F E n ∈ HR (AF F) σ J

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.ExistsBaseRingOfConjugationFixedTensor

theorem **exists_baseRing_of_conjugation_fixed_tensor** {F E R : Type} [Field F] [CharZero F] [Field E] [Algebra F E] [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (htr : ∀ x, ∃ t, algebraMap F E t = x + σ x) (y : R ⊗[F] E) (hy : sigR R σ y = y) : ∃ a : R, algebraMap R (R ⊗[F] E) a = y

theorem **sigR_algebraMap** {F E R : Type} [Field F] [Field E] [Algebra F E] [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (a : R) : sigR R σ (algebraMap R (R ⊗[F] E) a) = algebraMap R (R ⊗[F] E) a

theorem **det_bc** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) {n : ℕ} (g : GL (Fin n) A) : (Formalism.UG.bc φ g).1.det = φ g.1.det

theorem **inclR_injective** {F E : Type} [Field F] [Field E] [Algebra F E] (R : Type) [CommRing R] [Algebra F R] [Nontrivial R] : Function.Injective (inclR (F

theorem **fixed_mem_range** (y : AE D) (hy : cA D y = y) : ∃ a, algebraMap (AF D.host.F) (AE D) a = y

theorem **fixed_iff_idF** (y : (AE D)ˣ) : cA D (y : AE D) = y ↔ ∃ a, idF D a = y

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.FlatGenInjective

theorem **isIntPS_iff** (T : CDVR) {n d : ℕ} (f : Germ.PS T.Fr n d) : IsIntPS T f ↔ ∃ g : Germ.PS T.V n d, f = MvPowerSeries.map (algebraMap T.V T.Fr) g

theorem **flatGen_injective** {X X' Y : Scheme.{0}} (L : LB X) (F : X' ⟶ X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X')) (ε : K → ℤ) (a b : LB.Gen (LB.LinComb L (fun k => φ k ≫ F) ε)) (h : flatGen L F φ ε a = flatGen L F φ ε b) : a = b

theorem **series_integral** (C : IntegralChartA3 Xo c T io y) : letI

theorem **const_zero** (C : IntegralChartA3 Xo c T io y) : (∀ j, MvPowerSeries.constantCoeff (C.FT j) = 0) ∧ ∀ a j, MvPowerSeries.constantCoeff (C.actT a j) = 0

theorem **dim_eq** (C : IntegralChartA3 Xo c T io y) : chD C.ch = Fintype.card (Lt1 Dw)

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.FrBorDiagUnique

theorem **frBor_diag_unique** {S : Type} [CommRing S] [Algebra c.F S] (t t' : Ix Dw → Sˣ) (y y' : FrPt Xo c S) (h : FrBor t y y') (h' : FrBor t' y y') : t = t'

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.FullIffBijective

theorem **zeta_prim** {n : ℕ} (hn : n ≠ 0) : IsPrimitiveRoot (zeta n) n

theorem **zeta_mem_of_dvd** {F : IntermediateField ℚ ℂ} {m n : ℕ} (hn : n ≠ 0) (hmn : m ∣ n) (h : zeta n ∈ F) : zeta m ∈ F

theorem **zeta_pd_mem** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} (d : DeltaDat Xo c n) (h : d.IsLev) : zeta d.pd ∈ c.F

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.InducedEqOne

theorem **noDelta_phiPin** (nab : NabV Xo c) (m : SABMap Xo.univ.t Xo.univ.t (𝟙 Xo.X)) : (noDelta Xo c nab m n).PhiPin (fun _ => 1) 1 (fun _ => 0)

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.IsFractionRingIntO

theorem **embHom_apply** (P : PtC Y C) (τ : E →+* ℂ) (a : 𝒪 E) : embHom P τ a = P.emb τ a

theorem **lieMult** (P : PtC Y C) (τ : E →+* ℂ) : LieMult Y.t.act P.x0 (embHom P τ) (D.mult τ)

theorem **nonempty_flag** (P : PtC Y C) : Nonempty P.Flag

theorem **ext_of_generic** (P P' : PtC Y C) (h : P.y = P'.y) : P = P'

theorem **valuationRing_intO** (K : Type) [NormedField K] [IsUltrametricDist K] : ValuationRing (intO K)

theorem **isFractionRing_intO** (K : Type) [NormedField K] [IsUltrametricDist K] : IsFractionRing (intO K) K

theorem **isEmpty_of_isUnit** (hp : 2 ≤ p) (h : IsUnit (p : O)) : IsEmpty (PtC Y C)

theorem **main** (P : PtC Y C) : Nonempty P.Flag ∧ (∀ P' : PtC Y C, P.y = P'.y → P = P') ∧ (2 ≤ p → ¬ IsUnit (p : O))

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.JetTruncNotInjective

theorem **orig_eq** : orig R ι = MvPolynomial.idealOfVars ι R

theorem **mk_eq_zero_iff** (p : MvPolynomial ι R) : (Ideal.Quotient.mk _ p : Jet R ι m) = 0 ↔ ∀ e : ι →₀ ℕ, e.degree ≤ m → p.coeff e = 0

theorem **X_pow_ne_zero** [Nontrivial R] (i : ι) : (Ideal.Quotient.mk _ (X i ^ m) : Jet R ι m) ≠ 0

theorem **X_pow_succ_eq_zero** (i : ι) : (Ideal.Quotient.mk _ (X i ^ (m + 1)) : Jet R ι m) = 0

theorem **mem_degLE** [Fintype ι] [DecidableEq ι] (e : ι →₀ ℕ) : e ∈ degLE m ↔ e.degree ≤ m

theorem **coeff_psTrunc** [Fintype ι] [DecidableEq ι] (f : MvPowerSeries ι R) (e : ι →₀ ℕ) : (psTrunc m f).coeff e = if e.degree ≤ m then MvPowerSeries.coeff e f else 0

theorem **psJet_eq_iff** [Fintype ι] [DecidableEq ι] (f g : MvPowerSeries ι R) : psJet m f = psJet m g ↔ ∀ e : ι →₀ ℕ, e.degree ≤ m → MvPowerSeries.coeff e f = MvPowerSeries.coeff e g

theorem **jetMap_mk** {κ : Type} (φ : ι → MvPolynomial κ R) (hφ : ∀ i, φ i ∈ orig R κ) (p : MvPolynomial ι R) : (jetMap (m

theorem **jetAug_mk** (p : MvPolynomial ι R) : (jetAug : Jet R ι m →ₐ[R] R) (Ideal.Quotient.mk _ p) = constantCoeff p

theorem **jetTan_X** [DecidableEq ι] (l i : ι) : (jetTan l : Jet R ι (m + 1) →ₐ[R] DualNumber R) (Ideal.Quotient.mk _ (X i)) = if i = l then DualNumber.eps else 0

theorem **jetScale_X** (c : ι → R) (i : ι) : (jetScale c : Jet R ι m →ₐ[R] Jet R ι m) (Ideal.Quotient.mk _ (X i)) = c i • (Ideal.Quotient.mk _ (X i) : Jet R ι m)

theorem **jetTrunc_surjective** : Function.Surjective (jetTrunc : Jet R ι (m + 1) →ₐ[R] Jet R ι m)

theorem **jetTrunc_not_injective** [Nontrivial R] (i : ι) : ¬ Function.Injective (jetTrunc : Jet R ι (m + 1) →ₐ[R] Jet R ι m)

theorem **jetSubstPS_add_C** {κ : Type} [Fintype κ] [DecidableEq κ] (φ : ι → MvPowerSeries κ R) (c : ι → R) : (jetSubstPS (m

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.KerMemAbLocus

theorem **ker_mem_abLocus** (A : DatAt Y P) (q : Sp ↥C.S) : spMap (fC A) q ∈ A.ι ⁻¹' abLocus Y.t.G

theorem **isOpen_locus** (A : DatAt Y P) : IsOpen (A.ι ⁻¹' abLocus Y.t.G)

theorem **coord_ne_zero** (A : DatAt Y P) (χ : A.Ylat) (r : A.R) (hr : PsiCoord A.U.D.toPre χ r) : fC A r ≠ 0

theorem **norm_le** (A : DatAt Y P) (a : A.R) : ‖fC A a‖ ≤ 1

theorem **vPsiAt_eq** (A : DatAt Y P) (χ : A.Ylat) : (∃ r, PsiCoord A.U.D.toPre χ r) ∧ ∀ r, PsiCoord A.U.D.toPre χ r → vPsiAt A.U.D.toPre (fC A) p χ = -Real.logb p ‖fC A r‖

theorem **natCast_mem** (hp : 1 < p) (A : DatAt Y P) : (p : A.R) ∈ maximalIdeal A.R

theorem **betaAdd** (Y : Model Xo c β) (p : ℕ) : BetaAdd Y p

theorem **beta_balanced** (A : DatAt Y P) (x : 𝒪 E) (a b : A.Ylat) : A.beta (x • a) b = A.beta a (H.bar x • b)

theorem **beta_self_pos** (hp : 1 < p) (A : DatAt Y P) (a : A.Ylat) (ha : a ≠ 0) : 0 < A.beta a a

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.LevelGroupEqTopOfCoprime

theorem **natCast_mem_half** (l : Level H) : (l.M : 𝒪 E) ∈ l.Mb

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.LocClauseOfUnifPlaces

theorem **mem_dom_iff** (X : Data E r) {γ : GL (Fin r) E} (hγ : γ ∈ X.Γ) (y : Frame X.Ψ r) : lact γ y ∈ X.dom ↔ y ∈ X.dom

theorem **lact_rmul_lact** {Ψ : Finset (E →+* ℂ)} (γ' γ : GL (Fin r) E) (y b : Frame Ψ r) : lact γ' (rmul (lact γ y) b) = lact (γ' * γ) (rmul y b)

theorem **lact_one** {Ψ : Finset (E →+* ℂ)} (y : Frame Ψ r) : lact (1 : GL (Fin r) E) y = y

theorem **locClause_of_unif** {V : HSp D} {X : Comp → Data E r} {dict : GA D V → Σ k, Frame (X k).Ψ r} {Kf : Subgroup (GA D V)} {th : ∀ c, (X c).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)} {hr : V.rank = r} (hd : DictIsUnif V dict Kf th hr) (hK : Kf ≤ GfC V) (hN : ∀ g₀ : GA D V, {g : GA D V | ∃ k ∈ Kf, ∃ h ∈ GinfC V, g = g₀ * k * h} ∈ 𝒩 g₀) (hth : ∀ c τ, Continuous (archM V (th c τ))) : LocClause V dict

theorem **continuous_place** {F L : Type} [Field F] [NumberField F] [Field L] [NumberField L] [Algebra F L] {τ : L →+* ℂ} {θ : Ainf F ⊗[F] L →+* ℂ} (hθ : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ) : Continuous fun z : AF F ⊗[F] L => θ (Algebra.TensorProduct.map (fstA F) (AlgHom.id F L) z)

theorem **continuous_archM** (V : HSp D) {τ : D.host.E →+* ℂ} {θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ} (hθ : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ) : Continuous (archM V θ)

theorem **locClause_of_unif_places** {V : HSp D} {X : Comp → Data E r} {dict : GA D V → Σ k, Frame (X k).Ψ r} {Kf : Subgroup (GA D V)} {th : ∀ c, (X c).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)} {hr : V.rank = r} (hd : DictIsUnif V dict Kf th hr) (hK : Kf ≤ GfC V) (hN : ∀ g₀ : GA D V, {g : GA D V | ∃ k ∈ Kf, ∃ h ∈ GinfC V, g = g₀ * k * h} ∈ 𝒩 g₀) (hth : ∀ c τ, ∃ τ', Formalism.MainNotions.StandardLattice.IsInfEmb τ' (th c τ)) : LocClause V dict

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.OrthonormalBaseFrameNeg

theorem **isInfEmb_unique** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (τ : E →+* ℂ) (θ θ' : Ainf F ⊗[F] E →+* ℂ) (h : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ) (h' : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ') : θ = θ'

theorem **xiR_pos** (D : D7) : 0 < xiR D

theorem **lminor_sgn** (p0 : ℕ) (i : Fin (r + 1)) : lminor (Matrix.diagonal fun a : Fin r => ((sgn p0 (a : ℕ) : ℝ) : ℂ)) i = ((psgn p0 i : ℝ) : ℂ)

theorem **psgn_sq** (p0 i : ℕ) : psgn p0 i * psgn p0 i = 1

theorem **gram_eq** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (P : Matrix (Fin r) (Fin r) ℂ) : gram τ G P = P.transpose * (G.map τ * P.map (starRingEnd ℂ))

theorem **unit_of_x0Orth** (X : Data E r) (x₀ : (E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (h : IsOrthogonalFrame X x₀) : (fun τ : X.Ψ => (x₀ τ.1 : Fin r → Fin r → ℂ)) ∈ X.dom ∧ ∀ τ i, ‖dmin X.Gm (fun τ : X.Ψ => (x₀ τ.1 : Fin r → Fin r → ℂ)) τ i‖ = 1

theorem **relab_val** (c : Coeff l) (τ : E →+* ℂ) : relab c c.F.val.toRingHom τ = τ

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.PEqQOfNeg

theorem **_root_.GlobalTheory.DatumSign.trace_eq** : ∀ {V : Type} [AddCommGroup V] [Module ℝ V] [FiniteDimensional ℝ V] (T : V →ₗ[ℝ] V) (s : ℝ), ∀ (hs : (0 : ℝ) < s) (hT : ∀ (x : V), T (T x) = (s * s) • x), LinearMap.trace ℝ V T = s * ((2 : ℝ) * (Module.finrank ℝ (T - s • LinearMap.id).ker : ℝ) - (Module.finrank ℝ V : ℝ))

theorem **finrank_le** (B : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (A J K : V →ₗ[ℝ] V) (s : ℝ) (hs : 0 < s) (hA : ∀ x, A (A x) = -((s * s) • x)) (hJ : ∀ x, J (J x) = -x) (hK : ∀ x, K (K x) = -x) (hKA : K ∘ₗ A = A ∘ₗ K) (pJ : ∀ x, x ≠ 0 → 0 < B (J x) x) (pK : ∀ x, x ≠ 0 → 0 < B (K x) x) : finrank ℝ

(LinearMap.ker (J ∘ₗ A - s • LinearMap.id)) ≤ finrank ℝ (LinearMap.ker (K ∘ₗ A - s • LinearMap.id))

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.PiTwistNegation

theorem **halfI_ne_top** (T : Core D) (i : Fin 2) : halfI T i ≠ ⊤

theorem **bar_piOf_mem_K** (T : Core D) (X : ErrX D) (i : Fin 2) : ((barE D (piOf T X i) : 𝒪 D.host.E) : D.host.E) ∈ D.host.K

theorem **halfEX_iff_le** (T : Core D) (h : LawsM T) (X : ErrX D) : HalfEX T X ↔ ∀ i, T.Q.l.Mb ≤ Ideal.span {barE D (X.piG i)}

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.PowSqEqOneOfKilledBy

theorem **zCard_pos** (Dw : Datum H) {n : ℕ} (Λ : GenLat E n) (N M : ℕ) (hU : Λ.Unimod) (hL : Λ.LevOK N M) : 0 < Λ.zCard N (Dw.p + Dw.q)

theorem **pow_sq_eq_one_of_killedBy** {T : Over Xo.X} (t : T ⟶ Xo.univ.t.G) (h : KilledBy Xo (l.Mb ^ 2) t) : t ^ (l.M ^ 2) = 1

theorem **geoExhaust_of_field** {n : ℕ} (Λ : GenLat E n) (N : ℕ) {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G)) (hf : ∀ (k : Type) [Field k] (x : Sp k ⟶ Xo.X) (w : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x ⟶ Xo.univ.t.G)), (∀ i, InZh Xo Λ N (w i)) → GeoInj Xo x w → ∀ a, InZh Xo Λ N a → ∃ i, w i = a) (hz : ∀ i, InZh Xo Λ N (z i)) (hg : GeoInj Xo x0 z) : GeoExhaust Xo Λ N x0 z

theorem **zSplits_of** (hL : Laws) (hs : ∀ (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.LevOK N l.M → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ (∀ i, InZh Xo Λ N (z i)) ∧ GeoInj Xo (spMap φ ≫ x0) z) (hf : ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.LevOK N l.M → ∀ (k : Type) [Field k] (x : Sp k ⟶ Xo.X) (w : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x ⟶ Xo.univ.t.G)), (∀ i, InZh Xo Λ N (w i)) → GeoInj Xo x w → ∀ a, InZh Xo Λ N a → ∃ i, w i = a) : ZSplits Xo c

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.RelabBijective

theorem **relab_bijective** (c : Coeff l) (σ : c.F →+* ℂ) : Function.Bijective (relab (E

theorem **relab_conj** (c : Coeff l) (σ : c.F →+* ℂ) (τ : E →+* ℂ) : relab c σ (conjEmb τ) = conjEmb (relab c σ τ)

theorem **laU_upper** (U : Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (k : U.Comp) (τ : E →+* ℂ) (hτ : relab c σ τ ∈ (U.X k).Ψ) (i : ℕ) (hi : (U.X k).p0 + i < r) : laU U a n k (_, hτ) (_, hi) = (a (τ, i) : ℤ) + (n / 2 : ℕ)

theorem **laU_lower** (U : Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (k : U.Comp) (τ : E →+* ℂ) (hτ : relab c σ τ ∈ (U.X k).Ψ) (i : ℕ) (hi : i < (U.X k).p0) (hr : (U.X k).p0 - 1 - i < r) : laU U a n k (_, hτ) (_, hr) = -(a (conjEmb τ, i) : ℤ) - (n / 2 : ℕ)

theorem **laU_antitone** (U : Unf Xo c σ r) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (hd : Dom a n) (k : U.Comp) (τ : (U.X k).Ψ) (j j' : Fin r) (hj : j ≤ j') (hb : (U.X k).p0 ≤ (j : ℕ) ∨ (j' : ℕ) < (U.X k).p0) : laU U a n k τ j' ≤ laU U a n k τ j

theorem **laU_add_two** (U : Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : laU U a (n + 2) = laU U (fun x => a x + 1) n

theorem **laU_odd** (U : Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (m : ℕ) : laU U a (2 * m + 1) = laU U a (2 * m)

theorem **conj_mem** (X : Data E r) {τ : E →+* ℂ} (h : τ ∉ X.Ψ) : conjEmb τ ∈ X.Ψ

theorem **perFr_smul** (X : Data E r) (x : Frame X.Ψ r) (b : E) (v : Fin r → E) (τ : E →+* ℂ) (i : ℕ) : perFr X x (b • v) τ i = τ b * perFr X x v τ i

theorem **perFr_beyond** (X : Data E r) (x : Frame X.Ψ r) (v : Fin r → E) {τ : E →+* ℂ} (h : τ ∉ X.Ψ) (hr : 0 < r) {i : ℕ} (hi : X.p0 ≤ i + 1) : perFr X x v τ i = hermC (conjEmb τ) X.Gm (col (mat x (conjEmb τ, conj_mem X h)) (0, hr)) (Data.emb (conjEmb τ) v)

theorem **rdU_mul** (U : Unf Xo c σ r) (f g : Rul Xo c) : rdU U (f * g) = rdU U f * rdU U g

theorem **not_good_conj** (h : GoodEmb c σ) : ¬ GoodEmb c ((starRingEnd ℂ).comp σ)

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.RightInvariantOfCovolume

theorem **_root_.GlobalTheory.AutQuotCheck.cls_eq_iff** (g g' : G) : _root_.GlobalTheory.AutQuotCheck.cls Γ g = _root_.GlobalTheory.AutQuotCheck.cls Γ g' ↔ ∃ γ ∈ Γ, g' = γ * g

theorem **_root_.GlobalTheory.AutQuotCheck.cls_smul** (γ : Γ) (x : G) : _root_.GlobalTheory.AutQuotCheck.cls Γ (γ • x) = _root_.GlobalTheory.AutQuotCheck.cls Γ x

theorem **_root_.GlobalTheory.AutQuotCheck.measurable_cls** [MeasurableSpace G] : Measurable (_root_.GlobalTheory.AutQuotCheck.cls Γ)

theorem **_root_.GlobalTheory.AutQuotCheck.map_restrict_eq** [MeasurableSpace G] [Countable Γ] [MeasurableConstSMul Γ G] (μ : Measure G) [SMulInvariantMeasure Γ G μ] {F F' : Set G} (h : IsFundamentalDomain Γ F μ) (h' : IsFundamentalDomain Γ F' μ) : (μ.restrict F).map (_root_.GlobalTheory.AutQuotCheck.cls Γ) = (μ.restrict F').map (_root_.GlobalTheory.AutQuotCheck.cls Γ)

theorem **map_dirac_one_univ_eq_one** [MeasurableSpace G] : (((Measure.dirac (1 : G)).restrict Set.univ).map (cls Γ)) Set.univ = 1

theorem **_root_.GlobalTheory.AutQuotCheck.rightInvariant_of_covolume** (μ : Measure G) [μ.IsHaarMeasure] {F : Set G} (h : IsFundamentalDomain Γ F μ) (h0 : μ F ≠ 0) (h1 : μ F ≠ ⊤) (g : G) : μ.map (· * g) = μ

theorem **normalised_eq** (μ ν : Measure G) [μ.IsHaarMeasure] [ν.IsHaarMeasure] {F F' : Set G} (h : IsFundamentalDomain Γ F μ) (h' : IsFundamentalDomain Γ F' ν) : (μ F)⁻¹ • (μ.restrict F).map (cls Γ) = (ν F')⁻¹ • (ν.restrict F').map (cls Γ)

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.SectionBaseChangeOfBasis

theorem **coord_tmul** (lam : K' →ₗ[K] K) {R : Type} [CommRing R] [Algebra K R] (a : K') (r : R) : coord lam R (a ⊗ₜ r) = lam a • r

theorem **coord_map** (lam : K' →ₗ[K] K) {R T : Type} [CommRing R] [Algebra K R] [CommRing T] [Algebra K T] (ψ : R →ₐ[K] T) (v : K' ⊗[K] R) : coord lam T (Algebra.TensorProduct.map (AlgHom.id K' K') ψ v) = ψ (coord lam R v)

theorem **coord_mul** (lam : K' →ₗ[K] K) {R : Type} [CommRing R] [Algebra K R] (r : R) (v : K' ⊗[K] R) : coord lam R (((1 : K') ⊗ₜ[K] r) * v) = r * coord lam R v

theorem **sum_coord** {d : ℕ} (B : Module.Basis (Fin d) K K') {R : Type} [CommRing R] [Algebra K R] (v : K' ⊗[K] R) : v = ∑ j, B j ⊗ₜ[K] coord (B.coord j) R v

theorem **ext_tower** (x : K) : algebraMap K' (Ext K K' R) (algebraMap K K' x) = algebraMap K (Ext K K' R) x

theorem **ext_reduced** [CharZero K] (p : TRing K) : IsReduced (Ext K K' p.R)

theorem **exists_map** (ψ : R →ₐ[K] T) : ∃ (Ψ : Ext K K' R →ₐ[K'] Ext K K' T) (Ψ₀ : Ext K K' R →ₐ[K] Ext K K' T), (∀ x, Ψ₀ x = Ψ x) ∧ Ψ₀.comp (inR K' R) = (inR K' T).comp ψ ∧ ∀ lam v, crd lam (Ψ v) = ψ (crd lam v)

theorem **crd_mul** (lam : K' →ₗ[K] K) (r : R) (v : Ext K K' R) : crd lam (inR K' R r * v) = r * crd lam v

theorem **exists_lift** {S : Type} [CommRing S] [Algebra K S] [Algebra K' S] (h : ∀ x, algebraMap K' S (algebraMap K K' x) = algebraMap K S x) (φ : R →ₐ[K] S) : ∃ (Φ : Ext K K' R →ₐ[K'] S) (Φ₀ : Ext K K' R →ₐ[K] S), (∀ x, Φ₀ x = Φ x) ∧ Φ₀.comp (inR K' R) = φ ∧ ∀ (d : ℕ) (B : Module.Basis (Fin d) K K') (v : Ext K K' R), Φ v = ∑ j, algebraMap K' S (B j) * φ (crd (B.coord j) v)

theorem **wchar_inR** {ι : Type} [Fintype ι] (e : ι → ℕ) (t : ι → Rˣ) : wchar e (fun i => Units.map ((inR K' R : R →ₐ[K] Ext K K' R) : R →* Ext K K' R) (t i)) = (inR K' R : R →ₐ[K] Ext K K' R) (wchar e t)

theorem **frMap_ext** {S T : Type} [CommRing S] [Algebra c.F S] [Algebra c'.F S] [CommRing T] [Algebra c.F T] [Algebra c'.F T] (hS : AlgOver c c' hle S) (hT : AlgOver c c' hle T) (φ : S →ₐ[c.F] T) (Φ : S →ₐ[c'.F] T) (e : ∀ x, φ x = Φ x) {y : FrPt Xo c S} {z : FrPt Xo c T} (h : FrMap φ y z) : FrMap Φ (y.ext hle hS) (z.ext hle hT)

theorem **frMap_res** {S T : Type} [CommRing S] [Algebra c.F S] [Algebra c'.F S] [CommRing T] [Algebra c.F T] [Algebra c'.F T] (hS : AlgOver c c' hle S) (hT : AlgOver c c' hle T) (φ : S →ₐ[c.F] T) (Φ : S →ₐ[c'.F] T) (e : ∀ x, φ x = Φ x) {y : FrPt Xo c' S} {z : FrPt Xo c' T} (h : FrMap Φ y z) : FrMap φ (y.res hle hS) (z.res hle hT)

theorem **frBor_ext** {S : Type} [CommRing S] [Algebra c.F S] [Algebra c'.F S] (hS : AlgOver c c' hle S) {t : Ix Dw → Sˣ} {y z : FrPt Xo c S} (h : FrBor t y z) : FrBor t (y.ext hle hS) (z.ext hle hS)

theorem **section_base_change_of_basis** [Algebra c.F c'.F] (halg : ∀ x, algebraMap c.F c'.F x = incF c c' hle x) {d : ℕ} (B : Module.Basis (Fin d) c.F c'.F) (hWD : ValueWD Xo c) (hWD' : ValueWD Xo c') (hBC : HasBC Xo c) (hA : AllTest Xo c) (hA' : AllTest Xo c') (hB' : BorChartF Xo c') : Formalism.L2.Pj.SecBC Xo hle

theorem **secBC_of_laws** (hWD : ValueWD Xo c) (hWD' : ValueWD Xo c') (hBC : HasBC Xo c) (hA : AllTest Xo c) (hA' : AllTest Xo c') (hB' : BorChartF Xo c') : Formalism.L2.Pj.SecBC Xo hle

theorem **secBC_holds** : Formalism.L2.Pj.SecBC Xo hle

theorem **cuspBC_holds** {r : ℕ} (U : ∀ σ : c.F →+* ℂ, Formalism.L2.Pj.Unf Xo c σ r) : Formalism.L2.Pj.CuspBC hle U

theorem **val_of_isBC** {m : ℕ} {f' : Rul Xo c'} (hf' : Natural c' f') {as : Fin m → c'.F} {fs : Fin m → Rul Xo c} (h : IsBC hle f' as fs) (q : TRing c'.F) [Algebra c.F q.R] (hq : AlgOver c c' hle q.R) (y : FrPt Xo c' q.R) : f' q y = ∑ i, algebraMap c'.F q.R (as i) * evR Xo c (fs i) (y.res hle hq)

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.StandardLatticeGoodPlace

theorem **mem_std** {n : ℕ} (v : Fin n → E) : v ∈ stdLat E n ↔ ∀ j, v j ∈ (algebraMap (𝒪 E) E).range

private theorem **algebraMap_range_mem_iff_isIntegral** (x : E) : x ∈ (algebraMap (𝒪 E) E).range ↔ IsIntegral ℤ x

theorem **unimod_std** (H : Host E) (n : ℕ) : GenLat.Unimod ((stdLat E n) : GenLat E n)

theorem **trivAt_std** (n k : ℕ) : GenLat.TrivAt ((stdLat E n) : GenLat E n) k

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.SubsingletonOfLeOne

theorem **real_pos_of_int_ge** {ι : Type} [Fintype ι] (M : ι → ι → ℝ) (hs : ∀ i k, M i k = M k i) {c : ℝ} (hc : 0 < c) (hM : ∀ z : ι → ℤ, z ≠ 0 → c ≤ ∑ i, ∑ k, (z i : ℝ) * z k * M i k) (x : ι → ℝ) (hx : x ≠ 0) : 0 < ∑ i, ∑ k, x i * x k * M i k

theorem **exists_radius** {R C : Type} [CommRing R] [IsLocalRing R] [IsNoetherianRing R] [NormedField C] [IsUltrametricDist C] (f : R →+* C) (hle : ∀ a, ‖f a‖ ≤ 1) (hm : ∀ a ∈ IsLocalRing.maximalIdeal R, ‖f a‖ < 1) : ∃ ρ : ℝ, 0 < ρ ∧ ρ < 1 ∧ ∀ a ∈ IsLocalRing.maximalIdeal R, ‖f a‖ ≤ ρ

theorem **right_inverse_local** {R : Type} [CommRing R] [IsLocalRing R] (e : ℕ) {m n : Type} [Fintype m] [DecidableEq m] [Fintype n] (C : Matrix m n R) (hC : ∀ u, u ᵥ* C.map (Ideal.Quotient.mk (maximalIdeal R ^ (e + 1))) = 0 → u = 0) : ∃ D : Matrix n m R, C * D = 1

theorem **matrix_level_local** {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] {D : DegData R (maximalIdeal R)} {P : PolAV (Sp (FractionRing R))} {G : Over (Sp R)} [GrpObj G] [IsCommMonObj G] (U : UnifPre (maximalIdeal R) D P G) {O : Type} [CommRing O] [StarRing O] [Module O D.toPre.X] {oD : D.OAct O} {oP : P.OAct O} {aG : OAct O G} (hl : U.OLinear O oD oP aG.toRingAct) {g r : ℕ} {v : Fin g → (epsR R ⟶ G)} (hv : IsLieBasis v) {ρ : O →+* Module.End R (Fin g → R)} (hρ : ∀ a c, prodPts (ρ a c) v = prodPts c v ≫ aG.act a) (b : Module.Basis (Fin r) ℤ D.toPre.X) (e : ℕ) : ∃ (C : Matrix (Fin r) (Fin g) R) (D' : Matrix (Fin g) (Fin r) R), C * D' = 1 ∧ ∀ a k j, (C * (LinearMap.toMatrix' (ρ a))ᵀ - (Matrix.of fun k j => (b.repr (star a • b j) k : R)) * C) k j ∈ maximalIdeal R ^ (e + 1)

theorem **finrank_le_local** {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] [IsNoetherianRing R] {D : DegData R (maximalIdeal R)} {P : PolAV (Sp (FractionRing R))} {G : Over (Sp R)} [GrpObj G] [IsCommMonObj G] (U : UnifPre (maximalIdeal R) D P G) {O : Type} [CommRing O] [IsDomain O] [StarRing O] [Module.Finite ℤ O] [Module O D.toPre.X] {oD : D.OAct O} {oP : P.OAct O} {aG : OAct O G} (hl : U.OLinear O oD oP aG.toRingAct) {F : Type} [NontriviallyNormedField F] [IsAlgClosed F] [IsUltrametricDist F] (f : R →+* intO F) (hf : ∀ a ∈ maximalIdeal R, ‖((f a : intO F) : F)‖ < 1) {τ : O →+* F} (hτ : Function.Injective τ) {N : ℕ} (hN : LieMult aG (spMap ((intO F).subtype.comp f)) τ N) : Module.finrank O D.toPre.X ≤ N

theorem **coord_ne_zero** {Y : Model Xo c β} {p : ℕ} {C : SCF p} {P : PointsOfFlagModel.PtC Y C} (A : DatAt Y P) (χ : A.Ylat) (r : A.R) (hr : PsiCoord A.U.D.toPre χ r) : ((intO ɪC.S).subtype.comp A.f) r ≠ 0

theorem **betaAdd** (Y : Model Xo c β) (p : ℕ) : Formalism.L2.Pj.BetaAdd Y p

theorem **betaBal** (Y : Model Xo c β) (p : ℕ) : BetaBal Y p (IsCMField.complexConj E)

theorem **exists_lower** {Y : Model Xo c β} {p : ℕ} (hp : 1 < p) {C : SCF p} {P : PointsOfFlagModel.PtC Y C} (A : DatAt Y P) : ∃ κ : ℝ, 0 < κ ∧ ∀ a : A.Ylat, a ≠ 0 → κ ≤ A.beta a a

theorem **betaPosDef** (Y : Model Xo c β) {p : ℕ} (hp : 1 < p) : BetaPosDef Y p

theorem **rank_le** {Y : Model Xo c β} {p : ℕ} {C : SCF p} {P : PointsOfFlagModel.PtC Y C} (A : DatAt Y P) : ∃ ρ : ℕ, Module.finrank ℤ A.Ylat = 2 * (Module.finrank ℚ E / 2) * ρ ∧ ρ ≤ Dw.p ∧ ρ ≤ Dw.q

theorem **ylRank** (Y : Model Xo c β) (p : ℕ) : YlRank Y p (Dw.p + Dw.q) Dw.p ∧ YlRank Y p (Dw.p + Dw.q) Dw.q

theorem **main** (Y : Model Xo c β) {p : ℕ} (hp : 1 < p) : Formalism.L2.Pj.BetaAdd Y p ∧ BetaBal Y p (IsCMField.complexConj E) ∧ BetaPosDef Y p ∧ YlRank Y p (Dw.p + Dw.q) Dw.p

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.TotalDegreeEqOfMemHarmA

theorem **inPsi_conjugate_iff** (τ : E →+* ℂ) : InPsi H (ComplexEmbedding.conjugate τ) ↔ ¬ InPsi H τ

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.TransformIndicator

private theorem **mfr_ne_bot** : _root_.Formalism.L2.Pj.Mfr l ≠ ⊥

theorem **_root_.GlobalTheory.ClassSumProbe.finite_conj_half** : Finite (Dw.La ⧸ (_root_.Formalism.L2.Pj.Mfr l • ⊤ : Submodule (𝒪 E) Dw.La))

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.TransvMemZGam

theorem **emb_cc** (τ : E →+* ℂ) (x : E) : τ (cc x) = conj (τ x)

theorem **cc_kappa** (X : Data E r) : cc X.κ0 = -X.κ0

theorem **pair_real** (X : Data E r) (χ β : Matrix (Fin ρ) (Fin ρ) E) (hχ : IsHermE χ) (hβ : IsSkew β) : cc (Matrix.trace (χ.transpose * β) / (X.κ0 / 2)) = Matrix.trace (χ.transpose * β) / (X.κ0 / 2)

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.VVAlgebraMapNonneg

theorem **vV_algebraMap_nonneg** (a : V) : 0 ≤ vV V (algebraMap V K' a)

theorem **vV_one** : vV V (1 : K') = 0

theorem **vV_pow** {x : K'} (hx : x ≠ 0) (n : ℕ) : vV V (x ^ n) = n * vV V x

theorem **vV_map_unit** (i : K' →+* K'') (hi : ∀ a : V, i (algebraMap V K' a) ∈ Set.range (algebraMap V' K'')) (u : V˟) : vV V' (i (algebraMap V K' u)) = 0

theorem **exists_vV_map** (i : K' →+* K'') (hi : ∀ a : V, i (algebraMap V K' a) ∈ Set.range (algebraMap V' K'')) : ∃ r : ℕ, ∀ x, vV V' (i x) = (r : ℤ) * vV V x

theorem **nval_map_of_overLa** {K : Type} [Field K] [NumberField K] {la : HeightOneSpectrum (𝒪 K)} {T T' : CDVR} {io : K →+* T.Fr} {io' : K →+* T'.Fr} {e e' : ℕ} (ho : OverLa la T.V io e) (ho' : OverLa la T'.V io' e') (i : T.Fr →+* T'.Fr) (hc : ∀ a, i (io a) = io' a) (hi : ∀ a : T.V, i (algebraMap T.V T.Fr a) ∈ Set.range (algebraMap T'.V T'.Fr)) : ∃ r : ℕ, 0 < r ∧ e' = r * e ∧ (∀ x, vV T'.V (i x) = (r : ℤ) * vV T.V x) ∧ ∀ x, nval T'.V e' (i x) = nval T.V e x

theorem **intAll_ramified** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} (h : Formalism.L2.PjFree.IntAll Xo c) (la : HeightOneSpectrum (𝒪 c.F)) (T : CDVR) (io : c.F →+* T.Fr) (e : ℕ) (ho : OverLa la T.V io e) (hf : T.FinRes) (y : letI

Development.AbelianVarietyTheory.Counterexamples.LevelGroupExamples.ZeroNotAdmissibleOfRankPos

theorem **bar_bar** (H : Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

theorem **cSep_spec** (l : Level H) : cSep l ∈ Mfr l ∧ cSep l - 1 ∈ l.Mb

theorem **dvd_EF_of_mem** (x y : Dw.La) {a : 𝒪 E} (ha : a ∈ Ideal.span {(l.M : 𝒪 E)}) : (l.M : ℤ) ∣ Dw.EF x (a • y)

theorem **mul_mem_M** {a b : 𝒪 E} (ha : a ∈ l.Mb) (hb : b ∈ Mfr l) : a * b ∈ Ideal.span {(l.M : 𝒪 E)}

theorem **bar_mem_Mb** {a : 𝒪 E} (ha : a ∈ Mfr l) : H.bar a ∈ l.Mb

theorem **dvd_right** (x : Dw.La) {y : Dw.La} (hy : y ∈ (l.Mb • ⊤ : Submodule (𝒪 E) Dw.La)) : (l.M : ℤ) ∣ Dw.EF x (cSep l • y)

theorem **dvd_left** {x : Dw.La} (y : Dw.La) (hx : x ∈ (Mfr l • ⊤ : Submodule (𝒪 E) Dw.La)) : (l.M : ℤ) ∣ Dw.EF x (cSep l • y)

theorem **pairL_congr** {n : ℕ} (x0 x0' x x' : Fin n → Dw.La) (h0 : ∀ b, x0 b - x0' b ∈ (Mfr l • ⊤ : Submodule (𝒪 E) Dw.La)) (h1 : ∀ b, x b - x' b ∈ (l.Mb • ⊤ : Submodule (𝒪 E) Dw.La)) : (l.M : ℤ) ∣ pairL (l

theorem **zetaM_coe** (c : Coeff l) : ((zetaM c : c.F) : ℂ) = zeta l.M

theorem **zetaM_prim** (c : Coeff l) : IsPrimitiveRoot (zetaM c) l.M

theorem **levRep_mk** (x : Dw.La) : levRep (Submodule.Quotient.mk x : LevQ Dw l) - x ∈ (l.Mb • ⊤ : Submodule (𝒪 E) Dw.La)

theorem **zpow_congr** (c : Coeff l) {a b : ℤ} (h : (l.M : ℤ) ∣ a - b) : zetaM c ^ a = zetaM c ^ b

theorem **levelFourierDatum_mk** (c : Coeff l) {n : ℕ} (x0 x : Fin n → Dw.La) : x0Dat c x0 (fun b => (Submodule.Quotient.mk (x b) : LevQ Dw l)) = zetaM c ^ pairL (l

theorem **exists_eq_smul_of_dvd** (x : Dw.La) (m : ℕ) (hm : m ≠ 0) (h : ∀ y, (m : ℤ) ∣ Dw.EF x y) : ∃ z, x = (m : ℤ) • z

theorem **mem_of_dvd** {x : Dw.La} (h : ∀ y, (l.M : ℤ) ∣ Dw.EF x (cSep l • y)) : x ∈ (Mfr l • ⊤ : Submodule (𝒪 E) Dw.La)

theorem **levelFourierDatum_eq_iff** (c : Coeff l) {n : ℕ} (x0 x0' : Fin n → Dw.La) : x0Dat c x0 = x0Dat c x0' ↔ ∀ b, x0 b - x0' b ∈ (Mfr l • ⊤ : Submodule (𝒪 E) Dw.La)

theorem **finite_LevQ** : Finite (LevQ Dw l)

theorem **Mfr_ne_top** (l : Level H) : Mfr l ≠ ⊤

Development.AbelianVarietyTheory.Counterexamples.MetricPositivity.PosSemidefOfIsMetric

theorem **exists_isMetricAt** (hpq : 1 ≤ Dw.p + Dw.q) (x : Sp ℂ ⟶ Xo.X) (s : LB.Sec (PPx x)) (uu : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left)) (hgen : SecGen x s) (ht : ∀ τ i, IsTanAt x (uu τ i)) : ∃ N Gr, IsMetricAt x s uu N Gr

theorem **exists_isMetric** (hpq : 1 ≤ Dw.p + Dw.q) {σ : c.F →+* ℂ} (y : FrPt Xo c (CAt σ)) : ∃ N Gr, IsMetric y N Gr

theorem **posSemidef_of_isMetric** {σ : c.F →+* ℂ} {y : FrPt Xo c (CAt σ)} {N : ℝ} {Gr : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ} (h : IsMetric y N Gr) : 0 ≤ N ∧ ∀ b, (Gr b).IsHermitian ∧ ∀ a : Fin (Dw.mult b) → ℂ, 0 ≤ (star a ⬝ᵥ (Gr b).mulVec a).re

theorem **pos_of_isMetric** {σ : c.F →+* ℂ} {y : FrPt Xo c (CAt σ)} {N : ℝ} {Gr : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ} (h : IsMetric y N Gr) : 0 < N

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.CubeMapTripleProductUnique

theorem **act_add** (t : SAB H S) (a b : 𝒪 E) : t.act.act (a + b) = lift (t.act.act a) (t.act.act b) ≫ μ[t.G]

theorem **psum_singleton** (G : Over S) [GrpObj G] [IsCommMonObj G] {T : Over S} (x : Fin 3 → (T ⟶ G)) (i : Fin 3) : psum G x {i} = x i

theorem **cubeMap_tripleProduct_unique** {G' : Over S'} {G : Over S} [GrpObj G'] [IsCommMonObj G'] [GrpObj G] [IsCommMonObj G] (m m' : CubeMap G' G) (h : m.F = m'.F) : m.F3 = m'.F3

theorem **sabMap_ext** {t' : SAB H S'} {t : SAB H S} {f : S' ⟶ S} (m m' : SABMap t' t f) (h : m.φ = m'.φ) : m = m'

theorem **cmHostUnit_add_conj_eq_one** (H : Host E) : H.w0 + H.bar H.w0 = 1

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.DeltaIsGreatest

theorem **exists_chart** (χ : u.D.toPre.X) : ∃ j : PUnit → u.D.toPre.P.L.ι, ⊤ ≤ (u.D.toPre.c χ).left ⁻¹ᵁ u.D.toPre.P.L.U (j PUnit.unit)

theorem **psiNorm_unique** (χ : u.D.toPre.X) {r r' : ℝ} (h : u.PsiNorm χ r) (h' : u.PsiNorm χ r') : r = r'

theorem **big_third** (S : ThetaSys o Ω) (hg : 1 ≤ S.g) (h1 : 1 + 2 * (S.g : ℝ) * (2 : ℝ) ^ ((S.g : ℝ) / 2) < S.N) (h2 : 2 * ((S.N ^ S.g * S.g.factorial : ℕ) : ℝ) * ((S.g : ℝ) * (3 / 2 : ℝ) ^ S.g * (2 : ℝ) ^ ((S.g : ℝ) / 2 - 1)) * S.N < (S.r * S.N : ℕ)) : S.Big

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.DetCondOfNotIsSquare

theorem **epsScale_fst_snd** {k : Type} [Field k] (c : k) (x : DualNumber k) : (epsScale c x).fst = x.fst ∧ (epsScale c x).snd = c * x.snd

theorem **lieMult_unique** {k O : Type} [Field k] [CommRing O] {G : Over (Sp k)} [GrpObj G] [IsCommMonObj G] (a : OAct O G) (τ : O →+* k) {n m : ℕ} (hn : LieMult a (𝟙 _) τ n) (hm : LieMult a (𝟙 _) τ m) : n = m

theorem **map_rt_eq_neg** (H : Host E) {k : Type} [Field k] {ρ : k} (hρ : ρ ^ 2 = -(H.d : k)) (τ : 𝒪 E →+* k) (h : τ H.rt ≠ ρ) : τ H.rt = -ρ

theorem **ne_neg** (H : Host E) {k : Type} [Field k] [CharZero k] {ρ : k} (hρ : ρ ^ 2 = -(H.d : k)) : ρ ≠ -ρ

theorem **detCond_of_not_isSquare** {H : Host E} (D : Datum H) {S : Scheme.{0}} {G : Over S} [GrpObj G] (a : OAct (𝒪 E) G) {k : Type} [Field k] (hk : ¬ IsSquare (-(H.d : k))) (t : Sp k ⟶ S) (ρ : k) : DetCond D a t ρ

theorem **nonempty_ringHom** (k : Type) [Field k] [CharZero k] [IsAlgClosed k] : Nonempty (𝒪 E →+* k)

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.DominatesLiftUnique

theorem **j_over** {O : Type} [CommRing O] {β : O →+* c.F} (Y : Model Xo c β) : Y.j ≫ Y.s = pullback.snd _ _ ≫ spMap β

theorem **spMap_comp** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **transition_over** {l₂ : Level H} {Xo₂ : Moduli D l₂} {c₂ : Coeff l₂} {O O₂ : Type} [CommRing O] [CommRing O₂] {β : O →+* c.F} {β₂ : O₂ →+* c₂.F} (Y : Model Xo c β) (Y₂ : Model Xo₂ c₂ β₂) (b : O →+* O₂) (π : Y₂.X ⟶ Y.X) (h : IsTransition Y Y₂ b π) [Mono (spMap β)] (σ : c.F →+* c₂.F) (hσ : σ.comp β = β₂.comp b) : π ≫ Y.s = Y₂.s ≫ spMap b ∧ ∃ r : Xo₂.XF c₂ ⟶ Xo.XF c, Y₂.j ≫ π = r ≫ Y.j ∧ r ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap σ

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.ExistsGoodLevel

theorem **exists_place** (p : ℕ) (hp : p.Prime) : ∃ W : ValuationSubring ℂ, (p : ℂ) ∈ W.nonunits

theorem **exists_incl** {l : Level H} (c : Coeff l) (W : ValuationSubring ℂ) : ∃! b : 𝒪 c.F →+* c.V W, ∀ x, ((b x : c.V W) : c.F) = (x : c.F)

theorem **not_isUnit** {l : Level H} (c : Coeff l) (W : ValuationSubring ℂ) (p : ℕ) (hW : (p : ℂ) ∈ W.nonunits) : ¬ IsUnit (p : c.V W)

theorem **exists_lift** {l : Level H} {Xo : Moduli D l} {c : Coeff l} (W : ValuationSubring ℂ) (Y : Model Xo c (algebraMap (𝒪 c.F) c.F)) (b : 𝒪 c.F →+* c.V W) (hb : ∀ x, ((b x : c.V W) : c.F) = (x : c.F)) : ∃ a : Xo.XF c ⟶ pullback Y.s (spMap b), a ≫ pullback.fst _ _ = Y.j

theorem **goodAt_of_coprime** (l : Level H) (a b : ℕ) (ha : 3 ≤ a) (hb : 3 ≤ b) (hab : a.Coprime b) (hM : l.M = a * b) (p : ℕ) (hp : p.Prime) : l.GoodAt p

theorem **not_goodAt_pow** (l : Level H) (p k : ℕ) (hp : p.Prime) (hM : l.M = p ^ k) : ¬ l.GoodAt p

theorem **exists_good_level** (H : Host E) : ∃ l : Level H, l.A = 1 ∧ ∀ p : ℕ, p.Prime → l.GoodAt p

theorem **exists_prime_eq** {l : Level H} (c : Coeff l) (W : ValuationSubring ℂ) (p : ℕ) (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) [NumberField c.F] : ∃ v : IsDedekindDomain.HeightOneSpectrum (𝒪 c.F), (p : 𝒪 c.F) ∈ v.asIdeal ∧ v.valuationSubringAtPrime c.F = c.V W

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.FinExtHomInjective

theorem **isNsmul_b** {k : ℕ} {g h : GtPtF D F} (H : GtPtF.IsNsmul k g h) : h.b = g.b ^ k

theorem **not_allPre_fe** : ¬ AllPre fun {V} _ _ _ {_} {_} {_} _ _ U => U.FE (FractionRing V) 2

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.HeckeKerZero

theorem **comp_act_natCast** (t : SAB H S) {T : Over S} (x : T ⟶ t.G) (n : ℕ) : x ≫ t.act.act (n : 𝒪 E) = x ^ n

theorem **mem_tors_span** (t : SAB H S) {T : Over S} (x : T ⟶ t.G) (n : ℕ) : x ∈ t.tors T (Ideal.span {(n : 𝒪 E)}) ↔ x ^ n = 1

theorem **heckeKer_zero** {D : Datum H} {l : Level H} {B : Type} [CommRing B] {b : Base l B} {s : S ⟶ Sp B} (o : Obj D l b s) (k : Type) [Field k] [IsAlgClosed k] (x : Sp k ⟶ S) (β : B →+* k) (hx : x ≫ s = spMap β) (P : Ideal (𝒪 E)) (p : ℕ) (hp : 0 < p) : o.t.HeckeKer (Over.mk x) D P p 0 (torsSub o.t (Over.mk x) p)

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.IsQProjOverOfIsProjOver

theorem **_root_.GlobalTheory.SchemeWords.isProper_of_isProjOver** (h : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsProjOver f) : IsProper f

theorem **_root_.GlobalTheory.SchemeWords.isQProjOver_of_isProjOver** (h : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsProjOver f) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsQProjOver f

theorem **finiteType_of_isQProjOver** (h : IsQProjOver f) : LocallyOfFiniteType f ∧ QuasiCompact f ∧ IsSeparated f

theorem **_root_.GlobalTheory.SchemeWords.isProjOver_iff** : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsProjOver f ↔ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsQProjOver f ∧ IsProper f

theorem **faithfullyFlat** (h : IsCompletedStrictHens A R) : Module.FaithfullyFlat A R

theorem **injective** (h : IsCompletedStrictHens A R) : Function.Injective (algebraMap A R)

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.IsThetaFrameMulUnit

theorem **isHolTo_const** {V : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] {X : Scheme.{0}} {B : Set V} (hB : IsOpen B) (x : Sp ℂ ⟶ X) : IsHolTo B (fun _ => x)

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.MemOfEqMulInvPow

theorem **mem_of_eq_mul_inv_pow** (W : ValuationSubring ℂ) {x a : ℂ} (ha : a ∈ W) (n : ℕ) (h : x = a * x⁻¹ ^ n) : x ∈ W

theorem **exists_root_and_zeta** (l : Level H) (c : Coeff l) (W : ValuationSubring ℂ) : ∃ r z : c.V W, (((c.V W).subtype r : c.F) : ℂ) = H.rt0 ∧ (((c.V W).subtype z : c.F) : ℂ) = zeta l.A

theorem **localRing_ne_top** (l : Level H) (c : Coeff l) (W : ValuationSubring ℂ) {p : ℕ} (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) : c.V W ≠ ⊤

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.MemOfPowEqInt

theorem **mem_of_pow_eq_int** {F : Type} [Field F] (V : ValuationSubring F) (x : F) (n : ℕ) (hn : 0 < n) (m : ℤ) (h : x ^ n = m) : x ∈ V

theorem **exists_values** (c : Coeff l) (W : ValuationSubring ℂ) : ∃ r z : c.V W, (((c.V W).subtype r : c.F) : ℂ) = H.rt0 ∧ (((c.V W).subtype z : c.F) : ℂ) = zeta l.A

theorem **int_ne_rt** (c : Coeff l) (r : ℤ) : ((Int.castRingHom c.F r : c.F) : ℂ) ≠ H.rt0

theorem **nonempty_int** (D : Datum H) (c : Coeff l) : Nonempty (ArcIn D l (Int.castRingHom c.F) c.F (Int.castRingHom c.F))

theorem **rank_le** {D : Datum H} {c : Coeff l} {O : Type} [CommRing O] {β : O →+* c.F} {O' : Type} [CommRing O'] [IsDomain O'] {b : O →+* O'} (a : ArcIn D l β O' b) (r z : O) (hr : ((β r : c.F) : ℂ) = H.rt0) (hz : ((β z : c.F) : ℂ) = zeta l.A) (τ : 𝒪 E →+* AlgebraicClosure (FractionRing O')) : Module.finrank (𝒪 E) a.D₀.X ≤ max D.p D.q

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.ModuliNonemptyOfObj

theorem **constPt_idem** {k : Type} [Field k] {X : Scheme.{0}} (x : Sp (DualNumber k) ⟶ X) : constPt (constPt x) = constPt x

theorem **two_mul_half_degree** : 2 * (Module.finrank ℚ E / 2) = Module.finrank ℚ E

theorem **zeta_prim** (n : ℕ) (hn : 0 < n) : IsPrimitiveRoot (zeta n) n

theorem **le_of_conj** (l : Level H) (F : IntermediateField ℚ ℂ) (hc : ∀ (τ : E →+* ℂ) (a : E), τ a ∈ F) (hz : zeta l.M' ∈ F) : KA l ≤ F

theorem **coeff_nonempty** (l : Level H) : Nonempty (Coeff l)

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.PremisesOfProper

theorem **existsUnique_map** (V K : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] (t t' : SAB H (Sp V)) (tK : SAB H (Sp K)) (a : SABMap tK t (spMap (algebraMap V K))) (a' : SABMap tK t' (spMap (algebraMap V K))) : ∃! ψ : SABMap t t' (𝟙 (Sp V)), a.φ ≫ ψ.φ = a'.φ

theorem **extV_holds** : ExtVStatement

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.PsiUnitOnIffInterior

theorem **psiUnitOn_iff_interior** (D : Out.DegPre R) (χ : D.X) (U : Set (Sp R)) : PsiUnitOn D χ U ↔ PsiUnitOn D χ (interior U)

theorem **qkExt_compat** {D : Out.DegPre R} {F F' : Type} [Field F] [Algebra (FractionRing R) F] [Field F'] [Algebra (FractionRing R) F'] (f : F →+* F') {k : ℕ} (p : Out.QkPt D F k) (p' : Out.QkPt D F' k) (h : QkExt f p p') : f.comp (algebraMap (FractionRing R) F) = algebraMap (FractionRing R) F'

theorem **halfUnit_add_star** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : H.w0 + H.bar H.w0 = 1

theorem **pt_act** : D.xK y ≫ (oa.onB.act w).left = D.xK (w • y)

theorem **pt_sum** : D.xK y ≫ (padd D.P.A (𝟙 _) (oa.onB.act w)).left = D.xK (y + w • y)

theorem **pt_zero** : D.xK y ≫ (toUnit D.P.A ≫ η[D.P.A]).left = (toUnit (Out.TK R)).left ≫ eSec D.P.A

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.QuotientTiedToDatum

theorem **isGrpPullback_id** {S : Scheme.{0}} (G : Over S) [GrpObj G] : IsGrpPullback G G (𝟙 S) (𝟙 G.left)

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.RangeJSubsetAbLocus

theorem **extends_of_not_subset** (Y : Model Xo c β) (U : Y.X.Opens) (hU : ¬ Set.range Y.j ⊆ (U : Set Y.X)) (σ : 𝟙_ (Over (Xo.XF c)) ⟶ Y.oF.t.G) : Y.Extends U σ

theorem **range_j_subset_abLocus** (Y : Model Xo c β) : Set.range Y.j ⊆ abLocus Y.t.G

theorem **extends_iff_of_subset** (Y : Model Xo c β) (U : Y.X.Opens) (hU : Set.range Y.j ⊆ (U : Set Y.X)) (σ : 𝟙_ (Over (Xo.XF c)) ⟶ Y.oF.t.G) : Y.Extends U σ ↔ ∃ (j' : Xo.XF c ⟶ U.toScheme) (σ' : U.toScheme ⟶ Y.t.G.left), j' ≫ U.ι = Y.j ∧ σ' ≫ Y.t.G.hom = U.ι ∧ j' ≫ σ' = σ.left ≫ Y.m.φ

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.RigidAndInjective

theorem **dataRigid_of_laws** (hL : Laws) : DataRigidSt

theorem **dataRigid** : DataRigidSt

theorem **rigid_and_injective** : DataRigidSt ∧ QkToAkInjSt

Development.AbelianVarietyTheory.Counterexamples.ModuliSurvey.W0EqOfRtEq

theorem **goodAt_iff** (l : Level H) {p : ℕ} (hp : p.Prime) : l.GoodAt p ↔ ∀ k, l.M ≠ p ^ k

theorem **trace_sign** (t : ℂ) : Algebra.trace ℝ ℂ t = 2 * t.re ∧ Algebra.trace ℝ ℂ (Complex.I * t) = -2 * t.im

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.CongruenceBoundParameters

theorem **window_pos** {D : D7} {L : SizeData₂ D} (P : ParameterChoice' D L) (W : ℝ) (hW : W ∈ baseSizes L) (n : ℤ) (hn : n ∈ window L W) : 0 < n

theorem **two_lStar_lt_b** {D : D7} (L : SizeData₂ D) : 2 * LevelStarExponents.lStar L < Real.exp L.sc.L

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.CriterionWithoutRoot

theorem **q_eq** : W.q = ∑ k ∈ Finset.range PD.p, (1 + (letπ : LetA O n)) ^ k

theorem **φ_one_add** : W.φA (1 + letπ) = (1 + (letπ : LetA O n)) ^ PD.p

theorem **φpow_one_add** (j : ℕ) : (W.φA ^ j) (1 + letπ) = (1 + (letπ : LetA O n)) ^ (PD.p ^ j)

theorem **Q_eq** : W.Q = ∑ k ∈ Finset.range PD.p, ((1 + (letπ : LetA O n)) ^ (PD.p ^ (W.s - 1))) ^ k

theorem **value_Q** (i : ZMod n) (ζ : LL) (hζ : ∃ j : ℕ, IsPrimitiveRoot ζ (PD.p ^ j)) : PowerSeries.constantCoeff (W.lev i ζ W.Q) = ∑ k ∈ Finset.range PD.p, (ζ ^ (PD.p ^ (W.s - 1))) ^ k

theorem **p_ne_zero** : ((PD.p : ℕ) : LL) ≠ 0

theorem **qRoot_iff** (i : ZMod n) (ζ : LL) : QRoot W i ζ ↔ (∃ c : O, algebraMap O LL c = ζ) ∧ IsPrimitiveRoot ζ (PD.p ^ W.s)

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.CrystallineCharacterRatio

theorem **chi_eq_cyc** (P : LocPrime l) (j : Ω →+* Om l) (σ : Formalism.ConjugateSelfDual.G E Ω) (g : GQ l) (h : ∀ x, g (j x) = j (σ x)) : ((P.χQ g : ℚ_[l]ˣ) : ℚ_[l]) = (((ReciprocityTheory.Gal.cycZ E Ω l σ : ℤ_[l]ˣ) : ℤ_[l]) : ℚ_[l])

theorem **locChi_eps** (g : Br.G D.nbr i w hw) : ((Br.locChi D.nbr i w hw (epsU D.nd i) g : (Br.LL D.nbr i w hw)ˣ) : Br.LL D.nbr i w hw) = algebraMap ℚ_[D.nd.l i] (Br.LL D.nbr i w hw) (((Br.PD D.loc D.nbr i w hw).χ g : ℚ_[D.nd.l i]ˣ) : ℚ_[D.nd.l i])

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.CrystallineCyclotomic

theorem **nontrivial_B** : Nontrivial B

theorem **dcris_trivial** [FiniteDimensional Qp LL] : Dcris PD (Representation.trivial LL G LL) = LinearMap.range (AlgebraTensorModule.map (LinearMap.id : LL →ₗ[LL] LL) (IsScalarTower.toAlgHom Qp K₀ B).toLinearMap)

theorem **isCrys_trivial** [FiniteDimensional Qp LL] : IsCrys PD (Representation.trivial LL G LL)

theorem **isCrys_cyc** [FiniteDimensional Qp LL] : IsCrys PD (charRep (cyc PD LL))

theorem **exists_crys** (l : ℕ) [Fact l.Prime] (P : LocPrime l) (K LL : FinExt l) : IsCrys (P.PD K) (Representation.trivial ɪLL.1 ɪ(GK K) ɪLL.1)

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.EmbeddingValuationsAnti

theorem **v_strictAnti** {R : Ranks} (P : HCParam R) : StrictAnti P.v

theorem **b_strictMono** {R : Ranks} (P : HCParam R) : StrictMono P.b

theorem **neg_b_lt_v** {R : Ranks} (P : HCParam R) (i : Fin (R.k − 1)) (j : Fin R.m) : −P.b j < P.v i

theorem **nodup_hcSigma** {R : Ranks} (P : HCParam R) : (hcSigma P).Nodup

theorem **emb_injective** (D : D7) : Function.Injective D.host.emb

theorem **embTau_injective** (D : D7) : Function.Injective (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D)

theorem **embTau_ne_bar** (D : D7) (τ τ' : D.what₂.Tau) : embTauBar D τ ≠ GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ'

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.MemTypeSubspaceIffUnipotentCharacter

theorem **act_single** (i : X.Tw) (j : ℕ) (b : Bool) (g : G0g N) (x : Y.Fm i j b) : act Y i j b (MonoidAlgebra.single g 1) x = Y.actG i j b g x

theorem **act_mem** (i : X.Tw) (j : ℕ) (b : Bool) (W : Submodule (FractionRing (X.O i)) (Y.Fm i j b)) (hW : GStable Y i j b W) (z : MonoidAlgebra (FractionRing (X.O i)) (G0g N)) (x : Y.Fm i j b) (hx : x ∈ W) : act Y i j b z x ∈ W

theorem **eG_on_irred** (i : X.Tw) (j : ℕ) (b : Bool) (W : Submodule (FractionRing (X.O i)) (Y.Fm i j b)) (hW : GIrred Y i j b W) : (∀ x ∈ W, eG Y i j b x = x) ∨ (∀ x ∈ W, eG Y i j b x = 0)

theorem **act_comm** (i : X.Tw) (j : ℕ) (b : Bool) (T : Module.End (FractionRing (X.O i)) (Y.Fm i j b)) (hT : ∀ g x, Y.actG i j b g (T x) = T (Y.actG i j b g x)) (z : MonoidAlgebra (FractionRing (X.O i)) (G0g N)) (x : Y.Fm i j b) : act Y i j b z (T x) = T (act Y i j b z x)

theorem **eG_actG** (i : X.Tw) (j : ℕ) (b : Bool) (g : G0g N) (x : Y.Fm i j b) : eG Y i j b (Y.actG i j b g x) = Y.actG i j b g (eG Y i j b x)

theorem **mem_typeSubspace_iff_unipotent_character** (i : X.Tw) (j : ℕ) (b : Bool) (x : Y.Fm i j b) : x ∈ Fm0 Y i j b ↔ ∀ u : Y.U, Y.actG i j b u x = ((Y.psiU i b u : (FractionRing (X.O i))ˣ) : FractionRing (X.O i)) • x

theorem **goodPart_le_typeSubspace** (i : X.Tw) (j : ℕ) (b : Bool) : V0 Y i j b ≤ Fm0 Y i j b

theorem **twOK_of_psiOk** (i : X.Tw) (A Q : Finset (PlE D.host)) (k' : ℕ) (ψ' : X.RCG i Q →* (X.Ok i k')ˣ) (h : Formalism.MainNotions.BlochKatoReduction.Gen.PsiOk i A Q k' ψ') (b : Bool) : TwOK i k' Q (Formalism.MainNotions.BlochKatoReduction.Gen.psiAt b (ψ'.comp (X.art i Q))) ∧ UnrG (Formalism.MainNotions.BlochKatoReduction.Gen.psiAt b (ψ'.comp (X.art i Q))) ∧ Disjoint Q A

theorem **admissibleGoodIdempotent_iff_reduced** : GroupAlgebraIntegrality Y ↔ ∀ (i : X.Tw), X.NonzeroIndexPlace i → ∀ (b : Bool), (∃ c : MonoidAlgebra (X.O i) (G0g N), MonoidAlgebra.map (algebraMap (X.O i) (FractionRing (X.O i))).toAddMonoidHom c = ((D.what₂.restRowsFiniteGroupCard : ℕ) : FractionRing (X.O i)) • Y.eAlg i b) ∧ ∀ j ∈ Finset.Icc 1 D.R.m, ∀ g k P, HkIdx i k P → (Y.actG i j b g).toLinearMap ∘ₗ Y.hkF i j b k P = Y.hkF i j b k P ∘ₗ (Y.actG i j b g).toLinearMap

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.ReindexedFiltrationComparison

theorem **slow_add** (a b : ℤ) : slow (a + b) ≤ slow a + slow b

theorem **slow_mono** (a : ℤ) : slow a ≤ slow (a + 1)

theorem **slow_surj** (a : ℤ) : ∃ b, slow b = a

theorem **fil_antitone** (PD : PeriodData G Qp K₀ K B BdR) : Antitone PD.Fil

theorem **not_jump_neg_one** (P : LocPrime l) {K LL : FinExt l} (T : Lat K LL) (τ : Emb K LL) : ¬ IsJumpV ((slowly P).PD K) T.ρ τ (−1)

theorem **not_twistJumps** (P : LocPrime l) {K LL : FinExt l} {s : ℕ} (h : Good K s LL) (T T' : Lat K LL) (e : IsTwistRep (cycLL (P.PD K) ɪLL.1) T.ρ T'.ρ) (hT : IsDR (P.PD K) T.ρ) (τ : Emb K LL) (h0 : IsJumpV (P.PD K) T.ρ τ 0) : ¬ TwistJumps l (slowly P)

theorem **not_deRhamComparison** (P : LocPrime l) {K LL : FinExt l} {s : ℕ} (h : Good K s LL) (T T' : Lat K LL) (e : IsTwistRep (cycLL (P.PD K) ɪLL.1) T.ρ T'.ρ) (hT : IsDR (P.PD K) T.ρ) (τ : Emb K LL) (h0 : IsJumpV (P.PD K) T.ρ τ 0) : ¬ DeRhamComparison l (slowly P)

theorem **continuous_cyc** (P : LocPrime l) (K LL : FinExt l) : Continuous fun g => ((cycLL (P.PD K) ɪLL.1 g : (ɪLL.1)ˣ) : ɪLL.1)

theorem **twist_triv** (P : LocPrime l) (K LL : FinExt l) : IsTwistRep (cycLL (P.PD K) ɪLL.1) (trivLat K LL).ρ (tateLat P K LL).ρ

theorem **one_notMem** : (1 : BdR) ∉ PD.Fil 1

theorem **ddR_triv** : DdR PD (charRep (1 : G →* LLˣ)) = LinearMap.range (incK (Qp

theorem **isDR_triv** : IsDR PD (charRep (1 : G →* LLˣ))

theorem **jump_triv** (hK : Function.Surjective (algebraMap Qp K)) (τ : K →ₐ[Qp] LL) : IsJumpV PD (charRep (1 : G →* LLˣ)) τ 0

theorem **not_comparison_of_surj** (P : LocPrime l) {K LL : FinExt l} {s : ℕ} (h : Good K s LL) (hs : Function.Surjective (algebraMap ℚ_[l] ɪK.1)) (τ : Emb K LL) : ¬ DeRhamComparison l (slowly P)

theorem **not_comparison_slow** (P : LocPrime l) : ¬ DeRhamComparison l (slowly P)

theorem **not_forall** (P : LocPrime l) (AI : LocAinf l P) : ¬ ∀ (P' : LocPrime l) (_ : LocAinf l P'), DeRhamComparison l P'

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.SplitRowIffOfIsEmpty

theorem **conj_sqrt** (φ : H.E →+* ℂ) (x : H.E) : φ (H.conj x) = starRingEnd ℂ (φ x)

theorem **sqrt_sq** (φ : H.E →+* ℂ) : φ (H.sqd) ^ 2 = -(H.d : ℂ)

theorem **mem_halfType_false** (φ : H.E →+* ℂ) : φ ∈ halfType H false ↔ φ (H.sqd) = -(Complex.I * (Real.sqrt H.d : ℂ))

theorem **halfType_xor** (φ : H.E →+* ℂ) : Xor' (φ ∈ halfType H true) (φ ∈ halfType H false)

theorem **nonempty_emb** : Nonempty (Br.Kw D.nbr i w hw →ₐ[ℚ_[D.nd.l i]] Br.LL D.nbr i w hw)

theorem **embIn_xor** (τ : Br.Kw D.nbr i w hw →ₐ[ℚ_[D.nd.l i]] Br.LL D.nbr i w hw) : Xor' (EmbIn (N

theorem **not_both_columns** : ¬ (PlIn (N

theorem **bridge_place** (x : 𝒪 D.host.E) : x ∈ w.asIdeal ↔ ‖(D.nbr i w hw).j (algebraMap D.host.E D.nd.Om (x : D.host.E))‖ < 1

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.StrLeStrAddTwo

theorem **aPl_last_pos** (i : Fin D.R.m) (τ : D.what₂.Tau) : 0 < aPl D i (D.R.k + i) τ (D.R.k + i - 1)

theorem **str_le_str_add_two** (q : ℕ) (z : ℂ) (n : ℕ) : str q z n ≤ str q z (n + 2)

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.ThetaSeriesDatumMemLevelTypeForms

theorem **charZero_K** (i : X.Tw) : CharZero (FractionRing (X.O i))

theorem **AO_apply** (i : X.Tw) (j : ℕ) (b : Bool) (x : Y.Fm i j b) : AO Y i j b x = ((D.what₂.restRowsFiniteGroupCard : ℕ) : FractionRing (X.O i)) • eG Y i j b x

theorem **thetaSeriesDatum_mem_levelTypeForms** (E : ExpDat Y) (i : X.Tw) (j : ℕ) (b : Bool) (a : Y.Dat i j b) (ha : a ∈ E.Dat0 i j b) : Y.th i j b a ∈ Fm0 Y i j b

theorem **theta_le_integral_forms** (E : ExpDat Y) (i : X.Tw) (j : ℕ) (b : Bool) {Sg : Set (Y.Dat i j b)} (h : Sg ⊆ E.Dat0 i j b) : Th Y i j b Sg ≤ (Fm0 Y i j b).restrictScalars (X.O i)

theorem **clause_two** (E : ExpDat Y) (i : X.Tw) (j : ℕ) (b : Bool) {Sg : Set (Y.Dat i j b)} (h : Sg ⊆ E.Dat0 i j b) : ∀ x ∈ (Th Y i j b Sg).map (AO Y i j b), x ∈ V0 Y i j b

theorem **span_A_eq** (i : X.Tw) (j : ℕ) (b : Bool) (T : Submodule (X.O i) (Y.Fm i j b)) : Submodule.span (FractionRing (X.O i)) ((T.map (AO Y i j b) : Submodule (X.O i) (Y.Fm i j b)) : Set (Y.Fm i j b)) = Submodule.span (FractionRing (X.O i)) ((T.map (eO Y i j b) : Submodule (X.O i) (Y.Fm i j b)) : Set (Y.Fm i j b))

theorem **fullRk_iff** (E : ExpDat Y) (i : X.Tw) (j : ℕ) (b : Bool) : FullRk Y E i j b (SgJ Y E i j b) ↔ V0 Y i j b ≤ Submodule.span (FractionRing (X.O i)) (eTh Y i j b (SgJ Y E i j b) : Set (Y.Fm i j b))

theorem **descent** {O O' V : Type} [CommRing O] [IsDomain O] [CommRing O'] [IsDomain O'] [Algebra O O'] (hinj : Function.Injective (algebraMap O O')) [AddCommGroup V] [Module (FractionRing O) V] [Module O V] [IsScalarTower O (FractionRing O) V] (T : Submodule O V) (x : V) (c : O') (hc : c ≠ 0) (h : c • ((1 : O') ⊗ₜ[O] x) ∈ T.baseChange O') : x ∈ Submodule.span (FractionRing O) (T : Set V)

Development.AbelianVarietyTheory.Counterexamples.NonGaloisSurvey.UnitsOfFieldDD

theorem **sumEquiv_apply** (h : IsSumLat i i') (x : T.V) (y : T'.V) : sumEquiv i i' h (x, y) = i.f x + i'.f y

theorem **_root_.GlobalTheory.Survey.UnitsOfFieldRootsOfUnity.symm_act** (h : _root_.GlobalTheory.IndefiniteRowSideFullHecke.UnitDualParts.IsSumLat i i') (g : G) (z : T''.V) : (_root_.GlobalTheory.Survey.UnitsOfFieldRootsOfUnity.sumEquiv i i' h).symm (T''.ρ g z) = (T.ρ g ((_root_.GlobalTheory.Survey.UnitsOfFieldRootsOfUnity.sumEquiv i i' h).symm z).1, T'.ρ g ((_root_.GlobalTheory.Survey.UnitsOfFieldRootsOfUnity.sumEquiv i i' h).symm z).2)

theorem **latHom_ext** {T T' : LocLat G Qp LL O} {f g : LatHom T.L T'.L} (h : ∀ x, f.f x = g.f x) : f = g

theorem **map_of_zero** {T T' : LocLat G Qp LL O} (f : LatHom T.L T'.L) (h : ∀ x, f.f x = 0) (x : W.DD T) : W.map f x = 0

theorem **sum_N** {T T' T'' : LocLat G Qp LL O} (hT : IsCrys PD T.ρ) (hT' : IsCrys PD T'.ρ) (hT'' : IsCrys PD T''.ρ) (i : LatHom T.L T''.L) (i' : LatHom T'.L T''.L) (h : IsSumLat i i') (x : W.DD T) (y : W.DD T') : W.map i x + W.map i' y ∈ W.N T'' ↔ (x ∈ W.N T ∧ y ∈ W.N T')

theorem **sumFor_of_crys** (hc : ∀ (T T' T'' : LocLat G Qp LL O) (i : LatHom T.L T''.L) (i' : LatHom T'.L T''.L), IsSumLat i i' → IsCrys PD T.ρ → IsCrys PD T'.ρ → IsCrys PD T''.ρ) : SumFor W

Development.AbelianVarietyTheory.Counterexamples.PlaceInjectivity.NotPtIntZeroTarget

theorem **ptInt_iff** (X : Ch D u) : PtInt X ↔ ∀ x ∈ (X.fm.lat D.R.m).L, ∃ y ∈ X.fm.ptLat.L, X.fm.ptEmb y = Pt X (X.fm.emb D.R.m x)

Development.AbelianVarietyTheory.Counterexamples.PlaceInjectivity.PlaceInjective

theorem **embOf_mem** (D : D7) (ρ : D.what₂.Tau) : embOf D ρ ∈ D.host.cmType

theorem **place_injective** (D : D7) : Function.Injective fun ρ : D.what₂.Tau => InfinitePlace.mk (embOf D ρ)

theorem **place_bijective** (D : D7) : Function.Bijective fun ρ : D.what₂.Tau => InfinitePlace.mk (embOf D ρ)

Development.AbelianVarietyTheory.Counterexamples.PlaceInjectivity.StandardDeterminantUnits

theorem **val_le_one_of_intAt** {P : HeightOneSpectrum (𝒪 E)} {a : E} (h : IntAt P a) : P.valuation E a ≤ 1

private theorem **single_one_mem_stdAt** (P : HeightOneSpectrum (𝒪 E)) (j : Fin n) : (Pi.single j (1 : E) : Sp E n) ∈ stdAt E n P

theorem **val_det_le_one** {P : HeightOneSpectrum (𝒪 E)} (g : Sp E n →ₗ[E] Sp E n) (hg : ∀ x ∈ stdAt E n P, g x ∈ stdAt E n P) : P.valuation E (LinearMap.det g) ≤ 1

theorem **mem_at_conj** {N : Submodule (𝒪 E) (Sp E n)} (hl : N.FG) (hu : IsUnimodular (hStd E n) N) (P : HeightOneSpectrum (𝒪 E)) (x : Sp E n) : x ∈ localization₂ (conjPrime₂ P).asIdeal N ↔ ∀ y ∈ localization₂ P.asIdeal N, IntAt (conjPrime₂ P) (hStd E n y x)

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.DimPartsOfGoodIdempotentZero

theorem **dimPartsAt_iff** {Pg : Finset (Pl D.host.E)} (h : IsRecPlaces D Pg) (P : LevelP Pg D.R.r ℂ) : DimPartsAt h P ↔ DimAtWith (eCoef h) P

theorem **recMatching_eCoef** {Pg : Finset (Pl D.host.E)} (h : IsRecPlaces D Pg) : (recMatching h).eCoef = eCoef h

theorem **dimParts_of_anyMatching** (H : DimPartsAnyMatching) : DimParts

theorem **eLin_zero** (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ V.rank) : eLin V β L 0 = 0

theorem **goodPart_zero** {V : HSp D} {S : D.what₂.Tau → SPar} (θ : InfFam D) (M : Mult.DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) : Mult.goodPart θ M J χ 0 = ⊥

theorem **dimAtWith_zero** {Pg : Finset (Pl D.host.E)} (P : LevelP Pg D.R.r ℂ) : DimAtWith 0 P

theorem **dimParts_of_good_idempotent_zero** (h0 : D.ntypes.e0 = 0) : DimensionDecompositionAtBasePrime D

theorem **paramGap_of_params** (L : SizeData₂ D) (Pm : ParameterChoice' D L) : ParamGap D

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.ExistsCharNotOnto

theorem **char_withTup** (h : R.Char F) {mu tw e} (ht : (withTup R mu tw e).IsTup F) : (withTup R mu tw e).Char F where

theorem **withTup_nonzeroIndexPlace_iff** {mu tw e} (w) : (withTup R mu tw e).NonzeroIndexPlace F w ↔ R.NonzeroIndexPlace F w

theorem **char_withErr** (h : R.Char F) (e) : (withErr R e).Char F

theorem **not_onto_neg** {w} (hw : w ∈ R.twPl) (j : Fin D.R.m) (C CH : ℝ) : ¬ (withErr R fun _ _ => -1).LOnto F C CH

theorem **exists_char_not_onto** (h : R.Char F) {w} (hw : w ∈ R.twPl) (j : Fin D.R.m) : ∃ R' : RealDat D A Ks, R'.Char F ∧ R'.mu = R.mu ∧ R'.twPl = R.twPl ∧ ∀ C CH, ¬ R'.LOnto F C CH

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.FreeLiftMetric

theorem **majAt_scale** {d : ℕ} (X : InfinitePlace K → Matrix (Fin d) (Fin d) ℂ) (hX : ∀ v, IsUnit (X v).det) (t : InfinitePlace K → ℝ) (ht : ∀ v, 0 < t v) (v : InfinitePlace K) (x y : Fin d → K) : Formalism.HermitianLatticesTheory.majAt (Pi.basisFun K (Fin d)) (scaleFrame t X) v x y = (t v : ℂ) * Formalism.HermitianLatticesTheory.majAt (Pi.basisFun K (Fin d)) X v x y

theorem **scale_unit** {d : ℕ} (X : InfinitePlace K → Matrix (Fin d) (Fin d) ℂ) (hX : ∀ v, IsUnit (X v).det) (t : InfinitePlace K → ℝ) (ht : ∀ v, 0 < t v) (v : InfinitePlace K) : IsUnit (scaleFrame t X v).det

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.IrrationalRotationBk

theorem **buRational_withBk** (D : D7) (B) : (withBk D B).BuRational ↔ ∀ u (φ : (D.div.Nu u).V) (x : D.div.RR.V), D.what₂.Bu u (D.jN u φ) (D.jEE x) = D.what₂.embLL (B u φ x)

theorem **bk_unique** (D : D7) (B B') (h : (withBk D B).BuRational) (h' : (withBk D B').BuRational) : B = B'

theorem **exists_not_buRational** (D : D7) (u : D.what₂.Word) [Nontrivial (D.div.Nu u).V] [Nontrivial D.div.RR.V] : ∃ B, ¬ (withBk D B).BuRational

theorem **exists_bk_iff** (D : D7) : (∃ B, (withBk D B).BuRational) ↔ ValuesRational D

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.LeafHypothesesConst

theorem **abs_truncZero_le** (y : ℝ) (a : ℤ) : |(truncZero y : ℝ)| ≤ |y| ∧ |((a + truncZero y : ℤ) : ℝ)| ≤ |(a : ℝ) + y| + |y|

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.LeafIPOfTwoConstants

theorem **clauseB_of_lifts** [IsTotallyComplex K] {Df' : DefSide Bs} (S : IndefData Df) (S' : IndefData Df') (N : IndefNum Bs) (O : FunData K Df) {C b : ℝ} (hC : 0 ≤ C) (hb : 1 ≤ b) (hl : LevelLink S S') (h6 : S'.L6 N) (hm : Fintype.card Bs.J ≤ O.m) (hN : ∀ j, -(3 * C * O.f * (O.N : ℝ) ^ 2 * Real.log b) ≤ N.l6 j) : ClauseB S O C b

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.LevelVectorSizeBounds

theorem **four_le_of_sizes** (i : Fin D.R.m) (b : ℝ) (h : Sizes D i b) (τ : D.what₂.Tau) : 4 ≤ b

theorem **hookProd_pos** (μ : YoungDiagram) : 0 < Formalism.LatticeIndexTer.hookProd μ

theorem **locK_eq_one** {E : Type} [Field E] [NumberField E] {χ : Formalism.L2.Host.HChar E} {P : HeightOneSpectrum (𝒪 E)} (h : χ.UnrAt P) (a : 𝒪 E) (k : Eˣ) (hk : (k : E) = a) (ha : a ∉ P.asIdeal) : χ.locK P k = 1

theorem **not_both** {E : Type} [Field E] [NumberField E] {P : HeightOneSpectrum (𝒪 E)} {p q : ℕ} (hp : p.Prime) (hq : q.Prime) (hne : q ≠ p) (h1 : (p : 𝒪 E) ∈ P.asIdeal) (h2 : (q : 𝒪 E) ∈ P.asIdeal) : False

theorem **q_mem** {P : Pl D.host.E} {q : ℕ} (h : Nat.card (kP P) = q) : (q : 𝒪 D.host.E) ∈ P.asIdeal

theorem **g_mem** {F : Face D A} (h : R.Char F) {PP : Pl D.host.E} (hP : PP ∈ R.Pg) : ∃ k, ((D.host.g k : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal

theorem **isConjChars_base** {F : Face D A} (h : R.Char F) (i : Fin D.R.m) (t : T) : IsConjChars R i t (RingEquiv.refl ℂ) (R.toLevCh.mu t ((i : ℕ) + 1)) R.eta (fun PP => if PP ∈ R.Pg then (R.chi ((i : ℕ) + 1) PP)⁻¹ else R.xiR t PP)

theorem **isLocal_triv** : IsLocalLevel (D

theorem **lev_triv** (i : Fin D.R.m) (t : T) (m) : lev D.sp.cm (Vc D i).gram (R.βV i) (datV i R.Pg (R.Q t) R.triv m) = lev D.sp.cm (Vc D i).gram (R.βV i) (R.datVt i t)

theorem **grp_triv** (i : Fin D.R.m) (t : T) (m) : grpOf (Vc D i) (R.βV i) (datV i R.Pg (R.Q t) R.triv m) = (levEnv _ (R.βV i) (R.datVt i t)).J

theorem **eq_one_of_full_upper_block_eq_one** {κ : Type} [CommRing κ] {n r : ℕ} (hr : r ≤ n) (e : n = r) (g : GL (Fin n) κ) (h : upperLeftBlock hr g = 1) : g = 1

theorem **omegaG_triv** (i : Fin D.R.m) (m) (y : ∀ PP : Pl D.host.E, Matrix (Fin (Vc D i).rank) (Fin (Wc D i).rank) (kP PP)) : Datum.omegaG (R.datWp i R.triv) m y = Datum.omegaK R.Pg y

theorem **isLevelVec_base** {F : Face D A} (h : R.Char F) (i : Fin D.R.m) (t : T) : IsLevelVec R.ψ R.θ (Vc D i) (Wc D i) (cart (R.MW i (sg0 D A))) (R.ΛV i) (R.ΛW i) (R.βV i) (R.βW i) (R.datWp i R.triv) (R.Q t) (fun PP => if PP ∈ R.Pg then (R.chi ((i : ℕ) + 1) PP)⁻¹ else R.xiR t PP) (R.vacP i) (R.vP i t)

theorem **levFn_congr** {n : ℕ} (β : GL (Fin n) (Lev.EAf D)) (r : ℕ) (hr : r ≤ n) (t s : ℕ) (Pg Q : Finset (Pl D.host.E)) (P : LevelP Pg r ℂ) (cg cg' : ℕ → ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (xq xq' : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (h1 : ∀ c, ∀ PP ∈ Pg, cg c PP = cg' c PP) (h2 : ∀ PP ∈ Q, xq PP = xq' PP) (x) : levFn β (r, hr, t, s, Pg, Q, P, cg, xq) x = levFn β (r, hr, t, s, Pg, Q, P, cg', xq') x

theorem **chr_base** {F : Face D A} (h : R.Char F) (i : Fin D.R.m) (t : T) : chrOf (Vc D i) (R.βV i) (datV i R.Pg (R.Q t) R.triv (fun PP => if PP ∈ R.Pg then (R.chi ((i : ℕ) + 1) PP)⁻¹ else R.xiR t PP)) = (levEnv _ (R.βV i) (R.datVt i t)).χ

theorem **delta_base** (i : Fin D.R.m) : deltaAtEmb D i (sg0 D A) = RealDat.Delta i

theorem **apply_base** {F : Face D A} (hA : AdelicLiftScalar R F) (h : R.Char F) (i : Fin D.R.m) (b : ℝ) (hb : Sizes D i b) (t : T) (ht : PrimeToHost R t) : ∃ th, IsLift R i (sg0 D A) (R.ωP i t) (R.ptP i t) (R.cV i) (R.ioP i) (R.vV i t) (R.fdV i t) (R.HV i t) th ∧ ∀ ev, typePart R i (sg0 D A) t R.triv ev ≠ ⊥ → ∃ l : ℝ, 0 < l ∧ (∀ x ∈ typePart R i (sg0 D A) t R.triv ev, th (adjLev R.triv th x) = (l : ℂ) • x) ∧ mainTerm D i (sg0 D A) − 8 * (D.R.f : ℝ) * (D.R.nV : ℝ) ^ 2 * Lp D b ≤ Real.log l

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.MeasureFailsForLargeConstant

theorem **lcv_of_oneConstant** [IsTotallyComplex K] {d : ℕ} (hd : 0 < d) (X X' : InfinitePlace K → Matrix (Fin d) (Fin d) ℂ) (hX : ∀ v, IsUnit (X v).det) (hX' : ∀ v, IsUnit (X' v).det) (c : InfinitePlace K → ℝ) (hc : OneConstant X X' c) (M : Submodule (𝒪 K) (Fin d → K)) (hM : Submodule.IsLattice K M) : Formalism.DivisionValues.lcv X M / (d : ℝ) = Formalism.DivisionValues.lcv X' M / (d : ℝ) + 2 / (finrank ℚ K : ℝ) * ∑ v, Real.log (c v)

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.OfBeforeProjector

theorem **IsRat.zero** : IsRat O κ cc 0

theorem **IsRat.add** {F F' : GA D (Wc D i) → ℂ} (h : IsRat O κ cc F) (h' : IsRat O κ cc F') : IsRat O κ cc (F + F')

theorem **IsRat.mul** (b : D.what₂.LL) {F : GA D (Wc D i) → ℂ} (h : IsRat O κ cc F) : IsRat O κ cc (D.what₂.embLL b • F)

theorem **levChi_flag** (k : GA D (Wc D i)) : Formalism.IndefR.Lev.levChi (Wc D i) (R.βW i) (Formalism.IndefR.Lev.flagL (R.datWp i R.triv)) k = 1

theorem **IsRat.act** (hE : O.EquivAt κ cc) (hχ : R.grp_chiW) (g : G) {F : GA D (Wc D i) → ℂ} (h : IsRat O κ cc F) : IsRat O κ cc (Formalism.IndefR.Links.actL (Wc D i) (R.βW i) (R.datWp i R.triv) (R.eG g) F)

theorem **of_before_projector** (hE : O.EquivAt κ cc) (hχ : R.grp_chiW) (hG : R.thP_thG) (ph : ℂ) (h : ∀ x : A.fam.amb.FL i A.μ0, IsRat O κ cc (ph • R.thF i A.μ0 (jVx R i A.μ0 x) v₀)) : O.PhRat κ v₀ cc ph

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.OneLeAPlLast

theorem **labE_fixed** (τ : D.what₂.Tau) : labE D.what₂.embLL τ = tauE D τ

theorem **lab_fixed** (ρ : D.what₂.Tau) : lab D.what₂.embLL ρ = ρ

theorem **one_le_aPl_last** (i : Fin D.R.m) (τ : D.what₂.Tau) : 1 ≤ aPl D i (D.R.k + i) τ (D.R.k + i − 1)

theorem **parAt_fixed** (W : HSp D) (ρ : D.what₂.Tau) : parAt W D.what₂.embLL ρ = Formalism.L2.NTheta.holPar _ _ (Formalism.L2.NTheta.wtOf D W ρ)

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.PhaseBreaksTargetParts

theorem **inner_of_unitary** {X : Type} [NormedAddCommGroup X] [InnerProductSpace ℂ X] (U : X ≃ₗᵢ[ℂ] X) (g : X → X) (hg : ∀ a b, ⟪g a, g b⟫_ℂ = ⟪a, b⟫_ℂ) {κ : ℂ} (hκ : ‖κ‖ = 1) {p p' q q' : X} (h : U p = κ • g q) (h' : U p' = κ • g q') : ⟪p, p'⟫_ℂ = ⟪q, q'⟫_ℂ

theorem **gram_of_conjAt** {ph : ℂ} {c₀ : VsV D i D.what₂.LL →ₛₗ[D.what₂.embLL] (R.cV i).W} (h : ConjAt R i σ ph c₀) {μ' η' : (AE D)ˣ →* Circle} {c : CoefA (Vc D i)} {c₁ : VsV D i D.what₂.LL →ₛₗ[σ] c.W} {H : HMod (Bridge.Gf (Vc D i) → c.W)} {j : A.fam.amb.FL i A.μ0 →ₛₗ[σ] H.H} {th : H.H →ₗ[ℂ] (R.HW i σ).H} (hd : IsConjDatum R i σ c₀ μ' η' c c₁ H j th) {x x' : A.fam.amb.FL i A.μ0} {y y' : Fin (R.nW i) → D.what₂.LL} (hy : IsRatLift R i ph x y) (hy' : IsRatLift R i ph x' y') : @inner ℂ (R.HW i σ).H _ ((R.FW i).j σ y) ((R.FW i).j σ y') = @inner ℂ (R.HW i σ).H _ (th (j x)) (th (j x'))

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.SigmaMonoInConstant

theorem **logArg_nonneg** (O : FunData K Df) {b : ℝ} (hb : 1 ≤ b) : 0 ≤ Real.log (b * O.nr * O.g1 * O.g2 * O.pd * O.pd' * |(O.dE : ℝ)|)

theorem **sigma_mono_in_constant** (O : FunData K Df) {C C' b : ℝ} (h : C ≤ C') (hb : 1 ≤ b) : O.Sig0 C b ≤ O.Sig0 C' b

theorem **monotonicity₂** (O : FunData K Df) {C C' b : ℝ} (h0 : 0 ≤ C) (h : C ≤ C') (hb : 1 ≤ b) : O.bFP C b ≤ O.bFP C' b

Development.AbelianVarietyTheory.Counterexamples.RotationExamples.SobolevContinuityLowGeneration

theorem **sobCont_zero** {n : ℕ} {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] (Lie : Set (Matrix (Fin n) (Fin n) ℂ)) (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)) : Sobolev₂ Lie σ 0

theorem **sobCont_smul** {n : ℕ} {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] {Lie : Set (Matrix (Fin n) (Fin n) ℂ)} {σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)} {ℓ : H →ₗ[ℂ] ℂ} (c : ℂ) (h : Sobolev₂ Lie σ ℓ) : Sobolev₂ Lie σ (c • ℓ)

theorem **sobCont_add** {n : ℕ} {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] {Lie : Set (Matrix (Fin n) (Fin n) ℂ)} {σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)} {ℓ ℓ' : H →ₗ[ℂ] ℂ} (h : Sobolev₂ Lie σ ℓ) (h' : Sobolev₂ Lie σ ℓ') : Sobolev₂ Lie σ (ℓ + ℓ')

theorem **mult_equiv_cont** {θ} {M : DSModel V S} {J χ} {U : MMap M} (hU : U ∈ Mult θ M J χ) : (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ), ∀ (v : M.Vec) (w : M.sm τ), (w : M.H τ) = M.rep τ g (v τ) → U (Function.update v τ w) = rTr V (atTau θ V τ g) (U v)) ∧ ∀ τ (v : M.Vec) (x : GA D V), ∃ ℓ : M.H τ →ₗ[ℂ] ℂ, Sobolev₂ (lieAt V τ) (M.rep τ) ℓ ∧ ∀ w : M.sm τ, ℓ w = U (Function.update v τ w) x

theorem **eq_zero_of_lowGenerates** {θ} {M : DSModel V S} (hM : ∀ τ, LowGenerates M τ) {J χ} {U : MMap M} (hU : U ∈ Mult θ M J χ) (h0 : ∀ v : ∀ ρ, lowT M ρ, U (fun ρ => (v ρ : M.sm ρ)) = 0) : U = 0

Development.AbelianVarietyTheory.Counterexamples.SubgroupCharacterSums.SumSubgroupEqZeroIff

theorem **isChar_iff** (χ : G → ℂ) : IsChar χ ↔ ∃ V : FDRep ℂ G, V.character = χ

theorem **norm_iff** [Fintype G] (V : FDRep ℂ G) : (∑ x, V.character x * star (V.character x)) = (Fintype.card G : ℂ) ↔ Simple V

theorem **isIrrChar_iff** [Fintype G] (χ : G → ℂ) : IsIrrChar χ ↔ ∃ V : FDRep ℂ G, Simple V ∧ V.character = χ

theorem **sum_subgroup_eq_zero_iff** [Fintype G] (V : FDRep ℂ G) (U : Subgroup G) [DecidablePred (· ∈ U)] : (∑ u : G, if u ∈ U then V.character u else 0) = 0 ↔ ∀ v : V, (∀ u ∈ U, V.ρ u v = v) → v = 0

theorem **zero_isChar** : IsChar (fun _ : G => (0 : ℂ))

theorem **zero_not_isIrrChar** [Fintype G] : ¬ IsIrrChar (fun _ : G => (0 : ℂ))

theorem **mem_unipBlock_iff** (d a : ℕ) (u : GL (Fin d) F) : u ∈ (unipBlock d a : Set (GL (Fin d) F)) ↔ ∀ i j : Fin d, ¬ (i.val < a ∧ a ≤ j.val) → ((u : Matrix (Fin d) (Fin d) F) − 1) i j = 0

theorem **block_mul_eq_zero** {d a : ℕ} {X Y : Matrix (Fin d) (Fin d) F} (hX : ∀ i j : Fin d, ¬ (i.val < a ∧ a ≤ j.val) → X i j = 0) (hY : ∀ i j : Fin d, ¬ (i.val < a ∧ a ≤ j.val) → Y i j = 0) : X * Y = 0

theorem **exists_mem_unipBlock** (d a : ℕ) (X : Matrix (Fin d) (Fin d) F) (hX : ∀ i j : Fin d, ¬ (i.val < a ∧ a ≤ j.val) → X i j = 0) : ∃ u ∈ (unipBlock d a : Set (GL (Fin d) F)), (u : Matrix (Fin d) (Fin d) F) = 1 + X

theorem **isCuspChar_iff** {d : ℕ} (χ : GL (Fin d) F → ℂ) : IsCuspChar χ ↔ ∃ V : FDRep ℂ (GL (Fin d) F), Simple V ∧ V.character = χ ∧ ∀ a, 0 < a → a < d → ∀ v : V, (∀ u ∈ (unipSubgroup d a : Subgroup (GL (Fin d) F)), V.ρ u v = v) → v = 0

theorem **isCuspChar_iff_of_le_one** {d : ℕ} (hd : d ≤ 1) (χ : GL (Fin d) F → ℂ) : IsCuspChar χ ↔ IsIrrChar χ

theorem **isIrrChar_of_hom** {G : Type} [Group G] [Fintype G] (c : G →* ℂˣ) : IsIrrChar (fun x => (c x : ℂ))

theorem **isCuspChar_rank_one** (c : GL (Fin 1) F →* ℂˣ) : IsCuspChar (fun x => (c x : ℂ))

theorem **not_isCuspChar_one** {d : ℕ} (hd : 2 ≤ d) : ¬ IsCuspChar (fun _ : GL (Fin d) F => (1 : ℂ))

theorem **isCuspChar_of_hom** {d : ℕ} (c : GL (Fin d) F →* ℂˣ) (hc : ∀ a, 0 < a → a < d → ∃ t ∈ (unipBlock d a : Set (GL (Fin d) F)), (c t : ℂ) ≠ 1) : IsCuspChar (fun x => (c x : ℂ))

theorem **transv_mem** : transv ∈ (unipBlock 2 1 : Set (GL (Fin 2) (ZMod 2)))

theorem **signChar_transv** : signChar transv = -1

theorem **isCuspChar_signChar** : IsCuspChar (fun x => (signChar x : ℂ))

Development.AbelianVarietyTheory.Counterexamples.ThetaValuation.CongruenceSpanLevel

theorem **gamAt_of_lawsM** {D : D7} (T : Core D) (h : LawsM T) : GamAt T

Development.AbelianVarietyTheory.Counterexamples.ThetaValuation.UniformMultiplier

theorem **eq_latP** {r : ℕ} (D : D7) (Λ L : Submodule (𝒪 D.host.E) (Fin r → D.host.E)) (h : Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ L, v = a • u}) : L = lattice₂ D Λ

Development.AbelianVarietyTheory.Counterexamples.TransversalityLocal.FramePairFixatorUnitaryAndFixesFrame

theorem **exists_transv** {α : Type*} (rel : α → α → Prop) (hr : Equivalence rel) (S : Set α) : ∃ R, IsTransv rel S R

theorem **img_mul** (g g' : V ≃ₗ[E] V) (Λ : Submodule M.L.O V) : img (g * g') Λ = img g (img g' Λ)

theorem **img_one** (Λ : Submodule M.L.O V) : img (1 : V ≃ₗ[E] V) Λ = Λ

theorem **trel_equiv** (P : FPair H r) : Equivalence (TRel P) where

theorem **framePairFixator_unitary_and_fixes_frame** (P : FPair H r) {g : V ≃ₗ[E] V} (hg : g ∈ P.UW0) : IsUnitary H g ∧ ∀ b, g (P.A.f b) = P.A.f b ∧ g (P.B.f b) = P.B.f b

theorem **perp_stable** {g : V ≃ₗ[E] V} (hu : IsUnitary H g) (f : Fin (r + 1) → V) (hf : ∀ b, g (f b) = f b) {v : V} (hv : v ∈ H.perp (H.X f)) : g v ∈ H.perp (H.X f)

theorem **rel_equiv** (P : FPair H r) : Equivalence (Datum.Rel (P

theorem **reps_nonempty** (P : FPair H r) : Nonempty (Reps P)

theorem **wt_self** {P : FPair H r} (d : Datum P) : Datum.wt d d = 1

theorem **X_zero** (f : Fin 1 → V) : H.X (r

theorem **mem_perp_zero** (f : Fin 1 → V) (v : V) : v ∈ H.perp (H.X (r

theorem **mkW_inj** (f : Fin 1 → V) : Function.Injective (H.mkW (r

theorem **stdH_single** (x : Fin (m + 2) → E) (i) : (stdH M m).h x (Pi.single i 1) = x i

theorem **vec_zero** (c : F) : vec (E

theorem **vec_one** (c : F) : vec (E

theorem **h_vec** (c c' : F) : (stdH M m).h (vec m c) (vec m c') = algebraMap F E (1 + c * c')

theorem **theta_int** : theta M ∈ 𝒪[F]

theorem **theta_ne** : theta M ≠ 0

theorem **theta_some** {g} (h : M.split = some g) : theta M = (g.q : F)

theorem **unit_some** (g : SplitG M.L) : ∃ v : 𝒪[F], (v : F) * -(1 + (g.q : F) * g.q) = 1

theorem **D0** : (pair M m).Dm 0 = 1

theorem **D1** : (pair M m).Dm 1 = 1

theorem **relG** [CharZero F] : (pair M m).RelG (theta M)

theorem **cj_mem** {x : E} (hx : x ∈ M.L.O) : M.L.cj x ∈ M.L.O

theorem **alg_mem** {c : F} (hc : c ∈ 𝒪[F]) : algebraMap F E c ∈ M.L.O

theorem **mem_inn** (c : F) (x : (stdH M m).perp ((stdH M m).X (r

theorem **vec_mem** {c : F} (hc : c ∈ 𝒪[F]) (i) : vec (E

theorem **isInner** {c : F} (hc : c ∈ 𝒪[F]) : (frame M m c).IsInner (inn M m c)

theorem **lamStd** (c : F) : (frame M m c).LamStd (inn M m c) = Submodule.span M.L.O {v | ∀ i, v i ∈ M.L.O}

theorem **isTermOver** : IsTermOver (pair M m) (datum M m) (term M m)

Development.AbelianVarietyTheory.Counterexamples.TwistSurvey.ActsUnitarilyOfClauses

theorem **actsUnitarily_of_clauses** (R : RealDat D A Ks) (hE : R.models_HE) (gE : R.models_gE) (h0 : R.models_ν0) (hW : R.models_HW) (gW : R.models_gW) (hν : R.models_νW) (ht : R.fam_tie) (hX : R.models_cX) (hV : R.models_cV) : ActsUnitarily R

theorem **unitaryAction** : UnitaryAction

theorem **geometric_unitary** (R : RealDat D A Ks) (hU : R.LUnif) (hW : R.models_HW) (gW : R.models_gW) (h : ActsUnitarily R) (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ) (g : G) (x y : Fin (R.nW i) → D.what₂.LL) : inner ℂ ((R.FW i).j σ ((R.FW i).ρ g x)) ((R.FW i).j σ ((R.FW i).ρ g y)) = inner ℂ ((R.FW i).j σ x) ((R.FW i).j σ y)

theorem **not_isUnitaryAt_scalars** (σ : ℚ →+* ℂ) : ¬ IsUnitaryAt lineDatum σ (Representation.ofDistribMulAction ℚ ℚˣ lineDatum.Forms)

Development.AbelianVarietyTheory.Counterexamples.TwistSurvey.CarriesOfKilled

theorem **apply_eq_smul_of_unique_eigenvalue** {M : Type} [AddCommGroup M] [Module ℂ M] (f : M → M) (h0 : f 0 = 0) (P : Submodule ℂ M) (m : ℝ) (h : ∀ l : ℝ, P ≠ ⊥ → (∀ x ∈ P, f x = (l : ℂ) • x) → m = l) (hA : ∃ l : ℝ, ∀ x ∈ P, f x = (l : ℂ) • x) : ∀ x ∈ P, f x = (m : ℂ) • x

theorem **scal_of_one** {M C : Type} [AddCommGroup M] [Module ℂ M] (f : M → M) (h0 : f 0 = 0) (P : C → Submodule ℂ M) (k : C → Prop) (h1 : ∀ c c', k c → k c' → P c = P c') (hA : ∀ c, P c ≠ ⊥ → ∃ l : ℝ, ∀ x ∈ P c, f x = (l : ℂ) • x) : ∃ l : ℝ, ∀ x ∈ ⨆ (c : C) (_ : k c), P c, f x = (l : ℂ) • x

theorem **iff_of_pin** (R : RealDat3 D A3 Ks) (h : R.eigenvalue_at_base_of_scalar) : R.PThmA ↔ ∀ i t p, ∃ l : ℝ, ∀ x ∈ R.Wp3 i p, R.thG i t (LinearMap.adjoint (R.thG i t) x) = (l : ℂ) • x

theorem **ex_of_aForm** {A : FamIn D.what₂.LL G Pl0 T D.R.m} {R : RealDat D A Ks} {SF} (hA : GlobalTheory.LiftRows.AForm R SF) (i t c) (hn : GlobalTheory.Hk3.Wcc R SF i c ≠ ⊥) : ∃ l : ℝ, ∀ x ∈ GlobalTheory.Hk3.Wcc R SF i c, R.thG i t (LinearMap.adjoint (R.thG i t) x) = (l : ℂ) • x

theorem **scalC_of_aForm** (R : RealDat3 D A3 Ks) (hA : GlobalTheory.LiftRows.AForm R.toRealDat R.SF) : ScalC R

theorem **of_scalC** (R : RealDat3 D A3 Ks) (hA : ScalC R) (h : R.eigenvalue_at_base_of_scalar) (h1 : OnePart R) : R.PThmA

theorem **of_aForm** (R : RealDat3 D A3 Ks) (hA : GlobalTheory.LiftRows.AForm R.toRealDat R.SF) (h : R.eigenvalue_at_base_of_scalar) (h1 : OnePart R) : R.PThmA

Development.AbelianVarietyTheory.Counterexamples.TwistSurvey.ConjModelCast

theorem **conj_conj** {K : Type} [Field K] (f : K →+* ℂ) : (starRingEnd ℂ).comp ((starRingEnd ℂ).comp f) = f

theorem **emb_mul** {R : Ranks} (H : Host R) (g g' : R.GG) : H.emb (g * g') = (H.emb g).comp (H.sigmaOf g' : H.E →+* H.E)

theorem **conj_not_mem** (D : D7) (ρ : D.what₂.Tau) (g : D.R.GG) (h : D.host.emb g = (starRingEnd ℂ).comp (tauE D ρ)) : g ∉ D.R.Psi

theorem **tauE_ne_conj** (D : D7) (ρ : D.what₂.Tau) : tauE D ρ ≠ (starRingEnd ℂ).comp (tauE D ρ)

theorem **lab_exists** {D : D7} (σ : D.what₂.LL →+* ℂ) (ρ : D.what₂.Tau) : ∃ τ, labE σ τ = tauE D ρ ∨ labE σ τ = (starRingEnd ℂ).comp (tauE D ρ)

theorem **labE_conj** {D : D7} (σ : D.what₂.LL →+* ℂ) (τ : D.what₂.Tau) : labE ((starRingEnd ℂ).comp σ) τ = (starRingEnd ℂ).comp (labE σ τ)

theorem **lab_conj** {D : D7} (σ : D.what₂.LL →+* ℂ) (ρ : D.what₂.Tau) : lab ((starRingEnd ℂ).comp σ) ρ = lab σ ρ

theorem **parAt_conj** {D : D7} (W : HSp D) (σ : D.what₂.LL →+* ℂ) : Links.parAt W ((starRingEnd ℂ).comp σ) = fun ρ => dualPar (Links.parAt W σ ρ)

theorem **not_good_conj** {σ : D.what₂.LL →+* ℂ} (h : GoodEmb T.Q.c (res T σ)) : ¬ GoodEmb T.Q.c (res T ((starRingEnd ℂ).comp σ))

theorem **premises_met** (hl : LawsJ T) (i : Fin D.R.m) : ∃ σ : D.what₂.LL →+* ℂ, ¬ GoodEmb T.Q.c (res T σ) ∧ T.V ((i : ℕ) + 1) (resC T σ) = Wc D i

theorem **branch** (ha : AlgTie T) (hp : T.Q.PinField) (hr : ∀ τ, tauE D τ (T.Q.H.rt : D.host.E) = T.Q.H.rt0) (σ : D.what₂.LL →+* ℂ) (ρ : D.what₂.Tau) : GoodEmb T.Q.c (res T σ) ↔ labE σ (lab σ ρ) = tauE D ρ

theorem **branch_of_pins** (ha : AlgTie T) (hp : T.Q.Pins) (σ : D.what₂.LL →+* ℂ) (ρ : D.what₂.Tau) : GoodEmb T.Q.c (res T σ) ↔ labE σ (lab σ ρ) = tauE D ρ

Development.AbelianVarietyTheory.Counterexamples.TwistSurvey.ScalAtZeroOfLiftMat

theorem **maj_eq_zero** {n : ℕ} (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ) (v : InfinitePlace K) (x y : Fin n → K) (h : Formalism.HermitianLatticesTheory.majAt (Pi.basisFun K _) X v x x = 0) : Formalism.HermitianLatticesTheory.majAt (Pi.basisFun K _) X v x y = 0

Development.AbelianVarietyTheory.Counterexamples.TwistSurvey.SubsingletonOfEvInj

theorem **evAt_injective_of_ne** {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (hm : R.models_HW) (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ) (v : (R.MW i σ).Vec) (hv : ∀ τ, (v τ : (R.MW i σ).H τ) ≠ 0) : Function.Injective (evAt R i σ v)

Development.AbelianVarietyTheory.Counterexamples.UniformisationSurvey.BoxCubicalUnconditional

theorem **hom_ext** {G' G : Over S} [GrpObj G'] [GrpObj G] {j : ℕ} (p : IsPowGrp G' G j) {T : Over S} {x y : T ⟶ G'} (h : ∀ i, x ≫ p.pr i = y ≫ p.pr i) : x = y

theorem **comm_of_pow** {G' G : Over S} [GrpObj G'] [GrpObj G] [IsCommMonObj G] {j : ℕ} (p : IsPowGrp G' G j) : IsCommMonObj G' where

theorem **pow_unique** {G₁ G₂ G : Over S} [GrpObj G₁] [GrpObj G₂] [GrpObj G] {j : ℕ} (p : IsPowGrp G₁ G j) (q : IsPowGrp G₂ G j) : ∃ e : G₁ ≅ G₂, IsMonHom e.hom ∧ ∀ i, e.hom ≫ q.pr i = p.pr i

Development.AbelianVarietyTheory.Counterexamples.UniformisationSurvey.EquivIffOfIsEmpty

theorem **over_of_isPull** {T T' : Over (Spec (CommRingCat.of R))} {h : T'.left ⟶ T.left} {g : GtPt D T} {g' : GtPt D T'} (H : GtPt.IsPull h g g') : h ≫ T.hom = T'.hom

theorem **generic_maps_eq** (W : (Spec (CommRingCat.of R)).Opens) (χ : D.X) : (jK R ∣_ W) ≫ (toUnit (TW W) ≫ D.c χ).left = (jK R ⁻¹ᵁ W).ι ≫ ptK (D.c χ)

theorem **ext_val** {T : Over (Spec (CommRingCat.of R))} {g g' : GtPt D T} (hb : g.b = g'.b) (hs : ∀ χ y J, Geo.Tor.Good D.P.L y J → Geo.Tor.val (g.s χ) (𝟙 _) y J = Geo.Tor.val (g'.s χ) (𝟙 _) y J) : g = g'

theorem **isAdd_unique** {T : Over (Spec (CommRingCat.of R))} {g g' a a' : GtPt D T} (h : g.IsAdd g' a) (h' : g.IsAdd g' a') : a = a'

theorem **isPull_unique** {T T' : Over (Spec (CommRingCat.of R))} {h : T'.left ⟶ T.left} {g : GtPt D T} {a a' : GtPt D T'} (H : GtPt.IsPull h g a) (H' : GtPt.IsPull h g a') : a = a'

theorem **isIota_unique** {W : (Spec (CommRingCat.of R)).Opens} (ψW : PsiOn D W) {T : Over W.toScheme} {y : D.X} {a a' : GtPt D (overW W T)} (H : GtPt.IsIotaW ψW y a) (H' : GtPt.IsIotaW ψW y a') : a = a'

theorem **isNsmul_unique** {T : Over (Spec (CommRingCat.of R))} {k : ℕ} {g a a' : GtPt D T} (h : GtPt.IsNsmul k g a) (h' : GtPt.IsNsmul k g a') : a = a'

theorem **corr_of_isEmpty** {X Y : Scheme.{0}} [IsEmpty Y] (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) : LB.Gen.Corr L φ ε φ' ε' t t'

theorem **hom_eq** {A : Over (Spec (CommRingCat.of R))} (a b : T ⟶ A) : a = b

theorem **gtPt_eq** (g g' : GtPt D T) : g = g'

theorem **equiv_iff_of_isEmpty** (p p' : QkPtW D W ψW k T) : QkPtW.Equiv p p' ↔ ∃ z : D.X, p'.y = p.y + k • z

theorem **not_equiv_over_empty** {W : (Spec (CommRingCat.of InhTate.R)).Opens} (ψW : PsiOn InhTate.tatePre W) : ∃ (T : Over W.toScheme) (_ : IsEmpty T.left) (p p' : QkPtW InhTate.tatePre W ψW 2 T), ¬ QkPtW.Equiv p p'

Development.AbelianVarietyTheory.Counterexamples.UniformisationSurvey.MapPowLeOfIsPullUnif

theorem **_root_.GlobalTheory.UnifPullGuards.unifBaseChange_holds** : _root_.ReciprocityTheory.ChartRestriction.G.Out.UnifBaseChangeSt

Development.AbelianVarietyTheory.Counterexamples.UniformisationSurvey.NeTopOfAdmissibleBase

theorem **torusEndOf_iff** (h : X → X) (t : splitTorus S X ⟶ splitTorus S X) : TorusEndOf X h t ↔ ∀ χ, t.left.appTop.hom (charUnit S X χ).val = (charUnit S X (h χ)).val

theorem **torusEndOf_unique** {h : X → X} {t t' : splitTorus S X ⟶ splitTorus S X} (ht : TorusEndOf X h t) (ht' : TorusEndOf X h t') : t = t'

theorem **torusEndOf_comp** {h h' : X → X} {t t' : splitTorus S X ⟶ splitTorus S X} (ht : TorusEndOf X h t) (ht' : TorusEndOf X h' t') : TorusEndOf X (h ∘ h') (t ≫ t')

Development.AbelianVarietyTheory.Counterexamples.UniqueMatching.ExistsUniqueMatch

theorem **mono_j** (Y : Model Xo c β) [Mono (spMap β)] : Mono Y.j

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.CoordDifferentiableAt

theorem **one_le_tt_iff** (X : Data E r) (ξ : Fin r → E) (x : Frame X.Ψ r) (τ : X.Ψ) : 1 ≤ X.tt ξ x τ ↔ 0 < X.c τ ∗ X.maj x τ (Data.emb τ.1 ξ) ∧ X.c τ ∗ X.maj x τ (Data.emb τ.1 ξ) ≤ 1

theorem **cfwM_all_eta** (X : Data E r) {ps : Frame X.Ψ r → ℝ} {a : ℝ} (h : cfwM X ps a) (ha : 0 ≤ a) (η : ℝ) (hη : 0 < η) : ∃ C A' : ℝ, ∀ x ∈ X.dom, ∀ ξ : Fin r → E, X.IsIso ξ → (∀ τ, 1 ≤ X.tt ξ x τ) → ps x ≤ C + A' ∗ Real.log (1 + 1 / X.la1 x) − 8 ∗ Real.pi ∗ (1 − 2 ∗ η) ∗ a ∗ ⊓ τ, X.tt ξ x τ

theorem **sum_le_of_mem_multiLt** {n a0 T : ℕ} (h : a0 ≤ T + 1) {α : Fin n → ℕ} (hα : α ∈ ReciprocityTheory.ChartRestriction.G.AR.multiLt n a0) : (∑ q, α q) + (T − ∑ q, α q) = T

theorem **coord_differentiableAt** (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (j : Fin Fl.δ) : DifferentiableAt ℂ (fun z : V (nd X) => (WithLp.toLp 2 fun k => Fl.s k (cpt X (Fl.xs j) z) / Fl.w0 (cpt X (Fl.xs j) z) : V (nd X))) 0

theorem **exists_s_ne_of_empty** (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (h : Fl.δ = 0) : ∀ x ∈ X.dom, ∃ k, Fl.s k x ≠ 0

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.ExistsFundSetOrth

theorem **exists_fundSet_orth** {x0 : Frame X.Ψ r} (ho : IsOrth X x0) : ∃ FF, IsFundSet X x0 FF

theorem **muF_ne_zero** {x0 : Frame X.Ψ r} (ho : IsOrth X x0) : muF X x0 Set.univ ≠ 0

theorem **fundSet_pos** {x0 : Frame X.Ψ r} (ho : IsOrth X x0) {FF : Set (V (nd X))} (hF : IsFundSet X x0 FF) : muF X x0 FF ≠ 0

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.FirstMinimumRankZero

theorem **c_pos** (τ : X.Ψ) : 0 < X.c τ

theorem **tt_pos** {x : Frame X.Ψ r} (hx : x ∈ X.dom) {ξ : Fin r → E} (hξ : ξ ≠ 0) (τ : X.Ψ) : 0 < X.tt ξ x τ

theorem **scaled_maj_flow** {x : Frame X.Ψ r} (hx : x ∈ X.dom) {ξ : Fin r → E} (hξ : X.IsIso ξ) (s : X.Ψ → ℂ) (hs : X.flow ξ s x ∈ X.dom) (τ : X.Ψ) : X.c τ ∗ X.maj (X.flow ξ s x) τ (Data.emb τ.1 ξ) = 1 / (X.tt ξ x τ + (s τ).im)

theorem **flowMat_eq_transv** (ξ : Fin r → E) (τ : X.Ψ) (s : E) : X.flowMat ξ τ (τ.1 s) = (transvMat X.Gm ξ (s ∗ X.κ0)).map τ.1

theorem **first_minimum_rank_zero** (Y : Data E 0) (x : Frame Y.Ψ 0) : Y.la1 x = 0

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.HasColenNotUniqueWithoutGuard

theorem **vdet_le_of_span_le** (v : Valuation C NNReal) (hv : ∀ r : R, v (algebraMap R C r) ≤ 1) {X : Type} [Fintype X] [DecidableEq X] (B B' : Matrix X X C) (h : Submodule.span R (Set.range fun j x => B x j) ≤ Submodule.span R (Set.range fun j x => B' x j)) : v B.det ≤ v B'.det

theorem **hasColen_unique** (v : Valuation C NNReal) (hv : ∀ r : R, v (algebraMap R C r) ≤ 1) {X : Type} [Fintype X] [DecidableEq X] {M : Submodule R (X → C)} {c c' : NNReal} (h : HasColen R v M c) (h' : HasColen R v M c') : c = c'

theorem **intFun_eq_span** {X : Type} [Fintype X] [DecidableEq X] : intFun (C

theorem **hasColen_intFun** (v : Valuation C NNReal) {X : Type} [Fintype X] [DecidableEq X] : HasColen R v (intFun (C

theorem **hasColen_not_unique_without_guard** (v : Valuation C NNReal) (t : C) (ht : t ≠ 0) : HasColen C v (⊤ : Submodule C (Unit → C)) (v t)

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.MemSSgtOfZeroOnDom

theorem **fj_congr** (la : Wt X.Ψ r) {ξ : Fin r → E} (hξ : X.IsIso ξ) (ν : X.Ψ → ℝ) {P Q : Frame X.Ψ r → ℂ} (h : ∀ x ∈ X.dom, P x = Q x) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : X.fj la ξ ν P x = X.fj la ξ ν Q x

theorem **fj_one** (la : Wt X.Ψ r) (ξ : Fin r → E) (x : Frame X.Ψ r) : X.fj la ξ 0 (fun _ => 1) x = 1

theorem **chi_zero** {γ : GL (Fin r) E} (hγ : γ ∈ X.Γ) : X.χ 0 γ = 1

theorem **one_isBddSec** : X.IsBddSec 0 (fun _ => 1)

theorem **one_not_mem_SS** {ξ : Fin r → E} (hξ : X.IsIso ξ) (hd : X.dom.Nonempty) : (fun _ => 1) ∉ X.SS 0

theorem **SSgt_eq_SS** (la : Wt X.Ψ r) {c : ℝ} (hc : c ≤ 0) : X.SSgt la c = X.SS la

theorem **eq_zero_of_fj_zero** {la : Wt X.Ψ r} {P : Frame X.Ψ r → ℂ} (hP : X.IsSec la P) {ξ : Fin r → E} (hξ : X.IsIso ξ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (h : ∀ ν, X.fj la ξ ν P x = 0) (s : X.Ψ → ℂ) (hs : X.flow ξ s x ∈ X.dom) : P (X.flow ξ s x) = 0

theorem **zero_not_mem_dom** (hr : 0 < r) : (0 : Frame X.Ψ r) ∉ X.dom

theorem **mem_SSgt_of_zero_on_dom** (la : Wt X.Ψ r) (c : ℝ) {P : Frame X.Ψ r → ℂ} (h : ∀ x ∈ X.dom, P x = 0) : P ∈ X.SSgt la c

theorem **exists_ne_zero_mem_SSgt** (hr : 0 < r) (la : Wt X.Ψ r) (c : ℝ) : ∃ P ∈ X.SSgt la c, P ≠ 0 ∧ ∀ x ∈ X.dom, P x = 0

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.NonemptyAlgebra

theorem **_root_.GlobalTheory.AinfChecks.padicComplex** (p : ℕ) [Fact p.Prime] : ∃ (_ : Fact ¬IsUnit ((p : ℕ) : 𝒪_ℂ_[p])) (_ : IsAdicComplete (span {((p : ℕ) : 𝒪_ℂ_[p])}) 𝒪_ℂ_[p]), (∃ pflat : PreTilt 𝒪_ℂ_[p] p, pflat.untilt = ((p : ℕ) : 𝒪_ℂ_[p])) ∧ (∃ ε : PreTilt 𝒪_ℂ_[p] p, IsPrimitiveRoot (((frobeniusEquiv (PreTilt 𝒪_ℂ_[p] p) p).symm ε).untilt) p) ∧ _root_.ReciprocityTheory.PGsh.KerThetaStatement p 𝒪_ℂ_[p] ∧ _root_.ReciprocityTheory.PGsh.XiNuStatement p ℂ_[p] Valued.v 𝒪_ℂ_[p] ∧ _root_.ReciprocityTheory.PGsh.RegularStatement p 𝒪_ℂ_[p] ∧ _root_.ReciprocityTheory.PGsh.FlatW'Statement p 𝒪_ℂ_[p]

theorem **nonempty_algebra** (pflat : PreTilt 0 p) : Nonempty (Algebra (ZMod (p ^ 2)) (WittVector p (PreTilt 0 p)) ⧸ span {(p : WittVector p (PreTilt 0 p)) ^ 2, teichmuller p pflat})

theorem **not_regular_zmod** (p : ℕ) [Fact p.Prime] : ∃ (_ : Fact ¬IsUnit ((p : ℕ) : ZMod p)) (_ : IsAdicComplete (span {((p : ℕ) : ZMod p)}) (ZMod p)), ¬ RegularStatement p (ZMod p)

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.SumPnormMonAll

theorem **mon_sum** (x : Frame Ψ 2) : ‖mon τ₁ τ₂ 0 0 x‖ ^ 2 + ‖mon τ₁ τ₂ 0 1 x‖ ^ 2 + ‖mon τ₁ τ₂ 1 0 x‖ ^ 2 + ‖mon τ₁ τ₂ 1 1 x‖ ^ 2 = colSq x τ₁ * colSq x τ₂

theorem **mon_le** (a b : Fin 2) (x : Frame Ψ 2) : ‖mon τ₁ τ₂ a b x‖ ^ 2 ≤ colSq x τ₁ * colSq x τ₂

theorem **mon_rmul** (a c : Fin 2) (x : Frame Ψ 2) {b : Frame Ψ 2} (hb : b ∈ Bor Ψ 2) : mon τ₁ τ₂ a c (rmul x b) = b τ₁ 0 0 * b τ₂ 0 0 * mon τ₁ τ₂ a c x

theorem **mon_diff** (a b : Fin 2) : Differentiable ℂ (mon τ₁ τ₂ a b)

theorem **colSq_rmul** (x : Frame Ψ 2) {b : Frame Ψ 2} (hb : b ∈ Bor Ψ 2) (τ : Ψ) : colSq (rmul x b) τ = ‖b τ 0 0‖ ^ 2 * colSq x τ

theorem **mon_pow_mem** (hG : X.Gm = 1) (hΓ : X.Γ = ⊥) (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) {la : Wt X.Ψ 2} {m : ℕ} (h0 : ∀ τ, la τ 0 = m) (h1 : ∀ τ, la τ 1 = 0) (a b : Fin 2) (c : ℝ) : (fun x => mon τ₁ τ₂ a b x ^ m) ∈ X.SSgt la c

theorem **mon_mem** (hG : X.Gm = 1) (hΓ : X.Γ = ⊥) (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) (a b : Fin 2) (c : ℝ) : mon τ₁ τ₂ a b ∈ X.SSgt (wtOne X.Ψ) c

theorem **pnorm_one** (hG : X.Gm = 1) (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) {x : Frame X.Ψ 2} (hx : x ∈ X.dom) (P : Frame X.Ψ 2 → ℂ) : X.pnorm (wtOne X.Ψ) P x ^ 2 = ‖P x‖ ^ 2 / (colSq x τ₁ * colSq x τ₂)

theorem **sum_pnorm_monAll** (hG : X.Gm = 1) (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) {x : Frame X.Ψ 2} (hx : x ∈ X.dom) : ∑ k, X.pnorm (wtOne X.Ψ) (monAll τ₁ τ₂ k) x ^ 2 = 1

theorem **nd_eq_two** (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) : nd X = 2

theorem **idF_mem** (hG : X.Gm = 1) (hp : X.p0 = 2) : idF X.Ψ ∈ X.dom

theorem **swF_mem** (hG : X.Gm = 1) (hp : X.p0 = 2) : swF X.Ψ ∈ X.dom

theorem **idx_cases** (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) (k : Fin (nd X)) : k = idx X τ₁ ∨ k = idx X τ₂

theorem **secS_fst** : secS τ₁ τ₂ (idx X τ₁) = mon τ₁ τ₂ 1 0

theorem **secS_snd** (hne : τ₁ ≠ τ₂) : secS τ₁ τ₂ (idx X τ₂) = mon τ₁ τ₂ 0 1

theorem **sum_pnorm_secS** (hG : X.Gm = 1) (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) {x : Frame X.Ψ 2} (hx : x ∈ X.dom) : ∑ k, X.pnorm (wtOne X.Ψ) (secS τ₁ τ₂ k) x ^ 2 = (‖mon τ₁ τ₂ 1 0 x‖ ^ 2 + ‖mon τ₁ τ₂ 0 1 x‖ ^ 2) / (colSq x τ₁ * colSq x τ₂)

theorem **secW_le** (x : Frame X.Ψ 2) : ‖secW τ₁ τ₂ x‖ ^ 2 ≤ 2 * (colSq x τ₁ * colSq x τ₂)

theorem **le_secW** (x : Frame X.Ψ 2) : colSq x τ₁ * colSq x τ₂ ≤ 2 * (‖secW τ₁ τ₂ x‖ ^ 2 + (‖mon τ₁ τ₂ 1 0 x‖ ^ 2 + ‖mon τ₁ τ₂ 0 1 x‖ ^ 2))

theorem **secW_mem** (hG : X.Gm = 1) (hΓ : X.Γ = ⊥) (hne : τ₁ ≠ τ₂) (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) (c : ℝ) : secW τ₁ τ₂ ∈ X.SSgt (wtOne X.Ψ) c

theorem **zero_iff** (hall : ∀ τ, τ = τ₁ ∨ τ = τ₂) {x : Frame X.Ψ 2} (hx : x ∈ X.dom) : (mon τ₁ τ₂ 1 0 x = 0 ∧ mon τ₁ τ₂ 0 1 x = 0) ↔ ∃ j, ∃ b ∈ Bor X.Ψ 2, x = rmul (reps X j) b

theorem **det_fderiv_ne** {n : ℕ} {σ : Fin n → Fin n} (hσ : Function.Surjective σ) (i j : Fin n) : (fderiv ℂ (fun z : V n => (WithLp.toLp 2 fun k => z (σ k) / (1 + z i * z j) : V n)) 0).det ≠ 0

theorem **leviL_nonneg** (X : Data E r) (Xa : ArchInM X) (x0 : Frame X.Ψ r) (ho : IsOrth X x0) (z : V (nd X)) (hz : z ∈ OmF X x0) (ξ : Fin (nd X) → ℂ) : 0 ≤ leviO X x0 (phla X Xa.L) z ξ

theorem **leviRh_definite** (X : Data E r) (hp : X.p0 = 0 ∨ X.p0 = r) (x0 : Frame X.Ψ r) (ho : IsOrth X x0) (z : V (nd X)) (ξ : Fin (nd X) → ℂ) : leviRh X x0 z ξ = 0

theorem **margins_definite** (X : Data E r) (hp : X.p0 = 0 ∨ X.p0 = r) (Xa : ArchInM X) : Margin X Xa ∧ MarginC X Xa

theorem **model** (h : Module.finrank ℚ E = 4) : ∃ (X : Data E 2) (Xa : ArchInM X) (Fl : FlatM X Xa), Fl.δ = 2 ∧ PropDisc X ∧ Margin X Xa ∧ MarginC X Xa ∧ (∃ C : ℝ, ∀ x ∈ X.dom, Xa.ps0 x ≤ C) ∧ (∀ P, X.IsBddSec Xa.Fp P → P ∈ X.SS Xa.Fp) ∧ ModeHalfStatement X ∧ ∃ F ∈ X.SSgt Xa.Fp Xa.c1, ∃ j j', F (Fl.xs j) = 1 ∧ F (Fl.xs j') = 0

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.ValOfKindIndep

theorem **exists_value_top** [NeZero d] (X : RowData C V d I J) : ∃ Y : RowValue X, Y.nu = ⊤

theorem **no_equal_types** (G : ∀ g, Cart V d (I g) (J g)) (alt : Prop) {N : ℕ} (P : MvPolynomial (Fin N × WV I) C) (kind : Fin N → Option (Fin 2 × ZMod d × ZMod d)) (h : RowsOK G alt P kind) (ρ : Fin N) (μ μ' : Fin N × WV I →₀ ℕ) (hμ : μ ∈ P.support) (hμ' : μ' ∈ P.support) (w w' : WV I) (hw : μ (ρ, w) ≠ 0) (hw' : μ' (ρ, w') ≠ 0) (hne : w.1 ≠ w'.1) : alt ∧ tyW G w ≠ tyW G w'

theorem **Agree.mul** {σ : Type} {S : Set σ} {F F' K K' : MvPowerSeries σ C} (h : Agree S F F') (h' : Agree S K K') : Agree S (F * K) (F' * K')

theorem **Agree.pow** {σ : Type} {S : Set σ} {F F' : MvPowerSeries σ C} (h : Agree S F F') (k : ℕ) : Agree S (F ^ k) (F' ^ k)

theorem **Agree.exp** {σ : Type} {S : Set σ} {F F' : MvPowerSeries σ C} (h : Agree S F F') : Agree S (mvExp F) (mvExp F')

theorem **Agree.prod** {σ α : Type} {S : Set σ} (s : Finset α) {F F' : α → MvPowerSeries σ C} (h : ∀ i, Agree S (F i) (F' i)) : Agree S (∏ i ∈ s, F i) (∏ i ∈ s, F' i)

theorem **agree_rowCopy** {N : ℕ} (S : Set (Fin N × WV I)) (ρ : Fin N) (F F' : MvPowerSeries (WV I) C) (h : Agree {w | (ρ, w) ∈ S} F F') : Agree S (rowCopy ρ F) (rowCopy ρ F')

theorem **valOf_eq** (hL : Fin 2 → ZMod d → ZMod d → MvPowerSeries (WV I) C) {N : ℕ} (P : MvPolynomial (Fin N × WV I) C) (kind : Fin N → Option (Fin 2 × ZMod d × ZMod d)) (gr : Fin N → MvPowerSeries (WV I) C) : ValOf hL P kind gr = jetAt0 P (∏ ρ, (rowCopy ρ (gr ρ) * mvExp (rowCopy ρ (hOf hL (kind ρ)))))

theorem **agree_hOf_zero** [NeZero d] (X : RowData C V d I J) (T : Set (WV I)) (g0 : Fin 2) (hT : ∀ w ∈ T, w.1 = g0) (k : Option (Fin 2 × ZMod d × ZMod d)) : Agree T (hOf X.hL k) 0

theorem **hOf_mixed** [NeZero d] (X : RowData C V d I J) (w0 w1 : WV I) (h0 : w0.1 = 0) (h1 : w1.1 = 1) (g : Fin 2) (a a' : ZMod d) (l0 : OnLeafV X.G g a a' w0) (l1 : OnLeafV X.G g a a' w1) : X.hL g a a' = X.hL 0 (tyW X.G w0) (tyW X.G w1)

theorem **valOf_kind_indep** [NeZero d] (X : RowData C V d I J) {N : ℕ} (P : MvPolynomial (Fin N × WV I) C) (k k' : Fin N → Option (Fin 2 × ZMod d × ZMod d)) (hk : RowsOK X.G X.Alt P k) (hk' : RowsOK X.G X.Alt P k') (gr : Fin N → MvPowerSeries (WV I) C) : ValOf X.hL P k gr = ValOf X.hL P k' gr

Development.AbelianVarietyTheory.Counterexamples.UpperHalfSurvey.WeightEqZeroOfOneIsSec

theorem **dom_nonempty** (X : Data E r) : X.dom.Nonempty

theorem **dmin_ne_zero** (X : Data E r) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (τ : X.Ψ) (i : Fin (r + 1)) : dmin X.Gm x τ i ≠ 0

theorem **pnorm_eq_zero_iff** (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : X.pnorm la P x = 0 ↔ P x = 0

theorem **chi_zero** (X : Data E r) {γ : GL (Fin r) E} (hγ : γ ∈ X.Γ) : X.χ 0 γ = 1

theorem **one_isSec** (X : Data E r) : X.IsSec 0 (fun _ => 1) ∧ ∀ x, X.pnorm 0 (fun _ => 1) x = 1

theorem **weight_eq_zero_of_one_isSec** (X : Data E r) (la : Wt X.Ψ r) (h : X.IsSec la (fun _ => 1)) : la = 0

theorem **isSec_congr_on_dom** (X : Data E (r + 1)) (la : Wt X.Ψ (r + 1)) (P : Frame X.Ψ (r + 1) → ℂ) (hP : X.IsSec la P) (Q : Frame X.Ψ (r + 1) → ℂ) (hB : ∀ x ∈ X.dom, ∀ b ∈ Bor X.Ψ (r + 1), rmul x b ∈ X.dom) (hΓ : ∀ γ ∈ X.Γ, ∀ x ∈ X.dom, lact γ x ∈ X.dom) (hQ : ∀ x ∈ X.dom, Q x = P x) : X.IsSec la Q

Development.AbelianVarietyTheory.CrystallinePeriods.AdicMaps

theorem **levMap_mk** (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (n : ℕ) (r : R) : levMap I J f hf n (Ideal.Quotient.mk _ r) = Ideal.Quotient.mk _ (f r)

theorem **factorPow_evalₐ** (I : Ideal R) {m n : ℕ} (hle : m ≤ n) (x : AdicCompletion I R) : Ideal.Quotient.factorPow I hle (AdicCompletion.evalₐ I n x) = AdicCompletion.evalₐ I m x

theorem **levMap_compat** (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) {m n : ℕ} (hle : m ≤ n) : (Ideal.Quotient.factorPow J hle).comp ((levMap I J f hf n).comp (AdicCompletion.evalₐ I n : AdicCompletion I R →+* R ⧸ I ^ n)) = (levMap I J f hf m).comp (AdicCompletion.evalₐ I m : AdicCompletion I R →+* R ⧸ I ^ m)

theorem **evalₐ_adicMap** (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (n : ℕ) (x : AdicCompletion I R) : AdicCompletion.evalₐ J n (adicMap I J f hf x) = levMap I J f hf n (AdicCompletion.evalₐ I n x)

theorem **evalₐ_adicMap_mk** (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (n : ℕ) (x : AdicCompletion I R) (r : R) (h : AdicCompletion.evalₐ I n x = Ideal.Quotient.mk _ r) : AdicCompletion.evalₐ J n (adicMap I J f hf x) = Ideal.Quotient.mk _ (f r)

theorem **adicMap_alg** (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (r : R) : adicMap I J f hf (algebraMap R (AdicCompletion I R) r) = algebraMap S (AdicCompletion J S) (f r)

theorem **span_le_comap** (p : ℕ) (f : R →+* S) : Ideal.span {(p : R)} ≤ (Ideal.span {(p : S)}).comap f

theorem **padicMap_alg** (p : ℕ) (f : R →+* S) (r : R) : padicMap p f (algebraMap R (PHat p R) r) = algebraMap S (PHat p S) (f r)

theorem **exists_acris_completion** (l : ℕ) [Fact l.Prime] : ∃ e : Per.Acris l ≃+* AdicCompletion (Ideal.span {(l : Per.Acris0 l)}) (Per.Acris0 l), ∀ s : Per.Acris0 l, e (algebraMap (Per.Acris0 l) (Per.Acris l) s) = algebraMap (Per.Acris0 l) (AdicCompletion (Ideal.span {(l : Per.Acris0 l)}) (Per.Acris0 l)) s

theorem **coeff_galois** (l : ℕ) [Fact l.Prime] (g : PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) (x : Per.Fl l) (n : ℕ) (y : Per.OC l) (h : PreTilt.coeff n x = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : Per.OC l)}) y) : PreTilt.coeff n (Per.σF g x) = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : Per.OC l)}) (Per.σO g y)

Development.AbelianVarietyTheory.CrystallinePeriods.CrisIncl

theorem **exists_incl** (l : ℕ) [Fact l.Prime] : ∃ ι : Per.Bcris l →+* Per.BdR l, Per.IsCrisDR l ι

theorem **exists_restrict** (l : ℕ) [Fact l.Prime] (ι : Per.Bcris l →+* Per.BdR l) (h : Per.IsCrisDR l ι) : ∃ j : Per.Acris l →+* Per.BdRp l, (∀ x : Per.Acris l, ι (algebraMap (Per.Acris l) (Per.Bcris l) x) = algebraMap (Per.BdRp l) (Per.BdR l) (j x)) ∧ (∀ s : Per.Acris0 l, j (algebraMap (Per.Acris0 l) (Per.Acris l) s) = algebraMap (Per.AinfP l) (Per.BdRp l) (s : Per.AinfP l)) ∧ ∀ n : ℕ, ∃ N : ℕ, ∀ x : Per.Acris l, ∃ a : Per.Ainf l, Per.ev l n (((l : ℕ) : Per.BdRp l) ^ N * j x) = Ideal.Quotient.mk _ (algebraMap (Per.Ainf l) (Per.AinfP l) a)

theorem **factorial_mul_ev** (l : ℕ) [Fact l.Prime] (e : Per.Fl l) (y : Per.BdRp l) (he : e.untilt = 1) (h : Per.IsLog l e y) (n : ℕ) : ((n.factorial : ℕ) : Per.AinfP l ⧸ Per.kerP l ^ n) * Per.ev l n y = Ideal.Quotient.mk _ (∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : Per.AinfP l) * (algebraMap (Per.Ainf l) (Per.AinfP l) (Per.tm e - 1)) ^ (k + 1))

theorem **isLog_of_partial** (l : ℕ) [Fact l.Prime] (e : Per.Fl l) (he : e.untilt = 1) (y : Per.BdRp l) (h : ∀ n : ℕ, ∃ c : Per.AinfP l, ((n.factorial : ℕ) : Per.AinfP l) * c = ∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : Per.AinfP l) * (algebraMap (Per.Ainf l) (Per.AinfP l) (Per.tm e - 1)) ^ (k + 1) ∧ Per.ev l n y = Ideal.Quotient.mk _ c) : Per.IsLog l e y

theorem **log_unique** (l : ℕ) [Fact l.Prime] (e : Per.Fl l) (he : e.untilt = 1) (y y' : Per.BdRp l) (h : Per.IsLog l e y) (h' : Per.IsLog l e y') : y = y'

theorem **incl_aB** (l : ℕ) [Fact l.Prime] (ι : Per.Bcris l →+* Per.BdR l) (h : Per.IsCrisDR l ι) (a : Per.Ainf l) : ι (Per.aB l a) = algebraMap (Per.BdRp l) (Per.BdR l) (algebraMap (Per.AinfP l) (Per.BdRp l) (algebraMap (Per.Ainf l) (Per.AinfP l) a))

Development.AbelianVarietyTheory.CrystallinePeriods.CrisLoc

theorem **exists_root_ne_one** : ∃ e : Per.Fl l, e.untilt = 1 ∧ e ≠ 1

theorem **bcris_away** (e : Per.Fl l) (he : e.untilt = 1) (hne : e ≠ 1) : IsLocalization.Away (algebraMap (Per.Ainf l) (Per.Acris l) (Per.tm e - 1)) (Per.Bcris l)

theorem **bdR_away** (b : Per.AinfP l) (hb : Per.kerP l = Ideal.span {b}) : IsLocalization.Away (algebraMap (Per.AinfP l) (Per.BdRp l) b) (Per.BdR l)

theorem **mem_fil_neg** (b : Per.AinfP l) (hb : Per.kerP l = Ideal.span {b}) (n : ℕ) (x : Per.BdR l) : x ∈ Per.FilD l (-(n : ℤ)) ↔ ∃ y : Per.BdRp l, x * algebraMap (Per.BdRp l) (Per.BdR l) (algebraMap (Per.AinfP l) (Per.BdRp l) b) ^ n = algebraMap (Per.BdRp l) (Per.BdR l) y

theorem **unit_of_root** (e : Per.Fl l) (he : e.untilt = 1) (hne : e ≠ 1) : IsUnit (algebraMap (Per.Acris l) (Per.Bcris l) (algebraMap (Per.Ainf l) (Per.Acris l) (Per.tm e - 1)))

theorem **unit_of_root_aB** (e : Per.Fl l) (he : e.untilt = 1) (hne : e ≠ 1) : IsUnit (Per.aB l (Per.tm e - 1))

theorem **bcris_hom_ext** {R : Type*} [CommRing R] {f g : Per.Bcris l →+* R} (h : ∀ x : Per.Acris l, f (algebraMap _ _ x) = g (algebraMap _ _ x)) : f = g

theorem **exists_mul_pow_eq** (e : Per.Fl l) (he : e.untilt = 1) (hne : e ≠ 1) (b : Per.Bcris l) : ∃ (n : ℕ) (x : Per.Acris l), b * algebraMap (Per.Acris l) (Per.Bcris l) (algebraMap (Per.Ainf l) (Per.Acris l) (Per.tm e - 1)) ^ n = algebraMap (Per.Acris l) (Per.Bcris l) x

theorem **exists_pow_mul_eq_zero** (e : Per.Fl l) (he : e.untilt = 1) (hne : e ≠ 1) {x : Per.Acris l} (hx : algebraMap (Per.Acris l) (Per.Bcris l) x = 0) : ∃ n : ℕ, algebraMap (Per.Ainf l) (Per.Acris l) (Per.tm e - 1) ^ n * x = 0

theorem **exists_extension** (e : Per.Fl l) (he : e.untilt = 1) (hne : e ≠ 1) {R : Type*} [CommRing R] (g : Per.Acris l →+* R) (hg : IsUnit (g (algebraMap (Per.Ainf l) (Per.Acris l) (Per.tm e - 1)))) : ∃ F : Per.Bcris l →+* R, ∀ x : Per.Acris l, F (algebraMap _ _ x) = g x

theorem **bdR_hom_ext** {R : Type*} [CommRing R] {f g : Per.BdR l →+* R} (h : ∀ x : Per.BdRp l, f (algebraMap _ _ x) = g (algebraMap _ _ x)) : f = g

theorem **constantCoeff_root_ne_zero** (e : Per.Fl l) (hne : e ≠ 1) : WittVector.constantCoeff (Per.tm e - 1 : Per.Ainf l) ≠ 0

Development.AbelianVarietyTheory.CrystallinePeriods.CrisMaps

theorem **frob_extends** (l : ℕ) [Fact l.Prime] : Extends (Per.φA l) (frob l)

theorem **gal_extends** (g : Per.Gq l) : Extends (Per.σA g) (gal g)

theorem **frob_gal** (g : Per.Gq l) (x : Per.Acris l) : frob l (gal g x) = gal g (frob l x)

theorem **φB_algebraMap** (l : ℕ) [Fact l.Prime] (x : Per.Acris l) : Per.φB l (algebraMap (Per.Acris l) (Per.Bcris l) x) = algebraMap (Per.Acris l) (Per.Bcris l) (frob l x)

theorem **σB_algebraMap** (g : Per.Gq l) (x : Per.Acris l) : Per.σB g (algebraMap (Per.Acris l) (Per.Bcris l) x) = algebraMap (Per.Acris l) (Per.Bcris l) (gal g x)

Development.AbelianVarietyTheory.CrystallinePeriods.CrisMapsLoc

theorem **phiA_eq** (l : ℕ) [Fact l.Prime] : phiA l = frob l

theorem **sigA_eq** (g : Per.Gq l) : sigA g = gal g

theorem **phiA_extends** (l : ℕ) [Fact l.Prime] : Extends (Per.φA l) (phiA l)

theorem **sigA_extends** (g : Per.Gq l) : Extends (Per.σA g) (sigA g)

theorem **phiA_sigA** (g : Per.Gq l) (x : Per.Acris l) : phiA l (sigA g x) = sigA g (phiA l x)

theorem **φB_phiA** (l : ℕ) [Fact l.Prime] (x : Per.Acris l) : Per.φB l (algebraMap (Per.Acris l) (Per.Bcris l) x) = algebraMap (Per.Acris l) (Per.Bcris l) (phiA l x)

theorem **σB_sigA** (g : Per.Gq l) (x : Per.Acris l) : Per.σB g (algebraMap (Per.Acris l) (Per.Bcris l) x) = algebraMap (Per.Acris l) (Per.Bcris l) (sigA g x)

Development.AbelianVarietyTheory.CrystallinePeriods.CrystallineLogarithm.EvalLogAlgebraMap

theorem **exists_factor** {A B C : Type*} [Ring A] [Ring B] [Ring C] (f : A →+* C) (g : B →+* C) (hg : Function.Injective g) (h : ∀ x, ∃ y, f x = g y) : ∃ j : A →+* B, ∀ x, f x = g (j x)

theorem **Bridge.CrisIncl.exists_restrict** : ∀ (l : ℕ) [Fact l.Prime] (ι : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (h : IsCrisDR l ι), ∃ j : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) →+* (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)), (∀ x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)), ι (algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x) = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (j x)) ∧ (∀ s : ɪ(Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)), j (algebraMap (ɪ(Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) s) = algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Ref

theorem **Bridge.CrisIncl.exists_incl** : ∀ (l : ℕ) [Fact l.Prime], ∃ ι : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l)), IsCrisDR l ι

Development.AbelianVarietyTheory.CrystallinePeriods.CrystallineLogarithm.ExistsExtensionU

theorem **Bridge.CrisLoc.bcris_hom_ext_u** : ∀ (l : ℕ) [Fact l.Prime] {R : Type u} [CommRing R] {f g : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* R} (h : ∀ x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)), f (algebraMap _ _ x) = g (algebraMap _ _ x)), f = g

theorem **geom_eq** {A : Type*} [CommRing A] (p : ℕ) (v : A) : ∃ c : A, ∑ i ∈ Finset.range p, v ^ i = p + (v - 1) * c

theorem **step** {A : Type*} [CommRing A] (p n : ℕ) (v w : A) (h : v ^ p ^ (n + 1) - 1 = (p : A) ^ (n + 1) * ((v - 1) * w)) : ∃ z : A, v ^ p ^ (n + 1 + 1) - 1 = (p : A) ^ (n + 1 + 1) * ((v - 1) * (w + (p : A) ^ n * ((v - 1) * z)))

theorem **key** {A : Type*} [CommRing A] (p : ℕ) [IsAdicComplete (span {(p : A)}) A] (v r t : A) (h0 : v ^ p - 1 = p * ((v - 1) * r)) (hr : r - 1 ∈ jacobson (⊥ : Ideal A)) (hv : v - 1 ∈ jacobson (⊥ : Ideal A)) (ht : ∀ (n : ℕ) (w : A), v ^ p ^ n - 1 = (p : A) ^ n * w → (p : A) ^ n ∣ t - w) : ∃ L : A, IsUnit L ∧ t = (v - 1) * L

theorem **p_dvd_pow** {y : AcrisZero p R} (hy : y ∈ AcrisZero.pdIdeal p R) : (p : Acris p R) ∣ algebraMap (AcrisZero p R) (Acris p R) y ^ p

theorem **mem_jac** {y : AcrisZero p R} (hy : y ∈ AcrisZero.pdIdeal p R) : algebraMap (AcrisZero p R) (Acris p R) y ∈ jacobson (⊥ : Ideal (Acris p R))

theorem **p_dvd_geom** [IsDomain R] (e : Tilt p R) (he : PreTilt.untilt e = 1) : (p : AcrisZero p R) ∣ algebraMap (Ainf p R) (AcrisZero p R) (∑ i ∈ Finset.range p, Ainf.teich p e ^ i)

theorem **log_eq** [IsDomain R] [CharZero R] (η : tateModule p R) : ∃ L : Acris p R, IsUnit L ∧ Acris.logTeich η = algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 - 1) * L

theorem **main** [IsDomain R] [CharZero R] (η : tateModule p R) {B : Type*} [CommRing B] (g : Acris p R →+* B) (hg : IsUnit (g (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 − 1)))) : ∃ F : Bcris p R →+* B, ∀ x : Acris p R, F (algebraMap _ _ x) = g x

theorem **Bridge.CrisLoc.exists_extension_u** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1) {R : Type u} [CommRing R] (g : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) →+* R) (hg : IsUnit (g (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e − 1)))), ∃ F : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* R, ∀ x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)), F (algebraMap _ _ x) = g x

theorem **Bridge.CrisLoc.bdR_hom_ext_u** : ∀ (l : ℕ) [Fact l.Prime] {R : Type u} [CommRing R] {f g : (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l)) →+* R} (h : ∀ x : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)), f (algebraMap _ _ x) = g (algebraMap _ _ x)), f = g

Development.AbelianVarietyTheory.CrystallinePeriods.CrystallineLogarithm.ExistsPowDivFactorial

theorem **expand** {A : Type*} [CommRing A] {p : ℕ} (hp : p.Prime) (m : A) : ∃ h : A, (m + 1) ^ p = m ^ p + 1 + p * m * (1 + m * h)

theorem **val_fact** (p : ℕ) [hp : Fact p.Prime] (j : ℕ) : p ^ j ≤ padicValNat p (p ^ j * (p − 1))! + 1

theorem **isUnit_of_mem** {y : Acris p R} (hy : y ∈ (span {(p : Acris p R)}).radical) : IsUnit (1 + y)

theorem **p_mem** : (p : Acris p R) ∈ (span {(p : Acris p R)}).radical

theorem **fact_dvd** (a : Ainf p R) (ha : Ainf.theta p R a = 0) (n : ℕ) : (n ! : Acris p R) ∣ algebraMap (Ainf p R) (Acris p R) a ^ n

theorem **mem_rad** (a : Ainf p R) (ha : Ainf.theta p R a = 0) : algebraMap (Ainf p R) (Acris p R) a ∈ (span {(p : Acris p R)}).radical

theorem **pow_dvd_of_prod** (a : Ainf p R) (ha : Ainf.theta p R a = 0) {b : Acris p R} (hb : b ∈ (span {(p : Acris p R)}).radical) : ∃ k : ℕ, (p : Acris p R) ^ (k + 1) ∣ ((algebraMap (Ainf p R) (Acris p R) a * b) ^ (p − 1)) ^ k

theorem **exists_pow_dvd** [IsDomain R] (e : Tilt p R) (he : PreTilt.untilt e = 1) : ∃ k : ℕ, (p : Acris p R) ^ (k + 1) ∣ ((algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e) − 1) ^ (p − 1)) ^ k

theorem **step** [IsDomain R] (e : Tilt p R) (he : PreTilt.untilt e = 1) {z : Acris p R} (hz : (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e) ^ p − 1) * z = 0) : (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e) − 1) * z = 0

theorem **tower** {A : Type*} [CommRing A] (p : ℕ) (u y : A) (hy : u − 1 = p ^ 2 * y) (n : ℕ) : ∃ c : A, u ^ p ^ n − 1 = p ^ n * (u − 1) * (1 + p * c)

theorem **of_log** [IsDomain R] (e : tateModule p R) {z : Acris p R} (hz : Acris.logTeich e * z = 0) : (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e.1) − 1) * z = 0

theorem **log_change** [IsDomain R] (e : tateModule p R) {ε : tateModule p R} (hε : IsPrimitive ε) {z : Acris p R} (hz : Acris.logTeich ε * z = 0) : Acris.logTeich e * z = 0

theorem **main** [IsDomain R] (e : tateModule p R) {m : BcrisPlus p R} (hm : m ∈ BcrisPlus.periods p R) : ∀ x : Acris p R, m * algebraMap _ _ x = 0 → ∃ n : ℕ, (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e.1) − 1) ^ n * x = 0

theorem **Bridge.CrisLoc.exists_pow_mul_eq_zero** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1) {x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))} (hx : algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x = 0), ∃ n : ℕ, algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e − 1) ^ n * x = 0

theorem **binom_poly** {p : ℕ} (hp : p.Prime) : ∃ h : Polynomial ℤ, (1 + Polynomial.X) ^ p = 1 + Polynomial.X ^ p + (p : Polynomial ℤ) * Polynomial.X * (1 + Polynomial.X * h)

theorem **binom** {A : Type*} [CommRing A] {p : ℕ} (hp : p.Prime) (x : A) : ∃ h : A, (1 + x) ^ p = 1 + x ^ p + (p : A) * x * (1 + x * h)

theorem **mem_jac** {z : Acris p R} {n : ℕ} (h : (p : Acris p R) ∣ z ^ n) : z ∈ (⊥ : Ideal (Acris p R)).jacobson

theorem **fact_gamma** (n : ℕ) (x : Ainf p R) (hx : Ainf.theta p R x = 0) : (n ! : Acris p R) * algebraMap _ _ (AcrisZero.gamma n x hx) = algebraMap (Ainf p R) (Acris p R) x ^ n

theorem **p_dvd_pow** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : (p : Acris p R) ∣ algebraMap (Ainf p R) (Acris p R) x ^ p

theorem **p_dvd_gamma_pow** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : (p : Acris p R) ∣ algebraMap _ (Acris p R) (AcrisZero.gamma p x hx) ^ p

theorem **theta_mu** (η : tateModule p R) : Ainf.theta p R (Ainf.teich p η.1 − 1) = 0

theorem **exists_congr** (η : tateModule p R) : ∃ x c : Ainf p R, Ainf.theta p R x = 0 ∧ (p : Ainf p R) ∣ (Ainf.teich p η.1 − 1) ^ (p − 1) − x ^ p * c

theorem **exists_pow_eq_p_mul** (η : tateModule p R) : ∃ y ∈ (⊥ : Ideal (Acris p R)).jacobson, algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 − 1) ^ (p − 1) = p * y

theorem **assoc_pow** (η : tateModule p R) : Associated ((p : Acris p R) * (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1) − 1)) (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1) ^ p − 1)

theorem **exists_assoc_log** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : ∃ l : Acris p R, (p : Acris p R) ^ 2 * l = algebraMap (Ainf p R) _ u ^ p ^ 2 − 1 ∧ Associated l (Acris.log u hu)

theorem **assoc_log** (η : tateModule p R) : Associated (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 − 1)) (Acris.logTeich η)

theorem **isUnit_log_iff** {B : Type*} [CommRing B] (g : Acris p R →+* B) (hp : IsUnit (p : B)) {ε η : tateModule p R} (hε : IsPrimitive ε) (hη : η ≠ 1) : IsUnit (g (Acris.logTeich η)) ↔ IsUnit (g (Acris.logTeich ε))

theorem **away** [ValuationRing R] [IsAlgClosed (FractionRing R)] (η : tateModule p R) (hη : η ≠ 1) : IsLocalization.Away (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 − 1)) (Bcris p R)

theorem **Bridge.CrisLoc.bcris_away** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1), IsLocalization.Away (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e − 1)) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)))

theorem **mem_jac_of_pow** {x : A} {n : ℕ} (h : x ^ n ∈ jacobson (⊥ : Ideal A)) : x ∈ jacobson (⊥ : Ideal A)

theorem **isUnit_of_sub_one_mem** {x : A} (h : x − 1 ∈ jacobson (⊥ : Ideal A)) : IsUnit x

theorem **exists_limit** (a : A) [IsAdicComplete (span {a}) A] (d : ℕ → A) (h : ∀ n, a ^ n ∣ d (n + 1) − d n) : ∃ L, ∀ n, a ^ n ∣ d n − L

theorem **geom_sum_eq** (n : ℕ) (v : A) : ∃ g, ∑ i ∈ Finset.range n, v ^ i = n + (v − 1) * g

theorem **geom_sum_frob** (p : ℕ) [Fact p.Prime] (v : A) : ∃ g, ∑ i ∈ Finset.range p, (v ^ p) ^ i = (∑ i ∈ Finset.range p, v ^ i) ^ p + p * (1 - p ^ (p - 1) + (v - 1) * g)

theorem **p_mem_jac** : (p : Acris p R) ∈ jacobson (⊥ : Ideal (Acris p R))

theorem **factorial_dvd** (n : ℕ) (x : Ainf p R) (hx : Ainf.theta p R x = 0) : (n ! : Acris p R) ∣ algebraMap (Ainf p R) (Acris p R) x ^ n

theorem **p_dvd_pow** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : (p : Acris p R) ∣ algebraMap (Ainf p R) (Acris p R) x ^ p

theorem **mem_jac_of_theta** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : algebraMap (Ainf p R) (Acris p R) x ∈ jacobson (⊥ : Ideal (Acris p R))

theorem **exists_pow_eq_mul** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : ∃ c ∈ jacobson (⊥ : Ideal (Acris p R)), algebraMap (Ainf p R) (Acris p R) x ^ p = p * c

theorem **theta_teich_sub_one** (η : tateModule p R) : Ainf.theta p R (Ainf.teich p η.1 - 1) = 0

theorem **exists_unit** [IsDomain R] {ε : tateModule p R} (hε : IsPrimitive ε) : ∃ w : Acris p R, IsUnit w ∧ ∑ i ∈ Finset.range p, algebraMap (Ainf p R) (Acris p R) (Ainf.teich p ε.1) ^ i = p * w

theorem **exists_log_eq** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : ∃ l d : Acris p R, IsUnit d ∧ (p : Acris p R) ^ 2 * l = algebraMap (Ainf p R) (Acris p R) u ^ p ^ 2 - 1 ∧ Acris.log u hu = l * d

theorem **logTeich_dvd** [IsDomain R] {ε : tateModule p R} (hε : IsPrimitive ε) : Acris.logTeich ε ∣ algebraMap (Ainf p R) (Acris p R) (Ainf.teich p ε.1) - 1

theorem **log_linear** (η : tateModule p R) (a : ℤ_[p]) : Acris.logTeich (tateModule.padicPow η a) = algebraMap ℤ_[p] (Acris p R) a * Acris.logTeich η

theorem **logTeich_dvd_of_ne_one** [IsDomain R] {ε : tateModule p R} (hε : IsPrimitive ε) (η : tateModule p R) (hη : η ≠ 1) : Acris.logTeich ε ∣ algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 - 1)

theorem **main** [IsDomain R] (η : tateModule p R) (hη : η ≠ 1) (b : Bcris p R) : ∃ (n : ℕ) (x : Acris p R), b * algebraMap (Acris p R) (Bcris p R) (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p η.1 - 1)) ^ n = algebraMap (Acris p R) (Bcris p R) x

theorem **Bridge.CrisLoc.exists_mul_pow_eq** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1) (b : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))), ∃ (n : ℕ) (x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))), b * algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e - 1)) ^ n = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x

theorem **bound** {p v n k : ℕ} (hp : 2 ≤ p) (hv : p ^ v ≤ k + 1) (hk : p * (2 * n + 2) ≤ k) : p * (v + n) ≤ k

theorem **term_dvd** {p : ℕ} [Fact p.Prime] {A : Type*} [CommRing A] (f : ℤ_[p] →+* A) (hc : ∀ (m : ℕ) (y z : A), (p : A) ^ m * y = (p : A) ^ m * z → y = z) {x : A} (hx : (p : A) ∣ x ^ p) {k n : ℕ} (hk : p * (2 * n + 2) ≤ k) {a : A} (ha : ((k + 1 : ℕ) : A) * a = x ^ (k + 1)) : x * (p : A) ^ n ∣ a

theorem **succ_mul_term** (u : Ainf p R) (hu : Ainf.theta p R u = 1) (k : ℕ) : ((k + 1 : ℕ) : Acris p R) * term u hu k = algebraMap (Ainf p R) (Acris p R) (u - 1) ^ (k + 1)

theorem **partial_eq** (u : Ainf p R) (hu : Ainf.theta p R u = 1) (N : ℕ) : algebraMap (AcrisZero p R) (Acris p R) (Acris.logPartial N u hu) = ∑ k ∈ Finset.range N, (-1) ^ k * term u hu k

theorem **p_dvd_pow** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : (p : Acris p R) ∣ algebraMap (Ainf p R) (Acris p R) (u - 1) ^ p

theorem **dvd_tail** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : ∃ N, algebraMap (Ainf p R) (Acris p R) (u - 1) ∣ Acris.log u hu - ∑ k ∈ Finset.range N, (-1) ^ k * term u hu k

theorem **dvd_log** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : algebraMap (Acris p R) (Bcris p R) (algebraMap (Ainf p R) (Acris p R) (u - 1)) ∣ algebraMap (Acris p R) (Bcris p R) (Acris.log u hu)

theorem **isUnit_logTeich** [IsDomain R] [ValuationRing R] [IsAlgClosed (FractionRing R)] [CharZero R] (η : tateModule p R) (h : η ≠ 1) : IsUnit (Bcris.logTeich η)

theorem **Bridge.CrisLoc.unit_of_root** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1), IsUnit (algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e - 1)))

theorem **Bridge.RingBasics.exists_pow_div_factorial** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ), ∃ c : ℤ_[l], ((n.factorial : ℕ) : ℤ_[l]) * c = ((l : ℕ) : ℤ_[l]) ^ n

theorem **exists_one_add_pow** {p : ℕ} (hp : p.Prime) {A : Type*} [CommRing A] (y : A) : ∃ s : A, (1 + y) ^ p - 1 = y * (y ^ (p - 1) + p * (1 + y * s))

theorem **mem_jac_of_pow** {A : Type*} [CommRing A] {x : A} {n : ℕ} (h : x ^ n ∈ jacobson (⊥ : Ideal A)) : x ∈ jacobson (⊥ : Ideal A)

theorem **isUnit_one_add** {A : Type*} [CommRing A] {x : A} (h : x ∈ jacobson (⊥ : Ideal A)) : IsUnit (1 + x)

theorem **p_mem_jac** : (p : Acris p R) ∈ jacobson (⊥ : Ideal (Acris p R))

theorem **gamma_mul** (a b : ℕ) (x : Ainf p R) (hx : Ainf.theta p R x = 0) : AcrisZero.gamma a x hx * AcrisZero.gamma b x hx = ((a + b).choose a : ℕ) * AcrisZero.gamma (a + b) x hx

theorem **gamma_pow** (k : ℕ) (x : Ainf p R) (hx : Ainf.theta p R x = 0) : ∃ c : AcrisZero p R, AcrisZero.gamma p x hx ^ (k + 1) = c * AcrisZero.gamma (p * (k + 1)) x hx

theorem **p_dvd_gamma_pow** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : (p : AcrisZero p R) ∣ AcrisZero.gamma p x hx ^ p

theorem **exists_pow_eq_p_mul** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : ∃ z ∈ jacobson (⊥ : Ideal (Acris p R)), algebraMap (Ainf p R) (Acris p R) x ^ p = p * z

theorem **mem_jac_of_theta** (x : Ainf p R) (hx : Ainf.theta p R x = 0) : algebraMap (Ainf p R) (Acris p R) x ∈ jacobson (⊥ : Ideal (Acris p R))

theorem **log_eq** (u : Ainf p R) (hu : Ainf.theta p R u = 1) (a : Acris p R) (ha : a ∈ jacobson (⊥ : Ideal (Acris p R))) (h : algebraMap (Ainf p R) (Acris p R) u - 1 = p * a) : ∃ L : Acris p R, IsUnit L ∧ Acris.log u hu = p * a * L

theorem **teich_pow_sub_one** [IsDomain R] (e : Tilt p R) (he : e.untilt = 1) (h1 : Tilt.seq 1 e ≠ 1) : ∃ w : Acris p R, IsUnit w ∧ algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e) ^ p - 1 = p * (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e - 1) * w)

theorem **exists_primitive** (n : ℕ) : ∀ e : Tilt p R, e.untilt = 1 → Tilt.seq n e ≠ 1 → ∃ (e' : Tilt p R) (k : ℕ), e = e' ^ k ∧ e'.untilt = 1 ∧ Tilt.seq 1 e' ≠ 1

theorem **logTeich_padicPow'** (η : tateModule p R) (a : ℤ_[p]) : Acris.logTeich η | Acris.logTeich (tateModule.padicPow η a)

theorem **main** [IsDomain R] (e : Tilt p R) (he : e.untilt = 1) (hne : e ≠ 1) {B : Type} [CommRing B] (g : Acris p R →+* B) (hg : IsUnit (g (algebraMap (Ainf p R) (Acris p R) (Ainf.teich p e - 1)))) : ∃ F : Bcris p R →+* B, ∀ x : Acris p R, F (algebraMap _ _ x) = g x

theorem **Bridge.CrisLoc.exists_extension** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1) {R : Type} [CommRing R] (g : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) →+* R) (hg : IsUnit (g (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e - 1)))), ∃ F : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* R, ∀ x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)), F (algebraMap _ _ x) = g x

Development.AbelianVarietyTheory.CrystallinePeriods.CrystallineLogarithm.FactorialMulTruncLog

theorem **Bridge.CrisIncl.log_unique** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : PreTilt.untilt e = 1) (y y' : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (h : IsLog l e y) (h' : IsLog l e y'), y = y'

theorem **Bridge.CrisIncl.incl_aB** : ∀ (l : ℕ) [Fact l.Prime] (ι : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (h : IsCrisDR l ι) (a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), ι (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) a) = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) a))

theorem **factorial_mul_truncLog** {B : Type*} [CommRing B] [Algebra ℚ B] (n : ℕ) (x : B) : ((n.factorial : ℕ) : B) * truncLog n x = ∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : B) * x ^ (k + 1)

theorem **general** {p : ℕ} [Fact p.Prime] {R : Type*} [CommRing R] [Fact ¬IsUnit (p : R)] [IsAdicComplete (Ideal.span {(p : R)}) R] (u : Ainf p R) (hu : BdRPlus.theta p R (algebraMap (Ainf p R) (BdRPlus p R) u) = 1) (n : ℕ) : ((n.factorial : ℕ) : AinfInv p R / AinfInv.kerTheta p R ^ n) * BdRPlus.eval p R n (BdRPlus.log (algebraMap (Ainf p R) (BdRPlus p R) u) hu) = Ideal.Quotient.mk _ (∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : AinfInv p R) * (algebraMap (Ainf p R) (AinfInv p R) (u - 1)) ^ (k + 1))

theorem **Bridge.CrisIncl.factorial_mul_ev** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (he : PreTilt.untilt e = 1) (h : IsLog l e y) (n : ℕ), ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) / (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l)) ^ n) * Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y = Ideal.Quotient.mk _ (∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) * (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e - 1)) ^ (k + 1))

theorem **eq_truncLog** {B : Type*} [CommRing B] [Algebra ℚ B] (n : ℕ) (c x : B) (h : ((n.factorial : ℕ) : B) * c = ∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : B) * x ^ (k + 1)) : c = truncLog n x

theorem **theta_eq_one** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : BdRPlus.theta p R (algebraMap (Ainf p R) (BdRPlus p R) u) = 1

theorem **eq_log** (u : Ainf p R) (hu : BdRPlus.theta p R (algebraMap (Ainf p R) (BdRPlus p R) u) = 1) (y : BdRPlus p R) (h : ∀ n : ℕ, ∃ c : AinfInv p R, ((n.factorial : ℕ) : AinfInv p R) * c = ∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : AinfInv p R) * (algebraMap (Ainf p R) (AinfInv p R) (u - 1)) ^ (k + 1) ∧ BdRPlus.eval p R n y = Ideal.Quotient.mk _ c) : y = BdRPlus.log (algebraMap (Ainf p R) (BdRPlus p R) u) hu

theorem **Bridge.CrisIncl.isLog_of_partial** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : PreTilt.untilt e = 1) (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (h : ∀ n : ℕ, ∃ c : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)), ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) * c = ∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) * (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (WittVector.teichmuller l e - 1)) ^ (k + 1) ∧ Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y = Ideal.Quotient.mk _ c), IsLog l e y

Development.AbelianVarietyTheory.CrystallinePeriods.CrystallineLogarithm.MemAcris0OfFactorialMul

theorem **Bridge.cris_zero_to_period_apply** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap ɪ(Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) a = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) a)

theorem **Bridge.aB_apply** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) a = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) a)

theorem **Bridge.algebraMap_comp_ofPadicInt_apply** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), ((algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z)

theorem **Bridge.sigA_teich** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (y : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) (WittVector.teichmuller l y) = WittVector.teichmuller l ((Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) y)

theorem **Bridge.ev_ext** : ∀ (l : ℕ) [Fact l.Prime] (y y' : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (h : ∀ n, Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y = Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y'), y = y'

theorem **Bridge.thetaD_of_mem_FilP** : ∀ (l : ℕ) [Fact l.Prime] {y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))} (h : y ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) 1), (Reformulated.AutPerF.BdRPlus.thetaC l (PadicComplexInt l)) y = 0

theorem **Bridge.sigA_eq_map** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)), (Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) = WittVector.map

((Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)))

theorem **isUnit_general** {p : ℕ} [Fact p.Prime] {R : Type*} [CommRing R] [Fact ¬IsUnit (p : R)] [IsAdicComplete (Ideal.span {(p : R)}) R] [IsDomain R] [ValuationRing R] [CharZero R] [IsAlgClosed (FractionRing R)] {a : AinfInv p R} (h : AinfInv.kerTheta p R = Ideal.span {a}) : IsUnit (((algebraMap (BdRPlus p R) (BdR p R)).comp (algebraMap (AinfInv p R) (BdRPlus p R))) a)

theorem **Bridge.isUnit_of_generator** : ∀ (l : ℕ) [Fact l.Prime] {a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))} (h : (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l)) = Ideal.span {a}), IsUnit (((algebraMap (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)))) a)

theorem **Bridge.sigC_cont** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)), Continuous ((MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] g))

theorem **Bridge.mem_FilP** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))), y ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n ↔ Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y = 0

theorem **Bridge.alg_mem_kerP** : ∀ (l : ℕ) [Fact l.Prime] {a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))} (h : (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) a = 0), algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) a ∈ (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l))

theorem **Bridge.exists_acris_equiv** : ∀ (l : ℕ) [Fact l.Prime], ∃ e : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) ≃+* AdicCompletion (Ideal.span {((l : ℕ) : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)))}) ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))), ∀ s, e (algebraMap ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) s) = algebraMap _ _ s

theorem **Bridge.mem_filtration_iff_mem_kerTheta_pow** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))), y ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n ↔ Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y = 0

theorem **Bridge.ev_surj** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ), Function.Surjective (Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n)

theorem **Bridge.alg_mem_FilD_nat** : ∀ (l : ℕ) [Fact l.Prime] {n : ℕ} {z : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))} (h : z ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n), algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) z ∈ Reformulated.AutPerF.BdR.fil l ɪ(PadicComplexInt l) (n : ℤ)

theorem **Bridge.bcris_ext** : ∀ (l : ℕ) [Fact l.Prime] {R : Type} [CommRing R] {f g : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* R} (h : ∀ x, f (algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x) = g (algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x)), f = g

theorem **Bridge.mem_FilD_neg** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (x : (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))), x ∈ Reformulated.AutPerF.BdR.fil l ɪ(PadicComplexInt l) (-(n : ℤ)) ↔ ∀ y ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n, ∃ z : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)), algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) z = x * algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) y

theorem **Bridge.phiA_eq_map** : ∀ (l : ℕ) [Fact l.Prime], (Reformulated.AutPerF.Ainf.frob l ɪ(PadicComplexInt l)) = WittVector.map (frobenius ((Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) l)

theorem **Bridge.pD_apply** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), ((algebraMap (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)))) a = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) a)

theorem **Bridge.zB_apply** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z = ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) ((Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z)

theorem **Bridge.tower** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) x = algebraMap ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) x)

theorem **Bridge.zD_apply** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (((algebraMap (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)))).comp ((algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)))) z = ((algebraMap (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)))) (((algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))).comp (Reformulated.AutPerF.A

theorem **Bridge.toC_alg** : ∀ (l : ℕ) [Fact l.Prime] (a : ɪ(PadicComplexInt l)), ((Reformulated.AutPerF.IsUnitBall.awayEquiv (PadicComplexInt l) l).toRingEquiv.toRingHom) (algebraMap (ɪ(PadicComplexInt l)) (Localization.Away ((l : ℕ) : ɪ(PadicComplexInt l))) a) = (a : ℂ_[l])

theorem **Bridge.mem_Acris0_of_dividedPower** : ∀ (l : ℕ) [Fact l.Prime] {x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))} (hx : (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0) {n : ℕ} {y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))} (h : (n.factorial : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) * y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n), y ∈ (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))

theorem **Bridge.sigA_coeff** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (w : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) (n : ℕ), ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) w).coeff n = (Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) (w.coeff n)

theorem **Bridge.zA_apply** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z = WittVector.map (ZMod.castHom (dvd_refl l) ((Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l)))) (WittVector.fromPadicInt l z)

theorem **Bridge.zB_eq_aB_zA** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z = ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) ((Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z)

theorem **Bridge.ev_alg** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) a) = Ideal.Quotient.mk _ a

theorem **Bridge.sigDp_alg** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.BdRPlus.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) a) = algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (Reformulated.AutPerF.AinfInv.extend l ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g))) a)

theorem **Bridge.exists_bdRp_equiv** : ∀ (l : ℕ) [Fact l.Prime], ∃ e : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) ≃+* AdicCompletion ((Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), (∀ a, e (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) a) = algebraMap _ _ a) ∧ ∀ n y, AdicCompletion.evalₐ ((Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l))) n (e y) = Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y

theorem **Bridge.fontaineTheta_algebraMap** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), (((Reformulated.AutPerF.IsUnitBall.awayEquiv (PadicComplexInt l) l).toRingEquiv.toRingHom).comp (Reformulated.AutPerF.AinfInv.theta l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) a) = (((WittVector.fontaineTheta ɪ(PadicComplexInt l) l) a : ɪ(PadicComplexInt l)) : ℂ_[l])

theorem **Bridge.phiA_teich** : ∀ (l : ℕ) [Fact l.Prime] (y : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Ainf.frob l ɪ(PadicComplexInt l)) (WittVector.teichmuller l y) = WittVector.teichmuller l (y ^ l)

theorem **Bridge.phiA_coeff** : ∀ (l : ℕ) [Fact l.Prime] (w : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) (n : ℕ), ((Reformulated.AutPerF.Ainf.frob l ɪ(PadicComplexInt l)) w).coeff n = w.coeff n ^ l

theorem **Bridge.sigO_coe** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (y : ɪ(PadicComplexInt l)), (((MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g) y : ɪ(PadicComplexInt l)) : ℂ_[l]) = (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] g) (y : ℂ_[l])

theorem **Bridge.exists_of_mem_FilD_nat** : ∀ (l : ℕ) [Fact l.Prime] {n : ℕ} {x : (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))} (h : x ∈ Reformulated.AutPerF.BdR.fil l ɪ(PadicComplexInt l) (n : ℤ)), ∃ z ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n, algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) z = x

theorem **Bridge.Acris0_le** : ∀ (l : ℕ) [Fact l.Prime] {S : Subalgebra ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)))} (h : ∀ (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) (n : ℕ) (y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0 → (n.factorial : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) * y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n → y ∈ S), (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) ≤ S

theorem **Bridge.thetaD_alg** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.BdRPlus.thetaC l (PadicComplexInt l)) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) a) = (((Reformulated.AutPerF.IsUnitBall.awayEquiv (PadicComplexInt l) l).toRingEquiv.toRingHom).comp (Reformulated.AutPerF.AinfInv.theta l ɪ(PadicComplexInt l))) a

theorem **Bridge.sigD_alg** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (b : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.BdR.map l (MulSemiringAction.toRingEquiv (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)).toRingHom (algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) b) = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) b)

theorem **Bridge.padicIntMap_eq_algebraMap_comp** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), ((algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z)

theorem **Bridge.aB_a0B_alg** : ∀ (l : ℕ) [Fact l.Prime], (∀ a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) a = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) a)) ∧ ∀ s : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap ɪ(Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplex

theorem **Bridge.sigDp_mem_FilP** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (n : ℕ) (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))), y ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n → (Reformulated.AutPerF.BdRPlus.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) y ∈ Reformulated.AutPerF.BdRPlus.fil l ɪ(PadicComplexInt l) n

theorem **Bridge.pD_a0B_alg** : ∀ (l : ℕ) [Fact l.Prime], (∀ a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)), ((algebraMap (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)))) a = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) a)) ∧ (∀ s : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))

(Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap ɪ(Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l

theorem **Bridge.alg_mem_FilD_zero** : ∀ (l : ℕ) [Fact l.Prime] (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))), algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) y ∈ Reformulated.AutPerF.BdR.fil l ɪ(PadicComplexInt l) 0

theorem **Bridge.zB_alg** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z))

theorem **Bridge.Acris0_le_of_dividedPowers** : ∀ (l : ℕ) [Fact l.Prime] (S : Subalgebra ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)))), (∀ (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) (n : ℕ) (y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0 → ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ∗ y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n → y ∈ S) → (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) ≤ S

theorem **Bridge.zD_eq_alg** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (((algebraMap (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)))).comp ((algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)))) z = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Ainf.

theorem **Bridge.mem_Acris0_of_factorial_mul** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0 → ∀ (n : ℕ) (y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ∗ y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n → y ∈ (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))

theorem **Bridge.acris0_le** : ∀ (l : ℕ) [Fact l.Prime] (S : Subalgebra ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)))), (∀ x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0 → ∀ (n : ℕ) (y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ∗ y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n → y ∈ S) → (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) ≤ S

theorem **Bridge.cris_zero_to_period_algebra** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap ɪ(Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) a) = ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) a

theorem **Bridge.phiB_zB** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (Reformulated.AutPerF.Bcris.frob l ɪ(PadicComplexInt l)) ((((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z) = (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z

theorem **Bridge.acris0_le_3** : ∀ (l : ℕ) [Fact l.Prime] (S : Subalgebra ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l)))), (∀ x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0 → ∀ (n : ℕ) (y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), y ∈ (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) → ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ∗ y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n → y ∈ S) → (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)) ≤ S

theorem **Bridge.dividedPower_mem_Acris0** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) (n : ℕ) (y : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x = 0 → ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ∗ y = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x ^ n → y ∈ (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))

theorem **Bridge.aB_zA** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), ((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) ((Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l)) z) = (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))).comp (Reformulated.AutPerF.Ainf.ofPadicInt l ɪ(PadicComplexInt l))) z

theorem **Bridge.dividedPower_mem_acris0** : ∀ (l : ℕ) [Fact l.Prime] (a : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), a ∈ RingHom.ker ((WittVector.fontaineTheta ɪ(PadicComplexInt l) l)) → ∀ (n : ℕ) (x : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), ((n.factorial : ℕ) : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ∗ x = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) a ^ n → x ∈ (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))

theorem **Bridge.ev_sigDp** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (n : ℕ) (y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) (a : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))), Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n y = Ideal.Quotient.mk _ a → Reformulated.AutPerF.BdRPlus.eval l ɪ(PadicComplexInt l) n ((Reformulated.AutPerF.BdRPlus.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) y) = Ideal.Quotient.mk _ (Reformulated.AutPerF.AinfInv.extend l ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g))) a)

theorem **Bridge.CrisMaps.φB_algebraMap** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Bcris.frob l ɪ(PadicComplexInt l)) (algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x) = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris.frob l ɪ(PadicComplexInt l)) x)

theorem **Bridge.CrisLoc.bdR_frac** : ∀ (l : ℕ) [Fact l.Prime], IsFractionRing ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l)))

theorem **Bridge.CrisLoc.exists_root_ne_one** : ∀ (l : ℕ) [Fact l.Prime], ∃ e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l)), e.untilt = 1 ∧ e ≠ 1

theorem **mem_fil_neg_iff** (b : AinfInv p R) (hb : AinfInv.kerTheta p R = Ideal.span {b}) (n : ℕ) (x : BdR p R) : x ∈ BdR.fil p R (-(n : ℤ)) ↔ ∃ y : BdRPlus p R, x * algebraMap (BdRPlus p R) (BdR p R) (algebraMap (AinfInv p R) (BdRPlus p R) b) ^ n = algebraMap (BdRPlus p R) (BdR p R) y

theorem **Bridge.CrisLoc.mem_fil_neg** : ∀ (l : ℕ) [Fact l.Prime] (b : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (hb : (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l)) = Ideal.span {b}) (n : ℕ) (x : (Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))), x ∈ Reformulated.AutPerF.BdR.fil l ɪ(PadicComplexInt l) (-(n : ℤ)) ↔ ∃ y : (Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l)), x * algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) b) ^ n = algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) y

theorem **Bridge.AdicMaps.coeff_galois** : ∀ (l : ℕ) [Fact l.Prime] (g : PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) (x : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (n : ℕ) (y : ɪ(PadicComplexInt l)) (h : PreTilt.coeff n x = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : ɪ(PadicComplexInt l))}) y), PreTilt.coeff n ((Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) x) = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : ɪ(PadicComplexInt l))}) ((MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g) y)

theorem **Bridge.CrisMaps.σB_algebraMap** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Bcris.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g) (MulAction.injective g)) (algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) x) = algebraMap ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) x)

theorem **Bridge.AdicMaps.exists_acris_completion** : ∀ (l : ℕ) [Fact l.Prime], ∃ e : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) ≃+* AdicCompletion (Ideal.span {(l : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)))}) ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))), ∀ s : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)), e (algebraMap ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))) s) = algebraMap ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l))) (AdicCompletion (Ideal.span {(l : (Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)))}) ((Reformulated.AutPerF.acrisZero l ɪ(PadicComplexInt l)))) s

theorem **Bridge.CrisLoc.bdR_away** : ∀ (l : ℕ) [Fact l.Prime] (b : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (hb : (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l)) = Ideal.span {b}), IsLocalization.Away (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) b) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l)))

theorem **Bridge.CrisMaps.frob_gal** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Acris.frob l ɪ(PadicComplexInt l)) ((Reformulated.AutPerF.Acris.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) x) = (Reformulated.AutPerF.Acris.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) ((Reformulated.AutPerF.Acris.frob l ɪ(PadicComplexInt l)) x)

theorem **Bridge.RingBasics.σA_mul** : ∀ (l : ℕ) [Fact l.Prime] (g h : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) (g * h))) x = (Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) h)) x)

theorem **Bridge.RingBasics.φA_apply** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Ainf.frob l ɪ(PadicComplexInt l)) x = WittVector.frobenius x

theorem **Bridge.RingBasics.fontaineTheta_eq_zero_of_mem_ker** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (hx : x ∈ (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l)) ^ 1), (((Reformulated.AutPerF.IsUnitBall.awayEquiv (PadicComplexInt l) l).toRingEquiv.toRingHom).comp (Reformulated.AutPerF.AinfInv.theta l ɪ(PadicComplexInt l))) x = 0

theorem **Bridge.RingBasics.σF_mul** : ∀ (l : ℕ) [Fact l.Prime] (g h : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) (g * h))) x = (Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) ((Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) h)) x)

theorem **Bridge.RingBasics.norm_σC** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : ℂ_[l]), ‖(MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] g) x‖ = ‖x‖

theorem **Bridge.RingBasics.σC_coe** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (PadicAlgCl l)), (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] g) (x : ℂ_[l]) = ((g x : (PadicAlgCl l)) : ℂ_[l])

theorem **Bridge.RingBasics.σA_one** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) (1 : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)))) x = x

theorem **Bridge.RingBasics.awayMap_algebraMap₂** : ∀ (l : ℕ) [Fact l.Prime] (f : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) →+* (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), Reformulated.AutPerF.AinfInv.extend l f (algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) x) = algebraMap ((Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (f x)

theorem **Bridge.φA_zA** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), (Reformulated.AutPerF.Ainf.frob l ♭(PadicComplexInt l)) ((Reformulated.AutPerF.Ainf.ofPadicInt l ♭(PadicComplexInt l)) z) = (Reformulated.AutPerF.Ainf.ofPadicInt l ♭(PadicComplexInt l)) z

theorem **Bridge.RingBasics.untilt_σF** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.Tilt l ♭(PadicComplexInt l))), ((Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ♭(PadicComplexInt l) g)) x).untilt = (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ♭(PadicComplexInt l) g) x.untilt

theorem **Bridge.RingBasics.ker_σ** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)), (Reformulated.AutPerF.AinfInv.kerTheta l ♭(PadicComplexInt l)) ≤ Ideal.comap (Reformulated.AutPerF.AinfInv.extend l ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ♭(PadicComplexInt l) g)))) ((Reformulated.AutPerF.AinfInv.kerTheta l ♭(PadicComplexInt l)))

theorem **Bridge.RingBasics.norm_eq_val** : ∀ (l : ℕ) [Fact l.Prime] (x : ℂ_[l]), ‖x‖ = ((Valued.v x : ℝ≥0∞Real) : ℝ)

theorem **Bridge.AdicMaps.padicMap_alg** : ∀ {R S : Type} [CommRing R] [CommRing S] (p : ℕ) (f : R →+* S) (r : R), (Reformulated.AutPerF.adicMap _ _ f (Reformulated.AutPerF.span_natCast_le_comap p f)) (algebraMap R (AdicCompletion (Ideal.span {(p : R)}) R) r) = algebraMap S (AdicCompletion (Ideal.span {(p : S)}) S) (f r)

theorem **Bridge.RingBasics.p_ne_zero** : ∀ (l : ℕ) [Fact l.Prime] , ((l : ℕ) : ℂ_[l]) ≠ 0

theorem **Bridge.CrisIncl.factorial_mul_partial** : ∀ {R : Type} [CommRing R] (hR : ∀ m : ℕ, m ≠ 0 → IsUnit (m : R)) (x : R) (n : ℕ), ((n.factorial : ℕ) : R) * ∑ k ∈ Finset.range n, (-1) ^ k * Ring.inverse ((k + 1 : ℕ) : R) * x ^ (k + 1) = ∑ k ∈ Finset.range n, (-1) ^ k * ((n.factorial / (k + 1) : ℕ) : R) * x ^ (k + 1)

theorem **Bridge.AdicMaps.levMap_compat** : ∀ {R S : Type} [CommRing R] [CommRing S] (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) {m n : ℕ} (hle : m ≤ n), (Ideal.Quotient.factorPow J hle).comp (((Ideal.quotientMap (J ^ n) f (Reformulated.AutPerF.pow_le_comap_pow hf n))).comp (AdicCompletion.evalₐ I n : AdicCompletion I R →+* R ⧸ I ^ n)) = ((Ideal.quotientMap (J ^ m) f (Reformulated.AutPerF.pow_le_comap_pow hf m))).comp (AdicCompletion.evalₐ I m : AdicCompletion I R →+* R ⧸ I ^ m)

theorem **Bridge.AdicMaps.evalₐ_adicMap_mk** : ∀ {R S : Type} [CommRing R] [CommRing S] (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (n : ℕ) (x : AdicCompletion I R) (r : R) (h : AdicCompletion.evalₐ I n x = Ideal.Quotient.mk _ r), AdicCompletion.evalₐ J n (Reformulated.AutPerF.adicMap I J f hf x) = Ideal.Quotient.mk _ (f r)

theorem **Bridge.CrisLoc.bdR_hom_ext** : ∀ (l : ℕ) [Fact l.Prime] {R : Type} [CommRing R] {f g : (Reformulated.AutPerF.BdR l ♭(PadicComplexInt l)) →+* R} (h : ∀ x : (Reformulated.AutPerF.BdRPlus l ♭(PadicComplexInt l)), f (algebraMap _ _ x) = g (algebraMap _ _ x)), f = g

theorem **Bridge.RingBasics.norm_gal** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : PadicAlgCl l), ‖g x‖ = ‖x‖

theorem **Bridge.AdicMaps.evalₐ_adicMap** : ∀ {R S : Type} [CommRing R] [CommRing S] (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (n : ℕ) (x : AdicCompletion I R), AdicCompletion.evalₐ J n (Reformulated.AutPerF.adicMap I J f hf x) = (Ideal.quotientMap (J ^ n) f (Reformulated.AutPerF.pow_le_comap_pow hf n)) (AdicCompletion.evalₐ I n x)

theorem **Bridge.CrisLoc.constantCoeff_root_ne_zero** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ♭(PadicComplexInt l))) (hne : e ≠ 1), WittVector.constantCoeff (WittVector.teichmuller l e - 1 : (Reformulated.AutPerF.Ainf l ♭(PadicComplexInt l))) ≠ 0

theorem **Bridge.AdicMaps.adicMap_alg** : ∀ {R S : Type} [CommRing R] [CommRing S] (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (r : R), Reformulated.AutPerF.adicMap I J f hf (algebraMap R (AdicCompletion I R) r) = algebraMap S (AdicCompletion J S) (f r)

theorem **Bridge.AdicMaps.span_le_comap** : ∀ {R S : Type} [CommRing R] [CommRing S] (p : ℕ) (f : R →+* S), Ideal.span {(p : R)} ≤ (Ideal.span {(p : S)}).comap f

theorem **Bridge.RingBasics.σC_one** : ∀ (l : ℕ) [Fact l.Prime] (x : ℂ_[l]), (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] 1) x = x

theorem **Bridge.RingBasics.isUnit_l** : ∀ (l : ℕ) [Fact l.Prime] , IsUnit ((l : ℕ) : (Reformulated.AutPerF.AinfInv l ♭(PadicComplexInt l)))

theorem **Bridge.RingBasics.norm_p_lt** : ∀ (l : ℕ) [Fact l.Prime] , ‖((l : ℕ) : ℂ_[l])‖ < 1

theorem **Bridge.RingBasics.isUnit_nat** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (hn : n ≠ 0), IsUnit ((n : ℕ) : (Reformulated.AutPerF.AinfInv l ♭(PadicComplexInt l)))

theorem **Bridge.AdicMaps.levMap_mk** : ∀ {R S : Type} [CommRing R] [CommRing S] (I : Ideal R) (J : Ideal S) (f : R →+* S) (hf : I ≤ J.comap f) (n : ℕ) (r : R), (Ideal.quotientMap (J ^ n) f (Reformulated.AutPerF.pow_le_comap_pow hf n)) (Ideal.Quotient.mk _ r) = Ideal.Quotient.mk _ (f r)

theorem **Bridge.AdicMaps.factorPow_evalₐ** : ∀ {R : Type} [CommRing R] (I : Ideal R) {m n : ℕ} (hle : m ≤ n) (x : AdicCompletion I R), Ideal.Quotient.factorPow I hle (AdicCompletion.evalₐ I n x) = AdicCompletion.evalₐ I m x

theorem **mem_acrisZero_iff** (ϖ : PFlat p R) (x : AinfInv p R) : x ∈ acrisZero p R ↔ x ∈ dpSpan (AinfInv p R) (Ainf.xi ϖ) Set.univ

theorem **p_dvd_of_eq_zero** {x : Acris p R} (hx : Acris.toBdRPlus p R x = 0) : (p : Acris p R) ∣ x

theorem **toBdRPlus_injective** : Function.Injective (Acris.toBdRPlus p R)

theorem **algebraMap_injective** : Function.Injective (algebraMap (Acris p R) (Bcris p R))

theorem **Bridge.alg_Bcris_injective** : ∀ (l : ℕ) [Fact l.Prime], Function.Injective (algebraMap (Reformulated.AutPerF.Acris l ♭(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ♭(PadicComplexInt l)))

theorem **extend_extend_inv** {p : ℕ} [Fact p.Prime] {R : Type*} [CommRing R] [Fact ¬IsUnit (p : R)] {G : Type*} [Group G] [MulSemiringAction G R] (g : G) (x : AinfInv p R) : AinfInv.extend p (Ainf.map p (MulSemiringAction.toRingHom G R g)) (AinfInv.extend p (Ainf.map p (MulSemiringAction.toRingHom G R g⁻¹)) x) = x

theorem **Bridge.RingBasics.extP_inv** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.AinfInv l ♭(PadicComplexInt l))), Reformulated.AutPerF.AinfInv.extend l ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ♭(PadicComplexInt l) g))) (Reformulated.AutPerF.AinfInv.extend l ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ♭(PadicComplexInt l) g⁻¹))) x) = x

theorem **norm_eq_val** (l : ℕ) [Fact l.Prime] (y : ℂ_[l]) : ‖y‖ = (((PadicComplex.valued l).v y : ℝ≥0∞Real) : ℝ)

theorem **mem_iff** (l : ℕ) [Fact l.Prime] (y : ℂ_[l]) : y ∈ PadicComplexInt l ↔ ‖y‖ ≤ 1

theorem **Bridge.RingBasics.mem_span_pow_OC** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (x : ♭(PadicComplexInt l)), x ∈ (Ideal.span {((l : ℕ) : ♭(PadicComplexInt l))}) ^ n ↔ ‖(x : ℂ_[l])‖ ≤ ‖((l : ℕ) : ℂ_[l])‖ ^ n

theorem **Bridge.CrisLoc.bcris_hom_ext** : ∀ (l : ℕ) [Fact l.Prime] {R : Type} [CommRing R] {f g : (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l)) →+* R} (h : ∀ x : (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)), f (algebraMap _ _ x) = g (algebraMap _ _ x)), f = g

theorem **Bridge.CrisLoc.unit_of_generator** : ∀ (l : ℕ) [Fact l.Prime] (b : (Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) (hb : (Reformulated.AutPerF.AinfInv.kerTheta l ɪ(PadicComplexInt l)) = Ideal.span {b}), IsUnit (algebraMap ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdR l ɪ(PadicComplexInt l))) (algebraMap ((Reformulated.AutPerF.AinfInv l ɪ(PadicComplexInt l))) ((Reformulated.AutPerF.BdRPlus l ɪ(PadicComplexInt l))) b))

theorem **Bridge.padicInt_hom_ext** : ∀ (l : ℕ) [Fact l.Prime] (A : Type) [CommRing A] [IsHausdorff (Ideal.span {((l : ℕ) : A)}) A] (f g : ℤ_[l] →+* A), f = g

theorem **witt_frobenius_lift** (p : ℕ) [Fact p.Prime] (k : Type*) [CommRing k] [CharP k p] [PerfectRing k p] (x : WittVector p k) : ∃ z : WittVector p k, WittVector.frobenius x = x ^ p + (p : WittVector p k) * z

theorem **Bridge.RingBasics.frob_lift** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), ∃ z : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)), (Reformulated.AutPerF.Ainf.frob l ɪ(PadicComplexInt l)) x = x ^ l + (l : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))) * z

theorem **step** {A : Type*} [CommRing A] (p n : ℕ) (z y v : A) (h : (p : A) ^ n * (z * y) = v - 1) (hv : (p : A) ^ (n + 1) ∣ v - 1) : ∃ y', (p : A) ^ (n + 1) * (z * y') = v ^ p - 1 ∧ (p : A) ^ n ∣ y' - y

theorem **exists_seq** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : ∃ y : ℕ → AcrisZero p R, ∀ n, (p : AcrisZero p R) ^ n * ((algebraMap (Ainf p R) (AcrisZero p R) u - 1) * y n) = algebraMap (Ainf p R) (AcrisZero p R) u ^ p ^ (n + 1) - 1 ∧ (p : AcrisZero p R) ^ n ∣ y (n + 1) - y n

theorem **dvd_p_mul_log** (u : Ainf p R) (hu : Ainf.theta p R u = 1) : algebraMap (Ainf p R) (Acris p R) (u - 1) ∣ (p : Acris p R) * Acris.log u hu

theorem **isUnit_logTeich_of_seq** (n : ℕ) (η : tateModule p R) (h : Tilt.seq n η.1 ≠ 1) : IsUnit (Bcris.logTeich η)

theorem **isUnit_teich_sub_one** (e : Tilt p R) (he : e.untilt = 1) (hne : e ≠ 1) : IsUnit (((algebraMap (Acris p R) (Bcris p R)).comp (algebraMap (Ainf p R) (Acris p R))) (WittVector.teichmuller p e - 1))

theorem **Bridge.CrisLoc.unit_of_root_aB** : ∀ (l : ℕ) [Fact l.Prime] (e : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))) (he : e.untilt = 1) (hne : e ≠ 1), IsUnit (((algebraMap (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Bcris l ɪ(PadicComplexInt l))).comp (algebraMap (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris l ɪ(PadicComplexInt l)))) (WittVector.teichmuller l e - 1))

theorem **Bridge.RingBasics.θ_σA** : ∀ (l : ℕ) [Fact l.Prime] (g : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : (Reformulated.AutPerF.Ainf l ɪ(PadicComplexInt l))), (WittVector.fontaineTheta ɪ(PadicComplexInt l) l) ((Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) x) = (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g) ((WittVector.fontaineTheta ɪ(PadicComplexInt l) l) x)

theorem **Bridge.CrisMaps.frob_extends** : ∀ (l : ℕ) [Fact l.Prime], Extends (Reformulated.AutPerF.Ainf.frob l ɪ(PadicComplexInt l)) (Reformulated.AutPerF.Acris.frob l ɪ(PadicComplexInt l))

theorem **Bridge.CrisMaps.gal_extends** : ∀ (l : ℕ) [Fact l.Prime] (g : PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l), Extends (Reformulated.AutPerF.Ainf.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g)) (Reformulated.AutPerF.Acris.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) g))

theorem **Bridge.RingBasics.σC_mul** : ∀ (l : ℕ) [Fact l.Prime] (g h : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)) (x : ℂ_[l]), (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] (g * h)) x = (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] g) ((MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ℂ_[l] h) x)

theorem **Bridge.RingBasics.mem_OC** : ∀ (l : ℕ) [Fact l.Prime] (x : ℂ_[l]), x ∈ PadicComplexInt l ↔ ‖x‖ ≤ 1

theorem **Bridge.RingBasics.σF_one** : ∀ (l : ℕ) [Fact l.Prime] (x : (Reformulated.AutPerF.Tilt l ɪ(PadicComplexInt l))), (Reformulated.AutPerF.Tilt.map l (MulSemiringAction.toRingHom (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l) ɪ(PadicComplexInt l) (1 : (PadicAlgCl l ≃ₐ[ℚ_[l]] PadicAlgCl l)))) x = x

Development.AbelianVarietyTheory.CrystallinePeriods.RingBasics

theorem **norm_eq_val** (x : ℂ_[p]) : ‖x‖ = ((Valued.v x : NNReal) : ℝ)

theorem **mem_OC** (x : ℂ_[p]) : x ∈ 𝒪_ℂ_[p] ↔ ‖x‖ ≤ 1

theorem **p_ne_zero** : ((p : ℕ) : ℂ_[p]) ≠ 0

theorem **norm_p_lt** : ‖((p : ℕ) : ℂ_[p])‖ < 1

theorem **norm_gal** (g : Per.Gq p) (x : Per.Qb p) : ‖g x‖ = ‖x‖

theorem **mem_span_pow_OC** (n : ℕ) (x : Per.OC p) : x ∈ (Ideal.span {((p : ℕ) : Per.OC p)}) ^ n ↔ ‖(x : ℂ_[p])‖ ≤ ‖((p : ℕ) : ℂ_[p])‖ ^ n

theorem **σC_coe** (g : Per.Gq p) (x : Per.Qb p) : Per.σC g (x : ℂ_[p]) = ((g x : Per.Qb p) : ℂ_[p])

theorem **norm_σC** (g : Per.Gq p) (x : ℂ_[p]) : ‖Per.σC g x‖ = ‖x‖

theorem **σC_one** (x : ℂ_[p]) : Per.σC (1 : Per.Gq p) x = x

theorem **σC_mul** (g h : Per.Gq p) (x : ℂ_[p]) : Per.σC (g * h) x = Per.σC g (Per.σC h x)

theorem **σF_one** (x : Per.Fl p) : Per.σF (1 : Per.Gq p) x = x

theorem **σF_mul** (g h : Per.Gq p) (x : Per.Fl p) : Per.σF (g * h) x = Per.σF g (Per.σF h x)

theorem **untilt_σF** (g : Per.Gq p) (x : Per.Fl p) : (Per.σF g x).untilt = Per.σO g x.untilt

theorem **σA_one** (x : Per.Ainf p) : Per.σA (1 : Per.Gq p) x = x

theorem **σA_mul** (g h : Per.Gq p) (x : Per.Ainf p) : Per.σA (g * h) x = Per.σA g (Per.σA h x)

theorem **θ_σA** (g : Per.Gq p) (x : Per.Ainf p) : Per.θ p (Per.σA g x) = Per.σO g (Per.θ p x)

theorem **awayMap_algebraMap₂** (f : Per.Ainf p →+* Per.Ainf p) (x : Per.Ainf p) : Per.extP f (algebraMap (Per.Ainf p) (Per.AinfP p) x) = algebraMap (Per.Ainf p) (Per.AinfP p) (f x)

theorem **extP_inv** (g : Per.Gq p) (x : Per.AinfP p) : Per.extP (Per.σA g) (Per.extP (Per.σA g⁻¹) x) = x

theorem **ker_σ** (g : Per.Gq p) : Per.kerP p ≤ Ideal.comap (Per.extP (Per.σA g)) (Per.kerP p)

theorem **exists_pow_div_factorial** (n : ℕ) : ∃ c : ℤ_[l], ((n.factorial : ℕ) : ℤ_[l]) * c = ((l : ℕ) : ℤ_[l]) ^ n

theorem **φA_apply** (x : Per.Ainf l) : Per.φA l x = WittVector.frobenius x

theorem **frob_lift** (x : Per.Ainf l) : ∃ z : Per.Ainf l, Per.φA l x = x ^ l + (l : Per.Ainf l) * z

theorem **isUnit_l** : IsUnit ((l : ℕ) : Per.AinfP l)

theorem **isUnit_nat** (n : ℕ) (hn : n ≠ 0) : IsUnit ((n : ℕ) : Per.AinfP l)

theorem **fontaineTheta_eq_zero_of_mem_ker** (x : Per.AinfP l) (hx : x ∈ Per.kerP l ^ 1) : Per.θP l x = 0

Development.AbelianVarietyTheory.CrystallinePeriods.RingIface

theorem **alg_mem_kerP** {a : Per.Ainf l} (h : Per.θ l a = 0) : algebraMap (Per.Ainf l) (Per.AinfP l) a ∈ Per.kerP l

theorem **dp** (x : Per.Ainf l) (n : ℕ) (y : Per.AinfP l) (hx : Per.θ l x = 0) (h : ((n.factorial : ℕ) : Per.AinfP l) * y = algebraMap (Per.Ainf l) (Per.AinfP l) x ^ n) : y ∈ Per.Acris0 l

theorem **le** (S : Subalgebra (Per.Ainf l) (Per.AinfP l)) (h : ∀ x : Per.Ainf l, Per.θ l x = 0 → ∀ (n : ℕ) (y : Per.AinfP l), y ∈ Per.Acris0 l → ((n.factorial : ℕ) : Per.AinfP l) * y = algebraMap (Per.Ainf l) (Per.AinfP l) x ^ n → y ∈ S) : Per.Acris0 l ≤ S

theorem **phiA_teich** (y : Per.Fl l) : Per.φA l (WittVector.teichmuller l y) = WittVector.teichmuller l (y ^ l)

theorem **sigA_teich** (g : Per.Gq l) (y : Per.Fl l) : Per.σA g (WittVector.teichmuller l y) = WittVector.teichmuller l (Per.σF g y)

theorem **sigA_coeff** (g : Per.Gq l) (w : Per.Ainf l) (n : ℕ) : (Per.σA g w).coeff n = Per.σF g (w.coeff n)

theorem **sigO_coe** (g : Per.Gq l) (y : Per.OC l) : ((Per.σO g y : Per.OC l) : ℂ_[l]) = Per.σC g (y : ℂ_[l])

theorem **sigC_cont** (g : Per.Gq l) : Continuous (Per.σC g)

theorem **tower** (x : Per.Ainf l) : algebraMap (Per.Ainf l) (Per.Acris l) x = algebraMap (Per.Acris0 l) (Per.Acris l) (algebraMap (Per.Ainf l) (Per.Acris0 l) x)

theorem **sigD_alg** (g : Per.Gq l) (b : Per.BdRp l) : Per.σD g (algebraMap (Per.BdRp l) (Per.BdR l) b) = algebraMap (Per.BdRp l) (Per.BdR l) (Per.σDp g b)

theorem **ev_alg** (n : ℕ) (a : Per.AinfP l) : Per.ev l n (algebraMap (Per.AinfP l) (Per.BdRp l) a) = Ideal.Quotient.mk _ a

theorem **ev_ext** (y y' : Per.BdRp l) (h : ∀ n, Per.ev l n y = Per.ev l n y') : y = y'

theorem **phiA_coeff** (w : Per.Ainf l) (n : ℕ) : (Per.φA l w).coeff n = w.coeff n ^ l

theorem **fontaineTheta_algebraMap** (a : Per.Ainf l) : Per.θP l (algebraMap (Per.Ainf l) (Per.AinfP l) a) = ((Per.θ l a : Per.OC l) : ℂ_[l])

theorem **alg_mem_kerP** {a : Per.Ainf l} (h : Per.θ l a = 0) : algebraMap (Per.Ainf l) (Per.AinfP l) a ∈ Per.kerP l

theorem **ev_surj** (n : ℕ) : Function.Surjective (Per.ev l n)

theorem **thetaD_alg** (a : Per.AinfP l) : Per.θD l (algebraMap (Per.AinfP l) (Per.BdRp l) a) = Per.θP l a

theorem **thetaD_of_mem_FilP** {y : Per.BdRp l} (h : y ∈ Per.FilP l 1) : Per.θD l y = 0

theorem **sigDp_alg** (g : Per.Gq l) (a : Per.AinfP l) : Per.σDp g (algebraMap (Per.AinfP l) (Per.BdRp l) a) = algebraMap (Per.AinfP l) (Per.BdRp l) (Per.extP (Per.σA g) a)

theorem **isUnit_of_generator** {a : Per.AinfP l} (h : Per.kerP l = Ideal.span {a}) : IsUnit (Per.pD l a)

theorem **exists_bdRp_equiv** : ∃ e : Per.BdRp l ≃+* AdicCompletion (Per.kerP l) (Per.AinfP l), (∀ a, e (algebraMap (Per.AinfP l) (Per.BdRp l) a) = algebraMap _ _ a) ∧ ∀ n y, AdicCompletion.evalₐ (Per.kerP l) n (e y) = Per.ev l n y

theorem **bcris_ext** {R : Type} [CommRing R] {f g : Per.Bcris l →+* R} (h : ∀ x, f (algebraMap (Per.Acris l) (Per.Bcris l) x) = g (algebraMap (Per.Acris l) (Per.Bcris l) x)) : f = g

theorem **alg_mem_FilD_nat** {n : ℕ} {z : Per.BdRp l} (h : z ∈ Per.FilP l n) : algebraMap (Per.BdRp l) (Per.BdR l) z ∈ Per.FilD l (n : ℤ)

theorem **exists_of_mem_FilD_nat** {n : ℕ} {x : Per.BdR l} (h : x ∈ Per.FilD l (n : ℤ)) : ∃ z ∈ Per.FilP l n, algebraMap (Per.BdRp l) (Per.BdR l) z = x

theorem **zB_eq_aB_zA** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.zB l z = Formalism.Per.aB l (Formalism.Per.zA l z)

theorem **mem_filtration_iff_mem_kerTheta_pow** : ∀ (l : ℕ) [Fact l.Prime] (n : ℕ) (y : Formalism.Per.BdRp l), y ∈ Formalism.Per.FilP l n ↔ Formalism.Per.ev l n y = 0

theorem **toC_alg** : ∀ (l : ℕ) [Fact l.Prime] (a : Formalism.Per.OC l), Formalism.Per.toC l (algebraMap (Formalism.Per.OC l) (Localization.Away ((l : ℕ) : Formalism.Per.OC l)) a) = (a : ℂ_[l])

theorem **padicIntMap_eq_algebraMap_comp** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.zP l z = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) (Formalism.Per.zA l z)

theorem **dividedPower_mem_Acris0** : ∀ (l : ℕ) [Fact l.Prime] (x : Formalism.Per.Ainf l) (n : ℕ) (y : Formalism.Per.AinfP l), Formalism.Per.θ l x = 0 → ((n.factorial : ℕ) : Formalism.Per.AinfP l) * y = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) x ^ n → y ∈ Formalism.Per.Acris0 l

theorem **Acris0_le_of_dividedPowers** : ∀ (l : ℕ) [Fact l.Prime] (S : Subalgebra (Formalism.Per.Ainf l) (Formalism.Per.AinfP l)), (∀ (x : Formalism.Per.Ainf l) (n : ℕ) (y : Formalism.Per.AinfP l), Formalism.Per.θ l x = 0 → ((n.factorial : ℕ) : Formalism.Per.AinfP l) * y = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) x ^ n → y ∈ S) → Formalism.Per.Acris0 l ≤ S

theorem **aB_algebraMap_padicInt** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.aB l (algebraMap ℤ_[l] (Formalism.BasicNotions.Loc.Ainf l) z) = Formalism.Per.zB l z

theorem **zD_eq_alg** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.zD l z = algebraMap (Formalism.Per.BdRp l) (Formalism.Per.BdR l) (algebraMap (Formalism.Per.AinfP l) (Formalism.Per.BdRp l) (algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) (Formalism.Per.zA l z)))

theorem **aB_eq_alg** : ∀ (l : ℕ) [Fact l.Prime] (a : Formalism.Per.Ainf l), Formalism.Per.aB l a = algebraMap (Formalism.Per.Acris l) (Formalism.Per.Bcris l) (algebraMap (Formalism.Per.Ainf l) (Formalism.Per.Acris l) a)

theorem **aB_zl** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.aB l (algebraMap ℤ_[l] (Formalism.BasicNotions.Loc.Ainf l) z) = Formalism.Per.zB l z

theorem **zD_alg** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.zD l z = algebraMap (Formalism.Per.BdRp l) (Formalism.Per.BdR l) (algebraMap (Formalism.Per.AinfP l) (Formalism.Per.BdRp l) (algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) (Formalism.Per.zA l z)))

theorem **mem_Acris0_of_factorial_mul** : ∀ (l : ℕ) [Fact l.Prime] (x : Formalism.Per.Ainf l), Formalism.Per.θ l x = 0 → ∀ (n : ℕ) (y : Formalism.Per.AinfP l), ((n.factorial : ℕ) : Formalism.Per.AinfP l) * y = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) x ^ n → y ∈ Formalism.Per.Acris0 l

theorem **aB_apply** : ∀ (l : ℕ) [Fact l.Prime] (a : Formalism.Per.Ainf l), Formalism.Per.aB l a = algebraMap (Formalism.Per.Acris l) (Formalism.Per.Bcris l) (algebraMap (Formalism.Per.Ainf l) (Formalism.Per.Acris l) a)

theorem **aB_alg** : ∀ (l : ℕ) [Fact l.Prime] (w : Formalism.Per.Ainf l), Formalism.Per.aB l w = algebraMap (Formalism.Per.Acris l) (Formalism.Per.Bcris l) (algebraMap (Formalism.Per.Ainf l) (Formalism.Per.Acris l) w)

theorem **aB_zA** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.aB l (Formalism.Per.zA l z) = Formalism.Per.zB l z

theorem **pD_eq_alg** : ∀ (l : ℕ) [Fact l.Prime] (a : Formalism.Per.AinfP l), Formalism.Per.pD l a = algebraMap (Formalism.Per.BdRp l) (Formalism.Per.BdR l) (algebraMap (Formalism.Per.AinfP l) (Formalism.Per.BdRp l) a)

theorem **alg_mem_FilD_zero** : ∀ (l : ℕ) [Fact l.Prime] (y : Formalism.Per.BdRp l), algebraMap (Formalism.Per.BdRp l) (Formalism.Per.BdR l) y ∈ Formalism.Per.FilD l 0

theorem **zB_alg** : ∀ (l : ℕ) [Fact l.Prime] (z : ℤ_[l]), Formalism.Per.zB l z = algebraMap (Formalism.Per.Acris l) (Formalism.Per.Bcris l) (algebraMap (Formalism.Per.Ainf l) (Formalism.Per.Acris l) (Formalism.Per.zA l z))

theorem **acris0_le** : ∀ (l : ℕ) [Fact l.Prime] (S : Subalgebra (Formalism.Per.Ainf l) (Formalism.Per.AinfP l)), (∀ x : Formalism.Per.Ainf l, Formalism.Per.θ l x = 0 → ∀ (n : ℕ) (y : Formalism.Per.AinfP l), ((n.factorial : ℕ) : Formalism.Per.AinfP l) * y = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) x ^ n → y ∈ S) → Formalism.Per.Acris0 l ≤ S

theorem **acris0_le_3** : ∀ (l : ℕ) [Fact l.Prime] (S : Subalgebra (Formalism.Per.Ainf l) (Formalism.Per.AinfP l)), (∀ x : Formalism.Per.Ainf l, Formalism.Per.θ l x = 0 → ∀ (n : ℕ) (y : Formalism.Per.AinfP l), y ∈ Formalism.Per.Acris0 l → ((n.factorial : ℕ) : Formalism.Per.AinfP l) * y = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) x ^ n → y ∈ S) → Formalism.Per.Acris0 l ≤ S

theorem **dividedPower_mem** : ∀ (l : ℕ) [Fact l.Prime] (m : Formalism.Per.Ainf l) (j : ℕ) (y : Formalism.Per.AinfP l), Formalism.Per.θ l m = 0 → ((j.factorial : ℕ) : Formalism.Per.AinfP l) * y = algebraMap (Formalism.Per.Ainf l) (Formalism.Per.AinfP l) m ^ j → y ∈ Formalism.Per.Acris0 l

theorem **ev_sigDp** : ∀ (l : ℕ) [Fact l.Prime] (g : Formalism.Per.Gq l) (n : ℕ) (y : Formalism.Per.BdRp l) (a : Formalism.Per.AinfP l), Formalism.Per.ev l n y = Ideal.Quotient.mk _ a → Formalism.Per.ev l n (Formalism.Per.σDp g y) = Ideal.Quotient.mk _ (Formalism.Per.extP (Formalism.Per.σA g) a)

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.ExistsGrpObjIsAddLawHom

theorem **isAddLawHom_mul** : IsAddLawHom W μ[schemeOver W]

theorem **IsAddLawHom.unique** {m m' : schemeOver W ⊗ schemeOver W ⟶ schemeOver W} (h : IsAddLawHom W m) (h' : IsAddLawHom W m') : m = m'

theorem **ptHom_grpObj_inv** {A : Type} [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) : ptHom W ρ P hP ≫ (GrpObj.inv : schemeOver W ⟶ schemeOver W).left = ptHom W ρ _ hP.neg

theorem **exists_grpObj_isAddLawHom** (hΔ : IsUnit W.Δ) : ∃ _ : GrpObj (schemeOver W), IsCommMonObj (schemeOver W) ∧ η[schemeOver W].left = zero W ∧ IsAddLawHom W μ[schemeOver W] ∧ ∀ (A : Type) [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P), ∃ hN : IsUnimodularSol (W.map ρ) ((W.map ρ).toProjective.neg P), ptHom W ρ P hP ≫ (GrpObj.inv : schemeOver W ⟶ schemeOver W).left = ptHom W ρ _ hN

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.ExistsIsLocallyFreeDataUnique

@[simp] theorem **sectionsEquiv_apply** (s : Γ(lbSheaf M, V)) : sectionsEquiv M V s = s

theorem **isQuasicoherent** (hM : M.IsLB) : (lbSheaf M).IsQuasicoherent

theorem **exists_restrict_iso** (hM : M.IsLB) (x : X) : ∃ (U : X.Opens) (_ : x ∈ U), Nonempty (unitMod U.toScheme ≅ (lbSheaf M).restrict U.ι)

theorem **existsUnique_gen** (hM : M.IsLB) (hM' : M'.IsLB) (φ : lbSheaf M ≅ lbSheaf M') : ∃! w : LB.Gen (LB.div M' M), ∃ hw : w.IsGen, isoOfGen hM hM' w hw = φ

@[simp] theorem **genEquivIso_apply** (hM : M.IsLB) (hM' : M'.IsLB) (w : {w : LB.Gen (LB.div M' M) // w.IsGen}) : genEquivIso hM hM' w = isoOfGen hM hM' w.1 w.2

theorem **triv_div_iff** (hM : M.IsLB) (hM' : M'.IsLB) : Triv (LB.div M' M) ↔ Nonempty (lbSheaf M ≅ lbSheaf M')

theorem **triv_iff** (hM : M.IsLB) : Triv M ↔ Nonempty (lbSheaf M ≅ unitMod X)

theorem **triv_pull_iff** (hM : M.IsLB) (π : X' ⟶ X) : Triv (M.pull π) ↔ Nonempty ((Scheme.Modules.pullback π).obj (lbSheaf M) ≅ unitMod X')

theorem **exists_iso_iff** (F : X.Modules) : (∃ M : LB X, M.IsLB ∧ Nonempty (lbSheaf M ≅ F)) ↔ ∀ x : X, ∃ (U : X.Opens) (_ : x ∈ U), Nonempty (unitMod U.toScheme ≅ F.restrict U.ι)

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.ExistsUniqueEqConstMulEllTheta

theorem **appLE_originSec_eq_zero_iff_eSec** (T : Scheme.{0}) (t : T ⟶ (schemeOver W).left) (i : (originLB W).ι) (h : ⊤ ≤ t ⁻¹ᵁ (originLB W).U i) : (t.appLE ((originLB W).U i) ⊤ h).hom ((originSec W).f i) = 0 ↔ t = t ≫ (schemeOver W).hom ≫ eSec (schemeOver W)

theorem **rigidified_originLB** : Out.Rigidified (schemeOver W) (originLB W)

theorem **sym_originLB** : Out.Sym (schemeOver W) (originLB W)

theorem **kTriv_originLB** : Out.KTriv (schemeOver W) (originLB W)

theorem **ellPolAV_A** : (ellPolAV W).A = schemeOver W

theorem **ellPolAV_hom** : (ellPolAV W).A.hom = toSpec W

theorem **ellPolAV_grp** : (ellPolAV W).grp = grpObj W

@[simp] theorem **ellPolAV_g** : (ellPolAV W).g = 1

theorem **ellPolAV_L** : (ellPolAV W).L = originLB W

theorem **eSec_ellPolAV** : eSec (ellPolAV W).A = zero W

theorem **isAddLawHom_ellPolAV** : IsAddLawHom W μ[(ellPolAV W).A]

theorem **ptHom_ellPolAV_inv** {A : Type} [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) : ∃ hN : IsUnimodularSol (W.map ρ) ((W.map ρ).toProjective.neg P), ptHom W ρ P hP ≫ (GrpObj.inv : (ellPolAV W).A ⟶ (ellPolAV W).A).left = ptHom

W ρ _ hN

theorem **ellTheta_eq** : ellTheta W = originSec W

theorem **nonzeroLocus_ellTheta** : Out.nonzeroLocus (ellTheta W) = coordOpen W 2

theorem **secPull_ellTheta_eq_zero_iff** {T : Scheme.{0}} (t : T ⟶ (ellPolAV W).A.left) : (∀ j, (Cartier.secPull (ellTheta W) t).f j = 0) ↔ t = t ≫ (ellPolAV W).A.hom ≫ eSec (ellPolAV W).A

theorem **existsUnique_eq_const_mul_ellTheta** (s : (ellPolAV W).L.Sec) : ∃! c : R, ∀ i, s.f i = constFn (ellPolAV W).A.hom ((ellPolAV W).L.U i) c ∗ (ellTheta W).f i

theorem **exists_isLieBasis_ellPolAV** : ∃ v : epsR R ⟶ (ellPolAV W).A, (∃ h : IsUnimodularSol (W.map (algebraMap R (DualNumber R))) ![DualNumber.eps, -1, 0], v.left = ptHom W (algebraMap R (DualNumber R)) ![DualNumber.eps, -1, 0] h) ∧ IsLieBasis ![v]

theorem **isGrpPullback_ellPolAV** : IsGrpPullback (ellPolAV W').A (ellPolAV W).A (spMap f) (mapHom W f h)

theorem **lbIso_ellPolAV** : LBIso (ellPolAV W').L ((ellPolAV W).L.pull (mapHom W f h))

theorem **ellPolAV_baseChange** : (ellPolAV W').g = (ellPolAV W).g ∧ IsGrpPullback (ellPolAV W').A (ellPolAV W).A (spMap f) (mapHom W f h) ∧ LBIso ((ellPolAV W).L.pull (mapHom W f h)) (ellPolAV W').L

theorem **not_isGen_ellPolAV** [Nontrivial R] (t : (ellPolAV W).L.Gen) : ¬ t.IsGen

theorem **weierstrass_0_0_1_neg_one_0_isElliptic** : (⟨0, 0, 1, -1, 0⟩ : _root_.WeierstrassCurve ℚ).IsElliptic

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.ExistsUniqueEqConstMulOriginSec

theorem **twoChartFn_compat** (n : ℕ) {a : Γ(X, U₀)} {b : Γ(X, U₁)} (h : resG X inf_le_left a = (u : Γ(X, U₀ ⊓ U₁)) ^ n ∗ resG X inf_le_right b) (i j : Bool) : resG X (inf_le_left : (twoChart U₀ U₁ u).U i ⊓ (twoChart U₀ U₁ u).U j ≤ _) (twoChartFn u a b i) = ((twoChart U₀ U₁ u).g i j).val ^ n ∗ resG X inf_le_right (twoChartFn u a b j)

theorem **resG_injective_of_eq_basicOpen** {U V : X.Opens} (hU : IsAffineOpen U) {a : Γ(X, U)} (ha : a ∈ nonZeroDivisors Γ(X, U)) (hV : V = X.basicOpen a) (h : V ≤ U) : Function.Injective (resG X h)

theorem **appLE_injective_of_isIso** (e : Y ⟶ X) [IsIso e] {U : X.Opens} {V : Y.Opens} (hV : V = e ⁻¹ᵁ U) : Function.Injective (e.appLE U V hV.le).hom

theorem **evalPt_eq_zero_iff** {A : Type} [CommRing A] (x : Sp A ⟶ X) {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (a : Γ(X, U)) : evalPt x U h a = 0 ↔ (x.app U).hom a = 0

theorem **wchart_le_const** (L : LB X) (φ : Y ⟶ X) (n : ℕ) (k : Fin (n + 1) → L.ι) : LB.wchart L (fun _ : Fin (n + 1) => φ) k ≤ LB.wchart L (fun _ : Fin (n + 1) => φ) k ⊓ LB.wchart L (fun _ : Fin (n + 1) => φ) fun _ => k 0

theorem **exists_gs_finPowSec** {L : LB X} (hL : L.IsLB) {n : ℕ} (φ : Y ⟶ X) (y : Y) (k : Fin (n + 1) → L.ι) (hk : y ∈ LB.wchart L (fun _ : Fin (n + 1) => φ) k) : ∃ v : (Y.presheaf.stalk y)ˣ, ∀ s : (lbPow L (n + 1)).Sec, gs (𝟙 Y) y _ ((finPowSec hL s φ).f k) = v.val ∗ gs φ y _ (powSecFn s (k 0))

theorem **nonzeroLocus_finPowSec** {L : LB X} (hL : L.IsLB) {n : ℕ} (s : (lbPow L (n + 1)).Sec) (φ : Y ⟶ X) : nonzeroLocus (finPowSec hL s φ) = φ ⁻¹ᵁ nonzeroLocus s

theorem **eval₂Hom_surjective_of_isPushout** {A B C D : CommRingCat.{0}} {a : A ⟶ B} {b : A ⟶ C} {inl : B ⟶ D} {inr : C ⟶ D} (po : IsPushout a b inl inr) {R σ : Type} [CommRing R] (c : R →+* A) (r : σ → B) (h : Function.Surjective (MvPolynomial.eval₂Hom (a.hom.comp c) r)) : Function.Surjective (MvPolynomial.eval₂Hom inr.hom fun j => inl.hom (r j))

theorem **resG_originCofactor** (hU : U ≤ coordOpen W 1) (h : V ≤ U) : resG _ h (originCofactor W U hU) = originCofactor W V (h.trans hU)

theorem **ratioOn_mul_originCofactor** (hU : U ≤ coordOpen W 1) : ratioOn W U 1 hU 2 ∗ originCofactor W U hU = ratioOn W U 1 hU 0 ^ 3

theorem **originOpen_le** : originOpen W ≤ coordOpen W 1

theorem **isAffineOpen_originOpen** : IsAffineOpen (originOpen W)

theorem **isUnit_originCofactor** {U : (scheme W).Opens} (h : U ≤ originOpen W) : IsUnit (originCofactor W U (h.trans (originOpen_le W)))

theorem **originOpen_sup_coordOpen_two** : originOpen W ⊔ coordOpen W 2 = ⊤

theorem **isUnit_ratioOn_one_zero** {U : (scheme W).Opens} (h₀ : U ≤ originOpen W) (h₁ : U ≤ coordOpen W 2) : IsUnit (ratioOn W U 1 (h₀.trans (originOpen_le W)) 0)

theorem **basicOpen_originParam** : (scheme W).basicOpen (originParam W) = originOpen W ⊓ coordOpen W 2

theorem **isRegularFn_originParam** : Cartier.IsRegularFn (originParam W)

theorem **resG_originOpen_injective** : Function.Injective (resG _ (inf_le_left : originOpen W ⊓ coordOpen W 2 ≤ originOpen W))

@[simp] theorem **val_originUnit** : (originUnit W : Γ(scheme W, originOpen W ⊓ coordOpen W 2)) = ratioOn W _ 1 (inf_le_left.trans (originOpen_le W)) 0

@[simp] theorem **originLB_U_true** : (originLB W).U true = originOpen W

@[simp] theorem **originLB_U_false** : (originLB W).U false = coordOpen W 2

@[simp] theorem **originLB_g_true_false** : (originLB W).g true false = originUnit W

theorem **originLB_isLB** : (originLB W).IsLB

@[simp] theorem **originSec_f_true** : (originSec W).f true = originParam W

@[simp] theorem **originSec_f_false** : (originSec W).f false = (1 : Γ(scheme W, coordOpen W 2))

theorem **isRegularSec_originSec** : Cartier.IsRegularSec (originSec W) | true => isRegularFn_originParam W | false => .of_isUnit isUnit_one theorem nonzeroLocus_originSec : nonzeroLocus (originSec W) = coordOpen W 2

theorem **evalPt_zero_ratioOn** {U : (scheme W).Opens} (hU : U ≤ coordOpen W 1) (h : ⊤ ≤ zero W ⁻¹ᵁ U) (j : Fin 3) : evalPt (zero W) U h (ratioOn W U 1 hU j) = ![0, 1, 0] j

theorem **evalPt_zero_constOn** {U : (scheme W).Opens} (h : ⊤ ≤ zero W ⁻¹ᵁ U) (r : R) : evalPt (zero W) U h (constOn W U r) = r

theorem **evalPt_zero_originCofactor** {U : (scheme W).Opens} (hU : U ≤ coordOpen W 1) (h : ⊤ ≤ zero W ⁻¹ᵁ U) : evalPt (zero W) U h (originCofactor W U hU) = 1

theorem **zero_preimage_originOpen** : zero W ⁻¹ᵁ originOpen W = ⊤

theorem **originValue_originParam** : originValue W (originParam W) = 0

theorem **eq_spMap_zero_of_evalPt** {A : Type} [CommRing A] {x : Sp A ⟶ scheme W} {ρ : R →+* A} (hx : x ≫ toSpec W = spMap ρ) (h : ⊤ ≤ x ⁻¹ᵁ originOpen W) (hz : evalPt x _ h (originParam W) = 0) : x = spMap ρ ≫ zero W

theorem **eq_comp_zero_iff** {T : Scheme.{0}} (g : T ⟶ scheme W) : g = g ≫ toSpec W ≫ zero W ↔ ⊤ ≤ g ⁻¹ᵁ originOpen W ∧ (g.app (originOpen W)).hom (originParam W) = 0

theorem **secPull_originSec_eq_zero_iff** {T : Scheme.{0}} (g : T ⟶ scheme W) : (∀ j, (Cartier.secPull (originSec W) g).f j = 0) ↔ g = g ≫ toSpec W ≫ zero W

theorem **coordFn_overlap** (j : Fin 3) : resG _ (inf_le_left : originOpen W ⊓ coordOpen W 2 ≤ _) (originCofactor W _ (originOpen_le W) * ratioOn W _ 1 (originOpen_le W) j) = (originUnit W : Γ(scheme W, originOpen W ⊓ coordOpen W 2)) ^ 3 * resG _ inf_le_right (ratio W 2 j)

theorem **coordFn_compat** (j : Fin 3) (i k : (originLB W).ι) : resG _ (inf_le_left : (originLB W).U i ⊓ (originLB W).U k ≤ _) (coordFn W j i) = ((originLB W).g i k).val ^ 3 * resG _ inf_le_right (coordFn W j k)

@[simp] theorem **powSecFn_coordSec** (j : Fin 3) (i : (originLB W).ι) : powSecFn (coordSec W j) i = coordFn W j i

theorem **basicOpen_coordFn** (j : Fin 3) : ∀ i, (scheme W).basicOpen (coordFn W j i) = (originLB W).U i ⊓ coordOpen W j | true => by change (scheme W).basicOpen (originCofactor W _ (originOpen_le W) * ratioOn W _ 1 (originOpen_le W) j) = originOpen W ⊓ coordOpen W j rw [Scheme.basicOpen_mul, basicOpen_ratioOn, (scheme W).basicOpen_of_isUnit (isUnit_originCofactor le_rfl)] exact inf_eq_right.2 inf_le_left | false => basicOpen_ratioOn le_rfl theorem nonzeroLocus_coordSec (j : Fin 3) : nonzeroLocus (coordSec W j) = coordOpen W j

theorem **resG_coordFn** {V : (scheme W).Opens} {i : Fin 3} (hi : V ≤ coordOpen W i) (j : Fin 3) : ∀ (k : (originLB W).ι) (hk : V ≤ (originLB W).U k), resG _ hk (coordFn W j k) = ratioOn W V i hi j * resG _ hk (coordFn W i k) | true, hk => by have e : ∀ l, resG _ hk (coordFn W l true) = originCofactor W V (hk.trans (originOpen_le W)) * ratioOn W V 1 (hk.trans (originOpen_le W)) l

theorem **gs_coordFn** {Z : Scheme.{0}} (r : Z ⟶ scheme W) (z : Z) {i : Fin 3} (j : Fin 3) {k : (originLB W).ι} (hi : r z ∈ coordOpen W i) (hk : r z ∈ (originLB W).U k) : gs r z _ (coordFn W j k) = gs r z _ (ratio W i j) * gs r z _ (coordFn W i k)

theorem **exists_eq_constOn** (f₀ : Γ(scheme W, originOpen W)) (f₁ : Γ(scheme W, coordOpen W 2)) (h : resG _ inf_le_left f₀ = (originUnit W : Γ(scheme W, originOpen W ⊓ coordOpen W 2)) * resG _ inf_le_right f₁) : ∃ c : R, f₀ = constOn W _ c * originParam W ∧ f₁ = constOn W _ c

theorem **constOn_coordOpen_two_injective** : Function.Injective (constOn W (coordOpen W 2))

theorem **existsUnique_eq_const_mul_originSec** (s : (originLB W).Sec) : ∃! c : R, ∀ i, s.f i = constFn (toSpec W) _ c * (originSec W).f i

theorem **mapHom_app_originCofactor** {U : (scheme W).Opens} (hU : U ≤ coordOpen W 1) : ((mapHom W f h).app U).hom (originCofactor W U hU) = originCofactor W' (mapHom W f h ⁻¹ᵁ U) (((mapHom W f h).preimage_mono hU).trans (mapHom_preimage_coordOpen W f h 1).le)

theorem **mapHom_preimage_originOpen** : mapHom W f h ⁻¹ᵁ originOpen W = originOpen W'

theorem **mapHom_preimage_originLB_U** : ∀ i : Bool, mapHom W f h ⁻¹ᵁ (originLB W).U i = (originLB W').U i | true => mapHom_preimage_originOpen W f h | false => mapHom_preimage_coordOpen W f h 2 theorem mapHom_app_originSec : ∀ i : Bool, ((mapHom W f h).app ((originLB W).U i)).hom ((originSec W).f i) = resG _ (mapHom_preimage_originLB_U W f h i).le ((originSec W').f i) | true => mapHom_app_ratioOn W f h (originOpen_le W) 0 | false => (map_one _).trans (map_one _).symm theorem gs_mapHom_originSec (i : Bool) (y : scheme W') (hy : y ∈ (originLB W').U i) : gs (mapHom W f h) y _ ((originSec W).f i) = gs (𝟙 _) y _ ((originSec W').f i)

theorem **existsUnique_iso_pull_originLB** : ∃! w : LB.Iso (originLB W') ((originLB W).pull (mapHom W f h)), w.IsGen ∧ Cartier.SecMaps w (originSec W') (Cartier.secPull (originSec W) (mapHom W f h))

theorem **lbIso_pull_originLB** : LBIso (originLB W') ((originLB W).pull (mapHom W f h))

theorem **existsUnique_eq_const_mul_secPull** {R' : Type} [CommRing R'] {f : R →+* R'} {Y : Scheme.{0}} {φ : Y ⟶ scheme W} {p : Y ⟶ Sp R'} (H : IsPullback φ p (toSpec W) (spMap f)) (s : ((originLB W).pull φ).Sec) : ∃! c : R', ∀ j, s.f j = constFn p _ c * (Cartier.secPull (originSec W) φ).f j

theorem **existsUnique_eq_appLE_mul_secPull** {T Y : Scheme.{0}} [IsAffine T] {t : T ⟶ Sp R} {φ : Y ⟶ scheme W} {p : Y ⟶ T} (H : IsPullback φ p (toSpec W) t) (s : ((originLB W).pull φ).Sec) : ∃! c : Γ(T, ⊤), ∀ j, s.f j = (p.appLE ⊤ _ le_top).hom c * (Cartier.secPull (originSec W) φ).f j

theorem **isGen_one_pull_zero** : LB.Gen.IsGen (L

theorem **relAmple_originLB** : RelAmple (schemeOver W) (originLB W)

theorem **not_isGen_originLB** [Nontrivial R] (t : (originLB W).Gen) : ¬ t.IsGen

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.FirstCohomologyVanishesSpecOfIsLocalRing

theorem **le_pull** (M : LB X) (π : X' ⟶ X) {W : X.Opens} {i : M.ι} (h : W ≤ M.U i) : π ⁻¹ᵁ W ≤ (M.pull π).U (fun _ => i)

theorem **rs_rs** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (x : Γ(X, U)) : rs X h₂ (rs X h₁ x) = rs X (h₂.trans h₁) x

theorem **rs_self** {U : X.Opens} (h : U ≤ U) (x : Γ(X, U)) : rs X h x = x

theorem **rs_cst** (f : X ⟶ Spec (CommRingCat.of R)) {U V : X.Opens} (h : V ≤ U) (r : R) : rs X h (cst f U r) = cst f V r

theorem **isLB_pw** {H : LB X} (hH : H.IsLB) (t : ℤ) : (pw H t).IsLB where

theorem **isLB_tw** {M H : LB X} (hM : M.IsLB) (hH : H.IsLB) (t : ℤ) : (tw M H t).IsLB

theorem **pw_one** (H : LB X) : pw H 1 = H

theorem **tw_g** (M H : LB X) (t : ℤ) (p q : M.ι × H.ι) : (tw M H t).g p q = res (le_inf (inf_le_left.trans inf_le_left) (inf_le_right.trans inf_le_left) : M.U p.1 ⊓ H.U p.2 ⊓ (M.U q.1 ⊓ H.U q.2) ≤ M.U p.1 ⊓ M.U q.1) (M.g p.1 q.1) * res (le_inf (inf_le_left.trans inf_le_right) (inf_le_right.trans inf_le_right) : M.U p.1 ⊓ H.U p.2 ⊓ (M.U q.1 ⊓ H.U q.2) ≤ H.U p.2 ⊓ H.U q.2) (H.g p.2 q.2) ^ t

theorem **firstCohomologyVanishes_of_every_cover_trivial** (M : LB X) (hX : ∀ (I : Type) (V : I → X.Opens), ⨆ a, V a = ⊤ → ∃ a, V a = ⊤) : FirstCohomologyVanishes M

theorem **firstCohomologyVanishes_spec_of_isLocalRing** [IsLocalRing R] (M : LB (Spec (CommRingCat.of R))) : FirstCohomologyVanishes M

theorem **cst_bijective** (f : X ⟶ Spec (CommRingCat.of R)) [IsIso f] {W : X.Opens} (hW : W = ⊤) : Function.Bijective (cst f W)

theorem **triv_trivLB** (X : Scheme.{0}) : Triv (Inh.trivLB X)

theorem **nonzeroLocus_oneSec** (X : Scheme.{0}) : nonzeroLocus (oneSec X) = ⊤

theorem **veryAmpleBy_oneSec** (f : X ⟶ Spec (CommRingCat.of R)) [IsIso f] : VeryAmpleBy f (Inh.trivLB X) (m

theorem **isProjBy_of_isIso** (f : X ⟶ Spec (CommRingCat.of R)) [IsIso f] : IsProjBy f

theorem **isProjBy_id** (R : Type) [CommRing R] : IsProjBy (𝟙 (Spec (CommRingCat.of R)))

theorem **pull_trivLB_U** (π : X' ⟶ X) (j : ((Inh.trivLB X).pull π).ι) : ((Inh.trivLB X).pull π).U j = ⊤

theorem **cstSec_bijective** (π : X' ⟶ X) (f' : X' ⟶ Spec (CommRingCat.of R)) [IsIso f'] : Function.Bijective (cstSec π f')

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.IsScalarTowerOfAlgebraMapEq

theorem **equivFamilies_val** (s : SecOn M V) : (equivFamilies M V s).1 = coord s

theorem **compat** (s : SecOn M V) (i j : M.ι) {O : X.Opens} (hi : O ≤ V ⊓ M.U i) (hj : O ≤ V ⊓ M.U j) (hij : O ≤ M.U i ⊓ M.U j) : rs X hi (coord s i) = rs X hij (M.g i j).val * rs X hj (coord s j)

theorem **restrict_eq_map** (h : W ≤ V) (s : SecOn M V) : restrict M h s = (toMod M).presheaf.map (homOfLE h).op s

@[simp] theorem **coord_restrict** (h : W ≤ V) (s : SecOn M V) (i : M.ι) : coord (restrict M h s) i = rs X (inf_le_inf_right _ h) (coord s i)

@[simp] theorem **restrict_self** (h : V ≤ V) (s : SecOn M V) : restrict M h s = s

@[simp] theorem **restrict_restrict** {O : X.Opens} (h₁ : W ≤ V) (h₂ : O ≤ W) (s : SecOn M V) : restrict M h₂ (restrict M h₁ s) = restrict M (h₂.trans h₁) s

theorem **restrict_bijective** (h : W ≤ V) (h' : V ≤ W) : Function.Bijective (restrict M h)

theorem **ext_of_cover** {I : Type} (O : I → X.Opens) (hO : ∀ a, O a ≤ V) (hc : V ≤ ⨆ a, O a) {s t : SecOn M V} (h : ∀ a, restrict M (hO a) s = restrict M (hO a) t) : s = t

theorem **existsUnique_glue** {I : Type} (O : I → X.Opens) (hO : ∀ a, O a ≤ V) (hc : V ≤ ⨆ a, O a) (s : ∀ a, SecOn M (O a)) (h : ∀ a b, restrict M (inf_le_left : O a ⊓ O b ≤ O a) (s a) = restrict M (inf_le_right : O a ⊓ O b ≤ O b) (s b)) : ∃! t : SecOn M V, ∀ a, restrict M (hO a) t = s a

theorem **topEquiv_f** (s : SecOn M ⊤) (i : M.ι) : (topEquiv M s).f i = rs X (le_inf le_top le_rfl) (coord s i)

theorem **topEquiv_symm_apply** (s : LB.Sec M) : (topEquiv M).symm s = ofSec s ⊤

@[simp] theorem **restrict_ofSec** (s : LB.Sec M) (h : W ≤ V) : restrict M h (ofSec s V) = ofSec s W

theorem **ofSec_bijective** (h : V = ⊤) : Function.Bijective fun s : LB.Sec M => (ofSec s V : SecOn M V)

theorem **chart_apply** (hM : M.IsLB) (i : M.ι) (h : V ≤ M.U i) (s : SecOn M V) : chart hM i h s = rs X (le_inf le_rfl h) (coord s i)

theorem **coord_chart_symm** (hM : M.IsLB) (i j : M.ι) (h : V ≤ M.U i) (a : Γ(X, V)) : coord ((chart hM i h).symm a) j = rs X inf_le_left a * rs X (le_inf inf_le_right (inf_le_left.trans h) : V ⊓ M.U j ≤ M.U j ⊓ M.U i) (M.g j i).val

theorem **chart_restrict** (hM : M.IsLB) (i : M.ι) (h : V ≤ M.U i) (h' : W ≤ V) (s : SecOn M V) : chart hM i (h'.trans h) (restrict M h' s) = rs X h' (chart hM i h s)

theorem **chart_change** (hM : M.IsLB) (i j : M.ι) (hi : V ≤ M.U i) (hj : V ≤ M.U j) (s : SecOn M V) : chart hM i hi s = rs X (le_inf hi hj) (M.g i j).val * chart hM j hj s

theorem **existsUnique_of_local** (hM : M.IsLB) {I : Type} (O : I → X.Opens) (hO : ∀ a, O a ≤ V) (hc : V ≤ ⨆ a, O a) (c : I → M.ι) (hU : ∀ a, O a ≤ M.U (c a)) (x : ∀ a, Γ(X, O a)) (h : ∀ a b, rs X (inf_le_left : O a ⊓ O b ≤ O a) (x a) = rs X (inf_le_inf (hU a) (hU b)) (M.g (c a) (c b)).val * rs X (inf_le_right : O a ⊓ O b ≤ O b) (x b)) : ∃! t : SecOn M V, ∀ a, rs X (le_inf (hO a) (hU a)) (coord t (c a)) = x a

theorem **coord_congr** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) (s : SecOn L V) (p : M.ι) (q : L.ι) {O : X.Opens} (hp : O ≤ V ⊓ M.U p) (hq : O ≤ V ⊓ L.U q) (h : O ≤ M.U p ⊓ L.U q) : rs X hp (coord (congr hL hM w hw V s) p) = rs X h (show Γ(X, M.U p ⊓ L.U q)ˣ from w.s (p, q)).val * rs X hq (coord s q)

theorem **congr_restrict** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) (h : W ≤ V) (s : SecOn L V) : congr hL hM w hw W (restrict L h s) = restrict M h (congr hL hM w hw V s)

theorem **coord_pull** (π : X' ⟶ X) (s : SecOn M V) (j : (M.pull π).ι) : coord (pull M π V s) j = (π.appLE _ _ (pull_inf_le π V j)).hom (coord s (j PUnit.unit))

theorem **coord_pullLE** (π : X' ⟶ X) {V' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (s : SecOn M V) (j : (M.pull π).ι) : coord (pullLE M π V V' e s) j = (π.appLE (V ⊓ M.U (j PUnit.unit)) (V' ⊓ (M.pull π).U j) ((inf_le_inf_right _ e).trans (pull_inf_le π V j))).hom (coord s (j PUnit.unit))

theorem **pullLE_self** (π : X' ⟶ X) (s : SecOn M V) : pullLE M π V _ le_rfl s = pull M π V s

theorem **pull_restrict** (π : X' ⟶ X) (h : W ≤ V) (s : SecOn M V) : pull M π W (restrict M h s) = restrict (M.pull π) (π.preimage_mono h) (pull M π V s)

theorem **restrict_pullLE** (π : X' ⟶ X) {V' W' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (h : W' ≤ V') (s : SecOn M V) : restrict (M.pull π) h (pullLE M π V V' e s) = pullLE M π V W' (h.trans e) s

theorem **pullLE_restrict** (π : X' ⟶ X) {W' : X'.Opens} (h : W ≤ V) (e : W' ≤ π ⁻¹ᵁ W) (s : SecOn M V) : pullLE M π W W' e (restrict M h s) = pullLE M π V W' (e.trans (π.preimage_mono h)) s

theorem **chart_pullLE** (hM : M.IsLB) (π : X' ⟶ X) {V' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (j : (M.pull π).ι) (h : V ≤ M.U (j PUnit.unit)) (h' : V' ≤ (M.pull π).U j) (s : SecOn M V) : chart (LB.isLB_pull hM π) j h' (pullLE M π V V' e s) = (π.appLE V V' e).hom (chart hM (j PUnit.unit) h s)

theorem **tw_ext** {x y : SecOn (tw M H t) V} (h : ∀ p k, twCoord x p k = twCoord y p k) : x = y

@[simp] theorem **twCoord_add** (x y : SecOn (tw M H t) V) (p : M.ι) (k : H.ι) : twCoord (x + y) p k = twCoord x p k + twCoord y p k

@[simp] theorem **twCoord_smul** (a : Γ(X, V)) (x : SecOn (tw M H t) V) (p : M.ι) (k : H.ι) : twCoord (a • x) p k = rs X inf_le_left a * twCoord x p k

@[simp] theorem **twCoord_restrict** (h : W ≤ V) (x : SecOn (tw M H t) V) (p : M.ι) (k : H.ι) : twCoord (restrict (tw M H t) h x) p k = rs X (inf_le_inf_right _ h) (twCoord x p k)

theorem **res_tw_g** (t : ℤ) (p q : M.ι) (k l : H.ι) {O : X.Opens} (h : O ≤ M.U p ⊓ H.U k ⊓ (M.U q ⊓ H.U l)) (hM : O ≤ M.U p ⊓ M.U q) (hH : O ≤ H.U k ⊓ H.U l) : res h ((tw M H t).g (p, k) (q, l)) = res hM (M.g p q) * res hH (H.g k l) ^ t

theorem **twCoord_compat** (x : SecOn (tw M H t) V) (p q : M.ι) (k l : H.ι) {O : X.Opens} (hp : O ≤ V ⊓ (M.U p ⊓ H.U k)) (hq : O ≤ V ⊓ (M.U q ⊓ H.U l)) (hM : O ≤ M.U p ⊓ M.U q) (hH : O ≤ H.U k ⊓ H.U l) : rs X hp (twCoord x p k) = (res hM (M.g p q) * res hH (H.g k l) ^ t).val * rs X hq (twCoord x q l)

@[simp] theorem **twCoord_twMk** (y : ∀ (p : M.ι) (k : H.ι), Γ(X, V ⊓ (M.U p ⊓ H.U k))) (hy) (p : M.ι) (k : H.ι) : twCoord (twMk t y hy) p k = y p k

@[simp] theorem **twCoord_mulSec** (s : LB.Sec H) (x : SecOn (tw M H t) V) (p : M.ι) (k : H.ι) : twCoord (mulSec M t s V x) p k = rs X (inf_le_right.trans inf_le_right) (s.f k) * twCoord x p k

theorem **mulSec_restrict** (s : LB.Sec H) (h : W ≤ V) (x : SecOn (tw M H t) V) : mulSec M t s W (restrict _ h x) = restrict _ h (mulSec M t s V x)

theorem **mulSec_comm** (s s' : LB.Sec H) (x : SecOn (tw M H t) V) : mulSec M (t + 1) s V (mulSec M t s' V x) = mulSec M (t + 1) s' V (mulSec M t s V x)

theorem **mulSec_eq_smul** (s s' : LB.Sec H) (r : Γ(X, V)) (h : ∀ k, rs X (inf_le_right : V ⊓ H.U k ≤ H.U k) (s'.f k) = rs X inf_le_left r ∗ rs X (inf_le_right : V ⊓ H.U k ≤ H.U k) (s.f k)) (x : SecOn (tw M H t) V) : mulSec M t s' V x = r • mulSec M t s V x

theorem **isUnit_rs_of_le_nonzeroLocus** (s : LB.Sec H) (k : H.ι) {O : X.Opens} (hk : O ≤ H.U k) (h : O ≤ nonzeroLocus s) : IsUnit (rs X hk (s.f k))

theorem **mulSec_bijective** (s : LB.Sec H) (h : V ≤ nonzeroLocus s) : Function.Bijective (mulSec M t s V)

@[simp] theorem **mulSecEquiv_apply** (s : LB.Sec H) (h : V ≤ nonzeroLocus s) (x : SecOn (tw M H t) V) : mulSecEquiv M t s h x = mulSec M t s V x

theorem **twZeroIso_isGen** (hM : M.IsLB) : (twZeroIso M H).IsGen

theorem **twCoord_twZeroEquiv** (hM : M.IsLB) (hH : H.IsLB) (s : SecOn M V) (p : M.ι) (k : H.ι) : twCoord (twZeroEquiv hM hH V s) p k = rs X (inf_le_inf_left V inf_le_left) (coord s p)

theorem **twZeroEquiv_restrict** (hM : M.IsLB) (hH : H.IsLB) (h : W ≤ V) (s : SecOn M V) : twZeroEquiv hM hH W (restrict M h s) = restrict _ h (twZeroEquiv hM hH V s)

theorem **appLE_cst** {f : X ⟶ Spec (CommRingCat.of R)} {f' : X' ⟶ Spec (CommRingCat.of R')} {π : X' ⟶ X} {g : R →+* R'} (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom g)) {V : X.Opens} {V' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (r : R) : (π.appLE V V' e).hom (cst f V r) = cst f' V' (g r)

theorem **smul_def** (r : R) (s : SecOnR f M V) : r • s = cst f V r • s

theorem **isScalarTower_of_algebraMap_eq** [Algebra R Γ(X, V)] (h : algebraMap R Γ(X, V) = cst f V) : IsScalarTower R Γ(X, V) (SecOnR f M V)

@[simp] theorem **coord_of** (s : SecOn M V) (i : M.ι) : coord (of f M V s) i = GlobalTheory.LB.coord s i

@[**ext**] theorem ext {s t : SecOnR f M V} (h : ∀ i, coord s i = coord t i) : s = t

theorem **compat** (s : SecOnR f M V) (i j : M.ι) {O : X.Opens} (hi : O ≤ V ⊓ M.U i) (hj : O ≤ V ⊓ M.U j) (hij : O ≤ M.U i ⊓ M.U j) : rs X hi (coord s i) = rs X hij (M.g i j).val ∗ rs X hj (coord s j)

@[simp] theorem **coord_zero** (i : M.ι) : coord (0 : SecOnR f M V) i = 0

@[simp] theorem **coord_add** (s t : SecOnR f M V) (i : M.ι) : coord (s + t) i = coord s i + coord t i

@[simp] theorem **coord_neg** (s : SecOnR f M V) (i : M.ι) : coord (-s) i = -coord s i

@[simp] theorem **coord_sub** (s t : SecOnR f M V) (i : M.ι) : coord (s - t) i = coord s i - coord t i

@[simp] theorem **coord_smul** (a : Γ(X, V)) (s : SecOnR f M V) (i : M.ι) : coord (a • s) i = rs X inf_le_left a ∗ coord s i

@[simp] theorem **coord_base_smul** (r : R) (s : SecOnR f M V) (i : M.ι) : coord (r • s) i = cst f _ r ∗ coord s i

@[simp] theorem **restrict_of** (h : W ≤ V) (s : SecOn M V) : restrict f M h (of f M V s) = of f M W (SecOn.restrict M h s)

@[simp] theorem **coord_restrict** (h : W ≤ V) (s : SecOnR f M V) (i : M.ι) : coord (restrict f M h s) i = rs X (inf_le_inf_right _ h) (coord s i)

theorem **restrict_smul** (h : W ≤ V) (a : Γ(X, V)) (s : SecOnR f M V) : restrict f M h (a • s) = rs X h a • restrict f M h s

@[simp] theorem **restrict_self** (h : V ≤ V) (s : SecOnR f M V) : restrict f M h s = s

@[simp] theorem **restrict_restrict** {O : X.Opens} (h₁ : W ≤ V) (h₂ : O ≤ W) (s : SecOnR f M V) : restrict f M h₂ (restrict f M h₁ s) = restrict f M (h₂.trans h₁) s

theorem **restrict_bijective** (h : W ≤ V) (h' : V ≤ W) : Function.Bijective (restrict f M h)

theorem **ext_of_cover** {I : Type} (O : I → X.Opens) (hO : ∀ a, O a ≤ V) (hc : V ≤ ⨆ a, O a) {s t : SecOnR f M V} (h : ∀ a, restrict f M (hO a) s = restrict f M (hO a) t) : s = t

theorem **existsUnique_glue** {I : Type} (O : I → X.Opens) (hO : ∀ a, O a ≤ V) (hc : V ≤ ⨆ a, O a) (s : ∀ a, SecOnR f M (O a)) (h : ∀ a b, restrict f M (inf_le_left : O a ⊓ O b ≤ O a) (s a) = restrict f M (inf_le_right : O a ⊓ O b ≤ O b) (s b)) : ∃! t : SecOnR f M V, ∀ a, restrict f M (hO a) t = s a

@[simp] theorem **lin_of** (φ : SecOn L V →ₗ[Γ(X, V)] SecOn M V) (s : SecOn L V) : lin f φ (of f L V s) = of f M V (φ s)

@[simp] theorem **linEquiv_of** (φ : SecOn L V ≃ₗ[Γ(X, V)] SecOn M V) (s : SecOn L V) : linEquiv f φ (of f L V s) = of f M V (φ s)

theorem **topEquiv_f** (s : SecOnR f M ⊤) (i : M.ι) : (topEquiv f M s).f i = rs X (le_inf le_top le_rfl) (coord s i)

theorem **chart_apply** (hM : M.IsLB) (i : M.ι) (h : V ≤ M.U i) (s : SecOnR f M V) : chart f hM i h s = rs X (le_inf le_rfl h) (coord s i)

theorem **mulSec_bijective** (t : ℤ) (s : LB.Sec H) (h : V ≤ nonzeroLocus s) : Function.Bijective (mulSec f M t s V)

@[simp] theorem **pull_of** {V' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (s : SecOn M V) : pull M w V V' e (of f M V s) = of f' _ V' (SecOn.pullLE M π V V' e s)

theorem **pull_smul** {V' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (a : Γ(X, V)) (s : SecOnR f M V) : pull M w V V' e (a • s) = (π.appLE V V' e).hom a • pull M w V V' e s

theorem **pull_restrict** {V' W' : X'.Opens} (h : W ≤ V) (h' : W' ≤ V') (e : V' ≤ π ⁻¹ᵁ V) (e' : W' ≤ π ⁻¹ᵁ W) (s : SecOnR f M V) : pull M w W W' e' (restrict f M h s) = restrict f' _ h' (pull M w V V' e s)

theorem **iInf_le_iInf_of_subset** {s t : Finset κ} (h : s ⊆ t) : ⨅ i ∈ t, D i ≤ ⨅ i ∈ s, D i

theorem **secCube_obj** (s : Finset κ) : (secCube f M D).obj s = ModuleCat.of R (SecOnR f M (⨅ i ∈ s, D i))

theorem **res_secCube** {s t : Finset κ} (h : s ⊆ t) : GlobalTheory.Cube.res (secCube f M D) h = SecOnR.restrict f M (iInf_le_iInf_of_subset D h)

theorem **secCube_empty_bijective** : Function.Bijective fun s : LB.Sec M => (SecOnR.of f M _ (GlobalTheory.LB.ofSec s _) : (secCube f M D).obj ∅)

theorem **cechLB_X** [LinearOrder κ] (p : ℕ) : (cechLB f M D).X p = ModuleCat.of R (∀ s : {s : Finset κ // s.card = p + 1}, SecOnR f M (⨅ i ∈ s.1, D i))

theorem **cechLB_d** [LinearOrder κ] (p : ℕ) : (cechLB f M D).d p (p + 1) = ModuleCat.ofHom (GlobalTheory.Cube.d (secCube f M D) (p + 1))

theorem **isZero_cechLB_X** [LinearOrder κ] [Fintype κ] {p : ℕ} (h : Fintype.card κ ≤ p) : IsZero ((cechLB f M D).X p)

theorem **secCubeMul_app** {H : LB X} (t : ℤ) (s : LB.Sec H) (I : Finset κ) : ((secCubeMul f M D t s).app I).hom = SecOnR.mulSec f M t s (⊓ i ∈ I, D i)

theorem **preimage_biInf** (π : X' ⟶ X) (s : Finset κ) : π ⁻¹ᵁ (⊓ i ∈ s, D i) = ⊓ i ∈ s, π ⁻¹ᵁ D i

theorem **secCubePull_res** {s t : Finset κ} (h : s ⊆ t) (x : (secCube f M D).obj s) : secCubePull M D w t (GlobalTheory.Cube.res (secCube f M D) h x) = GlobalTheory.Cube.res (secCube f' (M.pull π) fun i => π ⁻¹ᵁ D i) h (secCubePull M D w s x)

theorem **coord_secOnRTrivEquiv** (f : X ⟶ Spec (CommRingCat.of R)) [IsIso f] {V : X.Opens} (hV : V = ⊤) (r : R) (i : (Inh.trivLB X).ι) : SecOnR.coord (secOnRTrivEquiv f hV r) i = cst f _ r

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.NotExistsOfSubsingleton

theorem **triv_noChart** : Triv (noChart X)

theorem **bijective_app** {F G : X.Modules} (e : F ≅ G) (V : X.Opens) : Function.Bijective (e.hom.app V)

theorem **not_exists_of_subsingleton** [Nonempty X] (F : X.Modules) (hF : ∀ V, Subsingleton Γ(F, V)) : ¬ ∃ M : LB X, M.IsLB ∧ Nonempty (lbSheaf M ≅ F)

theorem **triv_and_not_iso** [Nonempty X] : Triv (noChart X) ∧ ¬ Nonempty (lbSheaf (noChart X) ≅ unitMod X)

theorem **noChart_empty** [IsEmpty X] : (noChart X).IsLB ∧ Nonempty (lbSheaf (noChart X) ≅ unitMod X)

theorem **coord_secEquivHom** {M M' : LB X} (hM : M.IsLB) (hM' : M'.IsLB) (w : LB.Sec (LB.div M' M)) {V O : X.Opens} (s : SecOn M V) (p : M'.ι) (q : M.ι) (hp : O ≤ V ⊓ M'.U p) (hq : O ≤ V ⊓ M.U q) (h : O ≤ M'.U p ⊓ M.U q) : rs X hp (coord ((lbSheaf.secEquivHom hM hM' w).app V s) p) = rs X h (w.f (p, q)) * rs X hq (coord s q)

theorem **isoOfGen_hom** {M M' : LB X} (hM : M.IsLB) (hM' : M'.IsLB) (w : LB.Gen (LB.div M' M)) (hw : w.IsGen) : (lbSheaf.isoOfGen hM hM' w hw).hom = lbSheaf.secEquivHom hM hM' (DVR.genSec w hw)

theorem **hom_ext_one** {F : X.Modules} {φ ψ : unitMod X ⟶ F} (h : φ.app ⊤ (1 : Γ(X, ⊤)) = ψ.app ⊤ (1 : Γ(X, ⊤))) : φ = ψ

theorem **subsingleton_div_negOne** : Subsingleton (LB.Sec (LB.div (SheafModel.negOne R) (Inh.trivLB (PSpace R 1))))

theorem **nontrivial_div_one** : Nontrivial (LB.Sec (LB.div (GlobalTheory.ProjSpace.O1 R 1) (Inh.trivLB (PSpace R 1))))

theorem **not_triv_div** : ¬ Triv (LB.div (SheafModel.negOne R) (GlobalTheory.ProjSpace.O1 R 1))

theorem **not_triv_pull_id** : ¬ Triv ((GlobalTheory.ProjSpace.O1 R 1).pull (𝟙 _))

theorem **triv_pull_chart** {M : LB X} (hM : M.IsLB) (i : M.ι) : Triv (M.pull (M.U i).ι)

theorem **pullback_unit** (π : X' ⟶ X) : Nonempty ((Scheme.Modules.pullback π).obj (unitMod X) ≅ unitMod X')

theorem **triv_div_pull_comp** {X'' : Scheme.{0}} {M : LB X} (hM : M.IsLB) (π : X' ⟶ X) (ρ : X'' ⟶ X') : Triv (LB.div ((M.pull π).pull ρ) (M.pull (ρ ≫ π)))

theorem **pullback_local** {M : LB X} (hM : M.IsLB) (R : Type) [CommRing R] [IsLocalRing R] (π : Spec (CommRingCat.of R) ⟶ X) : Nonempty ((Scheme.Modules.pullback π).obj (lbSheaf M) ≅ unitMod _)

theorem **unitGen_isGen** (u : Γ(X, ⊤)ˣ) : (unitGen u).IsGen

theorem **isoOfGen_unitGen** (u : Γ(X, ⊤)ˣ) (s : SecOn (Inh.trivLB X) ⊤) : (lbSheaf.isoOfGen (Inh.trivLB_isLB X) (Inh.trivLB_isLB X) (unitGen u) (unitGen_isGen u)).hom.app ⊤ s = u.val • s

Development.AbelianVarietyTheory.Definitions.AbelianSchemeLaws.SpecMapAwayFnEquivAwayChart

theorem **coeff_eq_zero_of_simple_pole** {R B C κ : Type*} [CommRing R] [CommRing B] [CommRing C] (ρ : R →+* B) (ι : B →+* C) (hι : Function.Injective ι) (ε : B →+* R) (hρ : ∀ r, ε (ρ r) = r) (z : B) (hz : ε z = 0) (d : κ → ℕ) (hd : Function.Injective d) (e : κ → C) (v : κ → B) (hv : ∀ k, ι z ^ d k * e k = ι (v k)) (hε : ∀ k, ε (v k) = 1) (c : κ → R) (s : Finset κ) (b : B) (h : ι z * ∑ k ∈ s, ι (ρ (c k)) * e k = ι b) : ∀ k ∈ s, 2 ≤ d k → c k = 0

theorem **constOn_apply** (U : (scheme W).Opens) (r : R) : constOn W U r = constFn (toSpec W) U r

theorem **resG_constOn** (h : V ≤ U) (r : R) : resG _ h (constOn W U r) = constOn W V r

theorem **resG_ratioOn** (hi : U ≤ coordOpen W i) (h : V ≤ U) : resG _ h (ratioOn W U i hi j) = ratioOn W V i (h.trans hi) j

@[simp] theorem **ratioOn_self** (hi : U ≤ coordOpen W i) : ratioOn W U i hi i = 1

theorem **ratioOn_mul_ratioOn** (hi : U ≤ coordOpen W i) (hj : U ≤ coordOpen W j) : ratioOn W U i hi j * ratioOn W U j hj k = ratioOn W U i hi k

theorem **isUnit_ratioOn** (hi : U ≤ coordOpen W i) (hj : U ≤ coordOpen W j) : IsUnit (ratioOn W U i hi j)

theorem **basicOpen_ratioOn** (hi : U ≤ coordOpen W i) : (scheme W).basicOpen (ratioOn W U i hi j) = U ⊓ coordOpen W j

@[simp] theorem **awayFnEquiv_awayCoord** (i j : Fin 3) : awayFnEquiv W i (awayCoord W i j) = ratio W i j

@[simp] theorem **awayFnEquiv_awayConst** (i : Fin 3) (r : R) : awayFnEquiv W i (awayConst W i r) = constOn W (coordOpen W i) r

theorem **equation_ratioOn** (hi : U ≤ coordOpen W i) : (W.map (constOn W U)).toProjective.Equation (ratioOn W U i hi)

theorem **eval₂Hom_ratio_surjective** (i : Fin 3) : Function.Surjective (MvPolynomial.eval₂Hom (constOn W (coordOpen W i)) (ratio W i))

theorem **SpecMap_awayFnEquiv_awayChart** (i : Fin 3) : spMap (awayFnEquiv W i : AwayRing W i →+* Γ(scheme W, coordOpen W i)) ≫ awayChart W i = (isAffineOpen_coordOpen W i).fromSpec

theorem **fromSpec_coordOpen_eq_ptHom** (i : Fin 3) : (isAffineOpen_coordOpen W i).fromSpec = ptHom W (constOn W (coordOpen W i)) (ratio W i) (.of_isUnit (equation_ratioOn le_rfl) i (by simp))

theorem **evalPt_constOn** {x : Sp A ⟶ scheme W} {ρ : R →+* A} (hx : x ≫ toSpec W = spMap ρ) {U : (scheme W).Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (r : R) : evalPt x U h (constOn W U r) = ρ r

theorem **eq_ptHom_evalPt** {x : Sp A ⟶ scheme W} {ρ : R →+* A} (hx : x ≫ toSpec W = spMap ρ) {U : (scheme W).Opens} {i : Fin 3} (hi : U ≤ coordOpen W i) (h : ⊤ ≤ x ⁻¹ᵁ U) {P : Fin 3 → A} (hP : ∀ j, evalPt x U h (ratioOn W U i hi j) = P j) : ∃ hP, x = ptHom W ρ P hP

theorem **evalPt_ptHom_ratioOn_mul** (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) {U : (scheme W).Opens} {i : Fin 3} (hi : U ≤ coordOpen W i) (h : ⊤ ≤ ptHom W ρ P hP ⁻¹ᵁ U) (j : Fin 3) : evalPt (ptHom W ρ P hP) U h (ratioOn W U i hi j) * P i = P j

@[simp] theorem **affineFnEquiv_affineX** : affineFnEquiv W (affineX W) = ratio W 2 0

@[simp] theorem **affineFnEquiv_affineY** : affineFnEquiv W (affineY W) = ratio W 2 1

@[simp] theorem **affineFnEquiv_algebraMap** (r : R) : affineFnEquiv W (algebraMap R _ r) = constOn W _ r

@[simp] theorem **originFnEquiv_originZ** : originFnEquiv W (originZ W) = −ratio W 1 0

@[simp] theorem **originFnEquiv_originW** : originFnEquiv W (originW W) = −ratio W 1 2

@[simp] theorem **originFnEquiv_algebraMap** (r : R) : originFnEquiv W (algebraMap R _ r) = constOn W _ r

theorem **originW_mem_nonZeroDivisors** : originW W ∈ nonZeroDivisors (OriginRing W)

theorem **isRegularFn_ratio_one_two** : Cartier.IsRegularFn (ratio W 1 2)

theorem **mapHom_app_ratioOn** {U : (scheme W).Opens} {i : Fin 3} (hi : U ≤ coordOpen W i) (j : Fin 3) : ((mapHom W f h).app U).hom (ratioOn W U i hi j) = ratioOn W' (mapHom W f h ⁻¹ᵁ U) i (((mapHom W f h).preimage_mono hi).trans (mapHom_preimage_coordOpen W f h i).le) j

theorem **mapHom_app_constOn** (U : (scheme W).Opens) (r : R) : ((mapHom W f h).app U).hom (constOn W U r) = constOn W' (mapHom W f h ⁻¹ᵁ U) (f r)

Development.AbelianVarietyTheory.Definitions.AdelicHermitianForms.IsHermHyperbolic

theorem **isHerm_hyperbolic** : IsHerm (starRingEnd ℂ) !![0, 1; 1, 0] ∧ ∃ z : Fin 2 → ℂ, z ≠ 0 ∧ hf (starRingEnd ℂ) !![0, 1; 1, 0] z z = 0

Development.AbelianVarietyTheory.Definitions.AdelicHermitianForms.MemPowSub

theorem **nrm_apply** {A : Type} [CommGroup A] (σ : A →* A) (n : ℕ) (a : A) : nrm σ n a = ∏ i ∈ Finset.range n, σ^[i] a

theorem **dif_apply** {A : Type} [CommGroup A] (σ : A →* A) (a : A) : dif σ a = σ a / a

theorem **_root_.GlobalTheory.CNorm.mem_fix** {A : Type} [CommGroup A] (σ : A →* A) (a : A) : a ∈ _root_.GlobalTheory.CNorm.fix σ ↔ σ a = a

theorem **_root_.GlobalTheory.CNorm.mem_powSub** (p : ℕ) {F L : Type} [Field F] [Field L] (φ : F →+* L) (x : Fˣ) : x ∈ _root_.GlobalTheory.CNorm.powSub p φ ↔ ∃ r : L, r ^ p = φ (x : F)

theorem **_root_.GlobalTheory.CNorm.mem_kNorm** (p : ℕ) {F L : Type} [Field F] [Field L] (φ : F →+* L) (δ : F) (x : Fˣ) : x ∈ _root_.GlobalTheory.CNorm.kNorm p φ δ ↔ ∃ y : (AdjoinRoot (X ^ p − C (φ δ) : L[X]))ˣ, Algebra.norm L (y : AdjoinRoot (X ^ p − C (φ δ) : L[X])) = φ (x : F)

Development.AbelianVarietyTheory.Definitions.AdelicHermitianForms.NontrivialTransvection

theorem **tv_mulVec** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (x v : Fin n → R) (t : R) : (tv σ G x t).mulVec v = v + (t * hf σ G v x) • x

theorem **hf_add_smul_left** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (v x y : Fin n → R) (c : R) : hf σ G (v + c • x) y = hf σ G v y + c * hf σ G x y

theorem **hf_add_smul_right** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (v x y : Fin n → R) (c : R) : hf σ G y (v + c • x) = hf σ G y v + σ c * hf σ G y x

theorem **hf_conj** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (hσ : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) (x y : Fin n → R) : σ (hf σ G x y) = hf σ G y x

theorem **isU_tv** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (hσ : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) (x : Fin n → R) (t : R) (h : t + σ t + t * σ t * hf σ G x x = 0) : IsU σ G (tv σ G x t)

theorem **isU_of_mem_tvGen** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (hσ : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) {m} (hm : m ∈ TvGen σ G) : IsU σ G m

theorem **det_tv** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (x : Fin n → R) (t : R) : (tv σ G x t).det = 1 + t * hf σ G x x

theorem **det_of_mem_tvGen** (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) {m} (hm : m ∈ TvGen σ G) : m.det = 1

theorem **nontrivial_transvection** : !![1, Complex.I; 0, 1] ∈ TvGen (starRingEnd ℂ) !![0, 1; 1, 0]

theorem **not_mem_tvGen** : IsU (starRingEnd ℂ) !![0, 1; 1, 0] !![Complex.I, 0; 0, Complex.I] ∧ !![Complex.I, 0; 0, Complex.I] ∉ TvGen (starRingEnd ℂ) !![0, 1; 1, 0]

theorem **not_isU** : ¬ IsU (starRingEnd ℂ) !![0, 1; 1, 0] !![1, 1; 0, 1]

Development.AbelianVarietyTheory.Definitions.AdelicHermitianForms.SqSubLeNormSub

theorem **mem_sqSub** {F L : Type} [Field F] [Field L] (φ : F →+* L) (x : Fˣ) : x ∈ sqSub φ ↔ IsSquare (φ (x : F))

theorem **mem_normSub** {F L : Type} [Field F] [Field L] (φ : F →+* L) (δ : F) (x : Fˣ) : x ∈ normSub φ δ ↔ ∃ a b : L, φ (x : F) = a ^ 2 − φ δ * b ^ 2

theorem **sqSub_le_normSub** {F L : Type} [Field F] [Field L] (φ : F →+* L) (δ : F) : sqSub φ ≤ normSub φ δ

theorem **mem_sUnits** (S : Finset (HeightOneSpectrum (𝒪 F))) (x : Fˣ) : x ∈ sUnits F S ↔ ∀ v : HeightOneSpectrum (𝒪 F), v ∉ S → v.valuation F (x : F) = 1

theorem **admitted.mono** {F : Type} [Field F] [NumberField F] {S T : Finset (HeightOneSpectrum (𝒪 F))} (h : admitted F S) (hST : S ⊆ T) : admitted F T

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.AnalyticTorusPointBaseEqHolUnifEx

theorem **isTangent_zero** (v : epsR ℂ ⟶ (zeroPolAV (Sp ℂ)).A) : IsTangent v

theorem **isDiff_zero** {O : Type} [CommRing O] [StarRing O] (A : AnArc O) [IsEmpty (Fin A.gB)] (w : Fin A.gB → ℂ) (v : epsR ℂ ⟶ (zeroPolAV (Sp ℂ)).A) : (zeroUnif A).IsDiff w v

theorem **isScalarTower** : IsScalarTower ℂ (FractionRing ℂ) ℂ

theorem **ex_zero** : Λ.u.ex 0 = η[a.D₀.P.A]

theorem **isLift_fam** (z : Fin Λ.A.gB → ℂ) (α : Λ.A.X) : IsLift Λ.u (a.D₀.famφB (Λ.ε α)) (Λ.u.ex z).left ![z, z + Λ.A.ct α, Λ.A.ct α, 0]

theorem **fn_mem_Theta** (j : ℕ) (α : Λ.A.X) (s : a.D₀.SecB j (Λ.ε α)) : Λ.fn j α s ∈ Λ.A.Theta j α

theorem **analyticTorusPoint_base_eq_holUnif_ex** (v : Λ.A.V) : (Λ.ex0 v).b = toUnit _ ≫ Λ.u.ex v.2

theorem **isLift_gt** (v : Λ.A.V) (x : Λ.A.X) : IsLift Λ.u (Out.DegPre.gtφ a.D₀ ℂ (Λ.ex0 v).b (Λ.ε x)) (show Sp ℂ ⟶ _ from 𝟙 (Sp ℂ)) ![Λ.A.ct x + v.2, Λ.A.ct x, v.2, 0]

theorem **corr_of_genVal** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} {u : HolUnif A L g} (hL : L.IsLB) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Sp ℂ ⟶ A.left)} {ε : K → ℤ} {φ' : K' → (Sp ℂ ⟶ A.left)} {ε' : K' → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' :

LB.Gen (LB.LinComb L φ' ε')} (ht : t.IsGen) (ht' : t'.IsGen) {z : K → Fin g → ℂ} {z' : K' → Fin g → ℂ} (hz : IsLift u φ (𝟙 _) z) (hz' : IsLift u φ' (𝟙 _) z') (hs : ∀ f : (Fin g → ℂ) → ℂˣ, ∏ k, f (z k) ^ ε k = ∏ k, f (z' k) ^ ε' k) (h : genVal u hL t (𝟙 _) z = genVal u hL t' (𝟙 _) z') : LB.Gen.Corr L φ ε φ' ε' t t'

theorem **genVal_mul_of** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} {u : HolUnif A L g} {hL : L.IsLB} {Y : Scheme.{0}} {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y ⟶ A.left)} {ε : K → ℤ} {φ' : K' → (Y ⟶ A.left)} {ε' : K' → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' : LB.Gen (LB.LinComb L φ' ε')} {y : Sp ℂ ⟶ Y} {z : K → Fin g → ℂ} {z' : K' → Fin g → ℂ} {p q : ℂ} (h : genVal u hL t y z = p) (h' : genVal u hL t' y z' = q) : genVal u hL (LB.mulGen L φ ε φ' ε' t t') y (Sum.elim z z') = p * q

theorem **genVal_inv_of** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} {u : HolUnif A L g} {hL : L.IsLB} {Y : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ A.left)} {ε : K → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {y : Sp ℂ ⟶ Y} {z : K → Fin g → ℂ} {p : ℂ} (h : genVal u hL t y z = p) : genVal u hL (LB.invGen L φ ε t) y z = p⁻¹

theorem **genVal_pull_of** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} {u : HolUnif A L g} {hL : L.IsLB} {Y Y' : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ A.left)} {ε : K → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {f : Y' ⟶ Y} {y : Sp ℂ ⟶ Y'} {y' : Sp ℂ ⟶ Y} (e : y ≫ f = y') {z : K → Fin g → ℂ} {p : ℂ} (h : genVal u hL t y' z = p) : genVal u hL (LB.pullGen L φ ε t f) y z = p

theorem **genVal_shift** (v : Λ.A.V) (z : Fin Λ.A.gB → ℂ) (m : Λ.A.abelian₂) (hv : v.2 = z + m) (x : Λ.A.X) : IsLift Λ.u (Out.DegPre.gtφ a.D₀ ℂ (Λ.ex0 v).b (Λ.ε x)) (show Sp ℂ ⟶ _ from 𝟙 (Sp ℂ)) ![Λ.A.ct x + z, Λ.A.ct x, z, 0] ∧ genVal Λ.u a.D₀.P.isLB ((Λ.ex0 v).s (Λ.ε x)) (show Sp ℂ ⟶ _ from 𝟙 (Sp ℂ)) ![Λ.A.ct x + z, Λ.A.ct x, z, 0] = eTwoPi (v.1 x + Λ.A.H (Λ.A.ct x) m / (2 * Complex.I))

theorem **isZero_iff** (v : Λ.A.V) : (Λ.ex0 v).IsZero ↔ ∃ la : Λ.A.La, la.y = 0 ∧ v = Λ.A.P0 la

theorem **isIota** (la : Λ.A.La) : Out.GtPtF.IsIota (Λ.ε la.y) (Λ.ex0 (Λ.A.P0 la))

theorem **isAdd** (v v' : Λ.A.V) : (Λ.ex0 v).IsAdd (Λ.ex0 v') (Λ.ex0 (v + v'))

theorem **analyticTorusPoint_surjective** : Function.Surjective Λ.ex0

theorem **rep_iff** (k : ℕ) (p : Out.QkPt a.D₀ (FractionRing ℂ) k) (la : Λ.A.La) : Λ.Rep k p la ↔ ∃ q : Out.QkPt a.D₀ ℂ k, QkExt (algebraMap _ ℂ) p q ∧ q.y = Λ.ε la.y ∧ q.g = Λ.ex0 ((k : ℂ)⁻¹ • Λ.A.P0 la)

theorem **gtAct** (Λ : letI

theorem **exists_rep** (Λ : letI

theorem **rep_unique** (Λ : letI

theorem **rep_eta** (v : HypLattice H) : letI

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.ArcVarZpowInjective

private lemma **exists_pos_nat_mul_eq_int** {Idx : Type} [Fintype Idx] (q : Idx → ℚ) : ∃ N : ℕ, 0 < N ∧ ∃ w : Idx → ℤ, ∀ i, (w i : ℚ) = N * q i

theorem **_root_.GlobalTheory.ToricData.finite_form_le** (be : LinearMap.BilinForm ℤ X) (hs : ∀ y z, be y z = be z y) (hp : ∀ y, y ≠ 0 → 0 < be y y) (C : ℤ) : {y | be y y ≤ C}.Finite

theorem **arcVar_val** : (arcVar k : FractionRing (PowerSeries k)) = algebraMap (PowerSeries k) _ PowerSeries.X

theorem **arcVar_zpow_injective** : Function.Injective fun n : ℤ => arcVar k ^ n

theorem **arcExp_eq** (y : X) : arcExp k be y = arcVar k ^ (be y y / 2)

theorem **arcExp_val** (y : X) : (arcExp k be y : FractionRing (PowerSeries k)) = algebraMap (PowerSeries k) (FractionRing (PowerSeries k)) PowerSeries.X ^ (be y y / 2)

theorem **arcExp_neg** (y : X) : arcExp k be (-y) = arcExp k be y

theorem **polar_arcExp** (hs : ∀ y z, be y z = be z y) (he : ∀ y, (2 : ℤ) ∣ be y y) (y z : X) : polar (arcExp k be) y z = arcVar k ^ (be y z)

theorem **psiIn_arcExp_iff** (y : X) (n : ℕ) : (toricPre _ X (arcExp k be)).PsiIn y (Ideal.span {(PowerSeries.X : PowerSeries k)} ^ n) ↔ (n : ℤ) ≤ be y y / 2

theorem **arcData_tau_isConst** (y z : X) : IsConst ((arcData k be hs hp he).toPre.tau y z) (unitOn (arcVar k ^ (be y z)))

theorem **left_eq_hom** {S : Scheme.{0}} {T : Over S} (x : T ⟶ (zeroPolAV S).A) : x.left = T.hom

theorem **left_baseMap** (f : K →+* K') (q' : X → (FractionRing K')ˣ) {T : Over (Sp K')} (x : T ⟶ (toricPre K' X q').P.A) : x.left ≫ baseMap q f q' = T.hom ≫ spMap f

theorem **baseMap_pb** {K' : Type} [CommRing K'] [IsDomain K'] (f : K →+* K') (q' : X → (FractionRing K')ˣ) : IsGrpPullback (toricPre K' X q').P.A (toricPre K X q).P.A (spMap f) (baseMap q f q')

theorem **eSec_baseMap** : eSec (toricPre K' X q').P.A ≫ baseMap q f q' = spMap f ≫ eSec (toricPre K X q).P.A

theorem **end_baseMap** (g' : (toricPre K' X q').P.A ⟶ (toricPre K' X q').P.A) (g : (toricPre K X q).P.A ⟶ (toricPre K X q).P.A) : g'.left ≫ baseMap q f q' = baseMap q f q' ≫ g.left

theorem **c_baseMap** (y' y : X) : ((toricPre K' X q').c y').left ≫ baseMap q f q' = spMap f ≫ ((toricPre K X q).c y).left

theorem **baseIso_isGen** : (baseIso q f q').IsGen

theorem **baseIso_rig** (he : eSec (toricPre K' X q').P.A ≫ baseMap q f q' = spMap f ≫ eSec (toricPre K X q).P.A) : LB.Gen.Maps (LB.linCombIso (baseIso q f q') (K

theorem **valF_gv_unitOn** {F : Type} [Field F] {V : (Sp F).Opens} (hV : V = ⊤) (a : Fˣ) : valF (𝟙 (Sp F)) V (gv (unitOn a) V).val = a

theorem **valF_one** {F : Type} [Field F] {Y : Scheme.{0}} (x : Sp F ⟶ Y) {U : Y.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) : valF x U 1 = 1

theorem **genRel_baseIso_iff** {F : Type} [Field F] {J : Type} [Fintype J] {pt : J → (Sp F ⟶ (toricPre K' X q').P.A.left)} {pt₀ : J → (Sp F ⟶ (toricPre K X q).P.A.left)} {ε : J → ℤ} {t : LB.Gen (LB.LinComb (toricPre K' X q').P.L pt ε)} {t₀ : LB.Gen (LB.LinComb (toricPre K X q).P.L pt₀ ε)} {a a₀ : Fˣ} (ht : IsConst t (unitOn a)) (ht₀ : IsConst t₀ (unitOn a₀)) (x : F) : GenRel (baseIso q f q') t t₀ x ↔ (∀ j, pt j ≫ baseMap q f q' = pt₀ j) ∧ (a : F) = x * a₀

theorem **jFK_spMap** (hF : ∀ x, algebraMap (FractionRing K) F (algebraMap K _ x) = algebraMap (FractionRing K') F (algebraMap K' _ (f x))) : (TF (R

theorem **pointF_baseMap** (hF : ∀ x, algebraMap (FractionRing K) F (algebraMap K _ x) = algebraMap (FractionRing K') F (algebraMap K' _ (f x))) (b' : TF (R

theorem **psiF_isConst** (y : X) : IsConst (LB.pullGen _ _ _ ((toricPre K X q).psi y) (jFK (R

theorem **genRel_psi_iff** (hF : ∀ x, algebraMap (FractionRing K) F (algebraMap K _ x) = algebraMap (FractionRing K') F (algebraMap K' _ (f x))) (y' y : X) (x : F) : GenRel (baseIso q f q') (LB.pullGen _ _ _ ((toricPre K' X q').psi y') (jFK (R

theorem **genRel_pt_iff** (hF : ∀ x, algebraMap (FractionRing K) F (algebraMap K _ x) = algebraMap (FractionRing K') F (algebraMap K' _ (f x))) (t' t : Multiplicative X →* Fˣ) (χ' χ : X) (x : F) : GenRel (baseIso q f q') ((pt q' F t').s χ') ((pt q F t).s χ) x ↔ (t' (Multiplicative.ofAdd χ') : F) = x * t (Multiplicative.ofAdd χ)

theorem **genRel_arcData_psi** (k : Type) [Field k] (be : LinearMap.BilinForm ℤ X) (hs : ∀ y z, be y z = be z y) (hp : ∀ y, y ≠ 0 → 0 < be y y) (he : ∀ y, (2 : ℤ) ∣ be y y) (F : Type) [Field F] [Algebra (FractionRing (PowerSeries k)) F] [Algebra (FractionRing k) F] (u : F) (hF : ∀ x, algebraMap (FractionRing k) F (algebraMap k _ x) = algebraMap (FractionRing (PowerSeries k)) F (algebraMap (PowerSeries k) _ (PowerSeries.C x))) (hu : u = algebraMap (FractionRing (PowerSeries k)) F (algebraMap (PowerSeries k) _ PowerSeries.X)) (y' y : X) (h : be y' y' = be y y) : GenRel (baseIso (fun _ : X => (1 : (FractionRing k)ˣ)) (PowerSeries.C (R

theorem **twoMul_apply** (a b : ℤ) : twoMul a b = 2 * (a * b)

theorem **arcExp_twoMul** (k : Type) [Field k] (n : ℤ) : arcExp k twoMul n = arcVar k ^ (n ^ 2)

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.CardLeOfLinearIndependentTheta

theorem **translate_div_mem_Theta** {j : ℕ} [NeZero j] {α : A.X} {th : (Fin A.gB → ℂ) → ℂ} (h : th ∈ A.Theta j α) : (fun z => th (z - (j : ℂ)⁻¹ • A.ct α)) ∈ A.Theta j 0

theorem **card_le_of_linearIndependent_Theta** (j : ℕ) [NeZero j] (α : A.X) {N : ℕ} (f : Fin N → A.Theta j α) (hf : LinearIndependent ℂ f) : N ≤ j ^ A.gB

theorem **rank_Theta_le** (j : ℕ) [NeZero j] (α : A.X) : Module.rank ℂ (A.Theta j α) ≤ (j ^ A.gB : ℕ)

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.CoeffFSumDerivEqFderiv

theorem **perFactor_ne_zero** (w : ℂ) (la : A.La) (v : A.V) : A.perFactor w la v ≠ 0

theorem **_root_.GlobalTheory.AnArc.term_add_period** {j : ℕ} {α : A.X} {th : (Fin A.gB → ℂ) → ℂ} (h : th ∈ A.Theta j α) (w : ℂ) (la : A.La) (μ : A.X) (v : A.V) : A.term j α th μ (v + A.Pw w la) * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.eTwoPi (w * A.exponent j α μ) = A.perFactor w la v ^ j * (A.term j α th (μ + la.y) v * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.eTwoPi (w * A.exponent j α (μ + la.y)))

theorem **_root_.GlobalTheory.AnArc.Sum_add_period** {j : ℕ} {α : A.X} {th : (Fin A.gB → ℂ) → ℂ} (h : th ∈ A.Theta j α) (w : ℂ) (la : A.La) (v : A.V) : A.Sum j α th w (v + A.Pw w la) = A.perFactor w la v ^ j * A.Sum j α th w v

theorem **coeff_FSumShift** (k : ℕ) [NeZero k] (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (la : A.La) (z : A.V) (n : ℤ) : (A.FSumShift k α th la z).coeff n = ∑ μ ∈ A.level k (α + la.y) (n - A.be la.y α / k), A.term k α th μ (z + (k : ℂ)⁻¹ • A.P0 la)

theorem **IsRigFrame.comp** {L : RigLB G} {y : Sp ℂ ⟶ S} {e : V → (Sp ℂ ⟶ G.left)} {fr : L.L.ι → V → ℂ} (H : IsRigFrame L y e fr) (h : V' →L[ℂ] V) : IsRigFrame L y (fun z => e (h z)) (fun j z => fr j (h z))

theorem **IsAlgebraicIn.comp** {L : RigLB G} {y : Sp ℂ ⟶ S} {e : V → (Sp ℂ ⟶ G.left)} {θ : V → ℂ} {fr : L.L.ι → V → ℂ} (H : IsAlgebraicIn L y e θ fr) (h : V' → V) : IsAlgebraicIn L y (fun z => e (h z)) (fun z => θ (h z)) (fun j z => fr j (h z))

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.DegeneratingToriPeriodMapEqBaseOfYEqZero

theorem **e_add** (x y : ℂ) : eTwoPi (x + y) = eTwoPi x * eTwoPi y

@[simp] theorem **e_zero** : eTwoPi 0 = 1

theorem **e_ne_zero** (x : ℂ) : eTwoPi x ≠ 0

@[simp] theorem **e_intCast** (n : ℤ) : eTwoPi n = 1

theorem **e_nat_mul** (n : ℕ) (x : ℂ) : eTwoPi (n * x) = eTwoPi x ^ n

theorem **e_mul_intCast** (x : ℂ) (n : ℤ) : eTwoPi (x * n) = eTwoPi x ^ n

theorem **e_neg_div** (x : ℂ) : eTwoPi (-(x / (2 * Complex.I))) = Complex.exp (-(Real.pi * x))

theorem **H_def** (z w : Fin A.gB → ℂ) : A.H z w = hermForm A.Hm z w

theorem **H_add_left** (z z' w : Fin A.gB → ℂ) : A.H (z + z') w = A.H z w + A.H z' w

theorem **H_add_right** (z w w' : Fin A.gB → ℂ) : A.H z (w + w') = A.H z w + A.H z w'

theorem **H_smul_left** (c : ℂ) (z w : Fin A.gB → ℂ) : A.H (c • z) w = c * A.H z w

theorem **H_smul_right** (c : ℂ) (z w : Fin A.gB → ℂ) : A.H z (c • w) = starRingEnd ℂ c * A.H z w

@[simp] theorem **H_zero_left** (w : Fin A.gB → ℂ) : A.H 0 w = 0

@[simp] theorem **H_zero_right** (z : Fin A.gB → ℂ) : A.H z 0 = 0

theorem **H_neg_left** (z w : Fin A.gB → ℂ) : A.H (-z) w = -A.H z w

theorem **H_neg_right** (z w : Fin A.gB → ℂ) : A.H z (-w) = -A.H z w

theorem **H_conj** (z w : Fin A.gB → ℂ) : starRingEnd ℂ (A.H z w) = A.H w z

theorem **H_self_im** (z : Fin A.gB → ℂ) : (A.H z z).im = 0

theorem **H_pos** {z : Fin A.gB → ℂ} (hz : z ≠ 0) : 0 < (A.H z z).re

theorem **H_F** (a : O) (z w : Fin A.gB → ℂ) : A.H (A.F a z) w = A.H z (A.F (star a) w)

theorem **Ef_eq_im** (l m : A.abelian₂) : (A.Ef l m : ℝ) = (A.H l m).im

@[simp] theorem **Ef_self** (l : A.abelian₂) : A.Ef l l = 0

@[simp] theorem **chi_zero** : A.chi 0 = 1

@[simp] theorem **psi_zero** : A.psi 0 = 1

@[simp] theorem **coe_FLam** (a : O) (l : A.abelian₂) : (A.FLam a l : Fin A.gB → ℂ) = A.F a l

theorem **Ef_FLam** (a : O) (l m : A.abelian₂) : A.Ef (A.FLam a l) m = A.Ef l (A.FLam (star a) m)

theorem **coord_apply** (l : A.X →+ ℂ) (i) : A.coord l i = l (Module.Free.chooseBasis ℤ A.X i)

theorem **norm_eq_coord** (l : A.X →+ ℂ) : ‖l‖ = ‖A.coord l‖

theorem **finrank_hom** : Module.finrank ℂ (A.X →+ ℂ) = Module.finrank ℤ A.X

@[simp] theorem **evalCLM_apply** (x : A.X) (l : A.X →+ ℂ) : A.evalCLM x l = l x

theorem **finrank_V** : Module.finrank ℂ A.V = Module.finrank ℤ A.X + A.gB

@[simp] theorem **actV_fst_apply** (a : O) (v : A.V) (x : A.X) : (A.actV a v).1 x = v.1 (star a • x)

@[simp] theorem **actV_snd** (a : O) (v : A.V) : (A.actV a v).2 = A.F a v.2

@[simp] theorem **mk_y** (y : A.X) (n : A.X →+ ℤ) (l : A.abelian₂) : (mk y n l).y = y

@[simp] theorem **mk_n** (y : A.X) (n : A.X →+ ℤ) (l : A.abelian₂) : (mk y n l).n = n

@[simp] theorem **mk_l** (y : A.X) (n : A.X →+ ℤ) (l : A.abelian₂) : (mk y n l).l = l

@[simp] theorem **mk_eta** (v : A.La) : mk v.y v.n v.l = v

@[**ext**] theorem ext {v v' : A.La} (hy : v.y = v'.y) (hn : v.n = v'.n) (hl : v.l = v'.l) : v = v'

@[simp] theorem **zero_y** : (0 : A.La).y = 0

@[simp] theorem **zero_n** : (0 : A.La).n = 0

@[simp] theorem **zero_l** : (0 : A.La).l = 0

@[simp] theorem **add_y** (v v' : A.La) : (v + v').y = v.y + v'.y

@[simp] theorem **add_n** (v v' : A.La) : (v + v').n = v.n + v'.n

@[simp] theorem **add_l** (v v' : A.La) : (v + v').l = v.l + v'.l

@[simp] theorem **neg_y** (v : A.La) : (-v).y = -v.y

@[simp] theorem **neg_n** (v : A.La) : (-v).n = -v.n

@[simp] theorem **neg_l** (v : A.La) : (-v).l = -v.l

@[simp] theorem **sub_y** (v v' : A.La) : (v - v').y = v.y - v'.y

@[simp] theorem **sub_n** (v v' : A.La) : (v - v').n = v.n - v'.n

@[simp] theorem **sub_l** (v v' : A.La) : (v - v').l = v.l - v'.l

@[simp] theorem **zsmul_y** (k : ℤ) (v : A.La) : (k • v).y = k • v.y

@[simp] theorem **zsmul_n** (k : ℤ) (v : A.La) : (k • v).n = k • v.n

@[simp] theorem **zsmul_l** (k : ℤ) (v : A.La) : (k • v).l = k • v.l

@[simp] theorem **nsmul_y** (k : ℕ) (v : A.La) : (k • v).y = k • v.y

@[simp] theorem **nsmul_n** (k : ℕ) (v : A.La) : (k • v).n = k • v.n

@[simp] theorem **nsmul_l** (k : ℕ) (v : A.La) : (k • v).l = k • v.l

theorem **eq_add_parts** (v : A.La) : v = mk v.y 0 0 + mk 0 v.n 0 + mk 0 0 v.l

@[simp] theorem **smul_y** (a : O) (v : A.La) : (a • v).y = a • v.y

@[simp] theorem **smul_n_apply** (a : O) (v : A.La) (x : A.X) : (a • v).n x = v.n (star a • x)

@[simp] theorem **smul_l** (a : O) (v : A.La) : (a • v).l = A.FLam a v.l

@[simp] theorem **yLin_apply** (v : A.La) : A.yLin v = v.y

theorem **yLin_surjective** : Function.Surjective A.yLin

@[simp] theorem **degeneratingTori_basePeriodMap_fst_apply** (v : A.La) (x : A.X) : (A.P0 v).1 x = A.T v.y x + v.n x - A.H (A.ct x) v.l / (2 * Complex.I)

@[simp] theorem **degeneratingTori_basePeriodMap_snd** (v : A.La) : (A.P0 v).2 = A.ct v.y + v.l

@[simp] theorem **N_fst_apply** (v : A.La) (x : A.X) : (A.N v).1 x = A.be v.y x

@[simp] theorem **N_snd** (v : A.La) : (A.N v).2 = 0

theorem **Pw_apply** (w : ℂ) (v : A.La) : A.Pw w v = A.P0 v + w • A.N v

@[simp] theorem **Pw_fst_apply** (w : ℂ) (v : A.La) (x : A.X) : (A.Pw w v).1 x = A.T v.y x + v.n x - A.H (A.ct x) v.l / (2 * Complex.I) + w * A.be v.y x

@[simp] theorem **Pw_snd** (w : ℂ) (v : A.La) : (A.Pw w v).2 = A.ct v.y + v.l

theorem **degeneratingTori_periodMap_eq_base_of_y_eq_zero** (w : ℂ) {v : A.La} (hv : v.y = 0) : A.Pw w v = A.P0 v

theorem **degeneratingTori_basePeriodMap_smul** (a : O) (v : A.La) : A.P0 (a • v) = A.actV a (A.P0 v)

theorem **N_smul** (a : O) (v : A.La) : A.N (a • v) = A.actV a (A.N v)

theorem **Pw_smul** (w : ℂ) (a : O) (v : A.La) : A.Pw w (a • v) = A.actV a (A.Pw w v)

theorem **nn_apply** (v : A.La) : A.nn v = La.mk 0 (A.be v.y).toAddMonoidHom 0

theorem **nn_nn** (v : A.La) : A.nn (A.nn v) = 0

theorem **degeneratingTori_basePeriodMap_nilpotentMap** (v : A.La) : A.P0 (A.nn v) = A.N v

theorem **EF_apply** (v v' : A.La) : A.EF v v' = v'.n v.y - v.n v'.y + A.Ef v.l v'.l

theorem **EF_n_y** (n : A.X →+ ℤ) (y : A.X) : A.EF (La.mk 0 n 0) (La.mk y 0 0) = -n y

theorem **EF_n_left** (n : A.X →+ ℤ) (v : A.La) : A.EF (La.mk 0 n 0) v = -n v.y

theorem **EF_l_l** (l m : A.abelian₂) : (A.EF (La.mk 0 0 l) (La.mk 0 0 m) : ℝ) = (A.H l m).im

theorem **EF_isAlt** : A.EF.IsAlt

theorem **EF_smul** (a : O) (v v' : A.La) : A.EF (a • v) v' = A.EF v (star a • v')

theorem **ah_ne_zero** (l : A.abelian₂) (z : Fin A.gB → ℂ) : A.ah l z ≠ 0

theorem **ah_add** (l : A.abelian₂) (z c : Fin A.gB → ℂ) : A.ah l (z + c) = A.ah l z * Complex.exp (Real.pi * A.H c l)

theorem **mem_Theta** {j : ℕ} {α : A.X} {th : (Fin A.gB → ℂ) → ℂ} : th ∈ A.Theta j α ↔ Differentiable ℂ th ∧ ∀ (l : A.abelian₂) z, th (z + l) = (A.chi l * Complex.exp (Real.pi * A.H z l + Real.pi / 2 * A.H l l)) ^ j * Complex.exp (Real.pi * A.H (A.ct α) l) * th z

theorem **mul_mem_Theta** {j j' : ℕ} {α α' : A.X} {th th' : (Fin A.gB → ℂ) → ℂ} (h : th ∈ A.Theta j α) (h' : th' ∈ A.Theta j' α') : th * th' ∈ A.Theta (j + j') (α + α')

theorem **translate_mem_Theta** {j : ℕ} {α : A.X} {th : (Fin A.gB → ℂ) → ℂ} (h : th ∈ A.Theta j α) (μ : A.X) : (fun z => th (z + A.ct μ)) ∈ A.Theta j (α + j • μ)

theorem **differentiable_term** (j : ℕ) (α : A.X) {th : (Fin A.gB → ℂ) → ℂ} (h : Differentiable ℂ th) (μ : A.X) : Differentiable ℂ (A.term j α th μ)

theorem **two_mul_q** (μ : A.X) : 2 * A.q μ = A.be μ μ

@[simp] theorem **q_zero** : A.q 0 = 0

@[simp] theorem **q_neg** (μ : A.X) : A.q (-μ) = A.q μ

theorem **q_add** (μ ν : A.X) : A.q (μ + ν) = A.q μ + A.q ν + A.be μ ν

theorem **be_self_nonneg** (μ : A.X) : 0 ≤ A.be μ μ

theorem **two_mul_exponent** (j : ℕ) (α μ : A.X) : 2 * A.exponent j α μ = j * A.be μ μ + 2 * A.be μ α

theorem **finite_be_le** (R : ℤ) : {μ : A.X | A.be μ μ ≤ R}.Finite

theorem **be_le_exponent** {j : ℕ} (hj : j ≠ 0) (α μ : A.X) : A.be μ μ ≤ 4 * (A.exponent j α μ + A.be α α)

theorem **neg_be_le_exponent** {j : ℕ} (hj : j ≠ 0) (α μ : A.X) : -A.be α α ≤ A.exponent j α μ

theorem **finite_exponent_le** {j : ℕ} (hj : j ≠ 0) (α : A.X) (n : ℤ) : {μ | A.exponent j α μ ≤ n}.Finite

theorem **finite_exponent_eq** {j : ℕ} (hj : j ≠ 0) (α : A.X) (n : ℤ) : {μ | A.exponent j α μ = n}.Finite

@[simp] theorem **mem_level** {j : ℕ} [NeZero j] {α : A.X} {n : ℤ} {μ : A.X} : μ ∈ A.level j α n ↔ A.exponent j α μ = n

theorem **level_eq_empty** {j : ℕ} [NeZero j] {α : A.X} {n : ℤ} (hn : n < -A.be α α) : A.level j α n = ∅

theorem **coeff_FSum** (j : ℕ) [NeZero j] (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (v : A.V) (n : ℤ) : (A.FSum j α th v).coeff n = ∑ μ ∈ A.level j α n, A.term j α th μ v

theorem **coeff_FSum_eq_zero** (j : ℕ) [NeZero j] (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (v : A.V) {n : ℤ} (hn : n < -A.be α α) : (A.FSum j α th v).coeff n = 0

theorem **term_add_P0_l** {j : ℕ} {α : A.X} {th : (Fin A.gB → ℂ) → ℂ} (h : th ∈ A.Theta j α) (μ : A.X) (l : A.abelian₂) (v : A.V) : A.term j α th μ (v + A.P0 (La.mk 0 0 l)) = (A.chi l * Complex.exp (Real.pi * A.H v.2 l + Real.pi / 2 * A.H l l)) ^ j * A.term j α th μ v

theorem **term_add_P0_n** (j : ℕ) (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (μ : A.X) (n : A.X →+ ℤ) (v : A.V) : A.term j α th μ (v + A.P0 (La.mk 0 n 0)) = A.term j α th μ v

theorem **term_add_Pw_y** (j : ℕ) (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (w : ℂ) (μ y : A.X) (v : A.V) : (A.psi y * eTwoPi (v.1 y) * eTwoPi (w * A.q y)) ^ j * (A.term j α th μ (v + A.Pw w (La.mk y 0 0)) * eTwoPi (w * A.exponent j α μ)) = A.term j α th (μ + y) v * eTwoPi (w * A.exponent j α (μ + y))

@[simp] theorem **beInt_apply** (x y : ℤ) : beInt x y = 2 * x * y

@[simp] theorem **TInt_apply** (τ₀ : ℂ) (x y : ℤ) : TInt τ₀ x y = τ₀ * x * y

@[simp] theorem **coe_psiInt** (τ₀ : ℂ) (x : ℤ) : (psiInt τ₀ x : ℂ) = eTwoPi (τ₀ * x ^ 2 / 2)

theorem **psiInt_neg** (τ₀ : ℂ) (x : ℤ) : psiInt τ₀ (-x) = psiInt τ₀ x

theorem **psiInt_add** (τ₀ : ℂ) (x y : ℤ) : (psiInt τ₀ (x + y) : ℂ) = psiInt τ₀ x * psiInt τ₀ y * eTwoPi (TInt τ₀ x y)

theorem **tate_Pw** (τ₀ w : ℂ) (y : ℤ) (n : ℤ →+ ℤ) : ((tate τ₀).Pw w (La.mk (A

theorem **tate_EF** (τ₀ : ℂ) : (tate τ₀).EF (La.mk (A

theorem **tate_one_mem_Theta** (τ₀ : ℂ) (j : ℕ) (α : (tate τ₀).X) : (1 : (Fin (tate τ₀).gB → ℂ) → ℂ) ∈ (tate τ₀).Theta j α

theorem **tate_exponent** (τ₀ : ℂ) (μ : ℤ) : (tate τ₀).exponent 1 (0 : ℤ) μ = μ ^ 2

theorem **tate_Sum** (τ₀ w : ℂ) (v : (tate τ₀).V) : (tate τ₀).Sum 1 (0 : ℤ) 1 w v = ∑' μ : ℤ, eTwoPi (μ ^ 2 * (τ₀ + 2 * w) / 2 + μ * v.1 (1 : ℤ))

theorem **tate_FSum_coeff_zero** (τ₀ : ℂ) (v : (tate τ₀).V) : ((tate τ₀).FSum 1 0 1 v).coeff 0 = 1

@[simp] theorem **gaussEmb_apply** (a : GaussianInt) (i : Fin 1) : gaussEmb a i = a

@[simp] theorem **coe_gaussEquiv** (a : GaussianInt) : (gaussEquiv a : Fin 1 → ℂ) = gaussEmb a

theorem **hermForm_one** (z w : Fin 1 → ℂ) : hermForm 1 z w = z 0 * starRingEnd ℂ (w 0)

@[simp] theorem **gaussForm_apply** (a b : GaussianInt) : gaussForm a b = a.im * b.re - a.re * b.im

@[simp] theorem **gaussEf_apply** (a b : GaussianInt) : gaussEf (gaussEquiv a) (gaussEquiv b) = a.im * b.re - a.re * b.im

theorem **gaussEf_eq** (l m : gaussLam) : (gaussEf l m : ℝ) = (hermForm 1 (l : Fin 1 → ℂ) m).im

theorem **gaussEf_bij** : Function.Bijective gaussEf

theorem **gaussLam_discrete** : DiscreteTopology gaussLam

theorem **gaussLam_full** : Submodule.span ℝ (gaussLam : Set (Fin 1 → ℂ)) = ⊤

theorem **gaussChi_norm** (l : gaussLam) : ‖gaussChi l‖ = 1

theorem **gaussChi_add** (l m : gaussLam) : gaussChi (l + m) = gaussChi l * gaussChi m * Complex.exp (Real.pi * Complex.I * ((hermForm 1 (l : Fin 1 → ℂ) m).im : ℂ))

theorem **gaussChi_neg** (l : gaussLam) : gaussChi (-l) = gaussChi l

theorem **hermForm_one_pos** (z : Fin 1 → ℂ) (hz : z ≠ 0) : 0 < (hermForm 1 z z).re

theorem **gaussInt_ct_zero** (τ₀ : ℂ) : (gaussInt 0 τ₀).ct = 0

theorem **gaussInt_EF** (c τ₀ : ℂ) : (gaussInt c τ₀).EF (La.mk (A

theorem **gaussian_integers_period_component** (c τ₀ : ℂ) : ((gaussInt c τ₀).P0 (La.mk (A

@[simp] theorem **beGauss_apply** (a b : GaussianInt) : beGauss a b = 2 * (a.re * b.re + a.im * b.im)

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.ExistsChartComplexPoint

theorem **framеProd_ne_zero** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} (u : HolUnif A L g) {K : Type} [Fintype K] {j : K → L.ι} {z : K → (Fin g → ℂ)} (ε : K → ℤ) (h : ∀ k, ⊤ ≤ (u.ex (z k)).left ⁻¹ᵁ L.U (j k)) : framеProd u j z ε ≠ 0

theorem **top_le_wchart_iff** {X Y : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (j : K → L.ι) (x : Sp ℂ ⟶ Y) : ⊤ ≤ x ⁻¹ᵁ LB.wchart L φ j ↔ ∀ k, ⊤ ≤ (x ≫ φ k) ⁻¹ᵁ L.U (j k)

theorem **exists_chart_complexPoint** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) (x : Sp ℂ ⟶ X) : ∃ i, ⊤ ≤ x ⁻¹ᵁ L.U i

theorem **jFK_comp_jK_complex** : Out.jFK (R

theorem **toUnit_TF_comp_left** {B : Over (Sp ℂ)} (p : 𝟙_ (Over (Sp ℂ)) ⟶ B) : (show Sp ℂ ⟶ _ from (toUnit (Out.TF (R

theorem **ex₀_b_left** (v : d.A.V) : (show Sp ℂ ⟶ _ from (d.ex₀ v).b.left) = (d.u.ex v.2).left

theorem **ex_zero** : d.u.ex 0 = η[D₀.P.A]

theorem **gtφ_eq** (v : d.A.V) (x : d.A.X) (k : Fin 4) : (show Sp ℂ ⟶ _ from Out.DegPre.gtφ D₀ ℂ (d.ex₀ v).b (d.eX x) k) = (d.u.ex (![d.A.ct x + v.2, d.A.ct x, v.2, 0] k)).left

theorem **ex_comp_famφB** (z : Fin d.A.gB → ℂ) (α : d.A.X) (k : Fin 4) : (d.u.ex z).left ≫ D₀.famφB (d.eX α) k = (d.u.ex (![z, z + d.A.ct α, d.A.ct α, 0] k)).left

theorem **top_le_gtChart_iff** (v : d.A.V) (x : d.A.X) (j : Fin 4 → D₀.P.L.ι) : ⊤ ≤ LB.wchart D₀.P.L (Out.DegPre.gtφ D₀ ℂ (d.ex₀ v).b (d.eX x)) j ↔ ∀ k, ⊤ ≤ (d.u.ex (![d.A.ct x + v.2, d.A.ct x, v.2, 0] k)).left ⁻¹ᵁ D₀.P.L.U (j k)

theorem **top_le_famChart_iff** (z : Fin d.A.gB → ℂ) (α : d.A.X) (i : Fin 4 → D₀.P.L.ι) : ⊤ ≤ (d.u.ex z).left ⁻¹ᵁ LB.wchart D₀.P.L (D₀.famφB (d.eX α)) i ↔ ∀ k, ⊤ ≤ (d.u.ex (![z, z + d.A.ct α, d.A.ct α, 0] k)).left ⁻¹ᵁ D₀.P.L.U (i k)

theorem **hol_b** : IsHolTo Set.univ (fun v => (show Sp ℂ ⟶ _ from (d.ex₀ v).b.left))

theorem **hol_chi** (x : D₀.X) (j : Fin 4 → D₀.P.L.ι) : DifferentiableOn ℂ (fun v => chiVal (d.ex₀ v) x j) {v | ⊤ ≤ LB.wchart D₀.P.L (Out.DegPre.gtφ D₀ ℂ (d.ex₀ v).b x) j}

theorem **isTorusPt_ex₀** {v : d.A.V} (hv : v.2 = 0) : IsTorusPt (d.ex₀ v) fun x => eTwoPi (v.1 (d.eX.symm x))

theorem **shift** (w : ℂ) (la : d.A.La) : IsTorusPt (d.ex₀ (w • d.A.N la)) fun x => eTwoPi (w * d.A.be la.y (d.eX.symm x))

theorem **isZero_ex₀_zero** : (d.ex₀ 0).IsZero

theorem **isNsmul_ex₀** (k : ℕ) (v : d.A.V) : Out.GtPtF.IsNsmul k (d.ex₀ v) (d.ex₀ (k • v))

theorem **ex₀_eq_iff** (v v' : d.A.V) : d.ex₀ v = d.ex₀ v' ↔ ∃ la : d.A.La, la.y = 0 ∧ v − v' = d.A.P0 la

@[simp] theorem **qk_y** (k : ℕ) [NeZero k] (la : d.A.La) : (d.qk k la).y = d.eX la.y

@[simp] theorem **qk_g** (k : ℕ) [NeZero k] (la : d.A.La) : (d.qk k la).g = d.ex₀ ((k : ℂ)⁻¹ • d.A.P0 la)

theorem **qk_surjective** (k : ℕ) [NeZero k] : Function.Surjective (d.qk k)

theorem **qk_equiv_add** (k : ℕ) [NeZero k] (la la' : d.A.La) : Out.QkPt.Equiv D₀ ℂ (d.qk k la) (d.qk k (la + k • la'))

theorem **qk_equiv_iff** (k : ℕ) [NeZero k] (la la' : d.A.La) : Out.QkPt.Equiv D₀ ℂ (d.qk k la) (d.qk k la') ↔ ∃ m, la' = la + k • m

theorem **fr_eq_ah_mul** (l : d.A.abelian₂) (i : D₀.P.L.ι) (z : Fin d.A.gB → ℂ) (h : ⊤ ≤ (d.u.ex z).left ⁻¹ᵁ D₀.P.L.U i) : d.u.fr i z = d.A.ah l z * d.u.fr i (z + l)

theorem **ex_add_lam** (z : Fin d.A.gB → ℂ) (l : d.A.abelian₂) : d.u.ex (z + l) = d.u.ex z

theorem **isFn_fn₁** (th : D₀.P.L.Sec) : d.u.IsFn th (d.fn₁ th)

theorem **eq_fn₁** {th : D₀.P.L.Sec} {θ : (Fin d.A.gB → ℂ) → ℂ} (h : d.u.IsFn th θ) : θ = d.fn₁ th

theorem **fn₁_mem** (th : D₀.P.L.Sec) : d.fn₁ th ∈ d.A.Theta 1 0

theorem **fn₁_ne_zero** {th : D₀.P.L.Sec} (h : ∃ j, th.f j ≠ 0) : d.fn₁ th ≠ 0

theorem **IsFamFn.unique** {j : ℕ} {α : d.A.X} {s : D₀.SecB j (d.eX α)} {θ θ' : (Fin d.A.gB → ℂ) → ℂ} (h : d.IsFamFn j α s θ) (h' : d.IsFamFn j α s θ') : θ = θ'

theorem **SecFn.fn_add** (σ : d.SecFn) {j : ℕ} [NeZero j] {α : d.A.X} {s s' s'' : D₀.SecB j (d.eX α)} (h : ∀ i, s''.f i = s.f i + s'.f i) : σ.fn j α s'' = σ.fn j α s + σ.fn j α s'

theorem **SecFn.fn_smul** (σ : d.SecFn) {j : ℕ} [NeZero j] {α : d.A.X} {s s' : D₀.SecB j (d.eX α)} (a : ℂ) (h : ∀ i, s'.f i = constFn D₀.P.A.hom _ a * s.f i) : σ.fn j α s' = a • σ.fn j α s

theorem **valAt'_gv_unitOn** (t : ℂˣ) {W : (Sp ℂ).Opens} (h : ⊤ ≤ W) : valAt' (𝟙 (Sp ℂ)) W (gv (unitOn t) W).val = t

@[simp] theorem **expChar_apply** {X : Type} [AddCommGroup X] (l : X →+ ℂ) (x : X) : (expChar l (Multiplicative.ofAdd x) : ℂ) = eTwoPi (l x)

theorem **expChar_surjective** {X : Type} [AddCommGroup X] [Module.Free ℤ X] : Function.Surjective (expChar (X

theorem **expChar_eq_one_iff** {X : Type} [AddCommGroup X] (l : X →+ ℂ) : expChar l = 1 ↔ ∃ n : X →+ ℤ, ∀ x, l x = n x

theorem **frameProd_zeroUnif** {K : Type} [Fintype K] (j z) (ε : K → ℤ) : frameProd (zeroUnif A) j z ε = 1

theorem **lam_eq_zero** (l : A.abelian₂) : l = 0

theorem **mem_Theta** (j : ℕ) (α : A.X) (θ : (Fin A.gB → ℂ) → ℂ) : θ ∈ A.Theta j α

theorem **algebraMap_psiFrac** (x : A.X) : algebraMap (FractionRing ℂ) ℂ (psiFrac A x) = A.psi x

theorem **polar_bal** (a : O) (y z : A.X) : polar q (a • y) z = polar q y (star a • z)

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.IsCoprimeTwoSubRtTwoAddRt

theorem **coe_rt_sq** (H : Host E) : (H.rt : E) ^ 2 = −(H.d : E)

theorem **conj_embedding_rt** (H : Host E) (τ : E →+* ℂ) : starRingEnd ℂ (τ (H.rt : E)) = −τ (H.rt : E)

@[simp] theorem **bar_rt** (H : Host E) : H.bar H.rt = −H.rt

@[simp] theorem **bar_bar** (H : Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

@[simp] theorem **cmHost_bar_halfIntegerGenerator** (H : Host E) : H.bar H.w0 = 1 − H.w0

theorem **rt_eq** (H : Host E) : H.rt = 2 * H.w0 − 1

theorem **rt_mem_adjoin** (H : Host E) : H.rt ∈ Algebra.adjoin ℤ {H.w0}

theorem **rt_ne_zero** (H : Host E) : H.rt ≠ 0

theorem **two_sub_rt_mul_two_add_rt** (H : Host E) : (2 − H.rt) * (2 + H.rt) = ((H.d + 4 : ℕ) : 𝒪 E)

theorem **map_bar_span_two_sub_rt** (H : Host E) : (Ideal.span {2 − H.rt}).map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {2 + H.rt}

theorem **isCoprime_two_sub_rt_two_add_rt** (H : Host E) : IsCoprime (2 − H.rt) (2 + H.rt)

@[simp] theorem **level_M** (H : Host E) (A : ℕ) (hA : 0 < A) : (level H A hA).M = H.d + 4

@[simp] theorem **level_Mb** (H : Host E) (A : ℕ) (hA : 0 < A) : (level H A hA).Mb = Ideal.span {2 − H.rt}

@[simp] theorem **level_A** (H : Host E) (A : ℕ) (hA : 0 < A) : (level H A hA).A = A

theorem **Mb_ne_top** {H : Host E} (l : Level H) : l.Mb ≠ ⊤

theorem **level_goodAt** (H : Host E) (A : ℕ) (hA : 0 < A) {p : ℕ} (hp : ¬ p ∣ H.d + 4) : (level H A hA).GoodAt p

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.IsFShiftZeroIff

theorem **IsFactor.hasSum** {lin : ℂ → A.La → A.V →ₗ[ℂ] ℂ} {cst : ℂ → A.La → ℂ} (h : IsFactor A lin cst) (w : ℂ) (la : A.La) (v : A.V) (j : ℕ) (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (hth : th ∈ A.Theta j α) {s : ℂ} (hs : HasSum (fun m => A.term j α th m v * eTwoPi (w * A.exponent j α m)) s) : HasSum (fun m => A.term j α th m (v + A.Pw w la) * eTwoPi (w * A.exponent j α m)) (Complex.exp (lin w la v + cst w la) ^ j * s)

theorem **IsFactor.sum** {lin : ℂ → A.La → A.V →ₗ[ℂ] ℂ} {cst : ℂ → A.La → ℂ} (h : IsFactor A lin cst) (w : ℂ) (la : A.La) (v : A.V) (j : ℕ) (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (hth : th ∈ A.Theta j α) : A.Sum j α th w (v + A.Pw w la) = Complex.exp (lin w la v + cst w la) ^ j * A.Sum j α th w v

theorem **not_isFShift_zero** (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (la : A.La) (z : A.V) (S : ℂ⟦X⟧) : ¬ IsFShift A 0 α th la z S

theorem **isFShift_zero_iff** (k : ℕ) [NeZero k] (α : A.X) (th : (Fin A.gB → ℂ) → ℂ) (z : A.V) (S : ℂ⟦X⟧) : IsFShift A k α th 0 z S ↔ S = A.FSum k α th z

theorem **GenPts.LevelPt.ne_zero** {Φ : GenPts A t y} {k : ℕ} {p : Over.mk y ⟶ t.G} {la : A.La} (h : Φ.LevelPt k p la) : k ≠ 0

theorem **GenPts.LevelPt.dvd** {Φ : GenPts A t y} {k : ℕ} {p : Over.mk y ⟶ t.G} {la : A.La} (h : Φ.LevelPt k p la) (α : A.X) : (k : ℤ) ∣ A.be la.y α

theorem **GenPts.levelPt_one** (Φ : GenPts A t y) (k : ℕ) [NeZero k] : Φ.LevelPt k 1 0

theorem **FPts.lev_one** {V : Type} [AddCommGroup V] {act : 𝒪 E → V → V} {I : ℕ → Type} {F : Type} [Field F] {y₁ : Sp F ⟶ S} {φ : ∀ j, I j → V → F} (θ : FPts act t y₁ φ) (k : ℕ) (hk : 1 ≤ k) (i₀ : I k) (z₀ : V) (h : φ k i₀ z₀ ≠ 0) : θ.Lev k 1 (φ k)

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.IsFramedPointChangedByLetters

theorem **ArcDatum.overBase** (d : ArcDatum Xo c K₁) : d.OverBase

theorem **lieImg_iff_isLieImage** (W : Wit Y a χ) (ν : Out.GtPt a.D₀ (epsR K)) (v : epsR (PowerSeries K) ⟶ W.t.G) : LieImg W ν v ↔ IsLieImage W ν v

theorem **gtTanR_iff_gtTan** (ν : Out.GtPt a.D₀ (epsR K)) : GtTanR ν ↔ GtTan (a

theorem **arcBasis_iff_arcLieBasis** [Algebra c.F K] (W : Wit Y a χ) (ρ : (E →+* ℂ) → (𝒪 E →+* K)) (hρ : ∀ τ x, ρ τ x = algebraMap c.F K (embK c τ x)) (e : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Out.GtPt a.D₀ (epsR K)) : ArcBasis W ρ e ↔ ArcLieBasis W e

theorem **IsArcFr.isArcPt3At** [Algebra c.F K] [Algebra c.F (ArcS Dw K)] {W : Wit Y a χ} {e : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Out.GtPt a.D₀ (epsR K)} {th : a.D₀.P.L.Sec} {y : FrPt Xo c (ArcS Dw K)} (h : IsArcFr W e th (arcι Dw K) y) (hth : ∃ j, th.f j ≠ 0) : IsArcPt3At Y a χ W y th

theorem **IsArcFr.isFramedPoint_changed_by_letters** [Algebra c.F K] [Algebra c.F (ArcS Dw K)] {W : Wit Y a χ} {e : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Out.GtPt a.D₀ (epsR K)} {th : a.D₀.P.L.Sec} {ζ : ArcS Dw K} {y₀ y : FrPt Xo c (ArcS Dw K)} (h : IsArcFr W e th (arcι Dw K) y₀) (hc : GlobalTheory.Frames.FrChg (letM Dw K) ζ y₀ y) (hb : χ ≫ Y.s = spMap (PowerSeries.C (R

theorem **inBC_iff_isBC** (f : A →+* A') (a : ArcIn Dw l β A b) (a' : ArcIn Dw l β A' (f.comp b)) (p : Out.IsPullDeg f a.D₀ a'.D₀) : InBC f a a' p ↔ GlobalTheory.ArcPull.IsBC f a a' p

theorem **ThMul.refl** (th : D₀.P.L.Sec) : ThMul 1 th th

theorem **ThMul.trans** {γ δ : A} {t₁ t₂ t₃ : D₀.P.L.Sec} (h : ThMul γ t₁ t₂) (h' : ThMul δ t₂ t₃) : ThMul (δ * γ) t₁ t₃

theorem **ThNZ.ne_zero** {th : D₀.P.L.Sec} (h : ThNZ th) : ∃ j, th.f j ≠ 0

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.IsPrimitiveRootArcRoot

@[simp] theorem **rootChar_apply** (ζ : G) (f : X →ₗ[ℤ] ℤ) (χ : X) : rootChar ζ f (Multiplicative.ofAdd χ) = ζ ^ f χ

theorem **rootChar_add** (ζ : G) (f g : X →ₗ[ℤ] ℤ) : rootChar ζ (f + g) = rootChar ζ f * rootChar ζ g

theorem **rootChar_pow** (ζ : G) (f : X →ₗ[ℤ] ℤ) (m : ℕ) : rootChar (ζ ^ m) f = rootChar ζ f ^ m

theorem **rootChar_pow_eq_one** {ζ : G} {n : ℕ} (h : ζ ^ n = 1) (f : X →ₗ[ℤ] ℤ) : rootChar ζ f ^ n = 1

theorem **rootChar_eq_iff** {ζ : G} {n : ℕ} (hn : 0 < n) (h : IsPrimitiveRoot ζ n) (f g : X →ₗ[ℤ] ℤ) : rootChar ζ f = rootChar ζ g ↔ ∃ e, f = g + n • e

theorem **exists_rootChar_eq** [Module.Free ℤ X] {F : Type} [CommRing F] [IsDomain F] {ζ : Fˣ} {n : ℕ} (hn : 0 < n) (h : IsPrimitiveRoot ζ n) (t : Multiplicative X →* Fˣ) (ht : t ^ n = 1) : ∃ f, rootChar ζ f = t

theorem **polar_one** (a b : X) : polar (K

theorem **polar_one_smul** {O : Type} [Ring O] [StarRing O] [Module O X] (a : O) (y z : X) : polar (K

theorem **isIota_split** (y : X) (t : Multiplicative X →* Fˣ) : GtPtF.IsIota (D

@[simp] theorem **splitPt_y** {n : ℕ} (y : X) (t : Multiplicative X →* Fˣ) (ht : t ^ n = 1) : (splitPt K F y t ht).y = y

@[simp] theorem **splitPt_g** {n : ℕ} (y : X) (t : Multiplicative X →* Fˣ) (ht : t ^ n = 1) : (splitPt K F y t ht).g = pt _ F t

theorem **exists_eq_splitPt** {n : ℕ} (p : QkPt (splitPre K X) F n) : ∃ y t ht, p = splitPt K F y t ht

theorem **equiv_splitPt** {n : ℕ} (y y' : X) (t t' : Multiplicative X →* Fˣ) (ht : t ^ n = 1) (ht' : t' ^ n = 1) : QkPt.Equiv _ F (splitPt K F y t ht) (splitPt K F y' t' ht') ↔ (∃ z, y' = y + n • z) ∧ t' = t

theorem **equiv_refl_split** {n : ℕ} (p : QkPt (splitPre K X) F n) : QkPt.Equiv _ F p p

@[simp] theorem **qkPt_y** (a : HypLattice H) : (qkPt K ζ hζ a).y = a.1

@[simp] theorem **qkPt_g** (a : HypLattice H) : (qkPt K ζ hζ a).g = pt _ F (rootChar ζ (toDual H a.2))

theorem **qkAdd_qkPt** (a b : HypLattice H) : QkAdd (qkPt K ζ hζ a) (qkPt K ζ hζ b) (qkPt K ζ hζ (a + b))

theorem **qkAct_qkPt** (x : 𝒪 E) (a : HypLattice H) : letI

theorem **weilQkIs_qkPt** (a b : HypLattice H) : WeilQkIs _ F n (qkPt K ζ hζ a) (qkPt K ζ hζ b) ((ζ ^ hypForm H a b : Fˣ) : F)

theorem **isNsmul_qkPt** {n' m : ℕ} {ζ' : Fˣ} (hζ' : ζ' ^ n' = 1) (h : ζ ^ m = ζ') (a : HypLattice H) : GtPtF.IsNsmul m (qkPt K ζ hζ a).g (qkPt K ζ' hζ' a).g

theorem **equiv_qkPt** (hn : 0 < n) (h : IsPrimitiveRoot ζ n) (a b : HypLattice H) : QkPt.Equiv _ F (qkPt K ζ hζ a) (qkPt K ζ hζ b) ↔ ∃ e, a = b + n • e

theorem **exists_qkPt_eq** (hn : 0 < n) (h : IsPrimitiveRoot ζ n) (p : QkPt (splitPre K (𝒪 E)) F n) : ∃ a : HypLattice H, qkPt K ζ hζ a = p

theorem **qkExt_qkPt** {F' : Type} [Field F'] [Algebra (FractionRing K) F'] (f : F →+* F') (hf : f.comp (algebraMap (FractionRing K) F) = algebraMap (FractionRing K) F') {ζ' : F'ˣ} (hζ' : ζ' ^ n = 1) (h : (ζ' : F') = f ζ) (a : HypLattice H) (q : QkPt (splitPre K (𝒪 E)) F' n) : QkExt f (qkPt K ζ hζ a) q ↔ q = qkPt K ζ' hζ' a

theorem **pow_eq_one_of_prim** {ζ : ℕ → Fˣ} (h : ∀ n, 0 < n → IsPrimitiveRoot (ζ n) n) (n : ℕ) : ζ n ^ n = 1

@[simp] theorem **tateIsoQ_ζ** (ζ : ℕ → Fˣ) (hp : ∀ n, 0 < n → IsPrimitiveRoot (ζ n) n) (hc : ∀ n m, 0 < n → 0 < m → ζ (n * m) ^ m = ζ n) : (tateIsoQ H K ζ hp hc).ζ = ζ

@[simp] theorem **tateIsoQ_ev** (ζ : ℕ → Fˣ) (hp : ∀ n, 0 < n → IsPrimitiveRoot (ζ n) n) (hc : ∀ n m, 0 < n → 0 < m → ζ (n * m) ^ m = ζ n) (n : ℕ) (a : HypLattice H) : (tateIsoQ H K ζ hp hc).ev n a = qkPt K (ζ n) (pow_eq_one_of_prim hp n) a

theorem **isPrimitiveRoot_arcRoot** {a : ℕ} (h : a ∣ l.M') : IsPrimitiveRoot (arcRoot c k bF h) a

theorem **arcIn_D₀** : (arcIn c k bF).D₀ = toricPre k (𝒪 E) fun _ => 1

theorem **arcIn_be_apply** (y z : 𝒪 E) : (arcIn c k bF).be y z = 2 * l.M' * trForm H y z

theorem **arcIn_be_even** (y : 𝒪 E) : (2 : ℤ) ∣ (arcIn c k bF).be y y

theorem **arcIn_eta_equiv_iff** (a b : HypLattice H) : QkPt.Equiv _ _ ((arcIn c k bF).eta a) ((arcIn c k bF).eta b) ↔ ∃ e, a = b + l.M • e

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.NonemptyGtAnOfToric

theorem **AnArc.exists_rep** {O : Type} [CommRing O] [StarRing O] (A : AnArc O) (j : ℕ) [NeZero j] : ∃ (C : Type) (_ : Fintype C) (rep : C → A.X), ∀ α₀, ∃! t, ∃ mu, rep t = α₀ + j • mu

theorem **nonempty_gtAn_of_toric** {b : O →+* ℂ} (a : ArcIn D l β ℂ b) (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (hD : a.D₀ = ToricData.toricPre ℂ X fun _ => 1) : Nonempty (GtAn a)

theorem **nonempty_gtAn_arcIn** (c : Coeff l) (bF : c.F →+* ℂ) : Nonempty (GtAn (HostModel.arcIn c ℂ bF))

theorem **exists_gtAn_arcIn** : ∃ (_ : Algebra (FractionRing ℂ) ℂ) (_ : IsScalarTower ℂ (FractionRing ℂ) ℂ), Nonempty (GtAn (HostModel.arcIn (HostModel.coeff (HostModel.level HostModel.host 5 (by norm_num))) ℂ (HostModel.coeff _).F.val.toRingHom))

@[simp] theorem **y_apply** : letI

theorem **y_onto** : Function.Surjective Λ.y

theorem **ker_y** : letI

theorem **shift_be** : letI

theorem **dvd_be** : letI

theorem **A_dvd_be** : letI

theorem **M_dvd_be** : letI

theorem **exists_lifts** {k : ℕ} [NeZero k] (p : Out.QkPt a.D₀ (FractionRing ℂ) k) : letI

theorem **y_of_lifts** {k : ℕ} [NeZero k] (p : Out.QkPt a.D₀ (FractionRing ℂ) k) : letI

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.NonemptyOfFrameValues

theorem **nonempty_toric** : Nonempty (GtAn (arcIn c ℂ bF))

theorem **y_of_lifts** {k : ℕ} [NeZero k] (p : Out.QkPt a.D₀ (FractionRing ℂ) k) : letI

theorem **exists_closed** : ∃ (_ : Algebra (FractionRing ℂ) ℂ) (_ : IsScalarTower ℂ (FractionRing ℂ) ℂ), Nonempty (GtAn (arcIn (coeff (level host 5 (by norm_num))) ℂ (coeff _).F.val.toRingHom))

theorem **nonempty_of_frameValues** (Λ : letI

theorem **subsingleton_secFn** : Subsingleton d.SecFn

theorem **psi_eq** (x : d.A.X) : ∃ j : PUnit → D₀.P.L.ι, (d.A.psi x : ℂ) = valAt' (Out.jFK (R

theorem **isZero_torus_iff** (w : d.A.X →+ ℂ) : (d.ex₀ (w, 0)).IsZero ↔ ∃ n : d.A.X →+ ℤ, ∀ x, w x = n x

theorem **exists_not_isZero** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {D : Datum H} {l : Level H} {c : Coeff l} {O : Type} [CommRing O] {β : O →+* c.F} [Algebra (FractionRing ℂ) ℂ] [IsScalarTower ℂ (FractionRing ℂ) ℂ] {b : O →+* ℂ} {a : ArcIn D l β ℂ b} (Λ : GtAn₀ a) : letI

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.OneAddZeta21PowSevenAddPowFourteenEqZero

theorem **sq_sub_smul_derivation_eq_zero** {A B M : Type*} [CommRing A] [CommRing B] [Algebra A B] [AddCommGroup M] [Module A M] [Module B M] (D : Derivation A B M) (σ : B →+* B) (hσ : ∀ b, σ (σ b) = b) (hfix : ∀ b, σ b = b → b ∈ Set.range (algebraMap A B)) (w b : B) : (w - σ w) ^ 2 • D b = 0

theorem **formallyUnramified_of_isCoprime** {A B : Type*} [CommRing A] [CommRing B] [Algebra A B] (σ : B →+* B) (hσ : ∀ b, σ (σ b) = b) (hfix : ∀ b, σ b = b → b ∈ Set.range (algebraMap A B)) (w₁ w₂ : B) (h : IsCoprime (w₁ - σ w₁) (w₂ - σ w₂)) : Algebra.FormallyUnramified A B

theorem **unramified_of_isCoprime** (w₁ w₂ : 𝒪 E) (h : IsCoprime (w₁ - IsCMField.ringOfIntegersComplexConj E w₁) (w₂ - IsCMField.ringOfIntegersComplexConj E w₂)) : Algebra.Unramified (𝒪 (maximalRealSubfield E)) (𝒪 E)

theorem **complexConj_eq_neg_of_sq_eq_neg** {x : E} {n : ℕ} (hx : x ^ 2 = -(n : E)) : IsCMField.complexConj E x = -x

theorem **_root_.GlobalTheory.HostModel.coe_sq_eq_neg** {x : 𝒪 E} {n : ℕ} (hx : x ^ 2 = -(n : 𝒪 E)) : (x : E) ^ 2 = -(n : E)

theorem **sub_complexConj_eq_of_two_mul** {w x : 𝒪 E} {n : ℕ} (hx : x ^ 2 = -(n : 𝒪 E)) (hw : 2 * w = 1 + x) : w - IsCMField.ringOfIntegersComplexConj E w = x

theorem **unramified_of_sqrt** {x₁ x₂ w₁ w₂ : 𝒪 E} {m n : ℕ} (hmn : m.Coprime n) (h₁ : x₁ ^ 2 = -(m : 𝒪 E)) (h₂ : x₂ ^ 2 = -(n : 𝒪 E)) (hw₁ : 2 * w₁ = 1 + x₁) (hw₂ : 2 * w₂ = 1 + x₂) : Algebra.Unramified (𝒪 (maximalRealSubfield E)) (𝒪 E)

theorem **_root_.GlobalTheory.HostModel.exists_embedding_rt_eq** (H : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ∃ τ : E →+* ℂ, τ (H.rt : E) = H.rt0

theorem **isPrimitiveRoot_primitiveRootCyclotomicField21** : IsPrimitiveRoot primitiveRootCyclotomicField21 21

theorem **sum_zeta21_pow_multiples_of_three_eq_zero** : 1 + primitiveRootCyclotomicField21 ^ 3 + primitiveRootCyclotomicField21 ^ 6 + primitiveRootCyclotomicField21 ^ 9 + primitiveRootCyclotomicField21 ^ 12 + primitiveRootCyclotomicField21 ^ 15 + primitiveRootCyclotomicField21 ^ 18 = 0

theorem **halfSqrtNegSeven_quadratic** : halfSqrtNegSeven ^ 2 − halfSqrtNegSeven + 2 = 0

private theorem **isPrimitiveRoot_hostModel_primitiveRoot21** : IsPrimitiveRoot _root_.GlobalTheory.HostModel.primitiveRootCyclotomicField21 21

theorem **_root_.GlobalTheory.HostModel.sqrtNegSeven_sq** : _root_.GlobalTheory.HostModel.sqrtNegSeven ^ 2 = −((7 : ℕ) : 𝒪 _root_.GlobalTheory.HostModel.E21)

private theorem **ringOfIntegersComplexConj_zeta21_eq_pow_twenty** : IsCMField.ringOfIntegersComplexConj _root_.GlobalTheory.HostModel.E21 _root_.GlobalTheory.HostModel.primitiveRootCyclotomicField21 = _root_.GlobalTheory.HostModel.primitiveRootCyclotomicField21 ^ 20

private theorem **isUnit_one_sub_of_isPrimitiveRoot_twentyOne** {R : Type*} [CommRing R] [IsDomain R] {u : R} (hu : IsPrimitiveRoot u 21) : IsUnit (1 − u)

theorem **_root_.GlobalTheory.HostModel.unramified_E21_over_maximalRealSubfield** : Algebra.Unramified (𝒪 (maximalRealSubfield _root_.GlobalTheory.HostModel.E21)) (𝒪 _root_.GlobalTheory.HostModel.E21)

@[simp] theorem **host_d** : host.d = 7

@[simp] theorem **host_rt** : host.rt = sqrtNegSeven

@[simp] theorem **host_w0_eq_halfSqrtNegSeven** : host.w0 = halfSqrtNegSeven

@[simp] theorem **host_rt0_eq_I_mul_sqrt_seven** : host.rt0 = Complex.I * (Real.sqrt 7 : ℝ)

theorem **finrank_rat_E21_eq_twelve** : Module.finrank ℚ E21 = 12

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.PsiWeightZeroLinearCombination

theorem **wchart_triv** (φ : J → (Y ⟶ Z)) (j : J → (Inh.trivLB Z).ι) : LB.wchart (Inh.trivLB Z) φ j = ⊤

theorem **res_self** {U : Y.Opens} (h : U ≤ U) (x : Γ(Y, U)ˣ) : res h x = x

theorem **gv_inj** {V : Y.Opens} (hV : V = ⊤) {a b : Γ(Y, ⊤)ˣ} (h : gv a V = gv b V) : a = b

private theorem **res_res_lemma** {Y : Scheme.{0}} {U V W : Y.Opens} (h1 : V ≤ U) (h2 : W ≤ V) (x : Γ(Y, U)ˣ) : res h2 (res h1 x) = res (h2.trans h1) x

private theorem **linComb_trivLB_U_eq_univ** (φ : J → (Y ⟶ Z)) (ε : J → ℤ) (j : ((Inh.trivLB Z).LinComb φ ε).ι) : ((Inh.trivLB Z).LinComb φ ε).U j = ⊤

theorem **exists_isConst** {φ : J → (Y ⟶ Z)} {ε : J → ℤ} (t : LB.Gen (LB.LinComb (Inh.trivLB Z) φ ε)) : ∃ a : Γ(Y, ⊤)ˣ, IsConst t a

theorem **isConst_unique** {φ : J → (Y ⟶ Z)} {ε : J → ℤ} {t : LB.Gen (LB.LinComb (Inh.trivLB Z) φ ε)} {a b : Γ(Y, ⊤)ˣ} (ha : IsConst t a) (hb : IsConst t b) : a = b

theorem **gen_eq_of_isConst** {L : LB Y} {t t' : LB.Gen L} {a : Γ(Y, ⊤)ˣ} (h : IsConst t a) (h' : IsConst t' a) : t = t'

theorem **eq_of_corr** {φ : J → (Y ⟶ Z)} {ε : J → ℤ} {φ' : J' → (Y ⟶ Z)} {ε' : J' → ℤ} {t : LB.Gen (LB.LinComb (Inh.trivLB Z) φ ε)} {t' : LB.Gen (LB.LinComb (Inh.trivLB Z) φ' ε')} {a b : Γ(Y, ⊤)ˣ} (ha : IsConst t a) (hb : IsConst t' b) (h : LB.Gen.Corr (Inh.trivLB Z) φ ε φ' ε' t t') : a = b

theorem **corr_iff_eq** {φ : J → (Y ⟶ Z)} {ε : J → ℤ} {φ' : J' → (Y ⟶ Z)} {ε' : J' → ℤ} {t : LB.Gen (LB.LinComb (Inh.trivLB Z) φ ε)} {t' : LB.Gen (LB.LinComb (Inh.trivLB Z) φ' ε')} {a b : Γ(Y, ⊤)ˣ} (ha : IsConst t a) (hb : IsConst t' b) : LB.Gen.Corr (Inh.trivLB Z) φ ε φ' ε' t t' ↔ a = b

theorem **pullTop_injective** (f : W ⟶ Y) [IsIso f] : Function.Injective (pullTop f)

theorem **isConst_one_of_corr_oneGen** {φ : J → (Y ⟶ Z)} {ε : J → ℤ} (u : LB.Gen (LB.LinComb (Inh.trivLB Z) φ ε)) (f : W ⟶ Y) [IsIso f] (h : LB.Gen.Corr (Inh.trivLB Z) _ _ _ _ (LB.pullGen _ φ ε u f) (LB.oneGen _)) : IsConst u 1

theorem **unitOn_one** : unitOn (1 : Aˣ) = 1

theorem **unitOn_mul** (u v : Aˣ) : unitOn (u * v) = unitOn u * unitOn v

theorem **unitOn_inv** (u : Aˣ) : unitOn u⁻¹ = (unitOn u)⁻¹

theorem **unitOn_zpow** (u : Aˣ) (n : ℤ) : unitOn (u ^ n) = unitOn u ^ n

theorem **unitOn_pow** (u : Aˣ) (n : ℕ) : unitOn (u ^ n) = unitOn u ^ n

theorem **unitOn_bijective** : Function.Bijective (unitOn : Aˣ → Γ(Sp A, ⊤)ˣ)

theorem **unitOn_inj** {u v : Aˣ} : unitOn u = unitOn v ↔ u = v

theorem **pullTop_unitOn** (f : A →+* B) (u : Aˣ) : pullTop (Spec.map (CommRingCat.ofHom f)) (unitOn u) = unitOn (Units.map f.toMonoidHom u)

theorem **polar_comm** (a b : X) : polar q a b = polar q b a

theorem **psi_isConst** (y : X) : IsConst ((toricPre K X q).psi y) (unitOn (q y))

theorem **rigK_isConst** : IsConst (toricPre K X q).rigK 1

theorem **tau_isConst** (a b : X) : IsConst ((toricPre K X q).tau a b) (unitOn (polar q a b))

theorem **psiEven_iff** : (toricPre K X q).PsiEven ↔ ∀ y, q (−y) = q y

theorem **psiEven** (h : ∀ y, q (−y) = q y) : (toricPre K X q).PsiEven

theorem **tauBimult_iff** : (toricPre K X q).TauBimult ↔ ∀ a a' b, polar q (a + a') b = polar q a b * polar q a' b

theorem **tauBimult** (h : ∀ a a' b, polar q (a + a') b = polar q a b * polar q a' b) : (toricPre K X q).TauBimult

theorem **isConst_one_of_zero2** {J : Type} [Fintype J] {φ : J → (((toricPre K X q).P.A ⊗ (toricPre K X q).P.A).left ⟶ (toricPre K X q).P.A.left)} {ε : J → ℤ} (u : LB.Gen (LB.LinComb (toricPre K X q).P.L φ ε)) (h : LB.Gen.Corr (toricPre K X q).P.L _ _ _ _ (LB.pullGen _ φ ε u (zero2 (toricPre K X q).P.A)) (LB.oneGen _)) : IsConst u 1

theorem **psi_weight_zero_linear_combination** (O : Type) [CommRing O] [StarRing O] [Module O X] (h) (w : O) (hw : ∀ y, polar q y (w • y) = q y) : TraceFormTwistIso (toricPre K X q) (toricOAct q O h) w

theorem **jK_toOpen** (V : (Sp K).Opens) (x : K) : ((jK K).app V).hom (DegPre.toOpen V x) = ((TK K).left.presheaf.map (homOfLE le_top).op).hom ((Scheme.ΓSpecIso (CommRingCat.of (FractionRing K))).inv.hom (algebraMap K (FractionRing K) x))

theorem **psi_eq_toOpen** (y : X) (r : K) (hq : (q y : FractionRing K) = algebraMap K _ r) (j : PUnit → (toricPre K X q).P.L.ι) (V : (Sp K).Opens) (h : jK K ⁻¹ᵘ V ≤ (LB.LinComb (toricPre K X q).P.L (K

theorem **psiUnitOn** (y : X) (u : Kˣ) (h : (q y : FractionRing K) = algebraMap K _ u) : PsiUnitOn (toricPre K X q) y Set.univ

theorem **psiIn** (y : X) (J : Ideal K) (r : K) (hr : r ∈ J) (h : (q y : FractionRing K) = algebraMap K _ r) : (toricPre K X q).PsiIn y J

theorem **lieMult_zero** (O : Type) [CommRing O] [StarRing O] [Module O X] (h) {F : Type} [Field F] (f : K →+* F) (τ : O →+* F) : LieMult (ofRingAct (toricOAct q O h).onB.toRingAct) (spMap f) τ 0

theorem **cubAt_isConst** (x x' y : TF (R

theorem **pt_isConst** (t : Multiplicative X →* Fˣ) (χ : X) : IsConst ((pt q F t).s χ) (unitOn (t (Multiplicative.ofAdd χ)))

theorem **pt_bijective** : Function.Bijective (pt q F)

private theorem **eq_of_map_eq_of_eq_univ** {Y : Scheme.{0}} {W : Y.Opens} (hW : W = ⊤) (h : W ≤ ⊤) {a b : Γ(Y, ⊤)} (e : (Y.presheaf.map (homOfLE h).op).hom a = (Y.presheaf.map (homOfLE h).op).hom b) : a = b

theorem **psiIn_iff** (y : X) (J : Ideal K) : (toricPre K X q).PsiIn y J ↔ ∃ r ∈ J, (q y : FractionRing K) = algebraMap K _ r

theorem **polar_zpow_sq** (u : (FractionRing K)ˣ) (a b : ℤ) : polar (fun n : ℤ => u ^ (n ^ 2)) a b = u ^ (2 * a * b)

theorem **toricPre_tate** : toricPre InhTate.R ℤ (fun n => piK ^ (n ^ 2)) = tatePre

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.SqrtNegDiscMemCoeffField

theorem **_root_.GlobalTheory.HostModel.isPrimitiveRoot_zeta** {n : ℕ} (hn : 0 < n) : IsPrimitiveRoot (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.zeta n) n

lemma **_root_.GlobalTheory.HostModel.zeta_mul_pow** {n m : ℕ} (hn : 0 < n) (hm : 0 < m) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.zeta (n * m) ^ m = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.zeta n

theorem **zeta_pow_div** {a n : ℕ} (hn : 0 < n) (h : a ∣ n) : zeta n ^ (n / a) = zeta a

theorem **zeta_M'_pow_div_A** (l : Level H) : zeta l.M' ^ (l.M' / l.A) = zeta l.A

theorem **_root_.GlobalTheory.HostModel.zeta_mem_coeff_of_dvd** {l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H} (c : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l) {a : ℕ} (h : a ∣ l.M') : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.zeta a ∈ c.F

theorem **zeta_M_mem_coeff** {l : Level H} (c : Coeff l) : zeta l.M ∈ c.F

theorem **_root_.GlobalTheory.HostModel.zeta_A_mem_coeff** {l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H} (c : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.zeta l.A ∈ c.F

theorem **_root_.GlobalTheory.HostModel.sqrtNegDisc_mem_coeffField** {l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H} (c : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l) : H.rt0 ∈ c.F

theorem **coeff_F** (l : Level H) : (coeff l).F = IntermediateField.normalClosure ℚ E ℂ ⊔ IntermediateField.adjoin ℚ {zeta l.M'}

theorem **coeff_F_le_iff** (l : Level H) (F : IntermediateField ℚ ℂ) : (coeff l).F ≤ F ↔ zeta l.M' ∈ F ∧ ∀ (τ : E →+* ℂ) (a : E), τ a ∈ F

theorem **_root_.GlobalTheory.HostModel.exists_roots** (L : IntermediateField ℚ ℂ) [FiniteDimensional ℚ L] (F : Type) [Field F] [IsAlgClosed F] (f : L →+* F) : ∃ ζ : ℕ → Fˣ, (∀ n, 0 < n → IsPrimitiveRoot (ζ n) n) ∧ (∀ n m, 0 < n → 0 < m → ζ (n * m) ^ m = ζ n) ∧ ∀ n, 0 < n → ∀ z : L, (z : ℂ) = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.zeta n → (ζ n : F) = f z

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.ToMatrixTrFormBaseChange

@[**ext**] theorem ext {f g : ConjDual H} (h : ∀ x, toDual H f x = toDual H g x) : f = g

@[simp] theorem **smul_apply** (a : 𝒪 E) (f : ConjDual H) (x : 𝒪 E) : toDual H (a • f) x = toDual H f (H.bar a * x)

theorem **finrank_int** : finrank ℤ (ConjDual H) = finrank ℤ (𝒪 E)

@[simp] theorem **toDual_traceDual** (H : Host E) (x : 𝒪 E) : toDual H (traceDual H x) = trForm H x

theorem **traceDual_apply** (H : Host E) (x y : 𝒪 E) : toDual H (traceDual H x) y = Algebra.trace ℤ (𝒪 E) (H.bar x * y)

theorem **traceDual_injective** (H : Host E) : Function.Injective (traceDual H)

theorem **finrank_eq_of_injective** {M N : Type} [AddCommGroup M] [AddCommGroup N] [Module (𝒪 E) M] [Module (𝒪 E) N] [Module.Finite ℤ N] (φ : M →ₗ[𝒪 E] N) (hφ : Function.Injective φ) (h : finrank ℤ M = finrank ℤ N) : finrank (𝒪 E) M = finrank (𝒪 E) N

theorem **ConjDual.finrank_eq_one** (H : Host E) : finrank (𝒪 E) (ConjDual H) = 1

theorem **finrank_hypLattice** (H : Host E) : finrank (𝒪 E) (HypLattice H) = 2

@[simp] theorem **hypForm_apply** (H : Host E) (u v : HypLattice H) : hypForm H u v = toDual H u.2 v.1 − toDual H v.2 u.1

theorem **hypForm_self** (H : Host E) (u : HypLattice H) : hypForm H u u = 0

theorem **hypForm_smul** (H : Host E) (a : 𝒪 E) (u v : HypLattice H) : hypForm H (a • u) v = hypForm H u (H.bar a • v)

theorem **hypForm_bijective** (H : Host E) : Function.Bijective (hypForm H)

theorem **hypBasis_inl** (H : Host E) (i : BasisIndex E) : hypBasis H (Sum.inl i) = (RingOfIntegers.basis E i, 0)

theorem **hypBasis_inr** (H : Host E) (i : BasisIndex E) : hypBasis H (Sum.inr i) = (0, (toDual H).symm ((RingOfIntegers.basis E).coord i))

theorem **hypBasis_repr_inl** (H : Host E) (u : HypLattice H) (i : BasisIndex E) : (hypBasis H).repr u (Sum.inl i) = (RingOfIntegers.basis E).repr u.1 i

theorem **hypBasis_repr_inr** (H : Host E) (u : HypLattice H) (i : BasisIndex E) : (hypBasis H).repr u (Sum.inr i) = toDual H u.2 (RingOfIntegers.basis E i)

theorem **gram_transpose** (H : Host E) : (gram H)ᵀ = gram H

theorem **toMatrix_trForm_baseChange** (H : Host E) : LinearMap.BilinForm.toMatrix ((RingOfIntegers.basis E).baseChange ℝ) ((trForm H).baseChange ℝ) = gram H

theorem **gram_posDef** (H : Host E) : (gram H).PosDef

theorem **isUnit_det_gram** (H : Host E) : IsUnit (gram H).det

theorem **toMatrix_hypJ** (H : Host E) : LinearMap.toMatrix (hypRealBasis H) (hypRealBasis H) (hypJ H) = Matrix.fromBlocks 0 (−(gram H)⁻¹) (gram H) 0

theorem **hypJ_hypJ** (H : Host E) (x) : hypJ H (hypJ H x) = −x

theorem **toMatrix_hypForm** (H : Host E) : LinearMap.BilinForm.toMatrix (hypRealBasis H) ((hypForm H).baseChange ℝ) = Matrix.fromBlocks 0 (-1) 1 0

theorem **hypForm_hypJ_hypJ** (H : Host E) (x y) : (hypForm H).baseChange ℝ (hypJ H x) (hypJ H y) = (hypForm H).baseChange ℝ x y

theorem **hypForm_hypJ_pos** (H : Host E) (x) (hx : x ≠ 0) : 0 < (hypForm H).baseChange ℝ (hypJ H x) x

theorem **trace_mulMatrix** (a : 𝒪 E) : (mulMatrix a).trace = (Algebra.trace ℤ (𝒪 E) a : ℤ)

theorem **gram_mul_mulMatrix** (H : Host E) (a : 𝒪 E) : gram H * mulMatrix a = (mulMatrix (H.bar a))ᵀ * gram H

theorem **toMatrix_of_smul** (H : Host E) (a : 𝒪 E) (A : ℝ ⊗[ℤ] HypLattice H →ₗ[ℝ] ℝ ⊗[ℤ] HypLattice H) (hA : ∀ x, A (1 ⊗ₜ x) = 1 ⊗ₜ (a • x)) : LinearMap.toMatrix (hypRealBasis H) (hypRealBasis H) A = Matrix.fromBlocks (mulMatrix a) 0 0 (mulMatrix (H.bar a))ᵀ

theorem **sum_add_conj_eq_trace** (H : Host E) (a : 𝒪 E) : ∑ τ ∈ Finset.univ.filter (fun τ : E →+* ℂ => τ (H.rt : E) = H.rt0), (τ (a : E) + starRingEnd ℂ (τ (a : E))) = (Algebra.trace ℤ (𝒪 E) a : ℤ)

theorem **hypJ_comm** : hypJ H ∘ₗ A = A ∘ₗ hypJ H

theorem **trace_of_smul** : LinearMap.trace ℝ _ A = 2 * (Algebra.trace ℤ (𝒪 E) a : ℤ)

theorem **trace_hypJ_comp** : LinearMap.trace ℝ _ (hypJ H ∘ₗ A) = 0

theorem **hypJ_inl** (H : Host E) (x : 𝒪 E) : hypJ H (1 ⊗ₜ (x, 0)) = 1 ⊗ₜ (0, traceDual H x)

theorem **hypJ_inr** (H : Host E) (x : 𝒪 E) : hypJ H (1 ⊗ₜ (0, traceDual H x)) = -(1 ⊗ₜ (x, 0))

theorem **hypDatum_La** (H : Host E) : (hypDatum H).La = (𝒪 E × ConjDual H)

@[simp] theorem **hypDatum_p** (H : Host E) : (hypDatum H).p = 1

@[simp] theorem **hypDatum_q** (H : Host E) : (hypDatum H).q = 1

theorem **hypDatum_EF** (H : Host E) (x y : 𝒪 E) (f g : ConjDual H) : (hypDatum H).EF (x, f) (y, g) = toDual H f y - toDual H g x

theorem **hypDatum_smul** (H : Host E) (a x : 𝒪 E) (f : ConjDual H) : (a • (x, f) : (hypDatum H).La) = (a * x, a • f)

theorem **hypDatum_EF_inl_inl** (H : Host E) (x y : 𝒪 E) : (hypDatum H).EF (x, 0) (y, 0) = 0

theorem **hypDatum_EF_inr_inr** (H : Host E) (f g : ConjDual H) : (hypDatum H).EF (0, f) (0, g) = 0

theorem **hypDatum_EF_inr_inl** (H : Host E) (f : ConjDual H) (y : 𝒪 E) : (hypDatum H).EF (0, f) (y, 0) = toDual H f y

Development.AbelianVarietyTheory.Definitions.ArchimedeanThetaValues.TrFormBaseChangeNondegenerate

theorem **trForm_apply** (H : Host E) (x y : 𝒪 E) : trForm H x y = Algebra.trace ℤ (𝒪 E) (H.bar x * y)

theorem **trace_bar** (H : Host E) (a : 𝒪 E) : Algebra.trace ℤ (𝒪 E) (H.bar a) = Algebra.trace ℤ (𝒪 E) a

theorem **trForm_apply'** (H : Host E) (x y : 𝒪 E) : trForm H x y = Algebra.trace ℤ (𝒪 E) (x * H.bar y)

theorem **trForm_symm** (H : Host E) (x y : 𝒪 E) : trForm H x y = trForm H y x

theorem **trForm_isSymm** (H : Host E) : (trForm H).IsSymm

theorem **trForm_adj** (H : Host E) (a y z : 𝒪 E) : trForm H (a • y) z = trForm H y (H.bar a • z)

theorem **trForm_mul_left** (H : Host E) (a y z : 𝒪 E) : trForm H (a * y) z = trForm H y (H.bar a * z)

theorem **trForm_adj_star** (H : Host E) (a y z : 𝒪 E) : letI

theorem **trForm_bar_bar** (H : Host E) (x y : 𝒪 E) : trForm H (H.bar x) (H.bar y) = trForm H x y

theorem **embedding_bar** (H : Host E) (τ : E →+* ℂ) (a : 𝒪 E) : τ (H.bar a : 𝒪 E) = conj (τ a)

theorem **coe_trace_eq_sum** (a : 𝒪 E) : (Algebra.trace ℤ (𝒪 E) a : ℂ) = ∑ τ : E →+* ℂ, τ a

theorem **trForm_eq_sum** (H : Host E) (x y : 𝒪 E) : (trForm H x y : ℂ) = ∑ τ : E →+* ℂ, conj (τ x) * τ y

theorem **trForm_self_eq_sum** (H : Host E) (y : 𝒪 E) : (trForm H y y : ℝ) = ∑ τ : E →+* ℂ, ‖τ y‖ ^ 2

theorem **trForm_pos** (H : Host E) {y : 𝒪 E} (hy : y ≠ 0) : 0 < trForm H y y

theorem **trForm_nonneg** (H : Host E) (y : 𝒪 E) : 0 ≤ trForm H y y

theorem **trForm_nondegenerate** (H : Host E) : (trForm H).Nondegenerate

theorem **trForm_injective** (H : Host E) : Function.Injective (trForm H)

theorem **trForm_one_one** (H : Host E) : trForm H 1 1 = Module.finrank ℚ E

@[simp] theorem **realEmb_tmul** (τ : E →+* ℂ) (r : ℝ) (a : 𝒪 E) : realEmb τ (r ⊗ₜ a) = r * τ a

theorem **trForm_baseChange_eq_sum** (H : Host E) (v w : ℝ ⊗[ℤ] 𝒪 E) : ((trForm H).baseChange ℝ v w : ℂ) = ∑ τ : E →+* ℂ, conj (realEmb τ v) * realEmb τ w

theorem **trForm_baseChange_self_eq_sum** (H : Host E) (v : ℝ ⊗[ℤ] 𝒪 E) : (trForm H).baseChange ℝ v v = ∑ τ : E →+* ℂ, ‖realEmb τ v‖ ^ 2

theorem **trForm_baseChange_symm** (H : Host E) (v w : ℝ ⊗[ℤ] 𝒪 E) : (trForm H).baseChange ℝ v w = (trForm H).baseChange ℝ w v

theorem **trForm_baseChange_adj** (H : Host E) (a : 𝒪 E) (v w : ℝ ⊗[ℤ] 𝒪 E) : (trForm H).baseChange ℝ ((1 ⊗ₜ a) * v) w = (trForm H).baseChange ℝ v ((1 ⊗ₜ H.bar a) * w)

theorem **trForm_baseChange_nondegenerate** (H : Host E) : ((trForm H).baseChange ℝ).Nondegenerate

theorem **trForm_baseChange_pos** (H : Host E) {v : ℝ ⊗[ℤ] 𝒪 E} (hv : v ≠ 0) : 0 < (trForm H).baseChange ℝ v v

theorem **trForm_baseChange_nonneg** (H : Host E) (v : ℝ ⊗[ℤ] 𝒪 E) : 0 ≤ (trForm H).baseChange ℝ v v

theorem **trForm_rt_rt** (H : Host E) : trForm H H.rt H.rt = H.d * Module.finrank ℚ E

theorem **trForm_one_rt** (H : Host E) : trForm H 1 H.rt = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AdelicDataRowMeasureScaleTrivialTuple

theorem **mem_ThS** {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {f : WA V W → ℂ} : f ∈ ThS ψ V W ↔ IsThetaFun ψ V W f

theorem **CuspS_le** (θ : InfF D) (V : HSp D) : CuspS θ V ≤ AFs θ V

theorem **const_mem_AFs** (θ : InfF D) (V : HSp D) (c : ℂ) : (fun _ => c) ∈ AFs θ V

theorem **CuspS_eq_of_aniso** (θ : InfF D) {V : HSp D} (h : IsAniso V) : CuspS θ V = AFs θ V

theorem **isTypeAt_smul** {θ : InfF D} {V : HSp D} {τ : D.what₂.Tau} {S : SPar} {φ : GA D V → ℂ} (c : ℂ) (h : IsTypeAt θ V τ S φ) : IsTypeAt θ V τ S (c • φ)

theorem **nonempty_orthDec** (h : OrthEx D) : Nonempty (OrthDec D)

theorem **OrthDec.exists_rel** (Dc : OrthDec D) {V : HSp D} (hV : V.SigOK) : ∃ s : GlobalTheory.IndefiniteRowSideFullHecke.FockOrthogonalDecomposition.InvFam V, ∀ τ, GlobalTheory.FockCompare.Rel V τ (Dc.cpt V τ) (s τ)

theorem **mem_KfinS** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfF D} (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (Dc : OrthDec D) {V W : HSp D} (hV : V.SigOK) (hW : W.SigOK) (Φ : ThS ψ V W) : Φ ∈ KfinS ψ θ Dc V W ↔ IsPolFock ψ θ V W (Dc.cpt V) (Dc.cpt W) Φ.1

theorem **rules** : Wf.core.Rules

theorem **ker_left** {V W : HSp D} {χ χ' : IdC D} (ha : admitted V W χ χ') (Φ : ThS ψ V W) {γ : GA D V} (hγ : γ ∈ GQ V) {δ : GA D W} (hδ : δ ∈ GQ W) (g h) : Wf.ker V W χ χ' Φ (γ * g) (δ * h) = Wf.ker V W χ χ' Φ g h

theorem **isHC_one** (D : D7) : IsHC D 1

theorem **isAdmissibleIntegralBasisChange_one** {V W : HSp D} (h : V.rank ≠ W.rank) (hV : Even V.rank) (hW : Even W.rank) : admitted V W 1 1

theorem **isWeilOn_rank_zero** (ψ : AddChar (AF D.host.F) ℂ) {V W : HSp D} (hV : 0 < V.rank) (hW : W.rank = 0) (χ : IdC D) : IsWeilOn ψ V W χ 1 1

theorem **X_posDef** (D : D7) : ∀ j, j ≤ D.R.m → (D.sp.X j).IsPosDef

theorem **exists_towerSpaces** (D : D7) (t : ℕ) : ∃ (X Y : ℕ → HSp D) (i : ∀ j, (X j).V →ₗ[D.host.E] (Y j).V) (i' : (Y D.R.m).V →ₗ[D.host.E] (X D.R.m).V), (∀ j, j ≤ D.R.m → (X j).rank = D.R.k + j * (t + 1)) ∧ (∀ j, j ≤ D.R.m → (X j).SigOK ∧ (X j).IsPosDef) ∧ (∀ j, j ≤ D.R.m → (Y j).rank = D.R.r + j * t) ∧ (∀ j, (Y j).SigOK) ∧ (∀ j, j ≤ D.R.m → ∀ τ, (Y j).sig τ = (D.R.r + j * t - (D.R.m - j) / 2, (D.R.m - j) / 2)) ∧ (∀ j, j < D.R.m → HSpace.IsIsomEmb (X j) (Y j) (i j)) ∧ HSpace.IsIsomEmb (Y D.R.m) (X D.R.m) i' ∧ (t = D.R.t → (∀ j, X j = D.sp.X j) ∧ (∀ j, Y j = D.sp.Y j) ∧ (∀ j, HEq (i j) (D.sp.iXY j)) ∧ HEq i' (D.sp.iX'.comp D.sp.eYm.symm.toLinearMap))

theorem **nonempty_tow** (h : WittLowEx D) : Nonempty (Tow D)

theorem **nonempty_realMeas** (D : D7) : Nonempty (RealMeas D)

theorem **memberBes_spec** (D : D7) (u : D.what₂.Word) : (memberBes D u).e = GlobalTheory.BesselCompare.eMat (D.sp.Y 0) D.sp.hypE ∧ (memberBes D u).z = GlobalTheory.BesselCompare.co (D.sp.Y 0) (D.zmOf u) ∧ (∀ i, (memberBes D u).c i = D.host.xi0) ∧ ((memberBes D u).y : GA D (D.sp.Y 0)) = 1

theorem **memberBes_norm** (D : D7) (u : D.what₂.Word) : Formalism.UG.hform (D.sp.cm.sig : D.host.E →+* D.host.E) (D.sp.Y 0).gram (memberBes D u).z (memberBes D u).z = -((D.mem u).A.T : D.host.E)

theorem **firstPosition_translate_one** (u) : ((Rc.pos1 u).y : GA D (D.sp.Y 0)) = 1

theorem **posSym_y** (u) : ((Rc.posSym u).y : GA D (D.sp.Y 0)) = 1

theorem **RowIsom.injective** (h : Rc.RowIsom) (μ) {j} (hj : j ≤ D.R.m) : Function.Injective (Rc.row μ j)

theorem **nonempty_recFun** (θ : InfF D) (ψ : AddChar (AF D.host.F) ℂ) (hχ : ChrEx D) (hη : EtaEx D) (hΦ : IsStdChar D ψ (-1) → ∀ V W : HSp D, ∃ f, IsThetaFun ψ V W f ∧ f ≠ 0) (Dc : OrthDec D) (P : Pos1 D) : Nonempty (RecFun D θ ψ)

theorem **mem_CuspS** (h : FormSub D) {θ : InfF D} (hθ : IsInfFam θ) {V : HSp D} {φ : GA D V → ℂ} : φ ∈ CuspS θ V ↔ IsCuspForm θ V φ

theorem **mem_TypS** (h : FormSub D) (ht : TypAdd D) {θ : InfF D} (hθ : IsInfFam θ) {V : HSp D} (hV : V.SigOK) {P : D.what₂.Tau → SPar} (hP : ∀ τ, GoodPar V.rank (V.sig (tauE D τ)).1 (V.sig (tauE D τ)).2 (P τ)) {φ : GA D V → ℂ} : φ ∈ TypS θ V P ↔ IsAutFunSm θ V φ ∧ ∀ τ, IsTypeAt θ V τ (P τ) φ

theorem **weilEx_of** (h : WeilCoreEx D) (hr : WeilRules D) : WeilEx D

theorem **eq_of_isInfFam** {θ' : InfF D} (h' : IsInfFam θ') : θ' = θ

theorem **afTiedSm** : ThetaSmoothnessCondition T

theorem **formSub** : FormSub D

theorem **typTied** (ht : TypAdd D) : TypTied T

theorem **weilUnrTied** : WeilUnramifiedCondition W

theorem **allTied** (ht : TypAdd D) {A : ProvA} (hr : RealTied D) (hp : ProvisionCompatibility D A) : Nonempty (AllTied D X A)

theorem **petC_eq** (V : HSp D) : T.petC V = Rc.petC V

theorem **tieRowIsom** (hi : Rc.RowIsom) : X.TieRowIsom

theorem **tieBm** (hi : Rc.TopTie) : X.TieBm

theorem **carY_eq** (μ s i) : T.eChr (X.carY μ s i) = Rc.carY μ s i

theorem **tieIo** (hi : Rc.IoTie Wf) : X.TieIo

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AdmissibleRepresentationFactorisation

theorem **stepLB_of_some** (b : Bool) {st : ((E →+* ℂ) × ℕ) ⊕ ℕ} {x : Letter Dw} (h : letterOf Dw st = some x) : z.stepLB b st = (z.line b st x h).L

theorem **isGenId_genId** : IsGenId (z.line false st x h) (z.line true st x h) (z.genId hp hW h)

theorem **log_norm_genId** (y : z.Z.Pt) (a : Fin z.Z.m) (φ : Γ(z.Z.X, (z.Z.U a).1) →+* ɪy.C.S) (hφ : spMap φ ≫ z.Z.chart a = y.gen) {i j} (hi : ⊤ ≤ y.z ⁻¹ᵁ (z.line false st x h).L.U i) (hj : ⊤ ≤ y.z ⁻¹ᵁ (z.line true st x h).L.U j) : Real.log ʌvalF y.gen _ ((z.genId hp hW h).u i j).valʌ = lnStep (Y z.Z.i).t (y.gen ≫ z.Z.π) ((z.Z.fr a).bc φ) p false st - lnStep (Y z.Z.i).t (y.gen ≫ z.Z.π) ((z.Z.fr a).bc φ) p true st

theorem **pair_x_eq_LB_false** (i : Fin r) : (z.stepPair hp hW d h).x i = z.stepLB false (st i)

theorem **pair_x_prime_eq_LB_true** (i : Fin r) : (z.stepPair hp hW d h).x' i = z.stepLB true (st i)

theorem **dualPair_x_eq_dual_LB_false** (i : Fin r) : (z.dualPair hp hW d h).x i = (z.stepLB false (st i)).dual

theorem **dualPair_x_prime_eq_dual_LB_true** (i : Fin r) : (z.dualPair hp hW d h).x' i = (z.stepLB true (st i)).dual

theorem **dualPair_Dv** (i : Fin r) : (z.dualPair hp hW d h).Dv i = -vdiv hp hW d _ _ (z.genId hp hW (h i))

theorem **comp_cEmb** (τ : DV.Emb D p) : jc.comp (DV.cEmb D τ) = conjEmb (jc.comp τ)

theorem **mult_comp** (τ : DV.Emb D p) : S.Dw.mult (jc.comp τ) = DV.rkOf (DV.PsiOf D jc) N j τ

theorem **mult_conj_comp** (τ : DV.Emb D p) : S.Dw.mult (conjEmb (jc.comp τ)) = N - DV.rkOf (DV.PsiOf D jc) N j τ

theorem **length_letters** : (letters S.Dw).length = D.R.f * N

theorem **letterOf_lie** (τ : DV.Emb D p) (i : Fin (DV.rkOf (DV.PsiOf D jc) N j τ)) : letterOf S.Dw (.inl (jc.comp τ, i)) = some (jc.comp τ, i.cast (mult_comp S jc τ).symm)

theorem **letterOf_om** (τ : DV.Emb D p) (i : Fin (N - DV.rkOf (DV.PsiOf D jc) N j τ)) : letterOf S.Dw (.inl (conjEmb (jc.comp τ), i.rev)) = some (conjEmb (jc.comp τ), i.rev.cast (mult_conj_comp S jc τ).symm)

theorem **letterOf_whole** (k : Fin (D.R.f * N)) : letterOf S.Dw (.inr k) = some ((letters S.Dw).get (k.cast (length_letters S).symm))

theorem **mdlOf_lie** (τ : DV.Emb D p) (i) : ((mdlOf z jc).lie τ).x i = z.stepLB false (.inl (jc.comp τ, i)) ∧ ((mdlOf z jc).lie τ).x' i = z.stepLB true (.inl (jc.comp τ, i))

theorem **mdlOf_om** (τ : DV.Emb D p) (i) : ((mdlOf z jc).om τ).x i = (z.stepLB false (.inl (conjEmb (jc.comp τ), i.rev))).dual ∧ ((mdlOf z jc).om τ).x' i = (z.stepLB true (.inl (conjEmb (jc.comp τ), i.rev))).dual

theorem **mdlOf_lieW** (k) : (mdlOf z jc).lieW.x k = z.stepLB false (.inr k) ∧ (mdlOf z jc).lieW.x' k = z.stepLB true (.inr k)

theorem **mdlOf_omW** (k) : (mdlOf z jc).omW.x k = (z.stepLB false (.inr k.rev)).dual ∧ (mdlOf z jc).omW.x' k = (z.stepLB true (.inr k.rev)).dual

theorem **mdlOf_lie_Dv** (τ : DV.Emb D p) (i) : ((mdlOf z jc).lie τ).Dv i = vdiv hp hW _ _ _ (z.genId hp hW (letterOf_lie S jc τ i))

theorem **mdlOf_om_Dv** (τ : DV.Emb D p) (i) : ((mdlOf z jc).om τ).Dv i = -vdiv hp hW _ _ _ (z.genId hp hW (letterOf_om S jc τ i))

theorem **mdlOf_omW_Dv** (k) : (mdlOf z jc).omW.Dv k = -(mdlOf z jc).lieW.Dv k.rev

theorem **rootOf_lie** (i : Fin (DV.rkOf (DV.PsiOf D jc) N j τ)) (i' : Fin N) (h : (i' : ℕ) = i) : rootOf z jc (false, (t, i')) = ((mdlOf z jc).lie τ).x i ∧ rootOf z jc (true, (t, i')) = ((mdlOf z jc).lie τ).x' i

theorem **rootOf_om** (i : Fin (N - DV.rkOf (DV.PsiOf D jc) N j τ)) (i' : Fin N) (h : (i' : ℕ) = j + i) : rootOf z jc (false, (t, i')) = ((mdlOf z jc).om τ).x i ∧ rootOf z jc (true, (t, i')) = ((mdlOf z jc).om τ).x' i

theorem **one_add_lt_three** {p : ℕ} (hp : p.Prime) : 1 + 1 / ((p : ℝ) - 1) < (3 : ℕ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AdmissibleStageOfPackage

theorem **sp_comp** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **sp_epsMul_one** (R : Type) [CommRing R] : spMap (epsMul (1 : R)) = 𝟙 _

theorem **latDR_subset_latHT** {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} (t : SAB H X) {C : Type} [NontriviallyNormedField C] [IsUltrametricDist C] (p : ℕ) (x : Sp C ⟶ X) : latDR t x ⊆ latHT t p x

theorem **torusChar_epsVec** {O : Type} [CommRing O] (x : Sp O ⟶ X) (q : ℕ) (a : 𝒪 E) (I : Ideal O) (c : 𝒪 E →+ O ⧸ I) : (((torusChar H x q a).χ (DualNumber (O ⧸ I)) (GlobalTheory.Diag.epsVec _ c) : (DualNumber (O ⧸ I))ˣ) : DualNumber (O ⧸ I)).snd = c a

theorem **dual_base** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) : spMap (dualMap f) ≫ spMap (algebraMap A (DualNumber A)) = spMap (algebraMap B (DualNumber B)) ≫ spMap f

theorem **latHT_torus** {C : Type} [NontriviallyNormedField C] [IsUltrametricDist C] (p : ℕ) (hp : (p : C) ≠ 0) (x : Sp C ⟶ X) : latHT (torus H X) p x = latDR (torus H X) x

theorem **exists_flagPt** {O : Type} [CommRing O] {β : O →+* c.F} (Z : Model Xo c β) (Ω : Type) [Field Ω] (y : Sp Ω ⟶ gen Z) : ∃ z : FlagPt Z Ω, z.y = y

theorem **latDR_nonempty** (Z : Model Xo c (algebraMap (𝒪 c.F) c.F)) {C : Type} [NontriviallyNormedField C] [IsUltrametricDist C] (x : Sp C ⟶ Z.X) : (latDR Z.t x).Nonempty

theorem **polar_sq** (a b : ℝ) : polar (fun x : ℝ => x ^ 2) {a, b} = a * b

theorem **polar_cube** (a b c : ℝ) : polar (fun x : ℝ => x ^ 3) {a, b, c} = a * b * c

theorem **numL_pow** {A : Type} (num : Multiset A → ℝ) (d : ℕ) (a : A) : numL num d (MvPolynomial.X a ^ d) = num (Multiset.replicate d a)

theorem **arReal_deg** {Tau : Type} [Fintype Tau] {N g : ℕ} (x : Tau × Fin N) (S : Finset ℕ) (R : ℝ) : (arReal Tau N g).arDeg S R (MvPolynomial.X x ^ (Formalism.DegreeFormula.dimFl Tau N + 1)) = (if 2 ∈ S then R else 1) ^ (Formalism.DegreeFormula.dimFl Tau N + 1)

theorem **nonempty_flagPt** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {O : Type} [CommRing O] {β : O →+* c.F} (Z : Model Xo c β) [Nonempty (gen Z)] : Nonempty (FlagPt Z Qb)

theorem **grNorm_le_of_smul** {X : Scheme.{0}} {t : SAB H X} {C : Type} [NontriviallyNormedField C] {x : Sp C ⟶ X} {Λ Λ' : Set (Sp (DualNumber C) ⟶ t.G.left)} {m : ℕ} {u : Fin m → (Sp (DualNumber C) ⟶ t.G.left)} (hu : ∀ k, IsTanAtG t x (u k)) {a : C} (ha : a ≠ 0) (h : ∀ w ∈ Λ, spMap (epsMul a) ≫ w ∈ Λ') (i : Fin m) (hne : (Guard.grSet t x Λ u i).Nonempty) : ‖a‖ * grNorm t x Λ' u i ≤ grNorm t x Λ u i

theorem **lnLie_sub** (hB : HTBoundSt) {Z : Model Xo c (algebraMap (𝒪 c.F) c.F)} {Ω : Type} [Field Ω] (y : FlagPt Z Ω) (p : ℕ) [hp : Fact p.Prime] (io : Ω →+* ℂ_[p]) (τ : E →+* ℂ) (k : ℕ) : 0 ≤ y.lnLie io p false τ k - y.lnLie io p true τ k ∧ y.lnLie io p false τ k - y.lnLie io p true τ k ≤ 1 / ((p : ℝ) - 1) * Real.log p

theorem **phi_bounds** (hB : HTBoundSt) (p : ℕ) [Fact p.Prime] (y : Pt s) (x : DV.Rt D.what₂.Tau N) : ((x.2 : ℕ) < j → vDiff D.host.E p * Real.log p ≤ phi s p y x ∧ phi s p y x ≤ (vDiff D.host.E p + 1 / ((p : ℝ) - 1)) * Real.log p) ∧ (j ≤ (x.2 : ℕ) → -(1 / ((p : ℝ) - 1) * Real.log p) ≤ phi s p y x ∧ phi s p y x ≤ 0)

theorem **root_in_range** (x : DV.Rt D.what₂.Tau N) : ((x.2 : ℕ) < S.Dw.p → (x.2 : ℕ) < S.Dw.mult (DV.tauEmb D x.1)) ∧ (¬ (x.2 : ℕ) < S.Dw.p → S.Dw.p + S.Dw.q - 1 - x.2 < S.Dw.mult (conjEmb (DV.tauEmb D x.1)))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AevalDualModel

theorem **_root_.GlobalTheory.GoodMetric.PolyRep.lie_proj_model** [Fintype ι] (τ : ι) {R : Type} [CommRing R] (A : _root_.GlobalTheory.FlagBundle.Mat m R) : (_root_.GlobalTheory.GoodMetric.PolyRep.proj m τ).lie A = A τ

theorem **aeval_dual_model** {σ : Type} [Fintype σ] {R : Type} [CommRing R] (x a : σ → R) (p : MvPolynomial σ ℤ) : aeval (fun v => (TrivSqZeroExt.inl (x v) + TrivSqZeroExt.inr (a v) : DualNumber R)) p = TrivSqZeroExt.inl (aeval x p) + TrivSqZeroExt.inr (∑ v, aeval x (pderiv v p) * a v)

private lemma **aeval_dualNumber_inl_add_inr** {σ : Type} [Fintype σ] {R : Type} [CommRing R] (x y : σ → R) (p : MvPolynomial σ ℤ) : aeval (fun v => (TrivSqZeroExt.inl (x v) + TrivSqZeroExt.inr (y v) : DualNumber R)) p = TrivSqZeroExt.inl (aeval x p) + TrivSqZeroExt.inr (∑ v, aeval x (pderiv v p) * y v)

theorem **_root_.GlobalTheory.GoodMetric.PolyRep.ev_dual_model** [Fintype ι] (ρ : _root_.GlobalTheory.GoodMetric.PolyRep m) {R : Type} [CommRing R] (A : _root_.GlobalTheory.FlagBundle.Mat m R) (i j : Fin ρ.M) : ρ.ev (R

theorem **ev_detModel** (τ : ι) {R : Type} [CommRing R] (A : Mat m R) (i j : Fin 1) : (detModel m τ).ev A i j = (A τ).det

theorem **lie_detModel** [Fintype ι] (τ : ι) {R : Type} [CommRing R] (A : Mat m R) (i : Fin 1) : (detModel m τ).lie A i i = (A τ).trace

theorem **lie_transp_model** [Fintype ι] (ρ : PolyRep m) {R : Type} [CommRing R] (A : Mat m R) : ρ.lie (fun τ => (A τ)ᵀ) = (ρ.lie A)ᵀ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AevalFirstChernSurjective

theorem **cls_ofCharts** (L : LB X) (hL : L.IsLB) {n : ℕ} (f : Fin n → L.ι) (hf : ⊔ i, L.U (f i) = ⊤) : (ofCharts L hL f hf).cls = CechPic.cls L hL

theorem **cls_ofFintype** (L : LB X) (hL : L.IsLB) [Fintype L.ι] : (ofFintype L hL).cls = CechPic.cls L hL

theorem **exists_charts** [CompactSpace X] {L : LB X} (hL : L.IsLB) : ∃ (n : ℕ) (f : Fin n → L.ι), ⊔ i, L.U (f i) = ⊤

theorem **cls_surjective** [CompactSpace X] : Function.Surjective (cls (X

theorem **cls_eq_iff** {L M : FinLB X} : L.cls = M.cls ↔ LBIso L.lb M.lb

theorem **lb_dual** (L : FinLB X) : L.dual.lb = CechPic.dual L.lb

@[simp] theorem **cls_dual** (L : FinLB X) : L.dual.cls = -L.cls

@[simp] theorem **cls_triv** : (triv X).cls = 0

@[simp] theorem **cls_tens** (L M : FinLB X) : (L.tens M).cls = L.cls + M.cls

@[simp] theorem **cls_pull** (L : FinLB X) (f : Y ⟶ X) : (L.pull f).cls = CechPic.pull f L.cls

theorem **c1_eq** (L : FinLB X) : L.c1 = firstChernPic L.cls

@[simp] theorem **firstChern_dual** (L : FinLB X) : L.dual.c1 = -L.c1

@[simp] theorem **firstChern_trivial** : (triv X).c1 = 0

@[simp] theorem **firstChernTensor** (L M : FinLB X) : (L.tens M).c1 = L.c1 + M.c1

theorem **firstChernCongr** {L M : FinLB X} (h : LBIso L.lb M.lb) : L.c1 = M.c1

@[simp] theorem **pullback_firstChern** (L : FinLB X) (f : Y ⟶ X) : picSymPull f L.c1 = (L.pull f).c1

theorem **existsUnique_lift** [CompactSpace X] {A : Type} [CommRing A] [Algebra ℚ A] (φ : FinLB X → A) (hiso : ∀ L M : FinLB X, LBIso L.lb M.lb → φ L = φ M) (hadd : ∀ L M, φ (L.tens M) = φ L + φ M) : ∃! ψ : PicSym X →ₐ[ℚ] A, ∀ L : FinLB X, ψ L.c1 = φ L

theorem **aeval_firstChern_surjective** [CompactSpace X] : Function.Surjective (MvPolynomial.aeval (R

theorem **exists_mul_of_opens** {X : Scheme.{0}} (U : X.Opens) {G V : X.Opens} {W₀ T : U.toScheme.Opens} (hT : T ≤ W₀) (hW : W₀ ≤ U.ι ⁻¹ᵁ V) (e : T ≤ U.ι ⁻¹ᵁ (G ⊓ V)) (hG : U.ι ⁻¹ᵁ G ⊓ W₀ ≤ T) {a c : Γ(X, G ⊓ V)} {b : Γ(U.toScheme, W₀)} (h : (U.ι.appLE _ _ e).hom a = resG _ hT b * (U.ι.appLE _ _ e).hom c) {x : U.toScheme} (hx : x ∈ W₀) : ∃ (W : X.Opens) (hW : W ≤ V) (b' : Γ(X, W)), U.ι x ∈ W ∧ resG X (inf_le_inf_left G hW) a = resG X inf_le_right b' * resG X (inf_le_inf_left G hW) c

theorem **isUnit_eqn** (i : P.M.ι) {W : Z.X.Opens} (h : W ≤ Z.gen ⊓ P.M.U i) : IsUnit (P.eqn i h)

@[simp] theorem **resG_eqn** (i : P.M.ι) {W W' : Z.X.Opens} (h : W ≤ Z.gen ⊓ P.M.U i) (h' : W' ≤ W) : resG Z.X h' (P.eqn i h) = P.eqn i (h'.trans h)

theorem **eqn_rel** (i j : P.M.ι) {W : Z.X.Opens} (hi : W ≤ Z.gen ⊓ P.M.U i) (hj : W ≤ Z.gen ⊓ P.M.U j) (hij : W ≤ P.M.U i ⊓ P.M.U j) : P.eqn i hi = resG Z.X hij (P.M.g i j : Γ(Z.X, P.M.U i ⊓ P.M.U j)) * P.eqn j hj

theorem **eqn_mul_rel** (i j : P.M.ι) (i' j' : P'.M.ι) {W V : Z.X.Opens} (hi : W ≤ Z.gen ⊓ P.M.U i) (hj : W ≤ Z.gen ⊓ P.M.U j) (hi' : W ≤ Z.gen ⊓ P'.M.U i') (hj' : W ≤ Z.gen ⊓ P'.M.U j') (h : W ≤ V) (hij : V ≤ P.M.U i ⊓ P.M.U j) (hij' : V ≤ P'.M.U i' ⊓ P'.M.U j') : P.eqn i hi * P'.eqn i' hi' = resG Z.X h (res hij (P.M.g i j) * res hij' (P'.M.g i' j')).val * (P.eqn j hj * P'.eqn j' hj')

theorem **isSum_add** : IsSum P P' (P.add P')

theorem **isSum_neg** : IsSum P P.neg (zero Z)

theorem **IsIsom.refl** : IsIsom P P

theorem **IsIsom.isSum** (H : IsIsom P P') : IsSum P (zero Z) P'

theorem **isSum_ofCharts** {n : ℕ} (f : Fin n → P.M.ι) (hf : ⊔ i, P.M.U (f i) = ⊤) : IsSum P (zero Z) (P.ofCharts f hf)

theorem **compι_str** (C : Z.Comp) : Z.compι C ≫ Z.str = (Z.comp C).str

theorem **exists_comp** (x : Z.X) : ∃ C : Z.Comp, x ∈ C.opens

theorem **mem_basicOpen_const** {x : Z.X} {g : O} : x ∈ Z.X.basicOpen (Z.const g) ↔ g ∉ (Z.str x).asIdeal

theorem **mem_gen_iff_eq_bot** {x : Z.X} : x ∈ Z.gen ↔ (Z.str x).asIdeal = ⊥

theorem **gen_le_basicOpen** {g : O} (hg : g ≠ 0) : Z.gen ≤ Z.X.basicOpen (Z.const g)

theorem **gen_eq_basicOpen** {g : O} (hg : g ≠ 0) (hm : g ∈ maximalIdeal O) : Z.gen = Z.X.basicOpen (Z.const g)

theorem **isUnit_const** {g : O} (hg : g ≠ 0) {V : Z.X.Opens} (h : V ≤ Z.gen) : IsUnit (resG Z.X (le_top : V ≤ ⊤) (Z.const g))

theorem **exists_specializes** (x : Z.X) : ∃ y ∈ Z.gen, y ⤳ x

theorem **resG_injective** {V W : Z.X.Opens} (h : W ≤ V) (hW : Z.gen ⊓ V ≤ W) : Function.Injective (resG Z.X h)

theorem **exists_extension** {V : Z.X.Opens} (a : Γ(Z.X, Z.gen ⊓ V)) (h : ∀ x ∈ V, ∃ (W : Z.X.Opens) (hW : W ≤ V) (b : Γ(Z.X, W)), x ∈ W ∧ resG Z.X inf_le_right b = resG Z.X (inf_le_inf_left Z.gen hW) a) : ∃ b : Γ(Z.X, V), resG Z.X inf_le_right b = a

theorem **preimage_gen** : f ⁻¹ᵁ Z.gen = B.gen

theorem **appTop_const** (g : O) : f.appTop (Z.const g) = B.const g

theorem **eqn_pull** (j : (P.pull hf).M.ι) {W : B.X.Opens} (h : W ≤ B.gen ⊓ (P.pull hf).M.U j) {V : Z.X.Opens} (hV : V ≤ Z.gen ⊓ P.M.U (j ⟨⟩)) (e : W ≤ f ⁻¹ᵁ V) : (P.pull hf).eqn j h = (f.appLE V W e).hom (P.eqn (j ⟨⟩) hV)

theorem **IsSum.pull** (H : IsSum P P' P'') : VertDiv.IsSum (P.pull hf) (P'.pull hf) (P''.pull hf)

@[simp] theorem **cls_pull_zero** : ((zero Z).pull hf).cls = 0

@[simp] theorem **cls_pull_const** (g : O) (hg : g ≠ 0) : ((const Z g hg).pull hf).cls = B.div g hg

theorem **isIsom_of_units** (u : ∀ (p : P'.M.ι) (q : P.M.ι), Γ(Z.X, P'.M.U p ⊓ P.M.U q)ˣ) (hu : ∀ p q, P'.eqn p (inf_le_inf_left _ inf_le_left : Z.gen ⊓ (P'.M.U p ⊓ P.M.U q) ≤ _) = resG Z.X inf_le_right (u p q).val * P.eqn q (inf_le_inf_left _ inf_le_right)) :

IsIsom P P'

theorem **exists_ratio** (h : ∀ C, ((P.pull (Z.compι_str C))).IsIsom (P'.pull (Z.compι_str C))) (p : P'.M.ι) (q : P.M.ι) : ∃ b : Γ(Z.X, P'.M.U p ⊓ P.M.U q), P'.eqn p (inf_le_inf_left _ inf_le_left : Z.gen ⊓ (P'.M.U p ⊓ P.M.U q) ≤ _) = resG Z.X inf_le_right b ∗ P.eqn q (inf_le_inf_left _ inf_le_right)

theorem **isIsom_of_comp** (h : ∀ C, (P.pull (Z.compι_str C)).cls = (P'.pull (Z.compι_str C)).cls) : IsIsom P P'

theorem **toProd_eq_iff** : P.toProd = P'.toProd ↔ IsIsom P P'

theorem **toProd_add** : (P.add P').toProd = P.toProd + P'.toProd

theorem **toProd_zero** : (zero Z).toProd = 0

theorem **toProd_neg** : P.neg.toProd = -P.toProd

theorem **cls_surjective** : Function.Surjective (cls (Z

theorem **cls_eq_iff** : P.cls = P'.cls ↔ IsIsom P P'

@[simp] theorem **cls_add** : (P.add P').cls = P.cls + P'.cls

@[simp] theorem **cls_zero** : (zero Z).cls = 0

@[simp] theorem **cls_neg** : P.neg.cls = -P.cls

theorem **isSum_iff** : IsSum P P' P'' ↔ P''.cls = P.cls + P'.cls

theorem **exists_finLB** (D : Z.VDivZ) : ∃ (M : FinLB Z.X) (t : Γ(toMod M.lb, Z.gen)) (h : ∀ i, IsUnit (coord t i)), cls (M.lb, M.isLB, t, h) = D

@[simp] theorem **vLift_cls** (P : VPair Z) : vLift φ h P.cls = φ P

@[simp] theorem **vCls_cls** (P : VPair Z) : Z.vCls P.cls = cls P.M P.isLB

theorem **vRes_cls** (P : VPair Z) (C : Z.Comp) : Z.vRes C P.cls = (P.pull (Z.compι_str C)).cls

theorem **vDivZ_ext** {D D' : Z.VDivZ} (h : ∀ C, Z.vRes C D = Z.vRes C D') : D = D'

theorem **vCls_vRes** (D : Z.VDivZ) (C : Z.Comp) : (Z.comp C).vCls (Z.vRes C D) = pull (Z.compι C) (Z.vCls D)

@[simp] theorem **vRes_div** (g : O) (hg : g ≠ 0) (C : Z.Comp) : Z.vRes C (Z.div g hg) = (Z.comp C).div g hg

@[simp] theorem **vCls_div** (g : O) (hg : g ≠ 0) : Z.vCls (Z.div g hg) = 0

theorem **div_eq_addVal_smul** {ϖ g : O} (hϖ : Irreducible ϖ) (hg : g ≠ 0) : Z.div g hg = (IsDiscreteValuationRing.addVal O g).toNat • Z.div ϖ hϖ.ne_zero

@[simp] theorem **vPull_cls** {B : Model O} {f : B.X ⟶ Z.X} (hf : f ≫ Z.str = B.str) (P : VPair Z) : vPull hf P.cls = (P.pull hf).cls

theorem **picSymPull_mem_symDeg** {X Y : Scheme.{0}} (f : Y ⟶ X) {n : ℕ} {x : PicSym X} (hx : x ∈ symDeg (SmallPic X) n) : picSymPull f x ∈ symDeg (SmallPic Y) n

theorem **vDeg_apply** (d : ℕ) (D : Z.VDivZ) (x : PicSym Z.X) : Z.vDeg d D x = ∑ C ∈ Finset.univ.filter fun C : Z.Comp => (Z.comp C).relDim = d, (Z.comp C).vDeg (Z.vRes C D) (picSymPull (Z.compι C) x)

theorem **vDeg_of_mem** {d : ℕ} (D : Z.VDivZ) {x : PicSym Z.X} (hx : x ∈ symDeg (SmallPic Z.X) d) : Z.vDeg d D x = ∑ C, (Z.comp C).vDeg (Z.vRes C D) (picSymPull (Z.compι C) x)

theorem **vDeg_eq_zero** {d n : ℕ} (D : Z.VDivZ) (hn : n ≠ d) {x : PicSym Z.X} (hx : x ∈ symDeg (SmallPic Z.X) n) : Z.vDeg d D x = 0

theorem **vDeg_of_relDim** {d : ℕ} (h : ∀ C, (Z.comp C).relDim = d) (D : Z.VDivZ) (x : PicSym Z.X) : Z.vDeg d D x = ∑ C, (Z.comp C).vDeg (Z.vRes C D) (picSymPull (Z.compι C) x)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AlgebraMapGFEqGval

theorem **coMat_colVec** {a : Type} (A : Matrix (Fin S.rank) a E) : coMat S (colVec S A) = A

theorem **colVec_coMat** {a : Type} (x : a → S.V) : colVec S (coMat S x) = x

theorem **frame_lflagOf** (h : IsFrame (starRingEnd E) S.gram T A) : Formalism.MainNotions.S.Lat.frame (lflagOf h) = colVec S A

theorem **coMat_lflagOf** (h : IsFrame (starRingEnd E) S.gram T A) : coMat S (Formalism.MainNotions.S.Lat.frame (lflagOf h)) = A

theorem **zhatL_eq** (F : LFlag S r T) (F' : LFlag S r T') : Formalism.MainNotions.S.Lat.ZhatL F F' = pairM (starRingEnd E) S.gram (coMat S (Formalism.MainNotions.S.Lat.frame F')) (coMat S (Formalism.MainNotions.S.Lat.frame F))

theorem **gvalL_eq** (F : LFlag S r T) (F' : LFlag S r T') : Formalism.MainNotions.S.Lat.Gval F F' = Gq (starRingEnd E) T T' (Formalism.MainNotions.S.Lat.ZhatL F F')

theorem **liesAt_iff_isInCell** (w : Formalism.MainNotions.S.Point E (r + 1)) (Z : Matrix (Fin (r + 1)) (Fin (r + 1)) E) : LiesAt w Z ↔ IsInCell (ZhatOM Z) w

theorem **fixedFlag_e** (D : D7) (u : D.what₂.Word) : (fixedFlag D u).fl.e = D.sp.hypE

theorem **fixedFlag_z_eq_zmOf** (D : D7) (u : D.what₂.Word) : (fixedFlag D u).fl.z = D.zmOf u

theorem **fixedFlag_eq_lf** (D : D7) (u : D.what₂.Word) : fixedFlag D u = Formalism.MainNotions.S.lf D (fixedFlag D u).fl

theorem **coMat_fl** : coMat (hS D) (Formalism.MainNotions.S.Lat.frame t.fl) = recFrame D u

theorem **coMat_fl'** : coMat (hS D) (Formalism.MainNotions.S.Lat.frame t.fl') = t.fr

theorem **frame_fl_mem** (a : Fin (D.R.rS + 1)) : Formalism.MainNotions.S.Lat.frame t.fl a ∈ t.abelian₂

theorem **frame_fl'_mem** (a : Fin (D.R.rS + 1)) : Formalism.MainNotions.S.Lat.frame t.fl' a ∈ t.abelian₂

theorem **isLift_fl** : Formalism.MainNotions.S.Lat.IsLift t.abelian₂ t.fl

theorem **isLift_fl'** : Formalism.MainNotions.S.Lat.IsLift t.abelian₂ t.fl'

theorem **col_eq_of_Lam** {u'' : D.what₂.Word} (t' : RecTerm D u u'') (h : t.abelian₂ = t'.abelian₂) : t.col = t'.col

theorem **zhatL_fl** : Formalism.MainNotions.S.Lat.ZhatL t.fl t.fl' = t.Zhat

theorem **gval_fl** : Formalism.MainNotions.S.Lat.Gval t.fl t.fl' = t.G

theorem **algebraMap_GF_eq_gval** : algebraMap D.host.F D.host.E (t.GF D.host.F) = Formalism.MainNotions.S.Lat.Gval t.fl t.fl'

theorem **at_iff_isInCell** (w : Formalism.MainNotions.S.Point D.host.E (D.R.rS + 1)) : t.At w ↔ IsInCell (ZhatOM (Formalism.MainNotions.S.Lat.ZhatL t.fl t.fl')) w

theorem **isInCell_ptOf** (hG : t.GF D.host.F ≠ 0) (hR : t.relevant) : IsInCell (ZhatOM (Formalism.MainNotions.S.Lat.ZhatL t.fl t.fl')) t.ptOf

theorem **gpt_eq** (T T' : ℕ) (w : Formalism.MainNotions.S.Point D.host.E (D.R.rS + 1)) : Formalism.MainNotions.S.LocalSplitGroups.Gpt T T' w = Gq (starRingEnd D.host.E) (T : D.host.E) (T' : D.host.E) (ptMat w)

theorem **gpt_of_at** {w : Formalism.MainNotions.S.Point D.host.E (D.R.rS + 1)} (h : t.At w) : Formalism.MainNotions.S.LocalSplitGroups.Gpt (D.mem u).A.T (D.mem u').A.T w = t.G

theorem **gpt_ptOf_ne_zero_iff** : Formalism.MainNotions.S.LocalSplitGroups.Gpt (D.mem u).A.T (D.mem u').A.T t.ptOf ≠ 0 ↔ t.GF D.host.F ≠ 0 ∧ t.At t.ptOf

theorem **gpt_ptOf_of_not_at** (h : ¬ ∃ w, t.At w) : Formalism.MainNotions.S.LocalSplitGroups.Gpt (D.mem u).A.T (D.mem u').A.T t.ptOf = 0

theorem **gpt_ptOf_ne_zero** (hG : t.GF D.host.F ≠ 0) (hR : t.relevant) : Formalism.MainNotions.S.LocalSplitGroups.Gpt (D.mem u).A.T (D.mem u').A.T t.ptOf ≠ 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AmbientBaseHeq

theorem **ColLay.dGamDef** {lev : LevelData D} {eX : ErrX D} (h : ColLay D lev eX) : DGamDef D

theorem **ColLay.ofLay** {s : Scale} {C : FC} {cD : ℝ} (Y : Lay D) (hR : RecOK s C cD D) : ColLay D Y.lev Y.errX

theorem **ColLay.iff_lay** (Y : Lay D) : ColLay D Y.lev Y.errX ↔ DGamChoice_at D ∧ ∀ u (w : D.what₂.places.Pl), D.what₂.places.resChar w = (D.mem u).A.T → ∃ n : ℕ, n ≠ 0 ∧ (∀ x ∈ DV.MramL D u, D.nuIn u (n • x) ∈ LT.Nlam D u w) ∧ ∀ x, D.nuIn u x ∈ LT.Nlam D u w → InLoc (D.plEq w) (DV.MramL D u) (n • x)

theorem **GyLay.ofLay** {K : Ctx} (Y : Lay D) (hL : LedOK K.s K.C K.cD K.ch Y K.a K.b) : GyLay K D Y.gatT Y.cmp Y.Gy

theorem **GatNum.epTie** {L : SizeData₂ D} {X : G.Inst D L} (h : GatNum L.sc D) (he : G.EpDef X 1) : EpTie X

theorem **GatNum.ofTied** {L : SizeData₂ D} {X : G.Inst D L} {Gt : GatTies D} {Gy : G.LatInst D} (hl : G.LatTied Gt X Gy) (ht : EpTie X) (he : G.EpDef X 1) : GatNum L.sc D

theorem **GatNum.ofPt** {C : FC} (hC : C.cX.Ce = 1) (p : Pt C) : GatNum p.L.sc p.D

theorem **Ctx.Ce_eq_one** (K : Ctx) : K.C.cX.Ce = 1

theorem **GatNum.ofLay** {K : Ctx} (Y : Lay D) (hL : LedOK K.s K.C K.cD K.ch Y K.a K.b) {L : SizeData₂ D} (hs : L.sc = K.s) (hW : WtEq L K.a K.b) (hV : Val K.C K.cD K.ch L) {X : G.Inst D L} (hX : CanonInst X) (he : G.EpDef X K.C.cX.Ce) (hk : G.KapIDef X K.C.GK) (hf : ∀ w, G.ParametersFill X K.C.GK w) : GatNum K.s D

theorem **ExportAt.exists_good** {nch : NChoice D} {xd : ThData D} (h : ExportAt D nch xd) (hW : ∀ w, D.what₂.muAt w ≠ D.what₂.mu0 → D.ntw.NonzeroIndexPlace (nch.twI w)) (A : D.ntw.Tw → Finset (PlE D.host)) : ∃ (N0 : D.ntw.Tw → ℕ) (S : ∀ (i : D.ntw.Tw) (j : ℕ) (b : Bool), Finset (D.nexp.Dat i j b)) (Q : D.ntw.Tw → Finset (PlE D.host)) (k : D.ntw.Tw → ℕ) (ψ : ∀ i, D.ntw.RCG i (Q i) →* (D.ntw.Ok i (k i))ˣ), ∀ w, D.what₂.muAt w ≠ D.what₂.mu0 → TwIn.Ok ((twEOf xd, N0, S) : TwIn D) (nch.twI w) ∧ ∀ a : ℤ, IsCoprime a (D.ntw.l (nch.twI w) : ℤ) → IntegralHeckeForms.GoodTw D.nexp (BlochKatoReduction.bk0 D) (twEOf xd) N0 (nch.twI w) (twEps D (nch.twI w)) (S (nch.twI w)) (A (nch.twI w)) (Q (nch.twI w)) (k (nch.twI w)) (ψ (nch.twI w) ^ a)

theorem **ExportAt.congr** {nch nch' : NChoice D} {xd : ThData D} (e : nch'.twI = nch.twI) (h : ExportAt D nch xd) : ExportAt D nch' xd

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AmpleOnAbelianBase

theorem **numL_X_pow** (num : Multiset A → ℝ) (d : ℕ) (a : A) : numL num d (MvPolynomial.X a ^ d) = num (Multiset.replicate d a)

theorem **numL_homog** (num : Multiset A → ℝ) (d e : ℕ) (he : e ≠ d) (P : MvPolynomial A ℝ) (hP : P.IsHomogeneous e) : numL num d P = 0

theorem **realToy_arDeg** {Tau : Type} [Fintype Tau] {N g : ℕ} (x : Tau × Fin N) (S : Finset ℕ) (R : ℝ) : (realExample Tau N g).arDeg S R (MvPolynomial.X x ^ (Formalism.DegreeFormula.dimFl Tau N + 1)) = (if 2 ∈ S then R else 1) ^ (Formalism.DegreeFormula.dimFl Tau N + 1)

theorem **arDeg_homog** (S : Finset ℕ) (R : ℝ) (P : MvPolynomial (DV.Rt Tau N) ℝ) (d : ℕ) (hd : d ≠ Formalism.DegreeFormula.dimFl Tau N + 1) (hP : P.IsHomogeneous d) : I.arDeg S R P = 0

theorem **arMix_HT** (S : Finset ℕ) (R : ℝ) (p : ℕ) (P : MvPolynomial (DV.Rt Tau N) ℝ) : I.arMix S R p (DV.tagHT Tau N P) = I.arDeg (insert p S) R P

theorem **arMix_DR** (S : Finset ℕ) (R : ℝ) (p : ℕ) (P : MvPolynomial (DV.Rt Tau N) ℝ) : I.arMix S R p (DV.tagDR Tau N P) = I.arDeg (S.erase p) R P

theorem **locSt_padOf** : LocSt.ofPkg (padOf s L G M F I) = L

theorem **globSt_padOf** : GlobSt.ofPkg (padOf s L G M F I) = G

theorem **normDet_false** (p : ℕ) (hp : p.Prime) (τ : D.what₂.Tau) (y : Pt s) (a : Qb) : normDet s p false τ y a = normDet s p true τ y a * Real.exp (∑ i, phi s p y (τ, i))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ApolarMonomial

theorem **apolar_monomial** (m : V →₀ ℕ) (c : S) (f : MvPowerSeries V S) : apolar (MvPolynomial.monomial m c) f = c * ((m.prod fun _ n => (n.factorial : S)) * MvPowerSeries.coeff m f)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.AssembleLattice

theorem **assemble_R** : (assemble B C W V S N G).R = B.R

theorem **assemble_host** : (assemble B C W V S N G).host = B.host

theorem **assemble_sp** : (assemble B C W V S N G).sp = B.sp

theorem **assemble_rows** : (assemble B C W V S N G).what₂ = W.what₂

theorem **assemble_hc** : (assemble B C W V S N G).hc = B.hc

theorem **assemble_hcW** : (assemble B C W V S N G).hcW = B.hcW

theorem **assemble_u₀** : (assemble B C W V S N G).u₀ = B.u₀

theorem **assemble_mem** : (assemble B C W V S N G).mem = C.mem

theorem **assemble_div** : (assemble B C W V S N G).div = V.div

theorem **assemble_embE** : (assemble B C W V S N G).embE = B.embE

theorem **assemble_lattice** : (assemble B C W V S N G).LL0 = B.LL0

theorem **assemble_s** : (assemble B C W V S N G).s = S.s

theorem **assemble_nd** : (assemble B C W V S N G).nd = N.nd

theorem **assemble_loc** : (assemble B C W V S N G).loc = N.loc

theorem **assemble_fm** : (assemble B C W V S N G).fm = C.fm

theorem **assemble_jet** : (assemble B C W V S N G).jet = G.jet

theorem **assemble_gx** : (assemble B C W V S N G).gx = G.gx

theorem **assemble_vol** : (assemble B C W V S N G).vol = G.vol

theorem **assemble_cd** : (assemble B C W V S N G).cd = G.cd

theorem **assemble_dsig** : (assemble B C W V S N G).dsig = S.dsig

theorem **assemble_chTw** : (assemble B C W V S N G).chTw = C.chTw

theorem **assemble_cOf** : (assemble B C W V S N G).cOf = B.cOf

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.BaseDataEmbeddingConj

theorem **fields_star** (x : B.host.E) : letI

theorem **fields_sig** (x : B.host.E) : B.fields.sig x = B.host.conj x

theorem **fields_emb_apply** (τ : Tau B.R) (x : B.host.E) : B.fields.emb τ x = B.embE (B.host.sigmaOf (τ : B.R.GG) x)

theorem **fields_f** : B.fields.f = B.R.f

theorem **fields_embC** (τ : Tau B.R) : B.fields.embC B.embLL τ = B.host.emb (τ : B.R.GG)

theorem **fields_conjO_coe** (x : 𝒪 B.host.E) : ((B.fields.conjO x : 𝒪 B.host.E) : B.host.E) = B.host.conj (x : B.host.E)

theorem **primes_piGCong** (X : PreX B.toPre) : (B.primes X).PiGCong

theorem **primes_piG_norm** (X : PreX B.toPre) (i : Fin 2) : (B.primes X).piG i * B.fields.conjO ((B.primes X).piG i) = ((B.primes X).g i : 𝒪 B.fields.E)

theorem **primes_piT_norm** (X : PreX B.toPre) (u : B.Word) : (B.primes X).piT u * B.fields.conjO ((B.primes X).piT u) = ((B.primes X).T u : 𝒪 B.fields.E)

theorem **mem_wts_tops** (τ : Tau B.R) : τ ∈ B.wts.tops ↔ (τ : B.R.GG) ∈ B.host.topPlaces

theorem **wts_isOk** : B.wts.Ok where

theorem **primes_sizes** (X : PreX B.toPre) : Sizes (B.primes X) B.wts X.a where

theorem **baseData_embedding_conj** (X : PreX B.toPre) (Ch : Chars B.fields (B.primes X) B.R.m) (x : (B.dat X Ch).X.K) : (B.dat X Ch).sig0 (star x) = starRingEnd ℂ ((B.dat X Ch).sig0 x)

theorem **dat_e_even** (X : PreX B.toPre) (Ch : Chars B.fields (B.primes X) B.R.m) (j : Fin (B.dat X Ch).R.m) : ∃ z : ℤ, ∑ τ, ((B.dat X Ch).W.row τ).e j = 2 * (z : ℚ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.BaseLevelRowSideTransport

theorem **Refill.ofSG_to** (a : Late B C) (S : SStage B C a.W) (G : GJStage B C) : (Refill.ofSG a S G).to = ⟨a.W, a.V, S, a.N, G⟩

theorem **spEEc_eq** : (toChainReal R F tupT hm hTop).EEc = spEEc c F tupT hm hTop

theorem **spCC_eq** : (toChainReal R F tupT hm hTop).CC = spCC c F tupT hm hTop

theorem **spCCP_eq** : (toChainReal R F tupT hm hTop).CCP = spCCP c F tupT hm hTop

theorem **spCCX_eq** : (toChainReal R F tupT hm hTop).CCX = spCCX c F tupT hm hTop

theorem **spCCXP_eq** : (toChainReal R F tupT hm hTop).CCXP = spCCXP c F tupT hm hTop

theorem **okB_imp** {l : Bool} {μ : a.D.what₂.Tup} (h : okB c F tupT hm hTop l μ) : (toChainReal R F tupT hm hTop).IoOK l μ

theorem **eq_via** {N : Type} {r : ℂ} (f : N → ℂ) {u u' v : N} (h : r = f u) (e : v = u') (hu : u' = u) : r = f v

theorem **ioCB_eq** (l : Bool) (μ : a.D.what₂.Tup) (k : ℕ) (v : (toChainReal R F tupT hm hTop).CC l μ (k + 1)) : ioCB c l μ k v = ((toChainReal R F tupT hm hTop).ioC l μ k v).1

theorem **ioXB'_eq** (l : Bool) (μ : a.D.what₂.Tup) (v : (toChainReal R F tupT hm hTop).CCX l μ) : ioXB' c l μ v = ((toChainReal R F tupT hm hTop).ioX l μ v).1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.BrgStdR

theorem **StdSide.value** {ι : Type} {lam : X.Tau → DomWeight n} {rho : GL (Fin n) X.E →* Module.End X.K (VV X.K lam)} {SG : ι → Finset (Pl X.E)} {Fm : ι → Type} {val : ∀ t, Fm t → DecoSet X.E n ∅ (SG t) → VV X.K lam} {ad : ∀ t, Fm t → H → W} (s : StdSide σ mat rat ρ lam rho SG Fm val ad) : ∃ c : VV X.K lam →ₛₗ[σ] W, (LinearIndependent ℂ fun i => c (bVV X.K lam i)) ∧ ∀ h t, ∃ d : DecoSet X.E n ∅ (SG t), ∀ φ, ad t φ h = c (val t φ d)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CardLiesOverEqIndex

theorem **levDeg_pos** (h : l.Dvd l₂) (Xo : Moduli Dw l) (c : Coeff l) (Xo₂ : Moduli Dw l₂) (c₂ : Coeff l₂) : 0 < levDeg h Xo c Xo₂ c₂

theorem **card_lifts_pos** {k : Type} [Field k] [IsAlgClosed k] {X Y : Scheme.{0}} (π : X ⟶ Y) [IsFinite π] [Surjective π] (y : Sp k ⟶ Y) : 0 < Nat.card {x : Sp k ⟶ X // x ≫ π = y}

theorem **card_liesOver_eq_index** (hpq : 1 ≤ Dw.p + Dw.q) (h : l.Dvd l₂) {Xo : Moduli Dw l} {c : Coeff l} (Xo₂ : Moduli Dw l₂) (c₂ : Coeff l₂) (y : CxPt Xo c) : 0 < Nat.card {y₂ : CxPt Xo₂ c₂ // LiesOver h y₂ y} ∧ Nat.card {y₂ : CxPt Xo₂ c₂ // LiesOver h y₂ y} = ((trivAt Dw l₂).subgroupOf (relGroup Dw l l₂)).index

theorem **card_liesOver** (hpq : 1 ≤ Dw.p + Dw.q) (h : l.Dvd l₂) {Xo : Moduli Dw l} {c : Coeff l} (Xo₂ : Moduli Dw l₂) (c₂ : Coeff l₂) (y : CxPt Xo c) : Nat.card {y₂ : CxPt Xo₂ c₂ // LiesOver h y₂ y} = levDeg h Xo c Xo₂ c₂

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CardStabilizerForgetFr

theorem **mem_levRed** {P : HeightOneSpectrum (𝒪 E)} (l : Lev (𝒪 E ⧸ P.asIdeal) K n) (k : GLint E n P) : k ∈ (levRed P l).H ↔ redGL n P k ∈ l.H

theorem **levRed_lam** {P : HeightOneSpectrum (𝒪 E)} (l : Lev (𝒪 E ⧸ P.asIdeal) K n) (k : (levRed P l).H) : (levRed P l).lam k = l.lam (redGL n P k.1, k.2)

theorem **levRed_root** (P : HeightOneSpectrum (𝒪 E)) (l : Lev (𝒪 E ⧸ P.asIdeal) K n) : ∃ m : ℕ, 0 < m ∧ ∀ k, (levRed P l).lam k ^ m = 1

theorem **exists_redGL_eq** [NeZero n] (h : GLG E n Q) (w : Q) : ∃ k : GLint E n w.1, redGL n w.1 k = h w

theorem **redGL_surjective** (P : HeightOneSpectrum (𝒪 E)) : Function.Surjective (redGL (E

theorem **frBe_apply** {P : HeightOneSpectrum (𝒪 E)} (f : Sp E n ≃ₗ[E] Sp E n) {x : Sp E n} (hx : x ∈ stdAt E n P) : frBe P f (f x) = redV P x

theorem **frBe_off** {P : HeightOneSpectrum (𝒪 E)} (f : Sp E n ≃ₗ[E] Sp E n) {x : Sp E n} (hx : f.symm x ∉ stdAt E n P) : frBe P f x = 0

theorem **be_eq_frBe** {d : Deco E n ST SG} (hd : d.Ok) {w : SG} {f : Sp E n ≃ₗ[E] Sp E n} (hf : IsLiftFr d w f) : d.be w = frBe w.1 f

theorem **frBe_move** {P : HeightOneSpectrum (𝒪 E)} (γ f : Sp E n ≃ₗ[E] Sp E n) (c : GLint E n P) (x : Sp E n) : frBe P (γ ∗ f ∗ (c⁻¹).1) x = (redGL n P c).1.mulVec (frBe P f (γ.symm x))

theorem **forgetDeco_be_of_mem** (p : DecoF E n ST SG Q) (w : uSG SG Q) (h : w.1 ∈ SG) : (forgetDeco p).be w = p.deco.1.be ⟨w.1, h⟩

theorem **forgetDeco_be_of_not** (p : DecoF E n ST SG Q) (w : uSG SG Q) (h : w.1 ∉ SG) : (forgetDeco p).be w = frBe w.1 (p.fr (rightOf w h))

theorem **forgetDeco_ok** (p : DecoF E n ST SG Q) : (forgetDeco p).Ok

theorem **forgetFr_N** (p : DecoF E n ST SG Q) : (forgetFr p).1.N = p.deco.1.N

theorem **forgetFr_f** (p : DecoF E n ST SG Q) : (forgetFr p).1.f = p.deco.1.f

theorem **forgetFr_res** (p : DecoF E n ST SG Q) : (forgetFr p).res Finset.subset_union_left = p.deco

theorem **isLiftFr_forgetFr** (p : DecoF E n ST SG Q) (w : uSG SG Q) (h : w.1 ∉ SG) : IsLiftFr (forgetFr p).1 w (p.fr (rightOf w h))

theorem **forgetFr_surjective** : Function.Surjective (forgetFr : DecoF E n ST SG Q → DecoSet E n ST (uSG SG Q))

theorem **redGQ_of_mem** (a : GLG E n SG × GLQ E n Q) (w : uSG SG Q) (h : w.1 ∈ SG) : redGQ E n SG Q a w = a.1 ⟨w.1, h⟩

theorem **redGQ_of_not** (a : GLG E n SG × GLQ E n Q) (w : uSG SG Q) (h : w.1 ∉ SG) : redGQ E n SG Q a w = redGL n w.1 (a.2 (rightOf w h))

theorem **forgetFr_smul** (a : UF E n × GLG E n SG × GLQ E n Q) (p : DecoF E n ST SG Q) : forgetFr (a • p) = (a.1, redGQ E n SG Q a.2) • forgetFr p

theorem **redGQ_eq_one** {c : GLQ E n Q} (hc : ∀ P : Q, redGL n P.1 (c P) = 1) : redGQ E n SG Q (1, c) = 1

theorem **exists_of_forgetFr_eq** (hd : Disjoint SG Q) {p q : DecoF E n ST SG Q} (h : forgetFr p = forgetFr q) (P : Q) : ∃ κ : GLint E n P.1, redGL n P.1 κ = 1 ∧ p.fr P = q.fr P ∗ κ.1

theorem **forgetFr_eq_iff** (hd : Disjoint SG Q) (p q : DecoF E n ST SG Q) : forgetFr p = forgetFr q ↔ ∃ c : GLQ E n Q, (∀ P : Q, redGL n P.1 (c P) = 1) ∧ q = ((1, 1, c) : UF E n × GLG E n SG × GLQ E n Q) • p

theorem **forgetFr_base** [NeZero n] : forgetFr (DecoF.base E n ST SG Q) = DecoSet.base E n ST (uSG SG Q)

theorem **redGQ_mem** (h : Λ.grp) (c : (levRedFam 0).grp) (w : uSG SG Q) : redGQ X.E n SG Q (h.1, c.1) w ∈ (Λ.glue 0 w).H

theorem **forgetFr_frm_smul** (g : FrmG Λ (levRedFam 0)) (p : DecoF X.E n ST SG Q) : forgetFr (g • p) = redFG Λ 0 g • forgetFr p

theorem **redFG_surjective** : Function.Surjective (redFG Λ 0)

theorem **lev_lam_congr** {κ K : Type} [CommRing κ] [Field K] {l l' : Lev κ K n} (e : l = l') {a a' : GL (Fin n) κ} (e' : a = a') (h : a ∈ l.H) (h' : a' ∈ l'.H) : l.lam ⟨a, h⟩ = l'.lam ⟨a', h'⟩

theorem **levQ_chr_apply** (0 : LevQFam X.E X.K n Q) (c : 0.grp) : 0.chr c = ∏ P, (0 P).lam ⟨c.1 P, c.2 P (Set.mem_univ P)⟩

theorem **chr_redFG** (hd : Disjoint SG Q) (g : FrmG Λ (levRedFam 0)) : Λ.chr g.2.1 ∗ (levRedFam 0).chr g.2.2 = ((Λ.glue 0).inv.chr (redFG Λ 0 g).2)⁻¹

theorem **exists_redFG_smul** (hd : Disjoint SG Q) (a : FormG (Λ.glue 0).inv) (p q : DecoF X.E n ST SG Q) (h : a • forgetFr p = forgetFr q) : ∃ g : FrmG Λ (levRedFam 0), redFG Λ 0 g = a ∧ g • p = q

theorem **eq_of_redFG_eq** {g g' : FrmG Λ (levRedFam 0)} {p : DecoF X.E n ST SG Q} (h : g • p = g' • p) (e : redFG Λ 0 g = redFG Λ 0 g') : g = g'

theorem **card_stabilizer_forgetFr** (hd : Disjoint SG Q) (p : DecoF X.E n ST SG Q) : Nat.card (MulAction.stabilizer (FrmG Λ (levRedFam 0)) p) = Nat.card (MulAction.stabilizer (FormG (Λ.glue 0).inv) (forgetFr p))

theorem **unitOf_val** {pi : 𝒪 E} {P : HeightOneSpectrum (𝒪 E)} (h : pi ∉ P.asIdeal) : (unitOf pi : E) = pi

theorem **sumE_inj** {g g' : Sp E n ≃ₗ[E] Sp E n} {t : Eˣ} (h : sumE hN g t = sumE hN g' t) : g = g'

theorem **extQ_injective** : Function.Injective (extQ E hN Q)

theorem **tilH_N_f** {SG' : Finset (HeightOneSpectrum (𝒪 E))} (hG : ∀ w : SG, pi ∉ w.1.asIdeal) (hG' : ∀ w : SG', pi ∉ w.1.asIdeal) (d : DecoSet E n ⌀ SG) (d' : DecoSet E n ⌀ SG') (h : d.1.N = d'.1.N) : (DecoSet.tilH (ST

theorem **tilFr_fits** (p : DecoF E n ⌀ SG Q) (P : Q) (x : Sp E N) : sumE hN (p.fr P) (unitOf pi) x ∈ (DecoSet.tilH (ST

theorem **tilFr_eq** (p : DecoF E n ⌀ SG Q) : tilFr (ST

theorem **tilFr_deco** (p : DecoF E n ⌀ SG Q) : (tilFr (ST

theorem **tilFr_fr** (p : DecoF E n ⌀ SG Q) (P : Q) : (tilFr (ST

theorem **tilFr_fr_last** (p : DecoF E n ⌀ SG Q) (P : Q) : (tilFr (ST

theorem **tilFr_smul** (a : UF E n × GLG E n SG × GLQ E n Q) (p : DecoF E n ⌀ SG Q) : tilFr (ST

theorem **tilFr_injective** : Function.Injective (tilFr hN pi : DecoF E n ⌀ SG Q → DecoF E N ST SG Q)

theorem **forgetFr_tilFr** (p : DecoF E n ⌀ SG Q) : forgetFr (tilFr (ST

theorem **mem_levQX** (P : QE sel (mu : Chr X.F X.E X.K)) (k : GLint X.E (R.N R.m − 1) P.1) : k ∈ (levQX R sel mu P).H ↔ redGL (R.N R.m − 1) P.1 k ∈ (rowTyQ R (mu : Chr X.F X.E X.K) (QE sel (mu : Chr X.F X.E X.K)) P).H

theorem **levQX_lam** (P : QE sel (mu : Chr X.F X.E X.K)) (k : (levQX R sel mu P).H) : (levQX R sel mu P).lam k = (rowTyQ R (mu : Chr X.F X.E X.K) (QE sel (mu : Chr X.F X.E X.K)) P).lam (redGL (R.N R.m − 1) P.1 k.1, k.2)

theorem **levQX_finiteIndex** (P : QE sel (mu : Chr X.F X.E X.K)) : (levQX R sel mu P).H.FiniteIndex

theorem **levQX_root** (P : QE sel (mu : Chr X.F X.E X.K)) : ∃ m : ℕ, 0 < m ∧ ∀ k, (levQX R sel mu P).lam k ^ m = 1

theorem **toRow_N** (p : XPt Pr R sel mu) : (toRow sel mu p).1.N = p.deco.1.N

theorem **toRow_surjective** : Function.Surjective (toRow (Pr

theorem **isLiftFr_toRow** (p : XPt Pr R sel mu) (w : rowSG Pr sel mu) (h : w.1 ∉ Pr.vGs) : IsLiftFr (toRow sel mu p).1 w (p.fr (rightOf w h))

theorem **toRow_smul** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) (p : XPt Pr R sel mu) : toRow sel mu (g • p) = toRowG Ch sel mu P g • toRow sel mu p

theorem **stepX_smul_trivPt** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) (p : XPt Pr R sel mu) : (letI

theorem **chr_stepX** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) : (stepX Ch sel mu P).chr () g = ((rowTy Ch sel mu P).inv.chr (toRowG Ch sel mu P g).2)⁻¹

theorem **mtilde_rho_toRowG** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) : (mtildeGSD W Ch sel mu P).ρ (toRowG Ch sel mu P g) = ((((stepX Ch sel mu P).chr () g)⁻¹ : X.K×) : X.K) • (rowVal W (rowSG Pr sel mu)).rho g.1

theorem **exists_toRowG_smul** (a : FormG (rowTy Ch sel mu P).inv) (p q : XPt Pr R sel mu) (h : a • toRow sel mu p = toRow sel mu q) : ∃ g : FrmG (rowTyG Ch P) (levQX R sel mu), toRowG Ch sel mu P g = a ∧ g • p = q

theorem **card_stabilizer_toRow** (p : XPt Pr R sel mu) : Nat.card (MulAction.stabilizer (FrmG (rowTyG Ch P) (levQX R sel mu)) p) = Nat.card (MulAction.stabilizer (FormG (rowTy Ch sel mu P).inv) (toRow sel mu p))

theorem **tilF_deco** (p : XPt Pr R sel mu) : (tilF sel mu u p).deco = DecoSet.tilH (N_m_succ R) (Pr.piT u) p.deco

theorem **tilF_fr** (p : XPt Pr R sel mu) (P : QE sel (mu : Chr X.F X.E X.K)) : (tilF sel mu u p).fr P = sumE (N_m_succ R) (p.fr P) (unitOf (Pr.piT u))

theorem **forgetFr_tilF** (p : XPt Pr R sel mu) : forgetFr (tilF sel mu u p) = tilQ sel mu u (toRow sel mu p)

theorem **tilF_N** (p : XPt Pr R sel mu) : (tilF sel mu u p).deco.1.N = (tilQ sel mu u (toRow sel mu p)).1.N

theorem **tilF_smul** (a : UF X.E (R.N R.m − 1) × GLG X.E (R.N R.m − 1) Pr.vGs × GLQ X.E (R.N R.m − 1) (QE sel (mu : Chr X.F X.E X.K))) (p : XPt Pr R sel mu) : tilF sel mu u (a • p) = ((extUH (N_m_succ R) a.1, extGH (N_m_succ R) a.2.1, extQ X.E (N_m_succ R) _ a.2.2) : UF X.E (R.N R.m) × GLG X.E (R.N R.m) Pr.vGs × GLQ X.E (R.N R.m) (QE sel (mu : Chr X.F X.E X.K))) • tilF sel mu u p

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CenLocOfCenReal

theorem **cenLoc_of_cenReal** {D : D7} {M : MtFam D} (h : GlobalTheory.PendOf.CenReal D M) : CenLoc D M

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CenUsedOfCenReal

theorem **cenUsed_of_cenReal** {D : D7} {M : MtFam D} (h : GlobalTheory.PendOf.CenReal D M) : CenUsed D M

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CharpolyEqOfIsNilpotentSub

theorem **semiconj_exp** {R : Type*} [Ring R] [Algebra ℚ R] {x a b : R} (ha : IsNilpotent a) (hb : IsNilpotent b) (hx : x * a = b * x) : x * exp a = exp b * x

theorem **exp_add_smul** {N : Matrix n n K} (hN : IsNilpotent N) (s t : K) : exp ((s + t) • N) = exp (s • N) * exp (t • N)

theorem **exp_smul_mul** {N : Matrix n n K} (hN : IsNilpotent N) (s : K) : exp (s • N) * N = N * exp (s • N)

theorem **eq_of_exp_smul_eq** {N N' : Matrix n n K} (hN : IsNilpotent N) (hN' : IsNilpotent N') {c : K} (hc : c ≠ 0) (H : ∀ k : ℕ, exp (((k : K) * c) • N) = exp (((k : K) * c) • N')) : N = N'

theorem **charpoly_eq_of_isNilpotent_sub** {K V : Type*} [Field K] [Infinite K] [AddCommGroup V] [Module K V] [FiniteDimensional K V] {f g : End K V} (hc : Commute f g) (hn : IsNilpotent (g − f)) : f.charpoly = g.charpoly

theorem **isFrobSS_self** (a : WDRep h V) : IsFrobSS a.ρ a.N a

theorem **IsFrobSS.ρ_eq** {ρ : Weil h → End ℂ V} {N : End ℂ V} {a : WDRep h V} (ha : IsFrobSS ρ N a) {g : Weil h} (hg : (ρ g).IsSemisimple) : a.ρ g = ρ g

theorem **IsFrobSS.ρ_unique** {ρ : Weil h → End ℂ V} {N N' : End ℂ V} {a b : WDRep h V} (ha : IsFrobSS ρ N a) (hb : IsFrobSS ρ N' b) : a.ρ = b.ρ

theorem **IsFrobSS.iso** {ρ : Weil h → End ℂ V} {N : End ℂ V} {a : WDRep h V} {ρ' : Weil h → End ℂ V'} {N' : End ℂ V'} {b : WDRep h V'} (ha : IsFrobSS ρ N a) (hb : IsFrobSS ρ' N' b) (e : V ≃ₗ[ℂ] V') (hρ : ∀ g v, e (ρ g v) = ρ' g (e v)) (hN : ∀ v, e (N v) = N' (e v)) : a.Iso b

theorem **IsFrobSS.iso_of_smul** {ρ : Weil h → End ℂ V} {N : End ℂ V} {a b : WDRep h V} {c : ℂ} (hc : c ≠ 0) (ha : IsFrobSS ρ N a) (hb : IsFrobSS ρ (c • N) b) : a.Iso b

theorem **isFrobSS_of_unipotent** {ρ : Weil h → End ℂ V} {a : WDRep h V} (u : (End ℂ V)×) (hu : IsNilpotent ((u : End ℂ V) − 1)) (hc : ∀ g, Commute (u : End ℂ V) (ρ g)) (hρ : ∀ g, ρ g = a.ρ g * ((u ^ deg g : (End ℂ V)×) : End ℂ V)) : IsFrobSS ρ a.N a

theorem **exists_isFrobSS** (ρ : Representation ℂ (Weil h) V) (N : End ℂ V) (hρ : IsSmooth ρ) (hN : IsNilpotent N) (hc : ∀ g, ρ g * N = ((normV w : ℕ) : ℂ) ^ deg g • (N * ρ g)) : ∃ a : WDRep h V, IsFrobSS ρ N a

theorem **IsFrobSS.charpoly_eq** {ρ : Weil h → End ℂ V} {N : End ℂ V} {a : WDRep h V} (ha : IsFrobSS ρ N a) (g : Weil h) : (a.ρ g).charpoly = (ρ g).charpoly

theorem **IsFrobSS.ρ_ineIncl** {ρ : Representation ℂ (Weil h) V} {N : End ℂ V} {a : WDRep h V} (ha : IsFrobSS ρ N a) (hρ : IsSmooth ρ) (σ : Ine (F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ChiLenPicPushforwardOfIsClosedImmersion

theorem **rel'** (i i₁ : L.ι) (j j₁ : L'.ι) {W : Z.X.Opens} (h : W ≤ Z.gen ⊓ (L.U i ⊓ L'.U j)) (h₁ : W ≤ Z.gen ⊓ (L.U i₁ ⊓ L'.U j₁)) (hi : W ≤ L.U i ⊓ L.U i₁) (hj : W ≤ L'.U j ⊓ L'.U j₁) : res h (φ.u i j) * res hj (L'.g j j₁) = res hi (L.g i i₁) * res h₁ (φ.u i₁ j₁)

theorem **chart_eq** (p : (L.pull Z.gen.ι).ι × (L'.pull Z.gen.ι).ι) : (L.pull Z.gen.ι).U p.1 ⊓ (L'.pull Z.gen.ι).U p.2 = Z.gen.ι ⁻¹ᵁ (Z.gen ⊓ (L.U (p.1 ⟨⟩) ⊓ L'.U (p.2 ⟨⟩)))

theorem **isGen_iff** (u : ∀ (i : L.ι) (j : L'.ι), Γ(Z.X, Z.gen ⊓ (L.U i ⊓ L'.U j))×) : LB.Gen.IsGen (L

theorem **isGen_toIso** : φ.toIso.IsGen

theorem **bijective_res_pullU** {V : Z.X.Opens} (hV : V ≤ Z.gen) {W : Z.gen.toScheme.Opens} (h : W = Z.gen.ι ⁻¹ᵁ V) : Function.Bijective fun u => res h.le (pullU Z.gen.ι V u)

@[simp] theorem **vCls_div** : Z.vCls (φ.div hL hL') = cls L hL − cls L' hL'

theorem **div_eq_add** (φ' : Z.GenIso L' L'') (φ'' : Z.GenIso L L'') (h : ∀ i j k, res (inf_le_inf_left _ (inf_le_inf_right _ inf_le_left) : Z.gen ⊓ (L.U i ⊓ L'.U j ⊓ L''.U k) ≤ _) (φ''.u i k) = res (inf_le_inf_left _ inf_le_left) (φ.u i j) * res (inf_le_inf_left _

(inf_le_inf_right _ inf_le_right)) (φ'.u j k)) : φ''.div hL hL'' = φ.div hL hL' + φ'.div hL' hL''

@[simp] theorem **div_ofGlobal** (w : LB.Iso L' L) (hw : w.IsGen) : (ofGlobal w hw : Z.GenIso L L').div hL hL' = 0

@[simp] theorem **div_smul** (g : O) (hg : g ≠ 0) : (smul hL g hg).div hL hL = Z.div g hg

@[simp] theorem **div_symm** : φ.symm.div hL' hL = -φ.div hL hL'

@[simp] theorem **div_dual** : φ.dual.div (isLB_dual hL) (isLB_dual hL') = -φ.div hL hL'

@[simp] theorem **cDQ_toQ** (D : Z.VDivZ) : Z.cDQ (toQ D) = firstChernPic (Z.vCls D)

theorem **capDeg_vp_toQ** (p d : ℕ) (D : Z.VDivZ) (Q : PicSym Z.X) : Z.capDeg_vp p d (toQ D) Q = Z.vDeg d D Q / ramIdx O p

@[simp] theorem **vResQ_toQ** (C : Z.Comp) (D : Z.VDivZ) : vResQ C (toQ D) = Model.toQ (Z.vRes C D)

theorem **capDeg_vp_eq_sum** (p d : ℕ) (D : Z.VDivQ) (Q : PicSym Z.X) : Z.capDeg_vp p d D Q = ∑ C ∈ Finset.univ.filter fun C : Z.Comp => (Z.comp C).relDim = d, (Z.comp C).capDeg_vp p (vResQ C D) (picSymPull (Z.compι C) Q)

theorem **capDeg_vp_eq_zero** (p : ℕ) {d n : ℕ} (D : Z.VDivQ) (hn : n ≠ d) {x : PicSym Z.X} (hx : x ∈ symDeg (SmallPic Z.X) n) : Z.capDeg_vp p d D x = 0

theorem **vDeg_prod** {d : ℕ} {D : Z.VDivZ} {I : ∀ C, (Z.comp C).effVert} (h : ∀ C, Z.vRes C D = Model.ofEff (I C)) (L : Fin d → FinLB Z.X) : Z.vDeg d D (∏ i, (L i).c1) = ∑ C ∈ Finset.univ.filter fun C : Z.Comp => (Z.comp C).relDim = d, (interLen O (divMod (I C).1) fun i => ((L i).pull (Z.compι C)).lb : ℚ)

theorem **toQ_div** (g : O) (hg : g ≠ 0) : toQ (Z.div g hg) = (((IsDiscreteValuationRing.addVal O g).toNat : ℚ) / ramIdx O p) • toQ (Z.divP p)

theorem **firstChern_eq_sub_divisor** (L L' : FinLB Z.X) (u : Z.GenIso L.lb L'.lb) : L'.c1 = L.c1 - Z.cDQ (Z.vdiv L L' u)

theorem **vdiv_dual** (L L' : FinLB Z.X) (u : Z.GenIso L.lb L'.lb) : Z.vdiv L.dual L'.dual u.dual = -Z.vdiv L L' u

theorem **vdiv_symm** (L L' : FinLB Z.X) (u : Z.GenIso L.lb L'.lb) : Z.vdiv L' L u.symm = -Z.vdiv L L' u

theorem **vdiv_smul** (L : FinLB Z.X) (g : O) (hg : g ≠ 0) : Z.vdiv L L (.smul L.isLB g hg) = toQ (Z.div g hg)

theorem **chiLenPic_pushforward_of_isClosedImmersion** {X Y : Scheme.{0}} (R : Type) [CommRing R] [Algebra R Γ(X, ⊤)] [Algebra R Γ(Y, ⊤)] (i : Y ⟶ X) [IsClosedImmersion i] (hi : ∀ r : R, i.appTop (algebraMap R Γ(X, ⊤) r) = algebraMap R Γ(Y, ⊤) r) (H : Y.Modules) : chiLenPic R ((Scheme.Modules.pushforward i).obj H) = chiLenPic R H ∘ pull i

theorem **relDim_eq_of_isIso** : B'.relDim = B.relDim

theorem **vDeg_vPull_of_isIso** (D : B.VDivZ) (x : PicSym B.X) : B'.vDeg (Model.vPull hf D) (picSymPull f x) = B.vDeg D x

theorem **vDeg_div_of_cover** {ι : Type} [Fintype ι] {B : ι → Model O} (f : ∀ i, (B i).X ⟶ Z.X) [∀ i, IsOpenImmersion (f i)] (hf : ∀ i, f i ≫ Z.str = (B i).str) (hc : ∀ x, ∃ i, x ∈ Set.range (f i)) (hd : ∀ i j, (Set.range (f i) ∩ Set.range (f j)).Nonempty → i = j) (d : ℕ) {g : O} (hg : g ≠ 0) (x : PicSym Z.X) : Z.vDeg d (Z.div g hg) x = ∑ i ∈ Finset.univ.filter fun i => (B i).relDim = d, (B i).vDeg ((B i).div g hg) (picSymPull (f i) x)

theorem **capDeg_divP_copies** : ((copies O n (N + 1)).toITh p (N + 1)).capDeg ((copies O n (N + 1)).toITh p (N + 1)).divP ((twistingSheafOnCopies O n (N + 1)).c1 ^ (N + 1)) = n

theorem **natCast_mem_nonunits_V** {l : Level H} (c : Coeff l) (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) : (p : c.F) ∈ (c.V W).nonunits

theorem **isDiscreteValuationRing_V** {l : Level H} (c : Coeff l) (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) : IsDiscreteValuationRing (c.V W)

theorem **charP_residueField_of_mem_nonunits** {l : Level H} (c : Coeff l) (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) : CharP (ResidueField (c.V W)) p

theorem **dim_ithOf** : (ithOf hp hW Z d).dim = d + 1

theorem **LB_ithOf** : (ithOf hp hW Z d).LB = FinLB Z.X

theorem **firstChern_lineBundleTheory** (L : FinLB Z.X) : (ithOf hp hW Z d).c1 L = L.c1

theorem **firstChern_dual_eq_neg** (L : FinLB Z.X) : (ithOf hp hW Z d).c1 L.dual = -(ithOf hp hW Z d).c1 L

theorem **capDeg_ithOf** (D : (ithOf hp hW Z d).VDiv) (Q : (ithOf hp hW Z d).Ch) : haveI

theorem **firstChern_eq_sub_divisor** (L L' : FinLB Z.X) (u : GenIso Z L L') : (ithOf hp hW Z d).c1 L' = (ithOf hp hW Z d).c1 L - (ithOf hp hW Z d).cD (vdiv hp hW d L L' u)

theorem **vdiv_dual** (L L' : FinLB Z.X) (u : GenIso Z L L') : vdiv hp hW d L.dual L'.dual (NModel.GenIso.dual u) = -vdiv hp hW d L L' u

theorem **gen_eq_basicOpen** : (nmodel Z).gen = Z.X.basicOpen (p : Γ(Z.X, ⊤))

theorem **mem_gen_iff_inQ** {x : Z.X} : x ∈ (nmodel Z).gen ↔ Z.inQ x

theorem **chart_le_gen** (a : Fin Z.m) : (Z.U a).1 ≤ (nmodel Z).gen

theorem **opn_subset_gen** : Z.opn ⊆ (nmodel Z).gen

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ChtTrueYEta

theorem **exists_fam** : FamSt

theorem **Cht.goodIff** (ct : At S κ p) : Cht.GoodIff ct

theorem **Cht.genEx** (ct : At S κ p) : Cht.GenEx ct

theorem **chtTrue_genK** (ct : At S κ p) {Z : ct.Mdl} (η : ct.Gen Z) : Cht.IsGenK η ((chtTrue ct).genK η)

theorem **chtTrue_yEta** (ct : At S κ p) {Z : ct.Mdl} (η : ct.Gen Z) : (chtTrue ct).yEta η = (chtTrue ct).ptK Z ((chtTrue ct).genK η)

theorem **Cht.nonempty_dvp** (ct : At S κ p) {Z : ct.Mdl} (η : ct.Gen Z) : ∃ y : Cht.DVP ct, Nonempty (Cht.Ext y)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ClosureImageSpan

theorem **closure_image_span** {K : Type} [Field K] [NumberField K] {σ : K →+* ℂ} {V H : Type} [AddCommGroup V] [Module (𝒪 K) V] [AddCommGroup H] [Module ℂ H] (f : V →+ H) (hfs : ∀ (b : 𝒪 K) (x : V), f (b • x) = σ (b : K) • f x) (X : Set V) : AddSubgroup.closure {y | ∃ a : 𝒪 K, ∃ x ∈ f '' X, y = σ (a : K) • x} = AddSubgroup.closure {y | ∃ a : 𝒪 K, ∃ x ∈ f '' (Submodule.span (𝒪 K) X : Set V), y = σ (a : K) • x}

theorem **majAt_self** {K : Type} [Field K] [NumberField K] {n : ℕ} (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ) (v : InfinitePlace K) (x : Fin n → K) : Formalism.HermitianLatticesTheory.majAt (Pi.basisFun K (Fin n)) X v x x = ((Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (Pi.basisFun K (Fin n)) X v x) ^ 2 : ℝ) : ℂ)

theorem **norm_w0C_sq** {K : Type} [Field K] [NumberField K] [StarRing K] {Bs : DefBase K} {A : DefAmb Bs} (X : CxData A) (b : Bool) (hk : X.KForm b) (he : X.EWnorm b) (j : Bs.J) (x : Fin (X.dW b j) → K) : ‖X.w0C b j (cpt Bs.v0.embedding x)‖ ^ 2 = X.nP b * Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (Pi.basisFun K _) (X.XW b j) Bs.v0 x) ^ 2

theorem **eW_jW_eq** (Z : ReadL R F X0 Y) (i : Fin D.R.m) (y) : Y.mj.eW i (Z.jW i y) = GlobalTheory.Mem.LiftTheta.goodForm R.toRealDat F.lv2 false i y

theorem **eW_inj** (Y : Read (A3

theorem **jW_basis** (Z : ReadL R F X0 Y) (hH : Holds R F) (i : Fin D.R.m) : LinearIndependent ℂ (fun k => Z.jW i (Pi.single k 1)) ∧ Submodule.span ℂ (Set.range fun k => Z.jW i (Pi.single k 1)) = ⊤

theorem **chiKs** (hP : Pins R F cG) (i : Fin D.R.m) : (R.FW i).ChiKs

theorem **jW_lat** (Z : ReadL R F X0 Y) (hP : Pins R F cG) (hH : Holds R F) (i : Fin D.R.m) : OLatTie D.what₂.embLL (D.what₂.MjLat ((i : ℕ) + 1)) (Z.jW i) ((s0Of R F).M i)

theorem **jW_norm** (Z : ReadL R F X0 Y) (hH : Holds R F) (i : Fin D.R.m) (x : Fin ((s0Of R F).dW i) → D.what₂.LL) : Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (Pi.basisFun D.what₂.LL _) ((s0Of R F).XW i) (InfinitePlace.mk D.what₂.embLL) x) = ‖Z.jW i x‖

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CmHostRootComplexEqPmSqrt

theorem **IsHoriz.support_eq** {T : TorModel Y} (h : IsHoriz T) : (T.BD.idealSheaf.support : Set T.Z.X) = closure (genι T.Z '' (genBd T.Z : Set (gen T.Z)))

theorem **IsStable.levelActs** {T : TorModel Y} (h : IsStable T) : T.Z.LevelActs

theorem **IsRecTor.isLB_detOm** {T : TorModel Y} (h : IsRecTor T) : (GlobalTheory.FlagMod.detOm T.Z).IsLB

theorem **nonempty_sys** (h : 1 ≤ Dw.p + Dw.q) (Xo : ∀ l : Level H, Moduli Dw l) : Nonempty (Sys Xo)

theorem **UnfFam.U_eq** (u : UnfFam Xo c N ξ) {σ : c.F →+* ℂ} (h : Formalism.L2.PjFree.GoodEmb c σ) : u.U σ = u.good σ h

theorem **UnfFam.isEmpty_comp** (u : UnfFam Xo c N ξ) {σ : c.F →+* ℂ} (h : ¬ Formalism.L2.PjFree.GoodEmb c σ) : IsEmpty (u.U σ).Comp

theorem **nonempty_comp_of_isUnf** {σ : c.F →+* ℂ} (hσ : Formalism.L2.PjFree.GoodEmb c σ) {U : Formalism.L2.PjFree.Unf Xo c σ N} (h : Formalism.L2.PjFree.IsUnf U) (hN : 1 ≤ N) : Nonempty U.Comp

theorem **UnfFam.unf** (u : UnfFam Xo c N ξ) (hN : 1 ≤ N) (σ : c.F →+* ℂ) (h : Formalism.L2.PjFree.GoodEmb c σ) : Formalism.L2.PjFree.IsUnf (u.U σ) ∧ Nonempty (u.U σ).Comp ∧ (∀ k la γ, ((u.U σ).X k).χ la γ = 1) ∧ ∀ k x, FromTest (K

theorem **UnfFam.xi_eq** (u : UnfFam Xo c N ξ) (σ : c.F →+* ℂ) (k : (u.U σ).Comp) : ((u.U σ).X k).ξ0 = ξ

theorem **nonempty_unfFam** (Xo : Moduli Dw l) (c : Coeff l) (hN : Dw.p + Dw.q = N) (h1 : 1 ≤ N) (hξ : ReciprocityTheory.ChartRestriction.G.FD.cc ξ = -ξ) (h0 : ξ ≠ 0) (him : ∀ τ : E →+* ℂ, τ (H.rt : E) = H.rt0 → (τ ξ).im < 0) : Nonempty (UnfFam Xo c N ξ)

theorem **Amb.cm** (A : Amb D) (t : D.what₂.Tau) : DV.tauEmb D t (A.H.rt : D.host.E) = A.H.rt0

theorem **Amb.eq_ambD** (A : Amb D) : A = ambD D

theorem **Amb.im_neg_xi_neg** (A : Amb D) (τ : D.host.E →+* ℂ) (hτ : τ (A.H.rt : D.host.E) = A.H.rt0) : (τ (-D.host.xi0)).im < 0

theorem **norm_embedding_xi_zero** (D : D7) (τ : D.host.E →+* ℂ) : ‖τ D.host.xi0‖ = xiR D

theorem **Amb.nonempty_unfFam** (A : Amb D) {Dw : Datum A.H} {l : Level A.H} (Xo : Moduli Dw l) (c : Coeff l) {N : ℕ} (hN : Dw.p + Dw.q = N) (h1 : 1 ≤ N) : Nonempty (UnfFam Xo c N (-D.host.xi0))

theorem **Lv.F_eq** {A : Amb D} (κ : Lv A) : κ.c.F = fieldLL D

theorem **exists_lv_iff** (A : Amb D) (l : Level A.H) : (∃ κ : Lv A, κ.l = l) ↔ (∀ p : ℕ, p.Prime → l.GoodAt p) ∧ IsLargeFor A l.M'

theorem **exists_sig** (A : Amb D) {N j : ℕ} (hN : 2 ≤ N) (Dw : Datum A.H) (hp : Dw.p = j) (hq : Dw.p + Dw.q = N) (h : ∀ Xo : ∀ l : Level A.H, Moduli Dw l, SysEx Xo) : ∃ S : Sig A N j, S.Dw = Dw

theorem **Amb.exists_datum** (A : Amb D) (p q : ℕ) : ∃ Dw : Datum A.H, Dw.p = p ∧ Dw.q = q

theorem **nonempty_src_of_sig** {A : Amb D} {N j : ℕ} (S : Sig A N j) (κ : Lv A) : Nonempty (FinalTheorems S κ)

theorem **glob** : IsGlob S.sys.Y κ.good s.Ygl

theorem **isRec** : IsRecTor s.T

theorem **unf** (σ : κ.c.F →+* ℂ) (h : Formalism.L2.PjFree.GoodEmb κ.c σ) : Formalism.L2.PjFree.IsUnf (s.U σ) ∧ Nonempty (s.U σ).Comp ∧ (∀ k la γ, ((s.U σ).X k).χ la γ = 1) ∧ ∀ k x, FromTest (K

theorem **xi_eq** (σ : κ.c.F →+* ℂ) (k : (s.U σ).Comp) : ((s.U σ).X k).ξ0 = -D.host.xi0

theorem **xiAbs** (σ : κ.c.F →+* ℂ) (k : (s.U σ).Comp) (τ : D.host.E →+* ℂ) : ‖τ ((s.U σ).X k).ξ0‖ = xiR D

theorem **gen** : GenFrEx s.Xo κ.c

theorem **geo** : FnAlgClosed s.Xo κ.c

theorem **trans** {κ₂ : Lv A} (s₂ : FinalTheorems S κ₂) (h : κ.l.Dvd κ₂.l) (b : 𝒪 κ.c.F →+* 𝒪 κ₂.c.F) (hb : ∀ x, (((b x : 𝒪 κ₂.c.F) : κ₂.c.F) : ℂ) = ((x : κ.c.F) : ℂ)) : IsTorTrans s.T s₂.T b

theorem **neg_trace_hD** (S : letI

theorem **IsDatOf.isDatW** {A : Amb D} {j : ℕ} {Dw : Datum A.H} (h : IsDatOf (letI

theorem **exists_isDatOf_of_inR** (A : Amb D) {j : ℕ} (hj : InR D j) : ∃ Dw : Datum A.H, IsDatOf (letI

theorem **κg_dvd_κs** : F.κg.l.Dvd F.κs.l

theorem **pins** : F.par.Pins

theorem **cs_le** (κ : Lv A) : F.cs.F ≤ κ.c.F

theorem **trans_Ts** (N j : ℕ) (hN : 2 ≤ N) (hj : j ≤ N) (κ : Lv A) (h : F.κg.l.Dvd κ.l) (b : 𝒪 F.cs.F →+* 𝒪 κ.c.F) (hb : ∀ x, (((b x : 𝒪 κ.c.F) : κ.c.F) : ℂ) = ((x : F.cs.F) : ℂ)) : IsTorTrans (F.Ts N j hN hj) (F.src N j hN hj κ).T b

theorem **upar_spec** (j : ℕ) (σ : F.par.c.F →+* ℂ) (hσ : GoodEmb F.par.c σ) : IsUnf (F.Upar j σ) ∧ ∀ k (τ : D.host.E →+* ℂ), ‖τ ((F.Upar j σ).X k).ξ0‖ = xiR D

theorem **exists_datum_tied** (A : Amb D) {N i : ℕ} (hi : i ≤ N) : ∃ Dw : Datum A.H, Dw.p = i ∧ Dw.p + Dw.q = N ∧ ∀ j, InR D j → N = D.R.r → i = (D.R.sigW j).1 → IsDatW D j Dw

theorem **exists_srcFam** (A : Amb D) (hS : ∀ Dw : Datum A.H, 2 ≤ Dw.p + Dw.q → ∀ Xo : ∀ l : Level A.H, Moduli Dw l, SysEx Xo) (hF : OFull D) (hz : OZeta D) (pb : Fin 2 → HeightOneSpectrum (𝒪 D.host.K)) (hpb : ∀ k, ((D.host.g k : ℕ) : 𝒪 D.host.K) ∈ (pb k).asIdeal) (ex : ℕ) (hex : 0 < ex) (hg : IsLargeFor A (D.host.g 0 * D.host.g 1)) (hs : IsLargeFor A (Nat.lcm (D.host.g 0 * D.host.g 1) (15 ^ ex))) : ∃ F : SrcFam A, F.Pinned pb ex

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CmHostUnitAddConjEqOne

theorem **exists_coef_zero** (x : FractionRing R) : ∃ s, coef U hR s 0 = x

theorem **tau_spec** : letI

theorem **adm** : AdmBase B.R (maximalIdeal B.R)

theorem **bar_eq_of_conj** (A : Amb D) {c c' : 𝒪 D.host.E} (hc : (c' : D.host.E) = D.host.conj (c : D.host.E)) : c' = A.H.bar c

theorem **cmHostUnit_add_conj_eq_one** (A : Amb D) : w0 D + A.H.bar (w0 D) = 1

theorem **nonempty_bdat** (z : ct.BPt) : Nonempty (BDat ct.Xm.t z.1)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CmQuadraticFormEqAlternatingForm

theorem **has₂_iff** (B : Base) (S₁ S₂ : Finset ℂ) : Has₂ B S₁ S₂ ↔ Has B ∅ S₁ S₂

theorem **Enl.KsC_eq** {B B' : Base} (h : Enl B B') : B'.KsC = B.KsC

theorem **Enl.fixedCoefficientField_eq** {B B' : Base} (h : Enl B B') : B'.fixedCoefficientFieldInComplex = B.fixedCoefficientFieldInComplex

theorem **finite_offSet** (B : Pre) : (offSet B).Finite

theorem **mem_offSet_of_dvd_condN** (B : Pre) {p : ℕ} (h : p ∣ B.host.condN) : p ∈ offSet B

theorem **mem_offSet_of_mem_avPrimes** (X : Stages) {p : ℕ} (hp : p ∈ avPrimes X.toD7) : p ∈ offSet X.B.toPre

theorem **TwDat.notMem_avRec** (X : Stages) (T : TwDat X.B) (w : X.B.places.Pl) {v : HeightOneSpectrum (𝒪 X.B.host.E)} (hv : v ∈ T.Q w) : v ∉ AvRec X.toD7

theorem **lgOn_self** : LgOn Sm K fun B f δ _ => Sm B.toPre f δ B.KsC

theorem **LgOn.imp** {p q : ∀ B : Base, Fr B.toPre → Δ B.toPre → TwDat B → Prop} (hp : LgOn Sm K p) (h : ∀ B f δ T, K.Over B.toPre → Sm B.toPre f δ B.KsC → p B f δ T → q B f δ T) : LgOn Sm K q

theorem **LgOn.inter** {p q : ∀ B : Base, Fr B.toPre → Δ B.toPre → TwDat B → Prop} (hp : LgOn Sm K p) (hq : LgOn Sm K q) : LgOn Sm K fun B f δ T => p B f δ T ∧ q B f δ T

theorem **large₂.lgOn** {p : ∀ B : Base, Fr B.toPre → TwDat B → Prop} (hp : large₂ K p) : LgOn Sm K fun B f _ T => p B f T

theorem **lg_iff_lgOn** {p : ∀ B : Base, Fr B.toPre → TwDat B → Prop} : large₂ K p ↔ LgOn (Δ

theorem **large₂.inter** {p q : ∀ B : Base, Fr B.toPre → TwDat B → Prop} (hp : large₂ K p) (hq : large₂ K q) : large₂ K fun B f T => p B f T ∧ q B f T

theorem **large₂.imp** {p q : ∀ B : Base, Fr B.toPre → TwDat B → Prop} (hp : large₂ K p) (h : ∀ B f T, K.Over B.toPre → p B f T → q B f T) : large₂ K q

theorem **lgRec_bounds** : LgRec K fun B f D _ => Fld.listField D f ≤ B.KsC ∧ B.KsLeOm D

theorem **recEx_iff_recExG** (QO : ∀ D : D7, Obj D → Prop) : RecEx QO ↔ RecExG (AtLed QO)

theorem **Ctx.Good.level** {K : Ctx} {C5 c6 Cb : ℝ} (h : K.Good C5 c6 Cb) : K.P.H.levelPrimes = {K.P.H.g 0, K.P.H.g 1, K.P.H.pde, K.P.H.pde', DC.q0H K.P.H}

theorem **Ctx.Good.cmQuadraticForm_eq_alternatingForm** {K : Ctx} {C5 c6 Cb : ℝ} (h : K.Good C5 c6 Cb) : K.q1 = DC.q0H K.P.H

theorem **evG_of_ev** {p : Ctx → Prop} (h : Ev p) : EvG p

theorem **ev_le_splitBound** (n : ℕ) : Ev fun K => n ≤ K.P.H.splitBound

theorem **EvG.imp** {p q : Ctx → Prop} (hp : EvG p) (hpq : ∀ (K : Ctx) (C5 c6 Cb : ℝ), K.Good C5 c6 Cb → p K → q K) : EvG q

theorem **EvG.inter** {p q : Ctx → Prop} (hp : EvG p) (hq : EvG q) : EvG fun K => p K ∧ q K

theorem **evG_good** : EvG fun K => ∃ C5 c6 Cb : ℝ, K.Good C5 c6 Cb

theorem **evLgOnG_of_evLgOn** {Δ : Pre → Type} {Sm : ∀ B : Pre, Fr B → Δ B → IntermediateField ℚ ℂ → Prop} {p : Ctx → ∀ B : Base, Fr B.toPre → Δ B.toPre → TwDat B → Prop} (h : EvLgOn Sm p) : EvLgOnG Sm p

theorem **evLgG_of_evLg** {p : Ctx → ∀ B : Base, Fr B.toPre → TwDat B → Prop} (h : EvLg p) : EvLgG p

theorem **recAtG_of_recAt** {QL : ∀ D : D7, Lay D → Prop} (h : RecAt QL) : RecAtG fun _ => QL

theorem **RecAtG.imp** {QL QL' : ∀ (_ : Ctx) (D : D7), Lay D → Prop} (h : RecAtG QL) (hQ : ∀ (K : Ctx) (C5 c6 Cb : ℝ) (D : D7) (Y : Lay D), K.Good C5 c6 Cb → Over D K.P K.Lv → RecOK K.s K.C K.cD D → LedOK K.s K.C K.cD K.ch Y K.a K.b → LocalQuadraticParts.TWAt D Y.nch Y.xd → QL K D Y → QL' K D Y) : RecAtG QL'

theorem **recExG_of_recExL** {QL : ∀ D : D7, Lay D → Prop} (h : RecExL QL) : RecExG fun _ _ _ _ _ _ _ _ => QL

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CoeIntEmbEmbP

theorem **mapLinear_coe** {R S O : Type} [CommRing R] [CommRing S] [CommRing O] (f : R →+* S) {G B : Over (Sp R)} [GrpObj G] [IsCommMonObj G] [GrpObj B] [IsCommMonObj B] {a : OAct O G} {b : OAct O B} {q : G ⟶ B} (hq : IsLinearHom a b q) (σ : O →+* S) (v : LieEig f a σ) : Lie.mapLinear (isLinearHom_pull f hq) v.1 = (eigMap f hq σ v).1

theorem **integers** : (NormedField.valuation (K

theorem **coe_intEmb_embP** (τ : E →+* ℂ) (x : 𝒪 E) : (intEmb (embP P τ) x : ɪC.S) = P.emb τ x

theorem **q_lin** : letI

theorem **i_lin** : letI

theorem **kapB_of** (h : Nonempty (Split A)) (τ : E →+* ℂ) (u u' : LieB A τ) : kapB A τ u u' = kapAb h.some (HTLat.isSplitting_embO (integers C)) (embP P τ) u.1 u'.1

theorem **kapGt_of** (h : Nonempty (Split A)) (τ : E →+* ℂ) (u u' : LieGt A τ) : kapGt A τ u u' = kapExt (isLinearHom_pull A.f (q_lin A)) h.some (HTLat.isSplitting_embO (integers C)) (embP P τ) u.1 u'.1

theorem **LG_eq** : A.U.U.LG = A.t.PP

theorem **evA_eq** {s : SecAt A} (h : ∃! v, Q.IsValA s v) {v : ↥C.S} : Q.evA s = v ↔ Q.IsValA s v

theorem **evGt_eq** {s : SecAt A} (h : ∃! v, Q.IsValG s v) {v : ↥C.S} : Q.evGt s = v ↔ Q.IsValG s v

theorem **incT_exact** (τ : E →+* ℂ) : Function.Injective (incT A τ) ∧ Function.Exact (incT A τ) (resB A τ)

theorem **kapGt_eq** (τ : E →+* ℂ) (u u' : LieGt A τ) : kapGt A τ u u' = kapB A τ (resB A τ u) (resB A τ u')

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CoefSetNonemptyOfNonneg

theorem **toLaurent_injective** : Function.Injective F.toLaurent

theorem **norm_coeff_toLaurent** (g : PowerSeries F.O) (n : ℤ) : ‖(F.toLaurent g).coeff n‖ ≤ 1

theorem **isAdicComplete** {ϖ : F.O} (h : Irreducible ϖ) : IsAdicComplete (Ideal.span {ϖ}) F.O

theorem **exists_dvf** {l : Level H} (c : Coeff l) {p : ℕ} {W : ValuationSubring ℂ} (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) : ∃ (F : DVF p) (b : c.V W →+* F.O), Function.Injective b

theorem **dir_nonempty** : t.Dir.Nonempty

theorem **isFat_dir** : IsFat t.Dir

theorem **nontrivial_forms** : Nontrivial t.Forms

theorem **fracRoot_pow** (io : c.F →+* FractionRing A) {n : ℕ} (h : n ∣ l.M') : fracRoot io h ^ n = 1

theorem **exists_io** {K : Type} [Field K] {V : ValuationSubring K} {b : V →+* A} (hb : Function.Injective b) : ∃ io : K →+* FractionRing A, io.comp V.subtype = (algebraMap A (FractionRing A)).comp b

theorem **exists_chart** {Xo : Moduli Dw l} {Y : Model Xo c (c.V W).subtype} (hY : Y.N4) {b : c.V W →+* F.O} (a : ArcIn Dw l (c.V W).subtype F.O b) : ∃ (ϖ : F.O) (χ : Sp (PowerSeries F.O) ⟶ Y.X), Irreducible ϖ ∧ IsArcChart Y a ϖ χ ∧ Nonempty (Wit Y a χ)

theorem **exists_cusp** {Xo : Moduli Dw l} {Y : Model Xo c (c.V W).subtype} (hY : Y.N4) {b : c.V W →+* F.O} (hb : Function.Injective b) (a : ArcIn Dw l (c.V W).subtype F.O b) : ∃ t : Cusp Dw l (c.V W).subtype F.O, t.b = b ∧ HEq t.a a

theorem **exists_cusp_hyp** {Xo : Moduli (HostModel.hypDatum H) l} {Y : Model Xo c (c.V W).subtype} (hY : Y.N4) (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) : ∃ F : DVF p, Nonempty (Cusp (HostModel.hypDatum H) l (c.V W).subtype F.O)

theorem **isTanAtG_id** (v : epsR S ⟶ t.G) : IsTanAtG t (𝟙 _) v.left ↔ IsTangent v

theorem **isCombAtG_id** {n : ℕ} (v : Fin n → (epsR S ⟶ t.G)) (cf : Fin n → S) (w : epsR S ⟶ t.G) : IsCombAtG t (𝟙 _) (fun i => (v i).left) cf w.left ↔ IsLieComb cf v w

theorem **isFrame_id** {l : Level H} {c : Coeff l} (g : c.F →+* S) (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (epsR S ⟶ t.G)) (h : ∀ τ, (∀ k, IsTangent (u τ k) ∧ ∀ x : 𝒪 E, (u τ k ≫ t.act.act x).left = spMap (epsScaleR (g (embK c τ x))) ≫ (u τ k).left) ∧ ∀ w : epsR S ⟶ t.G, IsTangent w → (∀ x : 𝒪 E, (w ≫ t.act.act x).left = spMap (epsScaleR (g (embK c τ x))) ≫ w.left) → ∃! cf : Fin (Dw.mult τ) → S, IsLieComb cf (u τ) w) : IsFrame (Dw

theorem **le_preimage_genO** (hp : p ≠ 0) {k : Type} [Field k] [CharZero k] (x : Sp k ⟶ Z.X) : ⊤ ≤ x ⁻¹ᵁ Z.genO

theorem **exists_isArc** (hY : (Y Z.i).N4) {γ : t.Forms} (h : γ ∈ t.Dir) : ∃ x v, IsArc t γ x v

theorem **exists_isSeries** (hp : p ≠ 0) (hY : (Y Z.i).N4) (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) {γ : t.Forms} (h : γ ∈ t.Dir) : ∃ r, IsSeries t T f γ r

theorem **CuspRec.exists_equiv** : ∃ ε : (CuspRec.of t).cusp.Forms ≃ₗ[ℤ] t.Forms, (∀ γ, ε γ ∈ t.Dir ↔ γ ∈ (CuspRec.of t).cusp.Dir) ∧ ∀ γ x v, IsArc t (ε γ) x v ↔ IsArc (CuspRec.of t).cusp γ x v

theorem **CuspRec.exists_fjNorm_eq** : ∃ ε : t.Chr ≃ₗ[ℤ] (CuspRec.of t).cusp.Chr, ∀ (e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ) (M : Z.X.Modules) (T : GrTriv Z e M) (f : Γ(M, Z.genO)) (χ : t.Chr), fjNorm (CuspRec.of t).cusp T f (ε χ) = fjNorm t T f χ

theorem **nonempty_lbTriv** (l : PsiT D → ℕ → ℤ) : Nonempty (GrTriv ct.YY (ct.expo l) (ct.Lb l).toMod)

theorem **charts_YY** : (ct.Y ct.YY.i).N4

theorem **fjN_le_iff** (B : ℝ) : (∀ s χ, (fjTrue ct).fjN l f s χ ≤ B) ↔ ∀ (F : DVF p) (t : CuspOn ct.YY F) (χ : t.Chr), fjNorm t (lbTriv ct l) f χ ≤ B

theorem **fjNorm_eq** {F : DVF p} {t : CuspOn ct.YY F} {c : t.Chr → ↥F.K} (h : IsFJ t (lbTriv ct l) f c) (χ : t.Chr) : fjNorm t (lbTriv ct l) f χ = ‖c χ‖

theorem **IsFJ.unique** {F : DVF p} {t : CuspOn ct.YY F} {c c' : t.Chr → ↥F.K} (h : IsFJ t (lbTriv ct l) f c) (h' : IsFJ t (lbTriv ct l) f c') : c = c'

theorem **coefSet_nonempty_of_nonneg** {F : DVF p} (t : CuspOn ct.YY F) {χ : t.Chr} (h : Nonneg t.Dir χ) : (coefSet t.Dir (IsSeries t (lbTriv ct l) f) χ).Nonempty

theorem **fjN_eq_zero** (s : (fjTrue ct).FJd) (χ : (fjTrue ct).Chr s) (γ : s.cusp.Forms) (hγ : ∀ y, y ≠ 0 → 0 < (γ : LinearMap.BilinForm ℤ s.cusp.a.D₀.X) y y) (h : (show s.cusp.Chr from χ) γ < 0) : (fjTrue ct).fjN l f s χ = 0

theorem **fjN_le_one** (h : ∀ (F : DVF p) (t : CuspOn ct.YY F) γ r, IsSeries t (lbTriv ct l) f γ r → r ∈ F.toLaurent.range) (s : (fjTrue ct).FJd) (χ : (fjTrue ct).Chr s) : (fjTrue ct).fjN l f s χ ≤ 1

theorem **nonempty_fjd** {F : DVF p} {b : ct.YY.i.c.V ct.W →+* F.O} (hb : Function.Injective b) (a : ArcIn S.Dw ct.YY.i.l (ct.YY.i.c.V ct.W).subtype F.O b) : Nonempty (fjTrue ct).FJd

theorem **nonempty_fjd_hyp** (h : S.Dw = HostModel.hypDatum A.H) : Nonempty (fjTrue ct).FJd

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CoeffSpecAlgebraMapOfNeg

theorem **prod_X_ne_zero** : (∏ i, MvPowerSeries.X i : MvPowerSeries (Fin r) K) ≠ 0

theorem **powers_le** : Submonoid.powers (∏ i, MvPowerSeries.X i : MvPowerSeries (Fin r) K) ≤ nonZeroDivisors _

theorem **algebraMap_injective** : Function.Injective (algebraMap (MvPowerSeries (Fin r) K) (MultiRing K r))

theorem **isUnit_u** (i : Fin r) : IsUnit (u i : MultiRing K r)

theorem **exists_mul_pow** (Φ : MultiRing K r) : ∃ (n : ℕ) (f : MvPowerSeries (Fin r) K), Φ * (∏ i, u i) ^ n = algebraMap (MvPowerSeries (Fin r) K) (MultiRing K r) f

theorem **hasSubst** : MvPowerSeries.HasSubst fun i => (PowerSeries.X ^ a i : PowerSeries K)

theorem **specPS_apply** (f : MvPowerSeries (Fin r) K) : specPS a ha f = MvPowerSeries.subst (fun i => (PowerSeries.X ^ a i : PowerSeries K)) f

@[simp] theorem **specPS_X** (i : Fin r) : specPS a ha (MvPowerSeries.X i) = (PowerSeries.X ^ a i : PowerSeries K)

@[simp] theorem **specPS_C** (c : K) : specPS a ha (MvPowerSeries.C c) = PowerSeries.C c

theorem **finite_weight** (n : ℕ) : {m : Fin r →₀ ℕ | ∑ i, a i * m i = n}.Finite

theorem **coeff_specPS** (f : MvPowerSeries (Fin r) K) (n : ℕ) : PowerSeries.coeff n (specPS a ha f) = ∑ m ∈ (finite_weight a ha n).toFinset, MvPowerSeries.coeff m f

@[simp] theorem **spec_algebraMap** (f : MvPowerSeries (Fin r) K) : spec a ha (algebraMap (MvPowerSeries (Fin r) K) (MultiRing K r) f) = HahnSeries.ofPowerSeries ℤ K (specPS a ha f)

@[simp] theorem **spec_u** (i : Fin r) : spec a ha (u i : MultiRing K r) = HahnSeries.single (a i : ℤ) 1

theorem **spec_comp_ofLaurent** (a : Fin r → ℕ) (ha : ∀ i, 1 ≤ a i) (i : Fin r) : (spec a ha).comp (ofLaurent i) = powLS (K

theorem **powLS_one** : powLS (K

theorem **spec_one_bijective** : Function.Bijective (spec (K

theorem **sum_mem_dir** (h : IsMultiArc t δ y w) (a : Fin r → ℕ) (ha : ∀ i, 1 ≤ a i) : ∑ i, a i • δ i ∈ t.Dir

theorem **pos** (h : IsMultiArc t δ y w) : 0 < r

theorem **exists_isFrame_of_flagAt** {k : Type} [Field k] {x : Sp k ⟶ Z.X} {v : Vecs Dw (Y Z.i).t k} (h : Z.FlagAt x v) : ∃ (g : Z.i.c.F →+* k) (y₀ : Sp k ⟶ (Xo Z.i.l).XF Z.i.c), y₀ ≫ pullback.snd _ _ = spMap g ∧ y₀ ≫ (Y Z.i).j = x ≫ Z.π ∧ IsFrame (Y Z.i).t g (x ≫ Z.π) v

theorem **IsArc.isMultiArc_single** {γ : t.Forms} {x : Sp (LaurentSeries ↥F.K) ⟶ Z.X} {v : Vecs Dw (Y Z.i).t (LaurentSeries ↥F.K)} (hx : IsArc t γ x v) (i : Fin r) : IsMultiArc t (Pi.single i γ) (spMap (MultiRing.ofLaurent i) ≫ x ≫ Z.π) (v.bc (MultiRing.ofLaurent i))

theorem **exists_isMultiArc_single** (hY : (Y Z.i).N4) {γ : t.Forms} (h : γ ∈ t.Dir) (i : Fin r) : ∃ y w, IsMultiArc t (Pi.single i γ) y w

theorem **exists_isMultiArc_one_iff** (hY : (Y Z.i).N4) (δ : Fin 1 → t.Forms) : (∃ y w, IsMultiArc t δ y w) ↔ δ 0 ∈ t.Dir

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ColDictIsomReTr

theorem **measTie_reX** {X : HidaSetting D} {hb : Formalism.L2.SG.AdBridge.AdBridgeIso X.B.I} (h : X.MeasTie hb) : (reX Z X).MeasTie hb

theorem **measCompat_reX** {X : HidaSetting D} {ψ : AddChar (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.AF X.B.I.F) ℂ} (h : Formalism.L2.SG.MassDataConc.MeasCompatψ X ψ) : Formalism.L2.SG.MassDataConc.MeasCompatψ (reX Z X) ψ

theorem **locCompat_reX** {X : HidaSetting D} {ψ : AddChar (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.AF X.B.I.F) ℂ} {w : D.what₂.Word} (h : Formalism.L2.SG.LocVol.LocCompatX X ψ w) : Formalism.L2.SG.LocVol.LocCompatX (reX Z X) ψ (wd Z w)

theorem **colDict_congr** {X : HidaSetting D} {u : D.what₂.Word} {F : Forms0 D u} (T : Transport X u F) (hK : X.LevelSubgroupCharacter) (L : Cu X u hK →ₗ[ℂ] ((D.mem u).C.cls 0 → (D.mem u).C.Y 0 → ℂ)) (hL : ∀ x, L x = colDictL X T hK x) : (ColDictInto X T hK → ∀ x, L x ∈ (D.mem u).A.Mu0) ∧ (ColDictBij X T hK → Function.Injective L ∧ ∀ φ ∈ (D.mem u).A.Mu0, ∃ x, L x = φ) ∧ ∀ {what₂ : Finset (Submodule (𝒪 D.host.E) (VS D).V)} (cs : ColSys X u what₂) (hinj : Function.Injective (sharp hK cs)), ColDictIsom X T hK cs hinj → ∀ x y : ColT hinj, inner ℂ x y = (D.fm u).ipC 0 (L (toCu X hK x)) (L (toCu X hK y))

theorem **colDictInto_reTr** (h : ColDictInto X T Y.hK) : ColDictInto (reX Z X) (reTr Z T hy he) (reY Z Y).hK

theorem **colDictBij_reTr** (h : ColDictBij X T Y.hK) : ColDictBij (reX Z X) (reTr Z T hy he) (reY Z Y).hK

theorem **colDictIsom_reTr** (hF : FinX Y) (hF' : FinX (reY Z Y)) (h : ColDictIsom X T Y.hK (Y.cs u) (hF.shInj u)) : ColDictIsom (reX Z X) (reTr Z T hy he) (reY Z Y).hK ((reY Z Y).cs u) (hF'.shInj u)

theorem **levelMatch_reTr** {X : HidaSetting D} {u : D.what₂.Word} {F : Forms0 D u} {F' : Forms0 (reD D Z) u} (T : Transport X u F) (hl : F'.cls0.lev = F.cls0.lev) (hy : F'.cls0.y = F.cls0.y) (he : F'.val0.eY = F.val0.eY) (Y : Full X) (h : LevelMatch X T Y.hK) : LevelMatch (reX Z X) (reTr Z T hy he) (reY Z Y).hK

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ColDictLReTr

theorem **colDictL_reTr** {X : HidaSetting D} {u : D.what₂.Word} {F : Forms0 D u} {F' : Forms0 (reD D Z) u} (T : Transport X u F) (hy : F'.cls0.y = F.cls0.y) (he : F'.val0.eY = F.val0.eY) (hK : X.LevelSubgroupCharacter) (x : Cu X u hK) : colDictL (reX Z X) (reTr Z T hy he) hK x = colDictL X T hK x

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ComplexSidePlacesIffMemNonzeroIndexPlaces

theorem **restrict_Xg_tr_core** (hH : Holds z.R z.F) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : (z.restrict hH Wt' h hc).Xg = z.Xg ∧ (z.restrict hH Wt' h hc).tr = z.tr ∧ (z.restrict hH Wt' h hc).core = z.core

theorem **restrict_R** (hH : Holds z.R z.F) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : (z.restrict hH Wt' h hc).R = z.resR h

theorem **restrictRealData_constant_error** (h : Wt' ⊆ Wt) : (z.resR h).kap = z.R.kap ∧ (z.resR h).err0 = z.R.err0

theorem **restrictRealData_lift_scalar** (h : Wt' ⊆ Wt) (i : Fin X.toD7.R.m) (t : TupU I (usedOf I T Wt')) : (z.resR h).lam3 i t = z.R.lam3 i (inclU I (usedOf_mono I T h) t)

theorem **resR_mu_twPl** (h : Wt' ⊆ Wt) : (z.resR h).mu = z.resMu h ∧ (z.resR h).twPl = z.resTw Wt'

theorem **restrict_holds** (hH : Holds z.R z.F) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : Holds (z.restrict hH Wt' h hc).R (z.restrict hH Wt' h hc).F

theorem **inGeo_resGeo_iff** (h : Wt' ⊆ Wt) : GlobalTheory.RealGeo.InGeo (z.resGeo h) ↔ GlobalTheory.RealGeo.InGeo z.geo

theorem **restrict_inGeo_iff** (hH : Holds z.R z.F) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : GlobalTheory.RealGeo.InGeo (z.restrict hH Wt' h hc).geo ↔ GlobalTheory.RealGeo.InGeo z.geo

theorem **mem_nonzero_index_places_iff_at_place_or_conjugate** (w : X.B.places.Pl) : w ∈ z.W1 F0 ↔ ∃ j, NzAt z.R z.F F0 j (plB X w) ∨ NzAt z.R z.F F0 j (Formalism.DivisionValues.conjPl (plB X w))

theorem **nonzero_index_places_conj_stable** : ConjStable X (z.W1 F0)

theorem **nzAt_resR** (h : Wt' ⊆ Wt) (j : Fin X.toD7.R.m) (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : NzAt (z.resR h) (face0 X I hX (usedOf I T Wt') z.Xg z.tr) F0 j v ↔ NzAt z.R (face0 X I hX (usedOf I T Wt) z.Xg z.tr) F0 j v

theorem **restrict_nonzero_places_eq** (hH : Holds z.R z.F) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : (z.restrict hH Wt' h hc).W1 F0 = z.W1 F0

theorem **complex_side_places_iff_mem_nonzero_index_places** (kap lam err) (Dc Sp Zsc) (N : IndefNum (famR X I hX (usedOf I T Wt) z.Xg z.tr).fam3.base3) (hN : N.F0 = F0) (w : X.B.places.Pl) : (((z.R.updS kap lam).withErr err).toCx3 (reFace z.F Dc Sp Zsc)).W1 (reFace z.F Dc Sp Zsc).cr N (X.toD7.plEq w) ↔ w ∈ z.W1 F0

theorem **complex_side_face_numbers_iff_mem_nonzero_index_places** (kap lam err) (Dc Sp Zsc) (C CH : ℝ) (hZ : (Zsc (zsOf z.R z.F)).F0 = F0) (w : X.B.places.Pl) : (((z.R.updS kap lam).withErr err).toCx3 (reFace z.F Dc Sp Zsc)).W1 (reFace z.F Dc Sp Zsc).cr (((z.R.updS kap lam).withErr err).num (reFace z.F Dc Sp Zsc) C CH) (X.toD7.plEq w) ↔ w ∈ z.W1 F0

theorem **twisted_places_eq_complex_side_of_eq_nonzero_index** (hW : z.W1 F0 = Wt) (kap lam err) (Dc Sp Zsc) (C CH : ℝ) (hZ : (Zsc (zsOf z.R z.F)).F0 = F0) (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : v ∈ ((z.R.updS kap lam).withErr err).twPl ↔ (((z.R.updS kap lam).withErr err).toCx3 (reFace z.F Dc Sp Zsc)).W1 (reFace z.F Dc Sp Zsc).cr (((z.R.updS kap lam).withErr err).num (reFace z.F Dc Sp Zsc) C CH) v

theorem **restrictNonzero_nonzero_places** (hH : Holds z.R z.F) (h : z.W1 F0 ⊆ Wt) : (z.restrictNonzero F0 hH h).W1 F0 = z.W1 F0

theorem **restrictNonzero_mem_twisted_iff** (hH : Holds z.R z.F) (h : z.W1 F0 ⊆ Wt) (w : X.B.places.Pl) : X.toD7.plEq w ∈ (z.restrictNonzero F0 hH h).R.twPl ↔ w ∈ (z.restrictNonzero F0 hH h).W1 F0

theorem **restrictNonzero_holds** (hH : Holds z.R z.F) (h : z.W1 F0 ⊆ Wt) : Holds (z.restrictNonzero F0 hH h).R (z.restrictNonzero F0 hH h).F

theorem **restrictNonzero_geometry_iff** (hH : Holds z.R z.F) (h : z.W1 F0 ⊆ Wt) : GlobalTheory.RealGeo.InGeo (z.restrictNonzero F0 hH h).geo ↔ GlobalTheory.RealGeo.InGeo z.geo

theorem **restrictNonzero_twisted_iff_complex** (hH : Holds z.R z.F) (h : z.W1 F0 ⊆ Wt) (kap lam err) (Dc Sp Zsc) (C CH : ℝ) (hZ : (Zsc (zsOf (z.restrictNonzero F0 hH h).R (z.restrictNonzero F0 hH h).F)).F0 = F0) (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : v ∈ (((z.restrictNonzero F0 hH h).R.updS kap lam).withErr err).twPl ↔ ((((z.restrictNonzero F0 hH h).R.updS kap lam).withErr err).toCx3 (reFace (z.restrictNonzero F0 hH h).F Dc Sp Zsc)).W1 (reFace (z.restrictNonzero F0 hH h).F Dc Sp Zsc).cr ((((z.restrictNonzero F0 hH h).R.updS kap lam).withErr err).num (reFace (z.restrictNonzero F0 hH h).F Dc Sp Zsc) C CH) v

theorem **restrictE_Xg_tr_core** : (z.restrictE e Wt' h hc).Xg = z.Xg ∧ (z.restrictE e Wt' h hc).tr = z.tr ∧ (z.restrictE e Wt' h hc).core = z.core

theorem **restrictE_R** : (z.restrictE e Wt' h hc).R = (z.resR h).updS (Res.kapNew (z.resR h)) (z.resR h).lam3

theorem **restrictOfExists_lift_scalar** (i : Fin X.toD7.R.m) (t : TupU I (usedOf I T Wt')) : (z.restrictE e Wt' h hc).R.lam3 i t = z.R.lam3 i (inclU I (usedOf_mono I T h) t)

theorem **restrictOfExists_error_tuples_twisted** : (z.restrictE e Wt' h hc).R.err0 = z.R.err0 ∧ (z.restrictE e Wt' h hc).R.mu = z.resMu h ∧ (z.restrictE e Wt' h hc).R.twPl = z.resTw Wt'

theorem **restrictE_R_of_thmB** (hB : z.R.PThmB) : (z.restrictE e Wt' h hc).R = z.resR h

theorem **restrictE_kap** (hB : z.R.PThmB) : (z.restrictE e Wt' h hc).R.kap = z.R.kap

theorem **restrictE_R_eq_restrict** (hH : Holds z.R z.F) : (z.restrictE e Wt' h hc).R = (z.restrict hH Wt' h hc).R

theorem **restrictE_exAt** : (z.restrictE e Wt' h hc).R.toRealDat.ExAt

theorem **restrictE_holds** (hH : Holds z.R z.F) : Holds (z.restrictE e Wt' h hc).R (z.restrictE e Wt' h hc).F

theorem **restrictE_inGeo_iff** : GlobalTheory.RealGeo.InGeo (z.restrictE e Wt' h hc).geo ↔ GlobalTheory.RealGeo.InGeo z.geo

theorem **restrictOfExists_nonzero_places** : (z.restrictE e Wt' h hc).W1 F0 = z.W1 F0

theorem **restrict_exAt** (hH : Holds z.R z.F) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : (z.restrict hH Wt' h hc).R.toRealDat.ExAt

theorem **restrictNonzeroOfExists_characters_eq** : (z.restrictNonzeroOfExists F0 e h).Xg = z.Xg ∧ (z.restrictNonzeroOfExists F0 e h).tr = z.tr ∧ (z.restrictNonzeroOfExists F0 e h).core = z.core

theorem **restrictNonzeroOfExists_nonzero_places** : (z.restrictNonzeroOfExists F0 e h).W1 F0 = z.W1 F0

theorem **restrictNonzeroOfExists_mem_twisted_iff** (w : X.B.places.Pl) : X.toD7.plEq w ∈ (z.restrictNonzeroOfExists F0 e h).R.twPl ↔ w ∈ (z.restrictNonzeroOfExists F0 e h).W1 F0

theorem **restrictNonzeroOfExists_exists_lifts** : (z.restrictNonzeroOfExists F0 e h).R.toRealDat.ExAt

theorem **restrictNonzeroOfExists_holds** (hH : Holds z.R z.F) : Holds (z.restrictNonzeroOfExists F0 e h).R (z.restrictNonzeroOfExists F0 e h).F

theorem **restrictNonzeroOfExists_geometry_iff** : GlobalTheory.RealGeo.InGeo (z.restrictNonzeroOfExists F0 e h).geo ↔ GlobalTheory.RealGeo.InGeo z.geo

theorem **restrictNonzeroOfExists_twisted_iff_complex** (kap lam err) (Dc Sp Zsc) (C CH : ℝ) (hZ : (Zsc (zsOf (z.restrictNonzeroOfExists F0 e h).R (z.restrictNonzeroOfExists F0 e h).F)).F0 = F0) (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : v ∈ (((z.restrictNonzeroOfExists F0 e h).R.updS kap lam).withErr err).twPl ↔ ((((z.restrictNonzeroOfExists F0 e h).R.updS kap lam).withErr err).toCx3 (reFace (z.restrictNonzeroOfExists F0 e h).F Dc Sp Zsc)).W1 (reFace (z.restrictNonzeroOfExists F0 e h).F Dc Sp Zsc).cr ((((z.restrictNonzeroOfExists F0 e h).R.updS kap lam).withErr err).num (reFace (z.restrictNonzeroOfExists F0 e h).F Dc Sp Zsc) C CH) v

theorem **conjStable** : ConjStable X Wt

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ConcreteLevelAdelicSubset

theorem **mem_conjRows_iff** (X : HidaSetting D) (f : L2X X) : f ∈ conjRows X ↔ conjL (muX X) f ∈ HHX X

theorem **EEC_le_conjRows** (X : HidaSetting D) (G : GoodOp X) : EEC X G ≤ conjRows X

theorem **evalTieT_iff_evalOn** (R : RealDat D A Ks) (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (G : GoodOp X) (t : GA D (D.sp.Y 0)) (c : ℝ) (E : Submodule ℂ (MMap (MP R))) : EvalTieT (MP R) R.ν0 R.fd0 E X G (iotaX X h).toMulEquiv t c ↔ ∃ v₀ : (MP R).Vec, (∀ τ, ‖(v₀ τ : (MP R).H τ)‖ = 1) ∧ EvalOn R X h t v₀ c E (EEC X G)

theorem **SeeAt.levelMatchA** {R : RealDat D A Ks} {F : Face D A} {X : HidaSetting D} {h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D} {t : GA D (D.sp.Y 0)} (hs : SeeAt R F X h t) : LevelMatchA R F X h t

theorem **exists_rowJ_of_evalOn** {R : RealDat D A Ks} {F : Face D A} {X : HidaSetting D} {h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D} {G : GoodOp X} {t : GA D (D.sp.Y 0)} {v₀ : (MP R).Vec} {c : ℝ} {eE : D.what₂.EE →ₗ[ℂ] R.HE.H} (hc : 0 < c) (hv₀ : ∀ τ, ‖(v₀ τ : (MP R).H τ)‖ = 1) (hE : R.models_HE) (hrange : LinearMap.range eE = (R.EEat (R.levR F true)).comap R.HE.e) (hnorm : ∀ x, ‖eE x‖ ^ 2 = RealDat.nP F true * ‖x‖ ^ 2) (hle : R.EEat (R.levR F true) ≤ R.EEat R.triv) (hJ2 : EvalOn R X h t v₀ c (EP R F) (EEC X G)) : ∃ j : D.what₂.EE ≃ₗᵢ[ℂ] EEC X G, RowJ R eE X h G t j

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CongruenceLevelEqDepthSucc

theorem **lev_ext** {l l' : Lev κ K n} (e : l.H = l'.H) (e' : ∀ a h h', l.lam (a, h) = l'.lam (a, h')) : l = l'

theorem **mem_levDiag** (a : GL (Fin N) κ) : a ∈ (levDiag h l).H ↔ ∃ g ∈ l.H, diagEmb κ h g = a

theorem **diagEmb_mem_levDiag** (g : GL (Fin n) κ) : diagEmb κ h g ∈ (levDiag h l).H ↔ g ∈ l.H

theorem **levDiag_lam** {a : GL (Fin N) κ} {g : GL (Fin n) κ} (e : a = diagEmb κ h g) (ha : a ∈ (levDiag h l).H) (hg : g ∈ l.H) : (levDiag h l).lam ⟨a, ha⟩ = l.lam ⟨g, hg⟩

theorem **firstBlock_diagEmb_eq** {r : ℕ} (hr : r ≤ n) (bl : List ℕ) (g : GL (Fin n) κ) (hg : g ∈ blockUT n κ r bl) (hg' : diagEmb κ h g ∈ blockUT N κ r bl) : blk0 N κ (hr.trans h) bl ⟨_, hg'⟩ = blk0 n κ hr bl ⟨g, hg⟩

theorem **diagEmb_mem_blockUT'** {r : ℕ} (bl : List ℕ) {g : GL (Fin n) κ} (hg : g ∈ blockUT n κ r bl) : diagEmb κ h g ∈ blockUT N κ r bl

theorem **levDiag_HPar** {r t m : ℕ} (hN : N = n + 1) (hn : n = r + m * t) (hr : r ≤ n) (hr' : r ≤ N) (H₀ : Subgroup (GL (Fin r) κ)) (lam₀ : H₀ →* Kˣ) (χ : ℕ → κˣ →* Kˣ) : levDiag h ((HPar n κ hr (List.replicate m t) H₀, lamPar hr _ lam₀ χ) : Lev κ K n) = ⟨HPar N κ hr' (List.replicate m t) H₀ ⊓ fixCol N κ (r + m * t), (lamPar hr' _ lam₀ χ).comp (Subgroup.inclusion inf_le_left)⟩

theorem **sup_prinCong_le** : (levRed P l).H.map (extI hN P) ⊔ prinCong N P (c + 1) ≤ (levRed P (levDiag (by omega : n ≤ N) l)).H

theorem **mem_levCong** (k : GLint E N P) : k ∈ (levCong hN P c l).H ↔ ∃ j ∈ (levRed P l).H, (extI hN P j)⁻¹ * k ∈ prinCong N P (c + 1)

theorem **inv_mul_mem_prinCong** (a k : GLint E N P) : a⁻¹ * k ∈ prinCong N P c ↔ ∀ x ∈ stdAt E N P, ((qOf P : E) ^ c)⁻¹ • (k.1 x - a.1 x) ∈ stdAt E N P

theorem **mem_levCong_iff** (k : GLint E N P) : k ∈ (levCong hN P c l).H ↔ ∃ j ∈ (levRed P l).H, ∀ x ∈ stdAt E N P, ((qOf P : E) ^ (c + 1))⁻¹ • (k.1 x - sumE hN j.1 1 x) ∈ stdAt E N P

theorem **prinCong_le_levCong** : prinCong N P (c + 1) ≤ (levCong hN P c l).H

theorem **extI_mem_levCong** (j : GLint E n P) : extI hN P j ∈ (levCong hN P c l).H ↔ j ∈ (levRed P l).H

theorem **levCong_lam** {k : GLint E N P} (hk : k ∈ (levCong hN P c l).H) {j : GLint E n P} (hj : j ∈ (levRed P l).H) (h : (extI hN P j)⁻¹ * k ∈ prinCong N P (c + 1)) : (levCong hN P c l).lam ⟨k, hk⟩ = (levRed P l).lam ⟨j, hj⟩

theorem **levCong_lam_extI** {j : GLint E n P} (hk : extI hN P j ∈ (levCong hN P c l).H) (hj : j ∈ (levRed P l).H) : (levCong hN P c l).lam ⟨_, hk⟩ = (levRed P l).lam ⟨j, hj⟩

theorem **levCong_lam_prinCong** {k : GLint E N P} (h : k ∈ prinCong N P (c + 1)) : (levCong hN P c l).lam ⟨k, prinCong_le_levCong hN c l h⟩ = 1

theorem **levCong_last** {k : GLint E N P} (hk : k ∈ (levCong hN P c l).H) {x : Sp E N} (hx : x ∈ stdAt E N P) : IntAt P (((qOf P : E) ^ (c + 1))⁻¹ * (k.1 x (Br.iL hN) - x (Br.iL hN)))

theorem **levCong_root** : ∃ m : ℕ, 0 < m ∧ ∀ k, (levCong hN P c l).lam k ^ m = 1

theorem **pref_prefRep** {G D : Type} [Group G] [MulAction G D] (pref : D → Prop) {p : D} (hp : pref p) : pref (prefRep pref (Quotient.mk (MulAction.orbitRel G D) p))

theorem **N_succ_of_eq** {j : ℕ} (h : j = R.m) : R.N j = (R.N R.m - 1) + 1

theorem **topDep_succ** (M : ℕ) : topDep R M + 1 = max (4 * R.m + 1) (2 * R.m + 2 * M)

theorem **topDep_succ_eq_default** (M : ℕ) : topDep R M + 1 = (default : TopDepth R M).1

theorem **two_mul_le_topDep** (M : ℕ) : 2 * M ≤ topDep R M

theorem **topLevAt_levG_m** (c : ℕ) : (topLevAt Ch sel mu P c).levG R.m = levGTop Ch P

theorem **topLevAt_levQ_m** (c : ℕ) : (topLevAt Ch sel mu P c).levQ R.m = levQTop R sel mu c

theorem **topLevAt_finiteIndex** (c j : ℕ) (Q : QE sel (mu : Chr X.F X.E X.K)) : ((topLevAt Ch sel mu P c).levQ j Q).H.FiniteIndex

theorem **topLevAt_root** (c j : ℕ) (Q : QE sel (mu : Chr X.F X.E X.K)) : ∃ m : ℕ, 0 < m ∧ ∀ k, ((topLevAt Ch sel mu P c).levQ j Q).lam k ^ m = 1

theorem **topLev_levG_m** : (topLev Ch sel mu cst P).levG R.m = levGTop Ch P

theorem **topLev_levQ_m** : (topLev Ch sel mu cst P).levQ R.m = levQTop R sel mu (topDep R cst.M)

theorem **topLev_pref_m** (u : Pr.Word) (p : ChPt (R

theorem **topLev_pref_ne** (u : Pr.Word) {j : ℕ} (h : j ≠ R.m) (p : ChPt (R

theorem **topLev_finiteIndex** (j : ℕ) (Q : QE sel (mu : Chr X.F X.E X.K)) : ((topLev Ch sel mu cst P).levQ j Q).H.FiniteIndex

theorem **topLev_root** (j : ℕ) (Q : QE sel (mu : Chr X.F X.E X.K)) : ∃ m : ℕ, 0 < m ∧ ∀ k, ((topLev Ch sel mu cst P).levQ j Q).lam k ^ m = 1

theorem **mem_levGTop** (w : Pr.vGs) (a : GL (Fin (R.N R.m)) (𝒪 X.E ⧸ w.1.asIdeal)) : a ∈ (levGTop Ch P w).H ↔ ∃ g ∈ (rowTyG Ch P w).H, diagEmb _ (Nat.sub_le _ 1) g = a

theorem **mem_levGTop_grp** (a : GLG X.E (R.N R.m) Pr.vGs) : a ∈ (levGTop Ch P).grp ↔ ∃ h ∈ (rowTyG Ch P).grp, extGH (N_m_succ R) h = a

theorem **levGTop_eq** (w : Pr.vGs) : levGTop Ch P w = ⟨HPar (R.N R.m) _ ((r_le_X' R).trans (Nat.sub_le _ 1)) (List.replicate R.m R.t) (P w).H ⊓ fixCol _ _ (R.r + R.m * R.t), (lamPar _ _ (P w).lam (chiAt Ch w)).comp (Subgroup.inclusion inf_le_left)⟩

theorem **levGTop_PsymF** (w : Pr.vGs) : levGTop Ch (rPsymF Ch) w = ⟨HXmsym _ (R.N R.m) R.k R.r R.t R.m R.rS ((r_le_X' R).trans (Nat.sub_le _ 1)), lamXmsym (Ch.psi0 w) (chiAt Ch w) (R.N R.m) R.k R.r R.t R.m R.rS _⟩

theorem **mem_levQTop** (c : ℕ) (Q : QE sel (mu : Chr X.F X.E X.K)) (k : GLint X.E (R.N R.m) Q.1) : k ∈ (levQTop R sel mu c Q).H ↔ ∃ h ∈ (levQX R sel mu Q).H, ∀ x ∈ stdAt X.E (R.N R.m) Q.1, ((qOf Q.1 : X.E) ^ (c + 1))⁻¹ • (k.1 x - sumE (N_m_succ R) h.1 1 x) ∈ stdAt X.E (R.N R.m) Q.1

theorem **levQTop_lam** (c : ℕ) (Q : QE sel (mu : Chr X.F X.E X.K)) (k : (levQTop R sel mu c Q).H) (h : (levQX R sel mu Q).H) (e : ∀ x ∈ stdAt X.E (R.N R.m) Q.1, ((qOf Q.1 : X.E) ^ (c + 1))⁻¹ • (k.1.1 x - sumE (N_m_succ R) h.1.1 1 x) ∈ stdAt X.E (R.N R.m) Q.1) : (levQTop R sel mu c Q).lam k = (levQX R sel mu Q).lam h

theorem **extQ_mem_levQTop** (c : ℕ) (a : GLQ X.E (R.N R.m - 1) (QE sel (mu : Chr X.F X.E X.K))) : extQ X.E (N_m_succ R) _ a ∈ (levQTop R sel mu c).grp ↔ a ∈ (levQX R sel mu).grp

theorem **levQTop_eq_topLevQ'** {c c' : ℕ} (e : c + 1 = c') (hc : 0 < c') : levQTop R sel mu c = topLevQ R sel mu hc

theorem **topLev_levG_m_eq** : (topLev Ch sel mu cst P).levG R.m = topLevG Ch P

theorem **topLev_levQ_m_eq** : (topLev Ch sel mu cst P).levQ R.m = topLevQ R sel mu (default : TopDepth R cst.M).pos

theorem **topLevAt_ok** (c : ℕ) : (topLevAt Ch sel mu P c).Ok where

theorem **topLev_ok** : (topLev Ch sel mu cst P).Ok

theorem **extGH_mem_topLev** (h : (rowTyG Ch P).grp) : extGH (N_m_succ R) h.1 ∈ ((topLev Ch sel mu cst P).levG R.m).grp

theorem **extQ_mem_topLev** (c : (levQX R sel mu).grp) : extQ X.E (N_m_succ R) _ c.1 ∈ ((topLev Ch sel mu cst P).levQ R.m).grp

theorem **incl_extTop** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) : FrmG.incl _ _ (extTop Ch sel mu cst P g) = (extUH (N_m_succ R) g.1, extGH (N_m_succ R) g.2.1.1, extQ X.E (N_m_succ R) _ g.2.2.1)

theorem **extTop_injective** : Function.Injective (extTop Ch sel mu cst P)

theorem **chrG_extTop** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) : ((topLev Ch sel mu cst P).levG R.m).chr (extTop Ch sel mu cst P g).2.1 = (rowTyG Ch P).chr g.2.1

theorem **chrQ_extTop** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) : ((topLev Ch sel mu cst P).levQ R.m).chr (extTop Ch sel mu cst P g).2.2 = (levQX R sel mu).chr g.2.2

theorem **chr_extTop** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) : ((topLev Ch sel mu cst P).levG R.m).chr (extTop Ch sel mu cst P g).2.1 * ((topLev Ch sel mu cst P).levQ R.m).chr (extTop Ch sel mu cst P g).2.2 = (stepX Ch sel mu P).chr () g

theorem **tilF_extTop** (g : FrmG (rowTyG Ch P) (levQX R sel mu)) (p : XPt Pr R sel mu) : tilF sel mu u (g • p) = extTop Ch sel mu cst P g • tilF sel mu u p

theorem **levGAt_of_le** (Tp : TopLev X Pr R (QE sel (mu : Chr X.F X.E X.K))) {j : ℕ} (h : R.m ≤ j) (b : Idx sel mu L j) : levGAt sel mu L Tp j b = Tp.levG j

theorem **levQAt_of_le** (Tp : TopLev X Pr R (QE sel (mu : Chr X.F X.E X.K))) {j : ℕ} (h : R.m ≤ j) (b : Idx sel mu L j) : levQAt sel mu L Tp j b = Tp.levQ j

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ContinuousAndSmul

theorem **polMat_form** {b : Matrix (Fin d) (Fin d) ℂ} (hb : b.IsHermitian) : polMat (fun ξ => (∑ k, ∑ l, b k l * ξ k * starRingEnd ℂ (ξ l)).re) = b

theorem **chris_vecMul** {b : V d → Matrix (Fin d) (Fin d) ℂ} {z : V d} (h : IsUnit (b z).det) (ξ : V d) : (christoffel b z ξ).ofLp ᵥ* b z = ξ.ofLp ᵥ* Matrix.of fun k l => del (fun v => b v k l) z ξ

theorem **jet_zero** : jet b 0 = Prod.fst

theorem **jet_succ** (n : ℕ) (q : V d × V d) : jet b (n + 1) q = del (jet b n) q (q.2, -christoffel b q.1 q.2)

theorem **jet_one** : jet b 1 = Prod.snd

theorem **jet_two** (q : V d × V d) : jet b 2 q = -christoffel b q.1 q.2

theorem **jet_const** (b : Matrix (Fin d) (Fin d) ℂ) (n : ℕ) : jet (fun _ => b) (n + 2) = fun _ => 0

theorem **cutoff_nonneg** (t : ℝ) : 0 ≤ cutoff t

theorem **cutoff_pos_iff** {t : ℝ} : 0 < cutoff t ↔ t < 1

theorem **mean_comp** (q f : (Fin d → ℂ) → ℝ) (T : (Fin d → ℂ) ≃ᵐ[ℝ] (Fin d → ℂ)) : mean (q ∘ T) (f ∘ T) = mean q f

theorem **cutoff_le** (t : ℝ) : cutoff t ≤ 2

theorem **measurable_cutoff** : Measurable cutoff

theorem **isBounded_ball** : Bornology.IsBounded {ξ | q ξ < 1}

theorem **integrable_cutoff** : Integrable fun ξ => cutoff (q ξ)

theorem **mass_pos** : 0 < ∫ ξ, cutoff (q ξ)

theorem **mean_const** (r : ℝ) : mean q (fun _ => r) = r

theorem **isRegConst_const** {a θ : ChartForm p σ d} (ha : ∀ c y ξ, 0 ≤ a c y ξ) (hθ : ∀ c y ξ, 0 ≤ θ c y ξ) (r : ℝ) {A K δ : ℝ} (hA : 0 ≤ A) (hK : 0 ≤ K) (hδ : 0 < δ) : IsRegConst a θ (fun _ _ => r) A K δ where

theorem **rhoIn_eq** (ha : IsKaehlerForm R a) (hE : IsExph a E) (ψ : CxPts p σ → ℝ) {c c' : AlgChart p σ d} {y : CxPts p σ} (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) (t : ℝ) : rhoIn a E ψ c y t = rhoIn a E ψ c' y t

theorem **rho_eq** (ha : IsKaehlerForm R a) (hE : IsExph a E) (ψ : CxPts p σ → ℝ) {c : AlgChart p σ d} {y : CxPts p σ} (hy : y ∈ c.e.source) (t : ℝ) : rho a E ψ y t = rhoIn a E ψ c y t

theorem **exists_eq_sum** (ha : IsKaehlerForm R a) {c : AlgChart p σ d} {y : CxPts p σ} (hy : y ∈ c.e.source) : ∃ b : Matrix (Fin d) (Fin d) ℂ, a c y = fun ξ => (∑ k, ∑ l, b k l * ξ k * starRingEnd ℂ (ξ l)).re / Real.pi

theorem **continuous_and_smul** (ha : IsKaehlerForm R a) {c : AlgChart p σ d} {y : CxPts p σ} (hy : y ∈ c.e.source) : Continuous (a c y) ∧ ∀ (t : ℝ) ξ, a c y (t • ξ) = t ^ 2 * a c y ξ

theorem **rho_const** (ha : IsKaehlerForm R a) (E : ChartMap p σ d) (r : ℝ) (y : CxPts p σ) (t : ℝ) : rho a E (fun _ => r) y t = r

theorem **rho_zero** (ha : IsKaehlerForm R a) (hE : IsExph a E) (ψ : CxPts p σ → ℝ) (y : CxPts p σ) : rho a E ψ y 0 = ψ y

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ContinuousConjFin

theorem **coord_algebraMap** (τ : E →+* ℂ) (x : E) : coord τ (pInf E (algebraMap E (Ad E) x)) = τ x

theorem **coord_atEmb** (τ : E →+* ℂ) (z : ℂ) : coord τ (atEmb τ z) = z

theorem **atEmb_mul** (τ : E →+* ℂ) (z z' : ℂ) : atEmb τ z * atEmb τ z' = atEmb τ (z * z')

theorem **atEmb_eq** (τ : E →+* ℂ) (z : ℂ) : atEmb τ z = z.re • atEmb τ 1 + z.im • atEmb τ Complex.I

theorem **atEmb_conj** (τ : E →+* ℂ) (z : ℂ) : atEmb (ComplexEmbedding.conjugate τ) z = atEmb τ (starRingEnd ℂ z)

theorem **atEmb_mul_of_ne** {τ τ' : E →+* ℂ} (h : InfinitePlace.mk τ ≠ InfinitePlace.mk τ') (z z' : ℂ) : atEmb τ z * atEmb τ' z' = 0

theorem **lieS_apply** (i j : Fin a) (x : Cinf E) : lieS E a i j x = lie E a (toL (Matrix.single i j x))

theorem **lieS_comm** (i j k l : Fin a) (x y : Cinf E) : lieS E a i j x * lieS E a k l y - lieS E a k l y * lieS E a i j x = (if j = k then lieS E a i l (x * y) else 0) - (if l = i then lieS E a k j (x * y) else 0)

theorem **Eop_comm_aux** (τ τ' : E →+* ℂ) (i j k l : Fin a) : Eop τ i j * Eop τ' k l - Eop τ' k l * Eop τ i j = (if j = k then (1 / 4 : ℂ) • (lieS E a i l (atEmb τ 1 * atEmb τ' 1) - lieS E a i l (atEmb τ .I * atEmb τ' .I) - Complex.I • (lieS E a i l (atEmb τ 1 * atEmb τ' .I) + lieS E a i l (atEmb τ .I * atEmb τ' 1))) else 0) - (if l = i then (1 / 4 : ℂ) • (lieS E a k j (atEmb τ 1 * atEmb τ' 1) - lieS E a k j (atEmb τ .I * atEmb τ' .I) - Complex.I • (lieS E a k j (atEmb τ 1 * atEmb τ' .I) + lieS E a k j (atEmb τ .I * atEmb τ' 1))) else 0)

theorem **glRel_Eop** (τ : E →+* ℂ) : Formalism.REAL.GlRel (Eop (a

theorem **Eop_comm** {τ τ' : E →+* ℂ} (h : τ ≠ τ') (i j k l : Fin a) : Eop τ i j * Eop τ' k l = Eop τ' k l * Eop τ i j

theorem **cap_comm** (τ : E →+* ℂ) (u : ℂ) (i j : Fin a) : cap τ u ∗ Eop τ i j = Eop τ i j ∗ cap τ u

theorem **cap_highest** (τ : E →+* ℂ) (ν : Fin a → ℂ) (φ : Sm E a) (h : Formalism.REAL.IsHW (Eop τ) ν φ) (u : ℂ) : cap τ u φ = (∏ k : Fin a, (u + (ν k + (((a : ℂ) − 1) / 2 − (k : ℕ))))) • φ

theorem **map_qdet** {A B : Type} [Ring A] [Algebra ℂ A] [Ring B] [Algebra ℂ B] (f : A →ₐ[ℂ] B) (X : Fin a → Fin a → A) (u : ℂ) : f (Formalism.REAL.qdet X u) = Formalism.REAL.qdet (fun i j => f (X i j)) u

theorem **qdet_shift** {A : Type} [Ring A] [Algebra ℂ A] (X : Fin a → Fin a → A) (s u : ℂ) : Formalism.REAL.qdet (fun i j => X i j + if i = j then algebraMap ℂ A s else 0) u = Formalism.REAL.qdet X (u + s)

theorem **Zalg_stable** {V : Submodule ℂ (Sm E a)} (hV : ∀ X, ∀ φ ∈ V, lie E a X φ ∈ V) {z : Module.End ℂ (Sm E a)} (hz : z ∈ Zalg E a) : ∀ φ ∈ V, z φ ∈ V

theorem **continuous_chiInf** (χ : HChar E) : Continuous (chiInf χ)

theorem **chiInf_expU** (χ : HChar E) (z : Cinf E) : ((chiInf χ (expU z) : ℂˣ) : ℂ) = Complex.exp (dchi χ z)

theorem **chiInf_expU_atEmb** (χ : HChar E) (τ : E →+* ℂ) (z : ℂ) : ((chiInf χ (expU (atEmb τ z)) : ℂˣ) : ℂ) = Complex.exp (expo χ τ ∗ z + expo χ (ComplexEmbedding.conjugate τ) ∗ starRingEnd ℂ z)

theorem **expo_eq_of_exp** (χ : HChar E) (τ : E →+* ℂ) (p q : ℂ) (h : ∀ z, ((chiInf χ (expU (atEmb τ z)) : ℂˣ) : ℂ) = Complex.exp (p ∗ z + q ∗ starRingEnd ℂ z)) : expo χ τ = p

theorem **chart_mul_chi** (χ : HChar E) (φ : GA E a → ℂ) (g : GA E a) : chart (fun x => φ x ∗ (chiDet a χ x : ℂˣ)) g = fun x => chart φ g x ∗ ((chiDet a χ g : ℂˣ) ∗ extZero (chiInf χ) (Matrix.of x).det)

theorem **contDiffAt_chi_det** (χ : HChar E) {x : Lie E a} (hx : x ∈ invSet E a) : ContDiffAt ℝ ∞ (fun x : Lie E a => extZero (chiInf χ) (Matrix.of x).det) x

theorem **mul_chi_mem_Sm** (χ : HChar E) {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) : (fun x => φ x ∗ (chiDet a χ x : ℂˣ)) ∈ Sm E a

@[simp] theorem **mulChi_apply** (χ : HChar E) (φ : Sm E a) (g : GA E a) : (mulChi a χ φ).1 g = φ.1 g ∗ (chiDet a χ g : ℂˣ)

theorem **mulChi_mul** (χ χ' : HChar E) (φ : Sm E a) : mulChi a (χ ∗ χ') φ = mulChi a χ' (mulChi a χ φ)

theorem **mulChi_one** (φ : Sm E a) : mulChi a 1 φ = φ

theorem **rt_mulChi** (χ : HChar E) (h : GA E a) (φ : Sm E a) : rt E a h (mulChi a χ φ) = mulChi a χ (((chiDet a χ h : ℂˣ) : ℂ) • rt E a h φ)

theorem **fderiv_det_one** (X : Lie E a) : fderiv ℝ (fun x : Lie E a => (Matrix.of x).det) (oneL E a) X = (Matrix.of X).trace

theorem **lie_mulChi** (χ : HChar E) (X : Lie E a) (φ : Sm E a) : lie E a X (mulChi a χ φ) = mulChi a χ (lie E a X φ + dchi χ (Matrix.of X).trace • φ)

theorem **Eop_mulChi** (χ : HChar E) (τ : E →+* ℂ) (i j : Fin a) (φ : Sm E a) : Eop τ i j (mulChi a χ φ) = mulChi a χ (Eop τ i j φ + (if i = j then expo χ τ else 0) • φ)

theorem **cap_mulChi** (χ : HChar E) (τ : E →+* ℂ) (u : ℂ) (φ : Sm E a) : cap τ u (mulChi a χ φ) = mulChi a χ (cap τ (u + expo χ τ) φ)

theorem **Phi_mul_transpose** (a : ℕ) (R : Type) [CommRing R] : GlobalTheory.Tw.Phi a R ∗ (GlobalTheory.Tw.Phi a R)ᵀ = 1

theorem **thetaGL_val** [StarRing R] (g : GL (Fin a) R) : (thetaGL a (starRingEnd R) g).val = GlobalTheory.Tw.theta g.val

theorem **det_thetaGL** (c : R →+* R) (g : GL (Fin a) R) : GeneralLinearGroup.det (thetaGL a c g) = (Units.map c.toMonoidHom (GeneralLinearGroup.det g))⁻¹

theorem **conjPl_conjSrc** (c : E ≃+* E) (v : HeightOneSpectrum (𝒪 E)) : ConjPl c (conjSrc c v) v

theorem **eq_conjSrc** {c : E ≃+* E} {v v' : HeightOneSpectrum (𝒪 E)} (h : ConjPl c v v') : v = conjSrc c v'

theorem **conjSrc_injective** (c : E ≃+* E) : Function.Injective (conjSrc c)

theorem **conjLoc_spec** (c : E ≃+* E) (v : HeightOneSpectrum (𝒪 E)) : Continuous (conjLoc c v) ∧ (∀ k : E, conjLoc c v (algebraMap E _ k) = algebraMap E _ (c k)) ∧ ∀ x, Valued.v (conjLoc c v x) = Valued.v x

theorem **pLoc_conjFin** (c : E ≃+* E) (x : Af E) (v : HeightOneSpectrum (𝒪 E)) : pLoc E v (conjFin c x) = conjLoc c v (pLoc E (conjSrc c v) x)

theorem **conjFin_algebraMap** (c : E ≃+* E) (k : E) : conjFin c (algebraMap E (Af E) k) = algebraMap E (Af E) (c k)

theorem **continuous_conjFin** (c : E ≃+* E) : Continuous (conjFin c)

theorem **pInf_complexConj** {E : Type} [Field E] [NumberField E] [IsCMField E] (x : E) : pInf E (algebraMap E (Ad E) (IsCMField.complexConj E x)) = star (pInf E (algebraMap E (Ad E) x))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ContinuousMapPFin

@[simp] theorem **RU_apply** (x g : UA E n) (φ : UA E n → ℂ) : RU E n x φ g = φ (g ∗ x)

theorem **RU_RU** (x y : UA E n) (φ : UA E n → ℂ) : RU E n x (RU E n y φ) = RU E n (x ∗ y) φ

theorem **fin_mul_inf** {k x : UA E n} (hk : k ∈ finU E n) (hx : x ∈ infU E n) : k ∗ x = x ∗ k

theorem **mem_autU** {φ : UA E n → ℂ} : φ ∈ autU E n ↔ IsAutFormU φ

theorem **smooth_normal** {G V : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (N : Subgroup G) [hN : N.Normal] (x : G) {v : V} (h : IsOpen {k : N | ρ k v = v}) : IsOpen {k : N | ρ k (ρ x v) = ρ x v}

theorem **RU_mem_autU** (x : UA E n) (φ : UA E n → ℂ) (h : φ ∈ autU E n) : RU E n x φ ∈ autU E n

theorem **dual_dual** (lam : DomWeight n) : lam.dual.dual = lam

@[simp] theorem **Wt.at_embedding** (W : Wt E n) (v : InfinitePlace E) : W.at v.embedding = W v

theorem **conjugate_ne** (τ : E →+* ℂ) : ComplexEmbedding.conjugate τ ≠ τ

theorem **Wt.at_conjugate** (W : Wt E n) (τ : E →+* ℂ) : W.at (ComplexEmbedding.conjugate τ) = (W.at τ).dual

theorem **Wt.at_injective** : Function.Injective (Wt.at : Wt E n → _)

theorem **Wt.at_ofFam** {lam : (E →+* ℂ) → DomWeight n} (h : ∀ τ, lam (ComplexEmbedding.conjugate τ) = (lam τ).dual) (τ : E →+* ℂ) : (Wt.ofFam lam).at τ = lam τ

theorem **infW_card** (W : Wt E n) (τ : E →+* ℂ) : Multiset.card (infW W τ) = n

theorem **infW_alg** (W : Wt E n) (τ : E →+* ℂ) : (infW W τ).Nodup ∧ ∀ x ∈ infW W τ, ∃ k : ℤ, x = (k : ℂ) + ((n : ℂ) − 1) / 2

theorem **infW_conjugate** (W : Wt E n) (τ : E →+* ℂ) : infW W (ComplexEmbedding.conjugate τ) = (infW W τ).map fun z => -z

@[simp] theorem **postRep_apply** (W : Wt E n) (k : finU E n) (f : W.Sp →ₗ[ℂ] (UA E n → ℂ)) (v : W.Sp) : postRep W k f v = RU E n k (f v)

theorem **postRep_mem** (W : Wt E n) (k : finU E n) (f : W.Sp →ₗ[ℂ] (UA E n → ℂ)) (h : f ∈ algForms W) : postRep W k f ∈ algForms W

theorem **isSmooth_algRep** (W : Wt E n) : IsSmooth (algRep W)

theorem **algForms.val_left** {W : Wt E n} (f : algForms W) {γ : UA E n} (hγ : γ ∈ ratU E n) (g : UA E n) : algForms.val f (γ * g) = algForms.val f g

theorem **algForms.val_right** {W : Wt E n} (f : algForms W) (g : UA E n) (x : infU E n) : algForms.val f (g * x) = W.rep.dual x⁻¹ (algForms.val f g)

theorem **algForms.val_algRep** {W : Wt E n} (f : algForms W) (k : finU E n) (g : UA E n) : algForms.val (algRep W k f) g = algForms.val f (g * k)

@[simp] theorem **rep_val** (x : UA E n) (φ : π.V) : (π.rep x φ).1 = RU E n x φ.1

theorem **le_autU** : π.V ≤ autU E n

theorem **isIrreducible_rep** : π.rep.IsIrreducible

theorem **incl_ne_zero** : π.incl ≠ 0

theorem **mult_pos** [FiniteDimensional ℂ (π.rep.IntertwiningMap (autRep E n))] : 0 < π.mult

theorem **mem_finPart** {W : Wt E n} {f : algForms W} : f ∈ π.finPart W ↔ ∀ v, f.1 v ∈ π.V

theorem **isSmooth_finRep** (W : Wt E n) : IsSmooth (π.finRep W)

theorem **Iso.exists_map** (h : Iso π π') : ∃ T : π.V →ₗ[ℂ] (UA E n → ℂ), Function.Injective T ∧ (∀ φ, T φ ∈ π'.V) ∧ ∀ x φ, T (π.rep x φ) = RU E n x (T φ)

theorem **Iso.isCompAlong** (h : Iso π π') {Gv : Subgroup (UA E n)} {H W : Type} [Group H] [AddCommGroup W] [Module ℂ W] {p : Gv →* H} {σ : Representation ℂ H W} (hσ : IsCompAlong π.V Gv p σ) : IsCompAlong π'.V Gv p σ

theorem **Iso.isUnramU** (h : Iso π π') {w : HeightOneSpectrum (𝒪 E)} (hw : IsUnramU π.V w) : IsUnramU π'.V w

theorem **Iso.isSat** (h : Iso π π') {w : HeightOneSpectrum (𝒪 E)} {α : Fin (n / 2) → ℂˣ} (hα : π.IsSat w α) : π'.IsSat w α

theorem **Iso.mult_eq** (h : Iso π π') : π.mult = π'.mult

theorem **IsLocComp.congr** {w : HeightOneSpectrum (𝒪 E)} {σ σ' : SmIrr (GL (Fin n) (w.adicCompletion E))} (h : π.IsLocComp w σ) (hπ : Iso π π') (hσ : Nonempty (σ.ρ.Equiv σ'.ρ)) : π'.IsLocComp w σ'

theorem **RU_detFn** (χ : Formalism.L2.Host.HChar E) (x : UA E n) : RU E n x (detFn n χ) = (detChar n χ x : ℂ) • detFn n χ

theorem **detFn_ne_zero** (χ : Formalism.L2.Host.HChar E) : detFn n χ ≠ 0

theorem **continuous_map_pFin** : Continuous (GeneralLinearGroup.map (pFin E) : GA E n → GL (Fin n) (Af E))

theorem **detFn_mem** (χ : Formalism.L2.Host.HChar E) : detFn n χ ∈ autU E n

theorem **AutSub.rep_ofChar** (χ : Formalism.L2.Host.HChar E) (x : UA E n) (φ : (AutSub.ofChar n χ).V) : (AutSub.ofChar n χ).rep x φ = (detChar n χ x : ℂ) • φ

theorem **AutSub.mult_ofChar** (χ : Formalism.L2.Host.HChar E) : (AutSub.ofChar n χ).mult = 1

theorem **mk_eq_mk** {π π' : AutSub E n} : mk π = mk π' ↔ π.Iso π'

theorem **mk_surjective** : Function.Surjective (mk : AutSub E n → AU E n)

@[simp] theorem **mult_mk** (π : AutSub E n) : (mk π).mult = π.mult

@[simp] theorem **sph_mk** (π : AutSub E n) (w : HeightOneSpectrum (𝒪 E)) : (mk π).Sph w ↔ IsUnramU π.V w

@[simp] theorem **isLoc_mk** (π : AutSub E n) (w : HeightOneSpectrum (𝒪 E)) (σ : SmIrr (GL (Fin n) (w.adicCompletion E))) : (mk π).IsLoc w (Quotient.mk _ σ) ↔ π.IsLocComp w σ

theorem **IsLoc.loc** {π : AU E n} {w : HeightOneSpectrum (𝒪 E)} {σ : Irr (GL (Fin n) (w.adicCompletion E))} (h : π.IsLoc w σ) : π.IsLoc w (π.loc w)

@[simp] theorem **isSat_mk** (π : AutSub E n) (w : HeightOneSpectrum (𝒪 E)) (α : Fin (n / 2) → ℂˣ) : (mk π).IsSat w α ↔ π.IsSat w α

theorem **IsSat.satVec** {π : AU E n} {w : HeightOneSpectrum (𝒪 E)} {α : Fin (n / 2) → ℂˣ} (h : π.IsSat w α) : π.IsSat w (π.satVec w)

theorem **sat_card** (π : AU E n) (w : HeightOneSpectrum (𝒪 E)) : Multiset.card (π.sat w) = n / 2

theorem **zero_notMem_sat** (π : AU E n) (w : HeightOneSpectrum (𝒪 E)) : (0 : ℂ) ∉ π.sat w

@[simp] theorem **hasWt_mk** (π : AutSub E n) (τ : E →+* ℂ) (lam : DomWeight n) : (mk π).HasWt τ lam ↔ HasWeightAt π.V τ lam

theorem **HasWt.conjugate** {π : AU E n} {τ : E →+* ℂ} {lam : DomWeight n} (h : π.HasWt τ lam) : π.HasWt (ComplexEmbedding.conjugate τ) lam.dual

theorem **HasWt.wt** {π : AU E n} {τ : E →+* ℂ} {lam : DomWeight n} (h : π.HasWt τ lam) : π.HasWt τ (π.wt.at τ)

theorem **ofChar_eq_iff** {χ χ' : Formalism.L2.Host.HChar E} : ofChar n χ = ofChar n χ' ↔ detChar n χ = detChar n χ'

theorem **mult_ofChar** (χ : Formalism.L2.Host.HChar E) : (ofChar n χ).mult = 1

theorem **ofChar_one** : ofChar n (1 : Formalism.L2.Host.HChar E) = mk (AutSub.triv E n)

theorem **sph_triv** (w : HeightOneSpectrum (𝒪 E)) : (mk (AutSub.triv E n)).Sph w

theorem **isLoc_triv** (w : HeightOneSpectrum (𝒪 E)) : (mk (AutSub.triv E n)).IsLoc w (Quotient.mk _ (SmIrr.triv _))

theorem **infW_wt_triv** (τ : E →+* ℂ) : infW (mk (AutSub.triv E 1)).wt τ = {0}

theorem **hasWt_ofChar_one** (χ : Formalism.L2.Host.HChar E) (τ : E →+* ℂ) (lam : DomWeight 1) (h : ∀ x ∈ embU 1 τ, (detChar 1 χ x : ℂ) = cEmb τ x 0 0 ^ lam.w 0) : (ofChar 1 χ).HasWt τ lam

theorem **finPart_triv_ne_bot** : (AutSub.triv E n).finPart (fun _ => zeroWt n) ≠ ⊥

theorem **locPar_eq** {rc : ∀ w m, SmIrr (GL (Fin m) (w.adicCompletion E)) → LP w} {w : HeightOneSpectrum (𝒪 E)} (hrc : ∀ σ σ' : SmIrr (GL (Fin n) (w.adicCompletion E)), Nonempty (σ.ρ.Equiv σ'.ρ) → rc w n σ = rc w n σ') {π : AU E n} {σ : SmIrr (GL (Fin n) (w.adicCompletion E))} (h : π.loc w = Quotient.mk _ σ) : locPar rc π w = rc w n σ

theorem **isSat_iff** {n : ℕ} (π : AutSub H.E n) (w : PlE H) (α : Fin (n / 2) → ℂˣ) : π.IsSat w α ↔ IsSatAt π.V w α

theorem **unitaryRead_nbc** {rc : RecMap A} {emb inj loc sat sph inf} (hrc : ∀ w (σ σ' : LocIrr w R.nV), Nonempty (σ.ρ.Equiv σ'.ρ) → rc w R.nV σ = rc w R.nV σ') (hloc : ∀ (π : AU H.E R.nV) w σ σ', IsSplitPl H w → π.IsLoc w σ → π.IsLoc w σ' → σ = σ') (hsat : ∀ (π : AU H.E

R.nV) w, ¬ IsSplitPl H w → π.Sph w → ∃ α, π.IsSat w α) (hwt : ∀ (π : AU H.E R.nV) τ, ∃ lam, π.HasWt τ lam) : UnitaryRead (nbc rc emb inj loc sat sph inf) rc

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ContinuousOfDense

theorem **mfact_pos** (α : σ →₀ ℕ) : 0 < mfact α

theorem **mfact_cast** (α : σ →₀ ℕ) : ((mfact α : ℕ) : ℂ) = α.prod fun _ k => (k.factorial : ℂ)

theorem **wt_ne_zero** (α : σ →₀ ℕ) : wt α ≠ 0

theorem **star_wt_mul** (α : σ →₀ ℕ) : (starRingEnd ℂ) (wt α) * wt α = (mfact α : ℕ)

theorem **emb_monomial** (α : σ →₀ ℕ) (c : ℂ) : emb σ (monomial α c) = lp.single 2 α (c * wt α)

theorem **emb_apply** (f : MvPolynomial σ ℂ) (α : σ →₀ ℕ) : emb σ f α = coeff α f * wt α

theorem **emb_injective** : Function.Injective (emb σ)

theorem **dense_range_emb** : Dense (Set.range (emb σ))

theorem **isFockIsometry_embedding** : IsFockIP (emb σ)

@[simp] theorem **diag_apply** (c : (σ →₀ ℕ) → ℂ) (hc : ∀ α, ‖c α‖ = 1) (x : L2 σ) (α : σ →₀ ℕ) : diag c hc x α = c α * x α

theorem **norm_diag** (c : (σ →₀ ℕ) → ℂ) (hc : ∀ α, ‖c α‖ = 1) (x : L2 σ) : ‖diag c hc x‖ = ‖x‖

theorem **diag_emb_monomial** (c : (σ →₀ ℕ) → ℂ) (hc : ∀ α, ‖c α‖ = 1) (α : σ →₀ ℕ) (a : ℂ) : diag c hc (emb σ (monomial α a)) = c α • emb σ (monomial α a)

theorem **continuous_of_dense** {G H : Type*} [TopologicalSpace G] [NormedAddCommGroup H] [NormedSpace ℂ H] (π : G → H →L[ℂ] H) (hπ : ∀ g x, ‖π g x‖ = ‖x‖) {S : Set H} (hS : Dense S) (h : ∀ x ∈ S, Continuous fun g => π g x) (x : H) : Continuous fun g => π g x

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ContractsOfConstant

theorem **BasePointFree.not_isBasePoint** (h : D.BasePointFree V) (x : X) : ¬ D.IsBasePoint V x

theorem **BasePointFree.exists_ne_zero** (h : D.BasePointFree V) : ∃ f ∈ V, f ≠ 0

theorem **VanishesOn.eq_zero** {f : X.functionField} (h : D.VanishesOn f Set.univ) : f = 0

theorem **Eff.vanishesOn_one_iff** (h : D.Eff) {Z : Set X} : D.VanishesOn 1 Z ↔ Z ⊆ D.supp

theorem **IsEqn.generatesAt_aeval_iff** {x : X} {u : X.functionFieldˣ} (h : D.IsEqn x u) {ι : Type} (b : ι → X.functionField) {d : ℕ} {g : MvPolynomial ι k} (hg : g.IsHomogeneous d) : (d • D).GeneratesAt x (aeval b g) ↔ UnitAt x (aeval (fun i => b i * u) g)

theorem **contracts_of_constant** (hk : ∀ (x : X) (c : k), RegAt x (algebraMap k X.functionField c)) (hV : ∀ f ∈ V, f ∈ Set.range (algebraMap k X.functionField)) (C : Set X) : D.Contracts k V C

theorem **mem_of_mem_ratSet** : ∀ f ∈ h.ratSet, f ∈ D.sections

theorem **basePointFree_iff** : D.BasePointFree h.ratSet ↔ S.BasePointFree

theorem **generatesAt_aeval_iff** {d : ℕ} {g : MvPolynomial (Fin S.N) k} (hg : g.IsHomogeneous d) (x : S.Y) : (d • D).GeneratesAt x (aeval (fun j => h.rat (S.b j)) g) ↔ x ∈ nonzeroLocus (S.form hg)

theorem **sameImage_iff** (y y' : S.Y) : D.SameImage k h.ratSet y y' ↔ S.SameImage y y'

theorem **noContract_iff** (hb : S.BasePointFree) (bd : Set S.Y) : D.NoContract k h.ratSet bd ↔ S.NoContract bd

theorem **ratioField_resPt** : S.ratioField (resPt (genericPoint S.Y)) = (ratioField k h.ratSet).toSubfield.map (S.Y.residue (genericPoint S.Y)).hom

theorem **birational_iff** : D.Birational k h.ratSet ↔ (∃ j, S.b j ≠ 0) ∧ S.Birational

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CoordAtConstMulIff

theorem **tie_frameOf** : Comparison₂ P (frameOf Z P)

theorem **IsCoord.const_mul** {k : Type} [Field k] {φ : c.F →+* k} {x : Sp k ⟶ Z.X} {p : Sp k ⟶ Y.Y.X} {g : EigFrame Z φ x} {n : Letter Dw → ℤ} {σ : Y.Y.Fn} {a : k} (h : IsCoord Y p g n σ a) (b : c.F) : IsCoord Y p g n (constF e Y.Y b * σ) (φ b * a)

theorem **IsCoord.carries** {k : Type} [Field k] {φ : c.F →+* k} {x : Sp k ⟶ Z.X} {p : Sp k ⟶ Y.Y.X} {g : EigFrame Z φ x} {n : Letter Dw → ℤ} {σ : Y.Y.Fn} {a : k} (h : IsCoord Y p g n σ a) : Y.Carries p g

theorem **CoordAt.const_mul** {k : Type} [Field k] [Algebra c.F k] {P : FrPt Xo c k} {n : Letter Dw → ℤ} {σ : Y.Y.Fn} {a : k} (h : CoordAt Y P n σ a) (b : c.F) : CoordAt Y P n (constF e Y.Y b * σ) (algebraMap c.F k b * a)

theorem **coordAt_const_mul_iff** {k : Type} [Field k] [Algebra c.F k] {P : FrPt Xo c k} {n : Letter Dw → ℤ} {σ : Y.Y.Fn} {a : k} {b : c.F} (hb : b ≠ 0) : CoordAt Y P n (constF e Y.Y b * σ) a ↔ CoordAt Y P n σ (algebraMap c.F k b⁻¹ * a)

theorem **CoordAt.carried** {k : Type} [Field k] [Algebra c.F k] {P : FrPt Xo c k} {n : Letter Dw → ℤ} {σ : Y.Y.Fn} {a : k} (h : CoordAt Y P n σ a) : Carried Y P

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CoordinatesFlagVector

theorem **co_surjective** : Function.Surjective (co S)

theorem **mem_latV** (x : S.V) : x ∈ latV S Λ ↔ co S x ∈ Λ

theorem **B_eq** (x y : S.V) : (ofHSpace S).B x y = form (starRingEnd E) S.gram (co S x) (co S y)

theorem **isUnimod_latV** (h : SelfDual (starRingEnd E) S.gram Λ) : IsUnimod (ofHSpace S) (latV S Λ)

theorem **coMat_mulVec** {m : ℕ} (x : Fin m → S.V) (c : Fin m → E) : coMat S x *ᵥ c = co S (∑ a, c a • x a)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.CurvNormNonneg

theorem **nsq_pos** {M : ι → Matrix n n ℂ} (hM : ∀ τ, (M τ).PosDef) {v : ι → n → ℂ} (hv : v ≠ 0) : 0 < nsq M v

theorem **nsq_smul** (M : ι → Matrix n n ℂ) (t : ℝ) (v : ι → n → ℂ) : nsq M (t • v) = t ^ 2 * nsq M v

theorem **nsq_zero** (M : ι → Matrix n n ℂ) : nsq M 0 = 0

theorem **continuous_nsq** (M : ι → Matrix n n ℂ) : Continuous (nsq M)

theorem **nsq_conj** (M T : ι → Matrix n n ℂ) (v : ι → n → ℂ) : nsq (fun τ => (T τ)ᴴ * M τ * T τ) v = nsq M (fun τ => T τ *ᵥ v τ)

theorem **exists_bounds** {M : ι → Matrix n n ℂ} (hM : ∀ τ, (M τ).PosDef) : ∃ a b : ℝ, 0 < a ∧ 0 < b ∧ ∀ v, a * ‖v‖ ^ 2 ≤ nsq M v ∧ nsq M v ≤ b * ‖v‖ ^ 2

theorem **ratios_nonempty** [Nonempty ι] [Nonempty n] : (ratios h h0).Nonempty

theorem **ratios_bounds** (hh : ∀ τ, (h τ).PosDef) (hh0 : ∀ τ, (h0 τ).PosDef) : ∃ a b : ℝ, 0 < a ∧ ∀ t ∈ ratios h h0, a ≤ t ∧ t ≤ b

theorem **eigLo_pos** [Nonempty ι] [Nonempty n] (hh : ∀ τ, (h τ).PosDef) (hh0 : ∀ τ, (h0 τ).PosDef) : 0 < eigLo h h0 ∧ eigLo h h0 ≤ eigUp h h0

theorem **le_eigLo_iff** [Nonempty ι] [Nonempty n] (hh : ∀ τ, (h τ).PosDef) (hh0 : ∀ τ, (h0 τ).PosDef) (c : ℝ) : c ≤ eigLo h h0 ↔ ∀ v, c * nsq h0 v ≤ nsq h v

theorem **eigUp_le_iff** [Nonempty ι] [Nonempty n] (hh : ∀ τ, (h τ).PosDef) (hh0 : ∀ τ, (h0 τ).PosDef) (c : ℝ) : eigUp h h0 ≤ c ↔ ∀ v, nsq h v ≤ c * nsq h0 v

theorem **ratios_conj** [DecidableEq n] (h h0 T : ι → Matrix n n ℂ) (hT : ∀ τ, IsUnit (T τ).det) : ratios (fun τ => (T τ)ᴴ * h τ * T τ) (fun τ => (T τ)ᴴ * h0 τ * T τ) = ratios h h0

theorem **connNorm_nonneg** (h h0 : Pt ι p q → ι → Matrix n n ℂ) (β : Pt ι p q → Pt ι p q → ℝ) (z : Pt ι p q) : 0 ≤ connNorm h h0 β z

theorem **curvNorm_nonneg** (h : Pt ι p q → ι → Matrix n n ℂ) (β : Pt ι p q → Pt ι p q → ℝ) (z : Pt ι p q) : 0 ≤ curvNorm h β z

theorem **connNorm_self** (h : Pt ι p q → ι → Matrix n n ℂ) (β : Pt ι p q → Pt ι p q → ℝ) (z : Pt ι p q) : connNorm h h β z = 0

theorem **ddc_const** (c : ℝ) (z ξ : Pt ι p q) : ddc (fun _ => c) z ξ = 0

theorem **dHol_coord** (τ : ι) (a : Fin p) (b : Fin q) (z ξ : Pt ι p q) : dHol (fun w => w τ a b) z ξ = ξ τ a b ∧ dAnti (fun w => w τ a b) z ξ = 0

theorem **exp_le_ri** (i : Fin B.m) {z : Pt ι p q} (hz : z ∈ DomI ι p q) : Real.exp 1 ≤ B.ri i z

theorem **exp_le_r** {z : Pt ι p q} (hz : z ∈ DomI ι p q) : Real.exp 1 ≤ B.r z

theorem **r_eq** {z : Pt ι p q} (hz : z ∈ DomI ι p q) : B.r z = ∑ i, B.ri i z - (B.m - 1) * Real.exp 1

theorem **r_moeb** {γ : ι → GL (Fin p ⊕ Fin q) ℂ} (hγ : γ ∈ Γ) {z : Pt ι p q} (hz : z ∈ DomI ι p q) : B.r (moebI γ z) = B.r z

theorem **met_posDef** {z : Pt ι p q} (hz : z ∈ DomI ι p q) (τ : ι) : (B.met z τ).PosDef

theorem **met_moeb** {γ : ι → GL (Fin p ⊕ Fin q) ℂ} (hγ : γ ∈ Γ) {z : Pt ι p q} (hz : z ∈ DomI ι p q) (τ : ι) : (autE (γ τ) (z τ))ᴴ * B.met (moebI γ z) τ * autE (γ τ) (z τ) = B.met z τ

theorem **ratios_moeb** (hΓ : Γ ≤ UpqI ι p q) (c : ι → ℝ) {γ : ι → GL (Fin p ⊕ Fin q) ℂ} (hγ : γ ∈ Γ) {z : Pt ι p q} (hz : z ∈ DomI ι p q) : ratios (hodgeI c (moebI γ z)) (B.met (moebI γ z)) = ratios (hodgeI c z) (B.met z)

theorem **detNorm_eq** (c : ι → ℝ) (s : ℂ) {z : Pt ι p q} (hz : z ∈ DomI ι p q) (τ : ι) : detNorm c s z τ = ‖s‖ * Real.sqrt (c τ ^ (p + q))

theorem **onQuot_mk** {Γ : Subgroup (ι → GL (Fin p ⊕ Fin q) ℂ)} {f : Pt ι p q → ℝ} (hf : ∀ γ ∈ Γ, ∀ z ∈ DomI ι p q, f (moebI γ z) = f z) (z : DomI ι p q) : onQuot (Γ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.DefiniteSideNonzeroIndexPlacesAnyColumnsScale

theorem **cx_zs_scal** {Bs : DefBase K} {A : DefAmb Bs} (Y : CxData A) (κ lam e) : (reErr (reScal Y κ lam) e).zs = Y.zs

theorem **cx_liftMat_scal** {Bs : DefBase K} {A : DefAmb Bs} (Y : CxData A) {fc : DefSub A → DefFace Bs} (cr : Crd A fc) (κ lam e) (b : Bool) (j : Bs.J) (t : Bs.Tup) : (reErr (reScal Y κ lam) e).liftMat cr b j t = Y.liftMat cr b j t

theorem **cx_Iw_scal** {Bs : DefBase K} {A : DefAmb Bs} (Y : CxData A) {fc : DefSub A → DefFace Bs} (cr : Crd A fc) (κ lam e) (b : Bool) (j : Bs.J) (t : Bs.Tup) (w : HeightOneSpectrum (𝒪 K)) : ((reErr (reScal Y κ lam) e).data cr b).Iw j t w = (Y.data cr b).Iw j t w

theorem **definiteComplexNonzeroIndexPlaces_rescale_errors** {Bs : DefBase K} {A : DefAmb Bs} (Y : CxData A) {fc : DefSub A → DefFace Bs} (cr : Crd A fc) (κ lam e) (N : IndefNum Bs) (w : HeightOneSpectrum (𝒪 K)) : (reErr (reScal Y κ lam) e).W1 cr N w ↔ Y.W1 cr N w

theorem **cx_tC_face** (j : Fin m) (t : T) : Y.tC (crdOf3 A3 P0 ψ0 P1 ψ1 Dc' Sp' Zsc') false j t = Y.tC (crdOf3 A3 P0 ψ0 P1 ψ1 Dc Sp Zsc) false j t

theorem **cx_liftMat_face** (j : Fin m) (t : T) : Y.liftMat (crdOf3 A3 P0 ψ0 P1 ψ1 Dc' Sp' Zsc') false j t = Y.liftMat (crdOf3 A3 P0 ψ0 P1 ψ1 Dc Sp Zsc) false j t

theorem **cx_Iw_face** (j : Fin m) (t : T) (w : HeightOneSpectrum (𝒪 K)) : (Y.data (crdOf3 A3 P0 ψ0 P1 ψ1 Dc' Sp' Zsc') false).Iw j t w = (Y.data (crdOf3 A3 P0 ψ0 P1 ψ1 Dc Sp Zsc) false).Iw j t w

theorem **definiteSide_nonzeroIndexPlaces_any_columns_scale** (N : IndefNum A3.fam3.base3) (w : HeightOneSpectrum (𝒪 K)) : Y.W1 (crdOf3 A3 P0 ψ0 P1 ψ1 Dc' Sp' Zsc') N w ↔ ∃ j : Fin m, ((Y.data (crdOf3 A3 P0 ψ0 P1 ψ1 Dc Sp Zsc) false).Iw j A3.A.μ0 w ≠ 0 ∨ Formalism.DivisionValues.jw ((A3.side3 P0 ψ0 Y.zs).HT j A3.A.μ0) w ((A3.side3 P0 ψ0 Y.zs).Tl j A3.A.μ0) ≠ 0 ∨ N.F0 w j ≠ 0) ∨ ((Y.data (crdOf3 A3 P0 ψ0 P1 ψ1 Dc Sp Zsc) false).Iw j A3.A.μ0 (Formalism.DivisionValues.conjPl w) ≠ 0 ∨ Formalism.DivisionValues.jw ((A3.side3 P0 ψ0 Y.zs).HT j A3.A.μ0) (Formalism.DivisionValues.conjPl w) ((A3.side3 P0 ψ0 Y.zs).Tl j A3.A.μ0) ≠ 0 ∨ N.F0 (Formalism.DivisionValues.conjPl w) j ≠ 0)

theorem **toCx3_upd_reFace** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (kap lam err) (Dc : ∀ Zs : DefSub A3.fam3.amb3, DefCols (A3.side3 F.P1 F.ψ1 Zs)) (Sp : ∀ Zs, SPer (Dc Zs)) (Zsc : DefSub A3.fam3.amb3 → IndefScale A3.fam3.base3) : ((R.updS kap lam).withErr err).toCx3 (reFace F Dc Sp Zsc) = reErr (reScal (R.toCx3 F) (fun _ => kap) lam) err

theorem **complexSide_nonzeroIndexPlaces_iff** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (kap lam err) (Dc : ∀ Zs : DefSub A3.fam3.amb3, DefCols (A3.side3 F.P1 F.ψ1 Zs)) (Sp : ∀ Zs, SPer (Dc Zs)) (Zsc : DefSub A3.fam3.amb3 → IndefScale A3.fam3.base3) (N : IndefNum A3.fam3.base3) (w : HeightOneSpectrum (𝒪 D.what₂.LL)) : (((R.updS kap lam).withErr err).toCx3 (reFace F Dc Sp Zsc)).W1 (reFace F Dc Sp Zsc).cr N w ↔ ∃ j, NzAt R F N.F0 j w ∨ NzAt R F N.F0 j (Formalism.DivisionValues.conjPl w)

theorem **complexSide_nonzeroIndexPlaces_iff_self** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (N : IndefNum A3.fam3.base3) (w : HeightOneSpectrum (𝒪 D.what₂.LL)) : (R.toCx3 F).W1 F.cr N w ↔ ∃ j, NzAt R F N.F0 j w ∨ NzAt R F N.F0 j (Formalism.DivisionValues.conjPl w)

theorem **nzAt_res** {f : T' → T} {μ : T'} (R : RealDat3 D (A3.comap id (f μ)) Ks) (mu : HeightOneSpectrum (𝒪 D.what₂.LL) → T') (tw : Finset (HeightOneSpectrum (𝒪 D.what₂.LL))) (P0 : Subgroup G) (ψ0 : P0 →* D.what₂.LLˣ) (P1 : Subgroup G) (ψ1 : P1 →* D.what₂.LLˣ) (F0 : HeightOneSpectrum (𝒪 D.what₂.LL) → Fin D.R.m → ℝ) (j : Fin D.R.m) (w : HeightOneSpectrum (𝒪 D.what₂.LL)) : NzAt (Res.dat3 f μ R mu tw) (Res.bare3 _ P0 ψ0 P1 ψ1) F0 j w ↔ NzAt R (Res.bare3 _ P0 ψ0 P1 ψ1) F0 j w

theorem **complexSide_numberContents_eq_scaleAtSubspaces** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (kap lam err) (Dc : ∀ Zs : DefSub A3.fam3.amb3, DefCols (A3.side3 F.P1 F.ψ1 Zs)) (Sp : ∀ Zs, SPer (Dc Zs)) (Zsc : DefSub A3.fam3.amb3 → IndefScale A3.fam3.base3) (C CH : ℝ) : (((R.updS kap lam).withErr err).num (reFace F Dc Sp Zsc) C CH).F0 = (Zsc (zsOf R F)).F0

theorem **_root_.GlobalTheory.Mem.R3.columnChoiceReferenceFace_eq_bareFaceTwoLevels** (U : Set I.ι) (Xg : Finset (Gc X.B px → X.B.LL)) (tr : TrI X.B px) : face0 X I hX U Xg tr = Res.bare3 (famR X I hX U Xg tr) (lev0 I) (chr0 I) (lev1 I) (chr1 I)

theorem **pins_resR** (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') (e : z.R.toRealDat.ExAt) (hB : z.R.PThmB) : Pins (z.resR h) (face0 X I hX (usedOf I T Wt') z.Xg z.tr) z.cG

theorem **holds_resR** (h : Wt' ⊆ Wt) (hH : Holds z.R z.F) : Holds (z.resR h) (face0 X I hX (usedOf I T Wt') z.Xg z.tr)

theorem **levOf_resR** (h : Wt' ⊆ Wt) : LevOf X I (usedOf I T Wt') (z.resR h).toLevCh where

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.DeterminantCharacterUnitaryApply

theorem **loc_self** (v : HeightOneSpectrum (𝒪 L)) (x : (v.adicCompletion L)ˣ) : ((localIdele L v x : Ideles L) : AdeleRing (𝒪 L) L).2 v = x

theorem **loc_ne** {v w : HeightOneSpectrum (𝒪 L)} (h : w ≠ v) (x : (v.adicCompletion L)ˣ) : ((localIdele L v x : Ideles L) : AdeleRing (𝒪 L) L).2 w = 1

theorem **valAt_loc_self** (v : HeightOneSpectrum (𝒪 L)) (k : Lˣ) : valAt v (localIdele L v (atFin v k)) = v.valuation L (k : L)

theorem **valAt_loc_ne** {v w : HeightOneSpectrum (𝒪 L)} (h : w ≠ v) (x : (v.adicCompletion L)ˣ) : valAt w (localIdele L v x) = 1

theorem **valuation_unif** (v : HeightOneSpectrum (𝒪 L)) : v.valuation L (unif v : L) = WithZero.exp (-1 : ℤ)

theorem **unif_not_mem_unitIdl** (v : HeightOneSpectrum (𝒪 L)) : localIdele L v (atFin v (unif v)) ∉ unitIdl L

theorem **eq_refl_of_isConjI_id** {c : L ≃+* L} (h : IsConjI c (MonoidHom.id (Ideles L))) : c = RingEquiv.refl L

theorem **diag_eq** (k : Lˣ) : diagUnits L k = infDiag k * finPartI L (diagUnits L k)

theorem **fin_diag** (χ : HChar L) (k : Lˣ) : χ.1 (finPartI L (diagUnits L k)) = (χ.inf k)⁻¹

theorem **detSub_rank_one** (u : L) : detSub (Submodule.span (𝒪 L) {(fun _ => u : Fin 1 → L)}) = Submodule.span (𝒪 L) {u}

theorem **detSub_rank_zero** (M : Submodule (𝒪 L) (Fin 0 → L)) : detSub M = Submodule.span (𝒪 L) {1}

theorem **not_splitsC_one** : ¬ SplitsC L 1

theorem **emb_conj** (σ : H.E →+* ℂ) (z : H.E) : σ (H.conj z) = star (σ z)

theorem **conj_F** (k : H.F) : H.conj (k : H.E) = k

theorem **pack_chiD_injective** : Function.Injective (pack H).chiD

theorem **infIdele_mem_KD** (w : InfinitePlace H.E) (y : (w.Completion)ˣ) : infIdele H.E w y ∈ KD H

theorem **determinantCharacter_unitary_apply** {χ : HChar H.E} (hχ : IsChiPsi H χ) (h : KD H ≤ (χ.1.comp (finPartI H.E)).ker) (f : (letI

theorem **valAt_locF** (w : HeightOneSpectrum (𝒪 H.E)) (v : HeightOneSpectrum (𝒪 H.F)) (k : (H.F)ˣ) : open Classical in valAt w (locF H.F v k) = if w.under (𝒪 H.F) = v then w.valuation H.E ((k : H.F) : H.E) else 1

theorem **under_eq** {w w' : HeightOneSpectrum (𝒪 H.E)} (h : ConjPl H.conj.toRingEquiv w w') : w.under (𝒪 H.F) = w'.under (𝒪 H.F)

theorem **val_eq_of_mem_KD** {w w' : HeightOneSpectrum (𝒪 H.E)} (h : ConjPl H.conj.toRingEquiv w w') {a : Ideles H.E} (ha : a ∈ KD H) : valAt w a = valAt w' a

theorem **val_eq_of_mem_UOT** {w w' : HeightOneSpectrum (𝒪 H.E)} (h : ConjPl H.conj.toRingEquiv w w') {a : Ideles H.E} (ha : (QuotientGroup.mk a : DtT H) ∈ UOT H) : valAt w a = valAt w' a

theorem **infinite_of_split** {w w' : HeightOneSpectrum (𝒪 H.E)} (h : ConjPl H.conj.toRingEquiv w w') (hne : w' ≠ w) : Infinite (DtT H ⧸ UOT H)

theorem **valAt_finprod** {L : Type} [Field L] [NumberField L] (n : HeightOneSpectrum (𝒪 L) → ℤ) (hn : {w | n w ≠ 0}.Finite) (w : HeightOneSpectrum (𝒪 L)) : valAt w (∏ᶠ v : HeightOneSpectrum (𝒪 L), localIdele L v (atFin v (unif v)) ^ (n v).toNat) = WithZero.exp (-((n w).toNat : ℤ))

theorem **valuation_eq_exp** {L : Type} [Field L] [NumberField L] (w : HeightOneSpectrum (𝒪 L)) (k : Lˣ) : w.valuation L (k : L) = WithZero.exp (-ordF w k)

theorem **relDet_sign** (u k : H.Eˣ) (hu : (u : H.E) * H.conj (k : H.E) = k) (hu1 : (u : H.E) * H.conj (u : H.E) = 1) (hfin : {w | ordF w u ≠ 0}.Finite) {w w' : HeightOneSpectrum (𝒪 H.E)} (h : ConjPl H.conj.toRingEquiv w w') : ∃ x : Ideles H.E, relDetT H 1 (Submodule.span _ {fun _ => ((1 : H.Eˣ) : H.E)}) (Submodule.span _ {fun _ => (u : H.E)}) = QuotientGroup.mk (QuotientGroup.mk x) ∧ valAt w' x * w.valuation H.E (k : H.E) = valAt w x * w'.valuation H.E (k : H.E)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.DifferentiableAtChart

theorem **map_cornerGL_of_zero** (s : R →ₙ+* S) (p : S →+* T) (h : ∀ x, p (s x) = 0) (g : GL (Fin a) R) : GeneralLinearGroup.map p (cornerGL a s g) = 1

theorem **map_cornerGL_of_hom** (s : R →ₙ+* S) (p : S →+* T) (q : R →+* T) (h : ∀ x, p (s x) = q x) (g : GL (Fin a) R) : GeneralLinearGroup.map p (cornerGL a s g) = GeneralLinearGroup.map q g

theorem **det_map** (p : S →+* T) (g : GL (Fin a) S) : ((GeneralLinearGroup.det (GeneralLinearGroup.map p g) : Tˣ) : T) = p (GeneralLinearGroup.det g : Sˣ)

theorem **pLoc_locGL** {a : ℕ} (w : HeightOneSpectrum (𝒪 E)) (g : GL (Fin a) (w.adicCompletion E)) : GeneralLinearGroup.map (pLoc E w) (locGL E a w g) = g

theorem **pLoc_locGL_of_ne** {a : ℕ} {w v : HeightOneSpectrum (𝒪 E)} (h : v ≠ w) (g : GL (Fin a) (w.adicCompletion E)) : GeneralLinearGroup.map (pLoc E v) (locGL E a w g) = 1

theorem **finId_det_locGL** {a : ℕ} (w : HeightOneSpectrum (𝒪 E)) (g : GL (Fin a) (w.adicCompletion E)) : finId E (GeneralLinearGroup.det (locGL E a w g)) = localIdele E w (GeneralLinearGroup.det g)

theorem **continuous_finId** : Continuous (finId E)

theorem **continuous_infEq** : Continuous (infEq E)

theorem **continuous_infId** : Continuous (infId E)

theorem **pInf_algebraMap** (x : E) (w : InfinitePlace E) : pInf E (algebraMap E (Ad E) x) w = w.embedding x

theorem **ad_ext** {x y : Ad E} (h1 : pInf E x = pInf E y) (h2 : pFin E x = pFin E y) : x = y

theorem **ga_ext** {g h : GA E a} (h1 : GeneralLinearGroup.map (pInf E) g = GeneralLinearGroup.map (pInf E) h) (h2 : GeneralLinearGroup.map (pFin E) g = GeneralLinearGroup.map (pFin E) h) : g = h

@[simp] theorem **pInf_infGL** (y : GL (Fin a) (Cinf E)) : GeneralLinearGroup.map (pInf E) (infGL E a y) = y

@[simp] theorem **pFin_infGL** (y : GL (Fin a) (Cinf E)) : GeneralLinearGroup.map (pFin E) (infGL E a y) = 1

@[simp] theorem **pInf_finGL** (k : GL (Fin a) (Af E)) : GeneralLinearGroup.map (pInf E) (finGL E a k) = 1

@[simp] theorem **pFin_finGL** (k : GL (Fin a) (Af E)) : GeneralLinearGroup.map (pFin E) (finGL E a k) = k

theorem **infGL_mul_finGL** (g : GA E a) : infGL E a (GeneralLinearGroup.map (pInf E) g) * finGL E a (GeneralLinearGroup.map (pFin E) g) = g

theorem **infGL_comm** (y : GL (Fin a) (Cinf E)) (k : GL (Fin a) (Af E)) : infGL E a y * finGL E a k = finGL E a k * infGL E a y

theorem **det_infGL** (y : GL (Fin a) (Cinf E)) : GeneralLinearGroup.det (infGL E a y) = infId E (GeneralLinearGroup.det y)

theorem **det_finGL** (k : GL (Fin a) (Af E)) : GeneralLinearGroup.det (finGL E a k) = finId E (GeneralLinearGroup.det k)

theorem **contDiff_det** : ContDiff ℝ ∞ fun x : Lie E a => (Matrix.of x).det

theorem **isOpen_invSet** : IsOpen (invSet E a)

theorem **val_mem_invSet** (y : GL (Fin a) (Cinf E)) : (toL y.val) ∈ invSet E a

theorem **chart_val** (φ : GA E a → ℂ) (g : GA E a) (y : GL (Fin a) (Cinf E)) : chart φ g (toL y.val) = φ (g * infGL E a y)

theorem **chart_mul** (φ : GA E a → ℂ) (g : GA E a) (y : GL (Fin a) (Cinf E)) : chart φ (g * infGL E a y) = chart φ g ∘ lmul (toL y.val)

theorem **chart_rt_inf** (φ : GA E a → ℂ) (g : GA E a) (y : GL (Fin a) (Cinf E)) : chart (fun x => φ (x * infGL E a y)) g = chart φ g ∘ rmul (toL y.val)

theorem **chart_rt_fin** (φ : GA E a → ℂ) (g : GA E a) (k : GL (Fin a) (Af E)) : chart (fun x => φ (x * finGL E a k)) g = chart φ (g * finGL E a k)

theorem **rt_mem_Sm** (h : GA E a) (φ : GA E a → ℂ) (hφ : φ ∈ Sm E a) : GlobalTheory.PRep.rightReg ℂ (GA E a) ℂ h φ ∈ Sm E a

@[simp] theorem **rt_apply** (h g : GA E a) (φ : Sm E a) : (rt E a h φ).1 g = φ.1 (g * h)

theorem **contDiffAt_chart** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) {x : Lie E a} (hx : x ∈ invSet E a) : ContDiffAt ℝ ∞ (chart φ g) x

theorem **differentiableAt_chart** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) : DifferentiableAt ℝ (chart φ g) (oneL E a)

theorem **chart_lieD** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (X : Lie E a) (g : GA E a) {x : Lie E a} (hx : x ∈ invSet E a) : chart (lieD X φ) g x = fderiv ℝ (chart φ g) x (rmul X x)

theorem **lieD_mem_Sm** (X : Lie E a) {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) : lieD X φ ∈ Sm E a

@[simp] theorem **lie_apply** (X : Lie E a) (φ : Sm E a) (g : GA E a) : (lie E a X φ).1 g = lieD X φ.1 g

theorem **hasDerivAt_curve** (φ : Sm E a) (g : GA E a) (c : ℝ → GL (Fin a) (Cinf E)) (X : Lie E a) (h0 : c 0 = 1) (hc : HasDerivAt (fun t => toL (c t).val) X 0) : HasDerivAt (fun t => φ.1 (g * infGL E a (c t))) (lieD X φ.1 g) 0

theorem **lie_rt_fin** (X : Lie E a) (k : GL (Fin a) (Af E)) : lie E a X * rt E a (finGL E a k) = rt E a (finGL E a k) * lie E a X

theorem **lie_comm** (X Y : Lie E a) : lie E a X * lie E a Y − lie E a Y * lie E a X = lie E a (toL (Matrix.of X * Matrix.of Y)) − lie E a (toL (Matrix.of Y * Matrix.of X))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.DivisionPairOfAdelicFormsFaceColumnsData

theorem **TwIn.wd_wU** (x : TwIn ex) (u : D.what₂.Word) : x.wd (x.C.wU u) = u

theorem **TwIn.wU_wd** (x : TwIn ex) (m : (x.F.Dc (zsOf x.R x.F)).Mem) : x.C.wU (x.wd m) = m

theorem **phEff_one** (x : TwIn ex) : phEff x.R = fun _ => 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.DuStarPos

theorem **LevCh.piT_ne** {P : PreCh} (L : LevCh P) (u : P.Word) : (L.piT u : P.H.E) ≠ 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.DualGramInterpGram

theorem **lminor_pos** {M : Matrix (Fin m) (Fin m) ℂ} (h : M.PosDef) (k : Fin (m + 1)) : 0 < lminor M k

theorem **grSq_pos** {M : Matrix (Fin m) (Fin m) ℂ} (h : M.PosDef) (i : Fin m) : 0 < grSq M i

theorem **lminor_eq_prod** {T : Matrix (Fin m) (Fin m) ℂ} (hT : ∀ a c, c < a → T a c = 0) (k : Fin (m + 1)) : lminor T k = ∏ a : Fin k, T (a.castLE (Nat.le_of_lt_succ k.2)) (a.castLE (Nat.le_of_lt_succ k.2))

theorem **lminor_tri** {T : Matrix (Fin m) (Fin m) ℂ} (hT : ∀ a c, c < a → T a c = 0) (i : Fin m) : lminor T i.succ = lminor T i.castSucc * T i i

theorem **grSq_tri** (M : Matrix (Fin m) (Fin m) ℂ) {T : Matrix (Fin m) (Fin m) ℂ} (hT : ∀ a c, c < a → T a c = 0) (hd : ∀ a, T a a ≠ 0) (i : Fin m) : grSq (Tᴴ * M * T) i = ‖T i i‖ ^ 2 * grSq M i

theorem **grSq_map_conj** (M : Matrix (Fin m) (Fin m) ℂ) (i : Fin m) : grSq (M.map (starRingEnd ℂ)) i = grSq M i

theorem **grSq_diagonal** {d : Fin m → ℝ} (hd : ∀ a, d a ≠ 0) (i : Fin m) : grSq (diagonal fun a => (d a : ℂ)) i = d i

theorem **isLeast_grSq** {M : Matrix (Fin m) (Fin m) ℂ} (h : M.PosDef) (i : Fin m) : IsLeast {a | ∃ v : Fin m → ℂ, v i = 1 ∧ (∀ k, i < k → v k = 0) ∧ a = (star v ⬝ᵥ M *ᵥ v).re} (grSq M i)

theorem **mix_posDef** (hs : s ∈ Set.Icc (0 : ℝ) 1) (h₀ : M₀.PosDef) (h : M.PosDef) : (mix s M₀ M).PosDef

theorem **mixDual_posDef** (hs : s ∈ Set.Icc (0 : ℝ) 1) (h₀ : M₀.PosDef) (h : M.PosDef) : (mixDual s M₀ M).PosDef

theorem **dualGram_dualGram** (h : IsUnit M.det) : dualGram (dualGram M) = M

theorem **dualGram_mixDual** (hs : s ∈ Set.Icc (0 : ℝ) 1) (h₀ : M₀.PosDef) (h : M.PosDef) : dualGram (mixDual s M₀ M) = mix s (dualGram M₀) (dualGram M)

theorem **mix_conj** (T : Matrix (Fin m) (Fin m) ℂ) : mix s (Tᴴ * M₀ * T) (Tᴴ * M * T) = Tᴴ * mix s M₀ M * T

theorem **mixDual_conj** {T : Matrix (Fin m) (Fin m) ℂ} (hT : IsUnit T.det) : mixDual s (Tᴴ * M₀ * T) (Tᴴ * M * T) = Tᴴ * mixDual s M₀ M * T

theorem **mix_values** : mix 0 M₀ M = M₀ ∧ mix 1 M₀ M = M ∧ mix s M M = M

theorem **mixDual_values** (h₀ : IsUnit M₀.det) (h : IsUnit M.det) : mixDual 0 M₀ M = M₀ ∧ mixDual 1 M₀ M = M ∧ mixDual s M M = M

theorem **exists_isMap** (f : EigFrame Z φ x) (ψ : R →+* R') (hφ : φ' = ψ.comp φ) (hx : x' = spMap ψ ≫ x) : ∃ f' : EigFrame Z φ' x', f.IsMap ψ f'

theorem **co_spec** (i : Fin (Dw.mult τ)) : IsCombAtG Z.t x (f.v τ) (fun a => f.co g τ a i) (g.v τ i)

theorem **vec_eq** (i : Fin (Dw.mult τ)) : g.vec τ i = ∑ a, f.co g τ a i • f.vec τ a

theorem **co_eq** {T : Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) R} (h : ∀ i, IsCombAtG Z.t x (f.v τ) (fun a => T a i) (g.v τ i)) : f.co g τ = T

theorem **co_mul** : f.co g τ * g.co k τ = f.co k τ

theorem **co_self** : f.co f τ = 1

theorem **isUnit_det_co** : IsUnit (f.co g τ).det

theorem **IsMap.co** {ψ : R →+* R'} {f' g' : EigFrame Z φ' x'} (hf : f.IsMap ψ f') (hg : g.IsMap ψ g') (hx : x' = spMap ψ ≫ x) : f'.co g' τ = (f.co g τ).map ψ

theorem **Tri.co** {ψ : R →+* R'} {f' g' : EigFrame Z φ' x'} {t : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → R'ˣ} (h : f.Tri ψ g' t) (hf : f.IsMap ψ f') : (∀ a i, i < a → f'.co g' τ a i = 0) ∧ ∀ i, f'.co g' τ i i = t τ i

theorem **IsMap.tri** {ψ : R →+* R'} {f' : EigFrame Z φ' x'} (hf : f.IsMap ψ f') (hφ : φ' = ψ.comp φ) (hx : x' = spMap ψ ≫ x) : f.Tri ψ f' 1

theorem **interp_gram** (h : τ (H.rt : E) = H.rt0) : (interp s hs G₀ G).gram y f τ = mix (s y) (G₀.gram y f τ) (G.gram y f τ)

theorem **dualGram_interp_gram** (h : τ (H.rt : E) ≠ H.rt0) : dualGram ((interp s hs G₀ G).gram y f τ) = mix (s y) (dualGram (G₀.gram y f τ)) (dualGram (G.gram y f τ))

theorem **interp_gram_zero** (h : s y = 0) : (interp s hs G₀ G).gram y f τ = G₀.gram y f τ

theorem **π_genStr** : Y.π ≫ genStr Z = Y.Y.str ≫ spMap e.symm.toRingHom

theorem **spMap_evStalk** (p : Y.Y.pts σ) : spMap (evStalk p) ≫ stalkPt Y.Y (base p) = p.1

theorem **exists_adapted** (p : Y.Y.pts σ) : ∃ q : (∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Y.Y.Fnˣ) × Y.FrameAt p, (∀ τ i, p ∈ eqnSet σ (Y.gr τ i) (q.1 τ i)) ∧ Y.Adapted p q.1 q.2 1

theorem **genSq_pos** {p : Y.Y.pts σ} (g : Y.FrameAt p) : 0 < Y.genSq G n g

theorem **prodPow_mem** (h : ∀ τ i, p ∈ eqnSet σ (Y.gr τ i) (t τ i)) : p ∈ eqnSet σ (divOf Y n) (prodPow t n)

theorem **Adapted.compare** (h₁ : Y.Adapted p t₁ g₁ 1) (h : Y.Adapted p t g t') : ∃ u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Y.Y.X.presheaf.stalk (base p))ˣ, (∀ τ i, t₁ τ i = t τ i * Units.map (algebraMap _ Y.Y.X.functionField).toMonoidHom (u τ i)) ∧ ∀ τ i, grSq (G.gram _ g τ) i = ‖evStalk p (u τ i) * t' τ i‖ ^ 2 * grSq (G.gram _ g₁ τ) i

theorem **Adapted.mem_eqnSet** (h : Y.Adapted p t g t') (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) : p ∈ eqnSet σ (Y.gr τ i) (t τ i)

theorem **met_h** (h : Y.Adapted p t g t') : (Y.met G n).h (prodPow t n) p = Y.genSq G n g / ‖((prodPow t' n : ℂˣ) : ℂ)‖ ^ 2

theorem **adapted_of_tri** {y : Y.Y.X} {ψ : Y.Y.X.presheaf.stalk y →+* ℂ} {φ : c.F →+* Y.Y.X.presheaf.stalk y} {f : EigFrame Z φ (stalkPt Y.Y y ≫ Y.π ≫ genι Z)} (hp : p.1 = spMap ψ ≫ stalkPt Y.Y y) (a : f.Tri (algebraMap (Y.Y.X.presheaf.stalk y) Y.Y.X.functionField) Y.f0 t) (b : f.Tri ψ g t') : Y.Adapted p t g t' ∧ ∀ r, Var.val p (Y.Y.cx σ (algebraMap _ Y.Y.X.functionField r)) = ψ r

theorem **pn_met** {s : Y.Y.Fn} {a : ℂ} (h : IsCoord Y p.1 g n s a) : pn Y.Y σ _ (Y.met G n) (Y.Y.cx σ s) p = ‖a‖ ^ 2 * Y.genSq G n g

theorem **posSet_met** : (Y.met G n).posSet = Set.univ

theorem **met_congr** {G' : EigMet Z (embOf e σ)} (h : ∀ g τ, G.gram (Y.down p) g τ = G'.gram (Y.down p) g τ) (a : Y.Y.Fnˣ) : (Y.met G n).h a p = (Y.met G' n).h a p

theorem **met_add_h** {a a' : Y.Y.Fnˣ} (h : p ∈ eqnSet σ (divOf Y n) a) (h' : p ∈ eqnSet σ (divOf Y n') a') : (Y.met G (n + n')).h (a * a') p = (Y.met G n).h a p * (Y.met G n').h a' p

theorem **divOf_add** : divOf Y (n + n') = divOf Y n + divOf Y n'

theorem **pn_met_add** : pn Y.Y σ _ (Y.met G (n + n')) = pn Y.Y σ _ (mAdd Y.Y σ _ _ (Y.met G n) (Y.met G n'))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EisGEqZeroOfInfinite

theorem **eisG_map** (Λ : AddSubgroup ℂ) (δ : ℂ) (hδ : δ ≠ 0) (k : ℕ) : eisG (Λ.map (AddMonoidHom.mulLeft δ)) k = (δ ^ (2 * k))⁻¹ * eisG Λ k

theorem **discr_map** (Λ : AddSubgroup ℂ) (δ : ℂ) (hδ : δ ≠ 0) : TaylorLogMetric.discr (Λ.map (AddMonoidHom.mulLeft δ)) = (δ ^ 12)⁻¹ * TaylorLogMetric.discr Λ

theorem **jLat_map** (Λ : AddSubgroup ℂ) (δ : ℂ) (hδ : δ ≠ 0) : jLat (Λ.map (AddMonoidHom.mulLeft δ)) = jLat Λ

theorem **exists_isConjLat_self** (emb : LL →+* ℂ) (Λ : AddSubgroup ℂ) (h : TaylorLogMetric.discr Λ ≠ 0) (ha : ∃ a : LL, emb a = jLat Λ) : ∃ Λ', IsConjLat emb emb Λ Λ'

theorem **eisG_eq_zero_of_infinite** (Λ : AddSubgroup ℂ) (h : {w : ℂ | w ∈ Λ ∧ w ≠ 0 ∧ ‖w‖ ≤ 1}.Infinite) (k : ℕ) : eisG Λ k = 0

theorem **finite_of_isConjLat** {emb s : LL →+* ℂ} {Λ Λ' : AddSubgroup ℂ} (h : IsConjLat emb s Λ Λ') : {w : ℂ | w ∈ Λ' ∧ w ≠ 0 ∧ ‖w‖ ≤ 1}.Finite

theorem **ne_bot_of_isConjLat** {emb s : LL →+* ℂ} {Λ Λ' : AddSubgroup ℂ} (h : IsConjLat emb s Λ Λ') : Λ' ≠ ⊥

theorem **ellLat_ne_bot** (embE : H.E →+* LL) (g : R.GG) : ellLat H embE g ≠ ⊥

theorem **ne_zero_of_isGen** {embE : H.E →+* LL} {g : R.GG} {x : LL} (h : IsGen H embE g x) : x ≠ 0

theorem **gen_spec** {emb : LL →+* ℂ} {embE : H.E →+* LL} (h : FieldOK H emb embE) (g) : IsGen H embE g (gen H embE g)

theorem **prod_norm_eq_of_isGen** {embE : H.E →+* LL} {g : R.GG} {x y : LL} (hx : IsGen H embE g x) (hy : IsGen H embE g y) : ∏ s : LL →+* ℂ, ‖s x‖ = ∏ s : LL →+* ℂ, ‖s y‖

theorem **area_pos_of_isConjLat** {emb s : LL →+* ℂ} {Λ Λ' : AddSubgroup ℂ} (h : IsConjLat emb s Λ Λ') : (∃ L : PeriodPair, ∀ z, z ∈ Λ' ↔ z ∈ L.lattice) ∧ 0 < area Λ'

theorem **exists_isConjLat** (emb s : LL →+* ℂ) (Λ : AddSubgroup ℂ) (ha : ∃ a : LL, emb a = jLat Λ) : ∃ Λ', IsConjLat emb s Λ Λ'

theorem **conjLat_spec** {emb : LL →+* ℂ} {embE : H.E →+* LL} (h : FieldOK H emb embE) (s : LL →+* ℂ) : IsConjLat emb s (baseLat H) (conjLat H emb s)

theorem **om_eq** {emb : LL →+* ℂ} {embE : H.E →+* LL} (h : FieldOK H emb embE) {I : Type} (e : I ≃ R.Psi) (τ : I) (s : LL →+* ℂ) : (trueGx H emb embE e).Om τ s = area (conjLat H emb s) / (Real.pi * ‖s (gen H embE (e τ).1)‖ ^ 2)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EmbeddingCompatHeq

theorem **tauEmb_injective** (D : D7) : Function.Injective (DV.tauEmb D)

theorem **nonempty_cmList** (A : Amb D) : Nonempty (CmList A)

theorem **cmList_unique** (L L' : CmList A) : L = L'

theorem **card_tau** (D : D7) : Fintype.card D.what₂.Tau = Module.finrank ℚ D.host.E / 2

theorem **dim_flag** (Φ : s.Flag) (k : s.Comp) : (Φ.C k).Y.dim = Formalism.DegreeFormula.dimFl D.what₂.Tau N

theorem **exists_kit** (jC : ∀ (p : ℕ) [Fact p.Prime], PlL D p → (DV.KW p →+* ℂ)) (ng : NgPkg) (Φ : s.Flag) (ρ : KeySec s.Xo κ.c) (X : PadPkg D N j s.Comp jC ng) : ∃ k : Kit s L C jC ng, k.Φ = Φ ∧ k.key = ρ ∧ k.pad.loc = X.loc

theorem **wtA_big** (A : Amb D) {i : ℕ} (w : WeightFamilies.Wt D i) (t : D.what₂.Tau) (a : ℕ) : wtA w (DV.tauEmb D t, a) = ext0 (w.aP t) a

theorem **wtA_small** (A : Amb D) {i : ℕ} (w : WeightFamilies.Wt D i) (t : D.what₂.Tau) (a : ℕ) : wtA w (conjEmb (DV.tauEmb D t), a) = ext0 (w.aM t) a

theorem **wt_wtOfN** {i : ℕ} (w : WeightFamilies.Wt D i) (hp : S.Dw.p = (D.R.sigW i).1) (hq : S.Dw.q = (D.R.sigW i).2) : S.wt L (wtOfN N w) = GlobalTheory.AllWt.weightOf S.Dw (fun x => (wtA w x : ℤ)) ((w.n / 2 : ℕ) : ℤ)

theorem **Lw_wtOfN** {i : ℕ} (w : WeightFamilies.Wt D i) (hp : S.Dw.p = (D.R.sigW i).1) (hq : S.Dw.q = (D.R.sigW i).2) (Φ : s.Flag) (k : s.Comp) : s.Lw L Φ k (wtOfN N w) = (Φ.C k).LEx (exKey S.Dw (wtA w) w.n)

theorem **trRul_rfl** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {m : MD H} {c c' : Coeff m.1} (hle : c.F ≤ c'.F) (f : Rul m.2.2 c) : trRul (rfl : m = m) hle f = GlobalTheory.FieldBC.rulBC hle f

theorem **level_eq** : T.Q.l = F.κg.l

theorem **types_eq** {P : FormsOver K R k} (hP : FormsAre h t K R k P) : P.KO = T.Q.c.F ∧ ∀ j w, P.Cusp j w = (recB h.toLawsCore).Cusp j w

theorem **range_cmp** {P : FormsOver K R k} (hP : FormsAre h t K R k P) {j : ℕ} (hj : InR D j) (w : WeightFamilies.Wt D j) : Set.range (fun (f : P.Cusp j w) i => P.cmp j w i f) = Set.range (fun (f : (recB h.toLawsCore).Cusp j w) i => cmpAt h t k hj w.n i f.1)

theorem **exists_place** {P : FormsOver K R k} (hP : FormsAre h t K R k P) (la : HeightOneSpectrum (𝒪 P.KO)) : ∃ la' : HeightOneSpectrum (𝒪 T.Q.c.F), HEq la la'

theorem **pt_nonempty** {P : FormsOver K R k} (hP : FormsAre h t K R k P) {j : ℕ} (hj : InR D j) (la : HeightOneSpectrum (𝒪 P.KO)) : Nonempty (P.Pt j la)

theorem **parTie_par** (F : SrcFam A) : ParTie F.par F.pins.1 F

theorem **primes_first** (F : SrcFam A) : F.κg.primes = Finset.univ.image D.host.g

theorem **not_parTie** {Q : Par D} {h : Q.PinHost} {F : SrcFam (ambOf Q h)} (hA : Q.l.A ≠ 1) : ¬ ParTie Q h F

theorem **embedding_compat_heq** {B B' : WeightFfamilies.Prov D} (hB : B = B') (kL : B.KO →+* D.what₂.LL) (h1 : ∀ x : 𝒪 B.KO, kL (x : B.KO) = D.what₂.OtoLL (B.eO x)) (ι : B'.KO →+* ℂ) (h2 : ∀ x : 𝒪 B'.KO, D.what₂.embLL (D.what₂.OtoLL (B'.eO x)) = ι (x : B'.KO)) (x : B.KO) (x' : B'.KO) (hx : HEq x x') : D.what₂.embLL (kL x) = ι x'

theorem **kL_of_base** {P : FormsOver K R k} (hP : P.toWeightFamilies = recB h.toLawsCore) (x : P.KO) (x' : T.Q.c.F) (hx : HEq x x') : D.what₂.embLL (P.kL x) = (x' : ℂ)

theorem **two_lt_half** (D : D7) : 2 < Module.finrank ℚ D.host.E / 2

theorem **nonempty_key** {N j : ℕ} {S : Sig A N j} {κ : Lv A} (s : FinalTheorems S κ) : Nonempty (KeySec s.Xo κ.c)

theorem **nonempty_kits_iff** (F : SrcFam A) (L : CmList A) (C : CxSt D.what₂.LL) (jC : ∀ (p : ℕ) [Fact p.Prime], PlL D p → (DV.KW p →+* ℂ)) : Nonempty (Kits F L C jC ngJunk) ↔ ∀ N j hN hj (κ : Lv A), Nonempty (F.src N j hN hj κ).Flag

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EmbeddingMemCoefficientField

theorem **twTiedC_iff** {D : D7} (F : TwFrameC D) : TwistFrameLowIndexCompatibility D F (lowPr D) ↔ NTwTied D (toFrame F)

theorem **twTiedC_of_tied** {D : D7} {F : TwFrame D} (T : NTwTied D F) : TwistFrameLowIndexCompatibility D (ofFrame F) (lowPr D)

theorem **twConjAt_of_choiceC** {D : D7} {C : NChoice D} (h : ChoiceC D C) : TwConjAt D C

theorem **choiceC_int** {D : D7} {C : NChoice D} (h : ChoiceC D C) (w : D.what₂.places.Pl) (hw : TwistedAt D w) (x : D.LL0) : (if C.sd w then D.ntw.emb0 (C.twI w) x else D.ntw.emb0b (C.twI w) x) ∈ (algebraMap (D.ntw.O0 (C.twI w)) (FractionRing (D.ntw.O0 (C.twI w)))).range ↔ fracOe D C w (if C.sd w then D.ntw.emb0 (C.twI w) x else D.ntw.emb0b (C.twI w) x) ∈ (algebraMap (C.Oe w) (FractionRing (C.Oe w))).range

theorem **ringSpec_onto** (S : RingSpec D cf b0 b1 Sx T) (w : HeightOneSpectrum (𝒪 ɪD.LL0)) : ∃ i, ((T.l i : ℕ) : 𝒪 ɪD.LL0) ∈ w.asIdeal

theorem **embedding_mem_coefficient_field** (S : RingSpec D cf b0 b1 Sx T) (i : T.Tw) (x : ɪD.LL0) : S.j0 i x ∈ (S.K0 i).1

theorem **chiTied** : NormalizedCharacterCompatibility (withRings hE F S ex h0 th) (chiData hE F S ex h0 th)

theorem **deltaRead** : DeltaRead (withRings hE F S ex h0 th) (charFrame hE F S ex h0 th)

theorem **tiedC** (low : Finset ℕ) (hlow : ∀ p ∈ low, placesOver D.host.E p ⊆ ɪSx) (hconj : ∀ P P', P ∈ Sx → IsConjPl D.host P P' → P' ∈ Sx) (hk : ∀ (i : (withRings hE F S ex h0 th).ntw.Tw) j b k P, P ∉ (withRings hE F S ex h0 th).ntw.S0 i → ∃ s : Multiset (FractionRing ((withRings hE F S ex h0 th).ntw.O i)), (LinearMap.charpoly ((withRings hE F S ex h0 th).ntw.hk i j b k P)).map (algebraMap ((withRings hE F S ex h0 th).ntw.O i) (FractionRing ((withRings hE F S ex h0 th).ntw.O i))) = (s.map fun a => Polynomial.X − Polynomial.C a).prod) (hB : ∀ i j b, LinearMap.range ((withRings hE F S ex h0 th).nexp.embB i j b) = (withRings hE F S ex h0 th).nexp.Bsup i j b ⊓ (NForms.V0 (withRings hE F S ex h0 th) i j b).restrictScalars ((withRings hE F S ex h0 th).ntw.O i)) (hs : ∀ i (s s' : F.Sig i), F.jS i s = F.jS i s' → F.side i s = F.side i s' → (∀ k P, 1 ≤ k → k ≤ D.R.r → IsSplitPl D.host P → P ∉ T.S0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EmbeddingOneApplyDivThree

theorem **unitMap_norm** (σ : K →+* ℂ) (W : Type) [NormedAddCommGroup W] [InnerProductSpace ℂ W] [FiniteDimensional ℂ W] (x : (unitLat K (Module.finrank ℂ W)).V) : Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (unitLat K (Module.finrank ℂ W)).b (unitLat K (Module.finrank ℂ W)).X (InfinitePlace.mk σ) x) = ‖unitMap σ W x‖

theorem **embedding_one_apply_div_three** (x : Fin 1 → ℚ) : map1 1 x = (x 0 : ℂ) / 3

theorem **one_not_mem** : (fun _ => 1 : Fin 1 → ℚ) ∉ (seq1 1).L

theorem **metric_one** (x : Fin 1 → ℚ) : Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (seq1 1).b (seq1 1).X (InfinitePlace.mk (algebraMap ℚ ℂ)) x) = ‖map1 1 x‖

theorem **good_one** : (seq1 1).Good

theorem **rank_off** : Module.finrank ℚ (seq1 0).V = 1 ∧ Module.finrank ℚ (seq1 2).V = 1

theorem **jM_metric_all** (hH : Holds R F) (j : ℕ) (x : (mjH Z j).V) : Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (mjH Z j).b (mjH Z j).X (InfinitePlace.mk D.what₂.embLL) x) = ‖jM Z j x‖

theorem **card_mjH** (hH : Holds R F) (j : ℕ) : Fintype.card (mjH Z j).ι = Module.finrank ℂ (D.what₂.Mj j)

theorem **exists_tied** {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (hP : Pins R F cG) (hH : Holds R F) (Xd : DivX D) : ∃ X : DivX D, X.geo = Xd.geo ∧ X.MjH = mjH Z ∧ ∃ JM, MjHTie D A3.A X (A3.indefDown (s0Of R F)) Z.jW JM

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EpsZeroCompOfIsTanAtG

theorem **epsZero_comp_of_isTanAtG** (hv : IsTanAtG t x v) : sec (epsZero k) ≫ v = x ≫ unitSec t.G

theorem **tanEv_add** (hv : IsTanAtG t x v) {V : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ V) (s s' : Γ(omega t.G, V)) : tanEv t hv h (s + s') = tanEv t hv h s + tanEv t hv h s'

theorem **tanEv_smul** (hv : IsTanAtG t x v) {V : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ V) (a : Γ(X, V)) (s : Γ(omega t.G, V)) : tanEv t hv h (a • s) = (Scheme.ΓSpecIso (.of k)).hom (x.appLE V ⊤ h a) * tanEv t hv h s

theorem **tanEv_resM** (hv : IsTanAtG t x v) {V V' : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ V') (h' : V' ≤ V) (s : Γ(omega t.G, V)) : tanEv t hv h (resM _ h' s) = tanEv t hv (h.trans (x.preimage_mono h')) s

theorem **tanEv_dUnit** (hv : IsTanAtG t x v) {V : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ V) {W : t.G.left.Opens} (hW : V ≤ unitSec t.G ⁻¹ᵘ W) (hv' : ⊤ ≤ v ⁻¹ᵘ W) (f : Γ(t.G.left, W)) : tanEv t hv h (dUnit t.G hW f) = ((Scheme.ΓSpecIso (.of (DualNumber k))).hom (v.appLE W ⊤ hv' f)).snd

theorem **isIso_lift_rangeSub** {X : Scheme.{0}} {F G : X.Modules} (θ : F ⟶ G) [Mono θ] : IsIso ((rangeSub θ).lift θ fun _ s => (s, rfl))

theorem **mem_ofBundle** {F : B.X.Modules} (θ : F ⟶ B.om) [Mono θ] (hF hθ) {V : B.X.Opens} {s : Γ(B.om, V)} : s ∈ (ofBundle θ hF hθ).N.sec V ↔ ∃ r, θ.app V r = s

theorem **isTop_top** : (top B).IsTop

theorem **ext** {L L' : Lat B} (h : ∀ V, L.N.sec V = L'.N.sec V) : L = L'

theorem **isTop_iff** {L : Lat B} : L.IsTop ↔ L = top B

theorem **mem_stepTy** {s : Finset (Step Y i₀)} {σ : E →+* ℂ} {r : ℕ} : r ∈ stepTy s σ ↔ ∃ t ∈ s, t.σ = σ ∧ t.k = r

theorem **isFlagCl_flagClosure** {Z : Mdl Y i₀} {s : Finset (Step Y i₀)} (h : ∃ Z', IsFlagCl Z s Z') : IsFlagCl Z s (flagClosure Z s)

theorem **isFlagCl_empty** {Z : Mdl Y i₀} (h : ∃ f, IsMap Z Z f) : IsFlagCl Z ∅ Z

theorem **IsFlagCl.exists_isMap** {Z Z₁ Z₂ : Mdl Y i₀} {s : Finset (Step Y i₀)} (h₁ : IsFlagCl Z s Z₁) (h₂ : IsFlagCl Z s Z₂) : ∃ f, IsMap Z₁ Z₂ f

theorem **mem_allSteps** {B : Mdl Y i₀} {L : Lat B} {t : Step Y i₀} : t ∈ allSteps L ↔ ∃ σ r hr, t = ⟨B, L, σ, r, hr⟩

theorem **stepTy_allSteps** {B : Mdl Y i₀} (L : Lat B) (σ : E →+* ℂ) : stepTy (allSteps L) σ = Finset.range (Dw.mult σ + 1)

theorem **allSteps_ne_empty** {B : Mdl Y i₀} (L : Lat B) : allSteps L ≠ ∅

theorem **At.flagCl_empty** (ct : At S κ p) (Z : ct.Mdl) : ct.flagCl Z ∅ = Z

theorem **At.flagCl_allSteps** (ct : At S κ p) : ct.flagCl ct.Xm (allSteps (Lat.top ct.Xm)) = ct.YY

theorem **At.isFlagCl_flagCl** (ct : At S κ p) (Z : ct.Mdl) (s : Finset (Step ct.Y ⟨κ.l, κ.c, κ.good p ct.hp⟩)) (hid : ∃ f, IsMap Z Z f) (hdr : IsDRFlag Z ct.YY → IsFlagCl Z (allSteps (Lat.top Z)) ct.YY) (hex : ∃ Z', IsFlagCl Z s Z') : IsFlagCl Z s (ct.flagCl Z s)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EqOfPowEqOneOfNormSubLt

theorem **eq_of_pow_eq_one_of_norm_sub_lt** {x y : Om l} {m : ℕ} (hm : ¬ l ∣ m) (hx : x ^ m = 1) (hy : y ^ m = 1) (h : ‖x - y‖ < 1) : x = y

theorem **norm_eq_one_of_pow_eq_one** {x : Om l} {m : ℕ} (hm : 0 < m) (hx : x ^ m = 1) : ‖x‖ = 1

theorem **kur_algHom_ext** {B : Type*} [Semiring B] [Algebra ℚ_[l] B] {f g : ɪ(MaximalUnramifiedExtension l) →ₐ[ℚ_[l]] B} (h : ∀ (c : ɪ(MaximalUnramifiedExtension l)) (m : ℕ), 0 < m → ¬ l ∣ m → c ^ m = 1 → f c = g c) : f = g

theorem **frobUr_root** (c : ɪ(MaximalUnramifiedExtension l)) (m : ℕ) (h0 : 0 < m) (hm : ¬ l ∣ m) (hc : c ^ m = 1) : frobUr l c = c ^ l

theorem **frobUr_bijective** : Function.Bijective (frobUr l)

theorem **mem_adjExt** (x : Om l) : x ∈ (adjExt x).1

theorem **mem_GK** {K : FinExt l} {g : GQ l} : g ∈ GK K ↔ ∀ x ∈ K.1, g x = x

theorem **conj_mem_GK** {K : FinExt l} {g h : GQ l} (hh : h ∈ GK (mapExt g K)) : g⁻¹ * h * g ∈ GK K

theorem **phiB_algebraMap** (c : ɪ(MaximalUnramifiedExtension l)) : P.φB (algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B c) = algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B (frobUr l c)

theorem **phiSt_algebraMap** (c : ɪ(MaximalUnramifiedExtension l)) : phiSt P (algebraMap ɪ(MaximalUnramifiedExtension l) (Bst P) c) = algebraMap ɪ(MaximalUnramifiedExtension l) (Bst P) (frobUr l c)

theorem **actSt_algebraMap** (g : GQ l) (c c' : ɪ(MaximalUnramifiedExtension l)) (h : g (c : Om l) = (c' : Om l)) : actSt P g (algebraMap ɪ(MaximalUnramifiedExtension l) (Bst P) c) = algebraMap ɪ(MaximalUnramifiedExtension l) (Bst P) c'

@[simp] theorem **onB_tmul** (f : End ℚ_[l] (Bst P)) (v : V) (b : Bst P) : onB P LL V f (v ⊗ₜ b) = v ⊗ₜ f b

@[simp] theorem **phiV_tmul** (v : V) (b : Bst P) : phiV P LL V (v ⊗ₜ b) = v ⊗ₜ phiSt P b

@[simp] theorem **monV_tmul** (v : V) (b : Bst P) : monV P LL V (v ⊗ₜ b) = v ⊗ₜ monSt P b

@[simp] theorem **mulV_tmul** (c : ɪ(MaximalUnramifiedExtension l)) (v : V) (b : Bst P) : mulV P LL V c (v ⊗ₜ b) = v ⊗ₜ (algebraMap ɪ(MaximalUnramifiedExtension l) (Bst P) c * b)

@[simp] theorem **crisIncl_tmul** (v : V) (b : P.R.B) : crisIncl P LL V (v ⊗ₜ b) = v ⊗ₜ C b

theorem **end_ext** {f g : End LL (V ⊗[ℚ_[l]] Bst P)} (h : ∀ v b, f (v ⊗ₜ b) = g (v ⊗ₜ b)) : f = g

theorem **phiV_injective** : Function.Injective (phiV P LL V)

theorem **crisIncl_injective** : Function.Injective (crisIncl P LL V)

theorem **monV_phiV** : monV P LL V * phiV P LL V = (l : LL) • (phiV P LL V * monV P LL V)

theorem **phiV_mulV** (c : ɪ(MaximalUnramifiedExtension l)) : phiV P LL V * mulV P LL V c = mulV P LL V (frobUr l c) * phiV P LL V

theorem **monV_mulV** (c : ɪ(MaximalUnramifiedExtension l)) : monV P LL V * mulV P LL V c = mulV P LL V c * monV P LL V

theorem **exists_monV_pow_eq_zero** (x : V ⊗[ℚ_[l]] Bst P) : ∃ n : ℕ, (monV P LL V ^ n) x = 0

@[simp] theorem **stRep_tmul** (g : ɪ(GK K)) (v : V) (b : Bst P) : stRep P ρ g (v ⊗ₜ b) = ρ g v ⊗ₜ actSt P (g : GQ l) b

theorem **stRep_crisIncl** (g : ɪ(GK K)) (x : V ⊗[ℚ_[l]] P.R.B) : stRep P ρ g (crisIncl P LL V x) = crisIncl P LL V (actVB (P.PD K) ρ g x)

theorem **phiV_stRep** (g : ɪ(GK K)) : phiV P LL V ∗ stRep P ρ g = stRep P ρ g ∗ phiV P LL V

theorem **monV_stRep** (g : ɪ(GK K)) : monV P LL V ∗ stRep P ρ g = stRep P ρ g ∗ monV P LL V

theorem **stRep_mulV** (g : ɪ(GK K)) (c c' : ɪ(MaximalUnramifiedExtension l)) (h : (g : GQ l) (c : Om l) = (c' : Om l)) : stRep P ρ g ∗ mulV P LL V c = mulV P LL V c' ∗ stRep P ρ g

theorem **mem_Dst** {x : V ⊗[ℚ_[l]] Bst P} : x ∈ Dst P ρ ↔ ∀ g, stRep P ρ g x = x

theorem **mem_Dst_resTo** {K' : FinExt l} (h : K.1 ≤ K'.1) {x : V ⊗[ℚ_[l]] Bst P} : x ∈ Dst P (resTo ρ h) ↔ ∀ g : ɪ(GK K), (g : GQ l) ∈ GK K' → stRep P ρ g x = x

theorem **mem_Dpst** {x : V ⊗[ℚ_[l]] Bst P} : x ∈ Dpst P ρ ↔ ∃ K' : FinExt l, ∀ g : ɪ(GK K), (g : GQ l) ∈ GK K' → stRep P ρ g x = x

theorem **Dst_resTo_le** {K' : FinExt l} (h : K.1 ≤ K'.1) : Dst P (resTo ρ h) ≤ Dpst P ρ

theorem **Dpst_resTo** {K' : FinExt l} (h : K.1 ≤ K'.1) : Dpst P (resTo ρ h) = Dpst P ρ

theorem **mem_Dpst_of_commute** {T : End LL (V ⊗[ℚ_[l]] Bst P)} (hT : ∀ g, T ∗ stRep P ρ g = stRep P ρ g ∗ T) (x : V ⊗[ℚ_[l]] Bst P) (hx : x ∈ Dpst P ρ) : T x ∈ Dpst P ρ

theorem **stRep_mem_Dpst** (g : ɪ(GK K)) (x : V ⊗[ℚ_[l]] Bst P) (hx : x ∈ Dpst P ρ) : stRep P ρ g x ∈ Dpst P ρ

theorem **mulV_mem_Dpst** (c : ɪ(MaximalUnramifiedExtension l)) (x : V ⊗[ℚ_[l]] Bst P) (hx : x ∈ Dpst P ρ) : mulV P LL V c x ∈ Dpst P ρ

@[simp] theorem **coe_phiD** (x : ɪ(Dpst P ρ)) : (phiD P ρ x : V ⊗[ℚ_[l]] Bst P) = phiV P LL V x

@[simp] theorem **coe_monD** (x : ɪ(Dpst P ρ)) : (monD P ρ x : V ⊗[ℚ_[l]] Bst P) = monV P LL V x

@[simp] theorem **coe_actD** (g : ɪ(GK K)) (x : ɪ(Dpst P ρ)) : (actD P ρ g x : V ⊗[ℚ_[l]] Bst P) = stRep P ρ g x

@[simp] theorem **coe_mulD** (c : ɪ(MaximalUnramifiedExtension l)) (x : ɪ(Dpst P ρ)) : (mulD P ρ c x : V ⊗[ℚ_[l]] Bst P) = mulV P LL V c x

theorem **phiD_actD** (g : ɪ(GK K)) : phiD P ρ ∗ actD P ρ g = actD P ρ g ∗ phiD P ρ

theorem **monD_actD** (g : ɪ(GK K)) : monD P ρ ∗ actD P ρ g = actD P ρ g ∗ monD P ρ

theorem **monD_phiD** : monD P ρ ∗ phiD P ρ = (l : LL) • (phiD P ρ ∗ monD P ρ)

theorem **phiD_mulD** (c : ɪ(MaximalUnramifiedExtension l)) : phiD P ρ ∗ mulD P ρ c = mulD P ρ (frobUr l c) ∗ phiD P ρ

theorem **monD_mulD** (c : ɪ(MaximalUnramifiedExtension l)) : monD P ρ ∗ mulD P ρ c = mulD P ρ c ∗ monD P ρ

theorem **actD_mulD** (g : ɪ(GK K)) (c c' : ɪ(MaximalUnramifiedExtension l)) (h : (g : GQ l) (c : Om l) = (c' : Om l)) : actD P ρ g ∗ mulD P ρ c = mulD P ρ c' ∗ actD P ρ g

theorem **phiD_injective** : Function.Injective (phiD P ρ)

theorem **smul_toUr** (c : ɪ(MaximalUnramifiedExtension l)) (x : ɪ(Dpst P ρ)) : c • toUr P ρ x = toUr P ρ (mulD P ρ c x)

theorem **isPst_iff** : IsPst P ρ ↔ FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ) ∧ finrank ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ) = finrank ℚ_[l] V

theorem **IsPst.finiteDimensional** (h : IsPst P ρ) : FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)

theorem **phiD_bijective** [FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)] : Function.Bijective (phiD P ρ)

theorem **isNilpotent_monD** [FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)] : IsNilpotent (monD P ρ)

theorem **exists_actD_eq_one** [FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)] : ∃ K' : FinExt l, ∀ g : ɪ(GK K), (g : GQ l) ∈ GK K' → (∀ c, actD P ρ g ∗ mulD P ρ c = mulD P ρ c ∗ actD P ρ g) → actD P ρ g = 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EqZeroOfApplySelf

theorem **ordGe_pow_iff** {R : Type} [CommRing R] [IsDomain R] {u : R} (h0 : u ≠ 0) (hu : ¬ IsUnit u) (e a b n : ℕ) (i : ℚ) : OrdGe (u ^ e) i n (Ideal.span {u ^ a}) (Ideal.span {u ^ b}) ↔ i ∗ ((a : ℚ) − b) ≤ e ∗ n

theorem **minVal_eval** (B : X → X → ℝ) : (InvFun.minVal O).eval X B = sInf {x | (∃ y, B y y = x) ∧ x ≠ B 0 0}

theorem **minVal_linearOn** (γ : LinearMap.BilinForm ℤ X) : (InvFun.minVal O).LinearOn ![γ]

theorem **eq_zero_of_apply_self** {β : LinearMap.BilinForm ℤ (𝒪 E)} (hβ : β ∈ levelForms H 1 (𝒪 E)) (hp : ∀ y, 0 ≤ β y y) {y₀ : 𝒪 E} (hy : y₀ ≠ 0) (h : β y₀ y₀ = 0) (y : 𝒪 E) : β y y = 0

theorem **exists_isPiece** (M : ℕ) (hM : M ≠ 0) : 0 < (InvFun.minVal (𝒪 E)).evalZ ((M : ℤ) • HostModel.trForm H) ∧ (InvFun.minVal (𝒪 E)).LinearOn ![(M : ℤ) • HostModel.trForm H] ∧ ∃ ℓ, IsPiece H (InvFun.minVal (𝒪 E)) 1 ![(M : ℤ) • HostModel.trForm H] ℓ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EtaJUntw

theorem **extSub_zs** (R : RealDat D A Ks) (F : Face D A) : extSub R F ((useReal R).toCx (useFace R F)).zs = (R.toCx F).zs

theorem **restrictedFace_closedNumbers** : ((useFace R F).Zsc ((useReal R).toCx (useFace R F)).zs).F0 = (F.Zsc (R.toCx F).zs).F0

theorem **restrictedFace_eigenvalueMainTerm** : ((useFace R F).Zsc ((useReal R).toCx (useFace R F)).zs).l6m = (F.Zsc (R.toCx F).zs).l6m

theorem **nonzeroIndexPlaces_iff** (w : HeightOneSpectrum (𝒪 D.what₂.LL)) : (useReal R).NonzeroIndexPlace (useFace R F) w ↔ R.NonzeroIndexPlace F w

theorem **partAt_use** (b : Bool) (i : Fin D.R.m) (p : A.fam.base.Ix) (v : InfinitePlace D.what₂.LL) : (useReal R).PartAt (useFace R F) b i p v ↔ R.PartAt F b i p v

theorem **CharAll.use** (h : CharAll R F) : (useReal R).Char (useFace R F) where

theorem **qGuard_use** (h : QGuard R) : QGuard (useReal R)

theorem **char₀** : r.R₀.Char r.F₀

theorem **guard₀** : QGuard r.R₀

theorem **rho₀** : RhoTie r.R₀

theorem **bar₀** : BarTie r.fw₀

theorem **trueColumnChoices_heckeChar_reference_eq_one** : xi D.what₂.mu0 = 1

theorem **nonempty_QPl_iff** (μ : D.what₂.Tup) : Nonempty (r.X.QPl μ) ↔ ∃ v, ¬ (xi μ).UnrAt v

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EventuallyCongOne

theorem **lT_apply** (u : ι → GL (Fin n) ℂ) (F : Frames G → ℂ) (y : Frames G) : lT G u F y = F (frL G u y)

theorem **lT_mul** (u v : ι → GL (Fin n) ℂ) (F : Frames G → ℂ) : lT G (u * v) F = lT G v (lT G u F)

theorem **lT_one** (F : Frames G → ℂ) : lT G 1 F = F

theorem **lT_mem** (lam : ι → DomWeight n) (u : ι → GL (Fin n) ℂ) {F : Frames G → ℂ} (hF : F ∈ ValC G lam) : lT G u F ∈ ValC G lam

theorem **continuous_lT** (lam : ι → DomWeight n) [FiniteDimensional ℂ (ValC G lam)] (l : (Frames G → ℂ) →ₗ[ℂ] ℂ) {F : Frames G → ℂ} (hF : F ∈ ValC G lam) {X : Type} [TopologicalSpace X] {u : X → ι → GL (Fin n) ℂ} (hu : ∀ τ, Continuous fun x => (u x τ : Matrix (Fin n) (Fin n) ℂ)) : Continuous fun x => l (lT G (u x) F)

theorem **tr_mul** (b : S.β) (g h : S.G b) (F : Fr emb σ n → ℂ) : S.tr emb σ b (g * h) F = S.tr emb σ b g (S.tr emb σ b h F)

theorem **valL_eq** (hφ : φ ∈ S.Fi emb σ lam) (c : S.Cl) (g : S.G c.1) : valL S emb σ c.1 (g • S.rep c) φ = S.tr emb σ c.1 g (φ c)

theorem **valL_smul** (hφ : φ ∈ S.Fi emb σ lam) (b : S.β) (g : S.G b) (d : S.D b) : valL S emb σ b (g • d) φ = S.tr emb σ b g (valL S emb σ b d φ)

theorem **valL_rep** (hφ : φ ∈ S.Fi emb σ lam) (c : S.Cl) : valL S emb σ c.1 (S.rep c) φ = φ c

theorem **valL_mem** (hφ : φ ∈ S.Fi emb σ lam) (b : S.β) (d : S.D b) : valL S emb σ b d φ ∈ ValC (Gr emb σ n) lam

theorem **prFi_of_mem** (hφ : φ ∈ S.Fi emb σ lam) : prFi S emb σ lam φ = ⟨φ, hφ⟩

theorem **finN_apply** (g : GA D V) (i j : Fin n) : (finN V hn g : Matrix (Fin n) (Fin n) (AfE D.host.F D.host.E)) i j = (finE V g).1 (Fin.cast hn.symm i) (Fin.cast hn.symm j)

theorem **infN_apply** (g : GA D V) (τ : ι) (i j : Fin n) : (infN θ V hn t g τ : Matrix (Fin n) (Fin n) ℂ) i j = infMat D.sp.cm.sig V.gram (θ (t τ)) g (Fin.cast hn.symm i) (Fin.cast hn.symm j)

theorem **finN_of_inf** {k : GA D V} (hk : k ∈ GinfC V) : finN V hn k = 1

theorem **infN_of_fin** {k : GA D V} (hk : k ∈ GfC V) : infN θ V hn t k = 1

theorem **swapAE_sig** (z : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F ⊗[D.host.F] D.host.E) : swapAE D (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ D.sp.cm.sig z) = Formalism.UG.sigR D.sp.cm.sig _ (swapAE D z)

theorem **JR_one** (hG : V.gram = 1) (A : Type) [CommRing A] [Algebra D.host.F A] : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR (F

theorem **finN_mem** (hG : V.gram = 1) (g : GA D V) : finN V hn g ∈ UAd D.sp.cm n

theorem **exists_point** (hθ : IsInfFam θ) (hG : V.gram = 1) {x : GL (Fin n) (AfE D.host.F D.host.E)} (hx : x ∈ UAd D.sp.cm n) (u : ι → Matrix (Fin n) (Fin n) ℂ) (hu : ∀ τ, (u τ)ᴴ * u τ = 1) : ∃ g : GA D V, finN V hn g = x ∧ ∀ τ, (infN θ V hn t g τ : Matrix (Fin n) (Fin n) ℂ) = u τ

theorem **exists_rat** (hθ : IsInfFam θ) (hG : V.gram = 1) {γ : GA D V} (hγ : γ ∈ GQ V) : ∃ δ : URat D.sp.cm n, finN V hn γ = Formalism.UG.bc (Formalism.UG.inclE (F

theorem **continuous_infN** (hθ : IsInfFam θ) (τ : ι) : Continuous fun g => (infN θ V hn t g τ : Matrix (Fin n) (Fin n) ℂ)

theorem **eventually_congOne** (x₀ : GL (Fin n) (AfE D.host.F D.host.E)) (a : 𝒪 D.host.E) : ∀ᶠ κ in 𝒩 (1 : GA D V), CongOne a (x₀⁻¹ * finN V hn κ * x₀)

theorem **isAutFunDef_of** [Fintype ι] (hθ : IsInfFam θ) (G : ι → Matrix (Fin n) (Fin n) ℂ) (lam : ι → DomWeight n) [FiniteDimensional ℂ (ValC G lam)] (Φ : GA D V → Frames G → ℂ) (h1 : ∀ g, Φ g ∈ ValC G lam) (h2 : ∀ γ ∈ GQ V, ∀ g, lT G (infN θ V hn t γ) (Φ (γ * g)) = Φ g) (h3 : ∀ᶠ κ in 𝒩 (1 : GA D V), ∀ g, Φ (g * κ) = Φ g) (h4 : ∀ k ∈ GinfC V, ∀ g, Φ (g * k) = Φ g) (l : (Frames G → ℂ) →ₗ[ℂ] ℂ) : IsAutFunDef V fun g => l (lT G (infN θ V hn t g) (Φ g))

theorem **isAniso_of_gram** (hG : V.gram = 1) : IsAniso V

theorem **mem_cuspS** (hθ : IsInfFam θ) (hG : V.gram = 1) (hs : V.SigOK) (hd : V.IsDefinite) {φ : GA D V → ℂ} (h : IsAutFunDef V φ) : φ ∈ CuspS θ V

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EventuallyThickCardLeCyclicOrder

theorem **AdmStd.adm** {C : FC} (h : AdmStd C) : admitted C

theorem **AdmStd.CTu** {C : FC} (h : AdmStd C) : C.CTu = 25 ∧ C.CTu = 2 * C.P.CYL + 3 + 4

theorem **AdmStd.CP** {C : FC} (h : AdmStd C) : 1000 ≤ C.T.CP

theorem **admissibleStandard_constants** (CT2 Cdg C5 : ℝ) (full : Prop) {C8 p m l : ℝ} (h8 : 1 ≤ C8) (hp : 1000 ≤ p) (hm : 1088 ≤ m) (hl : 1 ≤ l) : AdmStd (consts CT2 C8 Cdg C5 full p m l)

theorem **iota_inv** (φ : Formalism.HC.Frame) (t : φ.A) : φ.iotaInv (φ.tau1 t).2 = t

theorem **top_ex** {R : Ranks} (H : Host R) {g : R.GG} (hg : g ∈ H.topPlaces) : ∃ t ∈ H.TT, g = R.tau1 t

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.EvlTrueIncTInjective

theorem **evlTrue_resB** (u : RayEv.LieGt (ct.dat y) (sideEmb D σ)) : (evlTrue ct).resB y σ u = RayEv.resB (ct.dat y) (sideEmb D σ) u

theorem **evlTrue_incT** (u : RayEv.LieT (ct.dat y) (sideEmb D σ)) : (evlTrue ct).incT y σ u = RayEv.incT (ct.dat y) (sideEmb D σ) u

theorem **evlTrue_kapGt** : (evlTrue ct).kapGt y σ = RayEv.kapGt (ct.dat y) (sideEmb D σ)

theorem **evlTrue_kapB** : (evlTrue ct).kapB y σ = RayEv.kapB (ct.dat y) (sideEmb D σ)

theorem **evlTrue_incT_injective** : Function.Injective ((evlTrue ct).incT y σ)

theorem **evlTrue_range_incT** : ((evlTrue ct).incT y σ).range = ((evlTrue ct).resB y σ).ker

theorem **evlTrue_kap** (u u' : (evlTrue ct).LieGt y σ) : (evlTrue ct).kapGt y σ u u' = (evlTrue ct).kapB y σ ((evlTrue ct).resB y σ u) ((evlTrue ct).resB y σ u')

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsAllLifts

theorem **exists_place** (B : Base) {l : ℕ} (hl : l.Prime) : ∃ w : B.places.Pl, B.places.resChar w = l

theorem **tau_nonempty** (B : Base) : Nonempty (Tau B.R)

theorem **exists_inert** {R : Ranks} (H : Host R) : ∃ l : ℕ, l.Prime ∧ 5 ≤ l ∧ (H.cls l = .signal ∨ H.cls l = .inertClass)

theorem **exists_of_jet** (B : Base) (C : Cols B) (vol : @HSpace B.host.E _ _ B.host.cmStar → ℝ) (hvol : ∀ W, 0 < vol W) (cd : CDHaar) (jet : ThetaJet B.R B.host) (gx : GlobalExtra B.LL (Tau B.R)) (h : ∀ i, ∃ u, Nonempty ((C.mem u).C.cls (jet.jOf i − 1)) ∧ Nonempty ((C.mem u).C.cls (jet.jOf i))) (h1 : ∀ i, ∃ u, Nonempty ((C.mem u).C.cls (jet.j1Of i − 1)) ∧ Nonempty ((C.mem u).C.cls (jet.j1Of i))) : ∃ G : GJStage B C, G.vol = vol ∧ G.cd = cd ∧ G.jet = jet ∧ G.gx = gx ∧ ∀ u w h, (G.gat u w h).FinClasses

theorem **nonempty_iff** (B : Base) (C : Cols B) : Nonempty (GJStage B C) ↔ Nonempty CDHaar

theorem **exists_all_lifts** (B : Base) (C : Cols B) (cd : CDHaar) (u : B.Word) (hu : ∀ j, j ≤ B.R.m → Nonempty ((C.mem u).C.cls j)) : ∃ G : GJStage B C, G.cd = cd ∧ Nonempty G.jet.Sit ∧ (∀ j, 1 ≤ j → j ≤ B.R.m → ∃ i, G.jet.jOf i = j) ∧ ∀ u w h, (G.gat u w h).FinClasses

theorem **sit_spec** {B : Base} {C : Cols B} (G : GJStage B C) (i : G.jet.Sit) : Nonempty ((C.mem (G.jetU i)).C.cls (G.jet.jOf i − 1)) ∧ Nonempty ((C.mem (G.jetU i)).C.cls (G.jet.jOf i)) ∧ 5 ≤ B.places.resChar (G.jetW i) ∧ (B.host.cls (B.places.resChar (G.jetW i)) = .signal ∨ B.host.cls (B.places.resChar (G.jetW i)) = .inertClass)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsBaseJointFld

theorem **exists_psi** (P : Pre) (l : ℕ) (hl : l.Prime) (A : Finset ℕ) : ∃ (Q : Finset (HeightOneSpectrum (𝒪 P.host.E))) (k : ℕ) (ψ : HChar P.host.E), IsPsi (H

theorem **exists_family** (P : Pre) (f : Fr P) (pl : Finset ℕ) (A : ℕ → Finset ℕ) : ∃ Ξ : Set (HChar P.host.E), ∀ (B : Base) (h : B.toPre = P), ∃ T : TwDat B, T.pl = pl ∧ T.Avoids A ∧ (∀ w, B.places.resChar w ∈ pl → T.psi w ≠ 1) ∧ T.set (h ▸ f) = h ▸ Ξ

theorem **exists_fixed** (B : Base) (S : Finset ℂ) (hS : ∀ z ∈ S, IsAlgebraic ℚ z) : ∃ B' : Base, B'.toPre = B.toPre ∧ B'.KsC = B.KsC ∧ ∀ z ∈ S, z ∈ B'.fixedCoefficientFieldInComplex

theorem **exists_over** (K : Ctx) : ∃ P, K.Over P

theorem **exists_muIs** (B : Base) (hB : StagesInhab.HasLetters B.toPre) (f : Fr B.toPre) (T : TwDat B) : ∃ (C : Cols B) (W : _) (V : _) (S : _) (N : _) (G : _) (xi : C.Tup → HChar B.host.E), MuIs ⟨B, C, W, V, S, N, G⟩ f T xi

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsChartOfMaximal

theorem **exists_maximal** {X : Scheme.{0}} (z : X) : ∃ η : X, η ⤳ z ∧ ∀ y : X, y ⤳ η → y = η

theorem **ext_maximal** {X : Scheme.{0}} [IsReduced X] {U : X.Opens} {a b : Γ(X, U)} (h : ∀ η ∈ U, (∀ y : X, y ⤳ η → y = η) → ∃ (k : Type) (_ : Field k) (w : Sp k ⟶ X), w (closedPoint k) = η ∧ valF w U a = valF w U b) : a = b

theorem **valF_map** {F F' : Type} [Field F] [Field F'] (f : F →+* F') {X : Scheme.{0}} {w : Sp F ⟶ X} {U : X.Opens} (h : ⊤ ≤ w ⁻¹ᵁ U) (a : Γ(X, U)) : valF (spMap f ≫ w) U a = f (valF w U a)

theorem **valF_natCast** {F : Type} [Field F] {X : Scheme.{0}} {w : Sp F ⟶ X} {U : X.Opens} (h : ⊤ ≤ w ⁻¹ᵁ U) (n : ℕ) : valF w U (n : Γ(X, U)) = n

theorem **valF_one** {F : Type} [Field F] {X : Scheme.{0}} {w : Sp F ⟶ X} {U : X.Opens} (h : ⊤ ≤ w ⁻¹ᵁ U) : valF w U 1 = 1

theorem **maximal_of_stalk** {X : Scheme.{0}} {x : X} {K : Type} [Field K] {f : X.presheaf.stalk x →+* K} (hf : Function.Injective f) (y : X) (hy : y ⤳ (spMap f ≫ X.fromSpecStalk x) (closedPoint K)) : y = (spMap f ≫ X.fromSpecStalk x) (closedPoint K)

theorem **mem_range_of_norm_le** {p : ℕ} (hp : p.Prime) {R K : Type} [CommRing R] [IsDomain R] [IsLocalRing R] [IsNoetherianRing R] [IsIntegrallyClosed R] [Field K] [Algebra R K] [IsFractionRing R K] [CharZero K] (hR : (p : R) ∈ maximalIdeal R) (hc : Cardinal.mk R ≤ Cardinal.continuum) {κ : K} (h : ∀ (C : SCF p) (ψ : R →+* ɪ(intO ɪC.S)) (θ : K →+* ɪC.S), θ.comp (algebraMap R K) = (intO ɪC.S).subtype.comp ψ → ‖θ κ‖ ≤ 1) : ∃ r, algebraMap R K r = κ

theorem **exists_chart_of_maximal** {η : Z.X} (h : ∀ y : Z.X, y ⤳ η → y = η) : ∃ a, η ∈ (Z.U a).1

theorem **ext_chart** {U : Z.X.Opens} {s t : Γ(Z.X, U)} (h : ∀ (k : Type) [Field k] (w : Sp k ⟶ Z.X) a, ⊤ ≤ w ⁻¹ᵁ (Z.U a).1 → ⊤ ≤ w ⁻¹ᵁ U → valF w U s = valF w U t) : s = t

theorem **exists_special** (z : Z.X) : ∃ x : Z.X, z ⤳ x ∧ (p : Z.X.residueField x) = 0

theorem **exists_extension** {U : Z.X.Opens} (f : ∀ a, Γ(Z.X, (Z.U a).1 ⊓ U)) (hf : ∀ (k : Type) [Field k] (w : Sp k ⟶ Z.X) a a', ⊤ ≤ w ⁻¹ᵁ (Z.U a).1 → ⊤ ≤ w ⁻¹ᵁ (Z.U a').1 → ⊤ ≤ w ⁻¹ᵁ U → valF w _ (f a) = valF w _ (f a')) (hb : ∀ (y : Z.Pt) a, ⊤ ≤ y.gen ⁻¹ᵁ (Z.U a).1 → ⊤ ≤ y.z ⁻¹ᵁ U → ‖valF y.gen _ (f a)‖ ≤ 1) {x : Z.X} (hx : x ∈ U) (hpx : (p : Z.X.residueField x) = 0) : ∃ (V : Z.X.Opens) (t : Γ(Z.X, V)), x ∈ V ∧ V ≤ U ∧ ∀ (k : Type) [Field k] (w : Sp k ⟶ Z.X) a, ⊤ ≤ w ⁻¹ᵁ (Z.U a).1 → ⊤ ≤ w ⁻¹ᵁ V → valF w V t = valF w _ (f a)

theorem **val_ne_zero** (M : LineModel Z b st x) {k : Type} [Field k] {w : Sp k ⟶ Z.X} {a : Fin Z.m} {i : Fin M.L.n} (ha : ⊤ ≤ w ⁻¹ᵁ (Z.U a).1) (hi : ⊤ ≤ w ⁻¹ᵁ M.L.U i) : valF w _ (M.r a i).val ≠ 0

theorem **cross_eq** (hty : ∀ τ k, k ≤ Dw.mult τ → k ∈ Z.ty τ) (M : LineModel Z b st x) (M' : LineModel Z b' st' x) {k : Type} [Field k] {w : Sp k ⟶ Z.X} {a a' : Fin Z.m} {i : Fin M.L.n} {j : Fin M'.L.n} (ha : ⊤ ≤ w ⁻¹ᵁ (Z.U a).1) (ha' : ⊤ ≤ w ⁻¹ᵁ (Z.U a').1) (hi : ⊤ ≤ w ⁻¹ᵁ M.L.U i) (hj : ⊤ ≤ w ⁻¹ᵁ M'.L.U j) : valF w _ (M'.r a j).val * valF w _ (M.r a' i).val = valF w _ (M'.r a' j).val * valF w _ (M.r a i).val

theorem **Bdd.symm** (h : Bdd Z b b' st st') : Bdd Z b' b st' st

theorem **bdd_step** (st : ((E →+* ℂ) × ℕ) ⊕ ℕ) : Bdd Z false true st st

theorem **IsQuot.mul_eq_one** {i : Fin M.L.n} {j : Fin M'.L.n} {V V' : Z.X.Opens} {s : Γ(Z.X, V)} {s' : Γ(Z.X, V')} (h : IsQuot M M' i j 1 s) (h' : IsQuot M' M j i 1 s') (hi : V ≤ M.L.U i) {k : Type} [Field k] {w : Sp k ⟶ Z.X} (hw : ⊤ ≤ w ⁻¹ᵁ V) (hw' : ⊤ ≤ w ⁻¹ᵁ V') : valF w V s * valF w V' s' = 1

theorem **exists_isQuot_special** (hB : Bdd Z b b' st st') (i : Fin M.L.n) (j : Fin M'.L.n) {z : Z.X} (hz : z ∈ M.L.U i ⊓ M'.L.U j) (hpz : (p : Z.X.residueField z) = 0) : ∃ (m : ℕ) (V : Z.X.Opens) (t : Γ(Z.X, V)), z ∈ V ∧ V ≤ M.L.U i ⊓ M'.L.U j ∧ IsQuot M M' i j (p ^ m) t

theorem **exists_isQuot** (hB : Bdd Z b b' st st') (i : Fin M.L.n) (j : Fin M'.L.n) {z : Z.X} (hz : z ∈ (nmodel Z).gen ⊓ (M.L.U i ⊓ M'.L.U j)) : ∃ (V : Z.X.Opens) (s : Γ(Z.X, V)), z ∈ V ∧ V ≤ (nmodel Z).gen ⊓ (M.L.U i ⊓ M'.L.U j) ∧ IsQuot M M' i j 1 s

theorem **exists_unit** (hB : Bdd Z b b' st st') (i : Fin M.L.n) (j : Fin M'.L.n) : ∃ u : Γ(Z.X, (nmodel Z).gen ⊓ (M.L.U i ⊓ M'.L.U j))ˣ, IsQuot M M' i j 1 u.val

theorem **IsGenId.unique** {u u' : GenIso Z M.L M'.L} (h : IsGenId M M' u) (h' : IsGenId M M' u') : u = u'

theorem **exists_isGenId** (hty : ∀ τ k, k ≤ Dw.mult τ → k ∈ Z.ty τ) (hB : GenId.Bdd Z b b' st st') (M : LineModel Z b st x) (M' : LineModel Z b' st' x) : ∃ u, IsGenId M M' u

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsDivStage

theorem **ext_one_tmul** [Fintype ι] (σ : K →+* ℂ) (e : V →ₛₗ[σ] M) (b : Module.Basis ι K V) (x : V) : letI

theorem **ext_inj** (σ : K →+* ℂ) (e : V →ₛₗ[σ] M) (b : Module.Basis ι K V) (h : LinearIndependent ℂ (fun a => e (b a))) : Function.Injective (ext σ e b)

theorem **li_comp** [Fintype ι] {V' ι' : Type} [AddCommGroup V'] [Module K V'] (σ : K →+* ℂ) (e : V →ₛₗ[σ] M) (b : Module.Basis ι K V) (h : LinearIndependent ℂ (fun a => e (b a))) (ψ : V' →ₗ[K] V) (hψ : Function.Injective ψ) (b' : Module.Basis ι' K V') : LinearIndependent ℂ (fun a => e (ψ (b' a)))

theorem **ker_rational** [Fintype ι] {J : Type} [Fintype J] (σ : K →+* ℂ) (e : V →ₛₗ[σ] M) (b : Module.Basis ι K V) (h : LinearIndependent ℂ (fun a => e (b a))) (N : J → M →ₗ[ℂ] M) (n : J → V →ₗ[K] V) (hn : ∀ j x, e (n j x) = N j (e x)) (φ : M) (hφ : φ ∈ Submodule.span ℂ (Set.range (fun a => e (b a)))) (h0 : ∀ j, N j φ = 0) : φ ∈ Submodule.span ℂ (e '' {x | ∀ j, n j x = 0})

theorem **span_image_le** [Fintype ι] (σ : K →+* ℂ) (e : V →ₛₗ[σ] M) (b : Module.Basis ι K V) (x : V) : e x ∈ Submodule.span ℂ (Set.range (fun a => e (b a)))

theorem **exists_frame** {n : ℕ} (v : InfinitePlace K) (b : Module.Basis (Fin n) K V) {E : Type} [NormedAddCommGroup E] [InnerProductSpace ℂ E] (e : V →ₛₗ[v.embedding] E) (h : LinearIndependent ℂ (fun a => e (b a))) : ∃ X : Matrix (Fin n) (Fin n) ℂ, IsUnit X.det ∧ ∀ x, Formalism.HermitianLatticesTheory.cnorm (Matrix.mulVec X⁻¹ (Formalism.HermitianLatticesTheory.vecAt b v x)) = ‖e x‖

theorem **fco_li** {ι : Type} [Fintype ι] [DecidableEq ι] (b : Module.Basis ι K V) (X : InfinitePlace K → Matrix ι ι ℂ) (v : InfinitePlace K) (hX : IsUnit (X v).det) : LinearIndependent ℂ (fun a => fco b X v (b a))

theorem **exists_frames** {n : ℕ} (A : HLat K) (hA : ∀ v, IsUnit (A.X v).det) (b : Module.Basis (Fin n) K V) (ψ : V →ₗ[K] A.V) (hψ : Function.Injective ψ) : ∃ X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ, (∀ v, IsUnit (X v).det) ∧ ∀ v x, Formalism.HermitianLatticesTheory.normAt b X v x = Formalism.HermitianLatticesTheory.normAt A.b A.X v (ψ x)

theorem **coord_basisFun** (e : K →+* ℂ) {W : Type} [AddCommGroup W] [Module ℂ W] {n : ℕ} (c : Fin n → W) (i : Fin n) : coord e c (Pi.basisFun K (Fin n) i) = c i

theorem **normAt_id** (e : K →+* ℂ) {W : Type} [NormedAddCommGroup W] [InnerProductSpace ℂ W] {n : ℕ} (c : OrthonormalBasis (Fin n) ℂ W) (x : Fin n → K) : Formalism.HermitianLatticesTheory.normAt (Pi.basisFun K (Fin n)) (fun _ => 1) (InfinitePlace.mk e) x = ‖coord e c x‖

theorem **stdLat_good** (n : ℕ) : (stdLat K n).Good

theorem **exists_smul_mem** {K : Type} [Field K] [NumberField K] {V : Type} [AddCommGroup V] [Module K V] [Module (𝓞 K) V] [IsScalarTower (𝓞 K) K V] (L : Submodule (𝓞 K) V) (hL : Submodule.span K (L : Set V) = ⊤) (x : V) : ∃ c : 𝓞 K, c ≠ 0 ∧ c • x ∈ L

theorem **subLat_good** (A : HLat K) (hA : A.Good) (S : Submodule K A.V) (X) (hX : ∀ v, IsUnit (X v).det) : (subLat A S X).Good

theorem **nm_rat** (v : A.vT) (x : (M.lat 0).V) : ∃ y, M.emb 0 y = A.Nm v (M.emb 0 x)

theorem **mram_le_span** : A.Mram ≤ Submodule.span ℂ (M.emb 0 '' (M.ramSub : Set (M.lat 0).V))

theorem **ramEmb_basis** {ι : Type} [Fintype ι] (b : Module.Basis ι LL M.ramSub) : LinearIndependent ℂ (fun a => ramEmb M (b a)) ∧ Submodule.span ℂ (Set.range (fun a => ramEmb M (b a))) = ⊤

theorem **exists_divStage** (B : Base) (C : Cols B) (W : RowsRest B C) : ∃ V : DivStage B C W, V.div.AllGood ∧ ∀ u, (V.div.Nu u).L = ((C.fm u).lat 0).L.comap ((V.nuIn u).restrictScalars (𝓞 B.LL))

theorem **RR_finrank** {B : Base} {C : Cols B} {W : RowsRest B C} (V : DivStage B C W) : Module.finrank B.LL V.div.RR.V = W.Drows

theorem **Nu_finrank** {B : Base} {C : Cols B} {W : RowsRest B C} (V : DivStage B C W) (u : B.Word) : Module.finrank B.LL (V.div.Nu u).V = (C.mem u).A.du

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsEmbeddingOfRingTie

theorem **exists_place_over** (F : Type) [Field F] [NumberField F] {p : ℕ} (hp : p.Prime) : ∃ v : HeightOneSpectrum (𝓞 F), (p : 𝓞 F) ∈ v.asIdeal

theorem **orbits_pos** {D : D7} {H : LevelData D} {u : D.what₂.Word} {w : D.what₂.places.Pl} {hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)} (h : LocalIs D H u w hg) : 0 < (D.gat u w hg).P.nV

theorem **rank_eq** {D : D7} {H : LevelData D} {u : D.what₂.Word} {w : D.what₂.places.Pl} {hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)} (h : LocalIs D H u w hg) : (D.gat u w hg).P.m + 1 = D.R.k

theorem **finClasses_and_trivStab** {D : D7} {H : LevelData D} {u : D.what₂.Word} {w : D.what₂.places.Pl} {hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)} (h : LocalIs D H u w hg) : (D.gat u w hg).FinClasses ∧ (Cmp.TrivStabU (H u) → (D.gat u w hg).TrivStab)

theorem **orbit_length** {D : D7} {Gt : GatTies D} {c : LatCmp D} (h : IsConcrete D Gt c) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) (v : Fin (D.gat u w hg).P.nV) : (D.gat u w hg).P.n' v = frobOrder D (D.what₂.places.resChar w)

theorem **coefficient_ring** {D : D7} {Gt : GatTies D} {c : LatCmp D} (h : IsConcrete D Gt c) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : (D.gat u w hg).O = ORec D w (frobOrder D (D.what₂.places.resChar w))

theorem **exists_embedding_of_ringTie** {D : D7} {u : D.what₂.Word} {w : D.what₂.places.Pl} {hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)} {v : Fin (D.gat u w hg).P.nV} (h : RingTie D u w hg v) : ∃ e : D.what₂.LL →+* (D.gat u w hg).Kf, ∀ x, (0 : WithTop ℤ) ≤ D.what₂.places.val w x ↔ e x ∈ Set.range (algebraMap (D.gat u w hg).O (D.gat u w hg).Kf)

theorem **localIs_record_free** (D : D7) (sc : Scale) (Ce : ℝ) (Lc : ∀ u, ColLevel D.R D.host.F D.host.E D.sp D.host.g (D.mem u).C (D.mem u).A (D.clsLat u)) (I) (ψ) (hd) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : LocalIs (record D sc Ce Lc I ψ hd) Lc u w hg

theorem **exists_hermTies** (D : D7) (sc : Scale) (Ce : ℝ) (Lc : ∀ u, ColLevel D.R D.host.F D.host.E D.sp D.host.g (D.mem u).C (D.mem u).A (D.clsLat u)) (I) (ψ) (hd) : ∃ H0 : Formalism.L2.Herm.Herm0 (record D sc Ce Lc I ψ hd), ∀ u, H0.lev0 u = Lc u

theorem **forms_and_basis** {D : D7} {Gt : GatTies D} {c : LatCmp D} (h : IsConcrete D Gt c) : (∀ u, Nonempty (Forms0 D u)) ∧ ∃ B, Cmp.StdB D B

theorem **exponent_eq** {D : D7} {Gt : GatTies D} {c : LatCmp D} (h : IsConcrete D Gt c) : ∃ sc : Scale, ∀ u w hg, (D.gat u w hg).e = epSc D.R sc 1 (D.what₂.places.resChar w) ∧ (D.gat u w hg).P.r = rSc sc D.R.k (D.what₂.places.resChar w)

theorem **block_rational** : Formalism.L2.G.BlockRationalityField

theorem **comparison_statements** (hJ : ∀ u, Joint0 H0.lev0 u (T u)) (h0 : ∀ u, LevelZeroCharacterRational (D

theorem **exists_isConcrete** (hJ : ∀ u, Joint0 H0.lev0 u (T u)) (h0 : ∀ u, LevelZeroCharacterRational (D

theorem **isConcrete_record_iff** (D : D7) (sc : Scale) (Ce : ℝ) (Lc : ∀ u, ColLevel D.R D.host.F D.host.E D.sp D.host.g (D.mem u).C (D.mem u).A (D.clsLat u)) (I) (ψ) (hd) (hd') : (∃ c, IsConcrete (record D sc Ce Lc I ψ hd') (recordTies D sc Ce Lc I ψ hd') c) ↔ ∃ c, IsConcrete (recC D sc Ce Lc I ψ hd) (gatTiesC D sc Ce Lc I ψ hd) c

theorem **baseLattice_standard** {D : D7} {Gt : GatTies D} {c : LatCmp D} (G : AllTies D Gt c) : ∃ B : GL (Fin D.R.k) D.host.E, Cmp.StdB D B ∧ ∀ u (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) (w' : HeightOneSpectrum (𝒪 D.host.F)) (h : ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal) (t : Fin D.R.k → w'.adicCompletion D.host.F), (∃ x ∈ Formalism.L2.Herm.locL w' (G.herm.lev0 u).Lb, ∀ a, ((G.herm.g u w hg).spl w' h (x a)).1 = t a) ↔ ∀ a, (((bc (((RingHom.fst _ _).comp ((G.herm.g u w hg).spl w' h).toRingHom).comp (inclE (F

theorem **isConcrete_iff** (D : D7) (Gt : GatTies D) (c : LatCmp D) : IsConcrete D Gt c ↔ IsRecC D Gt c

theorem **exists_concrete_of_record** (D : D7) (L0 : LevelData D) (T : ∀ u : D.what₂.Word, Forms0 D u) (hS : Formalism.L2.Herm.StdTied D) (hJ : ∀ u, Joint0 L0 u (T u)) (h0 : ∀ u, LevelZeroCharacterRational T u) (hb : ∀ u, ∃ a : Ideal (𝒪 D.host.E), (L0 u).Lb ⊔ a • Submodule.span (𝒪 D.host.E) (dualSet (D.sp.cm.sig : D.host.E →+* D.host.E) D.sp.gram0 (L0 u).Lb) = D.sp.crdLat0 (D.sp.X 0).L0) (sc : Scale) : ∃ I ψ hd c, IsConcrete (recC D sc 1 L0 I ψ hd) (gatTiesC D sc 1 L0 I ψ hd) c

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsExtendResidue

theorem **allSteps_injective** {B : Mdl Y i₀} : Function.Injective (allSteps : Lat B → Finset (Step Y i₀))

theorem **FlagFam.lat_full** {Z : Mdl Y i₀} (L : Lat Z) (h : L.IsTop ∨ P Z L) : (FlagFam.full L h).lat = L

theorem **Mdl.isLocalHom_inclV** (Z : Mdl Y i₀) : IsLocalHom Z.inclV

theorem **exists_extend_residue** {R R' k : Type} [CommRing R] [IsLocalRing R] [CommRing R'] [IsLocalRing R'] [Finite (ResidueField R')] [Field k] [IsAlgClosed k] (f : R →+* R') [IsLocalHom f] (ψ : ResidueField R →+* k) : ∃ φ : R' →+* k, φ.comp f = ψ.comp (residue R)

theorem **finite_quot_smul_top** {R M : Type} [CommRing R] [AddCommGroup M] [Module R M] [Module.Finite ℤ M] {n : ℕ} (hn : n ≠ 0) : Finite (M ⧸ (Ideal.span {(n : R)} • ⊤ : Submodule R M))

theorem **FlP.sub_le** {ct : At S κ p} (F : ct.FlP) (τ : PsiT D) (i : ℕ) : F.sub τ i ≤ ct.eigBar τ

theorem **FlP.ext** {ct : At S κ p} {F F' : ct.FlP} (h : ∀ τ i, F.sub τ i = F'.sub τ i) : F = F'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsIsAdelicChain

theorem **mult_anti** {V : HSp D} {S : D.what₂.Tau → SPar} (θ : InfFam D) (M : DSModel V S) {J J' : Set (GA D V)} {χ χ' : GA D V → ℂ} (h : ∀ k ∈ J', k ∈ J ∧ χ k = χ' k) : Mult θ M J χ ≤ Mult θ M J' χ'

theorem **principal_le_level** {V : HSp D} (β : GL (Fin V.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ V.rank) (P : LevelP L.Pg L.r ℂ) (k : GA D V) (hk : k ∈ levS V β (flag1L L)) : k ∈ levS V β (withP L P) ∧ levChi V β (withP L P) k = levChi V β (flag1L L) k

theorem **toMM_apply** (t : T) (j : ℕ) (U : R.CC t j) (v : (R.MY j).Vec) : toMM R t j U v = (U : SmT (R.MY j) →ₗ[ℂ] (GA D (Yc D j) → ℂ)) (PiTensorProduct.tprod ℂ v)

theorem **toMM_injective** (t : T) (j : ℕ) : Function.Injective (toMM R t j)

theorem **range_toMM** (t : T) (j : ℕ) : LinearMap.range (toMM R t j) = multAt R R.triv t j

theorem **multAt_le** (P : LevelP R.Pg D.R.r ℂ) (t : T) (j : ℕ) : multAt R P t j ≤ multAt R R.triv t j

theorem **map_lvlPart** (P : LevelP R.Pg D.R.r ℂ) (t : T) (j : ℕ) : (lvlPart R P t j).map (toMM R t j) = multAt R P t j

theorem **multAt_zero** (P : LevelP R.Pg D.R.r ℂ) (t : T) : multAt R P t 0 = Mult R.θ (R.MY 0) (levS _ (R.βY 0) (R.dat0 P)) (levChi _ (R.βY 0) (R.dat0 P))

theorem **EEat_le** (P : LevelP R.Pg D.R.r ℂ) (t : T) : R.EEat P ≤ multAt R P t 0

theorem **EEat_le_triv** (P : LevelP R.Pg D.R.r ℂ) : R.EEat P ≤ R.EEat R.triv

theorem **toMM_dnFrom_topLift** (t : T) (y : (R.HX t).H) : toMM R t 0 (Lift.dnFrom (R.CC t) (R.ioY t) (D.R.m − 1) (topLift R t y)) = PiTensorProduct.lift.symm (R.thDn t y)

theorem **pred_m_add_one** (D : D7) : D.R.m − 1 + 1 = D.R.m

theorem **downTo_multCast** (μ : D.what₂.Tup) {i j : ℕ} (h : i = j) (x : S.Mult μ i) : S.downTo μ j (multCast S μ h x) = S.downTo μ i x

theorem **iota_topEquiv** (μ : D.what₂.Tup) (x : S.Mult μ (D.R.m − 1 + 1)) : S.iota μ (topEquiv S μ x) = S.multZero μ (S.downTo μ (D.R.m − 1) (S.io μ (D.R.m − 1) x))

theorem **toMM_downTo** (R : RealDat D A Ks) (μ : D.what₂.Tup) (t : T) (e : ∀ j, S.Mult μ j →ₗ[ℂ] R.CC t j) (h0 : ∀ x, toMM R t 0 (e 0 (S.io μ 0 x)) = botLift R t (e 1 x)) (h1 : ∀ j, 0 < j → j + 1 < D.R.m → ∀ x, e j (S.io μ j x) = R.ioY t j (e (j + 1) x)) (n : ℕ) (hn : 0 < n) (hm : n < D.R.m) (x : S.Mult μ n) : toMM R t 0 (e 0 (S.downTo μ n x)) = postOn (R.MY 0) R.e0 (toMM R t 0 (Lift.dnFrom (R.CC t) (R.ioY t) n (e n x)))

theorem **toMM_iota** (R : RealDat D A Ks) (μ : D.what₂.Tup) (t : T) (e : ∀ j, S.Mult μ j →ₗ[ℂ] R.CC t j) (eA : S.Amb μ →ₗ[ℂ] (R.HX t).H) (h0 : ∀ x, toMM R t 0 (e 0 (S.io μ 0 x)) = botLift R t (e 1 x)) (h1 : ∀ j, 0 < j → j + 1 < D.R.m → ∀ x, e j (S.io μ j x) = R.ioY t j (e (j + 1) x)) (h2 : ∀ x, e (D.R.m − 1) (S.io μ (D.R.m − 1) x) = topLift R t (eA (topEquiv S μ x))) (a : S.Amb μ) : toMM R t 0 (e 0 ((S.multZero μ).symm (S.iota μ a))) = postOn (R.MY 0) R.e0 (PiTensorProduct.lift.symm (R.thDn t (eA a)))

theorem **exists_factor** {X Y Z : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup Y] [Module ℂ Y] [AddCommGroup Z] [Module ℂ Z] (g : Y →ₗ[ℂ] Z) (hg : Function.Injective g) (f : X →ₗ[ℂ] Z) (hf : ∀ x, f x ∈ LinearMap.range g) : ∃ h : X →ₗ[ℂ] Y, ∀ x, g (h x) = f x

theorem **eq_mul_of_eq_inv_mul** {n : ℝ} (hn : 0 < n) {a c : ℂ} (h : a = ((n⁻¹ : ℝ) : ℂ) * c) : c = (n : ℂ) * a

theorem **haar_chain** (hY : R.models_νY) (hY' : R.models_νY') (t : T) (j : ℕ) (hj : j < D.R.m) : (R.νY t j).IsHaarMeasure ∧ IsFundamentalDomain (GQ (Yc D j)) (R.fdY t j) (R.νY t j)

theorem **finiteDimensional_CC** (hF : GlobalTheory.Published.HarishChandraFinite) (hi : R.models_inf) (t : T) (j : ℕ) : FiniteDimensional ℂ (R.CC t j)

theorem **exists_lvlModel** (hF : GlobalTheory.Published.HarishChandraFinite) (hD : GlobalTheory.Published.CuspRapidDecay) (hi : R.models_inf) (hY : R.models_νY) (hY' : R.models_νY') (P : LevelP R.Pg D.R.r ℂ) (c : ℝ) (hc : 0 < c) (t : T) (j : ℕ) (hj : j < D.R.m) : ∃ H : HMod (R.CC t j), H.Is (lvlPart R P t j) fun U U' => (c : ℂ) * ipCC R t j U U'

theorem **downTo_withChain** (S : RowSide D EE) (c : ∀ μ, HChain D.R.m EE (S.Amb μ)) (μ : D.what₂.Tup) (j : ℕ) : (withChain S c).downTo μ j = (c μ).down j

theorem **down_chainOf** (S : RowSide D EE) (μ : D.what₂.Tup) (j : ℕ) : (chainOf S μ).down j = S.downTo μ j

theorem **iota_withChain** (S : RowSide D EE) (c : ∀ μ, HChain D.R.m EE (S.Amb μ)) (μ : D.what₂.Tup) (h : c μ = chainOf S μ) : (withChain S c).iota μ = S.iota μ

theorem **e_eE_mem** (Y : SideTie R F tup b S) (x : EE) : R.HE.e (Y.eE x) ∈ R.EEat (R.levR F b)

theorem **exists_isAdelicChain** (Y : SideTie R F tup b S) (hE : R.models_HE) (hK : KeepsLevel R F b) (t : T) (hH : ∀ j, j < D.R.m → ∃ H : HMod (R.CC t j), H.Is (lvlPart R (R.levR F b) t j) fun U U' => (((RealDat.nP F b)⁻¹ : ℝ) : ℂ) * ipCC R t j U U') : ∃ c, IsAdelicChain Y t c

theorem **topEquiv_withChain** (S : RowSide D EE) (c : ∀ μ, HChain D.R.m EE (S.Amb μ)) (μ : D.what₂.Tup) (x : (c μ).Sp (D.R.m − 1 + 1)) : topEquiv (withChain S c) μ x = (c μ).top (famCast (c μ).Sp (pred_m_add_one D) x)

theorem **iota_tie_withChain** (Y : SideTie R F tup b S) (hE : R.models_HE) (hT : R.chain_Th) (c : ∀ μ, HChain D.R.m EE (S.Amb μ)) (t : T) (hc : IsAdelicChain Y t (c (tup t))) (x : S.Amb (tup t)) : Y.eE ((withChain S c).iota (tup t) x) = R.Th t (Y.eA t x)

theorem **exists_multTie** (hF : GlobalTheory.Published.HarishChandraFinite) (hD : GlobalTheory.Published.CuspRapidDecay) (hM : R.IsModels) (hT : R.chain_Th) (hK : KeepsLevel R F b) (hinj : Function.Injective tup) (Y : SideTie R F tup b S) : ∃ (c : ∀ μ, HChain D.R.m EE (S.Amb μ)) (Y' : SideTie R F tup b (withChain S c)), Y'.eE = Y.eE ∧ Y'.eA = Y.eA ∧ (∀ μ, (withChain S c).iota μ = S.iota μ) ∧ Nonempty (MultTie Y')

theorem **isMultiplicityInner_mul_inner** {V : HSp D} {Sp : D.what₂.Tau → SPar} {M : DSModel V Sp} {μ : Measure (GA D V)} {𝓕 : Set (GA D V)} {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] (e : H →ₗ[ℂ] MMap M) {d : ℂ} (hd : d ≠ 0) (h : ∀ x y, ipM M μ 𝓕 (e x) (e y) = d * ⟪x, y⟫_ℂ) (x y : H) : IsMultIP M μ 𝓕 (e x) (e y) (d * ⟪x, y⟫_ℂ)

theorem **range_row** (t : T) {j : ℕ} (h0 : 0 < j) (hj : j < D.R.m) : LinearMap.range (M.row t j) = multAt R (R.levR F b) t j

theorem **ioY_mem** (t : T) {j : ℕ} (h0 : 0 < j) (hj : j + 1 < D.R.m) {U : R.CC t (j + 1)} (hU : U ∈ lvlPart R (R.levR F b) t (j + 1)) : R.ioY t j U ∈ lvlPart R (R.levR F b) t j

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsIsBIffCoreB

theorem **D_setDivZ** (w : ColW B f T) (Z : w.Lt) (N : ∀ u, Submodule (𝒪 B.LL) (Z.V.div.Nu u).V) (h : B.Word → ℝ) : w.D (setDivZ Z N h) = setDiv (w.D Z) N h

theorem **CoreB.isB** {K : Ctx} {w : ColW B f T} {Z : w.Lt} (b : CoreB K w Z) : IsB K (w.D Z) b.obj

theorem **IsB.ind** {K : Ctx} {R : ∀ D : D7, Obj2 D → Prop} (h : ∀ (B : Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt) (b : CoreB K w Z), R (w.D Z) b.obj) {D : D7} {O : Obj2 D} (hD : IsB K D O) : R D O

theorem **IsB.ind_obj** {K : Ctx} {R : ∀ D : D7, Obj D → Prop} (h : ∀ (B : Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt) (b : CoreB K w Z) (O : Obj (w.D Z)), ofObj O = b.obj → R (w.D Z) O) {D : D7} {O : Obj D} (hD : IsB K D (ofObj O)) : R D O

theorem **IsB.ind_lay** {K : Ctx} {R : ∀ D : D7, Lay D → Prop} (h : ∀ (B : Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt) (b : CoreB K w Z) (Y : Lay (w.D Z)), layObj Y = b.obj → R (w.D Z) Y) {D : D7} {Y : Lay D} (hD : IsB K D (layObj Y)) : R D Y

theorem **exists_isB_iff_coreB** (K : Ctx) : (∃ (D : D7) (O : Obj2 D), IsB K D O) ↔ ∃ (B : Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt), Nonempty (CoreB K w Z)

theorem **CoreB.pinU** {K : Ctx} {w : ColW B f T} {Z : w.Lt} (b : CoreB K w Z) : PinU K B f T w

theorem **CoreB.muAt_eq** {K : Ctx} {w : ColW B f T} {Z : w.Lt} (b : CoreB K w Z) (v : (w.D Z).what₂.places.Pl) : (w.D Z).what₂.muAt v = (w.D Z).what₂.mu0

theorem **CoreB.over** {K : Ctx} {w : ColW B f T} {Z : w.Lt} (b : CoreB K w Z) : Over (w.D Z) K.P K.Lv

theorem **CoreB.volTied** {K : Ctx} {w : ColW B f T} {Z : w.Lt} (b : CoreB K w Z) : Formalism.L2.Herm.VolumeCompatible (w.D Z)

theorem **ForVol.mono** {w : ColW B f T} {P Q : w.Lt → Prop} (h : ForVol w P) (hPQ : ∀ Z, P Z → Q Z) : ForVol w Q

theorem **HasB.forVol_isB** {K : Ctx} {w : ColW B f T} (h : HasB K B f T w) : ForVol w fun Z => ∃ O, IsB K (w.D Z) O

theorem **CoreB.layObj_lay** (b : CoreB K w Z) (hv : StandardLattice.VolumeEqMass₂ (w.D Z)) : layObj (b.lay hv) = b.obj

theorem **CoreB.isB_lay** (b : CoreB K w Z) (hv : StandardLattice.VolumeEqMass₂ (w.D Z)) : IsB K (w.D Z) (layObj (b.lay hv))

theorem **CoreB.tupUsed** (b : CoreB K w Z) : DQ.TupUsedAt (w.D Z)

theorem **ev_recOK** : Ev fun K => ∀ (B : Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt), CoreB K w Z → RecOK K.s K.C K.cD (w.D Z)

theorem **CoreB.led** (b : CoreB K w Z) (L : SizeData₂ (w.D Z)) (hs : L.sc = K.s) (hW : WtEq L K.a K.b) (hV : Val K.C K.cD K.ch L) (X : G.Inst (w.D Z) L) (hX : CanonInst X) (he : G.EpDef X K.C.cX.Ce) (hk : G.KapIDef X K.C.GK) (hf : ∀ v, G.ParametersFill X K.C.GK v) : G.LatTied b.obj.gatT X b.obj.Gy ∧ EpTie X ∧ ∀ u (v : (w.D Z).what₂.places.Pl) (hg : GIdx (w.D Z).host ((w.D Z).mem u).A.T ((w.D Z).what₂.places.resChar v)), (∀ x, b.obj.cmp.cmp u v hg ((w.D Z).nuIn u x) ∈ G.NP X b.obj.Gy K.C.GK u v hg → InLoc ((w.D Z).plEq v) (DV.MramL (w.D Z) u) x) ∧ ∃ n : ℕ, n ≠ 0 ∧ ∀ x ∈ DV.MramL (w.D Z) u, b.obj.cmp.cmp u v hg ((w.D Z).nuIn u (n • x)) ∈ G.NP X b.obj.Gy K.C.GK u v hg

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsIsHodgeFrameIff

theorem **cutoff_mem** (y : ℝ) : cutoff y ∈ Set.Icc (0 : ℝ) 1

theorem **cutoff_of_le** {y : ℝ} (h : y ≤ 2) : cutoff y = 1

theorem **cutoff_of_ge** {y : ℝ} (h : 3 ≤ y) : cutoff y = 0

theorem **contDiff_cutoff** : ContDiff ℝ ∞ cutoff ∧ ContDiff ℝ ∞ fun y => √(cutoff y)

theorem **cutWt_mem** (κ : ℝ) {R r : ℝ} (hR : 0 ≤ R) (hr : 0 ≤ r) : cutWt κ R r ∈ Set.Icc (0 : ℝ) 1

theorem **cutWt_of_ge** (κ : ℝ) {R r : ℝ} (hR : 2 ≤ R) (h : R ^ 3 ≤ r) : cutWt κ R r = 0

theorem **cutWt_of_le** (κ : ℝ) {R r : ℝ} (hR : 2 ≤ R) (hr : 0 < r) (h : r ≤ R ^ 2) : cutWt κ R r = 1 / (1 + (r / R) ^ κ)

theorem **OddPrimesWithRootsOfUnity.eigMet_gram** : (x.eigMet σ).gram = x.gram

theorem **OddPrimesWithRootsOfUnity.smoothOn_eigMet** : (x.eigMet σ).SmoothOn (manifold Z σ) Set.univ

theorem **OddPrimesWithRootsOfUnity.exp_le_r** (y : CxPts (genStr Z) σ) : Real.exp 1 ≤ x.r σ y

theorem **xiR_pos** (D : D7) : 0 < xiR D

theorem **FinalTheorems.im_neg_xi** (σ : κ.c.F →+* ℂ) (k : (s.U σ).Comp) (τ : ((s.U σ).X k).Ψ) : (τ.1 (-((s.U σ).X k).ξ0)).im = xiR D

theorem **_root_.GlobalTheory.ShFam.FinalTheorems.exists_isHodgeFrame_iff** : (∃ f, IsHodgeFrame s.U y f) ↔ ¬ LiesIn y (genBd s.T.Z)

theorem **hodge_gram** {f : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)} (hf : IsHodgeFrame s.U y f) (g : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)) (τ : D.host.E →+* ℂ) : (x.hodge σ).gram y g τ = ((2 * Real.pi * xiR D : ℝ) : ℂ) • ((f.co g τ)ᴴ * f.co g τ)

theorem **hodge_gram_of_liesIn** (h : LiesIn y (genBd s.T.Z)) (g : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)) (τ : D.host.E →+* ℂ) : (x.hodge σ).gram y g τ = x.gram y g τ

theorem **hodge_gram_riemann** (hσ : Formalism.L2.PjFree.GoodEmb κ.c σ) (W : UnfWit (s.U σ)) {i : Fin W.F.n} {P : (W.F.Di i).La →+ (Fin W.g → ℂ)} (hP : IsPeriod (W.F.Di i) (fun j => Formalism.L2.PjFree.relab κ.c σ (W.e j).1) P) {Hm : Matrix (Fin W.g) (Fin W.g) ℂ} (hH : IsRiem W.F i P Hm) {q : FrPt s.Xo κ.c (CAt σ)} {z : ∀ τ : D.host.E →+* ℂ, Fin (S.Dw.mult τ) → (Fin W.g → ℂ)} (hq : AnLie W.F i P q z) (τ : D.host.E →+* ℂ) : (x.hodge σ).gram (cxPt s.T.Z q) (cxFrame s.T.Z q) τ = (Real.pi : ℂ) • of fun b b' => hermForm Hm (z τ b') (z τ b)

theorem **hodge_gram_lieFrame** (L : CmList A) (k : s.Comp) {f : (s.fd k).dom} (hf : IsOrth (s.fd k) f.1) (t : D.what₂.Tau) (τ : D.host.E →+* ℂ) : (x.hodge κ.σ₀).gram (s.framePt k f) (s.lieFrame k f) τ = (hodgeC (s.fd k) (s.lab L k) t : ℂ) • 1

theorem **isMuExp_muExp** (h : ∃ C, x.IsMuExp C) : x.IsMuExp x.muExp

theorem **lt_kap** : 2 * x.muExp + 2 < x.kap

theorem **wt_mem** (R : ℝ) (y : CxPts (genStr s.T.Z) σ) : x.wt σ R y ∈ Set.Icc (0 : ℝ) 1

theorem **wt_eq** {R : ℝ} (hR : 2 ≤ R) (h : ¬ LiesIn y (genBd s.T.Z)) : x.wt σ R y = cutWt x.kap R (x.r σ y)

theorem **wt_eq_zero** {R : ℝ} (hR : 2 ≤ R) (h : LiesIn y (genBd s.T.Z) ∨ R ^ 3 ≤ x.r σ y) : x.wt σ R y = 0

theorem **posSet_metsOf** : Met.posSet (metsOf x Φ k R σ n) = Set.univ

theorem **posSet_refMet** : Met.posSet (refMet x Φ k σ n) = Set.univ

theorem **pn_metsOf_add** : (cxSt D.what₂.LL).pn _ σ _ (metsOf x Φ k R σ (n + n')) = (cxSt D.what₂.LL).pn _ σ _ ((cxSt D.what₂.LL).mAdd _ σ _ _ (metsOf x Φ k R σ n) (metsOf x Φ k R σ n'))

theorem **pn_refMet_add** : (cxSt D.what₂.LL).pn _ σ _ (refMet x Φ k σ (n + n')) = (cxSt D.what₂.LL).pn _ σ _ ((cxSt D.what₂.LL).mAdd _ σ _ _ (refMet x Φ k σ n) (refMet x Φ k σ n'))

theorem **pn_metsOf** {g : (Φ.C k).FrameAt p} {f : (Φ.C k).Y.Fn} {a : ℂ} (h : IsCoord (Φ.C k) p.1 g n f a) : (cxSt D.what₂.LL).pn _ σ _ (metsOf x Φ k R σ n) ((cxSt D.what₂.LL).cx _ σ f) p = ‖a‖ ^ 2 * ∏ z : Letter S.Dw, grSq ((x.eigMetAt _ R).gram _ g z.1) z.2 ^ n z

theorem **pn_metsOf_none** {g : (Φ.C k).FrameAt p} (hg : IsHodgeFrame s.U _ g) {f : (Φ.C k).Y.Fn} {a : ℂ} (h : IsCoord (Φ.C k) p.1 g n f a) : (cxSt D.what₂.LL).pn _ σ _ (metsOf x Φ k none σ n) ((cxSt D.what₂.LL).cx _ σ f) p = ‖a‖ ^ 2 * ∏ z : Letter S.Dw, (2 * Real.pi * xiR D) ^ n z

theorem **metsOf_some_eq** {R : ℝ} (hR : 2 ≤ R) (h : (Φ.C k).π (base p) ∈ genBd s.T.Z ∨ R ^ 3 ≤ rFlag x Φ k σ p) (a : (Φ.C k).Y.Fnˣ) : Met.h (metsOf x Φ k (some R) σ n) a p = Met.h (refMet x Φ k σ n) a p

theorem **exp_le_rFlag** : Real.exp 1 ≤ rFlag x Φ k σ p

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsIvDOfDefinite

theorem **toFun_pull_eq** {n m : ℕ} {r : (Fin n → O) →ₗ[O] X} {s : (Fin m → O) →ₗ[O] X} (hr : Function.Surjective r) (hs : Function.Surjective s) (B : X → X → ℝ) : φ.toFun n (fun a b => B (r a) (r b)) = φ.toFun m (fun a b => B (s a) (s b))

theorem **eval_eq** [Module.Finite O X] {n : ℕ} {r : (Fin n → O) →ₗ[O] X} (hr : Function.Surjective r) (B : X → X → ℝ) : φ.eval X B = φ.toFun n fun a b => B (r a) (r b)

theorem **eval_pi** (n : ℕ) (B : (Fin n → O) → (Fin n → O) → ℝ) : φ.eval (Fin n → O) B = φ.toFun n B

theorem **eval_comp** [Module.Finite O X] [Module.Finite O X₁] {r : X →ₗ[O] X₁} (hr : Function.Surjective r) (B : X₁ → X₁ → ℝ) : φ.eval X (fun a b => B (r a) (r b)) = φ.eval X₁ B

theorem **eval_equiv** [Module.Finite O X] [Module.Finite O X₁] (e : X ≃ₗ[O] X₁) (B : X₁ → X₁ → ℝ) : φ.eval X (fun a b => B (e a) (e b)) = φ.eval X₁ B

theorem **eval_of_subsingleton** [Module.Finite O X] [Subsingleton X] (B : X → X → ℝ) : φ.eval X B = 0

theorem **minVal_three** : (minVal ℤ).eval ℤ (fun a b => 3 * a * b) = 3

theorem **LinearOn.eval_eq_sum** [Module.Finite O X] [Module.Finite O X₁] {m : ℕ} {γ : Fin m → LinearMap.BilinForm ℤ X} (h : φ.LinearOn γ) {r : X →ₗ[O] X₁} (hr : Function.Surjective r) {B : X₁ → X₁ → ℝ} {c : Fin m → ℝ} (hc : ∀ i, 0 ≤ c i) (hB : ∀ a b, B (r a) (r b) = ∑ i, c i * (γ i a b : ℝ)) : φ.eval X₁ B = ∑ i, c i * φ.evalZ (γ i)

theorem **IsPiece.exists_eq_mul** {N M : ℕ} (hM : M ≠ 0) {γ : Fin m → levelForms H M X} {ℓ : LinearMap.BilinForm ℤ X →ₗ[ℤ] ℤ} (h : IsPiece H φ N (fun i => γ i) ℓ) (i : Fin m) : ∃ z : ℤ, N * φ.evalZ (γ i : LinearMap.BilinForm ℤ X) = M * z

theorem **ordGe_zero** (h : R) (n : ℕ) (I₁ I₂ : Ideal R) : OrdGe h 0 n I₁ I₂

theorem **OrdGe.map** {S : Type} [CommRing S] (f : R →+* S) {h : R} {i : ℚ} {n : ℕ} {I₁ I₂ : Ideal R} (H : OrdGe h i n I₁ I₂) : OrdGe (f h) i n (I₁.map f) (I₂.map f)

theorem **ordGe_span_iff** {h g₁ g₂ : R} {i : ℚ} {n : ℕ} : OrdGe h i n (Ideal.span {g₁}) (Ideal.span {g₂}) ↔ g₁ ^ i.num.toNat * g₂ ^ (-i.num).toNat ∣ h ^ (i.den * n) * (g₂ ^ i.num.toNat * g₁ ^ (-i.num).toNat)

theorem **Mdl.isOpen_opn** (Z : Mdl Y i₀) : IsOpen Z.opn

theorem **stalkId_top** {X : Scheme.{0}} (x : X) : FinalTheorems.stalkId (⊤ : X.IdealSheafData) x = ⊤

theorem **idlAlong_top** {X : Scheme.{0}} {R : Type} [CommRing R] [IsLocalRing R] (ξ : Sp R ⟶ X) : idlAlong ⊤ ξ = ⊤

theorem **FanFun.divGe_zero_iff** {i₁ : Idx H p} (φ : FanFun Y i₀ i₁) (Z : Mdl Y i₀) (F : Z.X.Modules) (et : Γ(F, Z.gen0)) : φ.DivGe Z F et 0 ↔ ∃ (M : Scheme.{0}) (f : M ⟶ Z.gen0.toScheme) (g : M ⟶ φ.Z.X), IsCommonMod Z φ.Z f g

theorem **π_mem_iff** (x : d.Z.X) : d.π x ∈ Set.range (Y i).j ↔ x ∈ d.Z.opn

@[reassoc] theorem **ρ_π** : d.ρ ≫ Z₀.π = d.Z.gen0.ι ≫ d.π

theorem **ρ_mem_opn_iff** (x : d.Z.gen0.toScheme) : d.ρ x ∈ Z₀.opn ↔ d.Z.gen0.ι x ∈ d.Z.opn

theorem **le_jIdl_iff** (I : Z₀.X.IdealSheafData) : I ≤ d.jIdl.1 ↔ I.comap d.ρ ≤ d.idealQ

theorem **jIdl_eq_map_comap** (ρQ : d.Z.gen0.toScheme ⟶ Z₀.gen0.toScheme) (h : ρQ ≫ Z₀.gen0.ι = d.ρ) : d.jIdl.1 = (d.idealQ.map ρQ).map Z₀.gen0.ι

theorem **PolData.jIdl_ofNoBoundary** (Z : Mdl Y i₀) [IsIso Z.π] (h : ∀ σ, Z.ty σ = ∅) (hb : ∀ x ∈ Z.gen0, x ∈ Z.opn) (φ : InvFun (𝒪 E)) : (PolData.ofNoBoundary Z h hb φ).jIdl.1 = ⊤

theorem **Mdl.mem_range_of_mem_gen0** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) {x : Z.X} (hx : x ∈ Z.gen0) : Z.π x ∈ Set.range (genι (Y Z.i))

theorem **Mdl.mem_opn_of_dim_eq_zero** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (h : Dw.dim = 0) (Z : Mdl Y i₀) : ∀ x ∈ Z.gen0, x ∈ Z.opn

theorem **At.phiAt_eq_sum** (ct : At S κ p) (φ : ct.PL) {Z : ct.Mdl} (y : ct.Pt Z) {X : Type} [AddCommGroup X] [Module (𝒪 D.host.E) X] [Module.Finite (𝒪 D.host.E) X] {m : ℕ} {γ : Fin m → LinearMap.BilinForm ℤ X} (h : φ.φ.LinearOn γ) {r : X →ₗ[𝒪 D.host.E] ct.Ylat y} (hr : Function.Surjective r) {c : Fin m → ℝ} (hc : ∀ i, 0 ≤ c i) (hB : ∀ a b, (ct.dat y).beta (r a) (r b) = ∑ i, c i * (γ i a b : ℝ)) : ct.phiAt φ y = ∑ i, c i * φ.φ.evalZ (γ i)

theorem **At.phiAt_congr** (ct : At S κ p) (φ : ct.PL) {Z Z' : ct.Mdl} (y : ct.Pt Z) (y' : ct.Pt Z') (e : ct.Ylat y ≃ₗ[𝒪 D.host.E] ct.Ylat y') (h : ∀ a b, (ct.dat y').beta (e a) (e b) = (ct.dat y).beta a b) : ct.phiAt φ y = ct.phiAt φ y'

theorem **At.exists_IvD_of_definite** (ct : At S κ p) (h : S.Dw.p = 0 ∨ S.Dw.q = 0) (b : Option ℕ) : ∃ d : ct.IvD b, (ct.jIdl b d).1 = ⊤

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsLabOfOrder

private theorem **exists_rep_tameEig_eq** {h : IsPlaceOver A w} (x : WD h) : ∃ (n : ℕ) (b : WDRep h (Fin n → ℂ)), ∀ δ : G E Ω, WD.tameEig x δ = WD.tameEig (WD.mk b) δ

theorem **lab_mul_wild** {h : IsPlaceOver A w} {τ : Fbar g →* ℂ} {γ u : G E Ω} (H : TeichmullerInertiaLabelCondition A w τ γ) (hu : u ∈ Wild E A) (x : WD h) : WD.lab τ (γ * u) x = WD.lab τ γ x

private theorem **tameEig_eq_tameEig_mk_of_rep** {h : IsPlaceOver A w} (x : WD h) : ∃ (n : ℕ) (b : WDRep h (Fin n → ℂ)), ∀ δ : G E Ω, WD.tameEig x δ = WD.tameEig (WD.mk b) δ

private theorem **roots_charpoly_pow_eq_map_pow'** {V : Type*} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (f : Module.End ℂ V) (r : ℕ) : (f ^ r).charpoly.roots = f.charpoly.roots.map (· ^ r)

theorem **lab_pow** {h : IsPlaceOver A w} {τ : Fbar g →* ℂ} {γ : G E Ω} (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : WD.lab τ (γ ^ g) x = WD.lab τ γ x

private theorem **exists_wild_sqrtChar_ne_one_of_normV_two** (h : IsPlaceOver A w) (hw : normV w = 2) {ϖ : 𝒪 E} (h1 : ϖ ∈ w.asIdeal) (h2 : ϖ ∉ w.asIdeal ^ 2) {x : Ω} (hx : x ^ 2 = algebraMap E Ω (ϖ : E)) : ∃ σ : Weil h, (σ : G E Ω) ∈ Wild E A ∧ sqrtChar h hx σ ≠ 1

theorem **lab_sqrtChar_two** {h : IsPlaceOver A w} {τ : Fbar 2 →* ℂ} {γ : G E Ω} (H : TeichmullerInertiaLabelCondition A w τ γ) {ϖ : 𝒪 E} (h1 : ϖ ∈ w.asIdeal) (h2 : ϖ ∉ w.asIdeal ^ 2) {x : Ω} (hx : x ^ 2 = algebraMap E Ω (ϖ : E)) : ¬ WD.tame (WD.lpOfChar h _ (isSmooth_sqrtChar h hx)) ∧ WD.lab τ γ (WD.lpOfChar h _ (isSmooth_sqrtChar h hx)) = 1

theorem **eq_one_of_cong** (h : IsPlaceOver A w) (hw : normV w = g) {l : ℕ} (hc : l.Coprime g) {u : Ω} (hu : u ^ l = 1) (h1 : A.valuation (u - 1) < 1) : u = 1

theorem **weil_fix** (h : IsPlaceOver A w) (hw : normV w = g) {l : ℕ} (hl : l ∣ g - 1) (σ : Weil h) {ζ : Ω} (hζ : ζ ^ l = 1) : (σ : G E Ω) ζ = ζ

theorem **kummer_apply** (h : IsPlaceOver A w) (hw : normV w = g) {l : ℕ} (hl : l ∣ g - 1) (ι : Ω →+* ℂ) {x : Ω} (h0 : x ≠ 0) {a : E} (hx : x ^ l = algebraMap E Ω a) (σ : Weil h) : (kummer h hw hl ι h0 hx σ : ℂ) = ι ((σ : G E Ω) x / x)

theorem **kummer_eq_one_iff** (h : IsPlaceOver A w) (hw : normV w = g) {l : ℕ} (hl : l ∣ g - 1) (ι : Ω →+* ℂ) {x : Ω} (h0 : x ≠ 0) {a : E} (hx : x ^ l = algebraMap E Ω a) (σ : Weil h) : kummer h hw hl ι h0 hx σ = 1 ↔ (σ : G E Ω) x = x

theorem **isSmooth_kummer** (h : IsPlaceOver A w) (hw : normV w = g) {l : ℕ} (hl : l ∣ g - 1) (ι : Ω →+* ℂ) {x : Ω} (h0 : x ≠ 0) {a : E} (hx : x ^ l = algebraMap E Ω a) : IsSmooth (kummer h hw hl ι h0 hx)

theorem **exists_lab_of_order** {h : IsPlaceOver A w} {τ : Fbar g →* ℂ} {γ : G E Ω} (H : TeichmullerInertiaLabelCondition A w τ γ) (hγ : TameGenAt A γ) {l : ℕ} (hp : l.Prime) (hl : l ∣ g - 1) : ∃ (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (hχ' : IsSmooth χ⁻¹) (c : ZMod g), orderOf c = l ∧ WD.lab τ γ (WD.lpOfChar h χ hχ) = X - C c ∧ WD.lab τ γ (WD.lpOfChar h χ⁻¹ hχ') = X - C c⁻¹

theorem **exists_rat_seven** [Fact (Nat.Prime 7)] (K : Type) [Field K] [Algebra ℚ K] [IsAlgClosure ℚ K] : ∃ (v : HeightOneSpectrum (𝒪 ℚ)) (B : ValuationSubring K) (hB : IsPlaceOver B v) (τ : Fbar 7 →* ℂ) (γ : G ℚ K) (χ : Weil hB →* ℂˣ) (hχ : IsSmooth χ) (hχ' : IsSmooth χ⁻¹), TameGenAt B γ ∧ WD.lab τ γ (WD.lpOfChar hB χ hχ) ≠ WD.lab τ γ (WD.lpOfChar hB χ⁻¹ hχ') ∧ WD.lab τ γ (WD.lpSum (WD.lpOfChar hB χ hχ) (WD.lpOfChar hB χ⁻¹ hχ')) = X ^ 2 + X + 1

theorem **exists_rat_two** (K : Type) [Field K] [Algebra ℚ K] [IsAlgClosure ℚ K] : ∃ (v : HeightOneSpectrum (𝒪 ℚ)) (B : ValuationSubring K) (hB : IsPlaceOver B v) (τ : Fbar 2 →* ℂ) (γ : G ℚ K) (L : WD hB), TameGenAt B γ ∧ TeichmullerInertiaLabelCondition B v τ γ ∧ L.lpDim = 1 ∧ ¬ L.tame ∧ WD.lab τ γ L = 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsMuIsRamified

theorem **g_big** (eps : ℝ) (h0 : 0 < eps) (h1 : eps < 1 / 52) : ∃ x0 : ℝ, ∀ (s : Scale) (R : Ranks) (g : ℕ), s.eps = eps → x0 ≤ s.x → RankVal s R → ((R.k : ℝ) * s.b) ^ 8 ≤ g → 2 * R.f * R.k < g - 1

theorem **not_ev_false** (hH : DC.HostExQ) (hLP : LevelPrime) (hSP : SmallPrimeOne) (h2b : StandardLattice.SiegelMassRatioCM) : ¬ Ev fun _ => False

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsNotCongPow

theorem **cong_symm** {x y : Ω} (h : A.valuation (x - y) < 1) : A.valuation (y - x) < 1

theorem **cong_trans** {x y z : Ω} (h : A.valuation (x - y) < 1) (h' : A.valuation (y - z) < 1) : A.valuation (x - z) < 1

theorem **val_pow_le_one** {x : Ω} (hx : A.valuation x ≤ 1) (n : ℕ) : A.valuation (x ^ n) ≤ 1

theorem **cong_pow** {x y : Ω} (hx : A.valuation x ≤ 1) (hy : A.valuation y ≤ 1) (h : A.valuation (x - y) < 1) (n : ℕ) : A.valuation (x ^ n - y ^ n) < 1

theorem **val_apply_le_one** {σ : G E Ω} (hσ : σ ∈ Dec (F

theorem **exists_not_cong_pow** [IsAlgClosed Ω] {m n : ℕ} (hn : 0 < n) (hmn : n < m) : ∃ x : Ω, A.valuation x ≤ 1 ∧ ¬ A.valuation (x ^ m - x ^ n) < 1

theorem **ActsAs.mul** {σ τ : G E Ω} {a b c d : ℕ} (hσ : ActsAs A q σ a b) (hτ : ActsAs A q τ c d) : ActsAs A q (σ * τ) (a + c) (b + d)

theorem **ActsAs.inv** {σ : G E Ω} {a b : ℕ} (hσ : ActsAs A q σ a b) : ActsAs A q σ⁻¹ b a

theorem **actsAs_zero_iff** {σ : G E Ω} : ActsAs A q σ 0 0 ↔ σ ∈ Ine (F

theorem **actsAs_one_iff** {σ : G E Ω} : ActsAs A q σ 1 0 ↔ IsFrob A q σ

theorem **ActsAs.unique** [IsAlgClosed Ω] (hq : 1 < q) {σ : G E Ω} {a b c d : ℕ} (h : ActsAs A q σ a b) (h' : ActsAs A q σ c d) : (a : ℤ) - b = (c : ℤ) - d

theorem **ActsAs.cancel** {φ : G E Ω} (hφ : IsFrob A q φ) {σ : G E Ω} {a b : ℕ} (n : ℕ) (h : ActsAs A q σ (a + n) (b + n)) : ActsAs A q σ a b

theorem **one_lt_normV** (w : HeightOneSpectrum (𝒪 E)) : 1 < normV w

theorem **mem_Weil** {σ : G E Ω} : σ ∈ Weil h ↔ ∃ a b, ActsAs A (normV w) σ a b

theorem **Weil_le_Dec** : Weil h ≤ Dec (F

theorem **Ine_le_Weil** : Ine (F

theorem **mem_Weil_of_isFrob** {σ : G E Ω} (hσ : IsFrob A (normV w) σ) : σ ∈ Weil h

@[simp] theorem **coe_ineIncl** (σ : Ine (F

theorem **deg_eq** {σ : Weil h} {a b : ℕ} (hσ : ActsAs A (normV w) (σ : G E Ω) a b) : deg σ = (a : ℤ) − b

@[simp] theorem **deg_mul** (σ τ : Weil h) : deg (σ * τ) = deg σ + deg τ

@[simp] theorem **deg_one** : deg (1 : Weil h) = 0

@[simp] theorem **deg_inv** (σ : Weil h) : deg σ⁻¹ = −deg σ

@[simp] theorem **degHom_apply** (σ : Weil h) : degHom h σ = Multiplicative.ofAdd (deg σ)

@[simp] theorem **deg_zpow** (σ : Weil h) (n : ℤ) : deg (σ ^ n) = n * deg σ

@[simp] theorem **deg_pow** (σ : Weil h) (n : ℕ) : deg (σ ^ n) = n * deg σ

theorem **actsAs_of_deg** {σ : Weil h} {a b : ℕ} (hσ : deg σ = (a : ℤ) − b) : ActsAs A (normV w) (σ : G E Ω) a b

theorem **deg_eq_zero_iff** {σ : Weil h} : deg σ = 0 ↔ (σ : G E Ω) ∈ Ine (F

theorem **deg_eq_one_iff** {σ : Weil h} : deg σ = 1 ↔ IsFrob A (normV w) (σ : G E Ω)

@[simp] theorem **deg_ineIncl** (σ : Ine (F

theorem **isFrob_frob** : IsFrob A (normV w) (frob h : G E Ω)

@[simp] theorem **deg_frob** : deg (frob h) = 1

theorem **deg_surjective** : Function.Surjective (deg (h

theorem **exists_eq_zpow_mul** {φ : Weil h} (hφ : deg φ = 1) (σ : Weil h) : ∃ τ : Ine (F

@[simp] theorem **unrChar_apply** (c : ℂˣ) (σ : Weil h) : unrChar h c σ = c ^ deg σ

@[simp] theorem **coe_qPow** (s : ℝ) : (qPow w s : ℂ) = ((normV w : ℕ) : ℂ) ^ ((s : ℝ) : ℂ)

@[simp] theorem **qPow_zero** : qPow w 0 = 1

theorem **qPow_add** (s t : ℝ) : qPow w (s + t) = qPow w s * qPow w t

theorem **coe_qPow_one** : (qPow w 1 : ℂ) = ((normV w : ℕ) : ℂ)

theorem **norm_qPow** (s : ℝ) : ‖(qPow w s : ℂ)‖ = ((normV w : ℕ) : ℝ) ^ s

theorem **compactSpace_Ine** (A : ValuationSubring Ω) : CompactSpace (Ine (F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsOkPins

theorem **exists_normal** (q : S.D b) : ∃ g : S.G b, S.NormalFor b f (g • q)

theorem **normal_eq** {b' : S.β} {f' : S.D b' → A} {q : S.D b} {q' : S.D b'} (h : S.NormalFor b f q) (h' : S.NormalFor b' f' q') (e : ∀ r, (∃ g : S.G b, r = f (g • q)) ↔ ∃ g : S.G b', r = f' (g • q')) : f q = f' q'

theorem **exists_pref** (u : B.Word) (a : I.ι) (j : ℕ) (hj : j ≠ B.R.m) (b : (I.stp0 u a j).β) (q : (I.stp0 u a j).D b) : ∃ g : (I.stp0 u a j).G b, I.pref u a j b ((I.stp0 u a j).act b g q)

theorem **pref_rep** (u : B.Word) (a : I.ι) (j : ℕ) (hj : j ≠ B.R.m) (c : (I.stp u a j).Cl) : I.pref u a j c.1 ((I.stp u a j).rep c)

theorem **cosetIn_empty** (u : B.Word) (a : I.ι) (j : ℕ) (hj : j + 1 ≠ B.R.m) (q : PtAt B.R I.sel (I.tup a) (I.cst a) u j) (P) : cosetIn q P = ∅

theorem **laws** (h : I.Ok) : I.Laws

theorem **mem_formsK** (S : GStep E K n) (hS : S.Free) (hF : (S.fam emb lam).Ok) (x : S.Cl → VV K lam) : x ∈ S.FormsK emb lam hF

theorem **forms_equiv** (S S' : GStep E K n) (hS : S.Free) (hS' : S'.Free) (hF : (S.fam emb lam).Ok) (hF' : (S'.fam emb lam).Ok) (e0 : S.Cl ≃ S'.Cl) (hL : ∀ c, S'.Lsh _ (S'.rep (e0 c)) = S.Lsh _ (S.rep c)) : ∃ e : (S.hlat emb lam hF).V ≃ₗ[K] (S'.hlat emb lam hF').V, (∀ x, x ∈ (S.hlat emb lam hF).L ↔ e x ∈ (S'.hlat emb lam hF').L) ∧ ∀ x, S'.embF emb σ lam hF' (e x) = fun c => S.embF emb σ lam hF x (e0.symm c)

theorem **mt_finite** {R : Ranks} {J : ChainIn E K ι R} (Q : PerIn J) (h : Q.Ok) (hH : ∀ b, Q.H b = ⊤) : FiniteDimensional ℂ (Q.Mt emb σ)

theorem **law0** (h : I.Ok) (l : I.Laws) : I.Law0 l

theorem **ipC_herm** (l : I.Laws) (u : B.Word) (x y) : starRingEnd ℂ (((I.cols l).fm u).ipC 0 y x) = ((I.cols l).fm u).ipC 0 x y

theorem **ipC_pos** (l : I.Laws) (u : B.Word) (x) (hx : x ∈ ((I.cols l).mem u).A.Mu0) (h0 : x ≠ 0) : 0 < (((I.cols l).fm u).ipC 0 x x).re

theorem **fits** (l : I.Laws) : RowsRestModel.Fits (I.cols l) fun a => Module.finrank ℂ ((I.perIn B.u₀ a).Mt B.fields.emb B.embLL)

theorem **later** (l : I.Laws) (hB : StagesInhab.HasLetters B.toPre) : Nonempty (TC.Later B (I.cols l))

theorem **yStd_spec** (h : I.Ok) (u : B.Word) (i : ℕ) (c : (I.stp u I.mu0 i).Cl) : PtOf (I.yStd u i c * kStd X u (B.R.N i)) (extTo I.SQ ((I.stp u I.mu0 i).rep c).1.1)

theorem **kStd_spec** (X : PreX B.toPre) (u : B.Word) (n : ℕ) : (∃ p : DecoF B.fields.E n ∅ ∅ ∅, PtOf (kStd X u n) p ∧ p.deco.1.N = stdLat B.fields.E n) ∧ (∀ v : HeightOneSpectrum (𝒪 B.host.F), (B.T u : 𝒪 B.host.F) ∉ v.asIdeal → bc (prV (E

theorem **exists_pre** : ∃ (P : Pre) (X : PreX P) (f : Top.Fr P), StagesInhab.HasLetters P ∧ InEx.Oriented X f.P ∧ (letI

theorem **exists_ok_pins** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X), StagesInhab.HasLetters B.toPre ∧ I.Ok ∧ TC.Pins B X ∧ I.muStar ≠ I.mu0

theorem **exists_all** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)), I.Ok ∧ TC.Pins B X ∧ I.Law0 l ∧ I.muStar ≠ I.mu0 ∧ Nonempty (TwForms (I.rec7 l Z))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsPotential

theorem **exists_potential** {G A X : Type} [Group G] [CommGroup A] (χ : G →* A) (r : G → X → X → Prop) (h1 : ∀ x, r 1 x x) (hi : ∀ g x y, r g x y → r g⁻¹ y x) (hm : ∀ g g' x y z, r g x y → r g' y z → r (g' * g) x z) (hs : ∀ g x, r g x x → χ g = 1) : ∃ φ : X → A, ∀ g x y, r g x y → (φ x)⁻¹ * φ y = χ g

theorem **isIntegral_det** {E : Type} [Field E] [NumberField E] {N : ℕ} (L : Submodule (𝒪 E) (Fin N → E)) (hfg : L.FG) (hsp : Submodule.span E (L : Set (Fin N → E)) = ⊤) (γ : Matrix (Fin N) (Fin N) E) (hγ : ∀ v ∈ L, γ.mulVec v ∈ L) : IsIntegral ℤ γ.det

theorem **chi_apply** : letI

private theorem **prod_sigmaOf** (x : H.E) : ∏ τ : Tau R, H.sigmaOf ((τ : R.Psi) : R.GG) x = algebraMap H.K H.E (Algebra.norm H.K x)

theorem **chi_of_integral** : letI

theorem **exists_pot** (N : ℕ) : letI

theorem **norm_uI_zpow** (a b : ℤ) : ‖((uI ^ a * uI ^ b : ℂˣ) : ℂ)‖ = 1

theorem **two_lt_alt** (u : B.Word) : 2 < B.T u

theorem **exists_cols** (B : Base) : ∃ C : Cols B, (∀ u i, Nonempty ((C.mem u).C.cls i)) ∧ (∀ u, (C.mem u).A.Mram ≠ ⊥) ∧ (∀ u i, (C.fm u).Fi i ≠ ⊥) ∧ C.mu0 ≠ C.muStar ∧ C.mu0 ≠ C.muStar' ∧ C.muStar ≠ C.muStar' ∧ Function.Injective C.chiD ∧ ∀ u i c, C.chiD (C.detCls u i c) ≠ 1

theorem **nonempty_cols0** (B : Pre) : Nonempty (Cols0 B)

theorem **T_win** (C : Cols0 B) (u : B.Word) : (C.mem u).A.Tstar ≤ ((C.mem u).A.T : ℝ) ∧ ((C.mem u).A.T : ℝ) ≤ (1 + 1 / ((B.R.f : ℝ) * B.R.k)) * (C.mem u).A.Tstar

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsPreDataDyadicFrame

theorem **spacesExist** : SpacesExist

theorem **mod_of_modEq** {p c M : ℕ} (hc : 1 < c) (hcM : c ∣ M) (h : p ≡ 1 [MOD M]) : p % c = 1

private theorem **coprime_of_mod_one** {x a b c : ℕ} (h : x % (a * b * c) = 1) : Nat.Coprime x a

theorem **win_eq** (T : ℝ) (hT : 0 < T) (n : ℝ) (hn : 0 < n) : (T ^ n * 4 / ((1 : ℕ) : ℝ)) ^ ((1 : ℝ) / n) * (2 : ℝ) ^ (-(2 : ℝ) / n) = T

theorem **exists_primes** (hP : NormPrimes) (R : Ranks) (H : Host R) : ∃ (g : Fin 2 → ℕ) (π : Fin 2 → 𝒪 H.E) (p p' : ℕ), (∀ i, (g i).Prime) ∧ g 0 < g 1 ∧ g 1 < p ∧ p < p' ∧ p.Prime ∧ p'.Prime ∧ p % 15 = 1 ∧ p' % 15 = 1 ∧ p % (H.QS * H.q * H.d) = 1 ∧ p' % (H.QS * H.q * H.d) = 1 ∧ (∀ i, g i % (H.QS * H.q * H.d) = 1) ∧ (∀ i, (π i : H.E) ∈ H.K) ∧ (∀ i, (π i : H.E) * H.conj (π i : H.E) = (g i : H.E)) ∧ (∀ i, ((H.QS * H.q * H.d : ℕ) : 𝒪 H.E) ∣ π i - 1) ∧ 2 * R.f * R.k + 1 < g 0 ∧ Disjoint (Formalism.CyclotomicNorms.BB R.r (g 0)) (Formalism.CyclotomicNorms.BB R.r (g 1)) ∧ 1 < H.QS * H.q * H.d

theorem **exists_preX_host** (hP : NormPrimes) (hS : SpacesExist) (R : Ranks) (H₀ : Host R) : ∃ B : Pre, B.R = R ∧ HEq B.host.toHostAx H₀.toHostAx ∧ Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **exists_preX** (hP : NormPrimes) (hS : SpacesExist) (R : Ranks) (hR : Nonempty (Host R)) : ∃ B : Pre, B.R = R ∧ Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **exists_preX_some** (hP : NormPrimes) (hS : SpacesExist) : ∃ B : Pre, Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **exists_preX_ranks** (hP : NormPrimes) (R : Ranks) (hR : Nonempty (Host R)) : ∃ B : Pre, B.R = R ∧ Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **exists_preX_two_words** (hP : NormPrimes) : ∃ B : Pre, Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **embLL_tauLL** (B : Base) (τ : Tau B.R) : B.embLL.comp (B.tauLL τ) = B.host.emb τ

theorem **T_big** (B : Pre) (X : PreX B) (u : B.Word) : 2 * B.R.f * B.R.N B.R.m + 1 < B.T u

theorem **T_ne_g** (B : Pre) (X : PreX B) (u : B.Word) (i : Fin 2) : B.T u ≠ B.host.g i

theorem **normPrimes** : NormPrimes

theorem **exists_preX_all_ranks** (R : Ranks) (hR : Nonempty (Host R)) : ∃ B : Pre, B.R = R ∧ Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **exists_preData_dyadicFrame** (R : Ranks) (hφ : R.toFrame = Formalism.HC.cyclicFrame R.e) (he : 2 ≤ R.e) : ∃ B : Pre, B.R = R ∧ Nontrivial B.Word ∧ Nonempty (PreX B)

theorem **tauLL_injective** (B : Base) : Function.Injective B.tauLL

theorem **piG_ne_one** (B : Pre) (X : PreX B) (i : Fin 2) : (X.piG i : B.host.E) ≠ 1

theorem **sizes_nondeg** (B : Pre) (X : PreX B) : 0 < X.a ∧ ∀ i, 2 * B.R.f * B.R.k + 1 < B.host.g i

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsSStageUnconditional

theorem **conjF_ne** (B : Pre) : B.conjF ≠ AlgHom.id B.host.F B.host.E

theorem **ncard_primesOver** (E : Type) [Field E] [NumberField E] (p : ℕ) (hp : p.Prime) : ((Ideal.span {(p : ℤ)}).primesOver (𝒪 E)).ncard = (placesOver E p).ncard

theorem **exists_atP** (B : Pre) (p : HeightOneSpectrum (𝒪 B.host.F)) : ∃ L : LocAt B.host.F B.host.E B.conjF p, (L.split.isSome ↔ resChar p ∈ ({B.host.g 0, B.host.g 1} : Finset ℕ))

theorem **exists_conjIsom** (V : Type) [NormedAddCommGroup V] [InnerProductSpace ℂ V] {a b : ℕ} (onb : OrthonormalBasis (Fin a) ℂ V) (hab : a ≤ b) : Nonempty (V ≃ₗᵢ⋆[ℂ] EuclideanSpace ℂ (Fin b))

theorem **exists_sStage** {B : Base} {C : Cols B} (W : RowsRest B C) (h1 : CodeWords B.toPre) (h2 : ColsPos C) (h3 : W.Drows ≤ W.Dall) (d : Tau B.R → ℝ) (hd : ∀ ρ, 0 < d ρ) (P : ∀ u, (C.mem u).A.Mu0 →ₗ[ℂ] EuclideanSpace ℂ (Fin W.Dall)) : ∃ S : SStage B C W, S.dsig = d ∧ (∀ u u', Nonempty (S.s.Term u u')) ∧ ∃ e : S.s.HH ≃ₗᵢ[ℂ] EuclideanSpace ℂ (Fin W.Dall), ∀ u x, e (S.s.Pu u x) = P u (S.s_Cu u x)

theorem **codeWords_of** {B : Base} {C : Cols B} {W : RowsRest B C} (S : SStage B C W) : CodeWords B.toPre

theorem **colsPos_of** {B : Base} {C : Cols B} {W : RowsRest B C} (S : SStage B C W) : ColsPos C

theorem **rows_le_of** {B : Base} {C : Cols B} {W : RowsRest B C} (S : SStage B C W) : W.Drows ≤ W.Dall

theorem **nonempty_iff** {B : Base} {C : Cols B} (W : RowsRest B C) : Nonempty (SStage B C W) ↔ CodeWords B.toPre ∧ ColsPos C ∧ W.Drows ≤ W.Dall

theorem **low_end_empty** {B : Base} {C : Cols B} (W : RowsRest B C) (w : B.Word) (τ : Tau B.R) (l : ℕ) (h : l + 1 < B.R.k - 1) (hG : 2 ≤ (B.hcW w τ).v (l, by omega) - (B.hcW w τ).v (l + 1, h)) (hu : (B.hcW w τ).u (l + 1, by omega) = (B.hcW w τ).v (l + 1, h) + 1 / 2) : IsEmpty (SStage B C W)

theorem **exists_codeParams** (R : Ranks) (Del : ℕ) (hD : 1 ≤ Del) : ∃ P : (Fin (R.k - 2) → Bool) → HCParam R, (∀ c c', (P c).v = (P c').v ∧ (P c).b = (P c').b ∧ (P c).e = (P c').e) ∧ (∀ c c' (h : 0 < R.k), (P c).u ⟨0, h⟩ = (P c').u ⟨0, h⟩ ∧ (P c).u (R.k - 1, by omega) = (P c').u (R.k - 1, by omega)) ∧ (∀ c (l : ℕ) (h : l + 1 < R.k - 1), (P c).v (l, by omega) - (P c).v (l + 1, h) = 4 * (Del : ℚ)) ∧ ∀ c (l : ℕ) (h : l < R.k - 2), (P c).u (l + 1, by omega) = (P c).v (l, by omega) - 1 / 2 - (if c (l, h) then (Del : ℚ) else 0)

theorem **exists_pre_codeWords** (B : Pre) : ∃ B' : Pre, B'.R = B.R ∧ HEq B'.host B.host ∧ CodeWords B' ∧ Fintype.card B'.Word = 2 ^ (B.R.k - 2)

theorem **colsPos_tcols** {B : Base} (C : Cols B) : ColsPos (RowsRestModel.tcols C)

theorem **exists_sStage_unconditional** (n : ℕ) (hn : 0 < n) : ∃ (B : Base) (C : Cols B) (W : RowsRest B C) (S : SStage B C W), Fintype.card B.Word = 2 ^ (B.R.k − 2) ∧ 3 ≤ B.R.k − 2 ∧ W.Drows = n ∧ Module.finrank ℂ S.s.HH = 2 * n ∧ (∀ u, Nontrivial (S.s.Cu u)) ∧ (∀ u, S.s.Pu u ≠ 0) ∧ ∀ u u', Nonempty (S.s.Term u u')

theorem **isEmpty_of_lt** {B : Base} {C : Cols B} (W : RowsRest B C) (h : W.Dall < W.Drows) : IsEmpty (SStage B C W)

theorem **delp_nonneg** {B : Base} {C : Cols B} {W : RowsRest B C} (S : SStage B C W) (h : 0 < W.Dall) : 0 ≤ W.delp

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsSameData

theorem **exists_restRows** (r₀ r : Rows.Data R) (hTau : HEq r₀.Tau r.Tau) (hinstTau : HEq r₀.instTau r.instTau) : ∃ x, r = overRows r₀ x

theorem **exists_restN** {r₀ r : Rows.Data R} (N₀ : NData R H r₀) (N : NData R H r) (hOm : HEq N₀.Om N.Om) (hinstOm : HEq N₀.instOm N.instOm) (halgOm : HEq N₀.algOm N.algOm) (hplOm : HEq N₀.plOm N.plOm) (hPlG : HEq N₀.PlG N.PlG) (hfinPlG : HEq N₀.finPlG N.finPlG) : ∃ x, N = overN N₀ r x

theorem **exists_restX** (X₀ : NTw R H r₀ L₀ N₀) (X : NTw R H r L (overN N₀ r n)) (hcc : HEq X₀.cc X.cc) (hTw : HEq X₀.Tw X.Tw) (hl : HEq X₀.l X.l) (hO0 : HEq X₀.O0 X.O0) (hcrO0 : HEq X₀.crO0 X.crO0) (halgO0 : HEq X₀.algO0 X.algO0) (htopO0 : HEq X₀.topO0 X.topO0) (hO : HEq X₀.O X.O) (hcrO : HEq X₀.crO X.crO) (halgO : HEq X₀.algO X.algO) (hOk : HEq X₀.Ok X.Ok) (hcrOk : HEq X₀.crOk X.crOk) (halgOk : HEq X₀.algOk X.algOk) (hzk : HEq X₀.zk X.zk) (hpD : HEq X₀.pD X.pD) (hdelta : HEq X₀.delta X.delta) (hchi1 : HEq X₀.chi1 X.chi1) (hSig : HEq X₀.Sig X.Sig) (hfinSig : HEq X₀.finSig X.finSig) (hjS : HEq X₀.jS X.jS) (hside : HEq X₀.side X.side) (hrhoS : HEq X₀.rhoS X.rhoS) (hS0 : HEq X₀.S0 X.S0) (hB : HEq X₀.B X.B) (hacgB : HEq X₀.acgB X.acgB) (hmodB : HEq X₀.modB X.modB) (hTheta : HEq X₀.Theta X.Theta) (hhk : HEq X₀.hk X.hk) (hhEig : HEq X₀.hEig X.hEig) (hLam : HEq X₀.abelian₂ X.abelian₂) (hsideL : H

theorem **exists_restY** {X₀ : NTw R H r₀ L₀ N₀} {x : RestX X₀ r L n} (Y₀ : NExp R H r₀ L₀ N₀ X₀) (Y : NExp R H r L (overN N₀ r n) (overX X₀ r L n x)) (hFm : HEq Y₀.Fm Y.Fm) (hacgFm : HEq Y₀.acgFm Y.acgFm) (hmodFmK : HEq Y₀.modFmK Y.modFmK) (hmodFmO : HEq Y₀.modFmO Y.modFmO) (hactG : HEq Y₀.actG Y.actG) (hU : HEq Y₀.U Y.U) (hpsiU : HEq Y₀.psiU Y.psiU) (heAlg : HEq Y₀.eAlg Y.eAlg) (hBsup : HEq Y₀.Bsup Y.Bsup) (hDat : HEq Y₀.Dat Y.Dat) (hth : HEq Y₀.th Y.th) (hlev : HEq Y₀.lev Y.lev) (hthPsi : HEq Y₀.thPsi Y.thPsi) (hthU : HEq Y₀.thU Y.thU) (hdelU : HEq Y₀.delU Y.delU) (hMtV : HEq Y₀.MtV Y.MtV) (hacgMt : HEq Y₀.acgMt Y.acgMt) (hmodMtK : HEq Y₀.modMtK Y.modMtK) (hmodMtO : HEq Y₀.modMtO Y.modMtO) (hMtLat : HEq Y₀.MtLat Y.MtLat) (hthP : HEq Y₀.thP Y.thP) (hhkF : HEq Y₀.hkF Y.hkF) (hembB : HEq Y₀.embB Y.embB) : ∃ y, Y = overY Y₀ r L n x y

theorem **CopyTw.exists_rest** {D D' : D7} {ι : D.what₂.LL →+* D'.what₂.LL} {e : ıD'.LL0 ≃+* ıD.LL0} (h : CopyTw D D' ι e) : ∃ r, D' = overD D r

theorem **SameTwist.refl** {A : Type} [CommGroup A] (t : Twist D A) : SameTwist t t

theorem **SameTwist.eq** {A : Type} [CommGroup A] {t t' : Twist D A} (h : SameTwist t t') : t = t'

theorem **SameBase.refl** {S T : HSp D} (B : Base S T) : SameBase B B

theorem **SameBase.eq** {S T : HSp D} {B B' : Base S T} (h : SameBase B B') : B = B'

theorem **SameData.refl** (ρ : Setup D.ntw) : SameData ρ ρ

theorem **SameData.eq** {ρ ρ' : Setup D.ntw} (h : SameData ρ ρ') : ρ = ρ'

theorem **sameTwist_dn** {A : Type} [CommGroup A] {t : Twist D A} {t' : Twist (overD D r) A} (h : SameTwist t t') : t = twistDn r t'

theorem **sameBase_up** {S T : HSp D} (B : Base S T) : SameBase B (baseUp r B)

theorem **sameData_up** (ρ : Setup D.ntw) : SameData ρ (setupUp r ρ)

theorem **SameTwist.symm** {A : Type} [CommGroup A] {t : Twist D A} {t' : Twist D' A} (h : SameTwist t t') : SameTwist t' t

theorem **SameTwist.trans** {D'' : D7} {A : Type} [CommGroup A] {t : Twist D A} {t' : Twist D' A} {t'' : Twist D'' A} (h : SameTwist t t') (h' : SameTwist t' t'') : SameTwist t t''

theorem **SameBase.eq_of** {S T : HSp D} {S' T' : HSp D'} {B : Base S T} {B₁ B₂ : Base S' T'} (h : SameBase B B₁) (h' : SameBase B B₂) : SameBase B₁ B₂

theorem **SameData.eq_up** (r : Rest D) {ρ : Setup D.ntw} {ρ' : Setup (overD D r).ntw} (h : SameData ρ ρ') : ρ' = setupUp r ρ

theorem **CopyTw.exists_sameData** {ι : D.what₂.LL →+* D'.what₂.LL} {e : ıD'.LL0 ≃+* ıD.LL0} (h : CopyTw D D' ι e) (ρ : Setup D.ntw) : ∃ ρ' : Setup D'.ntw, SameData ρ ρ'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsSameFOfDiv

theorem **SPack.re_ψ** : HEq (P.re Z h hF).ψ P.ψ

theorem **conjHH_eEECre** {X : HidaSetting D} (G : GoodOp X) (x : EEC X G) : conjHH (reX Z X) (reG Z G) (eEECre Z G x) = eHHre Z X (conjHH X G x)

theorem **exists_sameF_ofDiv** (N : ∀ u, Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) (ht : D.what₂.Word → ℝ) (F : ∀ u, Forms0 D u) : ∃ F' : ∀ u, Forms0 (reD D (New.ofDiv D N ht)) u, SameForms0LevelsAndValues (New.ofDiv D N ht) F F'

theorem **SPack.exists_re_ofDiv** (N : ∀ u, Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) (ht : D.what₂.Word → ℝ) {F : ∀ u, Forms0 D u} (P : SPack D F) : ∃ (F' : ∀ u, Forms0 (reD D (New.ofDiv D N ht)) u) (P' : SPack (reD D (New.ofDiv D N ht)) F'), P'.X = reX (New.ofDiv D N ht) P.X

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsSplittingCharacterSelf

theorem **admissibleEven_trivial** {V W : HSp D} (h : V.rank = W.rank) (hV : Even V.rank) : AdmE V W 1 1

theorem **exists_splittingCharacter_self** {V W : HSp D} (e : V.rank = W.rank) : ∃ χ : IdC D, AdmE V W χ χ

theorem **IsHC.mul** {χ ν : IdC D} (h : IsHC D χ) (k : IsHC D ν) : IsHC D (χ * ν)

theorem **AdmE.mul_right** {V W : HSp D} {χ χ' ν : IdC D} (h : AdmE V W χ χ') (h1 : IsHC D ν) (h2 : ResEta D ν 0) : AdmE V W χ (χ' * ν)

theorem **AdmE.mul_left** {V W : HSp D} {χ χ' ν : IdC D} (h : AdmE V W χ χ') (h1 : IsHC D ν) (h2 : ResEta D ν 0) : AdmE V W (χ * ν) χ'

theorem **WeilFun.patch_om_eq** {ψ : AddChar (AF D.host.F) ℂ} (Wf : WeilFun D ψ) (o) {V W : HSp D} (h : V.rank = W.rank) (χ χ') : (Wf.patch o).om V W χ χ' = o V W χ χ'

theorem **WeilFun.patch_om_ne** {ψ : AddChar (AF D.host.F) ℂ} (Wf : WeilFun D ψ) (o) {V W : HSp D} (h : V.rank ≠ W.rank) (χ χ') : (Wf.patch o).om V W χ χ' = Wf.om V W χ χ'

theorem **WeilFun.exists_eqRank** (he : WeilOnEqEx D) (ht : WeilTwEq D) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) (Wf : WeilFun D ψ) : ∃ Wf' : WeilFun D ψ, Wf'.EqRank ∧ ∀ V W χ χ', V.rank ≠ W.rank → Wf'.om V W χ χ' = Wf.om V W χ χ'

theorem **isWeilOn_both_zero** (ψ : AddChar (AF D.host.F) ℂ) {V W : HSp D} (hV : V.rank = 0) (hW : W.rank = 0) (χ χ' : IdC D) : IsWeilOn ψ V W χ χ' 1

theorem **splitFac_twist_eq** {V W : HSp D} (h : V.rank = W.rank) (χ χ' ν : IdC D) (x y : (AE D)ˣ) : splitFac V W χ (χ' * ν) x y = (ν x)⁻¹ * splitFac V W χ χ' x y ∧ splitFac V W (χ * ν) χ' x y = ν y * splitFac V W χ χ' x y

theorem **Realises.exists_liftEq** (h : Realises B Wf Tw Rc) (hE : Wf.EqRank) (hL : LiftAutEq D) (V V' : HSp D) (χ χ' : X.Chr) : ∃ t : X.Sch V V' →ₗ[ℂ] X.cusp V →ₗ[ℂ] X.AF V', (¬ (AdmE V V' (T.eChr χ) (T.eChr χ') ∧ V.SigOK) → t = X.th V V' χ χ') ∧ (AdmE V V' (T.eChr χ) (T.eChr χ') → V.SigOK → ∀ μ 𝓕, IsVolPair D V μ 𝓕 → ∀ Φ F x, T.ev V' (t Φ F) x = (T.petC V : ℂ) * ∫ g in 𝓕, W.evS V V' (W.om V V' χ χ' (g, x) Φ) 0 * T.ev V (F : X.AF V) g ∂μ)

theorem **Realises.exists_realisesE** (h : Realises B Wf Tw Rc) (hE : Wf.EqRank) (hL : LiftAutEq D) : ∃ (X' : ThetaX D) (T' : ThetaTied D X') (W' : WeilTied D X' T') (B' : BesselTied D X' T' W') (e : W'.ψ = W.ψ), RealisesE (θ

theorem **realisedExE_of** (h : RealisedEx D) (hL : LiftAutEq D) : RealisedExE D

theorem **eq_of_isInfFam** {θ' : InfF D} (h' : IsInfFam θ') : θ' = θ

theorem **afTiedSm** : ThetaSmoothnessCondition T

theorem **formSub** : FormSub D

theorem **typTied** (ht : TypAdd D) : TypTied T

theorem **weilUnrTied** : WeilUnramifiedCondition W

theorem **schEquiv_om** (V V' : HSp D) (χ χ') (p) (Φ : X.Sch V V') : h.schEquiv V V' (W.om V V' χ χ' p Φ) = Wf.om V V' (T.eChr χ) (T.eChr χ') p (h.schEquiv V V' Φ)

theorem **eqTied** : WeilThetaCompatibility W

theorem **allTied** (ht : TypAdd D) {A : ProvA} (hr : RealTied D) (hp : ProvisionCompatibility D A) : Nonempty (AllTied D X A)

theorem **petC_eq** (V : HSp D) : T.petC V = Rc.petC V

theorem **tieRowIsom** (hi : Rc.RowIsom) : X.TieRowIsom

theorem **tieBm** (hi : Rc.TopTie) : X.TieBm

theorem **carY_eq** (μ s i) : T.eChr (X.carY μ s i) = Rc.carY μ s i

theorem **tieIo** (hi : Rc.IoTie Wf) : X.TieIo

theorem **RecFun.exists_chrIs** (C : Formalism.L2.Host.CharFrame D) (xi : D.what₂.Tup → Formalism.L2.Host.HChar D.host.E) (hf : ∀ μ, ∃ n : ℕ, 0 < n ∧ (xi μ).OrderDvd n) (ht : ∀ μ, (xi μ).TrivOn D.host.F) (θ : InfF D) (ψ : AddChar (AF D.host.F) ℂ) : ∃ Rc : RecFun D θ ψ, Rc.ChrIs C xi

theorem **RecordIsForms.what₂** {C : Formalism.L2.Host.CharFrame D} {xi : D.what₂.Tup → Formalism.L2.Host.HChar D.host.E} (h : RecordIsForms D C xi) : RowsAreForms D C xi

theorem **RealisesE.chrRead** {θ : InfF D} {X : ThetaX D} {T : ThetaTied D X} {W : WeilTied D X T} {B : BesselTied D X T W} {Wf : WeilFun D W.ψ} {Tw : Tow D} {Rc : RecFun D θ W.ψ} (h : RealisesE B Wf Tw Rc) {C : Formalism.L2.Host.CharFrame D} {xi : D.what₂.Tup → Formalism.L2.Host.HChar D.host.E} (k : Rc.ChrIs C xi) : ChrRead T C xi

theorem **exists_chrRead** (hX : RealisedExE D) (hW : WeilEx D) (he : WeilOnEqEx D) (htw : WeilTwEq D) (Tw : Tow D) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (C : Formalism.L2.Host.CharFrame D) (xi : D.what₂.Tup → Formalism.L2.Host.HChar D.host.E) (hf : ∀ μ, ∃ n : ℕ, 0 < n ∧ (xi μ).OrderDvd n) (ht : ∀ μ, (xi μ).TrivOn D.host.F) : ∃ (X : ThetaX D) (T : ThetaTied D X) (W : WeilTied D X T), WeilThetaCompatibility W ∧ ChrRead T C xi

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsStagesClauses

theorem **ev_and** {p q : Ctx → Prop} (hp : Ev p) (hq : Ev q) : Ev fun K => p K ∧ q K

theorem **ev_imp** {p q : Ctx → Prop} (hp : Ev p) (h : ∀ K, p K → q K) : Ev q

theorem **ev_all** {p : Ctx → Prop} (h : ∀ K, p K) : Ev p

theorem **rowLay_iff** {D : D7} (Y : Lay D) : RowLay D Y.p0 Y.X ↔ ∀ w, D.what₂.muAt w = D.what₂.mu0 → ∀ j, 1 ≤ j → j ≤ D.R.m → D.what₂.Ij D.what₂.mu0 j w = 0 ∧ D.what₂.jw D.what₂.mu0 j w = 0

theorem **del_of_over** (K : Ctx) (X : Stages) (hO : K.Over X.B.toPre) : X.S.s.Del = (Delv K.s K.cD : ℚ)

theorem **exists_generator** (B : Base) (k : Fin 2) : ∃ w : (ZMod (B.host.g k))ˣ, orderOf w = B.host.g k - 1

theorem **exists_forced** (K : Ctx) (P : Pre) : ∃ S : Finset ℕ, ∀ p : ℕ, p.Prime → ((∃ i, p = P.host.g i) ∨ p = P.host.d ∨ (p : ℝ) ≤ K.s.X'' P.R.n) → p ∈ S

theorem **lg_forced** (K : Ctx) : large₂ K fun B _ T => ∀ p : ℕ, p.Prime → ((∃ i, p = B.host.g i) ∨ p = B.host.d ∨ (p : ℝ) ≤ K.s.X'' B.R.n) → p ∈ T.pl

theorem **isEmpty_core_untw** (K : Ctx) (X : Stages) (f : Fr X.B.toPre) : IsEmpty (Core K X f (TwDat.untw X.B ∅))

theorem **mem_iff** {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {Wt : Set X.B.places.Pl} {xi : X.C.Tup → HChar X.B.host.E} (h : MuIsW X f T Wt xi) (w : X.B.places.Pl) : w ∈ Wt ↔ X.W.muAt w ≠ X.C.mu0

theorem **conj_mem** {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {Wt : Set X.B.places.Pl} {xi : X.C.Tup → HChar X.B.host.E} (h : MuIsW X f T Wt xi) {w : X.B.places.Pl} (hw : w ∈ Wt) : X.B.places.conj w ∈ Wt

theorem **Q_off** {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {Wt : Set X.B.places.Pl} {xi : X.C.Tup → HChar X.B.host.E} (h : MuIsW X f T Wt xi) {w : X.B.places.Pl} (hw : w ∉ Wt) : T.Q w = ∅

theorem **tup_cases** {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {Wt : Set X.B.places.Pl} {xi : X.C.Tup → HChar X.B.host.E} (h : MuIsW X f T Wt xi) (μ : X.C.Tup) : μ = X.C.mu0 ∨ μ = X.C.muStar ∨ μ = X.C.muStar' ∨ ∃ w ∈ Wt, μ = X.W.muAt w

theorem **fval_untw** {K : Ctx} {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} (c : Core K X f T) (L : SizeData₂ X.toD7) (a : L.sc = K.s) (b : WtEq L K.a K.b) (d : Val K.C K.cD K.ch L) (w : X.B.places.Pl) (hw : X.W.muAt w = X.C.mu0) : ∀ j ∈ SizeData₂.lifts X.toD7, Fval L j w = 0

theorem **wj_pos** {D : D7} (L : SizeData₂ D) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (τ : D.what₂.Tau) : 0 < wj D j τ

theorem **framD_neg** {D : D7} (L : SizeData₂ D) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : FRamD D j < 0

theorem **mem_of_ramD** {K : Ctx} {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} (c : Core K X f T) (L : SizeData₂ X.toD7) (a : L.sc = K.s) (b : WtEq L K.a K.b) (d : Val K.C K.cD K.ch L) (w : X.B.places.Pl) (hg : X.B.places.resChar w ∉ L.gTypes) (hw : X.toD7.host.cls (X.B.places.resChar w) = PrimeClass.ramD) : w ∈ c.Wt

theorem **ev_recOK** : Ev fun K => ∀ (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (_ : Core K X f T), RecOK K.s K.C K.cD X.toD7

theorem **ev_exists_sizeData** : Ev fun K => ∀ (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (_ : Core K X f T), ∃ L : SizeData₂ X.toD7, L.sc = K.s ∧ WtEq L K.a K.b ∧ Val K.C K.cD K.ch L

theorem **ev_ramD_mem** : Ev fun K => ∀ (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (c : Core K X f T) (w : X.B.places.Pl), X.B.places.resChar w = X.B.host.d → w ∈ c.Wt

theorem **core_ext** {K : Ctx} {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} (c c' : Core K X f T) (h : c.O = c'.O) : c = c'

theorem **someBase_pre** (P : Pre) : (someBase P).toPre = P

theorem **ksCond_fixed** : KsCond fun B _ => B.KsC = (someBase B.toPre).KsC

theorem **not_exTrue_false** {KP : ∀ B : Base, Fr B.toPre → Prop} (hK : KsCond KP) : ¬ ExTrue KP fun _ _ _ _ _ => False

theorem **not_ksCond_false** : ¬ KsCond fun _ _ => False

theorem **not_exTrue_confined** {KP : ∀ B : Base, Fr B.toPre → Prop} (hK : KsCond KP) {Q : Slot} (g : Ctx → ∀ P : Pre, Fr P → IntermediateField ℚ ℂ → IntermediateField ℚ ℂ → Finset ℕ) (hQ : ∀ K X f T c, Q K X f T c → ∀ w ∈ c.Wt, X.B.places.resChar w ∈ g K X.B.toPre f X.B.KsC X.B.fixedCoefficientFieldInComplex) : ¬ ExTrue KP Q

theorem **exTrue_mono** {KP : ∀ B : Base, Fr B.toPre → Prop} {Q Q' : Slot} (h : ∀ K X f T c, Q K X f T c → Q' K X f T c) (hE : ExTrue KP Q) : ExTrue KP Q'

theorem **export_anti** {Q Q' : Slot} (h : ∀ K X f T c, Q' K X f T c → Q K X f T c) (hE : Export Q) : Export Q'

theorem **retwist_mono** {Q Q' Q'' : Slot} (h : ∀ K X f T c, Q' K X f T c → Q'' K X f T c) (hR : Retwist Q Q') : Retwist Q Q''

theorem **retwist_and** {Q Q' : Slot} {p : Ctx → Prop} (hp : Ev p) (hR : Retwist Q Q') : Retwist Q fun K X f T c => Q' K X f T c ∧ Q K X f T c ∧ p K

theorem **FromCore.elim** {Q : Slot} {D : D7} {Y : Lay D} (h : FromCore Q D Y) : ∃ (K : Ctx) (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (c : Core K X f T) (e : D = X.toD7), Q K X f T c ∧ layObj (e ▸ Y) = c.O

theorem **lidx_zero** {R V : Type} [CommRing R] [AddCommGroup V] [Module R V] [Subsingleton V] (p : Ideal R) [p.IsPrime] (A B : Submodule R V) : Formalism.HermitianLatticesTheory.lidx p A B = 0

theorem **exists_rowLay_of** (X : Stages) (η : ℝ) : ∃ (W : RowsRest X.B X.C) (V : DivStage X.B X.C W) (S : SStage X.B X.C W) (N : NStage X.B X.C W), W.muAt = X.W.muAt ∧ W.muTw = X.W.muTw ∧ W.Drows = X.W.Drows ∧ V.div.etaF = η ∧ DQ.MtStableAt (assemble X.B X.C W V S N X.G) ∧ ∃ p0 Z, RowLay (assemble X.B X.C W V S N X.G) p0 Z

theorem **exists_rowLay** : ∃ (D : D7) (p0 : BaseLevelRowSide D) (Z : DivX D), 0 < D.what₂.Drows ∧ RowLay D p0 Z

theorem **hasLetters_of_over** (K : Ctx) (P : Pre) (hO : K.Over P) : StagesInhab.HasLetters P

theorem **exists_muIsW** (B : Base) (hB : StagesInhab.HasLetters B.toPre) (f : Fr B.toPre) (T : TwDat B) (Wt : Set B.places.Pl) (hc : ∀ w, B.places.conj w ∈ Wt ↔ w ∈ Wt) (hs : ∀ w ∈ Wt, B.places.resChar w ∈ T.pl) (hu : ∀ w, w ∉ Wt → T.psi w = 1) : ∃ (C : Cols B) (W : _) (V : _) (S : _) (N : _) (G : _) (xi : C.Tup → HChar B.host.E), MuIsW (B, C, W, V, S, N, G) f T Wt xi

theorem **exists_muIsW_proper** : ∃ (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (Wt : Set X.B.places.Pl) (xi : X.C.Tup → HChar X.B.host.E), MuIsW X f T Wt xi ∧ (∃ w ∈ Wt, T.psi w ≠ 1) ∧ ∃ w, w ∉ Wt ∧ X.B.places.resChar w ∈ T.pl

theorem **exists_stages_clauses** (K : Ctx) (B : Base) (hO : K.Over B.toPre) (f : Fr B.toPre) (T : TwDat B) (Wt : Set B.places.Pl) (hc : ∀ w, B.places.conj w ∈ Wt ↔ w ∈ Wt) (hs : ∀ w ∈ Wt, B.places.resChar w ∈ T.pl) (hu : ∀ w, w ∉ Wt → T.psi w = 1) : ∃ (C : Cols B) (W : RowsRest B C) (V : DivStage B C W) (S : SStage B C W) (N : NStage B C W) (G : GJStage B C) (xi : C.Tup → HChar B.host.E), MuIsW (B, C, W, V, S, N, G) f T Wt xi ∧ DQ.MtStableAt (assemble B C W V S N G) ∧ S.s.Del = (Delv K.s K.cD : ℚ) ∧ V.div.etaF = 6 * K.s.x ^ (-K.C.P.CF) ∧ ∃ p0 Z, RowLay (assemble B C W V S N G) p0 Z

theorem **loc_Q** {K : Ctx} {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} (c : Core K X f T) (a : X.N.nd.Loc) : X.N.nd.Q a = T.Q (X.N.nd.w a)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsStdFStdLvC

theorem **car_of_cols** {D : D7} (C : ColsTiedSym D) : CarH C.herm.base C.forms ∧ CarT C.herm.base C.forms C.herm.atT

theorem **existsUnique_stdDt** : ∃! dt, I.StdDt l Z dt

theorem **stdDt_surjective** {dt} (h : I.StdDt l Z dt) : Function.Surjective dt

theorem **existsUnique_qpIs** : ∃! Qp, I.QpIs Qp

theorem **exists_stdO** (hI : I.Ok) (hP : Pins B X) : ∃ Qp O, I.QpIs Qp ∧ I.StdO l Z Qp O

theorem **stdKer_iff** (t : KerEntTied ex) : I.StdKer l Z t ↔ I.StdKerTie l Z t

theorem **exists_stdKer** (hI : I.Ok) (hP : Pins B X) (O : ∀ u a, GL3Theta.LiftObj (I.rec7 l Z) ((I.rec7 l Z).chTw u a)) (E : KerEnt (I.rec7 l Z) Y ex.tw) (hQ : I.QpIs Y.Qp) (hO : I.StdO l Z Y.Qp O) (hU : ∀ u a i, i < B.R.m → UnfIsAll (O u a) i) (hE : EntOk E O) (hS : ∀ u w h, ThAtServed (ex.ker u w h) E) : ∃ t : KerEntTied ex, I.StdKer l Z t ∧ t.O = O

theorem **qpIs_of_stdKer** (t : KerEntTied ex) (h : I.StdKer l Z t) : I.QpIs Y.Qp

theorem **top_of_topK** (W : TwFormsTiedS (I.rec7 l Z) (I.tw l l0 Z) T J F) (h : I.TopK l Z l0 W) (u a ha h1 β) : I.TopBase l Z u a (W.ch u a ha B.R.m h1 le_rfl β).lev.U fun c => ((W.ch u a ha B.R.m h1 le_rfl (T.chOf u a B.R.m c)).y (c, rfl)).1

theorem **topK_of_stdW** (W : TwFormsTiedS (I.rec7 l Z) (I.tw l l0 Z) T J F) (h : I.StdW l Z l0 W) : I.TopK l Z l0 W

theorem **exists_topPt** (hI : I.Ok) (hP : Pins B X) (u : B.Word) (a : I.ι) : ∃ k : GA B (B.R.N B.R.m), (∃ p : DecoF B.fields.E (B.R.N B.R.m) ∅ ∅ ∅, PtOf k p ∧ p.deco.1.N = stdLat B.fields.E (B.R.N B.R.m)) ∧ (∀ (P : (B.primes X).vT u) (r j : Fin (B.R.N B.R.m)), r.1 + 1 = B.R.N B.R.m → HasRes P.1 (adAt B.host.F P.1 ((k⁻¹ : GA B (B.R.N B.R.m)).1 r j)) (if j = r then Ideal.Quotient.mk P.1.asIdeal (B.fields.conjO (X.piT u)) else 0)) ∧ I.TopPt l Z u a k

theorem **ipMt_smul** (hI : I.Ok) (hP : Pins B X) (Z : Later B (I.cols l)) (u : B.Word) (a : I.ι) (s t : ℂ) (G G' : (I.per l u a).Mt) : I.ipMt l u a (s • G) (t • G') = starRingEnd ℂ s * t * I.ipMt l u a G G'

theorem **not_funKilled_both** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} {n : ℕ} {J : Matrix (Fin n) (Fin n) E} {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 F))} (S : Formalism.L2.Forms.StepLevel F E σ J T f bad) {p q : 𝒪 E} (hpq : IsCoprime p q) (hp : FunKilled S p) (hq : FunKilled S q) (v : HeightOneSpectrum (𝒪 F)) : (T : 𝒪 F) ∉ v.asIdeal

theorem **readsKTop_killed** (hI : I.Ok) (hP : Pins B X) (u : B.Word) (a : I.ι) {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {Ch : Type} (chOf : (I.stp u a B.R.m).Cl → Ch) (K : ∀ β : Ch, LevelChart (I.rec7 l Z) (B.T u) bad B.R.m le_rfl (fun c : {c // chOf c = β} => (I.chain l u a).stab B.R.m c.1) (fun c => latCrd (I.rec7 l Z) B.R.m le_rfl (I.latAt u a B.R.m c.1))) (h : I.ReadsKTop l Z u a chOf K) :

∃ K' : ∀ β : Ch, LevelChart (I.rec7 l Z) (B.T u) bad B.R.m le_rfl (fun c : {c // chOf c = β} => (I.chain l u a).stab B.R.m c.1) (fun c => latCrd (I.rec7 l Z) B.R.m le_rfl (I.latAt u a B.R.m c.1)), I.ReadsKTop l Z u a chOf K' ∧ ∀ β, FunKilled (K' β).lev (X.piT u)

theorem **hform_last** {E : Type} [CommRing E] (σ : E →+* E) {N : ℕ} (hN : 0 < N) {J : Matrix (Fin N) (Fin N) E} (hJ : J = 1) {p : E} {z : Fin N → E} (hz : ∀ j, z j = if j.1 + 1 = N then p else 0) (x : Fin N → E) : hform σ J z x = σ p * x ⟨N - 1, by omega⟩

theorem **sigD_conj** (D : D7) (x : D.host.E) : (sigD D : D.host.E →+* D.host.E) x = D.host.conj x

theorem **exists_stdF_stdLvC** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∃ (Ch : B.Word → ℕ → Type) (chOf : ∀ u i, ((I.rec7 l Z).mem u).C.cls i → Ch u i) (F : ∀ u, FormsTiedSym (I.rec7 l Z) u (Ch u) (chOf u)), (∀ u, I.StdF l Z u (F u) ∧ I.StdLvC l Z u (F u)) ∧ (∀ u i, Finite (Ch u i)) ∧ (∀ u, Nonempty (Unique (Ch u B.R.m))) ∧ (∀ u u', (F u).gl = (F u').gl) ∧ CharacterCompatible (F B.u₀).gl B.R.m

theorem **levJ_Uloc_eq** (u : B.Word) {Ch Ch' : ℕ → Type} {chOf : ∀ i, ((I.rec7 l Z).mem u).C.cls i → Ch i} {chOf' : ∀ i, ((I.rec7 l Z).mem u).C.cls i → Ch' i} (F : FormsTiedSym (I.rec7 l Z) u Ch chOf) (F' : FormsTiedSym (I.rec7 l Z) u Ch' chOf') (h : I.StdLvC l Z u F) (h' : I.StdLvC l Z u F') (hg : F.gl = F'.gl) (hL : F.levJ.Lb = F'.levJ.Lb) (j : Fin 2) (v) (hv : ((I.rec7 l Z).host.g j : 𝒪 (I.rec7 l Z).host.F) ∈ v.asIdeal) : F.levJ.Uloc v = F'.levJ.Uloc v

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsTangentEquiv

theorem **le_preimage_affPt** (hT : IsAffineOpen T) : ⊤ ≤ affPt hT ⁻¹ᵁ T

theorem **valR_affPt** (hT : IsAffineOpen T) (a : Γ(X, T)) : valR (affPt hT) T a = a

theorem **spMap_affPt** (hT : IsAffineOpen T) (hT' : IsAffineOpen T') (h : T' ≤ T) : spMap (resG X h) ≫ affPt hT = affPt hT'

theorem **exists_coordOpen** {V : X.Opens} {x : X} (hx : x ∈ V) : ∃ T, CoordOpen t T ∧ x ∈ T ∧ T ≤ V

theorem **CoordOpen.mono** (h : CoordOpen t T) (hT' : IsAffineOpen T') (e : T' ≤ T) : CoordOpen t T'

theorem **tanEv_dUnit_eq** {R : Type} [CommRing R] {x : Sp R ⟶ X} {v : Sp (DualNumber R) ⟶ t.G.left} (hv : IsTanAtG t x v) (h : ⊤ ≤ x ⁻¹ᵁ T) {W : t.G.left.Opens} (hW : T ≤ unitSec t.G ⁻¹ᵁ W) (f : Γ(t.G.left, W)) : ⊤ ≤ v ⁻¹ᵁ W ∧ tanEv t hv h (dUnit t.G hW f) = (valR v W f).snd

theorem **CoordOpen.exists_equiv** (h : CoordOpen t T) : ∃ θ : TanM (affPt h.1) t.G ≃ₗ[Γ(X, T)] Γ(Lie t.G, T), ∀ v, IsSecOf h.1 v (θ v)

theorem **IsSecOf.eq_resM** (h : CoordOpen t T) {hT' : IsAffineOpen T'} (e : T' ≤ T) {v : TanM (affPt h.1) t.G} {v' : TanM (affPt hT') t.G} {l : Γ(Lie t.G, T)} {l' : Γ(Lie t.G, T')} (hl : IsSecOf h.1 v l) (hl' : IsSecOf hT' v' l') (hv : (tv v').left = spMap (dualMap (resG X e)) ≫ (tv v).left) : l' = resM _ e l

theorem **exists_tangent_equiv** {C : Scheme.{0}} {t' : SAB H C} {g : C ⟶ X} (m : SABMap t' t g) : ∃ Ψ : ∀ (T : C.Opens) (h : CoordOpen t' T) (x : Sp Γ(C, T) ⟶ X), affPt h.1 ≫ g = x → (TanM x t.G ≃ₗ[Γ(C, T)] Γ(Lie t'.G, T)), ∀ T h x hx T' h' x' hx' (e : T' ≤ T) u u', (tv u').left = spMap (dualMap (resG C e)) ≫ (tv u).left → Ψ T' h' x' hx' u' = resM _ e (Ψ T h x hx u)

theorem **Mdl.exists_point** (h1 : 1 ≤ Dw.p + Dw.q) (Z : Mdl Y i₀) {k : Type} [Field k] [IsAlgClosed k] (φ : Z.i.c.V W →+* k) : ∃ x : Sp k ⟶ Z.X, x ≫ Z.s = spMap φ

theorem **IntChart.left_vec_map** (q : IntChart Z) {R R' : Type} [CommRing R] [CommRing R'] (ψ : R →+* R') {x : Sp R ⟶ Z.X} {x' : Sp R' ⟶ Z.X} (hx : spMap ψ ≫ x = x') (h : ⊤ ≤ x ⁻¹ᵁ q.U) (h' : ⊤ ≤ x' ⁻¹ᵁ q.U) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : (tv (q.vec x' h' σ a)).left = spMap (dualMap ψ) ≫ (tv (q.vec x h σ a)).left

theorem **IntChart.exists_basis** (q : IntChart Z) {R : Type} [CommRing R] (x : Sp R ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ q.U) (σ : E →+* ℂ) : ∃ (n : ℕ) (e : Fin (Dw.mult σ) ↪ Fin n) (b : Module.Basis (Fin n) R (TanM (x ≫ Z.π) (Y Z.i).t.G)), ∀ a, b (e a) = q.vec x h σ a

theorem **isAffineOpen_inf** {T T' : c.C.Opens} (hT : IsAffineOpen T) (hT' : IsAffineOpen T') : IsAffineOpen (T ⊓ T')

theorem **exists_subsheaf** {α : Type} [Fintype α] (rk : α → ℕ) (σ : α → (E →+* ℂ)) (ix : ∀ a, Fin (Dw.mult (σ a))) (P : Prop) (hb : ∀ (q : IntChart Z) (R : Type) [CommRing R] (x : Sp R ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ q.U), ∃ (κ : Type) (_ : Fintype κ) (e : α ↪ κ) (b : Module.Basis κ R (TanM (x ≫ Z.π) (Y Z.i).t.G)), (P → Function.Surjective e) ∧ ∀ a, b (e a) = q.vec x h (σ a) (ix a)) (ht : ∀ (q q' : IntChart Z) (R : Type) [CommRing R] (x : Sp R ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ q.U) (h' : ⊤ ≤ x ⁻¹ᵁ q'.U) (a : α), ∃ cf : α → R, cf a = valR x _ (q'.chg q hZ (σ a) (ix a) (ix a)) ∧ (∀ b, b ≠ a → rk a ≤ rk b → cf b = 0) ∧ q'.vec x h' (σ a) (ix a) = ∑ b, cf b • q.vec x h (σ b) (ix b)) (S : Finset α) (hS : ∀ a ∈ S, ∀ b, rk b < rk a → b ∈ S) : ∃ (G : c.C.Modules) (i : G ⟶ c.lie), IsCohSub i ∧ deg k G = ∑ a ∈ S, c.degStep hZ (σ a) (ix a) ∧ (P → S = Finset.univ → IsIso i)

theorem **isMaxSubDeg_Pm** : IsMaxSubDeg k c.lie c.Pm

theorem **sum_degStep_mem** (σ : E →+* ℂ) (n : ℕ) : ∑ m ∈ Finset.range n, c.degStep hZ σ m ∈ subDegs k c.lie

theorem **sum_degStep_le** (σ : E →+* ℂ) (n : ℕ) : ∑ m ∈ Finset.range n, c.degStep hZ σ m ≤ c.Pm

theorem **deg_lie** (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) (hu : ¬ p ∣ (discr E).natAbs) : deg k c.lie = ∑ σ, ∑ m ∈ Finset.range (Dw.mult σ), c.degStep hZ σ m

theorem **sum_emb_eq** (D : D7) (F : (D.host.E →+* ℂ) → ℤ) : ∑ σ, F σ = ∑ᶠ τ : PsiT D, (F (embOf D τ) + F (conjEmb (embOf D τ)))

theorem **degLie_false** (τ : PsiT D) : ct.degLie c τ false = ∑ m ∈ Finset.range (S.Dw.mult (embOf D τ)), c.degStep ct.hasDR (embOf D τ) m

theorem **degLie_true** (τ : PsiT D) : ct.degLie c τ true = ∑ m ∈ Finset.range (S.Dw.mult (conjEmb (embOf D τ))), c.degStep ct.hasDR (conjEmb (embOf D τ)) m

theorem **eDeg_eq** (hu : DV.UnramE D p) : c.eDeg = -∑ᶠ τ, (ct.degLie c τ false + ct.degLie c τ true)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsTangentOrigin

theorem **valAt_algebraMap** {K : Type} [Field K] {Z : Scheme.{0}} [IsIntegral Z] (y : Sp K ⟶ Z) (a : Z.presheaf.stalk (below y)) : valAt y (algebraMap _ Z.functionField a) = (Scheme.stalkClosedPointTo y).hom a

theorem **valAt_eq_zero_iff** {K : Type} [Field K] {Z : Scheme.{0}} [IsIntegral Z] (y : Sp K ⟶ Z) {f : Z.functionField} (h : RegAt (below y) f) : valAt y f = 0 ↔ ¬ UnitAt (below y) f

theorem **valAt_mul** {K : Type} [Field K] {Z : Scheme.{0}} [IsIntegral Z] (y : Sp K ⟶ Z) {f g : Z.functionField} (h : RegAt (below y) f) (h' : RegAt (below y) g) : valAt y (f * g) = valAt y f * valAt y g

theorem **genAt_iff** {Δ : Y.Dv} (h : Y.bpf Δ (Set.range w)) (x : Y.X) (k : Fin n) : GenAt Y w x k ↔ Δ.GeneratesAt x (w k)

theorem **exists_genAt** {Δ : Y.Dv} (h : Y.bpf Δ (Set.range w)) (x : Y.X) : ∃ k, GenAt Y w x k

theorem **mem_cutSet_iff** (y : GeomPt Y E) {k : Fin n} (hk : GenAt Y w (below y.1) k) : y ∈ cutSet Y E w c ↔ ∀ i, ∑ l, coef Y E c i l * valAt y.1 (w l / w k) = 0

theorem **le_cutOrd_iff** (hS : (cutSet Y E w c).Nonempty) (Δ : Y.Dv) {f : Y.Fn} (hf : f ≠ 0) (hr : ∀ y ∈ cutSet Y E w c, Δ.RegularAt (below y.1) f) (a : ℕ) : a ≤ cutOrd Y E w c Δ f ↔ ∀ y ∈ cutSet Y E w c, ∃ b ∈ IsLocalRing.maximalIdeal (Y.X.presheaf.stalk (below y.1)) ^ a, algebraMap _ Y.Fn b = f ∗ (eqn Δ (below y.1) : Y.Fn)

theorem **cutOrd_zero** (Δ : Y.Dv) : cutOrd Y E w c Δ 0 = 0

theorem **genAt_crd** (x : (proj F N).X) (k : Fin (N + 1)) : GenAt (proj F N) (crd F N) x k ↔ (proj F N).ι x ∈ chart F N k

theorem **sum_unitCoef** (i : Fin N) (v : Fin (N + 1) → AlgebraicClosure E) : ∑ l, coef (proj F N) E (unitCoef E N) i l ∗ v l = v i.succ

theorem **mem_cutSet_unit** (y : GeomPt (proj F N) E) : y ∈ cutSet (proj F N) E (crd F N) (unitCoef E N) ↔ ∀ i : Fin N, (proj F N).ι (below y.1) ∉ chart F N i.succ

theorem **back_ι** : back F N ≫ (proj F N).ι = 𝟙 _

theorem **back_str** : back F N ≫ (proj F N).str = PSpace.str F N

theorem **ptOf_ι** (v : Fin (N + 1) → AlgebraicClosure E) (hv : v ≠ 0) : (ptOf F E N v hv).1 ≫ (proj F N).ι = pt ((algebraMap E (AlgebraicClosure E)).comp (algebraMap F E)) v (span_range_eq_top_iff_ne_zero.2 hv)

theorem **ptOf_mem_chart** (v : Fin (N + 1) → AlgebraicClosure E) (hv : v ≠ 0) (j : Fin (N + 1)) : (proj F N).ι (below (ptOf F E N v hv).1) ∈ chart F N j ↔ v j ≠ 0

theorem **exists_ptOf** (y : GeomPt (proj F N) E) : ∃ v hv, y = ptOf F E N v hv

theorem **ptOf_eq_iff** (v v' : Fin (N + 1) → AlgebraicClosure E) (hv : v ≠ 0) (hv' : v' ≠ 0) : ptOf F E N v hv = ptOf F E N v' hv' ↔ ∃ u : (AlgebraicClosure E)ˣ, v' = u • v

theorem **cutSet_unit** : cutSet (proj F N) E (crd F N) (unitCoef E N) = {origin F E N}

theorem **not_cutGood_zero** (bd : (proj F (N + 1)).Cl) : ¬ cutGood (proj F (N + 1)) E (crd F (N + 1)) 0 bd

theorem **exists_poly_quot** {F : Type} [Field F] (Y : Var F) {i : Fin (Y.N + 1)} (hi : Y.ι (genericPoint Y.X) ∈ chart F Y.N i) {x : Y.X} (hx : Y.ι x ∈ chart F Y.N i) (a : Y.X.presheaf.stalk x) : ∃ p q : MvPolynomial (Fin (Y.N + 1)) F, UnitAt x (aeval (Y.coordFn hi) q) ∧ algebraMap _ Y.Fn a ∗ aeval (Y.coordFn hi) q = aeval (Y.coordFn hi) p

theorem **tangent_eq_zero** {F : Type} [Field F] [Algebra F E] (Y : Var F) {i : Fin (Y.N + 1)} (hi : Y.ι (genericPoint Y.X) ∈ chart F Y.N i) (y : GeomPt Y E) (hx : Y.ι (below y.1) ∈ chart F Y.N i) (u : Fin (Y.N + 1) → Y.X.presheaf.stalk (below y.1)) (hu : ∀ l, algebraMap _ Y.Fn (u l) = Y.coordFn hi l) (δ : Y.X.presheaf.stalk (below y.1) →+ AlgebraicClosure E) (hδ : IsTangent Y E y δ) (h : ∀ l, δ (u l) = 0) : δ = 0

theorem **origin_mem_chart** (j : Fin (N + 1)) : (proj F N).ι (below (origin F E N).1) ∈ chart F N j ↔ j = 0

theorem **cutGood_unit** : cutGood (proj F N) E (crd F N) (unitCoef E N) ((proj F N).supp (projHyp F N))

theorem **not_cutGood_unit_top** : ¬ cutGood (proj F N) E (crd F N) (unitCoef E N) ⊤

theorem **cutHt_unit** {C : CxSt F} (A : Adl C (proj F N)) : cutHt (proj F N) E (crd F N) (unitCoef E N) A = A.ht (below (origin F E N).1)

theorem **one_le_cutOrd_iff** (j : Fin (N + 1)) : 1 ≤ cutOrd (proj F N) E (crd F N) (unitCoef E N) (projHyp F N) (crd F N j) ↔ j ≠ 0

theorem **exists_tangent** {F : Type} [Field F] [Algebra F E] (Y : Var F) (y : GeomPt Y E) (z : Sp (DualNumber (AlgebraicClosure E)) ⟶ Y.X) (hz : z ≫ Y.str = spMap ((algebraMap _ (DualNumber (AlgebraicClosure E))).comp ((algebraMap E (AlgebraicClosure E)).comp (algebraMap F E)))) (hy : spMap (TrivSqZeroExt.fstHom (AlgebraicClosure E) _ _).toRingHom ≫ z = y.1) (hne : spMap (algebraMap _ (DualNumber (AlgebraicClosure E))) ≫ y.1 ≠ z) : ∃ δ, IsTangent Y E y δ ∧ δ ≠ 0

theorem **exists_tangent_origin** : ∃ δ, IsTangent (proj F 1) E (origin F E 1) δ ∧ δ ≠ 0

theorem **not_cutGood_double** (bd : (proj F 1).Cl) : ¬ cutGood (proj F 1) E ![1, crd F 1 1, crd F 1 1 ^ 2] (fun _ => ![0, 0, 1]) bd

theorem **cutGood_empty** (hd : 1 ≤ Y.dim) (bd : Y.Cl) (Δ : Y.Dv) (f : Y.Fn) : cutSet Y E ![1] (fun _ _ => 1) = ∅ ∧ cutGood Y E ![1] (fun _ _ => 1) bd ∧ cutOrd Y E ![1] (fun _ _ => 1) Δ f = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsTiedOfIsRecG

theorem **isRecG_retie** {sc D Gt c H F} (h : IsRecG sc D Gt c H F) (g') : IsRecG sc D Gt c ⟨H.lev0, g'⟩ F

theorem **gatNum_of_isRecG** {sc D Gt c H F} (h : IsRecG sc D Gt c H F) : GatNum sc D

theorem **exists_tied_of_isRecG** {sc D Gt c H F} (h : IsRecG sc D Gt c H F) : ∃ g : GroupCompatibilityAll D Gt c, g.herm.lev0 = H.lev0 ∧ g.forms = F ∧ IsRecG sc D Gt c g.herm g.forms

theorem **hP_assemble_eq** (p) : hP (assemble B C W V S N G) p = hP (assemble B C W' V' S' N' G') p

theorem **lamRec_stages** (u g) : lamRec (assemble B C W V S N G) u g = lamRec (assemble B C W' V' S' N' G') u g

theorem **tauRec_stages** (w : B.places.Pl) [h : Fact (B.places.resChar w).Prime] (n) : @tauRec (assemble B C W V S N G) w h n = @tauRec (assemble B C W' V' S' N' G') w h n

theorem **omTied_stages** (δ : ℂ) : Formalism.L2.GlobalTheoryTheory.OmegaCompatible (assemble B C W V S N G) δ ↔ Formalism.L2.GlobalTheoryTheory.OmegaCompatible (assemble B C W' V' S' N' G) δ

theorem **above_auto** (w : ColW B f T) (v) : w.I.sel v ∈ aboveF (L

theorem **vals_auto** (w : ColW B f T) (a P) : ∃ z : B.LL, B.embLL z = (((w.I.xi a).val P : ℂˣ) : ℂ)

theorem **roots_auto** (w : ColW B f T) (ht : TC.IsTw w.I) (a) (P) (hP : P ∈ w.I.Q a) : ∃ z : B.LL, IsPrimitiveRoot z (Nat.card (𝒪 B.fields.E ⧸ P.asIdeal))

theorem **exists_over** (L : Type) [Field L] [NumberField L] {p : ℕ} (hp : p.Prime) : ∃ v : HeightOneSpectrum (𝒪 L), (p : 𝒪 L) ∈ v.asIdeal

theorem **under_mem** (w : ColW B f T) {p : ℕ} {v : HeightOneSpectrum (𝒪 B.host.F)} (hv : (p : 𝒪 B.host.F) ∈ v.asIdeal) : (p : 𝒪 B.host.K) ∈ ((show HeightOneSpectrum (𝒪 B.host.E) from w.I.sel v).under (𝒪 B.host.K)).asIdeal

theorem **pinned_iff** (w : ColW B f T) : Pinned K B f T w ↔ TC.IsTw w.I ∧ ColIn.JetsIn w.I ∧ Half B w.I.sel

theorem **gIs2_iff** (w : ColW B f T) (G : GJStage B w.C) : GIs2 K w G ↔ ∀ W V S N, Formalism.L2.Theta.VolTieG (w.D ⟨W, V, S, N, G⟩) ∧ Nonempty (ColumnDataSplitPrimes K w ⟨W, V, S, N, G⟩)

theorem **cond_eq_of_twisted** {w : ColW B f T} {Z : w.Lt} {Wt} {nch : NChoice (w.D Z)} (n : NchT w Z Wt nch) (v) (hv : (w.D Z).what₂.muAt v ≠ (w.D Z).what₂.mu0) : nch.Q v = T.Q v

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsUniqueIsCrd

theorem **sameOn_iff** {K V : Type} {n : ℕ} (j j' : (Fin n → K) → V) : SameOn j j' ↔ j = j'

theorem **exists_sameOn** {K V : Type} {n n' : ℕ} (h : n = n') (j' : (Fin n' → K) → V) : ∃ j : (Fin n → K) → V, SameOn j j'

theorem **range_lvl** : LinearMap.range ((MtildeKCSD W Ch sel mu (mtildeKOk W Ch sel mu hX)).lvlE (g0Act W Ch sel mu (mtildeKOk W Ch sel mu hX)) P.grp P.chr).mulVecLin = (MtildeKCSD W Ch sel mu (mtildeKOk W Ch sel mu hX)).lvlPart (g0Act W Ch sel mu (mtildeKOk W Ch sel mu hX)) P.grp P.chr

theorem **isFormOf_unique** {V : Type} {y : V} {x x' : MtildeV W Ch sel mu P} (h : IsFormOf W Ch sel mu hX P x y) (h' : IsFormOf W Ch sel mu hX P x' y) : x = x'

theorem **exists_isFormOf** (y) : ∃ x, IsFormOf W Ch sel mu hX P x (frm W Ch sel mu hX P y)

theorem **isFormOf_onto** (x : MtildeV W Ch sel mu P) : ∃ y, IsFormOf W Ch sel mu hX P x (frm W Ch sel mu hX P y)

theorem **tAll_ixOf** (U : Set I.ι) (s : R3.TupU I U) : (tAll I (R3.ixOf I U s)).1 = s.1

theorem **rDim_eq** (U : Set I.ι) (Xg tr) (s : R3.TupU I U) (P ψ) : rDim X.toD7 (R3.famR X I hX Set.univ Xg tr).A P ψ (tAll I (R3.ixOf I U s)) = rDim X.toD7 (R3.famR X I hX U Xg tr).A P ψ s

theorem **lDim_eq** (U : Set I.ι) (Xg tr) (s : R3.TupU I U) (P ψ) (i) : lDim X.toD7 (R3.famR X I hX Set.univ Xg tr).A P ψ i (tAll I (R3.ixOf I U s)) = lDim X.toD7 (R3.famR X I hX U Xg tr).A P ψ i s

theorem **rowForm_eq** (Xg tr) (a : I.ι) (y) : R3.rowForm (D

theorem **existsUnique_isCrd** (Xg tr) (a : I.ι) {M : Type} [AddCommGroup M] [Module ℂ M] (e : Df.MtCx I a ≃ₗ[ℂ] M) (y) : ∃! v, IsCrd I hX a e (R3.rowForm (D

theorem **isCrd_determines** (Xg tr) (a : I.ι) {M : Type} [AddCommGroup M] [Module ℂ M] (e e' : Df.MtCx I a ≃ₗ[ℂ] M) (j : _ → M) (h : ∀ y, IsCrd I hX a e (R3.rowForm (D

theorem **lawsM** (t : RowsT K w Z Wt p0) : Formalism.L2.Pj.LawsM t.r.z.core.toCore

theorem **rank_inR** (t : RowsT K w Z Wt p0) (j : ℕ) (hj : Formalism.L2.Pj.InR (w.st Z).toD7 j) : (t.r.z.core.Q.Dw j).p = (B.R.sigW j).1 ∧ (t.r.z.core.Q.Dw j).q = (B.R.sigW j).2

theorem **rx_io** {t : RowsT K w Z Wt p0} (x : RXs t) (μ) (h : ¬ Formalism.IndefR.RecTie.Used0 (w.D Z) μ) : p0.s.iota μ = 0

theorem **isWtOfPar_unique** {N N' : ℕ} {S : SPar} (hS : ∀ x ∈ S, x.2 = true) {lam : DomWeight N} {lam' : DomWeight N'} (h : Bridge.IsWtOfPar lam S) (h' : Bridge.IsWtOfPar lam' S) (i : Fin N) (i' : Fin N') (e : (i : ℕ) = i') : lam.w i = lam'.w i'

theorem **isWtOfPar_sgTrue** (D : D7) (τ : D.what₂.Tau) : Bridge.IsWtOfPar (RowPar.of (D.hc τ)).sgTrue ((Ptw D).Dpar D.R.m τ)

theorem **lamRdef_eq** (X : Stages) (τ) (i : Fin X.toD7.sp.X'.rank) (i' : Fin (X.B.R.N X.B.R.m − 1)) (e : (i : ℕ) = i') : (Bridge.lamRdef (Ptw X.toD7) τ).w i = (X.B.wts.sgTrue τ).w i'

theorem **labLL_eq** (X : Stages) : labLL X.toD7 = X.B.fields.emb

theorem **exists_matEq** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (hG : V.gram = 1) (h : Bridge.Gf V) : ∃ x : UAd D.sp.cm n, R3.MatEq (R3.matIn V 1 h) (x : GL (Fin n) _)

theorem **gram_X'** {B : Base} {f : Fr B.toPre} {T : TwDat B} (w : ColW B f T) : B.sp.X'.gram = 1

theorem **rx_std_read** {K : Ctx} {B : Base} {f : Fr B.toPre} {T : TwDat B} {w : ColW B f T} {Z : w.Lt} {Wt : Set B.places.Pl} {p0 : BaseLevelRowSide (w.D Z)} {t : RowsT K w Z Wt p0} (x : RXs t) : ∃ c : VV B.LL B.wts.sgTrue → t.r.z.R.cX.W, ∀ (h : Bridge.Gf (w.st Z).toD7.sp.X'), ∃ y : UAd B.sp.cm (B.R.N B.R.m − 1), ∀ (s : R3.TupU w.I (R3.usedOf w.I T Wt)) (φ : (mtildeKGSD (Pr

theorem **lamLdef_eq** (X : Stages) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ X.B.R.m) (τ) (i : Fin (X.toD7.sp.V j).rank) (i' : Fin X.B.R.nV) (e : (i : ℕ) = i') : (Bridge.lamLdef (D

theorem **gram_V** {B : Base} {f : Fr B.toPre} {T : TwDat B} (w : ColW B f T) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ B.R.m) : (B.sp.V j).gram = 1

theorem **rank_V** {B : Base} {f : Fr B.toPre} {T : TwDat B} (w : ColW B f T) (j : ℕ) : (B.sp.V j).rank = B.R.nV

theorem **rx_std_readL** {K : Ctx} {B : Base} {f : Fr B.toPre} {T : TwDat B} {w : ColW B f T} {Z : w.Lt} {Wt : Set B.places.Pl} {p0 : BaseLevelRowSide (w.D Z)} {t : RowsT K w Z Wt p0} (x : RXs t) (i : Fin (w.st Z).toD7.R.m) : ∃ c : VV B.LL (B.wts.lamV ((i : ℕ) + 1)) → (t.r.z.R.cV i).W, ∀ (h : Bridge.Gf (Vc (w.st Z).toD7 i)), ∃ y : UAd B.sp.cm B.R.nV, ∀ (s : R3.TupU w.I (R3.usedOf w.I T Wt)) (φ : (mtKGSD (Pr

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsUniqueIsVal

theorem **nonempty_grTriv** (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) {L : LB Z.X} (h : IsGrLB Z e L) : Nonempty (GrTriv Z e (Z.ofMod (GlobalTheory.LB.toMod L)).toMod)

theorem **range_subset_opn** {x : Sp k ⟶ Z.X} {u : Vecs Dw (Y Z.i).t k} (h : Z.FlagAt x u) : Set.range x ⊆ Z.opn

theorem **GrTriv.existsUnique_isVal** (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) (f : Γ(F, V)) {x : Sp k ⟶ Z.X} (hx : ⊤ ≤ x ⁻¹ᵁ V) {u : Vecs Dw (Y Z.i).t k} (hu : Z.FlagAt x u) : ∃! r, T.IsVal f x u r

theorem **GrTriv.IsVal.change** {T : GrTriv Z e F} {f : Γ(F, V)} {x : Sp k ⟶ Z.X} {u u' : Vecs Dw (Y Z.i).t k} {r : k} (h : T.IsVal f x u r) (tu : ∀ σ i, IsTanAtG (Y Z.i).t (x ≫ Z.π) (u σ i)) (tu' : ∀ σ i, IsTanAtG (Y Z.i).t (x ≫ Z.π) (u' σ i)) {d : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → k} (hd : ∀ σ, DiagAt (Y Z.i).t (x ≫ Z.π) (u σ) (u' σ) (d σ)) : T.IsVal f x u' (r ∗ ∏ σ, ∏ a, d σ a ^ e σ a)

theorem **GrTriv.isVal_chart** (f : Γ(F, V)) {x : Sp k ⟶ Z.X} (hx : ⊤ ≤ x ⁻¹ᵁ V) (i : T.L.ι) (φ : Γ(Z.X, T.L.U i) →+* k) (h : spMap φ ≫ (T.hU i).fromSpec = x) : T.IsVal f x ((T.w i).bc φ) (valR x (V ⊓ T.L.U i) (GlobalTheory.LB.coord (T.iso.hom.app V f) i))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsUniqueTransUnit

theorem **feRel_iff** {mu α : D.X} {s : SecFMod D (FractionRing R) 1 α} {t : SecFMod D (FractionRing R) 1 (α + 1 • mu)} : FERel D mu α s t ↔ ∀ y J, Good D.P.L y J → sval t (𝟙 _) y J = sval s (𝟙 _ ≫ (DegPre.trF D (FractionRing R) mu).left) y J ∗ (uVal D (FractionRing R) 1 α (cP D _ mu) (𝟙 _) y J ∗ val (DegPre.famMult D (FractionRing R) 1 mu α) (𝟙 _ ≫ fst (TF (R

theorem **FERel.unique** {mu α : D.X} {s : SecFMod D (FractionRing R) 1 α} {t t' : SecFMod D (FractionRing R) 1 (α + 1 • mu)} (h : FERel D mu α s t) (h' : FERel D mu α s t') : t = t'

theorem **FERel.smul** {mu α : D.X} {s : SecFMod D (FractionRing R) 1 α} {t : SecFMod D (FractionRing R) 1 (α + 1 • mu)} (h : FERel D mu α s t) (c : FractionRing R) : FERel D mu α (c • s) (c • t)

theorem **exists_psi_val** [IsLocalRing R] (D : DegPre R) (χ : D.X) {a₀ : R} (h : PsiCoord D χ a₀) : ∃ (e : LB.Gen (LB.LinComb D.P.L (K

theorem **trB_comp** (a b : D.X) (h : a + b = 0) : (trB D a).left ≫ (trB D b).left = 𝟙 _

theorem **exists_equiv** (mu α : D.X) (eB : LB.Gen (LB.LinComb D.P.L (Sum.elim (D.famφB (α + 1 • mu)) (fun i => (trB D mu).left ≫ D.famφB α i)) (DegPre.deltaε 1))) (heB : eB.IsGen) : ∃ θ : SecBMod D 1 α ≃ₗ[R] SecBMod D 1 (α + 1 • mu), ∀ s (Z : Scheme.{0}) (ρ : Z ⟶ D.P.A.left) (z : Z) (J), Good D.P.L z J → sval (θ s) ρ z J = sval s (ρ ≫ (trB D mu).left) z J ∗ (val eB ρ z J).val

theorem **constVal_mul** {S : Type} [CommRing S] {Z : Scheme.{0}} (q : Z ⟶ Spec (CommRingCat.of S)) (z : Z) (a b : S) : constVal q z (a * b) = constVal q z a * constVal q z b

theorem **unit_calc** {S : Type} [CommRing S] {P₁ P₂ E₁ E₂ : Sˣ} {κ c₁ c₂ : S} (p₁ : P₁.val = c₁ * E₁.val) (p₂ : P₂.val = c₂ * E₂.val) (k : κ * c₂ = c₁) : κ * (E₁ * E₂⁻¹).val = (P₁ * P₂⁻¹).val

theorem **exists_transl** [IsLocalRing R] (Dg : DegData R (maximalIdeal R)) (mu α : Dg.toPre.X) : ∃ (θ : SecBMod Dg.toPre 1 α ≃ₗ[R] SecBMod Dg.toPre 1 (α + 1 • mu)) (κ : (FractionRing R)ˣ), (κ : Cls R) = psiCls Dg (mu + α) * (psiCls Dg α)⁻¹ ∧ ∀ s, FERel Dg.toPre mu α (resF Dg.toPre (FractionRing R) 1 α s) ((κ : FractionRing R) • resF Dg.toPre (FractionRing R) 1 _ (θ s))

theorem **exists_secGen** (χ : Dg.toPre.X) : ∃ e : SecBMod Dg.toPre 1 χ, ∀ s, ∃! a : R, s = a • e

theorem **secGen_spec** (χ : Dg.toPre.X) (s : SecBMod Dg.toPre 1 χ) : ∃! a : R, s = a • secGen Dg χ

theorem **secGenK_ne** (χ : Dg.toPre.X) : secGenK Dg χ ≠ 0

theorem **resF_smul_secGen** (χ : Dg.toPre.X) (a : R) : resF Dg.toPre (FractionRing R) 1 χ (a • secGen Dg χ) = algebraMap R (FractionRing R) a • secGenK Dg χ

theorem **existsUnique_transUnit** [IsIntegrallyClosed R] {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] (ρ : Dg.toPre.OAct O) {t} (ht : IsReading Dg O t) {w : O} (hw : w + star w = 1) (mu α : Dg.toPre.X) : ∃! u, IsTransUnit t w mu α u

theorem **expands_iff** (s : SecGK U) (F : Fam D.toPre (FractionRing R) 1) : U.Expands 1 s F ↔ F = expandFun U s

theorem **expand_bijective** (hR : AdmBase R I) [GeometricallyConnected G.hom] : Function.Bijective (expand U hR)

theorem **existsUnique_coef** (s : SecGK U) (χ : D.toPre.X) : ∃! x : FractionRing R, famComp D.toPre (FractionRing R) 1 χ (expand U hR s) = x • secGenK D χ

theorem **coef_spec** (s : SecGK U) (χ : D.toPre.X) : famComp D.toPre (FractionRing R) 1 χ (expand U hR s) = coef U hR s χ • secGenK D χ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsUnitOfNorm

theorem **psiCoord_iff** (χ : D.X) (r : R) : PsiCoord D χ r ↔ ∃ i, CoordAt D χ i r

theorem **gen_mem** {U : (Sp V).Opens} {y : Sp V} (h : y ∈ U) : gen V ∈ U

theorem **jK_mem** {U : (Sp R).Opens} {y : Sp R} (h : y ∈ U) : jK R (gen (FractionRing R)) ∈ U

theorem **gm_inj** {U : (Sp V).Opens} (h : gen V ∈ U) {a b : Γ(Sp V, U)ˣ} (e : gm (𝟙 (Sp V)) (gen V) U a = gm (𝟙 (Sp V)) (gen V) U b) : a = b

theorem **gmK_inj** {U : (Sp R).Opens} (h : jK R (gen (FractionRing R)) ∈ U) {a b : Γ(Sp R, U)ˣ} (e : gm (jK R) (gen _) U a = gm (jK R) (gen _) U b) : a = b

theorem **Rel.mul** {x x' y y'} (h : Rel g x y) (h' : Rel g x' y') : Rel g (x * x') (y * y')

theorem **Rel.inv** {x y} (h : Rel g x y) : Rel g x⁻¹ y⁻¹

theorem **Rel.congr** {x x' y} (h : Rel g x y) (e : x = x') : Rel g x' y

theorem **Rel.unique** {x y y'} (h : Rel g x y) (h' : Rel g x y') : y = y'

theorem **Rel.point** {X : Scheme.{0}} (p : Sp R ⟶ X) (U : X.Opens) (h : p (spMap g (gen V)) ∈ U) (a : Γ(X, U)ˣ) : Rel g (gm (jK R ≫ p) (gen _) U a) (gm (spMap g ≫ p) (gen V) U a)

theorem **toStk_injective** : Function.Injective (toStk V)

theorem **toStk_map** (r : R) : toStk V (g r) = gs (spMap g) (gen V) ⊤ (DegPre.toOpen (R

theorem **toStk_valR** {U : (Sp V).Opens} (h : ⊤ ≤ U) (a : Γ(Sp V, U)) : toStk V (valR (𝟙 (Sp V)) U a) = gs (𝟙 (Sp V)) (gen V) U a

theorem **isUnit_toStk** {v : V} (hv : v ≠ 0) : IsUnit (toStk V v)

theorem **Rel.elt** {r : R} (hr : g r ≠ 0) {x y} (hx : x.val = gs (jK R) (gen _) ⊤ (DegPre.toOpen (R

theorem **corr_of_val_gen** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Sp V ⟶ X)} {ε : K → ℤ} {φ' : K' → (Sp V ⟶ X)} {ε' : K' → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' : LB.Gen (LB.LinComb L φ' ε')} (h : ∀ J, Good L (gen V) J → val t (𝟙 _) (gen V) J = val t' (𝟙 _) (gen V) J) : LB.Gen.Corr L φ ε φ' ε' t t'

theorem **exists_gen_of_top** {X Y : Scheme.{0}} {L : LB X} (hL : L.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (j₀ : K → L.ι) (h : ⊤ ≤ LB.wchart L φ j₀) (w : Γ(Y, ⊤)ˣ) : ∃ e : LB.Gen (LB.LinComb L φ ε), e.IsGen ∧ ∀ y j, y ∈ LB.wchart L φ j → gm (𝟙 Y) y (LB.wchart L φ j) (e.s j) = trF φ ε (𝟙 Y) y j j₀ * gm (𝟙 Y) y ⊤ w

theorem **cT_left** (χ : D.X) : (cT D (Over.mk (spMap g)) χ).left = spMap g ≫ (D.c χ).left

theorem **xK_eq** (χ : D.X) : D.xK χ = jK R ≫ (D.c χ).left

theorem **zero_left** : (toUnit (Over.mk (spMap g)) ≫ η[D.P.A]).left = spMap g ≫ eSec D.P.A

theorem **rel_psi** {χ : D.X} {i i₀ : D.P.L.ι} {r : R} (hi : (D.c χ).left (spMap g (gen V)) ∈ D.P.L.U i) (h : CoordAt D χ i₀ r) (hr : g r ≠ 0) {y} (hy : y.val = toStk V (g r)) : Rel g (gm (𝟙 _) (gen _) (LB.wchart D.P.L (K

theorem **rigV_val** {i : D.P.L.ι} (h : ⊤ ≤ eSec D.P.A ⁻¹ᵁ D.P.L.U i) : (rigV D g i).val = toStk V (g (rigAt D i))

theorem **isUnit_rigAt** {i : D.P.L.ι} (h : ⊤ ≤ eSec D.P.A ⁻¹ᵁ D.P.L.U i) : IsUnit (rigAt D i)

theorem **rigV_trans** {i i' : D.P.L.ι} (h : eSec D.P.A (spMap g (gen V)) ∈ D.P.L.U i) (h' : eSec D.P.A (spMap g (gen V)) ∈ D.P.L.U i') : rigV D g i = trV D (spMap g ≫ eSec D.P.A) i i' * rigV D g i'

theorem **exists_goodK** {J} (hJ : Good D.P.L (gen V) J) : ∃ JK, Good D.P.L (gen (FractionRing R)) JK ∧ ∀ x : Sp R ⟶ D.P.A.left, x ≫ D.P.A.hom = 𝟙 _ → JK (jK R ≫ x) = J (spMap g ≫ x)

theorem **rel_val** {J JK} (hJ : Good D.P.L (gen V) J) (hK : ∀ x : Sp R ⟶ D.P.A.left, x ≫ D.P.A.hom = 𝟙 _ → JK (jK R ≫ x) = J (spMap g ≫ x)) {Y : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ D.P.A.left)} {ε : K → ℤ} (t : LB.Gen (LB.LinComb D.P.L φ ε)) (p : Sp R ⟶ Y) (hp : ∀ k, (p ≫ φ k) ≫ D.P.A.hom = 𝟙 _) {q q'} (hq : q = jK R ≫ p) (hq' : q' = spMap g ≫ p) : Rel g (val t q (gen _) JK) (val t q' (gen V) J)

theorem **exists_unit_of_norm** {x : C} (h : ‖x‖ = 1) : ∃ u : (intO C)ˣ, ((u : intO C) : C) = x

theorem **exists_isBiTriv** {O : Type} [Ring O] [StarRing O] [Module O D.X] (oa : D.OAct O) (hD : D.TauBimult) (g : R →+* intO C) (hex : ∀ χ, ∃ r, PsiCoord D χ r) (hne : ∀ χ r, PsiCoord D χ r → g r ≠ 0) (t : D.X → D.X → Cˣ) (hs : ∀ a b, t a b = t b a) (hm : ∀ a a' b, t (a + a') b = t a b * t a' b) (hb : ∀ (x : O) a b, t (x • a) b = t a (star x • b)) (hn : ∀ a b r₁ r₂ r₃, PsiCoord D (a + b) r₁ → PsiCoord D a r₂ → PsiCoord D b r₃ → ‖(g r₁ : C)‖ = ‖(t a b : C)‖ * ‖(g r₂ : C)‖ * ‖(g r₃ : C)‖) : ∃ e : ∀ a b, BiGen D (Over.mk (spMap g)) a b, IsBiTriv oa e ∧ ∀ a b, IsTauVal g (e a b) (t a b)

theorem **IsTauVal.unique** {g : R →+* intO C} (hex : ∀ χ, ∃ r, PsiCoord D χ r) (hne : ∀ χ r, PsiCoord D χ r → g r ≠ 0) {a b : D.X} {e : BiGen D (Over.mk (spMap g)) a b} {t t' : C} (h : IsTauVal g e t) (h' : IsTauVal g e t') : t = t'

theorem **exists_lift** {X : Type} [AddCommGroup X] [Module.Finite ℤ X] (β : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) (p : ℕ) (h : ∀ a b, ∃ c : Cˣ, β a b = -Real.logb p ‖(c : C)‖) : ∃ t : X → X → Cˣ, (∀ a b, β a b = -Real.logb p ‖(t a b : C)‖) ∧ (∀ a b a' b', β a b = β a' b' → t a b = t a' b') ∧ ∀ a a' b, t (a + a') b = t a b * t a' b

theorem **exists_coord** (A : DatAt Y P) (χ : A.Ylat) : ∃ r, PsiCoord A.U.D.toPre χ r

theorem **coord_ne** (A : DatAt Y P) (χ : A.Ylat) (r : A.R) (h : PsiCoord A.U.D.toPre χ r) : A.f r ≠ 0

theorem **beta_coord** (A : DatAt Y P) {a b : A.Ylat} {r₁ r₂ r₃ : A.R} (h₁ : PsiCoord A.U.D.toPre (a + b) r₁) (h₂ : PsiCoord A.U.D.toPre a r₂) (h₃ : PsiCoord A.U.D.toPre b r₃) : A.beta a b = -Real.logb p ‖(A.f r₁ : ↥C.S) / (A.f r₂ * A.f r₃)‖

theorem **nonempty_reading** (hp : 1 < p) (A : DatAt Y P) : Nonempty (Reading A)

theorem **Reading.eq_of_isTauVal** {A : DatAt Y P} (ρ : Reading A) {a b : A.Ylat} {s : ↥C.S} (h : IsTauVal A.f (ρ.e a b) s) : s = ρ.t a b

theorem **Reading.add_right** {A : DatAt Y P} (ρ : Reading A) (a b b' : A.Ylat) : ρ.t a (b + b') = ρ.t a b * ρ.t a b'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsVResEqZero

theorem **vDeg_symm** (d : ℕ) (D D' : Z.VDivZ) (Q : PicSym Z.X) : Z.vDeg d D (firstChernPic (Z.vCls D') * Q) = Z.vDeg d D' (firstChernPic (Z.vCls D) * Q)

theorem **capDeg_vp_symm** (p d : ℕ) (D D' : Z.VDivQ) (Q : PicSym Z.X) : Z.capDeg_vp p d D (Z.cDQ D' * Q) = Z.capDeg_vp p d D' (Z.cDQ D * Q)

theorem **isLB_compLB** : (compLB Z).IsLB

theorem **vRes_compPair** (g : Z.Comp → O) (hg : ∀ C, g C ≠ 0) (C : Z.Comp) : Z.vRes C (compPair g hg).cls = (Z.comp C).div (g C) (hg C)

theorem **exists_vRes_eq_zero** {C C' : Z.Comp} (h : C ≠ C') : ∃ D : Z.VDivZ, Z.vRes C D ≠ 0 ∧ Z.vRes C' D = 0

theorem **capDeg_divP_cD** (d : ℕ) (D : (Z.toITh p d).VDiv) (Q : (Z.toITh p d).Ch) : (Z.toITh p d).capDeg (Z.toITh p d).divP ((Z.toITh p d).cD D * Q) = 0

theorem **ck_copies** (n : ℕ) : Formalism.BasicNotions.DV.ck (smulPair (Z

theorem **capDeg_empty** [IsEmpty Z.X] (d : ℕ) (D : (Z.toITh p d).VDiv) (Q : (Z.toITh p d).Ch) : (Z.toITh p d).capDeg D Q = 0

theorem **capDeg_mixed** (i : Fin 2) (x : PicSym (mixed O).X) : ((mixed O).toITh p (i + 1)).capDeg ((mixed O).toITh p (i + 1)).divP x = ((Model.projSpace O (i + 1)).toITh p).capDeg ((Model.projSpace O (i + 1)).toITh p).divP (picSymPull (Sigma.ι (fun i : Fin 2 => (Model.projSpace O (i + 1)).X) i) x)

theorem **capDeg_ithOf_symm** (D D' : (ithOf hp hW M d).VDiv) (Q : (ithOf hp hW M d).Ch) : (ithOf hp hW M d).capDeg D ((ithOf hp hW M d).cD D' * Q) = (ithOf hp hW M d).capDeg D' ((ithOf hp hW M d).cD D * Q)

theorem **divisorOfPrime_ne_zero** [Nonempty M.X] : haveI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExistsXdAllIff

theorem **AtMem.imp** {p q : LayP hL} (h : AtMem hL p) (hpq : ∀ K B f T w Z Wt p0 t nch n c, p K B f T w Z Wt p0 t nch n c → q K B f T w Z Wt p0 t nch n c) : AtMem hL q

theorem **AtMem.and** {p q : LayP hL} (h : AtMem hL p) (h' : AtMem hL q) : AtMem hL fun K B f T w Z Wt p0 t nch n c => p K B f T w Z Wt p0 t nch n c ∧ q K B f T w Z Wt p0 t nch n c

theorem **rootsLL_pow** {D : D7} (h : RootsLL D) : ∃ ζ : D.what₂.LL, ζ ^ D.what₂.restRowsFiniteGroupCard = 1 ∧ orderOf ζ = D.what₂.restRowsFiniteGroupCard ∧ 0 < orderOf ζ

theorem **sumUsed_ref** {D : D7} {M : MtFam D} (h : SumUsed D M) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : GlobalTheory.CoefDuty.ClassSum (M D.what₂.mu0 j h1 h2)

theorem **used_loc** {D : D7} (i : D.nd.Loc) : Formalism.IndefR.RecTie.Used0 D (D.nd.mu i)

theorem **sumUsed_loc** {D : D7} {M : MtFam D} (h : SumUsed D M) (i : D.nd.Loc) : GlobalTheory.CoefDuty.ClassSum (M.loc i)

theorem **exists_xdAll_iff** (D : D7) : (∃ xd : ThData D, XdAll xd) ↔ ∀ i j b, 1 ≤ j → j ≤ D.R.m → ∀ (a : D.nexp.Dat i j b) (u : D.nexp.U), D.nexp.actG i j b u (D.nexp.th i j b a) = ((D.nexp.psiU i b u : (FractionRing (D.ntw.O i))ˣ) : FractionRing (D.ntw.O i)) • D.nexp.th i j b a

theorem **isot_withLoc** {D : D7} (L : LocAll (lay0 D)) : Rows.IsotInternal (withLoc L).what₂ ↔ Rows.IsotInternal D.what₂

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExpoConjOfIsCsd

theorem **unip_add** {R : Type} [CommRing R] {r : ℕ} (x y : BlockIx a r → R) : unip (x + y) = unip x * unip y

theorem **det_unip** {R : Type} [CommRing R] {r : ℕ} (x : BlockIx a r → R) : GeneralLinearGroup.det (unip x) = 1

theorem **unip_algebraMap** {r : ℕ} (x : BlockIx a r → E) : unip (fun p => algebraMap E (Ad E) (x p)) = ratGL E a (unip x)

theorem **act_mulChi** (χ : HChar E) (o : Op E a) (φ : Sm E a) : act o (mulChi a χ φ) = mulChi a χ (((opMul χ o : ℂˣ) : ℂ) • act o φ + opAdd χ o • φ)

theorem **isOpen_ker_chiDet** (χ : HChar E) : IsOpen {k : GL (Fin a) (Af E) | chiDet a χ (finGL E a k) = 1}

theorem **chiDet_ratGL** (χ : HChar E) (γ : GL (Fin a) E) : chiDet a χ (ratGL E a γ) = 1

theorem **one_le_height** (y : GL (Fin a) (Cinf E)) : 1 ≤ height y

theorem **norm_det_le** (x : Matrix (Fin a) (Fin a) (Cinf E)) {B : ℝ} (hx : ‖toL x‖ ≤ B) : ‖x.det‖ ≤ a.factorial * B ^ a

theorem **chiDet_growth** (χ : HChar E) : ∃ (C : ℝ) (N : ℕ), ∀ y, ‖((chiDet a χ (infGL E a y) : ℂˣ) : ℂ)‖ ≤ C * height y ^ N

theorem **Zalg_mulChi** (χ : HChar E) {z : Module.End ℂ (Sm E a)} (hz : z ∈ Zalg E a) : ∃ z' ∈ Zalg E a, ∀ φ, z (mulChi a χ φ) = mulChi a χ (z' φ)

theorem **IsAutForm.mulChi** {φ : Sm E a} (h : IsAutForm φ) (χ : HChar E) : IsAutForm (mulChi a χ φ) where

theorem **IsCuspForm.mulChi** {φ : Sm E a} (h : IsCuspForm φ) (χ : HChar E) : IsCuspForm (mulChi a χ φ)

@[simp] theorem **op_val** (o : Op E a) (φ : π.V) : (π.op o φ).1 = act o φ.1

theorem **op_toTwist** (χ : HChar E) (o : Op E a) (φ : π.V) : (π.twist χ).op o (π.toTwist χ φ) = π.toTwist χ (((opMul χ o : ℂˣ) : ℂ) • π.op o φ + opAdd χ o • φ)

theorem **Iso.twist** (h : Iso π π') (χ : HChar E) : Iso (π.twist χ) (π'.twist χ)

theorem **twist_one** : π.twist 1 = π

theorem **twist_mul** (χ χ' : HChar E) : π.twist (χ * χ') = (π.twist χ).twist χ'

theorem **HasInfChar.unique** {τ : E →+* ℂ} {m m' : Multiset ℂ} (h : π.HasInfChar τ m) (h' : π.HasInfChar τ m') : m = m'

theorem **iso_cap** (T : π.V ≃ₗ[ℂ] π'.V) (hT : ∀ o φ, T (π.op o φ) = π'.op o (T φ)) (τ : E →+* ℂ) (u : ℂ) (φ : π.V) : ∃ h : cap τ u φ.1 ∈ π.V, (T ⟨_, h⟩).1 = cap τ u (T φ).1

theorem **Iso.hasInfChar** (h : Iso π π') {τ : E →+* ℂ} {m : Multiset ℂ} (hm : π.HasInfChar τ m) : π'.HasInfChar τ m

theorem **HasInfChar.twist** {τ : E →+* ℂ} {m : Multiset ℂ} (h : π.HasInfChar τ m) (χ : HChar E) : (π.twist χ).HasInfChar τ (m.map (· + expo χ τ))

theorem **continuous_sLoc** (w : HeightOneSpectrum (𝒪 E)) : Continuous (sLoc E w)

theorem **continuous_locGL** (w : HeightOneSpectrum (𝒪 E)) : Continuous (locGL E a w)

theorem **locRep_eq** (w : HeightOneSpectrum (𝒪 E)) (g : GL (Fin a) (w.adicCompletion E)) : π.locRep w g = π.op (.fin (locGL E a w g))

theorem **isSmooth_finPart** : GlobalTheory.PRep.IsSmooth π.finPart

theorem **isSmooth_locRep** (w : HeightOneSpectrum (𝒪 E)) : GlobalTheory.PRep.IsSmooth (π.locRep w)

theorem **IsLocComp.congr** {w : HeightOneSpectrum (𝒪 E)} {σ σ' : SmIrr (GL (Fin a) (w.adicCompletion E))} (h : π.IsLocComp w σ) (hπ : Iso π π') (hσ : Nonempty (σ.ρ.Equiv σ'.ρ)) : π'.IsLocComp w σ'

theorem **chiDet_locGL** (χ : HChar E) (w : HeightOneSpectrum (𝒪 E)) (g : GL (Fin a) (w.adicCompletion E)) : chiDet a χ (finGL E a (locGL E a w g)) = locDet a χ w g

theorem **isOpen_ker_locDet** (χ : HChar E) (w : HeightOneSpectrum (𝒪 E)) : IsOpen {g | locDet a χ w g = 1}

theorem **IsLocComp.twist** {w : HeightOneSpectrum (𝒪 E)} {σ : SmIrr (GL (Fin a) (w.adicCompletion E))} (h : π.IsLocComp w σ) (χ : HChar E) : (π.twist χ).IsLocComp w (σ.twist (locDet a χ w) (isOpen_ker_locDet χ w))

theorem **toUnits_conj_PhiK** (k : Kinf E a) : Unitary.toUnits (PhiK E a * k * (PhiK E a)⁻¹) = thetaGL a (starRingEnd (Cinf E)) (Unitary.toUnits k)

theorem **Iso.isCsd** {π π' : CuspSub L a} (h : Iso π π') (hπ : π.IsCsd) : π'.IsCsd

theorem **CuspSub.IsCsd.twist** {π : CuspSub L a} (h : π.IsCsd) {χ : HChar L} (hχ : ThetaInv a χ) : (π.twist χ).IsCsd

theorem **oneFn_ne_zero** : oneFn E a ≠ 0

theorem **lie_oneFn** (X : Lie E a) : lie E a X (oneFn E a) = 0

theorem **act_oneFn** (o : Op E a) : act o (oneFn E a) = opTriv o • oneFn E a

theorem **isStable_const** : IsStable (Submodule.span ℂ {oneFn E a})

theorem **isAutForm_const** {φ : Sm E a} (hφ : φ ∈ Submodule.span ℂ {oneFn E a}) : IsAutForm φ

theorem **cap_oneFn** (τ : E →+* ℂ) (u : ℂ) : cap τ u (oneFn E a) = (∏ k : Fin a, (u + (((a : ℂ) - 1) / 2 - (k : ℕ)))) • oneFn E a

theorem **CuspSub.op_triv_twist** (ha : a ≤ 1) (χ : HChar E) (o : Op E a) (φ : ((CuspSub.triv E ha).twist χ).V) : ((CuspSub.triv E ha).twist χ).op o φ = (((opMul χ o : ℂˣ) : ℂ) * opTriv o + opAdd χ o) • φ

@[simp] theorem **twist_mk** (π : CuspSub E a) (χ : HChar E) : (mk π).twist χ = mk (π.twist χ)

@[simp] theorem **twist_one** (π : CG E a) : π.twist 1 = π

theorem **twist_mul** (π : CG E a) (χ χ' : HChar E) : π.twist (χ * χ') = (π.twist χ).twist χ'

theorem **HasInfChar.infChar_eq** {π : CG E a} {τ : E →+* ℂ} {m : Multiset ℂ} (h : π.HasInfChar τ m) : π.infChar τ = m

theorem **card_infChar** (π : CG E a) (τ : E →+* ℂ) : Multiset.card (π.infChar τ) = a

theorem **HasInfChar.twist** {π : CG E a} {τ : E →+* ℂ} {m : Multiset ℂ} (h : π.HasInfChar τ m) (χ : HChar E) : (π.twist χ).HasInfChar τ (m.map (· + expo χ τ))

theorem **isRegAlg_iff** (π : CG E a) : π.IsRegAlg ↔ ∀ τ, (π.infChar τ).Nodup ∧ ∀ z ∈ π.infChar τ, ∃ n : ℤ, z = (n : ℂ) + ((a : ℂ) - 1) / 2

theorem **IsLoc.loc** {π : CG E a} {w : HeightOneSpectrum (𝒪 E)} {σ : Irr (GL (Fin a) (w.adicCompletion E))} (h : π.IsLoc w σ) : π.IsLoc w (π.loc w)

theorem **IsLoc.twist** {π : CG E a} {w : HeightOneSpectrum (𝒪 E)} {σ : Irr (GL (Fin a) (w.adicCompletion E))} (h : π.IsLoc w σ) (χ : HChar E) : (π.twist χ).IsLoc w (σ.twist (CuspSub.locDet a χ w) (CuspSub.isOpen_ker_locDet χ w))

theorem **twist_ofChar** (χ χ' : HChar E) : (ofChar χ).twist χ' = ofChar (χ * χ')

theorem **det_oneByOne** {R : Type} [CommRing R] (u : Rˣ) : GeneralLinearGroup.det (oneByOne u) = u

theorem **scalar_eq** {χ χ' : HChar E} {o o' : Op E 1} (T : ((CuspSub.triv E le_rfl).twist χ).V ≃ₗ[ℂ] ((CuspSub.triv E le_rfl).twist χ').V) (h : ∀ φ, T (CuspSub.op _ o φ) = CuspSub.op _ o' (T φ)) : ((opMul χ o : ℂˣ) : ℂ) * opTriv o + opAdd χ o = ((opMul χ' o' : ℂˣ) : ℂ) * opTriv o' + opAdd χ' o'

theorem **ofChar_injective** : Function.Injective (ofChar : HChar E → CG E 1)

@[simp] theorem **toChar_ofChar** (χ : HChar E) : toChar (ofChar χ) = χ

@[simp] theorem **tw_ofChar** (π : CG E a) (χ : HChar E) : π.tw (ofChar χ) = π.twist χ

theorem **hasInfChar_ofChar** (χ : HChar E) (τ : E →+* ℂ) : (ofChar χ).HasInfChar τ {expo χ τ}

theorem **infChar_ofChar** (χ : HChar E) (τ : E →+* ℂ) : (ofChar χ).infChar τ = {expo χ τ}

theorem **isRegAlg_ofChar** (χ : HChar E) : (ofChar χ).IsRegAlg ↔ ∀ τ, ∃ n : ℤ, expo χ τ = n

theorem **isLoc_ofChar** (χ : HChar E) (w : HeightOneSpectrum (𝒪 E)) : (ofChar χ).IsLoc w (Irr.ofChar (CuspSub.locDet 1 χ w) (CuspSub.isOpen_ker_locDet χ w))

theorem **isCsd_ofChar** (χ : HChar L) : (ofChar χ).IsCsd ↔ ThetaInv 1 χ

theorem **trace_theta** {R : Type} [CommRing R] [StarRing R] (X : Matrix (Fin a) (Fin a) R) : (-(GlobalTheory.Tw.Phi a R * Xᴴ * (GlobalTheory.Tw.Phi a R)ᵀ)).trace = -star X.trace

theorem **thetaInv_of** {χ : HChar L} (h1 : ∀ u, χ.1 (finId L (Units.map (conjFin (IsCMField.complexConj L).toRingEquiv).toMonoidHom u)⁻¹) = χ.1 (finId L u)) (h2 : ∀ z, dchi χ (-star z) = dchi χ z) : ThetaInv a χ

theorem **thetaInv_one_iff** (χ : HChar L) : ThetaInv 1 χ ↔ (∀ u, χ.1 (finId L (Units.map (conjFin (IsCMField.complexConj L).toRingEquiv).toMonoidHom u)⁻¹) = χ.1 (finId L u)) ∧ ∀ z, dchi χ (-star z) = dchi χ z

theorem **IsCsd.twist** {π : CG L a} {χ : HChar L} (hχ : (ofChar χ).IsCsd) (h : π.IsCsd) : (π.twist χ).IsCsd

theorem **expo_conj_of_isCsd** {χ : HChar L} (hχ : (ofChar χ).IsCsd) (τ : L →+* ℂ) : expo χ (ComplexEmbedding.conjugate τ) = -expo χ τ

theorem **expo_one** (τ : E →+* ℂ) : expo (1 : HChar E) τ = 0

theorem **host_isCMField** {R : Formalism.BasicNotions.Ranks} (H : Formalism.BasicNotions.Host R) : IsCMField H.E

theorem **host_complexConj** {R : Formalism.BasicNotions.Ranks} (H : Formalism.BasicNotions.Host R) (z : H.E) : haveI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExtUHInjective

theorem **stepY_gam** (b : β) (g : FrmG (Λ b) (θ b)) (x : Sp X.E n) : ((stepY ST y β Λ θ pref).gam b g : Matrix (Fin n) (Fin n) X.E) *ᵥ x = g.1.1 x

theorem **stepY_deco** (b : β) (g : (stepY ST y β Λ θ pref).G b) (p : (stepY ST y β Λ θ pref).D b) : (g • p).1.1.deco = ((g.1, g.2.1.1) : UF X.E n × GLG X.E n SG) • p.1.1.deco

theorem **stepY_fr** (b : β) (g : (stepY ST y β Λ θ pref).G b) (p : (stepY ST y β Λ θ pref).D b) (P : Q) : (g • p).1.1.fr P = g.1.1 * p.1.1.fr P * ((g.2.2.1 P)⁻¹).1

theorem **stepY_eq_one** (b : β) (g : (stepY ST y β Λ θ pref).G b) (p : (stepY ST y β Λ θ pref).D b) (h : g • p = p) (h1 : g.1 = 1) : g = 1

theorem **stabAut_inj** (b : β) (p : (stepY ST y β Λ θ pref).D b) : Function.Injective (stabAut y β Λ θ pref b p)

theorem **stepY_dvd** (hX : X.Ok) (R : Ranks) (hn : n ∈ gamRanks R) (b : β) (p : (stepY ST y β Λ θ pref).D b) : Nat.card (MulAction.stabilizer ((stepY ST y β Λ θ pref).G b) p) ∣ DGam X.E R

theorem **levQ_grp_idx** (θ : LevQFam X.E X.K n Q) (h : ∀ P, (θ P).H.FiniteIndex) : θ.grp.FiniteIndex

theorem **stepY_fin_orb** (hX : X.Ok) (hI : ∀ b P, (θ b P).H.FiniteIndex) (hy : ∀ N, Submodule.IsLattice X.E N → (y.ys N).Finite) (b : β) : Finite (MulAction.orbitRel.Quotient ((stepY ST y β Λ θ pref).G b) ((stepY ST y β Λ θ pref).D b))

theorem **stepY_ok** (hX : X.Ok) (hI : ∀ b P, (θ b P).H.FiniteIndex) (hr : ∀ b P g, ∃ m : ℕ, 0 < m ∧ (θ b P).lam g ^ m = 1) (hy : ∀ N, Submodule.IsLattice X.E N → (y.ys N).Finite) : (stepY ST y β Λ θ pref).Ok where

theorem **stepY_free** (hX : X.Ok) {Pr : Primes X} (hPr : Pr.Ok) (i : Fin 2) (w : SG) (hw : w.1 ∈ Pr.vG i) (hbig : 2 * X.f * n < Pr.g i - 1) (hu : ∀ b, ∀ k ∈ (Λ b w).H, IsNilpotent ((k : Matrix (Fin n) (Fin n) (𝒪 X.E ⧸ w.1.asIdeal)) - 1)) : (stepY ST y β Λ θ pref).Free

theorem **ptY_nonempty** [NeZero n] (h : (y.ys (stdLat X.E n)).Nonempty) : Nonempty (PtY ST SG Q y)

theorem **PtY.ext'** {y : Ys X n} {p p' : PtY ST SG Q y} (h1 : p.1.1.deco = p'.1.1.deco) (h2 : p.1.1.1.2 = p'.1.1.1.2) (h3 : p.1.2 = p'.1.2) : p = p'

theorem **PtY.mv_spec** {y : Ys X n} (p : PtY ST SG Q y) (d : DecoSet X.E n ST SG) (h1 : ∀ P : Q, d.1.at P.1 = p.1.1.deco.1.at P.1) (h2 : y.ys d.1.N = y.ys p.1.1.deco.1.N) : (p.mv d).1.1.deco = d ∧ (p.mv d).1.1.1.2 = p.1.1.1.2 ∧ (p.mv d).1.2 = p.1.2

theorem **PtY.mv_dia** {y : Ys X n} (p : PtY ST SG Q y) (P : ST) (a : (Res P.1)ˣ) : p.mv (DecoSet.dia1 P a p.1.1.deco) = ((((DecoSet.dia1 P a p.1.1.deco, p.1.1.1.2), p.1.1.2), p.1.2), p.2)

theorem **tdatY_ok** (hP : ∀ P : ST, conjPrime₂ P.1 ≠ P.1) (hT : ∀ P : ST, (T : 𝒪 X.E) ∈ P.1.asIdeal ∧ (T : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) : (tdatY ST T y β Λ θ pref).Ok T where

theorem **tdatY_deco** (b : β) (p : PtY ST SG Q y) : (tdatY ST T y β Λ θ pref).deco b p = p.1.1.deco.1.res (Finset.empty_subset _)

theorem **dn_ok_of_res** {d : Deco X.E n ST SG} (hd : d.Ok) (P : ST) (hP : MoveHyp ST SG P) {g : Sp X.E n → Res P.1} (h : ((d.res (Finset.empty_subset _)).dn P g).Ok) : (d.dn P g).Ok

theorem **mv_up** (p : PtY ST SG Q y) (P : ST) (hP : MoveHyp ST SG P) (hT : (T : 𝒪 X.E) ∈ P.1.asIdeal ∧ (T : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) (hQ : ∀ w : Q, w.1 ≠ P.1 ∧ w.1 ≠ conjPrime₂ P.1) (hy : y.Off P.1) {x : Sp X.E n} (hx : x ∈ p.1.1.deco.1.N) (hf : p.1.1.deco.1.f P x = 1) : (p.mv (p.1.1.deco.up P (T : 𝒪 X.E) x)).1.1.deco.1 = p.1.1.deco.1.up P (T : 𝒪 X.E) x ∧ (p.mv (p.1.1.deco.up P (T : 𝒪 X.E) x)).1.1.1.2 = p.1.1.1.2 ∧ (p.mv (p.1.1.deco.up P (T : 𝒪 X.E) x)).1.2 = p.1.2

theorem **mv_dn** (p : PtY ST SG Q y) (P : ST) (hP : MoveHyp ST SG P) (hQ : ∀ w : Q, w.1 ≠ P.1 ∧ w.1 ≠ conjPrime₂ P.1) (hy : y.Off P.1) {g : Sp X.E n → Res P.1} (hg : (p.1.1.deco.1.dn P g).Ok) : (p.mv (p.1.1.deco.dn P g)).1.1.deco.1 = p.1.1.deco.1.dn P g ∧ (p.mv (p.1.1.deco.dn P g)).1.1.1.2 = p.1.1.1.2 ∧ (p.mv (p.1.1.deco.dn P g)).1.2 = p.1.2

theorem **tdatY_tied** (hP : ∀ P : ST, MoveHyp ST SG P) (hT : ∀ P : ST, (T : 𝒪 X.E) ∈ P.1.asIdeal ∧ (T : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) (hQ : ∀ (P : ST) (w : Q), w.1 ≠ P.1 ∧ w.1 ≠ conjPrime₂ P.1) (hy : ∀ P : ST, y.Off P.1) : (tdatY ST T y β Λ θ pref).Tied T where

theorem **Ys.triv_off** (P : HeightOneSpectrum (𝒪 X.E)) : (Ys.triv X n).Off P

theorem **stepAt_ok** (hX : X.Ok) (hTp : Tp.Ok) (u : Pr.Word) (j : ℕ) (hL : j < R.m → ∀ b P, (L.levQ j b P).H.FiniteIndex ∧ ∀ g, ∃ m : ℕ, 0 < m ∧ (L.levQ j b P).lam g ^ m = 1) (hy : ∀ N, Submodule.IsLattice X.E N → ((ysAt R sel mu cst j).ys N).Finite) : (stepAt sel mu cst L Tp u j).Ok

theorem **ne_of_T_notMem** (hPr : Pr.Ok) (u : Pr.Word) (P : Pr.vT u) {w : HeightOneSpectrum (𝒪 X.E)} (h : (Pr.T u : 𝒪 X.E) ∉ w.asIdeal) : w ≠ P.1 ∧ w ≠ conjPrime₂ P.1

theorem **ysAt_off** (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ w ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : 𝒪 X.E) ∉ w.asIdeal) (j : ℕ) (P : Pr.vT u) : (ysAt R sel mu cst j).Off P.1

theorem **tdatAt_ok** (hX : X.Ok) (hPr : Pr.Ok) (u : Pr.Word) (j : ℕ) : (tdatAt sel mu cst L Tp u j).Ok (Pr.T u)

theorem **tdatAt_tied** (hX : X.Ok) (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ w ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : 𝒪 X.E) ∉ w.asIdeal) (j : ℕ) : (tdatAt sel mu cst L Tp u j).Tied (Pr.T u)

theorem **stepY_dvd_chain** {ST SG Q : Finset (HeightOneSpectrum (𝒪 X.E))} (hX : X.Ok) (i : ℕ) (hi : i ≤ R.m) (y : Ys X (R.N i)) (β : Type) [Fintype β] [DecidableEq β] (Λ : β → LevFam X.E X.K (R.N i) SG) (θ : β → LevQFam X.E X.K (R.N i) Q) (pref : DecoF X.E (R.N i) ST SG Q → Prop) (h0 : i = 0 → (stepY ST y β Λ θ pref).Free) (c : (stepY ST y β Λ θ pref).Cl) : Nat.card ((stepY ST y β Λ θ pref).Stab c) ∣ DGam X.E R

theorem **chainAt_ok** (hX : X.Ok) (u : Pr.Word) (rho rho' cD Th) (hS : ∀ i, (stepAt sel mu cst L Tp u i).Ok) (h0 : (stepAt sel mu cst L Tp u 0).Free) (h1 : ∀ i, i < R.m → ∀ b p, ‖rho i b p‖ = 1) (h2 : ∀ i, i < R.m → ∀ b p, ‖rho' i b p‖ = 1) : (chainAt sel mu cst L Tp u rho rho' cD Th).Ok where

theorem **exists_vG** (hX : X.Ok) (hPr : Pr.Ok) : ∃ w : Pr.vGs, w.1 ∈ Pr.vG 0

theorem **stepAt_free** (Ch : Chars X Pr R.m) (hX : X.Ok) (hPr : Pr.Ok) (hbig : 2 * X.f * R.k < Pr.g 0 - 1) (hL : ∀ b, L.levG 0 b = colLev Ch 0) (u : Pr.Word) : (stepAt sel mu cst L Tp u 0).Free

theorem **iniH_extUH** (γ : UF E n) (x : Sp E N) : iniH hN ((extUH hN γ).1 x) = γ.1 (iniH hN x)

theorem **extUH_last** (γ : UF E n) (x : Sp E N) : (extUH hN γ).1 x (Br.iL hN) = x (Br.iL hN)

theorem **extUH_injective** : Function.Injective (extUH (E

theorem **Lsh_empty** (d : DecoSet E n ∅ SG) : d.1.Lsh = d.1.N

theorem **tilH_N** (x : Sp E N) : x ∈ (DecoSet.tilH (ST

theorem **tilH_f** (P : ST) (x : Sp E N) (hx : x ∈ (DecoSet.tilH (ST

theorem **tilH_Lsh** (hpi : ∀ P : ST, pi ∈ P.1.asIdeal) (hdvd : ∀ r : 𝒪 E, (∀ P : ST, r ∈ P.1.asIdeal) → pi ∣ r) (x : Sp E N) : x ∈ (DecoSet.tilH (ST

theorem **tr_one** (b : S.β) (F : Fr emb σ n → ℂ) : S.tr emb σ b 1 F = F

theorem **tr_mul** (b : S.β) (g h : S.G b) (F : Fr emb σ n → ℂ) : S.tr emb σ b (g ∗ h) F = S.tr emb σ b g (S.tr emb σ b h F)

theorem **evV_rho** (b : S.β) (g : S.G b) (v : VV K lam) : evV emb (1 : Matrix (Fin n) (Fin n) E) lam σ (S.rho emb lam b g v) = S.tr emb σ b g (evV emb (1 : Matrix (Fin n) (Fin n) E) lam σ v)

theorem **embF_mem** (h : (S.fam emb lam).Ok) (x : S.FormsK emb lam h) : S.embF emb σ lam h x ∈ S.Fi emb σ lam

theorem **cl_rep** (c : S.Cl) : S.cl c.1 (S.rep c) = c

theorem **cl_smul** (b : S.β) (g : S.G b) (d : S.D b) : S.cl b (g • d) = S.cl b d

theorem **val_smul** (hf : S.Free) (φ : S.Cl → Fr emb σ n → ℂ) (b : S.β) (g : S.G b) (d : S.D b) : S.trTo emb σ b (g • d) (φ (S.cl b (g • d))) = S.tr emb σ b g (S.trTo emb σ b d (φ (S.cl b d)))

theorem **diaOne** (h : J.Ok) : J.LawDiaOne emb σ

theorem **diaMul** (h : J.Ok) : J.LawDiaMul emb σ

theorem **tr_mem** {n : ℕ} (S : GStep E K n) (emb : ι → E →+* K) (σ : K →+* ℂ) (lam : ι → DomWeight n) (b : S.β) (g : S.G b) {F : Fr emb σ n → ℂ} (hF : F ∈ ValC (Gr emb σ n) lam) : S.tr emb σ b g F ∈ ValC (Gr emb σ n) lam

theorem **valC_fd** {n : ℕ} (emb : ι → E →+* K) (σ : K →+* ℂ) (lam : ι → DomWeight n) : FiniteDimensional ℂ (ValC (Gr emb σ n) lam)

theorem **Fi_fd** {n : ℕ} (S : GStep E K n) (h : S.Ok) (emb : ι → E →+* K) (σ : K →+* ℂ) (lam : ι → DomWeight n) : FiniteDimensional ℂ (S.Fi emb σ lam)

theorem **mem_Fi** {n : ℕ} (S : GStep E K n) (hf : S.Free) (emb : ι → E →+* K) (σ : K →+* ℂ) (lam : ι → DomWeight n) (φ : S.Cl → Fr emb σ n → ℂ) (h : ∀ c, φ c ∈ ValC (Gr emb σ n) lam) : φ ∈ S.Fi emb σ lam

theorem **Fi_ne_bot** {n : ℕ} (S : GStep E K n) (hf : S.Free) (c : S.Cl) (emb : ι → E →+* K) (σ : K →+* ℂ) (lam : ι → DomWeight n) : S.Fi emb σ lam ≠ ⊥

theorem **atLaws** (hI : I.Ok) (h : J.Ok) (hc : ι → HCParam R) : J.Laws emb σ hc where

theorem **N_succ** (j : ℕ) : R.N j = (R.N j - 1) + 1

theorem **lev_eq_one** {n : ℕ} (h : (lev X Pr n).grp) : h = 1

theorem **top_ok** : (top X Pr R).Ok where

theorem **free** (hX : X.Ok) (hPr : Pr.Ok) (hbig : 2 ∗ X.f ∗ R.k < Pr.g 0 - 1) : ((chainIn X Pr R u Jt).S 0).Free

theorem **chainIn_ok** (hX : X.Ok) (hPr : Pr.Ok) (hbig : 2 ∗ X.f ∗ R.k < Pr.g 0 - 1) : (chainIn X Pr R u Jt).Ok where

theorem **cl_nonempty** (i : ℕ) : Nonempty ((chainIn X Pr R u Jt).S i).Cl

theorem **stepX_ok** (hX : X.Ok) : (stepX X Pr R).Ok

theorem **perIn_ok** (hX : X.Ok) (hPr : Pr.Ok) (sg : X.Tau → DomWeight (R.N R.m - 1)) : (perIn X Pr R u Jt sg).Ok where

theorem **perIn_over** (hPr : Pr.Ok) (sg : X.Tau → DomWeight (R.N R.m - 1)) (psi : ∀ P : Pr.vT u, AddChar (Res P.1) ℂ) : (perIn X Pr R u Jt sg).Over (atIn X Pr R u Jt psi) where

theorem **atIn_ok** (hX : X.Ok) (hPr : Pr.Ok) (hbig : 2 ∗ X.f ∗ R.k < Pr.g 0 - 1) (hf : X.f = R.f) : ∃ psi, (atIn X Pr R u Jt psi).Ok

theorem **all_ok** (hX : X.Ok) (hPr : Pr.Ok) (hbig : 2 ∗ X.f ∗ R.k < Pr.g 0 - 1) (hf : X.f = R.f) (u : Pr.Word) (sg : X.Tau → DomWeight (R.N R.m - 1)) (hc : X.Tau → HCParam R) (Pl : Rows.Places X.K) : ∃ (σ : X.K →+* ℂ) (I : ChainIn X.E X.K X.Tau R) (J : AtIn I) (Q : PerIn I), FOk X.emb σ ∧ I.Ok ∧ J.Ok ∧ Q.Ok ∧ Q.Over J ∧ J.Laws X.emb σ hc ∧ Nonempty (FmIn J Pl) ∧ (∀ i, Nonempty (I.S i).Cl) ∧ (I.S 0).Fi X.emb σ (lamT hc 0) ≠ ⊥ ∧ I.DG = DGam X.E R

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ExtensionPropertyPi

theorem **sigmaRel_mk** {ι : Type} {α β : ι → Type} {R : ∀ i, α i → β i → Prop} {i : ι} {a : α i} {b : β i} : SigmaRel R ⟨i, a⟩ ⟨i, b⟩ ↔ R i a b

theorem **SigmaRel.fst_eq** {ι : Type} {α β : ι → Type} {R : ∀ i, α i → β i → Prop} {a : Σ i, α i} {b : Σ i, β i} (h : SigmaRel R a b) : a.1 = b.1

theorem **Loc.bs_pi** (L : ι → ModuliIntegralCharts.Loc D r j p) (b : Option ℕ) : ModuliIntegralCharts.bs (Loc.pi L) b = fun v => ModuliIntegralCharts.bs (L v) b

@[simp] theorem **Loc.pi_Xm** (v : ι) : (Loc.pi L).Xm v = (L v).Xm

@[simp] theorem **Loc.pi_YY** (v : ι) : (Loc.pi L).YY v = (L v).YY

@[simp] theorem **Loc.pi_Xn** (n : ℕ) (v : ι) : (Loc.pi L).Xn n v = (L v).Xn n

@[simp] theorem **Loc.pi_fld** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).fld (Z

@[simp] theorem **Loc.pi_be** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).be (Z

@[simp] theorem **Loc.pi_good** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).good (Z

@[simp] theorem **Loc.pi_phiCy** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).phiCy (Z

@[simp] theorem **Loc.pi_dP** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).dP (Z

@[simp] theorem **Loc.pi_cJ** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) (n DF : ℕ) : (Loc.pi L).cJ (Z

@[simp] theorem **Loc.pi_isCp** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : ModuliIntegralCharts.isCp (Loc.pi L) (Z

@[simp] theorem **Loc.pi_psiV** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).psiV (Z

@[simp] theorem **Loc.pi_ht** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).ht (Z

@[simp] theorem **Loc.pi_om** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).om (Z

@[simp] theorem **Loc.pi_abAt** {Z : (Loc.pi L).Mdl} (v : ι) (η : (L v).Gen (Z v)) : (Loc.pi L).abAt (Z

@[simp] theorem **Loc.pi_isInv** {Z : (Loc.pi L).Mdl} (F : (Loc.pi L).Sh Z) : (Loc.pi L).IsInv F ↔ ∀ v, (L v).IsInv (F v)

@[simp] theorem **Loc.pi_isLocFree** {Z : (Loc.pi L).Mdl} (F : (Loc.pi L).Sh Z) : (Loc.pi L).IsLocFree F ↔ ∀ v, (L v).IsLocFree (F v)

@[simp] theorem **Loc.pi_nrm** {Z : (Loc.pi L).Mdl} (F : (Loc.pi L).Sh Z) (f : (Loc.pi L).Sec F) (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).nrm F f ⟨v, y⟩ = (L v).nrm (F v) (f v) y

@[simp] theorem **Loc.pi_vrat** {Z : (Loc.pi L).Mdl} {F₁ F₂ : (Loc.pi L).Sh Z} (φ : (Loc.pi L).GIso F₁ F₂) (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).vrat φ ⟨v, y⟩ = (L v).vrat (φ v) y

@[simp] theorem **Loc.pi_Lb** (l : ModuliIntegralCharts.PsiT D → ℕ → ℤ) (v : ι) : (Loc.pi L).Lb l v = (L v).Lb l

@[simp] theorem **Loc.pi_nrmRef** (l : ModuliIntegralCharts.PsiT D → ℕ → ℤ) (f : (Loc.pi L).Sec ((Loc.pi L).Lb l)) (v : ι) (y : (L v).Pt (L v).YY) : (Loc.pi L).nrmRef l f ⟨v, y⟩ = (L v).nrmRef l (f v) y

@[simp] theorem **Loc.pi_dvRho** (v : ι) (y : (L v).DVP) : (Loc.pi L).dvRho ⟨v, y⟩ = (L v).dvRho y

@[simp] theorem **Loc.pi_isChart** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).DVP) (e : (L v).Ext y) (χ : (L v).Chart (Z v) y e) : ModuliIntegralCharts.IsChart (Loc.pi L) (Z

@[simp] theorem **Loc.pi_phiAt** (φ : (Loc.pi L).PL) {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (Loc.pi L).phiAt φ (Z

@[simp] theorem **Loc.pi_fjN** (l : ModuliIntegralCharts.PsiT D → ℕ → ℤ) (f : (Loc.pi L).Sec ((Loc.pi L).Lb l)) (v : ι) (t : (L v).FJd) (χ : (L v).Chr t) : (Loc.pi L).fjN l f ⟨v, t⟩ χ = (L v).fjN l (f v) t χ

@[simp] theorem **Loc.pi_piPP** (v : ι) : (Loc.pi L).piPP v = (L v).piPP

@[simp] theorem **Loc.pi_keySh** (c : ℕ) (v : ι) : (Loc.pi L).keySh c v = (L v).keySh c

@[simp] theorem **Loc.pi_latStd** (n : ℕ) (v : ι) : (Loc.pi L).latStd n v = (L v).latStd n

@[simp] theorem **Loc.pi_symLift** (v : ι) (y : (L v).Pt (L v).Xm) (La : (Loc.pi L).Unim) (n : ℕ) (sg : (L v).At y (La v) n → (L v).Th y (La v) n) : (Loc.pi L).SymLift ⟨v, y⟩ La n sg ↔ (L v).SymLift y (La v) n sg

@[simp] theorem **Loc.pi_secJet** (v : ι) (y : (L v).Pt (L v).Xm) (La : (Loc.pi L).Unim) (n DF : ℕ) (sg : (L v).At y (La v) n → (L v).Th y (La v) n) : (Loc.pi L).secJet ⟨v, y⟩ La n DF sg = (L v).secJet y (La v) n DF sg

@[simp] theorem **Loc.pi_intJet** (v : ι) (y : (L v).Pt (L v).Xm) (La : (Loc.pi L).Unim) (n DF : ℕ) : (Loc.pi L).intJet ⟨v, y⟩ La n DF = (L v).intJet y (La v) n DF

@[simp] theorem **Loc.pi_isHTBlowup** (Z' Z : (Loc.pi L).Mdl) : (Loc.pi L).IsHTBlowup Z' Z ↔ ∀ v, (L v).IsHTBlowup (Z' v) (Z v)

@[simp] theorem **Loc.pi_degLdR** (v : ι) (c : (L v).Crv) : (Loc.pi L).degLdR ⟨v, c⟩ = (L v).degLdR c

@[simp] theorem **Loc.mem_sections_pi_iff** {Z : (Loc.pi L).Mdl} (F : (Loc.pi L).Sh Z) (f : (Loc.pi L).Sec F) : f ∈ (Loc.pi L).H0 F ↔ ∀ v, f v ∈ (L v).H0 (F v)

theorem **Loc.exists_pi_mdl** (v : ι) (Z : (L v).Mdl) : ∃ Z' : (Loc.pi L).Mdl, Z' v = Z

theorem **Loc.exists_pi_unim** (v : ι) (La : (L v).Unim) : ∃ La' : (Loc.pi L).Unim, La' v = La

theorem **Loc.exists_pi_sec** {Z : (Loc.pi L).Mdl} (F : (Loc.pi L).Sh Z) (v : ι) (f : (L v).Sec (F v)) : ∃ f' : (Loc.pi L).Sec F, f' v = f ∧ ∀ w, w ≠ v → f' w = 0

theorem **Loc.thetaExt_pi** (La : (Loc.pi L).Unim) (H : ∀ v, ∃ P : (L v).LBG (La v), (L v).resQ P = (L v).PPq (La v) ∧ (L v).symmG P ∧ (L v).rigG P ∧ ∀ y, ∃ M : (L v).LBB y (La v), (L v).principal M ∧ (L v).resGt P y = (L v).prPull M) : ∃ P : (Loc.pi L).LBG La, (Loc.pi L).resQ P = (Loc.pi L).PPq La ∧ (Loc.pi L).symmG P ∧ (Loc.pi L).rigG P ∧ ∀ y, ∃ M : (Loc.pi L).LBB y La, (Loc.pi L).principal M ∧ (Loc.pi L).resGt P y = (Loc.pi L).prPull M

@[simp] theorem **ExtA.pi_flagCl** (Z : (Loc.pi L).Mdl) (s : (ExtA.pi G).FlagData Z) (v : ι) : (ExtA.pi G).flagCl Z s v = (G v).flagCl (Z v) (s v)

@[simp] theorem **ExtA.pi_blowUp** (Z : (Loc.pi L).Mdl) (I : (ExtA.pi G).Idl Z) (v : ι) : (ExtA.pi G).blowUp Z I v = (G v).blowUp (Z v) (I v)

@[simp] theorem **ExtA.pi_diagCl** (Z₁ Z₂ : (Loc.pi L).Mdl) (c : (ExtA.pi G).Corr Z₁ Z₂) (v : ι) : (ExtA.pi G).diagCl Z₁ Z₂ c v = (G v).diagCl (Z₁ v) (Z₂ v) (c v)

@[simp] theorem **ExtA.pi_jIdl_none** (d : (ExtA.pi G).IvD none) (v : ι) : (ExtA.pi G).jIdl none d v = (G v).jIdl none (d v)

@[simp] theorem **ExtA.pi_jIdl_some** (n : ℕ) (d : (ExtA.pi G).IvD (some n)) (v : ι) : (ExtA.pi G).jIdl (some n) d v = (G v).jIdl (some n) (d v)

@[simp] theorem **ExtA.pi_ptK** (Z : (Loc.pi L).Mdl) (v : ι) (y : (L v).DVP) : (ExtA.pi G).ptK Z ⟨v, y⟩ = ⟨v, (G v).ptK (Z v) y⟩

@[simp] theorem **ExtA.pi_yEta** {Z : (Loc.pi L).Mdl} (v : ι) (η : (L v).Gen (Z v)) : (ExtA.pi G).yEta (Z

@[simp] theorem **ExtA.pi_omP** {n : ℕ} (h : (ExtA.pi G).HTn n) (v : ι) (y : (L v).Pt ((L v).Xn n)) : (ExtA.pi G).omP h ⟨v, y⟩ = (G v).omP (h v) y

@[simp] theorem **ExtA.pi_spz** {Z : (Loc.pi L).Mdl} (v : ι) (x x' : (G v).ZP (Z v)) : (ExtA.pi G).spz (Z

theorem **ExtA.pi_isHT** {n : ℕ} (h : (ExtA.pi G).HTn n) : (ExtA.pi G).IsHT h ↔ ∀ v, (G v).IsHT (h v)

theorem **ExtA.pi_blowUp_jIdl** (b : Option ℕ) (d : (ExtA.pi G).IvD b) (v : ι) : (ExtA.pi G).blowUp (ModuliIntegralCharts.bs (Loc.pi L) b) ((ExtA.pi G).jIdl b d) v = (G v).blowUp (ModuliIntegralCharts.bs (L v) b) ((G v).jIdl b (d v))

theorem **ExtA.zz_of_pi** {N : ℕ} {Z : (Loc.pi L).Mdl} (h : (ExtA.pi G).ZZ N Z) (v : ι) : (G v).ZZ N (Z v)

theorem **ExtA.aHT_pi_iff** : (ExtA.pi G).aHT ↔ ∀ v, (G v).aHT

theorem **ExtA.aKrull_pi_iff** : (ExtA.pi G).aKrull ↔ ∀ v, (G v).aKrull

theorem **ExtA.aNorm_pi** (H : ∀ v, (G v).aNorm) : (ExtA.pi G).aNorm

theorem **ExtA.aTube_pi** (H : ∀ v, (G v).aTube) : (ExtA.pi G).aTube

theorem **ExtA.aOrd1_pi** (H : ∀ v, (G v).aOrd1) : (ExtA.pi G).aOrd1

theorem **ExtA.aOrd2_pi** (H : ∀ v, (G v).aOrd2) : (ExtA.pi G).aOrd2

theorem **ExtA.aOrd3_pi** (H : ∀ v, (G v).aOrd3) : (ExtA.pi G).aOrd3

theorem **ExtA.aSub_pi** (H : ∀ v, (G v).aSub) : (ExtA.pi G).aSub

theorem **ExtA.aClose_pi** (H : ∀ v, (G v).aClose) : (ExtA.pi G).aClose

theorem **ExtA.baseCharts_pi** (H : ∀ v, (G v).ChartLiftsThroughSteps) : (ExtA.pi G).ChartLiftsThroughSteps

theorem **ExtA.aCl_pi** (H : ∀ v, (G v).aCl) : (ExtA.pi G).aCl

theorem **ExtA.aHTI_pi** (H : ∀ v, (G v).aHTI) : (ExtA.pi G).aHTI

theorem **ExtA.extensionProperty_pi** (H : ∀ v, (G v).p0) : (ExtA.pi G).p0

theorem **ExtA.pC_pi** (H : ∀ v, (G v).pC) : (ExtA.pi G).pC

theorem **ExtA.pNV_pi** (H : ∀ v, (G v).pNV) : (ExtA.pi G).pNV

theorem **ExtA.pTR_pi** (H : ∀ v, (G v).pTR) : (ExtA.pi G).pTR

theorem **ExtA.pORD_pi** (H : ∀ v, (G v).pORD) : (ExtA.pi G).pORD

theorem **ExtA.pFJ_pi** (H : ∀ v, (G v).pFJ) : (ExtA.pi G).pFJ

theorem **ExtA.pHTa_pi_iff** : (ExtA.pi G).pHTa ↔ ∀ v, (G v).pHTa

theorem **ExtA.zz_of_isHTBlowup** (HI : ∀ v, (G v).aHTI) {n : ℕ} (hn : 3 ≤ n) {Z' : (Loc.pi L).Mdl} (hZ : (Loc.pi L).IsHTBlowup Z' ((Loc.pi L).Xn n)) : (ExtA.pi G).ZZ n Z'

theorem **ExtA.pHTb_pi** (HI : ∀ v, (G v).aHTI) (H : ∀ v, (G v).pHTb) : (ExtA.pi G).pHTb

theorem **ExtA.pHTc_pi** (H : ∀ v, (G v).pHTc) : (ExtA.pi G).pHTc

theorem **ExtA.pHTd_pi** (H : ∀ v, (G v).pHTd) : (ExtA.pi G).pHTd

theorem **ExtA.pGG_pi_iff** : (ExtA.pi G).pGG ↔ ∀ v, (G v).pGG

theorem **ExtA.pDR_pi_iff** : (ExtA.pi G).pDR ↔ ∀ v, (G v).pDR

@[simp] theorem **LocB.pi_alp** : (LocB.pi B α hα).alp = α

@[simp] theorem **LocB.pi_phi** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (LocB.pi B α hα).phi (Z

@[simp] theorem **LocB.pi_tau** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (LocB.pi B α hα).tau (Z

@[simp] theorem **LocB.pi_shiftA** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (LocB.pi B α hα).shiftA (Z

@[simp] theorem **LocB.pi_psiA** {Z : (Loc.pi L).Mdl} (v : ι) (y : (L v).Pt (Z v)) : (LocB.pi B α hα).psiA (Z

@[simp] theorem **LocB.pi_oX** (v : ι) : (LocB.pi B α hα).oX v = (B v).oX

@[simp] theorem **LocB.pi_secOf** {F : (Loc.pi L).Sh (Loc.pi L).Xm} (φ : (Loc.pi L).GIso (LocB.pi B α hα).oX F) (v : ι) : (LocB.pi B α hα).secOf φ v = (B v).secOf (φ v)

@[simp] theorem **LocB.pi_dimB** (v : ι) (y : (L v).Pt (L v).Xm) : (LocB.pi B α hα).dimB ⟨v, y⟩ = (B v).dimB y

@[simp] theorem **LocB.pi_nfc** (v : ι) (y : (L v).Pt (L v).Xm) (nW n : ℕ) : (LocB.pi B α hα).nfc ⟨v, y⟩ nW n = (B v).nfc y nW n

@[simp] theorem **LocB.pi_NN** (v : ι) (y : (L v).Pt (L v).Xm) (nW n : ℕ) : (LocB.pi B α hα).NN ⟨v, y⟩ nW n = (B v).NN y nW n

theorem **LocB.exists_secOf_pi_iff** {F : (Loc.pi L).Sh (Loc.pi L).Xm} (s : (Loc.pi L).Sec F) : (∃ φ : (Loc.pi L).GIso (LocB.pi B α hα).oX F, (LocB.pi B α hα).secOf φ = s) ↔ ∀ v, ∃ φ : (L v).GIso (B v).oX (F v), (B v).secOf φ = s v

theorem **LocB.poincareExt_pi** (La : (Loc.pi L).Unim) (H : ∀ v, ∃ P : (B v).StrictHenselianAcyclic (La v), (B v).res2 P = (B v).PoiQ (La v) ∧ (B v).biext P ∧ (L v).symmG ((B v).dg P) ∧ (L v).rigG ((B v).dg P) ∧ ∀ y, ∃ M : (L v).LBB y (La v), (L v).principal M ∧ (L v).resGt ((B v).dg P) y = (L v).prPull M) : ∃ P : (LocB.pi B α hα).StrictHenselianAcyclic La, (LocB.pi B α hα).res2 P = (LocB.pi B α hα).PoiQ La ∧ (LocB.pi B α hα).biext P ∧ (Loc.pi L).symmG ((LocB.pi B α hα).dg P) ∧ (Loc.pi L).rigG ((LocB.pi B α hα).dg P) ∧ ∀ y, ∃ M : (Loc.pi L).LBB y La, (Loc.pi L).principal M ∧ (Loc.pi L).resGt ((LocB.pi B α hα).dg P) y = (Loc.pi L).prPull M

theorem **LocB.mul_pi_iff** : (LocB.pi B α hα).Mul ↔ ∀ v, (B v).Mul

theorem **LocB.fer_pi_iff** : (LocB.pi B α hα).FER ↔ ∀ v, (B v).FER

theorem **LocB.regR_pi_iff** : (LocB.pi B α hα).RegR ↔ ∀ v, (B v).RegR

theorem **LocB.locFree_pi_iff** : (LocB.pi B α hα).LocFree ↔ ∀ v, (B v).LocFree

theorem **LocB.fou_pi_iff** : (LocB.pi B α hα).Fou ↔ ∀ v, (B v).Fou

theorem **LocB.freeB_pi_iff** : (LocB.pi B α hα).FreeB ↔ ∀ v, (B v).FreeB

theorem **LocB.actF_pi_iff** : (LocB.pi B α hα).ActF ↔ ∀ v, (B v).ActF

theorem **LocB.jetB_pi_iff** : (LocB.pi B α hα).JetB ↔ ∀ v, (B v).JetB

theorem **LocB.vthF_pi_iff** : (LocB.pi B α hα).VthF ↔ ∀ v, (B v).VthF

theorem **LocB.desc_pi** (H : ∀ v, (B v).Desc) : (LocB.pi B α hα).Desc

theorem **Loc.perf_pi_iff** : BoundaryLocalData.LocB.Perf (Loc.pi L) ↔ ∀ v, BoundaryLocalData.LocB.Perf (L v)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FibLatPreSubSpanSubLe

theorem **exists_affine_le** {R : Type} [CommRing R] [IsLocalRing R] {X : Scheme.{0}} {z : Sp R ⟶ X} {U : X.Opens} (e : ⊤ ≤ z ⁻¹ᵁ U) : ∃ W : X.Opens, IsAffineOpen W ∧ ⊤ ≤ z ⁻¹ᵁ W ∧ W ≤ U

theorem **omVal_smul** (a : Γ(Sp ↥(intO ↥C.S), ⊤)) (s : Γ(omega P.G, ⊤)) : omVal P (a • s) = (show ↥(intO ↥C.S) from (Scheme.ΓSpecIso (.of ↥(intO ↥C.S))).hom a) • omVal P s

theorem **exists_omVal_eq** {w : P.OmC} (hw : w ∈ P.latOm) : ∃ s, omVal P s = w

theorem **evPt_apply** (e : ⊤ ≤ z ⁻¹ᵁ U) (s : Γ(omega G, U)) : evPt hφ e s = omVal P (pb hφ e s)

theorem **evPt_mem** (e : ⊤ ≤ z ⁻¹ᵁ U) (s : Γ(omega G, U)) : evPt hφ e s ∈ P.latOm

theorem **evPt_smul** (e : ⊤ ≤ z ⁻¹ᵁ U) (a : Γ(X, U)) (s : Γ(omega G, U)) : evPt hφ e (a • s) = evz z e a • evPt hφ e s

theorem **evPt_resM** (e : ⊤ ≤ z ⁻¹ᵁ V) (k : V ≤ U) (s : Γ(omega G, U)) : evPt hφ e (resM _ k s) = evPt hφ (e.trans (z.preimage_mono k)) s

theorem **fibLat_le** (M : Sub (omega G)) : fibLat hφ M ≤ P.latOm

theorem **evPt_mem_fibLat** {M : Sub (omega G)} (e : ⊤ ≤ z ⁻¹ᵁ U) {s : Γ(omega G, U)} (hs : s ∈ M.sec U) : evPt hφ e s ∈ fibLat hφ M

theorem **IsLiftAt.mem** (h : IsLiftAt hφ u v) : v ∈ P.latOm

theorem **IsLiftAt.of_sub** (h : IsLiftAt hφ u v) {w : P.OmC} (hw : w ∈ P.latOm) (e : v' − v = N • w) : IsLiftAt hφ u v'

theorem **isLiftAt_zero** (hφ : IsGrpPullback P.G G z φ) (N : ℕ) : IsLiftAt hφ (0 : Γ(modN (omega G) N, ⊤)) 0

theorem **IsLiftAt.add** (h : IsLiftAt hφ u v) (h' : IsLiftAt hφ u' v') : IsLiftAt hφ (u + u') (v + v')

theorem **IsLiftAt.sub_evPt** (h : IsLiftAt hφ u v) (e : ⊤ ≤ z ^-1ᵘ U) {s : Γ(omega G, U)} (hs : (cokernel.π (N • 𝟙 (omega G))).app U s = resM _ le_top u) : ∃ w ∈ P.latOm, v - evPt hφ e s = N • w

theorem **IsLiftAt.sub** (h : IsLiftAt hφ u v) (h' : IsLiftAt hφ u v') : ∃ w ∈ P.latOm, v - v' = N • w

theorem **exists_isLiftAt** (u : Γ(modN (omega G) N, ⊤)) : ∃ v, IsLiftAt hφ u v

theorem **span_range_evPt** (hU : IsAffineOpen U) (e : ⊤ ≤ z ^-1ᵘ U) : Submodule.span ɪ(intO ɪC.S) (Set.range (evPt hφ e)) = P.latOm

theorem **nsmul_mem_fibLat_preSub** (M : Sub (modN (omega G) N)) {w : P.OmC} (hw : w ∈ P.latOm) : N • w ∈ fibLat hφ (preSub (cokernel.π (N • 𝟙 (omega G))) M)

theorem **exists_eq_of_mem_preSub** {F : X.Modules} [F.IsQuasicoherent] {J : Type} [Fintype J] (u : J → Γ(modN F N, ⊤)) (hU : IsAffineOpen U) {s : Γ(F, U)} (hs : s ∈ (preSub (cokernel.π (N • 𝟙 F)) (spanSub u)).sec U) : ∃ (a : J → Γ(X, U)) (t : J → Γ(F, U)) (r : Γ(F, U)), (∀ i, (cokernel.π (N • 𝟙 F)).app U (t i) = resM _ le_top (u i)) ∧ s = N • r + ∑ i, a i • t i

theorem **fibLat_preSub_spanSub_le** {J : Type} [Fintype J] (u : J → Γ(modN (omega G) N, ⊤)) {L : Submodule ɪ(intO ɪC.S) P.OmC} (h₀ : ∀ w ∈ P.latOm, N • w ∈ L) (h₁ : ∀ i, ∃ v ∈ L, IsLiftAt hφ (u i) v) : fibLat hφ (preSub (cokernel.π (N • 𝟙 (omega G))) (spanSub u)) ≤ L

theorem **IsLiftAt.mem_fibLat** {M : Sub (modN (omega G) N)} (hu : u ∈ M.sec ⊤) (h : IsLiftAt hφ u v) : v ∈ fibLat hφ (preSub (cokernel.π (N • 𝟙 (omega G))) M)

theorem **IsLiftAt.pbMod** {X' : Scheme.{0}} {G' : Over X'} [GrpObj G'] {f : X' ⟶ X} {ψ : G'.left ⟶ G.left} (hψ : IsGrpPullback G' G f ψ) {z' : Sp ɪ(intO ɪC.S) ⟶ X'} {φ' : P.G.left ⟶ G'.left} (hφ' : IsGrpPullback P.G G' z' φ') (hz : z' ≫ f = z) (e : φ' ≫ ψ = φ) (h : IsLiftAt hφ u v) : IsLiftAt hφ' (pbMod hψ N le_top u) v

theorem **lv_zero** : lv 0 = 1

theorem **IsVal.isLiftAt** (hker : ∀ a, lv a = 1 → ∃ b, a = p ^ n • b) {t : P.T} (hv : IsVal hφ lv h t v) {a : La} (ha : lv a = t.lev n) : IsLiftAt hφ (h a) v

theorem **exists_isVal** (honto : ∀ q, q ^ p ^ n = 1 → ∃ a, lv a = q) : ∃ f : P.T →+ P.OmC, ∀ t, IsVal hφ lv h t (f t)

theorem **levPt_ɪ** (k : ℕ) (a : Dw.La) : (levPt P k a).left ≫ abVarɪ P = P.y ≫ (Y.oF.etaA a ^ (l.A / k)).left

theorem **ptsEquiv_left** (x : Pts (abVar P).A) : (ptsEquiv P x).left = x.left ≫ abVarɪ P

theorem **ptsEquiv_levPt** (k : ℕ) (a : Dw.La) : ptsEquiv P (levPt P k a) = (toUnit _ ≫ Y.oF.etaA a) ^ (l.A / k)

theorem **levPt_add** (k : ℕ) (a b : Dw.La) : levPt P k (a + b) = levPt P k a ∗ levPt P k b

theorem **levPt_spec** {k : ℕ} (hk : k ∣ l.A) : (∀ q, q ^ k = 1 ↔ ∃ a, levPt P k a = q) ∧ (∀ a, levPt P k a = 1 ↔ ∃ b, a = k • b) ∧ letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FieldTies

theorem **CorePin.reCore** {J : CoreJ D} {pb : Fin 2 → HeightOneSpectrum (𝒪 D.host.K)} (h : CorePin J.toCore pb) {dict mu chiW x₀ cE} (hM : LawsM (GlobalTheory.RealGeo.reCore J dict mu chiW x₀ cE).toCore) : CorePin (GlobalTheory.RealGeo.reCore J dict mu chiW x₀ cE).toCore pb

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FinTypRepCast

theorem **isotL_eq** : @principalLevelFrame.isotL b.D = @isotLB B C a b

theorem **finTypRep_cast** {E : Type} [Field E] [NumberField E] [IsCMField E] {n : ℕ} {S S' : Set (Formalism.L2.NG.FinTyp E n)} (e : S = S') (σ : Formalism.L2.Col.FinTypCls S) : Formalism.L2.Col.finTypRep S σ = Formalism.L2.Col.finTypRep S' (cast (congrArg Formalism.L2.Col.FinTypCls e) σ)

theorem **topMult_tr** (h : TopMultH R) : letI

theorem **chain_EEc** : (toChainReal R F tupT hm hTop).EEc = ((chainB b F tupT hm hTop).EEc)

theorem **chain_CC** : (toChainReal R F tupT hm hTop).CC = ((chainB b F tupT hm hTop).CC)

theorem **chain_CCP** : (toChainReal R F tupT hm hTop).CCP = ((chainB b F tupT hm hTop).CCP)

theorem **chain_CCX** : (toChainReal R F tupT hm hTop).CCX = ((chainB b F tupT hm hTop).CCX)

theorem **chain_CCXP** : (toChainReal R F tupT hm hTop).CCXP = ((chainB b F tupT hm hTop).CCXP)

theorem **chain_ioC** (t : T) (k : ℕ) (v : AmbT (R.MY (k + 1))) (hv : v ∈ ccJ (@multTB B C b a) R t (k + 1)) (hv0 : v ∈ ccJ (@MultT a.D) R t (k + 1)) : (ioCW (@multTB B C b a) R (ioYB b R) t k ⟨v, hv⟩).1 = (ioCW (@MultT a.D) R ((R.ioY)) t k ⟨v, hv0⟩).1

theorem **chain_ioX** (t : T) (v : AmbT R.MX) (hT : topW (@multTB B C b a) R) (hv : v ∈ multTB b a R.θ R.MX (levS _ R.βX (R.datXt t)) (levChi _ R.βX (R.datXt t))) (hv0 : v ∈ MultT R.θ R.MX (levS _ R.βX (R.datXt t)) (levChi _ R.βX (R.datXt t))) : (ioXW (@multTB B C b a) R (ioXB b R) hm hT t ⟨v, hv⟩).1 = (ioXW (@MultT a.D) R ((R.ioX)) hm hTop t ⟨v, hv0⟩).1

theorem **chain_ioOK** {c : Bool} {μ : a.D.what₂.Tup} (h : (chainB b F tupT hm hTop).IoOK c ((μ))) : (toChainReal R F tupT hm hTop).IoOK c μ

theorem **inj_tr** : letI

theorem **lift_Wc** : (toLiftReal R F tupT hm hTop hinj).Wc = ((liftB b F tupT hm hTop hinj).Wc)

theorem **lift_MultW** : (toLiftReal R F tupT hm hTop hinj).MultW = ((liftB b F tupT hm hTop hinj).MultW)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FiniteDimensionalLetter

theorem **letterQ_surjective** : Function.Surjective (letterQ m F τ)

theorem **ker_letterQ** : LinearMap.ker (letterQ m F τ) = letterKer m F τ

@[simp] theorem **letterMk_zero** : letterMk m F τ 0 = 0

theorem **letterMk_add** (x y : D) : letterMk m F τ (x + y) = letterMk m F τ x + letterMk m F τ y

theorem **letterMk_smul** (a : LL) (x : D) : letterMk m F τ (a • x) = algebraMap LL F a • letterMk m F τ x

theorem **letterMk_mul** (c : U) (x : D) : letterMk m F τ (m c x) = τ c • letterMk m F τ x

theorem **span_letterMk** : Submodule.span F (Set.range (letterMk m F τ)) = ⊤

theorem **letter_hom_ext** {M : Type*} [AddCommGroup M] [Module F M] {f g : Letter m F τ →ₗ[F] M} (h : ∀ x, f (letterMk m F τ x) = g (letterMk m F τ x)) : f = g

@[simp] theorem **letterMap_mk** (T : End LL D) (hT : ∀ c, T ∗ m c = m c ∗ T) (x : D) : letterMap m F τ T hT (letterMk m F τ x) = letterMk m F τ (T x)

theorem **finiteDimensional_letter** {ι : Type} [Fintype ι] (b : ι → D) (hb : ∀ x, ∃ c : ι → U, x = ∑ i, m (c i) (b i)) : FiniteDimensional F (Letter m F τ)

theorem **normV_eq_pow** (hw : w ∈ placesOver E l) : normV w = l ^ resDeg l w

theorem **Induces.valuation_lt_one_iff** (hj : Induces Bw) (x : Ω) : A.valuation x < 1 ↔ ‖Bw.j x‖ < 1

theorem **Induces.mem_placesOver** (hj : Induces Bw) (h : IsPlaceOver A w) : w ∈ placesOver E l

theorem **weilToGK_j** (σ : ɪ(Weil h)) (x : Ω) : (weilToGK Bw h σ : GQ l) (Bw.j x) = Bw.j ((σ : G E Ω) x)

theorem **frobUr_pow_root** (n : ℕ) (c : ɪ(MaximalUnramifiedExtension l)) (m : ℕ) (h0 : 0 < m) (hm : ¬ l ∣ m) (hc : c ^ m = 1) : (frobUr l ^ n) c = c ^ l ^ n

theorem **weilToGK_frobUr** (hj : Induces Bw) (σ : ɪ(Weil h)) {a b : ℕ} (hσ : ActsAs A (normV w) (σ : G E Ω) a b) (c : ɪ(MaximalUnramifiedExtension l)) : (weilToGK Bw h σ : GQ l) (((frobUr l ^ (resDeg l w * b)) c : ɪ(MaximalUnramifiedExtension l)) : Om l) = (((frobUr l ^ (resDeg l w * a)) c : ɪ(MaximalUnramifiedExtension l)) : Om l)

@[simp] theorem **coe_phiU** : (phiU P ρ : End LL ɪ(Dpst P ρ)) = phiD P ρ

theorem **commute_actD_phiU** (g : ɪ(GK K)) (n : ℤ) : Commute (actD P ρ g) ((phiU P ρ ^ n : (End LL ɪ(Dpst P ρ))ˣ) : End LL ɪ(Dpst P ρ))

theorem **phiD_pow_mulD** (n : ℕ) (c : ɪ(MaximalUnramifiedExtension l)) : phiD P ρ ^ n * mulD P ρ c = mulD P ρ ((frobUr l ^ n) c) * phiD P ρ ^ n

theorem **monD_phiU_zpow** (n : ℤ) : monD P ρ * ((phiU P ρ ^ n : (End LL ɪ(Dpst P ρ))ˣ) : End LL ɪ(Dpst P ρ)) = (l : LL) ^ n • (((phiU P ρ ^ n : (End LL ɪ(Dpst P ρ))ˣ) : End LL ɪ(Dpst P ρ)) * monD P ρ)

theorem **weilD_apply** (σ : ɪ(Weil h)) : weilD Bw h P ρ σ = actD P ρ (weilToGK Bw h σ) * ((phiU P ρ ^ (-alpha l σ) : (End LL ɪ(Dpst P ρ))ˣ) : End LL ɪ(Dpst P ρ))

theorem **weilD_ineIncl** (σ : ɪ(Ine (F

theorem **weilD_monD** (hj : Induces Bw) (σ : ɪ(Weil h)) : weilD Bw h P ρ σ * monD P ρ = ((normV w : ℕ) : LL) ^ deg σ • (monD P ρ * weilD Bw h P ρ σ)

theorem **weilD_mulD** (hj : Induces Bw) (σ : ɪ(Weil h)) (c : ɪ(MaximalUnramifiedExtension l)) : weilD Bw h P ρ σ * mulD P ρ c = mulD P ρ c * weilD Bw h P ρ σ

theorem **isSmooth_weilD** (hj : Induces Bw) : IsSmooth (weilD Bw h P ρ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FiniteDimensionalWdSpace

@[simp] theorem **wdMap_mk** (T : End ɪLL.1 ɪ(Dpst P ρ)) (hT : ∀ c, T * mulD P ρ c = mulD P ρ c * T) (x : ɪ(Dpst P ρ)) : wdMap P ι ρ T hT (wdMk P ι ρ x) = wdMk P ι ρ (T x)

@[simp] theorem **wdMk_zero** : wdMk P ι ρ 0 = 0

theorem **wdMk_add** (x y : ɪ(Dpst P ρ)) : wdMk P ι ρ (x + y) = wdMk P ι ρ x + wdMk P ι ρ y

theorem **wdMk_smul** (a : ɪLL.1) (x : ɪ(Dpst P ρ)) : wdMk P ι ρ (a • x) = ι (a : Om l) • wdMk P ι ρ x

theorem **wdMk_mul** (c : ɪ(MaximalUnramifiedExtension l)) (x : ɪ(Dpst P ρ)) : wdMk P ι ρ (mulD P ρ c x) = ι (c : Om l) • wdMk P ι ρ x

theorem **span_wdMk** : Submodule.span ℂ (Set.range (wdMk P ι ρ)) = ⊤

theorem **wd_ext** {f g : End ℂ (WDSpace P ι ρ)} (h : ∀ x, f (wdMk P ι ρ x) = g (wdMk P ι ρ x)) : f = g

@[simp] theorem **wdN_mk** (x : ɪ(Dpst P ρ)) : wdN P ι ρ (wdMk P ι ρ x) = wdMk P ι ρ (monD P ρ x)

theorem **isNilpotent_wdN** [FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)] : IsNilpotent (wdN P ι ρ)

theorem **IsPst.finite** [FiniteDimensional ɪLL.1 V] (hV : IsPst P ρ) : FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)

@[simp] theorem **lpDim_trivWD** (a : ℕ) : (trivWD h a).lpDim = a

@[simp] theorem **wdRho_mk** (σ : ɪ(Weil h)) (x : ɪ(Dpst P ρ)) : wdRho P ι Bw h hj ρ σ (wdMk P ι ρ x) = wdMk P ι ρ (weilD Bw h P ρ σ x)

theorem **isSmooth_wdRho** : IsSmooth (wdRho P ι Bw h hj ρ)

theorem **wdRho_wdN** (σ : ɪ(Weil h)) : wdRho P ι Bw h hj ρ σ * wdN P ι ρ = ((normV w : ℕ) : ℂ) ^ deg σ • (wdN P ι ρ * wdRho P ι Bw h hj ρ σ)

theorem **wdRep_spec** : ∃ u : (End ℂ (WDSpace P ι ρ))ˣ, (wdRep P ι Bw h hj ρ).N = wdN P ι ρ ∧ IsNilpotent ((u : End ℂ (WDSpace P ι ρ)) - 1) ∧ Commute (u : End ℂ (WDSpace P ι ρ)) (wdN P ι ρ) ∧ (∀ σ, Commute (u : End ℂ (WDSpace P ι ρ)) (wdRho P ι Bw h hj ρ σ)) ∧ ∀ σ, wdRho P ι Bw h hj ρ σ = (wdRep P ι Bw h hj ρ).ρ σ * ((u ^ deg σ : (End ℂ (WDSpace P ι ρ))ˣ) : End ℂ (WDSpace P ι ρ))

@[simp] theorem **wdRep_N** : (wdRep P ι Bw h hj ρ).N = wdN P ι ρ

theorem **wdRep_ineIncl** (σ : ɪ(Ine (F

theorem **wdPst_of_isPst** (hV : IsPst P ρ) : wdPst P ι Bw h hj ρ = haveI

theorem **wdPst_of_not_isPst** (hV : ¬ IsPst P ρ) : wdPst P ι Bw h hj ρ = trivWD h (finrank ɪLL.1 V)

theorem **lpDim_wdPst** (hV : IsPst P ρ) : (wdPst P ι Bw h hj ρ).lpDim = finrank ℂ (WDSpace P ι ρ)

theorem **mono0_wdPst** (hV : IsPst P ρ) : (wdPst P ι Bw h hj ρ).mono0 ↔ wdN P ι ρ = 0

theorem **unr_wdPst** (hV : IsPst P ρ) : (wdPst P ι Bw h hj ρ).unr ↔ wdN P ι ρ = 0 ∧ ∀ (σ : ɪ(Ine (F

theorem **IsCrysAt.isPotCrysAt** (hr : IsCrysAt P Bw r) : IsPotCrysAt P Bw r

theorem **IsModel.unique** {LL' : FinExt l} {r₁ r₂ : ɪGw →* GL (Fin a) ɪLL'.1} (h₁ : IsModel Bw r LL' r₁) (h₂ : IsModel Bw r LL' r₂) : r₁ = r₂

theorem **isModel_one** : IsModel Bw (1 : ɪGw →* GL (Fin a) (Om l)) Bw.LL 1

theorem **wdAt_of_isPstAt** (hr : IsPstAt P Bw r) : ∃ LL' r', IsModel Bw r LL' r' ∧ IsPst P (Bw.repE LL' r') ∧ wdAt P ι Bw h hj r = wdPst P ι Bw h hj (Bw.repE LL' r')

theorem **wdAt_of_not_isPstAt** (hr : ¬ IsPstAt P Bw r) : wdAt P ι Bw h hj r = trivWD h a

theorem **isJumpsW_jumpsW** {τ : E →+* Om l} (hJ : ∃ J, IsJumpsW P pl br r τ J) : IsJumpsW P pl br r τ (jumpsW P pl br r τ)

theorem **isCrysW_iff** {w : HeightOneSpectrum (𝒪 E)} (hw : w ∈ placesOver E l) : IsCrysW P pl br r w ↔ IsCrysAt P (br w hw) (resDec pl r w)

theorem **isPotCrysW_iff** {w : HeightOneSpectrum (𝒪 E)} (hw : w ∈ placesOver E l) : IsPotCrysW P pl br r w ↔ IsPotCrysAt P (br w hw) (resDec pl r w)

theorem **isdRW_iff** {w : HeightOneSpectrum (𝒪 E)} (hw : w ∈ placesOver E l) : IsdRW P pl br r w ↔ IsdRAt P (br w hw) (resDec pl r w)

theorem **tInv_mul** : tInv P ∗ P.tB = 1

theorem **smul_actQ_tInv** (g : GQ l) : (P.χQ g : ℚ_[l]) • P.actQ g (tInv P) = tInv P

theorem **smul_phiB_tInv** : (l : ℚ_[l]) • P.φB (tInv P) = tInv P

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FiniteTwistingNotIsGenerator

theorem **finiteTwisting_lineBundle** (R : Type) [CommRing R] (N : ℕ) : (finiteCoverTwistingSheaf R N).lb = GlobalTheory.ProjSpace.O1 R N

theorem **finiteTwisting_not_isGenerator** (R : Type) [CommRing R] [Nontrivial R] (N : ℕ) (t : (finiteCoverTwistingSheaf R (N + 1)).lb.Gen) : ¬ t.IsGen

theorem **placeW_spec** (h : JcPin D p la jc) : (∀ x : D.what₂.LL, D.what₂.embLL x ∈ placeW D p la jc ↔ la.1.valuation D.what₂.LL x ≤ 1) ∧ ∀ a : DV.KW p, jc a ∈ placeW D p la jc ↔ a ∈ (algebraMap (WittVector p (AlgebraicClosure (ZMod p))) (DV.KW p)).range

theorem **placeW_nonunit** (h : JcPin D p la jc) : (p : ℂ) ∈ (placeW D p la jc).nonunits

theorem **exists_placeW_nonunit** (D : D7) (p : ℕ) [Fact p.Prime] (la : PlL D p) : ∃ jc : DV.KW p →+* ℂ, JcPin D p la jc ∧ (p : ℂ) ∈ (placeW D p la jc).nonunits

theorem **valF_unit_ne** {F : Type} [Field F] {X : Scheme.{0}} (x : Sp F ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (f : Γ(X, U)ˣ) : valF x U f.val ≠ 0

theorem **gen_mem** {Z : Mdl Y i₀} (y : Z.Pt) (a : Fin Z.m) (φ : Γ(Z.X, (Z.U a).1) →+* ɪy.C.S) (hφ : spMap φ ≫ Z.chart a = y.gen) {V : Z.X.Opens} (hV : ⊤ ≤ y.z ⁻¹ᵁ V) : ⊤ ≤ y.gen ⁻¹ᵁ ((Z.U a).1 ⊓ V)

theorem **norm_r** {Z : Mdl Y i₀} {b : Bool} {st : ((E →+* ℂ) × ℕ) ⊕ ℕ} {x : Letter Dw} (M : LineModel Z b st x) (y : Z.Pt) (a : Fin Z.m) (i : Fin M.L.n) (φ : Γ(Z.X, (Z.U a).1) →+* ɪy.C.S) (hφ : spMap φ ≫ Z.chart a = y.gen) (hi : ⊤ ≤ y.z ⁻¹ᵁ M.L.U i) : ‖valF y.gen _ (M.r a i).val‖ = Real.exp (− lnStep (Y Z.i).t (y.gen ≫ Z.π) ((Z.fr a).bc φ) p b st)

theorem **norm_r_eq** {Z : Mdl Y i₀} {b : Bool} {st : ((E →+* ℂ) × ℕ) ⊕ ℕ} {x : Letter Dw} (M M' : LineModel Z b st x) (y : Z.Pt) (a : Fin Z.m) (i : Fin M.L.n) (i' : Fin M'.L.n) (φ : Γ(Z.X, (Z.U a).1) →+* ɪy.C.S) (hφ : spMap φ ≫ Z.chart a = y.gen) (hi : ⊤ ≤ y.z ⁻¹ᵁ M.L.U i) (hi' : ⊤ ≤ y.z ⁻¹ᵁ M'.L.U i') : ‖valF y.gen _ (M.r a i).val‖ = ‖valF y.gen _ (M'.r a i').val‖

theorem **exists_member** {Z : Mdl Y i₀} (L : FinLB Z.X) (y : Z.Pt) : ∃ i, ⊤ ≤ y.z ⁻¹ᵁ L.U i

theorem **n_eq** (hp : p.Prime) (z z' : admitted Y i₀) (h : z.Z.i.l.A = z'.Z.i.l.A) : z.n = z'.n

theorem **letterOf_inl** (x : Letter Dw) : letterOf Dw (.inl (x.1, x.2)) = some x

theorem **exponent_three** (p : ℕ) (hp : p.Prime) : 1 + 1 / ((p : ℝ) − 1) < (3 : ℕ)

theorem **exponent_two** : ¬ (1 + 1 / (((2 : ℕ) : ℝ) − 1) < (2 : ℕ))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FinrankEqZeroIff

theorem **evAt_fromSpecStalk** {X : Scheme.{0}} (x : X) {B : Type} [CommRing B] (f : X.presheaf.stalk x →+* B) {U : X.Opens} (hU : x ∈ U) (h : ⊤ ≤ (spMap f ≫ X.fromSpecStalk x) ⁻¹ᵁ U) (s : Γ(X, U)) : evAt (spMap f ≫ X.fromSpecStalk x) h s = f ((X.presheaf.germ U x hU).hom s)

theorem **base_ext** {C : Type} [NormedField C] [IsUltrametricDist C] (i : Idx H p) (x x' : Sp ɪ(intO C) ⟶ (Y i).X) (h : spMap (intO C).subtype ≫ x = spMap (intO C).subtype ≫ x') : x = x'

theorem **abP_red_iff** {Z₁ Z₂ : Mdl Y i₀} {C : SCF p} (y₁ : PtAt Z₁ C) (y₂ : PtAt Z₂ C) (h : NatImg Z₁ Z₂ y₁.gen y₂.gen) : Z₁.abP y₁.red ↔ Z₂.abP y₂.red

theorem **exists_isTrans** (hN : N2 Y) (i i' : Idx H p) (hl : i.l.Dvd i'.l) (hc : i.c.F ≤ i'.c.F) : ∃ mu, IsTrans i i' (hN.2.choose i i' hl hc) mu

theorem **corr_base** (Z : ct.Mdl) : Corr Z ct.Xm

theorem **abP_redK_iff** (Z : ct.Mdl) (y : DVP ct) : ct.abP (redK Z y) ↔ ct.abP (redK ct.Xm y)

theorem **finrank_eq_zero_iff** {Z : ct.Mdl} (x : ct.Pt Z) : Module.finrank (𝒪 D.host.E) (ct.Ylat x) = 0 ↔ Subsingleton (ct.Ylat x)

theorem **ptK_good** (hg : GoodIff ct) (Z : ct.Mdl) (y : DVP ct) : Subsingleton (ct.Ylat (ptK Z y)) ↔ dvRho y = 0

theorem **dvRho_pos_iff** (hg : GoodIff ct) (Z : ct.Mdl) (y : DVP ct) : 0 < dvRho y ↔ ¬ ct.abP (redK Z y)

theorem **imgU_inQ** (χ : Chart Z y e) : ct.inQ (imgU χ)

theorem **chAt_imgM** (χ : Chart Z y e) (h : chAt χ) : imgM χ = redK Z y

theorem **chAbExact_imgU** (χ : Chart Z y e) (h : chAbExact χ) : ¬ ct.abP (imgU χ)

theorem **goodIff_of** (h₁ : AbOfTrivSt) (h₂ : TrivOfAbSt) (ct : At S κ p) : GoodIff ct

theorem **genEx_of** (h₁ : StalkSt) (h₂ : FieldSt) (h₃ : ExtendSt) (h₄ : FamSt) (ct : At S κ p) : GenEx ct

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FinrankLineHyperplane

theorem **HasLatFlags.isTr_tr** (q q' : LatChart Z L) (e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ) : q.IsTr q' e (q.tr q' e).val

theorem **HasLatFlags.isLB** (e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ) : (latLB Z L e).IsLB

theorem **HasLatFlags.degHom_apply** {k : Type} [Field k] (c : Curve k Z.X) (e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ) : hL.degHom hp hW c e = c.degLat L e

theorem **IsFlagCl.hasLatFlags** {Z' : Mdl Y i₀} {L : Lat Z} (h : IsFlagCl Z (allSteps L) Z') (hty : ∀ σ, Z.ty σ = ∅) : HasLatFlags Z' L

theorem **IsHTBlowup.ty_eq** (H : IsHTBlowup ct Z (ct.Xn n)) (σ : D.host.E →+* ℂ) : Z.ty σ = ∅

theorem **htFam_spec** (h : ∃ L : Lat Z, IsHTLatOf (htmTrue ct) Z L) : IsHTLatOf (htmTrue ct) Z (htLat ct Z) ∧ (htFam ct Z).1 = allSteps (htLat ct Z)

theorem **hasLatFlags** (h : ∃ L : Lat Z, IsHTLatOf (htmTrue ct) Z L) : HasLatFlags (flagMdl ct Z) (htLat ct Z)

theorem **degOf_wtLB** {k : Type} [Field k] (c : Curve k (flagMdl ct Z).X) (l : PsiT D → ℕ → ℤ) : c.degOf (wtLB ct Z l) = c.degLat (htLat ct Z) (ct.expo l)

theorem **degGr_eq** (c : Curve ct.Fbar (flagMdl ct Z).X) (τ : PsiT D) (i : ℕ) : (crhTrue ct).degGr (Z

theorem **finrank_line_hyperplane** (l : ct.Sum1) (τ : PsiT D) : Module.finrank ct.kbar (l.bar.submoduleOf (ct.eigBar τ)) = 1 ∧ Module.finrank ct.kbar ((ct.psiBar.orthogonal l.bar).submoduleOf (ct.eigBar τ)) + 1 = N

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FinrankRiemannRochNsmulHyperplane

theorem **hdim_def** (i : ℕ) (Δ : Y.Dv) : Y.hdim i Δ = Module.finrank F (Y.H i Δ)

theorem **hdim_eq_zero_iff** (i : ℕ) (Δ : Y.Dv) : Y.hdim i Δ = 0 ↔ Subsingleton (Y.H i Δ)

theorem **hdim_zero** (Δ : Y.Dv) : Y.hdim 0 Δ = Module.finrank F (Y.H0 Δ)

theorem **subsingleton_H** {i : ℕ} (hi : Y.dim < i) (Δ : Y.Dv) : Subsingleton (Y.H i Δ)

theorem **hdim_eq_zero_of_lt** {i : ℕ} (hi : Y.dim < i) (Δ : Y.Dv) : Y.hdim i Δ = 0

theorem **chi_eq_sum** (Δ : Y.Dv) : Y.chi Δ = ∑ i ∈ Finset.range (Y.dim + 1), (-1 : ℤ) ^ i * Y.hdim i Δ

theorem **eulerChar_eq_finrank_riemannRoch** {Δ : Y.Dv} (h : ∀ i, 0 < i → Y.hdim i Δ = 0) : Y.chi Δ = Module.finrank F (Y.H0 Δ)

theorem **hdim_eq_of_toPic_eq** {Δ : Y.Dv} {L : LB Y.X} (hL : L.IsLB) (e : toPic Δ = cls L hL) (i : ℕ) : Y.hdim i Δ = GlobalTheory.hdim F (twist (structureModule Y.X) L) i

theorem **hdim_congr** {Δ Δ' : Y.Dv} (e : toPic Δ = toPic Δ') (i : ℕ) : Y.hdim i Δ = Y.hdim i Δ'

theorem **hdim_add_principal** (i : ℕ) (Δ : Y.Dv) (g : Y.Fnˣ) : Y.hdim i (Δ + principal g) = Y.hdim i Δ

theorem **chi_eq_chiPic** (Δ : Y.Dv) : Y.chi Δ = chiPic F (structureModule Y.X) (toPic Δ)

theorem **chi_zero** : Y.chi 0 = GlobalTheory.chi F (structureModule Y.X)

theorem **isAffineOpen_stdOpen** (j : Fin (Y.N + 1)) : IsAffineOpen (Y.stdOpen j)

theorem **iSup_stdOpen** : ⨆ j, Y.stdOpen j = ⊤

theorem **isAffineOpen_biInf_stdOpen** (s : Finset (Fin (Y.N + 1))) (hs : s.Nonempty) : IsAffineOpen (⨅ j ∈ s, Y.stdOpen j)

theorem **hdim_eq_cech** (i : ℕ) (Δ : Y.Dv) : Y.hdim i Δ = Module.finrank F ((Y.cechCx Δ).homology i)

theorem **inter_eq_multiDiff** (Δ : Fin Y.dim → Y.Dv) : Y.inter Δ = multiDiff Δ Y.chi 0

theorem **inter_eq_sum** (Δ : Fin Y.dim → Y.Dv) : Y.inter Δ = ∑ S : Finset (Fin Y.dim), (-1 : ℤ) ^ S.card * Y.chi (-∑ i ∈ S, Δ i)

theorem **inter_eq_of_toPic_eq** {Δ : Fin Y.dim → Y.Dv} {L : Fin Y.dim → LB Y.X} (hL : ∀ i, (L i).IsLB) (e : ∀ i, toPic (Δ i) = cls (L i) (hL i)) : Y.inter Δ = interNum F (structureModule Y.X) L

theorem **inter_congr** {Δ Δ' : Fin Y.dim → Y.Dv} (e : ∀ i, toPic (Δ i) = toPic (Δ' i)) : Y.inter Δ = Y.inter Δ'

theorem **inter_eq_zero_of_principal** (Δ : Fin Y.dim → Y.Dv) (i : Fin Y.dim) (g : Y.Fnˣ) (h : Δ i = principal g) : Y.inter Δ = 0

theorem **inter_comp_perm** (Δ : Fin Y.dim → Y.Dv) (σ : Equiv.Perm (Fin Y.dim)) : Y.inter (Δ ∘ σ) = Y.inter Δ

theorem **inter_const** (Δ : Y.Dv) : Y.inter (fun _ => Δ) = topSelfIntersection F (Y.lb Δ) Y.dim

theorem **del_linSys** {Δ : Y.Dv} {n : ℕ} (b : Fin n → Y.H0 Δ) : (Y.linSys b).del = Y.inter fun _ => Δ

theorem **diffVanish_chi** : DiffVanish (Y.dim + 1) Y.chi

@[simp] theorem **interForm_apply** (Δ : Fin Y.dim → Y.Dv) : Y.interForm Δ = Y.inter Δ

theorem **inter_update_add** (Δ : Fin Y.dim → Y.Dv) (i : Fin Y.dim) (A B : Y.Dv) : Y.inter (Function.update Δ i (A + B)) = Y.inter (Function.update Δ i A) + Y.inter (Function.update Δ i B)

theorem **inter_update_zsmul** (Δ : Fin Y.dim → Y.Dv) (i : Fin Y.dim) (n : ℤ) (A : Y.Dv) : Y.inter (Function.update Δ i (n • A)) = n * Y.inter (Function.update Δ i A)

theorem **inter_zsmul** (n : Fin Y.dim → ℤ) (Δ : Fin Y.dim → Y.Dv) : Y.inter (fun i => n i • Δ i) = (∏ i, n i) * Y.inter Δ

theorem **inter_const_nsmul** (n : ℕ) (Δ : Y.Dv) : Y.inter (fun _ => n • Δ) = (n : ℤ) ^ Y.dim * Y.inter fun _ => Δ

theorem **exists_polynomial_chi** (A Δ : Y.Dv) : ∃ p : Polynomial ℚ, p.natDegree ≤ Y.dim ∧ (∀ n : ℤ, p.eval (n : ℚ) = Y.chi (A + n • Δ)) ∧ p.coeff Y.dim = (Y.dim.factorial : ℚ)⁻¹ * Y.inter fun _ => Δ

theorem **finiteCoh_sheafOn** (Z : Y.Cl) (Δ : Y.Dv) : FiniteCoh F (Y.sheafOn Z Δ)

theorem **chiOn_eq_chi_pull** (Z : Y.Cl) (Δ : Y.Dv) : letI

theorem **interOn_eq_multiDiff** (Z : Y.Cl) {d : ℕ} (Δ : Fin d → Y.Dv) : Y.interOn Z Δ = multiDiff Δ (Y.chiOn Z) 0

theorem **interOn_eq_sum** (Z : Y.Cl) {d : ℕ} (Δ : Fin d → Y.Dv) : Y.interOn Z Δ = ∑ S : Finset (Fin d), (-1 : ℤ) ^ S.card * Y.chiOn Z (-∑ i ∈ S, Δ i)

theorem **interOn_congr** (Z : Y.Cl) {d : ℕ} {Δ Δ' : Fin d → Y.Dv} (e : ∀ i, toPic (Δ i) = toPic (Δ' i)) : Y.interOn Z Δ = Y.interOn Z Δ'

theorem **interOn_comp_perm** (Z : Y.Cl) {d : ℕ} (Δ : Fin d → Y.Dv) (σ : Equiv.Perm (Fin d)) : Y.interOn Z (Δ ∘ σ) = Y.interOn Z Δ

theorem **chiOn_top** (Δ : Y.Dv) : Y.chiOn ⊤ Δ = Y.chi Δ

theorem **interOn_top_eq_interNum** {d : ℕ} (Δ : Fin d → Y.Dv) : Y.interOn ⊤ Δ = interNum F (structureModule Y.X) fun i => Y.lb (Δ i)

theorem **interOn_top** (Δ : Fin Y.dim → Y.Dv) : Y.interOn ⊤ Δ = Y.inter Δ

theorem **interOn_top_cons** {A : Y.Dv} (f : Y.H0 A) (hf : (f : Y.Fn) ≠ 0) {d : ℕ} (Δ : Fin d → Y.Dv) : Y.interOn ⊤ (Fin.cons A Δ : Fin (d + 1) → Y.Dv) = interNumSub F (ThickeningDetermines.secIdeal (isLB_toLB A) (sec f)) fun i => Y.lb (Δ i)

theorem **diffVanish_chiOn** {Z : Y.Cl} {d : ℕ} (hd : topologicalKrullDim (Z : Set Y.X) ≤ d) : DiffVanish (d + 1) (Y.chiOn Z)

theorem **interOn_update_add** {Z : Y.Cl} {d : ℕ} (hd : topologicalKrullDim (Z : Set Y.X) ≤ d) (Δ : Fin d → Y.Dv) (i : Fin d) (A B : Y.Dv) : Y.interOn Z (Function.update Δ i (A + B)) = Y.interOn Z (Function.update Δ i A) + Y.interOn Z (Function.update Δ i B)

theorem **interOn_eq_zero** {Z : Y.Cl} {d : ℕ} (hd : topologicalKrullDim (Z : Set Y.X) ≤ d) (Δ : Fin (d + 1) → Y.Dv) : Y.interOn Z Δ = 0

theorem **isCurve_iff** (Z : Y.Cl) : Y.IsCurve Z ↔ IsCompleteCurve F Z

theorem **isCurve_top** (h : Y.dim = 1) : Y.IsCurve ⊤

theorem **IsCurve.eq_top** (h : Y.dim = 1) {C : Y.Cl} (hC : Y.IsCurve C) : C = ⊤

theorem **deg_eq_interOn** (C : Y.Cl) (Δ : Y.Dv) : Y.deg C Δ = Y.interOn C ![Δ]

theorem **deg_eq** (C : Y.Cl) (Δ : Y.Dv) : Y.deg C Δ = Y.chiOn C 0 - Y.chiOn C (-Δ)

theorem **deg_top** (h : Y.dim = 1) (Δ : Y.Dv) : Y.deg ⊤ Δ = Y.inter fun _ => Δ

theorem **deg_eq_of_toPic_eq** (C : Y.Cl) {Δ : Y.Dv} {L : LB Y.X} (hL : L.IsLB) (e : toPic Δ = cls L hL) : Y.deg C Δ = interNumRed F C ![L]

theorem **deg_congr** (C : Y.Cl) {Δ Δ' : Y.Dv} (e : toPic Δ = toPic Δ') : Y.deg C Δ = Y.deg C Δ'

theorem **deg_principal** (C : Y.Cl) (g : Y.Fnˣ) : Y.deg C (principal g) = 0

@[simp] theorem **deg_zero** (C : Y.Cl) : Y.deg C 0 = 0

theorem **deg_add** (A B : Y.Dv) : Y.deg C (A + B) = Y.deg C A + Y.deg C B

theorem **deg_neg** (Δ : Y.Dv) : Y.deg C (-Δ) = -Y.deg C Δ

theorem **deg_sub** (A B : Y.Dv) : Y.deg C (A - B) = Y.deg C A - Y.deg C B

theorem **deg_zsmul** (n : ℤ) (Δ : Y.Dv) : Y.deg C (n • Δ) = n * Y.deg C Δ

theorem **deg_nsmul** (n : ℕ) (Δ : Y.Dv) : Y.deg C (n • Δ) = n * Y.deg C Δ

theorem **deg_eq** (Δ : Y.Dv) : Y.deg C Δ = Y.chiOn C Δ - Y.chiOn C 0

theorem **deg_nonneg_of_generatesAt** {Δ : Y.Dv} {f : Y.Fn} (hf : f ∈ Y.H0 Δ) {x : Y.X} (hx : x ∈ C) (h : Δ.GeneratesAt x f) : 0 ≤ Y.deg C Δ ∧ ((∃ y ∈ C, Δ.VanishesAt y f) → 0 < Y.deg C Δ)

theorem **deg_nonneg_of_eff** {Δ : Y.Dv} (h : Y.Eff Δ) (hs : ¬ Y.subCl C (Y.supp Δ)) : 0 ≤ Y.deg C Δ ∧ ((∃ y ∈ C, y ∈ Y.supp Δ) → 0 < Y.deg C Δ)

theorem **nef_iff** : Y.nef Δ ↔ ∀ C, Y.IsCurve C → 0 ≤ Y.deg C Δ

theorem **finrank_riemannRoch_nsmul** (Δ : Y.Dv) (n : ℕ) : Module.finrank F (Y.H0 (n • Δ)) = lbH0 F (lbPow (Y.lb Δ) n)

theorem **big_iff** : Y.big Δ ↔ ∃ c : ℝ, 0 < c ∧ ∃ᶠ n : ℕ in atTop, c * (n : ℝ) ^ Y.dim ≤ Module.finrank F (Y.H0 (n • Δ))

theorem **nef_iff_of_toPic_eq** (hL : L.IsLB) (e : toPic Δ = cls L hL) : Y.nef Δ ↔ IsNef F L

theorem **big_iff_of_toPic_eq** (hL : L.IsLB) (e : toPic Δ = cls L hL) : Y.big Δ ↔ IsBig F L

theorem **ample_iff_of_toPic_eq** (hL : L.IsLB) (e : toPic Δ = cls L hL) : Y.ample Δ ↔ IsAmpleLB L

theorem **nef_congr** (e : toPic A = toPic B) : Y.nef A ↔ Y.nef B

theorem **big_congr** (e : toPic A = toPic B) : Y.big A ↔ Y.big B

theorem **ample_congr** (e : toPic A = toPic B) : Y.ample A ↔ Y.ample B

theorem **nef_zero** : Y.nef 0

theorem **nef.add** (hA : Y.nef A) (hB : Y.nef B) : Y.nef (A + B)

theorem **nef.nsmul** (h : Y.nef Δ) (n : ℕ) : Y.nef (n • Δ)

theorem **nef_nsmul_iff** {n : ℕ} (hn : 1 ≤ n) : Y.nef (n • Δ) ↔ Y.nef Δ

theorem **bpf.nef** {V : Set Y.Fn} (h : Y.bpf Δ V) : Y.nef Δ

theorem **nef_iff_of_dim_eq_one** (h : Y.dim = 1) : Y.nef Δ ↔ 0 ≤ Y.inter fun _ => Δ

theorem **ample.deg_pos** (h : Y.ample A) {C : Y.Cl} (hC : Y.IsCurve C) : 0 < Y.deg C A

theorem **ample.nef** (h : Y.ample A) : Y.nef A

theorem **ample.eventually_hdim_eq_zero** (h : Y.ample A) (Δ : Y.Dv) : ∀ᶠ n : ℕ in atTop, ∀ i, 0 < i → Y.hdim i (Δ + n • A) = 0

theorem **ample_hyp** {i : Fin (Y.N + 1)} (hi : Y.ι (genericPoint Y.X) ∈ ProjSpace.chart F Y.N i) : Y.ample (Y.hyp hi)

theorem **hasNonzeroSec_iff** : HasNonzeroSec (Y.lb Δ) ↔ Y.H0 Δ ≠ ⊥

theorem **big.isBig_add_of_riemannRoch_ne_bot** (hA : Y.big A) (hB : Y.H0 B ≠ ⊥) : Y.big (A + B)

theorem **big.add_eff** (hA : Y.big A) (hB : Y.Eff B) : Y.big (A + B)

theorem **big_nsmul_iff** {n : ℕ} (hn : 1 ≤ n) : Y.big (n • Δ) ↔ Y.big Δ

theorem **big.isBig_frequently_riemannRoch_ne_bot** (h : Y.big Δ) : ∃ᶠ n : ℕ in atTop, Y.H0 (n • Δ) ≠ ⊥

theorem **not_big_zero** (h : 1 ≤ Y.dim) : ¬ Y.big 0

theorem **hdim_zsmul_projHyp** (i : ℕ) (m : ℤ) : (proj F n).hdim i (m • projHyp F n) = GlobalTheory.hdim F (ProjCoh.sheafO F n m) i

theorem **chi_zsmul_projHyp** (m : ℤ) : (proj F n).chi (m • projHyp F n) = Ring.choose (m + n) n

theorem **finrank_riemannRoch_nsmul_hyperplane** (m : ℕ) : Module.finrank F ((proj F n).H0 (m • projHyp F n)) = (m + n).choose n

theorem **hdim_top_zsmul_projHyp** (hn : 1 ≤ n) (m : ℤ) : (proj F n).hdim n (m • projHyp F n) = (-m - 1).toNat.choose n

theorem **inter_projHyp** : (proj F n).inter (fun _ => projHyp F n) = 1

theorem **inter_zsmul_projHyp** (d : Fin n → ℤ) : (proj F n).inter (fun i => d i • projHyp F n) = ∏ i, d i

theorem **exists_inter_eq_one** : ∃ (Y : Var F) (Δ : Y.Dv), Y.dim = n ∧ Y.inter (fun _ => Δ) = 1

theorem **ample_projHyp** : (proj F n).ample (projHyp F n)

theorem **nef_projHyp** : (proj F n).nef (projHyp F n)

theorem **big_projHyp** : (proj F n).big (projHyp F n)

theorem **deg_projHyp_line** : (proj F 1).deg ⊤ (projHyp F 1) = 1

theorem **deg_zsmul_projHyp_line** (m : ℤ) : (proj F 1).deg ⊤ (m • projHyp F 1) = m

theorem **nef_zsmul_projHyp_line** (m : ℤ) : (proj F 1).nef (m • projHyp F 1) ↔ 0 ≤ m

theorem **big_zsmul_projHyp** (hn : 1 ≤ n) (m : ℤ) : (proj F n).big (m • projHyp F n) ↔ 0 < m

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FirstIndexEqOfUnfolded

theorem **flagConst_pos** (r : ℕ) : 0 < flagConst r

theorem **flagConst_add** (p q : ℕ) : flagConst (p + q) = grassConst p q * flagConst p * flagConst q

theorem **isOrth_x0F** (X : Data E r) : IsOrth X (x0F X)

theorem **muF_fundSet_le** (X : Data E r) {x0 x1 : Frame X.Ψ r} (ho : IsOrth X x0) (ho1 : IsOrth X x1) {FF CechHodgeFiltration : Set (V (nd X))} (hF : IsFundSet X x0 FF) (hF1 : IsFundSet X x1 CechHodgeFiltration) : muF X x0 FF ≤ muF X x1 CechHodgeFiltration

theorem **covol_eq** (X : Data E r) {x0 : Frame X.Ψ r} (ho : IsOrth X x0) {FF : Set (V (nd X))} (hF : IsFundSet X x0 FF) : muF X x0 FF < ⊤ ∧ covol X = flagConst r ^ X.Ψ.card * (muF X x0 FF).toReal

theorem **covol_pos** (X : Data E r) : 0 < covol X

theorem **idx_symm_inl** {r j : ℕ} (h : j ≤ r) (a : Fin j) : ((idx h).symm (Sum.inl a) : ℕ) = r - j + a

theorem **idx_symm_inr** {r j : ℕ} (h : j ≤ r) (a : Fin (r - j)) : ((idx h).symm (Sum.inr a) : ℕ) = a

theorem **reindex_sgn** {r j : ℕ} (h : j ≤ r) : reindex (idx h) (idx h) (diagonal fun a : Fin r => ((sgn (r − j) (a : ℕ) : ℝ) : ℂ)) = −Jpq j (r − j)

theorem **gram_eq_mul** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (x : Matrix (Fin r) (Fin r) ℂ) : (gram τ G x)ᵀ = xᴴ ∗ (G.map τ)ᵀ ∗ x

theorem **det_x0F** (τ : X.Ψ) : (mat (x0F X) τ).det ≠ 0

theorem **toUpq_val** (γ : GL (Fin r) E) (t : Tau) : (toUpq X hj ι γ t).1 = frameCo X hj (lact γ (x0F X)) (ι t)

theorem **frameCo_orth** (hp : X.p0 = r − j) {x : Frame X.Ψ r} (hx : IsOrth X x) (τ : X.Ψ) : (frameCo X hj x τ)ᴴ ∗ Jpq j (r − j) ∗ frameCo X hj x τ = Jpq j (r − j)

theorem **gammaU_le** [Fintype Tau] (hp : X.p0 = r − j) : gammaU X hj ι ≤ UpqI Tau j (r − j)

theorem **moeb_zero** {p q : ℕ} (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) : moeb g 0 = g.toBlocks₁₂ ∗ g.toBlocks₂₂⁻¹

theorem **frameCo_rmul_blocks** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (x : Frame X.Ψ r) (τ : X.Ψ) : ∃ B : Matrix (Fin (r − j)) (Fin (r − j)) ℂ, IsUnit B.det ∧ (frameCo X hj (rmul x b) τ).toBlocks₁₂ = (frameCo X hj x τ).toBlocks₁₂ ∗ B ∧ (frameCo X hj (rmul x b) τ).toBlocks₂₂ = (frameCo X hj x τ).toBlocks₂₂ ∗ B

theorem **domPt_rmul** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (x : Frame X.Ψ r) : domPt X hj ι (rmul x b) = domPt X hj ι x

theorem **zero_mem_hcDom** (p q : ℕ) : (0 : Matrix (Fin p) (Fin q) ℂ) ∈ hcDom p q

theorem **exists_upq_of_orth** (hp : X.p0 = r − j) {x : Frame X.Ψ r} (hx : IsOrth X x) (τ : X.Ψ) : ∃ g ∈ Upq j (r − j), g.1 = frameCo X hj x τ

theorem **domPt_mem** (hp : X.p0 = r − j) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : domPt X hj ι x ∈ DomI Tau j (r − j)

theorem **frameCo_isUnit₂₂** (hp : X.p0 = r − j) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (τ : X.Ψ) : IsUnit (frameCo X hj x τ).toBlocks₂₂.det

theorem **domPt_lact** [Fintype Tau] (hp : X.p0 = r − j) (γ : GL (Fin r) E) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : domPt X hj ι (lact γ x) = moebI (toUpq X hj ι γ) (domPt X hj ι x)

theorem **hodgeC_pos** (t : Tau) : 0 < hodgeC X ι t

theorem **psi_nonempty** : Nonempty X.Ψ

theorem **sum_nonempty** {r j : ℕ} (hj : j ≤ r) (hr : 0 < r) : Nonempty (Fin j ⊕ Fin (r − j))

theorem **ptOfFrame_eq** (k : Comp) (x x' : (Y k).dom) {γ : GL (Fin r) E} (hγ : γ ∈ (Y k).Γ) {b : Frame (Y k).Ψ r} (hb : b ∈ Bor (Y k).Ψ r) (e : x'.1 = rmul (lact γ x.1) b) : ptOfFrame hj Y hr hp κ B s k x = ptOfFrame hj Y hr hp κ B s k x'

theorem **le_of_isUnf** : j ≤ r

theorem **first_index_eq_of_unfolded** (k : U.Comp) : (U.X k).p0 = r − j

theorem **ptOfFrame_eq_of_frBor** {k k' : U.Comp} (x : (U.X k).dom) (x' : (U.X k').dom) (h : ∃ t : Ix Dw → (CAt σ)ˣ, FrBor t (U.un k x) (U.un k' x')) : ptOfFrame (le_of_isUnf U hU e) U.X hr (first_index_eq_of_unfolded U hU e) (cmEquiv U hU T) B s k x = ptOfFrame (le_of_isUnf U hU e) U.X hr (first_index_eq_of_unfolded U hU e) (cmEquiv U hU T) B s k' x'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FixedCoefficientFieldInComplex

theorem **rayField_iff** : RayField ↔ ∀ (K : Type) [Field K] [NumberField K] (q : Ideal (𝒪 K)), q ≠ ⊥ → ∃ (L : Type) (_ : Field L) (_ : NumberField L) (_ : Algebra K L), IsRayField K L q

theorem **exists_prime_congr** (e : L ≃ₐ[K] L') (p : Ideal (𝒪 K)) (σ : L ≃ₐ[K] L) (n : ℕ) (h : ∃ P : Ideal (𝒪 L), P.IsPrime ∧ P.LiesOver p ∧ ∀ x : 𝒪 L, σ • x − x ^ n ∈ P) : ∃ P : Ideal (𝒪 L'), P.IsPrime ∧ P.LiesOver p ∧ ∀ x : 𝒪 L', e.autCongr σ • x − x ^ n ∈ P

theorem **isFrob_congr** (e : L ≃ₐ[K] L') (p : HeightOneSpectrum (𝒪 K)) (σ : L ≃ₐ[K] L) : IsFrob L' p (e.autCongr σ) ↔ IsFrob L p σ

theorem **inInertia_congr** (e : L ≃ₐ[K] L') (p : HeightOneSpectrum (𝒪 K)) (σ : L ≃ₐ[K] L) : InInertia L' p (e.autCongr σ) ↔ InInertia L p σ

theorem **heckeVal_congr** [IsGalois K L] [IsGalois K L'] (e : L ≃ₐ[K] L') (χ : (L' ≃ₐ[K] L') →* ℂˣ) (p : HeightOneSpectrum (𝒪 K)) : heckeVal L (χ.comp e.autCongr.toMonoidHom) p = heckeVal L' χ p

theorem **IsRayField.of_algEquiv** {q : Ideal (𝒪 K)} (h : IsRayField K L q) (e : L ≃ₐ[K] L') : IsRayField K L' q

theorem **isRayField_rat_top** : IsRayField ℚ ℚ ⊤

theorem **not_isRayField_rat_three** : ¬ IsRayField ℚ ℚ (Ideal.span {3})

theorem **cycField_le_iff** {N : ℕ} {M : IntermediateField ℚ ℂ} : cycField N ≤ M ↔ rootOne N ∈ M

theorem **mem_cycField** {N : ℕ} (hN : N ≠ 0) {z : ℂ} (hz : z ^ N = 1) : z ∈ cycField N

theorem **cycField_mono** {N N' : ℕ} (hN' : N' ≠ 0) (h : N ∣ N') : cycField N ≤ cycField N'

theorem **mem_imField** {z : ℂ} : z ∈ imField φ ↔ ∃ x, φ x = z

theorem **imField_le_iff** {M : IntermediateField ℚ ℂ} : imField φ ≤ M ↔ ∀ x, φ x ∈ M

theorem **algOf_apply** (M : IntermediateField ℚ ℂ) (h : ∀ x, φ x ∈ M) (x : E) : letI

theorem **finrank_mul_degOver** (M : IntermediateField ℚ ℂ) (h : ∀ x, φ x ∈ M) : Module.finrank ℚ E ∗ degOver φ M h = Module.finrank ℚ M

theorem **degOver_pos** (M : IntermediateField ℚ ℂ) [NumberField M] (h : ∀ x, φ x ∈ M) : 0 < degOver φ M h

theorem **IsRayFieldIn.emb_mem** {q : Ideal (𝒪 E)} {M : IntermediateField ℚ ℂ} (h : IsRayFieldIn φ q M) (x : E) : φ x ∈ M

theorem **IsRayFieldIn.numberField** {q : Ideal (𝒪 E)} {M : IntermediateField ℚ ℂ} (h : IsRayFieldIn φ q M) : NumberField M

theorem **IsRayFieldIn.isRayField** {q : Ideal (𝒪 E)} {M : IntermediateField ℚ ℂ} [NumberField M] (h : IsRayFieldIn φ q M) : letI

theorem **IsRayField.exists_isRayFieldIn** {L : Type} [Field L] [NumberField L] [Algebra E L] {q : Ideal (𝒪 E)} (h : IsRayField E L q) : ∃ M, IsRayFieldIn φ q M

theorem **exists_isRayFieldIn** (h : RayField) {q : Ideal (𝒪 E)} (hq : q ≠ ⊥) : ∃ M, IsRayFieldIn φ q M

theorem **natIdeal_ne_bot** {n : ℕ} (hn : n ≠ 0) : natIdeal E n ≠ ⊥

theorem **imField_le_kummerField** (h : ℕ) : imField φ ≤ kummerField φ h

theorem **cycField_le_kummerField** (h : ℕ) : cycField h ≤ kummerField φ h

theorem **mem_kummerField** {h : ℕ} {y : E} (hy : y ∈ powIdealElts E h) {z : ℂ} (hz : z ^ h = φ y) : z ∈ kummerField φ h

theorem **kummerField_le_iff** {h : ℕ} {M : IntermediateField ℚ ℂ} : kummerField φ h ≤ M ↔ (∀ x, φ x ∈ M) ∧ rootOne h ∈ M ∧ ∀ y ∈ powIdealElts E h, ∀ z : ℂ, z ^ h = φ y → z ∈ M

theorem **pow_mem_powIdealElts** {h : ℕ} {y : E} (hy : y ∈ powIdealElts E h) (k : ℕ) : y ^ k ∈ powIdealElts E (h * k)

theorem **kummerField_mono** {h h' : ℕ} (h0 : h' ≠ 0) (hd : h ∣ h') : kummerField φ h ≤ kummerField φ h'

theorem **kummerField_one** : kummerField φ 1 = imField φ

theorem **valueField_le_iff** {χ : HChar E} {s : ℝ} {M : IntermediateField ℚ ℂ} : valueField χ s ≤ M ↔ χ.ValuesIn s M.val.toRingHom

theorem **valuesIn_iff** {χ : HChar E} {s : ℝ} {C : Type} [Field C] [NumberField C] (e : C →+* ℂ) : χ.ValuesIn s e ↔ valueField χ s ≤ imField e

theorem **torsionCoords_mono** (W : WeierstrassCurve ℂ) {n n' : ℕ} (h : n ∣ n') : torsionCoords W n ⊆ torsionCoords W n'

theorem **torsionCoords_one** (W : WeierstrassCurve ℂ) : torsionCoords W 1 = ∅

theorem **hilbertField_le_torsionField** : hilbertField d F ≤ torsionField d A

theorem **torsionField_le_discField** : torsionField d A ≤ discField d A

theorem **discField_le_iff** {M : IntermediateField ℚ ℂ} : discField d A ≤ M ↔ sqrtNegNat d ∈ M ∧ F ≤ M ∧ torsionCoords (A.baseChange ℂ) 12 ⊆ M ∧ ∀ z : ℂ, z ^ 12 = (A.Δ : ℂ) → z ∈ M

theorem **discField_eq_of_mem** {u : ℂ} (hu : u ∈ torsionField d A) (h0 : u ≠ 0) : torsionField d A ⊔ IntermediateField.adjoin ℚ {z : ℂ | z ^ 12 = u ^ 12 * (A.Δ : ℂ)} = discField d A

theorem **hilbertField_eq** (h : IsGoodCMCurve d F A) {L : PeriodPair} (hL : IsIntegersLattice d L) : hilbertField d F = ℚ⟮sqrtNegNat d, (latticeCurve L).j⟯

theorem **typeNorm_mul** (p q : ℕ) (γ γ' : H.E) : typeNorm H p q (γ * γ') = typeNorm H p q γ * typeNorm H p q γ'

theorem **typeNorm_ne_zero** (p q : ℕ) {γ : H.E} (h : γ ≠ 0) : typeNorm H p q γ ≠ 0

theorem **typeNorm_natCast** (p q n : ℕ) : typeNorm H p q (n : H.E) = (n : H.E) ^ (R.f * (p + q))

theorem **kPsiField_eq** (b : ClassBasis H.E) (p q : ℕ) (ψ : Fin b.k → ℂ) (hψ : ∀ i, ψ i ^ b.h i = H.φ₀ (typeNorm H p q (b.γ i : H.E))) : kPsiField H b p q = imField H.φ₀ ⊔ cycField (classNumber H.E) ⊔ IntermediateField.adjoin ℚ (Set.range ψ)

theorem **modulus_ne_zero** (P : Pre) : modulus P ≠ 0

theorem **rootOrder_ne_zero** (P : Pre) : rootOrder P ≠ 0

theorem **nonempty_fieldDat** (hR : RayField) (hA : CMCurveGoodOutsidePrimeSt) (P : Pre) : Nonempty (FieldDat P)

theorem **hstar_pos** : 0 < D.hstar

theorem **finrank_mul_hstar** : Module.finrank ℚ P.host.E * D.hstar = Module.finrank ℚ D.ray

theorem **M_le_M'** : D.M ≤ D.M'

theorem **M'_le_Om** : D.M' ≤ D.Om

theorem **ray_le_Om** : D.ray ≤ D.Om

theorem **cycField_le_Om** : cycField (rootOrder P) ≤ D.Om

theorem **V_le_Om** : D.V ≤ D.Om

theorem **imField_le_Om** : imField P.host.φ₀ ≤ D.Om

theorem **listField_le_iff** {P : Pre} (D : FieldDat P) (C : Chars P) {N : IntermediateField ℚ ℂ} : listField D C ≤ N ↔ imField P.host.φ₀ ≤ N ∧ D.M' ≤ N ∧ D.ray ≤ N ∧ rootsField P ≤ N ∧ charField C ≤ N ∧ ∀ j, 1 ≤ j → j ≤ P.R.m → D.KPsi j ≤ N

theorem **om_mem_rootsField** {P : Pre} (C : Chars P) (k : Fin 2) (a : (ZMod (P.host.g k))ˣ) : ((C.W.om k a : ℂˣ) : ℂ) ∈ rootsField P

theorem **mem_LLC** {z : ℂ} : z ∈ B.LLC ↔ ∃ x, B.embLL x = z

theorem **mem_LL0C** {z : ℂ} : z ∈ B.fixedCoefficientFieldInComplex ↔ ∃ x ∈ B.LL0, B.embLL x = z

theorem **mem_KsC** {z : ℂ} : z ∈ B.KsC ↔ ∃ x ∈ B.Ks, B.embLL x = z

theorem **KsC_le_LL0C** : B.KsC ≤ B.fixedCoefficientFieldInComplex

theorem **fixedField_le_coefficientField** : B.fixedCoefficientFieldInComplex ≤ B.LLC

theorem **imField_KstoC** : imField B.KstoC = B.KsC

theorem **imField_LL0toC** : imField B.AutomorphicRingToComplex = B.fixedCoefficientFieldInComplex

theorem **finrank_KsC** : Module.finrank ℚ B.KsC = Module.finrank ℚ B.Ks

theorem **discr_KsC** : discr B.KsC = discr B.Ks

theorem **imField_le_KsC** : imField B.host.φ₀ ≤ B.KsC

theorem **rootsField_le_KsC** : rootsField B.toPre ≤ B.KsC

theorem **valIn_iff** : B.ValIn C ↔ charField C ≤ B.KsC ∧ valueField C.chiPsi 0 ≤ B.fixedCoefficientFieldInComplex

theorem **lowerOK_iff** : B.LowerOK D C c M ↔ listField D C ≤ B.KsC ∧ valueField C.chiPsi 0 ≤ B.fixedCoefficientFieldInComplex ∧ B.RootsIn c M ∧ B.RayInLL

theorem **fieldsOK_iff** : B.FieldsOK D C c M ↔ (listField D C ≤ B.KsC ∧ B.KsC ≤ D.Om) ∧ valueField C.chiPsi 0 ≤ B.fixedCoefficientFieldInComplex ∧ B.RootsIn c M ∧ B.RayInLL

theorem **ValIn.characterValues_mem_fixedField** (h : B.ValIn C) : (∀ j, (C.mu0 j).ValuesIn 0 B.AutomorphicRingToComplex) ∧ C.eta.ValuesIn (1 / 2) B.AutomorphicRingToComplex ∧ C.chiPsi.ValuesIn 0 B.AutomorphicRingToComplex ∧ C.del.ValuesIn 0 B.AutomorphicRingToComplex ∧ C.del'.ValuesIn 0 B.AutomorphicRingToComplex

theorem **RootsIn.mono** (h : B.RootsIn c M) {c' M' : ℕ} (hc : c' ∣ c) (hM : M' ≤ M) : B.RootsIn c' M'

theorem **KsExact.lower** (h : B.KsExact D C) : D.ray ≤ B.KsC ∧ B.CurveIn D ∧ charField C ≤ B.KsC ∧ B.KummerIn D

theorem **KsExact.ksLeOm_iff** (h : B.KsExact D C) : B.KsLeOm D ↔ listField D C ≤ D.Om

theorem **rayIn** : B.RayIn D

theorem **curveIn** : B.CurveIn D

theorem **valIn** : B.ValIn C

theorem **rootsIn** : B.RootsIn c M

theorem **kummerIn** : B.KummerIn D

theorem **ksLeOm** : B.KsLeOm D

theorem **listField_le** : listField D C ≤ B.KsC

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FixedVectorEqOfStandardData

theorem **ReadsTop.mem_chi** {u : B.Word} {a : I.ι} {Ch : Type} {chOf : (I.stp u a B.R.m).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)} {U : Ch → Subgroup (GA B (B.R.N B.R.m))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a B.R.m).Cl → GA B (B.R.N B.R.m)} {st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E} (h : I.ReadsTop l Z u a chOf Lb U chi y st) (c : (I.stp u a B.R.m).Cl) (d : XPt (B.primes X) B.R I.sel (I.tup a)) (g' : GA B (B.R.N B.R.m - 1)) (g : GA B (B.R.N B.R.m)) (hc : ((I.stp u a B.R.m).rep c).1.1 = tilF I.sel (I.tup a) u d) (hd : PtOf g' d) (hg : IsExt (I.rec7 l Z) (z0 X u) (inclE (F

theorem **ReadsTop.topBase** {u : B.Word} {a : I.ι} {Ch : Type} {chOf : (I.stp u a B.R.m).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)} {U : Ch → Subgroup (GA B (B.R.N B.R.m))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a B.R.m).Cl → GA B (B.R.N B.R.m)} {st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E} (h : I.ReadsTop l Z u a chOf Lb U chi y st) (β : Ch) (hβ : ∀ c, U (chOf c) = U β) : I.TopBase l Z u a (U β) y

theorem **ReadsKTop.readsK** {u : B.Word} {a : I.ι} {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {Ch : Type} {chOf : (I.stp u a B.R.m).Cl → Ch} {K : ∀ β : Ch, LevelChart (I.rec7 l Z) (B.T u) bad B.R.m le_rfl (fun c : {c // chOf c = β} => (I.chain l u a).stab B.R.m c.1) (fun c => latCrd (I.rec7 l Z) B.R.m le_rfl (I.latAt u a B.R.m c.1))} (h : I.ReadsKTop l Z u a chOf K) : I.ReadsK l Z u a B.R.m le_rfl chOf K

theorem **StdP.lat_eq** {P : TwPTiedS Lv} {E : TwPEvzS P} (h : I.StdP l Z l0 P E) (u a ha) (c : (I.stX a).Cl) : (P.cls u a ha).lat c = ((I.stX a).rep c).1.1.deco.1.N

theorem **StdP.unique** {P P' : TwPTiedS Lv} {E : TwPEvzS P} {E' : TwPEvzS P'} (h : I.StdP l Z l0 P E) (h' : I.StdP l Z l0 P' E') (u a) : E.eWz u a = E'.eWz u a ∧ ∀ ha, (∀ c, (P.cls u a ha).lat c = (P'.cls u a ha).lat c) ∧ ∀ h1, (P.til u a ha h1).z0 = (P'.til u a ha h1).z0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FlagComponentFramedPointFrame

theorem **isCombAtG_single** {X0 : Scheme.{0}} (t : SAB H X0) {R : Type} [CommRing R] (x : Sp R ⟶ X0) {m : ℕ} {v : Fin m → (Sp (DualNumber R) ⟶ t.G.left)} (hv : ∀ j, IsTanAtG t x (v j)) (j : Fin m) : IsCombAtG t x v (Pi.single j 1) (v j)

theorem **EigFrame.tri_refl** {Z : Model Xo c β} {R : Type} [CommRing R] {φ : c.F →+* R} {x : Sp R ⟶ Z.X} (f : EigFrame Z φ x) : f.Tri (RingHom.id R) f 1

theorem **Lw_zero** : C.Lw 0 = 0

theorem **Lw_add** (la la' : GlobalTheory.AllWt.Weight Dw) : C.Lw (la + la') = C.Lw la + C.Lw la'

theorem **LEx_congr** {a b : Ix Dw → ℤ} (h : ∀ x, a (some x) = b (some x)) : C.LEx a = C.LEx b

theorem **LEx_single** (x : Σ τ : E →+* ℂ, Fin (Dw.mult τ)) : C.LEx (Pi.single (some x) 1) = -C.gr x.1 x.2

theorem **exKey_two_mul** (a : (E →+* ℂ) × ℕ → ℕ) (m : ℕ) : exKey Dw a (2 * m) = GlobalTheory.AllWt.exOf Dw (GlobalTheory.AllWt.weightOf Dw (fun x => (a x : ℤ)) m) 0

theorem **FlagCpt.LEx_exKey** {Z : Model Xo c β} {e : F ≃+* c.F} (C : FlagCpt Z e) (a : (E →+* ℂ) × ℕ → ℕ) (m : ℕ) : C.LEx (exKey Dw a (2 * m)) = C.Lw (GlobalTheory.AllWt.weightOf Dw (fun x => (a x : ℤ)) m)

theorem **KeySec.pow_mem** (κ : KeySec Xo c) (m : ℕ) : (κ.ρ : Rul Xo c) ^ m ∈ GlobalTheory.AllWt.secZ Xo c (fun i => (m : ℤ) * keyEx Dw i)

theorem **KeySec.mul_pow_mem** (κ : KeySec Xo c) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) : f * (κ.ρ : Rul Xo c) ^ (n / 2) ∈ GlobalTheory.AllWt.secZ Xo c (exKey Dw a n)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FnAlgClosedOfFixedPoints

theorem **actsBy_of_dvd** {C : Curve H} {τ : ℂ ≃+* ℂ} {io : C.P.A.left ⟶ C.P.A.left} {m n : ℕ} (hn : n ≠ 0) (h : m ∣ n) {s : ℤ × ℤ} (hC : C.ActsBy τ io n s) : C.ActsBy τ io m s

theorem **splitLat_all** (h1 : ∃ D₁ : Datum H, D₁.p = 1 ∧ D₁.q = 0) (D : Datum H) : SplitLat D

theorem **congr_of_section** (F : FamilyUnif Xo σ ty) (τ : ℂ ≃+* ℂ) {i i' : Fin F.n} {P : (F.Di i).La →+ (Fin g → ℂ)} {P' : (F.Di i').La →+ (Fin g → ℂ)} (hP' : IsPeriod (F.Di i') ty P') (hu : spMap (τ : ℂ →+* ℂ) ≫ F.u i P = F.u i' P') {n : ℕ} (hn : 0 < n) (s : 𝟙_ (Over Xo.X) ⟶ Xo.univ.t.G) {x : (F.Di i).La} {x' w : (F.Di i').La} (h1 : toUnit _ ≫ s = F.ex i P ((n : ℂ)⁻¹ • P x)) (h2 : toUnit _ ≫ s = F.ex i' P' ((n : ℂ)⁻¹ • P' x')) (key : spMap (τ : ℂ →+* ℂ) ≫ (F.ex i P ((n : ℂ)⁻¹ • P x)).left = (F.ex i' P' ((n : ℂ)⁻¹ • P' w)).left) : ∃ e, w = x' + n • e

theorem **conj_point_eq** (F : FamilyUnif Xo σ ty) (hD : 1 ≤ D.p + D.q) (hσ : σ (baseKA l).ρ = H.rt0) (hty : IsTypeFn H D.p D.q ty) (τ : ℂ ≃+* ℂ) (hτσ : (τ : ℂ →+* ℂ).comp σ = σ) (hfix : FixesCurve H τ l.M') (i : Fin F.n) (P : (F.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (F.Di i) ty P) (hs : IsSplit hP) : spMap (τ : ℂ →+* ℂ) ≫ F.u i P = F.u i P

theorem **fnAlgClosed_of_fixed_points** (F : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (h1 : 1 ≤ D.p + D.q) (hty : IsTypeFn H D.p D.q ty) (c : Coeff l) (hfix : ∀ τ : ℂ ≃+* ℂ, (∀ x ∈ c.F, τ x = x) → ∀ i, ∃ P, IsPeriod (F.Di i) ty P ∧ spMap (τ : ℂ →+* ℂ) ≫ F.u i P = F.u i P) : FnAlgClosed Xo c

theorem **fnAlgClosed_of_curveOver** (Xo : Moduli D l) (c : Coeff l) (hD : 1 ≤ D.p + D.q) (h1 : ∃ D₁ : Datum H, D₁.p = 1 ∧ D₁.q = 0) (hC : CurveOver H c.F l.M') : FnAlgClosed Xo c

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FormAtIdOfIsMonomial

theorem **minVal_nat_mul** (n k : ℕ) (B : (Fin n → O) → (Fin n → O) → ℝ) : (InvFun.minVal O).toFun n (fun a b => k * B a b) = k * (InvFun.minVal O).toFun n B

theorem **isTame_posMin** : IsTame (posMin O)

theorem **not_isTame_minVal** : ¬ IsTame (InvFun.minVal ℤ)

theorem **comb_eval** (a b : ℕ) (ψ φ : InvFun O) (X : Type) [AddCommGroup X] [Module O X] [Module.Finite O X] (B : X → X → ℝ) : (comb a ψ b φ).eval X B = a * ψ.eval X B + b * φ.eval X B

theorem **isTame_comb** {ψ φ : InvFun O} (h : IsTame ψ) (h' : IsTame φ) (a b : ℕ) : IsTame (comb a ψ b φ)

theorem **resInt_eval** [Module.Finite ℤ O] (φ : InvFun ℤ) (X : Type) [AddCommGroup X] [Module O X] [Module.Finite O X] [Module.Finite ℤ X] (B : X → X → ℝ) : (resInt O φ).eval X B = φ.eval X B

theorem **isTame_resInt** [Module.Finite ℤ O] {φ : InvFun ℤ} (h : IsTame φ) : IsTame (resInt O φ)

theorem **posMin_eval** {B : X → X → ℝ} (hB : ∀ y, 0 ≤ B y y) : (posMin O).eval X B = (InvFun.minVal O).eval X B

theorem **isCone_congr** (h : ∀ B : X → X → ℝ, (∀ y, 0 ≤ B y y) → φ.eval X B = φ'.eval X B) {m : ℕ} {γ : Fin m → LinearMap.BilinForm ℤ X} (hγ : IsCone bar φ γ) : IsCone bar φ' γ

theorem **isPolFanOn_congr** (h : ∀ B : X → X → ℝ, (∀ y, 0 ≤ B y y) → φ.eval X B = φ'.eval X B) (hφ : IsPolFanOn bar φ X) : IsPolFanOn bar φ' X

theorem **isPolFan_posMin_one** : IsPolFan id 1 (posMin ℤ) ∧ IsTame (posMin ℤ)

theorem **isTwInv_resInt** (φ : InvFun ℤ) : IsTwInv H (resInt (𝒪 E) φ)

theorem **isTwInv_posMin** : IsTwInv H (posMin (𝒪 E))

theorem **isPosFn_posMin** : IsPosFn H (posMin (𝒪 E))

theorem **refines_zero** (φs : InvFun ℤ) (𝒩 : Finset ℕ) : Refines H 0 φs 𝒩 (posMin (𝒪 E))

theorem **hasCones_mono** {Cn Cn' : ∀ (X : Type) [AddCommGroup X] [Module (𝒪 E) X] [Module.Finite (𝒪 E) X] (m : ℕ), (Fin m → LinearMap.BilinForm ℤ X) → Prop} (h : ∀ (X : Type) [AddCommGroup X] [Module (𝒪 E) X] [Module.Finite (𝒪 E) X] m γ, Cn X m γ → Cn' X m γ) {Z : Model Xo c β} (hZ : HasCones Cn Z) : HasCones Cn' Z

theorem **formAt_id_of_isMonomial** {ϖ : R} {be : LinearMap.BilinForm ℤ Dg.X} (h : IsMonomial Dg ![ϖ] ![be]) : FormAt Dg (RingHom.id R) ϖ be

theorem **inCone_of_isMonomial** {m : ℕ} {t : Fin m → R} {γ : Fin m → LinearMap.BilinForm ℤ Dg.X} (hs : ∀ i y z, γ i y z = γ i z y) (h : IsMonomial Dg t γ) : InCone Dg γ

theorem **eq_zero_of_inCone_empty** [IsLocalRing R] (hR : CDvr R) {ϖ : R} (hϖ : Irreducible ϖ) {be : LinearMap.BilinForm ℤ Dg.X} (hs : ∀ y z, be y z = be z y) (h : IsMonomial Dg ![ϖ] ![be]) (hc : InCone Dg (![] : Fin 0 → _)) : be = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FormalTwistingNeTrivial

theorem **isGen_iso_iff** (u : LB.Iso A B) : u.IsGen ↔ ∀ (x : X) (p q : B.ι × A.ι), x ∈ B.U p.1 ⊓ A.U p.2 → x ∈ B.U q.1 ⊓ A.U q.2 → gm (𝟙 X) x (B.U p.1 ⊓ A.U p.2) (u.s p) = gm (𝟙 X) x _ (B.g p.1 q.1) ∗ (gm (𝟙 X) x _ (A.g p.2 q.2))⁻¹ ∗ gm (𝟙 X) x (B.U q.1 ⊓ A.U q.2) (u.s q)

theorem **gm_congr** {U V : X.Opens} {a : Γ(X, U)ˣ} {b : Γ(X, V)ˣ} {W : X.Opens} {c : Γ(X, W)ˣ} (h : ∀ x ∈ U ⊓ V ⊓ W, gm (𝟙 X) x U a = gm (𝟙 X) x V b ∗ gm (𝟙 X) x W c) (f : Y ⟶ X) (y : Y) (hy : f y ∈ U ⊓ V ⊓ W) : gm f y U a = gm f y V b ∗ gm f y W c

theorem **exists_trans** (hA : A.IsLB) (hB : B.IsLB) (hC : C.IsLB) {u : LB.Iso A B} {v : LB.Iso B C} (hu : u.IsGen) (hv : v.IsGen) : ∃ w : LB.Iso A C, w.IsGen ∧ ∀ (Y : Scheme.{0}) (f : Y ⟶ X) (y : Y) (a : A.ι) (b : B.ι) (c : C.ι), f y ∈ A.U a → f y ∈ B.U b → f y ∈ C.U c → gm f y (C.U c ⊓ A.U a) (w.s (c, a)) = gm f y (C.U c ⊓ B.U b) (v.s (c, b)) ∗ gm f y (B.U b ⊓ A.U a) (u.s (b, a))

theorem **eq_res_pullU_iff** {Y Z : Scheme.{0}} (f : Z ⟶ Y) {U : Y.Opens} {V : Z.Opens} (h : V ≤ f ⁻¹ᵁ U) (a : Γ(Z, V)ˣ) (b : Γ(Y, U)ˣ) : a = res h (pullU f U b) ↔ ∀ z ∈ V, gm (𝟙 Z) z V a = gm f z U b

theorem **mem_succ** (L : FLB i) {e : ℕ} {a : L.ι} {x : X e} (h : x ∈ L.U e a) : i e x ∈ L.U (e + 1) a

theorem **gm_g** (L : FLB i) {e : ℕ} {a b : L.ι} {x : X e} (h : x ∈ L.U e a ⊓ L.U e b) : gm (𝟙 _) x _ (L.g e a b) = gm (i e) x _ (L.g (e + 1) a b)

theorem **lb_iso_pull** (L : FLB i) (e : ℕ) : LBIso (L.lb e) ((L.lb (e + 1)).pull (i e))

theorem **compat_iff** {L M : FLB i} (u : ∀ e, LB.Iso (L.lb e) (M.lb e)) : Compat u ↔ ∀ e (p : M.ι × L.ι), ∀ x ∈ M.U e p.1 ⊓ L.U e p.2, gm (𝟙 _) x (M.U e p.1 ⊓ L.U e p.2) ((u e).s p) = gm (i e) x (M.U (e + 1) p.1 ⊓ L.U (e + 1) p.2) ((u (e + 1)).s p)

theorem **reindex_iso** (L : FLB i) {κ : Type} (j : κ → L.ι) (h : ∀ e (x : X e), ∃ k, x ∈ L.U e (j k)) : (L.reindex j h).Iso L

theorem **Iso.refl** (L : FLB i) : L.Iso L

theorem **Iso.symm** {L M : FLB i} (h : L.Iso M) : M.Iso L

theorem **Iso.trans** {L M N : FLB i} (h : L.Iso M) (h' : M.Iso N) : L.Iso N

theorem **ofPull_lb** (L : LB Y) (hL : L.IsLB) (e : ℕ) : (ofPull q hq L hL).lb e = L.pull (q e)

theorem **iso_of_chartPull** {L M : FLB i} {A B : LB Y} (a : L.ι → A.ι) (b : M.ι → B.ι) (la : ∀ e k, L.U e k ≤ q e ⁻¹ᵁ A.U (a k)) (lb : ∀ e k, M.U e k ≤ q e ⁻¹ᵁ B.U (b k)) (ga : ∀ e (x : X e) k l, x ∈ L.U e k → x ∈ L.U e l → gm (𝟙 _) x _ (L.g e k l) = gm (q e) x _ (A.g (a k) (a l))) (gb : ∀ e (x : X e) k l, x ∈ M.U e k → x ∈ M.U e l → gm (𝟙 _) x _ (M.g e k l) = gm (q e) x _ (B.g (b k) (b l))) (hq : ∀ e, i e ≫ q (e + 1) = q e) (h : LBIso A B) : L.Iso M

theorem **ofPull_iso** {A B : LB Y} (hA : A.IsLB) (hB : B.IsLB) (h : LBIso A B) : (ofPull q hq A hA).Iso (ofPull q hq B hB)

theorem **ofPull_comp** (f : Y ⟶ Y') {A : LB Y'} (hA : A.IsLB) : (ofPull q hq (A.pull f) (pull_isLB hA f)).Iso (ofPull (fun e => q e ≫ f) (fun e => by rw [← Category.assoc, hq]) A hA)

theorem **fcls_eq_iff** {L M : FLB i} : fcls L = fcls M ↔ L.Iso M

theorem **fcls_surj** (c : FPic i) : ∃ L, fcls L = c

theorem **level_fcls** (e : ℕ) (L : FLB i) : level e (fcls L) = smallCls (L.lb e) (L.isLB e)

theorem **smallPull_level** (e : ℕ) (c : FPic i) : smallPull (i e) (level (e + 1) c) = level e c

theorem **fpull_smallCls** (L : LB Y) (hL : L.IsLB) : fpull q hq (smallCls L hL) = fcls (FLB.ofPull q hq L hL)

theorem **level_fpull** (e : ℕ) (c : SmallPic Y) : level e (fpull q hq c) = smallPull (q e) c

theorem **fpull_smallPull** (f : Y ⟶ Y') (c : SmallPic Y') : fpull q hq (smallPull f c) = fpull (fun e => q e ≫ f) (fun e => by rw [← Category.assoc, hq]) c

theorem **fpull_eq_iff** (q' : ∀ e, X e ⟶ Y') (hq' : ∀ e, i e ≫ q' (e + 1) = q' e) {L : LB Y} (hL : L.IsLB) {M : LB Y'} (hM : M.IsLB) : fpull q hq (smallCls L hL) = fpull q' hq' (smallCls M hM) ↔ ∃ u : ∀ e, LB.Iso (L.pull (q e)) (M.pull (q' e)), (∀ e, (u e).IsGen) ∧ FLB.Compat (L

theorem **fpull_id_injective** (Y : Scheme.{0}) : Function.Injective (fpull (X

theorem **projTrans_comp** (e : ℕ) : projTrans I N e ≫ baseChange N (Ideal.Quotient.mk (I ^ (e + 1 + 1))) = baseChange N (Ideal.Quotient.mk (I ^ (e + 1)))

theorem **formalTwisting_ne_trivial** (hI : I ≠ ⊤) : fcls (formalTwistingSheaf I (N + 1)) ≠ fpull (X

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.Forms

theorem **ExtN.ext** {z : Dat (w.st Z) w.I hX T Wt} {z' : Dat (w'.st Z') w'.I hX' T' Wt'} (e : ExtN z z') (h : ∀ v v', Below (w.D Z) (w'.D Z') v v' → (v ∈ z.W1 (fun _ _ => 0) ↔ v' ∈ z'.W1 fun _ _ => 0)) : Ext z z'

theorem **Ext.extN** {z : Dat (w.st Z) w.I hX T Wt} {z' : Dat (w'.st Z') w'.I hX' T' Wt'} (e : Ext z z') : ExtN z z'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FormsOver

theorem **Lv.good_σ₀** (κ : Lv A) : Formalism.L2.PjFree.GoodEmb κ.c κ.σ₀

theorem **FinalTheorems.isUnf** : Formalism.L2.PjFree.IsUnf (s.U κ.σ₀)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.FrobOrderSpec

theorem **frobOrder_spec** (D : D7) {T p : ℕ} (hg : GIdx D.host T p) : 2 ≤ frobOrder D p ∧ Even (frobOrder D p)

theorem **pack_ψ** : (pack D u Lc w h5 h2 he I ψ).ψ = ψ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.GeoExV

theorem **GeoEx.geoExV** {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m} {C : Formalism.L2.Pj.CoreJ D} [Algebra C.Q.c.F D.what₂.LL] {F : Face D A3.A} (h : GeoEx A3 C F) : GeoExV A3 C F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.GeoTiedPGraft

theorem **Keeps.refl** (X : DivX D) : Keeps X X

theorem **Keeps.trans** {X Y Z : DivX D} (h : Keeps X Y) (h' : Keeps Y Z) : Keeps X Z

theorem **Keeps.symm** {X Y : DivX D} (h : Keeps X Y) : Keeps Y X

theorem **Keeps.mjH** {X X₀ : DivX D} (h : Keeps X X₀) : X.MjH = X₀.MjH ∧ HEq X.jM X₀.jM

theorem **Keeps.num** {X X₀ : DivX D} (h : Keeps X X₀) : X.kapR = X₀.kapR ∧ X.logLam = X₀.logLam ∧ X.Qp = X₀.Qp ∧ X.QPl = X₀.QPl ∧ HEq X.qv X₀.qv ∧ HEq X.nu X₀.nu ∧ HEq X.ZQ X₀.ZQ ∧ HEq X.quotientLocalValueFirst X₀.quotientLocalValueFirst ∧ HEq X.typeDim X₀.typeDim

theorem **GeoOver.mono** {Q Q' : DivX D → Prop} (h : ∀ X, Q X → Q' X) {X₀ : DivX D} (g : GeoOver Q X₀) : GeoOver Q' X₀

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.GoodPartEq

theorem **goodPart_eq** : @goodPart b.D = @goodPartB B C a b

theorem **partFull_eq** : @partFull b.D = @partFullB B C a b

theorem **io_congr** (R : RealDat D A Ks) {m m' : type_of% (@MultT D)} (e : @m = @m') {β : Type} (f : ∀ m : type_of% (@MultT D), IoYT (@m) R → IoXT (@m) R → β) (y : IoYT (@m) R) (x : IoXT (@m) R) : f (@m) y x = f (@m') (castAlong (congrArg (fun m : type_of% (@MultT D) => IoYT (@m) R) e) y).1 (castAlong (congrArg (fun m : type_of% (@MultT D) => IoXT (@m) R) e) x).1

theorem **eeat_eq** : R.EEat = (@RealDat.EEat (@Late.D B C b) G ‹Group G› ‹Fintype G› Pl0 T (@famM B C a b G ‹Group G› ‹Fintype G› Pl0 T A) Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks››› (@alg B C a b Ks ‹Field Ks› iK) (@real B C a b G ‹Group G› ‹Fintype G› Pl0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› iK A R))

theorem **wall_eq** : R.Wall = (by with_unfolding_all exact ((letI

theorem **wat_eq** : R.Wat = (by with_unfolding_all exact ((letI

theorem **cc_eq** : R.CC = (@RealDat.CC (@Late.D B C b) G ‹Group G› ‹Fintype G› Pl0 T (@famM B C a b G ‹Group G› ‹Fintype G› Pl0 T A) Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› (@alg B C a b Ks ‹Field Ks› iK) (@real B C a b G ‹Group G› ‹Fintype G› Pl0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› iK A R))

theorem **thDn_eq** : R.thDn = (@RealDat.thDn (@Late.D B C b) G ‹Group G› ‹Fintype G› Pl0 T (@famM B C a b G ‹Group G› ‹Fintype G› Pl0 T A) Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› (@alg B C a b Ks ‹Field Ks› iK) (@real B C a b G ‹Group G› ‹Fintype G› Pl0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› iK A R))

theorem **pTh_eq** : R.PTh = (@RealDat.PTh (@Late.D B C b) G ‹Group G› ‹Fintype G› Pl0 T (@famM B C a b G ‹Group G› ‹Fintype G› Pl0 T A) Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› (@alg B C a b Ks ‹Field Ks› iK) (@real B C a b G ‹Group G› ‹Fintype G› Pl0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› iK A R))

theorem **pThF_eq** : R.PThF = (by with_unfolding_all exact ((letI

theorem **levIs_tr** {L : Links.LevCh a.D T} {μ0 : T} (h : L.Is μ0) : (levCh b L).Is μ0

theorem **isFam_tr** (h : R.IsFam) : letI

theorem **isModels_tr** (h : R.IsModels) : letI

theorem **isChainW_tr** (h : R.IsChainW) : letI

theorem **isThPW_tr** (h : R.IsThPW) : letI

theorem **exAt_tr** (h : R.ExAt) : letI

theorem **lLevelW_tr** (h : R.LLevelW F) : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.GoodPartHasComplexCoordinates

theorem **lin_cpt_semilinear** {K E : Type} [Field K] [AddCommGroup E] [Module ℂ E] (σ : K →+* ℂ) {n : ℕ} (j : (Fin n → K) →ₛₗ[σ] E) (x : Fin n → K) : lin (fun k => j (Pi.single k 1)) (cpt σ x) = j x

theorem **lin_bijective_of_form** {K V E ι : Type} [Field K] [AddCommGroup V] [Module K V] [AddCommGroup E] [Module ℂ E] (σ : K →+* ℂ) {n : ℕ} (j : V →ₛₗ[σ] E) (b : Module.Basis ι K V) (e : (Fin n → K) ≃ₗ[K] V) (hb : LinearIndependent ℂ (fun i => j (b i)) ∧ Submodule.span ℂ (Set.range (fun i => j (b i))) = ⊤) : Function.Bijective (lin fun k => j (e (Pi.single k 1)))

theorem **cjv_cpt_embLL** {n : ℕ} (x : Fin n → D.what₂.LL) : cjv s (cpt D.what₂.embLL x) = cpt Bs.v0.embedding x

theorem **rowCrd_cpt** (y : Fin ((lv (fc X.zs) true).dQ Bs.μ0) → D.what₂.LL) : R.rowCrd (cpt D.what₂.embLL y) = D.jEE (R.eRR y)

theorem **Read.embedding_eq_of_untwisted** (R : Read D X fc false) : Bs.v0.embedding = D.what₂.embLL

theorem **chainTie_iff_untwisted** : M.ChainTie ↔ ∀ u (x : Fin ((fc X.zs).Dc.a (R.wU u)) → D.what₂.LL) (y : Fin ((fc X.zs).Df1.nV Bs.μ0) → D.what₂.LL), D.what₂.perCol u D.what₂.mu0 ((M.cφ u).symm (cpt Bs.v0.embedding x)) (M.cF (cpt Bs.v0.embedding y)) = Bs.v0.embedding ((fc X.zs).Dc.Pt (R.wU u) Bs.μ0 x y)

theorem **colNorm_iff_untwisted** : M.ColNorm ↔ ∀ u, ∃ c : ℝ, 0 < c ∧ ∀ φ : D.what₂.Mu u, Formalism.DivisionValues.fnorm ((fc X.zs).Dc.Xc (R.wU u) Bs.v0) (M.cφ u φ) = c * ‖φ‖

theorem **rowTie_iff_untwisted** (eV : ∀ t, (Fin ((fc X.zs).Df1.nV t) → D.what₂.LL) →ₗ[D.what₂.LL] A.FR t) : M.RowTie eV ↔ ∃ lam : ℂ, lam ≠ 0 ∧ (∀ F, (lam • M.eRLin) F ∈ X.EEl true) ∧ (∀ e ∈ X.EEl true, ∃ F, (lam • M.eRLin) F = e) ∧ (∃ c : ℝ, 0 < c ∧ ∀ F, ‖(lam • M.eRLin) F‖ = c * ‖F‖) ∧ ∀ y : Fin ((fc X.zs).Df1.nV Bs.μ0) → D.what₂.LL, X.θ Bs.μ0 (X.jR Bs.μ0 (eV Bs.μ0 y)) = (lam • M.eRLin) (D.what₂.iota D.what₂.mu0 (M.cF (cpt Bs.v0.embedding y)))

theorem **castForms_heq** {a b : D.what₂.Tup} (h : a = b) (x : D.what₂.Amb a) : HEq (castForms h x) x

theorem **iota_castForms** {a b : D.what₂.Tup} (h : a = b) (x : D.what₂.Amb a) : D.what₂.iota b (castForms h x) = D.what₂.iota a x

theorem **perCol_castForms** {a b : D.what₂.Tup} (h : a = b) (u : D.what₂.Word) (φ : D.what₂.Mu u) (x : D.what₂.Amb a) : D.what₂.perCol u b φ (castForms h x) = D.what₂.perCol u a φ x

theorem **span_castForms** {a b : D.what₂.Tup} (h : a = b) {ι : Type} (f : ι → D.what₂.Amb a) (hf : Submodule.span ℂ (Set.range f) = D.what₂.V0 a) : Submodule.span ℂ (Set.range fun i => castForms h (f i)) = D.what₂.V0 b

theorem **iota_sharp_side** (D : D7) (μ : D.what₂.Tup) : (RowSide.sharp D).iota μ = D.what₂.iota μ

theorem **Read.ofTie_jV** (h : TupTie D A S₀ I.tup) (y : Fin ((A.fam.R A.μ0).nV (A.fam.lvl F.P1 F.ψ1)) → D.what₂.LL) : (Read.ofTie F X I h).jV y = castForms h.mu0 (I.jA A.μ0 ((rV0 D A F.P1 F.ψ1 A.μ0).mulVec y))

theorem **Read.ofTie_dcPt_iff** (h : TupTie D A S₀ I.tup) : (Read.ofTie F X I h).DcPt ↔ ∀ u x y, D.what₂.embLL ((F.Dc X.zs).Pt (I.wU u) A.μ0 x y) = D.what₂.perCol u (I.tup A.μ0) (D.jN u (I.eNu u x)) (I.jA A.μ0 ((rV0 D A F.P1 F.ψ1 A.μ0).mulVec y))

private theorem **subE_mulVec_eq_sum_subBasis** {K' : Type} [Field K'] [StarRing K'] {n : ℕ} (Z : Submodule K' (Fin n → K')) (c : Fin (subDim Z) → K') : (subE Z).mulVec c = ((∑ j, c j • subBasis Z j : Z) : Fin n → K')

private theorem **subE_mulVec_mem** {K' : Type} [Field K'] [StarRing K'] {n : ℕ} (Z : Submodule K' (Fin n → K')) (c : Fin (subDim Z) → K') : (subE Z).mulVec c ∈ Z

private theorem **exists_subE_mulVec_eq_of_mem** {K' : Type} [Field K'] [StarRing K'] {n : ℕ} (Z : Submodule K' (Fin n → K')) {z : Fin n → K'} (hz : z ∈ Z) : ∃ c, (subE Z).mulVec c = z

private theorem **gaMat_mul'** {K' : Type} [Field K'] [NumberField K'] [StarRing K'] {G' : Type} [Group G'] [Fintype G'] {P' : Type} (B : BlockDat K' G' P') (a b : MonoidAlgebra K' G') : B.S.gaMat B.r (a * b) = B.S.gaMat B.r a * B.S.gaMat B.r b

private theorem **range_lvlOp_eq_range_lvlPr_mul_vE** {K' : Type} [Field K'] [NumberField K'] [StarRing K'] {G' : Type} [Group G'] [Fintype G'] {P' : Type} (B : BlockDat K' G' P') (ℓ : Lvl K' G') : (∀ x, ∃ y, (B.S.lvlOp B.r ℓ.P ℓ.ψ (idemOf ℓ.X)).mulVec x = (B.S.lvlPr B.r ℓ.P ℓ.ψ * B.vE ℓ).mulVec y) ∧ ∀ y, ∃ x, (B.S.lvlPr B.r ℓ.P ℓ.ψ * B.vE ℓ).mulVec y = (B.S.lvlOp B.r ℓ.P ℓ.ψ (idemOf ℓ.X)).mulVec x

theorem **Read.ofTie_spec** (h : DivisionCompatible D A F.P0 F.ψ0 F.P1 F.ψ1 X.zs S₀ (X.data F.cr true) X₀ C NuC I) : (Read.ofTie F X I h.tup).ReadingFormsSpanGoodSubspace ∧ (Read.ofTie F X I h.tup).IotaRd F.cr ∧ (Read.ofTie F X I h.tup).NuTie ∧ (Read.ofTie F X I h.tup).RRTie F.cr ∧ Function.Bijective (Read.ofTie F X I h.tup).wU

theorem **toF_eq_placeGood** (Z : Submodule K B.S.Forms) (x : Fin (B.dZ ℓ Z) → K) : B.toF ℓ Z x = placeGood B ℓ ((B.zE ℓ Z).mulVec x)

theorem **placeGood_injective** : Function.Injective (placeGood B ℓ)

theorem **range_placeGood** : LinearMap.range (placeGood B ℓ) = BlockGood.goodR B ℓ

theorem **csum_placeGood** (x y : Fin (B.nV ℓ) → K) : CSDForm.csumForm B.S (placeGood B ℓ x) (placeGood B ℓ y) = (Nat.card ℓ.P : K) * (star x ·ᵥ (B.HV ℓ).mulVec y)

theorem **csum_toF_placeGood** (Z : Submodule K B.S.Forms) (hdet : IsUnit (B.HZ ℓ Z).det) (x : Fin (B.dZ ℓ Z) → K) (y : Fin (B.nV ℓ) → K) : CSDForm.csumForm B.S (B.toF ℓ Z x) (placeGood B ℓ y) = CSDForm.csumForm B.S (B.toF ℓ Z x) (B.toF ℓ Z ((B.Pq ℓ Z).mulVec y))

theorem **evaluationTie_goodPlacement** (hOk : A.fam.Ok n) : X.EVTie F.cr (cvOf A F.P0 F.ψ0 F.P1 F.ψ1) (goodSubspaceCoordinateMap F X.zs)

theorem **CxFacts.ofLinks** (hOk : A.fam.Ok n) (hsep : A.fam.TypesSep) {Y : CxData.CxCol (fc

theorem **CxFacts.ofPrimitive** (hOk : A.fam.Ok n) (hsep : A.fam.TypesSep) (hst : X.zs.Stable) (hin : X.zs.InBlock) (hpos : X.H4pos) (hn : NPTie X F.P0 F.P1) (hJ : X.JIsoR (cvOf A F.P0 F.ψ0 F.P1 F.ψ1)) (hR : X.RatLev F.cr (cvOf A F.P0 F.ψ0 F.P1 F.ψ1) true) (hDn : X.DnLev (cvOf A F.P0 F.ψ0 F.P1 F.ψ1) true) (hUp : X.UpLev (cvOf A F.P0 F.ψ0 F.P1 F.ψ1) true) (hB : X.ThmBop true) (hP : X.PartsE true) : CxFacts X F.cr (goodSubspaceCoordinateMap F X.zs) where

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.GqPtOfNeZeroIff

theorem **normOf_eq** (hP : IsFrame σ J T P) : normOf σ J P = T

theorem **ptEntry_replicate** (ζ : E) : ∀ (m i j : ℕ), Formalism.MainNotions.S.ptEntry (List.replicate m ((1 : E), 1) ++ [(ζ, 1)]) i j = if i = j then (if i < m then 1 else if i = m then ζ else 0) else 0 | 0, i, j => by simp only [List.replicate_zero, List.nil_append, Formalism.MainNotions.S.ptEntry] split_ifs <;> first | rfl | omega | m + 1, i, j => by rw [List.replicate_succ, List.cons_append, Formalism.MainNotions.S.ptEntry, ptEntry_replicate ζ m] split_ifs <;> first | rfl | omega omit [NumberField E] in theorem Gq_nullPt {ζ : E} (hζ : ζ ≠ 0) (h : ζ * σ ζ = T * T') : Gq σ T T' (Formalism.MainNotions.S.ptMat (nullPt σ T T' r)) = 0

theorem **Gq_mul_eps** (T T' : E) (Z : Matrix (Fin (r + 1)) (Fin (r + 1)) E) : Gq σ T T' (Z * eps E r) = Gq σ T T' Z

theorem **LiesAt.Gq_eq** {w : Formalism.MainNotions.S.Point E (r + 1)} {Z : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (h : LiesAt w Z) (T T' : E) : Gq σ T T' (Formalism.MainNotions.S.ptMat w) = Gq σ T T' Z

theorem **Col.lat_mem** (c : Col σ J P) : c.lat ∈ FrameLats σ J P

theorem **Col.existsUnique** (hJ : IsUnit J.det) (h : Λ ∈ FrameLats σ J P) : ∃! c : Col σ J P, ∃ s, InS σ J P s ∧ latMap s Λ = c.lat

theorem **lat_mem** : t.lat ∈ FrameLats σ J P

theorem **adm** : admitted σ J T' g t.lat P t.fr

theorem **mk_fr** : mk t.col t.fr t.adm = t

theorem **mk_eq_mk** (hJ : IsUnit J.det) (c : Col σ J P) (hA : admitted σ J T' g c.lat P A) (hB : admitted σ J T' g c.lat P B) : mk c A hA = mk c B hB ↔ Move σ J c.lat P A B

theorem **move_fr_mk** (hJ : IsUnit J.det) (c : Col σ J P) (hA : admitted σ J T' g c.lat P A) : Move σ J c.lat P (mk c A hA).fr A

theorem **rep_mk** (hJ : IsUnit J.det) (hP : IsFrame σ J T P) (c : Col σ J P) (hA : admitted σ J T' g c.lat P A) : (mk c A hA).Rep c.lat A

theorem **rep_self** : t.Rep t.lat t.fr

theorem **exists_rep** (hJ : IsUnit J.det) (hP : IsFrame σ J T P) (h : Λ ∈ FrameLats σ J P) (hA : admitted σ J T' g Λ P A) : ∃ t : Term σ J T' g P, t.Rep Λ A

theorem **Rep.minor_eq** (h : t.Rep Λ A) (l : ℕ) (hl : l ≤ r + 1) : t.minor l hl = SgAlg.lead (pairM σ J A P) l hl

theorem **Rep.G_eq** (h : t.Rep Λ A) (hP : IsFrame σ J T P) : t.G = Gq σ T T' (pairM σ J A P)

theorem **Rep.relevant_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) : t.relevant ↔ Relevant σ J P A

theorem **Rep.inS_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) : t.inS ↔ Unip P A

theorem **Rep.sameFlag_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) : t.sameFlag ↔ SameFlag P A

theorem **Rep.at_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) (w : Formalism.MainNotions.S.Point E (r + 1)) : t.At w ↔ LiesAt w (pairM σ J A P)

theorem **Rep.ptOf_eq** (h : t.Rep Λ A) (hJ : IsUnit J.det) : t.ptOf = ptOrNull σ (normOf σ J P) T' (pairM σ J A P)

theorem **Rep.BBne_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) (hP : IsFrame σ J T P) (hΛ : Λ ∈ FrameLats σ J P) (F : Type) [Field F] [Algebra F E] (p : HeightOneSpectrum (𝒪 F)) : t.BBne F p ↔ DefectAt F (defLat g Λ P A) Λ p

theorem **Rep.prim_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) (hP : IsFrame σ J T P) (hΛ : Λ ∈ FrameLats σ J P) : t.prim ↔ PrimIn (defLat g Λ P A) Λ

theorem **Rep.generic_iff** (h : t.Rep Λ A) (hJ : IsUnit J.det) (hP : IsFrame σ J T P) (hΛ : Λ ∈ FrameLats σ J P) : t.generic ↔ Gq σ T T' (pairM σ J A P) ≠ 0 ∧ PrimIn (defLat g Λ P A) Λ

theorem **G_eq_minor** (hP : IsFrame σ J T P) : t.G = t.minor (r + 1) le_rfl ∗ σ (t.minor (r + 1) le_rfl) − T ∗ T' ∗ (t.minor r (Nat.le_succ r) ∗ σ (t.minor r (Nat.le_succ r)))

theorem **algebraMap_GF** (F : Type) [Field F] [Algebra F E] (hσ : ∀ x, σ (σ x) = x) (hF : ∀ x, σ x = x → x ∈ (algebraMap F E).range) (hP : IsFrame σ J T P) (hT : σ T = T) (hT' : σ T' = T') : algebraMap F E (t.GF F) = t.G

theorem **M_eq** (hg : g ≠ 0) {u : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (hu : u ∈ UO E r) (h : Cong g t.lat (P ∗ u) t.fr) : t.M = pairLat g (P ∗ u) t.fr

theorem **M_le** : t.M ≤ t.lat

theorem **prim_iff_BB** : t.prim ↔ Subsingleton t.BB

theorem **prim_iff_BBne** (F : Type) [Field F] [NumberField F] [Algebra F E] : t.prim ↔ ∀ p, ¬ t.BBne F p

theorem **generic_iff** : t.generic ↔ t.G ≠ 0 ∧ t.prim

theorem **generic_iff_BB** : t.generic ↔ t.G ≠ 0 ∧ Subsingleton t.BB

theorem **generic_iff_BBne** (F : Type) [Field F] [NumberField F] [Algebra F E] : t.generic ↔ t.G ≠ 0 ∧ ∀ p, ¬ t.BBne F p

theorem **generic.mem_genConf** {t : Term σ J T' g P} (h : t.generic) (hg : g ≠ 0) (hP : IsFrame σ J T P) : ∃ u ∈ UO E r, ((t.lat, P ∗ u, t.fr) : Tri E n r) ∈ GenConf σ J T T' g (cT σ u ∗ t.Zhat)

theorem **at_ptOf** (h : ∃ w, t.At w) : t.At t.ptOf

theorem **At.Gq_eq** {t : Term σ J T' g P} {w : Formalism.MainNotions.S.Point E (r + 1)} (h : t.At w) (hP : IsFrame σ J T P) : Gq σ T T' (Formalism.MainNotions.S.ptMat w) = t.G

theorem **Gq_ptOf_of_not_at** (hP : IsFrame σ J T P) {ζ : E} (hζ : ζ ≠ 0) (h : ζ ∗ σ ζ = T ∗ T') (k : ¬ ∃ w, t.At w) : Gq σ T T' (Formalism.MainNotions.S.ptMat t.ptOf) = 0

theorem **Gq_ptOf_ne_zero_iff** (hP : IsFrame σ J T P) {ζ : E} (hζ : ζ ≠ 0) (h : ζ ∗ σ ζ = T ∗ T') : Gq σ T T' (Formalism.MainNotions.S.ptMat t.ptOf) ≠ 0 ↔ t.G ≠ 0 ∧ t.At t.ptOf

theorem **isId_iff_FeqF** (hJ : IsUnit J.det) (hP : IsFrame σ J T P) : t.isId ↔ t.FeqF

theorem **inS_iff_FeqF** (hJ : IsUnit J.det) (hP : IsFrame σ J T P) : t.inS ↔ t.FeqF

theorem **inS_iff_of_carries** (hJ : IsUnit J.det) (hP : IsFrame σ J T P) {c' : Col σ J P} {γ : Matrix (Fin n) (Fin n) E} (h : t.Carries c' γ) : t.inS ↔ InS σ J P γ

theorem **inS.props** {t : Term σ J T' g P} (h : t.inS) (hP : IsFrame σ J T P) (hT : σ T = T) : t.sameFlag ∧ t.relevant ∧ t.G = 0 ∧ T' = T

theorem **inS.prim** {t : Term σ J T' g P} (h : t.inS) (hP : IsFrame σ J T P) : t.prim

theorem **existsUnique_isId** (hP : IsFrame σ J T P) (c : Col σ J P) : ∃! t : Term σ J T g P, t.col = c ∧ t.isId

theorem **exists_generic** (hσ : ∀ x, σ (σ x) = x) (hJ : IsHermU σ J) (hg : g ≠ 0) {Z : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (h : ((Λ, P, A) : Tri E n r) ∈ GenConf σ J T T' g Z) (hG : Gq σ T T' Z ≠ 0) : ∃ t : Term σ J T' g P, t.Rep Λ A ∧ t.generic ∧ t.G = Gq σ T T' Z

theorem **exists_term_plane** (σ : E →+* E) (hσ : ∀ x, σ (σ x) = x) {g : ℕ} (hg : g ≠ 0) : ∃ t : Term (r

theorem **exists_term_split** (σ : E →+* E) (hσ : ∀ x, σ (σ x) = x) (τ : E →+* ℂ) (hτ : ∀ x, τ (σ x) = starRingEnd ℂ (τ x)) {g : ℕ} (hg : g ≠ 0) : (∃ t : Term σ (Inhab.splitJ E) (1 + 2 ∗ g) g (Inhab.splitP E), t.generic) ∧ ∃ t : Term σ (Inhab.splitJ E) 1 g (Inhab.splitP E), t.isId

theorem **hT_eq_sum** (V : HSpace E) (x y : V.V) : V.hT x y = ∑ j, (∑ i, star (V.frame.repr x i) ∗ V.gram i j) ∗ V.frame.repr y j

theorem **pairM_coMat** (V : HSpace E) {a b : Type} (x : a → V.V) (y : b → V.V) (i : a) (j : b) : pairM (starRingEnd E) V.gram (coMat V y) (coMat V x) i j = V.hD (y j) (x i)

theorem **isHermU_gram** (V : HSpace E) : IsHermU (starRingEnd E) V.gram

theorem **isFrame_spFrame** (S : Spaces R F E) {c : E} (hc : c ≠ 0) : IsFrame (starRingEnd E) (S.Y 0).gram (c ∗ star c ∗ S.T) (spFrame S c)

theorem **recFrame_eq** (u : D.what₂.Word) : recFrame D u = coMat (D.sp.Y 0) (Fin.snoc (α

theorem **isFrame_recFrame** (u : D.what₂.Word) : IsFrame (starRingEnd D.host.E) (D.sp.Y 0).gram (((D.mem u).A.T : ℕ) : D.host.E) (recFrame D u)

theorem **level_ne_zero** : D.host.g 0 ∗ D.host.g 1 ≠ 0

theorem **RecTerm.algebraMap_GF** (t : RecTerm D u u') : algebraMap D.host.F D.host.E (t.GF D.host.F) = t.minor (D.R.rS + 1) le_rfl ∗ star (t.minor (D.R.rS + 1) le_rfl) − ((D.mem u).A.T : D.host.E) ∗ ((D.mem u').A.T : D.host.E) ∗ (t.minor D.R.rS (Nat.le_succ _) ∗ star (t.minor D.R.rS (Nat.le_succ _)))

theorem **RecTerm.GF_eq_zero_iff** (t : RecTerm D u u') : t.GF D.host.F = 0 ↔ t.G = 0

theorem **RecTerm.isId_props** (t : RecTerm D u u') : (t.isId ↔ t.FeqF) ∧ (t.inS ↔ t.FeqF) ∧ (t.isId → t.relevant ∧ t.GF D.host.F = 0)

theorem **exists_norm_eq** (u u' : D.what₂.Word) : ∃ ζ : D.host.E, ζ ≠ 0 ∧ ζ ∗ starRingEnd D.host.E ζ = ((D.mem u).A.T : D.host.E) ∗ ((D.mem u').A.T : D.host.E)

theorem **RecTerm.Gq_ptOf_ne_zero_iff** (t : RecTerm D u u') : Gq (starRingEnd D.host.E) ((D.mem u).A.T : D.host.E) ((D.mem u').A.T : D.host.E) (Formalism.MainNotions.S.ptMat t.ptOf) ≠ 0 ↔ t.G ≠ 0 ∧ t.At t.ptOf

theorem **RecTerm.isId_prim** (t : RecTerm D u u') (h : t.isId) : t.prim ∧ Gq (starRingEnd D.host.E) ((D.mem u).A.T : D.host.E) ((D.mem u').A.T : D.host.E) (Formalism.MainNotions.S.ptMat t.ptOf) = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.GrNormPosOfBasis

theorem **sp_comp** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **sp_epsMul_one** (R : Type) [CommRing R] : spMap (epsMul (1 : R)) = 𝟙 _

theorem **torusChar_epsVec** {O : Type} [CommRing O] (x : Sp O ⟶ X) (q : ℕ) (a : 𝒪 E) (I : Ideal O) (c : 𝒪 E →+ O ⧸ I) : (((torusChar H x q a).χ (DualNumber (O ⧸ I)) (GlobalTheory.Diag.epsVec _ c) : (DualNumber (O ⧸ I))ˣ) : DualNumber (O ⧸ I)).snd = c a

theorem **exists_comb** (hu : ∀ k, IsTanAtG t x (u k)) (cf : Fin m → R) : ∃ w, IsTanAtG t x w ∧ IsCombAtG t x u cf w

theorem **comb_unique** (hu : ∀ k, IsTanAtG t x (u k)) {cf : Fin m → R} {w w' : Sp (DualNumber R) ⟶ t.G.left} (h : IsCombAtG t x u cf w) (h' : IsCombAtG t x u cf w') : w = w'

theorem **comb_tan** (hu : ∀ k, IsTanAtG t x (u k)) {cf : Fin m → R} {w : Sp (DualNumber R) ⟶ t.G.left} (h : IsCombAtG t x u cf w) : IsTanAtG t x w

theorem **comb_smul** (hu : ∀ k, IsTanAtG t x (u k)) {cf : Fin m → R} {w : Sp (DualNumber R) ⟶ t.G.left} (h : IsCombAtG t x u cf w) (a : R) : IsCombAtG t x u (fun k => a * cf k) (spMap (epsMul a) ≫ w)

theorem **indep_iff** {n : ℕ} {v : Fin n → (Sp (DualNumber R) ⟶ t.G.left)} (hv : ∀ k, IsTanAtG t x (v k)) (hi : Indep t x v) (A : Matrix (Fin n) (Fin m) R) (hA : ∀ i, IsCombAtG t x v (fun j => A j i) (u i)) : Indep t x u ↔ Function.Injective A.mulVec

theorem **indep_of_basis** (hu : ∀ k, IsTanAtG t x (u k)) (hb : ∀ w, IsTanAtG t x w → ∃! cf, IsCombAtG t x u cf w) : Indep t x u

theorem **exists_left_inv** {K : Type} [Field K] {n m : ℕ} (A : Matrix (Fin n) (Fin m) K) (h : Function.Injective A.mulVec) : ∃ B : Matrix (Fin m) (Fin n) K, B * A = 1

theorem **Indep.map** {K K' : Type} [Field K] [Field K'] (io : K →+* K') {x : Sp K ⟶ X} {m : ℕ} {u : Fin m → (Sp (DualNumber K) ⟶ t.G.left)} (hu : ∀ k, IsTanAtG t x (u k)) (h : Indep t x u) : Indep t (spMap io ≫ x) fun k => spMap (dualMap io) ≫ u k

theorem **grNorm_eq** (i : Fin m) : grNorm t x Λ u i = sInf (grSet t x Λ u i)

theorem **grNorm_pos_of_basis** {n : ℕ} {v : Fin n → (Sp (DualNumber C) ⟶ t.G.left)} (hv : ∀ k, IsTanAtG t x (v k)) (hb : ∀ w, IsTanAtG t x w → ∃! cf, IsCombAtG t x v cf w) (h1 : ∀ cf w, IsCombAtG t x v cf w → (∀ j, ‖cf j‖ ≤ 1) → w ∈ Λ) {b : ℝ} (h2 : ∀ cf w, IsCombAtG t x v cf w → w ∈ Λ → ∀ j, ‖cf j‖ ≤ b) (hu : ∀ k, IsTanAtG t x (u k)) (hi : Indep t x u) (i : Fin m) : (grSet t x Λ u i).Nonempty ∧ 0 < grNorm t x Λ u i

theorem **tri_mulVec** {K : Type} [Field K] {A : Matrix (Fin m) (Fin m) K} (hA : A.BlockTriangular id) {i : Fin m} {cf : Fin m → K} (h : ∀ k, i < k → cf k = 0) : (∀ k, i < k → A.mulVec cf k = 0) ∧ A.mulVec cf i = A i i * cf i

theorem **grNorm_tri** (hu : ∀ k, IsTanAtG t x (u k)) {d : Fin m → C} (hd : ∀ i, d i ≠ 0) (h : ∀ i, ∃ cf : Fin m → C, cf i = d i ∧ (∀ k, i < k → cf k = 0) ∧ IsCombAtG t x u cf (u' i)) (i : Fin m) : grNorm t x Λ u' i = ‖d i‖ * grNorm t x Λ u i

theorem **exists_flagPt** (y : Sp Ω ⟶ gen Z) : ∃ z : FlagPt Z Ω, z.y = y

theorem **indep_u** (y : FlagPt Z Ω) (τ : E →+* ℂ) : Indep Z.t (y.y ≫ genι Z) (y.u τ)

theorem **SameFlags.refl** (y : FlagPt Z Ω) : SameFlags y y 1

theorem **SameFlags.grNorm** {y y' : FlagPt Z Ω} {d : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Ωˣ} (h : SameFlags y y' d) {C : Type} [NontriviallyNormedField C] (io : Ω →+* C) (Λ : Set (Sp (DualNumber C) ⟶ Z.t.G.left)) (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) : grNorm Z.t (spMap io ≫ y'.y ≫ genι Z) Λ (fun k => spMap (dualMap io) ≫ y'.u τ k) i = ‖io (d τ i)‖ * grNorm Z.t (spMap io ≫ y.y ≫ genι Z) Λ (fun k => spMap (dualMap io) ≫ y.u τ k) i

theorem **intO_valRing** (C : Type) [NormedField C] [IsUltrametricDist C] : ValuationRing (intO C)

theorem **intO_frac** (C : Type) [NormedField C] [IsUltrametricDist C] : IsFractionRing (intO C) C

theorem **existsUnique_ext** (x : Sp C ⟶ Z.X) : ∃! xO : Sp ↥(intO C) ⟶ Z.X, spMap (intO C).subtype ≫ xO = x

theorem **grNorm_latDR_pos** (x : Sp C ⟶ Z.X) {m : ℕ} {u : Fin m → (Sp (DualNumber C) ⟶ Z.t.G.left)} (hu : ∀ k, IsTanAtG Z.t x (u k)) (hi : Indep Z.t x u) (i : Fin m) : (grSet Z.t x (latDR Z.t x) u i).Nonempty ∧ 0 < grNorm Z.t x (latDR Z.t x) u i

theorem **grNorm_latHT_pos** (hB : HTBoundSt) (p : ℕ) [hp : Fact p.Prime] (x : Sp ℂ_[p] ⟶ Z.X) {m : ℕ} {u : Fin m → (Sp (DualNumber ℂ_[p]) ⟶ Z.t.G.left)} (hu : ∀ k, IsTanAtG Z.t x (u k)) (hi : Indep Z.t x u) (i : Fin m) : (grSet Z.t x (latHT Z.t p x) u i).Nonempty ∧ 0 < grNorm Z.t x (latHT Z.t p x) u i

theorem **grNorm_lat_pos** {Ω : Type} [Field Ω] (y : FlagPt Z Ω) (p : ℕ) [Fact p.Prime] (io : Ω →+* ℂ_[p]) (b : Bool) (hB : b = true → HTBoundSt) (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) : 0 < grNorm Z.t (spMap io ≫ y.y ≫ genι Z) (lat Z.t p (spMap io ≫ y.y ≫ genι Z) b) (fun k => spMap (dualMap io) ≫ y.u τ k) i

theorem **SameFlags.lnLie** (k : ℕ) : y'.lnLie io p b τ k = y.lnLie io p b τ k + Real.log ‖io (diagAt d τ k)‖

theorem **SameFlags.lnRoot** (i : ℕ) : y'.lnRoot io p b τ i = y.lnRoot io p b τ i + Real.log ‖io (rootFac d τ i)‖

theorem **exists_pt** (k : s.Comp) : ∃ y : Pt s, compPt s y = k

theorem **lnAt_eq** (p : ℕ) [hp : Fact p.Prime] (b : Bool) (y : Pt s) (x : DV.Rt D.what₂.Tau N) : lnAt s p b y x = y.lnRoot (iotaP p) p b (DV.tauEmb D x.1) x.2

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.HPInvEqForm

theorem **upq_iff** {g : GL (Fin p ⊕ Fin q) ℂ} : g ∈ Upq p q ↔ (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ)ᴴ * Jpq p q * g = Jpq p q

theorem **Jpq_sq** : Jpq p q * Jpq p q = 1

theorem **upq_right** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) : (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) * Jpq p q * (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ)ᴴ = Jpq p q

private lemma **Jpq_mul_fromRows** {n : Type*} (C : Matrix (Fin p) n ℂ) (D : Matrix (Fin q) n ℂ) : Jpq p q * fromRows C D = fromRows C (-D)

private lemma **conjTranspose_fromRows_mul_Jpq_mul_fromRows** {n m : Type*} (A : Matrix (Fin p) n ℂ) (B : Matrix (Fin q) n ℂ) (C : Matrix (Fin p) m ℂ) (D : Matrix (Fin q) m ℂ) : (fromRows A B)ᴴ * Jpq p q * fromRows C D = Aᴴ * C - Bᴴ * D

private lemma **fromCols_mul_Jpq** {m : Type*} (A : Matrix m (Fin p) ℂ) (B : Matrix m (Fin q) ℂ) : fromCols A B * Jpq p q = fromCols A (-B)

private lemma **fromCols_mul_Jpq_mul_conjTranspose_fromCols** {m n : Type*} (A : Matrix m (Fin p) ℂ) (B : Matrix m (Fin q) ℂ) (C : Matrix n (Fin p) ℂ) (D : Matrix n (Fin q) ℂ) : fromCols A B * Jpq p q * (fromCols C D)ᴴ = A * Cᴴ - B * Dᴴ

private lemma **blocks_mul_fromRows** {n : Type*} (M : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) (X : Matrix (Fin p) n ℂ) (Y : Matrix (Fin q) n ℂ) : M * fromRows X Y = fromRows (M.toBlocks₁₁ * X + M.toBlocks₁₂ * Y) (M.toBlocks₂₁ * X + M.toBlocks₂₂ * Y)

theorem **kapQ_gram** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) (z : Matrix (Fin p) (Fin q) ℂ) : (kapQ g z)ᴴ * kapQ g z - ((g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₁₁ * z + (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₁₂)ᴴ * ((g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₁₁ * z + (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₁₂) = hQ z

theorem **kapQ_isUnit** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : IsUnit (kapQ g z).det

theorem **moeb_mul_kapQ** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z * kapQ g z = (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₁₁ * z + (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₁₂

theorem **hQ_moeb** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (kapQ g z)ᴴ * hQ (moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z) * kapQ g z = hQ z

theorem **moeb_mem** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z ∈ hcDom p q

private lemma **fromCols_mul_eq_blocks** {m : Type*} (X : Matrix m (Fin p) ℂ) (Y : Matrix m (Fin q) ℂ) (M : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) : fromCols X Y * M = fromCols (X * M.toBlocks₁₁ + Y * M.toBlocks₂₁) (X * M.toBlocks₁₂ + Y * M.toBlocks₂₂)

private lemma **mul_fromCols'** {m k : Type*} [Fintype m] (N : Matrix k m ℂ) (X : Matrix m (Fin p) ℂ) (Y : Matrix m (Fin q) ℂ) : N * fromCols X Y = fromCols (N * X) (N * Y)

theorem **hP_moeb** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : jW g z * hP z * (jW g z)ᴴ = hP (moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z)

theorem **hP_mul_eq_one** {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : hP z * (1 + z * (hQ z)⁻¹ * zᴴ) = 1

theorem **hP_inv_posDef** {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (hP z)⁻¹.PosDef

theorem **hP_posDef** {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (hP z).PosDef

theorem **jW_isUnit** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : IsUnit (jW g z).det

theorem **hP_inv_moeb** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (jW g z)ᴴ * (hP (moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z))⁻¹ * jW g z = (hP z)⁻¹

theorem **autE_isUnit** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : IsUnit (autE g z).det

theorem **mul_frame** {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) * fromBlocks 1 z 0 1 = fromBlocks 1 (moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z) 0 1 * fromBlocks (jW g z) 0 (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ).toBlocks₂₁ (kapQ g z)

theorem **blkMet_posDef** {A : Matrix (Fin p) (Fin p) ℂ} {B : Matrix (Fin q) (Fin q) ℂ} (hA : A.PosDef) (hB : B.PosDef) : (blkMet A B).PosDef

theorem **autE_blkMet** (g : GL (Fin p ⊕ Fin q) ℂ) (z : Matrix (Fin p) (Fin q) ℂ) (A : Matrix (Fin p) (Fin p) ℂ) (B : Matrix (Fin q) (Fin q) ℂ) : (autE g z)ᴴ * blkMet A B * autE g z = blkMet ((jW g z)ᴴ * A * jW g z) ((kapQ g z)ᴴ * B * kapQ g z)

theorem **hodge_posDef** {c : ℝ} (hc : 0 < c) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (hodge c z).PosDef

theorem **hodge_moeb** (c : ℝ) {g : GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ Upq p q) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (autE g z)ᴴ * hodge c (moeb (g : Matrix (Fin p ⊕ Fin q) (Fin p ⊕ Fin q) ℂ) z) * autE g z = hodge c z

theorem **det_hodge** (c : ℝ) {z : Matrix (Fin p) (Fin q) ℂ} (hz : z ∈ hcDom p q) : (hodge c z).det = (c : ℂ) ^ (p + q)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.HermitianCodeWordWeilElementMemQuadraticField

theorem **resCh_eq** {T : ℕ} (hT : T.Prime) (h : Ideal.absNorm P.asIdeal = T) : resCh P = T

theorem **card_res** (P : HeightOneSpectrum (𝒪 E)) : Nat.card (Res P) = Ideal.absNorm P.asIdeal

theorem **psiRes_intCast** {T : ℕ} (hT : T.Prime) (h : Ideal.absNorm P.asIdeal = T) (b : ℤ) : psiRes P (b : Res P) = Complex.exp (2 * Real.pi * Complex.I * b / (T : ℂ))

theorem **exists_intCast_res** {T : ℕ} (hT : T.Prime) (h : Ideal.absNorm P.asIdeal = T) (c : Res P) : ∃ b : ℤ, (b : Res P) = c

theorem **psiWeil_apply** {T : ℕ} (hT : T.Prime) (h : Ideal.absNorm P.asIdeal = T) (ξ : E) (c : Res P) (b : ℤ) (hb : redP P ξ * c = (b : Res P)) : psiWeil P ξ c = Complex.exp (2 * Real.pi * Complex.I * b / (T : ℂ))

theorem **psiRes_ne_one** (P : HeightOneSpectrum (𝒪 E)) : psiRes P ≠ 1

theorem **psiWeil_ne_one** {ξ : E} (h : redP P ξ ≠ 0) : psiWeil P ξ ≠ 1

theorem **psiWeil_pow** (ξ : E) (c : Res P) : psiWeil P ξ c ^ resCh P = 1

theorem **eq_of_sees** {w : HeightOneSpectrum (𝒪 X.K)} {τ : X.Tau} {P P' : Pr.vT u} (h : SeesP w τ P.1 ∨ SeesPbar w τ P.1) (h' : SeesP w τ P'.1 ∨ SeesPbar w τ P'.1) : P = P'

theorem **T_mem_vT** (P : Pr.vT u) : ((Pr.T u : ℕ) : 𝒪 X.E) ∈ P.1.asIdeal

theorem **tauAt_sees** (hX : X.AbelianUnramifiedExtensionConditions) {w : HeightOneSpectrum (𝒪 X.K)} (hw : ((Pr.T u : ℕ) : 𝒪 X.K) ∈ w.asIdeal) (P : Pr.vT u) : SeesP w (tauAt hPr u w P) P.1 ∨ SeesPbar w (tauAt hPr u w P) P.1

theorem **tauAt_mem_iff** (hX : X.AbelianUnramifiedExtensionConditions) {w : HeightOneSpectrum (𝒪 X.K)} (hw : ((Pr.T u : ℕ) : 𝒪 X.K) ∈ w.asIdeal) (P : Pr.vT u) (x : 𝒪 X.F) : X.embO (tauAt hPr u w P) (algebraMap (𝒪 X.F) (𝒪 X.E) x) ∈ w.asIdeal ↔ x ∈ (P.1.under (𝒪 X.F)).asIdeal

theorem **seesP_tauAt** (hX : X.AbelianUnramifiedExtensionConditions) {w : HeightOneSpectrum (𝒪 X.K)} (hs : seenC Pr u w) (P : Pr.vT u) : SeesP w (tauAt hPr u w P) P.1

theorem **seesPbar_tauAt** (hX : X.AbelianUnramifiedExtensionConditions) {w : HeightOneSpectrum (𝒪 X.K)} (hw : ((Pr.T u : ℕ) : 𝒪 X.K) ∈ w.asIdeal) (hs : ¬ seenC Pr u w) (P : Pr.vT u) : SeesPbar w (tauAt hPr u w P) P.1

theorem **normFock_self_re** (f : MvPolynomial (Fin N × Fin N) ℂ) : (normFock lam f f).re = (fockS f f).re / (fockS (extremal ℂ N lam) (extremal ℂ N lam)).re

theorem **normFock_self_pos** {f : MvPolynomial (Fin N × Fin N) ℂ} (hf : f ≠ 0) : 0 < (normFock lam f f).re

theorem **fockRatios_bddAbove** : BddAbove (fockRatios lam sg b)

theorem **fockRatios_nonempty** : (fockRatios lam sg b).Nonempty

theorem **fockOpNorm_nonneg** : 0 ≤ fockOpNorm lam sg b

theorem **fockOpNorm_le** {C : ℝ} (hC : 0 ≤ C) (h : ∀ f ∈ delSq ℂ N lam, (normFock sg (b f) (b f)).re ≤ C ^ 2 * (normFock lam f f).re) : fockOpNorm lam sg b ≤ C

theorem **fockOpNorm_congr** {lam lam' : DomWeight N} {sg sg' : DomWeight n} (h : lam.natW = lam'.natW) (h' : sg.natW = sg'.natW) : fockOpNorm lam sg b = fockOpNorm lam' sg' b

theorem **emb_mem_Qbar** (x : K) : σ₀ x ∈ TaylorLogMetric.Qbar

theorem **exists_embQbar** : ∃ ρ : TaylorLogMetric.Qbar →₀[ℚ] ℂ, ∀ x : K, StandardLatticesGL3.cj TaylorLogMetric.Qbar ρ (σ₀ x) = s x

theorem **eNorm_nonneg** (s : X.K →+* ℂ) (N : Submodule (𝒪 X.E) (Sp X.E a)) (N' : Submodule (𝒪 X.E) (Sp X.E b)) : 0 ≤ eNorm A s N N'

theorem **val_pos_iff_mem** (w : Pl.Pl) (x : 𝒪 X.K) : (0 : WithTop ℤ) < Pl.val w (x : X.K) ↔ x ∈ (plEq w).asIdeal

theorem **resChar_mem** (w : Pl.Pl) : ((Pl.resChar w : ℕ) : 𝒪 X.K) ∈ (plEq w).asIdeal

theorem **exists_int_red** (u : Pr.Word) (hi : ∀ P : Pr.vT u, IntAt P.1 ξ) : ∃ c : ℤ, ∀ P : Pr.vT u, redP P.1 ξ = (c : Res P.1)

theorem **psiWeil_same** (u : Pr.Word) (hi : ∀ P : Pr.vT u, IntAt P.1 ξ) (P P' : Pr.vT u) (a : ℤ) : psiWeil P.1 ξ (a : Res P.1) = psiWeil P'.1 ξ (a : Res P'.1)

theorem **exists_charT** (σ₀ : X.K →+* ℂ) (hr : ∀ u (z : ℂ), z ^ Pr.T u = 1 → ∃ y, σ₀ y = z) (hu : ∀ u, XiUnitAt (Pr.vT u) ξ) : ∃ Ps : CharT X Pr, Ps.Ok ∧ ∀ u P c, σ₀ (Ps.psiT u P c) = psiWeil P.1 ξ c

theorem **hermitianCodeWord_weil_element_mem_quadraticField** : B.host.xi0 ∈ B.fields.Kq

theorem **xiUnitAt_vT** (X : PreX B.toPre) (u : B.Word) : XiUnitAt (X

theorem **crv_of_ok** (hJ : J.Ok) : StandardLatticesGL3.Crv (assemble B C W V S N G) ⊤ J.δ

theorem **admissible_assembled_evenRanks** (i j : ℕ) {ST SG ST' SG' : Finset (HeightOneSpectrum (𝒪 B.fields.E))} : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.HodgeGramEqRiemann

theorem **im_hermForm_orth** (v w : (W.F.Di (W.ik k)).La) : (hermForm (((2 * a : ℝ) : ℂ) • 1) (W.Pm k x v) (W.Pm k x w)).im = (W.F.Di (W.ik k)).EF v w

theorem **hermForm_eq_of_orth** {i : Fin W.F.n} {P : (W.F.Di i).La →+ (Fin W.g → ℂ)} (hP : IsPeriod (W.F.Di i) (fun j => Formalism.L2.PjFree.relab c σ (W.e j).1) P) {Hm : Matrix (Fin W.g) (Fin W.g) ℂ} (hH : ∀ v w, (hermForm Hm (P v) (P w)).im = (W.F.Di i).EF v w) {y : FrPt Xo c (CAt σ)} {z : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Fin W.g → ℂ)} (hy : AnLie W.F i P y z) (e : y.x = (U.un k x).x) {τ : E →+* ℂ} {T : Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) ℂ} (hT : ∀ b, IsCombAt (U.un k x).x ((U.un k x).u τ) (show Fin (Dw.mult τ) → CAt σ from fun b' => T b' b) (y.u τ b)) : (of fun b b' => hermForm Hm (z τ b') (z τ b)) = ((2 * a : ℝ) : ℂ) • (Tᴴ * T)

theorem **isCombAt_co** {y y' : FrPt Xo c (CAt σ)} (e : y'.x = y.x) {g : EigFrame Z σ ((cxPt Z y).1 ≫ genι Z)} (hg : ∀ τ i, g.v τ i = (cxFrame Z y').v τ i) (τ : E →+* ℂ) (b : Fin (Dw.mult τ)) : IsCombAt y.x (y.u τ) (show Fin (Dw.mult τ) → CAt σ from fun b' => (cxFrame Z y).co g τ b' b) (y'.u τ b)

theorem **x_eq_of_cxPt** {y y' : FrPt Xo c (CAt σ)} (h : (cxPt Z y).1 = (cxPt Z y').1) : y.x = y'.x

theorem **spMap_dualMap_id** (R : Type) [CommRing R] : spMap (dualMap (RingHom.id R)) = 𝟙 _

theorem **isHodgeFrame_cxFrame** (hσ : Formalism.L2.PjFree.GoodEmb c σ) {k : (U σ).Comp} {x : ((U σ).X k).dom} (hx : IsOrth ((U σ).X k) x.1) : IsHodgeFrame U (cxPt Z ((U σ).un k x)) (cxFrame Z ((U σ).un k x))

theorem **liesIn_iff_of_comp** {σ' : c.F →+* ℂ} {y' : CxPts (genStr Z) σ'} {ψ : ℂ →+* ℂ} (e : y'.1 = spMap ψ ≫ y.1) (C : Closeds (gen Z)) : LiesIn y' C ↔ LiesIn y C

theorem **IsHodgeFrame.not_liesIn** (h : IsHodgeFrame U y g) : ¬ LiesIn y (genBd Z)

theorem **IsHodgeFrame.conj** (h : IsHodgeFrame U y g) {σ₁ : c.F →+* ℂ} {y₁ : CxPts (genStr Z) σ₁} {g₁ : EigFrame Z σ₁ (y₁.1 ≫ genι Z)} (hσ : σ = (starRingEnd ℂ).comp σ₁) (e : y.1 = spMap (starRingEnd ℂ) ≫ y₁.1) (hg : g₁.IsMap (starRingEnd ℂ) g) : IsHodgeFrame U y₁ g₁

theorem **exists_isHodgeFrame** (hy : ¬ LiesIn y (genBd Z)) : ∃ g, IsHodgeFrame U y g

theorem **IsHodgeFrame.unitary** (hf : IsHodgeFrame U y f) (hg : IsHodgeFrame U y g) (τ : E →+* ℂ) : (f.co g τ)ᴴ * f.co g τ = 1

theorem **IsHodgeFrame.gram_eq** (hf : IsHodgeFrame U y f) (g : EigFrame Z σ (y.1 ≫ genι Z)) (τ : E →+* ℂ) : (EigMet.hodge U b hb G₀).gram y g τ = (b : ℂ) • ((f.co g τ)ᴴ * f.co g τ)

theorem **hodge_gram_eq_riemann** (hσ : Formalism.L2.PjFree.GoodEmb c σ) (W : UnfWit (U σ)) {i : Fin W.F.n} {P : (W.F.Di i).La →+ (Fin W.g → ℂ)} (hP : IsPeriod (W.F.Di i) (fun j => Formalism.L2.PjFree.relab c σ (W.e j).1) P) {Hm : Matrix (Fin W.g) (Fin W.g) ℂ} (hH : IsRiem W.F i P Hm) {q : FrPt Xo c (CAt σ)} {z : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Fin W.g → ℂ)} (hq : AnLie W.F i P q z) (τ : E →+* ℂ) (hb : 0 < 2 * Real.pi * a) : (EigMet.hodge U _ hb G₀).gram (cxPt Z q) (cxFrame Z q) τ = (Real.pi : ℂ) • of fun b b' => hermForm Hm (z τ b') (z τ b)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.HomAppHomOfRestrictImage

@[simp] theorem **homEval_apply** (s : Γ(F, V)) (ψ : Γ(homMod F G, V)) : homEval G s ψ = homApp ψ le_rfl s

theorem **homApp_homIns** {e : ι → Γ(F, U)} (he : IsFrame e) (h : V ≤ U) (i : ι) (n : Γ(G, V)) (hW : W ≤ V) (s : Γ(F, W)) : homApp (homIns he h i n) hW s = (he.basis (hW.trans h)).repr s i • resM G hW n

theorem **sum_homIns** {e : ι → Γ(F, U)} (he : IsFrame e) (h : V ≤ U) (ψ : Γ(homMod F G, V)) : ∑ i, homIns he h i (homApp ψ le_rfl (resM F h (e i))) = ψ

theorem **homPi_apply** {e : ι → Γ(F, U)} (he : IsFrame e) (h : V ≤ U) (ψ : Γ(homMod F G, V)) (i : ι) : homPi he h ψ i = homApp ψ le_rfl (resM F h (e i))

theorem **homPi_res** {e : ι → Γ(F, U)} (he : IsFrame e) (h : V ≤ U) (h' : W ≤ V) (ψ : Γ(homMod F G, V)) (i : ι) : homPi he (h'.trans h) (resM _ h' ψ) i = resM G h' (homPi he h ψ i)

theorem **isFiniteQC_homMod** {r : ℕ} (hF : IsVB F r) [IsFiniteQC G] : IsFiniteQC (homMod F G)

theorem **isVB_homMod** {r : ℕ} (hF : IsVB F 1) (hG : IsVB G r) : IsVB (homMod F G) r

theorem **image_preimage_ι** (h : W ≤ U) : U.ι ''ᵁ U.ι ⁻¹ᵁ W = W

theorem **appIm_res** (φ : F.restrict U.ι ⟶ G.restrict U.ι) {T T' : U.toScheme.Opens} (h : T' ≤ T) (s : Γ(F, U.ι ''ᵁ T)) : appIm φ T' (resM F (U.ι.image_mono h) s) = resM G (U.ι.image_mono h) (appIm φ T s)

theorem **homApp_homOfRestrict** (φ : F.restrict U.ι ⟶ G.restrict U.ι) (h : W ≤ U) (s : Γ(F, W)) : homApp (homOfRestrict φ) h s = resM G (image_preimage_ι h).ge (appIm φ (U.ι ⁻¹ᵁ W) (resM F (image_preimage_ι h).le s))

theorem **homApp_homOfRestrict_image** (φ : F.restrict U.ι ⟶ G.restrict U.ι) (T : U.toScheme.Opens) (h : U.ι ''ᵁ T ≤ U) (s : Γ(F, U.ι ''ᵁ T)) : homApp (homOfRestrict φ) h s = appIm φ T s

theorem **homOfRestrict_comp** (φ : F.restrict U.ι ⟶ G.restrict U.ι) (ψ : G.restrict U.ι ⟶ H.restrict U.ι) (h : W ≤ U) (s : Γ(F, W)) : homApp (homOfRestrict ψ) h (homApp (homOfRestrict φ) h s) = homApp (homOfRestrict (φ ≫ ψ)) h s

theorem **homOfRestrict_id** (h : W ≤ U) (s : Γ(F, W)) : homApp (homOfRestrict (𝟙 (F.restrict U.ι))) h s = s

theorem **evz_res** (z : Sp R ⟶ X) (h : ⊤ ≤ z ⁻¹ᵁ V) (h' : ⊤ ≤ z ⁻¹ᵁ U) (hle : V ≤ U) (a : Γ(X, U)) : evz z h (resG X hle a) = evz z h' a

theorem **evz_comp** (ψ : R →+* R') (z : Sp R ⟶ X) (h : ⊤ ≤ z ⁻¹ᵁ U) (h' : ⊤ ≤ (spMap ψ ≫ z) ⁻¹ᵁ U) (a : Γ(X, U)) : evz (spMap ψ ≫ z) h' a = ψ (evz z h a)

theorem **top_le_preimage_iff** [IsLocalRing R] (z : Sp R ⟶ X) : ⊤ ≤ z ⁻¹ᵁ U ↔ z (IsLocalRing.closedPoint R) ∈ U

theorem **exists_extend** [F.IsQuasicoherent] (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) {ε : Γ(X, V) →+* R} {ε' : Γ(X, U) →+* R} (hε : ∀ a, ε (resG X h a) = ε' a) (l : Γ(F, U) →ₗ[ε'] R) : ∃ μ : Γ(F, V) →ₗ[ε] R, ∀ s, μ (resM F h s) = l s

theorem **form_ext** [F.IsQuasicoherent] (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) {ε : Γ(X, V) →+* R} {μ μ' : Γ(F, V) →ₗ[ε] R} (H : ∀ s, μ (resM F h s) = μ' (resM F h s)) : μ = μ'

theorem **IsLineForm.comp** {A M N : Type} [CommRing A] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module A N] {ε : A →+* R} {μ₀ : M →ₗ[ε] R} (h : IsLineForm μ₀) (φ : N ≃ₗ[A] M) : IsLineForm (μ₀.comp φ.toLinearMap)

theorem **exists_isLineForm** [IsLocalRing R] [F.IsQuasicoherent] (z : Sp R ⟶ X) (hV : IsAffineOpen V) (h : ⊤ ≤ z ⁻¹ᵁ V) {e : Fin 1 → Γ(F, W)} (he : IsFrame e) (hW : z (IsLocalRing.closedPoint R) ∈ W) : ∃ μ₀ : Γ(F, V) →ₗ[evz z h] R, IsLineForm μ₀

theorem **exists_isGreatest** {A M C : Type} [CommRing A] [AddCommGroup M] [Module A M] [Module.Finite A M] [NormedField C] [IsUltrametricDist C] {ε : A →+* C} (hε : ∀ a, ‖ε a‖ ≤ 1) (ν : M →ₗ[ε] C) : ∃ c, IsGreatest (Set.range fun m => ‖ν m‖) c

theorem **isLocalRing_intO** (C : Type) [NormedField C] [IsUltrametricDist C] : IsLocalRing (intO C)

theorem **nrmSet_subset** (φ : F ≅ G) (f : Γ(F, gen X p)) : nrmSet z F f ⊆ nrmSet z G (φ.hom.app _ f)

theorem **nrmSet_iso** (φ : F ≅ G) (f : Γ(F, gen X p)) : nrmSet z G (φ.hom.app _ f) = nrmSet z F f

theorem **inf_gen_eq** (U : X.Opens) : U ⊓ gen X p = X.basicOpen ((p : ℕ) : Γ(X, U))

theorem **isAffineOpen_inf_gen** (hU : IsAffineOpen U) : IsAffineOpen (U ⊓ gen X p)

theorem **zq_le_gen** : ⊤ ≤ zq z ⁻¹ᵁ gen X p

theorem **zq_le** (h : ⊤ ≤ z ⁻¹ᵁ U) : ⊤ ≤ zq z ⁻¹ᵁ (U ⊓ gen X p)

theorem **evq_res** (h : ⊤ ≤ z ⁻¹ᵁ U) (a : Γ(X, U)) : evq z hp h (resG X inf_le_left a) = ((intO C.S).subtype.comp (evz z h)) a

theorem **norm_nsmul_mul** {M A : Type} [CommRing A] [AddCommGroup M] [Module A M] {ε : A →+* C.S} (μ : M →ₗ[ε] C.S) (k : ℕ) (v : M) : ‖μ (p ^ k • v)‖ * (p : ℝ) ^ k = ‖μ v‖

theorem **nrmSet_eq** [F.IsQuasicoherent] (hU : IsAffineOpen U) (h : ⊤ ≤ z ⁻¹ᵁ U) (f : Γ(F, gen X p)) : nrmSet z F f = {r | ∃ μ : Γ(F, U ⊓ gen X p) →ₗ[evq z hp h] C.S, IsInt z hp μ ∧ r = ‖μ (resM F inf_le_right f)‖}

theorem **exists_isGreatest_vals** [IsFiniteQC F] (hU : IsAffineOpen U) (μ : Γ(F, U ⊓ gen X p) →ₗ[evq z hp h] C.S) : ∃ c, IsGreatest (vals z hp μ) c

theorem **pos_of_isGreatest** : 0 < c

theorem **norm_le_of_isInt** {μ : Γ(F, U ⊓ gen X p) →ₗ[evq z hp h] C.S} (hμ : IsInt z hp μ) (v : Γ(F, U ⊓ gen X p)) : ‖μ v‖ ≤ ‖μ₀ v‖ / c

theorem **exists_isInt** : ∃ μ : Γ(F, U ⊓ gen X p) →ₗ[evq z hp h] C.S, IsInt z hp μ ∧ ∀ v, ‖μ v‖ = ‖μ₀ v‖ / c

theorem **isGreatest_nrmSet** (f : Γ(F, gen X p)) : IsGreatest (nrmSet z F f) (‖μ₀ (resM F inf_le_right f)‖ / c)

theorem **isGreatest_nrmSet_hom** [(homMod G F).IsQuasicoherent] {e : ι → Γ(G, U)} (he : IsFrame e) (ψ : Γ(homMod G F, gen X p)) {c₁ : ℝ} (hc₁ : IsGreatest (vals z hp (μ₀.comp (homApp ψ inf_le_right))) c₁) : IsGreatest (nrmSet z (homMod G F) ψ) (c₁ / c)

theorem **isGreatest_nrmSet_compSec** {E : X.Modules} {r : ℕ} (hE : IsVB E r) (π : E ⟶ F) [Epi π] [IsFiniteQC F] [IsFiniteQC G] (hU : IsAffineOpen U) {e : ι → Γ(E, U)} (he : IsFrame e) (φ : F.restrict (gen X p).ι ≅ G.restrict (gen X p).ι) {μ : Γ(G, U ⊓ gen X p) →ₗ[evq z hp h] C.S} (hμ : IsLineForm μ) {c₁ c₂ : ℝ} (hc₁ : IsGreatest (vals z hp (μ.comp (equivOfRestrict φ inf_le_right).toLinearMap)) c₁) (hc₂ : IsGreatest (vals z hp μ) c₂) : IsGreatest (nrmSet z (homMod E G) (compSec π φ.hom)) (c₁ / c₂)

theorem **exists_isLineForm_gen** [F.IsQuasicoherent] (hF : IsVB (F.restrict (gen X p).ι) 1) (hU : IsAffineOpen U) : ∃ μ : Γ(F, U ⊓ gen X p) →ₗ[evq z hp h] C.S, IsLineForm μ

theorem **sSup_compSec_mul** {E E' : X.Modules} {r r' : ℕ} (hE : IsVB E r) (hE' : IsVB E' r') (π : E ⟶ F) [Epi π] (π' : E' ⟶ G) [Epi π'] [IsFiniteQC F] [IsFiniteQC G] (hG : IsVB (G.restrict (gen X p).ι) 1) (φ : F.restrict (gen X p).ι ≅ G.restrict (gen X p).ι) : 0 < sSup (nrmSet z (homMod E G) (compSec π φ.hom)) ∧ sSup (nrmSet z (homMod E' F) (compSec π' φ.inv)) = (sSup (nrmSet z (homMod E G) (compSec π φ.hom)))⁻¹

theorem **sSup_compSec_eq** {E : X.Modules} {r : ℕ} (hE : IsVB E r) (π : E ⟶ F) [Epi π] [IsFiniteQC G] {u : Γ(F, ⊤)} (hu : IsFrame fun _ : Fin 1 => u) (φ : F.restrict (gen X p).ι ≅ G.restrict (gen X p).ι) : sSup (nrmSet z G (homApp (homOfRestrict φ.hom) le_rfl (resM F le_top u))) = sSup (nrmSet z (homMod E G) (compSec π φ.hom))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.HomShInvOfGIso

theorem **nrm_nonneg** (ct : At S κ p) {Z : ct.Mdl} (F : ct.Sh Z) (f : ct.Sec F) (y : ct.Pt Z) : 0 ≤ ct.nrm F f y

theorem **phi_conj** (ct : At S κ p) {Z : ct.Mdl} (y : ct.Pt Z) (c c' : 𝒪 D.host.E) (h : (c' : D.host.E) = D.host.conj (c : D.host.E)) (a : ct.Ylat y) : ct.phi y (c' • a) = c • ct.phi y a

theorem **mult_tau** (t : D.what₂.Tau) : S.Dw.mult (DV.tauEmb D t) = j

theorem **zeroSh_isZero** (ct : At S κ p) (Z : ct.Mdl) : IsZero (zeroSh ct Z).toMod

theorem **nrm_of_isZero** (ct : At S κ p) {Z : ct.Mdl} (G : ct.Sh Z) (hG : IsZero G.toMod) (f : ct.Sec G) (y : ct.Pt Z) : ct.nrm G f y = 0

theorem **tau_inj** {t t' : D.what₂.Tau} (h : DV.tauEmb D t = DV.tauEmb D t') : t = t'

theorem **conj_not_cm** (A : Amb D) (t : D.what₂.Tau) : conjEmb (DV.tauEmb D t) (A.H.rt : D.host.E) ≠ A.H.rt0

theorem **tau_ne_conj** (A : Amb D) (t t' : D.what₂.Tau) : DV.tauEmb D t ≠ conjEmb (DV.tauEmb D t')

theorem **mult_conj** (t : D.what₂.Tau) : S.Dw.mult (conjEmb (DV.tauEmb D t)) + j = N

theorem **expo_tau** (ct : At S κ p) (l : PsiT D → ℕ → ℤ) (t : D.what₂.Tau) (i : Fin (S.Dw.mult (DV.tauEmb D t))) : ct.expo l (DV.tauEmb D t) i = l (D.eTau t) ((i : ℕ) + 1)

theorem **expo_conj** (ct : At S κ p) (l : PsiT D → ℕ → ℤ) (t : D.what₂.Tau) (i : Fin (S.Dw.mult (conjEmb (DV.tauEmb D t)))) : ct.expo l (conjEmb (DV.tauEmb D t)) i = -l (D.eTau t) (N - (i : ℕ))

theorem **expo_hodge** (ct : At S κ p) (t : D.what₂.Tau) : (∀ i, ct.expo (fun _ k => if k ≤ j then 1 else -1) (DV.tauEmb D t) i = 1) ∧ ∀ i, ct.expo (fun _ k => if k ≤ j then 1 else -1) (conjEmb (DV.tauEmb D t)) i = 1

theorem **isEmpty_gIso_zero** (ct : At S κ p) (Z : ct.Mdl) (F : ct.Sh Z) : IsEmpty (ct.GIso (zeroSh ct Z) F)

theorem **nonempty_gIso_Lb** (ct : At S κ p) (l : PsiT D → ℕ → ℤ) : Nonempty (ct.GIso (ct.Lb l) (ct.Lb l))

theorem **nrm_mul_nrm_symm** (ct : At S κ p) (sh : ShfSt ct) {Z : ct.Mdl} {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : ct.nrm (sh.homSh F₁ F₂) (sh.isoSec φ) y * ct.nrm (sh.homSh F₂ F₁) (sh.isoSec (sh.isoSymm φ)) y = 1

theorem **nrm_secOf_mul** (ct : At S κ p) (sh : ShfSt ct) {F : ct.Sh ct.Xm} (φ : ct.GIso sh.oX F) (y : ct.Pt ct.Xm) : ct.nrm F (sh.secOf φ) y * ct.nrm (sh.homSh F sh.oX) (sh.isoSec (sh.isoSymm φ)) y = 1

theorem **homSh_inv_of_gIso** (ct : At S κ p) (sh : ShfSt ct) {Z : ct.Mdl} {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) : ct.IsInv (sh.homSh F₁ F₂)

theorem **xn_injective** (ct : At S κ p) : Function.Injective ct.Xn

theorem **blowUp_i** (ct : At S κ p) (Z : ct.Mdl) (J : ct.Idl Z) : (ct.blowUp Z J).i = Z.i

theorem **htBlowup_level** (ct : At S κ p) {sh : ShfSt ct} {hq : HTpSt ct} (hm : HTmSt ct sh hq) {n m : ℕ} (hn : 3 ≤ n) (hm' : 3 ≤ m) {Z : ct.Mdl} (h₁ : hm.IsHTBlowup Z (ct.Xn n)) (h₂ : hm.IsHTBlowup Z (ct.Xn m)) : n = m

theorem **drFlag_unique** (ct : At S κ p) {Z' : ct.Mdl} (h : IsDRFlag ct.Xm Z') : (∃ f, IsMap Z' ct.YY f) ∧ ∃ g, IsMap ct.YY Z' g

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.HostPins

theorem **mem_qb** {ι : Qb →+* ℂ} {M : IntermediateField ℚ ℂ} {x : Qb} : x ∈ qb ι M ↔ ι x ∈ M

theorem **ExFld.recFld** {B : Pre} {f : Fr B} {δ : FieldDat B} {F : IntermediateField ℚ ℂ} (h : ExFld B f δ F) (hO : listField δ f ≤ δ.Om) : RecFld B f δ F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ImaginaryRootNeZero

theorem **expo_add** (w w' : Sym E →₀ ℤ) : expo Dw (w + w') = expo Dw w + expo Dw w'

theorem **expo_zero** : expo Dw 0 = 0

theorem **expo_single** (a : Sym E) (n : ℤ) (x : Letter Dw) : expo Dw (Finsupp.single a n) x = if letterOf Dw a.2.2 = some x then n else 0

theorem **imaginary_root_ne_zero** : starRingEnd ℂ H.rt0 ≠ H.rt0

theorem **mult_of_mem** {τ : E →+* ℂ} (h : τ (H.rt : E) = H.rt0) : Dw.mult τ = Dw.p

theorem **mult_conj_of_mem** {τ : E →+* ℂ} (h : τ (H.rt : E) = H.rt0) : Dw.mult (conjEmb τ) = Dw.q

theorem **expo_root_first** (a : ℕ → Bool) (R : Option ℝ) {τ : E →+* ℂ} (h : τ (H.rt : E) = H.rt0) (i : ℕ) (hi : i < Dw.p) : expo Dw (rootSym Dw a R τ i) (τ, (i, by rw [mult_of_mem Dw h]; exact hi)) = 1

theorem **expo_root_second** (a : ℕ → Bool) (R : Option ℝ) {τ : E →+* ℂ} (h : τ (H.rt : E) = H.rt0) (i : ℕ) (hi : Dw.p ≤ i) (hi' : i < Dw.p + Dw.q) : expo Dw (rootSym Dw a R τ i) (conjEmb τ, (Dw.p + Dw.q - 1 - i, by rw [mult_conj_of_mem Dw h]; omega)) = -1

theorem **parOf_eq** {w : Sym E →₀ ℤ} {R : Option ℝ} (h0 : w ≠ 0) (h : ∀ a ∈ w.support, a.2.1 = R) : parOf w = R

theorem **mem_letters** (x : Letter Dw) : x ∈ letters Dw

theorem **length_letters** : (letters Dw).length = ∑ τ : E →+* ℂ, Dw.mult τ

theorem **divOf_add** (n n' : Letter Dw → ℤ) : divOf Y (n + n') = divOf Y n + divOf Y n'

theorem **divOf_eq_neg** (ex : Ix Dw → ℤ) : divOf Y (fun x => ex (some x)) = -Y.LEx ex

theorem **isEqn_divOf** (n : Letter Dw → ℤ) (y : Y.Y.X) : (divOf Y n).IsEqn y (∏ x : Letter Dw, (Y.gr_spec y).choose_spec.choose_spec.choose x.1 x.2 ^ n x)

theorem **residueCharacteristic_prime** (v : HeightOneSpectrum (𝒪 F)) : (resP v).Prime

theorem **mem_setB** (κ : Lv A) (B : ℝ) (hB : 0 ≤ B) (p : ℕ) : p ∈ setB κ B ↔ p.Prime ∧ ((p : ℝ) ≤ B ∨ p = 2 ∨ p ∣ (discr D.host.E).natAbs ∨ p ∣ κ.l.M')

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.InGeoGeoScalIff

theorem **inGeo_geoScal_iff** (g : GlobalTheory.RealGeo.GeoHalf C R ph) (kap lam err) : GlobalTheory.RealGeo.InGeo (geoScal g kap lam err) ↔ GlobalTheory.RealGeo.InGeo g

theorem **baseW_scal** {A : FamIn D.what₂.LL G Pl0 T D.R.m} {R : RealDat D A Ks} {F : Face D A} (h : R.BaseW F) (k l) : (R.upd6 k l).BaseW F

theorem **exAt_scal** {A : FamIn D.what₂.LL G Pl0 T D.R.m} {R : RealDat D A Ks} (e : R.ExAt) (k l) : (R.upd6 k l).ExAt

theorem **nzAt_scal** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (kap lam) (F0 : HeightOneSpectrum (𝒪 D.what₂.LL) → Fin D.R.m → ℝ) (j : Fin D.R.m) (w : HeightOneSpectrum (𝒪 D.what₂.LL)) : NzAt (R.updS kap lam) F F0 j w ↔ NzAt R F F0 j w

theorem **kapNew_eq** (S : RealDat3 D A3 Ks) {κ : ℝ} (h : KapFixed S κ) : kapNew S = κ

theorem **kapNew_of_not** (S : RealDat3 D A3 Ks) (h : ¬ ∃ κ, KapFixed S κ) : kapNew S = S.kap

theorem **kapNew_of_thmB** (S : RealDat3 D A3 Ks) (h : S.PThmB) : kapNew S = S.kap

theorem **kapNew_of_scal** (S : RealDat3 D A3 Ks) (h : S.scal_kap) : kapNew S = S.kap

theorem **isScal_kapNew** (S : RealDat3 D A3 Ks) (F : Face3 D A3) (h0 : S.eigenvalue_at_base_of_scalar) (h1 : S.scal_off F) (h2 : S.zero_part_eigenvalue_convention F) (h3 : S.scal_abs F) (h4 : S.scal_tup) : (S.updS (kapNew S) S.lam3).IsScal F

theorem **pins_scal** (hf : Function.Injective f) {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (h : Pins R (bare3 _ P0 ψ0 P1 ψ1) cG) (e : R.toRealDat.ExAt) (n : NewTup R mu tw) (k l) (hs : ((dat3 f μ R mu tw).updS k l).IsScal (bare3 _ P0 ψ0 P1 ψ1)) : Pins ((dat3 f μ R mu tw).updS k l) (bare3 _ P0 ψ0 P1 ψ1) cG

theorem **pins_kapNew** (hf : Function.Injective f) {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (h : Pins R (bare3 _ P0 ψ0 P1 ψ1) cG) (e : R.toRealDat.ExAt) (n : NewTup R mu tw) : Pins ((dat3 f μ R mu tw).updS (kapNew (dat3 f μ R mu tw)) (dat3 f μ R mu tw).lam3) (bare3 _ P0 ψ0 P1 ψ1) cG

theorem **pins_resRE** (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') (e : z.R.toRealDat.ExAt) : Pins (z.resRE h) (face0 X I hX (usedOf I T Wt') z.Xg z.tr) z.cG

theorem **exAt_resR** (h : Wt' ⊆ Wt) (e : z.R.toRealDat.ExAt) : (z.resR h).toRealDat.ExAt

theorem **exAt_resRE** (h : Wt' ⊆ Wt) (e : z.R.toRealDat.ExAt) : (z.resRE h).toRealDat.ExAt

theorem **thmB_resR** (h : Wt' ⊆ Wt) (hB : z.R.PThmB) : (z.resR h).PThmB

theorem **resRE_of_thmB** (h : Wt' ⊆ Wt) (hB : z.R.PThmB) : z.resRE h = z.resR h

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.InGeoIff

theorem **omPins** {κ : ∀ i : Fin D.R.m, WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ : ∀ (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), (R.MW i σ).Vec} (O : ∀ i, OMDat C.toCore R i Unit fun _ => (A.comap id (f μ)).μ0) (i : Fin D.R.m) (h : OMPins C R κ v₀ O i) : OMPins C (dat2 f μ R mu tw) κ v₀ (fun i => omDat mu tw (O i)) i

theorem **inGeo** {ph : Fin D.R.m → ℂ} {g : GlobalTheory.RealGeo.GeoHalf C R ph} (h : GlobalTheory.RealGeo.InGeo g) : GlobalTheory.RealGeo.InGeo (geoHalf mu tw g)

theorem **inGeo_iff** {ph : Fin D.R.m → ℂ} {g : GlobalTheory.RealGeo.GeoHalf C R ph} : GlobalTheory.RealGeo.InGeo (geoHalf mu tw g) ↔ GlobalTheory.RealGeo.InGeo g

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.InclUInjective

theorem **plB_conj** (w : X.B.places.Pl) : plB X (X.B.places.conj w) = Formalism.DivisionValues.conjPl (plB X w)

theorem **ConjStable.symm_mem_iff** {Wt : Set X.B.places.Pl} (h : ConjStable X Wt) (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : (plB X).symm (Formalism.DivisionValues.conjPl v) ∈ Wt ↔ (plB X).symm v ∈ Wt

theorem **tuOf_mono** (I : ColIn X.B px) {U U' : Set I.ι} (h : U' ⊆ U) : tuOf I U' ⊆ tuOf I U

theorem **inclU_injective** {U U' : Set I.ι} (h : U' ⊆ U) : Function.Injective (inclU I h)

theorem **ixOf_inclU** {U U' : Set I.ι} (h : U' ⊆ U) (t : TupU I U') : ixOf I U (inclU I h t) = ixOf I U' t

theorem **plB_mem_twPl** (w : X.B.places.Pl) : plB X w ∈ z.R.twPl ↔ w ∈ Wt

theorem **xi_mu** {w : X.B.places.Pl} (hw : w ∈ Wt) : I.xi (ixOf I (usedOf I T Wt) (z.R.mu (plB X w))) = T.xiTw I.C w

theorem **mu_mem** (h : Wt' ⊆ Wt) {v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)} (hv : (plB X).symm v ∈ Wt') : (z.R.mu v).1 ∈ tuOf I (usedOf I T Wt')

theorem **mem_twPl** (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : v ∈ z.R.twPl ↔ (plB X).symm v ∈ Wt

theorem **mem_resTw** (h : Wt' ⊆ Wt) (v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)) : v ∈ z.resTw Wt' ↔ (plB X).symm v ∈ Wt'

theorem **inclU_resMu** (h : Wt' ⊆ Wt) {v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)} (hv : (plB X).symm v ∈ Wt') : inclU I (usedOf_mono I T h) (z.resMu h v) = z.R.mu v

theorem **resMu_off** (h : Wt' ⊆ Wt) {v : HeightOneSpectrum (𝒪 X.toD7.what₂.LL)} (hv : (plB X).symm v ∉ Wt') : z.resMu h v = mu0F (X

theorem **resMu_used** (h : Wt' ⊆ Wt) (t : TupU I (usedOf I T Wt')) : t = mu0F (X

theorem **muOf_res** (h : Wt' ⊆ Wt) : MuOf X I T Wt' (z.resMu h) (z.resTw Wt')

theorem **newTup** (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : Res.NewTup (A3

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.InfinitesimalCharacterReference

theorem **overRows_imp** {p q : RowP} (h : OverRows p) (hpq : ∀ K B f T w Z Wt hX z p0 Y, p K B f T w Z Wt hX z p0 Y → q K B f T w Z Wt hX z p0 Y) : OverRows q

theorem **overRows_and** {p q : RowP} (h : OverRows p) (h' : OverRows q) : OverRows fun K B f T w Z Wt hX z p0 Y => p K B f T w Z Wt hX z p0 Y ∧ q K B f T w Z Wt hX z p0 Y

theorem **overRows_atMem** {hL : GlobalTheory.Published.LocalLanglandsGL} {p : RowP} (h : OverRows p) : AtMem hL fun K B f T w Z Wt p0 t _ _ _ => p K B f T w Z Wt t.hX t.r.z p0 t.r.Y

theorem **overRows_allSt** {p : RowP} (h : OverRows p) : AllSt fun K B f T w P x => p K B f T w x.o.Z P.Wt P.hX x.m.z x.s.p0 x.r.Y

theorem **cenLoc_of_cenUsed** {D : D7} {M : MtFam D} (h : CenUsed D M) : CenLoc D M

theorem **sumTw_of_sumUsed** {D : D7} {M : MtFam D} (h : SumUsed D M) : SumTw D M

theorem **one_lt_norm** {D : D7} (w : PlE D.host) : (1 : ℝ) < ((Ideal.absNorm w.asIdeal : ℕ) : ℝ)

theorem **rowLoc_reset** {D : D7} (Rw : RowSide D) (σ : D.what₂.FT) : RowLoc D (reset Rw fun w => D.naut.lpUnr w (Multiset.replicate D.R.r 1)) σ

theorem **not_rowLoc_reset** {D : D7} (Rw : RowSide D) (σ : D.what₂.FT) (w : PlE D.host) (hs : IsSplitPl D.host w) (hw : ∀ k : Fin 2, w ∉ placesOver D.host.E (D.host.g k)) : ¬ RowLoc D (reset Rw fun w => D.naut.lpUnr w {((((Ideal.absNorm w.asIdeal : ℕ) : ℝ) ^ (((D.R.t : ℝ) - 1) / 2) : ℝ) : ℂ)}) σ

theorem **isoInf_string** {D : D7} (χ : D.naut.CG 1) (d : ℕ) (τ : D.host.E →+* ℂ) (s : ℂ) (h : D.naut.infG χ τ = {s}) : isoInf D.nd D.naut [(1, χ, d)] τ = (Multiset.range d).map fun c : ℕ => s + ((d : ℂ) - 1 - 2 * c) / 2

theorem **eta_of_charSide** {D : D7} {cf : CharFrame D} (C : CharSide D cf) (τ : D.what₂.Tau) : D.naut.infG D.naut.etaTinv (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ) = {-(1 / 2 : ℂ)} ∧ D.naut.infG D.naut.etaTinv (embTauBar D τ) = {(1 / 2 : ℂ)}

theorem **inf_eq_paramV** {D : D7} {cf : CharFrame D} (C : CharSide D cf) (j : Fin D.R.m) (μ : D.naut.CG 1) (τ : D.what₂.Tau) (hμ : D.naut.infG μ (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ) = {-(((D.hc τ).e j : ℚ) : ℂ)} ∧ D.naut.infG μ (embTauBar D τ) = {(((D.hc τ).e j : ℚ) : ℂ)}) : (hcSigma (D.hc τ)).map (· - 1 / 2) + isoInf D.nd D.naut [(1, D.naut.tw μ D.naut.etaTinv, D.R.t)] (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ) = NG.paramV D.R (D.hc τ) j (1 / 2) ∧ (hcSigma (D.hc τ)).map (fun y => -y + 1 / 2) + isoInf D.nd D.naut [(1, D.naut.tw μ D.naut.etaTinv, D.R.t)] (embTauBar D τ) = (NG.paramV D.R (D.hc τ) j (1 / 2)).map fun y => -y

theorem **infinitesimal_character_reference** {D : D7} {cf : CharFrame D} (C : CharSide D cf) (j : Fin D.R.m) (τ : D.what₂.Tau) : D.naut.infG (C.e1.symm (cf.mu0 j, (cf.isMu0 j).unitary)) (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ) = {-(((D.hc τ).e j : ℚ) : ℂ)} ∧ D.naut.infG (C.e1.symm (cf.mu0 j, (cf.isMu0 j).unitary)) (embTauBar D τ) = {(((D.hc τ).e j : ℚ) : ℂ)}

theorem **tieLab_of_parIn** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {cf : CharFrame D} {rep : D.what₂.FT → NG.FinTyp D.host.E D.R.r} {Rw : RowSide D} (P : ParIn hL D cf rep Rw) : SumRepresentations.tieLab D

theorem **liftAt_int** {D : D7} {Rw : RowSide D} {j : ℕ} {σ : D.what₂.FT} {π : D.nbc.AU j} {μ : D.naut.CG 1} (h : LiftAt D Rw j σ π μ) (τ : D.what₂.Tau) : ∀ z ∈ D.naut.infG μ (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ), ∃ n : ℤ, z = n

theorem **plGw_split** (D : D7) (k : Fin 2) (v : D.nd.PlG k) : IsSplitPl D.host (D.naut.plGw k v)

theorem **cancel_of_parIn** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {cf : CharFrame D} {rep : D.what₂.FT → NG.FinTyp D.host.E D.R.r} {Rw : RowSide D} (P : ParIn hL D cf rep Rw) (w : PlE D.host) (a b c : D.naut.LP w) (h : D.naut.lpSum a c = D.naut.lpSum b c) : a = b

theorem **local_lift_at_reference** {D : D7} {cf : CharFrame D} (C : CharSide D cf) (Rw : RowSide D) (j : Fin D.R.m) (σ : D.what₂.FT) (π : D.nbc.AU (j.1 + 1)) (hs : SphericalInertData.SphInert D.nd D.naut D.nbc (j.1 + 1) π) (hl : ∀ w, IsSplitPl D.host w → D.nbc.locAU (j.1 + 1) π w = D.naut.lpChar (D.naut.lpSum (Rw.locR σ w) (spehLoc D.nd D.naut ⟨1, C.e1.symm (cf.mu0 j, (cf.isMu0 j).unitary), D.R.t⟩ w)) (D.naut.locG D.naut.etaTinv w)) (hi : ∀ τ, D.nbc.infW (j.1 + 1) (D.nbc.wtAU (j.1 + 1) π) (GlobalTheory.IndefiniteRowSideFullHecke.CMTypeEmbeddings.embTau D τ) = NG.paramV D.R (D.hc τ) j (1 / 2) ∧ D.nbc.infW (j.1 + 1) (D.nbc.wtAU (j.1 + 1) π) (embTauBar D τ) = (NG.paramV D.R (D.hc τ) j (1 / 2)).map fun y => -y) : LiftAt D Rw (j.1 + 1) σ π (C.e1.symm (cf.mu0 j, (cf.isMu0 j).unitary)) where

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.InjectiveOfFlatLocal

theorem **exists_etale_local** (V : Type) [CommRing V] [IsLocalRing V] [IsNoetherianRing V] (k₀ k : Type) [Field k₀] [Field k] [Algebra k₀ k] [FiniteDimensional k₀ k] [Algebra.IsSeparable k₀ k] (φ : k₀ →+* ResidueField V) : ∃ (W : Type) (_ : CommRing W) (_ : IsLocalRing W) (_ : Algebra V W) (_ : IsLocalHom (algebraMap V W)) (j : k →+* ResidueField W), Module.Finite V W ∧ Algebra.Etale V W ∧ j.comp (algebraMap k₀ k) = (ResidueField.map (algebraMap V W)).comp φ

theorem **injective_of_flat_local** {A B : Type} [CommRing A] [CommRing B] [IsLocalRing A] [IsLocalRing B] (f : A →+* B) (hf : f.Flat) [hl : IsLocalHom f] : Function.Injective f

theorem **split_pull** {S : Scheme.{0}} {t₀ : SAB H S} {k : Type} [Field k] (q : Sp k ⟶ S) {S' : SplitExt (Sp k)} (hS : S'.IsGt ((Over.pullback q).obj t₀.G)) {R : Type} [CommRing R] [IsLocalRing R] (j : k →+* ResidueField R) {ι : Sp R ⟶ S} (hq : spMap j ≫ q = spMap (residue R) ≫ ι) {t : SAB H (Sp R)} (m : SABMap t t₀ ι) : ∃ S₁ : SplitExt (Sp (ResidueField R)), S₁.IsGt ((Over.pullback (spMap (residue R))).obj t.G)

theorem **residue_comp** {X : Scheme.{0}} (x : X) [IsNoetherianRing (X.presheaf.stalk x)] {R : Type} [CommRing R] [IsLocalRing R] [Algebra (Cpl X x) R] [IsLocalHom (algebraMap (Cpl X x) R)] : spMap ((ResidueField.map (algebraMap (Cpl X x) R)).comp (ResidueField.map (algebraMap (X.presheaf.stalk x) (Cpl X x)))) ≫ X.fromSpecResidueField x = spMap (residue R) ≫ spMap (algebraMap (Cpl X x) R) ≫ fromSpecCompletedStalk X x

theorem **isNoetherianRing_stalk** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) (x : Z.X) : IsNoetherianRing (Z.X.presheaf.stalk x)

theorem **charZero_stalk** (Z : Mdl Y i₀) (x : Z.X) : CharZero (Z.X.presheaf.stalk x)

theorem **generic_image** {X : Scheme.{0}} (x : X) {R : Type} [CommRing R] [IsDomain R] (φ : X.presheaf.stalk x →+* R) (hφ : Function.Injective φ) (ξ : Sp R) (hξ : ξ.asIdeal = ⊥) (y : X) (hy : y ~ (spMap φ ≫ X.fromSpecStalk x) ξ) : y = (spMap φ ≫ X.fromSpecStalk x) ξ

theorem **exists_bdat** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) (hU : (Y Z.i).N3) (x : Z.X) [IsDomain (Cpl Z.X x)] [IsIntegrallyClosed (Cpl Z.X x)] (R : Type) [CommRing R] [IsLocalRing R] [Algebra (Cpl Z.X x) R] [hl : IsLocalHom (algebraMap (Cpl Z.X x) R)] [hf : Module.Finite (Cpl Z.X x) R] [he : Algebra.Etale (Cpl Z.X x) R] (hs : ∀ (t : SAB H (Sp R)), SABMap t Z.t (spMap (algebraMap (Cpl Z.X x) R) ≫ fromSpecCompletedStalk Z.X x) → SABMap t (Y Z.i).t ((spMap (algebraMap (Cpl Z.X x) R) ≫ fromSpecCompletedStalk Z.X x) ≫ Z.π) → ∃ S₁ : SplitExt (Sp (ResidueField R)), S₁.IsGt ((Over.pullback (spMap (residue R))).obj t.G)) : ∃ A : BDat Z.t x, Function.Bijective (algebraMap (Cpl Z.X x) R) → A.Exact

theorem **nonempty_bdat** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) (hS : (Y Z.i).SplitFibres) (hU : (Y Z.i).N3) (x : Z.X) [IsDomain (Cpl Z.X x)] [IsIntegrallyClosed (Cpl Z.X x)] : Nonempty (BDat Z.t x)

theorem **exists_exact** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) (hU : (Y Z.i).N3) (x : Z.X) [IsDomain (Cpl Z.X x)] [IsIntegrallyClosed (Cpl Z.X x)] (hs : SplitAt Z.t x) : ∃ A : BDat Z.t x, A.Exact

theorem **pick_exact** {X : Scheme.{0}} {t₀ : SAB H X} {x : X} (h : Nonempty (BDat t₀ x)) (h' : ∃ A : BDat t₀ x, A.Exact) : (pick h).Exact

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IntertwinersExistScalarUpdateInvariant

theorem **baseWeight_scalar_update_invariant** {A : FamIn D.what₂.LL G Pl0 T D.R.m} {R : RealDat D A Ks} {F : Face D A} (h : R.BaseW F) (k l) : (R.upd6 k l).BaseW F

theorem **intertwinersExist_scalar_update_invariant** {A : FamIn D.what₂.LL G Pl0 T D.R.m} {R : RealDat D A Ks} (e : R.ExAt) (k l) : (R.upd6 k l).ExAt

theorem **nz_updS** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (k l) (F0 : HeightOneSpectrum (𝒪 D.what₂.LL) → Fin D.R.m → ℝ) (w : HeightOneSpectrum (𝒪 D.what₂.LL)) : (∃ j, NzAt (R.updS k l) F F0 j w ∨ NzAt (R.updS k l) F F0 j (Formalism.DivisionValues.conjPl w)) ↔ ∃ j, NzAt R F F0 j w ∨ NzAt R F F0 j (Formalism.DivisionValues.conjPl w)

theorem **kapOf_eq** (S : RealDat3 D A3 Ks) (κ : ℝ) (hκ : Lift.IsKap D.R.m (fun t => S.Th t ∘ₗ LinearMap.adjoint (S.Th t)) S.EEp3 S.lamN κ) (hn : ∃ t p, S.EEp3 p ≠ ⊥ ∧ ∏ j ∈ Finset.Icc 1 D.R.m, S.lamN t j p ≠ 0) : kapOf S = κ

theorem **kapOf_of_thmB** (S : RealDat3 D A3 Ks) (hB : S.PThmB) : kapOf S = S.kap

theorem **isScal_kapOf** (S : RealDat3 D A3 Ks) (F : Face3 D A3) (h0 : S.eigenvalue_at_base_of_scalar) (h1 : S.scal_off F) (h2 : S.zero_part_eigenvalue_convention F) (h3 : S.scal_abs F) (h4 : S.scal_tup) : (S.updS (kapOf S) S.lam3).IsScal F

theorem **pins_updS** (hf : Function.Injective f) {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (h : Pins R (bare3 _ P0 ψ0 P1 ψ1) cG) (e : R.toRealDat.ExAt) (n : NewTup R mu tw) (k l) (hs : ((dat3 f μ R mu tw).updS k l).IsScal (bare3 _ P0 ψ0 P1 ψ1)) : Pins ((dat3 f μ R mu tw).updS k l) (bare3 _ P0 ψ0 P1 ψ1) cG

theorem **pins_kapOf** (hf : Function.Injective f) {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (h : Pins R (bare3 _ P0 ψ0 P1 ψ1) cG) (e : R.toRealDat.ExAt) (n : NewTup R mu tw) : Pins ((dat3 f μ R mu tw).updS (kapOf (dat3 f μ R mu tw)) (dat3 f μ R mu tw).lam3) (bare3 _ P0 ψ0 P1 ψ1) cG

theorem **pins_res** (z : Dat X I hX T Wt) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') (e : z.R.toRealDat.ExAt) (k l) (hs : ((z.resR h).updS k l).IsScal (face0 X I hX (usedOf I T Wt') z.Xg z.tr)) : Pins ((z.resR h).updS k l) (face0 X I hX (usedOf I T Wt') z.Xg z.tr) z.cG

theorem **exAt_res** (z : Dat X I hX T Wt) (h : Wt' ⊆ Wt) (e : z.R.toRealDat.ExAt) : (z.resR h).toRealDat.ExAt

theorem **exists_restrict** (z : Dat X I hX T Wt) (e : z.R.toRealDat.ExAt) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') : ∃ z' : Dat X I hX T Wt', z'.Xg = z.Xg ∧ z'.tr = z.tr ∧ z'.core = z.core ∧ z'.R.toRealDat.ExAt ∧ (GlobalTheory.RealGeo.InGeo z.geo → GlobalTheory.RealGeo.InGeo z'.geo) ∧ ∀ F0, z'.W1 F0 = z.W1 F0

theorem **exists_exact** (z : Dat X I hX T Wt) (hL : Rw.HoldsL z.R z.F) (F0) (h : z.W1 F0 ⊆ Wt) : ∃ z' : Dat X I hX T (z.W1 F0), z'.core = z.core ∧ z'.R.toRealDat.ExAt ∧ (GlobalTheory.RealGeo.InGeo z.geo → GlobalTheory.RealGeo.InGeo z'.geo) ∧ z.W1 F0 = z'.W1 F0

theorem **pins_resK** (z : Dat X I hX T Wt) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') (e : z.R.toRealDat.ExAt) : Pins ((z.resR h).updS (kapOf (z.resR h)) (z.resR h).lam3) (face0 X I hX (usedOf I T Wt') z.Xg z.tr) z.cG

theorem **restrictEx_core** : (restrictEx z e Wt' h hc).core = z.core

theorem **restrictEx_exAt** : (restrictEx z e Wt' h hc).R.toRealDat.ExAt

theorem **restrictEx_inGeo** (hi : GlobalTheory.RealGeo.InGeo z.geo) : GlobalTheory.RealGeo.InGeo (restrictEx z e Wt' h hc).geo

theorem **restrictExists_nonzero_places** (F0) : (restrictEx z e Wt' h hc).W1 F0 = z.W1 F0

theorem **restrictEx_R_of_holds** (hH : Holds z.R z.F) : (restrictEx z e Wt' h hc).R = z.resR h

theorem **restrictEx_holds** (hH : Holds z.R z.F) : Holds (restrictEx z e Wt' h hc).R (restrictEx z e Wt' h hc).F

theorem **resMu_self** : z.resMu subset_rfl = z.R.mu

theorem **resTw_self** : z.resTw Wt = z.R.twPl

theorem **tup_empty** (z : Dat X I hX T Wt) (t : TupU I (usedOf I T (∅ : Set X.B.places.Pl))) : t = mu0F (X

theorem **inclU_not_surjective** (z : Dat X I hX T Wt) {w} (hw : w ∈ Wt) : ¬ Function.Surjective (inclU I (usedOf_mono I T (Set.empty_subset Wt)))

theorem **conj_needed** (h : ¬ ConjStable X Wt') : IsEmpty (Dat X I hX T Wt')

theorem **finite_of_dat** (z : Dat X I hX T Wt) : Wt.Finite

theorem **restrictE_twice** (e : z.R.toRealDat.ExAt) (h : Wt' ⊆ Wt) (hc : ConjStable X Wt') (h' : Wt'' ⊆ Wt') : (z.restrictE e Wt' h hc).resTw Wt'' = z.resTw Wt'' ∧ (z.restrictE e Wt' h hc).resMu h' = z.resMu (h'.trans h)

theorem **pair** (e : z.R.toRealDat.ExAt) {w} (hw : w ∈ Wt) : ∃ y : Dat X I hX T {w, X.B.places.conj w}, plB X w ∈ y.R.twPl ∧ y.R.twPl.card ≤ 2 ∧ y.core = z.core

theorem **subset_nonzero_places_indicator** (j : Fin X.toD7.R.m) : Wt ⊆ z.W1 fun v _ => if v ∈ z.R.twPl then 1 else 0

theorem **restrictNonzero_idempotent** (F0) (e : z.R.toRealDat.ExAt) (h : z.W1 F0 ⊆ Wt) : ((z.restrictNonzeroOfExists F0 e h).restrictNonzeroOfExists F0 (z.restrictNonzeroOfExists_exists_lifts F0 e h) (z.restrictNonzeroOfExists_nonzero_places F0 e h).le).W1 F0 = z.W1 F0

theorem **realData_restrict_id** {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 I0 T0 : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T0 D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] {μ : T0} (R : RealDat3 D (A3.comap id (id μ)) Ks) : Res.dat3 id μ R R.mu R.twPl = R

theorem **resR_self** : z.resR subset_rfl = z.R

theorem **updS_kapNew** {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 I0 T0 : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T0 D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] {S R : RealDat3 D A3 Ks} (e : S = R) (h : R.scal_kap) : S.updS (Res.kapNew S) S.lam3 = R

theorem **resRE_self** : z.resRE subset_rfl = z.R

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.InvariantDataMet

theorem **smoothOn_det** {n : ℕ} {M : Pt ι p q → Matrix (Fin n) (Fin n) ℂ} (h : ∀ a b, SmoothOn fun z => M z a b) : SmoothOn fun z => (M z).det

theorem **smoothOn_inv** {n : ℕ} {M : Pt ι p q → Matrix (Fin n) (Fin n) ℂ} (h : ∀ a b, SmoothOn fun z => M z a b) (hd : ∀ z ∈ DomI ι p q, (M z).det ≠ 0) (a b : Fin n) : SmoothOn fun z => (M z)⁻¹ a b

theorem **smoothOn_coord** (τ : ι) (a : Fin p) (b : Fin q) : SmoothOn fun z : Pt ι p q => z τ a b

theorem **smoothOn_star** {f : Pt ι p q → ℂ} (h : SmoothOn f) : SmoothOn fun z => star (f z)

theorem **smoothOn_hP** (τ : ι) (a b : Fin p) : SmoothOn fun z : Pt ι p q => hP (z τ) a b

theorem **smoothOn_hQ** (τ : ι) (a b : Fin q) : SmoothOn fun z : Pt ι p q => hQ (z τ) a b

theorem **smoothOn_hP_inv** (τ : ι) (a b : Fin p) : SmoothOn fun z : Pt ι p q => (hP (z τ))⁻¹ a b

theorem **smoothOn_hQ_inv** (τ : ι) (a b : Fin q) : SmoothOn fun z : Pt ι p q => (hQ (z τ))⁻¹ a b

theorem **bergman_apply** (z ξ η : Pt ι p q) : bergman z ξ η = ∑ τ, ((hP (z τ))⁻¹ * ξ τ * (hQ (z τ))⁻¹ * (η τ)ᴴ).trace

theorem **trace_pos** {A : Matrix (Fin p) (Fin p) ℂ} {B : Matrix (Fin q) (Fin q) ℂ} (hA : A.PosDef) (hB : B.PosDef) (ξ : Matrix (Fin p) (Fin q) ℂ) : 0 ≤ (A * ξ * B * ξᴴ).trace ∧ (ξ ≠ 0 → 0 < (A * ξ * B * ξᴴ).trace)

theorem **bergman_moeb** {g : ι → GL (Fin p ⊕ Fin q) ℂ} (hg : g ∈ UpqI ι p q) {z : Pt ι p q} (hz : z ∈ DomI ι p q) (ξ η : Pt ι p q) : bergman (moebI g z) (fun τ => tanAct (g τ) (z τ) (ξ τ)) (fun τ => tanAct (g τ) (z τ) (η τ)) = bergman z ξ η

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsAbsIrrCharCompIff

theorem **isAbsIrrChar_comp** {K G H : Type} [Field K] [Group G] [Group H] (e : G ≃* H) {χ : H → K} (h : IsAbsIrrChar χ) : IsAbsIrrChar fun g => χ (e g)

theorem **isAbsIrrChar_comp_iff** {K G H : Type} [Field K] [Group G] [Group H] (e : G ≃* H) {χ : H → K} : (IsAbsIrrChar fun g => χ (e g)) ↔ IsAbsIrrChar χ

theorem **famC_A** (hX : B.fields.Ok) (U : Set I.ι) (Xg : Finset (Gc B px → B.LL)) (tr tr' : TrI B px) : (famC I hX U Xg tr).A = (famC I hX U Xg tr').A

theorem **mem_tuOf** {U : Set I.ι} {τ : Tup B.fields.F B.fields.E B.fields.K I.S} : τ ∈ tuOf I U ↔ ∃ a ∈ U, I.tup a = τ

theorem **tup_mem_tuOf** {U : Set I.ι} {a : I.ι} : I.tup a ∈ tuOf I U ↔ a ∈ U

theorem **ixOf_eq** {U : Set I.ι} {t : TupU I U} {a : I.ι} (h : I.tup a = t.1) : ixOf I U t = a

theorem **tup_ixOf** {U : Set I.ι} (t : TupU I U) : I.tup (ixOf I U t) = t.1

theorem **ixOf_mu0F** (U : Set I.ι) : ixOf I U (mu0F (X

theorem **ixOf_injective** (U : Set I.ι) : Function.Injective (ixOf I U)

theorem **mem_range_ixOf** {U : Set I.ι} {a : I.ι} : a ∈ Set.range (ixOf I U) ↔ a = I.mu0 ∨ a ∈ U

theorem **usedOf_mono** (T : Top.TwDat B) {Wt Wt' : Set B.places.Pl} (h : Wt ⊆ Wt') : usedOf I T Wt ⊆ usedOf I T Wt'

theorem **family_infinitePlace** : (famC I hX U Xg tr).A.v0 = InfinitePlace.mk B.embLL

theorem **famC_μ0** : (famC I hX U Xg tr).A.μ0 = mu0F (X

theorem **famC_X** : (famC I hX U Xg tr).A.X = Xg

theorem **famC_r** : (famC I hX U Xg tr).A.r = B.R.r

theorem **famC_A_t** : (famC I hX U Xg tr).A.t = B.R.t

theorem **famC_trI** : (famC I hX U Xg tr).trI = tr

theorem **famC_muVal** (j : Fin B.R.m) (t : TupU I U) (P : PlS B px) : (famC I hX U Xg tr).A.muVal j t P = I.Ch.muVal j P.1 * xiValOf I t.1 P.1

theorem **embLL_xiValOf** (a : I.ι) (P : HeightOneSpectrum (𝒪 B.fields.E)) (h : ∃ z : B.LL, B.embLL z = (((I.xi a).val P : ℂˣ) : ℂ)) : B.embLL (xiValOf I (I.tup a) P : B.LL) = (((I.xi a).val P : ℂˣ) : ℂ)

theorem **psiC_of_mem** {PP : Pl B.host.E} (h : PP ∈ (B.primes px).vGs) : psiC I PP = I.Ch.psi0 (PP, h)

theorem **psiC_of_notMem** {PP : Pl B.host.E} (h : PP ∉ (B.primes px).vGs) : psiC I PP = 1

theorem **ValsIn.embLL_xiValOf** {I : ColIn B px} (h : ValsIn I) (a : I.ι) (P : HeightOneSpectrum (𝒪 B.fields.E)) (hP : (I.xi a).UnrAt P) : B.embLL (xiValOf I (I.tup a) P : B.LL) = (((I.xi a).val P : ℂˣ) : ℂ)

theorem **goodSet_iff** (eR : Gc X.B px ≃* G0g X.toD7.nd) (Xg : Finset (Gc X.B px → X.B.LL)) : GoodSet X (px

theorem **GoodSet.unique** {eR : Gc X.B px ≃* G0g X.toD7.nd} {Xg Xg' : Finset (Gc X.B px → X.B.LL)} (h : GoodSet X (px

theorem **GoodSet.irr** {eR : Gc X.B px ≃* G0g X.toD7.nd} {Xg : Finset (Gc X.B px → X.B.LL)} (h : GoodSet X (px

theorem **isIdG_refl** : IsPlacewiseIdentity X (px

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsBaseEnlargementFixedCoefFieldOfCoeffEnlargemen

theorem **isBaseEnlargementFixedCoefField_of_coeffEnlargement** {B B' : Base} (h : Enl B B') : Enl0 B B'

theorem **enlarges_refl** (B : Base) : Enl0 B B

theorem **datExt_of_datExt0** (h : DatExt0) : DatExt

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsBlockMetricDiagForm

theorem **form_sum** (h : M →ₗ[ℂ] M →ₗ⋆[ℂ] ℂ) (u : n → M) (v w : n → ℂ) : h (∑ a, v a • u a) (∑ a, w a • u a) = star w ⬝ᵥ gramOf h u *ᵥ v

theorem **diagForm_apply** (b : Module.Basis n ℂ M) (a : n → ℝ) (v w : M) : diagForm b a v w = ∑ i, a i * (b.repr v i * starRingEnd ℂ (b.repr w i))

theorem **diagForm_symm** (b : Module.Basis n ℂ M) (a : n → ℝ) (v w : M) : diagForm b a w v = starRingEnd ℂ (diagForm b a v w)

theorem **diagForm_pos** (b : Module.Basis n ℂ M) {a : n → ℝ} (ha : ∀ i, 0 < a i) {v : M} (hv : v ≠ 0) : 0 < (diagForm b a v v).re

theorem **dualGram_posDef** {A : Matrix n n ℂ} (hA : A.PosDef) : (dualGram A).PosDef

theorem **dualGram_conj** (A T : Matrix n n ℂ) : dualGram (Tᴴ * A * T) = (T⁻¹ᵀ)ᴴ * dualGram A * T⁻¹ᵀ

theorem **dualGram_smul_one** {a : ℂ} (ha : a ≠ 0) : dualGram (a • (1 : Matrix n n ℂ)) = a⁻¹ • 1

@[simp] theorem **left_tanOf** {v : Sp (DualNumber S) ⟶ t.G.left} (hv : IsTanAtG t x v) : (tv (tanOf hv)).left = v

theorem **isTan_tv** (v : TanM x t.G) : IsTanAtG t x (tv v).left

theorem **tan_ext** {v w : TanM x t.G} (h : (tv v).left = (tv w).left) : v = w

theorem **isComb_iff** {m : ℕ} (u : Fin m → TanM x t.G) (a : Fin m → S) (w : TanM x t.G) : IsCombAtG t x (fun i => (tv (u i)).left) a (tv w).left ↔ w = ∑ i, a i • u i

theorem **isTan_push** {v : Sp (DualNumber S) ⟶ t'.G.left} (hv : IsTanAtG t' x' v) : IsTanAtG t x (v ≫ m.φ)

theorem **left_pushL** (v : TanM x' t'.G) : (tv (pushL m x' hx v)).left = (tv v).left ≫ m.φ

theorem **pushL_bijective** : Function.Bijective (pushL m x' hx)

theorem **left_pushE** (v : TanM x' t'.G) : (tv (pushE m x' hx v)).left = (tv v).left ≫ m.φ

theorem **pushE_act** (a : 𝒪 E) (v : TanM x' t'.G) : pushE m x' hx (actR t'.act a v) = actR t.act a (pushE m x' hx v)

theorem **isTan_map** (φ : S →+* S') {x' : Sp S' ⟶ X} (hx : spMap φ ≫ x = x') {v : Sp (DualNumber S) ⟶ t.G.left} (hv : IsTanAtG t x v) : IsTanAtG t x' (spMap (dualMap φ) ≫ v)

theorem **IsBlockMetric.gram_posDef** {y : Sp ℂ ⟶ X} {h : TanM y t.G →ₗ[ℂ] TanM y t.G →ₗ⋆[ℂ] ℂ} (hh : IsBlockMetric h) {n : Type} [Fintype n] {u : n → TanM y t.G} (hu : LinearIndependent ℂ u) : (gramOf h u).PosDef

theorem **exists_isBlockMetric** (y : Sp ℂ ⟶ X) : ∃ h : TanM y t.G →ₗ[ℂ] TanM y t.G →ₗ⋆[ℂ] ℂ, IsBlockMetric h

theorem **spMap_evHom** {V : Z.affineOpens} {y : CxPts p σ} (hy : y ∈ ptsIn p σ V.1) : spMap (evHom hy) ≫ LogKS.chart V = y.1

theorem **IsLieMetric.of_anRead** {f : Z ⟶ X} {R : AnRead p σ D} {h} (hh : IsLieMetric t f R h) (R' : AnRead p σ D) : IsLieMetric t f R' h

theorem **smoothOn_dim_zero** (hD : D = 0) (R : AnRead p σ D) (T : Set (CxPts p σ)) (g : CxPts p σ → ℝ) : R.SmoothOn T g

theorem **isLieMetric_dim_zero** (hD : D = 0) (f : Z ⟶ X) (R : AnRead p σ D) {h : ∀ y : CxPts p σ, TanM (y.1 ≫ f) t.G →ₗ[ℂ] TanM (y.1 ≫ f) t.G →ₗ⋆[ℂ] ℂ} (hh : ∀ y, IsBlockMetric (h y)) : IsLieMetric t f R h

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsClosedFlagsIff

theorem **flags_surjective** : Function.Surjective (Frames.flags (k

theorem **isClosed_zero** {f : Frames k V → k} (hf : f ∈ regFns k V) : IsClosed {b | f b = 0}

theorem **isClosed_flags_iff** {s : Set (Flags k V)} : IsClosed s ↔ IsClosed (Frames.flags ⁻¹' s)

theorem **mem_step_flags** {b : Frames k V} {i : ι} {n : ℕ} {v : V i} : v ∈ b.flags.step i n ↔ ∀ a : Fin (finrank k (V i)), n ≤ (a : ℕ) → (b i).repr v a = 0

theorem **step_flags_le** {b : Frames k V} {i : ι} {n : ℕ} {P : Submodule k (V i)} : b.flags.step i n ≤ P ↔ ∀ a : Fin (finrank k (V i)), (a : ℕ) < n → b i a ∈ P

theorem **finrank_step** (F : Flags k V) (i : ι) (n : ℕ) : finrank k (F.step i n) = min n (finrank k (V i))

theorem **Flags.ext** {F F' : Flags k V} (h : ∀ i n, F.step i n = F'.step i n) : F = F'

theorem **dim_eq** {s : Closeds (Flags k V)} {n : ℕ} (h : topologicalKrullDim (s : Set (Flags k V)) = n) : dim s = n

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsClosedImmersionBaseChange

theorem **dim_sch** : topologicalKrullDim c.sch = 1

theorem **natCast_eq_zero** : (p : Γ(c.sch, ⊤)) = 0

theorem **deg_eq_of_imm** {C : Scheme.{0}} [Algebra (ZMod p) Γ(C, ⊤)] (i : C ⟶ X) [IsClosedImmersion i] [IsReduced C] (h : Set.range i = c.1) (hL : L.IsLB) : c.deg L = interNum (ZMod p) (structureModule C) ![L.pull i]

theorem **isProper_str** {O : Type} [CommRing O] [Finite (O ⧸ Ideal.span {(p : O)})] (s : X ⟶ Sp O) [IsProper s] : IsProper c.str

theorem **isCompleteCurve_top** : IsCompleteCurve (ZMod p) (⊤ : Closeds c.sch)

theorem **deg_of_cls_eq_add** (hH : IsAmpleLB H) (hN : N.IsLB) (hL : L.IsLB) (hM : M.IsLB) (e : cls N hN = cls L hL + cls M hM) : c.deg N = c.deg L + c.deg M

theorem **deg_pos** (hH : IsAmpleLB H) : 0 < c.deg H

theorem **cov_pos** (z : admitted Y i₀) : 0 < cov z

theorem **natCast_mem_nonunits** {l : Level H} (c : Coeff l) (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) : (p : c.F) ∈ (c.V W).nonunits

theorem **natCast_eq_zero_iff** (hp : p.Prime) (hW : (p : ℂ) ∈ W.nonunits) (Z : LocM.Mdl Y i₀) (x : Z.X) : (p : Z.X.residueField x) = 0 ↔ ¬ Z.inQ x

theorem **exists_isAmpleLB** (Z : LocM.Mdl Y i₀) : ∃ A : LB Z.X, IsAmpleLB A

theorem **degree_twistingSheaf_line** : (lineCrv p φ).deg (O1 O 1) = 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsDiscriminantTwelfthRootOfIdealLatticeIffDelta1

theorem **lamOf_rec** (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) : lamOf D.host cc δ = TaylorLogMetric.lamOf D cc δ

theorem **isDiscriminantTwelfthRoot_host_iff** (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) : IsDiscriminantTwelfthRootOfIdealLattice D.host cc δ ↔ TaylorLogMetric.IsDiscriminantTwelfthRootOfIdealLattice D cc δ

theorem **stdL_rec** (N : ℕ) : stdL D.host N = StandardLatticesGL3.stdL D N

theorem **detId_rec** {N : ℕ} (L L' : Lt D.host N) : detId D.host L L' = TaylorLogMetric.detId D L L'

theorem **La_rec** (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {N N' : ℕ} (L : Lt D.host N) (L' : Lt D.host N') : La D.host (TaylorLogMetric.embT D) (lamOf D.host cc δ) L L' = TaylorLogMetric.La D (TaylorLogMetric.tor0 D cc δ) L L'

theorem **isDgen_rec** (cD : D.what₂.Tau → ℂ) : IsDgen D.host (TaylorLogMetric.embT D) cD ↔ Formalism.L2.Ker.IsDgen D cD

theorem **chiFac_rec** (χ : TaylorLogMetric.IdE D → ℂ) (N N' : ℕ) (L : Lt D.host N) (L' : Lt D.host N') : chiFac D.host χ N N' L L' = Formalism.L2.Ker.Base.chiFac χ N N' L L'

theorem **VacIn.toRec_ofRec** (B : Formalism.L2.Ker.Base D cc δ) : (VacIn.ofRec B).toRec = B

theorem **VacIn.ofRec_toRec** (V : VacIn D.host (TaylorLogMetric.embT D) cc δ) : VacIn.ofRec V.toRec = V

theorem **VacIn.thMM_rec** (V : VacIn D.host (TaylorLogMetric.embT D) cc δ) : V.thMM = V.toRec.thMM

theorem **VacIn.eNat_rec** (V : VacIn D.host (TaylorLogMetric.embT D) cc δ) (χ : TaylorLogMetric.IdE D → ℂ) : V.eNat χ = V.toRec.eNat χ

theorem **perAA_top** : perAA H ⊤ = (H.AA0 : Submodule (𝒪 H.E) H.E)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsEndPullEndPull

theorem **sv_twistFun** (v : SecM A L b T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : sv (twistFun hL hm α hα v) r z J = (val α r z J).val * sv v (r ≫ ψ.left) z J

theorem **sv_twist** (v : SecM A L b T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : sv (twist hL hm hψ α hα v) r z J = (val α r z J).val * sv v (r ≫ ψ.left) z J

theorem **gen_ext** {X Y : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {u u' : LB.Gen (LB.LinComb M φ ε)} (hu : u.IsGen) (hu' : u'.IsGen) (h : ∀ y J, Good M y J → val u (𝟙 _) y J = val u' (𝟙 _) y J) : u = u'

theorem **ext** {L : LB A.left} (hL : L.IsLB) {θ θ' : ThetaGrp A L k T} (h : θ.pt = θ'.pt) (hv : ∀ y J, Good L y J → tval θ (𝟙 _) y J = tval θ' (𝟙 _) y J) : θ = θ'

theorem **exists_scalar** {L : LB A.left} (hL : L.IsLB) (k : ℤ) (c : Γ(T.left, ⊤)ˣ) : ∃ θ : ThetaGrp A L k T, θ.pt = 1 ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → tval θ r z J = gm (r ≫ (fst T A).left) z ⊤ c

theorem **pt_one** : (1 : ThetaGrp A L.L k T).pt = 1

theorem **pt_mul'** (θ θ' : ThetaGrp A L.L k T) : (θ * θ').pt = θ'.pt * θ.pt

theorem **pt_inv** (θ : ThetaGrp A L.L k T) : θ⁻¹.pt = θ.pt⁻¹

theorem **tval_one** : tval (1 : ThetaGrp A L.L k T) r z J = 1

theorem **tval_mul** (θ θ' : ThetaGrp A L.L k T) : tval (θ * θ') r z J = tval θ r z J * tval θ' (r ≫ (translT A θ.pt).left) z J

theorem **tval_inv** (θ : ThetaGrp A L.L k T) : tval θ⁻¹ r z J = (tval θ (r ≫ (translT A θ.pt⁻¹).left) z J)⁻¹

theorem **sv_op** {L : RigLB A} {k : ℤ} {T : Over S} (θ : ThetaGrp A L.L k T) (v : SecM A L.L k T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : sv (op θ v) r z J = (tval θ r z J).val * sv v (r ≫ (translT A θ.pt).left) z J

theorem **pt_scalar** (c : Γ(T.left, ⊤)ˣ) : (scalar k T L c).pt = 1

theorem **tval_scalar** (c : Γ(T.left, ⊤)ˣ) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : tval (scalar k T L c) r z J = gm (r ≫ (fst T A).left) z ⊤ c

theorem **scalar_comm** (c : Γ(T.left, ⊤)ˣ) (θ : ThetaGrp A L.L k T) : scalar k T L c * θ = θ * scalar k T L c

theorem **scalar_injective** : Function.Injective (scalar k T L)

theorem **action_scalar** (c : Γ(T.left, ⊤)ˣ) (v : SecM A L.L k T) : action k T L (scalar k T L c) v = c.val • v

theorem **exists_scalar_eq** (hp : Function.Surjective (fst T A).left.appTop.hom) {θ : ThetaGrp A L.L k T} (h : θ.pt = 1) : ∃ c, scalar k T L c = θ

theorem **pt_mul** (θ θ' : ThetaGrp A L.L k T) : (θ * θ').pt = θ.pt * θ'.pt

theorem **negT_snd** : (negT A T).left ≫ (snd T A).left = (snd T A ≫ (GrpObj.inv : A ⟶ A)).left

theorem **negT_fst** : (negT A T).left ≫ (fst T A).left = (fst T A).left

theorem **negT_negT** : (negT A T).left ≫ (negT A T).left = 𝟙 _

theorem **translT_negT** (x : T ⟶ A) : (translT A x).left ≫ (negT A T).left = (negT A T).left ≫ (translT A x⁻¹).left

theorem **exists_conj** (ν : LB.Gen (negLB A L.L k T)) (hν : ν.IsGen) (θ : ThetaGrp A L.L k T) : ∃ θ' : ThetaGrp A L.L k T, θ'.pt = θ.pt⁻¹ ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L.L z J → tval θ' r z J = nval ν r z J * (tval θ (r ≫ (negT A T).left) z J * (nval ν (r ≫ (translT A θ.pt⁻¹).left) z J)⁻¹)

theorem **pt_conj** (θ : ThetaGrp A L.L k T) : (conj ν hν θ).pt = θ.pt⁻¹

theorem **tval_conj** (θ : ThetaGrp A L.L k T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : tval (conj ν hν θ) r z J = nval ν r z J * (tval θ (r ≫ (negT A T).left) z J * (nval ν (r ≫ (translT A θ.pt⁻¹).left) z J)⁻¹)

theorem **conj_scalar** (c : Γ(T.left, ⊤)ˣ) : conj ν hν (scalar k T L c) = scalar k T L c

theorem **conj_conj** (θ : ThetaGrp A L.L k T) : conj ν hν (conj ν hν θ) = θ

theorem **pt_delta** (θ : ThetaGrp A L.L k T) : (delta hs hp θ).pt = θ.pt⁻¹

theorem **delta_scalar** (c : Γ(T.left, ⊤)ˣ) : delta hs hp (scalar k T L c) = scalar k T L c

theorem **delta_delta** (θ : ThetaGrp A L.L k T) : delta hs hp (delta hs hp θ) = θ

theorem **exists_lift** (c : CubicalStr A L) {V : Type} [CommRing V] [IsLocalRing V] (t : Sp V ⟶ S) {n : ℕ} {x : Over.mk t ⟶ A} (hx : x ^ n = 1) : ∃ θ : ThetaGrp A L.L n (Over.mk t), θ.pt = x

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsEqnIffGeneratesAtInv

theorem **gen_mem** {U : X.Opens} {x : X} (hx : x ∈ U) : genericPoint X ∈ U

theorem **memInf** {U V : X.Opens} {x : X} (h : x ∈ U) (h' : x ∈ V) : x ∈ U ⊓ V

theorem **fnOf_injective** (U : X.Opens) (h : genericPoint X ∈ U) : Function.Injective (fnOf U h)

@[simp] theorem **fnOf_resG** {U V : X.Opens} (e : V ≤ U) (h : genericPoint X ∈ V) (a : Γ(X, U)) : fnOf V h (resG X e a) = fnOf U (e h) a

theorem **algebraMap_germ** {U : X.Opens} {x : X} (hx : x ∈ U) (a : Γ(X, U)) : algebraMap (X.presheaf.stalk x) X.functionField (X.presheaf.germ U x hx a) = fnOf U (gen_mem hx) a

theorem **fn_ext** {U : X.Opens} {a b : Γ(X, U)} (h : ∀ hU : genericPoint X ∈ U, fnOf U hU a = fnOf U hU b) : a = b

theorem **UnitAt.regAt** (h : UnitAt x f) : RegAt x f

theorem **ZeroAt.regAt** (h : ZeroAt x f) : RegAt x f

private theorem **algebraMap_stalk_functionField_injective** (x : X) : Function.Injective (algebraMap (X.presheaf.stalk x) X.functionField)

theorem **zeroAt_iff** : ZeroAt x f ↔ RegAt x f ∧ ¬ UnitAt x f

theorem **regAt_zero** (x : X) : RegAt x 0

theorem **unitAt_one** (x : X) : UnitAt x 1

theorem **zeroAt_zero** (x : X) : ZeroAt x 0

theorem **RegAt.add** (hf : RegAt x f) (hg : RegAt x g) : RegAt x (f + g)

theorem **RegAt.mul** (hf : RegAt x f) (hg : RegAt x g) : RegAt x (f * g)

theorem **UnitAt.mul** (hf : UnitAt x f) (hg : UnitAt x g) : UnitAt x (f * g)

theorem **UnitAt.inv** (hf : UnitAt x f) : UnitAt x f⁻¹

theorem **UnitAt.ne_zero** (hf : UnitAt x f) : f ≠ 0

theorem **ZeroAt.mul** (hf : ZeroAt x f) (hg : RegAt x g) : ZeroAt x (f * g)

theorem **ZeroAt.add** (hf : ZeroAt x f) (hg : ZeroAt x g) : ZeroAt x (f + g)

theorem **unitAt_iff** (hf : f ≠ 0) : UnitAt x f ↔ RegAt x f ∧ RegAt x f⁻¹

theorem **regAt_fnOf** {U : X.Opens} (hx : x ∈ U) (h : genericPoint X ∈ U) (a : Γ(X, U)) : RegAt x (fnOf U h a)

theorem **unitAt_fnOf_iff** {U : X.Opens} (hx : x ∈ U) (h : genericPoint X ∈ U) (a : Γ(X, U)) : UnitAt x (fnOf U h a) ↔ x ∈ X.basicOpen a

theorem **unitAt_fnOf** {U : X.Opens} (hx : x ∈ U) (h : genericPoint X ∈ U) (a : Γ(X, U)ˣ) : UnitAt x (fnOf U h a)

theorem **RegAt.exists_fnOf** (h : RegAt x f) : ∃ (U : X.Opens) (hx : x ∈ U) (a : Γ(X, U)), fnOf U (gen_mem hx) a = f

theorem **UnitAt.exists_nhds** (h : UnitAt x f) : ∃ U : X.Opens, x ∈ U ∧ ∀ y ∈ U, UnitAt y f

private theorem **fnOf_map'** {U V : X.Opens} (e : V ⟶ U) (h : genericPoint X ∈ V) (a : Γ(X, U)) : fnOf V h (X.presheaf.map e.op a) = fnOf U (e.le h) a

theorem **exists_fnOf_eq** {U : X.Opens} (hU : genericPoint X ∈ U) (h : ∀ x ∈ U, RegAt x f) : ∃ a : Γ(X, U), fnOf U hU a = f

theorem **exists_unit_fnOf_eq** {U : X.Opens} (h : ∀ x ∈ U, UnitAt x f) : ∃ a : Γ(X, U)ˣ, ∀ hU, fnOf U hU a = f

theorem **regAt_genericPoint** (f : X.functionField) : RegAt (genericPoint X) f

theorem **unitAt_genericPoint** (hf : f ≠ 0) : UnitAt (genericPoint X) f

theorem **germOf_eq_iff** {x : X} {f g : X.functionFieldˣ} : germOf x f = germOf x g ↔ UnitAt x ((g : X.functionField) / f)

theorem **exists_isEqn_nhds** (D : CartierDiv X) (x : X) : ∃ U : X.Opens, x ∈ U ∧ ∃ g, ∀ y ∈ U, D.IsEqn y g

theorem **exists_isEqn** (D : CartierDiv X) (x : X) : ∃ g, D.IsEqn x g

theorem **IsEqn.iff_unitAt** (h : D.IsEqn x g) : D.IsEqn x g' ↔ UnitAt x ((g' : X.functionField) / g)

theorem **IsEqn.exists_nhds** (h : D.IsEqn x g) : ∃ U : X.Opens, x ∈ U ∧ ∀ y ∈ U, D.IsEqn y g

theorem **ext_of_isEqn** (h : ∀ x, ∃ g, D.IsEqn x g ∧ E.IsEqn x g) : D = E

theorem **isEqn_genericPoint** (D : CartierDiv X) (g : X.functionFieldˣ) : D.IsEqn (genericPoint X) g

theorem **isEqn_zero** (x : X) : (0 : CartierDiv X).IsEqn x 1

theorem **IsEqn.add** (h : D.IsEqn x g) (h' : E.IsEqn x g') : (D + E).IsEqn x (g * g')

theorem **IsEqn.neg** (h : D.IsEqn x g) : (-D).IsEqn x g⁻¹

theorem **IsEqn.sub** (h : D.IsEqn x g) (h' : E.IsEqn x g') : (D - E).IsEqn x (g / g')

theorem **IsEqn.nsmul** (h : D.IsEqn x g) (n : ℕ) : (n • D).IsEqn x (g ^ n)

theorem **IsEqn.zsmul** (h : D.IsEqn x g) (n : ℤ) : (n • D).IsEqn x (g ^ n)

theorem **isEqn_principal** (g : X.functionField˟) (x : X) : (principal g).IsEqn x g

theorem **principal_one** : principal (1 : X.functionField˟) = 0

theorem **principal_mul** (g g' : X.functionField˟) : principal (g * g') = principal g + principal g'

theorem **mul_eqn_eq** (f : X.functionField) (g g' : X.functionField˟) : f * g = f * g' * ((g : X.functionField) / g')

theorem **IsEqn.regularAt_iff** (h : D.IsEqn x g) : D.RegularAt x f ↔ RegAt x (f * g)

theorem **IsEqn.generatesAt_iff** (h : D.IsEqn x g) : D.GeneratesAt x f ↔ UnitAt x (f * g)

theorem **IsEqn.vanishesAt_iff** (h : D.IsEqn x g) : D.VanishesAt x f ↔ ZeroAt x (f * g)

theorem **vanishesAt_iff** : D.VanishesAt x f ↔ D.RegularAt x f ∧ ¬ D.GeneratesAt x f

theorem **GeneratesAt.regularAt** (h : D.GeneratesAt x f) : D.RegularAt x f

theorem **GeneratesAt.ne_zero** (h : D.GeneratesAt x f) : f ≠ 0

theorem **VanishesAt.regularAt** (h : D.VanishesAt x f) : D.RegularAt x f

theorem **isEqn_iff_generatesAt_inv** : D.IsEqn x g ↔ D.GeneratesAt x (g : X.functionField)⁻¹

theorem **generatesAt_genericPoint** (D : CartierDiv X) (hf : f ≠ 0) : D.GeneratesAt (genericPoint X) f

theorem **vanishesAt_zero** (D : CartierDiv X) (x : X) : D.VanishesAt x 0

theorem **RegularAt.add** (h : D.RegularAt x f) (h' : D.RegularAt x f') : D.RegularAt x (f + f')

theorem **VanishesAt.add** (h : D.VanishesAt x f) (h' : D.VanishesAt x f') : D.VanishesAt x (f + f')

theorem **RegularAt.mul** (h : D.RegularAt x f) (h' : E.RegularAt x f') : (D + E).RegularAt x (f * f')

theorem **GeneratesAt.mul** (h : D.GeneratesAt x f) (h' : E.GeneratesAt x f') : (D + E).GeneratesAt x (f * f')

theorem **VanishesAt.mul** (h : D.VanishesAt x f) (h' : E.RegularAt x f') : (D + E).VanishesAt x (f * f')

theorem **regularAt_zero_iff** : (0 : CartierDiv X).RegularAt x f ↔ RegAt x f

theorem **generatesAt_zero_iff** : (0 : CartierDiv X).GeneratesAt x f ↔ UnitAt x f

theorem **RegularAt.regAt_mul** (h : D.RegularAt x f) (h' : RegAt x f') : D.RegularAt x (f' * f)

theorem **regularAt_add_principal** : (D + principal g).RegularAt x f ↔ D.RegularAt x (f * g)

theorem **eff_iff** : D.Eff ↔ ∀ x g, D.IsEqn x g → RegAt x g

theorem **eff_zero** : (0 : CartierDiv X).Eff

theorem **Eff.add** (h : D.Eff) (h' : E.Eff) : (D + E).Eff

theorem **Eff.eq_zero** (h : D.Eff) (h' : (-D).Eff) : D = 0

theorem **IsEqn.mem_supp_iff** (h : D.IsEqn x g) : x ∈ D.supp ↔ ¬ UnitAt x g

theorem **notMem_supp_iff** : x ∉ D.supp ↔ D.GeneratesAt x 1

theorem **Eff.mem_supp_iff** (h : D.Eff) : x ∈ D.supp ↔ D.VanishesAt x 1

theorem **genericPoint_notMem_supp** (D : CartierDiv X) : genericPoint X ∉ D.supp

theorem **supp_ne_top** (D : CartierDiv X) : D.supp ≠ ⊤

theorem **supp_eq_bot_iff** : D.supp = ⊥ ↔ D = 0

theorem **supp_add_le** (D E : CartierDiv X) : (D + E).supp ≤ D.supp ⊔ E.supp

theorem **algebraMap_top** : algebraMap Γ(X, ⊤) X.functionField = fnOf ⊤ trivial

theorem **mem_secOn** {S : Set X} : f ∈ D.secOn S ↔ ∀ x ∈ S, D.RegularAt x f

theorem **mem_sections** : f ∈ D.sections ↔ ∀ x, D.RegularAt x f

theorem **mem_sections_iff_eff** : (g : X.functionField) ∈ D.sections ↔ (principal g + D).Eff

theorem **IsRatSec.germOf_eq** (hr : IsRatSec L r) {i j : L.ι} {x : X} (hi : x ∈ L.U i) (hj : x ∈ L.U j) : germOf x (r j) = germOf x (r i)

theorem **presents_divOfRatSec** (hL : L.IsLB) (hr : IsRatSec L r) : Presents (divOfRatSec hL hr) L r

theorem **Presents.unique** (h : Presents D L r) (h' : Presents E L r) : D = E

theorem **val_frameRat** (i₀ : L.ι) (h₀ : genericPoint X ∈ L.U i₀) (i : L.ι) (h : genericPoint X ∈ L.U i) : (frameRat L i₀ h₀ i : X.functionField) = fnOf _ (memInf h h₀) (L.g i i₀ : Γ(X, L.U i ⊓ L.U i₀))

private theorem **units_val_res_eq_resG** {U V : X.Opens} (e : V ≤ U) (u : Γ(X, U)˟) : ((res e u : Γ(X, V)˟) : Γ(X, V)) = resG X e (u : Γ(X, U))

theorem **isRatSec_frameRat** (hL : L.IsLB) (i₀ : L.ι) (h₀ : genericPoint X ∈ L.U i₀) : IsRatSec L (frameRat L i₀ h₀)

private theorem **coe_res_eq_resG** {U V : X.Opens} (e : V ≤ U) (u : Γ(X, U)˟) : ((res e u : Γ(X, V)˟) : Γ(X, V)) = resG X e (u : Γ(X, U))

theorem **frameRat_self** (hL : L.IsLB) (i₀ : L.ι) (h₀ : genericPoint X ∈ L.U i₀) : frameRat L i₀ h₀ i₀ = 1

theorem **presents_ofLB** (hL : L.IsLB) (i₀ : L.ι) (h₀ : genericPoint X ∈ L.U i₀) : Presents (ofLB hL i₀ h₀) L (frameRat L i₀ h₀)

theorem **exists_trans** (i j : EqnChart D) : ∃ a : Γ(X, i.1.1 ⊓ j.1.1)˟, ∀ h, fnOf _ h a.val = (i.1.2 : X.functionField) / (j.1.2 : X.functionField)

theorem **fnOf_toLB_g** (i j : (toLB D).ι) (h) : fnOf _ h ((toLB D).g i j).val = (i.1.2 : X.functionField) / (j.1.2 : X.functionField)

private theorem **val_res_eq_resG** {U V : X.Opens} (e : V ≤ U) (u : Γ(X, U)˟) : ((res e u : Γ(X, V)˟) : Γ(X, V)) = resG X e (u : Γ(X, U))

theorem **isLB_toLB** (D : CartierDiv X) : (toLB D).IsLB where

theorem **presents_toLB** (D : CartierDiv X) : Presents D (toLB D) fun i => i.1.2

private theorem **fnOf_map_homOfLE** {U V : X.Opens} (e : V ≤ U) (h : genericPoint X ∈ V) (a : Γ(X, U)) : fnOf V h ((X.presheaf.map (homOfLE e).op).hom a) = fnOf U (e h) a

theorem **rat_mul** (h : Presents D L r) (s : L.Sec) (i : L.ι) (hi : genericPoint X ∈ L.U i) : h.rat s * r i = fnOf _ hi (s.f i)

theorem **rat_mem** (h : Presents D L r) (s : L.Sec) : h.rat s ∈ D.sections

theorem **rat_injective** (h : Presents D L r) : Function.Injective h.rat

private theorem **fnOf_presheaf_map_homOfLE** {U V : X.Opens} (e : V ≤ U) (h : genericPoint X ∈ V) (a : Γ(X, U)) : fnOf V h ((X.presheaf.map (homOfLE e).op).hom a) = fnOf U (e h) a

theorem **exists_rat_eq** (h : Presents D L r) {f : X.functionField} (hf : f ∈ D.sections) : ∃ s, h.rat s = f

@[simp] theorem **coe_secEquiv** (h : Presents D L r) (s : L.Sec) : (h.secEquiv s : X.functionField) = h.rat s

theorem **generatesAt_rat_iff** (h : Presents D L r) (s : L.Sec) {x : X} : D.GeneratesAt x (h.rat s) ↔ x ∈ nonzeroLocus s

theorem **vanishesAt_rat_iff** (h : Presents D L r) (s : L.Sec) {x : X} : D.VanishesAt x (h.rat s) ↔ x ∉ nonzeroLocus s

theorem **add_principal** (h : Presents D L r) (g : X.functionField˟) : Presents (D + principal g) L fun i => r i * g

private theorem **val_res_units_eq_resG** {U V : X.Opens} (e : V ≤ U) (u : Γ(X, U)˟) : ((res e u : Γ(X, V)˟) : Γ(X, V)) = resG X e (u : Γ(X, U))

private theorem **fnOf_resG_mul** {U U' V : X.Opens} (e : V ≤ U) (e' : V ≤ U') (h : genericPoint X ∈ V) (a : Γ(X, U)) (b : Γ(X, U')) : fnOf V h (resG X e a * resG X e' b) = fnOf U (e h) a * fnOf U' (e' h) b

theorem **tens** (h : Presents D L r) (h' : Presents E M r') : Presents (D + E) (tens L M) fun p => r p.1 * r' p.2

private theorem **fnOf_res_units** {U V : X.Opens} (e : V ≤ U) (h : genericPoint X ∈ V) (u : Γ(X, U)˟) : fnOf V h ((res e u : Γ(X, V)˟) : Γ(X, V)) = fnOf U (e h) (u : Γ(X, U))

private theorem **fnOf_res_mul_inv_mul** {A B C V : X.Opens} (ea : V ≤ A) (eb : V ≤ B) (ec : V ≤ C) (h : genericPoint X ∈ V) (a : Γ(X, A)˟) (b : Γ(X, B)˟) (c : Γ(X, C)˟) : fnOf V h ((res ea a * (res eb b)⁻¹ * res ec c : Γ(X, V)˟) : Γ(X, V)) = fnOf A (ea h) (a : Γ(X, A)) * (fnOf B (eb h) (b : Γ(X, B)))⁻¹ * fnOf C (ec h) (c : Γ(X, C))

theorem **lbIso** (h : Presents D L r) (h' : Presents D M r') : LBIso L M

theorem **presents_zero** : Presents (0 : CartierDiv X) (LB.triv X) fun _ => 1

theorem **toPic_apply** (D : CartierDiv X) : toPic D = cls _ (isLB_toLB D)

theorem **Presents.toPic_eq** (h : Presents D L r) : toPic D = cls L h.isLB

theorem **toPic_principal** (g : X.functionField˟) : toPic (principal g) = 0

theorem **toPic_surjective** : Function.Surjective (toPic (X

theorem **fnOf_comp_algX** (U : X.Opens) (h : genericPoint X ∈ U) : (fnOf U h).comp (algX p U) = constFn p

@[simp] theorem **formOf_resForm** {U V : X.Opens} (e : V ≤ U) (h : genericPoint X ∈ V) (ω : FormsOn p d U) : formOf V h (resForm e ω) = formOf U (e h) ω

theorem **isFreeGen_formOf** {U : X.Opens} (h : genericPoint X ∈ U) {ω : FormsOn p d U} (hω : IsFrame ω) : IsFreeGen X.functionField (formOf U h ω)

theorem **formOf_eq_ratio_smul** {U U' : X.Opens} (h : genericPoint X ∈ U) (h' : genericPoint X ∈ U') {ω : FormsOn p d U} {ω' : FormsOn p d U'} (hω : IsFrame ω) (hω' : IsFrame ω') : formOf U' h' ω' = fnOf (U ⊓ U') (memInf h h') (hω.ratio hω').val • formOf U h ω

theorem **gen_mem_baseOpen** : genericPoint X ∈ baseOpen p d

theorem **isFrame_baseFrame** : IsFrame (baseFrame p d)

theorem **isFreeGen_baseForm** : IsFreeGen X.functionField (baseForm p d)

theorem **baseForm_ne_zero** : baseForm p d ≠ 0

theorem **presents_canonical** : Presents (canonical p d) (CanonLB.all p d).bundle (frameRat _ (baseChart p d) (gen_mem_baseOpen p d))

theorem **isDivOfForm_canonical** : IsDivOfForm (canonical p d) (baseForm p d)

theorem **IsDivOfForm.unique** {η : RatForms p d} (hη : η ≠ 0) (h : IsDivOfForm D η) (h' : IsDivOfForm E η) : D = E

theorem **IsDivOfForm.smul** {η : RatForms p d} (h : IsDivOfForm D η) (g : X.functionField˟) : IsDivOfForm (D + principal g) ((g : X.functionField) • η)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsFiniteQCHomModOfExact

theorem **isGreatest_vals_frame** {h : ⊤ ≤ z ⁻¹ᵁ U} {u : Γ(F, U)} (hu : IsFrame fun _ : Fin 1 => u) (ν : Γ(F, U ⊓ gen X p) →ₛₗ[evq z hp h] C.S) : IsGreatest (vals z hp ν) ‖ν (resM F inf_le_left u)‖

theorem **sSup_compSec_frames** {E : X.Modules} {r : ℕ} (hE : IsVB E r) (π : E ⟶ F) [Epi π] [IsFiniteQC F] [IsFiniteQC G] (hG : IsVB (G.restrict (gen X p).ι) 1) (φ : F.restrict (gen X p).ι ≅ G.restrict (gen X p).ι) (h : ⊤ ≤ z ⁻¹ᵁ U) {u : Γ(F, U)} {v : Γ(G, U)} (hu : IsFrame fun _ : Fin 1 => u) (hv : IsFrame fun _ : Fin 1 => v) (a : Γ(X, U ⊓ gen X p)) (ha : homApp (homOfRestrict φ.hom) inf_le_right (resM F inf_le_left u) = a • resM G inf_le_left v) : sSup (nrmSet z (homMod E G) (compSec π φ.hom)) = ‖evq z hp h a‖

theorem **sec_ext_affine** {W : X.Opens} {a b : Γ(G, W)} (e : ∀ (T : X.Opens) (h : T ≤ W), IsAffineOpen T → resM G h a = resM G h b) : a = b

theorem **exists_hom_of_exact** (φ : Q ⟶ P) (π : P ⟶ F) (hex : ∀ T : X.Opens, IsAffineOpen T → Function.Exact (φ.app T) (π.app T)) (hs : ∀ T : X.Opens, IsAffineOpen T → Function.Surjective (π.app T)) {U : X.Opens} (ψ : Γ(homMod P G, U)) (hψ : ∀ (W : X.Opens) (h : W ≤ U) (m : Γ(Q, W)), homApp ψ h (φ.app W m) = 0) : ∃ χ : Γ(homMod F G, U), (homMap π (𝟙 G)).app U χ = ψ

theorem **isFiniteQC_homMod_of_exact** [IsLocallyNoetherian X] {r r' : ℕ} (hQ : IsVB Q r') (hP : IsVB P r) (φ : Q ⟶ P) (π : P ⟶ F) (hex : ∀ T : X.Opens, IsAffineOpen T → Function.Exact (φ.app T) (π.app T)) (hs : ∀ T : X.Opens, IsAffineOpen T → Function.Surjective (π.app T)) (w : φ ≫ π = 0) [IsFiniteQC G] : IsFiniteQC (homMod F G)

theorem **isFiniteQC_homMod_cokernel** [IsLocallyNoetherian X] {r r' : ℕ} (hQ : IsVB Q r') (hP : IsVB P r) (φ : Q ⟶ P) [IsFiniteQC G] : IsFiniteQC (homMod (cokernel φ) G)

theorem **homOfRestrict_injective** {U : X.Opens} : Function.Injective (homOfRestrict : (F.restrict U.ι ⟶ G.restrict U.ι) → _)

theorem **isUnit_natCast_gen** (p : ℕ) : IsUnit ((p : ℕ) : Γ((gen X p).toScheme, ⊤))

theorem **homApp_mulNatIso** (p : ℕ) {W : X.Opens} (h : W ≤ gen X p) (s : Γ(F, W)) : homApp (homOfRestrict (mulNatIso F p).hom) h s = ((p : ℕ) : Γ(X, W)) • s

theorem **nsmul_id_app** (n : ℕ) (U : X.Opens) (s : Γ(F, U)) : (n • 𝟙 F).app U s = n • s

theorem **not_isZero_cokernel_pow** {p : ℕ} {C : SCF p} (z : Sp (intO C.S) ⟶ X) (hp : 1 < p) (hF : IsVB F 1) {k : ℕ} (hk : k ≠ 0) : ¬ IsZero (cokernel (p ^ k • 𝟙 F))

@[reassoc (attr

theorem **nsmul_id_modN** (F : 𝒜) (n : ℕ) : n • 𝟙 (modN F n) = 0

@[reassoc] theorem **modNRed_redMap** {F G : 𝒜} {n m : ℕ} (h : n ∣ m) (φ : modN F m ⟶ modN G m) : modNRed F h ≫ redMap h φ = φ ≫ modNRed G h

@[simp] theorem **redMap_id** (F : 𝒜) {n m : ℕ} (h : n ∣ m) : redMap h (𝟙 (modN F m)) = 𝟙 _

theorem **redMap_comp** {F G H : 𝒜} {n m : ℕ} (h : n ∣ m) (φ : modN F m ⟶ modN G m) (ψ : modN G m ⟶ modN H m) : redMap h (φ ≫ ψ) = redMap h φ ≫ redMap h ψ

@[simp] theorem **redMap_self** {F G : 𝒜} {m : ℕ} (h : m ∣ m) (φ : modN F m ⟶ modN G m) : redMap h φ = φ

@[reassoc (attr

@[simp] theorem **redMap_modNMap** {F G : 𝒜} {n m : ℕ} (h : n ∣ m) (φ : F ⟶ G) : redMap h (modNMap φ m) = modNMap φ n

theorem **exists_shortExact_congr** {F G H : 𝒜} {f f' : F ⟶ G} {g g' : G ⟶ H} (hf : f = f') (hg : g = g') : (∃ w : f ≫ g = 0, (ShortComplex.mk _ _ w).ShortExact) ↔ ∃ w : f' ≫ g' = 0, (ShortComplex.mk _ _ w).ShortExact

theorem **isVB_free** (Z : ct.Mdl) (a m : ℕ) : IsVB (Z.free a m) m

theorem **nrm_eq** (F : ct.Sh Z) (f : ct.Sec F) (y : ct.Pt Z) : ct.nrm F f y = sSup (nrmSet y.z F.toMod f)

theorem **nrm_ofMod** (F : Z.X.Modules) [IsFiniteQC F] (f : Γ(F, Z.genO)) (y : ct.Pt Z) : ct.nrm (Z.ofMod F) ((ct.presIso F).inv.app _ f) y = sSup (nrmSet y.z F f)

theorem **isLocallyNoetherian** (Z : ct.Mdl) : IsLocallyNoetherian Z.X

theorem **exists_homSh_iso** (F₁ F₂ : ct.Sh Z) : ∃ e : homMod F₁.toMod F₂.toMod ≅ (ct.homSh F₁ F₂).toMod, ∀ φ : ct.GIso F₁ F₂, ct.isoSec φ = e.hom.app _ (homOfRestrict φ.iso.hom)

theorem **isoSec_injective** (F₁ F₂ : ct.Sh Z) : Function.Injective (ct.isoSec : ct.GIso F₁ F₂ → _)

theorem **isVB_homSh** {F₁ F₂ : ct.Sh Z} {r : ℕ} (h₁ : ct.IsInv F₁) (h₂ : IsVB F₂.toMod r) : IsVB (ct.homSh F₁ F₂).toMod r

theorem **homSh_inv** (F₁ F₂ : ct.Sh Z) (h₁ : ct.IsInv F₁) (h₂ : ct.IsInv F₂) : ct.IsInv (ct.homSh F₁ F₂)

theorem **nrm_isoSec_eq** {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : ct.nrm (ct.homSh F₁ F₂) (ct.isoSec φ) y = sSup (nrmSet y.z (homMod F₁.toMod F₂.toMod) (compSec (𝟙 _) φ.iso.hom))

theorem **nrm_isoSec_pos_symm** {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : 0 < ct.nrm (ct.homSh F₁ F₂) (ct.isoSec φ) y ∧ ct.nrm (ct.homSh F₂ F₁) (ct.isoSec (ct.isoSymm φ)) y = (ct.nrm (ct.homSh F₁ F₂) (ct.isoSec φ) y)⁻¹

theorem **nrm_isoSec** {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : ct.nrm (ct.homSh F₁ F₂) (ct.isoSec φ) y = (p : ℝ) ^ (-(ct.vrat φ y))

theorem **vrat_symm** {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : ct.vrat (ct.isoSymm φ) y = -(ct.vrat φ y)

theorem **vrat_refl** (F : ct.Sh Z) (h : ct.IsInv F) (y : ct.Pt Z) : ct.vrat (ct.isoRefl F h) y = 0

theorem **rpow_neg_vrat** {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) {U : Z.X.Opens} (h : ⊤ ≤ y.z ⁻¹ᵘ U) {u : Γ(F₁.toMod, U)} {v : Γ(F₂.toMod, U)} (hu : VB.IsFrame fun _ : Fin 1 => u) (hv : VB.IsFrame fun _ : Fin 1 => v) (a : Γ(Z.X, U ⊓ Z.genO)) (ha : homApp (homOfRestrict φ.iso.hom) inf_le_right (resM _ inf_le_left u) = a • resM _ inf_le_left v) : (p : ℝ) ^ (-(ct.vrat φ y)) = ‖evq y.z ct.hp.ne_zero h a‖

theorem **vrat_eq_one** {F₁ F₂ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) {U : Z.X.Opens} (h : ⊤ ≤ y.z ⁻¹ᵘ U) {u : Γ(F₁.toMod, U)} {v : Γ(F₂.toMod, U)} (hu : VB.IsFrame fun _ : Fin 1 => u) (hv : VB.IsFrame fun _ : Fin 1 => v) (ha : homApp (homOfRestrict φ.iso.hom) inf_le_right (resM _ inf_le_left u) = ((p : ℕ) : Γ(Z.X, U ⊓ Z.genO)) • resM _ inf_le_left v) : ct.vrat φ y = 1

theorem **vrat_mulP** (F : ct.Sh Z) (hF : ct.IsInv F) (y : ct.Pt Z) : ct.vrat (ct.mulP F hF) y = 1

theorem **vrat_trans** {F₁ F₂ F₃ : ct.Sh Z} (φ : ct.GIso F₁ F₂) (ψ : ct.GIso F₂ F₃) (y : ct.Pt Z) : ct.vrat (φ.trans ψ) y = ct.vrat φ y + ct.vrat ψ y

theorem **oSh_inv** (Z : ct.Mdl) : ct.IsInv (ct.oSh Z)

theorem **isFrame_oOne** (Z : ct.Mdl) : VB.IsFrame fun _ : Fin 1 => ct.oOne Z

theorem **nrm_secOf** {F : ct.Sh Z} (φ : ct.GIso (ct.oSh Z) F) (y : ct.Pt Z) : ct.nrm F (ct.secOf φ) y = (p : ℝ) ^ (-(ct.vrat φ y))

theorem **SES.exactMod_ofMaps_iff** {F₁ F₂ F₃ : ct.Sh Z} (f : F₁.toMod ⟶ F₂.toMod) (g : F₂.toMod ⟶ F₃.toMod) {k k₀ : ℕ} (h : k ≤ k₀) : (SES.ofMaps f g k₀).ExactMod k ↔ ∃ w : modNMap f (p ^ k) ≫ modNMap g (p ^ k) = 0, (ShortComplex.mk _ _ w).ShortExact

theorem **SES.exactMod_self_iff** (s : ct.SES Z) : s.ExactMod s.k₀ ↔ ∃ w : s.f ≫ s.g = 0, (ShortComplex.mk _ _ w).ShortExact

theorem **SES.exactMod_of_splitting** {F₁ F₂ F₃ : ct.Sh Z} {f : F₁.toMod ⟶ F₂.toMod} {g : F₂.toMod ⟶ F₃.toMod} {w : f ≫ g = 0} (σ : (ShortComplex.mk f g w).Splitting) {k k₀ : ℕ} (h : k ≤ k₀) : (SES.ofMaps f g k₀).ExactMod k

theorem **SES.exactMod_sum** (F G : ct.Sh Z) {k k₀ : ℕ} (h : k ≤ k₀) : (SES.sum F G k₀).ExactMod k

theorem **SES.isZero_of_exactMod_zero** {F₁ F₂ F₃ : ct.Sh Z} {k : ℕ} (h : ((F₁, F₂, F₃, k, 0, 0) : ct.SES Z).ExactMod k) : IsZero (modN F₂.toMod (p ^ k))

theorem **SES.not_exactMod_zero** {F₁ F₂ F₃ : ct.Sh Z} (h : ct.IsInv F₂) (y : ct.Pt Z) {k : ℕ} (hk : k ≠ 0) : ¬ ((F₁, F₂, F₃, k, 0, 0) : ct.SES Z).ExactMod k

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsGrpPullbackAbVarModel

theorem **raynaud_spec** : (splitExt D.toPre).IsGt (raynaud U) ∧ Represents (raynaudPts U) ∧ (∀ T hT w x, raynaudCmp U T hT w = x ↔ FormalPoint.IsPointOf U (raynaudPts U T w) x) ∧ ∀ T T' hT hT' (k : T' ⟶ T) (x : T ⟶ raynaud U), raynaudCmp U T' hT' (k ≫ x) = k ≫ raynaudCmp U T hT x

theorem **raynaudAct_spec** (a : Λ) (T : Over (Sp R)) (x : T ⟶ raynaud U) : FormalPoint.IsSmulPt ιD a (raynaudPts U T x) (raynaudPts U T (x ≫ (raynaudAct U ιD).act a))

theorem **raynaudCmp_act** {ιA : A.OAct Λ} {ιG : RingAct G Λ} (hO : U.OLinear Λ ιD ιA ιG) (T : Over (Sp R)) (hT : IsAdicColimit I T) (x : T ⟶ raynaud U) (a : Λ) : raynaudCmp U T hT (x ≫ (raynaudAct U ιD).act a) = raynaudCmp U T hT x ≫ ιG.act a

theorem **existsUnique_corr_dual** : (∀ y : specO 0 (DualNumber 0) ⟶ P.G, ∃! x, Corr P x y) ∧ ∀ x : specO 0 (DualNumber 0) ⟶ P.Gt, ∃! y, Corr P x y

theorem **exists_lieCmp** : ∃ e : LieG P.Gt ≃ LieG P.G, ∀ x, Corr P x.1 (e x).1

theorem **eq_lieCmp_iff** (x : LieG P.Gt) (y : specO 0 (DualNumber 0) ⟶ P.G) : y = (P.lieCmp x).1 ↔ Corr P x.1 y

theorem **corr_lieCmp** (x : LieG P.Gt) : Corr P x.1 (P.lieCmp x).1

theorem **lieCmp_add** (x y : LieG P.Gt) : P.lieCmp (lieAdd x y) = lieAdd (P.lieCmp x) (P.lieCmp y)

theorem **lieCmp_scale** (c : 0) (x : LieG P.Gt) : P.lieCmp (AvDefs.lieScale P.Gt c x) = AvDefs.lieScale P.G c (P.lieCmp x)

theorem **omegaCmp_apply** (f : omega P.G) (x : LieG P.Gt) : (P.omegaCmp f).1 x = f.1 (P.lieCmp x)

theorem **integers_intO** : (fldVal C).Integers ɪ(intO ɪC.S) where

theorem **norm_p_lt_one** : ‖algebraMap ɪ(intO ɪC.S) ɪC.S (p : ɪ(intO ɪC.S))‖ < 1

theorem **norm_padicEmb** (x : ℚ_[p]) : ‖padicEmb C x‖ = ‖x‖

theorem **exists_eps_of_hom** (f : F →+* C) : ∃ ε : Eps p C, ∀ (n : ℕ) (h : zeta (p ^ n) ∈ F), (ε.1.lev n : C) = f (zeta (p ^ n), h)

theorem **isGrpPullback_abVarι** : IsGrpPullback (abVar P).A Y.oF.t.G P.y (abVarι P)

theorem **abAct_ι** (a : 𝒪 E) : letI

theorem **coe_embInt** (τ : E →+* ℂ) (a : 𝒪 E) : (embInt P τ a : ɪC.S) = P.emb τ a

theorem **embAt_injective** : Function.Injective (embAt P)

theorem **isSplitting_embInt** : IsSplitting (embC ɪC.S (embInt P))

theorem **dat_fac** : spMap ((algebraMap ɪ(intO ɪC.S) ɪC.S).comp A.f) ≫ A.ι = P.y ≫ Y.j

theorem **isGrpPullback_abVar_model** : IsGrpPullback (abVar P).A Y.t.G (spMap ((algebraMap ɪ(intO ɪC.S) ɪC.S).comp A.f) ≫ A.ι) (abVarι P ≫ Y.m.φ)

theorem **abToDat_φ** : abToDat A ≫ A.m.φ = abVarι P ≫ Y.m.φ

theorem **abToDat_hom** : abToDat A ≫ A.t.G.hom = (abVar P).A.hom ≫ spMap ((algebraMap ɪ(intO ɪC.S) ɪC.S).comp A.f)

theorem **isGrpPullback_abToDat** : IsGrpPullback (abVar P).A A.t.G (spMap ((algebraMap ɪ(intO ɪC.S) ɪC.S).comp A.f)) (abToDat A)

theorem **abAct_abToDat** (a : 𝒪 E) : letI

theorem **exists_pow_mem_span** : ∃ n : ℕ, ∀ a ∈ maximalIdeal A.R ^ n, A.f a ∈ Ideal.span {(p : ɪ(intO ɪC.S))}

theorem **isGrpPullback_htToModel** : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsGrpPullbackPtC

theorem **mem_torsSub** {n : ℕ} (hn : n ≠ 0) (a : Over.mk x ⟶ t.G) : a ∈ torsSub t x n ↔ a ^ n = 1

theorem **mem_thSub** {n : ℕ} (θ : ThetaGrp t.G t.PP.L n (Over.mk x)) : θ ∈ thSub t x n ↔ θ.pt ∈ torsSub t x n

theorem **ovr_surjective** [IsLocalRing R] (n : ℕ) : Function.Surjective (ovr t x n)

theorem **scal_injective** (n : ℕ) : Function.Injective (scal t x n)

theorem **scal_comm** (n : ℕ) (a : Rˣ) (g : thSub t x n) : scal t x n a * g = g * scal t x n a

theorem **Sec.IsBasis.exists_unit** {k : ℤ} {θ θ' : Sec t x k} (h : θ.IsBasis) (h' : θ'.IsBasis) : ∃ u : Rˣ, θ' = u • θ

theorem **Sec.IsBasis.ne_zero** [Nontrivial R] {k : ℤ} {θ : Sec t x k} (h : θ.IsBasis) : θ ≠ 0

theorem **act_scal** (n : ℕ) (a : Rˣ) (s : Sec t x n) : act t x n (scal t x n a) s = (a : R) • s

theorem **ovr_eq_one_iff** (hp : ConstFn t x) (n : ℕ) (g : thSub t x n) : ovr t x n g = 1 ↔ ∃ a, scal t x n a = g

theorem **exists_normGen** {α : 𝒪 E} {n : ℕ} (h : α * H.bar α = n) : ∃ u : LB.Gen (LB.LinComb t.PP.L (K

theorem **abFibre_iff** {B : Scheme.{0}} {φ : B ⟶ t.G.left} {q : B ⟶ Z} (h : IsPullback φ q t.G.hom x) : AbFibre t x g ↔ IsProper q ∧ SmoothOfRelativeDimension g q

theorem **AbFibre.tens** (h : AbFibre t x g) (Λ : Unim E) : AbFibre (tens t Λ) x (Λ.n * g)

theorem **AbFibre.isAbelianScheme** (h : AbFibre t x g) : IsAbelianScheme ((Over.pullback x).obj t.G) where

theorem **AbFibre.constFn** {R : Type} [CommRing R] {x : Sp R ⟶ X} (h : AbFibre t x g) : ConstFn t x

theorem **AbFibre.card_tors** (h : AbFibre t x g) {n : ℕ} (hn : n ≠ 0) : Nat.card (Tors t x n) = n ^ (2 * g)

theorem **AbFibre.finite_tors** (h : AbFibre t x g) (n : ℕ) : Finite (Tors t x n)

theorem **gen_ptO** : spMap (algebraMap ɪ(intO ɪC.S) ɪC.S) ≫ ptO A = ptC A

theorem **ptO_ι** : ptO A ≫ A.ι = P.yt

theorem **ptC_ι** : ptC A ≫ A.ι = P.y ≫ Y.j

theorem **isGrpPullback_ptC** : IsGrpPullback (abVar P).A A.t.G (ptC A) (abToDat A)

theorem **abFibre_ptC** : AbFibre A.t (ptC A) (Module.finrank ℚ E / 2 * (Dw.p + Dw.q))

theorem **exists_alp** {D : D7} {p n : ℕ} (h : NHyp D p n) : ∃ α : 𝒪 D.host.E, (α : D.host.E) * D.host.conj (α : D.host.E) = (n : D.host.E) ∧ ((∃ mm : ℕ, n = mm ^ 2 ∧ α = (mm : 𝒪 D.host.E)) ∨ (α : D.host.E) ∈ D.host.K)

theorem **alpOf_spec** {D : D7} {p n : ℕ} (h : NHyp D p n) : (alpOf D p n : D.host.E) * D.host.conj (alpOf D p n : D.host.E) = (n : D.host.E) ∧ ((∃ mm : ℕ, n = mm ^ 2 ∧ alpOf D p n = (mm : 𝒪 D.host.E)) ∨ (alpOf D p n : D.host.E) ∈ D.host.K)

theorem **alpOf_mul_bar** (A : Amb D) {n : ℕ} (h : NHyp D p n) : alpOf D p n * A.H.bar (alpOf D p n) = n

theorem **abFibre_tensAt** : Thg.AbFibre (ct.tensAt y Λ) (ct.ptC y) (D.R.f * N * Λ.n)

theorem **card_Tors** {n : ℕ} (hn : 0 < n) : Nat.card ((thgTrue ct).Tors y Λ n) = n ^ (2 * (D.R.f * N * Unim.n Λ))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsHalfPtIff

theorem **posOr1_pos** (x : ℝ) : 0 < posOr1 x

theorem **isHalfPt_iff** (D : D7) (Λ : AddSubgroup ℂ) (t : ℂ) : IsHalfPt D.host.d Λ t ↔ GL3Theta.IsTwoTorsionPointUnkilled D Λ t

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsIrreducibleIff

theorem **isIrreducible_iff** {V : Type} [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) : ρ.IsIrreducible ↔ Nontrivial V ∧ ∀ W : Submodule ℂ V, (∀ g, ∀ v ∈ W, ρ g v ∈ W) → W = ⊥ ∨ W = ⊤

@[simp] theorem **twRep_apply** {V : Type} [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (χ : G →* ℂˣ) (g : G) (v : V) : twRep ρ χ g v = ((χ g : ℂˣ) : ℂ) • ρ g v

theorem **Irr.ofChar_inj** {χ χ' : G →* ℂˣ} {hχ : IsOpen {g | χ g = 1}} {hχ' : IsOpen {g | χ' g = 1}} (h : Irr.ofChar χ hχ = Irr.ofChar χ' hχ') : χ = χ'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsKitWithKitWith

theorem **isKitWith_kitWith** (Φ : s.Flag) (key : KeySec s.Xo κ.c) {x : ToroidalCompactificationData s} {B : s.Bd L} (hB : IsBdOf x L B) (P : PolOf x Φ) {cts : Ats S κ} (hc : cts.Over jC) (hj : jC.Pinned) (hA : AdmEx S κ) : IsKitWith F s L jC (kitWith F L Φ key x B P cts hj hA) (degFlat s Φ (metsOf x Φ)) x P cts hj hA

theorem **pad_eq** : k.pad = padAt s (locStOf F cts) (globStOf F cts) (admissible_representation_factorisation S κ jC hj hA) (flatSt s k.Φ (metsOf x k.Φ)) (arIn s k.Φ (metsOf x k.Φ))

theorem **isBetti** {c : s.Comp} {τ : D.what₂.Tau} (hb : ∃ b, IsBetti s k.Φ L c τ (flatSec s k.Φ c τ) b) : IsBetti s k.Φ L c τ (flatSec s k.Φ c τ) (k.fl c τ)

theorem **r_eq** (c : s.Comp) {z} (hz : z ∈ DomI D.what₂.Tau j (N − j)) : (k.B c).r z = x.r κ.σ₀ (s.toCpt L c z)

theorem **bd_ne_invariant** (hb : (genBd s.T.Z : Set (gen s.T.Z : Scheme.{0})).Nonempty) : k.B ≠ GlobalTheory.TrueCx.invariantB _ _ (GlobalTheory.TrueCx.first_index_eq_of_unfolded _ s.isUnf S.p_eq) _

theorem **fl_eq** : k.fl = flBetti s k.Φ L

theorem **bd_ne_invariant** (hb : (genBd s.T.Z : Set (gen s.T.Z : Scheme.{0})).Nonempty) : k.B ≠ GlobalTheory.TrueCx.invariantB _ _ (GlobalTheory.TrueCx.first_index_eq_of_unfolded _ s.isUnf S.p_eq) _

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsLinearHomPull

theorem **exists_torPt** {T : Over (Spec R)} (x : T ⟶ splitTorus (Spec R) (Chr D)) : ∃ g : GtPt D T, g.b = 1 ∧ ∀ (χ : D.X) (Z : Scheme.{0}) (r : Z ⟶ T.left) y J, Good D.P.L y J → (val (g.s χ) r y J).val = gs r y ⊤ ((coord x).hom (AddMonoidAlgebra.single χ 1))

theorem **Φ_b** {T : Over (Spec R)} (x : T ⟶ Y.H) : (Y.Φ T x).b = x ≫ Y.q

theorem **q_linear** : IsLinearHom Y.act (actB oa) Y.q

theorem **Φ_i** {T : Over (Spec R)} (x : T ⟶ splitTorus (Spec R) (Chr D)) : (Y.Φ T (x ≫ Y.i)).b = 1 ∧ ∀ (χ : D.X) y J, Good D.P.L y J → (cv (Y.Φ T (x ≫ Y.i)) χ y J).val = gs (𝟙 _) y ⊤ ((coord x).hom (AddMonoidAlgebra.single χ 1))

theorem **i_q** : Y.i ≫ Y.q = 1

theorem **i_linear** : IsLinearHom (torAct D O) Y.act Y.i

theorem **isKernel** : IsPullback Y.i (toUnit _) Y.q η[D.P.A]

theorem **isLinearHom_pull** {a : OAct O G} {a' : OAct O G'} {φ : G ⟶ G'} (h : IsLinearHom a a' φ) : IsLinearHom (a.pullback (spMap f)) (a'.pullback (spMap f)) ((Over.pullback (spMap f)).map φ)

theorem **hom_lieMap** {a : OAct O G} {a' : OAct O G'} {φ : G ⟶ G'} (h : IsLinearHom a a' φ) (v : LieAt f a) : (lieMap f h v).hom = v.hom ≫ (Over.pullback (spMap f)).map φ

theorem **lieMap_act** {a : OAct O G} {a' : OAct O G'} {φ : G ⟶ G'} (h : IsLinearHom a a' φ) (x : O) (v : LieAt f a) : lieMap f h (x • v) = x • lieMap f h v

theorem **coe_eigMap** {a : OAct O G} {a' : OAct O G'} {φ : G ⟶ G'} (h : IsLinearHom a a' φ) (σ : O →+* S) (v : LieEig f a σ) : (eigMap f h σ v).1 = lieMap f h v.1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsLogAExtends

theorem **exists_zeta** (l : ℕ) [Fact l.Prime] : ∃ ζ : ℕ → Om l, ζ 0 = 1 ∧ (∀ m, ζ (m + 1) ^ l = ζ m) ∧ ζ 1 ≠ 1

theorem **rootT_ofRoots** (f : ℕ → Per.OC l) (hf : ∀ n, f (n + 1) ^ l = f n) (m : ℕ) : rootT (ReciprocityTheory.PGsh.ofRoots f hf) m = f m

theorem **exists_tilt** (ζ : ℕ → Om l) (h0 : ζ 0 = 1) (hs : ∀ m, ζ (m + 1) ^ l = ζ m) : ∃ ε : Per.Fl l, ∀ m, (rootT ε m : ℂ_[l]) = algebraMap (Om l) ℂ_[l] (ζ m)

theorem **isEps_of_root** {ζ : ℕ → Om l} (h0 : ζ 0 = 1) (h1 : ζ 1 ≠ 1) {ε : Per.Fl l} (h : ∀ m, (rootT ε m : ℂ_[l]) = algebraMap (Om l) ℂ_[l] (ζ m)) : IsEps ε

theorem **exists_isEps** (l : ℕ) [Fact l.Prime] : ∃ e : Per.Fl l, IsEps e

theorem **not_isEps_one** : ¬ IsEps (1 : Per.Fl l)

theorem **core_isEps** (C : Core l) : IsEps C.ε

theorem **exists_chi** (l : ℕ) [Fact l.Prime] : ∃ χ : GQ l →* ℤ_[l]ˣ, ∀ (g : GQ l) (m : ℕ) (x : Om l), x ^ (l ^ m) = 1 → g x = x ^ (PadicInt.toZModPow m (χ g : ℤ_[l])).val

theorem **mu_eq** (e : Per.Fl l) : mu e = xi e * mu ((frobeniusEquiv (Per.Fl l) l).symm e)

theorem **theta_mu** (e : Per.Fl l) : Per.θ l (mu e) = e.untilt − 1

theorem **theta_xi** {e : Per.Fl l} (h : IsEps e) : Per.θ l (xi e) = 0

theorem **core_isCrisDR** (C : Core l) : Per.IsCrisDR l C.ι

theorem **isLogA_one** : IsLogA (1 : Per.Fl l) 0

theorem **mem_pc** (R : Type) [CommRing R] (r : ℕ) (x : GlobalTheory.AdicMaps.PHat l R) : x ∈ ((Ideal.span {(l : R)}) ^ r • ⊤ : Submodule R (GlobalTheory.AdicMaps.PHat l R)) ↔ ∃ z : GlobalTheory.AdicMaps.PHat l R, x = ((l : ℕ) : GlobalTheory.AdicMaps.PHat l R) ^ r * z

theorem **pc_sep** (R : Type) [CommRing R] (x : GlobalTheory.AdicMaps.PHat l R) (h : ∀ r, ∃ z : GlobalTheory.AdicMaps.PHat l R, x = ((l : ℕ) : GlobalTheory.AdicMaps.PHat l R) ^ r * z) : x = 0

theorem **pc_lim** (R : Type) [CommRing R] (s : ℕ → R) (hs : ∀ r n, l ^ r ≤ n → ∃ z : R, s n − s (l ^ r) = (l : R) ^ r * z) : ∃ y : GlobalTheory.AdicMaps.PHat l R, ∀ r n, l ^ r ≤ n → ∃ z : GlobalTheory.AdicMaps.PHat l R, y − algebraMap R _ (s n) = ((l : ℕ) : GlobalTheory.AdicMaps.PHat l R) ^ r * z

private theorem **acris_sep** (x : Per.Acris l) (h : ∀ r, ∃ z : Per.Acris l, x = ((l : ℕ) : Per.Acris l) ^ r * z) : x = 0

private theorem **acris_lim** (s : ℕ → Per.Acris0 l) (hs : ∀ r n, l ^ r ≤ n → ∃ z : Per.Acris0 l, s n − s (l ^ r) = (l : Per.Acris0 l) ^ r ∗ z) : ∃ y : Per.Acris l, ∀ r n, l ^ r ≤ n → ∃ z : Per.Acris l, y − algebraMap (Per.Acris0 l) (Per.Acris l) (s n) = ((l : ℕ) : Per.Acris l) ^ r ∗ z

theorem **isLogA_unique** {e : Per.Fl l} {y y' : Per.Acris l} (h : IsLogA e y) (h' : IsLogA e y') : y = y'

theorem **exists_isLogA** {e : Per.Fl l} (he : e.untilt = 1) : ∃ y : Per.Acris l, IsLogA e y

theorem **isLogA_extends** {f : Per.Ainf l →+* Per.Ainf l} {F : Per.Acris l →+* Per.Acris l} (hf : GlobalTheory.CrisMaps.Extends f F) {e e' : Per.Fl l} (h : f (Per.tm e) = Per.tm e') {y : Per.Acris l} (hy : IsLogA e y) : IsLogA e' (F y)

theorem **isLogA_phiA** {e : Per.Fl l} {y : Per.Acris l} (hy : IsLogA e y) : IsLogA (e ^ l) (phiA l y)

theorem **isLogA_sigA** (g : Per.Gq l) {e : Per.Fl l} {y : Per.Acris l} (hy : IsLogA e y) : IsLogA (Per.σF g e) (sigA g y)

theorem **exists_part** (l : ℕ) [Fact l.Prime] : ∃ (ζ : ℕ → Om l) (ε : Per.Fl l) (t : Per.Acris l), ζ 0 = 1 ∧ (∀ m, ζ (m + 1) ^ l = ζ m) ∧ ζ 1 ≠ 1 ∧ (∀ m, (rootT ε m : ℂ_[l]) = algebraMap (Om l) ℂ_[l] (ζ m)) ∧ IsLogA ε t

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsNatDiagIffIsDiag

theorem **ProperLE.of_subset** {ty' ty : (E →+* ℂ) → Finset ℕ} (h : ∀ σ, ty' σ ⊆ ty σ) : ProperLE Dw ty' ty

theorem **SameFlag.of_properLE** {l : Level H} {c : Coeff l} {X : Scheme.{0}} {t : SAB H X} {R : Type} [CommRing R] {f : c.F →+* R} {x : Sp R ⟶ X} {u u' : Vecs Dw t R} {ty' ty : (E →+* ℂ) → Finset ℕ} (hu : IsFrame t f x u) (hu' : IsFrame t f x u') (hty : ProperLE Dw ty' ty) (h : SameFlag ty t x u u') : SameFlag ty' t x u u'

theorem **ofMod_iso** {F : Z.X.Modules} (h : ∃ S : Z.Sh, Nonempty (S.toMod ≅ F)) : Nonempty ((Z.ofMod F).toMod ≅ F)

theorem **blowUp_spec** {Z : Mdl Y i₀} {J : Z.Idl} (h : ∃ Z', IsNBlow Z J Z') : IsNBlow Z J (blowUp Z J)

theorem **natDiagCl_spec** {Z₁ Z₂ : Mdl Y i₀} (h : ∃ Z', IsNatDiag Z₁ Z₂ Z') : IsNatDiag Z₁ Z₂ (natDiagCl Z₁ Z₂)

theorem **diagCl_spec** {Z₁ Z₂ : Mdl Y i₀} (h : ∃ Z', IsDiag Z₁ Z₂ Z') : IsDiag Z₁ Z₂ (diagCl Z₁ Z₂)

theorem **NatCorr.refl** (Z : Mdl Y i₀) : NatCorr Z Z

theorem **NatCorr.trans** {Z₁ Z₂ Z₃ : Mdl Y i₀} (h : NatCorr Z₁ Z₂) (h' : NatCorr Z₂ Z₃) : NatCorr Z₁ Z₃

theorem **Corr.natCorr** {Z₁ Z₂ : Mdl Y i₀} (h : Corr Z₁ Z₂) : NatCorr Z₁ Z₂

theorem **IsMap.isNatMap** {Z₁ Z₂ : Mdl Y i₀} {f : Z₁.X ⟶ Z₂.X} (h : IsMap Z₁ Z₂ f) : IsNatMap Z₁ Z₂ f

theorem **IsNatMap.isMap** {Z₁ Z₂ : Mdl Y i₀} {f : Z₁.X ⟶ Z₂.X} (h : IsNatMap Z₁ Z₂ f) (hty : ∀ σ, Z₂.ty σ ⊆ Z₁.ty σ) : IsMap Z₁ Z₂ f

theorem **isNatDiag_iff_isDiag** {Z₁ Z₂ Z' : Mdl Y i₀} (hty : ∀ σ, Z₂.ty σ ⊆ Z₁.ty σ) : IsNatDiag Z₁ Z₂ Z' ↔ IsDiag Z₁ Z₂ Z'

theorem **Bases.natCorr_base** (B : Bases Y i₀) (Z : Mdl Y i₀) : NatCorr Z (B.X none)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsNatDiagIffIsDiagProperPart

theorem **isFrame_chart_bc** (Z : Mdl Y i₀) (a : Fin Z.m) {k : Type} [CommRing k] (φ : Γ(Z.X, (Z.U a).1) →+* k) : IsFrame (Y Z.i).t (φ.comp (Z.alg a)) (spMap φ ≫ Z.chart a ≫ Z.π) ((Z.fr a).bc φ)

theorem **comp_alg_eq** (Z : Mdl Y i₀) {a a' : Fin Z.m} {k : Type} [Field k] {φ : Γ(Z.X, (Z.U a).1) →+* k} {φ' : Γ(Z.X, (Z.U a').1) →+* k} (h : spMap φ ≫ Z.chart a = spMap φ' ≫ Z.chart a') : φ.comp (Z.alg a) = φ'.comp (Z.alg a')

theorem **flagAt_of_properLE** (Z : Mdl Y i₀) {ty ty' : (E →+* ℂ) → Finset ℕ} (h : ProperLE Dw ty' ty) {k : Type} [Field k] {w : Sp k ⟶ Z.X} {u : Vecs Dw (Y Z.i).t k} (hw : Pre.FlagAt { Z.toPre with ty

theorem **flagAt_withType** (Z : Mdl Y i₀) {ty : (E →+* ℂ) → Finset ℕ} (h₁ : ProperLE Dw ty Z.ty) (h₂ : ProperLE Dw Z.ty ty) {k : Type} [Field k] {w : Sp k ⟶ Z.X} {u : Vecs Dw (Y Z.i).t k} : (Z.withType ty h₁ h₂).FlagAt w u ↔ Z.FlagAt w u

theorem **natImg_withType_right** {Z₁ Z₂ : Mdl Y i₀} {ty : (E →+* ℂ) → Finset ℕ} (h₁ : ProperLE Dw ty Z₂.ty) (h₂ : ProperLE Dw Z₂.ty ty) {k : Type} [Field k] {w₁ : Sp k ⟶ Z₁.X} {w₂ : Sp k ⟶ Z₂.X} : NatImg Z₁ (Z₂.withType ty h₁ h₂) w₁ w₂ ↔ NatImg Z₁ Z₂ w₁ w₂

theorem **natImg_withType_left** {Z₁ Z₂ : Mdl Y i₀} {ty : (E →+* ℂ) → Finset ℕ} (h₁ : ProperLE Dw ty Z₁.ty) (h₂ : ProperLE Dw Z₁.ty ty) {k : Type} [Field k] {w₁ : Sp k ⟶ Z₁.X} {w₂ : Sp k ⟶ Z₂.X} : NatImg (Z₁.withType ty h₁ h₂) Z₂ w₁ w₂ ↔ NatImg Z₁ Z₂ w₁ w₂

theorem **isNatMap_withType_right** {Z₁ Z₂ : Mdl Y i₀} {ty : (E →+* ℂ) → Finset ℕ} (h₁ : ProperLE Dw ty Z₂.ty) (h₂ : ProperLE Dw Z₂.ty ty) {f : Z₁.X ⟶ Z₂.X} : IsNatMap Z₁ (Z₂.withType ty h₁ h₂) f ↔ IsNatMap Z₁ Z₂ f

theorem **isNatMap_withType_left** {Z₁ Z₂ : Mdl Y i₀} {ty : (E →+* ℂ) → Finset ℕ} (h₁ : ProperLE Dw ty Z₁.ty) (h₂ : ProperLE Dw Z₁.ty ty) {f : Z₁.X ⟶ Z₂.X} : IsNatMap (Z₁.withType ty h₁ h₂) Z₂ f ↔ IsNatMap Z₁ Z₂ f

theorem **isNatMap_id_withType** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) {ty ty' : (E →+* ℂ) → Finset ℕ} (h₁ : ProperLE Dw ty Z.ty) (h₂ : ProperLE Dw Z.ty ty) (h₁' : ProperLE Dw ty' Z.ty) (h₂' : ProperLE Dw Z.ty ty') : IsNatMap (Z.withType ty h₁ h₂) (Z.withType ty' h₁' h₂') (𝟙 Z.X)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsOpenImmersionGeneric

theorem **eps_over** (c : R) : spMap (epsMul c) ≫ spMap (algebraMap R (DualNumber R)) = spMap (algebraMap R (DualNumber R))

theorem **zs** : spMap (TrivSqZeroExt.fstHom R R R).toRingHom ≫ spMap (algebraMap R (DualNumber R)) = 𝟙 _

theorem **lift_ex** (w : Sp (DualNumber R) ⟶ t.G.left) (h : w ≫ t.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x ≫ g) : ∃ v : Sp (DualNumber R) ⟶ t'.G.left, v ≫ m.φ = w ∧ v ≫ t'.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x

theorem **tan_iff** (v : Sp (DualNumber R) ⟶ t'.G.left) (hv : v ≫ t'.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x) : IsTanAtG t' x v ↔ IsTanAtG t (x ≫ g) (v ≫ m.φ)

theorem **eig_iff** (v : Sp (DualNumber R) ⟶ t'.G.left) (hv : v ≫ t'.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x) (a : 𝒪 E) (c : R) : v ≫ (t'.act.act a).left = spMap (epsMul c) ≫ v ↔ (v ≫ m.φ) ≫ (t.act.act a).left = spMap (epsMul c) ≫ v ≫ m.φ

theorem **prod_left_congr** {X : Scheme.{0}} {G : Over S} [GrpObj G] {a b : X ⟶ S} (h : a = b) {n : ℕ} (u : Fin n → (Over.mk a ⟶ G)) (u' : Fin n → (Over.mk b ⟶ G)) (hu : ∀ i, (u i).left = (u' i).left) : (List.ofFn u).prod.left = (List.ofFn u').prod.left

theorem **comb_iff** {n : ℕ} (v : Fin n → (Sp (DualNumber R) ⟶ t'.G.left)) (hv : ∀ i, v i ≫ t'.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x) (cf : Fin n → R) (w : Sp (DualNumber R) ⟶ t'.G.left) (hw : w ≫ t'.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x) : IsCombAtG t' x v cf w ↔ IsCombAtG t (x ≫ g) (fun i => v i ≫ m.φ) cf (w ≫ m.φ)

theorem **frame_iff** {l : Level H} {c : Coeff l} (f : c.F →+* R) (u : Vecs Dw t' R) (hu : ∀ τ i, u τ i ≫ t'.G.hom = spMap (algebraMap R (DualNumber R)) ≫ x) : IsFrame t' f x u ↔ IsFrame (Dw

theorem **frame_of_frPt** (Y : Model Xo c β) {R : Type} [CommRing R] [Algebra c.F R] (y : FrPt Xo c R) (q : Sp R ⟶ Xo.XF c) (hq : q ≫ pullback.fst _ _ = y.x) : ∃ fr : Vecs Dw Y.t R, IsFrame Y.t (algebraMap c.F R) (q ≫ Y.j) fr

theorem **exists_chart** (Y : Model Xo c β) {Z : Scheme.{0}} (J : Xo.XF c ⟶ Z) [IsOpenImmersion J] (z : Xo.XF c) : ∃ (U : Z.affineOpens) (alg : c.F →+* Γ(Z, U.1)) (q : Sp Γ(Z, U.1) ⟶ Xo.XF c) (fr : Vecs Dw Y.t Γ(Z, U.1)), J z ∈ U.1 ∧ q ≫ J = U.2.fromSpec ∧ q ≫ pullback.snd _ _ = spMap alg ∧ IsFrame Y.t alg (q ≫ Y.j) fr

theorem **isOpenImmersion_generic** {F : Type} [Field F] (V : ValuationSubring F) [IsDiscreteValuationRing V] : IsOpenImmersion (spMap V.subtype)

theorem **compactSpace_XF** (Xo : Moduli Dw l) (c : Coeff l) : CompactSpace (Xo.XF c)

theorem **normal_of_iso** {X X' : Scheme.{0}} (f : X' ⟶ X) [IsIso f] (h : Geo.Tor.IsNormalScheme X) : Geo.Tor.IsNormalScheme X'

theorem **exists_base** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Y : LocalSystem Xo p W) (i₀ i : Idx H p) (hd : i₀.l.Dvd i.l) (hle : i₀.c.F ≤ i.c.F) : ∃ Z : Mdl Y i₀, Z.i = i ∧ IsIso Z.π ∧ ∀ σ, Z.ty σ = ∅

theorem **zeta_mem** {n m : ℕ} (hm : m ≠ 0) (h : n ∣ m) (F : IntermediateField ℚ ℂ) (hF : zeta m ∈ F) : zeta n ∈ F

theorem **exists_lvl** (hp : p ≠ 0) (i₀ : Idx H p) (b : Option ℕ) : ∃ i : Idx H p, IsLvl i₀ i b ∧ i₀.l.Dvd i.l ∧ i₀.c.F ≤ i.c.F

theorem **nonempty_bases** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Y : LocalSystem Xo p W) (i₀ : Idx H p) : Nonempty (Bases Y i₀)

theorem **frame_bc_univ** {l : Level H} (X₀ : Moduli Dw l) {c : Coeff l} {R T : Type} [CommRing R] [CommRing T] (f : c.F →+* R) (φ : R →+* T) (x : Sp R ⟶ X₀.X) (u : Vecs Dw X₀.univ.t R) (h : IsFrame X₀.univ.t f x u) : IsFrame X₀.univ.t (φ.comp f) (spMap φ ≫ x) (u.bc φ)

theorem **frame_bc** {l : Level H} {X₀ : Moduli Dw l} {c : Coeff l} {O : Type} [CommRing O] {β : O →+* c.F} (Y : Model X₀ c β) {R T : Type} [CommRing R] [CommRing T] (f : c.F →+* R) (φ : R →+* T) (q : Sp R ⟶ X₀.XF c) (u : Vecs Dw Y.t R) (h : IsFrame Y.t f (q ≫ Y.j) u) : IsFrame Y.t (φ.comp f) (spMap φ ≫ q ≫ Y.j) (u.bc φ)

theorem **comb_single** {S : Scheme.{0}} (t : SAB H S) {R : Type} [CommRing R] (x : Sp R ⟶ S) {n : ℕ} (v : Fin n → (Sp (DualNumber R) ⟶ t.G.left)) (hv : ∀ i, IsTanAtG t x (v i)) (i : Fin n) : IsCombAtG t x v (Pi.single i 1) (v i)

theorem **sameFlag_refl** (ty : (E →+* ℂ) → Finset ℕ) {S : Scheme.{0}} (t : SAB H S) {R : Type} [CommRing R] (x : Sp R ⟶ S) (u : Vecs Dw t R) (hu : ∀ τ i, IsTanAtG t x (u τ i)) : SameFlag ty t x u u

theorem **exists_flagAt** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) {k : Type} [Field k] (w : Sp k ⟶ Z.X) (hw : Set.range w ⊆ Z.opn) : ∃ u, Z.FlagAt w u

theorem **isMap_id** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) : IsMap Z Z (𝟙 _)

theorem **isDiag_self** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) : IsDiag Z Z Z

theorem **isNBlow_unit** (hp : p ≠ 0) (hW : (p : ℂ) ∈ W.nonunits) (Z : Mdl Y i₀) : IsNBlow Z (unitIdl Z) Z

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsOpenKerFinIdele

theorem **eq_one_of_pow_near** {z : ℂ} (h : ∀ n : ℕ, ‖z ^ n - 1‖ < 1 / 2) : z = 1

theorem **isOpen_ker_finIdele** (E : Type) [Field E] [NumberField E] (ψ : (FiniteAdeleRing (𝒪 E) E)ˣ →* ℂˣ) (hψ : Continuous ψ) : IsOpen {u | ψ u = 1}

theorem **expU_add** (z z' : ι → ℂ) : expU (z + z') = expU z * expU z'

theorem **expU_zero** : expU (0 : ι → ℂ) = 1

theorem **unit_ne_zero** (u : (ι → ℂ)ˣ) (i : ι) : (u : ι → ℂ) i ≠ 0

theorem **expU_log** (u : (ι → ℂ)ˣ) : expU (fun i => Complex.log ((u : ι → ℂ) i)) = u

theorem **continuous_expU** : Continuous (expU : (ι → ℂ) → (ι → ℂ)ˣ)

theorem **exists_lin** (ψ : (ι → ℂ)ˣ →* ℂˣ) (hψ : Continuous ψ) : ∃ L : (ι → ℂ) →L[ℝ] ℂ, ∀ z, ((ψ (expU z) : ℂˣ) : ℂ) = Complex.exp (L z)

theorem **extZero_unit** (ψ : (ι → ℂ)ˣ →* ℂˣ) (u : (ι → ℂ)ˣ) : extZero ψ u = (ψ u : ℂˣ)

theorem **extZero_eventually** {ψ : (ι → ℂ)ˣ →* ℂˣ} {L : (ι → ℂ) →L[ℝ] ℂ} (hL : ∀ z, ((ψ (expU z) : ℂˣ) : ℂ) = Complex.exp (L z)) (u : (ι → ℂ)ˣ) : extZero ψ =ᶠ[𝒩 (u : ι → ℂ)] fun x => (ψ u : ℂˣ) * Complex.exp (L fun i => Complex.log (x i / (u : ι → ℂ) i))

theorem **contDiffAt_extZero** (ψ : (ι → ℂ)ˣ →* ℂˣ) (hψ : Continuous ψ) (u : (ι → ℂ)ˣ) : ContDiffAt ℝ ∞ (extZero ψ) u

theorem **fderiv_extZero** {ψ : (ι → ℂ)ˣ →* ℂˣ} {L : (ι → ℂ) →L[ℝ] ℂ} (hψ : Continuous ψ) (hL : ∀ z, ((ψ (expU z) : ℂˣ) : ℂ) = Complex.exp (L z)) : fderiv ℝ (extZero ψ) 1 = L

theorem **exists_bound** (ψ : (ι → ℂ)ˣ →* ℂˣ) (hψ : Continuous ψ) : ∃ (C : ℝ) (N : ℕ), 0 ≤ C ∧ ∀ (u : (ι → ℂ)ˣ) (B : ℝ), 1 ≤ B → ‖(u : ι → ℂ)‖ ≤ B → ‖((u⁻¹ : (ι → ℂ)ˣ) : ι → ℂ)‖ ≤ B → ‖((ψ u : ℂˣ) : ℂ)‖ ≤ C * B ^ N

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsPrincipalMapDifferent

theorem **exists_frameAt** (q : LatChart Z L) {k : Type} [Field k] {x : Sp k ⟶ Z.X} (h : ⊤ ≤ x ⁻¹ᵁ q.U) (ho : Set.range x ⊆ Z.opn) : ∃ u, q.FrameAt x u

theorem **isTr_tr** {q q' : LatChart Z L} {e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ} (h : ∃ a : Γ(Z.X, q.U ⊓ q'.U)ˣ, q.IsTr q' e a.val) : q.IsTr q' e (q.tr q' e).val

theorem **span_diffGen** {l : Level H} (c : Coeff l) (W : ValuationSubring ℂ) (σ : E →+* ℂ) : Ideal.span {diffGen c W σ} = (differentIdeal ℤ (𝒪 E)).map (embV c W σ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsQuasicoherentDualOm

theorem **torsUnit_ι** (G : Over S) [GrpObj G] (N : ℕ) : torsUnit G N ≫ torsion.ι G N = η[G]

theorem **coe_embV** (σ : E →+* ℂ) (a : 𝒪 E) : (((embV c W σ a : c.V W) : c.F) : ℂ) = σ a

theorem **subtype_comp_embV** (σ : E →+* ℂ) : (c.V W).subtype.comp (embV c W σ) = embK c σ

theorem **embV_injective** : Function.Injective (embV c W)

theorem **isSplitting_embK** : IsSplitting (embK c)

theorem **splitMap_embV_injective** : Function.Injective (EigenSplit.splitMap (embV c W))

theorem **exists_isFrame_om** (x : Z.X) : ∃ (U : Z.X.Opens) (n : ℕ) (s : Fin n → Γ(Z.om, U)), x ∈ U ∧ VB.IsFrame s

theorem **mem_imHT** (t : Dw.La) : h.1 t ∈ (imHT h).sec ⊤

theorem **imHT_le** {U : Z.X.Opens} (hU : IsAffineOpen U) : (imHT h).sec U = Submodule.span _ (Set.range fun t => resM _ le_top (h.1 t))

theorem **mem_omP_iff** {U : Z.X.Opens} {s : Γ(Z.om, U)} : s ∈ (omP h).sec U ↔ (red Z n).app U s ∈ (imHT h).sec U

theorem **pow_smul_mem_omP** {U : Z.X.Opens} (s : Γ(Z.om, U)) : p ^ n • s ∈ (omP h).sec U

theorem **exists_isLift** (t : Dw.La) {U : Z.X.Opens} (hU : IsAffineOpen U) : ∃ s : Γ((omP h).toMod, U), IsLift h t s

theorem **isGrpPullback_cast** {i i' : Idx H p} (e : i = i') : IsGrpPullback (Y i).t.G (Y i').t.G (eqToHom (congrArg (fun i => (Y i).X) e)) (eqToHom (congrArg (fun i => (Y i).t.G.left) e)) ∧ ∀ a, ((Y i).t.act.act a).left ≫ eqToHom (congrArg (fun i => (Y i).t.G.left) e) = eqToHom (congrArg (fun i => (Y i).t.G.left) e) ≫ ((Y i').t.act.act a).left

theorem **toBase_π** : toBase Z B e ≫ B.π = Z.π ≫ eqToHom (congrArg (fun i => (Y i).X) e)

theorem **toBaseG_φ** : toBaseG Z B e ≫ B.tMap.φ = Z.tMap.φ ≫ eqToHom (congrArg (fun i => (Y i).t.G.left) e)

theorem **isGrpPullback_toBaseG** : IsGrpPullback Z.t.G B.t.G (toBase Z B e) (toBaseG Z B e)

theorem **act_toBaseG** (a : 𝒪 E) : (Z.t.act.act a).left ≫ toBaseG Z B e = toBaseG Z B e ≫ (B.t.act.act a).left

theorem **isPrincipalIdealRing_V** : IsPrincipalIdealRing (Z.i.c.V ct.W)

theorem **exists_eps** (σ : D.host.E →+* ℂ) : ∃ ε, ann (embV Z.i.c ct.W σ) = Submodule.span _ {ε} ∧ EigenSplit.splitMap (embV Z.i.c ct.W) ε σ ≠ 0

theorem **ann_eq_span_eps** (σ : D.host.E →+* ℂ) : ann (embV Z.i.c ct.W σ) = Submodule.span _ {eps ct Z σ}

theorem **eps_mem** (σ : D.host.E →+* ℂ) : eps ct Z σ ∈ ann (embV Z.i.c ct.W σ)

theorem **dif_ne_zero** (σ : D.host.E →+* ℂ) : dif ct Z σ ≠ 0

theorem **map_eps** (σ : D.host.E →+* ℂ) {R : Type} [CommRing R] (f : Z.i.c.V ct.W →+* R) (hf : IsSplitting fun τ => f.comp (embV Z.i.c ct.W τ)) : Algebra.TensorProduct.map f.toIntAlgHom (AlgHom.id ℤ _) (eps ct Z σ) = f (dif ct Z σ) • EigenSplit.idem hf σ

theorem **isFiniteQC_omP** : IsFiniteQC (omP h).toMod

theorem **htIdl_val** (H : ∀ x ∈ (htIdeal ct h).support, x ∉ Z.opn) : (htIdl ct h).1 = htIdeal ct h

theorem **isVB_tSheaf** : IsVB (tSheaf ct Z σ) (Module.finrank (Z.i.c.V ct.W) (tSub Z σ))

theorem **isSnd_sndMap** {m : ℕ} (H : ∃ g, IsSnd ct h σ (m

theorem **isFst_fstMap** {m : ℕ} (H : ∃ f, IsFst ct h σ (m

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsRationalUnitaryDeterminantMapRationalUnitary

theorem **congQ_subset_congG** : congQ H SQ ⊆ congG H

theorem **mem_KDq_of_congQ** {a : Ideles H.E} (h : a ∈ congQ H SQ) : a ∈ KDq H SQ

theorem **locF_mem_KDq** (v : HeightOneSpectrum (𝒪 H.F)) (k : (H.F)ˣ) : locF H.F v k ∈ KDq H SQ

theorem **KDq_le_iff** {B : Subgroup (Ideles H.E)} : KDq H SQ ≤ B ↔ (∀ a ∈ congQ H SQ, a ∈ B) ∧ ∀ v k, locF H.F v k ∈ B

theorem **KDq_le_KD** : KDq H SQ ≤ KD H

theorem **KDq_anti** {SQ SQ' : Finset (HeightOneSpectrum (𝒪 H.E))} (h : SQ ⊆ SQ') : KDq H SQ' ≤ KDq H SQ

theorem **congQ_empty** : congQ H ∅ = congG H

theorem **KDq_empty** : KDq H ∅ = KD H

theorem **KD_le_sup** : KD H ≤ KDq H SQ ⊔ unitIdl H.E

theorem **KDq_sup_unitIdl** : KDq H SQ ⊔ unitIdl H.E = KD H ⊔ unitIdl H.E

theorem **finite_level** : {v : HeightOneSpectrum (𝒪 H.E) | (∃ k, OverP H (H.g k) v) ∨ AtQ H SQ v}.Finite

theorem **mem_congQ_of_eq_one** {a : Ideles H.E} (hu : a ∈ unitIdl H.E) (h : ∀ v, (∃ k, OverP H (H.g k) v) ∨ AtQ H SQ v → (a : AdeleRing (𝒪 H.E) H.E).2 v = 1) : a ∈ congQ H SQ

theorem **infIdele_mem_KDq** (w : InfinitePlace H.E) (y : (w.Completion)ˣ) : infIdele H.E w y ∈ KDq H SQ

theorem **isOpen_KDq** : IsOpen (KDq H SQ : Set (Ideles H.E))

theorem **compAt_localIdele_self** (v : HeightOneSpectrum (𝒪 L)) (x : (v.adicCompletion L)ˣ) : compAt v (localIdele L v x) = x

theorem **compAt_localIdele_ne** {v w : HeightOneSpectrum (𝒪 L)} (hw : w ≠ v) (x : (v.adicCompletion L)ˣ) : compAt w (localIdele L v x) = 1

theorem **compAt_locF** (w : HeightOneSpectrum (𝒪 L)) (v : HeightOneSpectrum (𝒪 F)) (k : Fˣ) : compAt w (locF F v k) = if w.under (𝒪 F) = v then compAt w (diagUnits L (upF F k)) else 1

theorem **exists_diag_upF_eq** (k : Fˣ) {T : Set (HeightOneSpectrum (𝒪 L))} (hT : T.Finite) : ∃ (S : Finset (HeightOneSpectrum (𝒪 F))) (u : Ideles L), u ∈ unitIdl L ∧ (∀ w ∈ T, compAt w u = 1) ∧ diagUnits L (upF F k) = (∏ v ∈ S, locF F v k) * u

theorem **diag_upF_mem_KDq** (k : (H.F)ˣ) : diagUnits H.E (upF H.F k) ∈ KDq H SQ

@[simp] theorem **toDtT_mk** (a : Ideles H.E) : toDtT H SQ (QuotientGroup.mk a) = QuotientGroup.mk a

theorem **toDtT_surjective** : Function.Surjective (toDtT H SQ)

theorem **mk_mem_UOQ** {a : Ideles H.E} : (QuotientGroup.mk a : DtQ H SQ) ∈ UOQ H SQ ↔ a ∈ KDq H SQ ⊔ unitIdl H.E

theorem **comap_UOT** : (UOT H).comap (toDtT H SQ) = UOQ H SQ

@[simp] theorem **toDtTMod_mk** (x : DtQ H SQ) : toDtTMod H SQ (QuotientGroup.mk x) = QuotientGroup.mk (toDtT H SQ x)

theorem **toDtTMod_bijective** : Function.Bijective (toDtTMod H SQ)

theorem **toDtTMod_relDetQ** (N : ℕ) (M M' : Submodule (𝒪 H.E) (Fin N → H.E)) : toDtTMod H SQ (relDetQ H SQ N M M') = relDetT H N M M'

theorem **relDetQ_self** (N : ℕ) (M : Submodule (𝒪 H.E) (Fin N → H.E)) : relDetQ H SQ N M M = 1

theorem **tildeQ_mk** {χ : HChar H.E} (h : KDq H SQ ≤ (χ.1.comp (finPartI H.E)).ker) (a : Ideles H.E) : tildeQ H SQ χ (QuotientGroup.mk a) = χ.1 (finPartI H.E a)

theorem **tildeQ_eq_comp** {χ : HChar H.E} (h : KD H ≤ (χ.1.comp (finPartI H.E)).ker) : tildeQ H SQ χ = (tildeD H χ).comp (toDtT H SQ)

@[simp] theorem **tildeQ_one** : tildeQ H SQ 1 = 1

theorem **tildeQ_mul** {χ χ' : HChar H.E} (h : KDq H SQ ≤ (χ.1.comp (finPartI H.E)).ker) (h' : KDq H SQ ≤ (χ'.1.comp (finPartI H.E)).ker) : tildeQ H SQ (χ * χ') = tildeQ H SQ χ * tildeQ H SQ χ'

theorem **norm_tildeQ** {χ : HChar H.E} (hu : χ.IsUnitary) (x : DtQ H SQ) : ‖((tildeQ H SQ χ x : ℂˣ) : ℂ)‖ = 1

theorem **muTilQ_succ** (mu : Fin R.m → HChar H.E) (j : Fin R.m) : muTilQ H SQ mu (j.1 + 1) = tildeQ H SQ (mu j)

theorem **exists_eq_div_conj** : letI

theorem **mk_diag_eq** {z : H.E} {k k' : (H.E)ˣ} (h : z * H.conj (k : H.E) = k) (h' : z * H.conj (k' : H.E) = k') : (QuotientGroup.mk (diagUnits H.E k) : DtQ H SQ) = QuotientGroup.mk (diagUnits H.E k')

theorem **exists_isUnitaryPointsMapFinerDeterminants** : ∃ f, IsUnitaryPointsMapFinerDeterminants H SQ f

theorem **isUnitaryPointsMap_unitaryRationalPointsFiner** : IsUnitaryPointsMapFinerDeterminants H SQ (e1Q H SQ)

theorem **isRationalUnitaryDeterminantMap_rationalUnitary** : IsRationalUnitaryDeterminantMap H (e1T H)

theorem **toDtT_e1Q** : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsRationalUnitaryDeterminantMapRationalUnitaryTo

theorem **mem_unitIdl** {a : Ideles L} : a ∈ unitIdl L ↔ ∀ v, Valued.v ((a : AdeleRing (𝒪 L) L).2 v) = 1

theorem **finPartI_fst** (a : Ideles L) : (finPartI L a : AdeleRing (𝒪 L) L).1 = 1

theorem **finPartI_snd** (a : Ideles L) : (finPartI L a : AdeleRing (𝒪 L) L).2 = (a : AdeleRing (𝒪 L) L).2

theorem **finPartI_idem** (a : Ideles L) : finPartI L (finPartI L a) = finPartI L a

theorem **finPartI_eq_self** {a : Ideles L} (h : ∀ w : InfinitePlace L, (a : AdeleRing (𝒪 L) L).1 w = 1) : finPartI L a = a

theorem **finPartI_mem_unitIdl** {a : Ideles L} : finPartI L a ∈ unitIdl L ↔ a ∈ unitIdl L

theorem **conjPl_refl_iff** (v v' : HeightOneSpectrum (𝒪 L)) : ConjPl (RingEquiv.refl L) v v' ↔ v = v'

theorem **isConjI_refl** : IsConjI (RingEquiv.refl L) (MonoidHom.id (Ideles L)) where

theorem **tameLoc_one** (v : HeightOneSpectrum (𝒪 L)) : TameLoc v 1

theorem **tameLoc_loc** {χ : HChar L} (hu : χ.IsUnitary) {v : HeightOneSpectrum (𝒪 L)} (ht : χ.TameAt v) : TameLoc v (χ.loc v)

theorem **det_mem_detSub** {N : ℕ} {M : Submodule (𝒪 L) (Fin N → L)} (x : Fin N → Fin N → L) (hx : ∀ i, x i ∈ M) : (Matrix.of x).det ∈ detSub M

theorem **detSub_mono** {N : ℕ} {M M' : Submodule (𝒪 L) (Fin N → L)} (h : M ≤ M') : detSub M ≤ detSub M'

theorem **ordF_unif_self** (w : HeightOneSpectrum (𝒪 L)) : ordF w (unif w) = 1

theorem **ordSub_span_singleton** (w : HeightOneSpectrum (𝒪 L)) (u : Lˣ) : ordSub w (Submodule.span (𝒪 L) {(u : L)}) = ordF w u

theorem **negTwo_roots** (x : NegTwoField) (h : ∃ m : ℕ, 0 < m ∧ x ^ m = 1) : x = 1 ∨ x = -1

theorem **cmData_negTwo** : CMData ℚ negTwoConj (fun w : InfinitePlace NegTwoField => w.embedding) where

theorem **mem_KD_of_congG** {a : Ideles H.E} (h : a ∈ congG H) : a ∈ KD H

theorem **locF_mem_KD** (v : HeightOneSpectrum (𝒪 H.F)) (k : (H.F)ˣ) : locF H.F v k ∈ KD H

theorem **tildeD_mk** {χ : HChar H.E} (h : KD H ≤ (χ.1.comp (finPartI H.E)).ker) (a : Ideles H.E) : tildeD H χ (QuotientGroup.mk a) = χ.1 (finPartI H.E a)

theorem **muTilT_succ** (mu : Fin R.m → HChar H.E) (j : Fin R.m) : muTilT H mu (j.1 + 1) = tildeD H (mu j)

theorem **mem_UOT** {x : DtT H} : x ∈ UOT H ↔ ∃ a : Ideles H.E, (∀ v, Valued.v ((a : AdeleRing (𝒪 H.E) H.E).2 v) = 1) ∧ QuotientGroup.mk a = x

theorem **isRationalUnitaryDeterminantMap_rationalUnitaryToDeterminants** (h : ∃ f, IsRationalUnitaryDeterminantMap H f) : IsRationalUnitaryDeterminantMap H (e1T H)

theorem **relDetT_self** (N : ℕ) (M : Submodule (𝒪 H.E) (Fin N → H.E)) : relDetT H N M M = 1

theorem **isTwist_one** (S : Set ℕ) : IsTwist H S 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsSMulRegularNatCast

theorem **eq_zero_of_flat** {R T : Type} [CommRing R] [IsDomain R] [CharZero R] [CommRing T] (φ : R →+* T) (hφ : φ.Flat) {c : ℕ} (hc : c ≠ 0) {a : T} (h : (c : T) * a = 0) : a = 0

theorem **Mdl.isSMulRegular_natCast** (B : Mdl Y i₀) {U : B.X.Opens} (hU : IsAffineOpen U) {c : ℕ} (hc : c ≠ 0) {a : Γ(B.X, U)} (h : (c : Γ(B.X, U)) * a = 0) : a = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsSemisimpleOfISupEigenspace

theorem **isSemisimple_map** [FiniteDimensional ℂ V] (φ : End ℂ V →ₐ[ℂ] End ℂ V') {f : End ℂ V} (hf : f.IsSemisimple) : (φ f).IsSemisimple

theorem **isSemisimple_of_iSup_eigenspace** [FiniteDimensional ℂ V] {f : End ℂ V} (hf : ⨆ α, f.eigenspace α = ⊤) : f.IsSemisimple

theorem **isSemisimple_prodMap** [FiniteDimensional ℂ V] [FiniteDimensional ℂ V'] {f : End ℂ V} {g : End ℂ V'} (hf : f.IsSemisimple) (hg : g.IsSemisimple) : End.IsSemisimple (f.prodMap g : End ℂ (V × V'))

theorem **isSemisimple_tensorMap** [FiniteDimensional ℂ V] [FiniteDimensional ℂ V'] {f : End ℂ V} {g : End ℂ V'} (hf : f.IsSemisimple) (hg : g.IsSemisimple) : End.IsSemisimple (TensorProduct.map f g : End ℂ (V ⊗[ℂ] V'))

theorem **aeval_dualMap** (f : End ℂ V) (p : ℂ[X]) : aeval (f.dualMap : End ℂ (Dual ℂ V)) p = (aeval f p : End ℂ V).dualMap

theorem **isSemisimple_dualMap** [FiniteDimensional ℂ V] {f : End ℂ V} (hf : f.IsSemisimple) : End.IsSemisimple (f.dualMap : End ℂ (Dual ℂ V))

@[simp] theorem **diagEnd_apply** {ι : Type*} (c v : ι → ℂ) (i : ι) : diagEnd c v i = c i * v i

theorem **isSemisimple_diagEnd** {ι : Type*} [Finite ι] (c : ι → ℂ) : (diagEnd c).IsSemisimple

theorem **charpoly_diagEnd** {ι : Type*} [Fintype ι] (c : ι → ℂ) : (diagEnd c).charpoly = ∏ i, (X - C (c i))

theorem **map_absSpace_le** {f : End ℂ V} {g : End ℂ V'} (e : V →ₗ[ℂ] V') (he : ∀ v, e (f v) = g (e v)) (r : ℝ) : (absSpace f r).map e ≤ absSpace g r

theorem **mem_absSpace_iff** {f : End ℂ V} {g : End ℂ V'} (e : V ≃ₗ[ℂ] V') (he : ∀ v, e (f v) = g (e v)) (r : ℝ) (v : V) : e v ∈ absSpace g r ↔ v ∈ absSpace f r

theorem **IsPurePair.of_equiv** {q : ℝ} {f N : End ℂ V} {g M : End ℂ V'} (e : V ≃ₗ[ℂ] V') (he : ∀ v, e (f v) = g (e v)) (hN : ∀ v, e (N v) = M (e v)) (hp : IsPurePair q f N) : IsPurePair q g M

theorem **mem_absSpace_diagEnd** {ι : Type*} [Finite ι] (c : ι → ℂ) (r : ℝ) (v : ι → ℂ) : v ∈ absSpace (diagEnd c) r ↔ ∀ i, ‖c i‖ ≠ r → v i = 0

theorem **shiftEnd_apply** {n : ℕ} (v : Fin n → ℂ) (i : Fin n) : shiftEnd n v i = if 1 ≤ (i : ℕ) then v ⟨i - 1, by omega⟩ else 0

theorem **shiftEnd_pow_apply** {n : ℕ} (k : ℕ) (v : Fin n → ℂ) (i : Fin n) : (shiftEnd n ^ k) v i = if k ≤ (i : ℕ) then v ⟨i - k, by omega⟩ else 0

theorem **isNilpotent_shiftEnd** (n : ℕ) : IsNilpotent (shiftEnd n)

theorem **shiftEnd_ne_zero** {n : ℕ} (hn : 2 ≤ n) : shiftEnd n ≠ 0

theorem **isPurePair_diagEnd_shiftEnd** {q : ℝ} (hq : 1 < q) {n : ℕ} (c : Fin n → ℂ) (hc : ∀ i : Fin n, ‖c i‖ = q ^ (((i : ℕ) : ℝ) - ((n : ℝ) - 1) / 2)) : IsPurePair q (diagEnd c) (shiftEnd n)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsTwIffCanonEq

theorem **canon_isTw** : I.canon.IsTw

theorem **IsTw.cst_eq** (h : I.IsTw) (a : I.ι) : I.cst a = twCst I.Ch I.sel (I.tup a) I.glob

theorem **IsTw.idx_eq** (h : I.IsTw) (a : I.ι) : I.idx a = twIdx I.Ch I.sel (I.tup a) (twCst I.Ch I.sel (I.tup a) I.glob) (B.R.m - 2)

theorem **IsTw.idxFull_eq** (h : I.IsTw) (a : I.ι) : I.idxFull a = twIdxTop I.Ch I.sel (I.tup a) (I.cst a) (B.R.m - 2) (twLt I.Ch) (twTt I.sel (I.tup a))

theorem **isTw_iff_canon_eq** : I.IsTw ↔ I.canon = I

theorem **IsTw.deep** (h : I.IsTw) : I.Deep

theorem **qOf_mod_four** (B : Base) (X : PreX B.toPre) : ∀ w ∈ (B.primes X).vGs, qOf w % 4 = 1

theorem **val_of_ok** (h : I.Ok) (j : Fin B.R.m) (w : HeightOneSpectrum (𝒪 B.fields.E)) (hw : w ∈ (B.primes X).vGs) : ∃ y : B.fields.K, I.glob.σ₀ y = ((((I.glob.mu0 j).locK w (ciQU w)) : ℂˣ) : ℂ)⁻¹

theorem **Q_notMem_of_ok** (h : I.Ok) (a : I.ι) : ∀ P ∈ I.Q a, P ∉ (B.primes X).vGs

theorem **degOne_of_ok** (h : I.Ok) (a : I.ι) (P : rowSG (B.primes X) I.sel (I.tup a)) : TwDegOne B.fields P.1

theorem **sqrt_of_ok** (h : I.Ok) (a : I.ι) (P : rowSG (B.primes X) I.sel (I.tup a)) : ∃ z : B.fields.K, z ^ 2 = (qOf P.1 : B.fields.K)

theorem **root_of_ok** (h : I.Ok) (hD : I.Deep) (a : I.ι) (P : rowSG (B.primes X) I.sel (I.tup a)) (hP : P.1 ∉ (B.primes X).vGs) : ∃ z : B.fields.K, I.glob.σ₀ z = Complex.exp (2 * Real.pi * Complex.I / ((qOf P.1 ^ (2 * B.R.m) : ℕ) : ℂ))

theorem **twIdx_ok_of_ok** (h : I.Ok) (hD : I.Deep) (a : I.ι) : (twIdx I.Ch I.sel (I.tup a) (twCst I.Ch I.sel (I.tup a) I.glob) (B.R.m - 2)).Ok I.Ch I.sel (I.tup a) (twCst I.Ch I.sel (I.tup a) I.glob) I.glob (B.R.m - 2)

theorem **canon_ok** (h : I.Ok) (hD : I.Deep) : I.canon.Ok

theorem **IsTw.chF_ok** (ht : I.IsTw) (h : I.Ok) (W : Wts B.fields (B.primes X).Word B.R) (a : I.ι) (u : (B.primes X).Word) (j : ℕ) : (chF W I.sel (I.tup a) (I.idx a) u j).Ok

theorem **IsTw.chF_full_ok** (ht : I.IsTw) (h : I.Ok) (W : Wts B.fields (B.primes X).Word B.R) (a : I.ι) (u : (B.primes X).Word) (j : ℕ) : (chF W I.sel (I.tup a) (I.idxFull a) u j).Ok

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsUnitNatCastOfLeGenO

theorem **Mdl.isUnit_natCast_of_le_genO** (B : Mdl Y i₀) {V : B.X.Opens} (hV : V ≤ B.genO) : IsUnit ((p : ℕ) : Γ(B.X, V))

theorem **HTm.omP_sec_eq_top** {Z : Mdl Y i₀} {n : ℕ} (h : HTm.HTOn Z n) {V : Z.X.Opens} (hV : V ≤ Z.genO) : (HTm.omP h).sec V = ⊤

theorem **Step.admitted.of_isMap** {t : Step Y i₀} (h : t.admitted P Z) {f : Z'.X ⟶ Z.X} (hf : IsMap Z' Z f) : t.admitted P Z'

theorem **mem_union** {s s' : FlagFam P Z} {t : Step Y i₀} : t ∈ (s.union s').1 ↔ t ∈ s.1 ∨ t ∈ s'.1

theorem **isFlagCl_flgTrue** (Z : ct.Mdl) (s : (flgTrue ct).FlagData Z) : IsFlagCl Z s.1 ((flgTrue ct).flagCl Z s)

theorem **isHTLatOf_iff** {L : Lat Z} : IsHTLatOf (htmTrue ct) Z L ↔ (∃ n, IsHTBlowup ct Z (ct.Xn n)) ∧ ∀ y : ct.Pt Z, latAt ct y (omSub ct Z) = latAt ct y L.N

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsUnitaryOfCharIff

theorem **map_scalar** {R S : Type} [CommRing R] [CommRing S] (p : R →+* S) (u : Rˣ) : GeneralLinearGroup.map p (GeneralLinearGroup.scalar (Fin a) u) = GeneralLinearGroup.scalar (Fin a) (Units.map p.toMonoidHom u)

theorem **cornerGL_scalar** {R S : Type} [CommRing R] [CommRing S] (s : R →ₙ+* S) (u : Rˣ) : cornerGL a s (GeneralLinearGroup.scalar (Fin a) u) = GeneralLinearGroup.scalar (Fin a) (Units.map (corner s) u)

theorem **scalar_comm** {R : Type} [CommRing R] (u : Rˣ) (g : GL (Fin a) R) : GeneralLinearGroup.scalar (Fin a) u * g = g * GeneralLinearGroup.scalar (Fin a) u

theorem **infId_mul_finId** (z : Ideles E) : infId E (Units.map (pInf E).toMonoidHom z) * finId E (Units.map (pFin E).toMonoidHom z) = z

theorem **hasCentral_iff** : π.HasCentral ω ↔ π.HasCentChar ω

theorem **Iso.hasCentral** (h : Iso π π') (hω : π.HasCentral ω) : π'.HasCentral ω

theorem **HasCentral.unique** (h : π.HasCentral ω) (h' : π.HasCentral ω') : ω = ω'

theorem **HasCentral.twist** (h : π.HasCentral ω) (χ : HChar E) : (π.twist χ).HasCentral (ω * χ ^ a)

theorem **HasCentral.apply** (h : π.HasCentral ω) {φ : Sm E a} (hφ : φ ∈ π.V) (z : Ideles E) (g : GA E a) : φ.1 (GeneralLinearGroup.scalar (Fin a) z * g) = ((ω.1 z : ℂˣ) : ℂ) * φ.1 g

theorem **Iso.isUnitary** (h : Iso π π') (hu : π.IsUnitary) : π'.IsUnitary

theorem **HasCentral.isUnitary_iff** (h : π.HasCentral ω) : π.IsUnitary ↔ ω.IsUnitary

theorem **IsUnitary.norm_eq** (h : π.IsUnitary) {φ : Sm E a} (hφ : φ ∈ π.V) (z : Ideles E) (g : GA E a) : ‖φ.1 (GeneralLinearGroup.scalar (Fin a) z * g)‖ = ‖φ.1 g‖

theorem **isUnitary_one** : (1 : HChar E).IsUnitary

theorem **isUnitary_pow** {χ : HChar E} (h : χ.IsUnitary) (n : ℕ) : (χ ^ n).IsUnitary

theorem **isUnitary_of_pow** {χ : HChar E} {n : ℕ} (hn : n ≠ 0) (h : (χ ^ n).IsUnitary) : χ.IsUnitary

@[simp] theorem **isUnitary_mk** (π : CuspSub E a) : (mk π).IsUnitary ↔ π.IsUnitary

theorem **HasCentChar.unique** (h : π.HasCentChar ω) (h' : π.HasCentChar ω') : ω = ω'

theorem **HasCentChar.isUnitary_iff** (h : π.HasCentChar ω) : π.IsUnitary ↔ ω.IsUnitary

@[simp] theorem **isUnitary_ofChar_iff** (χ : HChar E) : (ofChar χ).IsUnitary ↔ χ.IsUnitary

theorem **IsUnitary.twist** (h : π.IsUnitary) {χ : HChar E} (hχ : χ.IsUnitary) : (π.twist χ).IsUnitary

@[simp] theorem **ofChar_toChar** (σ : CG E 1) : ofChar (toChar σ) = σ

theorem **isUnitary_toChar_iff** (σ : CG E 1) : (toChar σ).IsUnitary ↔ σ.IsUnitary

theorem **IsUnitary.tw** (h : π.IsUnitary) {σ : CG E 1} (hσ : σ.IsUnitary) : (π.tw σ).IsUnitary

theorem **IsUnitary.not_twist_absPow** (ha : 1 ≤ a) (h : π.IsUnitary) {ν : HChar E} {s : ℝ} (hν : ν.IsAbsPow s) (hs : s ≠ 0) : ¬ (π.twist ν).IsUnitary

@[simp] theorem **twist_val** (χ : {χ : HChar E // χ.IsUnitary}) : (π.twist χ).1 = π.1.twist χ.1

@[simp] theorem **tw_val** (σ : CGu E 1) : (π.tw σ).1 = π.1.tw σ.1

theorem **hasCentChar_centChar** : π.1.HasCentChar π.centChar.1

theorem **centChar_twist** (χ : {χ : HChar E // χ.IsUnitary}) : (π.twist χ).centChar.1 = π.centChar.1 * χ.1 ^ a

@[simp] theorem **equivChar_val** (σ : CGu E 1) : (equivChar E σ).1 = σ.1.toChar

@[simp] theorem **equivChar_symm_val** (χ : {χ : HChar E // χ.IsUnitary}) : ((equivChar E).symm χ).1 = CG.ofChar χ.1

theorem **ofChar_equivChar** (σ : CGu E 1) : CG.ofChar (equivChar E σ).1 = σ.1

theorem **equivChar_tw** (σ σ' : CGu E 1) : (equivChar E (σ.tw σ')).1 = (equivChar E σ).1 * (equivChar E σ').1

theorem **tw_tw** (σ σ' : CGu E 1) : (π.tw σ).tw σ' = π.tw (σ.tw σ')

theorem **exists_not_isUnitary** : ∃ σ : CG E 1, ¬ σ.IsUnitary

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.IsWtOfParUnique

theorem **matEq_refl** {A : Type} [CommRing A] {a : ℕ} (x : GL (Fin a) A) : MatEq x x

theorem **eq_of_matEq** {A : Type} [CommRing A] {a : ℕ} {x y : GL (Fin a) A} (h : MatEq x y) : x = y

theorem **crd_li** (lam : X.Tau → DomWeight n) : LinearIndependent ℂ fun i => crd σ lam (bVV X.K lam i)

theorem **expand** {W : Type} [AddCommGroup W] [Module ℂ W] {lam : X.Tau → DomWeight n} (c : VV X.K lam →ₛₗ[σ] W) (v : VV X.K lam) : c v = ∑ j, σ ((bVV X.K lam).repr v j) • c (bVV X.K lam j)

theorem **crd_act** (lam : X.Tau → DomWeight n) (f : Module.End X.K (VV X.K lam)) (v : VV X.K lam) : crd σ lam (f v) = actC σ lam f (crd σ lam v)

theorem **inj_of_li** {W : Type} [AddCommGroup W] [Module ℂ W] {lam : X.Tau → DomWeight n} (c : VV X.K lam →ₛₗ[σ] W) (h : LinearIndependent ℂ fun i => c (bVV X.K lam i)) (v : VV X.K lam) (hv : c v = 0) : v = 0

theorem **stdSide_model** (hc : CMData X.F X.E) {ι : Type} (lam : X.Tau → DomWeight n) (rho : GL (Fin n) X.E →* Module.End X.K (VV X.K lam)) (SG : ι → Finset (Pl X.E)) (Fm : ι → Type) (val : ∀ t, Fm t → DecoSet X.E n ∅ (SG t) → VV X.K lam) : StdSide (a

theorem **ad_ne_zero** {SG : ι → Finset (Pl X.E)} {Fm : ι → Type} {val : ∀ t, Fm t → DecoSet X.E n ∅ (SG t) → VV X.K lam} {ad : ∀ t, Fm t → H → W} (s : StdSide σ mat rat ρ lam rho SG Fm val ad) (t : ι) (φ : Fm t) (hφ : ∀ d, val t φ d ≠ 0) (h : H) : ad t φ h ≠ 0

theorem **ad_eq** {S : Finset (Pl X.E)} {Fm : ι → Type} {val : ∀ t, Fm t → DecoSet X.E n ∅ S → VV X.K lam} {ad : ∀ t, Fm t → H → W} (s : StdSide σ mat rat ρ lam rho (fun _ => S) Fm val ad) (t t' : ι) (φ : Fm t) (φ' : Fm t') (e : val t φ = val t' φ') : ad t φ = ad t' φ'

theorem **one_map** {SG : ι → Finset (Pl X.E)} {Fm : ι → Type} {val : ∀ t, Fm t → DecoSet X.E n ∅ (SG t) → VV X.K lam} {ad ad' : ∀ t, Fm t → H → W} (s : StdSide σ mat rat ρ lam rho SG Fm val ad) (s' : StdSide σ mat rat ρ lam rho SG Fm val ad') : ∃ T : W →ₗ[ℂ] W, ∀ t φ h, ad' t φ h = T (ad t φ h)

theorem **not_sign** [Nonempty H] {S : Finset (Pl X.E)} (v : VV X.K lam) (hv : v ≠ 0) (f : H → W) : ¬ StdSide σ mat rat ρ lam rho (fun _ : Bool => S) (fun _ => Unit) (fun _ _ _ => v) fun b _ => cond b f (-f)

theorem **transport** {K E ι' : Type} [Field K] [NumberField K] [Field E] [NumberField E] [Fintype ι'] (σ : K →+* ℂ) {a n : ℕ} (e : a = n) (lam : ι' → DomWeight a) (lam' : ι' → DomWeight n) (hl : ∀ τ (i : Fin a) (i' : Fin n), (i : ℕ) = i' → (lam τ).w i = (lam' τ).w i') (r : ∀ {k : ℕ} (l : ι' → DomWeight k), GL (Fin k) E →* Module.End K (VV K l)) (c0 : VV K lam →ₛₗ[σ] W) (hli : LinearIndependent ℂ fun i => c0 (bVV K lam i)) {Γ : Type} (rat : Γ → GL (Fin a) E → Prop) (ρ : Γ → W → W) (hi : ∀ γ γE, rat γ γE → ∀ v, c0 (r lam γE v) = ρ γ (c0 v)) : ∃ c : VV K lam' →ₛₗ[σ] W, (∀ γ γE g, rat γ γE → MatEq γE g → ∀ v, c (r lam' g v) = ρ γ (c v)) ∧ LinearIndependent ℂ fun i => c (bVV K lam' i)

theorem **isWtOfPar_unique** {N N' : ℕ} {S : SPar} (hS : ∀ x ∈ S, x.2 = true) {lam : DomWeight N} {lam' : DomWeight N'} (h : Bridge.IsWtOfPar lam S) (h' : Bridge.IsWtOfPar lam' S) (i : Fin N) (i' : Fin N') (e : (i : ℕ) = i') : lam.w i = lam'.w i'

theorem **isWtOfPar_sgTrue** (D : D7) (τ : D.what₂.Tau) : Bridge.IsWtOfPar (RowPar.of (D.hc τ)).sgTrue ((Ptw D).Dpar D.R.m τ)

theorem **lamRdef_eq** (X : Stages) (τ) (i : Fin X.toD7.sp.X'.rank) (i' : Fin (X.B.R.N X.B.R.m - 1)) (e : (i : ℕ) = i') : (Bridge.lamRdef (Ptw X.toD7) τ).w i = (X.B.wts.sgTrue τ).w i'

theorem **labLL_eq** (X : Stages) : labLL X.toD7 = X.B.fields.emb

theorem **exists_matEq** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (hG : V.gram = 1) (h : Bridge.Gf V) : ∃ x : UAd D.sp.cm n, R3.MatEq (R3.matIn V 1 h) (x : GL (Fin n) _)

theorem **lamLdef_eq** (X : Stages) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ X.B.R.m) (τ) (i : Fin (X.toD7.sp.V j).rank) (i' : Fin X.B.R.nV) (e : (i : ℕ) = i') : (Bridge.lamLdef (D

theorem **gram_X'** (X : Stages) {px : PreX X.B.toPre} (I : ColIn X.B px) (hI : I.Ok) : X.toD7.sp.X'.gram = 1

theorem **gram_V** (X : Stages) {px : PreX X.B.toPre} (I : ColIn X.B px) (hI : I.Ok) (i : Fin X.toD7.R.m) : (Vc X.toD7 i).gram = 1

theorem **matEq_unique** {A : Type} [CommRing A] {a b : ℕ} {x : GL (Fin a) A} {y y' : GL (Fin b) A} (h : MatEq x y) (h' : MatEq x y') : y = y'

theorem **exists_brgStdV** (hI : I.Ok) (L : Formalism.IndefR.Links.LevCh X.toD7 (TupU I U)) (i : Fin X.toD7.R.m) (hβ : L.βV i = 1) : ∃ cV : CoefA (Vc X.toD7 i), cV.Is ∧ ∃ jL, BrgStdV X I hX U Xg tr L i cV jL

theorem **exists_brgStdR** (hI : I.Ok) (L : Formalism.IndefR.Links.LevCh X.toD7 (TupU I U)) (hβ : L.βX = 1) (θ : InfFam X.toD7) (hθ : IsInfFam θ) (M : DSModel X.toD7.sp.X' (parB (Ptw X.toD7) X.toD7.R.m)) : ∃ cX : CoefA X.toD7.sp.X', cX.Is ∧ ∃ jR, BrgStdR X I hX U Xg tr L cX jR

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.JEvExists

theorem **PreChar.of_char** {F : Face D A} (h : R.Char F) : PreChar R

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.JetsInIffStepField

theorem **monoDeg_eq_sum** [Fintype T] (e : DVar T a b →₀ ℕ) (τ : T) : monoDeg e τ = ∑ β, ∑ α, e (τ, β, α)

theorem **exists_monoDeg** [Fintype T] (ha : 0 < a) (hb : 0 < b) (n : T → ℕ) : ∃ e : DVar T a b →₀ ℕ, ∀ τ, monoDeg e τ = n τ

theorem **nuDeg_nonneg** (B : Base) (u : B.Word) {i : ℕ} (hi : i < B.R.m) (τ : B.fields.Tau) : 0 ≤ nuDeg B u i τ

theorem **exists_monoDeg_nuDeg** (B : Base) (u : B.Word) {i : ℕ} (hi : i < B.R.m) : ∃ e : DVar B.fields.Tau (B.R.N i) (B.R.N (i + 1)) →₀ ℕ, ∀ τ, (monoDeg e τ : ℤ) = nuDeg B u i τ

private theorem **monoDeg_eq_sum_filter** {T : Type} {a b : ℕ} {d : DecidableEq T} (e : DVar T a b →₀ ℕ) (τ : T) : monoDeg e τ = ∑ k ∈ e.support.filter (fun k => k.1 = τ), e k

theorem **jetsIn_iff** : I.JetsIn ↔ ∀ (a : I.ι) (u : B.Word) (i : ℕ), i < B.R.m → ∀ N N' c x, I.JetRange a u i N N' c x → ∀ e, (∀ τ, (monoDeg e τ : ℤ) = nuDeg B u i τ) → I.jetNum a u i N N' c x e ∈ B.embLL.fieldRange

theorem **jetRange_std** (a : I.ι) (u : B.Word) (i : ℕ) : I.JetRange a u i (stdLat B.fields.E (B.R.N i)) (stdLat B.fields.E (B.R.N (i + 1))) 1 0

theorem **JetsIn.exists_eq** (h : I.JetsIn) (hi : i < B.R.m) (hx : I.JetRange a u i N N' c x) (he : ∀ τ, (monoDeg e τ : ℤ) = nuDeg B u i τ) : ∃ y : B.LL, B.embLL y = I.jetNum a u i N N' c x e

theorem **JetsIn.isAlgebraic** (h : I.JetsIn) (hi : i < B.R.m) (hx : I.JetRange a u i N N' c x) (he : ∀ τ, (monoDeg e τ : ℤ) = nuDeg B u i τ) : IsAlgebraic ℚ (I.jetNum a u i N N' c x e)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LargeFor

theorem **locTie** : LocTie h Φ.k

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LatAtLattice

theorem **latAt_lattice** (u : B.Word) (a : I.ι) (i : ℕ) (c : (I.stp u a i).Cl) : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LatAtOmPHtEval

theorem **pow_smul_eq** (w : P.OmC) : ((p : ɪC.S) ^ n) • w = p ^ n • w

theorem **isVal_iff** (hadd : ∀ a b, lv (a + b) = lv a * lv b) (hker : ∀ a, lv a = 1 → ∃ b, a = p ^ n • b) {t : P.T} {v v' : P.OmC} (hv : IsVal hφ lv h t v) : (∃ w ∈ P.latOm, v - v' = ((p : ɪC.S) ^ n) • w) ↔ IsVal hφ lv h t v'

theorem **fibLat_eq** (hadd : ∀ a b, lv (a + b) = lv a * lv b) (hker : ∀ a, lv a = 1 → ∃ b, a = p ^ n • b) (htors : ∀ a, lv a ^ p ^ n = 1) {J : Type} [Fintype J] (b : J → La) (hb : ∀ a, h a ∈ (spanSub fun i => h (b i)).sec ⊤) {f : P.T → P.OmC} (hf : ∀ t, IsVal hφ lv h t (f t)) : (fibLat hφ (preSub (cokernel.π (p ^ n • 𝟙 (omega G))) (spanSub fun i => h (b i)))).toAddSubgroup = P.latOm.toAddSubgroup ⊓ (Ospan (C

theorem **latAt_le** (M : Sub Z.om) : latAt ct y M ≤ (htpTrue ct).om y

theorem **IsValAt.mem** {t : (htpTrue ct).Tp y} {v : (htpTrue ct).OmV y} (hv : IsValAt ct h y t v) : v ∈ (htpTrue ct).om y

theorem **exists_val** : ∃ f : (htpTrue ct).Tp y →+ (htpTrue ct).OmV y, ∀ t, IsValAt ct h y t (f t)

theorem **isValAt_iff** {t : (htpTrue ct).Tp y} {v v' : (htpTrue ct).OmV y} (hv : IsValAt ct h y t v) : (∃ w ∈ (htpTrue ct).om y, v - v' = ((p : (ct.fld y).C) ^ n) • w) ↔ IsValAt ct h y t v'

theorem **latAt_omP** {f : (htpTrue ct).Tp y → (htpTrue ct).OmV y} (hf : ∀ t, IsValAt ct h y t (f t)) : latAt ct y (omP h) = (htpTrue ct).om y ⊓ (Ospan (C

theorem **IsHT.pull** {B : ct.Mdl} [IsIso B.π] (e : Z.i = B.i) {h : HTOn B n} (hh : IsHT ct h) : IsHT ct (pull Z e h)

theorem **A_Xn** (n : ℕ) : (ct.Xn n).i.l.A = κ.l.A * p ^ n

theorem **pow_dvd_A** (n : ℕ) : p ^ n ∣ (ct.Xn n).i.l.A

theorem **eq_of_i_eq** {n m : ℕ} (h : (ct.Xn n).i = (ct.Xn m).i) : n = m

theorem **blowUp_i** (J : ct.Idl Z) : (ct.blowUp Z J).i = Z.i

theorem **isValAt_htEval** (h : HTOn (ct.Xn n) n) (y : ct.Pt (ct.Xn n)) (t : (htpTrue ct).Tp y) : IsValAt ct h y t (htEval ct h y t)

theorem **isHT_iff** (h : HTOn (ct.Xn n) n) : IsHT ct h ↔ ∀ (y : ct.Pt (ct.Xn n)) (t : (htpTrue ct).Tp y), htEval ct h y t ∈ (htpTrue ct).om y ∧ ∃ v ∈ (htpTrue ct).om y, htEval ct h y t - (htpTrue ct).ht y t = ((p : (ct.fld y).C) ^ n) • v

theorem **IsModelOf.i_eq** {h : HTOn (ct.Xn n) n} (H : IsModelOf ct Z n h) : Z.i = (ct.Xn n).i

theorem **isHTBlowup_iff** {Z' : ct.Mdl} : IsHTBlowup ct Z' (ct.Xn n) ↔ ∃ h, IsModelOf ct Z' n h

theorem **htData_eq** (H : IsHTBlowup ct Z (ct.Xn n)) : ∃ (h : HTOn (ct.Xn n) n) (H' : IsModelOf ct Z n h), htData ct Z = ⟨n, pull Z (H'.i_eq ct) h⟩

theorem **latAt_omSub** (H : IsHTBlowup ct Z (ct.Xn n)) (y : ct.Pt Z) : latAt ct y (omSub ct Z) = (htpTrue ct).om y ⊓ (Ospan (C

theorem **le_gen** {U : Z.X.Opens} (e : ⊤ ≤ y.z ⁻¹ᵘ U) : ⊤ ≤ y.gen ⁻¹ᵘ U

theorem **lat_mono** {M M' : Sub Z.om} (h : ∀ U, M.sec U ≤ M'.sec U) : y.lat M ≤ y.lat M'

theorem **actOm_eOm** (σ : D.host.E →+* ℂ) (a : 𝒪 D.host.E) (v : y.Om) : y.actOm a (y.eOm σ v) = y.fK (embK Z.i.c σ a) • y.eOm σ v

theorem **latticeAt_omega_eq_inf_hodgeTateSpan** {n : ℕ} (h : HTOn (ct.Xn n) n) (y : ct.Pt (ct.Xn n)) : latAt ct y (omP h) = (htpTrue ct).om y ⊓ (Ospan (C

theorem **htSES_k₀** {Z : ct.Mdl} {n : ℕ} (H : IsHTBlowup ct Z (ct.Xn n)) (τ : PsiT D) : At.SES.k₀ ((htmTrue ct).htSES Z τ) = n - 2

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LiftWeightComplexPointEqTwist

theorem **isIntegral_char** (χ : G →* Kˣ) (g : G) : IsIntegral ℤ ((χ g : Kˣ) : K)

theorem **twRep_apply** (g : G) (x : Fin n → LL) : twRep A g x = (((A.chi g)⁻¹ : LLˣ) : LL) • A.ρ g x

theorem **twRep_of_chi_eq_one** {g : G} (h : A.chi g = 1) : twRep A g = A.ρ g

theorem **eOp_eq** (a : MonoidAlgebra LL G) : A.eOp (fun g => a.coeff g) = (twRep A).asAlgebraHom a

theorem **mem_eigenspace_iff₂** (J : Subgroup G) (lam : J →* LLˣ) (x : Fin n → LL) : x ∈ A.typeP J lam ↔ ∀ p : J, twRep A p x = ((lam p : LLˣ) : LL) • x

theorem **supLat_isLatIn** (h1 : A.IsKsForm) (h2 : A.MKFull) : IsLatIn ⊤ (supLat A)

theorem **supLat_stable** (h : A.MKStable) (g : G) : (supLat A).map ((twRep A g).restrictScalars (𝒪 LL)) ≤ supLat A

theorem **lPr_mul_lvlOp** (h : BlockHyp B ℓ v0 s) : B.lPr ℓ Z * B.S.lvlOp B.r ℓ.P ℓ.ψ (idemOf ℓ.X) = B.lPr ℓ Z

theorem **glCast_apply** (hr : r = n + 1) (h : GL (Fin (n + 1)) k) (i j : Fin r) : (glCast hr h : Matrix (Fin r) (Fin r) k) i j = (h : Matrix (Fin (n + 1)) (Fin (n + 1)) k) (Fin.cast hr i) (Fin.cast hr j)

theorem **glCast_mem_unip** (hr : r = n + 1) (h : GL (Fin (n + 1)) k) : glCast hr h ∈ unip k r ↔ IsUniU (h : Matrix (Fin (n + 1)) (Fin (n + 1)) k)

theorem **edgeSum_glCast** (hr : r = n + 1) (h : GL (Fin (n + 1)) k) : edgeSum supEdge (glCast hr h) = supd (h : Matrix (Fin (n + 1)) (Fin (n + 1)) k)

theorem **exists_uni_supd** (hn : 0 < n) (a : k) : ∃ u : GL (Fin (n + 1)) k, IsUniU (u : Matrix (Fin (n + 1)) (Fin (n + 1)) k) ∧ supd (u : Matrix (Fin (n + 1)) (Fin (n + 1)) k) = a

theorem **exists_psiU_eq** (hn : 0 < n) (v : ι) (a : kf v) : ∃ u ∈ Subgroup.pi Set.univ (fun v => uniSub (kf v) n), (∀ w, supd (((u w : mirabSub (kf w) n) : GL (Fin (n + 1)) (kf w)) : Matrix (Fin (n + 1)) (Fin (n + 1)) (kf w)) = Pi.single (M

theorem **rank_eq** (R : Ranks) : R.r = rankPred R + 1

theorem **rankPred_pos** (R : Ranks) : 0 < rankPred R

theorem **eG_mirIncl** (x : Mir R) : R.eG (mirIncl R x) = mirToGL R x

theorem **mirIncl_injective** : Function.Injective (mirIncl R)

theorem **mem_pi_unip_iff** (g : G0 R.Pg D.R.r) : g ∈ Subgroup.pi Set.univ (fun v : R.Pg => unip (kP v.1) D.R.r) ↔ ∃ u ∈ uni R, mirToGL R u = g

theorem **mem_base_iff** (h : R.grp_levels F) (g : G) : g ∈ F.P0 ↔ ∃ u ∈ uni R, mirIncl R u = g

theorem **IsLevelChar.unique** (h : R.grp_levels F) {χ χ' : ∀ v : R.Pg, AddChar (Res R v) (𝒪 D.what₂.LL)} (h1 : IsLevelChar R F χ) (h2 : IsLevelChar R F χ') : χ = χ'

theorem **exists_isLevelChar** (h : R.grp_levels F) : ∃ χ, IsLevelChar R F χ

theorem **levelChar_spec** (h : R.grp_levels F) : IsLevelChar R F (levelChar R F)

theorem **ofObj_onOf** : ChainArgs.ofObj (onOf R F c a) = a

theorem **tieU** : TieU (onOf R F c a) fun v => uniSub (Res R v) (rankPred D.R)

theorem **tiePsi** : TiePsi (onOf R F c a) (levelChar R F)

theorem **tieLevelNum** : TieLevelNum (onOf R F c a) (rankPred D.R) D.R.f

theorem **tieProj** : TieProj (onOf R F c a) (fun j => twRep (R.FW j)) fun _ => idemOf A.X

theorem **goodOp_idem** (hX : ∀ χ ∈ A.X, IsAbsIrrChar χ) (j : Fin D.R.m) : goodOp R j ∘ₗ goodOp R j = goodOp R j

theorem **goodOp_comm** (hX : ∀ χ ∈ A.X, IsAbsIrrChar χ) (j : Fin D.R.m) (g : G) : goodOp R j ∘ₗ twRep (R.FW j) g = twRep (R.FW j) g ∘ₗ goodOp R j

theorem **coordMap_injective** (j : Fin D.R.m) : Function.Injective (coordMap R F j)

theorem **range_coordMap** (j : Fin D.R.m) : LinearMap.range (coordMap R F j) = R.WL F false j

theorem **eig_eq** (h : R.grp_levels F) (j : Fin D.R.m) : eig ((onOf R F c a).ρ j) (onOf R F c a).U (onOf R F c a).ψ = (R.FW j).typeP F.P0 F.ψ0⁻¹

theorem **zg_eq** (h : R.grp_levels F) (j : Fin D.R.m) : (onOf R F c a).Zg j = R.WL F false j

theorem **cut_eq** (h : R.grp_levels F) (j : Fin D.R.m) : cut ((onOf R F c a).Zg j) ((onOf R F c a).Ms j) = (R.FW j).Mj F.P0 F.ψ0⁻¹ fun g => (idemOf A.X).coeff g

theorem **blockHyp** (hOk : A.fam.Ok s) (j : Fin D.R.m) (t : T) : BlockHyp (A.fam.L j t) (liftLvl F) A.v0 s

theorem **thetaOp_srcGood** (hOk : A.fam.Ok s) (j : Fin D.R.m) (t : T) : thetaOp R F j t ∘ₗ srcGood F j t = thetaOp R F j t

theorem **thetaOp_srcProj** (hOk : A.fam.Ok s) (j : Fin D.R.m) (t : T) : thetaOp R F j t ∘ₗ srcProj R F j t = thetaOp R F j t

theorem **goodOp_thetaOp** (hX : ∀ χ ∈ A.X, IsAbsIrrChar χ) (j : Fin D.R.m) (t : T) : goodOp R j ∘ₗ thetaOp R F j t = thetaOp R F j t

theorem **srcProj_idem** (hOk : A.fam.Ok s) (j : Fin D.R.m) (t : T) : srcProj R F j t ∘ₗ srcProj R F j t = srcProj R F j t

theorem **tjId** (hOk : A.fam.Ok s) (j : Fin D.R.m) (v : HeightOneSpectrum (𝒪 D.what₂.LL)) : (onOf R F c a).TjId j v (R.mu v)

theorem **tieCoord** (hOk : A.fam.Ok s) (h : R.grp_levels F) (hμ : R.tup_mu_conj) : TieCoord (onOf R F c a)

theorem **leafProj** (hX : ∀ χ ∈ A.X, IsAbsIrrChar χ) : LeafProj (onOf R F c a)

theorem **leafIdentity** (hOk : A.fam.Ok s) : LeafIdentity (onOf R F c a)

theorem **leafSup** (h : R.LIntegral) : LeafSup (onOf R F c a)

theorem **leafMackey** (h : R.grp_levels F) : LeafMackey (onOf R F c a)

theorem **leafLevelSq** (h : R.grp_record) : LeafLevelSq (onOf R F c a) (Res R) D.R.f

theorem **leafYBound** {L : SizeData₂ D} (Q : ParameterChoice' D L) : LeafYBound (onOf R F c a)

theorem **leafIndexU** (h : R.grp_record) (hY : LeafYBound (onOf R F c a)) : LeafIndexU (onOf R F c a)

theorem **leafUnip** (h : R.grp_record) : LeafUnip (onOf R F c a)

theorem **onOf_leaves** (hOk : A.fam.Ok s) (hC : R.Char F) (hI : R.LIntegral) {L : SizeData₂ D} (Q : ParameterChoice' D L) : TieCoord (onOf R F c a) ∧ LeafProj (onOf R F c a) ∧ LeafIdentity (onOf R F c a) ∧ LeafSup (onOf R F c a) ∧ LeafMackey (onOf R F c a) ∧ LeafIndexU (onOf R F c a) ∧ LeafUnip (onOf R F c a)

theorem **tC_cpt** {K : Type} [Field K] [NumberField K] [StarRing K] {Bs : DefBase K} {A : DefAmb Bs} (X : CxData A) {fc : DefSub A → DefFace Bs} (cr : Crd A fc) (b : Bool) (j : Bs.J) (t : Bs.Tup) (y : Fin ((lv (fc X.zs) b).dZ j t) → K) : X.tC cr b j t (cpt Bs.v0.embedding y) = X.θ' j t (X.jL j t (cr.toL X.zs b j t y))

theorem **liftWeight_complexPoint_eq_twist** {K : Type} [Field K] [NumberField K] [StarRing K] {Bs : DefBase K} {A : DefAmb Bs} (X : CxData A) (b : Bool) (j : Bs.J) (z : Fin (X.dW b j) → K) : X.w0C b j (cpt Bs.v0.embedding z) = ∑ k, Bs.v0.embedding (z k) • X.eW b j (X.bW b j Bs.v0 k)

theorem **out_sum** {s : Bool} {E ι : Type} [NormedAddCommGroup E] [InnerProductSpace ℂ E] [Fintype ι] (f : ι → Tw s E) : Tw.out (∑ k, f k) = ∑ k, Tw.out (f k)

theorem **j_coordMap** (j : Fin D.R.m) (z : Fin (finrank D.what₂.LL (R.WL F false j)) → D.what₂.LL) : (R.FW j).j (sg0 D A) (coordMap R F j z) = Tw.out (s

theorem **j_thetaMat** (h5 : R.LTheta F) (hr : R.LRatTh F) (j : Fin D.R.m) (t : T) (y : Fin ((A.fam.L j t).dZ (liftLvl F) (liftSub R F j t)) → D.what₂.LL) : (R.FW j).j (sg0 D A) (coordMap R F j ((thetaMat R F j t).mulVec y)) = R.thG j t (R.jV j t ((A.fam.L j t).toF (liftLvl F) (liftSub R F j t) y))

theorem **j_thetaOp** (h5 : R.LTheta F) (hr : R.LRatTh F) (j : Fin D.R.m) (t : T) (x : Fin (srcDim F j t) → D.what₂.LL) : (R.FW j).j (sg0 D A) (thetaOp R F j t x) = R.thG j t (R.jV j t ((A.fam.L j t).toF (liftLvl F) (liftSub R F j t) (((A.fam.L j t).lPr (liftLvl F) (liftSub R F j t)).mulVec x)))

theorem **range_thetaOp** {s : ℕ} (hOk : A.fam.Ok s) (j : Fin D.R.m) (t : T) (hb : Function.Surjective (thetaMat R F j t).mulVec) : LinearMap.range (thetaOp R F j t) = R.WL F false j

theorem **psi_mul_base** (h : R.grp_levels F) (u : Mir R) (hu : u ∈ (onOf R F c a).U) (hm : mirIncl R u ∈ F.P0) : (((onOf R F c a).ψ u : 𝒪 D.what₂.LL) : D.what₂.LL) * ((F.ψ0 (mirIncl R u, hm) : D.what₂.LLˣ) : D.what₂.LL) = 1

theorem **card_U** (h : R.grp_record) : Nat.card (onOf R F c a).U = D.host.g 0 ^ (D.R.r * (D.R.r - 1) / 2 * D.R.f) * D.host.g 1 ^ (D.R.r * (D.R.r - 1) / 2 * D.R.f)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LinearIndependentFlagVec

theorem **mem_Ospan_image** (b : Fin m → M) (s : Set (Fin m)) (v : M) : v ∈ Ospan (C

theorem **Adapted.exists_diag** {Λ : ℕ → AddSubgroup M} {b b' : Fin m → M} (h : Adapted C Λ b) (h' : Adapted C Λ b') : ∃ d : Fin m → C, Diag b b' d ∧ ∀ a, ‖d a‖ ≤ 1

theorem **Adapted.norm_diag** {Λ : ℕ → AddSubgroup M} {b b' : Fin m → M} (h : Adapted C Λ b) (h' : Adapted C Λ b') (li : LinearIndependent C b) {d : Fin m → C} (hd : Diag b b' d) (a : Fin m) : ‖d a‖ = 1

theorem **linearIndependent_of_diag** {w b : Fin m → M} {d : Fin m → C} (li : LinearIndependent C w) (h : Diag w b d) (hd : ∀ a, d a ≠ 0) : LinearIndependent C b

theorem **mem_span_of_diag** {w b : Fin m → M} (h : ∃ d : Fin m → C, Diag w b d) {i a : Fin m} (hi : i ≤ a) : b i ∈ Submodule.span C (w '' Set.Iic a)

theorem **finrank_span_Iic** {w : Fin m → M} (li : LinearIndependent C w) (a : Fin m) : Module.finrank C (Submodule.span C (w '' Set.Iic a)) = a + 1

theorem **mem_Ospan_of_coord** {n : ℕ} {V : Fin n → M} (li : LinearIndependent C V) (e : Fin m ↪ Fin n) {w : Fin m → M} (he : ∀ i, V (e i) = w i) {s : Set (Fin m)} {v : M} (hv : v ∈ Submodule.span C (w '' s)) {c : Fin n → C} (hc : ∀ k, ‖c k‖ ≤ 1) (e' : v = ∑ k, c k • V k) : v ∈ Ospan (C

theorem **Ospan_le** {s : Set M} {Λ : AddSubgroup M} (h : ∀ a : C, ‖a‖ ≤ 1 → ∀ v ∈ s, a • v ∈ Λ) : Ospan (C

theorem **tanOf_map_sum** (φ : S →+* T) (hx : spMap φ » x = x') {m : ℕ} (u : Fin m → TanM x t.G) (a : Fin m → S) : tanOf (isTan_map φ hx (isTan_tv (∑ i, a i • u i))) = ∑ i, φ (a i) • tanOf (isTan_map φ hx (isTan_tv (u i)))

theorem **left_bcTan** (φ : S →+* T) (hx : spMap φ » x = x') (v : TanM x t.G) : (tv (bcTan φ hx v)).left = spMap (dualMap φ) » (tv v).left

theorem **isGreatest_nrmSet_chart** {X : Scheme.{0}} {p : ℕ} {C : SCF p} (z : Sp ₁(intO ₁C.S) ⟶ X) (hp : p ≠ 0) {L : LB X} (hL : L.IsLB) {i : L.ι} (hU : IsAffineOpen (L.U i)) (h : ⊤ ≤ z ⁻¹ᵁ L.U i) (s : Γ(GlobalTheory.LB.toMod L, PtNorm.gen X p)) : IsGreatest (PtNorm.nrmSet z _ s) ‖valR (PtNorm.zq z) _ (GlobalTheory.LB.coord s i)‖

theorem **gen_π** : spMap (intO ₁y.C.S).subtype » y.z » Z.π = y.gen » Z.π

theorem **mem_lieL** {v : Lie y} : v ∈ lieL y ↔ ∃ w, IsTanAtG (Y Z.i).t (y.z » Z.π) w ∧ (tv v).left = spMap (dualMap (intO ₁y.C.S).subtype) » w

theorem **smul_mem_lieL** {a : ₁y.C.S} (ha : ‖a‖ ≤ 1) {v : Lie y} (hv : v ∈ lieL y) : a • v ∈ lieL y

theorem **mem_eig** {σ : E →+* ℂ} {v : Lie y} : v ∈ eig y σ ↔ ∀ a : 𝒪 E, act y a v = y.P.emb σ a • v

theorem **mem_eig_iff_left** {σ : E →+* ℂ} {v : Lie y} : v ∈ eig y σ ↔ ∀ a : 𝒪 E, (tv v).left ≫ ((Y Z.i).t.act.act a).left = spMap (epsMul (y.P.fK (embK Z.i.c σ a))) » (tv v).left

theorem **range_gen** : Set.range y.gen ⊆ Z.opn

theorem **isFrame_of_flagAt** {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) : IsFrame (Y Z.i).t y.P.fK (y.gen » Z.π) u

theorem **flagVec_mem_eig** {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : flagVec h σ a ∈ eig y σ

theorem **existsUnique_flagVec** {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) {σ : E →+* ℂ} {v : Lie y} (hv : v ∈ eig y σ) : ∃! c : Fin (Dw.mult σ) → ₁y.C.S, v = ∑ a, c a • flagVec h σ a

theorem **linearIndependent_flagVec** {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) (σ : E →+* ℂ) : LinearIndependent ₁y.C.S (flagVec h σ)

theorem **exists_diag_flagVec** (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) {u u' : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) (h' : Z.FlagAt y.gen u') (σ : E →+* ℂ) : ∃ d : Fin (Dw.mult σ) → ₁y.C.S, Diag (flagVec h σ) (flagVec h' σ) d

theorem **Wst_le** (σ : E →+* ℂ) (i : ℕ) : Wst y σ i ≤ eig y σ

theorem **Wst_succ** (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : Wst y σ (a + 1) = Submodule.span ₁y.C.S (flagVec h σ '' Set.Iic a)

theorem **Wst_eq_eig** {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) (σ : E →+* ℂ) {i : ℕ} (hi : Dw.mult σ ≤ i) : Wst y σ i = eig y σ

theorem **finrank_Wst_succ** (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) {u : Vecs Dw (Y Z.i).t ₁y.C.S} (h : Z.FlagAt y.gen u) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : Module.finrank ₁y.C.S (Wst y σ (a + 1)) = a + 1

theorem **flagAt_of_intFrame** {U : Z.X.Opens} {hU : IsAffineOpen U} {w : Vecs Dw (Y Z.i).t Γ(Z.X, U)} (hw : IntFrame Z hU w) (h : ⊤ ≤ y.gen ⁻¹ᵁ U) : Z.FlagAt y.gen (w.bc (valMap y.gen U h).hom)

theorem **gen_le** {U : Z.X.Opens} (h : ⊤ ≤ y.z ⁻¹ᵁ U) : ⊤ ≤ y.gen ⁻¹ᵁ U

theorem **adapted_taut** (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) {U : Z.X.Opens} {hU : IsAffineOpen U} {w : Vecs Dw (Y Z.i).t Γ(Z.X, U)} (hw : IntFrame Z hU w) (h : ⊤ ≤ y.z ⁻¹ᵁ U) (σ : E →+* ℂ) : Adapted ₁y.C.S (taut y σ) (flagVec (flagAt_of_intFrame hw (gen_le h)) σ)

theorem **gen_le_genO** (hp : p ≠ 0) : ⊤ ≤ y.gen ⁻¹ᵁ Z.genO

theorem **secVal_change** (hp : p ≠ 0) {F : Z.X.Modules} {L : LB Z.X} (e : F ≅ GlobalTheory.LB.toMod L) (f : Γ(F, Z.gen0)) {i i' : L.ι} (h : ⊤ ≤ y.gen ⁻¹ᵁ L.U i) (h' : ⊤ ≤ y.gen ⁻¹ᵁ L.U i') : secVal y e f i = valR y.gen _ (L.g i i').val * secVal y e f i'

theorem **nrm_eq_secVal** (hp : p ≠ 0) {F : Z.X.Modules} {L : LB Z.X} (hL : L.IsLB) (e : F ≅ GlobalTheory.LB.toMod L) (f : Γ(F, Z.gen0)) {i : L.ι} (hU : IsAffineOpen (L.U i)) (h : ⊤ ≤ y.z ⁻¹ᵁ L.U i) : Mdl.nrm F f y = ‖secVal y e f i‖

theorem **exists_diag_transition** (hty : ∀ σ, Z.ty σ = Finset.range (Dw.mult σ + 1)) {U U' : Z.X.Opens} {hU : IsAffineOpen U} {hU' : IsAffineOpen U'} {w : Vecs Dw (Y Z.i).t Γ(Z.X, U)} {w' : Vecs Dw (Y Z.i).t Γ(Z.X, U')} (hw : IntFrame Z hU w) (hw' : IntFrame Z hU' w') {g : Γ(Z.X, U ⊓ U')} {e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ} (hg : ∀ (k : Type) [Field k] (φ : Γ(Z.X, U) →+* k) (φ' : Γ(Z.X, U') →+* k) (d : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → k), spMap φ ≫ hU.fromSpec = spMap φ' ≫ hU'.fromSpec → Set.range (spMap φ ≫ hU.fromSpec) ⊆ Z.opn → (∀ σ a, ∃ cf : Fin (Dw.mult σ) → k, cf a = d σ a ∧ (∀ b, a < b → cf b = 0) ∧ IsCombAtG (Y Z.i).t (spMap φ ≫ hU.fromSpec ≫ Z.π) (w'.bc φ' σ) cf (w.bc φ σ a)) → valR (spMap φ ≫ hU.fromSpec) _ g = ∏ σ, ∏ a, d σ a ^ e σ a) (h : ⊤ ≤ y.gen ⁻¹ᵁ U) (h' : ⊤ ≤ y.gen ⁻¹ᵁ U') : ∃ d : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ɪy.C.S, (∀ σ, Diag (flagVec (flagAt_of_intFrame hw' h')

theorem **prod_expo** (l : PsiT D → ℕ → ℤ) {k : Type} [Field k] (d : ∀ σ : D.host.E →+* ℂ, Fin (S.Dw.mult σ) → k) (hd : ∀ σ a, d σ a ≠ 0) : ∏ σ, ∏ a, d σ a ^ ct.expo l σ a = ∏ s : PsiT D × Bool, ∏ a, d (embOf s) a ^ expOf N l s a

theorem **ty_YY** (σ : D.host.E →+* ℂ) : ct.YY.ty σ = Finset.range (S.Dw.mult σ + 1)

theorem **exists_grLB** (l : PsiT D → ℕ → ℤ) : ∃ L, IsGrLB ct.YY (ct.expo l) L

theorem **isGrLB_grLB** (l : PsiT D → ℕ → ℤ) : IsGrLB ct.YY (ct.expo l) (grLB ct l)

theorem **grAff** (l : PsiT D → ℕ → ℤ) (i : (grLB ct l).ι) : IsAffineOpen ((grLB ct l).U i)

theorem **intFrame_grFr** (l : PsiT D → ℕ → ℤ) (i : (grLB ct l).ι) : IntFrame ct.YY (grAff ct l i) (grFr ct l i)

theorem **nonempty_grIso** (l : PsiT D → ℕ → ℤ) : Nonempty ((ct.Lb l).toMod ≅ GlobalTheory.LB.toMod (grLB ct l))

theorem **exists_diag_chartVec** (l : PsiT D → ℕ → ℤ) (y : ct.Pt ct.YY) {i i' : (grLB ct l).ι} (h : ⊤ ≤ y.gen ⁻¹ᵁ (grLB ct l).U i) (h' : ⊤ ≤ y.gen ⁻¹ᵁ (grLB ct l).U i') : ∃ d : ∀ σ : D.host.E →+* ℂ, Fin (S.Dw.mult σ) → ɪy.C.S, (∀ σ, Diag (chartVec l y h' σ) (chartVec l y h σ) (d σ)) ∧ (∀ σ a, d σ a ≠ 0) ∧ valR y.gen _ ((grLB ct l).g i i').val = ∏ s : PsiT D × Bool, ∏ a, d (embOf s) a ^ expOf N l s a

theorem **IsReading.eq_mul** {l : PsiT D → ℕ → ℤ} {f : ct.Sec (ct.Lb l)} {y : ct.Pt ct.YY} {Λ Λ' : PsiT D × Bool → ℕ → AddSubgroup (Lie y)} {r r' : ℝ} (H : IsReading l f y Λ r) (H' : IsReading l f y Λ' r') {u u' : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → Lie y} {c : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → ɪy.C.S} (hu : ∀ s, Adapted ɪy.C.S (Λ s) (u s)) (hu' : ∀ s, Adapted ɪy.C.S (Λ' s) (u' s)) (hc : ∀ s, Diag (u s) (u' s) (c s)) : r' = r * ∏ s, ∏ a, ‖c s a‖ ^ expOf N l s a

theorem **IsReading.unique** {l : PsiT D → ℕ → ℤ} {f : ct.Sec (ct.Lb l)} {y : ct.Pt ct.YY} {Λ : PsiT D × Bool → ℕ → AddSubgroup (Lie y)} {r r' : ℝ} (H : IsReading l f y Λ r) (H' : IsReading l f y Λ r') : r' = r

theorem **IsReading.nrmIn_eq** {l : PsiT D → ℕ → ℤ} {f : ct.Sec (ct.Lb l)} {y : ct.Pt ct.YY} {Λ : PsiT D × Bool → ℕ → AddSubgroup (Lie y)} {r : ℝ} (H : IsReading l f y Λ r) : nrmIn l f y Λ = r

theorem **exists_chart** (l : PsiT D → ℕ → ℤ) (y : ct.Pt ct.YY) : ∃ i, ⊤ ≤ y.z ⁻¹ᵁ (grLB ct l).U i

theorem **isReading_taut** (l : PsiT D → ℕ → ℤ) (f : ct.Sec (ct.Lb l)) (y : ct.Pt ct.YY) : IsReading l f y (fun s => taut y (embOf s)) (ct.nrm (ct.Lb l) f y)

theorem **nrm_eq_nrmIn** (l : PsiT D → ℕ → ℤ) (f : ct.Sec (ct.Lb l)) (y : ct.Pt ct.YY) : ct.nrm (ct.Lb l) f y = nrmIn l f y fun s => taut y (embOf s)

theorem **IsReading.eq_nrm_mul** {l : PsiT D → ℕ → ℤ} {f : ct.Sec (ct.Lb l)} {y : ct.Pt ct.YY} {Λ : PsiT D × Bool → ℕ → AddSubgroup (Lie y)} {r : ℝ} (H : IsReading l f y Λ r) {u b : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → Lie y} {c : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → ɪy.C.S} (hu : ∀ s, Adapted ɪy.C.S (taut y (embOf s)) (u s)) (hb : ∀ s, Adapted ɪy.C.S (Λ s) (b s)) (hc : ∀ s, Diag (u s) (b s) (c s)) : r = ct.nrm (ct.Lb l) f y * ∏ s, ∏ a, ‖c s a‖ ^ expOf N l s a

theorem **exists_flagAt_adapted** (y : ct.Pt ct.YY) : ∃ (u : Vecs S.Dw (ct.Y ct.YY.i).t ɪy.C.S) (h : ct.YY.FlagAt y.gen u), ∀ σ, Adapted ɪy.C.S (taut y σ) (flagVec h σ)

theorem **exists_isReading** (l : PsiT D → ℕ → ℤ) (f : ct.Sec (ct.Lb l)) (y : ct.Pt ct.YY) {Λ : PsiT D × Bool → ℕ → AddSubgroup (Lie y)} {b : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → Lie y} (hb : ∀ s, Adapted ɪy.C.S (Λ s) (b s)) (li : ∀ s, LinearIndependent ɪy.C.S (b s)) (hW : ∀ s (a : Fin (S.Dw.mult (embOf s))), b s a ∈ Wst y (embOf s) ((a : ℕ) + 1)) : ∃ r, IsReading l f y Λ r

theorem **nrmIn_eq_nrm_mul** (l : PsiT D → ℕ → ℤ) (f : ct.Sec (ct.Lb l)) (y : ct.Pt ct.YY) {Λ : PsiT D × Bool → ℕ → AddSubgroup (Lie y)} {u b : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → Lie y} {c : ∀ s : PsiT D × Bool, Fin (S.Dw.mult (embOf s)) → ɪy.C.S} (hu : ∀ s, Adapted ɪy.C.S (taut y (embOf s)) (u s)) (hb : ∀ s, Adapted ɪy.C.S (Λ s) (b s)) (li : ∀ s, LinearIndependent ɪy.C.S (b s)) (hc : ∀ s, Diag (u s) (b s) (c s)) : nrmIn l f y Λ = ct.nrm (ct.Lb l) f y * ∏ s, ∏ a, ‖c s a‖ ^ expOf N l s a

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LinearIndependentOfCutSec

theorem **valAt_algebraMap** (a : Z.presheaf.stalk (below y)) : valAt y (algebraMap _ Z.functionField a) = (Scheme.stalkClosedPointTo y).hom a

theorem **valAt_add** (hf : RegAt (below y) f) (hg : RegAt (below y) g) : valAt y (f + g) = valAt y f + valAt y g

theorem **valAt_mul** (hf : RegAt (below y) f) (hg : RegAt (below y) g) : valAt y (f * g) = valAt y f * valAt y g

theorem **valAt_pow** (hf : RegAt (below y) f) (m : ℕ) : valAt y (f ^ m) = valAt y f ^ m

theorem **regAt_pow** {x : Z} (hf : RegAt x f) (m : ℕ) : RegAt x (f ^ m)

theorem **valAt_ne_zero** (hf : UnitAt (below y) f) : valAt y f ≠ 0

theorem **valAt_zero** : valAt y 0 = 0

theorem **valAt_one** : valAt y 1 = 1

theorem **below_comp** (φ : K →+* K') : below (spMap φ ≫ y) = below y

theorem **valAt_comp** (φ : K →+* K') (f : Z.functionField) : valAt (spMap φ ≫ y) f = φ (valAt y f)

theorem **valAt_fnOf** {U : Z.Opens} (hy : below y ∈ U) (h : genericPoint Z ∈ U) (a : Γ(Z, U)) : valAt y (fnOf U h a) = TangentOrigin.evalPt y U ((Points.le_preimage_iff _ _).2 hy) a

theorem **ratVal_eq_valAt** {k : Type} [CommRing k] {q : Z ⟶ Sp k} {σ : k →+* ℂ} (z : CxPts q σ) (hf : RegAt (under z) f) : ratVal f z = valAt z.1 f

theorem **valAt_const** {k : Type} [Field k] {Y : Var k} {y : Sp K ⟶ Y.X} {φ : k →+* K} (h : y ≫ Y.str = spMap φ) (a : k) : valAt y (algebraMap k Y.Fn a) = φ a

theorem **below_smul** : below (σ • y).1 = below y.1

theorem **valAt_smul** (f : Y.Fn) : valAt (σ • y).1 f = σ (valAt y.1 f)

theorem **map_coef** (i : Fin Y.dim) (l : Fin n) : σ (coef Y E c i l) = coef Y E c i l

theorem **smul_mem_cutSet** (hy : y ∈ cutSet Y E w c) : σ • y ∈ cutSet Y E w c

theorem **GenAt.unitAt** (h : GenAt Y w x k) (h' : GenAt Y w x k') : UnitAt x (w k / w k')

theorem **isEqn_eqn** (Δ : Y.Dv) (x : Y.X) : Δ.IsEqn x (eqn Δ x)

theorem **ResReg.regAt** (h : ResReg Y w Δ f x) (hk : GenAt Y w x k) (hg : Y.KY.IsEqn x g) (he : Δ.IsEqn x e) : RegAt x (f ∗ e ∗ g / w k ^ Y.dim)

theorem **dsVal_pick** {l : Fin Y.dim → Fin n} (hc : ∀ i j, c i j = if j = l i then 1 else 0) (g : Y.Fn) : dsVal Y E w c y k g = valAt y.1 (jacRat Y (fun i => w (l i) / w k) ∗ g)

theorem **resVal_smul** (σ : AlgebraicClosure E ≃ₐ[E] AlgebraicClosure E) (Δ : Y.Dv) (f : Y.Fn) (y : GeomPt Y E) : resVal Y E w c Δ f (σ • y) = σ (resVal Y E w c Δ f y)

theorem **IsCutHerm.unique** {H H' : ReciprocityTheory.GeometryOfNumbersTheory.HermFam E (cutSec Y E w c)} (h : IsCutHerm w c A H) (h' : IsCutHerm w c A H') : H = H'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LinearIndependentVec

theorem **le_preimage_inf** {X : Scheme.{0}} {R : Type} [CommRing R] {x : Sp R ⟶ X} {U U' : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (h' : ⊤ ≤ x ⁻¹ᵁ U') : ⊤ ≤ x ⁻¹ᵁ (U ⊓ U')

theorem **valR_zero_fn** {X : Scheme.{0}} {R : Type} [CommRing R] (x : Sp R ⟶ X) (U : X.Opens) : valR x U 0 = 0

theorem **spMap_valMap** {X : Scheme.{0}} {R : Type} [CommRing R] {U : X.Opens} (hU : IsAffineOpen U) (x : Sp R ⟶ X) (h : ⊤ ≤ x ⁻¹ᵁ U) : spMap (valMap x U h).hom ≫ hU.fromSpec = x

theorem **le_preimage_of_eq** {X : Scheme.{0}} {R : Type} [CommRing R] {U : X.Opens} {hU : IsAffineOpen U} {φ : Γ(X, U) →+* R} {x : Sp R ⟶ X} (hx : spMap φ ≫ hU.fromSpec = x) : ⊤ ≤ x ⁻¹ᵁ U

theorem **oneAt_eq_zero** {σ : E →+* ℂ} {n : ℕ} (h : Dw.mult σ ≤ n) : oneAt Dw σ n = 0

theorem **HasDR.exists_intChart** (hZ : HasDR Z) (x : Z.X) : ∃ c : IntChart Z, x ∈ c.U

theorem **aff_inf** : IsAffineOpen (c.U ⊓ c'.U)

theorem **vec_eq** {R : Type} [CommRing R] (φ : Γ(Z.X, c.U) →+* R) {x : Sp R ⟶ Z.X} (hx : spMap φ ≫ c.aff.fromSpec = x) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : c.vec x h σ a = tanOf (tan_bc c.int φ hx σ a)

theorem **linearIndependent_vec** {k : Type} [Field k] (x : Sp k ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (σ : E →+* ℂ) : LinearIndependent k (c.vec x h σ)

theorem **exists_tri** (hZ : HasDR Z) (σ : E →+* ℂ) : ∃ m : Fin (Dw.mult σ) → Fin (Dw.mult σ) → Γ(Z.X, c.U ⊓ c'.U), (∀ a b, a < b → m a b = 0) ∧ ∀ (R : Type) [CommRing R] (x : Sp R ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (h' : ⊤ ≤ x ⁻¹ᵁ c'.U) (a : Fin (Dw.mult σ)), c.vec x h σ a = ∑ b, valR x _ (m a b) • c'.vec x h' σ b

theorem **chg_eq_zero** (σ : E →+* ℂ) {a b : Fin (Dw.mult σ)} (h : a < b) : c.chg c' hZ σ a b = 0

theorem **vec_eq_sum** {R : Type} [CommRing R] (x : Sp R ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (h' : ⊤ ≤ x ⁻¹ᵁ c'.U) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : c.vec x h σ a = ∑ b, valR x _ (c.chg c' hZ σ a b) • c'.vec x h' σ b

theorem **diag** {k : Type} [Field k] (x : Sp k ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (h' : ⊤ ≤ x ⁻¹ᵁ c'.U) (σ : E →+* ℂ) : Diag (c'.vec x h' σ) (c.vec x h σ) fun a => valR x _ (c.chg c' hZ σ a a)

theorem **valR_chg_self** {k : Type} [Field k] (x : Sp k ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : valR x _ (c.chg c hZ σ a a) = 1

theorem **valR_chg_mul** {k : Type} [Field k] (x : Sp k ⟶ Z.X) (h : ⊤ ≤ x ⁻¹ᵁ c.U) (h' : ⊤ ≤ x ⁻¹ᵁ c'.U) (h'' : ⊤ ≤ x ⁻¹ᵁ c''.U) (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : valR x _ (c.chg c' hZ σ a a) ∗ valR x _ (c'.chg c'' hZ σ a a) = valR x _ (c.chg c'' hZ σ a a)

theorem **isUnit_chg** (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : IsUnit (c.chg c' hZ σ a a)

theorem **tr_add** (e e' : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ) : c.tr c' hZ (e + e') = c.tr c' hZ e ∗ c.tr c' hZ e'

theorem **tr_neg_oneAt** (σ : E →+* ℂ) (a : Fin (Dw.mult σ)) : (c.tr c' hZ (-oneAt Dw σ a)).val = resG _ (inf_comm c.U c'.U).le (c'.chg c hZ σ a a)

theorem **valR_grLB** {k : Type} [Field k] (x : Sp k ⟶ Z.X) (c c' : IntChart Z) {V : Z.X.Opens} (hV : V ≤ c.U ⊓ c'.U) (h : ⊤ ≤ x ⁻¹ᵁ V) : valR x V (res hV ((grLB hZ e).g c c')).val = ∏ σ, ∏ a, valR x _ (c.chg c' hZ σ a a) ^ e σ a

theorem **isLB_grLB** : (grLB hZ e).IsLB

theorem **diagAt_iff_chg** (c c' : IntChart Z) {k : Type} [Field k] (φ : Γ(Z.X, c.U) →+* k) (φ' : Γ(Z.X, c'.U) →+* k) (hx : spMap φ ≫ c.aff.fromSpec = spMap φ' ≫ c'.aff.fromSpec) (σ : E →+* ℂ) (d : Fin (Dw.mult σ) → k) : DiagAt (Y Z.i).t (spMap φ ≫ c.aff.fromSpec ≫ Z.π) (c'.w.bc φ' σ) (c.w.bc φ σ) d ↔ d = fun a => valR (spMap φ ≫ c.aff.fromSpec) _ (c.chg c' hZ σ a a)

theorem **isGrLB_grLB** : IsGrLB Z e (grLB hZ e)

theorem **IsGrLB.lbIso** {L : LB Z.X} (h : IsGrLB Z e L) : LBIso L (grLB hZ e)

theorem **IsGrLB.cls_eq** {L : LB Z.X} (h : IsGrLB Z e L) : cls L h.1 = grCls hZ e

theorem **strHom_eq** : strHom k c.C = c.I.subschemeι ≫ PSpace.str k c.n

theorem **isSmoothProjCurve** : VB.IsSmoothProjCurve k c.C

theorem **isProjCurve** : VB.IsProjCurve k c.C

theorem **exists_iso** {C : Scheme.{0}} [Algebra k Γ(C, ⊤)] (h : VB.IsSmoothProjCurve k C) (f : C ⟶ X) : ∃ (c : Curve k X) (e : C ≅ c.C), e.hom ≫ c.f = f ∧ e.hom ≫ strHom k c.C = strHom k C

theorem **degPic_cls** {L : LB X} (hL : L.IsLB) : c.degPic (cls L hL) = c.degOf L

theorem **degOf_eq** {C : Scheme.{0}} [Algebra k Γ(C, ⊤)] [IsIntegral C] (h : VB.IsProjCurve k C) (e : C ≅ c.C) (he : e.hom ≫ strHom k c.C = strHom k C) {L : LB X} (hL : L.IsLB) : c.degOf L = VB.degLB k (L.pull (e.hom ≫ c.f))

theorem **nonempty_of_point** (x : Sp k ⟶ X) : Nonempty (Curve k X)

theorem **exists_degOf_On** : ∃ c : Curve k (PSpace k 1), ∀ n : ℤ, c.degOf (ProjSpace.On k 1 n) = n

theorem **exists_isVB_lie** : ∃ g, VB.IsVB c.lie g

theorem **Pm_eq** {P : ℤ} (h : VB.IsMaxSubDeg k c.lie P) : c.Pm = P

theorem **degGr_eq** {e : ∀ σ : E →+* ℂ, Fin (Dw.mult σ) → ℤ} {L : LB Z.X} (h : IsGrLB Z e L) : c.degGr hZ e = VB.degLB k (L.pull c.f)

theorem **degStep_eq_zero** {σ : E →+* ℂ} {n : ℕ} (h : Dw.mult σ ≤ n) : c.degStep hZ σ n = 0

theorem **hasDR** : HasDR ct.YY

theorem **mult_embOf** (S : Sig A N j) (τ : PsiT D) : S.Dw.mult (embOf D τ) = j

theorem **mult_conj_embOf** (S : Sig A N j) (τ : PsiT D) : S.Dw.mult (conjEmb (embOf D τ)) = N − j

theorem **mult_le** (S : Sig A N j) (σ : D.host.E →+* ℂ) : S.Dw.mult σ ≤ N

theorem **expo_indWt_apply** (τ : PsiT D) (i : ℕ) (σ : D.host.E →+* ℂ) (a : Fin (S.Dw.mult σ)) : ct.expo (indWt τ i) σ a = (if σ = embOf D τ ∧ (a : ℕ) + 1 = i then 1 else 0) − if σ = conjEmb (embOf D τ) ∧ N − a = i then 1 else 0

theorem **expo_indWt** (τ : PsiT D) {i : ℕ} (h₁ : 1 ≤ i) (h₂ : i ≤ N) : ct.expo (indWt τ i) = oneAt S.Dw (embOf D τ) (i − 1) − oneAt S.Dw (conjEmb (embOf D τ)) (N − i)

theorem **expo_indWt_eq_zero** (τ : PsiT D) {i : ℕ} (h : i = 0 ∨ N < i) : ct.expo (indWt τ i) = 0

theorem **expo_eq_sum** (l : PsiT D → ℕ → ℤ) : ct.expo l = ∑ t : D.what₂.Tau, ∑ i ∈ Finset.Icc 1 N, l (D.eTau t) i • ct.expo (indWt (D.eTau t) i)

theorem **degPiece_eq_zero** (h : i = 0 ∨ N < i) : ct.degPiece c τ i = 0

theorem **degPiece_of_le** (h₁ : 1 ≤ i) (h₂ : i ≤ j) : ct.degPiece c τ i = c.degStep ct.hasDR (embOf D τ) (i − 1)

theorem **degPiece_of_lt** (h₁ : j < i) (h₂ : i ≤ N) : ct.degPiece c τ i = −c.degStep ct.hasDR (conjEmb (embOf D τ)) (N − i)

theorem **sum_degPiece_head** (h : i ≤ j) : ∑ m ∈ Finset.Icc 1 i, ct.degPiece c τ m = ∑ n ∈ Finset.range i, c.degStep ct.hasDR (embOf D τ) n

theorem **sum_degPiece_tail** (h : j ≤ i) : −∑ m ∈ Finset.Icc (i + 1) N, ct.degPiece c τ m = ∑ n ∈ Finset.range (N − i), c.degStep ct.hasDR (conjEmb (embOf D τ)) n

theorem **degWt_intCast** (l : PsiT D → ℕ → ℤ) {L : LB ct.YY.X} (h : IsGrLB ct.YY (ct.expo l) L) : ct.degWt c (fun τ i => l τ i) = VB.degLB k (L.pull c.f)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LinearIndependentVecC

theorem **frameCo_stdMat** (z : Pt Tau j (r − j)) (τ : X.Ψ) : frameCo X hj (stdMat X hj κ z) τ = fromBlocks 1 (z (κ.symm τ)) 0 1

theorem **domPt_stdMat** (z : Pt Tau j (r − j)) : domPt X hj κ (stdMat X hj κ z) = z

theorem **stdMat_mem** (hp : X.p0 = r − j) {z : Pt Tau j (r − j)} (hz : z ∈ DomI Tau j (r − j)) : stdMat X hj κ z ∈ X.dom

theorem **stdFrame_val** (hp : X.p0 = r − j) {z : Pt Tau j (r − j)} (hz : z ∈ DomI Tau j (r − j)) : (stdFrame X hj κ hp z).1 = stdMat X hj κ z

theorem **domPt_stdFrame** (hp : X.p0 = r − j) {z : Pt Tau j (r − j)} (hz : z ∈ DomI Tau j (r − j)) : domPt X hj κ (stdFrame X hj κ hp z).1 = z

theorem **isClosed_bdSet** : IsClosed (bdSet Y σ)

theorem **ptC_fst** (y : FrPt Xo c (CAt σ)) : ptC y ≫ pullback.fst _ _ = xC y

theorem **ptC_snd** (y : FrPt Xo c (CAt σ)) : ptC y ≫ pullback.snd _ _ = spMap σ

theorem **torPt_pr** (y : FrPt Xo c (CAt σ)) : (torPt Y y).1 ≫ torPr Y σ = ptC y ≫ Y.j

theorem **torPt_eq_iff** {y y' : FrPt Xo c (CAt σ)} : torPt Y y = torPt Y y' ↔ y.x = y'.x

theorem **q_of_isUnf** : Dw.q = r − j

theorem **domTor_domPt** (k : U.Comp) (x : (U.X k).dom) : domTor Y U hU e T k (domPt (U.X k) (le_of_isUnf U hU e) (cmEquiv U hU T k) x.1) = torPt Y (U.un k x)

theorem **domTor_moeb** (k : U.Comp) {g : Tau → GL (Fin j ⊕ Fin (r − j)) ℂ} (hg : g ∈ gammaU (U.X k) (le_of_isUnf U hU e) (cmEquiv U hU T k)) {z : Pt Tau j (r − j)} (hz : z ∈ DomI Tau j (r − j)) : domTor Y U hU e T k (moebI g z) = domTor Y U hU e T k z

theorem **exists_mem_framesOn** (V : Unf Xo c σ r) (k : V.Comp) (y : (V.X k).dom) : ∃ Z, ((k, y.1) : Σ k, Frame (V.toPj.X k).Ψ r) ∈ framesOn V.toPj Z

theorem **IsFlatSec.unique** {s s' : (U σ).Comp → Tau → ℂˣ} (h : IsFlatSec U sig hU hσ T s) (h' : IsFlatSec U sig hU hσ T s') : s = s'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LocStOfLoc

theorem **locAt_prime** (F : StFam S κ) (cts : Ats S κ) (p : ℕ) [hp : Fact p.Prime] : locAt F cts p = primeAt F cts p

theorem **locStOf_loc** (F : StFam S κ) (cts : Ats S κ) (p : ℕ) [Fact p.Prime] : (locStOf F cts).loc p = Loc.pi fun v => (F.st (cts p v)).fix.loc

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.LocalPrimeGaloisActionPeriodT

@[simp] theorem **lRoot_zero** : lRoot l 0 = (l : Om l)

theorem **lRoot_succ** (m : ℕ) : lRoot l (m + 1) ^ l = lRoot l m

theorem **lRoot_pow** (m : ℕ) : lRoot l m ^ (l ^ m) = (l : Om l)

theorem **lRoot_ne_zero** (m : ℕ) : lRoot l m ≠ 0

theorem **isPrimitiveRoot_zeta** (m : ℕ) : IsPrimitiveRoot (P.ζQ m) (l ^ m)

theorem **zeta_pow_eq_iff** (m a b : ℕ) : P.ζQ m ^ a = P.ζQ m ^ b ↔ (a : ZMod (l ^ m)) = (b : ZMod (l ^ m))

theorem **exists_isKummer** (g : GQ l) : ∃ c, IsKummer P g c

theorem **IsKummer.unique** {P : LocPrime l} {g : GQ l} {c c' : ℤ_[l]} (h : IsKummer P g c) (h' : IsKummer P g c') : c = c'

theorem **isKummer_kummer** (g : GQ l) : IsKummer P g (kummer P g)

theorem **kummer_eq_zero_iff** (g : GQ l) : kummer P g = 0 ↔ ∀ m, g (lRoot l m) = lRoot l m

@[simp] theorem **kummer_one** : kummer P 1 = 0

theorem **kummer_mul** (g h : GQ l) : (kummer P (g * h) : ℚ_[l]) = kummer P g + (P.χQ g : ℚ_[l]) * kummer P h

theorem **exists_kummer_ne_zero** : ∃ g, kummer P g ≠ 0

theorem **mem_GK_botExt** (g : GQ l) : g ∈ GK (botExt l)

theorem **localPrime_galoisAction_periodT** (g : GQ l) : P.actQ g P.tB = (P.χQ g : ℚ_[l]) • P.tB

theorem **localPrime_frobenius_periodT** : P.φB P.tB = (l : ℚ_[l]) • P.tB

theorem **phiB_actQ** (g : GQ l) (b : P.R.B) : P.φB (P.actQ g b) = P.actQ g (P.φB b)

theorem **phiB_injective** : Function.Injective P.φB

theorem **isUnit_tB** : IsUnit P.tB

theorem **nontrivial_B** : Nontrivial P.R.B

@[simp] theorem **stHom_C** (f : P.R.B →ₐ[ℚ_[l]] P.R.B) (y : Bst P) (b : P.R.B) : stHom P f y (C b) = C (f b)

@[simp] theorem **stHom_X** (f : P.R.B →ₐ[ℚ_[l]] P.R.B) (y : Bst P) : stHom P f y X = y

theorem **stHom_apply** (f : P.R.B →ₐ[ℚ_[l]] P.R.B) (y p : Bst P) : stHom P f y p = (p.map (f : P.R.B →+* P.R.B)).comp y

theorem **derivative_stHom** (f : P.R.B →ₐ[ℚ_[l]] P.R.B) (y p : Bst P) : derivative (stHom P f y p) = derivative y * stHom P f y (derivative p)

theorem **stHom_ext** {F G : Bst P →ₐ[ℚ_[l]] Bst P} (hC : ∀ b, F (C b) = G (C b)) (hX : F X = G X) : F = G

@[simp] theorem **actSt_C** (g : GQ l) (b : P.R.B) : actSt P g (C b) = C (P.actQ g b)

@[simp] theorem **actSt_X** (g : GQ l) : actSt P g X = X + C ((kummer P g : ℚ_[l]) • P.tB)

@[simp] theorem **phiSt_C** (b : P.R.B) : phiSt P (C b) = C (P.φB b)

@[simp] theorem **phiSt_X** : phiSt P X = (l : Bst P) * X

@[simp] theorem **monSt_apply** (p : Bst P) : monSt P p = -derivative p

theorem **phiSt_actSt** (g : GQ l) (p : Bst P) : phiSt P (actSt P g p) = actSt P g (phiSt P p)

theorem **monSt_actSt** (g : GQ l) (p : Bst P) : monSt P (actSt P g p) = actSt P g (monSt P p)

theorem **monSt_phiSt** (p : Bst P) : monSt P (phiSt P p) = (l : ℚ_[l]) • phiSt P (monSt P p)

theorem **monSt_C_mul** (b : P.R.B) (p : Bst P) : monSt P (C b * p) = C b * monSt P p

theorem **exists_monSt_pow_eq_zero** (p : Bst P) : ∃ n : ℕ, (monSt P ^ n) p = 0

theorem **monSt_eq_zero_iff** (p : Bst P) : monSt P p = 0 ↔ ∃ b, p = C b

theorem **phiSt_injective** : Function.Injective (phiSt P)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MSmoothOfWeightIffIsCInf

theorem **cxSt_cx** : (cxSt F).cx = Var.cx

theorem **cxSt_pn** : (cxSt F).pn = Var.pn

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MapSeqIndAll

theorem **ofCoord_good** {d : ℕ} {L : Submodule (𝒪 K) (Fin d → K)} {X : InfinitePlace K → Matrix (Fin d) (Fin d) ℂ} (hL : Submodule.IsLattice K L) (hX : ∀ v, IsUnit (X v).det) : (HLat.ofCoord L X).Good

theorem **unitLat_good** (n : ℕ) : (unitLat K n).Good

theorem **unitMap_b** (σ : K →+* ℂ) (W : Type) [NormedAddCommGroup W] [InnerProductSpace ℂ W] [FiniteDimensional ℂ W] (k : Fin (Module.finrank ℂ W)) : unitMap σ W ((unitLat K (Module.finrank ℂ W)).b k) = stdOrthonormalBasis ℂ W k

theorem **unitMap_basis** (σ : K →+* ℂ) (W : Type) [NormedAddCommGroup W] [InnerProductSpace ℂ W] [FiniteDimensional ℂ W] : LinearIndependent ℂ (fun k => unitMap σ W ((unitLat K (Module.finrank ℂ W)).b k)) ∧ Submodule.span ℂ (Set.range fun k => unitMap σ W ((unitLat K (Module.finrank ℂ W)).b k)) = ⊤

theorem **hLatTie_castV** {d : ℕ} {L : Submodule (𝒪 K) (Fin d → K)} {X : InfinitePlace K → Matrix (Fin d) (Fin d) ℂ} {A : HLat K} (h : HLat.ofCoord L X = A) : HLatTie A L X (castV h)

theorem **latSeq_succ** (i : Fin m) : latSeq M X W ((i : ℕ) + 1) = HLat.ofCoord (M i) (X i)

theorem **latSeq_off** {j : ℕ} (h : j = 0 ∨ m < j) : latSeq M X W j = unitLat K (Module.finrank ℂ (W j))

theorem **mapSeq_succ** (i : Fin m) : mapSeq M X W f ((i : ℕ) + 1) = (f i).comp (castV (latSeq_succ M X W i)).toLinearMap

theorem **mapSeq_off** {j : ℕ} (h : j = 0 ∨ m < j) : mapSeq M X W f j = (unitMap σ (W j)).comp (castV (latSeq_off M X W h)).toLinearMap

theorem **castV_ind** {H : Type} [AddCommGroup H] [Module ℂ H] (P : ∀ A : HLat K, (A.V →ₗ[σ] H) → Prop) {A B : HLat K} (h : A = B) {g : B.V →ₗ[σ] H} (hg : P B g) : P A (g.comp (castV h).toLinearMap)

theorem **mapSeq_ind** (P : ∀ (j : ℕ) (A : HLat K), (A.V →ₗ[σ] W j) → Prop) (hP : ∀ i : Fin m, P ((i : ℕ) + 1) (HLat.ofCoord (M i) (X i)) (f i)) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ m) : P j (latSeq M X W j) (mapSeq M X W f j)

theorem **mapSeq_ind_all** (P : ∀ (j : ℕ) (A : HLat K), (A.V →ₗ[σ] W j) → Prop) (hP : ∀ i : Fin m, P ((i : ℕ) + 1) (HLat.ofCoord (M i) (X i)) (f i)) (h0 : ∀ j, P j (unitLat K (Module.finrank ℂ (W j))) (unitMap σ (W j))) (j : ℕ) : P j (latSeq M X W j) (mapSeq M X W f j)

theorem **map_castV** {H : ℕ → HLat K} {g : ∀ j, (H j).V →ₗ[σ] W j} (hH : H = latSeq M X W) (hg : HEq g (mapSeq M X W f)) (i : Fin m) (h : HLat.ofCoord (M i) (X i) = H ((i : ℕ) + 1)) (x : Fin (d i) → K) : g ((i : ℕ) + 1) (castV h x) = f i x

theorem **mjH_succ** (i : Fin D.R.m) : mjH Z ((i : ℕ) + 1) = HLat.ofCoord ((s0Of R F).M i) ((s0Of R F).XW i)

theorem **mjH_off** {j : ℕ} (h : j = 0 ∨ D.R.m < j) : mjH Z j = unitLat D.what₂.LL (Module.finrank ℂ (D.what₂.Mj j))

theorem **mjH_good** (hH : Holds R F) (j : ℕ) : (mjH Z j).Good

theorem **jM_basis** (hH : Holds R F) (j : ℕ) : LinearIndependent ℂ (fun k => jM Z j ((mjH Z j).b k)) ∧ Submodule.span ℂ (Set.range fun k => jM Z j ((mjH Z j).b k)) = ⊤

theorem **jM_lat** (hP : Pins R F cG) (hH : Holds R F) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (x : (mjH Z j).V) : x ∈ (mjH Z j).L ↔ ∃ c : D.what₂.Mj j → 𝒪 D.what₂.LL, jM Z j x = ∑ v ∈ (D.what₂.MjLat j).gen, D.what₂.embLL ((c v : 𝒪 D.what₂.LL) : D.what₂.LL) • v

theorem **mjLat_sub** (hP : Pins R F cG) (hH : Holds R F) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : ∀ x ∈ (D.what₂.MjLat j).gen, x ∈ Set.range (jM Z j)

theorem **jM_metric** (hH : Holds R F) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (x : (mjH Z j).V) : Formalism.HermitianLatticesTheory.cnorm (Formalism.HermitianLatticesTheory.fcoord (mjH Z j).b (mjH Z j).X (InfinitePlace.mk D.what₂.embLL) x) = ‖jM Z j x‖

theorem **mjHTie_of_eq** (Xd : DivX D) (h : Xd.MjH = mjH Z) (hj : HEq Xd.jM (jM Z)) : MjHTie D A3.A Xd (A3.indefDown (s0Of R F)) Z.jW (mjIds Z Xd h) where

theorem **exists_mjHTie** (Xd : DivX D) (h : Xd.MjH = mjH Z) (hj : HEq Xd.jM (jM Z)) : ∃ JM, MjHTie D A3.A Xd (A3.indefDown (s0Of R F)) Z.jW JM

theorem **mjHTie_withMj** (hP : Pins R F cG) (hH : Holds R F) (Xd : DivX D) (iM : ∀ j c, (mjH Z j).V ↛ₗ[D.what₂.LL] Xd.geo.Fn (Xd.Yv j c)) (hiM : ∀ j c x, iM j c x ∈ Xd.geo.H0 (Xd.Yv j c) (Xd.Lw j c (DV.kapW D j))) : MjHTie D A3.A (withMj Z hP hH Xd iM hiM) (A3.indefDown (s0Of R F)) Z.jW (mjIds Z _ rfl)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemEqnSetOmegaCoeff

theorem **exists_mem_ptsIn** {U : Y.X.Opens} (h : genericPoint Y.X ∈ U) : ∃ y : Y.pts σ, y ∈ ptsIn Y.str σ U

theorem **val_cx** (a : Y.Fn) (y : Y.pts σ) : val y (Y.cx σ a) = ratVal a y

theorem **isOpen_eqnSet** (Δ : Y.Dv) (g : Y.Fnˣ) : IsOpen (eqnSet σ Δ g)

theorem **mem_eqnSet** (Δ : Y.Dv) (y : Y.pts σ) : y ∈ eqnSet σ Δ (eqn Δ y)

theorem **mem_eqnSet_one** : y ∈ eqnSet σ Δ 1 ↔ base y ∉ Y.supp Δ

theorem **fval_one** (f : Y.FnC σ) (y : Y.pts σ) : fval 1 f y = val y f

theorem **fval_cx** (a : Y.Fn) (y : Y.pts σ) : fval g (Y.cx σ a) y = ratVal (a * (g : Y.Fn)) y

theorem **tv_ne_zero** (h : y ∈ eqnSet σ Δ g) (h' : y ∈ eqnSet σ Δ g') : tv g g' y ≠ 0

theorem **mul_cx_eq** (f : Y.FnC σ) (g g' : Y.Fnˣ) : f * Y.cx σ g' = Y.cx σ (g' / g) * (f * Y.cx σ g)

theorem **fval_trans** (h : y ∈ eqnSet σ Δ g) (h' : y ∈ eqnSet σ Δ g') (f : Y.FnC σ) : fval g' f y = tv g g' y * fval g f y

theorem **tv_self** (g : Y.Fnˣ) (y : Y.pts σ) : tv g g y = 1

theorem **tv_mul_mul** {g₁ g₁' : Y.Fnˣ} (h : y ∈ eqnSet σ Δ g) (h' : y ∈ eqnSet σ Δ g') : tv (g * g₁) (g' * g₁') y = tv g g' y * tv g₁ g₁' y

theorem **tv_mul** {g'' : Y.Fnˣ} (h : y ∈ eqnSet σ Δ g) (h' : y ∈ eqnSet σ Δ g') : tv g g' y * tv g' g'' y = tv g g'' y

theorem **tv_pow** (h : y ∈ eqnSet σ Δ g) (h' : y ∈ eqnSet σ Δ g') (k : ℕ) : tv (g ^ k) (g' ^ k) y = tv g g' y ^ k

theorem **holoOn_tv** (Δ : Y.Dv) (g g' : Y.Fnˣ) : (Y.an σ).HoloOn (eqnSet σ Δ g ∩ eqnSet σ Δ g') (tv g g')

theorem **secReg_iff** (h : y ∈ eqnSet σ Δ g) : SecReg Δ f y ↔ RegC y (f * Y.cx σ g)

theorem **secZero_iff** (h : y ∈ eqnSet σ Δ g) : SecZero Δ f y ↔ SecReg Δ f y ∧ fval g f y = 0

theorem **secReg_cx_iff** {a : Y.Fn} : SecReg Δ (Y.cx σ a) y ↔ Δ.RegularAt (base y) a

theorem **secZero_cx_iff** {a : Y.Fn} : SecZero Δ (Y.cx σ a) y ↔ Δ.VanishesAt (base y) a

theorem **secReg_cx_of_mem** {a : Y.Fn} (h : a ∈ Y.H0 Δ) (y : Y.pts σ) : SecReg Δ (Y.cx σ a) y

theorem **hasVal_sum_smul** {ι : Type*} [Fintype ι] {e : ι → Y.FnC σ} (h : y ∈ eqnSet σ Δ g) (he : ∀ i, SecReg Δ (e i) y) (c : ι → ℂ) : HasVal y ((∑ i, c i • e i) * Y.cx σ g) (∑ i, c i * fval g (e i) y)

theorem **SecReg.mul** (h : SecReg Δ f y) (h' : SecReg Δ' f' y) : SecReg (Δ + Δ') (f * f') y

theorem **fval_mul** (hg : y ∈ eqnSet σ Δ g) (hg' : y ∈ eqnSet σ Δ' g') (h : SecReg Δ f y) (h' : SecReg Δ' f' y) : fval (g * g') (f * f') y = fval g f y * fval g' f' y

theorem **fval_pow** (hg : y ∈ eqnSet σ Δ g) (h : SecReg Δ f y) (k : ℕ) : fval (g ^ k) (f ^ k) y = fval g f y ^ k

theorem **isOpen_secReg** (Δ : Y.Dv) (f : Y.FnC σ) : IsOpen {y | SecReg Δ f y}

theorem **holoOn_fval** (Δ : Y.Dv) (g : Y.Fnˣ) (f : Y.FnC σ) : (Y.an σ).HoloOn ({y | SecReg Δ f y} ∩ eqnSet σ Δ g) (fval g f)

theorem **mem_posSet_iff** (hg : y ∈ eqnSet σ Δ g) : y ∈ m.posSet ↔ 0 < m.h g y

theorem **h_eq_exp** (hy : y ∈ m.posSet) (hg : y ∈ eqnSet σ Δ g) : m.h g y = Real.exp (-m.wt g y)

theorem **wt_trans** (hy : y ∈ m.posSet) (hg : y ∈ eqnSet σ Δ g) (hg' : y ∈ eqnSet σ Δ g') : m.wt g' y = m.wt g y + Real.log (‖tv g g' y‖ ^ 2)

theorem **ofCover_h** {i : ι} (hi : y ∈ S i) : (ofCover (Δ

theorem **smoothOn_of_cover** {S : Set (Y.pts σ)} (hS : IsOpen S) (h0 : S ⊆ m.posSet) {ι : Type} {T : ι → Set (Y.pts σ)} (hT : ∀ i, IsOpen (T i)) (hc : ∀ y, ∃ i, y ∈ T i) {e : ι → Y.Fnˣ} (sub : ∀ i, T i ⊆ eqnSet σ Δ (e i)) (h : ∀ i, (Y.an σ).SmoothOn (S ∩ T i) (m.wt (e i))) : m.SmoothOn S

theorem **cover_pair** (Δ Δ' : Y.Dv) (y : Y.pts σ) : ∃ i : Y.Fnˣ × Y.Fnˣ, y ∈ eqnSet σ Δ i.1 ∩ eqnSet σ Δ' i.2

theorem **tv_eq_gv** (hp : Presents Δ L r) {i j : L.ι} (hi : y ∈ ptsIn Y.str σ (L.U i)) (hj : y ∈ ptsIn Y.str σ (L.U j)) : tv (r j) (r i) y = gv Y.str σ L i j y

theorem **ptsIn_subset_eqnSet** (hp : Presents Δ L r) (i : L.ι) : ptsIn Y.str σ (L.U i) ⊆ eqnSet σ Δ (r i)

theorem **ofWeight_h** (hp : Presents Δ L r) (φ : Weight Y.str σ L) {i : L.ι} (hi : y ∈ ptsIn Y.str σ (L.U i)) : (ofWeight Y σ Δ L r hp φ).h (r i) y = Real.exp (-φ.φ i y)

theorem **mAdd_h** (hg : y ∈ eqnSet σ Δ g) (hg' : y ∈ eqnSet σ Δ' g') : (mAdd Y σ Δ Δ' m m').h (g * g') y = m.h g y * m'.h g' y

theorem **mPow_h** (hg : y ∈ eqnSet σ Δ g) (k : ℕ) : (mPow Y σ Δ k m).h (g ^ k) y = m.h g y ^ k

theorem **fsMet_h** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} (he : ∀ k, SecReg Δ (e k) y) (g : Y.Fnˣ) : (fsMet Y σ Δ e).h g y = (∑ k, ‖fval g (e k) y‖ ^ 2)⁻¹

theorem **mem_posSet_fsMet** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} : y ∈ (fsMet Y σ Δ e).posSet ↔ (∀ k, SecReg Δ (e k) y) ∧ ∃ k, ¬ SecZero Δ (e k) y

theorem **posSet_mAdd** : (mAdd Y σ Δ Δ' m m').posSet = m.posSet ∩ m'.posSet

theorem **posSet_subset_mPow** (k : ℕ) : m.posSet ⊆ (mPow Y σ Δ k m).posSet

theorem **wt_mAdd** (hy : y ∈ m.posSet) (hy' : y ∈ m'.posSet) (hg : y ∈ eqnSet σ Δ g) (hg' : y ∈ eqnSet σ Δ' g') : (mAdd Y σ Δ Δ' m m').wt (g * g') y = m.wt g y + m'.wt g' y

theorem **wt_mPow** (hg : y ∈ eqnSet σ Δ g) (k : ℕ) : (mPow Y σ Δ k m).wt (g ^ k) y = k * m.wt g y

theorem **wt_fsMet** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} (he : ∀ k, SecReg Δ (e k) y) (g : Y.Fnˣ) : (fsMet Y σ Δ e).wt g y = Real.log (∑ k, ‖fval g (e k) y‖ ^ 2)

theorem **pn_eq** (hg : y ∈ eqnSet σ Δ g) : pn Y σ Δ m f y = ‖fval g f y‖ ^ 2 * m.h g y

theorem **pn_eq_exp** (hy : y ∈ m.posSet) (hg : y ∈ eqnSet σ Δ g) : pn Y σ Δ m f y = ‖fval g f y‖ ^ 2 * Real.exp (-m.wt g y)

theorem **pn_of_notMem_supp** (hy : y ∈ m.posSet) (h : base y ∉ Y.supp Δ) : pn Y σ Δ m f y = ‖val y f‖ ^ 2 * Real.exp (-m.wt 1 y)

theorem **pnK_of_notMem_supp** (hy : y ∈ m.posSet) (h : base y ∉ Y.supp Δ) : pnK Y σ Δ m f y = ‖val y f‖ ^ 2 * Real.exp (-m.wt 1 y)

theorem **pn_nonneg** : 0 ≤ pn Y σ Δ m f y

theorem **pn_frame** (hg : y ∈ eqnSet σ Δ g) : pn Y σ Δ m (Y.cx σ (g⁻¹ : Y.Fnˣ)) y = m.h g y

theorem **pn_eq_zero_iff** (hy : y ∈ m.posSet) (hf : SecReg Δ f y) : pn Y σ Δ m f y = 0 ↔ SecZero Δ f y

theorem **pn_mul** {a : Y.FnC σ} (ha : RegC y a) (hf : SecReg Δ f y) : pn Y σ Δ m (a * f) y = ‖val y a‖ ^ 2 * pn Y σ Δ m f y

theorem **pn_smul** (c : ℂ) (hf : SecReg Δ f y) : pn Y σ Δ m (c • f) y = ‖c‖ ^ 2 * pn Y σ Δ m f y

theorem **pn_mAdd** (hf : SecReg Δ f y) (hf' : SecReg Δ' f' y) : pn Y σ _ (mAdd Y σ Δ Δ' m m') (f * f') y = pn Y σ Δ m f y * pn Y σ Δ' m' f' y

theorem **pn_mPow** (hf : SecReg Δ f y) (k : ℕ) : pn Y σ _ (mPow Y σ Δ k m) (f ^ k) y = pn Y σ Δ m f y ^ k

theorem **pn_fsMet** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} (he : ∀ k, SecReg Δ (e k) y) (hg : y ∈ eqnSet σ Δ g) : pn Y σ Δ (fsMet Y σ Δ e) f y = ‖fval g f y‖ ^ 2 / ∑ k, ‖fval g (e k) y‖ ^ 2

theorem **pn_fsMet_le** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} (he : ∀ k, SecReg Δ (e k) y) (c : κ → ℂ) : pn Y σ Δ (fsMet Y σ Δ e) (∑ k, c k • e k) y ≤ ∑ k, ‖c k‖ ^ 2

theorem **sum_pn_fsMet** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} (hy : y ∈ (fsMet Y σ Δ e).posSet) : ∑ k, pn Y σ Δ (fsMet Y σ Δ e) (e k) y = 1

theorem **pn_ofWeight** {L : LB Y.X} {r : L.ι → Y.Fnˣ} (hp : Presents Δ L r) (φ : Weight Y.str σ L) {i : L.ι} (hi : y ∈ ptsIn Y.str σ (L.U i)) : pn Y σ Δ (ofWeight Y σ Δ L r hp φ) f y = ‖fval (r i) f y‖ ^ 2 * Real.exp (-φ.φ i y)

theorem **pn_ofWeight_rat** {L : LB Y.X} {r : L.ι → Y.Fnˣ} (hp : Presents Δ L r) (φ : Weight Y.str σ L) (s : L.Sec) (y : Y.pts σ) : pn Y σ Δ (ofWeight Y σ Δ L r hp φ) (Y.cx σ (hp.rat s)) y = φ.nsq (secVal Y.str σ s) y

theorem **mSmooth.posSet_eq** (h : mSmooth Y σ Δ m) : m.posSet = univ

theorem **Met.SmoothOn.mAdd** {S : Set (Y.pts σ)} (hS : IsOpen S) (h : m.SmoothOn S) (h' : m'.SmoothOn S) : (mAdd Y σ Δ Δ' m m').SmoothOn S

theorem **Met.SmoothOn.mPow** {S : Set (Y.pts σ)} (hS : IsOpen S) (h : m.SmoothOn S) (k : ℕ) : (mPow Y σ Δ k m).SmoothOn S

theorem **smoothOn_fsMet** {κ : Type} [Fintype κ] (e : κ → Y.FnC σ) {S : Set (Y.pts σ)} (hS : IsOpen S) (h : S ⊆ (fsMet Y σ Δ e).posSet) : (fsMet Y σ Δ e).SmoothOn S

theorem **mSmooth_fsMet_iff** {κ : Type} [Fintype κ] {e : κ → Y.FnC σ} : mSmooth Y σ Δ (fsMet Y σ Δ e) ↔ ∀ y, (∀ k, SecReg Δ (e k) y) ∧ ∃ k, ¬ SecZero Δ (e k) y

theorem **mSmooth_fsMet_cx** {κ : Type} [Fintype κ] {b : κ → Y.Fn} (h : Y.bpf Δ (range b)) : mSmooth Y σ Δ (fsMet Y σ Δ fun k => Y.cx σ (b k))

theorem **mSmooth_ofWeight_iff** {L : LB Y.X} {r : L.ι → Y.Fnˣ} (hp : Presents Δ L r) (φ : Weight Y.str σ L) : mSmooth Y σ Δ (ofWeight Y σ Δ L r hp φ) ↔ φ.IsSmooth (Y.an σ)

theorem **mSmooth.isSmooth_toWeight** (h : mSmooth Y σ Δ m) {L : LB Y.X} {r : L.ι → Y.Fnˣ} (hp : Presents Δ L r) : (m.toWeight h.posSet_eq hp).IsSmooth (Y.an σ)

theorem **pn_ofWeight_toWeight** (hm : m.posSet = univ) {L : LB Y.X} {r : L.ι → Y.Fnˣ} (hp : Presents Δ L r) : pn Y σ Δ (ofWeight Y σ Δ L r hp (m.toWeight hm hp)) = pn Y σ Δ m

theorem **Met.SmoothOn.continuousOn_pn** {S : Set (Y.pts σ)} (hS : IsOpen S) (h : m.SmoothOn S) (hf : ∀ y ∈ S, SecReg Δ f y) : ContinuousOn (pn Y σ Δ m f) S

theorem **omega0_eq** (c : AlgChart Y.str σ Y.dim) (h : genericPoint Y.X ∈ c.U) : Y.omega0 = (omegaCoeff c : Y.Fn) • formOf c.U h (dOn Y.str c.x)

theorem **mem_eqnSet_omegaCoeff** (c : AlgChart Y.str σ Y.dim) (hy : y ∈ c.e.source) : y ∈ eqnSet σ Y.KY (omegaCoeff c)

theorem **tv_omegaCoeff** (c c' : AlgChart Y.str σ Y.dim) (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : tv (omegaCoeff c') (omegaCoeff c) y = transDet c.e c'.e y

theorem **vol_apply** (c : AlgChart Y.str σ Y.dim) {A : Set (Y.pts σ)} (hA : MeasurableSet A) (h : A ⊆ c.e.source) : vol Y σ A = ∫⁻ v in c.e '' A, ENNReal.ofReal (2 ^ Y.dim * ‖val (c.e.symm v) (Y.cx σ (omegaCoeff c))‖ ^ 2)

theorem **_root_.GlobalTheory.ProjVar.PosCurrent.congr** {D : ℕ} {O : Set (V D)} {u u' : V D → ℝ} (hO : IsOpen O) (h : PosCurrent O u) (e : ∀ᵐ v, v ∈ O → u v = u' v) : PosCurrent O u'

theorem **mPsh.congr** {ψ ψ' : Y.pts σ → ℝ} (h : mPsh Y σ Δ m ψ) (e : ∀ c : AlgChart Y.str σ Y.dim, ∀ᵐ v, v ∈ c.e.target → ψ (c.e.symm v) = ψ' (c.e.symm v)) : mPsh Y σ Δ m ψ'

theorem **mem_chartAt** (y : Y.pts σ) : y ∈ (chartAt y).e.source

theorem **differentiableAt_fval** (hc : y ∈ c.e.source) (hg : y ∈ eqnSet σ Δ g) (hs : ∀ i, SecReg Δ (s i) y) : DifferentiableAt ℂ (fun v => (WithLp.toLp 2 fun i => fval g (s i) (c.e.symm v) : V Y.dim)) (c.e y)

theorem **jac_chart** (hc : y ∈ c.e.source) (hc' : y ∈ c'.e.source) (hg : y ∈ eqnSet σ Δ g) (hs : ∀ i, SecReg Δ (s i) y) : jac c g s y = transDet c.e c'.e y * jac c' g s y

theorem **jac_frame** (hc : y ∈ c.e.source) (hg : y ∈ eqnSet σ Δ g) (hg' : y ∈ eqnSet σ Δ g') (hs : ∀ i, SecZero Δ (s i) y) : jac c g' s y = tv g g' y ^ Y.dim * jac c g s y

theorem **jac_ne_zero_iff** (hc : y ∈ c.e.source) (hg : y ∈ eqnSet σ Δ g) (hs : ∀ i, SecZero Δ (s i) y) : jac (chartAt y) (eqn Δ y) s y ≠ 0 ↔ jac c g s y ≠ 0

theorem **agreeOn_iff** {G G' : Y.FnC σ} {S : Set (Y.pts σ)} (h : regOn Y σ Δ G S) (h' : regOn Y σ Δ G' S) : agreeOn Y σ Δ G G' S ↔ ∀ y ∈ S, fval (eqn (Y.KY + Δ) y) G y = fval (eqn (Y.KY + Δ) y) G' y

theorem **resNorm_eq** (hc : y ∈ c.e.source) (hg : y ∈ eqnSet σ Δ g) (hg' : y ∈ eqnSet σ Δ' g') (hs : ∀ i, SecZero Δ' (s i) y) (G : Y.FnC σ) : resNorm Y σ Δ m Δ' m' s G y = ‖fval (omegaCoeff c * g) G y‖ ^ 2 * m.h g y / (‖jac c g' s y‖ ^ 2 * m'.h g' y ^ Y.dim)

theorem **resNorm_nonneg** (G : Y.FnC σ) : 0 ≤ resNorm Y σ Δ m Δ' m' s G y

theorem **mSmooth_projMet** : mSmooth _ σ _ (projMet F σ n)

theorem **jac_coord** {Y : Var F} (c : AlgChart Y.str σ Y.dim) {y : Y.pts σ} (hy : y ∈ c.e.source) (h : genericPoint Y.X ∈ c.U) : Var.jac c 1 (fun i => Y.cx σ (fnOf c.U h (c.x i))) y = 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemLevelOfLocalDepth

theorem **mem_iff_trace** {K G : Type} [Field K] [CharZero K] [Group G] [Fintype G] (X : Finset (G → K)) (hX : ∀ χ ∈ X, IsAbsIrrChar χ) (χ : G → K) : χ ∈ X ↔ IsAbsIrrChar χ ∧ ∑ g, (idemOf X).coeff g * χ g = χ 1

theorem **mem_vGs_iff** (h : I.Ok) (P : HeightOneSpectrum (𝒪 B.fields.E)) : P ∈ (B.primes px).vGs ↔ ∃ k, I.C.P.Div k P

theorem **ixOf_empty** (t : TupU I ∅) : ixOf I ∅ t = I.mu0

theorem **exists_under_of_selHalf** (h : SelHalf I) (p : ℕ) (hp : p.Prime) : ∃ 𝔭 : HeightOneSpectrum (𝒪 B.host.K), ∀ a, ∀ P : HeightOneSpectrum (𝒪 B.host.E), P ∈ I.Q a → ((p : ℕ) : 𝒪 B.host.E) ∈ P.asIdeal → P.under (𝒪 B.host.K) = 𝔭

theorem **mem_level_of_local_depth** (g : Gc B px) : g ∈ lev0 I ↔ ∀ w, g w ∈ unip (kP w.1) B.R.r

theorem **baseLevel_eq_unipotent** : lev0 I = (Formalism.L2.Col.rP0 (R

theorem **isGenericPsi_psiC** (h : I.Ok) : IsGenericPsi (B.primes px).vGs (psiC I)

theorem **embLL_xiValOf_of_ok** (h : I.Ok) (a : I.ι) (P : HeightOneSpectrum (𝒪 B.fields.E)) : B.embLL (xiValOf I (I.tup a) P : B.LL) = (((I.xi a).val P : ℂˣ) : ℂ)

theorem **valsIn_iff** (h : I.Ok) : ValsIn I ↔ ∀ a, ∀ P ∈ I.Q a, ∃ z : B.LL, IsPrimitiveRoot z (Nat.card (kP P))

theorem **exists_levOf_of** (h : I.Ok) (U : Set I.ι) (L : Formalism.IndefR.Links.LevCh X.toD7 (TupU I U)) (hC : L.C = Top.Fr.at X I.C) (hx : L.xiH = fun t => I.xi (ixOf I U t)) (hQ : L.Q = fun t => I.Q (ixOf I U t)) (hL : L.Is (mu0F (X

theorem **exists_levOf_empty** (h : I.Ok) : ∃ L, LevOf X I ∅ L

theorem **grpAt_iff** (h : I.Ok) : GrpAt X I ↔ ∀ k v, I.C.P.Div k (X.toD7.naut.plGw k v)

theorem **grpAt_of_grpTrue** (h : I.Ok) (hN : Top.GrpTrue X.N I.C) : GrpAt X I

theorem **goodSet_of_idem** (eR : Gc X.B px ≃* G0g X.toD7.nd) (Xg : Finset (Gc X.B px → X.B.LL)) (hX : ∀ χ ∈ Xg, IsAbsIrrChar χ) (he : ∀ g, X.B.embLL ((idemOf Xg).coeff g) = X.toD7.ntypes.e0.coeff (eR g)) : GoodSet X (px

theorem **GoodSet.nonempty** {eR : Gc X.B px ≃* G0g X.toD7.nd} {Xg : Finset (Gc X.B px → X.B.LL)} (h : GoodSet X (px

theorem **isIdG_iff** (eG : Gc X.B px ≃* Gc X.B px) : IsPlacewiseIdentity X (px

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemMaximalIdealBall

theorem **mem_maximalIdeal_ball** (a : 𝔯(intO C)) : a ∈ maximalIdeal 𝔯(intO C) ↔ ‖(a : C)‖ < 1

theorem **nonempty_ext** (y : DVP ct) : Nonempty (Ext y)

theorem **vpi_pos** (e : Ext y) : 0 < vpi e

theorem **img_spz** (χ : Chart Z y e) : ct.spz (imgU χ) (imgM χ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemNonzeroPlacesOfResidueChar

theorem **fvalP_eq** {D : D7} (L : SizeData₂ D) (j p : ℕ) : FvalP L j p = FK L.sc D j p

theorem **fk_zero** (t : RowsT K w Z Wt p0) (v : B.places.Pl) (hv : v ∉ Wt) (j : Fin B.R.m) : FK K.s (w.D Z) (j.1 + 1) (B.places.resChar v) = 0

theorem **fval_zero** (t : RowsT K w Z Wt p0) (L : SizeData₂ (w.D Z)) (hL : L.sc = K.s) (v : B.places.Pl) (hv : v ∉ Wt) (j : Fin B.R.m) : Fval L (j.1 + 1) v = 0

theorem **mj_same** (Z : R3.ReadL R F X0 Y) {X X' : DivX D} (h : MjAre Z X) (h' : MjAre Z X') (i : Fin D.R.m) : ∃ e : (X.MjH (i.1 + 1)).V ≃ₗ[D.what₂.LL] (X'.MjH (i.1 + 1)).V, (∀ x, X'.jM _ (e x) = X.jM _ x) ∧ ∀ x, e x ∈ (X'.MjH (i.1 + 1)).L ↔ x ∈ (X.MjH (i.1 + 1)).L

theorem **ijIdx_same** (Z : R3.ReadL R F X0 Y) {X X' : DivX D} (h : MjAre Z X) (h' : MjAre Z X') (i : Fin D.R.m) (μ : D.what₂.Tup) (v : D.what₂.places.Pl) : Idx.IjIdx X' μ (i.1 + 1) v = Idx.IjIdx X μ (i.1 + 1) v

theorem **fTypeG_ne** {D : D7} (h : DV.KapDef D) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : FTypeG D j ≠ 0

theorem **type_mem** (t : RowsT K w Z Wt p0) (v : B.places.Pl) (i : Fin 2) (hv : B.places.resChar v = B.host.g i) : v ∈ Wt

theorem **fk_forced** (h : Forced K B T) (Z : w.Lt) (j : ℕ) (v : B.places.Pl) (hv : FK K.s (w.D Z) j (B.places.resChar v) ≠ 0) : B.places.resChar v ∈ T.pl

theorem **step_start** (h2b : StandardLattice.SiegelMassRatioCM) : Step KPx (fun _ B _ T => T = TwDat.untw B T.pl) fun K B f T => ∃ w : ColW B f T, Start K B f T w

theorem **below_iff** (D : D7) (v v' : D.what₂.places.Pl) : Below D D v v' ↔ v = v'

theorem **builtC_forget** (𝒮 : Slots) (c : V4.Core K (w.st Z) f T) (h : BuiltC 𝒮 K B f T w Z c) : BuiltC trivialDecorations K B f T w Z c

theorem **resChar_eq_of_same** (B : Base) (v v' : B.places.Pl) (h : ∀ x ∈ B.LL0, 0 ≤ B.places.val v x ↔ 0 ≤ B.places.val v' x) : B.places.resChar v = B.places.resChar v'

theorem **datGood_vol** (h : DatGood K B f T w) : ∃ Z : w.Lt, Formalism.L2.Theta.VolTieG (w.D Z)

theorem **mem_nonzero_places_of_residue_char** {hX : B.fields.Ok} (z : R3.Dat (w.st Z) w.I hX T Wt) (hk : DV.KapDef (w.D Z)) (v : B.places.Pl) (i : Fin 2) (hv : B.places.resChar v = B.host.g i) : v ∈ z.W1 (F0K K.s (w.D Z))

theorem **datGood_g** (h : DatGood K B f T w) (v : B.places.Pl) (i : Fin 2) (hv : B.places.resChar v = B.host.g i) : B.host.g i ∈ T.pl

theorem **coh_expTw** (pl : Finset ℕ) (n : ℕ) (hc) (hr : ∀ p ∈ pl, ∃ z : B.LL, IsPrimitiveRoot z (p ^ n)) : Coh B (expTw B pl n hc)

theorem **not_coh_expTw** (hc) (v : B.places.Pl) : ¬ Coh B (expTw B {B.places.resChar v} (Module.finrank ℚ B.LL + 1) hc)

theorem **resChar_conj_base** (Z : w.Lt) (v : B.places.Pl) : B.places.resChar (B.places.conj v) = B.places.resChar v

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemPlacesOverOfIsConjPl

theorem **val_lt_iff** {A B : ValuationSubring Ω} (c : Ω ≃+* Ω) (hc : ∀ x, x ∈ A ↔ c x ∈ B) (y : Ω) : B.valuation (c y) < 1 ↔ A.valuation y < 1

theorem **mem_Ine_of_transport** {A B : ValuationSubring Ω} (c : Ω ≃+* Ω) (hc : ∀ x, x ∈ A ↔ c x ∈ B) {t γ : G E Ω} (hγ : ∀ x, γ (c x) = c (t x)) (ht : t ∈ Ine (F

theorem **exists_transport_elt** (s : E ≃+* E) (c : Ω ≃+* Ω) (hs : ∀ x, c (algebraMap E Ω x) = algebraMap E Ω (s x)) (t : G E Ω) : ∃ γ : G E Ω, ∀ x, γ (c x) = c (t x)

theorem **coe_conjInt** (x : 𝒪 H.E) : ((conjInt H x : 𝒪 H.E) : H.E) = H.conj (x : H.E)

theorem **isConjPl_iff** {w w' : PlE H} : IsConjPl H w w' ↔ ∀ x, x ∈ w.asIdeal ↔ conjInt H x ∈ w'.asIdeal

theorem **exists_isConjPl** (w : PlE H) : ∃ w', IsConjPl H w w'

theorem **eq_of_isConjPl** {w₀ w₁ w : PlE H} (h₀ : IsConjPl H w₀ w) (h₁ : IsConjPl H w₁ w) : w₀ = w₁

theorem **mem_placesOver_of_isConjPl** {w₀ w : PlE H} (h : IsConjPl H w₀ w) {p : ℕ} (hp : w₀ ∈ placesOver H.E p) : w ∈ placesOver H.E p

theorem **exists_ext** (N : NData R H what₂) {w₀ w : PlE H} (h : IsConjPl H w₀ w) : ∃ c : N.Om ≃+* N.Om, (∀ x : H.E, c (algebraMap H.E N.Om x) = algebraMap H.E N.Om (H.conj x)) ∧ ∀ x, x ∈ N.plOm.1 w₀ ↔ c x ∈ N.plOm.1 w

theorem **exists_place** (D : D7) (k : Fin 2) : ∃ v w, IsConjPl D.host (D.naut.plGw k v) w ∧ ∀ v', w ≠ D.naut.plGw k v'

theorem **natDegree_labLP** {D : D7} {T : GlobalTheory.RowTie.WDTie D.nd D.naut} (hL : LabConj D T) (k v) (w : PlE D.host) (hw : IsConjPl D.host (D.naut.plGw k v) w) (a : D.naut.LP w) : (D.naut.items₂ k w a).natDegree = (T.e w a).lpTame.lpDim

theorem **roots_labLP** {D : D7} {T : GlobalTheory.RowTie.WDTie D.nd D.naut} (hL : LabConj D T) (k v) (w : PlE D.host) (hw : IsConjPl D.host (D.naut.plGw k v) w) : ∃ (c : D.nd.Om ≃+* D.nd.Om) (γ : G D.host.E D.nd.Om), γ ∈ Ine (F

theorem **exists_gamma** (D : D7) (k : Fin 2) : ∃ γ : PlE D.host → G D.host.E D.nd.Om, (∀ v, γ (D.naut.plGw k v) = D.ntypes.tg k v) ∧ ∀ v w, IsConjPl D.host (D.naut.plGw k v) w → ∃ c : D.nd.Om ≃+* D.nd.Om, (∀ x : D.host.E, c (algebraMap D.host.E D.nd.Om x) = algebraMap D.host.E D.nd.Om (D.host.conj x)) ∧ (∀ x, x ∈ D.nd.plOm.1 (D.naut.plGw k v) ↔ c x ∈ D.nd.plOm.1 w) ∧ ∀ x, γ w (c x) = c (D.ntypes.tg k v x)

theorem **exists_labTrue_labConj** : ∀ (D : D7) (hE : IsEmpty D.nd.Loc) (Rw : RowWD D), ∃ lab h A C tm tg hl, (∀ k v, A.plGw k v = D.naut.plGw k v) ∧ ∃ T : GlobalTheory.RowTie.WDTie (setAut D hE lab h A C tm tg hl).nd (setAut D hE lab h A C tm tg hl).naut, LabTrue (setAut D hE lab h A C tm tg hl) T ∧ LabConj (setAut D hE lab h A C tm tg hl) T ∧ ∃ Rs : RowSide (setAut D hE lab h A C tm tg hl), (∀ σ w, T.e w (Rs.locR σ w) = Rw.par σ w) ∧ ∀ σ w, Rs.satR σ w = Rw.sat σ w

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemPrinCongOfSm

theorem **Sm.add** (h : Sm v γ A) (h' : Sm v γ B) : Sm v γ (A + B)

theorem **Sm.sub** (h : Sm v γ A) (h' : Sm v γ B) : Sm v γ (A − B)

theorem **Sm.mul** (h : Sm v γ A) (h' : Sm v δ B) : Sm v (γ ∗ δ) (A ∗ B)

theorem **Sm.mono** (h : Sm v γ A) (hγ : γ ≤ δ) : Sm v δ A

theorem **sm_one** : Sm v 1 (1 : Matrix (Fin n) (Fin n) k)

theorem **Sm.of_sub_one** (h : Sm v γ (A − 1)) (hγ : γ ≤ 1) : Sm v 1 A

theorem **Sm.mulVec** (h : Sm v γ A) {x : Fin n → k} (hx : ∀ i, v (x i) ≤ δ) (i : Fin n) : v ((A *ᵥ x) i) ≤ γ ∗ δ

theorem **int_of_mulVec** (h : Sm v γ (A − 1)) (hγ : γ < 1) {x : Fin n → k} (hx : ∀ i, v ((A *ᵥ x) i) ≤ 1) (i : Fin n) : v (x i) ≤ 1

theorem **locM_mul** (x y : GL (Fin n) (AfE F E)) : locM P (x ∗ y) = locM P x ∗ locM P y

theorem **locM_one** : locM P (1 : GL (Fin n) (AfE F E)) = 1

theorem **mat_mul** (f g : Sp E n ≃ₗ[E] Sp E n) : mat (f ∗ g) = mat f ∗ mat g

theorem **ratM_mul** (f g : Sp E n ≃ₗ[E] Sp E n) : ratM P (f ∗ g) = ratM P f ∗ ratM P g

theorem **ratM_one** : ratM P (1 : Sp E n ≃ₗ[E] Sp E n) = 1

theorem **ratM_mulVec** (f : Sp E n ≃ₗ[E] Sp E n) (x : Sp E n) : ratM P f *ᵥ (fun i => algebraMap E (P.adicCompletion E) (x i)) = fun i => algebraMap E _ (f x i)

theorem **locM_bc** (γ : GL (Fin n) E) : locM P (Formalism.UG.bc ι γ) = ratM P (Formalism.UG.glEquiv γ)

theorem **dep_ne_zero** (c : ℕ) : dep P c ≠ 0

theorem **val_qOf_lt** : P.valuation E (qOf P : E) < 1

theorem **dep_le_one** (c : ℕ) : dep P c ≤ 1

theorem **dep_succ_lt** (c : ℕ) : dep P (c + 1) < 1

theorem **dep_succ_le** (c : ℕ) : dep P (c + 1) ≤ dep P c

theorem **mem_stdAt_val** {x : Sp E n} : x ∈ stdAt E n P ↔ ∀ i, Valued.v (algebraMap E (P.adicCompletion E) (x i)) ≤ 1

theorem **sm_glint** (k : GLint E n P) : Sm Valued.v 1 (ratM P k.1)

theorem **mem_prinCong_of_sm** {c : ℕ} (k : GLint E n P) (h : Sm Valued.v (dep P c) (ratM P k.1 − 1)) : k ∈ prinCong n P c

theorem **exists_frame** (x : GL (Fin n) (AfE F E)) (P : HeightOneSpectrum (𝒪 E)) (c : ℕ) : ∃ f : Sp E n ≃ₗ[E] Sp E n, CharValCompare.IsFr P (Formalism.UG.CS.globLat x (stdLat E n)) f ∧ Sm Valued.v (dep P c) (locM P x⁻¹ ∗ ratM P f − 1)

theorem **exists_near** (c : ℕ) {x : GL (Fin n) (AfE F E)} (hx : x ∈ UAd hc n) : ∃ p : DecoF E n ST SG Q, Near c x p

theorem **near_one** (c : ℕ) : Near c (1 : GL (Fin n) (AfE F E)) (DecoF.base E n ST SG Q)

theorem **Near.exists_smul** {x : GL (Fin n) (AfE F E)} {p p' : DecoF E n ST SG Q} (h : Near c x p) (h' : Near c x p') : ∃ k : GLQ E n Q, (∀ P, k P ∈ prinCong n P.1 c) ∧ p' = ((1, 1, k) : UF E n × GLG E n SG × GLQ E n Q) • p

theorem **Near.left** {x : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : Near c x p) (γ : URat hc n) : Near c (Formalism.UG.bc ι (γ : GL (Fin n) E) ∗ x) (((ratHom hc n γ, 1, 1) : UF E n × GLG E n SG × GLQ E n Q) • p)

theorem **Near.right** {x k : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : Near c x p) (hk : k ∈ K1 hc n ST SG) (hQ : ∀ P ∈ Q, Sm Valued.v (dep P c) (locM P k⁻¹ − 1)) : Near c (x ∗ k) p

theorem **Near.change** {x x' : GL (Fin n) (AfE F E)} (hx : x ∈ UAd hc n) (hx' : x' ∈ UAd hc n) {p₀ p : DecoF E n ST SG Q} (h : Near c x p₀) (h' : Near c x' p₀) {g : GL (Fin n) (AfE F E)} (hp : Near c (g ∗ x) p) : Near c (g ∗ x') p

theorem **exists_cong** (c : ℕ) : ∃ a : 𝒪 E, a ≠ 0 ∧ ∀ k ∈ UAd hc n, CongOne a k → CongOne a k⁻¹ → ∀ (x : GL (Fin n) (AfE F E)) (p : DecoF E n ST SG Q), Near c x p → Near c (x ∗ k) p

theorem **near_of_isPtOf** {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) (c : ℕ) : Near c g p

theorem **near_ptOf** (c : ℕ) {x : GL (Fin n) (AfE F E)} (hx : x ∈ UAd hc n) : Near c x (ptOf ST SG Q c x)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemRangeOfCls

theorem **cls_eq_iff** {x y : (FractionRing R)ˣ} : (x : Cls R) = y ↔ ∃ u : Rˣ, (x : FractionRing R) ∗ algebraMap R _ (u : R) = y

theorem **cls_pow_eq_one** [IsIntegrallyClosed R] {n : ℕ} (hn : n ≠ 0) (c : Cls R) (h : c ^ n = 1) : c = 1

theorem **exists_lift** [IsIntegrallyClosed R] {X : Type} [AddCommGroup X] [Module.Finite ℤ X] (b : X → X → Cls R) (h₁ : ∀ a a' c, b (a + a') c = b a c ∗ b a' c) (h₂ : ∀ a c, b a c = b c a) : ∃ t : X → X → (FractionRing R)ˣ, (∀ a c, (t a c : Cls R) = b a c) ∧ (∀ a a' c, t (a + a') c = t a c ∗ t a' c) ∧ ∀ a c a' c', b a c = b a' c' → t a c = t a' c'

theorem **coord_rel** (Dg : DegData R (maximalIdeal R)) (r : Dg.toPre.X → R) (hr : ∀ χ, PsiCoord Dg.toPre χ (r χ)) : (∀ a a' b, ∃ u : Rˣ, r (a + a' + b) ∗ r a ∗ r a' ∗ r b ∗ u = r (a + b) ∗ r (a' + b) ∗ r (a + a')) ∧ ∀ a, ∃ u : Rˣ, r (-a) ∗ u = r a

theorem **coord_spec** (χ : Dg.toPre.X) : PsiCoord Dg.toPre χ (coord Dg χ)

theorem **coord_ne** (χ : Dg.toPre.X) : coord Dg χ ≠ 0

theorem **toK_cls_eq** {x y : R} (hx : x ≠ 0) (hy : y ≠ 0) (h : ∃ u : Rˣ, x ∗ u = y) : (toK x hx : Cls R) = toK y hy

theorem **toK_mul** {x y : R} (hx : x ≠ 0) (hy : y ≠ 0) : toK (x ∗ y) (mul_ne_zero hx hy) = toK x hx ∗ toK y hy

theorem **psiCls_eq** {χ : Dg.toPre.X} {r : R} (hr : PsiCoord Dg.toPre χ r) : psiCls Dg χ = toK r (GlobalTheory.PsiCoordLocal.ne_zero_of_coord hr)

theorem **tauCls_symm** (a b : Dg.toPre.X) : tauCls Dg a b = tauCls Dg b a

theorem **tauCls_add** (a a' b : Dg.toPre.X) : tauCls Dg (a + a') b = tauCls Dg a b ∗ tauCls Dg a' b

theorem **psiCls_neg** (a : Dg.toPre.X) : psiCls Dg (-a) = psiCls Dg a

theorem **tauCls_self** (a : Dg.toPre.X) : tauCls Dg a a = psiCls Dg a ^ 2

theorem **tauCls_bal** {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] (ρ : Dg.toPre.OAct O) (x : O) (a b : Dg.toPre.X) : tauCls Dg (x • a) b = tauCls Dg a (star x • b)

theorem **tauCls_half** [IsIntegrallyClosed R] {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] (ρ : Dg.toPre.OAct O) {w : O} (hw : w + star w = 1) (a : Dg.toPre.X) : tauCls Dg a (w • a) = psiCls Dg a

theorem **exists_reading** [IsIntegrallyClosed R] {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] (ρ : Dg.toPre.OAct O) : ∃ t, IsReading Dg O t

theorem **IsReading.cls** {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] {t} (h : IsReading Dg O t) (a b : Dg.toPre.X) : (t a b : Cls R) = tauCls Dg a b

theorem **mem_range_of_cls** {x : (FractionRing R)ˣ} {r : R} (hr : r ≠ 0) (h : (x : Cls R) = toK r hr) : (x : FractionRing R) ∈ (algebraMap R (FractionRing R)).range

theorem **IsReading.pos** {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] {t} (h : IsReading Dg O t) (a : Dg.toPre.X) : (t a a : FractionRing R) ∈ (algebraMap R (FractionRing R)).range

theorem **IsReading.pos_max** [CharZero R] [IsIntegrallyClosed R] {O : Type} [Ring O] [StarRing O] [Module O Dg.toPre.X] {t} (h : IsReading Dg O t) {a : Dg.toPre.X} (ha : a ≠ 0) : ∃ r ∈ maximalIdeal R, algebraMap R (FractionRing R) r = t a a

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemRiemannRochSpaceIffRegularAtEverywhere

theorem **ι_str** : Y.ι ≫ PSpace.str F Y.N = Y.str

theorem **isProjOver** : IsProjOver Y.str

theorem **isNormalScheme** : IsNormalScheme Y.X

theorem **topologicalKrullDim_eq** : topologicalKrullDim Y.X = Y.dim

theorem **isPrime_coneIdeal** : Y.coneIdeal.IsPrime

theorem **range_ι** : Set.range Y.ι = ProjectiveSpectrum.zeroLocus (ProjSpace.grading F Y.N) Y.coneIdeal

theorem **subschemeι_ker** : i.ker.subschemeι = inv i.toImage ≫ i

theorem **isoOfClosedImmersion_ι** : (isoOfClosedImmersion i d).hom ≫ (ofClosedImmersion i d).ι = i

theorem **exists_iso** {Z : Scheme.{0}} (p : Z ⟶ Sp F) (d : ℕ) [SmoothOfRelativeDimension d p] [GeometricallyConnected p] (h : IsProjOver p) : ∃ (Y : Var F) (e : Z ≅ Y.X), Y.dim = d ∧ e.hom ≫ Y.str = p

theorem **algebraMap_Γ** (a : F) : algebraMap F Γ(Y.X, ⊤) a = Y.str.appTop ((Scheme.ΓSpecIso (.of F)).inv a)

theorem **algebraMap_Fn** : algebraMap F Y.Fn = Cartier.constFn Y.str

theorem **regAt_algebraMap** (x : Y.X) (a : F) : RegAt x (algebraMap F Y.Fn a)

theorem **mem_riemannRochSpace_iff_regularAt_everywhere** {Δ : Y.Dv} {f : Y.Fn} : f ∈ Y.H0 Δ ↔ ∀ x, Δ.RegularAt x f

theorem **mem_riemannRoch_iff_effective** {Δ : Y.Dv} {g : Y.Fnˣ} : (g : Y.Fn) ∈ Y.H0 Δ ↔ Y.Eff (principal g + Δ)

theorem **eff_iff_one_mem** {Δ : Y.Dv} : Y.Eff Δ ↔ (1 : Y.Fn) ∈ Y.H0 Δ

theorem **eff_zero** : Y.Eff 0

theorem **eff_add** (a b : Y.Dv) (ha : Y.Eff a) (hb : Y.Eff b) : Y.Eff (a + b)

theorem **mul_mem_riemannRochSpace_add** (Δ Δ' : Y.Dv) (f g : Y.Fn) (hf : f ∈ Y.H0 Δ) (hg : g ∈ Y.H0 Δ') : f ∗ g ∈ Y.H0 (Δ + Δ')

theorem **riemannRoch_mono** (Δ Δ' : Y.Dv) (h : Y.Eff (Δ' - Δ)) : Y.H0 Δ ≤ Y.H0 Δ'

theorem **finrank_sections_projectiveLine** {Δ : Y.Dv} {L : LB Y.X} {r : L.ι → Y.Fnˣ} (h : Presents Δ L r) : Module.finrank F (Y.H0 Δ) = lbH0 F L

theorem **omega0_ne_zero** : Y.omega0 ≠ 0

theorem **isDivOfForm_KY** : IsDivOfForm Y.KY Y.omega0

theorem **exists_frame_KY** (x : Y.X) : ∃ (U : Y.X.Opens) (hx : x ∈ U) (ω : TopForm.FormsOn Y.str Y.dim U) (g : Y.Fnˣ), TopForm.IsFrame ω ∧ Y.omega0 = (g : Y.Fn) • formOf U (gen_mem hx) ω ∧ ∀ y ∈ U, Y.KY.IsEqn y g

theorem **regularAt_KY_iff** {U : Y.X.Opens} {x : Y.X} (hx : x ∈ U) {ω : TopForm.FormsOn Y.str Y.dim U} (hω : TopForm.IsFrame ω) {f c : Y.Fn} (e : f • Y.omega0 = c • formOf U (gen_mem hx) ω) : Y.KY.RegularAt x f ↔ RegAt x c

theorem **toPic_KY** (K : CanonLB Y.str Y.dim) : toPic Y.KY = cls K.bundle K.isLB

theorem **properCl_iff** (Z : Y.Cl) : Y.properCl Z ↔ genericPoint Y.X ∉ Z

theorem **properCl_supp** (Δ : Y.Dv) : Y.properCl (Y.supp Δ)

theorem **vanishOn.properCl** {Δ : Y.Dv} {f : Y.Fn} {Z : Y.Cl} (h : Y.vanishOn Δ f Z) (hf : f ≠ 0) : Y.properCl Z

theorem **Eff.vanishOn_one_iff** {Δ : Y.Dv} (h : Y.Eff Δ) {Z : Y.Cl} : Y.vanishOn Δ 1 Z ↔ Y.subCl Z (Y.supp Δ)

theorem **isCompleteCurve_iff** (Z : Y.Cl) : IsCompleteCurve F Z ↔ IsIrreducible (Z : Set Y.X) ∧ topologicalKrullDim (Z : Set Y.X) = 1

theorem **isVariety_sysOf** (h : 1 ≤ Y.dim) {L : LB Y.X} (hL : L.IsLB) {n : ℕ} (s : Fin n → L.Sec) : (Y.sysOf hL s).IsVariety

@[simp] theorem **rat_sec** {Δ : Y.Dv} (f : Y.H0 Δ) : (presents_toLB Δ).rat (sec f) = f

theorem **ratSet_linSys** : Presents.ratSet (S

theorem **bpf_iff** : Y.bpf Δ (Set.range fun j => (b j : Y.Fn)) ↔ (Y.linSys b).BasePointFree

theorem **phiBir_iff** : Y.phiBir Δ (Set.range fun j => (b j : Y.Fn)) ↔ (∃ j, (Y.linSys b).b j ≠ 0) ∧ (Y.linSys b).Birational

theorem **phiNoContr_iff** (h : (Y.linSys b).BasePointFree) (bd : Y.Cl) : Y.phiNoContr Δ (Set.range fun j => (b j : Y.Fn)) bd ↔ (Y.linSys b).NoContract bd

theorem **isLineBundle_twistingSheafOne** : Y.O1.IsLB

theorem **isAmple_projective_O_one** : IsAmpleLB Y.O1

theorem **nonzeroLocus_coordSec** (j : Fin (Y.N + 1)) : nonzeroLocus (Y.coordSec j) = Y.ι ⁻¹ᵁ chart F Y.N j

theorem **twistingSheafOne_chart_eq_preimage_standardChart** (i : Fin (Y.N + 1)) : Y.O1.U (Y.idx i) = Y.ι ⁻¹ᵁ chart F Y.N i

theorem **chartEval_coordSec** (i : Fin (Y.N + 1)) (g : MvPolynomial (Fin (Y.N + 1)) F) : Twist.chartEval F Y.coordSec (Y.idx i) le_rfl g = Y.chartHom i (aeval (ratio F Y.N i) g)

theorem **mem_nonzeroLocus_form** {d : ℕ} {g : MvPolynomial (Fin (Y.N + 1)) F} (hg : g.IsHomogeneous d) (y : Y.X) : y ∈ nonzeroLocus (Y.coordSys.form hg) ↔ Y.ι y ∈ Proj.basicOpen (grading F Y.N) g

theorem **basePointFree_coordSys** : Y.coordSys.BasePointFree

theorem **eq_of_sameImage** {y y' : Y.X} (h : Y.coordSys.SameImage y y') : y = y'

theorem **noContract_coordSys** (bd : Set Y.X) : Y.coordSys.NoContract bd

theorem **exists_coord_ne_zero** : ∃ i, Y.ι (genericPoint Y.X) ∈ chart F Y.N i

theorem **genericPoint_mem_twistingSheafOne_chart** : genericPoint Y.X ∈ Y.O1.U (Y.idx i)

theorem **presents_hyp** : Presents (Y.hyp hi) Y.O1 (frameRat _ (Y.idx i) (Y.genericPoint_mem_twistingSheafOne_chart hi))

theorem **aeval_coordFn** (g : MvPolynomial (Fin (Y.N + 1)) F) : aeval (Y.coordFn hi) g = fnOf _ (Y.genericPoint_mem_twistingSheafOne_chart hi) (Y.chartHom i (aeval (ratio F Y.N i) g))

theorem **coordFn_self** : Y.coordFn hi i = 1

theorem **coordFn_mem** (j : Fin (Y.N + 1)) : Y.coordFn hi j ∈ Y.H0 (Y.hyp hi)

theorem **generatesAt_coordFn_iff** (j : Fin (Y.N + 1)) (y : Y.X) : (Y.hyp hi).GeneratesAt y (Y.coordFn hi j) ↔ Y.ι y ∈ chart F Y.N j

theorem **eff_hyp** : Y.Eff (Y.hyp hi)

theorem **mem_supp_hyp** (y : Y.X) : y ∈ Y.supp (Y.hyp hi) ↔ Y.ι y ∉ chart F Y.N i

theorem **bpf_hyp** : Y.bpf (Y.hyp hi) (Set.range (Y.coordFn hi))

theorem **phiNoContr_hyp** (bd : Y.Cl) : Y.phiNoContr (Y.hyp hi) (Set.range (Y.coordFn hi)) bd

theorem **phiBir_hyp** : Y.phiBir (Y.hyp hi) (Set.range (Y.coordFn hi))

theorem **not_phiNoContr_of_constant** (h : Y.dim = 1) {Δ : Y.Dv} {V : Set Y.Fn} (hV : ∀ f ∈ V, f ∈ Set.range (algebraMap F Y.Fn)) : ¬ Y.phiNoContr Δ V ⊥

theorem **riemannRoch_finite** (F : Type) [Field F] : RiemannRochFinite F

theorem **proj_dim** : (proj F n).dim = n

theorem **proj_gen_mem** (i : Fin (n + 1)) : (proj F n).ι (genericPoint (proj F n).X) ∈ chart F n i

theorem **finrank_riemannRoch_hyperplane** : Module.finrank F ((proj F n).H0 (projHyp F n)) = n + 1

theorem **riemannRoch_hyperplane_ne_bot** : (proj F n).H0 (projHyp F n) ≠ ⊥

theorem **exists_nonconstant_line** : ∃ f ∈ (proj F 1).H0 (projHyp F 1), f ∉ Set.range (algebraMap F (proj F 1).Fn)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MemSupportBDIff

theorem **germOf_pullEqn** {i i' : B.ι} {y : X} (hi : f y ∈ B.U i) (hi' : f y ∈ B.U i') : germOf y (pullEqn B f h i hi) = germOf y (pullEqn B f h i' hi')

theorem **isEqn_pullDiv** (i : B.ι) {y : X} (hy : f y ∈ B.U i) : (pullDiv B f h).IsEqn y (pullEqn B f h i hy)

theorem **eff_pullDiv** : (pullDiv B f h).Eff

theorem **mem_supp_pullDiv** (y : X) : y ∈ (pullDiv B f h).supp ↔ f y ∈ B.idealSheaf.support

theorem **one_tmul_eq_iff** {Δ Δ' : Y.Dv} : (1 : ℚ) ⊗ₜ[ℤ] Δ = 1 ⊗ₜ Δ' ↔ ∃ q : ℕ, 0 < q ∧ q • Δ = q • Δ'

theorem **combNum_mul** (i : ι) : (combNum ρ t i : ℚ) * ((ρ i).n : ℕ) = (combDen ρ t : ℕ) * t i

theorem **isEqn_combEqn** (x : Y.X) : (∑ i, combNum ρ t i • (ρ i).D).IsEqn x (combEqn ρ t x)

theorem **isComb_comb** : IsComb (comb ρ t g) ρ (fun i => t i) g

theorem **qDiv_comb** : qDiv (comb ρ t g) = ∑ i, t i • qDiv (ρ i)

theorem **sum_coef_fund** [Fintype T] (u : T → Fin N → ℝ) (τ : T) (a : Fin N) : ∑ τ', ∑ i, coef u τ' i * fund τ' i τ a = u τ a

theorem **coef_nonneg** {u : T → Fin N → ℝ} (h : ∀ τ, Antitone (u τ)) (τ : T) {i : Fin N} (hi : (i : ℕ) + 1 < N) : 0 ≤ coef u τ i

@[simp] theorem **ratOf_cast** (q : ℚ) : ratOf q = q

theorem **mem_support_BD_iff** (k : s.Comp) (y : (Φ.C k).Y.X) : ((Φ.C k).π ≫ genι s.T.Z) y ∈ s.T.BD.idealSheaf.support ↔ (Φ.C k).π y ∈ genBd s.T.Z

theorem **eff_bdPull** (k : s.Comp) : (bdPull Φ k).Eff

theorem **mem_supp_bdPull** (k : s.Comp) (y : (Φ.C k).Y.X) : y ∈ (bdPull Φ k).supp ↔ (Φ.C k).π y ∈ genBd s.T.Z

theorem **phW_eq_sum** (k : s.Comp) (σ : D.what₂.LL →+* ℂ) (u : D.what₂.Tau → Fin N → ℝ) (y : Pts (Φ.C k).Y σ) : phW hm h0 k σ u y = ∑ τ, ∑ i, coef u τ i * phW hm h0 k σ (fund τ i) y

theorem **m_pos** : 0 < P.m

theorem **dir_apply** (τ : D.what₂.Tau) (i : Fin N) : P.dir τ i = P.m * (varpi j i : ℚ) + P.η * P.μ i

theorem **strictAnti_dir** (τ : D.what₂.Tau) : StrictAnti (P.dir τ)

theorem **qDiv_ab** (k : s.Comp) : qDiv (P.ab k) = qL Φ k P.dirQ − 1 ⊗ₜ P.bd k

theorem **eff_bd** (k : s.Comp) : (P.bd k).Eff

theorem **mem_supp_bd** (k : s.Comp) (y : (Φ.C k).Y.X) : y ∈ (P.bd k).supp ↔ (Φ.C k).π y ∈ genBd s.T.Z

theorem **nefFib_ab** (k : s.Comp) : (P.ab k).NefFib

theorem **reg_add** (u v : D.what₂.Tau → Fin N → ℝ) (t : ℝ) (y : Pts (Φ.C k).Y σ) : P.reg k σ (u + v) t y = P.reg k σ u t y + P.reg k σ v t y

theorem **reg_smul** (r : ℝ) (u : D.what₂.Tau → Fin N → ℝ) (t : ℝ) (y : Pts (Φ.C k).Y σ) : P.reg k σ (r • u) t y = r * P.reg k σ u t y

theorem **reg_zero** (u : D.what₂.Tau → Fin N → ℝ) {y : Pts (Φ.C k).Y σ} (hy : y ∈ offBd Φ k σ) : P.reg k σ u 0 y = phW hm h0 k σ u y

theorem **reg_central** {u : D.what₂.Tau → Fin N → ℝ} (hu : ∀ τ a b, u τ a = u τ b) (t : ℝ) {y : Pts (Φ.C k).Y σ} (hy : y ∈ offBd Φ k σ) : P.reg k σ u t y = phW hm h0 k σ u y

theorem **qDiv_mbarQ** (B : ℝ) (u : D.what₂.Tau → Fin N → ℚ) (r : ℚ) : qDiv (P.mbarQ k B u r) = qL Φ k u − r ⊗ₜ P.bd k

theorem **mbarOf_cast** (P : PolDat s Φ hm h0) (B : ℝ) (u : D.what₂.Tau → Fin N → ℚ) {r : ℚ} (hr : 0 < r) (k : s.Comp) : mbarOf P B (fun τ i => u τ i) r k = P.mbarQ k B u r

theorem **isComb_mbarOf** (P : PolDat s Φ hm h0) (B : ℝ) {u : D.what₂.Tau → Fin N → ℝ} (hu : ∀ τ i, ∃ q : ℚ, u τ i = q) {r : ℚ} (hr : 0 < r) (k : s.Comp) : IsComb (mbarOf P B u r k) (P.gens k B) (fun o => o.elim (r : ℝ) fun x => −(u x.1 x.2 − r * P.dir x.1 x.2)) fun σ y => P.reg k σ (fun τ i => u τ i − r * P.dir τ i) (Real.exp (−((r : ℝ) ^ 2)⁻¹)) y / 2

theorem **mbIs_of_eq** (O : Setup D) (P : PolDat O.s O.Φ O.met.hm O.met.h0) (h₁ : O.met.ab = P.ab) (h₂ : O.met.dir = P.dir) (h₃ : O.met.reg = P.reg) (h : O.Mb = mbarOf P) : O.MbIs

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ModelLamPDef

theorem **fits_of_rows** {C : Cols B} (W : RowsRest B C) : Fits C (fun μ => Module.finrank ℂ (W.Amb μ)) where

theorem **stdJ_single** {L : Type} [Field L] (σ : L →+* ℂ) (n : ℕ) (i : Fin n) : stdJ σ n (Pi.single i 1) = (EuclideanSpace.basisFun (Fin n) ℂ).toBasis i

theorem **stdJ_basis** {L : Type} [Field L] (σ : L →+* ℂ) (n : ℕ) {κ : Type} (b : Module.Basis κ L (Fin n → L)) : LinearIndependent ℂ (fun i => stdJ σ n (b i)) ∧ Submodule.span ℂ (Set.range (fun i => stdJ σ n (b i))) = ⊤

theorem **trunc_self** (a : ℕ) (x : Eu a) : trunc a a x = x

theorem **chain_lin** {R : Ranks} {ι : Type} [Fintype ι] {hc : ι → HCParam R} (C : ColumnChain R ι hc) (j : ℕ) : IsLinearMap ℂ (Formalism.ThetaLiftIndependence.chainOpM C.wt C.Kmap j)

theorem **mdim_m** (R : Ranks) (n e : ℕ) : mdim n e R.m = e

theorem **nonempty_iff** (C : Cols B) : Nonempty (RowsRest B C) ↔ ∃ d, Fits C d

theorem **two_lt_alt** (B : Pre) (u : B.Word) : 2 < B.T u

theorem **tcols_ipC** (C : Cols B) (u : B.Word) (x y : ((tcols C).mem u).C.cls 0 → Unit → ℂ) : ((tcols C).fm u).ipC 0 x y = ∑ c, starRingEnd ℂ (x c ()) * y c ()

theorem **tcols_fits** (C : Cols B) : Fits (tcols C) (fun _ => 1) where

theorem **exists_rows** (C₀ : Cols B) (n n₀ a : ℕ) : ∃ (C : Cols B) (W : RowsRest B C), W.what₂.Drows = n ∧ W.what₂.D0 = n₀ ∧ (∀ μ j, Module.finrank ℂ (W.what₂.Mt μ j) = a) ∧ (∀ μ, Module.finrank ℂ (W.what₂.Amb μ) = 1) ∧ (∀ u, Nontrivial (W.what₂.Mu u)) ∧ W.what₂.muStar ≠ W.what₂.mu0

theorem **trunc_trunc** {a b c : ℕ} (h : a = b) (x : Eu a) : trunc b c (trunc a b x) = trunc a c x

theorem **mdim_self** (n j : ℕ) : mdim n n j = n

theorem **congr_symm_eq** {a b : ℕ} (h : a = b) (x : Eu b) : (LinearIsometryEquiv.piLpCongrLeft 2 ℂ ℂ (finCongr h)).symm x = trunc b a x

theorem **model_buDef** {C : Cols B} {d : C.Tup → ℕ} (h : Fits C d) (n₀ a : ℕ) : Rows.BuDef (model h (d C.mu0) n₀ a).what₂

theorem **model_lamPDef** {C : Cols B} {d : C.Tup → ℕ} (h : Fits C d) (n n₀ a : ℕ) : Rows.LamPDef (model h n n₀ a).what₂

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MonodromyZeroOfWeilDeligneRepClassIff

theorem **IsSmooth.of_open** {h : IsPlaceOver A w} {M : Type*} [Monoid M] {f : Weil h →* M} {U : Set (Ine (F

theorem **IsSmooth.of_le** {h : IsPlaceOver A w} {M M' : Type*} [Monoid M] [Monoid M'] {f : Weil h →* M} {g : Weil h →* M'} (hf : IsSmooth f) (hg : ∀ σ, f σ = 1 → g σ = 1) : IsSmooth g

theorem **IsSmooth.of_le₂** {h : IsPlaceOver A w} {M M' M'' : Type*} [Monoid M] [Monoid M'] [Monoid M''] {f : Weil h →* M} {f' : Weil h →* M'} {g : Weil h →* M''} (hf : IsSmooth f) (hf' : IsSmooth f') (hg : ∀ σ, f σ = 1 → f' σ = 1 → g σ = 1) : IsSmooth g

theorem **IsSmooth.of_trivial** {h : IsPlaceOver A w} {M : Type*} [Monoid M] {f : Weil h →* M} (hf : ∀ σ, f (ineIncl h σ) = 1) : IsSmooth f

theorem **IsSmooth.finite_range** {h : IsPlaceOver A w} {M : Type*} [Monoid M] {f : Weil h →* M} (hf : IsSmooth f) : (Set.range fun σ : Ine (F

theorem **isSmooth_unrChar** (h : IsPlaceOver A w) (c : ℂˣ) : IsSmooth (unrChar h c)

theorem **comm_apply** (a : WDRep h V) (σ : Weil h) (v : V) : a.ρ σ (a.N v) = ((normV w : ℕ) : ℂ) ^ deg σ • a.N (a.ρ σ v)

theorem **conj_N** (a : WDRep h V) (σ : Weil h) : a.ρ σ * a.N * a.ρ σ⁻¹ = ((normV w : ℕ) : ℂ) ^ deg σ • a.N

theorem **Iso.refl** (a : WDRep h V) : a.Iso a

theorem **Iso.symm** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) : b.Iso a

theorem **Iso.trans** {a : WDRep h V} {b : WDRep h V'} {c : WDRep h V''} (hab : a.Iso b) (hbc : b.Iso c) : a.Iso c

theorem **iso_transport** (a : WDRep h V) (e : V ≃ₗ[ℂ] V') : a.Iso (a.transport e)

@[simp] theorem **sum_ρ** (a : WDRep h V) (b : WDRep h V') (σ : Weil h) : (a.sum b).ρ σ = (a.ρ σ).prodMap (b.ρ σ)

@[simp] theorem **sum_N** (a : WDRep h V) (b : WDRep h V') : (a.sum b).N = a.N.prodMap b.N

theorem **Iso.sum** {a : WDRep h V} {a' : WDRep h V'} {b : WDRep h V''} {b' : WDRep h V'''} (ha : a.Iso a') (hb : b.Iso b') : (a.sum b).Iso (a'.sum b')

@[simp] theorem **twist_ρ** (a : WDRep h V) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (σ : Weil h) : (a.twist χ hχ).ρ σ = (χ σ : ℂ) • a.ρ σ

@[simp] theorem **twist_N** (a : WDRep h V) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : (a.twist χ hχ).N = a.N

theorem **Iso.twist** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : (a.twist χ hχ).Iso (b.twist χ hχ)

@[simp] theorem **tw_ρ** (a : WDRep h V) (s : ℝ) (σ : Weil h) : (a.tw s).ρ σ = (((normV w : ℕ) : ℂ) ^ ((s : ℝ) : ℂ)) ^ deg σ • a.ρ σ

@[simp] theorem **tw_N** (a : WDRep h V) (s : ℝ) : (a.tw s).N = a.N

@[simp] theorem **tensor_ρ** (a : WDRep h V) (b : WDRep h V') (σ : Weil h) : (a.tensor b).ρ σ = TensorProduct.map (a.ρ σ) (b.ρ σ)

@[simp] theorem **tensor_N** (a : WDRep h V) (b : WDRep h V') : (a.tensor b).N = a.N.rTensor V' + b.N.lTensor V

theorem **Iso.tensor** {a : WDRep h V} {a' : WDRep h V'} {b : WDRep h V''} {b' : WDRep h V'''} (ha : a.Iso a') (hb : b.Iso b') : (a.tensor b).Iso (a'.tensor b')

@[simp] theorem **dual_ρ** (a : WDRep h V) (σ : Weil h) : a.dual.ρ σ = (a.ρ σ⁻¹).dualMap

@[simp] theorem **dual_N** (a : WDRep h V) : a.dual.N = -a.N.dualMap

theorem **Iso.dual** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) : a.dual.Iso b.dual

@[simp] theorem **ofChars_ρ** {ι : Type*} [Finite ι] (χ : ι → Weil h →* ℂˣ) (hχ : ∀ i, IsSmooth (χ i)) (σ : Weil h) : (ofChars h χ hχ).ρ σ = diagEnd fun i => (χ i σ : ℂ)

@[simp] theorem **ofChars_N** {ι : Type*} [Finite ι] (χ : ι → Weil h →* ℂˣ) (hχ : ∀ i, IsSmooth (χ i)) : (ofChars h χ hχ).N = 0

@[simp] theorem **diag_ρ** {ι : Type*} [Finite ι] (c : ι → ℂˣ) (σ : Weil h) : (diag h c).ρ σ = diagEnd fun i => (c i : ℂ) ^ deg σ

@[simp] theorem **diag_N** {ι : Type*} [Finite ι] (c : ι → ℂˣ) : (diag h c).N = 0

@[simp] theorem **sp_ρ** (n : ℕ) (σ : Weil h) : (sp h n).ρ σ = diagEnd fun i => (spEig w n i : ℂ) ^ deg σ

@[simp] theorem **sp_N** (n : ℕ) : (sp h n).N = shiftEnd n

@[simp] theorem **ofMatrix_ρ** {ι : Type*} [Fintype ι] [DecidableEq ι] (ρ : Weil h →* GL ι ℂ) (N : Matrix ι ι ℂ) (h1 h2 h3 h4) (σ : Weil h) : (ofMatrix ρ N h1 h2 h3 h4).ρ σ = Matrix.toLin' (ρ σ : Matrix ι ι ℂ)

@[simp] theorem **ofMatrix_N** {ι : Type*} [Fintype ι] [DecidableEq ι] (ρ : Weil h →* GL ι ℂ) (N : Matrix ι ι ℂ) (h1 h2 h3 h4) : (ofMatrix ρ N h1 h2 h3 h4).N = Matrix.toLin' N

@[simp] theorem **coe_toGL** {ι : Type*} [Fintype ι] [DecidableEq ι] (a : WDRep h (ι → ℂ)) (σ : Weil h) : (a.toGL σ : Matrix ι ι ℂ) = LinearMap.toMatrix' (a.ρ σ)

theorem **Iso.finrank_eq** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) : finrank ℂ V = finrank ℂ V'

theorem **Iso.charpoly_eq** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) (σ : Weil h) : (a.ρ σ).charpoly = (b.ρ σ).charpoly

theorem **Iso.mono0** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) (ha : a.Mono0) : b.Mono0

theorem **Iso.ρ_eq_one** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) {σ : Weil h} (ha : a.ρ σ = 1) : b.ρ σ = 1

theorem **Iso.unr** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) (ha : a.Unr) : b.Unr

theorem **Iso.tempered** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) (ha : a.Tempered) : b.Tempered

theorem **map_N_eigenspace** (a : WDRep h V) {φ : Weil h} (hφ : deg φ = 1) {α : ℂ} {v : V} (hv : v ∈ End.eigenspace (a.ρ φ) α) : a.N v ∈ End.eigenspace (a.ρ φ) (((normV w : ℕ) : ℂ) * α)

theorem **unr_diag** {ι : Type*} [Finite ι] (c : ι → ℂˣ) : (diag h c).Unr

theorem **tempered_sp** (n : ℕ) : (sp h n).Tempered

theorem **iso_std** (a : WDRep h V) : a.Iso (std a).2

theorem **mk_eq_mk** {a : WDRep h V} {b : WDRep h V'} : mk a = mk b ↔ a.Iso b

theorem **sound** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) : mk a = mk b

theorem **quotientMk_eq** (a : Σ n : ℕ, WDRep h (Fin n → ℂ)) : Quotient.mk (isoSetoid h) a = mk a.2

theorem **exists_rep** (x : WD h) : ∃ (n : ℕ) (a : WDRep h (Fin n → ℂ)), mk a = x

theorem **ind** {P : WD h → Prop} (hP : ∀ (n : ℕ) (a : WDRep h (Fin n → ℂ)), P (mk a)) (x : WD h) : P x

theorem **exists_ofMatrix** (x : WD h) : ∃ (n : ℕ) (ρ : Weil h →* GL (Fin n) ℂ) (N : Matrix (Fin n) (Fin n) ℂ) (h1 : _) (h2 : _) (h3 : _) (h4 : _), mk (WDRep.ofMatrix ρ N h1 h2 h3 h4) = x

@[simp] theorem **lpDim_mk** (a : WDRep h V) : lpDim (mk a) = finrank ℂ V

@[simp] theorem **lpSum_mk** (a : WDRep h V) (b : WDRep h V') : lpSum (mk a) (mk b) = mk (a.sum b)

@[simp] theorem **lpTwist_mk** (a : WDRep h V) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : lpTwist (mk a) χ hχ = mk (a.twist χ hχ)

@[simp] theorem **lpTw_mk** (a : WDRep h V) (s : ℝ) : lpTw (mk a) s = mk (a.tw s)

@[simp] theorem **lpChar_mk** (a : WDRep h V) (b : WDRep h V') : lpChar (mk a) (mk b) = mk (a.tensor b)

@[simp] theorem **lpDual_mk** (a : WDRep h V) : lpDual (mk a) = mk a.dual

theorem **coe_unitOf** {z : ℂ} (hz : z ≠ 0) : (unitOf z : ℂ) = z

@[simp] theorem **monodromyZero_ofWeilDeligneRepClass_iff** (a : WDRep h V) : mono0 (mk a) ↔ a.N = 0

@[simp] theorem **unr_mk** (a : WDRep h V) : unr (mk a) ↔ a.Unr

@[simp] theorem **tempered_mk** (a : WDRep h V) : tempered (mk a) ↔ a.Tempered

@[simp] theorem **charPol_mk** (a : WDRep h V) (σ : Weil h) : charPol (mk a) σ = (a.ρ σ).charpoly

@[simp] theorem **eig_mk** (a : WDRep h V) (σ : Weil h) : eig (mk a) σ = (a.ρ σ).charpoly.roots

@[simp] theorem **frobEig_mk** (a : WDRep h V) : frobEig (mk a) = (a.ρ (frob h)).charpoly.roots

theorem **inePol_of_mem** (x : WD h) {σ : G E Ω} (hσ : σ ∈ Weil h) : inePol x σ = charPol x ⟨σ, hσ⟩

theorem **inePol_of_not_mem** (x : WD h) {σ : G E Ω} (hσ : σ ∉ Weil h) : inePol x σ = 1

theorem **inePol_of_mem_Ine** (x : WD h) {σ : G E Ω} (hσ : σ ∈ Ine (F

theorem **charPol_monic** (x : WD h) (σ : Weil h) : (charPol x σ).Monic

theorem **charPol_natDegree** (x : WD h) (σ : Weil h) : (charPol x σ).natDegree = lpDim x

theorem **charPol_eq_prod** (x : WD h) (σ : Weil h) : charPol x σ = ((eig x σ).map fun α => X - C α).prod

theorem **card_eig** (x : WD h) (σ : Weil h) : Multiset.card (eig x σ) = lpDim x

theorem **charPol_eq_of_unr** {x : WD h} (hx : unr x) {σ τ : Weil h} (hστ : deg σ = deg τ) : charPol x σ = charPol x τ

theorem **eig_eq_frobEig** {x : WD h} (hx : unr x) {φ : Weil h} (hφ : deg φ = 1) : eig x φ = frobEig x

theorem **frobEig_diag** {ι : Type*} [Fintype ι] (c : ι → ℂˣ) : frobEig (mk (WDRep.diag h c)) = Finset.univ.val.map fun i => (c i : ℂ)

@[simp] theorem **lpDim_lpSp** (n : ℕ) : lpDim (lpSp h n) = n

theorem **tempered_lpSp** (n : ℕ) : tempered (lpSp h n)

theorem **not_mono0_lpSp** {n : ℕ} (hn : 2 ≤ n) : ¬ mono0 (lpSp h n)

theorem **_root_.GlobalTheory.WeilDeligne.WDRep.Tempered.norm_eq_one** {a : WDRep h V} (ha : a.Tempered) (hN : a.N = 0) {φ : Weil h} (hφ : deg φ = 1) : absSpace (a.ρ φ) 1 = ⊤

theorem **tempered_diag_iff** {ι : Type*} [Finite ι] (c : ι → ℂˣ) : (WDRep.diag h c).Tempered ↔ ∀ i, ‖(c i : ℂ)‖ = 1

theorem **sp_two_spec** : (WD.lpSp h 2).lpDim = 2 ∧ (WD.lpSp h 2).tempered ∧ ¬ (WD.lpSp h 2).mono0 ∧ ¬ (WD.lpSp h 2).unr

theorem **not_tempered_diag_spEig** : ¬ (WD.mk (WDRep.diag h (WDRep.spEig w 2))).tempered

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MulVecMemLatMap

theorem **up_tri** (h : IsUp u) : u.BlockTriangular id

theorem **mul_diag** (hu : u.BlockTriangular id) (hv : v.BlockTriangular id) (i : Fin m) : (u ∗ v) i i = u i i ∗ v i i

theorem **up_one** : IsUp (1 : Matrix (Fin m) (Fin m) E)

theorem **up_mul** (hu : IsUp u) (hv : IsUp v) : IsUp (u ∗ v)

theorem **up_det** (hu : IsUp u) : u.det = 1

theorem **up_mul_inv** (hu : IsUp u) : u ∗ u⁻¹ = 1

theorem **up_inv_mul** (hu : IsUp u) : u⁻¹ ∗ u = 1

theorem **up_inv** (hu : IsUp u) : IsUp u⁻¹

theorem **lo_cT** (σ : E →+* E) (hu : IsUp u) : IsLo (cT σ u)

theorem **lo_mul** (hu : IsLo u) (hv : IsLo v) : IsLo (u ∗ v)

theorem **lo_det** (hu : IsLo u) : u.det = 1

theorem **lead_block_mul** (Z : Matrix (Fin m) (Fin m) E) (hu : IsUp u) {l : ℕ} (hl : l ≤ m) : (Z ∗ u).submatrix (Fin.castLE hl) (Fin.castLE hl) = Z.submatrix (Fin.castLE hl) (Fin.castLE hl) ∗ u.submatrix (Fin.castLE hl) (Fin.castLE hl)

theorem **lead_mul_up** (Z : Matrix (Fin m) (Fin m) E) (hu : IsUp u) (l : ℕ) (hl : l ≤ m) : SgAlg.lead (Z ∗ u) l hl = SgAlg.lead Z l hl

theorem **lead_lo_mul** (Z : Matrix (Fin m) (Fin m) E) (hu : IsLo u) (l : ℕ) (hl : l ≤ m) : SgAlg.lead (u ∗ Z) l hl = SgAlg.lead Z l hl

theorem **strict_conj** {r : ℕ} {u v A : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (hu : IsUp u) (hv : IsUp v) (hA : IsStrict A) : IsStrict (v ∗ A ∗ u) ∧ simpleSum (v ∗ A ∗ u) = simpleSum A

theorem **cT_mul** {a b c : Type} [Fintype b] (A : Matrix a b E) (B : Matrix b c E) : cT σ (A ∗ B) = cT σ B ∗ cT σ A

theorem **pairM_isom** (hs : IsUn σ J s) : pairM σ J (s ∗ P') (s ∗ P) = pairM σ J P' P

theorem **pairM_mul** (u v : Matrix (Fin (r + 1)) (Fin (r + 1)) E) : pairM σ J (P' ∗ u) (P ∗ v) = cT σ v ∗ pairM σ J P' P ∗ u

theorem **eOf_mul** (hu : IsUp u) : eOf (P ∗ u) = eOf P ∗ u.submatrix Fin.castSucc Fin.castSucc

theorem **up_block** (hu : IsUp u) : IsUp (u.submatrix Fin.castSucc Fin.castSucc)

theorem **spanX_mul** (hu : IsUp u) : spanX (P ∗ u) = spanX P

theorem **perpX_mul** (hu : IsUp u) : perpX σ J (P ∗ u) = perpX σ J P

theorem **mulVec_mem_spanX** {x : Fin n → E} (hx : x ∈ spanX P) : s ∗ᵥ x ∈ spanX (s ∗ P)

theorem **isFrame_mul** {T : E} (h : IsFrame σ J T P) (hu : IsUp u) : IsFrame σ J T (P ∗ u)

theorem **isFrame_isom** (hJ : IsUnit J.det) {T : E} (h : IsFrame σ J T P) (hs : IsUn σ J s) : IsFrame σ J T (s ∗ P)

theorem **isUn_inv** (hJ : IsUnit J.det) (hs : IsUn σ J s) : IsUn σ J s⁻¹

theorem **inS_iff** : InS σ J P s ↔ IsUn σ J s ∧ ∃ a : Matrix (Fin (r + 1)) (Fin (r + 1)) E, IsUp a ∧ s ∗ P = P ∗ a

theorem **inN_iff** : InN σ J P s ↔ IsUn σ J s ∧ (∃ a : Matrix (Fin r) (Fin r) E, IsUp a ∧ s ∗ eOf P = eOf P ∗ a) ∧ ∀ x ∈ perpX σ J P, ∃ c, s ∗ᵥ x − x = eOf P ∗ᵥ c

theorem **inS_one** : InS σ J P 1

theorem **inS_mul** {t : Matrix (Fin n) (Fin n) E} (hs : InS σ J P s) (ht : InS σ J P t) : InS σ J P (s ∗ t)

theorem **inS_inv** (hJ : IsUnit J.det) (hs : InS σ J P s) : InS σ J P s⁻¹

theorem **inS_of_inN** {T : E} (hP : IsFrame σ J T P) (hs : InN σ J P s) : InS σ J P s

theorem **inN_one** : InN σ J P 1

theorem **inN_perp** (hs : InN σ J P s) (x : Fin n → E) : s ∗ᵥ x ∈ perpX σ J P ↔ x ∈ perpX σ J P

theorem **inN_mul** {t : Matrix (Fin n) (Fin n) E} (hs : InN σ J P s) (ht : InN σ J P t) : InN σ J P (s ∗ t)

theorem **inN_inv** (hJ : IsUnit J.det) (hs : InN σ J P s) : InN σ J P s⁻¹

theorem **mem_UO_iff** : u ∈ UO E r ↔ IsUp u ∧ u ∈ triRing E r

theorem **UO_mul** (hu : u ∈ UO E r) (hv : v ∈ UO E r) : u ∗ v ∈ UO E r

theorem **UO_inv** (hu : u ∈ UO E r) : u⁻¹ ∈ UO E r

theorem **frameLat_mul** (hu : u ∈ UO E r) : frameLat (P ∗ u) = frameLat P

theorem **mem_UO_of_le** (hP : Free P) (hu : IsUp u) (h : frameLat (P ∗ u) ≤ frameLat P) : u ∈ UO E r

theorem **frameLat_isom** (s : Matrix (Fin n) (Fin n) E) : frameLat (s ∗ P) = latMap s (frameLat P)

theorem **mulVec_mem_latMap** (hs : IsUnit s.det) (x : Fin n → E) : s ∗ᵥ x ∈ latMap s M ↔ x ∈ M

theorem **mulVec_mem_span** (hs : IsUnit s.det) (x : Fin n → E) : s ∗ᵥ x ∈ Submodule.span E (latMap s M : Set (Fin n → E)) ↔ x ∈ Submodule.span E (M : Set (Fin n → E))

theorem **primIn_latMap** (hs : IsUnit s.det) : PrimIn (latMap s M) (latMap s Λ) ↔ PrimIn M Λ

theorem **selfDual_latMap** (hJ : IsUnit J.det) (hs : IsUn σ J s) (h : SelfDual σ J Λ) : SelfDual σ J (latMap s Λ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.MultiplicationLatticeSecond

theorem **isInv_left** (φ : a.GIso F₁ F₂) : a.IsInv F₁

theorem **isInv_right** (φ : a.GIso F₁ F₂) : a.IsInv F₂

@[simp] theorem **iso_mk** (φ : F₁.toMod.restrict Z.genO.ι ≅ F₂.toMod.restrict Z.genO.ι) (h₁ : a.IsInv F₁) (h₂ : a.IsInv F₂) : (mk φ h₁ h₂).iso = φ

theorem **ext** {φ ψ : a.GIso F₁ F₂} (h : φ.iso = ψ.iso) : φ = ψ

@[simp] theorem **iso_refl** (h : a.IsInv F₁) : (refl h).iso = Iso.refl _

@[simp] theorem **iso_symm** (φ : a.GIso F₁ F₂) : φ.symm.iso = φ.iso.symm

@[simp] theorem **iso_trans** (φ : a.GIso F₁ F₂) (ψ : a.GIso F₂ F₃) : (φ.trans ψ).iso = φ.iso ≪≫ ψ.iso

@[simp] theorem **symm_symm** (φ : a.GIso F₁ F₂) : φ.symm.symm = φ

theorem **bs_loc** (b : Option ℕ) : bs st.loc b = ct.B.X b

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NStageTrivialCharacterCoeffEqAlt

theorem **NLay0.eta_nonneg** (L : NLay0 R H what₂ LL0) (j : ℕ) (w : what₂.places.Pl) : 0 ≤ L.eta j w

theorem **NLay0.zOk_fixZ** (L : NLay0 R H what₂ LL0) : L.fixZ.ZOk

theorem **NLay0.ZOk.reEta** {L : NLay0 R H what₂ LL0} (h : L.ZOk) (η : ℕ → what₂.places.Pl → ℚ) (hη : ∀ j w, 0 ≤ η j w) : @NLay0.ZOk R H (rowsEta what₂ η hη) ‹NumberField what₂.LL› ‹StarRing what₂.LL› LL0 (L.reEta η hη)

theorem **NLay0.etaOk_reEta** (L : NLay0 R H what₂ LL0) : @NLay0.EtaOk R H (rowsEta what₂ L.eta L.eta_nonneg) ‹NumberField what₂.LL› ‹StarRing what₂.LL› LL0 (L.reEta L.eta L.eta_nonneg)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NatToBEquivariant

theorem **isKernel_pull** {S S' : Scheme.{0}} (f : S' ⟶ S) {K G B : Over S} [GrpObj B] {i : K ⟶ G} {q : G ⟶ B} (h : IsPullback i (toUnit K) q η[B]) : IsPullback ((Over.pullback f).map i) (toUnit _) ((Over.pullback f).map q) η[(Over.pullback f).obj B]

theorem **eigMap_exact** {R S O : Type} [CommRing R] [CommRing S] [CommRing O] (f : R →+* S) {K G B : Over (Sp R)} [GrpObj K] [IsCommMonObj K] [GrpObj G] [IsCommMonObj G] [GrpObj B] [IsCommMonObj B] {c : OAct O K} {a : OAct O G} {b : OAct O B} {i : K ⟶ G} {q : G ⟶ B} (hi : IsLinearHom c a i) (hq : IsLinearHom a b q) (h : IsPullback i (toUnit K) q η[B]) (σ : O →+* S) : Function.Injective (eigMap f hi σ) ∧ Function.Exact (eigMap f hi σ) (eigMap f hq σ)

theorem **eigDefect_map** {ρ : Λ →+* Module.End O M} {ρ' : Λ →+* Module.End O M'} {f : M →ₗ[O] M'} (hf : IsEquivariant ρ ρ' f) (i : J) (x : M) : f.baseChange C (eigDefect h ρ i x) = eigDefect h ρ' i (f x)

theorem **contr_comp** {ρ : Λ →+* Module.End O M} {ρ' : Λ →+* Module.End O M'} {f : M →ₗ[O] M'} (hf : IsEquivariant ρ ρ' f) (ψ : LinearMap.BilinForm O M') (i : J) (x x' : M) : contr h ρ (ψ.compl₁₂ f f) i x x' = contr h ρ' ψ i (f x) (f x')

theorem **contr_of_eigen** {ρ : Λ →+* Module.End O M} (ψ : LinearMap.BilinForm O M) (i : J) {x : M} (hx : (1 : C) ⊗ₜ[O] x ∈ eigenSub (baseChangeAction C ρ) (σ i)) (x' : M) : contr h ρ ψ i x x' = 0

theorem **natToB_equivariant** : IsEquivariant (natAct hq) (DeRham.act (dp0 O) (𝟙 _) b) (natToB hq)

theorem **natToLie_natLift** (v : Lie a) : natToLie hq (natLift hq sp v) = v

theorem **kapExt_eq** (i : J) (u u' : Lie a) : kapExt hq sp h i u u' = kapAb sp h i (Lie.mapLinear hq u) (Lie.mapLinear hq u')

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonemptyAllIff

theorem **goodCl_of_all** (A : NormalizedTwistCompatibility D C I) : GoodCl D C I

theorem **nonempty_all_iff** : Nonempty (NormalizedTwistCompatibility D C I) ↔ Nonempty (NAllC D C) ∧ GoodCl D C I

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonemptyExt

theorem **nonempty_ext** (v : HeightOneSpectrum (𝒪 F)) : Nonempty (Ext v)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonemptyGrid

theorem **listPin_anti** {D : D7} {I : Type} {Lc : LevCh D I} {μ0 : I} (hL : Lc.Is μ0) {CF CF' b : ℝ} (h : CF ≤ CF') (hb : 1 ≤ b) (hp : ListPin D Lc CF' b) : ListPin D Lc CF b

theorem **not_listPin** {D : D7} {I : Type} (Lc : LevCh D I) (CF b : ℝ) (h : q0 D = D.host.g 0) : ¬ ListPin D Lc CF b

theorem **not_listPin_of_empty** {D : D7} {I : Type} (Lc : LevCh D I) (CF b : ℝ) {i : D.ntw.Tw} (hi : D.ntw.NonzeroIndexPlace i) (h : IsEmpty (D.ntw.abelian₂ i)) : ¬ ListPin D Lc CF b

theorem **listPin_transfer** {D : D7} {I I' : Type} {Lc : LevCh D I} {Lc' : LevCh D I'} {μ0 : I'} (hL : Lc'.Is μ0) (hC : Lc'.C = Lc.C) {CF b : ℝ} (G : GridData Lc (1, CF, b)) (hF : FieldHas G (embLL0 D)) (hT : GlobalTheory.LamTie.IsLamTieFull D.ntw (lamOf G)) (P : Places Lc') (hv : P.v0 = G.pl.v0) : ListPin D Lc' CF b

theorem **listPin_iff_of_untwisted** {D : D7} {I : Type} (Lc : LevCh D I) (CF b : ℝ) (h : ∀ i, ¬ D.ntw.NonzeroIndexPlace i) : ListPin D Lc CF b ↔ ∃ G : GridData Lc (1, CF, b), FieldHas G (embLL0 D)

theorem **listPin_of_fit** (p : PtAt ex Csz) {CF : ℝ} {G : Grid p CF} (h : GridFit p G) : ListPin D p.R.toRealDat.toLevCh CF L.sc.b

theorem **fit_of_listPin** (p : PtAt ex Csz) {CF : ℝ} (hp : ListPin D p.R.toRealDat.toLevCh CF L.sc.b) {xd v0 rep} (ht : GlobalTheory.EX.TwistAt L ex.nch xd v0 rep) (h1 : ValLL0 D p.R.C) (h3 : Formalism.L2.Host.PsiChoiceCompatible D p.R.C ex.nch p.Tp) : ∃ G : Grid p CF, GridFit p G

theorem **settingsFormsExist_of_grid** : SettingsFormsExistList hL cDep → SettingsFormsExist hL

theorem **seriesExist_of_grid** : SeriesExistList hL cDep → SeriesExist hL

theorem **moduliPointsExist_of_grid** : ModuliPointsExistList hL cDep → SB2 hL

theorem **chain** (h0 : SettingsFormsExistList hL cDep) (h1 : SeriesExistList hL cDep) (h2 : ModuliPointsExistList hL cDep) (h3 : SC hL) : AllSt fun _ _ _ _ _ P x => LI hL x → ∃ x' : St P, Nonempty (LC hL x') ∧ ListSt cDep x'

theorem **lev_at_state** {K : Ctx} {B : GlobalTheory.Mem.Base} {f : Fr B.toPre} {T : TwDat B} {w : ColW B f T} {P : Pre2 K w} (x : St P) : x.m.z.R.toRealDat.toLevCh.C = cfOf x ∧ ∃ μ0, x.m.z.R.toRealDat.toLevCh.Is μ0

theorem **nonempty_grid** {B : Base} {f : Fr B.toPre} {T : TwDat B} (w : ColW B f T) (Z : w.Lt) {hX : B.fields.Ok} {Wt : Set B.places.Pl} (z : Dat (w.st Z) w.I hX T Wt) (sc : Consts) (h3 : ∀ k, q0 (w.st Z).toD7 ≠ B.host.g k) : Nonempty (GridData z.R.toRealDat.toLevCh sc)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonemptyRayDataOfNeZero

theorem **one_mul_mem** {R : Type} [CommRing R] {A : Ideal R} {a b : R} (ha : a - 1 ∈ A) (hb : b - 1 ∈ A) : a * b - 1 ∈ A

theorem **exists_beta** {M : ℕ} {α : 𝒪 K} (h : IsCoprime (Ideal.span {α}) (Ideal.span {(M : 𝒪 K)})) : ∃ β : 𝒪 K, β ≠ 0 ∧ β * α - 1 ∈ Ideal.span {(M : 𝒪 K)}

theorem **exists_compl** {M : ℕ} (I : primeTo K M) : ∃ L : primeTo K M, ∀ J : primeTo K M, icl J = icl I → ∃ α : 𝒪 K, J.1 * L.1 = Ideal.span {α}

theorem **finite_quot** {M : ℕ} (hM : M ≠ 0) : Finite (rayCon K M).Quotient

theorem **exists_inv** {M : ℕ} (q : (rayCon K M).Quotient) : IsUnit q

theorem **ne_bot_of_cop** {M : ℕ} (hM : 2 ≤ M) {I : Ideal (𝒪 K)} (h : IsCoprime I (Ideal.span {(M : 𝒪 K)})) : I ≠ ⊥

theorem **rcl_of** {M : ℕ} {I : Ideal (𝒪 K)} (h : I ∈ primeTo K M) : rcl M I = (rayCon K M).mk' (I, h)

theorem **rcl_mul** {M : ℕ} {I J : Ideal (𝒪 K)} (h1 : I ∈ primeTo K M) (h2 : J ∈ primeTo K M) : rcl M (I * J) = rcl M I * rcl M J

theorem **nonempty_rayData_of_ne_zero** (K : Type) [Field K] [NumberField K] (M : ℕ) (hM : M ≠ 0) : Nonempty (RayData K M)

theorem **cls_mul_of_two_le** (hM : 2 ≤ M) {I J : Ideal (𝒪 K)} (hI : IsCoprime I (Ideal.span {(M : 𝒪 K)})) (hJ : IsCoprime J (Ideal.span {(M : 𝒪 K)})) : D.cls (I * J) = D.cls I * D.cls J

theorem **cls_one** : D.cls 1 = 1

theorem **cls_top** : D.cls ⊤ = 1

theorem **cls_eq_one_iff** {I : Ideal (𝒪 K)} (hI : IsCoprime I (Ideal.span {(M : 𝒪 K)})) (hI' : I ≠ ⊥) : D.cls I = 1 ↔ ∃ a b : 𝒪 K, a ≠ 0 ∧ b ≠ 0 ∧ a - 1 ∈ Ideal.span {(M : 𝒪 K)} ∧ b - 1 ∈ Ideal.span {(M : 𝒪 K)} ∧ Ideal.span {a} * I = Ideal.span {b}

theorem **cls_span** {a : 𝒪 K} (ha : a ≠ 0) (h : a - 1 ∈ Ideal.span {(M : 𝒪 K)}) : D.cls (Ideal.span {a}) = 1

theorem **clsHom_surjective** : Function.Surjective D.clsHom

theorem **ker_clsHom** : Con.ker D.clsHom = RayCl.rayCon K M

theorem **equivRayGroup_rcl** {I : Ideal (𝒪 K)} (hI : IsCoprime I (Ideal.span {(M : 𝒪 K)})) (hI' : I ≠ ⊥) : D.equivRayGroup (RayCl.rcl M I) = D.cls I

theorem **finite_cnt** (c : D.G) (n : ℕ) : Finite {I : Ideal (𝒪 K) // IsCoprime I (Ideal.span {(M : 𝒪 K)}) ∧ Ideal.absNorm I = n ∧ D.cls I = c}

theorem **cnt_of_ne_zero** (c : D.G) {n : ℕ} (hn : n ≠ 0) : D.cnt c n = (Nat.card {I : Ideal (𝒪 K) // IsCoprime I (Ideal.span {(M : 𝒪 K)}) ∧ I ≠ ⊥ ∧ Ideal.absNorm I = n ∧ D.cls I = c} : ℕ)

theorem **cnt_zero** (hM : 2 ≤ M) (c : D.G) : D.cnt c 0 = 0

theorem **cnt_one** (c : D.G) : D.cnt c 1 = if c = 1 then 1 else 0

theorem **coef_one** (χ : D.G →* ℂˣ) : D.coef χ 1 = 1

theorem **card_eq_classNumber** (D : RayData K 1) : Fintype.card D.G = classNumber K

theorem **cls_two_ne_one** (D : RayData ℚ 5) : D.cls (Ideal.span {2}) ≠ 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonemptyTorusExtensionOfIsAlgClosed

@[**ext**] theorem ext {l m : Cochar G} (h : l.1 = m.1) : l = m

@[simp] theorem **val_add** (l m : Cochar G) : (l + m).1 = l.1 * m.1

@[simp] theorem **val_zero** : (0 : Cochar G).1 = 1

@[simp] theorem **val_neg** (l : Cochar G) : (-l).1 = l.1⁻¹

@[simp] theorem **val_zsmul** (n : ℤ) (l : Cochar G) : (n • l).1 = l.1 ^ n

@[simp] theorem **map_val** (u : G ⟶ H) [IsMonHom u] (l : Cochar G) : (map u l).1 = l.1 ≫ u

theorem **map_id** (l : Cochar G) : map (𝟙 G) l = l

theorem **map_comp** (u : G ⟶ H) (v : H ⟶ K) [IsMonHom u] [IsMonHom v] (l : Cochar G) : map (u ≫ v) l = map v (map u l)

@[simp] theorem **actHom_val** {O : Type} [CommRing O] (a : OAct O G) (c : O) (l : Cochar G) : (actHom a c l).1 = l.1 ≫ a.act c

theorem **pullTorus_left** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) : (pullTorus h i).left ≫ φ = bcMap f X ≫ i.left

theorem **hom_ext_of_pullback** (h : IsGrpPullback G' G f φ) {T : Over S'} {x y : T ⟶ G'} (e : x.left ≫ φ = y.left ≫ φ) : x = y

theorem **eq_pullTorus_iff** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) (i' : splitTorus S' X ⟶ G') : i' = pullTorus h i ↔ i'.left ≫ φ = bcMap f X ≫ i.left

theorem **pullTorus_mul** (h : IsGrpPullback G' G f φ) (i j : splitTorus S X ⟶ G) : pullTorus h (i * j) = pullTorus h i * pullTorus h j

theorem **bcMap_id** : bcMap (𝟙 S) X = 𝟙 _

theorem **bcMap_comp** (g : S'' ⟶ S') (f : S' ⟶ S) : bcMap (g ≫ f) X = bcMap g X ≫ bcMap f X

theorem **pullTorus_pullTorus** {g : S'' ⟶ S'} {φ' : G''.left ⟶ G'.left} {f'' : S'' ⟶ S} {φ'' : G''.left ⟶ G.left} (h' : IsGrpPullback G'' G' g φ') (h : IsGrpPullback G' G f φ) (h'' : IsGrpPullback G'' G f'' φ'') (ef : f'' = g ≫ f) (eφ : φ'' = φ' ≫ φ) (i : splitTorus S X ⟶ G) : pullTorus h' (pullTorus h i) = pullTorus h'' i

theorem **pullTorus_torusMap_comp** {Y : Type} [AddCommGroup Y] (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) (u : X →+ Y) : pullTorus h (torusMap S u ≫ i) = torusMap S' u ≫ pullTorus h i

theorem **pullTorus_comp** {H : Over S} [GrpObj H] {H' : Over S'} [GrpObj H'] {ψ : H'.left ⟶ H.left} (h : IsGrpPullback G' G f φ) (k : IsGrpPullback H' H f ψ) (u : G ⟶ H) (u' : G' ⟶ H') (e : u'.left ≫ ψ = φ ≫ u.left) (i : splitTorus S X ⟶ G) : pullTorus k (i ≫ u) = pullTorus h i ≫ u'

theorem **exists_isKernel_pullTorus** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) {B : Over S} [GrpObj B] (q : G ⟶ B) [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) : ∃ q' : G' ⟶ (Over.pullback f).obj B, IsMonHom q' ∧ IsPullback (pullTorus h i) (toUnit _) q'

η[(Over.pullback f).obj B]

theorem **pull_val** (h : IsGrpPullback G' G f φ) (l : Cochar G) : (pull h l).1 = pullTorus h l.1

theorem **pull_left** (h : IsGrpPullback G' G f φ) (l : Cochar G) : (pull h l).1.left ≫ φ = bcMap f (ULift ℤ) ≫ l.1.left

theorem **eq_pull_iff** (h : IsGrpPullback G' G f φ) (l : Cochar G) (l' : Cochar G') : l' = pull h l ↔ l'.1.left ≫ φ = bcMap f (ULift ℤ) ≫ l.1.left

theorem **pull_id** (h : IsGrpPullback G G (𝟙 S) (𝟙 G.left)) (l : Cochar G) : pull h l = l

theorem **pull_pull** {g : S'' ⟶ S'} {φ' : G''.left ⟶ G'.left} {f'' : S'' ⟶ S} {φ'' : G''.left ⟶ G.left} (h' : IsGrpPullback G'' G' g φ') (h : IsGrpPullback G' G f φ) (h'' : IsGrpPullback G'' G f'' φ'') (ef : f'' = g ≫ f) (eφ : φ'' = φ' ≫ φ) (l : Cochar G) : pull h' (pull h l) = pull h'' l

theorem **pull_comp** {g : S'' ⟶ S'} {φ' : G''.left ⟶ G'.left} (h' : IsGrpPullback G'' G' g φ') (h : IsGrpPullback G' G f φ) (l : Cochar G) : pull (h'.comp h) l = pull h' (pull h l)

theorem **pull_map** {H : Over S} [GrpObj H] [IsCommMonObj H] {H' : Over S'} [GrpObj H'] [IsCommMonObj H'] {ψ : H'.left ⟶ H.left} (h : IsGrpPullback G' G f φ) (k : IsGrpPullback H' H f ψ) (u : G ⟶ H) (u' : G' ⟶ H') [IsMonHom u] [IsMonHom u'] (e : u'.left ≫ ψ = φ ≫ u.left) (l : Cochar G) : pull k (map u l) = map u' (pull h l)

theorem **pull_actHom** {O : Type} [CommRing O] (h : IsGrpPullback G' G f φ) (a : OAct O G) (a' : OAct O G') (e : ∀ c, (a'.act c).left ≫ φ = φ ≫ (a.act c).left) (c : O) (l : Cochar G) : pull h (actHom a c l) = actHom a' c (pull h l)

theorem **pull_of_iso** {H : Over S} [GrpObj H] [IsCommMonObj H] (e : H ≅ G) [IsMonHom e.hom] (l : Cochar G) : pull (isGrpPullback_of_iso e) l = map e.inv l

theorem **pull_injective** [QuasiCompact f] [IsSchemeTheoreticallyDominant f] [IsSeparated G.hom] (h : IsGrpPullback G' G f φ) : Function.Injective (pull h)

@[simp] theorem **comap_apply** (u : G ⟶ H) [IsMonHom u] (χ : Char H) (l : Cochar G) : comap u χ l = χ (Cochar.map u l)

@[simp] theorem **pull_apply** (h : IsGrpPullback G' G f φ) (χ : Char G') (l : Cochar G) : pull h χ l = χ (Cochar.pull h l)

theorem **pull_id** (h : IsGrpPullback G G (𝟙 S) (𝟙 G.left)) (χ : Char G) : pull h χ = χ

theorem **pull_pull** {g : S'' ⟶ S'} {φ' : G''.left ⟶ G'.left} {f'' : S'' ⟶ S} {φ'' : G''.left ⟶ G.left} (h' : IsGrpPullback G'' G' g φ') (h : IsGrpPullback G' G f φ) (h'' : IsGrpPullback G'' G f'' φ'') (ef : f'' = g ≫ f) (eφ : φ'' = φ' ≫ φ) (χ : Char G'') : pull h (pull h' χ) = pull h'' χ

theorem **pull_comp** {g : S'' ⟶ S'} {φ' : G''.left ⟶ G'.left} (h' : IsGrpPullback G'' G' g φ') (h : IsGrpPullback G' G f φ) (χ : Char G'') : pull (h'.comp h) χ = pull h (pull h' χ)

theorem **pull_bijective** {h : IsGrpPullback G' G f φ} (hb : Function.Bijective (Cochar.pull h)) : Function.Bijective (pull h)

@[simp] theorem **smul_val** (a : 𝒪 E) (l : CocharOf t) : (a • l).1 = l.1 ≫ t.act.act a

theorem **pull_apply** (m : SABMap t' t f) (l : CocharOf t) : pull m l = Cochar.pull m.cart l

@[**ext**] theorem ext {χ χ' : CharOf t} (h : ∀ l, χ l = χ' l) : χ = χ'

@[simp] theorem **add_apply** (χ χ' : CharOf t) (l : CocharOf t) : (χ + χ') l = χ l + χ' l

@[simp] theorem **zero_apply** (l : CocharOf t) : (0 : CharOf t) l = 0

@[simp] theorem **smul_apply** (a : 𝒪 E) (χ : CharOf t) (l : CocharOf t) : (a • χ) l = χ (H.bar a • l)

theorem **comap_act** (a : 𝒪 E) (χ : CharOf t) : (haveI

@[simp] theorem **pull_apply** (m : SABMap t' t f) (χ : CharOf t') (l : CocharOf t) : pull m χ l = χ (CocharOf.pull m l)

theorem **pull_pull** {t'' : SAB H S''} {g : S'' ⟶ S'} {f'' : S'' ⟶ S} (m' : SABMap t'' t' g) (m : SABMap t' t f) (m'' : SABMap t'' t f'') (ef : f'' = g ≫ f) (eφ : m''.φ = m'.φ ≫ m.φ) (χ : CharOf t'') : pull m (pull m' χ) = pull m'' χ

theorem **torusCochar_appTop_chi** (l : X →+ ℤ) (x : X) : (torusCochar S l).left.appTop.hom (Rosenlicht.chi X x) = Rosenlicht.chi (ULift.{0} ℤ) (ULift.up (l x))

theorem **torusCochar_comp** (l : Y →+ ℤ) (u : X →+ Y) : torusCochar S (l.comp u) = torusCochar S l ≫ torusMap S u

theorem **torusCochar_add** (l m : X →+ ℤ) : torusCochar S (l + m) = torusCochar S l ∗ torusCochar S m

@[simp] theorem **ofTorus_val** (i : splitTorus S X ⟶ G) [IsMonHom i] (l : X →+ ℤ) : (ofTorus i l).1 = torusCochar S l ≫ i

theorem **pull_ofTorus** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) [IsMonHom i] (l : X →+ ℤ) : Cochar.pull h (ofTorus i l) = ofTorus (pullTorus h i) l

theorem **pull_ofTorus_of_left** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) [IsMonHom i] (i' : splitTorus S' X ⟶ G') [IsMonHom i'] (e : i'.left ≫ φ = bcMap f X ≫ i.left) (l : X →+ ℤ) : Cochar.pull h (ofTorus i l) = ofTorus i' l

theorem **map_ofTorus** {H : Over S} [GrpObj H] [IsCommMonObj H] (u : G ⟶ H) [IsMonHom u] (i : splitTorus S X ⟶ G) [IsMonHom i] (l : X →+ ℤ) : Cochar.map u (ofTorus i l) = ofTorus (i ≫ u) l

theorem **ofTorus_torusMap_comp** (i : splitTorus S X ⟶ G) [IsMonHom i] (u : X →+ Y) (l : Y →+ ℤ) : ofTorus (torusMap S u ≫ i) l = ofTorus i (l.comp u)

theorem **ofTorus_injective** [Nonempty S] (i : splitTorus S X ⟶ G) [IsMonHom i] [Mono i] : Function.Injective (ofTorus i)

theorem **mem_range_ofTorus** [ConnectedSpace S] (i : splitTorus S X ⟶ G) [IsMonHom i] (l : Cochar G) (u : Gm S ⟶ splitTorus S X) [IsMonHom u] (e : u ≫ i = l.1) : ∃ m, ofTorus i m = l

theorem **ofTorus_id_bijective** [ConnectedSpace S] : Function.Bijective (ofTorus (𝟙 (splitTorus S X)))

theorem **ofTorus_bijective_of_isKernel** [ConnectedSpace S] (i : splitTorus S X ⟶ G) [IsMonHom i] {B : Over S} [GrpObj B] (q : G ⟶ B) [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) (hB : ∀ u : Gm S ⟶ B, IsMonHom u → u = toUnit _ ≫ η[B]) : Function.Bijective (ofTorus i)

theorem **subsingleton_cochar_of_isProper** {A : Type} [CommRing A] [IsLocalRing A] [IsNoetherianRing A] (B : Over (Sp A)) [GrpObj B] [IsProper B.hom] : Subsingleton (Cochar B)

theorem **ofTorus_bijective_of_isProper** {A : Type} [CommRing A] [IsLocalRing A] [IsNoetherianRing A] {G : Over (Sp A)} [GrpObj G] [IsCommMonObj G] (i : splitTorus (Sp A) X ⟶ G) [IsMonHom i] {B : Over (Sp A)} [GrpObj B] [IsProper B.hom] (q : G ⟶ B) [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) : Function.Bijective (ofTorus i)

theorem **ofTorus_pullTorus_bijective** {A : Type} [CommRing A] [IsLocalRing A] [IsNoetherianRing A] {G₀ : Over (Sp A)} [GrpObj G₀] [IsCommMonObj G₀] {p : Sp A ⟶ S} {ψ : G₀.left ⟶ G.left} (h : IsGrpPullback G₀ G p ψ) (i : splitTorus S X ⟶ G) [IsMonHom i] {B : Over

S} [GrpObj B] [IsProper B.hom] (q : G ⟶ B) [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) : Function.Bijective (ofTorus (pullTorus h i))

theorem **pull_bijective_of_ofTorus** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) [IsMonHom i] (hi : Function.Bijective (ofTorus i)) (hi' : Function.Bijective (ofTorus (pullTorus h i))) : Function.Bijective (Cochar.pull h)

theorem **ofTorus_bijective_of_torusExtension** (T : TorusExtension G) : letI

theorem **Cochar.pull_bijective_of_torusExtension** (hT : Nonempty (TorusExtension G)) {A : Type} [CommRing A] [IsLocalRing A] [IsNoetherianRing A] {G₀ : Over (Sp A)} [GrpObj G₀] [IsCommMonObj G₀] {p : Sp A ⟶ Sp K} {ψ : G₀.left ⟶ G.left} (h : IsGrpPullback G₀ G p ψ) : Function.Bijective (Cochar.pull h)

theorem **Char.pull_bijective_of_torusExtension** (hT : Nonempty (TorusExtension G)) {A : Type} [CommRing A] [IsLocalRing A] [IsNoetherianRing A] {G₀ : Over (Sp A)} [GrpObj G₀] [IsCommMonObj G₀] {p : Sp A ⟶ Sp K} {ψ : G₀.left ⟶ G.left} (h : IsGrpPullback G₀ G p ψ) : Function.Bijective (Char.pull h)

theorem **nonempty_torusExtension_of_isAlgClosed** {Ω : Type} [Field Ω] [IsAlgClosed Ω] {G : Over (Sp Ω)} [GrpObj G] (h : IsSemiAbelian G) : Nonempty (TorusExtension G)

theorem **CharOf.pull_bijective_of_isAlgClosed** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Ω A : Type} [Field Ω] [IsAlgClosed Ω] [CommRing A] [IsLocalRing A] [IsNoetherianRing A] {t : SAB H (Sp Ω)} {t' : SAB H (Sp A)} {p : Sp A ⟶ Sp Ω} (m : SABMap t' t p) : Function.Bijective (CharOf.pull m)

theorem **eval_bijective** (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] : Function.Bijective (AddMonoidHom.eval : X →+ (X →+ ℤ) →+ ℤ)

@[simp] theorem **charEquivOfTorus_apply** (i : splitTorus S X ⟶ G) [IsMonHom i] (hi : Function.Bijective (ofTorus i)) (x : X) (l : X →+ ℤ) : charEquivOfTorus i hi x (ofTorus i l) = l x

theorem **eq_charEquivOfTorus_iff** (i : splitTorus S X ⟶ G) [IsMonHom i] (hi : Function.Bijective (ofTorus i)) (x : X) (χ : Char G) : χ = charEquivOfTorus i hi x ↔ ∀ l, χ (ofTorus i l) = l x

theorem **pull_charEquivOfTorus** (h : IsGrpPullback G' G f φ) (i : splitTorus S X ⟶ G) [IsMonHom i] (hi : Function.Bijective (ofTorus i)) (hi' : Function.Bijective (ofTorus (pullTorus h i))) (x : X) : Char.pull h (charEquivOfTorus _ hi' x) = charEquivOfTorus i hi x

theorem **ofTorus_bijective_of_gtCoord** [IsLocalRing R] [IsNoetherianRing R] (hI : I ≠ ⊤) (g : Out.GtCoord I D e G) : haveI

theorem **thickPt_thick** (hρ : ∀ x ∈ I ^ (e + 1), ρ x = 0) : thickPt e hρ ≫ Out.thick R I e = spMap ρ

theorem **thickLift_fst** (hρ : ∀ x ∈ I ^ (e + 1), ρ x = 0) (h : IsGrpPullback G₀ G (spMap ρ) φ) : thickLift hρ h ≫ pullback.fst G.hom (Out.thick R I e) = φ

theorem **isGrpPullback_thickLift** (hρ : ∀ x ∈ I ^ (e + 1), ρ x = 0) (h : IsGrpPullback G₀ G (spMap ρ) φ) : IsGrpPullback G₀ (Out.thickObj R I e G) (thickPt e hρ) (thickLift hρ h)

theorem **gtTorus_left** (g : Out.GtCoord I D e G) (hρ : ∀ x ∈ I ^ (e + 1), ρ x = 0) (h : IsGrpPullback G₀ G (spMap ρ) φ) : (gtTorus g hρ h).left ≫ φ = bcMap (thickPt e hρ) D.X ≫ g.i.left ≫ pullback.fst G.hom (Out.thick R I e)

theorem **pullTorus_gtTorus** {A' : Type} [CommRing A'] {σ : A →+* A'} {ρ' : R →+* A'} {G₁ : Over (Sp A')} [GrpObj G₁] {ψ : G₁.left ⟶ G₀.left} {φ' : G₁.left ⟶ G.left} (g : Out.GtCoord I D e G) (hρ : ∀ x ∈ I ^ (e + 1), ρ x = 0) (hρ' : ∀ x ∈ I ^ (e + 1), ρ' x = 0) (h : IsGrpPullback G₀ G (spMap ρ) φ) (h' : IsGrpPullback G₁ G (spMap ρ') φ') (k : IsGrpPullback G₁ G₀ (spMap σ) ψ) (eρ : ρ' = σ.comp ρ) (eφ : φ' = ψ ≫ φ) : pullTorus k (gtTorus g hρ h) = gtTorus g hρ' h'

theorem **ofTorus_gtTorus_bijective** [IsLocalRing A] [IsNoetherianRing A] [IsCommMonObj G₀] (g : Out.GtCoord I D e G) (hρ : ∀ x ∈ I ^ (e + 1), ρ x = 0) (h : IsGrpPullback G₀ G (spMap ρ) φ) : Function.Bijective (ofTorus (gtTorus g hρ h))

theorem **map_pow_one_eq_zero** (hρ : ∀ x ∈ I, ρ x = 0) : ∀ x ∈ I ^ (0 + 1), ρ x = 0

theorem **torus_comp_act** (P : PELUnif I t) (hρ : ∀ x ∈ I, ρ x = 0) (m : SABMap t₀ t (spMap ρ)) (a : 𝒪 E) : torus P hρ m ≫ t₀.act.act a = torusMap (Sp A) (DistribSMul.toAddMonoidHom P.D.toPre.X (H.bar a)) ≫ torus P hρ m

theorem **smul_cochar** (P : PELUnif I t) (hρ : ∀ x ∈ I, ρ x = 0) (m : SABMap t₀ t (spMap ρ)) (a : 𝒪 E) (l : P.D.toPre.X →+ ℤ) : a • cochar P hρ m l = cochar P hρ m (l.comp (DistribSMul.toAddMonoidHom P.D.toPre.X (H.bar a)))

theorem **cochar_bijective** (P : PELUnif I t) (hρ : ∀ x ∈ I, ρ x = 0) (m : SABMap t₀ t (spMap ρ)) : Function.Bijective (cochar P hρ m)

@[simp] theorem **charEquivOfKer_apply** (P : PELUnif I t) (hρ : ∀ x ∈ I, ρ x = 0) (m : SABMap t₀ t (spMap ρ)) (x : P.D.toPre.X) (l : P.D.toPre.X →+ ℤ) : charEquivOfKer P hρ m x (cochar P hρ m l) = l x

theorem **pull_charEquivOfKer** {A' : Type} [CommRing A'] [IsLocalRing A'] [IsNoetherianRing A'] {σ : A →+* A'} {ρ' : R →+* A'} {t₁ : SAB H (Sp A')} (P : PELUnif I t) (hρ : ∀ x ∈ I, ρ x = 0) (hρ' : ∀ x ∈ I, ρ' x = 0) (m : SABMap t₀ t (spMap ρ)) (m' : SABMap t₁ t (spMap ρ')) (n : SABMap t₁ t₀ (spMap σ)) (eρ : ρ' = σ.comp ρ) (eφ : m'.φ = n.φ ≫ m.φ) (x : P.D.toPre.X) : CharOf.pull n (charEquivOfKer P hρ' m' x) = charEquivOfKer P hρ m x

theorem **map_eq_zero_of_isLocalHom** [IsLocalRing R] {k : Type} [Field k] (ρ : R →+* k) [IsLocalHom ρ] : ∀ x ∈ IsLocalRing.maximalIdeal R, ρ x = 0

@[simp] theorem **PELUnif.charEquiv_apply** [IsLocalRing R] {k : Type} [Field k] {ρ : R →+* k} [IsLocalHom ρ] {t₀ : SAB H (Sp k)} (P : PELUnif (IsLocalRing.maximalIdeal R) t) (m : SABMap t₀ t (spMap ρ)) (x : P.D.toPre.X) (l : P.D.toPre.X →+ ℤ) : PELUnif.charEquiv P m x (PELUnif.cochar P (map_eq_zero_of_isLocalHom ρ) m l) = l x

theorem **isSpec_pull** (h₁ : IsGrpPullback G₁ G f₁ φ₁) (h₀ : IsGrpPullback G₀ G f₀ φ₀) (l : Cochar G) : IsSpec h₁ h₀ (Cochar.pull h₁ l) (Cochar.pull h₀ l)

theorem **IsSpec.add** {h₁ : IsGrpPullback G₁ G f₁ φ₁} {h₀ : IsGrpPullback G₀ G f₀ φ₀} {l₁ m₁ : Cochar G₁} {l₀ m₀ : Cochar G₀} (a : IsSpec h₁ h₀ l₁ l₀) (b : IsSpec h₁ h₀ m₁ m₀) : IsSpec h₁ h₀ (l₁ + m₁) (l₀ + m₀)

theorem **IsSpec.unique** {h₁ : IsGrpPullback G₁ G f₁ φ₁} {h₀ : IsGrpPullback G₀ G f₀ φ₀} (hi : Function.Injective (Cochar.pull h₁)) {l₁ : Cochar G₁} {l₀ m₀ : Cochar G₀} (a : IsSpec h₁ h₀ l₁ l₀) (b : IsSpec h₁ h₀ l₁ m₀) : l₀ = m₀

@[simp] theorem **sp_pull** (l : Cochar G) : sp h₁ h₀ hb (Cochar.pull h₁ l) = Cochar.pull h₀ l

theorem **isSpec_iff** (l₁ : Cochar G₁) (l₀ : Cochar G₀) : IsSpec h₁ h₀ l₁ l₀ ↔ sp h₁ h₀ hb l₁ = l₀

@[simp] theorem **res_apply** (χ : Char G₀) (l : Cochar G₁) : res h₁ h₀ hb χ l = χ (sp h₁ h₀ hb l)

theorem **pull_res** (χ : Char G₀) : Char.pull h₁ (res h₁ h₀ hb χ) = Char.pull h₀ χ

theorem **eq_res_iff** (χ : Char G₀) (χ₁ : Char G₁) : χ₁ = res h₁ h₀ hb χ ↔ Char.pull h₁ χ₁ = Char.pull h₀ χ

theorem **sp_pull_comm** {V' S₁' S₀' : Scheme.{0}} {G' : Over V'} [GrpObj G'] [IsCommMonObj G'] {G₁' : Over S₁'} [GrpObj G₁'] [IsCommMonObj G₁'] {G₀' : Over S₀'} [GrpObj G₀'] [IsCommMonObj G₀'] {f₁' : S₁' ⟶ V'} {f₀' : S₀' ⟶ V'} {φ₁' : G₁'.left ⟶ G'.left} {φ₀' : G₀'.left ⟶

G'.left} (h₁' : IsGrpPullback G₁' G' f₁' φ₁') (h₀' : IsGrpPullback G₀' G' f₀' φ₀') (hb' : Function.Bijective (Cochar.pull h₁')) {v : V' → V} {s₁ : S₁' → S₁} {s₀ : S₀' → S₀} {ψ : G'.left → G.left} {ψ₁ : G₁'.left → G₁.left} {ψ₀ : G₀'.left → G₀.left} (k : IsGrpPullback G' G v ψ) (k₁ : IsGrpPullback G₁' G₁ s₁ ψ₁) (k₀ : IsGrpPullback G₀' G₀ s₀ ψ₀) (e₁ : f₁' ≫ v = s₁ ≫ f₁) (e₀ : f₀' ≫ v = s₀ ≫ f₀) (c₁ : φ₁' ≫ ψ = ψ₁ ≫ φ₁) (c₀ : φ₀' ≫ ψ = ψ₀ ≫ φ₀) (l : Cochar G₁) : sp h₁' h₀' hb' (Cochar.pull k₁ l) = Cochar.pull k₀ (sp h₁ h₀ hb l)

theorem **pull_res_comm** {V' S₁' S₀' : Scheme.{0}} {G' : Over V'} [GrpObj G'] [IsCommMonObj G'] {G₁' : Over S₁'} [GrpObj G₁'] [IsCommMonObj G₁'] {G₀' : Over S₀'} [GrpObj G₀'] [IsCommMonObj G₀'] {f₁' : S₁' → V'} {f₀' : S₀' → V'} {φ₁' : G₁'.left → G'.left} {φ₀' : G₀'.left → G'.left} (h₁' : IsGrpPullback G₁' G' f₁' φ₁') (h₀' : IsGrpPullback G₀' G' f₀' φ₀') (hb' : Function.Bijective (Cochar.pull h₁')) {v : V' → V} {s₁ : S₁' → S₁} {s₀ : S₀' → S₀} {ψ : G'.left → G.left} {ψ₁ : G₁'.left → G₁.left} {ψ₀ : G₀'.left → G₀.left} (k : IsGrpPullback G' G v ψ) (k₁ : IsGrpPullback G₁' G₁ s₁ ψ₁) (k₀ : IsGrpPullback G₀' G₀ s₀ ψ₀) (e₁ : f₁' ≫ v = s₁ ≫ f₁) (e₀ : f₀' ≫ v = s₀ ≫ f₀) (c₁ : φ₁' ≫ ψ = ψ₁ ≫ φ₁) (c₀ : φ₀' ≫ ψ = ψ₀ ≫ φ₀) (χ : Char G₀') : Char.pull k₁ (res h₁' h₀' hb' χ) = res h₁ h₀ hb (Char.pull k₀ χ)

theorem **pull_bijective_of_torus** {Y : Type} [AddCommGroup Y] (j : splitTorus V Y → G) [IsMonHom j] (hj : Function.Surjective (ofTorus (pullTorus h₁ j))) (hi : Function.Injective (Cochar.pull h₁)) : Function.Bijective (Cochar.pull h₁)

theorem **res_charEquivOfTorus** {X Y : Type} [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] [AddCommGroup Y] [Module.Free ℤ Y] [Module.Finite ℤ Y] (j : splitTorus V Y → G) [IsMonHom j] (hj : Function.Bijective (ofTorus (pullTorus h₁ j))) (i : splitTorus S₀ X → G₀) [IsMonHom i] (hi : Function.Bijective (ofTorus i)) (r : X →+ Y) (e : pullTorus h₀ j = torusMap S₀ r ≫ i) (x : X) : res h₁ h₀ hb (charEquivOfTorus i hi x) = charEquivOfTorus _ hj (r x)

theorem **CocharOf.sp_apply** (m₁ : SABMap t₁ t f₁) (m₀ : SABMap t₀ t f₀) (hb : Function.Bijective (CocharOf.pull m₁)) (l : CocharOf t₁) : CocharOf.sp m₁ m₀ hb l = TorusPart.sp m₁.cart m₀.cart hb l

@[simp] theorem **CharOf.res_apply** (m₁ : SABMap t₁ t f₁) (m₀ : SABMap t₀ t f₀) (hb : Function.Bijective (CocharOf.pull m₁)) (χ : CharOf t₀) (l : CocharOf t₁) : CharOf.res m₁ m₀ hb χ l = χ (CocharOf.sp m₁ m₀ hb l)

theorem **CharOf.pull_res** (m₁ : SABMap t₁ t f₁) (m₀ : SABMap t₀ t f₀) (hb : Function.Bijective (CocharOf.pull m₁)) (χ : CharOf t₀) : CharOf.pull m₁ (CharOf.res m₁ m₀ hb χ) = CharOf.pull m₀ χ

theorem **charEquivSplitTorus_id** [ConnectedSpace S] (n : ℤ) : charEquivSplitTorus S (ULift.{0} ℤ) (ULift.up n) ⟨𝟙 (Gm S), inferInstance⟩ = n

theorem **res_splitTorus** {V S₁ S₀ : Scheme.{0}} [ConnectedSpace V] [ConnectedSpace S₁] [ConnectedSpace S₀] (f₁ : S₁ → V) (f₀ : S₀ → V) (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] : ∃ hb : Function.Bijective (Cochar.pull (SplitTorus.isGrpPullback_bcMap f₁ X)), ∀ x : X, res _ (SplitTorus.isGrpPullback_bcMap f₀ X) hb (charEquivSplitTorus S₀ X x) = charEquivSplitTorus S₁ X x

theorem **exists_rank_drop** : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : Host E) (t : SAB H (Sp (PowerSeries ℂ))) (t₁ : SAB H (Sp (FractionRing (PowerSeries ℂ)))) (t₀ : SAB H (Sp ℂ)) (m₁ : SABMap t₁ t (spMap (algebraMap _ _))) (_ : SABMap t₀ t (spMap PowerSeries.constantCoeff)), Nontrivial (CharOf t₀) ∧ Nontrivial (CocharOf t₀) ∧ Subsingleton (CocharOf t₁) ∧ Subsingleton (CharOf t₁) ∧ Function.Bijective (CocharOf.pull m₁)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonnegAdmissibleIffPositiveDefinite

theorem **IsFat.bound** {k : ℕ} (h : ∀ γ ∈ P, ∀ δ : L, ∃ C : ℕ, ∀ N : ℕ, C ≤ N → N • γ + k • δ ∈ P) {γ : L} (hγ : γ ∈ P) (δ : L) : ∃ C : ℕ, (∀ N : ℕ, C ≤ N → N • γ + k • δ ∈ P) ∧ ∀ χ, Nonneg P χ → |k * χ δ| ≤ C * χ γ

theorem **IsFat.finite** [Module.Finite ℤ L] (h : IsFat P) {γ : L} (hγ : γ ∈ P) (n : ℤ) : {χ : Module.Dual ℤ L | Nonneg P χ ∧ χ γ = n}.Finite

theorem **exists_forall_ne_zero** {T : Set (Module.Dual ℤ L)} (hT : T.Finite) (h : ∀ φ ∈ T, φ ≠ 0) : ∃ δ : L, ∀ φ ∈ T, φ δ ≠ 0

theorem **IsFat.exists_isolates** [Module.Finite ℤ L] (h : IsFat P) (hne : P.Nonempty) (hχ : Nonneg P χ) : ∃ γ, Isolates P γ χ

theorem **IsExpansion.coeff_eq** {c : Module.Dual ℤ L → K} (h : IsExpansion P Val c) {γ : L} (hγ : Isolates P γ χ) {r : LaurentSeries K} (hr : Val γ r) : r.coeff (χ γ) = c χ

theorem **coefNorm_eq_zero** (h : ¬ Nonneg P χ) : coefNorm P Val χ = 0

theorem **coefSet_nonempty** [Module.Finite ℤ L] (h : IsFat P) (hne : P.Nonempty) (hV : ∀ γ ∈ P, ∃ r, Val γ r) (hχ : Nonneg P χ) : (coefSet P Val χ).Nonempty

theorem **IsExpansion.coefNorm_eq** [Module.Finite ℤ L] {c : Module.Dual ℤ L → K} (hc : IsExpansion P Val c) (h : IsFat P) (hne : P.Nonempty) (hV : ∀ γ ∈ P, ∃ r, Val γ r) (χ : Module.Dual ℤ L) : coefNorm P Val χ = ‖c χ‖

theorem **IsExpansion.unique** [Module.Finite ℤ L] {c c' : Module.Dual ℤ L → K} (hc : IsExpansion P Val c) (hc' : IsExpansion P Val c') (h : IsFat P) (hne : P.Nonempty) (hV : ∀ γ ∈ P, ∃ r, Val γ r) : c = c'

theorem **coefSet_le** {B : ℝ} (h : ∀ γ ∈ P, ∀ r, Val γ r → ∀ n, ‖r.coeff n‖ ≤ B) : ∀ x ∈ coefSet P Val χ, x ≤ B

theorem **coefNorm_le** {B : ℝ} (hB : 0 ≤ B) (h : ∀ γ ∈ P, ∀ r, Val γ r → ∀ n, ‖r.coeff n‖ ≤ B) : coefNorm P Val χ ≤ B

theorem **coefNorm_equiv** (ε : L ≃ₗ[ℤ] L') {P' : Set L'} {Val' : L' → LaurentSeries K → Prop} (hP : ∀ γ, ε γ ∈ P' ↔ γ ∈ P) (hV : ∀ γ r, Val' (ε γ) r ↔ Val γ r) (χ' : Module.Dual ℤ L') : coefNorm P Val (ε.dualMap χ') = coefNorm P' Val' χ'

theorem **isFat_pos** : IsFat {n : ℤ | 0 < n}

theorem **isExpansion_single** [DecidableEq (Module.Dual ℤ L)] {χ₀ : Module.Dual ℤ L} (h : Nonneg P χ₀) (a : K) : IsExpansion P (fun γ r => r = HahnSeries.single (χ₀ γ) a) (fun χ => if χ = χ₀ then a else 0)

theorem **exists_sq_le** {β : LinearMap.BilinForm ℤ X} (hs : ∀ y z, β y z = β z y) (hp : ∀ y, y ≠ 0 → 0 < β y y) (φ : Module.Dual ℤ X) : ∃ C : ℕ, ∀ y, φ y ^ 2 ≤ C * β y y

theorem **exists_abs_le** {β : LinearMap.BilinForm ℤ X} (hs : ∀ y z, β y z = β z y) (hp : ∀ y, y ≠ 0 → 0 < β y y) (δ : LinearMap.BilinForm ℤ X) : ∃ C : ℕ, ∀ y, |δ y y| ≤ C * β y y

theorem **isFat_adm** : IsFat (admitted H N X)

theorem **nonneg_admissible_iff_positiveDefinite** (χ : Module.Dual ℤ (levelForms H N X)) : Nonneg (admitted H N X) χ ↔ ∀ γ : levelForms H N X, (∀ y, y ≠ 0 → 0 < (γ : LinearMap.BilinForm ℤ X) y y) → 0 ≤ χ γ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NontrivialReferenceForms

theorem **repr_j** (x : H.V) (i : H.ι) : c.repr (j x) i = σ (H.b.repr x i)

theorem **exists_coord** (hX : ∀ v, IsUnit (H.X v).det) : ∃ e : M ≃ₗ[ℂ] EuclideanSpace ℂ H.ι, ∀ x, Formalism.HermitianLatticesTheory.normAt H.b H.X (InfinitePlace.mk σ) x = ‖e (j x)‖

theorem **exists_olat** (hG : H.Good) {O : Type} [CommRing O] (ι : O →+* K) (hι : ∀ y, IsIntegral ℤ (ι y)) {N : Type} [AddCommGroup N] [Module ℂ N] (e : M ≃ₗ[ℂ] N) : ∃ L : Rows.OLat O (σ.comp ι) N, (∀ y, y ∈ L.carrier ↔ ∃ x ∈ H.L, e (j x) = y) ∧ ∀ y ∈ L.gen, y ∈ L.carrier

theorem **pack** {O : Type} [CommRing O] (ι : O →+* X.K) (hι : ∀ y, IsIntegral ℤ (ι y)) : ∃ (n : ℕ) (e : MtP W Ch sel mu P σ ≃ₗ[ℂ] Eu n) (L : Rows.OLat O (σ.comp ι) (Eu n)), (∀ x, Formalism.HermitianLatticesTheory.normAt (MtildeH W Ch hX sel mu P).b (MtildeH W Ch hX sel mu P).X (InfinitePlace.mk σ) x = ‖e (toMtP W Ch sel mu P σ x)‖) ∧ (∀ y, y ∈ L.carrier ↔ ∃ x ∈ (MtildeH W Ch hX sel mu P).L, e (toMtP W Ch sel mu P σ x) = y) ∧ ∀ y ∈ L.gen, y ∈ L.carrier

theorem **basis_cx** : LinearIndependent ℂ (fun i => toMtP W Ch sel mu P σ ((MtildeH W Ch hX sel mu P).b i)) ∧ Submodule.span ℂ (Set.range fun i => toMtP W Ch sel mu P σ ((MtildeH W Ch hX sel mu P).b i)) = ⊤

theorem **conj_eig** {G V : Type} [Group G] [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (H H' : Subgroup G) (χ : H → ℂ) (χ' : H' → ℂ) (t : G) (h : ∀ a : H', ∃ a' : H, a'.1 * t = t * a.1 ∧ χ' a = χ a') (F : V) (hF : F ≠ 0) (he : ∀ a : H, ρ a.1 F = χ a • F) : ∃ F' : V, F' ≠ 0 ∧ ∀ a : H', ρ a.1 F' = χ' a • F'

theorem **exists_perm** (n k rS : ℕ) (κ : Type) [CommRing κ] : ∃ p : GL (Fin n) κ, ∀ M, toSymOrd κ n k rS M * p = p * M

theorem **exists_conj** : ∃ t : G0 X.E Pr.vGs R.r, ∀ a : (rPsymF (R

theorem **nontrivial_symF** (h : Nontrivial (MtP W Ch sel mu (rPsym (R

theorem **packI** (a : I.ι) : ∃ (n : ℕ) (e : MtCx I a ≃ₗ[ℂ] Eu n) (L : Rows.OLat B.O (B.embLL.comp B.OtoLL) (Eu n)), (∀ x, Formalism.HermitianLatticesTheory.normAt (MtHL I hX a).b (MtHL I hX a).X (InfinitePlace.mk B.embLL) x = ‖e (toCx I hX a x)‖) ∧ (∀ y, y ∈ L.carrier ↔ ∃ x ∈ (MtHL I hX a).L, e (toCx I hX a x) = y) ∧ ∀ y ∈ L.gen, y ∈ L.carrier

theorem **exists_same** (h : I.Ok) (u : B.Word) (a : I.ι) : ∃ g : MtCx I a ≃ₗ[ℂ] (I.perIn u a).Mt B.fields.emb B.embLL, ∀ F p, (g F).1 () p = F.1 (toRow I.sel (I.tup a) p.1.1)

theorem **exists_ka** (B : Base) : ∃ (kb : ∀ j, Tau B.R → Fin (B.R.sigW j).1 → ℤ) (ks : ∀ j, Tau B.R → Fin (B.R.sigW j).2 → ℤ), ∀ (C : Cols B) (W : RowsRest B C), W.kaBig = kb → W.kaSmall = ks → ∀ V S N G, DV.KapDef (assemble B C W V S N G)

theorem **md_zero** (R : Ranks) (e : ℕ) : md R.m e 0 = 0

theorem **sub_md** (m e j : ℕ) (h : j ≠ m) : Subsingleton (Eu (md m e j))

theorem **rows_standIn** (hk : ∀ V S N G, DV.KapDef (assemble B C (what₂ hf d L e0 eT kb ks) V S N G)) : StandIn (what₂ hf d L e0 eT kb ks) where

theorem **nonempty_rowsDef** (h : I.Ok) : Nonempty (RowsDef I l hX)

theorem **lat_iff** {K : Type} [Field K] [NumberField K] [StarRing K] (H : HLat K) (σ : K →+* ℂ) {N : Type} [AddCommGroup N] [Module ℂ N] (j : H.V →ₛₗ[σ] N) (hj : Function.Injective j) {O : Type} [CommRing O] (ι : O →+* K) (hι : ∀ y, IsIntegral ℤ (ι y)) (L : Rows.OLat O (σ.comp ι) N) (h1 : ∀ y, y ∈ L.carrier ↔ ∃ x ∈ H.L, j x = y) (h2 : ∀ y ∈ L.gen, y ∈ L.carrier) (x : H.V) : x ∈ H.L ↔ ∃ c : N → 𝒪 K, j x = ∑ v ∈ L.gen, σ ((c v : 𝒪 K) : K) • v

theorem **cx_facts** (a : I.ι) : ∃ j : (MtHL I hX a).V →ₛₗ[B.embLL] MtCx I a, (∀ x, j x = toCx I hX a x) ∧ Function.Injective j ∧ (MtHL I hX a).Good ∧ LinearIndependent ℂ (fun i => j ((MtHL I hX a).b i)) ∧ Submodule.span ℂ (Set.range fun i => j ((MtHL I hX a).b i)) = ⊤

theorem **exists_ambTrue** (Z : TC.Later B (I.cols l)) (e : ∀ a : (I.cols l).Tup, MtCx I a ≃ₗ[ℂ] Z.W.Amb a) (h : DefTrue I l hX Z.W e) : ∃ A, AmbTrue I l hX Z e A

theorem **exists_later** (hB : StagesInhab.HasLetters B.toPre) (Rw : RowsDef I l hX) : ∃ Z : TC.Later B (I.cols l), Z.W = Rw.W

theorem **nontrivial_reference_forms** (h : I.Ok) : Nontrivial (MtCx I I.mu0)

theorem **buDef_iff** (Rw : RowsDef I l hX) : Rows.BuDef Rw.W.what₂ ↔ ∀ u φ F, Rw.W.perCol u (I.cols l).mu0 φ F = 0

theorem **exists_all** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (hX : B.fields.Ok) (Z : TC.Later B (I.cols l)) (e : ∀ a : (I.cols l).Tup, MtCx I a ≃ₗ[ℂ] Z.W.Amb a) (A : AmbForm (I.rec7 l Z) (RowSide.sharp (I.rec7 l Z))), I.Ok ∧ TC.Pins B X ∧ I.muStar ≠ I.mu0 ∧ DefTrue I l hX Z.W e ∧ StandIn Z.W ∧ AmbTrue I l hX Z e A ∧ Nontrivial (Z.W.Amb (I.cols l).mu0)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonzeroIndexPlacesAdelicDataResizedEq

theorem **ExtModel.same_carriers** (h : Enl B B') : HEq (w'.st Z').toD7.host (w.st Z).toD7.host ∧ HEq (w'.st Z').toD7.sp (w.st Z).toD7.sp ∧ HEq (tauE (w'.st Z').toD7) (tauE (w.st Z).toD7) ∧ HEq (fun V => GA (w'.st Z').toD7 V) (fun V => GA (w.st Z).toD7 V) ∧ HEq (PolF (w'.st Z').toD7) (PolF (w.st Z).toD7) ∧ HEq (@Formalism.IndefR.Datum.WO (w'.st Z').toD7) (@Formalism.IndefR.Datum.WO (w.st Z).toD7) ∧ HEq (@Bridge.Gf (w'.st Z').toD7) (@Bridge.Gf (w.st Z).toD7) ∧ HEq (w'.st Z').toD7.hc (w.st Z).toD7.hc ∧ Formalism.IndefR.Datum.InfFam (w'.st Z').toD7 = Formalism.IndefR.Datum.InfFam (w.st Z).toD7 ∧ HEq (@lieAt (w'.st Z').toD7) (@lieAt (w.st Z).toD7) ∧ HEq (Formalism.L2.Pj.hostOfRecord (w'.st Z').toD7) (Formalism.L2.Pj.hostOfRecord (w.st Z).toD7) ∧ HEq (Formalism.L2.Pj.datumOfRecord (w'.st Z').toD7) (Formalism.L2.Pj.datumOfRecord (w.st Z).toD7)

theorem **ExtModel.QE_one** (sel) : Formalism.L2.Col.QE (X

theorem **ExtModel.rowSG_ref** (U : Set w.I.ι) : Formalism.L2.Col.rowSG (B.primes w.px) w.I.sel (mu0F (X

theorem **ExtModel.deco_eq** (h : Enl B B') (hp : HEq w'.px w.px) (U : Set w.I.ι) (U' : Set w'.I.ι) (j : ℕ) : (Formalism.L2.Col.mtKGSD (Pr

theorem **ExtModel.par_eq** (h : Enl B B') : HEq (fun W : HSp (w'.st Z').toD7 => parAt W (w'.st Z').toD7.what₂.embLL) (fun W : HSp (w.st Z).toD7 => parAt W (w.st Z).toD7.what₂.embLL)

theorem **ExtModel.not_ext** {z : Dat (w.st Z) w.I hX T Wt} {z' : Dat (w.st Z) w.I hX T W₁} (h : z'.R.ψ ≠ z.R.ψ ∨ z'.R.nW ≠ z.R.nW) : ¬ Ext z z'

theorem **ExtModel.nonzeroIndexPlaces_adelicDataResized_eq** {z : Dat (w.st Z) w.I hX T Wt} {z' : Dat (w.st Z) w.I hX T W₁} (e : Ext z z') : z.W1 (fun _ _ => 0) = z'.W1 fun _ _ => 0

theorem **ExtModel.below_exists** (D D' : D7) (h : Set.range D.what₂.embLL ⊆ Set.range D'.what₂.embLL) (v' : D'.what₂.places.Pl) : ∃! v, Below D D' v v'

theorem **ExtModel.mmap_eq** (h : Enl B B') (i : Fin (w.st Z).toD7.R.m) (i' : Fin (w'.st Z').toD7.R.m) (hi : (i : ℕ) = i') {S : (w.st Z).toD7.what₂.Tau → SPar} {S' : (w'.st Z').toD7.what₂.Tau → SPar} (M : Formalism.IndefR.Mult.DSModel (Wc (w.st Z).toD7 i) S) (M' : Formalism.IndefR.Mult.DSModel (Wc (w'.st Z').toD7 i') S') (a : HEq M'.H M.H) (b : HEq M'.nacg M.nacg) (c : HEq M'.ips M.ips) (d : HEq M'.σ M.σ) (l : HEq M'.lam M.lam) (e : HEq M'.ε M.ε) (g : HEq M'.fr M.fr) : Formalism.IndefR.Mult.MMap M' = Formalism.IndefR.Mult.MMap M ∧ LowT M' = LowT M ∧ M'.Vec = M.Vec ∧ HEq (inferInstance : Module ℂ (Formalism.IndefR.Mult.MMap M')) (inferInstance : Module ℂ (Formalism.IndefR.Mult.MMap M)) ∧ HEq (inferInstance : Module ℂ (LowT M')) (inferInstance : Module ℂ (LowT M))

theorem **ExtModel.sign_at_base_place_eq_embedding** (z : Dat (w.st Z) w.I hX T Wt) : sg0 (w.st Z).toD7 (famR (w.st Z) w.I hX (usedOf w.I T Wt) z.Xg z.tr).A = (w.st Z).toD7.what₂.embLL

theorem **ExtModel.more_carriers** (h : Enl B B') (i : Fin (w.st Z).toD7.R.m) (i' : Fin (w'.st Z').toD7.R.m) (hi : (i : ℕ) = i') (k : WeightFamilies.Wt (w.st Z).toD7 ((i : ℕ) + 1)) : (∃ k' : WeightFamilies.Wt (w'.st Z').toD7 ((i' : ℕ) + 1), HEq k'.aP k.aP ∧ HEq k'.aM k.aM ∧ k'.n = k.n) ∧ Measure (GA (w'.st Z').toD7 (Vc (w'.st Z').toD7 i')) = Measure (GA (w.st Z).toD7 (Vc (w.st Z).toD7 i)) ∧ Measure (GA (w'.st Z').toD7 (Wc (w'.st Z').toD7 i')) = Measure (GA (w.st Z).toD7 (Wc (w.st Z).toD7 i))

theorem **ExtModel.pol_eq** (h : Enl B B') (n : ℕ) : Formalism.L2.DIV.Pol ℂ B'.fields.Tau n = Formalism.L2.DIV.Pol ℂ B.fields.Tau n ∧ HEq B'.wts.lamV B.wts.lamV

theorem **ExtModel.meas_eq** {z : Dat (w.st Z) w.I hX T Wt} {z' : Dat (w.st Z) w.I hX T W₁} (e : Ext z z') (i : Fin (w.st Z).toD7.R.m) : z'.R.νW i = z.R.νW i ∧ z'.R.fdW i = z.R.fdW i ∧ z'.R.ΛY = z.R.ΛY

theorem **ExtModel.premise_zero** (h : Enl B B') (U : Set w.I.ι) (U' : Set w'.I.ι) (j : ℕ) (d d') : HEq (MvPolynomial.map (B'.embLL : B'.fields.K →+* ℂ) ((0 : (Formalism.L2.Col.mtKGSD (Pr

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonzeroIndexPlacesTransferAlongSameFiniteGroup

theorem **LevOf.transfer** (b : Late B C) {U : Set I.ι} {L : Links.LevCh a.D (TupU I U)} (h : LevOf a.st I U L) : LevOf b.st I U (levCh b L)

theorem **GoodSet.transfer** (s : SameGrp a b) {eR : Gc B px ≃* G0g a.D.nd} {Xg : Finset (Gc B px → B.LL)} (h : GoodSet a.st (px

theorem **GrpAt.pl_transfer** (s : SameGrp a b) (h : GrpAt a.st I) : ∃ plg : (Σ k : Fin 2, b.D.nd.PlG k) ≃ (B.primes px).vGs, ∀ kv, ((plg kv).1 : Pl b.D.host.E) = b.D.naut.plGw kv.1 kv.2 ∧ I.C.P.Div kv.1 (plg kv).1

theorem **transfer_Xg** : (z.transfer s).Xg = z.Xg

theorem **transfer_tr** : (z.transfer s).tr = z.tr

theorem **transfer_field** : ((z.transfer s).core.Q.c.F : Type) = z.core.Q.c.F

theorem **transfer_mu** : (z.transfer s).R.mu = (z.R.mu)

theorem **transfer_twPl** : (z.transfer s).R.twPl = (z.R.twPl)

theorem **transfer_kap** : (z.transfer s).R.kap = z.R.kap

theorem **transfer_liftScalars** : (z.transfer s).R.lam3 = (z.R.lam3)

theorem **transfer_errorTerms** : (z.transfer s).R.err0 = (z.R.err0)

theorem **transfer_phs** : (z.transfer s).R.phs = (z.R.phs)

theorem **transfer_Pg** : (z.transfer s).R.Pg = (z.R.Pg)

theorem **transfer_eG** : (z.transfer s).R.eG = (z.R.eG)

theorem **transfer_cG** : (z.transfer s).cG = (by with_unfolding_all exact ((z.cG.trans s.iso) :))

theorem **transfer_nW** : (z.transfer s).R.nW = (z.R.nW)

theorem **transfer_FW** : (z.transfer s).R.FW = (z.R.FW)

theorem **transfer_HX** : (z.transfer s).R.HX = (z.R.HX)

theorem **transfer_HV** : (z.transfer s).R.HV = (z.R.HV)

theorem **transfer_HE** : (z.transfer s).R.HE = (z.R.HE)

theorem **transfer_HW** : (z.transfer s).R.HW = (z.R.HW)

theorem **transfer_Th** : (z.transfer s).R.Th = (z.R.Th)

theorem **transfer_thG** : (z.transfer s).R.thG = (z.R.thG)

theorem **transfer_zs** : zsOf z.R z.F = (by with_unfolding_all exact ((zsOf (z.transfer s).R (z.transfer s).F) :))

theorem **transfer_EEat** : z.R.toRealDat.EEat = ((transfer s z).R.EEat)

theorem **transfer_Wall** : z.R.toRealDat.Wall = ((transfer s z).R.Wall)

theorem **transfer_Wat** : z.R.toRealDat.Wat = ((transfer s z).R.Wat)

theorem **transfer_CC** : z.R.toRealDat.CC = ((transfer s z).R.CC)

theorem **eigenspace_eq_of_transfer** : z.R.EEp3 = ((transfer s z).R.EEp3)

theorem **transfer_formParts** : z.R.Wp3 = ((transfer s z).R.Wp3)

theorem **transfer_toCx3** : z.R.toCx3 z.F = ((transfer s z).R.toCx3 (transfer s z).F)

theorem **transfer_s0Of** : HEq (s0Of z.R z.F) (s0Of (z.transfer s).R (z.transfer s).F)

theorem **transfer_s1Of** : HEq (s1Of z.R z.F) (s1Of (z.transfer s).R (z.transfer s).F)

theorem **nonzeroIndexPlaces_transferAlongSameFiniteGroup** (F0 : HeightOneSpectrum (𝒪 B.LL) → Fin B.R.m → ℝ) : z.W1 F0 = (z.transfer s).W1 F0

theorem **holds_transfer** (h : Holds z.R z.F) : Holds (z.transfer s).R (z.transfer s).F

theorem **exAt_transfer** (h : z.R.toRealDat.ExAt) : (z.transfer s).R.toRealDat.ExAt

theorem **inGeo_transfer** (h : InGeo z.geo) : InGeo (z.transfer s).geo

theorem **errEq_transfer** (h : ErrEq z.R z.F) : ErrEq (z.transfer s).R (z.transfer s).F

theorem **nonempty_dat_iff** (s : SameGrp a b) : Nonempty (Dat a.st I hX T Wt) ↔ Nonempty (Dat b.st I hX T Wt)

theorem **tr_z_eq_transfer** : (r.tr c).z = r.z.transfer c.same

theorem **tr_tup** : (r.tr c).Y.tup = (r.Y.tup)

theorem **tr_jA** : (r.tr c).Y.jA = (r.Y.jA)

theorem **transfer_baseForms** : (r.tr c).Y.jA0 = (r.Y.jA0)

theorem **tr_eMt** : (r.tr c).Y.eMt = (r.Y.eMt)

theorem **tr_eE** : (r.tr c).Y.eE = (r.Y.eE)

theorem **tr_jW** : (r.tr c).Z.jW = (r.Z.jW)

theorem **tr_eRR** : HEq r.Z.eRR (r.tr c).Z.eRR

theorem **transfer_rowLattice** : HEq r.Z.eRR0 (r.tr c).Z.eRR0

theorem **Refill.ofDiv_D** (L : ∀ u, Submodule (𝒪 B.LL) (a.V.div.Nu u).V) (h : B.Word → ℝ) : (Refill.ofDiv a L h).to.D = GlobalTheory.EX.OB.setDiv a.D L h

theorem **Refill.refill_divisionSide** (L : ∀ u, Submodule (𝒪 B.LL) (a.V.div.Nu u).V) (h : B.Word → ℝ) (p0 : BaseLevelRowSide a.D) : baseLevelRowSideTransport (Refill.ofDiv a L h) p0 = (@BaseLevelRowSide.mk (EX.OB.setDiv (@Late.D B C a) L h) (@EX.OB.sideTr (@Late.D B C a) L h (@Rows.Data.EE0 (EX.OB.setDiv (@Late.D B C a) L h).R (EX.OB.setDiv (@Late.D B C a) L h).what₂) (@Rows.Data.instEE0 (EX.OB.setDiv (@Late.D B C a) L h).R (EX.OB.setDiv (@Late.D B C a) L h).what₂) (@Rows.Data.instEE0i (EX.OB.setDiv (@Late.D B C a) L h).R (EX.OB.setDiv (@Late.D B C a) L h).what₂) (@Rows.Data.instEE0f (EX.OB.setDiv (@Late.D B C a) L h).R (EX.OB.setDiv (@Late.D B C a) L h).what₂) (@BaseLevelRowSide.s (@Late.D B C a) p0)))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonzeroPlacesOverPrimes

theorem **stepW_built** {𝒮 : Slots} (h2 : StepW Start (Has2 𝒮)) (h3 : StepW (pinAnd (Has2 𝒮)) (Has3 𝒮)) (h4 : StepW (pinAnd (Has3 𝒮)) (Has4 𝒮)) (h5 : StepW (pinAnd (Has4 𝒮)) (HasC (BuiltC 𝒮))) : StepW Start (HasC (BuiltC 𝒮))

theorem **supp_of_untw** (w : ColW B f T) (h : T = TwDat.untw B T.pl) : Supp K B f T w

theorem **coh_of_untw** (h : T = TwDat.untw B T.pl) : Coh B T

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonzeroPlacesOverPrimesLifts

theorem **HoldsL.of_holds** {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 I0 T : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] {R : RealDat3 D A3 Ks} {F : Face3 D A3} (h : Holds R F) : HoldsL R F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NonzeroPlacesOverPrimesOfLifts

theorem **holdsLU_of_holdsU** (h : UnitaryCaseCondition) : LowerUnitaryCaseCondition

theorem **holdsW_of_holdsU** (h : UnitaryCaseCondition) : WeightCaseCondition

theorem **nonzeroPlacesOverPrimes_of_lifts** (h : NonzeroPlacesOverPrimesLifts) : NonzeroPlacesOverPrimes

theorem **bkLaw_of_bkDef** (D : D7) (h : WeightDegreeSizes.BkDef D) : BkLaw D

theorem **sideData_same_stages** {Z : w.Lt} (p0 : BaseLevelRowSide (w.D Z)) : p0To p0 Z.S Z.G = p0

theorem **coreJ_self** {C : Cols B} (a : Late B C) (c : Formalism.L2.Pj.CoreJ a.D) : coreJ (a

theorem **realData_heq_self** {C : Cols B} (a : Late B C) {G : Type} [Group G] [Fintype G] {Pl0 I0 T : Type} {Ks : Type} [Field Ks] [NumberField Ks] [iK : Algebra Ks a.D.what₂.LL] {A : FamIn3 a.D.what₂.LL G Pl0 I0 (List ℕ) T a.D.R.m} (R : RealDat3 a.D A Ks) : HEq R (real3 (a

theorem **rowsSame_refl** {Z : w.Lt} {p0 : BaseLevelRowSide (w.D Z)} (t : RowsT K w Z Wt p0) : RowsSame t t where

theorem **ty_chain** (j : ℕ) : (E.T →ₗ[ℂ] (chB b F y').Amb j) = (E.T →ₗ[ℂ] (y.C F).Amb j)

theorem **ty_top** : (E.T →ₗ[ℂ] (chB b F y').AmbX) = (E.T →ₗ[ℂ] (y.C F).AmbX)

theorem **ty_lift** (E : ℕ → Hs) : (∀ i : Fin b.D.R.m, (E i).T →ₗ[ℂ] (GA b.D ((lfB b F y').V i) → (lfB b F y').TauV i)) = (∀ i : Fin a.D.R.m, (E i).T →ₗ[ℂ] (GA a.D ((y.L F).V i) → (y.L F).TauV i))

theorem **ty_model** (E : ℕ → Hs) : (letI

theorem **frames_copy** : HEq (rowPsi b y.PY).ψ y.PY.ψ ∧ HEq (fun i => (rowPsi b (y.PW i)).ψ) fun i => (y.PW i).ψ

theorem **copyR_heq** {Z : w.Lt} (Z' : w.Lt) {hX : B.fields.Ok} (z : Dat (w.st Z) w.I hX T Wt) : HEq (copyR Z' z) (real3 (a

theorem **exists_bkLaw** (D : D7) : ∃ b, BkLaw (GlobalTheory.Survey.IrrationalRotationBlocks.withBk D b)

theorem **exists_not_bkLaw** (D : D7) (h : WeightDegreeSizes.RationalMapValues D) (u : D.what₂.Word) [Nontrivial (D.div.Nu u).V] [Nontrivial D.div.RR.V] : ∃ b, ¬ BkLaw (GlobalTheory.Survey.IrrationalRotationBlocks.withBk D b)

theorem **St.kapDef** {P : Pre2 K w} (x : St P) : DV.KapDef (w.D x.o.Z)

theorem **St.gIs** {P : Pre2 K w} (x : St P) : GIs2 K w x.o.Z.G

theorem **Pre2.holdsL** (P : Pre2 K w) : HoldsL P.z.R P.z.F

theorem **hasPk_of_hasBare** (h : HasBare K B f T w) : HasPk K B f T w

theorem **std_cpR** (z : Dat a.st I hX T Wt) (h : z.Std) : @BrgStd b.st px I hX (usedOf I T Wt) z.Xg z.tr (coreJ b z.core).Q.c.F _ _ ((by exact @alg B C a b (ɪ(@ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff.F (@Formalism.HC.HostRaw.E (@Late.st B C a).toD7.R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.st B C a).toD7.R.toFrame (@Host.toHostAx (@Late.st B C a).toD7.R (@Late.st B C a).toD7.host))) (@Formalism.HC.HostRaw.instField (@Late.st B C a).toD7.R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.st B C a).toD7.R.toFrame (@Host.toHostAx (@Late.st B C a).toD7.R (@Late.st B C a).toD7.host))) (@Formalism.HC.HostRaw.instNF (@Late.st B C a).toD7.R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.st B C a).toD7.R.toFrame (@Host.toHostAx (@Late.st B C a).toD7.R (@Late.st B C a).toD7.host))) (Formalism.L2.Pj.instCMHost (@Late.st B C a).toD7) (@Formalism.L2.Pj.Par.H (@Late.st B C a).toD7 (@Forma

theorem **std_of_copy** (z₀ : Dat a.st I hX T Wt) (z : Dat b.st I hX T Wt) (h1 : z.Xg = z₀.Xg) (h2 : z.tr = z₀.tr) (h3 : HEq z.core (coreJ b z₀.core)) (h4 : HEq z.R (real3 b (iK

theorem **exists_std_pt** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (hG : V.gram = 1) (h : Bridge.Gf V) : ∃ x : Formalism.L2.Col.UAd D.sp.cm n, MatEq (matIn V 1 h) (x : GL (Fin n) _)

theorem **St.std** {P : Pre2 K w} (x : St P) : x.m.z.Std

theorem **Pre2.vStd** (P : Pre2 K w) (i) : FWT.VStd (w.st P.Z₀).toD7 i

theorem **gram_X** (Z : w.Lt) : (w.st Z).toD7.sp.X'.gram = 1

theorem **Pre2.std_row** (P : Pre2 K w) : (w.st P.Z₀).toD7.sp.X'.rank = B.R.N B.R.m − 1 ∧ (w.st P.Z₀).toD7.sp.X'.gram = 1 ∧ P.z.R.βX = 1

theorem **Pre2.std_lift** (P : Pre2 K w) (i) : (Vc (w.st P.Z₀).toD7 i).rank = B.R.nV ∧ (Vc (w.st P.Z₀).toD7 i).gram = 1 ∧ P.z.R.βV i = 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotBaseCondFalse

theorem **evG_all** {p : Ctx → Prop} (h : ∀ K, p K) : EvG p

theorem **fromCore_iff** (Q : V4.Slot) (D : D7) (Y : Lay D) : V4.FromCore Q D Y ↔ ∃ K, FromCoreAt Q K D Y

theorem **exTrueG_of** {KP : BaseP} {Q : V4.Slot} (h : V4.ExTrue KP Q) : ExTrueG KP Q

theorem **exportG_of** {Q : V4.Slot} (h : V4.Export Q) : ExportG Q

theorem **retwistG_of** {Q Q' : V4.Slot} (h : V4.Retwist Q Q') : RetwistG Q Q'

theorem **baseCond_of** {KP : BaseP} (h : V4.KsCond KP) {PP : ∀ P : Pre, Fr P → Prop} (hP : EvG fun K => ∃ (P : Pre) (f : Fr P), K.Over P ∧ PP P f) : BaseCond fun B f => KP B f ∧ PP B.toPre f

theorem **lg_forced** (K : Ctx) : large₂ K fun B _ T => Forced K B T

theorem **lg_all** (K : Ctx) {p : ∀ B : Base, Fr B.toPre → TwDat B → Prop} (h : ∀ B f T, K.Over B.toPre → p B f T) : large₂ K p

theorem **Step.of_evG** (h : EvG fun K => ∀ B f T, K.Over B.toPre → KP B f → Forced K B T → p K B f T → q K B f T) : Step KP p q

theorem **Step.of_all** (h : ∀ K B f T, K.Over B.toPre → KP B f → Forced K B T → p K B f T → q K B f T) : Step KP p q

theorem **Step.trans** (h : Step KP p q) (h' : Step KP q r) : Step KP p r

theorem **Step.and** (h : Step KP p q) (h' : Step KP p r) : Step KP p fun K B f T => q K B f T ∧ r K B f T

theorem **Step.of_evLgG** (h : EvLgG q) : Step KP p q

theorem **exTrueG_iff_step** {Q : V4.Slot} : ExTrueG KP Q ↔ Step KP (fun _ _ _ _ => True) (HasCore Q)

theorem **firstG_of** {Q : V4.Slot} (h : ExTrueG KP Q) : FirstG KP Q

theorem **exportF_of** {Q : V4.Slot} (h : ExportG Q) : ExportF KP Q

theorem **retwistA_of** {Q Q' : V4.Slot} (h : RetwistG Q Q') : RetwistA KP (fun _ _ _ _ => True) Q Q'

theorem **lg_of_lgOn** {Δ : Pre → Type} {Sm : ∀ B : Pre, Fr B → Δ B → IntermediateField ℚ ℂ → Prop} {K : Ctx} {p : ∀ B : Base, Fr B.toPre → Δ B.toPre → TwDat B → Prop} (h : LgOn Sm K p) : large₂ K fun B f T => (∃ δ, Sm B.toPre f δ B.KsC) → ∃ δ, Sm B.toPre f δ B.KsC ∧ p B f δ T

theorem **ExTrueG.mono** {KP : BaseP} {Q Q' : V4.Slot} (hE : ExTrueG KP Q) (h : ∀ K X f T c, Q K X f T c → Q' K X f T c) : ExTrueG KP Q'

theorem **ExportG.anti** {Q Q' : V4.Slot} (hE : ExportG Q) (h : ∀ K X f T c, Q' K X f T c → Q K X f T c) : ExportG Q'

theorem **RetwistG.strengthen** {Q Q₁ Q' : V4.Slot} (hR : RetwistG Q Q') (h : ∀ K X f T c, Q₁ K X f T c → Q K X f T c) : RetwistG Q₁ fun K X f T c => Q' K X f T c ∧ Q₁ K X f T c

theorem **LargeBaseConditionG.passes** {KP : BaseP} {Q Q' R : V4.Slot} (h : LargeBaseConditionG KP Q R) (h1 : ExTrueG KP Q) (h2 : ExportG Q) (h3 : RetwistG Q Q') : ExTrueG KP (fun K X f T c => Q K X f T c ∧ R K X f T c) ∧ ExportG (fun K X f T c => Q K X f T c ∧ R K X f T c) ∧ RetwistG (fun K X f T c => Q K X f T c ∧ R K X f T c) fun K X f T c => Q' K X f T c ∧ R K X f T c

theorem **baseCond_true** (h2b : StandardLattice.SiegelMassRatioCM) : BaseCond fun _ _ => True

theorem **not_baseCond_false** : ¬ BaseCond fun _ _ => False

theorem **not_exTrueG_false** {KP : BaseP} (hK : BaseCond KP) : ¬ ExTrueG KP fun _ _ _ _ _ => False

theorem **not_firstG_false** {KP : BaseP} (hK : BaseCond KP) : ¬ FirstG KP fun _ _ _ _ _ => False

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotConditionOfFalse

theorem **exists_preX** {K : Ctx} {P : Pre} (hO : K.Over P) (X₀ : PreX P) (Q : PiBar P.host) : ∃ X : PreX P, InEx.Oriented X Q ∧ ∀ u, P.cOf u ∗ (X.piT u : P.host.E) = (X.piT P.u₀ : P.host.E)

theorem **step_colW** (h2b : StandardLattice.SiegelMassRatioCM) : Step KPx (fun _ _ _ _ => True) fun _ B f T => Nonempty (ColW B f T)

theorem **prePin_coordPre** (P : PreCh) (L : LevCh P) : PrePin (GlobalTheory.StdFrameObj.coordPre P L)

theorem **ksCond** (hR : GlobalTheory.NT.RayField) (hP : GlobalTheory.Published.PrincipalIdealSt) (hA : GlobalTheory.Published.CMCurveGoodOutsidePrimeSt) : V4.KsCond fun B f => ∃ δ : Fld.FieldDat B.toPre, B.KsExact δ f

theorem **baseCond_KPx** (h2b : StandardLattice.SiegelMassRatioCM) (hR : GlobalTheory.NT.RayField) (hP : GlobalTheory.Published.PrincipalIdealSt) (hA : GlobalTheory.Published.CMCurveGoodOutsidePrimeSt) : BaseCond KPx

theorem **not_step_false** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) {KP : BaseP} (hK : BaseCond KP) : ¬ Step KP (fun _ _ _ _ => True) fun _ _ _ _ => False

theorem **exists_colW** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) (hR : GlobalTheory.NT.RayField) (hP : GlobalTheory.Published.PrincipalIdealSt) (hA : GlobalTheory.Published.CMCurveGoodOutsidePrimeSt) : ∃ (K : Ctx) (B : Base) (f : Fr B.toPre) (T : TwDat B), K.Over B.toPre ∧ KPx B f ∧ Forced K B T ∧ Nonempty (ColW B f T)

theorem **not_stepW_false** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) (hR : GlobalTheory.NT.RayField) (hP : GlobalTheory.Published.PrincipalIdealSt) (hA : GlobalTheory.Published.CMCurveGoodOutsidePrimeSt) : ¬ StepW (fun _ _ _ _ _ => True) fun _ _ _ _ _ => False

theorem **ColW.kpx** {B : Base} {f : Fr B.toPre} {T : TwDat B} (w : ColW B f T) : KPx B f

theorem **not_corePropertyOf_false** (K : Ctx) (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (c : V4.Core K X f T) : ¬ positionOf (fun _ _ _ _ _ _ _ => False) K X f T c

theorem **positionOf_cols** {P : CP} {K : Ctx} {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {c : V4.Core K X f T} (h : positionOf P K X f T c) : ∃ w : ColW X.B f T, X.C = w.C

theorem **exists_later** {K : Ctx} {B : Base} {f : Fr B.toPre} {T : TwDat B} (hO : K.Over B.toPre) (w : ColW B f T) (G : GJStage B w.C) : ∃ Z : w.Lt, Z.G = G

theorem **GIs.exists_colT** {K : Ctx} {B : Base} {f : Fr B.toPre} {T : TwDat B} (hO : K.Over B.toPre) {w : ColW B f T} {G : GJStage B w.C} (h : GIs K w G) : ∃ Z : w.Lt, Z.G = G ∧ Formalism.L2.Theta.VolTieG (w.D Z) ∧ Nonempty (ColT w Z)

theorem **GIs.not_double** {K : Ctx} {B : Base} {f : Fr B.toPre} {T : TwDat B} (hO : K.Over B.toPre) {w : ColW B f T} {G : GJStage B w.C} (h : GIs K w G) (X : @HSpace B.host.E _ _ B.host.cmStar) (hX : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotGeoClFalse

theorem **geoCl_of** {Γ : GeoP} {D : D7} {J : Formalism.L2.Pj.Core D} {X : DivX D} (h : Γ D J X) : geoCl Γ D J X

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotIsBettiZero

theorem **expo_detSym** (τ : GlobalTheory.AllWt.Psi H) (a : ℕ → Bool) (R : Option ℝ) (z : Letter Dw) : expo Dw (detSym Dw a R τ.1) z = if z.1 = τ.1 then 1 else if z.1 = conjEmb τ.1 then -1 else 0

theorem **expo_detSym_eq** (τ : GlobalTheory.AllWt.Psi H) (a : ℕ → Bool) (R : Option ℝ) : expo Dw (detSym Dw a R τ.1) = fun z => -GlobalTheory.AllWt.exOf Dw (flatWt Dw τ) 0 (some z)

theorem **carriedC_un** (k : s.Comp) (x : ((s.U κ.σ₀).X k).dom) : CarriedC (Φ.C k) ((s.U κ.σ₀).un k x)

theorem **bettiCo_ne_zero** (k : s.Comp) (τ : D.what₂.Tau) (x : ((s.U κ.σ₀).X k).dom) : bettiCo s L k τ x.1 ≠ 0

theorem **detEx_eq** (S : Sig A N j) (L : CmList A) (τ : D.what₂.Tau) : detEx S τ = fun z => -GlobalTheory.AllWt.exOf S.Dw (flatWt S.Dw (L.ψ τ)) 0 (some z)

theorem **IsBetti.coordAtC** (h : IsBetti s Φ L k τ σ b) (x : ((s.U κ.σ₀).X k).dom) : CoordAtC (Φ.C k) ((s.U κ.σ₀).un k x) (detEx S τ) σ (b * bettiCo s L k τ x.1)

theorem **IsBetti.unique** (h : IsBetti s Φ L k τ σ b) (h' : IsBetti s Φ L k τ σ b') : b = b'

theorem **IsBetti.const_mul** (h : IsBetti s Φ L k τ σ b) {a : κ.c.F} (ha : a ≠ 0) : IsBetti s Φ L k τ (constF κ.e (Φ.C k).Y a * σ) (Units.mk0 (κ.σ₀ a) ((map_ne_zero _).2 ha) * b)

theorem **isBetti_flBetti** (h : ∃ b, IsBetti s Φ L k τ (flatSec s Φ k τ) b) : IsBetti s Φ L k τ (flatSec s Φ k τ) (flBetti s Φ L k τ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotLamAreShift

theorem **setLam_off** {D : D7} (X : DivX D) (l) (j σ) {e} (h : e ≠ D.what₂.embLL) : (setLam X l).logLam j σ e = l j σ e

theorem **isGeoOf_set** {D : D7} {Q : GlobalTheory.ShFam.KitClass D} {J c} {X : DivX D} (h : GlobalTheory.ShFam.IsGeoOf Q J c X) (l) (t : GlobalTheory.ShFam.Pad.SurfSt) : GlobalTheory.ShFam.IsGeoOf Q J c (t.put (setLam X l))

theorem **geoOf_set** {Q : ∀ D : D7, GlobalTheory.ShFam.KitClass D} {D : D7} {J} {c : ℕ} (hc : 0 < c) {X : DivX D} (h : GlobalTheory.ShFam.IsGeoOf (Q D) J c X) (l) : geoOf Q D J (GlobalTheory.ShFam.Pad.surfSt.put (setLam X l))

theorem **geoOf_surf** {Q : ∀ D : D7, GlobalTheory.ShFam.KitClass D} {D : D7} {J} {X : DivX D} (h : geoOf Q D J X) : Nonempty X.Surf ∧ Nonempty X.SurfFp

theorem **lamAre_set** (Z : R3.ReadL R F X0 Y) (lab : D.what₂.FT → A3.fam3.base3.Ix) (hl : ∀ σ, (∃ c, OverIx Z σ c) → OverIx Z σ (lab σ)) (X : DivX D) (t : GlobalTheory.ShFam.Pad.SurfSt) : LamAre Z (t.put (setLam X (lamOf Z lab)))

theorem **exists_lab** (Z : R3.ReadL R F X0 Y) (hne : Nonempty D.what₂.FT → Nonempty A3.fam3.base3.Ix) : ∃ lab : D.what₂.FT → A3.fam3.base3.Ix, ∀ σ, (∃ c, OverIx Z σ c) → OverIx Z σ (lab σ)

theorem **nonempty_ix** {Z : R3.ReadL R F X0 Y} {X : DivX D} (h : LamAre Z X) : Nonempty D.what₂.FT → Nonempty A3.fam3.base3.Ix

theorem **overIx_of_via** {Z : R3.ReadL R F X0 Y} {cOf} (h : LamVia Z cOf) (σ) : OverIx Z σ (cOf σ)

theorem **eq_of_overIx** {Z : R3.ReadL R F X0 Y} {cOf} (h : LamVia Z cOf) {i : Fin D.R.m} (hi : iSupIndep (D.what₂.isot ((i : ℕ) + 1))) {σ c} (h0 : D.what₂.isot ((i : ℕ) + 1) σ ≠ ⊥) (hc : OverIx Z σ c) : c = cOf σ

theorem **logLam_eq** {Z : R3.ReadL R F X0 Y} {X : DivX D} (hX : LamAre Z X) {cOf} (h : LamVia Z cOf) {i : Fin D.R.m} (hi : iSupIndep (D.what₂.isot ((i : ℕ) + 1))) {σ} (h0 : D.what₂.isot ((i : ℕ) + 1) σ ≠ ⊥) {e} (he : e ≠ D.what₂.embLL) : X.logLam ((i : ℕ) + 1) σ e = Real.log ((R3.s0Of R F).lam i A3.A.μ0 (cOf σ) (InfinitePlace.mk e))

theorem **logLam_all** {Z : R3.ReadL R F X0 Y} {X : DivX D} (hX : LamAre Z X) (h : R3.Lam3 Z) (hi : ∀ i : Fin D.R.m, iSupIndep (D.what₂.isot ((i : ℕ) + 1))) (hv : InfinitePlace.mk D.what₂.embLL = A3.A.v0) : ∃ cOf : D.what₂.FT → A3.fam3.base3.Ix, ∀ (i : Fin D.R.m) σ, D.what₂.isot ((i : ℕ) + 1) σ ≠ ⊥ → ∀ e, X.logLam ((i : ℕ) + 1) σ e = Real.log ((R3.s0Of R F).lam i A3.A.μ0 (cOf σ) (InfinitePlace.mk e))

theorem **logLam_conj** {Z : R3.ReadL R F X0 Y} {X : DivX D} (hX : LamAre Z X) (h : R3.Lam3 Z) (hi : ∀ i : Fin D.R.m, iSupIndep (D.what₂.isot ((i : ℕ) + 1))) (hv : InfinitePlace.mk D.what₂.embLL = A3.A.v0) (i : Fin D.R.m) (σ) (h0 : D.what₂.isot ((i : ℕ) + 1) σ ≠ ⊥) (e) : X.logLam ((i : ℕ) + 1) σ (ComplexEmbedding.conjugate e) = X.logLam ((i : ℕ) + 1) σ e

theorem **not_lamAre_shift** {Z : R3.ReadL R F X0 Y} {X : DivX D} (hX : LamAre Z X) {cOf} (h : LamVia Z cOf) (i : Fin D.R.m) (hi : iSupIndep (D.what₂.isot ((i : ℕ) + 1))) (σ : D.what₂.FT) {e} (he : e ≠ D.what₂.embLL) : ¬ LamAre Z (setLam X fun j σ e => X.logLam j σ e + 1)

theorem **mk_emb** (R : RealDat3 D A3 Ks) : InfinitePlace.mk D.what₂.embLL = A3.A.v0

theorem **mjAre_set** {Z : R3.ReadL R F X0 Y} {X : DivX D} (h : MjAre Z X) (l) (t : GlobalTheory.ShFam.Pad.SurfSt) : MjAre Z (t.put (setLam X l))

theorem **oAt_any** (t : RowsT K w Z Wt p0) (h : R3.Lam3 t.r.Z) (X : DivX (w.D Z)) : ∃ l, Nonempty (OAt anyGeometry t ((GlobalTheory.ShFam.Pad.SurfSt.of X).put (setLam X l)))

theorem **oAt_geoOf** (Q) (t : RowsT K w Z Wt p0) (h : R3.Lam3 t.r.Z) {c : ℕ} (hc : 0 < c) (X : DivX (w.st Z).toD7) (hX : GlobalTheory.ShFam.IsGeoOf (Q _) t.r.z.core.toCore c X) : ∃ l, Nonempty (OAt (geoOf Q) t (GlobalTheory.ShFam.Pad.surfSt.put (setLam X l)))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotLv3MvFixH

theorem **pt_swap** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] (hc : CMData F E) {n : ℕ} [NeZero n] {ST SG Q Q' : Finset (HeightOneSpectrum (𝒪 E))} (hQ : Q' = ∅) {y k k' g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} {q : DecoF E n ST SG Q'} (h : IsPtOf hc (y * k) p) (h' : IsPtOf hc (y * k') p) (hq : IsPtOf hc (g * k) q) : IsPtOf hc (g * k') q

theorem **reads_le** {B : Base} {X : PreX B.toPre} (I : TC B X) (u : B.Word) (i : ℕ) {Lb Lb'} {U U' : Subgroup (GA B (B.R.N i))} {chi : U →* ℂˣ} {chi' : U' →* ℂˣ} {y y' st st' k k'} (hy : y' = y) (h : I.Reads u I.mu0 i (fun _ => ()) (fun _ => Lb) (fun _ => U) (fun _ => chi) y st k) (h' : I.Reads u I.mu0 i (fun _ => ()) (fun _ => Lb') (fun _ => U') (fun _ => chi') y' st' k') (hU : U ≤ UAd B.sp.cm (B.R.N i)) (x : U) : ∃ hx : x.1 ∈ U', chi' ⟨x.1, hx⟩ = chi x

theorem **heq_chi** {G : Type} [Group G] {U U' : Subgroup G} {chi : U →* ℂˣ} {chi' : U' →* ℂˣ} (h : ∀ x : U, ∃ hx : x.1 ∈ U', chi' ⟨x.1, hx⟩ = chi x) (h' : ∀ x : U', ∃ hx : x.1 ∈ U, chi ⟨x.1, hx⟩ = chi' x) : U' = U ∧ HEq chi' chi

theorem **U_le** (Z : w.Lt) (u : B.Word) (F : Forms0 (w.D Z) u) : (show Subgroup (GA B (B.R.N 0)) from F.cls0.lev.U) ≤ UAd B.sp.cm (B.R.N 0)

theorem **car_eq** {Z Z' : w.Lt} (t : ColT w Z) (t' : ColT w Z') (u : B.Word) : car Z' u (t'.C.forms u).to0.cls0.lev = car Z u (t.C.forms u).to0.cls0.lev

theorem **y_eq** {Z Z' : w.Lt} (t : ColT w Z) (t' : ColT w Z') (u : B.Word) : (t'.C.forms u).to0.cls0.y = ((t.C.forms u).to0.cls0.y)

theorem **eY_eq** {Z Z' : w.Lt} (t : ColT w Z) (t' : ColT w Z') (u : B.Word) : (t'.C.forms u).to0.val0.eY = ((t.C.forms u).to0.val0.eY)

theorem **sameStd** : SameStd

theorem **SameCar_symm** (h : SameCar F F') : SameCar F' F

theorem **colDictL_car** (h : SameCar F F') {X : HidaSetting D} {u} (t : Transport X u (F u)) (hK : X.LevelSubgroupCharacter) : colDictL X (trCar h t) hK = colDictL X t hK

theorem **chi_of_heq** {G : Type} [Group G] {U U' : Subgroup G} {chi : U →* ℂˣ} {chi' : U' →* ℂˣ} (e : U' = U) (h : HEq chi' chi) (x : G) (hx : x ∈ U') : chi' ⟨x, hx⟩ = chi ⟨x, e ▸ hx⟩

theorem **lev_car** (h : SameCar F F') {X : HidaSetting D} {u} {t : Transport X u (F u)} {hK : X.LevelSubgroupCharacter} (l : LevelMatch X t hK) : LevelMatch X (trCar h t) hK

theorem **sxMv_of** (H : SatMv) : SxMv

theorem **satMv** : SatMv

theorem **sxMv** : SxMv

theorem **up_eE** {B : Base} {C : Cols B} (a : Late B C) (S' : SStage B C a.W) (N : ∀ u, Submodule (𝒪 B.LL) (a.V.div.Nu u).V) (h : B.Word → ℝ) (p : RowPk a.D) : (p.up a S' N h).G' = p.G' ∧ (p.up a S' N h).Ks = p.Ks ∧ HEq (p.up a S' N h).eE p.eE

theorem **fam_up** (Z : w.Lt) (S' : SStage B w.C Z.W) (N : ∀ u, Submodule (𝒪 B.LL) (Z.V.div.Nu u).V) (h : B.Word → ℝ) (hX : B.fields.Ok) (U Xg tr) : famM (a

theorem **face_up** (Z : w.Lt) (S' : SStage B w.C Z.W) (N : ∀ u, Submodule (𝒪 B.LL) (Z.V.div.Nu u).V) (h : B.Word → ℝ) (hX : B.fields.Ok) (U Xg tr) : face (a

theorem **not_lv3Mv_fixH** {𝒮 : Slots} {Z : w.Lt} (l : Lv3 𝒮 K w Z) : ¬ LevelThreeCondition (fixH 𝒮 (Z.V.div.h B.u₀))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotMemPlaceholder

theorem **nonempty_stFams** (D : D7) : Nonempty (StFams D)

theorem **exists_mem** (st : StFams D) {A : Amb D} {N j : ℕ} {S : Sig A N j} {κ : Lv A} (s : FinalTheorems S κ) (L : CmList A) {jC : JC D} (hj : jC.Pinned) (hA : AdmEx S κ) (hP : ∀ (Φ : s.Flag) (x : ToroidalCompactificationData s), Nonempty (PolOf x Φ)) : ∃ k d, IsKitOf st A N j S κ s L jC k d

theorem **core_of_mem** (hi : 1 ≤ j ∧ j < N) {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) (adm : AdmSt D N j jC) : ∃ m : GlobalTheory.Mean4O.MetIn k.Φ, (GlobalTheory.Mean4O.Setup.ofKit hi k adm m).Core

theorem **not_mem_placeholder** (hb : (genBd s.T.Z : Set (gen s.T.Z : Scheme.{0})).Nonempty) (Φ : s.Flag) (key : KeySec s.Xo κ.c) (pad : PadPkg D N j s.Comp jC ngTrue) (d : s.Comp → D.what₂.Tau → ℝ) : ¬ IsKitOf st A N j S κ s L jC (KitEx.kitOf s L _ Φ key pad) d

theorem **qDiv_mb** {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) : ∃ bd : ∀ c, (k.Φ.C c).Y.Dv, (∀ c, (bd c).Eff ∧ ∀ y, y ∈ (bd c).supp ↔ (k.Φ.C c).π y ∈ genBd s.T.Z) ∧ ∀ (B : ℝ) (u : D.what₂.Tau → Fin N → ℚ) (r : ℚ), 0 < r → ∀ c, qDiv (k.Mb B (fun τ i => u τ i) r c) = qL k.Φ c u − r ⊗ₜ bd c

theorem **innerPt_spec** (Φ : s.Flag) (c : s.Comp) : IsClosed ({innerPt s Φ c} : Set (Φ.C c).Y.X) ∧ (Φ.C c).π (innerPt s Φ c) ∉ genBd s.T.Z

theorem **parOf_detSym** (a : ℕ → Bool) (τ : D.host.E →+* ℂ) : parOf (detSym S.Dw a none τ) = none

theorem **metsOf_none_h** (x x' : ToroidalCompactificationData s) (Φ : s.Flag) (c : s.Comp) (σ : D.what₂.LL →+* ℂ) (n : Letter S.Dw → ℤ) {z : Pts (Φ.C c).Y σ} (hz : (Φ.C c).π (under z) ∉ genBd s.T.Z) (a : (Φ.C c).Y.Fnˣ) : Var.Met.h (metsOf x Φ c none σ n) a z = Var.Met.h (metsOf x' Φ c none σ n) a z

theorem **ht_det_eq** (x x' : ToroidalCompactificationData s) (Φ : s.Flag) (c : s.Comp) (a : ℕ → Bool) (τ : D.host.E →+* ℂ) : (adlOf (Φ.C c) (metsOf x Φ c) (detSym S.Dw a none τ)).ht (innerPt s Φ c) = (adlOf (Φ.C c) (metsOf x' Φ c) (detSym S.Dw a none τ)).ht (innerPt s Φ c)

theorem **deg_unique** {k : KitT s L jC} {d d' : s.Comp → D.what₂.Tau → ℝ} {st' : StFams D} (h : IsKitOf st A N j S κ s L jC k d) (h' : IsKitOf st' A N j S κ s L jC k d') : d = d'

theorem **not_mem_add_one** {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) (τ : D.what₂.Tau) : ¬ IsKitOf st A N j S κ s L jC k (d + 1)

theorem **isBetti_of_mem** {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) (c : s.Comp) (τ : D.what₂.Tau) : IsFlat (k.Φ.C c) (DV.tauEmb D τ) (flatSec s k.Φ c τ) ∧ IsBetti s k.Φ L c τ (flatSec s k.Φ c τ) (k.fl c τ)

theorem **not_mem_fl** {k : KitT s L jC} {d d' : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) (τ : D.what₂.Tau) : ¬ IsKitOf st A N j S κ s L jC { k with fl

theorem **eq_setup_kit** (hi : 1 ≤ j ∧ j < N) {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) : ∃ (adm : AdmSt D N j jC) (m : GlobalTheory.Mean4O.MetIn k.Φ), (GlobalTheory.Mean4O.Setup.ofKit hi k adm m).Core ∧ (GlobalTheory.Mean4O.Setup.ofKit hi k adm m).pad = k.pad

theorem **ats_place** {F : StFam S κ} {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} {x : ToroidalCompactificationData s} {P : PolOf x k.Φ} {cts : Ats S κ} {hj : jC.Pinned} {hA : AdmEx S κ} (h : IsKitWith F s L jC k d x P cts hj hA) (p : ℕ) [Fact p.Prime] (v : PlL D p) (a : D.what₂.LL) : ((κ.e a : κ.c.F) : ℂ) ∈ (cts p v).W ↔ v.1.valuation D.what₂.LL a ≤ 1

theorem **normDet_of_mem** {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} (h : IsKitOf st A N j S κ s L jC k d) : ∃ (e : k.pad.Pt = Pt s) (x : ToroidalCompactificationData s), ∀ p τ y, k.pad.normDetT p τ y = normDetAt s p true τ (cast e y) ((flatSt s k.Φ (metsOf x k.Φ)).sv τ (cast e y)) ∧ k.pad.normDetH p τ y = normDetAt s p false τ (cast e y) ((flatSt s k.Φ (metsOf x k.Φ)).sv τ (cast e y))

theorem **deg_defect** {F : StFam S κ} {k : KitT s L jC} {d : s.Comp → D.what₂.Tau → ℝ} {x : ToroidalCompactificationData s} {P : PolOf x k.Φ} {cts : Ats S κ} {hj : jC.Pinned} {hA : AdmEx S κ} (h : IsKitWith F s L jC k d x P cts hj hA) (c : s.Comp) (τ : D.what₂.Tau) : d c τ = degFrame s k.Φ (metsOf x k.Φ) c τ + ((N − j : ℕ) : ℝ) * (2 * Real.log (placeRankTwoOne.rdK D) − Real.log D.host.rdE)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotNonemptyZero

theorem **locPar_unr** (hL : GlobalTheory.Published.LocalLanglandsGL) (χ : HChar E) (h : IsPlaceOver A w) (hu : χ.UnrAt w) : (ParLaws.locPar hL (CG.ofChar χ) h).unr ∧ (ParLaws.locPar hL (CG.ofChar χ) h).frobEig = {(((χ.val w)⁻¹ : ℂˣ) : ℂ)}

theorem **ordF_unif** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : ordF v (unif v) = 1

theorem **prod_val** (h : IsEtaT H χ) (v : HeightOneSpectrum (𝒪 H.F)) : ∏ P ∈ aboveF (L

theorem **above_eq** {w w' : PlE H} (hc : IsConjPl H w w') : aboveF (L

theorem **val_inert** (h : IsEtaT H χ) {w : PlE H} (hw : ¬ IsSplitPl H w) : χ.val w = −1

theorem **val_split** (h : IsEtaT H χ) {w w' : PlE H} (hw : IsSplitPl H w) (hc : IsConjPl H w w') : χ.val w * χ.val w' = 1

theorem **unr_symm** (w) (x) : D.naut.unr ((T.e w).symm x) ↔ x.unr

theorem **frob_symm** (w) (x) (h : x.unr) : D.naut.frobEig ((T.e w).symm x) = x.frobEig

theorem **dim_symm** (w) (x) : D.naut.lpDim ((T.e w).symm x) = x.lpDim

theorem **ofU_spec** {χ : HChar D.host.E} (h : χ.IsUnitary) {w} (hu : χ.UnrAt w) : D.naut.unr ((T.e w).symm (ParLaws.locG hL D.nd.plOm (ofU χ h) w)) ∧ D.naut.frobEig ((T.e w).symm (ParLaws.locG hL D.nd.plOm (ofU χ h) w)) = {(((χ.val w)⁻¹ : ℂˣ) : ℂ)}

theorem **exists_cgTrue** (hE : IsEmpty D.nd.Loc) : ∃ K : CGK D.naut, Nonempty (CGTrue hL (setCG D hE K) (wdCG hE K T))

theorem **exists_tie** (D : D7) (hE : IsEmpty D.nd.Loc) : ∃ D' : D7, IsEmpty D'.nd.Loc ∧ Nonempty (GlobalTheory.RowTie.WDTie D'.nd D'.naut)

theorem **exists_record** (hL : GlobalTheory.Published.LocalLanglandsGL) (hF : GlobalTheory.Published.AutomorphicFormsFiniteGL) (D : D7) (hE : IsEmpty D.nd.Loc) : ∃ (D' : D7) (T : GlobalTheory.RowTie.WDTie D'.nd D'.naut), Nonempty (CGTrue hL D' T)

theorem **ralg_iff** (C : CGTrue hL D T) {a} (π : D.naut.CG a) : haveI

theorem **not_nonempty_zero** (C : CGTrue hL D T) : ¬ Nonempty (D.naut.CG 0)

theorem **nontrivial_one** (C : CGTrue hL D T) : Nontrivial (D.naut.CG 1)

theorem **frob_one** (C : CGTrue hL D T) (π : D.naut.CG 1) (w) (hu : haveI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotRedDouble

theorem **join_over** [Mono (genι Z)] (C : FlagCpt Z e) (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) (w : Sp (DualNumber C.Y.Fn) ⟶ Z.oF.t.G.left) (hw : w ≫ Z.m.φ = C.f0.v τ i) : w ≫ Z.oF.t.G.hom = spMap (algebraMap C.Y.Fn (DualNumber C.Y.Fn)) ≫ (letI

theorem **mono_genι** (Y : Model Xo c (algebraMap (𝒪 c.F) c.F)) : Mono (genι Y)

theorem **Lw_weightOf** (C : FlagCpt Z e) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : C.Lw (GlobalTheory.AllWt.weightOf Dw (fun x => (a x : ℤ)) ((n / 2 : ℕ) : ℤ)) = C.LEx (exKey Dw a n)

theorem **exKey_none** (a : (E →+* ℂ) × ℕ → ℕ) {n : ℕ} (hn : Even n) : exKey Dw a n none = 0

theorem **dim_count** (f p q : ℕ) : f ∗ ((p + q) ∗ (p + q − 1) / 2) = f ∗ p ∗ q + f ∗ (p ∗ (p − 1) / 2) + f ∗ (q ∗ (q − 1) / 2)

theorem **genericPt_eq** (Y : GlobalTheory.ProjVar.Var F) (y : Y.X) : genericPt Y = spMap (algebraMap (Y.X.presheaf.stalk y) Y.X.functionField) ≫ stalkPt Y y

theorem **cover** {σ : c.F →+* ℂ} {N : ℕ} {U : Formalism.L2.PjFree.Unf Xo c σ N} (Φ : FlagTor Z e U) (z : gen Z) : ∃ i, z ∈ Set.range (Φ.C i).π

theorem **comp_eq** {σ : c.F →+* ℂ} {N : ℕ} {U : Formalism.L2.PjFree.Unf Xo c σ N} (Φ : FlagTor Z e U) {k : Type} [Field k] [IsAlgClosed k] {φ : c.F →+* k} {x : Sp k ⟶ gen Z} (hx : x ≫ genStr Z = spMap φ) (g : EigFrame Z φ (x ≫ genι Z)) {i j : U.Comp} {p : Sp k ⟶ (Φ.C i).Y.X} {q : Sp k ⟶ (Φ.C j).Y.X} (hp : p ≫ (Φ.C i).π = x) (hq : q ≫ (Φ.C j).π = x) (cp : (Φ.C i).Carries p g) (cq : (Φ.C j).Carries q g) : i = j

theorem **bd_eq_zero** (C : FlagCpt Z e) (h : ∀ y, C.π y ∉ genBd Z) : C.bd = 0

theorem **not_red_double** (Y : GlobalTheory.ProjVar.Var F) (Δ : Y.Dv) (hΔ : Y.Eff Δ) (y : Y.X) (hy : y ∈ Y.supp Δ) : ¬ ∀ (y : Y.X) (g : Y.Fnˣ), (2 • Δ).IsEqn y g → ∀ (f : Y.Fn) (n : ℕ), RegAt y f → RegAt y (f ^ n / g) → RegAt y (f / g)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.NotTopPtOne

theorem **adAt_mul** {N : ℕ} (P : HeightOneSpectrum (𝒪 E)) (a b : GL (Fin N) (AfE F E)) : ((a ∗ b : GL (Fin N) (AfE F E)).1).map (adAt F P) = a.1.map (adAt F P) ∗ b.1.map (adAt F P)

theorem **adAt_inv_one** {N : ℕ} (P : HeightOneSpectrum (𝒪 E)) {a : GL (Fin N) (AfE F E)} (h : a.1.map (adAt F P) = 1) : ((a⁻¹ : GL (Fin N) (AfE F E)).1).map (adAt F P) = 1

theorem **adAt_of_prV** {N : ℕ} (P : HeightOneSpectrum (𝒪 E)) {k : GL (Fin N) (AfE F E)} (h1 : bc (prV (E

theorem **prV_of_adAt** {N : ℕ} (v : HeightOneSpectrum (𝒪 F)) {k : GL (Fin N) (AfE F E)} (h : ∀ w : HeightOneSpectrum (𝒪 E), w.under (𝒪 F) = v → k.1.map (adAt F w) = 1) : bc (prV (E

theorem **topPt_base** {u : B.Word} {a : I.ι} {k : GA B (B.R.N B.R.m)} [NeZero (B.R.N B.R.m − 1)] (h : I.TopPt l Z u a k) : PtOf k (tilF (Pr

theorem **topPt_last** (hI : I.Ok) {u : B.Word} {a : I.ι} {k : GA B (B.R.N B.R.m)} [NeZero (B.R.N B.R.m − 1)] (h : I.TopPt l Z u a k) (P : I.Q a) (i : Fin (B.R.N B.R.m)) : adAt B.host.F P.1 (k.1 i (Br.iL (N_m_succ B.R))) = algebraMap B.host.E _ ((Pi.single (Br.iL (N_m_succ B.R)) ((X.piT u : 𝒪 B.host.E) : B.host.E) : Fin _ → B.host.E) i)

theorem **exists_topPt** (hI : I.Ok) (hP : TC.Pins B X) (u : B.Word) (a : I.ι) : ∃ k : GA B (B.R.N B.R.m), (∃ p : DecoF B.fields.E (B.R.N B.R.m) ⌀ ⌀ ⌀, PtOf k p ∧ p.deco.1.N = stdLat B.fields.E (B.R.N B.R.m)) ∧ (∀ v : HeightOneSpectrum (𝒪 B.host.F), (B.T u : 𝒪 B.host.F) ∉ v.asIdeal → ¬ I.UnderQ a v → bc (prV (E

theorem **not_topPt_one** (hI : I.Ok) {u : B.Word} {a : I.ι} [NeZero (B.R.N B.R.m − 1)] (P : I.Q a) (hpi : ((X.piT u : 𝒪 B.host.E) : B.host.E) ≠ 1) : ¬ I.TopPt l Z u a 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.OmPinsTr

theorem **lawsM_tr** {T : Core a.D} (h : LawsM T) : LawsM (core b T)

theorem **lawsJ_tr** {T : CoreJ a.D} (h : LawsJ T) : LawsJ (coreJ b T)

theorem **omPins_tr** {O : OMDat T R i Tp tup} {κ v₀ ι x₀} (h : O.Pins κ v₀ ι x₀) : @OMDat.Pins b.D G _ _ Pl0 T0 (famM b A) (core b T) ((@alg B C a b (ɪ(@ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff.F (@Formalism.HC.HostRaw.E (@Late.D B C a).R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.D B C a).R.toFrame (@Host.toHostAx (@Late.D B C a).R (@Late.D B C a).host))) (@Formalism.HC.HostRaw.instField (@Late.D B C a).R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.D B C a).R.toFrame (@Host.toHostAx (@Late.D B C a).R (@Late.D B C a).host))) (@Formalism.HC.HostRaw.instNF (@Late.D B C a).R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.D B C a).R.toFrame (@Host.toHostAx (@Late.D B C a).R (@Late.D B C a).host))) (instCMHost (@Late.D B C a)) (@Par.H (@Late.D B C a) (@Core.Q (@Late.D B C a) T)) (@Par.l (@Late.D B C a) (@Core.Q (@Late.D B C a) T)) (@Par.c (@Late.D B C a) (@Core.Q (@Late.D B C a)

theorem **inGeo_tr** {ph : Fin a.D.R.m → ℂ} {x : GeoHalf T R ph} (h : InGeo x) : @InGeo b.D G _ _ Pl0 T0 (famM b A) (coreJ b T) ((@alg B C a b (ɪ(@ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff.F (@Formalism.HC.HostRaw.E (@Late.D B C a).R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.D B C a).R.toFrame (@Host.toHostAx (@Late.D B C a).R (@Late.D B C a).host))) (@Formalism.HC.HostRaw.instField (@Late.D B C a).R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.D B C a).R.toFrame (@Host.toHostAx (@Late.D B C a).R (@Late.D B C a).host))) (@Formalism.HC.HostRaw.instNF (@Late.D B C a).R.toFrame (@Formalism.HC.HostAx.toHostRaw (@Late.D B C a).R.toFrame (@Host.toHostAx (@Late.D B C a).R (@Late.D B C a).host))) (instCMHost (@Late.D B C a)) (@Par.H (@Late.D B C a) (@Core.Q (@Late.D B C a) (@CoreJ.toCore (@Late.D B C a) T))) (@Par.l (@Late.D B C a) (@Core.Q (@Late.D B C a) (@CoreJ.toCore (@Late.D B C

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.OrZeroOf

theorem **levCh_is** {L : Formalism.IndefR.Links.LevCh D T} {f : T' → T} (hf : Function.Injective f) {μ : T'} (h : L.Is (f μ)) : (levCh L f).Is μ where

theorem **isGrp** (h : R.IsGrp (bare2 _ P0 ψ0 P1 ψ1)) : (dat2 f μ R mu tw).IsGrp (bare2 _ P0 ψ0 P1 ψ1)

theorem **isFam** (h : R.IsFam) : (dat2 f μ R mu tw).IsFam

theorem **isModels** (h : R.IsModels) : (dat2 f μ R mu tw).IsModels

theorem **exAt** (e : R.ExAt) : (dat2 f μ R mu tw).ExAt

theorem **orZero_of** {X : Type} [Zero X] {p : X → Prop} {x : X} (h : p x) : OrZero p x

theorem **isChainW** (h : R.IsChainW) (e : R.ExAt) : (dat2 f μ R mu tw).IsChainW

theorem **isThPW** (h : R.IsThPW) (e : R.ExAt) : (dat2 f μ R mu tw).IsThPW

theorem **baseW** (hf : Function.Injective f) (h : R.BaseW (bare2 _ P0 ψ0 P1 ψ1)) (e : R.ExAt) : (dat2 f μ R mu tw).BaseW (bare2 _ P0 ψ0 P1 ψ1)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.OscActMonomialOfUnique

theorem **oscAct_apply** (Z : Mat P Q) : oscAct P Q Z = ∑ l, ∑ l', Z l l' • unitOp l l'

theorem **oscAct_single** (a b : Ix P Q) (c : ℂ) : oscAct P Q (Matrix.single a b c) = c • unitOp a b

theorem **oscAct_Eab** (a b : Ix P Q) : oscAct P Q (Eab a b) = unitOp a b

theorem **unitOp_inl_inl** (i j : Fin P) (f : OscPol P Q) : unitOp (.inl i) (.inl j) f = X (.inl i) * pderiv (.inl j) f + if i = j then (1 / 2 : ℂ) • f else 0

theorem **unitOp_inl_inr** (i : Fin P) (j : Fin Q) (f : OscPol P Q) : unitOp (.inl i) (.inr j) f = X (.inl i) * (X (.inr j) * f)

theorem **unitOp_inr_inl** (i : Fin Q) (j : Fin P) (f : OscPol P Q) : unitOp (.inr i) (.inl j) f = -pderiv (.inr i) (pderiv (.inl j) f)

theorem **unitOp_inr_inr** (i j : Fin Q) (f : OscPol P Q) : unitOp (.inr i) (.inr j) f = -(X (.inr j) * pderiv (.inr i) f) - if i = j then (1 / 2 : ℂ) • f else 0

theorem **unitOp_diag** (l : Ix P Q) (α : Ix P Q →₀ ℕ) : unitOp l l (monomial α 1) = ((sgn l : ℂ) * (((α l : ℕ) : ℂ) + 1 / 2)) • monomial α 1

theorem **pderiv_pderiv_comm** (a b : Ix P Q) (f : OscPol P Q) : pderiv a (pderiv b f) = pderiv b (pderiv a f)

theorem **low_mul_low** (a b : Ix P Q) : low a * low b = low b * low a

theorem **rai_mul_rai** (a b : Ix P Q) : rai a * rai b = rai b * rai a

theorem **low_mul_rai** (a b : Ix P Q) : low a * rai b = rai b * low a + if a = b then 1 else 0

theorem **unitOp_comm** (i j k l : Ix P Q) : unitOp i j * unitOp k l - unitOp k l * unitOp i j = (if j = k then unitOp i l else 0) - if l = i then unitOp k j else 0

theorem **isLieAct_oscAct** (P Q : ℕ) : IsLieAct (oscAct P Q)

theorem **mfact_add_single** (α : σ →₀ ℕ) (l : σ) : FockSeq.mfact (α + Finsupp.single l 1) = (α l + 1) * FockSeq.mfact α

theorem **inner_X_mul** (h : IsFockIP ι) (l : σ) (f g : MvPolynomial σ ℂ) : inner ℂ (ι (X l * f)) (ι g) = inner ℂ (ι f) (ι (pderiv l g))

theorem **inner_pderiv** (h : IsFockIP ι) (l : σ) (f g : MvPolynomial σ ℂ) : inner ℂ (ι (pderiv l f)) (ι g) = inner ℂ (ι f) (ι (X l * g))

theorem **inner_rai** (h : IsFockIP ι) (l : Ix P Q) (f g : OscPol P Q) : inner ℂ (ι (rai l f)) (ι g) = (sgn l : ℂ) * inner ℂ (ι f) (ι (low l g))

theorem **inner_low** (h : IsFockIP ι) (l : Ix P Q) (f g : OscPol P Q) : inner ℂ (ι (low l f)) (ι g) = (sgn l : ℂ) * inner ℂ (ι f) (ι (rai l g))

theorem **inner_unitOp** (h : IsFockIP ι) (a b : Ix P Q) (f g : OscPol P Q) : inner ℂ (ι (unitOp a b f)) (ι g) = (sgn a : ℂ) * (sgn b : ℂ) * inner ℂ (ι f) (ι (unitOp b a g))

theorem **adjZ_apply** (Z : Mat P Q) (a b : Ix P Q) : adjZ Z a b = (sgn a : ℂ) * (sgn b : ℂ) * (starRingEnd ℂ) (Z b a)

theorem **inner_oscAct** (h : IsFockIP ι) (Z : Mat P Q) (f g : OscPol P Q) : inner ℂ (ι (oscAct P Q Z f)) (ι g) = inner ℂ (ι f) (ι (oscAct P Q (adjZ Z) g))

theorem **renum_lowB** (l : Ln p q r s) (f : Pol p q r s) : renum p q r s (lowB l f) = low (lineEquiv p q r s l) (renum p q r s f)

theorem **renum_raiB** (l : Ln p q r s) (f : Pol p q r s) : renum p q r s (raiB l f) = rai (lineEquiv p q r s l) (renum p q r s f)

theorem **renum_omE** (l l' : Ln p q r s) (f : Pol p q r s) : renum p q r s (omE l l' f) = unitOp (lineEquiv p q r s l) (lineEquiv p q r s l') (renum p q r s f)

theorem **renum_omAct** (Z : Mat (p * r + q * s) (p * s + q * r)) (f : Pol p q r s) : renum p q r s (omAct p q r s (Matrix.reindex (lineEquiv p q r s).symm (lineEquiv p q r s).symm Z) f) = oscAct _ _ Z (renum p q r s f)

theorem **isFockIsometry_comp_renameEquiv** {σ τ H : Type} [DecidableEq σ] [DecidableEq τ] [NormedAddCommGroup H] [InnerProductSpace ℂ H] {ι : MvPolynomial τ ℂ →ₗ[ℂ] H} (h : IsFockIP ι) (e : σ ≃ τ) : IsFockIP (ι ∘ₗ (renameEquiv ℂ e).toLinearMap)

theorem **isFockIP** (O : Osc P Q T0) : IsFockIP O.ι

theorem **norm_vacuum** (O : Osc P Q T0) : ‖O.ι 1‖ = 1

theorem **inner_oscAct** (O : Osc P Q T0) (Z : Mat P Q) (f g : OscPol P Q) : inner ℂ (O.ι (oscAct P Q Z f)) (O.ι g) = inner ℂ (O.ι f) (O.ι (oscAct P Q (adjZ Z) g))

theorem **isDer** (O : Osc P Q T0) {Z : Mat P Q} (hZ : Z ∈ upq P Q) (f : OscPol P Q) : IsDer T0 O.0m Z (O.ι f) (O.ι (oscAct P Q Z f))

theorem **smooth** (O : Osc P Q T0) (f : OscPol P Q) : Smooth T0 O.0m (O.ι f)

theorem **nonempty_fockData_iff** : Nonempty (FockData T T1 T2) ↔ Nonempty (Osc _ _ T)

theorem **oscAct_monomial_of_unique** (Z : Mat P Q) (α : Ix P Q →₀ ℕ) (c : ℂ) : oscAct P Q Z (monomial α c) = ((expo α : ℂ) * (Z.trace / 2)) • monomial α c

theorem **norm_sq_zpow** (g : T0.G) (α : Ix P Q →₀ ℕ) : ‖((T0.sq g : ℂˣ) : ℂ) ^ expo α‖ = 1

theorem **rankOneRep_monomial** (g : T0.G) (α : Ix P Q →₀ ℕ) (c : ℂ) : rankOneRep T0 g (FockSeq.emb _ (monomial α c)) = ((T0.sq g : ℂˣ) : ℂ) ^ expo α • FockSeq.emb _ (monomial α c)

theorem **loweringOperator_eq** (l : Ix P Q) : oscillatorLowering l = low l

theorem **raisingOperator_eq** (l : Ix P Q) : oscillatorRaising l = rai l

theorem **matrixUnitOperator_eq** (l l' : Ix P Q) : oscillatorMatrixUnit l l' = unitOp l l'

theorem **omAct0_eq** (P Q : ℕ) : omAct0 P Q = oscAct P Q

theorem **isOsc_fockOf** (O : GlobalTheory.Osc P Q T0) : IsOsc T0 (fockOf O) O.Om

theorem **IsOsc.fockOf_toOsc** {F : Fock (Ix P Q)} {Om : T0.G →* (F.H →L[ℂ] F.H)} (h : IsOsc T0 F Om) : fockOf h.toOsc = F

theorem **toOsc_isOsc_fockOf** (O : GlobalTheory.Osc P Q T0) : (isOsc_fockOf O).toOsc = O

theorem **nonempty_osc_iff** : Nonempty (GlobalTheory.Osc P Q T0) ↔ ∃ (F : Fock (Ix P Q)) (Om : T0.G →* (F.H →L[ℂ] F.H)), IsOsc T0 F Om

theorem **isOsc_rankOne** [Unique (Ix P Q)] (T0 : GrpData P Q) : IsOsc T0 (seqFock (Ix P Q)) (GlobalTheory.Osc.rankOneRep T0)

theorem **isOsc_circlePos** : IsOsc _ (fockOf GlobalTheory.Osc.circlePos) GlobalTheory.Osc.circlePos.Om

theorem **isOsc_circleNeg** : IsOsc _ (fockOf GlobalTheory.Osc.circleNeg) GlobalTheory.Osc.circleNeg.Om

theorem **not_isOsc_one** (T0 : GrpData 1 0) (F : Fock (Ix 1 0)) : ¬ IsOsc T0 F 1

theorem **hasCoeff_coe** {σ : Type} [DecidableEq σ] (F : Fock σ) (f : MvPolynomial σ ℂ) : HasCoeff F (F.ι f) (f : MvPowerSeries σ ℂ)

theorem **coeff_zero_gaussPoly** (A : Matrix (Fin P) (Fin Q) ℂ) (x : Ix P Q → ℂ) : coeff 0 (gaussPoly A x) = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PadTrueNormDetConst

theorem **padTrue_arDeg** : (padTrue s Φ hm jC hj hA L G).arDeg = (arIn s Φ hm).arDeg

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PairVEqZeroOfMemLieSpan

theorem **ospan_eq_span** {C V : Type} [NormedField C] [IsUltrametricDist C] [AddCommGroup V] [Module C V] [Module ɪ(intO C) V] [IsScalarTower ɪ(intO C) C V] (s : Set V) : Ospan (C

theorem **span_tateIncl** : Submodule.span ɪC.S (Set.range (tateIncl P)) = ⊤

theorem **alC_tateIncl** (t : P.T) : P.alC (tateIncl P t) = htMap P t

theorem **htToMdl_φ** {Z : ct.Mdl} (y : ct.Pt Z) : letI : Fact p.Prime

theorem **isGrpPullback_htToMdl** {Z : ct.Mdl} (y : ct.Pt Z) : letI : Fact p.Prime

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PeriodComparisonProvisional

theorem **samePt_of_heq** {r j p r' j' : ℕ} {Λ : ModuliIntegralCharts.Loc D r j p} {Λ' : ModuliIntegralCharts.Loc D r' j' p} (hr : r = r') (hj : j = j') (hΛ : HEq Λ Λ') {y : Λ.Pt Λ.YY} {y' : Λ'.Pt Λ'.YY} (hy : HEq y y') : Λ.fld y = Λ'.fld y' ∧ (∃ e : Λ.Ylat y ≃ₗ[𝒪 D.host.E] Λ'.Ylat y', ∀ a b, Λ'.be y' (e a) (e b) = Λ.be y a b) ∧ (Λ.good y ↔ Λ'.good y') ∧ Λ.phiCy y = Λ'.phiCy y' ∧ Λ.dP y = Λ'.dP y'

theorem **lat_of_heq** {r j p r' j' : ℕ} {Λ : ModuliIntegralCharts.Loc D r j p} {Λ' : ModuliIntegralCharts.Loc D r' j' p} (hr : r = r') (hj : j = j') (hΛ : HEq Λ Λ') {y : Λ.Pt Λ.YY} {y' : Λ'.Pt Λ'.YY} (hy : HEq y y') {Y : Type} [AddCommGroup Y] [Module (𝒪 D.host.E) Y] (β : Y → Y → ℝ) (hY : ∃ e : Λ.Ylat y ≃ₗ[𝒪 D.host.E] Y, ∀ a b, β (e a) (e b) = Λ.be y a b) : ∃ e : Λ'.Ylat y' ≃ₗ[𝒪 D.host.E] Y, ∀ a b, β (e a) (e b) = Λ'.be y' a b

theorem **pt_eq_of_heq** {r j p r' j' : ℕ} {Λ : ModuliIntegralCharts.Loc D r j p} {Λ' : ModuliIntegralCharts.Loc D r' j' p} (hr : r = r') (hj : j = j') (hΛ : HEq Λ Λ') : Λ.Pt Λ.YY = Λ'.Pt Λ'.YY

theorem **loc_heq_of_eq** {P P' : ModuliIntegralCharts.Prov D} (hP : P = P') {d : P.Dat} {d' : P'.Dat} (hd : HEq d d') (p : ℕ) : HEq (P.loc d p) (P'.loc d' p)

theorem **rk_dat_eq** (h : LawsP T) {Lev : Type} (κ : Lev) {j : ℕ} (h1 : 1 ≤ j) (hm : j < D.R.m) : T.G.P.rk (T.G.dat j) = (datW D κ j).N

theorem **jj_dat_eq** (h : LawsP T) {Lev : Type} (κ : Lev) {j : ℕ} (h1 : 1 ≤ j) (hm : j < D.R.m) : T.G.P.jj (T.G.dat j) = (datW D κ j).j

theorem **LocTie.pts** (H : LocTie h k) : PtsTie h k

theorem **LocTie.samePt** (H : LocTie h k) {j : ℕ} (h1 : 1 ≤ j) (hm : j < D.R.m) {p : ℕ} {η : T.G.PY j p} {η' : ((k.at (datW D F.κg j)).pad.loc p).Pt ((k.at (datW D F.κg j)).pad.loc p).YY} (he : HEq η η') : (T.G.P.loc (T.G.dat j) p).fld η = ((k.at (datW D F.κg j)).pad.loc p).fld η' ∧ (∃ e : (T.G.P.loc (T.G.dat j) p).Ylat η ≃ₗ[𝒪 D.host.E] ((k.at (datW D F.κg j)).pad.loc p).Ylat η', ∀ a b, ((k.at (datW D F.κg j)).pad.loc p).be η' (e a) (e b) = (T.G.P.loc (T.G.dat j) p).be η a b) ∧ ((T.G.P.loc (T.G.dat j) p).good η ↔ ((k.at (datW D F.κg j)).pad.loc p).good η') ∧ (T.G.P.loc (T.G.dat j) p).phiCy η = ((k.at (datW D F.κg j)).pad.loc p).phiCy η' ∧ (T.G.P.loc (T.G.dat j) p).dP η = ((k.at (datW D F.κg j)).pad.loc p).dP η'

theorem **LocTie.lat** (H : LocTie h k) {j : ℕ} (h1 : 1 ≤ j) (hm : j < D.R.m) {la : HeightOneSpectrum (𝒪 T.Q.c.F)} {z : PointsOfFlagModel.Pt (T.G.Tw j (WeightAlgebra.resCh la)) .yy} (hz : T.G.IsAlgPt j la z) {η : T.G.PY j (WeightAlgebra.resCh la)} (hr : T.G.rel j _ η z) {η' : ((k.at (datW D F.κg j)).pad.loc (WeightAlgebra.resCh la)).Pt ((k.at (datW D F.κg j)).pad.loc (WeightAlgebra.resCh la)).YY} (he : HEq η η') (A : DatAt ((T.G.Tw j (WeightAlgebra.resCh la)).Y z.1) z.2.2.1) : ∃ e : ((k.at (datW D F.κg j)).pad.loc (WeightAlgebra.resCh la)).Ylat η' ≃ₗ[𝒪 D.host.E] A.Ylat, ∀ a b, A.beta (e a) (e b) = ((k.at (datW D F.κg j)).pad.loc (WeightAlgebra.resCh la)).be η' a b

theorem **LocTie.localTie_of_provisional** (K : CutSt C) (R : ResSt C) {I : Type} {dat : I → Dat D (Lv (ambOf T.Q h.pins.1))} {hI : ∀ i, (dat i).Indef} {iW : ℕ → I} (hiW : ∀ j, 1 ≤ j → j < D.R.m → dat (iW j) = datW D F.κg j) (hP : T.G.P = indefiniteDataFamily (shimOf K R k) I dat hI) (hd : ∀ j, 1 ≤ j → j < D.R.m → HEq (T.G.dat j) (iW j)) : LocTie h k

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PlaceCharacterEqReferenceIff

theorem **injective_of_norm_sq** {F : Face D A} {b : Bool} {V W : Type} [NormedAddCommGroup V] [InnerProductSpace ℂ V] [NormedAddCommGroup W] [InnerProductSpace ℂ W] (e : V →ₗ[ℂ] W) (h : ∀ x, ‖e x‖ ^ 2 = RealDat.nP F b * ‖x‖ ^ 2) : Function.Injective e

theorem **inner_of_norm_sq** {F : Face D A} {b : Bool} {V W : Type} [NormedAddCommGroup V] [InnerProductSpace ℂ V] [NormedAddCommGroup W] [InnerProductSpace ℂ W] (e : V →ₗ[ℂ] W) (h : ∀ x, ‖e x‖ ^ 2 = RealDat.nP F b * ‖x‖ ^ 2) (x y : V) : ⟪e x, e y⟫_ℂ = ((RealDat.nP F b : ℝ) : ℂ) * ⟪x, y⟫_ℂ

theorem **eE_injective** : Function.Injective Y.eE

theorem **eW_injective** (i : Fin D.R.m) : Function.Injective (Y.eW i)

theorem **isot_eq_of_separated** (h : Y.Separated) (i : Fin D.R.m) (π : D.what₂.FT) : D.what₂.isot ((i : ℕ) + 1) π = (R.Wp i (Y.ty π)).comap (Y.eW i)

theorem **eW_injective** (i : Fin D.R.m) : Function.Injective (Y.eW i)

theorem **iSup_isot** (i : Fin D.R.m) (π : D.what₂.FT) : (⨆ π', ⨆ (_ : Y.ty π' = Y.ty π), D.what₂.isot ((i : ℕ) + 1) π') = (R.Wp i (Y.ty π)).comap (Y.eW i)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PlaceModelImp

theorem **Stages.late_st** (X : Stages) : X.toLate.st = X

theorem **cd_eq** (c c' : CDHaar) : c = c'

theorem **autFun_imp** : ∀ V f, @IsAutFun a.D V f → @IsAutFun b.D ((V)) ((f))

theorem **autFun_eq** : @IsAutFun b.D = @autFunB B C a b

theorem **autFunSm_imp** : ∀ θ V f, @IsAutFunSm a.D θ V f → @IsAutFunSm b.D ((θ)) ((V)) ((f))

theorem **autFunSm_eq** : @IsAutFunSm b.D = @autFunSmB B C a b

theorem **multMap_imp** : ∀ V S θ M J χ f, @IsMultMap a.D V S θ M J χ f → @IsMultMap b.D ((V)) ((S)) ((θ)) ((ds b M)) ((J)) ((χ)) ((f))

theorem **multMap_eq** : @IsMultMap b.D = @multMapB B C a b

theorem **thetaFun_imp** : ∀ ψ S T f, @IsThetaFunO a.D ψ S T f → @IsThetaFunO b.D ((ψ)) ((S)) ((T)) ((f))

theorem **thetaFun_eq** : @IsThetaFunO b.D = @thetaFunB B C a b

theorem **vacuum_imp** : ∀ m n x1 x2 x3 x4 x5 x6 x7, @IsVacuum a.D m n x1 x2 x3 x4 x5 x6 x7 → @IsVacuum b.D m n ((x1)) ((x2)) ((x3)) ((x4)) ((x5)) ((x6)) ((x7))

theorem **vacuum_eq** : @IsVacuum b.D = @vacuumB B C a b

theorem **placeModel_imp** : ∀ m n x1 x2 x3 x4 x5 x6 x7 x8, @IsPlaceModel a.D m n x1 x2 x3 x4 x5 x6 x7 x8 → @IsPlaceModel b.D m n ((x1)) ((x2)) ((x3)) ((x4)) ((x5)) ((x6)) ((x7)) ((x8))

theorem **placeModel_eq** : @IsPlaceModel b.D = @placeModelB B C a b

theorem **itW_imp** : ∀ V W S S' θ M M' ω pt It, @IsItW a.D V W S S' θ M M' ω pt It → @IsItW b.D ((V)) ((W)) ((S)) ((S')) ((θ)) ((ds b M)) ((ds b M')) ((ω)) ((pt)) ((It))

theorem **itW_eq** : @IsItW b.D = @itWB B C a b

theorem **pairSetup_imp** : ∀ ψ θ S T x1 x2 x3 x4 x5 x6, @Fock.Sk.PairSetup a.D ψ θ S T x1 x2 x3 x4 x5 x6 → @Fock.Sk.PairSetup b.D ((ψ)) ((θ)) ((S)) ((T)) ((x1)) ((x2)) ((x3)) ((x4)) ((x5)) ((x6))

theorem **pairSetup_eq** : @Fock.Sk.PairSetup b.D = @pairSetupB B C a b

theorem **inertTie_imp** : ∀ G i1 i2 P I Ty T m A x1 x2 x3, @Bridge.InertTie a.D G i1 i2 P I Ty T m A x1 x2 x3 → @Bridge.InertTie b.D G i1 i2 P I Ty T m ((A)) ((x1)) ((x2)) ((x3))

theorem **inertTie_eq** : @Bridge.InertTie b.D = @inertTieB B C a b

theorem **famTie_imp** : ∀ G i1 i2 P T m A cR cL lR lL x x1 x2 x3 x4 x5 x6 x7 x8 x9, @Bridge.FamTie a.D G i1 i2 P T m A cR cL lR lL x x1 x2 x3 x4 x5 x6 x7 x8 x9 → @Bridge.FamTie b.D G i1 i2 P T m ((A)) ((cR)) ((cL)) ((lR)) ((lL)) ((@famReal B C a b G i1 i2 P T m A cR cL lR lL x)) ((x1)) ((x2)) ((x3)) ((x4)) ((x5)) ((x6)) ((x7)) ((x8)) ((x9))

theorem **famTie_eq** : @Bridge.FamTie b.D = @famTieB B C a b

theorem **weilDatum_imp** : ∀ ψ S T x1 x2 ω, @IsWeilDatumO a.D ψ S T x1 x2 ω → @IsWeilDatumO b.D ((ψ)) ((S)) ((T)) ((x1)) ((x2)) ((ω))

theorem **weilDatum_eq** : @IsWeilDatumO b.D = @weilDatumB B C a b

theorem **weilFull_imp** : ∀ ψ S T x1 x2 ω, @WeilFull.IsWeilFull a.D ψ S T x1 x2 ω → @WeilFull.IsWeilFull b.D ((ψ)) ((S)) ((T)) ((x1)) ((x2)) ((ω))

theorem **weilFull_eq** : @WeilFull.IsWeilFull b.D = @weilFullB B C a b

theorem **mult_eq** : @Mult.Mult b.D = @multB B C a b

theorem **multT_eq** : @MultT b.D = @multTB B C a b

theorem **castAlong_rfl** {α : Type} (x : α) : (castAlong rfl x).1 = x

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PointPropertyEqSizeFree

theorem **has_pointProperty_iff** {C : FC} (p : Pt C) (QO : ∀ D : D7, Obj2 D → Prop) : p.Has (pointPropertyOfObjects QO) ↔ QO p.D (pointObj p)

theorem **pointPropertyOfObjects_eq_ptO** (QO : ∀ D : D7, Obj2 D → Prop) : pointPropertyOfObjects QO = ptO (fun D O => QO D (ofObj O))

theorem **ofObj_ofPt** {C : FC} (p : Pt C) : ofObj (Obj.ofPt p) = pointObj p

theorem **ofObj_obj** {C : FC} {D : D7} (Y : Lay D) (p : LedPart C Y) : ofObj (Y.obj p) = layObj Y

theorem **ofObj_tr** {D : D7} (O : Obj D) (N : ∀ u, Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) (h : D.what₂.Word → ℝ) : ofObj (O.tr N h) = (ofObj O).tr N h

theorem **layObj_ofPt** {C : FC} (p : Pt C) : layObj (Lay.ofPt p) = pointObj p

theorem **layObj_injective** {D : D7} : Function.Injective (layObj (D

theorem **pointObj_pt** {C : FC} {D : D7} {Y : Lay D} (p : LedPart C Y) : pointObj p.pt = (layObj Y).tr (nPrime p.L) p.ht

theorem **allDiv.has_pt** {QO : ∀ D : D7, Obj2 D → Prop} {C : FC} {D : D7} {Y : Lay D} (H : allDiv QO D Y) (p : LedPart C Y) : p.pt.Has (pointPropertyOfObjects QO)

theorem **allDiv.ofPt_pt** {QO : ∀ D : D7, Obj2 D → Prop} {C : FC} {D : D7} {Y : Lay D} (H : allDiv QO D Y) (p : LedPart C Y) : QO p.pt.D (ofObj (Obj.ofPt p.pt))

theorem **Obj2.tr_X** {D : D7} (O : Obj2 D) (N : ∀ u, Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) (h : D.what₂.Word → ℝ) : (O.tr N h).X = divXtr O.X N h

theorem **Obj2.tr_tr** {D : D7} (O : Obj2 D) (N N' : ∀ u, Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) (h h' : D.what₂.Word → ℝ) : (O.tr N h).tr N' h' = O.tr N' h'

theorem **pointProperty_and** (Q Q' : ∀ D : D7, Obj2 D → Prop) : pointPropertyOfObjects (fun D O => Q D O ∧ Q' D O) = (pointPropertyOfObjects Q).and (pointPropertyOfObjects Q')

theorem **allDiv_and_iff** (Q Q' : ∀ D : D7, Obj2 D → Prop) (D : D7) (Y : Lay D) : allDiv (fun D O => Q D O ∧ Q' D O) D Y ↔ allDiv Q D Y ∧ allDiv Q' D Y

theorem **allDiv_mono** {Q Q' : ∀ D : D7, Obj2 D → Prop} (hQ : ∀ D O, Q D O → Q' D O) {D : D7} {Y : Lay D} (H : allDiv Q D Y) : allDiv Q' D Y

theorem **nonempty_objects_iff** (D : D7) : Nonempty (Obj2 D) ↔ Nonempty (ThetaX D) ∧ Nonempty (ErrX D) ∧ Nonempty (NChoice D) ∧ Nonempty (ThData D) ∧ Nonempty (LevelData D) ∧ Nonempty (DivX D) ∧ (∃ P : BaseLevelRowSide D, Nonempty (AmbForm D P.s)) ∧ Nonempty (AmbForm D (RowSide.sharp D)) ∧ Nonempty (letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PolyRep

theorem **ev_proj** (τ : ι) {R : Type} [CommRing R] (A : Mat m R) : (proj m τ).ev A = A τ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PositionOfIntro

theorem **positionOf_intro** {P : CP} {K : Ctx} {B : Base} {f : Fr B.toPre} {T : TwDat B} {w : ColW B f T} {Z : w.Lt} {c : V4.Core K (w.st Z) f T} (h : P K B f T w Z c) : positionOf P K (w.st Z) f T c

theorem **positionOf_ind** {P : CP} {R : V4.Slot} (h : ∀ K B f T (w : ColW B f T) (Z : w.Lt) (c : V4.Core K (w.st Z) f T), P K B f T w Z c → R K (w.st Z) f T c) {K : Ctx} {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {c : V4.Core K X f T} (hc : positionOf P K X f T c) : R K X f T c

theorem **positionOf_mono** {P P' : CP} (h : ∀ K B f T w Z c, P K B f T w Z c → P' K B f T w Z c) (K : Ctx) (X : Stages) (f : Fr X.B.toPre) (T : TwDat X.B) (c : V4.Core K X f T) (hc : positionOf P K X f T c) : positionOf P' K X f T c

theorem **StepW.of_all** {p q : WP} (h : ∀ K B f T w, K.Over B.toPre → Forced K B T → p K B f T w → q K B f T w) : StepW p q

theorem **StepW.trans** {p q r : WP} (h : StepW p q) (h' : StepW q r) : StepW p r

theorem **StepW.and** {p q r : WP} (h : StepW p q) (h' : StepW p r) : StepW p fun K B f T w => q K B f T w ∧ r K B f T w

theorem **step_hasCore** {admitted : AtP} {p : WP} {P : CP} (h0 : Step KPx admitted fun K B f T => ∃ w : ColW B f T, p K B f T w) (h1 : StepW p (HasC P)) : Step KPx admitted (HasCore (positionOf P))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PredMemGamRanks

theorem **frMul_mul** (γ δ : GL (Fin n) E) (y : Fr emb σ n) : frMul emb σ (γ * δ) y = frMul emb σ γ (frMul emb σ δ y)

theorem **frMul_one** (y : Fr emb σ n) : frMul emb σ 1 y = y

theorem **trFr_apply** (γ : GL (Fin n) E) (c : Kˣ) (F : Fr emb σ n → ℂ) (y : Fr emb σ n) : trFr emb σ γ c F y = σ c * F (frMul emb σ γ⁻¹ y)

theorem **exists_mv** (b : S.β) (d : S.D b) : ∃ g : S.G b, g • S.rep (S.cl b d) = d

theorem **mv_smul** (b : S.β) (d : S.D b) : S.mv b d • S.rep (S.cl b d) = d

theorem **autO_finite** (hE : PosStar E) {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) : Finite (autO N)

theorem **card_autO_gmap** {g : Sp E n ≃ₗ[E] Sp E n} (hg : g ∈ UF E n) (N : Submodule (𝒪 E) (Sp E n)) : Nat.card (autO (Formalism.HermitianLatticesTheory.gmap g N)) = Nat.card (autO N)

theorem **autOrd_finite** (hE : PosStar E) (n : ℕ) : (autOrd E n).Finite

theorem **gamOrd_finite** (hE : PosStar E) (R : Ranks) : (gamOrd E R).Finite

theorem **dvd_DGam** (hE : PosStar E) (R : Ranks) (hn : n ∈ gamRanks R) {N : Submodule (𝒪 E) (Sp E n)} (h1 : Submodule.IsLattice E N) (h2 : Formalism.HermitianLatticesTheory.IsUnimodular (hStd E n) N) : Nat.card (autO N) ∣ DGam E R

theorem **DGam_dvd** (hE : PosStar E) (R : Ranks) {M : ℕ} (h : ∀ a ∈ gamOrd E R, a ∣ M) : DGam E R ∣ M

theorem **DGam_pos** (hE : PosStar E) (R : Ranks) : 0 < DGam E R

theorem **N_mem_gamRanks** (R : Ranks) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ R.m) : R.N j ∈ gamRanks R

theorem **pred_mem_gamRanks** (R : Ranks) : R.N R.m − 1 ∈ gamRanks R

theorem **nV_mem_gamRanks** (R : Ranks) : R.nV ∈ gamRanks R

theorem **N_even** (R : Ranks) (j : ℕ) : ∃ a : ℕ, R.N j = 2 * a

theorem **floor_sub_half** {R : Ranks} (P : HCParam R) (i : Fin R.m) (j : ℕ) : ((⌊P.e i − (R.N j : ℚ) / 2⌋ : ℤ) : ℚ) = P.e i − (R.N j : ℚ) / 2

theorem **stabMap_inj** (h : Q.Ok) (c : Q.S'.Cl) : Function.Injective (Q.stabMap h c)

theorem **prefRep_spec** {G D : Type} [Group G] [MulAction G D] (pref : D → Prop) (q : MulAction.orbitRel.Quotient G D) : Quotient.mk (MulAction.orbitRel G D) (prefRep pref q) = q

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PrimInIffSubsingleton

theorem **LieN.of_mul** (hu : IsUp u) (h : LieN σ J (P * u) Y A) : LieN σ J P Y (u * A * u⁻¹)

theorem **LieN.of_isom** (hJ : IsUnit J.det) (hs : IsUn σ J s) (h : LieN σ J (s * P) Y A) : LieN σ J P (s⁻¹ * Y * s) A

theorem **Relevant.mul** (h : Relevant σ J P P') (hu : IsUp u) (hv : IsUp v) : Relevant σ J (P * u) (P' * v)

theorem **Relevant.isom** (hJ : IsUnit J.det) (hs : IsUn σ J s) (h : Relevant σ J P P') : Relevant σ J (s * P) (s * P')

theorem **relevant_self** (hP : Free P) : Relevant σ J P P

theorem **eps_mul_eps** : eps E r * eps E r = 1

theorem **LiesAt.mul** {w : Formalism.MainNotions.S.Point E (r + 1)} (h : LiesAt w Z) (hv : IsLo v) (hu : IsUp u) : LiesAt w (v * Z * u)

theorem **liesAt_ptOfMat** (h : ∃ w, LiesAt w Z) : LiesAt (ptOfMat Z) Z

theorem **ptOfMat_congr** {Z' : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (h : ∀ w, LiesAt w Z ↔ LiesAt w Z') : ptOfMat Z = ptOfMat Z'

theorem **Gq_mul** (T T' : E) (hv : IsLo v) (hu : IsUp u) : Gq σ T T' (v * Z * u) = Gq σ T T' Z

theorem **map_Gq** (hσ : ∀ x, σ (σ x) = x) {T T' : E} (hT : σ T = T) (hT' : σ T' = T') : σ (Gq σ T T' Z) = Gq σ T T' Z

theorem **Gq_TF** {T : E} (hT : σ T = T) : Gq σ T T (TF r T) = 0

theorem **Unip.norm_eq** {T T' : E} (h : Unip P P') (hP : IsFrame σ J T P) (hP' : IsFrame σ J T' P') : T' = T

theorem **mem_defLat** (x : Fin n → E) : x ∈ defLat g Λ P P' ↔ x ∈ Λ ∧ (g : E) • x ∈ frameLat P ⊔ frameLat P'

theorem **defLat_eq_pairLat** (hg : g ≠ 0) (hP : PrimIn (frameLat P) Λ) (hu : u ∈ UO E r) (h : Cong g Λ (P * u) P') : defLat g Λ P P' = pairLat g (P * u) P'

theorem **defectAt_iff** (F : Type) [Field F] [Algebra F E] (p : HeightOneSpectrum (𝒪 F)) : DefectAt F M Λ p ↔ ∃ x ∈ Λ, x ∈ Submodule.span E (M : Set (Fin n → E)) ∧ ∀ a : 𝒪 F, a ∉ p.asIdeal → algebraMap (𝒪 F) (𝒪 E) a • x ∉ M

theorem **primIn_iff_subsingleton** (h : M ≤ Λ) : PrimIn M Λ ↔ Subsingleton (Defect M Λ)

theorem **primIn_iff_defectAt** (F : Type) [Field F] [NumberField F] [Algebra F E] (h : M ≤ Λ) : PrimIn M Λ ↔ ∀ p, ¬ DefectAt F M Λ p

theorem **defectAt_latMap** (F : Type) [Field F] [Algebra F E] (hs : IsUnit s.det) (p : HeightOneSpectrum (𝒪 F)) : DefectAt F (latMap s M) (latMap s Λ) p ↔ DefectAt F M Λ p

theorem **exists_UO** (hP : Free P) (h : PrimIn (frameLat P) Λ) (h' : PrimIn (frameLat P) (latMap s Λ)) (hs : InS σ J P s) : ∃ a ∈ UO E r, s ∗ P = P ∗ a

theorem **Unip.exists_UO** (h : Unip P P') (hP : IntFrame σ J T Λ P) (hP' : frameLat P' ≤ Λ) : ∃ a ∈ UO E r, P' = P ∗ a

theorem **tower_admissible_self** (h : IntFrame σ J T Λ P) : admitted σ J T g Λ P P

theorem **Transp.refl** : Transp σ J P Λ A Λ A

theorem **Transp.symm** (hJ : IsUnit J.det) (h : Transp σ J P Λ A Λ' B) : Transp σ J P Λ' B Λ A

theorem **Transp.trans** (h : Transp σ J P Λ A Λ' B) (k : Transp σ J P Λ' B Λ'' C) : Transp σ J P Λ A Λ'' C

theorem **move_equivalence** (hJ : IsUnit J.det) : Equivalence (Move σ J Λ P)

theorem **Move.transp** (hP : IsFrame σ J T P) (h : Move σ J Λ P A B) : Transp σ J P Λ A Λ B

theorem **Transp.pairM_eq** (h : Transp σ J P Λ A Λ' B) : ∃ l u, IsLo l ∧ IsUp u ∧ pairM σ J B P = l ∗ pairM σ J A P ∗ u

theorem **Transp.Gq_eq** (h : Transp σ J P Λ A Λ' B) (T T' : E) : Gq σ T T' (pairM σ J B P) = Gq σ T T' (pairM σ J A P)

theorem **Transp.lead_eq** (h : Transp σ J P Λ A Λ' B) (l : ℕ) (hl : l ≤ r + 1) : SgAlg.lead (pairM σ J B P) l hl = SgAlg.lead (pairM σ J A P) l hl

theorem **Transp.liesAt_iff** (hJ : IsUnit J.det) (h : Transp σ J P Λ A Λ' B) (w : Formalism.MainNotions.S.Point E (r + 1)) : LiesAt w (pairM σ J B P) ↔ LiesAt w (pairM σ J A P)

theorem **Transp.relevant_iff** (hJ : IsUnit J.det) (h : Transp σ J P Λ A Λ' B) : Relevant σ J P B ↔ Relevant σ J P A

theorem **Transp.unip_iff** (hJ : IsUnit J.det) (h : Transp σ J P Λ A Λ' B) : Unip P B ↔ Unip P A

theorem **Transp.sameFlag_iff** (hJ : IsUnit J.det) (h : Transp σ J P Λ A Λ' B) : SameFlag P B ↔ SameFlag P A

theorem **Transp.adm** (hJ : IsUnit J.det) (hP : Free P) (h1 : PrimIn (frameLat P) Λ) (h2 : PrimIn (frameLat P) Λ') (h : Transp σ J P Λ A Λ' B) (hA : admitted σ J T' g Λ P A) : admitted σ J T' g Λ' P B

theorem **Transp.defLat_eq** (hJ : IsUnit J.det) (hP : Free P) (h1 : PrimIn (frameLat P) Λ) (h2 : PrimIn (frameLat P) Λ') (h : Transp σ J P Λ A Λ' B) : ∃ s, IsUnit s.det ∧ latMap s Λ = Λ' ∧ latMap s (defLat g Λ P A) = defLat g Λ' P B

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PullRigIsSymmetric

theorem **gm_combGen** (F : J → (X' ⟶ X)) (δ : J → ℤ) (φ : K → (Y ⟶ X')) (ε : K → ℤ) (t : ∀ i, LB.Gen (LB.LinComb (L.pull (F i)) φ ε)) (r : Z ⟶ Y) (z : Z) (jj : K → (LB.LinComb L F δ).ι) (h : r z ∈ LB.wchart (LB.LinComb L F δ) φ jj) : gm r z (LB.wchart (LB.LinComb L F δ) φ jj) ((combGen L F δ φ ε t).s jj) = ∏ i, gm r z (LB.wchart (L.pull (F i)) φ (fun k _ => jj k i)) ((t i).s (fun k _ => jj k i)) ^ δ i

theorem **trF_comb** (F : J → (X' ⟶ X)) (δ : J → ℤ) (φ : K → (Y ⟶ X')) (ε : K → ℤ) (r : Z ⟶ Y) (z : Z) (jj jj' : K → (LB.LinComb L F δ).ι) (h : r z ∈ LB.wchart (LB.LinComb L F δ) φ jj) (h' : r z ∈ LB.wchart (LB.LinComb L F δ) φ jj') : trF (L

theorem **isGen_combGen** (F : J → (X' ⟶ X)) (δ : J → ℤ) (φ : K → (Y ⟶ X')) (ε : K → ℤ) (t : ∀ i, LB.Gen (LB.LinComb (L.pull (F i)) φ ε)) (ht : ∀ i, (t i).IsGen) : (combGen L F δ φ ε t).IsGen

theorem **good_factor** {F : J → (X' ⟶ X)} {δ : J → ℤ} {z : Z} {I : (Z ⟶ X') → (LB.LinComb L F δ).ι} (hI : Good (LB.LinComb L F δ) z I) (i : J) : Good (L.pull (F i)) z (fun χ _ => I χ i)

theorem **val_combGen** {F : J → (X' ⟶ X)} {δ : J → ℤ} {φ : K → (Y ⟶ X')} {ε : K → ℤ} (t : ∀ i, LB.Gen (LB.LinComb (L.pull (F i)) φ ε)) (r : Z ⟶ Y) {z : Z} {I : (Z ⟶ X') → (LB.LinComb L F δ).ι} (hI : Good (LB.LinComb L F δ) z I) : val (combGen L F δ φ ε t) r z I = ∏ i, val (t i) r z (fun χ _ => I χ i) ^ δ i

theorem **isCub_prod** {B U J : Type} [CommGroup B] [CommGroup U] [Fintype J] (θ : J → B → B → B → U) (h : ∀ i, IsCub (θ i)) (δ : J → ℤ) : IsCub fun x y z => ∏ i, θ i x y z ^ δ i where

theorem **pullRig_isSymmetric** {L : RigLB G} (c : CubicalStr G L) (hs : L.IsSymmetric) (f : A ⟶ G) [IsMonHom f] : (pullRig L f).IsSymmetric

theorem **isSymmetric_iff** (L : RigLB G) : L.IsSymmetric ↔ ∃ sy, BoxCompare.IsCanonSym L.L sy

theorem **combRig_isSymmetric** {L : RigLB G} (c : CubicalStr G L) (hs : L.IsSymmetric) : (combRig u δ L).IsSymmetric

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PullbackAndTensorModule

theorem **gramE_isHermitian** : Λ.gramE.IsHermitian

theorem **form_eq_sum** (x y : Fin Λ.n → E) : Λ.form x y = ∑ i, ∑ j, x i ∗ Λ.gramE i j ∗ star (y j)

theorem **mem_iff** (x : Fin Λ.n → E) : x ∈ Λ.L ↔ ∀ y ∈ Λ.L, ∃ c : 𝒪 E, (c : E) = Λ.form x y

theorem **nondegenerate** : Λ.form.Nondegenerate

theorem **finrank_L** : Module.finrank (𝒪 E) Λ.L = Λ.n

theorem **isIntegral_form** (x y : Λ.L) : IsIntegral ℤ (Λ.form x y)

@[simp] theorem **coe_hInt** (x y : Λ.L) : algebraMap (𝒪 E) E (Λ.hInt x y) = Λ.form x y

theorem **hInt_unimodular** : Λ.hInt.Unimodular

theorem **stdForm_apply** {n : ℕ} (x y : Fin n → E) : stdForm E n x y = ∑ i, x i ∗ star (y i)

theorem **stdForm_pos** {n : ℕ} (σ : E →+* ℂ) (x : Fin n → E) (hx : x ≠ 0) : 0 < (σ (stdForm E n x x)).re

theorem **mem_stdLat** {n : ℕ} (x : Fin n → E) : x ∈ Formalism.L2.Pj.stdLat E n ↔ ∀ i, ∃ c : 𝒪 E, algebraMap (𝒪 E) E c = x i

theorem **single_mem_stdLat** {n : ℕ} (i : Fin n) : (Pi.single i 1 : Fin n → E) ∈ Formalism.L2.Pj.stdLat E n

theorem **isLattice_stdLat** (n : ℕ) : Submodule.IsLattice E (Formalism.L2.Pj.stdLat E n) where

theorem **unimod_stdLat** (n : ℕ) : ((Formalism.L2.Pj.stdLat E n) : Formalism.L2.Pj.GenLat E n).Unimod

theorem **hInt_std** (n : ℕ) (x y : (std E n).L) : (std E n).hInt x y = ∑ i, stdCoord n i x ∗ star (stdCoord n i y)

theorem **gramE_std** (n : ℕ) : (std E n).gramE = (1 : Matrix (Fin n) (Fin n) E)

theorem **sumOfNorms_std** (n : ℕ) : (std E n).SumOfNorms

theorem **bar_eq_star** (H : Host E) (c : 𝒪 E) : H.bar c = star c

theorem **halfUnit_add_star** (H : Host E) : H.w0 + star H.w0 = 1

theorem **isAdj_sab** (t : SAB H X) : IsAdj t.act t.PP.L

theorem **pp_eq** : (tens t Λ).PP = Λ.thetaOn t.act H.w0 t.PP

theorem **lamPt_eq** {T : Over X} (z : T ⟶ (tens t Λ).G) : lamPt t Λ z = pull ((tens t Λ).G ◁ z).left (cls _ (poi t Λ).isLB)

theorem **lamPt_mul** {T : Over X} (z z' : T ⟶ (tens t Λ).G) : lamPt t Λ (z ∗ z') = lamPt t Λ z + lamPt t Λ z'

theorem **pp_iso_pull** : LBIso (tens t Λ).PP.L ((poi t Λ).L.pull (lift (𝟙 (tens t Λ).G) ((tens t Λ).act.act H.w0)).left)

theorem **lam_pp_iso** : LBIso (abelian₂ (tens t Λ).G (tens t Λ).PP.L) (poi t Λ).L

theorem **lam_pp_tmul** (m n : Λ.L) : LBIso (cubeLB (tens t Λ).G ![fst t.G t.G ≫ (isTensor t Λ).tmul m, snd t.G t.G ≫ (isTensor t Λ).tmul n] (tens t Λ).PP.L) (cubeLB t.G ![fst t.G t.G ≫ t.act.act (Λ.hInt m n), snd t.G t.G] t.PP.L)

theorem **isProper** [IsProper t.G.hom] : IsProper (tens t Λ).G.hom

theorem **abLocus_subset** : abLocus t.G ⊆ abLocus (tens t Λ).G

theorem **smoothOfRelativeDimension** {g : ℕ} (hg : SmoothOfRelativeDimension g t.G.hom) : SmoothOfRelativeDimension (Λ.n ∗ g) (tens t Λ).G.hom

theorem **kTriv** (hK : Out.KTriv t.G t.PP.L) : Out.KTriv (tens t Λ).G (tens t Λ).PP.L

theorem **relAmple** (laws : Out.Laws) [IsLocallyNoetherian X] [IsProper t.G.hom] {g : ℕ} (hg : SmoothOfRelativeDimension g t.G.hom) (hA : Out.RelAmple t.G t.PP.L) (hK : Out.KTriv t.G t.PP.L) (hΛ : Λ.SumOfNorms) : Out.RelAmple (tens t Λ).G (tens t Λ).PP.L

theorem **pp_iso_mapLeft** (Λ' : Unim E) (f : Λ'.L →ₗ[𝒪 E] Λ.L) (hf : Λ'.hInt.IsIsometry Λ.hInt f) : LBIso (tens t Λ').PP.L ((tens t Λ).PP.L.pull ((isTensor t Λ').mapLeft (isTensor t Λ) f).left)

theorem **isLinearHom_stdPr** (n : ℕ) (i : Fin n) : IsLinearHom (tens t (Unim.std E n)).act t.act (stdPr t n i)

theorem **stdIso_pr** (n : ℕ) (i : Fin n) : (stdIso t n).hom ≫ Out.powPr t.G i = stdPr t n i

theorem **pp_std_iso** (n : ℕ) : LBIso (tens t (Unim.std E n)).PP.L (Out.boxLB t.PP.L (stdPow t n).pr)

theorem **isLinearPullback_sabMap** : IsLinearPullback t'.act t.act f m.φ

theorem **tmul_map** (v : Λ.L) : ((isTensor t' Λ).tmul v).left ≫ (map Λ m).φ = m.φ ≫ ((isTensor t Λ).tmul v).left

theorem **amb_w0_eq_boundaryLocalData_w0** (A : Amb D) : A.H.w0 = BoundaryLocalData.w0 D

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.PushInvertibleModelOfIso

theorem **genInv_fst** : genInv Y π ≫ pullback.fst _ _ ≫ π = pullback.fst _ _

theorem **genInv_snd** : genInv Y π ≫ pullback.snd _ _ = pullback.snd _ _

theorem **exists_sabMap_genInv** {t : SAB H X} (m : SABMap t Y.t π) : ∃ n : SABMap Y.oF.t t ((Y.jQ ≫ genInv Y π) ≫ pullback.fst _ _), n.φ ≫ m.φ = Y.m.φ

theorem **modelOfIso_j** (hn : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme X) (t : SAB H X) (m : SABMap t Y.t π) : (modelOfIso Y π hn t m).j ≫ π = Y.j

theorem **pushInvertible_modelOfIso** (hn : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme X) (t : SAB H X) (m : SABMap t Y.t π) (hD : GlobalTheory.Models.IsDedFrac β) (hY : PushInvertible Y.t.G Y.t.PP.L) : PushInvertible t.G t.PP.L

theorem **isFrame_thetaSec** (M : GChart Y) : GlobalTheory.LB.IsFrame (thetaSec M)

theorem **thetaSec_change** (M M' : GChart Y) : (pushMod Y.t).presheaf.map (homOfLE (inf_le_right : M.U ⊓ M'.U ≤ M'.U)).op (thetaSec M') = thRat M M' • (pushMod Y.t).presheaf.map (homOfLE (inf_le_left : M.U ⊓ M'.U ≤ M.U)).op (thetaSec M)

theorem **ofFrames_thetaSec** : GlobalTheory.LB.ofFrames (fun M : GChart Y => isFrame_thetaSec M) = pushPP Y

theorem **nonempty_pushMod_iso** (h : (pushPP Y).IsLB) : Nonempty (pushMod Y.t ≅ toMod (pushPP Y))

theorem **isVB_pushMod** (h : (pushPP Y).IsLB) : GlobalTheory.VB.IsVB (pushMod Y.t) 1

theorem **thRat_pow_mul_keyUnit** (k : ℕ) (M M' : GChart Y) : thRat M M' ^ (2 ∗ k) ∗ (keyUnit k M M').val = (detChg M.toMChart M'.toMChart).val ^ k

theorem **gm_keyUnit** (k : ℕ) (M M' : GChart Y) (y : Y.X) : gm (𝟙 Y.X) y _ (keyUnit k M M') = (gm (𝟙 Y.X) y _ (thUnit M M'))⁻¹ ^ (2 ∗ k) ∗ gm (𝟙 Y.X) y _ (detChg M.toMChart M'.toMChart) ^ k

theorem **isLB_keyLB** (k : ℕ) : (keyLB Y k).IsLB

theorem **range_base_j** : Set.range ct.base.j = ct.Xm.opn

theorem **pushInvertible** : PushInvertible ct.Xm.t.G ct.Xm.t.PP.L

theorem **isLB_keyL** (c : ℕ) : (ct.keyL c).IsLB

theorem **isVB_pushMod** : GlobalTheory.VB.IsVB (pushMod ct.Xm.t) 1

theorem **isInv_piPP** : ct.IsInv ct.piPP

theorem **isInv_keySh** (c : ℕ) : ct.IsInv (ct.keySh c)

theorem **isFrame_keyGen** (c : ℕ) (M : ct.KeyChart) : GlobalTheory.LB.IsFrame (keyGen c M)

theorem **keyCoef_keyMk** (c : ℕ) {V : ct.Xm.X.Opens} (f : ∀ M : ct.KeyChart, Γ(ct.Xm.X, V ⊓ M.U)) (hf : f ∈ secOn (ct.keyL c) V) : keyCoef (keyMk c f hf) = f

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RSideA

theorem **RSideW.toA** {z : Dat (w.st Z) w.I hX T Wt} {Wt' : Set B'.places.Pl} {R' : RealDat3 (w'.st Z').toD7 (famR (w'.st Z') w'.I hX' (usedOf w'.I T' Wt') Xg tr) Ks} (h : RSideW z Wt' R') : RSideA z Wt' R'

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RankOneBaseWeightEqCmTheta

theorem **generic_dim** (A : DatAt Y P) : A.U.A.g = Module.finrank ℚ E / 2 ∗ (Dw.p + Dw.q)

theorem **dim_add_rank** (A : DatAt Y P) : A.U.D.toPre.P.g + Module.finrank ℤ A.Ylat = Module.finrank ℚ E / 2 ∗ (Dw.p + Dw.q)

theorem **finrank_host_eq** (D : D7) : Module.finrank ℚ D.host.E = 2 ∗ D.R.f

theorem **complexConj_eq_conj** (D : D7) (x : D.host.E) : IsCMField.complexConj D.host.E x = D.host.conj x

theorem **rankOne_base_weight_eq_cmTheta** (A : Amb D) : A.H.w0 = w0 D

theorem **smooth_abDim** : SmoothOfRelativeDimension (ct.abDim y) (ct.dat y).U.D.toPre.P.A.hom

theorem **beLin_apply** (a b : ct.Ylat y) : ct.beLin y a b = (ct.dat y).beta a b

theorem **isTauVal_tauVal** (a b : ct.Ylat y) : Per.IsTauVal (ct.dat y).f ((ct.evidence y).e a b) ((ct.tauVal y a (ct.phi y b) : ((ct.fld y).C)ˣ) : (ct.fld y).C)

theorem **isTauVal_mulConst** (a b : ct.Ylat y) : Per.IsTauVal (ct.dat y).f ((ct.evidence y).e a b) ((ct.mulConst y a b : ((ct.fld y).C)ˣ) : (ct.fld y).C)

theorem **mulFst_eq** (a b : ct.Ylat y) : ct.mulFst y a b = ct.phi y b

theorem **mulSnd_eq** (a b : ct.Ylat y) : ct.mulSnd y a b = ct.phi y a

theorem **mulConst_eq** (a b : ct.Ylat y) : ct.mulConst y a b = ct.tauVal y a (ct.phi y b)

theorem **abDim_add** : ct.abDim y + Module.finrank ℤ (ct.Ylat y) = D.R.f * N

theorem **motTrue_mul** (a b : ct.Ylat y) : (motTrue ct).multiplicationLatticeFirst y a b = ct.phi y b ∧ (motTrue ct).multiplicationLatticeSecond y a b = ct.phi y a ∧ (motTrue ct).mulC y a b = (motTrue ct).tau y a (ct.phi y b)

theorem **motTrue_psiV_eq_be** (a : ct.Ylat y) : (motTrue ct).psiV y a = (motTrue ct).be y a (w0 D • a)

theorem **motTrue_psiV_half** (a : ct.Ylat y) : (motTrue ct).psiV y a = (motTrue ct).be y a a / 2

theorem **motTrue_psiV_eq_vPsiAt** (a : ct.Ylat y) : (motTrue ct).psiV y a = vPsiAt (ct.dat y).U.D.toPre (DatAtCheck.fC (ct.dat y)) p a

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RationalRootsDef

theorem **muIsW_of_muIs** {X : Stages} {f : Fr X.B.toPre} {T : TwDat X.B} {xi : X.C.Tup → HChar X.B.host.E} (h : MuIs X f T xi) : MuIsW X f T {w | X.B.places.resChar w ∈ T.pl} xi

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RealIoX

theorem **real_ioY** {A : FamIn a.D.what₂.LL G Pl0 T a.D.R.m} (R : RealDat a.D A Ks) : @HEq (IoYT (@MultT a.D) R) ((R.ioY)) _ (@RealDat.ioY b.D G _ _ Pl0 T (famM b A) Ks _ _ (alg b iK) (real b R))

theorem **real_ioX** {A : FamIn a.D.what₂.LL G Pl0 T a.D.R.m} (R : RealDat a.D A Ks) : @HEq (IoXT (@MultT a.D) R) ((R.ioX)) _ (@RealDat.ioX b.D G _ _ Pl0 T (famM b A) Ks _ _ (alg b iK) (real b R))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RealMoveOfParts

theorem **realMove_of_parts** {X : RealSide D} {X' : RealSide D'} {ι : D.what₂.LL →+* D'.what₂.LL} {eK : X.core.Q.c.F ≃+* X'.core.Q.c.F} {γ : X.Γ ≃* X'.Γ} (a : Frames X X') (b : Chars X X') (c : Grps X X' ι γ) (q : PlCl X X') (o : IoCl X X' ι) (s : FormsCl X X' ι eK γ) : Nonempty (RealMove X X' ι)

theorem **finite_tup** (P : Pre2 K w) : Finite (TupU w.I (usedOf w.I T P.Wt))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RecExGOfRecAtG

theorem **exists_du_nat** (eps : ℝ) (h0 : 0 < eps) (h1 : eps < 1 / 52) : ∃ x0 : ℝ, ∀ (s : Scale) (C : FC) (cD ch : ℝ) (P : PreCh) (q1 : ℕ) (a b : P.R.A → ℤ) (c : P.Word → Fin (P.R.k - 2) → Bool), s.eps = eps → x0 ≤ s.x → ChOK s C cD ch P q1 a b c → ∃ du : P.Word → ℕ, (∀ u, 0 < du u) ∧ ∀ u, (du u : ℝ) = duP P u

theorem **not_evG_false** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) : ¬ EvG fun _ => False

theorem **exists_of_evG** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) {p : Ctx → Prop} (hp : EvG p) : ∃ (K : Ctx) (C5 c6 Cb : ℝ), K.Good C5 c6 Cb ∧ p K

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RedivisionStructureEq

theorem **reD_ofDiv** (D : D7) (N : ∀ u, Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) (h : D.what₂.Word → ℝ) : reD D (New.ofDiv D N h) = GlobalTheory.EX.OB.setDiv D N h

theorem **reD_rows** : (reD D Z).what₂ = D.what₂

theorem **reD_host** : (reD D Z).host = D.host

theorem **redivision_structure_eq** : (reD D Z).s = mix D.s Z.Z.s

theorem **reD_atP** : (reD D Z).s.atP = D.s.atP

theorem **reD_HH** : (reD D Z).s.HH = Z.Z.s.HH

theorem **reD_h** : (reD D Z).div.h = Z.h

theorem **reD_L** (u : D.what₂.Word) : ((reD D Z).div.Nu u).L = Z.N u

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ReferenceCharactersFactorTame

theorem **isFr_latFr** (P : HeightOneSpectrum (𝒪 E)) {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) : IsFr P N (latFr P N)

theorem **isLiftFr_ptFr** (p : DecoF E n ST SG Q) (w : SG) : IsLiftFr p.deco.1 w (ptFr p w.1)

theorem **ptFr_of_mem** (p : DecoF E n ST SG Q) {P : HeightOneSpectrum (𝒪 E)} (h : P ∉ SG) (h' : P ∈ Q) : ptFr p P = p.fr (P, h')

theorem **isFr_ptFr** (p : DecoF E n ST SG Q) (P : HeightOneSpectrum (𝒪 E)) : IsFr P p.deco.1.N (ptFr p P)

theorem **ptFr_congr** {ST' : Finset (HeightOneSpectrum (𝒪 E))} {p : DecoF E n ST SG Q} {p' : DecoF E n ST' SG Q} (hN : p.deco.1.N = p'.deco.1.N) (hbe : p.deco.1.be = p'.deco.1.be) (hfr : p.fr = p'.fr) : ptFr p = ptFr p'

theorem **frsF_eq_ptFr** {X : Fields} {Pr : Primes X} {R : Ranks} (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S : Set ℕ} (mu : Tup X.F X.E X.K S) {u : Pr.Word} {j : ℕ} (p : ChPt (R

theorem **extTo_deco** (SQ : Finset (HeightOneSpectrum (𝒪 E))) (p : DecoF E n ST SG Q) : (extTo SQ p).deco = p.deco

theorem **extTo_fr** (SQ : Finset (HeightOneSpectrum (𝒪 E))) (p : DecoF E n ST SG Q) (P : SQ) : (extTo SQ p).fr P = ptFr p P.1

theorem **extTo_self** (hd : Disjoint SG Q) (p : DecoF E n ST SG Q) : extTo Q p = p

theorem **finite_ordSub** {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) : {w | ordSub w (detSub N) ≠ 0}.Finite

theorem **finite_detIdl** (SB : Finset (HeightOneSpectrum (𝒪 E))) {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) : Function.HasFiniteMulSupport fun w : HeightOneSpectrum (𝒪 E) => if w ∈ SB ∨ conjPrime₂ w ∈ SB then 1 else localIdele E w (atFin w (unif w)) ^ (ordSub w (detSub N)).toNat

theorem **compAt_detIdl** (SB : Finset (HeightOneSpectrum (𝒪 E))) {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) (fr : HeightOneSpectrum (𝒪 E) → Sp E n ≃ₗ[E] Sp E n) (w : HeightOneSpectrum (𝒪 E)) : compAt w (detIdl SB N fr) = if w ∈ SB then algebraMap E

_ ((LinearEquiv.det (fr w) : Eˣ) : E) else if conjPrime₂ w ∈ SB then 1 else algebraMap E _ ((unif w : Eˣ) : E) ^ (ordSub w (detSub N)).toNat

theorem **finPartI_detIdl** (SB : Finset (HeightOneSpectrum (𝒪 E))) {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) (fr : HeightOneSpectrum (𝒪 E) → Sp E n ≃ₗ[E] Sp E n) : finPartI E (detIdl SB N fr) = detIdl SB N fr

theorem **ordSub_detSub_eq** {P : HeightOneSpectrum (𝒪 E)} {N : Submodule (𝒪 E) (Sp E n)} {f : Sp E n ≃ₗ[E] Sp E n} (hN : Submodule.IsLattice E N) (hf : IsFr P N f) : ordSub P (detSub N) = ordF P (LinearEquiv.det f)

theorem **val_eq_one_of_sub** (h : v (x - 1) < 1) : v x = 1

theorem **val_mul_sub_one_lt** (hx : v (x - 1) < 1) (hy : v (y - 1) < 1) : v (x * y - 1) < 1

theorem **val_inv_sub_one_lt** (hx : v (x - 1) < 1) : v (x⁻¹ - 1) < 1

theorem **adAt_apply** (P : HeightOneSpectrum (𝒪 E)) (z : AfE F E) : adAt F P z = afIso F E z P

theorem **adAt_inclE** (P : HeightOneSpectrum (𝒪 E)) (x : E) : adAt F P (Formalism.UG.inclE (F

theorem **adAt_finAf** (P : HeightOneSpectrum (𝒪 E)) (a : Ideles E) : adAt F P (finAf F E a : AfE F E) = compAt P a

theorem **hasRes_coe** (P : HeightOneSpectrum (𝒪 E)) (r : 𝒪 E) : HasRes P (algebraMap E _ (algebraMap (𝒪 E) E r)) (Ideal.Quotient.mk P.asIdeal r)

theorem **valued_algebraMap** (P : HeightOneSpectrum (𝒪 E)) (x : E) : Valued.v (algebraMap E (P.adicCompletion E) x) = P.valuation E x

theorem **HasRes.unique** {P : HeightOneSpectrum (𝒪 E)} {z : P.adicCompletion E} {c c' : 𝒪 E ⧸ P.asIdeal} (h : HasRes P z c) (h' : HasRes P z c') : c = c'

theorem **IsPtOf.of_extTo** {SQ : Finset (HeightOneSpectrum (𝒪 E))} (hQ : Q ⊆ SQ) (hd : Disjoint SG Q) {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g (extTo SQ p)) : IsPtOf hc g p

theorem **IsPtOf.extTo** {SQ : Finset (HeightOneSpectrum (𝒪 E))} (hQ : SQ ⊆ Q) (hd : Disjoint SG SQ) {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) : IsPtOf hc g (extTo SQ p)

theorem **isPtOf_base** [NeZero n] : IsPtOf hc 1 (DecoF.base E n ST SG Q)

theorem **conjPl_star_iff** (w w' : HeightOneSpectrum (𝒪 E)) : ConjPl (starRingAut : E ≃+* E) w w' ↔ w' = conjPrime₂ w

theorem **valuation_star** (P : HeightOneSpectrum (𝒪 E)) (y : E) : P.valuation E (star y) = (conjPrime₂ P).valuation E y

theorem **sigR_mem_rInt** {r : AfE F E} (hr : r ∈ RInt F E) : Formalism.UG.sigR hc.sig _ r ∈ RInt F E

theorem **exists_approx_small** (z : AfE F E) (T : Finset (HeightOneSpectrum (𝒪 E))) {γ : WithZero (Multiplicative ℤ)} (hγ : γ ≠ 0) : ∃ (y : E) (N : ℕ) (r : AfE F E), r ∈ RInt F E ∧ (∀ P ∈ T, Valued.v (N : P.adicCompletion E) < γ) ∧ z = Formalism.UG.inclE (F

theorem **valued_nat_mul_lt** {P : HeightOneSpectrum (𝒪 E)} {N : ℕ} {r : AfE F E} (hr : r ∈ RInt F E) {γ : WithZero (Multiplicative ℤ)} (h : Valued.v (N : P.adicCompletion E) < γ) : Valued.v (adAt F P (N * r)) < γ

theorem **sigR_inclE** (y : E) : Formalism.UG.sigR hc.sig _ (Formalism.UG.inclE (F

theorem **valued_sigR** (P : HeightOneSpectrum (𝒪 E)) (z : AfE F E) : Valued.v (adAt F P (Formalism.UG.sigR hc.sig _ z)) = Valued.v (adAt F (conjPrime₂ P) z)

theorem **exists_real_approx** {P : HeightOneSpectrum (𝒪 E)} (hP : conjPrime₂ P ≠ P) (u : (P.adicCompletion E)ˣ) : ∃ k : Fˣ, Valued.v ((u : P.adicCompletion E) - algebraMap E _ (algebraMap F E k)) < Valued.v (u : P.adicCompletion E)

theorem **isConjPl_iff** (w w' : HeightOneSpectrum (𝒪 H.E)) : IsConjPl H w w' ↔ w' = conjPrime₂ w

theorem **under_eq_iff** (w w' : HeightOneSpectrum (𝒪 H.E)) : w'.under (𝒪 H.F) = w.under (𝒪 H.F) ↔ w' = w ∨ w' = conjPrime₂ w

theorem **mem_KDq_of_local** (a : Ideles H.E) (h : ∀ v : HeightOneSpectrum (𝒪 H.F), ∃ k : (H.F)ˣ, ∀ w : HeightOneSpectrum (𝒪 H.E), w.under (𝒪 H.F) = v → Valued.v (compAt w a / algebraMap H.E _ (algebraMap H.F H.E k)) = 1 ∧ ((∃ i, OverP H (H.g i) w) ∨ AtQ H SQ w → Valued.v (compAt w a / algebraMap H.E _ (algebraMap H.F H.E k) - 1) < 1)) : a ∈ KDq H SQ

theorem **LevOk.level** (h : LevOk H SG SQ) {w : HeightOneSpectrum (𝒪 H.E)} (hw : (∃ i, OverP H (H.g i) w) ∨ AtQ H SQ w) : w ∈ uSG SG SQ ∨ conjPrime₂ w ∈ uSG SG SQ

theorem **LevOk.mono** (h : LevOk H SG SQ) {Q : Finset (HeightOneSpectrum (𝒪 H.E))} (hQ : Q ⊆ SQ) : LevOk H SG Q

theorem **mem_KDq_of_conj** (h : LevOk H SG SQ) (b : Ideles H.E) (h1 : ∀ w, Valued.v (compAt w b) = Valued.v (compAt (conjPrime₂ w) b)) (h2 : ∀ P ∈ uSG SG SQ, Valued.v (compAt P b / adAt H.F P (Formalism.UG.sigR (host_cmData R H).sig _ (finAf H.F H.E b : AfE H.F H.E)) - 1) < 1) : b ∈ KDq H SQ

@[simp] theorem **xiDet_one** (p : DecoF H.E n ST SG Q) : xiDet H SQ 1 p = 1

theorem **rhoAt_one** (i : ℕ) (p : DecoF H.E n ST SG Q) : rhoAt SQ C 1 i p = (muTilQ H SQ C.mu0 (i + 1) (detPt H SQ p))⁻¹ * tildeQ H SQ C.chiPsi (detPt H SQ p) ^ (-((R.N (i + 1) / 2 : ℕ) : ℤ))

theorem **rhoAt'_one** (i : ℕ) (p : DecoF H.E n ST SG Q) : rhoAt' SQ C 1 i p = muTilQ H SQ C.mu0 (i + 1) (detPt H SQ p) * tildeQ H SQ C.chiPsi (detPt H SQ p) ^ ((R.N i / 2 : ℕ) : ℤ)

theorem **rhoAt_eq** (ξ : HChar H.E) (i : ℕ) (p : DecoF H.E n ST SG Q) : rhoAt SQ C ξ i p = ((muTilQ H SQ C.mu0 (i + 1) * tildeQ H SQ ξ)⁻¹ * tildeQ H SQ C.chiPsi ^ (-((R.N (i + 1) / 2 : ℕ) : ℤ))) (detPt H SQ p)

theorem **rhoAt'_eq** (ξ : HChar H.E) (i : ℕ) (p : DecoF H.E n ST SG Q) : rhoAt' SQ C ξ i p = (muTilQ H SQ C.mu0 (i + 1) * tildeQ H SQ ξ * tildeQ H SQ C.chiPsi ^ ((R.N i / 2 : ℕ) : ℤ)) (detPt H SQ p)

theorem **norm_tilde_character_eq_one** (j : ℕ) (x : DtQ H SQ) : ‖((muTilQ H SQ C.mu0 j x : ℂˣ) : ℂ)‖ = 1

theorem **norm_xiDet** {ξ : HChar H.E} (hu : ξ.IsUnitary) (p : DecoF H.E n ST SG Q) : ‖((xiDet H SQ ξ p : ℂˣ) : ℂ)‖ = 1

theorem **norm_rhoAt** {ξ : HChar H.E} (hu : ξ.IsUnitary) (i : ℕ) (p : DecoF H.E n ST SG Q) : ‖((rhoAt SQ C ξ i p : ℂˣ) : ℂ)‖ = 1

theorem **norm_rhoAt'** {ξ : HChar H.E} (hu : ξ.IsUnitary) (i : ℕ) (p : DecoF H.E n ST SG Q) : ‖((rhoAt' SQ C ξ i p : ℂˣ) : ℂ)‖ = 1

theorem **KDq_le_ker_of_tame** {χ : HChar H.E} (hT : χ.TrivOn H.F) (hta : ∀ v, χ.TameAt v) (hu : ∀ v, ¬ ((∃ k, OverP H (H.g k) v) ∨ AtQ H SQ v) → χ.UnrAt v) : KDq H SQ ≤ (χ.1.comp (finPartI H.E)).ker

theorem **reference_characters_factor_tame** (j : Fin R.m) : KDq H SQ ≤ ((C.mu0 j).1.comp (finPartI H.E)).ker

theorem **chiPsi_factors** : KDq H SQ ≤ (C.chiPsi.1.comp (finPartI H.E)).ker

theorem **xiDet_eq** {ξ : HChar H.E} (h : KDq H SQ ≤ (ξ.1.comp (finPartI H.E)).ker) (p : DecoF H.E n ST SG Q) : xiDet H SQ ξ p = ξ.1 (ptIdl H SQ p)

theorem **rhoAt_eq_hecke** {ξ : HChar H.E} (h : KDq H SQ ≤ (ξ.1.comp (finPartI H.E)).ker) (j : Fin R.m) (p : DecoF H.E n ST SG Q) : rhoAt SQ C ξ j.1 p = ((C.mu0 j * ξ).1 (ptIdl H SQ p))⁻¹ * C.chiPsi.1 (ptIdl H SQ p) ^ (-((R.N (j.1 + 1) / 2 : ℕ) : ℤ))

theorem **rhoAt'_eq_hecke** {ξ : HChar H.E} (h : KDq H SQ ≤ (ξ.1.comp (finPartI H.E)).ker) (j : Fin R.m) (p : DecoF H.E n ST SG Q) : rhoAt' SQ C ξ j.1 p = (C.mu0 j * ξ).1 (ptIdl H SQ p) * C.chiPsi.1 (ptIdl H SQ p) ^ ((R.N j.1 / 2 : ℕ) : ℤ)

theorem **toDtQ_mk** {SQ : Finset (HeightOneSpectrum (𝒪 H.E))} (h : Q ⊆ SQ) (a : Ideles H.E) : toDtQ H h (QuotientGroup.mk a) = QuotientGroup.mk a

theorem **tildeQ_toDtQ** {SQ : Finset (HeightOneSpectrum (𝒪 H.E))} (h : Q ⊆ SQ) {χ : HChar H.E} (hχ : KDq H Q ≤ (χ.1.comp (finPartI H.E)).ker) (x : DtQ H SQ) : tildeQ H Q χ (toDtQ H h x) = tildeQ H SQ χ x

theorem **levOk_base** (B : Base) (X : PreX B.toPre) : LevOk B.host (B.primes X).vGs ∅

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RefilledStagesRecord

theorem **reS_self** (D : D7) : reS D (SNew.self D) = D

theorem **refilledStages_record** (X : Stages) (Z : SNew X.toD7) : (reStages X Z).toD7 = reS X.toD7 Z

theorem **reS_atP** : (reS D Z).s.atP = D.s.atP

theorem **reS_Del** : (reS D Z).s.Del = D.s.Del

theorem **reS_letter** : (reS D Z).s.letter = D.s.letter

theorem **reS_cdScale** : (reS D Z).s.cdScale = D.s.cdScale

theorem **reS_dsig** : (reS D Z).dsig = D.dsig

theorem **reS_rows** : (reS D Z).what₂ = D.what₂

theorem **reS_HH** : (reS D Z).s.HH = Z.s.HH

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RelabelledWeight

theorem **hL** {X : HidaSetting a.D} (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf a.D) : (XL a Z X).B.I = Formalism.L2.SG.InstExistsPf.instOf (reL a Z).D

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ReplacementValuesBaseLevelTypePartsEq

theorem **finTypRep_trans_cast** {E : Type} [Field E] [NumberField E] [IsCMField E] {n : ℕ} {S S' : Set (Formalism.L2.NG.FinTyp E n)} (e : S = S') {α : Type} (f : α ≃ Formalism.L2.Col.FinTypCls S) (x : α) : Formalism.L2.Col.finTypRep S (f x) = Formalism.L2.Col.finTypRep S' (f.trans (Equiv.cast (congrArg Formalism.L2.Col.FinTypCls e)) x)

theorem **occ_eq** : occA Y = occB c Y

theorem **rep_eq** (π : a.D.what₂.FT) : principalLevelFrame.rowRepR R.θ (Y.eFT π) = Formalism.L2.Col.finTypRep (occB c Y) (eFTB c Y π)

theorem **prtE_tr** (π : a.D.what₂.FT) : Y.Ish.prt π = (isotLB c.to a R.θ Y.PY (R.MY 0) (levS _ (R.βY 0) (R.dat0 (R.levR F true))) (levChi _ (R.βY 0) (R.dat0 (R.levR F true))) R.e0 (principalLevelFrame.FinTyp.conjT (Formalism.L2.Col.finTypRep (occB c Y) (eFTB c Y π)))).map (PiTensorProduct.lift : MMap (R.MY 0) ≃ₗ[ℂ] _).toLinearMap

theorem **replacementValues_baseLevelTypeParts_eq** (π : a.D.what₂.FT) : Y.Iz.prt π = (isotLB c.to a R.θ Y.PY (R.MY 0) (levS _ (R.βY 0) (R.dat0 (R.levR F false))) (levChi _ (R.βY 0) (R.dat0 (R.levR F false))) R.e0 (principalLevelFrame.FinTyp.conjT (Formalism.L2.Col.finTypRep (occB c Y) (eFTB c Y π)))).map (PiTensorProduct.lift : MMap (R.MY 0) ≃ₗ[ℂ] _).toLinearMap

theorem **prtWE_tr** (i : Fin a.D.R.m) (π : a.D.what₂.FT) : Y.IL.prtW i π = (isotLB c.to a R.θ (Y.PW i) (R.MW i (sg0 a.D A.A)) (levS _ (R.βW i) (R.datWp i (R.levL F false (sg0 a.D A.A)))) (levChi _ (R.βW i) (R.datWp i (R.levL F false (sg0 a.D A.A)))) (R.eWf i (sg0 a.D A.A)) (Formalism.L2.Col.finTypRep (occB c Y) (eFTB c Y π))).map (PiTensorProduct.lift : MMap (R.MW i (sg0 a.D A.A)) ≃ₗ[ℂ] _).toLinearMap

theorem **isomFin_tr** (i : Fin a.D.R.m) : @principalLevelFrame.IsIsomFin c.to.D (Yc c.to.D 0) (Wc c.to.D ((i))) ((Y.ιW i))

theorem **jEEu** : ∀ y, (Y.Ish.jE (a.D.jEE (eRRu Z u y))).1 = PiTensorProduct.lift (R.HE.e (Tw.out (s

theorem **jEE0u** : ∀ y, (Y.Iz.jE (X0.s.jEE (riemannRochSpaceCoordinateEquiv Z u y))).1 = PiTensorProduct.lift (R.HE.e (Tw.out (s

theorem **metu** : ∀ v, v ≠ A.A.v0 → ∀ x y, Formalism.HermitianLatticesTheory.majAt a.D.div.RR.b a.D.div.RR.X v (eRRu Z u x) (eRRu Z u y) = Formalism.HermitianLatticesTheory.majAt (Pi.basisFun a.D.what₂.LL _) ((cxW R F u).data F.cr true).XE v x y

theorem **met0u** : ∀ v, v ≠ A.A.v0 → ∀ x y, Formalism.HermitianLatticesTheory.majAt X0.s.RR.b X0.s.RR.X v (riemannRochSpaceCoordinateEquiv Z u x) (riemannRochSpaceCoordinateEquiv Z u y) = Formalism.HermitianLatticesTheory.majAt (Pi.basisFun a.D.what₂.LL _) ((cxW R F u).data F.cr false).XE v x y

theorem **etaJu** : ∀ (i : Fin a.D.R.m) w, a.D.what₂.etaJ ((i : ℕ) + 1) w = if a.D.plEq w ∈ R.twPl then A.A.fam.etaJ F.P0 F.ψ0 (A.fam3.toSub2 (cxW R F u).zs) i (R.mu (a.D.plEq w)) (a.D.plEq w) else 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RestrictedCharacterDataValid

theorem **isInert** (h : R.IsInert) : (dat3 f μ R mu tw).IsInert

theorem **pThmB** (h : R.PThmB) : (dat3 f μ R mu tw).PThmB

theorem **isScal** (h : R.IsScal (bare3 _ P0 ψ0 P1 ψ1)) (hB : R.PThmB) : (dat3 f μ R mu tw).IsScal (bare3 _ P0 ψ0 P1 ψ1) where

theorem **isTup** (hf : Function.Injective f) (h : R.IsTup) (n : NewTup R mu tw) : (dat3 f μ R mu tw).IsTup where

theorem **restricted_character_data_valid** (hf : Function.Injective f) (h : R.WeakCharacterisation (bare3 _ P0 ψ0 P1 ψ1)) (e : R.toRealDat.ExAt) (hB : R.PThmB) (n : NewTup R mu tw) : (dat3 f μ R mu tw).WeakCharacterisation (bare3 _ P0 ψ0 P1 ψ1) where

theorem **keep** {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (h : Keep (A3

theorem **pins** (hf : Function.Injective f) {cG : G0 R.Pg D.R.r ≃* G0g D.nd} (h : Pins R (bare3 _ P0 ψ0 P1 ψ1) cG) (e : R.toRealDat.ExAt) (hB : R.PThmB) (n : NewTup R mu tw) : Pins (dat3 f μ R mu tw) (bare3 _ P0 ψ0 P1 ψ1) cG where

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RestrictionPoincareBundle

theorem **smallCls_eq_zero_iff** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) : smallCls L hL = 0 ↔ ∃ t : LB.Gen L, t.IsGen

theorem **smallPull_comp** {X Y Z : Scheme.{0}} (g : Z ⟶ Y) (f : Y ⟶ X) (c : SmallPic X) : smallPull (g ≫ f) c = smallPull g (smallPull f c)

theorem **isRigCls_iff** {L : LB G.left} (hL : L.IsLB) : IsRigCls G (smallCls L hL) ↔ Out.Rigidified G L

theorem **isRigCls_diagCls** (w : G ⟶ G) [IsMonHom w] (P : RigPic (G ⊗ G)) : IsRigCls G (diagCls w P)

@[reassoc] theorem **sqMap_fst** : sqMap h ≫ (fst G G).left = (fst G' G').left ≫ φ

@[reassoc] theorem **sqMap_snd** : sqMap h ≫ (snd G G).left = (snd G' G').left ≫ φ

theorem **sq_hom_ext** {T : Scheme.{0}} {a b : T ⟶ (G ⊗ G).left} (h₁ : a ≫ (fst G G).left = b ≫ (fst G G).left) (h₂ : a ≫ (snd G G).left = b ≫ (snd G G).left) : a = b

theorem **eSec_sqMap** : eSec (G' ⊗ G') ≫ sqMap h = f ≫ eSec (G ⊗ G)

theorem **smallPull_diagCls** {w : G ⟶ G} {w' : G' ⟶ G'} (hw : w'.left ≫ φ = φ ≫ w.left) (P : RigPic (G ⊗ G)) : smallPull φ (diagCls w P) = diagCls w' (sqPull h P)

theorem **diag_poiCls** : diag t Λ (poiCls t Λ) = ppCls t Λ

theorem **smallPull_diag** (P : RigPic ((tens t Λ).G ⊗ (tens t Λ).G)) : smallPull (map Λ m).φ (diag t Λ P) = diag t' Λ (sqPull (map Λ m).cart P)

theorem **mapRight_comm** (t : IsTensor A b G a M) (t' : IsTensor A' b' G' a' M) (t₁ : IsTensor A₁ b₁ G₁ a₁ M) (t₁' : IsTensor A₁' b₁' G₁' a₁' M) {f : S' ⟶ S} {φ : G₁.left ⟶ G.left} {φ' : G₁'.left ⟶ G'.left} {ψ : A₁.left ⟶ A.left} {ψ' : A₁'.left ⟶ A'.left} (hψ : IsGrpPullback A₁ A f ψ) (hψ' : IsGrpPullback A₁' A' f ψ') (hm : ∀ m, (t₁.tmul m).left ≫ ψ = φ ≫ (t.tmul m).left) (hm' : ∀ m, (t₁'.tmul m).left ≫ ψ' = φ' ≫ (t'.tmul m).left) {q : G ⟶ G'} (hq : IsLinearHom a a' q) {q₁ : G₁ ⟶ G₁'} (hq₁ : IsLinearHom a₁ a₁' q₁) (e : q₁.left ≫ φ' = φ ≫ q.left) : (t₁.mapRight t₁' hq₁).left ≫ ψ' = ψ ≫ (t.mapRight t' hq).left

@[reassoc] theorem **thickMap_fst** (G : Over (Sp R)) : thickMap R I e G ≫ pullback.fst G.hom (thick R I (e + 1)) = pullback.fst G.hom (thick R I e)

@[reassoc] theorem **thickMap_snd** (G : Over (Sp R)) : thickMap R I e G ≫ pullback.snd G.hom (thick R I (e + 1)) = pullback.snd G.hom (thick R I e) ≫ thickTrans R I e

theorem **isGrpPullback_thickMap** (G : Over (Sp R)) [GrpObj G] : IsGrpPullback (thickObj R I e G) (thickObj R I (e + 1) G) (thickTrans R I e) (thickMap R I e G)

theorem **map_thickMap** {G G' : Over (Sp R)} (u : G ⟶ G') : ((Over.pullback (thick R I e)).map u).left ≫ thickMap R I e G' = thickMap R I e G ≫ ((Over.pullback (thick R I (e + 1))).map u).left

theorem **isLinearHom_q** (e : ℕ) : IsLinearHom (t.act.pullback (thick R I e) _) ((abPartAct U).pullback (thick R I e) _) (U.U.gt e).q

theorem **isLinearHom_qTens** (e : ℕ) : IsLinearHom ((tens t Λ).act.pullback (thick R I e) _) ((Λ.act (abPartAct U)).pullback (thick R I e) _) (qTens U Λ e)

theorem **q_thickMap** (e : ℕ) : (U.U.gt e).q.left ≫ thickMap R I e _ = thickMap R I e _ ≫ (U.U.gt (e + 1)).q.left

theorem **qTens_thickMap** (e : ℕ) : (qTens U Λ e).left ≫ thickMap R I e _ = thickMap R I e _ ≫ (qTens U Λ (e + 1)).left

theorem **isGrpPullback_levToDat** (G : Over (Sp A.R)) [GrpObj G] (e : ℕ) : IsGrpPullback (thickObj _ (defIdeal A) e ((Over.pullback (spMap A.f)).obj G)) (thickObj A.R (maximalIdeal A.R) e G) (sE A.f le_rfl e) (levToDat A G e)

@[reassoc] theorem **levToDat_fst** (G : Over (Sp A.R)) (e : ℕ) : levToDat A G e ≫ pullback.fst _ _ = pullback.fst _ _ ≫ pullback.fst G.hom (spMap A.f)

@[reassoc] theorem **prLev_levToDat** (e : ℕ) : (prLev A Λ e).left ≫ levToDat A _ e = levToDat A _ e ≫ (qTens A.U Λ e).left

theorem **prPt_fst** (e : ℕ) : prPt A Λ e ≫ pullback.fst _ _ = levToDat A _ e ≫ (qTens A.U Λ e).left ≫ pullback.fst _ _

theorem **prPt_hom** (e : ℕ) : prPt A Λ e ≫ (bPt A Λ).hom = pullback.fst _ _ ≫ (gPt A Λ).hom

theorem **gtTrans_prPt** (e : ℕ) : gtTrans A Λ e ≫ prPt A Λ (e + 1) = prPt A Λ e

theorem **isGrpPullback_gPt_base** : IsGrpPullback (gPt A Λ) (tens Y.t Λ).G (z ≫ π) (pullback.fst _ _ ≫ (tens.map Λ A.m).φ)

theorem **gPtTo_φ** : gPtTo A Λ m z hz ≫ (tens.map Λ m).φ = pullback.fst _ _ ≫ (tens.map Λ A.m).φ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RightTranslationActionApply

theorem **rhoC_apply** (γ : UF X.E n) : rhoC σ lam γ = rhoP lam (embGL (embAt σ X.emb) (ufGL X.E n γ))

theorem **rhoC_map** (γ : UF X.E n) (v : VV X.K lam) : rhoC σ lam γ (MvPolynomial.map σ v.1) = MvPolynomial.map σ (rhoV X.emb lam (ufGL X.E n γ) v).1

theorem **mem_cForms** (Λ : LevFam X.E X.K n SG) (F : DecoSet X.E n ST SG → Pol ℂ X.Tau n) : F ∈ cForms ST σ lam Λ ↔ ∀ d, F d ∈ VV ℂ lam Λ ∀ (γ : UF X.E n) (h : Λ.grp), F ((γ, (h.1 : GLG X.E n SG)) • d) = σ (Λ.chr h : X.Kˣ) • rhoC σ lam γ (F d)

theorem **toCForm_injective** (L : DecoSet X.E n ST SG → Submodule (𝒪 X.E) (Sp X.E n)) (Λ : LevFam X.E X.K n SG) : Function.Injective (toCForm σ lam L Λ)

theorem **tr_mem_cForms** (Λ : LevFam X.E X.K n SG) (a : GLG X.E n SG) (ha : ∀ w, (Λ w).NormBy (a w)) {F : DecoSet X.E n ST SG → Pol ℂ X.Tau n} (hF : F ∈ cForms ST σ lam Λ) : (fun d => F (((1 : UF X.E n), a⁻¹) • d)) ∈ cForms ST σ lam Λ

theorem **mem_cForms_split** {r : ℕ} {SG0 : Finset (HeightOneSpectrum (𝒪 X.E))} {ι : GLG X.E r SG0 →* GLG X.E n SG} {ΛP ΛK : LevFam X.E X.K n SG} {Λ0 : LevFam X.E X.K r SG0} (H : LevFam.Split ι ΛP ΛK Λ0) (F : DecoSet X.E n ST SG → Pol ℂ X.Tau n) : F ∈ cForms ST σ lam ΛP ↔ F ∈ cForms ST σ lam ΛK ∧ ∀ (a : Λ0.grp) d, F (((1 : UF X.E n), (ι a.1)⁻¹) • d) = σ ((Λ0.chr a)⁻¹ : X.Kˣ) • F d

theorem **isSubLev_top** (Λ : LevFam X.E X.K n SG) (θ : LevQFam X.E X.K n Q) : IsSubLev Λ Λ θ ⊤

theorem **IsSubLev.rat** {Λ Λ' : LevFam X.E X.K n SG} {θ : LevQFam X.E X.K n Q} {H : Subgroup (FrmG Λ θ)} (h : IsSubLev Λ Λ' θ H) (γ : UF X.E n) : ((γ, 1) : FrmG Λ θ) ∈ H

theorem **frLaw_comp_iff** (F : DecoSet X.E n ST (uSG SG Q) → Pol ℂ X.Tau n) : FrLaw σ lam Λ (levRedFam θ) H (fun p => F (forgetFr p)) ↔ F ∈ cForms ST σ lam (Λ'.glue θ).inv

theorem **FrLaw.eq_of_forgetFr_eq** {F : DecoF X.E n ST SG Q → Pol ℂ X.Tau n} (h : FrLaw σ lam Λ (levRedFam θ) H F) {p q : DecoF X.E n ST SG Q} (e : forgetFr p = forgetFr q) : F p = F q

theorem **FrLaw.exists_form** {F : DecoF X.E n ST SG Q → Pol ℂ X.Tau n} (h : FrLaw σ lam Λ (levRedFam θ) H F) : ∃ G ∈ cForms ST σ lam (Λ'.glue θ).inv, ∀ p, G (forgetFr p) = F p

theorem **lamT_eq** (u : Pr.Word) (j : ℕ) : lamT (W.hc u) j = W.lamTrue u j

theorem **mem_MtP_iff** (F : RowPt Pr R sel mu → Pol ℂ X.Tau (R.N R.m − 1)) : F ∈ MtP W Ch sel mu P σ ↔ ∃ h : F ∈ MtC W Ch sel mu σ, ∀ a : P.grp, rightTranslationAction W Ch sel mu σ a.1 ⟨F, h⟩ = σ (P.chr a : X.Kˣ) • (⟨F, h⟩ : MtC W Ch sel mu σ)

theorem **MtP_le_MtC** : MtP W Ch sel mu P σ ≤ MtC W Ch sel mu σ

theorem **toMtilde_apply** (φ : MtildeV W Ch sel mu P) (d : RowPt Pr R sel mu) : (toMtP W Ch sel mu P σ φ).1 d = MvPolynomial.map σ (φ.1 d).1

theorem **toMtilde_injective** : Function.Injective (toMtP W Ch sel mu P σ)

theorem **isSubLev_rowKH** : IsSubLev (rowTyG Ch P) (rowTyK Ch) (levQX R sel mu) (rowKH Ch sel mu P)

theorem **ysAt_m** : ysAt R sel mu cst R.m = Ys.triv X _

theorem **topPt_val** (u : Pr.Word) (p : PtY (Pr.vT u) Pr.vGs (QE sel (mu : Chr X.F X.E X.K)) (ysAt R sel mu cst R.m)) : topPt sel mu cst u p.1.1 = p

theorem **mem_perAtH_Mt** (F : ∀ b, (stepX Ch sel mu P).D b → Pol ℂ X.Tau (R.N R.m − 1)) : F ∈ (perAtH W Ch sel mu cst L P u rho rho' cD Th H).Mt X.emb σ ↔ FrLaw σ W.sgTrue (rowTyG Ch P) (levQX R sel mu) H fun p : XPt Pr R sel mu => F () (trivPt X _ _ _ _ p)

theorem **mtEquiv_apply** (F : MtP W Ch sel mu P σ) (p : (stepX Ch sel mu P).D ()) : (mtEquiv W Ch sel mu cst L P u rho rho' cD Th σ hd F).1 () p = F.1 (toRow sel mu p.1.1)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RingKrullDimIntegers

theorem **eq_std** (C : CxSt F) : C = std F

theorem **pn_nonneg** (f : C.FnC Y σ) (y : Pts Y σ) : 0 ≤ C.pn Y σ Δ m f y

theorem **pn_sum** {ι : Type} [Fintype ι] (c : ι → ℂ) (f : ι → Y.Fn) (y : Pts Y σ) (g : Y.Fnˣ) (hg : Δ.IsEqn (under y) g) (hf : ∀ i, Δ.RegularAt (under y) (f i)) : C.pn Y σ Δ m (∑ i, c i • C.cx Y σ (f i)) y = ‖∑ i, c i * ratVal (f i * (g : Y.Fn)) y‖ ^ 2 * C.pn Y σ Δ m (C.cx Y σ (g⁻¹ : Y.Fnˣ)) y

theorem **mSmooth_ofWeight_iff** {L : LB Y.X} {r : L.ι → Y.Fnˣ} (h : Presents Δ L r) (φ : Weight Y.str σ L) : C.mSmooth Y σ Δ (C.ofWeight Y σ Δ L r h φ) ↔ φ.IsCInf

theorem **NefFib.isModel** (h : A.NefFib) : A.IsModel

theorem **IsArTop.isModel** {t : ℝ} (h : A.IsArTop t) : A.IsModel

theorem **Nef.isModel** (h : A.Nef) : A.IsModel

theorem **ringKrullDim_integers** : ringKrullDim (𝒪 F) = 1

theorem **isArithVar_pspace** (n : ℕ) : IsArithVar (PSpace.str (𝒪 F) n) n

theorem **valAt_stalk** {K : Type} [Field K] {Z : Scheme.{0}} [IsIntegral Z] (y : Sp K ⟶ Z) (a : Z.presheaf.stalk (below y)) : valAt y (algebraMap _ Z.functionField a) = (Scheme.stalkClosedPointTo y).hom a

theorem **isEqn_eqn** {Z : Scheme.{0}} [IsIntegral Z] (Δ : CartierDiv Z) (x : Z) : Δ.IsEqn x (eqn Δ x)

theorem **valAt_zpow** {K : Type} [Field K] {Z : Scheme.{0}} [IsIntegral Z] (y : Sp K ⟶ Z) {f : Z.functionField} (hf : UnitAt (below y) f) : valAt y f ≠ 0 ∧ ∀ m : ℤ, valAt y (f ^ m) = valAt y f ^ m

theorem **ratVal_one** {Y : Var F} {σ : F →+* ℂ} (y : Pts Y σ) : ratVal (1 : Y.Fn) y = 1

theorem **unitAt_eqn_zero** {Y : Var F} (x : Y.X) : UnitAt x (eqn (0 : Y.Dv) x : Y.Fn)

theorem **Induced.divide** (h : A.Induced M L k cf) (j : ℕ+) : (A.divide j).Induced M L k fun v => cf v / (j : ℕ)

theorem **Induced.twist_fin** (h : A.Induced M L k cf) {c : HeightOneSpectrum (𝒪 F) → ℝ} (hc : (Function.support c).Finite) : (A.twist 0 c).Induced M L k (cf + c)

theorem **pn_triv** (σ : F →+* ℂ) (τ : ℕ) {f : Y.Fn} {y : Pts Y σ} (hf : RegAt (under y) f) : C.pn Y σ _ (C.mPow Y σ 0 τ ((triv C Y).m σ)) (C.cx Y σ f) y = ‖ratVal f y‖ ^ 2

theorem **induced_triv** (M : Model Y) : (triv C Y).Induced M (HermLB.triv M.p) 1 0

theorem **nrmInf_triv** (σ : F →+* ℂ) (τ : ℕ) (y : Pts Y σ) : (triv C Y).nrmInf σ τ 1 y = 1

theorem **nrmFin_triv** (v : HeightOneSpectrum (𝒪 F)) (τ : ℕ) (e : Ext v) (y : AlgPt Y) : (triv C Y).nrmFin v τ 1 e y = 1

theorem **IsModel.divide** (h : A.IsModel) (j : ℕ+) : (A.divide j).IsModel

theorem **nefFib_triv** (C : CxSt F) (M : Model Y) : (triv C Y).NefFib

theorem **isModel_triv_proj** (C : CxSt F) (n : ℕ) {c : HeightOneSpectrum (𝒪 F) → ℝ} (hc : (Function.support c).Finite) (j : ℕ+) : (((triv C (proj F n)).twist 0 c).divide j).IsModel

theorem **mSmooth_triv** (σ : F →+* ℂ) : C.mSmooth Y σ 0 ((triv C Y).m σ)

theorem **one_mem_small_triv** (τ : ℕ) : 1 ∈ (triv C Y).small τ

theorem **supInf_twist** {C : CxSt F} {Y : Var F} (A : Adl C Y) (ci : InfinitePlace F → ℝ) (cf : HeightOneSpectrum (𝒪 F) → ℝ) (v : InfinitePlace F) (τ : ℕ) (f : Y.Fn) : (A.twist ci cf).supInf v τ f = Real.exp (−((τ : ℝ) * ci v)) * A.supInf v τ f

theorem **supFin_twist** {C : CxSt F} {Y : Var F} (A : Adl C Y) (ci : InfinitePlace F → ℝ) (cf : HeightOneSpectrum (𝒪 F) → ℝ) (v : HeightOneSpectrum (𝒪 F)) (τ : ℕ) (f : Y.Fn) : (A.twist ci cf).supFin v τ f = Real.exp (−((τ : ℝ) * cf v)) * A.supFin v τ f

theorem **supInf_ofW** (C : CxSt F) (Y : Var F) (L : Y.Dv) (W : HLat F) (ι : W.V →ₗ[F] Y.Fn) (v : InfinitePlace F) (τ : ℕ) (f : Y.Fn) : (ofW C Y L W ι).supInf v τ f = ⨆ y, Real.sqrt (C.pn Y v.embedding (τ • L) (C.mPow Y v.embedding L τ (C.fsMet Y v.embedding L (fun k : W.ι => ∑ i, (W.X v) i k • C.cx Y v.embedding (ι (W.b i))))) (C.cx Y v.embedding f) y)

theorem **deg_ofW** (C : CxSt F) (Y : Var F) (L : Y.Dv) (W : HLat F) (ι : W.V →ₗ[F] Y.Fn) : (ofW C Y L W ι).deg = (Y.inter (fun _ => L) : ℝ)

theorem **deg_twist** {C : CxSt F} {Y : Var F} (A : Adl C Y) (ci : InfinitePlace F → ℝ) (cf : HeightOneSpectrum (𝒪 F) → ℝ) : (A.twist ci cf).deg = A.deg

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.RowRatW

theorem **hypH_of** (c : GlobalTheory.Published.DiscreteSeriesCover) (h1 : GlobalTheory.PrimIn.Hypotheses3) (h2 : GlobalTheory.Published.RegularisedSiegelWeilFirstTerm) (h3 : GlobalTheory.Published.HeckeLPartialContinuation) (m : MultL) (r : RowRatW) : HypH

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SSupCompSecId

theorem **nrm_hSec** (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : ct.nrm (hSh ct F₁ F₂) (hSec ct φ) y = sSup (nrmSet y.z (homMod F₁.toMod F₂.toMod) (compSec (𝟙 _) φ.iso.hom))

theorem **nrm_hSec_pos** (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : 0 < ct.nrm (hSh ct F₁ F₂) (hSec ct φ) y ∧ ct.nrm (hSh ct F₂ F₁) (hSec ct φ.symm) y = (ct.nrm (hSh ct F₁ F₂) (hSec ct φ) y)⁻¹

theorem **nrm_eq_vr** (φ : ct.GIso F₁ F₂) (y : ct.Pt Z) : ct.nrm (hSh ct F₁ F₂) (hSec ct φ) y = (p : ℝ) ^ (−(vr ct φ y))

theorem **sSup_compSec_id** {X : Scheme.{0}} {F G : X.Modules} {C : Formalism.MainNotions.DIV.Cons.SCF p} (z : Sp (Formalism.MainNotions.DIV.Cons.intO C.S) ⟶ X) (hp : p ≠ 0) (hF : IsVB F 1) [IsFiniteQC G] {u : Γ(F, ⊤)} (hu : GlobalTheory.VB.IsFrame

fun _ : Fin 1 => u) (φ : F.restrict (GlobalTheory.PtNorm.gen X p).ι ≅ G.restrict (GlobalTheory.PtNorm.gen X p).ι) : sSup (nrmSet z G (homApp (homOfRestrict φ.hom) le_rfl (resM F le_top u))) = sSup (nrmSet z (homMod F G) (compSec (𝟙 _) φ.hom))

theorem **exists_ats** (h : ∀ (p : ℕ), p.Prime → ∀ W : ValuationSubring ℂ, (p : ℂ) ∈ W.nonunits → ∃ ct : At S κ p, ct.W = W) (jC : ∀ (p : ℕ) [Fact p.Prime], PlL D p → (DV.KW p →+* ℂ)) (hj : ∀ (p : ℕ) [Fact p.Prime] v, JcPin D p v (jC p v)) : ∃ cts : Ats S κ, cts.Over jC

theorem **over_induces** {cts : Ats S κ} {jC : ∀ (p : ℕ) [Fact p.Prime], PlL D p → (DV.KW p →+* ℂ)} (h : cts.Over jC) (p : ℕ) [Fact p.Prime] (v : PlL D p) (hj : JcPin D p v (jC p v)) (x : D.what₂.LL) : ((κ.e x : κ.c.F) : ℂ) ∈ (cts p v).W ↔ v.1.valuation D.what₂.LL x ≤ 1

theorem **isBCAt_bcAt** (ct : At S κ p) (Z : CodeAt S κ) : IsBCAt ct Z (bcAt ct Z)

theorem **bcAt_eq_blowUp** (ct : At S κ p) (Z : CodeAt S κ) (J : ct.Idl ct.Xm) (hJ : ∀ Z', IsBCAt ct Z Z' ↔ IsNBlow ct.Xm J Z') : bcAt ct Z = ct.blowUp ct.Xm J

theorem **bc_eq_blowUp** (hp : p.Prime) (Z : CodeAt S κ) (h : ∀ v, ∃ d, ∀ Z', IsBCAt (@cts p ⟨hp⟩ v) Z Z' ↔ IsNBlow _ ((F.st (@cts p ⟨hp⟩ v)).tr.jIdl none d) Z') : ∃ d : ((locStOf F cts).ex p).IvD none, (globStOf F cts).bc p Z = ((locStOf F cts).ex p).blowUp ((locStOf F cts).loc p).Xm (((locStOf F cts).ex p).jIdl none d)

theorem **theta_clauses** (hp : p.Prime) : (∀ Z (y : ((locStOf F cts).loc p).Pt Z) (a : ((locStOf F cts).loc p).Ylat y), ((locStOf F cts).loc p).psiV y a = ((locStOf F cts).loc p).be y a a / 2) ∧ ((locStOf F cts).loc p).IsInv ((locStOf F cts).loc p).piPP ∧ ∀ c, ((locStOf F cts).loc p).IsInv (((locStOf F cts).loc p).keySh c)

theorem **nrm_pbKF_iff** (hp : p.Prime) {c : ℕ} (kf : ngTrue.KF (D.R.f ∗ N) c) : (∀ y, ((locStOf F cts).loc p).nrm (((locStOf F cts).loc p).keySh c) ((locStOf F cts).pbKF p kf) y = 1) ↔ ∀ v y, (@cts p ⟨hp⟩ v).nrm ((@cts p ⟨hp⟩ v).keySh c) ((@cts p ⟨hp⟩ v).pbKF kf) y = 1

theorem **alp_eq** (hp : p.Prime) : ((locStOf F cts).locB p hp).alp = F.α p

theorem **xn_injective** (hp : p.Prime) : Function.Injective ((locStOf F cts).loc p).Xn

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SameDegreeZeroFormsSelf

theorem **SRest.rowTie** {D : D7} {F : ∀ u, Forms0 D u} {P : SPack D F} (S : SRest P) : RowDictTie P.X P.G S.hI.eHH

theorem **pl_heq_of_pin** {D : D7} (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (pin : ∀ x : X.B.I.E, HEq (X.ιE x) x) : HEq X.pl (Formalism.L2.Host.tauPl D).symm

theorem **SPack.ιE_heq** {D : D7} {F : ∀ u, Forms0 D u} (P : SPack D F) : HEq P.X.ιE (RingEquiv.refl D.host.E)

theorem **SPack.pl_heq** {D : D7} {F : ∀ u, Forms0 D u} (P : SPack D F) : HEq P.X.pl (Formalism.L2.Host.tauPl D).symm

theorem **SPack.locTied_of_sIso** {D : D7} {F : ∀ u, Forms0 D u} (P : SPack D F) (h : SIso D.s (sOf P.fin)) : Formalism.L2.SG.Pf.LocCP.LocTied D

theorem **SRest.finrank_HHX** {D : D7} {F : ∀ u, Forms0 D u} {P : SPack D F} (S : SRest P) : Module.finrank ℂ (HHX P.X) = D.what₂.Dall

theorem **SRest.toNew_s_Cu** {D : D7} {F : ∀ u, Forms0 D u} {P : SPack D F} (S : SRest P) (u : D.what₂.Word) (x : D.s.Cu u) : (P.toNew S.finrank_HHX S.j).s_Cu u (S.hI.eCu u x) = D.s_Cu u x

theorem **SRest.toNew_sEE** {D : D7} {F : ∀ u, Forms0 D u} {P : SPack D F} (S : SRest P) (x : D.what₂.EE) : (P.toNew S.finrank_HHX S.j).sEE x = S.hI.eHH (D.sEE x)

theorem **exists_sRest_of_sTies** {D : D7} (F : ∀ u, Forms0 D u) (T : Formalism.L2.SG.SpectralTiesReal.STies D) (hF : FinX T.Y) (tr : ∀ u, Transport T.X u (F u)) (into : ∀ u, ColDictInto T.X (tr u) T.Y.hK) (bij : ∀ u, ColDictBij T.X (tr u) T.Y.hK) (lev : ∀ u, LevelMatch T.X (tr u) T.Y.hK) (colPin : ∀ (u : D.what₂.Word) (x : D.s.Cu u), colDict T.X (tr u) T.Y.hK (into u) (bij u) ((T.hI hF).eCu u x) = D.s_Cu u x) (inst : T.X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (pin : ∀ x : T.X.B.I.E, HEq (T.X.ιE x) x) (G : GoodOp T.X) (rowTie : RowDictTie T.X G (T.hI hF).eHH) (locTied : Formalism.L2.SG.Pf.LocCP.LocTied D) : ∃ (P : SPack D F) (S : SRest P), S.toSTies = T

theorem **sameDegreeZeroForms_self** {D : D7} (F : ∀ u, Forms0 D u) : SameDegreeZeroForms (SNew.self D) F F

theorem **nonempty_sPack_of_steps** {D : D7} (F : ∀ u, Forms0 D u) (hG : Formalism.L2.SG.ConcreteSettingExists.GapHyp D) (hC : Formalism.L2.SG.SplitPrimeNormLemmas.CoefWeightedLoc) (s1 : Formalism.L2.SG.SplitPrimeNormLemmas.LatExists D) (s2 : Formalism.L2.SG.SplitPrimeNormLemmas.PsiExists D) (s3 : Formalism.L2.SG.SplitPrimeNormLemmas.RefExists D) (s4 : Formalism.L2.SG.SplitPrimeNormLemmas.XuExists D) (s5 : Formalism.L2.SG.SplitPrimeNormLemmas.BunExists D) (s6 : Formalism.L2.SG.SplitPrimeNormLemmas.PiExists D) (s7 : Formalism.L2.SG.SplitPrimeNormLemmas.MuExists D) (s8 : GlobalTheory.St.WBlkExists D) (s9 : Formalism.L2.SG.SplitPrimeNormLemmas.MExists D) (s10 : GlobalTheory.St.RescaleStep3 D) (tw : GlobalTheory.St.WittExists D) (t0 : GlobalTheory.St.GenusMatch D F) (t3 : GlobalTheory.St.LevelConj D F) (t4 : GlobalTheory.St.HcExists D F) (t5 : GlobalTheory.St.JExists D F) (hY : ∀ X : HidaSetting

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SameRedEquivalence

theorem **sameRed_equivalence** (F' F'' : FlagD H r) : Equivalence (SameRed F' F'')

theorem **isPairMap** (b : RedMap F F' F'' Λ g) : IsPairMap F F' F'' Λ g b.out

theorem **mk_out** (b : RedMap F F' F'' Λ g) : mk b.out b.isPairMap = b

theorem **mk_eq_mk** (h : IsPairMap F F' F'' Λ g β) (h₁ : IsPairMap F F' F'' Λ g β₁) : mk β h = mk β₁ h₁ ↔ SameRed F' F'' β β₁

theorem **ext** {b c : RedMap F F' F'' Λ g} (h : SameRed F' F'' b.out c.out) : b = c

theorem **fixSp_sup_le** (F F' F'' : FlagD H r) (β : V ≃ₗ[K] V) : fixSp F F' F'' β ⊔ F.X ≤ perp H (xPlus F)

theorem **corank_congr** (h : SameRed F' F'' β β₁) : corank F F' F'' β₁ = corank F F' F'' β

theorem **corank_add** (F F' F'' : FlagD H r) (β : V ≃ₗ[K] V) : corank F F' F'' β + finrank K ɪ(fixSp F F' F'' β ⊔ F.X) = finrank K (perp H (xPlus F))

theorem **corank_symm** (h : IsPairMap F F' F'' Λ g β) : corank F F'' F' β.symm = corank F F' F'' β

theorem **IsScalar.congr** (h : IsScalar F F' F'' β) (e : SameRed F' F'' β β₁) : IsScalar F F' F'' β₁

theorem **IsScalar.symm** (h : IsScalar F F' F'' β) (hp : IsPairMap F F' F'' Λ g β) : IsScalar F F'' F' β.symm

theorem **refl** (F : FlagD H r) : Carry (LinearEquiv.refl K V) F F

theorem **symm** (h : Carry s F F₁) : Carry s.symm F₁ F

theorem **perp_iff** (h : Carry s F F₁) {A B : Submodule K V} (e : ∀ x, s x ∈ B ↔ x ∈ A) (x : V) : s x ∈ perp H B ↔ x ∈ perp H A

theorem **perp** (h : Carry s F F₁) (x : V) : s x ∈ perp H F₁.X ↔ x ∈ perp H F.X

theorem **isComp** (h : Carry s F F₁) (k : Carry s F' G') {u : V} {a a' : K} (e : IsComp F F' u a a') : IsComp F₁ G' (s u) a a'

theorem **V0** (h : Carry s F F₁) (k : Carry s F' G') (x : V) : s x ∈ V0 F₁ G' ↔ x ∈ V0 F F'

theorem **xPlus** (h : Carry s F F₁) (x : V) : s x ∈ xPlus F₁ ↔ x ∈ xPlus F

theorem **norLift** (h : Carry s F F₁) (k : Carry s F' G') (x : V) : s x ∈ norLift F₁ G' ↔ x ∈ norLift F F'

theorem **symm** : Moves F G' G'' β₁ F' F'' β s₁.symm s₂.symm

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ScaledTwistingSheafSpec

theorem **idealSheaf_zero** (X : Scheme.{0}) : (EffCartierData.zero X).idealSheaf = ⊤

theorem **eff_ofEff** (C : EffCartierData X) : (ofEff C).Eff

theorem **stalkId_comap_eq_top** (I : X.IdealSheafData) (g : X' ⟶ X) (x : X') : FinalTheorems.stalkId (I.comap g) x = ⊤ ↔ g x ∉ I.support

theorem **posAt_ofEff** (C : EffCartierData X) (g : X' ⟶ X) (x : X') : (ofEff C).PosAt g x ↔ g x ∈ C.idealSheaf.support

theorem **zeroAt_ofEff** (C : EffCartierData X) (g : X' ⟶ X) (x : X') : (ofEff C).ZeroAt g x ↔ g x ∉ C.idealSheaf.support

theorem **eff_opp_ofEff** (C : EffCartierData X) : (ofEff C).opp.Eff ↔ C.idealSheaf = ⊤

theorem **norm_valF_unit** {X : Scheme.{0}} {C : Type} [NontriviallyNormedField C] [IsUltrametricDist C] (z : Sp ɪ(intO C) ⟶ X) (U : X.Opens) (h : ⊤ ≤ z ⁻¹ᵘ U) (u : Γ(X, U)ˣ) : ‖valF (spMap (intO C).subtype ≫ z) U u.valꟳ = 1

theorem **logBd_pullGenLB** {X Q : Scheme.{0}} {D : LB X} (t : D.Gen) (ι : Q ⟶ X) (p : ℕ) : LogBd D ι (pullGenLB t ι) p 0

theorem **LogBd.mono** {X Q : Scheme.{0}} {D : LB X} {ι : Q ⟶ X} {w : (D.pull ι).Gen} {p : ℕ} {δ δ' : ℝ} (h : LogBd D ι w p δ) (hδ : δ ≤ δ') : LogBd D ι w p δ'

theorem **FinLB.lbIso_pull** {X X' : Scheme.{0}} (L : FinLB X) (f : X' ⟶ X) : LBIso (L.pull f).lb (L.lb.pull f)

theorem **dominates_tor** {c : Coeff l} {O O' : Type} [CommRing O] [CommRing O'] {β : O →+* c.F} {β' : O' →+* c.F} {Z : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Model (Xo l) c β} (T : TorModel Z) {Y' : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Model (Xo l) c β'} {b : O →+* O'} (h : Dominates Z Y' b) : Dominates T.Z Y' b

theorem **proj** : IsProjOver Z.s

theorem **exists_preCode** (Z' : GModel Xo c) : ∃ (Z : PreCode Xo c) (e : Z'.X ≅ Z.X), Z.Iso Z' e

theorem **exists_code** (Z' : GModel Xo c) (h : DomLoc Y hg Z') : ∃ (Z : Code Y c hg) (e : Z'.X ≅ Z.X), Z.Iso Z' e

theorem **exists_code_tor** {Z' : GModel Xo c} (h : DomLoc Y hg Z') (T : TorModel Z') : ∃ (Z : Code Y c hg) (e : T.Z.X ≅ Z.X), Z.Iso T.Z e

theorem **normal** : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme Z.X

theorem **flat** : Flat Z.s

theorem **ampleHodge_of_ample** {Z' : GModel Xo c} (T : TorModel Z') (h : IsRecTor T) (e : T.Z.X ≅ Z.X) (he : Z.Iso T.Z e) {B : CDiv Z.X} (h₁ : B.pos.comap e.hom = T.BD.idealSheaf) (h₂ : B.neg = ⊤) : Z.AmpleHodge T.m0 B

theorem **exists_eff_posBd** {Z' : GModel Xo c} (T : TorModel Z') (e : T.Z.X ≅ Z.X) (he : Z.Iso T.Z e) : ∃ B : CDiv Z.X, B.pos.comap e.hom = T.BD.idealSheaf ∧ B.neg = ⊤ ∧ B.Eff ∧ Z.PosBd B

theorem **nefFib_of_ample** (h : IsAmpleLB M.L.lb) : M.NefFib

theorem **ampleGen_of_ample** (h : IsAmpleLB M.L.lb) : M.AmpleGen

theorem **limRA_of_ample** (h : IsAmpleLB M.L.lb) : M.LimRA

theorem **LimRA.ampleGen** (h : M.LimRA) : M.AmpleGen

theorem **scaledTwistingSheaf_spec** (Z : Code Y c hg) (m : ℕ) : (scaledTwistingSheaf Z m).NefFib ∧ (scaledTwistingSheaf Z m).AmpleGen ∧ (scaledTwistingSheaf Z m).LimRA

theorem **bcOf_eq** {S : Sig A N j} {κ : Lv A} {M : ℕ → Type} (dflt : ∀ p, M p) (a : ∀ p, p.Prime → At S κ p) (emb : ∀ p hp, (a p hp).Mdl → M p) {p : ℕ} (hp : p.Prime) (Z : CodeAt S κ) (h : ∃ Z', IsBCAt (a p hp) Z Z') : bcOf dflt a emb p Z = emb p hp h.choose

theorem **choose_eq_blowUp** {S : Sig A N j} {κ : Lv A} {p : ℕ} (ct : At S κ p) (Z : CodeAt S κ) (X : ct.Mdl) (J : ct.Idl X) (h : ∃ Z', IsBCAt ct Z Z') (hJ : ∀ Z', IsBCAt ct Z Z' ↔ IsNBlow X J Z') : h.choose = ct.blowUp X J

theorem **exists_codeAt_iso_with_positive_ample_divisor** {S : Sig A N j} {κ : Lv A} (s : FinalTheorems S κ) : ∃ (Z : CodeAt S κ) (e : s.T.Z.X ≅ Z.X) (B : CDiv Z.X), Z.Iso s.T.Z e ∧ B.pos.comap e.hom = s.T.BD.idealSheaf ∧ B.neg = ⊤ ∧ B.Eff ∧ Z.PosBd B ∧ Z.AmpleHodge s.T.m0 B

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SecondChernDegreeVanishesTrueSurfaces

theorem **isProjOver** : IsProjOver C.str

theorem **strHom_eq** {k : Type} [Field k] (C : ProjCode k) : strHom k C.X = C.str

theorem **exists_iso** {Y : Scheme.{0}} {f : Y ⟶ Sp R} (h : IsProjOver f) : ∃ (C : ProjCode R) (e : C.X ≅ Y), e.hom ≫ f = C.str

theorem **isProjOver_toSpZ** {R : Type} [CommRing R] [Module.Finite ℤ R] {Y : Scheme.{0}} {f : Y ⟶ Sp R} (h : IsProjOver f) : IsProjOver (toSpZ Y)

theorem **isArithVar_toSpZ** {k : Type} [CommRing k] {Y : Scheme.{0}} {p : Y ⟶ Sp k} {D : ℕ} (h : IsArithVar p D) : IsArithVar (toSpZ Y) D

@[simp] theorem **P_g** (Q : PPCode S) : Q.P.g = Q.g

theorem **isProjOver** (Q : PPCode S) : IsProjOver (toSpZ Q.P.A.left)

theorem **exists_isPull** (P : PPAS S) (hg : 1 ≤ P.g) (f : S' ⟶ S) (h : IsProjOver (toSpZ (pullback P.A.hom f))) : ∃ (Q : PPCode S') (φ : Q.P.A.left ⟶ P.A.left), Q.g = P.g ∧ Q.P.IsPull P f φ

theorem **exists_isPull_of_isIso** (P : PPAS S) (hg : 1 ≤ P.g) (f : S' ⟶ S) [IsIso f] (h : IsProjOver (toSpZ P.A.left)) : ∃ (Q : PPCode S') (φ : Q.P.A.left ⟶ P.A.left), Q.g = P.g ∧ Q.P.IsPull P f φ

theorem **exists_const** {R : Type} [CommRing R] [Module.Finite ℤ R] (P : Out.PolAV (Sp R)) (hg : 1 ≤ P.g) {f : S' ⟶ Sp R} (hf : IsProjOver f) : ∃ (Q : PPCode S') (φ : Q.P.A.left ⟶ P.A.left), Q.g = P.g ∧ Q.P.IsPull (.ofPolAV P) f φ

theorem **secondChern_eq_double_sum** {h : VMetric s.C.str (omega s.A.P.A) s.A.P.g} (H : s.A.P.IsHodge h) {a : ℝ} (ha : h.IsCh2Deg a) : s.ch2 = a

theorem **degOm_eq** : s.degOm = (chi ℂ ((Scheme.Modules.pullback (cxFibre.fst s.C.str)).obj (omega s.A.P.A)) : ℝ) − s.A.g ∗ (chi ℂ (structureModule (cxFibre s.C.str)) : ℝ)

theorem **cxDeg_pullback_iso** {k : Type} [CommRing k] {X Y : Scheme.{0}} {p : X ⟶ Sp k} {q : Y ⟶ Sp k} (e : Y ≅ X) (w : e.hom ≫ p = q) (E : X.Modules) (r : ℕ) : cxDeg q ((Scheme.Modules.pullback e.hom).obj E) r = cxDeg p E r

theorem **Surf.exists_iso** {X : Scheme.{0}} {p : X ⟶ Sp ℤ} (hX : IsArithVar p 1) (P : PPAS X) (hg : 1 ≤ P.g) (h : IsProjOver (toSpZ P.A.left)) : ∃ (s : Surf) (e : s.C.X ≅ X) (φ : s.A.P.A.left ⟶ P.A.left), e.hom ≫ p = s.C.str ∧ s.A.g = P.g ∧ s.A.P.IsPull P e.hom φ ∧ s.degOm = P.degOmega p

theorem **secondChernFinite_elliptic** (h : s.A.g = 1) : s.ch2Q = (topSelfIntersection (ZMod s.p) (det (omega s.A.P.A) 1) 2 : ℚ) / 2

theorem **secondChernFinite_eq** : s.ch2Q = IndefiniteRowSideFullHecke.GenericFiniteMaps.degCh2 (ZMod s.p) (omega s.A.P.A)

theorem **isFlagCh2_iff** (H : IndefiniteRowSideFullHecke.GenericFiniteMaps.CoverStatement) {a : ℚ} : s.IsFlagCh2 a ↔ a = s.ch2Q

theorem **degCh2_pullback_iso** {k : Type} [Field k] {X Y : Scheme.{0}} [Algebra k Γ(X, ⊤)] [Algebra k Γ(Y, ⊤)] (e : Y ≅ X) (w : e.hom ≫ strHom k X = strHom k Y) {E : X.Modules} {r : ℕ} (hE : IsVB E r) : degCh2 k ((Scheme.Modules.pullback e.hom).obj E) r = degCh2 k E r

theorem **SurfFp.exists_iso** (p : ℕ) [Fact p.Prime] {T : Scheme.{0}} [Algebra (ZMod p) Γ(T, ⊤)] (hT : IsProjSurface (ZMod p) T) (P : PPAS T) (hg : 1 ≤ P.g) (h : IsProjOver (toSpZ P.A.left)) : ∃ (s : SurfFp) (e : s.C.X ≅ T) (φ : s.A.P.A.left ⟶ P.A.left), s.p = p ∧ s.A.g = P.g ∧ s.A.P.IsPull P e.hom φ ∧ s.ch2Q = degCh2 (ZMod p) (omega P.A) P.g

@[simp] theorem **of_put** (t : SurfSt) (X : DivX D) : of (t.put X) = t

@[simp] theorem **put_of** (X : DivX D) : (of X).put X = X

theorem **hasTrueSurf_put** (X : DivX D) : HasTrueSurf (surfSt.put X)

theorem **HasTrueSurf.recordIsGeometric** (h : HasTrueSurf X) : IndefiniteRowSideFullHecke.ZeroSectionBase.RecordIsGeometric D X

theorem **HasTrueSurf.secondChernDegree_vanishes_trueSurfaces** (h : HasTrueSurf X) (H : IndefiniteRowSideFullHecke.ZeroSectionBase.Ch2HodgeVanishes) : Formalism.MainNotions.WeightThresholdDefs.Conclusion4b X

theorem **HasTrueSurf.secondChern_eq_of_trueSurfaces** (h : HasTrueSurf X) {r : ℝ} (H : ∀ (S : Scheme.{0}) (p : S ⟶ Sp ℤ) [IsIntegral S], IsArithVar p 1 → ∀ P : PPAS S, 1 ≤ P.g → P.ch2Omega p = r / 2 * P.degOmega p) : Formalism.MainNotions.WeightThresholdDefs.Conclusion4a X r

theorem **HasTrueSurf.degCh2_eq_zero** (h : HasTrueSurf X) (H : Formalism.MainNotions.WeightThresholdDefs.Conclusion4b X) (p : ℕ) [Fact p.Prime] {T : Scheme.{0}} [Algebra (ZMod p) Γ(T, ⊤)] (hT : IsProjSurface (ZMod p) T) (P : PPAS T) (hg : 1 ≤ P.g) (hP : IsProjOver (toSpZ P.A.left)) : degCh2 (ZMod p) (omega P.A) P.g = 0

theorem **Surf.exists_const** (F : Type) [Field F] [NumberField F] (W : _root_.WeierstrassCurve (𝒪 F)) [W.IsElliptic] : ∃ (s : Surf) (e : s.C.X ≅ PSpace (𝒪 F) 1) (φ : s.A.P.A.left ⟶ (WeierstrassCurve.ellPolAV W).A.left), s.A.g = 1 ∧ s.A.P.IsPull (.ofPolAV (WeierstrassCurve.ellPolAV W)) (e.hom ≫ PSpace.str (𝒪 F) 1) φ

theorem **omega_sq_sqrtTwentyNine** : (QuadraticAlgebra.omega : QuadraticFieldTwentyNine) ^ 2 = 1 + 5 * QuadraticAlgebra.omega

theorem **fundamentalUnit_sq** : fundamentalUnitSqrtTwentyNine ^ 2 = 1 + 5 * fundamentalUnitSqrtTwentyNine

theorem **tateCurve_discriminant** : tateCurveSqrtTwentyNine.Δ = -fundamentalUnitSqrtTwentyNine ^ 10

theorem **Surf.exists_g_eq_one** : ∃ s : Surf, s.A.g = 1

theorem **SurfFp.exists_const** (p : ℕ) [Fact p.Prime] (W : _root_.WeierstrassCurve (ZMod p)) [W.IsElliptic] : ∃ (C : ProjCode (ZMod p)) (h : IsProjSurface (ZMod p) C.X) (A : PPCode C.X) (φ : A.P.A.left ⟶ (WeierstrassCurve.ellPolAV W).A.left), Nonempty (C.X ≅ PSpace (ZMod p) 2) ∧ (SurfFp.mk p C h A).A.g = 1 ∧ A.P.IsPull (.ofPolAV (WeierstrassCurve.ellPolAV W)) C.str φ

theorem **SurfFp.secondChernFinite_constantFamily** (s : SurfFp) (P : PPAS (Sp (ZMod s.p))) (hg : P.g = 1) (hs : s.A.g = 1) {φ : s.A.P.A.left ⟶ P.A.left} (H : s.A.P.IsPull P s.C.str φ) : s.ch2Q = 0

theorem **SurfFp.exists_g_eq_one** (p : ℕ) [Fact p.Prime] : ∃ s : SurfFp, s.p = p ∧ s.A.g = 1 ∧ s.ch2Q = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SeriesRatSOf

theorem **seriesRatS_of** (h : SeriesRat) : SeriesRatS

theorem **SeriesRatS.untw** (h : SeriesRatS) : UntwRatS

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SettingsFormsExistOfGrid

theorem **depth0_mono** (D : D7) {C C' b : ℝ} (h : C ≤ C') (hb : 1 ≤ b) : depth0 D C b ≤ depth0 D C' b

theorem **gridPin_anti** {D : D7} {I : Type} {Lc : LevCh D I} {μ0 : I} (hL : Lc.Is μ0) {CF CF' b : ℝ} (h : CF ≤ CF') (hb : 1 ≤ b) (hp : GridPin D Lc CF' b) : GridPin D Lc CF b

theorem **not_gridPin** {D : D7} {I : Type} (Lc : LevCh D I) (CF b : ℝ) (h : q0 D = D.host.g 0) : ¬ GridPin D Lc CF b

theorem **fixed_not_diagonal** (𝒞 : FamClass) : ∃ P : ℝ → ∀ {eps : ℝ}, FamP eps → ℝ → Prop, (∃ CF0 : ℝ, ∀ CF, CF0 ≤ CF → Closed 𝒞 (P CF)) ∧ ∀ (eps : ℝ) (Fm : FamP eps) (CH : ℝ), ¬ P Fm.CF Fm CH

theorem **gridPin_of_fit** (p : PtAt ex Csz) {CF : ℝ} {G : Grid p CF} (h : GridFit p G) : GridPin D p.R.toRealDat.toLevCh CF L.sc.b

theorem **fit_of_gridPin** (p : PtAt ex Csz) {μ0} (hL : p.R.toRealDat.toLevCh.Is μ0) {CF CF' : ℝ} (h : CF ≤ CF') (hb : 1 ≤ L.sc.b) (hp : GridPin D p.R.toRealDat.toLevCh CF' L.sc.b) {xd v0 rep} (ht : GlobalTheory.EX.TwistAt L ex.nch xd v0 rep) (h1 : ValLL0 D p.R.C) (h3 : Formalism.L2.Host.PsiChoiceCompatible D p.R.C ex.nch p.Tp) : ∃ G : Grid p CF, GridFit p G

theorem **not_gridPin_of_empty** {D : D7} {I : Type} (Lc : LevCh D I) (CF b : ℝ) {i : D.ntw.Tw} (hi : D.ntw.NonzeroIndexPlace i) (h : IsEmpty (D.ntw.abelian₂ i)) : ¬ GridPin D Lc CF b

theorem **settingsFormsExist_of_grid** : SettingsFormsExistGrid hL → SettingsFormsExist hL

theorem **seriesExist_of_grid** : SeriesExistGrid hL → SeriesExist hL

theorem **moduliPointsExist_of_grid** : ModuliPointsExistGrid hL → SB2 hL

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SpecialNotMono0

theorem **CGK.naut_ofSide** {R : Ranks} {H : Host R} {what₂ : Rows.Data R} {N : NData R H what₂} (A : NAut N) (hE : IsEmpty N.Loc) : (CGK.ofSide A).naut hE = A

theorem **setBC_self** (D : D7) : setBC D D.nbc = D

theorem **setTy_self** (D : D7) (hE : IsEmpty D.nd.Loc) : setTy D hE D.ntypes D.nStage_trivialCharacter_compat D.nStage_trivialCharacter_coeff_eq = D

theorem **setCG_CG** (D : D7) (hE : IsEmpty D.nd.Loc) (K : CGK D.naut) : (setCG D hE K).naut.CG = K.CG

theorem **tame_lpSp** {E Ω : Type} [Field E] [NumberField E] [Field Ω] [Algebra E Ω] [IsAlgClosure E Ω] {A : ValuationSubring Ω} {w : HeightOneSpectrum (𝒪 E)} (h : IsPlaceOver A w) (n : ℕ) : (WD.lpSp h n).tame

theorem **not_mono0_lpSp** {E Ω : Type} [Field E] [NumberField E] [Field Ω] [Algebra E Ω] [IsAlgClosure E Ω] {A : ValuationSubring Ω} {w : HeightOneSpectrum (𝒪 E)} (h : IsPlaceOver A w) {n : ℕ} (hn : 2 ≤ n) : ¬ (WD.lpSp h n).mono0

theorem **RowWD.special_not_mono0** (D : D7) (σ w) : ¬ ((RowWD.special D).par σ w).mono0

theorem **BlocksUnique.not_cusp** {D : D7} (h : BlocksUnique D) (χ : D.naut.CG 1) (hχ : D.naut.csd χ) (hχ' : D.naut.ralg χ) (π : D.naut.CG D.R.r) (hπ : D.naut.csd π) (hπ' : D.naut.ralg π) : ¬ ∀ w, IsSplitPl D.host w → isoLoc D.nd D.naut [(1, χ, D.R.r)] w = isoLoc D.nd D.naut [(D.R.r, π, 1)] w

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SplitFacTwistRight

theorem **nonempty_tow** (D : D7) : Nonempty (Tow D)

theorem **orthEx** (D : D7) : OrthEx D

theorem **nonempty_orthDec** (D : D7) : Nonempty (OrthDec D)

theorem **om_rational** {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {χ χ' : IdC D} {om : GA D V × GA D W →* Module.End ℂ (ThS ψ V W)} (h : IsWeilOn ψ V W χ χ' om) {γ : GA D V} (hγ : γ ∈ GQ V) {δ : GA D W} (hδ : δ ∈ GQ W) (Φ Φ' : ThS ψ V W) (w : WA V W) (e : Φ'.1 = Heisenberg₂ ψ V W w Φ.1) : (om (γ, δ) Φ).1 (actHom V W γ δ w) = Φ.1 w

theorem **root_of_at** {e d : AE D} (hd : d * cA D d = 1) (hs : (e + cA D e) * (d - 1) = d - 1) : e * d + (1 - e) = d * cA D (e * d + (1 - e))

theorem **splitFac_twist_right** {V W : HSp D} (h : V.rank ≠ W.rank) (χ χ' ν : IdC D) (x y : (AE D)ˣ) : splitFac V W χ (χ' * ν) x y = (if V.rank < W.rank then (ν x)⁻¹ else ν x) * splitFac V W χ χ' x y

theorem **splitFac_twist_left** {V W : HSp D} (h : V.rank ≠ W.rank) (χ χ' ν : IdC D) (x y : (AE D)ˣ) : splitFac V W (χ * ν) χ' x y = (if V.rank < W.rank then ν y else (ν y)⁻¹) * splitFac V W χ χ' x y

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SplitMapEqZeroOfMemAnn

theorem **app_mem_imSub** (φ : G ⟶ F) (s : Γ(G, V)) : φ.app V s ∈ (imSub φ).sec V

theorem **mem_imSub_iff** (φ : G ⟶ F) [F.IsQuasicoherent] [G.IsQuasicoherent] (hU : IsAffineOpen U) {s : Γ(F, U)} : s ∈ (imSub φ).sec U ↔ ∃ t, φ.app U t = s

@[reassoc (attr

theorem **mem_preSub_iff** (φ : F ⟶ G) (N : Sub G) {s : Γ(F, V)} : s ∈ (preSub φ N).sec V ↔ φ.app V s ∈ N.sec V

theorem **sectionsHom_app_sum** {ι : Type} [Fintype ι] (u : ι → Γ(F, ⊤)) (a : ι → Γ(X, V)) : (sectionsHom u).app V (∑ i, a i • resM _ le_top (freeBasis X i)) = ∑ i, a i • resM F le_top (u i)

theorem **mem_spanSub_iff** {ι : Type} [Fintype ι] (u : ι → Γ(F, ⊤)) [F.IsQuasicoherent] (hU : IsAffineOpen U) {s : Γ(F, U)} : s ∈ (spanSub u).sec U ↔ ∃ a : ι → Γ(X, U), ∑ i, a i • resM F le_top (u i) = s

theorem **res_mem_spanSub** {ι : Type} (u : ι → Γ(F, ⊤)) (i : ι) (V : X.Opens) : resM F le_top (u i) ∈ (spanSub u).sec V

theorem **π_app_eq_zero_iff** [F.IsQuasicoherent] (hU : IsAffineOpen U) (N : ℕ) (s : Γ(F, U)) : (cokernel.π (N • 𝟙 F)).app U s = 0 ↔ ∃ r, N • r = s

theorem **nsmul_sec_modN** (N : ℕ) (s : Γ(modN F N, V)) : N • s = 0

theorem **resG_minor** {U V : X.Opens} (h : V ≤ U) (l : Fin r → Γ(dual F, U)) (m : Fin r → Γ(N.toMod, U)) : resG X h (Matrix.of fun i k => dualApp (l i) le_rfl (N.incl.app U (m k))).det = (Matrix.of fun i k => dualApp (resM _ h (l i)) le_rfl (N.incl.app V (resM _ h (m k)))).det

theorem **minors_subset** [(dual F).IsQuasicoherent] [N.toMod.IsQuasicoherent] {U V : X.Opens} (hU : IsAffineOpen U) (f : Γ(X, U)) (h : V ≤ U) (hV : V = X.basicOpen f) : minors N r V ⊆ Ideal.span (resG X h '' minors N r U)

theorem **splitMap_eq_zero_of_mem_ann** [IsDomain R] {i k : ι} (h : σ k ≠ σ i) {x : R ⊗[ℤ] O} (hx : x ∈ ann (σ i)) : splitMap σ x k = 0

theorem **exists_ann_eq_span** [IsDomain R] [IsPrincipalIdealRing R] [Finite ι] (hσ : Function.Injective σ) (h : Function.Injective (splitMap σ)) (i : ι) : ∃ ε, ann (σ i) = Submodule.span R {ε} ∧ splitMap σ ε i ≠ 0

theorem **map_eq_smul_idem** [IsDomain R] [DecidableEq ι] (hσ : Function.Injective σ) {i : ι} {x : R ⊗[ℤ] O} (hx : x ∈ ann (σ i)) {R' : Type} [CommRing R'] (f : R →+* R') (h : IsSplitting fun k => f.comp (σ k)) : Algebra.TensorProduct.map f.toIntAlgHom (AlgHom.id ℤ O) x = f (splitMap σ x i) • idem h i

theorem **pb_apply** (e : T ≤ f ⁻¹ᵁ V) (s : Γ(omega G, V)) : pb h e s = (omegaBase h).app T (pullbackSections f (omega G) e s)

theorem **pb_smul** (e : T ≤ f ⁻¹ᵁ V) (a : Γ(S, V)) (s : Γ(omega G, V)) : pb h e (a • s) = f.appLE V T e a • pb h e s

theorem **pb_resM** (e : T ≤ f ⁻¹ᵁ V') (k : V' ≤ V) (s : Γ(omega G, V)) : pb h e (resM _ k s) = pb h (e.trans (f.preimage_mono k)) s

theorem **resM_pb** (e : T ≤ f ⁻¹ᵁ V) (k : T' ≤ T) (s : Γ(omega G, V)) : resM _ k (pb h e s) = pb h (k.trans e) s

theorem **pb_pb** (h' : IsGrpPullback G'' G' g ψ) {k : S'' ⟶ S} {χ : G''.left ⟶ G.left} (h'' : IsGrpPullback G'' G k χ) (hk : g ≫ f = k) (hχ : ψ ≫ φ = χ) {W : S''.Opens} (e : T ≤ f ⁻¹ᵁ V) (e' : W ≤ g ⁻¹ᵁ T) (e'' : W ≤ k ⁻¹ᵁ V) (s : Γ(omega G, V)) : pb h' e' (pb h e s) = pb h'' e'' s

theorem **span_range_pb** (hT : IsAffineOpen T) (hV : IsAffineOpen V) (e : T ≤ f ⁻¹ᵁ V) : Submodule.span Γ(S', T) (Set.range (pb h e)) = ⊤

theorem **pushOm_app** (s : Γ(omega G, V)) : (pushOm h).app V s = pb h (le_refl (f ⁻¹ᵁ V)) s

theorem **pbMod_π** (N : ℕ) (e : T ≤ f ⁻¹ᵁ V) (s : Γ(omega G, V)) : pbMod h N e ((cokernel.π (N • 𝟙 (omega G))).app V s) = (cokernel.π (N • 𝟙 (omega G'))).app T (pb h e s)

theorem **pbMod_resM** (N : ℕ) (e : T ≤ f ⁻¹ᵁ V') (k : V' ≤ V) (u : Γ(modN (omega G) N, V)) : pbMod h N e (resM _ k u) = pbMod h N (e.trans (f.preimage_mono k)) u

theorem **resM_pbMod** (N : ℕ) (e : T ≤ f ⁻¹ᵁ V) (k : T' ≤ T) (u : Γ(modN (omega G) N, V)) : resM _ k (pbMod h N e u) = pbMod h N (k.trans e) u

theorem **pbMod_omegaMap** (a : G ⟶ G) (a' : G' ⟶ G') [IsMonHom a] [IsMonHom a'] (w : a'.left ≫ φ = φ ≫ a.left) (N : ℕ) (e : T ≤ f ⁻¹ᵁ V) (u : Γ(modN (omega G) N, V)) : pbMod h N e ((modNMap (omegaMap a) N).app V u) = (modNMap (omegaMap a') N).app T (pbMod h N e u)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.StableIffCarrierEq

theorem **sepWith_trI** : SepWith A3 A3.trI ↔ A3.fam3.TrISep

theorem **swapCast_one_tmul** {E : Type} [Field E] [NumberField E] {F1 F2 : Subfield E} (h : F1 = F2) (x : E) : swapCast h (1 ⊗ₜ[F1] x) = x ⊗ₜ[F2] 1

theorem **toCx3_reFace** (R : RealDat3 D A3 Ks) (F : Face3 D A3) (Dc Sp Zsc) : R.toCx3 (reFace F Dc Sp Zsc) = R.toCx3 F

theorem **rowForm_add** (A : FamIn D.what₂.LL G Pl0 T D.R.m) (P : Subgroup G) (ψ : P →* D.what₂.LLˣ) (t : T) (x y) : rowForm A P ψ t (x + y) = rowForm A P ψ t x + rowForm A P ψ t y

theorem **rowForm_smul** (A : FamIn D.what₂.LL G Pl0 T D.R.m) (P : Subgroup G) (ψ : P →* D.what₂.LLˣ) (t : T) (c : D.what₂.LL) (x) : rowForm A P ψ t (c • x) = c • rowForm A P ψ t x

theorem **liftForm_add** (A : FamIn D.what₂.LL G Pl0 T D.R.m) (P : Subgroup G) (ψ : P →* D.what₂.LLˣ) (i : Fin D.R.m) (t : T) (x y) : liftForm A P ψ i t (x + y) = liftForm A P ψ i t x + liftForm A P ψ i t y

theorem **liftForm_smul** (A : FamIn D.what₂.LL G Pl0 T D.R.m) (P : Subgroup G) (ψ : P →* D.what₂.LLˣ) (i : Fin D.R.m) (t : T) (c : D.what₂.LL) (x) : liftForm A P ψ i t (c • x) = c • liftForm A P ψ i t x

theorem **sum_mem** (M : Rows.OLat O φ W) {ι : Type} (s : Finset ι) (f : ι → W) (h : ∀ i ∈ s, f i ∈ M.carrier) : ∑ i ∈ s, f i ∈ M.carrier

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.StepZeroTwistReferenceTupleEqId

theorem **ptLaw_stepY** {X : Fields} {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 X.E))} (y : Ys X n) (β : Type) [Fintype β] [DecidableEq β] (Λ : β → LevFam X.E X.K n SG) (θ : β → LevQFam X.E X.K n Q) (pref : DecoF X.E n ST SG Q → Prop) (hy : ∀ N, ∅ ∈ y.ys N) (b : β) (c : ℕ) (h0 : ∀ P : Q, ∀ k ∈ prinCong n P.1 c, ∃ h : k ∈ (θ b P).H, (θ b P).lam ⟨k, h⟩ = 1) : PtLaw (stepY ST y β Λ θ pref) b (fun p : DecoF X.E n ST SG Q => ⟨(p, ∅), hy _⟩) c

theorem **PtLaw.reRep** {E K : Type} [Field E] [NumberField E] [StarRing E] [Field K] {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} {S : GStep E K n} {b : S.β} {pt : DecoF E n ST SG Q → S.D b} {c : ℕ} (h : PtLaw S b pt c) (pref : ∀ b, S.D b → Prop) : PtLaw (S.reRep pref) b pt c

theorem **valL_near** (hp : PtLaw S b pt c) (hφ : φ ∈ S.Fi emb σ lam) {x : GL (Fin n) (AfE D.host.F D.host.E)} {p p' : DecoF D.host.E n ST SG Q} (h : Near c x p) (h' : Near c x p') : valL S emb σ b (pt p') φ = valL S emb σ b (pt p) φ

theorem **adFun_eq** (hG : V.gram = 1) (hx : x₀ ∈ UAd D.sp.cm n) (hp : PtLaw S b pt c) (hφ : φ ∈ S.Fi emb σ lam) {g : GA D V} {p : DecoF D.host.E n ST SG Q} (h : Near c (finN V hn g ∗ x₀) p) : adFun θ V hn t emb σ S b pt c x₀ l φ g = l (lT (Gr emb σ n) (infN θ V hn t g) (valL S emb σ b (pt p) φ))

theorem **adFun_congr** (hG : V.gram = 1) (hx : x₀ ∈ UAd D.sp.cm n) {x₁ : GL (Fin n) (AfE D.host.F D.host.E)} (hx' : x₁ ∈ UAd D.sp.cm n) {p₀ : DecoF D.host.E n ST SG Q} (h₀ : Near c x₀ p₀) (h₁ : Near c x₁ p₀) (hp : PtLaw S b pt c) (hφ : φ ∈ S.Fi emb σ lam) : adFun θ V hn t emb σ S b pt c x₀ l φ = adFun θ V hn t emb σ S b pt c x₁ l φ

theorem **isAutFunDef_adFun** (hθ : IsInfFam θ) (hG : V.gram = 1) (he : ∀ τ, embAt σ emb τ = tauE D (t τ)) (hx : x₀ ∈ UAd D.sp.cm n) (hp : PtLaw S b pt c) (hφ : φ ∈ S.Fi emb σ lam) : IsAutFunDef V (adFun θ V hn t emb σ S b pt c x₀ l φ)

theorem **eq_zero_of_adFun** (hθ : IsInfFam θ) (hG : V.gram = 1) (hx : x₀ ∈ UAd D.sp.cm n) (hp : PtLaw S b pt c) (hφ : φ ∈ S.Fi emb σ lam) (h0 : adFun θ V hn t emb σ S b pt c x₀ (LinearMap.proj (fr0 emb σ n)) φ = 0) {x : GL (Fin n) (AfE D.host.F D.host.E)} (hx' : x ∈ UAd D.sp.cm n) {p : DecoF D.host.E n ST SG Q} (h : Near c x p) : valL S emb σ b (pt p) φ = 0

theorem **adForm_val** (hθ : IsInfFam θ) (hG : V.gram = 1) (hs : V.SigOK) (hd : V.IsDefinite) (he : ∀ τ, embAt σ emb τ = tauE D (t τ)) (hx : x₀ ∈ UAd D.sp.cm n) (hp : PtLaw S b pt c) (hφ : φ ∈ S.Fi emb σ lam) : (adForm θ hn t emb σ l lam hθ hG hs hd he hx hp φ).1 = adFun θ V hn t emb σ S b pt c x₀ l φ

theorem **col_mem** {R : Ranks} {ι : Type} [Fintype ι] {hc : ι → HCParam R} (C : ColumnChain R ι hc) (V : ∀ i, Submodule ℂ (C.cls i → C.Y i → ℂ)) (hV : ∀ i < R.m, ∀ φ ∈ V i, C.K i φ ∈ V (i + 1)) {φ} (hφ : φ ∈ V 0) : C.col φ ∈ V R.m

theorem **pairB_extV** {ι : Type} [Fintype ι] {n : ℕ} (sg : ι → DomWeight n) : pairB sg (extV ℂ sg).1 (extV ℂ sg).1 = 1

theorem **pairB_topVec** {ι : Type} [Fintype ι] {n : ℕ} (sg : ι → DomWeight n) : pairB sg (topVec sg) (topVec sg) = (Module.finrank ℂ (VV ℂ sg) : ℂ)

theorem **zero_succ_ne** (R : Ranks) : 0 + 1 ≠ R.m

theorem **Ok.gram_X** (h : I.Ok) (j : ℕ) : (B.sp.X j).gram = 1

theorem **Ok.exists_isPtOf** (h : I.Ok) (u : B.Word) (a : I.ι) {n : ℕ} (p : DecoF B.host.E n ((B.primes X).vT u) (B.primes X).vGs (I.Q a)) : ∃ x, IsPtOf B.sp.cm x p

theorem **Ok.xTop_spec** (h : I.Ok) (u : B.Word) (a : I.ι) : IsPtOf B.sp.cm (h.xTop u a) (I.baseTop u a)

theorem **ptLaw_zero** (u : B.Word) (b : Idx I.sel (I.tup I.mu0) (I.idx I.mu0) 0) (c : ℕ) : PtLaw (I.stp u I.mu0 0) b (I.ptAt u I.mu0 (zero_succ_ne B.R) b) c

theorem **ptLaw_top** (u : B.Word) (a : I.ι) : PtLaw (I.stp u a B.R.m) (idxTop I.sel (I.tup a) (I.idx a)) (I.ptAt u a (Nat.succ_ne_self _) (idxTop I.sel (I.tup a) (I.idx a))) (I.depTop a)

theorem **embAt_eq** (τ : Tau B.R) : embAt B.embLL B.fields.emb τ = tauE (I.rec7 l Z) (eT l Z τ)

theorem **src_mem** (u : B.Word) (φ : Z.W.Mu u) : src l Z u φ ∈ (I.stp u I.mu0 0).Fi B.fields.emb B.embLL (lamT (B.hcW u) 0)

theorem **carries_of_kFi** (l0 : I.Law0 l) (hK : Formalism.MainNotions.OrbitPlaces.KFi (I.rec7 l Z) (I.tw l l0 Z)) (u : B.Word) (a : I.ι) (ha : DV.Used (I.rec7 l Z) a) : Carries l u a

theorem **twistedIdempotent_mem_fixed** (u : B.Word) (a : I.ι) {φ : (I.stp u I.mu0 0).Cl → Fr B.fields.emb B.embLL (B.R.N 0) → ℂ} (hφ : φ ∈ (I.stp u I.mu0 0).Fi B.fields.emb B.embLL (lamT (B.hcW u) 0)) : (I.cols l).e0Tw u a φ ∈ (I.stp u a 0).Fi B.fields.emb B.embLL (lamT (B.hcW u) 0)

theorem **topF_mem** {u : B.Word} {a : I.ι} (hC : Carries l u a) (φ : Z.W.Mu u) : topF l Z u a φ ∈ (I.stp u a B.R.m).Fi B.fields.emb B.embLL (lamT (B.hcW u) B.R.m)

theorem **colAFw_zero** (w) (u μ) : colAFw h l Z hθ w u μ 0 = colZero h l Z hθ u

theorem **colAFw_top** (w) (u μ) : colAFw h l Z hθ w u μ B.R.m = colTop h l Z hθ u μ w

theorem **colAFw_mid** (w) (u μ) {j : ℕ} (h0 : j ≠ 0) (hm : j ≠ B.R.m) : colAFw h l Z hθ w u μ j = 0

theorem **colAF_zero** (u μ) : colAF h l Z hθ u μ 0 = colZero h l Z hθ u

theorem **colAF_top** (u μ) : colAF h l Z hθ u μ B.R.m = colTop h l Z hθ u μ (topVec B.wts.sgTrue)

theorem **colAF_mid** (u μ) {j : ℕ} (h0 : j ≠ 0) (hm : j ≠ B.R.m) : colAF h l Z hθ u μ j = 0

theorem **perIn_sgT** (u : B.Word) (a : I.ι) : (I.perIn u a).sgT = B.wts.sgTrue

theorem **colZero_apply** (u : B.Word) (φ : Z.W.Mu u) (g : Formalism.L2.Theta.GA (I.rec7 l Z) ((I.rec7 l Z).sp.X 0)) (p : ChPt (Pr

theorem **colTop_apply** (u : B.Word) (a : I.ι) (w) (φ : Z.W.Mu u) (hφ : topF l Z u a φ ∈ (I.stp u a B.R.m).Fi B.fields.emb B.embLL (lamT (B.hcW u) B.R.m)) (g : Formalism.L2.Theta.GA (I.rec7 l Z) ((I.rec7 l Z).sp.X B.R.m)) {x₀ : GL (Fin (B.R.N B.R.m)) (AfE B.host.F B.host.E)} (hx : x₀ ∈ UAd B.sp.cm _) (h₀ : Near (I.depTop a) x₀ (I.baseTop u a)) (p : ChPt (Pr

theorem **colTop_apply_of_carries** {u : B.Word} {a : I.ι} (hC : Carries l u a) (w) (φ : Z.W.Mu u) (g : Formalism.L2.Theta.GA (I.rec7 l Z) ((I.rec7 l Z).sp.X B.R.m)) {x₀ : GL (Fin (B.R.N B.R.m)) (AfE B.host.F B.host.E)} (hx : x₀ ∈ UAd B.sp.cm _) (h₀ : Near (I.depTop a) x₀ (I.baseTop u a)) (p : ChPt (Pr

theorem **colZero_translate** (u : B.Word) {x₀ : GL (Fin (B.R.N 0)) (AfE B.host.F B.host.E)} (hx : x₀ ∈ UAd B.sp.cm _) : ∃ k : Formalism.L2.Theta.GA (I.rec7 l Z) ((I.rec7 l Z).sp.X 0), finX l Z (Nat.zero_le _) k = x₀ ∧ infX l Z θ (Nat.zero_le _) k = 1 ∧ ∀ (φ : Z.W.Mu u) (g : Formalism.L2.Theta.GA (I.rec7 l Z) ((I.rec7 l Z).sp.X 0)) (p : ChPt (Pr

theorem **kStd_mem** (hPr : (B.primes X).Ok) (u : B.Word) (n : ℕ) : kStd X u n ∈ UAd B.sp.cm n

theorem **idx_zero_eq** (u : B.Word) (a : I.ι) (b : Idx I.sel (I.tup a) (I.idx a) 0) (c : (I.stp u a 0).Cl) : c.1 = b

theorem **colZero_injective** (u : B.Word) : Function.Injective (colZero h l Z hθ u)

theorem **colAF_zero_injective** (u μ) : Function.Injective (colAF h l Z hθ u μ 0)

theorem **stepZeroTwist_reference_tuple_eq_id** (u : B.Word) (φ : (I.stp u I.mu0 0).Cl → Fr B.fields.emb B.embLL (B.R.N 0) → ℂ) : (I.cols l).e0Tw u I.mu0 φ = φ

theorem **perCol_eq** (u : B.Word) (a : (I.cols l).Tup) (ha : a ≠ I.mu0) (φ : Z.W.Mu u) (F : Z.W.Amb a) : (I.rec7 l Z).what₂.perCol u a φ F = (I.per l u a).Ptil (topF l Z u a φ) ((Z.W.eMtTw u a).symm F)

theorem **periodColumn_eq_reference** (u : B.Word) (φ : Z.W.Mu u) (F : Z.W.Amb (I.cols l).mu0) : (I.rec7 l Z).what₂.perCol u I.mu0 φ F = (I.per l u I.mu0).Ptil (topF l Z u I.mu0 φ) ((Z.W.eMt0 u).symm F)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SubsetTwistExceptionalPlaces

theorem **OverSet.mono** {D : D7} {S : Finset (PlE D.host)} {s s' : Finset ℕ} (h : OverSet D S s) (hs : s ⊆ s') : OverSet D S s'

theorem **OverSet.subset** {D : D7} {S S' : Finset (PlE D.host)} {s : Finset ℕ} (h : OverSet D S s) (hs : ∀ p ∈ s, placesOver D.host.E p ⊆ ↑S') : S ⊆ S'

theorem **OverSet.subset_twistExceptionalPlaces** {D : D7} {S : Finset (PlE D.host)} {s low : Finset ℕ} (h : OverSet D S s) (hs : s ⊆ low) {F : TwFrameC D} (T : TwistFrameLowIndexCompatibility D F low) (i : D.ntw.Tw) : S ⊆ D.ntw.S0 i

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SubsingletonRankZero

theorem **logNorm_infId** (u : (Cinf E)ˣ) : logNorm E (infId E u) = ∑ w, 2 * Real.log ‖(u : Cinf E) w‖

theorem **expo_absC** (s : ℂ) (τ : E →+* ℂ) : expo (absC E s) τ = s

theorem **infChar_absC** (s : ℂ) (τ : E →+* ℂ) : (CG.ofChar (absC E s)).infChar τ = {s}

theorem **isRegAlg_absC** (s : ℂ) : (CG.ofChar (absC E s)).IsRegAlg ↔ ∃ n : ℤ, s = n

theorem **not_isRegAlg_half** : ¬ (CG.ofChar (absC E (1 / 2))).IsRegAlg

theorem **absC_class_injective** : Function.Injective fun s : ℂ => CG.ofChar (absC E s)

theorem **infinite_rank_one** : Infinite (CG E 1)

theorem **twist_absC_ne** {π : CG E a} (ha : 0 < a) {τ : E →+* ℂ} {m : Multiset ℂ} (h : π.HasInfChar τ m) {s : ℂ} (hs : s ≠ 0) : π.twist (absC E s) ≠ π

theorem **subsingleton_rank_zero** : Subsingleton (CG E 0)

theorem **chiDet_notMem** (ha : 2 ≤ a) (π : CuspSub E a) (χ : HChar E) : mulChi a χ (oneFn E a) ∉ π.V

theorem **isCsd_absC** (s : ℂ) : (CG.ofChar (absC L s)).IsCsd ↔ s = 0

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.SumMulEqZeroOfSumToReal

theorem **mem_balForms** {β : LinearMap.BilinForm ℤ X} : β ∈ balForms bar X ↔ (∀ y z, β y z = β z y) ∧ ∀ (a : O) y z, β (a • y) z = β y (bar a • z)

theorem **mem_realForms** {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} : B ∈ realForms bar X ↔ (∀ y z, B y z = B z y) ∧ ∀ (a : O) y z, B (a • y) z = B y (bar a • z)

theorem **balForms_eq_levelForms** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (X : Type) [AddCommGroup X] [Module (𝒪 E) X] : balForms H.bar X = levelForms H 1 X

theorem **mem_realForms_iff_symB** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) {X : Type} [AddCommGroup X] [Module (𝒪 E) X] {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} : B ∈ realForms H.bar X ↔ B ∈ Formalism.MainNotions.ModuliIntegralCharts.SymB (fun x : E => IsCMField.complexConj E x) X

theorem **mem_balForms_int** {X : Type} [AddCommGroup X] {β : LinearMap.BilinForm ℤ X} : β ∈ balForms (id : ℤ → ℤ) X ↔ ∀ y z, β y z = β z y

@[simp] theorem **pull_apply** (r : X →ₗ[O] X₁) (β : LinearMap.BilinForm ℤ X₁) (a b : X) : pull r β a b = β (r a) (r b)

theorem **pull_mem** (r : X →ₗ[O] X₁) {β : LinearMap.BilinForm ℤ X₁} (h : β ∈ balForms bar X₁) : pull r β ∈ balForms bar X

@[simp] theorem **toReal_apply** (β : LinearMap.BilinForm ℤ X) (a b : X) : toReal β a b = (β a b : ℝ)

theorem **toReal_mem** {β : LinearMap.BilinForm ℤ X} (h : β ∈ balForms bar X) : toReal β ∈ realForms bar X

theorem **sum_mul_eq_zero_of_sum_toReal** {L ι : Type} [AddCommGroup L] [Fintype ι] {j : L →ₗ[ℤ] LinearMap.BilinForm ℤ X} (hj : Function.Injective j) (ℓ : L →ₗ[ℤ] ℤ) {β : ι → L} {c : ι → ℝ} (h : ∑ i, c i • toReal (j (β i)) = 0) : ∑ i, c i * (ℓ (β i) : ℝ) = 0

@[simp] theorem **pullR_apply** (r : X →ₗ[O] X₁) (B : X₁ →ₗ[ℤ] X₁ →ₗ[ℤ] ℝ) (a b : X) : pullR r B a b = B (r a) (r b)

theorem **IsPosDef.nonneg** {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (h : IsPosDef B) (y : X) : 0 ≤ B y y

theorem **isPosDef_iff_basis** {ι : Type} [Fintype ι] (b : Module.Basis ι ℤ X) (B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) : IsPosDef B ↔ ∀ c : ι → ℝ, c ≠ 0 → 0 < ∑ i, ∑ j, c i * c j * B (b i) (b j)

theorem **mem_realForms_of_mem_psdCone** {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (h : B ∈ psdCone bar X) : B ∈ realForms bar X

theorem **nonneg_of_mem_psdCone** {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (h : B ∈ psdCone bar X) (y : X) : 0 ≤ B y y

theorem **IsPosDef.mem_psdCone** [Module.Projective O X] {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (h : IsPosDef B) (hB : B ∈ realForms bar X) : B ∈ psdCone bar X

theorem **mem_psdCone_of_comp** {r : X →ₗ[O] X₁} (hr : Function.Surjective r) {B₁ : X₁ →ₗ[ℤ] X₁ →ₗ[ℤ] ℝ} (h : B₁ ∈ psdCone bar X₁) {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (hB : ∀ a b, B a b = B₁ (r a) (r b)) : B ∈ psdCone bar X

theorem **pullR_mem_psdCone** {r : X →ₗ[O] X₁} (hr : Function.Surjective r) {B₁ : X₁ →ₗ[ℤ] X₁ →ₗ[ℤ] ℝ} (h : B₁ ∈ psdCone bar X₁) : pullR r B₁ ∈ psdCone bar X

theorem **zero_mem_psdCone** : (0 : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) ∈ psdCone bar X

theorem **smul_mem_psdCone** {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (h : B ∈ psdCone bar X) {t : ℝ} (ht : 0 ≤ t) : t • B ∈ psdCone bar X

theorem **isPartOfBasis_map** {v : Fin m → L} (h : IsPartOfBasis v) (e : L ≃ₗ[ℤ] L') : IsPartOfBasis fun i => e (v i)

theorem **isPartOfBasis_comp** {v : Fin m → L} (h : IsPartOfBasis v) (s : Fin m' ↪ Fin m) : IsPartOfBasis fun i => v (s i)

@[simp] theorem **evalR_toReal** (β : LinearMap.BilinForm ℤ X) : evalR φ (toReal β) = φ.evalZ β

theorem **evalR_pullR** {r : X →ₗ[O] X₁} (hr : Function.Surjective r) (B : X₁ →ₗ[ℤ] X₁ →ₗ[ℤ] ℝ) : evalR φ (pullR r B) = evalR φ B

theorem **evalZ_pull** {r : X →ₗ[O] X₁} (hr : Function.Surjective r) (β : LinearMap.BilinForm ℤ X₁) : φ.evalZ (pull r β) = φ.evalZ β

@[simp] theorem **evalR_zero** : evalR φ (0 : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) = 0

theorem **linearOn_iff** (γ : Fin m → LinearMap.BilinForm ℤ X) : φ.LinearOn γ ↔ ∀ c : Fin m → ℝ, (∀ i, 0 ≤ c i) → evalR φ (∑ i, c i • toReal (γ i)) = ∑ i, c i * φ.evalZ (γ i)

theorem **linearOn_comp** {γ : Fin m → LinearMap.BilinForm ℤ X} (h : φ.LinearOn γ) (s : Fin m' ↪ Fin m) : φ.LinearOn fun i => γ (s i)

theorem **linearOn_pull_iff** {r : X →ₗ[O] X₁} (hr : Function.Surjective r) {γ : Fin m → LinearMap.BilinForm ℤ X₁} : (φ.LinearOn fun i => pull r (γ i)) ↔ φ.LinearOn γ

theorem **isStrict_iff_isStrictAt** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (φ : InvFun (𝓞 E)) {X : Type} [AddCommGroup X] [Module (𝓞 E) X] [Module.Finite (𝓞 E) X] (γ : Fin m → LinearMap.BilinForm ℤ X) : IsStrict H.bar φ γ ↔ IsStrictAt H φ γ

theorem **IsStrict.reindex** {γ : Fin m → LinearMap.BilinForm ℤ X} (h : IsStrict bar φ γ) (s : Fin m' ≃ Fin m) : IsStrict bar φ fun i => γ (s i)

theorem **IsStrict.of_pull** {r : X →ₗ[O] X₁} (hr : Function.Surjective r) {γ : Fin m → LinearMap.BilinForm ℤ X₁} (h : IsStrict bar φ fun i => pull r (γ i)) : IsStrict bar φ γ

@[simp] theorem **pull_symm_pull** (e : X ≃ₗ[O] X₁) (β : LinearMap.BilinForm ℤ X₁) : pull e.symm.toLinearMap (pull e.toLinearMap β) = β

@[simp] theorem **pull_pull_symm** (e : X ≃ₗ[O] X₁) (β : LinearMap.BilinForm ℤ X) : pull e.toLinearMap (pull e.symm.toLinearMap β) = β

theorem **linearIndependent** (h : IsCone bar φ γ) : LinearIndependent ℤ γ

theorem **injective** (h : IsCone bar φ γ) : Function.Injective γ

theorem **comp** (h : IsCone bar φ γ) (s : Fin m' ↪ Fin m) (hs : IsStrict bar φ fun i => γ (s i)) : IsCone bar φ fun i => γ (s i)

theorem **reindex** (h : IsCone bar φ γ) (s : Fin m' ≃ Fin m) : IsCone bar φ fun i => γ (s i)

theorem **pull_equiv** {γ : Fin m → LinearMap.BilinForm ℤ X₁} (h : IsCone bar φ γ) (e : X ≃ₗ[O] X₁) : IsCone bar φ fun i => pull e.toLinearMap (γ i)

theorem **evalZ_smul** (h : IsCone bar φ γ) (M : ℕ) (i : Fin m) : φ.evalZ ((M : ℤ) • γ i) = M * φ.evalZ (γ i)

theorem **linearOn_smul** (h : IsCone bar φ γ) (M : ℕ) : φ.LinearOn fun i => (M : ℤ) • γ i

theorem **isStrict_smul** (h : IsCone bar φ γ) {M : ℕ} (hM : M ≠ 0) : IsStrict bar φ fun i => (M : ℤ) • γ i

theorem **IsCone.isPartOfBasis_level** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {φ : InvFun (𝓞 E)} {X : Type} [AddCommGroup X] [Module (𝓞 E) X] [Module.Finite (𝓞 E) X] {γ : Fin m → LinearMap.BilinForm ℤ X} (h : IsCone H.bar φ γ) {M : ℕ} (hM : M ≠ 0) {δ : Fin m → levelForms H M X} (hδ : ∀ i, (δ i : LinearMap.BilinForm ℤ X) = (M : ℤ) • γ i) : IsPartOfBasis δ

theorem **IsPolFan.mono** {q q' : ℕ} (h : IsPolFan bar q φ) (hq : q' ≤ q) : IsPolFan bar q' φ

theorem **NoSelfInt.mono** {q q' : ℕ} (h : NoSelfInt bar q φ) (hq : q' ≤ q) : NoSelfInt bar q' φ

theorem **IsPolFanOn.evalZ_nonneg** (h : IsPolFanOn bar φ X) {β : LinearMap.BilinForm ℤ X} (hβ : β ∈ balForms bar X) (hp : ∀ y, 0 ≤ β y y) : 0 ≤ φ.evalZ β

theorem **IsPolFanOn.exists_nat** (h : IsPolFanOn bar φ X) {β : LinearMap.BilinForm ℤ X} (hβ : β ∈ balForms bar X) (hp : ∀ y, 0 ≤ β y y) : ∃ n : ℕ, φ.evalZ β = n

theorem **IsPolFanOn.evalR_nonneg** (h : IsPolFanOn bar φ X) {B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ} (hB : B ∈ psdCone bar X) : 0 ≤ evalR φ B

theorem **evalR_smul_of_linearOn** {γ : Fin m → LinearMap.BilinForm ℤ X} (hγ : φ.LinearOn γ) {c : Fin m → ℝ} (hc : ∀ i, 0 ≤ c i) {t : ℝ} (ht : 0 ≤ t) : evalR φ (t • ∑ i, c i • toReal (γ i)) = t * evalR φ (∑ i, c i • toReal (γ i))

theorem **IsPolFanOn.of_equiv** (h : IsPolFanOn bar φ X) (e : X ≃ₗ[O] X₁) : IsPolFanOn bar φ X₁ where

theorem **NoEquivRays.of_pull_equiv** {γ : Fin m → LinearMap.BilinForm ℤ X₁} (e : X ≃ₗ[O] X₁) (h : NoEquivRays O fun i => pull e.toLinearMap (γ i)) : NoEquivRays O γ

theorem **isPolFanOn_of_subsingleton** [Subsingleton X] : IsPolFanOn bar φ X

theorem **isPolFan_zero** [IsDomain O] : IsPolFan bar 0 φ

theorem **noSelfInt_zero** [IsDomain O] : NoSelfInt bar 0 φ

theorem **IsFine.comp** (h : IsFine N γ) (s : Fin m' → Fin m) : IsFine N fun i => γ (s i)

theorem **IsFine.exists_min** (h : IsFine N γ) (y : X) : ∃ μ₀ : X, ∀ c : Fin m → ℝ, (∀ i, 0 ≤ c i) → ∀ μ, ∑ i, c i * (γ i (y + N • μ₀) (y + N • μ₀) : ℝ) ≤ ∑ i, c i * (γ i (y + N • μ) (y + N • μ) : ℝ)

theorem **IsFine.sum** (h : IsFine N γ) (a : Fin m' → Fin m → ℕ) : IsFine N fun j => ∑ i, a j i • γ i

theorem **IsFine.smul** (h : IsFine N γ) (M : ℕ) : IsFine N fun i => (M : ℤ) • γ i

theorem **isFine_pull_iff** {O : Type} [CommRing O] [Module O X] [Module O X₁] {r : X →ₗ[O] X₁} (hr : Function.Surjective r) {γ : Fin m → LinearMap.BilinForm ℤ X₁} : (IsFine N fun i => pull r (γ i)) ↔ IsFine N γ

theorem **isFine_one** (h : ∀ i y, 0 ≤ γ i y y) : IsFine 1 γ

theorem **isFine_single** {β : LinearMap.BilinForm ℤ X} (h : ∀ y, 0 ≤ β y y) (N : ℕ) : IsFine N ![β]

@[simp] theorem **sqForm_apply** (l : X →ₗ[ℤ] ℤ) (a b : X) : sqForm l a b = l a * l b

@[simp] theorem **mulForm_apply** (a b : ℤ) : mulForm a b = a * b

theorem **real_apply_int** (B : ℤ →ₗ[ℤ] ℤ →ₗ[ℤ] ℝ) (a b : ℤ) : B a b = B 1 1 * (a * b)

theorem **eq_smul_mulForm** (β : LinearMap.BilinForm ℤ ℤ) : β = β 1 1 • mulForm

theorem **evalR_minVal_int** {B : ℤ →ₗ[ℤ] ℤ →ₗ[ℤ] ℝ} (h : 0 ≤ B 1 1) : evalR (InvFun.minVal ℤ) B = B 1 1

theorem **evalZ_minVal_int** {β : LinearMap.BilinForm ℤ ℤ} (h : 0 ≤ β 1 1) : (InvFun.minVal ℤ).evalZ β = β 1 1

theorem **isCone_mulForm** : IsCone id (InvFun.minVal ℤ) ![mulForm]

theorem **isCone_empty_int** {m : ℕ} [IsEmpty (Fin m)] (γ : Fin m → LinearMap.BilinForm ℤ ℤ) : IsCone id (InvFun.minVal ℤ) γ

theorem **IsCone.eq_mulForm** {φ : InvFun ℤ} {m : ℕ} {γ : Fin m → LinearMap.BilinForm ℤ ℤ} (h : IsCone id φ γ) (i : Fin m) : γ i = mulForm

theorem **toReal_mulForm_mem_psdCone** : toReal mulForm ∈ psdCone (id : ℤ → ℤ) ℤ

theorem **isPolFanOn_minVal_int** : IsPolFanOn id (InvFun.minVal ℤ) ℤ where

theorem **subsingleton_or_equiv_int** (X : Type) [AddCommGroup X] [Module ℤ X] [Module.Finite ℤ X] [Module.Projective ℤ X] (h : Module.finrank ℤ X ≤ 1) : Subsingleton X ∨ Nonempty (ℤ ≃ₗ[ℤ] X)

theorem **isPolFan_minVal_one** : IsPolFan id 1 (InvFun.minVal ℤ)

theorem **noSelfInt_int_one** (φ : InvFun ℤ) : NoSelfInt id 1 φ

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.T2SpaceAdelicGeneralLinearGroupNumberField

theorem **continuous_unip** {r : ℕ} : Continuous (unip : (BlockIx a r → Ad E) → GA E a)

theorem **splitCentre_eq_scalar** (t : ℝ) : splitCentre E a t = GeneralLinearGroup.scalar (Fin a) (infId E (expU fun _ : InfinitePlace E => (t : ℂ)))

theorem **splitCentre_comm** (t : ℝ) (g : GA E a) : splitCentre E a t * g = g * splitCentre E a t

theorem **splitCentre_add** (s t : ℝ) : splitCentre E a (s + t) = splitCentre E a s * splitCentre E a t

theorem **splitCentre_zero** : splitCentre E a 0 = 1

theorem **splitCentre_intCast** (n : ℤ) : splitCentre E a n = splitCentre E a 1 ^ n

theorem **det_splitCentre_one** : GeneralLinearGroup.det (splitCentre E a 1) = eIdele E ^ a

theorem **ratGL_mem_latticeGL** (γ : GL (Fin a) E) : ratGL E a γ ∈ latticeGL E a

theorem **zpow_mem_latticeGL** (n : ℤ) : splitCentre E a 1 ^ n ∈ latticeGL E a

theorem **mem_latticeGL_iff** {g : GA E a} : g ∈ latticeGL E a ↔ ∃ (γ : GL (Fin a) E) (n : ℤ), ratGL E a γ * splitCentre E a 1 ^ n = g

theorem **leftInv_latticeGL_iff** {X : Type*} {φ : GA E a → X} : LeftInv (latticeGL E a) φ ↔ (∀ γ g, φ (ratGL E a γ * g) = φ g) ∧ ∀ (n : ℤ) g, φ (splitCentre E a n * g) = φ g

theorem **leftInv_latticeGL** {X : Type*} {φ : GA E a → X} (h : ∀ γ g, φ (ratGL E a γ * g) = φ g) (h' : ∀ t g, φ (splitCentre E a t * g) = φ g) : LeftInv (latticeGL E a) φ

theorem **integrableOn_unip** (hc : Continuous φ) (hl : ∀ γ g, φ (ratGL E a γ * g) = φ g) {r : ℕ} [MeasurableSpace (BlockIx a r → Ad E)] [BorelSpace (BlockIx a r → Ad E)] (ν : Measure (BlockIx a r → Ad E)) [ν.IsAddHaarMeasure] {𝓕 : Set (BlockIx a r → Ad E)} (h𝓕 : IsAddFundamentalDomain (ratN E a r) 𝓕 ν) (g : GA E a) : IntegrableOn (fun x => φ (unip x * g)) 𝓕 ν

theorem **mem_cuspCont** : φ ∈ cuspCont E a ↔ Continuous φ ∧ (∀ γ g, φ (ratGL E a γ * g) = φ g) ∧ (∀ t g, φ (splitCentre E a t * g) = φ g) ∧ IsCuspidal φ

theorem **leftInv_of_mem_cuspCont** (h : φ ∈ cuspCont E a) : LeftInv (latticeGL E a) φ

theorem **mul_right_mem_cuspCont** (h : φ ∈ cuspCont E a) (k : GA E a) : (fun g => φ (g * k)) ∈ cuspCont E a

theorem **mem_cuspCont_of_isCuspForm** {φ : Sm E a} (h : IsCuspForm φ) (h' : ∀ t g, φ.1 (splitCentre E a t * g) = φ.1 g) : φ.1 ∈ cuspCont E a

theorem **mem_cuspCont_of_le_one** (ha : a ≤ 1) : φ ∈ cuspCont E a ↔ Continuous φ ∧ (∀ γ g, φ (ratGL E a γ * g) = φ g) ∧ ∀ t g, φ (splitCentre E a t * g) = φ g

theorem **one_mem_cuspCont_iff** : (fun _ => (1 : ℂ)) ∈ cuspCont E a ↔ a ≤ 1

theorem **mem_classes** : F ∈ classes V μ ↔ ∃ φ ∈ V, lift (latticeGL E a) F =ᵐ[μ] φ.1

theorem **toL2_mem_classes** {φ : Sm E a} (hφ : φ ∈ V) (h : MemL2Quotient (latticeGL E a) μ φ.1) : h.toL2 ∈ classes V μ

theorem **memL2_of_ae_eq** (hφ : LeftInv (latticeGL E a) φ) (e : lift (latticeGL E a) F =ᵐ[μ] φ) : MemL2Quotient (latticeGL E a) μ φ

theorem **mem_L2cusp_of_ae_eq** (hφ : φ ∈ cuspCont E a) (e : lift (latticeGL E a) F =ᵐ[μ] φ) : F ∈ L2cusp μ

theorem **toL2_mem_cuspidal** (h : MemL2Quotient (latticeGL E a) μ φ) (hφ : φ ∈ cuspCont E a) : h.toL2 ∈ L2cusp μ

theorem **L2cusp_le** {U : Submodule ℂ (L2 (latticeGL E a) μ)} (hU : IsClosed (U : Set (L2 (latticeGL E a) μ))) (h : ∀ φ ∈ cuspCont E a, ∀ h : MemL2Quotient (latticeGL E a) μ φ, h.toL2 ∈ U) : L2cusp μ ≤ U

theorem **closure_classes_le_L2cusp** (h : ∀ φ ∈ V, IsCuspForm φ) (h' : ∀ φ ∈ V, ∀ t g, φ.1 (splitCentre E a t * g) = φ.1 g) : (classes V μ).topologicalClosure ≤ L2cusp μ

theorem **classes_le_L2cusp** (h : ∀ φ ∈ V, IsCuspForm φ) (h' : ∀ φ ∈ V, ∀ t g, φ.1 (splitCentre E a t * g) = φ.1 g) : classes V μ ≤ L2cusp μ

theorem **isClosed_L2cusp** : IsClosed (L2cusp μ : Set (L2 (latticeGL E a) μ))

theorem **memL2_iff_isSqInt** (hc : Continuous φ) (h : ∀ γ g, φ (ratGL E a γ * g) = φ g) (h' : ∀ t g, φ (splitCentre E a t * g) = φ g) : MemL2Quotient (latticeGL E a) μ φ ↔ IsSqInt φ

theorem **memL2_of_isAutForm** {φ : Sm E a} (h : IsAutForm φ) (h' : IsSqInt φ.1) : MemL2Quotient (latticeGL E a) μ φ.1

theorem **toL2_ne_zero** (h : MemL2Quotient (latticeGL E a) μ φ) (hc : Continuous φ) (h0 : φ ≠ 0) : h.toL2 ≠ 0

theorem **lift_toL2ₗ** (h : ∀ φ ∈ V, MemL2Quotient (latticeGL E a) μ φ.1) (φ : V) : lift (latticeGL E a) (toL2ₗ μ V h φ) =ᵐ[μ] φ.1.1

theorem **range_toL2ₗ** (h : ∀ φ ∈ V, MemL2Quotient (latticeGL E a) μ φ.1) : LinearMap.range (toL2ₗ μ V h) = classes V μ

theorem **toL2ₗ_injective** (h : ∀ φ ∈ V, MemL2Quotient (latticeGL E a) μ φ.1) (hc : ∀ φ ∈ V, Continuous φ.1) : Function.Injective (toL2ₗ μ V h)

theorem **isSubrep_L2cusp** : IsSubrep (R (latticeGL E a) μ) (L2cusp μ)

theorem **isUnitaryRep_Rcusp** : IsUnitaryRep (Rcusp μ)

theorem **R_splitCentre** (t : ℝ) (hF : F ∈ L2cusp μ) : R (latticeGL E a) μ (splitCentre E a t) F = F

theorem **Rcusp_splitCentre** (t : ℝ) : Rcusp μ (splitCentre E a t) = 1

theorem **R_mem_classes** {g : GA E a} (hg : ∀ φ ∈ V, rt E a g φ ∈ V) (hF : F ∈ classes V μ) : R (latticeGL E a) μ g F ∈ classes V μ

theorem **R_toL2ₗ** (h : ∀ φ ∈ V, MemL2Quotient (latticeGL E a) μ φ.1) {g : GA E a} (φ : V) (hg : rt E a g φ.1 ∈ V) : R (latticeGL E a) μ g (toL2ₗ μ V h φ) = toL2ₗ μ V h ⟨_, hg⟩

theorem **mul_comm_one** (g h : GA E 1) : g * h = h * g

theorem **memL2_of_rank_one** {φ : GA E 1 → ℂ} (hc : Continuous φ) (h : LeftInv (latticeGL E 1) φ) : MemL2Quotient (latticeGL E 1) μ φ

theorem **exists_one_mem_L2cusp** : ∃ F ∈ L2cusp μ, F ≠ 0 ∧ lift (latticeGL E 1) F =ᵐ[μ] fun _ => 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.T2SpaceComplexPointsOfProjVariety

theorem **of_algebraMap** (a : F) : of σ (algebraMap F A a) = algebraMap ℂ (CxExt σ A) (σ a)

theorem **of_smul** (a : F) (x : A) : of σ (a • x) = σ a • of σ x

theorem **of_injective** : Function.Injective (of σ : A → CxExt σ A)

private theorem **complex_smul_of_eq** (c : ℂ) (a : A) : c • of σ a = (letI

@[elab_as_elim] theorem **induction** {P : CxExt σ A → Prop} (x : CxExt σ A) (smul : ∀ (c : ℂ) (a : A), P (c • of σ a)) (add : ∀ x y, P x → P y → P (x + y)) : P x

@[simp] theorem **lift_of** {B : Type} [CommRing B] [Algebra ℂ B] (φ : A →+* B) (h : ∀ a, φ (algebraMap F A a) = algebraMap ℂ B (σ a)) (a : A) : lift φ h (of σ a) = φ a

@[simp] theorem **map_of** (f : A →ₐ[F] A') (a : A) : map σ f (of σ a) = of σ (f a)

theorem **map_injective** {f : A →ₐ[F] A'} (hf : Function.Injective f) : Function.Injective (map σ f)

theorem **exists_components** (f : A →ₐ[F] A') (x : CxExt σ A) : ∃ (ι : Type) (s : Finset ι) (e : ι → ℂ) (π : ι → CxExt σ A → A) (π' : ι → CxExt σ A' → A'), x = ∑ i ∈ s, e i • of σ (π i x) ∧ (∀ i y, π' i (map σ f y) = f (π i y)) ∧ ∀ i y a, π' i (y * of σ a) = π' i y * a

theorem **cx_smul** (a : F) (f : Y.Fn) : Y.cx σ (a • f) = σ a • Y.cx σ f

theorem **cx_injective** : Function.Injective (Y.cx σ)

theorem **mem_ptsIn_iff** {y : Y.pts σ} {U : Y.X.Opens} : y ∈ ptsIn Y.str σ U ↔ base y ∈ U

theorem **evStalk_germ** {y : Y.pts σ} {U : Y.X.Opens} (h : base y ∈ U) (a : Γ(Y.X, U)) : evStalk y (Y.X.presheaf.germ U _ h a) = ev Y.str σ U a y

theorem **evStalk_ne_zero_iff** {y : Y.pts σ} (a : Y.X.presheaf.stalk (base y)) : evStalk y a ≠ 0 ↔ IsUnit a

theorem **germIncl_injective** (y : Y.pts σ) : Function.Injective (germIncl y)

theorem **HasVal.unique** (h : HasVal y f c) (h' : HasVal y f c') : c = c'

theorem **HasVal.regC** (h : HasVal y f c) : RegC y f

theorem **RegC.hasVal** (h : RegC y f) : HasVal y f (val y f)

theorem **HasVal.val_eq** (h : HasVal y f c) : val y f = c

theorem **val_of_not_regC** (h : ¬ RegC y f) : val y f = 0

theorem **regC_of_val_ne_zero** (h : val y f ≠ 0) : RegC y f

theorem **hasVal_germIncl** (a) : HasVal y (germIncl y a) (germEv y a)

theorem **hasVal_algebraMap** (c : ℂ) : HasVal y (algebraMap ℂ _ c) c

theorem **hasVal_one** : HasVal y 1 1

theorem **hasVal_zero** : HasVal y 0 0

theorem **HasVal.add** (h : HasVal y f c) (h' : HasVal y f' c') : HasVal y (f + f') (c + c')

theorem **HasVal.mul** (h : HasVal y f c) (h' : HasVal y f' c') : HasVal y (f * f') (c * c')

theorem **HasVal.smul** (a : ℂ) (h : HasVal y f c) : HasVal y (a • f) (a * c)

theorem **HasVal.neg** (h : HasVal y f c) : HasVal y (-f) (-c)

theorem **HasVal.sub** (h : HasVal y f c) (h' : HasVal y f' c') : HasVal y (f - f') (c - c')

theorem **HasVal.pow** (h : HasVal y f c) (n : ℕ) : HasVal y (f ^ n) (c ^ n)

theorem **HasVal.sum** {ι : Type*} {s : Finset ι} {f : ι → Y.FnC σ} {c : ι → ℂ} (h : ∀ i ∈ s, HasVal y (f i) (c i)) : HasVal y (∑ i ∈ s, f i) (∑ i ∈ s, c i)

theorem **HasVal.of_mul** {u : Y.FnC σ} {d : ℂ} (hu : HasVal y u d) (hd : d ≠ 0) (h : HasVal y (u * f) c) : HasVal y f (c / d)

theorem **regC_mul_iff** {u : Y.FnC σ} (hu : val y u ≠ 0) : RegC y (u * f) ↔ RegC y f

theorem **val_mul** (h : RegC y f) (h' : RegC y f') : val y (f * f') = val y f * val y f'

theorem **val_unit_mul** {u : Y.FnC σ} (hu : val y u ≠ 0) (f : Y.FnC σ) : val y (u * f) = val y u * val y f

theorem **hasVal_cx_stalk** (a : Y.X.presheaf.stalk (base y)) : HasVal y (Y.cx σ (algebraMap _ Y.X.functionField a)) (evStalk y a)

theorem **hasVal_cx_fnOf** {U : Y.X.Opens} (h : base y ∈ U) (h' : genericPoint Y.X ∈ U) (a : Γ(Y.X, U)) : HasVal y (Y.cx σ (fnOf U h' a)) (ev Y.str σ U a y)

theorem **val_cx_fnOf** {U : Y.X.Opens} (h : base y ∈ U) (h' : genericPoint Y.X ∈ U) (a : Γ(Y.X, U)) : val y (Y.cx σ (fnOf U h' a)) = ev Y.str σ U a y

theorem **regC_cx_iff** {g : Y.Fn} : RegC y (Y.cx σ g) ↔ RegAt (base y) g

theorem **val_cx_ne_zero_iff** {g : Y.Fn} : val y (Y.cx σ g) ≠ 0 ↔ UnitAt (base y) g

theorem **exists_spread** {x : Y.X} (a : CxExt σ (Y.X.presheaf.stalk x)) : ∃ (U : Y.X.Opens) (A : Y.pts σ → ℂ), x ∈ U ∧ (Y.an σ).HoloOn (ptsIn Y.str σ U) A ∧ ∀ z ∈ ptsIn Y.str σ U, ∃ a', germIncl z a' = CxExt.map σ (Y.stalkIncl x) a ∧ germEv z a' = A z

theorem **RegC.exists_nhds** (h : RegC y f) : ∃ N : Set (Y.pts σ), IsOpen N ∧ y ∈ N ∧ N ⊆ regSet f ∧ (Y.an σ).HoloOn N fun z => val z f

theorem **isOpen_regSet** (f : Y.FnC σ) : IsOpen (regSet f)

theorem **holoOn_val** (f : Y.FnC σ) : (Y.an σ).HoloOn (regSet f) fun z => val z f

theorem **continuousOn_val** (f : Y.FnC σ) : ContinuousOn (fun z => val z f) (regSet f)

theorem **exists_open_regC** (f : Y.FnC σ) : ∃ U : Y.X.Opens, genericPoint Y.X ∈ U ∧ ptsIn Y.str σ U ⊆ regSet f

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.TeichmullerInertiaLabelCondition

theorem **mem_Wild** {σ : G E Ω} : σ ∈ Wild E A ↔ ∀ y : Ω, y ≠ 0 → A.valuation (σ y / y - 1) < 1

theorem **Wild_le_Ine** : Wild E A ≤ Ine (F

theorem **conj_mem_Wild** {φ σ : G E Ω} (hφ : φ ∈ Dec (F

theorem **tameGenAt_iff** {γ : G E Ω} : TameGenAt A γ ↔ γ ∈ Ine (F

theorem **tame_relation** {γ φ : G E Ω} {q : ℕ} (hγ : γ ∈ Ine (F

theorem **mem_tameSpace** (a : WDRep h V) {v : V} : v ∈ a.tameSpace ↔ ∀ σ : Weil h, (σ : G E Ω) ∈ Wild E A → a.ρ σ v = v

theorem **ρ_mem_tameSpace** (a : WDRep h V) (σ : Weil h) : ∀ v ∈ a.tameSpace, a.ρ σ v ∈ a.tameSpace

theorem **N_mem_tameSpace** (a : WDRep h V) : ∀ v ∈ a.tameSpace, a.N v ∈ a.tameSpace

@[simp] theorem **coe_tamePart_ρ** (a : WDRep h V) (σ : Weil h) (v : a.tameSpace) : (a.tamePart.ρ σ v : V) = a.ρ σ v

@[simp] theorem **coe_tamePart_N** (a : WDRep h V) (v : a.tameSpace) : (a.tamePart.N v : V) = a.N v

theorem **tamePart_ρ_of_wild** (a : WDRep h V) {σ : Weil h} (hσ : (σ : G E Ω) ∈ Wild E A) : a.tamePart.ρ σ = 1

theorem **tame_tamePart** (a : WDRep h V) : a.tamePart.Tame

theorem **tame_iff** (a : WDRep h V) : a.Tame ↔ a.tameSpace = ⊤

@[simp] theorem **coe_tameEquiv** {a : WDRep h V} {b : WDRep h V'} (e : V ≃ₗ[ℂ] V') (he) (v : a.tameSpace) : (tameEquiv (a

theorem **Iso.tamePart** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) : a.tamePart.Iso b.tamePart

theorem **Tame.iso_tamePart** {a : WDRep h V} (ha : a.Tame) : a.tamePart.Iso a

theorem **tamePart_sum** (a : WDRep h V) (b : WDRep h V') : (a.sum b).tamePart.Iso (a.tamePart.sum b.tamePart)

theorem **charpoly_tamePart_congr** {a : WDRep h V} {b : WDRep h V'} (e : V ≃ₗ[ℂ] V') (he : ∀ σ : Weil h, (σ : G E Ω) ∈ Ine (F

theorem **Tame.conj_eq_pow** {a : WDRep h V} (ha : a.Tame) {γ φ : Weil h} (hγ : (γ : G E Ω) ∈ Ine (F

theorem **Tame.exists_pow_eq_one** {a : WDRep h V} (ha : a.Tame) {γ : Weil h} (hγ : (γ : G E Ω) ∈ Ine (F

theorem **Tame.roots_map_pow** {a : WDRep h V} (ha : a.Tame) {γ : Weil h} (hγ : (γ : G E Ω) ∈ Ine (F

theorem **IsTeich.injective** {τ : Fbar g →* ℂ} (hτ : IsTeich τ) : Function.Injective τ

theorem **IsTeich.map_zero** {τ : Fbar g →* ℂ} (hτ : IsTeich τ) : τ 0 = 0

theorem **IsTeich.exists_eq** {τ : Fbar g →* ℂ} (hτ : IsTeich τ) {n : ℕ} (hn : 0 < n) (hc : n.Coprime g) {z : ℂ} (hz : z ^ n = 1) : ∃ x, τ x = z

theorem **eq_of_rootsBar_eq** {f f' : (ZMod g)[X]} (hf : f.Monic) (hf' : f'.Monic) (e : rootsBar f = rootsBar f') : f = f'

theorem **IsTeich.existsUnique_monic** {τ : Fbar g →* ℂ} (hτ : IsTeich τ) {m : Multiset ℂ} (h1 : ∀ z ∈ m, ∃ x, τ x = z) (h2 : m.map (· ^ g) = m) : ∃! f : (ZMod g)[X], f.Monic ∧ (rootsBar f).map τ = m

@[simp] theorem **lpTame_mk** (a : WDRep h V) : lpTame (mk a) = mk a.tamePart

theorem **lpDim_lpTame_le** (x : WD h) : lpDim (lpTame x) ≤ lpDim x

theorem **tame_iff_lpDim** (x : WD h) : tame x ↔ lpDim (lpTame x) = lpDim x

@[simp] theorem **tame_mk** (a : WDRep h V) : tame (mk a) ↔ a.Tame

theorem **tame_iff_inePol** (x : WD h) : tame x ↔ ∀ σ ∈ Wild E A, inePol x σ = (X - 1) ^ lpDim x

theorem **tame_of_unr** {x : WD h} (hx : unr x) : tame x

theorem **tameEig_mk** (a : WDRep h V) {γ : G E Ω} (hγ : γ ∈ Weil h) : tameEig (mk a) γ = (a.tamePart.ρ ⟨γ, hγ⟩).charpoly.roots

theorem **card_tameEig** (x : WD h) {γ : G E Ω} (hγ : γ ∈ Weil h) : Multiset.card (tameEig x γ) = lpDim (lpTame x)

theorem **tameEig_lpSum** (x y : WD h) {γ : G E Ω} (hγ : γ ∈ Weil h) : tameEig (lpSum x y) γ = tameEig x γ + tameEig y γ

theorem **exists_pow_tameEig** (x : WD h) {γ : G E Ω} (hγ : γ ∈ Ine (F

theorem **tameEig_map_pow** (x : WD h) {γ : G E Ω} (hγ : γ ∈ Ine (F

theorem **TeichmullerInertiaLabelCondition.of_tameGenAt** {τ : Fbar g →* ℂ} {γ : G E Ω} (hw : normV w = g) (hγ : TameGenAt A γ) (hτ : ∀ x y, x ≠ 0 → y ≠ 0 → τ x = τ y → x = y) : TeichmullerInertiaLabelCondition A w τ γ

theorem **existsUnique_lab** (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : ∃! f : (ZMod g)[X], f.Monic ∧ (rootsBar f).map τ = tameEig x γ

theorem **lab_of_not** (H : ¬ TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : lab τ γ x = 1

theorem **lab_monic** (τ : Fbar g →* ℂ) (γ : G E Ω) (x : WD h) : (lab τ γ x).Monic

theorem **map_rootsBar_lab** (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : (rootsBar (lab τ γ x)).map τ = tameEig x γ

theorem **lab_eq_iff** (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) {f : (ZMod g)[X]} : lab τ γ x = f ↔ f.Monic ∧ (rootsBar f).map τ = tameEig x γ

theorem **lab_congr** {x y : WD h} (e : γ ∈ Ine (F

theorem **inePol_lpTame** (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : inePol (lpTame x) γ = ((rootsBar (lab τ γ x)).map fun y => X - C (τ y)).prod

theorem **natDegree_lab** (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : (lab τ γ x).natDegree = lpDim (lpTame x)

theorem **natDegree_lab_le** (τ : Fbar g →* ℂ) (γ : G E Ω) (x : WD h) : (lab τ γ x).natDegree ≤ lpDim x

theorem **natDegree_lab_eq_iff** (H : TeichmullerInertiaLabelCondition A w τ γ) (x : WD h) : (lab τ γ x).natDegree = lpDim x ↔ tame x

theorem **tame_of_natDegree_lab** {x : WD h} (hx : (lab τ γ x).natDegree = lpDim x) : tame x

theorem **inePol_wild_of_natDegree_lab** {x : WD h} (hx : (lab τ γ x).natDegree = lpDim x) {σ : G E Ω} (hσ : ∀ y : Ω, y ≠ 0 → A.valuation (σ y / y - 1) < 1) : inePol x σ = (X - 1) ^ lpDim x

theorem **lab_coeff_zero_ne_zero** (τ : Fbar g →* ℂ) (γ : G E Ω) (x : WD h) : (lab τ γ x).coeff 0 ≠ 0

theorem **inePol_eq_of_natDegree_lab** (τ : Fbar g →* ℂ) (γ : G E Ω) (x : WD h) (hx : (lab τ γ x).natDegree = lpDim x) : inePol x γ = ((rootsBar (lab τ γ x)).map fun y => X − C (τ y)).prod

theorem **lab_of_lpDim_eq_zero** (τ : Fbar g →* ℂ) (γ : G E Ω) {x : WD h} (hx : lpDim x = 0) : lab τ γ x = 1

@[simp] theorem **lab_lpZero** (τ : Fbar g →* ℂ) (γ : G E Ω) : lab τ γ (lpZero h) = 1

theorem **lab_lpSum** (τ : Fbar g →* ℂ) (γ : G E Ω) (x y : WD h) : lab τ γ (lpSum x y) = lab τ γ x * lab τ γ y

theorem **lab_mk_congr** {a : WDRep h V} {b : WDRep h V'} (e : V ≃ₗ[ℂ] V') (he : ∀ σ : Weil h, (σ : G E Ω) ∈ Ine (F

theorem **lab_lpTwist** (τ : Fbar g →* ℂ) (γ : G E Ω) (x : WD h) {χ : Weil h →* ℂˣ} (hχ : IsSmooth χ) (h1 : ∀ σ : Ine (F

theorem **lab_lpTw** (τ : Fbar g →* ℂ) (γ : G E Ω) (x : WD h) (s : ℝ) : lab τ γ (lpTw x s) = lab τ γ x

theorem **lab_eq_iff_of_tame** (H : TeichmullerInertiaLabelCondition A w τ γ) {x : WD h} (hx : tame x) {f : (ZMod g)[X]} : lab τ γ x = f ↔ f.Monic ∧ (rootsBar f).map τ = (inePol x γ).roots

theorem **rootsBar_X_sub_C_pow** (c : ZMod g) (n : ℕ) : rootsBar ((X − C c) ^ n) = n • {algebraMap (ZMod g) (Fbar g) c}

theorem **lab_mk_of_trivial** (H : TeichmullerInertiaLabelCondition A w τ γ) (a : WDRep h V) (ha : ∀ σ : Ine (F

theorem **lab_lpSp** (H : TeichmullerInertiaLabelCondition A w τ γ) (n : ℕ) : lab τ γ (lpSp h n) = (X − 1) ^ n

theorem **lab_lpOfChar_eq_iff** (H : TeichmullerInertiaLabelCondition A w τ γ) {χ : Weil h →* ℂˣ} (hχ : IsSmooth χ) (ht : ∀ σ : Weil h, (σ : G E Ω) ∈ Wild E A → χ σ = 1) (c : ZMod g) : lab τ γ (lpOfChar h χ hχ) = X − C c ↔ τ (algebraMap (ZMod g) (Fbar g) c) = χ ⟨γ, Ine_le_Weil h H.2.1⟩

theorem **lab_lpOfChar_of_wild** (τ : Fbar g →* ℂ) (γ : G E Ω) {χ : Weil h →* ℂˣ} (hχ : IsSmooth χ) {σ : Weil h} (hσ : (σ : G E Ω) ∈ Wild E A) (h1 : χ σ ≠ 1) : lab τ γ (lpOfChar h χ hχ) = 1

theorem **IsSmooth.eq_one_of_tameGenAt** {h : IsPlaceOver A w} {M : Type*} [Monoid M] {f : Weil h →* M} (hf : IsSmooth f) {γ : G E Ω} (hγ : TameGenAt A γ) (h1 : f (ineIncl h ⟨γ, hγ.1⟩) = 1) (h2 : ∀ σ : Weil h, (σ : G E Ω) ∈ Wild E A → f σ = 1) (σ : Ine (F

theorem **exists_tameInertiaSetting** (h : IsPlaceOver A w) (hw : normV w = g) : ∃ (τ : Fbar g →* ℂ) (γ : G E Ω), TameGenAt A γ ∧ TeichmullerInertiaLabelCondition A w τ γ

theorem **exists_normV_eq** (g : ℕ) (hg : g.Prime) : ∃ v : HeightOneSpectrum (𝒪 ℚ), normV v = g

theorem **sqrtChar_eq_one_iff** (h : IsPlaceOver A w) {x : Ω} {a : E} (hx : x ^ 2 = algebraMap E Ω a) (σ : Weil h) : sqrtChar h hx σ = 1 ↔ (σ : G E Ω) x = x

theorem **isSmooth_sqrtChar** (h : IsPlaceOver A w) {x : Ω} {a : E} (hx : x ^ 2 = algebraMap E Ω a) : IsSmooth (sqrtChar h hx)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.TendstoCutWt

theorem **isCutoff** : GlobalTheory.IndefiniteRowSideFullHecke.HermitianInterpolation.IsCutoff cutoff

theorem **tendsto_cutWt** {κ r : ℝ} (hκ : 0 < κ) (hr : 0 < r) : Tendsto (fun R => cutWt κ R r) atTop (𝒩 1)

theorem **cutWt_zero** {R r : ℝ} (hR : 2 ≤ R) (hr : 0 < r) (h : r ≤ R ^ 2) : cutWt 0 R r = 1 / 2

theorem **cutWt_mem_Ioo** (κ : ℝ) {R r : ℝ} (hR : 2 ≤ R) (hr : 0 < r) (h : r ≤ R ^ 2) : cutWt κ R r ∈ Set.Ioo (0 : ℝ) 1

theorem **grSq_dual_rev** {m : ℕ} {M : Matrix (Fin m) (Fin m) ℂ} (h : M.PosDef) (i : Fin m) : grSq ((dualGram M).submatrix Fin.rev Fin.rev) i = (grSq M i.rev)⁻¹

theorem **three_le_kap** (h : ∃ C, x.IsMuExp C) : 3 ≤ x.kap

theorem **tendsto_wt** (h : ∃ C, x.IsMuExp C) (hy : ¬ LiesIn y (genBd s.T.Z)) : Tendsto (fun R => x.wt σ R y) atTop (𝒩 1)

theorem **wt_conj** (c : CxPts (genStr s.T.Z) σ' → CxPts (genStr s.T.Z) σ) (hc : IsConjMap c) (R : ℝ) (z : CxPts (genStr s.T.Z) σ') : x.wt σ' R z = x.wt σ R (c z)

theorem **gram_some** (R : ℝ) (g : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)) (t : D.what₂.Tau) : (x.eigMetAt σ (some R)).gram y g (DV.tauEmb D t) = mix (x.wt σ R y) (x.gram y g _) ((x.hodge σ).gram y g _)

theorem **comparison** {C : ℝ} (h : x.IsMuExp C) (hy : ¬ LiesIn y (genBd s.T.Z)) (R : ℝ) (g : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)) (t : D.what₂.Tau) (v : Fin (S.Dw.mult (DV.tauEmb D t)) → ℂ) : x.r σ y ^ (−C) * (star v ⬝ᵥ x.gram y g _ *ᵥ v).re ≤ (star v ⬝ᵥ (x.eigMetAt σ (some R)).gram y g _ *ᵥ v).re ∧ (star v ⬝ᵥ (x.eigMetAt σ (some R)).gram y g _ *ᵥ v).re ≤ x.r σ y ^ C * (star v ⬝ᵥ x.gram y g _ *ᵥ v).re

theorem **gram_some_ne** {R : ℝ} (hR : 2 ≤ R) (hy : ¬ LiesIn y (genBd s.T.Z)) (hr : x.r σ y ≤ R ^ 2) (g : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)) (t : D.what₂.Tau) (h : x.gram y g (DV.tauEmb D t) ≠ (x.hodge σ).gram y g _) : (x.eigMetAt σ (some R)).gram y g (DV.tauEmb D t) ≠ x.gram y g _ ∧ (x.eigMetAt σ (some R)).gram y g (DV.tauEmb D t) ≠ (x.hodge σ).gram y g _

theorem **hodge_indep** (x' : ToroidalCompactificationData s) (hy : ¬ LiesIn y (genBd s.T.Z)) (g : EigFrame s.T.Z σ (y.1 ≫ genι s.T.Z)) (τ : D.host.E →+* ℂ) : (x.hodge σ).gram y g τ = (x'.hodge σ).gram y g τ

theorem **pn_lieFac** (Φ : s.Flag) (k : s.Comp) {σ : D.what₂.LL →+* ℂ} {p : Pts (Φ.C k).Y σ} {g : (Φ.C k).FrameAt p} (hg : IsHodgeFrame s.U _ g) (a : (D.host.E →+* ℂ) × ℕ → ℕ) {f : (Φ.C k).Y.Fn} (h : IsCoord (Φ.C k) p.1 g (fun z => −(a (z.1, z.2) : ℤ)) f 1) : (cxSt D.what₂.LL).pn _ σ _ (metsOf x Φ k none σ fun z => −(a (z.1, z.2) : ℤ)) ((cxSt D.what₂.LL).cx _ σ f) p = Formalism.L2.PjFree.lieFac S.Dw (xiR D) a ^ 2

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ThetaChr

theorem **IsBuiltE.isBuilt** {𝒮 : Slots} (K X f T c) (h : IsBuiltE 𝒮 K X f T c) : IsBuilt 𝒮 K X f T c

theorem **IsGoodE.isGood** {𝒮 : Slots} (K X f T c) (h : IsGoodE 𝒮 K X f T c) : IsGood 𝒮 K X f T c

theorem **IsGoodE.isBuiltE** {𝒮 : Slots} (K X f T c) (h : IsGoodE 𝒮 K X f T c) : IsBuiltE 𝒮 K X f T c

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ThetaCubIsPullCubical

theorem **mem_of_comm** {f : Z ⟶ W'} {g : W' ⟶ X} {f' : Z ⟶ W} {g' : W ⟶ X} (e : f ≫ g = f' ≫ g') (z : Z) {U : X.Opens} (h : g (f z) ∈ U) : g' (f' z) ∈ U

theorem **gw_combIso** (hu : ∀ i, u' i ≫ F = Φ ≫ u i) (δ : J → ℤ) (w : LB.Iso L' (L.pull F)) (f : Z ⟶ W') (z : Z) (j : J → L.ι) (k : J → L'.ι) (hj : Φ (f z) ∈ LB.wchart L u j) (hk : f z ∈ LB.wchart L' u' k) : gw (L

theorem **pGen_combIso** (hu : ∀ i, u' i ≫ F = Φ ≫ u i) (δ : J → ℤ) {w : LB.Iso L' (L.pull F)} (hw : PGen w) : PGen (L

theorem **gm_pullAlong** (L : LB X) (F : X' ⟶ X) (h : Y' ⟶ Y) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (ψ : K → (Y' ⟶ X')) (hψ : ∀ k, ψ k ≫ F = h ≫ φ k) (t : LB.Gen (LB.LinComb L φ ε)) (r : Z ⟶ Y') (z : Z) (jj : K → PUnit → L.ι) (hz : r z ∈ LB.wchart (L.pull F) ψ jj) : gm r z

(LB.wchart (L.pull F) ψ jj) ((GeneratorPullback.pullAlong L F h φ ε ψ hψ t).s jj) = gm (r ≫ h) z (LB.wchart L φ fun k => jj k ⟨⟩) (t.s fun k => jj k ⟨⟩)

theorem **pMaps_combIso** (hu : ∀ i, u' i ≫ F = Φ ≫ u i) (δ : J → ℤ) {w : LB.Iso L' (L.pull F)} {ε : K → ℤ} {ψ' : K → (Y₀' ⟶ X')} {ψ : K → (Y₀ ⟶ X)} {a₀ : LB.Gen (LB.LinComb L' ψ' ε)} {t₀ : LB.Gen (LB.LinComb L ψ ε)} {h₀ : Y₀' ⟶ Y₀} (H : PMaps w ψ' ψ ε a₀ t₀ h₀) {φ' : K → (Y' ⟶ W')} {φ : K → (Y ⟶ W)} {h : Y' ⟶ Y} (hφ : ∀ k, φ' k ≫ Φ = h ≫ φ k) {ρ' : J → (Y' ⟶ Y₀')} {ρ : J → (Y ⟶ Y₀)} (hρ' : ∀ i k, φ' k ≫ u' i = ρ' i ≫ ψ' k) (hρ : ∀ i k, φ k ≫ u i = ρ i ≫ ψ k) (hh : ∀ i, h ≫ ρ i = ρ' i ≫ h₀) : PMaps (L

theorem **isPullCubical_comb** (m : CubeMap G' G) (n : CubeMap A' A) (he : n.e = m.e) (u : J → (A ⟶ G)) [∀ i, IsMonHom (u i)] (u' : J → (A' ⟶ G')) [∀ i, IsMonHom (u' i)] (δ : J → ℤ) (hu : ∀ i, (u' i).left ≫ m.F = n.F ≫ (u i).left) {L : RigLB G} {L' : RigLB G'} {c : CubicalStr G L} {c' : CubicalStr G' L'} (H : m.IsPullCubical c' c) : n.IsPullCubical (combCub u' δ c') (combCub u δ c)

theorem **isMonHom_psum** [GrpObj T] {n : ℕ} (x : Fin n → (T ⟶ G)) (hx : ∀ i, IsMonHom (x i)) (I : Finset (Fin n)) : IsMonHom (psum G x I)

theorem **isLinearHom_crd** (i : Fin P.n) : IsLinearHom b a (crd t P i)

@[reassoc] theorem **tmul_crd** (m : M) (i : Fin P.n) : t.tmul m ≫ crd t P i = a.act (P.incl m i)

theorem **isLB_biLB** {L : LB G.left} (hL : L.IsLB) (x z : T ⟶ A) : (biLB t P h L x z).IsLB

theorem **cls_biLB** {L : LB G.left} (hL : L.IsLB) (x z : T ⟶ A) (hb : (biLB t P h L x z).IsLB) : cls _ hb = bi t P h (cls L hL) x z

theorem **pull_bi** (c : Pic G.left) (g : T' ⟶ T) (x z : T ⟶ A) : pull g.left (bi t P h c x z) = bi t P h c (g ≫ x) (g ≫ z)

theorem **bi_one_left** (c : Pic G.left) (z : T ⟶ A) : bi t P h c 1 z = 0

theorem **lam_act** [IsCommMonObj G] (x y : T ⟶ G) (c : O) : lam (cls L hL) (x ≫ a.act c) y = lam (cls L hL) x (y ≫ a.act (star c))

theorem **bi_mul_left** (x y z : T ⟶ A) : bi t P h (cls L hL) (x ∗ y) z = bi t P h (cls L hL) x z + bi t P h (cls L hL) y z

theorem **bi_mul_right** (x y z : T ⟶ A) : bi t P h (cls L hL) x (y ∗ z) = bi t P h (cls L hL) x y + bi t P h (cls L hL) x z

theorem **bi_comm** (x z : T ⟶ A) : bi t P h (cls L hL) x z = bi t P h (cls L hL) z x

theorem **bi_act** (x z : T ⟶ A) (c : O) : bi t P h (cls L hL) (x ≫ b.act c) z = bi t P h (cls L hL) x (z ≫ b.act (star c))

theorem **bi_tmul** (x z : T ⟶ G) (m n : M) : bi t P h (cls L hL) (x ≫ t.tmul m) (z ≫ t.tmul n) = lam (cls L hL) (x ≫ a.act (h m n)) z

theorem **lam_bi** {w : O} (hw : w + star w = 1) (x z : T ⟶ A) : lam (bi t P h (cls L hL) (𝟙 A) (b.act w)) x z = bi t P h (cls L hL) x z

theorem **biadd_ext** {V : Type 1} [AddCommGroup V] (β κ : (T ⟶ A) → (T ⟶ A) → V) (β₁ : ∀ x y z, β (x ∗ y) z = β x z + β y z) (β₂ : ∀ x y z, β x (y ∗ z) = β x y + β x z) (κ₁ : ∀ x y z, κ (x ∗ y) z = κ x z + κ y z) (κ₂ : ∀ x y z, κ x (y ∗ z) = κ x y + κ x z) (e : ∀ m n (x z : T ⟶ G), β (x ≫ t.tmul m) (z ≫ t.tmul n) = κ (x ≫ t.tmul m) (z ≫ t.tmul n)) (x z : T ⟶ A) : β x z = κ x z

theorem **bi_mapLeft** {A' : Over S} [GrpObj A'] [IsCommMonObj A'] {b' : OAct O A'} {M' : Type} [AddCommGroup M'] [Module O M'] (t' : IsTensor A' b' G a M') (P' : Summand O M') (h' : HermForm O M') (f : M' →ₗ[O] M) (hf : h'.IsIsometry h f) (x z : T ⟶ A') : bi t' P' h' (cls L hL) x z = bi t P h (cls L hL) (x ≫ t'.mapLeft t f) (z ≫ t'.mapLeft t f)

theorem **bi_eq_sum** {I : Type} [Fintype I] (l : I → (M →ₗ[O] O)) (hl : ∀ m n, h m n = ∑ i, l i m ∗ star (l i n)) (x z : T ⟶ A) : bi t P h (cls L hL) x z = ∑ i, lam (cls L hL) (x ≫ t.mapLeft (IsTensor.self a) (l i)) (z ≫ t.mapLeft (IsTensor.self a) (l i))

theorem **biRig_isSymmetric** {L : RigLB G} (c : CubicalStr G L) (hs : L.IsSymmetric) : (biRig t P h x z L).IsSymmetric

theorem **theta_isSymmetric** (w : O) {L : RigLB G} (c : CubicalStr G L) (hs : L.IsSymmetric) : (theta t P h w L).IsSymmetric

theorem **cls_poi** (L : RigLB G) : cls _ (poi t P h L).isLB = bi t P h (cls L.L L.isLB) (fst A A) (snd A A)

theorem **cls_theta** (w : O) (L : RigLB G) : cls _ (theta t P h w L).isLB = bi t P h (cls L.L L.isLB) (𝟙 A) (b.act w)

theorem **biLB_iso_pull** (L : RigLB G) (x z : T ⟶ A) : LBIso (biLB t P h L.L x z) ((poi t P h L).L.pull (lift x z).left)

theorem **theta_iso_pull** (w : O) (L : RigLB G) : LBIso (theta t P h w L).L ((poi t P h L).L.pull (lift (𝟙 A) (b.act w)).left)

theorem **lamPt_mul** (z z' : T ⟶ A) : lamPt t P h (cls L.L L.isLB) (z ∗ z') = lamPt t P h (cls L.L L.isLB) z + lamPt t P h (cls L.L L.isLB) z'

theorem **lam_theta** (x z : T ⟶ A) : lam (cls _ (theta t P h w L).isLB) x z = bi t P h (cls L.L L.isLB) x z

theorem **cube_theta_iso** (x z : T ⟶ A) : LBIso (cubeLB A ![x, z] (theta t P h w L).L) (biLB t P h L.L x z)

theorem **lam_theta_iso** : LBIso (abelian₂ A (theta t P h w L).L) (poi t P h L).L

theorem **theta_isAdj** : IsAdj b (theta t P h w L).L

theorem **theta_prim** : LBIso (theta t P h w L).L (cubeLB A ![𝟙 A, b.act w] (theta t P h w L).L)

theorem **theta_tmul** (m n : M) : LBIso (cubeLB A ![fst G G ≫ t.tmul m, snd G G ≫ t.tmul n] (theta t P h w L).L) (cubeLB G ![fst G G ≫ a.act (h m n), snd G G] L.L)

theorem **theta_kTriv** [Module.Finite O M] [Module.Projective O M] (hu : h.Unimodular) (hK : Out.KTriv G L.L) : Out.KTriv A (theta t P h w L).L

theorem **theta_iso_mapLeft** {A' : Over S} [GrpObj A'] [IsCommMonObj A'] {b' : OAct O A'} {M' : Type} [AddCommGroup M'] [Module O M'] (t' : IsTensor A' b' G a M') (P' : Summand O M') (h' : HermForm O M') (f : M' →ₗ[O] M) (hf : h'.IsIsometry h f) : LBIso (theta t' P' h' w L).L ((theta t P h w L).L.pull (t'.mapLeft t f).left)

theorem **theta_iso_box** {k : ℕ} (l : Fin k → (M →ₗ[O] O)) (hl : ∀ m n, h m n = ∑ i, l i m ∗ star (l i n)) (hprim : LBIso L.L (cubeLB G ![𝟙 G, a.act w] L.L)) : LBIso (theta t P h w L).L (Out.boxLB L.L fun i => t.mapLeft (IsTensor.self a) (l i))

theorem **mapLeft_self_comm** (l : M →ₗ[O] O) : (t₁.mapLeft (IsTensor.self a₁) l).left ≫ φ = ψ ≫ (t.mapLeft (IsTensor.self a) l).left

theorem **biMap_comm** {T₁ : Over S'} (x₁ z₁ : T₁ ⟶ A₁) (x z : T ⟶ A) (U : Over.mk (T₁.hom ≫ f) ⟶ T) (hx : x₁.left ≫ ψ = U.left ≫ x.left) (hz : z₁.left ≫ ψ = U.left ≫ z.left) (q : BIx P.n) : (biMap t₁ P h x₁ z₁ q).left ≫ φ = U.left ≫ (biMap t P h x z q).left

theorem **thetaCub_isPullCubical** (w : O) (m : CubeMap G₁ G) (hF : m.F = φ) (he : m.e = f) {L : RigLB G} {L₁ : RigLB G₁} {c : CubicalStr G L} {c₁ : CubicalStr G₁ L₁} (H : m.IsPullCubical c₁ c) : (cubeMapOf hψ.1).IsPullCubical (thetaCub t₁ P h w c₁) (thetaCub t P h w c)

theorem **isSemiAbelian_tensor** {O : Type} [CommRing O] {M : Type} [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] {a : OAct O G} {b : OAct O A} (t : IsTensor A b G a M) (hG : IsSemiAbelian G) : IsSemiAbelian A

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ThetaSqMulSqrtEtaSq

theorem **Frame.lies** {z : NgC g} {x : KeyIn ℂ} (h : z.Frame x) : z.Lies x

theorem **exists_frame** (hg : 1 ≤ g) (z : NgC g) : ∃ x : KeyIn ℂ, z.Frame x

theorem **exists_of_keyIn** (x : KeyIn ℂ) : ∃ z : NgC x.P.g, z.Lies x

theorem **det_Y_pos** (z : NgC g) : 0 < z.Y.det

theorem **thetaSq_eq** (z : NgC g) : z.thetaSq = (Real.sqrt (2 ^ g * z.Y.det))⁻¹

theorem **lieGram_eq** (z : NgC g) : z.lieGram = (((4 * Real.pi)⁻¹ • (z.Y)⁻¹ : Matrix (Fin g) (Fin g) ℝ)).map fun x : ℝ => (x : ℂ)

theorem **etaSq_eq** (z : NgC g) : z.etaSq = (4 * Real.pi) ^ g * z.Y.det

theorem **thetaSq_pos** (z : NgC g) : 0 < z.thetaSq

theorem **etaSq_pos** (z : NgC g) : 0 < z.etaSq

theorem **thetaSq_mul_sqrt_etaSq** (z : NgC g) : z.thetaSq * Real.sqrt z.etaSq = Real.sqrt ((2 * Real.pi) ^ g)

theorem **famUnit** (h : IsKF g c κ) : FamUnit κ

theorem **valUnit** (h : IsKF g c κ) : ValUnit κ

theorem **famHomog** (h : IsKF g c κ) (S : Type) [CommRing S] [IsDomain S] [CharZero S] (P : Out.PolAV (Sp S)) (hg rig hr th hth v hv th' hth' v' hv') (a : S) (M : Matrix (Fin P.g) (Fin P.g) S) (u u' : S) (hP : P.g = g) (ha : ∀ i, LB.Sec.f th' i = constFn P.A.hom _ a * LB.Sec.f th i) (hM : ∀ i, IsLieComb (M i) v (v' i)) (hu : IsValue κ ⟨P, hg, rig, hr, th, hth, v, hv⟩ u) (hu' : IsValue κ ⟨P, hg, rig, hr, th', hth', v', hv'⟩ u') : a ^ (2 * c) * u' = M.det ^ c * u

theorem **value_normalAt** (h : IsKF g c κ) (x : KeyIn ℂ) (hx : x.P.g = g) {τ M : Matrix (Fin x.P.g) (Fin x.P.g) ℂ} {a b : Fin x.P.g → ℂ} {c₀ : ℂ} (hc : c₀ ≠ 0) (hM : M.det ≠ 0) (hN : NormalAt x τ M a b c₀) (t : ℂ) (ht : ∀ (y : KeyIn ℂ) (e : y.P.g = x.P.g), NormalAt y (τ.submatrix (Fin.cast e) (Fin.cast e)) 1 (fun i => a (i.cast e)) (fun i => b (i.cast e)) 1 → (κ ℂ y : ℂ) = t) : c₀ ^ (2 * c) * (κ ℂ x : ℂ) = M.det ^ c * t

theorem **ext** (h : IsKF g c κ) (h' : IsKF g c κ') (f : NgC g → ℂˣ) (hf : ∀ (z : NgC g) x, z.Frame x → κ ℂ x = f z) (hf' : ∀ (z : NgC g) x, z.Frame x → κ' ℂ x = f z) : κ = κ'

theorem **not_one** (hg : 1 ≤ g) (hc : 1 ≤ c) : ¬ IsKF g c (fun _ _ _ _ => 1)

theorem **isKF_cutRule** {g c : ℕ} (s : ℤˣ) {κ : Rule} (hH : ∀ (K : Type) [Field K] [CharZero K] (P : Out.PolAV (Sp K)) (hg rig hr th hth v hv th' hth' v' hv') (a : K) (M : Matrix (Fin P.g) (Fin P.g) K), P.g = g → (∀ i, LB.Sec.f th' i = constFn P.A.hom _ a * LB.Sec.f th i) → (∀ i, IsLieComb (M i) v (v' i)) → a ^ (2 * c) * (κ K ⟨P, hg, rig, hr, th', hth', v', hv'⟩ : K) = M.det ^ c * κ K ⟨P, hg, rig, hr, th, hth, v, hv⟩) (hF : Funct κ) (hU : UnitSp κ) : IsKF g c (cutRule g s κ) where

theorem **isKF** : IsKF g c kf.rule

theorem **rule_frame** {z : NgC g} {x : KeyIn ℂ} (h : z.Frame x) : kf.rule ℂ x = (kf.val z)⁻¹

theorem **rule_unique** {κ : Rule} (h : IsKF g c κ) (hv : ∀ (z : NgC g) (x : KeyIn ℂ), z.Frame x → κ ℂ x = (kf.val z)⁻¹) : κ = kf.rule

@[**ext**] theorem ext {kf kf' : KF g c} (h : kf.val = kf'.val) : kf = kf'

theorem **rule_injective** (hg : 1 ≤ g) : Function.Injective (rule : KF g c → Rule)

theorem **coef_frame** {z : NgC g} {x : KeyIn ℂ} (h : z.Frame x) : kf.coef ℂ x = kf.val z

theorem **coef_homog** (K : Type) [Field K] [CharZero K] (P : Out.PolAV (Sp K)) (hg rig hr th hth v hv th' hth' v' hv') (a : K) (M : Matrix (Fin P.g) (Fin P.g) K) (hP : P.g = g) (ha : ∀ i, LB.Sec.f th' i = constFn P.A.hom _ a * LB.Sec.f th i) (hM : ∀ i, IsLieComb (M i) v (v' i)) : M.det ^ c * (kf.coef K ⟨P, hg, rig, hr, th', hth', v', hv'⟩ : K) = a ^ (2 * c) * kf.coef K ⟨P, hg, rig, hr, th, hth, v, hv⟩

theorem **coef_funct** {K K' : Type} [Field K] [CharZero K] [Field K'] [CharZero K'] (f : K →+* K') {x : KeyIn K} {x' : KeyIn K'} (h : KeyPB x' x f) : (kf.coef K' x' : K') = f (kf.coef K x)

theorem **exists_unit_coef** (S : Type) [CommRing S] [IsDomain S] [IsNoetherianRing S] [IsIntegrallyClosed S] [CharZero S] (y : KeyIn S) : ∃ u : Sˣ, IsValue kf.coef y (u : S)

theorem **valUnit_coef** : ValUnit kf.coef

theorem **norm_pos** (z : NgC g) : 0 < kf.norm z

theorem **norm_eq_div** (z : NgC g) : kf.norm z = ‖(kf.val z : ℂ)‖ / Real.sqrt ((2 * Real.pi) ^ g) ^ c

theorem **norm_eq** (z : NgC g) : kf.norm z = ‖(kf.val z : ℂ)‖ * Real.exp (c * g * (-(1 / 2) * Real.log (2 * Real.pi)))

theorem **norm_eq_of_isArchNorm** {z : NgC g} {x : KeyIn ℂ} (hz : z.Lies x) {nth nη : ℝ} (hn : IsArchNorm x nth nη) : kf.norm z = ‖(kf.coef ℂ x : ℂ)‖ / (nth ^ c * Real.sqrt nη ^ c)

theorem **sign_neg_one_ne** : kf.sign (-1) ≠ kf

theorem **exists_rule_eq** (hg : 1 ≤ g) (H : ∀ (z : NgC g) (x x' : KeyIn ℂ), z.Frame x → z.Frame x' → KeyPB x' x (RingHom.id ℂ)) {κ : Rule} (h : IsKF g c κ) : ∃ kf : KF g c, kf.rule = κ

theorem **ngTrue_NgC** (g : ℕ) : ngTrue.NgC g = NgC g

theorem **ngTrue_KF** (g c : ℕ) : ngTrue.KF g c = KF g c

theorem **ngTrue_kfNorm** {g c : ℕ} (kf : KF g c) (z : NgC g) : ngTrue.kfNorm (g

theorem **exists_kf_four** (g : ℕ) : ∃ kf : KF g 4, ∀ z, kf.val z = 1

theorem **ngTrue_keyFormula** (g : ℕ) : ∃ c : ℕ, 1 ≤ c ∧ 4 ∣ c ∧ ∃ kf : ngTrue.KF g c, ∀ z, ngTrue.kfNorm kf z = Real.exp (c * g * (-(1 / 2) * Real.log (2 * Real.pi)))

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ThirdLevelOfGoodTwists

theorem **kept_refl** {D : D7} (c : NChoice D) : Kept c c

theorem **kept_withLoc** {D : D7} (L : LocAll (lay0 D)) (c : NChoice D) : Kept c (chL L c)

theorem **third_level_of_good_twists** {S : SExt} {Γ : GeoP} (h : Has3 (prmR (NEg hL) S Γ).s K B f T w) : Has3 (prmR (NEb hL) S Γ).s K B f T w

theorem **qOf_off** (v : B.places.Pl) (h : v ∉ P.Wt) : QOf P v = ∅

theorem **qOf_on** (v : B.places.Pl) (h : v ∈ P.Wt) : QOf P v = T.Q v

theorem **qOf_eq** (v : B.places.Pl) : QOf P v = T.Q v

theorem **mem_of_twisted** (x : St P) (v : B.places.Pl) (h : TwistedAt (Dof x) v) : v ∈ P.Wt

theorem **del_mul_ne_one** {p : ℕ} (hp : p.Prime) (hS : p ∈ offSet B.toPre) {δ : HChar B.host.E} (hδ : IsDelta (H

theorem **twisted_iff** (x : St P) (v : B.places.Pl) : TwistedAt (Dof x) v ↔ v ∈ P.Wt

theorem **eta_zero** {x : St P} (l : LA hL x) (j : ℕ) (v : B.places.Pl) (h1 : 1 ≤ j) (h2 : j ≤ B.R.m) : x.o.η j v = 0

theorem **chain** (h0 : StepW Start HasBare) (hA : SA hL) (hM : SMt) (hR : SRat hL) (hC : SCusp hL) (b0 : SettingsFormsExist hL) (b1 : SeriesExist hL) (b2 : SB2 hL) (hc : SC hL) (hF : SFin hL) (S : SExt) (Γ : GeoP) : StepW Start (Has3 (prmR (NEb hL) S Γ).s)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.TieEmbeddingCastLevelTie

theorem **extMap_φ** (X : ExtOf Os Ks O Kf) (eO : O ≃+* O') (eK : Kf ≃+* Kf') (hc : ∀ x, eK (algebraMap O Kf x) = algebraMap O' Kf' (eO x)) (hfrac : IsFractionRing O' Kf') (hcomp : IsAdicComplete (IsLocalRing.maximalIdeal O') O') (x : Os) : (X.map eO eK hc hfrac hcomp).φ x = eO (X.φ x)

theorem **extMap_φK** (X : ExtOf Os Ks O Kf) (eO : O ≃+* O') (eK : Kf ≃+* Kf') (hc : ∀ x, eK (algebraMap O Kf x) = algebraMap O' Kf' (eO x)) (hfrac : IsFractionRing O' Kf') (hcomp : IsAdicComplete (IsLocalRing.maximalIdeal O') O') (y : Ks) : (X.map eO eK hc hfrac hcomp).φK y = eK (X.φK y)

theorem **frobOrder_eq** (D : D7) (p : ℕ) : frobOrder D p = nFr D p

theorem **pack_eq** (D : D7) (u : D.what₂.Word) (Lc : ColLevel D.R D.host.F D.host.E D.sp D.host.g (D.mem u).C (D.mem u).A (D.clsLat u)) (w : D.what₂.places.Pl) (h5 : 5 ≤ D.what₂.places.resChar w) (h2 : 2 ≤ nFr D (D.what₂.places.resChar w)) (he : Even (nFr D (D.what₂.places.resChar w))) (I : LocId Lc (D.what₂.places.resChar w) (nOrb (zH D.R.toFrame (hP D (D.what₂.places.resChar w))))) (WQ (D.what₂.places.resChar w) (nFr D (D.what₂.places.resChar w))) (QQ (D.what₂.places.resChar w) (nFr D (D.what₂.places.resChar w)))) (ψ : Formalism.L2.Herm.awayFrom₂ Lc (D.what₂.places.resChar w) →* (ORec D w (nFr D (D.what₂.places.resChar w)))ˣ) : pack D u Lc w h5 h2 he I ψ = locPackC D u Lc w h5 h2 he I ψ

theorem **packAll_eq** : packAll D Lc I ψ = packC D Lc I ψ

theorem **record_eq** (hd' : ∀ u w h, ((packAll D Lc I ψ u w h).gat (rSc sc D.R.k (D.what₂.places.resChar w)) (epSc D.R sc Ce (D.what₂.places.resChar w))).d = Module.finrank ℂ (D.mem u).A.Mu0) : record D sc Ce Lc I ψ hd' = recC D sc Ce Lc I ψ hd

theorem **recordAllTies_level** (H0 : Formalism.L2.Herm.Herm0 (recC D sc Ce Lc I ψ hd)) (hL : ∀ u, H0.lev0 u = Lc u) (T : ∀ u : D.what₂.Word, Forms0 (recC D sc Ce Lc I ψ hd) u) (B : GL (Fin D.R.k) D.host.E) (hBs : Cmp.StdB (recC D sc Ce Lc I ψ hd) B) (hB) (CH) (hH) (u : D.what₂.Word) : (groupCompatibilityOfRecursion D sc Ce Lc I ψ hd H0 hL T B hBs hB CH hH).herm.lev0 u = Lc u

theorem **recordAllTies_forms** (H0 : Formalism.L2.Herm.Herm0 (recC D sc Ce Lc I ψ hd)) (hL : ∀ u, H0.lev0 u = Lc u) (T : ∀ u : D.what₂.Word, Forms0 (recC D sc Ce Lc I ψ hd) u) (B : GL (Fin D.R.k) D.host.E) (hBs : Cmp.StdB (recC D sc Ce Lc I ψ hd) B) (hB) (CH) (hH) : (groupCompatibilityOfRecursion D sc Ce Lc I ψ hd H0 hL T B hBs hB CH hH).forms = T

theorem **IsConcrete.nonempty_allTies** {D : D7} {Gt : GatTies D} {c : LatCmp D} (h : IsConcrete D Gt c) : Nonempty (GroupCompatibilityAll D Gt c)

theorem **AllTies.standardSub** {D : D7} {Gt : GatTies D} {c : LatCmp D} (G : AllTies D Gt c) : Joints.StdSubAt D

theorem **IsConcrete.ringTie** {D : D7} {Gt : GatTies D} {c : LatCmp D} (h : IsConcrete D Gt c) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) (v : Fin (D.gat u w hg).P.nV) : RingTie D u w hg v

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.TiedColumnData

theorem **GIs2.gIs** {w : ColW B f T} {G : GJStage B w.C} (h : GIs2 K w G) : GIs K w G

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ToSTiesRPin

theorem **SRest.toSTiesR_X** : (S.toSTiesR Rc).X = P.X

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ToZModPowKummerEqIff

@[simp] theorem **rootSeq_zero** (a : Ω) : rootSeq l a 0 = a

theorem **rootSeq_succ_pow** (a : Ω) (n : ℕ) : rootSeq l a (n + 1) ^ l = rootSeq l a n

theorem **rootSeq_pow** (a : Ω) (n : ℕ) : rootSeq l a n ^ l ^ n = a

theorem **exists_unitySeq** : ∃ z : ℕ → Ω, (∀ n, IsPrimitiveRoot (z n) (l ^ n)) ∧ ∀ n, z (n + 1) ^ l = z n

theorem **unitySeq_primitive** (n : ℕ) : IsPrimitiveRoot (unitySeq Ω l n) (l ^ n)

theorem **unitySeq_succ_pow** (n : ℕ) : unitySeq Ω l (n + 1) ^ l = unitySeq Ω l n

theorem **unitySeq_pow_eq_iff** (n a b : ℕ) : unitySeq Ω l n ^ a = unitySeq Ω l n ^ b ↔ (a : ZMod (l ^ n)) = b

theorem **unityPow_eq_pow_iff** {n k : ℕ} {c : ℤ_[l]} : unityPow Ω l n c = unitySeq Ω l n ^ k ↔ PadicInt.toZModPow n c = k

theorem **unityPow_eq_iff** {n : ℕ} {c d : ℤ_[l]} : unityPow Ω l n c = unityPow Ω l n d ↔ PadicInt.toZModPow n c = PadicInt.toZModPow n d

theorem **unityPow_add** (n : ℕ) (c d : ℤ_[l]) : unityPow Ω l n (c + d) = unityPow Ω l n c * unityPow Ω l n d

theorem **unityPow_mul** (n : ℕ) (c d : ℤ_[l]) : unityPow Ω l n (c * d) = unityPow Ω l n d ^ (PadicInt.toZModPow n c).val

theorem **unityPow_pow** (n : ℕ) (c : ℤ_[l]) : unityPow Ω l n c ^ l ^ n = 1

theorem **existsUnique_unityPow** {y : ℕ → Ω} (h0 : ∀ n, y n ^ l ^ n = 1) (hy : ∀ n, y (n + 1) ^ l = y n) : ∃! c : ℤ_[l], ∀ n, y n = unityPow Ω l n c

theorem **rootOf_pow** (a : E) (n : ℕ) : rootOf Ω l a n ^ l ^ n = algebraMap E Ω a

theorem **rootOf_ne_zero** {a : E} (ha : a ≠ 0) (n : ℕ) : rootOf Ω l a n ≠ 0

theorem **existsUnique_kummer** (a : Eˣ) (σ : G E Ω) : ∃! c : ℤ_[l], ∀ n, σ (rootOf Ω l (a : E) n) = zetaPow E Ω l n c * rootOf Ω l (a : E) n

theorem **kummer_spec** (a : Eˣ) (σ : G E Ω) (n : ℕ) : σ (rootOf Ω l (a : E) n) = zetaPow E Ω l n (kummer l a σ) * rootOf Ω l (a : E) n

theorem **apply_zetaPow** (σ : G E Ω) (n : ℕ) (c : ℤ_[l]) : σ (zetaPow E Ω l n c) = zetaPow E Ω l n ((cycZ E Ω l σ : ℤ_[l]ˣ) * c)

theorem **kummer_mul** (a : Eˣ) (σ τ : G E Ω) : kummer l a (σ * τ) = kummer l a σ + (cycZ E Ω l σ : ℤ_[l]ˣ) * kummer l a τ

@[simp] theorem **kummer_one** (a : Eˣ) : kummer l a (1 : G E Ω) = 0

theorem **toZModPow_kummer_eq_iff** (a : Eˣ) (σ τ : G E Ω) (n : ℕ) : PadicInt.toZModPow n (kummer l a σ) = PadicInt.toZModPow n (kummer l a τ) ↔ σ (rootOf Ω l (a : E) n) = τ (rootOf Ω l (a : E) n)

theorem **continuous_kummer** (a : Eˣ) : Continuous (kummer l a : G E Ω → ℤ_[l])

theorem **cycZ_of_mem_Ine** (h : IsPlaceOver A w) (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) {σ : G E Ω} (hσ : σ ∈ Ine (F

theorem **cycZ_weil** {h : IsPlaceOver A w} (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (g : Weil h) : (((cycZ E Ω l g : ℤ_[l]ˣ) : ℤ_[l]) : ℚ_[l]) = ((normV w : ℕ) : ℚ_[l]) ^ deg g

@[simp] theorem **ineIncl_ineConj** {h : IsPlaceOver A w} (g : Weil h) (σ : Ine (F

@[simp] theorem **coe_ineConj** {h : IsPlaceOver A w} (g : Weil h) (σ : Ine (F

theorem **continuous_ineConj** {h : IsPlaceOver A w} (g : Weil h) : Continuous (ineConj g)

@[simp] theorem **toAdd_kummerChar** (h : IsPlaceOver A w) (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (a : Eˣ) (σ : Ine (F

theorem **continuous_kummerChar** (h : IsPlaceOver A w) (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (a : Eˣ) : Continuous (kummerChar h hl a)

theorem **kummerChar_ineConj** {h : IsPlaceOver A w} (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (a : Eˣ) (g : Weil h) (σ : Ine (F

theorem **exists_uniformiser** : ∃ ϖ : 𝒪 E, ϖ ∈ w.asIdeal ∧ ϖ ∉ w.asIdeal ^ 2

theorem **unif_mem** : unif w ∈ w.asIdeal

theorem **unif_notMem** : unif w ∉ w.asIdeal ^ 2

theorem **surjective_of_isUnit** {H : Type*} [Group H] [TopologicalSpace H] [CompactSpace H] {t : H →* Multiplicative ℤ_[l]} (ht : Continuous t) {σ : H} (hσ : IsUnit (t σ).toAdd) : Function.Surjective t

theorem **surjective_kummerChar** (h : IsPlaceOver A w) (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (a : Eˣ) {σ : G E Ω} (hσ : σ ∈ Ine (F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ToroidalBoundaryMetricData

theorem **ev_eq_zero_iff** {C : Closeds Y} {V : Y.affineOpens} {t : Γ(Y, V.1)} (h : (Scheme.IdealSheafData.vanishingIdeal C).ideal V = Ideal.span {t}) {y : CxPts p σ} (hy : y ∈ ptsIn p σ V.1) : ev p σ V.1 t y = 0 ↔ LiesIn y C

theorem **isOpen_not_liesIn** (C : Closeds Y) : IsOpen {y : CxPts p σ | ¬ LiesIn y C}

theorem **pos_iff** (y : CxPts p σ) : 0 < n y ↔ ¬ LiesIn y C

theorem **of_anRead** (R' : AnRead p σ d) : IsDivNorm R' C n

theorem **smoothOn_log** : R.SmoothOn {y | ¬ LiesIn y C} fun y => Real.log (n y ^ 2)

theorem **isDivNorm_of_eqn** [IsAffine Y] (R : AnRead p σ d) {C : Closeds Y} {t : Γ(Y, ⊤)} (h : (Scheme.IdealSheafData.vanishingIdeal C).ideal ⟨⊤, isAffineOpen_top Y⟩ = Ideal.span {t}) : IsDivNorm R C fun y => ‖ev p σ ⊤ t y‖ / √(1 + ‖ev p σ ⊤ t y‖ ^ 2)

theorem **IsKaehlerForm.trans** {R : AnRead p σ d} {a : ChartForm p σ d} (h : IsKaehlerForm R a) {c c' : AlgChart p σ d} {y : CxPts p σ} (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) (ξ : Fin d → ℂ) : a c y ξ = a c' y (c.dtrans c' y ξ)

theorem **IsKaehlerForm.of_anRead** {R : AnRead p σ d} {a : ChartForm p σ d} (h : IsKaehlerForm R a) (R' : AnRead p σ d) : IsKaehlerForm R' a

theorem **isPotential_curvForm** {R : AnRead p σ d} {B : ReciprocityTheory.ChartRestriction.G.Semi.LB Y} {φ : Weight p σ B} (h : φ.IsSmooth R) (i : B.ι) : IsPotential R (curvForm φ) (ptsIn p σ (B.U i)) (φ.φ i)

theorem **isKaehlerForm_curvForm** {R : AnRead p σ d} {B : ReciprocityTheory.ChartRestriction.G.Semi.LB Y} {φ : Weight p σ B} (h : φ.IsSmooth R) (hp : φ.StrictPos R) : IsKaehlerForm R (curvForm φ)

theorem **ddcForm_coord** (c : AlgChart p σ d) (j : Fin d) {y : CxPts p σ} (hy : y ∈ c.e.source) (ξ : Fin d → ℂ) : ddcForm (fun z => Complex.normSq (c.e z j)) c y ξ = Complex.normSq (ξ j) / Real.pi

theorem **isKaehlerForm_ddcForm** {R : AnRead p σ d} {φ : CxPts p σ → ℝ} (h : R.SmoothOn Set.univ φ) (hp : ∀ y, ∃ c : AlgChart p σ d, y ∈ c.e.source ∧ ∀ ξ, ξ ≠ 0 → 0 < ddcForm φ c y ξ) : IsKaehlerForm R (ddcForm φ)

theorem **act_vec** (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) (a : 𝒪 E) : actR Z.t.act a (f.vec τ i) = φ (embK c τ a) • f.vec τ i

theorem **existsUnique_sum** (τ : E →+* ℂ) (w : TanM x Z.t.G) (hw : ∀ a, actR Z.t.act a w = φ (embK c τ a) • w) : ∃! a : Fin (Dw.mult τ) → R, w = ∑ i, a i • f.vec τ i

theorem **linearIndependent_vec** (τ : E →+* ℂ) : LinearIndependent R (f.vec τ)

theorem **ptF_fst** : GlobalTheory.AllWt.ptF y ≫ pullback.fst _ _ = y.x

theorem **ptF_snd** : GlobalTheory.AllWt.ptF y ≫ pullback.snd _ _ = spMap (algebraMap c.F S)

theorem **lieEquiv_act** (a : 𝒪 E) (v : TanM y.x Xo.univ.t.G) : lieEquiv Z y hx (actR Xo.univ.t.act a v) = actR Z.t.act a (lieEquiv Z y hx v)

theorem **EigFrame.ofFrPt_v** (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) : ∃ w : Sp (DualNumber S) ⟶ Z.oF.t.G.left, IsTanAtG Z.oF.t (GlobalTheory.AllWt.ptF y) w ∧ w ≫ Z.m0.toSABMap.φ = y.u τ i ∧ w ≫ Z.m.φ = (EigFrame.ofFrPt Z y hx).v τ i

theorem **cxPt_congr** {y y' : FrPt Xo c (CAt σ)} (h : y.x = y'.x) : cxPt Z y = cxPt Z y'

theorem **not_liesIn_cxPt** (y : FrPt Xo c (CAt σ)) : ¬ LiesIn (cxPt Z y) (genBd Z)

theorem **liesIn_genBd_iff** (y : CxPts (genStr Z) σ) : LiesIn y (genBd Z) ↔ ∃ i, LiesIn y (x.cpt i)

theorem **nrm_pos** {y : CxPts (genStr Z) σ} (h : ¬ LiesIn y (genBd Z)) (i : Fin x.m) : 0 < x.nrm σ i y

theorem **m_eq_zero_iff** : x.m = 0 ↔ (genBd Z : Set (gen Z : Scheme.{0})) = ∅

theorem **gram_posDef** (f : EigFrame Z σ (y.1 ≫ genι Z)) (τ : E →+* ℂ) : (x.gram y f τ).PosDef

theorem **gram_change** (f g : EigFrame Z σ (y.1 ≫ genι Z)) (τ : E →+* ℂ) (T : Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) ℂ) (h : ∀ i, IsCombAtG Z.t (y.1 ≫ genι Z) (f.v τ) (fun a => T a i) (g.v τ i)) : x.gram y g τ = Tᴴ ∗ x.gram y f τ ∗ T

theorem **met_vec_eq_zero** (f g : EigFrame Z σ (y.1 ≫ genι Z)) {τ τ' : E →+* ℂ} (h : τ ≠ τ') (i : Fin (Dw.mult τ)) (i' : Fin (Dw.mult τ')) : x.met σ y (f.vec τ i) (g.vec τ' i') = 0

theorem **gram_smooth** (V : (gen Z).affineOpens) {φ : c.F →+* Γ(gen Z, V.1)} (F : EigFrame Z φ (LogKS.chart V ≫ genι Z)) (τ : E →+* ℂ) (a b : Fin (Dw.mult τ)) : ∃ g : CxPts (genStr Z) σ → ℂ, (manifold Z σ).SmoothOn (ptsIn _ σ V.1) (fun y => (g y).re) ∧ (manifold Z σ).SmoothOn (ptsIn _ σ V.1) (fun y => (g y).im) ∧ ∀ y (hy : y ∈ ptsIn _ σ V.1) (f : EigFrame Z σ (y.1 ≫ genι Z)), (∀ i, f.v τ i = spMap (dualMap (evHom hy)) ≫ F.v τ i) → x.gram y f τ a b = g y

theorem **gram_conj** {y' : CxPts (genStr Z) σ'} (e : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) (f : EigFrame Z σ (y.1 ≫ genι Z)) (f' : EigFrame Z σ' (y'.1 ≫ genι Z)) (τ : E →+* ℂ) (h : ∀ i, f'.v τ i = spMap (dualMap (starRingEnd ℂ)) ≫ f.v τ i) : x.gram y' f' τ = (x.gram y f τ).map (starRingEnd ℂ)

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ToroidalCompactificationData

theorem **frameCo_eq** (hp : X.p0 = r − j) {f : Frame X.Ψ r} (hf : f ∈ X.dom) (τ : X.Ψ) : frameCo X hj f τ = fromBlocks 1 (moeb (frameCo X hj f τ) 0) 0 1 ∗ fromBlocks (coW X hj f τ) 0 (frameCo X hj f τ).toBlocks₂₁ (coOm X hj f τ)

theorem **isUnit_coOm** (hp : X.p0 = r − j) {f : Frame X.Ψ r} (hf : f ∈ X.dom) (τ : X.Ψ) : IsUnit (coOm X hj f τ).det

theorem **isUnit_coW** (hp : X.p0 = r − j) {f : Frame X.Ψ r} (hf : f ∈ X.dom) (τ : X.Ψ) : IsUnit (coW X hj f τ).det

theorem **co_x0F** (τ : X.Ψ) : coW X hj (x0F X) τ = 1 ∧ coOm X hj (x0F X) τ = 1

theorem **hodge_co** (hp : X.p0 = r − j) {f : Frame X.Ψ r} (hf : IsOrth X f) (τ : X.Ψ) : (coW X hj f τ)ᴴ ∗ (hP (moeb (frameCo X hj f τ) 0))⁻¹ ∗ coW X hj f τ = 1 ∧ (coOm X hj f τ)ᴴ ∗ hQ (moeb (frameCo X hj f τ) 0) ∗ coOm X hj f τ = 1

theorem **Lv.relab_σ₀** (κ : Lv A) (τ : D.host.E →+* ℂ) : Formalism.L2.PjFree.relab κ.c κ.σ₀ τ = τ

theorem **Sig.mult_psi** (S : Sig A N j) (L : CmList A) (t : D.what₂.Tau) : S.Dw.mult (L.ψ t).1 = j

theorem **Sig.mult_conj** (S : Sig A N j) (L : CmList A) (t : D.what₂.Tau) : S.Dw.mult (conjEmb (L.ψ t).1) = N − j

theorem **j_le** : j ≤ N

theorem **p0_eq** : (s.fd k).p0 = N − j

theorem **zOf_mem** (f : (s.fd k).dom) : s.zOf L k f ∈ DomI D.what₂.Tau j (N − j)

theorem **zOf_frameOver** {z : Pt D.what₂.Tau j (N − j)} (hz : z ∈ DomI D.what₂.Tau j (N − j)) : s.zOf L k (s.frameOver L k z) = z

theorem **framePt_eq** {f f' : (s.fd k).dom} {g : D.what₂.Tau → GL (Fin j ⊕ Fin (N − j)) ℂ} (hg : g ∈ gammaU (s.fd k) s.j_le (s.lab L k)) (h : moebI g (s.zOf L k f) = s.zOf L k f') : s.framePt k f = s.framePt k f'

theorem **toCpt_zOf** (f : (s.fd k).dom) : s.toCpt L k (s.zOf L k f) = s.framePt k f

theorem **toCpt_moeb** {g : D.what₂.Tau → GL (Fin j ⊕ Fin (N − j)) ℂ} (hg : g ∈ gammaU (s.fd k) s.j_le (s.lab L k)) {z : Pt D.what₂.Tau j (N − j)} (hz : z ∈ DomI D.what₂.Tau j (N − j)) : s.toCpt L k (moebI g z) = s.toCpt L k z

theorem **not_liesIn_toCpt** (z : Pt D.what₂.Tau j (N − j)) : ¬ LiesIn (s.toCpt L k z) (genBd s.T.Z)

theorem **ToroidalCompactificationData.gramW_posDef** (k : s.Comp) (f : (s.fd k).dom) (t : D.what₂.Tau) : (x.gramW L k f t).PosDef

theorem **ToroidalCompactificationData.gramOm_posDef** (k : s.Comp) (f : (s.fd k).dom) (t : D.what₂.Tau) : (x.gramOm L k f t).PosDef

theorem **ToroidalCompactificationData.gram_hodge** (k : s.Comp) (f : (s.fd k).dom) (t : D.what₂.Tau) (h : ∀ τ, x.gram (s.framePt k f) (s.lieFrame k f) τ = (hodgeC (s.fd k) (s.lab L k) t : ℂ) • 1) : x.gramW L k f t = (hodgeC (s.fd k) (s.lab L k) t : ℂ) • 1 ∧ x.gramOm L k f t = (hodgeC (s.fd k) (s.lab L k) t : ℂ) • 1

theorem **ToroidalCompactificationData.met_posDef** (k : s.Comp) (z : Pt D.what₂.Tau j (N − j)) (t : D.what₂.Tau) : (x.metW L k z t).PosDef ∧ (x.metOm L k z t).PosDef

theorem **isBdOf_bdOf** (h : ∃ B, IsBdOf x L B) : IsBdOf x L (bdOf x L h)

theorem **nrm_zOf** (i : Fin (B k).m) (f : (s.fd k).dom) : (B k).nrm i (s.zOf L k f) = x.nrm κ.σ₀ (Fin.cast (h.m_eq k) i) (s.framePt k f)

theorem **r_eq** (hz : z ∈ DomI D.what₂.Tau j (N − j)) : (B k).r z = x.r κ.σ₀ (s.toCpt L k z)

theorem **m_ne_zero** (hb : (genBd s.T.Z : Set (gen s.T.Z : Scheme.{0})).Nonempty) : (B k).m ≠ 0

theorem **nrm_unique** (h' : IsBdOf x L B') (hz : z ∈ DomI D.what₂.Tau j (N − j)) (i : Fin (B k).m) : (B k).nrm i z = (B' k).nrm (Fin.cast ((h.m_eq k).trans (h'.m_eq k).symm) i) z

theorem **met_eq** (hz : z ∈ DomI D.what₂.Tau j (N − j)) (t : D.what₂.Tau) : (B k).metW z t = x.metW L k z t ∧ (B k).metOm z t = x.metOm L k z t

theorem **kae_unique** (h' : IsBdOf x L B') (hz : z ∈ DomI D.what₂.Tau j (N − j)) : (B k).kae z = (B' k).kae z

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.TrueColumnChoicesHeckeCharReferenceEqOne

theorem **afIso_tmul_one** (x : E) : afIso F E (x ⊗ₜ[F] 1) = algebraMap E _ x

theorem **mem_rInt_iff** (y : AfE F E) : y ∈ RInt F E ↔ ∀ P : HeightOneSpectrum (𝒪 E), Valued.v (afIso F E y P) ≤ 1

theorem **mem_suppAt_iff** (v : HeightOneSpectrum (𝒪 F)) (y : (AfE F E)ˣ) : y ∈ suppAt E v ↔ ∀ P : HeightOneSpectrum (𝒪 E), P.under (𝒪 F) ≠ v → afIso F E (y : AfE F E) P = 1

theorem **afIso_finAf** (a : Ideles E) : afIso F E (finAf F E a : AfE F E) = (a : AdeleRing (𝒪 E) E).2

theorem **finAf_surjective** : Function.Surjective (finAf F E)

theorem **Reads.unique** {μ : Chr F E K} {ξ ξ' : HChar E} (h : Reads σ μ ξ) (h' : Reads σ μ ξ') : ξ = ξ'

theorem **Reads.inj** {μ μ' : Chr F E K} {ξ : HChar E} (h : Reads σ μ ξ) (h' : Reads σ μ' ξ) : μ = μ'

theorem **reads_one** : Reads σ (1 : Chr F E K) (1 : HChar E)

theorem **Reads.mul** {μ μ' : Chr F E K} {ξ ξ' : HChar E} (h : Reads σ μ ξ) (h' : Reads σ μ' ξ') : Reads σ (μ ∗ μ') (ξ ∗ ξ')

theorem **Reads.apply** {μ : Chr F E K} {ξ : HChar E} (h : Reads σ μ ξ) (a : Ideles E) (y : (AfE F E)ˣ) (hy : afIso F E (y : AfE F E) = (a : AdeleRing (𝒪 E) E).2) : ((ξ.1 a : ℂˣ) : ℂ) = σ ((μ y : Kˣ) : K)

theorem **Reads.orderDvd** {μ : Chr F E K} {ξ : HChar E} (h : Reads σ μ ξ) {n : ℕ} (hn : μ ^ n = 1) : ξ.OrderDvd n

theorem **chiId_ne_zero** {L : Type} [Field L] [NumberField L] (χ : HChar L) (𝔞 : Submodule (𝒪 L) L) : chiId χ 𝔞 ≠ 0

theorem **chiId_span_singleton** {L : Type} [Field L] [NumberField L] (χ : HChar L) (u : Lˣ) : chiId χ (Submodule.span (𝒪 L) {(u : L)}) = ((∏ᶠ w : HeightOneSpectrum (𝒪 L), χ.val w ^ (ordF w u).toNat : ℂˣ) : ℂ)

theorem **chiId_one** {L : Type} [Field L] [NumberField L] (χ : HChar L) : chiId χ 1 = 1

theorem **detId_std** {R : Ranks} (H : Host R) (N : ℕ) : Th.detId H (Th.stdL H N) (Th.stdL H N) = 1

theorem **AmbIn.δ_ne_zero** {R : Ranks} {H : Host R} (J : AmbIn H) : J.δ ≠ 0

theorem **glob_σ₀** : I.glob.σ₀ = B.embLL

theorem **globalXi_reference** : I.glob.xi0 = B.host.xi0

theorem **globalCharacters_reference** : I.glob.mu0 = I.C.mu0

theorem **A_σ₀** : I.A.σ₀ = B.embLL

theorem **A_δ** : I.A.δ = I.J.δ

theorem **A_cD** : I.A.cD = I.J.cD

theorem **A_AA** : I.A.AA = (B.host.AA0 : Submodule (𝒪 B.host.E) B.host.E)

theorem **Q_subset_SQ** (a : I.ι) : I.Q a ⊆ I.SQ

theorem **mem_SQ** {P : HeightOneSpectrum (𝒪 B.host.E)} : P ∈ I.SQ ↔ ∃ a, P ∈ I.Q a

theorem **twistPrimes_reference_tuple_eq_empty** : I.Q I.mu0 = ∅

theorem **exists_of_mem_Q** {a : I.ι} {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ I.Q a) : ∃ v ∈ Qprime (I.tup a : Chr B.fields.F B.fields.E B.fields.K), I.sel v = P ∧ GoodPl B.fields.E I.S v

theorem **Ok.trueColumnChoices_heckeChar_reference_eq_one** (h : I.Ok) : I.xi I.mu0 = 1

theorem **Ok.xi_inj** (h : I.Ok) : Function.Injective I.xi

theorem **Ok.fromHecke** (h : I.Ok) : I.Ch.FromHecke B.wts B.embLL

theorem **Ok.mem_of_dvd_QSq** (h : I.Ok) {p : ℕ} (hp : p.Prime) (hd : p ∣ B.host.QS * B.host.q) : p ∈ I.S

theorem **Ok.mem_of_dvd_QS** (h : I.Ok) {p : ℕ} (hp : p.Prime) (hd : p ∣ B.host.QS) : p ∈ I.S

theorem **Ok.small_mem** (h : I.Ok) {p : ℕ} (hp : p ∈ B.host.small) : p ∈ I.S

theorem **Ok.two_q_d_mem** (h : I.Ok) : 2 ∈ I.S ∧ B.host.q ∈ I.S ∧ B.host.d ∈ I.S

theorem **Ok.not_dvd_condN** (h : I.Ok) {p : ℕ} (hp : p.Prime) (hS : p ∉ I.S) : ¬ p ∣ B.host.condN

theorem **Ok.hm** (h : I.Ok) : B.R.m = (B.R.m - 2) + 2

theorem **seamOf_ok** : (seamOf B C W V S N G).Ok

theorem **seamOf_lat** {n : ℕ} (M : Submodule (𝒪 B.host.E) (Fin n → B.host.E)) : (seamOf B C W V S N G).lat M = M

theorem **AmbIn.amb_eq_ambOf** (J : AmbIn B.host) (χ : Submodule (𝒪 B.host.E) B.host.E → ℂ) : J.amb B χ = ambOf (seamOf B C W V S N G) (J.toRec B C W V S N G) χ J.cD

theorem **AmbIn.ok_iff** (J : AmbIn B.host) : J.Ok ↔ TaylorLogMetric.IsDiscriminantTwelfthRootOfIdealLattice (assemble B C W V S N G) ⊤ J.δ ∧ IsDgen (assemble B C W V S N G) J.cD

theorem **duStarOf_eq** (u : B.Word) : duStarOf B.host (B.hcW u) = StandardLattice.duStarU (assemble B C W V S N G) u

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.TwistedLevelExtension

theorem **scal_scl** {n : ℕ} (q : ℕ) (e : ℕ → ℤ) {u : E} (h0 : u ≠ 0) (hu : P.valuation E u = 1) : LinearEquiv.smulOfNeZero E (Sp E n) u h0 ∈ stabL E n (LocE.sclLat P q n e)

theorem **scal_std** {n : ℕ} {u : E} (h0 : u ≠ 0) (hu : P.valuation E u = 1) : LinearEquiv.smulOfNeZero E (Sp E n) u h0 ∈ stabL E n (stdAt E n P)

theorem **scal_prX** {n c kj q : ℕ} (hc : c = 0) {u : E} (h0 : u ≠ 0) (hu : P.valuation E u = 1) : LinearEquiv.smulOfNeZero E (Sp E n) u h0 ∈ Loc.prX (Loc.Cbe (LocE.JQ P q kj n c) 1)

theorem **vq_lt** : P.valuation E (qOf P : E) < 1

theorem **q_ne** : (qOf P : E) ≠ 0

theorem **v_one_add** : P.valuation E (1 + (qOf P : E)) = 1

theorem **scal_not** {n r c : ℕ} (hn : 0 < n) (hc : 1 < c) (h0 : (1 + (qOf P : E)) ≠ 0) : ((LinearEquiv.smulOfNeZero E (Sp E n) _ h0, scal_std P h0 (v_one_add P)) : GLint E n P) ∉ twK1H P n r c

theorem **trv_apply** {n : ℕ} {a b : Fin n} (h : a ≠ b) (x : Sp E n) : trv h x = x + Pi.single a (x b)

theorem **single_mem** {n : ℕ} {a b : Fin n} (e : ℕ → ℤ) (he : e a ≤ e b) {x : Sp E n} (hx : x ∈ LocE.sclLat P (qOf P) n e) : (Pi.single a (x b) : Sp E n) ∈ LocE.sclLat P (qOf P) n e

theorem **trv_scl** {n : ℕ} {a b : Fin n} (h : a ≠ b) (e : ℕ → ℤ) (he : e a ≤ e b) : trv h ∈ stabL E n (LocE.sclLat P (qOf P) n e)

theorem **trv_prX** {n c k rS r t : ℕ} (hc : c = 0) {a b : Fin n} (h : a ≠ b) (ha : a.1 = 0) (hk : 0 < k) (hr : r = k + 2 * rS + 1) (hb : r ≤ b.1) : trv h ∈ Loc.prX (Loc.Cbe (LocE.JGsym P (qOf P) k rS r t n c) 1)

theorem **trv_red** {n : ℕ} {a b : Fin n} (h : a ≠ b) {x : Sp E n} (hx : x ∈ stdAt E n P) : redV P (trv h x) = (Matrix.transvection a b (1 : 𝒪 E ⧸ P.asIdeal)).mulVec (redV P x)

theorem **not_ok_Q** (Λt) {P} (hP : P ∈ QE sel (mu : Chr X.F X.E X.K)) (hG : P ∉ Pr.vGs) : ¬ (twIdxTop Ch sel mu cst m2 Λt (twTt sel mu)).Ok Ch sel mu cst G m2

theorem **not_ok_G** (θt) (hs : cst.sharp = true) {w} (hw : w ∈ Pr.vGs) : ¬ (twIdxTop Ch sel mu cst m2 (twLt Ch) θt).Ok Ch sel mu cst G m2

theorem **not_ok_of_idx_top** {B : Base} {Y : PreX B.toPre} (I : TC B Y) {a : I.ι} (ht : I.idx a = twIdxTop I.Ch I.sel (I.tup a) (I.cst a) (B.R.m - 2) (twLt I.Ch) (twTt I.sel (I.tup a))) : ¬ I.Ok

theorem **exists_deep** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X), StagesInhab.HasLetters B.toPre ∧ I.Ok ∧ TC.Pins B X ∧ I.Deep ∧ ∀ a, (I.cst a).A = B.R.m

theorem **Q_star_nonempty** {B : Base} {X : PreX B.toPre} {I : TC B X} (h : I.Ok) : (I.Q I.muStar).Nonempty

theorem **exists_isTw** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws), I.Ok ∧ I.IsTw ∧ TC.Pins B X ∧ I.Law0 l ∧ (I.Q I.muStar).Nonempty ∧ Nonempty (TC.Later B (I.cols l))

theorem **exists_ok_not_isTw** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X), I.Ok ∧ I.Deep ∧ ¬ I.IsTw

theorem **topDep_eq** {B : Base} {X : PreX B.toPre} {I : TC B X} (ht : I.IsTw) (a : I.ι) : topDep B.R (I.cst a).M + 1 = twTopDepth B.R

theorem **kerEquivC_of_isTw** {B : Base} {X : PreX B.toPre} {I : TC B X} (ht : I.IsTw) (h : I.Ok) (W : Wts B.fields (B.primes X).Word B.R) (a : I.ι) (u : B.Word) (hOk : ChTh.Ok (Ch

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.UpLReStage

theorem **sx_eq** {w : ColW B f T} {Z : w.Lt} {Wt : Set B.places.Pl} {p0 : BaseLevelRowSide (w.D Z)} (r : RowsT K w Z Wt p0) (col : ColT w Z) : SX K B f T w Z Wt p0 r col = (pkOf r).SAt fun u => (col.C.forms u).to0

theorem **SameCar.refl** {D : D7} (F : ∀ u, Forms0 D u) : SameCar F F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.ValTransOfEq

theorem **hilbTie_ofEq** (e : P = P') (h : HilbTie P ip j) : HilbTie P' ip (j.trans (LinearEquiv.ofEq _ _ e))

theorem **projTie_ofEq** {Y : Type} {f : Y → X} {g g' : Y → N} (e : P = P') (h : ProjTie P ip j f g) (hg : ∀ y, g' y = g y) : ProjTie P' ip (j.trans (LinearEquiv.ofEq _ _ e)) f g'

theorem **le_of_eqs** {Q Q' : Submodule ℂ N} (e : P = P') (e' : Q = Q') (h : P ≤ Q) : P' ≤ Q'

theorem **val_trans_ofEq** (e : P = P') (x : X) : ((j.trans (LinearEquiv.ofEq _ _ e)) x).1 = (j x).1

theorem **map_incl** {Q Q' : Submodule ℂ N} {Z : Type} (e : P = P') (e' : Q = Q') (h : P ≤ Q) (j : X ≃ₗ[ℂ] P) {f : Q → Z} (f' : Q' → Z) (hf : ∀ v, f' (LinearEquiv.ofEq _ _ e' v) = f v) (y : X) : f' (Submodule.inclusion (le_of_eqs e e' h) ((j.trans (LinearEquiv.ofEq _ _ e)) y)) = f (Submodule.inclusion h (j y))

theorem **stepB_eq** (l : Bool) (μ : a.D.what₂.Tup) (k : ℕ) (v : (toChainReal R F tupT hm hTop).CC l μ (k + 1)) : stepB c l μ k (LinearEquiv.ofEq _ _ (congrFun (congrFun (congrFun (spCC_eq c F tupT hm hTop) l) μ) (k + 1)) v) = ((toChainReal R F tupT hm hTop).ioC l μ k v).1

theorem **topB_eq** (l : Bool) (μ : a.D.what₂.Tup) (v : (toChainReal R F tupT hm hTop).CCX l μ) : topB c l μ (LinearEquiv.ofEq _ _ (congrFun (congrFun (spCCX_eq c F tupT hm hTop) l) μ) v) = ((toChainReal R F tupT hm hTop).ioX l μ v).1

theorem **rowSideReal_tr** {I : RowIds X (toChainReal R F tupT hm hTop) l} {n : ℕ} {bs : OrthonormalBasis (Fin n) ℂ EE} {ty : Fin n → a.D.what₂.FT} (H : RowSideReal X (toChainReal R F tupT hm hTop) l I bs ty) : RowSideReal (rowSide c X) (chainB c.to F tupT hm hTop) l (rowIds c I) bs ((ty))

theorem **spWc_eq** : (toLiftReal R F tupT hm hTop hinj).Wc = spWc c F tupT hm hTop hinj

theorem **spMultW_eq** : (toLiftReal R F tupT hm hTop hinj).MultW = spMultW c F tupT hm hTop hinj

theorem **liftTie_tr** {I : LiftIds (toLiftReal R F tupT hm hTop hinj)} (H : LiftTie (toLiftReal R F tupT hm hTop hinj) I) : LiftTie (liftB c.to F tupT hm hTop hinj) (liftIds c I)

theorem **recTied_tr** {X0 : BaseLevelRowSide a.D} {Ish : RowIds (RowSide.sharp a.D) (toChainReal R F tupT hm hTop) true} {Iz : RowIds X0.s (toChainReal R F tupT hm hTop) false} {IL : LiftIds (toLiftReal R F tupT hm hTop hinj)} {K : ColReal a.D} (H : RecTied a.D X0 (toChainReal R F tupT hm hTop) Ish Iz (toLiftReal R F tupT hm hTop hinj) IL K) : RecTied c.to.D (baseLevelRowSideTransport c X0) (chainB c.to F tupT hm hTop) ((rowIds c Ish)) ((by with_unfolding_all exact @rowIds B C a c G ‹Group G› ‹Fintype G› Pl0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks›› iK A R F tupT hm hTop (@Rows.Data.EE0 (@Late.D B C a).R (@Late.D B C a).what₂) (@Rows.Data.instEE0 (@Late.D B C a).R (@Late.D B C a).what₂) (@Rows.Data.instEE0i (@Late.D B C a).R (@Late.D B C a).what₂) (@Rows.Data.instEE0f (@Late.D B C a).R (@Late.D B C a).what₂) (@BaseLevelRowSide.s (@Late.D B C a) X0) false Iz)) (liftB c.to F tupT hm hTop h

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.VolumeLeOfTrivialFirstPair

theorem **volume_le_of_trivial_first_pair** (hok : A3.fam3.Ok3 s) (hK : X.KFormL) (hJ : X.JIsoL (cvOf3 A3 P0 ψ0 P1 ψ1)) {gW : ∀ j, Representation ℂ G (X.W j)} (hU : LiftActionUnitary X gW) (hG : LiftGroupEquivariance A3 X gW) (j : Fin m) (t : T) : X.goodPartComplexPoints (cvOf3 A3 P0 ψ0 P1 ψ1) false j t ≤ X.goodPartComplexPoints (cvOf3 A3 ⊥ 1 P1 ψ1) false j t

theorem **exists_crd** (hok : A3.fam3.Ok3 s) (hK : X.KFormL) (hJ : X.JIsoL (cvOf3 A3 ⊥ 1 P1 ψ1)) (hRZ : X.RatZ) {gW : ∀ j, Representation ℂ G (X.W j)} (hU : LiftActionUnitary X gW) (hG : LiftGroupEquivariance A3 X gW) (j : Fin m) (t : T) (w : X.W j) (hw : LinearMap.adjoint (X.θ' j t) w ∈ X.goodPartComplexPoints (cvOf3 A3 ⊥ 1 P1 ψ1) false j t) : ∃ c, lin (X.uL (crdOf3 A3 ⊥ 1 P1 ψ1 Dc Sp Zsc) false j t) c = LinearMap.adjoint (X.θ' j t) w

theorem **range_w0C_le** (hok : A3.fam3.Ok3 s) (hK : X.KFormL) (hJ : X.JIsoL (cvOf3 A3 P0 ψ0 P1 ψ1)) (hRZ : X.RatZ) {gW : ∀ j, Representation ℂ G (X.W j)} (hU : LiftActionUnitary X gW) (hG : LiftGroupEquivariance A3 X gW) (hL0 : LevelTypeOfLifts A3 X (P1

theorem **pHecke** (h : R.PHecke (bare3 _ P0 ψ0 P1 ψ1)) : (dat3 f μ R mu tw).PHecke (bare3 _ P0 ψ0 P1 ψ1)

theorem **liftOnto** (h : R.PHecke (bare3 _ P0 ψ0 P1 ψ1)) (ho : ∀ hW, HeckeIntertwiningLifts (A3.comap id (f μ)) (R.toCx3 (bare3 _ P0 ψ0 P1 ψ1)) hW → U2.OntoC (A3.comap id (f μ)) (R.toCx3 (bare3 _ P0 ψ0 P1 ψ1)) P0 ψ0 P1 ψ1 hW) : ∀ hW, HeckeIntertwiningLifts (A3.comap f μ) ((dat3 f μ R mu tw).toCx3 (bare3 _ P0 ψ0 P1 ψ1)) hW → U2.OntoC (A3.comap f μ) ((dat3 f μ R mu tw).toCx3 (bare3 _ P0 ψ0 P1 ψ1)) P0 ψ0 P1 ψ1 hW

theorem **pLevW** {s : ℕ} (hok : (A3.comap id (f μ)).fam3.Ok3 s) (hs : (A3.comap id (f μ)).fam3.TypesSep3) (hC : R.WeakCharacterisation (bare3 _ P0 ψ0 P1 ψ1)) (h : PrimRows3 R (bare3 _ P0 ψ0 P1 ψ1)) : (dat3 f μ R mu tw).PLevW (bare3 _ P0 ψ0 P1 ψ1)

theorem **primRows3** {s : ℕ} (hok : (A3.comap id (f μ)).fam3.Ok3 s) (hs : (A3.comap id (f μ)).fam3.TypesSep3) (hC : R.WeakCharacterisation (bare3 _ P0 ψ0 P1 ψ1)) (h : PrimRows3 R (bare3 _ P0 ψ0 P1 ψ1)) : PrimRows3 (dat3 f μ R mu tw) (bare3 _ P0 ψ0 P1 ψ1) where

theorem **intersectionNumber_baseChange** (hf : Function.Injective f) (hC : R.WeakCharacterisation (bare3 _ P0 ψ0 P1 ψ1)) (h : Holds R (bare3 _ P0 ψ0 P1 ψ1)) : Holds (dat3 f μ R mu tw) (bare3 _ P0 ψ0 P1 ψ1) where

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.WdPstCycRepNeTrivWD

theorem **cycRep_apply** (K LL : FinExt l) (g : ɪ(GK K)) (v : ɪLL.1) : cycRep P K LL g v = (P.χQ (g : GQ l) : ℚ_[l]) • v

theorem **stRep_per** (K LL : FinExt l) (g : ɪ(GK K)) : stRep P (cycRep P K LL) g ((1 : ɪLL.1) ⊗ₜ[ℚ_[l]] (C (tInv P) : Bst P)) = (1 : ɪLL.1) ⊗ₜ C (tInv P)

theorem **per_ne_zero** (K LL : FinExt l) : per P K LL ≠ 0

theorem **phiD_per** (K LL : FinExt l) : phiD P _ (per P K LL) = (l : ɪLL.1)⁻¹ • per P K LL

theorem **phiD_pow_per** (K LL : FinExt l) (n : ℕ) : (phiD P _ ^ n) (per P K LL) = ((l : ɪLL.1)⁻¹ ^ n) • per P K LL

theorem **wdMk_per_ne_zero** (K LL : FinExt l) : wdMk P ι _ (per P K LL) ≠ 0

theorem **isPst_cycRep** (h : ∀ (K : FinExt l) (b : P.R.B), ¬ ∀ g ∈ GK K, P.actQ g b − b = (kummer P g : ℚ_[l]) • P.tB) (K : FinExt l) : IsPst P (cycRep P K (botExt l))

theorem **weilD_of_phiD** [FiniteDimensional ɪ(MaximalUnramifiedExtension l) (DpstUr P ρ)] (hj : Induces Bw) (hx : ∀ g, actD P ρ g x = x) (hφ : ∀ n : ℕ, (phiD P ρ ^ n) x = ((l : ɪLL.1)⁻¹ ^ n) • x) (σ : ɪ(Weil h)) (hσ : deg σ = 1) : weilD Bw h P ρ σ x = ((normV w : ℕ) : ɪLL.1) • x

theorem **apply_eq_of_nilpotent** {W : Type} [AddCommGroup W] [Module ℂ W] (hd : finrank ℂ W = 1) {v : W} (hv : v ≠ 0) (u : End ℂ W) (hu : IsNilpotent (u − 1)) : u v = v

theorem **frobEig_wdPst_of_phiD** [FiniteDimensional ɪLL.1 V] (h1 : finrank ɪLL.1 V = 1) (hV : IsPst P ρ) (hx : ∀ g, actD P ρ g x = x) (hφ : ∀ n : ℕ, (phiD P ρ ^ n) x = ((l : ɪLL.1)⁻¹ ^ n) • x) (hv : wdMk P ι ρ x ≠ 0) : (wdPst P ι Bw h hj ρ).frobEig = {((normV w : ℕ) : ℂ)}

theorem **frobEig_wdPst_cycRep** (hV : IsPst P (cycRep P Bw.Kw LL)) : (wdPst P ι Bw h hj (cycRep P Bw.Kw LL)).frobEig = {((normV w : ℕ) : ℂ)}

theorem **wdPst_cycRep_ne_trivWD** (hV : IsPst P (cycRep P Bw.Kw LL)) : wdPst P ι Bw h hj (cycRep P Bw.Kw LL) ≠ trivWD h 1

theorem **lpDim_wdAt** {a : ℕ} (r : ɪ(Dec (F

theorem **exists_induces** (hw : w ∈ placesOver E l) (h : IsPlaceOver A w) : ∃ (LL : FinExt l) (Bw : LocBridge l E Ω (Dec (F

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.WeakCharInertPartRealAdelicDataTransport

theorem **_root_.GlobalTheory.Mem.R3.SameGrp.pl_back** (s : SameGrp a b) (kv : Σ k : Fin 2, b.D.nd.PlG k) : (a.D.naut.plGw kv.1 ((s.e kv.1).symm kv.2) : HeightOneSpectrum (𝒪 B.host.E)) = b.D.naut.plGw kv.1 kv.2

theorem **grpCl_tr** (s : SameGrp a b) {Pg c Dv eG} {eR : G ≃* G0g a.D.nd} (h : GrpCl a.D Pg c Dv eG eR) : GrpCl b.D ((Pg)) c ((Dv)) ((eG)) (eR.trans s.iso)

theorem **psiPPTie_transport** (s : SameGrp a b) {P0 P1 : Subgroup G} {ψ0 ψ1} {eR : G ≃* G0g a.D.nd} (h : PsiPPTie a.D P0 ψ0 P1 ψ1 eR) : PsiPPTie b.D P0 ((ψ0)) P1 ((ψ1)) (eR.trans s.iso)

theorem **grpRecord_tr** (s : SameGrp a b) (h : R.grp_record) : letI

theorem **isGrp_tr** (s : SameGrp a b) (h : R.IsGrp F) : letI

theorem **baseW_tr** (s : SameGrp a b) (h : R.BaseW F) : letI

theorem **keep_tr** (s : SameGrp a b) {cG : G0 R.Pg a.D.R.r ≃* G0g a.D.nd} (h : Keep (A3

theorem **keep_back** (s : SameGrp b a) {cG : G0 (levCh b R.toLevCh).Pg b.D.R.r ≃* G0g b.D.nd} (h : @Keep b.D G _ _ Pl0 I0 T (famInThreeCoefficientChange b A) Ks _ _ (alg b iK) ((by with_unfolding_all exact ((real b R) :))) ((by with_unfolding_all exact ((face b F) :))) ((by with_unfolding_all exact (cG :)))) : Keep (A3

theorem **weakCharInertPart_realAdelicDataTransport** (s : SameGrp a b) (h : R.WeakCharacterisation F) : letI

theorem **pins_tr** (s : SameGrp a b) {cG : G0 R.Pg a.D.R.r ≃* G0g a.D.nd} (h : Pins R F cG) : letI

theorem **holds_tr** (h : Holds R F) : letI

theorem **errEq_tr** (h : ErrEq R F) : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.WeilDeligneParamMonodromyEqZeroIff

@[simp] theorem **tval_mul** (σ τ : Ine (F

@[simp] theorem **tval_one** : tval t 1 = 0

@[simp] theorem **tval_pow** (σ : Ine (F

theorem **isTameChar_tameL** (h : IsPlaceOver A w) (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) : IsTameChar (tameL h hl)

theorem **IsTameChar.exists_mul** (h : IsPlaceOver A w) (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (ht' : IsTameChar t') : ∃ c : Qbar l, c ≠ 0 ∧ ∀ σ, tval t σ = c * tval t' σ

theorem **IsTameChar.tval_ineConj** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (g : Weil h) (σ : Ine (F

theorem **IsTameChar.exists_tval_ne_zero** (ht : IsTameChar t) {U : Subgroup (Ine (F

@[simp] theorem **ineIncl_inePart** {φ : Weil h} (hφ : deg φ = 1) (g : Weil h) : ineIncl h (inePart hφ g) = (φ ^ deg g)⁻¹ * g

@[simp] theorem **inePart_zpow_mul** {φ : Weil h} (hφ : deg φ = 1) (k : ℤ) (σ : Ine (F

@[simp] theorem **inePart_ineIncl** {φ : Weil h} (hφ : deg φ = 1) (σ : Ine (F

theorem **inePart_mul** {φ : Weil h} (hφ : deg φ = 1) (g g' : Weil h) : inePart hφ (g * g') = ineConj (φ ^ deg g')⁻¹ (inePart hφ g) * inePart hφ g'

theorem **toEndC_apply** (ι : Qbar l ≃+* ℂ) (M : Matrix n n (Qbar l)) : toEndC ι M = Matrix.toLin' (M.map ι)

theorem **toEndC_smul** (ι : Qbar l ≃+* ℂ) (c : Qbar l) (M : Matrix n n (Qbar l)) : toEndC ι (c • M) = ι c • toEndC ι M

theorem **toEndC_injective** (ι : Qbar l ≃+* ℂ) : Function.Injective (toEndC (n

theorem **charpoly_toEndC** (ι : Qbar l ≃+* ℂ) (M : Matrix n n (Qbar l)) : (toEndC ι M).charpoly = M.charpoly.map ι.toRingHom

theorem **IsMonodromy.of_nhds** (hN : IsNilpotent N) (hr : ∀ᶠ σ : Ine (F

theorem **isMonodromy_zero_iff** : IsMonodromy t r 0 ↔ IsSmooth (r.comp (Weil h).subtype)

theorem **IsMonodromy.unique** (ht : IsTameChar t) (hN : IsMonodromy t r N) (hN' : IsMonodromy t r N') : N = N'

theorem **IsMonodromy.smul** {c : Qbar l} (hc : ∀ σ, tval t σ = c * tval t' σ) (hN : IsMonodromy t r N) : IsMonodromy t' r (c • N)

theorem **IsMonodromy.comm** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (g : Weil h) : (r g : Matrix n n (Qbar l)) * N = ((normV w : ℕ) : Qbar l) ^ deg g • (N * r g)

theorem **IsMonodromy.mul_exp** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (g : Weil h) (s : Qbar l) : (r g : Matrix n n (Qbar l)) * exp (s • N) = exp ((((normV w : ℕ) : Qbar l) ^ deg g * s) • N) * r g

theorem **IsTameChar.tval_inePart_mul** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hφ : deg φ = 1) (g g' : Weil h) : tval t (inePart hφ (g * g')) = (((normV w : ℕ) : Qbar l) ^ deg g')⁻¹ * tval t (inePart hφ g) + tval t (inePart hφ g')

theorem **wdMat_zpow_mul** (hφ : deg φ = 1) (k : ℤ) (σ : Ine (F

theorem **wdMat_ineIncl** (hφ : deg φ = 1) (σ : Ine (F

@[simp] theorem **wdMat_zero** (hφ : deg φ = 1) (g : Weil h) : wdMat t hφ r 0 g = (r g : Matrix n n (Qbar l))

theorem **wdMat_one** (hφ : deg φ = 1) : wdMat t hφ r N 1 = 1

theorem **IsMonodromy.wdMat_mul** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (g g' : Weil h) : wdMat t hφ r N (g * g') = wdMat t hφ r N g * wdMat t hφ r N g'

theorem **IsMonodromy.wdMat_comm** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (g : Weil h) : wdMat t hφ r N g * N = ((normV w : ℕ) : Qbar l) ^ deg g • (N * wdMat t hφ r N g)

theorem **IsMonodromy.exists_conj** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (hφ' : deg φ' = 1) : ∃ b : Qbar l, ∀ g, exp (b • N) * wdMat t hφ r N g = wdMat t hφ' r N g * exp (b • N)

theorem **IsMonodromy.charpoly_wdMat** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (g : Weil h) : (wdMat t hφ r N g).charpoly = (r g : Matrix n n (Qbar l)).charpoly

@[simp] theorem **IsMonodromy.rep_apply** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (ι : Qbar l ≃+* ℂ) (g : Weil h) : hN.rep hl ht hφ ι g = toEndC ι (wdMat t hφ r N g)

theorem **IsMonodromy.isSmooth_rep** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (ι : Qbar l ≃+* ℂ) : IsSmooth (hN.rep hl ht hφ ι)

theorem **IsMonodromy.rep_comm** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (ι : Qbar l ≃+* ℂ) (g : Weil h) : hN.rep hl ht hφ ι g * toEndC ι N = ((normV w : ℕ) : ℂ) ^ deg g • (toEndC ι N * hN.rep hl ht hφ ι g)

theorem **IsMonodromy.exists_isFrobSS** (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (ι : Qbar l ≃+* ℂ) : ∃ a : WDRep h (n → ℂ), IsFrobSS (fun g => toEndC ι (wdMat t hφ r N g)) (toEndC ι N) a

theorem **IsWDOf.unique** {x y : WD h} (hx : IsWDOf ι r x) (hy : IsWDOf ι r y) : x = y

theorem **IsWDOf.wdOf_eq** {x : WD h} (hx : IsWDOf ι r x) : wdOf h hl ι r = x

theorem **exists_isWDOf_iff** (ht : IsTameChar t) : (∃ x : WD h, IsWDOf ι r x) ↔ ∃ N, IsMonodromy t r N

theorem **IsMonodromy.wdOf_spec** (ht : IsTameChar t) (hN : IsMonodromy t r N) (hφ : deg φ = 1) (ι : Qbar l ≃+* ℂ) : ∃ a : WDRep h (n → ℂ), IsFrobSS (hN.rep hl ht hφ ι) (toEndC ι N) a ∧ wdOf h hl ι r = WD.mk a

theorem **wdOf_of_not** (H : ¬ ∃ x : WD h, IsWDOf ι r x) : wdOf h hl ι r = WD.mk (WDRep.diag h fun _ : n => 1)

theorem **lpDim_wdOf** (ι : Qbar l ≃+* ℂ) (r : G E Ω →* GL n (Qbar l)) : (wdOf h hl ι r).lpDim = Fintype.card n

theorem **IsMonodromy.charPol_wdOf** (ht : IsTameChar t) (hN : IsMonodromy t r N) (ι : Qbar l ≃+* ℂ) (g : Weil h) : (wdOf h hl ι r).charPol g = (r g : Matrix n n (Qbar l)).charpoly.map ι.toRingHom

theorem **IsMonodromy.weilDeligneParam_monodromy_eq_zero_iff** (ht : IsTameChar t) (hN : IsMonodromy t r N) (ι : Qbar l ≃+* ℂ) : (wdOf h hl ι r).mono0 ↔ N = 0

theorem **IsMonodromy.unr_wdOf_iff** (ht : IsTameChar t) (hN : IsMonodromy t r N) (ι : Qbar l ≃+* ℂ) : (wdOf h hl ι r).unr ↔ ∀ σ ∈ Ine (F

theorem **wdOf_eq_mk** (a : WDRep h (n → ℂ)) (ha : a.N = 0) (hr : ∀ g : Weil h, toEndC ι (r g : Matrix n n (Qbar l)) = a.ρ g) : wdOf h hl ι r = WD.mk a

theorem **IsWDOf.of_conj** {r' : G E Ω →* GL n (Qbar l)} (P : GL n (Qbar l)) (hP : ∀ g : Weil h, P * r g = r' g * P) {x : WD h} (hx : IsWDOf ι r x) : IsWDOf ι r' x

theorem **wdOf_eq_of_conj** {r' : G E Ω →* GL n (Qbar l)} (P : GL n (Qbar l)) (hP : ∀ g : Weil h, P * r g = r' g * P) : wdOf h hl ι r = wdOf h hl ι r'

theorem **coe_kummerRep** (a : Eˣ) (σ : G E Ω) : (kummerRep l a σ : Matrix (Fin 2) (Fin 2) (Qbar l)) = !![algebraMap ℚ_[l] (Qbar l) ((cycZ E Ω l σ : ℤ_[l]ˣ) : ℤ_[l]), algebraMap ℚ_[l] (Qbar l) (kummer l a σ); 0, 1]

theorem **continuous_kummerRep** (a : Eˣ) : Continuous fun σ : G E Ω => (kummerRep l a σ : Matrix (Fin 2) (Fin 2) (Qbar l))

theorem **isMonodromy_kummerRep** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) : IsMonodromy (tameL h hl) (kummerRep l (unifUnit w)) !![0, 1; 0, 0]

theorem **wdOf_kummerRep** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ι : Qbar l ≃+* ℂ) : (wdOf h hl ι (kummerRep l (unifUnit w))).lpDim = 2 ∧ ¬(wdOf h hl ι (kummerRep l (unifUnit w))).mono0 ∧ (wdOf h hl ι (kummerRep l (unifUnit w))).charPol (frob h) = (X - C ((normV w : ℕ) : ℂ)) * (X - 1)

theorem **wdOf_cyc** (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (ι : Qbar l ≃+* ℂ) : wdOf h hl ι ((Matrix.GeneralLinearGroup.scalar (Fin 1)).comp (cyc E Ω l)) = (WD.mk (WDRep.diag h fun _ : Fin 1 => 1)).lpTw 1

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.Wp3ApplyEqSqMulSubSqMul

theorem **eec_eq** : R.EEc = (by with_unfolding_all exact ((letI

theorem **wc_eq** : R.Wc' = (by with_unfolding_all exact ((letI

theorem **carries_eq** : R.Carries = (by with_unfolding_all exact ((letI

theorem **realAdelicInertTransport_typeRows_eq** : R.EEp3 = (by with_unfolding_all exact ((letI

theorem **wp3_apply_eq_sq_mul_sub_sq_mul** : R.Wp3 = (by with_unfolding_all exact ((letI

theorem **parts_eq** (_F : Face3 a.D A) : ((R.EEp3, R.toRealDat.EEat, R.Wp3) : Parts R) = ((by with_unfolding_all exact ((letI

theorem **toCx3_eq** : R.toCx3 F = (@RealDat3.toCx3 (@Late.D B C b) G ‹Group G› ‹Fintype G› Pl0 I0 T (@famInThreeCoefficientChange B C a b G ‹Group G› ‹Fintype G› Pl0 I0 T A) Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks››› (@alg B C a b Ks ‹Field Ks› iK) (@real3 B C a b G ‹Group G› ‹Fintype G› Pl0 I0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks››› iK A R) (@face3 B C a b G ‹Group G› ‹Fintype G› Pl0 I0 T A F))

theorem **trISem_eq** : R.TrISem F = (@RealDat3.TrISem (@Late.D B C b) G ‹Group G› ‹Fintype G› Pl0 I0 T (@famInThreeCoefficientChange B C a b G ‹Group G› ‹Fintype G› Pl0 I0 T A) Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks››› (@alg B C a b Ks ‹Field Ks› iK) (@real3 B C a b G ‹Group G› ‹Fintype G› Pl0 I0 T Ks ‹Field Ks› ‹@NumberField Ks ‹Field Ks››› iK A R) (@face3 B C a b G ‹Group G› ‹Fintype G› Pl0 I0 T A F))

theorem **isInert_tr** (h : R.IsInert) : letI

theorem **isTup_tr** (h : R.IsTup) : letI

theorem **isScal_tr** (h : R.IsScal F) : letI

theorem **primRows3_tr** (h : PrimRows3 R F) : letI

theorem **realAdelicInertTransport_furtherRows** (h : UnusedChoiceStatements R F) : letI

Development.AbelianVarietyTheory.Definitions.AutomorphicMembership.XiAtPenultimateEqXiCpl1

theorem **castSp_apply** {E : Type} [Field E] {a b : ℕ} (h : a = b) (x : Sp E a) (i : Fin b) : castSp E h x i = x (Fin.cast h.symm i)

theorem **castSp_comp** {E : Type} [Field E] {a b : ℕ} (x : HomE E a b) : castSp E rfl ∘ₗ x = x

theorem **thC_zero** {E : Type} [Field E] [NumberField E] [StarRing E] {ι : Type} [Fintype ι] {a b : ℕ} (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (AA : Submodule (𝒪 E) E) (SS : Finset (HeightOneSpectrum (𝒪 E))) (N : Submodule (𝒪 E) (Sp E a)) (N' : Submodule (𝒪 E) (Sp E b)) (c : ℕ) (θ : VCx ι a b → ℂ) : thC emb δ d sc AA SS N N' c (fun _ => 0) θ = 0

theorem **PhiQ_comp_of_red** {E K : Type} [Field E] [NumberField E] [StarRing E] [Field K] {P : HeightOneSpectrum (𝒪 E)} (hq : ((qOf P : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) (ξ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (k₀ r c' n' n : ℕ) (hc : c' = 0) (κ : GLint E n P) (hκ : TopStep.redGL n P κ = 1) (x : HomE E (n' + c') (n + 0)) : LocE.PhiQ (qOf P) ξ k₀ r c' n' 0 n (κ.1.toLinearMap ∘ₗ x) = LocE.PhiQ (qOf P) ξ k₀ r c' n' 0 n x

theorem **Idx.some_get** {j : ℕ} (b : Idx sel mu L j) (h : j < R.m) : some (b.get h) = b.1

theorem **Idx.get_eq** {j : ℕ} {b : Idx sel mu L j} {al : L.β j} (e : b.1 = some al) (h : j < R.m) : b.get h = al

theorem **Idx.of_get** {j : ℕ} (b : Idx sel mu L j) (h h' : j < R.m) : Idx.of (b.get h) h' = b

theorem **Idx.eq_top** {j : ℕ} (b : Idx sel mu L j) (h : R.m ≤ j) : b = Idx.top sel mu L h

theorem **ysAt_ys_of_eq** {j : ℕ} (h : j + 1 = R.m) (N : Submodule (𝒪 X.E) (Sp X.E (R.N j))) : (ysAt R sel mu cst j).ys N = ybSet sel mu cst N

theorem **ysAt_ys_of_ne** {j : ℕ} (h : j + 1 ≠ R.m) (N : Submodule (𝒪 X.E) (Sp X.E (R.N j))) : (ysAt R sel mu cst j).ys N = {∅}

theorem **PtAt.mem_ybSet** {u : Pr.Word} {j : ℕ} (p : PtAt R sel mu cst u j) (h : j + 1 = R.m) : p.1.2 ∈ ybSet sel mu cst p.1.1.deco.1.N

theorem **PtAt.repY_mem** {u : Pr.Word} {j : ℕ} (p : PtAt R sel mu cst u j) (h : j + 1 = R.m) : repY p.1.2 ∈ p.1.2

theorem **exists_ptAt** {u : Pr.Word} {j : ℕ} (q : ChPt (R

theorem **eq_tilAt_iff** {u : Pr.Word} {j : ℕ} (h : j = R.m) (p' : ChPt (R

theorem **XiTop_tilF** {u : Pr.Word} (hX : X.Ok) (hPr : Pr.Ok) (hsel : IsSel X sel) (hc : cst.Ok) (H : TopStep.TilOk Pr sel mu u) {n : ℕ} (hm : R.m = n + 2) (p : ChPt (R

theorem **xiOver_rfl** (Y : Set (Sp X.E (R.N (n + 1)))) (p' : ChPt (R

theorem **xiAt_entry** (h : i + 2 < R.m) {al : L.β i} {be : L.β (i + 1)} (e : b.1 = some al) (e' : b'.1 = some be) (p : PtAt R sel mu cst u i) (p' : PtAt R sel mu cst u (i + 1)) : xiAt Ch sel mu cst L u i b b' p p' = XiEntry Ch sel mu cst L i al be p.1.1 p'.1.1

theorem **xiAt_penultimate_eq_xiCpl1** (h : i + 2 = R.m) {al : L.β i} (e : b.1 = some al) (p : PtAt R sel mu cst u i) (p' : PtAt R sel mu cst u (i + 1)) : xiAt Ch sel mu cst L u i b b' p p' = XiCpl1 Ch sel mu cst L i al p.1.1 p'.1.1 p'.1.2

theorem **xiAt_of_le** (h : R.m ≤ i) (p : PtAt R sel mu cst u i) (p' : PtAt R sel mu cst u (i + 1)) : xiAt Ch sel mu cst L u i b b' p p' = 0

theorem **xiAt_last** : xiAt Ch sel mu cst L u (n + 1) b b' p p' = xiOver Ch sel mu cst n h p.1.1 p.1.2 p'.1.1

theorem **phiT_of_xiAt_ne_zero** {u : Pr.Word} {i : ℕ} {b : Idx sel mu L i} {b' : Idx sel mu L (i + 1)} {p : PtAt R sel mu cst u i} {p' : PtAt R sel mu cst u (i + 1)} {x : HomE X.E (R.N i) (R.N (i + 1))} (hx : xiAt Ch sel mu cst L u i b b' p p' x ≠ 0) (P : Pr.vT u) : ∀ y ∈ p.1.1.deco.1.at P.1, x y ∈ p'.1.1.deco.1.at P.1 ∧ p'.1.1.deco.1.f P (x y) = p.1.1.deco.1.f P y

theorem **thOf_zero** (u : Pr.Word) {a b : ℕ} (N : Submodule (𝒪 X.E) (Sp X.E a)) (N' : Submodule (𝒪 X.E) (Sp X.E b)) : thOf sel mu cst A u N N' 0 = 0

theorem **thAt_of_le** (h : R.m ≤ i) (p : PtAt R sel mu cst u i) (p' : PtAt R sel mu cst u (i + 1)) : thAt Ch sel mu cst L A u i b b' p p' = 0

theorem **matK_eq_xiAt** {i : ℕ} (h : i + 2 < R.m) (al : L.β i) (be : L.β (i + 1)) (p : PtAt R sel mu cst u i) (p' : PtAt R sel mu cst u (i + 1)) : matK W Ch sel mu cst L T i al be p.1.1 p'.1.1 = kerOf (T.th i) (Sall sel mu u) (cAof sel mu cst) (cBof sel mu cst u) p.1.1.deco.1.N (frsF sel mu p.1.1) p'.1.1.deco.1.N (frsF sel mu p'.1.1) (xiAt Ch sel mu cst L u i (Idx.of al (by omega)) (Idx.of be (by omega)) p p')

theorem **firstCoupledKernel_eq_columnChainDatum** (h : n + 2 = R.m) (al : L.β n) (b' : Idx sel mu L (n + 1)) (p : PtAt R sel mu cst u n) (p' : PtAt R sel mu cst u (n + 1)) : cp1K W Ch sel mu cst L T al p.1.1 p'.1.1 p'.1.2 = kerOf (T.th n) (Sall sel mu u) (cAof sel mu cst) (cBof sel mu cst u) p.1.1.deco.1.N (frsF sel mu p.1.1) p'.1.1.deco.1.N (frsF sel mu p'.1.1) (xiAt Ch sel mu cst L u n (Idx.of al (by omega)) b' p p')

theorem **k2top_eq_xiOver** (h : n + 1 + 1 = R.m) (p : PtAt R sel mu cst u (n + 1)) (q : ChPt (R

Development.AbelianVarietyTheory.Definitions.CMLattices.CmHostBarHalfIntegerGenerator

theorem **existsUnique_isKForm** (D : Datum H) (ε : ℤˣ) : ∃! h, IsKForm D ε h

theorem **cm_generator_sq_quadratic_eq** (H : Host E) : H.w0 ∗ H.w0 = H.w0 − (((H.d + 1) / 4 : ℕ) : 𝒪 E)

theorem **bar_rt** (H : Host E) : H.bar H.rt = −H.rt

theorem **cmHost_bar_halfIntegerGenerator** (H : Host E) : H.bar H.w0 = 1 − H.w0

theorem **isHermitianForm_linear_left** (hh : IsKForm D ε h) (x y : D.La) : h (H.w0 • x) y = wK H ∗ h x y

theorem **ef_swap** (D : Datum H) (x y : D.La) : D.EF y x = −D.EF x y

theorem **isKForm_herm** (hh : IsKForm D ε h) (x y : D.La) : (h y x : 𝒪 E) = H.bar (h x y)

theorem **isKForm_self** (hh : IsKForm D ε h) (x : D.La) : (h x x : 𝒪 E) = (((ε : ℤ) ∗ D.EF (H.w0 • x) x : ℤ) : 𝒪 E)

theorem **isKForm_adj** (hh : IsKForm D ε h) (s : 𝒪 E) (x y : D.La) : h (s • x) y = h x (H.bar s • y)

theorem **nonempty_genFib** (hB : Out.PolBaseChangeSt) (P : Out.PolAV (Sp (𝒪 K))) : Nonempty (GenFib P)

theorem **existsUnique_isCoord** {P : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) {u : epsR (𝒪 K) ⟶ P.A} (hu : IsTangent u) : ∃! c, F.IsCoord u c

theorem **eigLat_le** {P : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) {O : Type} [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K) : F.eigLat act τ ≤ F.lieLat

theorem **isDiffOf_id** {P : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) : F.IsDiffOf F (𝟙 _) LinearMap.id

theorem **exists_genFib_dim_one** (hB : Out.PolBaseChangeSt) (hC : CMCurve) : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (P : Out.PolAV (Sp (𝒪 M))) (F : GenFib P), F.P₀.g = 1

theorem **exists_root_of_splits** (H : Host E) (h : Splits E K) : ∃ ρ : 𝒪 K, ρ ^ 2 = −(H.d : 𝒪 K)

theorem **hasType_neg** {ρ : 𝒪 K} (hρ : ρ ^ 2 = −(H.d : 𝒪 K)) {G : Over (Sp (𝒪 K))} [GrpObj G] {act : OAct (𝒪 E) G} {p q : ℕ} (h : HasType H act ρ p q) : HasType H act (−ρ) q p

Development.AbelianVarietyTheory.Definitions.CMLattices.ExistsIsThetaDegOfIsLocalModel

theorem **homExtDvr_iff** : HomExtDvr ↔ ∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f

theorem **logOm_of_finrank_eq_two** {d : ℕ} [NeZero d] (h2 : Module.finrank ℚ K = 2) (hD : discr K = −d) (χ : DirichletCharacter ℂ d) : logOm K χ = (Real.eulerMascheroniConstant + Real.log (2 ∗ Real.pi / √d) : ℝ) − deriv χ.LFunction 1 / χ.LFunction 1

theorem **Ell.HasPeriod.unique** {ω : 𝒪 K} {C : Ell H (𝒪 K) ω} {x y : ℝ} (hx : C.HasPeriod x) (hy : C.HasPeriod y) : x = y

theorem **exists_isThetaDeg_of_isLocalModel** (hU : ThetaUnifAV) (x : KeyIn K) (h : ∀ ᵖ, ∃ M, KeyPow.IsLocalModel x ᵖ 1 M) : ∃ nth : (K →+* ℂ) → ℝ, IsThetaDeg x ((1 / 2) ∗ (∑ σ, Real.log (nth σ)) / Module.finrank ℚ K)

theorem **finrank_D₁** (Z : CMLat hP p q) (b : Bool) : Module.finrank (𝒪 E) (Z.D₁ b).La = 1

theorem **exists_smul_eq_smul** {D₁ : Datum H} (h : Module.finrank (𝒪 E) D₁.La = 1) {a : D₁.La} (ha : a ≠ 0) (b : D₁.La) : ∃ r s : 𝒪 E, r ≠ 0 ∧ r • b = s • a

theorem **f_injective** (Z : CMLat hP p q) (hpq : p + q = D.p + D.q) (j : Fin p ⊕ Fin q) : Function.Injective (Z.f j)

theorem **hh_ne_zero** (Z : CMLat hP p q) (hpq : p + q = D.p + D.q) (j : Fin p ⊕ Fin q) : Z.hh j ≠ 0

theorem **exists_of_rank_one** (D : Datum H) (h : D.p + D.q = 1) : ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP D.p D.q), IsTypeFn H D.p D.q ty ∧ Nontrivial D.La ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

theorem **exists_of_selfDual** (𝔞 : Submodule (𝒪 E) E) (hfg : 𝔞.FG) (hne : 𝔞 ≠ ⊥) (hd : ∀ y : E, y ∈ 𝔞 ↔ ∀ x ∈ 𝔞, ∃ n : ℤ, Algebra.trace ℚ E (x * IsCMField.complexConj E y / (H.rt : E)) = n) : ∃ (D : Datum H) (_ : D.La ≃ₗ[𝒪 E] 𝔞) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP D.p D.q), D.p = 1 ∧ D.q = 0 ∧ IsTypeFn H D.p D.q ty ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

theorem **exists_cyclotomic** (p q : ℕ) (h : p + q = 1) : ∃ (D : Datum CyclotomicTwentyEight.host) (g : ℕ) (ty : Fin g → (CyclotomicTwentyEight.CyclotomicFieldTwentyEight →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP D.p D.q), D.p = p ∧ D.q = q ∧ IsTypeFn CyclotomicTwentyEight.host D.p D.q ty ∧ Nontrivial D.La ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

Development.AbelianVarietyTheory.Definitions.CMLattices.IntegralOf

theorem **integral_of** (T : CDVR) (io : c.F →+* T.Fr) (y : letI

Development.AbelianVarietyTheory.Definitions.CMLattices.IsCyclotomicExtensionCyclotomicField28

theorem **baseChange_neg_form** (D : Datum H) (x y : ℝ ⊗[ℤ] D.La) : (-D.EF).baseChange ℝ x y = -D.EF.baseChange ℝ x y

@[simp] theorem **oppDatum_p** (D : Datum H) : (oppDatum D).p = D.q

@[simp] theorem **oppDatum_q** (D : Datum H) : (oppDatum D).q = D.p

theorem **exists_type_period** (D : Datum H) : ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)), IsTypeFn H D.p D.q ty ∧ IsPeriod D ty Pe

theorem **zeta_prim** : IsPrimitiveRoot (IsCyclotomicExtension.zeta 28 ℚ CyclotomicFieldTwentyEight) 28

theorem **z_prim** : IsPrimitiveRoot z 28

theorem **z_pow_14** : z ^ 14 = -1

theorem **z_root** : z ^ 12 - z ^ 10 + z ^ 8 - z ^ 6 + z ^ 4 - z ^ 2 + 1 = 0

theorem **half_sq** : half ^ 2 - half + 2 = 0

theorem **rt_sq** : (2 * half - 1) ^ 2 = -((7 : ℕ) : 𝒪 CyclotomicFieldTwentyEight)

theorem **unram** : Algebra.Unramified (𝒪 (maximalRealSubfield CyclotomicFieldTwentyEight)) (𝒪 CyclotomicFieldTwentyEight)

theorem **conj_cyclotomicTwentyEight_eq_neg_pow_thirteen** : IsCMField.ringOfIntegersComplexConj CyclotomicFieldTwentyEight z = -z ^ 13

theorem **aeval_derivative** : aeval (z : CyclotomicFieldTwentyEight) (derivative (minpoly ℚ (z : CyclotomicFieldTwentyEight))) = (dd : CyclotomicFieldTwentyEight)

theorem **trace_dual** (w : CyclotomicFieldTwentyEight) : (∀ a : 𝒪 CyclotomicFieldTwentyEight, ∃ n : ℤ, Algebra.trace ℚ CyclotomicFieldTwentyEight ((a : CyclotomicFieldTwentyEight) * w) = n) ↔ ∃ b : 𝒪 CyclotomicFieldTwentyEight, w = (dd : CyclotomicFieldTwentyEight)⁻¹ * b

theorem **conj_bt** : IsCMField.ringOfIntegersComplexConj CyclotomicFieldTwentyEight bt = (1 - z ^ 7) * (1 + z ^ 10)

theorem **norm_bt** : bt * IsCMField.ringOfIntegersComplexConj CyclotomicFieldTwentyEight bt * host.rt = (-z + z ^ 3 - z ^ 9) * dd

theorem **unit_eq** : (-z + z ^ 3 - z ^ 9) * (z ^ 3 + z ^ 7) = 1

theorem **exists_datum** : ∃ D : Datum host, D.p = 1 ∧ D.q = 0

Development.AbelianVarietyTheory.Definitions.CMLattices.IsEmptyCmLatZero

theorem **baseChange_neg_apply** (D : Datum H) (x y : ℝ ⊗[ℤ] D.La) : (-D.EF).baseChange ℝ x y = -D.EF.baseChange ℝ x y

theorem **zeta_prim** : IsPrimitiveRoot (IsCyclotomicExtension.zeta 28 ℚ CyclotomicFieldTwentyEight) 28

theorem **z_prim** : IsPrimitiveRoot z 28

theorem **z_pow_14** : z ^ 14 = -1

theorem **z_root** : z ^ 12 - z ^ 10 + z ^ 8 - z ^ 6 + z ^ 4 - z ^ 2 + 1 = 0

theorem **half_sq** : half ^ 2 - half + 2 = 0

theorem **rt_sq** : (2 * half - 1) ^ 2 = -((7 : ℕ) : 𝒪 CyclotomicFieldTwentyEight)

theorem **unram** : Algebra.Unramified (𝒪 (maximalRealSubfield CyclotomicFieldTwentyEight)) (𝒪 CyclotomicFieldTwentyEight)

theorem **conj_cyclotomicTwentyEight_eq_neg_pow_thirteen** : IsCMField.ringOfIntegersComplexConj CyclotomicFieldTwentyEight z = -z ^ 13

theorem **aeval_derivative** : aeval (z : CyclotomicFieldTwentyEight) (derivative (minpoly ℚ (z : CyclotomicFieldTwentyEight))) = (dd : CyclotomicFieldTwentyEight)

theorem **trace_dual** (w : CyclotomicFieldTwentyEight) : (∀ a : 𝒪 CyclotomicFieldTwentyEight, ∃ n : ℤ, Algebra.trace ℚ CyclotomicFieldTwentyEight ((a : CyclotomicFieldTwentyEight) * w) = n) ↔ ∃ b : 𝒪 CyclotomicFieldTwentyEight, w = (dd : CyclotomicFieldTwentyEight)⁻¹ * b

theorem **conj_bt** : IsCMField.ringOfIntegersComplexConj CyclotomicFieldTwentyEight bt = (1 - z ^ 7) * (1 + z ^ 10)

theorem **norm_bt** : bt * IsCMField.ringOfIntegersComplexConj CyclotomicFieldTwentyEight bt * host.rt = (-z + z ^ 3 - z ^ 9) * dd

theorem **unit_eq** : (-z + z ^ 3 - z ^ 9) * (z ^ 3 + z ^ 7) = 1

theorem **exists_datum** : ∃ D : Datum host, D.p = 1 ∧ D.q = 0

theorem **exists_cmLat** : ∃ (D : Datum host) (g : ℕ) (ty : Fin g → (CyclotomicFieldTwentyEight →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP 1 0), D.p = 1 ∧ D.q = 0 ∧ Nontrivial D.La ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

theorem **exists_cmLat_neg** : ∃ (D : Datum host) (g : ℕ) (ty : Fin g → (CyclotomicFieldTwentyEight →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP 0 1), D.p = 0 ∧ D.q = 1 ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

theorem **logOm_quadratic** {K : Type} [Field K] [NumberField K] {d : ℕ} [NeZero d] (h2 : Module.finrank ℚ K = 2) (hD : discr K = -d) (χ : DirichletCharacter ℂ d) : logOm K χ = (Real.eulerMascheroniConstant + Real.log (2 * Real.pi / √d) : ℝ) - deriv χ.LFunction 1 / χ.LFunction 1

theorem **hasPeriod_unique** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {M : Type} [Field M] [NumberField M] {ω : 𝒪 M} {C : Ell H (𝒪 M) ω} {x y : ℝ} (hx : C.HasPeriod x) (hy : C.HasPeriod y) : x = y

theorem **bar_hh** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} {hP : IsPeriod D ty Pe} {p q : ℕ} (Z : CMLat hP p q) (j : Fin p ⊕ Fin q) : H.bar (Z.hh j) = Z.hh j

theorem **im_mul_form_pos** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) {ρ : ℂ} (hρ : ρ.im ≠ 0) {a : D.La} (ha : a ∈ PureType.lat hP ρ) (h0 : a ≠ 0) : 0 < ρ.im * D.EF (H.w0 • a) a

theorem **hh_sign** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} {hP : IsPeriod D ty Pe} {p q : ℕ} (Z : CMLat hP p q) (hpq : p + q = D.p + D.q) (j : Fin p ⊕ Fin q) {a : (Z.D₁ j.isLeft).La} (ha : a ≠ 0) : 0 < (Z.D₁ j.isLeft).EF (Z.hh j • H.w0 • a) a * (Z.D₁ j.isLeft).EF (H.w0 • a) a

theorem **exists_sum** (D₁ D₂ : Datum H) {g₁ g₂ : ℕ} {t₁ : Fin g₁ → (E →+* ℂ)} {t₂ : Fin g₂ → (E →+* ℂ)} {P₁ : D₁.La →+ (Fin g₁ → ℂ)} {P₂ : D₂.La →+ (Fin g₂ → ℂ)} (h₁ : IsPeriod D₁ t₁ P₁) (h₂ : IsPeriod D₂ t₂ P₂) (y₁ : IsTypeFn H D₁.p D₁.q t₁) (y₂ : IsTypeFn H D₂.p D₂.q t₂) : ∃ (D : Datum H) (e : D.La ≃ₗ[𝒪 E] D₁.La × D₂.La) (P : D.La →+ (Fin (g₁ + g₂) → ℂ)), D.p = D₁.p + D₂.p ∧ D.q = D₁.q + D₂.q ∧ (∀ x y, D.EF x y = D₁.EF (e x).1 (e y).1 + D₂.EF (e x).2 (e y).2) ∧ IsTypeFn H D.p D.q (Fin.append t₁ t₂) ∧ IsPeriod D (Fin.append t₁ t₂) P ∧ ∀ x, P x = Fin.append (P₁ (e x).1) (P₂ (e x).2)

theorem **exists_cmLat_mixed** (D₀ : Datum H) (hp : D₀.p = 1) (hq : D₀.q = 0) : ∃ (D : Datum H) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP 1 1), D.p = 1 ∧ D.q = 1 ∧ IsTypeFn H 1 1 ty ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

theorem **exists_cmLat_mixed_cyclotomic** : ∃ (D : Datum CyclotomicTwentyEight.host) (g : ℕ) (ty : Fin g → (CyclotomicTwentyEight.CyclotomicFieldTwentyEight →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (Z : CMLat hP 1 1), D.p = 1 ∧ D.q = 1 ∧ IsTypeFn CyclotomicTwentyEight.host 1 1 ty ∧ ∀ j, Function.Injective (Z.f j) ∧ Z.hh j = 1

theorem **isFam_of_isTen** {R : Type} [CommRing R] {ω : R} {C : Ell H R ω} {D : Datum H} {h : D.La → D.La → OK H} {T : PAT H (AlgebraicGeometry.Spec (CommRingCat.of R))} (t : IsTen C D h T) : IsFam C T h ⊤ fun a => t.ten.tmul a.1

theorem **isEmpty_cmLat_zero** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) [Nontrivial D.La] : IsEmpty (CMLat hP 0 0)

Development.AbelianVarietyTheory.Definitions.CommutativeAlgebra.AbByPOfComm

theorem **isChar_one** (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] : IsChar K L (fun _ : G => (1 : L))

Development.AbelianVarietyTheory.Definitions.CommutativeAlgebra.CompletionDVR

theorem **completionDVR** (W : Type*) [CommRing W] [IsDomain W] [IsDiscreteValuationRing W] : (∃ _ : IsDomain (AdicCompletion (maximalIdeal W) W), IsDiscreteValuationRing (AdicCompletion (maximalIdeal W) W)) ∧ IsAdicComplete (maximalIdeal (AdicCompletion (maximalIdeal W) W)) (AdicCompletion (maximalIdeal W) W) ∧ Function.Injective (algebraMap W (AdicCompletion (maximalIdeal W) W)) ∧ IsLocalHom (algebraMap W (AdicCompletion (maximalIdeal W) W)) ∧ maximalIdeal (AdicCompletion (maximalIdeal W) W) = (maximalIdeal W).map (algebraMap W (AdicCompletion (maximalIdeal W) W))

Development.AbelianVarietyTheory.Definitions.CommutativeAlgebra.InvariantSubspaceMobiusInverseRecEqOfFinrankLt

theorem **invariantSubspaceMobiusInverseRec_succ** (s : F[X] → ℂ) (n : ℕ) (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) : invariantSubspaceMobiusInverseRec s (n + 1) V g = s (LinearMap.charpoly g) − ∑ᶠ W : {W : Submodule F V // W ≠ ⊤ ∧ ∀ x ∈ W, g x ∈ W}, invariantSubspaceMobiusInverseRec s n W.1 (g.restrict W.2.2)

theorem **invariantSubspaceMobiusInverseRec_eq_of_finrank_lt** (s : F[X] → ℂ) (n : ℕ) : ∀ (m : ℕ) (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V), Module.finrank F V < n → Module.finrank F V < m → invariantSubspaceMobiusInverseRec s n V g = invariantSubspaceMobiusInverseRec s m V g

theorem **psi_eq** (s : F[X] → ℂ) (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) : psi s V g = s (LinearMap.charpoly g) − ∑ᶠ W : {W : Submodule F V // W ≠ ⊤ ∧ ∀ x ∈ W, g x ∈ W}, psi s W.1 (g.restrict W.2.2)

theorem **isAdditive_natDegree** : IsAdditive (fun P : F[X] => (P.natDegree : ℂ))

Development.AbelianVarietyTheory.Definitions.Degenerations.AdmissibleBaseOfCompleteDvr

theorem **spMap_comp'** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap (g.comp f) = spMap g ≫ spMap f

theorem **ArcPt.comp** {Y : Model Xo c β} {R₁ R₂ : Type} [CommRing R₁] [IsDomain R₁] [CommRing R₂] [IsDomain R₂] {g : PowerSeries O' →+* R₁} {y₁ : Sp (FractionRing R₁) ⟶ Xo.XF c} (hy : ArcPt Y b g y₁) (h : R₁ →+* R₂) (h' : FractionRing R₁ →+* FractionRing R₂) (hf : h.Finite) (hh : ∀ r, h' (algebraMap R₁ _ r) = algebraMap R₂ _ (h r)) : ArcPt Y b (h.comp g) (spMap h' ≫ y₁)

theorem **RealAt.comp** {Y : Model Xo c β} {a : ArcIn D l β O' b} {R₁ R₂ : Type} [CommRing R₁] [IsDomain R₁] [CommRing R₂] [IsDomain R₂] {g : PowerSeries O' →+* R₁} {y₁ : Sp (FractionRing R₁) ⟶ Xo.XF c} (hR : RealAt Y a g y₁) (h : R₁ →+* R₂) (h' : FractionRing R₁ →+* FractionRing R₂) (hf : h.Finite) (hh : ∀ r, h' (algebraMap R₁ _ r) = algebraMap R₂ _ (h r)) : RealAt Y a (h.comp g) (spMap h' ≫ y₁)

theorem **serBase_algebraMap** (K : Type) [Field K] [IsAlgClosed K] [Algebra (FractionRing O') K] [Algebra (PowerSeries O') K] [FaithfulSMul (PowerSeries O') K] [Algebra.IsAlgebraic (PowerSeries O') K] (h : ∀ x, algebraMap (FractionRing O') K (algebraMap O' _ x) = algebraMap (PowerSeries O') K (PowerSeries.C x)) : SerBase (algebraMap (PowerSeries O') K)

theorem **Mdl.admissible_base_of_complete_dvr** {C : Type} [NormedField C] {o : Full D C} (M : Mdl o) : AdmBase M.W (IsLocalRing.maximalIdeal M.W) ∧ CharZero M.W

Development.AbelianVarietyTheory.Definitions.Degenerations.AdmissibleBasePowerSeriesSpanX

theorem **ne_top_of_admissibleBase** {R : Type} [CommRing R] {I : Ideal R} (h : AdmBase R I) : I ≠ ⊤

theorem **admissibleBase_powerSeries_span_X** : AdmBase (PowerSeries ℚ) (Ideal.span {PowerSeries.X})

theorem **zeroU_LG** [IsHausdorff I R] : (zeroU H R I).LG = (zeroTriple H R I).PP

theorem **exists_hyp** (h : AdmBase R I) : ∃ (t : SAB H (Sp R)) (D : DegData R I) (A : PolAV (Sp (FractionRing R))) (U : Unif I D A t.G), U.LG = t.PP

theorem **apply_zero** (h : PelSt) : ∃ (t : SAB GlobalTheory.HostModel.host (Sp (PowerSeries ℚ))) (P : PELUnif (Ideal.span {PowerSeries.X}) t), P.TorsionIso

Development.AbelianVarietyTheory.Definitions.Degenerations.AdmissiblePadicIntMaximalIdeal

theorem **cDvr_iff** (V : Type) [CommRing V] [IsDomain V] [IsLocalRing V] : CDvr V ↔ IsDiscreteValuationRing V ∧ IsAdicComplete (IsLocalRing.maximalIdeal V) V ∧ CharZero V

theorem **snd_hom** (T : Over Z) : (snd T G).left ≫ G.hom = (fst T G).left ≫ T.hom

theorem **mem_chart** {T : Over Z} {i : A.ι} (hT : ∀ t, T.hom t ∈ A.U i) {Y : Scheme.{0}} (r : Y ⟶ (T ⊗ G).left) (y : Y) : (r ≫ (snd T G).left ≫ G.hom) y ∈ A.U i

theorem **exists_chartFrame** (hA : A.IsLB) (T : Over Z) (i : A.ι) (hT : ∀ t, T.hom t ∈ A.U i) : ∃ e, IsChartFrame G A N T i e

theorem **whisker_snd** {T T' : Over Z} (h : T' ⟶ T) : (h ▷ G).left ≫ (snd T G).left = 𝟙 _ ≫ (snd T' G).left

theorem **IsChartFrame.pull** {T T' : Over Z} (h : T' ⟶ T) {i : A.ι} {e} (he : IsChartFrame G A N T i e) (y : (T' ⊗ G).left) (J) (hJ : Good (A.pull G.hom) y J) : sval (L

theorem **lin_of** (hA : A.IsLB) {T T' : Over Z} (h : T' ⟶ T) {i j : A.ι} (hi : ∀ t, T'.hom t ∈ A.U i) (hj : ∀ t, T'.hom t ∈ A.U j) {e e'} (he : IsChartFrame G A N T j e) (he' : IsChartFrame G A N T' i e') (b : Fin 1 → _) (hb : b 0 = e) (c : Fin 1 → Γ(T'.left, ⊤)) (hc : ∀ y, gs (fst T' G).left y ⊤ (c 0) = (gm (𝟙 _ ≫ (snd T' G).left ≫ G.hom) y _ (A.g j i) ^ N).val) : Lin G (A.pull G.hom) N h b c e'

theorem **basisAlong_of** (hA : A.IsLB) {T T' : Over Z} (h : T' ⟶ T) {i : A.ι} (hi : ∀ t, T'.hom t ∈ A.U i) (hb : Function.Bijective (fst T' G).left.appTop.hom) {e} (he : IsChartFrame G A N T i e) : BasisAlong G (A.pull G.hom) N h (fun _ : Fin 1 => e)

theorem **gs_unit_overlap** (i j : A.ι) (a : Γ(Z, A.U i ⊓ A.U j)ˣ) (y : ((Over.mk (A.U i ⊓ A.U j).ι) ⊗ G).left) : gs (fst (Over.mk (A.U i ⊓ A.U j).ι) G).left y ⊤ ((A.U i ⊓ A.U j).topIso.inv.hom a.val) = (gm (𝟙 _ ≫ (snd (Over.mk (A.U i ⊓ A.U j).ι) G).left ≫ G.hom) y _ a).val

theorem **bijective_fst** [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] {V : Type} [CommRing V] [IsNoetherianRing V] (χ : Sp V ⟶ Z) : Function.Bijective (fst (Over.mk χ) G).left.appTop.hom

theorem **exists_ampFrames** [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] (hA : IsAmpleLB A) : ∃ F : AmpFrames G (A.pull G.hom) N, F.A = A ∧ F.n = 1 ∧ F.r = -1 - N ∧ F.s = 1

theorem **abLocus_eq_univ** [IsProper G.hom] : abLocus G = Set.univ

theorem **cDvr_padicInt** (p : ℕ) [Fact p.Prime] : CDvr ℤ_[p]

theorem **nonempty_point_chart** (R : Type) [CommRing R] (d : ℕ) (i : Fin (d + 1)) : Nonempty (Sp R ⟶ (ProjSpace.chart R d i).toScheme)

theorem **twistingSheaf_transition_ne_one** (R : Type) [CommRing R] [IsDomain R] : (ProjSpace.O1 R 1).g (0 : Fin 2) (1 : Fin 2) ≠ 1

theorem **ampFrames_constFamily** {R : Type} [CommRing R] (C : _root_.WeierstrassCurve R) : (∃ F : AmpFrames (constFamily C) ((ProjSpace.O1 R 1).pull (constFamily C).hom) N, F.A = ProjSpace.O1 R 1 ∧ F.n = 1 ∧ F.r = -1 - N ∧ F.s = 1) ∧ abLocus (constFamily C) = Set.univ

theorem **exists_ampFrames_nontrivial** (p : ℕ) [Fact p.Prime] (C : _root_.WeierstrassCurve ℤ_[p]) : ∃ F : AmpFrames (constFamily C) ((ProjSpace.O1 ℤ_[p] 1).pull (constFamily C).hom) N, F.n = 1 ∧ F.r = -1 - N ∧ F.s = 1 ∧ (∃ i j, F.A.g i j ≠ 1) ∧ CDvr ℤ_[p] ∧ ∀ i, ∃ χ : Sp ℤ_[p] ⟶ (F.A.U i).toScheme, Set.range (jK ℤ_[p] ≫ χ ≫ (F.A.U i).ι) ⊆ abLocus (constFamily C)

theorem **exists_isFrame** [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] (hA : A.IsLB) (i : A.ι) (hi : ∀ z, z ∈ A.U i) {R₁ : Type} [CommRing R₁] [IsDomain R₁] (x : Sp (FractionRing R₁) ⟶ Z) : ∃ e, IsChartFrame G A N (Over.mk x) i e ∧ IsFrame G (A.pull G.hom) N x (fun _ : Fin 1 => e)

theorem **not_basisAlong_of_isEmpty** (L : LB G.left) [IsEmpty L.ι] {T T' : Over Z} (h : T' ⟶ T) [Nontrivial Γ(T'.left, ⊤)] (b : Fin 1 → LB.Sec (LPow G L (N : ℤ) T)) : ¬ BasisAlong G L N h b

theorem **admissible_padicInt_maximalIdeal** (p : ℕ) [Fact p.Prime] : admitted ℤ_[p] (IsLocalRing.maximalIdeal ℤ_[p])

theorem **arcBase_examples** (p : ℕ) [Fact p.Prime] : ArcBase ℚ 1 ∧ ArcBase ℤ_[p] (p : ℤ_[p])

Development.AbelianVarietyTheory.Definitions.Degenerations.ArcBaseExamples

theorem **snd_hom** (T : Over Z) : (snd T G).left ≫ G.hom = (fst T G).left ≫ T.hom

theorem **mem_chart** {T : Over Z} {i : A.ι} (hT : ∀ t, T.hom t ∈ A.U i) {Y : Scheme.{0}} (r : Y ⟶ (T ⊗ G).left) (y : Y) : (r ≫ (snd T G).left ≫ G.hom) y ∈ A.U i

theorem **exists_chartFrame** (hA : A.IsLB) (T : Over Z) (i : A.ι) (hT : ∀ t, T.hom t ∈ A.U i) : ∃ e, IsChartFrame G A N T i e

theorem **whisker_snd** {T T' : Over Z} (h : T' ⟶ T) : (h ▷ G).left ≫ (snd T G).left = 𝟙 _ ≫ (snd T' G).left

theorem **IsChartFrame.pull** {T T' : Over Z} (h : T' ⟶ T) {i : A.ι} {e} (he : IsChartFrame G A N T i e) (y : (T' ⊗ G).left) (J) (hJ : Good (A.pull G.hom) y J) : sval (L

theorem **lin_of** (hA : A.IsLB) {T T' : Over Z} (h : T' ⟶ T) {i j : A.ι} (hi : ∀ t, T'.hom t ∈ A.U i) (hj : ∀ t, T'.hom t ∈ A.U j) {e e'} (he : IsChartFrame G A N T j e) (he' : IsChartFrame G A N T' i e') (b : Fin 1 → _) (hb : b 0 = e) (c : Fin 1 → Γ(T'.left, ⊤)) (hc : ∀ y, gs (fst T' G).left y ⊤ (c 0) = (gm (𝟙 _ ≫ (snd T' G).left ≫ G.hom) y _ (A.g j i) ^ N).val) : Lin G (A.pull G.hom) N h b c e'

theorem **basisAlong_of** (hA : A.IsLB) {T T' : Over Z} (h : T' ⟶ T) {i : A.ι} (hi : ∀ t, T'.hom t ∈ A.U i) (hb : Function.Bijective (fst T' G).left.appTop.hom) {e} (he : IsChartFrame G A N T i e) : BasisAlong G (A.pull G.hom) N h (fun _ : Fin 1 => e)

theorem **gs_unit_overlap** (i j : A.ι) (a : Γ(Z, A.U i ⊓ A.U j)ˣ) (y : ((Over.mk (A.U i ⊓ A.U j).ι) ⊗ G).left) : gs (fst (Over.mk (A.U i ⊓ A.U j).ι) G).left y ⊤ ((A.U i ⊓ A.U j).topIso.inv.hom a.val) = (gm (𝟙 _ ≫ (snd (Over.mk (A.U i ⊓ A.U j).ι) G).left ≫ G.hom) y _ a).val

theorem **exists_ampFrames** [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] (hA : IsAmpleLB A) : ∃ F : AmpFrames G (A.pull G.hom) N, F.A = A ∧ F.n = 1 ∧ F.r = -1 - N ∧ F.s = 1

theorem **abLocus_eq_univ** [IsProper G.hom] : abLocus G = Set.univ

theorem **cDvr_padicInt** (p : ℕ) [Fact p.Prime] : CDvr ℤ_[p]

theorem **ampFrames_constFamily** (C : _root_.WeierstrassCurve ℤ) : (∃ F : AmpFrames (constFamily C) ((ProjSpace.O1 ℤ 1).pull (constFamily C).hom) N, F.A = ProjSpace.O1 ℤ 1 ∧ F.n = 1) ∧ abLocus (constFamily C) = Set.univ

theorem **bijective_fst** [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] {V : Type} [CommRing V] [IsNoetherianRing V] (χ : Sp V ⟶ Z) : Function.Bijective (fst (Over.mk χ) G).left.appTop.hom

theorem **exists_isFrame** [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] (hA : A.IsLB) (i : A.ι) (hi : ∀ z, z ∈ A.U i) {R₁ : Type} [CommRing R₁] [IsDomain R₁] (x : Sp (FractionRing R₁) ⟶ Z) : ∃ e, IsChartFrame G A N (Over.mk x) i e ∧ IsFrame G (A.pull G.hom) N x (fun _ : Fin 1 => e)

theorem **not_basisAlong_of_isEmpty** (L : LB G.left) [IsEmpty L.ι] {T T' : Over Z} (h : T' ⟶ T) [Nontrivial Γ(T'.left, ⊤)] (b : Fin 1 → LB.Sec (LPow G L (N : ℤ) T)) : ¬ BasisAlong G L N h b

theorem **admissible_padicInt_maximalIdeal** (p : ℕ) [Fact p.Prime] : admitted ℤ_[p] (IsLocalRing.maximalIdeal ℤ_[p])

theorem **arcBase_examples** (p : ℕ) [Fact p.Prime] : ArcBase ℚ 1 ∧ ArcBase ℤ_[p] (p : ℤ_[p])

Development.AbelianVarietyTheory.Definitions.Degenerations.CharZeroOfIsPrimitiveRoot

theorem **splitAt_iff** {W : Type} [CommRing W] [IsLocalRing W] (t : SAB H (Sp W)) : SplitAt t ↔ ∃ S : SplitExt (Sp (IsLocalRing.ResidueField W)), S.IsGt ((Over.pullback (spMap (IsLocalRing.residue W))).obj t.G)

theorem **isPol_iff_polAt** {K : Type} [Field K] (t : SAB H (Sp K)) : IsPol t ↔ GlobalTheory.LN.PolAt t (𝟙 _)

theorem **NumOf.pt_unique** {R : Type} [CommRing R] [IsDomain R] {D : DegPre R} {F : Type} [Field F] [Algebra (FractionRing R) F] {N : NumD F} (ν : NumOf D F N) {g : GtPtF D F} {v v' : N.Gt} (h : ν.Pt g v) (h' : ν.Pt g v') : v = v'

theorem **ArcF.over_refl** {D : Datum H} {R : Type} [CommRing R] [IsDomain R] {F : Type} [Field F] [Algebra (FractionRing R) F] (A : ArcF D R F) : A.Over (RingHom.id F) A

theorem **Roots.pow** {C : Type} [Field C] {u : Cˣ} {r : ℕ → Cˣ} (h : Roots u r) {j : ℕ} (hj : 0 < j) : r j ^ j = u

theorem **existsUnique_twRel** {O : Type} [CommRing O] {N : NumD O} {k : ℕ} (β : N.Form k) (u r : Oˣ) {j : ℕ} (h : r ^ j = u) (p : N.QT j) : ∃! p' : (N.twist β u).QT j, TwRel β u r p p'

theorem **isCoordPt_one_iff** {S : Scheme.{0}} (P : PolAV S) (k : ℕ) {T : Over S} {I : Type} (x : I → (T ⟶ P.A)) (v : I → Γ(T.left, ⊤)) : IsCoordPt P k x 1 v ↔ Odd k ∧ IsUnit (k : Γ(T.left, ⊤)) ∧ IsThetaPtOver P k x v

theorem **compat_self** (H : Host E) (n : ℕ) : Compat H n (zeta n) H.rt0

theorem **charZero_of_isPrimitiveRoot** {K : Type} [Field K] {ζ : ℕ → Kˣ} (h : ∀ n, 0 < n → IsPrimitiveRoot (ζ n) n) : CharZero K

theorem **Full.charZero** {D : Datum H} {K : Type} [Field K] (o : Full D K) : CharZero K

theorem **ArcF.charZero** {D : Datum H} {R : Type} [CommRing R] [IsDomain R] {F : Type} [Field F] [Algebra (FractionRing R) F] (A : ArcF D R F) : CharZero F

theorem **isUnit_natCast_sp** {K : Type} [Field K] {m : ℕ} (hm : (m : K) ≠ 0) : IsUnit (m : Γ((baseOver (Sp K)).left, ⊤))

theorem **Full.exists_of_obj** {D : Datum H} {l : Geo.Tor.Level H} {B : Type} [CommRing B] {b : Base l B} {K : Type} [Field K] [IsAlgClosed K] (β : B →+* K) (o : Obj D l b (spMap β)) : ∃ f : Full D K, f.P = objPolAV o ∧ f.ρ = β b.ρ ∧ (f.I.ζ l.A : K) = β b.ζ

theorem **Full.exists_of_point** {D : Datum H} {l : Geo.Tor.Level H} {B : Type} [CommRing B] {b : Base l B} {S : Scheme.{0}} {s : S ⟶ Sp B} (o : Obj D l b s) {K : Type} [Field K] [IsAlgClosed K] (β : B →+* K) (x : Sp K ⟶ S) (hx : x ≫ s = spMap β) : ∃ f : Full D K, f.P.g = Module.finrank ℚ E / 2 * (D.p + D.q) ∧ f.ρ = β b.ρ ∧ (f.I.ζ l.A : K) = β b.ζ

theorem **vF_const** {F : Type} [Field F] {X Y : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {t : LB.Gen (LB.LinComb (Inh.trivLB X) φ ε)} {a : Γ(Y, ⊤)ˣ} (ht : IsConst t a) (r : Sp F ⟶ Y) (J : (Sp F ⟶ X) → (Inh.trivLB X).ι) (u : Fˣ) (hu : pullTop r a = unitOn u) : vF t r J = u

theorem **vF_one** {F : Type} [Field F] {X Y : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (r : Sp F ⟶ Y) (J : (Sp F ⟶ X) → (Inh.trivLB X).ι) : vF (cg φ ε 1) r J = 1

theorem **pt_toric** (t : Multiplicative X →* Fˣ) : (toricNumOf q he hb F).Pt (pt q F t) ((toricNum q he hb F).tor t)

theorem **toric_integral** (S : ValuationSubring F) (hS : ∀ r : K, algebraMap (FractionRing K) F (algebraMap K _ r) ∈ S) : (toricNumOf q he hb F).Integral S

theorem **hypArc_wp** (n : ℕ) (a b : HypLattice H) : (hypArc H K ρ hρ ζ hp hc).N.wp ((hypArc H K ρ hρ ζ hp hc).q n a) ((hypArc H K ρ hρ ζ hp hc).q n b) = ζ n ^ hypForm H a b

theorem **hypBe_bal** (k : ℕ) (hk : 0 < k) : (hypArc H K ρ hρ ζ hp hc).Bal (hypBe H K ρ hρ ζ hp hc k hk)

theorem **lbIso_of_g_one** {X : Scheme.{0}} {L M : LB X} (hL : ∀ i j, L.g i j = 1) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **zetaU_prim** (n : ℕ) (hn : 0 < n) : IsPrimitiveRoot (zetaU n) n

theorem **zetaU_pow** (n m : ℕ) (hn : 0 < n) (hm : 0 < m) : zetaU (n * m) ^ m = zetaU n

theorem **not_vanishC_X** (H : Host E) (m : ℕ) : ¬ VanishC (Test.zeroDatum H) m (MvPolynomial.X 0)

theorem **vanishC_zero** (D : Datum H) {m : ℕ} (hm : Odd m) : VanishC D m 0

theorem **exists_arcF** : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : Host E) (D : Datum H) (_ : Algebra (FractionRing ℤ) ℂ) (A : ArcF D ℤ ℂ) (β : A.N.Form 1), D.p = 1 ∧ D.q = 1 ∧ Nontrivial A.N.X ∧ A.Bal β ∧ A.ρ = H.rt0

theorem **realises_zero** {k : ℕ} (β : (zeroArc H K).N.Form k) (u : ℂˣ) (r : ℕ → ℂˣ) (hr : Roots u r) : Realises (zeroFull H H.rt0 H.sqrt_neg_disc_sq zetaU zetaU_prim zetaU_pow) (zeroArc H K) β u r

Development.AbelianVarietyTheory.Definitions.Degenerations.ExistsGenPullTriv

theorem **isGrpPullback_unit'** {S S' : Scheme.{0}} (f : S' ⟶ S) : IsGrpPullback (𝟙_ (Over S')) (𝟙_ (Over S)) f f

theorem **isCPPullback_unit** {S S' : Scheme.{0}} (f : S' ⟶ S) : IsCPPullback (CP.triv (𝟙_ (Over S'))) (CP.triv (𝟙_ (Over S))) f f

theorem **exists_gen_pull_triv** {Y Z : Scheme.{0}} (φ : Fin 2 → (Y ⟶ Y)) (ε : Fin 2 → ℤ) (r : Z ⟶ Y) (u : Γ(Z, ⊤)ˣ) : ∃ t : LB.Gen ((LB.LinComb (Inh.trivLB Y) φ ε).pull r), t.IsGen ∧ ∀ j, t.s j = res le_top u

theorem **torusAlg_repWt** (I : Ideal R) (X : Type) [AddCommGroup X] (χ : X) (e : ℕ) : (torusAlg I X).RepWt 1 χ e

Development.AbelianVarietyTheory.Definitions.Degenerations.ExistsLevelPoint

theorem **isDominant_gPt** (p : ℕ) [Fact p.Prime] : IsDominant (gPt p)

theorem **toG_over** (u : ρ.UTest p y₀) (a : u.PtC) : u.toG a ≫ u.G.hom = gPt p ≫ spMap u.j

theorem **exists_intGood** (u : ρ.UTest p y₀) : ∃ J, u.IntGood J ∧ ∀ z, Good u.D.toPre.P.L z J

theorem **gtN_unique** (u : ρ.UTest p y₀) (g : GtPtF u.D.toPre (PadicAlgCl p)) (ν ν' : u.D.toPre.X → ℝ) (h : u.GtN g ν) (h' : u.GtN g ν') : ν = ν'

theorem **j_mem_max** (u : ρ.UTest p y₀) (x : u.V) (hx : x ∈ IsLocalRing.maximalIdeal u.V) : u.j x ∈ IsLocalRing.maximalIdeal (OO p)

theorem **lev_ne_top** (u : ρ.UTest p y₀) (e : ℕ) : u.lev e ≠ ⊤

theorem **exists_level_point** (u : ρ.UTest p y₀) (c : Sp (OO p) ⟶ u.G.left) (hc : c ≫ u.G.hom = spMap u.j) (e : ℕ) : ∃ ce : Sp (OO p ⁄ u.lev e) ⟶ (thickObj u.V (IsLocalRing.maximalIdeal u.V) e u.G).left, ce ≫ (fstG (IsLocalRing.maximalIdeal u.V) e u.G).left = spMap (Ideal.Quotient.mk (u.lev e)) ≫ c

theorem **ser_int** (t : ρ.GeoTest p y₀) (a : t.u.PtC) (w : S.J → PadicAlgCl p) (hw : Crd (t.u.H S.N) t.σ (t.u.toG a) w) : ∃ ν : t.u.D.toPre.X →+ ℝ, Ser t.l.q S.N t.cls w ν ∧ ((∃ m, ∀ α, ν α = QuadraticMap.polar t.l.q m α) → t.u.Int a)

theorem **hasU_of_hasGeo** (h : ρ.HasGeo) : ρ.HasU

theorem **hasTests_of_hasU** (h : ρ.HasU) : S.HasTests

theorem **basis_rep** (u : ρ.UTest p y₀) (l : u.Lab) (k : ℕ) {J : Type} [Fintype J] (σ : J → (u.H k).Sec) (cls : J → u.D.toPre.X) (f : J → ∀ α, u.D.toPre.SecB k α) (h : u.Basis l k σ cls f) (α : u.D.toPre.X) : ∃! β, β ∈ Set.range cls ∧ ∃ m, α = β + k • m

theorem **norm_unit** {X : Scheme.{0}} (ψ : Sp (OO p) ⟶ X) (U : X.Opens) (h : ⊤ ≤ ψ ⁻¹ᵁ U) (f : Γ(X, U)ˣ) : ‖Geo.Tor.valR (gPt p ≫ ψ) U f.val‖ = 1

theorem **norm_indep** {X : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (ψ : K → (Sp (OO p) ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L (fun k => gPt p ≫ ψ k) ε)) (ht : t.IsGen) (j j' : K → L.ι) (hj : ∀ k, ⊤ ≤ ψ k ⁻¹ᵁ L.U (j k)) (hj' : ∀ k, ⊤ ≤ ψ k ⁻¹ᵁ L.U (j' k)) : ‖valF (𝟙 (Sp (PadicAlgCl p))) _ (t.s j).val‖ = ‖valF (𝟙 (Sp (PadicAlgCl p))) _ (t.s j').val‖ ∧ valF (𝟙 (Sp (PadicAlgCl p))) _ (t.s j).val ≠ 0

theorem **exists_ext** (u : ρ.UTest p y₀) (b : TF (R

theorem **exists_gtN** (u : ρ.UTest p y₀) (g : GtPtF u.D.toPre (PadicAlgCl p)) : ∃ ν, u.GtN g ν

Development.AbelianVarietyTheory.Definitions.Degenerations.ExistsMultiplicativeGroup

theorem **valR_eq_valF** {F : Type} [Field F] {X : Scheme.{0}} (x : Sp F ⟶ X) (U : X.Opens) (f : Γ(X, U)) : GlobalTheory.ZH.valR x U f = valF x U f

theorem **FibPre.isLB_H** {k : Type} [Field k] (Φ : FibPre k) (d : ℕ) : (Φ.H d).IsLB

theorem **bijective_constFn** {X : Scheme.{0}} (f : X ⟶ Sp k) [IsIso f] : Function.Bijective (constFn f ⊤)

theorem **constFn_zero** {X : Scheme.{0}} (f : X ⟶ Sp k) (U : X.Opens) : constFn f U 0 = 0

theorem **exists_sec** (E : SplitExt (Sp k)) (G : Over (Sp k)) [GrpObj G] [IsCommMonObj G] (D : E.GtDatum G) (ha : IsAffine E.B.left) (j : E.ChartIdx) (hj : ⊤ ≤ E.chart j) (S : Finset E.X) : ∃ φ : ((trivPre E G D ha j hj).H 1).Sec, ∀ α l, (α ∈ S → (trivPre E G D ha j hj).wt φ α l = 1) ∧ (α ∉ S → (trivPre E G D ha j hj).wt φ α l = 0)

theorem **exists_good** (E : SplitExt (Sp k)) (G : Over (Sp k)) [GrpObj G] [IsCommMonObj G] (D : E.GtDatum G) [IsIso E.B.hom] (ha : IsAffine E.B.left) (j : E.ChartIdx) (hj : ⊤ ≤ E.chart j) (N : ℕ) (q : QuadraticMap ℤ E.X ℝ) (hq : Filter.Tendsto q Filter.cofinite Filter.atTop) (J : Type) [Fintype J] (c : E.X → J) (hc : Function.Surjective c) (hcl : ∀ α β, c α = c β ↔ ∃ m, α = β + N • m) : ∃ (cls : J → E.X) (F : J → ((trivPre E G D ha j hj).H 1).Sec), (∀ i, c (cls i) = i) ∧ (trivPre E G D ha j hj).Good N q cls F

theorem **exists_fibPre** (k : Type) [Field k] (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] : ∃ (Φ : FibPre k) (_ : Φ.E.X ≃ₗ[ℤ] X), (∀ N, ∃ π : Φ.Pol N, π.PB.g = 0) ∧ ∀ (N : ℕ) (q : QuadraticMap ℤ Φ.E.X ℝ), Filter.Tendsto q Filter.cofinite Filter.atTop → ∀ (J : Type) [Fintype J] (c : Φ.E.X → J), Function.Surjective c → (∀ α β, c α = c β ↔ ∃ m, α = β + N • m) → ∃ (cls : J → Φ.E.X) (F : J → (Φ.H 1).Sec), (∀ i, c (cls i) = i) ∧ Φ.Good N q cls F

theorem **exists_multiplicativeGroup** (k : Type) [Field k] : ∃ (Φ : FibPre k) (q : QuadraticMap ℤ Φ.E.X ℝ) (cls : ZMod 81 → Φ.E.X) (F : ZMod 81 → (Φ.H 1).Sec) (π : Φ.Pol 81), Φ.Good 81 q cls F ∧ π.PB.g + Module.finrank ℤ Φ.E.X = 1 ∧ Function.Injective cls

theorem **big_one** : Big 1 81 100000

theorem **affineCharts_isLB** (E : SplitExt (Sp k)) {L : LB E.B.left} (hL : L.IsLB) : (affineCharts E L).IsLB

theorem **ofBundle_iso** (E : SplitExt (Sp k)) (G : Over (Sp k)) [GrpObj G] [IsCommMonObj G] (D : E.GtDatum G) {L : LB E.B.left} (hL : L.IsLB) : LBIso (ofBundle E G D hL).N₀ L

theorem **lbIso_of_gens** {X : Scheme.{0}} {A B : LB X} (s : A.Gen) (hs : s.IsGen) (t : B.Gen) (ht : t.IsGen) : LBIso A B

theorem **exists_gen_self** {X Y : Scheme.{0}} {L : LB X} (hL : L.IsLB) (f f' : Y ⟶ X) (h : f' = f) : ∃ t : (LB.LinComb L (K

theorem **exists_pol** (P : Out.PolAV (Sp k)) (N : ℕ) : ∃ (Φ : FibPre k) (π : Φ.Pol N), π.PB = P

theorem **laurent_const** (a : k) : laurent k X (AddMonoidAlgebra.single 0 (constFn (𝟙 (Sp k)) ⊤ a)) = constFn (torusFib k X).G.hom _ a

theorem **closure_eq_top_iff** {J : Type} (a : J → X) : Subring.closure (Set.range (fun c : k => (AddMonoidAlgebra.single 0 (constFn (𝟙 (Sp k)) ⊤ c) : AddMonoidAlgebra Γ(Sp k, ⊤) X)) ∪ Set.range fun j => AddMonoidAlgebra.single (a j) 1) = ⊤ ↔ AddSubmonoid.closure (Set.range a) = ⊤

theorem **immBy_iff** {J : Type} (a : J → X) (j₀ : J) (h₀ : a j₀ = 0) : ImmBy (torusFib k X).G.hom ((torusFib k X).H 1) (fun j => torusSec k X 1 (AddMonoidAlgebra.single (a j) 1)) {j₀} ↔ AddSubmonoid.closure (Set.range a) = ⊤

theorem **immBy_three** : ImmBy (torusFib k ℤ).G.hom ((torusFib k ℤ).H 1) (fun j => torusSec k ℤ 1 (AddMonoidAlgebra.single (![0, 1, -1] j) 1)) {(0 : Fin 3)}

theorem **not_immBy_two** : ¬ ImmBy (torusFib k ℤ).G.hom ((torusFib k ℤ).H 1) (fun j => torusSec k ℤ 1 (AddMonoidAlgebra.single (![0, 1] j) 1)) {(0 : Fin 2)}

theorem **hilb_univ** {k J : Type} [Field k] [Infinite k] [Finite J] (d : ℕ) : hilb (Set.univ : Set (J → k)) d = Module.finrank k (MvPolynomial.homogeneousSubmodule J k d)

theorem **not_good_empty** (k : Type) [Field k] (N : ℕ) : ¬ (torusFib k ℤ).Good N squareForm (J

Development.AbelianVarietyTheory.Definitions.Degenerations.ExistsPrimitiveRootPadicInt

theorem **res_one** {X : Scheme.{0}} {U V : X.Opens} (h : V ≤ U) : res h (1 : Γ(X, U)ˣ) = 1

theorem **pullU_one** {X Y : Scheme.{0}} (f : Y ⟶ X) (U : X.Opens) : pullU f U (1 : Γ(X, U)ˣ) = 1

theorem **res_res** {X : Scheme.{0}} {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (u : Γ(X, U)ˣ) : res h' (res h u) = res (h'.trans h) u

theorem **oneG_gen** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (oneG φ ε).IsGen

theorem **wchart_triv** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (j : K → (trivLB X).ι) : LB.wchart (trivLB X) φ j = ⊤

theorem **corr_of_one** {X Y : Scheme.{0}} (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (h : ∀ j, t.s j = 1) (h' : ∀ j, t'.s j = 1) : LB.Gen.Corr L φ ε φ' ε' t t'

theorem **isoOne** {X : Scheme.{0}} (L M : LB X) (hL : ∀ i j, L.g i j = 1) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **hypS_zero** : HypS (zeroPolAV B).A (zeroPolAV B).L

theorem **liftFun_bij** (T : Over B) (W : (T ⊗ (zeroPolAV B).A).left.Opens) (hW : W = ⊤) : Function.Bijective (liftFun (zeroPolAV B).A (T

theorem **translT_id** {T : Over B} (x : T ⟶ (zeroPolAV B).A) : (translT (zeroPolAV B).A x).left = 𝟙 _

theorem **corr_comm** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ φ' : K → (Y ⟶ X)) (ε : K → ℤ) (a : LB.Gen (LB.LinComb (trivLB X) φ ε)) (b : LB.Gen (LB.LinComb (trivLB X) φ' ε)) (h h' : Y ⟶ Y) (hh : h = 𝟙 Y) (hh' : h' = 𝟙 Y) (c : LB.Gen (LB.LinComb (trivLB X) (Y

theorem **weil_zero** (k : ℤ) {T : Over B} (x y : T ⟶ (zeroPolAV B).A) : WeilIs (zeroPolAV B).A (zeroPolAV B).L k x y 1

theorem **fn_one** {R : Type} [CommRing R] : fn (1 : R) = 1

theorem **liftFun_one** {T : Over B} (A : Over B) (W : (T ⊗ A).left.Opens) : liftFun A (T

theorem **basis_zero** (T : Over B) (k : ℤ) (J : Type) [Fintype J] [Unique J] (σ : LB.Sec (LPow (zeroPolAV B).A (zeroPolAV B).L k T)) : ∃! c : J → Γ(T.left, ⊤), ∀ j, σ.f j = ∑ i, liftFun (zeroPolAV B).A _ (c i) * (oneS _ _ : LB.Sec (LPow (zeroPolAV B).A (zeroPolAV B).L k T)).f j

theorem **value_one** (k : ℤ) {T : Over B} (z : T ⟶ (zeroPolAV B).A) (τ : LB.Gen (LB.LinComb (zeroPolAV B).L (Y

theorem **mul_s_one** {X Y : Scheme.{0}} (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (h : ∀ j, t.s j = 1) (h' : ∀ j, t'.s j = 1) (j) : (LB.mulGen L φ ε φ' ε' t t').s j = 1

theorem **pull_s_one** {X Y Z : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (f : Z ⟶ Y) (h : ∀ j, t.s j = 1) (j) : (LB.pullGen L φ ε t f).s j = 1

theorem **inv_s_one** {X Y : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (h : ∀ j, t.s j = 1) (j) : (LB.invGen L φ ε t).s j = 1

theorem **rigT_one** (T : Over B) (k : ℤ) (j) : (rigT (zeroPolAV B).A (zeroPolAV B).L (oneG _ _) T k).s j = 1

theorem **tors_zero** (n : ℕ) {T : Over B} (x : T ⟶ (zeroPolAV B).A) : IsTorsFrame (zeroPolAV B) (oneG _ _) n x (oneG _ _)

theorem **symLift_zero** (k : ℤ) (n : ℕ) {T : Over B} (x : T ⟶ (zeroPolAV B).A) : SymLift (zeroPolAV B).A (zeroPolAV B).L k n x ⟨x, oneG _ _, oneG_gen _ _⟩

theorem **twist_one** {T : Over B} (ψ : T ⊗ (zeroPolAV B).A ⟶ T ⊗ (zeroPolAV B).A) (m) (hm : ψ ≫ snd T _ = m) (a b : ℤ) : IsTwistPull (zeroPolAV B).A (zeroPolAV B).L ψ m hm a b (oneG _ _) (oneS _ _) (oneS _ _)

theorem **act_one** {T : Over B} (x : T ⟶ (zeroPolAV B).A) (k : ℤ) (σ : LB.Sec (LPow (zeroPolAV B).A (zeroPolAV B).L k T)) (h : IsActOf (zeroPolAV B).A (zeroPolAV B).L k ⟨x, oneG _ _, oneG_gen _ _⟩ (oneS _ _) σ) (j) : σ.f j = 1

theorem **norm_fracToCl_le** (x : FractionRing ℤ_[p]) : ‖fracToCl p x‖ ≤ 1 ↔ x ∈ Set.range (algebraMap ℤ_[p] (FractionRing ℤ_[p]))

theorem **valF_one** {F : Type} [Field F] {X : Scheme.{0}} (x : Sp F ⟶ X) (U : X.Opens) (hU : U = ⊤) (f : Γ(X, U)) (hf : f = 1) : valF x U f = 1

theorem **psiNorm_utest0** ⟨χ⟩ : (utest0 ζ hζ hu R fg frac p yK).PsiNorm χ 1

theorem **integral_utest0** ⟨a⟩ : (utest0 ζ hζ hu R fg frac p yK).Integral a

theorem **arithBase_intAway** (K : Type) [Field K] [NumberField K] : ArithBase (intAway K) K where

theorem **isUnit_three_intAway** (K : Type) [Field K] [NumberField K] : IsUnit (3 : intAway K)

Development.AbelianVarietyTheory.Definitions.Degenerations.ExistsUniquePartner

theorem **Under.iff_match** {D₀ : DegData R I} {U : PreU I D₀.toPre A G} {U₀ : UnifPre I D₀ A G} : Under U U₀ ↔ GlobalTheory.UU.Match U U₀

theorem **withPsi_eq** (D : DegPre R) (psi) (h) : withPsi D psi h = setPsi D psi h

theorem **spMap_ringHom_id** : spMap (RingHom.id R) = 𝟙 _

theorem **grpPullback_id** {S : Scheme.{0}} (H : Over S) [GrpObj H] {f : S ⟶ S} (hf : f = 𝟙 S) : IsGrpPullback H H f (𝟙 H.left)

theorem **cpMap_id** {S : Scheme.{0}} {H : Over S} [GrpObj H] [IsCommMonObj H] (P : CP H) {f : S ⟶ S} (hf : f = 𝟙 S) (hw : 𝟙 H.left ≫ H.hom = H.hom ≫ f) : IsCPMap P P f (𝟙 H.left) hw (idIso P.L.L)

theorem **mono_fstG** (I : Ideal R) (e : ℕ) (G : Over (Spec (CommRingCat.of R))) : Mono (fstG I e G).left

theorem **pow_le** (C : Cmp φ U U' F) (e : ℕ) : I ^ (e + 1) ≤ (I' ^ (e + 1)).comap φ

theorem **baseMap_thick** (C : Cmp φ U U' F) (e : ℕ) : C.baseMap e ≫ thick R I e = thick V I' e ≫ spMap φ

theorem **levMap_fst** (C : Cmp φ U U' F) (e : ℕ) : C.levMap e ≫ (fstG I e G).left = (fstG I' e G').left ≫ F

theorem **levMap_hom** (C : Cmp φ U U' F) (e : ℕ) : C.levMap e ≫ (thickObj R I e G).hom = (thickObj V I' e G').hom ≫ C.baseMap e

theorem **comp_levMap_fst** (C : Cmp φ U U' F) (e : ℕ) {Z : Scheme.{0}} (x' : Z ⟶ (thickObj V I' e G').left) : (x' ≫ C.levMap e) ≫ (fstG I e G).left = x' ≫ (fstG I' e G').left ≫ F

theorem **existsUnique_partner** (C : Cmp φ U U' F) (e : ℕ) {Z : Scheme.{0}} (x' : Z ⟶ (thickObj V I' e G').left) : ∃! x : Z ⟶ (thickObj R I e G).left, x ≫ (fstG I e G).left = x' ≫ (fstG I' e G').left ≫ F

theorem **levMap_pG** (C : Cmp φ U U' F) (e : ℕ) : C.levMap e ≫ (U.gt e).pG.left = (U'.gt e).pG.left ≫ C.β

theorem **one_fstG** (I : Ideal R) (e : ℕ) (G : Over (Spec (CommRingCat.of R))) [GrpObj G] [IsCommMonObj G] (T : Over (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))) : (1 : T ⟶ thickObj R I e G).left ≫ (fstG I e G).left = (T.hom ≫ thick R I e) ≫ eSec G

theorem **one_pG** {e : ℕ} (g : GtCoord I D e G) (T : Over (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))) : (1 : T ⟶ thickObj R I e G).left ≫ g.pG.left = (T.hom ≫ thick R I e) ≫ eSec D.P.A

theorem **v_rig_level** (C : Cmp φ U U' F) (e : ℕ) {Z : Scheme.{0}} (r : Z ⟶ Spec (CommRingCat.of (V ⧸ I' ^ (e + 1)))) (z : Z) {J J'} (hJ : Good D.P.L z J) (hJ' : Good D'.P.L z J') : val (L

Development.AbelianVarietyTheory.Definitions.Degenerations.InducedModelMorphismCompStructureMap

theorem **mem_OO_iff** {p : ℕ} [Fact p.Prime] (x : PadicAlgCl p) : x ∈ OO p ↔ ‖x‖ ≤ 1

theorem **not_isUnit_OO_iff** {p : ℕ} [Fact p.Prime] (x : OO p) : ¬ IsUnit x ↔ ‖(x : PadicAlgCl p)‖ < 1

theorem **top_le_preimage_iff** {X Y : Scheme.{0}} (y : Y ⟶ X) (U : X.Opens) : ⊤ ≤ y ⁻¹ᵁ U ↔ Set.range y ⊆ U

theorem **top_le_preimage_of_specializes** {X Y : Scheme.{0}} {y : Y ⟶ X} {z : X} (h : ∀ w, y w ⤳ z) {U : X.Opens} (hz : z ∈ U) : ⊤ ≤ y ⁻¹ᵁ U

theorem **pt_apply** (w : Sp Ω) : M.pt w = genericPoint M.Z

theorem **pt_specializes** (w : Sp Ω) (z : M.Z) : M.pt w ⤳ z

theorem **top_le_preimage_pt** {U : M.Z.Opens} (hU : (U : Set M.Z).Nonempty) : ⊤ ≤ M.pt ⁻¹ᵁ U

theorem **nonempty_inf** {U V : M.Z.Opens} (hU : (U : Set M.Z).Nonempty) (hV : (V : Set M.Z).Nonempty) : ((U ⊓ V : M.Z.Opens) : Set M.Z).Nonempty

theorem **res_apply** (U : M.Z.Opens) (hU : (U : Set M.Z).Nonempty) (f : Γ(M.Z, U)) : M.res U hU f = valF M.pt U f

theorem **valF_resG** {U V : M.Z.Opens} (h : V ≤ U) (hV : (V : Set M.Z).Nonempty) (f : Γ(M.Z, U)) : valF M.pt V (resG M.Z h f) = valF M.pt U f

theorem **res_resG** {U V : M.Z.Opens} (h : V ≤ U) (hV : (V : Set M.Z).Nonempty) (f : Γ(M.Z, U)) : M.res V hV (resG M.Z h f) = M.res U (hV.mono h) f

theorem **res_comp_resG** {U V : M.Z.Opens} (h : V ≤ U) (hV : (V : Set M.Z).Nonempty) : (M.res V hV).comp (resG M.Z h) = M.res U (hV.mono h)

private theorem **stalkClosedPointTo_injective** : Function.Injective (Scheme.stalkClosedPointTo (R

theorem **res_injective** (U : M.Z.Opens) (hU : (U : Set M.Z).Nonempty) : Function.Injective (M.res U hU)

theorem **valF_injective** {U : M.Z.Opens} (hU : (U : Set M.Z).Nonempty) : Function.Injective (valF M.pt U)

theorem **resG_eq_of_valF_eq** {U V W : M.Z.Opens} (hU : W ≤ U) (hV : W ≤ V) (hW : (W : Set M.Z).Nonempty) {f : Γ(M.Z, U)} {g : Γ(M.Z, V)} (h : valF M.pt U f = valF M.pt V g) : resG M.Z hU f = resG M.Z hV g

theorem **valF_constFn** {U : M.Z.Opens} (hU : (U : Set M.Z).Nonempty) (a : o) : valF M.pt U (constFn M.s U a) = algebraMap o Ω a

theorem **mem_O** {z : M.Z} {x : Ω} : x ∈ M.O z ↔ ∃ (U : M.Z.Opens) (f : Γ(M.Z, U)), z ∈ U ∧ valF M.pt U f = x

theorem **valF_mem_O** {z : M.Z} {U : M.Z.Opens} (hz : z ∈ U) (f : Γ(M.Z, U)) : valF M.pt U f ∈ M.O z

theorem **exists_affine_of_mem_O** {z : M.Z} {x : Ω} (hx : x ∈ M.O z) {V : M.Z.Opens} (hV : z ∈ V) : ∃ (U : M.Z.Opens) (f : Γ(M.Z, U)), IsAffineOpen U ∧ z ∈ U ∧ U ≤ V ∧ valF M.pt U f = x

theorem **mem_ringOn** {U : Set M.Z} {x : Ω} : x ∈ M.ringOn U ↔ ∀ z ∈ U, x ∈ M.O z

theorem **ringOn_le_O** {U : Set M.Z} {z : M.Z} (hz : z ∈ U) : M.ringOn U ≤ M.O z

theorem **ringOn_anti** {U V : Set M.Z} (h : U ⊆ V) : M.ringOn V ≤ M.ringOn U

theorem **mem_ringOn_iff_exists** {U : M.Z.Opens} (hU : (U : Set M.Z).Nonempty) {x : Ω} : x ∈ M.ringOn U ↔ ∃ f : Γ(M.Z, U), valF M.pt U f = x

theorem **range_res** (U : M.Z.Opens) (hU : (U : Set M.Z).Nonempty) : (M.res U hU).range = (M.ringOn U).toSubring

theorem **coe_sectionsEquiv** (U : M.Z.Opens) (hU : (U : Set M.Z).Nonempty) (f : Γ(M.Z, U)) : (M.sectionsEquiv U hU f : Ω) = valF M.pt U f

theorem **stalkHom_germ** {z : M.Z} {U : M.Z.Opens} (hz : z ∈ U) (f : Γ(M.Z, U)) : M.stalkHom z (M.Z.presheaf.germ U z hz f) = valF M.pt U f

theorem **stalkHom_injective** (z : M.Z) : Function.Injective (M.stalkHom z)

theorem **coe_stalkEquiv** (z : M.Z) (a : M.Z.presheaf.stalk z) : (M.stalkEquiv z a : Ω) = M.stalkHom z a

theorem **stalkEquiv_germ** {z : M.Z} {U : M.Z.Opens} (hz : z ∈ U) (f : Γ(M.Z, U)) : M.stalkEquiv z (M.Z.presheaf.germ U z hz f) = ⟨valF M.pt U f, valF_mem_O hz f⟩

theorem **isUnit_O_iff** {z : M.Z} (x : M.O z) : IsUnit x ↔ (x : Ω) ≠ 0 ∧ (x : Ω)⁻¹ ∈ M.O z

theorem **isUnit_valF_iff** {z : M.Z} {U : M.Z.Opens} (hz : z ∈ U) (f : Γ(M.Z, U)) : IsUnit (⟨valF M.pt U f, valF_mem_O hz f⟩ : M.O z) ↔ z ∈ M.Z.basicOpen f

theorem **O_le_of_specializes** {ξ z : M.Z} (h : ξ ⤳ z) : M.O z ≤ M.O ξ

theorem **O_genericPoint** : M.O (genericPoint M.Z) = ⊤

theorem **exists_eq_div** {V : M.Z.Opens} (hV : IsAffineOpen V) (hne : (V : Set M.Z).Nonempty) (x : Ω) : ∃ a b : Γ(M.Z, V), valF M.pt V b ≠ 0 ∧ x = valF M.pt V a / valF M.pt V b

theorem **isFractionRing_ringOn** {V : M.Z.Opens} (hV : IsAffineOpen V) (hne : (V : Set M.Z).Nonempty) : IsFractionRing (M.ringOn V) Ω

theorem **isOpen_setOf_mem_O** (x : Ω) : IsOpen {z | x ∈ M.O z}

theorem **isIntegrallyClosed_O** (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme M.Z) (z : M.Z) : IsIntegrallyClosed (M.O z)

theorem **isOpen_setOf_div** {J : Type} [Finite J] (t : J → Ω) (i : J) : IsOpen {z | M.Div t i z}

theorem **div_genericPoint** {J : Type} {t : J → Ω} {i : J} (h : t i ≠ 0) : M.Div t i (genericPoint M.Z)

theorem **Div.of_specializes** {J : Type} {t : J → Ω} {i : J} {ξ z : M.Z} (h : M.Div t i z) (hs : ξ ⤳ z) : M.Div t i ξ

theorem **Div.div_iff_isUnit** {J : Type} {t : J → Ω} {i : J} {z : M.Z} (h : M.Div t i z) (k : J) : M.Div t k z ↔ IsUnit ⟨t k / t i, h.2 k⟩ : M.O z)

theorem **div_mul_iff** {J : Type} {t : J → Ω} {i : J} {z : M.Z} {c : Ω} (hc : c ≠ 0) : M.Div (fun j => c * t j) i z ↔ M.Div t i z

private lemma **isIntegral_of_isIntegral_of_le** {S T : Subalgebra o Ω} (hST : S ≤ T) {x : Ω} (hx : IsIntegral S x) : IsIntegral T x

theorem **IsCl.isNm** {J : Type} {t : J → Ω} (h : M.IsCl t) (hn : ∀ z, IsIntegrallyClosed (M.O z)) : M.IsNm t

theorem **val_eq** {y : Sp K ⟶ M.Z} {U : M.Z.Opens} (hy : Set.range y ⊆ U) {f : Γ(M.Z, U)} {x : Ω} (hf : valF M.pt U f = x) : M.val y x = valF y U f

theorem **val_valF** {y : Sp K ⟶ M.Z} {U : M.Z.Opens} (hy : Set.range y ⊆ U) (f : Γ(M.Z, U)) : M.val y (valF M.pt U f) = valF y U f

theorem **val_of_notMem** {y : Sp K ⟶ M.Z} (w : Sp K) {x : Ω} (hx : x ∉ M.O (y w)) : M.val y x = 0

theorem **val_pt** (x : Ω) : M.val M.pt x = x

theorem **val_algebraMap** {y : Sp K ⟶ M.Z} {σ : o →+* K} (hy : y ≫ M.s = spMap σ) (a : o) : M.val y (algebraMap o Ω a) = σ a

theorem **specializes_of_range_eq** {y : Sp K ⟶ M.Z} {z : M.Z} (h : Set.range y = {z}) (w : Sp K) : y w ⤳ z

theorem **range_eq_singleton** (y : Sp K ⟶ M.Z) (w : Sp K) : Set.range y = {y w}

theorem **valLoc_valF** {C : Type} [CommRing C] [IsLocalRing C] (y : Sp C ⟶ M.Z) {z : M.Z} (h : y (IsLocalRing.closedPoint C) ⤳ z) {U : M.Z.Opens} (hz : z ∈ U) (f : Γ(M.Z, U)) : valLoc y z h ⟨valF M.pt U f, valF_mem_O hz f⟩ = valR y U f

theorem **isUnit_valLoc_iff** {C : Type} [CommRing C] [IsLocalRing C] (y : Sp C ⟶ M.Z) {z : M.Z} (h : y (IsLocalRing.closedPoint C) = z) (x : M.O z) : IsUnit (valLoc y z (h ▸ specializes_rfl) x) ↔ IsUnit x

theorem **isLocalHom_valLoc** {C : Type} [CommRing C] [IsLocalRing C] (y : Sp C ⟶ M.Z) {z : M.Z} (h : y (IsLocalRing.closedPoint C) = z) : IsLocalHom (valLoc y z (h ▸ specializes_rfl))

theorem **val_spMap_comp** {C K : Type} [CommRing C] [IsLocalRing C] [Field K] (σ : C →+* K) (y : Sp C ⟶ M.Z) {z : M.Z} (h : y (IsLocalRing.closedPoint C) ⤳ z) (x : M.O z) : M.val (spMap σ ≫ y) x = σ (valLoc y z h x)

theorem **valLoc_eq_val** (y : Sp K ⟶ M.Z) {z : M.Z} (h : y (IsLocalRing.closedPoint K) ⤳ z) (x : M.O z) : valLoc y z h x = M.val y x

@[simp] theorem **valHomOf_apply** (y : Sp K ⟶ M.Z) (z : M.Z) (h : ∀ w, y w ⤳ z) (x : M.O z) : valHomOf y z h x = M.val y x

@[simp] theorem **valHom_apply** (y : Sp K ⟶ M.Z) (z : M.Z) (h : Set.range y = {z}) (x : M.O z) : valHom y z h x = M.val y x

theorem **val_ne_zero_iff** {y : Sp K ⟶ M.Z} {z : M.Z} (h : Set.range y = {z}) (x : M.O z) : M.val y x ≠ 0 ↔ IsUnit x

theorem **ker_valHom** (y : Sp K ⟶ M.Z) (z : M.Z) (h : Set.range y = {z}) : RingHom.ker (valHom y z h) = IsLocalRing.maximalIdeal (M.O z)

theorem **val_add** (h : ∀ w, y w ⤳ z) {a b : Ω} (ha : a ∈ M.O z) (hb : b ∈ M.O z) : M.val y (a + b) = M.val y a + M.val y b

theorem **val_mul** (h : ∀ w, y w ⤳ z) {a b : Ω} (ha : a ∈ M.O z) (hb : b ∈ M.O z) : M.val y (a * b) = M.val y a * M.val y b

theorem **val_one** (y : Sp K ⟶ M.Z) : M.val y 1 = 1

theorem **val_zero** (y : Sp K ⟶ M.Z) : M.val y 0 = 0

theorem **isHom_id** : M.IsHom M (AlgHom.id o Ω) (𝟙 M.Z)

theorem **IsHom.comp** {Ω'' : Type} [Field Ω''] [Algebra o Ω''] {M'' : RatModel o Ω''} {ψ : Ω' →ₐ[o] Ω''} {g : M''.Z ⟶ M'.Z} (hf : M.IsHom M' φ f) (hg : M'.IsHom M'' ψ g) : M.IsHom M'' (ψ.comp φ) (g ≫ f)

theorem **IsHom.inducedModelMorphism_comp_structureMap** (hf : M.IsHom M' φ f) : f ≫ M.s = M'.s

theorem **IsHom.unique** {g : M'.Z ⟶ M.Z} (hf : M.IsHom M' φ f) (hg : M.IsHom M' φ g) : f = g

theorem **IsHom.valF_app** (hf : M.IsHom M' φ f) {U : M.Z.Opens} (hU : (U : Set M.Z).Nonempty) (a : Γ(M.Z, U)) : valF M'.pt (f ⁻¹ᵁ U) (f.app U a) = φ (valF M.pt U a)

theorem **IsHom.map_mem_O** (hf : M.IsHom M' φ f) {z : M'.Z} {x : Ω} (hx : x ∈ M.O (f z)) : φ x ∈ M'.O z

theorem **IsHom.val_comp** (hf : M.IsHom M' φ f) {K : Type} [Field K] (y : Sp K ⟶ M'.Z) (w : Sp K) {x : Ω} (hx : x ∈ M.O (f (y w))) : M.val (y ≫ f) x = M'.val y (φ x)

theorem **IsHom.div** (hf : M.IsHom M' φ f) {J : Type} {t : J → Ω} {i : J} {z : M'.Z} (h : M.Div t i (f z)) : M'.Div (fun j => φ (t j)) i z

theorem **gPt_specializes** (h : y (cPt p) = z) (w : Sp (PadicAlgCl p)) : (gPt p ≫ y) w ⤳ z

theorem **val_gPt** (h : y (cPt p) = z) (x : M.O z) : M.val (gPt p ≫ y) x = (valLoc y z (h ▸ specializes_rfl) x : OO p)

theorem **norm_val_le_one** (h : y (cPt p) = z) {x : Ω} (hx : x ∈ M.O z) : ‖M.val (gPt p ≫ y) x‖ ≤ 1

theorem **norm_val_lt_one_iff** (h : y (cPt p) = z) {x : Ω} (hx : x ∈ M.O z) : ‖M.val (gPt p ≫ y) x‖ < 1 ↔ ¬ IsUnit (⟨x, hx⟩ : M.O z)

theorem **norm_val_eq_one_iff** (h : y (cPt p) = z) {x : Ω} (hx : x ∈ M.O z) : ‖M.val (gPt p ≫ y) x‖ = 1 ↔ IsUnit (⟨x, hx⟩ : M.O z)

theorem **Cen.norm_val_le_one** {U : Set M.Z} (h : M.Cen U z y) {x : Ω} (hx : x ∈ M.O z) : ‖M.val (gPt p ≫ y) x‖ ≤ 1

theorem **Cen.norm_val_lt_one_iff** {U : Set M.Z} (h : M.Cen U z y) {x : Ω} (hx : x ∈ M.O z) : ‖M.val (gPt p ≫ y) x‖ < 1 ↔ ¬ IsUnit (⟨x, hx⟩ : M.O z)

theorem **const_E** (z : M.Z) : (const M J).E z = Submodule.span (M.O z) (Set.range fun i => Pi.single i (1 : Ω))

theorem **mem_const_iff** {z : M.Z} {v : J → Ω} : v ∈ (const M J).E z ↔ ∀ j, v j ∈ M.O z

theorem **valF_spMap_algebraMap** (a : A) : valF (spMap (algebraMap A Ω)) ⊤ ((Scheme.ΓSpecIso (CommRingCat.of A)).inv a) = algebraMap A Ω a

private theorem **valR_global_eq_algebraMap_gammaSpecIso** {L B : Type} [Field L] [CommRing B] [Algebra B L] (y : Γ(Sp B, ⊤)) : valR (spMap (algebraMap B L)) ⊤ y = algebraMap B L ((Scheme.ΓSpecIso (CommRingCat.of B)).hom y)

private theorem **valR_restrict_basicOpen_eq_valR_global** {L B : Type} [Field L] [CommRing B] [Algebra B L] (r y : Γ(Sp B, ⊤)) (h : ⊤ ≤ (spMap (algebraMap B L)) ⁻¹ᵁ (Sp B).basicOpen r) : valR (spMap (algebraMap B L)) _ (algebraMap Γ(Sp B, ⊤) Γ(Sp B, (Sp B).basicOpen r) y) = valR (spMap (algebraMap B L)) ⊤ y

private theorem **le_spMap_preimage_basicOpen_of_algebraMap_ne_zero** {L B : Type} [Field L] [CommRing B] [Algebra B L] (g : B) (hg : algebraMap B L g ≠ 0) : ⊤ ≤ (spMap (algebraMap B L)) ⁻¹ᵁ (Sp B).basicOpen ((Scheme.ΓSpecIso (CommRingCat.of B)).inv g)

theorem **exists_valF_basicOpen_eq_div** (a b : A) (hb : algebraMap A Ω b ≠ 0) : ∃ f : Γ(Sp A, (Sp A).basicOpen ((Scheme.ΓSpecIso (CommRingCat.of A)).inv b)), valF (spMap (algebraMap A Ω)) _ f = algebraMap A Ω a / algebraMap A Ω b

private theorem **exists_valF_basicOpen_eq_div_pow** (g : A) (hg : algebraMap A Ω g ≠ 0) (f : Γ(Sp A, (Sp A).basicOpen ((Scheme.ΓSpecIso (CommRingCat.of A)).inv g))) : ∃ (a : A) (n : ℕ), valF (spMap (algebraMap A Ω)) _ f = algebraMap A Ω a / algebraMap A Ω g ^ n

private theorem **exists_valF_eq_div_of_not_mem_prime** {z : PrimeSpectrum A} {U : (Sp A).Opens} (hz : z ∈ U) (f : Γ(Sp A, U)) : ∃ a b : A, b ∉ z.asIdeal ∧ valF (spMap (algebraMap A Ω)) U f = algebraMap A Ω a / algebraMap A Ω b

theorem **mem_O_ofAffine** {z : PrimeSpectrum A} {x : Ω} : x ∈ (ofAffine o Ω A).O z ↔ ∃ a b : A, b ∉ z.asIdeal ∧ x = algebraMap A Ω a / algebraMap A Ω b

theorem **algebraMap_mem_O_ofAffine** (z : PrimeSpectrum A) (a : A) : algebraMap A Ω a ∈ (ofAffine o Ω A).O z

theorem **inv_mem_O_ofAffine_iff** {z : PrimeSpectrum A} {a : A} (ha : a ≠ 0) : (algebraMap A Ω a)⁻¹ ∈ (ofAffine o Ω A).O z ↔ a ∉ z.asIdeal

theorem **mem_ringOn_basicOpen_ofAffine** {b : A} (hb : b ≠ 0) {x : Ω} : x ∈ (ofAffine o Ω A).ringOn {z : PrimeSpectrum A | b ∉ z.asIdeal} ↔ ∃ (a : A) (n : ℕ), x = algebraMap A Ω a / algebraMap A Ω b ^ n

theorem **mem_ringOn_univ_ofAffine** {x : Ω} : x ∈ (ofAffine o Ω A).ringOn Set.univ ↔ x ∈ (algebraMap A Ω).range

theorem **val_ofAffine** {K : Type} [Field K] (σ : A →+* K) (a : A) : (ofAffine o Ω A).val (spMap σ) (algebraMap A Ω a) = σ a

theorem **isIntegrallyClosed_O_ofAffine** [IsIntegrallyClosed A] (z : PrimeSpectrum A) : IsIntegrallyClosed ((ofAffine o Ω A).O z)

theorem **isHom_ofAffine** {Ω' B : Type} [Field Ω'] [Algebra o Ω'] [CommRing B] [IsDomain B] [Algebra o B] [Algebra.FiniteType o B] [Algebra B Ω'] [IsScalarTower o B Ω'] [IsFractionRing B Ω'] (φ : Ω →ₐ[o] Ω') (ψ : A →+* B) (h : ∀ a, algebraMap B Ω' (ψ a) = φ (algebraMap A Ω a)) : (ofAffine o Ω A).IsHom (ofAffine o Ω' B) φ (spMap ψ)

theorem **isCl_ofAffine_self** [IsDomain o] [IsFractionRing o Ω] {J : Type} (t : J → o) (ht : Ideal.span (Set.range t) = ⊤) : (ofAffine o Ω o).IsCl fun j => algebraMap o Ω (t j)

Development.AbelianVarietyTheory.Definitions.Degenerations.IsAffineSplitTorusLeft

theorem **isUnit_resG_basicOpen** (X : Scheme.{0}) (s : Γ(X, ⊤)) : IsUnit (resG X (le_top : X.basicOpen s ≤ ⊤) s)

theorem **stable_top** (i : splitTorus S X ⟶ H) : Stable i ⊤

theorem **charUnit_zero_val** (S : Scheme.{0}) (X : Type) [AddCommGroup X] : (charUnit S X 0).val = 1

theorem **hasWt_one** (i : splitTorus S X ⟶ H) (V : H.left.Opens) : HasWt i V 1 0

private theorem **appLE_univ_univ_apply_eq_appTop** {A B : Scheme.{0}} (f : A ⟶ B) (u : Γ(B, ⊤)) (h : ⊤ ≤ f ⁻¹ᵁ ⊤) : (f.appLE ⊤ ⊤ h).hom u = f.appTop.hom u

private theorem **scheme_appTop_comp_apply** {A B C : Scheme.{0}} (f : A ⟶ B) (g : B ⟶ C) (u : Γ(C, ⊤)) : (f ≫ g).appTop.hom u = f.appTop.hom (g.appTop.hom u)

private theorem **over_left_appTop_comp_apply** {A B C : Over S} (f : A ⟶ B) (g : B ⟶ C) (u : Γ(C.left, ⊤)) : (f ≫ g).left.appTop.hom u = f.left.appTop.hom (g.left.appTop.hom u)

theorem **hasWt_charUnit** (χ : X) : HasWt (𝟙 (splitTorus S X)) ⊤ (charUnit S X χ).val χ

theorem **HasWt.mem_range_of_id** {u : Γ((splitTorus S X).left, ⊤)} (hu : HasWt (𝟙 (splitTorus S X)) ⊤ u 0) : u ∈ Set.range ((splitTorus S X).hom.appLE ⊤ ⊤ le_top).hom

theorem **Charts.ofTorus_torus** (S : Scheme.{0}) [IsAffine S] (d : ℕ) : (Charts.ofTorus S d).Torus (d

theorem **Charts.ofTorus_split** (S : Scheme.{0}) [IsAffine S] (d : ℕ) : (Charts.ofTorus S d).Split (d

theorem **hasWt_top_iff** (i : splitTorus S X ⟶ H) (u : Γ(H.left, ⊤)) (χ : X) : HasWt i ⊤ u χ ↔ ∀ (Y : Over S) (t : Y ⟶ splitTorus S X) (g : Y ⟶ H), ((t ≫ i) ∗ g).left.appTop.hom u = t.left.appTop.hom (charUnit S X χ).val ∗ g.left.appTop.hom u

theorem **hasWt_iso_iff** (e : splitTorus S X ≅ H) [IsMonHom e.hom] (u : Γ(H.left, ⊤)) (χ : X) : HasWt e.hom ⊤ u χ ↔ HasWt (𝟙 (splitTorus S X)) ⊤ (e.hom.left.appTop.hom u) χ

theorem **Charts.ofTorus_pull_split** (S : Scheme.{0}) [IsAffine S] (d : ℕ) {S' : Scheme.{0}} (t : S' ⟶ S) [IsAffineHom (pullback.fst (splitTorus S (Fin d → ℤ)).hom t)] : Charts.Split (H

theorem **Charts.ofTorus_splitOn** (R : Type) [CommRing R] (d : ℕ) (Sg : Set (Sp R)) : (Charts.ofTorus (Sp R) d).SplitOn (d

Development.AbelianVarietyTheory.Definitions.Degenerations.IsFractionRingR

theorem **isFractionRing_R** (S : ThetaSys o Ω) : IsFractionRing S.R Ω

theorem **idIso_isGen** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) : (idIso L).IsGen

theorem **valR_gen** {X Y : Scheme.{0}} (L : LB X) (e : Y ⟶ X) (t : LB.Gen (L.pull e)) (ht : t.IsGen) {K : Type} [Field K] (y : Sp K ⟶ Y) (i j : L.ι) (hi : Set.range (y ≫ e) ⊆ L.U i) (hj : Set.range (y ≫ e) ⊆ L.U j) : valR y _ (t.s fun _ => i).val = valR (y ≫ e) _ (L.g i j).val ∗ valR y _ (t.s fun _ => j).val

theorem **pre_sec** {S : ThetaSys o Ω} (ρ : S.Real) (c : CubicalStr ρ.P.A (ρ.P.L, ρ.rig, ρ.P.isLB, ρ.rig_gen)) : (pre ρ c).Sec ∧ abLocus (pre ρ c).G = Set.univ ∧ ∀ W : (affModel S).Z.Opens, IsProper (pullback.snd (pre ρ c).G.hom W.ι)

theorem **exists_pre** (hC : CubeSt) [IsNoetherianRing o] (S : ThetaSys o Ω) (ρ : S.Real) : ∃ (M : RatModel o Ω) (Q : ZHPre ρ M ⊤), Q.Sec ∧ abLocus Q.G = Set.univ ∧ ∀ W : M.Z.Opens, IsProper (pullback.snd Q.G.hom W.ι)

Development.AbelianVarietyTheory.Definitions.Degenerations.IsNormalSchemeSp

theorem **isNormalScheme_sp** (R : Type) [CommRing R] [IsDomain R] [IsIntegrallyClosed R] : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme (Sp R)

Development.AbelianVarietyTheory.Definitions.Degenerations.LabKerUnique

theorem **res_one** {X : Scheme.{0}} {U V : X.Opens} (h : V ≤ U) : res h (1 : Γ(X, U)ˣ) = 1

theorem **pullU_one** {X Y : Scheme.{0}} (f : Y ⟶ X) (U : X.Opens) : pullU f U (1 : Γ(X, U)ˣ) = 1

theorem **res_res** {X : Scheme.{0}} {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (u : Γ(X, U)ˣ) : res h' (res h u) = res (h'.trans h) u

theorem **res_pullU_id** {X : Scheme.{0}} (f : X ⟶ X) (hf : f = 𝟙 X) {U V : X.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : V ≤ U) (u : Γ(X, U)ˣ) : res h (pullU f U u) = res h' u

theorem **oneG_gen** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (oneG φ ε).IsGen

theorem **wchart_triv** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (j : K → (trivLB X).ι) : LB.wchart (trivLB X) φ j = ⊤

theorem **corr_of_one** {X Y : Scheme.{0}} (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (h : ∀ j, t.s j = 1) (h' : ∀ j, t'.s j = 1) : LB.Gen.Corr L φ ε φ' ε' t t'

theorem **isoOne** {X : Scheme.{0}} (L M : LB X) (hL : ∀ i j, L.g i j = 1) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **hypS_zero** : HypS (zeroPolAV B).A (zeroPolAV B).L

theorem **liftFun_bij** (T : Over B) (W : (T ⊗ (zeroPolAV B).A).left.Opens) (hW : W = ⊤) : Function.Bijective (liftFun (zeroPolAV B).A (T

theorem **translT_id** {T : Over B} (x : T ⟶ (zeroPolAV B).A) : (translT (zeroPolAV B).A x).left = 𝟙 _

theorem **corr_comm** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ φ' : K → (Y ⟶ X)) (ε : K → ℤ) (a : LB.Gen (LB.LinComb (trivLB X) φ ε)) (b : LB.Gen (LB.LinComb (trivLB X) φ' ε)) (h h' : Y ⟶ Y) (hh : h = 𝟙 Y) (hh' : h' = 𝟙 Y) (c : LB.Gen (LB.LinComb (trivLB X) (Y

theorem **weil_zero** (k : ℤ) {T : Over B} (x y : T ⟶ (zeroPolAV B).A) : WeilIs (zeroPolAV B).A (zeroPolAV B).L k x y 1

theorem **fn_one** {R : Type} [CommRing R] : fn (1 : R) = 1

theorem **liftFun_one** {T : Over B} (A : Over B) (W : (T ⊗ A).left.Opens) : liftFun A (T

theorem **basis_zero** (T : Over B) (k : ℤ) (J : Type) [Fintype J] [Unique J] (σ : LB.Sec (LPow (zeroPolAV B).A (zeroPolAV B).L k T)) : ∃! c : J → Γ(T.left, ⊤), ∀ j, σ.f j = ∑ i, liftFun (zeroPolAV B).A _ (c i) ∗ (oneS _ _ : LB.Sec (LPow (zeroPolAV B).A (zeroPolAV B).L k T)).f j

theorem **value_one** (k : ℤ) {T : Over B} (z : T ⟶ (zeroPolAV B).A) (τ : LB.Gen (LB.LinComb (zeroPolAV B).L (Y

theorem **mul_s_one** {X Y : Scheme.{0}} (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (h : ∀ j, t.s j = 1) (h' : ∀ j, t'.s j = 1) (j) : (LB.mulGen L φ ε φ' ε' t t').s j = 1

theorem **pull_s_one** {X Y Z : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (f : Z ⟶ Y) (h : ∀ j, t.s j = 1) (j) : (LB.pullGen L φ ε t f).s j = 1

theorem **inv_s_one** {X Y : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (h : ∀ j, t.s j = 1) (j) : (LB.invGen L φ ε t).s j = 1

theorem **rigT_one** (T : Over B) (k : ℤ) (j) : (rigT (zeroPolAV B).A (zeroPolAV B).L (oneG _ _) T k).s j = 1

theorem **tors_zero** (n : ℕ) {T : Over B} (x : T ⟶ (zeroPolAV B).A) : IsTorsFrame (zeroPolAV B) (oneG _ _) n x (oneG _ _)

theorem **symLift_zero** (k : ℤ) (n : ℕ) {T : Over B} (x : T ⟶ (zeroPolAV B).A) : SymLift (zeroPolAV B).A (zeroPolAV B).L k n x ⟨x, oneG _ _, oneG_gen _ _⟩

theorem **twist_one** {T : Over B} (ψ : T ⊗ (zeroPolAV B).A ⟶ T ⊗ (zeroPolAV B).A) (m) (hm : ψ ≫ snd T _ = m) (a b : ℤ) : IsTwistPull (zeroPolAV B).A (zeroPolAV B).L ψ m hm a b (oneG _ _) (oneS _ _) (oneS _ _)

theorem **act_one** {T : Over B} (x : T ⟶ (zeroPolAV B).A) (k : ℤ) (σ : LB.Sec (LPow (zeroPolAV B).A (zeroPolAV B).L k T)) (h : IsActOf (zeroPolAV B).A (zeroPolAV B).L k ⟨x, oneG _ _, oneG_gen _ _⟩ (oneS _ _) σ) (j) : σ.f j = 1

theorem **big0** (ζ : o) (hζ : IsPrimitiveRoot ζ ((3 * 81) ^ 2)) (hu : IsUnit ((3 * 81 : ℕ) : o)) (R : Subalgebra o Ω) (fg : R.FG) (frac : ∀ x : Ω, ∃ a ∈ R, ∃ b ∈ R, x = a / b) : (sys0 ζ hζ hu R fg frac).Big

theorem **exists_real** : ∃ S : ThetaSys ℂ ℂ, Nonempty S.Real ∧ S.Big

theorem **delta_eq** (S : ThetaSys o Ω) {p : ℕ} [Fact p.Prime] {y : S.R →+* PadicAlgCl p} (τ : S.Test p y) : S.delta y = τ.δ

theorem **psiNorm_unique** {S : ThetaSys o Ω} {ρ : S.Real} {p : ℕ} [Fact p.Prime] {y₀ : S.R →+* PadicAlgCl p} (u : ρ.UTest p y₀) (χ : u.D.toPre.X) {r r' : ℝ} (h : u.PsiNorm χ r) (h' : u.PsiNorm χ r') : r = r'

theorem **pt_zero** : ρ.pt 0 = 1

theorem **pt_nsmul** (a : S.Gam) (n : ℕ) : ρ.pt (n • a) = ρ.pt a ^ n

theorem **pt_injective** : Function.Injective ρ.pt

theorem **N_odd** (ρ : S.Real) : Odd S.N ∧ 81 ≤ S.N

theorem **lab_q_unique** (l l' : u.Lab) : l.q = l'.q

theorem **lab_ker_unique** (l l' : u.Lab) : l.yy.ker = l'.yy.ker

theorem **lab_wt_eq** (l : u.Lab) (α : u.D.toPre.X) : ∃ x : S.Gam, ∀ t : S.perp l.yy.ker, l.wt α t = ZMod.castHom (dvd_mul_left S.N S.r) (ZMod S.N) (eStd S.g _ t x)

theorem **big_one** (S : ThetaSys o Ω) (hg : S.g = 1) (hN : S.N = 81) : (S.r = 243 → S.Big) ∧ (S.r = 81 → ¬ S.Big)

Development.AbelianVarietyTheory.Definitions.Degenerations.MemOfIsIntegralElem

theorem **mem_of_isIntegralElem** {R C : Type} [CommRing R] [Field C] (S : ValuationSubring C) (j : R →+* C) (hj : ∀ r, j r ∈ S) {x : C} (hx : j.IsIntegralElem x) : x ∈ S

theorem **not_allC_false** (hi : Function.Injective (i.comp g)) (halg : ∀ x : K, ∃ p : Polynomial (PowerSeries O'), p ≠ 0 ∧ p.eval₂ (i.comp g) x = 0) (hg : g.IsIntegral) : ¬ AllC g i fun _ _ => False

theorem **SerBase.not_allC_false** {K : Type} [Field K] [Algebra (FractionRing O') K] {i : PowerSeries O' →+* K} (hi : SerBase i) : ¬ AllC (RingHom.id _) i fun _ _ => False

theorem **ArcPt.not_allC_false** (h : ArcPt Y b g y₁) : ¬ AllC g (algebraMap R₁ (FractionRing R₁)) fun _ _ => False

theorem **exists_sum** (R : K) {n : ℕ} (hn : 0 < n) (hR : R ^ n = i PowerSeries.X) {N : ℕ} (hN : 0 < N) (ω : X → K) (hω : ∀ z, ω z ^ N = 1) (e : X → ℕ) (he : ∀ t, {z | e z ≤ t}.Finite) : ∃ v : K, (∀ S : ValuationSubring K, (∀ r, i r ∈ S) → v ∈ S) ∧ ∀ (C : Type) [NontriviallyNormedField C] [IsUltrametricDist C] [CompleteSpace C] (gam : K →+* C), (∀ f, ‖gam (i f)‖ ≤ 1) → ‖gam (i PowerSeries.X)‖ < 1 → HasSum (fun z => gam (ω z) * gam R ^ e z) (gam v)

theorem **two_halfSq** (y : 𝒪 E) : arcForm l y y = 2 * halfSq l y

theorem **arcForm_symm** (y z : 𝒪 E) : arcForm l y z = arcForm l z y

theorem **halfSq_add** (y z : 𝒪 E) (m : ℕ) : halfSq l (y + m • z) = halfSq l y + m * arcForm l y z + m ^ 2 * halfSq l z

theorem **halfSq_pos** {y : 𝒪 E} (hy : y ≠ 0) : 0 < halfSq l y

theorem **halfSq_zero** : halfSq l 0 = 0

theorem **halfSq_nonneg** (y : 𝒪 E) : 0 ≤ halfSq l y

theorem **finite_halfSq** (y : 𝒪 E) {m : ℕ} (hm : 0 < m) (t : ℤ) : {χ : 𝒪 E | halfSq l (y + m • χ) ≤ t}.Finite

theorem **exists_lift** : ∃ A : Lift ((arcIn c k bF).restrict β) F, Nontrivial A.N.X

theorem **exists_theta** (m : ℕ) (x : HypLattice H) : ∃ v : K, 0 < m → (∀ S : ValuationSubring K, (∀ r, i r ∈ S) → v ∈ S) ∧ ∀ (C : Type) [NontriviallyNormedField C] [IsUltrametricDist C] [CompleteSpace C] (gam : K →+* C), (∀ f, ‖gam (i f)‖ ≤ 1) → ‖gam (i PowerSeries.X)‖ < 1 → HasSum (fun χ => (thTerm l (Units.map gam.toMonoidHom (ζ (m ^ 2))) (Units.map gam.toMonoidHom (r (m ^ 2))) m x χ : C)) (gam v)

theorem **exists_coord** (hc : ∀ n m, 0 < n → 0 < m → r (n * m) ^ m = r n) (m j : ℕ) (z x : HypLattice H) : ∃ w : K, 0 < m → 0 < j → ∀ (C : Type) [NontriviallyNormedField C] [IsUltrametricDist C] [CompleteSpace C] (gam : K →+* C), (∀ f, ‖gam (i f)‖ ≤ 1) → ‖gam (i PowerSeries.X)‖ < 1 → HasSum (fun χ : 𝒪 E => (Units.map gam.toMonoidHom (r (m ^ 2 * j)) ^ halfSq l (m ^ 2 • z.1) * (Units.map gam.toMonoidHom (ζ j) ^ toDual H z.2 x.1 * Units.map gam.toMonoidHom (r j) ^ arcForm l z.1 x.1 * (Units.map gam.toMonoidHom (ζ j) ^ toDual H z.2 χ * Units.map gam.toMonoidHom (r j) ^ arcForm l z.1 χ) ^ m * thTerm l (Units.map gam.toMonoidHom (ζ (m ^ 2))) (Units.map gam.toMonoidHom (r (m ^ 2))) m x χ) : Cˣ).val) (gam w)

theorem **exists_cExt** (W : Type) [CommRing W] [IsDomain W] [IsLocalRing W] (h : CDV W) : ∃ (C : Type) (_ : NontriviallyNormedField C) (_ : IsUltrametricDist C) (_ : CompleteSpace C) (_ : IsAlgClosed C) (_ : CharZero C) (_ : Algebra (FractionRing W) C), CExt W (IsLocalRing.maximalIdeal W) C

theorem **polar_sq** (a a' b : ℤ) : polar (fun n : ℤ => q ^ n ^ 2) (a + a') b = polar (fun n : ℤ => q ^ n ^ 2) a b * polar (fun n : ℤ => q ^ n ^ 2) a' b

Development.AbelianVarietyTheory.Definitions.Degenerations.NormNatCastPadicAlgClOfCoprime

theorem **eStd_add_left** (a a' b : Fin g ⊕ Fin g → ZMod n) : eStd g n (a + a') b = eStd g n a b + eStd g n a' b

theorem **eStd_add_right** (a b b' : Fin g ⊕ Fin g → ZMod n) : eStd g n a (b + b') = eStd g n a b + eStd g n a b'

theorem **eStd_swap** (a b : Fin g ⊕ Fin g → ZMod n) : eStd g n b a = -eStd g n a b

@[simp] theorem **eStd_self** (a : Fin g ⊕ Fin g → ZMod n) : eStd g n a a = 0

@[simp] theorem **eStd_zero_left** (b : Fin g ⊕ Fin g → ZMod n) : eStd g n 0 b = 0

@[simp] theorem **eStd_zero_right** (a : Fin g ⊕ Fin g → ZMod n) : eStd g n a 0 = 0

theorem **eStd_neg_left** (a b : Fin g ⊕ Fin g → ZMod n) : eStd g n (-a) b = -eStd g n a b

theorem **eStd_neg_right** (a b : Fin g ⊕ Fin g → ZMod n) : eStd g n a (-b) = -eStd g n a b

@[simp] theorem **eStdBilin_apply** (a b : Fin g ⊕ Fin g → ZMod n) : eStdBilin g n a b = eStd g n a b

theorem **eStd_single_inr** (a : Fin g ⊕ Fin g → ZMod n) (i : Fin g) : eStd g n a (Pi.single (.inr i) 1) = a (.inl i)

theorem **eStd_single_inl** (a : Fin g ⊕ Fin g → ZMod n) (i : Fin g) : eStd g n a (Pi.single (.inl i) 1) = -a (.inr i)

theorem **eq_zero_of_eStd_eq_zero** {a : Fin g ⊕ Fin g → ZMod n} (h : ∀ b, eStd g n a b = 0) : a = 0

theorem **eStdBilin_bijective** : Function.Bijective (eStdBilin g n)

theorem **nsmul_eq_zero_iff_castHom** {r N : ℕ} [NeZero (r * N)] (a : ZMod (r * N)) : r • a = 0 ↔ ZMod.castHom (dvd_mul_left N r) (ZMod N) a = 0

theorem **mem_perp** {H : AddSubgroup S.Gam} {a : S.Gam} : a ∈ S.perp H ↔ ∀ h ∈ H, eStd S.g _ a h = 0

theorem **mem_perp'** {H : AddSubgroup S.Gam} {a : S.Gam} : a ∈ S.perp H ↔ ∀ h ∈ H, eStd S.g _ h a = 0

theorem **perp_anti** {H H' : AddSubgroup S.Gam} (h : H ≤ H') : S.perp H' ≤ S.perp H

theorem **le_perp_comm** {H H' : AddSubgroup S.Gam} : H ≤ S.perp H' ↔ H' ≤ S.perp H

theorem **le_perp_perp** (H : AddSubgroup S.Gam) : H ≤ S.perp (S.perp H)

@[simp] theorem **perp_bot** : S.perp ⊥ = ⊤

@[simp] theorem **perp_top** : S.perp ⊤ = ⊥

@[simp] theorem **perp_perp** (H : AddSubgroup S.Gam) : S.perp (S.perp H) = H

theorem **bot_isoFree** : S.IsoFree ⊥

theorem **prim_ζN** : IsPrimitiveRoot S.ζN S.N

theorem **psi_apply** (a : ZMod S.N) : S.psi a = S.ζN ^ a.val

theorem **psi_eq_one_iff** {a : ZMod S.N} : S.psi a = 1 ↔ a = 0

theorem **isUnit_natCard** (T : AddSubgroup S.Gam) : IsUnit (Nat.card T : o)

@[simp] theorem **ρ_zero** : S.ρ 0 = 1

theorem **ρ_add_of_eStd** {a b : S.Gam} (h : eStd S.g _ a b = 0) : S.ρ a * S.ρ b = S.ρ (a + b)

theorem **commute_ρ** {a b : S.Gam} (h : eStd S.g _ a b = 0) : Commute (S.ρ a) (S.ρ b)

theorem **ρ_mul_ρ_neg** (a : S.Gam) : S.ρ a * S.ρ (-a) = 1

theorem **isUnit_ρ** (a : S.Gam) : IsUnit (S.ρ a)

theorem **nsmul_eq_zero_of_forall_hom** {T : AddSubgroup S.Gam} {t : T} (h : ∀ c : T →+ ZMod S.N, c t = 0) : S.r • (t : S.Gam) = 0

theorem **sum_psi** [IsDomain o] {A : Type} [AddCommGroup A] [Fintype A] (c : A →+ ZMod S.N) : ∑ a, S.psi (c a) = if c = 0 then (Fintype.card A : o) else 0

theorem **proj_eq_sum** (T : AddSubgroup S.Gam) [Fintype T] (c : T →+ ZMod S.N) : S.proj T c = Ring.inverse (Nat.card T : o) • ∑ t : T, S.psi (-(c t)) • S.ρ t

theorem **proj_eq_one_of_eq_bot** {T : AddSubgroup S.Gam} (h : T = ⊥) (c : T →+ ZMod S.N) : S.proj T c = 1

theorem **commute_ρ_proj** {T : AddSubgroup S.Gam} (c : T →+ ZMod S.N) {h : S.Gam} (hh : h ∈ S.perp T) : Commute (S.ρ h) (S.proj T c)

theorem **ρ_mul_proj** {T : AddSubgroup S.Gam} (hT : T ≤ S.perp T) (c : T →+ ZMod S.N) (t : T) : S.ρ t * S.proj T c = S.psi (c t) • S.proj T c

theorem **proj_mul_ρ** {T : AddSubgroup S.Gam} (hT : T ≤ S.perp T) (c : T →+ ZMod S.N) (t : T) : S.proj T c * S.ρ t = S.psi (c t) • S.proj T c

theorem **proj_mul_of_weight** [IsDomain o] {T : AddSubgroup S.Gam} (c c₀ : T →+ ZMod S.N) {K : Type} {B : Matrix S.J K o} (hB : ∀ t : T, S.ρ t * B = S.psi (c₀ t) • B) : S.proj T c * B = if c = c₀ then B else 0

theorem **proj_mul_proj_of_le** [IsDomain o] {T T' : AddSubgroup S.Gam} (h : T' ≤ T) (hT : T ≤ S.perp T) (c' : T' →+ ZMod S.N) (c : T →+ ZMod S.N) : S.proj T' c' * S.proj T c = if c' = c.comp (AddSubgroup.inclusion h) then S.proj T c else 0

theorem **proj_mul_proj** [IsDomain o] {T : AddSubgroup S.Gam} (hT : T ≤ S.perp T) (c c' : T →+ ZMod S.N) : S.proj T c * S.proj T c' = if c = c' then S.proj T c else 0

theorem **proj_idem** [IsDomain o] {T : AddSubgroup S.Gam} (hT : T ≤ S.perp T) (c : T →+ ZMod S.N) : S.proj T c * S.proj T c = S.proj T c

theorem **proj_mul_proj_eq_zero** [IsDomain o] {T : AddSubgroup S.Gam} (hT : T ≤ S.perp T) {c c' : T →+ ZMod S.N} (h : c ≠ c') : S.proj T c * S.proj T c' = 0

theorem **sum_proj_eq_one** [IsDomain o] (T : AddSubgroup S.Gam) [Fintype (T →+ ZMod S.N)] : ∑ c, S.proj T c = 1

theorem **finsum_proj_eq_one** [IsDomain o] (T : AddSubgroup S.Gam) : ∑ᶠ c, S.proj T c = 1

theorem **IsoFree.proj_spec** [IsDomain o] {T : AddSubgroup S.Gam} (hT : S.IsoFree T) : (∀ c c', S.proj T c * S.proj T c' = if c = c' then S.proj T c else 0) ∧ ∑ᶠ c, S.proj T c = 1 ∧ ∀ c, ∀ h ∈ S.perp T, S.ρ h * S.proj T c = S.proj T c * S.ρ h

theorem **FF_mem** (T : AddSubgroup S.Gam) (c : T →+ ZMod S.N) (k : S.J × (T → ZMod ((S.r * S.N) ^ 2))) : S.FF T c k ∈ S.R

theorem **map_FF** {K : Type} [CommRing K] (y : S.R →+* K) (T : AddSubgroup S.Gam) [Fintype T] (c : T →+ ZMod S.N) (k : S.J × (T → ZMod ((S.r * S.N) ^ 2))) (hk : S.FF T c k ∈ S.R) : y (S.FF T c k, hk) = ∑ t : T, y (algebraMap o S.R S.ζ) ^ (k.2 t).val * ∑ j, y (algebraMap o S.R (S.proj T c j k.1)) * y (S.ev t j, S.ev_mem t j)

theorem **FF_eq_of_eq_bot** {T : AddSubgroup S.Gam} (h : T = ⊥) (c : T →+ ZMod S.N) (k : S.J × (T → ZMod ((S.r * S.N) ^ 2))) : S.FF T c k = algebraMap o Ω (S.ζ ^ (k.2 0).val) * S.ev 0 k.1

theorem **norm_ζ** {p : ℕ} [Fact p.Prime] (y : S.R →+* PadicAlgCl p) : ‖y (algebraMap o S.R S.ζ)‖ = 1

theorem **norm_eq_one_of_isUnit** {p : ℕ} [Fact p.Prime] {y : S.R →+* PadicAlgCl p} (hy : ∀ a : o, ‖y (algebraMap o S.R a)‖ ≤ 1) {a : o} (ha : IsUnit a) : ‖y (algebraMap o S.R a)‖ = 1

theorem **q_pos** {α : τ.X} (h : α ≠ 0) : 0 < τ.q α

theorem **q_nonneg** (α : τ.X) : 0 ≤ τ.q α

theorem **exists_shortest** (β : τ.X) : ∃ v : τ.X, ∀ v0, τ.q (β + S.N • v) ≤ τ.q (β + S.N • v + S.N • v0)

theorem **shortest_zero** (v0 : τ.X) : τ.q 0 ≤ τ.q (0 + S.N • v0)

@[simp] theorem **wt_nsmul** (v : τ.X) : τ.wt (S.N • v) = 0

theorem **perp_ker_le_ker** : S.perp τ.yy.ker ≤ τ.yy.ker

theorem **norm_th_le** (x : S.Gam) : ‖y ⟨S.th x, S.th_mem x⟩‖ ≤ τ.δ

theorem **δ_eq** (τ' : S.Test p y) : τ.δ = τ'.δ

theorem **ker_eq** (τ' : S.Test p y) : τ.yy.ker = τ'.yy.ker

theorem **mem_supI** {z : M.Z} {x : S.Gam} : x ∈ S.supI M z ↔ M.Div S.th x z

@[simp] theorem **perp_TT** (z : M.Z) : S.perp (S.TT M z) = S.HH M z

theorem **supI_mono** {ξ z : M.Z} (h : ξ ⤳ z) : S.supI M z ⊆ S.supI M ξ

theorem **HH_mono** {ξ z : M.Z} (h : ξ ⤳ z) : S.HH M z ≤ S.HH M ξ

theorem **TT_anti** {ξ z : M.Z} (h : ξ ⤳ z) : S.TT M ξ ≤ S.TT M z

theorem **TT_genericPoint** : S.TT M (genericPoint M.Z) = ⊥

theorem **ArithBase.isDomain** (o Ω : Type) [CommRing o] [Field Ω] [Algebra o Ω] [h : ArithBase o Ω] : IsDomain o

theorem **ArithBase.charZero** (o Ω : Type) [CommRing o] [Field Ω] [Algebra o Ω] [h : ArithBase o Ω] : CharZero o

theorem **hasTests_of_eq_one** (hth : ∀ x, S.th x = 1) (hev : ∀ x i, S.ev x i = 1) : S.HasTests

theorem **Test.mem_E_iff_of_norm_eq_one** {p : ℕ} [Fact p.Prime] {y : S.R →+* PadicAlgCl p} (τ : S.Test p y) (hth : ∀ x, ‖y ⟨S.th x, S.th_mem x⟩‖ = 1) (hev : ∀ x i, ‖y (S.ev x i, S.ev_mem x i)‖ = 1) (v : S.J → PadicAlgCl p) : v ∈ τ.E ↔ ∀ j, ‖v j‖ ≤ 1

theorem **isEE_const_of_eq_one** (hth : ∀ x, S.th x = 1) (hev : ∀ x i, S.ev x i = 1) (M : RatModel o Ω) (U₀ : M.Z.Opens) : S.IsEE M U₀ (RatModel.Lat.const M S.J)

theorem **exists_at_of_cen** (M : RatModel o Ω) {U₀ : M.Z.Opens} (hle : S.R ≤ M.ringOn U₀) {z : M.Z} {p : ℕ} [Fact p.Prime] {y : Sp (OO p) ⟶ M.Z} (hc : M.Cen U₀ z y) : ∃ y₀ : S.R →+* PadicAlgCl p, S.At M y y₀ ∧ ∀ a : o, ⦀y₀ (algebraMap o S.R a)⦀ ≤ 1

theorem **finite_FF** : Set.Finite {x : Ω | ∃ T c k, x = S.FF T c k}

theorem **exists_opens_inv_mem** (M : RatModel o Ω) : ∃ U : M.Z.Opens, (U : Set M.Z).Nonempty ∧ (∀ x, (S.th x)⁻¹ ∈ M.ringOn U) ∧ ∀ T c k, (S.FF T c k)⁻¹ ∈ M.ringOn U

theorem **isProjOver_spMap_self** (o : Type) [CommRing o] : IsProjOver (spMap (algebraMap o o))

theorem **isNormalScheme_ofAffine_self** : IsNormalScheme (RatModel.ofAffine o Ω o).Z

theorem **mem_O_or_inv_mem_O** (z : PrimeSpectrum o) (x : Ω) : x ∈ (RatModel.ofAffine o Ω o).O z ∨ x⁻¹ ∈ (RatModel.ofAffine o Ω o).O z

theorem **hasDiv_ofAffine_self** {J : Type} [Finite J] {t : J → Ω} (ht : t ≠ 0) : (RatModel.ofAffine o Ω o).HasDiv t

theorem **ThetaSys.exists_adm** (S : ThetaSys o Ω) (h : S.R = ⊥) : ∃ U₀, S.admitted (RatModel.ofAffine o Ω o) U₀

theorem **exists_cen_ofAffine_self** {p : ℕ} [Fact p.Prime] (φ : o →+* PadicAlgCl p) (hφ : ∀ a, ‖φ a‖ ≤ 1) (hinj : Function.Injective φ) {U : (RatModel.ofAffine o Ω o).Z.Opens} (hU : (U : Set (RatModel.ofAffine o Ω o).Z).Nonempty) : ∃ (z : _) (y : Sp (OO p) ⟶ (RatModel.ofAffine o Ω o).Z), (RatModel.ofAffine o Ω o).Cen U z y

theorem **ThetaSys.exists_admissible_isEulerSystem** (S : ThetaSys o Ω) (h : S.R = ⊥) (hth : ∀ x, S.th x = 1) (hev : ∀ x i, S.ev x i = 1) : ∃ (M : RatModel o Ω) (U₀ : M.Z.Opens) (EE : M.Lat S.J), S.admitted M U₀ ∧ S.IsEE M U₀ EE ∧ ∀ (p : ℕ) [Fact p.Prime], (∃ φ : o →+* PadicAlgCl p, (∀ a, ‖φ a‖ ≤ 1) ∧ Function.Injective φ) → ∃ (z : M.Z) (y : Sp (OO p) ⟶ M.Z) (y₀ : S.R →+* PadicAlgCl p), M.Cen U₀ z y ∧ S.At M y y₀ ∧ (∀ a, ⦀y₀ (algebraMap o S.R a)⦀ ≤ 1) ∧ Nonempty (S.Test p y₀)

theorem **heisMat_mul** {h : ZMod N} (hh : 2 * h = 1) (u v u' v' : ZMod N) : heisMat ψ h u v * heisMat ψ h u' v' = ψ (h * (u * v' - v * u')) • heisMat ψ h (u + u') (v + v')

theorem **heisMat_zero_apply** (h v i j : ZMod N) : heisMat ψ h 0 v i j = if j = i then ψ (v * i) else 0

theorem **heisMat_zero_zero** (h : ZMod N) {u : ZMod N} (v : ZMod N) (hu : u ≠ 0) : heisMat ψ h u v 0 0 = 0

theorem **exists_eq_div_bot** (x : Ω) : ∃ a ∈ (⊥ : Subalgebra o Ω), ∃ b ∈ (⊥ : Subalgebra o Ω), x = a / b

theorem **hasTests_pointSys** : (pointSys Ω hN hr hζ hu).HasTests

theorem **norm_th_heisSys** (x : (𝐒).Gam) : ‖y ⟨(𝐒).th x, (𝐒).th_mem x⟩‖ = ‖y (algebraMap o _ ϖ)‖ ^ (x (.inl (0 : Fin 1))).valMinAbs.natAbs ^ 2

theorem **norm_ev_heisSys** (x : (𝐒).Gam) (j : ZMod N) : ‖y ⟨(𝐒).ev x j, (𝐒).ev_mem x j⟩‖ = ‖y (algebraMap o _ ϖ)‖ ^ (r ^ 2 * N * j.valMinAbs.natAbs ^ 2)

theorem **hasTests_heisSys_one** [Nontrivial o] : (heisSys Ω hN hr hζ hu hodd 1 one_ne_zero).HasTests

theorem **heisSys_ρ_ne_one** [Nontrivial o] : (heisSys Ω hN hr hζ hu hodd ϖ hϖ).ρ (Pi.single (.inl (0 : Fin 1)) 1) ≠ 1

theorem **powers_le** : Submonoid.powers (n : 𝒪 K) ≤ nonZeroDivisors _

theorem **injective** : Function.Injective (algebraMap (𝒪 K) (AwayBase K n))

theorem **finiteType** : Algebra.FiniteType ℤ (AwayBase K n)

theorem **isUnit_natCast** : IsUnit (n : AwayBase K n)

theorem **exists_isPrimitiveRoot** (m : ℕ) [NeZero m] : ∃ ζ : AwayBase (CyclotomicField m ℚ) n, IsPrimitiveRoot ζ m

theorem **norm_natCast_padicAlgCl** (p : ℕ) [Fact p.Prime] (m : ℕ) : ‖(m : PadicAlgCl p)‖ = ‖(m : ℚ_[p])‖

theorem **norm_natCast_padicAlgCl_of_coprime** (p : ℕ) [Fact p.Prime] {m : ℕ} (h : p.Coprime m) : ‖(m : PadicAlgCl p)‖ = 1

theorem **AwayBase.exists_point** (p : ℕ) [Fact p.Prime] (hp : p.Coprime n) : ∃ φ : AwayBase K n →+* PadicAlgCl p, (∀ a, ‖φ a‖ ≤ 1) ∧ Function.Injective φ

theorem **AwayBase.exists_point_bot** [NeZero n] (p : ℕ) [Fact p.Prime] (hp : p.Coprime n) : ∃ y : (⊥ : Subalgebra (AwayBase K n) (FractionRing (AwayBase K n))) →+* PadicAlgCl p, ∀ a, ⦀y (algebraMap (AwayBase K n) _ a)⦀ ≤ 1

theorem **exists_thetaSys_dim_zero** {r N : ℕ} (hN : 2 ≤ N) (hr : 1 ≤ r) : ∃ (o Ω : Type) (_ : CommRing o) (_ : Field Ω) (_ : Algebra o Ω) (_ : ArithBase o Ω) (S : ThetaSys o Ω), S.g = 0 ∧ S.N = N ∧ S.r = r ∧ S.HasTests ∧ ∃ (M : RatModel o Ω) (U₀ : M.Z.Opens) (EE : M.Lat S.J), S.admitted M U₀ ∧ S.IsEE M U₀ EE ∧ ∀ (p : ℕ) [Fact p.Prime], p.Coprime (r * N) → ∃ (z : M.Z) (y : Sp (OO p) ⟶ M.Z) (y₀ : S.R →+* PadicAlgCl p), M.Cen U₀ z y ∧ S.At M y y₀ ∧ (∀ a, ⦀y₀ (algebraMap o S.R a)⦀ ≤ 1) ∧ Nonempty (S.Test p y₀)

theorem **exists_thetaSys_dim_one** {r N : ℕ} (hN : 2 ≤ N) (hr : 1 ≤ r) (hodd : Odd N) : ∃ (o Ω : Type) (_ : CommRing o) (_ : Field Ω) (_ : Algebra o Ω) (_ : ArithBase o Ω) (S : ThetaSys o Ω), S.g = 1 ∧ S.N = N ∧ S.r = r ∧ Nat.card S.J = N ∧ (∃ a, S.ρ a ≠ 1) ∧ S.HasTests ∧ ∃ (M : RatModel o Ω) (U₀ : M.Z.Opens) (EE : M.Lat S.J), S.admitted M U₀ ∧ S.IsEE M U₀ EE ∧ ∀ (p : ℕ) [Fact p.Prime], p.Coprime (r * N) →

∃ (z : M.Z) (y : Sp (OO p) ⟶ M.Z) (y₀ : S.R →+* PadicAlgCl p), M.Cen U₀ z y ∧ S.At M y y₀ ∧ (∀ a, ‖y₀ (algebraMap o S.R a)‖ ≤ 1) ∧ Nonempty (S.Test p y₀)

Development.AbelianVarietyTheory.Definitions.Degenerations.NotAllMatchRankZero

theorem **_root_.GlobalTheory.Ex.NumD.Form.half_add** {N : _root_.GlobalTheory.Ex.NumD O} {k : ℕ} (β : N.Form k) (x y : N.X) : β.be (x + y) (x + y) / 2 = β.be x x / 2 + β.be y y / 2 + β.be x y

theorem **_root_.GlobalTheory.Ex.NumD.Form.div_mul** {N : _root_.GlobalTheory.Ex.NumD O} {k : ℕ} (β : N.Form k) {j : ℕ} (hj : j ∣ k) (x y : N.X) : β.be x y / j * j = β.be x y

theorem **_root_.GlobalTheory.Ex.NumD.Form.div_add** {N : _root_.GlobalTheory.Ex.NumD O} {k : ℕ} (β : N.Form k) {j : ℕ} (hj : j ∣ k) (x y z : N.X) : β.be x (y + z) / j = β.be x y / j + β.be x z / j

theorem **_root_.GlobalTheory.Ex.NumD.twist_τ** {k : ℕ} (β : N.Form k) (u : Oˣ) (x y : N.X) : (N.twist β u).τ x y = N.τ x y * u ^ β.be x y

theorem **_root_.GlobalTheory.Ex.NumD.twG_pow** {k : ℕ} (β : N.Form k) (u : Oˣ) (g : N.Gt) (l : N.X → ℤ) (hl : ∀ a b, l (a + b) = l a + l b) (n : ℕ) : (N.twG β u g l hl ^ n).b = (g ^ n).b ∧ ∀ z : N.X, (N.twG β u g l hl ^ n).s z = (g ^ n).s z * u ^ (l z * n)

theorem **twist_tate_ψ** (q u : Oˣ) (k : ℕ) (hk : k ≠ 0) (n : ℤ) : ((tate q).twist (tateBe q k hk) u).ψ n = (tate (q * u ^ k)).ψ n

theorem **homOfIso_rel** {N N' : NumD O} (I : NumD.Iso N N') (g : N.Gt) (g' : N'.Gt) : (homOfIso I).Rel g g' ↔ I.Rel g g'

theorem **_root_.GlobalTheory.Ex.TateInh.evT_one** (N : _root_.GlobalTheory.Ex.NumD O) (m : ℕ) (p : N.QT (m ^ 2)) (χ : N.X) : N.evT m p 1 χ = N.txT m p χ

theorem **phi_factor** (N : NumD O) {k : ℕ} (p : N.QT k) (h : N.Gt) (α : N.X) : h.s (p.1.1.toAdd + α) * N.uV k (N.c α) p.1.2.b h.b * p.1.2.s α = h.s p.1.1.toAdd * cubN N.θ k 1 p.1.2.b h.b * (h * p.1.2).s α

theorem **txT_twist** (N : NumD O) {k : ℕ} (β : N.Form k) (u : Oˣ) (m : ℕ) (hm : m ^ 2 ∣ k) (ho : Odd m) (p : N.QT (m ^ 2)) (χ : N.X) : (N.twist β u).txT m (N.twQ β u hm p) χ = (u ^ (β.be (p.1.1.toAdd + m • χ) (p.1.1.toAdd + m • χ) / (2 * m ^ 2)) : Oˣ) * N.txT m p χ

theorem **sv_add** {q : Oˣ} (g : (tate q).Gt) (a b : ℤ) : sv g (a + b) = sv g a * sv g b

theorem **tate_cub** {q : Oˣ} (k : ℕ) (a b x : (tate q).A) : cubN (tate q).θ k a b x = 1

theorem **sv_pow** {q : Oˣ} (g : (tate q).Gt) (k : ℕ) (a : ℤ) : sv (g ^ k) a = sv g a ^ k

theorem **sv_mul** {q : Oˣ} (g : (tate q).Gt) (k : ℕ) (a : ℤ) : sv g (k * a) = sv g a ^ k

theorem **sv_q** {q : Oˣ} {k : ℕ} (p : (tate q).QT k) (a : ℤ) : sv p.1.2 a ^ k = q ^ (2 * yv p * a)

theorem **tate_qeq** {q : Oˣ} {k : ℕ} {p p' : (tate q).QT k} (h : (tate q).QEq p p') : ∃ z : ℤ, yv p' = yv p + k * z ∧ ∀ a : ℤ, sv p'.1.2 a = sv p.1.2 a * q ^ (2 * z * a)

theorem **tate_wp_eq** {q : Oˣ} {k : ℕ} (p r : (tate q).QT k) : (tate q).wp p r = sv p.1.2 (yv r) / sv r.1.2 (yv p)

theorem **tate_wp** {q : Oˣ} {k : ℕ} {p p' : (tate q).QT k} (h : (tate q).QEq p p') (r : (tate q).QT k) : (tate q).wp p' r = (tate q).wp p r

theorem **tate_txT** {q : Oˣ} (m : ℕ) (p : (tate q).QT (m ^ 2)) (χ : ℤ) : (tate q).txT m p χ = ((q ^ yv p ^ 2 / sv p.1.2 (yv p) ^ ((m ^ 2 - 1) / 2) * (q ^ χ ^ 2 * sv p.1.2 χ ^ m) : Oˣ) : O)

theorem **tate_tx** [TopologicalSpace O] {q : Oˣ} {m : ℕ} (ho : Odd m) {p p' : (tate q).QT (m ^ 2)} (h : (tate q).QEq p p') : (tate q).tx m p' = (tate q).tx m p

theorem **qEq_incl** (N : NumD O) {k k' : ℕ} (p p' : N.QT k) (q q' : N.QT (k * k')) (hk : k' ≠ 0) (a : q.1.1 = p.1.1 ^ k') (b : q.1.2 = p.1.2) (c : q'.1.1 = p'.1.1 ^ k') (d : q'.1.2 = p'.1.2) : N.QEq p p' ↔ N.QEq q q'

theorem **_root_.GlobalTheory.Ex.TateInh.qEq_mul** (N : _root_.GlobalTheory.Ex.NumD O) {k : ℕ} {p p' q q' : N.QT k} (h : N.QEq p p') (h' : N.QEq q q') : N.QEq (p * q) (p' * q')

theorem **not_allMatch_rank_zero** : ¬ AllMatch fun {_} _ {N} {_} {_} _ _ _ => Subsingleton N.X

theorem **tateMatch_A** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) (hn : Odd n) : (tateMatch q n hn).A

theorem **tateMatch_B** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) (hn : Odd n) : (tateMatch q n hn).B (AddEquiv.refl _)

theorem **tateMatch_not_B** (r : ℂˣ) : ¬ (tateMatch (r ^ 3 ^ 1) 3 (by decide)).B (AddEquiv.neg ℤ)

theorem **tateMatch_term** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) (hn : Odd n) : (tateMatch q n hn).Term (AddEquiv.refl _)

theorem **isoRefl_rel** (N : NumD O) (g g' : N.Gt) : (isoRefl N).Rel g g' ↔ g' = g

theorem **tateMatch_onQ** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) (hn : Odd n) : (tateMatch q n hn).OnQ (isoRefl _)

theorem **tateMatch_onG** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) (hn : Odd n) : (tateMatch q n hn).OnG (isoRefl _)

theorem **not_inG** (r : Oˣ) : ¬ (tate (r ^ 3)).InG (tateQ r 3 1)

theorem **arcTx_coeff_one** : PowerSeries.coeff 1 ((tate (1 : ℤˣ)).arcTx (tateBe 1 1 one_ne_zero).be 1 1) = 2

theorem **tateMatch_onB** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) (hn : Odd n) : (tateMatch q n hn).OnB (isoRefl _).toIso

Development.AbelianVarietyTheory.Definitions.Degenerations.NotFrobGUnitRat

theorem **wt_pos** (i : Option (HeightOneSpectrum k[X])) : 0 < wt k i

theorem **ord_mul** (i) {x y : RatFunc k} (hx : x ≠ 0) (hy : y ≠ 0) : ord k i (x * y) = ord k i x + ord k i y

theorem **ord_nonneg_iff** (i) {x : RatFunc k} (hx : x ≠ 0) : 0 ≤ ord k i x ↔ val k i x ≤ 1

theorem **good_mul** {x y : RatFunc k} (hx : x ≠ 0) (hy : y ≠ 0) (h : Good k x) (h' : Good k y) : Good k (x * y)

theorem **good_inv** {x : RatFunc k} (h : Good k x) : Good k x^{-1}

theorem **ord_none** {p : k[X]} (hp : p ≠ 0) : ord k none (algebraMap k[X] (RatFunc k) p) = -(p.natDegree : ℤ)

theorem **ord_some_of_notMem** {p : k[X]} (P : HeightOneSpectrum k[X]) (hp : p ∉ P.asIdeal) : ord k (some P) (algebraMap k[X] (RatFunc k) p) = 0

theorem **ord_some_self** {p : k[X]} (hp : p ≠ 0) (P : HeightOneSpectrum k[X]) (hP : P.asIdeal = Ideal.span {p}) : ord k (some P) (algebraMap k[X] (RatFunc k) p) = p.natDegree

theorem **good_unit** {p : k[X]} (hp : IsUnit p) : Good k (algebraMap k[X] (RatFunc k) p)

theorem **good_prime** {p : k[X]} (hp : Prime p) : Good k (algebraMap k[X] (RatFunc k) p)

theorem **good_poly** (p : k[X]) (hp : p ≠ 0) : Good k (algebraMap k[X] (RatFunc k) p)

theorem **good_all** {x : RatFunc k} (hx : x ≠ 0) : Good k x

theorem **places_v_X** : (places k).v none (Units.mk0 RatFunc.X RatFunc.X_ne_zero) = -1

theorem **ampleCub_triv** {S : Scheme.{0}} (G : Over S) [GrpObj G] [IsAffine G.left] : AmpleCub G (Inh.trivLB _)

theorem **pullPow_triv** {X : Scheme.{0}} (F : X ⟶ X) (q : ℤ) : PullPow F (Inh.trivLB X) q

theorem **frobG_one_iff** {k : Type} [Field k] (G : Over (Sp k)) [GrpObj G] (q : ℕ) : FrobG G 1 q ↔ ∀ U : G.left.Opens, IsAffineOpen U → ∀ f : Γ(G.left, U), ∃ m, 1 ≤ m ∧ f = f ^ (q ^ m)

theorem **frobG_one_one** {k : Type} [Field k] (G : Over (Sp k)) [GrpObj G] : FrobG G 1 1

theorem **frobG_unit_finite** (k : Type) [Field k] [Fintype k] : FrobG (𝟙_ (Over (Sp k))) 1 (Fintype.card k)

theorem **not_frobG_unit_rat** {q : ℕ} (hq : 2 ≤ q) : ¬ FrobG (𝟙_ (Over (Sp ℚ))) 1 q

theorem **absFrob_comm** {k : Type} [Field k] [Fintype k] {X Y : Over (Sp k)} (f : X.left ⟶ Y.left) : absFrob X ≫ f = f ≫ absFrob Y

theorem **absFrob_base** (k : Type) [Field k] [Fintype k] (X : Over (Sp k)) [IsIso X.hom] : absFrob X = 𝟙 _

theorem **frobOver_comm** {k : Type} [Field k] [Fintype k] {X Y : Over (Sp k)} (f : X ⟶ Y) : frobOver X ≫ f = f ≫ frobOver Y

theorem **frobG_frobOver** {k : Type} [Field k] [Fintype k] (G : Over (Sp k)) [GrpObj G] : FrobG G (frobOver G) (Fintype.card k)

theorem **not_frobG_zero** {k : Type} [Field k] (G : Over (Sp k)) [GrpObj G] (F : End G) : ¬ FrobG G F 0

theorem **places_v_const** {k K : Type} [Field k] [Field K] [Algebra k K] (P : Places k K) (i : P.I) (c : kˣ) : P.v i (Units.map (algebraMap k K : k →* K) c) = 0

Development.AbelianVarietyTheory.Definitions.Degenerations.OfDerivationFnOn

theorem **Model.j_hom** (M : Model R A) : M.j ≫ M.Y.hom = A.hom ≫ spMap (algebraMap R K)

theorem **extends_id** (M : Model R A) (U : M.Y.left.Opens) : Extends M M (𝟙 A) U U.ι

theorem **isoAt_id** (M : Model R A) (y : M.Y.left) : IsoAt M M (𝟙 A) y

theorem **IsoAt.defAt** {M : Model R A} {N : Model R A'} {f : A ⟶ A'} {y : M.Y.left} (h : IsoAt M N f y) : DefAt M N f y

theorem **Model.le_refl** (M : Model R A) : M.Le M

theorem **Model.Le.trans** {M N P : Model R A} (h : M.Le N) (h' : N.Le P) : M.Le P

theorem **derDual_isUnit** (s : S) : IsUnit (derDual d S s)

theorem **locDual_fst** (x : Localization S) : (locDual d S x).fst = x

theorem **locDer_mul** (x y : Localization S) : locDer d S (x * y) = x * locDer d S y + y * locDer d S x

theorem **locDer_algebraMap** (b : B) : locDer d S (algebraMap B _ b) = algebraMap B _ (d b)

theorem **locDer_mk** (a : B) (s : S) : locDer d S (Localization.mk a s) = Localization.mk (s.1 * d a - a * d s.1) (s * s)

theorem **secDer_fnOn** (d : Derivation K B B) (U : (Sp B).Opens) (b : B) : secDer d U (fnOn B U b) = fnOn B U (d b)

theorem **cstK_spec** (U : (Sp B).Opens) (c : K) : cstK (Over.mk (spMap (algebraMap K B))) U c = fnOn B U (algebraMap K B c)

theorem **VF.ofDerivation_fnOn** (d : Derivation K B B) (U : (Sp B).Opens) (b : B) : (VF.ofDerivation d).D U (fnOn B U b) = fnOn B U (d b)

theorem **jac_pderiv** (n : ℕ) (U : (Sp (MvPolynomial (Fin n) K)).Opens) : jac (A

theorem **isFrame_pderiv** (n : ℕ) : IsFrame (A

theorem **polyPt_X** {S : Scheme.{0}} {G Z : Over S} [GrpObj G] (F : End G) (x : Z ⟶ G) : polyPt F Polynomial.X x = x ≫ (show G ⟶ G from F)

theorem **unr_self** {R : Type} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {π : R} (h : Irreducible π) : Unr π R

Development.AbelianVarietyTheory.Definitions.Degenerations.OfTorusBased

private theorem **appLE_univ_univ_apply_eq_appTop** {X Y : Scheme.{0}} (g : Y ⟶ X) (h : ⊤ ≤ g ⁻¹ᵁ (⊤ : X.Opens)) : g.appLE ⊤ ⊤ h = g.appTop

Development.AbelianVarietyTheory.Definitions.Degenerations.PlOrdSomeOfAssociated

theorem **plDeg_pos** (i : Option (MonIrr k)) : 0 < plDeg k i

theorem **plOrd_mul** (i : Option (MonIrr k)) {x y : RatFunc k} (hx : x ≠ 0) (hy : y ≠ 0) : plOrd k i (x * y) = plOrd k i x + plOrd k i y

theorem **plOrd_none** {p : k[X]} (hp : p ≠ 0) : plOrd k none (algebraMap k[X] (RatFunc k) p) = -(p.natDegree : ℤ)

theorem **plOrd_some_of_not_dvd** (π : MonIrr k) {p : k[X]} (h : ¬ π.1 ∣ p) : plOrd k (some π) (algebraMap k[X] (RatFunc k) p) = 0

theorem **plOrd_some_of_associated** (π : MonIrr k) {p : k[X]} (h : Associated π.1 p) : plOrd k (some π) (algebraMap k[X] (RatFunc k) p) = p.natDegree

theorem **PF.mul** {x y : RatFunc k} (hx : x ≠ 0) (hy : y ≠ 0) (a : PF x) (b : PF y) : PF (x * y)

theorem **PF.inv** {x : RatFunc k} (hx : x ≠ 0) (a : PF x) : PF x⁻¹

private theorem **pf_algebraMap_of_isUnit** {u : k[X]} (hu : IsUnit u) : PF (algebraMap k[X] (RatFunc k) u)

private theorem **pf_algebraMap_of_irreducible** {q : k[X]} (hq : Irreducible q) : PF (algebraMap k[X] (RatFunc k) q)

theorem **pf_poly** (p : k[X]) : p ≠ 0 → PF (algebraMap k[X] (RatFunc k) p)

theorem **pf_all** (x : RatFunc k) : x ≠ 0 → PF x

theorem **projLinePlaces_X** : (projLinePlaces k).v none (Units.mk0 RatFunc.X RatFunc.X_ne_zero) = -1

private theorem **appLE_apply_of_eq_id'** {Y : Scheme.{0}} (φ : Y ⟶ Y) (hφ : φ = 𝟙 Y) {U : Y.Opens} (e : U ≤ φ ⁻¹ᵁ U) (f : Γ(Y, U)) : (φ.appLE U U e).hom f = f

theorem **frobG_one** {k : Type} [Field k] (G : Over (Sp k)) [GrpObj G] : FrobG G 1 1

theorem **linComb_triv_gen** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) : ∃ t : LB.Gen (LB.LinComb (Inh.trivLB X) φ ε), t.IsGen

theorem **pullPow_triv** {X : Scheme.{0}} (F : X ⟶ X) (q : ℤ) : PullPow F (Inh.trivLB X) q

theorem **ampleCub_triv** {S : Scheme.{0}} (G : Over S) [GrpObj G] [IsAffine G.left] : AmpleCub G (Inh.trivLB G.left)

Development.AbelianVarietyTheory.Definitions.Degenerations.PullChartIsLB

theorem **valR_of_field** {F : Type} [Field F] {X : Scheme.{0}} (x : Sp F ⟶ X) (U : X.Opens) (f : Γ(X, U)) : valR x U f = valF x U f

theorem **npw_isLB** {X : Scheme.{0}} {H : LB X} (hH : H.IsLB) (d : ℕ) : (npw H d).IsLB where

theorem **pullU_res_eq** {X Y : Scheme.{0}} (f : Y ⟶ X) {U V : X.Opens} (h : V ≤ U) (u : Γ(X, U)ˣ) : pullU f V (res h u) = res (Scheme.Hom.preimage_mono f h) (pullU f U u)

theorem **pullChart_isLB** {X Y : Scheme.{0}} (f : Y ⟶ X) {L : LB X} (hL : L.IsLB) : ((L.ι, fun l => f ⁻¹ᵁ L.U l, fun l l' => pullU f (L.U l ⊓ L.U l') (L.g l l')) : LB Y).IsLB

theorem **FibPre.H_isLB** {k : Type} [Field k] (Φ : FibPre k) (d : ℕ) : (Φ.H d).IsLB

theorem **finite_forms** {k J : Type} [Field k] [Finite J] (C : Set (J → k)) (d : ℕ) : Module.Finite k ((MvPolynomial.homogeneousSubmodule J k d).map (LinearMap.pi fun v : C => (MvPolynomial.aeval (R

Development.AbelianVarietyTheory.Definitions.Degenerations.SepticCharacterInjective

theorem **sym_of_mul** {A U : Type} [CommGroup A] [CommGroup U] {θ : A → A → A → U} (hθ : IsCub θ) {σ : A → U} (h1 : σ 1 = 1) (h : ∀ x y, σ (x * y) * θ x x⁻¹ y⁻¹ = σ x * σ y * θ y y⁻¹ x) : (∀ x, σ x * σ x⁻¹ = 1) ∧ ∀ x y u, θ x⁻¹ y⁻¹ u⁻¹ * σ (x * y) * σ (x * u) * σ (y * u) = θ x y u * σ (x * y * u) * σ x * σ y * σ u

@[simp] theorem **c_zero** : N.c 0 = 1

theorem **c_neg** (a : N.X) : N.c (-a) = (N.c a)⁻¹

theorem **c_nsmul** (k : ℕ) (a : N.X) : N.c (k • a) = N.c a ^ k

theorem **τ_symm** (a b : N.X) : N.τ a b = N.τ b a

theorem **τ_add_left** (a a' b : N.X) : N.τ (a + a') b * N.θ (N.c a) (N.c a') (N.c b) = N.τ a b * N.τ a' b

theorem **τ_add_right** (a b b' : N.X) : N.τ a (b + b') * N.θ (N.c b) (N.c b') (N.c a) = N.τ a b * N.τ a b'

@[simp] theorem **τ_zero_left** (b : N.X) : N.τ 0 b = 1

@[simp] theorem **τ_zero_right** (a : N.X) : N.τ a 0 = 1

@[simp] theorem **ψ_zero** : N.ψ 0 = 1

theorem **ψ_add** (a b : N.X) : N.ψ (a + b) = N.τ a b * N.ψ a * N.ψ b

@[simp] theorem **mul_b** (g g' : N.Gt) : (g * g').b = g.b * g'.b

theorem **mul_s** (g g' : N.Gt) (χ : N.X) : (g * g').s χ = g.s χ * g'.s χ / N.θ (N.c χ) g.b g'.b

@[simp] theorem **one_b** : (1 : N.Gt).b = 1

@[simp] theorem **generator_inv_base** (χ : N.X) : (1 : N.Gt).s χ = 1

@[simp] theorem **inv_b** (g : N.Gt) : g⁻¹.b = g.b⁻¹

theorem **generator_inv_apply** (g : N.Gt) (χ : N.X) : g⁻¹.s χ = N.θ (N.c χ) g.b g.b⁻¹ / g.s χ

@[simp] theorem **s_zero** (g : N.Gt) : g.s 0 = 1

@[simp] theorem **pow_b** (g : N.Gt) (k : ℕ) : (g ^ k).b = g.b ^ k

theorem **pow_s** (g : N.Gt) (k : ℕ) (χ : N.X) : (g ^ k).s χ * cubN N.θ k 1 g.b (N.c χ) = g.s χ ^ k

@[simp] theorem **projB_apply** (g : N.Gt) : N.projB g = g.b

@[simp] theorem **io_b** (y : N.X) : (N.io y).b = N.c y

@[simp] theorem **io_s** (y χ : N.X) : (N.io y).s χ = N.τ y χ

theorem **io_add** (a b : N.X) : N.io (a + b) = N.io a * N.io b

@[simp] theorem **ioHom_apply** (y : Multiplicative N.X) : N.ioHom y = N.io y.toAdd

@[simp] theorem **io_zero** : N.io 0 = 1

theorem **io_neg** (a : N.X) : N.io (-a) = (N.io a)⁻¹

theorem **io_nsmul** (k : ℕ) (a : N.X) : N.io (k • a) = N.io a ^ k

@[simp] theorem **tor_b** (t : Multiplicative N.X →* Oˣ) : (N.tor t).b = 1

@[simp] theorem **torus_generator_apply** (t : Multiplicative N.X →* Oˣ) (χ : N.X) : (N.tor t).s χ = t (Multiplicative.ofAdd χ)

theorem **tor_mul** (t t' : Multiplicative N.X →* Oˣ) : N.tor (t * t') = N.tor t * N.tor t'

@[simp] theorem **torHom_apply** (t : Multiplicative N.X →* Oˣ) : N.torHom t = N.tor t

@[simp] theorem **tor_one** : N.tor 1 = 1

theorem **tor_inv** (t : Multiplicative N.X →* Oˣ) : N.tor t⁻¹ = (N.tor t)⁻¹

theorem **tor_pow** (t : Multiplicative N.X →* Oˣ) (k : ℕ) : N.tor (t ^ k) = N.tor t ^ k

theorem **tor_injective** : Function.Injective N.tor

theorem **exists_tor_iff** (g : N.Gt) : (∃ t, N.tor t = g) ↔ g.b = 1

theorem **range_tor** : Set.range N.tor = {g | g.b = 1}

theorem **range_torHom** : N.torHom.range = N.projB.ker

theorem **tor_mul_s_apply** (t : Multiplicative N.X →* Oˣ) (g : N.Gt) (χ : N.X) : (N.tor t * g).s χ = t (Multiplicative.ofAdd χ) * g.s χ

theorem **mem_QT** {k : ℕ} {p : Multiplicative N.X × N.Gt} : p ∈ N.QT k ↔ p.2 ^ k = N.io p.1.toAdd

theorem **QT_pow_b** {k : ℕ} (p : N.QT k) : p.1.2.b ^ k = N.c p.1.1.toAdd

theorem **qEq_iff** {k : ℕ} (p p' : N.QT k) : N.QEq p p' ↔ ∃ z, p' = p * N.qTriv k z

theorem **qEq_equivalence** (k : ℕ) : Equivalence (N.QEq (k

theorem **wp_self** {k : ℕ} (p : N.QT k) : N.wp p p = 1

theorem **wp_swap** {k : ℕ} (p p' : N.QT k) : N.wp p' p = (N.wp p p')⁻¹

theorem **wp_qTor** {k : ℕ} (t : Multiplicative N.X →* Oˣ) (ht : t ^ k = 1) (p' : N.QT k) : N.wp (N.qTor t ht) p' = t p'.1.1

theorem **wp_of_eq_zero** {k : ℕ} (p p' : N.QT k) (h : p.1.1.toAdd = 0) (h' : p'.1.1.toAdd = 0) : N.wp p p' = N.weil k p.1.2.b p'.1.2.b

theorem **Iso.rel_one** {N' N : NumD 0} (I : Iso N' N) : I.Rel 1 1

theorem **tate_τ** (q : 0ˣ) (a b : ℤ) : (tate q).τ a b = q ^ (2 * a * b)

theorem **tate_vt** (q : 0ˣ) (t : Multiplicative ℤ →* 0ˣ) (n : ℤ) : (tate q).vt ((tate q).tor t) n = (q ^ n ^ 2 * t (Multiplicative.ofAdd n) : 0ˣ)

theorem **tateForm_apply** {C : Type} [NormedField C] (q : Cˣ) (n : ℤ) : tateForm q n = (n : ℝ) ^ 2 * (-Real.log ‖(q : C)‖)

theorem **tate_int** {C : Type} [NormedField C] (q : Cˣ) (hq : ‖(q : C)‖ < 1) : (tate q).Int (tateForm q) where

theorem **tate_tors** {C : Type} [NormedField C] (q : Cˣ) (n : ℕ) : (tate q).toNumB.Tors n 0 where

theorem **isCub_cube** : IsCub cube

theorem **sym_one** : sym 1 = 1

theorem **sym_mul** : ∀ x y, sym (x * y) * cube x x⁻¹ y⁻¹ = sym x * sym y * cube y y⁻¹ x

theorem **cube_nondeg** : ∀ b : A₃, (∀ b', cubN cube 3 1 b' b / cubN cube 3 1 b b' = 1) → b = 1

theorem **bun_nondeg** (f : Multiplicative (ZMod 3) →* 0ˣ) (hf : Function.Injective f) (s : A₃ → 0) (b : A₃) (hb : ∀ b' : A₃, (bun f s).weil 3 b b' = 1) : b = 1

theorem **septicCharacter_injective** : Function.Injective chi7

theorem **exists_weil_ne_one** : ∃ (N : NumD (ZMod 7)) (b b' : N.A), b ^ 3 = 1 ∧ b' ^ 3 = 1 ∧ N.weil 3 b b' ≠ 1

Development.AbelianVarietyTheory.Definitions.Degenerations.ShortSquareFormIff

theorem **exists_chart_eq_top** {k : Type} [Field k] {L : LB (Sp k)} (hL : L.IsLB) : ∃ i, L.U i = ⊤

theorem **lbIso_of_chart_top** {X : Scheme.{0}} {L M : LB X} (hL : L.IsLB) (i₀ : L.ι) (hi : L.U i₀ = ⊤) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **exists_torus_datum** : ∃ (O : X → RigLB (𝟙_ (Over (Sp k)))) (h₁ : _) (h₂ : _) (h₃ : _), Nonempty ((SplitExt.mk (𝟙_ (Over (Sp k))) (Diag.isAbelianScheme_unit _) X O h₁ h₂ h₃).GtDatum (splitTorus (Sp k) X))

theorem **linComb_g_eq_one** {X Y : Scheme.{0}} {L : LB X} (hL : ∀ i j, L.g i j = 1) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (j j' : (LB.LinComb L φ ε).ι) : (LB.LinComb L φ ε).g j j' = 1

theorem **wt_torusSec** (d : ℕ) (x : AddMonoidAlgebra Γ(Sp k, ⊤) X) (α : X) (l) : (torusFib k X).wt (torusSec k X d x) α l = x.coeff α

theorem **constFn_id_bijective** (R : Type) [CommRing R] : Function.Bijective (constFn (𝟙 (Sp R)) ⊤)

theorem **torus_good** (N : ℕ) (q : QuadraticMap ℤ X ℝ) (hq : Filter.Tendsto q Filter.cofinite Filter.atTop) {J : Type} [Fintype J] (cls : J → X) (hs : ∀ α, Short q N α ↔ α ∈ Set.range cls) (hi : ∀ i j m, cls i = cls j + N • m → i = j) : (torusFib k X).Good N q cls fun i => torusSec k X 1 (AddMonoidAlgebra.single (cls i) 1)

theorem **squareForm_apply** (x : ℤ) : squareForm x = ((x ^ 2 : ℤ) : ℝ)

theorem **squareForm_tendsto** : Filter.Tendsto squareForm Filter.cofinite Filter.atTop

theorem **short_squareForm_iff** (r : ℕ) (α : ℤ) : Short squareForm (2 * r + 1) α ↔ -(r : ℤ) ≤ α ∧ α ≤ r

theorem **gm_good** (k : Type) [Field k] (r : ℕ) : (torusFib k ℤ).Good (2 * r + 1) squareForm (J

theorem **big_one_three_pow** : Big 1 81 19683 where

theorem **gm_all** (k : Type) [Field k] : ∃ (Φ : FibPre k) (q : QuadraticMap ℤ Φ.E.X ℝ) (J : Type) (_ : Fintype J) (cls : J → Φ.E.X) (F : J → (Φ.H 1).Sec) (π : Φ.Pol 81), Φ.Good 81 q cls F ∧ Fintype.card J = 81 ∧ π.PB.g + Module.finrank ℤ Φ.E.X = 1 ∧ Big 1 81 19683

Development.AbelianVarietyTheory.Definitions.Divisors.ActAtApply

theorem **actAt_apply** {p q : ℕ} (w : T) (a : Mat p q →ₗ[ℂ] Module.End ℂ (V w)) (Z : Mat p q) : actAt w a Z = atPl w (a Z)

Development.AbelianVarietyTheory.Definitions.Divisors.AdelicAdmissibleSmul

@[simp] lemma **sw_apply** (z : K × K) : sw K z = (z.2, z.1)

lemma **symp_eq** (J : Matrix ι ι (K × K)) (hJ : IsHerm (sw K) J) (x y : Matrix Unit ι (K × K)) : symp (sw K) (xi K) (J₁ K) J x y = (fun j => (x () j).1) ·ᵥ (J.map Prod.fst *ᵥ fun j => (y () j).2) - (fun j => (y () j).1) ·ᵥ (J.map Prod.fst *ᵥ fun j => (x () j).2)

lemma **isUnit_fst** (J : Matrix ι ι (K × K)) (hJ : IsHerm (sw K) J) : IsUnit (J.map Prod.fst).det

lemma **g₁_val** (J : Matrix ι ι (K × K)) (g : UG (sw K) J) : ((g₁ J g : GL ι K) : Matrix ι ι K) = ((g : GL ι (K × K)) : Matrix ι ι (K × K)).map Prod.fst

lemma **unit_fst** (J : Matrix ι ι (K × K)) (g : UG (sw K) J) : ((g₁ J g : GL ι K) : Matrix ι ι K)ᵀ * J.map Prod.fst * ((g : GL ι (K × K)) : Matrix ι ι (K × K)).map Prod.snd = J.map Prod.fst

lemma **om_apply** (lam : Kˣ →* ℂ) (J : Matrix ι ι (K × K)) (g : UG (sw K) J) (φ : (ι → K) → ℂ) (x : ι → K) : om lam J g φ x = lam (Matrix.GeneralLinearGroup.det (g₁ J g)) * φ ((((g₁ J g)⁻¹ : GL ι K) : Matrix ι ι K) *ᵥ x)

lemma **darb_act** (J : Matrix ι ι (K × K)) (hJ : IsHerm (sw K) J) (g : UG (sw K) J) (w : Matrix Unit ι (K × K)) : (darb J hJ).e (actB (sw K) g w) = (((g₁ J g : GL ι K) : Matrix ι ι K) *ᵥ ((darb J hJ).e w).1, (((g₁ J g)⁻¹ : GL ι K) : Matrix ι ι K)ᵀ *ᵥ ((darb J hJ).e w).2)

lemma **om_impl** (ψ : AddChar K ℂ) (lam : Kˣ →* ℂ) (J : Matrix ι ι (K × K)) (hJ : IsHerm (sw K) J) (g : UG (sw K) J) (w : Matrix Unit ι (K × K)) (φ : (ι → K) → ℂ) : om lam J g ((darb J hJ).rho ψ w φ) = (darb J hJ).rho ψ (actB (sw K) g w) (om lam J g φ)

lemma **herm_one** : IsHerm (sw K) (J₁ K)

lemma **herm_dbl** : IsHerm (sw K) (dbl (J₁ K))

lemma **isQuad** : IsQuad (sw K) (xi K) where

lemma **dbl_coord** : IsDblCoord (sw K) (xi K) (J₁ K) (J₁ K) (darb (J₁ K) herm_one) (darb (dbl (J₁ K)) herm_dbl)

lemma **det_siegel** (p : UG (sw K) (dbl (J₁ K))) (t : GL Unit (K × K)) (h : ((p : GL (Unit ⊕ Unit) (K × K)) : Matrix (Unit ⊕ Unit) (Unit ⊕ Unit) (K × K)) * (diagCols Unit : Matrix (Unit ⊕ Unit) Unit (K × K)) = (diagCols Unit : Matrix (Unit ⊕ Unit) Unit (K × K)) * (t : Matrix Unit Unit (K × K))) : (((g₁ _ p : GL (Unit ⊕ Unit) K) : Matrix (Unit ⊕ Unit) (Unit ⊕ Unit) K) *ᵥ fun _ => (1 : K)) = (fun _ => ((t : Matrix Unit Unit (K × K)) () ()).1) ∧ ((g₁ _ p : GL (Unit ⊕ Unit) K) : Matrix (Unit ⊕ Unit) (Unit ⊕ Unit) K).det * ((t : Matrix Unit Unit (K × K)) () ()).2 = ((t : Matrix Unit Unit (K × K)) () ()).1

lemma **inv_form** (ψ : AddChar K ℂ) (hψ : ψ ≠ 1) (l : ((Unit ⊕ Unit → K) → ℂ) →ₗ[ℂ] ℂ) (hl : ∀ (z : Matrix Unit Unit (K × K)) φ, l ((darb (dbl (J₁ K)) herm_dbl).rho ψ (z * (diagCols Unit : Matrix (Unit ⊕ Unit) Unit (K × K))ᵀ) φ) = l φ) (g : GL (Unit ⊕ Unit) K) (s : K) (hg

: ((g : Matrix (Unit ⊕ Unit) (Unit ⊕ Unit) K) ∗ᵥ fun _ => (1 : K)) = fun _ => s) (φ : (Unit ⊕ Unit → K) → ℂ) : l (fun x => φ (((g⁻¹ : GL _ K) : Matrix (Unit ⊕ Unit) (Unit ⊕ Unit) K) ∗ᵥ x)) = l φ

lemma **mem_sp** (κ : Type) [Fintype κ] (φ : (κ → K) → ℂ) : φ ∈ (sbClass K).sp κ

lemma **modu_one** (u : Kˣ) : (sbClass K).modu u = 1

lemma **om_split** (ψ : AddChar K ℂ) (lam : Kˣ →∗ ℂ) (J : Matrix ι ι (K × K)) (hJ : IsHerm (sw K) J) : IsSplitting (sbClass K) ψ (darb J hJ) (actB (sw K)) (om lam J)

lemma **norm_onto** (a : Kˣ) : a ∈ (⊥ : Subgroup Kˣ) ⊔ (Units.map (Algebra.norm K : K × K →∗ K)).range

theorem **isWeilB** (ψ : AddChar K ℂ) (hψ : ψ ≠ 1) (χ : (K × K)ˣ →∗ Circle) (hχ : ResEtaPow (R

lemma **om_comm** (lam lam' : Kˣ →∗ ℂ) (g h : UG (sw K) (J₁ K)) : Commute (om lam (J₁ K) g) (om lam' (J₁ K) h)

theorem **isWeilPair** (ψ : AddChar K ℂ) (hψ : ψ ≠ 1) (χA χB : (K × K)ˣ →∗ Circle) (hA : ResEtaPow (R

theorem **schr_mul** {R : Type} [CommRing R] [Invertible (2 : R)] (ψ : AddChar R ℂ) {κ : Type} [Fintype κ] (w w' : (κ → R) × (κ → R)) (φ : (κ → R) → ℂ) : schr ψ w (schr ψ w' φ) = ψ (⅟(2 : R) ∗ (w.1 ⬝ᵥ w'.2 − w'.1 ⬝ᵥ w.2)) • schr ψ (w + w') φ

theorem **exists_three** : ∃ (_ : Invertible (2 : ZMod 3)) (ψ : AddChar (ZMod 3) ℂ) (χ : (ZMod 3 × ZMod 3)ˣ →∗ Circle), ψ ≠ 1 ∧ ResEtaPow (R

theorem **isQuad_complex** : IsQuad (Complex.conjAe : ℂ →ₐ[ℝ] ℂ) Complex.I where

theorem **om_split_real** {ι : Type} [Fintype ι] [DecidableEq ι] [Invertible (2 : ℝ)] (ψ : AddChar ℝ ℂ) (lam : ℝˣ →∗ ℂ) (J : Matrix ι ι (ℝ × ℝ)) (hJ : IsHerm (sw ℝ) J) : IsSplitting realClass ψ (darb J hJ) (actB (sw ℝ)) (om lam J)

theorem **det_siegel** {JB : Matrix β β E} (hB : IsHerm c JB) (p : UG c (dbl JB)) (t : Matrix β β E) (h : ((p : GL (β ⊕ β) E) : Matrix (β ⊕ β) (β ⊕ β) E) ∗ (diagCols β : Matrix (β ⊕ β) β E) = (diagCols β : Matrix (β ⊕ β) β E) ∗ t) : ((p : GL (β ⊕ β) E) : Matrix (β ⊕ β) (β ⊕ β) E).det ∗ c t.det = t.det

theorem **rho_diag** (ψ : AddChar R ℂ) {ξ : E} {JA : Matrix α α E} {JB : Matrix β β E} {d : Darboux (symp c ξ JA JB) κ} {dd : Darboux (symp c ξ JA (dbl JB)) (κ ⊕ κ)} (h : IsDblCoord c ξ JA JB d dd) (z : Matrix α β E) (φ : (κ ⊕ κ → R) → ℂ) (u : κ → R) : dd.rho ψ (z ∗ (diagCols β : Matrix (β ⊕ β) β E)ᵀ) φ (Sum.elim u u) = φ (Sum.elim (u + (d.e z).1) (u + (d.e z).1))

theorem **exists_levi** {ξ : E} (hq : IsQuad c ξ) (JB : Matrix Unit Unit E) (t : Eˣ) : ∃ (p : UG c (dbl JB)) (T : GL Unit E), GeneralLinearGroup.det T = t ∧ ((p : GL (Unit ⊕ Unit) E) : Matrix (Unit ⊕ Unit) (Unit ⊕ Unit) E) ∗ (diagCols Unit : Matrix (Unit ⊕ Unit) Unit E) = (diagCols Unit : Matrix (Unit ⊕ Unit) Unit E) ∗ (T : Matrix Unit Unit E)

theorem **hasDiagForm_of** (C : TestClass R) (S : Submodule ℂ ((κ → R) → ℂ)) (m : S →ₗ[ℂ] ℂ) (hS : ∀ φ ∈ C.sp (κ ⊕ κ), (fun u => φ (Sum.elim u u)) ∈ S) (hT : ∀ (f : S) (a : κ → R), ∃ h : (fun u => f.1 (u + a)) ∈ S, m ⟨_, h⟩ = m f) (h0 : ∃ φ, ∃ h : φ ∈ C.sp (κ ⊕ κ), m ⟨_, hS φ h⟩ ≠ 0) (hadm : ∀ l : ((κ ⊕ κ → R) → ℂ) →ₗ[ℂ] ℂ, (∀ φ (h : φ ∈ C.sp (κ ⊕ κ)), l φ = m ⟨_, hS φ h⟩) → l ∈ C.adm (κ ⊕ κ)) : HasDiagForm C κ

theorem **siegel_scalar_unique** {C : TestClass R} (hC : HasDiagForm C κ) {ψ : AddChar R ℂ} {ξ : E} {JA : Matrix α α E} {JB : Matrix β β E} {d : Darboux (symp c ξ JA JB) κ} {dd : Darboux (symp c ξ JA (dbl JB)) (κ ⊕ κ)} (h : IsDblCoord c ξ JA JB d dd) {χ χ' : Eˣ →∗ Circle} {om : UG c (dbl JB) →∗ Module.End ℂ ((κ ⊕ κ → R) → ℂ)} (h1 : SiegelRule c C ψ ξ JA (dbl JB) dd χ (diagCols β) om) (h2 : SiegelRule c C ψ ξ JA (dbl JB) dd χ' (diagCols β) om) (p : UG c (dbl JB)) (t : GL β E) (hp : ((p : GL (β ⊕ β) E) : Matrix (β ⊕ β) (β ⊕ β) E) ∗ (diagCols β : Matrix (β ⊕ β) β E) = (diagCols β : Matrix (β ⊕ β) β E) ∗ (t : Matrix β β E)) : χ (GeneralLinearGroup.det t) = χ' (GeneralLinearGroup.det t)

theorem **siegel_char_unique** {C : TestClass R} (hC : HasDiagForm C κ) {ψ : AddChar R ℂ} {ξ : E} (hq : IsQuad c ξ) {JA : Matrix α α E} {JB : Matrix Unit Unit E} {d : Darboux (symp c ξ JA JB) κ} {dd : Darboux (symp c ξ JA (dbl JB)) (κ ⊕ κ)} (h : IsDblCoord c ξ JA JB d dd) {χ χ' : Eˣ →∗ Circle} {om : UG c (dbl JB) →∗ Module.End ℂ ((κ ⊕ κ → R) → ℂ)} (h1 : SiegelRule c C ψ ξ JA (dbl JB) dd χ (diagCols Unit) om) (h2 : SiegelRule c C ψ ξ JA (dbl JB) dd χ' (diagCols Unit) om) : χ = χ'

theorem **hasDiagForm_sb** (F : Type) [Field F] [ValuativeRel F] [TopologicalSpace F] [IsNonarchimedeanLocalField F] (κ : Type) [Fintype κ] : HasDiagForm (sbClass F) κ

theorem **mem_schFun** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] {φ : V → ℂ} : φ ∈ schFun V ↔ ∃ f : SchwartzMap V ℂ, (f : V → ℂ) = φ

theorem **hasDiagForm_real** (κ : Type) [Fintype κ] : HasDiagForm realClass κ

theorem **detOm_apply** (lam : Eˣ →∗ Circle) (J : Matrix β β E) (X : Type) (g : UG c J) (φ : X → ℂ) : detOm lam J X g φ = (lam (GeneralLinearGroup.det (g : GL β E)) : ℂ) • φ

theorem **isUSplitting_detOm** (hC : ∀ (κ : Type) [Fintype κ] (a : ℂ), ∀ l ∈ C.adm κ, a • l ∈ C.adm κ) (ψ : AddChar R ℂ) {ξ : E} {JA : Matrix α α E} {JB : Matrix β β E} [IsEmpty κ] (d : Darboux (symp c ξ JA JB) κ) {lam : Eˣ →∗ Circle} (hl : Continuous lam) : IsUSplitting C ψ d (actB c) (detOm lam JB _)

theorem **isUWeilB_rank_zero** (hC : ∀ (κ : Type) [Fintype κ] (a : ℂ), ∀ l ∈ C.adm κ, a • l ∈ C.adm κ) (ψ : AddChar R ℂ) {ξ : E} {JA : Matrix α α E} {JB : Matrix β β E} (hB : IsHerm c JB) [IsEmpty α] [IsEmpty κ] (d : Darboux (symp c ξ JA JB) κ) {lam : Eˣ →∗ Circle} (hl : Continuous lam) : IsUWeilB c C ψ ξ JA JB d (lam.comp (h90 c)) (detOm lam JB _)

theorem **real_admissible_smul** (κ : Type) [Fintype κ] (a : ℂ) : ∀ l ∈ realClass.adm κ, a • l ∈ realClass.adm κ

theorem **adelic_admissible_smul** (F : Type) [Field F] [NumberField F] [LocallyCompactSpace (AdeleRing (𝒪 F) F)] (κ : Type) [Fintype κ] (a : ℂ) : ∀ l ∈ (adeleClass F).adm κ, a • l ∈ (adeleClass F).adm κ

theorem **argC_apply** (z : ℂˣ) : (argC z : ℂ) = z / ‖(z : ℂ)‖

theorem **continuous_argC** : Continuous argC

theorem **argC_h90_conj_eq_sq** (z : ℂˣ) : argC (h90 (Complex.conjAe : ℂ →ₐ[ℝ] ℂ) z) = argC z ^ 2

theorem **isUWeilB_complex** [Invertible (2 : ℝ)] [IsModuleTopology ℝ ℂ] (ψ : AddChar ℝ ℂ) {β : Type} [Fintype β] [DecidableEq β] {JB : Matrix β β ℂ} (hB : IsHerm (Complex.conjAe : ℂ →ₐ[ℝ] ℂ) JB) (k : ℤ) : IsUWeilB (Complex.conjAe : ℂ →ₐ[ℝ] ℂ) realClass ψ Complex.I (1 : Matrix Empty Empty ℂ) JB (darbZero _ _ _ Empty) (argC ^ (2 ∗ k)) (detOm (argC ^ k) JB _)

theorem **resEtaPow_argC** : ResEtaPow (⊥ : Subgroup ℝˣ) argC 1

theorem **real_modu** (a : ℝˣ) : (realClass.modu a : ℝ) = |(a : ℝ)|

theorem **isUWeilB_adele** (F : Type) [Field F] [NumberField F] : ∃ (_ : LocallyCompactSpace (AdeleRing (𝒪 F) F)) (_ : Invertible (2 : AdeleRing (𝒪 F) F)), IsQuad (swapA (AdeleRing (𝒪 F) F)) (1, -1) ∧ ∀ (ψ : AddChar (AdeleRing (𝒪 F) F) ℂ) (lam : (AdeleRing (𝒪 F) F × AdeleRing (𝒪 F) F)ˣ →∗ Circle), Continuous lam → IsUWeilB (swapA (AdeleRing (𝒪 F) F)) (adeleClass F) ψ (1, -1) (1 : Matrix Empty Empty _) (1 : Matrix Unit Unit _) (darbZero _ _ _ Empty) (lam.comp (h90 (swapA _))) (detOm lam _ _)

theorem **split_char_unique** {F : Type} [Field F] [ValuativeRel F] [TopologicalSpace F] [IsNonarchimedeanLocalField F] [Invertible (2 : F)] {α κ : Type} [Fintype α] [DecidableEq α] [Fintype κ] (ψ : AddChar F ℂ) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) {JA : Matrix α α (F × F)} {JB : Matrix Unit Unit (F × F)} (hA : IsHerm (swapA F) JA) (hB : IsHerm (swapA F) JB) (σ : κ ≃ α × Unit) (χ χ' : (F × F)ˣ →* Circle) (hc : Continuous χ) (hχ : ResEtaPow (⊥ : Subgroup Fˣ) χ (Fintype.card α)) : IsUWeilSplit (swapA F) (sbClass F) ψ (1, -1) JA (dbl JB) (splitDarboux hA hB.dbl (dblNum σ)) χ' (diagCols Unit) (splitOm (distribHaarChar F) χ (dbl JB) (dblNum σ)) ↔ χ' = χ

Development.AbelianVarietyTheory.Definitions.Divisors.AdjKltOfLtDisc

theorem **form_ne_zero** (h : AdjKlt c n T S η φ) : η ≠ 0

theorem **apply_zero** (h : AdjKlt c n T S η φ) : φ 0 = 0

theorem **lt_disc** (h : AdjKlt c n T S η φ) {Q : Subring K} (hQ : IsLocModel c Q) {t u : Q} {E : Place c} (hT : T ≤ Q) (hm : Ideal.span {t, u} = (letI

theorem **mono** (h : AdjKlt c n T S η φ) (hT : T ≤ T') : AdjKlt c n T' S η φ

theorem **mem_maximalIdeal_iff** (P : Place c) (b : P.ring) (hb : (b : K) ≠ 0) : b ∈ IsLocalRing.maximalIdeal P.ring ↔ P.ord (b : K) ≠ 0

theorem **neighbour_ne** {Q : Subring K} (hQ : IsLocModel c Q) {t u : Q} {S E : Place c} (hm : Ideal.span {t, u} = (letI

theorem **adjKlt_of_lt_disc** (S : Place c) (hη : η ≠ 0) (hφ : φ 0 = 0) (h : ∀ E : Place c, E.Contains T → E.ring ≠ S.ring → -1 < E.disc η φ) : AdjKlt c n T S η φ

Development.AbelianVarietyTheory.Definitions.Divisors.AevalEqZeroOfMemConeIdeal

theorem **coneMap_of_isHomogeneous** (y : Fin N → Ω) {d : ℕ} {g : MvPolynomial (Fin N) K} (hg : g.IsHomogeneous d) : coneMap K y g = Polynomial.X ^ d * Polynomial.C (aeval y g)

theorem **mem_coneIdeal_of_isHomogeneous** (y : Fin N → Ω) {d : ℕ} {g : MvPolynomial (Fin N) K} (hg : g.IsHomogeneous d) : g ∈ coneIdeal K y ↔ aeval y g = 0

theorem **coeff_coneMap** (y : Fin N → Ω) (g : MvPolynomial (Fin N) K) (d : ℕ) : (coneMap K y g).coeff d = aeval y (homogeneousComponent d g)

theorem **mem_coneIdeal_iff** (y : Fin N → Ω) (g : MvPolynomial (Fin N) K) : g ∈ coneIdeal K y ↔ ∀ d, aeval y (homogeneousComponent d g) = 0

theorem **isPrime_coneIdeal** (y : Fin N → Ω) : (coneIdeal K y).IsPrime

theorem **isHomogeneous_coneIdeal** (y : Fin N → Ω) : (coneIdeal K y).IsHomogeneous (homogeneousSubmodule (Fin N) K)

theorem **aeval_eq_zero_of_mem_coneIdeal** {y : Fin N → Ω} {g : MvPolynomial (Fin N) K} (hg : g ∈ coneIdeal K y) : aeval y g = 0

theorem **coneIdeal_smul** {a : Ω} (ha : a ≠ 0) (y : Fin N → Ω) : coneIdeal K (a • y) = coneIdeal K y

theorem **coneIdeal_eq_bot** {y : Fin N → Ω} (h : AlgebraicIndependent K y) : coneIdeal K y = ⊥

theorem **mem_coneIdeal_coord** {N' : ℕ} (M : Matrix (Fin N) (Fin N') K) (y : Fin N → Ω) (g : MvPolynomial (Fin N') K) : g ∈ coneIdeal K (fun k => ∑ j, algebraMap K Ω (M j k) * y j) ↔ coordSubst M g ∈ coneIdeal K y

theorem **aeval_coordSubst** {Ω : Type} [CommRing Ω] [Algebra K Ω] (M : Matrix (Fin N) (Fin N') K) (y : Fin N → Ω) (g : MvPolynomial (Fin N') K) : aeval y (coordSubst M g) = aeval (fun k => ∑ j, algebraMap K Ω (M j k) * y j) g

theorem **coordSubst_coordSubst** {M : Matrix (Fin N) (Fin N') K} {P : Matrix (Fin N') (Fin N) K} (h : M * P = 1) (g : MvPolynomial (Fin N) K) : coordSubst M (coordSubst P g) = g

theorem **sum_sum_eq** {Ω : Type} [CommRing Ω] [Algebra K Ω] {M : Matrix (Fin N) (Fin N') K} {P : Matrix (Fin N') (Fin N) K} (h : M * P = 1) (y : Fin N → Ω) (j : Fin N) : ∑ k, algebraMap K Ω (P k j) * ∑ i, algebraMap K Ω (M i k) * y i = y j

theorem **isChowFormInt_linSubst** {M : Matrix (Fin N) (Fin N') K} {P : Matrix (Fin N') (Fin N) K} (h : M * P = 1) (h' : P * M = 1) {I : Ideal (MvPolynomial (Fin N) K)} {I' : Ideal (MvPolynomial (Fin N') K)} (hI : ∀ g, g ∈ I' ↔ coordSubst M g ∈ I) {D δ : ℕ} {f : MvPolynomial (Fin (D + 1) × Fin N) K} (hf : IsChowFormInt I D δ f) : IsChowFormInt I' D δ (linSubst M f)

theorem **isChowFormInt_basis** (ev : V →ₗ[K] Ω) {N N' : ℕ} (b : Basis (Fin N) K V) (b' : Basis (Fin N') K V) {D δ : ℕ} {f : MvPolynomial (Fin (D + 1) × Fin N) K} (hf : IsChowFormInt (imageIdeal ev b) D δ f) : IsChowFormInt (imageIdeal ev b') D δ (linSubst (b.toMatrix b') f)

theorem **IsImageDeg.unique** {ev : V →ₗ[K] Ω} {D δ δ' : ℕ} (h : IsImageDeg ev D δ) (h' : IsImageDeg ev D δ') : δ = δ'

theorem **IsImageDeg.pos** {ev : V →ₗ[K] Ω} {D δ : ℕ} (h : IsImageDeg ev D δ) : 0 < δ

theorem **IsImageDeg.imageDeg_eq** {ev : V →ₗ[K] Ω} {D δ : ℕ} (h : IsImageDeg ev D δ) : imageDeg ev D = δ

theorem **imageDeg_pos_iff** (ev : V →ₗ[K] Ω) (D : ℕ) : 0 < imageDeg ev D ↔ IsImageDeg ev D (imageDeg ev D)

theorem **isImageHeight_imageHeight** (h : ∃ x, IsImageHeight Nm L ev D x) : IsImageHeight Nm L ev D (imageHeight Nm L ev D)

theorem **exists_isImageHeight_iff** : (∃ x, IsImageHeight Nm L ev D x) ↔ (∃ (N n : ℕ) (b : Basis (Fin N) K V) (w : Matrix (Fin N) (Fin n) K) (A : (K →+* ℂ) → Matrix (Fin N) (Fin N) ℂ), IsFrame Nm L b w A) ∧ 0 < imageDeg ev D

theorem **Presents.isImageHeight** {T : Adelic.HermSys K} (hT : Presents T Nm L ev) {h : ℝ} (hh : T.IsHeight h) : IsImageHeight Nm L ev T.D h

theorem **Presents.imageDeg_eq** {T : Adelic.HermSys K} (hT : Presents T Nm L ev) {δ : ℕ} {f : MvPolynomial (Fin (D + 1) × Fin T.N) K} (hf : IsChowFormInt T.imageIdeal D δ f) : imageDeg ev D = δ

theorem **imageIdeal_pointOf** (K : Type) [Field K] {Ω : Type} [Field Ω] [Algebra K Ω] {N : ℕ} (y : Fin N → Ω) : imageIdeal (pointOf K y) (Pi.basisFun K (Fin N)) = coneIdeal K y

theorem **imageIdeal_std** (N : ℕ) : imageIdeal (stdPoint N) (Pi.basisFun ℚ (Fin N)) = ⊥

theorem **map_vdm** {R S : Type} [CommRing R] [CommRing S] (φ : R →+* S) (la : ι → Fin N → R) : φ (vdm la) = vdm fun τ i => φ (la τ i)

theorem **vdm_pos** {R : Type} [CommRing R] [LinearOrder R] [IsStrictOrderedRing R] {la : ι → Fin N → R} (h : ∀ τ, StrictAnti (la τ)) : 0 < vdm la

theorem **vdm_smul** {R : Type} [CommRing R] (t : R) (la : ι → Fin N → R) : vdm (t • la) = t ^ Formalism.DegreeFormula.dimFl ι N * vdm la

theorem **strictAnti_stdWt** (τ : ι) : StrictAnti (stdWt ι N τ)

theorem **weylProd_eq** (la : ι → Fin N → ℝ) : Formalism.DegreeFormula.weylProd la = vdm la / (vdm (stdWt ι N) : ℤ)

theorem **one_sub_rhoOf_mul** {u' u : ι → Fin N → ℤ} (h : vdm u ≠ 0) : (1 - rhoOf u' u) * (vdm u : ℤ) = (vdm u' : ℤ)

theorem **rhoOf_lt_one** {u' u : ι → Fin N → ℤ} (h' : ∀ τ, StrictAnti (u' τ)) (h : ∀ τ, StrictAnti (u τ)) : rhoOf u' u < 1

theorem **log_le_cStar** {n : ℕ} {abelian₂ : ℝ} (hn : 1 ≤ n) (hL : 1 ≤ abelian₂) : 0 ≤ Real.log abelian₂ ∧ Real.log abelian₂ ≤ cStar n abelian₂

theorem **logCard_coe** (m : ℕ) : logCard m = ENNReal.ofReal (Real.log m)

theorem **logCard_top** : logCard ⊤ = ⊤

theorem **countVol_mono** (D d a : ℕ) {m m' : ℕ → ℕ∞} (h : ∀ τ, m τ ≤ m' τ) : countVol D d a m ≤ countVol D d a m'

theorem **le_volSlope_iff** {δ : ℝ} (hδ : 0 < δ) (vol : ℝ → ℝ≥0∞) (a : ℝ) : (a : EReal) ≤ volSlope δ vol ↔ ∀ t : ℝ, ENNReal.ofReal (δ * (a + t)) ≤ vol t

theorem **IsLimSlope.unique** {D : ℕ} {ar geo : ℚ → ℝ} {a b : ℝ} (ha : IsLimSlope D ar geo a) (hb : IsLimSlope D ar geo b) : a = b

Development.AbelianVarietyTheory.Definitions.Divisors.AltPullApply

theorem **matAct_apply** (ρ : O →+* Module.End R V) {m : ℕ} (M : Matrix (Fin m) (Fin m) O) (x : Fin m → V) (i : Fin m) : matAct ρ M x i = ∑ j, ρ (M i j) (x j)

@[simp] theorem **altPull_apply** {W W' : Type} [AddCommGroup W] [Module R W] [AddCommGroup W'] [Module R W'] (u : W' →ₗ[R] W) (q : ℕ) (f : Alt R W E q) (x : Fin q → W') : altPull E u q f x = f fun i => u (x i)

theorem **altOp_comm** (ρ : O →+* Module.End R V) (σ : Representation R Γ V) (hσ : ∀ γ b, σ γ * ρ b = ρ b * σ γ) {m : ℕ} (ε : Matrix (Fin m) (Fin m) O →₀ E) (q : ℕ) (γ : Γ) : altRep E (powRep σ m) q γ * altOp ρ ε q = altOp ρ ε q * altRep E (powRep σ m) q γ

Development.AbelianVarietyTheory.Definitions.Divisors.AnalyticChartsRealizationCompPointEq

theorem **isDPMorphism_of_rat** {C C' : Type} [CommRing C] [CommRing C'] [Algebra ℚ C'] {J : Ideal C} {J' : Ideal C'} (δ : DividedPowers J) (δ' : DividedPowers J') (φ : C →+* C') (h : ∀ c ∈ J, φ c ∈ J') : IsDPMorphism δ δ' φ

theorem **isPDNilpotent_of_rat** {C : Type} [CommRing C] [Algebra ℚ C] {J : Ideal C} (δ : DividedPowers J) (h : IsNilpotent J) : IsNilDP δ

theorem **analyticChartsRealization_comp_point_eq** {T T' : K.Obj} (f : T ⟶ T') : spMap (Ideal.quotientMap T'.J (f.φ : T.C →+* T'.C) f.mem) ≫ spMap ((Ideal.Quotient.mk T.J).comp T.ρ) ≫ r.x T.k = spMap ((Ideal.Quotient.mk T'.J).comp T'.ρ) ≫ r.x T'.k

theorem **top_le_preimage** {Y : Scheme.{0}} {x : Sp ℂ ⟶ Y} {U : Y.Opens} {p : Sp ℂ} (h : x p ∈ U) : ⊤ ≤ x ⁻¹ᵁ U

theorem **isHolTo_affine** {V : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] {T : Scheme.{0}} [IsAffine T] {D : Set V} (hD : IsOpen D) {z : V → (Sp ℂ ⟶ T)} (hz : ∀ r : Γ(T, ⊤), DifferentiableOn ℂ (fun b ↦ valAt' (z b) ⊤ r) D) : IsHolTo D z

theorem **valAt'_spMap** {R : Type} [CommRing R] (ψ : R →+* ℂ) (t : R) : valAt' (spMap ψ) ⊤ ((Scheme.ΓSpecIso (CommRingCat.of R)).inv.hom t) = ψ t

theorem **perCl_injective** (F : Fam I g Λ) : Function.Injective F.allCharts.perCl

theorem **pt_const** {C : Type} [CommRing C] (σ : B →+* C) (e : C) (h : IsUnit e) : (pt σ e h).comp (algebraMap B (Ring B)) = σ

theorem **pt_coord** {C : Type} [CommRing C] (σ : B →+* C) (e : C) (h : IsUnit e) : pt σ e h (coord B) = e

theorem **comp_pt** {C C' : Type} [CommRing C] [CommRing C'] (φ : C →+* C') {σ : B →+* C} {e : C} (h : IsUnit e) {σ' : B →+* C'} {e' : C'} (h' : IsUnit e') (hσ : ∀ b, φ (σ b) = σ' b) (he : φ e = e') : φ.comp (pt σ e h) = pt σ' e' h'

theorem **analyticAt_fibre** (p : Tot I 1) : AnalyticAt ℂ (fun p : Tot I 1 ↦ p.2 0) p

theorem **isUnit_expFn** (W : Opens (Tot I 1)) : IsUnit (expFn W)

theorem **ev_comp_pt** (K : Charts (fam I)) (k : K.K) (p : Tot I 1) (hp : p ∈ K.W k) : (Hol.ev p hp).toRingHom.comp (pt ((K.cst k).comp σ) (expFn (K.W k)) (isUnit_expFn _)) = expPt σ p

theorem **isFor** (K : Charts (fam I)) : (real σ K).IsFor (unif σ)

theorem **based** : (unif σ).Based

theorem **unif_ne** (z : I → ℂ) : (unif σ).E (z, 0) ≠ (unif σ).E (z, fun _ ↦ Real.pi * Complex.I)

theorem **isPeriod_unique** {c : FirstDeRham (str B)} {per per' : ℤ →+ Hol (fam I).V} (h : IsPeriod (unif σ) c per) (h' : IsPeriod (unif σ) c per') : per = per'

theorem **isPeriod_zero_zero** {s : X ⟶ Sp B} {σ : B →+* Hol F.V} (u : Unif F s σ) : IsPeriod u 0 0

Development.AbelianVarietyTheory.Definitions.Divisors.AnalyticOnNhdFinChartIff

theorem **leviV_re_eq_levi** {h : V D → ℝ} {z : Fin D → ℂ} (hh : ContDiffAt ℝ 2 h (toV D z)) (ξ : Fin D → ℂ) : (leviV D (fun v => (h v : ℂ)) (toV D z) ξ).re = levi (h ∘ toV D) z ξ

theorem **levi_half** (h : (Fin D → ℂ) → ℝ) (z w : Fin D → ℂ) : levi (fun v => h v / 2) z w = levi h z w / 2

theorem **refl** (A : FinHolAtlas e X) (P : LocPot X) : SameForm A P P

theorem **symm** (h : SameForm A P Q) : SameForm A Q P

theorem **trans** (h : SameForm A P Q) (h' : SameForm A Q T) : SameForm A P T

theorem **of_re** (hQ : IsPotential A Q) (h : ∀ x, ∃ W : Set X, IsOpen W ∧ x ∈ W ∧ ∃ G : X → ℂ, A.HoloOn W G ∧ ∀ z ∈ W, P.φ x z = Q.φ x z + (G z).re) : SameForm A P Q

theorem **locDens_eq** {j : Fin A.n} {y : Fin e → ℂ} (hy : y ∈ (A.chart j).target) : locDens A Φ j y = locDens A Ψ j y

theorem **mInt_eq** (g : X → ℝ) : mInt A Φ g = mInt A Ψ g

theorem **mIntegrable_iff** (g : X → ℝ) : MIntegrable A Φ g ↔ MIntegrable A Ψ g

@[simp] theorem **finChart_source** : c.finChart.source = c.e.source

theorem **finChart_target** : c.finChart.target = toV D ⁻¹' c.e.target

@[simp] theorem **finChart_apply** (y : CxPts p σ) (a : Fin D) : c.finChart y a = c.e y a

theorem **toV_finChart** (y : CxPts p σ) : toV D (c.finChart y) = c.e y

theorem **finChart_symm_apply** (z : Fin D → ℂ) : c.finChart.symm z = c.e.symm (toV D z)

theorem **finChart_eq_ev** (hy : y ∈ c.e.source) (a : Fin D) : c.finChart y a = ev p σ c.U (c.x a) y

theorem **analyticOnNhd_finChart_iff** {E : Type*} [NormedAddCommGroup E] [NormedSpace ℂ E] [CompleteSpace E] (hS : IsOpen S) (f : CxPts p σ → E) : AnalyticOnNhd ℂ (f ∘ c.finChart.symm) (c.finChart.target ∩ c.finChart.symm ⁻¹' S) ↔ DifferentiableOn ℂ (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)

theorem **contDiffOn_finChart_iff** {E : Type*} [NormedAddCommGroup E] [NormedSpace ℝ E] (n : WithTop ℕ∞) (S : Set (CxPts p σ)) (f : CxPts p σ → E) : ContDiffOn ℝ n (f ∘ c.finChart.symm) (c.finChart.target ∩ c.finChart.symm ⁻¹' S) ↔ ContDiffOn ℝ n (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)

theorem **analyticOnNhd_trans** : AnalyticOnNhd ℂ (c'.finChart ∘ c.finChart.symm) (c.finChart.target ∩ c.finChart.symm ⁻¹' c'.finChart.source)

theorem **leviV_re_eq** {f : CxPts p σ → ℝ} (hf : ContDiffAt ℝ 2 (f ∘ c.e.symm) (c.e y)) (ξ : Fin D → ℂ) : (leviV D (fun v => (f (c.e.symm v) : ℂ)) (c.e y) ξ).re = levi (f ∘ c.finChart.symm) (c.finChart y) ξ

theorem **exists_levi_eq** (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) {f : CxPts p σ → ℝ} (hf : ContDiffAt ℝ 2 (f ∘ c'.finChart.symm) (c'.finChart y)) (w : Fin D → ℂ) : ∃ w', (w ≠ 0 → w' ≠ 0) ∧ levi (f ∘ c.finChart.symm) (c.finChart y) w = levi (f ∘ c'.finChart.symm) (c'.finChart y) w'

theorem **finAtlasOf_chart** (s : Finset R.ι) (hs) (j : Fin s.card) : (R.finAtlasOf s hs).chart j = (R.chart (s.equivFin.symm j)).finChart

theorem **isAlgAtlas_finAtlasOf** (s : Finset R.ι) (hs) : IsAlgAtlas (R.finAtlasOf s hs)

theorem **isAlgAtlas_finAtlas** [CompactSpace (CxPts p σ)] : IsAlgAtlas R.finAtlas

theorem **chart_eq** (j : Fin A.n) : A.chart j = (hA.anRead.chart j).finChart

theorem **isCompatChart** (c : AlgChart p σ D) : IsCompatChart A c.finChart

theorem **compatible** (hA' : IsAlgAtlas A') : A.Compatible A'

theorem **holoOn_iff** (R : AnRead p σ D) (hS : IsOpen S) {f : CxPts p σ → ℂ} : A.HoloOn S f ↔ R.HoloOn S f

theorem **smoothOn_iff** (R : AnRead p σ D) (hS : IsOpen S) {f : CxPts p σ → ℝ} : A.SmoothOn S f ↔ R.SmoothOn S f

theorem **smooth_iff** (R : AnRead p σ D) {f : CxPts p σ → ℝ} : A.Smooth f ↔ R.SmoothOn univ f

theorem **holoMap** {k' : Type} [CommRing k'] {Y' : Scheme.{0}} {p' : Y' ⟶ Sp k'} {σ' : k' →+* ℂ} {D' : ℕ} {A' : FinHolAtlas D' (CxPts p' σ')} (hA' : IsAlgAtlas A') {f : Y' ⟶ Y} {m : CxPts p' σ' → CxPts p σ} (hm : ∀ z, (m z).1 = z.1 ≫ f) : HoloMap A' A m

theorem **finAtlasOf_compatible** (s : Finset R.ι) (hs) (s' : Finset R'.ι) (hs') : (R.finAtlasOf s hs).Compatible (R'.finAtlasOf s' hs')

theorem **HoloOn.exists_log** {g : CxPts p σ → ℂ} (h : R.HoloOn S g) (hS : IsOpen S) {y : CxPts p σ} (hy : y ∈ S) (h0 : g y ≠ 0) : ∃ W, IsOpen W ∧ y ∈ W ∧ W ⊆ S ∧ ∃ G : CxPts p σ → ℂ, R.HoloOn W G ∧ ∀ z ∈ W, (G z).re = Real.log ‖g z‖

theorem **finAtlas_compatible** : R.finAtlas.Compatible R'.finAtlas

theorem **isCompatChart_finAtlas** (c : AlgChart p σ D) : IsCompatChart R.finAtlas c.finChart

theorem **finAtlas_holoOn_iff** (hS : IsOpen S) {f : CxPts p σ → ℂ} : R.finAtlas.HoloOn S f ↔ R.HoloOn S f

theorem **finAtlas_smoothOn_iff** (hS : IsOpen S) {f : CxPts p σ → ℝ} : R.finAtlas.SmoothOn S f ↔ R.SmoothOn S f

theorem **mInt_finAtlas_eq** [T2Space (CxPts p σ)] {Φ : Fin D → LocPot (CxPts p σ)} (hΦ : ∀ i, IsPotential R.finAtlas (Φ i)) {g : CxPts p σ → ℝ} (hg : MIntegrable R.finAtlas Φ g) : MIntegrable R'.finAtlas Φ g ∧ mInt R.finAtlas Φ g = mInt R'.finAtlas Φ g

theorem **locPot_U** (x : CxPts p σ) : φ.locPot.U x = ptsIn p σ (B.U (pick φ.isLB x))

theorem **locPot_φ** (x y : CxPts p σ) : φ.locPot.φ x y = φ.φ (pick φ.isLB x) y / 2

theorem **locSec_apply** (s : B.ι → CxPts p σ → ℂ) (x : CxPts p σ) : φ.locSec s x = s (pick φ.isLB x)

theorem **half_eq** {i j : B.ι} {y : CxPts p σ} (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) : φ.φ i y / 2 = φ.φ j y / 2 + Real.log ‖gv p σ B i j y‖

theorem **exists_half_eq** {i j : B.ι} {y : CxPts p σ} (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) : ∃ W, IsOpen W ∧ y ∈ W ∧ ∃ G : CxPts p σ → ℂ, R.HoloOn W G ∧ ∀ z ∈ W, φ.φ i z / 2 = φ.φ j z / 2 + (G z).re

theorem **IsSmooth.isPotential** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) : IsPotential A φ.locPot

theorem **isSmooth_of_isPotential** (hA : IsAlgAtlas A) (h : IsPotential A φ.locPot) : φ.IsSmooth R

theorem **norm_mul_exp_eq_sqrt** {s : B.ι → CxPts p σ → ℂ} (hs : s ∈ fibSec p σ B) {i : B.ι} {y : CxPts p σ} (hi : y ∈ ptsIn p σ (B.U i)) : ‖s i y‖ * Real.exp (-(φ.φ i y / 2)) = √(φ.nsq s y)

theorem **isSection_of_isAlgAtlas** (hA : IsAlgAtlas A) {s : B.ι → CxPts p σ → ℂ} (hs : s ∈ R.holSec B) : IsSection A φ.locPot (φ.locSec s)

theorem **logNormSec_locSec** (s : B.ι → CxPts p σ → ℂ) (y : CxPts p σ) : logNormSec φ.locPot (φ.locSec s) y = 1 / 2 * Real.log (φ.nsq s y)

theorem **exp_logNormSec_locSec** {s : B.ι → CxPts p σ → ℂ} {y : CxPts p σ} (h : φ.nsq s y ≠ 0) : Real.exp (2 * logNormSec φ.locPot (φ.locSec s) y) = φ.nsq s y

theorem **IsSmooth.contDiffAt_finChart** (h : φ.IsSmooth R) (c : AlgChart p σ D) {i : B.ι} {y : CxPts p σ} (hy : y ∈ c.e.source) (hi : y ∈ ptsIn p σ (B.U i)) : ContDiffAt ℝ 2 (φ.φ i ∘ c.finChart.symm) (c.finChart y)

theorem **IsSmooth.exists_levi_eq** (h : φ.IsSmooth R) (c c' : AlgChart p σ D) {i j : B.ι} {y : CxPts p σ} (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) (w : Fin D → ℂ) : ∃ w', (w ≠ 0 → w' ≠ 0) ∧ levi (φ.φ i ∘ c.finChart.symm) (c.finChart y) w = levi (φ.φ j ∘ c'.finChart.symm) (c'.finChart y) w'

theorem **IsSmooth.leviV_re_eq** (h : φ.IsSmooth R) (c : AlgChart p σ D) {i : B.ι} {y : CxPts p σ} (hy : y ∈ c.e.source) (hi : y ∈ ptsIn p σ (B.U i)) (ξ : Fin D → ℂ) : (leviV D (fun v => (φ.φ i (c.e.symm v) : ℂ)) (c.e y) ξ).re = levi (φ.φ i ∘ c.finChart.symm) (c.finChart y) ξ

theorem **IsSmooth.levi_locPot** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) {x : CxPts p σ} {j : Fin A.n} {z : Fin D → ℂ} (hz : z ∈ (A.chart j).target ∩ (A.chart j).symm ⁻¹' φ.locPot.U x) (w : Fin D → ℂ) : ∃ (i : B.ι) (c : R.ι), ∃ y ∈ (R.chart c).e.source ∩ ptsIn p σ (B.U i), ∃ ξ, (w ≠ 0 → ξ ≠ 0) ∧ levi (φ.locPot.φ x ∘ (A.chart j).symm) z w = (leviV D (fun v => (φ.φ i ((R.chart c).e.symm v) : ℂ)) ((R.chart c).e y) ξ).re / 2

theorem **IsSmooth.leviV_eq_locPot** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) {i : B.ι} {c : AlgChart p σ D} {y : CxPts p σ} (hy : y ∈ c.e.source ∩ ptsIn p σ (B.U i)) (ξ : Fin D → ℂ) : ∃ j, ∃ z ∈ (A.chart j).target ∩ (A.chart j).symm ⁻¹' φ.locPot.U y, ∃ w, (ξ ≠ 0 → w ≠ 0) ∧ (leviV D (fun v => (φ.φ i (c.e.symm v) : ℂ)) (c.e y) ξ).re = 2 * levi (φ.locPot.φ y ∘ (A.chart j).symm) z w

theorem **IsSmooth.isPsh_iff** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) : IsPsh A φ.locPot ↔ φ.SemiPos R

theorem **IsSmooth.isPosPot_iff** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) : IsPosPot A φ.locPot ↔ φ.StrictPos R

theorem **sameForm_of_re** (hA : IsAlgAtlas A) {Q : LocPot (CxPts p σ)} (hQ : IsPotential A Q) (h : ∀ x, ∃ i, x ∈ ptsIn p σ (B.U i) ∧ ∃ W, IsOpen W ∧ x ∈ W ∧ ∃ G : CxPts p σ → ℂ, R.HoloOn W G ∧ ∀ z ∈ W, φ.φ i z / 2 = Q.φ x z + (G z).re) : SameForm A φ.locPot Q

theorem **sameForm_of_eq** (hA : IsAlgAtlas A) {Q : LocPot (CxPts p σ)} (hQ : IsPotential A Q) (h : ∀ x, ∃ i, x ∈ ptsIn p σ (B.U i) ∧ ∀ z, φ.φ i z / 2 = Q.φ x z) : SameForm A φ.locPot Q

theorem **IsSmooth.sameForm_tens** {ψ : Weight p σ B'} (h' : ψ.IsSmooth R) (h : φ.IsSmooth R) (hA : IsAlgAtlas A) : SameForm A (ψ.tens φ).locPot (ψ.locPot + φ.locPot)

theorem **IsSmooth.sameForm_zpow** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) (n : ℤ) : SameForm A (φ.zpow n).locPot (φ.locPot.zsmul n)

theorem **IsSmooth.sameForm_dual** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) : SameForm A φ.dual.locPot (φ.locPot.zsmul (-1))

theorem **sameForm_addFn** (A : FinHolAtlas D (CxPts p σ)) (u : CxPts p σ → ℝ) : SameForm A (φ.addFn u).locPot (φ.locPot + LocPot.ofFun fun y => u y / 2)

theorem **sameForm_ofFun** (A : FinHolAtlas D (CxPts p σ)) (u : CxPts p σ → ℝ) : SameForm A (ofFun u).locPot (LocPot.ofFun fun y => u y / 2)

theorem **IsSmooth.sameForm_ofIso** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) (hB' : B'.IsLB) (w : LB.Iso B B') (hw : w.IsGen) : SameForm A (φ.ofIso hB' w hw).locPot φ.locPot

theorem **IsSmooth.sameForm_pull** (h : φ.IsSmooth R) (hA : IsAlgAtlas A) {Y' : Scheme.{0}} {q : Y' ⟶ Sp k} {D' : ℕ} {A' : FinHolAtlas D' (CxPts q σ)} (hA' : IsAlgAtlas A') (f : Y' ⟶ Y) (hf : f ≫ p = q) : SameForm A' (φ.pull f hf).locPot (φ.locPot.pullback (PtsOver.map f hf _) (continuous_of_comp (f

theorem **locDens_locPot** (hA : IsAlgAtlas A) {L : Fin D → LB Y} {χ : ∀ a, Weight p σ (L a)} (h : ∀ a, (χ a).IsSmooth R) {j : Fin A.n} {z : Fin D → ℂ} (hz : z ∈ (A.chart j).target) {i : ∀ a, (L a).ι} (hi : ∀ a, (A.chart j).symm z ∈ ptsIn p σ ((L a).U (i a))) : locDens A (fun a => (χ a).locPot) j z = maDens (fun a v => (χ a).φ (i a) ((A.chart j).symm v) / 2) z

theorem **isPotential_locPot** (h : φ.IsSmooth R) : IsPotential R.finAtlas φ.locPot

theorem **isPotential_locPot_iff** : IsPotential R.finAtlas φ.locPot ↔ φ.IsSmooth R

theorem **isSection_locSec** {s : B.ι → CxPts p σ → ℂ} (hs : s ∈ R.holSec B) : IsSection R.finAtlas φ.locPot (φ.locSec s)

theorem **isPsh_iff_semiPos** (h : φ.IsSmooth R) : IsPsh R.finAtlas φ.locPot ↔ φ.SemiPos R

theorem **isPosPot_iff_strictPos** (h : φ.IsSmooth R) : IsPosPot R.finAtlas φ.locPot ↔ φ.StrictPos R

theorem **mInt_locPot_eq** [T2Space (CxPts p σ)] {L : Fin D → LB Y} {χ : ∀ a, Weight p σ (L a)} (h : ∀ a, (χ a).IsSmooth R) (R' : AnRead p σ D) {g : CxPts p σ → ℝ} (hg : Continuous g) : MIntegrable R.finAtlas (fun a => (χ a).locPot) g ∧ mInt R.finAtlas (fun a => (χ a).locPot) g = mInt R'.finAtlas (fun a => (χ a).locPot) g

Development.AbelianVarietyTheory.Definitions.Divisors.AnisoClauseOfFirstTerm

theorem **firstTerm_iff** : GlobalTheory.Published.FirstTermIdentityResidues ↔ AnisoClause ∧ IsoClause

theorem **anisoClause_of_firstTerm** (h : GlobalTheory.Published.FirstTermIdentityResidues) : AnisoClause

theorem **isoClause_of_firstTerm** (h : GlobalTheory.Published.FirstTermIdentityResidues) : IsoClause

theorem **firstTerm_of_clauses** (ha : AnisoClause) (hi : IsoClause) : GlobalTheory.Published.FirstTermIdentityResidues

Development.AbelianVarietyTheory.Definitions.Divisors.AxesVanishingStructureSheafDegOne

theorem **pairRes_pull2_bijective** (φ : X' ⟶ X) (s' : X' ⟶ Sp R') : Function.Bijective (pairRes φ s').pull2

theorem **ovw** (f : Y ⟶ A) : f.left ≫ A.hom = Y.hom ≫ spMap (RingHom.id R)

Development.AbelianVarietyTheory.Definitions.Divisors.BadElt

theorem **mem_eig** {K Γ V : Type} [Field K] [Group Γ] [AddCommGroup V] [Module K V] (ρ : Representation K Γ V) (H : Subgroup Γ) (lam : H →* Kˣ) (v : V) : v ∈ eig ρ H lam ↔ ∀ h : H, ρ h v = ((lam h : Kˣ) : K) • v

Development.AbelianVarietyTheory.Definitions.Divisors.BarCompSubsetClosure

theorem **LatFrame.mem_span_iff** {F V : Type} [Field F] [NumberField F] [AddCommGroup V] [Module F V] [Module (𝒪 F) V] [IsScalarTower (𝒪 F) F V] {Nm : (F →+* ℂ) → V → ℝ} {M : Submodule (𝒪 F) V} (f : LatFrame Nm M) (x : V) : f.b.equivFun x ∈ Submodule.span (𝒪 F) (Set.range f.w.col) ↔ x ∈ M

theorem **gap_eq** (v : Fin N → ℤ) {i j : Fin N} (h : (j : ℕ) = i + 1) : gap v i = v i - v j

theorem **ofGaps_anchor** (g : ℕ → ℤ) (i : Fin N) : ofGaps i g i = 0

theorem **gap_ofGaps** (a : ℕ) (g : ℕ → ℤ) {k : ℕ} (h : k + 1 < N) : gap (ofGaps a g : Fin N → ℤ) k = g k

theorem **ofGaps_gap** (v : Fin N → ℤ) (a i : Fin N) : ofGaps a (gap v) i = v i - v a

theorem **floorGaps_add_remGaps** (n : ℤ) (v : Fin N → ℤ) (a i : Fin N) : n * floorGaps a n v i + remGaps a n v i = v i - v a

theorem **gap_remGaps** (a : ℕ) {n : ℤ} (hn : 0 < n) (v : Fin N → ℤ) (k : ℕ) : 0 ≤ gap (remGaps a n v) k ∧ gap (remGaps a n v) k < n

theorem **kap_add** (j : ℕ) : FlagTan.kap Dw + (P.u₂ + dimFlag Dw • P.u₁ + j • P.u₁) = P.wt j

theorem **u₁_add_wt** (j : ℕ) : P.u₁ + P.wt j = P.wt (j + 1)

theorem **gap_u₁** (τ : Psi H) {k : ℕ} (h : k + 1 < Dw.p + Dw.q) : gap (P.u₁ τ) k = gap ((P.uMinus - FlagTan.kap Dw) τ) k / P.T₁

theorem **gap_rem** (τ : Psi H) (k : ℕ) : 0 ≤ gap (P.rem τ) k ∧ gap (P.rem τ) k ≤ P.T + dimFlag Dw

theorem **gap_ub** (τ : Psi H) {k : ℕ} (h : k + 1 < Dw.p + Dw.q) : gap (P.ub τ) k = gap (P.u₁ τ) k / (dimFlag Dw + 3)

theorem **anchored** (τ : Psi H) {i : Fin (Dw.p + Dw.q)} (h : (i : ℕ) = P.a) : P.u₁ τ i = 0 ∧ P.rem τ i = 0 ∧ P.u₂ τ i = 0 ∧ P.ub τ i = 0

theorem **uMinus_eq** (τ : Psi H) {a : Fin (Dw.p + Dw.q)} (h : (a : ℕ) = P.a) (i : Fin (Dw.p + Dw.q)) : P.uMinus τ i = P.wt P.T τ i + (P.uMinus τ a - FlagTan.kap Dw τ a)

theorem **gap_uMinus_sub_kap** (τ : Psi H) {k : ℕ} (h : k + 1 < Dw.p + Dw.q) : gap ((P.uMinus - FlagTan.kap Dw) τ) k = gap (P.u τ) k + if k + 1 = Dw.p then -P.c₀ - (2 * ((Dw.p + Dw.q : ℕ) : ℤ) - 2) else 2

theorem **dimFlag_eq** (Dw : Datum H) : CompHt.dimFlag Dw = dimFlag Dw

theorem **opensRange_inc** : X.inc.opensRange = flagOpen X

theorem **top_le_inc** : ⊤ ≤ X.inc ⁻¹ᵁ flagOpen X

theorem **dense_flagOpen** : Dense (flagOpen X : Set X.YY)

theorem **ptY_str** {S : Type} [CommRing S] [Algebra c.F S] (P : FrPt Xo c S) : X.ptY P ≫ flagStr X = spMap (algebraMap c.F S)

theorem **coef_toSecO** {S : Type} [CommRing S] (m : H0 (flagStr X) (X.LL e) (flagOpen X)) (k : X.F.J) (w : Sp S ⟶ X.Yo) (hz : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k) (hO : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ flagOpen X) : FlagCover.coef (toSecO m) k w hz = H0.val (B

theorem **bijective_toSecO** : Function.Bijective (toSecO : H0 (flagStr X) (X.LL e) (flagOpen X) → X.SecO e)

theorem **bijective_ruleOf** : Function.Bijective (ruleOf X e)

theorem **evR_eq_val_secOf** {S : Type} [CommRing S] [Algebra c.F S] [_root_.IsReduced S] (f : secZ Xo c e) {P Q : FrPt Xo c S} {k : X.F.J} {z : Sp S ⟶ X.YY} {hz : ⊤ ≤ z ⁻¹ᵁ X.F.U k} (hO : ⊤ ≤ z ⁻¹ᵁ flagOpen X) {t : Ix Dw → S×} (hQ : X.F.ChartFr k z hz Q) (ht : FrBor t Q P) : evR Xo c f.1 P = (∏ ix, t ix ^ e ix).val * H0.val (B

theorem **exists_chartFr** {S : Type} [CommRing S] (z : Sp S ⟶ X.YY) (hO : ⊤ ≤ z ⁻¹ᵁ flagOpen X) : ∃ _ : Algebra c.F S, ∀ (k : X.F.J) (hz : ⊤ ≤ z ⁻¹ᵁ X.F.U k), ∃ Q : FrPt Xo c S, X.F.ChartFr k z hz Q

theorem **secOf_mul** {e₁ e₂ : Ix Dw → ℤ} (h : e₁ + e₂ = e) (f₁ : secZ Xo c e₁) (f₂ : secZ Xo c e₂) (f : secZ Xo c e) (hf : f.1 = f₁.1 * f₂.1) : secOf X e f = ((commonLL X).addChart h).mul (secOf X e₁ f₁) (secOf X e₂ f₂)

theorem **j_mem_barComp** (x : Xo.XF c) : Y.j x ∈ barComp Y Z ↔ x ∈ (Z : Set (Xo.XF c))

theorem **barComp_subset_closure** : (barComp Y Z : Set Y.X) ⊆ closure (Y.j '' (Z : Set (Xo.XF c)))

theorem **dense_U₀** : Dense (U₀ X Z : Set (Y₀ X Z))

theorem **ptY_le_iff** {S : Type} [CommRing S] [Algebra c.F S] (P : FrPt Xo c S) : ⊤ ≤ X.ptY P ⁻¹ᵁ (X.π ⁻¹ᵁ barComp Y Z) ↔ LiesOn Z P

theorem **L₀_eq** (la : AllWt.Weight Dw) : L₀ X Z la = (X.Llam la).pull (inc₀ X Z)

theorem **exOf_add_zero** (a b : AllWt.Weight Dw) : exOf Dw (a + b) 0 = exOf Dw a 0 + exOf Dw b 0

theorem **isLB_L₀** (la : AllWt.Weight Dw) : (L₀ X Z la).IsLB

theorem **eq_zero_of_sec₀** {e : Ix Dw → ℤ} {f : secZ Xo c e} (hf : f.1 ∈ suppOn Z) (h : sec₀ X Z e f = 0) : f = 0

theorem **sec₀_mul** {e₁ e₂ e : Ix Dw → ℤ} (h : e₁ + e₂ = e) (f₁ : secZ Xo c e₁) (f₂ : secZ Xo c e₂) (f : secZ Xo c e) (hf : f.1 = f₁.1 * f₂.1) : sec₀ X Z e f = (((commonLL X).addChart h).pull (inc₀ X Z)).mul (sec₀ X Z e₁ f₁) (sec₀ X Z e₂ f₂)

theorem **mem_vanBd** {la : AllWt.Weight Dw} {s : H0 (str₀ X Z) (L₀ X Z la) ⊤} : s ∈ vanBd X Z la ↔ ∀ y ∈ bd₀ X Z, y ∉ nonzeroLocus (toSec s)

theorem **cusp_le_secZ** (la : AllWt.Weight Dw) : Cusp Z U la ≤ secZ Xo c (exOf Dw la 0)

theorem **cuspSec_injective** (la : AllWt.Weight Dw) : Function.Injective (cuspSec X Z U la)

theorem **cuspSec_ofSpace** (la : AllWt.Weight Dw) (s : space X Z U la) : cuspSec X Z U la (ofSpace X Z U la s) = restr le_top s.1

theorem **ofSpace_injective** (la : AllWt.Weight Dw) : Function.Injective (ofSpace X Z U la)

theorem **Extends.restr_equiv** (h : Extends X Z U la) (f : Cusp Z U la) : restr le_top (h.equiv f).1 = cuspSec X Z U la f

theorem **Extends.space_le** (h : Extends X Z U la) : space X Z U la ≤ vanBd X Z la

theorem **Extends.toVanBd_injective** (h : Extends X Z U la) : Function.Injective h.toVanBd

Development.AbelianVarietyTheory.Definitions.Divisors.BaseIdealIdeal

theorem **baseIdeal_ideal** (f : W ⟶ Spec (CommRingCat.of R)) (J : Ideal R) (U : W.affineOpens) : (baseIdeal f J).ideal U = J.map (cst f U.1)

theorem **IsSystem.baseIdeal_le** {f : W ⟶ Spec (CommRingCat.of R)} {I : Ideal R} {t : ℕ → W.IdealSheafData} (h : IsSystem f I t) (e : ℕ) : baseIdeal f (I ^ (e + 1)) ≤ t e

Development.AbelianVarietyTheory.Definitions.Divisors.BasePlaceNotMemChiGridBadPlaces

theorem **isCartanInv_neg** {V : HSp D} {τ : D.what₂.Tau} {s : Matrix (Fin V.rank) (Fin V.rank) ℂ} (h : IsCartanInv (gramAt D V τ) s) : IsCartanInv (gramAt D (negSp V) τ) (-s)

theorem **mem_S0_of_sum_eq** {v : Pl D.host.F} : v ∈ P.S0 ↔ v = P.v0 ∨ v ∈ S1 D

theorem **under_w** {v : Pl D.host.F} (h : v ∈ P.S0) : (P.w v).under (𝒪 D.host.F) = v

theorem **card_chosenPlaces_eq** : P.W0.card = P.S0.card

theorem **depth_pos** (sc : Consts) (v : Pl D.host.F) : 0 < P.depth sc v

theorem **modul_ne_bot** (sc : Consts) : P.modul sc ≠ ⊥

theorem **stairId_st** (sc : Consts) (c : ℕ) (p : Fin c) : stairId (P.stA sc) (P.stAbar sc) c p = stairId D.host.AA D.host.AAbar c p * ((P.modul sc : Ideal (𝒪 D.host.E)) : FractionalIdeal (𝒪 D.host.E)⁰ D.host.E) ^ (p.1 + 1) * (((P.modulBar sc : Ideal (𝒪 D.host.E)) : FractionalIdeal (𝒪 D.host.E)⁰ D.host.E) ^ (c - p.1))⁻¹

theorem **Chars.basePlace_not_mem_chiGridBadPlaces** {L : LevCh D T} {P : Places L} {C : ℝ} (X : Chars P C) : P.v0 ∉ S1 D

theorem **colY_lt** (L : LevCh D T) {j : ℕ} (h : j < D.R.m) : columnBlock L j = ⟨Yc D j, L.ΛY j⟩

theorem **colY_m** (L : LevCh D T) : columnBlock L D.R.m = ⟨D.sp.X', L.ΛX⟩

theorem **InRange.small** {m : ℕ} {p : Pair T} (h : p.InRange m) : p.small.InRange m

theorem **rank_big** (p : Pair T) : p.big.rank D = p.small.rank D + p.len D

theorem **src_tgt** (a : Arrow T) : (a.src = a.pair.small ∧ a.tgt = a.pair.big) ∨ (a.src = a.pair.big ∧ a.tgt = a.pair.small)

theorem **node_zero_zero** : G.node (.grid 0 0) = ⟨Yc D 0, L.ΛY 0⟩

theorem **node_top_zero** : G.node (.top 0) = G.node (.core D.R.m)

theorem **isV** {X : Node} (h : X.InRange D.R.m) (h0 : ∀ j, X ≠ .grid 0 j) : (G.node X).IsV (X.rank D) (X.neg D.R.m)

theorem **levCong_le** (X : Node) : G.levCong X ≤ G.levMax X

theorem **chr_core** {i : ℕ} (h : i + 1 ≤ D.R.m) : G.chrS (.core i) = G.ch.chi (D.R.m - i) ∧ G.chrT (.core i) = G.ch.chi (D.R.m - i)

theorem **chr_mu_zero** (t : T) {j : ℕ} (h : j + 1 ≤ D.R.m) : (G.chrS (.mu t 0 j), G.chrT (.mu t 0 j)) = Formalism.IndefR.Lift.chrStep (L.mu t) L.eta (j + 1)

theorem **finite_datum** (h : (L.ΛY 0).FG) : Finite G.Datum

theorem **kerA_coreUp** (i : ℕ) (f : WO (G.pS (.core i)) (G.pT (.core i)) → ℂ) (g : GA D (G.V (.core i))) (h : GA D (G.V (.core (i + 1)))) : W.kerA (.coreUp i) f g h = star (W.kerA (.coreDown i) f h g)

Development.AbelianVarietyTheory.Definitions.Divisors.BaseRankDimension

theorem **ev_of_atAll** {𝒜 : FamClass} {σ : RawSel Bd 𝒜} {q : BdP Bd} (h : AtAll q 𝒜 σ) {eps : ℝ} (Fm : FamP eps) (hc : 𝒜 Fm) : Ev σ Fm hc q

Development.AbelianVarietyTheory.Definitions.Divisors.BlockGLMap

theorem **blockGL_val** (a b : ℕ) (A : Type) [CommRing A] (g : GL (Fin a) A) (h : GL (Fin b) A) : Units.val (blockGL a b A g h) = Matrix.reindex finSumFinEquiv finSumFinEquiv (Matrix.fromBlocks g.val 0 0 h.val)

theorem **_root_.GlobalTheory.EtC.blockGL_map** (a b : ℕ) {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (g : GL (Fin a) A) (h : GL (Fin b) A) : _root_.GlobalTheory.EtC.blockGL a b B (Matrix.GeneralLinearGroup.map f g) (Matrix.GeneralLinearGroup.map f h) = Matrix.GeneralLinearGroup.map f (_root_.GlobalTheory.EtC.blockGL a b A g h)

Development.AbelianVarietyTheory.Definitions.Divisors.BlockPairCycPow

theorem **crysWith_cyc_pow** (m : ℕ) : CrysWith (rk1 (cyc E Ω l ^ m)) fun _ => {-(m : ℤ)}

theorem **blockPair_cyc_pow** (m : ℕ) : BlockPair E Ω l 1 1 (fun _ _ => -(m : ℤ)) fun _ => -(m : ℤ)

Development.AbelianVarietyTheory.Definitions.Divisors.BlockSphereMeanLinForm

theorem **isChowForm_point** (k : Type) [Field k] : IsChowForm (Ideal.span {(X 1 : MvPolynomial (Fin 2) k)}) 0 1 (X (0, 0))

theorem **linSubst_linForm** {R : Type} [CommRing R] {N n : ℕ} (M : Matrix (Fin N) (Fin n) R) (c : Fin N → R) : linSubst M (linForm c) = linForm (fun k => ∑ j, M j k * c j)

theorem **_root_.GlobalTheory.ChowHt.Model.coeff_linForm** {R : Type} [CommRing R] {N : ℕ} (c : Fin N → R) (k : Fin N) : (_root_.GlobalTheory.ChowHt.Model.linForm c).coeff (Finsupp.single (0, k) 1) = c k

theorem **gaussAbs_linForm** {F : Type} [Field F] [NumberField F] {N : ℕ} (v : FinitePlace F) (c : Fin N → F) : ReciprocityTheory.MultiplicativeHeights.gaussAbs v (linForm c) = ⨆ k, v (c k)

theorem **blockSphereMean_linForm** {N : ℕ} (c : Fin N → ℂ) (hc : c ≠ 0) : ReciprocityTheory.MultiplicativeHeights.blockSphereMean 1 N (linForm c) = Real.log (Real.sqrt (∑ i, ‖c i‖ ^ 2)) - (harmonic (N - 1) : ℝ) / 2

theorem **formHeight_linForm** {F : Type} [Field F] [NumberField F] {N n : ℕ} (w : Matrix (Fin N) (Fin n) F) (A : (F →+* ℂ) → Matrix (Fin N) (Fin N) ℂ) (hA : ∀ σ, (A σ).det ≠ 0) (y : Fin N → F) (hy : y ≠ 0) : formHeight 1 w A (linForm y) = pointHeight w A y

theorem **isChowForm_line** (k : Type) [Field k] : IsChowForm (⊥ : Ideal (MvPolynomial (Fin 2) k)) 1 1 (Matrix.of fun i j : Fin 2 => (X (i, j) : MvPolynomial (Fin 2 × Fin 2) k)).det

theorem **not_isChowForm_top** {k : Type} [Field k] {N D δ : ℕ} (f : MvPolynomial (Fin (D + 1) × Fin N) k) : ¬ IsChowForm (⊤ : Ideal (MvPolynomial (Fin N) k)) D δ f

Development.AbelianVarietyTheory.Definitions.Divisors.BplusOfEquivariant

theorem **levCol_lt** (L : LevCh D T) (t : T) {j : ℕ} (h : j + 1 < D.R.m) : HEq (levCol L t j) (Formalism.L2.DIV.lev D.sp.cm (Yc D (j + 1)).gram (L.βY (j + 1)) (L.datYt t (j + 1) L.triv))

theorem **levCol_top** (L : LevCh D T) (t : T) {j : ℕ} (h : ¬ j + 1 < D.R.m) : HEq (levCol L t j) (Formalism.L2.DIV.lev D.sp.cm D.sp.X'.gram L.βX (L.datXt t))

theorem **levJ_row** (G : GridData L sc) (t : T) (i : ℕ) : levJ G t (.grid i 0) = G.levCong (.grid i 0)

theorem **Vol.IsAt.haar_fund** {M : Vol G} {X : Node} {K : Subgroup (GL (Fin (G.V X).rank) (Lev.EAf D))} (h : M.IsAt X K) : (M.ν X).IsHaarMeasure ∧ IsFundamentalDomain (GQ (G.V X)) (M.fd X) (M.ν X)

theorem **Vol.Is.isRow** {M : Vol G} {t : T} (h : M.Is t) : M.IsRow

theorem **node_isV** (G : GridData L sc) {μ0 : T} (hL : L.Is μ0) {X : Node} (h : X.InRange D.R.m) : (G.node X).IsV (X.rank D) (X.neg D.R.m)

theorem **levJ_open_compact** (G : GridData L sc) {μ0 : T} (hL : L.Is μ0) (t : T) {X : Node} (h : X.InRange D.R.m) : IsOpen (((levJ G t X).comap (Lev.finE (G.V X)) : Subgroup (GA D (G.V X))) : Set (GA D (G.V X))) ∧ IsCompact ((Lev.levA (G.V X) (levJ G t X) : Subgroup (GA D (G.V X))) : Set (GA D (G.V X)))

theorem **exists_vol** (G : GridData L sc) {μ0 : T} (hL : L.Is μ0) (t : T) : ∃ M : Vol G, M.Is t

theorem **liftA_apply** (a : Arrow T) (f : Sch G a) (F : Fn G a.src) (h : GA D (G.V a.tgt)) : liftA W M a f F h = ∫ g in M.fd a.src, W.kerA a f g h * F g ∂M.ν a.src

theorem **liftA_smul** (a : Arrow T) (f : Sch G a) (c : ℂ) (F : Fn G a.src) : liftA W M a f (c • F) = c • liftA W M a f F

theorem **liftA_add** {a : Arrow T} {f : Sch G a} {F F' : Fn G a.src} (h : LiftConv W M a f F) (h' : LiftConv W M a f F') : liftA W M a f (F + F') = liftA W M a f F + liftA W M a f F'

theorem **pathLift_comp** (p : Path T X Y) (d : PureT G p) (F : Fn G X) : ∀ {Z : Node} (q : Path T Y Z) (e : PureT G q), liftP W M (p.comp q) (d.append e) F = liftP W M q e (liftP W M p d F) | _, .nil, _ => rfl | _, .snoc a q, e => congrArg (liftA W M a e.2) (pathLift_comp p d F q e.1) theorem pathConv_comp (p : Path T X Y) (d : PureT G p) (F : Fn G X) : ∀ {Z : Node} (q : Path T Y Z) (e : PureT G q), PathConv W M (p.comp q) (d.append e) F ↔ PathConv W M p d F ∧ PathConv W M q e (liftP W M p d F) | _, .nil, _ => (and_iff_left trivial).symm | _, .snoc a q, e => by show PathConv W M (p.comp q) (d.append e.1) F ∧ LiftConv W M a e.2 (liftP W M (p.comp q) (d.append e.1) F) ↔ _ rw [pathConv_comp p d F q e.1, pathLift_comp, and_assoc] rfl theorem pathLift_smul (c : ℂ) (F : Fn G X) : ∀ {Y : Node} (p : Path T X Y) (d : PureT G p), liftP W M p d (c • F) = c • liftP W M p d F | _, .nil, _ => rfl | _,

theorem **liftT_add** (p : Path T X Y) (τ τ' : TotT G p) (F : Fn G X) : liftT W M p (τ + τ') F = liftT W M p τ F + liftT W M p τ' F

theorem **embRel_iff** {A B : HSp D} {c : ℕ} {ε : D.host.E} (d : HermitianOrthogonalExtensionData A B c ε) (h : GA D A) (g : GA D B) : EmbRel B A d.ι g h ↔ g = stabilizerEmbedding d h

theorem **besInt_eq** (φ : Fn G (.top 0)) (F : Fn G (.grid D.R.m 0)) : besInt W φ F = GlobalTheory.Per.besFun W.ψ G.besTop (embTop G) (star φ) F

theorem **Bplus_eq** (φ : Fn G (.top 0)) (F : Fn G (.grid D.R.m 0)) : Bplus W M φ F = GlobalTheory.Per.besPer W.ψ G.besTop (M.ν (.top 0)) (M.fd (.top 0)) M.νN M.fdN (embTop G) (star φ) F

theorem **Bplus_smul** (c c' : ℂ) (φ : Fn G (.top 0)) (F : Fn G (.grid D.R.m 0)) : Bplus W M (c • φ) (c' • F) = c * c' * Bplus W M φ F

theorem **Bplus_of_equivariant** (hM : M.νN M.fdN = 1) (φ : Fn G (.top 0)) (F : Fn G (.grid D.R.m 0)) (hψ : ∀ n ∈ NFlag _ G.besTop.e, ‖besChar W.ψ G.besTop n‖ = 1) (hF : ∀ n ∈ NFlag _ G.besTop.e, ∀ g, F (n * g) = besChar W.ψ G.besTop n * F g) : Bplus W M φ F = ∫ h in M.fd (.top 0), φ h * F (embTop G h) ∂M.ν (.top 0)

theorem **pairVec_apply** {V : HSp D} (c : CoefA V) (ξ : Module.Dual ℂ c.W) (Ψ : Bridge.Gf V → c.W) (h : GA D V) : pairVec c ξ Ψ h = ξ (c.ρ h⁻¹ (Ψ ((finHom V).rangeRestrict h)))

theorem **fPsiPhi_smul** (ρ : G.RealVec) {c : CoefA (G.V (.top 0))} (ξ : Module.Dual ℂ c.W) (a b : ℂ) (Ψ : Bridge.Gf (G.V (.top 0)) → c.W) (Φ : G.Datum) (u : GA D (Yc D 0) → ℂ) : fPsiPhi W M ρ ξ (a • Ψ) Φ (b • u) = a * b * fPsiPhi W M ρ ξ Ψ Φ u

theorem **fPsiPhi_add** {ρ : G.RealVec} {c : CoefA (G.V (.top 0))} {ξ : Module.Dual ℂ c.W} {Ψ : Bridge.Gf (G.V (.top 0)) → c.W} {Φ : G.Datum} {u u' : GA D (Yc D 0) → ℂ} (h : FConv W M ρ ξ Ψ Φ u) (h' : FConv W M ρ ξ Ψ Φ u') : fPsiPhi W M ρ ξ Ψ Φ (u + u') = fPsiPhi W M ρ ξ Ψ Φ u + fPsiPhi W M ρ ξ Ψ Φ u'

Development.AbelianVarietyTheory.Definitions.Divisors.BsLApply

theorem **upU_eq** {s : ℕ} (y : Low s → ℂ) : upU y = (GlobalTheory.FlagFibre.uni y)ᵀ

theorem **bsL_apply** {J : Type} (ε : Fin n ≃ Fin d ⊕ J) (w : V n) (a : Fin d) : bsL ε w a = w (ε.symm (.inl a))

theorem **bsL_pt** {J : Type} (ε : Fin n ≃ Fin d ⊕ J) (v : V d) (y : J → ℂ) : bsL ε (pt ε v y) = v

theorem **fb_pt** {J : Type} (ε : Fin n ≃ Fin d ⊕ J) (v : V d) (y : J → ℂ) : fb ε (pt ε v y) = y

Development.AbelianVarietyTheory.Definitions.Divisors.CapDegUniformiserLine

theorem **not_isEff_neg_div** {g : O} (hg : g ≠ 0) (hm : g ∈ maximalIdeal O) : ¬ IsEff (-Z.div g hg)

theorem **div_ne_zero** {g : O} (hg : g ≠ 0) (hm : g ∈ maximalIdeal O) : Z.div g hg ≠ 0

theorem **chiLenPic_mul** (I J : Z.effVert) (c : Pic Z.X) : chiLenPic O (divMod (I * J).1) c = chiLenPic O (divMod I.1) (c - effCls J) + chiLenPic O (divMod J.1) c

theorem **effDeg_symm** (I J : Z.effVert) (Q : PicSym Z.X) : effDeg I (firstChernPic (effCls J) * Q) = effDeg J (firstChernPic (effCls I) * Q)

theorem **vDeg_symm** (D D' : Z.VDivZ) (Q : PicSym Z.X) : Z.vDeg D (firstChernPic (Z.vCls D') * Q) = Z.vDeg D' (firstChernPic (Z.vCls D) * Q)

theorem **capDeg_symm** (D D' : (Z.toITh p).VDiv) (Q : (Z.toITh p).Ch) : (Z.toITh p).capDeg D ((Z.toITh p).cD D' * Q) = (Z.toITh p).capDeg D' ((Z.toITh p).cD D * Q)

theorem **capDeg_divP_cD** (D : (Z.toITh p).VDiv) (Q : (Z.toITh p).Ch) : (Z.toITh p).capDeg (Z.toITh p).divP ((Z.toITh p).cD D * Q) = 0

theorem **cls_div_symm** {E E' : Z.X.Modules} {r : ℕ} (φ : Z.GenIso E E') (hE : IsVB E r) (hE' : IsVB E' r) : (GenIso.div φ.symm hE' hE).cls = -(φ.div hE hE').cls

theorem **not_isEff_smul_symm** {E : Z.X.Modules} (hE : IsVB E 1) {g : O} (hg : g ≠ 0) (hm : g ∈ maximalIdeal O) : ¬ IsEff (GenIso.div (GenIso.smul E g hg).symm hE hE).cls

theorem **nonempty_coh_base** : Nonempty (O ≃ₗ[O] coh (twist (structureModule (base O).X) (LB.triv _)) 0)

theorem **vDeg_div_base** {g : O} (hg : g ≠ 0) : (base O).vDeg ((base O).div g hg) 1 = (IsDiscreteValuationRing.addVal O g).toNat

theorem **capDeg_divP_base** (p : ℕ) [Fact p.Prime] [CharZero O] [CharP (ResidueField O) p] : ((base O).toITh p).capDeg ((base O).toITh p).divP 1 = 1

theorem **capDeg_uniformiser_line** (p : ℕ) [Fact p.Prime] [CharZero O] [CharP (ResidueField O) p] {ϖ : O} (hϖ : Irreducible ϖ) : ((projSpace O 1).toITh p).capDeg (toQ ((projSpace O 1).div ϖ hϖ.ne_zero)) (c1 (formalTwistingSheaf O 1) (ProjSpace.O1_isLB O 1)) = 1 / ramIdx O p

theorem **isIso_diag** : IsIso (Z.genRes.map (biprod.map (smulHom L (Z.const g)) (smulHom M (Z.const g'))))

theorem **cls_div_diagPair** : ((diagPair hL hM g g' hg hg').div 0).cls = Z.div g hg ∧ ((diagPair hL hM g g' hg hg').div 1).cls = Z.div g' hg'

Development.AbelianVarietyTheory.Definitions.Divisors.CechDeRhamCoefMapOneFormsRestrictAlongFunctor

theorem **semiHom_coef** (T : C') : SemiHom σ (m.α T)

theorem **cechDeRhamCoefMapOneForms_D** (b : F.obj (m.G.obj T)) : q1 m T (D R _ b) = D R' _ (m.α T b)

theorem **prodFirstProjection_smul** (b : F.obj (m.G.obj T)) (ω : Ω[F.obj (m.G.obj T)⁄R]) : q1 m T (b • ω) = m.α T b • q1 m T ω

theorem **cechDeRhamCoefMapOneForms_naturality** (φ : T ⟶ T') (ω : Ω[F.obj (m.G.obj T)⁄R]) : q1 m T' (p1 F (m.G.map φ) ω) = p1 F' φ (q1 m T ω)

theorem **qOm_pOm** {j : ℕ} (φ : T ⟶ T') (x : Om R (F.obj (m.G.obj T)) j) : qOm m j T' (pOm F j (m.G.map φ) x) = pOm F' j φ (qOm m j T x)

theorem **qOm_dOne** (ω : Ω[F.obj (m.G.obj T)⁄R]) : qOm m 2 T (dOne R _ ω) = dOne R' _ (q1 m T ω)

theorem **qOm_dTwo** (x : Om R (F.obj (m.G.obj T)) 2) : qOm m 3 T (dTwo R _ x) = dTwo R' _ (qOm m 2 T x)

theorem **qOm_wedge** (a b : Ω[F.obj (m.G.obj T)⁄R]) : qOm m 2 T (wedge R _ a b) = wedge R' _ (q1 m T a) (q1 m T b)

theorem **pullTot1_f** (c : Tot1 F) : (pullTot1 m c).1 T T' φ = m.α T' (c.1 _ _ (m.G.map φ))

theorem **pullTot1_ω** (c : Tot1 F) (T : C') : (pullTot1 m c).2 T = q1 m T (c.2 _)

theorem **cechDeRhamPullbackTotalDegreeTwo_function_part** (c : Tot2 F) : (pullTot2 m c).1 T T' T'' φ ψ = m.α T'' (c.1 _ _ _ (m.G.map φ) (m.G.map ψ))

theorem **pullTot2_a** (c : Tot2 F) : (pullTot2 m c).2.1 T T' φ = q1 m T' (c.2.1 _ _ (m.G.map φ))

theorem **pullTot2_η** (c : Tot2 F) (T : C') : (pullTot2 m c).2.2 T = qOm m 2 T (c.2.2 _)

theorem **pullTot1_dTot0** (g : Tot0 F) : pullTot1 m (dTot0 F g) = dTot0 F' fun T ↦ m.α T (g _)

theorem **pullTot2_dTot1** (c : Tot1 F) : pullTot2 m (dTot1 F c) = dTot1 F' (pullTot1 m c)

theorem **pullTot2_cupTot** (a b : Tot1 F) : pullTot2 m (cupTot F a b) = cupTot F' (pullTot1 m a) (pullTot1 m b)

theorem **coe_pullZZ1** (c : cechDeRhamTotalCocyclesOne F) : (pullZZ1 m c).1 = pullTot1 m c.1

theorem **coe_pullZZ2** (c : cechDeRhamTotalCocyclesTwo F) : (pullZZ2 m c).1 = pullTot2 m c.1

theorem **pullHyp1_mk** (c : cechDeRhamTotalCocyclesOne F) : pullHyp1 m (Hyp1.mk F c) = Hyp1.mk F' (pullZZ1 m c)

theorem **pullHyp2_mk** (c : cechDeRhamTotalCocyclesTwo F) : pullHyp2 m (Hyp2.mk F c) = Hyp2.mk F' (pullZZ2 m c)

theorem **pullHyp2_cup** (a b : Hyp1 F) : pullHyp2 m (cupHyp F a b) = cupHyp F' (pullHyp1 m a) (pullHyp1 m b)

theorem **toO_pullHyp1** (c : Hyp1 F) : toO F' (pullHyp1 m c) = m.pull1 (toO F c)

theorem **pullHyp1_ofForm** (ω : Closed1 F) : pullHyp1 m (ofForm F ω) = ofForm F' (pullClosed1 m ω)

theorem **cechDeRhamCoefMapOneForms_restrictAlongFunctor** (F : C ⥤ CommAlgCat.{0} R) (G : C' ⥤ C) (T : C') (ω : Ω[F.obj (G.obj T)⁄R]) : q1 (CoefMap.restrict F G) T ω = ω

theorem **qOm_restrict** (F : C ⥤ CommAlgCat.{0} R) (G : C' ⥤ C) {j : ℕ} (T : C') (x : Om R (F.obj (G.obj T)) j) : qOm (CoefMap.restrict F G) j T x = x

theorem **pullHyp1_id** (c : Hyp1 F) : pullHyp1 (CoefMap.id F) c = c

theorem **pullHyp2_id** (c : Hyp2 F) : pullHyp2 (CoefMap.id F) c = c

theorem **pullHyp1_comp** (n : CoefMap τ F' F'') (c : Hyp1 F) : pullHyp1 (m.comp n) c = pullHyp1 n (pullHyp1 m c)

theorem **pullHyp2_comp** (n : CoefMap τ F' F'') (c : Hyp2 F) : pullHyp2 (m.comp n) c = pullHyp2 n (pullHyp2 m c)

theorem **cechDeRhamCoefMapOneForms_eq_of_natTrans** (T : C') (ω : Ω[F.obj (m.G.obj T)⁄R]) : q1 m' T (p1 F (θ.app T) ω) = q1 m T ω

theorem **qOm_pOm_app** {j : ℕ} (T : C') (x : Om R (F.obj (m.G.obj T)) j) : qOm m' j T (pOm F j (θ.app T) x) = qOm m j T x

theorem **pullHyp1_eq_of_natTrans** (c : Hyp1 F) : pullHyp1 m c = pullHyp1 m' c

theorem **pullHyp2_eq_of_natTrans** (c : Hyp2 F) : pullHyp2 m c = pullHyp2 m' c

theorem **coneForm_apply** (ω : LinearMap.ker (dOne R B)) (T : C) : (coneForm e he ω).1.1 T = (SemiHom.of_algHom (e T)).m1 ω.1

theorem **coneMap_mk** (ω : LinearMap.ker (dOne R B)) : coneMap e he (Submodule.Quotient.mk ω) = ofForm F (coneForm e he ω)

theorem **coneMap_bijective** {T₀ : C} (h : Limits.IsInitial T₀) (hb : Function.Bijective (e T₀)) : Function.Bijective (coneMap e he)

Development.AbelianVarietyTheory.Definitions.Divisors.CechDeRhamTotalDifferentialOneFunctionPart

theorem **semiHom_mp** {T T' : C} (φ : T ⟶ T') : SemiHom (RingHom.id R) (mp F φ : F.obj T →+* F.obj T')

theorem **p1_D** (b : F.obj T) : p1 F φ (D R _ b) = D R _ (mp F φ b)

theorem **oneFormsMapOfFunctorMorphism_smul** (b : F.obj T) (ω : Ω[F.obj T⁄R]) : p1 F φ (b • ω) = mp F φ b • p1 F φ ω

theorem **pOm_smul** {j : ℕ} (b : F.obj T) (x : Om R (F.obj T) j) : pOm F j φ (b • x) = mp F φ b • pOm F j φ x

theorem **oneFormsMapOfFunctorMorphism_comp** (ω : Ω[F.obj T⁄R]) : p1 F (φ ≫ ψ) ω = p1 F ψ (p1 F φ ω)

theorem **oneFormsMapOfFunctorMorphism_id** (ω : Ω[F.obj T⁄R]) : p1 F (𝟙 T) ω = ω

theorem **pOm_comp** {j : ℕ} (x : Om R (F.obj T) j) : pOm F j (φ ≫ ψ) x = pOm F j ψ (pOm F j φ x)

theorem **pOm_id** {j : ℕ} (x : Om R (F.obj T) j) : pOm F j (𝟙 T) x = x

theorem **pOm_dOm** {j : ℕ} (x : Om R (F.obj T) j) : pOm F (j + 1) φ (dOm R _ j x) = dOm R _ j (pOm F j φ x)

theorem **pOm_dOne** (ω : Ω[F.obj T⁄R]) : pOm F 2 φ (dOne R _ ω) = dOne R _ (p1 F φ ω)

theorem **pOm_dTwo** (x : Om R (F.obj T) 2) : pOm F 3 φ (dTwo R _ x) = dTwo R _ (pOm F 2 φ x)

theorem **pOm_wedge** (a b : Ω[F.obj T⁄R]) : pOm F 2 φ (wedge R _ a b) = wedge R _ (p1 F φ a) (p1 F φ b)

theorem **dTot0_f** (g : Tot0 F) : (dTot0 F g).1 T T' φ = g T' - mp F φ (g T)

theorem **dTot0_ω** (g : Tot0 F) (T : C) : (dTot0 F g).2 T = D R _ (g T)

theorem **cechDeRhamTotalDifferentialOne_functionPart** (c : Tot1 F) : (dTot1 F c).1 T T' T'' φ ψ = mp F ψ (c.1 T T' φ) - c.1 T T'' (φ ≫ ψ) + c.1 T' T'' ψ

theorem **dTot1_a** (c : Tot1 F) : (dTot1 F c).2.1 T T' φ = c.2 T' - p1 F φ (c.2 T) - D R _ (c.1 T T' φ)

theorem **dTot1_η** (c : Tot1 F) (T : C) : (dTot1 F c).2.2 T = dOne R _ (c.2 T)

theorem **Tot1.ext** {c c' : Tot1 F} (h1 : ∀ T T' φ, c.1 T T' φ = c'.1 T T' φ) (h2 : ∀ T, c.2 T = c'.2 T) : c = c'

theorem **Tot2.ext** {c c' : Tot2 F} (h1 : ∀ T T' T'' φ ψ, c.1 T T' T'' φ ψ = c'.1 T T' T'' φ ψ) (h2 : ∀ T T' φ, c.2.1 T T' φ = c'.2.1 T T' φ) (h3 : ∀ T, c.2.2 T = c'.2.2 T) : c = c'

theorem **mem_cechDeRhamTotalCocyclesOne_iff** {c : Tot1 F} : c ∈ cechDeRhamTotalCocyclesOne F ↔ c.1 ∈ Z1 F ∧ (∀ ⦃T T' : C⦄ (φ : T ⟶ T'), c.2 T' = p1 F φ (c.2 T) + D R _ (c.1 T T' φ)) ∧ ∀ T, dOne R _ (c.2 T) = 0

theorem **mem_cechDeRhamTotalCocyclesTwo_iff** {c : Tot2 F} : c ∈ cechDeRhamTotalCocyclesTwo F ↔ c.1 ∈ Z2 F ∧ (∀ ⦃T T' T'' : C⦄ (φ : T ⟶ T') (ψ : T' ⟶ T''), c.2.1 T T'' (φ ≫ ψ) = p1 F ψ (c.2.1 T T' φ) + c.2.1 T' T'' ψ + D R _ (c.1 T T' T'' φ ψ)) ∧ (∀ ⦃T T' : C⦄ (φ : T ⟶ T'), c.2.2 T' = pOm F 2 φ (c.2.2 T) + dOne R _ (c.2.1 T T' φ)) ∧ ∀ T, dTwo R _ (c.2.2 T) = 0

theorem **cechDeRhamTotalCocyclesOne.comp** (c : cechDeRhamTotalCocyclesOne F) (φ : T ⟶ T') (ψ : T' ⟶ T'') : c.1.1 T T'' (φ ≫ ψ) = mp F ψ (c.1.1 T T' φ) + c.1.1 T' T'' ψ

theorem **cechDeRhamTotalCocyclesOne.conn** (c : cechDeRhamTotalCocyclesOne F) (φ : T ⟶ T') : c.1.2 T' = p1 F φ (c.1.2 T) + D R _ (c.1.1 T T' φ)

theorem **cechDeRhamTotalCocyclesOne.closed** (c : cechDeRhamTotalCocyclesOne F) (T : C) : dOne R _ (c.1.2 T) = 0

theorem **cechDeRhamTotalCocyclesTwo.comp** (c : cechDeRhamTotalCocyclesTwo F) (φ : T ⟶ T') (ψ : T' ⟶ T'') (χ : T'' ⟶ T''') : mp F χ (c.1.1 T T' T'' φ ψ) + c.1.1 T T'' T''' (φ ≫ ψ) χ = c.1.1 T T' T''' φ (ψ ≫ χ) + c.1.1 T' T'' T''' ψ χ

theorem **cechDeRhamTotalCocyclesTwo.one** (c : cechDeRhamTotalCocyclesTwo F) (φ : T ⟶ T') (ψ : T' ⟶ T'') : c.1.2.1 T T'' (φ ≫ ψ) = p1 F ψ (c.1.2.1 T T' φ) + c.1.2.1 T' T'' ψ + D R _ (c.1.1 T T' T'' φ ψ)

theorem **cechDeRhamTotalCocyclesTwo.two** (c : cechDeRhamTotalCocyclesTwo F) (φ : T ⟶ T') : c.1.2.2 T' = pOm F 2 φ (c.1.2.2 T) + dOne R _ (c.1.2.1 T T' φ)

theorem **cechDeRhamTotalCocyclesTwo.closed** (c : cechDeRhamTotalCocyclesTwo F) (T : C) : dTwo R _ (c.1.2.2 T) = 0

theorem **dTot0_mem** (g : Tot0 F) : dTot0 F g ∈ cechDeRhamTotalCocyclesOne F

theorem **dTot1_mem** (c : Tot1 F) : dTot1 F c ∈ cechDeRhamTotalCocyclesTwo F

theorem **Hyp1.mk_surjective** : Function.Surjective (Hyp1.mk F)

theorem **Hyp2.mk_surjective** : Function.Surjective (Hyp2.mk F)

theorem **Hyp1.mk_eq_zero** {c : cechDeRhamTotalCocyclesOne F} : Hyp1.mk F c = 0 ↔ ∃ g, dTot0 F g = c.1

theorem **Hyp2.mk_eq_zero** {c : cechDeRhamTotalCocyclesTwo F} : Hyp2.mk F c = 0 ↔ ∃ e, dTot1 F e = c.1

theorem **Hyp1.mk_eq_mk** {c c' : cechDeRhamTotalCocyclesOne F} : Hyp1.mk F c = Hyp1.mk F c' ↔ ∃ g, dTot0 F g = c.1 - c'.1

theorem **Hyp2.mk_eq_mk** {c c' : cechDeRhamTotalCocyclesTwo F} : Hyp2.mk F c = Hyp2.mk F c' ↔ ∃ e, dTot1 F e = c.1 - c'.1

theorem **cupTot_apply** : (cupTot F a b).1 T T' T'' φ ψ = mp F ψ (a.1 T T' φ) ∗ b.1 T' T'' ψ

theorem **cupTot_a** : (cupTot F a b).2.1 T T' φ = a.1 T T' φ • b.2 T' − b.1 T T' φ • p1 F φ (a.2 T)

theorem **cupTot_η** (T : C) : (cupTot F a b).2.2 T = wedge R _ (a.2 T) (b.2 T)

theorem **cupTot_mem** (a b : cechDeRhamTotalCocyclesOne F) : cupTot F a.1 b.1 ∈ cechDeRhamTotalCocyclesTwo F

theorem **coe_cupZZ** (a b : cechDeRhamTotalCocyclesOne F) : (cupZZ F a b).1 = cupTot F a.1 b.1

theorem **cupHyp_mk** (a b : cechDeRhamTotalCocyclesOne F) : cupHyp F (Hyp1.mk F a) (Hyp1.mk F b) = Hyp2.mk F (cupZZ F a b)

theorem **toO_mk** (c : cechDeRhamTotalCocyclesOne F) : toO F (Hyp1.mk F c) = H1.mk F (cechDeRhamTotalCocyclesOne.fn F c)

theorem **Fil1_eq_range** : Fil1 F = LinearMap.range (ofForm F)

theorem **ofForm_eq_zero** {ω : Closed1 F} : ofForm F ω = 0 ↔ ∃ g : C0 F, (∀ ⦃T T' : C⦄ (φ : T ⟶ T'), g T' = mp F φ (g T)) ∧ ∀ T, D R _ (g T) = ω.1.1 T

Development.AbelianVarietyTheory.Definitions.Divisors.CentreAtOfIsAt

theorem **QNum.isQCart** [IsLocallyNoetherian X] {p : X ⟶ Sp k} {N : LB X} {b : ℚ} {D : X → ℚ} (h : QNum p N b D) : IsQCart D

Development.AbelianVarietyTheory.Definitions.Divisors.CharZeroFunctionField

theorem **isBallFam_int** : IsBallFam (⊤ : AddSubgroup ℤ) fun r => {x | |(x : ℝ)| ≤ r} where

theorem **exists_pair** (L A : HermLB p) : ∃ P : Pair p, P.L = L ∧ P.A = A

theorem **ratVal_nat** {σ : k →+* ℂ} (y : CxPts p σ) (n : ℕ) : ratVal (n : X.functionField) y = n

theorem **isRatSec_one** : IsRatSec (LB.triv X) fun _ => 1

theorem **isNef_trivShift** (t : ℝ) (ht : 0 ≤ t) : (ArithBase.trivShift t).IsNef 0

theorem **loc_eq** (a b : ℤ) (f) (i j) : pair.loc a b f i j = f

private theorem **pair_L_phi_apply_eq_zero** (σ : ℤ →+* ℂ) (i) (y : CxPts (𝟙 (Sp ℤ)) σ) : (pair.L.φ σ).φ i y = 0 + 2 ∗ 0

private theorem **pair_A_phi_apply_eq_two_log_three** (σ : ℤ →+* ℂ) (i) (y : CxPts (𝟙 (Sp ℤ)) σ) : (pair.A.φ σ).φ i y = 0 + 2 ∗ Real.log 3

theorem **nu_nat** (a b : ℤ) (σ) (y) (n : ℕ) : pair.nu a b σ y n = n ∗ Real.exp (-(b ∗ Real.log 3))

theorem **regAt_nat** (x : Sp ℤ) (n : ℕ) : RegAt x (n : (Sp ℤ).functionField)

theorem **charZero_functionField** : CharZero (Sp ℤ).functionField

private theorem **nonempty_cxPts_spec_int** : Nonempty (CxPts (𝟙 (Sp ℤ)) (Int.castRingHom ℂ))

theorem **nat_mem_Bl_iff** (a : ℤ) (n : ℕ) : (n : (Sp ℤ).functionField) ∈ pair.Bl a 1 1 ↔ n ≤ 3

theorem **good** : pair.Good 0 1 (2 : ℕ) (3 : ℕ) where

theorem **not_good** : ¬ pair.Good 0 1 (2 : ℕ) (4 : ℕ)

private theorem **trivShift_phi_apply_eq_two_mul** (t : ℝ) (σ : ℤ →+* ℂ) (c) (y : CxPts (𝟙 (Sp ℤ)) σ) : ((ArithBase.trivShift t).φ σ).φ c y = 0 + 2 ∗ t

theorem **rep** (a b : ℤ) : pair.Rep a b (ArithBase.trivShift (b ∗ Real.log 3))

Development.AbelianVarietyTheory.Definitions.Divisors.ChartMeasApplyOfSubset

theorem **isOpen_over** (e : OpenPartialHomeomorph X (V n)) {S : Set X} (hS : IsOpen S) : IsOpen (e.target ∩ e.symm ⁻¹' S)

theorem **mem_over** (e : OpenPartialHomeomorph X (V n)) {S : Set X} {y : X} (hy : y ∈ e.source) (hS : y ∈ S) : e y ∈ e.target ∩ e.symm ⁻¹' S

theorem **contDiffOn_real** {g : V n → ℂ} {U : Set (V n)} (hU : IsOpen U) (hg : DifferentiableOn ℂ g U) : ContDiffOn ℝ ∞ g U

theorem **contDiffOn_log_normSq** {g : V n → ℂ} {U : Set (V n)} (hU : IsOpen U) (hg : DifferentiableOn ℂ g U) (h0 : ∀ v ∈ U, g v ≠ 0) : ContDiffOn ℝ ∞ (fun v => Real.log (‖g v‖ ^ 2)) U

theorem **chartMeas_apply** (e : OpenPartialHomeomorph X (V n)) (ρ : X → ℝ≥0∞) {A : Set X} (hA : MeasurableSet A) : chartMeas e ρ A = ∫⁻ v in e '' (e.source ∩ A), ρ (e.symm v)

theorem **chartMeas_apply_of_subset** (e : OpenPartialHomeomorph X (V n)) (ρ : X → ℝ≥0∞) {A : Set X} (hA : MeasurableSet A) (h : A ⊆ e.source) : chartMeas e ρ A = ∫⁻ v in e '' A, ρ (e.symm v)

theorem **measurableSet_image** (e : OpenPartialHomeomorph X (V n)) {A : Set X} (hA : MeasurableSet A) (h : A ⊆ e.source) : MeasurableSet (e '' A)

theorem **chartMeas_eq** (e e' : OpenPartialHomeomorph X (V n)) {ρ ρ' : X → ℝ≥0∞} (hd : DifferentiableOn ℂ (e' ∘ e.symm) (e.target ∩ e.symm ⁻¹' e'.source)) {A : Set X} (hA : MeasurableSet A) (h : A ⊆ e.source) (h' : A ⊆ e'.source) (hρ : ∀ y ∈ A, ρ y = ENNReal.ofReal (‖transDet e e' y‖ ^ 2) ∗ ρ' y) : chartMeas e ρ A = chartMeas e' ρ' A

theorem **iSup_apply_of_compat** (μ : ι → Measure X) (S : ι → Set X) (hS : ∀ i, IsOpen (S i)) (h0 : ∀ i A, MeasurableSet A → μ i A = μ i (S i ∩ A)) (hc : ∀ i j A, MeasurableSet A → A ⊆ S i → A ⊆ S j → μ i A = μ j A) (i : ι) {A : Set X} (hA : MeasurableSet A) (hAi : A ⊆ S i) : (⨆ j, μ j) A = μ i A

theorem **ext_of_cover** {μ ν : Measure X} (S : ι → Set X) (hS : ∀ i, IsOpen (S i)) (hc : ∀ y, ∃ i, y ∈ S i) (h : ∀ i A, MeasurableSet A → A ⊆ S i → μ A = ν A) : μ = ν

theorem **glueMeas_apply** (e : ι → OpenPartialHomeomorph X (V n)) (ρ : ι → X → ℝ≥0∞) (hd : ∀ i j, DifferentiableOn ℂ (e j ∘ (e i).symm) (ovl e i j)) (hρ : ∀ i j y, y ∈ (e i).source → y ∈ (e j).source → ρ i y = ENNReal.ofReal (‖transDet (e i) (e j) y‖ ^ 2) ∗ ρ j y) (i : ι) {A : Set X} (hA : MeasurableSet A) (h : A ⊆ (e i).source) : glueMeas e ρ A = chartMeas (e i) (ρ i) A

theorem **differentiableOn_prod** {κ : Type∗} [Fintype κ] {f : κ → V n → ℂ} {U : Set (V n)} (h : ∀ a, DifferentiableOn ℂ (f a) U) : DifferentiableOn ℂ (fun v => ∏ a, f a v) U

theorem **log_normSq_prod_zpow** {κ : Type∗} [Fintype κ] (g : κ → ℂ) (hg : ∀ a, g a ≠ 0) (m : κ → ℤ) : Real.log (‖∏ a, g a ^ m a‖ ^ 2) = ∑ a, m a ∗ Real.log (‖g a‖ ^ 2)

Development.AbelianVarietyTheory.Definitions.Divisors.ChernFormSecondChernClass

theorem **pm_one** : pm 1 = C (1 / 6) ∗ X 0 ^ 2 + C (1 / 3) ∗ X 1

theorem **pm_two** : pm 2 = C (1 / 30) ∗ X 0 ^ 4 + C (1 / 15) ∗ X 0 ^ 2 ∗ X 1 + C (1 / 5) ∗ X 1 ^ 2

theorem **pm_three** : pm 3 = C (1 / 140) * X 0 ^ 6 + C (1 / 70) * X 0 ^ 4 * X 1 + C (3 / 70) * X 0 ^ 2 * X 1 ^ 2 + C (1 / 7) * X 1 ^ 3

theorem **hs_small** (a b : ℝ) : aeval ![a + b, a * b] (hs 0) = 1 ∧ aeval ![a + b, a * b] (hs 1) = a + b ∧ aeval ![a + b, a * b] (hs 3) = a ^ 3 + a ^ 2 * b + a * b ^ 2 + b ^ 3 ∧ aeval ![a + b, a * b] (hs 4) = a ^ 4 + a ^ 3 * b + a ^ 2 * b ^ 2 + a * b ^ 3 + b ^ 4 ∧ aeval ![a + b, a * b] (hs 5) = a ^ 5 + a ^ 4 * b + a ^ 3 * b ^ 2 + a ^ 2 * b ^ 3 + a * b ^ 4 + b ^ 5

theorem **sum_idx** {S : Type} [AddCommMonoid S] (d : ℕ) (f : ℕ × ℕ → S) : ∑ x ∈ idx d, f x = ∑ i ∈ Finset.range (d + 3), ∑ j ∈ Finset.range i, f (i, j)

theorem **polynomialOfCoeffs_eq_swapPolynomial** (d : ℕ) (a b : ℤ) : upperParabolic (idx d) expo (coefW (swapR d a b)) = swapP d a b

theorem **beta** (j : ℕ) : ∫ t in (0 : ℝ)..1, (t * (1 - t)) ^ j = (j.factorial * j.factorial : ℝ) / (2 * j + 1).factorial

theorem **aeval_pm** (m : ℕ) (a b : ℝ) : aeval ![a + b, a * b] (pm m) = ∫ t in (0 : ℝ)..1, ((a * t + b * (1 - t)) * (a * (1 - t) + b * t)) ^ m

theorem **pm_eq_fibPoly_val** (m : ℕ) (a b : ℝ) : aeval ![a + b, a * b] (pm m) = aeval ![a + b, a * b] (fibPoly m)

theorem **downP_eq_baseT** (d : ℕ) (R : ℕ → MvPolynomial (Fin 2) ℝ) (A a b : ℝ) : aeval ![A, a, b, a * b] (downP (idx d) expo (coefW R)) = aeval ![a + b, a * b, A] (baseT d R)

theorem **omF_eq** {E : X.Modules} {m : VMetric p E 2} (H : m.IsCInf) (fl : Vert.Flag E 2) (x : (h.atlas σ).Pt) (a : VBExt.AdaptedChart (lineSC fl) 1 1) (ha : x.pt ∈ ptsIn p σ a.U) : omF h σ m fl x = ofValues (extTrace (fun v => (m.met σ).gram a.frame (x.ch.symm v)) x.crd)

theorem **dens_ofValues** (α : ReciprocityTheory.L2DbarEstimates.V 1 → ReciprocityTheory.L2DbarEstimates.V 1 → ℂ) : (dens ((ofValues α : Ev 1) : Alt 1)).re = 2 * diagVal α (EuclideanSpace.single 0 1)

theorem **glue_congr** {M ι : Type} [TopologicalSpace M] [MeasurableSpace M] {n : ℕ} (e : ι → OpenPartialHomeomorph M (ReciprocityTheory.L2DbarEstimates.V n)) {ρ ρ' : ι → M → ENNReal} (H : ∀ i, ∀ y ∈ (e i).source, ρ i y = ρ' i y) : ChartGlue.glueMeas e ρ = ChartGlue.glueMeas e ρ'

theorem **part_omF** (h : IsRegVar p 1) {E : X.Modules} {m : VMetric p E 2} (H : m.IsCInf) (fl : Vert.Flag E 2) : (h.atlas σ).part (fld _ (omF h σ m fl)) 1 = Met.bcMeas (S

theorem **integral_omF_line** (σ : ℤ →+* ℂ) : ((isRegVar_proj 1).atlas σ).integral (fld _ (omF (isRegVar_proj 1) σ TautFlag.stdH.h TautFlag.flag)) = 1

theorem **wInt_c1F** (h : IsRegVar p 1) (L : HermLB p) (g : CxPts p σ → ℝ) : wInt (h.atlas σ) g (fld _ (c1F h σ L)) = (L.φ σ).chernInt 1 Set.univ g

theorem **chernForm_firstChernClass** {B : BottChern} (R : ArRing B h) (L : HermLB p) : R.om (R.c1 L) σ = c1F h σ L

theorem **evalF_X** {M : Type} [TopologicalSpace M] {n : ℕ} (A : HolAtlas M n) {ι : Type} (w : ι → A.Pt → Ev n) (i : ι) (P Q : MvPolynomial ι ℝ) : evalF A w (MvPolynomial.X i) = fld A (w i) ∧ evalF A w (P * Q) = evalF A w P * evalF A w Q ∧ evalF A w 1 = 1

theorem **chernAt_submatrix** {n : ℕ} {r r' : Type} [Fintype r] [DecidableEq r] [Fintype r'] [DecidableEq r'] (e : r' ≃ r) {G : ReciprocityTheory.L2DbarEstimates.V n → Matrix r r ℂ} {G' : ReciprocityTheory.L2DbarEstimates.V n → Matrix r' r' ℂ} {v : ReciprocityTheory.L2DbarEstimates.V n} (he : G' =ᶠ[nhds v] fun u => (G u).submatrix e e) (j : ℕ) : chernAt G v j = chernAt G' v j

theorem **chernForm_secondChernClass** {B : BottChern} (R : ArRing B h) (E : HermVB p 2) (fl : Vert.Flag E.E 2) : R.om (R.cl E 2) σ = c2F h σ E.h fl

Development.AbelianVarietyTheory.Definitions.Divisors.CircleAsphGNontrivial

theorem **circleAsph_G_nontrivial** : Nontrivial (Multiplicative (AddSubgroup.zmultiples (1 : ℝ)))

Development.AbelianVarietyTheory.Definitions.Divisors.CmHostImaginaryGeneratorNeZero

theorem **cmHost_imaginaryGenerator_ne_zero** : D.host.xi0 ≠ 0

theorem **negSp_gram** (V : HSp D) : (negSp V).gram = -V.gram

theorem **HR_neg** (V : HSp D) : HR (AF D.host.F) D.sp.cm.sig V.gram = HR (AF D.host.F) D.sp.cm.sig (negSp V).gram

theorem **mt_negGA** (V : HSp D) (g : GA D V) : mt (negSp V) (negGA V g) = mt V g

theorem **continuous_negGA** (V : HSp D) : Continuous (negGA V)

theorem **isSelfDualC_neg** {V : HSp D} {Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)} (h : IsSelfDualC V Λ) : IsSelfDualC (negSp V) Λ

theorem **stairId_succ** {𝔞 𝔞' : FractionalIdeal (𝒪 D.host.E)⁰ D.host.E} (h : 𝔞' ≠ 0) {c : ℕ} (p q : Fin c) (hpq : q.1 = p.1 + 1) : stairId 𝔞 𝔞' c p = 𝔞 * 𝔞' * stairId 𝔞 𝔞' c q

theorem **stairId_rev** (𝔞 𝔞' : FractionalIdeal (𝒪 D.host.E)⁰ D.host.E) {c : ℕ} (p : Fin c) : stairId 𝔞' 𝔞 c p.rev = (stairId 𝔞 𝔞' c p)⁻¹

theorem **gram_sub** {k l : ℕ} (f : Fin k → Fin c) (g : Fin l → Fin c) : Formalism.UG.cT (D.sp.cm.sig : D.host.E →+* D.host.E) (d.a.submatrix id f) * B.gram * d.a.submatrix id g = (antiDiagonalSignGramMatrix c ε).submatrix f g

theorem **isIsoSub_iso** (ρ : ℕ) (h : 2 * ρ ≤ c) : IsIsoSub B (d.iso ρ h)

theorem **mem_congLev** {𝔪 : Ideal (𝒪 E)} {g : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} : g ∈ congLev hc J Λ 𝔪 ↔ g ∈ UA hc J ∧ g ∈ Formalism.UG.stabGL (hatL (F

theorem **congLev_top** : congLev hc J Λ ⊤ = UA hc J ⊓ Formalism.UG.stabGL (hatL (F

theorem **congLev_mono** {𝔪 𝔫 : Ideal (𝒪 E)} (h : 𝔪 ≤ 𝔫) : congLev hc J Λ 𝔪 ≤ congLev hc J Λ 𝔫

theorem **homLat_fg** {m n : ℕ} {Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} {Λ' : Submodule (𝒪 D.host.E) (Fin m → D.host.E)} (h : Submodule.IsLattice D.host.E Λ) (h' : Λ'.FG) : Module.Finite (𝒪 D.host.E) (homLat Λ Λ')

theorem **finite_cos** {m n : ℕ} {Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} {Λ' : Submodule (𝒪 D.host.E) (Fin m → D.host.E)} (h : Submodule.IsLattice D.host.E Λ) (h' : Λ'.FG) {𝔪 : Ideal (𝒪 D.host.E)} (h𝔪 : 𝔪 ≠ ⊥) : Finite (Cos Λ Λ' 𝔪)

theorem **diagSp_gram** (n : ℕ) (neg : Fin n → Bool) (p q : ℕ) : (diagSp D n neg p q).gram = Matrix.diagonal fun i : Fin n => (Examples₂.sgn (neg i) : D.host.E)

Development.AbelianVarietyTheory.Definitions.Divisors.CommAlgFunctorCoboundaryZeroMemCocyclesOne

theorem **mp_comp** {T T' T'' : C} (φ : T ⟶ T') (ψ : T' ⟶ T'') (b : F.obj T) : mp F (φ ≫ ψ) b = mp F ψ (mp F φ b)

theorem **mp_id** (T : C) (b : F.obj T) : mp F (𝟙 T) b = b

theorem **amitsur_d0_apply** (g : C0 F) {T T' : C} (φ : T ⟶ T') : d0 F g T T' φ = g T' - mp F φ (g T)

theorem **d1_apply** (f : C1 F) {T T' T'' : C} (φ : T ⟶ T') (ψ : T' ⟶ T'') : d1 F f T T' T'' φ ψ = mp F ψ (f T T' φ) - f T T'' (φ ≫ ψ) + f T' T'' ψ

theorem **Z1.comp** (f : Z1 F) {T T' T'' : C} (φ : T ⟶ T') (ψ : T' ⟶ T'') : f.1 T T'' (φ ≫ ψ) = mp F ψ (f.1 T T' φ) + f.1 T' T'' ψ

theorem **Z1.id** (f : Z1 F) (T : C) : f.1 T T (𝟙 T) = 0

theorem **commAlgFunctorCocyclesOne_eq_ker** : Z1 F = LinearMap.ker (d1 F)

theorem **commAlgFunctorCoboundaryZero_mem_cocyclesOne** (g : C0 F) : d0 F g ∈ Z1 F

theorem **differential_one_mem_scaled_lattice** (f : C1 F) : d1 F f ∈ Z2 F

theorem **H1.mk_surjective** : Function.Surjective (H1.mk F)

theorem **H2.mk_surjective** : Function.Surjective (H2.mk F)

theorem **H1.mk_eq_zero** {f : Z1 F} : H1.mk F f = 0 ↔ ∃ g, d0 F g = f.1

theorem **H2.mk_eq_zero** {z : Z2 F} : H2.mk F z = 0 ↔ ∃ f, d1 F f = z.1

theorem **H1.mk_eq_mk** {f f' : Z1 F} : H1.mk F f = H1.mk F f' ↔ ∃ g, d0 F g = f.1 - f'.1

theorem **H2.mk_eq_mk** {z z' : Z2 F} : H2.mk F z = H2.mk F z' ↔ ∃ f, d1 F f = z.1 - z'.1

theorem **cupZ_apply** (a b : Z1 F) {T T' T'' : C} (φ : T ⟶ T') (ψ : T' ⟶ T'') : (cupZ F a b).1 T T' T'' φ ψ = mp F ψ (a.1 T T' φ) ∗ b.1 T' T'' ψ

theorem **cup_mk** (a b : Z1 F) : cup F (H1.mk F a) (H1.mk F b) = H2.mk F (cupZ F a b)

theorem **H1.eq_zero_of_isInitial** {T₀ : C} (h : Limits.IsInitial T₀) (c : H1 F) : c = 0

theorem **pullCocycleOne_apply** (f : Z1 F) {T T' : C'} (φ : T ⟶ T') : (m.pullbackOneCocycle f).1 T T' φ = m.α T' (f.1 _ _ (m.G.map φ))

theorem **pullCocycleTwo_apply** (z : Z2 F) {T T' T'' : C'} (φ : T ⟶ T') (ψ : T' ⟶ T'') : (m.pullbackTwoCocycle z).1 T T' T'' φ ψ = m.α T'' (z.1 _ _ _ (m.G.map φ) (m.G.map ψ))

theorem **pullCohomologyOne_class** (f : Z1 F) : m.pull1 (H1.mk F f) = H1.mk F' (m.pullbackOneCocycle f)

theorem **pullCohomologyTwo_class** (z : Z2 F) : m.pull2 (H2.mk F z) = H2.mk F' (m.pullbackTwoCocycle z)

theorem **pullCohomologyTwo_cup** (a b : H1 F) : m.pull2 (cup F a b) = cup F' (m.pull1 a) (m.pull1 b)

theorem **pullCohomologyOne_id** (c : H1 F) : (id F).pull1 c = c

theorem **pullCohomologyTwo_id** (c : H2 F) : (id F).pull2 c = c

theorem **pullCohomologyOne_comp** (n : CoefMap τ F' F'') (c : H1 F) : (m.comp n).pull1 c = n.pull1 (m.pull1 c)

theorem **pullCohomologyTwo_comp** (n : CoefMap τ F' F'') (c : H2 F) : (m.comp n).pull2 c = n.pull2 (m.pull2 c)

theorem **pullCohomologyOne_eq_of_natTrans** {σ' : R →+* R'} (m' : CoefMap σ' F F') (θ : m.G ⟶ m'.G) (h : ∀ T b, m'.α T (mp F (θ.app T) b) = m.α T b) (c : H1 F) : m.pull1 c = m'.pull1 c

theorem **pullCohomologyTwo_eq_of_natTrans** {σ' : R →+* R'} (m' : CoefMap σ' F F') (θ : m.G ⟶ m'.G) (h : ∀ T b, m'.α T (mp F (θ.app T) b) = m.α T b) (c : H2 F) : m.pull2 c = m'.pull2 c

Development.AbelianVarietyTheory.Definitions.Divisors.CommGroupMulMulEqMulInvMulMulMul

theorem **memInf** {U V : Y.Opens} {y : Y} (h : y ∈ U) (h' : y ∈ V) : y ∈ U ⊓ V

@[simp] theorem **ger_resG** {U V : Y.Opens} (h : V ≤ U) (y : Y) (hy : y ∈ V) (a : Γ(Y, U)) : ger y hy (resG Y h a) = ger y (h hy) a

theorem **fun_ext_ger** {U : Y.Opens} {a b : Γ(Y, U)} (h : ∀ y hy, ger y hy a = ger y hy b) : a = b

theorem **ger_appLE** (r : T ⟶ Y) {U : Y.Opens} {W : T.Opens} (h : W ≤ r ⁻¹ᵁ U) (z : T) (hz : z ∈ W) (a : Γ(Y, U)) : ger z hz ((r.appLE U W h).hom a) = (r.stalkMap z).hom (ger (r z) (h hz) a)

theorem **val_tr** (B : LB Y) (y : Y) {i j : B.ι} (h : y ∈ B.U i ⊓ B.U j) : (tr B y i j).val = ger y h (B.g i j).val

theorem **tr_tens** (A B : LB Y) (y : Y) {a b : (tens A B).ι} (ha : y ∈ (tens A B).U a) (hb : y ∈ (tens A B).U b) : tr (tens A B) y a b = tr A y a.1 b.1 ∗ tr B y a.2 b.2

theorem **tr_div** (A B : LB Y) (y : Y) {a b : (LB.div A B).ι} (ha : y ∈ (LB.div A B).U a) (hb : y ∈ (LB.div A B).U b) : tr (LB.div A B) y a b = tr A y a.1 b.1 ∗ (tr B y a.2 b.2)⁻¹

theorem **tr_zpow** (L : LB Y) (n : ℤ) (y : Y) {a b : (lbZPow L n).ι} (ha : y ∈ (lbZPow L n).U a) (hb : y ∈ (lbZPow L n).U b) : tr (lbZPow L n) y a b = tr L y (a PUnit.unit) (b PUnit.unit) ^ n

theorem **tr_pull** (B : LB Y) (r : T ⟶ Y) (z : T) {a b : (B.pull r).ι} (ha : z ∈ (B.pull r).U a) (hb : z ∈ (B.pull r).U b) : tr (B.pull r) z a b = Units.map (r.stalkMap z).hom.toMonoidHom (tr B (r z) (a PUnit.unit) (b PUnit.unit))

theorem **base_smul** (c : k) (f : H0 p B O) : c • f = algX p O c • f

theorem **germ_apply** (i : B.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ B.U i) (f : H0 p B O) : germ i y hy hi f = ger y (memInf hy hi) (f.loc i)

theorem **germ_fun_smul** (i : B.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ B.U i) (a : Γ(Y, O)) (f : H0 p B O) : germ i y hy hi (a • f) = ger y hy a ∗ germ i y hy hi f

theorem **ger_algX** (y : Y) (hy : y ∈ O) (c : k) : ger y hy (algX p O c) = const p y c

theorem **germ_smul** (i : B.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ B.U i) (c : k) (f : H0 p B O) : germ i y hy hi (c • f) = const p y c ∗ germ i y hy hi f

theorem **germ_change** (i j : B.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ B.U i) (hj : y ∈ B.U j) (f : H0 p B O) : germ i y hy hi f = tr B y i j ∗ germ j y hy hj f

theorem **ext_germ** {f f' : H0 p B O} (h : ∀ y (hy : y ∈ O), ∃ i hi, germ i y hy hi f = germ i y hy hi f') : f = f'

@[simp] theorem **germ_restr** (h : O' ≤ O) (i : B.ι) (y : Y) (hy : y ∈ O') (hi : y ∈ B.U i) (f : H0 p B O) : germ i y hy hi (restr h f) = germ i y (h hy) hi f

@[simp] theorem **germ_ofSec** (i : B.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ B.U i) (σ : LB.Sec B) : germ i y hy hi (ofSec p O σ) = ger y hi (σ.f i)

@[simp] theorem **restr_ofSec** (h : O' ≤ O) (σ : LB.Sec B) : restr h (ofSec p O σ) = ofSec p O' σ

theorem **ofSec_smul** (c : k) (σ : LB.Sec B) : ofSec p O (algX p ⊤ c • σ) = c • ofSec p O σ

theorem **bijective_ofSec_top** : Function.Bijective (ofSec p (B

theorem **finrank_top** {E : Type} [Field E] (p : Y ⟶ Sp E) (B : LB Y) : letI

theorem **exists_of_germs** (hB : B.IsLB) {τ : Type*} (V : τ → Y.Opens) (c : τ → B.ι) (hO : ∀ t, V t ≤ O) (hc : ∀ t, V t ≤ B.U (c t)) (cover : ∀ y ∈ O, ∃ t, y ∈ V t) (u : ∀ t, Γ(Y, V t)) (rule : ∀ y t t' (h : y ∈ V t) (h' : y ∈ V t'), ger y h (u t) = tr B y (c t) (c t') ∗ ger y h' (u t')) : ∃ f : H0 p B O, ∀ t y (h : y ∈ V t), germ (c t) y (hO t h) (hc t h) f = ger y h (u t)

theorem **germ_pullSL** (r : T ⟶ Y) {q : T ⟶ Sp k'} {φ : k →+* k'} (hq : r ≫ p = q ≫ spMap φ) {W : T.Opens} (h : W ≤ r ⁻¹ᵁ O) (j : (B.pull r).ι) (z : T) (hz : z ∈ W) (hj : z ∈ (B.pull r).U j) (f : H0 p B O) : germ j z hz hj (pullSL r hq h f) = (r.stalkMap z).hom (germ (j PUnit.unit) (r z) (h hz) (mem_pull.1 hj) f)

theorem **germ_pull** (r : T ⟶ Y) {W : T.Opens} (h : W ≤ r ⁻¹ᵁ O) (j : (B.pull r).ι) (z : T) (hz : z ∈ W) (hj : z ∈ (B.pull r).U j) (f : H0 p B O) : germ j z hz hj (pull r h f) = (r.stalkMap z).hom (germ (j PUnit.unit) (r z) (h hz) (mem_pull.1 hj) f)

@[simp] theorem **pull_restr** (r : T ⟶ Y) {W : T.Opens} (h' : O' ≤ O) (h : W ≤ r ⁻¹ᵁ O') (f : H0 p B O) : pull r h (restr h' f) = pull r (h.trans (r.preimage_mono h')) f

theorem **stalkMap_const** (r : T ⟶ Y) (z : T) (c : k) : (r.stalkMap z).hom (const p (r z) c) = const (r ≫ p) z c

theorem **val_apply** (x : Sp R ⟶ Y) (i : B.ι) (hO : ⊤ ≤ x ⁻¹ᵁ O) (hi : ⊤ ≤ x ⁻¹ᵁ B.U i) (f : H0 p B O) : val x i hO hi f = evalPt x (O ⊓ B.U i) (le_preimage_inf hO hi) (f.loc i)

theorem **val_change** (x : Sp R ⟶ Y) (i j : B.ι) (hO : ⊤ ≤ x ⁻¹ᵁ O) (hi : ⊤ ≤ x ⁻¹ᵁ B.U i) (hj : ⊤ ≤ x ⁻¹ᵁ B.U j) (f : H0 p B O) : val x i hO hi f = evalPt x (B.U i ⊓ B.U j) (le_preimage_inf hi hj) (B.g i j).val ∗ val x j hO hj f

theorem **val_fun_smul** (x : Sp R ⟶ Y) (i : B.ι) (hO : ⊤ ≤ x ⁻¹ᵁ O) (hi : ⊤ ≤ x ⁻¹ᵁ B.U i) (a : Γ(Y, O)) (f : H0 p B O) : val x i hO hi (a • f) = evalPt x O hO a ∗ val x i hO hi f

theorem **val_ofSec** (x : Sp R ⟶ Y) (i : B.ι) (hO : ⊤ ≤ x ⁻¹ᵁ O) (hi : ⊤ ≤ x ⁻¹ᵁ B.U i) (σ : LB.Sec B) : val x i hO hi (ofSec p O σ) = evalPt x _ hi (σ.f i)

theorem **val_pull** (r : T ⟶ Y) {W : T.Opens} (h : W ≤ r ⁻¹ᵁ O) (x : Sp R ⟶ T) (j : (B.pull r).ι) (hW : ⊤ ≤ x ⁻¹ᵁ W) (hj : ⊤ ≤ x ⁻¹ᵁ (B.pull r).U j) (hO : ⊤ ≤ (x ≫ r) ⁻¹ᵁ O) (hi : ⊤ ≤ (x ≫ r) ⁻¹ᵁ B.U (j PUnit.unit)) (f : H0 p B O) : val x j hW hj (pull r h f) = val (x ≫ r) (j PUnit.unit) hO hi f

theorem **nowhereZero_of_exists** {p : Y ⟶ Sp k} {B : LB Y} {O : Y.Opens} {f : H0 p B O} (h : ∀ y (hy : y ∈ O), ∃ i hi, IsUnit (germ i y hy hi f)) : NowhereZero f

theorem **isUnit_ger_iff** {B : LB Y} (σ : LB.Sec B) {y : Y} {i : B.ι} (hi : y ∈ B.U i) : IsUnit (ger y hi (σ.f i)) ↔ y ∈ nonzeroLocus σ

theorem **nowhereZero_pull_ofSec** {p : Y ⟶ Sp k} {B : LB Y} {O : Y.Opens} (σ : LB.Sec B) (r : T ⟶ Y) {W : T.Opens} (h : W ≤ r ⁻¹ᵁ O) (hσ : W ≤ r ⁻¹ᵁ nonzeroLocus σ) : NowhereZero (H0.pull r h (H0.ofSec p O σ))

theorem **exists_mul** (σ : H0 p P O) (f : H0 p B O) : ∃ g : H0 p N O, ∀ t y hy ha hb hc, germ (C.c t) y hy hc g = germ (C.a t) y hy ha σ ∗ germ (C.b t) y hy hb f

theorem **sec_ext** {g g' : H0 p N O} (h : ∀ t y hy, y ∈ P.U (C.a t) → y ∈ B.U (C.b t) → ∀ hc, germ (C.c t) y hy hc g = germ (C.c t) y hy hc g') : g = g'

theorem **germ_mulFun** (σ : H0 p P O) (f : H0 p B O) (t : C.τ) (y : Y) (hy : y ∈ O) (ha : y ∈ P.U (C.a t)) (hb : y ∈ B.U (C.b t)) (hc : y ∈ N.U (C.c t)) : germ (C.c t) y hy hc (C.mulFun σ f) = germ (C.a t) y hy ha σ ∗ germ (C.b t) y hy hb f

theorem **germ_mul** (σ : H0 p P O) (f : H0 p B O) (t : C.τ) (y : Y) (hy : y ∈ O) (ha : y ∈ P.U (C.a t)) (hb : y ∈ B.U (C.b t)) (hc : y ∈ N.U (C.c t)) : germ (C.c t) y hy hc (C.mul σ f) = germ (C.a t) y hy ha σ ∗ germ (C.b t) y hy hb f

theorem **resG_loc_mul** (σ : H0 p P O) (f : H0 p B O) (t : C.τ) {W : Y.Opens} (ha : W ≤ O ⊓ P.U (C.a t)) (hb : W ≤ O ⊓ B.U (C.b t)) (hc : W ≤ O ⊓ N.U (C.c t)) : resG Y hc ((C.mul σ f).loc _) = resG Y ha (σ.loc _) ∗ resG Y hb (f.loc _)

theorem **val_mul** {R : Type} [CommRing R] (σ : H0 p P O) (f : H0 p B O) (t : C.τ) (x : Sp R ⟶ Y) (hO : ⊤ ≤ x ⁻¹ᵁ O) (ha : ⊤ ≤ x ⁻¹ᵁ P.U (C.a t)) (hb : ⊤ ≤ x ⁻¹ᵁ B.U (C.b t)) (hc : ⊤ ≤ x ⁻¹ᵁ N.U (C.c t)) : val x (C.c t) hO hc (C.mul σ f) = val x (C.a t) hO ha σ ∗ val x (C.b t) hO hb f

theorem **restr_mul** (h : O' ≤ O) (σ : H0 p P O) (f : H0 p B O) : restr h (C.mul σ f) = C.mul (restr h σ) (restr h f)

theorem **bijective_mul** {σ : H0 p P O} (hσ : NowhereZero σ) : Function.Bijective (C.mul σ)

@[simp] theorem **mulEquiv_apply** {σ : H0 p P O} (hσ : NowhereZero σ) (f : H0 p B O) : C.mulEquiv hσ f = C.mul σ f

theorem **pullSL_mul** (r : T ⟶ Y) {q : T ⟶ Sp k'} {φ : k →+* k'} (hq : r ≫ p = q ≫ spMap φ) {W : T.Opens} (h : W ≤ r ⁻¹ᵁ O) (σ : H0 p P O) (f : H0 p B O) : pullSL r hq h (C.mul σ f) = (C.pull r).mul (pullSL r hq h σ) (pullSL r hq h f)

theorem **pull_mul** (r : T ⟶ Y) {W : T.Opens} (h : W ≤ r ⁻¹ᵁ O) (σ : H0 p P O) (f : H0 p B O) : H0.pull r h (C.mul σ f) = (C.pull r).mul (H0.pull r h σ) (H0.pull r h f)

theorem **isLB_tw** (hM : M.IsLB) (hL : L.IsLB) (n : ℤ) : (tw M L n).IsLB

theorem **mem_zpow_U** {y : Y} {j : L.ι} : y ∈ (lbZPow L n).U (fun _ => j) ↔ y ∈ L.U j

theorem **mem_tw_U** {y : Y} {i : M.ι} {j : L.ι} : y ∈ (tw M L n).U (twIdx n i j) ↔ y ∈ M.U i ∧ y ∈ L.U j

theorem **tr_tw** (y : Y) {i i' : M.ι} {j j' : L.ι} (h : y ∈ (tw M L n).U (twIdx n i j)) (h' : y ∈ (tw M L n).U (twIdx n i' j')) : tr (tw M L n) y (twIdx n i j) (twIdx n i' j') = tr M y i i' ∗ tr L y j j' ^ n

theorem **commGroup_mul_mul_eq_mul_inv_mul_mul_mul** {G : Type*} [CommGroup G] (m a b x y : G) : m ∗ a ∗ x = a ∗ (b ∗ y)⁻¹ ∗ (m ∗ b ∗ (x ∗ y))

Development.AbelianVarietyTheory.Definitions.Divisors.CompEqOneIffPush

theorem **IsLin.id** (ρ : RingAct A O) : IsLin ρ ρ (𝟙 A)

theorem **IsLin.comp** {ρ : RingAct A O} {ρ' : RingAct A' O} {ρ'' : RingAct A'' O} {f : A ⟶ A'} {g : A' ⟶ A''} (hf : IsLin ρ ρ' f) (hg : IsLin ρ' ρ'' g) : IsLin ρ ρ'' (f ≫ g)

theorem **IsIsogDeg.of_isIso** (f : A ⟶ A') [IsIso f] : IsIsogDeg f 1

theorem **IsIsogDeg.comp** {f : A ⟶ A'} {g : A' ⟶ A''} {d e : ℕ} (hf : IsIsogDeg f d) (hg : IsIsogDeg g e) : IsIsogDeg (f ≫ g) (d ∗ e)

theorem **IsIsogDeg.epi** {f : A ⟶ A'} {d : ℕ} (hf : IsIsogDeg f d) : Epi f

theorem **IsIsogDeg.of_isPullback** {C C' : Over S'} {f : A ⟶ A'} {g : C ⟶ C'} {d : ℕ} (hf : IsIsogDeg f d) {u : C.left ⟶ A.left} {v : C'.left ⟶ A'.left} (H : IsPullback u g.left f.left v) : IsIsogDeg g d

theorem **IsIsogDeg.ker** {f : A ⟶ A'} {d : ℕ} (hf : IsIsogDeg f d) : IsIsogDeg (toUnit (ker f)) d

theorem **killedBy_span** (ρ : RingAct A O) (s : Set O) {T : Over S} (x : T ⟶ A) : KilledBy ρ (Ideal.span s) x ↔ ∀ a ∈ s, x ≫ ρ.act a = 1

theorem **IsLin.pull** {ρ : RingAct A O} {ρ' : RingAct A' O} {g : A ⟶ A'} (h : IsLin ρ ρ' g) (f : S' ⟶ S) : IsLin (actPull ρ f) (actPull ρ' f) ((Over.pullback f).map g)

theorem **comp_eq_one_iff_push** {C C' : Over S'} [GrpObj C] [GrpObj C'] {f : S' ⟶ S} {u : C.left ⟶ A.left} {v : C'.left ⟶ A'.left} (hu : IsGrpPullback C A f u) (hv : IsGrpPullback C' A' f v) {g : A ⟶ A'} {g' : C ⟶ C'} (hg : g'.left ≫ v = u ≫ g.left) {T : Over S'} (x : T ⟶ C) : x ≫ g' = 1 ↔ pushHom hu T x ≫ g = 1

theorem **π_lin** (i : Fin (n + 1)) : IsLin ρ (C.act i) (C.π i)

theorem **π_deg** (i : Fin (n + 1)) : IsIsogDeg (C.π i) (d ^ (i : ℕ))

theorem **ker_mono** {T : Over S} (x : T ⟶ A) {i j : Fin (n + 1)} (hij : i ≤ j) (h : x ≫ C.π i = 1) : x ≫ C.π j = 1

theorem **killedBy_of_π** {T : Over S} (x : T ⟶ A) {i : Fin (n + 1)} (h : x ≫ C.π i = 1) : KilledBy ρ 𝔴 x

theorem **killedBy_of_φ** (i : Fin n) {T : Over S} (y : T ⟶ C.B i.castSsucc) (h : y ≫ C.φ i = 1) : KilledBy (C.act _) 𝔴 y

theorem **existsUnique_close** (i : Fin (n + 1)) {a : O} (ha : a ∈ 𝔴) : ∃! g : C.B i ⟶ A, C.π i ≫ g = ρ.act a

theorem **exists_iso_last** {ϖ : O} (h𝔴 : 𝔴 = Ideal.span {ϖ}) {e : ℕ} (hϖ : IsIsogDeg (ρ.act ϖ) e) : ∃ g : C.B (Fin.last n) ≅ A, C.π (Fin.last n) ≫ g.hom = ρ.act ϖ

theorem **φ_comm** (m : ChainMap C' C f u) (i : Fin n) : (C'.φ i).left ≫ m.ψ i.succ = m.ψ i.castSucc ≫ (C.φ i).left

theorem **Chain.exists_pull** {A' : Over S'} [GrpObj A'] [IsCommMonObj A'] {ρ' : RingAct A' O} {ρ : RingAct A O} {𝔴 : Ideal O} {d n : ℕ} (C : Chain ρ 𝔴 d n) {f : S' ⟶ S} {u : A'.left ⟶ A.left} (hu : IsGrpPullback A' A f u) (hact : ∀ a, (ρ'.act a).left ≫ u = u ≫ (ρ.act a).left) : ∃ C' : Chain ρ' 𝔴 d n, Nonempty (ChainMap C' C f u)

Development.AbelianVarietyTheory.Definitions.Divisors.CompactBIffCompact

theorem **Datum.compactB_iff_compact** (D : Datum p) : D.CompactB ↔ D.Compact

Development.AbelianVarietyTheory.Definitions.Divisors.ComparisonMemCechHodgeFiltration

theorem **connForm_add** {B : Type} [CommRing B] [Algebra R B] (f g : Z1 (O (dp0 R) s)) (x : Sp B ⟶ X) (hx : x ≫ s = spMap (algebraMap R B)) : connForm (f + g) x hx = connForm f x hx + connForm g x hx

theorem **connForm_smul** {B : Type} [CommRing B] [Algebra R B] (r : R) (f : Z1 (O (dp0 R) s)) (x : Sp B ⟶ X) (hx : x ≫ s = spMap (algebraMap R B)) : connForm (r • f) x hx = r • connForm f x hx

theorem **connForm_algHom** {B B' : Type} [CommRing B] [Algebra R B] [CommRing B'] [Algebra R B'] (α : B →ₐ[R] B') (f : Z1 (O (dp0 R) s)) (x : Sp B ⟶ X) (hx : x ≫ s = spMap (algebraMap R B)) (x' : Sp B' ⟶ X) (hx' : x' ≫ s = spMap (algebraMap R B')) (e : spMap (α : B →+* B') ≫ x = x') : connForm f x' hx' = (SemiHom.of_algHom α).m1 (connForm f x hx) + D R B' (ev f (Thick.trivHom (hx

theorem **fst_semiDiag** (c : TrivSqZeroExt B Ω[B⁄R]) : (semiDiag ha c).fst = α c.fst

theorem **snd_semiDiag** (c : TrivSqZeroExt B Ω[B⁄R]) : (semiDiag ha c).snd = ha.m1 c.snd

theorem **connForm_pull** (f : Z1 (O (dp0 R) s)) : connForm (h.coef.pullbackOneCocycle f) x' hx' = ha.m1 (connForm f x hx) + D R' B' (ev f (semiTriv h ha hx hx' e))

theorem **pullHyp1_ptCmp** {C' : Type u'} [Category.{v'} C'] {F' : C' ⥤ CommAlgCat.{0} R'} (x' : ∀ T : C', Sp (F'.obj T) ⟶ X') (hx' : ∀ T, x' T ≫ s' = spMap (algebraMap R' (F'.obj T))) (hc' : ∀ ⦃T T' : C'⦄ (a : T ⟶ T'), spMap (mp F' a : F'.obj T →+* F'.obj T') ≫ x' T = x' T') (h : IsMor (dp0 R) s (dp0 R') s' σ φ) (m : CoefMap σ F F') (e : ∀ T, spMap (m.α T) ≫ x (m.G.obj T) = x' T ≫ φ) (c : FirstDeRham s) : pullHyp1 m (ptCmp x hx hc c) = ptCmp x' hx' hc' (h.pull1 c)

theorem **Sec.res_res** (s : X ⟶ Sp R) {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (c : Sec s U) : Sec.res s (h'.trans h) c = Sec.res s h' (Sec.res s h c)

theorem **cechHodgeFiltration_eq_range** (s : X ⟶ Sp R) : CechHodgeFiltration s = LinearMap.range (clForm s)

theorem **affPt_w** (s : X ⟶ Sp R) (U : X.affineOpens) : affPt s U ≫ s = spMap (algebraMap R (Sec s U.1))

theorem **affPt_res** (s : X ⟶ Sp R) {U V : X.affineOpens} (h : V.1 ≤ U.1) : spMap (Sec.res s h : Sec s U.1 →+* Sec s V.1) ≫ affPt s U = affPt s V

theorem **cmpZZ_ω** (s : X ⟶ Sp R) (f : Z1 (O (dp0 R) s)) (U : AffOp X) : (cmpZZ s f).1.2 U = connForm f (affPt s U.unop) (affPt_w s U.unop)

theorem **cmpZZ_f** (s : X ⟶ Sp R) (f : Z1 (O (dp0 R) s)) {U V : AffOp X} (a : U ⟶ V) : (cmpZZ s f).1.1 U V a = ev f (Thick.trivHom (hx

theorem **comparison_mk** (s : X ⟶ Sp R) (f : Z1 (O (dp0 R) s)) : cmp1 s (H1.mk _ f) = Hyp1.mk _ (cmpZZ s f)

theorem **structureSheaf_comparison_comm** (s : X ⟶ Sp R) (c : FirstDeRham s) : toHO1 s (cmp1 s c) = cmpO s (toH1O s c)

theorem **comparison_mem_cechHodgeFiltration** (s : X ⟶ Sp R) {c : FirstDeRham s} (hc : c ∈ F1 s) : cmp1 s c ∈ CechHodgeFiltration s

theorem **cmpForm_apply** (s : X ⟶ Sp R) (f : Form s) (U : AffOp X) : (cmpForm s f).1.1 U = connForm f.1 (affPt s U.unop) (affPt_w s U.unop)

theorem **comparison_formClass** (s : X ⟶ Sp R) (f : Form s) : cmp1 s (Form.cl s f) = clForm s (cmpForm s f)

theorem **Pair.le_fst** {φ : X' ⟶ X} {p q : Pair φ} (h : p ≤ q) : p.1.1 ≤ q.1.1

theorem **Pair.le_snd** {φ : X' ⟶ X} {p q : Pair φ} (h : p ≤ q) : p.1.2 ≤ q.1.2

theorem **secPull_affPt** (s : X ⟶ Sp R) (s' : X' ⟶ Sp R') (p : Pair φ) : spMap (secPull s s' p) ≫ affPt s p.1.2 = affPt s' p.1.1 ≫ φ

theorem **semiHom_secPull** (h : φ ≫ s = s' ≫ spMap σ) (p : Pair φ) : SemiHom σ (secPull s s' p)

theorem **isPullForm_iff** {ω : GlobForm s} {ω' : GlobForm s'} : IsPullForm h ω ω' ↔ ∀ p : Pair φ, (semiHom_secPull h p).m1 (ω.1 (op p.1.2)) = ω'.1 (op p.1.1)

theorem **IsPull1.zero** : IsPull1 h 0 0

theorem **IsPull1.add** {a b : HH1 s} {a' b' : HH1 s'} (ha : IsPull1 h a a') (hb : IsPull1 h b b') : IsPull1 h (a + b) (a' + b')

theorem **IsPull1.smul** {a : HH1 s} {a' : HH1 s'} (ha : IsPull1 h a a') (r : R) : IsPull1 h (r • a) (σ r • a')

theorem **IsPull2.zero** : IsPull2 h 0 0

theorem **IsPull2.add** {a b : HH2 s} {a' b' : HH2 s'} (ha : IsPull2 h a a') (hb : IsPull2 h b b') : IsPull2 h (a + b) (a' + b')

theorem **IsPull2.smul** {a : HH2 s} {a' : HH2 s'} (ha : IsPull2 h a a') (r : R) : IsPull2 h (r • a) (σ r • a')

theorem **IsPull1.cup** {a b : HH1 s} {a' b' : HH1 s'} (ha : IsPull1 h a a') (hb : IsPull1 h b b') : IsPull2 h (cupHH s a b) (cupHH s' a' b')

theorem **IsPull1.toO** {a : HH1 s} {a' : HH1 s'} (ha : IsPull1 h a a') : IsPullO h (toHO1 s a) (toHO1 s' a')

theorem **IsPullForm.cl** {ω : ClosedForm s} {ω' : ClosedForm s'} (e : IsPullForm h ω.1 ω'.1) : IsPull1 h (clForm s ω) (clForm s' ω')

theorem **comparison_pullback** (h : IsMor (dp0 R) s (dp0 R') s' σ φ) (c : FirstDeRham s) : IsPull1 h.w (cmp1 s c) (cmp1 s' (h.pull1 c))

theorem **isPull1_id** (h : 𝟙 X ≫ s = s ≫ spMap (RingHom.id R)) (c : HH1 s) : IsPull1 h c c

theorem **isPull2_id** (h : 𝟙 X ≫ s = s ≫ spMap (RingHom.id R)) (c : HH2 s) : IsPull2 h c c

theorem **affPt_eq** (U : (Sp B).affineOpens) : affPt (algStr R B) U = Spec.map (secMap (𝟙 (Sp B)) U.1)

theorem **res_toSec** ⦃U V : AffOp (Sp B)⦄ (a : U ⟶ V) (b : B) : mp (OAff (algStr R B)) a (toSec R B U.unop b) = toSec R B V.unop b

theorem **affEquiv_mk** (ω : LinearMap.ker (dOne R B)) : affEquiv R B (Submodule.Quotient.mk ω) = clForm _ (coneForm (F

theorem **dOne_line** (ω : Ω[R[X]⁄R]) : dOne R R[X] ω = 0

theorem **lineEquiv_mk** (ω : Ω[R[X]⁄R]) : lineEquiv R (Submodule.Quotient.mk ω) = affEquiv R R[X] (Submodule.Quotient.mk ⟨ω, dOne_line R ω⟩)

theorem **connForm_tdt** (x : Sp B ⟶ Sp R[X]) (hx : x ≫ lineStr R = spMap (algebraMap R B)) : connForm (tdt R).1 x hx = (Spec.preimage x).hom Polynomial.X • D R B ((Spec.preimage x).hom Polynomial.X)

theorem **comparison_tdt_line** : cmp1 (lineStr R) (Form.cl _ (tdt R)) = lineEquiv R (Submodule.Quotient.mk ((Polynomial.X : R[X]) • D R R[X] Polynomial.X))

Development.AbelianVarietyTheory.Definitions.Divisors.CongModRefl

theorem **Pull.fibConst** {d : DeltaDat Xo c 0} {f : Rul Xo c} {g : Rul d.Xo₂ d.c₂} (h : Pull d f g) : FibConst d g

Development.AbelianVarietyTheory.Definitions.Divisors.ConstantUniqueOfEqProduct

theorem **hecke_pow** {η : (AE D)ˣ →* Circle} (hη : IsHeckeChar D η) (n : ℕ) : IsHeckeChar D (η ^ n)

theorem **res_pow** {η : (AE D)ˣ →* Circle} (hη : ResEta D η 1) (n : ℕ) : ResEta D (η ^ n) n

theorem **exp_pow** {η : (AE D)ˣ →* Circle} {τ : D.what₂.Tau} (hη : HasInfExp D η τ (1 / 2)) (n : ℕ) : HasInfExp D (η ^ n) τ ((n : ℚ) / 2)

theorem **unramAt_of_everywhere** {η : (AE D)ˣ →* Circle} (hη : UnramEverywhere D η) {v : HeightOneSpectrum (𝒪 D.host.F)} {ε : AF D.host.F} (hε : IsPlaceIdem D v ε) : UnramAt ε η

theorem **isPair_pow** {η : (AE D)ˣ →* Circle} (hη : IsEtaTilde D η) {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} (hθ : IsInfFam θ) (k : ℕ) (W : HSp D) (hW : W.SigOK) (om : GA D (Examples₂.posSpace (E

theorem **exists_isPair** (hη : ∃ η, IsEtaTilde D η) (hex : ∀ ψ : AddChar (AF D.host.F) ℂ, IsStdChar D ψ (-1) → ∀ (S T : HSp D) (χS χT : (AE D)ˣ →* Circle), IsHeckeChar D χS → IsHeckeChar D χT → ResEta D χS S.rank → ResEta D χT T.rank → ∃ om, IsWeilDatumO ψ S T χS χT om) (k : ℕ) (W : HSp D) (hW : W.SigOK) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (V : HSp D) (χV χW : (AE D)ˣ →* Circle) (om : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)), V.rank = 2 * k ∧ IsPair ψ θ V W χV χW om

theorem **markAd_one** {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} (l : Level H) (W : HSp D) (ι : Dw.La →ₗ[𝒪 D.host.E] (Fin W.rank → D.host.E)) : MarkAd l W rfl ι 1 ι

theorem **markAd_of_markBy** {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {r : ℕ} {c : Coeff l} {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (V : HSp D) (hr : V.rank = r) (dict : GA D V → Σ k, Frame (U.X k).Ψ r) (ι : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) {g : ℕ} {lab : Fin g → D.host.E →+* ℂ} (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) lab) (ik : U.Comp → Fin F.n) (φ : ∀ k, (F.Di (ik k)).La ≃ₗ[𝒪 D.host.E] (U.X k).Λ) (h γ : GA D V) (hm : MarkBy U V hr dict ι F ik φ h γ) : MarkAd l V hr ι (γ * h) ((U.X (dict h).1).Λ.subtype ∘ₗ (φ (dict h).1).toLinearMap ∘ₗ F.t (ik (dict h).1))

theorem **finite_levQc** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} (Dw : Datum H) (l : Level H) : Finite (LevQc Dw l)

theorem **hatC_phiHat** {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Level H} (c : Coeff l) {n : ℕ} (φ : (Fin n → LevQc Dw l) → c.F) (q : Fin n → LevQ Dw l) : inC c (phiHat c φ q) = hatC l (fun x => inC c (φ x)) q

theorem **row_identity** {O : OMDat C R i Tp tup} {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ : (R.MW i (sg0 D A)).Vec} {cc : (C.U ((i : ℕ) + 1) O.s0).Comp → ℂ} {ph : ℂ} (hs : SerOf O κ v₀ cc ph) {ι x₀} (hp : O.Pins κ v₀ ι x₀) (t : Tp) (x : O.Cl' t → VsV D i O.c'.F) (hx : O.Plc t x) (f) (hf : f ∈ sec (C.Q.Xo ((i : ℕ) + 1)) O.c' (wtA κ) (Vc D i).rank) (hth : IsThClsD (C.Q.Xo ((i : ℕ) + 1)) O.c' (O.dδ t) (O.w t) (O.La t) (O.Nc t) (O.gam t) (O.vol t) (O.ep t) O.io x f) (q : AutQ (Wc D i)) : ph * O.thAd t x q = 1 * cc (O.dict q.out).1 * (lieFac (C.Q.Dw ((i : ℕ) + 1)) (xiR D) (wtA κ) : ℂ) * jFn (Wc D i) (inC O.c') (rdU (unfBC O.hle (C.U ((i : ℕ) + 1)) (inC O.c'))) O.dict O.mu f q

theorem **phase_eq_inv_mul_product** {p a c l m r : ℂ} (h0 : p ≠ 0) (h : p * a = c * l * m * r) : 1 * a = p⁻¹ * c * l * m * r

theorem **constant_unique_of_eq_product** {p a c c' l m r : ℂ} (h0 : p ≠ 0) (ha : a ≠ 0) (h : p * a = c * l * m * r) (h' : p * a = c' * l * m * r) : c = c'

theorem **phase_one** {O : OMDat C R i Tp tup} {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ : (R.MW i (sg0 D A)).Vec} {cc : (C.U ((i : ℕ) + 1) O.s0).Comp → ℂ} {ph : ℂ} (hs : SerOf O κ v₀ cc ph) : SerOf O κ v₀ (fun k => ph⁻¹ * cc k) 1

theorem **const_unique** {O : OMDat C R i Tp tup} {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ : (R.MW i (sg0 D A)).Vec} {cc cc' : (C.U ((i : ℕ) + 1) O.s0).Comp → ℂ} {ph : ℂ} (hs : SerOf O κ v₀ cc ph) (hs' : SerOf O κ v₀ cc' ph) {t : Tp} {h J'} (h1 : RepsR R i (tup t) h (O.bad t) J' (O.Lh t) (O.gam t) (O.vol t)) {y x} (h2 : Place R i (tup t) h O.c0' y x) {f} (hf : f ∈ sec (C.Q.Xo ((i : ℕ) + 1)) O.c' (wtA κ) (Vc D i).rank) (h3 : IsThClsD (C.Q.Xo ((i : ℕ) + 1)) O.c' (O.dδ t) (O.w t) (O.La t) (O.Nc t) (O.gam t) (O.vol t) (O.ep t) O.io x f) {g : GA D (Wc D i)} (hg : R.thF i (tup t) y v₀ g ≠ 0) : cc (O.dict g).1 = cc' (O.dict g).1

Development.AbelianVarietyTheory.Definitions.Divisors.ContinuousBcPt

theorem **bcPt_val** (w : j ≫ p = q ≫ spMap ρ) (τ : K →+* ℂ) (z : CxPts q τ) : (bcPt w τ z).1 = z.1 ≫ j

theorem **continuous_bcPt** (w : j ≫ p = q ≫ spMap ρ) (τ : K →+* ℂ) : Continuous (bcPt w τ)

theorem **cInfAt_bcPt** (w : j ≫ p = q ≫ spMap ρ) (τ : K →+* ℂ) {f : CxPts p (τ.comp ρ) → ℝ} {z : CxPts q τ} (h : CInfAt f (bcPt w τ z)) : CInfAt (fun y => f (bcPt w τ y)) z

Development.AbelianVarietyTheory.Definitions.Divisors.ContinuousLvMat

theorem **cT_add** {p q : Type} (X Y : Matrix p q A) : cT s (X + Y) = cT s X + cT s Y

theorem **cT_sub** {p q : Type} (X Y : Matrix p q A) : cT s (X − Y) = cT s X − cT s Y

theorem **cT_cT** (h : ∀ x, s (s x) = x) {p q : Type} (X : Matrix p q A) : cT s (cT s X) = X

theorem **carry_frame** (g : Gram s J C D Z J₀) (hi : Ji ∗ J₀ = 1) : carry s J C D Z Ji C' D' Z' ∗ C = C' ∧ carry s J C D Z Ji C' D' Z' ∗ Z = Z' ∧ carry s J C D Z Ji C' D' Z' ∗ D = D'

theorem **rext** (hs : carry s J C D Z Ji C D Z = 1) {p : ℕ} {X Y : Matrix (Fin p) (Fin n) A} (h1 : X ∗ C = Y ∗ C) (h2 : X ∗ Z = Y ∗ Z) (h3 : X ∗ D = Y ∗ D) : X = Y

theorem **lext** (hs : carry s J C D Z Ji C D Z = 1) {p : ℕ} {X Y : Matrix (Fin n) (Fin p) A} (h1 : cT s C ∗ X = cT s C ∗ Y) (h2 : cT s Z ∗ X = cT s Z ∗ Y) (h3 : cT s D ∗ X = cT s D ∗ Y) : X = Y

theorem **unitary** (g : Gram s J C D Z J₀) (g' : Gram s J C' D' Z' J₀) (hs : carry s J C D Z Ji C D Z = 1) {M : Matrix (Fin n) (Fin n) A} (e1 : M ∗ C = C') (e2 : M ∗ Z = Z') (e3 : M ∗ D = D') : cT s M ∗ J ∗ M = J

theorem **gram_of_unitary** (g : Gram s J C D Z J₀) {M : Matrix (Fin n) (Fin n) A} (hM : cT s M ∗ J ∗ M = J) : Gram s J (M ∗ C) (M ∗ D) (M ∗ Z) J₀

theorem **gram_levi** (hs : ∀ x, s (s x) = x) (g : Gram s J C D Z J₀) (α : GL (Fin j) A) : Gram s J (C ∗ α.1) (D ∗ cT s (α⁻¹).1) Z J₀

theorem **gram_unip_iff** (hs : ∀ x, s (s x) = x) (g : Gram s J C D Z J₀) (h₀ : cT s Ji ∗ J₀ = 1) (hi : J₀ ∗ Ji = 1) (b : Matrix (Fin j) (Fin m) A) (c : Matrix (Fin j) (Fin j) A) : Gram s J C (D − Z ∗ (Ji ∗ cT s b) + C ∗ c) (Z + C ∗ b) J₀ ↔ c + cT s c + b ∗ (Ji ∗ cT s b) = 0

theorem **ad_cT** {p q : Type} (X : Matrix p q E) : adMat F (cT (σ : E →+* E) X) = cT (sigR (AF F) σ) (adMat F X)

theorem **sig_sig** (h : IsCMExt F E σ) (x : AF F ⊗[F] E) : sigR (AF F) σ (sigR (AF F) σ x) = x

theorem **gram_rat** (f : Fr σ J C D Z J₀) : Gram (σ : E →+* E) J C D Z J₀

theorem **gram_ad** (f : Fr σ J C D Z J₀) : Gram (sigR (AF F) σ) (JR (AF F) J) (adMat F C) (adMat F D) (adMat F Z) (adMat F J₀)

theorem **ad_carry** (Ji : Matrix (Fin m) (Fin m) E) (C' D' : Matrix (Fin n) (Fin j) E) (Z' : Matrix (Fin n) (Fin m) E) : adMat F (carry (σ : E →+* E) J C D Z Ji C' D' Z') = carry (sigR (AF F) σ) (JR (AF F) J) (adMat F C) (adMat F D) (adMat F Z) (adMat F Ji) (adMat F C') (adMat F D') (adMat F Z')

theorem **sum_ad** (f : Fr σ J C D Z J₀) : carry (sigR (AF F) σ) (JR (AF F) J) (adMat F C) (adMat F D) (adMat F Z) (adMat F J₀⁻¹) (adMat F C) (adMat F D) (adMat F Z) = 1

theorem **inv_mul₀** (f : Fr σ J C D Z J₀) : J₀⁻¹ ∗ J₀ = 1

theorem **mul_inv₀** (f : Fr σ J C D Z J₀) : J₀ ∗ J₀⁻¹ = 1

theorem **inv_mul_ad** (f : Fr σ J C D Z J₀) : adMat F J₀⁻¹ ∗ adMat F J₀ = 1

theorem **mul_inv_ad** (f : Fr σ J C D Z J₀) : adMat F J₀ ∗ adMat F J₀⁻¹ = 1

theorem **cT_inv_ad** (f : Fr σ J C D Z J₀) : cT (sigR (AF F) σ) (adMat F J₀⁻¹) ∗ adMat F J₀ = 1

theorem **lvMat_frame** (f : Fr σ J C D Z J₀) (α : GL (Fin j) (AF F ⊗[F] E)) : lvMat f α ∗ adMat F C = adMat F C ∗ α.1 ∧ lvMat f α ∗ adMat F Z = adMat F Z ∧ lvMat f α ∗ adMat F D = adMat F D ∗ cT (sigR (AF F) σ) (α⁻¹).1

theorem **continuous_lvMat** (f : Fr σ J C D Z J₀) : Continuous (lvMat f)

theorem **isLv_lv** (f : Fr σ J C D Z J₀) : IsLv σ J C D Z (lv f)

theorem **isLv_unique** (f : Fr σ J C D Z J₀) {l l' : GL (Fin j) (AF F ⊗[F] E) →* GAd σ J} (h : IsLv σ J C D Z l) (h' : IsLv σ J C D Z l') : l = l'

theorem **isLv_injective** (f : Fr σ J C D Z J₀) {l : GL (Fin j) (AF F ⊗[F] E) →* GAd σ J} (h : IsLv σ J C D Z l) : Function.Injective l

theorem **exists_crd_iff** (f : Fr σ J C D Z J₀) (b : Matrix (Fin j) (Fin m) (AF F ⊗[F] E)) (c : Matrix (Fin j) (Fin j) (AF F ⊗[F] E)) : (∃ u, Crd σ J C D Z J₀ u b c) ↔ c + cT (sigR (AF F) σ) c + b ∗ plus σ J₀ b = 0

theorem **crd_unique** (f : Fr σ J C D Z J₀) {u u' : GAd σ J} {b b' : Matrix (Fin j) (Fin m) (AF F ⊗[F] E)} {c c' : Matrix (Fin j) (Fin j) (AF F ⊗[F] E)} (h : Crd σ J C D Z J₀ u b c) (h' : Crd σ J C D Z J₀ u' b' c') : u = u' ↔ b = b' ∧ c = c'

theorem **bse_eq** (h : IsCMExt F E σ) : (GlobalTheory.Doubling.absE F E (GlobalTheory.Eis.Conv.halfE F E) : ℝ) = (bse F)⁻¹

theorem **fr_three** (h : IsCMExt F E σ) : Fr σ !![0, 0, 1; 0, 1, 0; 1, 0, 0] (!![1; 0; 0] : Matrix (Fin 3) (Fin 1) E) !![0; 0; 1] (!![0; 1; 0] : Matrix (Fin 3) (Fin 1) E) 1

theorem **inhabited** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E) (J : Matrix (Fin 3) (Fin 3) E) (C D Z : Matrix (Fin 3) (Fin 1) E) (J₀ : Matrix (Fin 1) (Fin 1) E) (l : GL (Fin 1) (AF F ⊗[F] E) →* GAd σ J) (u : GAd σ J) (c : Matrix (Fin 1) (Fin 1) (AF F ⊗[F] E)), Fr σ J C D Z J₀ ∧ IsLv σ J C D Z l ∧ Function.Injective l ∧ Crd σ J C D Z J₀ u 1 c ∧ u ∈ fixG σ J C

Development.AbelianVarietyTheory.Definitions.Divisors.CovolXOfCovol

theorem **core_eq_FO** : p.T.Q.c.F = FO D

theorem **RecTied.core_size** (h : RecTied D L) : fieldSz p.T.Q.c.F ≤ 2 ∗ Formalism.MainNotions.WeightAlgebra.PO D L

theorem **two_PO_le** (D : D7) (L : SizeData₂ D) (hL : 2 ≤ L.sc.L) : 2 ∗ Formalism.MainNotions.WeightAlgebra.PO D L ≤ 217 ∗ zBase L ^ 2

theorem **RecTied.core_size_le** (h : RecTied D L) (hL : 2 ≤ L.sc.L) : fieldSz p.T.Q.c.F ≤ 217 ∗ zBase L ^ 2

theorem **goodCov_iff** : GoodCov p i Y ↔ Y.GoodL L.sc.L (391 ∗ zBase L ^ 2)

theorem **evTied_iff** {eps : ℝ} (Fm : FamP eps) (Q : ∀ (x : ℝ) (hx : Fm.x1 < x) (ι : Fm.Ix x), Pt Fm x hx ι → Prop) : EvTied Fm Q ↔ EvPt Fm fun x hx ι p => RecTied (Fm.D x ι) (Fm.L x ι) → Q x hx ι p

theorem **EvTied.and** {eps : ℝ} {Fm : FamP eps} {Q Q' : ∀ (x : ℝ) (hx : Fm.x1 < x) (ι : Fm.Ix x), Pt Fm x hx ι → Prop} (h : EvTied Fm Q) (h' : EvTied Fm Q') : EvTied Fm fun x hx ι p => Q x hx ι p ∧ Q' x hx ι p

theorem **CovolX.covol** (h : CovolX) (ht : Closed FamHyp fun Fm _ => EvPt Fm fun x _ ι _ => RecTied (Fm.D x ι) (Fm.L x ι)) : Covol

Development.AbelianVarietyTheory.Definitions.Divisors.CrystallineConnectionFormCoboundaryZero

@[simp] theorem **fst_derSec** : (derSec D b).fst = b

@[simp] theorem **snd_derSec** : (derSec D b).snd = D b

theorem **isCentralScalar_kaehler** : IsCentralScalar B Ω[B⁄R]

theorem **isScalarTower_kaehler** : IsScalarTower R Bᵐᵒᵖ Ω[B⁄R]

theorem **fst_diagMap** [Algebra B B'] [IsScalarTower R B B'] (c : TrivSqZeroExt B Ω[B⁄R]) : (diagMap R B B' c).fst = algebraMap B B' c.fst

theorem **snd_diagMap** [Algebra B B'] [IsScalarTower R B B'] (c : TrivSqZeroExt B Ω[B⁄R]) : (diagMap R B B' c).snd = KaehlerDifferential.map R R B B' c.snd

theorem **crystallineConnectionForm_coboundaryZero** (g : C0 (O (dp0 R) s)) (x : Sp B ⟶ X) (hx : x ≫ s = spMap (algebraMap R B)) : connForm (d0 _ g, commAlgFunctorCoboundaryZero_mem_cocyclesOne _ g) x hx = D R B (g (Thick.triv B x hx))

theorem **connForm_map** [Algebra B B'] [IsScalarTower R B B'] (f : Z1 (O (dp0 R) s)) (x : Sp B ⟶ X) (hx : x ≫ s = spMap (algebraMap R B)) (x' : Sp B' ⟶ X) (hx' : x' ≫ s = spMap (algebraMap R B')) (e : spMap (algebraMap B B') ≫ x = x') : connForm f x' hx' = KaehlerDifferential.map R R B B' (connForm f x hx) = D R B' (ev f (Thick.trivHom (hx

theorem **Thick.Hom.le_preimage** {T T' : Thick γ s} (φ : T ⟶ T') {U : X.Opens} (h : ⊤ ≤ T.x ⁻¹ᵘ U) : ⊤ ≤ T'.x ⁻¹ᵘ U

theorem **isChern_zero** (L : LB X) (h : ∀ i j, L.g i j = 1) : IsChern γ s L 0

theorem **firstDeRham_base_eq_zero** (s : X ⟶ Sp R) [IsIso s] (c : FirstDeRham s) : c = 0

theorem **π_coordLift** (T : Thick (dp0 R) (lineStr R)) : T.π (coordLift T) = (Spec.preimage T.x).hom Polynomial.X

theorem **coordLift_sub_mem** {T T' : Thick (dp0 R) (lineStr R)} (φ : T ⟶ T') : coordLift T' - φ.φ (coordLift T) ∈ T'.J

theorem **dpow_two_add** {B : Type} [CommRing B] {J : Ideal B} (δ : DividedPowers J) {a b : B} (ha : a ∈ J) (hb : b ∈ J) : δ.dpow 2 (a + b) = δ.dpow 2 a + a * b + δ.dpow 2 b

theorem **cl_tdt_ne_zero** [Nontrivial R] (h2 : (2 : R) = 0) : Form.cl _ (tdt R) ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.CurvSubQuotTautPairGram

theorem **blockSS_eq** (K T : Matrix (m ⊕ l) (m ⊕ l) ℂ) : blockSS K T = T.toBlocks₁₁ + orthCoeff K * T.toBlocks₂₁

theorem **blockQQ_eq** (K T : Matrix (m ⊕ l) (m ⊕ l) ℂ) : blockQQ K T = T.toBlocks₂₂ - T.toBlocks₂₁ * orthCoeff K

theorem **conn_sub** (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (ξ : E) : conn (fun x => subGram (H x)) z ξ = blockSS (H z) (conn H z ξ)

theorem **mContDiffAt_orthCoeff** {k : WithTop ℕ∞} (h : MContDiffAt k H z) (hs : IsUnit (subGram (H z)).det) : MContDiffAt k (fun x => orthCoeff (H x)) z

theorem **mdbar_orthCoeff** (h : MContDiffAt 1 H z) (hh : ∀ᶠ x in 𝒩 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (η : E) : mdbar (fun x => orthCoeff (H x)) z η = secondFormAdj H z η

theorem **curv_sub** (h : MContDiffAt 2 H z) (hh : ∀ᶠ x in 𝒩 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (ξ η : E) : curv (fun x => subGram (H x)) z ξ η = blockSS (H z) (curv H z ξ η) - secondFormAdj H z η * secondForm H z ξ

theorem **conn_quot** (h : MContDiffAt 1 H z) (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (ξ : E) : conn (fun x => quotGram (H x)) z ξ = blockQQ (H z) (conn H z ξ)

theorem **curv_quot** (h : MContDiffAt 2 H z) (hh : ∀ᶠ x in 𝒩 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (ξ η : E) : curv (fun x => quotGram (H x)) z ξ η = blockQQ (H z) (curv H z ξ η) + secondForm H z ξ * secondFormAdj H z η

theorem **tautPairGram_eq** (z : ℂ) : tautPairGram z = (pairFrame z)ᵀ * (fun _ : ℂ => 1) z * (pairFrame z)ᴴᵀ

theorem **mContDiffAt_pairFrame** (k : WithTop ℕ∞) (z : ℂ) : MContDiffAt k pairFrame z

theorem **det_pairFrame** (z : ℂ) : (pairFrame z).det = 1

theorem **isPosFrame_tautPairGram** : IsPosFrame Set.univ tautPairGram

theorem **curv_tautPairGram** (z ξ η : ℂ) : curv tautPairGram z ξ η = 0

theorem **secondForm_tautPairGram** (ξ : ℂ) : secondForm tautPairGram 0 ξ = of fun _ _ => ξ

theorem **secondFormAdj_tautPairGram** (η : ℂ) : secondFormAdj tautPairGram 0 η = of fun _ _ => conj η

theorem **curv_sub_quot_tautPairGram** (ξ η : ℂ) : curv (fun x => subGram (tautPairGram x)) 0 ξ η () () = -(ξ * conj η) ∧ curv (fun x => quotGram (tautPairGram x)) 0 ξ η () () = ξ * conj η ∧ extTrace tautPairGram 0 ξ η = kappa * (ξ * conj η)

Development.AbelianVarietyTheory.Definitions.Divisors.CuspIso

theorem **GapGe.mono** {g g' : ℤ} {la : AllWt.Weight Dw} (h : GapGe g la) (e : g' ≤ g) : GapGe g' la

Development.AbelianVarietyTheory.Definitions.Divisors.CxTotalEqFullSumIrrOfCxCountReg

theorem **fullSum_eq_fullSumIrr_of_term_zero** (G : HTGroup p) (wt : Polynomial G.F → ℂ) (Kf : Subgroup G.GAf) (f : G.GAf → ℂ) (r : ℕ) (K : Subgroup (Gr G r)) (φ : Gr G r → ℂ) (h : ∀ P : Pair G, (∃ μ, P.Mark μ) → ¬ Irreducible P.pol → wt P.pol * P.term Kf f r K φ = 0) : fullSum G wt Kf f r K φ = fullSumIrr G wt Kf f r K φ

Development.AbelianVarietyTheory.Definitions.Divisors.DatumOfIsIntegersLattice

theorem **stable_empty** (L : PeriodPair) (F : IntermediateField ℚ ℂ) (T : Empty → ℂ) : Stable L F T

theorem **stable_of_isRat** {L : PeriodPair} {F : IntermediateField ℚ ℂ} {ι : Type} {T : ι → ℂ} (h : ∀ i, GlobalTheory.ConjPoint.IsRat L F (T i)) : Stable L F T

theorem **stable_single** (L : PeriodPair) (t : ℂ) : Stable L ℚ(L.weierstrassP t, L.derivWeierstrassP t) (fun _ : Unit => t)

Development.AbelianVarietyTheory.Definitions.Divisors.DensConst

theorem **dens_const** (N : MLine p σ) (c : AlgChart p σ D) (y : CxPts p σ) : dens (fun _ => N) c y = c.chernDens N.φ y

theorem **part_const** (N : MLine p σ) (ε : ℤˣ) : part (fun _ : Fin D => N) ε = N.φ.chernPart D ε

theorem **continuous_gΦ** : Continuous (gΦ t)

theorem **cInfAt_gΦ** {f : CxPts t (Int.castRingHom ℂ) → ℝ} {z : CxPts (gq t) (Rat.castHom ℂ)} (h : CInfAt f (gΦ t z)) : CInfAt (fun w => f (gΦ t w)) z

Development.AbelianVarietyTheory.Definitions.Divisors.DetJacMatCoordsEqTransitionJacobianDet

theorem **compactSpace** (h : IsProjOver p) : CompactSpace (CxPts p σ)

theorem **isOpen_ptsIn** (U : Y.Opens) : IsOpen (ptsIn p σ U)

theorem **continuousOn_ev** (f : Γ(Y, U)) : ContinuousOn (ev p σ U f) (ptsIn p σ U)

theorem **continuous_iff** {B : Type*} [TopologicalSpace B] {e : B → CxPts p σ} : Continuous e ↔ ∀ U : Y.Opens, IsOpen (e ⁻¹' ptsIn p σ U) ∧ ∀ f : Γ(Y, U), ContinuousOn (ev p σ U f ∘ e) (e ⁻¹' ptsIn p σ U)

theorem **mem_ptsIn** : y ∈ ptsIn p σ U ↔ y.1 (IsLocalRing.closedPoint ℂ) ∈ U

theorem **ptsIn_mono** (h : W ≤ U) : ptsIn p σ W ⊆ ptsIn p σ U

@[simp] theorem **mem_ptsIn_inf** : y ∈ ptsIn p σ (U ⊓ W) ↔ y ∈ ptsIn p σ U ∧ y ∈ ptsIn p σ W

@[simp] theorem **mem_ptsIn_top** : y ∈ ptsIn p σ ⊤

theorem **exists_mem_ptsIn** {ι : Sort*} {U : ι → Y.Opens} (h : iSup U = ⊤) (y : CxPts p σ) : ∃ i, y ∈ ptsIn p σ (U i)

theorem **ev_eq** (h : y ∈ ptsIn p σ U) (f : Γ(Y, U)) : ev p σ U f y = evHom h f

theorem **evHom_eq_evalPt** (h : y ∈ ptsIn p σ U) : evHom h = TangentOrigin.evalPt y.1 U h

theorem **ev_of_notMem** (h : y ∉ ptsIn p σ U) (f : Γ(Y, U)) : ev p σ U f y = 0

theorem **ev_res** (e : W ≤ U) (h : y ∈ ptsIn p σ W) (f : Γ(Y, U)) : ev p σ W (resG Y e f) y = ev p σ U f y

theorem **ev_mul** (f g : Γ(Y, U)) (y : CxPts p σ) : ev p σ U (f * g) y = ev p σ U f y * ev p σ U g y

theorem **ev_add** (f g : Γ(Y, U)) (y : CxPts p σ) : ev p σ U (f + g) y = ev p σ U f y + ev p σ U g y

theorem **ev_algX** (h : y ∈ ptsIn p σ U) (a : k) : ev p σ U (algX p U a) y = σ a

theorem **ev_ne_zero** (h : y ∈ ptsIn p σ U) {f : Γ(Y, U)} (hf : IsUnit f) : ev p σ U f y ≠ 0

theorem **mem_ptsIn_basicOpen** (h : y ∈ ptsIn p σ U) (f : Γ(Y, U)) : y ∈ ptsIn p σ (Y.basicOpen f) ↔ ev p σ U f y ≠ 0

theorem **ofHom_mem** (hU : IsAffineOpen U) (φ : Γ(Y, U) →+* ℂ) (h : ∀ a, φ (algX p U a) = σ a) : ofHom hU φ h ∈ ptsIn p σ U

theorem **ev_ofHom** (hU : IsAffineOpen U) (φ : Γ(Y, U) →+* ℂ) (h : ∀ a, φ (algX p U a) = σ a) (f : Γ(Y, U)) : ev p σ U f (ofHom hU φ h) = φ f

theorem **holo_of_affine** {E : Type*} [NormedAddCommGroup E] [NormedSpace ℂ E] (hU : IsAffineOpen U) {T : Set E} (hT : IsOpen T) {s : E → CxPts p σ} (hs : ∀ v ∈ T, s v ∈ ptsIn p σ U) (hd : ∀ a : Γ(Y, U), DifferentiableOn ℂ (ev p σ U a ∘ s) T) : ContinuousOn s T ∧ ∀ (W : Y.Opens) (f : Γ(Y, W)), DifferentiableOn ℂ (ev p σ W f ∘ s) (T ∩ s ⁻¹' ptsIn p σ W)

theorem **isFrame** : IsFrame (dOn p c.x)

@[simp] theorem **restr_source** (S : Set (CxPts p σ)) (hS : IsOpen S) : (c.restr S hS).e.source = c.e.source ∩ S

theorem **holo_trans** : DifferentiableOn ℂ (c'.e ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' c'.e.source)

theorem **differentiableAt_ev** {W : Y.Opens} (f : Γ(Y, W)) {y : CxPts p σ} (hy : y ∈ c.e.source) (hW : y ∈ ptsIn p σ W) : DifferentiableAt ℂ (ev p σ W f ∘ c.e.symm) (c.e y)

theorem **exists_forms_map** {W : Y.Opens} {y : CxPts p σ} (hy : y ∈ c.e.source) (hW : y ∈ ptsIn p σ W) : ∃ Λ : FormsOn p D W →ₗ[evHom hW] ℂ, ∀ f, Λ (dOn p f) = (c.jacMat f y).det

theorem **jacMat_res** {W W' : Y.Opens} (h : W' ≤ W) (f : Fin D → Γ(Y, W)) {y : CxPts p σ} (hy : y ∈ c.e.source) (hW : y ∈ ptsIn p σ W') : c.jacMat (fun a => resG Y h (f a)) y = c.jacMat f y

theorem **det_jacMat_coords_eq_transitionJacobianDet** {y : CxPts p σ} (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : (c.jacMat c'.x y).det = transDet c.e c'.e y

theorem **transDet_self** {y : CxPts p σ} (hy : y ∈ c.e.source) : transDet c.e c.e y = 1

theorem **ev_coeff_dOn** {W : Y.Opens} (h : W ≤ c.U) (f : Fin D → Γ(Y, W)) {y : CxPts p σ} (hy : y ∈ c.e.source) (hW : y ∈ ptsIn p σ W) : ev p σ W (c.isFrame.coeff h (dOn p f)) y = (c.jacMat f y).det

theorem **ev_ratio** {y : CxPts p σ} (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : ev p σ _ (c.isFrame.ratio c'.isFrame).val y = transDet c.e c'.e y

theorem **exists_of_etale** {U : Y.Opens} (hU : IsAffineOpen U) (x : Fin D → Γ(Y, U)) (hx : (MvPolynomial.eval₂Hom (algX p U) x).Etale) {y : CxPts p σ} (hy : y ∈ ptsIn p σ U) : ∃ c : AlgChart p σ D, y ∈ c.e.source ∧ ∃ h : c.U = U, c.x = h ▸ x

theorem **exists_etale_coords** [SmoothOfRelativeDimension D p] (z : Y) : ∃ (U : Y.Opens) (x : Fin D → Γ(Y, U)), IsAffineOpen U ∧ z ∈ U ∧ (MvPolynomial.eval₂Hom (algX p U) x).Etale

theorem **exists_mem_source** [SmoothOfRelativeDimension D p] (y : CxPts p σ) : ∃ c : AlgChart p σ D, y ∈ c.e.source

Development.AbelianVarietyTheory.Definitions.Divisors.DetNeZeroOfIsWittBasis

theorem **mem_unipN_iff** (g : UA σ J₂) : g ∈ unipN σ J₂ ↔ (mat g - 1) * blockColumnMatrix (1 : Matrix (Fin n) (Fin n) (AE F E)) 1 = 0

theorem **cocompact_unipN** (hcm : IsCMExt F E σ) {J : Matrix (Fin n) (Fin n) E} {l : E} (hl : l ≠ 0) (hJ : ∀ i j, σ ((l⁻¹ • J) j i) = -(l⁻¹ • J) i j) (hdet : J.det ≠ 0) (h₂ : J₂ = dblGram finSumFinEquiv J) : LocallyCompactSpace (unipN σ J₂) ∧ ∃ C : Set (unipN σ J₂), IsCompact C ∧ ∀ g : unipN σ J₂, ∃ γ : ratN σ J₂, γ • g ∈ C

theorem **existsUnique_isNormN** (hcm : IsCMExt F E σ) {J : Matrix (Fin n) (Fin n) E} {l : E} (hl : l ≠ 0) (hJ : ∀ i j, σ ((l⁻¹ • J) j i) = -(l⁻¹ • J) i j) (hdet : J.det ≠ 0) (h₂ : J₂ = dblGram finSumFinEquiv J) : ∃! νN : Measure (unipN σ J₂), IsNormN σ νN

Development.AbelianVarietyTheory.Definitions.Divisors.DiagHermitianTrivialOfSumFourSquares

private theorem **lkMid_sh_mul_eq_sh_mul_sh_rev** (a b : G.C) : _root_.GlobalTheory.LKMid.sh G (a * b) = _root_.GlobalTheory.LKMid.sh G b * _root_.GlobalTheory.LKMid.sh G a

private theorem **lkMid_sh_mul_sh_units_inv_eq_one** (y : (G.C)ˣ) : _root_.GlobalTheory.LKMid.sh G y * _root_.GlobalTheory.LKMid.sh G ((y⁻¹ : (G.C)ˣ) : G.C) = 1

theorem **_root_.GlobalTheory.LKDef.DPair.rel_equivalence** (G : _root_.GlobalTheory.LKTrip.HTGroup p) : Equivalence (_root_.GlobalTheory.LKDef.DPair.Rel (G

theorem **pol_eq_of_rel** {P Q : DPair G} (h : P.Rel Q) : Q.pol = P.pol

theorem **_root_.GlobalTheory.LKDef.DPair.mk_eq_iff** (P Q : _root_.GlobalTheory.LKDef.DPair G) : Quot.mk (_root_.GlobalTheory.LKDef.DPair.Rel (G

theorem **_root_.GlobalTheory.LKDef.DPair.rel_out** (P : _root_.GlobalTheory.LKDef.DPair G) : P.Rel (Quot.out (Quot.mk (_root_.GlobalTheory.LKDef.DPair.Rel (G

theorem **beta_ne_pair** (P : DPair G) (Q : Pair G) (v : InfinitePlace G.F) (hv : sigPosH (G.gram_herm v) = 1) (y : Fin G.n → ℂ) (hy : (star y ⬝ᵥ (G.gram v *ᵥ y)).re < 0) : P.β ≠ Q.β

theorem **MoveDef.of_isTest** (H : MoveDef) (G : HTGroup p) (ha : G.a = 1) (hn : Odd G.n) (wt sel : Polynomial G.F → ℂ) (hsel : IsSel G sel) (L : G.Level) (f : G.GAf → ℂ) (hf : IsTest (MonoidHom.id _) L.K.toSubgroup L.K.toSubgroup f) (r : ℕ) (hr : 0 < r) (K : Subgroup (Gr G r)) (φ : Gr G r → ℂ) (hφ : IsTest (sigG G r) (KHT p G.m G.n r G.Fw) K φ) : markedSumIrr G wt sel L.K.toSubgroup f r K φ = dSumIrr G wt L.K.toSubgroup f r K φ

theorem **exD_irreducible** : Irreducible exD.pol

theorem **padicInt_four_squares** (l : ℕ) [Fact l.Prime] (z : ℤ_[l]) : ∃ a b c d : ℤ_[l], a * a + b * b + c * c + d * d = z

theorem **finiteAdeleRingAway_sum_four_squares_eq_neg_one** (p : ℕ) : ∃ a b c d : AfP p, a * a + b * b + c * c + d * d = -1

theorem **conj_diagHermitian** : H3.map (IsCMField.complexConj QI : QI →+* QI) = H3

theorem **isHermitian_diagHermitian** : (H3.map (IsCMField.complexConj QI : QI →+* QI))ᵀ = H3

theorem **diagHermitian_mul_self** : H3 * H3 = 1

theorem **det_diagHermitian_ne_zero** : H3.det ≠ 0

theorem **gramMatrix_diagHermitian** (τ : QI →+* ℂ) : gramAt H3 τ = Matrix.diagonal ![1, -1, -1]

theorem **signature_diagHermitian** (τ : QI →+* ℂ) : sigPos isHermitian_diagHermitian τ = 1 ∧ sigNeg isHermitian_diagHermitian τ = 2

theorem **not_cube** (b : QI) : b ^ 3 ≠ 5 * (2 + gi)

theorem **quat_mul** {R S : Type} [CommRing R] [Ring S] [Algebra R S] (u i j k : S) (a a' b c d : R) (ha : a' = -a) (h : a * a + b * b + c * c + d * d = -1) (uu : u * u = u) (ui : u * i = i) (uj : u * j = j) (uk : u * k = k) (iu : i * u = i) (ii : i * i = -u) (ij : i * j = k) (ik : i * k = -j) (ju : j * u = j) (ji : j * i = -k) (jj : j * j = -u) (jk : j * k = i) (ku : k * u = k) (ki : k * i = j) (kj : k * j = -i) (kk : k * k = -u) : (1 + (a - 1) • u + b • i + c • j + d • k) * (1 + (a' - 1) • u + b • i + c • j + d • k) = 1

theorem **adjointInvolution_diagHermitian** (x : Matrix (Fin 3) (Fin 3) QI) : (hermStar H3 det_diagHermitian_ne_zero x).unop = H3 * (x.map (IsCMField.complexConj QI : QI →+* QI))ᵀ * H3

theorem **gi_mul_gi** : gi * gi = -1

theorem **diagHermitian_trivial_of_sumFourSquares** (R : Type) [CommRing R] [Algebra ℚ R] (a b c d : R) (h : a * a + b * b + c * c + d * d = -1) : ∃ x : (R ⊗[ℚ] Matrix (Fin 3) (Fin 3) QI)ˣ, (x : R ⊗[ℚ] Matrix (Fin 3) (Fin 3) QI) * ((1 : R) ⊗ₜ H3) * (starT (hermStar H3 det_diagHermitian_ne_zero) R x).unop = 1

theorem **cubeRootMatrix_mul_conjTranspose** : g3 * (g3.map (IsCMField.complexConj QI : QI →+* QI))ᵀ = (5 : QI) • 1

theorem **cubeRootMatrix_pow_three** : g3 ^ 3 = algebraMap QI _ (5 * (2 + gi))

theorem **det_cubeRootMatrix_ne_zero** : g3.det ≠ 0

theorem **irreducible_cubic** : Irreducible (Polynomial.X ^ 3 - Polynomial.C (5 * (2 + gi)) : Polynomial QI)

theorem **minpoly_cubeRootMatrix** : minpoly QI g3 = Polynomial.X ^ 3 - Polynomial.C (5 * (2 + gi))

theorem **unitaryOneTwo_rank_eq_three** : unitaryOneTwoGaussian.n = 3

theorem **signature_unitaryOneTwo** (v : InfinitePlace unitaryOneTwoGaussian.F) : sigPosH (unitaryOneTwoGaussian.gram_herm v) = 1

theorem **unitaryOneTwo_numSignaturePlaces** : unitaryOneTwoGaussian.a = 1

theorem **gramMatrix_mul_diagHermitian** (τ : QI →+* ℂ) : gramAt H3 τ * H3.map τ = 1

theorem **definitePair_minpoly** : definitePairUnitaryOneTwo.pol = Polynomial.X ^ 3 - Polynomial.C (5 * (2 + gi))

theorem **definitePair_minpoly_irreducible** : Irreducible definitePairUnitaryOneTwo.pol

theorem **definitePair_unit_ne_pairUnit** (Q : Pair unitaryOneTwoGaussian) : definitePairUnitaryOneTwo.β ≠ Q.β

theorem **exists_rank_three** : ∃ G : HTGroup 5, G.n = 3 ∧ G.a = 1 ∧ Odd G.n ∧ (∃ sel, IsSel G sel) ∧ Nonempty G.Level ∧ ∃ q : DCls G, Irreducible (Quot.out q).pol

Development.AbelianVarietyTheory.Definitions.Divisors.DiagValExtTraceNonneg

theorem **chg_submatrix** {m' : Type} [Fintype m'] (A : Matrix n n' ℂ) (H : Matrix n n ℂ) (e : m' ≃ n) {l' : Type} (f : l' → n') : chg (A.submatrix e f) (H.submatrix e e) = (chg A H).submatrix f f

theorem **chg_fromBlocks** : chg (fromBlocks A₁ B 0 A₂) H = fromBlocks (chg A₁ H.toBlocks₁₁) (A₁ᵀ * (H.toBlocks₁₁ * Bᴴᵀ + H.toBlocks₁₂ * A₂ᴴᵀ)) ((Bᵀ * H.toBlocks₁₁ + A₂ᵀ * H.toBlocks₂₁) * A₁ᴴᵀ) ((Bᵀ * H.toBlocks₁₁ + A₂ᵀ * H.toBlocks₂₁) * Bᴴᵀ + (Bᵀ * H.toBlocks₁₂ + A₂ᵀ * H.toBlocks₂₂) * A₂ᴴᵀ)

theorem **subGram_chg** : subGram (chg (fromBlocks A₁ B 0 A₂) H) = chg A₁ (subGram H)

theorem **quotGram_chg** (hA : IsUnit A₁.det) (hH : IsUnit (subGram H).det) : quotGram (chg (fromBlocks A₁ B 0 A₂) H) = chg A₂ (quotGram H)

theorem **trace_mul_adj_nonneg** {S : Matrix m m ℂ} {Q : Matrix l l ℂ} (hS : S.PosDef) (hQ : Q.PosDef) (b : Matrix l m ℂ) : 0 ≤ (b * adj S Q b).trace

theorem **trace_mul_adj_chg** {A₁ : Matrix m m ℂ} {A₂ : Matrix l l ℂ} (h₁ : IsUnit A₁.det) (h₂ : IsUnit A₂.det) (S : Matrix m m ℂ) (Q : Matrix l l ℂ) (b b' : Matrix l m ℂ) : (A₂⁻¹ * b * A₁ * adj (chg A₁ S) (chg A₂ Q) (A₂⁻¹ * b' * A₁)).trace = (b * adj S Q b').trace

theorem **extTrace_eq** (h : MContDiffAt 1 K z) (hp : ∀ᶠ x in 𝒩 z, (K x).PosDef) (ξ η : E) : extTrace K z ξ η = kappa * (secondForm K z ξ * secondFormAdj K z η).trace

theorem **diagVal_extTrace_nonneg** (h : MContDiffAt 1 K z) (hp : ∀ᶠ x in 𝒩 z, (K x).PosDef) (ξ : E) : 0 ≤ diagVal (extTrace K z) ξ

theorem **extTrace_congr** {K' : E → Matrix (m ⊕ l) (m ⊕ l) ℂ} (h : K =ᶠ[𝒩 z] K') : extTrace K z = extTrace K' z

theorem **extTrace_chg** {A : E → Matrix (m ⊕ l) (m ⊕ l) ℂ} (h : MContDiffAt 1 K z) (hp : ∀ᶠ x in 𝒩 z, (K x).PosDef) (hA : MContDiffAt 1 A z) (hu : IsUnit (A z).det) (hA0 : ∀ η, mdbar A z η = 0) (ht : ∀ᶠ x in 𝒩 z, (A x).toBlocks₂₁ = 0) : extTrace (fun x => chg (A x) (K x)) z = extTrace K z

theorem **extTrace_comp_holo** {E' : Type*} [NormedAddCommGroup E'] [NormedSpace ℂ E'] [NormedSpace ℝ ℂ E'] [IsScalarTower ℝ ℂ E'] {φ : E' → E} {w : E'} (h : MContDiffAt 1 K (φ w)) (hp : ∀ᶠ x in 𝒩 (φ w), (K x).PosDef) (hφ : ContDiffAt ℂ 1 φ w) (ξ η : E') : extTrace (fun x => K (φ x)) w ξ η = extTrace K (φ w) (fderiv ℂ φ w ξ) (fderiv ℂ φ w η)

theorem **diagVal_extTrace_smul** (h : MContDiffAt 1 K z) (hp : ∀ᶠ x in 𝒩 z, (K x).PosDef) (c : ℂ) (ξ : E) : diagVal (extTrace K z) (c • ξ) = ‖c‖ ^ 2 * diagVal (extTrace K z) ξ

theorem **crd_app** (φ : F ⟶ E) (t : Γ(F, U)) (hy : y ∈ ptsIn p σ U) (h : y ∈ ptsIn p σ c.U) : c₀.crd p σ (φ.app U t) y = (Matrix.of fun a b => c₀.crd p σ (φ.app c.U (c.e b)) y a) ∗ᵥ c.crd p σ t y

theorem **crd_of_notMem** (t : Γ(E, U)) (hy : y ∉ ptsIn p σ U) : c₀.crd p σ t y = 0

theorem **gram_eq** [Fintype ι] (c : Chart E r) (hc : y ∈ ptsIn p σ c.U) (t : ι → Γ(E, U)) (hy : y ∈ ptsIn p σ U) : h.gram t y = chg (Matrix.of fun a b => c.crd p σ (t b) y a) (h.G c y)

theorem **gram_app_trans** (φ : F ⟶ E) (c c' : Chart F s) (hc : y ∈ ptsIn p σ c.U) : h.gram (fun a => φ.app c'.U (c'.e a)) y = chg (c.tr p σ c' y) (h.gram (fun a => φ.app c.U (c.e a)) y)

theorem **IsCInf.gram** {h : Met p σ E r} (H : h.IsCInf) [Fintype ι] (t : ι → Γ(E, U)) (hy : y ∈ ptsIn p σ U) : MInfAt (h.gram t) y

theorem **IsConj.gram** {h : Met p σ E r} {h' : Met p σ' E r} (H : h.IsConj h') (t : ι → Γ(E, U)) {y' : CxPts p σ'} (e : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) (hy : y ∈ ptsIn p σ U) : h'.gram t y' = (h.gram t y).map star

theorem **crd_frame** (hy : y ∈ ptsIn p σ a.U) (i j : Fin r ⊕ Fin s) : (a.mid hS).crd p σ (a.frame j) y (finSumFinEquiv i) = (1 : Matrix _ _ ℂ) i j

theorem **tr_apply** (i j : Fin r ⊕ Fin s) : a.tr p σ hS a' y i j = (a.mid hS).crd p σ (a'.frame j) y (finSumFinEquiv i)

theorem **tr_eq** (hy : y ∈ ptsIn p σ a.U) (hy' : y ∈ ptsIn p σ a'.U) : a.tr p σ hS a' y = fromBlocks (a.sub.tr p σ a'.sub y) (a.tr p σ hS a' y).toBlocks₁₂ 0 (a.quot.tr p σ a'.quot y)

theorem **gram_frame** (hy : y ∈ ptsIn p σ a.U) : h.gram a.frame y = (h.G (a.mid hS) y).submatrix finSumFinEquiv finSumFinEquiv

theorem **posDef_gram_frame** (hy : y ∈ ptsIn p σ a.U) : (h.gram a.frame y).PosDef

theorem **gram_frame_trans** (hy : y ∈ ptsIn p σ a.U) (hy' : y ∈ ptsIn p σ a'.U) : h.gram a'.frame y = chg (a.tr p σ hS a' y) (h.gram a.frame y)

theorem **exists_adapted** (y : CxPts p σ) : ∃ a : AdaptedChart S r s, y ∈ ptsIn p σ a.U

theorem **quot_G** (hy : y ∈ ptsIn p σ a.U) : (h.quot hS h₁ h₃).G a.quot y = quotGram (h.gram a.frame y)

theorem **IsCInf.quot** (H : h.IsCInf) : (h.quot hS h₁ h₃).IsCInf

theorem **IsConj.quot** {h' : Met p σ' S.X₂ (r + s)} (H : h.IsConj h') : (h.quot hS h₁ h₃).IsConj (h'.quot hS h₁ h₃)

theorem **form_restrict** (hi : IsSubbundle i) (hF : IsVB F s) {U : X.Opens} (t t' : Γ(F, U)) {y : CxPts p σ} (hy : y ∈ ptsIn p σ U) : (h.restrict hi hF).form t t' y = h.form (i.app U t) (i.app U t') y

theorem **IsCInf.restrict** (H : h.IsCInf) (hi : IsSubbundle i) (hF : IsVB F s) : (h.restrict hi hF).IsCInf

theorem **IsConj.restrict** {h' : Met p σ' E r} (H : h.IsConj h') (hi : IsSubbundle i) (hF : IsVB F s) : (h.restrict hi hF).IsConj (h'.restrict hi hF)

theorem **eventually_posDef** (hy : y ∈ c.e.source) (ha : y ∈ ptsIn p σ a.U) : ∀ᶠ v in 𝒩 (c.e y), (h.gram a.frame (c.e.symm v)).PosDef

theorem **bcForm_eq** (H : h.IsCInf) (hy : y ∈ c.e.source) (ha : y ∈ ptsIn p σ a.U) (ha' : y ∈ ptsIn p σ a'.U) : h.bcForm a' c y = h.bcForm a c y

theorem **bcDens_eq** (H : h.IsCInf) (hy : y ∈ c.e.source) (ha : y ∈ ptsIn p σ a.U) : h.bcDens hS h₁ h₃ c y = 2 ∗ diagVal (h.bcForm a c y) (EuclideanSpace.single 0 1)

theorem **bcDens_nonneg** (H : h.IsCInf) (hy : y ∈ c.e.source) : 0 ≤ h.bcDens hS h₁ h₃ c y

theorem **bcDens_trans** (H : h.IsCInf) (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : h.bcDens hS h₁ h₃ c y = ‖ChartGlue.transDet c.e c'.e y‖ ^ 2 ∗ h.bcDens hS h₁ h₃ c' y

theorem **bcMeas_apply** [SecondCountableTopology (CxPts p σ)] (H : h.IsCInf) {T : Set (CxPts p σ)} (hT : MeasurableSet T) (hc : T ⊆ c.e.source) : h.bcMeas hS h₁ h₃ T = ∫⁻ v in c.e '' T, ENNReal.ofReal (h.bcDens hS h₁ h₃ c (c.e.symm v))

theorem **bcMeas_eq_zero** (H : ∀ (c : AlgChart p σ 1), ∀ y ∈ c.e.source, h.bcDens hS h₁ h₃ c y = 0) : h.bcMeas hS h₁ h₃ = 0

theorem **bcMeas_rank_zero** {h : Met p σ S.X₂ (0 + s)} (h₁ : IsVB S.X₁ 0) : h.bcMeas hS h₁ h₃ = 0

theorem **IsCInf.quot** (H : h.IsCInf) : (h.quot hS h₁ h₃).IsCInf

theorem **IsCInf.restrict** (H : h.IsCInf) {i : F ⟶ E} (hi : IsSubbundle i) (hF : IsVB F s) : (h.restrict hi hF).IsCInf

theorem **stepSeq_shortExact** : (stepSeq fl i).ShortExact

theorem **IsCInf.grMet** (H : h.IsCInf) : (h.grMet fl i).IsCInf

theorem **flagBc_rank_one** (h : VMetric p E 1) (fl : Vert.Flag E 1) : h.FlagBcFinite fl ∧ h.flagBc fl = 0

Development.AbelianVarietyTheory.Definitions.Divisors.DivPowerCompatTwoFormDerivativeCompOneForm

private theorem **dDR_dpow_succ_sub_smul_mem_rel** (n : ℕ) {a : B} (ha : a ∈ J) : dDR R B (ι1 (D R B (δ.dpow (n + 1) a) − δ.dpow n a • D R B a)) ∈ rel R δ

theorem **dDR_mem_rel** {x : DR R B} (hx : x ∈ rel R δ) : dDR R B x ∈ rel R δ

theorem **mDR_mem_rel** {x : DR R B} (hx : x ∈ rel R δ) : h.mDR x ∈ rel R' δ'

theorem **divPowerCompatOneFormMap_mk** (x : Ω[B⁄R]) : km1 h hd (Submodule.Quotient.mk x) = Submodule.Quotient.mk (h.m1 x)

theorem **km_mk** (j : ℕ) (x : Om R B j) : km h hd j (Submodule.Quotient.mk x) = Submodule.Quotient.mk (h.mOm j x)

theorem **divPowerCompatFunctionDifferential_apply** (b : B) : kd0 R δ b = Submodule.Quotient.mk (D R B b)

theorem **divPowerCompatOneFormDerivative_mk** (x : Ω[B⁄R]) : kd1 R δ (Submodule.Quotient.mk x) = Submodule.Quotient.mk (dOne R B x)

theorem **divPowerCompatTwoFormDerivative_mk** (x : Om R B 2) : kd2 R δ (Submodule.Quotient.mk x) = Submodule.Quotient.mk (dTwo R B x)

theorem **divPowerCompatOneFormMap_functionDifferential** (x : B) : km1 h hd (kd0 R δ x) = kd0 R' δ' (α x)

theorem **divPowerCompatFormMap_oneFormDerivative** (x : K1 R δ) : km h hd 2 (kd1 R δ x) = kd1 R' δ' (km1 h hd x)

theorem **divPowerCompatFormMap_twoFormDerivative** (x : K R δ 2) : km h hd 3 (kd2 R δ x) = kd2 R' δ' (km h hd 2 x)

theorem **divPowerCompatOneFormDerivative_comp_differential** (x : B) : kd1 R δ (kd0 R δ x) = 0

theorem **divPowerCompatTwoFormDerivative_comp_oneForm** (x : K1 R δ) : kd2 R δ (kd1 R δ x) = 0

theorem **divPowerCompatOneFormMap_id** {τ : R →+* R} {β : B →+* B} (h : SemiHom τ β) (hd : IsDPMorphism δ δ β) (e : ∀ b, β b = b) (x : K1 R δ) : km1 h hd x = x

theorem **km_id** {τ : R →+* R} {β : B →+* B} (h : SemiHom τ β) (hd : IsDPMorphism δ δ β) (e : ∀ b, β b = b) (j : ℕ) (x : K R δ j) : km h hd j x = x

theorem **divPowerCompatOneFormMap_comp** (e : ∀ b, β (α b) = γ b) (x : K1 R δ) : km1 h' hd' (km1 h hd x) = km1 h'' hd'' x

theorem **km_km** (e : ∀ b, β (α b) = γ b) (j : ℕ) (x : K R δ j) : km h' hd' j (km h hd j x) = km h'' hd'' j x

theorem **morSemi** (T : Thick κ' t') : SemiHom (B

theorem **morDP** (T : Thick κ' t') : IsDPMorphism (m.obj T).δ T.δ (RingHom.id T.B)

theorem **t_eq** : D.t = D.i ≫ D.s

theorem **mono_i** : Mono D.i

theorem **pt_i** (U : D.X.affineOpens) : pt D U ≫ D.i = spMap (Ideal.Quotient.mk _) ≫ affPt D.s U

private theorem **sec_res_comp_algebraMap** {U V : D.X.affineOpens} (h : V.1 ≤ U.1) : (Sec.res D.s h : Sec D.s U.1 →+* Sec D.s V.1).comp (algebraMap D.R (Sec D.s U.1)) = algebraMap D.R (Sec D.s V.1)

theorem **res_le** {U V : D.X.affineOpens} (h : V.1 ≤ U.1) : D.I.map (algebraMap D.R (Sec D.s U.1)) ≤ (D.I.map (algebraMap D.R (Sec D.s V.1))).comap (Sec.res D.s h : Sec D.s U.1 →+* Sec D.s V.1)

theorem **res_dp** {U V : D.X.affineOpens} (h : V.1 ≤ U.1) : IsDPMorphism (δ U) (δ V) (Sec.res D.s h : Sec D.s U.1 →+* Sec D.s V.1)

theorem **res_pt** {U V : D.X.affineOpens} (h : V.1 ≤ U.1) : spMap (Ideal.quotientMap _ (Sec.res D.s h : Sec D.s U.1 →+* Sec D.s V.1) (res_le D h)) ≫ pt D U = pt D V

theorem **map_sq** {R A : Type} [CommRing R] [CommRing A] {I : Ideal R} (h : I ^ 2 = 0) (f : R →+* A) : I.map f ^ 2 = 0

theorem **Inhab.span_two_sq** : Ideal.span {(2 : ZMod 4)} ^ 2 = 0

theorem **Lift.lineModFour_ideal** : (Lift.lineModFour.L.obj Inhab.wholeLine).J ≠ ⊥ ∧ (Lift.lineModFour.L.obj Inhab.wholeLine).J ≠ ⊤

theorem **Lift.semi** (U : AffOp D.X) : SemiHom (RingHom.id D.R) (Λ.e U : Sec D.s U.unop.1 →+* (Λ.L.obj U).B)

theorem **Pr.lineModFour_ideal** : Pr.lineModFour.T.J ≠ ⊥ ∧ Pr.lineModFour.T.J ≠ ⊤

theorem **levE_eeLineModFour** : LevE _ _ _ 2 eeLineModFour

theorem **Inhab.isFree_id** {R : Type} [CommRing R] {I : Ideal R} {X : Scheme.{0}} {κ : DividedPowers I} {t : X ⟶ Sp R} {n : ℕ} {T : Thick κ t} (h : IsLev T.δ n) : IsFree n (𝟙 T) (Empty.elim : Empty → T.B)

theorem **Inhab.rel_bot** (R B : Type) [CommRing R] [CommRing B] [Algebra R B] : rel R (dp0 B) = ⊥

theorem **Inhab.rel_dual_ne_bot** : rel (ZMod 2) (divided₂ (SqExt.dual (ZMod 2)).sq) ≠ ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.DividedWeightUnique

theorem **wts_isDiv** (ι : Type) (N kc p T Dd c0 G V : ℕ) (b : ℝ) (hT : 1 ≤ T) : (wts ι N kc p T Dd c0 G V b).IsDiv

theorem **wts_hyp** (ι : Type) (N kc p T Dd c0 G V : ℕ) (b : ℝ) (h1 : 1 ≤ kc) (h2 : kc < p) (h3 : p < N) (h4 : N − p < p) (hT : 4 ∗ Dd + 4 ≤ T) (hG1 : 4 ∗ (c0 + 2 ∗ N + T + Dd + 1) + 1 ≤ G) (hG2 : 24 ∗ N ∗ T + 1 ≤ G) (hG3 : (T + Dd + 2) ∗ (T + Dd + 2 + 2 ∗ N) ≤ G) (hV : 4 ∗ ((2 ∗ N + c0 + (T + Dd + 1) ∗ G ∗ N + N) + (T + Dd + 1) ∗ G ∗ N + c0 + 2 ∗ N) + 20 ∗ N ∗ T ≤ V) (hb : (((T + Dd + 1) ∗ G ∗ N + (T + Dd + 1) ∗ V + 6 ∗ N + 2 ∗ c0 : ℕ) : ℝ) ≤ 2 ∗ b) : (wts ι N kc p T Dd c0 G V b).Hyp

theorem **exists_wts** (ι : Type) (N kc p T Dd c0 Hb : ℕ) (h1 : 1 ≤ kc) (h2 : kc < p) (h3 : p < N) (h4 : N − p < p) (hT : 4 ∗ Dd + 4 ≤ T) : ∃ b0 : ℝ, ∀ b, b0 ≤ b → ∃ W : Wts ι N, W.Hyp ∧ W.IsDiv ∧ W.kc = kc ∧ W.p = p ∧ W.T = T ∧ W.Dd = Dd ∧ W.c0 = c0 ∧ W.b = b ∧ (Hb : ℤ) ≤ W.h0 ∧ 0 ≤ W.S ∧ ((kc : ℝ) + 1) ∗ W.S ≤ b ∧ ((kc : ℝ) + 1) ∗ W.eta ≤ b

theorem **dividedWeight_unique** {ι : Type} {N : ℕ} (W : Wts ι N) (v : ι → Fin N → ℤ) (hk : 1 ≤ W.kc) (hN : W.kc ≤ N) (h : W.IsDiv) (h' : Wts.IsDiv { W with u1

theorem **chi_facts** (D : D7) : (∀ n : ℕ, chiK D n = ((jacobiSym n D.host.d : ℤ) : ℂ)) ∧ (chiK D).IsPrimitive ∧ (chiK D).Odd

theorem **f_pos** (D : D7) : 1 ≤ D.R.f

theorem **page_L** (D : D7) (N kc : ℕ) (x S η : ℝ) : (page D N kc x S η).L = x

theorem **page_b** (D : D7) (N kc : ℕ) (x S η : ℝ) : (page D N kc x S η).b = Real.exp x

theorem **page_params** (D : D7) (N kc : ℕ) (x S η : ℝ) (hrd : 4 ≤ D.host.rdE) (hx : 2187 ≤ x) (h1 : Real.log D.host.rdE ≤ x) (h2 : |dlogL D 1| ≤ x) (hk : kc < N) (hS0 : 0 ≤ S) (hS : ((kc : ℝ) + 1) ∗ S ≤ Real.exp x) (hη : ((kc : ℝ) + 1) ∗ η ≤ Real.exp x) : PageParams (page D N kc x S η)

theorem **shifts_proper** : shifts.Proper

theorem **joint** {D : D7} (Y : ShimX D.what₂.Tau) (K : Type) [Field K] [NumberField K] [StarRing K] (kc c0 : ℕ) (CI Ce C2 : ℝ) (hCe : 0 ≤ Ce) (hC2 : 0 ≤ C2) (h1 : 1 ≤ kc) (h2 : kc < Y.j) (h3 : Y.N − Y.j < Y.j) (hrd : 4 ≤ D.host.rdE) (hbad : Y.lev.card ≤ 15) (sl : ℝ → (D.what₂.Tau → Fin Y.N → ℝ) → Y.Comp → ℝ) (hsl : ∀ B u, GlobalTheory.IndefiniteRowSideFullHecke.ComponentMean.IsSlope Y B u (sl B u)) : ∃ (P : SlopeBoundParameters) (F : SlopeBoundLiftData K P) (W : Wts D.what₂.Tau Y.N), W.kc = kc ∧ W.c0 = c0 ∧ PageParams P ∧ MeanTie Y F W shifts CI ∧ LiftTie Y F W ∧ MeanSizes Y W shifts ∧ SumBound P W shifts Ce C2 ∧ ∃ e : Y.Comp ≃ F.ι, ∀ c, F.sl (e c) = sl (bZero Y W) (reWt W.u') c

theorem **twoFlag_isSlope** (D : D7) (N j : ℕ) (hj : 1 ≤ j) (hjN : j < N) (B : ℝ) (u : D.what₂.Tau → Fin N → ℝ) : GlobalTheory.IndefiniteRowSideFullHecke.ComponentMean.IsSlope (twoFlag D.what₂.Tau N j hj hjN) B u (fun c : Bool => (if c then 1 else 0) + (0 − ∑ τ, ∑ i, |u τ i|))

theorem **exists_joint** (D : D7) (K : Type) [Field K] [NumberField K] [StarRing K] (N j kc c0 : ℕ) (CI Ce C2 : ℝ) (hCe : 0 ≤ Ce) (hC2 : 0 ≤ C2) (h1 : 1 ≤ kc) (h2 : kc < j) (hjN : j < N) (h3 : N − j < j) (hrd : 4 ≤ D.host.rdE) : ∃ (Y : ShimX D.what₂.Tau) (P : SlopeBoundParameters) (F : SlopeBoundLiftData K P) (W : Wts D.what₂.Tau Y.N) (A : Shifts), Y.N = N ∧ Y.j = j ∧ W.kc = kc ∧ W.c0 = c0 ∧ PageParams P ∧ MeanTie Y F W A CI ∧ LiftTie Y F W ∧ MeanSizes Y W A ∧ SumBound P W A Ce C2 ∧ A.Proper ∧ (∃ p ∈ A.Pe, 0 < A.e p) ∧ 0 < A.C7 ∧ ∃ c c', F.sl c ≠ F.sl c'

Development.AbelianVarietyTheory.Definitions.Divisors.DlogRatioVolumeFormNeZero

theorem **IsFinCover.of_charZero** [CharZero F] (pd : IsPd k x) (fin : FiniteDimensional F K) (smooth : ∀ i, Algebra.Smooth k (rB k K x {i})) : IsFinCover k K x

theorem **IsFinCover.exists_basis** (h : IsFinCover k K x) : ∃ b : Module.Basis (Fin d) K Ω[K⁄k], ∀ j, b j = D k K (algebraMap F K (x j.succ / x 0))

theorem **IsFinCover.exists_isTop** (h : IsFinCover k K x) : ∃ τ : DR k K ⁻¹[K], IsTop k K x τ

theorem **IsFinCover.dlogRatioVolumeForm_ne_zero** (h : IsFinCover k K x) : vol0 k K x ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.DominatesIffIsLocalHom

theorem **nonUnit_coe_iff** {R : Subring K} (x : R) : NonUnit R (x : K) ↔ ¬ IsUnit x

theorem **coe_algebraMap** (x : MvPolynomial (Fin r) R) : ((algebraMap _ (topRing R r) x : topRing R r) : FractionRing (MvPolynomial (Fin r) R)) = algebraMap _ _ x

theorem **unit_const** (y : (origin R r).primeCompl) : IsUnit (MvPolynomial.constantCoeff (y : MvPolynomial (Fin r) R))

theorem **sect_algebraMap** (x : MvPolynomial (Fin r) R) : sect R r (algebraMap _ (topRing R r) x) = MvPolynomial.constantCoeff x

theorem **eval_eq** : MvPolynomial.eval₂Hom (incl R r) (coord R r) = algebraMap (MvPolynomial (Fin r) R) (topRing R r)

theorem **coord_ne_zero** (i : Fin r) : coord R r i ≠ 0

theorem **IsEtTower.poly** {c : k →+* K} {R : Subring K} [IsLocalRing R] (h : IsLocModel c R) (h' : IsIntegrallyClosed R) (r : ℕ) : IsEtTower c R ((topRing R r).subtype.comp ((incl R r).comp (c.codRestrict R h.const_mem))) (topRing R r) (incl R r) (sect R r) (coord R r) where

theorem **IsEtTower.nonUnit_coord** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {r : ℕ} {ι : R →+* T} {ψ : T →+* R} {τ : Fin r → T} (h : IsEtTower cK R cL T ι ψ τ) (i : Fin r) : NonUnit T (τ i)

theorem **IsEtTower.ratRes** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {r : ℕ} {ι : R →+* T} {ψ : T →+* R} {τ : Fin r → T} (h : IsEtTower cK R cL T ι ψ τ) (hR : RatRes cK R) : RatRes cL T

theorem **exists_isEtTower_line** (k : Type) [Field k] : ∃ (L : Type) (_ : Field L) (cL : k →+* L) (T : Subring L) (ι : (⊤ : Subring k) →+* T) (ψ : T →+* (⊤ : Subring k)) (τ : Fin 1 → T), IsEtTower (RingHom.id k) ⊤ cL T ι ψ τ ∧ τ 0 ≠ 0 ∧ NonUnit T (τ 0) ∧ RatRes cL T ∧ Dominates cL T ⊤

theorem **not_dominates_top** (ρ : K →+* L) {O : Subring K} {f : K} (hf : NonUnit O f) (h0 : f ≠ 0) : ¬ Dominates ρ ⊤ O

theorem **ratRes_iff_surjective** {c : k →+* K} {R : Subring K} [IsLocalRing R] (hc : ∀ a, c a ∈ R) : RatRes c R ↔ Function.Surjective ((IsLocalRing.residue R).comp (c.codRestrict R hc))

theorem **not_ratRes_complex** : ¬ RatRes (algebraMap ℝ ℂ) ⊤

theorem **dominates_iff_isLocalHom** {ρ : K →+* L} {B : Subring L} {O : Subring K} : Dominates ρ B O ↔ ∃ h : ∀ f ∈ O, ρ f ∈ B, IsLocalHom (ρ.restrict O B h)

theorem **IsEtTower.exists_deriv** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {r : ℕ} {ι : R →+* T} {ψ : T →+* R} {τ : Fin r → T} (h : IsEtTower cK R cL T ι ψ τ) (v : Fin r → R) : ∃ φ : T →+* DualNumber R, (∀ a, φ (ι a) = TrivSqZeroExt.inl a) ∧ ∀ i, φ (τ i) = TrivSqZeroExt.inr (v i)

theorem **IsEtTower.coord_indep** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {r : ℕ} {ι : R →+* T} {ψ : T →+* R} {τ : Fin r → T} (h : IsEtTower cK R cL T ι ψ τ) (a : Fin r → R) (ha : ∑ i, ι (a i) * τ i = 0) (j : Fin r) : a j = 0

theorem **IsEtTower.coord_ne_zero** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {r : ℕ} {ι : R →+* T} {ψ : T →+* R} {τ : Fin r → T} (h : IsEtTower cK R cL T ι ψ τ) (j : Fin r) : τ j ≠ 0

theorem **IsEtTower.ker_eq_span** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {r : ℕ} {ι : R →+* T} {ψ : T →+* R} {τ : Fin r → T} (h : IsEtTower cK R cL T ι ψ τ) : RingHom.ker ψ = Ideal.span (Set.range τ)

theorem **IsEtTower.bijective_of_zero** {cK : k →+* K} {R : Subring K} {cL : k →+* L} {T : Subring L} {ι : R →+* T} {ψ : T →+* R} {τ : Fin 0 → T} (h : IsEtTower cK R cL T ι ψ τ) : Function.Bijective ψ

Development.AbelianVarietyTheory.Definitions.Divisors.DualRef

theorem **Setup.Core.dualRef** {D : Formalism.BasicNotions.D7} {O : Setup D} (h : O.Core) : O.DualRef

Development.AbelianVarietyTheory.Definitions.Divisors.EigLatLe

theorem **eigLat_le** (W : WTZ D.host.E) : eigLat U lev χ W ≤ latW U W

Development.AbelianVarietyTheory.Definitions.Divisors.EightDvdNCol

theorem **eight_dvd_nCol** : 8 ∣ nCol Dw

theorem **sum_le_nCol** : ∑ f : Lt1 Dw, 2 * rk f ≤ nCol Dw

theorem **OscLe.mono** {cx : Lt1 Dw → ℕ} {b b' : ℝ} (h : OscLe Xo c p cx b) (hb : b ≤ b') : OscLe Xo c p cx b'

Development.AbelianVarietyTheory.Definitions.Divisors.EllFamNontrivial

theorem **ellFam_nontrivial** : (fun _ ↦ Complex.I) ∈ ellFam.V ∧ ellFam.τ (1, 0) (fun _ ↦ Complex.I) = (fun _ ↦ Complex.I) ∧ ellFam.τ (0, 1) (fun _ ↦ Complex.I) = fun _ ↦ 1

theorem **ellFam_free** (l : ℤ × ℤ) (z : Fin 1 → ℂ) (hz : z ∈ ellFam.V) (h : ellFam.τ l z = 0) : l = 0

theorem **Fam.shift_zero** (F : Fam I g Λ) (p : Tot I g) : F.shift 0 p = p

theorem **Fam.shift_add** (F : Fam I g Λ) (l m : Λ) (p : Tot I g) : F.shift (l + m) p = F.shift l (F.shift m p)

theorem **Fam.shift_analytic** (F : Fam I g Λ) (l : Λ) {W : Set (Tot I g)} (h : ∀ p ∈ W, p.1 ∈ F.V) : AnalyticOnNhd ℂ (F.shift l) W

theorem **Close.map** {W : Opens (Tot I g)} {C C' : Type} [CommRing C] [CommRing C'] {J : Ideal C} {J' : Ideal C'} {ρ₁ ρ₂ : Hol W →+* C} (h : Close J ρ₁ ρ₂) (φ : C →+* C') (hφ : ∀ c ∈ J, φ c ∈ J') : Close J' (φ.comp ρ₁) (φ.comp ρ₂)

theorem **Close.comap** {W W' : Opens (Tot I g)} {C : Type} [CommRing C] {J : Ideal C} {ρ₁ ρ₂ : Hol W' →+* C} (h : Close J ρ₁ ρ₂) (ψ : Hol W →+* Hol W') (hψ : ∀ a f, dw a (ψ f) = ψ (dw a f)) : Close J (ρ₁.comp ψ) (ρ₂.comp ψ)

theorem **Close.sub_mem** {W : Opens (Tot I g)} {C : Type} [CommRing C] {J : Ideal C} {ρ₁ ρ₂ : Hol W →+* C} (h : Close J ρ₁ ρ₂) (f : Hol W) : ρ₂ f − ρ₁ f ∈ J

theorem **fn_cst** (k : K.K) (f : Hol F.V) {p : Tot I g} (hp : p ∈ K.W k) : (K.cst k f).fn p = f.fn p.1

theorem **fn_sh** {l : Λ} {k k' : K.K} (h : K.le l k' k) (f : Hol (K.W k)) {p : Tot I g} (hp : p ∈ K.W k') : (sh h f).fn p = f.fn (F.shift l p)

theorem **sh_refl** {k : K.K} (h : K.le 0 k k) (f : Hol (K.W k)) : sh h f = f

theorem **sh_sh** {l m : Λ} {k k' k'' : K.K} (h : K.le l k' k) (h' : K.le m k'' k') (h'' : K.le (l + m) k'' k) (f : Hol (K.W k)) : sh h' (sh h f) = sh h'' f

theorem **sh_cst** {l : Λ} {k k' : K.K} (h : K.le l k' k) (f : Hol F.V) : sh h (K.cst k f) = K.cst k' f

theorem **dw_sh** {l : Λ} {k k' : K.K} (h : K.le l k' k) (a : Fin g) (f : Hol (K.W k)) : dw a (sh h f) = sh h (dw a f)

theorem **dw_cst** (k : K.K) (a : Fin g) (f : Hol F.V) : dw a (K.cst k f) = 0

theorem **Hom.ext'** {T T' : K.Obj} {f f' : Hom T T'} (h : ∀ c, f.φ c = f'.φ c) (h' : f.deg = f'.deg) : f = f'

theorem **hom_ext** {T T' : K.Obj} {f f' : T ⟶ T'} (h : ∀ c, f.φ c = f'.φ c) (h' : f.deg = f'.deg) : f = f'

@[simp] theorem **comp_φ** {T T' T'' : K.Obj} (f : T ⟶ T') (f' : T' ⟶ T'') (c : T.C) : (f ≫ f').φ c = f'.φ (f.φ c)

@[simp] theorem **comp_deg** {T T' T'' : K.Obj} (f : T ⟶ T') (f' : T' ⟶ T'') : (f ≫ f').deg = f.deg + f'.deg

@[simp] theorem **id_deg** (T : K.Obj) : Hom.deg (𝟙 T) = 0

theorem **mp_Oc** {T T' : K.Obj} (f : T ⟶ T') : mp K.Oc f = f.φ

theorem **Fam.allCharts_based** (F : Fam I g Λ) : F.allCharts.Based

theorem **Fam.allCharts_rich** (F : Fam I g Λ) : F.allCharts.Rich

theorem **Fam.transl_deg** (F : Fam I g Λ) (l : Λ) : (F.transl l).deg = l

theorem **Charts.dualAt_J_ne_bot** {F : Fam I g Λ} (K : Charts F) (k : K.K) (p : Tot I g) (hp : p ∈ K.W k) : (K.dualAt k p hp).J ≠ ⊥

theorem **step_tangentAt** {W : Opens (Tot I g)} (a : Fin g) (p : Tot I g) (hp : p ∈ W) : Step (Ideal.span {DualNumber.eps}) ((algebraMap ℂ (DualNumber ℂ)).comp (Hol.ev p hp).toRingHom) (tangentAt a p hp)

Development.AbelianVarietyTheory.Definitions.Divisors.EndValuedForm11Coeffs

theorem **hasChernIntegral_iff** (E : Fr (ratOf X) (fun _ : Unit => m) 0) (O : X.X.Opens) (H : Mat (fun _ : Unit => m) X.Fn → CxPts X.str σ → Mat (fun _ : Unit => m) ℂ) (t : ℝ) : HasChernIntegral E O H (MvPolynomial.X ((), X.dim)) t ↔ HasInt (fun c : AlgChart X.str σ X.dim => c.e) (topDens E H) (ptsIn X.str σ O) t

Development.AbelianVarietyTheory.Definitions.Divisors.EntireEqOfHalfPlane

theorem **single_apply** (v : HeightOneSpectrum (𝒪 E)) (x : (v.adicCompletion E)ˣ) (w : HeightOneSpectrum (𝒪 E)) : (single v x : Af E) w = Function.update (fun w : HeightOneSpectrum (𝒪 E) => (1 : w.adicCompletion E)) v x w

@[simp] theorem **single_self** (v : HeightOneSpectrum (𝒪 E)) (x : (v.adicCompletion E)ˣ) : (single v x : Af E) v = x

theorem **suppAt_single** (v : HeightOneSpectrum (𝒪 E)) (x : (v.adicCompletion E)ˣ) : SuppAt v (single v x)

theorem **finPart_single** (χ : HChar E) (v : HeightOneSpectrum (𝒪 E)) (x : (v.adicCompletion E)ˣ) : finPart χ (single v x) = χ.loc v x

@[simp] theorem **coe_coord** (v : HeightOneSpectrum (𝒪 E)) (y : (Af E)ˣ) : (coord v y : v.adicCompletion E) = (y : Af E) v

theorem **SuppAt.eq_single** {v : HeightOneSpectrum (𝒪 E)} {y : (Af E)ˣ} (h : SuppAt v y) : y = single v (coord v y)

theorem **SuppAt.finPart_eq** {v : HeightOneSpectrum (𝒪 E)} {y : (Af E)ˣ} (h : SuppAt v y) (χ : HChar E) : finPart χ y = χ.loc v (coord v y)

theorem **levelLe_iff_finPart** (χ : HChar E) (v : HeightOneSpectrum (𝒪 E)) (n : ℕ) : χ.LevelLe v n ↔ ∀ y : (Af E)ˣ, SuppAt v y → Valued.v ((y : Af E) v) = 1 → Valued.v ((y : Af E) v − 1) ≤ WithZero.exp (−(n : ℤ)) → finPart χ y = 1

theorem **unifAt_spec** (v : HeightOneSpectrum (𝒪 E)) : SuppAt v (unifAt v) ∧ Valued.v ((unifAt v : Af E) v) = WithZero.exp (−1 : ℤ)

theorem **finPart_unifAt** (χ : HChar E) (v : HeightOneSpectrum (𝒪 E)) : finPart χ (unifAt v) = χ.val v

theorem **finPart_eq_val_zpow** {χ : HChar E} {v : HeightOneSpectrum (𝒪 E)} (h : χ.UnrAt v) {y : (Af E)ˣ} (hy : SuppAt v y) {n : ℤ} (hn : Valued.v ((y : Af E) v) = WithZero.exp (−n)) : finPart χ y = χ.val v ^ n

@[simp] theorem **finPart_mul** (χ χ' : HChar E) (y : (Af E)ˣ) : finPart (χ * χ') y = finPart χ y * finPart χ' y

@[simp] theorem **finPart_inv** (χ : HChar E) (y : (Af E)ˣ) : finPart χ⁻¹ y = (finPart χ y)⁻¹

theorem **infDiag_eq** (k : Eˣ) : infDiag k = Units.map (MonoidHom.inl (InfiniteAdeleRing E) (Af E)) (Units.map (algebraMap E (InfiniteAdeleRing E)).toMonoidHom k)

theorem **finPart_mul_inf** (χ : HChar E) (k : Eˣ) : finPart χ (Units.map (algebraMap E (Af E)).toMonoidHom k) * χ.inf k = 1

theorem **hasInf_iff_two_mul** (χ : HChar E) (φ : I → E →+* ℂ) (m : I → ℤ) : HasInf χ φ m ↔ χ.HasInf φ (fun i => 2 * m i)

theorem **hasInfRatio_iff** (χ : HChar E) (φ : I → E →+* ℂ) (m : I → ℤ) : HasInfRatio χ φ m ↔ ∀ k : Eˣ, ((χ.inf k : ℂˣ) : ℂ) = ∏ i, (φ i (k : E) / starRingEnd ℂ (φ i (k : E))) ^ m i

theorem **TrivOnF.inf_eq_one** {χ : HChar E} (h : TrivOnF χ) (k : Eˣ) (hk : (k : E) ∈ maximalRealSubfield E) : χ.inf k = 1

theorem **ExpAt.unique** {χ : HChar E} {v : HeightOneSpectrum (𝒪 E)} {n n' : ℕ} (h : ExpAt χ v n) (h' : ExpAt χ v n') : n = n'

theorem **expAt_zero_iff** (χ : HChar E) (v : HeightOneSpectrum (𝒪 E)) : ExpAt χ v 0 ↔ χ.UnrAt v

theorem **ExpAt.not_unrAt** {χ : HChar E} {v : HeightOneSpectrum (𝒪 E)} {n : ℕ} (h : ExpAt χ v n) (hn : n ≠ 0) : ¬ χ.UnrAt v

theorem **isCond_iff_expAt** (χ : HChar E) {𝔣 : Ideal (𝒪 E)} (h𝔣 : 𝔣 ≠ ⊥) : χ.IsCond 𝔣 ↔ ∀ v, ExpAt χ v (multiplicity v.asIdeal 𝔣)

theorem **entire_eq_of_halfPlane** {f g : ℂ → ℂ} (hf : Differentiable ℂ f) (hg : Differentiable ℂ g) (a : ℝ) (h : ∀ s : ℂ, a < s.re → f s = g s) : f = g

theorem **IsHeckeL.unique** {χ : HChar E} {f g : ℂ → ℂ} (hf : IsHeckeL χ f) (hg : IsHeckeL χ g) : f = g

theorem **IsLatSum.unique** {φ : I → E →+* ℂ} {k : ℕ} {κ : I → ℕ} {L : Set E} {f g : ℂ → ℂ} (hf : IsLatSum φ k κ L f) (hg : IsLatSum φ k κ L g) : f = g

theorem **IsToricSum.unique** {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {c : Ideal (OF E)} {χ : HChar E} {T T' : ℂ} (h : IsToricSum φ s κ c χ T) (h' : IsToricSum φ s κ c χ T') : T = T'

theorem **mem_orderC** {c : Ideal (OF E)} {x : E} : x ∈ orderC c ↔ ∃ (a : OF E) (b : 𝒪 E), b ∈ Ideal.map (algebraMap (OF E) (𝒪 E)) c ∧ x = algebraMap (OF E) E a + algebraMap (𝒪 E) E b

theorem **algebraMap_mem_orderC** (c : Ideal (OF E)) (a : OF E) : algebraMap (OF E) E a ∈ orderC c

theorem **orderC_mono** {c c' : Ideal (OF E)} (h : c' ≤ c) : orderC c' ≤ orderC c

theorem **hatO_mono** {c c' : Ideal (OF E)} (h : c' ≤ c) : hatO c' ≤ hatO c

theorem **hatOUnits_mono** {c c' : Ideal (OF E)} (h : c' ≤ c) : hatOUnits c' ≤ hatOUnits c

theorem **mem_hatOUnits** {c : Ideal (OF E)} {u : (Af E)ˣ} : u ∈ hatOUnits c ↔ (u : Af E) ∈ hatO c ∧ (↑u⁻¹ : Af E) ∈ hatO c

theorem **algebraMap_mem_hatO** {c : Ideal (OF E)} {x : E} (hx : x ∈ orderC c) : algebraMap E (Af E) x ∈ hatO c

@[simp] theorem **picMap_mk** {c c' : Ideal (OF E)} (h : c' ≤ c) (y : (Af E)ˣ) : picMap h (QuotientGroup.mk y) = QuotientGroup.mk y

theorem **picMap_surjective** {c c' : Ideal (OF E)} (h : c' ≤ c) : Function.Surjective (picMap h)

theorem **mem_lattice** {c : Ideal (OF E)} {y : (Af E)ˣ} {x : E} : x ∈ lattice c y ↔ algebraMap E (Af E) x * (↑y⁻¹ : Af E) ∈ hatO c

theorem **mul_mem_lattice** {c : Ideal (OF E)} {y : (Af E)ˣ} {a x : E} (ha : a ∈ orderC c) (hx : x ∈ lattice c y) : a * x ∈ lattice c y

@[simp] theorem **coe_latticeSub** (c : Ideal (OF E)) (y : (Af E)ˣ) : (latticeSub c y : Set E) = lattice c y

theorem **lattice_mul_unit** {c : Ideal (OF E)} (y : (Af E)ˣ) {u : (Af E)ˣ} (hu : u ∈ hatOUnits c) : lattice c (y * u) = lattice c y

theorem **mul_mem_lattice_princ** (c : Ideal (OF E)) (y : (Af E)ˣ) (a : Eˣ) (x : E) : (a : E) * x ∈ lattice c (Units.map (algebraMap E (Af E)).toMonoidHom a * y) ↔ x ∈ lattice c y

theorem **out_mem_lattice** {c : Ideal (OF E)} {y : (Af E)ˣ} (q : modUnits (lattice c y)) : q.1.out ∈ lattice c y ∧ q.1.out ≠ 0

theorem **term_mul** (φ : I → E →+* ℂ) (k : ℕ) (κ : I → ℕ) (σ : ℂ) (a x : E) : term φ k κ σ (a * x) = term φ k κ σ a * term φ k κ σ x

theorem **term_mul_inf** {χ : HChar E} {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} (h : HasInfRatio χ φ fun i => (κ i : ℤ) + s) (a : Eˣ) : term φ (2 * s) κ 0 a * (((χ.inf a : ℂˣ) : ℂ) * ∏ i, (φ i a * starRingEnd ℂ (φ i a)) ^ s) = 1

theorem **modF_unifAt** (v : HeightOneSpectrum (𝒪 E)) : modF (unifAt v) = ((Ideal.absNorm v.asIdeal : ℕ) : ℝ)⁻¹

theorem **cmValueConst_eq** (E : Type) [Field E] [NumberField E] (d k : ℕ) (κ : I → ℕ) : C0 E d k κ = (|(discr (maximalRealSubfield E) : ℝ)|) ^ (((k : ℝ) - 1) / 2) * (∏ i, Real.Gamma ((k : ℝ) + κ i)) * (2 * Real.pi / Real.sqrt d) ^ (∑ i, κ i)

theorem **cmValueConst_pos** (E : Type) [Field E] [NumberField E] {d k : ℕ} (hd : 0 < d) (hk : 1 ≤ k) (κ : I → ℕ) : 0 < C0 E d k κ

theorem **latDisc_ne_zero** (Λ : PeriodPair) : latDisc Λ ≠ 0

theorem **beta_eq** (E : Type) [Field E] [NumberField E] (d s : ℕ) (κ : I → ℕ) (Λ : PeriodPair) (T : ℂ) : beta E d s κ Λ T = ((C0 E d (2 * s) κ : ℂ) * T) ^ 12 / latDisc Λ ^ (2 * s * Fintype.card I + 2 * ∑ i, κ i)

Development.AbelianVarietyTheory.Definitions.Divisors.EqOfHasValueAtInfinite

theorem **HasValueAt.unique** {R : RatFunc ℂ} {x w w' : ℂ} (h : HasValueAt R x w) (h' : HasValueAt R x w') : w = w'

theorem **hasValueAt_iff** {R : RatFunc ℂ} {x w : ℂ} : HasValueAt R x w ↔ ∃ a b : ℂ[X], b.eval x ≠ 0 ∧ R = algebraMap _ _ a / algebraMap _ _ b ∧ w * b.eval x = a.eval x

theorem **hasValueAt_algebraMap** (p : ℂ[X]) (x : ℂ) : HasValueAt (algebraMap _ _ p) x (p.eval x)

theorem **hasValueAt_C** (a x : ℂ) : HasValueAt (RatFunc.C a) x a

theorem **hasValueAt_one** (x : ℂ) : HasValueAt 1 x 1

theorem **hasValueAt_X** (x : ℂ) : HasValueAt RatFunc.X x x

theorem **HasValueAt.add** {R S : RatFunc ℂ} {x v w : ℂ} (h : HasValueAt R x v) (h' : HasValueAt S x w) : HasValueAt (R + S) x (v + w)

theorem **HasValueAt.mul** {R S : RatFunc ℂ} {x v w : ℂ} (h : HasValueAt R x v) (h' : HasValueAt S x w) : HasValueAt (R * S) x (v * w)

theorem **HasValueAt.inv** {R : RatFunc ℂ} {x v : ℂ} (h : HasValueAt R x v) (hv : v ≠ 0) : HasValueAt R⁻¹ x v⁻¹

theorem **HasValueAt.neg** {R : RatFunc ℂ} {x v : ℂ} (h : HasValueAt R x v) : HasValueAt (-R) x (-v)

theorem **HasValueAt.sub** {R S : RatFunc ℂ} {x v w : ℂ} (h : HasValueAt R x v) (h' : HasValueAt S x w) : HasValueAt (R - S) x (v - w)

theorem **HasValueAt.pow** {R : RatFunc ℂ} {x v : ℂ} (h : HasValueAt R x v) (m : ℕ) : HasValueAt (R ^ m) x (v ^ m)

theorem **HasValueAt.zpow** {R : RatFunc ℂ} {x v : ℂ} (h : HasValueAt R x v) (hv : v ≠ 0) (m : ℤ) : HasValueAt (R ^ m) x (v ^ m)

theorem **HasValueAt.multiset_prod** {ι : Type} (S : Multiset ι) {R : ι → RatFunc ℂ} {x : ℂ} {v : ι → ℂ} (h : ∀ i ∈ S, HasValueAt (R i) x (v i)) : HasValueAt (S.map R).prod x (S.map v).prod

theorem **eq_of_hasValueAt_infinite** {R S : RatFunc ℂ} {T : Set ℂ} (hT : T.Infinite) (h : ∀ x ∈ T, ∃ w, HasValueAt R x w ∧ HasValueAt S x w) : R = S

theorem **IsRatIn.unique** {q : ℕ} (hq : 2 ≤ q) {F : ℂ → ℂ} {R S : RatFunc ℂ} (h : IsRatIn q F R) (h' : IsRatIn q F S) : R = S

theorem **IsRatIn.mul** {q : ℕ} {F G : ℂ → ℂ} {R S : RatFunc ℂ} (h : IsRatIn q F R) (h' : IsRatIn q G S) : IsRatIn q (fun z => F z * G z) (R * S)

theorem **IsRatIn.add** {q : ℕ} {F G : ℂ → ℂ} {R S : RatFunc ℂ} (h : IsRatIn q F R) (h' : IsRatIn q G S) : IsRatIn q (fun z => F z + G z) (R + S)

theorem **HasContValue.unique** {q : ℕ} (hq : 2 ≤ q) {F : ℂ → ℂ} {z₀ w w' : ℂ} (h : HasContValue q F z₀ w) (h' : HasContValue q F z₀ w') : w = w'

@[simp] theorem **substRat_X** (a : RatFunc ℂ) (ha : Transcendental ℂ a) : substRat a ha RatFunc.X = a

@[simp] theorem **substRat_C** (a : RatFunc ℂ) (ha : Transcendental ℂ a) (c : ℂ) : substRat a ha (RatFunc.C c) = RatFunc.C c

theorem **HasValueAt.substRat** {a R : RatFunc ℂ} (ha : Transcendental ℂ a) {x y w : ℂ} (h : HasValueAt a x y) (h' : HasValueAt R y w) : HasValueAt (substRat a ha R) x w

theorem **transcendental_C_mul_X** (u : ℂˣ) : Transcendental ℂ (RatFunc.C (u : ℂ) * RatFunc.X)

@[simp] theorem **scaleRat_X** (u : ℂˣ) : scaleRat u RatFunc.X = RatFunc.C (u : ℂ) * RatFunc.X

@[simp] theorem **reflectRat_X** (u : ℂˣ) : reflectRat u RatFunc.X = (RatFunc.C (u : ℂ) * RatFunc.X)⁻¹

@[simp] theorem **scaleRat_C** (u : ℂˣ) (c : ℂ) : scaleRat u (RatFunc.C c) = RatFunc.C c

@[simp] theorem **reflectRat_C** (u : ℂˣ) (c : ℂ) : reflectRat u (RatFunc.C c) = RatFunc.C c

@[simp] theorem **natUnit_val** (q : ℕ) (hq : q ≠ 0) : (natUnit q hq : ℂ) = q

@[simp] theorem **negPowUnit_val** (q : ℕ) (hq : q ≠ 0) (s : ℂ) : (negPowUnit q hq s : ℂ) = (q : ℂ) ^ (-s)

theorem **HasValueAt.reflectRat** {q : ℕ} (hq : q ≠ 0) {R : RatFunc ℂ} {z w : ℂ} (h : HasValueAt R ((q : ℂ) ^ (-(1 - z))) w) : HasValueAt (reflectRat (natUnit q hq) R) ((q : ℂ) ^ (-z)) w

theorem **HasValueAt.scaleRat** {q : ℕ} (hq : q ≠ 0) {R : RatFunc ℂ} {z s w : ℂ} (h : HasValueAt R ((q : ℂ) ^ (-(z + s))) w) : HasValueAt (scaleRat (negPowUnit q hq s) R) ((q : ℂ) ^ (-z)) w

theorem **IsRatIn.shift** {q : ℕ} (hq : q ≠ 0) {F : ℂ → ℂ} {R : RatFunc ℂ} (h : IsRatIn q F R) (s : ℂ) : IsRatIn q (fun z => F (z + s)) (scaleRat (negPowUnit q hq s) R)

@[simp] theorem **lFactor_zero** : lFactor 0 = 1

@[simp] theorem **lFactor_singleton** (a : ℂˣ) : lFactor {a} = (1 - RatFunc.C (a : ℂ) * RatFunc.X)⁻¹

@[simp] theorem **checkFun_apply** {G : Type} [Group G] (c : G → ℂ) (x : G) : checkFun c x = c x⁻¹

@[simp] theorem **checkFun_checkFun** {G : Type} [Group G] (c : G → ℂ) : checkFun (checkFun c) = c

theorem **HasGJ.unique** {R S : RatFunc ℂ} (h : HasGJ μ Ξ c R) (h' : HasGJ μ Ξ c S) : R = S

theorem **HasGJValue.unique** {z₀ w w' : ℂ} (h : HasGJValue μ Ξ c z₀ w) (h' : HasGJValue μ Ξ c z₀ w') : w = w'

private theorem **valuation_det_le_one_of_entries_integer** {y : GL n k} (h : ∀ i j, (y : Matrix n n k) i j ∈ (valuation k).integer) : valuation k (GeneralLinearGroup.det y : k) ≤ 1

theorem **_root_.GlobalTheory.LGrp.nabs_det_of_mem_KO** {x : GL n k} (hx : x ∈ _root_.GlobalTheory.SplitWeil.KO k n) : _root_.GlobalTheory.SplitWeil.nabs k (GeneralLinearGroup.det x) = 1

private theorem **smGL_ord_pow_eq_mul** (x : kˣ) (m : ℕ) : _root_.GlobalTheory.SmGL.ord k (x ^ m) = m * _root_.GlobalTheory.SmGL.ord k x

private theorem **exists_unit_ord_eq_one_and_nabs_eq_inv_resCard** : ∃ ϖ : kˣ, _root_.GlobalTheory.SmGL.ord k ϖ = 1 ∧ _root_.GlobalTheory.SplitWeil.nabs k ϖ = (_root_.GlobalTheory.SplitWeil.resCard k : ℝ)⁻¹

theorem **_root_.GlobalTheory.LGrp.nabs_eq_zpow** (x : kˣ) : _root_.GlobalTheory.SplitWeil.nabs k x = (_root_.GlobalTheory.SplitWeil.resCard k : ℝ) ^ (-_root_.GlobalTheory.SmGL.ord k x)

theorem **cosetInd_mem_SB** (g : GL n k) : cosetInd g ∈ SB (Matrix n n k)

Development.AbelianVarietyTheory.Definitions.Divisors.EqOfSpecializesOfFinite

theorem **ArMod.finrank_pos** {N : ℕ} (m : ArMod N) : 0 < Module.finrank ℤ m.k

theorem **GeomPt.base_push** (π : X' ⟶ X) (hπ : π ≫ p = q) (z : GeomPt q ι) : (z.push π hπ).base = π z.base

theorem **GeomPt.charZero** (z : GeomPt p ι) : CharZero (X.residueField z.base)

theorem **isProper_int** {N : ℕ} (hX : IsArithVar p N) : IsProper (p ≫ spMap (Int.castRingHom k))

theorem **GeomPt.numberField** {N : ℕ} (hX : IsArithVar p N) (z : GeomPt p ι) : NumberField (X.residueField z.base)

theorem **GeomPt.card_emb_pos** {N : ℕ} (hX : IsArithVar p N) (z : GeomPt p ι) : 0 < Nat.card (X.residueField z.base →+* ℂ)

theorem **GeomPt.push_surjective** (π : X' ⟶ X) (hπ : π ≫ p = q) [Surjective π] [LocallyOfFiniteType π] : Function.Surjective (GeomPt.push (ι

theorem **eq_of_specializes_of_finite** (f : X ⟶ Y) [LocallyOfFiniteType f] {x w : X} (h : x ⤳ w) (e : f w = f x) (hf : (f.residueFieldMap x).hom.Finite) : x = w

theorem **finite_of_specializes** (t : X ⟶ Sp ℤ) [LocallyOfFiniteType t] {x a : X} (h : x ⤳ a) [CharZero (X.residueField x)] [Algebra.IsAlgebraic ℚ (X.residueField x)] : (t.residueFieldMap a).hom.Finite

theorem **dim_le_one_iff** (T : Type) [TopologicalSpace T] [T0Space T] [QuasiSober T] : topologicalKrullDim T ≤ 1 ↔ ∀ a : T, (∀ b, a ⤳ b → b ⤳ a) ∨ ∀ b, b ⤳ a → a ⤳ b

theorem **dim_closure_le_one_iff** (x : X) : topologicalKrullDim (closure ({x} : Set X)) ≤ 1 ↔ ∀ a b, x ⤳ a → a ⤳ b → x = a ∨ a = b

theorem **asIdeal_eq_bot** [IsDomain k] [Module.Finite ℤ k] (p : X ⟶ Sp k) (x : X) [CharZero (X.residueField x)] : (p x).asIdeal = ⊥

theorem **exists_specializes_ne** (hX : IsArithVar p N) (x : X) [CharZero (X.residueField x)] : ∃ b, x ⤳ b ∧ x ≠ b

theorem **GeomPt.isAlgPt** (hX : IsArithVar p N) (z : GeomPt p ι) : IsAlgPt z.base

theorem **exists_geomPt_mem** (hX : IsArithVar p N) (ι : k →+* AlgebraicClosure ℚ) (x : X) [CharZero (X.residueField x)] {S : Set X} (hS : IsLocallyClosed S) (hx : x ∈ S) : ∃ z : GeomPt p ι, z.base ∈ S

theorem **exists_geomPt** (hX : IsArithVar p N) (ι : k →+* AlgebraicClosure ℚ) {x : X} (hx : IsAlgPt x) : ∃ z : GeomPt p ι, z.base = x

theorem **dense_range_base** [IsIntegral X] (hX : IsArithVar p N) (ι : k →+* AlgebraicClosure ℚ) : Dense (Set.range (GeomPt.base : GeomPt p ι → X))

theorem **isProper_of_comp** (hX : IsArithVar p N) {N' : ℕ} (hX' : IsArithVar q N') (π : X' ⟶ X) (hπ : π ≫ p = q) : IsProper π

theorem **dense_preimage_push** [IsIntegral X] [IsIntegral X'] (hX : IsArithVar p N) {N' : ℕ} (hX' : IsArithVar q N') (π : X' ⟶ X) (hπ : π ≫ p = q) [Surjective π] (hd : 0 < π.residueDegree (genericPoint X')) {Z : Set (GeomPt p ι)} (hZ : Dense (GeomPt.base '' Z)) : Dense (GeomPt.base '' (GeomPt.push π hπ ⁻¹' Z))

theorem **TieArOn.mono** {A : SlopeTestRemainingChoices Y} {S T : Set Y.ALB} (h : TieArOn A T) (hS : S ⊆ T) : TieArOn A S

theorem **approx_mono** (A : SlopeTestRemainingChoices Y) {s t : ℝ} (h : s ≤ t) : approx A s ⊆ approx A t

theorem **TieArOn.tieAr** {A : SlopeTestRemainingChoices Y} (h : TieArOn A Set.univ) : TieAr A

theorem **TieAr.on_pos** {A : SlopeTestRemainingChoices Y} (h : TieAr A) : TieArOn A {L | Y.semipos L ∧ Y.nefFib L}

theorem **TieAr.on_approx** {A : SlopeTestRemainingChoices Y} (h : TieAr A) : ∃ s₀ : ℝ, 0 < s₀ ∧ TieArOn A (approx A s₀)

theorem **IsBundleModel.exists_rel** {A : SlopeTestRemainingChoices Y} {c : Y.Comp} {m : ArMod (dimY Y)} {b : Y.Pt → m.X} {L : Y.ALB} {sp nf : Prop} (h : IsCompModel A c m b) (hb : IsBundleModel A c m b L sp nf) : ∃ (m' : ArMod (dimY Y)) (H : HermLB m'.p), (sp ↔ H.SemiPos (dimY Y)) ∧ (nf ↔ NefFib H) ∧ IsRelModel A c L m' H

theorem **TieArOn.exists_family** {A : SlopeTestRemainingChoices Y} {S : Set Y.ALB} (h : TieArOn A S) {L : Y.ALB} (hL : L ∈ S) : ∃ (m : Y.Comp → ArMod (dimY Y)) (H : ∀ c, HermLB (m c).p), (Y.semipos L ↔ ∀ c, (H c).SemiPos (dimY Y)) ∧ (Y.nefFib L ↔ ∀ c, NefFib (H c)) ∧ ∀ c, IsRelModel A c L (m c) (H c)

theorem **TieArOn.exists_rel** {A : SlopeTestRemainingChoices Y} {S : Set Y.ALB} (h : TieArOn A S) {L : Y.ALB} (hL : L ∈ S) (c : Y.Comp) : ∃ (m : ArMod (dimY Y)) (H : HermLB m.p) (n : ℕ) (ν : ℝ) (r : m.X → Y.Pt → Prop), 0 < n ∧ 0 < ν ∧ H.IsSelfInter (ν * (n : ℝ) ^ (dimY Y + 1) * Y.arSelfC L c) ∧ gdeg H (dimY Y) = ν * (n : ℝ) ^ dimY Y * Y.geoC L c ∧ (Y.semipos L → H.SemiPos (dimY Y)) ∧ (Y.nefFib L → NefFib H) ∧ (∀ x, IsAlgPt x → ∃ y, r x y) ∧ (∀ y, Y.compPt y = c → ∃ x, r x y) ∧ (∀ x y, r x y → IsAlgPt x ∧ Y.compPt y = c ∧ IsPtHt H x (n * A.ht L y)) ∧ ∀ Z, A.dense c Z → Dense {x : m.X | ∃ y ∈ Z, r x y}

theorem **TieAr.exists_rel** {A : SlopeTestRemainingChoices Y} (h : TieAr A) {L : Y.ALB} (hs : Y.semipos L) (hn : Y.nefFib L) (c : Y.Comp) : ∃ (m : ArMod (dimY Y)) (H : HermLB m.p), H.SemiPos (dimY Y) ∧ NefFib H ∧ IsRelModel A c L m H

theorem **IsCompModel.dense_comp** {A : SlopeTestRemainingChoices Y} {c : Y.Comp} {m : ArMod (dimY Y)} {b : Y.Pt → m.X} (h : IsCompModel A c m b) : A.dense c (Y.compPt ⁻¹' {c}) ∧ ¬ A.dense c ∅

theorem **TieArOn.thick** {A : SlopeTestRemainingChoices Y} {S : Set Y.ALB} (h : TieArOn A S) (F : Type) [Nonempty F] : TieArOn (thickeningHeights A F) S

theorem **tieArOn_univ** (hE : GlobalTheory.Published.ArExists) (D : D7) : TieArOn (aux D) Set.univ

theorem **tieAr** (hE : GlobalTheory.Published.ArExists) (D : D7) : TieAr (aux D)

theorem **tieAr_thick** (hE : GlobalTheory.Published.ArExists) (D : D7) : TieAr (thickeningHeights (aux D) ℕ)

theorem **mem_approx** (D : D7) {s₀ : ℝ} (h : 0 < s₀) : (fun _ => (1, 0) : (shim D).ALB) ∈ approx (aux D) s₀

Development.AbelianVarietyTheory.Definitions.Divisors.EqZeroOfForallPowDvd

theorem **In.top_iff** {f : MvPowerSeries σ Q} : In (⊤ : Ideal B) f ↔ ∃ s : MvPowerSeries σ B, map (algebraMap B Q) s = f

theorem **hasSubst_up_nmul** {μ : Fin d → MvPowerSeries (Fin d ⊕ Fin d) B} (hμ : ∀ i, constantCoeff (μ i) = 0) (n : ℕ) : HasSubst (up Q (nmul μ n))

theorem **IsFrame.eq_zero_of_forall_pow_dvd** (h : IsFrame p Q θ) {x : B} (hx : ∀ n : ℕ, (p : B) ^ n ∣ x) : x = 0

theorem **HasVal.unique** [Fintype σ] [DecidableEq σ] (h : IsFrame p Q θ) {z : σ → B} {f : MvPowerSeries σ Q} {q q' : Q} (h1 : HasVal p z f q) (h2 : HasVal p z f q') : q = q'

theorem **HasVal.var** [Fintype σ] [DecidableEq σ] (z : σ → B) (i : σ) : HasVal p z (X i : MvPowerSeries σ Q) (algebraMap B Q (z i))

theorem **hasPeriod_iff** [Algebra B Q] {θ : B →+* Q} (h : IsFrame p Q θ) (t : G.Tate) (f : MvPowerSeries (Fin d) Q) (y : B) : c.HasPeriod θ t f y ↔ ∀ (n : ℕ) (z : Fin d → B), (∀ i, θ (z i) = t.pt n (c.x n i)) → ∃ q : Q, HasVal p z f q ∧ ∃ b : B, (p : Q) ^ n * q = algebraMap B Q (y + p ^ n * b)

theorem **HasPeriod.unique** [Algebra B Q] {θ : B →+* Q} (h : IsFrame p Q θ) {t : G.Tate} {f : MvPowerSeries (Fin d) Q} {y y' : B} (h1 : c.HasPeriod θ t f y) (h2 : c.HasPeriod θ t f y') : y = y'

theorem **hasPeriod_X_iff** [Algebra B Q] {θ : B →+* Q} (h : IsFrame p Q θ) (t : G.Tate) (i : Fin d) (y : B) : c.HasPeriod θ t (X i : MvPowerSeries (Fin d) Q) y ↔ y = 0

theorem **isLie_iff** [Algebra O Q] (h : Function.Injective (algebraMap O Q)) (f : MvPowerSeries (Fin d) Q) (l : G.dual.Lie) : c.IsLie f l ↔ ∀ n : ℕ, ∃ s : MvPowerSeries (Fin d) O, map (algebraMap O Q) s = defect c.μ (p ^ n) f ∧ ∀ (w : G.dual.omega) (φ : CartierDual O (G.A n)), PDiv.dUnit O φ = w.1 n → (p : O) ^ n ∣ l w − φ (c.ev n s)

theorem **IsLie.unique** [Algebra O Q] [IsHausdorff (Ideal.span {(p : O)}) O] {f : MvPowerSeries (Fin d) Q} {l l' : G.dual.Lie} (h1 : c.IsLie f l) (h2 : c.IsLie f l') : l = l'

theorem **isLie_zero** [Algebra O Q] (h : Function.Injective (algebraMap O Q)) : c.IsLie (0 : MvPowerSeries (Fin d) Q) 0

theorem **isFrame_padic** (p : ℕ) [Fact p.Prime] : IsFrame p ℚ_[p] (RingHom.id ℤ_[p]) where

Development.AbelianVarietyTheory.Definitions.Divisors.EqZeroOfHasRepUnit

theorem **fam_ev** {U T : Over S} {M : LB (A ⊗ U).left} (h : Fam A M) (u : T ⟶ U) : Fam A (ev M (fst A T) (snd A T ≫ u))

theorem **rep_ext** {B S' Y : Over S} {b : B ⟶ S'} {P : LB (A ⊗ B).left} (h : Rep A b P) {u u' : Y ⟶ B} (hb : u ≫ b = u' ≫ b) (e : Same A P P u u') : u = u'

theorem **isChart_of_rep** {B : Over S} {P : LB (A ⊗ B).left} (h : Rep A (toUnit B) P) : IsChart A P where

theorem **rep_baseChange** {B : Over S} {P : LB (A ⊗ B).left} (h : Rep A (toUnit B) P) (S' : Over S) : Rep A (snd B S') (ev P (fst A _) (snd A _ ≫ fst B S')) where

theorem **liftAt_baseChange** {g : ℕ} {B : Over S} {P : LB (A ⊗ B).left} (h : Fam A P) (hg : SmoothOfRelativeDimension g B.hom) (S' : Over S) (x : (B ⊗ S').left) : LiftAt A g (snd B S') (ev P (fst A _) (snd A _ ≫ fst B S')) x

theorem **good_snd** {g : ℕ} {B : Over S} (h : Good g B.hom) (S' : Over S) : Good g (snd B S').left

theorem **hasRep_of_dual** (H : GlobalTheory.Published.DualAbelianScheme) (hA : IsAbelianScheme A) {g : ℕ} (hg : SmoothOfRelativeDimension g A.hom) (S' : Over S) : HasRep A g S'

theorem **hasRepAF_unit** (S' : Over S) [IsAffine S'.left] : HasRepAF (𝟙_ (Over S)) 0 S'

theorem **isIso_of_rep_unit** {B S' : Over S} {b : B ⟶ S'} {P : LB (𝟙_ (Over S) ⊗ B).left} (h : Rep (𝟙_ (Over S)) b P) : IsIso b

theorem **eq_zero_of_hasRep_unit** {g : ℕ} {S' : Over S} (hS : Nonempty S'.left) (h : HasRep (𝟙_ (Over S)) g S') : g = 0

theorem **inj_of_isPerfect** {B : Over S} [GrpObj B] {P : LB (A ⊗ B).left} (h : IsPerfect A B P) (ha : IsAdditive (swap P)) : Inj A P

theorem **fibInj_of_inj** {U : Over S} {M : LB (A ⊗ U).left} (h : Inj A M) : FibInj A M

theorem **eq_of_inj_triv** {U Y : Over S} (h : Inj A (LB.triv (A ⊗ U).left)) (u u' : Y ⟶ U) : u = u'

theorem **dimLe_self** : DimLe A A

theorem **ell** {R : Type} [CommRing R] [IsNoetherianRing R] (W : WeierstrassCurve R) [W.IsElliptic] : haveI

theorem **ell_rep** (H : GlobalTheory.Published.DualAbelianScheme) {R : Type} [CommRing R] [IsNoetherianRing R] (W : WeierstrassCurve R) [W.IsElliptic] : haveI

theorem **same_unit** {T U Y : Over S} {N : LB (𝟙_ (Over S) ⊗ T).left} {M : LB (𝟙_ (Over S) ⊗ U).left} (hN : Fam (𝟙_ (Over S)) N) (hM : Fam (𝟙_ (Over S)) M) (y : Y ⟶ T) (u : Y ⟶ U) : Same (𝟙_ (Over S)) N M y u

theorem **isChart_unit_iff** {U : Over S} {M : LB (𝟙_ (Over S) ⊗ U).left} (hM : Fam (𝟙_ (Over S)) M) : IsChart (𝟙_ (Over S)) M ↔ IsOpenImmersion U.hom

theorem **isChart_open** {U V : Over S} {M : LB (A ⊗ U).left} (h : IsChart A M) (j : V ⟶ U) [IsOpenImmersion j.left] : IsChart A (ev M (fst A V) (snd A V ≫ j))

theorem **not_dimLe_unit_ell** {R : Type} [CommRing R] [Nontrivial R] (W : WeierstrassCurve R) [W.IsElliptic] : ¬ DimLe (WeierstrassCurve.ellPolAV W).A (𝟙_ _)

theorem **not_rep_ell_unit** {R : Type} [CommRing R] [IsNoetherianRing R] [Nontrivial R] (W : WeierstrassCurve R) [W.IsElliptic] (P : LB ((WeierstrassCurve.ellPolAV W).A ⊗ 𝟙_ _).left) : ¬ Rep (WeierstrassCurve.ellPolAV W).A (toUnit _) P

Development.AbelianVarietyTheory.Definitions.Divisors.EtaleEmpty

theorem **etale_empty** (A : WeightedAtlas n Y ι La) (z : Y → V n) : Etale A ∅ z

Development.AbelianVarietyTheory.Definitions.Divisors.EvenDefect

theorem **Half.refines** {Q : Data R} {ρ : R →+* R'} {D : DegPre R'} {e : D.X ≃+ Q.X} (h : Half Q ρ D e fun _ _ => 1) : Refines Q ρ D e

Development.AbelianVarietyTheory.Definitions.Divisors.EventuallyMemIntMatIff

theorem **AlmostSph.finite** {φf : ∀ v, GLoc σ Q v → ℂ} (h : AlmostSph σ Q φf) (g : GAd σ Q) : (Function.mulSupport fun v => φf v (proj σ Q v g)).Finite

theorem **AlmostInt.finite** {φ : ∀ v : HeightOneSpectrum (𝒪 F), (Fin m → Fin n → Fv F v ⊗[F] E) → ℂ} (h : AlmostInt φ) (y : Fin m → Fin n → Af F ⊗[F] E) : (Function.mulSupport fun v => φ v fun i j => Algebra.TensorProduct.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F v) (AlgHom.id F E) (y i j)).Finite

theorem **indicator_intMat_mem_SB** (v : HeightOneSpectrum (𝒪 F)) : (Set.indicator (intMat F E m n v) 1 : _ → ℂ) ∈ WeilM.SB (Fin m → Fin n → Fv F v ⊗[F] E)

theorem **matCoord_map** {d : ℕ} (b : Module.Basis (Fin d) F E) {R R' : Type} [CommRing R] [Algebra F R] [TopologicalSpace R] [IsTopologicalRing R] [CommRing R'] [Algebra F R'] [TopologicalSpace R'] [IsTopologicalRing R'] (φ : R →ₐ[F] R') (x : Fin m → Fin n → R ⊗[F] E) : matCoord b R' (fun i j => Algebra.TensorProduct.map φ (AlgHom.id F E) (x i j)) = fun k => φ (matCoord b R x k)

theorem **eventually_mem_intMat_iff** {d : ℕ} (b : Module.Basis (Fin d) F E) : ∀ᶠ v : HeightOneSpectrum (𝒪 F) in Filter.cofinite, ∀ x, x ∈ intMat F E m n v ↔ ∀ k, matCoord b (Fv F v) x k ∈ v.adicCompletionIntegers F

theorem **finProd_mem_SB** {φ : ∀ v : HeightOneSpectrum (𝒪 F), (Fin m → Fin n → Fv F v ⊗[F] E) → ℂ} (hφ : ∀ v, φ v ∈ WeilM.SB (Fin m → Fin n → Fv F v ⊗[F] E)) (h : AlmostInt φ) : finProd φ ∈ WeilM.SB (Fin m → Fin n → Af F ⊗[F] E)

theorem **finProd_indicator** : finProd (fun v => Set.indicator (intMat F E m n v) 1) = Set.indicator {y | ∀ v i j, Algebra.TensorProduct.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F v) (AlgHom.id F E) (y i j) ∈ intLoc F E v} 1

theorem **compFin_eq_map_sndA** (v : HeightOneSpectrum (𝒪 F)) (z : AF F ⊗[F] E) : compFin F E v z = Algebra.TensorProduct.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F v) (AlgHom.id F E) (Algebra.TensorProduct.map (sndA F) (AlgHom.id F E) z)

theorem **gauss_eq_prod** (V : HSp D) (l : D.what₂.Tau → ℝ) (s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ) (z : D.what₂.Tau → Matrix (Fin V.rank) (Fin n) ℂ) : StdUnn.gauss V l s z = ∏ τ, gaussAt V l s τ (z τ)

theorem **evalPoly_rename** {m : ℕ} (τ : D.what₂.Tau) (p : MvPolynomial (Fin m × Fin n × Bool) ℂ) (z : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ) : StdUnn.evalPoly (MvPolynomial.rename (Prod.mk τ) p) z = evalPolyAt p (z τ)

theorem **compFin_eq** (v : HeightOneSpectrum (𝒪 D.host.F)) (z : AE D) : AdelicU.compFin D.host.F D.host.E v z = Algebra.TensorProduct.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV D.host.F v) (AlgHom.id D.host.F D.host.E) (finPart D z)

theorem **pureFun_eq** (p : D.what₂.Tau → MvPolynomial (Fin V.rank × Fin n × Bool) ℂ) (φ : ∀ v : HeightOneSpectrum (𝒪 D.host.F), (Fin V.rank → Fin n → Fv D.host.F v ⊗[D.host.F] D.host.E) → ℂ) (x : Mx D V.rank n) : pureFun θ V l s p φ x = StdUnn.evalPoly (∏ τ, MvPolynomial.rename (Prod.mk τ) (p τ)) (fun τ => compAt θ τ x) * StdUnn.gauss V l s (fun τ => compAt θ τ x) * finProd φ fun i j => finPart D (x i j)

theorem **IsFactorizable.mem_fockA** {Φ : Mx D V.rank n → ℂ} (h : IsFactorizable θ V n l s Φ) : Φ ∈ StdUnn.fockA θ V n l s

theorem **isFactorizable_vacFun** : IsFactorizable θ V n l s (vacFun θ V n l s)

theorem **vacFun_zero** : vacFun θ V n l s 0 = 1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsAlgHomOfComp

theorem **exists_algHom_of_comp** {K S A B : Type} [CommRing K] [CommRing S] [Algebra K S] [CommRing A] [Algebra K A] [CommRing B] [Algebra K B] (f : S →ₐ[K] A) (g : S →ₐ[K] B) (χ : A →+* B) (h : ∀ r, χ (f r) = g r) : ∃ χ' : A →ₐ[K] B, (∀ a, χ' a = χ a) ∧ χ'.comp f = g

theorem **glue_val** {K S : Type} [CommRing K] [CommRing S] [Algebra K S] {Z : Scheme.{0}} (z : Sp S ⟶ Z) {J : Type} {U : J → Z.Opens} (hc : ⨆ k, U k = ⊤) (val : ∀ (S' : Type) [CommRing S'] [Algebra K S'] (ψ : S →ₐ[K] S') (k : J), ⊤ ≤ (spMap ψ.toRingHom ≫ z) ⁻¹ᵁ U k → S') (nat : ∀ (S' S'' : Type) [CommRing S'] [Algebra K S'] [CommRing S''] [Algebra K S''] (ψ : S →ₐ[K] S') (χ : S' →ₐ[K] S'') (k k' : J) (h : ⊤ ≤ (spMap ψ.toRingHom ≫ z) ⁻¹ᵁ U k) (h' : ⊤ ≤ (spMap (χ.comp ψ).toRingHom ≫ z) ⁻¹ᵁ U k'), val S'' (χ.comp ψ) k' h' = χ (val S' ψ k h)) : ∃! a : S, ∀ (S' : Type) [CommRing S'] [Algebra K S'] (ψ : S →ₐ[K] S') (k : J) (h : ⊤ ≤ (spMap ψ.toRingHom ≫ z) ⁻¹ᵁ U k), ψ a = val S' ψ k h

theorem **liftTan_univ** (y : Sp S ⟶ Xo.XF c) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt (y ≫ pullback.fst _ _) w) : liftTan Y y hw ≫ Y.m0.toSABMap.φ = w

theorem **liftTan_over** (y : Sp S ⟶ Xo.XF c) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt (y ≫ pullback.fst _ _) w) : liftTan Y y hw ≫ Y.oF.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ y

theorem **isTanAtG_liftTan** (y : Sp S ⟶ Xo.XF c) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt (y ≫ pullback.fst _ _) w) : IsTanAtG Y.t (y ≫ Y.j) (liftTan Y y hw ≫ Y.m.φ)

theorem **toModel_left** (y : Sp S ⟶ Xo.XF c) {x₀ : Sp S ⟶ Xo.X} (h₀ : y ≫ pullback.fst _ _ = x₀) {x : Sp S ⟶ Y.X} (hx : y ≫ Y.j = x) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt x₀ w) : (tv (toModel Y y h₀ hx hw)).left = liftTan Y y (h₀ ▸ hw) ≫ Y.m.φ

theorem **toModel_eq_iff** (y : Sp S ⟶ Xo.XF c) {x₀ : Sp S ⟶ Xo.X} (h₀ : y ≫ pullback.fst _ _ = x₀) {x : Sp S ⟶ Y.X} (hx : y ≫ Y.j = x) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt x₀ w) (v : TanM x Y.t.G) : toModel Y y h₀ hx hw = v ↔ ∃ u : Sp (DualNumber S) ⟶ Y.oF.t.G.left, u ≫ Y.oF.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ y ∧ u ≫ Y.m0.toSABMap.φ = w ∧ u ≫ Y.m.φ = (tv v).left

theorem **toModel_injective** (y : Sp S ⟶ Xo.XF c) {x₀ : Sp S ⟶ Xo.X} (h₀ : y ≫ pullback.fst _ _ = x₀) {x : Sp S ⟶ Y.X} (hx : y ≫ Y.j = x) {w w' : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt x₀ w) (hw' : IsTanAt x₀ w') (e : toModel Y y h₀ hx hw = toModel Y y h₀ hx hw') : w = w'

theorem **tanPull_toModel** (φ : S →+* S') (y : Sp S ⟶ Xo.XF c) {x₀ : Sp S ⟶ Xo.X} (h₀ : y ≫ pullback.fst _ _ = x₀) {x : Sp S ⟶ Y.X} (hx : y ≫ Y.j = x) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt x₀ w) {y' : Sp S' ⟶ Xo.XF c} (ey : spMap φ ≫ y = y') {x₀' : Sp S' ⟶ Xo.X} (h₀' : y' ≫ pullback.fst _ _ = x₀') {x' : Sp S' ⟶ Y.X} (hx' : y' ≫ Y.j = x') (e : spMap φ ≫ x = x') {w' : Sp (DualNumber S') ⟶ Xo.univ.t.G.left} (hw' : IsTanAt x₀' w') (ew : w' = spMap (dualMap φ) ≫ w) : tanPull φ e (toModel Y y h₀ hx hw) = toModel Y y' h₀' hx' hw'

theorem **xF_ext** {y y' : Sp S ⟶ Xo.XF c} (hy : y ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (hy' : y' ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (e : y ≫ Y.j = y' ≫ Y.j) : y = y'

theorem **vec_iff_toModel** {M : MChart Y} {x : Sp S ⟶ Y.X} (h : ⊤ ≤ x ⁻¹ᵁ M.U) {x₀ : Sp S ⟶ Xo.X} (Λ : M.Lift (ev x M.U h) x₀) {y : Sp S ⟶ Xo.XF c} (ey : Λ.xF = y) (h₀ : y ≫ pullback.fst _ _ = x₀) (hx : y ≫ Y.j = x) (cv : Fin M.n → S) {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hw : IsTanAt x₀ w) : Λ.Vec cv w ↔ toModel Y y h₀ hx hw = vec M x h cv

theorem **frX_fst** (P : FrPt Xo c S) : frX P ≫ pullback.fst _ _ = P.x

theorem **frX_snd** (P : FrPt Xo c S) : frX P ≫ pullback.snd _ _ = spMap (algebraMap c.F S)

theorem **eq_frX** (P : FrPt Xo c S) {y : Sp S ⟶ Xo.XF c} (h₁ : y ≫ pullback.fst _ _ = P.x) (h₂ : y ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) : y = frX P

theorem **frX_frMap** [Algebra c.F S'] {ψ : S →ₐ[c.F] S'} {P : FrPt Xo c S} {P' : FrPt Xo c S'} (h : FrMap ψ P P') : spMap ψ.toRingHom ≫ frX P = frX P'

theorem **tanPull_frVec** [Algebra c.F S'] {ψ : S →ₐ[c.F] S'} {P : FrPt Xo c S} {P' : FrPt Xo c S'} (h : FrMap ψ P P') {x : Sp S ⟶ Y.X} (hx : frX P ≫ Y.j = x) {x' : Sp S' ⟶ Y.X} (hx' : frX P' ≫ Y.j = x') (e : spMap ψ.toRingHom ≫ x = x') (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : tanPull ψ.toRingHom e (frVec Y P hx σ i) = frVec Y P' hx' σ i

theorem **vec_mem_eigLie** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (σ : E →+* ℂ) (cv : Fin M.n → S) : vec M x h cv ∈ eigLie c x σ ↔ M.Eig (ev x M.U h) σ cv

theorem **ev_appTop** {Z Z' : Scheme.{0}} (z : Sp S ⟶ Z') (π : Z' ⟶ Z) (a : Γ(Z, ⊤)) : ev z ⊤ le_top (π.appTop.hom a) = ev (z ≫ π) ⊤ le_top a

theorem **spMap_ev_top** (ρ : Sp T ⟶ Sp S) : spMap ((ev ρ ⊤ le_top).comp (toΓ S)) = ρ

theorem **isoMaps_secMul** {Z : Scheme.{0}} {L N : LB Z} (w : LB.Iso L N) {s : L.Sec} {s' : N.Sec} (h : IsoMaps w s s') (g : Γ(Z, ⊤)) : IsoMaps w (secMul g s) (secMul g s')

theorem **ev_res_secMul** {Z : Scheme.{0}} {L : LB Z} (g : Γ(Z, ⊤)) (s : L.Sec) (k : L.ι) {W : Z.Opens} (h : W ≤ L.U k) (r : Sp T ⟶ Z) (hr : ⊤ ≤ r ⁻¹ᵁ W) : ev r W hr (resG Z h ((secMul g s).f k)) = ev r ⊤ le_top g * ev r W hr (resG Z h (s.f k))

theorem **appLE_res_appTop** {Z Z' B : Scheme.{0}} (g : Z' ⟶ Z) (p : Z ⟶ B) (p' : Z' ⟶ B) (e : g ≫ p = p') {U : Z.Opens} {V : Z'.Opens} (h : V ≤ g ⁻¹ᵁ U) (b : Γ(B, ⊤)) : resG Z' le_top (p'.appTop.hom b) = (g.appLE U V h).hom (resG Z le_top (p.appTop.hom b))

theorem **thRel_cast** {x₀ : Sp S ⟶ Xo.X} {x x' : Sp S ⟶ Y.X} (e : x = x') {t' : SAB H (Sp S)} (m1 : SABMap t' Xo.univ.t x₀) (m2 : SABMap t' Y.t x) : ∃ m2' : SABMap t' Y.t x', m2'.φ = m2.φ ∧ ∀ U θ s, ThRel m1 m2 U θ s → ThRel m1 m2' U θ s

theorem **thRel_change** (M M' : GChart Y) {x : Sp S ⟶ Y.X} (hh : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) {x₀ : Sp S ⟶ Xo.X} {t' : SAB H (Sp S)} (m1 : SABMap t' Xo.univ.t x₀) (m2 : SABMap t' Y.t x) {s : LB.Sec (PPx x₀)} (hs : ThRel m1 m2 M.U M.θ s) : ThRel m1 m2 M'.U M'.θ (secMul ((fst (Over.mk x₀) Xo.univ.t.G).left.appTop.hom (toΓ S (ev x _ hh (thRat M M')))) s)

theorem **thTie_change** (M M' : GChart Y) {x : Sp S ⟶ Y.X} (h : ⊤ ≤ x ⁻¹ᵁ M.U) (h' : ⊤ ≤ x ⁻¹ᵁ M'.U) (hh : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) {x₀ : Sp S ⟶ Xo.X} (Λ : M.toMChart.Lift (ev x M.U h) x₀) (Λ' : M'.toMChart.Lift (ev x M'.U h') x₀) {s s' : LB.Sec (PPx x₀)} (hs : Λ.ThTie s) (hs' : Λ'.ThTie s') (i : (PPx x₀).ι) : s'.f i = scal x₀ (ev x _ hh (thRat M M')) i * s.f i

theorem **frPt_eq_of_tie** {M : MChart Y} {x : Sp S ⟶ Y.X} (h : ⊤ ≤ x ⁻¹ᵁ M.U) {u : ∀ τ : E →+* ℂ, Matrix (Fin M.n) (Fin (Dw.mult τ)) S} {P Q : FrPt Xo c S} (hP : M.Comparison₂ (ev x M.U h) u P) (hQ : M.Comparison₂ (ev x M.U h) u Q) : P = Q

theorem **frBor_of_frVec** {M : MChart Y} {x : Sp S ⟶ Y.X} (h : ⊤ ≤ x ⁻¹ᵁ M.U) {P P' : FrPt Xo c S} (hx : frX P ≫ Y.j = x) (hx' : frX P' ≫ Y.j = x) (t : Ix Dw → Sˣ) (b : ∀ σ : E →+* ℂ, Matrix (Fin (Dw.mult σ)) (Fin (Dw.mult σ)) S) (hb : ∀ σ, (b σ).BlockTriangular id) (hd : ∀ σ i, b σ i i = t (some ⟨σ, i⟩)) (hv : ∀ σ i, frVec Y P' hx' σ i = ∑ j, b σ j i • frVec Y P hx σ j) (hs : ∀ e : P'.x = P.x, ∀ i, (e ▸ P'.s).f i = scal P.x (t none : S) i * P.s.f i) : FrBor t P P'

theorem **frVec_of_frBor** {M : MChart Y} {x : Sp S ⟶ Y.X} (h : ⊤ ≤ x ⁻¹ᵁ M.U) {P P' : FrPt Xo c S} (hx : frX P ≫ Y.j = x) (hx' : frX P' ≫ Y.j = x) {t : Ix Dw → Sˣ} (hB : FrBor t P P') (σ : E →+* ℂ) : ∃ b : Matrix (Fin (Dw.mult σ)) (Fin (Dw.mult σ)) S, b.BlockTriangular id ∧ (∀ i, b i i = t (some ⟨σ, i⟩)) ∧ ∀ i, frVec Y P' hx' σ i = ∑ j, b j i • frVec Y P hx σ j

theorem **frX_frBor** {P P' : FrPt Xo c S} {t : Ix Dw → Sˣ} (hB : FrBor t P P') : frX P' = frX P

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsAlgHomOfIsBaseChange

theorem **extProj_coe** (D : ℕ) (x : ∧[A]^D M) : extProj A M D x = x

theorem **exists_algHom_of_isBaseChange** {φ : M →ₗ[A] N} (hφ : IsBaseChange B φ) : ∃ F : ExteriorAlgebra B N →ₐ[B] B ⊗[A] ExteriorAlgebra A M, ∀ m, F (ExteriorAlgebra.ι B (φ m)) = 1 ⊗ₜ ExteriorAlgebra.ι A m

theorem **isBaseChange_exteriorPower** {φ : M →ₗ[A] N} (hφ : IsBaseChange B φ) (D : ℕ) : IsBaseChange B (exteriorPowerMapOver B D φ)

theorem **IsFreeGen.of_isBaseChange** {φ : M →ₗ[A] N} (hφ : IsBaseChange B φ) {ω : M} (h : IsFreeGen A ω) : IsFreeGen B (φ ω)

theorem **isFreeGen_ιMulti** {D : ℕ} (b : Module.Basis (Fin D) A M) : IsFreeGen A (exteriorPower.ιMulti A D b)

theorem **IsFreeGen.repr_smul** (h : IsFreeGen A ω) (x : M) : h.repr x • ω = x

theorem **IsFreeGen.unit_smul** (h : IsFreeGen A ω) (u : Aˣ) : IsFreeGen A ((u : A) • ω)

theorem **IsFreeGen.smul_iff** (h : IsFreeGen A ω) {a : A} : IsFreeGen A (a • ω) ↔ IsUnit a

theorem **dForm_eq** (α : k →+* A) (f : Fin D → A) : dForm α f = exteriorPower.ιMulti A D fun i => dFun α (f i)

theorem **span_dForm** (α : k →+* A) (D : ℕ) : Submodule.span A (Set.range (dForm α (D

@[simp] theorem **formMap_dForm** (ρ : A →+* B) (h : ρ.comp α = β) (f : Fin D → A) : formMap D ρ h (dForm α f) = dForm β fun i => ρ (f i)

theorem **forms_ext** {M : Type*} [AddCommGroup M] [Module B M] {ρ : A →+* B} {φ ψ : Forms α D →ₛₗ[ρ] M} (h : ∀ f, φ (dForm α f) = ψ (dForm α f)) : φ = ψ

theorem **formMap_formMap** (ρ : A →+* B) (σ : B →+* C) (τ : A →+* C) (e : σ.comp ρ = τ) (h : ρ.comp α = β) (h' : σ.comp β = γ) (h'' : τ.comp α = γ) (x : Forms α D) : formMap D σ h' (formMap D ρ h x) = formMap D τ h'' x

theorem **formMap_of_eq_id** (ρ : A →+* A) (e : ρ = RingHom.id A) (h : ρ.comp α = α) (x : Forms α D) : formMap D ρ h x = x

theorem **IsFreeGen.formMap** {ρ : A →+* B} (hρ : ρ.Etale) (h : ρ.comp α = β) {ω : Forms α D} (hω : IsFreeGen A ω) : IsFreeGen B (formMap D ρ h ω)

theorem **isFreeGen_dForm** (x : Fin D → A) (b : Module.Basis (Fin D) A (Omega α)) (hb : ∀ i, b i = dFun α (x i)) : IsFreeGen A (dForm α x)

theorem **exists_isFreeGen** (h : α.IsStandardSmoothOfRelativeDimension D) : ∃ ω : Forms α D, IsFreeGen A ω

theorem **map_dForm_mul_sub_mem** (c : Forms α D →ₗ[A] A) (I : Ideal A) (h : A) (g : Fin D → A) (hg : ∀ i, g i ∈ I) : c (dForm α fun i => h * g i) - h ^ D * c (dForm α g) ∈ I

theorem **dForm_eq_det_smul** (f x : Fin D → A) (a : Matrix (Fin D) (Fin D) A) (h : ∀ i, dFun α (f i) = ∑ j, a i j • dFun α (x j)) : dForm α f = a.det • dForm α x

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsArchOne

theorem **ArchOne.upper** {X : Data E r} {L Fp : Wt X.Ψ r} {T : ℕ} (h : ArchOne X L Fp T) : ∃ (Xa : ArchInM X) (Fl : FlatM X Xa), Xa.L = L ∧ Xa.Fp = Fp ∧ Xa.T = T ∧ GlobalTheory.Shares.Margins X Xa ∧ ∀ (m : ℕ) (f : Fin m → Frame X.Ψ r → ℂ), (∀ q, f q ∈ X.SS (Fp + T • L)) → (∀ u : Fin m → ℂ, (∀ x ∈ X.dom, ∑ q, u q * f q x = 0) → u = 0) → m ≤ Fl.δ * (T + 2 * ReciprocityTheory.ChartRestriction.G.AR.Inst.nd X + 6).choose (ReciprocityTheory.ChartRestriction.G.AR.Inst.nd X)

theorem **ArchOne.jetInjAt** {X : Data E r} {L Fp : Wt X.Ψ r} {T : ℕ} (h : ArchOne X L Fp T) : ∃ Xa : ArchInM X, Xa.L = L ∧ Xa.Fp = Fp ∧ Xa.T = T ∧ GlobalTheory.Shares.Margins X Xa ∧ ∃ Fl : FlatM X Xa, JetInjAt X Xa Fl

theorem **exists_archOne** (h : Module.finrank ℚ E = 4) : ∃ (X : Data E 2) (L Fp : Wt X.Ψ 2) (T : ℕ), ArchOne X L Fp T

theorem **even_sOf** (M : TwoLevels E r U C) : Even (sOf M)

theorem **le_sOf** (M : TwoLevels E r U C) : 2 * r + 50 * r ^ 2 * M.dimX + 1 ≤ sOf M

theorem **sOf_sub** (M : TwoLevels E r U C) (H : Subgroup U) : sOf (sub M H) = sOf M

theorem **cusp_sub** (M : TwoLevels E r U C) (H : Subgroup U) (la : Wt M.Ψ r) : (sub M H).cusp la = M.cusp la

theorem **IsPullRep.restrict** {M : TwoLevels E r U C} {la : Wt M.Ψ r} {ρ : Representation ℂ U (M.cusp la)} (h : IsPullRep M la ρ) (H : Subgroup U) : IsPullRep (sub M H) la (ρ.comp H.subtype)

theorem **ranks_zero** (M : TwoLevels E r U C) (ka : Wt M.Ψ r) (h : Module.Finite ℂ (M.cusp ka)) : Ranks M ka 0 0 1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsBaseEqChain

theorem **isFrob_frobA** (h : (p : Γ((fibI p B).subscheme, ⊤)) = 0) (M : AM B Λ) : IsFrob p M.V (frobA p h M).V

theorem **sc_mkA** (F : B.X.Modules) : Sc χ (mkA χ F)

theorem **Sc.isAdd** {M : AM B Λ} (h : Sc χ M) : IsAdd M

theorem **Sc.coker** {M N : AM B Λ} (f : M ⟶ N) (hN : Sc χ N) : Sc χ (cokernel f)

theorem **additive_frobFun** (h : (p : Γ((fibI p B).subscheme, ⊤)) = 0) : (frobFun p h).Additive

theorem **Sc.frob** (h : (p : Γ((fibI p B).subscheme, ⊤)) = 0) {M : AM B Λ} (hM : Sc χ M) : Sc χ (frobA p h M)

theorem **frob_act** (h : (p : Γ((fibI p B).subscheme, ⊤)) = 0) {M : AM B Λ} (c : Λ) (g : Γ(B.X, ⊤)) (hM : (M.ρ c : M.V ⟶ M.V) = smulHom M.V g) : ((frobA p h M).ρ c : (frobA p h M).V ⟶ (frobA p h M).V) = smulHom _ (g ^ p)

theorem **smulA_mul** (a b : O) : smulA (sA χ O) (cst O a) ≫ smulA (sA χ O) (cst O b) = smulA (sA χ O) (cst O (b * a))

theorem **isExtA_mul** {a b c : O} (hb : b ≠ 0) (h : b * a = c) : IsExtA (cokernel (smulA (sA χ O) (cst O a))) (cokernel (smulA (sA χ O) (cst O c))) (cokernel (smulA (sA χ O) (cst O b)))

theorem **frob_cok** (h : (p : Γ((fibI p (Model.base O)).subscheme, ⊤)) = 0) {M : AM (Model.base O) Λ} (a : O) (ha : a ^ p ∣ (p : O)) (e : M.V ≅ cokernel (smulHom (sO O) (cst O a))) : Nonempty ((frobA p h M).V ≅ cokernel (smulHom (sO O) (cst O (a ^ p))))

theorem **frob_iter** (h : (p : Γ((fibI p (Model.base O)).subscheme, ⊤)) = 0) {M : AM (Model.base O) Λ} (hM : Sc χ M) (a : O) (e : M.V ≅ cokernel (smulHom (sO O) (cst O a))) (i : ℕ) (ha : a ^ p ^ i ∣ (p : O)) : Sc χ ((frobA p h)^[i] M) ∧ Nonempty (((frobA p h)^[i] M).V ≅ cokernel (smulHom (sO O) (cst O (a ^ p ^ i))))

theorem **exists_eqChain** (o : ℕ) (ϖ : O) (u : Oˣ) (hu : ϖ ^ (p ^ (o + 1) * (p - 1)) = u * p) : ∃ Q : EqChain p (Model.base O) Λ o, Q.lw.r = 1 ∧ Sc χ Q.lw.E' ∧ (∀ n, ¬ IsZero (Q.q n).mod.V) ∧ ∀ n, ¬ IsZero (Q.l n).mod.V

theorem **exists_base_eqChain** (p : ℕ) [Fact p.Prime] (o : ℕ) (Λ : Type) [CommRing Λ] (χ : Λ →+* ℤ) : ∃ (O : Type) (_ : CommRing O) (_ : IsDomain O) (_ : IsDiscreteValuationRing O) (_ : CharZero O) (_ : CharP (ResidueField O) p) (Q : EqChain p (Model.base O) Λ o), Q.lw.r = 1 ∧ Sc χ Q.lw.E' ∧ (∀ n, ¬ IsZero (Q.q n).mod.V) ∧ ∀ n, ¬ IsZero (Q.l n).mod.V

theorem **isExt_of_isExtA** {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] {B : Vert.Model O} {Λ : Type} [CommRing Λ] {P₁ P₂ P₃ : EPres B Λ} (h : IsExtA P₁.mod P₂.mod P₃.mod) : IsExt P₁.toPres.mod P₂.toPres.mod P₃.toPres.mod

theorem **IsEqLatPres.isLatPres** {Z : LocM.Mdl Y i₀} {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] {B : Vert.Model O} {h : B.X ⟶ Z.X} {r : ℕ} {x : Fin r → Letter Dw} {d : Bool} {P : EPres B (𝒪 E)} (k : IsEqLatPres Z h x d P) : IsLatPres Z h x d P.toPres

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsBlockMatrixTwoByTwoFinEq

theorem **blockMatrix_map** (f : A →+* B) (a b c d : Matrix (Fin n) (Fin n) A) : (StdUnn.blk4 a b c d).map f = StdUnn.blk4 (a.map f) (b.map f) (c.map f) (d.map f)

theorem **cT_blk4** (τ : A →+* A) (a b c d : Matrix (Fin n) (Fin n) A) : cT τ (StdUnn.blk4 a b c d) = StdUnn.blk4 (cT τ a) (cT τ c) (cT τ b) (cT τ d)

theorem **cT_zero'** (τ : A →+* A) : cT τ (0 : Matrix (Fin n) (Fin n) A) = 0

theorem **cT_one'** (τ : A →+* A) : cT τ (1 : Matrix (Fin n) (Fin n) A) = 1

theorem **cT_neg_one** (τ : A →+* A) : cT τ (-1 : Matrix (Fin n) (Fin n) A) = -1

theorem **upperLeft_blockMatrix** (a b c d : Matrix (Fin n) (Fin n) A) : StdUnn.blkA (StdUnn.blk4 a b c d) = a

theorem **blockMatrix_mulVec_lower** (a b c d : Matrix (Fin n) (Fin n) A) (x : Fin (n + n) → A) (i : Fin n) : (StdUnn.blk4 a b c d *ᵥ x) (finSumFinEquiv (Sum.inr i)) = (c *ᵥ fun j => x (finSumFinEquiv (Sum.inl j))) i + (d *ᵥ fun j => x (finSumFinEquiv (Sum.inr j))) i

theorem **ext_matR** {g h : HA R σ Q} (e : matR R σ g = matR R σ h) : g = h

theorem **matR_mul** (g h : HA R σ Q) : matR R σ (g * h) = matR R σ g * matR R σ h

theorem **matR_one** : matR R σ (1 : HA R σ Q) = 1

theorem **JR_eq** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : JR (F

@[simp] theorem **matR_mkPt** (M N : Matrix (Fin (n + n)) (Fin (n + n)) (R ⊗[F] E)) (h1 : M * N = 1) (h2 : N * M = 1) (h : cT (sigR R σ) M * JR R Q * M = JR R Q) : matR R σ (mkPt R σ M N h1 h2 h) = M

theorem **cT_herm** (b : HermR R σ n) : cT (sigR R σ) b.1 = b.1

theorem **mat_nLoc** (b : HermR R σ n) : matR R σ (nLoc R σ hQ b) = StdUnn.blk4 1 b.1 0 1

theorem **mat_wLoc** : matR R σ (wLoc R σ hQ) = StdUnn.blk4 0 1 (-1) 0

theorem **nLoc_add** (b b' : HermR R σ n) : nLoc R σ hQ (b + b') = nLoc R σ hQ b * nLoc R σ hQ b'

@[simp] theorem **nLoc_zero** : nLoc R σ hQ (0 : HermR R σ n) = 1

theorem **mem_upperHalfR** {x : Fin (n + n) → R ⊗[F] E} : x ∈ upperHalfR R ↔ ∀ i : Fin n, x (finSumFinEquiv (Sum.inr i)) = 0

theorem **lower_eq_zero** {p : HA R σ Q} (hp : p ∈ upperParabolic R σ Q) (i j : Fin n) : matR R σ p (finSumFinEquiv (.inr i)) (finSumFinEquiv (.inl j)) = 0

theorem **blkA_mulR** {p q : HA R σ Q} (hq : q ∈ upperParabolic R σ Q) : StdUnn.blkA (matR R σ (p * q)) = StdUnn.blkA (matR R σ p) * StdUnn.blkA (matR R σ q)

theorem **coe_leviUp** (p : upperParabolic R σ Q) : (leviUp R σ Q p : Matrix (Fin n) (Fin n) (R ⊗[F] E)) = StdUnn.blkA (matR R σ p.1)

theorem **coe_aUp** (p : upperParabolic R σ Q) : (aUp R σ Q p : R ⊗[F] E) = (StdUnn.blkA (matR R σ p.1)).det

theorem **mem_siegelParabolic_of_blockUpperTriangular** {g : HA R σ Q} {a b d d' : Matrix (Fin n) (Fin n) (R ⊗[F] E)} (hg : matR R σ g = StdUnn.blk4 a b 0 d) (hd : d' * d = 1) : g ∈ upperParabolic R σ Q

theorem **nLoc_mem_upP** (b : HermR R σ n) : nLoc R σ hQ b ∈ upperParabolic R σ Q

@[simp] theorem **aUp_nLoc** (b : HermR R σ n) : aUp R σ Q ⟨nLoc R σ hQ b, nLoc_mem_upP hQ b⟩ = 1

theorem **cT_cT** (hσ : ∀ x, σ (σ x) = x) (M : Matrix (Fin n) (Fin n) (R ⊗[F] E)) : cT (sigR R σ) (cT (sigR R σ) M) = M

theorem **mat_mLoc** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (a : GL (Fin n) (R ⊗[F] E)) : matR R σ (mLoc R σ hσ hQ a) = StdUnn.blk4 a.1 0 0 (cT (sigR R σ) (a⁻¹).1)

theorem **mLoc_mem_upP** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (a : GL (Fin n) (R ⊗[F] E)) : mLoc R σ hσ hQ a ∈ upperParabolic R σ Q

@[simp] theorem **aUp_mLoc** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (a : GL (Fin n) (R ⊗[F] E)) : aUp R σ Q ⟨mLoc R σ hσ hQ a, mLoc_mem_upP R σ hσ hQ a⟩ = Matrix.GeneralLinearGroup.det a

theorem **sigR_map** (φ : R →ₐ[F] R') (y : R ⊗[F] E) : sigR R' σ (Algebra.TensorProduct.map φ (AlgHom.id F E) y) = Algebra.TensorProduct.map φ (AlgHom.id F E) (sigR R σ y)

theorem **mapH_nLoc** (φ : R →ₐ[F] R') (b : HermR R σ n) : GlobalTheory.IndefiniteRowSideFullHecke.AdelicUnitaryGroups.mapH σ Q φ (nLoc R σ hQ b) = nLoc R' σ hQ (hermMap φ b)

theorem **mapH_wLoc** (φ : R →ₐ[F] R') : GlobalTheory.IndefiniteRowSideFullHecke.AdelicUnitaryGroups.mapH σ Q φ (wLoc R σ hQ) = wLoc R' σ hQ

theorem **continuous_nLoc** [TopologicalSpace R] [IsTopologicalRing R] [FiniteDimensional F E] : Continuous (nLoc R σ hQ (n

theorem **mulVec_mem_upperHalfR** {M : Matrix (Fin (n + n)) (Fin (n + n)) (R ⊗[F] E)} (hM : ∀ i j : Fin n, M (finSumFinEquiv (.inr i)) (finSumFinEquiv (.inl j)) = 0) {x : Fin (n + n) → R ⊗[F] E} (hx : x ∈ upperHalfR R) : M *ᵥ x ∈ upperHalfR R

theorem **mem_siegelParabolic_iff** {g : HA R σ Q} : g ∈ upperParabolic R σ Q ↔ ∀ i j : Fin n, matR R σ g (finSumFinEquiv (.inr i)) (finSumFinEquiv (.inl j)) = 0 ∧ matR R σ g⁻¹ (finSumFinEquiv (.inr i)) (finSumFinEquiv (.inl j)) = 0

theorem **exists_blockMatrixTwoByTwoFin_eq** {A : Type} (M : Matrix (Fin (n + n)) (Fin (n + n)) A) : ∃ a b c d, M = StdUnn.blk4 a b c d

theorem **mem_siegelParabolic_iff_lowerLeft_eq_zero** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) {g : HA R σ Q} : g ∈ upperParabolic R σ Q ↔ ∀ i j : Fin n, matR R σ g (finSumFinEquiv (.inr i)) (finSumFinEquiv (.inl j)) = 0

theorem **wLoc_notMem_upP** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) [Nontrivial (R ⊗[F] E)] (hn : 0 < n) : wLoc R σ hQ ∉ upperParabolic R σ Q

theorem **nLoc_injective** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : Function.Injective (nLoc R σ hQ)

theorem **mLoc_mul** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (a a' : GL (Fin n) (R ⊗[F] E)) : mLoc R σ hσ hQ (a * a') = mLoc R σ hσ hQ a * mLoc R σ hσ hQ a'

theorem **mLoc_one** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : mLoc R σ hσ hQ (1 : GL (Fin n) (R ⊗[F] E)) = 1

theorem **matR_mapH** (g : HA R σ Q) : matR R' σ (GlobalTheory.IndefiniteRowSideFullHecke.AdelicUnitaryGroups.mapH σ Q φ g) = (matR R σ g).map (Algebra.TensorProduct.map φ (AlgHom.id F E)).toRingHom

theorem **mapH_mem_upP** {g : HA R σ Q} (hg : g ∈ upperParabolic R σ Q) : GlobalTheory.IndefiniteRowSideFullHecke.AdelicUnitaryGroups.mapH σ Q φ g ∈ upperParabolic R' σ Q

theorem **aUp_mapH** (p : upperParabolic R σ Q) : aUp R' σ Q ⟨GlobalTheory.IndefiniteRowSideFullHecke.AdelicUnitaryGroups.mapH σ Q φ p, mapH_mem_upP φ p.2⟩ = Units.map (Algebra.TensorProduct.map φ (AlgHom.id F E)).toRingHom.toMonoidHom (aUp R σ Q p)

theorem **mapH_mLoc** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (a : GL (Fin n) (R ⊗[F] E)) : GlobalTheory.IndefiniteRowSideFullHecke.AdelicUnitaryGroups.mapH σ Q φ (mLoc R σ hσ hQ a) = mLoc R' σ hσ hQ (bc (Algebra.TensorProduct.map φ (AlgHom.id F E)).toRingHom a)

theorem **continuous_matR** : Continuous (matR R σ (Q

theorem **continuous_leviUp** : Continuous (leviUp R σ Q)

theorem **continuous_aUp** : Continuous (aUp R σ Q)

theorem **isClosed_upP** [T2Space R] : IsClosed (upperParabolic R σ Q : Set (HA R σ Q))

theorem **hgtUp_pos** (g : HA R σ Q) : 0 < hgtUp R σ Q ν K g

theorem **flatUp_apply** (Φ : HA R σ Q → ℂ) (s₀ s : ℂ) (g : HA R σ Q) : flatUp R σ Q ν K Φ s₀ s g = Φ g * npow (hgtUp R σ Q ν K g) (s - s₀)

@[simp] theorem **flatUp_self** (Φ : HA R σ Q → ℂ) (s₀ : ℂ) : flatUp R σ Q ν K Φ s₀ s₀ = Φ

theorem **flatUp_flatUp** (Φ : HA R σ Q → ℂ) (s₀ s₁ s : ℂ) : flatUp R σ Q ν K (flatUp R σ Q ν K Φ s₀ s₁) s₁ s = flatUp R σ Q ν K Φ s₀ s

theorem **sphUp_ne_zero** (s : ℂ) (g : HA R σ Q) : sphUp R σ Q ν K s g ≠ 0

theorem **flatUp_sphUp** (s₀ s : ℂ) : flatUp R σ Q ν K (sphUp R σ Q ν K s₀) s₀ s = sphUp R σ Q ν K s

theorem **hgtUp_mul** (h : IsIwasawaUp R σ Q ν K) (p : upperParabolic R σ Q) {k : HA R σ Q} (hk : k ∈ K) : hgtUp R σ Q ν K (p * k) = ν (aUp R σ Q p)

theorem **hgtUp_left** (h : IsIwasawaUp R σ Q ν K) (p : upperParabolic R σ Q) (g : HA R σ Q) : hgtUp R σ Q ν K (p * g) = ν (aUp R σ Q p) * hgtUp R σ Q ν K g

theorem **hgtUp_right** (h : IsIwasawaUp R σ Q ν K) (g : HA R σ Q) {k : HA R σ Q} (hk : k ∈ K) : hgtUp R σ Q ν K (g * k) = hgtUp R σ Q ν K g

theorem **hgtUp_of_mem** (h : IsIwasawaUp R σ Q ν K) {k : HA R σ Q} (hk : k ∈ K) : hgtUp R σ Q ν K k = 1

theorem **sphUp_of_mem** (h : IsIwasawaUp R σ Q ν K) (s : ℂ) {k : HA R σ Q} (hk : k ∈ K) : sphUp R σ Q ν K s k = 1

theorem **hgtUp_mLoc** (h : IsIwasawaUp R σ Q ν K) (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (a : GL (Fin n) (R ⊗[F] E)) : hgtUp R σ Q ν K (mLoc R σ hσ hQ a) = ν (Matrix.GeneralLinearGroup.det a)

theorem **mem_psUp** {μ : (R ⊗[F] E)ˣ →* Circle} {s : ℂ} {Φ : HA R σ Q → ℂ} : Φ ∈ psUp R σ Q ν K μ s ↔ (∀ (p : upperParabolic R σ Q) (g : HA R σ Q), Φ (p * g) = μ (aUp R σ Q p) * npow (ν (aUp R σ Q p)) (s + (n : ℂ) / 2) * Φ g) ∧ Continuous Φ ∧ IsKFinite K Φ

theorem **apply_nLoc_mul** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) {μ : (R ⊗[F] E)ˣ →* Circle} {s : ℂ} {Φ : HA R σ Q → ℂ} (hΦ : Φ ∈ psUp R σ Q ν K μ s) (b : HermR R σ n) (g : HA R σ Q) : Φ (nLoc R σ hQ b * g) = Φ g

theorem **apply_mLoc_mul** (hσ : ∀ x, σ (σ x) = x) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) {μ : (R ⊗[F] E)ˣ →* Circle} {s : ℂ} {Φ : HA R σ Q → ℂ} (hΦ : Φ ∈ psUp R σ Q ν K μ s) (a : GL (Fin n) (R ⊗[F] E)) (g : HA R σ Q) : Φ (mLoc R σ hσ hQ a * g) = μ (Matrix.GeneralLinearGroup.det a) * npow (ν (Matrix.GeneralLinearGroup.det a)) (s + (n : ℂ) / 2) * Φ g

theorem **intwLoc_eq** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (db : Measure (HermR R σ n)) (f : ℂ → HA R σ Q → ℂ) (s : ℂ) (g : HA R σ Q) : intwLoc R σ hQ db f s g = ∫ b, f s (wLoc R σ hQ * nLoc R σ hQ b * g) ∂db

theorem **intwConv_iff** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (db : Measure (HermR R σ n)) (f : ℂ → HA R σ Q → ℂ) (s : ℂ) (g : HA R σ Q) : IntwConv R σ hQ db f s g ↔ Integrable (fun b => f s (wLoc R σ hQ * nLoc R σ hQ b * g)) db

theorem **intwLoc_add** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) {db : Measure (HermR R σ n)} {f f' : ℂ → HA R σ Q → ℂ} {s : ℂ} {g : HA R σ Q} (h : IntwConv R σ hQ db f s g) (h' : IntwConv R σ hQ db f' s g) : intwLoc R σ hQ db (f + f') s g = intwLoc R σ hQ db f s g + intwLoc R σ hQ db f' s g

theorem **intwLoc_nLoc_mul** [IsTopologicalRing R] [FiniteDimensional F E] (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (db : Measure (HermR R σ n)) [db.IsAddRightInvariant] (f : ℂ → HA R σ Q → ℂ) (s : ℂ) (b₀ : HermR R σ n) (g : HA R σ Q) : intwLoc R σ hQ db f s (nLoc R σ hQ b₀ * g) = intwLoc R σ hQ db f s g

theorem **hgtUp_mul_of_isCompact** (hν : Continuous ν) (hK : IsCompact (K : Set (HA R σ Q))) (p : upperParabolic R σ Q) {k : HA R σ Q} (hk : k ∈ K) : hgtUp R σ Q ν K (p * k) = ν (aUp R σ Q p)

theorem **isIwasawaUp_of_isCompact** (hν : Continuous ν) (hK : IsCompact (K : Set (HA R σ Q))) (h : ∀ g : HA R σ Q, ∃ p : upperParabolic R σ Q, ∃ k ∈ K, g = p * k) : IsIwasawaUp R σ Q ν K

theorem **continuous_hgtUp** (h : IsIwasawaUp R σ Q ν K) (hν : Continuous ν) (hK : IsCompact (K : Set (HA R σ Q))) : Continuous (hgtUp R σ Q ν K)

theorem **sphUp_mem** (h : IsIwasawaUp R σ Q ν K) (hν : Continuous ν) (hK : IsCompact (K : Set (HA R σ Q))) (s : ℂ) : sphUp R σ Q ν K s ∈ psUp R σ Q ν K 1 s

theorem **flatUp_mem** (h : IsIwasawaUp R σ Q ν K) (hν : Continuous ν) (hK : IsCompact (K : Set (HA R σ Q))) {μ : (R ⊗[F] E)ˣ →* Circle} {s₀ : ℂ} {Φ : HA R σ Q → ℂ} (hΦ : Φ ∈ psUp R σ Q ν K μ s₀) (s : ℂ) : flatUp R σ Q ν K Φ s₀ s ∈ psUp R σ Q ν K μ s

theorem **mem_KUpAd** {g : GAd σ Q} : g ∈ KUpAd σ Q ↔ (∀ v, proj σ Q v g ∈ KUpFin σ Q v) ∧ ∀ w, projArch σ Q w g ∈ KUpArch σ Q w

theorem **emb_mem_KUpAd** {v : HeightOneSpectrum (𝒪 F)} {k : GLoc σ Q v} (hk : k ∈ KUpFin σ Q v) : emb σ Q v k ∈ KUpAd σ Q

theorem **embArch_mem_KUpAd** {w : InfinitePlace F} {k : GArch σ Q w} (hk : k ∈ KUpArch σ Q w) : embArch σ Q w k ∈ KUpAd σ Q

theorem **proj_nLoc** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (v : HeightOneSpectrum (𝒪 F)) (b : HermR (AF F) σ n) : proj σ Q v (nLoc _ σ hQ b) = nLoc _ σ hQ (hermMap (evFin F v) b)

theorem **projArch_nLoc** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (w : InfinitePlace F) (b : HermR (AF F) σ n) : projArch σ Q w (nLoc _ σ hQ b) = nLoc _ σ hQ (hermMap (evInf F w) b)

theorem **proj_wLoc** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (v : HeightOneSpectrum (𝒪 F)) : proj σ Q v (wLoc _ σ hQ) = wLoc _ σ hQ

theorem **projArch_wLoc** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (w : InfinitePlace F) : projArch σ Q w (wLoc _ σ hQ) = wLoc _ σ hQ

@[simp] theorem **coe_projUp** (v : HeightOneSpectrum (𝒪 F)) (p : upperParabolic (AF F) σ Q) : (projUp v p : GLoc σ Q v) = proj σ Q v p

@[simp] theorem **coe_projArchUp** (w : InfinitePlace F) (p : upperParabolic (AF F) σ Q) : (projArchUp w p : GArch σ Q w) = projArch σ Q w p

theorem **aUp_projUp** (v : HeightOneSpectrum (𝒪 F)) (p : upperParabolic (AF F) σ Q) : aUp _ σ Q (projUp v p) = unitFin F E v (aUp _ σ Q p)

theorem **aUp_projArchUp** (w : InfinitePlace F) (p : upperParabolic (AF F) σ Q) : aUp _ σ Q (projArchUp w p) = unitInf F E w (aUp _ σ Q p)

theorem **exists_mul_fin** (hI : ∀ g : GAd σ Q, ∃ p : upperParabolic (AF F) σ Q, ∃ k ∈ KUpAd σ Q, g = p * k) (v : HeightOneSpectrum (𝒪 F)) (g : GLoc σ Q v) : ∃ p : upperParabolic (Fv F v) σ Q, ∃ k ∈ KUpFin σ Q v, g = p * k

theorem **exists_mul_arch** (hI : ∀ g : GAd σ Q, ∃ p : upperParabolic (AF F) σ Q, ∃ k ∈ KUpAd σ Q, g = p * k) (w : InfinitePlace F) (g : GArch σ Q w) : ∃ p : upperParabolic w.Completion σ Q, ∃ k ∈ KUpArch σ Q w, g = p * k

theorem **mem_siegelParabolic_iff_forall_place** {g : GAd σ Q} : g ∈ upperParabolic (AF F) σ Q ↔ (∀ w, projArch σ Q w g ∈ upperParabolic _ σ Q) ∧ ∀ v, proj σ Q v g ∈ upperParabolic _ σ Q

@[simp] theorem **coe_embUp** (v : HeightOneSpectrum (𝒪 F)) (p : upperParabolic (Fv F v) σ Q) : (embUp v p : GAd σ Q) = emb σ Q v p

@[simp] theorem **coe_embArchUp** (w : InfinitePlace F) (p : upperParabolic w.Completion σ Q) : (embArchUp w p : GAd σ Q) = embArch σ Q w p

theorem **aUp_embUp** (v : HeightOneSpectrum (𝒪 F)) (p : upperParabolic (Fv F v) σ Q) : aUp _ σ Q (embUp v p) = ideleAt F E v (aUp _ σ Q p)

theorem **aUp_embArchUp** (w : InfinitePlace F) (p : upperParabolic w.Completion σ Q) : aUp _ σ Q (embArchUp w p) = ideleAtInf F E w (aUp _ σ Q p)

theorem **isCompact_KUpFin** (v : HeightOneSpectrum (𝒪 F)) : IsCompact (KUpFin σ Q v : Set (GLoc σ Q v))

theorem **isOpen_KUpFin** (v : HeightOneSpectrum (𝒪 F)) : IsOpen (KUpFin σ Q v : Set (GLoc σ Q v))

theorem **isCompact_KUpArch** (hc : IsCMExt F E σ) (w : InfinitePlace F) : IsCompact (KUpArch σ Q w : Set (GArch σ Q w))

theorem **isCompact_KUpAd** (hc : IsCMExt F E σ) : IsCompact (KUpAd σ Q : Set (GAd σ Q))

theorem **eventually_proj_mem_KUpFin** (g : GAd σ Q) : ∀ᶠ v in Filter.cofinite, proj σ Q v g ∈ KUpFin σ Q v

theorem **finite_mulSupport_hgtUp** (g : GAd σ Q) : (Function.mulSupport fun v => hgtUp _ σ Q (absEv F E v) (KUpFin σ Q v) (proj σ Q v g)).Finite

theorem **isIwasawaUp_fin_of** (v : HeightOneSpectrum (𝒪 F)) : IsIwasawaUp _ σ Q (absEv F E v) (KUpFin σ Q v)

theorem **isIwasawaUp_arch_of** (w : InfinitePlace F) : IsIwasawaUp _ σ Q (absEw F E w) (KUpArch σ Q w)

theorem **isIwasawaUp_ad_of** : IsIwasawaUp _ σ Q (absE F E) (KUpAd σ Q)

theorem **hgtUp_ad_eq_of** (g : GAd σ Q) : hgtUp _ σ Q (absE F E) (KUpAd σ Q) g = (∏ w, hgtUp _ σ Q (absEw F E w) (KUpArch σ Q w) (projArch σ Q w g)) * ∏ᶠ v, hgtUp _ σ Q (absEv F E v) (KUpFin σ Q v) (proj σ Q v g)

theorem **Jst_eq** : StdUnn.Jst D n = StdUnn.blk4 0 1 (-1) 0

theorem **upperParabolic_adele** : upperParabolic (AF D.host.F) D.sp.cm.sig (StdUnn.Jst D n) = StdUnn.P D n

theorem **KUpAd_adele** : KUpAd D.sp.cm.sig (StdUnn.Jst D n) = StdUnn.K D n

theorem **matR_adele** (g : StdUnn.G D n) : matR (AF D.host.F) D.sp.cm.sig g = StdUnn.mat g

theorem **exists_mul_adele** (g : StdUnn.G D n) : ∃ p : upperParabolic (AF D.host.F) D.sp.cm.sig (StdUnn.Jst D n), ∃ k ∈ KUpAd D.sp.cm.sig (StdUnn.Jst D n), g = p * k

theorem **isIwasawaUp_fin** (v : HeightOneSpectrum (𝒪 D.host.F)) : IsIwasawaUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEv D.host.F D.host.E v) (KUpFin _ _ v)

theorem **isIwasawaUp_arch** (w : InfinitePlace D.host.F) : IsIwasawaUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEw D.host.F D.host.E w) (KUpArch _ _ w)

theorem **isIwasawaUp_ad** : IsIwasawaUp _ D.sp.cm.sig (StdUnn.Jst D n) (absA D) (StdUnn.K D n)

theorem **sphUp_mem_fin** (v : HeightOneSpectrum (𝒪 D.host.F)) (s : ℂ) : sphUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEv D.host.F D.host.E v) (KUpFin _ _ v) s ∈ psUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEv D.host.F D.host.E v) (KUpFin _ _ v) 1 s

theorem **sphUp_mem_arch** (w : InfinitePlace D.host.F) (s : ℂ) : sphUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEw D.host.F D.host.E w) (KUpArch _ _ w) s ∈ psUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEw D.host.F D.host.E w) (KUpArch _ _ w) 1 s

theorem **isHeight_iff** {H : StdUnn.G D n → ℝ≥0} : StdUnn.IsHeight H ↔ H = hgtUp _ D.sp.cm.sig (StdUnn.Jst D n) (absA D) (StdUnn.K D n)

theorem **isHeight_eq_prod** {H : StdUnn.G D n → ℝ≥0} (h : StdUnn.IsHeight H) (g : StdUnn.G D n) : H g = (∏ w, hgtUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEw D.host.F D.host.E w) (KUpArch _ _ w) (projArch _ _ w g)) * ∏' v, hgtUp _ D.sp.cm.sig (StdUnn.Jst D n) (absEv D.host.F D.host.E v) (KUpFin _ _ v) (proj _ _ v g)

theorem **hermN_eq** : ResAniso.hermN D n = HermR (AF D.host.F) D.sp.cm.sig n

theorem **continuous_hermAddEquiv** : Continuous (hermAddEquiv D n)

theorem **continuous_hermAddEquiv_symm** : Continuous (hermAddEquiv D n).symm

@[simp] theorem **coe_hermEquiv** (b : ResAniso.hermN D n) : (hermEquiv D n b).1 = b.1

@[simp] theorem **coe_hermEquiv_symm** (b : HermR (AF D.host.F) D.sp.cm.sig n) : ((hermEquiv D n).symm b).1 = b.1

theorem **hermEquiv_eq** (b : ResAniso.hermN D n) : hermEquiv D n b = hermAddEquiv D n b

theorem **isAddHaarMeasure_map_hermEquiv** (db : Measure (ResAniso.hermN D n)) [db.IsAddHaarMeasure] : (db.map (hermEquiv D n)).IsAddHaarMeasure

theorem **nElt_eq** (b : ResAniso.hermN D n) : ResAniso.nElt b = nLoc (AF D.host.F) D.sp.cm.sig (Jst_eq D n) (hermEquiv D n b)

theorem **wElt_eq** : ResAniso.wElt D n = wLoc (AF D.host.F) D.sp.cm.sig (Jst_eq D n)

theorem **intw_eq** (db : Measure (ResAniso.hermN D n)) (f : ℂ → StdUnn.G D n → ℂ) (s : ℂ) (g : StdUnn.G D n) : ResAniso.intw db f s g = intwLoc (AF D.host.F) D.sp.cm.sig (Jst_eq D n) (db.map (hermEquiv D n)) f s g

theorem **intwConv_map_iff** (db : Measure (ResAniso.hermN D n)) (f : ℂ → StdUnn.G D n → ℂ) (s : ℂ) (g : StdUnn.G D n) : IntwConv (AF D.host.F) D.sp.cm.sig (Jst_eq D n) (db.map (hermEquiv D n)) f s g ↔ Integrable (fun b => f s (ResAniso.wElt D n * ResAniso.nElt b * g)) db

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsBplusNeZero

theorem **isMeasLevW_unique** {V : HSp D} {K : Subgroup (GL (Fin V.rank) (Lev.EAf D))} {good} {μ μ' : Measure (GA D V)} (hg : ∀ ρ m, good ρ m → m.IsHaarMeasure) (h : Lev.IsMeasLevW V K good μ) (h' : Lev.IsMeasLevW V K good μ') : μ = μ'

theorem **isAt_unique** {M M' : Vol G} {X : Node} {K} (h : M.IsAt X K) (h' : M'.IsAt X K) : M.ν X = M'.ν X

theorem **liftA_indep** {M M' : Vol G} (a : Arrow T) {K} (h : M.IsAt a.src K) (h' : M'.IsAt a.src K) (f : Sch G a) (F : Fn G a.src) (hinv : ∀ x, ∀ γ ∈ GQ (G.V a.src), ∀ g, W.kerA a f (γ * g) x * F (γ * g) = W.kerA a f g x * F g) : liftA W M a f F = liftA W M' a f F

theorem **embTop_hom** : ∃ s : GA D (G.V (.top 0)) →* GA D (G.V (.grid D.R.m 0)), Continuous s ∧ embTop G = s ∧ ∀ γ ∈ GQ (G.V (.top 0)), s γ ∈ GQ (G.V (.grid D.R.m 0))

private theorem **besChar_mul_eq_of_embRel** {A B : HSp D} {ι : A.V →ₗ[D.host.E] B.V} (d : BesDat A B ι) (ψ : AddChar (AF D.host.F) ℂ) (g n : GA D B) (h : GA D A) (hrel : EmbRel B A ι g h) : besChar ψ d (n * g) = besChar ψ d n

theorem **exists_Bplus_ne_zero** {M : Vol G} (hM : M.IsRow) : ∃ φ F, BplusConv W M φ F ∧ Bplus W M φ F ≠ 0

theorem **tensT_single** {X Y : Node} {p : Path T X Y} (P : PolT G p) (d : PureT G p) (c c' : ℂ) : tensT W (Finsupp.single P c) (Finsupp.single d c') = Finsupp.single (Path.Fam.zip (fun a (f : Sch G a) Q => W.fock a.pair f Q) d P) (c * c')

theorem **levJ_col** (t : T) (j : ℕ) : levJ G t (.grid 0 (j + 1)) = levCol L t j

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsChartOfIsLocalRing

theorem **ev_congr** {z z' : Sp S ⟶ Z} (e : z = z') (U : Z.Opens) (h : ⊤ ≤ z ⁻¹ᵁ U) (h' : ⊤ ≤ z' ⁻¹ᵁ U) : ev z U h = ev z' U h'

theorem **ev_appLE** (z : Sp S ⟶ Z') (π : Z' ⟶ Z) {U : Z.Opens} {V : Z'.Opens} (e : V ≤ π ⁻¹ᵁ U) (h : ⊤ ≤ z ⁻¹ᵁ V) (h' : ⊤ ≤ (z ≫ π) ⁻¹ᵁ U) (a : Γ(Z, U)) : ev z V h ((π.appLE U V e).hom a) = ev (z ≫ π) U h' a

theorem **ev_res** (z : Sp S ⟶ Z) {U V : Z.Opens} (e : V ≤ U) (h : ⊤ ≤ z ⁻¹ᵁ V) (h' : ⊤ ≤ z ⁻¹ᵁ U) (a : Γ(Z, U)) : ev z V h (resG Z e a) = ev z U h' a

theorem **ev_spMap** (φ : S →+* S') (z : Sp S ⟶ Z) (U : Z.Opens) (h : ⊤ ≤ z ⁻¹ᵁ U) (h' : ⊤ ≤ (spMap φ ≫ z) ⁻¹ᵁ U) : ev (spMap φ ≫ z) U h' = φ.comp (ev z U h)

theorem **le_preimage_comp** {T : Scheme.{0}} (g : T ⟶ Z') {z : Z' ⟶ Z} {U : Z.Opens} (h : ⊤ ≤ z ⁻¹ᵁ U) : ⊤ ≤ (g ≫ z) ⁻¹ᵁ U

theorem **le_preimage_map** {T : Scheme.{0}} {z : T ⟶ Z'} (π : Z' ⟶ Z) {U : Z.Opens} {V : Z'.Opens} (e : V ≤ π ⁻¹ᵁ U) (h : ⊤ ≤ z ⁻¹ᵁ V) : ⊤ ≤ (z ≫ π) ⁻¹ᵁ U

theorem **le_preimage_mono** {T : Scheme.{0}} {z : T ⟶ Z} {U V : Z.Opens} (e : V ≤ U) (h : ⊤ ≤ z ⁻¹ᵁ V) : ⊤ ≤ z ⁻¹ᵁ U

theorem **le_preimage_inf** {T : Scheme.{0}} {z : T ⟶ Z} {U V : Z.Opens} (h : ⊤ ≤ z ⁻¹ᵁ U) (h' : ⊤ ≤ z ⁻¹ᵁ V) : ⊤ ≤ z ⁻¹ᵁ (U ⊓ V)

theorem **spMap_ev** (z : Sp S ⟶ Z) {U : Z.Opens} (hU : IsAffineOpen U) (h : ⊤ ≤ z ⁻¹ᵁ U) : spMap (ev z U h) ≫ hU.fromSpec = z

theorem **tau_le** {U V : Z.Opens} (hU : IsAffineOpen U) (e : U ≤ V) : ⊤ ≤ tau hU ⁻¹ᵁ V

theorem **ev_tau_self** {U : Z.Opens} (hU : IsAffineOpen U) (h : ⊤ ≤ tau hU ⁻¹ᵁ U) : ev (tau hU) U h = RingHom.id _

theorem **ev_tau** {U V : Z.Opens} (hU : IsAffineOpen U) (e : U ≤ V) (h : ⊤ ≤ tau hU ⁻¹ᵁ V) (a : Γ(Z, V)) : ev (tau hU) V h a = resG Z e a

theorem **ev_spMap_tau** {U : Z.Opens} (hU : IsAffineOpen U) (f : Γ(Z, U) →+* S) (h : ⊤ ≤ (spMap f ≫ tau hU) ⁻¹ᵁ U) : ev (spMap f ≫ tau hU) U h = f

theorem **ext_ev_loc** {V : Z.Opens} {a b : Γ(Z, V)} (h : ∀ p ∈ V, ∃ N : Z.Opens, p ∈ N ∧ ∀ (S : Type) [CommRing S] (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ V), ⊤ ≤ z ⁻¹ᵁ N → ev z V hz a = ev z V hz b) : a = b

theorem **ext_ev** {V : Z.Opens} {a b : Γ(Z, V)} (h : ∀ (S : Type) [CommRing S] (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ V), ev z V hz a = ev z V hz b) : a = b

theorem **isLB_of_ev** (L : LB Z) (hc : ⨆ i, L.U i = ⊤) (h : ∀ (i j k : L.ι) (S : Type) [CommRing S] (z : Sp S ⟶ Z) (hij : ⊤ ≤ z ⁻¹ᵁ (L.U i ⊓ L.U j)) (hjk : ⊤ ≤ z ⁻¹ᵁ (L.U j ⊓ L.U k)) (hik : ⊤ ≤ z ⁻¹ᵁ (L.U i ⊓ L.U k)), ev z _ hij (L.g i j).val * ev z _ hjk (L.g j k).val = ev z _ hik (L.g i k).val) : L.IsLB

theorem **exists_chart_of_isLocalRing** [IsLocalRing S] {J : Type} {U : J → Z.Opens} (hc : ⨆ i, U i = ⊤) (z : Sp S ⟶ Z) : ∃ i, ⊤ ≤ z ⁻¹ᵁ U i

theorem **smT_epsHom** (φ : S →+* S') {x : Sp S ⟶ X0} {x' : Sp S' ⟶ X0} (h : spMap φ ≫ x = x') (a : S) (v : overAt x (DualNumber S) ⟶ G) : smT x' (φ a) (epsHom φ h ≫ v) = epsHom φ h ≫ smT x a v

theorem **tanPull_left** (φ : S →+* S') {x : Sp S ⟶ X0} {x' : Sp S' ⟶ X0} (h : spMap φ ≫ x = x') (v : TanM x G) : (tv (tanPull φ h v)).left = spMap (dualMap φ) ≫ (tv v).left

theorem **tan_ext_left** {x : Sp S ⟶ X0} {u v : TanM x G} (h : (tv u).left = (tv v).left) : u = v

theorem **tanPull_actR** {O : Type} [CommRing O] (A : OAct O G) (φ : S →+* S') {x : Sp S ⟶ X0} {x' : Sp S' ⟶ X0} (h : spMap φ ≫ x = x') (a : O) (v : TanM x G) : tanPull φ h (actR A a v) = actR A a (tanPull φ h v)

theorem **tv_sum'** {x : Sp S ⟶ X0} {κ : Type} (s : Finset κ) (f : κ → TanM x G) : tv (∑ i ∈ s, f i) = ∏ i ∈ s, tv (f i)

theorem **dualMap_comp** {S'' : Type} [CommRing S''] (φ : S →+* S') (ψ : S' →+* S'') : dualMap (ψ.comp φ) = (dualMap ψ).comp (dualMap φ)

theorem **exists_lieBasis** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) : ∃ b : Module.Basis (Fin M.n) S (TanM x Y.t.G), (∀ j, (tv (b j)).left = spMap (dualMap (ev x M.U h)) ≫ M.v₀ j) ∧ ∀ a j, actR Y.t.act a (b j) = ∑ i, ev x M.U h (M.ρ a i j) • b i

theorem **lieBasis_left** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (j : Fin M.n) : (tv (lieBasis M x h j)).left = spMap (dualMap (ev x M.U h)) ≫ M.v₀ j

theorem **actR_lieBasis** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (a : 𝒪 E) (j : Fin M.n) : actR Y.t.act a (lieBasis M x h j) = ∑ i, ev x M.U h (M.ρ a i j) • lieBasis M x h i

theorem **tanPull_lieBasis** (M : MChart Y) (φ : S →+* S') {x : Sp S ⟶ Y.X} {x' : Sp S' ⟶ Y.X} (e : spMap φ ≫ x = x') (h : ⊤ ≤ x ⁻¹ᵁ M.U) (h' : ⊤ ≤ x' ⁻¹ᵁ M.U) (j : Fin M.n) : tanPull φ e (lieBasis M x h j) = lieBasis M x' h' j

theorem **vec_apply** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (cv : Fin M.n → S) : vec M x h cv = ∑ j, cv j • lieBasis M x h j

theorem **tanPull_vec** (M : MChart Y) (φ : S →+* S') {x : Sp S ⟶ Y.X} {x' : Sp S' ⟶ Y.X} (e : spMap φ ≫ x = x') (h : ⊤ ≤ x ⁻¹ᵁ M.U) (h' : ⊤ ≤ x' ⁻¹ᵁ M.U) (cv : Fin M.n → S) : tanPull φ e (vec M x h cv) = vec M x' h' (fun j => φ (cv j))

theorem **actR_vec** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (a : 𝒪 E) (cv : Fin M.n → S) : actR Y.t.act a (vec M x h cv) = vec M x h (((M.ρ a).map (ev x M.U h)).mulVec cv)

theorem **isAffineOpen_inf** (M M' : MChart Y) : IsAffineOpen (M.U ⊓ M'.U)

theorem **chg_map** (M M' : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (h' : ⊤ ≤ x ⁻¹ᵁ M'.U) (hh : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) : (chg M M').map (ev x _ hh) = (lieBasis M x h).toMatrix (lieBasis M' x h')

theorem **vec_chg** (M M' : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (h' : ⊤ ≤ x ⁻¹ᵁ M'.U) (hh : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) (cv : Fin M'.n → S) : vec M x h (((chg M M').map (ev x _ hh)).mulVec cv) = vec M' x h' cv

theorem **rdet_self** {n : ℕ} (A : Matrix (Fin n) (Fin n) R) : rdet A = A.det

theorem **map_rdet** (φ : R →+* R') {n n' : ℕ} (A : Matrix (Fin n) (Fin n') R) : φ (rdet A) = rdet (A.map φ)

theorem **rdet_toMatrix_mul** {n₁ n₂ n₃ : ℕ} (b₁ : Module.Basis (Fin n₁) R N) (b₂ : Module.Basis (Fin n₂) R N) (b₃ : Module.Basis (Fin n₃) R N) : rdet (b₁.toMatrix b₂) * rdet (b₂.toMatrix b₃) = rdet (b₁.toMatrix b₃)

theorem **isUnit_rdet_toMatrix** {n₁ n₂ : ℕ} (b₁ : Module.Basis (Fin n₁) R N) (b₂ : Module.Basis (Fin n₂) R N) : IsUnit (rdet (b₁.toMatrix b₂))

theorem **detChg_val** (M M' : MChart Y) : (detChg M M').val = rdet (chg M M')

theorem **ev_detChg** (M M' : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (h' : ⊤ ≤ x ⁻¹ᵁ M'.U) (hh : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) : ev x _ hh (detChg M M').val = rdet ((lieBasis M x h).toMatrix (lieBasis M' x h'))

theorem **ev_detChg_mul** (M M' M'' : MChart Y) (x : Sp S ⟶ Y.X) (h₁ : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) (h₂ : ⊤ ≤ x ⁻¹ᵁ (M'.U ⊓ M''.U)) (h₃ : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M''.U)) : ev x _ h₁ (detChg M M').val * ev x _ h₂ (detChg M' M'').val = ev x _ h₃ (detChg M M'').val

theorem **resG_resG** {Z : Scheme.{0}} {U V W : Z.Opens} (h : V ≤ U) (h' : W ≤ V) (a : Γ(Z, U)) : resG Z h' (resG Z h a) = resG Z (h'.trans h) a

theorem **exists_gChart** [IsNoetherianRing O] (hP : PushInvertible Y.t.G Y.t.PP.L) (x : Y.X) : ∃ M : GChart Y, x ∈ M.U

theorem **ev_thRat** (M M' : GChart Y) (g : Sp S ⟶ Y.t.G.left) (i : Y.t.PP.L.ι) (h : ⊤ ≤ g ⁻¹ᵁ (Y.t.G.hom ⁻¹ᵁ M.U ⊓ Y.t.PP.L.U i)) (h' : ⊤ ≤ g ⁻¹ᵁ (Y.t.G.hom ⁻¹ᵁ M'.U ⊓ Y.t.PP.L.U i)) (hh : ⊤ ≤ (g ≫ Y.t.G.hom) ⁻¹ᵁ (M.U ⊓ M'.U)) : ev g _ h' (M'.θ i) = ev (g ≫ Y.t.G.hom) _ hh (thRat M M') * ev g _ h (M.θ i)

theorem **thRat_unique** (M M' : GChart Y) {W : Y.X.Opens} (hW : W ≤ M.U) (hW' : W ≤ M'.U) {a b : Γ(Y.X, W)} (H : ∀ e ∈ ({a, b} : Set Γ(Y.X, W)), ∀ (S : Type) [CommRing S] (g : Sp S ⟶ Y.t.G.left) (i : Y.t.PP.L.ι) (h : ⊤ ≤ g ⁻¹ᵁ (Y.t.G.hom ⁻¹ᵁ M.U ⊓ Y.t.PP.L.U i)) (h' : ⊤ ≤ g ⁻¹ᵁ (Y.t.G.hom ⁻¹ᵁ M'.U ⊓ Y.t.PP.L.U i)) (hh : ⊤ ≤ (g ≫ Y.t.G.hom) ⁻¹ᵁ W), ev g _ h' (M'.θ i) = ev (g ≫ Y.t.G.hom) _ hh e * ev g _ h (M.θ i)) : a = b

theorem **thRat_mul** (M M' M'' : GChart Y) {W : Y.X.Opens} (h₁ : W ≤ M.U ⊓ M'.U) (h₂ : W ≤ M'.U ⊓ M''.U) (h₃ : W ≤ M.U ⊓ M''.U) : resG _ h₁ (thRat M M') * resG _ h₂ (thRat M' M'') = resG _ h₃ (thRat M M'')

theorem **thRat_self** (M : GChart Y) : thRat M M = 1

theorem **isUnit_thRat** (M M' : GChart Y) : IsUnit (thRat M M')

theorem **thUnit_val** (M M' : GChart Y) : (thUnit M M').val = thRat M M'

theorem **ev_thRat_mul** (M M' M'' : GChart Y) (x : Sp S ⟶ Y.X) (h₁ : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M'.U)) (h₂ : ⊤ ≤ x ⁻¹ᵁ (M'.U ⊓ M''.U)) (h₃ : ⊤ ≤ x ⁻¹ᵁ (M.U ⊓ M''.U)) : ev x _ h₁ (thRat M M') * ev x _ h₂ (thRat M' M'') = ev x _ h₃ (thRat M M'')

theorem **ev_unit_inv** {Z : Scheme.{0}} (z : Sp S ⟶ Z) {U : Z.Opens} (h : ⊤ ≤ z ⁻¹ᵁ U) (u : Γ(Z, U)ˣ) : ev z U h (u⁻¹).val = ((Units.map (ev z U h).toMonoidHom u)⁻¹).val

theorem **isLB_detLie** [IsNoetherianRing O] (hP : PushInvertible Y.t.G Y.t.PP.L) : (detLie Y).IsLB

theorem **isLB_detOm** [IsNoetherianRing O] (hP : PushInvertible Y.t.G Y.t.PP.L) : (detOm Y).IsLB

theorem **isLB_pushPP** [IsNoetherianRing O] (hP : PushInvertible Y.t.G Y.t.PP.L) : (pushPP Y).IsLB

theorem **isLB_de** [IsNoetherianRing O] (hP : PushInvertible Y.t.G Y.t.PP.L) : (de Y).IsLB

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsComplSpace

theorem **exists_frame_hyp** (D : D7) (t : ℕ) : ∃ b : Module.Basis (Fin (t + t)) D.host.E (GlobalTheory.OrthSumSp.hypSp D t).V, ∀ i j, (GlobalTheory.OrthSumSp.hypSp D t).hT (b i) (b j) = -blk (0 : Matrix (Fin t) (Fin t) D.host.E) 1 1 0 i j

theorem **exists_compl_space** (D : D7) (V : HSp D) (t : ℕ) : ∃ b : Module.Basis (Fin (t + t)) D.host.E (GlobalTheory.OrthSumSp.hypSp D t).V, -(hsum V { GlobalTheory.OrthSumSp.hypSp D t with frame

theorem **complMet_of** (D : D7) (V : HSp D) (han : IsAniso V) (t : ℕ) : ComplMet D.sp.cm.sig V.rank t False

theorem **exists_complMet** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E), ∀ m t, ComplMet σ m t False

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsCoordAtPoints

theorem **le_chart** (j : ℤ) : ⊤ ≤ (x.1 ≫ (G.ideal s).subschemeι) ⁻¹ᵁ (tw G.L₂ G.L j).U (twIdx j (G.second.chartAt (ptOf x.pt.1)) (G.chartAt (ptOf x.pt.1)))

theorem **c₁_pos** (τ : K →+* ℂ) : 0 < x.c₁ τ

theorem **c₂_pos** (τ : K →+* ℂ) : 0 < x.c₂ τ

theorem **ht_eq_pointHeight** : x.ht = pointHeight G.w G.A (G.coords x.pt.1)

theorem **ht_eq_ptHeight** : x.ht = G.ptHeight x.pt

theorem **c₂_conj** (τ : K →+* ℂ) : x.c₂ ((starRingEnd ℂ).comp τ) = x.c₂ τ

theorem **exp_neg_weight** (j : ℕ) (τ : K →+* ℂ) : Real.exp (-(Geo.weightOn s j τ).φ (fun _ => twIdx j (G.second.chartAt (ptOf x.pt.1)) (G.chartAt (ptOf x.pt.1))) (x.along τ)) = ((x.c₂ τ * x.c₁ τ ^ j) ^ 2)⁻¹

theorem **nsq_eq** (j : ℕ) (τ : K →+* ℂ) {ι : Type} [Fintype ι] (c : ι → ℂ) (g : ι → G.SecOn s j) : (Geo.weightOn s j τ).nsq (∑ i, c i • cval τ (g i)) (x.along τ) = ‖∑ i, c i * τ (x.val j (g i))‖ ^ 2 / (x.c₂ τ * x.c₁ τ ^ j) ^ 2

theorem **nrm_eq** (j : ℕ) (τ : K →+* ℂ) (g : G.SecOn s j) : x.nrm j τ g = ‖τ (x.val j g)‖ / (x.c₂ τ * x.c₁ τ ^ j)

theorem **val_mulSecOn** (j : ℕ) (u : Fin G.N → K) (g : G.SecOn s j) : x.val (j + 1) (mulSecOn G.isLB₂ G.isLB (G.comb u) (Nat.cast_succ j).symm g) = (∑ k, u k * G.coords x.pt.1 k) * x.val j g

theorem **along_injective** (hs : G.IsGood s) {Ω : Type} [Field Ω] (φ : K →+* Ω) : Function.Injective fun x : RatPt G s => x.along φ

theorem **RatPt.along_bijective** (hs : G.IsGood s) (h : Function.Surjective fun x : RatPt G s => x.along (algebraMap K (AlgebraicClosure K))) {Ω : Type} [Field Ω] [IsAlgClosed Ω] (φ : K →+* Ω) : Function.Bijective fun x : RatPt G s => x.along φ

theorem **exists_coord_atPoints** (hs : G.IsGood s) (h : Function.Surjective fun x : RatPt G s => x.along (algebraMap K (AlgebraicClosure K))) {δ : ℕ} (x : Fin δ ≃ RatPt G s) : ∃ Co : Coord G hs δ, Co.AtPoints x

theorem **Coord.AtPoints.norm_ev** {hs : G.IsGood s} {δ : ℕ} {Co : Coord G hs δ} {x : Fin δ ≃ RatPt G s} (h : Co.AtPoints x) (p : Fin δ) (τ : K →+* ℂ) (g : G.SecOn s 0) : ‖τ (Co.ev 0 g p)‖ / Co.c₂ τ p = (x p).nrm 0 τ g

theorem **Coord.AtPoints.vecHt_eq** {hs : G.IsGood s} {δ : ℕ} {Co : Coord G hs δ} {x : Fin δ ≃ RatPt G s} (h : Co.AtPoints x) (p : Fin δ) : vecHt (fun k => Co.gen k p) (fun τ => Co.c₁ τ p) = (x p).ht

theorem **nrm_normOn_sq_eq** (hs : G.IsGood s) (h : Function.Surjective fun x : RatPt G s => x.along (algebraMap K (AlgebraicClosure K))) (j : ℕ) (τ : K →+* ℂ) (g : G.SecOn s j) : (hs.normOn j).nrm τ g ^ 2 = ∑ᶠ x : RatPt G s, x.nrm j τ g ^ 2

theorem **RatPt.ptHeight_geomPt** (x : RatPt (G.baseChange E) s) : haveI : Algebra.IsAlgebraic F (AlgebraicClosure E)

theorem **Split.along_surjective** (h : Split G.toTuple E s) : Function.Surjective fun x : RatPt (G.baseChange E) s => x.along (algebraMap E (AlgebraicClosure E))

theorem **Split.geomPts_eq** (h : Split G.toTuple E s) : G.geomPts E (G.linSection E s) = Set.range (RatPt.geomPt G E s)

theorem **RatPt.geomPt_injective** (hs : (G.baseChange E).IsGood s) : Function.Injective (RatPt.geomPt G E s)

theorem **Split.exists_coord** (h : Split G.toTuple E s) (hs : (G.baseChange E).IsGood s) : ∃ (x : Fin (G.geomPts E (G.linSection E s)).ncard ≃ RatPt (G.baseChange E) s) (Co : Coord (G.baseChange E) hs (G.geomPts E (G.linSection E s)).ncard), Co.AtPoints x ∧ G.cycleHeight E (G.linSection E s) = ∑ p, (x p).ht

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsCoreGoodMatch

theorem **exists_core_good_match** (hO : OFull D) (hZ : OZeta D) (hV : VolTieG D) (h : 5 ≤ D.host.splitBound) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) : ∃ T : CoreJ D, LawsJ T ∧ ∀ j, InR D j → ∀ e, ∃ (Y : Deep T.toCore j) (o : ℝ), Y.Good e o ∧ Y.Match ∧ Y.d.l₂.M = D.host.g 0 * D.host.g 1 * 15 ^ e ∧ ¬ Y.Pull 1 0

theorem **chi_eq_one** (h : Y.Good e o) (σ : Y.d.c₂.F →+* ℂ) (hσ : GoodEmb Y.d.c₂ σ) (k) (la) (γ) (hγ : γ ∈ ((Y.U σ).X k).Γ) : ((Y.U σ).X k).χ la γ = 1

theorem **exp_unique** (h : Y.Good e o) (h' : Y.Good e' o') : e = e'

theorem **size_nonneg** (h : Y.Good e o) : 0 ≤ o

theorem **rank_eq** (h : Y.Good e o) : D.R.r = (T.Q.Dw j).p + (T.Q.Dw j).q

theorem **pull_add** {f f' : T.R j} {g g' : Rul Y.d.Xo₂ Y.d.c₂} (h : Y.Pull f g) (h' : Y.Pull f' g') : Y.Pull (f + f') (g + g')

theorem **pull_mul** {f f' : T.R j} {g g' : Rul Y.d.Xo₂ Y.d.c₂} (h : Y.Pull f g) (h' : Y.Pull f' g') : Y.Pull (f * f') (g * g')

theorem **latOver_mono** (G : WTZ D.host.E) {B B' : Set (T.R j)} (h : B ⊆ B') : Y.latOver G B ≤ Y.latOver G B'

theorem **latOver_empty** (G : WTZ D.host.E) : Y.latOver G ∅ = ⊥

theorem **nrmR_smul** (σ : D.what₂.LL →+* ℂ) (a : D.what₂.LL) (f : Fin (R.nW i) → D.what₂.LL) : nrmR R i σ (a • f) = ‖σ a‖ * nrmR R i σ f

theorem **latType_eq** : latType R F i = latAll R i ⊓ (spType R F i).restrictScalars (𝒪 D.what₂.LL)

theorem **latGood_inf_latBad** (h : eGood R i ∘ₗ eGood R i = eGood R i) : latGood R F i ⊓ latBad R F i = ⊥

theorem **exZ_shiftG** (s : ℤ) : exZ Dw (shiftG Dw a (2 * m) s) = exOf Dw (laOf Dw a (2 * m) + s • hodgeWt Dw) m

theorem **uMinus_apply** (n kc : ℕ) (c0 : ℤ) (τ : Psi H) (k : Fin (Dw.p + Dw.q)) : uMinus Dw a n kc c0 τ k = if (k : ℕ) < Dw.p then uWt Dw a n kc τ k else uWt Dw a n kc τ k + c0

theorem **anchor_big** (kc : ℕ) (h : kc < Dw.p) (τ : Psi H) : anchor Dw a (2 * m) (kc + 1) τ = a (τ.1, kc) + m

theorem **abs_laOf** (τ : Psi H) (k : Fin (Dw.p + Dw.q)) : |laOf Dw a (2 * m) τ k| = (if (k : ℕ) < Dw.p then a (τ.1, k) else a (conjEmb τ.1, Dw.q - 1 - ((k : ℕ) - Dw.p)) : ℕ) + m

theorem **expL_le** {L : ℝ} (h : 0 ≤ L) : expL L * Real.log 3 ≤ 3 * L

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsDatNeZero

theorem **sm_whole** {D : Type} [CommRing D] [Algebra R D] (x : Pt X D) : Sm (whole X) x

theorem **Pt.w** {D : Type} [CommRing D] [Algebra R D] (x : Pt X D) : x.left ≫ X.hom ≫ 𝟙 _ = spMap (algebraMap R D)

theorem **cotCentral** : IsCentralScalar R (Cot R D)

theorem **cotTower** : IsScalarTower R Rᵐᵒᵖ (Cot R D)

theorem **fv_of_form** {X : Over (Sp R)} (f : Form (X.hom ≫ 𝟙 _)) (D : Type) [CommRing D] [Algebra R D] (D' : Type) [CommRing D'] [Algebra R D'] (φ : D →ₐ[R] D') (x : Pt X D) : fv f.1 D D' φ x = 0

theorem **Dat.coc_ofForm** (t : G.TorsionTower A) (f : Form (A.hom ≫ 𝟙 _)) (hf : Form.cl _ f ∈ primDR A) (n : ℕ) : (Dat.ofForm t f hf).coc n = 0

theorem **Dat.om_ofForm** (t : G.TorsionTower A) (f : Form (A.hom ≫ 𝟙 _)) (hf : Form.cl _ f ∈ primDR A) (n : ℕ) : (Dat.ofForm t f hf).om n = cotVal f.1 (t.y n)

theorem **cotVal_form_res** (t : G.TorsionTower A) (f : Form (A.hom ≫ 𝟙 _)) (n : ℕ) : cotMap (G.res n) (cotVal f.1 (t.y (n + 1))) = cotVal f.1 (t.y n)

theorem **isCan_ofForm** (t : G.TorsionTower A) (f : Form (A.hom ≫ 𝟙 _)) (hf : Form.cl _ f ∈ primDR A) : IsCan t ⟨_, hf⟩ (G.rigOfOmega (fun n ↦ cotVal f.1 (t.y n)), cotVal_form_res t f))

theorem **prim_dlog_ne_zero** [Nontrivial R] : (⟨_, MultGrp.cl_dlog_mem (R

theorem **Dat.dlog_f_ne_zero** (p : ℕ) [Fact p.Prime] [Nontrivial R] : (Dat.dlog p R).f ≠ 0

theorem **exists_dat_ne_zero** (R : Type) [CommRing R] [Nontrivial R] (p : ℕ) [Fact p.Prime] : ∃ c, Nonempty (Dat (PDivGroup.muTower p R) c) ∧ c ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsDataOfPositivity

theorem **exists_intGens** : ∃ (n : ℕ) (w : Fin n → Fin G.N → F), Submodule.span ℤ (Set.range w) = G.lattice.restrictScalars ℤ

theorem **exists_regen** : ∃ (n : ℕ) (w : Fin n → Fin G.N → F) (hw : _) (Λ : ℝ), (regen G w hw Λ).ShortGenerators ∧ (regen G w hw Λ).lattice = G.lattice

theorem **positivity_regen** (h : G.Positivity) {n : ℕ} (w : Fin n → Fin G.N → F) (hw) (Λ : ℝ) : (regen G w hw Λ).Positivity

theorem **mulW_mem_global** (j : ℕ) (u : Fin G.N → F) {f : G.Sec j} (hf : f ∈ G.global j) : G.mulW j u f ∈ G.global (j + 1)

theorem **basis_mem_lat** (j : ℕ) (i) : Module.finBasis F (G.global j) i ∈ lat G j

theorem **isFull_lat** (j : ℕ) : IsFull (lat G j)

theorem **mulStable** : (data G T hT β J hβ).MulStable

theorem **shortFamily** (j : ℕ) : (data G T hT β J hβ).ShortFamily j 1

theorem **nrm_ofBasis_eq** {V n : Type} [AddCommGroup V] [Module F V] [Fintype n] [DecidableEq n] (b : Module.Basis n F V) (σ : F →+* ℂ) (x : V) : (ofBasis b).nrm σ x = Real.sqrt (∑ i, ‖σ (b.repr x i)‖ ^ 2)

theorem **exists_bound** {V V' : Type} [AddCommGroup V] [Module F V] [AddCommGroup V'] [Module F V'] {N m m' : ℕ} (b : Module.Basis (Fin m) F V) (b' : Module.Basis (Fin m') F V') (μ : (Fin N → F) →ₗ[F] V →ₗ[F] V') (σ : F →+* ℂ) {A : Matrix (Fin N) (Fin N) ℂ} (hA : A.det ≠ 0) : ∃ c : ℝ, 0 ≤ c ∧ ∀ w f, (ofBasis b').nrm σ (μ w f) ≤ c * frameNorm A (fun k => σ (w k)) * (ofBasis b).nrm σ f

theorem **exists_mulBound** : ∃ β₀ : ℝ, ∀ β ≥ β₀, ∀ (hβ : 1 ≤ β) (J : ℝ), (data G T hT β J hβ).MulBound fun _ => β

theorem **exists_data_of_positivity** (h : G.Positivity) : ∃ X : Data F, X.D = G.D ∧ X.T = T ∧ X.d = Module.finrank F (G.global T) ∧ X.del = G.del ∧ X.Positivity ∧ X.MulStable ∧ X.MulBound (fun _ => X.beta) ∧ X.ShortGenerators ∧ X.ShortFamily X.D X.LamD ∧ X.ShortFamily X.T X.LamT

theorem **exists_data** {M : ℕ} (hM : 1 ≤ M) {T : ℕ} (hT : M + 1 ≤ T) : ∃ X : Data F, X.D = M ∧ X.T = T ∧ 0 < X.d ∧ X.del = 1 ∧ X.Positivity ∧ X.MulStable ∧ X.MulBound (fun _ => X.beta) ∧ X.ShortGenerators ∧ X.ShortFamily X.D X.LamD ∧ X.ShortFamily X.T X.LamT

theorem **not_shortGenerators_geo** [IsTotallyComplex F] {M : ℕ} (hM : 1 ≤ M) : ¬ (Proj.geo F hM).ShortGenerators

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsDefiniteDatumGoodObj

theorem **OEp_le** (D : Datum F) (p : ℕ) : D.OEp p ≤ ZbarP p

theorem **exists_good** (F : Type) [Field F] [NumberField F] {N : ℕ} (hN : 3 ≤ N) : ∃ p, Good F p N

theorem **conj_mem_type_iff** (H : Host E) (τ : E →+* ℂ) : ((starRingEnd ℂ).comp τ) (H.rt : E) = H.rt0 ↔ ¬ τ (H.rt : E) = H.rt0

theorem **sum_mult_eq** (D : Geo.Tor.Datum H) (a : E) : ∑ τ : E →+* ℂ, (D.mult τ : ℂ) * τ a = ∑ τ ∈ Finset.univ.filter (fun τ : E →+* ℂ => τ (H.rt : E) = H.rt0), ((D.p : ℂ) * τ a + (D.q : ℂ) * starRingEnd ℂ (τ a))

theorem **dim_ofHost** (D : Geo.Tor.Datum H) : (Datum.ofHost D).dim = D.dim

theorem **definite_ofHost_iff** (D : Geo.Tor.Datum H) : (Datum.ofHost D).Definite ↔ D.p = 0 ∨ D.q = 0

theorem **existsUnique_embedding** {k : Type} [Field k] [IsAlgClosed k] (ξ : ℂ →+* k) (σ : 𝒪 E →+* k) : ∃! τ : E →+* ℂ, ∀ a : 𝒪 E, ξ (τ (a : E)) = σ a

theorem **determinantConditionAt_of_determinantCondition** (D : Geo.Tor.Datum H) (p : ℕ) {k : Type} [Field k] [IsAlgClosed k] {G : Over S} [GrpObj G] (act : OAct (𝒪 E) G) (x : Sp k ⟶ S) (ξ : ℂ →+* k) (h : DetCond D act x (ξ H.rt0)) : DetCondP (Datum.ofHost D) p act x (ξ.comp ((Datum.ofHost D).OEp p).subtype)

theorem **exists_obj_ofHost** (D : Geo.Tor.Datum H) (hD : 1 ≤ D.p + D.q) (p N : ℕ) (hN : 0 < N) : ∃ o : Obj (Datum.ofHost D) p N (𝟙 (Sp ℂ) ≫ spMap ((Datum.ofHost D).OEp p).subtype), SmoothOfRelativeDimension (Module.finrank ℚ E / 2 * (Datum.ofHost D).n) o.A.hom ∧ ∃ x : 𝟙_ (Over (Sp ℂ)) ⟶ o.A, x ≠ 1

theorem **exists_datum_good_obj** : ∃ (D : Datum HostModel.E21) (p : ℕ), D.n = 2 ∧ D.dim = 6 ∧ Good HostModel.E21 p 5 ∧ ∃ o : Obj D p 5 (𝟙 (Sp ℂ) ≫ spMap (D.OEp p).subtype), SmoothOfRelativeDimension 12 o.A.hom ∧ ∃ x : 𝟙_ (Over (Sp ℂ)) ⟶ o.A, x ≠ 1

theorem **exists_definite_datum_good_obj** : ∃ (D : Datum CM.CyclotomicTwentyEight.CyclotomicFieldTwentyEight) (p : ℕ), D.n = 1 ∧ D.dim = 0 ∧ D.Definite ∧ Good CM.CyclotomicTwentyEight.CyclotomicFieldTwentyEight p 5 ∧ ∃ o : Obj D p 5 (𝟙 (Sp ℂ) ≫ spMap (D.OEp p).subtype), ∃ x : 𝟙_ (Over (Sp ℂ)) ⟶ o.A, x ≠ 1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsDimEqOne

@[ext] theorem **hom_ext** {A B : AbVar K} {f g : A ⟶ B} (h : f.hom = g.hom) : f = g

@[simp] theorem **id_hom** (A : AbVar K) : Hom.hom (𝟙 A) = 𝟙 A.X

@[simp] theorem **comp_hom** {A B C : AbVar K} (f : A ⟶ B) (g : B ⟶ C) : Hom.hom (f ≫ g) = f.hom ≫ g.hom

@[simp] theorem **mk_hom** {A B : AbVar K} (f : A.X ⟶ B.X) [IsMonHom f] : (Hom.mk f : Hom A B).hom = f

@[simp] theorem **add_hom** {A B : AbVar K} (f g : A ⟶ B) : Hom.hom (f + g) = f.hom ∗ g.hom

@[simp] theorem **zero_hom** (A B : AbVar K) : Hom.hom (0 : A ⟶ B) = 1

@[simp] theorem **neg_hom** {A B : AbVar K} (f : A ⟶ B) : Hom.hom (-f) = f.hom⁻¹

@[simp] theorem **nsmul_hom** {A B : AbVar K} (n : ℕ) (f : A ⟶ B) : Hom.hom (n • f) = f.hom ^ n

@[simp] theorem **zsmul_hom** {A B : AbVar K} (n : ℤ) (f : A ⟶ B) : Hom.hom (n • f) = f.hom ^ n

@[simp] theorem **ptMap_apply** {A B : AbVar K} (f : A ⟶ B) (x : A.Pt) : ptMap f x = x ≫ f.hom

theorem **lev_tmap** {A B : AbVar K} (f : A ⟶ B) (t : Tl l A) (n : ℕ) : (tmap l A B f t).lev n = t.lev n ≫ f.hom

theorem **tmap_comp** {A B C : AbVar K} (f : A ⟶ B) (g : B ⟶ C) : tmap l A C (f ≫ g) = tmap l B C g ∘ₗ tmap l A B f

theorem **tmap_id** (A : AbVar K) : tmap l A A (𝟙 A) = LinearMap.id

theorem **tmap_galT** {A B : AbVar K} (f : A ⟶ B) (σ : Gal K) : tmap l A B f ∘ₗ galT l A σ = galT l B σ ∘ₗ tmap l A B f

@[simp] theorem **prodList_nil** : prodList ([] : List (AbVar K)) = zero K

@[simp] theorem **prodList_cons** (A : AbVar K) (s : List (AbVar K)) : prodList (A :: s) = prod A (prodList s)

theorem **ell_isAmpleCls** (W : WeierstrassCurve K) [W.IsElliptic] : IsAmpleCls (A

theorem **exists_dim_eq_one** (K : Type) [Field K] : ∃ A : AbVar K, A.dim = 1

theorem **exists_dim_one** (K : Type) [Field K] : ∃ A : AbVar K, A.dim = 1

theorem **isIsogeny_id** (A : AbVar K) : IsIsogeny (𝟙 A)

theorem **deg_id** (A : AbVar K) : deg (𝟙 A) = 1

theorem **lam_eq** {A : AbVar K} (c : Pic A.X.left) {Y : Over (Sp K)} (x y : Y ⟶ A.X) : lam c x y = CechPic.lam c x y

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsField

theorem **hypsMet_of** (D : D7) (A V : HSp D) (han : IsAniso V) : HypsMet D.host.F D.host.E D.sp.cm.sig A.rank V.rank

theorem **hypsMet** (D : D7) (n m : ℕ) : HypsMet D.host.F D.host.E D.sp.cm.sig n m

theorem **ranges_met** (D : D7) (n : ℕ) (hn : 1 ≤ n) : (∃ m, 1 ≤ m ∧ 2 * n < m ∧ HypsMet D.host.F D.host.E D.sp.cm.sig n m) ∧ (∃ m, 1 ≤ m ∧ (n < m ∧ m ≤ 2 * n) ∧ HypsMet D.host.F D.host.E D.sp.cm.sig n m) ∧ ∃ m, 1 ≤ m ∧ m ≤ n ∧ HypsMet D.host.F D.host.E D.sp.cm.sig n m

theorem **exists_field** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E), ∀ n m, HypsMet F E σ n m

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsFlagModelField

theorem **idem_low** {R L : Type} [CommRing R] [Field L] (φ : R →+* L) (hφ : Function.Injective φ) {n m : ℕ} (hm : m ≤ 1) (e : Matrix (Fin n) (Fin n) R) (he : e * e = e) (N : Submodule L (Fin n → L)) (hN : ∀ v, v ∈ N ↔ (e.map φ).mulVec v = v) (hr : Module.finrank L N = m) : (∑ ix : Fin m → Fin n, ∏ i, e (ix i) (ix i) = 1) ∧ ∀ ix : Fin m → Fin n, (∏ i, e (ix i) (ix i)) • e = e.submatrix id ix * e.submatrix ix id

theorem **initSpan_low** {R N : Type} [CommRing R] [AddCommGroup N] [Module R N] {m : ℕ} (hm : m ≤ 1) {v v' : Fin m → N} (h : initSpan R v m = initSpan R v' m) (i : ℕ) : initSpan R v i = initSpan R v' i

theorem **initSpan_frVec_top** {S : Type} [CommRing S] [Algebra c.F S] (P : FrPt Xo c S) (M : MChart Y) (h : ⊤ ≤ (frX P ≫ Y.j) ⁻¹ᵁ M.U) (σ : E →+* ℂ) : initSpan S (frVec Y P rfl σ) (Dw.mult σ) = eigLie c (frX P ≫ Y.j) σ

theorem **nonempty_proj_id** : Nonempty (Proj (RingHom.id c.F))

theorem **nonempty_proj_of_mem** (p : Proj (RingHom.id c.F)) (h : ∀ σ k, p.κ σ k ∈ β.range) : Nonempty (Proj β)

theorem **str_eq** {x : Sp S ⟶ Y.X} {g : O →+* S} (h : x ≫ Y.s = spMap g) : str x = g

theorem **ev_str** (M : MChart Y) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (o : O) : ev x M.U h (str (tau M.hU) o) = str x o

theorem **str_gen** [Algebra c.F S] {y : Sp S ⟶ Xo.XF c} (hy : y ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (o : O) : str (y ≫ Y.j) o = algebraMap c.F S (β o)

theorem **pm_map** (M : MChart Y) (σ : E →+* ℂ) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) : (pm p M σ).map (ev x M.U h) = ∑ k, str x (p.κ σ k) • (M.ρ (p.a σ k)).map (ev x M.U h)

theorem **vec_pm** (M : MChart Y) (σ : E →+* ℂ) (x : Sp S ⟶ Y.X) (h : ⊤ ≤ x ⁻¹ᵁ M.U) (cv : Fin M.n → S) : vec M x h (((pm p M σ).map (ev x M.U h)).mulVec cv) = ∑ k, str x (p.κ σ k) • actR Y.t.act (p.a σ k) (vec M x h cv)

theorem **pm_gen** [Algebra c.F S] (M : MChart Y) (σ : E →+* ℂ) {y : Sp S ⟶ Xo.XF c} (hy : y ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (h : ⊤ ≤ (y ≫ Y.j) ⁻¹ᵁ M.U) : (∀ cv, M.Eig (ev _ M.U h) σ cv ↔ ((pm p M σ).map (ev _ M.U h)).mulVec cv = cv) ∧ (pm p M σ).map (ev _ M.U h) * (pm p M σ).map (ev _ M.U h) = (pm p M σ).map (ev _ M.U h)

theorem **pm_low** (hm : ∀ σ, Dw.mult σ ≤ 1) (hg : GenIn Y) (M : MChart Y) (σ : E →+* ℂ) : pm p M σ * pm p M σ = pm p M σ ∧ (∑ ix : Fin (Dw.mult σ) → Fin M.n, ∏ i, pm p M σ (ix i) (ix i) = 1) ∧ ∀ ix : Fin (Dw.mult σ) → Fin M.n, (∏ i, pm p M σ (ix i) (ix i)) • pm p M σ = (pm p M σ).submatrix id ix * (pm p M σ).submatrix ix id

theorem **exists_eCh** (hm : ∀ σ, Dw.mult σ ≤ 1) (hg : GenIn Y) (M : GChart Y) (x : Y.X) (hx : x ∈ M.U) : ∃ k : ECh Y p, x ∈ k.V

theorem **ECh.fix_iff** (k : ECh Y p) (σ : E →+* ℂ) (z : Sp S ⟶ Y.X) (hz : ⊤ ≤ z ⁻¹ᵁ k.V) (h : ⊤ ≤ z ⁻¹ᵁ k.M.U) (cv : Fin k.M.n → S) : ((pm p k.M.toMChart σ).map (ev z _ h)).mulVec cv = cv ↔ ∃ a, ((k.u σ).map (ev z _ hz)).mulVec a = cv

theorem **eigFrames_top** (hm : ∀ σ, Dw.mult σ ≤ 1) (hc : ∀ x : Y.X, ∃ k : ECh Y p, x ∈ k.V) : (eigFrames p hm hc).TopEigen

theorem **exists_flagModel** [IsNoetherianRing O] (p : Proj β) (hp : Dw.p ≤ 1) (hq : Dw.q ≤ 1) (hg : GenIn Y) (hP : PushInvertible Y.t.G Y.t.PP.L) : ∃ X : FlagModel Y, X.YY = Y.X

theorem **exists_field** (A : Type) [CommRing A] [IsDomain A] : ∃ (L : Type) (_ : Field L) (f : A →+* L), Function.Injective f

theorem **genIn_of** {O : Type} [CommRing O] {β : O →+* c.F} {Y : Model Xo c β} (h : ∀ (M : MChart Y) (L : Type) [Field L] (f : Γ(Y.X, M.U) →+* L), Function.Injective f → ∃ (_ : Algebra c.F L) (y : Sp L ⟶ Xo.XF c), y ≫ pullback.snd _ _ = spMap (algebraMap c.F L) ∧ y ≫ Y.j = spMap f ≫ tau M.hU) : GenIn Y

theorem **genIn_dvr** {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Algebra O c.F] [IsFractionRing O c.F] (Y : Model Xo c (algebraMap O c.F)) : GenIn Y

theorem **spec_generic** {A L : Type} [CommRing A] [Field L] (f : A →+* L) (hf : Function.Injective f) (q : PrimeSpectrum L) (t : PrimeSpectrum A) : PrimeSpectrum.comap f q ⤳ t

theorem **genIn_id** (Y : Model Xo c (RingHom.id c.F)) : GenIn Y

theorem **exists_flagModel_field** (Y : Model Xo c (RingHom.id c.F)) (hp : Dw.p ≤ 1) (hq : Dw.q ≤ 1) (hP : PushInvertible Y.t.G Y.t.PP.L) : ∃ X : FlagModel Y, X.YY = Y.X

theorem **exists_flagModel_dvr** {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Algebra O c.F] [IsFractionRing O c.F] (Y : Model Xo c (algebraMap O c.F)) (p : Proj (RingHom.id c.F)) (hint : ∀ σ k, p.κ σ k ∈ (algebraMap O c.F).range) (hp : Dw.p ≤ 1) (hq : Dw.q ≤ 1) (hP : PushInvertible Y.t.G Y.t.PP.L) : ∃ X : FlagModel Y, X.YY = Y.X

theorem **nonempty_of_frPt** {O : Type} [CommRing O] {β : O →+* c.F} {Y : Model Xo c β} (X : FlagCover Y) {K : Type} [Field K] [Algebra c.F K] (P : FrPt Xo c K) : Nonempty X.YY

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsFlagModelHermSum

theorem **ProjSpace.isRegSurf_line** : IsRegSurf (PSpace.str ℤ 1)

theorem **deg_pos** : 0 < C.deg

theorem **deg_self** (h : IsRegSurf p) : (self h).deg = 1

theorem **IsInter.unique** (hX : IsArithVar p 1) {a b : ℝ} (ha : L.IsInter M a) (hb : L.IsInter M b) : a = b

theorem **IsInter.triv_shift** {t a : ℝ} (h : ((triv p).shift (fun _ => t) fun _ => rfl).IsInter M a) [Finite (k →+* ℂ)] : a = t + ArInt.gdeg M 1

theorem **VMetric.isOrthSum_orthSum** (h₁ : VMetric p E₁ r) (h₂ : VMetric p E₂ s) : (h₁.orthSum h₂).IsOrthSum h₁ h₂

theorem **VMetric.IsCInf.orthSum** {h₁ : VMetric p E₁ r} {h₂ : VMetric p E₂ s} (H₁ : h₁.IsCInf) (H₂ : h₂.IsCInf) : (h₁.orthSum h₂).IsCInf

theorem **IsFlagCh2.unique** {D : ℕ} (hX : IsArithVar p D) {H : h.IsCInf} {fl : Vert.Flag E r} (ha : IsFlagCh2 H fl a) (hb : IsFlagCh2 H fl b) : a = b

theorem **isFlagCh2_rank_one** {h : VMetric p E 1} (H : h.IsCInf) (fl : Vert.Flag E 1) : IsFlagCh2 H fl a ↔ (grLB fl 0 H).IsSelfInter (2 * a)

theorem **isFlagCh2_rank_zero** {h : VMetric p E 0} (H : h.IsCInf) (fl : Vert.Flag E 0) : IsFlagCh2 H fl a ↔ a = 0

theorem **IsCh2Deg.unique** (ha : h.IsCh2Deg a) (hb : h.IsCh2Deg b) : a = b

theorem **IsCh2Deg.ch2Deg_eq** (ha : h.IsCh2Deg a) : h.ch2Deg = a

theorem **IsCh2Deg.eq_div** (ha : h.IsCh2Deg a) (M : h.FlagModel) (hb : IsFlagCh2 M.cInf M.flag b) : a = b / M.deg

theorem **isCh2Deg_rank_zero** (hX : IsRegSurf p) {h : VMetric p E 0} (H : h.IsCInf) (fl : Vert.Flag E 0) : h.IsCh2Deg 0

theorem **ProjSpace.exists_flagModel_hermSum** : haveI

theorem **cxDeg_congr** {E E' : X.Modules} (φ : E ≅ E') (r : ℕ) : cxDeg p E r = cxDeg p E' r

theorem **isVB_omega** : IsVB (omega P.A) P.g

theorem **hasTheta_ofPolAV** (Q : Out.PolAV S) : (ofPolAV Q).HasTheta

theorem **posDef** : d.H.PosDef

theorem **ex_zero** : (d.u.ex 0).left ≫ d.φ = y ≫ unitSec P.A

theorem **differentiableAt_fn** {W : P.A.left.Opens} (f : Γ(P.A.left, W)) (h : ⊤ ≤ (y ≫ unitSec P.A) ⁻¹ᵁ W) : DifferentiableAt ℂ (d.fn W f) 0

theorem **pull_sqMap_lam** {A : Over S} {A' : Over S'} [GrpObj A] [GrpObj A'] {f : S' ⟶ S} {φ : A'.left ⟶ A.left} (h : IsGrpPullback A' A f φ) (c : Pic A.left) : CechPic.pull (sqMap h.1.w) (lam c (fst A A) (snd A A)) = lam (CechPic.pull φ c) (fst A' A') (snd A' A')

theorem **nonempty_riemannData** (Q : Out.PolAV S) (y : Sp ℂ ⟶ S) : Nonempty ((ofPolAV Q).RiemannData y)

theorem **IsDualRiemann.unique** (H : P.IsDualRiemann t h) (H' : P.IsDualRiemann t h') : h = h'

theorem **IsDualRiemann.scale** (H : P.IsDualRiemann t h) (s : ℝ) : P.IsDualRiemann (Real.exp (-s) * t) (h.scale s)

theorem **IsHodge.flat** (H : P.IsHodge h) : P.IsDualRiemann (1 / 2) (h.scale (Real.log (2 * Real.pi)))

theorem **IsCh2Omega.unique** (ha : P.IsCh2Omega p a) (hb : P.IsCh2Omega p b) : a = b

theorem **IsCh2Omega.ch2Omega_eq** (ha : P.IsCh2Omega p a) : P.ch2Omega p = a

theorem **IsHodge.ch2Omega_eq** (H : P.IsHodge h) (ha : h.IsCh2Deg a) : P.ch2Omega p = a

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsFrameOfRichRec

theorem **exists_val_of_chrEq** {c : (Formalism.L2.Theta.AE D)ˣ →* Circle} {χ : Formalism.L2.Host.HChar D.host.E} (h : ChrEq c χ) (x : (Formalism.L2.Theta.AE D)ˣ) : ∃ y, ((c x : Circle) : ℂ) = ((χ.1 y : ℂˣ) : ℂ)

theorem **level_of_is** {T : Core D} (hT : LawsM T) {Lc : LevCh D T0} {μ0 : T0} (hL : Lc.Is μ0) (hO : T.pb = Lc.C.P.pb) : T.Q.l.Mb = ∏ PP ∈ Lc.Pg, PP.asIdeal

theorem **exists_half_of_is** {Lc : LevCh D T0} {μ0 : T0} (hL : Lc.Is μ0) {p : ℕ} (hp : p = D.host.pde ∨ p = D.host.pde') : ∃ P : HeightOneSpectrum (𝒪 D.host.K), (p : 𝒪 D.host.K) ∈ P.asIdeal ∧ ∀ PP ∈ Lc.Q t, (p : 𝒪 D.host.E) ∈ PP.asIdeal → P.asIdeal.map (algebraMap (𝒪 D.host.K) (𝒪 D.host.E)) ≤ PP.asIdeal

theorem **ownAt_special** (v : HeightOneSpectrum (𝒪 D.what₂.LL)) : ownAt D v = D.host.pde ∨ ownAt D v = D.host.pde'

theorem **oneHalf_of_is** {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) {μ0 : T0} (hL : R.toLevCh.Is μ0) : ∃ Pk : ℕ → HeightOneSpectrum (𝒪 D.host.K), ∀ v, ((ownAt D v : ℕ) : 𝒪 D.host.K) ∈ (Pk (ownAt D v)).asIdeal ∧ ∀ PP ∈ GlobalTheory.OnTies.placesOver R (R.mu v) (ownAt D v), (Pk (ownAt D v)).asIdeal.map (algebraMap (𝒪 D.host.K) (𝒪 D.host.E)) ≤ PP.asIdeal

theorem **coreTies_of_is** {L : SizeData₂ D} {EX : ErrX D} {J : CoreJ D} [Algebra J.Q.c.F D.what₂.LL] {R : RealDat D A J.Q.c.F} {μ0 : T0} (hL : R.toLevCh.Is μ0) (hO : J.pb = R.C.P.pb) (hc : LawsCore J.toCore) (hcan : J.Q.Canon) (hr : ∀ j, 1 ≤ (J.Q.Dw j).p + (J.Q.Dw j).q) (hE : HalfEX J.toCore EX) (h₁ : FieldOK0 D D.host.pde) (h₂ : FieldOK0 D D.host.pde') (hs : FieldSmall D L) (v₁ : ∀ x, ((R.C.del.1 x : ℂˣ) : ℂ) ∈ FO D) (v₂ : ∀ x, ((R.C.del'.1 x : ℂˣ) : ℂ) ∈ FO D) : CoreTies L EX J R

theorem **exists_frame_of_richRec** (D : D7) (EX : ErrX D) (h : RichRec D) (h₁ : FieldOK0 D D.host.pde) (h₂ : FieldOK0 D D.host.pde') : ∃ C : Formalism.L2.Host.CharFrame D, (∀ k, barE D (EX.piG k) ∈ (C.P.pb k).asIdeal.map (algebraMap (𝒪 D.host.K) (𝒪 D.host.E))) ∧ (∀ x, ((C.del.1 x : ℂˣ) : ℂ) ∈ FO D) ∧ (∀ x, ((C.del'.1 x : ℂˣ) : ℂ) ∈ FO D) ∧ ∃ Lc : LevCh D (Option Bool), Lc.C = C ∧ Lc.Is none ∧ Lc.xiH (some false) = C.del ∧ Lc.xiH (some true) = C.del' ∧ (∃ PP ∈ Lc.Q (some false), ((D.host.pde : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) ∧ ∃ PP ∈ Lc.Q (some true), ((D.host.pde' : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal

theorem **exists_runsHold** : ∃ CK x₁ : ℝ, ∀ (D : D7) (G : Type) (J : CoreJ D) (r : TwoRuns J G) (EX : ErrX D) (L : SizeData₂ D) (Pm : ParameterChoice' D L) (Tp : ParamsT D L Pm) (hK : DV.KapDef D), (∀ v j, InR D j → ParOK (r.used v).full.toTop EX (ownAt D v) j) → PageNum Tp x₁ → FieldSmall D L → RunsHold Tp hK EX CK r

theorem **exists_all** : ∃ CK x₁ : ℝ, ∀ (D : D7), RecOK D x₁ → RichRec D → ∀ (G : Type) [Group G] [Fintype G], ∃ (L : SizeData₂ D) (Pm : ParameterChoice' D L) (Tp : ParamsT D L Pm) (hK : DV.KapDef D) (EX : ErrX D) (C : Formalism.L2.Host.CharFrame D) (J : CoreJ D) (r : TwoRuns J G), LawsJ J ∧ J.pb = C.P.pb ∧ (∀ v, LawsFull (r.used v).full) ∧ OwnPin r ∧ PageNum Tp x₁ ∧ RunsHold Tp hK EX CK r ∧ (∃ Lc : LevCh D (Option Bool), Lc.C = C ∧ Lc.Is none ∧ Lc.xiH (some false) = C.del ∧ Lc.xiH (some true) = C.del' ∧ (∃ PP ∈ Lc.Q (some false), ((D.host.pde : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) ∧ ∃ PP ∈ Lc.Q (some true), ((D.host.pde' : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) ∧ (∃ a : Algebra J.Q.c.F D.what₂.LL, @AlgTie D J a) ∧ ∀ (Pl0 T0 : Type) (A : FamIn D.what₂.LL G Pl0 T0 D.R.m) [Algebra J.Q.c.F D.what₂.LL] (R : RealDat D A J.Q.c.F) (μ0 : T0), R.C = C → R.toLevCh.Is μ0 → ∀ (mu : HeightOneSpectrum (𝒪 D.what₂.LL) →

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsGDatumOneOneOne

theorem **exists_gDatum_one_one_one** : ∃ G : GDatum, G.a = 1 ∧ G.n = 1 ∧ G.m = 1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsGenReads

theorem **realM_mt** {θ : InfFam D} (hθ : IsInfFam θ) (W : HSp D) (h : GA D W) (τ : D.host.E →+* ℂ) (hτ : ∃ ρ, τ = tauE D ρ) (th : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ) (hth : IsInfEmb τ th) : realM θ (Formalism.L2.Theta.mt W h) τ = archM W th h

theorem **exists_atComp** {θ : InfFam D} (hθ : IsInfFam θ) {W : HSp D} {r : ℕ} (hr : W.rank = r) {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (dict : GA D W → Σ k, Frame (U.X k).Ψ r) (ι : Dw.La →ₗ[𝒪 D.host.E] (Fin W.rank → D.host.E)) {g : ℕ} (e : Fin g ≃ Σ τ : D.host.E →+* ℂ, Fin (Dw.mult τ)) (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) (fun j => relab c σ (e j).1)) (ik : U.Comp → Fin F.n) (φ : ∀ k, (F.Di (ik k)).La ≃ₗ[𝒪 D.host.E] (U.X k).Λ) (th : ∀ k, (U.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (hΨ : ∀ k, ∀ τ ∈ (U.X k).Ψ, ∃ ρ, τ = tauE D ρ) (hth : ∀ k τ, IsInfEmb τ.1 (th k τ)) (hD : DictLat U W hr dict ι e F ik φ th x₀) (h : GA D W) : ∃ γ, AtComp θ W hr U.X dict x₀ h γ

theorem **exists_genReads** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {σ : c.F →+* ℂ} {r : ℕ} (U : Unf Xo c σ r) (hσ : GoodEmb c σ) (hU : IsUnf U) (h1 : 1 ≤ r) (k : U.Comp) (x : (U.X k).dom) (hx : IsUnitFr (U.X k) x.1) : ∃ (g : ℕ) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] (U.X k).Λ) (idx : Fin g ≃ (U.X k).Ψ × Fin r) (e : Fin g ≃ Lt1 Dw) (ϑ : (Fin g → ℂ) → ℂ), IsSympl (U.X k) bas ∧ IdxIs c σ (U.X k) idx e ∧ GenReads U k bas idx e x ϑ ∧ MeanOne (U.X k) bas idx x.1 (U.X k).Λ ϑ

theorem **ccIs_smul** (h : CcIs θ V W hr U dict mu x₀ om f₀ cc) (a : ℂ) (ha : ‖a‖ = 1) : CcIs θ V W hr U dict mu x₀ om (a • f₀) (fun k => a * cc k)

theorem **ccIs_unique** (hσ : GoodEmb c σ) (hU : IsUnf U) (h1 : 1 ≤ r) (h : CcIs θ V W hr U dict mu x₀ om f₀ cc) (h' : CcIs θ V W hr U dict mu x₀ om f₀ cc') (g γ : GA D W) (hg : AtComp θ W hr U.X dict x₀ g γ) (hd : (dict g).2 ∈ (U.X (dict g).1).dom) (hu : IsUnitFr (U.X (dict g).1) (dict g).2) (w : WO V W) (hw : w.map (finPart D) = 0) (hne : om (1, γ * g) f₀ w ≠ 0) : cc (dict g).1 = cc' (dict g).1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsGivenStageEqOne

theorem **infix_map_range** {α : Type*} (f : ℕ → α) {p n : ℕ} (h : p + 2 ≤ n) : [f p, f (p + 1)] <:+: (List.range n).map f

theorem **canFlip_append** {γ : List Edge} {l l' : List (ℕ × ℕ)} (h : CanFlip γ l) (h' : CanFlip (l.foldl (fun γ x => flipPath x.1 x.2 γ) γ) l') : CanFlip γ (l ++ l')

theorem **canFlip_col** {m i : ℕ} (hi : i < m) : ∀ j, j ≤ m → CanFlip ((List.range (m + m)).map (midEdge m i 0)) ((List.range j).map fun j => (i, j)) ∧ ((List.range j).map fun j => (i, j)).foldl (fun γ x => flipPath x.1 x.2 γ) ((List.range (m + m)).map (midEdge m i 0)) = (List.range (m + m)).map (midEdge m i j) | 0, _ => ⟨.nil _, rfl⟩ | j + 1, hj => by obtain ⟨h1, h2⟩

theorem **canFlip_cols** {m : ℕ} : ∀ i, i ≤ m → CanFlip ((List.range (m + m)).map (midEdge m 0 0)) ((List.range i).flatMap fun i => (List.range m).map fun j => (i, j)) ∧ ((List.range i).flatMap fun i => (List.range m).map fun j => (i, j)).foldl (fun γ x => flipPath x.1 x.2 γ) ((List.range (m + m)).map (midEdge m 0 0)) = (List.range (m + m)).map (midEdge m i 0) | 0, _ => ⟨.nil _, rfl⟩ | i + 1, hi => by obtain ⟨h1, h2⟩

theorem **canFlip_flipOrder** (m : ℕ) : CanFlip (givenPath m) (flipOrder m) ∧ (flipOrder m).length = m ^ 2 ∧ (flipOrder m).foldl (fun γ x => flipPath x.1 x.2 γ) (givenPath m) = oppositePath m

theorem **borelSpaceKw** (w : Pl D.host.E) : BorelSpace (Kw w)

theorem **isUnif_unifAt** (w : Pl D.host.E) : IsUnif (unifAt w)

theorem **depthAt_w** (G : GridData L sc) {v : Pl D.host.F} (hv : v ∈ G.pl.S0) : cosExp G (G.pl.w v) = G.pl.depth sc v ∧ depthAt G (G.pl.w v) = G.pl.depth sc v

theorem **depthAt_of_notMem** (G : GridData L sc) {w : Pl D.host.E} (hw : w ∉ G.pl.W0) : cosExp G w = 0 ∧ depthAt G w = 1

theorem **rank_vertex** (G : GridData L sc) (t : T) (w : Pl D.host.E) {X : ℕ × ℕ} (h : X.1 ≤ D.R.m ∧ X.2 ≤ D.R.m) : (locGrid G t w).rank X = (G.V (.grid X.1 X.2)).rank

theorem **isTame_loc** {χ : HChar D.host.E} {w : Pl D.host.E} (h : χ.TameAt w) : IsTame (χ.loc w)

theorem **isTame_muBase** (w : Pl D.host.E) (j : ℕ) : IsTame ((muBase L j).loc w)⁻¹

theorem **atPl_spec** (w : Pl D.host.E) (x : Kw w) : infPart D (atPl w x) = 0 ∧ ∀ Q, cpt (D

theorem **ae_ext** {z z' : AE D} (h : infPart D z = infPart D z') (h' : ∀ Q, cpt (D

theorem **atPl_add** (w : Pl D.host.E) (x y : Kw w) : atPl w (x + y) = atPl w x + atPl w y

theorem **colFrame_lt** (L : LevCh D T) {j : ℕ} (h : j < D.R.m) : HEq (colFrame L j) (L.βY j)

theorem **colFrame_m** (L : LevCh D T) : HEq (colFrame L D.R.m) L.βX

theorem **isFrame_β** {μ₀ : T} (hL : L.Is μ₀) {i j : ℕ} (hi : i ≤ D.R.m) (hj : j ≤ D.R.m) : IsFrame (B.β i j) (G.Λ (.grid i j))

theorem **ixFin_val** (j : ℕ) : (∀ a, (ixFin G t w j (.inl a)).val = a.val) ∧ ∀ c b, (ixFin G t w j (.inr (c, b))).val = D.R.r + (b.val + D.R.t * c.val)

theorem **flips_path** {k : Type*} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {Γ : Grid k} (s : Assignment) (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (l : List (ℕ × ℕ)) (S : Γ.PiLocalization₂) : (S.flips s ψ ϖ e l).path = l.foldl (fun γ x => flipPath x.1 x.2 γ) S.path

theorem **flips_level** {k : Type*} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {Γ : Grid k} (s : Assignment) (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (l : List (ℕ × ℕ)) (S : Γ.PiLocalization₂) (h : ∀ i j, S.level (i + 1, j + 1) = KO k _) : (S.flips s ψ ϖ e l).level = S.level

theorem **flipStage_path_level** (W : G.Weil) (Φ : G.Datum) : (flipStage t w B W Φ).path = oppositePath D.R.m ∧ (flipStage t w B W Φ).level = givenLevel G t w

theorem **exists_stepFn_eq_one** (j : ℕ) : ∃ x, stepFn L t w j x = 1

theorem **exists_givenStage_eq_one** (Φ : G.Datum) : ∃ x, (givenStage t w B Φ).tensor x = 1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIntMulIntegral

theorem **Datum.EF_smulF_eq_sum** (c : 𝒪 D.F) (x y : D.La) : D.EF (D.smulF c x) y = ∑ k, (RingOfIntegers.basis D.F).repr c k * D.EF (D.smulF (RingOfIntegers.basis D.F k) x) y

theorem **Datum.skewFun_apply** (x y : D.La) (c : 𝒪 D.F) : D.skewFun x y (algebraMap (𝒪 D.F) D.F c) = (D.EF (D.smulF c x) y : ℚ)

theorem **Datum.trace_skew_mul** (x y : D.La) (c : 𝒪 D.F) : Algebra.trace ℚ D.F (D.skew x y * algebraMap (𝒪 D.F) D.F c) = (D.EF (D.smulF c x) y : ℚ)

theorem **eq_of_trace_mul** {F : Type} [Field F] [NumberField F] {s t : F} (h : ∀ c : 𝒪 F, Algebra.trace ℚ F (s * algebraMap (𝒪 F) F c) = Algebra.trace ℚ F (t * algebraMap (𝒪 F) F c)) : s = t

theorem **Datum.bar_bar** (c : 𝒪 D.F) : D.bar (D.bar c) = c

theorem **Datum.EF_smulF_right** (c : 𝒪 D.F) (x y : D.La) : D.EF x (D.smulF c y) = D.EF (D.smulF (D.bar c) x) y

theorem **Datum.EF_swap** (x y : D.La) : D.EF y x = - D.EF x y

theorem **Datum.skew_swap** (x y : D.La) : D.skew y x = - IsCMField.complexConj D.F (D.skew x y)

theorem **Datum.conj_ω** : IsCMField.complexConj D.F (D.ω : D.F) = D.tr - D.ω

theorem **Datum.ω_sq** : (D.ω : D.F) ^ 2 = D.tr * D.ω - D.nm

theorem **Datum.conj_del** : IsCMField.complexConj D.F D.del = - D.del

theorem **Datum.del_sq** : D.del ^ 2 = ((D.tr ^ 2 - 4 * D.nm : ℤ) : D.F)

theorem **Datum.conj_hermF** (x y : D.La) : IsCMField.complexConj D.F (D.hermF x y) = D.hermF y x

theorem **Datum.gram_herm** (e : Fin D.n → D.La) : ((D.gram e).map (IsCMField.complexConj D.F : D.F →+* D.F))ᵀ = D.gram e

theorem **Datum.isIntegral_ω** : IsIntegral ℚ (D.ω : D.F)

theorem **Datum.quadQ_natDegree** : D.quadQ.natDegree = 2

theorem **Datum.quadQ_monic** : D.quadQ.Monic

theorem **Datum.aeval_quadQ** : aeval (D.ω : D.F) D.quadQ = 0

theorem **Datum.ω_ne_rat** (q : ℚ) : (D.ω : D.F) ≠ algebraMap ℚ D.F q

theorem **Datum.minpoly_ω** : minpoly ℚ (D.ω : D.F) = D.quadQ

theorem **Datum.hKq** : Module.finrank ℚ D.Kq = 2 ∧ ∀ τ : D.F →+* ℂ, ∃ x ∈ D.Kq, (τ x).im ≠ 0

theorem **Datum.uOf_ω** (r : ℚ_[p]) (hr : r ^ 2 - D.tr * r + D.nm = 0) : D.uOf r hr (D.ω, IntermediateField.mem_adjoin_simple_self ℚ _) = r

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsClosedImmersionGenericPoint

theorem **CInfAt.congr** (h : CInfAt f y) (e : f' =ᶠ[𝓝 y] f) : CInfAt f' y

theorem **cInfAt_const** (c : ℝ) (y : CxPts p σ) : CInfAt (fun _ => c) y

theorem **CInfAt.comp₂** (h : CInfAt f y) (h' : CInfAt f' y) {G : ℝ × ℝ → ℝ} (hG : ContDiff ℝ ∞ G) : CInfAt (fun z => G (f z, f' z)) y

theorem **CInfAt.add** (h : CInfAt f y) (h' : CInfAt f' y) : CInfAt (fun z => f z + f' z) y

theorem **CInfAt.const_mul** (h : CInfAt f y) (c : ℝ) : CInfAt (fun z => c * f z) y

theorem **CInfAt.neg** (h : CInfAt f y) : CInfAt (fun z => -f z) y

theorem **CInfAt.continuousAt** (h : CInfAt f y) : ContinuousAt f y

theorem **ev_app** {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) (W : X.Opens) (g : Γ(X, W)) (z : CxPts q σ) : ev q σ (π ⁻¹ᵁ W) (π.app W g) z = ev p σ W g (PtsOver.map π hπ _ z)

theorem **CInfAt.pull** {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) {z : CxPts q σ} (h : CInfAt f (PtsOver.map π hπ _ z)) : CInfAt (fun w => f (PtsOver.map π hπ _ w)) z

theorem **contDiffOn_chart** {S : Set (CxPts p σ)} (h : ∀ y ∈ S, CInfAt f y) (c : AlgChart p σ D) : ContDiffOn ℝ ∞ (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)

theorem **cInfAt_of_chart** {S : Set (CxPts p σ)} (hS : IsOpen S) (c : AlgChart p σ D) (h : ContDiffOn ℝ ∞ (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)) (hy : y ∈ c.e.source) (hyS : y ∈ S) : CInfAt f y

theorem **cInfAt_iff_smoothOn** (R : AnRead p σ D) {S : Set (CxPts p σ)} (hS : IsOpen S) : (∀ y ∈ S, CInfAt f y) ↔ R.SmoothOn S f

theorem **isCInf_iff_isSmooth** (R : AnRead p σ D) : φ.IsCInf ↔ φ.IsSmooth R

theorem **IsCInf.addFn** (h : φ.IsCInf) {u : CxPts p σ → ℝ} (hu : ∀ y, CInfAt u y) : (φ.addFn u).IsCInf

theorem **IsCInf.tens** {ψ : Weight p σ A} (h' : ψ.IsCInf) (h : φ.IsCInf) : (ψ.tens φ).IsCInf

theorem **IsCInf.dual** (h : φ.IsCInf) : φ.dual.IsCInf

theorem **IsCInf.zpow** (h : φ.IsCInf) (n : ℤ) : (φ.zpow n).IsCInf

theorem **IsCInf.pull** (h : φ.IsCInf) {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) : (φ.pull π hπ).IsCInf

theorem **isCInf_ofFun** {u : CxPts p σ → ℝ} (hu : ∀ y, CInfAt u y) : (ofFun u).IsCInf

theorem **isCInf_fsBasis** {α : Type} [Fintype α] [DecidableEq α] (hB : B.IsLB) (b : α → B.Sec) (hb : ⊔ j, nonzeroLocus (b j) = ⊤) (M : Matrix α α ℂ) (hM : M.det ≠ 0) : (fsBasis p σ hB b hb M hM).IsCInf

theorem **IsCInf.continuous_nsq** (h : φ.IsCInf) (s : B.Sec) : Continuous (φ.nsq (secVal p σ s))

theorem **IsCInf.contDiffAt** (h : φ.IsCInf) (c : AlgChart p σ D) {y : CxPts p σ} (hy : y ∈ c.e.source) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : ContDiffAt ℝ 2 (fun v => φ.φ i (c.e.symm v)) (c.e y)

theorem **semiPosReg_iff** (h : φ.IsCInf) (R : AnRead p σ D) : φ.SemiPosReg D ↔ φ.SemiPos R

theorem **hess_eq** (h : φ.IsCInf) (hy : y ∈ c.e.source) {i j : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) : c.hess φ j y = c.hess φ i y

theorem **chernDens_eq** (h : φ.IsCInf) (hy : y ∈ c.e.source) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : c.chernDens φ y = D.factorial / Real.pi ^ D * (c.hess φ i y).det.re

theorem **chernDens_trans** (h : φ.IsCInf) (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : c.chernDens φ y = ‖transDet c.e c'.e y‖ ^ 2 * c'.chernDens φ y

theorem **chernDens_nonneg** (h : φ.IsCInf) (h' : φ.SemiPosReg D) (hy : y ∈ c.e.source) : 0 ≤ c.chernDens φ y

theorem **chernDens_zero** (c : AlgChart p σ 0) (y : CxPts p σ) : c.chernDens φ y = 1

theorem **chernDens_addFn_const** (t : ℝ) (y : CxPts p σ) : c.chernDens (φ.addFn fun _ => t) y = c.chernDens φ y

theorem **ChernIntegrableOn.congr** (h : φ.ChernIntegrableOn D S f) (hS : MeasurableSet S) (e : EqOn f g S) : φ.ChernIntegrableOn D S g

theorem **chernInt_congr** (hS : MeasurableSet S) (e : EqOn f g S) : φ.chernInt D S f = φ.chernInt D S g

theorem **chernInt_add** (hf : φ.ChernIntegrableOn D S f) (hg : φ.ChernIntegrableOn D S g) : φ.chernInt D S (fun y => f y + g y) = φ.chernInt D S f + φ.chernInt D S g

theorem **chernInt_const_mul** (a : ℝ) : φ.chernInt D S (fun y => a * f y) = a * φ.chernInt D S f

theorem **chernPart_addFn_const** (t : ℝ) (ε : ℤˣ) : (φ.addFn fun _ => t).chernPart D ε = φ.chernPart D ε

theorem **chernPart_eq_zero** (ε : ℤˣ) (hS : MeasurableSet S) (h : ∀ c : AlgChart p σ D, Disjoint c.e.source S) : φ.chernPart D ε S = 0

theorem **chernPart_apply** (h : φ.IsCInf) (ε : ℤˣ) (c : AlgChart p σ D) (hS : MeasurableSet S) (hc : S ⊆ c.e.source) : φ.chernPart D ε S = ∫⁻ v in c.e '' S, ENNReal.ofReal ((ε : ℤ) * c.chernDens φ (c.e.symm v))

theorem **chernPart_neg_eq_zero** (h : φ.IsCInf) (h' : φ.SemiPosReg D) : φ.chernPart D (-1) = 0

theorem **UnitAt.ord_eq_zero** (h : UnitAt z f) : Scheme.ord f z = 0

theorem **ord_algebraMap** (hz : Order.coheight z = 1) {a : Z.presheaf.stalk z} (ha : a ≠ 0) : Scheme.ord (algebraMap _ Z.functionField a) z = (Module.length (Z.presheaf.stalk z) (Z.presheaf.stalk z ⧸ Ideal.span {a})).toNat

theorem **IsEqn.mult_eq** (h : D.IsEqn z g) : D.mult z = Scheme.ord (g : Z.functionField) z

theorem **mult_add** (D E : CartierDiv Z) (z : Z) : (D + E).mult z = D.mult z + E.mult z

theorem **coheight_eq_one_of_mult_ne_zero** (h : D.mult z ≠ 0) : Order.coheight z = 1

theorem **mem_supp_of_mult_ne_zero** (h : D.mult z ≠ 0) : z ∈ D.supp

theorem **finite_support_mult** [CompactSpace Z] (D : CartierDiv Z) : (Function.support D.mult).Finite

theorem **ratVal_fnOf** {U : Z.Opens} (hy : y ∈ ptsIn q σ U) (h : genericPoint Z ∈ U) (a : Γ(Z, U)) : ratVal (fnOf U h a) y = ev q σ U a y

theorem **ratVal_mul** (hf : RegAt (under y) f) (hg : RegAt (under y) g) : ratVal (f * g) y = ratVal f y * ratVal g y

theorem **ratVal_ne_zero** (hf : UnitAt (under y) f) : ratVal f y ≠ 0

theorem **isOpen_offSupp** (D : CartierDiv Z) : IsOpen (offSupp q σ D)

theorem **mem_offSupp** {D : CartierDiv Z} : y ∈ offSupp q σ D ↔ under y ∉ D.supp

theorem **IsRatSec.unitAt** (hL : L.IsLB) (hr : IsRatSec L r) (hy : y ∈ offSupp q σ (divOfRatSec hL hr)) {i : L.ι} (hi : y ∈ ptsIn q σ (L.U i)) : UnitAt (under y) (r i)

theorem **logNorm_eq** (φ : Weight q σ L) (hr : IsRatSec L r) (hy : y ∈ offSupp q σ (divOfRatSec φ.isLB hr)) {i : L.ι} (hi : y ∈ ptsIn q σ (L.U i)) : logNorm φ r y = Real.log ‖ratVal (r i : Z.functionField) y‖ - φ.φ i y / 2

theorem **logNorm_addFn** (φ : Weight q σ L) (u : CxPts q σ → ℝ) (r : L.ι → Z.functionFieldˣ) (y : CxPts q σ) : logNorm (φ.addFn u) r y = logNorm φ r y - u y / 2

theorem **exists_isRatSec** (hL : L.IsLB) : ∃ r, IsRatSec L r

theorem **fnOf_sec_ne_zero** {s : L.Sec} (hs : genericPoint Z ∈ nonzeroLocus s) {i : L.ι} (h : genericPoint Z ∈ L.U i) : fnOf _ h (s.f i) ≠ 0

theorem **val_ratOfSec** {i : L.ι} (h : genericPoint Z ∈ L.U i) : (ratOfSec s hs i : Z.functionField) = fnOf _ h (s.f i)

theorem **isRatSec_ratOfSec** : IsRatSec L (ratOfSec s hs)

theorem **notMem_supp_divOfSec** {z : Z} : z ∉ (divOfSec hL s hs).supp ↔ z ∈ nonzeroLocus s

theorem **offSupp_divOfSec** : offSupp q σ (divOfSec hL s hs) = ptsIn q σ (nonzeroLocus s)

theorem **mult_divOfSec** [IsLocallyNoetherian Z] {z : Z} (hz : Order.coheight z = 1) {i : L.ι} (hi : z ∈ L.U i) : (divOfSec hL s hs).mult z = (Module.length (Z.presheaf.stalk z) (Z.presheaf.stalk z ⧸ Ideal.span {Z.presheaf.germ _ z hi (s.f i)})).toNat

theorem **mult_divOfSec_nonneg** [IsLocallyNoetherian Z] (z : Z) : 0 ≤ (divOfSec hL s hs).mult z

theorem **logNorm_ratOfSec** (φ : Weight q σ L) (hy : y ∈ ptsIn q σ (nonzeroLocus s)) : logNorm φ (ratOfSec s hs) y = Real.log (φ.nsq (secVal q σ s) y) / 2

theorem **IsRatSec.pow_one** (hr : IsRatSec L r) : IsRatSec (lbPow L 1) fun j => r (j PUnit.unit)

theorem **divOfRatSec_pow_one** (hL : L.IsLB) (hr : IsRatSec L r) (h : (lbPow L 1).IsLB) : divOfRatSec h hr.pow_one = divOfRatSec hL hr

theorem **logNorm_pow_one** (φ : Weight q σ L) (hr : IsRatSec L r) (hy : y ∈ offSupp q σ (divOfRatSec φ.isLB hr)) : logNorm (φ.pow 1) (fun j => r (j PUnit.unit)) y = logNorm φ r y

theorem **genSmooth_of_smooth** (p : X ⟶ Sp k) (D : ℕ) [SmoothOfRelativeDimension D p] : GenSmooth p D

theorem **flat_int_iff_charZero** [IsIntegral X] (t : X ⟶ Sp ℤ) : Flat t ↔ CharZero Γ(X, ⊤)

theorem **isNoetherianRing** : IsNoetherianRing k

theorem **isNoetherian** : IsNoetherian X

theorem **isLocallyNoetherian** : IsLocallyNoetherian X

theorem **charZero** : CharZero Γ(X, ⊤)

theorem **charZero_base** : CharZero k

theorem **finite_emb** : Finite (k →+* ℂ)

theorem **compactSpace** (σ : k →+* ℂ) : CompactSpace (CxPts p σ)

theorem **of_isClosedImmersion** (i : Y ⟶ X) [IsClosedImmersion i] [IsIntegral Y] (hc : CharZero Γ(Y, ⊤)) {e : ℕ} (he : topologicalKrullDim Y = (e + 1 : ℕ)) : IsArithVar (i ≫ p) e

theorem **eq_conj_of_conj** (h : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) : σ' = (starRingEnd ℂ).comp σ

theorem **isSmooth** {D : ℕ} (R : AnRead p σ D) : (L.φ σ).IsSmooth R

theorem **nsq_shift** (t : (k →+* ℂ) → ℝ) (ht : ∀ σ, t ((starRingEnd ℂ).comp σ) = t σ) (s : L.B.ι → CxPts p σ → ℂ) (y : CxPts p σ) : ((L.shift t ht).φ σ).nsq s y = (L.φ σ).nsq s y * Real.exp (-(2 * t σ))

theorem **zero_mem_smallSec** : 0 ∈ L.smallSec

theorem **semiPos_iff** {D : ℕ} (R : ∀ σ, AnRead p σ D) : L.SemiPos D ↔ ∀ σ, (L.φ σ).SemiPos (R σ)

theorem **IsArithHeight.recursion_one** (H : L.IsArithHeight h) {Z : Scheme.{0}} (i : Z ⟶ X) [IsClosedImmersion i] [IsIntegral Z] [IsLocallyNoetherian Z] (hc : CharZero Γ(Z, ⊤)) {e : ℕ} (he : 1 ≤ e) (hd : topologicalKrullDim Z = (e + 1 : ℕ)) {r : (L.restr i).B.ι → Z.functionFieldˣ} (hr : IsRatSec (L.restr i).B r) : (L.restr i).ArchIntegrable (L.restr i) e hr ∧ h (i (genericPoint Z)) = ∑ᶠ z, (((L.restr i).div hr).mult z : ℝ) * h (i z) - (L.restr i).archInt (L.restr i) e hr

theorem **IsArithHeight.closedPoint** (H : L.IsArithHeight h) {x : X} (hx : IsClosed {x}) [Finite (X.residueField x)] : h x = Real.log (Nat.card (X.residueField x))

theorem **exists_isClosedImmersion_genericPoint** (x : X) : ∃ (Z : Scheme.{0}) (i : Z ⟶ X) (_ : IsClosedImmersion i) (_ : IsIntegral Z), i (genericPoint Z) = x

theorem **isNoetherian_sub** : IsNoetherian Z

theorem **exists_dim_eq** : ∃ d : ℕ, topologicalKrullDim Z = d

theorem **dim_ne_zero** (hc : CharZero Γ(Z, ⊤)) : topologicalKrullDim Z ≠ 0

theorem **vertical_or_horizontal** : (∃ (ℓ : ℕ) (_ : Fact ℓ.Prime), Nonempty (Algebra (ZMod ℓ) Γ(Z, ⊤))) ∨ (CharZero Γ(Z, ⊤) ∧ ∃ e : ℕ, topologicalKrullDim Z = (e + 1 : ℕ))

theorem **IsArithHeight.unique** (hX : IsArithVar p D) (H : L.IsArithHeight h) (H' : L.IsArithHeight h') : h = h'

theorem **IsSelfInter.unique** (hX : IsArithVar p D) [IsIntegral X] {a b : ℝ} (ha : L.IsSelfInter a) (hb : L.IsSelfInter b) : a = b

theorem **IsArithHeight.isNef_iff** (hX : IsArithVar p D) (H : L.IsArithHeight h) : L.IsNef D ↔ L.SemiPos D ∧ ∀ (Z : Scheme.{0}) (i : Z ⟶ X) [IsClosedImmersion i] [IsIntegral Z], topologicalKrullDim Z = 1 → 0 ≤ h (i (genericPoint Z))

theorem **Weight.nsq_pull_secPull** {B : LB X} (φ : Weight p σ B) {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) (s : B.Sec) (z : CxPts q σ) : (φ.pull π hπ).nsq (secVal q σ (secPull s π)) z = φ.nsq (secVal p σ s) (PtsOver.map π hπ _ z)

theorem **IsArithVar.finite_residueField** {D : ℕ} (hX : IsArithVar p D) {x : X} (hx : IsClosed {x}) : Finite (X.residueField x)

theorem **IsArithHeight.recursion_sec** (H : L.IsArithHeight h) [IsClosedImmersion i] (hc : CharZero Γ(Z, ⊤)) {e : ℕ} (he : 1 ≤ e) (hd : topologicalKrullDim Z = (e + 1 : ℕ)) : (∀ σ, ((L.restr i).φ σ).ChernIntegrableOn e (ptsIn _ σ (nonzeroLocus (secPull u i))) fun y => Real.log ((L.φ σ).nsq (secVal p σ u) (PtsOver.map i rfl _ y)) / 2) ∧ h (i (genericPoint Z)) = ∑ᶠ z, ((divOfSec (L.restr i).isLB _ hu).mult z : ℝ) * h (i z) - ∑ᶠ σ, ((L.restr i).φ σ).chernInt e (ptsIn _ σ (nonzeroLocus (secPull u i))) fun y => Real.log ((L.φ σ).nsq (secVal p σ u) (PtsOver.map i rfl _ y)) / 2

theorem **curveDeg_sec** (h0 : ∀ σ (y : CxPts (i ≫ p) σ), y ∈ ptsIn _ σ (nonzeroLocus (secPull u i))) : (L.restr i).curveDeg (isRatSec_ratOfSec _ hu) = ∑ᶠ z, ((divOfSec (L.restr i).isLB _ hu).mult z : ℝ) * Real.log (Nat.card (Z.residueField z)) - ∑ᶠ σ, ∑ᶠ y : CxPts (i ≫ p) σ, Real.log ((L.φ σ).nsq (secVal p σ u) (PtsOver.map i rfl _ y)) / 2

theorem **semiPosReg_fsBasis** {α : Type} [Fintype α] [DecidableEq α] (hB : B.IsLB) (b : α → B.Sec) (hb : ⊔ j, nonzeroLocus (b j) = ⊤) (M : Matrix α α ℂ) (hM : M.det ≠ 0) : (fsBasis p σ hB b hb M hM).SemiPosReg D

theorem **SemiPosReg.addFn_const** {φ : Weight p σ B} (h : φ.SemiPosReg D) (t : ℝ) : (φ.addFn fun _ => t).SemiPosReg D

theorem **semiPos_fs** {B : LB X} (hB : B.IsLB) {α : Type} [Fintype α] [DecidableEq α] (b : α → B.Sec) (hb : ⊔ j, nonzeroLocus (b j) = ⊤) (A : (k →+* ℂ) → Matrix α α ℂ) (hA : ∀ σ, (A σ).det ≠ 0) (hc : ∀ σ, ∃ U ∈ Matrix.unitaryGroup α ℂ, A ((starRingEnd ℂ).comp σ) = (A σ).map (starRingEnd ℂ) * U) : (fs p hB b hb A hA hc).SemiPos D

theorem **SemiPos.shift** {L : HermLB p} (h : L.SemiPos D) (t : (k →+* ℂ) → ℝ) (ht : ∀ σ, t ((starRingEnd ℂ).comp σ) = t σ) : (L.shift t ht).SemiPos D

theorem **isNef_triv** : (triv p).IsNef D where

theorem **isIntegral_of_isDomain** [IsDomain k] : IsIntegral (PSpace k N)

theorem **isArithVar_int** : IsArithVar (PSpace.str ℤ N) N

theorem **hermitianTwistingSheaf_weight** (σ : k →+* ℂ) : (hermitianTwistingSheaf k N).φ σ = ProjAn.fsO1 k σ N

theorem **isArithVar_id** [IsDomain k] [CharZero k] [Module.Finite ℤ k] (hd : ringKrullDim k = 1) : IsArithVar (𝟙 (Sp k)) 0

theorem **isArithVar_int** : IsArithVar (𝟙 (Sp ℤ)) 0

theorem **finsum_ord_prime** {ℓ : ℕ} (hℓ : ℓ.Prime) : ∑ᶠ z, (Scheme.ord (fnOf ⊤ (Opens.mem_top _) (e ℓ)) z : ℝ) * Real.log (Nat.card (Z.residueField z)) = Real.log ℓ

theorem **product_formula** {q : Z ⟶ Sp ℤ} {σ : ℤ →+* ℂ} (y : CxPts q σ) {f : Z.functionField} (hf : f ≠ 0) : UnitAt (under y) f ∧ ∑ᶠ z, (Scheme.ord f z : ℝ) * Real.log (Nat.card (Z.residueField z)) = Real.log ‖ratVal f y‖

theorem **mem_asIdeal_iff** (n : ℤ) : n ∈ (i (genericPoint Z)).asIdeal ↔ (n : Γ(Z, ⊤)) = 0

theorem **surjective_intCast** [IsClosedImmersion i] : Function.Surjective (Int.castRingHom Γ(Z, ⊤))

theorem **isArithHeight_trivShift** (t : ℝ) : (trivShift t).IsArithHeight fun x => if x.asIdeal = ⊥ then t else Real.log (Nat.card (ℤ ⧸ x.asIdeal)) where

theorem **isSelfInter_trivShift** (t : ℝ) : (trivShift t).IsSelfInter t

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsEffAddNsmulDiv

theorem **shortExact_cokernel_comp** {C : Type*} [Category C] [Abelian C] {A B E : C} (f : A ⟶ B) (g : B ⟶ E) [Mono g] : (kernelCokernelCompSequence.snakeInput f g).L₃.ShortExact where

theorem **secIdeal_canonical** : ThickeningDetermines.secIdeal D.isLB_lb D.canonical = D.idealSheaf

theorem **nonempty_cokernel_iso** {G : X.Modules} [G.IsQuasicoherent] {φ : G ⟶ structureModule X} (h : IsMulBy φ D.canonical) : Nonempty (cokernel φ ≅ divMod D.idealSheaf)

theorem **nonempty_cut_iso** : Nonempty (cut (structureModule X) D.isLB_lb D.canonical ≅ divMod D.idealSheaf)

theorem **support_divMod** : support (divMod D.idealSheaf) = D.idealSheaf.support

theorem **exists_shortExact_add** : ∃ S : ShortComplex X.Modules, S.ShortExact ∧ Nonempty (S.X₁ ≅ twistZ (divMod D.idealSheaf) D'.lb (-1)) ∧ Nonempty (S.X₂ ≅ divMod (D.idealSheaf * D'.idealSheaf)) ∧ Nonempty (S.X₃ ≅ divMod D'.idealSheaf)

theorem **idealSheaf_eq_of_germ** (h : ∀ (x : X) (i : D.ι) (j : D'.ι) (hi : x ∈ D.U i) (hj : x ∈ D'.U j), Associated ((X.presheaf.germ _ x hi).hom (D.f i)) ((X.presheaf.germ _ x hj).hom (D'.f j))) : D.idealSheaf = D'.idealSheaf

theorem **germ_add_f** (x : X) (p : (D.add D').ι) (h : x ∈ (D.add D').U p) : (X.presheaf.germ _ x h).hom ((D.add D').f p) = (X.presheaf.germ _ x h.1).hom (D.f p.1) * (X.presheaf.germ _ x h.2).hom (D'.f p.2)

theorem **idealSheaf_mul_left_cancel** {E D D' : EffCartierData X} (h : E.idealSheaf * D.idealSheaf = E.idealSheaf * D'.idealSheaf) : D.idealSheaf = D'.idealSheaf

theorem **le_basicOpen_of_isUnit** {U V : X.Opens} (h : V ≤ U) {a : Γ(X, U)} (ha : IsUnit (resG X h a)) : V ≤ X.basicOpen a

theorem **topologicalKrullDim_pspace** (N : ℕ) : topologicalKrullDim (PSpace O N) = (N + 1 : ℕ)

theorem **const_eq** (g : O) : algebraMap O Γ(Z.X, ⊤) g = Z.str.appTop ((Scheme.ΓSpecIso (.of O)).inv g)

theorem **exists_isAmpleLB** : ∃ A : LB Z.X, IsAmpleLB A

theorem **mem_gen** {x : Z.X} : x ∈ Z.gen ↔ Z.str x ≠ closedPoint O

theorem **mem_gen_iff_eq_bot** {x : Z.X} : x ∈ Z.gen ↔ (Z.str x).asIdeal = ⊥

theorem **mem_basicOpen_const** {x : Z.X} {g : O} : x ∈ Z.X.basicOpen (Z.const g) ↔ g ∉ (Z.str x).asIdeal

theorem **gen_le_basicOpen** {g : O} (hg : g ≠ 0) : Z.gen ≤ Z.X.basicOpen (Z.const g)

theorem **gen_eq_basicOpen** {g : O} (hg : g ≠ 0) (hm : g ∈ maximalIdeal O) : Z.gen = Z.X.basicOpen (Z.const g)

theorem **isUnit_const** {g : O} (hg : g ≠ 0) {V : Z.X.Opens} (h : V ≤ Z.gen) : IsUnit (resG Z.X (le_top : V ≤ ⊤) (Z.const g))

theorem **isRegularFn_const** {g : O} (hg : g ≠ 0) : IsRegularFn (Z.const g)

theorem **genericPoint_mem_gen** : genericPoint Z.X ∈ Z.gen

theorem **exists_notMem_gen** : ∃ x : Z.X, x ∉ Z.gen

theorem **resG_injective** {V W : Z.X.Opens} (h : W ≤ V) (hW : Z.gen ⊓ V ≤ W) : Function.Injective (resG Z.X h)

theorem **dim_special_add_one_le** : topologicalKrullDim ((Z.gen : Set Z.X)ᶜ : Set Z.X) + 1 ≤ topologicalKrullDim Z.X

theorem **exists_dim_eq** : ∃ d : ℕ, topologicalKrullDim Z.X = (d + 1 : ℕ)

theorem **dim_eq** : topologicalKrullDim Z.X = (Z.relDim + 1 : ℕ)

theorem **relDim_eq_iff** {d : ℕ} : Z.relDim = d ↔ topologicalKrullDim Z.X = (d + 1 : ℕ)

theorem **dim_special_le** : topologicalKrullDim ((Z.gen : Set Z.X)ᶜ : Set Z.X) ≤ Z.relDim

@[simp] theorem **relDim_projSpace** (N : ℕ) : (projSpace O N).relDim = N

theorem **isProper_of_comp_str** : IsProper f

theorem **preimage_gen** : f ⁻¹ᵁ Z.gen = Z₁.gen

theorem **appTop_const** (g : O) : f.appTop (Z.const g) = Z₁.const g

theorem **IsVertical.suppDim_lt** (h : Z.IsVertical F) : suppDim F < (Z.relDim + 1 : ℕ)

theorem **IsVertical.finiteLenTwists** [IsFiniteQC F] (h : Z.IsVertical F) : FiniteLenTwists O F

theorem **IsVertical.diffVanish** [IsFiniteQC F] (h : Z.IsVertical F) : DiffVanish (Z.relDim + 1) (chiLenPic O F)

theorem **IsVertical.deg_prod** [IsFiniteQC F] (h : Z.IsVertical F) {ι : Type} [Fintype ι] {L : ι → LB Z.X} (hL : ∀ i, (L i).IsLB) : h.deg (∏ i, c1 (L i) (hL i)) = if Fintype.card ι = Z.relDim then (interLen O F L : ℚ) else 0

@[simp] theorem **idealSheaf_effData** (I : Z.effVert) : (effData I).idealSheaf = I.1

theorem **le_basicOpen_effData** (I : Z.effVert) (i : (effData I).ι) : Z.gen ⊓ (effData I).U i ≤ Z.X.basicOpen ((effData I).f i)

theorem **isVertical_divMod** (I : Z.effVert) : Z.IsVertical (divMod I.1)

theorem **effCls_eq** {I : Z.effVert} {D : EffCartierData Z.X} (h : D.idealSheaf = I.1) : effCls I = cls _ D.isLB_lb

theorem **effCls_mul** (I J : Z.effVert) : effCls (I * J) = effCls I + effCls J

theorem **effDeg_mul** (I J : Z.effVert) : effDeg (I * J) = effDeg I + effDeg J

theorem **ofEff_mul** (I J : Z.effVert) : ofEff (I * J) = ofEff I + ofEff J

@[simp] theorem **ofEff_one** : ofEff (1 : Z.effVert) = 0

theorem **ofEff_injective** : Function.Injective (ofEff (Z

theorem **exists_eq_ofEff_sub** (D : Z.VDivZ) : ∃ I J, D = ofEff I - ofEff J

@[simp] theorem **vLift_ofEff** {A : Type*} [AddCommGroup A] (f : Z.effVert → A) (h : ∀ I J, f (I ∗ J) = f I + f J) (I : Z.effVert) : vLift f h (ofEff I) = f I

theorem **vDivZ_hom_ext** {A : Type*} [AddCommGroup A] {f g : Z.VDivZ →+ A} (h : ∀ I, f (ofEff I) = g (ofEff I)) : f = g

@[simp] theorem **vCls_ofEff** (I : Z.effVert) : Z.vCls (ofEff I) = effCls I

@[simp] theorem **vDeg_ofEff** (I : Z.effVert) : Z.vDeg (ofEff I) = effDeg I

theorem **vDeg_ofEff_prod** (I : Z.effVert) {ι : Type} [Fintype ι] {L : ι → LB Z.X} (hL : ∀ i, (L i).IsLB) : Z.vDeg (ofEff I) (∏ i, c1 (L i) (hL i)) = if Fintype.card ι = Z.relDim then (interLen 0 (divMod I.1) L : ℚ) else 0

theorem **vDeg_eq_zero** (D : Z.VDivZ) {n : ℕ} (hn : n ≠ Z.relDim) {x : PicSym Z.X} (hx : x ∈ symDeg (SmallPic Z.X) n) : Z.vDeg D x = 0

theorem **coe_effConst** (g : 0) (hg : g ≠ 0) : (Z.effConst g hg).1 = Scheme.IdealSheafData.ofIdealTop (Ideal.span {Z.const g})

theorem **div_mul** {g g' : 0} (hg : g ≠ 0) (hg' : g' ≠ 0) (h : g ∗ g' ≠ 0) : Z.div (g ∗ g') h = Z.div g hg + Z.div g' hg'

theorem **div_of_isUnit** {g : 0} (h : IsUnit g) (hg : g ≠ 0) : Z.div g hg = 0

theorem **div_pow** {g : 0} (hg : g ≠ 0) (n : ℕ) (h : g ^ n ≠ 0) : Z.div (g ^ n) h = n • Z.div g hg

theorem **div_eq_addVal_smul** {ϖ g : 0} (hϖ : Irreducible ϖ) (hg : g ≠ 0) : Z.div g hg = (IsDiscreteValuationRing.addVal 0 g).toNat • Z.div ϖ hϖ.ne_zero

@[simp] theorem **vCls_div** (g : 0) (hg : g ≠ 0) : Z.vCls (Z.div g hg) = 0

theorem **keepsRegular** {I : Z.effVert} {D : EffCartierData Z.X} (h : D.idealSheaf = I.1) : D.KeepsRegular f

theorem **effPull_mul** (I J : Z.effVert) : effPull hf (I ∗ J) = effPull hf I ∗ effPull hf J

@[simp] theorem **vPull_ofEff** (I : Z.effVert) : vPull hf (ofEff I) = ofEff (effPull hf I)

theorem **vCls_vPull** (D : Z.VDivZ) : Z₁.vCls (vPull hf D) = pull f (Z.vCls D)

@[simp] theorem **vPull_div** (g : 0) (hg : g ≠ 0) : vPull hf (Z.div g hg) = Z₁.div g hg

theorem **isUnit_eqn** (i : P.M.ι) {W : Z.X.Opens} (h : W ≤ Z.gen ⊓ P.M.U i) : IsUnit (P.eqn i h)

@[simp] theorem **resG_eqn** (i : P.M.ι) {W W' : Z.X.Opens} (h : W ≤ Z.gen ⊓ P.M.U i) (h' : W' ≤ W) : resG Z.X h' (P.eqn i h) = P.eqn i (h'.trans h)

theorem **eqn_rel** (i j : P.M.ι) {W : Z.X.Opens} (hi : W ≤ Z.gen ⊓ P.M.U i) (hj : W ≤ Z.gen ⊓ P.M.U j) (hij : W ≤ P.M.U i ⊓ P.M.U j) : P.eqn i hi = resG Z.X hij (P.M.g i j : Γ(Z.X, P.M.U i ⊓ P.M.U j)) ∗ P.eqn j hj

theorem **eqn_mul_rel** (i j : P.M.ι) (i' j' : P'.M.ι) {W V : Z.X.Opens} (hi : W ≤ Z.gen ⊓ P.M.U i) (hj : W ≤ Z.gen ⊓ P.M.U j) (hi' : W ≤ Z.gen ⊓ P'.M.U i') (hj' : W ≤ Z.gen ⊓ P'.M.U j') (h : W ≤ V) (hij : V ≤ P.M.U i ⊓ P.M.U j) (hij' : V ≤ P'.M.U i' ⊓ P'.M.U j') : P.eqn i hi ∗ P'.eqn i' hi' = resG Z.X h (res hij (P.M.g i j) ∗ res hij' (P'.M.g i' j')).val ∗ (P.eqn j hj ∗ P'.eqn j' hj')

theorem **Extends.resG_f** (h : P.Extends g s) (i : P.M.ι) {W : Z.X.Opens} (hW : W ≤ Z.gen ⊓ P.M.U i) : resG Z.X (hW.trans inf_le_right) (s.f i) = resG Z.X le_top (Z.const g) ∗ P.eqn i hW

theorem **Extends.le_basicOpen** (h : P.Extends g s) (hg : g ≠ 0) (i : P.M.ι) : Z.gen ⊓ P.M.U i ≤ Z.X.basicOpen (s.f i)

theorem **Extends.isRegularSec** (h : P.Extends g s) (hg : g ≠ 0) : IsRegularSec s

theorem **Extends.smul** (h : P.Extends g s) (g' : 0) : P.Extends (g' ∗ g) (Z.const g' • s)

theorem **exists_extends** {a : 0} (ha : a ≠ 0) (hm : a ∈ maximalIdeal 0) : ∃ (m : ℕ) (s : P.M.Sec), P.Extends (a ^ m) s

theorem **IsSum.ofEff_sub** (H : IsSum P P' P'') (h : P.Extends g s) (h' : P'.Extends g' s') (h'' : P''.Extends g'' s'') (hg : g ≠ 0) (hg' : g' ≠ 0) (hg'' : g'' ≠ 0) : Model.ofEff (h''.eff hg'') − Z.div g'' hg'' = (Model.ofEff (h.eff hg) − Z.div g hg) + (Model.ofEff (h'.eff hg') − Z.div g' hg')

theorem **exists_extends'** : ∃ q : {g : 0 // g ≠ 0} × P.M.Sec, P.Extends q.1 q.2

theorem **IsSum.cls_eq** (H : IsSum P P' P'') : P''.cls = P.cls + P'.cls

theorem **IsIsom.isSum** (H : IsIsom P P') : IsSum P (zero Z) P'

theorem **IsIsom.refl** : IsIsom P P

@[simp] theorem **cls_zero** : (zero Z).cls = 0

theorem **Extends.cls_eq** (h : P.Extends g s) (hg : g ≠ 0) : P.cls = Model.ofEff (h.eff hg) − Z.div g hg

theorem **IsIsom.cls_eq** (H : IsIsom P P') : P.cls = P'.cls

@[simp] theorem **cls_add** : (P.add P').cls = P.cls + P'.cls

@[simp] theorem **cls_neg** : P.neg.cls = −P.cls

theorem **extends_one** (h : ofSec s Z.gen = P.t) : P.Extends 1 s

theorem **Extends.cls_eq_ofEff** (h : P.Extends 1 s) : P.cls = Model.ofEff (h.eff one_ne_zero)

@[simp] theorem **cls_ofEff** (I : Z.effVert) : (ofEff I).cls = Model.ofEff I

@[simp] theorem **cls_const** (g : 0) (hg : g ≠ 0) : (const Z g hg).cls = Z.div g hg

@[simp] theorem **vCls_cls** : Z.vCls P.cls = CechPic.cls P.M P.isLB

theorem **cls_surjective** : Function.Surjective (cls (Z

theorem **cls_eq_iff** : P.cls = P'.cls ↔ IsIsom P P'

theorem **Extends.pull** (h : P.Extends g s) : (P.pull hf).Extends g (Cartier.secPull s f)

@[simp] theorem **cls_pull** : (P.pull hf).cls = vPull hf P.cls

theorem **exists_isEff_add_nsmul_div** (D : Z.VDivZ) {a : 0} (ha : a ≠ 0) (hm : a ∈ maximalIdeal 0) : ∃ m : ℕ, IsEff (D + m • Z.div a ha)

theorem **vDeg_prod_of_add_nsmul_div** {D : Z.VDivZ} {a : 0} (ha : a ≠ 0) {m : ℕ} {E : Z.effVert} (h : ofEff E = D + m • Z.div a ha) {L : Fin Z.relDim → LB Z.X} (hL : ∀ i, (L i).IsLB) : Z.vDeg D (∏ i, c1 (L i) (hL i)) = interLen 0 (divMod E.1) L − m ∗ interLen 0 (divMod (Z.effConst a ha).1) L

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsFlagBundleRankOne

theorem **isFlagBundle_empty** (B : Scheme.{0}) (E : Empty → B.Modules) (n : Empty → ℕ) (fl : ∀ a, Vert.Flag ((Scheme.Modules.pullback (𝟙 B)).obj (E a)) (n a)) : IsFlagBundle (𝟙 B) E n fl

theorem **isFlagBundle_rank_one** (B : Scheme.{0}) {ι : Type} (E : ι → B.Modules) (fl : ∀ a, Vert.Flag ((Scheme.Modules.pullback (𝟙 B)).obj (E a)) 1) : IsFlagBundle (𝟙 B) E (fun _ => 1) fl

theorem **exists_isFlagBundle_rank_one** (B : Scheme.{0}) {ι : Type} (E : ι → B.Modules) (hE : ∀ a, IsVB (E a) 1) : ∃ fl, IsFlagBundle (𝟙 B) E (fun _ => 1) fl

theorem **bcFl_eq** : bcFl Y = -(((Y.j : ℝ) - 1) / Y.j) * testI Y + -(((Y.N : ℝ) - Y.j - 1) / ((Y.N : ℝ) - Y.j)) * -testI Y

theorem **bcFl_of_eq_two_mul** (h : Y.N = 2 * Y.j) : bcFl Y = 0

theorem **HhTest.vol_mul** {hh : Fn Y} (h : HhTest Y hh) (r s : ℝ) : Y.vol * bcFl Y - 2 * r * Y.vol * testI Y + s * (2 * Y.vol * testI Y) = Y.vol * (hh (testF Y) - 2 * testI Y * (r - 1 / 2 - s))

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsGHeckeMapForget

theorem **GBase.isMap_map** (β : R₁ →+* R₂) (gb : GBase G R₁) : GBase.IsMap β gb (gb.map β)

theorem **GBase.isMap_iff** {β : R₁ →+* R₂} {gb₁ : GBase G R₁} {gb₂ : GBase G R₂} : GBase.IsMap β gb₁ gb₂ ↔ gb₂ = gb₁.map β

theorem **GBase.IsMap.id** (gb : GBase G R₁) : GBase.IsMap (RingHom.id R₁) gb gb

theorem **IsGGPullOver.over** (h : IsGGPullOver β o' o f) : f ≫ s = s' ≫ spMap β

theorem **IsGGPullOver.isMap** (h : IsGGPullOver β o' o f) : GBase.IsMap β gb₁ gb₂

theorem **IsGGPullOver.pull** (h : IsGGPullOver β o' o f) : IsGGPull o' o f

theorem **isGGPull_iff** {gb : GBase G R} {N : ℕ} {s' : S' ⟶ Sp R} {s : S ⟶ Sp R} {o' : GObj gb N s'} {o : GObj gb N s} {f : S' ⟶ S} : IsGGPull o' o f ↔ Nonempty (GObjMap o' o f)

theorem **isGGPullOver_id_iff** {gb : GBase G R} {N : ℕ} {s' : S' ⟶ Sp R} {s : S ⟶ Sp R} {o' : GObj gb N s'} {o : GObj gb N s} {f : S' ⟶ S} : IsGGPullOver (RingHom.id R) o' o f ↔ f ≫ s = s' ∧ Nonempty (GObjMap o' o f)

theorem **IsGGPull.refl** {gb : GBase G R} {N : ℕ} {s : S ⟶ Sp R} (o : GObj gb N s) : IsGGPull o o (𝟙 S)

theorem **GBase.IsMap.comp** {β : R₁ →+* R₂} {β' : R₂ →+* R₃} (h' : GBase.IsMap β' gb₂ gb₃) (h : GBase.IsMap β gb₁ gb₂) : GBase.IsMap (β'.comp β) gb₁ gb₃

theorem **IsGGPull.comp** (h₁ : IsGGPull o₃ o₂ f) (h₂ : IsGGPull o₂ o₁ f') : IsGGPull o₃ o₁ (f ≫ f')

theorem **IsGGPullOver.comp** {β : R₁ →+* R₂} {β' : R₂ →+* R₃} (h₁ : IsGGPullOver β' o₃ o₂ f) (h₂ : IsGGPullOver β o₂ o₁ f') : IsGGPullOver (β'.comp β) o₃ o₁ (f ≫ f')

theorem **GObj.eta_zero** {gb : GBase G R} {M : ℕ} {s : S ⟶ Sp R} (o : GObj gb M s) : o.eta 0 = 1

theorem **GObj.eta_neg** {gb : GBase G R} {M : ℕ} {s : S ⟶ Sp R} (o : GObj gb M s) (x : G.La) : o.eta (-x) = (o.eta x)⁻¹

theorem **GObj.eta_nsmul** {gb : GBase G R} {M : ℕ} {s : S ⟶ Sp R} (o : GObj gb M s) (n : ℕ) (x : G.La) : o.eta (n • x) = o.eta x ^ n

theorem **mem_gHeckeSub** {gb : GBase G R} {M : ℕ} {s : S ⟶ Sp R} (o : GObj gb M s) (g : G.La →ₗ[G.O] G.La) (N : ℕ) (x : G.La) : o.eta (N • g x) ∈ gHeckeSub o g N

theorem **IsGHecke.isMonHom** (h : IsGHecke g c k N o o' ψ) : IsMonHom ψ

theorem **IsGHecke.act** (h : IsGHecke g c k N o o' ψ) (c' : G.O) : o.act.act c' ≫ ψ = ψ ≫ o'.act.act c'

theorem **IsGHecke.eta** (h : IsGHecke g c k N o o' ψ) (x : G.La) : o'.eta x = o.eta (g x) ≫ ψ

theorem **IsGHecke.cover** (h : IsGHecke g c k N o o' ψ) : GlobalTheory.ZM.IsGaloisCover (gHeckeSub o g N) (fun h => (Hecke.transl h.1).left) ψ.left

theorem **IsGHecke.pol** (h : IsGHecke g c k N o o' ψ) : Nonempty (LBIso (Hecke.lbPow o.P (k ^ 2 / c)) (o'.P.pull (Hecke.sq2 ψ)))

theorem **IsGHecke.dvd** (h : IsGHecke g c k N o o' ψ) : c ∣ k ^ 2

theorem **IsGHecke.mul_exp** (h : IsGHecke g c k N o o' ψ) : c * (k ^ 2 / c) = k ^ 2

theorem **IsGHecke.kills** (h : IsGHecke g c k N o o' ψ) (x : G.La) : o.eta (N • g x) ≫ ψ = 1

theorem **IsGHeckeMap.over** (h : IsGHeckeMap g c k N X X' π) : π ≫ X.s = X'.s

theorem **IsGHeckeMap.dvd** (h : IsGHeckeMap g c k N X X' π) : c ∣ k ^ 2

theorem **IsGHecke.exists_isGHeckeMap** (X : GenModel gb N) {o₁ : GObj gb N X'.s} {ψ₁ : X'.univ.A ⟶ o₁.A} (h : IsGHecke g c k N X'.univ o₁ ψ₁) : ∃ π : X'.X ⟶ X.X, IsGHeckeMap g c k N X X' π

theorem **gHeckeSub_smul_id_eq_one** {gb : GBase G R} {N k : ℕ} {s : S ⟶ Sp R} (o : GObj gb (N * k) s) : ∀ h ∈ gHeckeSub o (k • LinearMap.id) N, h = 1

theorem **isGHecke_forgetMul** {gb : GBase G R} {N k : ℕ} (hN : 0 < N) (hk : 0 < k) {s : S ⟶ Sp R} (o : GObj gb (N * k) s) : IsGHecke (k • LinearMap.id) (k ^ 2) k N o (o.forgetMul hN hk) (𝟙 o.A)

theorem **exists_isGHeckeMap_forget** {gb : GBase G R} {N k : ℕ} (hN : 0 < N) (hk : 0 < k) (X : GenModel gb N) (X' : GenModel gb (N * k)) : ∃ π : X'.X ⟶ X.X, IsGHeckeMap (k • LinearMap.id) (k ^ 2) k N X X' π

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsHoloDelta

theorem **exists_isHolo_delta** : ∃ φ : TR.Sm ℚ 2, φ ≠ 0 ∧ IsHolo lamDelta φ ∧ ∀ u ∈ K1G 1, TR.finRep ℚ 2 u φ = φ

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsLevWithGeometric

theorem **exists_sep** (l : Level H) : ∃ e : 𝒪 E, e ∈ Mfr l ∧ 1 - e ∈ l.Mb ∧ H.bar e ∈ l.Mb ∧ H.bar (1 - e) ∈ Mfr l

theorem **smul_modN** {t : 𝒪 E} (ht : t ∈ l.Mb) {a : Dw.La} (ha : a ∈ (Mfr l • ⊤ : Submodule (𝒪 E) Dw.La)) : Dw.ModN l.M (t • a)

theorem **modN_of_dvd** {k m : ℕ} (h : k ∣ m) {a : Dw.La} (ha : Dw.ModN m a) : Dw.ModN k a

theorem **modN_of_forall_dvd** {k : ℕ} (hk : 0 < k) {a : Dw.La} (h : ∀ y, (k : ℤ) ∣ Dw.EF a y) : Dw.ModN k a

theorem **dvd_EF_sub** {k : ℕ} {a a' b b' : Dw.La} (ha : Dw.ModN k (a - a')) (hb : Dw.ModN k (b - b')) : (k : ℤ) ∣ Dw.EF a b - Dw.EF a' b'

theorem **zpow_congr** {G : Type} [CommGroup G] {ζ : G} {k : ℕ} (h : ζ ^ k = 1) {a b : ℤ} (hab : (k : ℤ) ∣ a - b) : ζ ^ a = ζ ^ b

theorem **unitOn_bijective** (R : Type) [CommRing R] : Function.Bijective (unitOn (R

theorem **isPrimitiveRoot_zetaM** : IsPrimitiveRoot (zetaM c) l.M

theorem **map_rootOn** (k : ℕ) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) : Units.map ψ.toRingHom.toMonoidHom (rootOn c k S) = rootOn c k T

theorem **isPrimitiveRoot_rootOn** {k : ℕ} (hk : k ∣ l.M) {S : Type} [CommRing S] [Algebra c.F S] (K : Type) [Field K] (χ : S →+* K) : IsPrimitiveRoot (Units.map χ.toMonoidHom (rootOn c k S)) k

theorem **rootOn_pow** {k : ℕ} (hk : k ∣ l.M) (S : Type) [CommRing S] [Algebra c.F S] : rootOn c k S ^ k = 1

theorem **levPt_add** {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) (a b : Dw.La) : levPt Xo x0 (a + b) = levPt Xo x0 a ∗ levPt Xo x0 b

theorem **levPt_pow** {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) (a : Dw.La) : levPt Xo x0 a ^ l.M = 1

theorem **halfPt_pow** {k : ℕ} (hk : k ∣ l.M) {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) (a : Dw.La) : halfPt Xo k x0 a ^ k = 1

theorem **halfPt_congr** {k : ℕ} (hk : k ∣ l.M) {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) {a b : Dw.La} (h : Dw.ModN k (a − b)) : halfPt Xo k x0 a = halfPt Xo k x0 b

theorem **comp_halfPt** (k : ℕ) {S T : Type} [CommRing S] [CommRing T] {x0 : Sp S ⟶ Xo.X} {x1 : Sp T ⟶ Xo.X} (h : ptO Xo x1 ⟶ ptO Xo x0) (a : Dw.La) : h ≫ halfPt Xo k x0 a = halfPt Xo k x1 a

theorem **weilIs_baseChange** {k : ℕ} {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) {x0 : Sp S ⟶ Xo.X} {x1 : Sp T ⟶ Xo.X} (h : ptO Xo x1 ⟶ ptO Xo x0) (hh : h.left = spMap ψ) {p q : ptO Xo x0 ⟶ Xo.univ.t.G} {u : Sˣ} (hu : WeilIs Xo.univ.t.G Xo.univ.t.PP.L k p q (unitOn u)) : WeilIs Xo.univ.t.G Xo.univ.t.PP.L k (h ≫ p) (h ≫ q) (unitOn (Units.map ψ.toMonoidHom u))

theorem **weilIs_unique** {k : ℕ} {S : Type} [CommRing S] {x0 : Sp S ⟶ Xo.X} {p q : ptO Xo x0 ⟶ Xo.univ.t.G} {u v : Sˣ} (hu : WeilIs Xo.univ.t.G Xo.univ.t.PP.L k p q (unitOn u)) (hv : WeilIs Xo.univ.t.G Xo.univ.t.PP.L k p q (unitOn v)) : u = v

theorem **IsLevWith.baseChange** {k : ℕ} {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) {x0 : Sp S ⟶ Xo.X} {x1 : Sp T ⟶ Xo.X} (h : ptO Xo x1 ⟶ ptO Xo x0) (hh : h.left = spMap ψ) {ζ : Sˣ} {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (hev : IsLevWith Xo k ζ x0 ev) : IsLevWith Xo k (Units.map ψ.toMonoidHom ζ) x1 (fun q => h ≫ ev q)

theorem **IsLevWith.unique_field** {k : ℕ} (hk : k ∣ l.M) {K : Type} [Field K] {x0 : Sp K ⟶ Xo.X} {ζ : Kˣ} (hζ : IsPrimitiveRoot ζ k) {ev ev' : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (h : IsLevWith Xo k ζ x0 ev) (h' : IsLevWith Xo k ζ x0 ev') : ev = ev'

theorem **IsLevOf.unique** {k : ℕ} {S : Type} [CommRing S] [Algebra c.F S] {x0 : Sp S ⟶ Xo.X} {ev ev' : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (h : IsLevOf Xo c k x0 ev) (h' : IsLevOf Xo c k x0 ev') : ev = ev'

theorem **exists_isLevWith_geometric** {k : ℕ} (hk : k ∣ l.M) {K : Type} [Field K] [IsAlgClosed K] (x0 : Sp K ⟶ Xo.X) {ζ : Kˣ} (hζ : IsPrimitiveRoot ζ k) : ∃ ev, IsLevWith Xo k ζ x0 ev

theorem **exists_isLevOf_geometric** {k : ℕ} (hk : k ∣ l.M) {K : Type} [Field K] [IsAlgClosed K] [Algebra c.F K] (x0 : Sp K ⟶ Xo.X) : ∃ ev, IsLevOf Xo c k x0 ev

theorem **exists_isLevWith_cover** {k : ℕ} (hk : k ∣ l.M) {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) {ζ : Sˣ} (hkS : IsUnit (k : S)) (hζ1 : ζ ^ k = 1) (hζ : ∀ (K : Type) [Field K] (χ : S →+* K), IsPrimitiveRoot (Units.map χ.toMonoidHom ζ) k) : ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (ev : LatMod Dw k → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ IsLevWith Xo k (Units.map φ.toMonoidHom ζ) (spMap φ ≫ x0) ev

theorem **IsLevOf.baseChange** {k : ℕ} {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) {x0 : Sp S ⟶ Xo.X} {x1 : Sp T ⟶ Xo.X} (h : ptO Xo x1 ⟶ ptO Xo x0) (hh : h.left = spMap ψ.toRingHom) {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (hev : IsLevOf Xo c k x0 ev) : IsLevOf Xo c k x1 (fun q => h ≫ ev q)

theorem **exists_isLevOf_cover** {k : ℕ} (hk : k ∣ l.M) {S : Type} [CommRing S] [Algebra c.F S] (x0 : Sp S ⟶ Xo.X) : ∃ (B : Type) (_ : CommRing B) (_ : Algebra c.F B) (φ : S →ₐ[c.F] B) (ev : LatMod Dw k → (ptO Xo (spMap φ.toRingHom ≫ x0) ⟶ Xo.univ.t.G)), φ.toRingHom.FaithfullyFlat ∧ IsLevOf Xo c k (spMap φ.toRingHom ≫ x0) ev

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsLocModelIsPrimeOf

@[simp] theorem **ordQ_single** (g : K) (t : ℚ) : E.ordQ (Finsupp.single g t) = t ∗ E.ord g

theorem **ext** {E F : Place c} (h : E.ring = F.ring) : E = F

@[simp] theorem **ord_one** : E.ord 1 = 0

theorem **ord_mul** {f g : K} (hf : f ≠ 0) (hg : g ≠ 0) : E.ord (f ∗ g) = E.ord f + E.ord g

theorem **ord_inv** (f : K) : E.ord f⁻¹ = −E.ord f

theorem **ord_zpow** (f : K) (m : ℤ) : E.ord (f ^ m) = m ∗ E.ord f

theorem **ord_pow** (f : K) (m : ℕ) : E.ord (f ^ m) = m ∗ E.ord f

theorem **ord_nonneg_iff** {f : K} (hf : f ≠ 0) : 0 ≤ E.ord f ↔ f ∈ E.ring

theorem **ord_eq_zero** {u : E.ring} (hu : IsUnit u) : E.ord (u : K) = 0

theorem **ord_irreducible** {t : E.ring} (ht : Irreducible t) : E.ord (t : K) = 1

theorem **ring_ne_top** : E.ring ≠ ⊤

theorem **Contains.mono** {E : Place c} {R R' : Subring K} (h : E.Contains R') (hR : R ≤ R') : E.Contains R

theorem **contains_ring_iff** (F : Place c) : F.Contains E.ring.toSubring ↔ F = E

theorem **isPrimeOf_self** : E.IsPrimeOf E.ring.toSubring

theorem **isPrimeOf_ring_iff** (F : Place c) : F.IsPrimeOf E.ring.toSubring ↔ F = E

theorem **isFreeGen_toRat** {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) : IsFreeGen K (E.toRat n ω)

theorem **toRat_ne_zero** {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) : E.toRat n ω ≠ 0

theorem **IsFormOrd.ne_zero** {η : Forms c n} {a : ℤ} (h : E.IsFormOrd η a) : η ≠ 0

theorem **IsFormOrd.unique** {η : Forms c n} {a b : ℤ} (h : E.IsFormOrd η a) (h' : E.IsFormOrd η b) : a = b

theorem **IsFormOrd.formOrd_eq** {η : Forms c n} {a : ℤ} (h : E.IsFormOrd η a) : E.formOrd η = a

theorem **formOrd_smul_toRat** {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) {f : K} (hf : f ≠ 0) : E.formOrd (f • E.toRat n ω) = E.ord f

theorem **formOrd_toRat** {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) : E.formOrd (E.toRat n ω) = 0

theorem **Klt.form_ne_zero** (h : Klt c n R η φ) : η ≠ 0

theorem **Klt.apply_zero** (h : Klt c n R η φ) : φ 0 = 0

theorem **Klt.lt_disc** (h : Klt c n R η φ) {E : Place c} (hE : E.Contains R) : −1 < E.disc η φ

theorem **Klt.mono** (h : Klt c n R η φ) (hR : R ≤ R') : Klt c n R' η φ

theorem **EffAt.form_ne_zero** (h : EffAt c n R η φ) : η ≠ 0

theorem **EffAt.apply_zero** (h : EffAt c n R η φ) : φ 0 = 0

theorem **EffAt.disc_le** (h : EffAt c n R η φ) {E : Place c} (hE : E.IsPrimeOf R) : E.disc η φ ≤ 0

theorem **klt_ring_iff** (E : Place c) (hη : η ≠ 0) (hφ : φ 0 = 0) : Klt c n E.ring.toSubring η φ ↔ -1 < E.disc η φ

theorem **effAt_ring_iff** (E : Place c) (hη : η ≠ 0) (hφ : φ 0 = 0) : EffAt c n E.ring.toSubring η φ ↔ E.disc η φ ≤ 0

theorem **klt_uniformiser_iff** (E : Place c) {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) {t : E.ring} (ht : Irreducible t) (s : ℚ) : Klt c n E.ring.toSubring (E.toRat n ω) (Finsupp.single (t : K) s) ↔ s < 1

theorem **effAt_uniformiser_iff** (E : Place c) {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) {t : E.ring} (ht : Irreducible t) (s : ℚ) : EffAt c n E.ring.toSubring (E.toRat n ω) (Finsupp.single (t : K) s) ↔ 0 ≤ s

theorem **isLocModel_self** : IsLocModel (RingHom.id k) ⊤ where

theorem **isLocModel** : IsLocModel c E.ring.toSubring where

theorem **exists_isFreeGen** [PerfectField k] {x : Fin n → K} (hx : TrBasis c x) : ∃ ω : Forms E.const n, IsFreeGen E.ring ω

theorem **isFormOrd_formOrd** [PerfectField k] {x : Fin n → K} (hx : TrBasis c x) {η : Forms c n} (hη : η ≠ 0) : E.IsFormOrd η (E.formOrd η)

theorem **formOrd_smul** [PerfectField k] {x : Fin n → K} (hx : TrBasis c x) {η : Forms c n} (hη : η ≠ 0) {f : K} (hf : f ≠ 0) : E.formOrd (f • η) = E.ord f + E.formOrd η

theorem **mem_locRing** {x : X} {f : X.functionField} : f ∈ locRing x ↔ RegAt x f

theorem **ord_placeEquiv** (E : DivOver p) (f : X.functionField^×) : (placeEquiv p E).ord f = E.ord f

theorem **formOrd_placeEquiv** (E : DivOver p) (η : RatForms p n) : (placeEquiv p E).formOrd η = E.formOrd η

theorem **contains_placeEquiv** (E : DivOver p) (x : X) : (placeEquiv p E).Contains (locRing x) ↔ E.Through x

theorem **disc_placeEquiv** (E : DivOver p) (η : RatForms p n) (m : ℕ) (g : X.functionField^×) : (placeEquiv p E).disc η (Finsupp.single (g : X.functionField) (1 / (m : ℚ))) = discrep E η m g

theorem **nonempty_place_affineLine** : Nonempty (Place (Cartier.constFn (spMap (algebraMap k (Polynomial k)))))

theorem **isLocModel_locRing** [LocallyOfFiniteType p] (z : X) : IsLocModel (Cartier.constFn p) (locRing z)

theorem **kltAt_iff_exists_klt** (Δ : X → ℚ) (x : X) : KltAt p n Δ x ↔ ∃ (η : RatForms p n) (m : ℕ) (g : X.functionField^×), 0 < m ∧ IsLogEqn p Δ η x m g ∧ Klt (Cartier.constFn p) n (locRing x) η (Finsupp.single (g : X.functionField) (1 / (m : ℚ)))

theorem **kltAt_iff_klt** [PerfectField k] [LocallyOfFiniteType p] [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] (hn : topologicalKrullDim X = n) {Δ : X → ℚ} {x : X} {η : RatForms p n} {m : ℕ} {g : X.functionField^×} (hη : η ≠ 0) (hm : 0 < m) (h : IsLogEqn p Δ η x m g) : KltAt p n Δ x ↔ Klt (Cartier.constFn p) n (locRing x) η (Finsupp.single (g : X.functionField) (1 / (m : ℚ)))

theorem **exists_isLocModel_isPrimeOf** : ∃ (R : Subring _) (E : Place (Cartier.constFn (spMap (algebraMap k (Polynomial k))))), IsLocModel (Cartier.constFn (spMap (algebraMap k (Polynomial k)))) R ∧ E.IsPrimeOf R

theorem **exists_threshold_affineLine** [PerfectField k] : ∃ (R : Subring (Sp (Polynomial k)).functionField) (η : Forms (Cartier.constFn (spMap (algebraMap k (Polynomial k)))) 1) (t : (Sp (Polynomial k)).functionField), IsLocModel (Cartier.constFn (spMap (algebraMap k (Polynomial k)))) R ∧ ∀ s : ℚ, (Klt _ 1 R η (Finsupp.single t s) ↔ s < 1) ∧ (EffAt _ 1 R η (Finsupp.single t s) ↔ 0 ≤ s)

theorem **trBasis_fractionRing** (n : ℕ) : TrBasis (algebraMap k (FractionRing (MvPolynomial (Fin n) k))) fun i => algebraMap (MvPolynomial (Fin n) k) _ (MvPolynomial.X i)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsLogMult

theorem **isEqn_pullCart** (s : Y ⇢ X) [IsDominant s] {D : CartierDiv X} {y : Y} {g : X.functionField^×} (h : D.IsEqn (s y) g) : (pullCart s D).IsEqn y (fnPull s g)

theorem **toPic_pullCart** (s : Y ⇢ X) [IsDominant s] (D : CartierDiv X) : toPic (pullCart s D) = CechPic.pull s (toPic D)

theorem **pullCart_add** (s : Y ⇢ X) [IsDominant s] (D E : CartierDiv X) : pullCart s (D + E) = pullCart s D + pullCart s E

theorem **IsLogCanonicalMultiple.exists_isLogMult** {p : X ⇢ Sp k} {n : ℕ} {Δ : X → ℚ} {m : ℕ} {G : CartierDiv X} (h : IsLogCanonicalMultiple p n Δ m G) (hm : 0 < m) : ∃ η, IsLogMult p n Δ η m G

theorem **IsMultIdeal.unique** {p : X ⇢ Sp k} {n : ℕ} {η : RatForms p n} {m : ℕ} {G : CartierDiv X} {J J' : X.IdealSheafData} (h : IsMultIdeal p n η m G J) (h' : IsMultIdeal p n η m G J') : J = J'

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsOb1NeZero

theorem **one_hom_eq** {R₀ : Type} [CommRing R₀] (A : Over (Sp R₀)) [GrpObj A] : lift (𝟙 A) (1 : A ⟶ A) = lift (𝟙 A) (toUnit A ≫ η[A])

theorem **mul_eq_zero_of_mem** {a b : R} (ha : a ∈ I) (hb : b ∈ I) : a * b = 0

theorem **smul_idl** (r : R) (x : Idl hI hs hk) : π r • x = r • x

theorem **sq_of_aug** (α : Aug I π R₁) {a b : R} (ha : a ∈ I) (hb : b ∈ I) : a * b = 0

theorem **exists_isSig** (α : Aug I π R₁) {B B₀ : Type} [CommRing B] [Algebra R B] [CommRing B₀] [Algebra R₀ B₀] (p : B →+* B₀) (semi : SemiHom π p) (surj : Function.Surjective p) (ker : ∀ b, p b = 0 ↔ b ∈ I.map (algebraMap R B)) : ∃ σ, α.IsSig p σ

theorem **isOb2_zero** (L : Pre I π F) (h : L.IsFun) (α : Aug I π R₁) (w : Glob1 F) : L.IsOb2 α w 0

theorem **isOb1_zero** (L : Pre I π F) (L' : Pre I π F') (n : CoefMap (RingHom.id R₀) F F') (l : ∀ T : C', L.B (n.G.obj T) →ₐ[R] L'.B T) (h : L.IsNat L' n l) (α : Aug I π R₁) (w : Glob1 F) : L.IsOb1 L' n l α w 0

theorem **q_sec** (α : Aug I π R₁) (i : I) : α.q (sec α i) = 0

theorem **k_sec** (α : Aug I π R₁) (i : I) : α.k (sec α i) = i

theorem **sec_smul** (α : Aug I π R₁) (r : R) (i : I) : sec α (r • i) = π r • sec α i

theorem **sig_inj** (α : Aug I π R₁) {B B₀ : Type} [CommRing B] [Algebra R B] [Module.Flat R B] [CommRing B₀] [Algebra R₀ B₀] (p : B →+* B₀) (semi : SemiHom π p) (surj : Function.Surjective p) {σ : R₁ ⊗[R₀] B₀ →+ B} (hσ : α.IsSig p σ) {y : R₁ ⊗[R₀] B₀} (hy : α.KerQ y) (h0 : σ y = 0) : y = 0

theorem **sig_unique** (α : Aug I π R₁) {B B₀ : Type} [CommRing B] [Algebra R B] [CommRing B₀] [Algebra R₀ B₀] (p : B →+* B₀) (surj : Function.Surjective p) {σ σ' : R₁ ⊗[R₀] B₀ →+ B} (hσ : α.IsSig p σ) (hσ' : α.IsSig p σ') (y : R₁ ⊗[R₀] B₀) : σ y = σ' y

theorem **theta_unique** {B B₀ B' : Type} [CommRing B] [Algebra R B] [CommRing B₀] [Algebra R₀ B₀] [CommRing B'] [Algebra R B'] {p : B →+* B₀} (surj : Function.Surjective p) {α β : B →ₐ[R] B'} {θ θ' : Ω[B⁄R₀] →+ B'} (h : IsTheta I p α β θ) (h' : IsTheta I p α β θ') (w :

Ω[B₀⁄R₀]) : θ w = θ' w

theorem **isOb2_unique** (L : Pre I π F) (α : Aug I π R₁) (w : Glob1 F) {c c' : CatCoh.H2 (ext R₁ F)} (h : L.IsOb2 α w c) (h' : L.IsOb2 α w c') : c = c'

theorem **isOb1_unique** (L : Pre I π F) (L' : Pre I π F') (n : CoefMap (RingHom.id R₀) F F') (l : ∀ T : C', L.B (n.G.obj T) →ₐ[R] L'.B T) (α : Aug I π R₁) (w : Glob1 F) {c c' : CatCoh.H1 (ext R₁ F')} (h : L.IsOb1 L' n l α w c) (h' : L.IsOb1 L' n l α w c') : c = c'

theorem **sh_comp** (i j : I) : (sh j).comp (sh i) = sh (i + j)

theorem **sh_apply** (i : I) (b : R[X]) : sh i b = b + (i : R) • derivative b

theorem **map_sh** (i : I) (b : R[X]) : mapRingHom π (sh i b) = mapRingHom π b

theorem **shPre_isFun_iff** (c : ∀ T T' : C, (T ⟶ T') → I) : (shPre hs hk c).IsFun ↔ ∀ ⦃T T' T'' : C⦄ (a : T ⟶ T') (b : T' ⟶ T''), c _ _ (a ≫ b) = c _ _ a + c _ _ b

theorem **not_isFun** {i : I} (hi : i ≠ 0) : ¬ (shPre (C

theorem **isTheta_th** (α : Aug I π R₁) {σ : R₁ ⊗[R₀] R₀[X] →+ R[X]} (hσ : α.IsSig (mapRingHom π) σ) (i j : I) : IsTheta I (mapRingHom π) (sh i) (sh j) (th α σ (i − j))

theorem **exists_isOb2** (c : ∀ T T' : C, (T ⟶ T') → I) (α : Aug I π R₁) : ∃ z, (shPre hs hk c).IsOb2 α (dX R₀ C) z

theorem **exists_isOb1_ne_zero** {i : I} (hi : i ≠ 0) (α : Aug I π R₁) : ∃ (L L' : Pre I π (lineF R₀ (SingleObj (Multiplicative I)))) (l : ∀ T, L.B T →ₐ[R] L'.B T) (z : CatCoh.H1 _), L.IsFun ∧ L'.IsFun ∧ L.IsLift L' (CoefMap.id _) l ∧ L.IsOb1 L' (CoefMap.id _) l α (dX R₀ _) z ∧ z ≠ 0

theorem **four_two** : ∃ (I : Ideal (ZMod 4)) (π : ZMod 4 →+* ZMod 2), I ^ 2 = 0 ∧ Function.Surjective π ∧ RingHom.ker π = I ∧ ∃ i : I, i ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsObUNeZero

theorem **sig_mem** (L : Pre I π F) (α : Aug I π R₁) (T : C) {σ : R₁ ⊗[R₀] F.obj T →+ L.B T} (hσ : α.IsSig (L.p T) σ) (y : R₁ ⊗[R₀] F.obj T) : σ y ∈ I.map (algebraMap R (L.B T))

theorem **sig_nat** (L : Pre I π F) (α : Aug I π R₁) {σ : ∀ T, R₁ ⊗[R₀] F.obj T →+ L.B T} (hσ : ∀ T, α.IsSig (L.p T) (σ T)) {T T' : C} (a : T ⟶ T') (y : R₁ ⊗[R₀] F.obj T) : σ T' (extHom R₁ (mp F a) y) = L.ρ a (σ T y)

theorem **kerQ_nat** (α : Aug I π R₁) {T T' : C} (a : T ⟶ T') {y : R₁ ⊗[R₀] F.obj T} (hy : α.KerQ y) : α.KerQ (extHom R₁ (mp F a) y)

theorem **sig_surj** (L : Pre I π F) (α : Aug I π R₁) (T : C) {σ : R₁ ⊗[R₀] F.obj T →+ L.B T} (hσ : α.IsSig (L.p T) σ) {x : L.B T} (hx : x ∈ I.map (algebraMap R (L.B T))) : ∃ y, α.KerQ y ∧ σ y = x

theorem **lift_of_isObU_zero** (L : Pre I π F) (α : Aug I π R₁) {u : ∀ (T T' : C) (_ : T ⟶ T'), (F.obj T')ˣ} (h : IsObU L α u 0) : ∃ v : ∀ (T T' : C) (_ : T ⟶ T'), L.B T', (∀ T T' a, L.ρ T' (v T T' a) = (u T T' a : F.obj T')) ∧ ∀ (T T' T'' : C) (a : T ⟶ T') (b : T' ⟶ T''), L.ρ b (v T T' a) * v T' T'' b = v T T'' (a ≫ b)

theorem **isObU_unique** (L : Pre I π F) (α : Aug I π R₁) {u : ∀ (T T' : C) (_ : T ⟶ T'), (F.obj T')ˣ} {c c' : CatCoh.H2 (ext R₁ F)} (h : IsObU L α u c) (h' : IsObU L α u c') : c = c'

theorem **exists_isObU** (hs : Function.Surjective π) (L : Pre I π F) (hL : L.IsFun) (α : Aug I π R₁) {u : ∀ (T T' : C) (_ : T ⟶ T'), (F.obj T')ˣ} (hu : UCoc F u) : ∃ c, IsObU L α u c

theorem **red_sq** : RingHom.ker red ^ 2 = 0

theorem **exists_isObU_ne_zero** {R₁ : Type} [CommRing R₁] [Algebra (ZMod 8) R₁] (α : Aug (RingHom.ker red) red R₁) : ∃ c, c ≠ 0 ∧ IsObU (cstPre (ZMod.ringHom_surjective red) (SingleObj (ZMod 8)ˣ)) α (fun _ _ a ↦ a) c

theorem **isCoc_of_isFrame** {L : LB X} (hL : L.IsLB) {V : C → X.Opens} {hV : ∀ ⦃T T' : C⦄, (T ⟶ T') → V T' ≤ V T} {u : ∀ (T T' : C) (_ : T ⟶ T'), Γ(X, V T')ˣ} (h : IsFrame L V hV u) : IsCoc V hV u

theorem **isBas_ch** {L : LB X} (hL : L.IsLB) : IsBas (opOf (chG L))

theorem **isFrame_ch** {L : LB X} (hL : L.IsLB) : IsFrame L (opOf (chG L)) (opOf_le (chG L)) fun T T' a ↦ res (le_inf T'.2 ((opOf_le (chG L) a).trans T.2)) (L.g T'.1.2 T.1.2)

theorem **exists_obAt** (hI : I ^ 2 = 0) (hs : Function.Surjective π) (S : Sys I π s₀) (α : Aug I π R₁) {L : LB X} (hL : L.IsLB) : ∃ d, ObAt S α L (chG L) d

theorem **isObL_triv** (hI : I ^ 2 = 0) (hs : Function.Surjective π) (S : Sys I π s₀) (α : Aug I π R₁) : IsObL S α (LB.triv X) 0

theorem **nonempty_of_isBas** [Nonempty X] {V : C → X.Opens} (h : IsBas V) : Nonempty C

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsStdOfInjective

theorem **repr_mul_succ** (Q : PDPoly B) (p : Q.P) (n : ℕ) : Q.e.repr (p * Q.e (n + 1)) 0 = 0

theorem **mem_ext_iff** (Q : PDPoly B) (p : Q.P) : p ∈ Q.ext J ↔ Q.e.repr p 0 ∈ J

theorem **comap_ext** (Q : PDPoly B) : (Q.ext J).comap (algebraMap B Q.P) = J

theorem **isStd_unique** {Q : PDPoly B} {δ : DividedPowers J} {η η' : DividedPowers (Q.ext J)} (h : Q.IsStd δ η) (h' : Q.IsStd δ η') : η = η'

theorem **exists_isStd_of_injective** (K : Type) [CommRing K] [Algebra ℚ K] [Algebra B K] (hK : Function.Injective (algebraMap B K)) (Q : PDPoly B) (δ : DividedPowers J) : ∃ η : DividedPowers (Q.ext J), Q.IsStd δ η

theorem **exists_isStd_int** : ∃ η : DividedPowers ((stdPoly ℤ).ext ⊥), (stdPoly ℤ).IsStd (dividedPowersBot ℤ) η

theorem **exists_isStd_padic** (p : ℕ) [Fact p.Prime] : ∃ η : DividedPowers ((stdPoly ℤ_[p]).ext (Ideal.span {(p : ℤ_[p])})), (stdPoly ℤ_[p]).IsStd (PadicInt.dividedPowers p) η

theorem **mem_two** (x : ZMod 4) : x ∈ Ideal.span {(2 : ZMod 4)} ↔ x = 0 ∨ x = 2

theorem **gam_mul** (m n : ℕ) (hm : m ≠ 0) (hn : n ≠ 0) (x y : ZMod 4) : gam m x * gam n y = 0

theorem **gam_isPreDP_and_ne_dpow** : IsPreDP (Ideal.span {(2 : ZMod 4)}) gam ∧ ∀ δ : DividedPowers (Ideal.span {(2 : ZMod 4)}), δ.dpow ≠ gam

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsTauZeroNonzeroAll

theorem **map_tauExpr** {S T : Type} [CommRing S] [CommRing T] (φ : S →+* T) {k : ℕ} {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) : φ (tauExpr cf x w) = tauExpr cf x (fun q => φ (w q))

theorem **tauExpr_scale** {S : Type} [CommRing S] {k : ℕ} {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) (t : S) : tauExpr cf x (fun q => t * w q) = t ^ n * tauExpr cf x w

theorem **IsAdm.spans** {R : ℕ} (hR : 1 < R) {n : ℕ} {x₀ : Fin n → Dw.La} (h : IsAdm R x₀) : EtaAn.Spans (shiftFam R x₀ : (Fin n → LatMod Dw R) → Fin n → LatMod Dw (R ^ 4))

theorem **exists_isAdm** {R : ℕ} (hR : 1 < R) : ∃ x₀ : Fin (Dw.p + Dw.q) → Dw.La, IsAdm R x₀

theorem **exists_lev_baseChange** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) {y : FrPt Xo c S} {z : FrPt Xo c T} (hm : FrMap ψ y z) {ev : LatMod Dw (R ^ 4) → (ptO Xo y.x ⟶ Xo.univ.t.G)} {w : LatMod Dw (R ^ 4) → S} (hev : IsLevOf Xo c (R ^ 4) y.x ev) (hw : ∀ q, IsNullVal Xo c (R ^ 2) y (ev q) (w q)) : ∃ ev' : LatMod Dw (R ^ 4) → (ptO Xo z.x ⟶ Xo.univ.t.G), IsLevOf Xo c (R ^ 4) z.x ev' ∧ ∀ q, IsNullVal Xo c (R ^ 2) z (ev' q) (ψ (w q))

theorem **map_val_eq** (hR : Odd R) {S₁ S₂ T : Type} [CommRing S₁] [Algebra c.F S₁] [CommRing S₂] [Algebra c.F S₂] [CommRing T] [Algebra c.F T] (ψ₁ : S₁ →ₐ[c.F] T) (ψ₂ : S₂ →ₐ[c.F] T) {y₁ : FrPt Xo c S₁} {y₂ : FrPt Xo c S₂} {z : FrPt Xo c T} (h₁ : FrMap ψ₁ y₁ z) (h₂ : FrMap ψ₂ y₂ z) {ev₁ : LatMod Dw (R ^ 4) → (ptO Xo y₁.x ⟶ Xo.univ.t.G)} {w₁ : LatMod Dw (R ^ 4) → S₁} {ev₂ : LatMod Dw (R ^ 4) → (ptO Xo y₂.x ⟶ Xo.univ.t.G)} {w₂ : LatMod Dw (R ^ 4) → S₂} (hev₁ : IsLevOf Xo c (R ^ 4) y₁.x ev₁) (hw₁ : ∀ q, IsNullVal Xo c (R ^ 2) y₁ (ev₁ q) (w₁ q)) (hev₂ : IsLevOf Xo c (R ^ 4) y₂.x ev₂) (hw₂ : ∀ q, IsNullVal Xo c (R ^ 2) y₂ (ev₂ q) (w₂ q)) : ψ₁ (tauExpr cf x w₁) = ψ₂ (tauExpr cf x w₂)

theorem **tauValAt_self** {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} {ev : LatMod Dw (R ^ 4) → (ptO Xo y.x ⟶ Xo.univ.t.G)} {w : LatMod Dw (R ^ 4) → S} (hev : IsLevOf Xo c (R ^ 4) y.x ev) (hw : ∀ q, IsNullVal Xo c (R ^ 2) y (ev q) (w q)) : TauValAt Xo c R cf x y (tauExpr cf x w)

theorem **TauValAt.map_eq** (hR : Odd R) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {y₀ : FrPt Xo c S} {y : FrPt Xo c T} (hm : FrMap φ y₀ y) {v₀ : S} {v : T} (h₀ : TauValAt Xo c R cf x y₀ v₀) (h : TauValAt Xo c R cf x y v) : φ v₀ = v

theorem **TauValAt.unique** (hR : Odd R) {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} {v v' : S} (h : TauValAt Xo c R cf x y v) (h' : TauValAt Xo c R cf x y v') : v = v'

theorem **exists_cover** (hR : Odd R) (hM : R ^ 4 ∣ l.M) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : ∃ (S' : Type) (_ : CommRing S') (_ : Algebra c.F S') (φ : S →ₐ[c.F] S') (y' : FrPt Xo c S') (ev : LatMod Dw (R ^ 4) → (ptO Xo y'.x ⟶ Xo.univ.t.G)) (w : LatMod Dw (R ^ 4) → S'), φ.toRingHom.FaithfullyFlat ∧ FrMap φ y y' ∧ IsLevOf Xo c (R ^ 4) y'.x ev ∧ ∀ q, IsNullVal Xo c (R ^ 2) y' (ev q) (w q)

theorem **exists_tauValAt** (hR : Odd R) (hM : R ^ 4 ∣ l.M) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : ∃ v, TauValAt Xo c R cf x y v

theorem **TauValAt.dvd** {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} {v : S} (h : TauValAt Xo c R cf x y v) : R ^ 4 ∣ l.M

theorem **TauValAt.map** (hR : Odd R) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {y₀ : FrPt Xo c S} {y : FrPt Xo c T} (hm : FrMap φ y₀ y) {v₀ : S} (h₀ : TauValAt Xo c R cf x y₀ v₀) : TauValAt Xo c R cf x y (φ v₀)

theorem **TauValAt.frBor** {S : Type} [CommRing S] [Algebra c.F S] {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} (h : FrBor t y y') {v : S} (hv : TauValAt Xo c R cf x y v) : TauValAt Xo c R cf x y' ((t none : S) ^ n * v)

theorem **existsUnique_isTauRule** (hR : Odd R) (hM : R ^ 4 ∣ l.M) : ∃! f, IsTauRule Xo c R cf x f

theorem **IsTauRule.mem_sec** (hR : Odd R) {f : Rul Xo c} (hf : IsTauRule Xo c R cf x f) : f ∈ sec Xo c 0 n

theorem **IsTauRule.tauValAt_evR** (hR : Odd R) {f : Rul Xo c} (hf : IsTauRule Xo c R cf x f) {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} (hy : Tested y) : TauValAt Xo c R cf x y (evR Xo c f y)

theorem **tested_genFr** (h : GenFrEx Xo c) (Z : Comp Xo c) : Tested (genFr h Z)

theorem **genVal_idem** (Z : Comp Xo c) : genVal Z (idem Z) = 1

theorem **NonzeroOn.ne_zero** {Z : Comp Xo c} {f : Rul Xo c} (h : NonzeroOn Z f) : f ≠ 0

theorem **exists_gen_val_ne_zero** {R : ℕ} (hR : Odd R) (h1 : 1 < R) (hM : R ^ 4 ∣ l.M) {n : ℕ} (x₀ : Fin n → Dw.La) {K : Type} [Field K] [Algebra c.F K] (y : FrPt Xo c K) : ∃ u : Fin n → LatMod Dw R, ∀ v, TauValAt Xo c R (fun _ : Unit => 1) (fun _ => shiftFam R x₀ u) y v → v ≠ 0

theorem **exists_isGenRule_nonzeroOn** {R : ℕ} (hR : Odd R) (h1 : 1 < R) (hM : R ^ 4 ∣ l.M) {n : ℕ} (x₀ : Fin n → Dw.La) (Z : Comp Xo c) : ∃ u f, IsGenRule Xo c R x₀ u f ∧ NonzeroOn Z f

theorem **shift_sig** (h : HasPts U Z) (s : ∀ τ, FlatSec U x Z τ) (m : ℕ) : (FlatSec.shift h s fun _ => -(4 * (m : ℤ))).sig = idem Z * ∏ τ, (s τ).sig ^ (4 * m)

theorem **pow_mem_secZ** {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) (m : ℕ) : f ^ m ∈ secZ Xo c (m • e)

theorem **ex_tau_key** (m : ℕ) : (8 • fun i => (ex Dw 0 m i : ℤ)) + m • (4 : ℤ) • keyEx Dw = exOf Dw ((8 * (m : ℤ)) • hodgeWt Dw + parWt Dw fun _ => 4 * (m : ℤ)) 0

theorem **pow_mul_key_mem_secZ** {m : ℕ} {K f : Rul Xo c} (hK : K ∈ secZ Xo c ((4 : ℤ) • keyEx Dw)) (hf : f ∈ sec Xo c 0 m) : f ^ 8 * K ^ m ∈ secZ Xo c (exOf Dw ((8 * (m : ℤ)) • hodgeWt Dw + parWt Dw fun _ => 4 * (m : ℤ)) 0)

theorem **thetaOn_mem_secZ** (h : HasPts U Z) (s : ∀ τ, FlatSec U x Z τ) {K : (Rul Xo c)ˣ} (hK : (K : Rul Xo c) ∈ secZ Xo c ((4 : ℤ) • keyEx Dw)) {f : Rul Xo c} (hf : f ∈ sec Xo c 0 (Dw.p + Dw.q)) : thetaOn h s K f ∈ secZ Xo c (exOf Dw ((8 * ((Dw.p + Dw.q : ℕ) : ℤ)) • hodgeWt Dw) 0)

theorem **thetaOn_mem_suppOn** (h : HasPts U Z) (s : ∀ τ, FlatSec U x Z τ) (K : (Rul Xo c)ˣ) (f : Rul Xo c) : thetaOn h s K f ∈ suppOn Z

theorem **nonzeroOn_thetaOn_iff** (h : HasPts U Z) (s : ∀ τ, FlatSec U x Z τ) (K : (Rul Xo c)ˣ) (f : Rul Xo c) : NonzeroOn Z (thetaOn h s K f) ↔ NonzeroOn Z f

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsUnitTermZero

theorem **term_mem** (y : Φ.Y) (mu : Fin s → Φ.X y) (c : Φ.Q) (j : ℕ) (h : (j : ℤ) ≤ ∑ i, qW W (Φ.δ y i) mu) : Φ.term y mu c ∈ maximalIdeal (Φ.V y) ^ j

theorem **isClassSum_unique** (y : Φ.Y) (N : ℕ) (κ : Fin s → Φ.X y) (c : Φ.Q) {x x' : Φ.V y} (h : Φ.IsClassSum y N κ c (RingHom.id _) x) (h' : Φ.IsClassSum y N κ c (RingHom.id _) x') : x = x'

theorem **exists_isUnit_term_zero** (y : Φ.Y) : ∃ c : Φ.Q, IsUnit (Φ.term y (0 : Fin s → Φ.X y) c)

theorem **exists_rescale** : ∃ v : ∀ y, (Φ.V y)ˣ, ∀ y : (Φ.rescale v).Y, ∃ c : (Φ.rescale v).Q, (Φ.rescale v).term y (0 : Fin s → (Φ.rescale v).X y) c = 1

theorem **low_finite** (y : Φ.Y) (P : (Fin s → Φ.X y) → Prop) (j : ℕ) : {mu | P mu ∧ ∑ i, qW W (Φ.δ y i) mu < (j : ℤ)}.Finite

theorem **exists_sum** (y : Φ.Y) (P : (Fin s → Φ.X y) → Prop) (c : Φ.Q) : ∃ x : Φ.V y, ∀ j : ℕ, ∃ S : Finset (Fin s → Φ.X y), (∀ mu ∈ S, P mu) ∧ (∀ mu, P mu → mu ∉ S → (j : ℤ) ≤ ∑ i, qW W (Φ.δ y i) mu) ∧ x - ∑ mu ∈ S, Φ.term y mu c ∈ maximalIdeal (Φ.V y) ^ j

theorem **isClassSum_lim** (y : Φ.Y) (N : ℕ) (κ : Fin s → Φ.X y) (c : Φ.Q) : Φ.IsClassSum y N κ c (RingHom.id _) (lim Φ y (fun mu => ∀ v, CongMod N (mu v) (κ v)) c)

theorem **tate_exact** (N : ℕ) : VSys.tate.pre.Exact N

theorem **tate_classSums** (N₁ : ℕ) (hN : 1 ≤ N₁) : VSys.tate.pre.ClassSums N₁

theorem **not_lam** : ¬ const.abelian₂

theorem **not_exact** : ¬ const.Exact 3

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsIsotropicComplGram

theorem **incM_mul_inv** {k : ℕ} (T : GL (Fin k) E) : incM F T.1 * incM F (T⁻¹).1 = 1

theorem **mat_eq_inBasis** (T : GL (Fin (n + n)) E) (g : UA σ J₂) : mat g = incM F T.1 * inBasis σ T g * incM F (T⁻¹).1

theorem **inBasis_mul** (T : GL (Fin (n + n)) E) (g h : UA σ J₂) : inBasis σ T (g * h) = inBasis σ T g * inBasis σ T h

theorem **exists_isWeylElt** (hσ : ∀ x, σ (σ x) = x) {J : Matrix (Fin n) (Fin n) E} {l : E} (hl : l ≠ 0) (hJ : ∀ i j, σ ((l⁻¹ • J) j i) = -(l⁻¹ • J) i j) (h₂ : J₂ = dblGram finSumFinEquiv J) (hT : IsWittBasis σ J l T) : ∃ w ∈ ratU σ J₂, IsWeylElt σ T w

theorem **IsWeylElt.mat_mul** (h : IsWeylElt σ T w) : mat w * incM F T.1 = incM F T.1 * blk 0 1 (-1) 0

theorem **IsWeylElt.mat_mul_self** (h : IsWeylElt σ T w) : mat (w * w) = -1

theorem **IsWeylElt.mulVec_diag** (h : IsWeylElt σ T w) {J : Matrix (Fin n) (Fin n) E} {l : E} (hT : IsWittBasis σ J l T) {v : Fin (n + n) → AE F E} (hv : v ∈ diagSub (AE F E) finSumFinEquiv) : mat w *ᵥ v ∈ antiSub (AE F E) finSumFinEquiv

theorem **IsWeylElt.not_mem_siegelOf** (h : IsWeylElt σ T w) {J : Matrix (Fin n) (Fin n) E} {l : E} (hT : IsWittBasis σ J l T) (hn : 1 ≤ n) : w ∉ siegelOf finSumFinEquiv (UG (cA σ) (incM F J₂)).subtype

theorem **IntConv.add** (h : IntConv νN w f g) (h' : IntConv νN w f' g) : IntConv νN w (f + f') g

theorem **IntConv.smul** (h : IntConv νN w f g) (c : ℂ) : IntConv νN w (c • f) g

theorem **intM_smul** (c : ℂ) (f : UA σ J₂ → ℂ) (g : UA σ J₂) : intM νN w (c • f) g = c * intM νN w f g

@[simp] theorem **secMap_apply** (Φ : (κ → AF F) → ℂ) (g : UA σ J₂) : secMap om Φ g = om (1, g) Φ 0

theorem **kFin_iff** (Φ : (κ → AF F) → ℂ) : FiniteDimensional ℂ (Submodule.span ℂ {Ψ | ∃ k ∈ K, Ψ = om (1, k) Φ}) ↔ ∃ W : Submodule ℂ ((κ → AF F) → ℂ), FiniteDimensional ℂ W ∧ ∀ k ∈ K, om (1, k) Φ ∈ W

theorem **mem_secSpace** {f : UA σ J₂ → ℂ} : f ∈ secSpace om K ↔ ∃ Φ ∈ schA F κ, FiniteDimensional ℂ (Submodule.span ℂ {Ψ | ∃ k ∈ K, Ψ = om (1, k) Φ}) ∧ f = fun g => om (1, g) Φ 0

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsLawsOfSignature

theorem **det_eq_one_of_half** (l : Level H) (Di : Datum H) {r : ℕ} (X : Data E r) (φ : Di.La ≃ₗ[𝒪 E] X.Λ) (ef : ∀ a b, ((Di.EF a b : ℤ) : ℚ) = Algebra.trace ℚ E (-X.ξ0 * herm X.Gm (φ a : Fin r → E) (φ b))) (γ : GL (Fin r) E) (hU : IsUStab X.Gm X.Λ γ) (hh : ∀ a ∈ l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E) • (⊤ : Submodule (𝒪 E) Di.La), ∃ b : Di.La, (γ : Matrix (Fin r) (Fin r) E).mulVec (φ a : Fin r → E) = (φ a : Fin r → E) + (l.M : ℕ) • (φ b : Fin r → E)) : (γ : Matrix (Fin r) (Fin r) E).det = 1

theorem **exists_unf** (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (ξ₀ : E) (Gm : Matrix (Fin r) (Fin r) E) (φ : Dw.La ≃ₗ[𝒪 E] (Fin r → E)) (hpq : 1 ≤ Dw.p + Dw.q) (hA : l.A = 1) (hσ : GoodEmb c σ) (hξ : cc ξ₀ = -ξ₀) (hneg : ∀ τ : E →+* ℂ, τ (H.rt : E) = H.rt0 → (τ ξ₀).im < 0) (hφ : Function.Injective φ ∧ Submodule.span E (Set.range φ) = ⊤ ∧ (∀ a b, cc (Gm a b) = Gm b a) ∧ Gm.det ≠ 0 ∧ ∀ x y, ((Dw.EF x y : ℤ) : ℚ) = Algebra.trace ℚ E (-ξ₀ * herm Gm (φ x) (φ y))) : ∃ U : Formalism.L2.PjFree.Unf Xo c σ r, Formalism.L2.PjFree.IsUnf U ∧ (∀ k la γ, (U.X k).χ la γ = 1) ∧ (∀ k, (U.X k).Gm = Gm ∧ (U.X k).ξ0 = ξ₀) ∧ (∀ k, GlobalTheory.Dict.HasPrinc (U.X k) l.M) ∧ ∀ k, ∀ γ ∈ (U.X k).Γ, (γ : Matrix (Fin r) (Fin r) E).det = 1

theorem **exists_laws** (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (hpq : 1 ≤ Dw.p + Dw.q) (hA : l.A = 1) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (hF : FnAlgClosed Xo c) (h1 : ∃ D₁ : Datum H, D₁.p = 1 ∧ D₁.q = 0) : ∃ U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q), Laws U (Real.sqrt H.d)

theorem **exists_laws_of_signature** (h1 : ∃ D₁ : Datum H, D₁.p = 1 ∧ D₁.q = 0) (p q : ℕ) (hpq : 1 ≤ p + q) : ∃ (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q)), Dw.p = p ∧ Dw.q = q ∧ Nonempty (Comp Xo c) ∧ Laws U (Real.sqrt H.d)

theorem **exists_laws_cyclotomic** (p q : ℕ) (hpq : 1 ≤ p + q) : ∃ (Dw : Datum GlobalTheory.CM.CyclotomicTwentyEight.host) (l : Level GlobalTheory.CM.CyclotomicTwentyEight.host) (Xo : Moduli Dw l) (c : Coeff l) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q)) (x : ℝ), Dw.p = p ∧ Dw.q = q ∧ Nonempty (Comp Xo c) ∧ Laws U x

theorem **archEx_of_definite** {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {r : ℕ} {U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r} {x : ℝ} (h : Laws U x) (hq : Dw.q = 0) : ArchEx U x

theorem **hasPts** (h : Laws U x) (Z : Comp Xo c) : HasPts U Z

theorem **htValAt_nonempty** (la : HeightOneSpectrum (𝒪 c.F)) {T : CDVR} {io : c.F →+* T.Fr} (y₀ : IntFr Xo c T io) (hy : y₀.IsAlg) : (htValAt la y₀).Nonempty

theorem **htValOn_nonempty** (h : Laws U x) (Z : Comp Xo c) (la : HeightOneSpectrum (𝒪 c.F)) : (htValOn Z la).Nonempty

theorem **finConst_of** (h₁ : HTConst U x) (h₂ : HTShift Xo c) : FinConst U x

theorem **flatDegree_of** (h : Laws U x) {om : ℝ} (h₁ : ArchEx U x) (h₂ : FinConst U x) (h₃ : FrameDeg U x om) : FlatDegree U x om

theorem **exists_laws_record** (D : Formalism.BasicNotions.D7) (p q : ℕ) (hpq : 1 ≤ p + q) : GlobalTheory.IndefiniteRowSideFullHecke.EigenvectorLieDegree.IsHostField (hostOfRecord D) ∧ ∃ (Dw : Datum (hostOfRecord D)) (l : Level (hostOfRecord D)) (Xo : Moduli Dw l) (c : Coeff l) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q)), Dw.p = p ∧ Dw.q = q ∧ Nonempty (Comp Xo c) ∧ Laws U (Real.sqrt (hostOfRecord D).d)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsLeadOfUnif

theorem **LabCover.exists_trait** (C : LabCover D) (W : Type) [CommRing W] [IsDomain W] [IsLocalRing W] (hW : GlobalTheory.LN.CDvr W) (φ : R →+* W) (ξ : Sp W ⟶ C.S) (hφ : Function.Injective φ) (hl : IsLocalHom φ) (hξ : ξ ≫ C.π = spMap φ) : ∃ (W' : Type) (_ : CommRing W')

(_ : IsDomain W') (_ : IsLocalRing W') (io : W →+* W') (T : Trait C W'), Function.Injective io ∧ io.Finite ∧ T.φ = io.comp φ ∧ T.ξ = spMap io ≫ ξ

theorem **Trait.exists_toric_rank_one** : ∃ (D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (C : LabCover D) (W : Type) (_ : CommRing W) (_ : IsDomain W) (_ : IsLocalRing W), Module.finrank ℤ D.toPre.X = 1 ∧ Nonempty (Trait C W)

theorem **IntFam.exists_lead_of_unif** {V : Type} [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] {D : DegData V (maximalIdeal V)} {A : PolAV (Sp (FractionRing V))} {G : Over (Sp V)} [GrpObj G] [IsCommMonObj G] (U : Unif (maximalIdeal V) D A G) : ∃ F : IntFam (maximalIdeal V) D.toPre 1, F.Lead 0 (F.f 0) ∧ ¬ VanMod (maximalIdeal V) (F.f 0) 0

theorem **IntFam.exists_toric_rank_one** : ∃ (D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (F : IntFam (maximalIdeal (PowerSeries ℚ)) D.toPre 1), Module.finrank ℤ D.toPre.X = 1 ∧ F.Lead 0 (F.f 0) ∧ ¬ VanMod (maximalIdeal (PowerSeries ℚ)) (F.f 0) 0

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsLevFnPow

theorem **norm_int_le** {E : Type} [Field E] [NumberField E] (τ : E →+* PadicAlgCl l) (x : 𝒪 E) : ‖τ x‖ ≤ 1

theorem **exists_adele** (F : Type) [Field F] [NumberField F] (τ : F →+* PadicAlgCl l) (K : IntermediateField ℚ_[l] (PadicAlgCl l)) [FiniteDimensional ℚ_[l] K] (hK : ∀ x, τ x ∈ K) : ∃ ev : FiniteAdeleRing (𝒪 F) F →+* PadicAlgCl l, (∀ x, ev (algebraMap F _ x) = τ x) ∧ (∀ a, ev a ∈ K) ∧ ∀ a : FiniteAdeleRing (𝒪 F) F, (∀ v, a v ∈ v.adicCompletionIntegers F) → ‖ev a‖ ≤ 1

theorem **exists_ev** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (τ : E →+* PadicAlgCl l) (K : IntermediateField ℚ_[l] (PadicAlgCl l)) [FiniteDimensional ℚ_[l] K] (hK : ∀ x, τ x ∈ K) : ∃ ev : E ⊗[F] FiniteAdeleRing (𝒪 F) F →+* PadicAlgCl l, (∀ x, ev (x ⊗ₜ[F] 1) = τ x) ∧ (∀ z, ev z ∈ K) ∧ ∀ z ∈ Formalism.L2.Herm.intAf F E, ‖ev z‖ ≤ 1

theorem **exists_levFn_pow** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} {LL : Type} [CommRing LL] (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) (D : LevDat E LL n) : ∃ M : ℕ, 0 < M ∧ ∀ x, levFn β D x ^ M = 1

theorem **exists_field** (L : Type) [Field L] [NumberField L] (e : L →+* PadicAlgCl l) {M : ℕ} (hM : 0 < M) : ∃ K : IntermediateField ℚ_[l] (PadicAlgCl l), FiniteDimensional ℚ_[l] K ∧ (∀ x, e x ∈ K) ∧ ∀ z, z ^ M = 1 → z ∈ K

theorem **embInt_injective** (K : IntermediateField ℚ_[l] (PadicAlgCl l)) : Function.Injective (embInt K)

theorem **range_embInt** (K : IntermediateField ℚ_[l] (PadicAlgCl l)) : Set.range (embInt K) = {x | x ∈ K ∧ ‖x‖ ≤ 1}

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsMemSecFinHyperbolic

theorem **exists_isQuad_conjR** (h : IsCMExt F E σ) (R : Type) [CommRing R] [Algebra F R] : ∃ δ, IsQuad (conjR R σ) δ

theorem **continuous_psiAt** {ψ : AddChar (AF F) ℂ} (hψ : Continuous ψ) (v : HeightOneSpectrum (𝒪 F)) : Continuous (LatVac.psiAt ψ v)

theorem **psiAt_ne_one** (hF : IsTotallyReal F) {ψ : AddChar (AF F) ℂ} (hψ : IsCharAF F ψ) (v : HeightOneSpectrum (𝒪 F)) : LatVac.psiAt ψ v ≠ 1

theorem **exists_mem_secFin_hyperbolic** (hc : IsCMExt F E σ) (hQ : Q = StdUnn.blk4 0 1 (-1) 0) {ψ : AddChar (AF F) ℂ} (hψ : IsCharAF F ψ) (v : HeightOneSpectrum (𝒪 F)) : ∃ f ∈ secFin σ hQ ψ 1 !![0, 1; 1, 0] v, f 1 ≠ 0 ∧ f ∈ psUp _ σ Q (absEv F E v) (KUpFin σ Q v) (charAt 1 v) ((((2 : ℕ) : ℂ) - n) / 2)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsMonoFamNontrivial

theorem **genOf_iff** (Xo : Moduli Dw l) (c : Coeff l) {R : Type} [CommRing R] [IsDomain R] (y : Sp (FractionRing R) ⟶ Xo.XF c) (t : SAB H (Sp R)) : GlobalTheory.UPoinc.GenOf Xo c y t ↔ Classifies Xo c t (jK R) y

theorem **Classifies.pull** {Xo : Moduli Dw l} {c : Coeff l} {T T' T'' : Scheme.{0}} {t : SAB H T} {g : T' ⟶ T} {y : T' ⟶ Xo.XF c} (h : Classifies Xo c t g y) (f : T'' ⟶ T') : Classifies Xo c t (f ≫ g) (f ≫ y)

theorem **tauIn_iff** {V : Type} [CommRing V] [IsDomain V] (Q : DegPre V) (τ₀ : Gens Q) {N : ℕ} (s : Fin N → V) (ε : Fin N → LinearMap.BilinForm ℤ Q.X) : TauIn Q τ₀ s ε ↔ ∀ a b, GlobalTheory.RigChart.IsUnitTau Q s ε a b (τ₀ a b)

theorem **openCls_iff** (C : MonoFam Xo c φ R) : OpenCls C ↔ ∃ yo : Sp (Localization.Away (∏ i, C.s i)) ⟶ Xo.XF c, Classifies Xo c C.t (spMap (algebraMap R (Localization.Away (∏ i, C.s i)))) yo ∧ ∀ g : Sp (FractionRing R) ⟶ Sp (Localization.Away (∏ i, C.s i)), g ≫ spMap (algebraMap R (Localization.Away (∏ i, C.s i))) = jK R → g ≫ yo = C.y

theorem **Fam.rank_le** [IsNoetherianRing R] (C : Fam Xo c R) : Module.finrank (𝒪 E) C.P.D.toPre.X ≤ min Dw.p Dw.q

theorem **level_pos** (l : Geo.Tor.Level H) : 0 < l.M'

theorem **MonoFam.s_ne_zero** (C : MonoFam Xo c φ R) (i : Fin C.m) : C.s i ≠ 0

theorem **MonoFam.exists_tauIn** (C : MonoFam Xo c φ R) : ∃ τ₀ : Gens C.P.D.toPre, TauIn C.P.D.toPre τ₀ C.s (fun i => (l.M' : ℤ) • C.δ i)

theorem **exists_monoFam_of_point** (φ : InvFun (𝒪 E)) (hR : GlobalTheory.Mum.Base R) (x : Sp R ⟶ Xo.XF c) : ∃ C : MonoFam Xo c φ R, C.m = 0 ∧ C.y = jK R ≫ x ∧ Nonempty (SABMap C.t Xo.univ.t (x ≫ pullback.fst _ _))

theorem **Fam.exists_monoFam** [IsNoetherianRing R] (C : Fam Xo c R) (hφ : IsPolFan H.bar (min Dw.p Dw.q) φ) {u : R} (hu : u ∈ maximalIdeal R) {β : LinearMap.BilinForm ℤ C.P.D.toPre.X} (hβ : β ∈ balForms H.bar C.P.D.toPre.X) (h : IsMonomial C.P.D.toPre ![u] ![(l.M' : ℤ) • β]) : ∃ C' : MonoFam Xo c φ R, C'.toFam = C

theorem **exists_monoFam_nontrivial** : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (φ : InvFun (𝒪 E)) (C : MonoFam Xo c φ (PowerSeries ℂ)), IsPolFan H.bar (min Dw.p Dw.q) φ ∧ NoSelfInt H.bar (min Dw.p Dw.q) φ ∧ GlobalTheory.Mum.Base (PowerSeries ℂ) ∧ Nontrivial C.P.D.toPre.X ∧ 1 ≤ C.m

theorem **sepAt_rescale** (Q : DegPre V) (s : Fin m → V) (gam : Fin m → LinearMap.BilinForm ℤ Q.X) (ev : V →+* K) (k₀ : Subfield K) (u : Fin m → V×) (h : SepAt Q s gam ev k₀) : SepAt Q (fun i => u i • s i) gam ev k₀

theorem **exists_val_eq** (k₀ : Subfield K) {Y : Scheme.{0}} {M : LB Y} {I : Type} [Fintype I] {p : I → (Sp k₀ ⟶ Y)} {n : I → ℤ} (τ : LB.Gen (LB.LinComb M p n)) {z : Sp K} {J : (Sp K ⟶ Y) → M.ι} (hJ : Good M z J) : ∃ x : k₀×, val τ (spMap k₀.subtype) z J = gm (𝟙 (Sp K)) z ⊤ (unitOn (Units.map k₀.subtype.toMonoidHom x))

theorem **one_isGen₂** {W W' Y : Scheme.{0}} (f : Y ⟶ W) (g : Y ⟶ W') : LB.Gen.IsGen ((fun _ => 1) : LB.Iso ((Inh.trivLB W).pull f) ((Inh.trivLB W').pull g))

theorem **exists_iso_pull_pull** {B B₀ B' : Scheme.{0}} {M : LB B} (hM : M.IsLB) (β : B₀ ⟶ B) (π : B' ⟶ B₀) : ∃ w : LB.Iso (M.pull (π ≫ β)) ((M.pull β).pull π), w.IsGen ∧ ∀ (Z : Scheme.{0}) (f : Z ⟶ B') (z : Z) (j k : PUnit → M.ι), β (π (f z)) ∈ M.U (j PUnit.unit) → β (π (f z)) ∈ M.U (k PUnit.unit) → gw w f z j k = gm (f ≫ π ≫ β) z _ (M.g (j PUnit.unit) (k PUnit.unit))

theorem **sepAt_of_range** (Q : DegPre V) (s : Fin m → V) (gam : Fin m → LinearMap.BilinForm ℤ Q.X) (ev : V →+* K) (k₀ : Subfield K) (h0 : ∀ i, ev (s i) = 0) (h : ∀ v, ev v ∈ k₀) : SepAt Q s gam ev k₀

theorem **lamUnit_one** {W W' B' Z : Scheme.{0}} {π₁ : B' ⟶ W} {π₀ : B' ⟶ W'} {X : Type} [AddCommGroup X] (x : X → (Z ⟶ B')) (a b : X) (z : Z) (J J₀) : lamUnit ((fun _ => 1) : LB.Iso ((Inh.trivLB W).pull π₁) ((Inh.trivLB W').pull π₀)) x a b z J J₀ = 1

theorem **sepAt_toric_iff** {X : Type} [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (q : X → (FractionRing V)ˣ) (s : Fin m → V) (gam : Fin m → LinearMap.BilinForm ℤ X) (ev : V →+* K) (k₀ : Subfield K) : SepAt (toricPre V X q) s gam ev k₀ ↔ (∀ i, ev (s i) = 0) ∧ ∃ α : Fin m → Kˣ, ∀ (a b : X) (u : Vˣ), (polar q a b : FractionRing V) = (∏ i, algebraMap V (FractionRing V) (s i) ^ gam i a b) * algebraMap V _ u → ∃ x ∈ k₀, ev u = (∏ i, (α i : K) ^ gam i a b) * x

theorem **sepAt_per_iff** (v : Aˣ) (b : ℤ × ℤ → ℤ) (ev : A →+* K) (k₀ : Subfield K) : SepAt (toricPre A (ℤ × ℤ) (per v b)) ![tt] ![frm] ev k₀ ↔ ev tt = 0 ∧ ∃ α : Kˣ, ∀ y z : ℤ × ℤ, ∃ x ∈ k₀, ev (v ^ (b (y + z) − b y − b z) : Aˣ) = (α : K) ^ frm y z * x

theorem **sepAt_mixed_iff** (ev : A →+* K) (k₀ : Subfield K) : SepAt (toricPre A (ℤ × ℤ) (per vs fun y => y.1 * y.2)) ![tt] ![frm] ev k₀ ↔ ev tt = 0 ∧ ev ss ∈ k₀

theorem **sepAt_cone_iff** (ev : A →+* K) (k₀ : Subfield K) : SepAt (toricPre A (ℤ × ℤ) (per vs fun y => nrm y)) ![tt] ![frm] ev k₀ ↔ ev tt = 0

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsNotIsConstTerm

theorem **foldr_char** {φ : GAd σ J → ℂ} (hφ : ∀ g h, φ (g * h) = φ g * φ h) (l : List (ℝ → GAd σ J)) : ∃ c : ℂ, l.foldr lineDer φ = c • φ

theorem **isAut_char** (K : Subgroup (GAd σ J)) {φ : GAd σ J → ℂ} (hφ : ∀ g h, φ (g * h) = φ g * φ h) (hm : IsSmoothModerate σ J φ) : IsAut σ J K φ

theorem **isAut_one** (K : Subgroup (GAd σ J)) : IsAut σ J K fun _ => 1

theorem **exists_central_line** (h : IsCMExt F E σ) (hJ : IsHermND σ J) {w : InfinitePlace F} (τ : EmbOver E w) : ∃ γ : ℝ → GAd σ J, IsCentral σ J (Finsupp.single [γ] 1) ∧ ∀ t : ℝ, archMat σ J τ (projArch σ J w (γ t)) = NormedSpace.exp (t • (Complex.I • (1 : Matrix (Fin n) (Fin n) ℂ)))

theorem **isConstTerm_const** {r : ℕ} (C : Matrix (Fin n) (Fin r) E) (c : ℂ) : IsConstTerm σ J C (fun _ => c) fun _ => c

theorem **isConstTerm_zero** {r : ℕ} {C : Matrix (Fin n) (Fin r) E} {φ : GAd σ J → ℂ} (h : IsCuspidal σ J φ) (hC : 0 < r ∧ LinearIndependent E C.transpose ∧ Formalism.UG.cT (σ : E →+* E) C * J * C = 0) : IsConstTerm σ J C φ 0

theorem **exists_not_isConstTerm** (h : ¬ IsCuspidal σ J fun _ => 1) : ∃ (r : ℕ) (C : Matrix (Fin n) (Fin r) E), 0 < r ∧ Formalism.UG.cT (σ : E →+* E) C * J * C = 0 ∧ ¬ IsConstTerm σ J C (fun _ => 1) 0

theorem **isMaxCpt_infPts** (h : IsCompact (infPts σ J : Set (GAd σ J))) : IsMaxCpt σ J (infPts σ J)

theorem **isSqInt_of_bounded** (hP : GlobalTheory.Published.SiegelSetUnitary) (h : IsCMExt F E σ) (hJ : IsHermND σ J) [Countable (ratPts σ J)] {φ : GAd σ J → ℂ} (hc : Continuous φ) {M : ℝ} (hM : ∀ g, ‖φ g‖ ≤ M) : IsSqInt σ J φ

theorem **exists_exp** (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) {γ : ℝ → GAd σ J} (hγ : IsRealLine σ J γ) : ∃ Z : Matrix (Fin n) (Fin n) ℂ, ∀ t : ℝ, archMat σ J τ (projArch σ J w (γ t)) = NormedSpace.exp (t • Z)

theorem **isSm_entry** (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) (i j : Fin n) : IsSm σ J fun g => archMat σ J τ (projArch σ J w g) i j

theorem **commute_of_isCentral** (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) {γ : ℝ → GAd σ J} {Z : Matrix (Fin n) (Fin n) ℂ} (hZ : ∀ t : ℝ, archMat σ J τ (projArch σ J w (γ t)) = NormedSpace.exp (t • Z)) (hc : IsCentral σ J (Finsupp.single [γ] 1)) (y : GArch σ J w) : Z * archMat σ J τ y = archMat σ J τ y * Z

theorem **line_mem_infPts** {γ : ℝ → GAd σ J} (hγ : IsRealLine σ J γ) (t : ℝ) : γ t ∈ infPts σ J

theorem **isAut_of_open** {U K : Subgroup (GAd σ J)} (hU : IsOpen (U : Set (GAd σ J))) (hi : infPts σ J ≤ U) (hK : K ≤ infPts σ J) {φ : GAd σ J → ℂ} (hl : ∀ γ ∈ ratPts σ J, ∀ g, φ (γ * g) = φ g) (hr : ∀ u ∈ U, ∀ g, φ (g * u) = φ g) (hb : ∀ g, ‖φ g‖ ≤ 1) : IsAut σ J K φ

theorem **infinite_fin** (K : Type) [Field K] [NumberField K] : Infinite (IsDedekindDomain.HeightOneSpectrum (𝒪 K))

theorem **ratLoc_inj** (v : IsDedekindDomain.HeightOneSpectrum (𝒪 F)) : Function.Injective (ratLoc σ J v)

theorem **exists_isAut_nonconst** (hn : 0 < n) (hc : IsCompact (infPts σ J : Set (GAd σ J))) : ∃ φ : GAd σ J → ℂ, (∀ K ≤ infPts σ J, IsAut σ J K φ) ∧ φ 1 = 1 ∧ ∃ g, φ g = 0

theorem **isCompact_infPts_one** (h : IsCMExt F E σ) : IsCompact (infPts σ (1 : Matrix (Fin n) (Fin n) E) : Set (GAd σ (1 : Matrix (Fin n) (Fin n) E)))

theorem **exists_nonconst_one** (h : IsCMExt F E σ) (hn : 0 < n) : ∃ K φ, IsMaxCpt σ (1 : Matrix (Fin n) (Fin n) E) K ∧ IsAut σ 1 K φ ∧ ∃ g, φ g ≠ φ 1

theorem **hasDerivAt_entry** {Z : Matrix (Fin n) (Fin n) ℂ} (P A : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : HasDerivAt (fun t : ℝ => (P * NormedSpace.exp (t • Z) * A) i j) ((P * Z * A) i j) 0

theorem **entry_le_opNorm** (A : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : ‖A i j‖₊ ≤ opNorm A

theorem **isAutOn_entry** (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) (K : Subgroup (GAd σ J)) (i j : Fin n) : IsAutOn σ J ⊥ K fun g => archMat σ J τ (projArch σ J w g) i j

theorem **not_zfin_exp** (h : IsCMExt F E σ) (hJ : IsHermND σ J) {w : InfinitePlace F} (τ : EmbOver E w) (i : Fin n) : IsSm σ J (fun g => Complex.exp (archMat σ J τ (projArch σ J w g) i i)) ∧ ¬ FiniteDimensional ℂ (Submodule.span ℂ {f | ∃ z, IsCentral σ J z ∧ f = dop z fun g => Complex.exp (archMat σ J τ (projArch σ J w g) i i)})

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsNotRelZero

theorem **translates_one** {H : Type} [MulOneClass H] (φ : H → ℂ) : translates (Finsupp.single 1 1) φ = φ ∧ zdeg (Finsupp.single (1 : H) 1) = 1

theorem **toTens_surjective** {P Q : Type} [AddCommGroup P] [Module ℂ P] [AddCommGroup Q] [Module ℂ Q] : Function.Surjective (toTens (P

theorem **exists_matched** (i j : ℕ) (b : (Unit × PolF D (G.V (.grid i (j + 1))).rank (G.V (.grid i j)).rank) × PolF D (G.V (.grid i j)).rank (G.V (.grid (i + 1) j)).rank →₀ ℂ) : ∃ a : (Unit × PolF D (G.V (.grid i (j + 1))).rank (G.V (.grid (i + 1) (j + 1))).rank) × PolF D (G.V (.grid (i + 1) (j + 1))).rank (G.V (.grid (i + 1) j)).rank →₀ ℂ, ∀ t : T, ρ.Matched (t

theorem **exists_not_rel_zero** (i j : ℕ) : ∃ a, (a, 0) ∉ ρ.rel i j

theorem **exists_isVac** (W : G.Weil) {μ₀ : T} (hL : L.Is μ₀) {p : Pair T} (hp : p.InRange D.R.m) : ∃ f, W.IsVac p f

theorem **exists_rel** (X M M' Y : Node) : ∃ R : Submodule ℂ (PolRoute G X M Y × PolRoute G X M' Y), (∀ a, ∃ b, (a, b) ∈ R) ∧ (∀ b, ∃ a, (a, b) ∈ R) ∧ R ≠ ⊤

theorem **nonempty_realMatch** (W : G.Weil) {μ₀ : T} (hL : L.Is μ₀) : Nonempty (RealMatch W)

theorem **flipTensor_eq** (i j : ℕ) (w : Pl D.host.E) (n : ℕ) (S : (locGrid G t w).Args → ℂ) (x : (locGrid G t w).Args) : (locGrid G t w).flipTensor .one (psiAt W w) (unifAt w) n i j (givenLevel G t w) S x = flipAt W t i j w n ((locGrid G t w).sliceB i j S x) ((locGrid G t w).joinA i j x)

theorem **IsSquareConst.unique** (h : SquareNonzero W B ρ t i j) {c c' : ℂ} (hc : IsSquareConst W B ρ t i j c) (hc' : IsSquareConst W B ρ t i j c') : c = c'

theorem **IsSquareConst.of_integrable** {c : ℂ} (hc : IsSquareConst W B ρ t i j c) {M : Vol G} (hM : M.Is t) {θ θ'} (hθ : IsMatched B t ρ i j θ θ') {g g₁} (h : IntegrableOn (kerBig W θ g g₁) (M.fd _) (M.ν _)) (h' : IntegrableOn (kerSmall W θ' g g₁) (M.fd _) (M.ν _)) : ∫ u in M.fd _, kerBig W θ g g₁ u ∂M.ν _ = c * ∫ u in M.fd _, kerSmall W θ' g g₁ u ∂M.ν _

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsOfPoint

theorem **TautBase.exists_of_point** {o W : Type} [CommRing o] [CommRing W] [IsDomain W] [IsLocalRing W] [IsNoetherianRing W] (b : o →+* W) (P : PolAV (Sp W)) (p : 𝟙_ (Over (Sp W)) ⟶ P.A) : ∃ (T : TautBase o W) (a : Fin T.r → ℤ), T.B = P.A ∧ T.g = P.g ∧ T.r = 1 ∧ HEq (T.c₀ a) p

theorem **TautBase.exists_ell** {o W : Type} [CommRing o] [CommRing W] [IsDomain W] [IsLocalRing W] [IsNoetherianRing W] (b : o →+* W) (E : _root_.WeierstrassCurve W) [E.IsElliptic] (p : 𝟙_ (Over (Sp W)) ⟶ (GlobalTheory.WeierstrassCurve.ellPolAV E).A) : ∃ (T : TautBase o W) (a : Fin T.r → ℤ), T.B = (GlobalTheory.WeierstrassCurve.ellPolAV E).A ∧ T.g = 1 ∧ T.r = 1 ∧ HEq (T.c₀ a) p

theorem **TautBase.c₀_zero** {o W : Type} [CommRing o] [CommRing W] [IsLocalRing W] (T : TautBase o W) : T.c₀ 0 = 1

theorem **TautBase.c₀_neg** {o W : Type} [CommRing o] [CommRing W] [IsLocalRing W] (T : TautBase o W) (a : Fin T.r → ℤ) : T.c₀ (-a) = (T.c₀ a)⁻¹

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsOverIff

theorem **exists_over_iff** {R : Type} [CommRing R] [IsDomain R] [IsIntegrallyClosed R] (y : Sp (FractionRing R) ⟶ Xo.XF c) : (∃ σ : Sp R ⟶ Sp c.F, jK R ≫ σ = y ≫ pullback.snd _ _) ↔ ∀ n : ℕ, n ≠ 0 → IsUnit (n : R)

theorem **UPoinOn.exists_extension** {φ : InvFun (𝒪 E)} {Z : Scheme.{0}} {zs : Z ⟶ Sp c.F} {i : Xo.XF c ⟶ Z} {OZ : LB Z} {N d : ℕ} {J : Z.IdealSheafData} {S : Scheme.{0}} {π : S ⟶ Z} {j : Xo.XF c ⟶ S} {BD : EffCartierData S} (h : UPoinOn Xo c φ zs i OZ N d J π j BD) {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] [Algebra ℚ R] (hR : GlobalTheory.Mum.Base R) {t : SAB H (Sp R)} (P : PELUnif (maximalIdeal R) t) {y : Sp (FractionRing R) ⟶ Xo.XF c} (hy : GenOf Xo c y t) : ∃ ξ : Sp R ⟶ Z, jK R ≫ ξ = y ≫ i

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsPlacePlane

theorem **not_contains_top** (E : Place c) : ¬ E.Contains ⊤

theorem **klt_top_iff** (η : Forms c n) (φ : K →₀ ℚ) : Klt c n ⊤ η φ ↔ η ≠ 0 ∧ φ 0 = 0

theorem **disc_smul** [PerfectField k] {x : Fin n → K} (hx : TrBasis c x) (E : Place c) {η : Forms c n} (hη : η ≠ 0) {f : K} (hf : f ≠ 0) (φ : K →₀ ℚ) : E.disc (f • η) (φ + Finsupp.single f 1) = E.disc η φ

theorem **klt_smul_iff** [PerfectField k] {x : Fin n → K} (hx : TrBasis c x) (R : Subring K) {η : Forms c n} {f : K} (hf : f ≠ 0) (φ : K →₀ ℚ) : Klt c n R (f • η) (φ + Finsupp.single f 1) ↔ Klt c n R η φ

theorem **disc_single_mul** (E : Place c) (η : Forms c n) {f g : K} (hf : f ≠ 0) (hg : g ≠ 0) (s : ℚ) : E.disc η (Finsupp.single (f * g) s) = E.disc η (Finsupp.single f s + Finsupp.single g s)

theorem **simple_pole** (E : Place c) {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) {t : E.ring} (ht : Irreducible t) : ¬ Klt c n E.ring.toSubring ((t : K)⁻¹ • E.toRat n ω) 0 ∧ EffAt c n E.ring.toSubring ((t : K)⁻¹ • E.toRat n ω) 0

theorem **simple_zero** (E : Place c) {ω : Forms E.const n} (hω : IsFreeGen E.ring ω) {t : E.ring} (ht : Irreducible t) : Klt c n E.ring.toSubring ((t : K) • E.toRat n ω) 0 ∧ ¬ EffAt c n E.ring.toSubring ((t : K) • E.toRat n ω) 0

theorem **isEmpty_place_self** : IsEmpty (Place (RingHom.id k))

theorem **isPrimeOf_of_isAt** {E : DivOver p} {x z : X} (hE : E.IsAt z) (hz : z ⁼ x) : (placeEquiv p E).IsPrimeOf (locRing x)

theorem **nonneg_of_effAt** {Δ : X → ℚ} {η : RatForms p n} {x z : X} {m : ℕ} {g : X.functionField×} (h : IsLogEqn p Δ η x m g) (hm : 0 < m) (he : EffAt (Cartier.constFn p) n (locRing x) η (Finsupp.single (g : X.functionField) (1 / (m : ℚ)))) {E : DivOver p} (hE : E.IsAt z) (hz : z ⁼ x) : 0 ≤ Δ z

theorem **exists_klt_smooth** [PerfectField k] [SmoothOfRelativeDimension n p] (hn : topologicalKrullDim X = n) (x : X) : IsLocModel (Cartier.constFn p) (locRing x) ∧ ∃ η φ, Klt (Cartier.constFn p) n (locRing x) η φ

theorem **projectiveSpace** (d : ℕ) (x : (ProjVar.proj ℂ d).X) : IsLocModel (Cartier.constFn (ProjVar.proj ℂ d).str) (locRing x) ∧ ∃ η φ, Klt (Cartier.constFn (ProjVar.proj ℂ d).str) (ProjVar.proj ℂ d).dim (locRing x) η φ

theorem **dForm_blowup** (c : k →+* K) (t u : K) : dForm c ![t, t * u] = t • dForm c ![t, u]

theorem **exists_chart** [PerfectField k] (E : Place c) {x : Fin 2 → K} (hx : TrBasis c x) : ∃ t u : E.ring, Irreducible t ∧ IsUnit u ∧ IsUnit (u - 1) ∧ IsFreeGen E.ring (dForm E.const ![t, u])

theorem **three_lines** [PerfectField k] (E : Place c) {x : Fin 2 → K} (hx : TrBasis c x) : ∃ a b : K, a ∈ E.ring ∧ b ∈ E.ring ∧ dForm c ![a, b] ≠ 0 ∧ ∀ s : ℚ, E.disc (dForm c ![a, b]) (Finsupp.single a s + Finsupp.single b s + Finsupp.single (b - a) s) = 1 - 3 * s ∧ ∀ R : Subring K, E.Contains R → 2 / 3 ≤ s → ¬ Klt c 2 R (dForm c ![a, b]) (Finsupp.single a s + Finsupp.single b s + Finsupp.single (b - a) s)

theorem **exists_place_plane** : Nonempty (Place (Cartier.constFn (spMap (algebraMap k (Polynomial (Polynomial k)))))) ∧ ∃ x : Fin 2 → (Sp (Polynomial (Polynomial k))).functionField, TrBasis (Cartier.constFn (spMap (algebraMap k (Polynomial (Polynomial k))))) x

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsPointGaussian

theorem **latNorm_scaleLat** (La : Submodule O U) {a : C} (ha : a ≠ 0) (x : U) : latNorm v (scaleLat a La) x = v a⁻¹ * latNorm v La x

theorem **grNorm_scaleLat** (La : Submodule O U) {a : C} (ha : a ≠ 0) (f : Fin N → U) (k : Fin N) : grNorm v (scaleLat a La) f k = v a⁻¹ * grNorm v La f k

theorem **phiAt_scale_inv** (p : ℕ) (h0 : v (p : C) ≠ 0) (h1 : v (p : C) < 1) (al : U →ₗ[C] U') (LW : Submodule O U) (Lom : Submodule O U') (f : Fin N → U) (j : ℕ) (lam : Fin N → ℝ) (hW : ∀ k : Fin N, (k : ℕ) < j → grNorm v LW f k ≠ 0) : phiAt_vp p v al (scaleLat (p : C)⁻¹ LW) Lom f j lam = phiAt_vp p v al LW Lom f j lam - Real.log p * ∑ k : Fin N, if (k : ℕ) < j then lam k else 0

theorem **pairV_eq_zero** {τ τ' : Λ →+* O} {x y : P.V} (hx : x ∈ P.eigV τ) (hy : y ∈ P.eigV τ') (a : Λ) (ha : algebraMap O C (τ a) ≠ algebraMap O C (τ' (star a))) : P.pairV x y = 0

theorem **pairV_proj** (i : J) (x : P.V) {y : P.V} (hy : y ∈ P.eigV ((σ i).comp (starRingEnd Λ))) : P.pairV (proj h P.actV i x) y = P.pairV x y

theorem **mem_latTSub_iff** (i : J) {y : P.V} (hy : y ∈ P.eigV ((σ i).comp (starRingEnd Λ))) : y ∈ P.latTSub ((σ i).comp (starRingEnd Λ)) ↔ ∀ x ∈ P.latTQuot h i, ∃ c : O, algebraMap O C c = P.pairV x y

theorem **pairV_bLie_eq_zero** (x : LieG P.Gt) {w : P.V} (hw : w ∈ P.W) : P.pairV (P.bLie x) w = 0

theorem **omHT_le_eigOm** (i : J) : P.omHT h i ≤ (P.eigOm (σ i)).restrictScalars O

theorem **mem_wHT_iff** (i : J) (x : P.V) : x ∈ P.wHT h i ↔ x ∈ P.latTQuot h i ∧ P.alC x = 0

theorem **gaussMat_star** (z : GaussianInt) : gaussMat (star z) = star (gaussMat z)

theorem **gaussBasis_apply** : gaussBasis 0 = (1, 0) ∧ gaussBasis 1 = (0, 1)

theorem **exists_point_gaussian** : ∃ (_ : Fact ¬IsUnit ((2 : ℕ) : 𝒪_ℂ_[2])) (_ : IsAdicComplete (Ideal.span {((2 : ℕ) : 𝒪_ℂ_[2])}) 𝒪_ℂ_[2]) (a₁ : Algebra ℤ_[2] 𝒪_ℂ_[2]) (a₂ : Algebra ℤ_[2] ℂ_[2]) (_ : @IsScalarTower ℤ_[2] 𝒪_ℂ_[2] ℂ_[2] a₁.toSMul Algebra.toSMul a₂.toSMul) (P : Point 2 (Valued.v : Valuation ℂ_[2] ℝ≥0) 𝒪_ℂ_[2] GaussianInt) (σ : Fin 2 → GaussianInt →+* 𝒪_ℂ_[2]), P.A.g = 2 ∧ Module.finrank ℂ_[2] P.V = 4 ∧ IsSplitting (embC ℂ_[2] σ) ∧ (σ 0).comp (starRingEnd _) = σ 1 ∧ σ 0 ≠ σ 1 ∧ ∀ i, P.eigV (σ i) ≠ ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsPullOfMoveInv

theorem **pull_zero** (d : DeltaDat Xo c 0) : Pull d 0 0

theorem **pull_one** (d : DeltaDat Xo c 0) : Pull d 1 1

theorem **pull_add** {d : DeltaDat Xo c 0} {f f' : Rul Xo c} {g g' : Rul d.Xo₂ d.c₂} (h : Pull d f g) (h' : Pull d f' g') : Pull d (f + f') (g + g')

theorem **pull_mul** {d : DeltaDat Xo c 0} {f f' : Rul Xo c} {g g' : Rul d.Xo₂ d.c₂} (h : Pull d f g) (h' : Pull d f' g') : Pull d (f ∗ f') (g ∗ g')

theorem **pull_smul** {d : DeltaDat Xo c 0} {f : Rul Xo c} {g : Rul d.Xo₂ d.c₂} (h : Pull d f g) (a : c.F) : Pull d (a • f) (d.eF a • g)

theorem **pull_inj** {d : DeltaDat Xo c 0} (hd : d.IsLev) {f f' : Rul Xo c} {g : Rul d.Xo₂ d.c₂} (h : Pull d f g) (h' : Pull d f' g) (k : Type) [Field k] [IsAlgClosed k] [Algebra c.F k] (y : FrPt Xo c k) : evR Xo c f y = evR Xo c f' y

theorem **not_pull_zero_one** {d : DeltaDat Xo c 0} (hd : d.IsLev) (k : Type) [Field k] [IsAlgClosed k] [Algebra c.F k] (y : FrPt Xo c k) : ¬ Pull d 0 1

theorem **pull_id** (nab : NabV Xo c) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) : Pull (GlobalTheory.RawSpan.idCover nab 0) f f

theorem **exists_pull_of_moveInv** (d : DeltaDat Xo c 0) (hd : d.IsLev) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (g : Rul d.Xo₂ d.c₂) (hg : g ∈ sec d.Xo₂ d.c₂ a n) (hi : GlobalTheory.ThetaSumDescent.MoveInv d g) : ∃ f ∈ sec Xo c a n, Pull d f g

theorem **fibConst_of_moveInv** (d : DeltaDat Xo c 0) (hd : d.IsLev) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (g : Rul d.Xo₂ d.c₂) (hg : g ∈ sec d.Xo₂ d.c₂ a n) (hi : GlobalTheory.ThetaSumDescent.MoveInv d g) : FibConst d g

theorem **intAt_zero** : IntAt Xo c 0

theorem **intAt_add** {f g : Rul Xo c} (hf : IntAt Xo c f) (hg : IntAt Xo c g) : IntAt Xo c (f + g)

theorem **intAt_mul** {f g : Rul Xo c} (hf : IntAt Xo c f) (hg : IntAt Xo c g) : IntAt Xo c (f ∗ g)

theorem **laQ_eq** : LaQ Dw l = LevQ Dw l

theorem **exists_isLift** : ∃ lift : Gam Dw l → Dw.La →ₗ[𝒪 E] Dw.La, IsLift lift

theorem **congMod_iff** (h h' : Dw.La →ₗ[𝒪 E] Dw.La) : CongMod l.Mb h h' ↔ ∀ a, (Submodule.Quotient.mk (h a) : LaQ Dw l) = Submodule.Quotient.mk (h' a)

theorem **lift_mk** {lift : Gam Dw l → Dw.La →ₗ[𝒪 E] Dw.La} (h : IsLift lift) (γ : Gam Dw l) (a : Dw.La) : (Submodule.Quotient.mk (lift γ a) : LaQ Dw l) = γ (Submodule.Quotient.mk a)

theorem **lift_unique** {lift lift' : Gam Dw l → Dw.La →ₗ[𝒪 E] Dw.La} (h : IsLift lift) (h' : IsLift lift') (γ : Gam Dw l) : CongMod l.Mb (lift γ) (lift' γ)

theorem **lift_mul** {lift : Gam Dw l → Dw.La →ₗ[𝒪 E] Dw.La} (h : IsLift lift) (γ γ' : Gam Dw l) : CongMod l.Mb (lift (γ ∗ γ')) (lift γ ∘ₗ lift γ')

theorem **eq_of_congMod** {lift : Gam Dw l → Dw.La →ₗ[𝒪 E] Dw.La} (h : IsLift lift) {γ γ' : Gam Dw l} (e : CongMod l.Mb (lift γ) (lift γ')) : γ = γ'

theorem **mb_ne_bot** : l.Mb ≠ ⊥

theorem **finite_laQ** : Finite (LaQ Dw l)

theorem **finite_gam** : Finite (Gam Dw l)

theorem **mb_ne_top** : l.Mb ≠ ⊤

theorem **nontrivial_laQ** (hr : 1 ≤ Dw.p + Dw.q) : Nontrivial (LaQ Dw l)

theorem **nontrivial_gam** (hr : 1 ≤ Dw.p + Dw.q) : Nontrivial (Gam Dw l)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsRatPointOfFiniteType

theorem **mem_ratHull** {R : Subalgebra o Ω} {x : Ω} : x ∈ ratHull R ↔ ∃ n : ℤ, n ≠ 0 ∧ (n : Ω) ∗ x ∈ R

theorem **le_ratHull** (R : Subalgebra o Ω) : R ≤ ratHull R

theorem **ratHull_mono** {R R' : Subalgebra o Ω} (h : R ≤ R') : ratHull R ≤ ratHull R'

theorem **isUnit_intCast_ratHull** [CharZero Ω] (R : Subalgebra o Ω) {n : ℤ} (hn : n ≠ 0) : IsUnit (n : ratHull R)

theorem **isLocalization_ratHull** [CharZero Ω] (R : Subalgebra o Ω) : letI

theorem **existsUnique_ratLift** [CharZero Ω] (R : Subalgebra o Ω) {K : Type} [CommRing K] (hK : ∀ n : ℤ, n ≠ 0 → IsUnit (n : K)) (f : R →+* K) : ∃! g : ratHull R →+* K, g.comp (toRat R) = f

theorem **existsUnique_ratPoint** [CharZero Ω] (R : Subalgebra o Ω) {K : Type} [Field K] [CharZero K] (x : Sp K ⟶ Sp R) : ∃! x' : Sp K ⟶ Sp (ratHull R), x' ≫ spMap (toRat R) = x

theorem **spMap_val_toRat** (R : Subalgebra o Ω) : spMap (ratHull R).val.toRingHom ≫ spMap (toRat R) = spMap R.val.toRingHom

theorem **exists_ratPoint_of_finiteType** {X : Scheme.{0}} (s : X ⟶ Sp ℚ) [LocallyOfFiniteType s] (x : X) : ∃ (Ω : Type) (_ : Field Ω) (_ : ArithBase ℤ Ω) (R : Subalgebra ℤ Ω), R.FG ∧ (∀ z : Ω, ∃ a ∈ R, ∃ b ∈ R, z = a / b) ∧ ∃ u : Sp (ratHull R) ⟶ X, ∀ w, (spMap (ratHull R).val.toRingHom ≫ u) w = x

theorem **UOver.comp** (h : UOver Xo c P σ u) {R'' : Type} [CommRing R''] (τ : R' →+* R'') : UOver Xo c P (τ.comp σ) (spMap τ ≫ u)

theorem **uOver_baseChange_iff** {R₁ : Type} [CommRing R₁] {P₁ : PolAV (Sp R₁)} {τ : R →+* R₁} {φ : P₁.A.left ⟶ P.A.left} (hφ : IsGrpPullback P₁.A P.A (spMap τ) φ) (hL : LBIso P₁.L (P.L.pull φ)) (σ₁ : R₁ →+* R') : UOver Xo c P₁ σ₁ u ↔ UOver Xo c P (σ₁.comp τ) u

theorem **exists_pol_baseChange** {S S' : Scheme.{0}} (P : PolAV S) (f : S' ⟶ S) : ∃ (P' : PolAV S') (φ : P'.A.left ⟶ P.A.left), IsGrpPullback P'.A P.A f φ ∧ LBIso P'.L (P.L.pull φ)

theorem **exists_uOver_id** (u : Sp R' ⟶ Xo.XF c) : ∃ P : PolAV (Sp R'), UOver Xo c P (RingHom.id R') u

theorem **exists_uOver_finiteType** [IsDomain R'] [IsLocalRing R'] (u : Sp R' ⟶ Xo.XF c) : ∃ (R : Subring R') (P : PolAV (Sp R)), Algebra.FiniteType ℤ R ∧ UOver Xo c P R.subtype u

theorem **exists_ratPoint** (Xo : Moduli D l) (c : Coeff l) (x : Xo.XF c) : ∃ (Ω : Type) (_ : Field Ω) (_ : ArithBase ℤ Ω) (R : Subalgebra ℤ Ω), R.FG ∧ (∀ z : Ω, ∃ a ∈ R, ∃ b ∈ R, z = a / b) ∧ ∃ u : Sp (ratHull R) ⟶ Xo.XF c, ∀ w, (spMap (ratHull R).val.toRingHom ≫ u) w = x

theorem **exists_spread_grp** {B Ω : Type} [CommRing B] [IsDomain B] [Field Ω] [Algebra B Ω] [IsFractionRing B Ω] {G₁ G₂ : Over (Sp B)} [GrpObj G₁] [GrpObj G₂] (a₁ : IsAbelianScheme G₁) (a₂ : IsAbelianScheme G₂) {G : Over (Sp Ω)} [GrpObj G] {α₁ : G.left ⟶ G₁.left} {α₂ : G.left ⟶ G₂.left} (h₁ : IsGrpPullback G G₁ (spMap (algebraMap B Ω)) α₁) (h₂ : IsGrpPullback G G₂ (spMap (algebraMap B Ω)) α₂) : ∃ a : B, a ≠ 0 ∧ ∀ (B' : Type) [CommRing B'] (g : B →+* B') (j : B' →+* Ω), IsUnit (g a) → Function.Injective j → j.comp g = algebraMap B Ω → ∃ (β : pullback G₁.hom (spMap g) ⟶ G₂.left) (φ : G.left ⟶ pullback G₁.hom (spMap g)), IsGrpPullback ((Over.pullback (spMap g)).obj G₁) G₂ (spMap g) β ∧ IsGrpPullback G ((Over.pullback (spMap g)).obj G₁) (spMap j) φ ∧ φ ≫ pullback.fst _ _ = α₁ ∧ φ ≫ β = α₂

theorem **exists_le_isUnit** {R : Subalgebra o Ω} (hR : R.FG) {a : ratHull R} (ha : a ≠ 0) : ∃ (R' : Subalgebra o Ω) (h : R ≤ R'), R'.FG ∧ IsUnit ((Subalgebra.inclusion (ratHull_mono h)).toRingHom a)

theorem **UOver.exists_spread** {Xo : Moduli D l} {c : Coeff l} {R : Subalgebra o Ω} (hR : R.FG) (hfr : ∀ z : Ω, ∃ a ∈ R, ∃ b ∈ R, z = a / b) {P : PolAV (Sp R)} {u : Sp (ratHull R) ⟶ Xo.XF c} (h : UOver Xo c P R.val.toRingHom (spMap (ratHull R).val.toRingHom ≫ u)) : ∃ (R' : Subalgebra o Ω) (hle : R ≤ R'), R'.FG ∧ UOver Xo c P ((toRat R').comp (Subalgebra.inclusion hle).toRingHom) (spMap (Subalgebra.inclusion (ratHull_mono hle)).toRingHom ≫ u)

theorem **exists_uOver_ratPoint** (Xo : Moduli D l) (c : Coeff l) (x : Xo.XF c) : ∃ (Ω : Type) (_ : Field Ω) (_ : ArithBase ℤ Ω) (R : Subalgebra ℤ Ω), R.FG ∧ (∀ z : Ω, ∃ a ∈ R, ∃ b ∈ R, z = a / b) ∧ ∃ (P : PolAV (Sp R)) (u : Sp (ratHull R) ⟶ Xo.XF c), UOver Xo c P (toRat R) u ∧ ∀ w, (spMap (ratHull R).val.toRingHom ≫ u) w = x

theorem **UCompact.isFractionRing** {Xo : Moduli D l} {c : Coeff l} (C : UCompact Xo c) : letI

theorem **UCompact.isPreimmersion_i** {Xo : Moduli D l} {c : Coeff l} (C : UCompact Xo c) : IsPreimmersion C.i

theorem **UCompact.exists_odd_prime_isUnit** {Xo : Moduli D l} {c : Coeff l} (C : UCompact Xo c) : ∃ n : ℕ, n.Prime ∧ Odd n ∧ IsUnit (n : C.o)

theorem **USetup.pulled_uΩ** : UOver Xo c X₁.ρ.P X₁.S.R.val.toRingHom X₁.uΩ

theorem **USetup.hasDiv** : X₁.M.HasDiv X₁.S.th

theorem **USetup.existsUnique_lies** {K : Type} [Field K] [CharZero K] (x : Sp K ⟶ Sp X₁.S.R) : ∃! y, X₁.Lies x y

theorem **USetup.lies_uΩ** : X₁.Lies (spMap X₁.S.R.val.toRingHom) X₁.uΩ

theorem **USetup.Lies.uOver** {X₁ : USetup C} {K : Type} [CommRing K] {f : X₁.S.R →+* K} {y : Sp K ⟶ Xo.XF c} (h : X₁.Lies (spMap f) y) : UOver Xo c X₁.ρ.P f y

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsResolProjSpace

theorem **posDef_gram** (σ : F →+* ℂ) : (f.gram σ).PosDef

theorem **form_gram** (σ : F →+* ℂ) (x : Fin f.N → ℂ) : star x ⬝ᵥ f.gram σ *ᵥ x = ((frameNorm (f.A σ) x ^ 2 : ℝ) : ℂ)

theorem **nrm_fam** (σ : F →+* ℂ) (x : V) : f.fam.nrm σ x = Nm σ x

theorem **isConjInv_fam** : f.fam.IsConjInv

theorem **isFull** [IsScalarTower (𝒪 F) F V] : IsFull M

theorem **restr_pull** (hB : B.IsLB) {T : Scheme.{0}} (r : T ⟶ Y) {O O' : T.Opens} (h : O ≤ r ⁻¹ᵁ U) (h' : O' ≤ O) (f : H0 p B U) : restr h' (H0.pull r h f) = H0.pull r (h'.trans h) f

theorem **off_eq** : M.off = M.σ ⁻¹ᵁ U

theorem **bijective_pullOff** (hB : B.IsLB) : Function.Bijective (M.pullOff (B

theorem **notMem_support** (h : NoBase hB W U) : ∀ z ∈ M.off, z ∉ ((baseIdeal hB W).comap M.σ).support

theorem **lift_injective** [IsReduced Y] (hU : Dense (U : Set Y)) : Function.Injective Δ.lift

theorem **ofSec_comb** (abelian₂ : ℝ) (u : Fin f.N → F) : H0.ofSec M.str ⊤ ((M.sys Δ φ f abelian₂).comb u) = Δ.lift (φ (f.b.equivFun.symm u))

theorem **isAlgebraicData** [IsReduced Y] (hU : Dense (U : Set Y)) (hc : GeometricallyConnected M.str) (hD : 1 ≤ D) (abelian₂ : ℝ) : (M.sys Δ φ f abelian₂).IsAlgebraicData

theorem **bd_ne_top** (hc : GeometricallyConnected M.str) : M.bd ≠ ⊤

theorem **ger_e** (t : R.Δ₂.P.ι × C.J × R.Δ₁.P.ι) (z : R.X) (h₂ : z ∈ R.Δ₂.P.U t.1) (h₁ : z ∈ R.Δ₁.P.U t.2.2) (h) : ger z h (R.e.f ((R.com.effChart R.Δ₁.isLB R.Δ₂.isLB rfl).c t)) = ger z h₂ (R.Δ₂.t.f t.1) * ger z h₁ (R.Δ₁.t.f t.2.2) ^ D

theorem **off_le_nonzeroLocus_e** (h₁ : NoBase (C.isLB a₁) W₁ U) (h₂ : NoBase (C.isLB a₂) W₂ U) : R.off ≤ nonzeroLocus R.e

theorem **mulPole_adjEquiv** (hω : NowhereZero ω) (h₁ : NoBase (C.isLB a₁) W₁ U) (j : ℕ) {s : Λ} (h : κ + (a₂ + D • a₁ + j • a₁) = s) (m : H0 R.str (R.pb s) R.off) : R.mulPole j R.off (R.adjEquiv ω hω h₁ j h m) = R.mulCan ω h m

theorem **germ_adjEquiv** (hω : NowhereZero ω) (h₁ : NoBase (C.isLB a₁) W₁ U) (j : ℕ) {s : Λ} (h : κ + (a₂ + D • a₁ + j • a₁) = s) (m : H0 R.str (R.pb s) R.off) (t : R.K.bundle.ι × C.J × R.Δ₁.P.ι) (z : R.X) (hz : z ∈ R.off) (hP : z ∈ R.Δ₁.P.U t.2.2) (ha : z ∈ (R.adj j).U (R.adjIdx j t)) (hω' : z ∈ (LB.div R.K.bundle (R.pb κ)).U (t.1, R.com.idx κ t.2.1)) (hs : z ∈ (R.pb s).U (R.com.idx s t.2.1)) : ger z hP (R.Δ₁.t.f t.2.2) ^ j * germ (R.adjIdx j t) z hz ha (R.adjEquiv ω hω h₁ j h m) = germ (B

theorem **adjEquiv_mul** (hω : NowhereZero ω) (h₁ : NoBase (C.isLB a₁) W₁ U) (j : ℕ) {s s' : Λ} (h : κ + (a₂ + D • a₁ + j • a₁) = s) (h' : κ + (a₂ + D • a₁ + (j + 1) • a₁) = s') (hs : a₁ + s = s') (w : W₁) (m : H0 R.str (R.pb s) R.off) : R.adjEquiv ω hω h₁ (j + 1) h'

((R.com.addChart hs).mul (restr le_top (H0.pull R.σ le_top w.1)) m) = (oneChart (isLB_tens R.K.isLB (R.com.isLB _)) R.Δ₁.isLB_moving (Nat.cast_succ j).symm).mul (restr le_top (R.Δ₁.lift w)) (R.adjEquiv ω hω h₁ j h m)

theorem **rangeIf_le** : rangeIf ψ q ≤ LinearMap.range ψ

theorem **apply_invIf** (x : rangeIf ψ q) : ψ (invIf ψ hψ q x) = x

theorem **invIf_injective** : Function.Injective (invIf ψ hψ q)

theorem **isFull_latIf** : IsFull (latIf ψ hψ q M)

theorem **exists_famIf** : ∃ g : HermFam F (rangeIf ψ q), g.IsConjInv ∧ (q → ∀ σ x, g.nrm σ x = Nm σ (invIf ψ hψ q x))

theorem **coe_dataEquiv** {j : ℕ} (hj : j ≤ T) (s : S j) : (dataEquiv G T hT ψ hψ hM hle h0 k hj s).1 = ψ j s

theorem **dataEquiv_mem** {j : ℕ} (hj : j ≤ T) (s : S j) : dataEquiv G T hT ψ hψ hM hle h0 k hj s ∈ (dataOf G T hT ψ hψ hM hle h0 k).M j ↔ s ∈ M j

theorem **nrm_dataEquiv** {j : ℕ} (hj : j ≤ T) (σ : F →+* ℂ) (s : S j) : ((dataOf G T hT ψ hψ hM hle h0 k).nm j).nrm σ (dataEquiv G T hT ψ hψ hM hle h0 k hj s) = Nm j σ s

theorem **exists_resol_projSpace** (F : Type) [Field F] [NumberField F] {M : ℕ} (hM : 1 ≤ M) : ∃ (W₁ : Submodule F (H0 (PSpace.str F M) (lbZPow (ProjSpace.O1 F M) 1) ⊤)) (W₂ : Submodule F (H0 (PSpace.str F M) (lbZPow (ProjSpace.O1 F M) 2) ⊤)) (R : Resol (Common.powers (ProjSpace.O1_isLB F M)) (ProjSpace.chart F M (Fin.last M)) M W₁ W₂), nonzeroLocus R.Δ₁.t = R.opn ∧ nonzeroLocus R.Δ₂.t = R.opn ∧ R.opn ≠ ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsRetractionMulP

theorem **isInv_bot** {O A : Type} [CommRing O] [CommRing A] [HopfAlgebra O A] (a : A) : IsInv (⊥ : Subgroup (WithConv (A →ₐ[O] O))) a

theorem **exists_retraction_mulP** {p : ℕ} {O : Type} [CommRing O] (G : PDivGroup p O) (m : ℕ) : ∃ ρ : G.A (m + 1) →ₗ[O] G.A m, ∀ a, ρ (G.mulP m a) = a

theorem **QuotTower.kerIdeal_full** {p : ℕ} {O : Type} [CommRing O] (G : PDivGroup p O) (m : ℕ) : kerIdeal ((QuotTower.full G).j m) = torsionIdeal O (G.A (m + 1)) p

theorem **QuotTower.map_res_kerIdeal** {p : ℕ} {O : Type} [CommRing O] {G : PDivGroup p O} (T : QuotTower G) (m : ℕ) : Ideal.map (G.res (m + 1) : G.A (m + 2) →ₐ[O] G.A (m + 1)) (kerIdeal (T.j (m + 1))) = kerIdeal (T.j m)

theorem **QuotTower.kerIdeal_full_mu_ne** (p : ℕ) [Fact p.Prime] (O : Type) [CommRing O] [Nontrivial O] : kerIdeal ((QuotTower.full (PDivGroup.mu p O)).j 0) ≠ augIdeal O ((PDivGroup.mu p O).A 1)

theorem **ptIdeal_bot** {O A : Type} [CommRing O] [CommRing A] [HopfAlgebra O A] : ptIdeal (⊥ : Subgroup (WithConv (A →ₐ[O] O))) = augIdeal O A

theorem **isInv_algebraMap** {O A : Type} [CommRing O] [CommRing A] [HopfAlgebra O A] (Q : Subgroup (WithConv (A →ₐ[O] O))) (c : O) : IsInv Q (algebraMap O A c)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsRootSelectionUnitaryRankOne

theorem **conj_mul_star** (σ : S →+* Sᵐᵒᵖ) (b : S) (x y : Sˣ) (t : Aˣ) (s : A) (hx : (x : S) * b * (σ x).unop = algebraMap A S t) (hy : (y : S) * b * (σ y).unop = algebraMap A S s * b) : ((x * y * x⁻¹ : Sˣ) : S) * (σ ((x * y * x⁻¹ : Sˣ) : S)).unop = algebraMap A S s

theorem **scal_val** (G : HTGroup p) (a : G.Fˣ) : ((scal G a : G.Cˣ) : G.C) = algebraMap G.F G.C a

theorem **sh_algebraMap** (G : HTGroup p) (x : G.F) : sh G (algebraMap G.F G.C x) = algebraMap G.F G.C (IsCMField.complexConj G.F x)

theorem **sh_one_eq** (G : HTGroup p) : sh G 1 = 1

theorem **Pair.ofScalar_pol** (G : HTGroup p) (hn : G.n = 1) (a : G.Fˣ) (c : ℚˣ) (h : (a : G.F) * IsCMField.complexConj G.F a = algebraMap ℚ G.F c) : (Pair.ofScalar G hn a c h).pol = Polynomial.X - Polynomial.C (a : G.F)

theorem **exists_isSel** (G : HTGroup p) (h : G.a ≤ 1) : ∃ sel, IsSel G sel

theorem **exists_isSel_of_isSimple** (G : HTGroup p) (h : IsSimple G) : ∃ sel, IsSel G sel

theorem **isTest_indicator** {H : Type} [Group H] (θ : H →* H) (K : Subgroup H) (hK : K.map θ ≤ K) : IsTest θ K K (Set.indicator (K : Set H) 1)

theorem **match_refl** (G : HTGroup p) (K : Subgroup (Gr G 1)) (φ : Gr G 1 → ℂ) : Match G 1 K φ K φ

theorem **conj_gi** : IsCMField.complexConj QI gi = -gi

theorem **two_add_gi_ne_zero** : (2 + gi : QI) ≠ 0

theorem **norm_gaussian_element_eq_five** : ((a21 : QI)) * IsCMField.complexConj QI a21 = algebraMap ℚ QI ((Units.mk0 (5 : ℚ) (by norm_num) : ℚˣ) : ℚ)

theorem **isSimple_unitaryRankOne** : IsSimple (ex 1 one_ne_zero)

theorem **exPair_mark** : ∃ μ : ℂ, exPair.Mark μ

theorem **exists_rootSelection_unitaryRankOne** : ∃ sel, IsSel (ex 1 one_ne_zero) sel

theorem **exists_cls_mark** : ∃ q : Cls (ex 1 one_ne_zero), ∃ P : Pair (ex 1 one_ne_zero), Quot.mk _ P = q ∧ ∃ μ, P.Mark μ

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsTensElliptic

theorem **comp_tensφ** {S : Scheme.{0}} {B B' T : Over S} [GrpObj B] {r : ℕ} (pr : Fin r → (B' ⟶ B)) (x : T ⟶ B') (k : (Fin r × Fin r) × Fin 4) : x.left ≫ tensφ pr k = lamPts (x ≫ pr k.1.1) (x ≫ pr k.1.2) k.2

theorem **formula_trans** (D : DegPre R) {r : ℕ} (W : Matrix (Fin r) (Fin r) ℤ) {B' : Over (Sp R)} {L' : LB B'.left} (pr : Fin r → (B' ⟶ D.P.A)) {w : LB.Iso L' (LB.LinComb D.P.L (tensφ pr) (tensε W))} (hw : w.IsGen) (x : TK R ⟶ B') (y : Fin r → D.X) (hx : ∀ i, x ≫ pr i = toUnit (TK R) ≫ D.c (y i)) (z : (TK R).left) (J J₀) (hJ : Good D.P.L z J) (hJ₀ : Good D.P.L z J₀) (k k₀ : L'.ι) (hk : x.left z ∈ L'.U k) (hk₀ : x.left z ∈ L'.U k₀) : gm x.left z _ (w.s (fun a => J (x.left ≫ tensφ pr a), k)) * ∏ i, ∏ j, val (D.tau (y i) (y j)) (𝟙 _) z J ^ W i j = (gm x.left z _ (L'.g k k₀))⁻¹ * (gm x.left z _ (w.s (fun a => J₀ (x.left ≫ tensφ pr a), k₀)) * ∏ i, ∏ j, val (D.tau (y i) (y j)) (𝟙 _) z J₀ ^ W i j)

theorem **exists_psi** (D : DegPre R) {r : ℕ} (W : Matrix (Fin r) (Fin r) ℤ) {B' : Over (Sp R)} {L' : LB B'.left} (hL : L'.IsLB) (pr : Fin r → (B' ⟶ D.P.A)) {w : LB.Iso L' (LB.LinComb D.P.L (tensφ pr) (tensε W))} (hw : w.IsGen) (x : TK R ⟶ B') (y : Fin r → D.X) (hx : ∀ i, x ≫ pr i = toUnit (TK R) ≫ D.c (y i)) : ∃ t : LB.Gen (LB.LinComb L' (K

theorem **exists_tens** [IsLocalRing R] [IsNoetherianRing R] (D : DegPre R) {r : ℕ} (W : Matrix (Fin r) (Fin r) ℤ) (hp : ((W + W.transpose).map (Int.cast : ℤ → ℝ)).PosDef) (hd : IsUnit (W + W.transpose).det) : ∃ D' : DegPre R, D'.P.g = r ∗ D.P.g ∧ Nonempty (Tens D D' W)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsTjCorrDiagModel

theorem **p_mem_uIdeal** (D : Datum p) (r : ℤ) : (p : 𝒪 D.F) ∈ uIdeal D r

theorem **sub_mem_uIdeal** (D : Datum p) (r : ℤ) : D.ω − (r : 𝒪 D.F) ∈ uIdeal D r

theorem **exists_prime_uIdeal_le** (D : Datum p) : ∃ (r : ℤ) (w : Ideal (𝒪 D.F)), w.IsMaximal ∧ uIdeal D r ≤ w

theorem **IsPIsog.exists_eq_pow** {N : ℕ} {s : S ⟶ Sp R} {c : ℕ} {o o' : Obj b N s} {ψ : o.A ⟶ o'.A} (h : IsPIsog c o o' ψ) : ∃ k : ℕ, c = p ^ k

theorem **IsPIsog.pos** {N : ℕ} {s : S ⟶ Sp R} {c : ℕ} {o o' : Obj b N s} {ψ : o.A ⟶ o'.A} (h : IsPIsog c o o' ψ) : 0 < c

theorem **TjCorr.p_mem** {N : ℕ} {F : RepFam b N} {r : ℤ} {w : Ideal (𝒪 D.F)} {j c : ℕ} (T : TjCorr F r w j c) : (p : 𝒪 D.F) ∈ w

theorem **TjCorr.w_ne_top** {N : ℕ} {F : RepFam b N} {r : ℤ} {w : Ideal (𝒪 D.F)} {j c : ℕ} (T : TjCorr F r w j c) : w ≠ ⊤

theorem **exists_charZero_point** (b : Base D R) {Y : Scheme.{0}} (s : Y ⟶ Sp R) (hp : ∀ (U : Y.Opens) (f : Γ(Y, U)), (p : Γ(Y, U)) ∗ f = 0 → f = 0) (U : Y.Opens) (y : Y) (hy : y ∈ U) : ∃ (k : Type) (_ : Field k) (_ : IsAlgClosed k) (_ : CharZero k) (x : Sp k ⟶ Y), ∀ q, x q ∈ U

theorem **TjCorr.exists_charZero_point** {N : ℕ} {F : RepFam b N} {r : ℤ} {w : Ideal (𝒪 D.F)} {j c : ℕ} (T : TjCorr F r w j c) (U : T.Y.Opens) (y : T.Y) (hy : y ∈ U) : ∃ (k : Type) (_ : Field k) (_ : IsAlgClosed k) (_ : CharZero k) (x : Sp k ⟶ T.Y), ∀ q, x q ∈ U

theorem **TjCorr.eq_of_isTj** {N : ℕ} {F : RepFam b N} [Nonempty F.X] {r : ℤ} {w : Ideal (𝒪 D.F)} {j c j' : ℕ} (T : TjCorr F r w j c) (h : IsTj r w j' T.o T.o' T.ψ) : j' = j

theorem **IsPIsog.id** {N : ℕ} {s : S ⟶ Sp R} (o : Obj b N s) : IsPIsog 1 o o (𝟙 o.A)

theorem **KilledBy.one** {N : ℕ} {s : S ⟶ Sp R} (o : Obj b N s) (v : Ideal (𝒪 D.F)) (T : Over S) : KilledBy o v (1 : T ⟶ o.A)

theorem **IsTj.id** {N : ℕ} {s : S ⟶ Sp R} (r : ℤ) (w : Ideal (𝒪 D.F)) (o : Obj b N s) : IsTj r w 0 o o (𝟙 o.A)

theorem **exists_tjCorr_diag** {N : ℕ} (F : RepFam b N) (r : ℤ) (w : Ideal (𝒪 D.F)) (hw : w.IsPrime) (hu : uIdeal D r ≤ w) (hp : ∀ (U : F.X.Opens) (f : Γ(F.X, U)), (p : Γ(F.X, U)) ∗ f = 0 → f = 0) : ∃ T : TjCorr F r w 0 1, IsIso T.s ∧ IsIso T.t ∧ IsIso T.ψ

theorem **pReg_of_flat** {Y : Scheme.{0}} (s : Y ⟶ Sp R) [Flat s] (hR : ∀ a : R, (p : R) ∗ a = 0 → a = 0) (U : Y.Opens) (f : Γ(Y, U)) (hf : (p : Γ(Y, U)) ∗ f = 0) : f = 0

theorem **pReg_of_isUnit** {Y : Scheme.{0}} (s : Y ⟶ Sp R) (hR : IsUnit (p : R)) (U : Y.Opens) (f : Γ(Y, U)) (hf : (p : Γ(Y, U)) ∗ f = 0) : f = 0

theorem **exists_tjCorr_diag_model** {N : ℕ} (M : Model b N) (hR : ∀ a : R, (p : R) ∗ a = 0 → a = 0) : ∃ (r : ℤ) (w : Ideal (𝒪 D.F)) (T : TjCorr M.toRepFam r w 0 1), IsIso T.s ∧ IsIso T.t ∧ IsIso T.ψ

theorem **isWOrd_iff** {N : ℕ} {s : S ⟶ Sp R} (o : Obj b N s) {r : ℤ} {w : Ideal (𝒪 D.F)} (hw : w.IsPrime) (hu : uIdeal D r ≤ w) {k : Type} [Field k] [IsAlgClosed k] [CharP k p] (x : Sp k ⟶ S) : IsWOrd o w x ↔ Nat.card {g : Over.mk x ⟶ o.A // KilledBy o w g} = Ideal.absNorm w ^ (D.m ∗ (D.n − 1))

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsToricRankOne

theorem **Modif.exists_not_isIso** {R : Type} [CommRing R] [IsDomain R] (h : ¬ IsIntegrallyClosed R) : ∃ M : Modif R, ¬ IsIso M.π

theorem **Modif.exists_nontrivial** : ∃ (R : Type) (_ : CommRing R) (_ : IsDomain R) (M : Modif R), ¬ IsIso M.π

theorem **Has.labCover** (hR : Base R) {D : DegData R (maximalIdeal R)} (h : Has D) (n : ℕ) (hn : n.Prime) (ho : Odd n) (hu : IsUnit (n : R)) : ∃ C : LabCover D, C.toModif = Modif.self R ∧ C.n = n

theorem **Has.hasFin** (hR : Base R) {D : DegData R (maximalIdeal R)} (h : Has D) : HasFin D

theorem **base_powerSeries** : Base (PowerSeries ℚ)

theorem **LabCover.exists_toric_rank_one** : ∃ D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ)), Module.finrank ℤ D.toPre.X = 1 ∧ Has D ∧ HasFin D ∧ ∃ C : LabCover D, C.n = 3

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsToricRankOneBis

theorem **sval_zeroT** (D : DegPre R) (a : D.P.A ⟶ D.P.A) (n j : ℤ) (β : D.X) {Z : Scheme.{0}} (x : Z ⟶ D.P.A.left) (z : Z) (J) (hJ : Good D.P.L z J) : sval (zeroT D a n j β) x z J = 0

theorem **zero_small_tfe** (I : Ideal R) (D : DegPre R) (a : D.P.A ⟶ D.P.A) (h h' : D.X ⟶+ D.X) (n j : ℤ) : Small I (w

theorem **Small.hasAdicSum_iff** {I : Ideal R} {D : DegPre R} {a : D.P.A ⟶ D.P.A} {n j : ℤ} {Ix : Type} {w : Ix → D.X} {s : ∀ i, SecT D a n j (w i)} (hs : Small I s) (F : ∀ δ, SecT D a n j δ) : HasAdicSum I s F ↔ IsSum I s F

theorem **Small.expandsIn_iff** (U : UnifPre I D A G) {e : ℕ} {b : thickObj R I e G ⟶ thickObj R I e G} {a : D.toPre.P.A ⟶ D.toPre.P.A} {n j : ℤ} (τ : SecE U b n j) {Ix : Type} {w : Ix → D.toPre.X} {s : ∀ i, SecT D.toPre a n j (w i)} (hs : Small I s) : ExpandsIn U τ s ↔ TExp U τ s

theorem **HasAdicSum.small** {I : Ideal R} {D : DegPre R} {a : D.P.A ⟶ D.P.A} {n j : ℤ} {Ix : Type} {w : Ix → D.X} {s : ∀ i, SecT D a n j (w i)} {F : ∀ δ, SecT D a n j δ} (H : HasAdicSum I s F) : Small I (w

theorem **HasAdicSum.unique** {I : Ideal R} {D : DegPre R} {a : D.P.A ⟶ D.P.A} {n j : ℤ} {Ix : Type} {w : Ix → D.X} {s : ∀ i, SecT D a n j (w i)} {F F' : ∀ δ, SecT D a n j δ} (H : HasAdicSum I s F) (H' : HasAdicSum I s F') (e : ℕ) (δ : D.X) (Z : Scheme.{0}) (x : Z ⟶ (thickObj R I e D.P.A).left) (z : Z) (J) (hJ : Good D.P.L z J) : sval (F δ) (x ≫ pullback.fst D.P.A.hom (thick R I e)) z J = sval (F' δ) (x ≫ pullback.fst D.P.A.hom (thick R I e)) z J

theorem **Small.hasAdicSum_self** {I : Ideal R} {D : DegPre R} {a : D.P.A ⟶ D.P.A} {n j : ℤ} {f : ∀ β, SecT D a n j β} (hf : Small I (w

theorem **HasAdicSum.not_const** {I : Ideal R} {D : DegPre R} {a : D.P.A ⟶ D.P.A} {n j : ℤ} {β : D.X} {s : SecT D a n j β} {e : ℕ} (hs : ¬ Van I s e) (F : ∀ δ, SecT D a n j δ) : ¬ HasAdicSum I (w

theorem **ExpandsIn.not_const** (U : UnifPre I D A G) {e : ℕ} {b : thickObj R I e G ⟶ thickObj R I e G} {a : D.toPre.P.A ⟶ D.toPre.P.A} {n j : ℤ} (τ : SecE U b n j) {β : D.toPre.X} {s : SecT D.toPre a n j β} (hs : ¬ Van I s e) : ¬ ExpandsIn U τ (w

theorem **Van.iff_chart** (I : Ideal R) {D : DegPre R} {K : Type} [Fintype K] {φ : K → (D.P.A.left ⟶ D.P.A.left)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb D.P.L φ ε)) (e : ℕ) : Van I s e ↔ ∀ j : K → D.P.L.ι, ((pullback.fst D.P.A.hom (thick R I e)).app _).hom (s.f j) = 0

theorem **Van.iff_pull** (U : UnifPre I D A G) {K : Type} [Fintype K] {φ : K → (D.toPre.P.A.left ⟶ D.toPre.P.A.left)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb D.toPre.P.L φ ε)) (e : ℕ) : Van I s e ↔ ∀ (Z : Scheme.{0}) (x : Z ⟶ (thickObj R I e G).left) (z : Z) (J), Good D.toPre.P.L z J → sval s (x ≫ (U.gt e).pG.left) z J = 0

theorem **ExpandsIn.iff_exists** (U : UnifPre I D A G) {e : ℕ} {b : thickObj R I e G ⟶ thickObj R I e G} {a : D.toPre.P.A ⟶ D.toPre.P.A} {n j : ℤ} (τ : SecE U b n j) {Ix : Type} {w : Ix → D.toPre.X} (s : ∀ i, SecT D.toPre a n j (w i)) : ExpandsIn U τ s ↔ ∃ S : Finset Ix, ∀ (Z : Scheme.{0}) (x : Z ⟶ (thickObj R I e G).left) (z : Z) (J), Good D.toPre.P.L z J → (∀ i ∉ S, sval (s i) (x ≫ (U.gt e).pG.left) z J = 0) ∧ ∀ J', Good U.LG.L z J' → evalE U τ x z J J' = ∑ i ∈ S, sval (s i) (x ≫ (U.gt e).pG.left) z J * (val ((U.gt e).t (w i)) x z J).val

theorem **ExpandsIn.iff_expandsAt** (U : UnifPre I D A G) (e k : ℕ) (b : thickObj R I e G ⟶ thickObj R I e G) (a : D.toPre.P.A ⟶ D.toPre.P.A) (σ : H0 U.LG.L k) (f : ∀ α, D.toPre.SecB k α) (τ : SecE U b k 0) (hτ : ∀ (Z : Scheme.{0}) (x : Z ⟶ _) z J', Good U.LG.L z J' → sval τ x z J' = sval σ (x ≫ (fstG I e G).left) z J') (s : ∀ α, SecT D.toPre a k 0 α) (hs : ∀ α (Z : Scheme.{0}) (x : Z ⟶ _) z J, Good D.toPre.P.L z J → sval (s α) x z J = sval (f α) x z J) : ExpandsIn U τ (w

theorem **formEq_delta** {D : DegPre R} (a : D.P.A ⟶ D.P.A) (k : ℕ) (mu β : D.X) : FormEq (dφ D a mu β (β + k • mu)) (dε k 0) (DegPre.deltaφ D (FractionRing R) k β mu) (DegPre.deltaε k)

theorem **of_integral** {I : Ideal R} {D : DegPre R} {k : ℕ} (F : Mum.IntFam I D k) (a : D.P.A ⟶ D.P.A) (h h' : D.X →+ D.X) : ∃ f : ∀ β, SecT D a k 0 β, (∀ β (Z : Scheme.{0}) (r : Z ⟶ D.P.A.left) (z : Z) (J), Good D.P.L z J → sval (f β) r z J = sval (F.f β) r z J) ∧ (∀ β e, Van I (f β) e ↔ Mum.VanMod I (F.f β) e) ∧ Small I (w

theorem **exists_secE** (U : UnifPre I D A G) {e k : ℕ} (b : thickObj R I e G ⟶ thickObj R I e G) (σ : H0 U.LG.L k) : ∃ τ : SecE U b k 0, ∀ (Z : Scheme.{0}) (x : Z ⟶ _) z J', Good U.LG.L z J' → sval τ x z J' = sval σ (x ≫ (fstG I e G).left) z J'

theorem **finite_of_expandsAt** (U : UnifPre I D A G) {e k : ℕ} {σ : H0 U.LG.L k} {g : ∀ α, D.toPre.SecB k α} (x : U.ExpandsAt e k σ g) : {α | ¬ Mum.VanMod I (g α) e}.Finite

theorem **exists_of_unif** {V : Type} [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] {D : DegData V (maximalIdeal V)} {A : PolAV (Sp (FractionRing V))} {G : Over (Sp V)} [GrpObj G] [IsCommMonObj G] (U : Unif (maximalIdeal V) D A G) (a : D.toPre.P.A ⟶ D.toPre.P.A) (h h' : D.toPre.X →+ D.toPre.X) : ∃ f : ∀ β, SecT D.toPre a (1 : ℕ) 0 β, HasAdicSum (maximalIdeal V) (w

theorem **exists_toric_rank_one** : ∃ (A : PolAV (Sp (FractionRing (PowerSeries ℚ)))) (G : Over (Sp (PowerSeries ℚ))) (_ : GrpObj G) (_ : IsCommMonObj G) (D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (U : Unif (maximalIdeal _) D A G), Module.finrank ℤ D.toPre.X = 1 ∧ ∃ f : ∀ β, SecT D.toPre (𝟙 _) (1 : ℕ) 0 β, HasAdicSum (maximalIdeal _) (w

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsTvalNeZero

theorem **mem_locLat_const** (g : Matrix ι ι K) (x : ι → K) : x ∈ locLat (fun _ : HeightOneSpectrum (𝒪 K) => g) ↔ ∀ i, (g *ᵥ x) i ∈ (algebraMap (𝒪 K) K).range

theorem **finite_not_le** (a : K) : {w : HeightOneSpectrum (𝒪 K) | ¬ w.valuation K a ≤ 1}.Finite

theorem **finite_not_intAt** (g : Matrix ι ι K) : {w : HeightOneSpectrum (𝒪 K) | ¬ IntAt w g}.Finite

theorem **isAdelic_const** {g : Matrix ι ι K} (hg : g.det ≠ 0) : IsAdelic (fun _ : HeightOneSpectrum (𝒪 K) => g)

theorem **isAdelic_single** (v : HeightOneSpectrum (𝒪 K)) {g : Matrix ι ι K} (hg : g.det ≠ 0) : IsAdelic (fun w : HeightOneSpectrum (𝒪 K) => if w = v then g else 1)

theorem **not_isAdelic_zero** [Nonempty ι] (v : HeightOneSpectrum (𝒪 K)) : ¬ IsAdelic (fun _ : HeightOneSpectrum (𝒪 K) => (0 : Matrix ι ι K))

theorem **detDeg_const** {g : Matrix ι ι K} (hg : g.det ≠ 0) : detDeg (fun _ : HeightOneSpectrum (𝒪 K) => g) (fun σ => g.map σ) = 0

theorem **matNorm_map** (g : Matrix ι ι K) (σ : K →+* ℂ) (x : ι → K) : matNorm (fun σ => g.map σ) σ x = matNorm (fun _ => 1) σ (g *ᵥ x)

theorem **detDeg_nat** {n : ℕ} (hn : n ≠ 0) : detDeg (fun _ : HeightOneSpectrum (𝒪 K) => !![(n : K)]) (fun _ => 1) = Real.log n

theorem **inv_mem** {n : ℕ} (hn : n ≠ 0) : ![(n : K)⁻¹] ∈ locLat (fun _ : HeightOneSpectrum (𝒪 K) => !![(n : K)])

theorem **rhoMat_one** (la : Fin n → ℕ) (K : Type) [Field K] [CharZero K] : rhoMat la (1 : Matrix _ _ K) = 1

theorem **rhoMat_mul** (la : Fin n → ℕ) {K : Type} [Field K] [CharZero K] (g h : Matrix (Fin n) (Fin n) K) : rhoMat la (g * h) = rhoMat la h * rhoMat la g

theorem **rhoMat_scalar** (la : Fin n → ℕ) {K : Type} [Field K] [CharZero K] (t : K) : rhoMat la (t • (1 : Matrix _ _ K)) = t ^ size la • 1

theorem **evalAt_eq_sum** (la : Fin n → ℕ) {K : Type} [Field K] [CharZero K] (y : Matrix (Fin n) (Fin n) K) (f : nabla la K) : evalAt y f.1 = ∑ i, (basis la K).repr f i * evalAt y (map (Int.castRingHom K) (intVec la i))

theorem **evalAt_mul** (la : Fin n → ℕ) {K : Type} [Field K] [CharZero K] (g y : Matrix (Fin n) (Fin n) K) (j : Idx la) : evalAt (g * y) (map (Int.castRingHom K) (intVec la j)) = ∑ i, rhoMat la g i j * evalAt y (map (Int.castRingHom K) (intVec la i))

theorem **tval_mul** (la : ∀ t, Fin (N t) → ℕ) {K : Type} [Field K] [CharZero K] (x : TIdx la → K) (g y : ∀ t, Matrix (Fin (N t)) (Fin (N t)) K) : tval la x (fun t => g t * y t) = tval la (fibA la 1 g *ᵥ x) y

theorem **kron_mul** {J : T → Type} [∀ t, Fintype (J t)] {R : Type} [CommRing R] (M M' : ∀ t, Matrix (J t) (J t) R) : kron M * kron M' = kron fun t => M t * M' t

theorem **onMat_mulVec** (la : Fin n → ℕ) (f : nabla la ℂ) : onMat la *ᵥ (basis la ℂ).repr f = (onBasis la).repr (toSpace la f)

theorem **sum_sq_onMat** (la : Fin n → ℕ) (X : Matrix (Fin n) (Fin n) ℂ) (f : nabla la ℂ) : ∑ i, ‖((onMat la * rhoMat la X) *ᵥ (basis la ℂ).repr f) i‖ ^ 2 = normSq la (Formalism.GradedWeylDivisibility.rowAct ℂ X f.1)

theorem **nonempty_tIdx** (la : ∀ t, Fin (N t) → ℕ) (h : ∀ t, Antitone (la t)) : Nonempty (TIdx la)

theorem **exists_tval_ne_zero** (la : Fin n → ℕ) (h : Antitone la) : ∃ x : TIdx (fun _ : Unit => la) → ℚ, tval (fun _ : Unit => la) x (fun _ => 1) = 1

theorem **fibA_mul** (la : ∀ t, Fin (N t) → ℕ) {K : Type} [Field K] [CharZero K] (a b : K) (g h : ∀ t, Matrix (Fin (N t)) (Fin (N t)) K) : fibA la a g * fibA la b h = fibA la (a * b) fun t => h t * g t

theorem **fibY_eq** (la : ∀ t, Fin (N t) → ℕ) (s : ℝ) (X : ∀ t, Matrix (Fin (N t)) (Fin (N t)) ℂ) : fibY la s X = fibY la s (fun _ => 1) * fibA la 1 X

theorem **fibA_two** : fibA (fun _ : Unit => ![1, 0]) (1 : ℚ) (fun _ => (2 : ℚ) • 1) = (2 : ℚ) • 1

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUnifOfIsMumfordOf

theorem **Data.IsMonomialTau.isPos** {Q : Data R} {m : ℕ} {t : Fin m → R} {β : Fin m → LinearMap.BilinForm ℤ Q.X} (h : Q.IsMonomialTau t β) {I : Ideal R} (ht : ∀ i, t i ∈ I) (h₀ : ∀ i y, 0 ≤ β i y y) (h₁ : ∀ y, y ≠ 0 → ∃ i, 0 < β i y y) : Q.IsPos I

theorem **Data.val_tau_zero** (Q : Data R) (b : Q.X) (z J) (hJ : Good Q.M.L z J) : val (Q.tau 0 b) (𝟙 _) z J = 1

theorem **IsMumfordOf.isCN** [IsLocalRing R] {ξ : Sp R ⟶ S} {Q : Data R} (h : IsMumfordOf F j e ξ Q) : IsCN R

theorem **IsMumfordOf.isPos** [IsLocalRing R] {ξ : Sp R ⟶ S} {Q : Data R} (h : IsMumfordOf F j e ξ Q) : Q.IsPos (maximalIdeal R)

theorem **IsMumfordOf.unif** [IsLocalRing R] {ξ : Sp R ⟶ S} {Q : Data R} (h : IsMumfordOf F j e ξ Q) : ∀ (R' : Type) [CommRing R'] [IsDomain R'] [IsLocalRing R'] (ρ : R →+* R'), IsCN R' → Function.Injective ρ → ρ.Finite → ∀ (D : DegData R' (maximalIdeal R')) (ε : D.toPre.X ≃+ Q.X), Refines Q ρ D.toPre ε → ∃ (G' : Over (Sp R')) (_ : GrpObj G') (_ : IsCommMonObj G') (π : G'.left ⟶ G.left) (A : PolAV (Sp (FractionRing R'))) (U : Unif (maximalIdeal R') D A G'), IsUnifOf F j e (spMap ρ ≫ ξ) π U

theorem **exists_tau** (D : DegPre R) (a b : D.X) : ∃ t : LB.Gen (LB.LinComb D.P.L (lamPts (toUnit (TK R) ≫ D.c a) (toUnit (TK R) ≫ D.c b)) negε), t.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (TK R).left) (z : Z) (J), Good D.P.L z J → val t r z J = val (D.tau a b) r z J

theorem **gw_idPull** {X Y : Scheme.{0}} (L : LB X) (f : Y ⟶ X) (y : Y) (j k : L.ι) (hj : f y ∈ L.U j) (hk : f y ∈ L.U k) : gw (idPull L) f y j k = gm f y _ (L.g j k)

theorem **idPull_isGen** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) : (idPull L).IsGen

theorem **grpPullback_id** {S : Scheme.{0}} (G : Over S) [GrpObj G] : IsGrpPullback G G (𝟙 S) (𝟙 G.left)

theorem **mono_jK** : Mono (jK R)

theorem **prod_neg_lamPts** {S : Scheme.{0}} {B : Over S} [GrpObj B] {T : Over S} (x y : T ⟶ B) {U : Type} [CommGroup U] (h : (T.left ⟶ B.left) → U) : ∏ k, (h (𝟙 _ ≫ lamPts x y k))⁻¹ ^ negε k = h (x ∗ y).left ∗ (h x.left)⁻¹ ∗ (h y.left)⁻¹ ∗ h (1 : T ⟶ B).left

theorem **refines_ofDeg** (D : DegPre R) (hB : D.TauBimult) : Refines (ofDeg D hB) (RingHom.id R) D (AddEquiv.refl _)

theorem **tate_isMonomialTau** (k : ℤ) : (tate k).IsMonomialTau ![PowerSeries.X] ![k • mulForm]

theorem **tate_isPos** {k : ℤ} (hk : 0 < k) : (tate k).IsPos (maximalIdeal (PowerSeries ℚ))

theorem **not_isPos_tate_zero** : ¬ (tate 0).IsPos (maximalIdeal (PowerSeries ℚ))

theorem **refines_tate** : Refines (tate 2) (RingHom.id _) tatePre (AddEquiv.refl _)

theorem **isCN_powerSeries** : IsCN (PowerSeries ℚ)

theorem **exists_pos_refined** : ∃ (Q : Data (PowerSeries ℚ)) (D : DegData (PowerSeries ℚ) (maximalIdeal _)) (ε : D.toPre.X ≃+ Q.X), Nontrivial Q.X ∧ IsCN (PowerSeries ℚ) ∧ Q.IsPos (maximalIdeal _) ∧ Refines Q (RingHom.id _) D.toPre ε

theorem **exists_unif_of_isMumfordOf** {o : Type} [CommRing o] {ζ : o} {g n : ℕ} (F : SFam g n ζ) {S : Scheme.{0}} (j : F.X ⟶ S) {G : Over S} [GrpObj G] (e : F.univ.A.left ⟶ G.left) [IsLocalRing R] (ξ : Sp R ⟶ S) (D : DegData R (maximalIdeal R)) (h : IsMumfordOf F j e ξ (ofDeg D.toPre D.tau_bimult)) : ∃ (G' : Over (Sp R)) (_ : GrpObj G') (_ : IsCommMonObj G') (π : G'.left ⟶ G.left) (A : PolAV (Sp (FractionRing R))) (U : Unif (maximalIdeal R) D A G'), IsUnifOf F j e (spMap (RingHom.id R) ≫ ξ) π U

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUniqueAdjoint

theorem **existsUnique_adjoint** (X₁ : Setup F W i) (a : Matrix (Fin 8) (Fin 8) ℤ) : ∃! a' : Matrix (Fin 8) (Fin 8) ℤ, (W + W.transpose) ∗ a' = a.transpose ∗ (W + W.transpose)

theorem **polar_adjoint** {s : ℕ} (S a a' : Matrix (Fin s) (Fin s) ℤ) (h : S ∗ a' = a.transpose ∗ S) (y χ : Fin s → ℤ) : (a.mulVec y) ⬝ᵥ (S.mulVec χ) = y ⬝ᵥ (S.mulVec (a'.mulVec χ))

theorem **exists_endU** (m : UMem X₁ z₁ act V ev X t δ) (a : Matrix (Fin 8) (Fin 8) ℤ) : ∃ E : EndU m.U.toUnifPre, E.aA.left ≫ m.φ = m.φ ≫ (act a).left

theorem **terms_sum** (m : UMem X₁ z₁ act V ev X t δ) {x₀ : X₁.S.Gam} {C : Type} {lab : C → X₁.S.Gam} {T : (Fin 8 → X) → C → V} (h : m.Terms x₀ lab T) (c : C) : SumF (maximalIdeal V) (FractionRing V) (fun χ => algebraMap V (FractionRing V) (T (m.e₈.symm χ) c) ∗ m.yK (X₁.S.th x₀, X₁.S.th_mem x₀)) (m.yK (X₁.S.th (lab c), X₁.S.th_mem _))

theorem **exists_frames** (m : UMem X₁ z₁ act V ev X t δ) (a : X₁.S.Gam) : ∃ (x : TFK (R

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUniqueIsLv

theorem **existsUnique_isLv** (f : Fr σ J C D Z J₀) : ∃ lv : GL (Fin j) (AF F ⊗[F] E) →* GAd σ J, IsLv σ J C D Z lv ∧ Function.Injective lv ∧ ∀ lv', IsLv σ J C D Z lv' → lv' = lv

theorem **all_inhabited** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E) (J : Matrix (Fin 3) (Fin 3) E) (C D Z : Matrix (Fin 3) (Fin 1) E) (J₀ : Matrix (Fin 1) (Fin 1) E) (l : GL (Fin 1) (AF F ⊗[F] E) →* GAd σ J) (u : GAd σ J) (c : Matrix (Fin 1) (Fin 1) (AF F ⊗[F] E)), Fr σ J C D Z J₀ ∧ IsLv σ J C D Z l ∧ Function.Injective l ∧ Crd σ J C D Z J₀ u 1 c ∧ u ∈ fixG σ J C

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUniqueIsTanFrameOfIsEmpty

theorem **der_add** (φ : K₀ →ₐ[c.F] K) (v : FrPt Xo c (DualNumber K)) (s s' : RulK Xo c K₀) : der φ v (s + s') = der φ v s + der φ v s'

theorem **der_smul** (φ : K₀ →ₐ[c.F] K) (v : FrPt Xo c (DualNumber K)) (a : K₀) (s : RulK Xo c K₀) : der φ v (a • s) = φ a ∗ der φ v s

theorem **der_sum** (φ : K₀ →ₐ[c.F] K) (v : FrPt Xo c (DualNumber K)) {ι : Type} (I : Finset ι) (a : ι → K₀) (s : ι → RulK Xo c K₀) : der φ v (∑ i ∈ I, a i • s i) = ∑ i ∈ I, φ (a i) ∗ der φ v (s i)

theorem **IsDef.val_eq** {y : FrPt Xo c K} {v : FrPt Xo c (DualNumber K)} (h : IsDef y v) (φ : K₀ →ₐ[c.F] K) {e : Ix Dw → ℤ} {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) : val ((cst c K).comp φ) v s = inl (val φ y s) + inr (der φ v s)

theorem **IsDef.der_mul** {y : FrPt Xo c K} {v : FrPt Xo c (DualNumber K)} (h : IsDef y v) (φ : K₀ →ₐ[c.F] K) {e e' : Ix Dw → ℤ} {f g : Rul Xo c} (hf : f ∈ secZ Xo c e) (hg : g ∈ secZ Xo c e') : der φ v (1 ⊗ₜ (f ∗ g)) = evR Xo c f y ∗ der φ v (1 ⊗ₜ g) + evR Xo c g y ∗ der φ v (1 ⊗ₜ f)

theorem **existsUnique_isShift** (N : FIx Dw → K) (v : FrPt Xo c (DualNumber K)) : ∃! v', IsShift N v v'

theorem **isShift_zero** (v : FrPt Xo c (DualNumber K)) : IsShift (0 : FIx Dw → K) v v

theorem **IsShift.unique** {N N' : FIx Dw → K} {v v' : FrPt Xo c (DualNumber K)} (h : IsShift N v v') (h' : IsShift N' v v') : N = N'

theorem **IsDef.shift** {y : FrPt Xo c K} {N : FIx Dw → K} {v v' : FrPt Xo c (DualNumber K)} (h : IsDef y v) (h' : IsShift N v v') : IsDef y v'

theorem **IsDef.exists_isShift** {y : FrPt Xo c K} {v v' : FrPt Xo c (DualNumber K)} (h : IsDef y v) (h' : IsDef y v') (hx : v'.x = v.x) : ∃ N : FIx Dw → K, IsShift N v v'

theorem **isDef_const** {y : FrPt Xo c K} {y₀ : FrPt Xo c (DualNumber K)} (h : FrMap (cst c K) y y₀) : IsDef y y₀

theorem **IsVert.isDef** {y : FrPt Xo c K} {N : FIx Dw → K} {v : FrPt Xo c (DualNumber K)} (h : IsVert y N v) : IsDef y v

theorem **existsUnique_isVert** (y : FrPt Xo c K) (N : FIx Dw → K) : ∃! v, IsVert y N v

theorem **IsVert.unique** {y : FrPt Xo c K} {N N' : FIx Dw → K} {v : FrPt Xo c (DualNumber K)} (h : IsVert y N v) (h' : IsVert y N' v) : N = N'

theorem **exists_isVert_iff** {y : FrPt Xo c K} {v : FrPt Xo c (DualNumber K)} : (∃ N : FIx Dw → K, IsVert y N v) ↔ IsDef y v ∧ v.x = spMap (cst c K).toRingHom ≫ y.x

theorem **der_dir** {ι : Type} [Fintype ι] (φ : K⁰ →ₐ[c.F] K) {W : FrPt Xo c (TrivSqZeroExt K (ι → K))} {m : ι → K} {v : FrPt Xo c (DualNumber K)} (h : FrMap (dir c m) W v) {e : Ix Dw → ℤ} {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) : der φ v s = ∑ k, (val ((inlAlgHom c.F K (ι → K)).comp φ) W s).snd k * m k

theorem **IsDef.der_shift** {y : FrPt Xo c K} {N : FIx Dw → K} {v v' : FrPt Xo c (DualNumber K)} (h : IsDef y v) (h' : IsShift N v v') {z : FIx Dw → FrPt Xo c (DualNumber K)} (hz : ∀ k, IsVert y (Pi.single k (1 : K)) (z k)) (φ : K⁰ →ₐ[c.F] K) {e : Ix Dw → ℤ} {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) : der φ v' s = der φ v s + ∑ k, N k * der φ (z k) s

theorem **IsVert.der_eq_weight** {y : FrPt Xo c K} {N : FIx Dw → K} {v : FrPt Xo c (DualNumber K)} (h : IsVert y N v) (hN : ∀ σ i j, i < j → N (some (σ, i, j)) = 0) (φ : K⁰ →ₐ[c.F] K) {e : Ix Dw → ℤ} {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) : der φ v s = (∑ k, (e k : K) * diagOf N k) * val φ y s

theorem **IsVert.der_of_triangular** {y : FrPt Xo c K} {N : FIx Dw → K} {v : FrPt Xo c (DualNumber K)} (h : IsVert y N v) (hN : ∀ σ i j, i < j → N (some (σ, i, j)) = 0) (φ : K⁰ →ₐ[c.F] K) (e : Ix Dw → ℤ) : ∃ a : K, ∀ s ∈ secK Xo c K₀ e, der φ v s = a * val φ y s

theorem **isHalf_psiSet** : IsHalf (psiSet H)

theorem **VIx.toF_injective** : Function.Injective (VIx.toF (Dw

theorem **IsHalf.sum_univ** {Φ : Finset (E →+* ℂ)} (h : IsHalf Φ) (g : (E →+* ℂ) → ℕ) : ∑ σ, g σ = ∑ τ ∈ Φ, (g τ + g (conjEmb τ))

theorem **IsHalf.card** {Φ : Finset (E →+* ℂ)} (h : IsHalf Φ) : Φ.card = Module.finrank ℚ E / 2

theorem **mult_conj** (τ : E →+* ℂ) : (Dw.mult τ = Dw.p ∧ Dw.mult (conjEmb τ) = Dw.q) ∨ (Dw.mult τ = Dw.q ∧ Dw.mult (conjEmb τ) = Dw.p)

theorem **card_HIx** {Φ : Finset (E →+* ℂ)} (h : IsHalf Φ) : Fintype.card (HIx Dw Φ) = Dw.dim

theorem **card_VIx** : Fintype.card (VIx Dw) = Module.finrank ℚ E / 2 * (Dw.p.choose 2 + Dw.q.choose 2)

theorem **card_TIx** {Φ : Finset (E →+* ℂ)} (h : IsHalf Φ) : Fintype.card (TIx Dw Φ) = dimFlag Dw

theorem **exOf_kap** : exOf Dw (kap Dw) 0 = detTanEx Dw

theorem **IsDef.isTanX** {y : FrPt Xo c K} {v : FrPt Xo c (DualNumber K)} (h : IsDef y v) : IsTanX y v.x

theorem **KSIso.existsUnique** {Φ : Finset (E →+* ℂ)} {y : FrPt Xo c K} (h : KSIso Φ y) {x' : Sp (DualNumber K) ⟶ Xo.X} (hx : IsTanX y x') : ∃! M, IsKS Φ y x' M

theorem **KSIso.injective** {Φ : Finset (E →+* ℂ)} {y : FrPt Xo c K} (h : KSIso Φ y) {x' x'' : Sp (DualNumber K) ⟶ Xo.X} (hx : IsTanX y x') (hx' : IsTanX y x'') {M : HIx Dw Φ → K} (k : IsKS Φ y x' M) (k' : IsKS Φ y x'' M) : x' = x''

theorem **KSIso.exists_universal** {Φ : Finset (E →+* ℂ)} {y : FrPt Xo c K} (h : KSIso Φ y) : ∃ (W : FrPt Xo c (TrivSqZeroExt K (HIx Dw Φ → K))) (z : (HIx Dw Φ → K) → FrPt Xo c (DualNumber K)), ∀ M, FrMap (dir c M) W (z M) ∧ IsDef y (z M) ∧ IsKS Φ y (z M).x M

theorem **IsTanFrame.isDef** {Φ : Finset (E →+* ℂ)} {y : FrPt Xo c K} {v : TIx Dw Φ → FrPt Xo c (DualNumber K)} (h : IsTanFrame Φ y v) : ∀ k, IsDef y (v k) | .inl k => (h.1 k).1 | .inr k => (h.2 k).isDef theorem KSIso.exists_isTanFrame {Φ : Finset (E →+* ℂ)} {y : FrPt Xo c K} (h : KSIso Φ y) : ∃ v, IsTanFrame Φ y v

theorem **existsUnique_isTanFrame_of_isEmpty** {Φ : Finset (E →+* ℂ)} [IsEmpty (HIx Dw Φ)] (y : FrPt Xo c K) : ∃! v, IsTanFrame Φ y v

theorem **isEmpty_HIx** (Φ : Finset (E →+* ℂ)) (h : Dw.p = 0 ∨ Dw.q = 0) : IsEmpty (HIx Dw Φ)

theorem **sum_FIx** (g : FIx Dw → K) : ∑ k, g k = ∑ k : VIx Dw, g k.toF + ∑ k ∈ Finset.univ.filter (fun k => k ∉ Set.range VIx.toF), g k

theorem **IsDef.der_shift_frame** {y : FrPt Xo c K} {v : VIx Dw → FrPt Xo c (DualNumber K)} (hv : ∀ k : VIx Dw, IsVert y (Pi.single k.toF (1 : K)) (v k)) {N : FIx Dw → K} {w w' : FrPt Xo c (DualNumber K)} (h : IsDef y w) (h' : IsShift N w w') (φ : K⁰ →ₐ[c.F] K) (e : Ix Dw → ℤ) : ∃ a : K, ∀ s ∈ secK Xo c K₀ e, der φ w' s = der φ w s + ∑ k : VIx Dw, N k.toF * der φ (v k) s + a * val φ y s

theorem **IsVert.der_eq** {y : FrPt Xo c K} {v : VIx Dw → FrPt Xo c (DualNumber K)} (hv : ∀ k : VIx Dw, IsVert y (Pi.single k.toF (1 : K)) (v k)) {N : FIx Dw → K} {w : FrPt Xo c (DualNumber K)} (h : IsVert y N w) (φ : K⁰ →ₐ[c.F] K) (e : Ix Dw → ℤ) : ∃ a : K, ∀ s ∈ secK Xo c K₀ e, der φ w s = ∑ k : VIx Dw, N k.toF * der φ (v k) s + a * val φ y s

theorem **KSIso.der_eq** {Φ : Finset (E →+* ℂ)} {y : FrPt Xo c K} (h : KSIso Φ y) {v : TIx Dw Φ → FrPt Xo c (DualNumber K)} (hv : IsTanFrame Φ y v) {w : FrPt Xo c (DualNumber K)} (hw : IsDef y w) {M : HIx Dw Φ → K} (hM : IsKS Φ y w.x M) (φ : K⁰ →ₐ[c.F] K) (e : Ix Dw → ℤ) : ∃ (C : VIx Dw → K) (a : K), ∀ s ∈ secK Xo c K₀ e, der φ w s = ∑ k, M k * der φ (v (.inl k)) s + ∑ k, C k * der φ (v (.inr k)) s + a * val φ y s

theorem **wedge_comb** (φ : K⁰ →ₐ[c.F] K) (v : ι → FrPt Xo c (DualNumber K)) (a : Matrix ι ι K⁰) (s : ι → RulK Xo c K₀) : wedge φ v (fun i => ∑ j, a i j • s j) = φ a.det * wedge φ v s

theorem **wedge_comp** {κ : Type} [Fintype κ] (φ : K⁰ →ₐ[c.F] K) (v : κ → FrPt Xo c (DualNumber K)) (e : ι ≃ κ) (s : ι → RulK Xo c K₀) : wedge φ (v ∘ e) s = wedge φ v (s ∘ e.symm)

theorem **wedge_perm** (φ : K⁰ →ₐ[c.F] K) (v : ι → FrPt Xo c (DualNumber K)) (σ : Equiv.Perm ι) (s : ι → RulK Xo c K₀) : wedge φ (v ∘ σ) s = (Equiv.Perm.sign σ : ℤ) * wedge φ v s

theorem **jac_perm** (φ : K⁰ →ₐ[c.F] K) (y : FrPt Xo c K) (v : ι → FrPt Xo c (DualNumber K)) (σ : Equiv.Perm ι) (s₀ : RulK Xo c K₀) (s : ι → RulK Xo c K₀) : jac φ y (v ∘ σ) s₀ s = (Equiv.Perm.sign σ : ℤ) * jac φ y v s₀ s

theorem **wedge_jac_of_comb** {φ : K⁰ →ₐ[c.F] K} {y : FrPt Xo c K} {v w : ι → FrPt Xo c (DualNumber K)} {e : Ix Dw → ℤ} {P : Matrix ι ι K} {a : ι → K} (h : ∀ k, ∀ s ∈ secK Xo c K₀ e, der φ (w k) s = ∑ j, P k j * der φ (v j) s + a k * val φ y s) {s : ι → RulK Xo c K₀} (hs : ∀ i, s i ∈ secK Xo c K₀ e) : ((∀ i, val φ y (s i) = 0) → wedge φ w s = P.det * wedge φ v s) ∧ ∀ s₀ ∈ secK Xo c K₀ e, jac φ y w s₀ s = P.det * jac φ y v s₀ s

theorem **KSIso.chain_rule** {y : FrPt Xo c K} (h : KSIso Φ y) {v w : TIx Dw Φ → FrPt Xo c (DualNumber K)} (hv : IsTanFrame Φ y v) {M : HIx Dw Φ → HIx Dw Φ → K} {N : VIx Dw → FIx Dw → K} (hM : ∀ k, IsDef y (w (.inl k)) ∧ IsKS Φ y (w (.inl k)).x (M k)) (hN : ∀ k, IsVert y (N k) (w (.inr k))) (φ : K⁰ →ₐ[c.F] K) {e : Ix Dw → ℤ} {s : TIx Dw Φ → RulK Xo c K₀} (hs : ∀ i, s i ∈ secK Xo c K₀ e) : ((∀ i, val φ y (s

i) = 0) → wedge φ w s = (Matrix.of M).det * (Matrix.of fun k j : VIx Dw => N k j.toF).det * wedge φ v s) ∧ ∀ s₀ ∈ secK Xo c K₀ e, jac φ y w s₀ s = (Matrix.of M).det * (Matrix.of fun k j : VIx Dw => N k j.toF).det * jac φ y v s₀ s

theorem **KSIso.wedge_jac_eq** {y : FrPt Xo c K} (h : KSIso Φ y) {v w : TIx Dw Φ → FrPt Xo c (DualNumber K)} (hv : IsTanFrame Φ y v) (hw : IsTanFrame Φ y w) (φ : K₀ →ₐ[c.F] K) {e : Ix Dw → ℤ} {s : TIx Dw Φ → RulK Xo c K₀} (hs : ∀ i, s i ∈ secK Xo c K₀ e) : ((∀ i, val φ y (s i) = 0) → wedge φ w s = wedge φ v s) ∧ ∀ s₀ ∈ secK Xo c K₀ e, jac φ y w s₀ s = jac φ y v s₀ s

theorem **KSIso.existsUnique_isWedgeVal** {y : FrPt Xo c K} (h : KSIso Φ y) (o : ι ≃ TIx Dw Φ) (φ : K₀ →ₐ[c.F] K) {e : Ix Dw → ℤ} {s : ι → RulK Xo c K₀} (hs : ∀ i, s i ∈ secK Xo c K₀ e) (h0 : ∀ i, val φ y (s i) = 0) : ∃! d, IsWedgeVal Φ o φ y s d

theorem **existsUnique_isWedgeVal_of_isEmpty** [IsEmpty (HIx Dw Φ)] (o : ι ≃ TIx Dw Φ) (φ : K₀ →ₐ[c.F] K) {y : FrPt Xo c K} {s : ι → RulK Xo c K₀} (h0 : ∀ i, val φ y (s i) = 0) : ∃! d, IsWedgeVal Φ o φ y s d

theorem **IsWedgeVal.reorder** {o : ι ≃ TIx Dw Φ} {φ : K₀ →ₐ[c.F] K} {y : FrPt Xo c K} {s : ι → RulK Xo c K₀} {d : K} (h : IsWedgeVal Φ o φ y s d) (o' : ι ≃ TIx Dw Φ) : IsWedgeVal Φ o' φ y s ((Equiv.Perm.sign (o'.trans o.symm) : ℤ) * d)

theorem **IsJacVal.reorder** {o : ι ≃ TIx Dw Φ} {φ : K₀ →ₐ[c.F] K} {y : FrPt Xo c K} {s₀ : RulK Xo c K₀} {s : ι → RulK Xo c K₀} {d : K} (h : IsJacVal Φ o φ y s₀ s d) (o' : ι ≃ TIx Dw Φ) : IsJacVal Φ o' φ y s₀ s ((Equiv.Perm.sign (o'.trans o.symm) : ℤ) * d)

theorem **IsWedgeVal.comb** {o : ι ≃ TIx Dw Φ} {φ : K₀ →ₐ[c.F] K} {y : FrPt Xo c K} {s : ι → RulK Xo c K₀} {d : K} (h : IsWedgeVal Φ o φ y s d) (a : Matrix ι ι K₀) : IsWedgeVal Φ o φ y (fun i => ∑ j, a i j • s j) (φ a.det * d)

theorem **ramIdx_spec** {p : ℕ} (hp : p ≠ 0) (h : (p : T.V) ∈ IsLocalRing.maximalIdeal T.V) : 0 < ramIdx T p ∧ nval T.V (ramIdx T p) (p : T.Fr) = 1

theorem **normOfOrd_le_iff** {p : ℕ} (hp : 1 < p) (w : EReal) (b : ℝ) : normOfOrd p w ≤ ENNReal.ofReal ((p : ℝ) ^ b) ↔ ((-b : ℝ) : EReal) ≤ w

theorem **wedgeNorm_le_iff** {p : ℕ} (hp : 1 < p) (φ : CoefHom c io K₀) (y : IntFr Xo c T io) (s : ι → RulK Xo c K₀) (b : ℝ) : wedgeNorm Φ p φ y s ≤ ENNReal.ofReal ((p : ℝ) ^ b) ↔ ((-b : ℝ) : EReal) ≤ wedgeOrd_vp Φ p φ y s

theorem **wedgeOrd_ne_bot_iff** {n : ℕ} {φ : CoefHom c io K₀} {y : IntFr Xo c T io} {s : ι → RulK Xo c K₀} : wedgeOrd Φ n φ y s ≠ ⊥ ↔ ∃! w, IsWedgeOrd Φ n φ y s w

theorem **wedgeOrd_eq** {n : ℕ} {φ : CoefHom c io K₀} {y : IntFr Xo c T io} {s : ι → RulK Xo c K₀} (h : wedgeOrd Φ n φ y s ≠ ⊥) {w : EReal} (hw : IsWedgeOrd Φ n φ y s w) : wedgeOrd Φ n φ y s = w

theorem **nval_sign_mul** (n : ℕ) (u : ℤˣ) (x : T.Fr) : nval T.V n (((u : ℤ) : T.Fr) * x) = nval T.V n x

theorem **wedgeOrd_eq_of_unique** (n : ℕ) {φ : CoefHom c io K₀} {y : IntFr Xo c T io} {s : ι → RulK Xo c K₀} {o : ι ≃ TIx Dw Φ} (h : letI

theorem **KSIso.wedgeOrd_eq** {y : IntFr Xo c T io} (h : letI

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUniqueIsWeilChar

theorem **mul_mem_lie** (x y : LieG G) : ptMap (TrivSqZeroExt.fstHom O O O) ≫ (x.1 * y.1) = toUnit (specO O O) ≫ η[G]

theorem **omega_lieAdd** (f : omega G) (x y : LieG G) : f.1 (lieAdd x y) = f.1 x + f.1 y

theorem **omega_lieScale** (f : omega G) (c : O) (x : LieG G) : f.1 (AvDefs.lieScale G c x) = c * f.1 x

theorem **omega_lieZero** (f : omega G) : f.1 (lieZero G) = 0

theorem **omegaAct_apply** [StarRing Λ] (a : Λ) (f : omega G) (x : LieG G) : (omegaAct α a f).1 x = f.1 (AvDefs.lieAct G α (star a) x)

theorem **isEig_lieAdd** {τ : Λ →+* O} {x y : LieG G} (hx : AvDefs.IsEig G α τ x) (hy : AvDefs.IsEig G α τ y) : AvDefs.IsEig G α τ (lieAdd x y)

theorem **isEig_lieZero** (τ : Λ →+* O) : AvDefs.IsEig G α τ (lieZero G)

theorem **isEig_lieScale** {τ : Λ →+* O} (c : O) {x : LieG G} (hx : AvDefs.IsEig G α τ x) : AvDefs.IsEig G α τ (AvDefs.lieScale G c x)

theorem **isAl_dlogTv** (χ : Tv G p) : IsAl χ (dlogTv χ).1

theorem **eq_dlogTv** {χ : Tv G p} {f : omega G} (h : IsAl χ f.1) : f = dlogTv χ

theorem **dlogTv_mul** {χ χ' χ'' : Tv G p} (h : ∀ n, χ''.χ n = χ.χ n * χ'.χ n) : dlogTv χ'' = dlogTv χ + dlogTv χ'

theorem **map_span_dlogTv** : (Submodule.span O (Set.range (dlogTv (G

theorem **nilp** (m : ℕ) : IsNilpotent (p : O ⧸ Ideal.span {(p : O) ^ m})

theorem **isLocalRing** : IsLocalRing O

theorem **factor_red** (m : ℕ) : ptMap (Ideal.Quotient.factorₐ O (Ideal.span_singleton_le_span_singleton.mpr (pow_dvd_pow (p : O) (Nat.le_succ m)))) ≫ ptMap (red p O (m + 1)) = ptMap (red p O m)

theorem **hom_ext_red** {X : Over (Sp O)} {Q Q' : specO O O ⟶ X} (h : ∀ m, ptMap (red p O m) ≫ Q = ptMap (red p O m) ≫ Q') : Q = Q'

theorem **exists_toG** (x : specO O O ⟶ P.Gt) : ∃ Q : specO O O ⟶ P.G, ∀ m, ptMap (red p O m) ≫ Q = P.cmp _ (nilp m) (ptMap (red p O m) ≫ x)

theorem **red_toG** (x : specO O O ⟶ P.Gt) (m : ℕ) : ptMap (red p O m) ≫ P.toG x = P.cmp _ (nilp m) (ptMap (red p O m) ≫ x)

theorem **toG_bijective** : Function.Bijective P.toG

theorem **toG_act** (a : Λ) (x : specO O O ⟶ P.Gt) : P.toG (x ≫ P.oGt.act a) = P.toG x ≫ P.oG.act a

theorem **exists_generic** : ∃ j : (specO O O ⟶ P.G) →* Pts P.A.A, Function.Injective j ∧ ∀ Q z, z.left ≫ P.gen = spMap (algebraMap O C) ≫ Q.left ↔ z = j Q

theorem **eq_toA_iff** (x : specO O O ⟶ P.Gt) (z : Pts P.A.A) : z = P.toA x ↔ z.left ≫ P.gen = spMap (algebraMap O C) ≫ (P.toG x).left

theorem **toA_injective** : Function.Injective P.toA

theorem **toA_act** (a : Λ) (x : specO O O ⟶ P.Gt) : P.toA (x ≫ P.oGt.act a) = P.toA x ≫ P.oA.act a

theorem **finite_flat** (N : ℕ) (hN : 1 ≤ N) : IsFinite (torsion P.Gt N).hom ∧ Flat (torsion P.Gt N).hom

theorem **tateTorus_le_tateGt** : P.tateTorus ≤ P.tateGt

theorem **char_ext** {n : ℕ} {f f' : torsion P.Gt (p ^ n) ⟶ Gm (Sp O)} (h : ∀ w : specO O O ⟶ torsion P.Gt (p ^ n), algebraMap O C (unitOf (w ≫ f) : O) = algebraMap O C (unitOf (w ≫ f') : O)) : f = f'

theorem **toA_pow** {N : ℕ} (w : specO O O ⟶ torsion P.Gt N) : P.toA (w ≫ torsion.ι P.Gt N) ^ N = 1

theorem **existsUnique_isWeilChar** (t : P.T) : ∃! χ : Tv P.Gt p, P.IsWeilChar t χ

theorem **isWeilChar_weilChar** (t : P.T) : P.IsWeilChar t (P.weilChar t)

theorem **weilChar_add** (s t : P.T) (n : ℕ) : (P.weilChar (s + t)).χ n = (P.weilChar s).χ n * (P.weilChar t).χ n

theorem **dlogTv_weilChar_act** (a : Λ) (t : P.T) : dlogTv (P.weilChar (P.actT a t)) = omegaAct P.oGt a (dlogTv (P.weilChar t))

theorem **eq_al₀_iff** (t : P.T) (f : omega P.Gt) : f = P.al₀ t ↔ ∃ χ : Tv P.Gt p, P.IsWeilChar t χ ∧ IsAl χ f.1

theorem **al₀_act** (a : Λ) (t : P.T) : P.al₀ (P.actT a t) = omegaAct P.oGt a (P.al₀ t)

theorem **al_tmul** (c : O) (t : P.T) : P.al (c ⊗ₜ t) = c • P.al₀ t

theorem **map_range_al_le** : (LinearMap.range P.al).map (omega P.Gt).subtype ≤ htSpan P.Gt p

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUniqueLiesUnder

theorem **isComb_twist** (M : Adl C Y) (h : ∀ σ z, PosAt (M.m σ) z) (c : InfinitePlace F → ℝ) : IsComb (M.twist c 0) (fun _ : Unit => M) (fun _ => 1) fun σ _ => c (InfinitePlace.mk σ)

theorem **LiesUnder.compPt_eq** {k : s.Comp} {y : Pt s} {x : (Φ.C k).Y.X} (h : LiesUnder Φ k y x) : compPt s y = k

theorem **existsUnique_liesUnder** (y : Pt s) : ∃! x, LiesUnder Φ (compPt s y) y x

theorem **liesUnder_basePt** (y : Pt s) : LiesUnder Φ (compPt s y) y (basePt Φ y)

theorem **π_basePt** (y : Pt s) : (Φ.C _).π (basePt Φ y) = below y.y

theorem **expo_single** (a : Sym E) (n : ℤ) (x : Letter Dw) : expo Dw (Finsupp.single a n) x = if letterOf Dw a.2.2 = some x then n else 0

theorem **expo_rootSym** (a : ℕ → Bool) (R : Option ℝ) (τ : E →+* ℂ) (i : ℕ) : expo Dw (rootSym Dw a R τ i) = rootEx Dw τ i

theorem **shimXAt_eq** {A : Amb D} {F : SrcFam A} {L : CmList A} {jC : ∀ (p : ℕ) [Fact p.Prime], PlL D p → (DV.KW p →+* ℂ)} {ng : NgPkg} (k : Kits F L (cxSt D.what₂.LL) jC ng) {N j : ℕ} (hN : 2 ≤ N) (hj : j ≤ N) (κ : Lv A) (h : 1 ≤ j ∧ j < N) (loc : LocSt D N j ng) (glob : GlobSt fun p => (loc.loc p).Mdl) (adm : AdmSt D N j jC) (met : MetIn (k N j hN hj κ).Φ) (e : (k N j hN hj κ).pad = padAt _ loc glob adm (flatSt _ _ met.hm) (arIn _ _ met.hm)) : shimXAt (cutSt _) (resSt _) k N j hN hj κ h = (ofKit h (k N j hN hj κ) adm met).Y

theorem **setB_eq** (B : ℝ) : FlagAr.setB O.Y B = Pad.setB O.κ B

theorem **mult_pos** : 0 < W.mult

theorem **heightData_satisfies** (Ps : ℝ → O.s.Comp → MvPolynomial (Option (D.what₂.Tau × Fin O.N)) ℝ) : O.IsFlagHeightData (O.aux Ps)

theorem **IsFlagHeightData.eq_aux** {O : Setup D} {A : SlopeTestRemainingChoices O.Y} (h : O.IsFlagHeightData A) : A = O.aux A.Ps

theorem **htIs_hts** : O.HtIs O.hts

theorem **Reads.mem_int_iff** {O : Setup D} {A : SlopeTestRemainingChoices O.Y} {T : Real0 O.Y} {H : Ht O.Y} (h : O.Reads A T H) (y : Pt O.s) : y ∈ H.int ↔ ∀ a i, h.pts.conj (compPt O.s y) y a i ∉ (T.M _).Z a

theorem **ObjOf.mono** {D : D7} {Tr Tr' : Setup D → Prop} (h : ∀ O, Tr O → Tr' O) {Y : DV.ShimX D.what₂.Tau} {A : SlopeTestRemainingChoices Y} {T : Real0 Y} {H : Ht Y} (o : ObjOf Tr Y A T H) : ObjOf Tr' Y A T H

theorem **ObjOf.withPs** {D : D7} {Tr : Setup D → Prop} {Y : DV.ShimX D.what₂.Tau} {A : SlopeTestRemainingChoices Y} {T : Real0 Y} {H : Ht Y} (o : ObjOf Tr Y A T H) (Ps : ℝ → Y.Comp → MvPolynomial (Option (D.what₂.Tau × Fin Y.N)) ℝ) : ObjOf Tr Y { A with Ps

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUniqueSubMemInitSpan

theorem **tri_mul_diag** {B B' : Matrix (Fin m) (Fin m) R} (h : B.BlockTriangular id) (h' : B'.BlockTriangular id) (i : Fin m) : (B * B') i i = B i i * B' i i

theorem **mem_initSpan** {v : Fin m → N} {i : ℕ} {w : N} : w ∈ initSpan R v i ↔ ∃ a : Fin m → R, (∀ j : Fin m, i ≤ j → a j = 0) ∧ ∑ j, a j • v j = w

theorem **initSpan_mono** (v : Fin m → N) {i i' : ℕ} (h : i ≤ i') : initSpan R v i ≤ initSpan R v i'

theorem **initSpan_le_of_tri** {v v' : Fin m → N} {B : Matrix (Fin m) (Fin m) R} (hB : B.BlockTriangular id) (h : ∀ i, v' i = ∑ j, B j i • v j) (i : ℕ) : initSpan R v' i ≤ initSpan R v i

theorem **sub_diag_mem_initSpan** {v v' : Fin m → N} {B : Matrix (Fin m) (Fin m) R} (hB : B.BlockTriangular id) (h : ∀ i, v' i = ∑ j, B j i • v j) (i : Fin m) : v' i - B i i • v i ∈ initSpan R v i

theorem **existsUnique_sub_mem_initSpan** {v : Fin m → N} (hv : LinearIndependent R v) (i : Fin m) {w : N} (hw : w ∈ initSpan R v (i + 1)) : ∃! a : R, w - a • v i ∈ initSpan R v i

theorem **inM** {k : F.J} {z : Sp S ⟶ Z} (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) : ⊤ ≤ x ⁻¹ᵁ (F.M k).U

theorem **sum_vecAt** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (a : Fin (Dw.mult σ) → S) : ∑ i, a i • F.vecAt k z hz hx σ i = vec (F.M k).toMChart x (F.inM hz hx) ((F.uAt k z hz σ).mulVec a)

theorem **uAt_split** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (σ : E →+* ℂ) : ∃ w : Matrix (Fin (Dw.mult σ)) (Fin (F.M k).n) S, w * F.uAt k z hz σ = 1

theorem **uAt_spMap** (k : F.J) (φ : S →+* S') (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (hz' : ⊤ ≤ (spMap φ ≫ z) ⁻¹ᵁ F.U k) (σ : E →+* ℂ) : F.uAt k (spMap φ ≫ z) hz' σ = (F.uAt k z hz σ).map φ

theorem **linearIndependent_vecAt** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) : LinearIndependent S (F.vecAt k z hz hx σ)

theorem **tanPull_vecAt** (k : F.J) (φ : S →+* S') (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (hz' : ⊤ ≤ (spMap φ ≫ z) ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) {x' : Sp S' ⟶ Y.X} (hx' : (spMap φ ≫ z) ≫ π = x') (e : spMap φ ≫ x = x') (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : tanPull φ e (F.vecAt k z hz hx σ i) = F.vecAt k (spMap φ ≫ z) hz' hx' σ i

theorem **B_tri** (k k' : F.J) (σ : E →+* ℂ) : (F.B k k' σ).BlockTriangular id

theorem **B_map_tri** (k k' : F.J) (σ : E →+* ℂ) (f : Γ(Z, F.U k ⊓ F.U k') →+* S) : ((F.B k k' σ).map f).BlockTriangular id

theorem **chg_uAt** (k k' : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (hz' : ⊤ ≤ z ⁻¹ᵁ F.U k') (hh : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (hM : ⊤ ≤ x ⁻¹ᵁ ((F.M k).U ⊓ (F.M k').U)) (σ : E →+* ℂ) : (chg (F.M k).toMChart (F.M k').toMChart).map (ev x _ hM) * F.uAt k' z hz' σ = F.uAt k z hz σ * (F.B k k' σ).map (ev z _ hh)

theorem **vecAt_eq_sum** (k k' : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (hz' : ⊤ ≤ z ⁻¹ᵁ F.U k') (hh : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : F.vecAt k' z hz' hx σ i = ∑ j, (F.B k k' σ).map (ev z _ hh) j i • F.vecAt k z hz hx σ j

theorem **B_map_mul** (k k' k'' : F.J) (z : Sp S ⟶ Z) (h₁ : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) (h₂ : ⊤ ≤ z ⁻¹ᵁ (F.U k' ⊓ F.U k'')) (h₃ : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k'')) (σ : E →+* ℂ) : (F.B k k' σ).map (ev z _ h₁) * (F.B k' k'' σ).map (ev z _ h₂) = (F.B k k'' σ).map (ev z _ h₃)

theorem **B_map_self** (k : F.J) (z : Sp S ⟶ Z) (h : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k)) (σ : E →+* ℂ) : (F.B k k σ).map (ev z _ h) = 1

theorem **ev_B_diag_mul** (k k' k'' : F.J) (z : Sp S ⟶ Z) (h₁ : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) (h₂ : ⊤ ≤ z ⁻¹ᵁ (F.U k' ⊓ F.U k'')) (h₃ : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k'')) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : ev z _ h₁ (F.B k k' σ i i) * ev z _ h₂ (F.B k' k'' σ i i) = ev z _ h₃ (F.B k k'' σ i i)

theorem **isUnit_B_diag** (k k' : F.J) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : IsUnit (F.B k k' σ i i)

theorem **T_some** (k k' : F.J) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : (F.T k k' (some ⟨σ, i⟩)).val = F.B k k' σ i i

theorem **ev_T_none** (k k' : F.J) (z : Sp S ⟶ Z) (h : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (hM : ⊤ ≤ x ⁻¹ᵁ ((F.M k).U ⊓ (F.M k').U)) : ev z _ h (F.T k k' none).val = ev x _ hM (thRat (F.M k) (F.M k'))

theorem **TAt_mul** (k k' k'' : F.J) (z : Sp S ⟶ Z) (h₁ : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) (h₂ : ⊤ ≤ z ⁻¹ᵁ (F.U k' ⊓ F.U k'')) (h₃ : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k'')) : F.TAt k k' z h₁ * F.TAt k' k'' z h₂ = F.TAt k k'' z h₃

theorem **isLB_line** (ix : Ix Dw) : (F.line ix).IsLB

theorem **isLB_gr** (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : (F.gr σ i).IsLB

theorem **isLB_de** : F.de.IsLB

theorem **ev_LL_g** (e : Ix Dw → ℤ) (k k' : F.J) (z : Sp S ⟶ Z) (h : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) : ev z _ h ((F.LL e).g k k').val = (∏ ix, F.TAt k k' z h ix ^ (-e ix)).val

theorem **isLB_LL** (e : Ix Dw → ℤ) : (F.LL e).IsLB

theorem **isLB_detOm** : F.detOm.IsLB

theorem **LL_add_g** (e e' : Ix Dw → ℤ) (k k' : F.J) : (F.LL (e + e')).g k k' = (show Γ(Z, F.U k ⊓ F.U k')ˣ from (F.LL e).g k k') * (show Γ(Z, F.U k ⊓ F.U k')ˣ from (F.LL e').g k k')

theorem **LL_single_neg** (ix : Ix Dw) : F.LL (Pi.single ix (-1)) = F.line ix

theorem **LL_single_none** : F.LL (Pi.single none 1) = F.de

theorem **initSpan_vecAt_eq** (k k' : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (hz' : ⊤ ≤ z ⁻¹ᵁ F.U k') {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : ℕ) : initSpan S (F.vecAt k z hz hx σ) i = initSpan S (F.vecAt k' z hz' hx σ) i

theorem **step_eq** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : ℕ) : F.step z hx σ i = initSpan S (F.vecAt k z hz hx σ) i

theorem **step_mono** (z : Sp S ⟶ Z) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) {i i' : ℕ} (h : i ≤ i') : F.step z hx σ i ≤ F.step z hx σ i'

theorem **tanPull_mem_step** (φ : S →+* S') (z : Sp S ⟶ Z) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) {z' : Sp S' ⟶ Z} (ez : spMap φ ≫ z = z') {x' : Sp S' ⟶ Y.X} (hx' : z' ≫ π = x') (e : spMap φ ≫ x = x') (σ : E →+* ℂ) (i : ℕ) {v : TanM x Y.t.G} (hv : v ∈ F.step z hx σ i) : tanPull φ e v ∈ F.step z' hx' σ i

theorem **mem_step_iff_of_injective** {V K : Type} [CommRing V] [IsLocalRing V] [CommRing K] (φ : V →+* K) (hφ : Function.Injective φ) (z : Sp V ⟶ Z) {x : Sp V ⟶ Y.X} (hx : z ≫ π = x) {zK : Sp K ⟶ Z} (ez : spMap φ ≫ z = zK) {xK : Sp K ⟶ Y.X} (hxK : zK ≫ π = xK) (e : spMap φ ≫ x = xK) (σ : E →+* ℂ) (i : ℕ) (v : TanM x Y.t.G) : v ∈ F.step z hx σ i ↔ tanPull φ e v ∈ F.step zK hxK σ i

theorem **existsUnique_sub_mem_step** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : F.vecAt k z hz hx σ i ∈ F.step z hx σ (i + 1) ∧ ∀ w ∈ F.step z hx σ (i + 1), ∃! a : S, w - a • F.vecAt k z hz hx σ i ∈ F.step z hx σ i

theorem **vecAt_sub_mem_step** (k k' : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) (hz' : ⊤ ≤ z ⁻¹ᵁ F.U k') (hh : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : F.vecAt k' z hz' hx σ i - ev z _ hh ((F.gr σ i).g k k').val • F.vecAt k z hz hx σ i ∈ F.step z hx σ i

theorem **exists_retraction** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : ℕ) : ∃ r : TanM x Y.t.G →ₗ[S] F.step z hx σ i, ∀ v : F.step z hx σ i, r v = v

theorem **dualRestrict_surjective** (k : F.J) (z : Sp S ⟶ Z) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (σ : E →+* ℂ) (i : ℕ) : Function.Surjective (F.step z hx σ i).dualRestrict

theorem **de_iso_pull** (h : (FlagMod.de Y).IsLB) : LBIso F.de ((FlagMod.de Y).pull π)

theorem **detOm_iso_pull** (h : (FlagMod.detOm Y).IsLB) : LBIso F.detOm ((FlagMod.detOm Y).pull π)

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsUnramPtsDefinite

theorem **exZ_zero_model** : exZ Dw (0 : WTZ E) = 0

theorem **mul_smul_idem_mem_cuspOn** {Z : Comp Xo c} {G : WTZ E} {f : Rul Xo c} (hf : f ∈ cuspOn U Z G) (a : c.F) : f * (a • idem Z) ∈ cuspOn U Z G

theorem **TwSec.constModel_spec** {Z : Comp Xo c} (h : HasPts U Z) (a : c.F) (ha : a ≠ 0) : (TwSec.constModel x h a ha).degLo = 0 ∧ IntOff ∅ (TwSec.constModel x h a ha).lo

theorem **exists_twSec_const** (hpq : 1 ≤ Dw.p + Dw.q) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) (Z : Comp Xo c) (a : c.F) (ha : a ≠ 0) : ∃ s : TwSec U x Z 0, s.sig = a • idem Z ∧ (∀ σ, s.cInf σ = ‖σ a‖) ∧ (∀ w, s.lo w = vAdd w a) ∧ s.degLo = 0

theorem **TwSec.lo_le_hi_of_hasPts** {Z : Comp Xo c} {G₀ : WTZ E} (s : TwSec U x Z G₀) (h : HasPts U Z) (w : HeightOneSpectrum (𝒪 c.F)) : s.lo w ≤ s.hi w

theorem **TwSec.cInf_eq_of_hasPts** {Z : Comp Xo c} {G₀ : WTZ E} (s t : TwSec U x Z G₀) (h : HasPts U Z) (e : s.sig = t.sig) : s.cInf = t.cInf

theorem **TwSec.sig_ne_zero_of_hasPts** {Z : Comp Xo c} {G₀ : WTZ E} (s : TwSec U x Z G₀) (h : HasPts U Z) : s.sig ≠ 0

theorem **unramPts_of_index_one** {B : Finset ℕ} (h : ∀ (Z : Comp Xo c) (w : HeightOneSpectrum (𝒪 c.F)), (∀ p ∈ B, ((p : ℕ) : 𝒪 c.F) ∉ w.asIdeal) → ∃ (T : CDVR) (ι : c.F →+* T.Fr) (y : IntFr Xo c T ι), OverLa w T.V ι 1 ∧ algOn Z T ι y) : UnramPts Xo c B

theorem **mem_badPrimes_iff** {p : ℕ} (hp : p.Prime) : p ∈ badPrimes H l ↔ p = 2 ∨ p = H.d ∨ p ∣ (discr E).natAbs ∨ p ∣ l.M ∨ p ∣ l.A

theorem **not_intOff_half** {K : Type} [Field K] [NumberField K] (w : HeightOneSpectrum (𝒪 K)) : ¬ IntOff ∅ (fun _ : HeightOneSpectrum (𝒪 K) => (1 / 2 : ℝ))

theorem **mem_holSub_model** {Cp : Type} (X : Cp → Data E r) (la : ∀ k, Wt (X k).Ψ r) (P : FDSec X) : P ∈ holSub X la ↔ ∀ k, P k ∈ (X k).SS (la k)

theorem **isSec_inv** (X : Data E r) {la la' : Wt X.Ψ r} (hla : la' + la = 0) {e s t : Frame X.Ψ r → ℂ} (he : X.IsSec 0 e) (ht : X.IsSec la t) (hee : e ∗ e = e) (hes : e ∗ s = s) (het : e ∗ t = t) (hst : s ∗ t = e) : X.IsSec la' s

theorem **mul_mem_cuspOn_of_isSec** {Z : Comp Xo c} {G₀ : WTZ E} (h0 : Even G₀.2) {s : Rul Xo c} (hs : s ∈ secZ Xo c (exZ Dw G₀)) (hZ : s ∈ suppOn Z) (hh : ∀ σ, GoodEmb c σ → ∀ k, ((U σ).X k).IsSec (modelWt (U σ) G₀ k) (rdU (U σ) s k)) (hb : ∀ σ, GoodEmb c σ → ∃ C : ℝ, ∀ z ∈ framesOn (U σ) Z, homN (U σ) x G₀ s z ≤ C) {G : WTZ E} (hG : Even G.2) {f : Rul Xo c} (hf : f ∈ cuspOn U Z G) : f ∗ s ∈ cuspOn U Z (G + G₀)

theorem **exists_twSec_of_dom** (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) {Z : Comp Xo c} {la : Weight Dw} {δ : HeightOneSpectrum (𝒪 c.F) → ℝ} (s : ConstSec U x Z la δ) (hδ : (Function.support δ).Finite) {a : (E →+* ℂ) × ℕ → ℕ} (ha : Dom a 0) (hla : exZ Dw (wtOf Dw (-la) 0) = exZ Dw (wtZ a 0)) : ∃ t : TwSec U x Z (wtOf Dw la 0), t.sig = s.sig ∧ t.cInf = s.cInf ∧ t.lo = δ + s.rho_vw ∧ t.hi = t.lo

theorem **exists_twSec_flat** (hq : Dw.q = 0) (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) {Z : Comp Xo c} {τ : Psi H} (s : FlatSec U x Z τ) : ∃ t : TwSec U x Z (wtOf Dw (flatWt Dw τ) 0), t.sig = s.sig ∧ t.cInf = s.cInf ∧ t.lo = s.rho_vw ∧ t.hi = t.lo ∧ t.degLo = s.deg

theorem **exists_unramPts_definite** (hq : Dw.q = 0) (hp : 1 ≤ Dw.p) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (hF : FnAlgClosed Xo c) : ∃ B : Finset ℕ, (∀ p ∈ B, p.Prime) ∧ UnramPts Xo c B

theorem **exists_twSec_ne_zero** (x : ℝ) : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q)) (Z : Comp Xo c) (G₀ : WTZ E) (B : Finset ℕ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) ∧ (∃ i, exZ Dw G₀ i = -1) ∧ (∀ p ∈ B, p.Prime) ∧ UnramPts Xo c B ∧ ∃ t : TwSec U x Z G₀, IntOff B t.lo ∧ t.hi = t.lo

theorem **TwSec.degLo_symm** {Z : Comp Xo c} {G₀ : WTZ E} (s : TwSec U x Z G₀) (h : s.hi = s.lo) : s.symm.degLo = -s.degLo

Development.AbelianVarietyTheory.Definitions.Divisors.ExistsValuedHeckeCharRamified

theorem **isCMType_places** : IsCMType (fun w : InfinitePlace E => w.embedding)

theorem **IsCMType.bijective** {I : Type} {φ : I → E →+* ℂ} (h : IsCMType φ) : Function.Bijective (fun i => InfinitePlace.mk (φ i))

theorem **not_isCMType_empty** (φ : Empty → E →+* ℂ) : ¬ IsCMType φ

theorem **expIn_eq_zero** {𝔣 : Ideal (OF E)} {v : HeightOneSpectrum (OF E)} : expIn 𝔣 v = 0 ↔ ¬ v.asIdeal ∣ 𝔣

theorem **expIn_pow** (v : HeightOneSpectrum (OF E)) (n : ℕ) : expIn (v.asIdeal ^ n) v = n

theorem **levelLe_one** (P : HeightOneSpectrum (𝒪 E)) (n : ℕ) : LevelLe (1 : HChar E) P n

theorem **isValChar_one** {I : Type} [Fintype I] (φ : I → E →+* ℂ) (Q : Finset (HeightOneSpectrum (OF E))) : IsValChar φ 0 (fun _ => 0) ⊤ Q (1 : HChar E)

theorem **exists_over** (v : HeightOneSpectrum (OF E)) : ∃ P : HeightOneSpectrum (𝒪 E), Over P v

theorem **exists_ramified** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 : Ideal (OF E)} {Q : Finset (HeightOneSpectrum (OF E))} {χ : HChar E} (hf : IsBaseCond 𝔣) (hQ : IsFinSetSplitPlacesCoprimeConductor 𝔣 Q) (h : IsValChar φ s κ 𝔣 Q χ) : ∃ P, ¬ UnrAt χ P

theorem **not_isValChar_one** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 : Ideal (OF E)} {Q : Finset (HeightOneSpectrum (OF E))} (hf : IsBaseCond 𝔣) (hQ : IsFinSetSplitPlacesCoprimeConductor 𝔣 Q) : ¬ IsValChar φ s κ 𝔣 Q (1 : HChar E)

theorem **IntAt.add** {l : ℕ} {a b : ℂ} (ha : IntAt l a) (hb : IntAt l b) : IntAt l (a + b)

theorem **IntAt.mul** {l : ℕ} {a b : ℂ} (ha : IntAt l a) (hb : IntAt l b) : IntAt l (a ∗ b)

theorem **unitAt_natCast** {l m : ℕ} (hm : m ≠ 0) (h : m.Coprime l) : UnitAt l (m : ℂ)

theorem **not_intAt_inv** {l : ℕ} (hl : l.Prime) : ¬ IntAt l (l : ℂ)⁻¹

theorem **not_unitAt_self** {l : ℕ} (hl : l.Prime) : ¬ UnitAt l (l : ℂ)

theorem **lInf_eq** {I : Type} [Fintype I] (m : I → ℤ) (s : ℂ) : LInf m s = ∏ i, Complex.Gammaℂ (s + (|m i| : ℤ))

theorem **LevelLe.mono** {χ : HChar E} {P : HeightOneSpectrum (𝒪 E)} {m n : ℕ} (h : LevelLe χ P m) (hmn : m ≤ n) : LevelLe χ P n

theorem **expAt_zero_iff** {χ : HChar E} {P : HeightOneSpectrum (𝒪 E)} : ExpAt χ P 0 ↔ UnrAt χ P

theorem **levelLe_inv** {χ : HChar E} {P : HeightOneSpectrum (𝒪 E)} {n : ℕ} : LevelLe χ⁻¹ P n ↔ LevelLe χ P n

theorem **IsValChar.inv** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 : Ideal (OF E)} {Q : Finset (HeightOneSpectrum (OF E))} {χ : HChar E} (h : IsValChar φ s κ 𝔣 Q χ) : IsValChar (fun i => ComplexEmbedding.conjugate (φ i)) s κ 𝔣 Q χ⁻¹

theorem **IsValChar.unrAt** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 : Ideal (OF E)} {Q : Finset (HeightOneSpectrum (OF E))} {χ : HChar E} (h : IsValChar φ s κ 𝔣 Q χ) {v : HeightOneSpectrum (OF E)} {P : HeightOneSpectrum (𝒪 E)} (hP : Over P v) (hQ : v ∉ Q) (hf : ¬ v.asIdeal ∣ 𝔣) : UnrAt χ P

theorem **IsValChar.of_empty** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 : Ideal (OF E)} {Q : Finset (HeightOneSpectrum (OF E))} {χ : HChar E} (hQ : IsFinSetSplitPlacesCoprimeConductor 𝔣 Q) (h : IsValChar φ s κ 𝔣 ∅ χ) : IsValChar φ s κ 𝔣 Q χ

theorem **IsValChar.expIn_eq** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 𝔤 : Ideal (OF E)} {χ : HChar E} (h : IsValChar φ s κ 𝔣 ∅ χ) (h' : IsValChar φ s κ 𝔤 ∅ χ) (v : HeightOneSpectrum (OF E)) : expIn 𝔣 v = expIn 𝔤 v

theorem **IsValChar.inf_eq** {I : Type} [Fintype I] {φ : I → E →+* ℂ} {s : ℕ} {κ : I → ℕ} {𝔣 : Ideal (OF E)} {Q Q' : Finset (HeightOneSpectrum (OF E))} {χ χ' : HChar E} (h : IsValChar φ s κ 𝔣 Q χ) (h' : IsValChar φ s κ 𝔣 Q' χ') (k : Eˣ) : (χ ∗ χ'⁻¹).inf k = 1

theorem **condQ_empty** (𝔣 : Ideal (OF E)) : condQ 𝔣 ∅ = 𝔣

theorem **splits_of_absNorm** {l : ℕ} (hl : l.Prime) (P : HeightOneSpectrum (𝒪 E)) (v : HeightOneSpectrum (OF E)) (hv : Over P v) (hN : Ideal.absNorm P.asIdeal = l) (x : 𝒪 E) (hx : x - cj x ∉ P.asIdeal) : Splits v

theorem **sub_conj_notMem** {z : 𝒪 E} (hz : IsPrimitiveRoot z 21) {P : Ideal (𝒪 E)} (hP : P ≠ ⊤) : z - cj z ∉ P

theorem **zeta_prim** : IsPrimitiveRoot (IsCyclotomicExtension.zeta_spec 21 ℚ K).toInteger 21

theorem **unramFin_K** : UnramFin K

theorem **absNorm_of_mem** (P : HeightOneSpectrum (𝒪 K)) (h : ((43 : ℕ) : 𝒪 K) ∈ P.asIdeal) : Ideal.absNorm P.asIdeal = 43

theorem **not_dvd_discr** : ¬ ((43 : ℕ) : ℤ) ∣ discr K

theorem **splits_of_mem** (v : HeightOneSpectrum (OF K)) (h : ((43 : ℕ) : OF K) ∈ v.asIdeal) : Splits v

theorem **exists_place** : ∃ v : HeightOneSpectrum (OF K), ((43 : ℕ) : OF K) ∈ v.asIdeal

theorem **isBaseCond_pow** (v : HeightOneSpectrum (OF K)) (h : ((43 : ℕ) : OF K) ∈ v.asIdeal) (n : ℕ) : IsBaseCond (v.asIdeal ^ n) ↔ n = 1 ∨ n = 2

theorem **inhabited_K** : UnramFin K ∧ IsCMType (fun w : InfinitePlace K => w.embedding) ∧ ∃ v : HeightOneSpectrum (OF K), Splits v ∧ IsBaseCond v.asIdeal ∧ IsBaseCond (v.asIdeal ^ 2) ∧ ¬ IsBaseCond (v.asIdeal ^ 3) ∧ IsFinSetSplitPlacesCoprimeConductor v.asIdeal ∅ ∧ ¬ IsFinSetSplitPlacesCoprimeConductor v.asIdeal {v} ∧ IsFinSetSplitPlacesCoprimeConductor ⊤ {v}

theorem **exists_isValChar_K** (s : ℕ) (κ : InfinitePlace K → ℕ) : ∃ v : HeightOneSpectrum (OF K), IsBaseCond (v.asIdeal ^ 2) ∧ IsFinSetSplitPlacesCoprimeConductor (v.asIdeal ^ 2) ∅ ∧ ∃ χ : HChar K, IsValChar (fun w : InfinitePlace K => w.embedding) s κ (v.asIdeal ^ 2) ∅ χ ∧ ∃ P, ¬ UnrAt χ P

theorem **absNorm_of_mem'** (P : HeightOneSpectrum (𝒪 K)) (h : ((127 : ℕ) : 𝒪 K) ∈ P.asIdeal) : Ideal.absNorm P.asIdeal = 127

theorem **splits_of_mem'** (v : HeightOneSpectrum (OF K)) (h : ((127 : ℕ) : OF K) ∈ v.asIdeal) : Splits v

theorem **exists_place'** : ∃ v : HeightOneSpectrum (OF K), ((127 : ℕ) : OF K) ∈ v.asIdeal

theorem **exists_valuedHeckeChar_ramified** (s : ℕ) (κ : InfinitePlace K → ℕ) : ∃ v q : HeightOneSpectrum (OF K), IsBaseCond (v.asIdeal ^ 2) ∧ IsFinSetSplitPlacesCoprimeConductor (v.asIdeal ^ 2) {q} ∧ ∃ χ : HChar K, IsValChar (fun w : InfinitePlace K => w.embedding) s κ (v.asIdeal ^ 2) {q} χ ∧ ∃ P, ¬ UnrAt χ P

Development.AbelianVarietyTheory.Definitions.Divisors.ExtFormAdjEqNegSecondForm

@[simp] theorem **mdel_apply** (A : E → Matrix m n ℂ) (z ξ : E) (a : m) (b : n) : mdel A z ξ a b = del (fun x => A x a b) z ξ

@[simp] theorem **mdbar_apply** (A : E → Matrix m n ℂ) (z η : E) (a : m) (b : n) : mdbar A z η a b = dbar (fun x => A x a b) z η

@[simp] theorem **mddbar_apply** (A : E → Matrix m n ℂ) (z ξ η : E) (a : m) (b : n) : mddbar A z ξ η a b = ddbar (fun x => A x a b) z ξ η

theorem **mdel_add_mdbar** (A : E → Matrix m n ℂ) (z ξ : E) : mdel A z ξ + mdbar A z ξ = of fun a b => fderiv ℝ (fun x => A x a b) z ξ

theorem **mdel_congr** (h : A =ᶠ[𝒩 z] B) (ξ : E) : mdel A z ξ = mdel B z ξ

theorem **mdbar_congr** (h : A =ᶠ[𝒩 z] B) (ξ : E) : mdbar A z ξ = mdbar B z ξ

@[simp] theorem **mdel_const** (C : Matrix m n ℂ) (z ξ : E) : mdel (fun _ => C) z ξ = 0

@[simp] theorem **mdbar_const** (C : Matrix m n ℂ) (z ξ : E) : mdbar (fun _ => C) z ξ = 0

theorem **mdel_add_vec** (A : E → Matrix m n ℂ) (z ξ ξ' : E) : mdel A z (ξ + ξ') = mdel A z ξ + mdel A z ξ'

theorem **mdel_smul_vec** (A : E → Matrix m n ℂ) (z : E) (c : ℂ) (ξ : E) : mdel A z (c • ξ) = c • mdel A z ξ

theorem **mdbar_add_vec** (A : E → Matrix m n ℂ) (z η η' : E) : mdbar A z (η + η') = mdbar A z η + mdbar A z η'

theorem **mdbar_smul_vec** (A : E → Matrix m n ℂ) (z : E) (c : ℂ) (η : E) : mdbar A z (c • η) = conj c • mdbar A z η

theorem **isSesq_mddbar** (A : E → Matrix m n ℂ) (z : E) : IsSesq (mddbar A z)

theorem **mdel_transpose** (A : E → Matrix m n ℂ) (z ξ : E) : mdel (fun x => (A x)ᵀ) z ξ = (mdel A z ξ)ᵀ

theorem **mdbar_transpose** (A : E → Matrix m n ℂ) (z ξ : E) : mdbar (fun x => (A x)ᵀ) z ξ = (mdbar A z ξ)ᵀ

theorem **mdel_conjTranspose** (A : E → Matrix m n ℂ) (z ξ : E) : mdel (fun x => (A x)ᴴ) z ξ = (mdbar A z ξ)ᴴ

theorem **mdbar_conjTranspose** (A : E → Matrix m n ℂ) (z ξ : E) : mdbar (fun x => (A x)ᴴ) z ξ = (mdel A z ξ)ᴴ

theorem **mdel_neg** (A : E → Matrix m n ℂ) (z ξ : E) : mdel (fun x => -A x) z ξ = -mdel A z ξ

theorem **mdbar_neg** (A : E → Matrix m n ℂ) (z ξ : E) : mdbar (fun x => -A x) z ξ = -mdbar A z ξ

theorem **MContDiffAt.of_le** (h : MContDiffAt k A z) (hjk : j ≤ k) : MContDiffAt j A z

theorem **MContDiffAt.diff** (h : MContDiffAt 1 A z) (a : m) (b : n) : DifferentiableAt ℝ (fun x => A x a b) z

theorem **mContDiffAt_const** (C : Matrix m n ℂ) : MContDiffAt k (fun _ : E => C) z

theorem **MContDiffAt.add** (h : MContDiffAt k A z) (h' : MContDiffAt k B z) : MContDiffAt k (fun x => A x + B x) z

theorem **MContDiffAt.sub** (h : MContDiffAt k A z) (h' : MContDiffAt k B z) : MContDiffAt k (fun x => A x - B x) z

theorem **MContDiffAt.neg** (h : MContDiffAt k A z) : MContDiffAt k (fun x => -A x) z

theorem **MContDiffAt.smul** {f : E → ℂ} (hf : ContDiffAt ℝ k f z) (h : MContDiffAt k A z) : MContDiffAt k (fun x => f x • A x) z

theorem **MContDiffAt.transpose** (h : MContDiffAt k A z) : MContDiffAt k (fun x => (A x)ᵀ) z

theorem **MContDiffAt.conjTranspose** (h : MContDiffAt k A z) : MContDiffAt k (fun x => (A x)ᴴ) z

theorem **MContDiffAt.submatrix** {m' n' : Type*} (h : MContDiffAt k A z) (f : m' → m) (g : n' → n) : MContDiffAt k (fun x => (A x).submatrix f g) z

theorem **MContDiffAt.mul** [Fintype n] {B : E → Matrix n l ℂ} (h : MContDiffAt k A z) (h' : MContDiffAt k B z) : MContDiffAt k (fun x => A x * B x) z

theorem **MContDiffAt.det** [Fintype n] [DecidableEq n] {A : E → Matrix n n ℂ} (h : MContDiffAt k A z) : ContDiffAt ℝ k (fun x => (A x).det) z

theorem **MContDiffAt.inv** [Fintype n] [DecidableEq n] {A : E → Matrix n n ℂ} (h : MContDiffAt k A z) (hu : IsUnit (A z).det) : MContDiffAt k (fun x => (A x)⁻¹) z

theorem **MContDiffAt.eventually_isUnit** [Fintype n] [DecidableEq n] {A : E → Matrix n n ℂ} (h : MContDiffAt k A z) (hu : IsUnit (A z).det) : ∀ᶠ x in 𝒩 z, IsUnit (A x).det

theorem **MContDiffAt.eventually_two** [Finite m] [Finite n] (h : MContDiffAt 2 A z) : ∀ᶠ x in 𝒩 z, MContDiffAt 2 A x

theorem **MContDiffAt.mdel** (h : MContDiffAt k A z) (hjk : j + 1 ≤ k) (ξ : E) : MContDiffAt j (fun x => mdel A x ξ) z

theorem **MContDiffAt.mdbar** (h : MContDiffAt k A z) (hjk : j + 1 ≤ k) (η : E) : MContDiffAt j (fun x => mdbar A x η) z

theorem **mdel_add** (h : MContDiffAt 1 A z) (h' : MContDiffAt 1 B z) (ξ : E) : mdel (fun x => A x + B x) z ξ = mdel A z ξ + mdel B z ξ

theorem **mdbar_add** (h : MContDiffAt 1 A z) (h' : MContDiffAt 1 B z) (ξ : E) : mdbar (fun x => A x + B x) z ξ = mdbar A z ξ + mdbar B z ξ

theorem **mdel_smul** {f : E → ℂ} (hf : DifferentiableAt ℝ f z) (h : MContDiffAt 1 A z) (ξ : E) : mdel (fun x => f x • A x) z ξ = f z • mdel A z ξ + del f z ξ • A z

theorem **mdbar_smul** {f : E → ℂ} (hf : DifferentiableAt ℝ f z) (h : MContDiffAt 1 A z) (ξ : E) : mdbar (fun x => f x • A x) z ξ = f z • mdbar A z ξ + dbar f z ξ • A z

theorem **mdel_mul** [Fintype n] {B : E → Matrix n l ℂ} (h : MContDiffAt 1 A z) (h' : MContDiffAt 1 B z) (ξ : E) : mdel (fun x => A x * B x) z ξ = A z * mdel B z ξ + mdel A z ξ * B z

theorem **mdbar_mul** [Fintype n] {B : E → Matrix n l ℂ} (h : MContDiffAt 1 A z) (h' : MContDiffAt 1 B z) (ξ : E) : mdbar (fun x => A x * B x) z ξ = A z * mdbar B z ξ + mdbar A z ξ * B z

theorem **mdel_inv** [Fintype n] [DecidableEq n] {A : E → Matrix n n ℂ} (h : MContDiffAt 1 A z) (hu : IsUnit (A z).det) (ξ : E) : mdel (fun x => (A x)⁻¹) z ξ = -((A z)⁻¹ * mdel A z ξ * (A z)⁻¹)

theorem **mdbar_inv** [Fintype n] [DecidableEq n] {A : E → Matrix n n ℂ} (h : MContDiffAt 1 A z) (hu : IsUnit (A z).det) (ξ : E) : mdbar (fun x => (A x)⁻¹) z ξ = -((A z)⁻¹ * mdbar A z ξ * (A z)⁻¹)

theorem **mdbar_mdel** (h : MContDiffAt 2 A z) (ξ η : E) : mdbar (fun x => mdel A x ξ) z η = mddbar A z ξ η

theorem **mdel_mdbar** (h : MContDiffAt 2 A z) (ξ η : E) : mdel (fun x => mdbar A x η) z ξ = mddbar A z ξ η

theorem **conjTranspose_mdel** {A : E → Matrix n n ℂ} (h : ∀ᶠ x in 𝓝 z, (A x).IsHermitian) (ξ : E) : (mdel A z ξ)ᴴ = mdbar A z ξ

theorem **conjTranspose_mddbar** {A : E → Matrix n n ℂ} (h2 : MContDiffAt 2 A z) (h : ∀ᶠ x in 𝓝 z, (A x).IsHermitian) (ξ η : E) : (mddbar A z ξ η)ᴴ = mddbar A z η ξ

theorem **IsPosFrame.isHermFrame** (h : IsPosFrame U H) : IsHermFrame U H

theorem **IsHermFrame.mContDiffAt** (h : IsHermFrame U H) (hz : z ∈ U) : MContDiffAt ∞ H z

theorem **IsHermFrame.two** (h : IsHermFrame U H) (hz : z ∈ U) : MContDiffAt 2 H z

theorem **IsHermFrame.one** (h : IsHermFrame U H) (hz : z ∈ U) : MContDiffAt 1 H z

theorem **IsHermFrame.eventually_herm** (h : IsHermFrame U H) (hz : z ∈ U) : ∀ᶠ x in 𝓝 z, (H x).IsHermitian

theorem **IsHermFrame.of_mContDiffAt** (hU : IsOpen U) (h : ∀ z ∈ U, MContDiffAt ∞ H z) (h1 : ∀ z ∈ U, (H z).IsHermitian) (h2 : ∀ z ∈ U, IsUnit (H z).det) : IsHermFrame U H

theorem **IsHermFrame.isPosFrame** (h : IsHermFrame U H) (h' : ∀ z ∈ U, (H z).PosDef) : IsPosFrame U H

theorem **IsHermFrame.dual** (h : IsHermFrame U H) : IsHermFrame U fun x => dualGram (H x)

theorem **conn_eq_of_isSymm** (h : ∀ᶠ x in 𝓝 z, (H x).IsSymm) (ξ : E) : conn H z ξ = (H z)⁻¹ * mdel H z ξ

theorem **conn_transpose_mul** (hu : IsUnit (H z).det) (ξ : E) : (conn H z ξ)ᵀ * H z = mdel H z ξ

theorem **transpose_mul_conn** (hu : IsUnit (H z).det) (ξ : E) : (H z)ᵀ * conn H z ξ = (mdel H z ξ)ᵀ

theorem **eq_conn_of_compatible** (hu : IsUnit (H z).det) {ξ : E} {T : Matrix n n ℂ} (h : Tᵀ * H z = mdel H z ξ) : T = conn H z ξ

theorem **del_form** (hH : MContDiffAt 1 H z) (hu : IsUnit (H z).det) {s t : E → n → ℂ} (hs : ∀ a, ContDiffAt ℝ 1 (fun x => s x a) z) (ht : ∀ a, ContDiffAt ℝ 1 (fun x => t x a) z) (ξ : E) : del (fun x => form (H x) (s x) (t x)) z ξ = form (H z) (covDel H s z ξ) (t z) + form (H z) (s z) fun a => dbar (fun x => t x a) z ξ

theorem **conn_add_vec** (H : E → Matrix n n ℂ) (z ξ ξ' : E) : conn H z (ξ + ξ') = conn H z ξ + conn H z ξ'

theorem **conn_smul_vec** (H : E → Matrix n n ℂ) (z : E) (c : ℂ) (ξ : E) : conn H z (c • ξ) = c • conn H z ξ

theorem **conn_congr** (h : H =ᶠ[𝓝 z] H') (ξ : E) : conn H z ξ = conn H' z ξ

theorem **curv_congr** (h : H =ᶠ[𝓝 z] H') (ξ η : E) : curv H z ξ η = curv H' z ξ η

theorem **MContDiffAt.conn** (h : MContDiffAt k H z) (hu : IsUnit (H z).det) (hjk : j + 1 ≤ k) (ξ : E) : MContDiffAt j (fun x => conn H x ξ) z

theorem **MContDiffAt.curv** (h : MContDiffAt k H z) (hu : IsUnit (H z).det) (hjk : j + 2 ≤ k) (ξ η : E) : MContDiffAt j (fun x => curv H x ξ η) z

@[simp] theorem **conn_const** (C : Matrix n n ℂ) (z ξ : E) : conn (fun _ => C) z ξ = 0

@[simp] theorem **curv_const** (C : Matrix n n ℂ) (z ξ η : E) : curv (fun _ => C) z ξ η = 0

theorem **curv_eq** (h : MContDiffAt 2 H z) (hu : IsUnit (H z).det) (ξ η : E) : curv H z ξ η = (H z)ᵀ⁻¹ * (mdbar H z η)ᵀ * (H z)ᵀ⁻¹ * (mdel H z ξ)ᵀ - (H z)ᵀ⁻¹ * (mddbar H z ξ η)ᵀ

theorem **isSesq_curv** (h : MContDiffAt 2 H z) (hu : IsUnit (H z).det) : IsSesq (curv H z)

theorem **adj_curv** (h : MContDiffAt 2 H z) (hh : ∀ᶠ x in 𝓝 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (ξ η : E) : adj (H z) (H z) (curv H z ξ η) = curv H z η ξ

theorem **form_curv** (h : MContDiffAt 2 H z) (hh : ∀ᶠ x in 𝓝 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (ξ η : E) (v w : n → ℂ) : form (H z) (curv H z ξ η *ᵥ v) w = form (H z) v (curv H z η ξ *ᵥ w)

theorem **conn_smul** {f : E → ℝ} (hf : DifferentiableAt ℝ f z) (hf0 : f z ≠ 0) (h : MContDiffAt 1 H z) (hu : IsUnit (H z).det) (ξ : E) : conn (fun x => (f x : ℂ) • H x) z ξ = conn H z ξ + del (fun x => (Real.log (f x) : ℂ)) z ξ • (1 : Matrix n n ℂ)

theorem **curv_smul** {f : E → ℝ} (hf : ContDiffAt ℝ 2 f z) (hf0 : f z ≠ 0) (h : MContDiffAt 2 H z) (hu : IsUnit (H z).det) (ξ η : E) : curv (fun x => (f x : ℂ) • H x) z ξ η = curv H z ξ η - leviForm (fun x => Real.log (f x)) z ξ η • (1 : Matrix n n ℂ)

theorem **curv_line** [Unique n] {f : E → ℝ} (hf : ContDiffAt ℝ 2 f z) (hf0 : f z ≠ 0) (hH : ∀ᶠ x in 𝓝 z, H x default default = f x) (ξ η : E) (a b : n) : curv H z ξ η a b = -leviForm (fun x => Real.log (f x)) z ξ η

theorem **conn_dual** (h : MContDiffAt 1 H z) (hu : IsUnit (H z).det) (ξ : E) : conn (fun x => dualGram (H x)) z ξ = -(conn H z ξ)ᵀ

theorem **curv_dual** (h : MContDiffAt 2 H z) (hu : IsUnit (H z).det) (ξ η : E) : curv (fun x => dualGram (H x)) z ξ η = -(curv H z ξ η)ᵀ

theorem **conn_changeFrame** {A : E → Matrix n n ℂ} (h : MContDiffAt 1 H z) (hu : IsUnit (H z).det) (hA : MContDiffAt 1 A z) (hAu : IsUnit (A z).det) (hA0 : ∀ η, mdbar A z η = 0) (ξ : E) : conn (fun x => (A x)ᵀ * H x * (A x)ᴴᵀ) z ξ = (A z)⁻¹ * (conn H z ξ * A z + mdel A z ξ)

theorem **curv_changeFrame** {A : E → Matrix n n ℂ} (h : MContDiffAt 2 H z) (hu : IsUnit (H z).det) (hA : MContDiffAt 2 A z) (hAu : IsUnit (A z).det) (hA0 : ∀ᶠ x in 𝓝 z, ∀ η, mdbar A x η = 0) (ξ η : E) : curv (fun x => (A x)ᵀ * H x * (A x)ᴴᵀ) z ξ η = (A z)⁻¹ * curv H z ξ η * A z

theorem **conn_comp_holo** (h : MContDiffAt 1 H (φ w)) (hφ : DifferentiableAt ℂ φ w) (ξ : E') : conn (fun x => H (φ x)) w ξ = conn H (φ w) (fderiv ℂ φ w ξ)

theorem **curv_comp_holo** (h : MContDiffAt 2 H (φ w)) (hu : IsUnit (H (φ w)).det) (hφ : ContDiffAt ℂ 2 φ w) (ξ η : E') : curv (fun x => H (φ x)) w ξ η = curv H (φ w) (fderiv ℂ φ w ξ) (fderiv ℂ φ w η)

theorem **formAbsLe_iff** {V : Type*} {α β : V → V → ℂ} {C : ℝ} : FormAbsLe α C β ↔ ∀ ξ, |diagVal α ξ| ≤ C * diagVal β ξ

theorem **diagVal_formOf** (G : Matrix ι ι ℂ) (ξ : ι → ℂ) : diagVal (formOf G) ξ = (form G ξ ξ).re

theorem **diagVal_formOf_eq_gnorm_sq** {G : Matrix ι ι ℂ} (hG : G.PosSemidef) (ξ : ι → ℂ) : diagVal (formOf G) ξ = gnorm G ξ ^ 2

theorem **leviForm_self_eq_levi** {u : (ι → ℂ) → ℝ} {z : ι → ℂ} (hu : ContDiffAt ℝ 2 u z) (ξ : ι → ℂ) : leviForm u z ξ ξ = ReciprocityTheory.AnForm.levi u z ξ

theorem **diagVal_ddc** {u : (ι → ℂ) → ℝ} {z : ι → ℂ} (hu : ContDiffAt ℝ 2 u z) (ξ : ι → ℂ) : diagVal (ddc u z) ξ = ReciprocityTheory.AnForm.levi u z ξ / (2 * Real.pi)

theorem **leviForm_eq_form_hessC** {D : ℕ} {u : (Fin D → ℂ) → ℝ} {z : Fin D → ℂ} (hu : ContDiffAt ℝ 2 u z) (ξ η : Fin D → ℂ) : leviForm u z ξ η = form (ReciprocityTheory.AnForm.hessC u z) ξ η

theorem **OneFormBound.mono** {A : (ι → ℂ) → (ι → ℂ) → Matrix n n ℂ} (h : OneFormBound U H G A C) (hC : C ≤ C') : OneFormBound U H G A C'

theorem **TwoFormBound.mono** {F : (ι → ℂ) → (ι → ℂ) → (ι → ℂ) → Matrix n n ℂ} (h : TwoFormBound U H G F C) (hC : C ≤ C') : TwoFormBound U H G F C'

theorem **TwoFormBound.norm_form_le** {F : (ι → ℂ) → (ι → ℂ) → (ι → ℂ) → Matrix n n ℂ} (h : TwoFormBound U H G F C) {z : ι → ℂ} (hz : z ∈ U) (ξ : ι → ℂ) (v : n → ℂ) : ‖form (H z) (F z ξ ξ *ᵥ v) v‖ ≤ C * gnorm (G z) ξ ^ 2 * gnorm (H z) v ^ 2

theorem **twoFormBound_curv_two_mul** [DecidableEq n] (hH : IsPosFrame U H) (hG : ∀ z ∈ U, (G z).PosSemidef) (hC : 0 ≤ C) (h : ∀ z ∈ U, ∀ ξ v, ‖form (H z) (curv H z ξ ξ *ᵥ v) v‖ ≤ C * gnorm (G z) ξ ^ 2 * gnorm (H z) v ^ 2) : TwoFormBound U H G (curv H) (2 * C)

theorem **wedge_comm** {V : Type*} (α β : V → V → ℂ) : wedge α β = wedge β α

theorem **lineCurv_smul** (H : E → Matrix n n ℂ) (z : E) (w : n → ℂ) {c : ℂ} (hc : c ≠ 0) : lineCurv H z (c • w) = lineCurv H z w

theorem **curvMinor_comm** (H : E → Matrix n n ℂ) (z : E) (a b : n) : curvMinor H z a b = curvMinor H z b a

theorem **curvMinor_self** (H : E → Matrix n n ℂ) (z : E) (a : n) (ξ₁ η₁ ξ₂ η₂ : E) : curvMinor H z a a ξ₁ η₁ ξ₂ η₂ = 0

theorem **chTwo_eq** (H : E → Matrix n n ℂ) (z ξ₁ η₁ ξ₂ η₂ : E) : chTwo H z ξ₁ η₁ ξ₂ η₂ = (wedge (chernOne H z) (chernOne H z) ξ₁ η₁ ξ₂ η₂ − 2 * chernTwo H z ξ₁ η₁ ξ₂ η₂) / 2

theorem **chernOne_line** [Unique n] {H : E → Matrix n n ℂ} {z : E} {f : E → ℝ} (hf : ContDiffAt ℝ 2 f z) (hf0 : f z ≠ 0) (hH : ∀ᶠ x in 𝒩 z, H x default default = f x) (ξ η : E) : chernOne H z ξ η = −ddc (fun x => Real.log (f x)) z ξ η

theorem **MContDiffAt.subGram** (h : MContDiffAt k H z) : MContDiffAt k (fun x => subGram (H x)) z

theorem **MContDiffAt.quotGram** (h : MContDiffAt k H z) (hu : IsUnit (HermFrame.subGram (H z)).det) : MContDiffAt k (fun x => quotGram (H x)) z

theorem **IsHermFrame.sub** (h : IsHermFrame U H) (hu : ∀ z ∈ U, IsUnit (subGram (H z)).det) : IsHermFrame U fun x => subGram (H x)

theorem **IsHermFrame.quot** (h : IsHermFrame U H) (hu : ∀ z ∈ U, IsUnit (subGram (H z)).det) : IsHermFrame U fun x => quotGram (H x)

theorem **IsPosFrame.sub** (h : IsPosFrame U H) : IsPosFrame U fun x => subGram (H x)

theorem **IsPosFrame.quot** (h : IsPosFrame U H) : IsPosFrame U fun x => quotGram (H x)

theorem **secondForm_eq** (H : E → Matrix (m ⊕ l) (m ⊕ l) ℂ) (z ξ : E) : secondForm H z ξ = (conn H z ξ).toBlocks₂₁

theorem **form_secondForm** (hh : (H z).IsHermitian) (hu : IsUnit (subGram (H z)).det) (ξ : E) (s : m → ℂ) (q : l → ℂ) : form (quotGram (H z)) (secondForm H z ξ *ᵥ s) q = form (subGram (H z)) s (secondFormAdj H z ξ *ᵥ q)

theorem **form_secondFormAdj_mul** (hh : (H z).IsHermitian) (hu : IsUnit (subGram (H z)).det) (ξ : E) (s : m → ℂ) : form (subGram (H z)) (−(secondFormAdj H z ξ * secondForm H z ξ) *ᵥ s) s = −form (quotGram (H z)) (secondForm H z ξ *ᵥ s) (secondForm H z ξ *ᵥ s)

theorem **extForm_eq_neg_mdbar** (H : E → Matrix (m ⊕ l) (m ⊕ l) ℂ) (z η : E) : extForm H z η = −mdbar (fun x => orthCoeff (H x)) z η

theorem **extForm_eq_neg_secondFormAdj** (h : MContDiffAt 1 H z) (hh : ∀ᶠ x in 𝒩 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (η : E) : extForm H z η = −secondFormAdj H z η

theorem **adj_adj** {A : Matrix m m ℂ} {B : Matrix l l ℂ} (hA : A.IsHermitian) (hAu : IsUnit A.det) (hB : B.IsHermitian) (hBu : IsUnit B.det) (T : Matrix l m ℂ) : adj B A (adj A B T) = T

theorem **extFormAdj_eq_neg_secondForm** (h : MContDiffAt 1 H z) (hh : ∀ᶠ x in 𝒩 z, (H x).IsHermitian) (hu : IsUnit (H z).det) (hs : IsUnit (subGram (H z)).det) (ξ : E) : extFormAdj H z ξ = −secondForm H z ξ

theorem **conn_submatrix_equiv** [Fintype n] [DecidableEq n] (K : E → Matrix n n ℂ) (e : m ≃ n) (z ξ : E) : conn (fun x => (K x).submatrix e e) z ξ = (conn K z ξ).submatrix e e

theorem **curv_submatrix_equiv** [Fintype n] [DecidableEq n] (K : E → Matrix n n ℂ) (e : m ≃ n) (z ξ η : E) : curv (fun x => (K x).submatrix e e) z ξ η = (curv K z ξ η).submatrix e e

theorem **contDiff_one_add_normSq** (e : ℕ) : ContDiff ℝ ∞ fun z : Fin e → ℂ => 1 + ∑ k, ‖z k‖ ^ 2

theorem **isPosFrame_tautGram** (e : ℕ) : IsPosFrame Set.univ (tautGram e)

theorem **curv_tautGram_zero** (e : ℕ) (ξ η : Fin e → ℂ) : curv (tautGram e) 0 ξ η = −(∑ k, ξ k * conj (η k)) • (1 : Matrix (Fin 1) (Fin 1) ℂ)

Development.AbelianVarietyTheory.Definitions.Divisors.ExtId

theorem **ext_id** (F : Type) [Field F] [Algebra (FractionRing R) F] : Ext (RingHom.id R) (RingHom.id F)

Development.AbelianVarietyTheory.Definitions.Divisors.FacAlg

theorem **Wit.fac_alg** : Wit.fac.Alg

Development.AbelianVarietyTheory.Definitions.Divisors.FaithfulSMulAlgClosure

theorem **faithfulSMul_algClosure** (A F : Type) [CommRing A] [Field F] [Algebra A F] [IsFractionRing A F] : FaithfulSMul A (AlgebraicClosure F)

theorem **isAlgebraic_algClosure** (A F : Type) [CommRing A] [IsDomain A] [Field F] [Algebra A F] [IsFractionRing A F] : Algebra.IsAlgebraic A (AlgebraicClosure F)

theorem **mem_of_pow_mem** {x : K} {n : ℕ} (hn : 0 < n) (h : x ^ n ∈ integralClosure A K) : x ∈ integralClosure A K

theorem **exists_pow_eq** [IsAlgClosed K] (x : BplusIn A K) {n : ℕ} (hn : 0 < n) : ∃ y : BplusIn A K, y ^ n = x

theorem **not_isUnit_algebraMap** [FaithfulSMul A K] {a : A} (h : ¬ IsUnit a) : ¬ IsUnit (algebraMap A (BplusIn A K) a)

theorem **not_isUnit_natCast** [FaithfulSMul A K] {p : ℕ} (h : ¬ IsUnit (p : A)) : ¬ IsUnit (p : BplusIn A K)

theorem **exists_zeta** [IsAlgClosed K] [CharZero K] (p : ℕ) [Fact p.Prime] : ∃ ζ : ℕ → BplusIn A K, (∀ n, IsPrimitiveRoot (ζ n) (p ^ n)) ∧ ∀ n, ζ (n + 1) ^ p = ζ n

@[simp] theorem **coe_incl** {A' : Type} [CommRing A'] [Algebra A' K] (φ : A →+* A') (h : ∀ a, algebraMap A' K (φ a) = algebraMap A K a) (x : BplusIn A K) : (incl φ h x : K) = x

theorem **AffOp.res_toX** {U U' : AffOp X} (h : U'.1 ≤ U.1) : spMap (AffOp.res h) ≫ U.toX = U'.toX

theorem **AffOp.val_toX** (U : AffOp X) : spMap (algebraMap U.Bp (Kbar X)) ≫ U.toX = genPt X

theorem **isTangent_iff_lieG** (v : epsR 0 ⟶ A) : IsTangent v ↔ Pdiv.ptMap (TrivSqZeroExt.fstHom 0 0 0) ≫ (v : Pdiv.specO 0 (DualNumber 0) ⟶ A) = toUnit (Pdiv.specO 0 0) ≫ η[A]

theorem **isTangent_lieG** (x : Pdiv.LieG A) : IsTangent (G

theorem **pt_one** (n : ℕ) (B : Type) [CommRing B] [Algebra 0 B] (hp : p ≠ 0) : toConv (θ.pt n B (1, one_pow _)) = 1

theorem **ofTower_bijective** (t : G.TorsionTower A) (n : ℕ) (B : Type) [CommRing B] [Algebra 0 B] : Function.Bijective ((ofTower t).pt n B)

theorem **ofTower_isIsoOver** (t : G.TorsionTower A) (W : Set (Sp 0)) : (ofTower t).IsIsoOver W

theorem **ofTower_isOpenNearZero** (t : G.TorsionTower A) : (ofTower t).IsOpenNearZero

theorem **ofTower_intertwines** (t : G.TorsionTower A) (f : A ⟶ A) [IsMonHom f] : (ofTower t).Intertwines f (t.map t f)

theorem **ofTower_tanTors** (t : G.TorsionTower A) (n : ℕ) (x : Pdiv.LieG A) : (ofTower t).pt n _ (tanTors p n x) = t.tanPt (isTangent_lieG x) n

theorem **gmCoord_gmVec** (c : 0) : gmCoord (B

theorem **fst_gmCoord** (x : Pdiv.LieG (Gm (Sp 0))) : (gmCoord (B

theorem **omega_gm_eq** (w : Pdiv.omega (Gm (Sp 0))) : w = w.1 (gmVec 0 1) • gmForm 0

theorem **isOmega_muTower** : (TorsionMap.ofTower (muTower p 0)).IsOmega (LinearMap.toSpanSingleton 0 _ (dlogT p 0) ∘ₗ LinearMap.proj (gmVec 0 1) ∘ₗ (Pdiv.omega _).subtype)

theorem **EF_dualVec** {J : Type} (b : Module.Basis J ℤ Dw.La) (x : Dw.La) (j : J) : Dw.EF x (dualVec b j) = b.repr x j

theorem **nonempty_basis** : Nonempty (Module.Basis (Fin GG.height) ℤ Dw.La)

theorem **toTate_repr** (b : Module.Basis (Fin GG.height) ℤ Dw.La) (x : Dw.La) : (h.constTate b).toTate (fun i => (b.repr x i : ℤ_[p])) = τ x

theorem **stable** (b : Module.Basis (Fin GG.height) ℤ Dw.La) (a : 𝒪 E) : (h.constTate b).Stable (act a) (Matrix.of fun k i => (b.repr (a • b i) k : ℤ_[p]))

theorem **exists_constTate** : ∃ c : ConstTate ε GG, ∀ a, ∃ ρ, c.Stable (act a) ρ

theorem **weil_lattice** (n : ℕ) (k : Type) [Field k] [IsAlgClosed k] [Algebra R k] (hp : (p : k) ≠ 0) (hk : Set.range (spMap (algebraMap R k)) ⊆ abLocus t.G) (z z' : Tors t.G (p ^ n) k) (x y : Dw.La) (u : kˣ) (hz : θ.pt n k z = tatePt (τ x) n k) (hz' : θ.pt n k z' = tatePt (τ y) n k) (hu : (u : k) = algebraMap R k (muRoot (((Dw.EF x y : ℤ) : ℤ_[p]) • ε) n)) : Out.WeilIs t.G t.PP.L ((p ^ n : ℕ) : ℤ) z.1 z'.1 (unitOn u)

theorem **exists_completion** (K : Type) [IsDomain R] [IsIntegrallyClosed R] [Field K] [IsAlgClosed K] [CharZero K] [Algebra R K] [IsFractionRing R K] (hp : ¬ IsUnit (p : R)) : ∃ (S : GlobalTheory.Published.CycPerfd p) (e : AdicCompletion (Ideal.span {(p : R)}) R ≃+* S.R), letI

theorem **exists_pullbacks** {X Z : Scheme.{0}} [IsIntegral X] (g : X ⟶ Z) (tZ : SAB H Z) : ∃ t : ∀ U : AffOp X, SAB H (Sp U.Bp), Nonempty (∀ U, SABMap (t U) tZ (U.toX ≫ g))

theorem **isOfType_obj** {l : Level H} {B k : Type} [CommRing B] [Field k] [IsAlgClosed k] {b : Base l B} (β : B →+* k) (o : Obj Dw l b (spMap β)) : IsOfType Dw o.t

theorem **IsOfType.charZero** {k : Type} [Field k] {t : SAB H (Sp k)} (h : IsOfType Dw t) : CharZero k

theorem **genericType_of_obj** {X Z : Scheme.{0}} [IsIntegral X] {g : X ⟶ Z} {tZ : SAB H Z} {l : Level H} {B : Type} [CommRing B] {b : Base l B} (β : B →+* Kbar X) (o : Obj Dw l b (spMap β)) (m : SABMap o.t tZ (genPt X ≫ g)) : GenericType Dw g tZ

theorem **IsLevelTower.zero_eq** {Y : LocalSystem Xo p W} {i₀ : Idx H p} {S : Scheme.{0}} {y₀ : S ⟶ (Y i₀).X} {i : ℕ → Idx H p} {y : ∀ n, S ⟶ (Y (i n)).X} (h : IsLevelTower Y i₀ y₀ i y) : (i 0).l.A = i₀.l.A ∧ (i 0).l.M' = i₀.l.M' ∧ (i 0).c.F = i₀.c.F

theorem **GroupSt.all** (h : GroupSt) (hg : GenericSt) : GroupAllSt

theorem **GroupAllSt.good** (h : GroupAllSt) : GroupGoodSt

Development.AbelianVarietyTheory.Definitions.Divisors.FderivFnCongr

theorem **_root_.GlobalTheory.AnPer.Hol.fn_of_mem** (f : _root_.GlobalTheory.AnPer.Hol W) {p : E} (h : p ∈ W) : f.fn p = f.1 (p, h)

theorem **_root_.GlobalTheory.AnPer.Hol.ext_fn** {f g : _root_.GlobalTheory.AnPer.Hol W} (h : ∀ p ∈ W, f.fn p = g.fn p) : f = g

theorem **_root_.GlobalTheory.AnPer.Hol.analytic** (f : _root_.GlobalTheory.AnPer.Hol W) : AnalyticOnNhd ℂ f.fn W

theorem **_root_.GlobalTheory.AnPer.Hol.fn_mk** (F : E → ℂ) (h : AnalyticOnNhd ℂ F W) {p : E} (hp : p ∈ W) : (_root_.GlobalTheory.AnPer.Hol.mk F h).fn p = F p

theorem **_root_.GlobalTheory.AnPer.Hol.ev_apply** (p : E) (hp : p ∈ W) (f : _root_.GlobalTheory.AnPer.Hol W) : _root_.GlobalTheory.AnPer.Hol.ev p hp f = f.fn p

theorem **_root_.GlobalTheory.AnPer.Hol.fn_add** (f g : _root_.GlobalTheory.AnPer.Hol W) {p : E} (hp : p ∈ W) : (f + g).fn p = f.fn p + g.fn p

theorem **_root_.GlobalTheory.AnPer.Hol.fn_mul** (f g : _root_.GlobalTheory.AnPer.Hol W) {p : E} (hp : p ∈ W) : (f * g).fn p = f.fn p * g.fn p

theorem **_root_.GlobalTheory.AnPer.Hol.fn_smul** (c : ℂ) (f : _root_.GlobalTheory.AnPer.Hol W) {p : E} (hp : p ∈ W) : (c • f).fn p = c * f.fn p

theorem **_root_.GlobalTheory.AnPer.Hol.fn_algebraMap** (c : ℂ) {p : E} (hp : p ∈ W) : (algebraMap ℂ (_root_.GlobalTheory.AnPer.Hol W) c).fn p = c

theorem **_root_.GlobalTheory.AnPer.Hol.fn_comap** (Φ : E' → E) (hΦ : AnalyticOnNhd ℂ Φ W') (h : ∀ p ∈ W', Φ p ∈ W) (f : _root_.GlobalTheory.AnPer.Hol W) {p : E'} (hp : p ∈ W') : (_root_.GlobalTheory.AnPer.Hol.comap Φ hΦ h f).fn p = f.fn (Φ p)

theorem **_root_.GlobalTheory.AnPer.Hol.fderiv_fn_congr** (f : _root_.GlobalTheory.AnPer.Hol W) {F : E → ℂ} (e : ∀ q ∈ W, f.fn q = F q) {p : E} (hp : p ∈ W) : fderiv ℂ f.fn p = fderiv ℂ F p

theorem **_root_.GlobalTheory.AnPer.Hol.fn_derFun** (v : E) (f : _root_.GlobalTheory.AnPer.Hol W) {p : E} (hp : p ∈ W) : (_root_.GlobalTheory.AnPer.Hol.derFun v f).fn p = fderiv ℂ f.fn p v

theorem **_root_.GlobalTheory.AnPer.Hol.fn_der** (v : E) (f : _root_.GlobalTheory.AnPer.Hol W) {p : E} (hp : p ∈ W) : (_root_.GlobalTheory.AnPer.Hol.der v f).fn p = fderiv ℂ f.fn p v

theorem **_root_.GlobalTheory.AnPer.Hol.fn_der_comap** (Φ : E' → E) (hΦ : AnalyticOnNhd ℂ Φ W') (h : ∀ p ∈ W', Φ p ∈ W) (v : E') (f : _root_.GlobalTheory.AnPer.Hol W) {p : E'} (hp : p ∈ W') : (_root_.GlobalTheory.AnPer.Hol.der v (_root_.GlobalTheory.AnPer.Hol.comap Φ hΦ h f)).fn p = fderiv ℂ f.fn (Φ p) (fderiv ℂ Φ p v)

Development.AbelianVarietyTheory.Definitions.Divisors.FibreRankCompOfSurjective

theorem **finrank_eq_ker_mul_range** (k A B : Type) [Field k] [CommRing A] [CommRing B] [HopfAlgebra k A] [HopfAlgebra k B] [Module.Finite k A] (f : B →ₐc[k] A) : finrank k A = finrank k (A ⧸ kerIdeal f) * finrank k (LinearMap.range (f : B →ₐ[k] A).toLinearMap)

theorem **fibreDim_congr** (e : M ≃ₗ[O] N) (q : PrimeSpectrum O) : fibreDim M q = fibreDim N q

theorem **fibreRank_comp_of_surjective** (g : N →ₗ[O] P) {f : M →ₗ[O] N} (hf : Function.Surjective f) (q : PrimeSpectrum O) : fibreRank (g ∘ₗ f) q = fibreRank g q

theorem **fibreRank_equiv_comp** (e : N ≃ₗ[O] P) (f : M →ₗ[O] N) (q : PrimeSpectrum O) : fibreRank (e.toLinearMap ∘ₗ f) q = fibreRank f q

theorem **fibreDim_dual** (q : PrimeSpectrum O) : fibreDim (CartierDual O M) q = fibreDim M q

theorem **rankAtStalk_dual** (q : PrimeSpectrum O) : rankAtStalk (CartierDual O M) q = rankAtStalk M q

theorem **fibreRank_transpose** (f : M →ₗ[O] N) (q : PrimeSpectrum O) : fibreRank (CartierDual.transpose f) q = fibreRank f q

theorem **fibreDim_eq_ker_mul_range** (f : B →ₐc[O] A) (q : PrimeSpectrum O) : fibreDim A q = fibreDim (A ⧸ kerIdeal f) q * fibreRank (f : B →ₐ[O] A).toLinearMap q

theorem **fibreDim_ne_zero** (q : PrimeSpectrum O) : fibreDim A q ≠ 0

theorem **dual_map_surjective** (f : B →ₐc[O] A) (h : ∀ q : PrimeSpectrum O, fibreDim A q = fibreDim (A ⧸ kerIdeal f) q * fibreDim B q) : Function.Surjective (CartierDual.map f)

theorem **fibreDim_ker_dual** [IsCocomm O A] (g : A →ₐc[O] A) (q : PrimeSpectrum O) : fibreDim (CartierDual O A ⧸ kerIdeal (CartierDual.map g)) q = fibreDim (A ⧸ kerIdeal g) q

theorem **ker_eq_of_fibreDim_eq** {D E : Type} [CommRing D] [CommRing E] [Algebra O D] [Algebra O E] [Module.Finite O D] [Module.Finite O E] [Module.Projective O E] (π : D →ₐ[O] E) (hπ : Function.Surjective π) (I : Ideal D) (hI : I ≤ RingHom.ker π) (h : ∀ q : PrimeSpectrum O, fibreDim (D ⧸ I) q = fibreDim E q) : RingHom.ker π = I

Development.AbelianVarietyTheory.Definitions.Divisors.FiniteNotLocFrOne

theorem **exists_root** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ∃ ξ : E, cc ξ = -ξ ∧ ξ ^ 2 = -(H.d : E) ∧ ∀ τ : E →+* ℂ, τ (H.rt : E) = H.rt0 → (τ ξ).im < 0

theorem **exists_defCase** (p : ℕ) (hp : 1 ≤ p) : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), Dw.p = p ∧ 0 < ξ ∧ Nonempty (Comp Xo c) ∧ DefCase U ξ

theorem **posDef_of_isMetric** {σ : c.F →+* ℂ} {y : FrPt Xo c (CAt σ)} {N : ℝ} {Gr : Mats Dw ℂ} (h : GlobalTheory.LamT.IsMetric y N Gr) (b : E →+* ℂ) : (Gr b).PosDef

theorem **exists_archFr** (hr : 1 ≤ Dw.p + Dw.q) (y₀ : FrPt Xo c c.F) (σ : c.F →+* ℂ) : ∃ N X, ArchFr y₀ σ N X

theorem **archFr_unique** (hr : 1 ≤ Dw.p + Dw.q) {y₀ : FrPt Xo c c.F} {σ : c.F →+* ℂ} {N N' : ℝ} {X X' : Mats Dw ℂ} (h : ArchFr y₀ σ N X) (h' : ArchFr y₀ σ N' X') : N = N' ∧ ∀ τ, ‖(X τ).det‖ = ‖(X' τ).det‖

theorem **denseIn_of_residue** {K : Type} [Field K] (T : CDVR) (io : K →+* T.Fr) (π : T.V) (hπ : Irreducible π) (p : K) (hp : algebraMap T.V T.Fr π = io p) (h1 : ∀ v : T.V, ∃ (k : K) (h : T.V), algebraMap T.V T.Fr v = io k + algebraMap T.V T.Fr (π * h)) : DenseIn T io

theorem **denseIn_of_initial** {K : Type} [Field K] [NumberField K] (w : HeightOneSpectrum (𝒪 K)) (T₀ : CDVR) (io₀ : K →+* T₀.Fr) (hini : ∀ (T : CDVR) (io : K →+* T.Fr) (e : ℕ), OverLa w T.V io e → ∃ (g : T₀.V →+* T.V) (χ : T₀.Fr →+* T.Fr), IsLocalHom g ∧ χ.comp io₀ = io ∧ χ.comp (algebraMap T₀.V T₀.Fr) = (algebraMap T.V T.Fr).comp g ∧ ∀ ϖ : T₀.V, Irreducible ϖ → vV (K'

theorem **DenseIn.matrix** {K : Type} [Field K] {T : CDVR} {io : K →+* T.Fr} (hD : DenseIn T io) {n : Type} [Fintype n] [DecidableEq n] (g : Matrix n n T.Fr) (hg : g.det ≠ 0) : ∃ (B : Matrix n n K) (Y : Matrix n n T.V), B.det ≠ 0 ∧ IsUnit Y.det ∧ g = B.map io * Y.map (algebraMap T.V T.Fr)

theorem **frChg_of_integral** (T : CDVR) (io : c.F →+* T.Fr) : letI

theorem **exists_integral** (hr : 1 ≤ Dw.p + Dw.q) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (y₀ : FrPt Xo c c.F) (w : HeightOneSpectrum (𝒪 c.F)) (T : CDVR) (io : c.F →+* T.Fr) (e : ℕ) (ho : OverLa w T.V io e) : letI

theorem **exists_locFr** (hr : 1 ≤ Dw.p + Dw.q) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (y₀ : FrPt Xo c c.F) (w : HeightOneSpectrum (𝒪 c.F)) : ∃ a B, LocFr y₀ w a B

theorem **finite_not_locFr_one** (hq : Dw.q = 0) (hp : 1 ≤ Dw.p) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (y₀ : FrPt Xo c c.F) : {w | ¬ LocFr y₀ w 1 1}.Finite

theorem **locFr_unique** (hr : 1 ≤ Dw.p + Dw.q) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) {y₀ : FrPt Xo c c.F} {w : HeightOneSpectrum (𝒪 c.F)} {a a' : c.F} {B B' : Mats Dw c.F} (h : LocFr y₀ w a B) (h' : LocFr y₀ w a' B') : vAdd w a = vAdd w a' ∧ ∀ τ, vAdd w (B τ).det = vAdd w (B' τ).det

theorem **nonempty_ptData** (h : DefCase U ξ) (Z : Comp Xo c) : Nonempty (PtData Z)

theorem **deg_eq** (h : DefCase U ξ) {Z : Comp Xo c} (P P' : PtData Z) (e : P.y₀ = P'.y₀) : P.degTh = P'.degTh ∧ ∀ τ, P.degLie τ = P'.degLie τ

theorem **exists_ptData** (p : ℕ) (hp : 1 ≤ p) : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (Z : Comp Xo c), Dw.p = p ∧ Nonempty (PtData Z)

theorem **nonempty_ptData_all** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), DefCase U ξ → ∀ Z : Comp Xo c, Nonempty (PtData Z)

theorem **not_locFr_mul** (hr : 1 ≤ Dw.p + Dw.q) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) {y₀ : FrPt Xo c c.F} {w : HeightOneSpectrum (𝒪 c.F)} {a t : c.F} {B : Mats Dw c.F} (h : LocFr y₀ w a B) (ht : vAdd w t ≠ 0) : ¬ LocFr y₀ w (t * a) B

theorem **tval_eq_of_reads** {Z : Comp Xo c} {y_0 : FrPt Xo c c.F} {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {x x' : TIdx (part Dw a) → c.F} {f : Rul Xo c} (h : Reads Z y_0 a n x f) (h' : Reads Z y_0 a n x' f) (Y : Mats Dw c.F) (hY : ∀ τ, IsUnit (Y τ).det) : tval (part Dw a) x Y = tval (part Dw a) x' Y

Development.AbelianVarietyTheory.Definitions.Divisors.FinitePrimesUnder

theorem **mem_hatId_top** {x : AE D} (h : x ∈ OhE D) : x ∈ hatId D ⊤

theorem **skewLat_top** (N : ℕ) : skewLat D ⊤ N = stdL D N

theorem **pt_unitBlock_mem** (δ : ℂ) (o a d : ℕ) : pt D (tor0 D ⊤ δ) δ (unitBlock D o a d) ∈ La0 D (tor0 D ⊤ δ) a d

theorem **isStdDatum_top** (P : Pair D ⊤ δ t0 Q) (hF : P.F = P.e0) (hJ : P.J = intG D P.d) (o a : ℕ) : IsStdDatum P ⊤ o ∧ IsBlockLevel P ⊤ a

theorem **IsMinPol.eq_minpoly** {β : ℂ} {p : K[X]} (h : IsMinPol σ β p) : p = @minpoly K ℂ _ _ σ.toAlgebra β

theorem **exists_isMinPol** {β : ℂ} (h : IsAlgebraic ℚ β) [Algebra ℚ K] : ∃ p, IsMinPol σ β p

theorem **IsMinPol.ne_zero** {β : ℂ} {p : K[X]} (h : IsMinPol σ β p) (h0 : β ≠ 0) : p.coeff 0 ≠ 0 ∧ p.natDegree ≠ 0

theorem **meanVal_of_mem** {b : K} {p : K[X]} (h : IsMinPol σ (σ b) p) (w : Formalism.PairingDivisionLocalBalance.Pl K) : meanVal_vw w p = vw w b

theorem **prime_unique** {w : Formalism.PairingDivisionLocalBalance.Pl K} {l l' : ℕ} (hl : l.Prime) (hl' : l'.Prime) (h : (l : 𝒪 K) ∈ w.asIdeal) (h' : (l' : 𝒪 K) ∈ w.asIdeal) : l = l'

theorem **finite_primes_under** {p : Formalism.PairingDivisionLocalBalance.Pl K → Prop} (h : {w | p w}.Finite) : {l : ℕ | l.Prime ∧ ∃ w, p w ∧ (l : 𝒪 K) ∈ w.asIdeal}.Finite

theorem **finite_primes_den** {ι : Type} [Finite ι] {n : ι → ℤ → ℕ → ℝ} {u : ι → ℤ} (b : ∀ i s, ¬ (1 ≤ s ∧ s ≤ u i) → n i s = 0) (c : ∀ i s, {l | l.Prime ∧ n i s l ≠ 0}.Finite) : {l | l.Prime ∧ ∃ i s, n i s l ≠ 0}.Finite

theorem **le_add_truncZero** {n a : ℤ} {y : ℝ} (h : (n : ℝ) ≤ a + y) : n ≤ a + truncZero y

theorem **ranks_exact** (i j : ℕ) : botBig D i j + 2 * ((D.R.m − (j + 1)) / 2) = rankV D (i + 1) (j + 1) ∧ botSmall D i j + 2 * ((D.R.m − j) / 2) = rankV D i j ∧ 2 * halfRank D i j = rankV D (i + 1) (j + 1) + rankV D i j

theorem **two_mul_wtA_sub** (C : ℝ) (i j : ℕ) (τ : D.what₂.Tau) : 2 * (wtA D C i j τ − wtA' D C i j τ) = 2 * ((num D C τ).e (j + 1) + cen D C τ (D.R.m − i)) − (botBig D i j + botSmall D i j : ℕ)

theorem **condE_ne_bot** (P : Places L) : condE P ≠ ⊥

theorem **log_levQ** (P : Places L) : Real.log (levQ P) = ∑ v ∈ P.S0, (condExp P v : ℝ) * Real.log (Ideal.absNorm v.asIdeal)

theorem **alphaH_div_betaH** : alphaH L t i j / betaH G i j = nuH G t i j

theorem **chrEq_nu** {$μ_0$: T} (hL : L.Is $μ_0$) : ChrEq (G.nu t i (j + 1)) (nuH G t i j)

theorem **IsTorVal.coord_unique** {s : ℕ} {zd zd' g g'} (h : IsTorVal G t i j s zd g) (h' : IsTorVal G t i j s zd' g') : zd = zd'

theorem **exists_isIntExp** (h : GlobalTheory.NormSum.IntSt) : ∃ C : ℝ, 0 < C ∧ ∀ (D : D7) (T : Type) (L : LevCh D T) (sc : Consts) (G : GridData L sc), GlobalTheory.NormSum.GoodLevel D (levQ G.pl) → 1 ≤ Real.log sc.b → Real.log D.host.rdE ≤ Real.log sc.b → ∃ dl, IsIntExp G C dl

theorem **rq_base** (ij : Fin D.R.m × Fin D.R.m) : rq $μ_0$ ij = 1

theorem **rq_mul** {ij : Fin D.R.m × Fin D.R.m} {t : T} (ht : t = $μ_0$ ∨ ∃ w, μ w = t) {c_0 c : ℂ} (h_0 : IsSquareConst W B ρ $μ_0$ ij.1 ij.2 c_0) (h_1 : IsSquareConst W B ρ t ij.1 ij.2 c) : D.what₂.embLL (rq t ij) * c_0 = c

theorem **delOf_ch** {l : ℕ} (hl : l.Prime) {w : Formalism.PairingDivisionLocalBalance.Pl D.what₂.LL} (hw : (l : 𝒪 D.what₂.LL) ∈ w.asIdeal) : delOf L (ch w) = L.C.delAt l

theorem **exists_isSqOf** {W : G.Weil} {B : Frames G} {ρ : RealMatch W} {C : ℝ} {$μ_0$: T} {μ : Formalism.PairingDivisionLocalBalance.Pl D.what₂.LL → T} {rq : T → Fin D.R.m × Fin D.R.m → D.what₂.LL} {zA zA' : Fin D.R.m × Fin D.R.m → D.what₂.LL} {dl} (h1 : ∀ ij : Fin D.R.m × Fin D.R.m, ∃ c : T → ℂ, c $μ_0$ ≠ 0 ∧ ∀ t, (t = $μ_0$ ∨ ∃ w, μ w = t) → SquareNonzero W B ρ t ij.1 ij.2 ∧ IsSquareConst W B ρ t ij.1 ij.2 (c t) ∧ D.what₂.embLL (rq t ij) * c $μ_0$ = c t) (h2 : ∃ δ t0 : ℂ, ∀ ij : Fin D.R.m × Fin D.R.m, ∃ P P' : GlobalTheory.NormSum.Pair D ⊤ δ t0 (levQ G.pl), IsBottom G $μ_0$ ij.1 ij.2 (condE G.pl) P P' ∧ D.what₂.embLL (zA ij) = (P.norm : ℂ) ∧ D.what₂.embLL (zA' ij) = (P'.norm : ℂ)) (h3 : ∀ (ij : Fin D.R.m × Fin D.R.m) (s : ℕ), 1 ≤ s → s ≤ halfRank D ij.1 ij.2 → ∃ zd g n, IsTorVal G $μ_0$ ij.1 ij.2 s zd g ∧ IsDenom G ij.1 ij.2 s n ∧ {l | l.Prime ∧ n l ≠ 0}.Finite) (h4 : IsIntExp G C dl) (h5 : {w | μ w ≠ $μ_0$}

Development.AbelianVarietyTheory.Definitions.Divisors.FirstChernIntegralEqDegree

theorem **vInfAt_iff_cInfAt** {u : CxPts p σ → ℝ} : VInfAt u y ↔ CInfAt u y

theorem **VInfAt.congr** (h : VInfAt f y) (e : f' =ᶠ[𝒩 y] f) : VInfAt f' y

theorem **vInfAt_const** (c : F) (y : CxPts p σ) : VInfAt (fun _ => c) y

theorem **vInfAt_ev** {W : X.Opens} (g : Γ(X, W)) (hy : y ∈ ptsIn p σ W) : VInfAt (ev p σ W g) y

theorem **VInfAt.comp** (h : VInfAt f y) {Ψ : F → G} {T : Set F} (hT : IsOpen T) (hy : f y ∈ T) (hΨ : ContDiffOn ℝ ∞ Ψ T) : VInfAt (fun z => Ψ (f z)) y

theorem **VInfAt.comp_contDiff** (h : VInfAt f y) {Ψ : F → G} (hΨ : ContDiff ℝ ∞ Ψ) : VInfAt (fun z => Ψ (f z)) y

theorem **VInfAt.prodMk** {f' : CxPts p σ → G} (h : VInfAt f y) (h' : VInfAt f' y) : VInfAt (fun z => (f z, f' z)) y

theorem **vInfAt_pi** {ι : Type} [Fintype ι] {f : CxPts p σ → ι → F} (h : ∀ i, VInfAt (fun z => f z i) y) : VInfAt f y

theorem **VInfAt.continuousAt** (h : VInfAt f y) : ContinuousAt f y

theorem **VInfAt.pull** {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) {z : CxPts q σ} (h : VInfAt f (PtsOver.map π hπ _ z)) : VInfAt (fun w => f (PtsOver.map π hπ _ w)) z

theorem **VInfAt.contDiffAt_chart** (h : VInfAt f y) {D : ℕ} (c : AlgChart p σ D) (hy : y ∈ c.e.source) : ContDiffAt ℝ ∞ (fun v => f (c.e.symm v)) (c.e y)

theorem **mem_ptsIn_iff_of_conj** {y : CxPts p σ} {y' : CxPts p σ'} (h : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) {W : X.Opens} : y' ∈ ptsIn p σ' W ↔ y ∈ ptsIn p σ W

theorem **ev_conj'** {y : CxPts p σ} {y' : CxPts p σ'} (h : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) (W : X.Opens) (g : Γ(X, W)) : ev p σ' W g y' = star (ev p σ W g y)

theorem **crd_eq** (s : Γ(E, V)) (h : W ≤ V) (h' : W ≤ c.U) (hy : y ∈ ptsIn p σ W) (a : Fin r) : c.crd p σ s y a = ev p σ W ((c.isFrame.basis h').repr (resM E h s) a) y

theorem **crd_resM** (s : Γ(E, V)) (h : W ≤ V) (hy : y ∈ ptsIn p σ W) : c.crd p σ (resM E h s) y = c.crd p σ s y

theorem **crd_add** (s t : Γ(E, V)) (y : CxPts p σ) : c.crd p σ (s + t) y = c.crd p σ s y + c.crd p σ t y

theorem **crd_smul** (g : Γ(X, V)) (s : Γ(E, V)) (y : CxPts p σ) : c.crd p σ (g • s) y = ev p σ V g y • c.crd p σ s y

theorem **crd_zero** (y : CxPts p σ) : c.crd p σ (0 : Γ(E, V)) y = 0

theorem **crd_sum** {ι : Type*} (T : Finset ι) (s : ι → Γ(E, V)) (y : CxPts p σ) : c.crd p σ (∑ i ∈ T, s i) y = ∑ i ∈ T, c.crd p σ (s i) y

theorem **crd_frame** (hy : y ∈ ptsIn p σ c.U) (b : Fin r) : c.crd p σ (c.e b) y = Pi.single b 1

theorem **tr_mulVec** (s : Γ(E, V)) (hy : y ∈ ptsIn p σ V) (h' : y ∈ ptsIn p σ c'.U) : c.tr p σ c' y *ᵥ c'.crd p σ s y = c.crd p σ s y

theorem **tr_mul** (h' : y ∈ ptsIn p σ c'.U) (h'' : y ∈ ptsIn p σ c''.U) : c.tr p σ c' y * c'.tr p σ c'' y = c.tr p σ c'' y

theorem **tr_self** (h : y ∈ ptsIn p σ c.U) : c.tr p σ c y = 1

theorem **isUnit_tr** (h : y ∈ ptsIn p σ c.U) (h' : y ∈ ptsIn p σ c'.U) : IsUnit (c.tr p σ c' y)

theorem **det_tr** (h : y ∈ ptsIn p σ c.U) (h' : y ∈ ptsIn p σ c'.U) : (c.tr p σ c' y).det = gv p σ (det E r) c c' y

theorem **crd_conj** {y' : CxPts p σ'} (h : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) (s : Γ(E, V)) : c.crd p σ' s y' = star (c.crd p σ s y)

theorem **tr_conj** {y' : CxPts p σ'} (h : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) : c.tr p σ' c' y' = (c.tr p σ c' y).map star

theorem **vInfAt_crd** (s : Γ(E, V)) (hy : y ∈ ptsIn p σ V) (h : y ∈ ptsIn p σ c.U) (a : Fin r) : VInfAt (fun z => c.crd p σ s z a) y

theorem **crd_pull** (hE : IsVB E r) {q : Y → Sp k} (π : Y → X) (hπ : π ≫ p = q) (s : Γ(E, V)) {T : Y.Opens} (hT : T ≤ π ⁻¹ᵘ V) {z : CxPts q σ} (hz : z ∈ ptsIn q σ T) (hc : z ∈ ptsIn q σ (π ⁻¹ᵘ c.U)) : (c.pull hE π).crd q σ (pullbackSections π E hT s) z = c.crd p σ s (PtsOver.map π hπ _ z)

theorem **tr_dual** (h : y ∈ ptsIn p σ c.U) (h' : y ∈ ptsIn p σ c'.U) : c.dual.tr p σ c'.dual y = (c'.tr p σ c y)ᵀ

theorem **crd_map** {E' : X.Modules} (φ : E ≅ E') (s : Γ(E, V)) (y : CxPts p σ) : (c.map φ).crd p σ (φ.hom.app V s) y = c.crd p σ s y

theorem **tr_of_notMem** (h' : y ∉ ptsIn p σ c'.U) : c.tr p σ c' y = 0

theorem **tr_mul'** (h' : y ∈ ptsIn p σ c'.U) : c.tr p σ c' y * c'.tr p σ c'' y = c.tr p σ c'' y

theorem **tr_pull** (hE : IsVB E r) {q : Y → Sp k} (π : Y → X) (hπ : π ≫ p = q) {z : CxPts q σ} (h : PtsOver.map π hπ _ z ∈ ptsIn p σ c.U) (h' : PtsOver.map π hπ _ z ∈ ptsIn p σ c'.U) : (c.pull hE π).tr q σ (c'.pull hE π) z = c.tr p σ c' (PtsOver.map π hπ _ z)

theorem **tr_map** {E' : X.Modules} (φ : E ≅ E') (y : CxPts p σ) : (c.map φ).tr p σ (c'.map φ) y = c.tr p σ c' y

theorem **tr_ofLB** {L : LB X} (hL : L.IsLB) {i j : L.ι} (hi : y ∈ ptsIn p σ (L.U i)) (hj : y ∈ ptsIn p σ (L.U j)) (a b : Fin 1) : (ofLB hL i).tr p σ (ofLB hL j) y a b = gv p σ L i j y

theorem **form_chg** (A : Matrix n m ℂ) (H : Matrix n n ℂ) (u w : m → ℂ) : form (chg A H) u w = form H (A *ᵥ u) (A *ᵥ w)

theorem **chg_apply** (A : Matrix n m ℂ) (H : Matrix n n ℂ) (a b : m) : chg A H a b = form H (fun i => A i a) fun i => A i b

theorem **chg_chg** (A : Matrix n m ℂ) (B : Matrix m l ℂ) (H : Matrix n n ℂ) : chg B (chg A H) = chg (A * B) H

@[simp] theorem **chg_one** [DecidableEq n] (H : Matrix n n ℂ) : chg 1 H = H

@[simp] theorem **chg_zero** (H : Matrix n n ℂ) : chg (0 : Matrix n m ℂ) H = 0

theorem **chg_smul** (A : Matrix n m ℂ) (t : ℂ) (H : Matrix n n ℂ) : chg A (t • H) = t • chg A H

theorem **chg_map_star** (A : Matrix n m ℂ) (H : Matrix n n ℂ) : chg (A.map star) (H.map star) = (chg A H).map star

theorem **posDef_chg** [DecidableEq n] {A H : Matrix n n ℂ} (hA : IsUnit A) (h : H.PosDef) : (chg A H).PosDef

theorem **det_chg** [DecidableEq n] (A H : Matrix n n ℂ) : (chg A H).det = (‖A.det‖ ^ 2 : ℝ) * H.det

theorem **dualGram_chg** [DecidableEq n] {A B : Matrix n n ℂ} (h : A * B = 1) (H : Matrix n n ℂ) : dualGram (chg A H) = chg Bᵀ (dualGram H)

theorem **posDef_dualGram** [DecidableEq n] {H : Matrix n n ℂ} (h : H.PosDef) : (dualGram H).PosDef

theorem **dualGram_map_star** [DecidableEq n] (H : Matrix n n ℂ) : dualGram (H.map star) = (dualGram H).map star

theorem **det_re_pos** [DecidableEq n] {H : Matrix n n ℂ} (h : H.PosDef) : 0 < H.det.re

theorem **MInfAt.congr** (h : MInfAt A y) (e : A' =ᶠ[𝓝 y] A) : MInfAt A' y

theorem **MInfAt.mContDiffAt** (h : MInfAt A y) {D : ℕ} (c : AlgChart p σ D) (hy : y ∈ c.e.source) : MContDiffAt ∞ (fun v => A (c.e.symm v)) (c.e y)

theorem **MInfAt.comp** (h : MInfAt A y) {F : Type} [NormedAddCommGroup F] [NormedSpace ℝ F] {Ψ : (m → n → ℂ) → F} {T : Set (m → n → ℂ)} (hT : IsOpen T) (hy : A y ∈ T) (hΨ : ∀ M ∈ T, ContDiffAt ℝ ∞ Ψ M) : VInfAt (fun z => Ψ (A z)) y

theorem **mContDiffAt_id** (M : m → n → ℂ) : MContDiffAt ∞ (fun N : m → n → ℂ => Matrix.of N) M

theorem **MInfAt.chg** {H : CxPts p σ → Matrix m m ℂ} (h : MInfAt A y) (hH : MInfAt H y) : MInfAt (fun z => chg (A z) (H z)) y

theorem **isOpen_isUnit_det** [DecidableEq m] : IsOpen {M : m → m → ℂ | IsUnit (of M).det}

theorem **MInfAt.dualGram** [DecidableEq m] {H : CxPts p σ → Matrix m m ℂ} (h : MInfAt H y) (hu : IsUnit (H y).det) : MInfAt (fun z => dualGram (H z)) y

theorem **MInfAt.log_det** [DecidableEq m] {H : CxPts p σ → Matrix m m ℂ} (h : MInfAt H y) (hu : (H y).det.re ≠ 0) : CInfAt (fun z => Real.log (H z).det.re) y

theorem **VB.IsVB.mem_pick** {E : X.Modules} {r : ℕ} (h : IsVB E r) (y : CxPts p σ) : y ∈ ptsIn p σ (h.pick y).U

theorem **form_eq** (c : Chart E r) (hc : y ∈ ptsIn p σ c.U) (s t : Γ(E, V)) (hy : y ∈ ptsIn p σ V) : h.form s t y = HermFrame.form (h.G c y) (c.crd p σ s y) (c.crd p σ t y)

theorem **form_frame** (c : Chart E r) (hc : y ∈ ptsIn p σ c.U) (a b : Fin r) : h.form (c.e a) (c.e b) y = h.G c y a b

theorem **form_resM** (s t : Γ(E, V)) (hW : W ≤ V) (hy : y ∈ ptsIn p σ W) : h.form (resM E hW s) (resM E hW t) y = h.form s t y

theorem **form_add_left** (s s' t : Γ(E, V)) (y : CxPts p σ) : h.form (s + s') t y = h.form s t y + h.form s' t y

theorem **form_smul_left** (g : Γ(X, V)) (s t : Γ(E, V)) (y : CxPts p σ) : h.form (g • s) t y = ev p σ V g y * h.form s t y

theorem **conj_form** (s t : Γ(E, V)) (y : CxPts p σ) : starRingEnd ℂ (h.form s t y) = h.form t s y

theorem **G_of_notMem** (c : Chart E r) (hc : y ∉ ptsIn p σ c.U) : h.G c y = 0

theorem **ext_of_exists** {h h' : Met p σ E r} (H : ∀ y, ∃ c : Chart E r, y ∈ ptsIn p σ c.U ∧ h.G c y = h'.G c y) : h = h'

theorem **ext_form** {h h' : Met p σ E r} (H : ∀ (V : X.Opens) (s t : Γ(E, V)) y, y ∈ ptsIn p σ V → h.form s t y = h'.form s t y) : h = h'

theorem **ofCover_G** {hE : IsVB E r} {ι : Sort*} {c : ι → Chart E r} {hc} {H : ι → CxPts p σ → Matrix (Fin r) (Fin r) ℂ} {hH} (ht : ∀ i j y, y ∈ ptsIn p σ (c i).U → y ∈ ptsIn p σ (c j).U → H j y = chg ((c i).tr p σ (c j) y) (H i y)) {i : ι} (hy : y ∈ ptsIn p σ (c i).U) : (ofCover hE c hc H hH).G (c i) y = H i y

theorem **isCInf_of_exists** (H : ∀ y, ∃ c : Chart E r, y ∈ ptsIn p σ c.U ∧ MInfAt (h.G c) y) : h.IsCInf

theorem **isConj_of_exists** {h' : Met p σ' E r} (H : ∀ (y : CxPts p σ) (y' : CxPts p σ'), y'.1 = spMap (starRingEnd ℂ) ≫ y.1 → ∃ c : Chart E r, y ∈ ptsIn p σ c.U ∧ h'.G c y' = (h.G c y).map star) : h.IsConj h'

theorem **pull_G** {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) (c : Chart E r) {z : CxPts q σ} (hz : PtsOver.map π hπ _ z ∈ ptsIn p σ c.U) : (h.pull π hπ).G (c.pull h.isVB π) z = h.G c (PtsOver.map π hπ _ z)

theorem **form_pull** {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) (s t : Γ(E, V)) {T : Y.Opens} (hT : T ≤ π ⁻¹ᵁ V) {z : CxPts q σ} (hz : z ∈ ptsIn q σ T) : (h.pull π hπ).form (pullbackSections π E hT s) (pullbackSections π E hT t) z = h.form s t (PtsOver.map π hπ _ z)

theorem **dual_G** (c : Chart E r) (hy : y ∈ ptsIn p σ c.U) : h.dual.G c.dual y = dualGram (h.G c y)

theorem **map_G** (φ : E ≅ E') (c : Chart E r) (hy : y ∈ ptsIn p σ c.U) : (h.map φ).G (c.map φ) y = h.G c y

theorem **form_map** (φ : E ≅ E') (s t : Γ(E, V)) (hy : y ∈ ptsIn p σ V) : (h.map φ).form (φ.hom.app V s) (φ.hom.app V t) y = h.form s t y

theorem **form_expMul** (u : CxPts p σ → ℝ) (s t : Γ(E, V)) (y : CxPts p σ) : (h.expMul u).form s t y = Real.exp (u y) ∗ h.form s t y

theorem **ofWeight_G** {L : LB X} (φ : Weight p σ L) {i : L.ι} (hy : y ∈ ptsIn p σ (L.U i)) : (ofWeight φ).G (Chart.ofLB φ.isLB i) y = (Real.exp (-φ.φ i y) : ℂ) • 1

theorem **ofFrame_G** {e : Fin r → Γ(E, ⊤)} (he : IsFrame e) (H : CxPts p σ → Matrix (Fin r) (Fin r) ℂ) (hH : ∀ y, (H y).PosDef) (y : CxPts p σ) : (ofFrame he H hH).G (⊤, e, he) y = H y

theorem **IsCInf.pull** (H : h.IsCInf) {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) : (h.pull π hπ).IsCInf

theorem **IsCInf.dual** (H : h.IsCInf) : h.dual.IsCInf

theorem **IsCInf.map** (H : h.IsCInf) (φ : E ≅ E') : (h.map φ).IsCInf

theorem **IsCInf.expMul** (H : h.IsCInf) {u : CxPts p σ → ℝ} (hu : ∀ y, CInfAt u y) : (h.expMul u).IsCInf

theorem **IsCInf.detWeight** (H : h.IsCInf) : h.detWeight.IsCInf

theorem **isCInf_ofWeight** {L : LB X} {φ : Weight p σ L} (H : φ.IsCInf) : (ofWeight φ).IsCInf

theorem **isCInf_ofFrame** {e : Fin r → Γ(E, ⊤)} (he : IsFrame e) {H : CxPts p σ → Matrix (Fin r) (Fin r) ℂ} (hH : ∀ y, (H y).PosDef) (hs : ∀ y, MInfAt H y) : (ofFrame he H hH).IsCInf

theorem **IsConj.form_eq** (H : h.IsConj h') {V : X.Opens} (s t : Γ(E, V)) {y : CxPts p σ} {y' : CxPts p σ'} (e : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) (hy : y ∈ ptsIn p σ V) : h'.form s t y' = starRingEnd ℂ (h.form s t y)

theorem **IsConj.pull** (H : h.IsConj h') {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) : (h.pull π hπ).IsConj (h'.pull π hπ)

theorem **IsConj.dual** (H : h.IsConj h') : h.dual.IsConj h'.dual

theorem **IsConj.map** (H : h.IsConj h') (φ : E ≅ E') : (h.map φ).IsConj (h'.map φ)

theorem **IsConj.expMul** (H : h.IsConj h') {u : CxPts p σ → ℝ} {u' : CxPts p σ' → ℝ} (hu : ∀ (y : CxPts p σ) (y' : CxPts p σ'), y'.1 = spMap (starRingEnd ℂ) ≫ y.1 → u' y' = u y) : (h.expMul u).IsConj (h'.expMul u')

theorem **det_map_star_re** {n : Type} [Fintype n] [DecidableEq n] (M : Matrix n n ℂ) : (M.map star).det.re = M.det.re

theorem **IsConj.detWeight** (H : h.IsConj h') : h.detWeight.IsConj h'.detWeight

theorem **isConj_ofWeight** {L : LB X} {φ : Weight p σ L} {φ' : Weight p σ' L} (H : φ.IsConj φ') : (ofWeight φ).IsConj (ofWeight φ')

theorem **detWeight_ofWeight** {L : LB X} (φ : Weight p σ L) {i : L.ι} {y : CxPts p σ} (hy : y ∈ ptsIn p σ (L.U i)) : (ofWeight φ).detWeight.φ (Chart.ofLB φ.isLB i) y = φ.φ i y

theorem **isIsometry_iff** {h' : VMetric p E' r} {φ : E ≅ E'} : h.IsIsometry h' φ ↔ h' = h.map φ

theorem **IsCInf.pull** (H : h.IsCInf) {q : Y ⟶ Sp k} (π : Y ⟶ X) (hπ : π ≫ p = q) : (h.pull π hπ).IsCInf

theorem **IsCInf.isCInfNorm** (H : h.IsCInf) : h.IsCInfNorm

theorem **IsCInf.dual** (H : h.IsCInf) : h.dual.IsCInf

theorem **IsCInf.map** (H : h.IsCInf) (φ : E ≅ E') : (h.map φ).IsCInf

theorem **IsCInf.scale** (H : h.IsCInf) (t : ℝ) : (h.scale t).IsCInf

theorem **isCInf_ofLB** (L : HermLB p) : (ofLB L).IsCInf

theorem **isCInf_orthonormal** {e : Fin r → Γ(E, ⊤)} (he : IsFrame e) : (orthonormal p he).IsCInf

theorem **firstChernIntegral_eq_degree** (H : h.IsCInf) : h.first_chern_integral = ArInt.gdeg (det H) 1

Development.AbelianVarietyTheory.Definitions.Divisors.FirstChernLeviEqConstant

theorem **IsBdSec.exp_le** {L : LocPot X} {s : X → X → ℂ} (h : IsBdSec A Z r L s) (x : X) (hx : x ∉ Z) : Real.exp 1 ≤ r x

theorem **IsBd.compl_nonempty** [Nonempty X] (h : IsBd A Z r Om) : Zᶜ.Nonempty

theorem **int_eq** (P : Fin e → M.Pot) (u : M.Fun) : M.int P u = (∑ σ, mInt (M.A σ) (fun i => P i σ) (u σ)) / ((M.n : ℝ) + 1)

theorem **QPolBd.qPol** {F : An M ι N} {d : ι → Fin N → ℝ} (h : F.QPolBd d) : F.QPol d

theorem **TieLim.tiePs** (h : TieLim T A) (hc : T.Conv) : TiePs T A

theorem **TiePs.conv** (h : TiePs T A) : T.Conv

theorem **Real0.Pts.mean_le** {c : Y.Comp} {y : Y.Pt} {g g' : (T.M c).Fun} (h : ∀ σ i, g σ (P.conj c y σ i) ≤ g' σ (P.conj c y σ i)) : P.mean c y g ≤ P.mean c y g'

theorem **Real0.Pts.mean_const** (c : Y.Comp) (y : Y.Pt) (a : ℝ) : P.mean c y (fun _ _ => a) = a

Development.AbelianVarietyTheory.Definitions.Divisors.FlatTensorVacuum

theorem **symN_odd** (ρ : ℕ) : symN k n (2 ∗ ρ + 1) = N k n (2 ∗ ρ + 1) (ρ + 1)

theorem **symN_even** (ρ : ℕ) : symN k n (2 ∗ ρ) = N k n (2 ∗ ρ) ρ ⊔ N k n (2 ∗ ρ) (ρ + 1)

theorem **symN_conj** {u : GL (n ⊕ Fin c) k} (hu : u ∈ symN k n c) (g : GL n k) : (embX c g)⁻¹ ∗ u ∗ embX c g ∈ symN k n c

theorem **Kept.not_free** {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} (h : Kept rs) : ¬ Free (c / 2 + 1) rs

theorem **stepPt_eq** (x₀ : Matrix n n' k) (e : StepVar n c n' → k) : stepPt x₀ e = affPt (c / 2 + 1) x₀ (fun rs => e (rs.1, Or.inl rs.2)) + keptPt fun rs => e (rs.1, Or.inr rs.2)

theorem **relIndex_ne_zero_of_isOpen** {G : Type*} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] {U K H : Subgroup G} (hU : IsOpen (U : Set G)) (hK : IsCompact (K : Set G)) (hH : H ≤ K) : U.relIndex H ≠ 0

theorem **invVol_pos** {C : Subgroup (GL (n ⊕ Fin c) k)} (ho : IsOpen (C : Set (GL (n ⊕ Fin c) k))) (hc : IsCompact (C : Set (GL (n ⊕ Fin c) k))) : 0 < invVol C

theorem **invVol_KO** : invVol (KO k (n ⊕ Fin c)) = 1

theorem **topRank_add** (hs : 1 ≤ G.s) (j : ℕ) : topRank G j + G.m = G.rank (G.m, j) + j + 1

theorem **fill_theta** (x : ColArgs G) (y : RowArgs G) (e : Vars G) (j : ℕ) : fill hs x y e (.theta G.m j) = (downMap hs x e j)ᵀ

theorem **fill_ex** (x : ColArgs G) (y : RowArgs G) (e : Vars G) (i : ℕ) : fill hs x y e (.ex i G.m) = (y i)ᵀ

theorem **onColumn_apply** (h : ∀ j : ℕ, GL (Fin (G.rank (G.m, j))) k) (j : ℕ) : onColumn h (G.m, j) = h j

theorem **toFrame_stair** (ϖ : kˣ) (e : ℕ) {j : ℕ} (hj : j < G.m) : toFrame hs hj (stair G ϖ e j) = diagGL (Sum.elim (fun _ => 1) fun p => ϖ ^ (e * (p.val + 1)))

theorem **moved_false** (g : G.VertexGroup) (T : G.Args → ℂ) (z : G.Args) : moved G false g T z = ((oppositePath G.m).map fun b => apow (gdet (g b.src))⁻¹ ((G.rank b.tgt : ℂ) / 2) * apow (gdet (g b.tgt))⁻¹ (-(G.rank b.src : ℂ) / 2)).prod * T (G.moveArgs .one g z)

theorem **sum_unf** (hfin : (unfSupport ψ ν keep ρ T).Finite) (f : UnfTerm G → ℂ) : ((unf ψ ν keep ρ T hfin).map f).sum = ∑ᶠ β ∈ unfSupport ψ ν keep ρ T, f (unfTerm ψ ν keep ρ T β)

theorem **unfSupport_eq** : unfSupport ψ ν keep ρ T = unfSupport ψ ν true ρ T

theorem **unfTerm_coef_pos** (hJ : ∀ j, 0 < j → j < G.m → IsOpen (J (G.m, j) : Set (GL (Fin (G.rank (G.m, j))) k)) ∧ J (G.m, j) ≤ KO k _) (β : Cosets hs J) : 0 < (unfTerm ψ ν keep ρ T β).coef

theorem **through_elt** (h : ∀ j : ℕ, GL (Fin (G.rank (G.m, j))) k) {j : ℕ} (hj : j + 2 ≤ G.m) : (Reps.through hs J h).elt (fun i => DoubleCoset.mk _ _ (h (i + 1))) (G.m, j) = h j

theorem **flatTensor_vacuum** (hψ : ∀ a ∈ 𝒪[k], ψ a = 1) (hν : IsUnitVol ν) (S : G.Args → ℂ) (h1 : ∀ z, (∀ b ∈ oppositePath G.m, ∀ i j, z b i j ∈ 𝒪[k]) → S z = 1) (h0 : ∀ z, ¬ (∀ b ∈ oppositePath G.m, ∀ i j, z b i j ∈ 𝒪[k]) → S z = 0) : Integrable (flatIntegrand hs ψ S 0 0) (varsVol G ν) ∧ flatTensor hs ψ ν S 0 0 = 1

Development.AbelianVarietyTheory.Definitions.Divisors.FormalCoboundarySmul

theorem **_root_.GlobalTheory.Katz.formalCoboundary_add** {μ : Fin d → MvPowerSeries (Fin d ⊕ Fin d) R} (h : ∀ i, constantCoeff (μ i) = 0) (f g : MvPowerSeries (Fin d) R) : _root_.GlobalTheory.Katz.cob1 μ (f + g) = _root_.GlobalTheory.Katz.cob1 μ f + _root_.GlobalTheory.Katz.cob1 μ g

theorem **_root_.GlobalTheory.Katz.formalCoboundary_smul** {μ : Fin d → MvPowerSeries (Fin d ⊕ Fin d) R} (h : ∀ i, constantCoeff (μ i) = 0) (r : R) (f : MvPowerSeries (Fin d) R) : _root_.GlobalTheory.Katz.cob1 μ (r • f) = r • _root_.GlobalTheory.Katz.cob1 μ f

theorem **coeff_lift** (f : MvPowerSeries σ k) (α : σ →₀ ℕ) : coeff α (lift p f) = algebraMap (WittVector p k) (Fr p k) (WittVector.teichmuller p (coeff α f))

theorem **_root_.GlobalTheory.Katz.constantCoeff_lift** {f : MvPowerSeries σ k} (h : constantCoeff f = 0) : constantCoeff (_root_.GlobalTheory.Katz.lift p f) = 0

theorem **_root_.GlobalTheory.Katz.Chart.nil₂** (n : ℕ) (i : Fin d ⊕ Fin d) : IsNilpotent (c.x₂ n i)

Development.AbelianVarietyTheory.Definitions.Divisors.FormalDegreeUpperC

theorem **AdelC.full** {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] {R : RealDat D A Ks} {F : Face D A} (h : AdelC R F) : AdelicLiftScalar R F

theorem **ConjRatLift5.full** (h : ConjRatLift5) : GlobalTheory.ConjScalar.ConjRatLift

Development.AbelianVarietyTheory.Definitions.Divisors.FrameSignSimple

theorem **frameSign_simple** {R : Type} [CommRing R] (n n' : Matrix (Fin 5) (Fin 5) R) (hn : ∀ i j, j ≤ i → n i j = if i = j then 1 else 0) (hn' : ∀ i j, j ≤ i → n' i j = if i = j then 1 else 0) (h : n'ᵀ * antiDiagonalSignGramMatrix 5 (-1 : R) * n = antiDiagonalSignGramMatrix 5 (-1 : R)) : (n 0 1 - n' 0 1) + (n 1 2 - n' 1 2) = ∑ p : Fin 4, frameSign R 5 p.castSucc * n p.castSucc p.succ * frameSign R 5 p.succ

theorem **readOK** {ι : Type} [Fintype ι] {n : ℕ} (σ : ι → DomWeight n) : ReadOK σ

theorem **covOf_funOf** {ι : Type} [Fintype ι] {n : ℕ} (σ : ι → DomWeight n) (F : VV ℂ σ) : covOf σ (funOf σ F) = F

theorem **funOf_rhoV** {E : Type} [Field E] [NumberField E] {ι : Type} [Fintype ι] {n : ℕ} (emb : ι → E →+* ℂ) (σ : ι → DomWeight n) (γ : GL (Fin n) E) (F : VV ℂ σ) : funOf σ (rhoK σ (embGL emb γ) F) = rhoK (fun τ => (σ τ).dual) (invTrK (embGL emb γ)) (funOf σ F)

theorem **finEmb_eq** {A B : HSp D} {c : ℕ} {ε : D.host.E} (d : HermitianOrthogonalExtensionData A B c ε) (y : GfC A) (x : GfC B) (h : EmbRel B A d.ι x.1 y.1) : finEmb d y = x

theorem **finEmb_mul** {A B : HSp D} {c : ℕ} {ε : D.host.E} (d : HermitianOrthogonalExtensionData A B c ε) (y y' : GfC A) : finEmb d (y * y') = finEmb d y * finEmb d y'

theorem **exists_places** : ∃ PA : Finset (Pl D.host.E), (∀ w ∈ PA, FractionalIdeal.count D.host.E w D.host.AA ≠ 0 ∨ FractionalIdeal.count D.host.E w D.host.AAbar ≠ 0) ∧ ∀ w, FractionalIdeal.count D.host.E w D.host.AA ≠ 0 → ∃! w', w' ∈ PA ∧ w'.under (𝒪 D.host.F) = w.under (𝒪 D.host.F)

theorem **nonempty_empty** {G : GridData L sc} (W : G.Weil) (N : Std G (∅ : Set T)) (Z : Normal W N) : Nonempty (GridChoice W ∅)

Development.AbelianVarietyTheory.Definitions.Divisors.FstTszeSec

theorem **_root_.GlobalTheory.AnPer.Charts.tsze_nil** {O : Type} [CommRing O] [Algebra ℂ O] : IsNilpotent (RingHom.ker (TrivSqZeroExt.fstHom ℂ O (Fin g → O)).toRingHom)

theorem **_root_.GlobalTheory.AnPer.Charts.fst_tszeSec** {O : Type} [CommRing O] [Algebra ℂ O] (D : Fin g → Derivation ℂ O O) (h : O) : (_root_.GlobalTheory.AnPer.Charts.tszeSec D h).fst = h

theorem **_root_.GlobalTheory.AnPer.Charts.snd_tszeSec** {O : Type} [CommRing O] [Algebra ℂ O] (D : Fin g → Derivation ℂ O O) (h : O) (a : Fin g) : (_root_.GlobalTheory.AnPer.Charts.tszeSec D h).snd a = D a h

theorem **_root_.GlobalTheory.AnPer.Charts.tsze_expand** {O : Type} [CommRing O] (h : O) (v : Fin g → O) : (TrivSqZeroExt.inl h + TrivSqZeroExt.inr v : TrivSqZeroExt O (Fin g → O)) = TrivSqZeroExt.inl h + ∑ a, TrivSqZeroExt.inl (v a) * TrivSqZeroExt.inr (Pi.single a (1 : O))

Development.AbelianVarietyTheory.Definitions.Divisors.GalMapMul

theorem **cpre_mk** (a : Cocycle K L m) (b : Cocycle L M n) (h : m + n = k) : cpre a M n k h (CohomologyClass.mk b) = CohomologyClass.mk (zcomp a b h)

theorem **cpost_mk** (a : Cocycle K L m) (b : Cocycle L M n) (h : m + n = k) : cpost K b m k h (CohomologyClass.mk a) = CohomologyClass.mk (zcomp a b h)

theorem **_root_.GlobalTheory.EtC.pull_id** (E : _root_.GlobalTheory.EtC.Spc) (F : _root_.GlobalTheory.EtC.Shv R) (n : ℤ) (x : _root_.GlobalTheory.EtC.H R E F n) : _root_.GlobalTheory.EtC.pull R (𝟙 E) F n x = x

theorem **_root_.GlobalTheory.EtC.pull_comp** {E E' E'' : _root_.GlobalTheory.EtC.Spc} (f : E ⟶ E') (g : E' ⟶ E'') (F : _root_.GlobalTheory.EtC.Shv R) (n : ℤ) (x : _root_.GlobalTheory.EtC.H R E'' F n) : _root_.GlobalTheory.EtC.pull R (f ≫ g) F n x = _root_.GlobalTheory.EtC.pull R f F n (_root_.GlobalTheory.EtC.pull R g F n x)

theorem **_root_.GlobalTheory.EtC.galMap_one** : _root_.GlobalTheory.Et.galMap Ω p 1 = 𝟙 _

theorem **_root_.GlobalTheory.EtC.galMap_mul** (σ τ : Ω ≃ₐ[k] Ω) : _root_.GlobalTheory.Et.galMap Ω p (σ * τ) = _root_.GlobalTheory.Et.galMap Ω p σ ≫ _root_.GlobalTheory.Et.galMap Ω p τ

Development.AbelianVarietyTheory.Definitions.Divisors.GalRigidOfSurjective

theorem **isPPol_of_principal** {S : Scheme.{0}} {h M : ℕ} {B : Type} [CommRing B] {ζ : B} {s : S ⟶ Sp B} (c : SiegPre h M ζ s) (hc : c.Principal) : IsPPol c.A c.P

theorem **isComplAt_id** (K : Type) [Field K] : IsComplAt (RingHom.id K)

theorem **torsRat_one** {R : Type} [CommRing R] [IsDomain R] (G : Over (Sp R)) [GrpObj G] : TorsRat G 1

theorem **galRigid_of_surjective** {A R : Type} [CommRing A] [CommRing R] {f : A →+* R} (hf : Function.Surjective f) : GalRigid f

Development.AbelianVarietyTheory.Definitions.Divisors.GenReads

theorem **CcIs.norm** {θ : InfFam D} {V W : HSp D} {r : ℕ} {hr : W.rank = r} {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {σ : c.F →+* ℂ} {U : Unf Xo c σ r} {dict : GA D W → Σ k, Frame (U.X k).Ψ r} {mu : GA D W → ℂ} {x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ} {om : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} {f₀ : WO V W → ℂ} {cc : U.Comp → ℂ} (h : CcIs θ V W hr U dict mu x₀ om f₀ cc) (k : U.Comp) : ‖cc k‖ = 1

Development.AbelianVarietyTheory.Definitions.Divisors.GenSubHwOfSurjective

theorem **mem_wtSub** {act : Mat R S →ₗ[ℂ] Module.End ℂ V} {μ : Ix R S → ℚ} {v : V} : v ∈ wtSub act μ ↔ ∀ a, act (Eab a a) v = (μ a : ℂ) • v

theorem **mem_hwSub** {act : Mat R S →ₗ[ℂ] Module.End ℂ V} {μ : Ix R S → ℚ} {v : V} : v ∈ hwSub act μ ↔ (∀ a, act (Eab a a) v = (μ a : ℂ) • v) ∧ ∀ a b, posC a b → act (Eab a b) v = 0

theorem **hwSub_le_isoPart** (act : Mat R S →ₗ[ℂ] Module.End ℂ V) (Λ : Ix R S → ℚ) : hwSub act Λ ≤ isoPart act Λ

theorem **subset_genSub** (act : Mat R S →ₗ[ℂ] Module.End ℂ V) (T : Set V) : T ⊆ genSub act T

theorem **act_mem_genSub** (act : Mat R S →ₗ[ℂ] Module.End ℂ V) (T : Set V) (Z : Mat R S) {v : V} (hv : v ∈ genSub act T) : act Z v ∈ genSub act T

theorem **genSub_le** {act : Mat R S →ₗ[ℂ] Module.End ℂ V} {T : Set V} {N : Submodule ℂ V} (hT : T ⊆ N) (hN : ∀ Z, ∀ y ∈ N, act Z y ∈ N) : genSub act T ≤ N

theorem **genSub_isoPart** (act : Mat R S →ₗ[ℂ] Module.End ℂ V) (Λ : Ix R S → ℚ) : genSub act (isoPart act Λ) = genSub act (hwSub act Λ)

theorem **map_mem_wtSub** (h : ∀ X v, b X (π v) = π (a X v)) {μ : Ix R S → ℚ} {v : V} (hv : v ∈ wtSub a μ) : π v ∈ wtSub b μ

theorem **map_mem_hwSub** (h : ∀ X v, b X (π v) = π (a X v)) {μ : Ix R S → ℚ} {v : V} (hv : v ∈ hwSub a μ) : π v ∈ hwSub b μ

theorem **genSub_hw_of_surjective** (hπ : Function.Surjective π) (h : ∀ X v, b X (π v) = π (a X v)) {Λ : Ix R S → ℚ} (hg : genSub a (hwSub a Λ) = ⊤) : genSub b (hwSub b Λ) = ⊤

theorem **exists_upq_add** (Z : Mat R S) : ∃ X ∈ upq R S, ∃ Y ∈ upq R S, Z = X + Complex.I • Y

@[simp] theorem **rT_apply** (h : UA σ J) (f : UA σ J → ℂ) (g : UA σ J) : rT h f g = f (g * h)

theorem **cmp_mul** {w : InfinitePlace F} (τ : EmbOver E w) (g h : UA σ J) : cmp τ (g * h) = cmp τ g * cmp τ h

theorem **cmp_one** {w : InfinitePlace F} (τ : EmbOver E w) : cmp τ (1 : UA σ J) = 1

theorem **incM_map** {m k : Type} (A : Matrix m k E) {w : InfinitePlace F} (τ : EmbOver E w) : (incM F A).map (piInf τ) = A.map τ.1

theorem **genBy_iff_hw** {Λ : Ix R S → ℚ} : GenBy τ w C T Λ ↔ ∃ act, IsGKAt τ w C T act ∧ genSub act (hwSub act Λ) = ⊤

theorem **HasRDer.unique** {γ : Multiplicative ℝ →* UA σ J} {f f₁ f₂ : UA σ J → ℂ} (h₁ : HasRDer γ f f₁) (h₂ : HasRDer γ f f₂) : f₁ = f₂

theorem **liftAt_map** [IsTotallyReal F] {ξ : E} (hξ : ∀ w, ((τ w).1 ξ).im ≠ 0) (G : Matrix (Fin n) (Fin n) ℂ) (w' : InfinitePlace F) : (liftAt τ w ξ G).map (piInf (τ w')) = if w' = w then G else 1

theorem **at_liftAt** [IsTotallyReal F] (ξ : E) (G : Matrix (Fin n) (Fin n) ℂ) : At (evI F) (liftAt τ w ξ G − 1)

theorem **map_cA_map** (hCM : IsCMExt F E σ) (τ : EmbOver E w) (A : Matrix (Fin n) (Fin n) (AE F E)) : (A.map (cA σ)).map (piInf τ) = (A.map (piInf τ)).map star

theorem **exists_lift** (hCM : IsCMExt F E σ) (τ : ∀ w : InfinitePlace F, EmbOver E w) (w : InfinitePlace F) (J : Matrix (Fin n) (Fin n) E) : ∃ L : Matrix (Fin n) (Fin n) ℂ → Matrix (Fin n) (Fin n) (AE F E), Continuous L ∧ L 1 = 1 ∧ (∀ G H, L (G * H) = L G * L H) ∧ (∀ G, At (evI F) (L G − 1)) ∧ (∀ G, (L G).map (piInf (τ w)) = G) ∧ (∀ G w', w' ≠ w → (L G).map (piInf (τ w')) = 1) ∧ ∀ G, Gᵀ * herm J (τ w) * G.map star = herm J (τ w) → (L G)ᵀ * incM F J * (L G).map (cA σ) = incM F J

theorem **exists_lineAt** (hCM : IsCMExt F E σ) (hC : IsFrame (herm J (τ w)) C) {X : Mat R S} (hX : X ∈ upq R S) : ∃ γ ∈ LinesAt σ J τ w, IsGenAt (τ w) C γ X

theorem **IsRightAct.unique** (hCM : IsCMExt F E σ) (hC : IsFrame (herm J (τ w)) C) {a b : Mat R S →ₗ[ℂ] Module.End ℂ T} (ha : IsRightAct τ w C T a) (hb : IsRightAct τ w C T b) : a = b

theorem **exists_isGKAt_range** (hCM : IsCMExt F E σ) (hC : IsFrame (herm J (τ w)) C) (M : GKMod R S) (φ : M.V →ₗ[ℂ] (UA σ J → ℂ)) (hφ : ∀ γ ∈ LinesAt σ J τ w, ∀ X, IsGenAt (τ w) C γ X → ∀ v, HasRDer γ (φ v) (φ (M.act X v))) : ∃ act, IsGKAt τ w C (LinearMap.range φ) act ∧ ∀ X v, act X (φ.rangeRestrict v) = φ.rangeRestrict (M.act X v)

theorem **genBy_range** (hCM : IsCMExt F E σ) (hC : IsFrame (herm J (τ w)) C) (M : GKMod R S) (φ : M.V →ₗ[ℂ] (UA σ J → ℂ)) (hφ : ∀ γ ∈ LinesAt σ J τ w, ∀ X, IsGenAt (τ w) C γ X → ∀ v, HasRDer γ (φ v) (φ (M.act X v))) {Λ : Ix R S → ℚ} (hΛ : genSub M.act (hwSub M.act Λ) = ⊤) : GenBy τ w C (LinearMap.range φ) Λ

theorem **cPair_eq_c₁** (hR : R + (p + q) + (i + j) = a + a') (hS : S + (i + j) = p + q) : cPair R S = c₁ a a' p q

theorem **cPair_eq_c₂** (hi : i ≤ p) (ha : p + i ≤ a) : cPair (a - p - i) (p - i) = c₂ a p

theorem **x₂_lt_x₁** (hκ : ((a + a' : ℕ) : ℤ) < κ) (hy : 2 * (p + q) ≤ a + a') : x₂ a p < x₁ a a' p q κ ∧ y₂ a p < y₁ a a' p q κ

theorem **single_mem_wtSub** (c : Ix R S) : Pi.single c (1 : ℂ) ∈ wtSub (Matrix.toLin' : Mat R S ≃ₗ[ℂ] _).toLinearMap (stdWt c)

theorem **stdMod_gen** : genSub (stdMod (R + 1) S).act (hwSub (stdMod (R + 1) S).act (stdWt (.inl 0))) = ⊤

theorem **coef_apply** (τ : EmbOver E w) (i : Fin n) (v : Ix R S → ℂ) (g : UA σ J) : coef τ C i v g = ((cmp τ g * C) *ᵥ v) i

theorem **hasRDer_coef** (τ : EmbOver E w) (i : Fin n) {γ : Multiplicative ℝ →* UA σ J} {X : Mat R S} (hX : IsGenAt τ C γ X) (v : Ix R S → ℂ) : HasRDer γ (coef τ C i v) (coef τ C i (X *ᵥ v))

theorem **stable_genBy_coef** (hCM : IsCMExt F E σ) {Y : Spc F E n} {C : Matrix (Fin n) (Ix (R + 1) S) ℂ} (hC : IsAdapted τ w Y C) (i : Fin n) : Stable τ w Y C (LinearMap.range (coef (σ

theorem **isAdapted_hyp** : IsAdapted τ w (hypPlane F E) hypFrame

theorem **exists_stable_hyp** (hCM : IsCMExt F E σ) : ∃ T : Submodule ℂ (UA σ (hypPlane F E).J → ℂ), Stable τ w (hypPlane F E) hypFrame T ∧ GenBy τ w hypFrame T (stdWt (.inl 0)) ∧ ∀ act, IsRightAct τ w hypFrame T act → act (Eab (.inr 0) (.inl 0)) ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.GenericPointAsIdeal

theorem **isConst_trivShift** (t : ℝ) : IsConst (trivShift t) t

theorem **isConst_triv** : IsConst (HermLB.triv _) 0

theorem **IsConst.tens** (h : IsConst L s) (h' : IsConst M t) : IsConst (L.tens M) (s + t)

theorem **IsConst.dual** (h : IsConst L s) : IsConst L.dual (-s)

theorem **IsConst.pow** (h : IsConst L s) (m : ℕ) : IsConst (L.pow m) (m * s)

theorem **IsConst.zp** (h : IsConst L s) (n : ℤ) : IsConst (zp L n) (n * s)

theorem **isConst_comb** {ι : Type} [Fintype ι] {L : ι → HermLB (𝟙 (Sp ℤ))} {t : ι → ℝ} (h : ∀ i, IsConst (L i) (t i)) (n : ι → ℤ) : IsConst (comb L n) (∑ i, n i * t i)

theorem **IsConst.isArithHeight** (h : IsConst L t) : L.IsArithHeight fun x => if x.asIdeal = ⊥ then t else Real.log (Nat.card (ℤ ⧸ x.asIdeal)) where

theorem **genericPoint_asIdeal** : (genericPoint (Sp ℤ)).asIdeal = ⊥

theorem **IsConst.isSelfInter** (h : IsConst L t) : L.IsSelfInter t

theorem **isArFun_trivShift** {ι : Type} [Fintype ι] (t : ι → ℝ) : IsArFun (fun i => trivShift (t i)) 0 (linFun t)

theorem **linFun_X** {ι : Type} [Fintype ι] (t : ι → ℝ) (i : ι) : linFun t (MvPolynomial.X i) = t i

theorem **isAlgPt_generic** : IsAlgPt (genericPoint (Sp ℤ))

theorem **arVol_nonneg** {k : Type} [CommRing k] {X : Scheme.{0}} {p : X ⟶ Sp k} (L : HermLB p) (D : ℕ) : 0 ≤ arVol L D

theorem **eq_zero_of_pow** {ι : Type} [Fintype ι] (d : ℕ) (Φ : MvPolynomial ι ℝ →ₗ[ℝ] ℝ) (h : ∀ n : ι → ℤ, Φ (lin (fun i => (n i : ℝ)) ^ d) = 0) (h' : ∀ (P : MvPolynomial ι ℝ) (e : ℕ), e ≠ d → P.IsHomogeneous e → Φ P = 0) : Φ = 0

private theorem **lin_pow_eq_eval_sum_multinomial** {ι : Type} [Fintype ι] [DecidableEq ι] (Ψ : MvPolynomial ι ℝ →ₗ[ℝ] ℝ) (m : ℕ) (r : ι → ℝ) : Ψ (_root_.GlobalTheory.ArInt.lin r ^ m) = MvPolynomial.eval r (∑ k ∈ (Finset.univ : Finset ι).piAntidiag m, MvPolynomial.monomial (Finsupp.equivFunOnFinite.symm k) ((Nat.multinomial Finset.univ k : ℝ) * Ψ (MvPolynomial.monomial (Finsupp.equivFunOnFinite.symm k) 1)))

private theorem **eq_zero_of_isHomogeneous_of_forall_lin_pow_eq_zero** {ι : Type} [Fintype ι] (Ψ : MvPolynomial ι ℝ →ₗ[ℝ] ℝ) (m : ℕ) (hΨ : ∀ n : ι → ℤ, Ψ (_root_.GlobalTheory.ArInt.lin (fun i => (n i : ℝ)) ^ m) = 0) {P : MvPolynomial ι ℝ} (hP : P.IsHomogeneous m) : Ψ P = 0

theorem **_root_.GlobalTheory.ArInt.Model.isArFun_unique** {k : Type} [CommRing k] {X : Scheme.{0}} [IsIntegral X] {p : X ⟶ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp k} {D E : ℕ} (hX : _root_.GlobalTheory.IsArithVar p E) {ι : Type} [Fintype ι] {L : ι → _root_.GlobalTheory.HermLB p} {Φ Φ' : MvPolynomial ι ℝ →ₗ[ℝ] ℝ} (h : _root_.GlobalTheory.ArInt.IsArFun L D Φ) (h' : _root_.GlobalTheory.ArInt.IsArFun L D Φ') : Φ = Φ'

theorem **isGeoFun_unique** {k : Type} [CommRing k] {X : Scheme.{0}} {p : X ⟶ Sp k} {D : ℕ} {ι : Type} [Fintype ι] {L : ι → HermLB p} {Φ Φ' : MvPolynomial ι ℝ →ₗ[ℝ] ℝ} (h : IsGeoFun L D Φ) (h' : IsGeoFun L D Φ') : Φ = Φ'

theorem **IsConst.isPtHt** (h : IsConst L t) : IsPtHt L (genericPoint (Sp ℤ)) t

theorem **IsConst.encard_smallSec** (h : IsConst L t) [Subsingleton L.B.ι] : L.smallSec.encard = {n : ℤ | |(n : ℝ)| ≤ Real.exp t}.encard

theorem **arVol_trivShift** (ht : 0 ≤ t) : arVol (trivShift t) 0 = (t : EReal)

theorem **isSelfInter_fnB** (u : ℝ) : (fnB (p

theorem **chInt_const** (L : HermLB (𝟙 (Sp ℤ))) (u : ℝ) : chInt L 0 (fun _ _ => u) = u

theorem **gdeg_eq_one** (L : HermLB (𝟙 (Sp ℤ))) : gdeg L 0 = 1

theorem **isGeoFun_int** {ι : Type} [Fintype ι] (L : ι → HermLB (𝟙 (Sp ℤ))) : IsGeoFun L 0 (lcoeff ℝ 0)

Development.AbelianVarietyTheory.Definitions.Divisors.GhostFrobenius

theorem **_root_.GlobalTheory.PDivLat.ghost_add** (n : ℕ) (a b : _root_.GlobalTheory.Sandwich.Ainf p O) : _root_.GlobalTheory.PDivLat.ghost p O n (a + b) = _root_.GlobalTheory.PDivLat.ghost p O n a + _root_.GlobalTheory.PDivLat.ghost p O n b

theorem **_root_.GlobalTheory.PDivLat.ghost_mul** (n : ℕ) (a b : _root_.GlobalTheory.Sandwich.Ainf p O) : _root_.GlobalTheory.PDivLat.ghost p O n (a * b) = _root_.GlobalTheory.PDivLat.ghost p O n a * _root_.GlobalTheory.PDivLat.ghost p O n b

theorem **_root_.GlobalTheory.PDivLat.ghost_zero** (n : ℕ) : _root_.GlobalTheory.PDivLat.ghost p O n 0 = 0

theorem **ghost_frobenius** (n : ℕ) (a : Ainf p O) : ghost p O n (WittVector.frobenius a) = ghost p O (n + 1) a

Development.AbelianVarietyTheory.Definitions.Divisors.GlRes

theorem **glH** : IsExpLieAlgebra (glG E a) (glL E a)

Development.AbelianVarietyTheory.Definitions.Divisors.GlobalHeightBoundIffAtGoodPrimes

theorem **globalHeightBound_iff_atGoodPrimes** {D : D7} (Y : ShimX D.what₂.Tau) : GlobalTheory.CovHt.DeRhamMinusHodgeTateGoodPrimesAt Y ↔ ∀ p : ℕ, p.Prime → p ≠ 2 → UnramE D p → p ∉ Y.lev → GlobAt Y p

Development.AbelianVarietyTheory.Definitions.Divisors.GrNormCompCast

theorem **grNorm_comp_cast** {X : Scheme.{0}} {t : SAB H X} {C : Type} [NontriviallyNormedField C] [IsUltrametricDist C] (w : Sp C ⟶ X) (Λ : Set (Sp (DualNumber C) ⟶ t.G.left)) {m n : ℕ} (e : m = n) (u : Fin n → (Sp (DualNumber C) ⟶ t.G.left)) (i : Fin m) : GlobalTheory.ShFam.Pad.grNorm t w Λ (fun k => u (k.cast e)) i = GlobalTheory.ShFam.Pad.grNorm t w Λ u (i.cast e)

theorem **initial_cast** {m : ℕ} (τ : E →+* ℂ) (e : m = Dw.mult τ) : Initial fun i : Fin m => ((τ, i.cast e) : Letter Dw)

theorem **stepNorms_cast** {m : ℕ} (τ : E →+* ℂ) (e : m = Dw.mult τ) (h : τ (H.rt : E) = H.rt0 → vDiff E p = 0) : StepNorms Dw p (fun i : Fin m => .inl (τ, i)) fun i => (τ, i.cast e)

theorem **pairwise_letters** : (letters Dw).Pairwise fun x y => x.1 = y.1 → (x.2 : ℕ) < y.2

theorem **xLie_initial** (jc : KW p →+* ℂ) (τ : Emb D p) : Initial (xLie S jc τ)

theorem **xOm_initial** (jc : KW p →+* ℂ) (τ : Emb D p) : Initial fun i => xOm S jc τ (Fin.rev i)

theorem **xAll_bijective** : Function.Bijective (xAll S)

theorem **xAll_initial** : Initial (xAll S)

theorem **xLie_stepNorms** (jc : KW p →+* ℂ) (τ : Emb D p) (h : vDiff D.host.E p = 0) : StepNorms S.Dw p (fun i => .inl (jc.comp τ, i)) (xLie S jc τ)

theorem **xOm_stepNorms** (jc : KW p →+* ℂ) (τ : Emb D p) (h : vDiff D.host.E p = 0) : StepNorms S.Dw p (fun i => .inl (conjEmb (jc.comp τ), (Fin.rev (Fin.rev i) : Fin _))) fun i => xOm S jc τ (Fin.rev i)

theorem **xAll_stepNorms** (p : ℕ) : StepNorms S.Dw p (fun k : Fin (D.R.f * N) => .inr k) (xAll S)

Development.AbelianVarietyTheory.Definitions.Divisors.GridTestLatticeLargestSlopeBound

theorem **deltaForm_left** (x : A) (v : M) (δ : Γ) (g : A) : deltaForm ι U ρ x v (ι δ * g) = ρ δ (deltaForm ι U ρ x v g)

theorem **deltaForm_right** (x : A) (v : M) (g : A) {u : A} (hu : u ∈ U) : deltaForm ι U ρ x v (g * u) = deltaForm ι U ρ x v g

theorem **deltaForm_self** (x : A) (v : M) (h : ∀ γ, x⁻¹ * ι γ * x ∈ U → γ = 1) : deltaForm ι U ρ x v x = v

theorem **deltaForm_eq_zero** (x : A) (v : M) (g : A) (h : ∀ γ, ∀ u ∈ U, g ≠ ι γ * x * u) : deltaForm ι U ρ x v g = 0

theorem **isV_Xp** : (G.node (.top 0)).IsV (rankP D D.R.m) 0

theorem **rank_Xp** : (Xp G).rank = rankP D D.R.m

theorem **psi_left** (rp : ClassReps G) {n : ℕ} (e : Fin n ≃ TestIx G × VVIdx (LaP G)) (a : Fin n) (γ : UR D.sp.cm (Xp G).gram) (g : GLf G) : psi rp e a (ratIn G γ * g) = rhoW G γ (psi rp e a g)

theorem **psi_right** (rp : ClassReps G) {n : ℕ} (e : Fin n ≃ TestIx G × VVIdx (LaP G)) (a : Fin n) (g : GLf G) {k : GLf G} (hk : k ∈ G.Kplus) : psi rp e a (g * k) = psi rp e a g

theorem **ClassReps.gridClassReps_delta_apply_rep** (rp : ClassReps G) (hN : rp.Neat) (c c' : G.ClPlus) (v : WtSp G) : rp.delta c v (rp.x c') = if c' = c then v else 0

theorem **testFunction_value_at_rep** (rp : ClassReps G) (hN : rp.Neat) {n : ℕ} (e : Fin n ≃ TestIx G × VVIdx (LaP G)) (a : Fin n) (c : G.ClPlus) : psi rp e a (rp.x c) = valAt (LaP G) e (datumOf e a, c) (Pi.single a 1)

theorem **posDef_Xp** (φ : D.host.E →+* ℂ) : ((Xp G).gram.map φ).PosDef

theorem **isSelfDualE_latOf** (g : UA D.sp.cm (Xp G).gram) : IsSelfDualE (D.sp.cm.sig : D.host.E →+* D.host.E) (Xp G).gram (latOf G g)

theorem **star_emb** (τ : D.what₂.Tau) (x : D.host.E) : star (emb D τ x) = emb D τ ((D.sp.cm.sig : D.host.E →+* D.host.E) x)

theorem **exists_isTestLat** (rp : ClassReps G) (n : ℕ) (e : Fin n ≃ TestIx G × VVIdx (LaP G)) : ∃ X' M, IsTestLat G rp n e X' M

theorem **IsTestLat.gridTestLattice_largestSlope_bound** (hS : SlopeE) {v₀ : InfinitePlace D.what₂.LL} (hv : Formalism.DivisionValues.IsConjAt v₀) {R : ℝ} (hR : 1 ≤ R) (hLa : SpreadLe (LaP G) R) {rp : ClassReps G} {n : ℕ} {e : Fin n ≃ TestIx G × VVIdx (LaP G)} {X' : InfinitePlace D.what₂.LL → Matrix (Fin n) (Fin n) ℂ} {M : Submodule (𝒪 D.what₂.LL) (Fin n → D.what₂.LL)} (h : IsTestLat G rp n e X' M) : Formalism.L2.Indef.Tree.ArchimedeanLeafBounds.Cuts.TestLatticeCondition n X' M (bound (Fintype.card D.what₂.Tau) (Xp G).rank R)

Development.AbelianVarietyTheory.Definitions.Divisors.HasBlockOfOccurs

theorem **eq_one_of_occurs** {L V J : Type} [Field L] [AddCommGroup V] [Module L V] {op : J → Module.End L V} {x : J → L} (hx : Formalism.L2.Col.Occurs op x) {j : J} (hj : op j = 1) : x j = 1

theorem **Reads.hasBlock_of_occurs** {w : HeightOneSpectrum (𝒪 K)} (R : Reads F lam B w) (h0 : ∀ P, B.op (P, 0) = 1) {x : B.Pl × ℕ → K} (hx : Formalism.L2.Col.Occurs B.op x) (hb : Formalism.L2.Col.HasBlocks B.n B.q B.t B.bl x) : R.HasBlock fun i => R.e0 (x (R.ix i))

theorem **sysOf_zero** {l : ℕ} [Fact l.Prime] {Pl I : Type} (n : ℕ) (pt : Pl → Fin n → I) (θ : I → Qbar l) (P : Pl) : sysOf n pt θ P 0 = 1

theorem **sysOf_succ** {l : ℕ} [Fact l.Prime] {Pl I : Type} (n : ℕ) (pt : Pl → Fin n → I) (θ : I → Qbar l) (P : Pl) (k : Fin n) : sysOf n pt θ P (k + 1) = θ (pt P k)

theorem **Reads.HasBlock.dvd** {l : ℕ} [Fact l.Prime] {ι : Type} [DecidableEq ι] {N : ℕ} {F : Frame l ι N} {lam : ι → ℕ → ℕ} {K : Type} [Field K] [NumberField K] [StarRing K] {G0 : Type} [Group G0] [Fintype G0] {Pl0 : Type} {B : BlockDat K G0 Pl0} {w : HeightOneSpectrum (𝒪 K)} {R : Reads F lam B w} {θ : F.I → PadicAlgCl l} (h : R.HasBlock θ) {Pl : Type} (pt : Pl → Fin B.n → F.I) (P : Pl) (Q : B.Pl) (hQ : ∀ k, R.ix (pt P k) = (Q, (k : ℕ) + 1)) : (B.bl.map fun ν => blockPoly (R.e0 (ν Q)) (B.q Q) B.t).prod ∣ heckePolyOf B.n (B.q Q) (sysOf B.n pt θ P)

Development.AbelianVarietyTheory.Definitions.Divisors.HasBoundedPolesGlobalData

theorem **LinSys.IsVariety.isIntegral** {k : Type} [Field k] {S : LinSys k} (h : S.IsVariety) : IsIntegral S.Y

theorem **LinSys.ofSec_comb** {k : Type} [Field k] (S : LinSys k) (O : S.Y.Opens) (u : Fin S.N → k) : H0.ofSec S.str O (S.comb u) = ∑ j, u j • H0.ofSec S.str O (S.b j)

theorem **isSmooth_weight** (hb : T.BasePointFree) (hh : T.Hermitian) {σ : F →+* ℂ} {D : ℕ} (R : AnRead T.str σ D) : (T.weight hb hh σ).IsSmooth R

theorem **weight_φ** (hb : T.BasePointFree) (hh : T.Hermitian) (σ : F →+* ℂ) (i : T.L.ι) (y : CxPts T.str σ) : (T.weight hb hh σ).φ i y = Real.log (archSize (T.A σ) (fun j => secVal T.str σ (T.b j) i y) ^ 2)

theorem **IsAlgebraicData.baseChange** (h : T.IsAlgebraicData) : (T.baseChange E).IsAlgebraicData

theorem **U_baseChange** : (T.baseChange E).U = T.bcFst E ⁻¹ᵁ T.U

theorem **IsGeometric.baseChange** (h : T.IsGeometric) : (T.baseChange E).IsGeometric

theorem **IsGeometric.nonempty_U** (h : T.IsGeometric) : (T.U : Set T.Y).Nonempty

theorem **one_le_n** : 1 ≤ G.n

theorem **isSmooth_φTw** (m : ℤ) {σ : F →+* ℂ} (R : AnRead G.str σ G.D) : (G.φTw m σ).IsSmooth R

theorem **isLB_adj** (j : ℕ) : (G.adj j).IsLB

theorem **restr_injective** (B : LB G.Y) : Function.Injective (H0.restr le_top : H0 G.str B ⊤ → H0 G.str B G.U)

theorem **finrank_global** (j : ℕ) : Module.finrank F (G.global j) = lbH0 F (G.adj j)

theorem **mulW_apply** (j : ℕ) (u : Fin G.N → F) : G.mulW j u = mulSec (isLB_tens G.K.isLB G.isLBA) G.isLB (G.comb u) (Nat.cast_succ j).symm

theorem **mulE_mem_global** (j : ℕ) (f : H0 G.str (G.adj₂ j) ⊤) : G.mulE j f ∈ G.global j

theorem **mulE_apply** (j : ℕ) (f : H0 G.str (G.adj₂ j) ⊤) : G.mulE j f = Vanishing.mulE G.K G.isLB G.isLBA G.isLB₂ G.e rfl (H0.restr le_top f)

theorem **mulE_injective** (j : ℕ) : Function.Injective (G.mulE j)

theorem **sqInt_eq** (j : ℕ) {σ : F →+* ℂ} (R : AnRead G.str σ G.D) (f : (G.adj₂ j).ι → CxPts G.str σ → ℂ) : G.sqInt j σ f = R.topInt G.K (G.φTw (j + G.D) σ) (assocVal f)

theorem **sqInt_lt_top** (j : ℕ) (σ : F →+* ℂ) {ι : Type*} [Fintype ι] (c : ι → ℂ) (f : ι → H0 G.str (G.adj₂ j) ⊤) : G.sqInt j σ (∑ i, c i • cval σ (f i)) < ⊤

theorem **IsGood.suppIn_e** : SuppIn (nonzeroLocus G.e) (G.ideal s)

theorem **IsGood.res_mulW** (j : ℕ) (u : Fin G.N → F) (m : G.Sec j) : hs.res (j + 1) (G.mulW j u m) = mulSecOn G.isLB₂ G.isLB (G.comb u) (Nat.cast_succ j).symm (hs.res j m)

theorem **IsGood.res_mulE** (j : ℕ) (f : H0 G.str (G.adj₂ j) ⊤) : hs.res j (G.mulE j f) = resDs G.K G.isLB G.isLB₂ (G.sec s) hs.etale suppIn_top rfl f

theorem **IsGood.nrm_normOn_sq** (j : ℕ) (σ : F →+* ℂ) (g : G.SecOn s j) : (hs.normOn j).nrm σ g ^ 2 = ∑ᶠ y, (weightOn s j σ).nsq (cval σ g) y

theorem **isPullback_fst** : IsPullback (G.fst E) (G.baseChange E).str G.str (spMap (algebraMap F E))

theorem **linearIndependent_toExt** (j n : ℕ) (v : Fin n → G.Sec j) (hv : LinearIndependent F v) : LinearIndependent E fun i => G.toExt E j (v i)

theorem **extSpace_global** (j : ℕ) : extSpace (G.toExt E j) (G.global j) = (G.baseChange E).global j

theorem **isGood_baseChange_iff** {s : Fin G.D → Fin G.N → E} : (G.baseChange E).IsGood s ↔ G.IsGoodSection E s

theorem **latRank_M** (j : ℕ) : latRank F (X.M j) = Module.finrank F (X.V j)

theorem **isExtension_extMap** (j : ℕ) : IsExtension (extMap (X.toExt E j) (X.V j))

theorem **nrm_toExtV** (j : ℕ) (τ : E →+* ℂ) (x : X.V j) : ((X.baseChange E).nm j).nrm τ (X.toExtV E j x) = (X.nm j).nrm (τ.comp (algebraMap F E)) x

theorem **finrank_V_baseChange** (j : ℕ) : Module.finrank E ((X.baseChange E).V j) = Module.finrank F (X.V j)

theorem **d_eq_finrank** : X.d = Module.finrank F (X.V X.T)

theorem **a₀_add** : X.a₀ + X.D + 1 = X.T

theorem **sum_Ra** : ∑ a ∈ Finset.range (X.a₀ + 1), X.Ra a = X.R

theorem **R_pos** : 0 < X.R

theorem **d_baseChange** (E : Type) [Field E] [NumberField E] [Algebra F E] : (X.baseChange E).d = X.d

theorem **z_baseChange** (E : Type) [Field E] [NumberField E] [Algebra F E] : (X.baseChange E).z = X.z

theorem **MulStable.mem** (h : X.MulStable) {j : ℕ} (hj : j < X.T) (u : Fin X.N → F) (f : X.V j) : X.mulW j u f ∈ X.V (j + 1)

theorem **MulStable.mulBound** (h : X.MulStable) {β : (F →+* ℂ) → ℝ} (hb : ∀ σ, X.MulBoundAt σ (β σ)) : X.MulBound β

theorem **meanLog_const** (c : ℝ) : meanLog F (fun _ => c) = Real.log c

theorem **ShortFamily.mean** {j : ℕ} {Λ : ℝ} (h : X.ShortFamily j Λ) : X.ShortFamilyMean j Λ

theorem **Hyp.meanHyp** (h : X.Hyp) : X.MeanHyp (fun _ => X.beta) fun _ => 1

theorem **MeanHyp.isAdelic** {β κ : (F →+* ℂ) → ℝ} (h : X.MeanHyp β κ) : X.IsAdelic

theorem **MeanHyp.two_le_theta** {β κ : (F →+* ℂ) → ℝ} (h : X.MeanHyp β κ) (hd : 0 < X.del) (t : ℝ) : 2 ≤ X.theta t

theorem **Geo.hasBoundedPoles_globalData** {F : Type} [Field F] [NumberField F] (G : Geo F) {T : ℕ} (hT : G.D + 1 ≤ T) : (G.globalData T hT).HasBoundedPoles

theorem **Geo.shortFamily_globalData** {F : Type} [Field F] [NumberField F] (G : Geo F) {T : ℕ} (hT : G.D + 1 ≤ T) (j : ℕ) : (G.globalData T hT).ShortFamily j 1

theorem **U_eq** : (tuple F M).U = nonzeroLocus (coord F M (Fin.last M))

theorem **isAlgebraicData** (hM : 1 ≤ M) : (tuple F M).IsAlgebraicData

theorem **isLB_bundleA** : (bundleA F M).IsLB

theorem **exists_e** : ∃ e : (LB.div (bundleA F M) (tw (O1 F M) (O1 F M) M)).Sec, nonzeroLocus e = nonzeroLocus (coord F M (Fin.last M))

theorem **hasNonzeroSec_pow** (M n : ℕ) : HasNonzeroSec (lbPow (O1 F M) n)

theorem **positivity** : (geo F hM).Positivity

theorem **cls_adj** {Y : Scheme.{0}} {K L : LB Y} (hK : K.IsLB) (hL : L.IsLB) (M j : ℕ) (h₁ h₂ h₃) : cls (tw (tens K (tens L (tw L L M))) L j) h₁ = cls (tens K (lbPow L (M + 1))) h₂ + cls (lbPow L (j + 1)) h₃

theorem **finrank_global_pos** (j : ℕ) : 0 < Module.finrank F ((geo F hM).global j)

theorem **d_pos** (T : ℕ) (hT : M + 1 ≤ T) : 0 < (data F hM T hT).d

Development.AbelianVarietyTheory.Definitions.Divisors.HasDivPlace

theorem **CongOne.mul** {N : ℕ} {g h : G.GAf} (hg : L.CongOne N g) (hh : L.CongOne N h) : L.CongOne N (g * h)

Development.AbelianVarietyTheory.Definitions.Divisors.HasPolChartOf

theorem **half_iff** {Q : Data R} {ρ : R →+* R'} {D : DegPre R'} {e : D.X ≃+ Q.X} {f : D.X → D.X → (FractionRing R')ˣ} : Half Q ρ D e f ↔ ∃ β, HalfAt Q ρ D e f β

Development.AbelianVarietyTheory.Definitions.Divisors.HasQIff

theorem **IsRAct.actBij** {ι : Type} {act : ι → SL(2, ℤ) → ι} (h : IsRAct act) : ActBij act

theorem **hasQ_iff** (N : ℕ) (f : ℍ → ℂ) (a : ℕ → ℂ) : HasQ N f a ↔ ∀ τ : ℍ, HasSum (fun n : ℕ => a n * Complex.exp (2 * Real.pi * Complex.I * τ / N) ^ n) (f τ)

Development.AbelianVarietyTheory.Definitions.Divisors.HasRepOfPublished

theorem **Fam.ev** {T Y : Over S} {N : LB (A ⊗ T).left} (h : Fam A N) (y : Y ⟶ T) : Fam A (ev N (fst A Y) (snd A Y ≫ y))

theorem **Inj.fibInj** {U : Over S} {M : LB (A ⊗ U).left} (h : Inj A M) : FibInj A M

Development.AbelianVarietyTheory.Definitions.Divisors.HasSumZeroOfTwoLe

theorem **coord_ne** (y : (Af E)ˣ) (v : HeightOneSpectrum (𝒪 E)) : Valued.v ((y : Af E) v) ≠ 0

theorem **modF_finite** (y : (Af E)ˣ) : (Function.mulSupport fun v : HeightOneSpectrum (𝒪 E) => ((Ideal.absNorm v.asIdeal : ℕ) : ℝ) ^ (WithZero.log (Valued.v ((y : Af E) v)))).Finite

theorem **absNorm_pos** (v : HeightOneSpectrum (𝒪 E)) : (0 : ℝ) < ((Ideal.absNorm v.asIdeal : ℕ) : ℝ)

theorem **modF_mul** (y y' : (Af E)ˣ) : modF (y * y') = modF y * modF y'

theorem **modF_pos** (y : (Af E)ˣ) : 0 < modF y

theorem **modF_princ** (k : Eˣ) : modF (Units.map (algebraMap E (Af E)).toMonoidHom k) = (|((Algebra.norm ℚ (k : E) : ℚ) : ℝ)|)⁻¹

theorem **hatO_integral** {c : Ideal (OF E)} {y : Af E} (hy : y ∈ hatO c) (v : HeightOneSpectrum (𝒪 E)) : Valued.v (y v) ≤ 1

theorem **modF_unit** {c : Ideal (OF E)} {u : (Af E)ˣ} (hu : u ∈ hatOUnits c) : modF u = 1

theorem **lattice_top_one** (x : E) : x ∈ lattice (⊤ : Ideal (OF E)) 1 ↔ ∃ b : 𝒪 E, x = algebraMap (𝒪 E) E b

theorem **prod_normSq** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) (x : E) : ∏ i, ‖φ i x‖ ^ 2 = |((Algebra.norm ℚ x : ℚ) : ℝ)|

theorem **prod_cpow** (t : Finset I) (r : I → ℝ) (hr : ∀ i, 0 ≤ r i) (z : ℂ) : ∏ i ∈ t, ((r i : ℝ) : ℂ) ^ z = ((∏ i ∈ t, r i : ℝ) : ℂ) ^ z

theorem **term_unit** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) (s : ℕ) (κ : I → ℕ) (σ : ℂ) (u : (OF E)ˣ) : term φ (2 * s) κ σ (algebraMap (OF E) E u) = 1

theorem **term_out** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) (s : ℕ) (κ : I → ℕ) (σ : ℂ) (x : E) : term φ (2 * s) κ σ (Quotient.mk (unitRel (E

theorem **term_differentiable** (φ : I → E →+* ℂ) (k : ℕ) (κ : I → ℕ) {a : E} (ha : a ≠ 0) : Differentiable ℂ fun σ => term φ k κ σ a

theorem **isLatSum_single** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) (s : ℕ) (κ : I → ℕ) {a : E} (ha : a ≠ 0) : IsLatSum φ (2 * s) κ {a} fun σ => term φ (2 * s) κ σ a

theorem **hasSum_smul** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) {s : ℕ} {κ : I → ℕ} {L L' : Set E} (a : Eˣ) (hL : ∀ x, (a : E) * x ∈ L' ↔ x ∈ L) {σ S : ℂ} (hS : HasSum (fun q : modUnits L => term φ (2 * s) κ σ q.1.out) S) : HasSum (fun q : modUnits L' => term φ (2 * s) κ σ q.1.out) (term φ (2 * s) κ σ a * S)

theorem **isLatSum_smul** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) {s : ℕ} {κ : I → ℕ} {L L' : Set E} (a : Eˣ) (hL : ∀ x, (a : E) * x ∈ L' ↔ x ∈ L) {S : ℂ → ℂ} (hS : IsLatSum φ (2 * s) κ L S) : IsLatSum φ (2 * s) κ L' fun σ => term φ (2 * s) κ σ a * S σ

theorem **toric_term_class** [IsTotallyComplex E] {φ : I → E →+* ℂ} (h : IsCMType φ) {s : ℕ} {κ : I → ℕ} {c : Ideal (OF E)} {χ : HChar E} (hχ : ∀ u ∈ hatOUnits c, finPart χ u = 1) (hi : HasInf χ φ fun i => (κ i : ℤ) + s) {y y' : (Af E)ˣ} (hy : (QuotientGroup.mk y : PicO c) = QuotientGroup.mk y') {S S' : ℂ → ℂ} (hS : IsLatSum φ (2 * s) κ (lattice c y) S) (hS' : IsLatSum φ (2 * s) κ (lattice c y') S') : ((finPart χ y : ℂ) * ((modF y ^ s : ℝ) : ℂ))⁻¹ * S 0 = ((finPart χ y' : ℂ) * ((modF y' ^ s : ℝ) : ℂ))⁻¹ * S' 0

theorem **summable_int** {φ : E →+* ℂ} (h : IsCMType fun _ : Unit => φ) (h2 : Module.finrank ℚ E = 2) {r : ℝ} (hr : r < -2) : Summable fun b : 𝒪 E => ‖φ b‖ ^ r

theorem **norm_term_unit** (φ : E →+* ℂ) (k κ : ℕ) (σ : ℂ) {x : E} (hx : x ≠ 0) : ‖term (fun _ : Unit => φ) k (fun _ => κ) σ x‖ = ‖φ x‖ ^ (-(k : ℝ) - 2 * σ.re)

theorem **summable_term** {φ : E →+* ℂ} (h : IsCMType fun _ : Unit => φ) (h2 : Module.finrank ℚ E = 2) (k κ : ℕ) {σ : ℂ} (hσ : 1 < (k : ℝ) / 2 + σ.re) : Summable fun q : modUnits (lattice (⊤ : Ideal (OF E)) 1) => term (fun _ : Unit => φ) k (fun _ => κ) σ q.1.out

theorem **isLatSum_int** [IsTotallyComplex E] {φ : E →+* ℂ} (h : IsCMType fun _ : Unit => φ) (h2 : Module.finrank ℚ E = 2) (z : (𝒪 E)ˣ) {s κ : ℕ} (hz : φ (z : 𝒪 E) ^ (2 * (s + κ)) ≠ 1) : IsLatSum (fun _ : Unit => φ) (2 * s) (fun _ => κ) (lattice (⊤ : Ideal (OF E)) 1) 0

theorem **mem_hatO_top** {y : Af E} : y ∈ hatO (⊤ : Ideal (OF E)) ↔ ∀ v, Valued.v (y v) ≤ 1

theorem **exists_princ** [IsPrincipalIdealRing (𝒪 E)] (y : (Af E)ˣ) : ∃ (a : Eˣ) (u : (Af E)ˣ), u ∈ hatOUnits (⊤ : Ideal (OF E)) ∧ y = Units.map (algebraMap E (Af E)).toMonoidHom a * 1 * u

theorem **subsingleton_picO** [IsPrincipalIdealRing (𝒪 E)] : Subsingleton (PicO (⊤ : Ideal (OF E)))

theorem **isToricSum_int** [IsTotallyComplex E] [IsPrincipalIdealRing (𝒪 E)] {φ : E →+* ℂ} (h : IsCMType fun _ : Unit => φ) (h2 : Module.finrank ℚ E = 2) (z : (𝒪 E)ˣ) {s κ : ℕ} (hz : φ (z : 𝒪 E) ^ (2 * (s + κ)) ≠ 1) (χ : HChar E) : IsToricSum (fun _ : Unit => φ) s (fun _ => κ) ⊤ χ 0

theorem **isToricSum_eis** : ∃ φ : Eis →+* ℂ, IsCMType (fun _ : Unit => φ) ∧ ∀ χ : HChar Eis, IsToricSum (fun _ : Unit => φ) 1 (fun _ => 1) ⊤ χ 0

theorem **hasSum_zero_of_two_le** {φ : I → E →+* ℂ} {s : ℕ} (hs : 2 ≤ s) {κ : I → ℕ} {L : Set E} {S : ℂ → ℂ} (h : IsLatSum φ (2 * s) κ L S) : HasSum (fun q : modUnits L => term φ (2 * s) κ 0 q.1.out) (S 0)

theorem **hasInfRatio_conj** (χ : HChar E) (φ : I → E →+* ℂ) (m : I → ℤ) : HasInfRatio χ (fun i => ComplexEmbedding.conjugate (φ i)) m ↔ HasInfRatio χ φ fun i => -m i

theorem **not_isLatSum_empty** (φ : Empty → E →+* ℂ) (k : ℕ) (κ : Empty → ℕ) {L : Set E} (hL : (modUnits L).Infinite) (S : ℂ → ℂ) : ¬ IsLatSum φ k κ L S

theorem **absNorm_rat** (p : HeightOneSpectrum (𝒪 ℚ)) : Ideal.absNorm p.asIdeal = (primesEquiv p : ℕ)

theorem **hasProd_one** {s : ℂ} (hs : 1 < s.re) : HasProd (eulerFactor (1 : HChar ℚ) s) (riemannZeta s)

theorem **not_isHeckeL_one** (Λ : ℂ → ℂ) : ¬ IsHeckeL (1 : HChar ℚ) Λ

theorem **mem_rat** (p : HeightOneSpectrum (𝒪 ℚ)) (n : ℕ) : (n : 𝒪 ℚ) ∈ p.asIdeal ↔ (primesEquiv p : ℕ) ∣ n

theorem **val_rat_one** (p : HeightOneSpectrum (𝒪 ℚ)) {n : ℕ} (h : ¬ (primesEquiv p : ℕ) ∣ n) : p.valuation ℚ (n : ℚ) = 1

theorem **val_rat_self** (p : HeightOneSpectrum (𝒪 ℚ)) : p.valuation ℚ ((primesEquiv p : ℕ) : ℚ) = WithZero.exp (-1 : ℤ)

theorem **finOrder_rat** (k : ℚˣ) (h : ∀ p : HeightOneSpectrum (𝒪 ℚ), p.valuation ℚ (k : ℚ) = 1) : IsOfFinOrder k

theorem **exists_dirichlet** (G : ℚˣ →* ℂˣ) (hG : ∀ (n m : ℕ) (hn : (n : ℚ) ≠ 0) (hm : (m : ℚ) ≠ 0), ¬ 3 ∣ n → n ≡ m [MOD 9] → G (Units.mk0 n hn) = G (Units.mk0 m hm)) : ∃ D : DirichletCharacter ℂ 9, ∀ (n : ℕ) (hn : (n : ℚ) ≠ 0), ¬ 3 ∣ n → D n = G (Units.mk0 n hn)

theorem **exists_isHeckeL** : ∃ (χ : HChar ℚ) (D : DirichletCharacter ℂ 9), IsUnitary χ ∧ D ≠ 1 ∧ (∃ w, ¬ χ.UnrAt w) ∧ IsHeckeL χ (DirichletCharacter.LFunction D)

Development.AbelianVarietyTheory.Definitions.Divisors.HasTraitsOnOfHasTraitsP

theorem **LatTo.refl** : m.LatTo m

theorem **nonempty_of_regular** (hact : IsAct X₁ act) (hz : X₁.S.R ≤ X₁.M.O z₁) (hV : IsCN V) (ev : X₁.M.Z.presheaf.stalk z₁ →+* V) (hev : IsLocalHom ev) (L : Type) [AddCommGroup L] [Subsingleton L] (hL : I → L → L) : Nonempty (GMem X₁ z₁ act V ev L (fun _ => 1) hL)

theorem **exists_gMem** (X₁ : GSet) (act : I → (X₁.ρ.P.A → X₁.ρ.P.A)) (h : IsAct X₁ act) : ∃ (z₁ : X₁.M.Z) (ev : X₁.M.Z.presheaf.stalk z₁ →+* X₁.Ω₁), Nonempty (GMem X₁ z₁ act X₁.Ω₁ ev PUnit (fun _ => 1) (fun _ x => x))

theorem **Tower.self_ext** {X₁ : GSet} {act : I → (X₁.ρ.P.A → X₁.ρ.P.A)} (h : IsAct X₁ act) {z₁ : X₁.M.Z} {V : Type} [CommRing V] (ev : X₁.M.Z.presheaf.stalk z₁ →+* V) (hev : IsLocalHom ev) : (Tower.self X₁ act h).Ext (z₁

theorem **Tower.d_pos** {X₁ : GSet} {act : I → (X₁.ρ.P.A → X₁.ρ.P.A)} (T : Tower X₁ act) : 0 < T.d

theorem **isAct_ofSetup** {X₁ : Setup F W i} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A → X₁.ρ.P.A)} (h : GlobalTheory.Poinc.Sep.IsAct X₁ act) (k : I → Matrix (Fin 8) (Fin 8) ℤ) : IsAct (ofSetup X₁) (fun a => act (k a))

theorem **exists_gMem_ofSetup** (X₁ : Setup F W i) : ∃ act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A → X₁.ρ.P.A), GlobalTheory.Poinc.Sep.IsAct X₁ act ∧ (∀ (I : Type) (k : I → Matrix (Fin 8) (Fin 8) ℤ), IsAct (ofSetup X₁) (fun a => act (k a))) ∧ ∃ (z₁ : X₁.M.Z) (ev : X₁.M.Z.presheaf.stalk z₁ →+* X₁.Ω₁), Nonempty (GMem (I

theorem **ofTower_ext** {X₁ : Setup F W i} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A → X₁.ρ.P.A)} (k : I → Matrix (Fin 8) (Fin 8) ℤ) (𝔗 : GlobalTheory.Poinc.Sep.Tower X₁ act) {z₁ : X₁.M.Z} {z₂ : 𝔗.X₂.M.Z} {V V' : Type} [CommRing V] [CommRing V'] (ev : X₁.M.Z.presheaf.stalk z₁ →+* V) (io : V →+* V') (ev₂ : 𝔗.X₂.M.Z.presheaf.stalk z₂ →+* V') : (ofTower k 𝔗).Ext (z₁

theorem **deep_ofSetup** {X₁ : Setup F W i} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A → X₁.ρ.P.A)} (k : I → Matrix (Fin 8) (Fin 8) ℤ) (h : ∀ N₀ : ℕ, ∃ 𝔗 : GlobalTheory.Poinc.Sep.Tower X₁ act, N₀ ≤ 𝔗.X₂.S.r * 𝔗.X₂.S.N) : Deep (ofSetup X₁) (fun a => act (k a))

theorem **ULink.uOver** {C : UCompact Xo c} {X₁ : GSet} {act : 𝒪 E → (X₁.ρ.P.A → X₁.ρ.P.A)} {u : Sp (ratHull X₁.S.R) → Xo.XF c} {π : X₁.M.Z → C.Z} (h : ULink C X₁ act u π) : UOver Xo c X₁.ρ.P (toRat X₁.S.R) u

theorem **hasTraitsOn_of_hasTraitsP** (h : GlobalTheory.USep.HasTraitsP ζ Φ) (U : M.Opens) : HasTraitsOn ζ Φ U

theorem **hasTraitsOn_top** : HasTraitsOn ζ Φ ⊤ ↔ GlobalTheory.USep.HasTraitsP ζ Φ

theorem **hasTraitsOn_of_hasTraits** {N₁ : ℕ} {Ψ : VSys (H.bar : 𝒪 E → 𝒪 E) (min Dw.p Dw.q) !![(1 : ℤ)] N₁ (M.presheaf.stalk z₁)} (h : GlobalTheory.USep.HasTraits ζ Ψ) (U : M.Opens) : HasTraitsOn ζ Ψ.pre U

Development.AbelianVarietyTheory.Definitions.Divisors.HeightTrivialParameterEqZero

theorem **exists_bridge** (τ : E →+* Qbar l) : ∃ (w : HeightOneSpectrum (𝒪 E)) (A : ValuationSubring Ω) (_ : IsPlaceOver A w) (LL : FinExt l) (Bw : LocBridge l E Ω (Dec (F

theorem **jumps_unique** {n : ℕ} {r : G E Ω →* GL (Fin n) (Qbar l)} {J J' : (E →+* Qbar l) → Multiset ℤ} (h : CrysWith r J) (h' : CrysWith r J') : J = J'

theorem **not_crysWith_cyc** : ¬ CrysWith (rk1 (cyc E Ω l ^ 1)) fun _ => {1}

theorem **grp_rank_one** {R : Type} [CommRing R] (mu : Unit → ℕ → ℕ) (g : Unit → GL (Fin 1) R) (v : NablaT R 1 mu) : grp R 1 mu g v = ((g ())⁻¹.val 0 0) ^ (mu () 0) • v

theorem **exists_ne_zero_rank_one** (K : Type) [Field K] (mu : Unit → ℕ → ℕ) : ∃ v : NablaT K 1 mu, v ≠ 0

theorem **height_trivialParameter_eq_zero** : Formalism.CohomologicalDescent.ShortWeightBounds.h0 parOne = 0

theorem **weight_unique** (hC : GlobalTheory.Published.ChebotarevDegreeOne) {inc : Unit → (E →+* Qbar l)} {W : Wts Unit} {F : Frame l Unit parOne.Nr} {Bk : (F.I → Qbar l) → Prop} (h : GalSide Ω inc parOne W F Bk) {mu mu' : Unit → ℕ → ℕ} {θ : F.I → Qbar l} (h1 : F.IsEig mu θ) (h2 : F.IsEig mu' θ) : mu () 0 = mu' () 0

theorem **not_galSide** (hC : GlobalTheory.Published.ChebotarevDegreeOne) (inc : Unit → (E →+* Qbar l)) (W : Wts Unit) (Bk : (Unit → Qbar l) → Prop) : (Inhab.twoFrame l 1).Ok ∧ ¬ GalSide Ω inc parOne W (Inhab.twoFrame l 1) Bk

theorem **invFrame_isEig** : (invFrame l).IsEig (fun _ _ => 0) fun _ => 1

theorem **eq_of_invFrame_isEig** {mu : Unit → ℕ → ℕ} {θ : Unit → PadicAlgCl l} (h : (invFrame l).IsEig mu θ) : mu () 0 = 0 ∧ θ = fun _ => 1

theorem **galSide_invFrame** (inc : Unit → (E →+* Qbar l)) (W : Wts Unit) (hj : W.j () = 0) : GalSide Ω inc parOne W (invFrame l) fun _ => True

theorem **linked_three** : Linked 3 2 (fun (_ : Unit) a => if a = 0 then 3 else 0) (fun _ a => if a = 0 then 2 else 1)

theorem **not_linked_three** : ¬ Linked 3 2 (fun (_ : Unit) a => if a = 0 then 2 else 0) (fun _ _ => 1)

theorem **isFrobPerm_rat** (l : ℕ) [Fact l.Prime] : IsFrobPerm (E

Development.AbelianVarietyTheory.Definitions.Divisors.HilbertNinetyQuotientHomEqIff

theorem **SB_eq** (X : Type) [TopologicalSpace X] : SB X = GlobalTheory.SplitWeil.SB X

theorem **IsUnitaryOn.comp** [TopologicalSpace R] {C : TestClass R} {om : G →* Module.End ℂ ((κ → R) → ℂ)} (h : IsUnitaryOn C om) {H : Type} [Group H] (f : H →* G) : IsUnitaryOn C (om.comp f)

theorem **IsContOn.comp** [TopologicalSpace G] {C : TestClass R} {om : G →* Module.End ℂ ((κ → R) → ℂ)} (h : IsContOn C om) {H : Type} [Group H] [TopologicalSpace H] {f : H →* G} (hf : Continuous f) : IsContOn C (om.comp f)

theorem **IsContOn.of_isLocallyConstant** [TopologicalSpace G] {C : TestClass R} {om : G →* Module.End ℂ ((κ → R) → ℂ)} (h : ∀ φ ∈ C.sp κ, IsLocallyConstant fun g => om g φ) : IsContOn C om

theorem **norm_eq_one_of_lintegral_eq** {X : Type} [MeasurableSpace X] {μ : Measure X} {f : X → ℂ} {a : ℂ} (h : ∫⁻ x, ‖(a • f) x‖ₑ ^ 2 ∂μ = ∫⁻ x, ‖f x‖ₑ ^ 2 ∂μ) (h0 : ∫⁻ x, ‖f x‖ₑ ^ 2 ∂μ ≠ 0) (h1 : ∫⁻ x, ‖f x‖ₑ ^ 2 ∂μ ≠ ⊤) : ‖a‖ = 1

theorem **IsUnitaryOn.norm_eq_one** [TopologicalSpace R] {C : TestClass R} {om om' : G →* Module.End ℂ ((κ → R) → ℂ)} (h : IsUnitaryOn C om) (h' : IsUnitaryOn C om') {μ : @Measure (κ → R) (borel _)} (hμ : @Measure.IsAddHaarMeasure _ _ _ (borel _) μ) {φ : (κ → R) → ℂ} (hφ : φ ∈ C.sp κ) (h0 : ∫⁻ x, ‖φ x‖ₑ ^ 2 ∂μ ≠ 0) (h1 : ∫⁻ x, ‖φ x‖ₑ ^ 2 ∂μ ≠ ⊤) {g : G} {a : ℂ} (ha : om' g φ = a • om g φ) : ‖a‖ = 1

theorem **mem_normOne** {u : Eˣ} : u ∈ normOne c ↔ (u : E) ∗ c u = 1

theorem **hilbertNinetyQuotientHom_eq_iff** {z u : Eˣ} : h90 c z = u ↔ (u : E) ∗ c z = z

theorem **tilde_norm_quotient** (χ : Eˣ →* Circle) (hχ : ∀ z : Eˣ, c z = z → χ z = 1) (z : Eˣ) : tilde c χ hχ (h90 c z, z, rfl) = χ z

theorem **isTilde_iff** (χ : Eˣ →* Circle) (hχ : ∀ z : Eˣ, c z = z → χ z = 1) (u : Eˣ) (t : Circle) : IsTilde c χ u t ↔ ∃ hu : u ∈ (h90 c).range, t = tilde c χ hχ (u, hu)

theorem **IsTilde.unique** {χ : Eˣ →* Circle} (hχ : ∀ z : Eˣ, c z = z → χ z = 1) {u : Eˣ} {t t' : Circle} (h : IsTilde c χ u t) (h' : IsTilde c χ u t') : t = t'

theorem **exists_eq** (z : E) : ∃ a b : R, z = algebraMap R E a + algebraMap R E b ∗ ξ

theorem **algebraMap_injective** : Function.Injective (algebraMap R E)

theorem **invol** (z : E) : c (c z) = z

theorem **range_h90_le_normOne** : (h90 c).range ≤ normOne c

theorem **exists_of_fixed** {z : E} (hz : c z = z) : ∃ a : R, z = algebraMap R E a

theorem **exists_unit_of_fixed** {z : Eˣ} (hz : c z = z) : ∃ a : Rˣ, z = Units.map (algebraMap R E : R →* E) a

theorem **triv_fixed** {χ : Eˣ →* Circle} (hχ : ∀ a : Rˣ, χ (Units.map (algebraMap R E : R →* E) a) = 1) (z : Eˣ) (hz : c z = z) : χ z = 1

theorem **algebraMap_norm** (z : E) : algebraMap R E (Algebra.norm R z) = z ∗ c z

theorem **IsQuad.h90_range_eq_normOne_of_isQuad** {R : Type} [Field R] [Invertible (2 : R)] {E : Type} [CommRing E] [Algebra R E] {c : E →ₐ[R] E} {ξ : E} (h : IsQuad c ξ) : (h90 c).range = normOne c

theorem **det_mem_normOne** {ι : Type} [Fintype ι] [DecidableEq ι] {J : Matrix ι ι E} (hJ : IsHerm c J) (g : UG c J) : Matrix.GeneralLinearGroup.det (g : GL ι E) ∈ normOne c

theorem **isLagr_diagCols** (JB : Matrix β β E) : IsLagr c (dbl JB) (diagCols β : Matrix (β ⊕ β) β E)

theorem **ResEtaPow.map_conj** {ξ : E} (h : IsQuad c ξ) {K : Subgroup Rˣ} {χ : Eˣ →* Circle} {n : ℕ} (hχ : ResEtaPow K χ n) (z : Eˣ) : χ (Units.map (c : E →* E) z) = (χ z)⁻¹

theorem **ResEtaPow.triv_of_even** {K : Subgroup Rˣ} {χ : Eˣ →* Circle} {n : ℕ} (hχ : ResEtaPow K χ n) (hn : Even n) (a : Rˣ) : χ (Units.map (algebraMap R E : R →* E) a) = 1

theorem **IsUWeilB.isWeilB** {χ : Eˣ →* Circle} {om : UG c JB →* Module.End ℂ ((κ → R) → ℂ)} (h : IsUWeilB c C ψ ξ JA JB d χ om) : IsWeilB c C ψ ξ JA JB d χ om

theorem **IsUWeilPair.isWeilPair** (h : IsUWeilPair c C ψ ξ JA JB d χA χB om) : IsWeilPair c C ψ ξ JA JB d χA χB om

theorem **IsUWeilPair.isUSplitting** (h : IsUWeilPair c C ψ ξ JA JB d χA χB om) : IsUSplitting C ψ d (actAB c) om

Development.AbelianVarietyTheory.Definitions.Divisors.HodgeTateLatticePointFinrankTateModule

theorem **latNorm_smul** (La : Submodule O U) {c : C} (hc : c ≠ 0) (x : U) : latNorm v La (c • x) = v c ∗ latNorm v La x

theorem **exists_grNorm_eq_mul** {f f' : Fin N → U} (h : ∀ i, step C f' i = step C f i) (k : Fin N) (hk : f' k ∉ step C f' k) : ∃ c : C, c ≠ 0 ∧ ∀ La : Submodule O U, grNorm v La f' k = v c ∗ grNorm v La f k

theorem **phiAt_vp_sub_eq** (p : ℕ) (al : U →ₗ[C] U') {U₁ : Submodule C U} {j : ℕ} (F F' : AdaptedFrame (LinearMap.ker al ⊓ U₁) U₁ N j) (h : ∀ i, step C F'.f i = step C F.f i) (LW LW' : Submodule O U) (Lom Lom' : Submodule O U') (lam : Fin N → ℝ) (hW : ∀ k : Fin N, (k : ℕ) < j → grNorm v LW F.f k ≠ 0 ∧ grNorm v LW' F.f k ≠ 0) (hom : ∀ k : Fin N, j ≤ k → grNorm v Lom (al ∘ F.f) k ≠ 0 ∧ grNorm v Lom' (al ∘ F.f) k ≠ 0) : phiAt_vp p v al LW Lom F'.f j lam − phiAt_vp p v al LW' Lom' F'.f j lam = phiAt_vp p v al LW Lom F.f j lam − phiAt_vp p v al LW' Lom' F.f j lam

theorem **card_torsion** (n : ℕ) : Nat.card {x : Pts P.A.A // x ^ p ^ n = 1} = p ^ (2 ∗ P.A.g ∗ n)

theorem **nonempty_T_equiv** : Nonempty (P.T ≃ₗ[ℤ_[p]] (Fin (2 ∗ P.A.g) → ℤ_[p]))

theorem **hodgeTateLatticePoint_finrank_tateModule** : Module.finrank ℤ_[p] P.T = 2 ∗ P.A.g

theorem **pairT_bijective** : Function.Bijective (pairT P.A P.eps)

theorem **pairV_tmul** (c d : C) (s t : P.T) : P.pairV (c ⊗ₜ s) (d ⊗ₜ t) = pairT P.A P.eps s t • (c ∗ d)

theorem **pairV_act** (a : Λ) (x y : P.V) : P.pairV (P.actV a x) y = P.pairV x (P.actV (star a) y)

theorem **pairV_swap** (x y : P.V) : P.pairV x y = −P.pairV y x

theorem **isAlt_pairV** : P.pairV.IsAlt

theorem **finrank_V** : Module.finrank C P.V = 2 ∗ P.A.g

theorem **exists_dual_bases** : ∃ (n : ℕ) (e s : Module.Basis (Fin n) ℤ_[p] P.T), (∀ i t, pairT P.A P.eps (s i) t = e.repr t i) ∧ ∀ (w : P.V) (j : Fin n), (s.baseChange C).repr w j = P.pairV w (1 ⊗ₜ e j)

theorem **existsUnique_pairV** (l : P.T →ₗ[ℤ_[p]] C) : ∃! w : P.V, ∀ t, P.pairV w (1 ⊗ₜ t) = l t

theorem **eq_of_pairV** {w w' : P.V} (h : ∀ t : P.T, P.pairV w (1 ⊗ₜ t) = P.pairV w' (1 ⊗ₜ t)) : w = w'

theorem **inclT_tmul** (c : O) (t : P.T) : P.inclT (c ⊗ₜ t) = algebraMap O C c ⊗ₜ t

theorem **one_tmul_mem_latT** (t : P.T) : (1 : C) ⊗ₜ t ∈ P.latT

theorem **latTSub_le_latTQuot** (i : J) : P.latTSub (σ i) ≤ P.latTQuot h i

theorem **fg_latT** : P.latT.FG

theorem **fg_latTQuot** (i : J) : (P.latTQuot h i).FG

theorem **mem_latT_iff** (w : P.V) : w ∈ P.latT ↔ ∀ t : P.T, ∃ c : O, algebraMap O C c = P.pairV w (1 ⊗ₜ t)

theorem **alC_tmul** (c : C) (t : P.T) : P.alC (c ⊗ₜ t) = c ⊗ₜ P.alₒ t

theorem **alC_inclT** (x : O ⊗[ℤ_[p]] P.T) : P.alC (P.inclT x) = P.inclOm (P.al x)

theorem **isEquivariant_alC** : IsEquivariant P.actV P.actOm P.alC

theorem **omHTAll_eq** : P.omHTAll = (LinearMap.range P.al).map P.inclOm

theorem **omHTAll_le_latOm** : P.omHTAll ≤ P.latOm

theorem **omHT_eq** (i : J) : P.omHT h i = eigQuotLat P.actOm h i P.omHTAll

theorem **omHT_le_omRec** (i : J) : P.omHT h i ≤ P.omRec h i

theorem **fg_omHT** (i : J) : (P.omHT h i).FG

theorem **fg_wHT** (i : J) : (P.wHT h i).FG

theorem **iSup_eigW** [Fintype J] : ⨆ i, P.eigW (σ i) = P.W

theorem **pairV_bLie** (x : LieG P.Gt) (t : P.T) : P.pairV (P.bLie x) (1 ⊗ₜ t) = algebraMap O C ((P.alₒ t).1 x)

theorem **bLie_add** (x y : LieG P.Gt) : P.bLie (lieAdd x y) = P.bLie x + P.bLie y

theorem **bLie_zero** : P.bLie (lieZero P.Gt) = 0

theorem **bLie_scale** (c : O) (x : LieG P.Gt) : P.bLie (AvDefs.lieScale P.Gt c x) = c • P.bLie x

theorem **bLie_act** (a : Λ) (x : LieG P.Gt) : P.bLie (AvDefs.lieAct P.Gt P.oGt a x) = P.actV a (P.bLie x)

theorem **bLie_mem_latT** (x : LieG P.Gt) : P.bLie x ∈ P.latT

theorem **latLie_le_latT** : P.latLie ≤ P.latT

theorem **wRec_le** (τ : Λ →+* O) : P.wRec τ ≤ eigSubLat P.actV ((algebraMap O C).comp τ) P.latLie

theorem **wRec_le_latTSub** (τ : Λ →+* O) : P.wRec τ ≤ P.latTSub τ

theorem **nonempty_frame** (τ : Λ →+* O) : Nonempty (P.Frame τ (Module.finrank C (P.eigV τ)) (Module.finrank C (P.eigW τ)))

theorem **exists_smul_mem_latT** (x : P.V) : ∃ a : O, a ≠ 0 ∧ a • x ∈ P.latT

theorem **exists_smul_mem_latTQuot** (i : J) (F : P.Frame (σ i) N j) (k : Fin N) : ∃ a : C, a ≠ 0 ∧ a • F.f k ∈ P.latTQuot h i

theorem **grAt_wHT** (i : J) (F : P.Frame (σ i) N j) (k : Fin N) (hk : (k : ℕ) < j) : ∃ c : C, c ≠ 0 ∧ grAt C (P.wHT h i) F.f k = Submodule.span O {c • (step C F.f k).mkQ (F.f k)} ∧ grNorm v (P.wHT h i) F.f k = (v c)⁻¹

theorem **grAt_omHT** (i : J) (F : P.Frame (σ i) N j) (k : Fin N) (hk : j ≤ k) : ∃ c : C, c ≠ 0 ∧ grAt C (P.omHT h i) (P.alC ∘ F.f) k = Submodule.span O {c • (step C (P.alC ∘ F.f) k).mkQ ((P.alC ∘ F.f) k)} ∧ grNorm v (P.omHT h i) (P.alC ∘ F.f) k = (v c)⁻¹

theorem **exists_oAct_polPull** {S S' : Scheme.{0}} (f : S' ⟶ S) (B : Out.PolAV S) (o : B.OAct Λ) : ∃ o' : (StrictPull.polPull B f).OAct Λ, ∀ a, (o'.act a).left ≫ pullback.fst B.A.hom f = pullback.fst B.A.hom f ≫ (o.act a).left

theorem **exists_point_dim_one** : ∃ (_ : Fact ¬IsUnit ((2 : ℕ) : 𝒪_ℂ_[2])) (_ : IsAdicComplete (Ideal.span {((2 : ℕ) : 𝒪_ℂ_[2])}) 𝒪_ℂ_[2]) (a₁ : Algebra ℤ_[2] 𝒪_ℂ_[2]) (a₂ : Algebra ℤ_[2] ℂ_[2]) (_ : @IsScalarTower ℤ_[2] 𝒪_ℂ_[2] ℂ_[2] a₁.toSMul Algebra.toSMul a₂.toSMul) (P : Point 2 (Valued.v : Valuation ℂ_[2] ℝ≥0) 𝒪_ℂ_[2] ℤ), P.A.g = 1 ∧ Module.finrank ℂ_[2] P.V = 2 ∧ IsSplitting (embC ℂ_[2] fun _ : Unit => Int.castRingHom 𝒪_ℂ_[2])

Development.AbelianVarietyTheory.Definitions.Divisors.HolCuspLevDeltaNeBot

theorem **dcFn_left** (U : Subgroup G) (x : G) {u : G} (hu : u ∈ U) (g : G) : dcFn U x (u * g) = dcFn U x g

theorem **dcFn_right** (U : Subgroup G) (x : G) {u : G} (hu : u ∈ U) (g : G) : dcFn U x (g * u) = dcFn U x g

theorem **dcFn_self** (U : Subgroup G) (x : G) : dcFn U x x = 1

theorem **dcFn_fin** [TopologicalSpace G] [IsTopologicalGroup G] {U : Subgroup G} (ho : IsOpen (U : Set G)) (hc : IsCompact (U : Set G)) (x : G) : {q : G ⧸ U | dcFn U x q.out ≠ 0}.Finite

theorem **apply_out** (g : G) : f.toFun ((g : G ⧸ U).out) = f.toFun g

theorem **mem_supp** {q : G ⧸ U} : q ∈ f.supp ↔ f.toFun q.out ≠ 0

theorem **mk_mem_supp** {g : G} : (g : G ⧸ U) ∈ f.supp ↔ f.toFun g ≠ 0

theorem **heckeOp_apply** (ρ : Representation ℂ G M) (f : HeckeFn U) (m : M) : heckeOp ρ f m = ∑ q ∈ f.supp, f.toFun q.out • ρ q.out m

theorem **heckeOp_term** (ρ : Representation ℂ G M) (f : HeckeFn U) {m : M} (hm : m ∈ Hi.fixd ρ U) (g : G) : f.toFun (g : G ⧸ U).out • ρ (g : G ⧸ U).out m = f.toFun g • ρ g m

theorem **heckeOp_eq_sum** (ρ : Representation ℂ G M) (f : HeckeFn U) {m : M} (hm : m ∈ Hi.fixd ρ U) {ι : Type} (s : Finset ι) (x : ι → G) (hinj : Set.InjOn (fun i => (x i : G ⧸ U)) s) (hcov : ∀ g, f.toFun g ≠ 0 → ∃ i ∈ s, (x i : G ⧸ U) = g) : heckeOp ρ f m = ∑ i ∈ s, f.toFun (x i) • ρ (x i) m

theorem **heckeOp_mem_fixd** (ρ : Representation ℂ G M) (f : HeckeFn U) {m : M} (hm : m ∈ Hi.fixd ρ U) : heckeOp ρ f m ∈ Hi.fixd ρ U

theorem **heckeOp_mem_of_stable** (ρ : Representation ℂ G M) (f : HeckeFn U) {W : Submodule ℂ M} (hW : ∀ g, ∀ m ∈ W, ρ g m ∈ W) {m : M} (hm : m ∈ W) : heckeOp ρ f m ∈ W

theorem **map_heckeOp** {N : Type} [AddCommGroup N] [Module ℂ N] (ρ : Representation ℂ G M) (σ : Representation ℂ G N) (f : HeckeFn U) (T : M →ₗ[ℂ] N) (hT : ∀ g m, T (ρ g m) = σ g (T m)) (m : M) : T (heckeOp ρ f m) = heckeOp σ f (T m)

theorem **heckeOp_one** (ρ : Representation ℂ G M) {m : M} (hm : m ∈ Hi.fixd ρ U) : heckeOp ρ (HeckeFn.one U) m = m

private theorem **qdet_two'** {A : Type} [Ring A] [Algebra ℂ A] (E : Fin 2 → Fin 2 → A) (u : ℂ) : Formalism.REAL.qdet E u = (algebraMap ℂ A u + E 0 0) * (algebraMap ℂ A (u - 1) + E 1 1) - E 1 0 * E 0 1

private theorem **casimir'** {V : Type} [AddCommGroup V] [Module ℂ V] (E : Fin 2 → Fin 2 → Module.End ℂ V) (hE : Formalism.REAL.GlRel E) (k c : ℂ) (φ : V) (h0 : E 0 0 φ + E 1 1 φ = c • φ) (h1 : E 0 1 φ - E 1 0 φ = (Complex.I * k) • φ) (h2 : E 0 0 φ - E 1 1 φ = Complex.I • (E 0 1 φ + E 1 0 φ)) (u : ℂ) : Formalism.REAL.qdet E (u + 1 / 2) φ = ((u + (c + k - 1) / 2) * (u + (c - k + 1) / 2)) • φ

theorem **two_le_kw** (lam : InfinitePlace F → DomWeight 2) (v : InfinitePlace F) : 2 ≤ kw lam v

theorem **three_le_kw** (lam : InfinitePlace F → DomWeight 2) (v : InfinitePlace F) : 3 ≤ kw lam v ↔ (lam v).w 1 < (lam v).w 0

@[simp] theorem **kn_cast** (lam : InfinitePlace F → DomWeight 2) (v : InfinitePlace F) : (kn lam v : ℤ) = kw lam v

theorem **infChar_prod** (lam : InfinitePlace F → DomWeight 2) (v : InfinitePlace F) (u : ℂ) : (((GlobalTheory.UnitaryBC.infOfWeight (lam v)).map Neg.neg).map (u + ·)).prod = (u + ((cw lam v : ℂ) + kw lam v - 1) / 2) * (u + ((cw lam v : ℂ) - kw lam v + 1) / 2)

theorem **isHolo_iff** (lam : InfinitePlace F → DomWeight 2) (φ : TR.Sm F 2) : IsHolo lam φ ↔ TR.IsCuspForm φ ∧ ∀ v, TR.Eop v 0 1 φ - TR.Eop v 1 0 φ = (Complex.I * (kw lam v : ℂ)) • φ ∧ TR.Eop v 0 0 φ - TR.Eop v 1 1 φ = Complex.I • (TR.Eop v 0 1 φ + TR.Eop v 1 0 φ) ∧ TR.Eop v 0 0 φ + TR.Eop v 1 1 φ = (cw lam v : ℂ) • φ

theorem **Eop_finRep** (v : InfinitePlace F) (i j : Fin 2) (g : GF F) (φ : TR.Sm F 2) : TR.Eop v i j (TR.finRep F 2 g φ) = TR.finRep F 2 g (TR.Eop v i j φ)

@[simp] theorem **mem_holCusp** {lam : InfinitePlace F → DomWeight 2} {φ : TR.Sm F 2} : φ ∈ holCusp lam ↔ IsHolo lam φ

theorem **finRep_mem_holCusp** (lam : InfinitePlace F → DomWeight 2) (g : GF F) (φ : TR.Sm F 2) (h : φ ∈ holCusp lam) : TR.finRep F 2 g φ ∈ holCusp lam

theorem **mem_holCuspLev** {lam : InfinitePlace F → DomWeight 2} {U : Subgroup (GF F)} {φ : TR.Sm F 2} : φ ∈ holCuspLev lam U ↔ IsHolo lam φ ∧ ∀ u ∈ U, TR.finRep F 2 u φ = φ

@[simp] theorem **heckeHol_val** (lam : InfinitePlace F → DomWeight 2) {U : Subgroup (GF F)} (f : HeckeFn U) (φ : holCuspLev lam U) : (heckeHol lam f φ).1 = heckeOp (TR.finRep F 2) f φ.1

theorem **heckeHol_one** (lam : InfinitePlace F → DomWeight 2) (U : Subgroup (GF F)) : heckeHol lam (HeckeFn.one U) = 1

theorem **holoOcc_iff** (lam : InfinitePlace F → DomWeight 2) (τ : SmIrr (GF F)) : HoloOcc lam τ ↔ ∃ f : τ.V →ₗ[ℂ] holCusp lam, f ≠ 0 ∧ ∀ g x, f (τ.ρ g x) = holRep lam g (f x)

theorem **holCuspLev_delta_ne_bot** : holCuspLev lamDelta (Nf.K1G 1) ≠ ⊥

theorem **rtr_mem_equivD** (g : GF F) (Φ : GF F → V) (h : Φ ∈ equivD Q ρ) : Hi.rtr ℂ (GF F) V g Φ ∈ equivD Q ρ

theorem **mem_formsD** {U : Subgroup (GF F)} {Φ : GF F → V} : Φ ∈ formsD Q ρ U ↔ (∀ γ g, Φ (Q.toFin γ * g) = ρ (Q.toInf γ) (Φ g)) ∧ ∀ u ∈ U, ∀ g, Φ (g * u) = Φ g

theorem **heckeD_apply** {U : Subgroup (GF F)} (f : HeckeFn U) (Φ : formsD Q ρ U) (g : GF F) : (heckeD Q ρ f Φ).1 g = ∑ q ∈ f.supp, f.toFun q.out • Φ.1 (g * q.out)

theorem **heckeD_one** (U : Subgroup (GF F)) : heckeD Q ρ (HeckeFn.one U) = 1

theorem **occurs_iff** (τ : SmIrr (GF F)) : Q.Occurs ρ τ ↔ ∃ f : τ.V →ₗ[ℂ] equivD Q ρ, f ≠ 0 ∧ ∀ g x, f (τ.ρ g x) = repD Q ρ g (f x)

theorem **exists_mem_formsD** (U : Subgroup (GF F)) (x : V) (hx : ∀ γ, Q.toFin γ ∈ U → ρ (Q.toInf γ) x = x) : ∃ Φ ∈ formsD Q ρ U, Φ 1 = x

theorem **normSq_coefDen** (g : Matrix (Fin 2) (Fin 2) ℝ) : Complex.normSq (coefDen g) = g 0 0 ^ 2 + g 0 1 ^ 2 + g 1 0 ^ 2 + g 1 1 ^ 2 + 2 * g.det

theorem **coefDen_ne_zero** {g : Matrix (Fin 2) (Fin 2) ℝ} (h : 0 < g.det) : coefDen g ≠ 0

theorem **lowCoef_of_pos** (k : ℕ) (s : ℂ) {g : Matrix (Fin 2) (Fin 2) ℝ} (h : 0 < g.det) : lowCoef k s g = (g.det : ℂ) ^ (((k : ℂ) + s) / 2) * (2 * Complex.I) ^ k / coefDen g ^ k

theorem **lowCoef_of_nonpos** (k : ℕ) (s : ℂ) {g : Matrix (Fin 2) (Fin 2) ℝ} (h : g.det ≤ 0) : lowCoef k s g = 0

theorem **lowCoef_zero** (k : ℕ) {g : Matrix (Fin 2) (Fin 2) ℝ} (h : 0 < g.det) : lowCoef k 0 g = ((g.det ^ ((k : ℝ) / 2) : ℝ) : ℂ) * (2 * Complex.I) ^ k / coefDen g ^ k

theorem **lowCoef_eq_mul** (k : ℕ) (s : ℂ) (g : Matrix (Fin 2) (Fin 2) ℝ) : lowCoef k s g = (g.det : ℂ) ^ (s / 2) * lowCoef k 0 g

theorem **lowCoef_one** (k : ℕ) (s : ℂ) : lowCoef k s 1 = 1

theorem **lowCoef_ne_zero** (k : ℕ) (s : ℂ) {g : Matrix (Fin 2) (Fin 2) ℝ} (h : 0 < g.det) : lowCoef k s g ≠ 0

theorem **lowCoef_smul** (k : ℕ) (s : ℂ) {t : ℝ} (ht : 0 < t) (g : Matrix (Fin 2) (Fin 2) ℝ) : lowCoef k s (t • g) = (t : ℂ) ^ s * lowCoef k s g

theorem **lowCoef_neg** (k : ℕ) (s : ℂ) (g : Matrix (Fin 2) (Fin 2) ℝ) : lowCoef k s (-g) = (-1) ^ k * lowCoef k s g

theorem **det_rotM** (θ : ℝ) : (rotM θ).det = 1

theorem **coefDen_rot** (θ φ : ℝ) (g : Matrix (Fin 2) (Fin 2) ℝ) : coefDen (rotM θ * g * rotM φ) = Complex.exp (θ * Complex.I) * coefDen g * Complex.exp (φ * Complex.I)

theorem **lowCoef_rot** (k : ℕ) (s : ℂ) (θ φ : ℝ) (g : Matrix (Fin 2) (Fin 2) ℝ) : lowCoef k s (rotM θ * g * rotM φ) = Complex.exp (-(k * (θ + φ) * Complex.I)) * lowCoef k s g

theorem **lowCoef_weight** (lam : InfinitePlace F → DomWeight 2) (v : InfinitePlace F) {g : Matrix (Fin 2) (Fin 2) ℝ} (h : 0 < g.det) : lowCoef (kn lam v) (-(cw lam v : ℂ)) g = (g.det : ℂ) ^ ((lam v).w 0 + 1) * (2 * Complex.I) ^ kn lam v / coefDen g ^ kn lam v

theorem **testInf_one** (lam : InfinitePlace F → DomWeight 2) : testInf lam 1 = 1

theorem **isOpen_iwa** : IsOpen (iwa w : Set (GL (Fin 2) (w.adicCompletion F)))

theorem **isCompact_iwa** : IsCompact (iwa w : Set (GL (Fin 2) (w.adicCompletion F)))

theorem **valuation_unif** : valuation (w.adicCompletion F) ((atFin w (unif w) : (w.adicCompletion F)ˣ) : w.adicCompletion F) = uniformizer (w.adicCompletion F)

theorem **tDiag_mul** (a : ℤ) (j : ℕ) : tDiag w (a * j) = tDiag w a ^ j

theorem **iwaInd_eq** (m : ℤ) : iwaInd w m = GlobalTheory.IndefiniteRowSideFullHecke.IwahoriDoubleCosets.cosetInd (tEnt w m)

@[simp] theorem **iwaHecke_toFun** (m : ℤ) : (iwaHecke w m).toFun = iwaInd w m

theorem **gf_ext** {g h : GF F} (e : ∀ v, atW v g = atW v h) : g = h

@[simp] theorem **atW_locGL** (x : GL (Fin 2) (w.adicCompletion F)) : atW w (locGL F 2 w x) = x

theorem **atW_locGL_of_ne** {v : Pl F} (h : v ≠ w) (x : GL (Fin 2) (w.adicCompletion F)) : atW v (locGL F 2 w x) = 1

theorem **awayW_mul_loc** (g : GF F) : awayW w g * locGL F 2 w (atW w g) = g

@[simp] theorem **atW_awayW** (g : GF F) : atW w (awayW w g) = 1

@[simp] theorem **awayW_locGL** (x : GL (Fin 2) (w.adicCompletion F)) : awayW w (locGL F 2 w x) = 1

@[simp] theorem **awayW_awayW** (g : GF F) : awayW w (awayW w g) = awayW w g

theorem **mem_levAt** {U : Subgroup (GF F)} {K : Subgroup (GL (Fin 2) (w.adicCompletion F))} {g : GF F} : g ∈ levAt w U K ↔ awayW w g ∈ U ∧ atW w g ∈ K

theorem **continuous_atW** : Continuous (atW w)

theorem **continuous_awayW** : Continuous (awayW w)

theorem **isOpen_levAt** {U : Subgroup (GF F)} {K : Subgroup (GL (Fin 2) (w.adicCompletion F))} (hU : IsOpen (U : Set (GF F))) (hK : IsOpen (K : Set (GL (Fin 2) (w.adicCompletion F)))) : IsOpen (levAt w U K : Set (GF F))

theorem **isCompact_levAt** {U : Subgroup (GF F)} {K : Subgroup (GL (Fin 2) (w.adicCompletion F))} (hU : IsCompact (U : Set (GF F))) (hK : IsCompact (K : Set (GL (Fin 2) (w.adicCompletion F)))) : IsCompact (levAt w U K : Set (GF F))

@[simp] theorem **HeckeFn.tensAt_apply** {U : Subgroup (GF F)} {K : Subgroup (GL (Fin 2) (w.adicCompletion F))} (f : HeckeFn U) (h : HeckeFn K) (g : GF F) : (f.tensAt w h).toFun g = f.toFun (awayW w g) * h.toFun (atW w g)

Development.AbelianVarietyTheory.Definitions.Divisors.HolomorphicSectionSignatureOneOne

theorem **sgn_sq** (q a : ℕ) : sgn q a * sgn q a = 1

theorem **OrthFr.det_ne** {Hm x₀ : Mt n} {q : ℕ} (h : OrthFr Hm q x₀) : x₀.det ≠ 0

theorem **OrthFr.mem** {Hm x₀ : Mt n} {q : ℕ} (h : OrthFr Hm q x₀) : x₀ ∈ frDom Hm q

theorem **mul_mem_frDom** {Hm u x : Mt n} {q : ℕ} (hH : Hm.det ≠ 0) (hu : u ∈ Formalism.QuantumDeterminants.UH Hm) (hx : x ∈ frDom Hm q) : u * x ∈ frDom Hm q

theorem **torEl_mem** {Hm x₀ : ι → Mt n} {q : ι → ℕ} (h : ∀ τ, OrthFr (Hm τ) (q τ) (x₀ τ)) (s : ι → Fin n → ℝ) : torEl x₀ s ∈ UG Hm

theorem **HolFn.torus** {Hm x₀ : ι → Mt n} {q : ι → ℕ} (h : ∀ τ, OrthFr (Hm τ) (q τ) (x₀ τ)) {w : ι → Fin n → ℤ} {f : (ι → Mt n) → ℂ} (hf : HolFn Hm q x₀ w f) (s : ι → Fin n → ℝ) {u : ι → Mt n} (hu : u ∈ UG Hm) : f (u * torEl x₀ s) = torCh w s * f u

theorem **weight_eq_zero_of_const** {Hm x₀ : ι → Mt n} {q : ι → ℕ} (h : ∀ τ, OrthFr (Hm τ) (q τ) (x₀ τ)) {w : ι → Fin n → ℤ} (hf : HolFn Hm q x₀ w fun _ => 1) (s : ι → Fin n → ℝ) : torCh w s = 1

theorem **holFn_const** (Hm x₀ : ι → Mt n) (q : ι → ℕ) (z : ℂ) : HolFn Hm q x₀ 0 fun _ => z

theorem **orthFrame_signatureOneOne** (τ : Unit) : OrthFr (H11 τ) 1 (x11 τ)

theorem **ne_zero_of_mem_frameDomain** {x : Mt 2} (hx : x ∈ frDom (H11 ()) 1) : x 1 0 ≠ 0

theorem **holomorphicSection_signatureOneOne** (k : ℤ) : HolSec H11 (fun _ => 1) (fun _ => ![-k, 0]) fun x => x () 1 0 ^ (-k)

theorem **holomorphicFunction_lowestVector** (k : ℤ) : HolFn H11 (fun _ => 1) x11 (fun _ => ![-k, 0]) fun u => u () 1 1 ^ (-k)

theorem **norm_lowestCoefficient_le_one** (k : ℕ) {u : Unit → Mt 2} (hu : u ∈ UG H11) : ‖u () 1 1 ^ (-(k : ℤ))‖ ≤ 1

theorem **exp_one_one** (z : ℂ) : (NormedSpace.exp (z • (1 : Mt 1)) : Mt 1) 0 0 = Complex.exp z

theorem **der_pow** (k : ℕ) (z : ℂ) (u : Unit → Mt 1) : der () (z • (1 : Mt 1)) (fun v => v () 0 0 ^ k) u = k * z * u () 0 0 ^ k

theorem **killedFn_pow** (k : ℕ) : KilledFn (fun _ : Unit => (1 : Mt 1)) (fun _ => 1) (fun _ _ => (k : ℤ)) fun v => v () 0 0 ^ k

theorem **exists_finPair** (V : HSp D) (hs : V.SigOK) : ∃ μ ℱ, FinPair V μ ℱ ∧ μ ℱ ≠ 0

theorem **fix_iff** (τ : D.host.E →+* ℂ) : τ D.host.sqrtMinusD = D.host.φ₀ D.host.sqrtMinusD ↔ ∃ ρ, τ = tauE D ρ

theorem **orthFr_x0Mat** {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (hM : HolShape M) (τ : D.what₂.Tau) : OrthFr (gramAt D V τ) (nNeg (M.ε τ)) (x0Mat M τ)

theorem **tauE_inj** : Function.Injective (tauE D)

theorem **gram_model** (V : HSp D) (ρ : D.what₂.Tau) (x : Mt V.rank) : ReciprocityTheory.ChartRestriction.G.FD.gram (tauE D ρ) (-V.gramᵀ) x = (-(xᴴ * gramAt D V ρ * x))ᵀ

theorem **plOf_tauE** {β : Type} (b : β) (f : D.what₂.Tau → β) (ρ : D.what₂.Tau) : plOf b f (tauE D ρ) = f ρ

theorem **goodDict_of_ctx** {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {σ : c.F →+* ℂ} {V : HSp D} (C : Ctx Xo c σ V V.rank) (hC : C.OK) {S : D.what₂.Tau → SPar} (M : DSModel V S) (hM : HolShape M) (hq : ∀ ρ, nNeg (M.ε ρ) = Dw.q) (w : D.what₂.Tau → Fin V.rank → ℤ) : ∃ θ dict Kf, IsLevelK V l.M Kf ∧ GoodDict θ M w rfl C.M.X (fun _ τ => plOf 0 w τ.1) dict (fun _ => 1) (fun _ => 1) Kf

theorem **exists_holModel** (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) (lam : Fin (D.sp.W j).rank → ℝ) (hl : Formalism.QuantumDeterminants.IsHCParam lam) (hσ : ∃ (H : Type) (_ : NormedAddCommGroup H) (_ : InnerProductSpace ℂ H) (σ : Mt (D.sp.W j).rank → (H →L[ℂ] H)), IsDS (fun a => decide ((a : ℕ) < (D.R.sigW j).1)) lam σ) : ∃ (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.W j) S), HolShape M ∧ (∀ ρ, M.lam ρ = lam) ∧ ∀ ρ, nNeg (M.ε ρ) = (D.R.sigW j).2

theorem **exists_goodDict** (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) (lam : Fin (D.sp.W j).rank → ℝ) (hl : Formalism.QuantumDeterminants.IsHCParam lam) (hσ : ∃ (H : Type) (_ : NormedAddCommGroup H) (_ : InnerProductSpace ℂ H) (σ : Mt (D.sp.W j).rank → (H →L[ℂ] H)), IsDS (fun a => decide ((a : ℕ) < (D.R.sigW j).1)) lam σ) (w : D.what₂.Tau → Fin (D.sp.W j).rank → ℤ) : ∃ (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.W j) S) (θ : InfFam D) (n : ℕ) (X : Fin n → Data D.host.E (D.sp.W j).rank) (la : ∀ k, Wt (X k).Ψ _) (dict : _) (Kf : Subgroup (GA D (D.sp.W j))), HolShape M ∧ (∀ ρ, M.lam ρ = lam) ∧ 0 < n ∧ 3 ≤ (D.sp.W j).rank ∧ IsLevelK (D.sp.W j) (D.host.g 0 * D.host.g 1) Kf ∧ GoodDict θ M w rfl X la dict (fun _ => 1) (fun _ => 1) Kf

theorem **exists_goodDict_last** (lam : Fin (D.sp.W D.R.m).rank → ℝ) (hl : Formalism.QuantumDeterminants.IsHCParam lam) (w : D.what₂.Tau → Fin (D.sp.W D.R.m).rank → ℤ) : ∃ (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.W D.R.m) S) (θ : InfFam D) (n : ℕ) (X : Fin n → Data D.host.E (D.sp.W D.R.m).rank) (la : ∀ k, Wt (X k).Ψ _) (dict : _) (Kf : Subgroup (GA D (D.sp.W D.R.m))), HolShape M ∧ (∀ ρ, M.lam ρ = lam) ∧ 0 < n ∧ 3 ≤ (D.sp.W D.R.m).rank ∧ IsLevelK (D.sp.W D.R.m) (D.host.g 0 * D.host.g 1) Kf ∧ GoodDict θ M w rfl X la dict (fun _ => 1) (fun _ => 1) Kf

theorem **exists_goodDict_of_discreteSeries** (h : GlobalTheory.Published.DiscreteSeriesExists) (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) (lam : Fin (D.sp.W j).rank → ℝ) (hl : Formalism.QuantumDeterminants.IsHCParam lam) (w : D.what₂.Tau → Fin (D.sp.W j).rank → ℤ) : ∃ (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.W j) S) (θ : InfFam D) (n : ℕ) (X : Fin n → Data D.host.E (D.sp.W j).rank) (la : ∀ k, Wt (X k).Ψ _) (dict : _) (Kf : Subgroup (GA D (D.sp.W j))), HolShape M ∧ (∀ ρ, M.lam ρ = lam) ∧ 0 < n ∧ 3 ≤ (D.sp.W j).rank ∧ IsLevelK (D.sp.W j) (D.host.g 0 * D.host.g 1) Kf ∧ GoodDict θ M w rfl X la dict (fun _ => 1) (fun _ => 1) Kf

theorem **exists_isReg** (V : HSp D) {μ : Measure (GA D V)} {ℱ : Set (GA D V)} (h : FinPair V μ ℱ) : ∃ ρ, IsReg V μ ℱ ρ

theorem **holForm_const** (θ : InfFam D) {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (J : Set (GA D V)) (z : ℂ) : HolForm θ M 0 J (fun _ => 1) fun _ => z

theorem **mem_GG_iff** (V : HSp D) (X : ReciprocityTheory.ChartRestriction.G.FD.Data D.host.E V.rank) (hX : ModelW V rfl X) (τ : X.Ψ) (ρ : D.what₂.Tau) (hτ : τ.1 = tauE D ρ) (x : ReciprocityTheory.ChartRestriction.G.FD.Frame X.Ψ V.rank) : ReciprocityTheory.ChartRestriction.G.FD.mat x τ ∈ frDom (gramAt D V ρ) X.p0 ↔ (ReciprocityTheory.ChartRestriction.G.FD.mat x τ).det ≠ 0 ∧ ∀ i : Fin (V.rank + 1), 0 < ReciprocityTheory.ChartRestriction.G.FD.psgn X.p0 i * (ReciprocityTheory.ChartRestriction.G.FD.dmin X.Gm x τ i).re

Development.AbelianVarietyTheory.Definitions.Divisors.HopfLaurent

theorem **lifts_const** {X : Scheme.{0}} (s : X ⟶ Sp R) {B : Type} (M : Type) [CommRing B] [Algebra R B] [AddCommGroup M] [Module B M] [Module Bᵐᵒᵖ M] [IsCentralScalar B M] (P : Sp B ⟶ X) (h : P ≫ s = spMap (algebraMap R B)) : Lifts s P (spMap (TrivSqZeroExt.inlHom B M) ≫ P)

theorem **G_hom** : (G R).hom = spMap (algebraMap R R[T;T⁻¹])

theorem **unit_eq** (B : Type) [CommRing B] [Algebra R B] : unitPt (G R) B = spMap (algebraMap R B) ≫ spMap (Bialgebra.counitAlgHom R R[T;T⁻¹]).toRingHom

theorem **comp'** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **inj'** {A B : Type} [CommRing A] [CommRing B] {p q : A →+* B} (h : spMap p = spMap q) : p = q

theorem **spMap_pre** {C : Type} [CommRing C] (z : Sp C ⟶ (G R).left) : spMap (pre z) = z

theorem **pre_spMap** {C : Type} [CommRing C] (φ : R[T;T⁻¹] →+* C) : pre (R

theorem **hom_ext'** {C : Type} [CommRing C] {φ ψ : R[T;T⁻¹] →+* C} (h0 : φ.comp (algebraMap R _) = ψ.comp (algebraMap R _)) (h1 : φ (T 1) = ψ (T 1)) : φ = ψ

theorem **lifts_iff** {B M : Type} [CommRing B] [Algebra R B] [AddCommGroup M] [Module B M] [Module Bᵐᵒᵖ M] [IsCentralScalar B M] (φ : R[T;T⁻¹] →+* TrivSqZeroExt B M) : Lifts (G R).hom (unitPt (G R) B) (spMap φ) ↔ φ.comp (algebraMap R _) = (TrivSqZeroExt.inlHom B M).comp (algebraMap R B) ∧ (φ (T 1)).fst = 1

Development.AbelianVarietyTheory.Definitions.Divisors.HtValOnSubset

theorem **htValOn_subset** (Z : Comp Xo c) (la : HeightOneSpectrum (𝒪 c.F)) : htValOn Z la ⊆ htValSet Xo c la

Development.AbelianVarietyTheory.Definitions.Divisors.HxiTZeroComb

theorem **pullEt_zero** {Y Y' : Scheme.{0}} (u : Y ⟶ Y') (n : ℤ) : pullEt l E u n 0 = pullE l E u n

theorem **opT_zero** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (p : S ⟶ B) (b : B' ⟶ B) (ε : Matrix (Fin m) (Fin m) O →₀ E) (n : ℤ) : opT l act p b ε n 0 = op l act p b ε n

theorem **HxiT_zero** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (p : S ⟶ B) (b : B' ⟶ B) (ε : Matrix (Fin m) (Fin m) O →₀ E) (k : ℕ) (i : ℤ) : HxiT l act p b ε k 0 i = Hxi l act p b ε k i

theorem **galEt_zero** (E : Type) [CommRing E] [Algebra ℚ_[l] E] {k : Type} [Field k] (Ω : Type) [Field Ω] [Algebra k Ω] {Y : Scheme.{0}} (q : Y ⟶ Spec (CommRingCat.of k)) (n : ℤ) (σ : Ω ≃ₐ[k] Ω) : galEt l E Ω q n 0 σ = galE l E Ω q n σ

@[simp] theorem **opT_zero_comb** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (p : S ⟶ B) (b : B' ⟶ B) (n t : ℤ) : opT l act p b (0 : Matrix (Fin m) (Fin m) O →₀ E) n t = 0

theorem **HxiT_zero_comb** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (p : S ⟶ B) (b : B' ⟶ B) (k : ℕ) (t i : ℤ) : HxiT l act p b (0 : Matrix (Fin m) (Fin m) O →₀ E) k t i = ⊥

theorem **opT_add** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (p : S ⟶ B) (b : B' ⟶ B) (ε ε' : Matrix (Fin m) (Fin m) O →₀ E) (n t : ℤ) : opT l act p b (ε + ε') n t = opT l act p b ε n t + opT l act p b ε' n t

Development.AbelianVarietyTheory.Definitions.Divisors.Hypotheses3

theorem **Hypotheses3.quoted** (h : Hypotheses3) : GlobalTheory.HSCore.Quoted

Development.AbelianVarietyTheory.Definitions.Divisors.HypothesisBundleConversions

theorem **joinOut_iff** : JoinOut ↔ GlobalTheory.IndefiniteRowSideFullHecke.ClassZetaValuationBounds.ToricDenominator ∧ GlobalTheory.IndefiniteRowSideFullHecke.LatticeNormTransport.Transport ∧ GlobalTheory.ToricTransport.TwistedTransport ∧ GlobalTheory.IndefiniteRowSideFullHecke.ZetaValueIdeals.ZetaDenominators ∧ GlobalTheory.SqTop.Refl

theorem **JoinOut.intro** (h₁ : GlobalTheory.IndefiniteRowSideFullHecke.ClassZetaValuationBounds.ToricDenominator) (h₂ : GlobalTheory.IndefiniteRowSideFullHecke.LatticeNormTransport.Transport) (h₃ : GlobalTheory.ToricTransport.TwistedTransport) (h₄ : GlobalTheory.IndefiniteRowSideFullHecke.ZetaValueIdeals.ZetaDenominators) (h₅ : GlobalTheory.SqTop.Refl) : JoinOut

theorem **JoinOut.transport** (J : JoinOut) : GlobalTheory.IndefiniteRowSideFullHecke.LatticeNormTransport.Transport

theorem **JoinOut.twisted** (J : JoinOut) : GlobalTheory.ToricTransport.TwistedTransport

theorem **JoinOut.zeta** (J : JoinOut) : GlobalTheory.IndefiniteRowSideFullHecke.ZetaValueIdeals.ZetaDenominators

theorem **LitCM.joinOut** (C : LitCM) : JoinOut

theorem **LitN.toE** (H : LitN) : LitNE

theorem **HypothesesU.toE** (P : HypothesesU) : HypothesesUE

theorem **HypothesesUE.toU** (P : HypothesesUE) (C : GlobalTheory.Mem.LiteralAutomorphy.LitCM) : HypothesesU

theorem **HypothesesUE.geometricLemma** (P : HypothesesUE) : GlobalTheory.Published.GeometricLemma

Development.AbelianVarietyTheory.Definitions.Divisors.IdlAlongNeTopOfValOn

theorem **trait_nonneg** (T : Trait F) (x : T.D.toPre.X) : 0 ≤ T.β x x

Development.AbelianVarietyTheory.Definitions.Divisors.InclEquiv

theorem **incl_injective** (X : Data F) (j : ℕ) : Function.Injective (incl X j)

Development.AbelianVarietyTheory.Definitions.Divisors.IndEtOfForallFinset

theorem **IndFEt.indEt** {A O : Type} [CommRing A] [CommRing O] [Algebra A O] (h : IndFEt A O) : IndEt A O

theorem **indEt_of_etale** (A O : Type) [CommRing A] [CommRing O] [Algebra A O] [Algebra.Etale A O] : IndEt A O

theorem **indFEt_of_finite_etale** (A O : Type) [CommRing A] [CommRing O] [Algebra A O] [Algebra.Etale A O] [Module.Finite A O] : IndFEt A O

theorem **indEt_of_forall_finset** (A O : Type) [CommRing A] [CommRing O] [Algebra A O] (h : ∀ s : Finset O, ∃ T : Subalgebra A O, Algebra.Etale A T ∧ (s : Set O) ⊆ T) : IndEt A O

theorem **contCone_idCone** (F : Shv R) (E : Spc) : ContCone R F (idCone E)

Development.AbelianVarietyTheory.Definitions.Divisors.IntLatGEqCanLatOfSurjective

theorem **mem_rangeSub** {x : c.S} : x ∈ rangeSub c ↔ x ∈ Set.range (algebraMap Γ(X, c.U.1) c.S)

theorem **mem_intLatG_of** {h : HdR c.A} (hh : h ∈ canSet s c) (hi : ∀ c₀, IsIntClass c c₀ → h c₀ ∈ Set.range (algebraMap Γ(X, c.U.1) c.S)) : h ∈ intLatG s c

theorem **mem_intLat_of** {h : HdR c.A} (hh : h ∈ canSet s c) (hi : ∀ ψ : LinearMap.BilinForm c.S (HdR c.A), IsPolForm (dp0 c.S) (𝟙 _) c.A (t.PP.L.pull c.φ) ψ → ∀ w ∈ intLatG s c, ψ h w ∈ Set.range (algebraMap Γ(X, c.U.1) c.S)) : h ∈ intLat s c

theorem **intLatG_apply_mem** {h : HdR c.A} (hh : h ∈ intLatG s c) {c₀ : primDR c.A} (hc : IsIntClass c c₀) : h c₀ ∈ Set.range (algebraMap Γ(X, c.U.1) c.S)

theorem **intLat_pair_mem** {h : HdR c.A} (hh : h ∈ intLat s c) {ψ : LinearMap.BilinForm c.S (HdR c.A)} (hψ : IsPolForm (dp0 c.S) (𝟙 _) c.A (t.PP.L.pull c.φ) ψ) {w : HdR c.A} (hw : w ∈ intLatG s c) : ψ h w ∈ Set.range (algebraMap Γ(X, c.U.1) c.S)

theorem **intLatG_le_canLat** : intLatG s c ≤ canLat s c

theorem **intLat_le_canLat'** : intLat s c ≤ canLat s c

theorem **intLatG_eq_canLat_of_surjective** (hs : Function.Surjective (algebraMap Γ(X, c.U.1) c.S)) : intLatG s c = canLat s c

theorem **intLat_eq_canLat_of_surjective'** (hs : Function.Surjective (algebraMap Γ(X, c.U.1) c.S)) : intLat s c = canLat s c

Development.AbelianVarietyTheory.Definitions.Divisors.IntegralBound

theorem **regSeq_cons** {Y : Scheme.{0}} {L : LB Y} (hL : L.IsLB) (F : Y.Modules) (a : L.Sec) (r : List L.Sec) : RegSeq hL F (a :: r) ↔ Mono (GlobalTheory.cutHom F hL a) ∧ RegSeq hL (GlobalTheory.cut F hL a) r

Development.AbelianVarietyTheory.Definitions.Divisors.InvVolConjLevelOne

theorem **exists_finEmb** {A B : HSp D} {c : ℕ} {ε : D.host.E} (d : HermitianOrthogonalExtensionData A B c ε) (y : GfC A) : ∃ x : GfC B, EmbRel B A d.ι x.1 y.1

theorem **finEmb_rel** {A B : HSp D} {c : ℕ} {ε : D.host.E} (d : HermitianOrthogonalExtensionData A B c ε) (y : GfC A) : EmbRel B A d.ι (finEmb d y).1 y.1

theorem **readOK_of_equiv** (φ : VV ℂ σ ≃ₗ[ℂ] VV ℂ fun τ => (σ τ).dual) (h : ∀ F, IsFunForm σ F (φ F)) : ReadOK σ

theorem **one_le_sLen** : 1 ≤ sLen D

theorem **isUnitVol_unitVol** (w : Pl D.host.E) : IsUnitVol (unitVol w)

theorem **invVol_conjLevel_one** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {Γ : Grid k} (hs : 1 ≤ Γ.s) {j : ℕ} (hj : j < Γ.m) : invVol (conjLevel hs (fun _ => KO k _) 1 hj) = 1

theorem **unfTerm_coef_one** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {Γ : Grid k} (hs : 1 ≤ Γ.s) (ψ : AddChar k Circle) (ν : Measure k) (keep : Bool) (S : Γ.Args → ℂ) : (unfTerm ψ ν keep (Reps.through hs (fun _ => KO k _) fun _ => 1) S fun _ => DoubleCoset.mk _ _ 1).coef = 1

theorem **aaScalar_eq_one** {w : Pl D.host.E} (h : FractionalIdeal.count D.host.E w D.host.AA = 0) (h' : FractionalIdeal.count D.host.E w D.host.AAbar = 0) (hψ : ∀ a ∈ 𝒪[Kw w], psiAt W w a = 1) : aaScalar W w = 1

theorem **topRank_eq** (t : T) (w : Pl D.host.E) {j : ℕ} (h : j ≤ D.R.m) : topRank (locGrid G t w) j = (G.V (.top j)).rank

theorem **vCore_smul** (a b : ℂ) (Ψ F) : S.vCore t Φ (a • Ψ) (b • F) = a * b * S.vCore t Φ Ψ F

theorem **SumOK.vCore_add** (h : S.SumOK t Φ) (Ψ F F') : S.vCore t Φ Ψ (F + F') = S.vCore t Φ Ψ F + S.vCore t Φ Ψ F'

theorem **SumOK.vCore_eq** (h : S.SumOK t Φ) (τ : CosFam G t → PureDat G t →₀ ℂ) (hτ : ∀ β ∈ S.supp t Φ, S.IsExpansion t Φ β (τ β)) (Ψ F) : S.vCore t Φ Ψ F = ∑ β ∈ h.fin.toFinset, (∏ w, ((S.term t Φ β w).coef : ℂ)) * S.classSum Φ β (τ β) Ψ F

theorem **wt_top_eq** : wt G (.top 0) = LaP G

theorem **rank_zero_m** : D.sp.X'.rank = (G.V (.grid 0 D.R.m)).rank

theorem **vPair_smul** (c : ℂ) (u : (R.HX t).H) (Ψ Φ) : vPair W S t (c • u) Ψ Φ = c * vPair W S t u Ψ Φ

theorem **vPair_add** {Φ : G.Datum} (h : S.SumOK t Φ) (u u' : (R.HX t).H) (Ψ) : vPair W S t (u + u') Ψ Φ = vPair W S t u Ψ Φ + vPair W S t u' Ψ Φ

theorem **vSum_smul** (c : D.what₂.LL) (F : A.fam.amb.FR t) (a : Fin n) : vSum W S rp e t (c • F) a = D.what₂.embLL c * vSum W S rp e t F a

theorem **vSum_add** {a : Fin n} (h : S.SumOK t (datumOf e a)) (F F' : A.fam.amb.FR t) : vSum W S rp e t (F + F') a = vSum W S rp e t F a + vSum W S rp e t F' a

Development.AbelianVarietyTheory.Definitions.Divisors.IotaScalarOne

theorem **iotaScalar_one** {C : Type} [Field C] (N : ℕ) : IotaScalar (1 : Matrix (Fin N) (Fin N) C) 0 0 1

theorem **lead_pow** {R C : Type} [CommRing R] [Field C] (f : R →+* C) (ξ : R) (n : ℕ) : Lead f ξ (ξ ^ n) n 1

Development.AbelianVarietyTheory.Definitions.Divisors.IrreducibleOfIrreducibleMap

theorem **isMultiHomogeneous_iff_block** {B : ℕ} {a : Fin B → ℕ} {f : MvPolynomial (Fin B × Fin N) k} : ReciprocityTheory.MultiplicativeHeights.IsMultiHomogeneous a f ↔ GlobalTheory.Chow.IsBlockHomogeneous f a

theorem **irreducible_of_irreducible_map** {K σ : Type} [Field K] [Algebra k K] {f : MvPolynomial σ k} (h : Irreducible (MvPolynomial.map (algebraMap k K) f)) : Irreducible f

theorem **IsChowForm.isChowFormInt** {I : Ideal (MvPolynomial (Fin N) k)} {e dl : ℕ} {f : MvPolynomial (Fin (e + 1) × Fin N) k} (h : IsChowForm I e dl f) : IsChowFormInt I e dl f

theorem **ne_zero** (h : IsChowFormInt I e dl f) : f ≠ 0

theorem **isChowForm** (h : IsChowFormInt I e dl f) (hf : Irreducible (MvPolynomial.map (algebraMap k (AlgebraicClosure k)) f)) : IsChowForm I e dl f

theorem **C_mul** (h : IsChowFormInt I e dl f) {c : k} (hc : c ≠ 0) : IsChowFormInt I e dl (C c * f)

theorem **dvd_of_forall_aeval_eq_zero** (h : IsChowFormInt I e dl f) {p : MvPolynomial (Fin (e + 1) × Fin N) k} (hp : ∀ u : Fin (e + 1) × Fin N → AlgebraicClosure k, aeval u f = 0 → aeval u p = 0) : f ∣ p

theorem **unique** (h : IsChowFormInt I e dl f) (h' : IsChowFormInt I e dl' f') : dl = dl' ∧ ∃ c : k, c ≠ 0 ∧ f' = C c * f

theorem **span_eq_radical_elimIdeal** (h : IsChowFormInt I e dl f) (hI : I.IsHomogeneous (homogeneousSubmodule (Fin N) k)) : Ideal.span {f} = (GlobalTheory.Chow.elimIdeal (Fin (e + 1)) I).radical

theorem **zero_iff_of_isAlgClosed** (h : IsChowFormInt I e dl f) (hI : I.IsHomogeneous (homogeneousSubmodule (Fin N) k)) {Ω : Type*} [Field Ω] [IsAlgClosed Ω] [Algebra k Ω] (u : Fin (e + 1) × Fin N → Ω) : MvPolynomial.aeval u f = 0 ↔ ∃ y : Fin N → Ω, y ≠ 0 ∧ (∀ g ∈ I, MvPolynomial.aeval y g = 0) ∧ ∀ i, ∑ j, u (i, j) * y j = 0

theorem **exists_point** (h : IsChowFormInt I e dl f) : ∃ y : Fin N → AlgebraicClosure k, y ≠ 0 ∧ ∀ g ∈ I, MvPolynomial.aeval y g = 0

theorem **span_eq_elimIdeal** (h : IsChowFormInt I e dl f) (hp : I.IsPrime) (hI : I.IsHomogeneous (homogeneousSubmodule (Fin N) k)) : Ideal.span {f} = GlobalTheory.Chow.elimIdeal (Fin (e + 1)) I

theorem **pos** (h : IsChowFormInt I e dl f) : 0 < dl

theorem **single** (h : IsChowFormInt I e dl f) (hp : I.IsPrime) (hI : I.IsHomogeneous (homogeneousSubmodule (Fin N) k)) : IsChowFormCycle (Finsupp.single I 1) e dl f

theorem **not_isChowFormInt_top** {e dl : ℕ} (f : MvPolynomial (Fin (e + 1) × Fin N) k) : ¬ IsChowFormInt (⊤ : Ideal (MvPolynomial (Fin N) k)) e dl f

theorem **component** (h : IsChowFormCycle Z e dl f) {P : Ideal (MvPolynomial (Fin N) k)} (hP : P ∈ Z.support) : P.IsPrime ∧ P.IsHomogeneous (homogeneousSubmodule (Fin N) k) ∧ ∃ d g, IsChowFormInt P e d g

theorem **ne_zero** (h : IsChowFormCycle Z e dl f) : f ≠ 0

theorem **homog** (h : IsChowFormCycle Z e dl f) : ReciprocityTheory.MultiplicativeHeights.IsMultiHomogeneous (fun _ => dl) f

theorem **C_mul** (h : IsChowFormCycle Z e dl f) {a : k} (ha : a ≠ 0) : IsChowFormCycle Z e dl (C a * f)

theorem **eq_of_components** (h : IsChowFormCycle Z e dl f) (d : Ideal (MvPolynomial (Fin N) k) → ℕ) (g : Ideal (MvPolynomial (Fin N) k) → MvPolynomial (Fin (e + 1) × Fin N) k) (hg : ∀ P ∈ Z.support, IsChowFormInt P e (d P) (g P)) : dl = Z.sum (fun P m => m * d P) ∧ ∃ c : k, c ≠ 0 ∧ f = C c * Z.prod (fun P m => g P ^ m)

theorem **unique** (h : IsChowFormCycle Z e dl f) (h' : IsChowFormCycle Z e dl' f') : dl = dl' ∧ ∃ c : k, c ≠ 0 ∧ f' = C c * f

theorem **add** (h : IsChowFormCycle Z e dl f) (h' : IsChowFormCycle Z' e dl' f') : IsChowFormCycle (Z + Z') e (dl + dl') (f * f')

theorem **zero_iff_of_isAlgClosed** (h : IsChowFormCycle Z e dl f) {Ω : Type*} [Field Ω] [IsAlgClosed Ω] [Algebra k Ω] (u : Fin (e + 1) × Fin N → Ω) : MvPolynomial.aeval u f = 0 ↔ ∃ P ∈ Z.support, ∃ y : Fin N → Ω, y ≠ 0 ∧ (∀ g ∈ P, MvPolynomial.aeval y g = 0) ∧ ∀ i, ∑ j, u (i, j) * y j = 0

theorem **isChowFormCycle_zero_iff** {e dl : ℕ} {f : MvPolynomial (Fin (e + 1) × Fin N) k} : IsChowFormCycle 0 e dl f ↔ dl = 0 ∧ ∃ c : k, c ≠ 0 ∧ f = C c

theorem **IsChowFormCycle.nsmul** {Z : Ideal (MvPolynomial (Fin N) k) →₀ ℕ} {e dl : ℕ} {f : MvPolynomial (Fin (e + 1) × Fin N) k} (h : IsChowFormCycle Z e dl f) (m : ℕ) : IsChowFormCycle (m • Z) e (m * dl) (f ^ m)

theorem **isChowFormCycle_single_iff** {P : Ideal (MvPolynomial (Fin N) k)} {e dl : ℕ} {f : MvPolynomial (Fin (e + 1) × Fin N) k} : IsChowFormCycle (Finsupp.single P 1) e dl f ↔ P.IsPrime ∧ P.IsHomogeneous (homogeneousSubmodule (Fin N) k) ∧ IsChowFormInt P e dl f

theorem **irreducible_sq_add_sq** : Irreducible (X 0 ^ 2 + X 1 ^ 2 : MvPolynomial (Fin 2) ℚ)

theorem **isPrime_span_sq_add_sq** : (Ideal.span {(X 0 ^ 2 + X 1 ^ 2 : MvPolynomial (Fin 2) ℚ)}).IsPrime

theorem **isHomogeneous_span_sq_add_sq** : (Ideal.span {(X 0 ^ 2 + X 1 ^ 2 : MvPolynomial (Fin 2) ℚ)}).IsHomogeneous (homogeneousSubmodule (Fin 2) ℚ)

theorem **isChowFormInt_sq_add_sq** : IsChowFormInt (Ideal.span {(X 0 ^ 2 + X 1 ^ 2 : MvPolynomial (Fin 2) ℚ)}) 0 2 (X (0, 0) ^ 2 + X (0, 1) ^ 2)

theorem **not_isChowForm_sq_add_sq** (δ : ℕ) (f : MvPolynomial (Fin (0 + 1) × Fin 2) ℚ) : ¬ IsChowForm (Ideal.span {(X 0 ^ 2 + X 1 ^ 2 : MvPolynomial (Fin 2) ℚ)}) 0 δ f

theorem **isChowFormCycle_two_sq_add_sq** : IsChowFormCycle (Finsupp.single (Ideal.span {(X 0 ^ 2 + X 1 ^ 2 : MvPolynomial (Fin 2) ℚ)}) 2) 0 4 ((X (0, 0) ^ 2 + X (0, 1) ^ 2) ^ 2)

Development.AbelianVarietyTheory.Definitions.Divisors.IsActIffIsDiagEnd

theorem **isAct_iff_isDiagEnd** {X₁ : Setup F W i} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A →ₗ X₁.ρ.P.A)} : IsAct X₁ act ↔ IsDiagEnd F W X₁.ρ.P X₁.u act

theorem **isAct_iff** {X₁ : Setup F W i} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A →ₗ X₁.ρ.P.A)} : IsAct X₁ act ↔ ∃ c : SiegPre g n (zetaAway n) (X₁.u ≫ F.s), SiegPB c F.univ X₁.u ∧ ∃ pr : Fin 8 → (X₁.ρ.P.A →ₗ c.A), (∀ a, IsMonHom (pr a)) ∧ (∀ (T' : Over (Sp X₁.S.R)) (y : Fin 8 → (T' →ₗ c.A)), ∃! x : T' →ₗ X₁.ρ.P.A, ∀ a, x ≫ pr a = y a) ∧ LBIso X₁.ρ.P.L (LB.LinComb c.P (K

theorem **IsAct.isMonHom** {X₁ : Setup F W i} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A →ₗ X₁.ρ.P.A)} (h : IsAct X₁ act) (E : Matrix (Fin 8) (Fin 8) ℤ) : IsMonHom (act E)

theorem **nonempty_isAct** (X₁ : Setup F W i) : Nonempty {act // IsAct X₁ act}

theorem **UMem.mono_ne_zero** (mu : Fin 8 → X) : mono W t δ mu ≠ 0

theorem **UMem.mono_mem** {mu : Fin 8 → X} (h : mu ≠ 0) : mono W t δ mu ∈ maximalIdeal V

theorem **exists_unif_good** {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] (hV : IsCN V) (P : PolAV (Sp V)) : letI

theorem **UMem.nonempty_of_regular** {X₁ : Setup F W i} {z₁ : X₁.M.Z} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A →ₗ X₁.ρ.P.A)} (hact : IsAct X₁ act) (hz : X₁.S.R ≤ X₁.M.O z₁) {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] (hV : IsCN V) (ev : X₁.M.Z.presheaf.stalk z₁ →+* V) (hev : IsLocalHom ev) (X : Type) [AddCommGroup X] [Subsingleton X] {m : ℕ} (t : Fin m → V) (δ : Fin m → LinearMap.BilinForm ℤ X) : Nonempty (UMem X₁ z₁ act V ev X t δ)

theorem **exists_uMem** (X₁ : Setup F W i) : ∃ act, IsAct X₁ act ∧ ∃ (z₁ : X₁.M.Z) (ev : X₁.M.Z.presheaf.stalk z₁ →+* X₁.Ω₁), Nonempty (UMem X₁ z₁ act X₁.Ω₁ ev PUnit (m

Development.AbelianVarietyTheory.Definitions.Divisors.IsAddHaarMeasureVolumeLie

theorem **schur_zero** (n : ℕ) : schur n (fun _ => 0) = 1

theorem **sphW_of_not_antitone** (q : ℕ) {n : ℕ} (α : Fin n → ℂ) {l : Fin n → ℕ} (h : ¬ Antitone l) : sphW q α l = 0

theorem **IsTestInf.mul** {f h : Lie E a → ℂ} (hf : IsTestInf f) (hh : ContDiffOn ℝ ∞ h (invSet E a)) : IsTestInf fun x => f x * h x

theorem **exists_isTestInf** {y : Lie E a} (hy : y ∈ invSet E a) {U : Set (Lie E a)} (hU : U ∈ nhds y) : ∃ ρ : Lie E a → ℝ, IsTestInf (fun x => (ρ x : ℂ)) ∧ tsupport ρ ⊆ U ∧ (∀ x, ρ x ∈ Set.Icc (0 : ℝ) 1) ∧ ρ y = 1

theorem **IsTestInf.integrable** {f : Lie E a → ℂ} (hf : IsTestInf f) {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) : Integrable fun x => f x * chart φ g x

theorem **smoothInf_smul** (f : Lie E a → ℂ) (c : ℂ) (φ : GA E a → ℂ) : smoothInf f (c • φ) = c • smoothInf f φ

theorem **smoothInf_left** (f : Lie E a → ℂ) (φ : GA E a → ℂ) (γ g : GA E a) : smoothInf f (fun x => φ (γ * x)) g = smoothInf f φ (γ * g)

theorem **smoothInf_rt_fin** (f : Lie E a → ℂ) (φ : GA E a → ℂ) (k : GL (Fin a) (Af E)) (g : GA E a) : smoothInf f (fun x => φ (x * finGL E a k)) g = smoothInf f φ (g * finGL E a k)

theorem **smoothInf_mem_smoothSpan** (π : CuspSub E a) {f : Lie E a → ℂ} (hf : IsTestInf f) {φ : Sm E a} (hφ : φ ∈ π.V) : smoothInf f φ.1 ∈ smoothSpan π

theorem **exists_smoothInf_ne_zero** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) {g : GA E a} (hg : φ g ≠ 0) : ∃ f : Lie E a → ℂ, IsTestInf f ∧ smoothInf f φ g ≠ 0

theorem **smoothSpan_ne_bot** (π : CuspSub E a) : smoothSpan π ≠ ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.IsAdelicOne

theorem **isAdelic_one** : IsAdelic (fun _ : HeightOneSpectrum (𝒪 K) => (1 : Matrix ι ι K))

Development.AbelianVarietyTheory.Definitions.Divisors.IsAdmissibleIntegralBasisChangeOne

theorem **isFF_iff** : IsFF k F ↔ IsFunctionField k F

theorem **valuation_toRing** [IsFunctionField k F] (v : Nef.Place k F) : (toRing v).valuation = v.1

theorem **ord_toRing** [IsFunctionField k F] (v : Nef.Place k F) (x : F) : (toRing v).ord x = ord v x

theorem **sum_ramificationIdx** [IsAlgClosed k] {E : Type} [Field E] [Algebra k E] [Algebra F E] [IsScalarTower k F E] [IsFunctionField k F] [IsFunctionField k E] [FiniteDimensional F E] [Algebra.IsSeparable F E] (v : GroundTheory.FunctionField.Place k F) : ∑ w ∈ (v.finite_liesOver (K'

Development.AbelianVarietyTheory.Definitions.Divisors.IsAffCoverSpec

theorem **mem_filAt_iff** {p : ℕ} {m : ℤ} {a : Alt n} : a ∈ filAt n p m ↔ ∀ T, (bas n).repr a T ≠ 0 → (T.card : ℤ) = m ∧ p ≤ holCount T

theorem **mem_filAt_nat** {p k : ℕ} {a : Alt n} : a ∈ filAt n p k ↔ HasDeg k a ∧ a ∈ holGe n p

theorem **mem_smCx_C** {p : ℕ} {m : ℤ} {α : smForms A} : α ∈ (smCx A p).C m ↔ ∀ x, α.1 x ∈ filAt n p m

theorem **mem_smCx_C_nat** {p k : ℕ} {α : smForms A} : α ∈ (smCx A p).C k ↔ A.IsDeg k α ∧ A.TypeGe p α

theorem **smCx_C_neg** {p : ℕ} {m : ℤ} (hm : m < 0) : (smCx A p).C m = ⊥

theorem **mem_smCx_Z_nat** {p k : ℕ} {α : smForms A} : α ∈ (smCx A p).Z k ↔ α.1 ∈ A.closedForms k ∧ A.TypeGe p α

theorem **mem_smCx_B_nat** {k : ℕ} {α : smForms A} : α ∈ (smCx A 0).B k ↔ α.1 ∈ A.exactForms k

theorem **isCx_smCx** (p : ℕ) : (smCx A p).IsCx

@[simp] theorem **coe_zEquiv** (k : ℕ) (α : (smCx A 0).Z k) : (zEquiv A k α : A.Fld) = α.1.1

theorem **hsmEquiv_mk** (k : ℕ) (α : (smCx A 0).Z k) : hsmEquiv A k (Submodule.Quotient.mk α) = HolAtlas.HSm.mk A k (zEquiv A k α)

theorem **mem_FSm_iff** (p k : ℕ) (c : HSm A k) : c ∈ FSm A p k ↔ hsmEquiv A k c ∈ HolAtlas.HSm.fil A p k

theorem **map_FSm** (p k : ℕ) : (FSm A p k).map (hsmEquiv A k).toLinearMap = HolAtlas.HSm.fil A p k

theorem **FormOn.of_isForm** {α : A.Fld} (h : A.IsForm α) (S : Set M) : FormOn A S α where

theorem **FormOn.zero** (S : Set M) : FormOn A S 0

theorem **isAffCover_spec** (K : Type) [Field K] : letI

Development.AbelianVarietyTheory.Definitions.Divisors.IsAffineOpenAndIsPushout

theorem **appLE_appLE** {f : X'' ⟶ X'} {g : X' ⟶ X} {k : X'' ⟶ X} (w : f ≫ g = k) {U : X.Opens} {V : X'.Opens} {W : X''.Opens} (e₁ : V ≤ g ⁻¹ᵁ U) (e₂ : W ≤ f ⁻¹ᵁ V) (e₃ : W ≤ k ⁻¹ᵁ U) (a : Γ(X, U)) : f.appLE V W e₂ (g.appLE U V e₁ a) = k.appLE U W e₃ a

theorem **id_appLE** {U V : X.Opens} (h : V ≤ U) (a : Γ(X, U)) : (𝟙 X : X ⟶ X).appLE U V h a = resG X h a

theorem **le_preimage_of_comp** {f : X'' ⟶ X'} {g : X' ⟶ X} {k : X'' ⟶ X} (w : f ≫ g = k) {U : X.Opens} {W : X''.Opens} : W ≤ f ⁻¹ᵁ g ⁻¹ᵁ U ↔ W ≤ k ⁻¹ᵁ U

theorem **resM_cls** (h : V ≤ e ⁻¹ᵁ W) (h' : V' ≤ V) (f : Γ(X, W)) (hf) : resM _ h' (cls h f hf) = cls (h'.trans h) f hf

theorem **cls_congr** (h : V ≤ e ⁻¹ᵁ W) (h' : V ≤ e ⁻¹ᵁ W') (hT : V ≤ e ⁻¹ᵁ T) (k : T ≤ W) (k' : T ≤ W') {f : Γ(X, W)} {f' : Γ(X, W')} (hf hf') (H : resG X k f = resG X k' f') : cls h f hf = cls h' f' hf'

theorem **cls_add** (h : V ≤ e ⁻¹ᵁ W) (f g : Γ(X, W)) (hf hg hfg) : cls h (f + g) hfg = cls h f hf + cls h g hg

theorem **cls_mul** (h : V ≤ e ⁻¹ᵁ W) (a f : Γ(X, W)) (hf haf) : cls h (a * f) haf = e.appLE W V h a • cls h f hf

theorem **cls_zero** (h : V ≤ e ⁻¹ᵁ W) {f : Γ(X, W)} (hf) (h0 : f = 0) : cls h f hf = 0

theorem **le_preimage_iff** : V ≤ e ⁻¹ᵁ π ⁻¹ᵁ U ↔ V ≤ U

theorem **appLE_base** (h : V ≤ e ⁻¹ᵁ W) (k : W ≤ π ⁻¹ᵁ U) (hV : V ≤ U) (a : Γ(S, U)) : e.appLE W V h (π.appLE U W k a) = resG S hV a

theorem **le_nhd** (h : V ≤ e ⁻¹ᵁ W) : V ≤ e ⁻¹ᵁ (W ⊓ π ⁻¹ᵁ e ⁻¹ᵁ W)

theorem **center_apply** (f : Γ(X, W)) : center e W π f = resG X inf_le_left f - π.appLE _ _ inf_le_right (e.appLE W _ le_rfl f)

theorem **appLE_center** (f : Γ(X, W)) : e.appLE _ _ le_rfl (center e W π f) = 0

theorem **d_apply** (h : V ≤ e ⁻¹ᵁ W) (f : Γ(X, W)) : d he h f = cls (le_nhd he h) (center e W π f) (appLE_center he f)

theorem **d_eq_cls** (h : V ≤ e ⁻¹ᵁ W) (f : Γ(X, W)) (hf : e.appLE W _ le_rfl f = 0) : d he h f = cls h f hf

theorem **resM_d** (h : V ≤ e ⁻¹ᵁ W) (h' : V' ≤ V) (f : Γ(X, W)) : resM _ h' (d he h f) = d he (h'.trans h) f

theorem **d_resG** (h : V ≤ e ⁻¹ᵁ W') (k : W' ≤ W) (f : Γ(X, W)) : d he h (resG X k f) = d he (h.trans (e.preimage_mono k)) f

theorem **d_mul** (h : V ≤ e ⁻¹ᵁ W) (f g : Γ(X, W)) : d he h (f * g) = e.appLE W V h f • d he h g + e.appLE W V h g • d he h f

theorem **d_base** (h : V ≤ e ⁻¹ᵁ W) (k : W ≤ π ⁻¹ᵁ U) (a : Γ(S, U)) : d he h (π.appLE U W k a) = 0

theorem **d_one** (h : V ≤ e ⁻¹ᵁ W) : d he h 1 = 0

theorem **hom_ext** {φ ψ : conormal e ⟶ F} (H : ∀ (W : X.Opens) (f : Γ(X, W)), φ.app _ (d he (le_refl (e ⁻¹ᵁ W)) f) = ψ.app _ (d he le_rfl f)) : φ = ψ

theorem **lift_d** (D : Der π e F) (h : V ≤ e ⁻¹ᵁ W) (f : Γ(X, W)) : D.lift.app V (d he h f) = D.app h f

theorem **homEquivDer_app** (φ : conormal e ⟶ F) (h : V ≤ e ⁻¹ᵁ W) (f : Γ(X, W)) : (homEquivDer F he φ).app h f = φ.app V (d he h f)

theorem **map_d** (h : V ≤ e ⁻¹ᵁ W) {W' : X'.Opens} (h' : V ≤ e' ⁻¹ᵁ W') (k : W' ≤ φ ⁻¹ᵁ W) (f : Γ(X, W)) : (map he' w we).app V (d he h f) = d he' h' (φ.appLE W W' k f)

theorem **map_id** {φ : X ⟶ X} (hφ : φ = 𝟙 X) (w : φ ≫ π = π) (we : e ≫ φ = e) : map he w we = 𝟙 _

theorem **map_comp** (he'' : e'' ≫ π'' = 𝟙 S) (w' : ψ ≫ π' = π'') (we' : e'' ≫ ψ = e') {χ : X'' ⟶ X} (hχ : ψ ≫ φ = χ) (w'' : χ ≫ π = π'') (we'' : e'' ≫ χ = e) : map he' w we ≫ map he'' w' we' = map he'' w'' we''

theorem **map_eq_zero** (hφ : π' ≫ e = φ) : map he' w we = 0

theorem **le_of_sq** (we : e' ≫ φ = f ≫ e) (h : V ≤ e ⁻¹ᵁ W) {T : S'.Opens} (hT : T ≤ f ⁻¹ᵁ V) : T ≤ e' ⁻¹ᵁ φ ⁻¹ᵁ W

theorem **baseMap_d** (h : V ≤ e ⁻¹ᵁ W) {T : S'.Opens} (hT : T ≤ f ⁻¹ᵁ V) {W' : X'.Opens} (h' : T ≤ e' ⁻¹ᵁ W') (k : W' ≤ φ ⁻¹ᵁ W) (a : Γ(X, W)) : (baseMap he' w we).app T (pullbackSections f _ hT (d he h a)) = d he' h' (φ.appLE W W' k a)

theorem **pullback_hom_ext** {F : S'.Modules} {α β : (Scheme.Modules.pullback f).obj (conormal e) ⟶ F} (H : ∀ (W : X.Opens) (a : Γ(X, W)), α.app _ (pullbackSections f _ (le_refl (f ⁻¹ᵁ e ⁻¹ᵁ W)) (d he le_rfl a)) = β.app _ (pullbackSections f _ le_rfl (d he le_rfl a))) : α = β

theorem **isAffineOpen_and_isPushout** {Y T : Scheme.{0}} {f : T ⟶ S} {g : Y ⟶ X} {p : Y ⟶ T} (H : IsPullback g p π f) {T₀ : T.Opens} (hU : IsAffineOpen U) (hT : IsAffineOpen T₀) (hW : IsAffineOpen W) (k : T₀ ≤ f ⁻¹ᵁ U) (k' : W ≤ π ⁻¹ᵁ U) : IsAffineOpen (g ⁻¹ᵁ W ⊓ p ⁻¹ᵁ T₀) ∧ IsPushout (π.appLE U W k') (f.appLE U T₀ k) (g.appLE W (g ⁻¹ᵁ W ⊓ p ⁻¹ᵁ T₀) inf_le_left) (p.appLE T₀ _ inf_le_right)

theorem **exists_isChart_inf** (hU : IsAffineOpen U) (hW : IsAffineOpen W) (k : W ≤ π ⁻¹ᵁ U) {x : S} (hx : e x ∈ W) (hV : x ∈ V) : ∃ U', x ∈ U' ∧ U' ≤ V ∧ U' ≤ U ∧ IsChart π e U' (W ⊓ π ⁻¹ᵁ U')

theorem **exists_isChart** {x : S} (hV : x ∈ V) : ∃ U W, x ∈ U ∧ U ≤ V ∧ IsChart π e U W

theorem **aug_apply** (a : Γ(X, W)) : c.aug he a = e.appLE W U c.le a

theorem **aug_eq_zero_iff** {a : Γ(X, W)} : c.aug he a = 0 ↔ e.appLE W _ le_rfl a = 0

theorem **ofAugCot_d** (a : Γ(X, W)) : letI

theorem **ofAugCot_bijective** : Function.Bijective (c.ofAugCot he)

theorem **augCotEquiv_d** (a : Γ(X, W)) : letI

theorem **d_surjective** : Function.Surjective (d he c.le)

theorem **kaehlerEquiv_d** (a : Γ(X, W)) : letI

theorem **hom_ext_of_local** {α β : F ⟶ F'} (H : ∀ (V : S.Opens) (x : S), x ∈ V → ∃ U : S.Opens, x ∈ U ∧ U ≤ V ∧ ∀ s, α.app U s = β.app U s) : α = β

theorem **prod_hom_ext** {Y P : Scheme.{0}} {π₂ : Y ⟶ S} {e₂ : S ⟶ Y} (he₂ : e₂ ≫ π₂ = 𝟙 S) {p₁ : P ⟶ X} {p₂ : P ⟶ Y} (H : IsPullback p₁ p₂ π π₂) {q : P ⟶ S} {s : S ⟶ P} [QuasiCompact s] (hs : s ≫ q = 𝟙 S) (w₁ : p₁ ≫ π = q) (w₂ : p₂ ≫ π₂ = q) (v₁ : s ≫ p₁ = e) (v₂ : s ≫ p₂ = e₂) {α β : conormal s ⟶ F} (h₁ : map hs w₁ v₁ ≫ α = map hs w₁ v₁ ≫ β) (h₂ : map hs w₂ v₂ ≫ α = map hs w₂ v₂ ≫ β) : α = β

theorem **IsChart.exists_isBaseChange** [QuasiCompact e] (c : IsChart π e U W) {R A : Type} [CommRing R] [CommRing A] [Algebra R A] (ε : A →ₐ[R] R) [Algebra R Γ(S, U)] [Module R Γ(conormal e, U)] [IsScalarTower R Γ(S, U) Γ(conormal e, U)] (α : A →+* Γ(X, W)) (po : IsPushout (CommRingCat.ofHom (algebraMap R A)) (CommRingCat.ofHom (algebraMap R Γ(S, U))) (CommRingCat.ofHom α) (π.appLE U W c.le_base)) (hα : ∀ a, e.appLE W U c.le (α a) = algebraMap R Γ(S, U) (ε a)) : ∃ l : AugCot ε →ₗ[R] Γ(conormal e, U), IsBaseChange Γ(S, U) l ∧ ∀ a, l (AugCot.d ε a) = d he c.le (α a)

theorem **IsChart.isFrame_d** [QuasiCompact e] (c : IsChart π e U W) {R A ι : Type} [Fintype ι] [CommRing R] [CommRing A] [Algebra R A] (ε : A →ₐ[R] R) (ρ : R →+* Γ(S, U)) (α : A →+* Γ(X, W)) (po : IsPushout (CommRingCat.ofHom (algebraMap R A)) (CommRingCat.ofHom ρ) (CommRingCat.ofHom α) (π.appLE U W c.le_base)) (hα : ∀ a, e.appLE W U c.le (α a) = ρ (ε a)) (b : Module.Basis ι R (AugCot ε)) (a : ι → A) (hb : ∀ i, b i = AugCot.d ε (a i)) : IsFrame fun i => d he c.le (α (a i))

theorem **isIso_baseMap** {π' : X' ⟶ S'} {e' : S' ⟶ X'} {φ : X' ⟶ X} {f : S' ⟶ S} [QuasiCompact e] [QuasiCompact e'] (he' : e' ≫ π' = 𝟙 S') (H : IsPullback φ π' π f) (we : e' ≫ φ = f ≫ e) : IsIso (baseMap he' H.w we)

theorem **exists_isFrame** [QuasiCompact e] [Smooth π] (x : S) : ∃ (U : S.Opens) (n : ℕ) (s : Fin n → Γ(conormal e, U)), x ∈ U ∧ IsFrame s

theorem **isVB_conormal** [QuasiCompact e] (n : ℕ) [h : SmoothOfRelativeDimension n π] : IsVB (conormal e) n

theorem **exists_isVB_conormal** [QuasiCompact e] [Smooth π] [PreconnectedSpace S] : ∃ n, IsVB (conormal e) n

theorem **isZero_conormal_id** : IsZero (conormal (𝟙 S))

Development.AbelianVarietyTheory.Definitions.Divisors.IsAlgAtlasAtlasOf

theorem **ManIs.κ_eq** {e : ℕ} {M : Man e} (I : ManIs M Y) : M.κ = (Module.finrank ℚ F : ℝ)⁻¹

theorem **isAlgAtlas_atlasOf** (σ : F →+* ℂ) : IsAlgAtlas (atlasOf Y σ)

theorem **isBd_fsPot** (σ : F →+* ℂ) : IsBd (atlasOf Y σ) ∅ (fun _ => Real.exp 1) (fsPot Y σ)

theorem **exists_manIs** (Y : Var F) : ∃ M : Man Y.dim, Nonempty (ManIs M Y)

theorem **generatesAt_genAt** (Δ : Y.Dv) (x : Y.X) : Δ.GeneratesAt x (genAt Δ x)

theorem **exists_pn_eq** {f : Y.Fn} {y : Pts Y σ} (hf : Δ.GeneratesAt (under y) f) : ∃ a : ℝ, 0 < a ∧ ∀ m : C.Met Y σ Δ, C.pn Y σ Δ m (C.cx Y σ f) y = a * C.pn Y σ Δ m (C.cx Y σ (genAt Δ (under y))) y

theorem **lnRatio_eq** {m m' : C.Met Y σ Δ} {f : Y.Fn} {y : Pts Y σ} (hf : Δ.GeneratesAt (under y) f) (h : PosAt m y) (h' : PosAt m' y) : lnRatio m m' y = lnN m f y − lnN m' f y

theorem **isPot_ofWeight** {L : LB Y.X} {r : L.ι → Y.Fnˣ} (h : Presents Δ L r) (φ : Weight Y.str σ L) : IsPot id (C.ofWeight Y σ Δ L r h φ) φ.locPot

theorem **formOf_ofWeight** {Z : ∀ σ, Set (Pts Y σ)} {r : ∀ σ, Pts Y σ → ℝ} {Om : ∀ σ, LocPot (Pts Y σ)} (h : ∀ σ, IsBd (atlasOf Y σ) (Z σ) (r σ) (Om σ)) {Δ : Y.Dv} {L : LB Y.X} {g : L.ι → Y.Fnˣ} (hL : Presents Δ L g) (φ : ∀ σ, Weight Y.str σ L) (hφ : ∀ σ, (φ σ).IsCInf) : (manIsOf Z r Om h).FormOf (fun (_ : Unit) σ => C.ofWeight Y σ Δ L g hL (φ σ)) (fun _ => 1) fun _ => (φ _).locPot

theorem **deg_pos** : 0 < V.deg

theorem **ν_pos** : 0 < V.ν

theorem **deg_ofModel** (M : TrueGeo.Model Y) (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme M.X) : (ofModel M h).deg = 1

theorem **induces_ofModel** {A : Adl C Y} {M : TrueGeo.Model Y} (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme M.X) {L : HermLB M.p} {k : ℕ+} (hL : A.Induced M L k 0) : (ofModel M h).Induces A L k

theorem **exists_isDown** {e : ℕ} {M : Man e} (I : ManIs M Y) : ∃ ε π, V.IsDown I ε π

Development.AbelianVarietyTheory.Definitions.Divisors.IsAlgLowZero

theorem **coIpStd_zero** {J : Type} [Fintype J] (e : J → P) (H : M →ₗ*[ℂ] M →ₗ[ℂ] ℂ) (f g : P [∧^Fin 0]→ₗ[R] M) : coIpStd e H f g = coIp e H f g

theorem **gd_eq_gdR** {P M : Type} [AddCommGroup P] [Module ℝ P] [AddCommGroup M] [Module ℂ M] [Module ℝ M] [IsScalarTower ℝ ℂ M] (L : P →ₗ[ℝ] Module.End ℂ M) {q : ℕ} (f : P [∧^Fin q]→ₗ[ℝ] M) : GlobalTheory.BorelCoh.gd L f = gdR L f

theorem **codOp_eq** {ε : T → Fin n → Bool} (E : T → Fin n → Fin n → Module.End ℂ M) (ξ : T → Fin n → Fin n → Matrix (Fin d) (Fin d) ℂ) : codOp (ε

theorem **trOp_apply** {ε : T → Fin n → Bool} (E : T → Fin n → Fin n → Module.End ℂ M) (x : NC ε) : trOp E x = E x.1.1 x.1.2.2 x.1.2.1

theorem **isAlgLow_zero** {T : Type} [Fintype T] [DecidableEq T] (n : ℕ) : IsAlgLow (T

Development.AbelianVarietyTheory.Definitions.Divisors.IsAmbOfIsArithVar

theorem **isAmb_of_isArithVar** {p : X ⟶ Sp k} {D : ℕ} (h : IsArithVar p D) : IsAmb p

theorem **IsAmb.sub** {p : X ⟶ Sp k} (h : IsAmb p) (i : Y ⟶ X) [IsClosedImmersion i] : IsAmb (i ≫ p)

theorem **IsAmb.isNoetherian** {p : X ⟶ Sp k} (h : IsAmb p) : IsNoetherian X

theorem **cloι_genericPoint** (x : X) : cloι x (genericPoint (clo x)) = x

theorem **isMix_zero** (p : X ⟶ Sp k) : IsMix p 0 fun _ _ x => Real.log (Nat.card (X.residueField x)) where

Development.AbelianVarietyTheory.Definitions.Divisors.IsBasOpens

theorem **opOf_le** (G : C ⥤ AffOp X) ⦃T T' : C⦄ (a : T ⟶ T') : opOf G T' ≤ opOf G T

theorem **isCoc_one** (V : C → X.Opens) (hV : ∀ ⦃T T' : C⦄, (T ⟶ T') → V T' ≤ V T) : IsCoc V hV fun _ _ _ ↦ 1

theorem **isBas_opens** : IsBas (C

theorem **isObU_one** (L : Pre I π F) (α : Aug I π R₁) (σ : ∀ T, R₁ ⊗[R₀] F.obj T →+ L.B T) (hσ : ∀ T, α.IsSig (L.p T) (σ T)) : IsObU L α (fun _ _ _ ↦ 1) 0

Development.AbelianVarietyTheory.Definitions.Divisors.IsChartMemberP

theorem **pre_toVSys** (N₁ : ℕ) (hl : Φ.abelian₂) (hf : Φ.Four N₁) : (Φ.toVSys N₁ hl hf).pre = Φ

theorem **VSys.pre_lam** {N₁ : ℕ} (Φ : VSys bar q W N₁ A) : Φ.pre.abelian₂

theorem **VSys.pre_four** {N₁ : ℕ} (Φ : VSys bar q W N₁ A) : Φ.pre.Four N₁

theorem **IsThetaTerms.zero** {D : DegPre V₁} {N₁ : ℕ} {X C : Type} [AddCommGroup X] {e : X ≃+ D.X} {T : X → C → V₁} (h : IsThetaTerms D N₁ e T) : IsThetaTerms₀ D N₁ e T

theorem **HasTraits.reg** {N₁ : ℕ} {Ψ : VSys (id : ℤ → ℤ) g W N₁ (X₁.M.Z.presheaf.stalk z₁)} (h : HasTraits X₁ z₁ Ψ) : HasTraitsR X₁ z₁ Ψ.pre

Development.AbelianVarietyTheory.Definitions.Divisors.IsCompactIsOpenUnitsAt

theorem **at_partFin** (v : Pl F) (z : M) : At (evFin F v) (partFin v z)

theorem **partFin_of_at** {v : Pl F} {z : M} (hz : At (evFin F v) z) : partFin v z = z

theorem **evI_realAd_one** : evI F (realAd F 1) = 1

theorem **at_partInf** (z : M) : At (evI F) (partInf F z)

theorem **partInf_of_at** {z : M} (hz : At (evI F) z) : partInf F z = z

theorem **mem_UAtSub** {X : Spc F E n} {ev : AF F → R} {g : UA σ X.J} : g ∈ UAtSub σ X ev ↔ g ∈ UAt σ X ev

theorem **mem_KAt** {X : Spc F E n} {K : Subgroup (UA σ X.J)} {ev : AF F → R} {g : UA σ X.J} : g ∈ KAt K ev ↔ g ∈ K ∧ g ∈ UAt σ X ev

theorem **mem_idelesAt** {ev : AF F → R} {l : (AE F E)ˣ} : l ∈ idelesAt E ev ↔ At ev ((l : AE F E) − 1)

theorem **mem_unitsAt** {v : Pl F} {l : (AE F E)ˣ} : l ∈ unitsAt E v ↔ l ∈ idelesAt E (evFin F v) ∧ compFin F E v l ∈ intLoc F E v ∧ compFin F E v (l⁻¹ : (AE F E)ˣ) ∈ intLoc F E v

theorem **unitsAt_le** (v : Pl F) : unitsAt E v ≤ idelesAt E (evFin F v)

theorem **isClosed_UAtSub** (X : Spc F E n) (ev : AF F → R) : IsClosed (UAtSub σ X ev : Set (UA σ X.J))

theorem **isCompact_KAt** {X : Spc F E n} {K : Subgroup (UA σ X.J)} (hK : IsCompact (K : Set (UA σ X.J))) (ev : AF F → R) : IsCompact (KAt K ev : Set (UA σ X.J))

theorem **isClosed_idelesAt** (ev : AF F → R) : IsClosed (idelesAt E ev : Set (AE F E)ˣ)

theorem **compFin_smul** (v : Pl F) (a : AF F) (z : AE F E) : compFin F E v (a • z) = evFin F v a • compFin F E v z

theorem **compInf_smul** (w : InfinitePlace F) (a : AF F) (z : AE F E) : compInf F E w (a • z) = evInf F w a • compInf F E w z

theorem **at_evFin_iff** {v : Pl F} {z : AE F E} : At (evFin F v) z ↔ (∀ w, compInf F E w z = 0) ∧ ∀ w, w ≠ v → compFin F E w z = 0

theorem **at_evI_iff** {z : AE F E} : At (evI F) z ↔ ∀ w, compFin F E w z = 0

theorem **ideleAt_mem** (v : Pl F) (x : (Fv F v ⊗[F] E)ˣ) : ideleAt F E v x ∈ idelesAt E (evFin F v)

theorem **ideleAt_unitFin** {v : Pl F} {l : (AE F E)ˣ} (hl : l ∈ idelesAt E (evFin F v)) : ideleAt F E v (unitFin F E v l) = l

theorem **absE_of_mem** {v : Pl F} {l : (AE F E)ˣ} (hl : l ∈ idelesAt E (evFin F v)) : absE F E l = absEv F E v (unitFin F E v l)

theorem **isCompact_isOpen_intLoc** (v : Pl F) : IsCompact (intLoc F E v : Set (Fv F v ⊗[F] E)) ∧ IsOpen (intLoc F E v : Set (Fv F v ⊗[F] E))

theorem **isCompact_isOpen_unitsAt** (v : Pl F) : IsCompact {l : idelesAt E (evFin F v) | l.1 ∈ unitsAt E v} ∧ IsOpen {l : idelesAt E (evFin F v) | l.1 ∈ unitsAt E v}

theorem **existsUnique_isNormK** {K : Subgroup (UA σ X.J)} (hK : IsCompact (K : Set (UA σ X.J))) (ev : AF F → R) : ∃! μ, IsNormK K ev μ

theorem **idelesAt_facts** (ev : AF F → R) : IsTopologicalGroup (idelesAt E ev) ∧ LocallyCompactSpace (idelesAt E ev) ∧ SecondCountableTopology (idelesAt E ev) ∧ T2Space (idelesAt E ev) ∧ BorelSpace (idelesAt E ev)

theorem **exists_haar_idelesAt** (ev : AF F → R) : ∃ μ : Measure (idelesAt E ev), μ.IsHaarMeasure

theorem **haar_idelesAt_eq_smul** {ev : AF F → R} (μ' μ : Measure (idelesAt E ev)) [μ'.IsHaarMeasure] [μ.IsHaarMeasure] : ∃ c : ℝ≥0, c ≠ 0 ∧ μ' = c • μ

theorem **existsUnique_isNormE** (v : Pl F) : ∃! μ, IsNormE E v μ

theorem **absE_of_mem_unitsAt** {v : Pl F} {l : (AE F E)ˣ} (hl : l ∈ unitsAt E v) : absE F E l = 1

theorem **locOp_mem** (v : Pl F) (T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)) (φ : (κ → Fv F v) → ℂ) : locOp v T φ ∈ SB (κ → Fv F v)

theorem **acts_locOp** {v : Pl F} {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} {φ : (κ → Fv F v) → ℂ} (h : ∃ φ₁, φ₁ ∈ SB (κ → Fv F v) ∧ Acts (evFin F v) T φ φ₁) : Acts (evFin F v) T φ (locOp v T φ)

theorem **locOp_eq** {v : Pl F} {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} {φ φ₁ : (κ → Fv F v) → ℂ} (hφ : φ ∈ SB (κ → Fv F v)) (hφ₁ : φ₁ ∈ SB (κ → Fv F v)) (h : Acts (evFin F v) T φ φ₁) : locOp v T φ = φ₁

theorem **acts_locOpInf** {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} {f : SInf F κ} (h : ∃ f₁ : SInf F κ, Acts (evI F) T f f₁) : Acts (evI F) T f (locOpInf T f)

theorem **exists_rest_inf** : ∃ r : ((κ → AF F) → ℂ), r 0 ≠ 0 ∧ ∀ f : SInf F κ, Rest (evI F) f r

theorem **eq_of_tens_inf** {r : ((κ → AF F) → ℂ)} (h0 : r 0 ≠ 0) (hr : ∀ x z, At (evI F) z → r (x + z) = r x) {f g : (κ → mixedEmbedding.mixedSpace F) → ℂ} (h : tens (evI F) f r = tens (evI F) g r) : f = g

theorem **locOpInf_eq** {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} {f f₁ : SInf F κ} (h : Acts (evI F) T f f₁) : locOpInf T f = f₁

theorem **tens_lin** {R : Type} (ev : AF F → R) (a b : ℂ) (φ φ' : (κ → R) → ℂ) (r : ((κ → AF F) → ℂ)) : tens ev (a • φ + b • φ') r = a • tens ev φ r + b • tens ev φ' r

theorem **locOp_lin** {v : Pl F} {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} (hT : IsLinearMap ℂ T) (hex : ∀ φ ∈ SB (κ → Fv F v), ∃ φ₁, φ₁ ∈ SB (κ → Fv F v) ∧ Acts (evFin F v) T φ φ₁) (a b : ℂ) {φ φ' : (κ → Fv F v) → ℂ} (hφ : φ ∈ SB (κ → Fv F v)) (hφ' : φ' ∈ SB (κ → Fv F v)) : locOp v T (a • φ + b • φ') = a • locOp v T φ + b • locOp v T φ'

theorem **locOpInf_lin** {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} (hT : IsLinearMap ℂ T) (hex : ∀ f : SInf F κ, ∃ f₁ : SInf F κ, Acts (evI F) T f f₁) (a b : ℂ) (f f' : SInf F κ) : locOpInf T (a • f + b • f') = a • locOpInf T f + b • locOpInf T f'

theorem **locOp_of_fix** {v : Pl F} {T : ((κ → AF F) → ℂ) → ((κ → AF F) → ℂ)} (hT : ∀ Φ ∈ schA F κ, T Φ = Φ) {φ : (κ → Fv F v) → ℂ} (hφ : φ ∈ SB (κ → Fv F v)) : locOp v T φ = φ

theorem **locOp_rho** (co : Darboux Bf κ) {ψ : AddChar (AF F) ℂ} (hψ : Continuous ψ) (v : Pl F) (u : V) {φ : (κ → Fv F v) → ℂ} (hφ : φ ∈ SB (κ → Fv F v)) : Acts (evFin F v) (co.rho ψ (partFin v u)) φ (locOp v (co.rho ψ (partFin v u)) φ) ∧ locOp v (co.rho ψ (partFin v u)) φ = schr (GlobalTheory.LatVac.psiAt ψ v) (GlobalTheory.LatVac.locCo co v u) φ

theorem **locOpInf_rho** (co : Darboux Bf κ) {ψ : AddChar (AF F) ℂ} (hψ : IsStd ψ) (u : V) (f : SInf F κ) : Acts (evI F) (co.rho ψ (partInf F u)) f (locOpInf (co.rho ψ (partInf F u)) f) ∧ ⇑(locOpInf (co.rho ψ (partInf F u)) f) = schrF (trP F κ) (locV (co.e u).1, locV (co.e u).2) f

theorem **smul_shiftA** (H : Plane E n n₁) (H' : Plane E d d₁) (a : AF F) (l : (AE F E)ˣ) : a • shiftA H H' l = (a • l.1) • Matrix.vecMulVec (incV F H.x) (incV F H'.y)

theorem **smul_shiftB** (H : Plane E n n₁) (H' : Plane E d d₁) (a : AF F) (l : (AE F E)ˣ) : a • shiftB σ H H' l = cA σ (a • l.1) • Matrix.vecMulVec (incV F H.y) (incV F H'.x)

theorem **acts_locOp_om** (hA : ActsEx) {B : Base F E} {M : Model σ X W} {χX χW : Chr F E} {om : Rep σ X W} {r s r' s' : ℕ} (hP : IsPair B M χX χW om r s r' s') {v : Pl F} {g : UA σ X.J} (hg : g ∈ UAt σ X (evFin F v)) {h : UA σ W.J} (hh : h ∈ UAt σ W (evFin F v)) {φ : (Fin n × Fin d → Fv F v) → ℂ} (hφ : φ ∈ SB (Fin n × Fin d → Fv F v)) : Acts (evFin F v) (om (g, h)) φ (locOp v (om (g, h)) φ)

theorem **coneFinInt_eq** (hψ : Continuous B.ψ) (φ : (Fin n × Fin d → Fv F v) → ℂ) (k : UA σ X.J) {l : (AE F E)ˣ} (hl : l ∈ idelesAt E (evFin F v)) (y : Fin n₁ × Fin d₁ → Fv F v) : coneFinInt B M om v ev u ν z φ k l y = ev (schr (GlobalTheory.LatVac.psiAt B.ψ v) (GlobalTheory.LatVac.locCo M.co v (u l)) (locOp v (om (k, 1)) φ)) y ∗ (ν l : ℂ) ∗ ((absEv F E v (unitFin F E v l) : ℝ) : ℂ) ^ z

theorem **coneFinInt_lin** (hψ : Continuous B.ψ) {k : UA σ X.J} (hk : ∀ φ ∈ SB (Fin n × Fin d → Fv F v), ∃ φ₁, φ₁ ∈ SB (Fin n × Fin d → Fv F v) ∧ Acts (evFin F v) (om (k, 1)) φ φ₁) (a b : ℂ) {φ φ' : (Fin n × Fin d → Fv F v) → ℂ} (hφ : φ ∈ SB (Fin n × Fin d → Fv F v)) (hφ' : φ' ∈ SB (Fin n × Fin d → Fv F v)) (l : (AE F E)ˣ) (y : Fin n₁ × Fin d₁ → Fv F v) : coneFinInt B M om v ev u ν z (a • φ + b • φ') k l y = a ∗ coneFinInt B M om v ev u ν z φ k l y + b ∗ coneFinInt B M om v ev u ν z φ' k l y

theorem **coneFin_lin** (hψ : Continuous B.ψ) (hk : ∀ k ∈ KAt K (evFin F v), ∀ φ ∈ SB (Fin n × Fin d → Fv F v), ∃ φ₁, φ₁ ∈ SB (Fin n × Fin d → Fv F v) ∧ Acts (evFin F v) (om (k, 1)) φ φ₁) (a b : ℂ) {φ φ' : (Fin n × Fin d → Fv F v) → ℂ} (hφ : φ ∈ SB (Fin n × Fin d → Fv F v)) (hφ' : φ' ∈ SB (Fin n × Fin d → Fv F v)) (y : Fin n₁ × Fin d₁ → Fv F v) (h1 : ∀ k : KAt K (evFin F v), Integrable (fun l : idelesAt E (evFin F v) => coneFinInt B M om v ev u ν z φ k.1 l.1 y) μ) (h1' : ∀ k : KAt K (evFin F v), Integrable (fun l : idelesAt E (evFin F v) => coneFinInt B M om v ev u ν z φ' k.1 l.1 y) μ) (h2 : Integrable (fun k : KAt K (evFin F v) => ∫ l, coneFinInt B M om v ev u ν z φ k.1 l.1 y ∂μ) μK) (h2' : Integrable (fun k : KAt K (evFin F v) => ∫ l, coneFinInt B M om v ev u ν z φ' k.1 l.1 y ∂μ) μK) : coneFin B M om v ev K μK μ u ν z (a • φ + b • φ') y = a ∗ coneFin B M om v ev K μK μ u ν z φ y + b ∗ cone

theorem **coneFin_of_fixed** (hK : IsNormK K (evFin F v) μK) {φ : (Fin n × Fin d → Fv F v) → ℂ} (hφ : φ ∈ SB (Fin n × Fin d → Fv F v)) (hfix : ∀ k ∈ KAt K (evFin F v), Acts (evFin F v) (om (k, 1)) φ φ) (y : Fin n₁ × Fin d₁ → Fv F v) : coneFin B M om v ev K μK μ u ν z φ y = ∫ l, ev (locOp v (M.co.rho B.ψ (partFin v (u l.1))) φ) y ∗ (ν l.1 : ℂ) ∗ ((absE F E l.1 : ℝ) : ℂ) ^ z ∂μ

theorem **coneInfInt_apply** (f : SInf F (Fin n × Fin d)) (k : UA σ X.J) (l : (AE F E)ˣ) (y : Fin n₁ × Fin d₁ → mixedEmbedding.mixedSpace F) : coneInfInt B M om ei u ν z f k l y = ei (locOpInf (M.co.rho B.ψ (partInf F (u l))) (locOpInf (om (k, 1)) f)) y ∗ (ν l : ℂ) ∗ ((absE F E l : ℝ) : ℂ) ^ z

theorem **coe_coneInf** {f : SInf F (Fin n × Fin d)} (h : ∃ g : SInf F (Fin n₁ × Fin d₁), ⇑g = coneInfFun B M om ei K μK μ u ν z f) : ⇑(coneInf B M om ei K μK μ u ν z f) = coneInfFun B M om ei K μK μ u ν z f

theorem **coneInf_eq** {f : SInf F (Fin n × Fin d)} {g : SInf F (Fin n₁ × Fin d₁)} (h : ⇑g = coneInfFun B M om ei K μK μ u ν z f) : coneInf B M om ei K μK μ u ν z f = g

theorem **coneInfInt_lin** (hψ : IsStd B.ψ) {k : UA σ X.J} (hk : ∀ f : SInf F (Fin n × Fin d), ∃ f₁ : SInf F (Fin n × Fin d), Acts (evI F) (om (k, 1)) f f₁) (a b : ℂ) (f f' : SInf F (Fin n × Fin d)) (l : (AE F E)ˣ) : coneInfInt B M om ei u ν z (a • f + b • f') k l = a • coneInfInt B M om ei u ν z f k l + b • coneInfInt B M om ei u ν z f' k l

theorem **mem_KAt_inf** {B : Base F E} {K : Subgroup (UA σ X.J)} (hK : IsStab B K) {k : UA σ X.J} : k ∈ KAt K (evI F) ↔ k ∈ KInf σ X B.emb

Development.AbelianVarietyTheory.Definitions.Divisors.IsCoprimeOfNatCast

theorem **mem_sup_span_mul** {O : Type} [CommRing O] {I : Ideal O} {a b x : O} (h : IsCoprime a b) (ha : x ∈ I ⊔ Ideal.span {a}) (hb : x ∈ I ⊔ Ideal.span {b}) : x ∈ I ⊔ Ideal.span {a * b}

theorem **absNorm_eq_of_split** {E : Type} [Field E] [NumberField E] {p : ℕ} (hp : p.Prime) (h : {v : HeightOneSpectrum (𝓞 E) | ((p : ℕ) : 𝓞 E) ∈ v.asIdeal}.ncard = Module.finrank ℚ E) {P : Ideal (𝓞 E)} (hP : P.IsPrime) (hw : (p : 𝓞 E) ∈ P) : Ideal.absNorm P = p

theorem **natCast_M_mem** (l : Level H) : (l.M : 𝓞 E) ∈ l.Mb

theorem **bar_bar'** (a : 𝓞 E) : H.bar (H.bar a) = a

theorem **bar_algebraMap** (a : 𝓞 (maximalRealSubfield E)) : H.bar (algebraMap _ (𝓞 E) a) = algebraMap _ (𝓞 E) a

theorem **exists_norm** (y : 𝓞 E) : ∃ z : 𝓞 (maximalRealSubfield E), algebraMap _ (𝓞 E) z = y * H.bar y

theorem **halfPrime_eq** (l : Level H) {v : Ideal (𝓞 (maximalRealSubfield E))} (hv : v.IsMaximal) {P : Ideal (𝓞 E)} (hP : P.IsPrime) (h : halfPrime l v ≤ P) : halfPrime l v = P

theorem **halfPrime_isPrime** (l : Level H) {v : Ideal (𝓞 (maximalRealSubfield E))} (hv : v.IsMaximal) (hM : (l.M : 𝓞 (maximalRealSubfield E)) ∈ v) : (halfPrime l v).IsPrime

theorem **superDiag_mul** {r : ℕ} {k : Type} [Field k] {u u' : GL (Fin r) k} (hu : u ∈ unitri r k) (hu' : u' ∈ unitri r k) : superDiag ((u * u' : GL (Fin r) k) : Matrix (Fin r) (Fin r) k) = superDiag (u : Matrix (Fin r) (Fin r) k) + superDiag (u' : Matrix (Fin r) (Fin r) k)

theorem **genChar_apply** (ψ : ∀ v : Sq Hs, AddChar (ZMod (Hs.g v.1)) K) (u : levelU Hs) : ((genChar ψ u : Kˣ) : K) = psiU ψ (u : LevelGroup Hs)

theorem **exists_factor** (a : LevelGroup Hs) : ∃ b ∈ factorG Hs 0, ∃ c ∈ factorG Hs 1, a = b * c ∧ (∀ v, v.1 = 0 → b v = a v) ∧ ∀ v, v.1 = 1 → c v = a v

theorem **isTypeLevel_of_eq** {m : ℕ} (hA : l.A = 1) (hM : l.M = D.host.g 0 * D.host.g 1 * m) (hm : Nat.Coprime (D.host.g 0 * D.host.g 1) m) : IsTypeLevel l ∧ cofactor l = m

theorem **isTypeLevel_of_pin** {Q : Par D} (h : Q.PinLevel) : IsTypeLevel Q.l ∧ cofactor Q.l = 1

theorem **g_mem_typePrime** (v : Sq D.host) : ((D.host.g v.1 : ℕ) : 𝓞 D.host.E) ∈ typePrime l v

theorem **g_dvd** (i : Fin 2) : D.host.g i ∣ D.host.g 0 * D.host.g 1

theorem **g_coprime** {i j : Fin 2} (h : i ≠ j) : Nat.Coprime (D.host.g i) (D.host.g j)

theorem **isCoprime_of_natCast** {O : Type} [CommRing O] {I J : Ideal O} {a b : ℕ} (h : Nat.Coprime a b) (ha : (a : O) ∈ I) (hb : (b : O) ∈ J) : IsCoprime I J

theorem **typePrime_coprime** : Pairwise (Function.onFun IsCoprime (typePrime l))

theorem **iInf_typePrime_le** : ⊓ v, typePrime l v ≤ l.Mb ⊔ Ideal.span {((D.host.g 0 * D.host.g 1 : ℕ) : 𝓞 D.host.E)}

theorem **IsTypeLevel.typePrime_spec** (hl : IsTypeLevel l) (v : Sq D.host) : (typePrime l v).IsPrime ∧ Ideal.absNorm (typePrime l v) = D.host.g v.1

theorem **IsTypeLevel.smul_eq_zsmul** (hl : IsTypeLevel l) (v : Sq D.host) (a : 𝓞 D.host.E) : ∃ n : ℤ, ∀ x : QMod Dw.La (typePrime l v), a • x = n • x

theorem **nonempty_addEquiv_pi** {M : Type} [AddCommGroup M] {p r : ℕ} [hp : Fact p.Prime] (h : ∀ x : M, p • x = 0) (hN : Nat.card M = p ^ r) : Nonempty (M ≃+ (Fin r → ZMod p))

theorem **IsTypeLevel.nonempty_typeFrame** (hl : IsTypeLevel l) (hr : Dw.p + Dw.q = D.R.r) : Nonempty (TypeFrame Dw l)

theorem **typeRed_mulVec** (fr : TypeFrame Dw l) (γ : LevelAut Dw l) (v : Sq D.host) (q : LaMod Dw l) : (typeRed fr γ v : Matrix (Fin D.R.r) (Fin D.R.r) (ZMod (D.host.g v.1))).mulVec (fr v (qProj (J

theorem **mem_grpR_iff** {γ : LevelAut Dw l} : γ ∈ grpR Dw l ↔ ∀ v, γ ∈ trivMod Dw.La l.Mb (typePrime l v)

theorem **typeRed_eq_one** (fr : TypeFrame Dw l) {γ : LevelAut Dw l} : typeRed fr γ = 1 ↔ γ ∈ grpR Dw l

theorem **le_partIdeal** : ∀ o, l.Mb ≤ partIdeal l o ∣ none => le_sup_left ∣ some _ => le_sup_left theorem IsTypeLevel.partIdeal_coprime (hl : IsTypeLevel l) : Pairwise (Function.onFun IsCoprime (partIdeal l))

theorem **IsTypeLevel.iInf_partIdeal_le** (hl : IsTypeLevel l) : ⊓ o, partIdeal l o ≤ l.Mb

theorem **IsTypeLevel.grpG_inf_grpR** (hl : IsTypeLevel l) : grpG Dw l ⊓ grpR Dw l = ⊥

theorem **IsTypeLevel.typeRed_bijective** (hl : IsTypeLevel l) (fr : TypeFrame Dw l) : Function.Bijective ((typeRed fr).comp (grpG Dw l).subtype)

theorem **typeRed_typeEmb** (hl : IsTypeLevel l) (fr : TypeFrame Dw l) (a : LevelGroup D.host) : typeRed fr (typeEmb hl fr a) = a

theorem **typeEmb_mem** (hl : IsTypeLevel l) (fr : TypeFrame Dw l) (a : LevelGroup D.host) : typeEmb hl fr a ∈ grpG Dw l

theorem **IsTypeLevel.commute** (hl : IsTypeLevel l) {γ δ : LevelAut Dw l} (hγ : γ ∈ grpG Dw l) (hδ : δ ∈ grpR Dw l) : Commute γ δ

theorem **prodEquiv_apply** (hl : IsTypeLevel l) (fr : TypeFrame Dw l) (a : LevelGroup D.host) (b : grpR Dw l) : prodEquiv hl fr (a, b) = typeEmb hl fr a * b

theorem **typeRed_prodEquiv** (hl : IsTypeLevel l) (fr : TypeFrame Dw l) (x : LevelGroup D.host × grpR Dw l) : typeRed fr (prodEquiv hl fr x) = x.1

theorem **chiHR_apply** (fr : TypeFrame Dw l) (ψ : ∀ v : Sq D.host, AddChar (ZMod (D.host.g v.1)) K) (h : grpHR fr) : ((chiHR fr ψ h : Kˣ) : K) = psiU ψ (typeRed fr h)

Development.AbelianVarietyTheory.Definitions.Divisors.IsCorrOpOfAvg

theorem **pullTot_id** (E : Spc) : pullTot l (𝟙 E) = LinearMap.id

theorem **pullTot_comp** {E E' E'' : Spc} (f : E ⟶ E') (g : E' ⟶ E'') : pullTot l (f ≫ g) = pullTot l f ∘ₗ pullTot l g

private theorem **galMap_one'** {k : Type} [Field k] (Ω : Type) [Field Ω] [Algebra k Ω] {X : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) : GlobalTheory.Et.galMap Ω p (1 : Ω ≃ₐ[k] Ω) = 𝟙 _

theorem **galTot_one** {k : Type} [Field k] (Ω : Type) [Field Ω] [Algebra k Ω] {X : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) : galTot l Ω p 1 = LinearMap.id

private theorem **galMap_mul'** {k : Type} [Field k] (Ω : Type) [Field Ω] [Algebra k Ω] {X : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) (σ τ : Ω ≃ₐ[k] Ω) : GlobalTheory.Et.galMap Ω p (σ * τ) = GlobalTheory.Et.galMap Ω p σ ≫ GlobalTheory.Et.galMap Ω p τ

theorem **galTot_mul** {k : Type} [Field k] (Ω : Type) [Field Ω] [Algebra k Ω] {X : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) (σ τ : Ω ≃ₐ[k] Ω) : galTot l Ω p (σ * τ) = galTot l Ω p σ ∘ₗ galTot l Ω p τ

theorem **isGalois_id** (X : Scheme.{0}) : IsGalois (𝟙 X) Unit (fun _ => 𝟙 X)

theorem **splits_fold** (X : Scheme.{0}) (J : Type) : Splits (Sigma.desc fun _ : J => 𝟙 X) (𝟙 X) (Sigma.ι (fun _ : J => X))

theorem **isCorrOp_fold** {X Z' : Scheme.{0}} {J : Type} [Fintype J] (f : J → (X ⟶ Z')) : IsCorrOp l (Sigma.desc fun _ : J => 𝟙 X) (Sigma.desc f) (∑ j, pullTot l (sch.map (f j)))

Development.AbelianVarietyTheory.Definitions.Divisors.IsCuspSpaceBot

theorem **toGL_val** {n : ℕ} (g : GL (Fin n) R) : toGL (g : Matrix (Fin n) (Fin n) R) = g

Development.AbelianVarietyTheory.Definitions.Divisors.IsCyclotomicExtensionQI

theorem **mem_simGroup** {j : A →+* S} {σ : S →+* S} {H : Matrix (Fin n) (Fin n) S} {x : GL (Fin n) S × Aˣ} : x ∈ simGroup j σ H ↔ (x.1 : Matrix (Fin n) (Fin n) S)ᵀ * H * (x.1 : Matrix (Fin n) (Fin n) S).map σ = j (x.2 : A) • H

@[simp] theorem **simMap_fst** {j : A →+* S} {σ : S →+* S} {H : Matrix (Fin n) (Fin n) S} {j' : A' →+* S'} {σ' : S' →+* S'} {H' : Matrix (Fin n) (Fin n) S'} (φ : S →+* S') (ψ : A →+* A') (hσ : ∀ x, φ (σ x) = σ' (φ x)) (hj : ∀ a, φ (j a) = j' (ψ a)) (hH : H.map φ = H') (x : simGroup j σ H) : ((((simMap φ ψ hσ hj hH x : simGroup j' σ' H') : GL (Fin n) S' × A'ˣ).1 : GL (Fin n) S') : Matrix (Fin n) (Fin n) S') = (((x : GL (Fin n) S × Aˣ).1 : GL (Fin n) S) : Matrix (Fin n) (Fin n) S).map φ

@[simp] theorem **simMap_snd** {j : A →+* S} {σ : S →+* S} {H : Matrix (Fin n) (Fin n) S} {j' : A' →+* S'} {σ' : S' →+* S'} {H' : Matrix (Fin n) (Fin n) S'} (φ : S →+* S') (ψ : A →+* A') (hσ : ∀ x, φ (σ x) = σ' (φ x)) (hj : ∀ a, φ (j a) = j' (ψ a)) (hH : H.map φ = H') (x : simGroup j σ H) : ((((simMap φ ψ hσ hj hH x : simGroup j' σ' H') : GL (Fin n) S' × A'ˣ).2 : A'ˣ) : A') = ψ (((x : GL (Fin n) S × Aˣ).2 : Aˣ) : A)

@[simp] theorem **conjQ_apply** (x : F) : conjQ F x = IsCMField.complexConj F x

@[simp] theorem **conjT_tmul** (R : Type) [CommRing R] [Algebra ℚ R] (r : R) (x : F) : conjT F R (r ⊗ₜ x) = r ⊗ₜ IsCMField.complexConj F x

theorem **simR_bc** (R : Type) [CommRing R] [Algebra ℚ R] (g : GQpts F H) : ((simR R (bc R g) : Rˣ) : R) = algebraMap ℚ R (simQ g : ℚ)

theorem **isHermitian_gramAt** (hH : (H.map (IsCMField.complexConj F : F →+* F))ᵀ = H) (τ : F →+* ℂ) : (gramAt H τ).IsHermitian

theorem **sigPos_eq_sigPosH** {F : Type} [Field F] [NumberField F] [IsCMField F] {n : ℕ} {H : Matrix (Fin n) (Fin n) F} (hH : (H.map (IsCMField.complexConj F : F →+* F))ᵀ = H) (τ : F →+* ℂ) : sigPos hH τ = sigPosH (isHermitian_gramAt hH τ)

theorem **mem_uGroup** {σ : S →+* Sᵐᵒᵖ} {x : Sˣ × Aˣ} : x ∈ uGroup σ ↔ ((x.1 : Sˣ) : S) * (σ ((x.1 : Sˣ) : S)).unop = algebraMap A S ((x.2 : Aˣ) : A)

@[simp] theorem **uMap_fst** {σ : S →+* Sᵐᵒᵖ} {σ' : S' →+* S'ᵐᵒᵖ} (φ : S →+* S') (ψ : A →+* A') (hσ : ∀ x, (σ' (φ x)).unop = φ (σ x).unop) (hj : ∀ a, φ (algebraMap A S a) = algebraMap A' S' (ψ a)) (x : uGroup (A

@[simp] theorem **uMap_snd** {σ : S →+* Sᵐᵒᵖ} {σ' : S' →+* S'ᵐᵒᵖ} (φ : S →+* S') (ψ : A →+* A') (hσ : ∀ x, (σ' (φ x)).unop = φ (σ x).unop) (hj : ∀ a, φ (algebraMap A S a) = algebraMap A' S' (ψ a)) (x : uGroup (A

theorem **starT_tmul** (R : Type) [CommRing R] [Algebra ℚ R] (r : R) (x : C) : (starT star R (r ⊗ₜ x)).unop = r ⊗ₜ (star x).unop

theorem **uSimR_ubc** (R : Type) [CommRing R] [Algebra ℚ R] (g : UQpts star) : ((uSimR R (ubc R g) : Rˣ) : R) = algebraMap ℚ R (uSimQ g : ℚ)

theorem **ubcMap_ubc** {R R' : Type} [CommRing R] [Algebra ℚ R] [CommRing R'] [Algebra ℚ R'] (f : R →ₐ[ℚ] R') (g : UQpts star) : ubcMap f (ubc R g) = ubc R' g

@[simp] theorem **embOf_apply** (τ : F →+* PadicAlgCl p) (x : F) : ((embOf τ x : FwOf τ) : PadicAlgCl p) = τ x

theorem **bcMap_bc** {R R' : Type} [CommRing R] [Algebra ℚ R] [CommRing R'] [Algebra ℚ R'] (f : R →ₐ[ℚ] R') (g : GQpts F H) : bcMap f (bc R g) = bc R' g

theorem **toPadic_fst** (g : G.GQ) : (((G.toP g).1 : ℚ_[p]ˣ) : ℚ_[p]) = ((G.sim g : ℚ) : ℚ_[p])

theorem **toPadic_snd** (g : G.GQ) (w : G.W) : ((G.toP g).2 w : Matrix (Fin G.n) (Fin G.n) (G.Fw w)) = G.rho w (((g : G.Cˣ × ℚˣ).1 : G.Cˣ) : G.C)

theorem **stQ_refl** (γ : G.GQ) : G.StQ γ γ

theorem **stQ_symm** {γ γ' : G.GQ} (h : G.StQ γ γ') : G.StQ γ' γ

theorem **stQ_trans** {γ γ' γ'' : G.GQ} (h : G.StQ γ γ') (h' : G.StQ γ' γ'') : G.StQ γ γ''

theorem **stAf_toAf** (γ₀ : G.GQ) : G.StAf γ₀ (G.toAf γ₀)

theorem **isSS_one** : G.IsSS 1

theorem **isEllR_one** : G.IsEllR 1

theorem **sim_pos_of_isEllR** {γ : G.GR} (h : G.IsEllR γ) : 0 < (((γ : (ℝ ⊗[ℚ] G.C)ˣ × ℝˣ).2 : ℝˣ) : ℝ)

theorem **map_nonsingInv** {K L : Type*} [CommRing K] [CommRing L] (f : K →+* L) {A : Matrix (Fin n) (Fin n) K} (hA : IsUnit A.det) : (A⁻¹).map f = (A.map f)⁻¹

theorem **hermStar_apply** (H : Matrix (Fin n) (Fin n) F) (hdet : H.det ≠ 0) (x : Matrix (Fin n) (Fin n) F) : (hermStar H hdet x).unop = (H.map (IsCMField.complexConj F : F →+* F))⁻¹ * (x.map (IsCMField.complexConj F : F →+* F))ᵀ * H.map (IsCMField.complexConj F : F →+* F)

theorem **map_conj_conj** (x : Matrix (Fin n) (Fin n) F) : (x.map (IsCMField.complexConj F : F →+* F)).map (IsCMField.complexConj F : F →+* F) = x

theorem **hermStar_hermStar** {H : Matrix (Fin n) (Fin n) F} (hH : (H.map (IsCMField.complexConj F : F →+* F))ᵀ = H) (hdet : H.det ≠ 0) (x : Matrix (Fin n) (Fin n) F) : (hermStar H hdet (hermStar H hdet x).unop).unop = x

theorem **hermStar_algebraMap** (H : Matrix (Fin n) (Fin n) F) (hdet : H.det ≠ 0) (a : F) : (hermStar H hdet (algebraMap F (Matrix (Fin n) (Fin n) F) a)).unop = algebraMap F (Matrix (Fin n) (Fin n) F) (IsCMField.complexConj F a)

theorem **map_algebraMap_eq_scalar** {K L : Type*} [Field K] [CommRing L] (f : K →+* L) (a : K) : (algebraMap K (Matrix (Fin n) (Fin n) K) a).map f = Matrix.scalar (Fin n) (f a)

theorem **mul_eq_smul_one_comm** {K : Type*} [Field K] {A B : Matrix (Fin n) (Fin n) K} {c : K} (hc : c ≠ 0) : A * B = c • 1 ↔ B * A = c • 1

theorem **hermStar_mul_eq_iff** {H : Matrix (Fin n) (Fin n) F} (hH : (H.map (IsCMField.complexConj F : F →+* F))ᵀ = H) (hdet : H.det ≠ 0) {c : F} (hc : c ≠ 0) (g : Matrix (Fin n) (Fin n) F) : g * (hermStar H hdet g).unop = c • 1 ↔ gᵀ * H * g.map (IsCMField.complexConj F : F →+* F) = c • H

theorem **mem_UQpts_hermStar_iff** {H : Matrix (Fin n) (Fin n) F} (hH : (H.map (IsCMField.complexConj F : F →+* F))ᵀ = H) (hdet : H.det ≠ 0) (x : (Matrix (Fin n) (Fin n) F)ˣ × ℚˣ) : x ∈ UQpts (hermStar H hdet) ↔ ((x.1 : Matrix (Fin n) (Fin n) F))ᵀ * H * (x.1 : Matrix (Fin n) (Fin n) F).map (IsCMField.complexConj F : F →+* F) = algebraMap ℚ F (x.2 : ℚ) • H

theorem **finrank_QI** : Module.finrank ℚ QI = 2

theorem **zeta_sq** : (IsCyclotomicExtension.zeta 4 ℚ QI) ^ 2 = -1

theorem **hK_QI** : Module.finrank ℚ (⊤ : Subfield QI) = 2 ∧ ∀ τ : QI →+* ℂ, ∃ x ∈ (⊤ : Subfield QI), (τ x).im ≠ 0

theorem **exists_sqrt_neg_one** : ∃ s : ℤ_[5], s * s = -1

theorem **sqrtNegOne_sq** : sqrtNegOne ^ 2 + 1 = 0

theorem **herm_Hc** (c : ℚ) : ((Hc c).map (IsCMField.complexConj QI : QI →+* QI))ᵀ = Hc c

theorem **eig_Hc** (c : ℚ) (τ : QI →+* ℂ) (i : Fin 1) : (isHermitian_gramAt (herm_Hc c) τ).eigenvalues i = c

theorem **sig_Hc** (c : ℚ) (hc : c ≠ 0) (τ : QI →+* ℂ) : (sigPos (herm_Hc c) τ = 1 ∧ sigNeg (herm_Hc c) τ = 1 - 1) ∨ (sigPos (herm_Hc c) τ = 0 ∧ sigNeg (herm_Hc c) τ = 1)

theorem **sigPos_Hc** (c : ℚ) (τ : QI →+* ℂ) : sigPos (herm_Hc c) τ = if 0 < c then 1 else 0

theorem **card_infinitePlace_QI** : Fintype.card (InfinitePlace QI) = 1

theorem **a_ex_one** : (ex 1 one_ne_zero).a = 1

theorem **a_ex_neg_one** : (ex (-1) (by norm_num)).a = 0

Development.AbelianVarietyTheory.Definitions.Divisors.IsDeltaDelAtOwnAt

theorem **isDelta_delAt_ownAt** (C : Formalism.L2.Host.CharFrame D) (v : HeightOneSpectrum (𝒪 D.what₂.LL)) : Formalism.L2.Host.IsDelta (H

theorem **ownAt_ne_of_plc_eq** {v : HeightOneSpectrum (𝒪 D.what₂.LL)} (h : Formalism.EigenvalueLevelCeiling.plc v = D.host.pde) : ownAt D v ≠ Formalism.EigenvalueLevelCeiling.plc v

theorem **inj** (h : CoreTies L EX J R) : Function.Injective D.what₂.OtoLL

theorem **fieldAt** (h : CoreTies L EX J R) (v : HeightOneSpectrum (𝒪 D.what₂.LL)) : FieldOK0 D (ownAt D v)

theorem **delAt_mem** (h : CoreTies L EX J R) (l : ℕ) (x) : (((R.C.delAt l).1 x : 𝕔ˣ) : 𝕔) ∈ FO D

theorem **delAt_valuesIn** (h : CoreTies L EX J R) (l : ℕ) : (R.C.delAt l).ValuesIn 0 (FO D).val.toRingHom

theorem **reScal_iff** (lam : type_of% R.lam) (kap : ℝ) : CoreTies L EX J (reScal R lam kap) ↔ CoreTies L EX J R

theorem **ownPin_of** {J : CoreJ D} {r : TwoRuns J G} (h₁ : ∀ j n : ℕ, InR D j → (r.first.full.dδ j n).pd = D.host.pde) (h₂ : ∀ j n : ℕ, InR D j → (r.second.full.dδ j n).pd = D.host.pde') : OwnPin r

theorem **PageNum.mono** {L : SizeData₂ D} {Pm : ParameterChoice' D L} {Tp : ParamsT D L Pm} {x₁ x₂ : ℝ} (h : PageNum Tp x₁) (hx : x₂ ≤ x₁) : PageNum Tp x₂

Development.AbelianVarietyTheory.Definitions.Divisors.IsDominantOfSection

theorem **isDominant_of_section** {Z W : Scheme.{0}} {p : Z ⟶ W} {s : W ⟶ Z} (h : s ≫ p = 𝟙 _) : IsDominant p

theorem **Dbl.prod_dim** (Y : Var F) : topologicalKrullDim (Dbl.prod Y).Z = (Y.dim + Y.dim : ℕ)

theorem **Dbl.prod_small** (Y : Var F) (h : Y.dim ≤ 1) : (Dbl.prod Y).Small

theorem **Dbl.prod_ne** (Y : Var F) (h : 0 < Y.dim) : (Dbl.prod Y).p₁ ≠ (Dbl.prod Y).p₂

Development.AbelianVarietyTheory.Definitions.Divisors.IsEquivariantToBidual

@[reassoc (attr

theorem **isEquivariant_inv** {φ : F ≅ G} (h : IsEquivariant ρ ρ' φ.hom) : IsEquivariant ρ' ρ φ.inv

theorem **eigenι_comp_act** {c : O →+* O} (hc : Function.Involutive c) (a : O) : eigenι (ρ.comp c) σ ≫ act ρ a = σ.comp c a • eigenι (ρ.comp c) σ

@[reassoc (attr

@[reassoc (attr

theorem **smul_eq_smulHom_comp** (a : Γ(X, ⊤)) (φ : F ⟶ G) : a • φ = smulHom F a ≫ φ

theorem **dualMap_smul** (a : Γ(X, ⊤)) (φ : F ⟶ G) : dualMap (a • φ) = a • dualMap φ

@[reassoc (attr

theorem **isEquivariant_toBidual** : IsEquivariant ρ (dualAct (dualAct ρ)) (cartierDualToBidual F)

@[reassoc (attr

theorem **hom_ext_of_isFrame** {κ : Type} [Fintype κ] {e : κ → Γ(F, ⊤)} (he : IsFrame e) {φ ψ : F ⟶ G} (h : ∀ i, φ.app ⊤ (e i) = ψ.app ⊤ (e i)) : φ = ψ

theorem **isFrame_d_Spec** {R A : CommRingCat.{0}} [Algebra R A] (ε : A →ₐ[R] R) {π : Spec A ⟶ Spec R} {e : Spec R ⟶ Spec A} (he : e ≫ π = 𝟙 _) (hπ : π = Spec.map (CommRingCat.ofHom (algebraMap R A))) (hε : e = Spec.map (CommRingCat.ofHom ε.toRingHom)) {κ : Type} [Fintype κ] (b : Module.Basis κ R (AugCot ε)) (a : κ → A) (hb : ∀ i, b i = AugCot.d ε (a i)) : IsFrame fun i => d he (W

Development.AbelianVarietyTheory.Definitions.Divisors.IsEtLevZero

theorem **wSub_zero** (D : Datum p) (w0 : Ideal (𝒪 D.F)) : wSub D w0 0 = ⊤

theorem **isEtLev_zero** {b : Base D R} {N : ℕ} {s : S ⟶ Sp R} (o : Obj b N s) (w0 : Ideal (𝒪 D.F)) : IsEtLev o w0 0 (fun _ => 1)

theorem **IsEtLev.congr** (h : IsEtLev o w0 r lv) {x y : D.La} (hxy : x - y ∈ wSub D w0 r) : lv x = lv y

theorem **IsEtLev.congr_mk** (h : IsEtLev o w0 r lv) {x y : D.La} (hxy : Submodule.Quotient.mk (p

theorem **mk_liftAct** (g : LevGrp D w0 r) (x : D.La) : Submodule.Quotient.mk (p

theorem **IsEtLev.twist** (h : IsEtLev o w0 r lv) (g : LevGrp D w0 r) : IsEtLev o w0 r (fun x => lv (liftAct g x))

Development.AbelianVarietyTheory.Definitions.Divisors.IsEulerContraction

theorem **extD_apply** (η : abelian₂ K) : extD K η = CechDR.dDR ℤ K η

theorem **mem_locRing** {y : X} {f : X.functionField} : f ∈ locRing y ↔ RegAt y f

Development.AbelianVarietyTheory.Definitions.Divisors.IsExpLieAlgebraComplexUnits

theorem **mem_solG** {P : Polynomial ℂ} {f : G → ℂ} : f ∈ solG h b P ↔ ∃ hf : h.IsSmoothFun f, Polynomial.aeval (∑ i, h.rightDerivEnd (F

Development.AbelianVarietyTheory.Definitions.Divisors.IsFracOfAdjoinEqTop

theorem **xs_ne** (i : Fin (d + 1)) : xs k d i ≠ 0

theorem **isPd_xs** : IsPd k (xs k d) where

theorem **zpow_mem_of_inv** {S : Subalgebra k F} {a : F} (ha : a ∈ S) (ha' : a⁻¹ ∈ S) (m : ℤ) : a ^ m ∈ S

theorem **ratio_mem** {I : Finset (Fin (d + 1))} {i : Fin (d + 1)} (hi : i ∈ I) (j : Fin (d + 1)) : x j / x i ∈ rA k x I

theorem **rA_mono** {I J : Finset (Fin (d + 1))} (h : I ⊆ J) : rA k x I ≤ rA k x J

theorem **mem_tw_iff** (hx : ∀ i, x i ≠ 0) (m : ℤ) {I : Finset (Fin (d + 1))} {i : Fin (d + 1)} (hi : i ∈ I) (f : F) : f ∈ tw k x m I ↔ (x 0 / x i) ^ m * f ∈ rA k x I

theorem **tw_eq_span** (hx : ∀ i, x i ≠ 0) (m : ℤ) {I : Finset (Fin (d + 1))} {i : Fin (d + 1)} (hi : i ∈ I) : tw k x m I = (Submodule.span (rA k x I) {(x i / x 0) ^ m}).restrictScalars k

theorem **isBundle_tw** (hx : ∀ i, x i ≠ 0) (m : ℤ) : IsBundle k x (tw k x m) where

theorem **isFrac_of_adjoin_eq_top** (S : Set F) (h : IntermediateField.adjoin k S = ⊤) : IsFractionRing (Algebra.adjoin k S) F

theorem **ratio_mem_field** (hx : ∀ i, x i ≠ 0) (i a : Fin (d + 1)) : x a / x i ∈ IntermediateField.adjoin k (Set.range (yc x i))

theorem **adjoin_yc_eq_top** (hP : IsPd k x) (i : Fin (d + 1)) : IntermediateField.adjoin k (Set.range (yc x i)) = ⊤

theorem **trdeg_eq** (hP : IsPd k x) : Cardinal.mk (Fin d) ≤ Algebra.trdeg k F

theorem **indep_yc** (hP : IsPd k x) (i : Fin (d + 1)) : AlgebraicIndependent k (yc x i)

theorem **rA_single_eq** (hx : ∀ i, x i ≠ 0) (i : Fin (d + 1)) : rA k x {i} = Algebra.adjoin k (Set.range (yc x i))

theorem **isFrac_rA** (hP : IsPd k x) (i : Fin (d + 1)) : IsFractionRing (rA k x {i}) F

theorem **rB_self_eq** (hP : IsPd k x) (i : Fin (d + 1)) : rB k F x {i} = rA k x {i}

theorem **smooth_chart** (hP : IsPd k x) (i : Fin (d + 1)) : Algebra.Smooth k (rA k x {i})

theorem **isFinCover_self** (hP : IsPd k x) : IsFinCover k F x where

theorem **model** (k : Type) [Field k] (d : ℕ) : IsPd k (xs k d) ∧ IsFinCover k (Fd k d) (xs k d) ∧ ∀ m : ℤ, IsBundle k (xs k d) (tw k (xs k d) m)

theorem **isFinCover_complex** : IsFinCover ℝ ℂ (F

theorem **mem_omA_iff_tw** {k F : Type} [Field k] [Field F] [Algebra k F] {d : ℕ} (x : Fin (d + 1) → F) (hx : ∀ i, x i ≠ 0) (I : Finset (Fin (d + 1))) (f : F) : f ∈ omA k x I ↔ (∏ j, x 0 / x j) * f ∈ tw k x (-((d : ℤ) + 1)) I

theorem **formPair_eq_cup** {k K : Type} [Field k] [Field K] [Algebra k K] {ι : Type} [LinearOrder ι] [Fintype ι] (t : GlobalTheory.CechDR.DR k K →ₗ[k] k) (a b : Finset ι → GlobalTheory.CechDR.DR k K) : formPair k t a b = t (cup (sg k K) a b Finset.univ)

Development.AbelianVarietyTheory.Definitions.Divisors.IsFrameOfForallExists

theorem **resM_linearCombination** (e : ι → Γ(F, U)) (h : V ≤ U) (h' : W ≤ V) (a : ι → Γ(X, V)) : resM F h' (Fintype.linearCombination Γ(X, V) (fun i => resM F h (e i)) a) = Fintype.linearCombination Γ(X, W) (fun i => resM F (h'.trans h) (e i)) fun i => resG X h' (a i)

theorem **basis_equivFun_symm** (he : IsFrame e) (h : V ≤ U) (a : ι → Γ(X, V)) : (he.basis h).equivFun.symm a = ∑ i, a i • resM F h (e i)

@[simp] theorem **basis_apply** (he : IsFrame e) (h : V ≤ U) (i : ι) : he.basis h i = resM F h (e i)

theorem **coe_basis** (he : IsFrame e) (h : V ≤ U) : ⇑(he.basis h) = fun i => resM F h (e i)

theorem **repr_resM** (he : IsFrame e) (h : V ≤ U) (h' : W ≤ V) (s : Γ(F, V)) (i : ι) : (he.basis (h'.trans h)).repr (resM F h' s) i = resG X h' ((he.basis h).repr s i)

theorem **res** (he : IsFrame e) (h : V ≤ U) : IsFrame fun i => resM F h (e i)

theorem **isFrame_of_basis** {e : ι → Γ(F, U)} (b : ∀ V : X.Opens, V ≤ U → Module.Basis ι Γ(X, V) Γ(F, V)) (hb : ∀ V h i, b V h i = resM F h (e i)) : IsFrame e

theorem **isFrame_iff** {e : ι → Γ(F, U)} : IsFrame e ↔ ∀ (V : X.Opens) (h : V ≤ U), LinearIndependent Γ(X, V) (fun i => resM F h (e i)) ∧ ⊤ ≤ Submodule.span Γ(X, V) (Set.range fun i => resM F h (e i))

theorem **isFrame_of_forall_exists** {e : ι → Γ(F, U)} (H : ∀ (V : X.Opens) (h : V ≤ U) (x : X), x ∈ V → ∃ (W : X.Opens) (h' : W ≤ V), x ∈ W ∧ Function.Bijective (Fintype.linearCombination Γ(X, W) fun i => resM F (h'.trans h) (e i))) : IsFrame e

theorem **map_of_linearEquiv** (he : IsFrame e) {e' : ι → Γ(G, U)} (φ : ∀ V : X.Opens, V ≤ U → (Γ(F, V) ≃ₗ[Γ(X, V)] Γ(G, V))) (hφ : ∀ V h i, φ V h (resM F h (e i)) = resM G h (e' i)) : IsFrame e'

theorem **map_iso** (he : IsFrame e) (φ : F ⟶ G) [IsIso φ] : IsFrame fun i => φ.app U (e i)

theorem **comp_equiv** (he : IsFrame e) {κ : Type} [Fintype κ] (σ : κ ≃ ι) : IsFrame (e ∘ σ)

theorem **bijective_stalk** (he : IsFrame e) (x : X) (hx : x ∈ U) : Function.Bijective (Fintype.linearCombination (X.presheaf.stalk x) fun i => (F.presheaf.germ U x hx (e i) : (F.presheaf.stalk x :)))

@[simp] theorem **stalkBasis_apply** (he : IsFrame e) (x : X) (hx : x ∈ U) (i : ι) : he.stalkBasis x hx i = F.presheaf.germ U x hx (e i)

theorem **pullback** [F.IsQuasicoherent] (he : IsFrame e) (f : Y ⟶ X) : IsFrame fun i => QCoh.pullbackSections f F (le_refl (f ⁻¹ᵁ U)) (e i)

theorem **equiv_resM_frame** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (i : ι) : he.equiv he' h (resM F h (e i)) = resM G h (e' i)

theorem **equiv_resM** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : W ≤ V) (s : Γ(F, V)) : he.equiv he' (h'.trans h) (resM F h' s) = resM G h' (he.equiv he' h s)

theorem **iso_hom_app** {e : ι → Γ(F, ⊤)} {e' : ι → Γ(G, ⊤)} (he : IsFrame e) (he' : IsFrame e') (i : ι) : (he.iso he').hom.app ⊤ (e i) = e' i

theorem **isFrame_restrict_iff** (g : Y ⟶ X) [IsOpenImmersion g] {W : Y.Opens} (e : ι → Γ(F, g ''ᵁ W)) : IsFrame (F

theorem **isFrame_one** (U : X.Opens) : IsFrame (F

theorem **isFrame_biproduct** (ι : Type) [Fintype ι] [HasBiproduct fun _ : ι => structureModule X] : IsFrame fun i : ι => (biproduct.ι (fun _ : ι => structureModule X) i).app ⊤ (1 : Γ(X, ⊤))

theorem **isFrame_freeFrame** (ι : Type) [Fintype ι] : IsFrame (freeFrame X ι)

theorem **isVB_of_frames** (hr : Fintype.card ι = r) (h : ∀ x : X, ∃ (U : X.Opens) (e : ι → Γ(F, U)), x ∈ U ∧ IsFrame e) : IsVB F r

theorem **IsFrame.isVB** {e : ι → Γ(F, ⊤)} (he : IsFrame e) : IsVB F (Fintype.card ι)

theorem **isVB_freeModule** (ι : Type) [Fintype ι] : IsVB (freeModule X ι) (Fintype.card ι)

theorem **isVB_freeModule_fin** (r : ℕ) : IsVB (freeModule X (Fin r)) r

theorem **isVB_structureModule** : IsVB (structureModule X) 1

theorem **isFrame_chartSec** {L : LB X} (hL : L.IsLB) (i : L.ι) : IsFrame fun _ : Fin 1 => LB.chartSec hL i

theorem **isVB_toMod** {L : LB X} (hL : L.IsLB) : IsVB (toMod L) 1

theorem **isVB_iff_restrict** : IsVB F r ↔ ∀ x : X, ∃ U : X.Opens, x ∈ U ∧ Nonempty (freeModule U.toScheme (Fin r) ≅ F.restrict U.ι)

theorem **exists_affine** (h : IsVB F r) {x : X} (hx : x ∈ V) : ∃ (U : X.Opens) (e : Fin r → Γ(F, U)), IsAffineOpen U ∧ x ∈ U ∧ U ≤ V ∧ IsFrame e

theorem **of_iso** (h : IsVB F r) (φ : F ≅ G) : IsVB G r

theorem **isQuasicoherent** (h : IsVB F r) : F.IsQuasicoherent

theorem **isFiniteQC** (h : IsVB F r) : IsFiniteQC F

theorem **isLocallyFree** (h : IsVB F r) : F.IsLocallyFree

theorem **nonempty_basis_stalk** (h : IsVB F r) (x : X) : Nonempty (Module.Basis (Fin r) (X.presheaf.stalk x) (F.presheaf.stalk x :))

theorem **free_stalk** (h : IsVB F r) (x : X) : Module.Free (X.presheaf.stalk x) (F.presheaf.stalk x :)

theorem **finrank_stalk** (h : IsVB F r) (x : X) : Module.finrank (X.presheaf.stalk x) (F.presheaf.stalk x :) = r

theorem **finrank_fibre** (h : IsVB F r) (x : X) : Module.finrank (X.residueField x) (QCoh.fibre F x) = r

theorem **rank_unique** [Nonempty X] (h : IsVB F r) (h' : IsVB F s) : r = s

theorem **pullback** (h : IsVB F r) (f : Y ⟶ X) : IsVB ((Scheme.Modules.pullback f).obj F) r

theorem **restrict** (h : IsVB F r) (g : Y ⟶ X) [IsOpenImmersion g] : IsVB (F.restrict g) r

theorem **twist** (h : IsVB F r) {L : LB X} (hL : L.IsLB) : IsVB (twist F L) r

theorem **biprod** (hF : IsVB F r) (hG : IsVB G s) : IsVB (F ⊞ G) (r + s)

theorem **isVB_congr** (φ : F ≅ G) : IsVB F r ↔ IsVB G r

theorem **isVB_zero_iff** : IsVB F 0 ↔ IsZero F

theorem **isSheaf_presheaf** : N.presheaf.IsSheaf

@[simp] theorem **incl_app_mkSec** (s : Γ(F, V)) (hs : s ∈ N.sec V) : N.incl.app V (N.mkSec s hs) = s

theorem **incl_app_injective** (V : X.Opens) : Function.Injective (N.incl.app V)

theorem **incl_app_mem** (s : Γ(N.toMod, V)) : N.incl.app V s ∈ N.sec V

theorem **mem_sec_iff** {s : Γ(F, V)} : s ∈ N.sec V ↔ ∃ t, N.incl.app V t = s

@[reassoc (attr

theorem **exists_comp_incl_iff** (φ : G ⟶ F) : (∃ ψ : G ⟶ N.toMod, ψ ≫ N.incl = φ) ↔ ∀ V s, φ.app V s ∈ N.sec V

theorem **cokernel_π_app_eq_zero_iff** {s : Γ(F, V)} : (cokernel.π N.incl).app V s = 0 ↔ s ∈ N.sec V

theorem **isQuasicoherent** [F.IsQuasicoherent] (h : ∀ (U : X.Opens), IsAffineOpen U → ∀ (f : Γ(X, U)) (t : Γ(F, U)), resM F (X.basicOpen_le f) t ∈ N.sec _ → ∃ n : ℕ, f ^ n • t ∈ N.sec U) : N.toMod.IsQuasicoherent

Development.AbelianVarietyTheory.Definitions.Divisors.IsFreeGenOfFormallyEtale

theorem **isUnit_of_ord_eq_zero** (P : Place c) (b : P.ring) (hb : (b : K) ≠ 0) (h : P.ord (b : K) = 0) : IsUnit b

theorem **formallyEtale_incl** {Q : Subring K} {P : Place c} (h : P.IsPrimeOf Q) : (incl h.1).FormallyEtale

theorem **isFreeGen_of_formallyEtale** {A B : Type} [CommRing A] [CommRing B] {α : k →+* A} {β : k →+* B} {ρ : A →+* B} (hρ : ρ.FormallyEtale) (h : ρ.comp α = β) {x : Fin n → A} (hω : IsFreeGen A (dForm α x)) : IsFreeGen B (dForm β fun i => ρ (x i))

theorem **formOrd_eq_zero** {Q : Subring K} (hQ : IsLocModel c Q) {x : Fin n → Q} (hx : IsFreeGen Q (dForm (c.codRestrict Q hQ.const_mem) x)) {P : Place c} (h : P.IsPrimeOf Q) : P.formOrd (dForm c fun i => (x i : K)) = 0

theorem **exists_place** {Q : Subring K} (hQ : IsLocModel c Q) {t : Q} (ht : Prime t) : ∃ S : Place c, S.IsPrimeOf Q ∧ S.ord (t : K) = 1 ∧ ∀ P : Place c, P.IsPrimeOf Q → P.ring ≠ S.ring → P.ord (t : K) = 0

theorem **exists_plane** (k : Type) [Field k] : ∃ (Q : Subring (FractionRing (MvPolynomial (Fin 2) k))) (hQ : IsLocModel (algebraMap k _) Q) (x : Fin 2 → Q), (∀ i, (x i : FractionRing (MvPolynomial (Fin 2) k)) = algebraMap (MvPolynomial (Fin 2) k) _ (X i)) ∧ Ideal.span {x 0, x 1} = (letI

theorem **sum_ord_sub_one_le** [PerfectField k] {x : Fin n → K} (hx : TrBasis c x) (h0 : dForm c x ≠ 0) (hx0 : ∀ i, x i ≠ 0) (E : Place c) : ∑ i, E.ord (x i) − 1 ≤ E.formOrd (dForm c x)

theorem **mem_maximalIdeal_iff** (P : Place c) (b : P.ring) (hb : (b : K) ≠ 0) : b ∈ IsLocalRing.maximalIdeal P.ring ↔ P.ord (b : K) ≠ 0

theorem **neighbour_ne** {Q : Subring K} (hQ : IsLocModel c Q) {t u : Q} {S E : Place c} (hm : Ideal.span {t, u} = (letI

theorem **eq_of_centre** {Q : Subring K} (hQ : IsLocModel c Q) {t u : Q} {S E : Place c} (hu : ¬ IsUnit u) (hu0 : (u : K) ≠ 0) (hd : ringKrullDim Q = 2) (hp : Prime t) (hS : S.IsPrimeOf Q) (ht : S.ord (t : K) = 1) (hE : E.Contains Q) (h1 : E.ord (t : K) ≠ 0) (h2 : E.ord (u : K) = 0) : E = S

theorem **plane_threshold** (k : Type) [Field k] [CharZero k] : ∃ (K : Type) (_ : Field K) (c : k →+* K) (Q : Subring K) (S : Place c) (x : Fin 2 → K), IsLocModel c Q ∧ TrBasis c x ∧ dForm c x ≠ 0 ∧ S.IsPrimeOf Q ∧ S.ord (x 0) = 1 ∧ (∀ i, x i ≠ 0) ∧ (∀ s : ℚ, S.disc (dForm c x) (Finsupp.single (x 0) 1 + Finsupp.single (x 1) s) = -1 ∧ ¬ Klt c 2 Q (dForm c x) (Finsupp.single (x 0) 1 + Finsupp.single (x 1) s) ∧ (AdjKlt c 2 Q S (dForm c x) (Finsupp.single (x 0) 1 + Finsupp.single (x 1) s) ↔ s < 1)) ∧ ∃ (hQ : IsLocModel c Q) (t u : Q) (E : Place c), Ideal.span {t, u} = (letI

Development.AbelianVarietyTheory.Definitions.Divisors.IsGaloisCover

theorem **mk'** (hG : Finite G) (hfin : IsFinite π) (het : Etale π) (hon : Surjective π) (hinv : ∀ g, act g ≫ π = π) (hmul : ∀ g h, act (g * h) = act h ≫ act g) (hone : act 1 = 𝟙 W) (hfib : ∀ (K : Type) [Field K] [IsAlgClosed K] (a b : Sp K ⟶ W), a ≫ π = b ≫ π → ∃! g : G, b = a ≫ act g) : IsGal π hπ G act where

theorem **finite** (h : IsGal π hπ G act) : Finite G

theorem **isFinite** (h : IsGal π hπ G act) : IsFinite π

theorem **etale** (h : IsGal π hπ G act) : Etale π

theorem **surjective** (h : IsGal π hπ G act) : Surjective π

theorem **act_comp** (h : IsGal π hπ G act) (g : G) : act g ≫ π = π

theorem **act_mul** (h : IsGal π hπ G act) (g g' : G) : act (g * g') = act g' ≫ act g

theorem **act_one** (h : IsGal π hπ G act) : act 1 = 𝟙 W

theorem **fibre** (h : IsGal π hπ G act) (K : Type) [Field K] [IsAlgClosed K] (a b : Sp K ⟶ W) : a ≫ π = b ≫ π ↔ ∃! g : G, b = a ≫ act g

theorem **fibre_cx** (h : IsGal π hπ G act) (a b : CxPts sW (RingHom.id ℂ)) : a.1 ≫ π = b.1 ≫ π ↔ ∃! g : G, b.1 = a.1 ≫ act g

Development.AbelianVarietyTheory.Definitions.Divisors.IsGramOneZero

theorem **ιMulti_comb** {m : ℕ} (b : Fin m → V) (c : Matrix (Fin m) (Fin m) K) : exteriorPower.ιMulti K m (fun i => ∑ j, c i j • b j) = c.det • exteriorPower.ιMulti K m b

theorem **ιMulti_comb_int** {m : ℕ} (b : Fin m → V) (c : Matrix (Fin m) (Fin m) (𝒪 K)) : exteriorPower.ιMulti K m (fun i => ∑ j, c i j • b j) = algebraMap (𝒪 K) K c.det • exteriorPower.ιMulti K m b

theorem **detSpan_span** {m : ℕ} (b : Fin m → V) : detSpan K m (Submodule.span (𝒪 K) (Set.range b)) = Submodule.span (𝒪 K) {exteriorPower.ιMulti K m b}

theorem **idx_span** {m : ℕ} (b : Fin m → V) (hb : exteriorPower.ιMulti K m b ≠ 0) (c : Matrix (Fin m) (Fin m) (𝒪 K)) : idx K m (Submodule.span (𝒪 K) (Set.range b)) (Submodule.span (𝒪 K) (Set.range fun i => ∑ j, c i j • b j)) = Ideal.span {c.det}

theorem **idx_zero** (L N : Submodule (𝒪 K) V) : idx K 0 L N = ⊤

theorem **isGram_one_zero** (X : CMObj H K ρ σ₁ 1 0) : IsGram X (fun _ _ => X.hh (Sum.inl 0)) 0

theorem **isGram_zero_one** (X : CMObj H K ρ σ₁ 0 1) : IsGram X 0 (fun _ _ => X.hh (Sum.inr 0))

theorem **exists_isDiffs** (hU : ThetaUnifAV) (hL : Out.Laws) {p q : ℕ} (X : CMObj H K ρ σ₁ p q) : ∃ (FP : GenFib (X.A true).pol) (FQ : GenFib (X.A false).pol) (FB : GenFib X.B.pol) (MP : Fin p → ((Fin FP.P₀.g → K) →ₗ[K] (Fin FB.P₀.g → K))) (MQ : Fin q → ((Fin FQ.P₀.g → K) →ₗ[K] (Fin FB.P₀.g → K))), IsDiffs FP FQ FB MP MQ

theorem **index_one_zero** (h : IdealSt) (hH : GlobalTheory.IndefiniteRowSideFullHecke.EigenvectorLieDegree.IsHostField H) (X : CMObj H K ρ σ₁ 1 0) : ∃ n : Ideal (𝒪 E), n * n.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {X.hh (Sum.inl 0)} ∧ ∀ (FP : GenFib (X.A true).pol) (FB : GenFib X.B.pol) (M : (Fin FP.P₀.g → K) →ₗ[K] (Fin FB.P₀.g → K)), FP.IsDiffOf FB (phiP X 0) M → GenFib (X.A false).pol → ∀ τ : 𝒪 E →+* 𝒪 K, τ H.rt = ρ → idx K 1 (FB.eigLat X.B.t.act τ) (subP FP FB (fun _ => M) τ) = n.map τ

Development.AbelianVarietyTheory.Definitions.Divisors.IsHeckeMap

theorem **IsHecke.dvd** {g : D.La →ₗ[D.O] D.La} {c k N : ℕ} {s : S ⟶ Sp R} {o : Obj b (N * k) s} {o' : Obj b N s} {ψ : o.A ⟶ o'.A} (h : IsHecke g c k N o o' ψ) : c ∣ k ^ 2

theorem **IsHecke.pol** {g : D.La →ₗ[D.O] D.La} {c k N : ℕ} {s : S ⟶ Sp R} {o : Obj b (N * k) s} {o' : Obj b N s} {ψ : o.A ⟶ o'.A} (h : IsHecke g c k N o o' ψ) : Nonempty (LBIso (lbPow o.P (k ^ 2 / c)) (o'.P.pull (sq2 ψ)))

theorem **IsHecke.eta** {g : D.La →ₗ[D.O] D.La} {c k N : ℕ} {s : S ⟶ Sp R} {o : Obj b (N * k) s} {o' : Obj b N s} {ψ : o.A ⟶ o'.A} (h : IsHecke g c k N o o' ψ) (x : D.La) : o'.eta x = o.eta (g x) ≫ ψ

theorem **IsHeckeMap.over** {g : D.La →ₗ[D.O] D.La} {c k N : ℕ} {F : RepFam b N} {F' : RepFam b (N * k)} {π : F'.X ⟶ F.X} (h : IsHeckeMap g c k N F F' π) : π ≫ F.s = F'.s

theorem **IsHeckeMap.dvd** {g : D.La →ₗ[D.O] D.La} {c k N : ℕ} {F : RepFam b N} {F' : RepFam b (N * k)} {π : F'.X ⟶ F.X} (h : IsHeckeMap g c k N F F' π) : c ∣ k ^ 2

Development.AbelianVarietyTheory.Definitions.Divisors.IsHeightIffOfDegreeEq

theorem **pointHeight_eq** [NumberField F] [NumberField K] [Algebra F K] (w : Matrix (Fin N) (Fin n) F) (A : (F →+* ℂ) → Matrix (Fin N) (Fin N) ℂ) (y : Fin N → K) : pointHeight w A y = ((∑' v : FinitePlace K, Real.log (finSize w (algebraMap F K) v.1 y)) + ∑ τ : K →+* ℂ, Real.log (archSize (A (τ.comp (algebraMap F K))) fun j => τ (y j))) / (Module.finrank ℚ K : ℝ)

theorem **le_finSize** (w : Matrix (Fin N) (Fin n) F) (φ : F →+* K) (v : AbsoluteValue K ℝ) (y : Fin N → K) (k : Fin n) : v (∑ j, φ (w j k) * y j) ≤ finSize w φ v y

theorem **finSize_nonneg** (w : Matrix (Fin N) (Fin n) F) (φ : F →+* K) (v : AbsoluteValue K ℝ) (y : Fin N → K) : 0 ≤ finSize w φ v y

theorem **finSize_smul** (w : Matrix (Fin N) (Fin n) F) (φ : F →+* K) (v : AbsoluteValue K ℝ) (c : K) (y : Fin N → K) : finSize w φ v (c • y) = v c * finSize w φ v y

theorem **archSize_smul** (A : Matrix (Fin N) (Fin N) ℂ) (c : ℂ) (y : Fin N → ℂ) : archSize A (c • y) = ‖c‖ * archSize A y

theorem **finSize_pos** {w : Matrix (Fin N) (Fin n) F} (hw : Submodule.span F (Set.range w.col) = ⊤) (φ : F →+* K) (v : AbsoluteValue K ℝ) {y : Fin N → K} (hy : y ≠ 0) : 0 < finSize w φ v y

theorem **archSize_pos** {A : Matrix (Fin N) (Fin N) ℂ} (hA : A.det ≠ 0) {y : Fin N → ℂ} (hy : y ≠ 0) : 0 < archSize A y

theorem **norm_pairing_le** {A : Matrix (Fin N) (Fin N) ℂ} (hA : A.det ≠ 0) (u y : Fin N → ℂ) : ‖∑ j, u j * y j‖ ≤ frameNorm A u * archSize A y

theorem **pairing_mulVec_le** (w : Matrix (Fin N) (Fin n) F) (φ : F →+* K) {v : AbsoluteValue K ℝ} (hv : IsNonarchimedean v) {x : Fin n → K} (hx : ∀ k, v (x k) ≤ 1) (y : Fin N → K) : v (∑ j, (w.map φ).mulVec x j * y j) ≤ finSize w φ v y

theorem **isHeight_iff** {h : ℝ} : T.IsHeight h ↔ ∃ (d : ℕ) (f : MvPolynomial (Fin (T.D + 1) × Fin T.N) F), IsChowForm T.imageIdeal T.D d f ∧ ((T.D : ℝ) + 1) * d * h = formHeight d T.w T.A f

theorem **isHeight_of_isChowForm** {d : ℕ} {f : MvPolynomial (Fin (T.D + 1) × Fin T.N) F} (hf : IsChowForm T.imageIdeal T.D d f) : T.IsHeight (formHeight d T.w T.A f / (((T.D : ℝ) + 1) * d))

theorem **isHeight_iff_of_degree_eq** {d : ℕ} {g : MvPolynomial (Fin (T.D + 1) × Fin T.N) F} (hg : IsChowForm T.imageIdeal T.D d g) (hd : (d : ℤ) = T.del) {h : ℝ} : T.IsHeight h ↔ ∃ f : MvPolynomial (Fin (T.D + 1) × Fin T.N) F, IsChowForm T.imageIdeal T.D d f ∧ ((T.D : ℝ) + 1) * T.del * h = formHeight d T.w T.A f

theorem **lattice_baseChange** (E : Type) [Field E] [NumberField E] [Algebra F E] : (T.baseChange E).lattice = T.latticeOver E

theorem **wVec_mem** {E : Type} [Field E] [Algebra F E] (x : Fin T.n → 𝒪 E) : T.wVec (fun k => (x k : E)) ∈ T.latticeOver E

theorem **finNormPow_eq_chart** (v : AbsoluteValue K ℝ) (x : T.PtOver φ) {m : ℤ} (f : (lbZPow T.L m).Sec) (i : T.L.ι) (hi : ptOf x.1 ∈ T.L.U i) : T.finNormPow v x f = v (valR x.1 _ (Twist.zfun f i)) / finSize T.w φ v (fun j => valR x.1 _ ((T.b j).f i)) ^ m

theorem **archNormPow_eq_chart** (x : T.PtOver σ) {m : ℤ} (f : (lbZPow T.L m).Sec) (i : T.L.ι) (hi : ptOf x.1 ∈ T.L.U i) : T.archNormPow x f = ‖valR x.1 _ (Twist.zfun f i)‖ / archSize (T.A σ) (fun j => valR x.1 _ ((T.b j).f i)) ^ m

theorem **IsAlgebraicData.finSize_coords_pos** (h : T.IsAlgebraicData) (v : AbsoluteValue K ℝ) (x : T.PtOver φ) : 0 < finSize T.w φ v (T.coords x.1)

theorem **Hermitian.archSize_coords_pos** (h : T.Hermitian) (hb : T.BasePointFree) (x : T.PtOver σ) : 0 < archSize (T.A σ) (T.coords x.1)

theorem **Hermitian.archNorm_le** (h : T.Hermitian) (hb : T.BasePointFree) (x : T.PtOver σ) (u : Fin T.N → ℂ) : T.archNorm x u ≤ frameNorm (T.A σ) u

theorem **finNorm_mulVec_le_one** {v : AbsoluteValue K ℝ} (hv : IsNonarchimedean v) (x : T.PtOver φ) {a : Fin T.n → K} (ha : ∀ k, v (a k) ≤ 1) : T.finNorm v x ((T.w.map φ).mulVec a) ≤ 1

theorem **IsAdelic.isMetrized** {T : Tuple F} (h : T.IsAdelic) (hd : 0 < T.del) : T.IsMetrized

theorem **IsGoodSection.finite_geomPts** {T : Tuple F} {E : Type} [Field E] [Algebra F E] {s : Fin T.D → Fin T.N → E} (h : T.IsGoodSection E s) : (T.geomPts E (T.linSection E s)).Finite

theorem **spans** : (projTuple M).Spans

theorem **shortGenerators** : (projTuple M).ShortGenerators

theorem **isAdelic** (hM : 1 ≤ M) : (projTuple M).IsAdelic

theorem **isMetrized** (hM : 1 ≤ M) : (projTuple M).IsMetrized

theorem **exists_isHeight_line** : ∃ h, (projTuple 1).IsHeight h

Development.AbelianVarietyTheory.Definitions.Divisors.IsHomWeightHomWeight

theorem **FrameAt.eq_ptY** (h : FrameAt X U z j k w t) : z = X.ptY (U.un k w)

theorem **exists_frameAt_of_eq** (h : z = X.ptY (U.un k w)) (hj : ⊤ ≤ z ⁻¹ᵁ X.F.U j) : ∃ t, FrameAt X U z j k w t

theorem **exists_eq_ptY** (hU : IsUnf U) (hz : z ≫ flagStr X = spMap σ) (hO : ⊤ ≤ z ⁻¹ᵁ flagOpen X) : ∃ k w, z = X.ptY (U.un k w)

theorem **homWeight_φ** (la : AllWt.Weight Dw) {σ : c.F →+* ℂ} (j : X.F.J) (y : CxPts M.str σ) (z : CxPts (flagStr X) (goodOf c σ)) (hz : z.1 = flagPt M y) : (homWeight M U x la σ).φ (flagIdx M la j) y = (flagWeight X (U (goodOf c σ)) x la).φ j z

theorem **flagPt_str** {σ : c.F →+* ℂ} (y : CxPts M.str σ) : flagPt M y ≫ flagStr X = spMap (goodOf c σ)

theorem **mem_flagIdx** {j : X.F.J} : y ∈ ptsIn M.str σ ((Lmod M la).U (flagIdx M la j)) ↔ ⊤ ≤ (y.1 ≫ toFlag M) ⁻¹ᵁ X.F.U j

theorem **exists_eq_flagIdx** (i : (Lmod M la).ι) : ∃ j, i = flagIdx M la j

theorem **exists_flagPt_eq** (hU : IsUnf (U (goodOf c σ))) (hy : y ∈ ptsIn M.str σ M.off) : ∃ k w, flagPt M y = X.ptY ((U (goodOf c σ)).un k w)

theorem **exists_frameAt** (hU : IsUnf (U (goodOf c σ))) (hy : y ∈ ptsIn M.str σ M.off) {j : X.F.J} (hj : y ∈ ptsIn M.str σ ((Lmod M la).U (flagIdx M la j))) : ∃ k w t, FrameAt X (U (goodOf c σ)) (flagPt M y) j k w t

theorem **unique** (hU : IsUnf (U (goodOf c σ))) (h : IsHomWeight M U x la φ) (h' : IsHomWeight M U x la φ') (i : (Lmod M la).ι) (hy : y ∈ ptsIn M.str σ M.off) (hi : y ∈ ptsIn M.str σ ((Lmod M la).U i)) : φ.φ i y = φ'.φ i y

theorem **add** (hU : IsUnf (U (goodOf c σ))) {s : AllWt.Weight Dw} (e : a + b = s) {φa : Weight M.str σ (Lmod M a)} {φb : Weight M.str σ (Lmod M b)} {φs : Weight M.str σ (Lmod M s)} (ha : IsHomWeight M U x a φa) (hb : IsHomWeight M U x b φb) (hs : IsHomWeight M U x s φs) (hy : y ∈ ptsIn M.str σ M.off) {j : X.F.J} (hj : ⊤ ≤ (y.1 ≫ toFlag M) ⁻¹ᵁ X.F.U j) : φs.φ (flagIdx M s j) y = φa.φ (flagIdx M a j) y + φb.φ (flagIdx M b j) y

theorem **zero** (hU : IsUnf (U (goodOf c σ))) {φ : Weight M.str σ (Lmod M 0)} (h : IsHomWeight M U x 0 φ) (hy : y ∈ ptsIn M.str σ M.off) {j : X.F.J} (hj : ⊤ ≤ (y.1 ≫ toFlag M) ⁻¹ᵁ X.F.U j) : φ.φ (flagIdx M 0 j) y = 0

theorem **exists_eq** (_ : VolGood U x ω) : ∃ σ, volLogMax U x ω = |Real.log (volHom U x ω σ).toReal|

theorem **nonneg** (h : VolGood U x ω) : 0 ≤ volLogMax U x ω

Development.AbelianVarietyTheory.Definitions.Divisors.IsHomogeneousCompIdeal

theorem **frameNorm_conj** {N : ℕ} {A : Matrix (Fin N) (Fin N) ℂ} (hA : A.det ≠ 0) (x : Fin N → ℂ) : frameNorm (A.map (starRingEnd ℂ)) (fun j => starRingEnd ℂ (x j)) = frameNorm A x

theorem **exists_isFrame** [IsScalarTower (𝒪 K) K V] (hK : ∀ σ : K →+* ℂ, (starRingEnd ℂ).comp σ ≠ σ) {Nm : (K →+* ℂ) → V → ℝ} (hc : ∀ σ x, Nm ((starRingEnd ℂ).comp σ) x = Nm σ x) (hG : HasGram Nm) {L : Submodule (𝒪 K) V} (hL : L.FG) (hs : Submodule.span K (L : Set V) = ⊤) : ∃ (N n : ℕ) (b : Basis (Fin N) K V) (w : Matrix (Fin N) (Fin n) K) (A : (K →+* ℂ) → Matrix (Fin N) (Fin N) ℂ), IsFrame Nm L b w A

theorem **exists_isShortGen** {κ : Type} [Finite κ] (Nm : κ → V → ℝ) {L : Submodule (𝒪 K) V} (hL : L.FG) : ∃ (n : ℕ) (w : Fin n → V) (abelian₂ : ℝ), IsShortGen Nm L w abelian₂

theorem **IsShortGen.log_le_cStar** {κ : Type} {Nm : κ → V → ℝ} {L : Submodule (𝒪 K) V} {n : ℕ} {w : Fin n → V} {abelian₂ : ℝ} (h : IsShortGen Nm L w abelian₂) (hL : L ≠ ⊥) : 0 ≤ Real.log abelian₂ ∧ Real.log abelian₂ ≤ cStar n abelian₂

theorem **uPrime_apply** (u : Weight Dw) (τ : Psi H) (i : Fin (Dw.p + Dw.q)) : uPrime Dw u τ i = if (i : ℕ) < Dw.p then u τ i else u τ i + 3 * ((Dw.p + Dw.q : ℕ) : ℤ)

theorem **nsmul_wtOf** (t : ℕ) (la : Weight Dw) (m : ℤ) : t • wtOf Dw la m = wtOf Dw (t • la) (t • m)

theorem **genVal_apply** (h : GenFrEx Xo c) (Z : Comp Xo c) (f : Rul Xo c) : genVal Z f = evC h Z f

theorem **mem_imageIdeal_genPoint** (Z : Comp Xo c) {C : Submodule c.F (Rul Xo c)} {N : ℕ} (b : Basis (Fin N) c.F C) {d : ℕ} {g : MvPolynomial (Fin N) c.F} (hg : g.IsHomogeneous d) : g ∈ imageIdeal (genPoint Z C) b ↔ genVal Z (aeval (fun j => (b j : Rul Xo c)) g) = 0

theorem **isPrime_compIdeal** (Z : Comp Xo c) (G : WTZ E) {N : ℕ} (b : Basis (Fin N) c.F (cuspOn U Z G)) : (compIdeal U Z G b).IsPrime

theorem **isHomogeneous_compIdeal** (Z : Comp Xo c) (G : WTZ E) {N : ℕ} (b : Basis (Fin N) c.F (cuspOn U Z G)) : (compIdeal U Z G b).IsHomogeneous (homogeneousSubmodule (Fin N) c.F)

theorem **compDeg_eq** {Z : Comp Xo c} {G : WTZ E} {N δ : ℕ} {b : Basis (Fin N) c.F (cuspOn U Z G)} {f : MvPolynomial (Fin (dimFlag Dw + 1) × Fin N) c.F} (hf : IsChowFormInt (compIdeal U Z G b) (dimFlag Dw) δ f) : compDeg U Z G = δ

theorem **not_goodEmb_conj** {σ : c.F →+* ℂ} (h : GoodEmb c σ) : ¬ GoodEmb c ((starRingEnd ℂ).comp σ)

theorem **goodOf_conj** (σ : c.F →+* ℂ) : goodOf c ((starRingEnd ℂ).comp σ) = goodOf c σ

theorem **conj_comp_ne** (σ : c.F →+* ℂ) : (starRingEnd ℂ).comp σ ≠ σ

theorem **petOn_conj** (Z : Comp Xo c) (G : WTZ E) (σ : c.F →+* ℂ) (f : Rul Xo c) : petOn U x V dict Z G ((starRingEnd ℂ).comp σ) f = petOn U x V dict Z G σ f

theorem **isCompHeight_compHeight** {Z : Comp Xo c} {G : WTZ E} (h : GoodOn U x V dict Z G) (hd : 0 < compDeg U Z G) : IsCompHeight U x V dict Z G (compHeight U x V dict Z G)

theorem **CompPresents.isCompHeight** {Z : Comp Xo c} {G : WTZ E} {T : Adelic.HermSys c.F} (hT : CompPresents U x V dict Z G T) {h : ℝ} (hh : T.IsHeight h) : IsCompHeight U x V dict Z G h

theorem **depth_vw_nonneg** (G : WTZ E) {la : HeightOneSpectrum (𝒪 c.F)} {v : Rul Xo c → EReal} (hv : v ∈ valSet Xo c (algCls Xo c) la) : 0 ≤ depth_vw U G v

theorem **vZero_nonneg** (Z : Comp Xo c) : 0 ≤ vZero U Z

theorem **topDeg_stdWt** (Z : Comp Xo c) : topDeg U Z (stdWt (Psi H) (Dw.p + Dw.q)) = (growthOn U Z (wtOf Dw (stdWt (Psi H) (Dw.p + Dw.q)) 0)).toReal

theorem **delPrime_eq** (Z : Comp Xo c) {u : Weight Dw} (h : vdm u ≠ 0) : delPrime U Z u = (1 - rhoB Dw u) * topDeg U Z u

theorem **latVal_recVals** (Z : Comp Xo c) (G : WTZ E) : latVal U (recVals Xo c) Z G = latOn U Z G

theorem **smallOn_mono** (Vs : ValFam Xo c) (Z : Comp Xo c) (G : WTZ E) : Monotone (smallOn U x Vs Z G)

theorem **compVol_mono** (a : ℕ) (Vs : ValFam Xo c) (Z : Comp Xo c) (G : WTZ E) : Monotone (compVol U x a Vs Z G)

theorem **le_compVolSlope_iff** (a : ℕ) (Vs : ValFam Xo c) {Z : Comp Xo c} {la : Weight Dw} (h : 0 < topDeg U Z la) (s : ℝ) : (s : EReal) ≤ compVolSlope U x a Vs Z la ↔ ∀ t : ℝ, ENNReal.ofReal (topDeg U Z la * (s + t)) ≤ compVol U x a Vs Z (wtOf Dw la 0) t

theorem **isLimSlope_iff** {D' : D7} (Y : Formalism.BasicNotions.DV.ShimX D'.what₂.Tau) (B : ℝ) (u : D'.what₂.Tau → Fin Y.N → ℝ) (k : Y.Comp) (s : ℝ) : IsLimSlope (Formalism.MainNotions.WeightThresholdDefs.dimY Y) (fun t => Y.arSelfC (Y.Mbar B u t) k) (fun t => Y.geoC (Y.Mbar B u t) k) s ↔ IndefiniteRowSideFullHecke.EqualVolumeMeanCorrection.IsSlope Y B u k s

Development.AbelianVarietyTheory.Definitions.Divisors.IsHtFunGenericIff

theorem **isHtFun_generic_iff** [IsIntegral X] {p : X ⟶ Sp k} (L : ι → HermLB p) (D : ℕ) (Φ : MvPolynomial ι ℝ →ₗ[ℝ] ℝ) : IsHtFun L (genericPoint X) (D + 1) Φ ↔ IsArFun L D Φ

theorem **isIntFun_one_iff** {p : X ⟶ Sp k} (L : ι → HermLB p) (D : ℕ) (Λ : MvPolynomial ι ℝ →ₗ[ℝ] ℝ) : IsIntFun L (fun _ _ => 1) D Λ ↔ IsGeoFun L D Λ

theorem **vdmPol_one** (e : ℕ) : vdmPol 1 e = MvPolynomial.X (some 0) ^ 2 * MvPolynomial.X none ^ e

theorem **lam_B** (R : ℝ) : (M.lam R).B = M.lamB

theorem **isLB_pow** (m : ℕ) : (lbPow M.lamB m).IsLB

Development.AbelianVarietyTheory.Definitions.Divisors.IsIsoFromNormalization

theorem **isGen_fromSpecStalk** (C : Scheme.{0}) [IsIntegral C] : IsGen (K

Development.AbelianVarietyTheory.Definitions.Divisors.IsIsoGenResMapSmulHom

theorem **dim_add_dim_le** {S T W : Type*} [TopologicalSpace S] [TopologicalSpace T] [TopologicalSpace W] [QuasiSober S] [T0Space S] [QuasiSober T] [T0Space T] [QuasiSober W] [T0Space W] [IrreducibleSpace T] {f : S → T} (hf : IsClosedMap f) {j : W → S} (hj : Continuous j) (hi : Function.Injective j) (h : ∀ w, f (j w) = genericPoint T) : topologicalKrullDim W + topologicalKrullDim T ≤ topologicalKrullDim S

theorem **isPullback_genFibre** : IsPullback f.genFibreι f.genFibreStr f.f (Z.X.fromSpecResidueField (genericPoint Z.X))

theorem **dim_genFibre_add_le** : topologicalKrullDim f.genFibre + (Z.relDim + 1 : ℕ) ≤ (Z₁.relDim + 1 : ℕ)

theorem **exists_dim_genFibre** : ∃ n : ℕ, topologicalKrullDim f.genFibre = n ∧ n + Z.relDim ≤ Z₁.relDim

theorem **relDim_le** : Z.relDim ≤ Z₁.relDim

theorem **relDim_add** : Z.relDim + f.relDim = Z₁.relDim

theorem **dim_genFibre_le** : topologicalKrullDim f.genFibre ≤ f.relDim

theorem **diffVanish_genFibre** : DiffVanish (f.relDim + 1) (chiPic f.fnField (structureModule f.genFibre))

theorem **fibreDeg_prod** {ι : Type} [Fintype ι] {L : ι → LB Z₁.X} (hL : ∀ i, (L i).IsLB) : f.fibreDeg (∏ i, c1 (L i) (hL i)) = if Fintype.card ι = f.relDim then (interNum f.fnField (structureModule f.genFibre) fun i => (L i).pull f.genFibreι : ℚ) else 0

theorem **genEquiv_apply** {V : Z.X.Opens} (h : V ≤ Z.gen) (a : Γ(Z.X, V)) : genEquiv h a = (Z.gen.ι.app V).hom a

theorem **genEquiv_resG** {U V : Z.X.Opens} (h : V ≤ Z.gen) (h' : V ≤ U) (a : Γ(Z.X, U)) : genEquiv h (resG Z.X h' a) = (Z.gen.ι.appLE U _ (Z.gen.ι.preimage_mono h')).hom a

theorem **isIso_genRes_map_smulHom** (E : Z.X.Modules) {g : O} (hg : g ≠ 0) : IsIso (Z.genRes.map (smulHom E (Z.const g)))

theorem **smul_inv** (g : O) (hg : g ≠ 0) : (smul E g hg).inv = Z.genRes.map (smulHom E (Z.const g))

theorem **pull_inv** {φ : Z.GenIso E E'} {ψ : E' ⟶ E} (h : φ.inv = Z.genRes.map ψ) : (φ.pull hf).inv = Z₁.genRes.map ((Scheme.Modules.pullback f).map ψ)

theorem **pull_smul** (g : O) (hg : g ≠ 0) : (smul E g hg).pull hf = smul _ g hg

theorem **detUnit_rel** (c d : Chart E r) (c' d' : Chart E' r) [E.IsQuasicoherent] [E'.IsQuasicoherent] {V : Z.X.Opens} (hc : V ≤ c.U ⊓ c'.U) (hd : V ≤ d.U ⊓ d'.U) {T : Z.gen.toScheme.Opens} (h : T ≤ Z.gen.ι ⁻¹ᵁ V) : φ.detUnit c c' (h.trans (Z.gen.ι.preimage_mono hc)) = Units.map (Z.gen.ι.appLE V T h).hom.toMonoidHom (c.isFrame.detUnit d.isFrame (hc.trans inf_le_left) (hd.trans inf_le_left) * (c'.isFrame.detUnit d'.isFrame (hc.trans inf_le_right) (hd.trans inf_le_right))⁻¹) * φ.detUnit d d' (h.trans (Z.gen.ι.preimage_mono hd))

theorem **genEquiv_eqn_div** (c : Chart E r) (c' : Chart E' r) {W : Z.X.Opens} (h : W ≤ Z.gen ⊓ (c.U ⊓ c'.U)) : haveI

theorem **vCls_div** : Z.vCls (φ.div hE hE').cls = cls _ hE.isLB_det - cls _ hE'.isLB_det

theorem **cls_div_trans** (φ' : Z.GenIso E' E'') : (div (φ ≪≫ φ') hE hE'').cls = (φ.div hE hE').cls + (φ'.div hE' hE'').cls

theorem **eqn_div_of_hom** (hψ : φ.inv = Z.genRes.map ψ) (c : Chart E r) (c' : Chart E' r) {W V : Z.X.Opens} (h : W ≤ Z.gen ⊓ V) (hV : V ≤ c.U ⊓ c'.U) : (φ.div hE hE').eqn (c, c') (h.trans (inf_le_inf_left _ hV)) = resG Z.X (h.trans inf_le_right) ((c.isFrame.basis (hV.trans inf_le_left)).det fun i => ψ.app V (resM E' (hV.trans inf_le_right) (c'.e i)))

theorem **isEff_cls_div** (hψ : φ.inv = Z.genRes.map ψ) : IsEff (φ.div hE hE').cls

theorem **cls_div_of_smul** {φ : Z.GenIso E E} {g : O} (hg : g ≠ 0) (hφ : φ.inv = Z.genRes.map (smulHom E (Z.const g))) : (φ.div hE hE).cls = r • Z.div g hg

theorem **isVB_self** : IsVB E N

theorem **isVB_cokernel** (i : Fin (N + 1)) : IsVB (cokernel (fl.ι i)) (N - i)

theorem **isVB_gr** (i : Fin N) : IsVB (fl.gr i) 1

theorem **isLB_root** (i : Fin N) : (fl.root i).IsLB

theorem **cls_det** : cls _ fl.isVB_self.isLB_det = ∑ i, cls _ (fl.isLB_root i)

theorem **nonempty_gr_ofBiprod** {L M : X.Modules} (hL : IsVB L 1) (hM : IsVB M 1) : Nonempty ((ofBiprod hL hM).gr 0 ≅ L) ∧ Nonempty ((ofBiprod hL hM).gr 1 ≅ M)

theorem **gen_unique** (i : Fin (N + 1)) (ψ : Z.GenIso (P.fl.F i) (P.fl'.F i)) (h : ψ.hom ≫ Z.genRes.map (P.fl'.ι i) = Z.genRes.map (P.fl.ι i) ≫ φ.hom) : ψ = P.gen i

theorem **gen_step** (i : Fin N) : Z.genRes.map (P.fl.step i) ≫ (P.gen i.succ).hom = (P.gen i.castSucc).hom ≫ Z.genRes.map (P.fl'.step i)

theorem **π_grIso_hom** (i : Fin N) : Z.genRes.map (P.fl.π i) ≫ (P.grIso i).hom = (P.gen i.succ).hom ≫ Z.genRes.map (P.fl'.π i)

theorem **cls_root'** (i : Fin N) : cls _ (P.fl'.isLB_root i) = cls _ (P.fl.isLB_root i) - Z.vCls (P.div i).cls

theorem **cls_div_ofSMul** (fl : Flag E N) (g : O) (hg : g ≠ 0) (i : Fin N) : ((ofSMul fl g hg).div i).cls = Z.div g hg

Development.AbelianVarietyTheory.Definitions.Divisors.IsIsoQuotToImOmegaLieOfSplit

theorem **le_preimage_top** {X Y : Scheme.{0}} (f : X ⟶ Y) (V : X.Opens) : V ≤ f ⁻¹ᵁ ⊤

theorem **appLE_top** {X Y : Scheme.{0}} (f : X ⟶ Y) (a : Γ(Y, ⊤)) : f.appLE ⊤ ⊤ (le_preimage_top f ⊤) a = f.appTop a

theorem **sec_hom** (s : 𝟙_ (Over S) ⟶ T) : sec s ≫ T.hom = 𝟙 S

theorem **cotMap_d** (φ : T' ⟶ T) (h : s' ≫ φ = s) {V : S.Opens} {W : T.left.Opens} {W' : T'.left.Opens} (hW : V ≤ sec s ⁻¹ᵁ W) (hW' : V ≤ sec s' ⁻¹ᵁ W') (k : W' ≤ φ.left ⁻¹ᵁ W) (f : Γ(T.left, W)) : (cotMap φ h).app V (d (sec_hom s) hW f) = d (sec_hom s') hW' (φ.left.appLE W W' k f)

theorem **cotMap_id** {φ : T ⟶ T} (hφ : φ = 𝟙 T) (h : s ≫ φ = s) : cotMap φ h = 𝟙 _

theorem **cotMap_comp** {φ : T' ⟶ T} {ψ : T'' ⟶ T'} (h : s' ≫ φ = s) (h' : s'' ≫ ψ = s') {χ : T'' ⟶ T} (hχ : ψ ≫ φ = χ) (h'' : s'' ≫ χ = s) : cotMap φ h ≫ cotMap ψ h' = cotMap χ h''

theorem **cotMap_eq_zero** {φ : T' ⟶ T} (hφ : toUnit T' ≫ s = φ) (h : s' ≫ φ = s) : cotMap φ h = 0

theorem **cotMap_prod** [QuasiSeparated T.hom] [QuasiSeparated T'.hom] (i₁ : s ≫ lift (𝟙 T) (toUnit T ≫ s') = lift s s') (i₂ : s' ≫ lift (toUnit T' ≫ s) (𝟙 T') = lift s s') (p₁ : lift s s' ≫ fst T T' = s) (p₂ : lift s s' ≫ snd T T' = s') : cotMap _ i₁ ≫ cotMap _ p₁ + cotMap _ i₂ ≫ cotMap _ p₂ = 𝟙 _

theorem **cotMap_mul** {G : Over S} [GrpObj G] [QuasiSeparated G.hom] {x y : T ⟶ G} (hx : s ≫ x = η) (hy : s ≫ y = η) (h : s ≫ (x * y) = η) : cotMap (x * y) h = cotMap x hx + cotMap y hy

theorem **unitSec_eq_eSec** : unitSec G = eSec G

theorem **resM_dUnit** (h : V ≤ unitSec G ⁻¹ᵁ W) (h' : V' ≤ V) (f : Γ(G.left, W)) : resM _ h' (dUnit G h f) = dUnit G (h'.trans h) f

theorem **dUnit_resG** (h : V ≤ unitSec G ⁻¹ᵁ W') (k : W' ≤ W) (f : Γ(G.left, W)) : dUnit G h (resG _ k f) = dUnit G (h.trans ((unitSec G).preimage_mono k)) f

theorem **dUnit_mul** (h : V ≤ unitSec G ⁻¹ᵁ W) (f g : Γ(G.left, W)) : dUnit G h (f * g) = (unitSec G).appLE W V h f • dUnit G h g + (unitSec G).appLE W V h g • dUnit G h f

theorem **dUnit_base** (h : V ≤ unitSec G ⁻¹ᵁ W) (k : W ≤ G.hom ⁻¹ᵁ U) (a : Γ(S, U)) : dUnit G h (G.hom.appLE U W k a) = 0

theorem **omega_hom_ext** {F : S.Modules} {φ ψ : omega G ⟶ F} (H : ∀ (W : G.left.Opens) (f : Γ(G.left, W)), φ.app _ (dUnit G (le_refl (unitSec G ⁻¹ᵁ W)) f) = ψ.app _ (dUnit G le_rfl f)) : φ = ψ

theorem **omegaHomEquiv_app** {F : S.Modules} (φ : omega G ⟶ F) {V : S.Opens} {W : G.left.Opens} (h : V ≤ unitSec G ⁻¹ᵁ W) (f : Γ(G.left, W)) : (omegaHomEquiv G F φ).app h f = φ.app V (dUnit G h f)

theorem **omegaKaehlerEquiv_dUnit** {U : S.Opens} {W : G.left.Opens} (c : IsChart G.hom (unitSec G) U W) (a : Γ(G.left, W)) : letI

theorem **isVB_omega** (g : ℕ) [SmoothOfRelativeDimension g G.hom] : IsVB (omega G) g

theorem **isVB_Lie** (g : ℕ) [SmoothOfRelativeDimension g G.hom] : IsVB (Lie G) g

theorem **exists_isFrame_omega** [Smooth G.hom] (x : S) : ∃ (U : S.Opens) (n : ℕ) (s : Fin n → Γ(omega G, U)), x ∈ U ∧ IsFrame s

theorem **exists_isVB** [Smooth G.hom] [PreconnectedSpace S] : ∃ g, IsVB (omega G) g ∧ IsVB (Lie G) g

theorem **exists_isVB_of_isSemiAbelian** [h : IsSemiAbelian G] [PreconnectedSpace S] : ∃ g, IsVB (omega G) g ∧ IsVB (Lie G) g

theorem **omegaMap_dUnit** (φ : G ⟶ H) [IsMonHom φ] {V : S.Opens} {W : H.left.Opens} {W' : G.left.Opens} (h : V ≤ unitSec H ⁻¹ᵁ W) (h' : V ≤ unitSec G ⁻¹ᵁ W') (k : W' ≤ φ.left ⁻¹ᵁ W) (f : Γ(H.left, W)) : (omegaMap φ).app V (dUnit H h f) = dUnit G h' (φ.left.appLE W W' k f)

theorem **omegaMap_congr** {φ ψ : G ⟶ H} [IsMonHom φ] [IsMonHom ψ] (h : φ = ψ) : omegaMap φ = omegaMap ψ

@[simp] theorem **omegaMap_id** : omegaMap (𝟙 G) = 𝟙 _

theorem **omegaMap_comp** (φ : G ⟶ H) (ψ : H ⟶ K) [IsMonHom φ] [IsMonHom ψ] : omegaMap (φ ≫ ψ) = omegaMap ψ ≫ omegaMap φ

theorem **omegaMap_one** [IsMonHom (1 : G ⟶ H)] : omegaMap (1 : G ⟶ H) = 0

theorem **omegaMap_mul** [QuasiSeparated H.hom] (φ ψ : G ⟶ H) [IsMonHom φ] [IsMonHom ψ] [IsMonHom (φ ∗ ψ)] : omegaMap (φ ∗ ψ) = omegaMap φ + omegaMap ψ

@[simp] theorem **lieMap_id** : lieMap (𝟙 G) = 𝟙 _

theorem **lieMap_comp** (φ : G ⟶ H) (ψ : H ⟶ K) [IsMonHom φ] [IsMonHom ψ] : lieMap (φ ≫ ψ) = lieMap φ ≫ lieMap ψ

theorem **lieMap_one** [IsMonHom (1 : G ⟶ H)] : lieMap (1 : G ⟶ H) = 0

theorem **lieMap_mul** [QuasiSeparated H.hom] (φ ψ : G ⟶ H) [IsMonHom φ] [IsMonHom ψ] [IsMonHom (φ ∗ ψ)] : lieMap (φ ∗ ψ) = lieMap φ + lieMap ψ

theorem **unitSec_comp** (h : IsGrpPullback G' G f φ) : unitSec G' ≫ φ = f ≫ unitSec G

theorem **omegaBase_dUnit** (h : IsGrpPullback G' G f φ) {V : S.Opens} {W : G.left.Opens} (hW : V ≤ unitSec G ⁻¹ᵁ W) {T : S'.Opens} (hT : T ≤ f ⁻¹ᵁ V) {W' : G'.left.Opens} (hW' : T ≤ unitSec G' ⁻¹ᵁ W') (k : W' ≤ φ ⁻¹ᵁ W) (a : Γ(G.left, W)) : (omegaBase h).app T (pullbackSections f _ hT (dUnit G hW a)) = dUnit G' hW' (φ.appLE W W' k a)

theorem **omegaBase_naturality** (h : IsGrpPullback G' G f φ) (h' : IsGrpPullback H' H f ψ) (a : G ⟶ H) (a' : G' ⟶ H') [IsMonHom a] [IsMonHom a'] (w : a'.left ≫ ψ = φ ≫ a.left) : (Scheme.Modules.pullback f).map (omegaMap a) ≫ omegaBase h = omegaBase h' ≫ omegaMap a'

theorem **isIso_lieBase** (h : IsGrpPullback G' G f φ) {g : ℕ} (hg : IsVB (omega G) g) : IsIso (lieBase h)

theorem **lieBase_naturality** (h : IsGrpPullback G' G f φ) (h' : IsGrpPullback H' H f ψ) (a : G ⟶ H) (a' : G' ⟶ H') [IsMonHom a] [IsMonHom a'] (w : a'.left ≫ ψ = φ ≫ a.left) : (Scheme.Modules.pullback f).map (lieMap a) ≫ lieBase h' = lieBase h ≫ lieMap a'

theorem **isVB_omega_of_isGrpPullback** (h : IsGrpPullback G' G f φ) {g : ℕ} (hg : IsVB (omega G) g) : IsVB (omega G') g

@[reassoc] theorem **eigenImπ_dualImι** : eigenImπ (dualAct ρ) σ ≫ dualImι ρ σ = dualMap (eigenι ρ σ)

theorem **mono_dualImι** (h : IsVB F g) : Mono (dualImι ρ σ)

theorem **isIso_dualImι** (h : IsVB F g) (h' : IsVB (eigenIm (dualAct ρ) σ) r) : IsIso (dualImι ρ σ)

theorem **comp_eigenImMap_app_eq_zero_iff** {O : Type} [CommRing O] {E F F' : S.Modules} {ρ : O →+* End F} {ρ' : O →+* End F'} {σ : O →+* Γ(S, ⊤)} (k : E ⟶ eigenIm ρ σ) {φ : F ⟶ F'} [IsIso φ] (h : IsEquivariant ρ ρ' φ) {V : S.Opens} {s : Γ(E, V)} : (k ≫ eigenImMap σ h).app V s = 0 ↔ k.app V s = 0

theorem **act_omegaAct** (a : O) : act (omegaAct i) a = @omegaMap _ _ _ _ _ (i.act a) (i.hom a)

theorem **act_lieAct** (a : O) : act (lieAct i) a = @lieMap _ _ _ _ _ (i.act a) (i.hom a)

theorem **act_omegaConjAct** (c : O →+* O) (a : O) : act (omegaConjAct i c) a = @omegaMap _ _ _ _ _ (i.act (c a)) (i.hom _)

theorem **isEquivariant_omegaIsoDualLie** {g : ℕ} (h : IsVB (omega G) g) : IsEquivariant (omegaAct i) (dualAct (lieAct i)) (omegaIsoDualLie h).hom

theorem **omegaSubIsoDualLieSub_hom** {g : ℕ} (h : IsVB (omega G) g) [IsIso (eigenι (lieAct i) σ ≫ eigenπ (lieAct i) σ)] : (omegaSubIsoDualLieSub i σ h).hom = eigenι (omegaAct i) σ ≫ (omegaIsoDualLie h).hom ≫ dualMap (eigenι (lieAct i) σ)

@[reassoc] theorem **eigenImπ_omegaImι** : eigenImπ (omegaAct i) σ ≫ omegaImι i σ = cartierDualToBidual (omega G) ≫ dualMap (eigenι (lieAct i) σ)

theorem **isIso_omegaImι_of_epi** {g : ℕ} (h : IsVB (omega G) g) [e : Epi (dualMap (eigenι (lieAct i) σ))] : IsIso (omegaImι i σ)

@[reassoc] theorem **dualOmegaImIso_hom_ι** : (dualOmegaImIso i σ).hom ≫ eigenι (lieAct i) σ = dualMap (eigenImπ (omegaAct i) σ)

theorem **isVB_lieSub** {r : ℕ} (h : IsVB (omegaIm i σ) r) : IsVB (lieSub i σ) r

@[reassoc] theorem **eigenImπ_lieImι** : eigenImπ (lieAct i) σ ≫ lieImι i σ = dualMap (eigenι (omegaAct i) σ)

theorem **mono_lieImι** {g : ℕ} (h : IsVB (omega G) g) : Mono (lieImι i σ)

theorem **isIso_lieImι** {g r : ℕ} (h : IsVB (omega G) g) (h' : IsVB (lieIm i σ) r) : IsIso (lieImι i σ)

theorem **isIso_quotToIm_omega_lie_of_split** {k κ : Type} [Field k] [Algebra k Γ(S, ⊤)] [Fintype κ] {τ : κ → (O →+* k)} (hτ : IsSplitBy τ) {g : ℕ} (h : IsVB (omega G) g) (j : κ) : IsIso (quotToIm (omegaAct i) (scal S (τ j))) ∧ IsIso (quotToIm (lieAct i) (scal S (τ j)))

theorem **isEquivariant_omegaMap** {i' : OAct O H} (φ : G ⟶ H) [IsMonHom φ] (h : ∀ a, i.act a ≫ φ = φ ≫ i'.act a) : IsEquivariant (omegaAct i') (omegaAct i) (omegaMap φ)

theorem **isEquivariant_omegaMap_conj** {i' : OAct O H} (c : O →+* O) (φ : G ⟶ H) [IsMonHom φ] (h : ∀ a, i.act (c a) ≫ φ = φ ≫ i'.act a) : IsEquivariant (omegaAct i') (omegaConjAct i c) (omegaMap φ)

theorem **isEquivariant_lieMap** {i' : OAct O H} (φ : G ⟶ H) [IsMonHom φ] (h : ∀ a, i.act a ≫ φ = φ ≫ i'.act a) : IsEquivariant (lieAct i) (lieAct i') (lieMap φ)

theorem **isEquivariant_omegaBase** : IsEquivariant (pullAct (omegaAct i) f) (omegaAct i') (omegaBase h)

theorem **isEquivariant_lieBase** : IsEquivariant (pullAct (lieAct i) f) (lieAct i') (lieBase h)

theorem **isIso_omegaSubBase** {k κ : Type} [Field k] [Algebra k Γ(S, ⊤)] [Fintype κ] {τ : κ → (O →+* k)} (hτ : IsSplitBy τ) (j : κ) : IsIso (omegaSubBase (scal S (τ j)) h w)

theorem **isIso_lieSubBase** {k κ : Type} [Field k] [Algebra k Γ(S, ⊤)] [Fintype κ] {τ : κ → (O →+* k)} (hτ : IsSplitBy τ) {g : ℕ} (hg : IsVB (omega G) g) (j : κ) : IsIso (lieSubBase (scal S (τ j)) h w)

@[reassoc] theorem **map_eigenImπ_omegaImBase** : (Scheme.Modules.pullback f).map (eigenImπ _ σ) ≫ omegaImBase h w hg hf hε = omegaBase h ≫ eigenImπ _ _

theorem **omegaImBase_app_eq_zero_iff** {V : S'.Opens} {s : Γ((Scheme.Modules.pullback f).obj (omegaIm i σ), V)} : (omegaImBase h w hg hf hε).app V s = 0 ↔ resG S' (le_top : V ≤ ⊤) (f.appTop D) • s = 0

theorem **isIso_omegaImBase** {r : ℕ} (hr : IsVB (omegaIm i σ) r) : IsIso (omegaImBase h w hg hf hε)

theorem **isIso_lieSubBase_of_isVB** {r : ℕ} (hr : IsVB (omegaIm i σ) r) : IsIso (lieSubBase σ h w)

@[reassoc] theorem **map_eigenImπ_lieImBase** : (Scheme.Modules.pullback f).map (eigenImπ _ σ) ≫ lieImBase h w hf hε = lieBase h ≫ eigenImπ _ _

theorem **epi_lieImBase** : Epi (lieImBase h w hf hε)

theorem **lieImBase_app_eq_zero_iff** {V : S'.Opens} {s : Γ((Scheme.Modules.pullback f).obj (lieIm i σ), V)} : (lieImBase h w hf hε).app V s = 0 ↔ resG S' (le_top : V ≤ ⊤) (f.appTop D) • s = 0

theorem **isIso_lieImBase** {r : ℕ} (hr : IsVB (lieIm i σ) r) : IsIso (lieImBase h w hf hε)

theorem **isFrame_d_eps** : IsFrame fun _ : Fin 1 => d (sec_hom (epsZero O)) (le_preimage_top _ ⊤) (epsFn O)

theorem **epsIso_d** : (epsIso O).hom.app ⊤ (d (sec_hom (epsZero O)) (le_preimage_top _ _) (epsFn O)) = (1 : Γ(Sp O, ⊤))

theorem **epsZero_dualScale** (a : O) : epsZero O ≫ ptMap (dualScale a) = epsZero O

theorem **cotMap_dualScale** (a : O) : cotMap (ptMap (dualScale a)) (epsZero_dualScale a) = smulHom _ (cst O a)

theorem **epsZero_comp** (x : LieG G) : epsZero O ≫ x.1 = η

theorem **dualApp_tangent** (x : LieG G) {V : (Sp O).Opens} (w : Γ(omega G, V)) : dualApp (tangent x) le_top w = (epsIso O).hom.app V ((cotMap x.1 (epsZero_comp x)).app V w)

theorem **dualApp_tangent_dUnit** (x : LieG G) {V : (Sp O).Opens} {W : G.left.Opens} (h : V ≤ unitSec G ⁻¹ᵁ W) (f : Γ(G.left, W)) {T : (specO O (DualNumber O)).left.Opens} (hV : V ≤ sec (epsZero O) ⁻¹ᵁ T) (hT : T ≤ (specO O (DualNumber O)).hom ⁻¹ᵁ V) (k : T ≤ x.1.left ⁻¹ᵁ W) (a b : Γ(Sp O, V)) (e : x.1.left.appLE W T k f = (specO O (DualNumber O)).hom.appLE V T hT a + (specO O (DualNumber O)).hom.appLE V T hT b ∗ resG _ le_top (epsFn O)) : dualApp (tangent x) le_top (dUnit G h f) = b

theorem **tangent_mul** [QuasiSeparated G.hom] (x y : LieG G) (h) : tangent ⟨x.1 ∗ y.1, h⟩ = tangent x + tangent y

theorem **tangent_scale** (a : O) (x : LieG G) (h) : tangent ⟨ptMap (dualScale a) ≫ x.1, h⟩ = cst O a • tangent x

theorem **tangent_comp** (φ : G ⟶ H) [IsMonHom φ] (x : LieG G) (h) : tangent ⟨x.1 ≫ φ, h⟩ = (lieMap φ).app ⊤ (tangent x)

theorem **toOmega_apply** [QuasiSeparated G.hom] (w : Γ(omega G, ⊤)) (x : LieG G) : (toOmega G w).1 x = (Scheme.ΓSpecIso (.of O)).hom (dualApp (tangent x) le_top w)

theorem **toOmega_smul** [QuasiSeparated G.hom] (a : O) (w : Γ(omega G, ⊤)) : toOmega G (cst O a • w) = a • toOmega G w

theorem **dlogChar_apply** (a : X) : dlogChar S X a = dUnit _ (le_preimage_top _ _) (Rosenlicht.chi X a)

theorem **omegaMap_dlogChar** (u : Y ↠+ X) (b : Y) : (omegaMap (SplitTorus.torusMap S u)).app ⊤ (dlogChar S Y b) = dlogChar S X (u b)

theorem **exists_basis_augCot** (R : Type) [CommRing R] : ∃ c : Module.Basis κ R (AugCot (Bialgebra.counitAlgHom R (AddMonoidAlgebra R X))), ∀ i, c i = AugCot.d _ (single (b i) 1)

theorem **isFrame_dlogChar** : IsFrame fun i => dlogChar S X (b i)

theorem **isFrame_Gm** : IsFrame fun _ : Fin 1 => dlogChar S (ULift ℤ) (ULift.up 1)

theorem **isVB_Gm** : IsSemiAbelian (Gm S) ∧ IsVB (omega (Gm S)) 1 ∧ IsVB (Lie (Gm S)) 1

theorem **actsThrough_omegaAct_int** (i : OAct ℤ G) : ActsThrough (omegaAct i) (Int.castRingHom _)

theorem **actsThrough_lieAct_int** (i : OAct ℤ G) : ActsThrough (lieAct i) (Int.castRingHom _)

Development.AbelianVarietyTheory.Definitions.Divisors.IsKMOne

theorem **_root_.GlobalTheory.UCls.isKM_one** (c : E ≃ₐ[F] E) (G : Matrix (Fin n) (Fin n) E) (L : Set (Fin n → E)) (M : ℕ) : _root_.GlobalTheory.UCls.IsKM c G L M 1

Development.AbelianVarietyTheory.Definitions.Divisors.IsKSValueOneOfRetraction

theorem **pullOver_pullOver** {Z Y X : Over S} (g : Z ⟶ Y) (f : Y ⟶ X) (c : Cris.H1 κ (X.hom ≫ t)) : pullOver κ t g (pullOver κ t f c) = pullOver κ t (g ≫ f) c

theorem **mem_prim_iff** (A : Over S) [GrpObj A] (c : Cris.H1 κ (A.hom ≫ t)) : c ∈ prim κ t A ↔ pullOver κ t μ[A] c = pullOver κ t (fst A A) c + pullOver κ t (snd A A) c

theorem **of_isGrpPullback** (hσ : IsDPMorphism κ κ' σ) (hb : b ≫ t = t' ≫ spMap σ) (h : IsGrpPullback A' A b φ) : GrpMor κ t A κ' t' A' σ b φ

private theorem **mul_left_eq_lift_comp_mu** {X : Scheme.{0}} {G : Over X} [GrpObj G] {T : Over X} (x y : T ⟶ G) : (x ∗ y).left = pullback.lift x.left y.left ((Over.w x).trans (Over.w y).symm) ≫ μ[G].left

theorem **of_hom** {u : S ⟶ Sp R'} {B : Over S} [GrpObj B] (hσ : IsDPMorphism κ κ' σ) (hb : t = u ≫ spMap σ) (f : B ⟶ A) [IsMonHom f] : GrpMor κ t A κ' u B σ (𝟙 S) f.left where

theorem **isMor** (m : GrpMor κ t A κ' t' A' σ b φ) : IsMor κ (A.hom ≫ t) κ' (A'.hom ≫ t') σ φ

private theorem **grpMor_isMor_comp** (m : GrpMor κ t A κ' t' A' σ b φ) {Y : Over S'} (f : Y ⟶ A') : IsMor κ (A.hom ≫ t) κ' (Y.hom ≫ t') σ (f.left ≫ φ)

theorem **pullback_mem_primitive** (m : GrpMor κ t A κ' t' A' σ b φ) {c : Cris.H1 κ (A.hom ≫ t)} (hc : c ∈ prim κ t A) : m.isMor.pull1 c ∈ prim κ' t' A'

theorem **coe_pull** (m : GrpMor κ t A κ' t' A' σ b φ) (c : prim κ t A) : (m.pull c : Cris.H1 κ' (A'.hom ≫ t')) = m.isMor.pull1 c

theorem **pull_pull** {τ : R' ↠+∗ R''} {b' : S'' ⟶ S'} {ψ : A''.left ⟶ A'.left} {ρ : R ↠+∗ R''} {b'' : S'' ⟶ S} {χ : A''.left ⟶ A.left} (m : GrpMor κ t A κ' t' A' σ b φ) (m' : GrpMor κ' t' A' κ'' t'' A'' τ b' ψ) (m'' : GrpMor κ t A κ'' t'' A'' ρ b'' χ) (e : ∀ r, τ (σ r) = ρ r) (e' : ψ ≫ φ = χ) (c : prim κ t A) : m'.pull (m.pull c) = m''.pull c

theorem **comp** {τ : R' ↠+∗ R''} {b' : S'' ⟶ S'} {ψ : A''.left ⟶ A'.left} (m : GrpMor κ t A κ' t' A' σ b φ) (m' : GrpMor κ' t' A' κ'' t'' A'' τ b' ψ) : GrpMor κ t A κ'' t'' A'' (τ.comp σ) (b' ≫ b) (ψ ≫ φ) where

theorem **homMor** (f : A ⟶ B) [IsMonHom f] : GrpMor κ t B κ t A (RingHom.id R) (𝟙 S) f.left

theorem **push_apply** (f : A ⟶ B) [IsMonHom f] (h : Crys κ t A) (c : prim κ t B) : push κ t f h c = h (primPull κ t f c)

theorem **push_id** : push κ t (𝟙 A) = LinearMap.id

theorem **push_comp** (f : A ⟶ B) (g : B ⟶ C) [IsMonHom f] [IsMonHom g] : push κ t (f ≫ g) = push κ t g ∘ₗ push κ t f

theorem **push_mul** (f g : A ⟶ B) [IsMonHom f] [IsMonHom g] [IsMonHom (f ∗ g)] : push κ t (f ∗ g) = push κ t f + push κ t g

theorem **push_congr** {f g : A ⟶ B} [IsMonHom f] [IsMonHom g] (e : f = g) : push κ t f = push κ t g

theorem **act_apply** (x : O) [IsMonHom (a.act x)] : act κ t a x = push κ t (a.act x)

theorem **actAlg_tmul** (r : R) (x : O) (h : Crys κ t A) : actAlg κ t a (r ⊗ₜ x) h = r • act κ t a x h

theorem **isScalarTower_tensorModule** : letI

@[simp] theorem **liftSemi_tmul** {M M' : Type*} [AddCommGroup M] [Module R M] [AddCommGroup M'] [Module R' M'] (f : M →ₛₗ[algebraMap R R'] M') (x : R') (m : M) : liftSemi f (x ⊗ₜ m) = x • f m

theorem **GrpMor.IsBC.unique** (hm : m.IsBC) {h : Crys κ t A} {h₁ h₂ : Crys κ' t' A'} (e₁ : m.Corr h h₁) (e₂ : m.Corr h h₂) : h₁ = h₂

theorem **GrpMor.IsBC.exists** (hm : m.IsBC) (h : Crys κ t A) : ∃ h', m.Corr h h'

theorem **GrpMor.IsBC.corr_iff** (hm : m.IsBC) {h : Crys κ t A} {h' : Crys κ' t' A'} : m.Corr h h' ↔ hm.bcMap h = h'

theorem **GrpMor.Corr.push** {κ : DividedPowers I} {t : S ⟶ Sp R} {A B : Over S} [GrpObj A] [GrpObj B] {κ' : DividedPowers I'} {t' : S' ⟶ Sp R'} {A' B' : Over S'} [GrpObj A'] [GrpObj B'] {σ : R →+* R'} {b : S' ⟶ S} {φ : A'.left ⟶ A.left} {ψ : B'.left ⟶ B.left} {m : GrpMor κ t A κ' t' A' σ b φ} (n : GrpMor κ t B κ' t' B' σ b ψ) (f : A ⟶ B) (f' : A' ⟶ B') [IsMonHom f] [IsMonHom f'] (e : φ ≫ f.left = f'.left ≫ ψ) {h : Crys κ t A} {h' : Crys κ' t' A'} (hh : m.Corr h h') : n.Corr (push κ t f h) (push κ' t' f' h')

theorem **mem_Fil** {h : HdR A} : h ∈ Fil A ↔ ∀ (f : Form (A.hom ≫ 𝟙 _)) (hf : Form.cl _ f ∈ primDR A), h ⟨_, hf⟩ = 0

theorem **push_mem_Fil** (f : A ⟶ B) [IsMonHom f] {h : HdR A} (hh : h ∈ Fil A) : push (dp0 R) (𝟙 _) f h ∈ Fil B

theorem **pt_w** {D : Type} [CommRing D] [Algebra R D] (v : Over.mk (spMap (algebraMap R D)) ⟶ A) : v.left ≫ A.hom ≫ 𝟙 _ = spMap (algebraMap R D)

theorem **tanVal_apply** (v : epsR R ⟶ A) (f : Form (A.hom ≫ 𝟙 _)) : tanVal v f = TrivSqZeroExt.snd (R

theorem **tanVal_epsScale** (r : R) (v : epsR R ⟶ A) (f : Form (A.hom ≫ 𝟙 _)) : tanVal (epsScale r ≫ v) f = r ∗ tanVal v f

theorem **tanVal_one** (f : Form (A.hom ≫ 𝟙 _)) : tanVal (1 : epsR R ⟶ A) f = 0

theorem **tanVal_comp** (v : epsR R ⟶ A) (g : A ⟶ B) (m : IsMor (dp0 R) (B.hom ≫ 𝟙 _) (dp0 R) (A.hom ≫ 𝟙 _) (RingHom.id R) g.left) (f : Form (B.hom ≫ 𝟙 _)) : tanVal (v ≫ g) f = tanVal v (m.pullForm f)

theorem **spOver_comp** {D D' D'' : Type} [CommRing D] [Algebra R D] [CommRing D'] [Algebra R D'] [CommRing D''] [Algebra R D''] (φ : D →ₐ[R] D') (ψ : D' →ₐ[R] D'') : spOver ψ ≫ spOver φ = spOver (ψ.comp φ)

theorem **tanVal_mul** {v w : epsR R ⟶ A} (hv : IsTangent v) (hw : IsTangent w) (f : Form (A.hom ≫ 𝟙 _)) : tanVal (v ∗ w) f = tanVal v f + tanVal w f

theorem **isLieImage_one_iff** {h : HdR A} : IsLieImage h 1 ↔ h ∈ Fil A

private theorem **mul_left_eq_lift_comp_mu'** {X : Scheme.{0}} {G : Over X} [GrpObj G] {T : Over X} (x y : T ⟶ G) : (x ∗ y).left = pullback.lift x.left y.left ((Over.w x).trans (Over.w y).symm) ≫ μ[G].left

theorem **IsLieImage.add** {h h' : HdR A} {v v' : epsR R ⟶ A} (e : IsLieImage h v) (e' : IsLieImage h' v') : IsLieImage (h + h') (v ∗ v')

theorem **IsLieImage.smul** {h : HdR A} {v : epsR R ⟶ A} (e : IsLieImage h v) (r : R) : IsLieImage (r • h) (epsScale r ≫ v)

theorem **IsLieImage.push** {h : HdR A} {v : epsR R ⟶ A} (e : IsLieImage h v) (g : A ⟶ B) [IsMonHom g] : IsLieImage (push (dp0 R) (𝟙 _) g h) (v ≫ g)

theorem **IsLieImage.sub_mem** {h h' : HdR A} {v : epsR R ⟶ A} (e : IsLieImage h v) (e' : IsLieImage h' v) : h - h' ∈ Fil A

theorem **LieExact.toLie_eq_iff** (H : LieExact A) (a : OAct O A) {h : HdR A} {v : Serre.Lie a} : H.toLie a h = v ↔ IsLieImage h v.hom

theorem **LieExact.ker_toLie** (H : LieExact A) (a : OAct O A) : LinearMap.ker (H.toLie a) = Fil A

theorem **LieExact.toLie_surjective** (H : LieExact A) (a : OAct O A) : Function.Surjective (H.toLie a)

theorem **LieExact.quotEquiv_mk** (H : LieExact A) (a : OAct O A) (h : HdR A) : H.quotEquiv a (Submodule.Quotient.mk h) = H.toLie a h

theorem **LieExact.toLie_push** (H : LieExact A) (H' : LieExact B) (a : OAct O A) (b : OAct O' B) (g : A ⟶ B) [IsMonHom g] (h : HdR A) : H'.toLie b (push (dp0 R) (𝟙 _) g h) = Serre.Lie.map a b g (H.toLie a h)

theorem **LieExact.toLie_act** (H : LieExact A) (a : OAct O A) (x : O) (h : HdR A) : H.toLie a (act (dp0 R) (𝟙 _) a x h) = x • H.toLie a h

theorem **liftMor** (h : IsGrpPullback A₀ A t i) : GrpMor (dp0 R) (𝟙 _) A κ t A₀ (RingHom.id R) t i

theorem **crysToDR_apply** (h : IsGrpPullback A₀ A t i) (d : Crys κ t A₀) (c : primDR A) : crysToDR κ h d c = d ((liftMor κ h).pull c)

theorem **crysToDR_push** (h : IsGrpPullback A₀ A t i) (k : IsGrpPullback B₀ B t j) (f₀ : A₀ ⟶ B₀) (f : A ⟶ B) [IsMonHom f₀] [IsMonHom f] (e : i ≫ f.left = f₀.left ≫ j) (d : Crys κ t A₀) : crysToDR κ k (push κ t f₀ d) = push (dp0 R) (𝟙 _) f (crysToDR κ h d)

theorem **push_mem_liftFil** (h : IsGrpPullback A₀ A t i) (k : IsGrpPullback B₀ B t j) (f₀ : A₀ ⟶ B₀) (f : A ⟶ B) [IsMonHom f₀] [IsMonHom f] (e : i ≫ f.left = f₀.left ≫ j) {d : Crys κ t A₀} (hd : d ∈ liftFil κ h) : push κ t f₀ d ∈ liftFil κ k

theorem **crysToDR_corr** (h : IsGrpPullback A₀ A t i) {κ' : DividedPowers I'} {t' : S' ⟶ Sp R'} {C₀ : Over S'} [GrpObj C₀] {C : Over (Sp R')} [GrpObj C] {l : C₀.left ⟶ C.left} (h' : IsGrpPullback C₀ C t' l) {σ : R →+* R'} {b₀ : S' ⟶ S} {b : Sp R' ⟶ Sp R} {φ₀ : C₀.left ⟶ A₀.left} {φ : C.left ⟶ A.left} (m : GrpMor κ t A₀ κ' t' C₀ σ b₀ φ₀) (n : GrpMor (dp0 R) (𝟙 _) A (dp0 R') (𝟙 _) C σ b φ) (e : l ≫ φ = φ₀ ≫ i) {d : Crys κ t A₀} {d' : Crys κ' t' C₀} (hd : m.Corr d d') : n.Corr (crysToDR κ h d) (crysToDR κ' h' d')

theorem **IsGM.symm** {h : IsGrpPullback A₀ A t i} {h' : IsGrpPullback A₀ A' t i'} {x : HdR A} {x' : HdR A'} (e : IsGM κ h h' x x') : IsGM κ h' h x' x

theorem **constMor** : GrpMor (dp0 R) (𝟙 _) A (epsDP R) (epsRed R) A (algebraMap R (DualNumber R)) (𝟙 _) (Over.Hom.left (𝟙 A))

theorem **IsKSValue.add** {f f' : HdR A} {v v' : epsR R ⟶ A} (e : IsKSValue h f v) (e' : IsKSValue h f' v') : IsKSValue h (f + f') (v ∗ v')

theorem **IsKSValue.smul** {f : HdR A} {v : epsR R ⟶ A} (e : IsKSValue h f v) (r : R) : IsKSValue h (r • f) (epsScale r ≫ v)

theorem **evalAt_pull** {X X' : Scheme.{0}} (φ : X' ⟶ X) {B : Type} [CommRing B] (x : Sp B ⟶ X') {U : X.Opens} {V : X'.Opens} (hV : V ≤ φ ⁻¹ᵁ U) (h : ⊤ ≤ (x ≫ φ) ⁻¹ᵁ U) (h' : ⊤ ≤ x ⁻¹ᵁ V) (f : Γ(X, U)ˣ) : evalAt (x ≫ φ) U h f.val = evalAt x V h' (res hV (pullU φ U f)).val

theorem **evalAt_pull_g** {X X' : Scheme.{0}} (L : LB X) (φ : X' ⟶ X) {B : Type} [CommRing B] (x : Sp B ⟶ X') (j j' : PUnit → L.ι) (h : ⊤ ≤ (x ≫ φ) ⁻¹ᵁ (L.U (j PUnit.unit) ⊓ L.U (j' PUnit.unit))) (h' : ⊤ ≤ x ⁻¹ᵁ ((L.pull φ).U j ⊓ (L.pull φ).U j')) : evalAt (x ≫ φ) _ h (L.g (j PUnit.unit) (j' PUnit.unit)).val = evalAt x _ h' ((L.pull φ).g j j').val

theorem **isChern_pull** {X X' : Scheme.{0}} {κ : DividedPowers I} {κ' : DividedPowers I'} {s : X ⟶ Sp R} {s' : X' ⟶ Sp R'} {σ : R →+* R'} {φ : X' ⟶ X} (m : IsMor κ s κ' s' σ φ) {L : LB X} {z : Cris.H2 κ s} (hz : IsChern κ s L z) : IsChern κ' s' (L.pull φ) (m.pull2 z)

theorem **wedgeForm_apply** {n : ℕ} (a b : Fin n → prim κ t A) (x y : Crys κ t A) : wedgeForm κ t A a b x y = ∑ k, (x (a k) ∗ y (b k) - x (b k) ∗ y (a k))

theorem **IsPolForm.isAlt** {L : LB A.left} {ψ : LinearMap.BilinForm R (Crys κ t A)} (h : IsPolForm κ t A L ψ) : ψ.IsAlt

theorem **IsPolForm.pull** {L : LB A.left} {ψ : LinearMap.BilinForm R (Crys κ t A)} (h : IsPolForm κ t A L ψ) {κ' : DividedPowers I'} {t' : S' ⟶ Sp R'} {A' : Over S'} [GrpObj A'] {σ : R →+* R'} {b : S' ⟶ S} {φ : A'.left ⟶ A.left} (m : GrpMor κ t A κ' t' A' σ b φ) : ∃ ψ',

IsPolForm κ' t' A' (L.pull φ) ψ' ∧ ∀ x x' y y', m.Corr x x' → m.Corr y y' → ψ' x' y' = σ (ψ x y)

theorem **isPolForm_zero** (L : LB A.left) (h : ∀ i j, L.g i j = 1) : IsPolForm κ t A L 0

theorem **mem_eig** {O : Type} [CommRing O] {a : OAct O A} {τ : O →+* R} {h : Crys κ t A} : h ∈ eig κ t a τ ↔ ∀ x, act κ t a x h = τ x • h

theorem **LieExact.filToDual_toLie** {A : Over (Sp R)} [GrpObj A] [IsCommMonObj A] {O : Type} [CommRing O] (H : LieExact A) (a : OAct O A) (ψ : LinearMap.BilinForm R (HdR A)) (hψ : ∀ f ∈ Fil A, ∀ g ∈ Fil A, ψ f g = 0) (f : Fil A) (x : HdR A) : H.filToDual a ψ hψ f (H.toLie a x) = ψ f x

theorem **primDR_unit** (c : primDR (𝟙_ (Over (Sp R)))) : c = 0

theorem **HdR_unit** (h : HdR (𝟙_ (Over (Sp R)))) : h = 0

theorem **lieExact_unit** : LieExact (𝟙_ (Over (Sp R)))

theorem **allPrim_unit** : AllPrim (dp0 R) (𝟙 _) (𝟙_ (Over (Sp R)))

Development.AbelianVarietyTheory.Definitions.Divisors.IsLatBasisHtLatIffSpanEq

theorem **IsLatBasis.sum_mem_iff** {La : Submodule O U} {w : Fin n → U} (h : IsLatBasis C La w) (a : Fin n → C) : ∑ i, a i • w i ∈ La ↔ ∀ i, a i ∈ (algebraMap O C).range

theorem **isLatBasis_span** {w : Fin n → U} (hw : LinearIndependent C w) : IsLatBasis C (Submodule.span O (Set.range w)) w

theorem **val_vw_zero** : val_vw la p v 0 = ⊤

theorem **val_vw_of_ne** {x : C} (hx : x ≠ 0) : val_vw la p v x = (((vLa la (p : K) : ℝ) * (Real.log (v x) / Real.log (v (p : C))) : ℝ) : EReal)

theorem **val_vw_ne_bot** (x : C) : val_vw la p v x ≠ ⊥

theorem **val_vw_mul** (x y : C) : val_vw la p v (x * y) = val_vw la p v x + val_vw la p v y

theorem **val_vw_of_eq_one** {x : C} (hx : v x = 1) : val_vw la p v x = 0

theorem **val_vw_anti** (h0 : (p : C) ≠ 0) (h1 : v (p : C) < 1) {x y : C} (h : v x ≤ v y) : val_vw la p v y ≤ val_vw la p v x

theorem **val_vw_add** (h0 : (p : C) ≠ 0) (h1 : v (p : C) < 1) (x y : C) : min (val_vw la p v x) (val_vw la p v y) ≤ val_vw la p v (x + y)

theorem **lt_one_of_not_isUnit** (hv : v.Integers O) (hp : ¬ IsUnit (p : O)) : v (p : C) < 1

theorem **exists_log_eq** (hv : v.Integers O) (k : F →+* C) (kO : V →+* O) (hk : ∀ a, algebraMap O C (kO a) = k (algebraMap V F a)) : ∃ r : ℝ, ∀ x : F, x ≠ 0 → Real.log (v (k x)) = (vV V x : ℝ) * r

theorem **val_vw_map** (hv : v.Integers O) (h0 : (p : C) ≠ 0) (hp : ¬ IsUnit (p : O)) {io : K →+* F} {e : ℕ} (ho : OverLa la V io e) (k : F →+* C) (kO : V →+* O) (hk : ∀ a, algebraMap O C (kO a) = k (algebraMap V F a)) (x : F) : val_vw la p v (k x) = nval V e x

theorem **corr_act** {R : Type} [CommRing R] [Algebra O R] {x : specO O R ⟶ P.Gt} {y : specO O R ⟶ P.G} (h : Corr P x y) (a : Λ) : Corr P (x ≫ P.oGt.act a) (y ≫ P.oG.act a)

theorem **spMap_spMap** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

theorem **LieTie.scale** {Y : Scheme.{0}} {ψ : P.A.A.left ⟶ Y} {x : LieG P.Gt} {u : Sp (DualNumber C) ⟶ Y} (h : LieTie P ψ x u) (a : O) : LieTie P ψ (AvDefs.lieScale P.Gt a x) (spMap (epsMul (algebraMap O C a)) ≫ u)

theorem **LieTie.act** {Y : Scheme.{0}} {ψ : P.A.A.left ⟶ Y} {x : LieG P.Gt} {u : Sp (DualNumber C) ⟶ Y} (h : LieTie P ψ x u) (a : Λ) {f : Y ⟶ Y} (hf : (P.oA.act a).left ≫ ψ = ψ ≫ f) : LieTie P ψ (AvDefs.lieAct P.Gt P.oGt a x) (u ≫ f)

theorem **LieTie.unique** {Y S : Scheme.{0}} {ψ : P.A.A.left ⟶ Y} {s : Y ⟶ S} {y : Sp C ⟶ S} (hψ : IsPullback ψ P.A.A.hom s y) {x x' : LieG P.Gt} {u : Sp (DualNumber C) ⟶ Y} (h : LieTie P ψ x u) (h' : LieTie P ψ x' u) : x = x'

theorem **exists_lieTie** {Y : Scheme.{0}} (ψ : P.A.A.left ⟶ Y) (x : LieG P.Gt) : ∃ u, LieTie P ψ x u

theorem **pairV_bLie_eq_evLie** (x : LieG P.Gt) (w : P.V) : P.pairV (P.bLie x) w = evLie P x (P.alC w)

theorem **bLie_mem_eigV** {τ : Λ →+* O} {x : LieG P.Gt} (hx : AvDefs.IsEig P.Gt P.oGt τ x) : P.bLie x ∈ P.eigV τ

theorem **algebraMap_embInt** (hv : v.Integers O) (τ : E →+* ℂ) (a : 𝒪 E) : algebraMap O C (embInt c hv τ a) = algebraMap c.F C (embK c τ a)

theorem **embInt_conj** (hs : ∀ a : 𝒪 E, star a = H.bar a) (hv : v.Integers O) (τ : E →+* ℂ) : embInt c hv (conjEmb τ) = (embInt c hv τ).comp (starRingEnd (𝒪 E))

theorem **PointOver.isGrpPullback** (h : PointOver P x₀ k kO ψ) : IsGrpPullback P.A.A Xo.univ.t.G (spMap k ≫ x₀) ψ

theorem **PointOver.act** (h : PointOver P x₀ k kO ψ) (a : 𝒪 E) : (P.oA.act a).left ≫ ψ = ψ ≫ (Xo.univ.t.act.act a).left

theorem **htLat_psi** (τ : Psi H) : htLat c P τ.1 = P.latTQuot (σ

theorem **htLat_conj** (τ : Psi H) : htLat c P (conjEmb τ.1) = P.latTSub (embInt c P.integers (conjEmb τ.1))

theorem **latTSub_le_htLat** (τ : E →+* ℂ) : P.latTSub (embInt c P.integers τ) ≤ htLat c P τ

theorem **wRec_le_htLat** (τ : E →+* ℂ) : P.wRec (embInt c P.integers τ) ≤ htLat c P τ

theorem **LieTie.isEig** {V F : Type} [CommRing V] [Field F] [Algebra V F] {x₀ : Sp F ⟶ Xo.X} {k : F →+* C} {kO : V →+* O} {ψ : P.A.A.left ⟶ Xo.univ.t.G.left} (hP : PointOver P x₀ k kO ψ) {x : LieG P.Gt} {u : Sp (DualNumber C) ⟶ Xo.univ.t.G.left} (h : LieTie P ψ x u) {τ : E →+* ℂ} (hu : IsEigAt c τ u) : AvDefs.IsEig P.Gt P.oGt (embInt c P.integers τ) x

theorem **IsHTChange.exists_frBor** {ψ : P.A.A.left ⟶ Xo.univ.t.G.left} {z y : FrPt Xo c C} (h : IsHTChange P ψ z y) : ∃ t : Ix Dw → Cˣ, FrBor t z y ∧ v (t none) = 1

theorem **IsHTFrame.x_eq** (h : IsHTFrame p v O y₀ k y) : letI

theorem **IsHTFrame.val_vw_map** (h : IsHTFrame p v O y₀ k y) {la : HeightOneSpectrum (𝒪 c.F)} {e : ℕ} (ho : OverLa la T.V io e) (x : T.Fr) : val_vw la p v (k x) = nval T.V e x

theorem **IsHTFrame.isVal** (h : IsHTFrame p v O y₀ k y) {la : HeightOneSpectrum (𝒪 c.F)} {e : ℕ} (ho : OverLa la T.V io e) : IsVal la fun f : Rul Xo c => val_vw la p v (evR Xo c f y)

theorem **towerPadic** (p : ℕ) [Fact p.Prime] : @IsScalarTower ℤ_[p] 𝒪_ℂ_[p] ℂ_[p] (algPadicInt p).toSMul Algebra.toSMul (algPadicField p).toSMul

theorem **factPadic** (p : ℕ) [Fact p.Prime] : Fact ¬IsUnit (p : 𝒪_ℂ_[p])

theorem **IsHTFrameCp.mem** {p : ℕ} [Fact p.Prime] [Algebra c.F ℂ_[p]] {T : CDVR} {io : c.F →+* T.Fr} {y₀ : IntFr Xo c T io} {k : letI

theorem **htPrimes_eq** (B : ℝ) : htPrimes l B = ↑(((Finset.range (⌊B⌋₊ + 1)).filter Nat.Prime) ∪ (Nat.primeFactors (2 * (discr E).natAbs) ∪ l.M'.primeFactors))

theorem **htPrimes_finite** (B : ℝ) : (htPrimes l B).Finite

theorem **mixVals_of_mem** {B : ℝ} {la : HeightOneSpectrum (𝒪 c.F)} {p : ℕ} (hp : p ∈ htPrimes l B) (h : (p : 𝒪 c.F) ∈ la.asIdeal) : (mixVals Xo c B).1 la = htValSet Xo c la

theorem **mixVals_of_not_mem** {B : ℝ} {la : HeightOneSpectrum (𝒪 c.F)} {p : ℕ} (hp : p.Prime) (hn : p ∉ htPrimes l B) (h : (p : 𝒪 c.F) ∈ la.asIdeal) : (mixVals Xo c B).1 la = valSet Xo c (algCls Xo c) la

Development.AbelianVarietyTheory.Definitions.Divisors.IsLocCompIffEquiv

theorem **SoleTerm.mono** {ι : Type} {C : ι → ℂ} {p q p' q' : ι → Prop} (h : SoleTerm C p q) (hp : ∀ i, p i → p' i) (hq : ∀ i, q' i → q i) : SoleTerm C p' q'

theorem **isLocComp_iff_equiv** {ρc : CuspSub E n} {w : Pl E} {σ : SmIrr (GLw n w)} (h : ρc.IsLocComp w σ) (τ : SmIrr (GLw n w)) : ρc.IsLocComp w τ ↔ Nonempty (τ.ρ.Equiv σ.ρ)

Development.AbelianVarietyTheory.Definitions.Divisors.IsLocalRingTrivSqZeroExt

theorem **exists_frMap** (φ : S →ₐ[c.F] T) (y : FrPt Xo c S) : ∃ z, FrMap φ y z

theorem **frMap_unique** {φ : S →ₐ[c.F] T} {y : FrPt Xo c S} {z z' : FrPt Xo c T} (h : FrMap φ y z) (h' : FrMap φ y z') : z = z'

theorem **frMap_id** (y : FrPt Xo c S) : FrMap (AlgHom.id c.F S) y y

theorem **frMap_comp** {U : Type} [CommRing U] [Algebra c.F U] {φ : S →ₐ[c.F] T} {ψ : T →ₐ[c.F] U} {χ : S →ₐ[c.F] U} {y : FrPt Xo c S} {z : FrPt Xo c T} {w : FrPt Xo c U} (h : FrMap φ y z) (h' : FrMap ψ z w) (e : ∀ a, ψ (φ a) = χ a) : FrMap χ y w

theorem **frMap_of_comp** {U : Type} [CommRing U] [Algebra c.F U] {φ : S →ₐ[c.F] T} {ψ : T →ₐ[c.F] U} {χ : S →ₐ[c.F] U} {y : FrPt Xo c S} {z : FrPt Xo c T} {w : FrPt Xo c U} (h : FrMap φ y z) (h' : FrMap χ y w) (e : ∀ a, ψ (φ a) = χ a) : FrMap ψ z w

theorem **frMap_iff_frChg** {φ : S →ₐ[c.F] T} {g : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) S} {a : S} {y y' : FrPt Xo c S} {z z' : FrPt Xo c T} (h : FrChg g a y y') (hz : FrMap φ y z) : FrMap φ y' z' ↔ FrChg (fun τ => (g τ).map φ) (φ a) z z'

theorem **exists_chart** [IsLocalRing S] (w w' : FrPt Xo c S) (hw : w'.x = w.x) : ∃ (p A : TRing c.F) (j : Sp p.R ⟶ Xo.XF c) (y₀ : FrPt Xo c p.R) (π : p.R →ₐ[c.F] A.R) (X : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A.R) (s : (A.R)ˣ) (u : FrPt Xo c A.R) (a a' : A.R →ₐ[c.F] S), OpenChart j y₀ ∧ UnivFrame y₀ π X s u ∧ FrMap a u w ∧ FrMap a' u w'

theorem **fromTest_of_isLocalRing** [IsLocalRing S] (w : FrPt Xo c S) : FromTest (K

theorem **evR_eq** {f : Rul Xo c} (hf : IsNatural (K

theorem **evR_frMap** [IsLocalRing S] {f : Rul Xo c} (hf : IsNatural (K

theorem **evR_frBor** [IsLocalRing S] {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} (h : FrBor t y y') : evR Xo c f y' = ((∏ i, t i ^ e i : Sˣ) : S) * evR Xo c f y

theorem **exists_frPt** [IsLocalRing S] (x : Sp S ⟶ Xo.X) (hx : x ≫ Xo.s = sOf c S) : ∃ z : FrPt Xo c S, z.x = x

@[simp] theorem **val_tmul** (φ : K₀ →ₐ[c.F] S) (y : FrPt Xo c S) (a : K₀) (f : Rul Xo c) : val φ y (a ⊗ₜ f) = φ a * evR Xo c f y

theorem **val_smul** (φ : K₀ →ₐ[c.F] S) (y : FrPt Xo c S) (a : K₀) (s : RulK Xo c K₀) : val φ y (a • s) = φ a * val φ y s

theorem **tmul_mem_secK** {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) (a : K₀) : a ⊗ₜ f ∈ secK Xo c K₀ e

theorem **secK_induction** {e : Ix Dw → ℤ} {P : RulK Xo c K₀ → Prop} (h0 : P 0) (h1 : ∀ a f, f ∈ secZ Xo c e → P (a ⊗ₜ f)) (h2 : ∀ s s', P s → P s' → P (s + s')) {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) : P s

theorem **val_frMap** [IsLocalRing S] {e : Ix Dw → ℤ} {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) {φ : K₀ →ₐ[c.F] S} {ψ : S →ₐ[c.F] T} {χ : K₀ →ₐ[c.F] T} (hχ : ∀ a, ψ (φ a) = χ a) {y : FrPt Xo c S} {z : FrPt Xo c T} (h : FrMap ψ y z) : val χ z s = ψ (val φ y s)

theorem **val_frBor** [IsLocalRing S] {e : Ix Dw → ℤ} {s : RulK Xo c K₀} (hs : s ∈ secK Xo c K₀ e) (φ : K₀ →ₐ[c.F] S) {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} (h : FrBor t y y') : val φ y' s = ((∏ i, t i ^ e i : Sˣ) : S) * val φ y s

theorem **isUnit_det_unip** (N : FIx Dw → M) (σ : E →+* ℂ) : IsUnit (unip K N σ).det

theorem **isUnit_unip₀** (N : FIx Dw → M) : IsUnit (unip₀ K N)

@[simp] theorem **fst_dir** {ι : Type} [Fintype ι] (m : ι → K) (z : TrivSqZeroExt K (ι → K)) : (dir c m z).fst = z.fst

@[simp] theorem **snd_dir** {ι : Type} [Fintype ι] (m : ι → K) (z : TrivSqZeroExt K (ι → K)) : (dir c m z).snd = ∑ k, z.snd k * m k

theorem **exists_lift** {ι : Type} (y : FrPt Xo c K) (X : Sp (TrivSqZeroExt K (ι → K)) ⟶ Xo.X) (hX : X ≫ Xo.s = sOf c _) (hy : spMap (fstHom c.F K (ι → K)).toRingHom ≫ X = y.x) : ∃ W : FrPt Xo c (TrivSqZeroExt K (ι → K)), W.x = X ∧ FrMap (fstHom c.F K (ι → K)) W y

Development.AbelianVarietyTheory.Definitions.Divisors.IsLocallyConstantLinOm

theorem **algebraMap_trace** (h : IsQuad c δ) (z : A) : algebraMap k A (Algebra.trace k A z) = z + c z

theorem **trace_conj** (h : IsQuad c δ) (z : A) : Algebra.trace k A (c z) = Algebra.trace k A z

theorem **eq_zero_of_trace** (h : IsQuad c δ) {z : A} (hz : ∀ a, Algebra.trace k A (a * z) = 0) : z = 0

theorem **trB_apply** (u v : β → A) : trB c JB u v = -Algebra.trace k A (u ᵥ* JB *ᵥ fun i => c (v i))

theorem **trB_mulVec** {γ γ' : Type} [Fintype γ] [Fintype γ'] (M : Matrix β γ A) (N : Matrix β γ' A) (u : γ → A) (v : γ' → A) : trB c JB (M *ᵥ u) (N *ᵥ v) = -Algebra.trace k A (u ᵥ* (Mᵀ * JB * N.map c) *ᵥ fun i => c (v i))

theorem **symp_eq** (hq : IsQuad c δ) (hJ : ∀ i j, c (JB j i) = -JB i j) (x y : Matrix (Fin 2) β A) : WeilM.symp c (-1) (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) A) JB x y = trB c JB (x 0) (y 1) - trB c JB (y 0) (x 1)

theorem **exists_coord** (hq : IsQuad c δ) (hJ : ∀ i j, c (JB j i) = -JB i j) (hu : IsUnit JB.det) : ∃ (d : ℕ) (e : (β → A) ≃ₗ[k] (Fin d → k)) (co : Darboux (WeilM.symp c (-1) (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) A) JB) (Fin d)), ∀ x, (co.e x).1 = e (x 0) ∧ ∀ u, e u ᵥ* (co.e x).2 = trB c JB u (x 1)

theorem **actC_apply** (e : (β → A) ≃ₗ[k] (κ → k)) (g : GL β A) (u : β → A) : actC e g (e u) = e (g.val *ᵥ u)

theorem **actC_actC** (e : (β → A) ≃ₗ[k] (κ → k)) (g g' : GL β A) (x : κ → k) : actC e g (actC e g' x) = actC e (g * g') x

theorem **actC_one** (e : (β → A) ≃ₗ[k] (κ → k)) (x : κ → k) : actC e 1 x = x

theorem **linOm_apply** (e : (β → A) ≃ₗ[k] (κ → k)) (h : UG c JB) (φ : (κ → k) → ℂ) (x : κ → k) : linOm c JB e h φ x = φ (actC e (h : GL β A)⁻¹ x)

theorem **linOm_impl** [Fintype κ] [Invertible (2 : k)] (ψ : AddChar k ℂ) (e : (β → A) ≃ₗ[k] (κ → k)) (co : Darboux (WeilM.symp c (-1) (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) A) JB) κ) (hco : ∀ x, (co.e x).1 = e (x 0) ∧ ∀ u, e u ·ᵥ (co.e x).2 = trB c JB u (x 1)) (h : UG c JB) (w : Matrix (Fin 2) β A) (φ : (κ → k) → ℂ) : linOm c JB e h (co.rho ψ w φ) = co.rho ψ (actB c h w) (linOm c JB e h φ)

theorem **linOm_mem** (e : (β → A) ≃ₗ[k] (κ → k)) (h : UG c JB) {φ : (κ → k) → ℂ} (hφ : φ ∈ SB (κ → k)) : linOm c JB e h φ ∈ SB (κ → k)

theorem **isLocallyConstant_linOm** [TopologicalSpace A] [IsModuleTopology k A] (hq : IsQuad c δ) (e : (β → A) ≃ₗ[k] (κ → k)) {φ : (κ → k) → ℂ} (hφ : φ ∈ SB (κ → k)) : IsLocallyConstant fun h : UG c JB => linOm c JB e h φ

theorem **exists_lagr** {α : Type} [Fintype α] [DecidableEq α] {ξ : A} {JA : Matrix α α A} (co : Darboux (WeilM.symp c ξ JA JB) κ) {C : Matrix β ν A} (h : IsLagr c JB C) : ∃ Λ : Submodule k ((κ → k) × (κ → k)), (∀ w, w ∈ Λ ↔ ∃ z : Matrix α ν A, co.e (z ∗ Cᵀ) = w) ∧ (∀ a ∈ Λ, ∀ b ∈ Λ, a.1 ·ᵥ b.2 = b.1 ·ᵥ a.2) ∧ Module.finrank k Λ = Fintype.card κ

theorem **siegelRule_linOm** (hq : IsQuad c δ) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (e : (β → A) ≃ₗ[k] (κ → k)) (co : Darboux (WeilM.symp c (-1) (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) A) JB) κ) (hco : ∀ x, (co.e x).1 = e (x 0) ∧ ∀ u, e u ·ᵥ (co.e x).2 = trB c JB u (x 1)) {C : Matrix β ν A} (hL : IsLagr c JB C) : SiegelRule c (sbClass k) ψ (-1) _ _ co 1 C (linOm c JB e) ∧ ∃ l : ((κ → k) → ℂ) →ₗ[ℂ] ℂ, (∀ z : Matrix (Fin 2) ν A, ∀ φ ∈ SB (κ → k), l (co.rho ψ (z ∗ Cᵀ) φ) = l φ) ∧ ∃ φ ∈ SB (κ → k), l φ ≠ 0

theorem **exists_model** {F E : Type} [Field F] [Field E] [Algebra F E] {k : Type} [Field k] [Algebra F k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [Invertible (2 : k)] {σ : E →ₐ[F] E} {δ : k ⊗[F] E} (hq : IsQuad (conjR k σ) δ) {ψ : AddChar k ℂ} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) {n : ℕ} {Q : Matrix (Fin (n + n)) (Fin (n + n)) E} (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : ∃ (d : ℕ) (co : LocDbx σ (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) (k ⊗[F] E)) Q (Fin d)) (om : UG (conjR k σ) (JR k Q) →* Module.End ℂ ((Fin d → k) → ℂ)) (l : ((Fin d → k) → ℂ) →ₗ[ℂ] ℂ), IsLocModel σ (sbClass k) ψ _ 1 co om ∧ IsYAdm σ (sbClass k) ψ co l ∧ (∃ φ ∈ SB (Fin d → k), l φ ≠ 0) ∧ ∀ φ ∈ SB (Fin d → k), IsLocallyConstant fun g => om g φ

Development.AbelianVarietyTheory.Definitions.Divisors.IsLocallyConstantSplitOm

theorem **isLocallyConstant_orbit** {G V : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [AddCommGroup V] [Module ℂ V] (om : G →* Module.End ℂ V) {v : V} (h : ∀ᶠ g in 𝒩 (1 : G), om g v = v) : IsLocallyConstant fun g => om g v

theorem **eventually_comp_eq** {P V : Type} [TopologicalSpace P] [TopologicalSpace V] {p₀ : P} (f g : P → V → V) (hf : Continuous fun q : P × V => f q.1 q.2) (hg : Continuous fun q : P × V => g q.1 q.2) (f0 : ∀ v, f p₀ v = v) (g0 : ∀ v, g p₀ v = v) (hgf : ∀ p v, g p (f p v) = v) {φ : V → ℂ} (hφ : φ ∈ SB V) : ∀ᶠ p in 𝒩 p₀, ∀ v, φ (f p v) = φ v

theorem **eventually_char_eq_one** (μ : Fˣ →* Circle) (hμ : Continuous μ) : ∀ᶠ t in 𝒩 (1 : Fˣ), μ t = 1 ∧ distribHaarChar F t = 1

theorem **isLocallyConstant_splitOm** (χ : (F × F)ˣ →* Circle) (hχ : Continuous χ) (J : Matrix β β (F × F)) (σ : κ ≃ α × β) {φ : (κ → F) → ℂ} (hφ : φ ∈ SB (κ → F)) : IsLocallyConstant fun h => splitOm (distribHaarChar F) χ J σ h φ

theorem **isUnitaryOn_splitOm** (χ : (F × F)ˣ →* Circle) (J : Matrix β β (F × F)) (σ : κ ≃ α × β) : IsUnitaryOn (sbClass F) (splitOm (distribHaarChar F) χ J σ)

theorem **isUSplitting_splitOm** (ψ : AddChar F ℂ) (hA : IsHerm (swapA F) JA) (hB : IsHerm (swapA F) JB) (σ : κ ≃ α × β) (χ : (F × F)ˣ →* Circle) (hχ : Continuous χ) : IsUSplitting (sbClass F) ψ (splitDarboux hA hB σ) (actB (swapA F)) (splitOm (distribHaarChar F) χ JB σ)

theorem **isUWeilSplit_split** (ψ : AddChar F ℂ) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (hA : IsHerm (swapA F) JA) (hB : IsHerm (swapA F) JB) (σ : κ ≃ α × β) (χ : (F × F)ˣ →* Circle) (hc : Continuous χ) (hχ : ResEtaPow (⊥ : Subgroup Fˣ) χ (Fintype.card α)) : IsUWeilSplit (swapA F) (sbClass F) ψ (1, -1) JA (dbl JB) (splitDarboux hA hB.dbl (dblNum σ)) χ (diagCols β) (splitOm (distribHaarChar F) χ (dbl JB) (dblNum σ))

Development.AbelianVarietyTheory.Definitions.Divisors.IsMemberPOfIsMember

theorem **exists_gSet** : ∃ X₁ : GSet, X₁.S.g = 1

theorem **exists_zhAbove** (h : GlobalTheory.Published.SemistableReductionSt) : ∃ X₁ : GSet, X₁.S.g = 1 ∧ ZHAbove X₁ X₁.M (𝟙 _)

theorem **nonempty_of_regular** (hact : IsAct X₁ act) (hz : X₁.S.R ≤ X₁.M.O z₁) (hV : IsCN V) (ev : X₁.M.Z.presheaf.stalk z₁ →+* V) (hev : IsLocalHom ev) (L : Type) [AddCommGroup L] [Subsingleton L] (hL : I → L → L) : Nonempty (GMem X₁ z₁ act V ev L (fun _ => 1) hL)

theorem **exists_gMem** (X₁ : GSet) (act : I → (X₁.ρ.P.A → X₁.ρ.P.A)) (h : IsAct X₁ act) : ∃ (z₁ : X₁.M.Z) (ev : X₁.M.Z.presheaf.stalk z₁ →+* X₁.Ω₁), Nonempty (GMem X₁ z₁ act X₁.Ω₁ ev PUnit (fun _ => 1) (fun _ x => x))

theorem **d_pos** (T : Tower X₁ act) : 0 < T.d

theorem **exists_all** : ∃ (X₁ : GSet) (act : Bool → (X₁.ρ.P.A → X₁.ρ.P.A)), X₁.S.g = 1 ∧ act true = 𝟙 _ ∧ act false = (𝟙 _)⁻¹ ∧ IsAct X₁ act ∧ Nonempty (Tower X₁ act) ∧ ∃ (z₁ : X₁.M.Z) (ev : X₁.M.Z.presheaf.stalk z₁ →+* X₁.Ω₁), Nonempty (GMem X₁ z₁ act X₁.Ω₁ ev PUnit (fun _ => 1) (fun _ x => x))

theorem **mo_ne_zero** (m : GMem X₁ z₁ act V ev L mo hL) (mu : L) : mo mu ≠ 0

theorem **mo_mem** (m : GMem X₁ z₁ act V ev L mo hL) {mu : L} (h : mu ≠ 0) : mo mu ∈ maximalIdeal V

theorem **char_eq** (m : GMem X₁ z₁ act V ev L mo hL) (a : I) (E : EndU m.U.toUnifPre) (hE : E.aA.left ≫ m.φ = m.φ ≫ (act a).left) (mu : L) : E.h (m.e₈ mu) = m.e₈ (hL a mu)

theorem **char_eq_star** (m : GMem X₁ z₁ act V ev L mo hL) {O : Type} [CommRing O] [StarRing O] [Module O m.D.toPre.X] (ιD : m.D.toPre.OAct O) (ιA : m.A.OAct O) (ιG : RingAct m.G O) (lin : m.U.toUnifPre.OLinear O ιD ιA ιG) (a : I) (b : O) (h : (ιA.act b).left ≫ m.φ = m.φ ≫ (act a).left) (mu : L) : m.e₈ (hL a mu) = star b • m.e₈ mu

theorem **terms_sum** (m : GMem X₁ z₁ act V ev L mo hL) {x₀ : X₁.S.Gam} {C : Type} {lab : C → X₁.S.Gam} {T : L → C → V} (h : m.Terms x₀ lab T) (c : C) : SumF (maximalIdeal V) (FractionRing V) (fun χ => algebraMap V (FractionRing V) (T (m.e₈.symm χ) c) * m.yK (X₁.S.th x₀, X₁.S.th_mem x₀)) (m.yK (X₁.S.th (lab c), X₁.S.th_mem _))

theorem **isMemberP_of_isMember** {M : Scheme.{0}} {z₁ : M} (ζ : Sp (M.presheaf.stalk z₁) → pullback C.z (spMap C.b)) {N₁ : ℕ} {Ψ : VSys (H.bar : 𝒪 E → 𝒪 E) (min Dw.p Dw.q) !![(1 : ℤ)] N₁ (M.presheaf.stalk z₁)} {φ : InvFun (𝒪 E)} {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] {Cm : GlobalTheory.UChart.MonoFam Xo c φ V} {ξ : Sp V → pullback C.z (spMap C.b)} {y : Ψ.Y} {ρ : V →+* Ψ.V y} {e : Ψ.X y ≃ₗ[𝒪 E] Cm.P.D.toPre.X} {h : Fin (Ψ.m y) ≃ Fin Cm.m} (hm : GlobalTheory.USep.IsMember ζ Ψ Cm ξ y ρ e h) : IsMemberP ζ Ψ.pre N₁ Cm ξ y ρ e h

theorem **hasDiv_of_uLink** {act : 𝒪 E → (X₁.ρ.P.A → X₁.ρ.P.A)} {u : Sp (ratHull X₁.S.R) → Xo.XF c} {π : X₁.M.Z → C.Z} (h : ULink C X₁ act u π) : X₁.M.HasDiv X₁.S.th

theorem **psi_of_isMonomial** {R : Type} [CommRing R] [IsDomain R] {Dg : DegPre R} {m : ℕ} {s r : Fin m → R} {β : Fin m → LinearMap.BilinForm ℤ Dg.X} (h : IsMonomial Dg s β) (hr : ∀ k, r k ^ 2 = s k) (mu : Fin 1 → Dg.X) : ∃ u : Rˣ, Formalism.L2.Pj.PsiCoord Dg (mu 0) (u ∗ mono !![(1 : ℤ)] r β mu)

theorem **subsingleton_of_unit** (m : GMem X₁ z₁ act V ev L mo hL) (h : ∀ mu, IsUnit (mo mu)) : Subsingleton L

theorem **hL_add** (m : GMem X₁ z₁ act V ev L mo hL) (a : I) (mu nu : L) : hL a (mu + nu) = hL a mu + hL a nu

theorem **n_eq** (T : Tower X₁ act) : T.X₂.ρ.n = X₁.ρ.n

theorem **hasTraitsOn_mono** {M : Scheme.{0}} {z₁ : M} (ζ : Sp (M.presheaf.stalk z₁) ⟶ pullback C.z (spMap C.b)) (Φ : VPre (H.bar : 𝒪 E →ₗ 𝒪 E) (min Dw.p Dw.q) !![(1 : ℤ)] (M.presheaf.stalk z₁)) {U U' : M.Opens} (h : U ≤ U') (h' : HasTraitsOn ζ Φ U') : HasTraitsOn ζ Φ U

theorem **char_eq_smul** {act : 𝒪 E →ₗ (X₁.ρ.P.A ⟶ X₁.ρ.P.A)} {hL : 𝒪 E →ₗ L →ₗ L} (m : GMem X₁ z₁ act V ev L mo hL) [Module (𝒪 E) m.D.toPre.X] (ιD : letI

Development.AbelianVarietyTheory.Definitions.Divisors.IsMetricFrOfIsMetricOn

theorem **isMetricFr_of_isMetricOn** {E : Fr (ratOf X) m 0} {H : Mat m X.Fn →ₗ CxPts X.str σ →ₗ Mat m ℂ} (h : IsMetricOn E ⊤ H) : IsMetricFr E H

Development.AbelianVarietyTheory.Definitions.Divisors.IsMonomialTauOfTauIn

theorem **exists_tauIn** {Q : Data V} {N : ℕ} {s : Fin N → V} {ε : Fin N → LinearMap.BilinForm ℤ Q.X} (h : Q.IsMonomialTau s ε) : ∃ τ₀ : Gens Q, TauIn Q τ₀ s ε

theorem **isMonomialTau_of_tauIn** {Q : Data V} {τ₀ : Gens Q} {N : ℕ} {s : Fin N → V} {ε : Fin N → LinearMap.BilinForm ℤ Q.X} (h : TauIn Q τ₀ s ε) : Q.IsMonomialTau s ε

theorem **tate_tauIn** (k : ℤ) : ∃ τ₀ : Gens (Wit.tate k), TauIn (Wit.tate k) τ₀ ![PowerSeries.X] ![k • mulForm]

theorem **defOver_top** {K : Type} [Field K] {Y : Scheme.{0}} (P : Sp K ⟶ Y) : DefOver (⊤ : Subfield K) P

Development.AbelianVarietyTheory.Definitions.Divisors.IsMumfordFamOfSubsingleton

theorem **IsMumfordFam.isCN** [IsLocalRing R] {Q : Data R} (h : IsMumfordFam F y α₁ Q) : IsCN R

theorem **IsMumfordFam.isPos** [IsLocalRing R] {Q : Data R} (h : IsMumfordFam F y α₁ Q) : Q.IsPos (maximalIdeal R)

theorem **IsMumfordFam.unif** [IsLocalRing R] {Q : Data R} (h : IsMumfordFam F y α₁ Q) : ∀ (R' : Type) [CommRing R'] [IsDomain R'] [IsLocalRing R'] (ρ : R →+* R'), IsCN R' → Function.Injective ρ → ρ.Finite → ∀ (D : DegData R' (maximalIdeal R')) (ε : D.toPre.X ≃+ Q.X), Refines Q ρ D.toPre ε → ∃ (G' : Over (Sp R')) (_ : GrpObj G') (_ : IsCommMonObj G') (π : G'.left ⟶ G₁.left) (A : PolAV (Sp (FractionRing R'))) (U : Unif (maximalIdeal R') D A G'), IsUnifAlong F y α₁ ρ π U

theorem **exists_genericMap** {ρ : R →+* R'} (hρ : Function.Injective ρ) : ∃ κ : Sp (FractionRing R') ⟶ Sp (FractionRing R), κ ≫ jK R = jK R' ≫ spMap ρ

theorem **ne_zero_of_isRelParam** {o : Type} [CommRing o] [IsLocalRing R] {b : o →+* R} {m : ℕ} {t : Fin m → R} (h : IsRelParam b t) (i : Fin m) : t i ≠ 0

theorem **lbIso_pull_comp** {X Y Z : Scheme.{0}} {L : LB X} {M : LB Y} {a : Y ⟶ X} (h : LBIso M (L.pull a)) (j : Z ⟶ Y) {a' : Z ⟶ X} (e : a' = j ≫ a) : LBIso (M.pull j) (L.pull a')

theorem **isSquare_comp** {S S' S'' : Scheme.{0}} {G : Over S} {G' : Over S'} {G'' : Over S''} {φ : G'.left ⟶ G.left} {φ₂ : (G' ⊗ G').left ⟶ (G ⊗ G).left} {ψ : G''.left ⟶ G'.left} {ψ₂ : (G'' ⊗ G'').left ⟶ (G' ⊗ G').left} (h' : IsSquare ψ ψ₂) (h : IsSquare φ φ₂) : IsSquare (ψ ≫ φ) (ψ₂ ≫ φ₂)

theorem **IsUnifAlong.isUnifOf** (h₁ : IsGrpPullback G₁ G ξ π₁) (hy : y ≫ j = jK R ≫ ξ) (hα : α₁ ≫ e = pullback.fst G₁.hom (jK R) ≫ π₁) {R' : Type} [CommRing R'] [IsDomain R'] {I : Ideal R'} {ρ : R →+* R'} {G' : Over (Sp R')} [GrpObj G'] [IsCommMonObj G'] {π : G'.left ⟶ G₁.left} {D : DegData R' I} {A : PolAV (Sp (FractionRing R'))} {U : Unif I D A G'} (h : IsUnifAlong F y α₁ ρ π U) : IsUnifOf F j e (spMap ρ ≫ ξ) (π ≫ π₁) U

theorem **IsMumfordFam.isMumfordOf** [IsLocalRing R] (h₁ : IsGrpPullback G₁ G ξ π₁) (hy : y ≫ j = jK R ≫ ξ) (hα : α₁ ≫ e = pullback.fst G₁.hom (jK R) ≫ π₁) {Q : Data R} (h : IsMumfordFam F y α₁ Q) : IsMumfordOf F j e ξ Q

theorem **IsGenFibre.exists_isGrpPullback** [IsNoetherianRing R] [IsIntegrallyClosed R] (hG : IsSemiAbelian G) (he : IsGrpPullback F.univ.A G j e) (hG₁ : IsSemiAbelian G₁) (hy : y ≫ j = jK R ≫ ξ) (h : IsGenFibre F y α₁) : ∃ π₁ : G₁.left ⟶ G.left, IsGrpPullback G₁ G ξ π₁ ∧ α₁ ≫ e = pullback.fst G₁.hom (jK R) ≫ π₁

theorem **PreChart.hasPolChart** (hG : IsSemiAbelian G) (he : IsGrpPullback F.univ.A G j e) (h : PreChart F s j φ BD d x) : HasPolChart F s j e φ BD d x

Development.AbelianVarietyTheory.Definitions.Divisors.IsNefHermitianTwistingSheaf

theorem **ratVal_add** {σ : k →+* ℂ} {y : CxPts p σ} {f g : X.functionField} (hf : RegAt (under y) f) (hg : RegAt (under y) g) : ratVal (f + g) y = ratVal f y + ratVal g y

theorem **ratVal_intCast_mul** {σ : k →+* ℂ} {y : CxPts p σ} {f : X.functionField} (hf : RegAt (under y) f) (n : ℤ) : RegAt (under y) ((n : X.functionField) * f) ∧ ratVal ((n : X.functionField) * f) y = n * ratVal f y

theorem **isBallFam_Bl** (P : Pair p) (a b : ℤ) : IsBallFam (P.H a b) (P.Bl a b) where

theorem **div_eq** {B : LB X} {r : B.ι → X.functionFieldˣ} (hr : IsRatSec B r) {i j : B.ι} (h : genericPoint X ∈ B.U i ⊓ B.U j) : (r i : X.functionField) / r j = fnOf _ h (B.g i j : Γ(X, B.U i ⊓ B.U j))

theorem **rep_fst** (P : Pair p) : P.Rep 1 0 P.L

theorem **unitAt_zpow** {x : X} {f : X.functionField} (h : UnitAt x f) (a : ℤ) : UnitAt x (f ^ a)

theorem **loc_dbl** (t : X.functionField) (a b : ℤ) (hab : a + b = 1) (i j : A.B.ι) : (dbl A u hu).loc a b t i j = t * ratOfSec u hu j * ((ratOfSec u hu i : X.functionField) / ratOfSec u hu j) ^ a

theorem **unitAt_div** {x : X} {i j : A.B.ι} (hi : x ∈ A.B.U i) (hj : x ∈ A.B.U j) : UnitAt x ((ratOfSec u hu i : X.functionField) / ratOfSec u hu j)

theorem **mem_Bl** {v : A.B.Sec} (hv : v ∈ A.smallSec) {t : X.functionField} (ht : ∀ i (h : genericPoint X ∈ A.B.U i), t * ratOfSec u hu i = fnOf _ h (v.f i)) (a b : ℤ) (hab : a + b = 1) : t ∈ (dbl A u hu).Bl a b 1

theorem **good_dbl** {D : ℕ} (hA : IsAmpleLB A.B) (hN : A.IsNef D) (hs : u ∈ A.smallSec) {v : A.B.Sec} (hv : v ∈ A.smallSec) (hv0 : genericPoint X ∈ nonzeroLocus v) (hc : ∀ x, x ∈ nonzeroLocus u ∨ x ∈ nonzeroLocus v) : ∃ t, (∀ i (h : genericPoint X ∈ A.B.U i), t * ratOfSec u hu i = fnOf _ h (v.f i)) ∧ (dbl A u hu).Good D 2 1 t

theorem **not_good_dbl** {D : ℕ} (h0 : nonzeroLocus u ≠ ⊤) : ¬ (dbl A u hu).Good D 2 1 1

theorem **isNef_of_small** {D : ℕ} (hX : IsArithVar p D) (L : HermLB p) (hs : L.SemiPos D) (hf : ArInt.NefFib L) (hg : ∀ x : X, ∃ u ∈ L.smallSec, x ∈ nonzeroLocus u) : L.IsNef D where

theorem **isNef_hermitianTwistingSheaf** (N : ℕ) : (hermitianTwistingSheaf ℤ N).IsNef N

theorem **gen_mem_coord** (a : Fin 2) : genericPoint (PSpace ℤ 1) ∈ nonzeroLocus (coord ℤ 1 a)

theorem **coord_ne_top** : nonzeroLocus (coord ℤ 1 1) ≠ ⊤

theorem **exists_good** : IsArithVar (PSpace.str ℤ 1) 1 ∧ ∃ (P : Pair (PSpace.str ℤ 1)) (t : (PSpace ℤ 1).functionField), P.L = hermitianTwistingSheaf ℤ 1 ∧ P.A = hermitianTwistingSheaf ℤ 1 ∧ P.Good 1 2 1 t ∧ ¬ P.Good 1 2 1 1

theorem **exists_int_of_regAt** {f : (Sp ℤ).functionField} (h : ∀ x, RegAt x f) : ∃ n : ℤ, f = n

theorem **ratVal_int** {σ : k →+* ℂ} (y : CxPts p σ) (n : ℤ) : ratVal (n : X.functionField) y = n

theorem **nu_int** (a b : ℤ) (σ) (y) (n : ℤ) : pair.nu a b σ y n = |(n : ℝ)| * Real.exp (-(b * Real.log 3))

theorem **Bl_one** (a : ℤ) : pair.Bl a 1 1 = (fun n : ℤ => (n : (Sp ℤ).functionField)) '' Set.Icc (-3) 3

theorem **lc_one** (a : ℤ) : pair.lc a 1 1 = Real.log 7

theorem **rep_iff** (a b : ℤ) (t : ℝ) : pair.Rep a b (ArithBase.trivShift t) ↔ t = b * Real.log 3

theorem **not_good_two_two** : ¬ pair.Good 0 1 (2 : ℕ) (2 : ℕ)

theorem **nu_sq_of_rep** (P : Pair p) {a b : ℤ} {M : HermLB p} (h : P.Rep a b M) : ∃ ρ : M.B.ι → X.functionFieldˣ, IsRatSec M.B ρ ∧ ∀ (v : M.B.Sec) (f : X.functionField), (∀ c (hc : genericPoint X ∈ M.B.U c), f * ρ c = fnOf _ hc (v.f c)) → f ∈ P.H a b ∧ ∀ σ (y : CxPts p σ), P.nu a b σ y f ^ 2 = (M.φ σ).nsq (secVal p σ v) y

theorem **not_isBallFam_discrete** : ¬ IsBallFam (⊤ : AddSubgroup ℤ) fun r => {x | if x = 0 then 0 ≤ r else 1 ≤ r}

Development.AbelianVarietyTheory.Definitions.Divisors.IsNilDPOfCube

theorem **isDPNilpotent_of_cube** (δ : DividedPowers J) (h3 : ∀ a ∈ J, ∀ b ∈ J, ∀ c ∈ J, a * b * c = 0) (h2 : ∀ a ∈ J, ∀ c ∈ J, δ.dpow 2 a * c = 0) (h : ∀ n, 3 ≤ n → ∀ a ∈ J, δ.dpow n a = 0) : IsNilDP δ

@[ext] theorem **ext'** {x y : Bi R B} (h0 : x.1 = y.1) (h1 : x.2.1 = y.2.1) (h2 : x.2.2.1 = y.2.2.1) (h3 : x.2.2.2 = y.2.2.2) : x = y

@[simp] theorem **mk_0** : (mk b α β η).1 = b

@[simp] theorem **mul_0** : (x * y).1 = x.1 * y.1

@[simp] theorem **add_0** : (x + y).1 = x.1 + y.1

@[simp] theorem **sub_0** : (x - y).1 = x.1 - y.1

@[simp] theorem **smul_0** : (r • x).1 = r • x.1

@[simp] theorem **zero_0** : (0 : Bi R B).1 = 0

@[simp] theorem **one_0** : (1 : Bi R B).1 = 1

@[simp] theorem **mk_1** : (mk b α β η).2.1 = α

@[simp] theorem **mul_1** : (x * y).2.1 = x.1 • y.2.1 + y.1 • x.2.1

@[simp] theorem **add_1** : (x + y).2.1 = x.2.1 + y.2.1

@[simp] theorem **sub_1** : (x - y).2.1 = x.2.1 - y.2.1

@[simp] theorem **smul_1** : (r • x).2.1 = r • x.2.1

@[simp] theorem **zero_1** : (0 : Bi R B).2.1 = 0

@[simp] theorem **one_1** : (1 : Bi R B).2.1 = 0

@[simp] theorem **mk_2** : (mk b α β η).2.2.1 = β

@[simp] theorem **mul_2** : (x * y).2.2.1 = x.1 • y.2.2.1 + y.1 • x.2.2.1

@[simp] theorem **add_2** : (x + y).2.2.1 = x.2.2.1 + y.2.2.1

@[simp] theorem **sub_2** : (x - y).2.2.1 = x.2.2.1 - y.2.2.1

@[simp] theorem **smul_2** : (r • x).2.2.1 = r • x.2.2.1

@[simp] theorem **zero_2** : (0 : Bi R B).2.2.1 = 0

@[simp] theorem **one_2** : (1 : Bi R B).2.2.1 = 0

@[simp] theorem **mk_3** : (mk b α β η).2.2.2 = η

@[simp] theorem **mul_3** : (x * y).2.2.2 = x.1 • y.2.2.2 + y.1 • x.2.2.2 + wedge R B x.2.1 y.2.2.1 + wedge R B y.2.1 x.2.2.1

@[simp] theorem **add_3** : (x + y).2.2.2 = x.2.2.2 + y.2.2.2

@[simp] theorem **sub_3** : (x - y).2.2.2 = x.2.2.2 - y.2.2.2

@[simp] theorem **smul_3** : (r • x).2.2.2 = r • x.2.2.2

@[simp] theorem **zero_3** : (0 : Bi R B).2.2.2 = 0

@[simp] theorem **one_3** : (1 : Bi R B).2.2.2 = 0

theorem **algebraMap_eq** (r : R) : algebraMap R (Bi R B) r = mk (algebraMap R B r) 0 0 0

theorem **mem_ker** {x : Bi R B} : x ∈ RingHom.ker π ↔ x.1 = 0

theorem **dp_two** {x : Bi R B} (hx : x.1 = 0) : dp.dpow 2 x = sq x

theorem **dp_high** {n : ℕ} (hn : 3 ≤ n) (x : Bi R B) : dp.dpow n x = 0

theorem **isDPNilpotent_dp** : IsNilDP (dp (R

theorem **dOne_connForm** (f : Z1 (O (dp0 R) s)) (x : Sp B ⟶ X) (hx : x ≫ s = spMap (algebraMap R B)) : dOne R B (connForm f x hx) = 0

Development.AbelianVarietyTheory.Definitions.Divisors.IsNoetherianRingCompletedStalkSiegelFibrePower

theorem **siegPB_siegPull** (c : SiegPre h M ζ s) (f : S' ⟶ S) (s' : S' ⟶ Sp B) (hs : f ≫ s = s') : SiegPB (siegPull c f s' hs) c f

theorem **principal_siegPull** (c : SiegPre h M ζ s) (f : S' ⟶ S) (s' : S' ⟶ Sp B) (hs : f ≫ s = s') (hc : c.Principal) : (siegPull c f s' hs).Principal

theorem **lift_hom** {T : Scheme.{0}} (u : T ⟶ F.X) {r : ℕ} (p : Fin r → (T ⟶ F.univ.A.left)) (hp : ∀ i, p i ≫ F.univ.A.hom = u) : lift u p hp ≫ (powA F r).hom = u

theorem **lift_coord** {T : Scheme.{0}} (u : T ⟶ F.X) {r : ℕ} (p : Fin r → (T ⟶ F.univ.A.left)) (hp : ∀ i, p i ≫ F.univ.A.hom = u) (i : Fin r) : lift u p hp ≫ (coord F i).left = p i

theorem **hom_ext** {T : Scheme.{0}} {r : ℕ} (f g : T ⟶ (powA F r).left) (h₀ : f ≫ (powA F r).hom = g ≫ (powA F r).hom) (h : ∀ i, f ≫ (coord F i).left = g ≫ (coord F i).left) : f = g

theorem **ptAt_left** {T : Scheme.{0}} {r : ℕ} (f : T ⟶ (powA F r).left) (s' : T ⟶ Sp o) (hs : f ≫ toO F r = s') (i : Fin r) : (ptAt F f s' hs i).left ≫ Pdiv.PullbackComparison.pφ (f ≫ (powA F r).hom) F.univ.A = f ≫ (coord F i).left

theorem **cAt_add** {T : Scheme.{0}} {r : ℕ} (f : T ⟶ (powA F r).left) (s' : T ⟶ Sp o) (hs : f ≫ toO F r = s') (a b : Fin r → ℤ) : cAt F f s' hs (a + b) = cAt F f s' hs a ∗ cAt F f s' hs b

theorem **cAt_single** {T : Scheme.{0}} {r : ℕ} (f : T ⟶ (powA F r).left) (s' : T ⟶ Sp o) (hs : f ≫ toO F r = s') (i : Fin r) : cAt F f s' hs (Pi.single i 1) = ptAt F f s' hs i

theorem **spMap_base** : spMap (base F y) = toPow F y ≫ toO F r

theorem **principal_sieg** : (sieg F y).Principal

theorem **c₀_add** (a b : Fin r → ℤ) : c₀ F y (a + b) = c₀ F y a ∗ c₀ F y b

theorem **c₀_single** (i : Fin r) : c₀ F y (Pi.single i 1) = pt F y i

theorem **pt_left** (i : Fin r) : (pt F y i).left ≫ Pdiv.PullbackComparison.pφ (toPow F y ≫ (powA F r).hom) F.univ.A = toPow F y ≫ (coord F i).left

theorem **toPow_closedPoint** : toPow F y (closedPoint (W F y)) = y

theorem **isAdicComplete_completedStalkSiegelFibrePower** [IsNoetherianRing o] : IsAdicComplete (maximalIdeal (W F y)) (W F y)

theorem **isNoetherianRing_completedStalkSiegelFibrePower** [IsNoetherianRing o] : IsNoetherianRing (W F y)

theorem **isRegularLocalRing_stalk** [IsRegularRing o] : IsRegularLocalRing ((powA F r).left.presheaf.stalk y)

theorem **isRegularLocalRing_W_of_isRegularRing** [IsRegularRing o] : IsRegularLocalRing (W F y)

theorem **isDomain_W_of_isRegularRing** [IsRegularRing o] : IsDomain (W F y)

theorem **existsUnique_factor** [IsNoetherianRing o] (V : Type) [CommRing V] [IsLocalRing V] [IsAdicComplete (maximalIdeal V) V] (f : Sp V ⟶ (powA F r).left) (hf : f (closedPoint V) = y) : ∃! g : Sp V ⟶ Sp (W F y), g ≫ toPow F y = f

theorem **exists_local_hom** [IsNoetherianRing o] (V : Type) [CommRing V] [IsLocalRing V] [IsAdicComplete (maximalIdeal V) V] (f : Sp V ⟶ (powA F r).left) (hf : f (closedPoint V) = y) : ∃ ψ : W F y →+* V, IsLocalHom ψ ∧ spMap ψ ≫ toPow F y = f

theorem **isUnit_zetaAway** (n : ℕ) (hn : 3 ≤ n) : IsUnit (zetaAway n)

theorem **flat_spMap_base** : Flat (spMap (base F y))

theorem **flat_base** : (base F y).Flat

theorem **injective_base** [IsDomain o] : Function.Injective (base F y)

theorem **charZero_W_of_charZero** [IsDomain o] [CharZero o] : CharZero (W F y)

Development.AbelianVarietyTheory.Definitions.Divisors.IsPairOfIsTriple

theorem **incM_one** : incM F (1 : Matrix (Fin n) (Fin n) E) = 1

theorem **isEv_id** (B : Base F E) (M : Model σ X W) : IsEv B M M 1 1 ∅ LinearMap.id where

theorem **isLevi_one** (H' : Plane E d d₁) : IsLevi (σ

theorem **shiftA_levi** (H : Plane E n n₁) (H' : Plane E d d₁) {t : (AE F E)ˣ} {h₁ : UA σ W₁.J} {m : UA σ W.J} (h : IsLevi H' t h₁ m) (l : (AE F E)ˣ) : shiftA H H' l ∗ (mat m)ᵀ = cA σ (t⁻¹).1 • shiftA H H' l

theorem **shiftB_levi** (H : Plane E n n₁) (H' : Plane E d d₁) {t : (AE F E)ˣ} {h₁ : UA σ W₁.J} {m : UA σ W.J} (h : IsLevi H' t h₁ m) (l : (AE F E)ˣ) : shiftB σ H H' l ∗ (mat m)ᵀ = t.1 • shiftB σ H H' l

theorem **isCT_const** (H' : Plane E d d₁) (c : ℂ) : IsCT (σ

theorem **hasLaw_const** (H' : Plane E d d₁) (c : ℂ) : HasLaw (σ

theorem **eq_zero_of_hasLaw** (H' : Plane E d d₁) {χ χ' : Chr F E} {z z' : ℝ} (hz : z ≠ z') {T : UA σ W.J → ℂ} (h : HasLaw W₁ H' χ z T) (h' : HasLaw W₁ H' χ' z' T) {t : (AE F E)ˣ} {m : UA σ W.J} (hm : IsLevi (W₁

theorem **pl_adapted** (B : Base F E) (ε : E) : (pl E).IsAdapted σ B (hyp F ε) (pt F E) 0 1 where

theorem **nonempty_model** (h2 : Module.finrank F E = 2) (hσ : ∀ x, σ (σ x) = x) (h1 : ∀ i j, σ (X.J j i) = X.J i j) (h5 : ∀ i j, σ (W.J j i) = -W.J i j) (h1' : X.J.det ≠ 0) (h5' : W.J.det ≠ 0) : Nonempty (Model σ X W)

theorem **exists_cfg** (h2 : Module.finrank F E = 2) (hσ : ∀ x, σ (σ x) = x) (B : Base F E) : (∃ (G : Geo σ 0 (0 + 2)) (G₁ : Geo σ 0 0) (H : Plane E (0 + 2) 0) (H' : Plane E (0 + (0 + 2)) (0 + 0)), CfgB B G G₁ H H') ∧ ∃ (G : Geo σ (0 + 2) 0) (G₁ : Geo σ 0 0) (H : Plane E (0 + 2) 0) (H' : Plane E (0 + 2 + 0) (0 + 0)), CfgA B G G₁ H H'

theorem **resEta_div** {K : Subgroup (AF F)ˣ} {χ χ' : Chr F E} {a b m : ℕ} (h : ResEtaPow K χ a) (h' : ResEtaPow K χ' b) (hm : m = a + b) : ResEtaPow K (χ / χ') m

theorem **isHecke_div** {χ χ' : Chr F E} (h : IsHecke F E χ) (h' : IsHecke F E χ') : IsHecke F E (χ / χ')

theorem **isPair_of_isTriple** {a a' : ℕ} {B : Base F E} {G : Geo σ a a'} {Y : Chars F E} {om : Rep σ G.V G.W} {om' : Rep σ G.V' G.W} (h : IsTriple B G Y om om') : IsPair B G.M Y.β (Y.β / Y.α) om (a − G.p) G.p (a + a' − (G.p + G.q)) (G.p + G.q) ∧ IsPair B G.M' Y.α (Y.α / Y.β) om' G.q (a' − G.q) (a + a' − (G.p + G.q)) (G.p + G.q)

theorem **conv_range** {a a' : ℕ} {B : Base F E} {G : Geo σ a a'} (h : G.IsGood B) : a + a' + 2 ∗ min (a − G.p) G.p < 2 ∗ a

theorem **intInd_mem_SB** {κ : Type} [Fintype κ] (v : HeightOneSpectrum (𝒪 F)) : Set.indicator {x : κ → Fv F v | ∀ k, x k ∈ v.adicCompletionIntegers F} 1 ∈ GlobalTheory.WeilM.SB (κ → Fv F v)

theorem **const_mem_SB** {R κ : Type} [TopologicalSpace R] [IsEmpty κ] (φ : (κ → R) → ℂ) : φ ∈ SB (κ → R)

theorem **isEv_origin** (B : Base F E) {n₁ d₁ : ℕ} [IsEmpty (Fin n₁ × Fin d₁)] {X₁ : Spc F E n₁} {W₁ : Spc F E d₁} (M : Model σ X W) (M₁ : Model σ X₁ W₁) (hv : ∀ v, M₁.vac v = fun _ => M.vac v 0) (ι : Matrix (Fin n) (Fin n₁) E) (ι' : Matrix (Fin d) (Fin d₁) E) (I : Set (Matrix (Fin n) (Fin d) (AE F E))) (hI : ∀ u ∈ I, (M.co.e u).1 = 0) : IsEv B M M₁ ι ι' I (evOrigin (AF F) _ _) where

theorem **isStab_unique** (B : Base F E) {K K' : Subgroup (UA σ X.J)} (h : IsStab B K) (h' : IsStab B K') : K = K'

theorem **split2_unique** {a a' : ℕ} (ha : a ≠ a') {W : Spc F E (a + a')} {H' : Plane E (a + a') d₁} {Y : Chars F E} {f fa fb ga gb : UA σ W.J → ℂ} (h : Split2 W₁ H' Y f fa fb) (h' : Split2 W₁ H' Y f ga gb) (μ : Measure (unip σ W H')) (𝓕 : Set (unip σ W H')) (hμ :

μ.IsHaarMeasure) (hℱ : IsFundamentalDomain ((ratU σ W.J).subgroupOf (unip σ W H')) ℱ μ) (h1 : μ ℱ = 1) {t : (AE F E)ˣ} {m : UA σ W.J} (hm : IsLevi (W₁

Development.AbelianVarietyTheory.Definitions.Divisors.IsPeriodGens

theorem **divFrom_iff** : DivFrom L Q oQ iw s ε e ↔ ∃ g hD, DivBy L Q oQ iw s ε e g hD

Development.AbelianVarietyTheory.Definitions.Divisors.IsProjOverWeierstrass

theorem **Agree.symm** {X Y Y' : Scheme.{0}} {L : LB X} {h : Y ⟶ X} {h' : Y' ⟶ X} {k : ℕ} {s : SecK L h k} {s' : SecK L h' k} (H : Agree L s s') : Agree L s' s

theorem **GalEq.galEqN** {R : Type} [CommRing R] [IsDomain R] {D : DegPre R} {A : PolAV (Sp (FractionRing R))} {lab : Labels D A} (h : GalEq lab) {n : ℕ} (hn : 0 < n) : GalEqN n lab

theorem **val_psi** {Y : Type} [AddCommGroup Y] [Module.Free ℤ Y] [Module.Finite ℤ Y] (q : Y → (FractionRing K)ˣ) (y : Y) (z J) (hJ : Good (toricPre K Y q).P.L z J) : val ((toricPre K Y q).psi y) (𝟙 _) z J = gm (𝟙 _) z ⊤ (unitOn (q y))

theorem **val_tau** {Y : Type} [AddCommGroup Y] [Module.Free ℤ Y] [Module.Finite ℤ Y] (q : Y → (FractionRing K)ˣ) (a b : Y) (z J) (hJ : Good (toricPre K Y q).P.L z J) : val ((toricPre K Y q).tau a b) (𝟙 _) z J = gm (𝟙 _) z ⊤ (unitOn (polar q a b))

theorem **gm_tq** (q : X → (FractionRing K)ˣ) {r : ℕ} (W : Matrix (Fin r) (Fin r) ℤ) (y : Fin r → X) (z : (TK K).left) : gm (𝟙 (TK K).left) z ⊤ (unitOn (tq q W y)) = ∏ i, ∏ j, gm (𝟙 (TK K).left) z ⊤ (unitOn (polar q (y i) (y j))) ^ W i j

theorem **polar_tq** (u : (FractionRing K)ˣ) (y z : Fin 2 → ℤ) : polar (tq (fun n : ℤ => u ^ (n ^ 2)) (1 : Matrix (Fin 2) (Fin 2) ℤ)) y z = u ^ (4 * (y 0 * z 0 + y 1 * z 1))

theorem **isAct_gauss** (u : (FractionRing K)ˣ) : ∃ (_ : Module GaussianInt (toricPre K (Fin 2 → ℤ) (tq (fun n : ℤ => u ^ (n ^ 2)) 1)).X) (o : (toricPre K (Fin 2 → ℤ) (tq (fun n : ℤ => u ^ (n ^ 2)) 1)).OAct GaussianInt), (toricTens (fun n : ℤ => u ^ (n ^ 2)) 1).IsAct o rho

theorem **selfUnif** {W : Type} [CommRing W] [IsDomain W] [IsLocalRing W] (hW : GlobalTheory.LN.CDvr W) (P : PolAV (Sp W)) : ∃ (D : DegData W (maximalIdeal W)) (A : PolAV (Sp (FractionRing W))) (U : Unif (maximalIdeal W) D A P.A), U.LG.L = P.L

theorem **local_or_factor** [IsDiscreteValuationRing R] {W : Type} [CommRing W] [IsLocalRing W] (φ : R →+* W) : IsLocalHom φ ∨ ∃ ψ : FractionRing R →+* W, ψ.comp (algebraMap R _) = φ

theorem **exists_uCover** (hR : GlobalTheory.LN.CDvr R) {D : DegData R (maximalIdeal R)} {G : Over (Sp R)} [iG : GrpObj G] [cG : IsCommMonObj G] {A : PolAV (Sp (FractionRing R))} (hG : IsSemiAbelian G) (U : Unif (maximalIdeal R) D A G) : ∃ C : UCover R, C.toModif = Modif.self R ∧ C.A = A ∧ C.G.left = G.left

theorem **uCover_rank_one** : ∃ (C : UCover (PowerSeries ℚ)) (D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (G : Over (Sp (PowerSeries ℚ))) (_ : GrpObj G) (_ : IsCommMonObj G), C.A.g = 1 ∧ C.G.left = G.left ∧ Module.finrank ℤ D.toPre.X = 1 ∧ Nonempty (Unif (maximalIdeal _) D C.A G)

theorem **nz_secK** {L : LB X} (t : L.Sec) (h : Y ⟶ X) : nonzeroLocus (secK t h) = h ⁻¹ᵁ nonzeroLocus t

theorem **agree_secK** {L : LB X} (t : L.Sec) (h : Y ⟶ X) (h' : Y' ⟶ X) : Agree L (secK t h) (secK t h')

theorem **toProj_secK** {R : Type} [CommRing R] {N : ℕ} {L : LB X} (f : X ⟶ Sp R) (t : Fin (N + 1) → L.Sec) (ht : ⊔ i, nonzeroLocus (t i) = ⊤) (h : Y ⟶ X) : ∃ hs, GlobalTheory.ProjSpace.toProj (h ≫ f) (fun i => secK (t i) h) hs = h ≫ GlobalTheory.ProjSpace.toProj f t ht

theorem **traitReg_secK** (M : Modif R) (G : Over M.S) {L : LB G.left} (t : L.Sec) (α : Y ⟶ G.left) : TraitReg M G L (secK t α)

theorem **isProjOver_weierstrass** {R : Type} [CommRing R] (W : _root_.WeierstrassCurve R) : IsProjOver (GlobalTheory.WeierstrassCurve.toSpec W)

theorem **secs_multiplicative** (M : Modif (PowerSeries ℚ)) : ∃ (G₀ : Over (Sp (PowerSeries ℚ))) (_ : GrpObj G₀) (_ : IsSemiAbelian G₀) (_ : IsProper ((Over.pullback (jK (PowerSeries ℚ))).obj G₀).hom) (L : LB ((Over.pullback M.π).obj G₀).left) (_ : L.IsLB) (S : Secs M ((Over.pullback M.π).obj G₀) L (pullback.fst ((Over.pullback M.π).obj G₀).hom M.gp)) (i : splitTorus (Sp (ResidueField (PowerSeries ℚ))) ℤ ≅ S.Y), IsMonHom i.hom ∧ (∀ a, TraitReg M _ L (S.s a)) ∧ TraitReg M _ L S.0

Development.AbelianVarietyTheory.Definitions.Divisors.IsProperToroidalModelStructureMap

theorem **mem_levelForms** {β : LinearMap.BilinForm ℤ X} : β ∈ levelForms H N X ↔ (∀ y z, β y z = β z y) ∧ (∀ (a : 𝒪 E) y z, β (a • y) z = β y (H.bar a • z)) ∧ ∀ y z, (N : ℤ) ∣ β y z

theorem **mem_levelForms_iff_smul** (hN : N ≠ 0) {β : LinearMap.BilinForm ℤ X} : β ∈ levelForms H N X ↔ ∃ γ ∈ levelForms H 1 X, β = (N : ℤ) • γ

theorem **apply_self_eq_two_mul** {β : LinearMap.BilinForm ℤ X} (h : β ∈ levelForms H N X) (y : X) : β y y = 2 * β y (H.w0 • y)

@[simp] theorem **toSymB_apply** (β : levelForms H N X) (a b : X) : (toSymB H N X β : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) a b = ((β : LinearMap.BilinForm ℤ X) a b : ℝ)

theorem **toSymB_injective** : Function.Injective (toSymB H N X)

theorem **mem_range_toSymB** (B : Formalism.MainNotions.ModuliIntegralCharts.SymB (fun x : E => IsCMField.complexConj E x) X) : B ∈ LinearMap.range (toSymB H N X) ↔ ∀ a b, ∃ z : ℤ, (B : X →ₗ[ℤ] X →ₗ[ℤ] ℝ) a b = N * z

theorem **smul_trForm_mem** (H : Host E) (N : ℕ) : (N : ℤ) • HostModel.trForm H ∈ levelForms H N (𝒪 E)

theorem **IsPartOfBasis.linearIndependent** {v : Fin m → L} (h : IsPartOfBasis v) : LinearIndependent ℤ v

theorem **IsPartOfBasis.isUnit_of_smul** {v : Fin m → L} (h : IsPartOfBasis v) {i : Fin m} {k : ℤ} {w : L} (hw : k • w = v i) : IsUnit k

theorem **IsPartOfBasis.exists_dual** {v : Fin m → L} (h : IsPartOfBasis v) : ∃ φ : Fin m → (L →ₗ[ℤ] ℤ), ∀ i j, φ i (v j) = if i = j then 1 else 0

theorem **exists_isPartOfBasis_of_ne_zero** [Module.Free ℤ L] [Module.Finite ℤ L] {v : L} (hv : v ≠ 0) : ∃ (k : ℤ) (w : L), 0 < k ∧ v = k • w ∧ IsPartOfBasis ![w]

theorem **IsMonomial.nonneg** (h : IsMonomial Dg t β) (i : Fin m) (y : Dg.X) : 0 ≤ β i y y

theorem **IsMonomial.coord** (h : IsMonomial Dg t β) {y : Dg.X} {r : R} (hr : PsiCoord Dg y r) : ∃ (u : Rˣ) (n : Fin m → ℕ), (∀ i, β i y y = 2 * n i) ∧ r = u * ∏ i, t i ^ n i

theorem **IsMonomial.exists_pos** {I : Ideal R} (hI : I ≠ ⊤) {Dd : DegData R I} {β : Fin m → LinearMap.BilinForm ℤ Dd.toPre.X} (h : IsMonomial Dd.toPre t β) {y : Dd.toPre.X} (hy : y ≠ 0) : ∃ i, 0 < β i y y

theorem **IsMonomial.tau** (h : IsMonomial Dg t β) (hs : ∀ i y z, β i y z = β i z y) (ht : ∀ i, t i ≠ 0) {a b : Dg.X} {r r' r'' : R} (ha : PsiCoord Dg a r) (hb : PsiCoord Dg b r') (hab : PsiCoord Dg (a + b) r'') : ∃ u : Rˣ, algebraMap R (FractionRing R) r'' / (algebraMap R

_ r * algebraMap R _ r') = algebraMap R _ u * ∏ i, algebraMap R (FractionRing R) (t i) ^ β i a b

theorem **IsMonomial.betaAt_eq** (h : IsMonomial Dg t β) (hs : ∀ i y z, β i y z = β i z y) {C : Type} [NormedField C] (f : R →+* C) (hle : ∀ a, ‖f a‖ ≤ 1) (ht : ∀ i, f (t i) ≠ 0) (p : ℕ) (a b : Dg.X) : betaAt Dg f p a b = ∑ i, -Real.logb p ‖f (t i)‖ * β i a b

theorem **psiUnitOn_of_coord** {y : Dg.X} {r : R} (hr : PsiCoord Dg y r) {U : Set (Sp R)} (hU : ∀ p ∈ U, r ∉ p.asIdeal) : PsiUnitOn Dg y U

theorem **IsMonomial.psiUnitOn** (h : IsMonomial Dg t β) (y : Dg.X) {U : Set (Sp R)} (hU : ∀ p ∈ U, ∏ i, t i ∉ p.asIdeal) : PsiUnitOn Dg y U

theorem **isMonomial_arcData** (k : Type) [Field k] {X : Type} [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (be : LinearMap.BilinForm ℤ X) (hs : ∀ y z, be y z = be z y) (hp : ∀ y, y ≠ 0 → 0 < be y y) (he : ∀ y, (2 : ℤ) ∣ be y y) : IsMonomial (ToricData.arcData k be hs hp he).toPre ![PowerSeries.X] ![be]

theorem **exists_isMonomial_isPartOfBasis** (H : Host E) {N : ℕ} (hN : N ≠ 0) (k : Type) [Field k] : ∃ (Dd : DegData (PowerSeries k) (Ideal.span {PowerSeries.X})) (_ : Module (𝒪 E) Dd.toPre.X) (γ : Fin 1 → levelForms H N Dd.toPre.X), Nontrivial Dd.toPre.X ∧ IsMonomial Dd.toPre ![PowerSeries.X] (fun i => γ i) ∧ IsPartOfBasis γ

theorem **isEtaleCompletion_id** (A : Type) [CommRing A] [IsLocalRing A] : IsEtaleCompletion (RingHom.id A)

theorem **mem_genBd** {Y : Model Xo c β} {x : gen Y} : x ∈ genBd Y ↔ x ∉ Set.range Y.jQ

theorem **mem_genBd_iff** {Y : Model Xo c β} {x : gen Y} : x ∈ genBd Y ↔ genι Y x ∉ abLocus Y.t.G

theorem **mem_abLocus_iff** (h : IsToricChart Z ξ P γ) (p : Sp (ChartRing k m d)) : p ∈ abLocus t.G ↔ ∏ i, bdParam k m d i ∉ p.asIdeal

theorem **data_at_point_n3** (h : IsToricChart Z ξ P γ) : UniformisationConclusion t (abLocus t.G)

theorem **centre_mem_iff** (h : IsToricChart Z ξ P γ) : ξ (closedPoint _) ∈ genBd Z ↔ m ≠ 0

theorem **forms** (h : IsToricChart Z ξ P γ) : (∀ i y, 0 ≤ (γ i : LinearMap.BilinForm ℤ P.D.toPre.X) y y) ∧ (∀ i, γ i ≠ 0) ∧ ∀ y, y ≠ 0 → ∃ i, 0 < (γ i : LinearMap.BilinForm ℤ P.D.toPre.X) y y

theorem **nontrivial** (h : IsToricChart Z ξ P γ) (hx : ξ (closedPoint _) ∈ genBd Z) : Nontrivial P.D.toPre.X

theorem **arcChart_iff** (Y : Model Xo c β) : Y.N4 ↔ ∀ (O' : Type) [CommRing O'] [IsDomain O'] [CharZero O'] (b : O →+* O') (ϖ : O') (a : ArcIn D l β O' b), IsArcBase O' ϖ → ∃ χ : Sp (PowerSeries O') ⟶ Y.X, (∀ χ', χ' = χ ↔ IsArcChart Y a ϖ χ') ∧ (Irreducible ϖ → ∀ (y' : Sp O' ⟶ Y.X) (ev : PowerSeries O' →+* O'), y' ≫ Y.s = spMap b → Set.range (Out.jK O' ≫ y') ⊆ Set.range Y.j → Y.HasArcData a (Ideal.span {ϖ}) (RingHom.id O') ϖ y' → (∀ x, ev (PowerSeries.C x) = x) → ev PowerSeries.X = ϖ → spMap ev ≫ χ = y')

theorem **isProper_toroidalModel_structureMap** (Y : Model Xo c β) : IsProper Y.s

theorem **isProper_genStr** (Y : Model Xo c β) : IsProper (genStr Y)

theorem **b_mem_abLocus** (x : T.Z.X) : T.b x ∈ abLocus Y.t.G ↔ x ∈ abLocus T.Z.t.G

@[reassoc (attr

@[reassoc (attr

@[reassoc (attr

theorem **bQ_mem_genBd** (x : gen T.Z) : T.bQ x ∈ genBd Y ↔ x ∈ genBd T.Z

theorem **isPullback_jQ** : IsPullback (𝟙 _) T.Z.jQ Y.jQ T.bQ

theorem **isIso_bQ_restrict** : IsIso (T.bQ ∣_ (haveI

theorem **surjective_bQ** : Function.Surjective T.bQ

theorem **mem_support_BD_iff** (x : gen T.Z) : genι T.Z x ∈ T.BD.idealSheaf.support ↔ x ∈ genBd T.Z

theorem **j_notMem_support_BD** (x : Xo.XF c) : T.Z.j x ∉ T.BD.idealSheaf.support

theorem **isRegularLocalRing** (x : gen T.Z) : IsRegularLocalRing ((gen T.Z).presheaf.stalk x)

theorem **n3** (h : Y.N3) : T.Z.N3

theorem **lift_spec** {O' : Type} [CommRing O'] [IsDomain O'] {b' : O →+* O'} {a : ArcIn D l β O' b'} {ϖ : O'} {χ'' : Sp (PowerSeries O') ⟶ T.Z.X} (h : IsArcChart Y a ϖ (χ'' ≫ T.b)) : χ'' ≫ T.Z.s = spMap (PowerSeries.C (R

theorem **smooth_of_surjective** (Y : Model Xo c β) (h : Function.Surjective Y.jQ) : SmoothOfRelativeDimension D.dim (genStr Y)

theorem **surjective_jQ_of_dim_eq_zero** (Y : Model Xo c β) (h : D.dim = 0) : Function.Surjective Y.jQ

theorem **nonempty_torModel_of_definite** (Y : Model Xo c β) (h : D.p = 0 ∨ D.q = 0) : Nonempty (TorModel Y)

Development.AbelianVarietyTheory.Definitions.Divisors.IsQuasiRegularSeqOfIsFreeGen

theorem **exists_affine** {U : Y.Opens} {y : Y} (hy : y ∈ U) : ∃ W : Y.affineOpens, y ∈ W.1 ∧ W.1 ≤ U

theorem **stalkId_pow** (a : ℕ) (y : Y) : stalkId (I ^ a) y = stalkId I y ^ a

theorem **stalkId_eq_top** {y : Y} (hy : y ∉ I.support) : stalkId I y = ⊤

theorem **ι_mem_support** (z : I.subscheme) : I.subschemeι z ∈ I.support

theorem **exists_ι_eq** {y : Y} (hy : y ∈ I.support) : ∃ z : I.subscheme, I.subschemeι z = y

theorem **ker_stalkMap** (z : I.subscheme) : RingHom.ker (I.subschemeι.stalkMap z).hom = stalkId I (I.subschemeι z)

theorem **mem_ideal_iff_ger** (W : Y.affineOpens) (a : Γ(Y, W.1)) : a ∈ I.ideal W ↔ ∀ y hy, ger y hy a ∈ stalkId I y

theorem **stalkId_cutIdeal** (j : L.ι) (y : Y) (hj : y ∈ L.U j) : stalkId (cutIdeal hL s) y = Ideal.span (Set.range (secGer s j y hj))

theorem **vanishesAlong_cutIdeal** (i : ι) : VanishesAlong (s i) (cutIdeal hL s).subschemeι

theorem **secGer_change** (j j' : L.ι) (y : Y) (hj : y ∈ L.U j) (hj' : y ∈ L.U j') (i : ι) : secGer s j y hj i = tr L y j j' * secGer s j' y hj' i

theorem **formallyUnramified_quotient** (h : FormallyUnramified (I.subschemeι ≫ p)) (W : Y.affineOpens) : ((Ideal.Quotient.mk (I.ideal W)).comp (algX p W.1)).FormallyUnramified

theorem **isUnit_ger_jacFun** (h : FormallyUnramified ((cutIdeal hL s).subschemeι ≫ p)) {y : Y} (hy : y ∈ (cutIdeal hL s).support) (i : K.bundle.ι) (j : L.ι) (hij : y ∈ K.bundle.U i ⊓ L.U j) : IsUnit (ger y hij (K.jacFun s i j))

theorem **isQuasiRegularSeq_of_isFreeGen** {A : Type} [CommRing A] {α : k →+* A} {D : ℕ} (h : α.IsStandardSmoothOfRelativeDimension D) (x : Fin D → A) (hx : TopForm.IsFreeGen A (TopForm.dForm α x)) : IsQuasiRegularSeq A x

theorem **isQuasiRegularSeq_secGer** (h : FormallyUnramified ((cutIdeal hL s).subschemeι ≫ p)) {y : Y} (hy : y ∈ (cutIdeal hL s).support) (j : L.ι) (hj : y ∈ L.U j) : IsQuasiRegularSeq (Y.presheaf.stalk y) (secGer s j y hj)

Development.AbelianVarietyTheory.Definitions.Divisors.IsQuasicoherentEigenQuot

theorem **mem_eigenSub_iff** {s : Γ(F, V)} : s ∈ (eigenSub ρ σ).sec V ↔ ∀ a, (act ρ a).app V s = resG X (le_top : V ≤ ⊤) (σ a) • s

theorem **eigenι_act** (a : O) : eigenι ρ σ ≫ act ρ a = σ a • eigenι ρ σ

@[reassoc (attr

@[reassoc (attr

theorem **ActsThrough.isIso_eigenι** (h : ActsThrough ρ σ) : IsIso (eigenι ρ σ)

theorem **isEquivariant_eigenι** : IsEquivariant (scalarAct (eigen ρ σ) σ) ρ (eigenι ρ σ)

theorem **act_eigenπ** (a : O) : act ρ a ≫ eigenπ ρ σ = σ a • eigenπ ρ σ

@[reassoc (attr

@[simp] theorem **act_dualAct** (a : O) : act (dualAct ρ) a = dualMap (act ρ a)

theorem **ActsThrough.dual** (h : ActsThrough ρ σ) : ActsThrough (dualAct ρ) σ

theorem **IsEquivariant.dual** {φ : F ⟶ G} (h : IsEquivariant ρ ρ' φ) : IsEquivariant (dualAct ρ') (dualAct ρ) (dualMap φ)

theorem **pullback_map_smulHom** (f : Y ⟶ X) (a : Γ(X, ⊤)) : (Scheme.Modules.pullback f).map (smulHom F a) = smulHom _ (f.appTop a)

theorem **pullback_map_smul** (f : Y ⟶ X) (a : Γ(X, ⊤)) (φ : F ⟶ G) : (Scheme.Modules.pullback f).map (a • φ) = (f.appTop a : Γ(Y, ⊤)) • (Scheme.Modules.pullback f).map φ

@[simp] theorem **act_pullAct** (f : Y ⟶ X) (a : O) : act (pullAct ρ f) a = (Scheme.Modules.pullback f).map (act ρ a)

theorem **ActsThrough.pullback** (h : ActsThrough ρ σ) (f : Y ⟶ X) : ActsThrough (pullAct ρ f) (f.appTop.hom.comp σ)

theorem **IsEquivariant.pullback** {φ : F ⟶ G} (h : IsEquivariant ρ ρ' φ) (f : Y ⟶ X) : IsEquivariant (pullAct ρ f) (pullAct ρ' f) ((Scheme.Modules.pullback f).map φ)

@[reassoc (attr

theorem **isIso_eigenLift_congr** {τ τ' : O →+* Γ(X, ⊤)} (e : τ = τ') (φ : G ⟶ F) (h : ∀ a, φ ≫ act ρ a = τ a • φ) (h' : ∀ a, φ ≫ act ρ a = τ' a • φ) (H : IsIso (eigenLift φ h)) : IsIso (eigenLift φ h')

theorem **isIso_eigenLift** {ι : Type} [Fintype ι] {A : ι → X.Modules} (τ : ι → (O →+* Γ(X, ⊤))) (hτ : ∀ i j, i ≠ j → ∃ a, IsUnit (τ i a − τ j a)) (p : ∀ j, F ⟶ A j) (q : ∀ j, A j ⟶ F) (hpq : ∑ j, p j ≫ q j = 𝟙 F) (hqp : ∀ j, q j ≫ p j = 𝟙 _) (hp : ∀ j a, act ρ a ≫ p j = τ j a • p j) (hq : ∀ j a, q j ≫ act ρ a = τ j a • q j) (j : ι) : IsIso (eigenLift (q j) (hq j))

@[simp] theorem **splitMap_tmul** (σ : ι → (O →+* k)) (c : k) (a : O) (j : ι) : splitMap σ (c ⊗ₜ a) j = c ∗ σ j a

theorem **IsSplitBy.injective** (h : IsSplitBy σ) : Function.Injective σ

theorem **isSplitBy_of_injective** [Fintype ι] [Module.Free ℤ O] [Module.Finite ℤ O] (hσ : Function.Injective σ) (hn : Module.finrank ℤ O = Fintype.card ι) : IsSplitBy σ

theorem **IsSplitBy.exists_eq** [Fintype ι] (h : IsSplitBy σ) (τ : O →+* k) : ∃ j, σ j = τ

theorem **act_tensorAct_tmul** (c : k) (a : O) : act (tensorAct k ρ) (c ⊗ₜ a) = algebraMap k Γ(X, ⊤) c • act ρ a

theorem **eigenι_tensorAct** (j : ι) (x : k ⊗[ℤ] O) : eigenι ρ (scal X (σ j)) ≫ act (tensorAct k ρ) x = algebraMap k Γ(X, ⊤) (splitMap σ x j) • eigenι ρ (scal X (σ j))

@[simp] theorem **splitMap_idem** (j : ι) : splitMap σ (h.idem j) = Pi.single j 1

theorem **tmul_mul_idem** (j : ι) (a : O) : ((1 : k) ⊗ₜ[ℤ] a) ∗ h.idem j = (σ j a ⊗ₜ[ℤ] (1 : O)) ∗ h.idem j

theorem **idem_act** (j : ι) (a : O) : act (tensorAct k ρ) (h.idem j) ≫ act ρ a = scal X (σ j) a • act (tensorAct k ρ) (h.idem j)

theorem **act_idem** (j : ι) (a : O) : act ρ a ≫ act (tensorAct k ρ) (h.idem j) = scal X (σ j) a • act (tensorAct k ρ) (h.idem j)

theorem **proj_ι** (j : ι) : h.proj ρ j ≫ eigenι _ _ = act (tensorAct k ρ) (h.idem j)

@[reassoc (attr

@[reassoc] theorem **ι_proj_ne** {i j : ι} (hij : i ≠ j) : eigenι ρ (scal X (σ i)) ≫ h.proj ρ j = 0

theorem **act_proj** (j : ι) (a : O) : act ρ a ≫ h.proj ρ j = scal X (σ j) a • h.proj ρ j

theorem **isIso_eigenι_eigenπ** (h : IsSplitBy σ) [Fintype ι] (j : ι) : IsIso (eigenι ρ (scal X (σ j)) ≫ eigenπ ρ (scal X (σ j)))

theorem **sum_proj_ι** : ∑ j, h.proj ρ j ≫ eigenι _ _ = 𝟙 F

theorem **isFiniteQC_eigen** [IsFiniteQC F] (j : ι) : IsFiniteQC (eigen ρ (scal X (σ j)))

theorem **isIso_pullbackEigenHom** (f : Y ⟶ X) (j : ι) : IsIso (pullbackEigenHom ρ (scal X (σ j)) f)

@[simp] theorem **frob_apply** (x : X) : frob X p h x = x

@[simp] theorem **frob_preimage** (U : X.Opens) : frob X p h ⁻¹ᵁ U = U

theorem **frob_app** (U : X.Opens) (a : Γ(X, U)) : (frob X p h).app U a = (a ^ p : Γ(X, U))

theorem **frob_appTop** (a : Γ(X, ⊤)) : (frob X p h).appTop a = a ^ p

theorem **frob_appLE** (e : V ≤ frob X p h ⁻¹ᵁ U) (a : Γ(X, U)) : (frob X p h).appLE U V e a = resG X e a ^ p

@[reassoc] theorem **frob_naturality** (f : Y ⟶ X) (hY : (p : Γ(Y, ⊤)) = 0) : frob Y p hY ≫ f = f ≫ frob X p h

theorem **natCast_eq_zero** (k : Type) [CommRing k] [CharP k p] [Algebra k Γ(X, ⊤)] : (p : Γ(X, ⊤)) = 0

theorem **IsVB.frobPull** {r : ℕ} (hF : IsVB F r) : IsVB ((frobPull p h).obj F) r

theorem **cls_pull_frob** {L : LB X} (hL : L.IsLB) (h' : (L.pull (frob X p h)).IsLB) : CechPic.cls _ h' = p • CechPic.cls L hL

theorem **frob_comp_scal** (σ : O →+* k) : (frob X p h).appTop.hom.comp (scal X σ) = scal X ((frobenius k p).comp σ)

theorem **ActsThrough.frobPull** {ρ : O →+* End F} {σ : O →+* k} (hρ : ActsThrough ρ (scal X σ)) : ActsThrough (pullAct ρ (frob X p h)) (scal X ((frobenius k p).comp σ))

@[reassoc (attr

theorem **IsSplitBy.isIso_frobEigenHom** {ι : Type} [Fintype ι] {σ : ι → (O →+* k)} (hσ : IsSplitBy σ) (ρ : O →+* End F) (j : ι) : IsIso (frobEigenHom p h ρ (σ j))

Development.AbelianVarietyTheory.Definitions.Divisors.IsRatPath

theorem **triv_over** {S : Scheme.{0}} (G : Over S) [GrpObj G] {R : Type} [CommRing R] (b : Sp R ⟶ S) : triv G b ≫ G.hom = spMap (PowerSeries.C (R

Development.AbelianVarietyTheory.Definitions.Divisors.IsRatioOfWeight

theorem **norm_val** (P : Pair p) (a b : ℤ) (σ : k →+* ℂ) (y : CxPts p σ) (f : X.functionField) : ‖val P a b σ y f‖ = P.nu a b σ y f

theorem **ballIn_H** (P : Pair p) (a b : ℤ) (R : ℝ) : ballIn P (P.H a b) a b R = P.Bl a b R

theorem **gens_int** : Gens IntExample.pair 0 1 fun _ => 1 where

theorem **gram_point** : IsFam (S

private theorem **fderiv_half_comb_fderiv** {n : ℕ} {g : ReciprocityTheory.L2DbarEstimates.V n → ℂ} {y : ReciprocityTheory.L2DbarEstimates.V n} (hD : DifferentiableAt ℝ (fderiv ℝ g) y) (c : ℂ) (u u' v : ReciprocityTheory.L2DbarEstimates.V n) : fderiv ℝ (fun w => (fderiv ℝ g w u + c * fderiv ℝ g w u') / 2) y v = (fderiv ℝ (fderiv ℝ g) y v u + c * fderiv ℝ (fderiv ℝ g) y v u') / 2

private theorem **wd_eq_half_fderiv_sub_I** {n : ℕ} (j : Fin n) (g : ReciprocityTheory.L2DbarEstimates.V n → ℂ) : ReciprocityTheory.L2DbarEstimates.wd n j g = fun w => (fderiv ℝ g w (ReciprocityTheory.L2DbarEstimates.bas n j) + (-Complex.I) * fderiv ℝ g w (Complex.I • ReciprocityTheory.L2DbarEstimates.bas n j)) / 2

private theorem **wdb_eq_half_fderiv_add_I** {n : ℕ} (j : Fin n) (g : ReciprocityTheory.L2DbarEstimates.V n → ℂ) : ReciprocityTheory.L2DbarEstimates.wdb n j g = fun w => (fderiv ℝ g w (ReciprocityTheory.L2DbarEstimates.bas n j) + Complex.I * fderiv ℝ g w (Complex.I • ReciprocityTheory.L2DbarEstimates.bas n j)) / 2

private theorem **star_cHess_swap_of_contDiffAt** {n : ℕ} {f : ReciprocityTheory.L2DbarEstimates.V n → ℝ} {x : ReciprocityTheory.L2DbarEstimates.V n} (hf : ContDiffAt ℝ 2 f x) (i j : Fin n) : star (ReciprocityTheory.L2DbarEstimates.cHess n (fun v => ((f v : ℝ) : ℂ)) x j i) = ReciprocityTheory.L2DbarEstimates.cHess n (fun v => ((f v : ℝ) : ℂ)) x i j

theorem **det_cHess_pos** {n : ℕ} {f : V n → ℝ} {x : V n} (hf : ContDiffAt ℝ 2 f x) (hL : ∀ ξ : Fin n → ℂ, ξ ≠ 0 → 0 < (leviV n (fun v => ((f v : ℝ) : ℂ)) x ξ).re) : 0 < (cHess n (fun v => ((f v : ℝ) : ℂ)) x).det.re

theorem **VolDens.isRatio_ofWeight** {σ : k →+* ℂ} {D : ℕ} {B : LB X} (φ : Weight p σ B) (h : φ.IsCInf) (hp : ∀ c : AlgChart p σ D, ∀ y ∈ c.e.source, 0 < c.chernDens φ y) : (VolDens.ofWeight φ h hp).IsRatio φ fun _ => 1

theorem **volFS_meas** (k : Type) [CommRing k] (σ : k →+* ℂ) (N : ℕ) : (volFS k σ N).meas = ((ProjSpace.hermitianTwistingSheaf k N).φ σ).chernPart N 1

theorem **goodFam_int** : GoodFam IntExample.pair 0 1 fun _ => AddSubgroup.zmultiples ((2 : ℕ) : (Sp ℤ).functionField) where

theorem **gens_int'** : Gens IntExample.pair 1 1 ![((2 : ℕ) : (Sp ℤ).functionField), ((3 : ℕ) : (Sp ℤ).functionField)] where

theorem **isCan_int** : IsCan IntExample.pair (D

Development.AbelianVarietyTheory.Definitions.Divisors.IsRecTLocRecT

theorem **isRecT_locRecT** (hT : UpperTheory.Zero.LLC.RecTExists) (h : IsPlaceOver A w) : UpperTheory.Zero.LLC.IsRecT h (locRecT hT h)

Development.AbelianVarietyTheory.Definitions.Divisors.IsRightActFunPull

theorem **isRightAct_inv** (ρ : G →* (R →ₐ[K] R)) : IsRightAct (fun h => ρ h⁻¹)

theorem **IsRightAct.comp** {st : G → R →ₐ[K] R} (h : IsRightAct st) {G' : Type} [Group G'] (f : G' →* G) : IsRightAct (fun g => st (f g))

theorem **isRightAct_funPull** {Z : Type} [MulAction G Z] : IsRightAct (funPull K (Z

theorem **IsRightAct.equiv_apply** {st : G → R →ₐ[K] R} (hs : IsRightAct st) (h : G) (x : R) : hs.equiv h x = st h x

theorem **groupMat_apply** (u : R) (h g : G) : groupMat st u h g = st (h * g) u

theorem **mem_eigen** {χ : G →* Kˣ} {F : R} : F ∈ eigen st χ ↔ ∀ h, st h F = ((χ h : Kˣ) : K) • F

theorem **IsRightAct.leftRep_apply** {st : G → R →ₐ[K] R} (hs : IsRightAct st) (g : G) (x : R) : hs.leftRep g x = st g⁻¹ x

theorem **IsRightAct.mem_eigen_iff** {st : G → R →ₐ[K] R} (hs : IsRightAct st) {χ : G →* Kˣ} {F : R} : F ∈ eigen st χ ↔ ∀ g, hs.leftRep g F = ((χ⁻¹ g : Kˣ) : K) • F

theorem **IsRightAct.subRep_apply** {st : G → R →ₐ[K] R} (hs : IsRightAct st) (C : Submodule K R) (hC : ∀ h, ∀ x ∈ C, st h x ∈ C) (g : G) (x : C) : (hs.subRep C hC g x : R) = st g⁻¹ x

theorem **algOp_single** (g : G) (a : K) (x : R) : algOp st (MonoidAlgebra.single g a) x = a • st g⁻¹ x

theorem **algOp_apply** [Fintype G] (e : MonoidAlgebra K G) (x : R) : algOp st e x = ∑ g, e.coeff g • st g⁻¹ x

theorem **IsRightAct.algOp_eq** {st : G → R →ₐ[K] R} (hs : IsRightAct st) (e : MonoidAlgebra K G) : algOp st e = hs.leftRep.asAlgebraHom e

theorem **IsRightAct.algOp_mul** {st : G → R →ₐ[K] R} (hs : IsRightAct st) (e e' : MonoidAlgebra K G) : algOp st (e * e') = algOp st e * algOp st e'

theorem **algOp_mem** {C : Submodule K R} (hC : ∀ h, ∀ x ∈ C, st h x ∈ C) (e : MonoidAlgebra K G) {x : R} (hx : x ∈ C) : algOp st e x ∈ C

theorem **tr_apply** (χ : G →* Kˣ) (F : R) : tr st χ F = ∑ h, (((χ h)⁻¹ : Kˣ) : K) • st h F

theorem **ph_apply** (χ : G →* Kˣ) (u f : R) (h : G) : ph st χ u f h = tr st χ (f * st h u)

theorem **ps_apply** (u : R) (F : G → R) : ps st u F = ∑ h, F h * st h u

theorem **IsRightAct.tr_mem_eigen** {st : G → R →ₐ[K] R} (hs : IsRightAct st) (χ : G →* Kˣ) (F : R) : tr st χ F ∈ eigen st χ

theorem **tr_of_mem_eigen** {χ : G →* Kˣ} {F : R} (hF : F ∈ eigen st χ) : tr st χ F = (Fintype.card G : K) • F

theorem **tr_mem** {C : Submodule K R} (hC : ∀ h, ∀ x ∈ C, st h x ∈ C) (χ : G →* Kˣ) {x : R} (hx : x ∈ C) : tr st χ x ∈ C

theorem **mem_latCap** {L : Submodule O C} {W : Submodule K R} {F : C} : F ∈ latCap L W ↔ F ∈ L ∧ (F : R) ∈ W

theorem **latCap_le** (L : Submodule O C) (W : Submodule K R) : latCap L W ≤ L

theorem **mem_spanQ** {N : Submodule O C} {F : C} (hF : F ∈ N) : (F : R) ∈ spanQ N

theorem **mem_cutLat** {N : Submodule O C} {T : Module.End K R} {F : C} : F ∈ cutLat N T ↔ F ∈ N ∧ ∃ x ∈ spanQ N, T x = F

theorem **mem_cutLat_iff** {N : Submodule O C} {T : Module.End K R} (hT : IsIdempotentElem T) {F : C} : F ∈ cutLat N T ↔ F ∈ N ∧ T F = F

theorem **cutLat_one** (N : Submodule O C) : cutLat N 1 = N

theorem **cutLat_zero** (N : Submodule O C) : cutLat N 0 = ⊥

theorem **qProj_mk** (h : I ≤ J) (x : N) : qProj h (Submodule.Quotient.mk x) = Submodule.Quotient.mk x

theorem **qProj_surjective** (h : I ≤ J) : Function.Surjective (qProj (N

theorem **qProj_eq_zero** (h : I ≤ J) (q : QMod N I) : qProj h q = 0 ↔ q ∈ J • (⊤ : Submodule O (QMod N I))

theorem **map_mem_smul_top** {M M' : Type} [AddCommGroup M] [Module O M] [AddCommGroup M'] [Module O M'] (f : M →ₗ[O] M') {x : M} (hx : x ∈ J • (⊤ : Submodule O M)) : f x ∈ J • (⊤ : Submodule O M')

theorem **endRed_qProj** (h : I ≤ J) (γ : QMod N I →ₗ[O] QMod N I) (q : QMod N I) : endRed h γ (qProj h q) = qProj h (γ q)

theorem **autRed_qProj** (h : I ≤ J) (γ : QAut N I) (q : QMod N I) : autRed h γ (qProj h q) = qProj h (γ q)

theorem **mem_trivMod** {γ : QAut N I} : γ ∈ trivMod N I J ↔ ∀ q, γ q − q ∈ J • (⊤ : Submodule O (QMod N I))

theorem **autRed_eq_one** (h : I ≤ J) (γ : QAut N I) : autRed h γ = 1 ↔ γ ∈ trivMod N I J

theorem **trivMod_mono** {J' : Ideal O} (h : J ≤ J') : trivMod N I J ≤ trivMod N I J'

theorem **smul_top_qMod** : I • (⊤ : Submodule O (QMod N I)) = ⊥

theorem **trivMod_sup** : trivMod N I (I ⊔ J) = trivMod N I J

theorem **mem_iInf_smul_top** {ι M : Type} [Finite ι] [AddCommGroup M] [Module O M] {Js : ι → Ideal O} (hc : Pairwise (Function.onFun IsCoprime Js)) {x : M} (hx : ∀ i, x ∈ Js i • (⊤ : Submodule O M)) : x ∈ (⊓ i, Js i) • (⊤ : Submodule O M)

theorem **crt_apply** (q : QMod N I) (i : ι) : crt hle hc hI q i = qProj (hle i) q

theorem **crtAut_apply** (γ : QAut N I) (i : ι) : crtAut hle hc hI γ i = autRed (hle i) γ

theorem **endMat_mulVec** (γ : Module.End O M) (x : M) : (endMat e γ).mulVec (e x) = e (γ x)

theorem **autMat_mulVec** (γ : M ≃ₗ[O] M) (x : M) : (autMat e γ : Matrix (Fin r) (Fin r) (ZMod p)).mulVec (e x) = e (γ x)

theorem **autMat_injective** : Function.Injective (autMat (O

theorem **autMat_surjective** (hO : ∀ a : O, ∃ n : ℤ, ∀ x : M, a • x = n • x) : Function.Surjective (autMat (O

Development.AbelianVarietyTheory.Definitions.Divisors.IsSquareIntegrableAutomorphicSpace

theorem **IsCT.lin** : ∀ (u : List (Letter E)) (f : Vec u) (N m : ℕ) (h : wsize u + m = N) (c : ℂ) (ψ ψ' : Sm E m) (φ φ' : Sm E N), IsCT u f N m h ψ φ → IsCT u f N m h ψ' φ' → IsCT u f N m h (c • ψ + ψ') (c • φ + φ') | [], _, N, m, h, c, ψ, ψ', φ, φ', h1, h2 => by obtain rfl : m = N

theorem **IsCT.zero** : ∀ (u : List (Letter E)) (f : Vec u) (N m : ℕ) (h : wsize u + m = N), IsCT u f N m h 0 0 | [], _, N, m, h => by obtain rfl : m = N

Development.AbelianVarietyTheory.Definitions.Divisors.IsSquareIntegrableIffMemL2

theorem **kfin_iff** : FiniteDimensional ℂ (Submodule.span ℂ (Set.range fun k : K => fun g => φ (g * k))) ↔ GlobalTheory.Doubling.IsKFinite K φ

theorem **isAut_iff_isAutOn** : IsAut σ J K φ ↔ IsAutOn σ J (ratPts σ J) K φ

theorem **IsAutOn.mono** (h : IsAutOn σ J H K φ) (hH : H' ≤ H) (hK : K' ≤ K) : IsAutOn σ J H' K' φ

theorem **foldr_lineDer_char** (hφ : ∀ g h, φ (g * h) = φ g * φ h) (a : ℂ) (l : List (ℝ → GAd σ J)) : ∃ b : ℂ, l.foldr lineDer (a • φ) = b • φ

theorem **isAut_of_char** (K : Subgroup (GAd σ J)) (hφ : ∀ g h, φ (g * h) = φ g * φ h) (a : ℂ) (hm : IsSmoothModerate σ J (a • φ)) : IsAut σ J K (a • φ)

theorem **isAut_const** (K : Subgroup (GAd σ J)) (a : ℂ) : IsAut σ J K fun _ => a

theorem **isAutOn_const** (H K : Subgroup (GAd σ J)) (a : ℂ) : IsAutOn σ J H K fun _ => a

theorem **isMaxCpt_infPts** (h : IsCompact (infPts σ J : Set (GAd σ J))) : IsMaxCpt σ J (infPts σ J)

theorem **le_one_of_pow_mul_le** {t a M : ℝ} (ha : 0 < a) (h : ∀ k : ℕ, t ^ k * a ≤ M) : t ≤ 1

theorem **eq_one_of_powers_compact** {b Q : Matrix (Fin n) (Fin n) ℂ} {S : Set (Matrix (Fin n) (Fin n) ℂ)} (hS : IsCompact S) (hQ : Q * (bᴴ * b) = 1) (h1 : ∀ k : ℕ, (bᴴ * b) ^ k ∈ S) (h2 : ∀ k : ℕ, Q ^ k ∈ S) : bᴴ * b = 1

theorem **conj_mem_UJ** {R : Type} [CommRing R] {s : R →+* R} {A D : Matrix (Fin n) (Fin n) R} {c : GL (Fin n) R} (hc : Formalism.UG.cT s c.1 * A * c.1 = D) (g : GL (Fin n) R) : c⁻¹ * g * c ∈ Formalism.UG.UJ s D ↔ g ∈ Formalism.UG.UJ s A

theorem **mem_majCpt** {c : ∀ w : InfinitePlace F, GL (Fin n) (w.Completion ⊗[F] E)} {g : GAd σ J} : g ∈ majCpt σ J c ↔ g ∈ infPts σ J ∧ ∀ w, projArch σ J w g ∈ stabArch σ J w (c w)

theorem **isMaxCpt_majCpt** (h : IsCMExt F E σ) {c : ∀ w : InfinitePlace F, GL (Fin n) (w.Completion ⊗[F] E)} (hc : ∀ w, ∃ ε : Fin n → Bool, Formalism.UG.cT (sigR _ σ) (c w).1 * JR w.Completion J * (c w).1 = Matrix.diagonal fun i => if ε i then 1 else −1) : IsMaxCpt σ J (majCpt σ J c)

theorem **exists_isMaxCpt** (h : IsCMExt F E σ) (hJ : IsHermND σ J) : ∃ K, IsMaxCpt σ J K

theorem **isConstTerm_iff** : IsConstTerm σ J C φ c ↔ IsConstTermOn σ J (unipRad σ J C) φ c

theorem **isConstTerm_iff_unipFlag** : IsConstTerm σ J C φ c ↔ IsConstTermOn σ J (unipFlag σ J C fun _ => 0) φ c

theorem **isConstTermOn_const** (N : Subgroup (GAd σ J)) (a : ℂ) : IsConstTermOn σ J N (fun _ => a) fun _ => a

theorem **isConstTermOn_constTermOn** [LocallyCompactSpace N] (hφ : LeftInv (ratPts σ J ⊓ N) φ) : IsConstTermOn σ J N φ (constTermOn (ratPts σ J) N φ)

theorem **exists_mass_one** [LocallyCompactSpace N] (hc : Cocompact (ratPts σ J) N) : ∃ (μ : Measure N) (𝓕 : Set N), μ.IsHaarMeasure ∧ IsFundamentalDomain ((ratPts σ J).subgroupOf N) 𝓕 μ ∧ μ 𝓕 = 1

theorem **IsConstTermOn.unique** [LocallyCompactSpace N] (hc : Cocompact (ratPts σ J) N) (h : IsConstTermOn σ J N φ c) (h' : IsConstTermOn σ J N φ c') : c = c'

theorem **isConstTermOn_iff** [LocallyCompactSpace N] (hc : Cocompact (ratPts σ J) N) (hφ : LeftInv (ratPts σ J ⊓ N) φ) : IsConstTermOn σ J N φ c ↔ c = constTermOn (ratPts σ J) N φ

theorem **existsUnique_isConstTermOn** (d : Fin r → ℕ) (hφ : LeftInv (ratPts σ J ⊓ unipFlag σ J C d) φ) : ∃! c, IsConstTermOn σ J (unipFlag σ J C d) φ c

theorem **locallyCompact_unipRad** : LocallyCompactSpace (unipRad σ J C)

theorem **cocompact_unipRad** : Cocompact (ratPts σ J) (unipRad σ J C)

theorem **existsUnique_isConstTerm** (hφ : LeftInv (ratPts σ J) φ) : ∃! c, IsConstTerm σ J C φ c

theorem **IsConstTermOn.eq_constTermOn_par** {d : Fin r → ℕ} (hc : IsConstTermOn σ J (unipFlag σ J C d) φ c) (hφ : LeftInv (ratPts σ J ⊓ parFlag σ J C d) φ) [LocallyCompactSpace (unipFlag σ J C d)] : Cocompact (ratPts σ J ⊓ parFlag σ J C d) (unipFlag σ J C d) ∧ c = constTermOn (ratPts σ J ⊓ parFlag σ J C d) (unipFlag σ J C d) φ

theorem **IsConstTermOn.leftInv** {d : Fin r → ℕ} (hc : IsConstTermOn σ J (unipFlag σ J C d) φ c) (hφ : LeftInv (ratPts σ J ⊓ parFlag σ J C d) φ) : LeftInv (leftGrp σ J C d) c

theorem **IsConstTermOn.continuous** {d : Fin r → ℕ} (hc : IsConstTermOn σ J (unipFlag σ J C d) φ c) (hφ : LeftInv (ratPts σ J ⊓ parFlag σ J C d) φ) (hφc : Continuous φ) : Continuous c

theorem **isIsoCols_of_isWittFrame** {X Y : Matrix (Fin n) (Fin r) E} (hW : IsWittFrame σ J X Y) (hσ : ∀ x, σ (σ x) = x) (hJ : IsHermND σ J) {k : ℕ} (hk : k ≤ r) : IsIsoCols σ J (X.submatrix id (Fin.castLE hk))

Development.AbelianVarietyTheory.Definitions.Divisors.IsTaylorExpansionOrderOne

theorem **shA_apply** {l : Λ} {k k' : K.K} (h : K.le l k' k) (x : Hol (K.W k)) : shA h x = sh h x

theorem **forgetD_perCl** (per : Λ →+ Hol F.V) : (K.forgetD η M).pull1 (K.perCl per) = K.perClD η M (η.toAddMonoidHom.comp per)

theorem **Charts.dualD_J_ne_bot** (K : Charts F) (k : K.K) (v : I → ℂ) (p : Tot I g) (hp : p ∈ K.W k) : (K.dualD k v p hp).T.J ≠ ⊥

theorem **isOver_dual** (v z : I → ℂ) (hz : z ∈ F.V) {X : Scheme.{0}} (s : X ⟶ Sp ℂ) : IsOver (F

theorem **isTaylorExpansion_orderOne** [DecidableEq I] {W : Opens (Tot I g)} {W₀ : Opens (Tot Empty g)} (hW : ∀ p ∈ W₀, ((0 : I → ℂ), p.2) ∈ W) (v : I → ℂ) : IsTay 1 (fun i ↦ v i • (DualNumber.eps : DualNumber ℂ)) (tay1 hW v)

Development.AbelianVarietyTheory.Definitions.Divisors.IsTestLatUnique

theorem **finE_rat** (γ : GQ (Xp G)) (γE : UR D.sp.cm (Xp G).gram) (h : Bridge.IsRatMat (Xp G) γ γE.1) : Formalism.IndefR.Lev.finE (Xp G) γ = ratIn G γE

theorem **map_rhoVr** {θ : InfFam D} (hθ : IsInfFam θ) (V : HSp D) (lam : D.what₂.Tau → DomWeight V.rank) (γ : GQ V) (γE : GL (Fin V.rank) D.host.E) (h : Bridge.IsRatMat V γ γE) (x : GA D V) (v : VV D.what₂.LL lam) : rhoK lam (invTrK (realGL θ V (γ.1 ∗ x)⁻¹)) (mapVV D.what₂.embLL lam (rhoVr (labLL D) lam γE v)) = rhoK lam (invTrK (realGL θ V x⁻¹)) (mapVV D.what₂.embLL lam v)

theorem **scal_left** {θ : InfFam D} (hθ : IsInfFam θ) (ξ : Module.Dual ℂ (VV ℂ (LaP G))) (Ψ : GLf G → WtSp G) (hΨ : ∀ γ g, Ψ (ratIn G γ ∗ g) = rhoW G γ (Ψ g)) (γ : GQ (Xp G)) (γE : UR D.sp.cm (Xp G).gram) (h : Bridge.IsRatMat (Xp G) γ γE.1) (x : GA D (Xp G)) : scal θ ξ Ψ (γ.1 ∗ x) = scal θ ξ Ψ x

theorem **pair_rhoW** (γ : UR D.sp.cm (Xp G).gram) (x y : WtSp G) : pairK (LaP G) (gramKd (emb D) (Xp G).gram) (rhoW G γ x) (rhoW G γ y) = pairK (LaP G) (gramKd (emb D) (Xp G).gram) x y

theorem **mem_globLat_rat** (γ : GL (Fin n) E) (g : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (x : Fin n → E) : γ.1 ∗ᵥ x ∈ CS.globLat (bc (inclE (F

theorem **globLat_mul** [StarRing E] (g k : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (hk : k ∈ Formalism.UG.stabGL (Formalism.L2.Herm.hatL (F

theorem **mem_latOf_rat** (γ : UR D.sp.cm (Xp G).gram) (g : GLf G) (x : Fin (Xp G).rank → D.host.E) : γ.1.1 ∗ᵥ x ∈ latOf G (ratIn G γ ∗ g) ↔ x ∈ latOf G g

theorem **latOf_mul** (g : GLf G) {k : GLf G} (hk : k ∈ G.Kplus) : latOf G (g ∗ k) = latOf G g

theorem **psi_ne_zero** (rp : ClassReps G) (hN : rp.Neat) {n : ℕ} (e : Fin n ≃ TestIx G × VVIdx (LaP G)) (a : Fin n) : psi rp e a (rp.x (e a).1.2) ≠ 0

theorem **isTestLat_unique** {rp : ClassReps G} {n : ℕ} {e : Fin n ≃ TestIx G × VVIdx (LaP G)} {X X' M M'} (h : IsTestLat G rp n e X M) (h' : IsTestLat G rp n e X' M') : M = M'

theorem **exists_testLat** (h : (L.ΛY 0).FG) (rp : ClassReps G) : ∃ n e X M, 0 < n ∧ IsTestLat G rp n e X M

Development.AbelianVarietyTheory.Definitions.Divisors.IsTorsCharZero

theorem **hasRoots_one** : HasRoots A 1

theorem **isTorsChar_zero** (n : ℕ) : IsTorsChar (A

theorem **isSec_zero** (n : ℕ) {D : Type} [CommRing D] [Algebra R D] (x : Pt A D) : IsSec n (fun _ _ _ _ ↦ 0) x fun _ _ _ _ _ ↦ 0

theorem **diagSum_tmul** (D : Type) [CommRing D] [Bialgebra R D] [Coalgebra.IsCocomm R D] (m : ℕ) (a b : D) : diagSum R D m (a ⊗ₜ b) = ∑ i ∈ Finset.range m, mulBy R D i a ∗ b

Development.AbelianVarietyTheory.Definitions.Divisors.IsTorsionFreeOfForallExists

theorem **isCohSub_id** [h : IsFiniteQC F] : IsCohSub (𝟙 F)

theorem **isCohSub_of_isZero** (h : IsZero G) (i : G ⟶ F) : IsCohSub i

theorem **IsCohSub.comp** {i : G ⟶ F} (h : IsCohSub i) (j : F ⟶ H) [Mono j] : IsCohSub (i ≫ j)

theorem **isSubbundle_of_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) (h : IsVB S.X₃ s) : IsSubbundle S.f

theorem **isSubbundle_of_isIso** (i : G ⟶ F) [IsIso i] : IsSubbundle i

theorem **isSubbundle_inl** (h : IsVB G s) : IsSubbundle (biprod.inl : F ⟶ F ⊞ G)

theorem **exists_app_eq_of_epi** (φ : F ⟶ G) [Epi φ] (s : Γ(G, V)) {x : X} (hx : x ∈ V) : ∃ (W : X.Opens) (h : W ≤ V), x ∈ W ∧ ∃ t, φ.app W t = resM G h s

theorem **genericPoint_mem** {x : X} (hx : x ∈ V) : genericPoint X ∈ V

theorem **genRank_congr** (φ : F ≅ G) : genRank F = genRank G

theorem **IsVB.genRank_eq** (h : IsVB F r) : genRank F = r

theorem **genRank_eq_zero_iff** [IsFiniteQC F] : genRank F = 0 ↔ genericPoint X ∉ QCoh.support F

theorem **genRank_eq_zero_of_isZero** (h : IsZero F) : genRank F = 0

theorem **genRank_le_of_mono** (i : G ⟶ F) [Mono i] [IsFiniteQC F] : genRank G ≤ genRank F

theorem **genRank_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) [IsFiniteQC S.X₂] : genRank S.X₂ = genRank S.X₁ + genRank S.X₃

theorem **genRank_cokernel** (i : G ⟶ F) [Mono i] [h : IsFiniteQC F] : genRank F = genRank G + genRank (cokernel i)

theorem **Sub.mem_ofGeneric** {N : Submodule X.functionField (genStalk F)} {s : Γ(F, V)} : s ∈ (Sub.ofGeneric N).sec V ↔ ∀ h : genericPoint X ∈ V, F.presheaf.germ V _ h s ∈ N

theorem **Sub.range_stalkLinear_ofGeneric** (N : Submodule X.functionField (genStalk F)) : LinearMap.range (stalkLinear (Sub.ofGeneric N).incl (genericPoint X)) = N

theorem **IsTorsionFree.of_mono** (h : IsTorsionFree F) (i : G ⟶ F) [Mono i] : IsTorsionFree G

theorem **isTorsionFree_of_forall_exists** (H : ∀ (V : X.Opens) (h : genericPoint X ∈ V) (s : Γ(F, V)), F.presheaf.germ V _ h s = 0 → ∀ x ∈ V, ∃ (W : X.Opens) (h' : W ≤ V), x ∈ W ∧ resM F h' s = 0) : IsTorsionFree F

theorem **IsVB.isTorsionFree** (h : IsVB F r) : IsTorsionFree F

theorem **IsTorsionFree.eq_zero_of_smul** (h : IsTorsionFree F) {a : Γ(X, V)} (ha : a ≠ 0) {s : Γ(F, V)} (hs : a • s = 0) : s = 0

theorem **IsTorsionFree.isZero** [IsFiniteQC F] (h : IsTorsionFree F) (h0 : genRank F = 0) : IsZero F

@[reassoc (attr

theorem **range_stalkLinear_satι** (i : G ⟶ F) : LinearMap.range (stalkLinear (satι i) (genericPoint X)) = LinearMap.range (stalkLinear i (genericPoint X))

theorem **genRank_sat** (i : G ⟶ F) [Mono i] : genRank (sat i) = genRank G

theorem **mem_satSub_iff** (i : G ⟶ F) [F.IsQuasicoherent] [G.IsQuasicoherent] (hU : IsAffineOpen U) (hη : genericPoint X ∈ U) {s : Γ(F, U)} : s ∈ (satSub i).sec U ↔ ∃ a : Γ(X, U), a ≠ 0 ∧ ∃ t, i.app U t = a • s

theorem **germ_cokernel_π_eq_zero_iff** (i : G ⟶ F) (h : genericPoint X ∈ V) (s : Γ(F, V)) : (cokernel i).presheaf.germ V _ h ((cokernel.π i).app V s) = 0 ↔ s ∈ (satSub i).sec V

theorem **isSaturated_satι** (i : G ⟶ F) : IsSaturated (satι i)

theorem **IsSubbundle.isSaturated** {i : G ⟶ F} (h : IsSubbundle i) : IsSaturated i

theorem **isSaturated_iff_isIso_toSat** (i : G ⟶ F) [Mono i] : IsSaturated i ↔ IsIso (toSat i)

theorem **isIso_satι_of_genRank_eq** (i : G ⟶ F) [Mono i] [IsFiniteQC F] (h : genRank G = genRank F) : IsIso (satι i)

theorem **IsProjCurve.isProperCurve** (h : IsProjCurve k C) : IsProperCurve k C

theorem **IsProperCurve.isNoetherian** (h : IsProperCurve k C) : AlgebraicGeometry.IsNoetherian C

theorem **IsProjCurve.isNoetherian** (h : IsProjCurve k C) : AlgebraicGeometry.IsNoetherian C

theorem **IsProjCurve.finiteCoh** (h : IsProjCurve k C) (F : C.Modules) [IsFiniteQC F] : FiniteCoh k F

theorem **suppDim_le_zero** [IsIntegral C] (hd : topologicalKrullDim C = 1) {F : C.Modules} [IsFiniteQC F] (h : genRank F = 0) : QCoh.suppDim F ≤ 0

theorem **degOfRank_congr** (φ : F ≅ G) : degOfRank k F r = degOfRank k G r

theorem **degOfRank_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) (h₁ : FiniteCoh k S.X₁) (h₃ : FiniteCoh k S.X₃) : degOfRank k S.X₂ (r + s) = degOfRank k S.X₁ r + degOfRank k S.X₃ s

theorem **deg_def** : deg k F = chi k F − genRank F ∗ chi k (structureModule X)

theorem **IsVB.deg_eq** (h : IsVB F r) : deg k F = degOfRank k F r

theorem **deg_congr** (φ : F ≅ G) : deg k F = deg k G

theorem **deg_eq_zero_of_isZero** (h : IsZero F) : deg k F = 0

@[simp] theorem **deg_structureModule** : deg k (structureModule X) = 0

theorem **deg_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) [IsFiniteQC S.X₂] (h₁ : FiniteCoh k S.X₁) (h₃ : FiniteCoh k S.X₃) : deg k S.X₂ = deg k S.X₁ + deg k S.X₃

theorem **deg_freeModule** (h : FiniteCoh k (structureModule X)) (ι : Type) [Fintype ι] : deg k (freeModule X ι) = 0

theorem **IsFrame.deg_eq_zero** (h : FiniteCoh k (structureModule X)) {ι : Type} [Fintype ι] {e : ι → Γ(F, ⊤)} (he : IsFrame e) : deg k F = 0

theorem **IsProperCurve.deg_eq_hdim** (hC : IsProperCurve k X) [IsFiniteQC F] (h : genRank F = 0) : deg k F = hdim k F 0

theorem **deg_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) [IsFiniteQC S.X₁] [IsFiniteQC S.X₂] [IsFiniteQC S.X₃] : deg k S.X₂ = deg k S.X₁ + deg k S.X₃

theorem **deg_cokernel** (i : G ⟶ F) [Mono i] [IsFiniteQC G] [IsFiniteQC F] : deg k (cokernel i) = deg k F − deg k G

theorem **deg_le_of_genRank_eq** (i : G ⟶ F) [Mono i] [IsFiniteQC G] [IsFiniteQC F] (h : genRank G = genRank F) : deg k G ≤ deg k F

theorem **deg_le_deg_sat** (i : G ⟶ F) [Mono i] [IsFiniteQC G] [IsFiniteQC F] : deg k G ≤ deg k (sat i)

theorem **deg_eq_sum_eigen** {O ι : Type} [CommRing O] [Fintype ι] {σ : ι → (O →+∗ k)} (hσ : IsSplitBy σ) (ρ : O →+∗ End F) [IsFiniteQC F] : deg k F = ∑ j, deg k (eigen ρ (scal X (σ j))) ∧ genRank F = ∑ j, genRank (eigen ρ (scal X (σ j)))

theorem **degLB_eq** (hL : L.IsLB) : degLB k L = chi k (toMod L) − chi k (structureModule X)

theorem **degLB_congr** (hL : L.IsLB) (hM : M.IsLB) (h : LBIso L M) : degLB k L = degLB k M

theorem **IsVB.degLB_det_one** (h : IsVB F 1) : degLB k (det F 1) = deg k F

theorem **IsProjCurve.degLB_eq_interNum** (hC : IsProjCurve k X) (hL : L.IsLB) : degLB k L = interNum k (structureModule X) ![L]

theorem **IsProjCurve.degLB_of_cls_eq_add** (hC : IsProjCurve k X) (hN : N.IsLB) (hL : L.IsLB) (hM : M.IsLB) (e : CechPic.cls N hN = CechPic.cls L hL + CechPic.cls M hM) : degLB k N = degLB k L + degLB k M

@[simp] theorem **IsProjCurve.degPic_cls** (hC : IsProjCurve k X) (hL : L.IsLB) : hC.degPic (CechPic.cls L hL) = degLB k L

theorem **nonempty_pullback_toMod_iso** (hL : L.IsLB) (f : Y ⟶ X) : Nonempty ((Scheme.Modules.pullback f).obj (toMod L) ≅ toMod (L.pull f))

theorem **IsProjCurve.deg_frobPull_of_rank_one** (hC : IsProjCurve k X) (p : ℕ) [Fact p.Prime] (hp : (p : Γ(X, ⊤)) = 0) (h : IsVB F 1) : deg k ((frobPull p hp).obj F) = p ∗ deg k F

theorem **IsProjCurve.deg_dual_of_rank_one** (hC : IsProjCurve k X) (h : IsVB F 1) : deg k (dual F) = −deg k F

theorem **IsCohSub.deg_mem** {i : G ⟶ F} (h : IsCohSub i) : deg k G ∈ subDegs k F

theorem **zero_mem_subDegs** (F : X.Modules) : 0 ∈ subDegs k F

theorem **subDegs_mono** (j : F ⟶ H) [Mono j] : subDegs k F ⊆ subDegs k H

theorem **subDegs_congr** (φ : F ≅ H) : subDegs k F = subDegs k H

theorem **exists_isCohSub** (h : IsMaxSubDeg k F P) : ∃ (G : X.Modules) (i : G ⟶ F), IsCohSub i ∧ deg k G = P

theorem **deg_le** (h : IsMaxSubDeg k F P) (i : G ⟶ F) [Mono i] [hG : IsFiniteQC G] : deg k G ≤ P

theorem **nonneg** (h : IsMaxSubDeg k F P) : 0 ≤ P

theorem **unique** (h : IsMaxSubDeg k F P) (h' : IsMaxSubDeg k F Q) : P = Q

theorem **of_iso** (h : IsMaxSubDeg k F P) (φ : F ≅ H) : IsMaxSubDeg k H P

theorem **exists_isMaxSubDeg_iff** : (∃ P, IsMaxSubDeg k F P) ↔ BddAbove (subDegs k F)

theorem **IsProjCurve.isMaxSubDeg_of_rank_one** (hC : IsProjCurve k X) (h : IsVB F 1) : IsMaxSubDeg k F (max (deg k F) 0)

theorem **isSmoothProjCurve_line** : IsSmoothProjCurve k (PSpace k 1)

theorem **isVB_sheafO** (n : ℤ) : IsVB (sheafO k 1 n) 1

theorem **deg_sheafO** (n : ℤ) : deg k (sheafO k 1 n) = n

theorem **degLB_On** (n : ℤ) : degLB k (On k 1 n) = n

Development.AbelianVarietyTheory.Definitions.Divisors.IsUnitMkOfFormallyUnramified

theorem **isUnit_mk_of_formallyUnramified** {k A : Type*} [CommRing k] [CommRing A] {α : k →+* A} {D : ℕ} {ω : Forms α D} (hω : IsFreeGen A ω) (f : Fin D → A) {J : A} (hJ : J • ω = dForm α f) (hu : ((Ideal.Quotient.mk (Ideal.span (Set.range f))).comp α).FormallyUnramified) : IsUnit (Ideal.Quotient.mk (Ideal.span (Set.range f)) J)

theorem **resG_comp_algX** {U V : Y.Opens} (h : V ≤ U) : (resG Y h).comp (algX p U) = algX p V

@[simp] theorem **resForm_dOn** {U V : Y.Opens} (h : V ≤ U) (f : Fin D → Γ(Y, U)) : resForm h (dOn p f) = dOn p fun i => resG Y h (f i)

@[simp] theorem **resForm_resForm** {U V W : Y.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (η : FormsOn p D U) : resForm h₂ (resForm h₁ η) = resForm (h₂.trans h₁) η

@[simp] theorem **resForm_self** {U : Y.Opens} (h : U ≤ U) (η : FormsOn p D U) : resForm h η = η

theorem **resForm_smul** {U V : Y.Opens} (h : V ≤ U) (a : Γ(Y, U)) (η : FormsOn p D U) : resForm h (a • η) = resG Y h a • resForm h η

theorem **etale_resG** {U V : Y.Opens} (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) : (resG Y h).Etale

theorem **le_preimage_inf** {T : Scheme.{0}} {y : T ⟶ Y} {U U' : Y.Opens} {W : T.Opens} (h : W ≤ y ⁻¹ᵁ U) (h' : W ≤ y ⁻¹ᵁ U') : W ≤ y ⁻¹ᵁ (U ⊓ U')

theorem **le_iSup_affine** (V : Y.Opens) : V ≤ ⨆ W : {W : Y.Opens // IsAffineOpen W ∧ W ≤ V}, W.1

theorem **fun_ext_affine** {V : Y.Opens} {a b : Γ(Y, V)} (h : ∀ W : Y.Opens, IsAffineOpen W → ∀ hW : W ≤ V, resG Y hW a = resG Y hW b) : a = b

theorem **res** (hω : IsFrame ω) (hW : IsAffineOpen W) (h : W ≤ U) : IsFrame (resForm h ω)

private theorem **resG_resG_divisors** {U V W : Y.Opens} (h : V ≤ U) (h' : W ≤ V) (a : Γ(Y, U)) : resG Y h' (resG Y h a) = resG Y (h'.trans h) a

theorem **exists_coeff** (hω : IsFrame ω) (h : V ≤ U) (η : FormsOn p D V) : ∃ c : Γ(Y, V), ∀ W : Y.Opens, IsAffineOpen W → ∀ hW : W ≤ V, resForm hW η = resG Y hW c • resForm (hW.trans h) ω

theorem **coeff_unique** (hω : IsFrame ω) (h : V ≤ U) {η : FormsOn p D V} {c c' : Γ(Y, V)} (hc : ∀ W : Y.Opens, IsAffineOpen W → ∀ hW : W ≤ V, resForm hW η = resG Y hW c • resForm (hW.trans h) ω) (hc' : ∀ W : Y.Opens, IsAffineOpen W → ∀ hW : W ≤ V, resForm hW η = resG Y hW c' • resForm (hW.trans h) ω) : c = c'

theorem **resForm_eq_coeff_smul** (hω : IsFrame ω) (h : V ≤ U) (η : FormsOn p D V) (hW : IsAffineOpen W) (e : W ≤ V) : resForm e η = resG Y e (hω.coeff h η) • resForm (e.trans h) ω

theorem **coeff_eq_iff** (hω : IsFrame ω) (h : V ≤ U) {η : FormsOn p D V} {c : Γ(Y, V)} : hω.coeff h η = c ↔ ∀ W : Y.Opens, IsAffineOpen W → ∀ hW : W ≤ V, resForm hW η = resG Y hW c • resForm (hW.trans h) ω

theorem **coeff_smul_resForm** (hω : IsFrame ω) (hV : IsAffineOpen V) (h : V ≤ U) (η : FormsOn p D V) : hω.coeff h η • resForm h ω = η

@[simp] theorem **resG_coeff** (hω : IsFrame ω) (h : V ≤ U) (e : W ≤ V) (η : FormsOn p D V) : resG Y e (hω.coeff h η) = hω.coeff (e.trans h) (resForm e η)

@[simp] theorem **coeff_self** (hω : IsFrame ω) (h : V ≤ U) : hω.coeff h (resForm h ω) = 1

theorem **coeff_eq_of_eq_smul** (hω : IsFrame ω) (h : V ≤ U) {η : FormsOn p D V} {c : Γ(Y, V)} (e : η = c • resForm h ω) : hω.coeff h η = c

theorem **bijective_coeff** (hω : IsFrame ω) (hV : IsAffineOpen V) (h : V ≤ U) : Function.Bijective (hω.coeff h)

theorem **coeff_change** (hω : IsFrame ω) (hω' : IsFrame ω') (h : V ≤ U) (h' : V ≤ U') (η : FormsOn p D V) : hω.coeff h η = hω.coeff h (resForm h' ω') ∗ hω'.coeff h' η

theorem **resG_ratio** (hω : IsFrame ω) (hω' : IsFrame ω') (h : V ≤ U ⊓ U') : resG Y h (hω.ratio hω' : Γ(Y, U ⊓ U')) = hω.coeff (h.trans inf_le_left) (resForm (h.trans inf_le_right) ω')

theorem **resForm_eq_ratio_smul** (hω : IsFrame ω) (hω' : IsFrame ω') (hW : IsAffineOpen W) (h : W ≤ U) (h' : W ≤ U') : resForm h' ω' = resG Y (le_inf h h') (hω.ratio hω' : Γ(Y, U ⊓ U')) • resForm h ω

theorem **res_ratio_self** (hω : IsFrame ω) (h : V ≤ U ⊓ U) : Semi.res h (hω.ratio hω) = 1

theorem **res_ratio_mul** (hω : IsFrame ω) (hω' : IsFrame ω') (hω'' : IsFrame ω'') (h₁ : V ≤ U ⊓ U') (h₂ : V ≤ U' ⊓ U'') (h₃ : V ≤ U ⊓ U'') : Semi.res h₁ (hω.ratio hω') ∗ Semi.res h₂ (hω'.ratio hω'') = Semi.res h₃ (hω.ratio hω'')

theorem **exists_isFrame** [SmoothOfRelativeDimension D p] (x : Y) : ∃ (U : Y.Opens) (ω : FormsOn p D U), x ∈ U ∧ IsFrame ω

theorem **isFrame** (i : K.bundle.ι) : IsFrame (K.om i)

theorem **g_eq_ratio** (i j : K.bundle.ι) : K.bundle.g i j = (K.isFrame i).ratio (K.isFrame j)

theorem **coeff_eq_iff** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) {η : FormsOn p D V} {c : Γ(Y, V)} : K.coeff i h η = c ↔ ∀ W : Y.Opens, IsAffineOpen W → ∀ hW : W ≤ V, resForm hW η = resG Y hW c • resForm (hW.trans h) (K.om i)

theorem **coeff_smul_om** (i : K.bundle.ι) (hV : IsAffineOpen V) (h : V ≤ K.bundle.U i) (η : FormsOn p D V) : K.coeff i h η • resForm h (K.om i) = η

@[simp] theorem **resG_coeff** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (e : W ≤ V) (η : FormsOn p D V) : resG Y e (K.coeff i h η) = K.coeff i (e.trans h) (resForm e η)

@[simp] theorem **coeff_om** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) : K.coeff i h (resForm h (K.om i)) = 1

theorem **coeff_change** (i j : K.bundle.ι) (hi : V ≤ K.bundle.U i) (hj : V ≤ K.bundle.U j) (η : FormsOn p D V) : K.coeff i hi η = resG Y (le_inf hi hj) (K.bundle.g i j : Γ(Y, K.bundle.U i ⊓ K.bundle.U j)) * K.coeff j hj η

@[simp] theorem **coord_toSec** (η : FormsOn p D V) (i : K.bundle.ι) : coord (K.toSec V η) i = K.coeff i inf_le_right (resForm inf_le_left η)

theorem **chartEquiv_toSec** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (η : FormsOn p D V) : chartEquiv K.isLB i h (K.toSec V η) = K.coeff i h η

theorem **map_toSec** (e : W ≤ V) (η : FormsOn p D V) : (toMod K.bundle).presheaf.map (homOfLE e).op (K.toSec V η) = K.toSec W (resForm e η)

theorem **bijective_toSec** (i : K.bundle.ι) (hV : IsAffineOpen V) (h : V ≤ K.bundle.U i) : Function.Bijective (K.toSec V)

theorem **toSec_om** (i : K.bundle.ι) : K.toSec _ (K.om i) = chartSec K.isLB i

theorem **weak_admissible_iso** (a : K'.bundle.ι) (b : K.bundle.ι) : (K.iso K').s (a, b) = (K'.isFrame a).ratio (K.isFrame b)

private theorem **units_eq_mul_inv_mul_of_mul_eq** {R : Type*} [CommRing R] (x g h y : Rˣ) (e : (x : R) * h = g * y) : x = g * h⁻¹ * y

private theorem **frame_ratio_res_mul_g_eq_g_mul_ratio_res** (K K' : CanonLB p D) (a1 b1 : K'.bundle.ι) (a2 b2 : K.bundle.ι) {O : Y.Opens} (hO₁ : O ≤ K'.bundle.U a1 ⊓ K.bundle.U a2) (hO₂ : O ≤ K.bundle.U a2 ⊓ K.bundle.U b2) (hO₃ : O ≤ K'.bundle.U a1 ⊓ K'.bundle.U b1) (hO₄ : O ≤ K'.bundle.U b1 ⊓ K.bundle.U b2) : (res hO₁ ((K'.isFrame a1).ratio (K.isFrame a2)) : Γ(Y, O)) * (res hO₂ (K.bundle.g a2 b2) : Γ(Y, O)) = (res hO₃ (K'.bundle.g a1 b1) : Γ(Y, O)) * (res hO₄ ((K'.isFrame b1).ratio (K.isFrame b2)) : Γ(Y, O))

theorem **iso_isGen** : (K.iso K').IsGen

theorem **lbIso** : LBIso K.bundle K'.bundle

theorem **coeff_eq_iso_mul** (a : K'.bundle.ι) (b : K.bundle.ι) (ha : V ≤ K'.bundle.U a) (hb : V ≤ K.bundle.U b) (η : FormsOn p D V) : K'.coeff a ha η = resG Y (le_inf ha hb) ((K'.isFrame a).ratio (K.isFrame b)).val * K.coeff b hb η

private theorem **eq_of_resG_eq_on_affine_cover** {ι : Type} (U : ι → Y.Opens) (hU : ⨆ i, U i = ⊤) {V : Y.Opens} {a b : Γ(Y, V)} (h : ∀ (W : Y.Opens) (i : ι), IsAffineOpen W → ∀ (hW : W ≤ V), W ≤ U i → resG Y hW a = resG Y hW b) : a = b

theorem **isoOfGen_toSec** (η : FormsOn p D V) : (isoOfGen K.isLB K'.isLB (K.iso K') (K.iso_isGen K')).hom.app V (K.toSec V η) = K'.toSec V η

theorem **jacobian_smul_om** (i : K.bundle.ι) (hV : IsAffineOpen V) (h : V ≤ K.bundle.U i) (f : Fin D → Γ(Y, V)) : K.jacobian i h f • resForm h (K.om i) = dOn p f

@[simp] theorem **resG_jacobian** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (e : W ≤ V) (f : Fin D → Γ(Y, V)) : resG Y e (K.jacobian i h f) = K.jacobian i (e.trans h) fun a => resG Y e (f a)

theorem **jacobian_change** (i j : K.bundle.ι) (hi : V ≤ K.bundle.U i) (hj : V ≤ K.bundle.U j) (f : Fin D → Γ(Y, V)) : K.jacobian i hi f = resG Y (le_inf hi hj) (K.bundle.g i j : Γ(Y, K.bundle.U i ⊓ K.bundle.U j)) * K.jacobian j hj f

theorem **jacobian_mul_sub_mem** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (I : Ideal Γ(Y, V)) (u : Γ(Y, V)) (f : Fin D → Γ(Y, V)) (hf : ∀ a, f a ∈ I) : K.jacobian i h (fun a => u * f a) - u ^ D * K.jacobian i h f ∈ I

theorem **map_jacobian_mul** {R : Type*} [CommRing R] (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (ψ : Γ(Y, V) →+* R) (u : Γ(Y, V)) (f : Fin D → Γ(Y, V)) (hf : ∀ a, ψ (f a) = 0) : ψ (K.jacobian i h fun a => u * f a) = ψ u ^ D * ψ (K.jacobian i h f)

theorem **jacobian_eq_det_mul** (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (f x : Fin D → Γ(Y, V)) (m : Matrix (Fin D) (Fin D) Γ(Y, V)) (e : ∀ a, dFun (algX p V) (f a) = ∑ b, m a b • dFun (algX p V) (x b)) : K.jacobian i h f = m.det * K.jacobian i h x

theorem **isUnit_jacobian_iff** (i : K.bundle.ι) (hV : IsAffineOpen V) (h : V ≤ K.bundle.U i) (x : Fin D → Γ(Y, V)) : IsUnit (K.jacobian i h x) ↔ IsFrame (dOn p x)

theorem **VanishesAlong.appLE_resG** {s : LB.Sec L} {y : T ⟶ Y} (hs : VanishesAlong s y) (j : L.ι) {V : Y.Opens} (h : V ≤ L.U j) {W : T.Opens} (e : W ≤ y ⁻¹ᵁ V) : (y.appLE V W e).hom (resG Y h (s.f j)) = 0

theorem **vanishesAlong_iff_evalPt** {R : Type} [CommRing R] (s : LB.Sec L) (y : Sp R ⟶ Y) (j : L.ι) (h : ⊤ ≤ y ⁻¹ᵁ L.U j) : VanishesAlong s y ↔ evalPt y _ h (s.f j) = 0

theorem **lbPow_U_le** (n : ℕ) (j : (lbPow L n).ι) : (lbPow L n).U j ≤ L.U (j PUnit.unit)

private theorem **lbPow_g_res_eq_pow** (L : LB Y) (n : ℕ) (j j' : (lbPow L n).ι) {W : Y.Opens} (h : W ≤ (lbPow L n).U j ⊓ (lbPow L n).U j') (h₂ : W ≤ L.U (j PUnit.unit) ⊓ L.U (j' PUnit.unit)) : resG Y h ((lbPow L n).g j j').val = resG Y h₂ (L.g (j PUnit.unit) (j' PUnit.unit)).val ^ n

theorem **resG_tens_lbPow_g** (M : LB Y) (n : ℕ) (a b : (tens M (lbPow L n)).ι) {W : Y.Opens} (h : W ≤ (tens M (lbPow L n)).U a ⊓ (tens M (lbPow L n)).U b) (h₁ : W ≤ M.U a.1 ⊓ M.U b.1) (h₂ : W ≤ L.U (a.2 PUnit.unit) ⊓ L.U (b.2 PUnit.unit)) : resG Y h ((tens M (lbPow L n)).g a b).val = resG Y h₁ (M.g a.1 b.1).val * resG Y h₂ (L.g (a.2 PUnit.unit) (b.2 PUnit.unit)).val ^ n

theorem **isUnit_map_jacobian** {R : Type*} [CommRing R] (i : K.bundle.ι) (hV : IsAffineOpen V) (h : V ≤ K.bundle.U i) (f : Fin D → Γ(Y, V)) (hu : ((Ideal.Quotient.mk (Ideal.span (Set.range f))).comp (algX p V)).FormallyUnramified) (ψ : Γ(Y, V) →+* R) (hf : ∀ a, ψ (f a) = 0) : IsUnit (ψ (K.jacobian i h f))

theorem **isLB_adj** (hL : L.IsLB) (n : ℕ) : (K.adj L n).IsLB

theorem **resG_jacFun** (s : Fin D → LB.Sec L) (i : K.bundle.ι) (j : L.ι) (h : V ≤ K.bundle.U i ⊓ L.U j) : resG Y h (K.jacFun s i j) = K.jacobian i (h.trans inf_le_left) fun a => resG Y (h.trans inf_le_right) ((s a).f j)

theorem **jacFun_change_sub_mem** (s : Fin D → LB.Sec L) (i i' : K.bundle.ι) (j j' : L.ι) (h : V ≤ K.bundle.U i ⊓ L.U j) (h' : V ≤ K.bundle.U i' ⊓ L.U j') (h₁ : V ≤ K.bundle.U i ⊓ K.bundle.U i') (h₂ : V ≤ L.U j ⊓ L.U j') (I : Ideal Γ(Y, V)) (hI : ∀ a, resG Y (h'.trans inf_le_right) ((s a).f j') ∈ I) : resG Y h (K.jacFun s i j) - resG Y h₁ (K.bundle.g i i').val * resG Y h₂ (L.g j j').val ^ D * resG Y h' (K.jacFun s i' j') ∈ I

theorem **map_jacFun_change** {R : Type*} [CommRing R] (s : Fin D → LB.Sec L) (i i' : K.bundle.ι) (j j' : L.ι) (h : V ≤ K.bundle.U i ⊓ L.U j) (h' : V ≤ K.bundle.U i' ⊓ L.U j') (h₁ : V ≤ K.bundle.U i ⊓ K.bundle.U i') (h₂ : V ≤ L.U j ⊓ L.U j') (ψ : Γ(Y, V) →+* R) (hψ : ∀ a, ψ (resG Y (h'.trans inf_le_right) ((s a).f j')) = 0) : ψ (resG Y h (K.jacFun s i j)) = ψ (resG Y h₁ (K.bundle.g i i').val) * ψ (resG Y h₂ (L.g j j').val) ^ D * ψ (resG Y h' (K.jacFun s i' j'))

theorem **adj_U_le** (n : ℕ) (a : (K.adj L n).ι) : (K.adj L n).U a ≤ K.bundle.U a.1 ⊓ L.U (a.2 PUnit.unit)

theorem **dSec_f** {T : Scheme.{0}} (s : Fin D → LB.Sec L) (y : T ⟶ Y) (hs : ∀ a, VanishesAlong (s a) y) (c : ((K.adj L D).pull y).ι) : (K.dSec s y hs).f c = (y.appLE _ _ ((pull_U_le y c).trans (y.preimage_mono (K.adj_U_le D (c PUnit.unit))))).hom (K.jacFun s (c PUnit.unit).1 ((c PUnit.unit).2 PUnit.unit))

theorem **jacVal_eq** (s : Fin D → LB.Sec L) (y : Sp F ⟶ Y) (i : K.bundle.ι) (j : L.ι) (hi : ⊤ ≤ y ⁻¹ᵁ K.bundle.U i) (hj : ⊤ ≤ y ⁻¹ᵁ L.U j) (h₁ : V ≤ K.bundle.U i) (h₂ : V ≤ L.U j) (hy : ⊤ ≤ y ⁻¹ᵁ V) : K.jacVal s y i j hi hj = evalPt y V hy (K.jacobian i h₁ fun a => resG Y h₂ ((s a).f j))

theorem **jacVal_change** (s : Fin D → LB.Sec L) (y : Sp F ⟶ Y) (hs : ∀ a, VanishesAlong (s a) y) (i i' : K.bundle.ι) (j j' : L.ι) (hi : ⊤ ≤ y ⁻¹ᵁ K.bundle.U i) (hi' : ⊤ ≤ y ⁻¹ᵁ K.bundle.U i') (hj : ⊤ ≤ y ⁻¹ᵁ L.U j) (hj' : ⊤ ≤ y ⁻¹ᵁ L.U j') : K.jacVal s y i j hi hj = evalPt y _ (le_preimage_inf hi hi') (K.bundle.g i i').val * evalPt y _ (le_preimage_inf hj hj') (L.g j j').val ^ D * K.jacVal s y i' j' hi' hj'

theorem **jacVal_eq_pow_mul** (s : Fin D → LB.Sec L) (y : Sp F ⟶ Y) (i : K.bundle.ι) (j : L.ι) (hi : ⊤ ≤ y ⁻¹ᵁ K.bundle.U i) (hj : ⊤ ≤ y ⁻¹ᵁ L.U j) (h₁ : V ≤ K.bundle.U i) (h₂ : V ≤ L.U j) (hy : ⊤ ≤ y ⁻¹ᵁ V) (u : Γ(Y, V)) (f : Fin D → Γ(Y, V)) (e : ∀ a, resG Y h₂ ((s a).f j) = u * f a) (hf : ∀ a, evalPt y V hy (f a) = 0) : K.jacVal s y i j hi hj = evalPt y V hy u ^ D * evalPt y V hy (K.jacobian i h₁ f)

theorem **jacVal_eq_iso_mul** (s : Fin D → LB.Sec L) (y : Sp F ⟶ Y) (a : K'.bundle.ι) (b : K.bundle.ι) (j : L.ι) (ha : ⊤ ≤ y ⁻¹ᵁ K'.bundle.U a) (hb : ⊤ ≤ y ⁻¹ᵁ K.bundle.U b) (hj : ⊤ ≤ y ⁻¹ᵁ L.U j) : K'.jacVal s y a j ha hj = evalPt y _ (le_preimage_inf ha hb) ((K'.isFrame a).ratio (K.isFrame b)).val * K.jacVal s y b j hb hj

theorem **dSec_f_eq_jacVal** (s : Fin D → LB.Sec L) (y : Sp F ⟶ Y) (hs : ∀ a, VanishesAlong (s a) y) (c : ((K.adj L D).pull y).ι) (hc : ⊤ ≤ ((K.adj L D).pull y).U c) (hi : ⊤ ≤ y ⁻¹ᵁ K.bundle.U (c PUnit.unit).1) (hj : ⊤ ≤ y ⁻¹ᵁ L.U ((c PUnit.unit).2 PUnit.unit)) : (Scheme.ΓSpecIso (CommRingCat.of F)).hom.hom (resG _ hc ((K.dSec s y hs).f c)) = K.jacVal s y _ _ hi hj

theorem **isUnit_jacVal** (s : Fin D → LB.Sec L) (y : Sp F ⟶ Y) (hs : ∀ a, VanishesAlong (s a) y) (i : K.bundle.ι) (j : L.ι) (hi : ⊤ ≤ y ⁻¹ᵁ K.bundle.U i) (hj : ⊤ ≤ y ⁻¹ᵁ L.U j) (hV : IsAffineOpen V) (h₁ : V ≤ K.bundle.U i) (h₂ : V ≤ L.U j) (hy : ⊤ ≤ y ⁻¹ᵁ V) (hu : ((Ideal.Quotient.mk (Ideal.span (Set.range fun a => resG Y h₂ ((s a).f j)))).comp (algX p V)).FormallyUnramified) : IsUnit (K.jacVal s y i j hi hj)

theorem **jacOn_eq** (f : Fin (D + 1) → Γ(Y, U)) : jacOn p f = ∑ a : Fin (D + 1), ((-1 : Γ(Y, U)) ^ (a : ℕ) * f a) • dOn p fun b => f (a.succAbove b)

@[simp] theorem **resForm_jacOn** (h : V ≤ U) (f : Fin (D + 1) → Γ(Y, U)) : resForm h (jacOn p f) = jacOn p fun a => resG Y h (f a)

theorem **jacOn_mul** (u : Γ(Y, U)) (f : Fin (D + 1) → Γ(Y, U)) : jacOn p (fun a => u * f a) = u ^ (D + 1) • jacOn p f

theorem **resG_jacSec_f** (s : Fin (D + 1) → LB.Sec L) (c : (K.adj L (D + 1)).ι) (h : V ≤ (K.adj L (D + 1)).U c) (h₁ : V ≤ K.bundle.U c.1) (h₂ : V ≤ L.U (c.2 PUnit.unit)) : resG Y h ((K.jacSec s).f c) = K.coeff c.1 h₁ (jacOn p fun a => resG Y h₂ ((s a).f (c.2 PUnit.unit)))

theorem **map_coeff_jacOn** {R : Type*} [CommRing R] (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (ψ : Γ(Y, V) →+* R) (f : Fin (D + 1) → Γ(Y, V)) (hf : ∀ a : Fin D, ψ (f a.succ) = 0) : ψ (K.coeff i h (jacOn p f)) = ψ (f 0) * ψ (K.jacobian i h fun a => f a.succ)

theorem **evalPt_jacSec_f** {F : Type} [CommRing F] (s : Fin (D + 1) → LB.Sec L) (y : Sp F ⟶ Y) (hs : ∀ a : Fin D, VanishesAlong (s a.succ) y) (c : (K.adj L (D + 1)).ι) (hc : ⊤ ≤ y ⁻¹ᵁ (K.adj L (D + 1)).U c) (hi : ⊤ ≤ y ⁻¹ᵁ K.bundle.U c.1) (hj : ⊤ ≤ y ⁻¹ᵁ L.U (c.2 PUnit.unit)) : evalPt y _ hc ((K.jacSec s).f c) = evalPt y _ hj ((s 0).f (c.2 PUnit.unit)) * K.jacVal (fun a => s a.succ) y _ _ hi hj

theorem **jacobian_inv** {p : Y ⟶ Sp k} [SmoothOfRelativeDimension 1 p] (K : CanonLB p 1) {V : Y.Opens} (i : K.bundle.ι) (h : V ≤ K.bundle.U i) (x y : Γ(Y, V)) (e : x * y = 1) : K.jacobian i h ![y] = -y ^ 2 * K.jacobian i h ![x]

theorem **polyFn_comp** : (polyFn (k

private theorem **dFun_polyFn_eq_sum_pderiv** (g : MvPolynomial (Fin D) k) : dFun (algX (affStr k D) ⊤) (polyFn g) = ∑ b, polyFn (pderiv b g) • dFun (algX (affStr k D) ⊤) (polyFn (X b))

theorem **isFrame_dX** : IsFrame (dOn (affStr k D) fun a => polyFn (X a))

theorem **jacobian_affineSpace** (f : Fin D → MvPolynomial (Fin D) k) : (affineSpace k D).jacobian PUnit.unit le_rfl (fun a => polyFn (f a)) = polyFn (Matrix.of fun a b => pderiv b (f a)).det

theorem **jacVal_affineSpace** {F : Type} [CommRing F] (φ : MvPolynomial (Fin D) k →+* F) (f : Fin D → MvPolynomial (Fin D) k) : (affineSpace k D).jacVal (fun a => polySec (f a)) (spMap φ) PUnit.unit PUnit.unit le_rfl le_rfl = φ (Matrix.of fun a b => pderiv b (f a)).det

Development.AbelianVarietyTheory.Definitions.Divisors.IsValCharTwist

theorem **e_add_v** (τ : D.what₂.Tau) (j : Fin D.R.m) : 0 < (D.hc τ).e j + (D.hc τ).v (0, by have

theorem **gap_le** (C : ℝ) (i j : Fin D.R.m) (τ : D.what₂.Tau) : (gap D C : ℤ) ≤ (num D C τ).e (j + 1) + cen D C τ (D.R.m - i)

theorem **kap_add** (C : ℝ) (i j : Fin D.R.m) (τ : D.what₂.Tau) {s : ℕ} (hs : s ≤ halfRank D i j) : (kap D C i j s τ : ℤ) + s = (num D C τ).e (j + 1) + cen D C τ (D.R.m - i) ∧ 0 < kap D C i j s τ

theorem **expIn_condF** (P : Places L) (v : HeightOneSpectrum (𝒪 D.host.F)) : GlobalTheory.CMVal.expIn (condF P) v = if v ∈ P.S0 then condExp P v else 0

theorem **levelLe_mono** {χ : HChar D.host.E} {v : HeightOneSpectrum (𝒪 D.host.E)} {m n : ℕ} (h : χ.LevelLe v m) (hmn : m ≤ n) : χ.LevelLe v n

theorem **expAt_div** (G : GridData L sc) {k : ℕ} (h1 : 1 ≤ k) (h2 : k ≤ D.R.m) {θ : HChar D.host.E} (ht : ∀ P, θ.TameAt P) (hu : ∀ P : HeightOneSpectrum (𝒪 D.host.E), P.under (𝒪 D.host.F) ∉ S1 D → θ.UnrAt P) (P : HeightOneSpectrum (𝒪 D.host.E)) : GlobalTheory.CMVal.ExpAt (θ / G.ch.chiH k) P (GlobalTheory.CMVal.expIn (condF G.pl) (P.under (𝒪 D.host.F)))

theorem **isValChar_twist** {μ₀ : T} (hL : L.Is μ₀) (G : GridData L sc) (i j : Fin D.R.m) {s : ℕ} (hs : s ≤ halfRank D i j) {ξ : HChar D.host.E} (hf : ∃ n, 0 < n ∧ ξ.OrderDvd n) (htr : ξ.TrivOn D.host.F) (ht : ∀ P, ξ.TameAt P) (hu : ∀ P : HeightOneSpectrum (𝒪 D.host.E), P.under (𝒪 D.host.F) ∉ S1 D → ξ.UnrAt P) : GlobalTheory.CMVal.IsValChar (cmPhi D) s (kap D sc.cGap i j s) (condF G.pl) ∅ (nuH G μ₀ i j * ξ)

theorem **isValChar_nu** {μ₀ : T} (hL : L.Is μ₀) (G : GridData L sc) (i j : Fin D.R.m) {s : ℕ} (hs : s ≤ halfRank D i j) : GlobalTheory.CMVal.IsValChar (cmPhi D) s (kap D sc.cGap i j s) (condF G.pl) ∅ (nuH G μ₀ i j) ∧ ∀ c, GlobalTheory.CMVal.IsValChar (cmPhi D) s (kap D sc.cGap i j s) (condF G.pl) ∅ (nuH G μ₀ i j * delOf L c)

theorem **gauge_unique** {G : GridData L sc} {t : T} {i j : Fin D.R.m} {s : ℕ} {zd zd' g g'} (h : IsTorVal G t i j s zd g) (h' : IsTorVal G t i j s zd' g') : g = g'

theorem **bot_ranks** (i j : Fin D.R.m) : 2 ≤ botSmall D i j ∧ botSmall D i j < botBig D i j ∧ Even (botBig D i j + botSmall D i j) ∧ botBig D i j + botSmall D i j ≤ 2 * bigN D

Development.AbelianVarietyTheory.Definitions.Divisors.IsoOverOfIsOpenImmersion

theorem **memTop** (y : Y) : y ∈ (⊤ : Y.Opens)

theorem **Regular.injective_mul** {τ : H0 p P O} (hτ : Regular τ) (C : ProdChart P B N) : Function.Injective (C.mul τ)

theorem **Regular.exists_mul_eq** {τ : H0 p P O} (hτ : Regular τ) (C : ProdChart P B N) (g : H0 p N O) (h : ∀ t y hy ha hc, y ∈ B.U (C.b t) → germ (C.a t) y hy ha τ ∣ germ (C.c t) y hy hc g) : ∃ f, C.mul τ f = g

theorem **restr_injective_of_dense** [IsReduced Y] (h : O' ≤ O) (hd : Dense (O' : Set Y)) : Function.Injective (restr h : H0 p B O → H0 p B O')

theorem **isoOver_of_isOpenImmersion** (r : T ⟶ Y) [IsOpenImmersion r] (h : O ≤ r.opensRange) : IsoOver r O

theorem **isoOver_of_isIso** (r : T ⟶ Y) [IsIso (r ∣_ O)] : IsoOver r O

theorem **IsoOver.bijective** {r : T ⟶ Y} (h : IsoOver r O) {V : Y.Opens} (hV : V ≤ O) {W : T.Opens} (e : W ≤ r ⁻¹ᵁ V) (e' : r ⁻¹ᵁ V ≤ W) : Function.Bijective (r.appLE V W e).hom

theorem **mem_pIdx** {r : T ⟶ Y} {i : B.ι} {z : T} : z ∈ (B.pull r).U (pIdx B r i) ↔ r z ∈ B.U i

theorem **IsoOver.bijective_pull** {r : T ⟶ Y} (h : IsoOver r O) (hB : B.IsLB) {W : T.Opens} (e : W ≤ r ⁻¹ᵁ O) (e' : r ⁻¹ᵁ O ≤ W) : Function.Bijective (H0.pull (p

@[simp] theorem **ofSec_toSec** (f : H0 p B ⊤) : ofSec p ⊤ (toSec f) = f

theorem **ofSec_toSec'** (O : Y.Opens) (f : H0 p B ⊤) : ofSec p O (toSec f) = restr le_top f

theorem **ger_toSec** (f : H0 p B ⊤) (i : B.ι) (y : Y) (hi : y ∈ B.U i) : ger y hi ((toSec f).f i) = germ i y (memTop y) hi f

theorem **ger_secToPow** (t : P.Sec) (n : ℕ) (i : P.ι) (y : Y) (hi : y ∈ P.U i) (h : y ∈ (lbZPow P n).U fun _ => i) : ger y h ((secToPow t n).f fun _ => i) = ger y hi (t.f i) ^ n

theorem **nsmul** (a : Λ) (n : ℕ) {y : Y} {j j' : C.J} (h : y ∈ C.U j) (h' : y ∈ C.U j') : C.θ (n • a) y j j' = C.θ a y j j' ^ n

theorem **ideal_ext_stalk** {I I' : Y.IdealSheafData} (h : ∀ y, stalkId I y = stalkId I' y) : I = I'

theorem **stalkId_baseIdeal** (W : Submodule k (H0 p B ⊤)) (i : B.ι) (y : Y) (hi : y ∈ B.U i) : stalkId (baseIdeal hB W) y = Ideal.span (Set.range fun s : W => germ i y (memTop y) hi s.1)

theorem **mem_support_baseIdeal** (W : Submodule k (H0 p B ⊤)) (y : Y) : y ∈ (baseIdeal hB W).support ↔ ∀ s ∈ W, y ∉ nonzeroLocus (toSec s)

theorem **baseIdeal_span** {ι : Type} (v : ι → B.Sec) : baseIdeal hB (Submodule.span k (Set.range fun i => ofSec p ⊤ (v i))) = cutIdeal hB v

theorem **stalkId_comap_baseIdeal** (W : Submodule k (H0 p B ⊤)) (r : T ⟶ Y) (j : (B.pull r).ι) (z : T) (hj : z ∈ (B.pull r).U j) : stalkId ((baseIdeal hB W).comap r) z = Ideal.span (Set.range fun s : W => germ j z (memTop z) hj (H0.pull r le_top s.1))

theorem **stalkId_eq** (i : Δ.P.ι) (y : Y) (hi : y ∈ Δ.P.U i) : stalkId I y = Ideal.span {ger y hi (Δ.t.f i)}

theorem **mem_nonzeroLocus** (y : Y) : y ∈ nonzeroLocus Δ.t ↔ y ∉ I.support

theorem **regular** (p : Y ⟶ Sp k) (O : Y.Opens) : Regular (H0.ofSec p O Δ.t)

theorem **germ_pow** (p : Y ⟶ Sp k) {O : Y.Opens} (n : ℕ) (i : Δ.P.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ Δ.P.U i) (h : y ∈ (lbZPow Δ.P n).U fun _ => i) : germ (fun _ => i) y hy h (H0.ofSec p O (Δ.pow n)) = ger y hi (Δ.t.f i) ^ n

theorem **nowhereZero_pow** (p : Y ⟶ Sp k) {O : Y.Opens} (hO : ∀ y ∈ O, y ∉ I.support) (n : ℕ) : NowhereZero (H0.ofSec p O (Δ.pow n))

theorem **isLB_moving** : Δ.moving.IsLB

theorem **incl_injective** (O : T.Opens) : Function.Injective (Δ.incl O)

theorem **pull_mem_range** (w : W) : H0.pull r le_top w.1 ∈ LinearMap.range (Δ.incl ⊤)

theorem **incl_lift** (w : W) : Δ.incl ⊤ (Δ.lift w) = H0.pull r le_top w.1

theorem **germ_lift** (w : W) (j : Δ.moving.ι) (z : T) (hj : z ∈ Δ.moving.U j) : ger z hj.2 (Δ.t.f j.2) * germ j z (memTop z) hj (Δ.lift w) = germ j.1 z (memTop z) hj.1 (H0.pull r le_top w.1)

theorem **iSup_nonzeroLocus** {ι : Type} {v : ι → W} (hv : Submodule.span k (Set.range v) = ⊤) : ⨆ i, nonzeroLocus (toSec (Δ.lift (v i))) = ⊤

theorem **bijective_pull** {B : LB Y} (hB : B.IsLB) : Function.Bijective (H0.pull (p

theorem **existsUnique_lift** {S : Scheme.{0}} (x : S ⟶ Y) (hx : ⊤ ≤ x ⁻¹ᵁ U) : ∃! z : S ⟶ M.X, z ≫ M.σ = x

Development.AbelianVarietyTheory.Definitions.Divisors.JoinCastSucc

theorem **join_last** {R : Type} {D N : ℕ} (u : Fin D → Fin N → R) (ξ : Fin N → R) (j : Fin N) : join u ξ (Fin.last D, j) = ξ j

theorem **join_castSucc** {R : Type} {D N : ℕ} (u : Fin D → Fin N → R) (ξ : Fin N → R) (i : Fin D) (j : Fin N) : join u ξ (i.castSucc, j) = u i j

theorem **shortLat_std** (M : ℕ) : ShortLat (1 : Matrix (Fin (M + 1)) (Fin (M + 1)) ℚ) (fun _ => 1) 1

theorem **sl2** : ShortLat (1 : Matrix (Fin 2) (Fin 2) ℚ) (fun _ => 1) 1

theorem **shear2** : ShearInv (D

theorem **join_eval** {R : Type} [CommRing R] (v : Fin 1 → Fin 2 → R) (u : Fin 2 → R) : eval (join v u) (det2 R) = v 0 0 * u 1 - v 0 1 * u 0

theorem **last2** : LastNonzero (D

theorem **max2** : IsMaxPt (1 : Matrix (Fin 2) (Fin 2) ℂ) 1 (det2 ℂ) (fun p => if p.1 = p.2 then 1 else 0)

theorem **mh2** : IsMultiHomogeneous (fun _ => 1) (det2 ℚ)

theorem **core_hyp** : ∃ h 0 : ℝ, ShortLat (1 : Matrix (Fin 2) (Fin 2) ℚ) (fun _ => 1) 1 ∧ IsMultiHomogeneous (fun _ => 1) (det2 ℚ) ∧ ShearInv (D

theorem **lin_ne** {K : Type} [Field K] {N : ℕ} {y : Fin N → K} (hy : y ≠ 0) : lin y ≠ 0

theorem **lin_hom** {K : Type} [Field K] {N : ℕ} (y : Fin N → K) : (lin y).IsHomogeneous 1

theorem **nosq** {K : Type} [Field K] {N : ℕ} {z : Fin N → K} (hz : z ≠ 0) : NoSquare (lin z)

theorem **not_nosq** {K : Type} [Field K] {N : ℕ} {z : Fin N → K} (hz : z ≠ 0) : ¬ NoSquare (lin z ^ 2)

theorem **avoid** : AvoidPoly (projTuple 1) 1

Development.AbelianVarietyTheory.Definitions.Divisors.KerDatTuple

theorem **KerDat.isKernel** {ξ χ ξR Q ω φ} (h : KerDat R i ξ χ ξR Q ω φ) : IsKernel R i ξ Q ω φ

theorem **KerDat.vecOf** {ξ χ ξR Q ω φ} (h : KerDat R i ξ χ ξR Q ω φ) : VecOf R i Q (fun y => omegaK R.Pg y ∗ omegaQ Q (Wc D i).rank ξR (-1) y) φ

theorem **kerDat_tuple** (hL : R.toLevCh.Is A.μ0) (hP : R.IsThP) (t : T0) : KerDat R i (R.xiH t) (R.xi t) (R.xiR t) (R.Q t) (R.ωP i t) (R.vP i t)

Development.AbelianVarietyTheory.Definitions.Divisors.KerIdealCompOfSurjective

@[simp] theorem **ptsMap_ofConv** (f : C →ₐc[R] A) (x : WithConv (A →ₐ[R] B)) : (pointsMap B f x).ofConv = x.ofConv.comp (f : C →ₐ[R] A)

@[simp] theorem **ptsPush_ofConv** (g : B →ₐ[R] D) (x : WithConv (A →ₐ[R] B)) : (ptsPush A g x).ofConv = g.comp x.ofConv

theorem **convMul_apply_groupLike** (x y : WithConv (A →ₐ[R] B)) {u : A} (hu : IsGroupLikeElem R u) : (x ∗ y) u = x u ∗ y u

theorem **convPow_apply_groupLike** (x : WithConv (A →ₐ[R] B)) {u : A} (hu : IsGroupLikeElem R u) (n : ℕ) : (x ^ n) u = x u ^ n

@[simp] theorem **toPts_ofConv** [IsCocomm R C] (f : WithConv (C →ₐc[R] A)) : (toPts f).ofConv = (f.ofConv : C →ₐ[R] A)

theorem **toPts_injective** [IsCocomm R C] : Function.Injective (toPts : WithConv (C →ₐc[R] A) → _)

@[simp] theorem **precomp_ofConv** [IsCocomm R C] [IsCocomm R D] (h : D →ₐc[R] C) (f : WithConv (C →ₐc[R] A)) : (precomp A h f).ofConv = f.ofConv.comp h

@[simp] theorem **postcomp_ofConv** [IsCocomm R C] (h : A →ₐc[R] B) (f : WithConv (C →ₐc[R] A)) : (postcomp C h f).ofConv = h.comp f.ofConv

theorem **comp_mulBy_pts** [IsCocomm R A] {E : Type∗} [CommRing E] [Algebra R E] (x : A →ₐ[R] E) (n : ℕ) : x.comp (mulBy R A n : A →ₐ[R] A) = (toConv x ^ n).ofConv

theorem **comp_mulBy** [IsCocomm R C] (f : C →ₐc[R] A) (n : ℕ) : f.comp (mulBy R C n) = (toConv f ^ n).ofConv

theorem **mulBy_comp** [IsCocomm R C] [IsCocomm R A] (f : C →ₐc[R] A) (n : ℕ) : (mulBy R A n).comp f = (toConv f ^ n).ofConv

theorem **map_mulBy** [IsCocomm R C] [IsCocomm R A] (f : C →ₐc[R] A) (n : ℕ) (c : C) : f (mulBy R C n c) = mulBy R A n (f c)

theorem **mulBy_mulBy** [IsCocomm R A] (m n : ℕ) (a : A) : mulBy R A m (mulBy R A n a) = mulBy R A (m ∗ n) a

theorem **mulBy_groupLike** [IsCocomm R A] (n : ℕ) {u : A} (hu : IsGroupLikeElem R u) : mulBy R A n u = u ^ n

theorem **kerIdeal_le_ker_iff** (f : C →ₐc[R] A) {E : Type∗} [CommRing E] [Algebra R E] (x : A →ₐ[R] E) : kerIdeal f ≤ RingHom.ker x ↔ ∀ c, x (f c) = algebraMap R E (counit c)

theorem **torsionIdeal_le_ker_iff** [IsCocomm R A] (n : ℕ) {E : Type∗} [CommRing E] [Algebra R E] (x : A →ₐ[R] E) : torsionIdeal R A n ≤ RingHom.ker x ↔ toConv x ^ n = 1

theorem **kerIdeal_of_surjective** {f : C →ₐc[R] A} (hf : Function.Surjective f) : kerIdeal f = augIdeal R A

theorem **kerIdeal_comp** (g : A →ₐc[R] B) (f : C →ₐc[R] A) : kerIdeal (g.comp f) = Ideal.map (g : A →ₐ[R] B) (kerIdeal f)

theorem **kerIdeal_comp_of_surjective** (g : A →ₐc[R] B) {f : C →ₐc[R] A} (hf : Function.Surjective f) : kerIdeal (g.comp f) = kerIdeal g

theorem **map_torsionIdeal** [IsCocomm R C] [IsCocomm R A] {f : C →ₐc[R] A} (hf : Function.Surjective f) (n : ℕ) : Ideal.map (f : C →ₐ[R] A) (torsionIdeal R C n) = torsionIdeal R A n

theorem **torsionIdeal_le_of_dvd** [IsCocomm R A] {m n : ℕ} (h : m ∣ n) : torsionIdeal R A n ≤ torsionIdeal R A m

@[simp] theorem **liftSurj_apply** (i : A →ₐc[R] B) (hi : Function.Surjective i) (f : A →ₐc[R] C) (h : RingHom.ker i ≤ RingHom.ker f) (a : A) : liftSurj i hi f h (i a) = f a

@[simp] theorem **cotMap_d** (f : C →ₐc[R] A) (c : C) : cotMap f (dUnit R c) = dUnit R (f c)

theorem **cotMap_comp** (g : A →ₐc[R] B) (f : C →ₐc[R] A) : cotMap (g.comp f) = cotMap g ∘ₗ cotMap f

@[simp] theorem **cotMap_id** : cotMap (BialgHom.id R A) = LinearMap.id

theorem **cotMap_one** [IsCocomm R C] : cotMap (1 : WithConv (C →ₐc[R] A)).ofConv = 0

theorem **cotMap_mul** [IsCocomm R C] (f g : WithConv (C →ₐc[R] A)) : cotMap (f ∗ g).ofConv = cotMap f.ofConv + cotMap g.ofConv

theorem **cotMap_pow** [IsCocomm R C] (f : WithConv (C →ₐc[R] A)) (n : ℕ) : cotMap (f ^ n).ofConv = n • cotMap f.ofConv

theorem **dUnit_mul_groupLike** {u v : A} (hu : IsGroupLikeElem R u) (hv : IsGroupLikeElem R v) : dUnit R (u ∗ v) = dUnit R u + dUnit R v

@[simp] theorem **dualMonoidHom_ofConv** (f : WithConv (C →ₐc[R] A)) : (dualMonoidHom R A C f).ofConv = CartierDual.map f.ofConv

theorem **dual_mulBy** (n : ℕ) : CartierDual.map (mulBy R A n) = mulBy R (CartierDual R A) n

theorem **isGroupLikeElem_muT** : IsGroupLikeElem R (muT R m)

theorem **muT_pow** (k : ℕ) : muT R m ^ k = AddMonoidAlgebra.single (k : ZMod m) 1

theorem **muT_pow_self** : muT R m ^ m = 1

theorem **mu_algHom_ext** {x y : AddMonoidAlgebra R (ZMod m) →ₐ[R] B} (h : x (muT R m) = y (muT R m)) : x = y

theorem **muPoint_single** (u : B) (hu : u ^ m = 1) (k : ZMod m) : muPoint R u hu (AddMonoidAlgebra.single k 1) = u ^ k.val

@[simp] theorem **muPoint_muT** (u : B) (hu : u ^ m = 1) : muPoint R u hu (muT R m) = u

@[simp] theorem **muChar_muT** {A : Type∗} [CommRing A] [Bialgebra R A] (u : A) (hg : IsGroupLikeElem R u) (hu : u ^ m = 1) : muChar u hg hu (muT R m) = u

theorem **mu_bialgHom_ext** {A : Type∗} [CommRing A] [Bialgebra R A] {x y : AddMonoidAlgebra R (ZMod m) →ₐc[R] A} (h : x (muT R m) = y (muT R m)) : x = y

theorem **cast_digit** (n : ℕ) (a : ℤ_[p]) : (digit n a : ZMod (p ^ n)) = PadicInt.toZModPow n a

theorem **pow_dvd_sub_digit** (n : ℕ) (a : ℤ_[p]) : (p : ℤ_[p]) ^ n ∣ a - digit n a

theorem **pow_digit_eq** {n : ℕ} {x : M} (hx : x ^ p ^ n = 1) {a : ℤ_[p]} {k : ℕ} (h : (k : ZMod (p ^ n)) = PadicInt.toZModPow n a) : x ^ digit n a = x ^ k

theorem **pow_digit_natCast** {n : ℕ} {x : M} (hx : x ^ p ^ n = 1) (k : ℕ) : x ^ digit n (k : ℤ_[p]) = x ^ k

theorem **pow_digit_one** {n : ℕ} {x : M} (hx : x ^ p ^ n = 1) : x ^ digit n (1 : ℤ_[p]) = x

@[simp] theorem **digit_zero** (n : ℕ) : digit n (0 : ℤ_[p]) = 0

theorem **pow_digit_add** {n : ℕ} {x : M} (hx : x ^ p ^ n = 1) (a b : ℤ_[p]) : x ^ digit n (a + b) = x ^ digit n a ∗ x ^ digit n b

theorem **pow_digit_mul** {n : ℕ} {x : M} (hx : x ^ p ^ n = 1) (a b : ℤ_[p]) : x ^ digit n (a * b) = (x ^ digit n b) ^ digit n a

theorem **pow_digit_succ** {n : ℕ} {x : M} (hx : x ^ p ^ n = 1) (a : ℤ_[p]) : x ^ digit (n + 1) a = x ^ digit n a

Development.AbelianVarietyTheory.Definitions.Divisors.KerInfIntegrableOfMemRspan

theorem **norm_toL_mul** (x y : Matrix (Fin a) (Fin a) (Cinf E)) : ‖toL (x * y)‖ ≤ a * (‖toL x‖ * ‖toL y‖)

theorem **height_mul_le** (x y : GL (Fin a) (Cinf E)) : height (x * y) ≤ (1 + 2 * a) * (height x * height y)

theorem **integrable_of_modGr** {F : GA E a → ℂ} (hF : F ∈ modGr E a) {X : Type} [MeasurableSpace X] {ν : Measure X} {k : X → ℂ} {γ : X → GL (Fin a) (Cinf E)} (hk : IsRapidInf ν k γ) (g : GA E a) : Integrable (fun x => k x * F (g * infGL E a (γ x))) ν

theorem **rspan_le_modGr** (π : CuspSub E a) : rspan π ≤ modGr E a

theorem **kerInf_integrable_of_mem_rspan** {π : CuspSub E a} {F : GA E a → ℂ} (hF : F ∈ rspan π) {X : Type} [MeasurableSpace X] {ν : Measure X} {k : X → ℂ} {γ : X → GL (Fin a) (Cinf E)} (hk : IsRapidInf ν k γ) (g : GA E a) : Integrable (fun x => k x * F (g * infGL E a (γ x))) ν

theorem **kerInf_add** {π : CuspSub E a} {F G : GA E a → ℂ} (hF : F ∈ rspan π) (hG : G ∈ rspan π) {X : Type} [MeasurableSpace X] {ν : Measure X} {k : X → ℂ} {γ : X → GL (Fin a) (Cinf E)} (hk : IsRapidInf ν k γ) : kerInf ν k γ (F + G) = kerInf ν k γ F + kerInf ν k γ G

theorem **rspan_triv** (ha : a ≤ 1) : rspan (CuspSub.triv E ha) = Submodule.span ℂ {fun _ => 1}

theorem **rspan_twist_triv** (ha : a ≤ 1) (χ : Formalism.L2.Host.HChar E) : rspan ((CuspSub.triv E ha).twist χ) = Submodule.span ℂ {fun g => ((chiDet a χ g : ℂˣ) : ℂ)}

theorem **plGL_injective** (w : InfinitePlace E) : Function.Injective (plGL (a

Development.AbelianVarietyTheory.Definitions.Divisors.KerSizes

theorem **KerSizes.mono** {t : KerEntTied ex} {W : Wts t.X t.Pr.Word D.R} {Del h0 : ℕ} {CY CT CT' : ℝ} (h : KerSizes t W Del h0 CY CT) (hC : CT ≤ CT') (hL : 0 ≤ L.sc.L) : KerSizes t W Del h0 CY CT'

Development.AbelianVarietyTheory.Definitions.Divisors.KerToCoinvIffInvFactor

theorem **vanish_locAug_iff** {G H Y : Type} [Group G] [Group H] [TopologicalSpace Y] (om : G × H →* Module.End ℂ (Y → ℂ)) (T : (Y → ℂ) →ₗ[ℂ] ℂ) : (∀ z ∈ locAug om, T z = 0) ↔ ∀ g : G, ∀ φ ∈ SB Y, T (om (g, 1) φ) = T φ

theorem **secSpace_ne_bot** {G H Y : Type} [Group G] [Group H] [TopologicalSpace Y] (om : G × H →* Module.End ℂ (Y → ℂ)) {l : (Y → ℂ) →ₗ[ℂ] ℂ} (hl : ∃ φ ∈ SB Y, l φ ≠ 0) : secSpace om l ≠ ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.KernelNormConstantSpec

theorem **wtOf_cast** {n : ℕ} {E : Finset ℚ} (h : E.card = n) (hE : InCoset (((n : ℚ) - 1) / 2) E) (a : Fin n) : ((wtOf n E).w a : ℚ) = highestWeight E h a

theorem **wtOf_top_zero** (C : ℝ) (τ : D.what₂.Tau) : wtOf (rankP D D.R.m) (entries (Node.par D C (.top 0) τ)) = La D C τ

theorem **DefPair.nodes** {m : ℕ} {p : Pair T} (h : DefPair m p) : DefNode m p.small ∧ DefNode m p.big

theorem **wt_top_zero** (h : (G.V (.top 0)).rank = rankP D D.R.m) (τ : D.what₂.Tau) (a : Fin (G.V (.top 0)).rank) : (wt G (.top 0) τ).w a = (La D sc.cGap τ).w (Fin.cast h a)

theorem **halfInd_congr** {m n : ℕ} {e : AE D} {Λ Λ' : AddSubgroup (Mx D m n)} {x y : Mx D m n} (h : (x ∈ Λ ∧ IsHalf e x) ↔ (y ∈ Λ' ∧ IsHalf e y)) : halfInd e Λ x = halfInd e Λ' y

theorem **FinDatum.vec_eq** {t : T} {p : Pair T} {d : FinDatum G t p} {ΛS ΛT f₀ f} (h : d.IsTheta W ΛS ΛT f₀ f) : d.vec W ΛS ΛT f₀ = W.om p (d.z, 1) f

theorem **fockIn_onb** (p : Pair T) (f : WO (G.pS p) (G.pT p) → ℂ) : fockIn W W.onb p f = W.fock p f

theorem **kkIn_eq** (c : D.what₂.Tau → ℂ) (d : FinDatum G t p) (x : FinPt G p.small) (y : FinPt G p.big) : Z.kkIn c d x y = (∏ τ, opp p (c τ) ^ fockDeg G p τ) • Z.kk d x y

theorem **isTheta_vacuum** (t : T) (p : Pair T) : (FinDatum.vacuum G t p).IsTheta W (N.latS p) (N.latT p) (Z.vac p) (Z.vac p)

theorem **jetTab_vacuum** (t : T) (hp : DefPair D.R.m p) (k : DVar D.what₂.Tau (G.V p.small).rank (G.V p.big).rank →₀ ℕ) : jetTab W N.frC p (Z.vec (FinDatum.vacuum G t p)) ((N.u _)⁻¹ * N.u _) ((N.u _)⁻¹ * N.u _) k = Z.c0 p * opp p (((k.mapDomain (fvar G p)).prod fun v n => period Z.δ v.1 ^ n) * Formalism.MainNotions.GL3Theta.applyAt0 D (monomial (k.mapDomain (fvar G p)) 1) (Z.e0 p))

Development.AbelianVarietyTheory.Definitions.Divisors.KottwitzDimSignatureOneOne

theorem **infUniv_of_rep** {X : Scheme.{0}} {s : X ⟶ Sp (D.OEp p)} (univ : Obj D p N s) (h : ∀ (S : Scheme.{0}) (s' : S ⟶ Sp (D.OEp p)) (o : Obj D p N s'), ∃! f : S ⟶ X, f ≫ s = s' ∧ Nonempty (ObjMap o univ f)) : InfUniv univ

theorem **InfUniv.lift_point** {X : Scheme.{0}} {s : X ⟶ Sp (D.OEp p)} {univ : Obj D p N s} (h : InfUniv univ) (C C₀ : Type) [CommRing C] [IsLocalRing C] [IsArtinianRing C] [CommRing C₀] (π : C →+* C₀) (hπ : Function.Surjective π) (hsq : RingHom.ker π ^ 2 = ⊥) (t : Sp C ⟶ Sp (D.OEp p)) (hl : ∀ o₀ : Obj D p N (spMap π ≫ t), ∃ o : Obj D p N t, Nonempty (ObjMap o₀ o (spMap π))) (x₀ : Sp C₀ ⟶ X) (hx₀ : x₀ ≫ s = spMap π ≫ t) : ∃ x : Sp C ⟶ X, spMap π ≫ x = x₀ ∧ x ≫ s = t

theorem **not_infUniv_const** {k : Type} [Field k] {b : Sp k ⟶ Sp (D.OEp p)} (o : Obj D p N b) (u : Obj D p N (spMap (algebraMap k (DualNumber k)) ≫ b)) (m : ObjMap u o (spMap (algebraMap k (DualNumber k)))) : ¬ InfUniv u

theorem **exists_not_infUniv** : ∃ (D : Datum HostModel.E21) (p : ℕ), Good HostModel.E21 p 5 ∧ ∃ (s : Sp (DualNumber ℂ) ⟶ Sp (D.OEp p)) (u : Obj D p 5 s), ¬ InfUniv u

theorem **cls_eq_iff** (f g : H →ₗ[k] H) : cls Fil f = cls Fil g ↔ ∀ x ∈ Fil, f x - g x ∈ Fil

theorem **KSDim.rank_eq** {d : ℕ} (h : KSDim ρ φ Fil Z d) : Module.rank k ((ksSub ρ φ Fil Z).map (cls Fil)) = d

theorem **KSDim.unique** {d e : ℕ} (h : KSDim ρ φ Fil Z d) (h' : KSDim ρ φ Fil Z e) : d = e

theorem **mem_L₁** (x : k × k) : x ∈ L₁ k ↔ x.2 = 0

theorem **ksDim_plane** : KSDim (Int.castRingHom (Module.End k (k × k))) (ω₂ k) (L₁ k) ⊥ 1

theorem **ksDim_plane_fixed** : KSDim (Int.castRingHom (Module.End k (k × k))) (ω₂ k) (L₁ k) (L₁ k) 0

theorem **ksDim_plane_cm** : KSDim (diag k) (ω₂ k) (L₁ k) ⊥ 0

theorem **not_ksDim_plane_zero** : ¬ KSDim (Int.castRingHom (Module.End k (k × k))) (ω₂ k) (L₁ k) ⊥ 0

theorem **mem_F₄** (x : Fin 4 → k) : x ∈ F₄ k ↔ x 1 = 0 ∧ x 2 = 0

theorem **act₄_apply** (a : k × k) (x : Fin 4 → k) : act₄ k a x = ![a.1 * x 0, a.1 * x 1, a.2 * x 2, a.2 * x 3]

theorem **kottwitzDim_signatureOneOne** : KSDim (act₄ k) (ω₄ k) (F₄ k) ⊥ 1

Development.AbelianVarietyTheory.Definitions.Divisors.LeTwoMulOfForallScale

theorem **form_eq_sum** (H : Matrix n n ℂ) (v w : n → ℂ) : form H v w = ∑ a, ∑ b, v a * H a b * conj (w b)

theorem **form_single** [DecidableEq n] (H : Matrix n n ℂ) (a b : n) : form H (Pi.single a 1) (Pi.single b 1) = H a b

theorem **form_eq_transpose** (H : Matrix n n ℂ) (v w : n → ℂ) : form H v w = star w ⬝ᵥ Hᵀ *ᵥ v

theorem **form_add_left** (H : Matrix n n ℂ) (v v' w : n → ℂ) : form H (v + v') w = form H v w + form H v' w

theorem **form_add_right** (H : Matrix n n ℂ) (v w w' : n → ℂ) : form H v (w + w') = form H v w + form H v w'

theorem **form_smul_left** (H : Matrix n n ℂ) (c : ℂ) (v w : n → ℂ) : form H (c • v) w = c * form H v w

theorem **form_smul_right** (H : Matrix n n ℂ) (c : ℂ) (v w : n → ℂ) : form H v (c • w) = conj c * form H v w

theorem **form_sub_left** (H : Matrix n n ℂ) (v v' w : n → ℂ) : form H (v - v') w = form H v w - form H v' w

theorem **form_sub_right** (H : Matrix n n ℂ) (v w w' : n → ℂ) : form H v (w - w') = form H v w - form H v w'

theorem **form_add** (H K : Matrix n n ℂ) (v w : n → ℂ) : form (H + K) v w = form H v w + form K v w

theorem **form_smul** (c : ℂ) (H : Matrix n n ℂ) (v w : n → ℂ) : form (c • H) v w = c * form H v w

theorem **conj_form** {H : Matrix n n ℂ} (hH : H.IsHermitian) (v w : n → ℂ) : conj (form H v w) = form H w v

theorem **form_mulVec** (K : Matrix n n ℂ) (A : Matrix n m ℂ) (B : Matrix n l ℂ) (u : m → ℂ) (w : l → ℂ) : form K (A *ᵥ u) (B *ᵥ w) = u ⬝ᵥ (Aᵀ * K * Bᴴᵀ) *ᵥ star w

theorem **posDef_iff** {H : Matrix n n ℂ} : H.PosDef ↔ H.IsHermitian ∧ ∀ v, v ≠ 0 → 0 < form H v v

theorem **exists_root** {H : Matrix n n ℂ} (hH : H.PosSemidef) : ∃ R : (n → ℂ) →ₗ[ℂ] EuclideanSpace ℂ n, (∀ v w, form H v w = inner ℂ (R w) (R v)) ∧ ∀ v, gnorm H v = ‖R v‖

theorem **gnorm_nonneg** (H : Matrix n n ℂ) (v : n → ℂ) : 0 ≤ gnorm H v

theorem **form_self** {H : Matrix n n ℂ} (hH : H.PosSemidef) (v : n → ℂ) : form H v v = ((gnorm H v ^ 2 : ℝ) : ℂ)

theorem **gnorm_sq** {H : Matrix n n ℂ} (hH : H.PosSemidef) (v : n → ℂ) : gnorm H v ^ 2 = (form H v v).re

theorem **gnorm_smul** {H : Matrix n n ℂ} (hH : H.PosSemidef) (c : ℂ) (v : n → ℂ) : gnorm H (c • v) = ‖c‖ * gnorm H v

theorem **gnorm_add_le** {H : Matrix n n ℂ} (hH : H.PosSemidef) (v w : n → ℂ) : gnorm H (v + w) ≤ gnorm H v + gnorm H w

theorem **norm_form_le** {H : Matrix n n ℂ} (hH : H.PosSemidef) (v w : n → ℂ) : ‖form H v w‖ ≤ gnorm H v * gnorm H w

theorem **gnorm_parallelogram** {H : Matrix n n ℂ} (hH : H.PosSemidef) (v w : n → ℂ) : gnorm H (v + w) ^ 2 + gnorm H (v - w) ^ 2 = 2 * (gnorm H v ^ 2 + gnorm H w ^ 2)

@[simp] theorem **gnorm_zero** (H : Matrix n n ℂ) : gnorm H 0 = 0

theorem **gnorm_pos** {H : Matrix n n ℂ} (hH : H.PosDef) {v : n → ℂ} (hv : v ≠ 0) : 0 < gnorm H v

theorem **form_adj** [DecidableEq m] {H : Matrix m m ℂ} {K : Matrix n n ℂ} (hH : H.IsHermitian) (hu : IsUnit H.det) (hK : K.IsHermitian) (T : Matrix n m ℂ) (v : m → ℂ) (w : n → ℂ) : form K (T *ᵥ v) w = form H v (adj H K T *ᵥ w)

theorem **exists_bound** (hH : H.PosDef) (hK : K.PosSemidef) (T : Matrix n m ℂ) : ∃ C, 0 ≤ C ∧ ∀ v, gnorm K (T *ᵥ v) ≤ C * gnorm H v

theorem **opNorm_nonneg** (H : Matrix m m ℂ) (K : Matrix n n ℂ) (T : Matrix n m ℂ) : 0 ≤ opNorm H K T

theorem **opNorm_le_of_bound** {T : Matrix n m ℂ} {C : ℝ} (hC : 0 ≤ C) (h : ∀ v, gnorm K (T *ᵥ v) ≤ C * gnorm H v) : opNorm H K T ≤ C

theorem **gnorm_mulVec_le** (hH : H.PosDef) (hK : K.PosSemidef) (T : Matrix n m ℂ) (v : m → ℂ) : gnorm K (T *ᵥ v) ≤ opNorm H K T * gnorm H v

theorem **opNorm_le_iff** (hH : H.PosDef) (hK : K.PosSemidef) {T : Matrix n m ℂ} {C : ℝ} : opNorm H K T ≤ C ↔ 0 ≤ C ∧ ∀ v, gnorm K (T *ᵥ v) ≤ C * gnorm H v

@[simp] theorem **opNorm_zero** (H : Matrix m m ℂ) (K : Matrix n n ℂ) : opNorm H K 0 = 0

theorem **opNorm_add_le** (hH : H.PosDef) (hK : K.PosSemidef) (S T : Matrix n m ℂ) : opNorm H K (S + T) ≤ opNorm H K S + opNorm H K T

theorem **opNorm_smul_le** (hH : H.PosDef) (hK : K.PosSemidef) (c : ℂ) (T : Matrix n m ℂ) : opNorm H K (c • T) ≤ ‖c‖ * opNorm H K T

theorem **opNorm_mul_le** (hH : H.PosDef) (hK : K.PosDef) (hL : L.PosSemidef) (S : Matrix l n ℂ) (T : Matrix n m ℂ) : opNorm H L (S * T) ≤ opNorm K L S * opNorm H K T

theorem **le_two_mul_of_forall_scale** {x c a b : ℝ} (hc : 0 ≤ c) (ha : 0 ≤ a) (hb : 0 ≤ b) (h : ∀ s, 0 < s → x ≤ c * (s * a ^ 2 + b ^ 2 / s)) : x ≤ 2 * c * a * b

theorem **opNorm_le_of_selfAdjoint** (hH : H.PosSemidef) {T : Matrix m m ℂ} (hT : ∀ v w, form H (T *ᵥ v) w = form H v (T *ᵥ w)) {c : ℝ} (hc : 0 ≤ c) (h : ∀ v, ‖form H (T *ᵥ v) v‖ ≤ c * gnorm H v ^ 2) : opNorm H H T ≤ c

theorem **norm_form_mulVec_le** (hH : H.PosDef) (T : Matrix m m ℂ) (v : m → ℂ) : ‖form H (T *ᵥ v) v‖ ≤ opNorm H H T * gnorm H v ^ 2

theorem **isSesq_form** (H : Matrix n n ℂ) : IsSesq (form H)

theorem **IsSesq.polarisation** {F : V → V → M} (hF : IsSesq F) (x y : V) : F x y = (1 / 4 : ℂ) • ((F (x + y) (x + y) + (-1 : ℂ) • F (x - y) (x - y)) + (I • F (x + I • y) (x + I • y) + (-I) • F (x - I • y) (x - I • y)))

theorem **opNorm_le_of_diagonal** {H : Matrix n n ℂ} {G : Matrix m m ℂ} (hH : H.PosDef) (hG : G.PosSemidef) {F : (m → ℂ) → (m → ℂ) → Matrix n n ℂ} (hF : IsSesq F) (hs : ∀ ξ v w, form H (F ξ ξ *ᵥ v) w = form H v (F ξ ξ *ᵥ w)) {C : ℝ} (hC : 0 ≤ C) (h : ∀ ξ v, ‖form H (F ξ ξ *ᵥ v) v‖ ≤ C * gnorm G ξ ^ 2 * gnorm H v ^ 2) (ξ η : m → ℂ) : opNorm H H (F ξ η) ≤ 2 * C * gnorm G ξ * gnorm G η

theorem **blockQS_eq** (T : Matrix (m ⊕ l) (m ⊕ l) ℂ) : blockQS T = T.toBlocks₂₁

theorem **orthProj_mul_incl** (H : Matrix (m ⊕ l) (m ⊕ l) ℂ) : orthProj H * incl m l = 1

theorem **orthProj_mul_orthLift** (H : Matrix (m ⊕ l) (m ⊕ l) ℂ) : orthProj H * orthLift H = 0

theorem **quotMap_mul_incl** : quotMap m l * incl m l = 0

theorem **quotMap_mul_orthLift** (H : Matrix (m ⊕ l) (m ⊕ l) ℂ) : quotMap m l * orthLift H = 1

theorem **incl_mul_orthProj_add** (H : Matrix (m ⊕ l) (m ⊕ l) ℂ) : incl m l * orthProj H + orthLift H * quotMap m l = 1

theorem **form_incl** (H : Matrix (m ⊕ l) (m ⊕ l) ℂ) (u u' : m → ℂ) : form H (incl m l *ᵥ u) (incl m l *ᵥ u') = form (subGram H) u u'

theorem **orthLift_transpose_mul** (hu : IsUnit (subGram H).det) : (orthLift H)ᵀ * H = fromCols 0 (quotGram H)

theorem **form_orthLift_incl** (hu : IsUnit (subGram H).det) (q : l → ℂ) (u : m → ℂ) : form H (orthLift H *ᵥ q) (incl m l *ᵥ u) = 0

theorem **form_orthLift** (hu : IsUnit (subGram H).det) (q q' : l → ℂ) : form H (orthLift H *ᵥ q) (orthLift H *ᵥ q') = form (quotGram H) q q'

theorem **det_eq_det_subGram_mul** (hu : IsUnit (subGram H).det) : H.det = (subGram H).det * (quotGram H).det

theorem **isUnit_det_quotGram** (h : IsUnit H.det) (hu : IsUnit (subGram H).det) : IsUnit (quotGram H).det

theorem **isHermitian_subGram** (h : H.IsHermitian) : (subGram H).IsHermitian

theorem **conjTranspose_toBlocks₁₂** (h : H.IsHermitian) : H.toBlocks₁₂ᴴ = H.toBlocks₂₁

theorem **isHermitian_quotGram** (h : H.IsHermitian) : (quotGram H).IsHermitian

theorem **posDef_subGram** (h : H.PosDef) : (subGram H).PosDef

theorem **posDef_quotGram** (h : H.PosDef) : (quotGram H).PosDef

theorem **det_firstGram_succ** (H : Matrix (Fin N) (Fin N) ℂ) (i : Fin N) (hu : IsUnit (firstGram i.2.le H).det) : (firstGram (k

theorem **posDef_gradedGram** {H : Matrix (Fin N) (Fin N) ℂ} (h : H.PosDef) (i : Fin N) : (gradedGram H i).PosDef

Development.AbelianVarietyTheory.Definitions.Divisors.LeastSplitAuxiliaryPrimeThetaChainSpec

theorem **blockC'_between** (hm : G.NearMid) (hs : 1 ≤ G.s) (hg : G.s < G.ga) {i : ℕ} (h1 : 1 ≤ i) (h2 : i ≤ G.m) {x : ℚ} (hx : x ∈ G.blockC' i) : G.a i < x ∧ x < G.a i + G.ga

theorem **a_add_le** {i j : ℕ} (h : i < j) : G.a j + G.ga ≤ G.a i

theorem **cloudHalf_lt** {x : ℚ} (hx : x ∈ G.cloudHalf) : x < G.a G.m

theorem **card_weightX_zero** (hm : G.NearMid) (hs : 1 ≤ G.s) (hg : G.s < G.ga) : (G.weightX 0).card = G.nTop

theorem **inCoset_weightX_zero** (hs : Odd G.s) : InCoset (1 / 2) (G.weightX 0)

theorem **gapBound_iff** (C : ℝ) (g : ℕ) : GapBound D C g ↔ gap D C ≤ g

theorem **isLeast_gap** (C : ℝ) : IsLeast {g | GapBound D C g} (gap D C)

theorem **num_nearMid** (C : ℝ) (τ : D.what₂.Tau) : (num D C τ).NearMid

theorem **num_ranks** (C : ℝ) (τ : D.what₂.Tau) (i j : ℕ) : (num D C τ).rankD i j = rankV D i j ∧ (num D C τ).rankP i = rankP D i ∧ (num D C τ).rankX j = rankX D j ∧ (num D C τ).bigN = bigN D

theorem **odd_r** : Odd D.R.r

theorem **odd_sLen** : Odd (sLen D)

theorem **even_rankV** (i j : ℕ) : Even (rankV D i j) ↔ Odd (i + j)

theorem **even_rankP** (i : ℕ) : Even (rankP D i)

theorem **even_rankX** (j : ℕ) : Even (rankX D j)

theorem **sLen_lt_gap** (C : ℝ) : sLen D < gap D C

theorem **card_weight** (C : ℝ) (τ : D.what₂.Tau) : ((num D C τ).weightX 0).card = rankP D D.R.m

theorem **La_cast** (C : ℝ) (τ : D.what₂.Tau) (a : Fin (rankP D D.R.m)) : ((La D C τ).w a : ℚ) = highestWeight _ (card_weight D C τ) a

theorem **parV_zero** (C : ℝ) {j : ℕ} (hj : j ≤ D.R.m) (τ : D.what₂.Tau) : parV D C 0 j τ = (Ptw D).Dpar j τ

theorem **rPr_spec** : (Formalism.MainNotions.rPrAv D).Prime ∧ ¬ Formalism.MainNotions.rPrAv D ∣ LevelStarExponents.condE D * (D.host.pde - 1) * (D.host.pde' - 1) * Ideal.absNorm D.host.ccK

theorem **leastSplitAuxiliaryPrimeThetaChain_spec** : (q0 D).Prime ∧ q0 D % (Formalism.MainNotions.rPrAv D * LevelStarExponents.condE D) = 1 ∧ ¬ q0 D ∣ Ideal.absNorm D.host.ccK

theorem **leastSplitAuxiliaryPrime_ne_typePrimes** : q0 D ≠ D.host.pde ∧ q0 D ≠ D.host.pde'

theorem **sizeP_lt** (C b : ℝ) : sizeP D C b < depth0 D C b * Real.log (q0 D)

theorem **S1_mem_OmF** {v : HeightOneSpectrum (𝒪 D.host.F)} : v ∈ S1 D ↔ ((D.host.g 0 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal ∨ ((D.host.g 1 : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal ∨ ((D.host.pde : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal ∨ ((D.host.pde' : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal

Development.AbelianVarietyTheory.Definitions.Divisors.LevelOfOfSameBut

theorem **SameBut.symm** {ρ ρ' : Setup D.ntw} (h : SameBut ρ ρ') : SameBut ρ' ρ

theorem **isConjPl_of_lev** {P P' : Pl D.host.E} (h : Lev.IsConjPl P P') : Formalism.BasicNotions.IsConjPl D.host P P'

theorem **levelOf_congr** {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] {Rl : RealDat3 D A3 Ks} {cG : G0 Rl.Pg D.R.r ≃* G0g D.nd} {ρ ρ' : Setup D.ntw} (hρ : ρ.Good) (hρ' : ρ'.Good) (hp : RealPin Rl cG ρ) (hp' : RealPin Rl cG ρ') (hB : ρ'.B = ρ.B) (hv : ∀ i b (u : D.nexp.U), ∃ x, ((D.nexp.psiU i b u : (FractionRing (D.ntw.O i))ˣ) : FractionRing (D.ntw.O i)) = sideEmb D.ntw i b x) (h : ρ.LevelOf D.nexp) : ρ'.LevelOf D.nexp

theorem **adapted** : Adapted p w ρ

theorem **pin** : RealPin p.R p.cG ρ

theorem **norm** : IsNorm p.R p.v₀ (ρ.B (ex.nch.twI w) j (ex.nch.sd w)) ν

theorem **rat** : RatAt p w j hρ ν

theorem **level** : ρ.LevelOf D.nexp

theorem **levelOf_exportTie** (Z : ExportTie p.nexp p.v₀) : Z.ρ.LevelOf D.nexp

theorem **levelOf_of_sameBut** (hm : RecTies ρ) {ρ ρ' : Setup D.ntw} (hρ : ρ.Good) (hρ' : ρ'.Good) (hp : RealPin p.R p.cG ρ) (hp' : RealPin p.R p.cG ρ') (hs : SameBut ρ ρ') (h : ρ.LevelOf D.nexp) : ρ'.LevelOf D.nexp

theorem **mem_level_iff** (hm : RecTies ρ) {ρ : Setup D.ntw} (h : ρ.LevelOf D.nexp) (i : D.ntw.Tw) (j : Fin D.R.m) (b : Bool) (g : ρ.P.Gp) : ρ.eG i j b ((ρ.R.eG.trans p.cG) g) ∈ (ρ.B i j b).G.P.H ↔ g ∈ p.P.P0

theorem **level_chi** (hm : RecTies ρ) {ρ : Setup D.ntw} (hρ : ρ.Good) (h : ρ.LevelOf D.nexp) (i : D.ntw.Tw) (j : Fin D.R.m) (b : Bool) (u : ρ.P.P0) (hu : ρ.eG i j b ((ρ.R.eG.trans p.cG) u) ∈ (ρ.B i j b).G.P.H) : (((ρ.B i j b).G.P.chi (_, hu) : ℂˣ) : ℂ) = (D.what₂.embLL (ρ.P.ψ0 u : D.what₂.LLˣ))⁻¹ ∧ (((ρ.B i j b).G.P.chi (_, hu) : ℂˣ) : ℂ) ∈ (Formalism.L2.Host.embLL0 D).fieldRange

theorem **placesAt** (hw : D.plEq w ∈ p.R.twPl) {ρ : Setup D.ntw} (hρ : ρ.Good) (hp : RealPin p.R p.cG ρ) : PlacesAt p w j ρ

theorem **AtTw.and** (h : AtTw Q) (h' : AtTw Q') : AtTw fun Fm x hx i p w j => Q Fm x hx i p w j ∧ Q' Fm x hx i p w j

theorem **AtTw.imp** (h : AtTw Q) (f : ∀ {eps : ℝ} (Fm : FamP eps) (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x) (p : Pt Fm x hx i), p.Rows → More p → ∀ w, (Fm.D x i).plEq w ∈ p.R.twPl → ∀ j, Q Fm x hx i p w j → Q' Fm x hx i p w j) : AtTw Q'

theorem **AtTw.of_forall** (f : ∀ {eps : ℝ} (Fm : FamP eps) (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x) (p : Pt Fm x hx i), p.Rows → More p → ∀ w, (Fm.D x i).plEq w ∈ p.R.twPl → ∀ j, Q Fm x hx i p w j) : AtTw Q

theorem **rowAt_of_ratStep** (h : RatHypAt) (hR : RatStep) : RowAt

theorem **twPl** : TwPl

theorem **exists_fit** (h1 : TwAdm) (h2 : RowAt) (h3 : AdaptEx) : AtTw fun Fm x _ i p w j => Nonempty (ExportTie p.nexp p.v₀) ∧ ∀ Z : ExportTie p.nexp p.v₀, ∃ (ρ : Setup (Fm.D x i).ntw) (hρ : ρ.Good) (ν : ℂ), SameBut Z.ρ ρ ∧ Fit p w j hρ ν

theorem **Steps.exportSt** (h : Steps) : ExportSt

Development.AbelianVarietyTheory.Definitions.Divisors.LiftFrameEmbeddingBase

theorem **subQ_red** : SubQ O N lam ϖ (NablaT O N lam / redKer O N lam ϖ)

theorem **embeds_red** [Fintype ι] : Embeds O N lam lam ϖ (NablaT O N lam / redKer O N lam ϖ)

theorem **nontrivial_red** (h : ¬ IsUnit ϖ) : Nontrivial (NablaT O N (fun (_ : Unit) _ => 0) / redKer O N (fun (_ : Unit) _ => 0) ϖ)

theorem **exists_levelAction** {l : ℕ} [Fact l.Prime] {J : Type} [Group J] {ι : Type} [Fintype ι] {n : ℕ} {A : Type} [CommRing A] (S : Subring A) (K : IntermediateField ℚ_[l] (PadicAlgCl l)) (ev : ι → (A →+* PadicAlgCl l)) (h2 : ∀ τ z, ev τ z ∈ K) (h3 : ∀ τ, ∀ z ∈ S, ‖ev τ z‖ ≤ 1) (y : J →* GL (Fin n) S) (χ : J →* ℂˣ) {M : ℕ} (hM : 0 < M) (hχ : ∀ κ, χ κ ^ M = 1) (hK : ∀ z : PadicAlgCl l, z ^ M = 1 → z ∈ K) (io : PadicAlgCl l ≃+* ℂ) (sh : ι → ℤ) : ∃ (kap : J →* (ι → GL (Fin n) (integralClosure ℤ_[l] K))) (chi : J →* (integralClosure ℤ_[l] K)ˣ), (∀ κ τ a b, GlobalTheory.AdeleEv.embInt K ((kap κ τ).val a b) = ev τ ((y κ).val a b)) ∧ ∀ κ, io (GlobalTheory.AdeleEv.embInt K (chi κ) * ∏ τ, GlobalTheory.AdeleEv.embInt K (kap κ τ).val.det ^ sh τ) * ((χ κ : ℂˣ) : ℂ) = 1

theorem **exists_intHom** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} {J : Type} [Group J] (x : J →* GL (Fin n) (TensorProduct F E (Formalism.L2.Herm.Af F))) {β : GL (Fin n) (TensorProduct F E (Formalism.L2.Herm.Af F))} (hβ : β = 1) (h : ∀ κ, x κ ∈ Formalism.L2.DIV.St β) : ∃ y : J →* GL (Fin n) (Formalism.L2.Herm.intAf F E), ∀ κ a b, ((y κ).val a b : TensorProduct F E (Formalism.L2.Herm.Af F)) = (x κ).val a b

theorem **blk_op_zero** (j : Fin D.R.m) (t : p.P.T) (P : (blk p j t).Pl) : (blk p j t).op (P, 0) = 1

theorem **isHalf_lab** {l : ℕ} [Fact l.Prime] (e0 : D.what₂.LL →+* PadicAlgCl l) : IsHalf fun τ => e0.comp (Links.labLL D τ)

theorem **levJ_eq** (j : Fin D.R.m) (t : p.P.T) : levJ p j t = (levEnv (Vc D j) (p.R.βV j) (p.R.datVt j t)).J

theorem **levChi_apply** (j : Fin D.R.m) (t : p.P.T) (κ : levJ p j t) : levChi p j t κ = levFn (p.R.βV j) (Formalism.IndefR.Lev.flag1L (p.R.datVt j t)) (Formalism.UG.bc (Formalism.IndefR.Lev.swapAE D) κ.1.1)

theorem **exists_levEnv_pow** (V : Formalism.BasicNotions.TH.HSp D) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)) (Lv : Formalism.L2.DIV.LevDat D.host.E ℂ V.rank) : ∃ M : ℕ, 0 < M ∧ ∀ κ, (levEnv V β Lv).χ κ ^ M = 1

theorem **exists_levChi_pow** (j : Fin D.R.m) (t : p.P.T) : ∃ M : ℕ, 0 < M ∧ ∀ κ, levChi p j t κ ^ M = 1

theorem **mem_St_of_levEnv** {V : Formalism.BasicNotions.TH.HSp D} {β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)} {Lv : Formalism.L2.DIV.LevDat D.host.E ℂ V.rank} {g : Bridge.Gf V} (h : g ∈ (levEnv V β Lv).J) : Formalism.UG.bc (Formalism.IndefR.Lev.swapAE D) g.1 ∈ St β

theorem **exists_intHom_of** {V : Formalism.BasicNotions.TH.HSp D} {β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)} (hβ : β = 1) (J : Subgroup (Bridge.Gf V)) (h : ∀ g ∈ J, Formalism.UG.bc (Formalism.IndefR.Lev.swapAE D) g.1 ∈ St β) : ∃ y : J →* GL (Fin V.rank) (Formalism.L2.Herm.intAf D.host.F D.host.E), ∀ κ a b, ((y κ).val a b : Formalism.IndefR.Lev.EAf D) = Formalism.IndefR.Lev.swapAE D (κ.1.1.val a b)

theorem **exists_levInt** (j : Fin D.R.m) (t : p.P.T) : ∃ y : levJ p j t →* GL (Fin (Vc D j).rank) (Formalism.L2.Herm.intAf D.host.F D.host.E), ∀ κ a b, ((y κ).val a b : Formalism.IndefR.Lev.EAf D) = Formalism.IndefR.Lev.swapAE D (matOf j κ.1 a b)

theorem **exists_tied** (j : Fin D.R.m) (t : p.P.T) {l : ℕ} [Fact l.Prime] {N : ℕ} (hN : N = (Vc D j).rank) (e0 : D.what₂.LL →+* PadicAlgCl l) (inc : D.what₂.Tau → (D.host.E →+* PadicAlgCl l)) (half : IsHalf inc) (lab : ∀ τ, inc τ = e0.comp (Links.labLL D τ) ∨ inc τ = GlobalTheory.ExpFrame.cj (e0.comp (Links.labLL D τ))) (sh : D.what₂.Tau → ℤ) : ∃ (F : Frame l D.what₂.Tau N) (ix : F.I → (blk p j t).Pl × ℕ) (T : Tied p j t F e0 ix inc), T.sh = sh ∧ (∀ τ, T.bs τ = 1) ∧ (∀ x : 𝒪 D.what₂.LL, e0 x ∈ Set.range F.emb) ∧ Function.Injective F.emb ∧ ∃ K : IntermediateField ℚ_[l] (PadicAlgCl l), FiniteDimensional ℚ_[l] K ∧ Set.range F.emb = {x | x ∈ K ∧ ‖x‖ ≤ 1}

theorem **nonempty_tied** (j : Fin D.R.m) (t : p.P.T) {l : ℕ} [Fact l.Prime] (e0 : D.what₂.LL →+* PadicAlgCl l) : ∃ (F : Frame l D.what₂.Tau (Vc D j).rank) (ix : F.I → (blk p j t).Pl × ℕ), Nonempty (Tied p j t F e0 ix fun τ => e0.comp (Links.labLL D τ))

Development.AbelianVarietyTheory.Definitions.Divisors.LinEndOfOLinear

theorem **endData_tate** (m : ℤ) : EndData tatePre (𝟙 _) (𝟙 _) (AddMonoidHom.mulLeft m) (AddMonoidHom.mulLeft m)

theorem **not_endData_tate** : ¬ EndData tatePre (𝟙 _) (𝟙 _) 0 (AddMonoidHom.id _)

theorem **endData_of_oAct** {R : Type} [CommRing R] [IsDomain R] (O : Type) [Ring O] [StarRing O] (D : DegPre R) [Module O D.X] (io : D.OAct O) (a : O) : EndData D (io.onB.act a) (io.onB.act (star a)) (DistribSMul.toAddMonoidHom D.X a) (DistribSMul.toAddMonoidHom D.X (star a))

theorem **torusEndOf_id** {S : Scheme.{0}} (X : Type) [AddCommGroup X] : TorusEndOf X (AddMonoidHom.id X) (𝟙 (splitTorus S X))

theorem **linEnd_id** (U : UnifPre I D A G) : LinEnd U (𝟙 _) (AddMonoidHom.id _) (𝟙 G)

theorem **thickMap_map** {G H : Over (Sp R)} (b : G ⟶ H) (e : ℕ) : thickMap R I e G ≫ ((Over.pullback (thick R I (e + 1))).map b).left = ((Over.pullback (thick R I e)).map b).left ≫ thickMap R I e H

theorem **linEnd_of_oLinear** (U : UnifPre I D A G) (O : Type) [Ring O] [StarRing O] [Module O D.toPre.X] (ιD : D.OAct O) (ιA : A.OAct O) (ιG : RingAct G O) (hl : U.OLinear O ιD ιA ιG) (a : O) : LinEnd U (ιD.onB.act a) (DistribSMul.toAddMonoidHom D.toPre.X (star a)) (ιG.act a)

theorem **extLam_lam** (L : LB G.left) (hL : L.IsLB) : ExtLam G (L.pull (pullback.fst G.hom (jK R))) (abelian₂ G L)

theorem **exists_model** : ∃ (A : PolAV (Sp (FractionRing (PowerSeries ℚ)))) (G : Over (Sp (PowerSeries ℚ))) (_ : GrpObj G) (_ : IsCommMonObj G) (D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (U : Unif (maximalIdeal (PowerSeries ℚ)) D A G), LinEnd U.toUnifPre (𝟙 _) (AddMonoidHom.id _) (𝟙 G) ∧ Module.finrank ℤ D.toPre.X = 1 ∧ ExtLam G (U.LG.L.pull (pullback.fst G.hom (jK _))) (abelian₂ G U.LG.L)

Development.AbelianVarietyTheory.Definitions.Divisors.LineBundleGlobalSectionsDimCongrPicardClass

theorem **strHom_eq** {f : X ⟶ Sp k} (hf : ∀ r, algebraMap k Γ(X, ⊤) r = f.appTop ((Scheme.ΓSpecIso (.of k)).inv r)) : strHom k X = f

theorem **algebraMap_eq_strHom** (r : k) : algebraMap k Γ(X, ⊤) r = (strHom k X).appTop ((Scheme.ΓSpecIso (.of k)).inv r)

theorem **comp_strHom** [Algebra k Γ(Y, ⊤)] {g : Y ⟶ X} (hg : ∀ r : k, g.appTop (algebraMap k Γ(X, ⊤) r) = algebraMap k Γ(Y, ⊤) r) : g ≫ strHom k X = strHom k Y

theorem **range_redι** (Z : Closeds X) : Set.range (redι Z) = Z

theorem **isIntegral_red** {Z : Closeds X} (h : IsIrreducible (Z : Set X)) : IsIntegral (Scheme.IdealSheafData.vanishingIdeal Z).subscheme

theorem **topologicalKrullDim_red** (Z : Closeds X) : topologicalKrullDim (Scheme.IdealSheafData.vanishingIdeal Z).subscheme = topologicalKrullDim (Z : Set X)

theorem **isCompleteCurve_iff** {f : X ⟶ Sp k} (hf : ∀ r, algebraMap k Γ(X, ⊤) r = f.appTop ((Scheme.ΓSpecIso (.of k)).inv r)) [IsProper f] {Z : Closeds X} : IsCompleteCurve k Z ↔ IsIrreducible (Z : Set X) ∧ topologicalKrullDim (Z : Set X) = 1

theorem **isNef_iff** {f : X ⟶ Sp k} (hf : ∀ r, algebraMap k Γ(X, ⊤) r = f.appTop ((Scheme.ΓSpecIso (.of k)).inv r)) [IsProper f] {L : LB X} : IsNef k L ↔ L.IsLB ∧ ∀ Z : Closeds X, IsIrreducible (Z : Set X) → topologicalKrullDim (Z : Set X) = 1 → 0 ≤ interNumRed k Z ![L]

theorem **isCompleteCurve_range** (i : Y ⟶ X) [IsClosedImmersion i] [IsIntegral Y] (hd : topologicalKrullDim Y = 1) [IsProper (i ≫ strHom k X)] : IsCompleteCurve k (Set.range i, i.isClosedEmbedding.isClosed_range)

theorem **isNef_iff_isClosedImmersion** {L : LB X} : IsNef k L ↔ L.IsLB ∧ ∀ (C : Scheme.{0}) (i : C ⟶ X), IsClosedImmersion i → IsIntegral C → topologicalKrullDim C = 1 → IsProper (i ≫ strHom k X) → 0 ≤ interNumOn k i ![L]

theorem **exists_sec_of_cls_eq_add** (hN : N.IsLB) (hA : A.IsLB) (hB : B.IsLB) (e : cls N hN = cls A hA + cls B hB) (s : A.Sec) (t : B.Sec) : ∃ u : N.Sec, nonzeroLocus u = nonzeroLocus s ⊓ nonzeroLocus t

theorem **sec_ne_zero_of_mem** {s : L.Sec} {x : X} (h : x ∈ nonzeroLocus s) : s ≠ 0

theorem **sec_ne_zero_iff** [IsIntegral X] (s : L.Sec) : s ≠ 0 ↔ genericPoint X ∈ nonzeroLocus s

theorem **hasNonzeroSec_iff** [IsIntegral X] : HasNonzeroSec L ↔ ∃ s : L.Sec, genericPoint X ∈ nonzeroLocus s

theorem **hasNonzeroSec_iff_nontrivial** : HasNonzeroSec L ↔ Nontrivial (coh (twist (structureModule X) L) 0)

theorem **HasNonzeroSec.of_cls_eq** (h : HasNonzeroSec M) (hL : L.IsLB) (hM : M.IsLB) (e : cls L hL = cls M hM) : HasNonzeroSec L

theorem **HasNonzeroSec.of_cls_eq_add** [IsIntegral X] (h₁ : HasNonzeroSec A) (h₂ : HasNonzeroSec B) (hN : N.IsLB) (hA : A.IsLB) (hB : B.IsLB) (e : cls N hN = cls A hA + cls B hB) : HasNonzeroSec N

theorem **HasNonzeroSec.tens** [IsIntegral X] (h₁ : HasNonzeroSec A) (h₂ : HasNonzeroSec B) (hA : A.IsLB) (hB : B.IsLB) : HasNonzeroSec (tens A B)

theorem **IsGenerated.hasNonzeroSec** [Nonempty X] (h : IsGenerated L) : HasNonzeroSec L

theorem **HasNonzeroSec.pow** [IsIntegral X] (h : HasNonzeroSec L) (hL : L.IsLB) (n : ℕ) : HasNonzeroSec (lbPow L n)

theorem **EffCartierData.hasNonzeroSec** [Nonempty X] (D : EffCartierData X) : HasNonzeroSec D.lb

theorem **interNum_one_nonneg_of_sec** (hH : IsAmpleLB H) (hd : topologicalKrullDim C = 1) (hL : L.IsLB) (s : L.Sec) (hs : genericPoint C ∈ nonzeroLocus s) : 0 ≤ interNum k (structureModule C) ![L] ∧ (nonzeroLocus s ≠ ⊤ → 0 < interNum k (structureModule C) ![L])

theorem **interNum_one_of_cls_eq_add** (hH : IsAmpleLB H) (hd : topologicalKrullDim C = 1) {A B : LB C} (hL : L.IsLB) (hA : A.IsLB) (hB : B.IsLB) (e : cls L hL = cls A hA + cls B hB) : interNum k (structureModule C) ![L] = interNum k (structureModule C) ![A] + interNum k (structureModule C) ![B]

theorem **isLB_vecOne** (hL : L.IsLB) : ∀ i, ((![L] : Fin 1 → LB X) i).IsLB

theorem **interNumRed_one_eq** (hL : L.IsLB) (Z : Closeds X) : letI

theorem **IsCompleteCurve.isProper_strHom** (hZ : IsCompleteCurve k Z) : letI

theorem **interNumRed_one_congr** (hL : L.IsLB) (hM : M.IsLB) (e : cls L hL = cls M hM) (Z : Closeds X) : interNumRed k Z ![L] = interNumRed k Z ![M]

theorem **IsCompleteCurve.interNumRed_of_cls_eq_add** (hZ : IsCompleteCurve k Z) (hH : IsAmpleLB H) (hN : N.IsLB) (hA : A.IsLB) (hB : B.IsLB) (e : cls N hN = cls A hA + cls B hB) : interNumRed k Z ![N] = interNumRed k Z ![A] + interNumRed k Z ![B]

theorem **interNumRed_one_eq_zero** (hL : L.IsLB) (e : cls L hL = 0) (Z : Closeds X) : interNumRed k Z ![L] = 0

theorem **IsCompleteCurve.interNumRed_lbPow** (hZ : IsCompleteCurve k Z) (hH : IsAmpleLB H) (hL : L.IsLB) (n : ℕ) : interNumRed k Z ![lbPow L n] = n * interNumRed k Z ![L]

theorem **IsCompleteCurve.interNumRed_dual** (hZ : IsCompleteCurve k Z) (hH : IsAmpleLB H) (hL : L.IsLB) : interNumRed k Z ![dual L] = -interNumRed k Z ![L]

theorem **IsCompleteCurve.interNumRed_nonneg_of_sec** (hZ : IsCompleteCurve k Z) (hH : IsAmpleLB H) (hL : L.IsLB) (s : L.Sec) {x : X} (hx : x ∈ Z) (hs : x ∈ nonzeroLocus s) : 0 ≤ interNumRed k Z ![L] ∧ ((∃ y ∈ Z, y ∉ nonzeroLocus s) → 0 < interNumRed k Z ![L])

theorem **IsCompleteCurve.exists_open** (hZ : IsCompleteCurve k Z) : ∃ x ∈ Z, ∃ y ∈ Z, ∃ U : X.Opens, x ∈ U ∧ y ∉ U

theorem **IsAmpleLB.interNumRed_pos** (hA : IsAmpleLB A) (hZ : IsCompleteCurve k Z) : 0 < interNumRed k Z ![A]

theorem **IsNef.of_cls_eq** (h : IsNef k M) (hL : L.IsLB) (e : cls L hL = cls M h.isLB) : IsNef k L

theorem **IsNef.of_lbIso** (h : IsNef k M) (hL : L.IsLB) (e : LBIso L M) : IsNef k L

theorem **isNef_triv** : IsNef k (LB.triv X)

theorem **IsGenerated.isNef** (h : IsGenerated L) (hH : IsAmpleLB H) : IsNef k L

theorem **IsAmpleLB.isNef** (h : IsAmpleLB A) : IsNef k A

theorem **IsNef.of_cls_eq_add** (hA : IsNef k A) (hB : IsNef k B) (hH : IsAmpleLB H) (hN : N.IsLB) (e : cls N hN = cls A hA.isLB + cls B hB.isLB) : IsNef k N

theorem **IsNef.tens** (hA : IsNef k A) (hB : IsNef k B) (hH : IsAmpleLB H) : IsNef k (tens A B)

theorem **IsNef.pow** (h : IsNef k L) (hH : IsAmpleLB H) (n : ℕ) : IsNef k (lbPow L n)

theorem **isNef_pow_iff** (hH : IsAmpleLB H) (hL : L.IsLB) {n : ℕ} (hn : 1 ≤ n) : IsNef k (lbPow L n) ↔ IsNef k L

theorem **IsNef.of_cls_eq_sum** {ι : Type*} (s : Finset ι) {P : ι → LB X} (hP : ∀ i, IsNef k (P i)) (n : ι → ℕ) (hH : IsAmpleLB H) (hN : N.IsLB) (e : cls N hN = ∑ i ∈ s, n i • cls (P i) (hP i).isLB) : IsNef k N

theorem **IsGenerated.isNef_of_isProjOver** (h : IsGenerated L) {f : X ⟶ Sp k} (hf : IsProjOver f) : IsNef k L

theorem **IsNef.tens_of_isProjOver** (hA : IsNef k A) (hB : IsNef k B) {f : X ⟶ Sp k} (hf : IsProjOver f) : IsNef k (CechPic.tens A B)

theorem **IsNef.pull_of_isClosedImmersion** [Algebra k Γ(Y, ⊤)] (h : IsNef k L) (g : Y ⟶ X) [IsClosedImmersion g] (hg : ∀ r : k, g.appTop (algebraMap k Γ(X, ⊤) r) = algebraMap k Γ(Y, ⊤) r) : IsNef k (L.pull g)

theorem **IsAmpleLB.not_isNef_dual** (h : IsAmpleLB A) {Z : Closeds X} (hZ : IsCompleteCurve k Z) : ¬ IsNef k (dual A)

theorem **cls_lbPow_tens_lbPow** (hN : N.IsLB) (hA : A.IsLB) (m r : ℕ) (h : (lbPow (tens (lbPow N m) A) r).IsLB) : cls _ h = r • (m • cls N hN + cls A hA)

theorem **IsAmpleLB.exists_hasNonzeroSec** [IsLocallyNoetherian X] [Nonempty X] (hA : IsAmpleLB A) (hM : M.IsLB) : ∃ q : ℕ, 1 ≤ q ∧ ∀ (P : LB X) (hP : P.IsLB), cls P hP = cls M hM + q • cls A hA.isLB → HasNonzeroSec P

theorem **IsPsEff.exists_hasNonzeroSec** [IsLocallyNoetherian X] (h : IsPsEff N) (hA : IsAmpleLB A) {m : ℕ} (hm : 1 ≤ m) : ∃ r, 1 ≤ r ∧ HasNonzeroSec (lbPow (tens (lbPow N m) A) r)

theorem **IsPsEff.of_cls_eq** (h : IsPsEff M) (hL : L.IsLB) (e : cls L hL = cls M h.isLB) : IsPsEff L

theorem **IsPsEff.of_lbIso** (h : IsPsEff M) (hL : L.IsLB) (e : LBIso L M) : IsPsEff L

theorem **isPsEff_of_hasNonzeroSec_pow** (hH : IsAmpleLB H) (hL : L.IsLB) {n : ℕ} (hn : 1 ≤ n) (h : HasNonzeroSec (lbPow L n)) : IsPsEff L

theorem **HasNonzeroSec.isPsEff** (h : HasNonzeroSec L) (hL : L.IsLB) (hH : IsAmpleLB H) : IsPsEff L

theorem **EffCartierData.isPsEff** (D : EffCartierData X) (hH : IsAmpleLB H) : IsPsEff D.lb

theorem **IsGenerated.isPsEff** (h : IsGenerated L) (hH : IsAmpleLB H) : IsPsEff L

theorem **IsAmpleLB.isPsEff** (h : IsAmpleLB A) : IsPsEff A

theorem **IsPsEff.of_cls_eq_add** (hL : IsPsEff L) (hM : IsPsEff M) (hN : N.IsLB) (e : cls N hN = cls L hL.isLB + cls M hM.isLB) : IsPsEff N

theorem **IsPsEff.tens** (hL : IsPsEff L) (hM : IsPsEff M) : IsPsEff (tens L M)

theorem **IsPsEff.exists_of_dual** [IsLocallyNoetherian X] (hA : IsAmpleLB A) (h : IsPsEff (dual A)) : ∃ r : ℕ, 1 ≤ r ∧ HasNonzeroSec (lbZPow A (-r))

theorem **lineBundleGlobalSectionsDim_def** (L : LB X) : lbH0 k L = hdim k (twist (structureModule X) L) 0

theorem **lineBundleGlobalSectionsDim_eq_finrank** (L : LB X) [Module k L.Sec] [IsScalarTower k Γ(X, ⊤) L.Sec] : lbH0 k L = Module.finrank k L.Sec

theorem **lineBundleGlobalSectionsDim_congr_picardClass** (hL : L.IsLB) (hM : M.IsLB) (e : cls L hL = cls M hM) : lbH0 k L = lbH0 k M

theorem **finite_coh_zero_twist_congr** (hL : L.IsLB) (hM : M.IsLB) (e : cls L hL = cls M hM) (h : Module.Finite k (coh (twist (structureModule X) M) 0)) : Module.Finite k (coh (twist (structureModule X) L) 0)

theorem **finite_coh_zero_twist_of_isProjOver** {f : X ⟶ Sp k} (hf : ∀ r, algebraMap k Γ(X, ⊤) r = f.appTop ((Scheme.ΓSpecIso (.of k)).inv r)) (h : IsProjOver f) (hL : L.IsLB) : Module.Finite k (coh (twist (structureModule X) L) 0)

theorem **lineBundleHasNonzeroSection_of_globalSectionsDim_pos** (h : 0 < lbH0 k L) : HasNonzeroSec L

theorem **lineBundleGlobalSectionsDim_pos_iff** (h : Module.Finite k (coh (twist (structureModule X) L) 0)) : 0 < lbH0 k L ↔ HasNonzeroSec L

theorem **lineBundleGlobalSectionsDim_le_of_picardClass_add** [IsIntegral X] [IsLocallyNoetherian X] (hN : N.IsLB) (hA : A.IsLB) (hB : B.IsLB) (e : cls N hN = cls A hA + cls B hB) (h : HasNonzeroSec B) (hfin : Module.Finite k (coh (twist (structureModule X) N) 0)) : lbH0 k A ≤ lbH0 k N

theorem **isBig_iff** {f : X ⟶ Sp k} (hf : ∀ r, algebraMap k Γ(X, ⊤) r = f.appTop ((Scheme.ΓSpecIso (.of k)).inv r)) (h : IsProjOver f) {D : ℕ} (hD : topologicalKrullDim X = D) : IsBig k L ↔ L.IsLB ∧ ∃ c : ℝ, 0 < c ∧ ∃ᶠ m : ℕ in atTop, c * (m : ℝ) ^ D ≤ lbH0 k (lbPow L m)

theorem **IsBig.of_cls_eq** (h : IsBig k M) (hL : L.IsLB) (e : cls L hL = cls M h.isLB) : IsBig k L

theorem **IsBig.of_lbIso** (h : IsBig k M) (hL : L.IsLB) (e : LBIso L M) : IsBig k L

theorem **hasNonzeroSection_of_le_sectionBound** {c : ℝ} (hc : 0 < c) {m D : ℕ} (hm : 1 ≤ m) (h : c * (m : ℝ) ^ D ≤ lbH0 k L) : HasNonzeroSec L

theorem **IsBig.frequently_hasNonzeroSec** (h : IsBig k L) : ∃ᶠ m : ℕ in atTop, HasNonzeroSec (lbPow L m)

theorem **IsBig.isPsEff** [IsIntegral X] (h : IsBig k L) (hH : IsAmpleLB H) : IsPsEff L

theorem **IsBig.of_pow** {n : ℕ} (h : IsBig k (lbPow L n)) (hL : L.IsLB) (hn : 1 ≤ n) (hfin : ∀ m : ℕ, Module.Finite k (coh (twist (structureModule X) (lbPow L m)) 0)) : IsBig k L

theorem **IsBig.of_cls_eq_add** (hA : IsBig k A) (hB : B.IsLB) (h : HasNonzeroSec B) (hN : N.IsLB) (e : cls N hN = cls A hA.isLB + cls B hB) : IsBig k N

theorem **IsBig.tens** (hA : IsBig k A) (hB : B.IsLB) (h : HasNonzeroSec B) : IsBig k (tens A B)

theorem **IsBig.pow** (h : IsBig k L) {n : ℕ} (hn : 1 ≤ n) : IsBig k (lbPow L n)

theorem **isBig_pow_iff** (hL : L.IsLB) {n : ℕ} (hn : 1 ≤ n) : IsBig k (lbPow L n) ↔ IsBig k L

theorem **topologicalKrullDim_eq** : topologicalKrullDim (PSpace k N) = N

theorem **algebraMap_top_eq** (r : k) : algebraMap k Γ(PSpace k N, ⊤) r = (PSpace.str k N).appTop ((Scheme.ΓSpecIso (.of k)).inv r)

theorem **isProjOver_str** : IsProjOver (PSpace.str k N)

theorem **sectionDim_twistingSheaf_pow** (m : ℕ) : lbH0 k (lbPow (O1 k N) m) = (m + N).choose N

theorem **isNef_twistingSheaf** : IsNef k (O1 k N)

theorem **isBig_twistingSheaf** : IsBig k (O1 k N)

theorem **isPsEff_twistingSheaf** : IsPsEff (O1 k N)

theorem **triv_nef_not_big** : IsNef k (LB.triv (PSpace k (N + 1))) ∧ IsPsEff (LB.triv (PSpace k (N + 1))) ∧ ¬ IsBig k (LB.triv (PSpace k (N + 1)))

theorem **not_isPsEff_dual_twistingSheaf** : ¬ IsPsEff (dual (O1 k (N + 1))) ∧ ¬ IsBig k (dual (O1 k (N + 1)))

theorem **isCompleteCurve_top** : IsCompleteCurve k (⊤ : Closeds (PSpace k 1))

theorem **not_isNef_dual_twistingSheaf** : ¬ IsNef k (dual (O1 k 1))

Development.AbelianVarietyTheory.Definitions.Divisors.LineOppositeParabolicUnitaryLeviSectionEq

@[simp] theorem **twist_apply** (χ : G →* ℂˣ) (ρ : Representation ℂ G V) (g : G) (v : V) : twist χ ρ g v = (χ g : ℂ) • ρ g v

@[simp] theorem **charRep_apply** (χ : G →* ℂˣ) (g : G) (z : ℂ) : charRep χ g z = (χ g : ℂ) * z

@[simp] theorem **extProd_tmul** (π : Representation ℂ G V) (π' : Representation ℂ H W) (g : G × H) (v : V) (w : W) : extProd π π' g (v ⊗ₜ w) = π g.1 v ⊗ₜ π' g.2 w

theorem **mem_fixed** {ρ : Representation ℂ G V} {K : Subgroup G} {v : V} : v ∈ fixed ρ K ↔ ∀ k ∈ K, ρ k v = v

theorem **IsZonal.one_and_biinv** {ρ : Representation ℂ G V} {K : Subgroup G} {ω : G → ℂ} (h : IsZonal ρ K ω) : ω 1 = 1 ∧ ∀ k₁ ∈ K, ∀ k₂ ∈ K, ∀ g, ω (k₁ * g * k₂) = ω g

theorem **IsZonal.isSpherical** {ρ : Representation ℂ G V} {K : Subgroup G} {ω : G → ℂ} (h : IsZonal ρ K ω) : IsSpherical ρ K

theorem **twKer_one_eq** (ρ : Representation ℂ G V) (N : Subgroup G) : twKer ρ N 1 = Representation.Coinvariants.ker (ρ.comp N.subtype)

theorem **twKer_eq** (ρ : Representation ℂ G V) (N : Subgroup G) (ψ : N →* ℂˣ) : twKer ρ N ψ = Representation.Coinvariants.ker (twist ψ⁻¹ (ρ.comp N.subtype))

theorem **charRep_irreducible** (χ : G →* ℂˣ) : (charRep χ).IsIrreducible

theorem **charRep_unitarisable** (χ : G →* ℂˣ) (h : ∀ g, ‖(χ g : ℂ)‖ = 1) : IsUnitarisable (charRep χ)

theorem **isSmooth_iff** (ρ : Representation ℂ G V) : IsSmooth ρ ↔ SplitWeil.IsSmoothRep ρ

theorem **smoothDual_apply** (ρ : Representation ℂ G V) (g : G) (l : smoothPart ρ.dual) (v : V) : (smoothDual ρ g l).1 v = l.1 (ρ g⁻¹ v)

theorem **smoothRep_isSmooth** (ρ : Representation ℂ G V) : IsSmooth (smoothRep ρ)

@[simp] theorem **conjAut_coe** (U : Subgroup G) (g : Subgroup.normalizer (U : Set G)) (u : U) : (conjAut U g u : G) = g * u * (g : G)⁻¹

theorem **modulus_measure** (U : Subgroup G) (hU : IsClosed (U : Set G)) [MeasurableSpace U] [BorelSpace U] (μ : Measure U) [μ.IsHaarMeasure] [μ.Regular] (g : Subgroup.normalizer (U : Set G)) (X : Set U) : μ (conjAut U g '' X) = modulus U hU g * μ X

theorem **mem_U** {g : G} : g ∈ P.U ↔ ∃ h : g ∈ P.Q, P.p ⟨g, h⟩ = 1

theorem **U_le_Q** : P.U ≤ P.Q

theorem **s_injective** : Function.Injective P.s

theorem **Q_le_normalizer** : P.Q ≤ Subgroup.normalizer (P.U : Set G)

theorem **decomp** (q : P.Q) : (P.s (P.p q))⁻¹ * q ∈ P.U

theorem **closed_U** [T1Space M] : IsClosed (P.U : Set G)

theorem **delta_apply** (q : P.Q) : P.delta q = modulus P.U P.closed_U ⟨P.s (P.p q), P.Q_le_normalizer (P.s_mem _)⟩

theorem **delta_of_mem_U** (q : P.Q) (h : (q : G) ∈ P.U) : P.delta q = 1

theorem **deltaHalf_coe** (q : P.Q) : (P.deltaHalf q : ℂ) = (Real.sqrt (P.delta q) : ℝ)

theorem **nInd_apply** (σ : Representation ℂ M W) (g x : G) (f : P.IndSpace σ) : (P.nInd σ g f).1.1 x = f.1.1 (x * g)

theorem **IndSpace.law** {σ : Representation ℂ M W} (f : P.IndSpace σ) (q : P.Q) (g : G) : f.1.1 (q * g) = (P.deltaHalf q : ℂ) • σ (P.p q) (f.1.1 g)

theorem **nInd_isSmooth** (σ : Representation ℂ M W) : IsSmooth (P.nInd σ)

theorem **rJac_mk** (π : Representation ℂ G V) (m : M) (v : V) : P.rJac π m (Submodule.Quotient.mk v) = ((P.deltaHalf (P.sQ m) : ℂ))⁻¹ • Submodule.Quotient.mk (π (P.s m) v)

theorem **delta_of_U_eq_bot** (h : P.U = ⊥) (q : P.Q) : P.delta q = 1

theorem **blockL_le** (b : n → α) : blockL k b ≤ blockQ k b

theorem **blockPart_mul** {b : n → α} {A B : Matrix n n k} (hA : A.BlockTriangular b) (hB : B.BlockTriangular b) : blockPart b (A * B) = blockPart b A * blockPart b B

theorem **blockProj_val** (b : n → α) (q : blockQ k b) : (blockProj b q).val = blockPart b (q : GL n k).val

theorem **blockProj_mem** (b : n → α) (q : blockQ k b) : blockProj b q ∈ blockL k b

theorem **blockProj_of_mem** (b : n → α) (q : blockQ k b) (h : (q : GL n k) ∈ blockL k b) : blockProj b q = q

theorem **diagHom_val** (t : n → kˣ) : (diagHom k n t).val = diagonal fun i => (t i : k)

theorem **diagEntries_val** {b : n → α} (hb : Function.Injective b) (q : blockQ k b) (i : n) : (diagEntries hb q i : k) = (q : GL n k).val i i

theorem **diagHom_diagEntries** {b : n → α} (hb : Function.Injective b) (q : blockQ k b) : diagHom k n (diagEntries hb q) = blockProj b q

theorem **continuous_toHomUnits** {G A : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [Monoid A] [TopologicalSpace A] (f : G →* A) (hf : Continuous f) : Continuous f.toHomUnits

theorem **continuous_entry** (i j : n) : Continuous fun g : GL n k => g.val i j

theorem **isClosed_blockQ** (b : n → α) : IsClosed (blockQ k b : Set (GL n k))

theorem **isClosed_blockL** (b : n → α) : IsClosed (blockL k b : Set (GL n k))

theorem **continuous_blockProj** (b : n → α) : Continuous (blockProj (k

theorem **blockQ_inf_dual** (b : n → α) : blockQ k b ⊓ blockQ k (α

theorem **stdPar_isOpposite** (b : n → α) : (stdPar k b).IsOpposite (stdParOp k b)

theorem **blockProj_mem_blockL** {β : Type} [LinearOrder β] (b : n → α) (c : n → β) (q : blockQ k c) (h : (q : GL n k) ∈ blockL k b) : blockProj c q ∈ blockL k b

theorem **mem_blockQ_two** {g : GL (a ⊕ d) k} : g ∈ blockQ k (twoBlocks a d) ↔ g.val.toBlocks₂₁ = 0

theorem **toBlocks_mul** (x y : blockQ k (twoBlocks a d)) : ((x * y : blockQ k (twoBlocks a d)) : GL (a ⊕ d) k).val.toBlocks₁₁ = x.1.val.toBlocks₁₁ * y.1.val.toBlocks₁₁ ∧ ((x * y : blockQ k (twoBlocks a d)) : GL (a ⊕ d) k).val.toBlocks₂₂ = x.1.val.toBlocks₂₂ * y.1.val.toBlocks₂₂

theorem **blockPar_s_val** (m : GL a k × GL d k) : ((blockPar k a d).s m).val = fromBlocks m.1.val 0 0 m.2.val

theorem **blockPar_p_val** (q : (blockPar k a d).Q) : ((blockPar k a d).p q).1.val = q.1.val.toBlocks₁₁ ∧ ((blockPar k a d).p q).2.val = q.1.val.toBlocks₂₂

theorem **unrChar_apply** {k : Type} [Field k] (v : k →*₀ WithZero (Multiplicative ℤ)) (z : ℂˣ) (x : kˣ) : unrChar v z x = z ^ (-log (v x))

theorem **unrTorus_apply** {k n : Type} [Field k] [Fintype n] (v : k →*₀ WithZero (Multiplicative ℤ)) (α : n → ℂˣ) (t : n → kˣ) : unrTorus v α t = ∏ i, unrChar v (α i) (t i)

theorem **vZ_apply** (x : k) : vZ k x = IsNonarchimedeanLocalField.valueGroupWithZeroIsoInt k (valuation k x)

theorem **unrChar_isUnram** (z : ℂˣ) : SplitWeil.IsUnram (unrChar (vZ k) z)

theorem **unrChar_uniformiser** (z : ℂˣ) (ϖ : kˣ) (h1 : valuation k (ϖ : k) < 1) (hmax : ∀ x : k, valuation k x < 1 → valuation k x ≤ valuation k (ϖ : k)) : unrChar (vZ k) z ϖ = z

theorem **antiJ_apply** (n : ℕ) (i j : Fin n) : antiJ L n i j = if j = i.rev then 1 else 0

theorem **antiJ_mul_apply** (n : ℕ) (A : Matrix (Fin n) (Fin n) L) (i j : Fin n) : (antiJ L n * A) i j = A i.rev j

theorem **mul_antiJ_apply** (n : ℕ) (A : Matrix (Fin n) (Fin n) L) (i j : Fin n) : (A * antiJ L n) i j = A i j.rev

theorem **mem_GU_iff** (σ : L →+* L) (n : ℕ) (g : GL (Fin n) L) : g ∈ GU σ n ↔ ∀ i j, (g⁻¹).val i j = σ (g.val j.rev i.rev)

theorem **selfDualBlocks_id** (n : ℕ) : SelfDualBlocks (id : Fin n → Fin n)

theorem **SelfDualBlocks.dual** {n : ℕ} {b : Fin n → α} (hb : SelfDualBlocks b) : SelfDualBlocks (α

theorem **blockProj_mem_GU** (σ : L →+* L) {n : ℕ} {b : Fin n → α} (hb : SelfDualBlocks b) (q : blockQ L b) (hq : (q : GL (Fin n) L) ∈ GU σ n) : blockProj b q ∈ GU σ n

theorem **parU_isOpposite** (σ : L →+* L) {n : ℕ} {b : Fin n → α} (hb : SelfDualBlocks b) : (parU σ hb).IsOpposite (parUOp σ hb)

theorem **selfDualBlocks_line** (n : ℕ) : SelfDualBlocks (lineBlocks n)

theorem **mem_borelU_Q** (σ : L →+* L) (n : ℕ) (g : GU σ n) : g ∈ (borelU σ n).Q ↔ IsBorelU g

theorem **mem_borelU_U** (σ : L →+* L) (n : ℕ) (g : GU σ n) : g ∈ (borelU σ n).U ↔ IsUnipU g

theorem **locallyCompact_GU** [LocallyCompactSpace L] (σ : L →+* L) (hσ : Continuous σ) (n : ℕ) : LocallyCompactSpace (GU σ n)

theorem **ordL_eq** (x : L) : ordL x = -WithZero.log (Valued.v x)

theorem **unrU_apply** {n : ℕ} (α : Fin (n / 2) → ℂˣ) (t : Fin n → Lˣ) : unrU L α t = ∏ i, α i ^ ordL (t (Fin.castLE (Nat.div_le_self n 2) i) : L)

theorem **card_bcParam** (n : ℕ) (α : Fin (n / 2) → ℂˣ) : (bcParam n α).card = n

theorem **bcParam_of_weylConj** {n : ℕ} {α β : Fin (n / 2) → ℂˣ} (h : WeylConjU α β) : bcParam n α = bcParam n β

theorem **isSubquot_char_of_top** {G M : Type} [Group G] [TopologicalSpace G] [Group M] [TopologicalSpace M] [IsTopologicalGroup G] [LocallyCompactSpace G] [T1Space M] (P : LeviDatum G M) (hU : P.U = ⊥) (χ : M →* ℂˣ) (ω : G →* ℂˣ) (hQ : ∀ g, g ∈ P.Q) (hω : ∀ q : P.Q, ω q = χ (P.p q)) (hs : IsSmooth (charRep ω)) : IsSubquot (charRep ω) (P.nInd (charRep χ))

theorem **isSmooth_charRep** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] (ω : G →* ℂˣ) (h : {g | ω g = 1} ∈ nhds (1 : G)) : IsSmooth (charRep ω)

theorem **isSmooth_unram_det** {n : Type} [Fintype n] [DecidableEq n] {μ : kˣ →* ℂˣ} (hμ : SplitWeil.IsUnram μ) : IsSmooth (charRep (μ.comp (Matrix.GeneralLinearGroup.det (n

theorem **hasSatake_GL_one** (z : ℂˣ) : HasSatakeGL (charRep ((unrChar (vZ k) z).comp (Matrix.GeneralLinearGroup.det (n

theorem **borelGL_two_nontrivial** : (borelGL k (Fin 2)).U ≠ ⊥ ∧ (borelGL k (Fin 2)).Q ≠ ⊤

theorem **locallyCompact_GU_of_unram** [CompleteSpace L] {ϖ : L} {σ : L →+* L} {q : ℕ} (h : IsUnramInv ϖ σ q) (n : ℕ) : LocallyCompactSpace (GU σ n)

theorem **borelU_two_nontrivial** {ϖ : L} {σ : L →+* L} {q : ℕ} (h : IsUnramInv ϖ σ q) : (borelU σ 2).U ≠ ⊥

theorem **coinvFst_mk** (ρ : Representation ℂ (G × H) V) (h : H) (v : V) : coinvFst ρ h (Submodule.Quotient.mk v) = Submodule.Quotient.mk (ρ (1, h) v)

@[simp] theorem **tensorFst_tmul** (ω : Representation ℂ (G × H) V) (τ : Representation ℂ G W) (g : G × H) (v : V) (w : W) : tensorFst ω τ g (v ⊗ₜ w) = ω g v ⊗ₜ τ g.1 w

theorem **coinvTensor_mk** (ω : Representation ℂ (G × H) V) (τ : Representation ℂ G W) (h : H) (v : V) (w : W) : coinvTensor ω τ h (Submodule.Quotient.mk (v ⊗ₜ w)) = Submodule.Quotient.mk (ω (1, h) v ⊗ₜ w)

theorem **coinvChar_mk** (χ : A →* ℂˣ) (ρ : Representation ℂ (A × H) V) (h : H) (v : V) : coinvChar χ ρ h (Submodule.Quotient.mk v) = Submodule.Quotient.mk (ρ (1, h) v)

theorem **coinvChar_rel** (χ : A →* ℂˣ) (ρ : Representation ℂ (A × H) V) (a : A) (v : V) : (Submodule.Quotient.mk (ρ (a, 1) v) : TwJacquet ρ (MonoidHom.inl A H).range (charFst H χ)) = (χ a : ℂ) • Submodule.Quotient.mk v

theorem **twKer_charFst_le_ker** (χ : A →* ℂˣ) (ρ : Representation ℂ (A × H) V) (f : V →ₗ[ℂ] W) : twKer ρ (MonoidHom.inl A H).range (charFst H χ) ≤ LinearMap.ker f ↔ ∀ a v, f (ρ (a, 1) v) = (χ a : ℂ) • f v

theorem **coinvCharMid_mk** (χ : A →* ℂˣ) (ρ : Representation ℂ (G × (A × H)) V) (g : G × H) (v : V) : coinvCharMid χ ρ g (Submodule.Quotient.mk v) = Submodule.Quotient.mk (ρ (g.1, 1, g.2) v)

theorem **twKer_charFst_extProd** (χ χ' : A →* ℂˣ) (π : Representation ℂ H W) : (χ = χ' → twKer (extProd (charRep χ') π) (MonoidHom.inl A H).range (charFst H χ) = ⊥) ∧ (χ ≠ χ' → twKer (extProd (charRep χ') π) (MonoidHom.inl A H).range (charFst H χ) = ⊤)

theorem **IsMatrixCoeff.checkFun** {ρ : Representation ℂ G V} (hp : IsSmooth ρ) {c : G → ℂ} (h : IsMatrixCoeff ρ c) : IsMatrixCoeff (smoothDual ρ) (checkFun c)

theorem **mem_prod_U** (H : Type) [Group H] [TopologicalSpace H] (g : G × H) : g ∈ (P.prod H).U ↔ g.1 ∈ P.U ∧ g.2 = 1

theorem **mem_prodLeft_U** (H : Type) [Group H] [TopologicalSpace H] (g : H × G) : g ∈ (P.prodLeft H).U ↔ g.1 = 1 ∧ g.2 ∈ P.U

theorem **reparam_U** {M' : Type} [Group M'] [TopologicalSpace M'] (e : M' ≃ₜ* M) : (P.reparam e).U = P.U

theorem **reparam_L** {M' : Type} [Group M'] [TopologicalSpace M'] (e : M' ≃ₜ* M) : (P.reparam e).L = P.L

theorem **IsOpposite.reparam** (h : P.IsOpposite P') {M' : Type} [Group M'] [TopologicalSpace M'] (e : M' ≃ₜ* M) : (P.reparam e).IsOpposite (P'.reparam e)

theorem **IsOpposite.prod** (h : P.IsOpposite P') (H : Type) [Group H] [TopologicalSpace H] : (P.prod H).IsOpposite (P'.prod H)

theorem **twist_eq** (χ : G →* ℂˣ) (ρ : Representation ℂ G V) : twist χ ρ = SmGL.twist ρ χ

theorem **charRep_eq** (χ : G →* ℂˣ) : charRep χ = SmGL.ofChar χ

theorem **extProd_eq** (π : Representation ℂ G V) (π' : Representation ℂ H W) : extProd π π' = SmGL.extTensor π π'

theorem **isSubquot_iff** (π : Representation ℂ G W) (ρ : Representation ℂ G V) : IsSubquot π ρ ↔ SmGL.IsSubquotient π ρ

theorem **hasFiniteLength_iff** (ρ : Representation ℂ G V) : HasFiniteLength ρ ↔ SmGL.HasFiniteLength ρ

theorem **isUnitarisable_iff** (ρ : Representation ℂ G V) : IsUnitarisable ρ ↔ SmGL.IsUnitarisable ρ

theorem **isSpherical_iff** (ρ : Representation ℂ G V) (K : Subgroup G) : IsSpherical ρ K ↔ SmGL.IsSphericalFor ρ K

theorem **isZonal_iff** (ρ : Representation ℂ G V) (K : Subgroup G) (ω : G → ℂ) : IsZonal ρ K ω ↔ SmGL.IsZonal ρ K ω

theorem **smoothDual_eq** [TopologicalSpace G] [IsTopologicalGroup G] (ρ : Representation ℂ G V) : smoothDual ρ = SmGL.contra ρ

theorem **unrChar_eq** (z : ℂˣ) : unrChar (vZ k) z = SmGL.unrChar k z

theorem **parDatum_Q** (e : (Σ a, m a) ≃ n) : (parDatum k e).Q = blockQ k (SmGL.blk e)

theorem **parDatum_U** (e : (Σ a, m a) ≃ n) : (parDatum k e).U = SmGL.unipRad k e

@[simp] theorem **midIdx_val** (i : Fin n) : (midIdx i).val = i.val + 1

theorem **lineCases** {p : Fin (n + 2) → Prop} (h0 : p 0) (hm : ∀ i, p (midIdx i)) (hl : p (Fin.last (n + 1))) (i) : p i

theorem **sum_line** {M : Type} [AddCommMonoid M] (f : Fin (n + 2) → M) : ∑ i, f i = f 0 + ∑ i, f (midIdx i) + f (Fin.last (n + 1))

@[simp] theorem **rev_midIdx** (i : Fin n) : (midIdx i).rev = midIdx i.rev

@[simp] theorem **midIdx_ne_zero** (i : Fin n) : midIdx i ≠ 0

@[simp] theorem **midIdx_ne_last** (i : Fin n) : midIdx i ≠ Fin.last (n + 1)

@[simp] theorem **zero_ne_midIdx** (i : Fin n) : 0 ≠ midIdx i

@[simp] theorem **last_ne_midIdx** (i : Fin n) : Fin.last (n + 1) ≠ midIdx i

@[simp] theorem **zero_ne_lastIdx** : (0 : Fin (n + 2)) ≠ Fin.last (n + 1)

@[simp] theorem **midIdx_inj** {i j : Fin n} : midIdx i = midIdx j ↔ i = j

@[simp] theorem **lineBlocks_zero** : lineBlocks (n + 2) 0 = -1

@[simp] theorem **lineBlocks_mid** (i : Fin n) : lineBlocks (n + 2) (midIdx i) = 0

@[simp] theorem **lineBlocks_last** : lineBlocks (n + 2) (Fin.last (n + 1)) = 1

@[simp] theorem **lineMat_zero_zero** : lineMat a h d 0 0 = a

@[simp] theorem **lineMat_mid_mid** (i j : Fin n) : lineMat a h d (midIdx i) (midIdx j) = h i j

@[simp] theorem **lineMat_last_last** : lineMat a h d (Fin.last (n + 1)) (Fin.last (n + 1)) = d

@[simp] theorem **lineMat_zero_mid** (j : Fin n) : lineMat a h d 0 (midIdx j) = 0

@[simp] theorem **lineMat_zero_last** : lineMat a h d 0 (Fin.last (n + 1)) = 0

@[simp] theorem **lineMat_mid_zero** (i : Fin n) : lineMat a h d (midIdx i) 0 = 0

@[simp] theorem **lineMat_last_zero** : lineMat a h d (Fin.last (n + 1)) 0 = 0

@[simp] theorem **lineMat_mid_last** (i : Fin n) : lineMat a h d (midIdx i) (Fin.last (n + 1)) = 0

@[simp] theorem **lineMat_last_mid** (j : Fin n) : lineMat a h d (Fin.last (n + 1)) (midIdx j) = 0

theorem **lineMat_mul** (a' d' : L) (h' : Matrix (Fin n) (Fin n) L) : lineMat a h d ∗ lineMat a' h' d' = lineMat (a ∗ a') (h ∗ h') (d ∗ d')

theorem **lineMat_one** : lineMat (1 : L) (1 : Matrix (Fin n) (Fin n) L) 1 = 1

theorem **lineEmb_val** (x : Lˣ × GL (Fin n) L × Lˣ) : (lineEmb L n x).val = lineMat (x.1 : L) x.2.1.val (x.2.2 : L)

theorem **lineEmb_mem** (x : Lˣ × GL (Fin n) L × Lˣ) : lineEmb L n x ∈ blockL L (lineBlocks (n + 2))

theorem **lineEmb_mem_GU** [Valued L (WithZero (Multiplicative ℤ))] (hσ : ∀ x, σ (σ x) = x) (u : Lˣ) (h : GU σ n) : lineEmb L n (u, h.1, (Units.map σ.toMonoidHom u)⁻¹) ∈ GU σ (n + 2)

theorem **lineSec_val** (σ : L →+∗ L) (hσ : ∀ x, σ (σ x) = x) (x : Lˣ × GU σ n) : (lineSec σ hσ n x).1.val = lineMat (x.1 : L) x.2.1.val (σ x.1)⁻¹

@[simp] theorem **lineParabolicUnitary_leviSection_eq** : (lineParU σ hσ hc n).s = lineSec σ hσ n

@[simp] theorem **lineOppositeParabolicUnitary_leviSection_eq** : (lineParUOp σ hσ hc n).s = lineSec σ hσ n

theorem **lineParU_Q** : (lineParU σ hσ hc n).Q = (parU σ (selfDualBlocks_line (n + 2))).Q

theorem **lineParUOp_Q** : (lineParUOp σ hσ hc n).Q = (parUOp σ (selfDualBlocks_line (n + 2))).Q

theorem **lineParU_isOpposite** : (lineParU σ hσ hc n).IsOpposite (lineParUOp σ hσ hc n)

theorem **lineParU_p** (q : (lineParU σ hσ hc n).Q) : (((lineParU σ hσ hc n).p q).1 : L) = q.1.1.val 0 0 ∧ ∀ i j, ((lineParU σ hσ hc n).p q).2.1.val i j = q.1.1.val (midIdx i) (midIdx j)

theorem **lineParUOp_p** (q : (lineParUOp σ hσ hc n).Q) : (((lineParUOp σ hσ hc n).p q).1 : L) = q.1.1.val 0 0 ∧ ∀ i j, ((lineParUOp σ hσ hc n).p q).2.1.val i j = q.1.1.val (midIdx i) (midIdx j)

theorem **row_eq_zero_of_col** (g : GU σ n) (c : Fin n) (h : ∀ i, i ≠ c → g.1.val i c = 0) (j : Fin n) (hj : j ≠ c.rev) : g.1.val c.rev j = 0

theorem **mem_lineParU_Q** (g : GU σ (n + 2)) : g ∈ (lineParU σ hσ hc n).Q ↔ ∀ i, i ≠ 0 → g.1.val i 0 = 0

theorem **mem_lineParUOp_Q** (g : GU σ (n + 2)) : g ∈ (lineParUOp σ hσ hc n).Q ↔ ∀ i, i ≠ Fin.last (n + 1) → g.1.val i (Fin.last (n + 1)) = 0

theorem **mem_lineParU_Q_iff_smul** (g : GU σ (n + 2)) : g ∈ (lineParU σ hσ hc n).Q ↔ ∃ a : L, g.1.val ∗ᵥ Pi.single 0 1 = a • Pi.single 0 1

theorem **mem_lineParU_U** (g : GU σ (n + 2)) : g ∈ (lineParU σ hσ hc n).U ↔ (∀ i, i ≠ 0 → g.1.val i 0 = 0) ∧ g.1.val 0 0 = 1 ∧ ∀ i j, g.1.val (midIdx i) (midIdx j) = (1 : Matrix (Fin n) (Fin n) L) i j

theorem **borelU_le_lineParU** : (borelU σ (n + 2)).Q ≤ (lineParU σ hσ hc n).Q

theorem **mem_borelLineLevi_Q** (x : Lˣ × GU σ n) : x ∈ (borelLineLevi σ n).Q ↔ lineSec σ hσ n x ∈ (borelU σ (n + 2)).Q

theorem **unrL_apply** (z : ℂˣ) (x : Lˣ) : unrL L z x = z ^ ordL (x : L)

theorem **bcParam_consParam** (z : ℂˣ) (α : Fin (n / 2) → ℂˣ) : bcParam (n + 2) (consParam z α) = bcParam n α + {z, z⁻¹}

theorem **twParam_consParam** (w z : ℂˣ) (α : Fin (n / 2) → ℂˣ) : twParam w (n + 2) (consParam z α) = twParam w n α + {w⁻¹ ∗ z, w⁻¹ ∗ z⁻¹}

theorem **stringParam_of_nonpos** (q : ℂˣ) {t : ℤ} (h : t ≤ 0) : stringParam q t = 0

theorem **card_stringParam** (q : ℂˣ) (t : ℤ) : (stringParam q t).card = t.toNat

theorem **stringParam_add_two** (q : ℂˣ) {t : ℤ} (h : 0 ≤ t) : stringParam q (t + 2) = stringParam q t + {q ^ (-(t + 1)), q ^ (t + 1)}

theorem **continuous_of_unramInv** (h : IsUnramInv ϖ σ q) : Continuous σ

theorem **locallyCompact_lineLevi** (hσ : ∀ x, σ (σ x) = x) (hc : Continuous σ) [LocallyCompactSpace (GU σ (n + 2))] : LocallyCompactSpace (Lˣ × GU σ n)

theorem **lineParU_nontrivial** (h : IsUnramInv ϖ σ q) (n : ℕ) : (lineParU σ h.1 (continuous_of_unramInv h) n).U ≠ ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.LineSCShortExact

theorem **wInt_one** [MeasurableSpace M] (α : A.Fld) : wInt A (fun _ => 1) α = A.integral α

theorem **lineSC_shortExact** {E : X.Modules} (fl : Vert.Flag E 2) : (lineSC fl).ShortExact

theorem **mem_frameAt** {E : X.Modules} (fl : Vert.Flag E 2) {σ : k →+* ℂ} (y : CxPts p σ) : y ∈ ptsIn p σ (frameAt fl y).U

Development.AbelianVarietyTheory.Definitions.Divisors.LinearIndependentBasisCoords

theorem **ext_form** {h h' : HermFam K V} (e : ∀ σ x y, h.form σ x y = h'.form σ x y) : h = h'

theorem **csq_map** (a : ι → K) (v : ι → V) : csq h σ (fun i => σ (a i)) v = (h.form σ (∑ i, a i • v i) (∑ i, a i • v i)).re

theorem **cnrm_map** (a : ι → K) (v : ι → V) : cnrm h σ (fun i => σ (a i)) v = h.nrm σ (∑ i, a i • v i)

theorem **cnrm_one** (x : V) : cnrm h σ (fun _ : Unit => 1) (fun _ => x) = h.nrm σ x

theorem **csq_comb** {κ : Type*} [Fintype κ] (c : ι → ℂ) (a : ι → κ → K) (b : κ → V) : csq h σ c (fun i => ∑ k, a i k • b k) = csq h σ (fun k => ∑ i, c i * σ (a i k)) b

theorem **csq_nonneg** (c : ι → ℂ) (v : ι → V) : 0 ≤ csq h σ c v

theorem **cnrm_sq** (c : ι → ℂ) (v : ι → V) : cnrm h σ c v ^ 2 = csq h σ c v

theorem **nrm_pos** {x : V} (hx : x ≠ 0) : 0 < h.nrm σ x

@[simp] theorem **nrm_comap** [AddCommGroup V'] [Module K V'] (h : HermFam K V) (f : V' →ₗ[K] V) (hf : Function.Injective f) (σ : K →+* ℂ) (x : V') : (comap h f hf).nrm σ x = h.nrm σ (f x)

theorem **isConjInv_comap** [AddCommGroup V'] [Module K V'] {h : HermFam K V} (hh : h.IsConjInv) (f : V' →ₗ[K] V) (hf : Function.Injective f) : (comap h f hf).IsConjInv

@[simp] theorem **nrm_scale** (h : HermFam K V) (c : (K →+* ℂ) → ℝ) (hc : ∀ σ, 0 < c σ) (σ : K →+* ℂ) (x : V) : (scale h c hc).nrm σ x = c σ * h.nrm σ x

theorem **linearIndependent_basisCoords** {n ι : Type*} [Finite n] (b : Module.Basis n K V) (σ : K →+* ℂ) {v : ι → V} (hv : LinearIndependent K v) : LinearIndependent ℂ fun i => basisCoords b σ (v i)

theorem **form_ofGram** {n : Type*} [Fintype n] (b : Module.Basis n K V) (G : (K →+* ℂ) → Matrix n n ℂ) (hG : ∀ σ, (G σ).PosDef) (σ : K →+* ℂ) (i j : n) : (ofGram b G hG).form σ (b i) (b j) = G σ i j

theorem **isConjInv_ofGram** {n : Type*} [Fintype n] (b : Module.Basis n K V) {G : (K →+* ℂ) → Matrix n n ℂ} (hG : ∀ σ, (G σ).PosDef) (hc : ∀ σ i j, G (ComplexEmbedding.conjugate σ) i j = conj (G σ i j)) : (ofGram b G hG).IsConjInv

theorem **nrm_ofBasis** {n : Type*} [Fintype n] [DecidableEq n] (b : Module.Basis n K V) (σ : K →+* ℂ) (i : n) : (ofBasis b).nrm σ (b i) = 1

theorem **isConjInv_ofBasis** {n : Type*} [Fintype n] [DecidableEq n] (b : Module.Basis n K V) : (ofBasis b).IsConjInv

@[simp] theorem **IsExtension.basis_apply** {n : ℕ} (b : Module.Basis (Fin n) K V) (i : Fin n) : hι.basis b i = ι (b i)

theorem **IsExtension.finrank_eq** : Module.finrank E V' = Module.finrank K V

theorem **form_extend** (h : HermFam K V) (τ : E →+* ℂ) (x y : V) : (extend hι h).form τ (ι x) (ι y) = h.form (τ.comp j) x y

theorem **nrm_extend** (h : HermFam K V) (τ : E →+* ℂ) (x : V) : (extend hι h).nrm τ (ι x) = h.nrm (τ.comp j) x

theorem **isConjInv_extend** {h : HermFam K V} (hh : h.IsConjInv) : (extend hι h).IsConjInv

Development.AbelianVarietyTheory.Definitions.Divisors.LinearIndependentPullChart

@[simp] theorem **coe_extMap** (V : Submodule K X) (x : V) : (extMap ι V x : X') = ι x

theorem **isExtension_extMap** (h : ∀ (n : ℕ) (v : Fin n → X), LinearIndependent K v → LinearIndependent E fun i => ι (v i)) (V : Submodule K X) : IsExtension (extMap ι V)

theorem **extLat_span** (S : Set X) : extLat ι (Submodule.span (𝒪 K) S) = Submodule.span (𝒪 E) (ι '' S)

theorem **fg_extLat** {M : Submodule (𝒪 K) X} (h : M.FG) : (extLat ι M).FG

theorem **IsExtension.span_extLat** (hι : IsExtension ι) {M : Submodule (𝒪 K) X} (h : Submodule.span K (M : Set X) = ⊤) : Submodule.span E (extLat ι M : Set X') = ⊤

theorem **IsFull.finiteDimensional** (h : IsFull M) : FiniteDimensional K X

theorem **IsFull.latRank_eq** [NumberField K] (h : IsFull M) : ReciprocityTheory.GeometryOfNumbersTheory.latRank K M = Module.finrank K X

theorem **IsFull.extLat** (h : IsFull M) (hι : IsExtension ι) : IsFull (extLat ι M)

theorem **isFull_span** {n : Type*} [Finite n] (b : Module.Basis n K X) : IsFull (Submodule.span (𝒪 K) (Set.range b))

theorem **restr_injective** [IsIntegral Y] (h : O' ≤ O) (hne : (O' : Set Y).Nonempty) : Function.Injective (restr h : H0 p B O → H0 p B O')

theorem **finite_top** (hp : IsProjOver p) (hB : B.IsLB) : Module.Finite k (H0 p B ⊤)

theorem **linearIndependent_pullChart** (h : W ≤ r ⁻¹ᵁ O) (h' : r ⁻¹ᵁ O ≤ W) {ι : Type*} [Fintype ι] {f : ι → H0 p B O} (hf : LinearIndependent k f) : LinearIndependent E fun i => pullChart a H.w h (f i)

theorem **isExtension_pullChart_top** (hp : IsProjOver p) (hq : IsProjOver q) : IsExtension (pullChart (O

theorem **restr_pullChart** (a : ChartPull B' B r) (hq : r ≫ p = q ≫ spMap φ) (h : W ≤ r ⁻¹ᵁ O) (h' : W' ≤ r ⁻¹ᵁ O') (e : O' ≤ O) (e' : W' ≤ W) (f : H0 p B O) : restr e' (pullChart a hq h f) = pullChart a hq h' (restr e f)

theorem **mem_polar** {hP : P.IsLB} {hB : B.IsLB} {t : P.Sec} {ht : O ≤ nonzeroLocus t} {f : H0 p B O} : f ∈ polar p B hP hB t ht ↔ ∃ g : H0 p (tens P B) ⊤, restr le_top g = (tensChart hP hB).mul (H0.ofSec p O t) f

theorem **range_restr_le_polar** (hP : P.IsLB) (hB : B.IsLB) (t : P.Sec) (ht : 0 ≤ nonzeroLocus t) : LinearMap.range (restr (le_top : 0 ≤ ⊤)) ≤ polar p B hP hB t ht

theorem **cval_eq_val** (f : H0 p B 0) (i : B.ι) (y : CxPts p σ) (h0 : y ∈ ptsIn p σ 0) (hi : y ∈ ptsIn p σ (B.U i)) : cval σ f i y = val y.1 i h0 hi f

theorem **cval_ofSec** (s : B.Sec) : cval σ (H0.ofSec p ⊤ s) = secVal p σ s

theorem **cval_mem** (f : H0 p B ⊤) : cval σ f ∈ fibSec p σ B

theorem **cval_mem_holSec** {D : ℕ} (R : AnRead p σ D) (f : H0 p B ⊤) : cval σ f ∈ R.holSec B

theorem **assocVal_mem_holSec** {D : ℕ} {R : AnRead p σ D} {f : (tens (tens P B) N).ι → CxPts p σ → ℂ} (hf : f ∈ R.holSec _) : assocVal f ∈ R.holSec _

theorem **sum_tmul_eq_zero** {E Ω A ι : Type*} [Field E] [Field Ω] [IsAlgClosed Ω] [CommRing A] [Algebra E Ω] [Algebra E A] [Algebra.FormallyUnramified E A] [Algebra.FiniteType E A] [Fintype ι] (c : ι → Ω) (a : ι → A) (h : ∀ φ : A →ₐ[E] Ω, ∑ i, c i * φ (a i) = 0) : ∑ i, c i ⊗ₜ[E] a i = 0

theorem **finite_cxPts** (hq : IsFinite q) (τ : E →+* ℂ) : Finite (CxPts q τ)

theorem **linearIndependent_cval** (hq : Etale q) (hB : B.IsLB) (τ : E →+* ℂ) {ι : Type*} [Fintype ι] {g : ι → H0 q B ⊤} (hg : LinearIndependent E g) : LinearIndependent ℂ fun i y => cval τ (g i) (pick hB y) y

theorem **form_pointHerm** (τ : E →+* ℂ) [Fintype (CxPts q τ)] (g g' : H0 q B ⊤) : (pointHerm hf he φ).form τ g g' = ∑ y, starRingEnd ℂ (cval τ g (pick (φ τ).isLB y) y) * cval τ g' (pick (φ τ).isLB y) y * (Real.exp (-(φ τ).φ (pick (φ τ).isLB y) y) : ℝ)

theorem **csq_pointHerm** (τ : E →+* ℂ) {ι : Type*} [Fintype ι] (c : ι → ℂ) (g : ι → H0 q B ⊤) : csq (pointHerm hf he φ) τ c g = ∑ᶠ y, (φ τ).nsq (∑ i, c i • cval τ (g i)) y

theorem **nrm_pointHerm_sq** (τ : E →+* ℂ) (g : H0 q B ⊤) : (pointHerm hf he φ).nrm τ g ^ 2 = ∑ᶠ y, (φ τ).nsq (cval τ g) y

Development.AbelianVarietyTheory.Definitions.Divisors.LocallyCompactSpaceAxis

theorem **exists_data** (hc : IsCMExt F E σ) (hu : Algebra.Unramified (𝒪 F) (𝒪 E)) (hJ : IsHermND σ J) (hW : IsWittFrame σ J X Y) {L : Submodule (𝒪 E) (Fin n → E)} {a : GL (Fin n) (AF F ⊗[F] E)} (hA : GlobalTheory.EisLine.IsAdapted σ J X Y L a) : ∃ h, Data σ J X Y L a h

theorem **data_inhabited** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E) (J : Matrix (Fin 2) (Fin 2) E) (X Y : Matrix (Fin 2) (Fin (0 + 1)) E) (L : Submodule (𝒪 E) (Fin 2 → E)) (a : GL (Fin 2) (AF F ⊗[F] E)) (h : GAd σ J → ℝ), Data σ J X Y L a h

theorem **exists_isReps** (x : Matrix (Fin n) (Fin 1) E) : ∃ T, IsReps σ J x T

theorem **scales_of_diag** {C : Matrix (Fin n) (Fin (r + 1)) E} {p : GAd σ J} {l : Fin (r + 1) → (AF F ⊗[F] E)ˣ} (h : (matHom _ σ J p).1 * adMat F C = adMat F C * diagonal (fun k => ((l k : (AF F ⊗[F] E)ˣ) : AF F ⊗[F] E))) : Scales σ J (ResLine.col C) p (l 0)

theorem **exists_isFlow** (hc : IsCMExt F E σ) (hJ : IsHermND σ J) (hW : IsWittFrame σ J X Y) : ∃ a, IsFlow σ J X Y a

theorem **ht_flow** {L : Submodule (𝒪 E) (Fin n → E)} {a : GL (Fin n) (AF F ⊗[F] E)} {h : GAd σ J → ℝ} (H : Data σ J X Y L a h) {c : ℝ → GAd σ J} (hc : IsFlow σ J X Y c) (t : ℝ) : h (c t) = Real.exp t

theorem **isHt_of_data** {L : Submodule (𝒪 E) (Fin n → E)} {a : GL (Fin n) (AF F ⊗[F] E)} {h : GAd σ J → ℝ} (H : Data σ J X Y L a h) : IsHt (levelAll σ J L a) h

theorem **all_inhabited** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E) (J : Matrix (Fin 2) (Fin 2) E) (X Y : Matrix (Fin 2) (Fin (0 + 1)) E) (L : Submodule (𝒪 E) (Fin 2 → E)) (a : GL (Fin 2) (AF F ⊗[F] E)) (h : GAd σ J → ℝ) (c : ℝ → GAd σ J) (T : Set (ratPts σ J)), Data σ J X Y L a h ∧ IsFlow σ J X Y c ∧ IsReps σ J (ResLine.col X) T ∧ IsHt (levelAll σ J L a) h ∧ ∀ t, h (c t) = Real.exp t

theorem **isHt_right** {G : Type} [Group G] [TopologicalSpace G] {K : Subgroup G} {h : G → ℝ} (H : IsHt K h) (g : G) {k : G} (hk : k ∈ K) : h (g * k) = h g

theorem **cont_coord** (j : Fin 2) : Continuous fun g : Pl => g.toAdd j

theorem **isClosed_axis** (i : Fin 2) : IsClosed (axis i : Set Pl)

theorem **single_mem** (i : Fin 2) (c : ℝ) : Multiplicative.ofAdd (Pi.single i c) ∈ axis i

theorem **single_int** (i : Fin 2) (m : ℤ) : Multiplicative.ofAdd (Pi.single i (m : ℝ)) ∈ intPts

theorem **isFD** (i : Fin 2) (μ : Measure (axis i)) : IsFundamentalDomain (intPts.subgroupOf (axis i)) (unitInt i) μ

theorem **fin_unitInt** (i : Fin 2) (μ : Measure (axis i)) [μ.IsHaarMeasure] : μ (unitInt i) ≠ ⊤

theorem **countable_intPts** : Countable intPts

theorem **semiLat_plane** : SemiLat (axis 0) (axis 1) intPts Measure.haar Measure.haar (unitInt 0) (unitInt 1) where

theorem **semiLat_ne_zero** {G : Type} [Group G] [TopologicalSpace G] [MeasurableSpace G] {N M Λ : Subgroup G} {ν : Measure N} {π : Measure M} {𝒩 : Set N} {ℳ : Set M} (H : SemiLat N M Λ ν π 𝒩 ℳ) : ν 𝒩 ≠ 0

theorem **fd_one** {H : Type} [Group H] [MeasurableSpace H] [MeasurableSingletonClass H] (Λ : Subgroup H) (hΛ : ∀ g, g ∈ Λ) (μ : Measure H) : IsFundamentalDomain Λ ({1} : Set H) μ

theorem **haar_count** {H : Type} [Group H] [Finite H] [TopologicalSpace H] [DiscreteTopology H] [MeasurableSpace H] [BorelSpace H] : (Measure.count : Measure H).IsHaarMeasure

theorem **semiLat_finite** {G : Type} [Group G] [Finite G] [TopologicalSpace G] [DiscreteTopology G] [MeasurableSpace G] [BorelSpace G] (N M : Subgroup G) [hN : N.Normal] (hd : ∀ g ∈ N, g ∈ M → g = 1) (hs : ∀ g, ∃ u ∈ N, ∃ m ∈ M, g = u * m) : SemiLat N M ⊤ Measure.count Measure.count {1} {1}

Development.AbelianVarietyTheory.Definitions.Divisors.LocallyCompactUnipFlag

theorem **adjA_apply** (X : Matrix (Fin n) (Fin n) (AF F ⊗[F] E)) : adjA σ J X = J⁻¹.map (inclR (F

theorem **adjA_map** (X : Matrix (Fin n) (Fin n) E) : adjA σ J (X.map (inclR (F

theorem **idemA_eq** (C : Matrix (Fin n) (Fin r) E) (L : Matrix (Fin r) (Fin n) E) (j : Fin r) : idemA (F

theorem **sum_idemA** (C : Matrix (Fin n) (Fin r) E) (L : Matrix (Fin r) (Fin n) E) : ∑ j, idemA (F

theorem **lower_iff** {C : Matrix (Fin n) (Fin r) E} {d : Fin r → ℕ} {m : Matrix (Fin n) (Fin n) (AF F ⊗[F] E)} : (∀ i, ∃ a : Fin r → AF F ⊗[F] E, (∀ j, d i ≤ d j → a j = 0) ∧ m *ᵥ (fun k => inclR (F

theorem **form_mul_of_mul_cols** {C : Matrix (Fin n) (Fin r) E} {g : GAd σ J} {B : Matrix (Fin r) (Fin r) (AF F ⊗[F] E)} (h : (matHom _ σ J g).1 ∗ adMat F C = adMat F C ∗ B) : adMat F (cT (σ : E →+∗ E) C ∗ J) ∗ (matHom _ σ J g⁻¹).1 = cT (sigR (AF F) σ) B ∗ adMat F (cT (σ : E →+∗ E) C ∗ J)

theorem **unipFlag_const** (C : Matrix (Fin n) (Fin r) E) (c : ℕ) : unipFlag σ J C (fun _ => c) = unipRad σ J C

theorem **det_isUnit** : IsUnit J.det

theorem **cT_form** : cT (σ : E →+∗ E) J = J

theorem **cT_form_inv** : cT (σ : E →+∗ E) J⁻¹ = J⁻¹

theorem **form_mul_inv** : J.map (inclR (F

theorem **inv_mul_form** : J⁻¹.map (inclR (F

theorem **adjA_mul** (X Y : Matrix (Fin n) (Fin n) (AF F ⊗[F] E)) : adjA σ J (X ∗ Y) = adjA σ J Y ∗ adjA σ J X

theorem **adjA_mul_self_iff** (g : GL (Fin n) (AF F ⊗[F] E)) : adjA σ J g.1 ∗ g.1 = 1 ↔ g ∈ HR (AF F) σ J

theorem **adjA_mat** (g : GAd σ J) : adjA σ J (matHom _ σ J g).1 = (matHom _ σ J g⁻¹).1

theorem **perp_iff_idemA** (C : Matrix (Fin n) (Fin r) E) (L : Matrix (Fin r) (Fin n) E) (hL : L ∗ C = 1) (m : Matrix (Fin n) (Fin n) (AF F ⊗[F] E)) : (∀ x : Fin n → E, (cT (σ : E →+∗ E) C ∗ J) ∗ᵥ x = 0 → ∃ a, m ∗ᵥ (fun i => inclR (F

theorem **idemA_complete** (C : Matrix (Fin n) (Fin r) E) (L : Matrix (Fin r) (Fin n) E) (hL : L ∗ C = 1) (h0 : cT (σ : E →+∗ E) C ∗ J ∗ C = 0) (hLL : L ∗ J⁻¹ ∗ cT (σ : E →+∗ E) L = 0) : CompleteOrthogonalIdempotents (fam (adjA σ J) (idemA C L))

theorem **adjA_adjA** (X : Matrix (Fin n) (Fin n) (AF F ⊗[F] E)) : adjA σ J (adjA σ J X) = X

theorem **exists_dualCols** {C : Matrix (Fin n) (Fin r) E} (hC : IsIsoCols σ J C) : ∃ L : Matrix (Fin r) (Fin n) E, L ∗ C = 1 ∧ L ∗ J⁻¹ ∗ cT (σ : E →+∗ E) L = 0

theorem **mem_unipSet_iff** (C : Matrix (Fin n) (Fin r) E) (L : Matrix (Fin r) (Fin n) E) (hL : L ∗ C = 1) (h0 : cT (σ : E →+∗ E) C ∗ J ∗ C = 0) (hLL : L ∗ J⁻¹ ∗ cT (σ : E →+∗ E) L = 0) {d : Fin r → ℕ} {M : ℕ} (hd : ∀ j, d j < M) (g : GAd σ J) : g ∈ unipSet σ J C d ↔ (matHom _ σ J g).1 − 1 ∈ raise (fam (adjA σ J) (idemA C L)) (wt d M) 0

theorem **mem_unipFlag_iff_raise** (C : Matrix (Fin n) (Fin r) E) (L : Matrix (Fin r) (Fin n) E) (hL : L ∗ C = 1) (h0 : cT (σ : E →+∗ E) C ∗ J ∗ C = 0) (hLL : L ∗ J⁻¹ ∗ cT (σ : E →+∗ E) L = 0) {d : Fin r → ℕ} {M : ℕ} (hd : ∀ j, d j < M) (g : GAd σ J) : g ∈ unipFlag σ J C d ↔ (matHom _ σ J g).1 − 1 ∈ raise (fam (adjA σ J) (idemA C L)) (wt d M) 0

theorem **mem_unipFlag** {g : GAd σ J} : g ∈ unipFlag σ J C d ↔ g ∈ unipSet σ J C d

theorem **unipFlag_le_parFlag** : unipFlag σ J C d ≤ parFlag σ J C d

theorem **conj_mem_unipFlag** {p u : GAd σ J} (hp : p ∈ parFlag σ J C d) (hu : u ∈ unipFlag σ J C d) : p ∗ u ∗ p⁻¹ ∈ unipFlag σ J C d

theorem **isClosed_unipFlag** : IsClosed (unipFlag σ J C d : Set (GAd σ J))

theorem **locallyCompact_unipFlag** : LocallyCompactSpace (unipFlag σ J C d)

theorem **cocompact_unipFlag** : GlobalTheory.Cusp.Cocompact (ratPts σ J) (unipFlag σ J C d)

Development.AbelianVarietyTheory.Definitions.Divisors.LogAbsVpOfNormalised

theorem **mem_eigSubLat** {χ : Λ →+∗ C} {X : Submodule O V} {x : V} : x ∈ eigSubLat ρ χ X ↔ x ∈ X ∧ x ∈ eigenSub ρ χ

theorem **mem_eigQuotLat** {i : ι} {X : Submodule O V} {x : V} : x ∈ eigQuotLat ρ h i X ↔ ∃ y ∈ X, proj h ρ i y = x

theorem **eigQuotLat_le** (i : ι) (X : Submodule O V) : eigQuotLat ρ h i X ≤ (eigenSub ρ (σ i)).restrictScalars O

theorem **_root_.GlobalTheory.HTLat.eigSubLat_le_eigQuotLat** (i : ι) (X : Submodule O V) : _root_.GlobalTheory.HTLat.eigSubLat ρ (σ i) X ≤ _root_.GlobalTheory.HTLat.eigQuotLat ρ h i X

theorem **_root_.GlobalTheory.HTLat.eigSubLat_mono** (χ : Λ →+∗ C) {X Y : Submodule O V} (hXY : X ≤ Y) : _root_.GlobalTheory.HTLat.eigSubLat ρ χ X ≤ _root_.GlobalTheory.HTLat.eigSubLat ρ χ Y

theorem **eigQuotLat_mono** (i : ι) {X Y : Submodule O V} (hXY : X ≤ Y) : eigQuotLat ρ h i X ≤ eigQuotLat ρ h i Y

theorem **map_eigQuotLat** {V' : Type} [AddCommGroup V'] [Module C V'] [Module O V'] [IsScalarTower O C V'] {ρ' : Λ →+∗ Module.End C V'} {f : V →ₗ[C] V'} (hf : IsEquivariant ρ ρ' f) (i : ι) (X : Submodule O V) : (eigQuotLat ρ h i X).map (f.restrictScalars O) = eigQuotLat ρ' h i (X.map (f.restrictScalars O))

theorem **_root_.GlobalTheory.HTLat.algebraMap_embO** (φ : E →+∗ C) (a : 𝒪 E) : algebraMap O C (_root_.GlobalTheory.HTLat.embO hv φ a) = φ a

theorem **_root_.GlobalTheory.HTLat.isSplitting_embO** [IsAlgClosed C] [CharZero C] : _root_.GlobalTheory.EigenSplit.IsSplitting (fun φ : E →+∗ C => (algebraMap O C).comp (_root_.GlobalTheory.HTLat.embO hv φ))

theorem **isSplitting_int** (R : Type) [CommRing R] : IsSplitting (fun _ : Unit => Int.castRingHom R)

theorem **grLat_mono** {La La' : Submodule O U} (h : La ≤ La') (F' F : Submodule C U) : grLat La F' F ≤ grLat La' F' F

theorem **grLat_le** (La : Submodule O U) (F' F : Submodule C U) : grLat La F' F ≤ (F.map F'.mkQ).restrictScalars O

theorem **grLat_eq_inf** (La : Submodule O U) {F' F : Submodule C U} (h : F' ≤ F) : grLat La F' F = La.map (F'.mkQ.restrictScalars O) ⊓ (F.map F'.mkQ).restrictScalars O

theorem **mem_scaleLat** {a : C} {La : Submodule O U} {x : U} : x ∈ scaleLat a La ↔ ∃ y ∈ La, a • y = x

theorem **scaleLat_one** (La : Submodule O U) : scaleLat (1 : C) La = La

theorem **grLat_scaleLat** {a : C} (ha : a ≠ 0) (La : Submodule O U) (F' F : Submodule C U) : grLat (scaleLat a La) F' F = scaleLat a (grLat La F' F)

theorem **latNorm_span_singleton** (hv : v.Integers O) {g : U} (hg : g ≠ 0) (c : C) : latNorm v (Submodule.span O {g}) (c • g) = v c

theorem **exists_generator** (hv : v.Integers O) {La : Submodule O U} (hfg : La.FG) {x : U} (h₁ : La ≤ (Submodule.span C {x}).restrictScalars O) (h₂ : La ≠ ⊥) : ∃ c : C, c ≠ 0 ∧ La = Submodule.span O {c • x}

theorem **_root_.GlobalTheory.HTLat.fg_inf** (hv : v.Integers O) {La : Submodule O U} (hfg : La.FG) (F : Submodule C U) : (La ⊓ F.restrictScalars O).FG

theorem **fg_grLat** (hv : v.Integers O) {La : Submodule O U} (hfg : La.FG) (F' F : Submodule C U) : (grLat La F' F).FG

theorem **_root_.GlobalTheory.HTLat.step_zero** (f : Fin N → U) : _root_.GlobalTheory.HTLat.step C f 0 = ⊥

theorem **_root_.GlobalTheory.HTLat.step_mono** (f : Fin N → U) : Monotone (_root_.GlobalTheory.HTLat.step C f)

theorem **_root_.GlobalTheory.HTLat.mem_step_succ** (f : Fin N → U) (i : Fin N) : f i ∈ _root_.GlobalTheory.HTLat.step C f (i + 1)

theorem **step_succ** (f : Fin N → U) (i : Fin N) : step C f (i + 1) = step C f i ⊔ Submodule.span C {f i}

theorem **map_step** {U' : Type} [AddCommGroup U'] [Module C U'] (g : U →ₗ[C] U') (f : Fin N → U) (i : ℕ) : (step C f i).map g = step C (g ∘ f) i

theorem **map_mkQ_step_succ** (f : Fin N → U) (i : Fin N) : (step C f (i + 1)).map (step C f i).mkQ = Submodule.span C {(step C f i).mkQ (f i)}

theorem **_root_.GlobalTheory.HTLat.exists_grAt_eq** {v : Valuation C ℝ≥0} (hv : v.Integers O) {La : Submodule O U} (hfg : La.FG) (f : Fin N → U) (i : Fin N) (hf : f i ∉ _root_.GlobalTheory.HTLat.step C f i) {a : C} (ha : a ≠ 0) {y : U} (hy : y ∈ _root_.GlobalTheory.HTLat.step C f i) (hLa : a • f i + y ∈ La) : ∃ c : C, c ≠ 0 ∧ _root_.GlobalTheory.HTLat.grAt C La f i = Submodule.span O {c • (_root_.GlobalTheory.HTLat.step C f i).mkQ (f i)} ∧ _root_.GlobalTheory.HTLat.grNorm v La f i = (v c)⁻¹

theorem **AdaptedFrame.notMem_step** {W U₁ : Submodule C U} {j : ℕ} (F : AdaptedFrame W U₁ N j) (i : Fin N) : F.f i ∉ step C F.f i

theorem **AdaptedFrame.finrank_step** {W U₁ : Submodule C U} {j : ℕ} (F : AdaptedFrame W U₁ N j) {i : ℕ} (hi : i ≤ N) : Module.finrank C (step C F.f i) = i

theorem **nonempty_adaptedFrame** {W U₁ : Submodule C U} (h : W ≤ U₁) [FiniteDimensional C U₁] : Nonempty (AdaptedFrame W U₁ (Module.finrank C U₁) (Module.finrank C W))

theorem **logAbs_vp_mul** (p : ℕ) (v : Valuation C ℝ≥0) {r s : ℝ≥0} (hr : r ≠ 0) (hs : s ≠ 0) : logAbs_vp p v (r * s) = logAbs_vp p v r + logAbs_vp p v s

theorem **logAbs_vp_self** (p : ℕ) (v : Valuation C ℝ≥0) (h0 : v (p : C) ≠ 0) (h1 : v (p : C) < 1) : logAbs_vp p v (v (p : C)) = -Real.log p

theorem **logAbs_vp_of_normalised** {p : ℕ} (hp : 1 < p) (v : Valuation C ℝ≥0) (h : v (p : C) = (p : ℝ≥0)⁻¹) (r : ℝ≥0) : logAbs_vp p v r = Real.log r

theorem **AdaptedFrame.map_notMem_step** (al : U →ₗ[C] U') {U₁ : Submodule C U} {j : ℕ} (F : AdaptedFrame (LinearMap.ker al ⊓ U₁) U₁ N j) (i : Fin N) (hi : j ≤ i) : (al ∘ F.f) i ∉ step C (al ∘ F.f) i

Development.AbelianVarietyTheory.Definitions.Divisors.MapBaseTiedSel

theorem **Bundle.Comparison₂.psi** {b : Bundle D O c} (t : b.Comparison₂) : PsiPPTie D b.P.P0 b.P.ψ0 b.P.Ps b.P.ψs b.P.I.eG

theorem **Bundle.Comparison₂.pins3** {b : Bundle D O c} (t : b.Comparison₂) : RealPins3 b.B.Q

theorem **Bundle.Comparison₂.pins** {b : Bundle D O c} (t : b.Comparison₂) : RealPins b.B.Q

theorem **Bundle.Comparison₂.recPins** {b : Bundle D O c} (t : b.Comparison₂) : RealRecPins t.pins t.1

theorem **zsc_eq_ind** (P : RowSideFamily O.ex) (Del h0 : ℕ) (c : ℝ) : Ind.zsc P Del h0 c = zsc P Del h0 c

theorem **exists_part_eq** (p : Ind.Part D O c) (C : GlobalTheory.Cp.SquareIntegrableColumnsCompatible p.R.T p.P) (hD : p.Dcf = C.c.face) (hS : HEq p.Spf C.Sp) : ∃ (b : Bundle D O c) (t : b.Comparison₂), b.part t = p

theorem **map_base_tiedSel** (σ : RawSel Bundle 𝒜) (τ : TieSel σ) : (RawSel.map TiedBundle.base (compatibleSelection σ τ) : RawSel Bundle 𝒜) = @σ

theorem **map_part_tiedSel** (σ : RawSel Bundle 𝒜) (τ : TieSel σ) : (RawSel.map TiedBundle.part (compatibleSelection σ τ) : RawSel Ind.Part 𝒜) = @partSel 𝒜 σ τ

theorem **Row.of_tied** {Lk' : LkT TiedBundle} (hQ : ∀ K X f T c, Q K X f T c → Q' K X f T c) (hL : EvP fun _ K => ∀ β : BPt K, β.Is Q → ∀ (c : ℝ) (b : Bundle β.D β.O c), Lk K β c b → ∃ t : b.Comparison₂, Lk' K β c (b, t)) {H : ∀ 𝒜 : FamClass, RawSel TiedBundle 𝒜 → Prop} (h : Row Q' Lk' H) : Row Q Lk fun 𝒜 σ => ∃ τ : TieSel σ, H 𝒜 (compatibleSelection σ τ)

theorem **Row.descend** {Lk' : LkT TiedBundle} (hQ : ∀ K X f T c, Q K X f T c → Q' K X f T c) (hL : EvP fun _ K => ∀ β : BPt K, β.Is Q → ∀ (c : ℝ) (b : Bundle β.D β.O c), Lk K β c b → ∃ t : b.Comparison₂, Lk' K β c (b, t)) {H : ∀ 𝒜 : FamClass, RawSel Bundle 𝒜 → Prop} (h : Row Q' Lk' fun 𝒜 σ => H 𝒜 (RawSel.map TiedBundle.base σ)) : Row Q Lk H

theorem **Row.of_part** {Lk' : LkT Ind.Part} (hQ : ∀ K X f T c, Q K X f T c → Q' K X f T c) (hL : EvP fun _ K => ∀ β : BPt K, β.Is Q → ∀ (c : ℝ) (b : Bundle β.D β.O c), Lk K β c b → ∃ t : b.Comparison₂, Lk' K β c (b.part t)) {H : ∀ 𝒜 : FamClass, RawSel Ind.Part 𝒜 → Prop} (h : Row Q' Lk' H) : Row Q Lk fun 𝒜 σ => ∃ τ : TieSel σ, H 𝒜 (partSel σ τ)

theorem **Row.onTied** {H : ∀ 𝒜 : FamClass, RawSel Bundle 𝒜 → Prop} (h : Row Q Lk H) : Row Q Lk.onTied fun 𝒜 σ => H 𝒜 (RawSel.map TiedBundle.base σ)

theorem **Row.of_onTied** (hT : RowE Q Lk fun _ => HasTie) {H : ∀ 𝒜 : FamClass, RawSel Bundle 𝒜 → Prop} (h : Row Q Lk.onTied fun 𝒜 σ => H 𝒜 (RawSel.map TiedBundle.base σ)) : Row Q Lk H

Development.AbelianVarietyTheory.Definitions.Divisors.MapInclKMemTwCentResOne

theorem **sigKQ_inclK** (r : ℕ) (a : K) : sigKQ p K r (inclK p K r a) = inclK p K r a

theorem **sigKQ_inclQ** (r : ℕ) (b : Qr p r) : sigKQ p K r (inclQ p K r b) = inclQ p K r (frobQ p r b)

theorem **frobQ_norm** (r : ℕ) (b : Qr p r) : ‖((frobQ p r b : Qr p r) : PadicAlgCl p)‖ = ‖(b : PadicAlgCl p)‖

theorem **frobQ_intR** (r : ℕ) (b : Qr p r) (hb : b ∈ GlobalTheory.LKLoc.intR p (Q0 p) r) : frobQ p r b ∈ GlobalTheory.LKLoc.intR p (Q0 p) r

theorem **sigKQ_intKQ** (r : ℕ) (x : KQ p K r) (hx : x ∈ intKQ p K r) : sigKQ p K r x ∈ intKQ p K r

theorem **sigRes_Kres** (n r : ℕ) (g : GL (Fin n) (KQ p K r)) (hg : g ∈ Kres p K n r) : sigRes p K n r g ∈ Kres p K n r

theorem **nrmRes_one** (n : ℕ) (δ : GL (Fin n) (KQ p K 1)) : nrmRes p K n 1 δ = δ

theorem **mulMap_intKQ** (r : ℕ) (x : KQ p K r) (hx : x ∈ intKQ p K r) : ‖mulMap p K r x‖ ≤ 1

theorem **inv_p_not_mem_intKQ** (r : ℕ) : inclK p K r (p : K)⁻¹ ∉ intKQ p K r

theorem **Kres_ne_top** (r : ℕ) : Kres p K 1 r ≠ ⊤

theorem **frobQ_congr** (r : ℕ) (b : Qr p r) (hb : ‖(b : PadicAlgCl p)‖ ≤ 1) : ‖((frobQ p r b : Qr p r) : PadicAlgCl p) - (b : PadicAlgCl p) ^ p‖ < 1

theorem **exists_frobQ_ne** {r : ℕ} (hr : 2 ≤ r) : ∃ b : Qr p r, b ≠ 0 ∧ frobQ p r b ≠ b

theorem **inclQ_injective** (r : ℕ) : Function.Injective (inclQ p K r)

theorem **twCentRes_one_ne_top** {r : ℕ} (hr : 2 ≤ r) : twCentRes p K 1 r 1 ≠ ⊤

theorem **map_inclK_mem_twCentRes_one** (n r : ℕ) (g : GL (Fin n) K) : GeneralLinearGroup.map (inclK p K r).toRingHom g ∈ twCentRes p K n r 1

theorem **finrank_KQ** {r : ℕ} (hr : 0 < r) : Module.finrank ℚ_[p] (KQ p K r) = Module.finrank ℚ_[p] K * r

theorem **TOResU_gl_one_of_le** (γ : GL (Fin 1) (KQ p K 1)) (U : Subgroup (GL (Fin 1) (KQ p K 1))) (hU : U ≤ Kres p K 1 1) (φ : GL (Fin 1) (KQ p K 1) → ℂ) : TOResU p K 1 1 γ U φ = φ γ / (U.relIndex (Kres p K 1 1) : ℂ)

theorem **TOHT_torus_degree_one** (m : ℕ) (Kw : Fin m → IntermediateField ℚ_[p] (PadicAlgCl p)) (γ : GHT p m 1 1 Kw) (φ : GHT p m 1 1 Kw → ℂ) : TOHT p m 1 1 Kw γ φ = φ γ

theorem **sigHT_KHT** (m n r : ℕ) (Kw : Fin m → IntermediateField ℚ_[p] (PadicAlgCl p)) (g : GHT p m n r Kw) (hg : g ∈ KHT p m n r Kw) : sigHT p m n r Kw g ∈ KHT p m n r Kw

theorem **isTopologicalRing_KQ** (hK : FiniteDimensional ℚ_[p] K) {r : ℕ} (hr : 0 < r) : IsTopologicalRing (KQ p K r)

Development.AbelianVarietyTheory.Definitions.Divisors.MapSmulOfIsHausdorff

@[simp] theorem **lev_mk** (s : ℕ → M) (h : ∀ n, s n ^ p ^ n = 1 ∧ s (n + 1) ^ p = s n) (n : ℕ) : (mk s h).lev n = s n

theorem **lev_pow** (t : Tate p M) (n : ℕ) : t.lev n ^ p ^ n = 1

theorem **lev_succ** (t : Tate p M) (n : ℕ) : t.lev (n + 1) ^ p = t.lev n

@[**ext**] theorem ext {s t : Tate p M} (h : ∀ n, s.lev n = t.lev n) : s = t

@[simp] theorem **lev_add** (s t : Tate p M) (n : ℕ) : (s + t).lev n = s.lev n * t.lev n

@[simp] theorem **lev_zero** (n : ℕ) : (0 : Tate p M).lev n = 1

theorem **lev_add_pow** (t : Tate p M) (n k : ℕ) : t.lev (n + k) ^ p ^ k = t.lev n

theorem **lev_smul** (a : ℤ_[p]) (t : Tate p M) (n : ℕ) : (a • t).lev n = t.lev n ^ digit n a

@[simp] theorem **lev_map** {M' : Type} [CommGroup M'] (f : M →* M') (t : Tate p M) (n : ℕ) : (map p f t).lev n = f (t.lev n)

theorem **map_injective** {M' : Type} [CommGroup M'] {f : M →* M'} (hf : Function.Injective f) : Function.Injective (map p f)

theorem **map_smul_of_isHausdorff** {T O : Type} [AddCommGroup T] [Module ℤ_[p] T] [CommRing O] [Algebra ℤ_[p] O] [IsHausdorff (Ideal.span {(p : O)}) O] (f : T →+ O) (a : ℤ_[p]) (t : T) : f (a • t) = a • f t

theorem **Eps.isPrimitiveRoot** (ε : Eps p C) (n : ℕ) : IsPrimitiveRoot (ε.1.lev n) (p ^ n)

theorem **Eps.bijective** (ε : Eps p C) : Function.Bijective (LinearMap.toSpanSingleton ℤ_[p] (Tate p Cˣ) ε.1)

theorem **Eps.lev_equiv** (ε : Eps p C) (a : ℤ_[p]) (n : ℕ) : (ε.equiv a).lev n = ε.1.lev n ^ digit n a

theorem **weilIs_iff** (hk : 1 ≤ k) {z s : Pts A.A} (hz : z ^ k = 1) (hs : s ^ k = 1) (w : Cˣ) : Out.WeilIs A.A A.L (k : ℤ) z s (unitOn w) ↔ w = weil A k z s

theorem **weil_mul_left** (hk : 1 ≤ k) {z z' s : Pts A.A} (hz : z ^ k = 1) (hz' : z' ^ k = 1) (hs : s ^ k = 1) : weil A k (z * z') s = weil A k z s * weil A k z' s

theorem **weil_mul_right** (hk : 1 ≤ k) {z z' s : Pts A.A} (hz : z ^ k = 1) (hz' : z' ^ k = 1) (hs : s ^ k = 1) : weil A k s (z * z') = weil A k s z * weil A k s z'

theorem **weil_level** (hk : 1 ≤ k) {l : ℕ} (hl : 1 ≤ l) {z s : Pts A.A} (hz : z ^ k = 1) (hs : s ^ (k * l) = 1) : weil A (k * l) z s = weil A k z (s ^ l)

theorem **weil_perfect** (hk : 1 ≤ k) (κ : Pts A.A → Cˣ) (hκ : ∀ z z', z ^ k = 1 → z' ^ k = 1 → κ (z * z') = κ z * κ z') : ∃! s : Pts A.A, s ^ k = 1 ∧ ∀ z, z ^ k = 1 → weil A k z s = κ z

theorem **weil_self** (hk : 1 ≤ k) {z : Pts A.A} (hz : z ^ k = 1) : weil A k z z = 1

theorem **weil_pow_left** (hk : 1 ≤ k) {z s : Pts A.A} (hz : z ^ k = 1) (hs : s ^ k = 1) (m : ℕ) : weil A k (z ^ m) s = weil A k z s ^ m

theorem **weil_pow_right** (hk : 1 ≤ k) {z s : Pts A.A} (hz : z ^ k = 1) (hs : s ^ k = 1) (m : ℕ) : weil A k s (z ^ m) = weil A k s z ^ m

theorem **one_le_ppow** (n : ℕ) : 1 ≤ p ^ n

theorem **lev_weilT** (s t : TateA A p) (n : ℕ) : (weilT A p s t).lev n = weil A (p ^ n) (s.lev n) (t.lev n)

theorem **pow_pairT** (ε : Eps p C) (s t : TateA A p) (n : ℕ) : ε.1.lev n ^ digit n (pairT A ε s t) = weil A (p ^ n) (s.lev n) (t.lev n)

theorem **isAlt_pairT** (ε : Eps p C) : (pairT A ε).IsAlt

theorem **actPts_add** (T : Over S) (a b : Λ) (x : T ⟶ G) : x ≫ α.act (a + b) = (x ≫ α.act a) * (x ≫ α.act b)

theorem **lev_actTate** (p : ℕ) [Fact p.Prime] (T : Over S) (a : Λ) (t : Tate p (T ⟶ G)) (n : ℕ) : (actTate α p T a t).lev n = t.lev n ≫ α.act a

theorem **weil_act** {k : ℕ} (hk : 1 ≤ k) (a : Λ) {z s : Pts A.A} (hz : z ^ k = 1) (hs : s ^ k = 1) : weil A k (z ≫ o.act a) s = weil A k z (s ≫ o.act (star a))

theorem **pairT_act** {p : ℕ} [Fact p.Prime] (ε : Eps p C) (a : Λ) (s t : TateA A p) : pairT A ε (actTate o.toRingAct p _ a s) t = pairT A ε s (actTate o.toRingAct p _ (star a) t)

Development.AbelianVarietyTheory.Definitions.Divisors.MarkAd

theorem **SerOf.norm** {O : OMDat C R i Tp tup} {κ : WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ : (R.MW i (sg0 D A)).Vec} {cc : (C.U ((i : ℕ) + 1) O.s0).Comp → ℂ} {ph : ℂ} (h : SerOf O κ v₀ cc ph) : ‖ph‖ = 1

Development.AbelianVarietyTheory.Definitions.Divisors.MatchingOfParts

theorem **agree_of_parts** (h1 : KDecay) (h2 : KConst) (h3 : KCoinv) (h4 : KBeta) (h5 : KAlpha) : ThetaIntegralsAgree

theorem **matching_of_parts** (h1 : KReal) (h2 : KFin) : LocalMatching

Development.AbelianVarietyTheory.Definitions.Divisors.MaxWeightGL1EqOne

theorem **volRat_self** (V : Subgroup G) : volRat V V = 1

Development.AbelianVarietyTheory.Definitions.Divisors.MeanLogLeOfBounds

theorem **PointRes.mono** {X : Data F} {b b' J J' : ℝ} (h : PointRes X b J) (hb : b ≤ b') (hJ : J ≤ J') : PointRes X b' J'

theorem **PointRes.exists_coord** {X : Data F} {b J : ℝ} (h : PointRes X b J) : ∃ (m : ℕ) (θ : Fin m → X.V 0) (B : Fin m → (F →+* ℂ) → ℝ), (∀ i, θ i ∈ X.M 0) ∧ Submodule.span F (Set.range θ) = ⊤ ∧ (∀ i σ, 0 < B i σ) ∧ (∀ i, meanLog F (B i) ≤ b) ∧ ∀ (E : Type) [Field E] [NumberField E] [Algebra F E] (s : Fin X.D → Fin X.N → E), (∀ i, s i ∈ X.latticeOver E) → ∀ hs : (X.baseChange E).IsGood s, Split X.toTuple E s → ∃ Co : Coord (X.baseChange E).toGeo hs (X.geomPts E (X.linSection E s)).ncard, X.cycleHeight E (X.linSection E s) = (∑ p, vecHt (fun k => Co.gen k p) fun σ => Co.c₁ σ p) ∧ ∀ p, ∃ γ : (E →+* ℂ) → ℝ, (∀ τ, 0 < γ τ) ∧ (∀ i τ, ‖τ (Co.ev 0 (hs.res 0 (X.toExtV E 0 (θ i)).1) p)‖ / Co.c₂ τ p ≤ B i (τ.comp (algebraMap F E)) / γ τ) ∧ −meanLog E γ ≤ J + ((X.D : ℝ) + 3) * vecHt (fun k => Co.gen k p) fun τ => Co.c₁ τ p

Development.AbelianVarietyTheory.Definitions.Divisors.MeasurableCompInfGL

theorem **mem_rspan** {π : CuspSub E a} {φ : Sm E a} (h : φ ∈ π.V) : φ.1 ∈ rspan π

theorem **rt_mem_rspan** {π : CuspSub E a} {F : GA E a → ℂ} (hF : F ∈ rspan π) (h : GA E a) : (fun g => F (g * h)) ∈ rspan π

theorem **kerInf_mem_rspan** {π : CuspSub E a} {F : GA E a → ℂ} (hF : F ∈ rspan π) {X : Type} [MeasurableSpace X] (ν : Measure X) [SFinite ν] {k : X → ℂ} {γ : X → GL (Fin a) (Cinf E)} (hk : IsRapidInf ν k γ) : kerInf ν k γ F ∈ rspan π

theorem **rspan_le** {π : CuspSub E a} {S : Submodule ℂ (GA E a → ℂ)} (h₁ : ∀ φ ∈ π.V, φ.1 ∈ S) (h₂ : ∀ F ∈ S, ∀ h : GA E a, (fun g => F (g * h)) ∈ S) (h₃ : ∀ F ∈ S, ∀ (X : Type) [MeasurableSpace X] (ν : Measure X) [SFinite ν] (k : X → ℂ) (γ : X → GL (Fin a) (Cinf E)), IsRapidInf ν k γ → kerInf ν k γ F ∈ S) : rspan π ≤ S

theorem **one_le_htInf** (y : GL (Fin a) (Cinf E)) : 1 ≤ htInf y

theorem **htInf_pos** (y : GL (Fin a) (Cinf E)) : 0 < htInf y

theorem **htGL_le_htInf** (y : GL (Fin a) (Cinf E)) (w : InfinitePlace E) : htGL (Matrix.GeneralLinearGroup.map (Pi.evalRingHom (fun _ : InfinitePlace E => ℂ) w) y) ≤ htInf y

theorem **norm_entry_le_htInf** (y : GL (Fin a) (Cinf E)) (i j : Fin a) (w : InfinitePlace E) : ‖(y : Matrix (Fin a) (Fin a) (Cinf E)) i j w‖ ≤ htInf y ∧ ‖((y⁻¹ : GL (Fin a) (Cinf E)) : Matrix (Fin a) (Fin a) (Cinf E)) i j w‖ ≤ htInf y

theorem **height_le_htInf** (y : GL (Fin a) (Cinf E)) : height y ≤ 3 * htInf y

theorem **IsRapidInf.of_htInf** {X : Type} [MeasurableSpace X] {ν : Measure X} {k : X → ℂ} {γ : X → GL (Fin a) (Cinf E)} (hk : Measurable k) (hγ : ∀ i j, Measurable fun x => (γ x : Matrix (Fin a) (Fin a) (Cinf E)) i j) (hm : ∀ N : ℕ, Integrable (fun x => ‖k x‖ * htInf (γ x) ^ N) ν) : IsRapidInf ν k γ

theorem **measurable_chart** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) : Measurable (chart φ g)

theorem **measurable_comp_infGL** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) {X : Type} [MeasurableSpace X] {γ : X → GL (Fin a) (Cinf E)} (hγ : Measurable fun x => (γ x : Matrix (Fin a) (Fin a) (Cinf E))) : Measurable fun x => φ (g * infGL E a (γ x))

theorem **IsRapidInf.integrable** {X : Type} [MeasurableSpace X] {ν : Measure X} {k : X → ℂ} {γ : X → GL (Fin a) (Cinf E)} (hk : IsRapidInf ν k γ) {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) {g : GA E a} (hg : ∃ (C : ℝ) (N : ℕ), ∀ y, ‖φ (g * infGL E a y)‖ ≤ C * height y ^ N) : Integrable (fun x => k x * φ (g * infGL E a (γ x))) ν

theorem **rspan_ne_bot** (π : CuspSub E a) : rspan π ≠ ⊥

theorem **rspan_le_leftInv** (π : CuspSub E a) : rspan π ≤ SpanModel.leftInv E a

theorem **rspan_left** (π : CuspSub E a) {F : GA E a → ℂ} (hF : F ∈ rspan π) (γ : GL (Fin a) E) (g : GA E a) : F (ratGL E a γ * g) = F g

theorem **rspan_eq_mspanR** (π : CuspSub E a) : rspan π = mspanR π

theorem **plGL_apply** (w : InfinitePlace E) (g : GL (Fin a) ℂ) (i j : Fin a) (w' : InfinitePlace E) : (plGL w g : Matrix (Fin a) (Fin a) (Cinf E)) i j w' = if w' = w then (g : Matrix (Fin a) (Fin a) ℂ) i j else (1 : Matrix (Fin a) (Fin a) ℂ) i j

theorem **map_plGL** (w : InfinitePlace E) (g : GL (Fin a) ℂ) (w' : InfinitePlace E) : Matrix.GeneralLinearGroup.map (Pi.evalRingHom (fun _ : InfinitePlace E => ℂ) w') (plGL w g) = if w' = w then g else 1

theorem **isRapidInf_dirac** (y : GL (Fin a) (Cinf E)) : IsRapidInf (Measure.dirac ()) (fun _ : Unit => (1 : ℂ)) (fun _ => y)

theorem **kerInf_dirac** (y : GL (Fin a) (Cinf E)) (F : GA E a → ℂ) (g : GA E a) : kerInf (Measure.dirac ()) (fun _ : Unit => (1 : ℂ)) (fun _ => y) F g = F (g * infGL E a y)

Development.AbelianVarietyTheory.Definitions.Divisors.MemKFinOfIsLocallyConstant

theorem **blockMatrix_mul_blockColumn** {k : Type} (a b c d : Matrix (Fin n) (Fin n) A) (x y : Matrix (Fin n) k A) : StdUnn.blk4 a b c d * (Matrix.fromRows x y).submatrix finSumFinEquiv.symm id = (Matrix.fromRows (a * x + b * y) (c * x + d * y)).submatrix finSumFinEquiv.symm id

theorem **colY_mul** {k : Type} (t : Matrix (Fin n) k A) : columnBlock A n * t = (Matrix.fromRows t (0 : Matrix (Fin n) k A)).submatrix finSumFinEquiv.symm id

theorem **col_form** {k k' : Type} (x y : Matrix (Fin n) k A) (x' y' : Matrix (Fin n) k' A) : ((Matrix.fromRows x y).submatrix finSumFinEquiv.symm id)ᵀ * (StdUnn.blk4 0 1 (-1) 0 : Matrix (Fin (n + n)) (Fin (n + n)) A) * (Matrix.fromRows x' y').submatrix finSumFinEquiv.symm id = xᵀ * y' - yᵀ * x'

theorem **colY_map** {B : Type} [CommRing B] {Φ : Type} [FunLike Φ A B] [RingHomClass Φ A B] (f : Φ) : (columnBlock A n).map f = columnBlock B n

theorem **colX_map** {B : Type} [CommRing B] {Φ : Type} [FunLike Φ A B] [RingHomClass Φ A B] (f : Φ) : (colX A n).map f = colX B n

theorem **isLagr_colY_blk** {R : Type} [CommRing R] [Algebra R A] (c : A →ₐ[R] A) : IsLagr c (StdUnn.blk4 0 1 (-1) 0) (columnBlock A n)

@[simp] theorem **conjR_tmul** (r : R) (e : E) : conjR R σ (r ⊗ₜ[F] e) = r ⊗ₜ[F] σ e

theorem **conjR_eq_sigR** (z : R ⊗[F] E) : conjR R σ z = sigR R σ z

theorem **map_conjR** {α β : Type} (M : Matrix α β (R ⊗[F] E)) : M.map (conjR R σ) = M.map (sigR R σ)

theorem **UG_eq_HR** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : UG (conjR R σ) (JR R Q) = HR R σ Q

@[simp] theorem **coe_toUG** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (g : HA R σ Q) : ((toUG R σ hQ g : GL (Fin (n + n)) (R ⊗[F] E)) : Matrix _ _ _) = matR R σ g

@[simp] theorem **matR_toUG_symm** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) (g : UG (conjR R σ) (JR R Q)) : matR R σ ((toUG R σ hQ).symm g) = ((g : GL (Fin (n + n)) (R ⊗[F] E)) : Matrix _ _ _)

theorem **continuous_toUG** [TopologicalSpace R] (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : Continuous (toUG R σ hQ)

theorem **isLagr_colY** (hQ : Q = StdUnn.blk4 0 1 (-1) 0) : IsLagr (conjR R σ) (JR R Q) (columnBlock (R ⊗[F] E) n)

theorem **matR_mul_colY** (p : upperParabolic R σ Q) : matR R σ p.1 * columnBlock (R ⊗[F] E) n = columnBlock (R ⊗[F] E) n * (leviUp R σ Q p : Matrix (Fin n) (Fin n) (R ⊗[F] E))

theorem **kFin_iff** (φ : V) : FiniteDimensional ℂ (Submodule.span ℂ {φ' | ∃ k ∈ K, φ' = om k φ}) ↔ ∃ W : Submodule ℂ V, FiniteDimensional ℂ W ∧ ∀ k ∈ K, om k φ ∈ W

@[simp] theorem **coefMap_apply** (l : V →ₗ[ℂ] ℂ) (φ : V) (g : G) : coefMap om l φ g = l (om g φ)

theorem **coefMap_om** (l : V →ₗ[ℂ] ℂ) (φ : V) (h : G) : coefMap om l (om h φ) = fun g => coefMap om l φ (g * h)

theorem **mem_kFin_of_isLocallyConstant** [TopologicalSpace G] (hK : IsCompact (K : Set G)) {φ : V} (hφ : φ ∈ S) (h : IsLocallyConstant fun g => om g φ) : φ ∈ kFin S om K

theorem **mem_secOf** : f ∈ secOf σ hQ C om l K ↔ ∃ φ ∈ C.sp κ, FiniteDimensional ℂ (Submodule.span ℂ {φ' | ∃ k ∈ K, φ' = om (toUG R σ hQ k) φ}) ∧ f = fun g => l (om (toUG R σ hQ g) φ)

theorem **secOf_le_secLoc** (h : IsLocModel σ C ψ JA χ d om) (hl : IsYAdm σ C ψ d l) : secOf σ hQ C om l K ≤ secLoc σ hQ C ψ JA χ K

theorem **isKFinite_of_mem_secOf** (h : f ∈ secOf σ hQ C om l K) : IsKFinite K f

theorem **IsLocModel.apply_upP** (h : IsLocModel σ C ψ JA χ d om) (hl : IsYAdm σ C ψ d l) (p : upperParabolic R σ Q) {φ : (κ → R) → ℂ} (hφ : φ ∈ C.sp κ) : l (om (toUG R σ hQ p) φ) = χ (aUp R σ Q p) * npow (modNorm F E C (aUp R σ Q p)) ((m : ℂ) / 2) * l φ

theorem **law_of_mem_secOf** (h : IsLocModel σ C ψ JA χ d om) (hl : IsYAdm σ C ψ d l) (hf : f ∈ secOf σ hQ C om l K) (p : upperParabolic R σ Q) (g : HA R σ Q) : f (p * g) = χ (aUp R σ Q p) * npow (modNorm F E C (aUp R σ Q p)) ((m : ℂ) / 2) * f g

theorem **isKFinite_of_mem_secLoc** (hf : f ∈ secLoc σ hQ C ψ JA χ K) : IsKFinite K f

theorem **law_of_mem_secLoc** (hf : f ∈ secLoc σ hQ C ψ JA χ K) (p : upperParabolic R σ Q) (g : HA R σ Q) : f (p * g) = χ (aUp R σ Q p) * npow (modNorm F E C (aUp R σ Q p)) ((m : ℂ) / 2) * f g

theorem **indLaw_of_mem_secLoc** (hf : f ∈ secLoc σ hQ C ψ JA χ K) : IndLaw ((modNorm F E C).comp (aUp R σ Q)) (χ.comp (aUp R σ Q)) (((m : ℂ) − n) / 2 + (n : ℂ) / 2) f

theorem **mem_psUp_of_continuous** [TopologicalSpace R] (hf : f ∈ secLoc σ hQ C ψ JA χ K) (hc : Continuous f) : f ∈ psUp R σ Q (modNorm F E C) K χ (((m : ℂ) − n) / 2)

theorem **hTens_eq_trace** {R A α β : Type} [CommRing R] [CommRing A] [Algebra R A] (c : A →ₐ[R] A) [Fintype α] [Fintype β] (JA : Matrix α α A) (JB : Matrix β β A) (x y : Matrix α β A) : hTens c JA JB x y = Matrix.trace (JA^ᵀ * (x * JB * (y.map c)^ᵀ))

theorem **secFin_eq** (v : HeightOneSpectrum (𝒪 F)) [i : Invertible (2 : Fv F v)] : secFin σ hQ ψ χ JA v = secLoc σ hQ (sbClass (Fv F v)) (LatVac.psiAt ψ v) (JR (Fv F v) JA) (charAt χ v) (KUpFin σ Q v)

theorem **modNorm_sbClass** (v : HeightOneSpectrum (𝒪 F)) : modNorm F E (sbClass (Fv F v)) = absEv F E v

theorem **archClass_modu** (w : InfinitePlace F) (hw : w.IsReal) (a : w.Completionˣ) : (archClass w hw).modu a = ‖a.1‖₊

theorem **modNorm_archClass** (w : InfinitePlace F) (hw : w.IsReal) : modNorm F E (archClass w hw) = absEw F E w

theorem **law_of_mem_secFin** {v : HeightOneSpectrum (𝒪 F)} {f : GLoc σ Q v → ℂ} (hf : f ∈ secFin σ hQ ψ χ JA v) (p : upperParabolic (Fv F v) σ Q) (g : GLoc σ Q v) : f (p * g) = charAt χ v (aUp _ σ Q p) * npow (absEv F E v (aUp _ σ Q p)) ((m : ℂ) / 2) * f g

theorem **law_of_mem_secArch** {w : InfinitePlace F} {hw : w.IsReal} {f : GArch σ Q w → ℂ} (hf : f ∈ secArch σ hQ ψ χ JA w hw) (p : upperParabolic w.Completion σ Q) (g : GArch σ Q w) : f (p * g) = charAtInf χ w (aUp _ σ Q p) * npow (absEw F E w (aUp _ σ Q p)) ((m : ℂ) / 2) * f g

theorem **mem_psUp_fin** {v : HeightOneSpectrum (𝒪 F)} {f : GLoc σ Q v → ℂ} (hf : f ∈ secFin σ hQ ψ χ JA v) (hc : Continuous f) : f ∈ psUp _ σ Q (absEv F E v) (KUpFin σ Q v) (charAt χ v) (((m : ℂ) − n) / 2)

Development.AbelianVarietyTheory.Definitions.Divisors.MemL2OfSmoothVector

theorem **mem_typeOf** {σ : Γ →* (K →L[ℂ] K)} {v w : K} : w ∈ typeOf σ v ↔ ∀ c : Γ →₀ ℂ, (∀ γ, algOp σ c (σ γ v) = 0) → algOp σ c w = 0

theorem **self_mem_typeOf** (σ : Γ →* (K →L[ℂ] K)) (v : K) : v ∈ typeOf σ v

theorem **isTestGL_zero** : IsTestGL (fun _ : GA E a => (0 : ℂ))

theorem **mem_smv** {φ : Sm E a} : φ ∈ smv μ H ↔ ∀ l, word l φ ∈ good μ H

theorem **good_of_mem_smv** {φ : Sm E a} (h : φ ∈ smv μ H) : φ ∈ good μ H

theorem **lie_mem_smv** (X : Lie E a) {φ : Sm E a} (h : φ ∈ smv μ H) : lie E a X φ ∈ smv μ H

theorem **memL2_of_smoothVector** : ∀ φ ∈ smv μ H, MemL2Quotient (latticeGL E a) μ φ.1

Development.AbelianVarietyTheory.Definitions.Divisors.MemLevEigenIffTransl

theorem **transl_apply** (γ : LevelAut Dw l) (f : Rul Xo c) (p : TRing c.F) (y : FrPt Xo c p.R) : transl Xo c γ f p y = f p (move Xo c γ y)

theorem **pull_apply** (h : LevelAut Dw l) (f : Rul Xo c) (p : TRing c.F) (y : FrPt Xo c p.R) : pull Xo c h f p y = f p (move Xo c h⁻¹ y)

theorem **pull_inv** (g : LevelAut Dw l) : pull Xo c g⁻¹ = transl Xo c g

theorem **mem_secRing** {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) : f ∈ secRing Xo c

theorem **toSubmodule_secRing** : Subalgebra.toSubmodule (secRing Xo c) = ⊔ e, secZ Xo c e

theorem **prod_mem_secZ** {ι : Type} (s : Finset ι) {e : Ix Dw → ℤ} {f : ι → Rul Xo c} (hf : ∀ i, f i ∈ secZ Xo c e) : ∏ i ∈ s, f i ∈ secZ Xo c (s.card • e)

theorem **levR_move** (γ : LevelAut Dw l) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : levR (levFull Dw l γ) y (move Xo c γ y)

theorem **move_unique** {γ : LevelAut Dw l} {S : Type} [CommRing S] [Algebra c.F S] {y y' : FrPt Xo c S} (h : levR (levFull Dw l γ) y y') : move Xo c γ y = y'

theorem **move_one** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : move Xo c 1 y = y

theorem **move_mul** (γ δ : LevelAut Dw l) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : move Xo c (γ * δ) y = move Xo c δ (move Xo c γ y)

theorem **transl_one** (f : Rul Xo c) : transl Xo c 1 f = f

theorem **transl_mul** (γ δ : LevelAut Dw l) (f : Rul Xo c) : transl Xo c (γ * δ) f = transl Xo c γ (transl Xo c δ f)

theorem **isRightAct_pull** : IsRightAct (pull Xo c)

theorem **transl_mem_secZ** (γ : LevelAut Dw l) {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) : transl Xo c γ f ∈ secZ Xo c e

theorem **pull_mem_secZ** (h : LevelAut Dw l) {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) : pull Xo c h f ∈ secZ Xo c e

theorem **pull_mem_secRing** (h : LevelAut Dw l) {f : Rul Xo c} (hf : f ∈ secRing Xo c) : pull Xo c h f ∈ secRing Xo c

theorem **isRightAct_pullSec** : IsRightAct (pullSec Xo c hpq hA)

theorem **levMat_apply** (u : Rul Xo c) (h g : S) : levMat Xo c S u h g = pull Xo c (h * g) u

theorem **levTr_apply** (χ : S →* (c.F)ˣ) (F : Rul Xo c) : levTr Xo c S χ F = ∑ h : S, (((χ h)⁻¹ : (c.F)ˣ) : c.F) • pull Xo c h F

theorem **levPh_apply** (χ : S →* (c.F)ˣ) (u f : Rul Xo c) (h : S) : levPh Xo c S χ u f h = levTr Xo c S χ (f * pull Xo c h u)

theorem **mem_levEigen** {χ : S →* (c.F)ˣ} {F : Rul Xo c} : F ∈ levEigen Xo c S χ ↔ ∀ h : S, pull Xo c h F = ((χ h : (c.F)ˣ) : c.F) • F

theorem **mem_levEigen_iff_transl** {χ : S →* (c.F)ˣ} {F : Rul Xo c} : F ∈ levEigen Xo c S χ ↔ ∀ g : S, transl Xo c g F = (((χ g)⁻¹ : (c.F)ˣ) : c.F) • F

theorem **isRightAct_pullOn** : IsRightAct (pullOn Xo c S)

theorem **isRightAct_pullSecOn** : IsRightAct (pullSecOn Xo c S hpq hA)

theorem **coe_tr_pullSecOn** (χ : S →* (c.F)ˣ) (F : secRing Xo c) : (tr (pullSecOn Xo c S hpq hA) χ F : Rul Xo c) = levTr Xo c S χ F

theorem **coe_det_pullSecOn** [DecidableEq S] (u : secRing Xo c) : ((groupMat (pullSecOn Xo c S hpq hA) u).det : Rul Xo c) = levDet Xo c S u

theorem **levDet_mem_secZ** {e : Ix Dw → ℤ} {u : Rul Xo c} (hu : u ∈ secZ Xo c e) : levDet Xo c S u ∈ secZ Xo c (Nat.card S • e)

theorem **levTr_mem_levEigen** (χ : S →* (c.F)ˣ) (F : Rul Xo c) : levTr Xo c S χ F ∈ levEigen Xo c S χ

theorem **mem_chiLat** {χ : S →* (c.F)ˣ} {F : cuspW Xo c U G} : F ∈ chiLat S U G χ ↔ F ∈ latW U G ∧ ∀ h : S, pull Xo c h (F : Rul Xo c) = ((χ h : (c.F)ˣ) : c.F) • (F : Rul Xo c)

theorem **coe_cuspRep** (hpq : 1 ≤ Dw.p + Dw.q) (hA : l.A.Coprime l.M) (hC : KeepsCusp U G) (g : LevelAut Dw l) (f : cuspW Xo c U G) : (cuspRep U G hpq hA hC g f : Rul Xo c) = transl Xo c g f

theorem **coe_cuspPh** {G' : WTZ E} (χ : S →* (c.F)ˣ) (u : Rul Xo c) (hm : ∀ f ∈ cuspW Xo c U G', ∀ h, levPh Xo c S χ u f h ∈ cuspW Xo c U G) (f : cuspW Xo c U G') (h : S) : (cuspPh S χ u hm f h : Rul Xo c) = levPh Xo c S χ u f h

theorem **IsTypeLevel.coprime** (hl : IsTypeLevel l) : l.A.Coprime l.M

theorem **isRightAct_pullType** (hpq : 1 ≤ Dw.p + Dw.q) : IsRightAct (pullType Xo c hl fr)

theorem **typeOp_mul** (hpq : 1 ≤ Dw.p + Dw.q) (e e' : MonoidAlgebra c.F (LevelGroup D.host)) : typeOp Xo c hl fr (e * e') = typeOp Xo c hl fr e * typeOp Xo c hl fr e'

theorem **isIdempotentElem_typeOp** (hpq : 1 ≤ Dw.p + Dw.q) {e : MonoidAlgebra c.F (LevelGroup D.host)} (he : IsIdempotentElem e) : IsIdempotentElem (typeOp Xo c hl fr e)

theorem **latN_le** : latN fr U G ψ ≤ latW U G

theorem **mem_latGood_iff** (hpq : 1 ≤ Dw.p + Dw.q) (he : IsIdempotentElem (goodIdemTw om j)) {F : cuspW Xo c U G} : F ∈ latGood hl fr U G ψ om j ↔ F ∈ latN fr U G ψ ∧ goodOp Xo c hl fr om j F = F

theorem **mem_latBad_iff** (hpq : 1 ≤ Dw.p + Dw.q) (he : IsIdempotentElem (goodIdemTw om j)) {F : cuspW Xo c U G} : F ∈ latBad hl fr U G ψ om j ↔ F ∈ latN fr U G ψ ∧ goodOp Xo c hl fr om j F = 0

theorem **IsReduction.mem_ker** {red : stabLat V →* LevelAut Dw l} (h : IsReduction V ι red) (k : stabLat V) : k ∈ red.ker ↔ (k : GA D V) ∈ prinLev V l.M

Development.AbelianVarietyTheory.Definitions.Divisors.MemMixedTsub

theorem **xi_eq** (ε : PreTilt O p) : xi ε = ReciprocityTheory.PGsh.Ainf.xi p O ε

theorem **nu_eq** (ε : PreTilt O p) : nu ε = ReciprocityTheory.PGsh.Ainf.nu p O ε

theorem **mixed_al** (ε : PreTilt O p) (hμ : fontaineTheta O p (mu ε) = 0) (h : ℕ) (S : Finset (Fin h)) (x : Fin h → O) (i : Fin h) : (mixed ε h S).al x i = if i ∈ S then 0 else x i

theorem **mixed_bb** (ε : PreTilt O p) (hμ : fontaineTheta O p (mu ε) = 0) (h : ℕ) (S : Finset (Fin h)) (x : Fin h → O) (i : Fin h) : (mixed ε h S).bb x i = if i ∈ S then x i else 0

theorem **mem_mixed_Fil** (ε : PreTilt O p) (hμ : fontaineTheta O p (mu ε) = 0) (h : ℕ) (S : Finset (Fin h)) (x : Fin h → O) : x ∈ (mixed ε h S).Fil ↔ ∀ i ∈ S, x i = 0

theorem **mem_mixed_Lie** (ε : PreTilt O p) (hμ : fontaineTheta O p (mu ε) = 0) (h : ℕ) (S : Finset (Fin h)) (x : Fin h → O) : x ∈ (mixed ε h S).Lie ↔ ∀ i ∉ S, x i = 0

theorem **mem_mixed_Tsub** (ε : PreTilt O p) (h : ℕ) (S : Finset (Fin h)) (i : Fin h) (x : Fin h → O) : x ∈ (mixed ε h S).Tsub ((padicToO p O).comp (Pi.evalRingHom _ i)) ↔ ∀ j ≠ i, x j = 0

theorem **theta_mu** : fontaineTheta O p (mu ε) = 0

theorem **mu_ne_zero** : mu ε ≠ 0

theorem **isDomain_Ainf** : IsDomain (Ainf p O)

theorem **F_mul_G** : D.F * D.G = mu ε • (1 : Matrix (Fin h) (Fin h) (Ainf p O))

theorem **al_bb** (x : Fin h → O) : D.al (D.bb x) = 0

theorem **bb_al** (x : Fin h → O) : D.bb (D.al x) = 0

theorem **lie_inf_tsub_le_W** (τ : Λ →+* O) : D.Lie ⊓ D.Tsub τ ≤ D.W τ

theorem **map_al_le** (τ : Λ →+* O) : (D.Tsub τ).map D.al ≤ D.Fil ⊓ D.Hsub τ

theorem **bound_aux** (X Y : Matrix (Fin h) (Fin h) (Ainf p O)) (hXY : X * Y = mu ε • 1) (x : Fin h → O) (hx : Y.map (fontaineTheta O p) *ᵥ x = 0) : ∃ y, X.map (fontaineTheta O p) *ᵥ y = fontaineTheta O p (nu ε) • x

theorem **smul_Fil_le** (x : Fin h → O) (hx : x ∈ D.Fil) : fontaineTheta O p (nu ε) • x ∈ LinearMap.range D.al

theorem **smul_ker_le** (x : Fin h → O) (hx : x ∈ LinearMap.ker D.al) : fontaineTheta O p (nu ε) • x ∈ D.Lie

theorem **exists_mixed** (p : ℕ) [Fact p.Prime] : ∃ (_ : Fact ¬IsUnit ((p : ℕ) : 𝒪_ℂ_[p])) (_ : IsAdicComplete (span {((p : ℕ) : 𝒪_ℂ_[p])}) 𝒪_ℂ_[p]) (ε : PreTilt 𝒪_ℂ_[p] p) (D : Datum ε 2 (Fin 2 → ℤ_[p])), IsPrimitiveRoot (((frobeniusEquiv (PreTilt 𝒪_ℂ_[p] p) p).symm ε).untilt) p ∧ mu ε ≠ 0 ∧ D.Fil ≠ ⊥ ∧ D.Fil ≠ ⊤ ∧ D.Lie ≠ ⊥ ∧ D.Lie ≠ ⊤

Development.AbelianVarietyTheory.Definitions.Divisors.MemPass

theorem **ptE_smul** (t : E) (a : OF E) (x : Fin 2 → maximalRealSubfield E) : ptE t (a • x) = algebraMap (OF E) E a * ptE t x

theorem **mem_pass** (t : E) (c : Ideal (OF E)) (y : (Af E)ˣ) (x : Fin 2 → maximalRealSubfield E) : x ∈ pass t c y ↔ ptE t x ∈ lattice c y

Development.AbelianVarietyTheory.Definitions.Divisors.MemShaKernelIffLocallyTrivial

@[simp] theorem **fixMap_apply** (c : A →ₐ[k] A) (c' : A' →ₐ[k'] A') (φ : A →+* A') (hc : ∀ a, φ (c a) = c' (φ a)) (a : fixU k c) : (((fixMap c c' φ hc a : fixU k' c') : A'ˣ) : A') = φ ((a : Aˣ) : A)

@[simp] theorem **locInv_tmul** (S : Type) [Field S] [CharZero S] {L : Type} [CommRing L] [Algebra ℚ L] (c : L →ₐ[ℚ] L) (s : S) (x : L) : locInv S c (s ⊗ₜ x) = s ⊗ₜ c x

Development.AbelianVarietyTheory.Definitions.Divisors.MemSmallSecTrivShift

theorem **ev_pt** (v : V D) (P : MvPolynomial (Fin D) k) : ev _ σ ⊤ (CanonLB.polyFn P) (pt σ v) = MvPolynomial.eval₂Hom σ (fun a => v a) P

theorem **polyFn_surj** : Function.Surjective (CanonLB.polyFn : MvPolynomial (Fin D) k → _)

theorem **diff_pt** (a : Γ(Sp (MvPolynomial (Fin D) k), ⊤)) : DifferentiableOn ℂ (ev (CanonLB.affStr k D) σ ⊤ a ∘ pt σ) univ

theorem **u_pos** (v : V 1) : 0 < u v

theorem **hasFDerivAt_u** (v : V 1) : HasFDerivAt u ((2 * (v 0).re) • Complex.reCLM.comp crd + (2 * (v 0).im) • Complex.imCLM.comp crd) v

theorem **wdb_log** : wdb 1 0 (fun v => (Real.log (u v) : ℂ)) = fun v => v 0 * ((u v)⁻¹ : ℝ)

theorem **cHess_log** (v : V 1) : cHess 1 (fun v => (Real.log (u v) : ℂ)) v 0 0 = ((u v ^ 2)⁻¹ : ℝ)

theorem **integral_dens** : ∫ v : V 1, 1 / Real.pi * (u v ^ 2)⁻¹ = 1

theorem **isCInf_fsLine** : (fsLine σ).IsCInf

theorem **chernDens_fsLine** (v : V 1) : (std σ 1).chernDens (fsLine σ) ((std σ 1).e.symm v) = 1 / Real.pi * (u v ^ 2)⁻¹

theorem **chernInt_fsLine** : (fsLine σ).ChernIntegrableOn 1 univ (fun _ => 1) ∧ (fsLine σ).chernInt 1 univ (fun _ => 1) = 1

theorem **not_isArithVar_zmod** (n : ℕ) [NeZero n] {X : Scheme.{0}} (p : X ⟶ Sp (ZMod n)) (D : ℕ) : ¬ IsArithVar p D

theorem **not_isArithVar_rat** {X : Scheme.{0}} (p : X ⟶ Sp ℚ) (D : ℕ) : ¬ IsArithVar p D

theorem **selfInter_trivShift** [IsDomain k] [CharZero k] [Module.Finite ℤ k] (hd : ringKrullDim k = 1) {t a : ℝ} (h : (trivShift k t).IsSelfInter a) : a = t * Nat.card (k →+* ℂ)

theorem **isNef_trivShift_iff** (t : ℝ) : (ArithBase.trivShift t).IsNef 0 ↔ 0 ≤ t

theorem **mem_smallSec_trivShift** (t : ℝ) (s : (trivShift k t).B.Sec) : s ∈ (trivShift k t).smallSec ↔ ∀ σ (y : CxPts (𝟙 (Sp k)) σ), ‖ev _ σ ⊤ (s.f PUnit.unit) y‖ ≤ Real.exp t

Development.AbelianVarietyTheory.Definitions.Divisors.MemStdLatticeIffIntegralCoords

theorem **hostF_eq** (D : D7) : D.host.F = maximalRealSubfield D.host.E

theorem **sw_tmul** (a : Af D.host.F) (x : D.host.E) : sw D (a ⊗ₜ x) = x ⊗ₜ a

theorem **sw_symm_tmul** (a : Af D.host.F) (x : D.host.E) : (sw D).symm (x ⊗ₜ a) = a ⊗ₜ x

theorem **sw_sig** (y : ReciprocityTheory.Aut.AL D.host.E) : (sw D).symm (ReciprocityTheory.Aut.conjAL D.host.E y) = sigR (Af D.host.F) D.sp.cm.sig ((sw D).symm y)

theorem **mem_GA_iff** {V : HSp D} (hg : V.gram = 1) (κ : GL (Fin V.rank) (ReciprocityTheory.Aut.AL D.host.E)) : κ ∈ ReciprocityTheory.Aut.GA D.host.E V.rank ↔ Formalism.UG.bc (sw D).symm.toRingHom κ ∈ HR (Af D.host.F) D.sp.cm.sig V.gram

theorem **exists_lift** {V : HSp D} (hg : V.gram = 1) (κ : Formalism.AUTREP.GAf D.host.E V.rank) : ∃ g : GA D V, infHom V g = 1 ∧ finHom V g = Formalism.UG.bc (sw D).symm.toRingHom κ.1

theorem **finMat_unitPsi** {V : HSp D} (hg : V.gram = 1) (κ : Formalism.AUTREP.GAf D.host.E V.rank) : finMat D.sp.cm.sig V.gram ((unitPsi hg).ψ κ) = κ.1.1.map (sw D).symm

theorem **mem_std_lattice_iff_integral_coords** {V : HSp D} (h : OrthFrame V) (x : Fin V.rank → D.host.E) : x ∈ L0c D V ↔ ∀ a, x a ∈ (1 : Submodule (𝒪 D.host.E) D.host.E)

theorem **tmul_mem** (o : 𝒪 D.host.E) {a : Af D.host.F} (ha : a ∈ intAf D.host.F) : sw D (a ⊗ₜ (o : D.host.E)) ∈ ReciprocityTheory.Aut.ALint D.host.E

theorem **int_of_mem** {V : HSp D} (h : OrthFrame V) {y : Fin V.rank → Af D.host.F ⊗[D.host.F] D.host.E} (hy : y ∈ latHat D.host.F D.host.E (L0c D V)) (i : Fin V.rank) : sw D (y i) ∈ ReciprocityTheory.Aut.ALint D.host.E

theorem **smul_gen** {M : ℕ} (L : Set (Fin M → D.host.E)) (c : Af D.host.F ⊗[D.host.F] D.host.E) (hc : ∀ a ∈ intAf D.host.F, ∀ x ∈ L, c • (fun i => a ⊗ₜ[D.host.F] x i) ∈ latHat D.host.F D.host.E L) {w : Fin M → Af D.host.F ⊗[D.host.F] D.host.E} (hw : w ∈ latHat D.host.F D.host.E L) : c • w ∈ latHat D.host.F D.host.E L

theorem **smul_closure** {A B : Type} [CommRing A] [CommRing B] {M : ℕ} (e : B →+* A) (S : AddSubgroup (Fin M → A)) (T : Set B) (h : ∀ t ∈ T, ∀ w ∈ S, e t • w ∈ S) {r : B} (hr : r ∈ Subring.closure T) : ∀ w ∈ S, e r • w ∈ S

theorem **smul_mem** {V : HSp D} (h : OrthFrame V) {r : ReciprocityTheory.Aut.AL D.host.E} (hr : r ∈ ReciprocityTheory.Aut.ALint D.host.E) : ∀ w ∈ latHat D.host.F D.host.E (L0c D V), (sw D).symm r • w ∈ latHat D.host.F D.host.E (L0c D V)

theorem **one_mem_intAf** (F : Type) [Field F] [NumberField F] : (1 : Af F) ∈ intAf F

theorem **single_mem** {V : HSp D} (h : OrthFrame V) (j : Fin V.rank) : (Pi.single j 1 : Fin V.rank → Af D.host.F ⊗[D.host.F] D.host.E) ∈ latHat D.host.F D.host.E (L0c D V)

theorem **mem_latHat_iff** {V : HSp D} (h : OrthFrame V) (y : Fin V.rank → Af D.host.F ⊗[D.host.F] D.host.E) : y ∈ latHat D.host.F D.host.E (L0c D V) ↔ ∀ i, sw D (y i) ∈ ReciprocityTheory.Aut.ALint D.host.E

theorem **mulVec_mem** {V : HSp D} (h : OrthFrame V) {κ : Formalism.AUTREP.GAf D.host.E V.rank} (hκ : κ ∈ ReciprocityTheory.Aut.Umax D.host.E V.rank) {y : Fin V.rank → Af D.host.F ⊗[D.host.F] D.host.E} (hy : y ∈ latHat D.host.F D.host.E (L0c D V)) : (κ.1.1.map (sw D).symm.toRingHom) *ᵥ y ∈ latHat D.host.F D.host.E (L0c D V)

theorem **stdPsi_unitPsi** {V : HSp D} (h : OrthFrame V) : StdPsi (hostF_eq D) (unitPsi h.gram_eq)

theorem **exists_stdPsi** (D : D7) (n : ℕ) : ∃ (V : HSp D) (P : RowPsi V V.rank), V.rank = n ∧ StdPsi (hostF_eq D) P

theorem **nontrivial_Af** (F : Type) [Field F] [NumberField F] : Nontrivial (Af F)

theorem **incl_inj** (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] : Function.Injective (Algebra.TensorProduct.includeRight : E →ₐ[F] Af F ⊗[F] E)

theorem **rank_eq** {V : HSp D} {n : ℕ} {hF : D.host.F = maximalRealSubfield D.host.E} {P : RowPsi V n} (h : StdPsi hF P) : V.rank = n

theorem **conjG_conjG** (κ : Formalism.AUTREP.GAf L n) : conjG L n (conjG L n κ) = κ

theorem **exists_normOne** : ∃ u : L, u * IsCMField.complexConj L u = 1 ∧ IsCMField.complexConj L u ≠ u

theorem **conj_diag** {A : Type} [CommRing A] {m n : ℕ} (β : Matrix (Fin m) (Fin n) A) (β' : Matrix (Fin n) (Fin m) A) (h : β * β' = 1) (c : A) : (β * (Matrix.diagonal fun _ => c : Matrix (Fin n) (Fin n) A) : Matrix (Fin m) (Fin n) A) * β' = c • (1 : Matrix (Fin m) (Fin m) A)

theorem **not_stdPsi_twist** {V : HSp D} {n : ℕ} (hn : 0 < n) {P : RowPsi V n} (h : StdPsi (hostF_eq D) P) : ¬ StdPsi (hostF_eq D) (twist P)

theorem **exists_std_and_not** (D : D7) {n : ℕ} (hn : 0 < n) : ∃ (V : HSp D) (P Q : RowPsi V V.rank), V.rank = n ∧ StdPsi (hostF_eq D) P ∧ ¬ StdPsi (hostF_eq D) Q

Development.AbelianVarietyTheory.Definitions.Divisors.MemSubPtsCutBaseChangeIff

theorem **le_iff_ptOf_mem** {K : Type} [Field K] {Y : Scheme.{0}} (x : Sp K ⟶ Y) (U : Y.Opens) : ⊤ ≤ x ⁻¹ᵁ U ↔ ptOf x ∈ U

theorem **ptOf_spMap_comp** {K K' : Type} [Field K] [Field K'] {Y : Scheme.{0}} (τ : K →+* K') (x : Sp K ⟶ Y) : ptOf (spMap τ ≫ x) = ptOf x

theorem **ptOf_comp** {K : Type} [Field K] {Y Z : Scheme.{0}} (x : Sp K ⟶ Y) (g : Y ⟶ Z) : ptOf (x ≫ g) = g (ptOf x)

theorem **ptOf_resPt** {Y : Scheme.{0}} (y : Y) : ptOf (resPt y) = y

theorem **valR_ne_zero_iff** {K : Type} [Field K] {Y : Scheme.{0}} (x : Sp K ⟶ Y) {U : Y.Opens} (h : ptOf x ∈ U) (f : Γ(Y, U)) : valR x U f ≠ 0 ↔ ptOf x ∈ Y.basicOpen f

theorem **ideal_cutIdeal** {Y : Scheme.{0}} {L : LB Y} (hL : L.IsLB) {ι : Type} (s : ι → L.Sec) (W : Y.affineOpens) (c : L.ι) (h : W.1 ≤ L.U c) : (cutIdeal hL s).ideal W = Ideal.span (Set.range fun i => resG Y h ((s i).f c))

theorem **mem_support_cutIdeal** {Y : Scheme.{0}} {L : LB Y} (hL : L.IsLB) {ι : Type} (s : ι → L.Sec) (y : Y) : y ∈ (cutIdeal hL s).support ↔ ∀ i, y ∉ nonzeroLocus (s i)

theorem **mem_chartAt** (y : S.Y) : y ∈ S.L.U (S.chartAt y)

theorem **comp_str** : x ≫ S.str = spMap (S.baseHom x)

theorem **baseHom_ptOver** {φ : k →+* K} (x : S.PtOver φ) : S.baseHom x.1 = φ

theorem **le_chartAt** : ⊤ ≤ x ⁻¹ᵁ S.L.U (S.chartAt (ptOf x))

theorem **exists_frame_change** (i : S.L.ι) (hi : ptOf x ∈ S.L.U i) : ∃ c : K, c ≠ 0 ∧ (∀ s : S.L.Sec, valR x (S.L.U i) (s.f i) = c * S.val x s) ∧ ∀ (m : ℤ) (s : (lbZPow S.L m).Sec), valR x (S.L.U i) (zfun s i) = c ^ m * S.powVal x s

theorem **val_ne_zero_iff** (s : S.L.Sec) : S.val x s ≠ 0 ↔ ptOf x ∈ nonzeroLocus s

theorem **powVal_ne_zero_iff** {m : ℤ} (s : (lbZPow S.L m).Sec) : S.powVal x s ≠ 0 ↔ ptOf x ∈ nonzeroLocus s

theorem **val_add** (s t : S.L.Sec) : S.val x (s + t) = S.val x s + S.val x t

theorem **val_zero** : S.val x 0 = 0

theorem **valR_algebraMap** (a : k) : valR x ⊤ (algebraMap k Γ(S.Y, ⊤) a) = S.baseHom x a

theorem **val_smul** (a : k) (s : S.L.Sec) : S.val x (algebraMap k Γ(S.Y, ⊤) a • s) = S.baseHom x a * S.val x s

theorem **val_comb** (u : Fin S.N → k) : S.val x (S.comb u) = ∑ j, S.baseHom x (u j) * S.coords x j

theorem **val_spMap_comp** (τ : K →+* K') (s : S.L.Sec) : S.val (spMap τ ≫ x) s = τ (S.val x s)

theorem **powVal_spMap_comp** (τ : K →+* K') {m : ℤ} (s : (lbZPow S.L m).Sec) : S.powVal (spMap τ ≫ x) s = τ (S.powVal x s)

theorem **coords_spMap_comp** (τ : K →+* K') : S.coords (spMap τ ≫ x) = fun j => τ (S.coords x j)

theorem **BasePointFree.coords_ne_zero** (h : S.BasePointFree) (x : Sp K ⟶ S.Y) : S.coords x ≠ 0

theorem **powVal_form** {K : Type} [Field K] (x : Sp K ⟶ S.Y) {d : ℕ} {g : MvPolynomial (Fin S.N) k} (hg : g.IsHomogeneous d) : S.powVal x (S.form hg) = eval₂ (S.baseHom x) (S.coords x) g

theorem **mem_imageIdeal_iff** {d : ℕ} {g : MvPolynomial (Fin S.N) k} (hg : g.IsHomogeneous d) : g ∈ S.imageIdeal ↔ S.form hg = 0

theorem **isHomogeneous_imageIdeal** : S.imageIdeal.IsHomogeneous (homogeneousSubmodule (Fin S.N) k)

theorem **eval₂_coords_eq_zero** {K : Type} [Field K] (x : Sp K ⟶ S.Y) {g : MvPolynomial (Fin S.N) k} (hg : g ∈ S.imageIdeal) : eval₂ (S.baseHom x) (S.coords x) g = 0

theorem **sameImage_of_coords** {K : Type} [Field K] {φ : k →+* K} (x x' : S.PtOver φ) {c : K} (hc : c ≠ 0) (h : S.coords x'.1 = c • S.coords x.1) : S.SameImage (ptOf x.1) (ptOf x'.1)

theorem **isPullback_baseChange** : IsPullback (S.bcFst E) (S.baseChange E).str S.str (spMap (algebraMap k E))

theorem **toBase_bijective** (Ω : Type) [Field Ω] [Algebra E Ω] [Algebra k Ω] [IsScalarTower k E Ω] : Function.Bijective (toBase : (S.baseChange E).Pt Ω → S.Pt Ω)

theorem **exists_coords_baseChange** {K : Type} [Field K] (x : Sp K ⟶ (S.baseChange E).Y) : ∃ c : K, c ≠ 0 ∧ (S.baseChange E).coords x = c • S.coords (x ≫ S.bcFst E)

theorem **IsVariety.baseChange** (h : S.IsVariety) : (S.baseChange E).IsVariety

theorem **BasePointFree.baseChange** (h : S.BasePointFree) : (S.baseChange E).BasePointFree

theorem **mem_subPts_iff** (I : S.Y.IdealSheafData) {Ω : Type} [Field Ω] [Algebra k Ω] (x : S.Pt Ω) : x ∈ S.subPts I Ω ↔ ptOf x.1 ∈ I.support

theorem **mem_subPts_cut_iff** {ι : Type} (s : ι → Fin S.N → k) {Ω : Type} [Field Ω] [Algebra k Ω] (x : S.Pt Ω) : x ∈ S.subPts (S.cut s) Ω ↔ ∀ i, ∑ j, algebraMap k Ω (s i j) * S.coords x.1 j = 0

theorem **mem_subPts_cut_baseChange_iff** (E : Type) [Field E] [Algebra k E] {ι : Type} (s : ι → Fin S.N → E) {Ω : Type} [Field Ω] [Algebra E Ω] [Algebra k Ω] [IsScalarTower k E Ω] (x : (S.baseChange E).Pt Ω) : x ∈ (S.baseChange E).subPts ((S.baseChange E).cut s) Ω ↔ ∀ i, ∑ j, algebraMap E Ω (s i j) * S.coords (toBase x).1 j = 0

theorem **IsFiniteSub.finite_subPts** {S : LinSys k} {I : S.Y.IdealSheafData} (h : S.IsFiniteSub I) (Ω : Type) [Field Ω] [Algebra k Ω] : (S.subPts I Ω).Finite

Development.AbelianVarietyTheory.Definitions.Divisors.MemUQptsStarCIff

theorem **Datum.actQ_tmul** (b : D.O) (q : ℚ) (x : D.La) : D.actQ b (q ⊗ₜ x) = q ⊗ₜ (b • x)

theorem **Datum.EFQ_tmul** (q q' : ℚ) (x y : D.La) : D.EFQ (q ⊗ₜ x) (q' ⊗ₜ y) = (D.EF x y : ℚ) * (q * q')

theorem **Datum.EFQ_swap** (x y : D.VQ) : D.EFQ y x = - D.EFQ x y

theorem **Datum.EFQ_actQ** (b : D.O) (x y : D.VQ) : D.EFQ (D.actQ b x) y = D.EFQ x (D.actQ (D.star b).unop y)

theorem **Datum.exists_tmul** (v : D.VQ) : ∃ (q : ℚ) (x : D.La), q ≠ 0 ∧ v = q ⊗ₜ x

theorem **Datum.eq_zero_of_EFQ** {v : D.VQ} (h : ∀ w, D.EFQ v w = 0) : v = 0

theorem **Datum.EFQ_nondeg** : D.EFQ.Nondegenerate

theorem **Datum.EFQ_adjE_left** (g : Module.End ℚ D.VQ) (x y : D.VQ) : D.EFQ (D.adjE g x) y = D.EFQ x (g y)

theorem **Datum.EFQ_adjE_right** (g : Module.End ℚ D.VQ) (x y : D.VQ) : D.EFQ (g x) y = D.EFQ x (D.adjE g y)

theorem **Datum.end_ext** {g h : Module.End ℚ D.VQ} (H : ∀ x y, D.EFQ (g x) y = D.EFQ (h x) y) : g = h

theorem **Datum.adjE_adjE** (g : Module.End ℚ D.VQ) : D.adjE (D.adjE g) = g

theorem **Datum.adjE_mul** (g h : Module.End ℚ D.VQ) : D.adjE (g * h) = D.adjE h * D.adjE g

theorem **Datum.adjE_add** (g h : Module.End ℚ D.VQ) : D.adjE (g + h) = D.adjE g + D.adjE h

theorem **Datum.adjE_smul** (q : ℚ) (g : Module.End ℚ D.VQ) : D.adjE (q • g) = q • D.adjE g

theorem **Datum.adjE_one** : D.adjE 1 = 1

theorem **Datum.adjE_zero** : D.adjE 0 = 0

theorem **Datum.adjE_actQ** (b : D.O) : D.adjE (D.actQ b) = D.actQ (D.star b).unop

theorem **Datum.mem_endO** {g : Module.End ℚ D.VQ} : g ∈ D.endO ↔ ∀ b, D.actQ b * g = g * D.actQ b

theorem **Datum.adjE_mem** {g : Module.End ℚ D.VQ} (hg : g ∈ D.endO) : D.adjE g ∈ D.endO

theorem **Datum.coe_starC** (g : D.endO) : (((D.starC g).unop : D.endO) : Module.End ℚ D.VQ) = D.adjE g

theorem **Datum.starC_starC** (g : D.endO) : (D.starC (D.starC g).unop).unop = g

theorem **Datum.coe_actInt** (c : 𝒪 D.F) : (D.actInt c : Module.End ℚ D.VQ) = D.actQ (algebraMap (𝒪 D.F) D.O c)

theorem **Datum.actFL_coe** (c : 𝒪 D.F) : D.actFL (c : D.F) = D.actInt c

theorem **Datum.actF_coe** (c : 𝒪 D.F) : D.actF (c : D.F) = D.actInt c

theorem **Datum.actF_comm** (x : D.F) (g : D.endO) : D.actF x * g = g * D.actF x

theorem **Datum.algebraMap_endO** (x : D.F) : algebraMap D.F D.endO x = D.actF x

theorem **Datum.starC_algebraMap** (x : D.F) : (D.starC (algebraMap D.F D.endO x)).unop = algebraMap D.F D.endO (IsCMField.complexConj D.F x)

theorem **Datum.mem_UQpts_starC_iff** (x : D.endOˣ × ℚˣ) : x ∈ GlobalTheory.LKTrip.UQpts D.starC ↔ ∀ v w : D.VQ, D.EFQ (((x.1 : D.endO) : Module.End ℚ D.VQ) v) (((x.1 : D.endO) : Module.End ℚ D.VQ) w) = (x.2 : ℚ) * D.EFQ v w

Development.AbelianVarietyTheory.Definitions.Divisors.MiddleNotMem

theorem **twoPhase_upU** (x₀ : Matrix n n' k) (e : StepVar n 2 n' → k) (x' : Matrix (n'' ⊕ Fin 1) (n ⊕ Fin 2) k) (β : n → k) : twoPhase ((colE 2 2 β).val * stepPt x₀ e) (x' * (colE 2 2 (-β)).val) = twoPhase (stepPt x₀ e) x'

theorem **twoPhase_lowU** (x : Matrix (n ⊕ Fin 2) (n' ⊕ Fin 3) k) (x₀ : Matrix n'' n k) (e : StepVar n'' 1 n → k) (u : n → k) : twoPhase ((rowE 2 1 u).val * x) (stepPt x₀ e * (rowE 2 1 (-u)).val) = twoPhase x (stepPt x₀ e)

theorem **heis_mem** (ρ : ℕ) (u β : n → k) : rowE (2 * ρ) ρ u ∈ symN k n (2 * ρ) ∧ colE (2 * ρ) (ρ + 1) β ∈ symN k n (2 * ρ)

theorem **middle_not_mem** (ρ : ℕ) (u : n → k) (hu : u ≠ 0) : rowE (2 * ρ + 1) (ρ + 1) u ∉ symN k n (2 * ρ + 1)

theorem **symPr_zero** (C : Subgroup (GL (n ⊕ Fin 0) k)) : symPr C = C.comap (embX 0)

theorem **elt_one** (J : G.Level) : (Reps.through hs J fun _ => 1).elt (fun _ => DoubleCoset.mk _ _ 1) = 1

theorem **moved_one** (keep : Bool) (T : G.Args → ℂ) : moved G keep 1 T = T

theorem **vacuum_term** (hψ : ∀ a ∈ 𝒪[k], ψ a = 1) (hν : IsUnitVol ν) (S : G.Args → ℂ) (h1 : ∀ z, (∀ b ∈ oppositePath G.m, ∀ i j, z b i j ∈ 𝒪[k]) → S z = 1) (h0 : ∀ z, ¬ (∀ b ∈ oppositePath G.m, ∀ i j, z b i j ∈ 𝒪[k]) → S z = 0) (J : G.Level) (keep : Bool) : (unfTerm ψ ν keep (Reps.through hs J fun _ => 1) S fun _ => DoubleCoset.mk _ _ 1).datum 0 0 = 1 ∧ (fun _ => DoubleCoset.mk _ _ 1) ∈ unfSupport ψ ν keep (Reps.through hs J fun _ => 1) S

theorem **unf_two** (hm : G.m = 2) (hψ : ∀ a ∈ 𝒪[k], ψ a = 1) (hν : IsUnitVol ν) (S : G.Args → ℂ) (h1 : ∀ z, (∀ b ∈ oppositePath G.m, ∀ i j, z b i j ∈ 𝒪[k]) → S z = 1) (h0 : ∀ z, ¬ (∀ b ∈ oppositePath G.m, ∀ i j, z b i j ∈ 𝒪[k]) → S z = 0) (J : G.Level) (keep : Bool) : ∃ (hfin : (unfSupport ψ ν keep (Reps.through hs J fun _ => 1) S).Finite) (t : UnfTerm G), unf ψ ν keep _ S hfin = {t} ∧ t.coef = 1 ∧ t.datum 0 0 = 1

Development.AbelianVarietyTheory.Definitions.Divisors.MinSt5OfMinSt4

theorem **MinAt5.isThetaCompact** {F : SFam g n (zetaAway n)} {Z : Scheme.{0}} {z : Z ⟶ Sp (cycAway n)} {i : F.X ⟶ Z} {OZ : LB Z} (h : MinAt5 F z i OZ) : IsThetaCompact F z i ∧ OZ.IsLB ∧ IsAmpleLB OZ

Development.AbelianVarietyTheory.Definitions.Divisors.MoveArgsOneTranspose

theorem **flipOrder_nodup** (m : ℕ) : (flipOrder m).Nodup ∧ ∀ i j, (i, j) ∈ flipOrder m ↔ i < m ∧ j < m

theorem **moveArgs_one_transpose** {k : Type*} [Field k] (Γ : Grid k) (g : Γ.VertexGroup) (x : Γ.Args) (b : Edge) : (Γ.moveArgs .one g x b)ᵀ = ((g b.src)⁻¹).val * (x b)ᵀ * (g b.tgt).val

theorem **stepFn_twist** (t : T) (w : Pl D.host.E) (h1 : w ∉ L.Pg) (h2 : w ∈ L.Q t) (j : ℕ) {g : GL (Fin D.R.r) (Kw w)} (hg : g ∈ KO (Kw w)) _) : stepFn L t w j (fromBlocks g.val 0 0 0) = ((L.xiH t).loc w (GeneralLinearGroup.det g) : ℂˣ)

theorem **hasTypes_twist** (G : GridData L sc) (t : T) (w : Pl D.host.E) (h1 : w ∉ L.Pg) (h2 : w ∈ L.Q t) (j : ℕ) (hξ : IsTame ((L.xiH t).loc w)⁻¹) (hμ : IsUnram ((muBase L (j + 1)).loc w)⁻¹) : HasTypes ((locGrid G t w).mu (j + 1))⁻¹ (parahoric _ _ _ hξ).toLevel

(parahoric _ _ _ hξ).toLevel (stepFn L t w j)

theorem **stepGroup_twist** (t : T) (w : Pl D.host.E) (h1 : w ∉ L.Pg) (h2 : w ∈ L.Q t) (j : ℕ) (hξ : IsTame ((L.xiH t).loc w)⁻¹) : stepGroup L t w j = (parahoric _ _ _ hξ).J

theorem **intM_of_stepFn** (t : T) (w : Pl D.host.E) (j : ℕ) {x} (h : stepFn L t w j x ≠ 0) : IntM x

theorem **givenStage_not_const** {G : GridData L sc} (t : T) (w : Pl D.host.E) (B : Frames G) (Φ : G.Datum) : (∃ x, (givenStage t w B Φ).tensor x = 1) ∧ ∃ x, (givenStage t w B Φ).tensor x = 0

theorem **flips_eq_zero** {k : Type*} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {Γ : Grid k} (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (l : List (ℕ × ℕ)) (S : Γ.PiLocalization₂) (i j : ℕ) (x : Γ.Args) (hx : x (.theta (i + 1) j) ∉ matLat _ (stdLat k _)) : (S.flips .one ψ ϖ e (l ++ [(i, j)])).tensor x = 0

theorem **Tflip_eq_zero** {G : GridData L sc} (W : G.Weil) (B : Frames G) (t : T) (Φ : G.Datum) (w : Pl D.host.E) {n : ℕ} (hn : D.R.m = n + 1) (x : (locGrid G t w).Args) (hx : x (.theta (n + 1) n) ∉ matLat _ (stdLat (Kw w) _)) : Tflip W B t Φ w x = 0

theorem **edgeFactor_theta** {k : Type*} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (Γ : Grid k) (hη : IsUnram Γ.eta) (g : Γ.VertexGroup) (hg : ∀ X, g X ∈ KO k _) (i j : ℕ) (Φ : Matrix (Fin (Γ.rank (i, j + 1))) (Fin (Γ.rank (i, j))) k → ℂ) (x : Γ.Args) : Γ.edgeFactor .one g (.theta i j) * Φ (Γ.moveArgs .one g x (.theta i j))ᵀ = om (Γ.mu (j + 1))⁻¹ (g (i, j)) (g (i, j + 1)) Φ (x (.theta i j))ᵀ

Development.AbelianVarietyTheory.Definitions.Divisors.MuGeOfMuLe

theorem **muGe_of_muLe** {K : Type} [Field K] [NumberField K] {m : ℕ} {h : HermFam K (Fin m → K)} {E : Submodule (𝒪 K) (Fin m → K)} {ν : ℝ} (h₁ : MuLe h E ν) (h₂ : deg h E = m * ν) (h₃ : ∀ V : Submodule K (Fin m → K), latRank K (E ⊓ V.restrictScalars (𝒪 K)) = Module.finrank K V) : MuGe h E ν

Development.AbelianVarietyTheory.Definitions.Divisors.MulSmulIdemMem

theorem **intOff_zero** {K : Type} [Field K] [NumberField K] (B : Finset ℕ) : IntOff B (0 : HeightOneSpectrum (𝒪 K) → ℝ)

theorem **IntOff.add** {K : Type} [Field K] [NumberField K] {B : Finset ℕ} {ρ ρ' : HeightOneSpectrum (𝒪 K) → ℝ} (h : IntOff B ρ) (h' : IntOff B ρ') : IntOff B (ρ + ρ')

theorem **mul_smul_idem_mem** {Z : Comp Xo c} {G : WTZ E} {f : Rul Xo c} (hf : f ∈ cuspOn U Z G) (a : c.F) : f * (a • idem Z) ∈ cuspOn U Z G

theorem **exZ_add_eq** (G G' : WTZ E) : exZ Dw (G + G') = exZ Dw G + exZ Dw G'

theorem **exZ_neg_eq** (G : WTZ E) : exZ Dw (-G) = -exZ Dw G

theorem **sig_mem_suppOn** (s : TwSec U x Z G₀) : s.sig ∈ suppOn Z

theorem **inv_mem_suppOn** (s : TwSec U x Z G₀) : s.inv ∈ suppOn Z

theorem **sig_ne_zero** (s : TwSec U x Z G₀) : s.sig ≠ 0

theorem **lo_le_hi** (s : TwSec U x Z G₀) (w : HeightOneSpectrum (𝒪 c.F)) : s.lo w ≤ s.hi w

theorem **cInf_eq** (s t : TwSec U x Z G₀) (e : s.sig = t.sig) : s.cInf = t.cInf

theorem **cInf_congr** (s : TwSec U x Z G₀) {σ σ' : c.F →+* ℂ} (h : goodOf c σ = goodOf c σ') : s.cInf σ = s.cInf σ'

theorem **homN_mul_sig** (s : TwSec U x Z G₀) {G : WTZ E} (hG : Even G.2) (f : Rul Xo c) {σ : c.F →+* ℂ} {z} (hz : z ∈ framesOn (U (goodOf c σ)) Z) : homN (U (goodOf c σ)) x (G + G₀) (f * s.sig) z = homN (U (goodOf c σ)) x G f z * s.cInf σ

theorem **exists_val** (s : TwSec U x Z G₀) {w} {v : Rul Xo c → EReal} (hv : v ∈ valOn Z w) : ∃ b : ℝ, v s.sig = b ∧ v s.inv = ((-b : ℝ) : EReal) ∧ s.lo w ≤ b ∧ b ≤ s.hi w

theorem **degLo_mul** (s : TwSec U x Z G₀) (t : TwSec U x Z G₁) : (s.mul t).degLo = s.degLo + t.degLo

theorem **degLo_ofConst** {la : Weight Dw} {δ : HeightOneSpectrum (𝒪 c.F) → ℝ} (s : ConstSec U x Z la δ) (hδ : (Function.support δ).Finite) (h₁ h₂) : (ofConst s hδ h₁ h₂).degLo = s.deg + ∑ᶠ w, δ w * lw c.F w

theorem **const_spec** (h : HasPts U Z) (a : c.F) (ha : a ≠ 0) : (const x h a ha).sig = a • idem Z ∧ (∀ σ, (const x h a ha).cInf σ = ‖σ a‖) ∧ (∀ w, (const x h a ha).lo w = vAdd w a) ∧ (∀ w, (const x h a ha).hi w = vAdd w a) ∧ (const x h a ha).degLo = 0

theorem **twSec_inhabited** : (1 ≤ Dw.p + Dw.q → (∀ p : ℕ, p.Prime → l.GoodAt p) → (∀ σ, GoodEmb c σ → IsUnf (U σ)) → ∀ (Z : Comp Xo c) (a : c.F), a ≠ 0 → ∃ s : TwSec U x Z 0, s.sig = a • idem Z ∧ (∀ σ, s.cInf σ = ‖σ a‖) ∧ (∀ w, s.lo w = vAdd w a) ∧ s.degLo = 0) ∧ ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q)) (Z : Comp Xo c), 1 ≤ Dw.p ∧ (∀ σ, GoodEmb c σ → IsUnf (U σ)) ∧ ∀ a : c.F, a ≠ 0 → ∃ s : TwSec U x Z 0, s.sig = a • idem Z ∧ (∀ σ, s.cInf σ = ‖σ a‖) ∧ (∀ w, s.lo w = vAdd w a) ∧ s.degLo = 0

theorem **badPrimes_prime** (p : ℕ) (hp : p ∈ badPrimes H l) : p.Prime

theorem **UnramPts.mono** {B B' : Finset ℕ} (h : UnramPts Xo c B) (hB : B ⊆ B') : UnramPts Xo c B'

theorem **mem_badPrimes** {p : ℕ} : p ∈ badPrimes H l ↔ p.Prime ∧ (p = 2 ∨ p = H.d ∨ p ∣ (discr E).natAbs ∨ p ∣ l.M ∨ p ∣ l.A)

theorem **UnramPts.of_index_one** {B : Finset ℕ} (h : ∀ (Z : Comp Xo c) (w : HeightOneSpectrum (𝒪 c.F)), (∀ p ∈ B, ((p : ℕ) : 𝒪 c.F) ∉ w.asIdeal) → ∃ (T : CDVR) (ι : c.F →+* T.Fr) (y : IntFr Xo c T ι), OverLa w T.V ι 1 ∧ algOn Z T ι y) : UnramPts Xo c B

Development.AbelianVarietyTheory.Definitions.Divisors.MulVecEqOfBasis

theorem **inUse** (p : PtAt ex Csz) (t : Tup p) : InUse p t

theorem **mu_eq** (p : PtAt ex Csz) : (p.S false).μ = p.R.mu ∧ p.P.A3.fam3.base3.μ0 = p.P.A3.A.μ0

theorem **toR_eq** (p : PtAt ex Csz) (t : Tup p) (x) : p.F.cr.toR (p.R.toCx3 p.F).zs false t x = eV0 p t (((p.P.A3.A.fam.R t).zE (p.P.A3.A.fam.lvl p.P.P0 p.P.ψ0) ((p.R.toCx3 p.F).zs.Z t)).mulVec x)

theorem **mem_goodPart_iff** (p : PtAt ex Csz) (t : Tup p) (W : Submodule D.what₂.LL (p.P.A3.A.fam.R t).S.Forms) (x) : x ∈ (p.P.A3.A.fam.R t).inV0 (p.P.A3.A.fam.lvl p.P.P0 p.P.ψ0) W ↔ eV0 p t x ∈ W

theorem **hd_eq** (ij : Sq D) : sqRank D ij / 2 = SqNum.halfRank D ij.1 ij.2

theorem **isIntExp_mono** {T : Type} {Lc : LevCh D T} {sc : GlobalTheory.ChiGrid.Consts} {G : GridData Lc sc} {C C' : ℝ} {dl} (hb : 0 ≤ Real.log sc.b) (h : SqNum.IsIntExp G C dl) (hC : C ≤ C') : SqNum.IsIntExp G C' dl

theorem **mulVec_eq_of_basis** {G : Grid p CF} {W : G.Weil} {n : ℕ} (a : SlopeTestRemainingChoices p G W n) (t : Tup p) (hS : ∀ i, a.ch.S.SumOK t (a.ch.Phi i)) (V : Matrix (Fin n) (Fin ((Df p).nV t)) D.what₂.LL) (h : ∀ i k, D.what₂.embLL (V i k) = vK a.ch t (eV0 p t

(Pi.single k 1)) i) (x) (i : Fin n) : D.what₂.embLL (V.mulVec x i) = vK a.ch t (eV0 p t x) i

theorem **entry** {d : FunIn p CF} {a : SlopeTestRemainingChoices p d.G d.W d.n} (h : TieV d a) (t : Tup p) (i k) : D.what₂.embLL (d.V t i k) = vK a.ch t (eV0 p t (Pi.single k 1)) i

theorem **ratio_eq** {d : FunIn p CF} {a : SlopeTestRemainingChoices p d.G d.W d.n} (h : TieAt d a) (t : Tup p) : cTot a.ch p.P.A3.A.μ0 ≠ 0 ∧ D.what₂.embLL (∏ ij, d.rq t ij) = ratio a.ch t

theorem **exists_tie** {G : Grid p CF} {W : G.Weil} (hW : W.Over p.R.toRealDat) {n : ℕ} (a : SlopeTestRemainingChoices p G W n) (ha : a.OK) (hS : ∀ t i, a.ch.S.SumOK t (a.ch.Phi i)) (hR : ∀ t x i, vK a.ch t x i ∈ Set.range D.what₂.embLL) (hc : ∀ i j t, SquareNonzero W a.S.B a.rm t i j ∧ a.c i j t ≠ 0) (hr : ∀ i j t, sqRatio a.ch i j t ∈ Set.range D.what₂.embLL) (hB : BottomsWith G p.P.A3.A.μ0 fun _ => True) (h3 : ∀ (ij : Sq D) (s : ℕ), 1 ≤ s → s ≤ SqNum.halfRank D ij.1 ij.2 → ∃ zd g n, SqNum.IsTorVal G p.P.A3.A.μ0 ij.1 ij.2 s zd g ∧ SqNum.IsDenom G ij.1 ij.2 s n ∧ {l | l.Prime ∧ n l ≠ 0}.Finite) (h4 : ∃ dl, SqNum.IsIntExp G CF dl) : ∃ d : FunIn p CF, d.G = G ∧ Comparison₂ p CF d

theorem **pos_of_tie** (d : FunIn p CF) (h : Comparison₂ p CF d) : 0 < d.n

theorem **matrix_unique** {G : Grid p CF} {W : G.Weil} {n : ℕ} (a : SlopeTestRemainingChoices p G W n) (t : Tup p) (V V' : Matrix (Fin n) (Fin ((Df p).nV t)) D.what₂.LL) (h : ∀ x i, D.what₂.embLL (V.mulVec x i) = vK a.ch t (eV0 p t x) i) (h' : ∀ x i, D.what₂.embLL (V'.mulVec x i) = vK a.ch t (eV0 p t x) i) : V = V'

theorem **ofGrid_tie** {G : Grid p CF} {W : G.Weil} {hW n V} (h : Comparison₂ p CF (.ofGrid G W hW n V)) : (∀ w : PlL D, ((D.host.pde : ℕ) : 𝒪 D.what₂.LL) ∈ w.asIdeal) ∧ ∃ (S : Finset ℕ) (del : ℕ → ℝ), 1 ≤ del 2 ∧ Formalism.PairingDivisionLocalBalance.wt (K

Development.AbelianVarietyTheory.Definitions.Divisors.NAInterSupSSubset

theorem **mem_NA_iff** (u : GA E a) : u ∈ NA E a ↔ ∃ X : Pos a → Ad E, u = uGL X

theorem **superDiag_uGL** (X : Pos a → Ad E) : GlobalTheory.Whittaker.superDiag (uGL X) = supSum 0 X

theorem **whit_mul** {F : GA E a → ℂ} (hc : Continuous F) (hl : AutomorphicForm.IsLeftInvariant F) {u : GA E a} (hu : u ∈ NA E a) (g : GA E a) : whit F (u * g) = psiN u * whit F g

theorem **whit_rank_one** {F : GA E 1 → ℂ} (hc : Continuous F) (g : GA E 1) : whit F g = F g

theorem **exists_psiN_ne_one** : ∃ u ∈ NA E 2, psiN u ≠ 1

theorem **whit_const_two** : whit (fun _ : GA E 2 => (1 : ℂ)) 1 = 0

theorem **psiLoc_eq** (w : HeightOneSpectrum (𝒪 E)) (x : w.adicCompletion E) : psiLoc w x = (GroundTheory.Global.localAdelicChar E w x : ℂ)

theorem **diagPow_val** {K : Type} [CommRing K] {n : ℕ} (ϖ : Kˣ) (l : Fin n → ℕ) : (diagPow ϖ l : Matrix (Fin n) (Fin n) K) = Matrix.diagonal fun i => (ϖ : K) ^ l i

theorem **diagPow_zero** {K : Type} [CommRing K] {n : ℕ} (ϖ : Kˣ) : diagPow ϖ (0 : Fin n → ℕ) = 1

theorem **tl_comp** (ϖ : ∀ w : HeightOneSpectrum (𝒪 E), (w.adicCompletion E)ˣ) (l : HeightOneSpectrum (𝒪 E) →₀ (Fin a → ℕ)) (w : HeightOneSpectrum (𝒪 E)) : GeneralLinearGroup.map ((pLoc E w).comp (pFin E)) (tl ϖ l) = diagPow (ϖ w) (l w)

theorem **loc_mem_supS** (S : Finset (HeightOneSpectrum (𝒪 E))) (w : HeightOneSpectrum (𝒪 E)) (g : GL (Fin a) (w.adicCompletion E)) : finGL E a (locGL E a w g) ∈ supS E a S ↔ w ∈ S ∨ g = 1

theorem **NA_inter_supS_subset** (S : Finset (HeightOneSpectrum (𝒪 E))) (m : ℕ) : NA E (m + 1) ∩ supS E (m + 1) S ⊆ mirS E S m

theorem **isRapidInf_dirac** (c : ℂ) (y : GL (Fin a) (Cinf E)) : IsRapidInf (Measure.dirac ()) (fun _ => c) (fun _ : Unit => y)

theorem **not_isRapidInf_const** : ¬ IsRapidInf (volume : Measure ℝ) (fun _ => 1) (fun _ => (1 : GL (Fin a) (Cinf E)))

theorem **inf_mem_supS** (S : Finset (HeightOneSpectrum (𝒪 E))) (y : GL (Fin a) (Cinf E)) : infGL E a y ∈ supS E a S

theorem **mspanR_triv** (ha : a ≤ 1) : mspanR (CuspSub.triv E ha) = Submodule.span ℂ {fun _ => 1}

Development.AbelianVarietyTheory.Definitions.Divisors.NatCastNeZeroG

theorem **RelFlag.ne_zero** {C : CxSt F} {N : ℕ} {A A' : Fin N → Adl C Y} {p n k : ℕ} (h : RelFlag A A' p n k) : n ≠ 0

theorem **RelFlag.mono** {C : CxSt F} {N : ℕ} {A A' : Fin N → Adl C Y} {p n k k' : ℕ} (h : RelFlag A A' p n k) (hk : k' ≤ k) : RelFlag A A' p n k'

theorem **valAt_natCast** {K : Type} [Field K] {Z : Scheme.{0}} [IsIntegral Z] (y : Sp K ⟶ Z) (m : ℕ) : TrueGeo.valAt y (m : Z.functionField) = m

theorem **regAt_natCast** {Z : Scheme.{0}} [IsIntegral Z] (x : Z) (m : ℕ) : RegAt x (m : Z.functionField)

theorem **isUnit_natCast** (V : Cover Y) (z : V.G) {m : ℕ} (hm : m ≠ 0) : IsUnit (m : V.G.presheaf.stalk z)

theorem **natCast_ne_zero_X** (V : Cover Y) {m : ℕ} (hm : m ≠ 0) : (m : V.X.functionField) ≠ 0

theorem **eigAt_of_lt** (a : ℕ → Bool) (R : Option ℝ) (s : ℤ) {k : ℕ} (h : k < (letters Dw).length) : eigAt Dw a R s k = Finsupp.single (a, R, .inl ((letters Dw)[k].1, ((letters Dw)[k].2 : ℕ))) s

@[simp] theorem **eigSym_inl** (a : ℕ → Bool) (R : Option ℝ) {g : ℕ} (k : Fin g) : eigSym Dw a R (.inl k) = eigAt Dw a R 1 k

@[simp] theorem **eigSym_inr** (a : ℕ → Bool) (R : Option ℝ) {g : ℕ} (k : Fin g) : eigSym Dw a R (.inr k) = eigAt Dw a R (-1) (Fin.rev k)

theorem **expo_eigSym** (a : ℕ → Bool) (R : Option ℝ) {g : ℕ} (i : Fin g ⊕ Fin g) : expo Dw (eigSym Dw a R i) = expo Dw (wholeSym a R i)

Development.AbelianVarietyTheory.Definitions.Divisors.NatCastSmulMemHatL

theorem **natCast_smul_mem_hatL** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) {𝔪 : Ideal (𝒪 E)} {N : ℕ} (hN : (N : 𝒪 E) ∈ 𝔪) {z : Fin n → E ⊗[F] Formalism.L2.Herm.Af F} (hz : z ∈ Formalism.L2.Herm.hatL (F

theorem **eventually_cong_mat** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (β : GL (Fin n) (E ⊗[F] Af F)) {N : ℕ} (hN : N ≠ 0) {X : Type} [TopologicalSpace X] {x₀ : X} {φ : X → GL (Fin n) (Af F ⊗[F] E)} (hf : Continuous φ) (h₀ : φ x₀ = 1) : ∀ᶠ κ in 𝒩 x₀, ∃ z : Matrix (Fin n) (Fin n) (E ⊗[F] Af F), (∀ i j, z i j ∈ Formalism.L2.Herm.intAf F E) ∧ (β⁻¹ * Formalism.UG.bc (GlobalTheory.Mass.swapTens F E (Af F) : _ →+* _) (φ κ) * β).1 - 1 = N • z

theorem **mem_congLev_of_cong** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] {n : ℕ} {hc : CMData F E} {J : Matrix (Fin n) (Fin n) E} {Λ : Submodule (𝒪 E) (Fin n → E)} {𝔪 : Ideal (𝒪 E)} {N : ℕ} (hN : (N : 𝒪 E) ∈ 𝔪) {β x : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} (hβ : IsFrame β Λ) (h1 : x ∈ UA hc J) (h2 : x ∈ St β) {z : Matrix (Fin n) (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} (hz : ∀ i j, z i j ∈ Formalism.L2.Herm.intAf F E) (e : (β⁻¹ * x * β).1 - 1 = N • z) : x ∈ congLev hc J Λ 𝔪

theorem **congLev_open_compact** (V : HSp D) {Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)} (hΛ : Submodule.IsLattice D.host.E Λ) {𝔪 : Ideal (𝒪 D.host.E)} (h𝔪 : 𝔪 ≠ ⊥) : IsOpen (((congLev D.sp.cm V.gram Λ 𝔪).comap (finE V) : Subgroup (GA D V)) : Set (GA D V)) ∧ IsCompact ((levA V (congLev D.sp.cm V.gram Λ 𝔪) : Subgroup (GA D V)) : Set (GA D V))

theorem **exists_meas** (V : HSp D) (hV : V.SigOK) {q : ℕ} (hq : ∀ τ, (V.sig τ).2 = q) (K : Subgroup (GL (Fin V.rank) (EAf D))) (hK : IsOpen ((K.comap (finE V) : Subgroup (GA D V)) : Set (GA D V)) ∧ IsCompact ((levA V K : Subgroup (GA D V)) : Set (GA D V))) : ∃ μ 𝓕, if q = 0 then IsMeasDef V K μ 𝓕 else IsMeasFD V K μ 𝓕

Development.AbelianVarietyTheory.Definitions.Divisors.NefFibOfIsNef

theorem **nefFib_of_isNef** {L : HermLB p} {D : ℕ} (h : L.IsNef D) : NefFib L

Development.AbelianVarietyTheory.Definitions.Divisors.NoCommonZeroCombVal

theorem **mem_pick** (hB : B.IsLB) (y : CxPts p σ) : y ∈ ptsIn p σ (B.U (pick hB y))

theorem **ev_one** (h : y ∈ ptsIn p σ U) : ev p σ U 1 y = 1

theorem **ev_inv** (h : y ∈ ptsIn p σ U) (u : Γ(Y, U)ˣ) : ev p σ U (u⁻¹).val y = (ev p σ U u.val y)⁻¹

theorem **gv_ne_zero** {i j : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) : gv p σ B i j y ≠ 0

private theorem **ev_res_units_val_eq** {U₁ W₁ : Y.Opens} (e : W₁ ≤ U₁) (h : y ∈ ptsIn p σ W₁) (u : Γ(Y, U₁)ˣ) : ev p σ W₁ (Semi.res e u).val y = ev p σ U₁ u.val y

theorem **gv_self** (hB : B.IsLB) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : gv p σ B i i y = 1

theorem **gv_mul** (hB : B.IsLB) {i j l : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) (hl : y ∈ ptsIn p σ (B.U l)) : gv p σ B i j y * gv p σ B j l y = gv p σ B i l y

theorem **gv_tens** {c c' : (tens A B).ι} (h : y ∈ ptsIn p σ ((tens A B).U c)) (h' : y ∈ ptsIn p σ ((tens A B).U c')) : gv p σ (tens A B) c c' y = gv p σ A c.1 c'.1 y * gv p σ B c.2 c'.2 y

theorem **gv_dual** {i j : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) (hj : y ∈ ptsIn p σ (B.U j)) : gv p σ (dual B) i j y = (gv p σ B i j y)⁻¹

theorem **ev_iso_rel** {w : LB.Iso A B} (hw : w.IsGen) {a a' : A.ι} {b b' : B.ι} (ha : y ∈ ptsIn p σ (A.U a)) (ha' : y ∈ ptsIn p σ (A.U a')) (hb : y ∈ ptsIn p σ (B.U b)) (hb' : y ∈ ptsIn p σ (B.U b')) : ev p σ _ (w.s (b, a)).val y * gv p σ A a a' y = gv p σ B b b' y * ev p σ _ (w.s (b', a')).val y

theorem **ev_iso_ne_zero** (w : LB.Iso A B) {a : A.ι} {b : B.ι} (ha : y ∈ ptsIn p σ (A.U a)) (hb : y ∈ ptsIn p σ (B.U b)) : ev p σ _ (w.s (b, a)).val y ≠ 0

theorem **mem_of_mem_linComb** (hm : ∀ a, (m a).1 = z.1 ≫ f a) {ε : κ → ℤ} {j : κ → B.ι} (h : z ∈ ptsIn p' σ' ((LB.LinComb B f ε).U j)) (a : κ) : m a ∈ ptsIn p σ (B.U (j a))

private theorem **ev_units_val_eq_evHom** {V₁ : Y.Opens} {y₁ : CxPts p σ} (h : y₁ ∈ ptsIn p σ V₁) (u : Γ(Y, V₁)ˣ) : ev p σ V₁ u.val y₁ = ((Units.map (evHom h).toMonoidHom u : ℂˣ) : ℂ)

private theorem **pullU_val'** {Y₁ : Scheme.{0}} (g : Y₁ ⟶ Y) (W₁ : Y.Opens) (u : Γ(Y, W₁)ˣ) : (pullU g W₁ u).val = (g.app W₁).hom u.val

private theorem **ev_app_pullback_eq_ev** {k₁ : Type} [CommRing k₁] {Y₁ : Scheme.{0}} {p₁ : Y₁ ⟶ Sp k₁} {σ₁ : k₁ →+* ℂ} (g : Y₁ ⟶ Y) {W₁ : Y.Opens} {z₁ : CxPts p₁ σ₁} {y₁ : CxPts p σ} (hm : y₁.1 = z₁.1 ≫ g) (a : Γ(Y, W₁)) : ev p₁ σ₁ (g ⁻¹ᵁ W₁) ((g.app W₁).hom a) z₁ = ev p σ W₁ a y₁

theorem **gv_linComb** (hm : ∀ a, (m a).1 = z.1 ≫ f a) {ε : κ → ℤ} {j j' : κ → B.ι} (h : z ∈ ptsIn p' σ' ((LB.LinComb B f ε).U j)) (h' : z ∈ ptsIn p' σ' ((LB.LinComb B f ε).U j')) : gv p' σ' (LB.LinComb B f ε) j j' z = ∏ a, gv p σ B (j a) (j' a) (m a) ^ ε a

theorem **mem_pow_iff** {n : ℤ} {j : PUnit → B.ι} : y ∈ ptsIn p σ ((LB.LinComb B (fun _ => 𝟙 Y) fun _ => n).U j) ↔ y ∈ ptsIn p σ (B.U (j PUnit.unit))

theorem **gv_pow** {n : ℤ} {j j' : PUnit → B.ι} (h : y ∈ ptsIn p σ (B.U (j PUnit.unit))) (h' : y ∈ ptsIn p σ (B.U (j' PUnit.unit))) : gv p σ (LB.LinComb B (fun _ => 𝟙 Y) fun _ => n) j j' y = gv p σ B (j PUnit.unit) (j' PUnit.unit) y ^ n

theorem **rv_ne_zero** (hη : IsFrame η) (hη' : IsFrame η') (h : y ∈ ptsIn p σ U) (h' : y ∈ ptsIn p σ U') : rv σ hη hη' y ≠ 0

theorem **rv_mul** (hη : IsFrame η) (hη' : IsFrame η') (hη'' : IsFrame η'') (h : y ∈ ptsIn p σ U) (h' : y ∈ ptsIn p σ U') (h'' : y ∈ ptsIn p σ U'') : rv σ hη hη' y * rv σ hη' hη'' y = rv σ hη hη'' y

theorem **gv_canon** [SmoothOfRelativeDimension D p] (K : CanonLB p D) (a b : K.bundle.ι) : gv p σ K.bundle a b = rv σ (K.isFrame a) (K.isFrame b)

theorem **rv_chart** (c c' : AlgChart p σ D) (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : rv σ c.isFrame c'.isFrame y = transDet c.e c'.e y

theorem **secVal_apply** (s : B.Sec) (i : B.ι) : secVal p σ s i = ev p σ (B.U i) (s.f i)

theorem **secVal_mem** (s : B.Sec) : secVal p σ s ∈ fibSec p σ B

theorem **secVal_smul** (a : k) (s : B.Sec) : secVal p σ (algX p ⊤ a • s) = σ a • secVal p σ s

theorem **secVal_ne_zero_iff** (s : B.Sec) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : secVal p σ s i y ≠ 0 ↔ y ∈ ptsIn p σ (nonzeroLocus s)

theorem **secTens_mem** {s : A.ι → CxPts p σ → ℂ} {t : B.ι → CxPts p σ → ℂ} (hs : s ∈ fibSec p σ A) (ht : t ∈ fibSec p σ B) : secTens s t ∈ fibSec p σ (tens A B)

theorem **of_smul** (a : k) (s : B.Sec) : of p σ (algX p ⊤ a • s) = σ a • of p σ s

theorem **span_range_of** : Submodule.span ℂ (range (of p σ : B.Sec → _)) = ⊤

@[simp] theorem **lift_of** {E : Type*} [AddCommGroup E] [Module ℂ E] (φ : B.Sec →+ E) (h : ∀ a s, φ (algX p ⊤ a • s) = σ a • φ s) (s : B.Sec) : lift φ h (of p σ s) = φ s

@[simp] theorem **val_of** (s : B.Sec) : val (of p σ s) = secVal p σ s

theorem **val_mem_of** {M : Submodule ℂ (B.ι → CxPts p σ → ℂ)} (h : ∀ s, secVal p σ s ∈ M) (x : SecC p σ B) : val x ∈ M

theorem **val_mem** (x : SecC p σ B) : val x ∈ fibSec p σ B

theorem **log_normSq_mul** {a b : ℂ} (ha : a ≠ 0) (hb : b ≠ 0) : Real.log (‖a * b‖ ^ 2) = Real.log (‖a‖ ^ 2) + Real.log (‖b‖ ^ 2)

theorem **log_normSq_zpow** (a : ℂ) (n : ℤ) : Real.log (‖a ^ n‖ ^ 2) = n * Real.log (‖a‖ ^ 2)

theorem **normSq_mul_exp** {u : ℂ} (hu : u ≠ 0) (s : ℂ) (φ : ℝ) : ‖u * s‖ ^ 2 * Real.exp (-(φ + Real.log (‖u‖ ^ 2))) = ‖s‖ ^ 2 * Real.exp (-φ)

theorem **nsq_eq** (hs : s ∈ fibSec p σ B) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : φ.nsq s y = ‖s i y‖ ^ 2 * Real.exp (-φ.φ i y)

theorem **nsq_nonneg** (s : B.ι → CxPts p σ → ℂ) (y : CxPts p σ) : 0 ≤ φ.nsq s y

theorem **nsq_addFn** (u : CxPts p σ → ℝ) (s : B.ι → CxPts p σ → ℂ) (y : CxPts p σ) : (φ.addFn u).nsq s y = φ.nsq s y * Real.exp (-u y)

theorem **nsq_tens** (ψ : Weight p σ A) {t : A.ι → CxPts p σ → ℂ} (ht : t ∈ fibSec p σ A) (hs : s ∈ fibSec p σ B) (y : CxPts p σ) : (ψ.tens φ).nsq (secTens t s) y = ψ.nsq t y * φ.nsq s y

theorem **zpow_φ** (n : ℤ) (j : PUnit → B.ι) (y : CxPts p σ) : (φ.zpow n).φ j y = n * φ.φ (j PUnit.unit) y

theorem **pull_φ** {Y' : Scheme.{0}} {p' : Y' ⟶ Sp k} (f : Y' ⟶ Y) (hf : f ≫ p = p') (j : PUnit → B.ι) (z : CxPts p' σ) : (φ.pull f hf).φ j z = φ.φ (j PUnit.unit) (PtsOver.map f hf _ z)

theorem **ofPairs_φ** {hA : A.IsLB} {hB : B.IsLB} {ψ : A.ι → B.ι → CxPts p σ → ℝ} {h} {a : A.ι} {b : B.ι} (ha : y ∈ ptsIn p σ (A.U a)) (hb : y ∈ ptsIn p σ (B.U b)) : (ofPairs hA hB ψ h).φ a y = ψ a b y

theorem **NoCommonZero.sum_pos** (h : NoCommonZero e) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : 0 < ∑ a, ‖e a i y‖ ^ 2

theorem **nsq_fs** {hB : B.IsLB} {he : ∀ a, e a ∈ fibSec p σ B} {h : NoCommonZero e} (hs : s ∈ fibSec p σ B) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : (fs hB e he h).nsq s y = ‖s i y‖ ^ 2 / ∑ a, ‖e a i y‖ ^ 2

theorem **nsq_fs_le** {hB : B.IsLB} {he : ∀ a, e a ∈ fibSec p σ B} {h : NoCommonZero e} (c : α → ℂ) (y : CxPts p σ) : (fs hB e he h).nsq (∑ a, c a • e a) y ≤ ∑ a, ‖c a‖ ^ 2

theorem **sum_nsq_fs** {hB : B.IsLB} {he : ∀ a, e a ∈ fibSec p σ B} {h : NoCommonZero e} (y : CxPts p σ) : ∑ a, (fs hB e he h).nsq (e a) y = 1

theorem **combVal_eq** (b : α → B.Sec) (M : Matrix α α ℂ) (a : α) : combVal p σ b M a = SecC.val (∑ j, M j a • SecC.of p σ (b j))

theorem **noCommonZero_combVal** {b : α → B.Sec} (hb : ⊔ j, nonzeroLocus (b j) = ⊤) {M : Matrix α α ℂ} (hM : M.det ≠ 0) : NoCommonZero (combVal p σ b M)

theorem **fsBasis_φ** (hB : B.IsLB) (b : α → B.Sec) (hb : ⊔ j, nonzeroLocus (b j) = ⊤) (M : Matrix α α ℂ) (hM : M.det ≠ 0) (i : B.ι) (y : CxPts p σ) : (fsBasis p σ hB b hb M hM).φ i y = Real.log (∑ a, ‖∑ j, M j a * valR y.1 (B.U i) ((b j).f i)‖ ^ 2)

Development.AbelianVarietyTheory.Definitions.Divisors.NonemptyCuspFrame

theorem **IsKS.val_eq** {ω ω' : H0 M.str (LB.div M.K.bundle ((L₀ X Z (FlagTan.kap Dw)).pull M.σ)) M.off} (h : IsKS M ω) (h' : IsKS M ω') {K : Type} [Field K] [Algebra c.F K] {P Q : FrPt Xo c K} {t : Ix Dw → K×} {v : TIx Dw (psiSet H) → FrPt Xo c (DualNumber K)} {z : Sp K ⟶ M.X} {zv : TIx Dw (psiSet H) → (Sp (DualNumber K) ⟶ M.X)} {i : M.K.bundle.ι} {k : X.F.J} {O : M.X.Opens} (hO : O ≤ M.K.bundle.U i) {g : Fin (dimFlag Dw) → Γ(M.X, O)} (hz : ⊤ ≤ z ⁻¹ᵁ O) {hzv : ∀ b, ⊤ ≤ zv b ⁻¹ᵁ O} {hk : ⊤ ≤ (z ≫ toFlag M) ⁻¹ᵁ X.F.U k} (ho : ⊤ ≤ z ⁻¹ᵁ M.off) (hi) (hQ : X.F.ChartFr k (z ≫ toFlag M) hk Q) (ht : FrBor t Q P) (hv : IsTanFrame (psiSet H) P v) (hl : ∀ b, zv b ≫ toFlag M = X.ptY (v b)) (hd : (Matrix.of fun a b => (evalPt (zv (tanOrd Dw b)) O (hzv _) (g a)).snd).det ≠ 0) : H0.val (B

theorem **adjSec_injective** (h₁ : FreeOnOpen X Z U P.u₁) (j : ℕ) : Function.Injective (adjSec R ω h₁ j)

theorem **mulPole_adjSec** (h₁ : FreeOnOpen X Z U P.u₁) (j : ℕ) (f : Cusp Z U (P.wt j)) : R.mulPole j R.off (adjSec R ω h₁ j f) = R.mulCan ω.ω (P.kap_add j) (R.pullOff (cuspSec X Z U _ f))

theorem **nonempty_cuspFrame** {la : AllWt.Weight Dw} (h : GoodOn U x V dict Z (wtOf Dw la 0)) : Nonempty (CuspFrame Z U x V dict la)

theorem **data_numbers** : (h.data f₁ f₂ th k).D = dimFlag Dw ∧ (h.data f₁ f₂ th k).T = P.T ∧ (h.data f₁ f₂ th k).j₀ = 1 ∧ (h.data f₁ f₂ th k).n = f₁.n ∧ (h.data f₁ f₂ th k).abelian₂ = k.abelian₂ ∧ (h.data f₁ f₂ th k).beta = k.beta ∧ (h.data f₁ f₂ th k).LamD = k.LamD ∧ (h.data f₁ f₂ th k).LamT = k.LamT ∧ (h.data f₁ f₂ th k).J = k.J

theorem **coe_equivV** {j : ℕ} (hj : j ≤ P.T) (f : Cusp Z U (P.wt j)) : (h.equivV f₁ f₂ th k hj f).1 = adjSec R ω h.free₁ j f

theorem **equivV_mem** {j : ℕ} (hj : j ≤ P.T) (f : Cusp Z U (P.wt j)) : h.equivV f₁ f₂ th k hj f ∈ (h.data f₁ f₂ th k).M j ↔ f ∈ latOn U Z (wtOf Dw (P.wt j) 0)

theorem **nrm_equivV** {j : ℕ} (hj : j ≤ P.T) (σ : c.F →+* ℂ) (f : Cusp Z U (P.wt j)) : ((h.data f₁ f₂ th k).nm j).nrm σ (h.equivV f₁ f₂ th k hj f) = petOn U x V dict Z (wtOf Dw (P.wt j) 0) σ f

theorem **ofSec_comb** (w : Cusp Z U P.u₁) : H0.ofSec R.str ⊤ ((h.data f₁ f₂ th k).comb (f₁.b.equivFun w)) = R.Δ₁.lift (h.ext₁.equiv w)

theorem **mem_lattice** (w : Cusp Z U P.u₁) : f₁.b.equivFun w ∈ (h.data f₁ f₂ th k).lattice ↔ w ∈ latOn U Z (wtOf Dw P.u₁ 0)

theorem **normAt_eq** (σ : c.F →+* ℂ) (w : Cusp Z U P.u₁) : (h.data f₁ f₂ th k).normAt σ (f₁.b.equivFun w) = petOn U x V dict Z (wtOf Dw P.u₁ 0) σ w

theorem **ofSec_comb₂** (w : Cusp Z U P.u₂) : H0.ofSec R.str ⊤ ((h.data f₁ f₂ th k).second.comb (f₂.b.equivFun w)) = R.Δ₂.lift (h.ext₂.equiv w)

theorem **mem_lattice₂** (w : Cusp Z U P.u₂) : f₂.b.equivFun w ∈ (h.data f₁ f₂ th k).second.lattice ↔ w ∈ latOn U Z (wtOf Dw P.u₂ 0)

theorem **normAt_eq₂** (σ : c.F →+* ℂ) (w : Cusp Z U P.u₂) : (h.data f₁ f₂ th k).second.normAt σ (f₂.b.equivFun w) = petOn U x V dict Z (wtOf Dw P.u₂ 0) σ w

theorem **d_eq** : (h.data f₁ f₂ th k).d = Module.finrank c.F (Cusp Z U (P.wt P.T))

theorem **mulW_adjSec** (j : ℕ) (w : Cusp Z U P.u₁) (f : Cusp Z U (P.wt j)) (g : Cusp Z U (P.wt (j + 1))) (hg : g.1 = w.1 * f.1) : (h.data f₁ f₂ th k).mulW j (f₁.b.equivFun w) (adjSec R ω h.free₁ j f) = adjSec R ω h.free₁ (j + 1) g

theorem **log_be** (N : OT.Num) : Real.log (be N) = N.P7 + 2 * (N.D : ℝ) * Real.log ((N.D : ℝ) * N.ka) + N.V

Development.AbelianVarietyTheory.Definitions.Divisors.NonemptyGridDataIff

theorem **exists_sum** (X : LSp D) {ν q : ℕ} (hX : X.IsV ν q) (hq : q ≤ ν) (c : ℕ) (neg : Bool) (b : Fin c → FractionalIdeal (𝒪 D.host.E)⁰ D.host.E) (h0 : ∀ p, b p ≠ 0) (hd : ∀ p x, x ∈ b p.rev ↔ star x ∈ (b p)⁻¹) {ν' q' : ℕ} (h1 : ν + c = ν') (h2 : q + (if neg then (c + 1) / 2 else c / 2) = q') : ∃ Y : LSp D, Y.IsV ν' q' ∧ IsSum X Y c (if neg then -1 else 1) b

theorem **exists_line** (X : LSp D) {ν q : ℕ} (hX : X.IsV ν q) (hq : q ≤ ν) : ∃ Y : LSp D, Y.IsV (ν + 1) q ∧ IsSum X Y 1 1 fun _ => 1

theorem **exists_stair** (X : LSp D) {ν q : ℕ} (hX : X.IsV ν q) (hq : q ≤ ν) (c : ℕ) {a a' : FractionalIdeal (𝒪 D.host.E)⁰ D.host.E} (h0 : a ≠ 0) (hc : ∀ x, x ∈ a' ↔ star x ∈ a) {ν' q' : ℕ} (h1 : ν + c = ν') (h2 : q + (c + 1) / 2 = q') : ∃ Y : LSp D, Y.IsV ν' q' ∧ IsSum X Y c (-1) (stairId a a' c)

theorem **nonempty_vertices** : ∀ (D : D7) (T : Type) (L : LevCh D T) (a a' : FractionalIdeal (𝒪 D.host.E)⁰ D.host.E), IsSelfDualC (Yc D 0) (L.ΛY 0) → a ≠ 0 → (∀ x, x ∈ a' ↔ star x ∈ a) → Nonempty (Vertices L a a')

theorem **exists_gridData** {T : Type} (L : LevCh D T) (sc : Consts) (P : Places L) (h0 : P.v0 ∉ S1 D) (hY : IsSelfDualC (Yc D 0) (L.ΛY 0)) : ∃ G : GridData L sc, G.pl = P

theorem **nonempty_gridData_iff** {T : Type} (L : LevCh D T) (μ0 : T) (hL : L.Is μ0) (sc : Consts) : Nonempty (GridData L sc) ↔ ∃ P : Places L, P.v0 ∉ S1 D

theorem **isV_all** {T : Type} {L : LevCh D T} {μ0 : T} (hL : L.Is μ0) {sc : Consts} (G : GridData L sc) {X : Node} (h : X.InRange D.R.m) : (G.node X).IsV (X.rank D) (X.neg D.R.m)

theorem **chiH_ne_one** {T : Type} {L : LevCh D T} {P : Places L} {C : ℝ} (X : Chars P C) {i : ℕ} (h : 1 ≤ i) (h' : i ≤ D.R.m) : X.chiH i ≠ 1

theorem **modul_ne_top** {T : Type} {L : LevCh D T} (P : Places L) (sc : Consts) : P.modul sc ≠ ⊤

theorem **nonempty_gridData** {T : Type} (L : LevCh D T) (μ0 : T) (hL : L.Is μ0) (sc : Consts) (hg : ∀ k, D.host.g k % LevelStarExponents.condE D = 1) (h1 : D.host.pde % LevelStarExponents.condE D = 1) (h2 : D.host.pde' % LevelStarExponents.condE D = 1) (h3 : ∀ k, q0 D ≠ D.host.g k) (h4 : ∀ t, ∀ PP ∈ L.Q t, ((q0 D : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal) : Nonempty (GridData L sc)

theorem **exists_levCh_gridData** (C : Formalism.L2.Host.CharFrame D) (sc : Consts) (hg : ∀ k, D.host.g k % LevelStarExponents.condE D = 1) (h1 : D.host.pde % LevelStarExponents.condE D = 1) (h2 : D.host.pde' % LevelStarExponents.condE D = 1) (h3 : ∀ k, q0 D ≠ D.host.g k) : ∃ L : LevCh D Unit, L.C = C ∧ L.Is () ∧ Nonempty (GridData L sc)

theorem **isEmpty_gridData** {T : Type} (L : LevCh D T) (sc : Consts) (h : q0 D = D.host.g 0) : IsEmpty (GridData L sc)

theorem **ratio_st** {T : Type} {L : LevCh D T} (P : Places L) (sc : Consts) : P.stA sc * P.stAbar sc * (P.modul sc : Ideal (𝒪 D.host.E)) * (P.modulBar sc : Ideal (𝒪 D.host.E)) = D.host.AA * D.host.AAbar

theorem **par_core** {T : Type} (C : ℝ) (τ : D.what₂.Tau) {i : ℕ} (h : (Pair.core i : Pair T).InRange D.R.m) : Node.par D C (Pair.core i : Pair T).big τ = lift (Pair.core i : Pair T).ori (Pair.centre D C (Pair.core i : Pair T) τ) (Pair.len D (Pair.core i : Pair T)) (Node.par D C (Pair.core i : Pair T).small τ)

Development.AbelianVarietyTheory.Definitions.Divisors.NonemptyHomFixRepTrivial

theorem **isGrpPullback_id** (A : Over S) [GrpObj A] : IsGrpPullback A A (𝟙 S) (𝟙 A.left)

theorem **trivLB_isLB** (X : Scheme.{0}) : (trivLB X).IsLB

theorem **pull_g_one** {X Y : Scheme.{0}} (L : LB X) (h : ∀ i j, L.g i j = 1) (f : Y ⟶ X) (i j) : (L.pull f).g i j = 1

theorem **lbIso_of_g_one** {X : Scheme.{0}} (L M : LB X) (hL : ∀ i j, L.g i j = 1) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **nonempty_homFixRep_trivial** (A : Over S) [GrpObj A] : Nonempty (HomFixRep A (𝟙_ (Over S)) (trivLB _) (trivLB _))

Development.AbelianVarietyTheory.Definitions.Divisors.NonemptyIso

theorem **objMap_refl** {b : Base D R} {N : ℕ} {S : Scheme.{0}} {s : S ⟶ Sp R} (o : Obj b N s) : Nonempty (ObjMap o o (𝟙 S))

theorem **eq_id** (F : RepFam b N) (f : F.X ⟶ F.X) (hf : f ≫ F.s = F.s) (h : Nonempty (ObjMap F.univ F.univ f)) : f = 𝟙 _

theorem **nonempty_iso** (F G : RepFam b N) : ∃ e : F.X ≅ G.X, e.hom ≫ G.s = F.s

Development.AbelianVarietyTheory.Definitions.Divisors.NonemptyUWit

theorem **nonempty_uWit** (X : USetup C) : Nonempty (UWit X)

theorem **UIsLabW.uIsLab** {X : USetup C} {wi : UWit X} {p : ℕ} [Fact p.Prime] {z : X.M.Z} {ct : UCTrait X p z} {yy : X.S.Gam →+ ModN (X.S.r * X.S.N) ct.T.P.D.toPre.X} (h : UIsLabW X wi ct yy) : UIsLab X ct yy

Development.AbelianVarietyTheory.Definitions.Divisors.NonemptyValExtOfOverLa

theorem **log_v_p_neg** : Real.log (X.v (p : X.C) : ℝ) < 0

theorem **vX_vp_apply** {x : X.C} (hx : x ≠ 0) : vX_vp X x = (((Real.log (X.v (p : X.C) : ℝ))⁻¹ * Real.log (X.v x : ℝ) : ℝ) : WithTop ℝ)

theorem **vX_vp_p** : vX_vp X (p : X.C) = 1

theorem **isIntV_iff** (x : X.C) : IsIntV (vX_vp X) x ↔ X.v x ≤ 1

theorem **isIntV_jX** (r : T.V) : IsIntV (vX_vp X) (jX X r)

theorem **vX_vp_jX_unit** {r : T.V} (hr : IsUnit r) : vX_vp X (jX X r) = 0

theorem **vX_vp_ιF** {x : T.Fr} (hx : x ≠ 0) : vX_vp X (X.ιF x) = (((vV T.V x : ℝ) / (vV T.V (p : T.Fr) : ℝ) : ℝ) : WithTop ℝ)

theorem **toE_vX_vp_ιF** {e : ℕ} (he : vV T.V (p : T.Fr) = e) (x : T.Fr) : toE (vX_vp X (X.ιF x)) = nval T.V e x

theorem **rk_le** (f : Lt1 Dw) : rk f ≤ Dw.mult f.1

theorem **sum_gap_tail** (g : ℕ → ℕ) (hg : Antitone g) (m i : ℕ) (hi : i < m) : ∑ j ∈ Finset.range m, (if i < j + 1 then g j − (if j + 1 < m then g (j + 1) else 0) else 0) = g i

theorem **sum_gap_row** {a : (E →+* ℂ) × ℕ → ℕ} (ha : ∀ τ, Antitone fun i => a (τ, i)) (τ : E →+* ℂ) (i : Fin (Dw.mult τ)) : ∑ f ∈ Finset.univ.filter (fun f : Lt1 Dw => f.1 = τ ∧ i.val < rk f), gap Dw a f = a (τ, i)

theorem **lawX_F** (a : Fin (chD Ch.ch)) : (lawX Ch X).F a = MvPowerSeries.map (jX X) (Ch.FT a)

theorem **fgl_F** (a : Fin (chD Ch.ch)) : (fgl Ch X).F a = MvPowerSeries.subst (twoVars X.C _) (MvPowerSeries.map (jX X) (Ch.FT a))

theorem **exists_unit_minor** {R : Type} [CommRing R] [IsLocalRing R] {i d : ℕ} (U : Matrix (Fin i) (Fin d) R) (S : Matrix (Fin d) (Fin i) R) (h : U * S = 1) : ∃ r : Fin i → Fin d, IsUnit (Matrix.of fun a b => U a (r b)).det

theorem **isSatFrame_frame** (f : Lt1 Dw) : IsSatFrame (vX_vp X) (frame Ch X f)

theorem **nuLocC_eq_iInf** (e k : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : nuLocC Ch e k a n = ⊓ (P : MvPolynomial (LtN Dw n) T.Fr) (_ : P ∈ FFV (Dw

theorem **nuFac_le** (Ch : IntegralChartA3 Xo c T io y) (X : ValExt T p) (f : Lt1 Dw) (k : ℕ) : nuFac p y f k ≤ nuFacC Ch X f k

theorem **le_nuFac** {f : Lt1 Dw} {k : ℕ} {r : EReal} (h : ∀ (Ch : IntegralChartA3 Xo c T io y) (X : ValExt T p), r ≤ nuFacC Ch X f k) : r ≤ nuFac p y f k

theorem **not_isUnit_of_overLa** (la : PointsOfFlagModel.abelian₂ p c) {e : ℕ} (ho : OverLa la.1 T.V io e) : ¬ IsUnit (p : T.V)

theorem **nonempty_valExt_of_overLa** (la : PointsOfFlagModel.abelian₂ p c) {e : ℕ} (ho : OverLa la.1 T.V io e) : Nonempty (ValExt T p)

theorem **polyT_mem_FFV** {T : CDVR} {io : c.F →+* T.Fr} {p n : ℕ} {a : (E →+* ℂ) × ℕ → ℕ} (ha : ∀ τ, Antitone fun i => a (τ, i)) {B : ∀ f : Lt1 Dw, Fin (rk f) → Fin n → T.V} (hB : IsCols c T io p (gap Dw a) B) : polyT B (gap Dw a) ∈ FFV (Dw

theorem **nuLocC_le_nuLocOf** [StarRing (𝒪 E)] {T : CDVR} {io : c.F →+* T.Fr} {y : IntFr Xo c T io} (Ch : IntegralChartA3 Xo c T io y) (e k : ℕ) {p n : ℕ} {a : (E →+* ℂ) × ℕ → ℕ} (ha : ∀ τ, Antitone fun i => a (τ, i)) {B : ∀ f : Lt1 Dw, Fin (rk f) → Fin n → T.V} (hB : IsCols c T io p (gap Dw a) B) : nuLocC Ch e k a n ≤ nuLocOf Ch e k (polyT B (gap Dw a))

theorem **isTanAtG_map** {X0 : Scheme.{0}} (t : SAB H X0) {S S' : Type} [CommRing S] [CommRing S'] (φ : S →+* S') {x : Sp S ⟶ X0} {v : Sp (DualNumber S) ⟶ t.G.left} (hv : IsTanAtG t x v) : IsTanAtG t (spMap φ ≫ x) (spMap (dualMap φ) ≫ v)

theorem **nuPSup_le** {r : EReal} (h : ∀ (T : CDVR) (io : c.F →+* T.Fr) (e : ℕ), OverLa la.1 T.V io e → ∀ y : IntFr Xo c T io, cls T io y → nuP p y cx ≤ r) : nuPSup cls la cx ≤ r

theorem **le_nuPInf** {r : EReal} (h : ∀ (T : CDVR) (io : c.F →+* T.Fr) (e : ℕ), OverLa la.1 T.V io e → ∀ y : IntFr Xo c T io, cls T io y → r ≤ nuP p y cx) : r ≤ nuPInf cls la cx

Development.AbelianVarietyTheory.Definitions.Divisors.NontrivialOfArtinQE

theorem **act_zero** {A : Over S} [GrpObj A] (act : OActN O A) : act.act 0 = 1

theorem **matEnd_one** {A : Over S} [GrpObj A] (act : OActN O A) (m : ℕ) : matEnd act (1 : Matrix (Fin m) (Fin m) O) = 𝟙 _

theorem **matBC_one** {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) (m : ℕ) : matBC act p b (1 : Matrix (Fin m) (Fin m) O) = 𝟙 _

theorem **op_single_one** (l : ℕ) [Fact l.Prime] {E : Type} [CommRing E] [Algebra ℚ_[l] E] {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) (m : ℕ) (n : ℤ) : op l act p b (Finsupp.single (1 : Matrix (Fin m) (Fin m) O) (1 : E)) n = 1

theorem **Hxi_single_one** (l : ℕ) [Fact l.Prime] {E : Type} [CommRing E] [Algebra ℚ_[l] E] {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) (m : ℕ) (k : ℕ) (i : ℤ) : Hxi l act p b (Finsupp.single (1 : Matrix (Fin m) (Fin m) O) (1 : E)) k i = ⊤

theorem **pts_comm** {A : Over S} [GrpObj A] (act : OActN O A) {T : Over S} (x y : T ⟶ A) : x ∗ y = y ∗ x

theorem **comp_matEnd_pr** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (M : Matrix (Fin m) (Fin m) O) {T : Over S} (x : T ⟶ pow A m) (i : Fin m) : x ≫ matEnd act M ≫ pr A i = (List.ofFn fun j => (x ≫ pr A j) ≫ act.act (M i j)).prod

theorem **matEnd_mul** {A : Over S} [GrpObj A] (act : OActN O A) {m : ℕ} (M N : Matrix (Fin m) (Fin m) O) : matEnd act (M ∗ N) = matEnd act N ≫ matEnd act M

theorem **matBC_mul** {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) {m : ℕ} (M N : Matrix (Fin m) (Fin m) O) : matBC act p b (M ∗ N) = matBC act p b N ≫ matBC act p b M

theorem **pullEt_id** (Y : Scheme.{0}) (n t : ℤ) : pullEt l E (𝟙 Y) n t = LinearMap.id

theorem **pullEt_comp** {X Y Z : Scheme.{0}} (f : X ⟶ Y) (g : Y ⟶ Z) (n t : ℤ) : pullEt l E (f ≫ g) n t = pullEt l E f n t ∘ₗ pullEt l E g n t

theorem **opT_single** {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) {m : ℕ} (M : Matrix (Fin m) (Fin m) O) (c : E) (n t : ℤ) : opT l act p b (Finsupp.single M c) n t = c • pullEt l E (matBC act p b M) n t

theorem **opT_single_mul** {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) {m : ℕ} (M N : Matrix (Fin m) (Fin m) O) (n t : ℤ) : opT l act p b (Finsupp.single (M ∗ N) (1 : E)) n t = opT l act p b (Finsupp.single N (1 : E)) n t ∗ opT l act p b (Finsupp.single M (1 : E)) n t

theorem **altOp_single_mul** {R : Type} [CommRing R] {V : Type} [AddCommGroup V] [Module R V] [Algebra R E] (ρ : O →+* Module.End R V) {m : ℕ} (M N : Matrix (Fin m) (Fin m) O) (q : ℕ) : altOp ρ (Finsupp.single (M ∗ N) (1 : E)) q = altOp ρ (Finsupp.single N (1 : E)) q ∗ altOp ρ (Finsupp.single M (1 : E)) q

theorem **HxiT_single_one** {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) (m k : ℕ) (t i : ℤ) : HxiT l act p b (Finsupp.single (1 : Matrix (Fin m) (Fin m) O) (1 : E)) k t i = ⊤

theorem **galEt_eq** {Y : Scheme.{0}} (q : Y ⟶ Spec (CommRingCat.of k)) (n t : ℤ) (σ : Ω ≃ₐ[k] Ω) : galEt l E Ω q n t σ = (galHQ l Ω q n t σ).baseChange E

theorem **galEt_mul** {Y : Scheme.{0}} (q : Y ⟶ Spec (CommRingCat.of k)) (n t : ℤ) (σ τ : Ω ≃ₐ[k] Ω) : galEt l E Ω q n t (σ ∗ τ) = galEt l E Ω q n t σ ∗ galEt l E Ω q n t τ

theorem **opT_comm_pull** {A : Over S} [GrpObj A] (act : OActN O A) {B B' : Scheme.{0}} (p : S ⟶ B) (b : B' ⟶ B) {m : ℕ} (ε : Matrix (Fin m) (Fin m) O →₀ E) (n t : ℤ) (u : ksBC A m p b ⟶ ksBC A m p b) (h : ∀ M, u ≫ matBC act p b M = matBC act p b M ≫ u) (y : HEt l E (ksBC A m p b) n t) : pullEt l E u n t (opT l act p b ε n t y) = opT l act p b ε n t (pullEt l E u n t y)

theorem **galMap_matBC** {A : Over S} [GrpObj A] (act : OActN O A) (p : S ⟶ Spec (CommRingCat.of k)) {m : ℕ} (σ : Ω ≃ₐ[k] Ω) (M : Matrix (Fin m) (Fin m) O) : GlobalTheory.Et.galMap Ω ((pow A m).hom ≫ p) σ ≫ matBC act p _ M = matBC act p _ M ≫ GlobalTheory.Et.galMap Ω ((pow A m).hom ≫ p) σ

theorem **HxiT_gal_stable** {A : Over S} [GrpObj A] (act : OActN O A) (p : S ⟶ Spec (CommRingCat.of k)) {m : ℕ} (ε : Matrix (Fin m) (Fin m) O →₀ E) (d : ℕ) (t i : ℤ) (σ : Ω ≃ₐ[k] Ω) (x : HEt l E (GlobalTheory.Et.geom Ω ((pow A m).hom ≫ p)) (i + d) t) (hx : x ∈ HxiT l act p (Spec.map (CommRingCat.ofHom (algebraMap k Ω))) ε d t i) : galEt l E Ω ((pow A m).hom ≫ p) (i + d) t σ x ∈ HxiT l act p (Spec.map (CommRingCat.ofHom (algebraMap k Ω))) ε d t i

theorem **powMap_pr** {S' : Scheme.{0}} {A' : Over S'} {A : Over S} (f : S' ⟶ S) (φ : A'.left ⟶ A.left) (h : φ ≫ A.hom = A'.hom ≫ f) (m : ℕ) (i : Fin m) : powMap f φ h m ≫ (pr A i).left = (pr A' i).left ≫ φ

theorem **powMap_over** {S' : Scheme.{0}} {A' : Over S'} {A : Over S} (f : S' ⟶ S) (φ : A'.left ⟶ A.left) (h : φ ≫ A.hom = A'.hom ≫ f) (m : ℕ) : powMap f φ h m ≫ (pow A m).hom = (pow A' m).hom ≫ f

theorem **nontrivial_of_artinQE** (F : Type) [Field F] [Algebra ℚ_[l] F] (h : ArtinQE l F) : Nontrivial (HE l F (Spec (CommRingCat.of ℂ)) (0 : ℕ))

theorem **matEnd_shear** {A : Over S} [GrpObj A] (act : OActN O A) : matEnd act !![1, 1; 0, 1] ≫ pr A 0 = pr A 0 ∗ pr A 1 ∧ matEnd act !![1, 1; 0, 1] ≫ pr A 1 = pr A 1

Development.AbelianVarietyTheory.Definitions.Divisors.NormNegCmConstantEqAtAllEmbeddings

theorem **latGood_le** : latGood R F i ≤ latType R F i

theorem **uWt_anchor** (kc : ℕ) (τ : Psi H) (k : Fin (Dw.p + Dw.q)) (hk : (k : ℕ) + 1 = kc) : uWt Dw a n kc τ k = 0

theorem **uWt_add_anchor** (kc : ℕ) : uWt Dw a n kc + GlobalTheory.FlatLine.parWt Dw (anchor Dw a n kc) = laOf Dw a n

theorem **sizeK_nonneg** : 0 ≤ sizeK Dw a n

theorem **cZeroN_isLeast** (D : D7) (j : ℕ) : IsLeast {s : ℕ | 8 ∣ s ∧ D.R.r + (D.R.f * (D.R.sigW j).1 * (D.R.sigW j).2 + 1).choose 2 + 2 ≤ s ∧ 8 * D.R.f * D.R.r + 4 ≤ s} (cZeroN D j)

theorem **cZeroN_le** (D : D7) (j : ℕ) : cZeroN D j ≤ 22 * D.R.f ^ 2 * D.R.r ^ 4

theorem **cZero_spec** (D : D7) (j : ℕ) : 8 ∣ cZero D j ∧ 0 ≤ cZero D j ∧ cZero D j ≤ 22 * (D.R.f : ℤ) ^ 2 * (D.R.r : ℤ) ^ 4

theorem **log_finrank_le_fieldSz** (K : Type) [Field K] [NumberField K] : Real.log (Module.finrank ℚ K : ℝ) ≤ fieldSz K

theorem **log_rd_le_fieldSz** (K : Type) [Field K] [NumberField K] : Real.log |(discr K : ℝ)| / (Module.finrank ℚ K : ℝ) ≤ fieldSz K

theorem **le_expL** {L : ℝ} {e : ℕ} (h : e * Real.log 3 ≤ 3 * L) : e ≤ expL L

theorem **good_resEmb** {σ : Y.d.c₂.F →+* ℂ} (hσ : GoodEmb Y.d.c₂ σ) : GoodEmb T.Q.c (Y.resEmb σ)

theorem **Good.deg** {e : ℕ} {o : ℝ} (h : Y.Good e o) : Real.log (Module.finrank ℚ Y.d.c₂.F : ℝ) ≤ o

theorem **Good.rd** {e : ℕ} {o : ℝ} (h : Y.Good e o) : Real.log |(discr Y.d.c₂.F : ℝ)| ≤ o * Module.finrank ℚ Y.d.c₂.F

theorem **Good.mono** {e : ℕ} {o o' : ℝ} (h : Y.Good e o) (ho : o ≤ o') : Y.Good e o'

theorem **bar_root** : H.bar H.rt = -H.rt

theorem **exists_half** (s : ℤ) (hs : (15 : ℤ) ∣ s ^ 2 + H.d) (e : ℕ) : ∃ I : Ideal (𝓞 E), (∃ G : Set (𝓞 E), I = Ideal.span G ∧ G ⊆ Algebra.adjoin ℤ {H.w0}) ∧ I * I.map (H.bar : 𝓞 E →+* 𝓞 E) = Ideal.span {((15 ^ e : ℕ) : 𝓞 E)} ∧ I ⊔ I.map (H.bar : 𝓞 E →+* 𝓞 E) = ⊤

theorem **normalClosure_stable** (K : IntermediateField ℚ ℂ) (s : ℂ ≃+* ℂ) (a : ℂ) (ha : a ∈ IntermediateField.normalClosure ℚ K ℂ) : s a ∈ IntermediateField.normalClosure ℚ K ℂ

theorem **exists_root** (h : 5 ≤ D.host.splitBound) : ∃ s : ℤ, (15 : ℤ) ∣ s ^ 2 + D.host.d

theorem **norm_neg_cmConstant_eq_at_all_embeddings** (τ : D.host.E →+* ℂ) : ‖τ (-D.host.xi0)‖ = xiR D

theorem **rank_pos** (T : Core D) (hQ : T.Q.Pins) (j : ℕ) (hj : InR D j) : 1 ≤ (T.Q.Dw j).p + (T.Q.Dw j).q

theorem **exists_good_pin** (T : Core D) (hQ : T.Q.Pins) (j : ℕ) (hj : InR D j) (c' : Coeff T.Q.l) (hle : T.Q.c.F ≤ c'.F) (d : DeltaDat (T.Q.Xo j) c' 0) (e : ℕ) (a1 : d.pd = 15 ^ e) (a2 : d.IsLev) (a3 : d.l₂.A = 1) (a4 : FnAlgClosed d.Xo₂ d.c₂) (a5 : ∀ s : ℂ ≃+* ℂ, (∀ a ∈ T.Q.c.F, s a = a) → ∀ a ∈ c'.F, s a ∈ c'.F) (ι : (T.Q.Dw j).La →ₗ[𝓞 D.host.E] (Fin (D.sp.W j).rank → D.host.E)) (hι : GlobalTheory.Member.IotaOK (T.Q.Dw j) (D.sp.W j) ι) (x : (d.c₂.F →+* ℂ) → (D.host.E →+* ℂ) → Matrix (Fin D.R.r) (Fin D.R.r) ℂ) (hx : ∀ σ, GoodEmb d.c₂ σ → ∃ C : GlobalTheory.Member.Ctx d.Xo₂ d.c₂ σ (D.sp.W j) D.R.r, C.OK ∧ C.ι = ι ∧ C.x₀ = x σ) : ∃ U dict Kf, ((c', hle, d, U, dict, Kf) : Deep T j).Good e (fieldSz d.c₂.F) ∧ ∀ σ, GoodEmb d.c₂ σ → ((c', hle, d, U, dict, Kf) : Deep T j).Pin σ ι (x σ)

theorem **exists_good_of_cover** (T : Core D) (hQ : T.Q.Pins) (j : ℕ) (hj : InR D j) (c' : Coeff T.Q.l) (hle : T.Q.c.F ≤ c'.F) (d : DeltaDat (T.Q.Xo j) c' 0) (e : ℕ) (a1 : d.pd = 15 ^ e) (a2 : d.IsLev) (a3 : d.l₂.A = 1) (a4 : FnAlgClosed d.Xo₂ d.c₂) (a5 : ∀ s : ℂ ≃+* ℂ, (∀ a ∈ T.Q.c.F, s a = a) → ∀ a ∈ c'.F, s a ∈ c'.F) : ∃ U dict Kf, ((c', hle, d, U, dict, Kf) : Deep T j).Good e (fieldSz d.c₂.F)

theorem **exists_match_of_cover** (T : CoreJ D) (hT : LawsJ T) (j : ℕ) (hj : InR D j) (c' : Coeff T.Q.l) (hle : T.Q.c.F ≤ c'.F) (d : DeltaDat (T.Q.Xo j) c' 0) (e : ℕ) (a1 : d.pd = 15 ^ e) (a2 : d.IsLev) (a3 : d.l₂.A = 1) (a4 : FnAlgClosed d.Xo₂ d.c₂) (a5 : ∀ s : ℂ ≃+* ℂ, (∀ a ∈ T.Q.c.F, s a = a) → ∀ a ∈ c'.F, s a ∈ c'.F) : ∃ U dict Kf, ((c', hle, d, U, dict, Kf) : Deep T.toCore j).Good e (fieldSz d.c₂.F) ∧ Deep.Match ((c', hle, d, U, dict, Kf) : Deep T.toCore j)

theorem **exists_cover** (T : Core D) (hQ : T.Q.Pins) (j : ℕ) (hj : InR D j) (s : ℤ) (hs : (15 : ℤ) ∣ s ^ 2 + D.host.d) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) (e : ℕ) : ∃ (c' : Coeff T.Q.l) (_ : T.Q.c.F ≤ c'.F) (d : DeltaDat (T.Q.Xo j) c' 0), d.pd = 15 ^ e ∧ d.IsLev ∧ d.l₂.A = 1 ∧ FnAlgClosed d.Xo₂ d.c₂ ∧ ∀ s : ℂ ≃+* ℂ, (∀ a ∈ T.Q.c.F, s a = a) → ∀ a ∈ c'.F, s a ∈ c'.F

theorem **exists_good** (T : Core D) (hQ : T.Q.Pins) (j : ℕ) (hj : InR D j) (s : ℤ) (hs : (15 : ℤ) ∣ s ^ 2 + D.host.d) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) (e : ℕ) : ∃ (Y : Deep T j) (o : ℝ), Y.Good e o ∧ Y.d.l₂.M = D.host.g 0 * D.host.g 1 * 15 ^ e ∧ CompNe Y.d.Xo₂ Y.d.c₂

theorem **exists_good_match** (T : CoreJ D) (hT : LawsJ T) (j : ℕ) (hj : InR D j) (s : ℤ) (hs : (15 : ℤ) ∣ s ^ 2 + D.host.d) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) (e : ℕ) : ∃ (Y : Deep T.toCore j) (o : ℝ), Y.Good e o ∧ Y.Match

theorem **pull_one** {T : Core D} {j : ℕ} (Y : Deep T j) : Y.Pull 1 1

theorem **not_pull_one_zero** {T : Core D} {j : ℕ} (Y : Deep T j) {e : ℕ} {o : ℝ} (h : Y.Good e o) (hpos : 1 ≤ (T.Q.Dw j).p + (T.Q.Dw j).q) : ¬ Y.Pull 1 0

Development.AbelianVarietyTheory.Definitions.Divisors.NormSqColGrLineOfOrthogonal

theorem **card_rows** (N : ℕ) (l : Fin (N + 1)) : Fintype.card (Rows N l) = plDim N l + 1

@[simp] theorem **plVec_rowsEquiv** (g : Matrix (Fin N) (Fin N) A) (l : Fin (N + 1)) (I : Rows N l) : plVec g l (rowsEquiv N l I) = minor g l I

theorem **minor_map** (ψ : A →+* B) (g : Matrix (Fin N) (Fin N) A) (l : Fin (N + 1)) (I : Rows N l) : minor (g.map ψ) l I = ψ (minor g l I)

theorem **span_plVec** (g : GL (Fin N) A) (l : Fin (N + 1)) : Ideal.span (Set.range (plVec g.1 l)) = ⊤

theorem **minor_mul** (g b : Matrix (Fin N) (Fin N) A) (l : Fin (N + 1)) (hb : ∀ i j : Fin N, (j : ℕ) < l → (l : ℕ) ≤ i → b i j = 0) (I : Rows N l) : minor (g * b) l I = minor g l I * (b.submatrix (Fin.castLE l.is_le) (Fin.castLE l.is_le)).det

theorem **volSq_eq_det** (g : Matrix (Fin N) (Fin N) ℂ) (l : Fin (N + 1)) : (volSq g l : ℂ) = ((cols g l)ᴴ * cols g l).det

theorem **volSq_of_orthogonal** {g : Matrix (Fin N) (Fin N) ℂ} {d : Fin N → ℝ} (h : gᴴ * g = diagonal fun i => (d i : ℂ)) (l : Fin (N + 1)) : volSq g l = ∏ i : Fin l, d (Fin.castLE l.is_le i)

theorem **eval₂_genDet** (g : ι → Matrix (Fin N) (Fin N) A) : MvPolynomial.eval₂Hom (Int.castRingHom A) (fun x => g x.1 x.2.1 x.2.2) (genDet ι N) = ∏ τ, (g τ).det

theorem **genDet_ne_zero** : genDet ι N ≠ 0

theorem **univGL_apply** (τ : ι) (i j : Fin N) : (univGL ι N τ).1 i j = algebraMap _ _ (MvPolynomial.X (R

theorem **evalGL_X** (g : ι → GL (Fin N) A) (x : ι × Fin N × Fin N) : evalGL g (algebraMap _ _ (MvPolynomial.X (R

theorem **map_evalGL_univGL** (g : ι → GL (Fin N) A) (τ : ι) : GeneralLinearGroup.map (evalGL g) (univGL ι N τ) = g τ

theorem **evalGL_map_univGL** (ψ : GLRing ι N →+* A) : evalGL (fun τ => GeneralLinearGroup.map ψ (univGL ι N τ)) = ψ

theorem **toZ_ext** {X : Scheme.{0}} (f g : X ⟶ Sp ℤ) : f = g

theorem **isProjOver_pi** (κ : Type) [Finite κ] : ∀ (X : κ → Scheme.{0}) [HasProduct X], (∀ k, IsProjOver (toZ (X k))) → IsProjOver (toZ (∏ᶜ X))

theorem **isProjOver_amb** : IsProjOver (toZ (Amb l))

@[reassoc] theorem **plPt_π** (g : ι → GL (Fin N) A) (k : κ) : plPt a l g ≫ Pi.π _ k = pt (Int.castRingHom A) (plVec (g (a k)).1 (l k)) (span_plVec _ _)

@[reassoc] theorem **spMap_plPt** (ψ : A →+* B) (g : ι → GL (Fin N) A) : spMap ψ ≫ plPt a l g = plPt a l fun τ => GeneralLinearGroup.map ψ (g τ)

@[reassoc] theorem **flagPt_plEmb** (g : ι → GL (Fin N) A) : flagPt a l g ≫ plEmb a l = plPt a l g

@[reassoc] theorem **spMap_flagPt** (ψ : A →+* B) (g : ι → GL (Fin N) A) : spMap ψ ≫ flagPt a l g = flagPt a l fun τ => GeneralLinearGroup.map ψ (g τ)

theorem **flagPt_univGL** : flagPt a l (univGL ι N) = orbit a l

theorem **flagPt_eq_iff** (g h : ι → GL (Fin N) A) : flagPt a l g = flagPt a l h ↔ ∀ k, ∃ u : Aˣ, ∀ I, minor (h (a k)).1 (l k) I = u * minor (g (a k)).1 (l k) I

theorem **flagPt_mul** (g b : ι → GL (Fin N) A) (hb : ∀ k (i j : Fin N), (j : ℕ) < l k → (l k : ℕ) ≤ i → (b (a k)).1 i j = 0) : flagPt a l (g * b) = flagPt a l g

theorem **isProjOver_str** : IsProjOver (str a l)

theorem **charZero** : CharZero Γ(Part a l, ⊤)

theorem **isArithVar_of_dim** {D : ℕ} (h : topologicalKrullDim (Part a l) = (D + 1 : ℕ)) : IsArithVar (str a l) D

@[reassoc] theorem **comap_plEmb_π** {ι' κ' : Type} [Fintype ι'] [Finite κ'] (a' : κ' → ι') (f : ι' → ι) (e : κ' → κ) (h : ∀ k, a (e k) = f (a' k)) (k : κ') : comap a l a' f e h ≫ plEmb a' (fun k => l (e k)) ≫ Pi.π _ k = plEmb a l ≫ Pi.π _ (e k)

theorem **map_cxPt** (σ : ℤ →+* ℂ) (g : ι → GL (Fin N) ℂ) (k : κ) : PtsOver.map (plEmb a l ≫ Pi.π _ k) (toZ_ext _ (str a l)) _ (cxPt a l σ g) = projPt σ (plVec (g (a k)).1 (l k), span_plVec _ _)

theorem **detDual_φ** (σ : ℤ →+* ℂ) (g : ι → GL (Fin N) ℂ) (k : κ) (I : Rows N (l k)) (hI : minor (g (a k)).1 (l k) I ≠ 0) : ((detDual a l k).φ σ).φ (frameIdx a l k I) (cxPt a l σ g) = Real.log (volSq (g (a k)).1 (l k) / ‖minor (g (a k)).1 (l k) I‖ ^ 2)

theorem **iSup_nonzeroLocus_plSec** (k : κ) : ⨆ I, nonzeroLocus (plSec a l k I) = ⊤

theorem **sum_nsq_plSec** (σ : ℤ →+* ℂ) (k : κ) (y : CxPts (str a l) σ) : ∑ I, ((detDual a l k).φ σ).nsq (secVal _ σ (plSec a l k I)) y = 1

theorem **plSec_mem_smallSec** (k : κ) (I : Rows N (l k)) : plSec a l k I ∈ (detDual a l k).smallSec

theorem **semiPos_detDual** (k : κ) (D : ℕ) : (detDual a l k).SemiPos D

theorem **det_lowMat** (τ : ι) : (lowMat a l τ).det = 1

theorem **flagPt_eq_comp_cell** (w : ι → Equiv.Perm (Fin N)) (L b : ι → GL (Fin N) A) (hL : ∀ τ i j, ¬ Below a l τ (w τ i) j → (L τ).1 i j = (1 : Matrix _ _ A) (w τ i) j) (hb : ∀ τ i j, Below a l τ i j → (b τ).1 i j = 0) : flagPt a l (L * b) = spMap (MvPolynomial.eval₂Hom (Int.castRingHom A) fun x : CellVar a l => (L x.1.1).1 ((w x.1.1).symm x.1.2.1) x.1.2.2) ≫ cell a l w

theorem **flagPt_mul_upper** {A : Type} [CommRing A] (g b : ι → GL (Fin N) A) (hb : ∀ τ, (b τ).1.BlockTriangular id) : flagPt (Prod.fst : ι × Fin (N + 1) → ι) Prod.snd (g * b) = flagPt Prod.fst Prod.snd g

theorem **isHh_hh** (h : HasHh ι N) : IsHh ι N (hh h)

theorem **IsHh.eq_hh** {E : ℕ} (hd : topologicalKrullDim (Fl ι N) = (E + 1 : ℕ)) {Φ : Pol ι N →ₗ[ℝ] ℝ} (hΦ : IsHh ι N Φ) : Φ = hh ⟨Φ, hΦ⟩

theorem **IsHh.isSelfInter** {Φ : Pol ι N →ₗ[ℝ] ℝ} (hΦ : IsHh ι N Φ) (lam : ι → Fin N → ℤ) : (wtLine lam).IsSelfInter (Φ (linRoot (fun τ i => (lam τ i : ℝ)) ^ (dimFl ι N + 1)))

theorem **IsHh.eq_zero** {Φ : Pol ι N →ₗ[ℝ] ℝ} (hΦ : IsHh ι N Φ) {F : Pol ι N} {d : ℕ} (hd : d ≠ dimFl ι N + 1) (hF : F.IsHomogeneous d) : Φ F = 0

Development.AbelianVarietyTheory.Definitions.Divisors.NotCoinvQuotEmpty

theorem **comm** {G H Y : Type} [Group G] [Group H] (om : G × H →* Module.End ℂ (Y → ℂ)) (g : G) (h : H) (φ : Y → ℂ) : om (1, h) (om (g, 1) φ) = om (g, 1) (om (1, h) φ)

theorem **locAug_le_ker** (h : IsLinPair c ψ ξ JA JB d χ Cy om) (hl : IsYInv c ψ d Cy l) : locAug om ≤ LinearMap.ker (secMap om l)

theorem **locAug_le_SB** (h : IsLinPair c ψ ξ JA JB d χ Cy om) : locAug om ≤ SB (κ → k)

theorem **locAug_ne_SB** (h : IsLinPair c ψ ξ JA JB d χ Cy om) (hl : IsYForm c ψ d Cy l) : locAug om ≠ SB (κ → k)

theorem **sec_left** (h : IsLinPair c ψ ξ JA JB d χ Cy om) (hl : IsYInv c ψ d Cy l) {f : UG c JB → ℂ} (hf : f ∈ secSpace om l) (p : UG c JB) (t : GL ν E) (hp : ((p : GL β E) : Matrix β β E) * Cy = Cy * (t : Matrix ν ν E)) (x : UG c JB) : f (p * x) = (χ (Matrix.GeneralLinearGroup.det t) : ℂ) * (((MeasureTheory.distribHaarChar k (Units.map (Algebra.norm k : E →* k) (Matrix.GeneralLinearGroup.det t)) : ℝ) ^ ((Fintype.card α : ℝ) / 2) : ℝ) : ℂ) * f x

theorem **sec_right** (h : IsLinPair c ψ ξ JA JB d χ Cy om) {f : UG c JB → ℂ} (hf : f ∈ secSpace om l) (y : UG c JB) : (fun x => f (x * y)) ∈ secSpace om l

theorem **isYInv_eval_zero** (hd : ∀ z : Matrix α ν E, (d.e (z * Cyᵀ)).1 = 0) : IsYInv c ψ d Cy (LinearMap.proj (0 : κ → k))

theorem **SB_top** (Y : Type) [TopologicalSpace Y] [DiscreteTopology Y] [Finite Y] : SB Y = ⊤

theorem **locAug_one** (G H Y : Type) [Group G] [Group H] [TopologicalSpace Y] : locAug (1 : G × H →* Module.End ℂ (Y → ℂ)) = ⊥

theorem **coinvQuot_point** (G G' H Y Y' : Type) [Group G] [Group G'] [Group H] [TopologicalSpace Y] [DiscreteTopology Y] [Finite Y] [Nonempty Y] [TopologicalSpace Y'] [Unique Y'] : IsCoinvQuot (1 : G × H →* Module.End ℂ (Y → ℂ)) (1 : G' × H →* Module.End ℂ (Y' → ℂ))

theorem **not_coinvQuot_two** {G Y : Type} [Group G] [TopologicalSpace Y] (om : G × Unit →* Module.End ℂ (Y → ℂ)) : ¬ IsCoinvQuot om (1 : Unit × Unit →* Module.End ℂ (Bool → ℂ))

theorem **not_coinvQuot_empty** {G G' H Y' : Type} [Group G] [Group G'] [Group H] [TopologicalSpace Y'] (om : G × H →* Module.End ℂ (Empty → ℂ)) (om' : G' × H →* Module.End ℂ (Y' → ℂ)) (hle : locAug om' ≤ SB Y') : ¬ IsCoinvQuot om om'

theorem **not_coinvQuot_sign** (om : Unit × ℂˣ →* Module.End ℂ (Unit → ℂ)) (hom : ∀ u φ, om (1, u) φ = (u : ℂ) • φ) : ¬ IsCoinvQuot om (1 : Unit × ℂˣ →* Module.End ℂ (Unit → ℂ))

theorem **wittStep_plane** : IsWittStep c 1 (!![0, 1; 1, 0] : Matrix (Fin 2) (Fin 2) E) (1 : Matrix (Fin 0) (Fin 0) E)

theorem **not_wittStep_aniso** {a a' r : ℕ} (hr : 0 < r) (JA : Matrix (Fin a) (Fin a) E) (JA' : Matrix (Fin a') (Fin a') E) (h : ∀ x : Fin a → E, x ⬝ᵥ JA *ᵥ (c ∘ x) = 0 → x = 0) : ¬ IsWittStep c r JA JA'

theorem **lagr_plane** : IsLagr c (!![0, 1; -1, 0] : Matrix (Fin 2) (Fin 2) E) (!![1; 0] : Matrix (Fin 2) (Fin 1) E)

theorem **linPair_one** (ψ : AddChar k ℂ) (ξ : E) (JA : Matrix (Fin 0) (Fin 0) E) {β ν : Type} [Fintype β] [DecidableEq β] [Fintype ν] [DecidableEq ν] (JB : Matrix β β E) (Cy : Matrix β ν E) (hL : IsLagr c JB Cy) (co : Darboux (symp c ξ JA JB) Empty) : IsLinPair c ψ ξ JA JB co 1 Cy 1

theorem **rankZero** (ψ : AddChar k ℂ) : RankZero c ψ

theorem **rankZero_sqrt** (K : Type) [Field K] [CharZero K] [Invertible (2 : K)] [ValuativeRel K] [TopologicalSpace K] [IsNonarchimedeanLocalField K] (e : K) [Fact (∀ r : K, r ^ 2 ≠ e + 0 * r)] : ∃ (_ : TopologicalSpace (QuadraticAlgebra K e 0)) (_ : IsModuleTopology K (QuadraticAlgebra K e 0)) (c : QuadraticAlgebra K e 0 →ₐ[K] QuadraticAlgebra K e 0) (δ : QuadraticAlgebra K e 0) (ψ : AddChar K ℂ), IsQuad c δ ∧ Continuous ψ ∧ (∀ t, ‖ψ t‖ = 1) ∧ ψ ≠ 1 ∧ RankZero c ψ

Development.AbelianVarietyTheory.Definitions.Divisors.NotCompactOfMEqOne

theorem **Datum.not_compact_of_m_eq_one** (D : Datum p) (hm : D.m = 1) (hn : 2 ≤ D.n) (ha : Module.finrank ℚ D.F ≤ 2 * D.a) : ¬ D.Compact

theorem **Datum.compact_of_a_eq_zero** (D : Datum p) (ha : D.a = 0) : D.Compact

theorem **Datum.IsSimple.compact** {D : Datum p} (h : D.IsSimple) : D.Compact

Development.AbelianVarietyTheory.Definitions.Divisors.NotDominatesTopPlace

theorem **IsEtTower.refl** {c : k →+* K} {R : Subring K} (h : IsLocModel c R) (h' : IsIntegrallyClosed R) : IsEtTower c R c R (RingHom.id R) (RingHom.id R) (Fin.elim0 : Fin 0 → R) where

theorem **_root_.GlobalTheory.FieldPair.nonUnit_iff** {R : Subring K} (f : R) : _root_.GlobalTheory.FieldPair.NonUnit R f ↔ ¬ IsUnit f

theorem **_root_.GlobalTheory.FieldPair.nonUnit_zero** (R : Subring K) : _root_.GlobalTheory.FieldPair.NonUnit R 0

theorem **_root_.GlobalTheory.FieldPair.not_nonUnit_one** (R : Subring K) : ¬ _root_.GlobalTheory.FieldPair.NonUnit R 1

theorem **Dominates.refl** (R : Subring K) : Dominates (RingHom.id K) R R

theorem **_root_.GlobalTheory.FieldPair.Dominates.trans** {M : Type} [Field M] {ρ : K →+* L} {σ : L →+* M} {O : Subring K} {B : Subring L} {C : Subring M} (h : _root_.GlobalTheory.FieldPair.Dominates ρ B O) (h' : _root_.GlobalTheory.FieldPair.Dominates σ C B) : _root_.GlobalTheory.FieldPair.Dominates (σ.comp ρ) C O

theorem **_root_.GlobalTheory.FieldPair.ratRes_self** : _root_.GlobalTheory.FieldPair.RatRes (RingHom.id k) ⊤

Development.AbelianVarietyTheory.Definitions.Divisors.NotExistsDataOfNotSymm

theorem **exists_tautBase** {o W : Type} [CommRing o] [CommRing W] [IsLocalRing W] [IsNoetherianRing W] (b : o →+* W) (P : PolAV (Sp W)) (r : ℕ) (x : Fin r → (𝟙_ (Over (Sp W)) ⟶ P.A)) : ∃ T : TautBase o W, T.b = b ∧ T.g = P.g ∧ ∃ (hB : T.B = P.A) (hr : T.r = r), HEq T.grp P.grp ∧ HEq T.M.L P.L ∧ ∀ i : Fin T.r, HEq (T.cₒ (Pi.single i 1)) (x (Fin.cast hr i))

theorem **exists_tautBase_curve** {o W : Type} [CommRing o] [CommRing W] [IsLocalRing W] [IsNoetherianRing W] (b : o →+* W) (h37 : IsUnit (37 : W)) : ∃ T : TautBase o W, T.b = b ∧ T.g = 1 ∧ T.r = 1 ∧ (∀ t : T.M.L.Gen, ¬ t.IsGen) ∧ ∃ a, T.cₒ a ≠ 1

theorem **exists_tautBase_rat** : ∃ T : TautBase ℤ ℚ, T.g = 1 ∧ T.r = 1 ∧ (∀ t : T.M.L.Gen, ¬ t.IsGen) ∧ ∃ a, T.cₒ a ≠ 1

theorem **exists_tautBase_two_adic** : ∃ T : TautBase ℤ ℤ_[2], T.g = 1 ∧ T.r = 1 ∧ (∀ t : T.M.L.Gen, ¬ t.IsGen) ∧ ∃ a, T.cₒ a ≠ 1

theorem **cₒ_zero** : T.cₒ 0 = 1

theorem **val_τₒ_zero** (a : Fin T.r → ℤ) (z J) (hJ : Good T.M.L z J) : val (T.τₒ 0 a) (𝟙 _) z J = 1

theorem **τₒ_add_right** (a a' a'' : Fin T.r → ℤ) (z J) (hJ : Good T.M.L z J) : val (T.τₒ a (a' + a'')) (𝟙 _) z J * cubV T.cub z J (T.cₒ a) (T.cₒ a') (T.cₒ a'') = val (T.τₒ a a') (𝟙 _) z J * val (T.τₒ a a'') (𝟙 _) z J

theorem **exists_good_choice** : ∃ (z : (𝟙_ (Over (Sp W))).left) (J : _), Good T.M.L z J

theorem **val_τₒ_torus** (b : o →+* W) (r : ℕ) (u : Γ((𝟙_ (Over (Sp W))).left, ⊤)ˣ) (β : LinearMap.BilinForm ℤ (Fin r → ℤ)) (hβ : ∀ a a', β a a' = β a' a) (a a' : Fin r → ℤ) (z J) (hJ : Good (TautBase.torus b r u β hβ).M.L z J) : val ((TautBase.torus b r u β hβ).τₒ a a') (𝟙 _) z J = gm (𝟙 _) z ⊤ (u ^ β a a')

theorem **prod_lamPts_symm** {S : Scheme.{0}} {B : Over S} [GrpObj B] [IsCommMonObj B] {Z : Over S} (x y : Z ⟶ B) {U : Type} [CommGroup U] (h : (Z.left ⟶ B.left) → U) : ∏ i, h (lamPts x y i) ^ negε i = ∏ i, h (lamPts y x i) ^ negε i

theorem **tauCoord_symm** {R : Type} [CommRing R] [IsDomain R] [IsLocalRing R] (Q : Data R) (a b : Q.X) (k k' : (FractionRing R)ˣ) (h : Q.TauCoord a b k) (h' : Q.TauCoord b a k') : ∃ x x' : R, IsUnit x ∧ IsUnit x' ∧ (k : FractionRing R) * algebraMap R _ x = k' * algebraMap R _ x'

theorem **not_exists_data_of_not_symm** [IsDomain (MvPowerSeries (Fin 1) ℚ)] : ∃ ε : Fin 1 → LinearMap.BilinForm ℤ (Fin 2 → ℤ), ¬ ∃ (Q : Data (MvPowerSeries (Fin 1) ℚ)) (e : Q.X ≃+ (Fin 2 → ℤ)), ∀ a b, ∃ k : (FractionRing (MvPowerSeries (Fin 1) ℚ))ˣ, (k : FractionRing (MvPowerSeries (Fin 1) ℚ)) = Taut.perFactor ℚ 1 1 1 2 ε (e a) (e b) ∧ Q.TauCoord a b k

theorem **not_forall_exists_data** : ¬ ∀ {o W : Type} [CommRing o] [CommRing W] [IsDomain W] [IsLocalRing W] [IsNoetherianRing W] (T : TautBase o W) (N m n : ℕ) [IsDomain (MvPowerSeries (Fin N) W)] (ε : Fin N → LinearMap.BilinForm ℤ (Fin T.r → ℤ)), ∃ (Q : Data (MvPowerSeries (Fin N) W)) (e : Q.X ≃+ (Fin T.r → ℤ)) (β : Q.B.left ⟶ T.B.left), IsGrpPullback Q.B T.B (spMap MvPowerSeries.C) β ∧ (∀ a, (Q.c a).left ≫ β = spMap MvPowerSeries.C ≫ (T.cₒ (e a)).left) ∧ ∀ a b, ∃ k : (FractionRing (MvPowerSeries (Fin N) W))ˣ, (k : FractionRing (MvPowerSeries (Fin N) W)) = Taut.perFactor W m N n T.r ε (e a) (e b) ∧ Q.TauCoord a b k

Development.AbelianVarietyTheory.Definitions.Divisors.NotHasNonzeroSecOfIsEmpty

theorem **not_hasNonzeroSec_of_isEmpty** [IsEmpty X] : ¬ HasNonzeroSec L

theorem **empty_values** [IsEmpty X] : ¬ IsBig k L ∧ ¬ IsPsEff L

theorem **isNef_of_dim_le_zero** (hX : topologicalKrullDim X ≤ 0) (hL : L.IsLB) : IsNef k L

theorem **genericPoint_mem** [IsIntegral X] (hd : topologicalKrullDim X = 1) {Z : Closeds X} (hZ : IsCompleteCurve k Z) : genericPoint X ∈ Z

theorem **isNef_of_isPsEff** [IsIntegral X] (hd : topologicalKrullDim X = 1) (h : IsPsEff N) : IsNef k N

theorem **point_values** (n : ℤ) : IsNef k (lbZPow (O1 k 0) n) ∧ IsBig k (lbZPow (O1 k 0) n) ∧ IsPsEff (lbZPow (O1 k 0) n)

theorem **isPsEff_iff** (n : ℤ) : IsPsEff (lbZPow (O1 k (D + 1)) n) ↔ 0 ≤ n

theorem **isBig_iff'** (n : ℤ) : IsBig k (lbZPow (O1 k (D + 1)) n) ↔ 0 < n

theorem **isNef_iff_line** (n : ℤ) : IsNef k (lbZPow (O1 k 1) n) ↔ 0 ≤ n

theorem **degree_line** (n : ℤ) : interNumRed k (⊤ : Closeds (PSpace k 1)) ![lbZPow (O1 k 1) n] = n

Development.AbelianVarietyTheory.Definitions.Divisors.NotIsChowFormIntLineZero

theorem **_root_.GlobalTheory.CompHt.Test.imageDeg_zero** {K V Ω : Type} [Field K] [AddCommGroup V] [Module K V] [Field Ω] [Algebra K Ω] (D : ℕ) : _root_.GlobalTheory.CompHt.imageDeg (0 : V →ₗ[K] Ω) D = 0

theorem **not_isChowFormInt_line_zero** {K : Type} [Field K] {δ : ℕ} {g : MvPolynomial (Fin (0 + 1) × Fin 2) K} (hf' : IsChowFormInt (⊥ : Ideal (MvPolynomial (Fin 2) K)) 0 δ g) : False

theorem **imageDeg_line_zero** : imageDeg (stdPoint 2) 0 = 0

theorem **countVol_two_pow** {d a : ℕ} (ha : a ≠ 0) : countVol 1 d a (fun τ => ((2 ^ (τ ^ 2) : ℕ) : ℕ∞)) = ENNReal.ofReal (Real.log 2) / d

theorem **volSlope_zero** {δ : ℝ} : volSlope δ (fun _ => 0) = ⊥

theorem **dim_count** (p q : ℕ) : p * q + p * (p - 1) / 2 + q * (q - 1) / 2 = (p + q) * (p + q - 1) / 2

theorem **isShortGen_std** (N : ℕ) : IsShortGen (stdNorm N) (stdLat N) (Pi.basisFun ℚ (Fin N)) 1

theorem **exps_stdWt** (τ : Psi H) (i j : ℕ) (hi : i < Dw.p) (hj : j < Dw.q) : (wtOf Dw (stdWt (Psi H) (Dw.p + Dw.q)) 0).1 (τ.1, i) = (Dw.p + Dw.q : ℤ) - 1 - i ∧ (wtOf Dw (stdWt (Psi H) (Dw.p + Dw.q)) 0).1 (conjEmb τ.1, j) = -(j : ℤ)

theorem **exps_hodgeWt** (τ : Psi H) (i j : ℕ) (hi : i < Dw.p) (hj : j < Dw.q) : (wtOf Dw (hodgeWt Dw) 0).1 (τ.1, i) = 0 ∧ (wtOf Dw (hodgeWt Dw) 0).1 (conjEmb τ.1, j) = 1

theorem **valL_vw_one_nonneg** (G : WTZ E) (v : Rul Xo c → EReal) (h1 : depth_vw U G v ≠ ⊤) (h2 : depth_vw U G v ≠ ⊥) {w : cuspW Xo c U G} (hw : w ∈ latW U G) : 0 ≤ valL_vw U G 1 v w

theorem **valL_vw_base** (G : WTZ E) (v : Rul Xo c → EReal) (h : depth_vw U G v = ⊤) (τ : ℕ) (t : Rul Xo c) : valL_vw U G (τ + 1) v t = ⊥

Development.AbelianVarietyTheory.Definitions.Divisors.NotIsComplAtPowerSeriesC

theorem **Taut.s_ne_zero** (C : Taut ζ g n φ) (k : Fin C.m ⊕ Fin C.N) : C.s k ≠ 0

theorem **Taut.t_mem_ne_zero** (C : Taut ζ g n φ) (k : Fin C.m) : C.t k ∈ maximalIdeal C.R ∧ C.t k ≠ 0

theorem **Taut.comap_maximalIdeal** (C : Taut ζ g n φ) : (maximalIdeal C.R).comap C.iw = maximalIdeal C.W

theorem **isComplAt_intCast_rat** : IsComplAt (Int.castRingHom ℚ)

theorem **not_isComplAt_powerSeries_C** : ¬ IsComplAt (PowerSeries.C (R

Development.AbelianVarietyTheory.Definitions.Divisors.NotIsConstAlongConeZero

theorem **gm_gv** {Y Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (V : Y.Opens) (h : r z ∈ V) (a : Γ(Y, ⊤)ˣ) : gm r z V (InhTate.gv a V) = gm (𝟙 Z) z ⊤ (pullTop r a)

theorem **val_const** {W Y Z : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ W)} {ε : K → ℤ} {t : LB.Gen (LB.LinComb (Inh.trivLB W) φ ε)} {a : Γ(Y, ⊤)ˣ} (h : IsConst t a) (r : Z ⟶ Y) (z : Z) (J : (Z ⟶ W) → (Inh.trivLB W).ι) : Geo.Tor.val t r z J = gm (𝟙 Z) z ⊤ (pullTop r a)

theorem **pullTop_id** {Z : Scheme.{0}} (a : Γ(Z, ⊤)ˣ) : pullTop (𝟙 Z) a = a

theorem **good_triv** {W Z : Scheme.{0}} (z : Z) : Good (Inh.trivLB W) z (fun _ => PUnit.unit)

theorem **gm_eq_of_top** {Y Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) {V : Y.Opens} (hV : V = ⊤) (a : Γ(Y, V)ˣ) : gm r z V a = gm (𝟙 Z) z ⊤ (pullTop r (res hV.ge a))

theorem **one_isGen** {W Y : Scheme.{0}} (f g : Y ⟶ W) : LB.Gen.IsGen ((fun _ => 1) : LB.Iso ((Inh.trivLB W).pull f) ((Inh.trivLB W).pull g))

theorem **constHom_apply** {R k : Type} [CommRing R] [Field k] (ρ : R →+* DualNumber k) (r : R) : constHom ρ r = TrivSqZeroExt.inl (ρ r).fst

theorem **fst_constHom** {R k : Type} [CommRing R] [Field k] (ρ : R →+* DualNumber k) : spMap (TrivSqZeroExt.fstHom k k k).toRingHom ≫ spMap ρ = spMap (TrivSqZeroExt.fstHom k k k).toRingHom ≫ spMap (constHom ρ)

theorem **isUnitTau_iff** (y z : X) (τ₀) : IsUnitTau (toricPre R X q) t gam y z τ₀ ↔ ∃ u : Rˣ, IsConst τ₀ (unitOn u) ∧ (polar q y z : FractionRing R) = (∏ i, algebraMap R (FractionRing R) (t i) ^ gam i y z) * algebraMap R _ u

theorem **isConstAlong_iff** {k : Type} [Field k] (ρ : R →+* DualNumber k) (a : Fin m → k) : IsConstAlong (toricPre R X q) t gam ρ a ↔ ∀ (y z : X) (u : Rˣ), (polar q y z : FractionRing R) = (∏ i, algebraMap R (FractionRing R) (t i) ^ gam i y z) * algebraMap R _ u → ρ u = ((1 : DualNumber k) + (TrivSqZeroExt.inr (∑ i, a i * gam i y z) : DualNumber k)) * constHom ρ u

theorem **ss_mem** : ss ∈ maximalIdeal A

theorem **isUnit_one_add** {x : A} (h : x ∈ maximalIdeal A) : IsUnit (1 + x)

theorem **frm_self** (y : ℤ × ℤ) : frm y y = 2 * nrm y

theorem **nrm_cast** (y : ℤ × ℤ) : (nrm y : ℤ) = y.1 * y.1 + y.2 * y.2

theorem **polar_per** (v : Aˣ) (b : ℤ × ℤ → ℤ) (y z : ℤ × ℤ) : (polar (per v b) y z : FractionRing A) = algebraMap A (FractionRing A) tt ^ frm y z * algebraMap A _ (v ^ (b (y + z) - b y - b z) : Aˣ)

theorem **isConstAlong_per_iff** (v : Aˣ) (b : ℤ × ℤ → ℤ) {k : Type} [Field k] (ρ : A →+* DualNumber k) (a : Fin 1 → k) : IsConstAlong (toricPre A (ℤ × ℤ) (per v b)) ![tt] ![frm] ρ a ↔ ∀ y z : ℤ × ℤ, ρ (v ^ (b (y + z) - b y - b z) : Aˣ) = ((1 : DualNumber k) + (TrivSqZeroExt.inr (a 0 * frm y z) : DualNumber k)) * constHom ρ (v ^ (b (y + z) - b y - b z) : Aˣ)

theorem **tang_tt** : tang tt = 0

theorem **tang_ss** : tang ss = TrivSqZeroExt.inr 1

theorem **tang_max** (r : A) (hr : r ∈ maximalIdeal A) : (tang r).fst = 0

theorem **not_isRigid_of** {Q : DegPre A} {gam : Fin 1 → LinearMap.BilinForm ℤ Q.X} {a : Fin 1 → ℚ} (h : IsConstAlong Q ![tt] gam tang a) : ¬ IsRigid Q ![tt] gam

theorem **isRigid_mixed** : IsRigid (toricPre A (ℤ × ℤ) (per vs fun y => y.1 * y.2)) ![tt] ![frm]

theorem **not_isRigid_blowup** : ¬ IsRigid (toricPre A (ℤ × ℤ) (per vts fun y => y.1 * y.2)) ![tt] ![frm]

theorem **tang_vs_zpow** (n : ℤ) : tang (vs ^ n : Aˣ) = 1 + TrivSqZeroExt.inr (n : ℚ)

theorem **not_isRigid_cone** : ¬ IsRigid (toricPre A (ℤ × ℤ) (per vs fun y => nrm y)) ![tt] ![frm]

theorem **not_isConstAlong_cone_zero** : ¬ IsConstAlong (toricPre A (ℤ × ℤ) (per vs fun y => nrm y)) ![tt] ![frm] tang 0

theorem **isConstAlong_const** (Q : DegPre R) (gam : Fin m → LinearMap.BilinForm ℤ Q.X) {k : Type} [Field k] (ρ : R →+* DualNumber k) (hρ : constHom ρ = ρ) : IsConstAlong Q t gam ρ 0

theorem **toroidalChartsForFan** (h : RigidForFan) : GlobalTheory.Published.ToroidalChartsForFan

theorem **isMonomial_per** (v : Aˣ) (b : ℤ × ℤ → ℤ) : IsMonomial (toricPre A (ℤ × ℤ) (per v b)) ![tt] (![frm] : Fin 1 → LinearMap.BilinForm ℤ (ℤ × ℤ))

Development.AbelianVarietyTheory.Definitions.Divisors.NotIsIsoSpMapDnFst

@[simp] theorem **ptMap_left** {C C' : Type} [CommRing C] [CommRing C'] (s : Sp C ⟶ S) (φ : C →+* C') : (ptMap s φ).left = spMap φ

theorem **FieldChart.of_isChart** (h : IsChart A M) : FieldChart A M

theorem **Milnor.id** (T : Scheme.{0}) [IsAffine T] : Milnor (𝟙 T) (𝟙 T) (𝟙 T) (𝟙 T)

theorem **Milnor.of_isPullback** {R R₁ R₂ R₀ : CommRingCat.{0}} {a : R ⟶ R₁} {b : R ⟶ R₂} {c : R₁ ⟶ R₀} {d : R₂ ⟶ R₀} (h : IsPullback a b c d) (hd : Function.Surjective d) : Milnor (Spec.map a) (Spec.map b) (Spec.map c) (Spec.map d) where

theorem **dnFst_surjective** (k : Type) [CommRing k] : Function.Surjective (dnFst k)

theorem **dnFst_not_injective** (k : Type) [CommRing k] [Nontrivial k] : ¬ Function.Injective (dnFst k)

theorem **Milnor.dualNumber** (k : Type) [CommRing k] : Milnor (Spec.map (pullback.fst (CommRingCat.ofHom (dnFst k)) (CommRingCat.ofHom (dnFst k)))) (Spec.map (pullback.snd (CommRingCat.ofHom (dnFst k)) (CommRingCat.ofHom (dnFst k)))) (spMap (dnFst k)) (spMap (dnFst k))

theorem **lift_of_chartAt** {C C' : Type} [CommRing C] [IsLocalRing C] [CommRing C'] [Nontrivial C'] (s : Sp C ⟶ S) (φ : C →+* C') (hφ : Function.Surjective φ) (N : LB (A ⊗ Over.mk s).left) (h : ChartAt A M N) (u₀ : Over.mk (spMap φ ≫ s) ⟶ U) (h₀ : Same A N M (ptMap s φ) u₀) : ∃ u : Over.mk s ⟶ U, Same A N M (𝟙 _) u

theorem **TanLift.of_isChart** (h : IsChart A M) : TanLift A M

theorem **SmallLift.of_isChart** (h : IsChart A M) : SmallLift A M

theorem **not_isIso_spMap_dnFst** (k : Type) [CommRing k] [Nontrivial k] : ¬ IsIso (spMap (dnFst k))

theorem **dnFst_comp_dnInl** (k : Type) [CommRing k] : (dnFst k).comp (dnInl k) = RingHom.id k

Development.AbelianVarietyTheory.Definitions.Divisors.NotLeafMMZero

theorem **N_eq** : d.O.N = GlobalTheory.ChiGrid.bigN D

theorem **nr_eq** : d.O.nr = GlobalTheory.ChiGrid.nRk D

theorem **s_eq** : d.O.s = GlobalTheory.ChiGrid.sLen D

theorem **PPF_eq** (C b : ℝ) : d.O.PPF C b = GlobalTheory.ChiGrid.sizeP D C b

theorem **Lc_eq** : d.O.Lc = GlobalTheory.ChiGrid.cloudLen D

theorem **ga_eq** (C : ℝ) : d.O.ga C = GlobalTheory.ChiGrid.gap D C

theorem **ee_eq** (h : Ideal.absNorm d.G.pl.v0.asIdeal = GlobalTheory.ChiGrid.q0 D) (v : HeightOneSpectrum (𝒪 D.host.F)) : d.O.ee CF p.Z.b v = d.G.pl.depth (1, CF, p.Z.b) v

theorem **square_ranks** (C b : ℝ) : (∀ ij, d.O.dd ij = 2 * d.O.hd ij) ∧ (∀ ij, d.O.dd ij ≤ 2 * d.O.N) ∧ (∀ ij, d.O.yy ij + 1 ≤ d.O.m) ∧ ∀ q ∈ d.O.S0, d.O.cc q ≤ d.O.ee C b q

theorem **yy_eq** (ij : Sq D) : d.O.yy ij = (GlobalTheory.ChiGrid.Node.grid ij.1 (ij.2 + 1)).neg D.R.m + (GlobalTheory.ChiGrid.Node.grid (ij.1 + 1) ij.2).neg D.R.m

theorem **not_leafMM_zero** (G : Grid p CF) (W : G.Weil) (h : W.Over p.R.toRealDat) (C : ℝ) : ¬ LeafMM (FunIn.ofGrid G W h 0 fun _ => 0).O C

theorem **ofGrid_lattice** (G : Grid p CF) (W : G.Weil) (h : W.Over p.R.toRealDat) (V) : (FunIn.ofGrid G W h 1 V).M.FG ∧ (FunIn.ofGrid G W h 1 V).M ≠ ⊥ ∧ ∀ v, IsUnit ((FunIn.ofGrid G W h 1 V).X' v).det

theorem **pow_le_finrank** {K : Type} [Field K] [NumberField K] {q e : ℕ} (hq : q.Prime) {z : K} (hz : IsPrimitiveRoot z (q ^ (e + 1))) : q ^ e ≤ Module.finrank ℚ K

theorem **degree_of_fit** (p : PtAt ex Csz) (CF : ℝ) (G : Grid p CF) (h : GridFit p G) (h0 : Ideal.absNorm G.pl.v0.asIdeal = GlobalTheory.ChiGrid.q0 D) : GlobalTheory.ChiGrid.q0 D ^ ⌊GlobalTheory.ChiGrid.sizeP D CF p.Z.b / Real.log (GlobalTheory.ChiGrid.q0 D)⌋₊ ≤ Module.finrank ℚ D.LL0

theorem **nonempty_iff** (p : PtAt ex Csz) (CF : ℝ) : Nonempty (FunIn p CF) ↔ ∃ (G : GlobalTheory.ChiGrid.GridData p.R.toRealDat.toLevCh (1, CF, p.Z.b)) (W : G.Weil), W.Over p.R.toRealDat

Development.AbelianVarietyTheory.Definitions.Divisors.NotOrdGeOfNotMem

theorem **OrdGe.mem_O** {T : Trait F} {ξ : Sp T.V ⟶ M.Z} {x : Ω} {a : ℕ} (h : OrdGe M T ξ x a) : x ∈ M.O ((jK T.V ≫ ξ) (IsLocalRing.closedPoint (FractionRing T.V)))

theorem **OrdEq.mem_O** {T : Trait F} {ξ : Sp T.V ⟶ M.Z} {x : Ω} {a : ℕ} (h : OrdEq M T ξ x a) : x ∈ M.O ((jK T.V ≫ ξ) (IsLocalRing.closedPoint (FractionRing T.V)))

theorem **OrdGe.mono** {T : Trait F} {ξ : Sp T.V ⟶ M.Z} {x : Ω} {a b : ℕ} (h : OrdGe M T ξ x a) (hba : b ≤ a) : OrdGe M T ξ x b

theorem **OrdGe.add** {T : Trait F} {ξ : Sp T.V ⟶ M.Z} {x y : Ω} {a : ℕ} (hx : OrdGe M T ξ x a) (hy : OrdGe M T ξ y a) : OrdGe M T ξ (x + y) a

theorem **OrdGe.mul** {T : Trait F} {ξ : Sp T.V ⟶ M.Z} {x y : Ω} {a b : ℕ} (hx : OrdGe M T ξ x a) (hy : OrdGe M T ξ y b) : OrdGe M T ξ (x * y) (a + b)

Development.AbelianVarietyTheory.Definitions.Divisors.NotSymEOfNe

theorem **ThAt.card** {P : PolAV F.X} {rig : LB.Gen (P.L.pull (eSec P.A))} {m N : ℕ} {f : F.X ⟶ PSpace (cycAway n) N} (h : ThAt F W P rig m f) : N + 1 = m ^ (4 * (8 * g))

theorem **SymE.card** {S : ThetaSys o Ω} {ρ : S.Real} {m N : ℕ} {e : Fin N → Ω} (h : SymE ρ m e) : N = m ^ (4 * S.g)

theorem **SymE.cast_ne_zero** {S : ThetaSys o Ω} {ρ : S.Real} {m N : ℕ} {e : Fin N → Ω} (h : SymE ρ m e) : (m : Ω) ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.NotTwistSumZeroFamily

theorem **twistSum_ofUMem_iff** {X₁ : Setup F W i} {z₁ : X₁.M.Z} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A ⟶ X₁.ρ.P.A)} {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] {ev : X₁.M.Z.presheaf.stalk z₁ →+* V} {X : Type} [AddCommGroup X] {m : ℕ} {t : Fin m → V} {δ : Fin m → LinearMap.BilinForm ℤ X} (m : UMem X₁ z₁ act V ev X t δ) (x₀ : X₁.S.Gam) {C : Type} (lab : C → X₁.S.Gam) (T : (Fin 8 → X) → C → V) (sg : ℤˣ) (c : C) (τ : X₁.S.Gam) (e : ℕ) : GlobalTheory.GSep.TwistSum (ofSetup X₁) z₁ _ x₀ (ofUMem m) lab T sg c τ e ↔ GlobalTheory.Poinc.Sep.TwistSum X₁ z₁ act x₀ m lab T sg c τ e

theorem **twistAtEach_siegel** {X₁ : Setup F W i} {z₁ : X₁.M.Z} {act : Matrix (Fin 8) (Fin 8) ℤ → (X₁.ρ.P.A ⟶ X₁.ρ.P.A)} {x₀ : X₁.S.Gam} {sg : ℤˣ} (h : GlobalTheory.GSep.TwistAtEach (I

theorem **exists_germ** {a : X₁.Ω₁} (ha : a ∈ X₁.M.O z₁) : ∃ s : X₁.M.Z.presheaf.stalk z₁, X₁.M.stalkHom z₁ s = a

theorem **exists_germs** {x₀ : X₁.S.Gam} (hd : X₁.M.Div X₁.S.th x₀ z₁) (b : X₁.S.Gam) (e : ℕ) : ∃ zt h : X₁.M.Z.presheaf.stalk z₁, X₁.M.stalkHom z₁ zt = algebraMap X₁.o₁ X₁.Ω₁ (X₁.S.ζ ^ (X₁.S.r * X₁.S.N)) ∧ X₁.M.stalkHom z₁ h = algebraMap X₁.o₁ X₁.Ω₁ (X₁.S.ζ ^ e) * X₁.S.th b / X₁.S.th x₀

theorem **exists_yy** (mu : X) : ∃ a : X₁.S.Gam, m.YY a mu

theorem **yK_th_ne_zero** {x₀ : X₁.S.Gam} (hd : X₁.M.Div X₁.S.th x₀ z₁) : m.yK (X₁.S.th x₀, X₁.S.th_mem x₀) ≠ 0

theorem **val_stalkHom** [IsLocalHom ev] (a : X₁.M.Z.presheaf.stalk z₁) : X₁.M.val (jK V ≫ spMap ev ≫ X₁.M.Z.fromSpecStalk z₁) (X₁.M.stalkHom z₁ a) = algebraMap V (FractionRing V) (ev a)

theorem **wt_zero** (sg : ℤˣ) (wt : X → ZMod (X₁.S.r * X₁.S.N)) (hwt : ∀ a mu, m.YY a mu → wt mu = (sg : ℤ) • eStd X₁.S.g _ (0 : X₁.S.Gam) a) (mu : X) : wt mu = 0

theorem **twistSum_zero** {x₀ : X₁.S.Gam} (hd : X₁.M.Div X₁.S.th x₀ z₁) {C : Type} {lab : C → X₁.S.Gam} {T : X → C → V} (hT : m.Terms x₀ lab T) (sg : ℤˣ) (c : C) : GlobalTheory.GSep.TwistSum X₁ z₁ act x₀ m lab T sg c 0 0

theorem **not_twistSum_zero_family** {x₀ : X₁.S.Gam} (hd : X₁.M.Div X₁.S.th x₀ z₁) (sg : ℤˣ) (e : ℕ) : ¬ GlobalTheory.GSep.TwistSum X₁ z₁ act x₀ m (fun _ : Unit => x₀) (fun _ _ => 0) sg () 0 e

Development.AbelianVarietyTheory.Definitions.Divisors.OfCore

theorem **Core5.of_char** {R : RealDat D A Ks} {F : Face D A} (h : R.Char F) : Core5 R F

theorem **Core5.of_core** {R : RealDat D A Ks} {F : Face D A} (h : GlobalTheory.Hk3.Core R F) : Core5 R F

Development.AbelianVarietyTheory.Definitions.Divisors.OfDat

theorem **TiesA.of_dat** (z : Dat X I hX Tw Wt) (ok : (famR X I hX (usedOf I Tw Wt) z.Xg z.tr).fam3.Ok3 1) (hex : z.R.toRealDat.ExAt) (host : Formalism.IndefR.Links.HostHyp X.toD7) (par : ∃ L : SizeData₂ X.toD7, Nonempty (ParameterChoice' X.toD7 L)) (ty : GlobalTheory.Mem.Top.TypesTrue X.toD7.ntypes z.R.C.P z.R.C.W) (ser : SerClause X.toD7) (w1 : ∀ w, (z.R.toCx3 z.F).W1 z.F.cr (z.R.num z.F 0 0) w → w ∈ z.R.twPl) (nV : 8 * X.toD7.R.f * X.toD7.R.r + 4 ≤ X.toD7.R.nV) : TiesA z.R z.F

theorem **TiesG.of_dat** (z : Dat X I hX Tw Wt) (hU : z.R.toRealDat.LUnif) (hI : IoExt z.R.toRealDat) (hM : ∀ i, OneMapRowR z.core z.R.toRealDat z.geo.κ z.geo.v₀ z.geo.O (fun _ => 1) i) (hC : GlobalTheory.TwistField.CoverField z.core z.R.toRealDat) : TiesG z.core z.R z.geo.κ z.geo.v₀ z.geo.O

Development.AbelianVarietyTheory.Definitions.Divisors.OfFactorE

theorem **Canon.isLB** [IsLocalRing R] {Q : Data R} {G : Over (Sp R)} [GrpObj G] [IsCommMonObj G] {P : LB (G2 G).left} (h : Canon Q G P) : P.IsLB

theorem **PFactor.ofFactor_e** {D D' : DegPre R} (F : Factor D D') (hF : F.Alg) (hB : D.TauBimult) : (PFactor.ofFactor F hF hB).e = F.e ∧ (PFactor.ofFactor F hF hB).f = F.f

theorem **Wit.pfac_e** : Wit.pfac.e ≠ 0 ∧ Wit.pfac.e ≠ 1

Development.AbelianVarietyTheory.Definitions.Divisors.OmAUnipFourierInv

theorem **isPrimFrame_mul** {Λ Λ' : Submodule 𝒪[k] (D → k)} {X : Matrix A D k} (hX : IsPrimFrame Λ X) (g : GL D k) (hg : ∀ x, x ᵥ* g.val ∈ Λ' ↔ x ∈ Λ) : IsPrimFrame Λ' (X * g.val)

theorem **val_le_iff_mem** {ϖ : kˣ} (hϖ : IsUnif ϖ) (e : ℕ) (r : 𝒪[k]) : r ∈ IsLocalRing.maximalIdeal 𝒪[k] ^ e ↔ valuation k (r : k) ≤ valuation k ((ϖ ^ e : kˣ) : k)

theorem **sum_psi** {ϖ : kˣ} (hϖ : IsUnif ϖ) {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (e : ℕ) (s s' : Res k e) : ∑ t : Res k e, (ψ (((Quotient.out t : 𝒪[k]) : k) * ((((ϖ ^ e)⁻¹ : kˣ) : k) * (((Quotient.out s : 𝒪[k]) : k) - ((Quotient.out s' : 𝒪[k]) : k)))) : ℂ) = if s = s' then (Fintype.card (Res k e) : ℂ) else 0

theorem **sum_invFourierSum** {ϖ : kˣ} (hϖ : IsUnif ϖ) {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (e : ℕ) (P : Finset D) (Ψ : Matrix A D k → ℂ) (X : Matrix A D k) (S₀ : A → P → Res k e) : ∑ T : A → P → Res k e, invFourierSum ψ ϖ e P Ψ (setCols P X (liftRes T)) * (ψ (pairCols P (setCols P X ((ϖ ^ e)⁻¹ • liftRes S₀)) (liftRes T)) : ℂ) = (Fintype.card (Res k e) : ℂ) ^ (Fintype.card A * P.card) * Ψ (setCols P X ((ϖ ^ e)⁻¹ • liftRes S₀))

theorem **squareFlip_ne_zero** [IsEmpty A'] {ϖ : kˣ} (hϖ : IsUnif ϖ) {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {e : ℕ} (he : 1 ≤ e) (σ : A ⊕ A' ≃ D) (ν : kˣ →* ℂˣ) (P : Finset D) (Φ : Matrix A' D k → ℂ) (hΦ : Φ 0 ≠ 0) : squareFlip (haarVol (KO k A')) σ ψ ν ϖ e P Φ ≠ 0

theorem **exists_flip_ne_zero** (t s : ℕ) (ht : Odd t) (hs : Odd s) (a : Assignment) : ∃ (G : Grid k) (ϖ : kˣ) (ψ : AddChar k Circle) (S : G.PiLocalization₂), G.m = 1 ∧ G.t = t ∧ G.s = s ∧ G.IsGood ϖ ψ 2 ∧ S.path = givenPath 1 ∧ S.IsAdmissible a ϖ 2 ∧ PassesThrough 0 0 S.path ∧ (S.flip a ψ ϖ 2 (0, 0)).path = oppositePath 1 ∧ (S.flip a ψ ϖ 2 (0, 0)).tensor ≠ 0

theorem **gdet_unip** (b : Matrix D₁ D₂ k) : gdet (unip b) = 1

theorem **pair_two** (Y : Matrix A (D₁ ⊕ D₂) k) (T : Matrix A (Assignment.two.block D₁ D₂) k) : pairCols _ Y T = ∑ i, ∑ j : D₂, setCols _ (0 : Matrix A (D₁ ⊕ D₂) k) T i (.inr j) * Y i (.inr j)

theorem **pair_one** (Y : Matrix A (D₁ ⊕ D₂) k) (T : Matrix A (Assignment.one.block D₁ D₂) k) : pairCols _ Y T = ∑ i, ∑ j : D₁, setCols _ (0 : Matrix A (D₁ ⊕ D₂) k) T i (.inl j) * Y i (.inl j)

theorem **omB_unip_fourier** (ψ : AddChar k Circle) (m : Measure k) (ν α : kˣ →* ℂˣ) (b : Matrix D₁ D₂ k) (Φ : Matrix A (D₁ ⊕ D₂) k → ℂ) : omB ν α 1 (unip b) (fourierInt ψ m (Assignment.two.block D₁ D₂) Φ) = fourierInt ψ m (Assignment.two.block D₁ D₂) fun T => (ψ (trace (T.toCols₁ * b * T.toCols₂ᵀ)) : ℂ) * Φ T

theorem **omA_unip_fourier** (ψ : AddChar k Circle) (m : Measure k) (ν β : kˣ →* ℂˣ) (b : Matrix D₁ D₂ k) (Φ : Matrix A (D₁ ⊕ D₂) k → ℂ) : omA ν β 1 (unip b) (fourierInt ψ m (Assignment.one.block D₁ D₂) Φ) = fourierInt ψ m (Assignment.one.block D₁ D₂) fun T => (ψ (-trace

(T.toCols₁ ∗ b ∗ T.toCols₂ᵀ)) : ℂ) ∗ Φ T

theorem **omA_unip_fourier_inv** (ψ : AddChar k Circle) (m : Measure k) (ν β : kˣ →∗ ℂˣ) (b : Matrix D₁ D₂ k) (Φ : Matrix A (D₁ ⊕ D₂) k → ℂ) : omA ν β 1 (unip b) (fourierInt ψ⁻¹ m (Assignment.one.block D₁ D₂) Φ) = fourierInt ψ⁻¹ m (Assignment.one.block D₁ D₂) fun T => (ψ (trace (T.toCols₁ ∗ b ∗ T.toCols₂ᵀ)) : ℂ) ∗ Φ T

Development.AbelianVarietyTheory.Definitions.Divisors.OneFormMapEqKaehlerDifferentialMap

@[simp] theorem **sgn_ι** (ω : Ω[B⁄R]) : sgn (ι1 ω) = −ι1 ω

@[simp] theorem **sgn_sgn** (x : DR R B) : sgn (sgn x) = x

theorem **sgn_smul** (r : R) (x : DR R B) : sgn (r • x) = r • sgn x

theorem **ι_mul_ι_swap** (a b : Ω[B⁄R]) : ι1 a ∗ ι1 b = −(ι1 b ∗ ι1 a)

theorem **ι_mul** (ω : Ω[B⁄R]) (x : DR R B) : ι1 ω ∗ x = sgn x ∗ ι1 ω

theorem **ind1** {P : Ω[B⁄R] → Prop} (h : ∀ b c : B, P (b • D R B c)) (h0 : P 0) (ha : ∀ x y, P x → P y → P (x + y)) (ω : Ω[B⁄R]) : P ω

theorem **indDR** {P : DR R B → Prop} (h : ∀ b : B, P (algebraMap B _ b)) (hd : ∀ b : B, P (ι1 (D R B b))) (hm : ∀ x y, P x → P y → P (x ∗ y)) (ha : ∀ x y, P x → P y → P (x + y)) (x : DR R B) : P x

@[simp] theorem **mk_1** (x y : DR R B) : (mk x y).1 = x

@[simp] theorem **mk_2** (x y : DR R B) : (mk x y).2 = y

@[ext] theorem **ext'** {a b : Tw R B} (h1 : a.1 = b.1) (h2 : a.2 = b.2) : a = b

@[simp] theorem **mul_1** (a b : Tw R B) : (a ∗ b).1 = a.1 ∗ b.1

@[simp] theorem **mul_2** (a b : Tw R B) : (a ∗ b).2 = sgn a.1 ∗ b.2 + a.2 ∗ b.1

@[simp] theorem **add_1** (a b : Tw R B) : (a + b).1 = a.1 + b.1

@[simp] theorem **add_2** (a b : Tw R B) : (a + b).2 = a.2 + b.2

@[simp] theorem **zero_1** : (0 : Tw R B).1 = 0

@[simp] theorem **zero_2** : (0 : Tw R B).2 = 0

@[simp] theorem **one_1** : (1 : Tw R B).1 = 1

@[simp] theorem **one_2** : (1 : Tw R B).2 = 0

@[simp] theorem **ofFun_1** (b : B) : (ofFun (R

@[simp] theorem **ofFun_2** (b : B) : (ofFun (R

theorem **smul_def'** (b : B) (x : Tw R B) : b • x = ofFun b ∗ x

@[simp] theorem **smul_1** (r : R) (a : Tw R B) : (r • a).1 = r • a.1

@[simp] theorem **smul_2** (r : R) (a : Tw R B) : (r • a).2 = r • a.2

theorem **der_apply** (b : B) : der (R

theorem **lift_anti** (a b : Ω[B⁄R]) : der.liftKaehlerDifferential a ∗ der.liftKaehlerDifferential b + der.liftKaehlerDifferential b ∗ der.liftKaehlerDifferential a = (0 : Tw R B)

theorem **lift_sq** (a : Ω[B⁄R]) : der.liftKaehlerDifferential a ∗ der.liftKaehlerDifferential a = (0 : Tw R B)

theorem **graph_D** (b : B) : graph (ι1 (D R B b)) = mk (ι1 (D R B b)) 0

theorem **graph_1** (x : DR R B) : (graph x).1 = x

@[simp] theorem **dDR_algebraMap** (b : B) : dDR R B (algebraMap B _ b) = ι1 (D R B b)

@[simp] theorem **dDR_D** (b : B) : dDR R B (ι1 (D R B b)) = 0

theorem **dDR_mul** (x y : DR R B) : dDR R B (x ∗ y) = sgn x ∗ dDR R B y + dDR R B x ∗ y

theorem **dDR_smul** (b : B) (x : DR R B) : dDR R B (b • x) = ι1 (D R B b) ∗ x + b • dDR R B x

theorem **dDR_sgn** (x : DR R B) : dDR R B (sgn x) = −sgn (dDR R B x)

theorem **dDR_dDR** (x : DR R B) : dDR R B (dDR R B x) = 0

theorem **ι_mem** (ω : Ω[B⁄R]) : ι1 ω ∈ ∧[B]^1 Ω[B⁄R]

theorem **mul_mem** {i j : ℕ} {x y : DR R B} (hx : x ∈ ∧[B]^i Ω[B⁄R]) (hy : y ∈ ∧[B]^j Ω[B⁄R]) : x ∗ y ∈ ∧[B]^(i + j) Ω[B⁄R]

theorem **dDR_ι_mem** (ω : Ω[B⁄R]) : dDR R B (ι1 ω) ∈ ∧[B]^2 Ω[B⁄R]

theorem **dDR_mem** {j : ℕ} {x : DR R B} (hx : x ∈ ∧[B]^j Ω[B⁄R]) : dDR R B x ∈ ∧[B]^(j + 1) Ω[B⁄R]

theorem **sgn_of_mem** {j : ℕ} {x : DR R B} (hx : x ∈ ∧[B]^j Ω[B⁄R]) : sgn x = (−1) ^ j ∗ x

@[simp] theorem **coe_dOm** {j : ℕ} (x : Om R B j) : (dOm R B j x : DR R B) = dDR R B x

@[simp] theorem **dOm_dOm** {j : ℕ} (x : Om R B j) : dOm R B (j + 1) (dOm R B j x) = 0

@[simp] theorem **coe_dOne** (ω : Ω[B⁄R]) : (dOne R B ω : DR R B) = dDR R B (ι1 ω)

@[simp] theorem **coe_dTwo** (x : Om R B 2) : (dTwo R B x : DR R B) = dDR R B x

theorem **dOne_eq_dOm** (ω : Ω[B⁄R]) : dOne R B ω = dOm R B 1 ⟨ι1 ω, ι_mem ω⟩

theorem **dOm_zero** (b : B) (h : algebraMap B (DR R B) b ∈ ∧[B]^0 Ω[B⁄R]) : dOm R B 0 ⟨_, h⟩ = ⟨ι1 (D R B b), ι_mem _⟩

@[simp] theorem **coe_wedge** (a b : Ω[B⁄R]) : (wedge R B a b : DR R B) = ι1 a ∗ ι1 b

theorem **wedge_eq_ιMulti** (a b : Ω[B⁄R]) : wedge R B a b = exteriorPower.ιMulti B 2 ![a, b]

@[simp] theorem **wedge_self** (a : Ω[B⁄R]) : wedge R B a a = 0

theorem **wedge_swap** (a b : Ω[B⁄R]) : wedge R B a b = −wedge R B b a

theorem **wedge_smul_left** (r : R) (a b : Ω[B⁄R]) : wedge R B (r • a) b = r • wedge R B a b

theorem **wedge_smul_right** (r : R) (a b : Ω[B⁄R]) : wedge R B a (r • b) = r • wedge R B a b

@[simp] theorem **dOne_D** (b : B) : dOne R B (D R B b) = 0

@[simp] theorem **dTwo_dOne** (ω : Ω[B⁄R]) : dTwo R B (dOne R B ω) = 0

theorem **dOne_smul** (b : B) (ω : Ω[B⁄R]) : dOne R B (b • ω) = wedge R B (D R B b) ω + b • dOne R B ω

theorem **dOne_smul_D** (b c : B) : dOne R B (b • D R B c) = wedge R B (D R B b) (D R B c)

theorem **dTwo_wedge** {a b : Ω[B⁄R]} (ha : dOne R B a = 0) (hb : dOne R B b = 0) : dTwo R B (wedge R B a b) = 0

theorem **SemiHom.map_algebraMap** (h : SemiHom σ α) (r : R) : α (algebraMap R B r) = algebraMap R' B' (σ r)

theorem **SemiHom.of_algHom** [Algebra R B'] (φ : B →ₐ[R] B') : SemiHom (RingHom.id R) (φ : B →+* B')

theorem **SemiHom.m1'_D** (h : SemiHom σ α) (b : B) : h.m1' (D R B b) = D R' B' (α b)

@[simp] theorem **SemiHom.m1_D** (h : SemiHom σ α) (b : B) : h.m1 (D R B b) = D R' B' (α b)

@[simp] theorem **SemiHom.oneFormMap_smul_semilinear** (h : SemiHom σ α) (b : B) (ω : Ω[B⁄R]) : h.m1 (b • ω) = α b • h.m1 ω

@[simp] theorem **SemiHom.mDR_ι** (h : SemiHom σ α) (ω : Ω[B⁄R]) : h.mDR (ι1 ω) = ι1 (h.m1 ω)

@[simp] theorem **SemiHom.mDR_algebraMap** (h : SemiHom σ α) (b : B) : h.mDR (algebraMap B _ b) = algebraMap B' _ (α b)

theorem **SemiHom.mDR_smul** (h : SemiHom σ α) (b : B) (x : DR R B) : h.mDR (b • x) = α b • h.mDR x

theorem **SemiHom.mDR_sgn** (h : SemiHom σ α) (x : DR R B) : h.mDR (sgn x) = sgn (h.mDR x)

theorem **SemiHom.mDR_dDR** (h : SemiHom σ α) (x : DR R B) : h.mDR (dDR R B x) = dDR R' B' (h.mDR x)

theorem **SemiHom.mDR_mem** (h : SemiHom σ α) {j : ℕ} {x : DR R B} (hx : x ∈ ∧[B]^j Ω[B⁄R]) : h.mDR x ∈ ∧[B']^j Ω[B'⁄R']

@[simp] theorem **SemiHom.coe_mOm** (h : SemiHom σ α) {j : ℕ} (x : Om R B j) : (h.mOm j x : DR R' B') = h.mDR x

theorem **SemiHom.mOm_smul** (h : SemiHom σ α) {j : ℕ} (b : B) (x : Om R B j) : h.mOm j (b • x) = α b • h.mOm j x

theorem **SemiHom.mOm_dOm** (h : SemiHom σ α) {j : ℕ} (x : Om R B j) : h.mOm (j + 1) (dOm R B j x) = dOm R' B' j (h.mOm j x)

theorem **SemiHom.mOm_dOne** (h : SemiHom σ α) (ω : Ω[B⁄R]) : h.mOm 2 (dOne R B ω) = dOne R' B' (h.m1 ω)

theorem **SemiHom.mOm_dTwo** (h : SemiHom σ α) (x : Om R B 2) : h.mOm 3 (dTwo R B x) = dTwo R' B' (h.mOm 2 x)

theorem **SemiHom.mOm_wedge** (h : SemiHom σ α) (a b : Ω[B⁄R]) : h.mOm 2 (wedge R B a b) = wedge R' B' (h.m1 a) (h.m1 b)

theorem **SemiHom.oneFormMap_eq_self_of_id** {τ : R →+* R} {β : B →+* B} (h : SemiHom τ β) (e : ∀ b, β b = b) (ω : Ω[B⁄R]) : h.m1 ω = ω

theorem **SemiHom.mOm_id** {τ : R →+* R} {β : B →+* B} (h : SemiHom τ β) (e : ∀ b, β b = b) {j : ℕ} (x : Om R B j) : h.mOm j x = x

theorem **SemiHom.comp** (h : SemiHom σ α) (h' : SemiHom τ β) : SemiHom (τ.comp σ) (β.comp α)

theorem **SemiHom.oneFormMap_comp_oneFormMap** (h : SemiHom σ α) (h' : SemiHom τ β) (h'' : SemiHom ρ γ) (e : ∀ b, β (α b) = γ b) (ω : Ω[B⁄R]) : h'.m1 (h.m1 ω) = h''.m1 ω

theorem **SemiHom.mOm_mOm** (h : SemiHom σ α) (h' : SemiHom τ β) (h'' : SemiHom ρ γ) (e : ∀ b, β (α b) = γ b) {j : ℕ} (x : Om R B j) : h'.mOm j (h.mOm j x) = h''.mOm j x

theorem **SemiHom.oneFormMap_eq_kaehlerDifferential_map** [Algebra R B'] [Algebra B B'] [IsScalarTower R B B'] {τ : R →+* R} {β : B →+* B'} (h : SemiHom τ β) (e : ∀ b, β b = algebraMap B B' b) (ω : Ω[B⁄R]) : h.m1 ω = KaehlerDifferential.map R R B B' ω

Development.AbelianVarietyTheory.Definitions.Divisors.OneMemH0WeightZero

theorem **nrm_mul** {R : Type*} [CommRing R] {a b : R} (x y : ℍ[R, a, b]) : nrm (x * y) = nrm x * nrm y

theorem **mul_star_eq** {R : Type*} [CommRing R] {a b : R} (x : ℍ[R, a, b]) : x * star x = algebraMap R ℍ[R, a, b] (nrm x)

theorem **nrm_rat** {a b : F} (x : ℍ[F, a, b]) : nrm (rat x) = algebraMap F (AF F) (nrm x)

theorem **one_mem_H0_weight_zero** {a b : F} (ε : AF F) : (1 : DA F a b) ∈ H0 ε

Development.AbelianVarietyTheory.Definitions.Divisors.OrdGeOfForall

theorem **qW_nonneg** {s : ℕ} {W : Matrix (Fin s) (Fin s) ℤ} (hW : PosW W) {X : Type} [AddCommGroup X] {β : LinearMap.BilinForm ℤ X} (hs : ∀ a b, β a b = β b a) (hβ : ∀ y, 0 ≤ β y y) (y : Fin s → X) : 0 ≤ qW W β y

theorem **lin_nonneg** {s l : ℕ} {W : Matrix (Fin s) (Fin s) ℤ} (hW : PosW W) {X : Type} [AddCommGroup X] {β : LinearMap.BilinForm ℤ X} (hs : ∀ a b, β a b = β b a) (hβ : ∀ y, 0 ≤ β y y) (a : Fin l × Fin s → X) : 0 ≤ lin W a β

theorem **qW_nonneg** (y : Φ.Y) (i : Fin (Φ.m y)) (μ : Fin s → Φ.X y) : 0 ≤ qW W (Φ.δ y i) μ

theorem **lin_nonneg** (y : Φ.Y) (i : Fin (Φ.m y)) {l : ℕ} (a : Fin l × Fin s → Φ.X y) : 0 ≤ lin W a (Φ.δ y i)

theorem **monoT_eq** (y : Φ.Y) {l : ℕ} (a : Fin l × Fin s → Φ.X y) : Φ.monoT y a = ∏ j, Φ.t y j ^ (lin W a (Φ.δ y j)).toNat

theorem **OrdGe.mem** {φ : InvFun O} {d e : ℕ} {f : A} (h : Φ.OrdGe φ d e f) (y : Φ.Y) (hP : IsPrincipalIdealRing (Φ.V y)) (hm : Φ.m y = 1) (i : Fin (Φ.m y)) (b : ℕ) (hb : (b : ℝ) = d * φ.evalZ (Φ.δ y i)) : Φ.ev y f ∈ Ideal.span {Φ.t y i ^ b} ⊔ ((maximalIdeal A).map (Φ.ev y)) ^ e

theorem **ordGe_of_forall** {φ : InvFun O} {d e : ℕ} {f : A} (h : ∀ y, IsPrincipalIdealRing (Φ.V y) → Φ.m y = 1 → ∀ (i : Fin (Φ.m y)) (b : ℕ), (∀ n : ℕ, d * φ.evalZ (Φ.δ y i) ≤ n → b ≤ n) → Φ.ev y f ∈ Ideal.span {Φ.t y i ^ b} ⊔ ((maximalIdeal A).map (Φ.ev y)) ^ e) : Φ.OrdGe φ d e f

Development.AbelianVarietyTheory.Definitions.Divisors.OrthSubrepresentation

theorem **Kills.killsIrr** {G : Type} [Group G] [Finite G] {φ Y : G → ℂ} (h : Kills φ Y) : KillsIrr φ Y

theorem **character_conjRep** {G : Type} [Group G] [Finite G] {n : ℕ} (σ : Representation ℂ G (Fin n → ℂ)) (g : G) : (conjRep σ).character g = star (σ.character g)

theorem **orth_subrepresentation** {G : Type} [Group G] [Finite G] {Y : G → ℂ} (hY : GlobalTheory.BI.IsChar ℂ ℂ Y) {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (ρ : Representation ℂ G V) (h : ∑ᶠ g, star (Y g) * LinearMap.trace ℂ V (ρ g) = 0) (A : Subrepresentation ρ) : ∑ᶠ g, star (Y g) * LinearMap.trace ℂ A.toSubmodule (A.toRepresentation g) = 0

theorem **exists_irreducible_le** {G : Type} [Group G] {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (ρ : Representation ℂ G V) (C : Subrepresentation ρ) (hC : C.toSubmodule ≠ ⊥) : ∃ A : Subrepresentation ρ, A.toSubmodule ≠ ⊥ ∧ A.toSubmodule ≤ C.toSubmodule ∧ A.toRepresentation.IsIrreducible

theorem **kills_iff_killsIrr** {G : Type} [Group G] [Finite G] (φ : G → ℂ) {Y : G → ℂ} (hY : GlobalTheory.BI.IsChar ℂ ℂ Y) : Kills φ Y ↔ KillsIrr φ Y

theorem **J_nil** (θ : AlgebraicClosure F →* ℂ) (n : ℕ) : J θ [] n = fun _ => 1

theorem **J_cons** (θ : AlgebraicClosure F →* ℂ) (p : ℕ × ℕ) (L : List (ℕ × ℕ)) (n : ℕ) : J θ (p :: L) n = hc p.1 (deg L) (cusp θ p.1 p.2) (J θ L (deg L)) n

theorem **galAct_apply** (n : ℕ) (t : F' ≃ₐ[F] F') (x : GL (Fin n) F') : galAct F F' n t x = gal (t : F' →+* F') n x

theorem **isNorm_one** {F F' : Type} [Field F] [Field F'] [Algebra F F'] (s : F' ≃ₐ[F] F') (n : ℕ) : IsNorm s (1 : GL (Fin n) F') (1 : GL (Fin n) F)

Development.AbelianVarietyTheory.Definitions.Divisors.PadicActionConstInjective

theorem **height_eq_of_iso** [Fact p.Prime] [Nontrivial O] {G H : PDivGroup p O} (e : G ≅ H) : G.height = H.height

theorem **not_iso_mu_const** [Fact p.Prime] (hp : ¬ IsUnit (p : O)) : IsEmpty (mu p O ≅ const p O)

theorem **padicAction_const_injective** [Fact p.Prime] [Nontrivial O] : Function.Injective (padicAction (const p O))

theorem **level_zero** (G : PDivGroup p O) (a : G.A 0) : a = algebraMap O _ (counit a)

theorem **_root_.GlobalTheory.PDivGroup.Test.mulP_injective** (G : _root_.GlobalTheory.PDivGroup p O) (n : ℕ) : Function.Injective (G.mulP n)

theorem **mu_tate_char_p** [Fact p.Prime] (k : Type) [Field k] [CharP k p] (x : (mu p k).Tate) : x = 0

theorem **omega_eq_zero_of_isUnit** (G : PDivGroup p O) (hp : IsUnit (p : O)) (w : G.omega) : w = 0

theorem **hodgeTate_values** [Fact p.Prime] (hp : ¬ IsUnit (p : O)) : (∀ t, hodgeTate (mu p O) t = 0) ∧ hodgeTate (const p O) (constGen p O) ≠ 0

theorem **tate_torsionFree** [Fact p.Prime] (G : PDivGroup p O) {B : Type} [CommRing B] [Algebra O B] (x : G.TateOver B) (h : (p : ℤ_[p]) • x = 0) : x = 0

theorem **inl_apply** (a : A) : inl (O

theorem **inr_apply** (b : B) : inr (O

theorem **pts_ext** {E : Type} [CommRing E] [Algebra O E] {x y : WithConv (A ⊗[O] B →ₐ[O] E)} (h1 : pointsMap E (inl A B) x = pointsMap E (inl A B) y) (h2 : pointsMap E (inr A B) x = pointsMap E (inr A B) y) : x = y

theorem **swap_mul_swap** (G : PDivGroup p O) : swap G ∗ swap G = 1

theorem **incl_fst** (G H : PDivGroup p O) : incl G H ≫ fst G H = 𝟙 G

theorem **incl_swap_fst** (G : PDivGroup p O) : incl G G ≫ swap G ≫ fst G G = 0

theorem **swap_ne_scalar** [Fact p.Prime] [Nontrivial O] (a : ℤ_[p]) : swapAction (const p O) Polynomial.X ≠ padicAction _ a

Development.AbelianVarietyTheory.Definitions.Divisors.PairsOfSquareTriple

theorem **pairV** {a a' : ℕ} {G : Geo σ a a'} {om : Rep σ G.V G.W} {om' : Rep σ G.V' G.W} (h : IsTriple B G X om om') : IsPair B G.M X.β (X.β / X.α) om (a − G.p) G.p (a + a' − (G.p + G.q)) (G.p + G.q)

theorem **pairV'** {a a' : ℕ} {G : Geo σ a a'} {om : Rep σ G.V G.W} {om' : Rep σ G.V' G.W} (h : IsTriple B G X om om') : IsPair B G.M' X.α (X.α / X.β) om' G.q (a' − G.q) (a + a' − (G.p + G.q)) (G.p + G.q)

theorem **two_le_rank** {a a' : ℕ} {G : Geo σ a a'} {om : Rep σ G.V G.W} {om' : Rep σ G.V' G.W} (h : IsTriple B G X om om') : 2 ≤ a' ∧ a' < a

theorem **conj_conj** (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

Development.AbelianVarietyTheory.Definitions.Divisors.Params

theorem **TiesA.toB** {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] {R : RealDat3 D A3 Ks} {F : Face3 D A3} (h : TiesA R F) : TiesB R F

Development.AbelianVarietyTheory.Definitions.Divisors.PerCEmb

theorem **perC_emb** {E : Type} [Field E] [NumberField E] [IsCMField E] (X : Data E r) (x : Frame X.Ψ r) (v : Fin r → E) : perC X x (fun τ => Data.emb τ.1 v) = per X x v

Development.AbelianVarietyTheory.Definitions.Divisors.PointsNonemptyUnitaryRankOne

theorem **triv_norm** (G : HTGroup p) (R : Type) [CommRing R] [Algebra ℚ R] (y : G.Cˣ) : Triv G R (y ∗ sh G y) (Units.map (inC G R).toMonoidHom y⁻¹) 1

theorem **herm_one** (G : HTGroup p) (v : InfinitePlace G.F) : herm G v 1 = G.gram v

theorem **nonempty_posV_of_mark** {G : HTGroup p} (P : Pair G) (μ : ℂ) (h : P.Mark μ) : ∃ v, sigPosH (G.gram_herm v) = 1 ∧ Nonempty (PosV G v P.β)

theorem **fixL_one** (G : HTGroup p) (v : InfinitePlace G.F) (β : G.Cˣ) (z : Fin G.n → ℂ) : FixL G v β 1 z

theorem **same_refl** (G : HTGroup p) (v : InfinitePlace G.F) (β : G.Cˣ) (x : (AfP p ⊗[ℚ] G.C)ˣ) (t : (AfP p)ˣ) (hx : Triv G (AfP p) β x t) (Kf : Subgroup G.GAf) (K' : Subgroup (Gr G 1)) (a : Dat G v β) : Same G v β x t hx Kf K' a a

theorem **points_nonempty_unitaryRankOne** : ∃ v, sigPosH ((ex 1 one_ne_zero).gram_herm v) = 1 ∧ ∀ x t hx Kf K', Nonempty (Pts (ex 1 one_ne_zero) v exPair.β x t hx Kf K')

theorem **finitePlacesAway_nonempty** (p : ℕ) [Fact p.Prime] : Nonempty (AwayP p)

theorem **regAf_of_comm** (G : HTGroup p) (hc : ∀ a b : G.C, a ∗ b = b ∗ a) (g : G.GAf) : RegAf G g

theorem **sigG_one** (G : HTGroup p) (g : Gr G 1) : sigG G 1 g = g

theorem **hyps_ex** : IsSimple (ex 1 one_ne_zero) ∧ (∃ v, sigPosH ((ex 1 one_ne_zero).gram_herm v) = 1) ∧ ∃ (L : (ex 1 one_ne_zero).Level) (f : (ex 1 one_ne_zero).GAf → ℂ) (K' : Subgroup (Gr (ex 1 one_ne_zero) 1)) (f' : Gr (ex 1 one_ne_zero) 1 → ℂ), f 1 = 1 ∧ f' 1 = 1 ∧ IsTest (MonoidHom.id _) L.K.toSubgroup L.K.toSubgroup f ∧ (∀ g, f g ≠ 0 → RegAf (ex 1 one_ne_zero) g) ∧ IsTest (sigG (ex 1 one_ne_zero) 1) (KHT 5 (ex 1 one_ne_zero).m (ex 1 one_ne_zero).n 1 (ex 1 one_ne_zero).Fw) K' f'

Development.AbelianVarietyTheory.Definitions.Divisors.PolyBetween

theorem **HalfSplit.symm** {g g' : ℕ} {π π' : 𝒪 E} (h : HalfSplit l g g' π π') : HalfSplit l g' g π' π

theorem **PolyBetween.mono** {la : PointsOfFlagModel.abelian₂ p c} {n : ℕ} {P : MvPolynomial (LtN Dw n) c.F} {A B A' B' : ℝ} (h : PolyBetween Xo la P A B) (hA : A' ≤ A) (hB : B ≤ B') : PolyBetween Xo la P A' B'

theorem **JetNum.mono** {D : D7} {dF : ℕ} {cst cst' : ℝ} (h : JetNum D dF cst) (hc : cst ≤ cst') : JetNum D dF cst'

Development.AbelianVarietyTheory.Definitions.Divisors.PreUMono

theorem **preU_mono** : Monotone (preU φ)

Development.AbelianVarietyTheory.Definitions.Divisors.PrimPullTensorPull

@[simp] theorem **comp_zero** (X : DRChain.{u} C n) : X.comp 0 = LinearMap.id

@[simp] theorem **comp_succ** (X : DRChain.{u} C n) (i : Fin n) : X.comp i.succ = X.f i ∘ₗ X.comp i.castSucc

theorem **comp_eq_chainComp** (X : DRChain.{0} C n) : X.comp = chainComp X.f

theorem **primPull_comp** (f : B ⟶ B') (g : B' ⟶ B'') [IsMonHom f] [IsMonHom g] : primPull (dp0 C) (𝟙 _) (f ≫ g) = primPull (dp0 C) (𝟙 _) f ∘ₗ primPull (dp0 C) (𝟙 _) g

theorem **primPull_congr** {f g : B ⟶ B'} [IsMonHom f] [IsMonHom g] (e : f = g) : primPull (dp0 C) (𝟙 _) f = primPull (dp0 C) (𝟙 _) g

theorem **primPull_mul** (f g : B ⟶ B') [IsMonHom f] [IsMonHom g] [IsMonHom (f * g)] : primPull (dp0 C) (𝟙 _) (f * g) = primPull (dp0 C) (𝟙 _) f + primPull (dp0 C) (𝟙 _) g

theorem **primPull_id** : primPull (dp0 C) (𝟙 _) (𝟙 B) = LinearMap.id

theorem **opPull_add** (ρ : RingAct B O) (x y : O) : opPull ρ (x + y) = opPull ρ x + opPull ρ y

theorem **opPull_one** (ρ : RingAct B O) : opPull ρ 1 = 1

@[simp] theorem **tensorPull_tmul** (ρ : RingAct B O) (c : C) (x : O) : tensorPull ρ (c ⊗ₜ x) = c • opPull ρ x

theorem **mem_eigSub** {ρ : RingAct B O} {K : Type} [CommRing K] [Algebra K O] {χ : K →+* C} {h : primDR B} : h ∈ eigSub ρ χ ↔ ∀ x, opPull ρ (algebraMap K O x) h = χ x • h

theorem **primPull_opPull** {ρ : RingAct B O} {ρ' : RingAct B' O} {u : B' ⟶ B} (hu : GlobalTheory.IwLevel.IsLin ρ' ρ u) (x : O) : haveI

theorem **primPull_tensorPull** {ρ : RingAct B O} {ρ' : RingAct B' O} {u : B' ⟶ B} (hu : GlobalTheory.IwLevel.IsLin ρ' ρ u) (z : C ⊗[ℤ] O) : haveI

theorem **pullPart_apply** {ρ : RingAct B O} {ρ' : RingAct B' O} {K : Type} [CommRing K] [Algebra K O] (χ : K →+* C) (ε : C ⊗[ℤ] O) (u : B' ⟶ B) (hu : GlobalTheory.IwLevel.IsLin ρ' ρ u) (h : epsPart ρ χ ε) : haveI

theorem **π_close** (i : Fin (n + 1)) {a : O} (ha : a ∈ 𝔪) : X.π i ≫ X.close i ha = ρ.act a

theorem **close_isLin** (i : Fin (n + 1)) {a : O} (ha : a ∈ 𝔪) (hc : ∀ x, a * x = x * a) : IsLin (X.act i) ρ (X.close i ha)

theorem **isLin_eqToHom** {i j : Fin (n + 1)} (h : i = j) : IsLin (X.act i) (X.act j) (eqToHom (congrArg X.B h))

theorem **stepRev_isLin** (j : Fin n) : IsLin (X.act _) (X.act _) (X.stepRev j)

theorem **closeRev_isLin** {a : O} (ha : a ∈ 𝔪) (hc : ∀ x, a * x = x * a) : IsLin (X.act _) (X.act _) (X.closeRev ha)

Development.AbelianVarietyTheory.Definitions.Divisors.PrimitiveCrystalHodgeLiftClasses

theorem **schemeFil_eq_liftFil** (κ : DividedPowers I) {S : Scheme.{0}} {t : S ⟶ Sp R} {A₀ : Over S} [GrpObj A₀] {A : Over (Sp R)} [GrpObj A] {i : A₀.left ⟶ A.left} (h : IsGrpPullback A₀ A t i) (hp : AllPrim (dp0 R) (𝟙 _) A) : schemeFil κ A₀ A i h.1.w = liftFil κ h

theorem **isPDNilpotent_iff** (κ : DividedPowers I) : IsNilDP κ ↔ ∃ N, NilExp κ N

Development.AbelianVarietyTheory.Definitions.Divisors.PrimitiveStructureSheafDegTwoVanish

theorem **overW** {R₀ : Type} [CommRing R₀] {A B : Over (Sp R₀)} (f : A ⟶ B) : f.left ≫ B.hom = A.hom ≫ spMap (RingHom.id R₀)

theorem **ofSys_isFun** {X₀ : Scheme.{0}} {s₀ : X₀ ⟶ Sp R₀} (S : Sys I π s₀) : (ofSys S).IsFun

theorem **Pre.IsFun.restrict** {F : C ⥤ CommAlgCat.{0} R₀} {L : Pre I π F} (h : L.IsFun) (G : C' ⥤ C) : (L.restrict G).IsFun

Development.AbelianVarietyTheory.Definitions.Divisors.PrismaticDieudonnePerfdIff

theorem **PrismData.nonempty** (h : PrismaticDieudonnePerfd) (p : ℕ) [Fact p.Prime] : Nonempty (PrismData p)

theorem **prismaticDieudonnePerfd_iff** : PrismaticDieudonnePerfd ↔ ∀ (p : ℕ) [Fact p.Prime], Nonempty (PrismData p)

theorem **pairing_bidual** (G : PDivGroup p O) (t : G.Tate) (χ : G.dual.Tate) : pairing G.dual O χ (tateCongr O G.bidual t) = pairing G O t χ

theorem **constHom_comp_dualChar** (G : PDivGroup p O) (t : G.Tate) (χ : G.dual.Tate) : constHom G.dual χ ≫ dualChar G t = constHom (PDivGroup.mu p O) (pairing G O t χ)

theorem **dualHom_charHom** (G : PDivGroup p O) (χ : G.dual.Tate) : dualHom G (PDivGroup.mu p O) (charHom G χ) = constHom G.dual χ

theorem **baseChange_mulP** (G : PDivGroup p O) (n : ℕ) : (G.baseChange O').mulP n = PDiv.bcMap O' (G.mulP n)

@[simp] theorem **bcTate_pt** (t : G.Tate) (n : ℕ) (r : O') (a : G.A n) : (bcTate O' t).pt n (r ⊗ₜ a) = r * algebraMap O O' (t.pt n a)

theorem **isGroupLikeElem_one_tmul** (χ : G.dual.Tate) (n : ℕ) : IsGroupLikeElem O' ((1 : O') ⊗ₜ[O] charElem χ n) where

@[simp] theorem **charElem_bcChar** (χ : G.dual.Tate) (n : ℕ) : charElem (bcChar O' χ) n = 1 ⊗ₜ charElem χ n

theorem **muRoot_pairing_bc** (t : G.Tate) (χ : G.dual.Tate) (n : ℕ) : muRoot (pairing _ O' (bcTate O' t) (bcChar O' χ)) n = algebraMap O O' (muRoot (pairing G O t χ) n)

theorem **tateMap_bcTate** {H : PDivGroup p O} (f : G ⟶ H) (t : G.Tate) : tateMap _ _ O' (baseChangeMap O' f) (bcTate O' t) = bcTate O' (tateMap G H O f t)

theorem **dlog_bcChar** (χ : G.dual.Tate) : dlog _ (bcChar O' χ) = (G.toBase O').omega (dlog G χ)

theorem **tateMap_dualBaseChange** (χ : G.dual.Tate) : tateMap _ _ O' (dualBaseChange O' G).hom (bcChar O' χ) = bcTate O' χ

theorem **constHom_comp** (i j : Fin G.height) : constHom G (c.t i) ≫ charHom G (c.t' j) = if i = j then constHom (PDivGroup.mu p O) ε else 0

theorem **pairing_toTate** (x : Fin G.height → ℤ_[p]) (j : Fin G.height) : pairing G O (c.toTate x) (c.t' j) = x j • ε

theorem **toTate_injective** (hε : ∀ a : ℤ_[p], a • ε = 0 → a = 0) : Function.Injective c.toTate

theorem **stable_iff** (f : End G) (ρ : Matrix (Fin G.height) (Fin G.height) ℤ_[p]) : c.Stable f ρ ↔ ∀ x, tateMap G G O f (c.toTate x) = c.toTate (ρ.mulVec x)

theorem **stable_algebraMap** (a : ℤ_[p]) : c.Stable (algebraMap _ _ a) (algebraMap _ _ a)

theorem **Stable.add** {c : ConstTate ε G} {f g : End G} {ρ σ : Matrix (Fin G.height) (Fin G.height) ℤ_[p]} (hf : c.Stable f ρ) (hg : c.Stable g σ) : c.Stable (f + g) (ρ + σ)

theorem **Stable.mul** {c : ConstTate ε G} {f g : End G} {ρ σ : Matrix (Fin G.height) (Fin G.height) ℤ_[p]} (hf : c.Stable f ρ) (hg : c.Stable g σ) : c.Stable (f ∗ g) (ρ ∗ σ)

theorem **al_apply** (x : Fin G.height → O) : c.al x = ∑ i, x i • hodgeTate G (c.t i)

@[simp] theorem **alDual_apply** (x : G.Lie) (j : Fin G.height) : c.alDual x j = x (dlog G (c.t' j))

theorem **mem_W** (x : Fin G.height → O) : x ∈ c.W ↔ c.al x = 0

theorem **omegaMap_dual_al** (y : Fin G.height → O) : omegaMap _ _ G.bidual.hom (c.dual.al y) = ∑ j, y j • dlog G (c.t' j)

theorem **Stable.al** {c : ConstTate ε G} {f : End G} {ρ : Matrix (Fin G.height) (Fin G.height) ℤ_[p]} (hf : c.Stable f ρ) (φ : ℤ_[p] →+∗ O) (x : Fin G.height → O) : c.al ((ρ.map φ).mulVec x) = omegaDualMap G G f (c.al x)

theorem **exists_bijective** (C : Type) [Field C] [Algebra O C] [IsIntegrallyClosedIn O C] [IsAlgClosed C] [CharZero C] (ε : (PDivGroup.mu p O).Tate) (G : PDivGroup p O) : ∃ c : ConstTate ε G, Function.Bijective c.toTate

@[simp] theorem **piMap_apply** {A B : Type} [CommRing A] [CommRing B] (σ : A →+∗ B) (ι : Type) (x : ι → A) (i : ι) : piMap σ ι x i = σ (x i)

theorem **quot_exact** {N P Q : Submodule R X} (h : N ≤ P) (h' : P ≤ Q) : Function.Injective (quotIncl N h') ∧ Function.Exact (quotIncl N h') (quotProj h Q) ∧ Function.Surjective (quotProj h Q)

@[simp] theorem **muRoot_eps** (n : ℕ) : muRoot (eps S) n = zeta S.ε n

theorem **constHom_eps** : constHom _ (eps S) = epsHom (zeta S.ε) S.zeta_zero (zeta_succ_pow S.ε)

theorem **xiN_mul_muN** (n : ℕ) : xiN S n ∗ muN S (n + 1) = muN S n

theorem **phiInv_muN** (n : ℕ) : phiInv p S.R (muN S n) = muN S (n + 1)

theorem **phiInv_xiN** (n : ℕ) : phiInv p S.R (xiN S n) = xiN S (n + 1)

theorem **frobenius_muN** (n : ℕ) : frobenius (muN S (n + 1)) = muN S n

theorem **frobenius_xiN** (n : ℕ) : frobenius (xiN S (n + 1)) = xiN S n

theorem **theta_muN** (n : ℕ) : fontaineTheta S.R p (muN S n) = zeta S.ε n − 1

theorem **zeta_one_sub_one_dvd** : zeta S.ε 1 − 1 ∣ (p : S.R)

theorem **zeta_sub_one_dvd** (n : ℕ) : zeta S.ε (n + 1) − 1 ∣ zeta S.ε 1 − 1

theorem **isSMulRegular_zeta_sub_one** (n : ℕ) : IsSMulRegular S.R (zeta S.ε (n + 1) − 1)

theorem **zeta_one_ne_one** : zeta S.ε 1 ≠ 1

theorem **not_isUnit_zeta_sub_one** : ¬IsUnit (zeta S.ε 1 − 1)

theorem **smul_eps_eq_zero** (a : ℤ_[p]) (h : a • eps S = 0) : a = 0

theorem **muRoot_eps_of_over** {S S' : CycPerfd p} [Algebra S.R S'.R] {α : Ainf p S.R →+∗ Ainf p S'.R} (h : S.Over S' α) (n : ℕ) : muRoot (eps S') n = algebraMap S.R S'.R (muRoot (eps S) n)

theorem **mem_base** {h : ℕ} (x : Fin h → S.R) : x ∈ base S h ↔ ∃ y, (zeta S.ε 1 − 1) • y = x

@[simp] theorem **PrismData.eCoord_smul_e** (a : Ainf p S.R) : D.eCoord S (a • D.e S) = a

@[simp] theorem **PrismData.eCoord_smul** (m : D.M S (const p S.R)) : D.eCoord S m • D.e S = m

theorem **PrismData.eCoord_φ** (m : D.M S (const p S.R)) : D.eCoord S (D.φ S _ m) = frobenius (D.eCoord S m)

theorem **PrismData.toLie_surjective** (G : PDivGroup p S.R) : Function.Surjective (D.toLie G)

theorem **PrismData.mem_Fil** (G : PDivGroup p S.R) (m : D.Cov G) : m ∈ D.Fil G ↔ ∃ n, D.φ S G.dual m = frobenius (xi S.ε) • n

theorem **inj_apply** (x : Fin G.height → Ainf p S.R) : inj D c x = ∑ i, x i • D.map S G.dual (PDivGroup.mu p S.R) (dualChar G (c.t i)) (D.l S)

theorem **coord_apply** (m : D.Cov G) (j : Fin G.height) : coord D c m j = D.eCoord S (D.map S (const p S.R) G.dual (constHom G.dual (c.t' j)) m)

theorem **coord_smul_e** (m : D.Cov G) (j : Fin G.height) : coord D c m j • D.e S = D.covMap G (PDivGroup.mu p S.R) (charHom G (c.t' j)) m

theorem **coord_inj** (x : Fin G.height → Ainf p S.R) : coord D c (inj D c x) = Sandwich.mu S.ε • x

theorem **coord_φ** (m : D.Cov G) : coord D c (D.φ S G.dual m) = piMap frobenius _ (coord D c m)

theorem **K_zero** : K D c 0 = LinearMap.range (coord D c)

theorem **K_succ** (n : ℕ) : K D c (n + 1) = (K D c n).comap (piMap frobenius _)

theorem **mem_K_zero** (x : Fin G.height → Ainf p S.R) : x ∈ K D c 0 ↔ ∃ m, coord D c m = x

theorem **mem_K_succ** (n : ℕ) (x : Fin G.height → Ainf p S.R) : x ∈ K D c (n + 1) ↔ piMap frobenius _ x ∈ K D c n

theorem **mem_K** (n : ℕ) (x : Fin G.height → Ainf p S.R) : x ∈ K D c n ↔ ∃ m, ∀ j, (phiInv p S.R)^[n] (coord D c m j) = x j

theorem **muN_smul_mem_K** (n : ℕ) (x : Fin G.height → Ainf p S.R) : muN S n • x ∈ K D c n

theorem **xiN_smul_mem_K** (n : ℕ) {x : Fin G.height → Ainf p S.R} (hx : x ∈ K D c (n + 1)) : xiN S n • x ∈ K D c n

theorem **K_le_succ** (n : ℕ) : K D c n ≤ K D c (n + 1)

theorem **b_apply** (m : D.Cov G) (j : Fin G.height) : b D c m j = fontaineTheta S.R p (coord D c m j)

theorem **thetaK_zero** : thetaK D c 0 = LinearMap.range (b D c)

theorem **mem_H** (n : ℕ) (x : Fin G.height → S.R) : x ∈ H D c n ↔ ∃ k ∈ K D c (n + 1), (zeta S.ε 1 − 1) • piMap (fontaineTheta S.R p) _ k = (zeta S.ε (n + 1) − 1) • x

theorem **H_zero** : H D c 0 = thetaK D c 1

theorem **base_le_H** (n : ℕ) : base S G.height ≤ H D c n

theorem **H_succ_le** (n : ℕ) : H D c (n + 1) ≤ H D c n

theorem **qq_exact** (r : ℕ) : Function.Injective (qqIncl D c r) ∧ Function.Exact (qqIncl D c r) (qqProj D c r) ∧ Function.Surjective (qqProj D c r)

theorem **inj_const_surjective** : Function.Surjective (inj D (ConstTate.ofConst (eps S)))

theorem **H_eq_base** (h : Function.Surjective (inj D c)) (n : ℕ) : H D c n = base S G.height

theorem **H_const** (n : ℕ) : H D (ConstTate.ofConst (eps S)) n = base S 1

theorem **K_mu** (n : ℕ) : K D (ConstTate.ofMu (eps S)) n = ⊤

theorem **mem_H_mu** (n : ℕ) (x : Fin 1 → S.R) : x ∈ H D (ConstTate.ofMu (eps S)) n ↔ (zeta S.ε (n + 1) - 1) • x ∈ base S 1

theorem **H_mu_zero** : H D (ConstTate.ofMu (eps S)) 0 = ⊤ ∧ base S 1 ≠ ⊤

theorem **exists_constTate** (p : ℕ) [Fact p.Prime] : ∃ S : CycPerfd p, IsPrimitiveRoot (zeta S.ε 1) p ∧ ∀ G : PDivGroup p S.R, ∃ c : ConstTate (eps S) G, Function.Bijective c.toTate

Development.AbelianVarietyTheory.Definitions.Divisors.ProjectiveNotIsUnit

@[simp] theorem **Charts.ev_X** (C : Charts I Y β n) (e : ℕ) (i j : Fin (n + 1)) : C.ev e i (MvPolynomial.X j) = C.ρ e i j

@[simp] theorem **Charts.ev_C** (C : Charts I Y β n) (e : ℕ) (i : Fin (n + 1)) (r : R) : C.ev e i (MvPolynomial.C r) = cst ((Y e).hom ≫ thick R I e) (C.D e i) r

theorem **Charts.ρ_mul_ρ** (C : Charts I Y β n) (i j : Fin (n + 1)) : rs (Y 0).left (inf_le_left : C.D 0 i ⊓ C.D 0 j ≤ C.D 0 i) (C.ρ 0 i j) * rs (Y 0).left (inf_le_right : C.D 0 i ⊓ C.D 0 j ≤ C.D 0 j) (C.ρ 0 j i) = 1

theorem **Charts.le_of_isUnit** (C : Charts I Y β n) {i j : Fin (n + 1)} (h : IsUnit (C.ρ 0 i j)) : C.D 0 i ≤ C.D 0 j

theorem **rs_app** {X Z : Scheme.{0}} (b : Z ⟶ X) {U V : X.Opens} (h : V ≤ U) (h' : b ⁻¹ᵁ V ≤ b ⁻¹ᵁ U) (x : Γ(X, U)) : rs Z h' ((b.app U).hom x) = (b.app V).hom (rs X h x)

theorem **Charts.projective_not_isUnit** (R : Type) [CommRing R] (I : Ideal R) (hI : I ≠ ⊤) (e : ℕ) : ¬ IsUnit ((Charts.projective R I 1).ρ e 0 1)

theorem **Charts.projective_D_ne** (R : Type) [CommRing R] (I : Ideal R) (hI : I ≠ ⊤) : (Charts.projective R I 1).D 0 0 ≠ (Charts.projective R I 1).D 0 1

Development.AbelianVarietyTheory.Definitions.Divisors.PtAlgSpOver

theorem **spOver_ptAlg** (z : Pt (SpAlg R D) E) : spOver (ptAlg z) = z

theorem **ptAlg_spOver** (α : D →ₐ[R] E) : ptAlg (spOver α) = α

theorem **ptAlg_comp** (ψ : E →ₐ[R] E') (z : Pt (SpAlg R D) E) : ptAlg (spOver ψ ≫ z) = ψ.comp (ptAlg z)

Development.AbelianVarietyTheory.Definitions.Divisors.PullData

theorem **CanonOn.canon** (h : CanonOn Q W G P) : Canon Q G P

theorem **CanonOn.mono** (h : CanonOn Q W G P) (hW : W' ≤ W) : CanonOn Q W' G P

Development.AbelianVarietyTheory.Definitions.Divisors.PullHlComp

theorem **pick_spec** {α : Type*} [Zero α] {P : α → Prop} (h : ∃ a, P a) : P (pick P)

theorem **twRel_le** (q : ℕ) (i : ℤ) {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) : twRel q i A ≤ (twRel q i B).comap (Finsupp.lmapDomain _ _ (Prim.map f))

theorem **cpow_zero** (q : ℕ) {E : Spc} (x : Ht q E 2 1) : cpow q x 0 = (one q E).cast (by simp) (by simp)

theorem **cpow_succ** (q : ℕ) {E : Spc} (x : Ht q E 2 1) (k : ℕ) : cpow q x (k + 1) = (cup q (cpow q x k) x).cast (by push_cast; ring) (by push_cast; ring)

theorem **pull_red** (Q q : ℕ) {E E' : Spc} (f : E ⟶ E') (n i : ℤ) (x : Ht Q E' n i) : pull _ f _ n (red Q q E' n i x) = red Q q E n i (pull _ f _ n x)

theorem **HtL.smul_eq** (k k' : ℕ) (hk : k ≤ k') (E : Spc) (n i : ℤ) (r : ℤ_[l]) (x : HtL l k E n i) : r • x = (PadicInt.toZModPow k' r).val • x

theorem **pullHl_id** (E : Spc) (n i : ℤ) : pullHl l (𝟙 E) n i = LinearMap.id

theorem **pullHl_comp** {E E' E'' : Spc} (f : E ⟶ E') (g : E' ⟶ E'') (n i : ℤ) : pullHl l (f ≫ g) n i = (pullHl l f n i).comp (pullHl l g n i)

Development.AbelianVarietyTheory.Definitions.Divisors.Pullback1MemHodgeFiltration

private theorem **prod_dpow_mem_and_eq_zero_of_sq_zero** (δ : DividedPowers J) (h2 : ∀ a ∈ J, ∀ b ∈ J, a * b = 0) (h : ∀ n, 2 ≤ n → ∀ a ∈ J, δ.dpow n a = 0) (l : List (ℕ × B)) (hl : ∀ p ∈ l, p.2 ∈ J) : (1 ≤ (l.map Prod.fst).sum → (l.map fun p ↦ δ.dpow p.1 p.2).prod ∈ J) ∧ (2 ≤ (l.map Prod.fst).sum → (l.map fun p ↦ δ.dpow p.1 p.2).prod = 0)

theorem **isDPNilpotent_of_sq** (δ : DividedPowers J) (h2 : ∀ a ∈ J, ∀ b ∈ J, a * b = 0) (h : ∀ n, 2 ≤ n → ∀ a ∈ J, δ.dpow n a = 0) : IsNilDP δ

theorem **isDPNilpotent_bot** (B : Type) [CommRing B] : IsNilDP (dividedPowersBot B)

theorem **IsNilDP.exists_dpow_eq_zero** {δ : DividedPowers J} (h : IsNilDP δ) : ∃ N : ℕ, ∀ n, N ≤ n → ∀ a, δ.dpow n a = 0

theorem **isDPMorphism_of_eq_bot** (δ : DividedPowers J) (hJ : J = ⊥) (δ' : DividedPowers J') (f : B →+* B') : IsDPMorphism δ δ' f

theorem **squareZeroDividedPowers_dpow** (h : J ^ 2 = 0) {n : ℕ} (hn : 2 ≤ n) (a : B) : (divided₂ h).dpow n a = 0

theorem **isDPNilpotent_squareZero** (h : J ^ 2 = 0) : IsNilDP (divided₂ h)

theorem **isDPMorphism_sqDP** (h : J ^ 2 = 0) (h' : J' ^ 2 = 0) (f : B →+* B') (hf : ∀ a ∈ J, f a ∈ J') : IsDPMorphism (divided₂ h) (divided₂ h') f

theorem **dpExp_zero** (δ : DividedPowers J) : dpExp δ 0 = 1

theorem **dpExp_sqDP** (h : J ^ 2 = 0) {a : B} (ha : a ∈ J) : dpExp (divided₂ h) a = 1 + a

theorem **spMap_comp** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap (g.comp f) = spMap g ≫ spMap f

theorem **spMap_id** (A : Type) [CommRing A] : spMap (RingHom.id A) = 𝟙 _

theorem **Hom.ext'** {T T' : Thick γ s} {f g : Hom T T'} (h : f.φ = g.φ) : f = g

@[ext] theorem **hom_ext** {T T' : Thick γ s} {f g : T ⟶ T'} (h : ∀ b, f.φ b = g.φ b) : f = g

@[simp] theorem **comp_φ** {T T' T'' : Thick γ s} (f : T ⟶ T') (g : T' ⟶ T'') (b : T.B) : (f ≫ g).φ b = g.φ (f.φ b)

@[simp] theorem **id_φ** (T : Thick γ s) (b : T.B) : Hom.φ (𝟙 T) b = b

theorem **Hom.mem** {T T' : Thick γ s} (f : T ⟶ T') {b : T.B} (hb : b ∈ T.J) : f.φ b ∈ T'.J

theorem **mp_0** {γ : DividedPowers I} {s : X ⟶ Sp R} {T T' : Thick γ s} (f : T ⟶ T') : mp (O γ s) f = f.φ

theorem **ev_comp** {γ : DividedPowers I} {s : X ⟶ Sp R} (f : Z1 (O γ s)) {T T' T'' : Thick γ s} (φ : T ⟶ T') (ψ : T' ⟶ T'') : ev f (φ ≫ ψ) = ψ.φ (ev f φ) + ev f ψ

theorem **ev_sq** {γ : DividedPowers I} {s : X ⟶ Sp R} (f : Z1 (O γ s)) {T₁ T₂ T₃ T₄ : Thick γ s} (a : T₁ ⟶ T₂) (p : T₂ ⟶ T₄) (q : T₁ ⟶ T₃) (b : T₃ ⟶ T₄) (h : ∀ c, p.φ (a.φ c) = b.φ (q.φ c)) : p.φ (ev f a) + ev f p = b.φ (ev f q) + ev f b

theorem **IsMor.id** (γ : DividedPowers I) (s : X ⟶ Sp R) : IsMor γ s γ s (RingHom.id R) (𝟙 X)

theorem **IsMor.of_over** (γ : DividedPowers I) {s : X ⟶ Sp R} {s' : X' ⟶ Sp R} {φ : X' ⟶ X} (h : φ ≫ s = s') : IsMor γ s γ s' (RingHom.id R) φ

theorem **IsMor.comp** {σ : R →+* R'} {φ : X' ⟶ X} {τ : R' →+* R''} {ψ : X'' ⟶ X'} (h : IsMor γ s γ' s' σ φ) (h' : IsMor γ' s' γ'' s'' τ ψ) : IsMor γ s γ'' s'' (τ.comp σ) (ψ ≫ φ)

theorem **IsMor.pullCohomologyTwo_cup** (h : IsMor γ s γ' s' σ φ) (a b : H1 γ s) : h.pull2 (cup _ a b) = cup _ (h.pull1 a) (h.pull1 b)

theorem **IsMor.pullCohomologyOne_id** (h : IsMor γ s γ s (RingHom.id R) (𝟙 X)) (c : H1 γ s) : h.pull1 c = c

theorem **IsMor.pullCohomologyTwo_id** (h : IsMor γ s γ s (RingHom.id R) (𝟙 X)) (c : H2 γ s) : h.pull2 c = c

theorem **IsMor.pullback1_comp** (h : IsMor γ s γ' s' σ φ) (h' : IsMor γ' s' γ'' s'' τ ψ) (h'' : IsMor γ s γ'' s'' ρ χ) (e : ∀ r, τ (σ r) = ρ r) (e' : ψ ≫ φ = χ) (c : H1 γ s) : h'.pull1 (h.pull1 c) = h''.pull1 c

theorem **IsMor.pullback2_comp** (h : IsMor γ s γ' s' σ φ) (h' : IsMor γ' s' γ'' s'' τ ψ) (h'' : IsMor γ s γ'' s'' ρ χ) (e : ∀ r, τ (σ r) = ρ r) (e' : ψ ≫ φ = χ) (c : H2 γ s) : h'.pull2 (h.pull2 c) = h''.pull2 c

theorem **IsMor.of_dividedPowersZeroIdeal** {s : X ⟶ Sp R} (γ' : DividedPowers I') {s' : X' ⟶ Sp R'} {σ : R →+* R'} {φ : X' ⟶ X} (h : φ ≫ s = s' ≫ spMap σ) : IsMor (dp0 R) s γ' s' σ φ

theorem **Form.ev_sq** (f : Form s) {T₁ T₂ T₃ T₄ : Thick (dp0 R) s} (a : T₁ ⟶ T₂) (p : T₂ ⟶ T₄) (q : T₁ ⟶ T₃) (b : T₃ ⟶ T₄) (h₁ : T₁.J = ⊥) (h₂ : T₂.J = ⊥) (h : ∀ c, p.φ (a.φ c) = b.φ (q.φ c)) : ev f.1 p = b.φ (ev f.1 q) + ev f.1 b

theorem **hodgeFiltration_eq_range** (s : X ⟶ Sp R) : F1 s = LinearMap.range (Form.cl s)

theorem **IsMor.cl_pullForm** {s' : X' ⟶ Sp R'} {σ : R →+* R'} {φ : X' ⟶ X} (h : IsMor (dp0 R) s (dp0 R') s' σ φ) (f : Form s) : Form.cl s' (h.pullForm f) = h.pull1 (Form.cl s f)

theorem **IsMor.pullback1_mem_hodgeFiltration** {s' : X' ⟶ Sp R'} {σ : R →+* R'} {φ : X' ⟶ X} (h : IsMor (dp0 R) s (dp0 R') s' σ φ) {c : FirstDeRham s} (hc : c ∈ F1 s) : h.pull1 c ∈ F1 s'

theorem **red_w** {y : Sp E.C ⟶ X} (hy : y ≫ s = spMap (algebraMap R E.C)) : (spMap (E.π : E.C →+* E.B) ≫ y) ≫ s = spMap (algebraMap R E.B)

theorem **val_apply** (y : Sp E.C ⟶ X) (hy : y ≫ s = spMap (algebraMap R E.C)) (f : Z1 (O (dp0 R) s)) : E.val y hy f = ev f (E.secHom (E.red_w hy) E.i E.π_i) - ev f (E.idHom hy)

theorem **tangentValue_coboundary** (y : Sp E.C ⟶ X) (hy : y ≫ s = spMap (algebraMap R E.C)) (g : C0 (O (dp0 R) s)) : E.val y hy (d0 _ g, commAlgFunctorCoboundaryZero_mem_cocyclesOne _ g) = (g (Thick.triv E.C y hy) : E.C) - E.i (g (Thick.triv E.B _ (E.red_w hy)))

theorem **val_map** (φ : E.C →ₐ[R] E'.C) (φ₀ : E.B →ₐ[R] E'.B) (hπ : ∀ c, φ₀ (E.π c) = E'.π (φ c)) (hi : ∀ b, φ (E.i b) = E'.i (φ₀ b)) (y : Sp E.C ⟶ X) (hy : y ≫ s = spMap (algebraMap R E.C)) (y' : Sp E'.C ⟶ X) (hy' : y' ≫ s = spMap (algebraMap R E'.C)) (e : spMap (φ : E.C →+* E'.C) ≫ y = y') (f : Form s) : E'.val y' hy' f.1 = φ (E.val y hy f.1)

theorem **val_pull** {s' : X' ⟶ Sp R} {φ : X' ⟶ X} (h : IsMor (dp0 R) s (dp0 R) s' (RingHom.id R) φ) (y : Sp E.C ⟶ X') (hy : y ≫ s' = spMap (algebraMap R E.C)) (hy' : (y ≫ φ) ≫ s = spMap (algebraMap R E.C)) (f : Form s) : E.val y hy (h.pullForm f).1 = E.val (y ≫ φ) hy' f.1

theorem **val_self** {B : Type} [CommRing B] [Algebra R B] (y : Sp B ⟶ X) (hy : y ≫ s = spMap (algebraMap R B)) (f : Form s) : (self R B).val y hy f.1 = 0

theorem **π_val** (y : Sp E.C ⟶ X) (hy : y ≫ s = spMap (algebraMap R E.C)) (f : Form s) : E.π (E.val y hy f.1) = 0

theorem **val_const** (x : Sp E.B ⟶ X) (hx : x ≫ s = spMap (algebraMap R E.B)) (y : Sp E.C ⟶ X) (hy : y ≫ s = spMap (algebraMap R E.C)) (e : spMap (E.i : E.B →+* E.C) ≫ x = y) (f : Form s) : E.val y hy f.1 = 0

Development.AbelianVarietyTheory.Definitions.Divisors.PullbackBijectiveDegreesOneTwo

theorem **pullback_firstDifferential** (g : K.M0.c0) : m.p1 (K.D0 g) = K'.D0 (m.p0 g)

theorem **pullback_secondDifferential** (c : K.T1) : m.p2 (K.D1 c) = K'.D1 (m.p1 c)

theorem **p2_mem_Z2_of_mem_Z2** {c : K.T2} (h : c ∈ K.Z2) : m.p2 c ∈ K'.Z2

theorem **cohomZero_pullback_comp** (n : CxMap K' K'') (c : K.H0) : (m.comp n).h0 c = n.h0 (m.h0 c)

theorem **cohomOne_pullback_comp** (n : CxMap K' K'') (c : K.H1) : (m.comp n).h1 c = n.h1 (m.h1 c)

theorem **cohomTwo_pullback_comp** (n : CxMap K' K'') (c : K.H2) : (m.comp n).h2 c = n.h2 (m.h2 c)

Development.AbelianVarietyTheory.Definitions.Divisors.QuasiRegularAlongOfEtale

theorem **vanishesTo_iff_forall** {f : H0 p B O} : VanishesTo I a f ↔ ∀ y hy i hi, germ i y hy hi f ∈ stalkId I y ^ a

theorem **vanishesTo_of_exists** {f : H0 p B O} (h : ∀ y ∈ I.support, ∀ hy : y ∈ O, ∃ i hi, germ i y hy hi f ∈ stalkId I y ^ a) : VanishesTo I a f

theorem **vanishesTo_iff_affine** {f : H0 p B O} : VanishesTo I a f ↔ ∀ (W : Y.affineOpens) (i : B.ι) (h : W.1 ≤ O ⊓ B.U i), resG Y h (f.loc i) ∈ I.ideal W ^ a

theorem **vanishesTo_zero** (f : H0 p B O) : VanishesTo I 0 f

theorem **VanishesTo.mono** {f : H0 p B O} (h : VanishesTo I b f) (e : a ≤ b) : VanishesTo I a f

theorem **VanishesTo.restr** {f : H0 p B O} (h : VanishesTo I a f) (e : O' ≤ O) : VanishesTo I a (restr e f)

theorem **VanishesTo.mul** {P N : LB Y} (C : ProdChart P B N) {σ : H0 p P O} {f : H0 p B O} (hσ : VanishesTo I a σ) (hf : VanishesTo I b f) : VanishesTo I (a + b) (C.mul σ f)

theorem **suppIn_top** : SuppIn ⊤ I

theorem **SuppIn.le** (h : SuppIn O I) : ⊤ ≤ I.subschemeι ⁻¹ᵁ O

theorem **vanishesTo_restr_iff** (h : SuppIn O' I) (e : O' ≤ O) {f : H0 p B O} : VanishesTo I a (restr e f) ↔ VanishesTo I a f

@[simp] theorem **mem_filTop** {f : H0 p B O} : f ∈ filTop p B O I a ↔ VanishesTo I a f

@[simp] theorem **mem_Fil** {f : H0 p B O} : f ∈ Fil I a V ↔ f ∈ V ∧ VanishesTo I a f

theorem **Fil_le** : Fil I a V ≤ V

theorem **Fil_le_filTop** : Fil I a V ≤ filTop p B O I a

@[simp] theorem **Fil_top** : Fil I a (⊤ : Submodule k (H0 p B O)) = filTop p B O I a

@[simp] theorem **Fil_zero** : Fil I 0 V = V

theorem **Fil_anti** (e : a ≤ b) : Fil I b V ≤ Fil I a V

theorem **germ_resTo** (h : SuppIn O I) (j : (B.pull I.subschemeι).ι) (z : I.subscheme) (hj : z ∈ (B.pull I.subschemeι).U j) (f : H0 p B O) : germ j z (Opens.mem_top z) hj (resTo h f) = (I.subschemeι.stalkMap z).hom (germ (j PUnit.unit) _ (h _ (ι_mem_support I z)) (mem_pull.1 hj) f)

@[simp] theorem **resTo_restr** (h : SuppIn O' I) (e : O' ≤ O) (f : H0 p B O) : resTo h (restr e f) = resTo (fun y hy => e (h y hy)) f

theorem **resTo_mul** {P N : LB Y} (C : ProdChart P B N) (h : SuppIn O I) (σ : H0 p P O) (f : H0 p B O) : resTo h (C.mul σ f) = (C.pull I.subschemeι).mul (resTo h σ) (resTo h f)

theorem **resTo_eq_zero_iff** (h : SuppIn O I) {f : H0 p B O} : resTo h f = 0 ↔ VanishesTo I 1 f

@[simp] theorem **resTo_ofSec** (h : SuppIn O I) (σ : LB.Sec B) : resTo h (H0.ofSec p O σ) = secOn p I σ

theorem **germ_secOn** (j : (B.pull I.subschemeι).ι) (z : I.subscheme) (hj : z ∈ (B.pull I.subschemeι).U j) (σ : LB.Sec B) : germ j z (Opens.mem_top z) hj (secOn p I σ) = (I.subschemeι.stalkMap z).hom (ger _ (mem_pull.1 hj) (σ.f (j PUnit.unit)))

theorem **nowhereZero_secOn** {σ : LB.Sec B} (h : SuppIn (nonzeroLocus σ) I) : NowhereZero (secOn p I σ)

theorem **valOn_resTo** (h : SuppIn O I) (x : Sp R ⟶ I.subscheme) (i : B.ι) (hi : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ B.U i) (hO : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ O) (f : H0 p B O) : valOn x i hi (resTo h f) = val (x ≫ I.subschemeι) i hO hi f

theorem **valOn_secOn** (x : Sp R ⟶ I.subscheme) (i : B.ι) (hi : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ B.U i) (σ : LB.Sec B) : valOn x i hi (secOn p I σ) = evalPt (x ≫ I.subschemeι) _ hi (σ.f i)

theorem **valOn_mul** {P N : LB Y} (C : ProdChart P B N) (σ : sectionsOnClosedSubscheme p P I) (f : sectionsOnClosedSubscheme p B I) (t : C.τ) (x : Sp R ⟶ I.subscheme) (ha : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ P.U (C.a t)) (hb : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ B.U (C.b t)) (hc : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ N.U (C.c t)) : valOn x (C.c t) hc ((C.pull I.subschemeι).mul σ f) = valOn x (C.a t) ha σ * valOn x (C.b t) hb f

theorem **germ_mulSec** (w : L.Sec) (h : n + 1 = l) (f : H0 p (tw M L n) O) (i : M.ι) (j : L.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ M.U i) (hj : y ∈ L.U j) : germ (twIdx l i j) y hy (mem_tw_U.2 ⟨hi, hj⟩) (mulSec hM hL w h f) = ger y hj (w.f j) * germ (twIdx n i j) y hy (mem_tw_U.2 ⟨hi, hj⟩) f

theorem **resTo_mulSec** (hO : SuppIn O I) (w : L.Sec) (h : n + 1 = l) (f : H0 p (tw M L n) O) : resTo hO (mulSec hM hL w h f) = mulSecOn hM hL w h (resTo hO f)

theorem **valOn_mulSecOn** {R : Type} [CommRing R] (w : L.Sec) (h : n + 1 = l) (g : sectionsOnClosedSubscheme p (tw M L n) I) (x : Sp R ⟶ I.subscheme) (i : M.ι) (j : L.ι) (hj : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ L.U j) (h₁ : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ (tw M L n).U (twIdx n i j)) (h₂ : ⊤ ≤ (x ≫ I.subschemeι) ⁻¹ᵁ (tw M L l).U (twIdx l i j)) : valOn x (twIdx l i j) h₂ (mulSecOn hM hL w h g) = evalPt (x ≫ I.subschemeι) _ hj (w.f j) * valOn x (twIdx n i j) h₁ g

theorem **monoVal_const** {R : Type*} [CommMonoid R] (c : R) (α : MonoDeg ι a) : monoVal (fun _ => c) α.1 = c ^ a

theorem **zfun_monoSec** (s : ι → L.Sec) (α : MonoDeg ι a) (j : L.ι) : Twist.zfun (monoSec s α) j = monoVal (fun i => (s i).f j) α.1

theorem **germ_mulMono** (s : ι → L.Sec) (α : MonoDeg ι a) (h : n + a = l) (m : H0 p (tw M L n) O) (i : M.ι) (j : L.ι) (y : Y) (hy : y ∈ O) (hi : y ∈ M.U i) (hj : y ∈ L.U j) : germ (twIdx l i j) y hy (mem_tw_U.2 ⟨hi, hj⟩) (mulMono hM hL s α h m) = monoVal (secGer s j y hj) α.1 * germ (twIdx n i j) y hy (mem_tw_U.2 ⟨hi, hj⟩) m

theorem **tw_ext** (hM : M.IsLB) (hL : L.IsLB) {f g : H0 p (tw M L n) O} (h : ∀ i j y hy (hi : y ∈ M.U i) (hj : y ∈ L.U j), germ (twIdx n i j) y hy (mem_tw_U.2 ⟨hi, hj⟩) f = germ (twIdx n i j) y hy (mem_tw_U.2 ⟨hi, hj⟩) g) : f = g

theorem **mulMono_succ** (s : ι → L.Sec) (α : MonoDeg ι a) (β : MonoDeg ι (a + 1)) (i : ι) (e : β.1 = α.1 + Finsupp.single i 1) {l' : ℤ} (h : n + a = l) (h' : l + 1 = l') (h'' : n + (a + 1 : ℕ) = l') (m : H0 p (tw M L n) O) : mulMono hM hL s β h'' m = mulSec hM hL (s i) h' (mulMono hM hL s α h m)

theorem **mulMono_zero** (s : ι → L.Sec) (α : MonoDeg ι 0) (h : n + (0 : ℕ) = n) (m : H0 p (tw M L n) O) : mulMono hM hL s α h m = m

theorem **VanishesTo.mulMono** (s : ι → L.Sec) (α : MonoDeg ι a) (h : n + a = l) {m : H0 p (tw M L n) O} (hm : VanishesTo (cutIdeal hL s) b m) : VanishesTo (cutIdeal hL s) (a + b) (mulMono hM hL s α h m)

theorem **vanishesTo_mulMono** (s : ι → L.Sec) (α : MonoDeg ι a) (h : n + a = l) (m : H0 p (tw M L n) O) : VanishesTo (cutIdeal hL s) a (mulMono hM hL s α h m)

theorem **zone_f** (j : (lbZPow L 0).ι) : (Twist.zone L).f j = 1

theorem **nonzeroLocus_zone** : nonzeroLocus (Twist.zone L) = ⊔ j, L.U j

theorem **leadAt_sum** (m : MonoDeg ι a → R) : LeadAt φ x (∑ α, monoVal x α.1 * m α) fun α => φ (m α)

theorem **LeadAt.add** (h : LeadAt φ x f γ) (h' : LeadAt φ x f' γ') : LeadAt φ x (f + f') (γ + γ')

theorem **LeadAt.mul** (h : LeadAt φ x f γ) (r : R) : LeadAt φ x (r * f) fun α => φ r * γ α

theorem **LeadAt.unit_mul** (h : LeadAt φ x f γ) (u : Rˣ) : LeadAt φ (fun i => u * x i) (u ^ a * f) γ

theorem **leadAt_zero_iff** (hφ : RingHom.ker φ ≤ Ideal.span (Set.range x)) : LeadAt φ x f (0 : MonoDeg ι a → S) ↔ f ∈ Ideal.span (Set.range x) ^ (a + 1)

theorem **LeadAt.unique** (hq : IsQuasiRegularSeq R x) (hφ : Ideal.span (Set.range x) ≤ RingHom.ker φ) (h : LeadAt φ x f γ) (h' : LeadAt φ x f γ') : γ = γ'

theorem **leadAt_change** (h : n + a = l) {S : Type*} [CommRing S] {y : Y} {φ : Y.presheaf.stalk y →+* S} (hy : y ∈ O) {i i' : M.ι} {j j' : L.ι} (hi : y ∈ M.U i) (hj : y ∈ L.U j) (hi' : y ∈ M.U i') (hj' : y ∈ L.U j') {γ : MonoDeg ι a → S} (e : LeadAt φ (secGer s j' y hj') (germ (twIdx l i' j') y hy (mem_tw_U.2 ⟨hi', hj'⟩)) f) γ) : LeadAt φ (secGer s j y hj) (germ (twIdx l i j) y hy (mem_tw_U.2 ⟨hi, hj⟩)) f) fun α => φ (tr (tw M L n) y (twIdx n i j) (twIdx n i' j')).val * γ α

theorem **hasLead_of_exists** (e : ∀ z, ∃ i j hi hj, LeadIn hL s hO f c z i j hi hj) : HasLead hL s hO h f c

theorem **HasLead.add** (e : HasLead hL s hO h f c) (e' : HasLead hL s hO h f' c') : HasLead hL s hO h (f + f') (c + c')

theorem **HasLead.smul** (e : HasLead hL s hO h f c) (r : k) : HasLead hL s hO h (r • f) (r • c)

theorem **hasLead_zero_iff** (hM : M.IsLB) : HasLead hL s hO h f 0 ↔ VanishesTo (cutIdeal hL s) (a + 1) f

theorem **hasLead_sum** (hM : M.IsLB) (m : MonoDeg ι a → H0 p (tw M L n) O) : HasLead hL s hO h (∑ α, mulMono hM hL s α h (m α)) fun α => resTo hO (m α)

theorem **HasLead.unique** (hM : M.IsLB) (hq : QuasiRegularAlong hL s) (e : HasLead hL s hO h f c) (e' : HasLead hL s hO h f c') : c = c'

theorem **exists_hasLead** (hM : M.IsLB) (hq : QuasiRegularAlong hL s) (hf : VanishesTo (cutIdeal hL s) a f) : ∃ c, HasLead hL s hO h f c

theorem **lead_eq_iff** (hM : M.IsLB) (hq : QuasiRegularAlong hL s) (f : filTop p (tw M L l) O (cutIdeal hL s) a) : lead hO h hM hq f = c ↔ HasLead hL s hO h f.1 c

theorem **germ_dsOn** (z : (cutIdeal hL s).subscheme) (i : K.bundle.ι) (j : L.ι) (hi : (cutIdeal hL s).subschemeι z ∈ K.bundle.U i) (hj : (cutIdeal hL s).subschemeι z ∈ L.U j) : germ (fun _ => twIdx D i j) z (Opens.mem_top z) (mem_pull.2 (mem_tw_U.2 (hi, hj))) (dsOn K hL s) = ((cutIdeal hL s).subschemeι.stalkMap z).hom (ger _ (memInf hi hj) (K.jacFun s i j))

theorem **valOn_dsOn** {R : Type} [CommRing R] (x : Sp R ⟶ (cutIdeal hL s).subscheme) (i : K.bundle.ι) (j : L.ι) (hi : ⊤ ≤ (x ≫ (cutIdeal hL s).subschemeι) ⁻¹ᵁ K.bundle.U i) (hj : ⊤ ≤ (x ≫ (cutIdeal hL s).subschemeι) ⁻¹ᵁ L.U j) (h : ⊤ ≤ (x ≫ (cutIdeal hL s).subschemeι) ⁻¹ᵁ (tw K.bundle L D).U (twIdx D i j)) : valOn x (twIdx D i j) h (dsOn K hL s) = K.jacVal s (x ≫ (cutIdeal hL s).subschemeι) i j hi hj

theorem **nowhereZero_dsOn** : NowhereZero (dsOn K hL s)

theorem **quasiRegularAlong_of_etale** : QuasiRegularAlong hL s

theorem **resDs_eq_iff** (hO : SuppIn O (cutIdeal hL s)) (h : n + D = l) (f : H0 p (tw (tens K.bundle B) L l) O) (g : sectionsOnClosedSubscheme p (tw B L n) (cutIdeal hL s)) : resDs K hL hB s hS hO h f = g ↔ resTo hO f = mulDs K hL hB s hS h g

theorem **resTo_mulE** (hO : SuppIn O (cutIdeal hL s)) (h : n + D = l) (f : H0 p (tw (tens K.bundle B) L l) O) : resTo hO (mulE K hL hA hB e h f) = mulEOn K hL hA hB s e he h (resTo hO f)

theorem **germ_mulEDs** (g : sectionsOnClosedSubscheme p (tw B L n) (cutIdeal hL s)) (z : (cutIdeal hL s).subscheme) (i : K.bundle.ι) (a : A.ι) (b : B.ι) (j : L.ι) (hi : (cutIdeal hL s).subschemeι z ∈ K.bundle.U i) (ha : (cutIdeal hL s).subschemeι z ∈ A.U a) (hb : (cutIdeal hL s).subschemeι z ∈ B.U b) (hj : (cutIdeal hL s).subschemeι z ∈ L.U j) : germ (fun _ => twIdx (M

theorem **valOn_mulEDs** {R : Type} [CommRing R] (g : sectionsOnClosedSubscheme p (tw B L n) (cutIdeal hL s)) (x : Sp R ⟶ (cutIdeal hL s).subscheme) (i : K.bundle.ι) (a : A.ι) (b : B.ι) (j : L.ι) (hi : ⊤ ≤ (x ≫ (cutIdeal hL s).subschemeι) ⁻¹ᵁ K.bundle.U i) (hj : ⊤ ≤ (x ≫ (cutIdeal hL s).subschemeι) ⁻¹ᵁ L.U j) (hι : ⊤ ≤ (x ≫ (cutIdeal hL s).subschemeι) ⁻¹ᵁ (tw (tens K.bundle A) L n).U (twIdx (M

theorem **mulEDs_mulSecOn** (w : L.Sec) (h : n + 1 = l) (g : sectionsOnClosedSubscheme p (tw B L n) (cutIdeal hL s)) : mulEDs K hL hA hB s hS l e he (mulSecOn hB hL w h g) = mulSecOn (isLB_tens K.isLB hA) hL w h (mulEDs K hL hA hB s hS n e he g)

theorem **Res_eq_iff** (m : H0 p (tw (tens K.bundle A) L n) O) (g : sectionsOnClosedSubscheme p (tw B L n) (cutIdeal hL s)) : Res K hL hA hB s hS n e he hO m = g ↔ resTo hO m = mulEDs K hL hA hB s hS n e he g

theorem **mulEDs_Res** (m : H0 p (tw (tens K.bundle A) L n) O) : mulEDs K hL hA hB s hS n e he (Res K hL hA hB s hS n e he hO m) = resTo hO m

theorem **Res_mulSec** (w : L.Sec) (h : n + 1 = l) (m : H0 p (tw (tens K.bundle A) L n) O) : Res K hL hA hB s hS l e he hO (mulSec (isLB_tens K.isLB hA) hL w h m) = mulSecOn hB hL w h (Res K hL hA hB s hS n e he hO m)

theorem **Res_mulE** (h : n + D = l) (f : H0 p (tw (tens K.bundle B) L l) O) : Res K hL hA hB s hS n e he hO (mulE K hL hA hB e h f) = resDs K hL hB s hS hO h f

theorem **Res_restr** {O' : Y.Opens} (hO' : SuppIn O' (cutIdeal hL s)) (h : O' ≤ O) (m : H0 p (tw (tens K.bundle A) L n) O) : Res K hL hA hB s hS n e he hO' (restr h m) = Res K hL hA hB s hS n e he hO m

theorem **Res_eq_zero_iff** (m : H0 p (tw (tens K.bundle A) L n) O) : Res K hL hA hB s hS n e he hO m = 0 ↔ VanishesTo (cutIdeal hL s) 1 m

theorem **sym_eq_iff** (f : Fil (cutIdeal hL s) a V) (g : SymTarget p B hL s a n) : sym K hL hA hB s hS e he hO h V f = g ↔ HasSymbol K hL hA hB s hS e he hO h f.1 g

theorem **sym_eq_zero_iff** (f : Fil (cutIdeal hL s) a V) : sym K hL hA hB s hS e he hO h V f = 0 ↔ f.1 ∈ Fil (cutIdeal hL s) (a + 1) V

theorem **ker_sym** : LinearMap.ker (sym K hL hA hB s hS e he hO h V) = (Fil (cutIdeal hL s) (a + 1) V).submoduleOf (Fil (cutIdeal hL s) a V)

theorem **sym_sum** (m : MonoDeg (Fin D) a → H0 p (tw (tens K.bundle A) L n) O) (f : Fil (cutIdeal hL s) a V) (hf : f.1 = ∑ α, mulMono (isLB_tens K.isLB hA) hL s α h (m α)) : sym K hL hA hB s hS e he hO h V f = fun α => Res K hL hA hB s hS n e he hO (m α)

theorem **sym_mulMono** (α : MonoDeg (Fin D) a) (m : H0 p (tw (tens K.bundle A) L n) O) (f : Fil (cutIdeal hL s) a V) (hf : f.1 = mulMono (isLB_tens K.isLB hA) hL s α h m) : sym K hL hA hB s hS e he hO h V f = Pi.single α (Res K hL hA hB s hS n e he hO m)

theorem **tr_chartPull** (a : ChartPull B' B r) (z : T) {i j : B'.ι} (hi : z ∈ B'.U i) (hj : z ∈ B'.U j) : tr B' z i j = Units.map (r.stalkMap z).hom.toMonoidHom (tr B (r z) (a.idx i) (a.idx j))

theorem **germ_pullChartFun** (a : ChartPull B' B r) (h : W ≤ r ⁻¹ᵁ O) (f : H0 p B O) (i : B'.ι) (z : T) (hz : z ∈ W) (hi : z ∈ B'.U i) : germ i z hz hi (pullChartFun a q h f) = (r.stalkMap z).hom (germ (a.idx i) (r z) (h hz) (a.le i hi) f)

theorem **stalkMap_const_of_comm** (hq : r ≫ p = q ≫ spMap φ) (z : T) (c : k) : (r.stalkMap z).hom (const p (r z) c) = const q z (φ c)

theorem **germ_pullChart** (a : ChartPull B' B r) (hq : r ≫ p = q ≫ spMap φ) (h : W ≤ r ⁻¹ᵁ O) (f : H0 p B O) (i : B'.ι) (z : T) (hz : z ∈ W) (hi : z ∈ B'.U i) : germ i z hz hi (pullChart a hq h f) = (r.stalkMap z).hom (germ (a.idx i) (r z) (h hz) (a.le i hi) f)

theorem **VanishesTo.pullChart** {I : Y.IdealSheafData} {J : T.IdealSheafData} {n : ℕ} {f : H0 p B O} (hf : VanishesTo I n f) (a : ChartPull B' B r) (hq : r ≫ p = q ≫ spMap φ) (h : W ≤ r ⁻¹ᵁ O) (hI : ∀ z, (stalkId I (r z)).map (r.stalkMap z).hom ≤ stalkId J z) : VanishesTo J n (pullChart a hq h f)

theorem **isLB_triv** : (triv k D).IsLB

theorem **bijective_polyFn** : Function.Bijective (polyFn (k

theorem **ideal_top** (g : Fin D → MvPolynomial (Fin D) k) : (ideal g).ideal ⟨⊤, isAffineOpen_top _⟩ = (Ideal.span (Set.range g)).map polyFn

theorem **vanishesTo_iff** (g : Fin D → MvPolynomial (Fin D) k) (a : ℕ) (f : MvPolynomial (Fin D) k) : VanishesTo (ideal g) a (H0.ofSec (affStr k D) ⊤ (polySec f)) ↔ f ∈ Ideal.span (Set.range g) ^ a

theorem **isIso_of_span** {g : Fin D → MvPolynomial (Fin D) k} (hg : Ideal.span (Set.range g) = Ideal.span (Set.range X)) : IsIso ((ideal g).subschemeι ≫ affStr k D)

theorem **etale_of_span** {g : Fin D → MvPolynomial (Fin D) k} (hg : Ideal.span (Set.range g) = Ideal.span (Set.range X)) : Etale ((ideal g).subschemeι ≫ affStr k D)

theorem **tr_triv** (y) (i j : (triv k D).ι) : tr (triv k D) y i j = 1

theorem **mem_tw_triv** (n : ℤ) (y : Sp (MvPolynomial (Fin D) k)) : y ∈ (tw (triv k D) (triv k D) n).U (twIdx n PUnit.unit PUnit.unit)

theorem **suppIn_one** (I) : SuppIn (nonzeroLocus (one (k

theorem **det_mul_valOn_Res** {R : Type} [CommRing R] {g : Fin D → MvPolynomial (Fin D) k} (hS : Etale ((ideal g).subschemeι ≫ affStr k D)) (n : ℤ) (f : MvPolynomial (Fin D) k) (x : Sp R ⟶ (ideal g).subscheme) (φ : MvPolynomial (Fin D) k →+* R) (hx : x ≫ (ideal g).subschemeι = spMap φ) : φ (Matrix.of fun a b => pderiv b (g a)).det * valOn x (twIdx n PUnit.unit PUnit.unit) (fun _ _ => mem_tw_triv _ _) (Res (affineSpace k D) isLB_triv isLB_triv isLB_triv _ hS n one (suppIn_one _) suppIn_top (form n f)) = φ f

Development.AbelianVarietyTheory.Definitions.Divisors.QuoIff

theorem **quo_iff** (PP : ℝ) (h0 : ℤ) (S eta : ℝ) (u1 : Weight (dat p i)) : Quo p i PP u1 ↔ (wtsAt p i PP h0 S eta u1).IsDiv

Development.AbelianVarietyTheory.Definitions.Divisors.RankEqCardIffExistsFrameMatrixFirstColumnBlock

theorem **apow_mul** (x y : kˣ) (z : ℂ) : apow (x * y) z = apow x z * apow y z

@[simp] theorem **apow_one** (z : ℂ) : apow (1 : kˣ) z = 1

theorem **apow_add** (x : kˣ) (z w : ℂ) : apow x (z + w) = apow x z * apow x w

@[simp] theorem **apow_zero** (x : kˣ) : apow x 0 = 1

theorem **apow_ne_zero** (x : kˣ) (z : ℂ) : apow x z ≠ 0

theorem **apow_neg** (x : kˣ) (z : ℂ) : apow x (-z) = (apow x z)⁻¹

theorem **apow_inv** (x : kˣ) (z : ℂ) : apow x⁻¹ z = apow x (-z)

theorem **nabs_of_valuation_eq_one** {x : kˣ} (hx : valuation k (x : k) = 1) : nabs k x = 1

theorem **apow_of_valuation_eq_one** {x : kˣ} (hx : valuation k (x : k) = 1) (z : ℂ) : apow x z = 1

theorem **CondLE.div** {ϖ : kˣ} {e : ℕ} {α β : kˣ →* ℂˣ} (hα : CondLE ϖ e α) (hβ : CondLE ϖ e β) : CondLE ϖ e (α / β)

theorem **CondLE.mul** {ϖ : kˣ} {e : ℕ} {α β : kˣ →* ℂˣ} (hα : CondLE ϖ e α) (hβ : CondLE ϖ e β) : CondLE ϖ e (α * β)

theorem **condLE_of_isUnram** (ϖ : kˣ) (e : ℕ) {χ : kˣ →* ℂˣ} (h : IsUnram χ) : CondLE ϖ e χ

theorem **IsRightTranslationIsometric.condν** {ϖ : kˣ} {ψ : AddChar k Circle} {α β : kˣ →* ℂˣ} {e : ℕ} (h : IsRightTranslationIsometric ϖ ψ α β e) : CondLE ϖ e (α / β)

theorem **IsRightTranslationIsometric.inv** {ϖ : kˣ} {ψ : AddChar k Circle} {α β : kˣ →* ℂˣ} {e : ℕ} (h : IsRightTranslationIsometric ϖ ψ α β e) : IsRightTranslationIsometric ϖ ψ⁻¹ α β e

theorem **IsCoeffRing.inv** {ϖ : kˣ} {ψ : AddChar k Circle} {α β : kˣ →* ℂˣ} {e : ℕ} {R : Subring ℂ} (h : IsCoeffRing ϖ ψ α β e R) : IsCoeffRing ϖ ψ⁻¹ α β e R

theorem **exists_isUnif** : ∃ ϖ : kˣ, IsUnif ϖ

theorem **IsUnif.mem** {ϖ : kˣ} (hϖ : IsUnif ϖ) : (ϖ : k) ∈ 𝒪[k]

theorem **IsUnif.maximalIdeal_eq** {ϖ : kˣ} (hϖ : IsUnif ϖ) : IsLocalRing.maximalIdeal 𝒪[k] = Ideal.span {(⟨ϖ, hϖ.mem⟩ : 𝒪[k])}

theorem **IsUnif.irreducible** {ϖ : kˣ} (hϖ : IsUnif ϖ) : Irreducible (⟨ϖ, hϖ.mem⟩ : 𝒪[k])

theorem **exists_ramified** {ϖ : kˣ} (hϖ : IsUnif ϖ) : ∃ ν : kˣ →* ℂˣ, ¬ IsUnram ν ∧ CondLE ϖ 2 ν ∧ ∀ x, ‖(ν x : ℂ)‖ = 1

theorem **exists_localSetting** : ∃ (ϖ : kˣ) (ψ : AddChar k Circle) (α : kˣ →* ℂˣ), IsRightTranslationIsometric ϖ ψ α 1 2

theorem **invT_val** (h : GL D k) : (invT h).val = (h⁻¹).valᵀ

theorem **invT_inv_val** (h : GL D k) : ((invT h)⁻¹).val = h.valᵀ

@[simp] theorem **invT_invT** (h : GL D k) : invT (invT h) = h

theorem **gdet_mul** (g h : GL D k) : gdet (g * h) = gdet g * gdet h

theorem **gdet_invT** (h : GL D k) : gdet (invT h) = (gdet h)⁻¹

theorem **valuation_gdet_of_mem_KO** {g : GL D k} (hg : g ∈ KO k D) : valuation k (gdet g : k) = 1

theorem **invT_mem_KO** {g : GL D k} (hg : g ∈ KO k D) : invT g ∈ KO k D

theorem **omA_mul** (ν β : kˣ →* ℂˣ) (g g' : GL A k) (h h' : GL D k) (Φ : Matrix A D k → ℂ) : omA ν β (g * g') (h * h') Φ = omA ν β g h (omA ν β g' h' Φ)

theorem **omB_mul** (ν α : kˣ →* ℂˣ) (g g' : GL A' k) (h h' : GL D k) (Ψ : Matrix A' D k → ℂ) : omB ν α (g * g') (h * h') Ψ = omB ν α g h (omB ν α g' h' Ψ)

theorem **comp_mul_mem_SB** (g : GL A k) (h : GL D k) {Φ : Matrix A D k → ℂ} (hΦ : Φ ∈ SB (Matrix A D k)) : (fun X => Φ (g.val * X * h.val)) ∈ SB (Matrix A D k)

theorem **omA_mem_SB** (ν β : kˣ →* ℂˣ) (g : GL A k) (h : GL D k) {Φ : Matrix A D k → ℂ} (hΦ : Φ ∈ SB (Matrix A D k)) : omA ν β g h Φ ∈ SB (Matrix A D k)

theorem **omB_mem_SB** (ν α : kˣ →* ℂˣ) (g : GL A' k) (h : GL D k) {Ψ : Matrix A' D k → ℂ} (hΨ : Ψ ∈ SB (Matrix A' D k)) : omB ν α g h Ψ ∈ SB (Matrix A' D k)

theorem **frameMatrixFirstColumnBlock_mul_apply** {B : Type*} (σ : A ⊕ A' ≃ D) (M : Matrix D B k) (i : A) (j : B) : (X0 k σ * M) i j = M (σ (Sum.inl i)) j

theorem **frameMatrixSecondColumnBlock_mul_apply** {B : Type*} (σ : A ⊕ A' ≃ D) (M : Matrix D B k) (i : A') (j : B) : (Y0 k σ * M) i j = M (σ (Sum.inr i)) j

theorem **frameSecondBlock_mul_frameFirstBlock_transpose_eq_zero** (σ : A ⊕ A' ≃ D) (h : GL D k) : (Y0 k σ * h.val) * (X0 k σ * (invT h).val)ᵀ = 0

theorem **rank_eq_card_iff_exists_frameMatrixFirstColumnBlock** (σ : A ⊕ A' ≃ D) (X : Matrix A D k) : X.rank = Fintype.card A ↔ ∃ h : GL D k, X = X0 k σ * (invT h).val

theorem **mem_smul_lat** (c : kˣ) (Λ : Submodule 𝒪[k] (D → k)) (x : D → k) : x ∈ c • Λ ↔ c⁻¹ • x ∈ Λ

theorem **mem_matLat** {Λ : Submodule 𝒪[k] (D → k)} {X : Matrix A D k} : X ∈ matLat A Λ ↔ ∀ i, X.row i ∈ Λ

theorem **mem_diagLat** {c : D → kˣ} {x : D → k} : x ∈ diagLat c ↔ ∀ j, valuation k (x j) ≤ valuation k (c j : k)

theorem **mem_stdLat** {x : D → k} : x ∈ stdLat k D ↔ ∀ j, x j ∈ 𝒪[k]

theorem **mem_matLat_stdLat** {X : Matrix A D k} : X ∈ matLat A (stdLat k D) ↔ ∀ i j, X i j ∈ 𝒪[k]

theorem **smul_diagLat** (u : kˣ) (c : D → kˣ) : u • diagLat c = diagLat fun j => u * c j

theorem **mem_dualLat** {Λ : Submodule 𝒪[k] (D → k)} {x : D → k} : x ∈ dualLat Λ ↔ ∀ y ∈ Λ, x ⬝ᵥ y ∈ 𝒪[k]

theorem **dualLat_diagLat** (c : D → kˣ) : dualLat (diagLat c) = diagLat fun j => (c j)⁻¹

theorem **isPrimitiveFrame_standardBasis** [Fintype A] [DecidableEq D] {A' : Type*} [Fintype A'] (σ : A ⊕ A' ≃ D) : IsPrimFrame (stdLat k D) (X0 k σ)

theorem **mem_glLat** {Λ : Submodule 𝒪[k] (D → k)} {h : GL D k} : h ∈ glLat Λ ↔ ∀ x, x ᵥ* h.val ∈ Λ ↔ x ∈ Λ

theorem **mem_glLat_iff_image** {Λ : Submodule 𝒪[k] (D → k)} {h : GL D k} : h ∈ glLat Λ ↔ (fun x => x ᵥ* h.val) '' Λ = Λ

theorem **glLat_stdLat** : glLat (stdLat k D) = KO k D

theorem **vecMul_blockScale_mem** (ϖ : kˣ) (e : ℕ) (P : Finset D) (x : D → k) : x ᵥ* (blockScale ϖ e P).val ∈ blockLat ϖ e P ↔ x ∈ stdLat k D

theorem **blockLat_empty** (ϖ : kˣ) (e : ℕ) : blockLat ϖ e (∅ : Finset D) = stdLat k D

theorem **latA_blockLat** (ϖ : kˣ) (e : ℕ) (P : Finset D) : latA ϖ e (blockLat ϖ e P) = blockLat ϖ e Pᶜ

theorem **pow_smul_le** {ϖ : kˣ} (hϖ : IsUnif ϖ) (e : ℕ) (Λ : Submodule 𝒪[k] (D → k)) : ϖ ^ e • Λ ≤ Λ

theorem **hasDepth_indicator** {ϖ : kˣ} (hϖ : IsUnif ϖ) (e : ℕ) (Λ : Submodule 𝒪[k] (D → k)) : HasDepth ϖ e Λ ((matLat A Λ : Set (Matrix A D k)).indicator 1)

theorem **liftRes_mem** {B : Type*} {e : ℕ} (T : A → B → Res k e) (i : A) (j : B) : liftRes T i j ∈ 𝒪[k]

theorem **pairCols_eq_trace** (P : Finset D) (Y : Matrix A D k) (T : Matrix A P k) : pairCols P Y T = trace (T * (Y.submatrix id (Subtype.val : P → D))ᵀ)

theorem **fourierSum_apply** (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (P : Finset D) (Φ : Matrix A D k → ℂ) (Y : Matrix A D k) : fourierSum ψ ϖ e P Φ Y = if ∀ i, ∀ j ∈ P, ((ϖ ^ e : kˣ) : k) * Y i j ∈ 𝒪[k] then ((resCard k : ℂ) ^ (e * Fintype.card A * P.card))⁻¹ * ∑ T : A → P → Res k e, Φ (setCols P Y (liftRes T)) * (ψ (pairCols P Y (liftRes T)) : ℂ) else 0

theorem **invFourierSum_apply** (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (P : Finset D) (Ψ : Matrix A D k → ℂ) (T : Matrix A D k) : invFourierSum ψ ϖ e P Ψ T = if ∀ i, ∀ j ∈ P, T i j ∈ 𝒪[k] then ∑ S : A → P → Res k e, Ψ (setCols P T ((ϖ ^ e)⁻¹ • liftRes S)) * (ψ (-pairCols P T ((ϖ ^ e)⁻¹ • liftRes S)) : ℂ) else 0

theorem **invFourierSum_inv_apply** (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (P : Finset D) (Ψ : Matrix A D k → ℂ) (T : Matrix A D k) : invFourierSum ψ⁻¹ ϖ e P Ψ T = if ∀ i, ∀ j ∈ P, T i j ∈ 𝒪[k] then ∑ S : A → P → Res k e, Ψ (setCols P T ((ϖ ^ e)⁻¹ • liftRes S)) * (ψ (pairCols P T ((ϖ ^ e)⁻¹ • liftRes S)) : ℂ) else 0

theorem **invFourierSum_inv_eq_neg** (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (P : Finset D) (Ψ : Matrix A D k → ℂ) (T : Matrix A D k) : invFourierSum ψ⁻¹ ϖ e P Ψ T = invFourierSum ψ ϖ e P Ψ (setCols P T (Matrix.of fun i j => -T i j))

theorem **fourierSum_empty** (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (Φ : Matrix A D k → ℂ) : fourierSum ψ ϖ e ∅ Φ = Φ

theorem **invFourierSum_empty** (ψ : AddChar k Circle) (ϖ : kˣ) (e : ℕ) (Φ : Matrix A D k → ℂ) : invFourierSum ψ ϖ e ∅ Φ = Φ

theorem **mem_shellSet** {ϖ : kˣ} {n : A → ℤ} {g : GL A k} : g ∈ shellSet ϖ n ↔ ∃ x ∈ KO k A, ∃ y ∈ KO k A, g = x * (diagGL fun i => ϖ ^ n i) * y

theorem **shellSet_zero** (ϖ : kˣ) : shellSet ϖ (0 : A → ℤ) = KO k A

theorem **shellSum_smul** {a : ℕ} (μ : Measure (GL (Fin a) k)) (ϖ : kˣ) (ν : kˣ →* ℂˣ) (z c : ℂ) (φ : Matrix (Fin a) (Fin a) k → ℂ) : shellSum μ ϖ ν z (fun g => c * φ g) = c * shellSum μ ϖ ν z φ

theorem **bigZetaIntegral_eq** {a : ℕ} (μ : Measure (GL (Fin a) k)) (ϖ : kˣ) (σ : Fin a ⊕ A' ≃ D) (ν β : kˣ →* ℂˣ) (Φ : Matrix (Fin a) D k → ℂ) (h : GL D k) : bigP μ ϖ σ ν β Φ h = (β (gdet h) : ℂ) * apow (gdet h) (-(a : ℂ) / 2) * bigF μ ϖ ν Φ (X0 k σ * (invT h).val)

theorem **smallP_eq** (μ : Measure (GL A' k)) (σ : A ⊕ A' ≃ D) (ν α : kˣ →* ℂˣ) (Ψ : Matrix A' D k → ℂ) (h : GL D k) : smallP μ σ ν α Ψ h = smallF μ ν (omB ν α 1 h Ψ) (Y0 k σ)

theorem **smallF_mul_left** (μ : Measure (GL A' k)) [μ.IsHaarMeasure] (ν : kˣ →* ℂˣ) (Ψ : Matrix A' D k → ℂ) (g_0 : GL A' k) (Y : Matrix A' D k) : (ν (gdet g_0) : ℂ) * apow (gdet g_0) ((Fintype.card D : ℂ) / 2) * smallF μ ν Ψ (g_0.val * Y) = smallF μ ν Ψ Y

theorem **exists_secondBlock_of_frameFirstBlock_eq** (σ : A ⊕ A' ≃ D) {h h' : GL D k} (H : X0 k σ * (invT h').val = X0 k σ * (invT h).val) : ∃ C : GL A' k, Y0 k σ * h'.val = C.val * (Y0 k σ * h.val) ∧ gdet h' = gdet C * gdet h

theorem **splitTheta_fFun_frameFirstBlock_mul** (μ : Measure (GL A' k)) [μ.IsHaarMeasure] (σ : A ⊕ A' ≃ D) (ν : kˣ →* ℂˣ) (Ψ : Matrix A' D k → ℂ) (h : GL D k) : fFun μ σ ν Ψ (X0 k σ * (invT h).val) = (ν (gdet h) : ℂ) * apow (gdet h) ((Fintype.card D : ℂ) / 2) * smallF μ ν Ψ (Y0 k σ * h.val)

theorem **shell_zero_example** (ϖ : kˣ) (ν : kˣ →* ℂˣ) (z : ℂ) (φ : Matrix A A k → ℂ) (hφ : ∀ g ∈ KO k A, φ g.val * (ν (gdet g) : ℂ) = 1) : shell (haarVol (KO k A)) ϖ ν z φ 0 = 1

theorem **smallF_of_isEmpty** [IsEmpty A'] (ν : kˣ →* ℂˣ) (Ψ : Matrix A' D k → ℂ) (Y : Matrix A' D k) : smallF (haarVol (KO k A')) ν Ψ Y = Ψ 0

theorem **schwartzDatum_standardBasis_of_isEmpty** [IsEmpty A'] (σ : A ⊕ A' ≃ D) (ν : kˣ →* ℂˣ) (ϖ : kˣ) (e : ℕ) (Ψ : Matrix A' D k → ℂ) : phiA (haarVol (KO k A')) σ ν ϖ e (blockLat ϖ e Finset.univ) Ψ (X0 k σ) = Ψ 0

theorem **mem_blockLat_two** (ϖ : kˣ) (e : ℕ) (x : D₁ ⊕ D₂ → k) : x ∈ blockLat ϖ e (Assignment.two.block D₁ D₂) ↔ (∀ j, valuation k (x (.inl j)) ≤ 1) ∧ ∀ j, valuation k (x (.inr j)) ≤ valuation k (((ϖ ^ e)⁻¹ : kˣ) : k)

theorem **mem_blockLat_one** (ϖ : kˣ) (e : ℕ) (x : D₁ ⊕ D₂ → k) : x ∈ blockLat ϖ e (Assignment.one.block D₁ D₂) ↔ (∀ j, valuation k (x (.inl j)) ≤ valuation k (((ϖ ^ e)⁻¹ : kˣ) : k)) ∧ ∀ j, valuation k (x (.inr j)) ≤ 1

theorem **compl_block_two** : (Assignment.two.block D₁ D₂)ᶜ = Assignment.one.block D₁ D₂

theorem **compl_block_one** : (Assignment.one.block D₁ D₂)ᶜ = Assignment.two.block D₁ D₂

theorem **rank_square** (i j : ℕ) : G.rank (i + 1, j + 1) + G.rank (i, j) = G.rank (i, j + 1) + G.rank (i + 1, j)

theorem **alpha_div_beta** (i j : ℕ) : G.alpha i j / G.beta i j = G.nu i j

theorem **kappa_inner** (i j : ℕ) : G.kappa (.ex i j) (i, j) / G.kappa (.theta i j) (i, j) = (G.nu i j)⁻¹

theorem **kappa_outer** (i j : ℕ) : G.kappa (.theta (i + 1) j) (i + 1, j + 1) / G.kappa (.ex i (j + 1)) (i + 1, j + 1) = G.nu i j

theorem **exists_isGood** (m r t s : ℕ) (ht : Odd t) (hs : Odd s) : ∃ (G : Grid k) (ϖ : kˣ) (ψ : AddChar k Circle), G.m = m ∧ G.r = r ∧ G.t = t ∧ G.s = s ∧ G.IsGood ϖ ψ 2

theorem **pathAct_one** (s : Assignment) (γ : List Edge) (T : G.Args → ℂ) : G.pathAct s γ 1 T = T

theorem **pathAct_mul** (s : Assignment) (γ : List Edge) (g g' : G.VertexGroup) (T : G.Args → ℂ) : G.pathAct s γ (g * g') T = G.pathAct s γ g (G.pathAct s γ g' T)

theorem **isTensor_pureTensor** (ϖ : kˣ) (e : ℕ) (γ : List Edge) (φ : ∀ b : Edge, Matrix (Fin (G.rank b.tgt)) (Fin (G.rank b.src)) k → ℂ) (hφ : ∀ b ∈ γ, IsNatDatum ϖ e (φ b)) : G.IsTensor ϖ e γ (G.lines₂ γ φ)

theorem **isOpen_detCong** (c : kˣ) (n : Type*) [Fintype n] [DecidableEq n] : IsOpen (detCong c n : Set (GL n k))

theorem **mul_mul_mem_matLat_iff** {m n : Type*} [Fintype m] [DecidableEq m] [Fintype n] [DecidableEq n] {g : GL m k} {h : GL n k} (hg : g ∈ KO k m) (hh : h ∈ KO k n) (X : Matrix m n k) : g.val * X * h.val ∈ matLat m (stdLat k n) ↔ X ∈ matLat m (stdLat k n)

theorem **Assignment.read_mem_KO** (s : Assignment) {n : Type*} [Fintype n] [DecidableEq n] {g : GL n k} (hg : g ∈ KO k n) : s.read g ∈ KO k n

theorem **Grid.IsGood.condLE_kappa** {G : Grid k} {ϖ : kˣ} {ψ : AddChar k Circle} {e : ℕ} (h : G.IsGood ϖ ψ e) {b : Edge} (hb : b.InGrid G.m) (X : ℕ × ℕ) : CondLE ϖ e (G.kappa b X)

theorem **Grid.IsGood.exists_isAdmissible** {G : Grid k} {ϖ : kˣ} {ψ : AddChar k Circle} {e : ℕ} (h : G.IsGood ϖ ψ e) (hϖ : IsUnif ϖ) (s : Assignment) {γ : List Edge} (hγ : IsPath γ) (hm : ∀ b ∈ γ, b.InGrid G.m) : ∃ S : G.PiLocalization₂, S.path = γ ∧ S.IsAdmissible s ϖ e ∧ S.tensor 0 = 1

theorem **isOpen_map_invT** {n : Type*} [Fintype n] [DecidableEq n] {J : Subgroup (GL n k)} (ho : IsOpen (J : Set (GL n k))) (hK : J ≤ KO k n) : IsOpen (J.map invT : Set (GL n k)) ∧ J.map invT ≤ KO k n

Development.AbelianVarietyTheory.Definitions.Divisors.RatBasisLinearIndependentInt

theorem **isPreDP_dpow** (δ : DividedPowers J) : IsPreDP J δ.dpow

theorem **isLev_top_of_subsingleton** [Subsingleton B] (δ : DividedPowers J) (N : ℕ) : IsLev δ N

theorem **fil_eq_dpFil** (δ : DividedPowers J) (N : ℕ) : fil δ N = GlobalTheory.Cris.ExpBij.dpFil δ N

theorem **_root_.GlobalTheory.PDA.isLev_iff_prod** (δ : DividedPowers J) (N : ℕ) : _root_.GlobalTheory.PDA.IsLev δ N ↔ ∀ l : List (ℕ × B), (∀ p ∈ l, p.2 ∈ J) → N ≤ (l.map Prod.fst).sum → (l.map fun p ↦ δ.dpow p.1 p.2).prod = 0

theorem **_root_.GlobalTheory.PDA.isDPNilpotent_iff_exists_isLevel** (δ : DividedPowers J) : _root_.GlobalTheory.Cris.IsNilDP δ ↔ ∃ N, _root_.GlobalTheory.PDA.IsLev δ N

theorem **_root_.GlobalTheory.PDA.fil_zero_eq_top** (δ : DividedPowers J) : _root_.GlobalTheory.PDA.fil δ 0 = ⊤

private lemma **prod_dpow_mem_of_one_le_sum** (δ : DividedPowers J) : ∀ l : List (ℕ × B), (∀ p ∈ l, p.2 ∈ J) → 1 ≤ (l.map Prod.fst).sum → (l.map fun p ↦ δ.dpow p.1 p.2).prod ∈ J | [], _, h => by simp at h | p :: l, hl, h => by rw [List.map_cons, List.prod_cons] by_cases hp : p.1 = 0 · refine J.mul_mem_left _ (prod_dpow_mem_of_one_le_sum δ l (fun q hq => hl q (List.mem_cons_of_mem _ hq)) ?_) simpa [hp] using h · exact J.mul_mem_right _ (δ.dpow_mem hp (hl p (by simp))) set_option autoImplicit false in open DividedPowers in variable {B : Type} [CommRing B] {J : Ideal B} in attribute [instance] _root_.GlobalTheory.PDA.PDPoly.ring _root_.GlobalTheory.PDA.PDPoly.alg in theorem _root_.GlobalTheory.PDA.fil_one_eq (δ : DividedPowers J) : _root_.GlobalTheory.PDA.fil δ 1 = J

theorem **isLev_one_iff_eq_bot** (δ : DividedPowers J) : IsLev δ 1 ↔ J = ⊥

theorem **isLev_sqDP** (h : J ^ 2 = 0) : IsLev (GlobalTheory.Cris.divided₂ h) 2

theorem **exists_isLev_two_not_one** : ∃ δ : DividedPowers (Ideal.span {(2 : ZMod 4)}), IsLev δ 2 ∧ ¬ IsLev δ 1

theorem **_root_.GlobalTheory.PDA.PDPoly.e_succ_mem_aug** (Q : _root_.GlobalTheory.PDA.PDPoly B) (n : ℕ) : Q.e (n + 1) ∈ Q.aug

theorem **_root_.GlobalTheory.PDA.PDPoly.e_one_mem_ext** (Q : _root_.GlobalTheory.PDA.PDPoly B) (J : Ideal B) : Q.e 1 ∈ Q.ext J

theorem **_root_.GlobalTheory.PDA.PDPoly.algebraMap_mem_ext** (Q : _root_.GlobalTheory.PDA.PDPoly B) {a : B} (ha : a ∈ J) : algebraMap B Q.P a ∈ Q.ext J

theorem **PDPoly.e_one_pow** (Q : PDPoly B) (n : ℕ) : Q.e 1 ^ n = (n.factorial : Q.P) * Q.e n

theorem **PDPoly.IsStd.not_isLev** [Nontrivial B] {Q : PDPoly B} {δ : DividedPowers J} {η : DividedPowers (Q.ext J)} (h : Q.IsStd δ η) (N : ℕ) : ¬ IsLev η N

theorem **PDPoly.exists_isStd_of_rat** [Algebra ℚ B] (Q : PDPoly B) (δ : DividedPowers J) : ∃ η : DividedPowers (Q.ext J), Q.IsStd δ η

theorem **ratBasis_apply** (n : ℕ) : ratBasis n = C ((n.factorial : ℚ)⁻¹) * X ^ n

private lemma **ratBasis_eq_C_inv_factorial_mul_X_pow** (n : ℕ) : _root_.GlobalTheory.PDA.ratBasis n = C ((n.factorial : ℚ)⁻¹) * X ^ n

theorem **_root_.GlobalTheory.PDA.ratBasis_zero** : _root_.GlobalTheory.PDA.ratBasis 0 = 1

theorem **_root_.GlobalTheory.PDA.ratBasis_mul** (m n : ℕ) : _root_.GlobalTheory.PDA.ratBasis m * _root_.GlobalTheory.PDA.ratBasis n = ((m + n).choose m : ℚ[X]) * _root_.GlobalTheory.PDA.ratBasis (m + n)

theorem **ratPoly_isStd** : ratPoly.IsStd (dividedPowersBot ℚ) (RatAlgebra.dividedPowers (ratPoly.ext ⊥))

theorem **_root_.GlobalTheory.PDA.ratBasis_linearIndependent_int** : LinearIndependent ℤ _root_.GlobalTheory.PDA.ratBasis

theorem **intBasis_coe** (n : ℕ) : (intBasis n : ℚ[X]) = ratBasis n

private lemma **intBasis_cast_eq_ratBasis** (n : ℕ) : (_root_.GlobalTheory.PDA.intBasis n : ℚ[X]) = _root_.GlobalTheory.PDA.ratBasis n

theorem **_root_.GlobalTheory.PDA.intBasis_zero** : _root_.GlobalTheory.PDA.intBasis 0 = 1

theorem **_root_.GlobalTheory.PDA.intBasis_mul** (m n : ℕ) : _root_.GlobalTheory.PDA.intBasis m * _root_.GlobalTheory.PDA.intBasis n = ((m + n).choose m : _root_.GlobalTheory.PDA.intForm) * _root_.GlobalTheory.PDA.intBasis (m + n)

theorem **stdPoly_two** : (stdPoly (ZMod 2)).e 1 ^ 2 = 0 ∧ (stdPoly (ZMod 2)).e 2 ≠ 0

Development.AbelianVarietyTheory.Definitions.Divisors.ReadsOnlyOmegaFree

theorem **topL_stdM** {κ : Type} [Zero κ] [One κ] {b a : ℕ} (h : a ≤ b) : topL a (stdM κ b a) = 1

theorem **eq_stdM_iff** {κ : Type} [Zero κ] [One κ] {b a : ℕ} (h : a ≤ b) (y : Matrix (Fin b) (Fin a) κ) : (LowZero a y ∧ topL a y = 1) ↔ y = stdM κ b a

theorem **rank_W_le_V** : (Wc D i).rank ≤ (Vc D i).rank

theorem **readsOnly_omegaFree** (Q : Finset (Pl D.host.E)) (ξR : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (y₀ : ∀ PP : Pl D.host.E, Matrix (Fin (Vc D i).rank) (Fin (Wc D i).rank) (kP PP)) : ReadsOnly (R.Pg ∪ Q) (omegaFree R i Q ξR y₀)

theorem **omegaK_mul_omegaQ** (Q : Finset (Pl D.host.E)) (ξR : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (y : ∀ PP : Pl D.host.E, Matrix (Fin (Vc D i).rank) (Fin (Wc D i).rank) (kP PP)) : omegaK R.Pg y * omegaQ Q (Wc D i).rank ξR (-1) y = (if ∀ PP ∈ Q, LowZero (Wc D i).rank (y PP) then 1 else 0) * omegaFree R i Q ξR (fun PP => stdM (kP PP) _ _) y

theorem **omegaFree_ne_zero** (Q : Finset (Pl D.host.E)) (hd : Disjoint R.Pg Q) (ξR : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (y₀ : ∀ PP : Pl D.host.E, Matrix (Fin (Vc D i).rank) (Fin (Wc D i).rank) (kP PP)) : ∃ y, omegaFree R i Q ξR y₀ y ≠ 0

theorem **phiOf_iff_phiIs** (t : T0) {La : Type} [AddCommGroup La] [Module (𝒪 D.host.E) La] (ι : La →ₗ[𝒪 D.host.E] (Fin (Wc D i).rank → D.host.E)) (N : Submodule (𝒪 D.host.E) La) (φ : (Fin (Vc D i).rank → La ⧸ N) → ℂ) : PhiOf R i (R.Q t) (fun y => omegaK R.Pg y * omegaQ (R.Q t) (Wc D i).rank (R.xiR t) (-1) y) ι N φ ↔ PhiIs R i (RingHom.id ℂ) t ι N φ

theorem **VecOf.smul** {Q : Finset (Pl D.host.E)} {Ω : DatFn D i} {φ : WO (Vc D i) (Wc D i) → ℂ} (h : VecOf R i Q Ω φ) (c : ℂ) : VecOf R i Q (c • Ω) (c • φ)

Development.AbelianVarietyTheory.Definitions.Divisors.RedSurjective

theorem **p_eq_zero** (p : ℕ) (k : Type) [Field k] [CharP k p] : (p : k) = 0

Development.AbelianVarietyTheory.Definitions.Divisors.ReferenceMetricMulDualEqOne

theorem **posAt_of_posSet** {F : Type} [Field F] [NumberField F] {Y : Var F} {σ : F →+* ℂ} {Δ : Y.Dv} {m : Var.Met Y σ Δ} (h : m.posSet = Set.univ) (z : Pts Y σ) : PosAt (C

theorem **pn_met_mul_neg** {Φ : s.Flag} {k : s.Comp} {σ : D.what₂.LL →+* ℂ} (G : EigMet s.T.Z (embOf κ.e σ)) (n : Letter S.Dw → ℤ) {z : Pts (Φ.C k).Y σ} {f : (Φ.C k).Y.Fn} (hf : (divOf (Φ.C k) n).GeneratesAt (under z) f) : (cxSt _).pn _ σ _ ((Φ.C k).met G n) ((cxSt _).cx _ σ f) z * (cxSt _).pn _ σ _ ((Φ.C k).met G (-n)) ((cxSt _).cx _ σ f⁻¹) z = 1

theorem **coreWith_withCore** (x : ToroidalCompactificationData O.s) (P : PolOf x O.Φ) (h : ∃ B, IsBdOf x O.L B) : (O.withCore x P h).CoreWith x P

theorem **coreWith_ofKit** {A : Amb D} {N j : ℕ} (h : 1 ≤ j ∧ j < N) {S : Sig A N j} {κ : Lv A} {s : FinalTheorems S κ} {L : CmList A} {jC : ∀ (p : ℕ) [Fact p.Prime], PlL D p → (DV.KW p →+* ℂ)} {ng : NgPkg} (k : Kit s L (cxSt D.what₂.LL) jC ng) (adm : AdmSt D N j jC) {x : ToroidalCompactificationData s} {P : PolOf x k.Φ} (hB : IsBdOf x L k.B) (hf : k.fl = flBetti s k.Φ L) (hM : k.Mb = mbarOf P) : (ofKit h k adm (metTrue x k.Φ P)).CoreWith x P

theorem **hm_eq** : O.met.hm = metsOf x O.Φ

theorem **h0_eq** : O.met.h0 = refMet x O.Φ

theorem **r_eq** : O.met.r = rFlag x O.Φ

theorem **ab_eq** : O.met.ab = P.ab

theorem **dir_eq** : O.met.dir = P.dir

theorem **reg_eq** : O.met.reg = P.reg

theorem **two_mul_chg** (c : O.s.Comp) (u : D.what₂.Tau → Fin O.N → ℝ) (σ : D.what₂.LL →+* ℂ) (z : Pts (O.var c) σ) : 2 * O.chg c none u σ z = phW (metsOf x O.Φ) (refMet x O.Φ) c σ u z

theorem **mbIs** : O.MbIs

theorem **mbIs** : O.MbIs

theorem **strictAnti_dir** (τ : D.what₂.Tau) : StrictAnti (O.met.dir τ)

theorem **ample_ab** : (O.var c).ample (O.met.ab c).D

theorem **exp_le_r** (z : Pts (O.var c) σ) : Real.exp 1 ≤ O.met.r c σ z

theorem **referenceMetric_mul_dual_eq_one** (n : Letter O.S.Dw → ℤ) {z : Pts (O.var c) σ} {f : (O.var c).Fn} (hf : (divOf (O.Φ.C c) n).GeneratesAt (under z) f) : (cxSt _).pn _ σ _ (O.met.h0 c σ n) ((cxSt _).cx _ σ f) z * (cxSt _).pn _ σ _ (O.met.h0 c σ (-n)) ((cxSt _).cx _ σ f⁻¹) z = 1

theorem **referenceMetric_posAt** (n : Letter O.S.Dw → ℤ) (z : Pts (O.var c) σ) : PosAt (O.met.h0 c σ n) z

theorem **posAt_hm** (R : Option ℝ) (n : Letter O.S.Dw → ℤ) (z : Pts (O.var c) σ) : PosAt (O.met.hm c R σ n) z

theorem **chg_some_eq_zero** {R : ℝ} (hR : 2 ≤ R) (u : D.what₂.Tau → Fin O.N → ℝ) {z : Pts (O.var c) σ} (hz : (O.Φ.C c).π (under z) ∈ genBd O.s.T.Z ∨ R ^ 3 ≤ O.met.r c σ z) : O.chg c (some R) u σ z = 0

theorem **ObjCore.withPs** {D : D7} {Y : DV.ShimX D.what₂.Tau} {A : SlopeTestRemainingChoices Y} {T : Real0 Y} {H : Ht Y} (o : ObjCore Y A T H) (Ps : ℝ → Y.Comp → MvPolynomial (Option (D.what₂.Tau × Fin Y.N)) ℝ) : ObjCore Y { A with Ps

Development.AbelianVarietyTheory.Definitions.Divisors.SRelOneOneIff

theorem **sRel_one_one_iff** {h n : ℕ} {B' B : Type} [CommRing B'] [CommRing B] {ζ' : B'} {ζ : B} {S' S : Scheme.{0}} {s' : S' ⟶ Sp B'} {s : S ⟶ Sp B} (c' : SiegPre h n ζ' s') (c : SiegPre h n ζ s) (f : S' ⟶ S) : SRel c' c 1 1 f ↔ SiegPB c' c f

Development.AbelianVarietyTheory.Definitions.Divisors.SameDiv

theorem **SameDiv.refl** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) (σ : L.Sec) : SameDiv σ σ

theorem **SameDiv.symm** {X : Scheme.{0}} {L M : LB X} {σ : L.Sec} {τ : M.Sec} (h : SameDiv σ τ) : SameDiv τ σ

Development.AbelianVarietyTheory.Definitions.Divisors.SameIdIff

theorem **Rep.of_isDual** {B : Over S} [GrpObj B] {P : LB (A ⊗ B).left} (h : IsDual A B P) : Rep A (toUnit B) P where

theorem **same_id_iff** {T U : Over S} {N : LB (A ⊗ T).left} {M : LB (A ⊗ U).left} (hN : N.IsLB) (hM : M.IsLB) (u : T ⟶ U) : Same A N M (𝟙 T) u ↔ LBIso N (ev M (fst A T) (snd A T ≫ u))

theorem **same_iff** {T U Y : Over S} {N : LB (A ⊗ T).left} {M : LB (A ⊗ U).left} (hN : N.IsLB) (hM : M.IsLB) (y : Y ⟶ T) (u : Y ⟶ U) : Same A N M y u ↔ evc hN (fst A Y) (snd A Y ≫ y) = evc hM (fst A Y) (snd A Y ≫ u)

Development.AbelianVarietyTheory.Definitions.Divisors.SecMapApply

theorem **secMap_apply** (om : G × H →* Module.End ℂ (Y → ℂ)) (l : (Y → ℂ) →ₗ[ℂ] ℂ) (φ : Y → ℂ) (h : H) : secMap om l φ h = l (om (1, h) φ)

Development.AbelianVarietyTheory.Definitions.Divisors.SecSpaceBotNeBot

theorem **ind_apply** (w g : UA σ J₂) (𝒟 : Set (unipN σ J₂)) : (fun x : unipN σ J₂ => ind w g 𝒟 (w * x * g)) = 𝒟.indicator fun _ => 1

theorem **exists_intM_eq_one** {νN : Measure (unipN σ J₂)} (h : IsNormN σ νN) (w g : UA σ J₂) : ∃ f, IntConv νN w f g ∧ intM νN w f g = 1

theorem **coefM_pole** {νN : Measure (unipN σ J₂)} {w : UA σ J₂} {f : UA σ J₂ → ℂ} (hf : ∀ g, IntConv νN w f g) : HasIntCont νN w (fun s => s⁻¹ • f) ∧ (∀ g, coefM νN w (fun s => s⁻¹ • f) 0 (-1) g = intM νN w f g) ∧ (∀ g, contM νN w (fun s => s⁻¹ • f) 0 g = 0) ∧ ∀ g, contM νN w (fun s => s⁻¹ • f) (-1) g = -intM νN w f g

theorem **exists_ratN** (hσ : ∀ x, σ (σ x) = x) {J : Matrix (Fin n) (Fin n) E} {l : E} (hl : l ≠ 0) (hJ : ∀ i j, σ ((l⁻¹ • J) j i) = -(l⁻¹ • J) i j) (h₂ : J₂ = dblGram finSumFinEquiv J) {T : GL (Fin (n + n)) E} (h : IsWittBasis σ J l T) (c : ℕ) : ∃ u : ratN σ J₂, inBasis σ T ((u : unipN σ J₂) : UA σ J₂) = blk 1 (c • 1) 0 1

theorem **infinite_ratN** (hσ : ∀ x, σ (σ x) = x) {J : Matrix (Fin n) (Fin n) E} {l : E} (hl : l ≠ 0) (hJ : ∀ i j, σ ((l⁻¹ • J) j i) = -(l⁻¹ • J) i j) (h₂ : J₂ = dblGram finSumFinEquiv J) {T : GL (Fin (n + n)) E} (h : IsWittBasis σ J l T) (hn : 1 ≤ n) : Infinite (ratN σ J₂)

theorem **not_intConv_one** (hcm : IsCMExt F E σ) {J : Matrix (Fin n) (Fin n) E} {l : E} (hl : l ≠ 0) (hJ : ∀ i j, σ ((l⁻¹ • J) j i) = -(l⁻¹ • J) i j) (h₂ : J₂ = dblGram finSumFinEquiv J) {T : GL (Fin (n + n)) E} (h : IsWittBasis σ J l T) (hn : 1 ≤ n) {νN : Measure (unipN σ J₂)} (hν : IsNormN σ νN) (w g : UA σ J₂) : ¬ IntConv νN w (fun _ => 1) g

theorem **intM_rank_zero** {J₂ : Matrix (Fin (0 + 0)) (Fin (0 + 0)) E} {νN : Measure (unipN σ J₂)} (hν : IsNormN σ νN) (w g : UA σ J₂) (f : UA σ J₂ → ℂ) : intM νN w f g = f g

theorem **secSpace_bot_ne_bot** {JV : Matrix (Fin m) (Fin m) E} {κ : Type} [Fintype κ] (om : UA σ JV × UA σ J₂ →* Module.End ℂ ((κ → AF F) → ℂ)) : secSpace om ⊥ ≠ ⊥

theorem **exists_case** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E) (J₂ : Matrix (Fin (2 + 2)) (Fin (2 + 2)) E) (T : GL (Fin (2 + 2)) E) (w : UA σ J₂) (νN : Measure (unipN σ J₂)), ComplMet σ 1 1 False ∧ IsWeylElt σ T w ∧ IsNormN σ νN ∧ w ∉ siegelOf finSumFinEquiv (UG (cA σ) (incM F J₂)).subtype ∧ (∃ f, IntConv νN w f 1 ∧ intM νN w f 1 = 1) ∧ ¬ IntConv νN w (fun _ => 1) 1

Development.AbelianVarietyTheory.Definitions.Divisors.SecondChernDegreeFubiniStudyLine

theorem **quot_eq_map** {S : ShortComplex X.Modules} (h : Met p σ S.X₂ (0 + 1)) (hS : S.ShortExact) (h₁ : IsVB S.X₁ 0) (h₃ : IsVB S.X₃ 1) [IsIso S.g] : h.quot hS h₁ h₃ = h.map (asIso S.g)

theorem **isometric_grLB** {E : X.Modules} {h : VMetric p E 1} (H : h.IsCInf) (fl : Vert.Flag E 1) : (VMetric.det H).Isometric (VMetric.grLB fl 0 H)

theorem **isCh2Deg_rank_one** (HC : Published.ArCalc) [IsIntegral X] (hX : IsRegSurf p) {E : X.Modules} {h : VMetric p E 1} (H : h.IsCInf) {a : ℝ} (ha : (VMetric.det H).IsSelfInter a) : h.IsCh2Deg (a / 2)

theorem **isometric_det_ofLB** (L : HermLB p) : L.Isometric (VMetric.det (VMetric.isCInf_ofLB L))

theorem **isCh2Deg_scale** (HC : Published.ArCalc) [IsIntegral X] (hX : IsRegSurf p) {E : X.Modules} {h : VMetric p E 1} (H : h.IsCInf) {a : ℝ} (ha : (VMetric.det H).IsSelfInter a) (t : ℝ) : (h.scale t).IsCh2Deg (a / 2 - t / 2 * h.first_chern_integral)

theorem **secondChernDegree_fubiniStudy_line** (HC : Published.ArCalc) : haveI

theorem **eq_of_isFlagCh2** [IsIntegral X] (hX : IsRegSurf p) {E : X.Modules} {r : ℕ} {h : VMetric p E r} {H : h.IsCInf} {fl : Vert.Flag E r} {a b : ℝ} (ha : h.IsCh2Deg a) (hb : VMetric.IsFlagCh2 H fl b) : a = b

theorem **eq_zero_of_isCh2Deg_std** (HC : Published.ArCalc) {a : ℝ} : haveI

Development.AbelianVarietyTheory.Definitions.Divisors.SecondCountableTopologyOfIsProjOver

theorem **exists_finite** [CompactSpace (CxPts p σ)] : ∃ s : Finset R.ι, ∀ y, ∃ i ∈ s, y ∈ (R.chart i).e.source

theorem **holoOn_ev** (W : Y.Opens) (f : Γ(Y, W)) : R.HoloOn (ptsIn p σ W) (ev p σ W f)

theorem **exists_eventuallyEq** (c : AlgChart p σ D) {v : V D} (hv : v ∈ c.e.target) {E : Type*} (f : CxPts p σ → E) : ∃ i, c.e.symm v ∈ (R.chart i).e.source ∧ f ∘ c.e.symm =ᶠ[𝒩 v] (f ∘ (R.chart i).e.symm) ∘ (R.chart i).e ∘ c.e.symm

theorem **HoloOn.chart** {f : CxPts p σ → ℂ} (h : R.HoloOn S f) (hS : IsOpen S) (c : AlgChart p σ D) : DifferentiableOn ℂ (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)

theorem **SmoothOn.chart** {f : CxPts p σ → ℝ} (h : R.SmoothOn S f) (hS : IsOpen S) (c : AlgChart p σ D) : ContDiffOn ℝ ∞ (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)

theorem **holoOn_iff** (R R' : AnRead p σ D) (hS : IsOpen S) {f : CxPts p σ → ℂ} : R.HoloOn S f ↔ R'.HoloOn S f

theorem **smoothOn_iff** (R R' : AnRead p σ D) (hS : IsOpen S) {f : CxPts p σ → ℝ} : R.SmoothOn S f ↔ R'.SmoothOn S f

theorem **HoloOn.mono** {f : CxPts p σ → ℂ} (h : R.HoloOn S f) (e : T ⊆ S) : R.HoloOn T f

theorem **SmoothOn.mono** {f : CxPts p σ → ℝ} (h : R.SmoothOn S f) (e : T ⊆ S) : R.SmoothOn T f

theorem **HoloOn.congr** {f g : CxPts p σ → ℂ} (h : R.HoloOn S f) (e : ∀ y ∈ S, g y = f y) : R.HoloOn S g

theorem **SmoothOn.congr** {f g : CxPts p σ → ℝ} (h : R.SmoothOn S f) (e : ∀ y ∈ S, g y = f y) : R.SmoothOn S g

theorem **continuousOn_of_charts** {E : Type*} [TopologicalSpace E] {f : CxPts p σ → E} (hS : IsOpen S) (h : ∀ i, ContinuousOn (f ∘ (R.chart i).e.symm) ((R.chart i).e.target ∩ (R.chart i).e.symm ⁻¹' S)) : ContinuousOn f S

theorem **HoloOn.continuousOn** {f : CxPts p σ → ℂ} (h : R.HoloOn S f) (hS : IsOpen S) : ContinuousOn f S

theorem **SmoothOn.continuousOn** {f : CxPts p σ → ℝ} (h : R.SmoothOn S f) (hS : IsOpen S) : ContinuousOn f S

theorem **HoloOn.smoothOn_log** {f : CxPts p σ → ℂ} (h : R.HoloOn S f) (hS : IsOpen S) (h0 : ∀ y ∈ S, f y ≠ 0) : R.SmoothOn S fun y => Real.log (‖f y‖ ^ 2)

theorem **holoOn_of_chart** (c : AlgChart p σ D) {f : CxPts p σ → ℂ} (h : DifferentiableOn ℂ (f ∘ c.e.symm) (c.e.target ∩ c.e.symm ⁻¹' S)) : R.HoloOn (c.e.source ∩ S) f

theorem **holoOn_of_cover** {ι : Type*} {T : ι → Set (CxPts p σ)} (hT : ∀ j, IsOpen (T j)) (hc : ∀ y ∈ S, ∃ j, y ∈ T j) {f : CxPts p σ → ℂ} (h : ∀ j, R.HoloOn (S ∩ T j) f) : R.HoloOn S f

theorem **smoothOn_of_cover** {ι : Type*} {T : ι → Set (CxPts p σ)} (hT : ∀ j, IsOpen (T j)) (hc : ∀ y ∈ S, ∃ j, y ∈ T j) {f : CxPts p σ → ℝ} (h : ∀ j, R.SmoothOn (S ∩ T j) f) : R.SmoothOn S f

theorem **holoOn_ev_of_subset** {W : Y.Opens} (f : Γ(Y, W)) (h : S ⊆ ptsIn p σ W) : R.HoloOn S (ev p σ W f)

theorem **HoloOn.mul** {f g : CxPts p σ → ℂ} (hf : R.HoloOn S f) (hg : R.HoloOn S g) : R.HoloOn S fun y => f y * g y

theorem **HoloOn.zpow** {f : CxPts p σ → ℂ} (hf : R.HoloOn S f) (h0 : ∀ y ∈ S, f y ≠ 0) (n : ℤ) : R.HoloOn S fun y => f y ^ n

theorem **HoloOn.inv** {f : CxPts p σ → ℂ} (hf : R.HoloOn S f) (h0 : ∀ y ∈ S, f y ≠ 0) : R.HoloOn S fun y => (f y)⁻¹

theorem **holSec_eq** (R R' : AnRead p σ D) (B : LB Y) : R.holSec B = R'.holSec B

theorem **secVal_mem_holSec** (s : B.Sec) : secVal p σ s ∈ R.holSec B

@[simp] theorem **coe_valC** (x : SecC p σ B) : (R.valC B x : B.ι → CxPts p σ → ℂ) = SecC.val x

theorem **secTens_mem_holSec** {s : A.ι → CxPts p σ → ℂ} {t : B.ι → CxPts p σ → ℂ} (hs : s ∈ R.holSec A) (ht : t ∈ R.holSec B) : secTens s t ∈ R.holSec (tens A B)

theorem **IsSmooth.of_anRead** (h : φ.IsSmooth R) (R' : AnRead p σ D) : φ.IsSmooth R'

theorem **IsSmooth.addFn** (h : φ.IsSmooth R) {u : CxPts p σ → ℝ} (hu : R.SmoothOn univ u) : (φ.addFn u).IsSmooth R

theorem **IsSmooth.tens** {ψ : Weight p σ A} (h' : ψ.IsSmooth R) (h : φ.IsSmooth R) : (ψ.tens φ).IsSmooth R

theorem **IsSmooth.dual** (h : φ.IsSmooth R) : φ.dual.IsSmooth R

theorem **IsSmooth.zpow** (h : φ.IsSmooth R) (n : ℤ) : (φ.zpow n).IsSmooth R

theorem **isSmooth_ofPairs** {hA : A.IsLB} {hB : B.IsLB} {ψ : A.ι → B.ι → CxPts p σ → ℝ} {h} (hψ : ∀ a b, R.SmoothOn (ptsIn p σ (A.U a) ∩ ptsIn p σ (B.U b)) (ψ a b)) : (ofPairs hA hB ψ h).IsSmooth R

theorem **IsSmooth.cancel** (hA : A.IsLB) {χ : Weight p σ (CechPic.tens A B)} (h' : χ.IsSmooth R) (h : φ.IsSmooth R) : (cancel hA χ φ).IsSmooth R

theorem **IsSmooth.ofIso** (h : φ.IsSmooth R) (hA : A.IsLB) (w : LB.Iso B A) (hw : w.IsGen) : (φ.ofIso hA w hw).IsSmooth R

theorem **isSmooth_fs** {α : Type*} [Fintype α] {e : α → B.ι → CxPts p σ → ℂ} (hB : B.IsLB) (he : ∀ a, e a ∈ R.holSec B) (h : NoCommonZero e) : (fs hB e (fun a => (he a).1) h).IsSmooth R

theorem **isSmooth_fsBasis** {α : Type*} [Fintype α] [DecidableEq α] (hB : B.IsLB) (b : α → B.Sec) (hb : ⊔ j, nonzeroLocus (b j) = ⊤) (M : Matrix α α ℂ) (hM : M.det ≠ 0) : (fsBasis p σ hB b hb M hM).IsSmooth R

theorem **isSmooth_ofFun** {u : CxPts p σ → ℝ} (hu : R.SmoothOn univ u) : (ofFun u).IsSmooth R

theorem **IsSmooth.continuous_nsq** (h : φ.IsSmooth R) {s : B.ι → CxPts p σ → ℂ} (hs : s ∈ R.holSec B) : Continuous (φ.nsq s)

theorem **rv_mul_jacobian** {a : K.bundle.ι} (hy : y ∈ ptsIn p σ c.U) (ha : y ∈ ptsIn p σ (K.bundle.U a)) : rv σ c.isFrame (K.isFrame a) y * ev p σ _ (K.jacobian a inf_le_left fun i => resG Y (inf_le_right : _ ⊓ c.U ≤ _) (c.x i)) y = 1

theorem **rv_mul_jacobian_eq_det** {a : K.bundle.ι} {W : Y.Opens} (h : W ≤ K.bundle.U a) (g : Fin D → Γ(Y, W)) (hy : y ∈ c.e.source) (hW : y ∈ ptsIn p σ W) : rv σ c.isFrame (K.isFrame a) y * ev p σ W (K.jacobian a h g) y = (c.jacMat g y).det

theorem **topDens_eq** (hf : f ∈ fibSec p σ (tens K.bundle B)) (hy : y ∈ c.e.source) {a : K.bundle.ι} {b : B.ι} (ha : y ∈ ptsIn p σ (K.bundle.U a)) (hb : y ∈ ptsIn p σ (B.U b)) : c.topDens K φ f y = 2 ^ D * (‖c.topCoeff K f a b y‖ ^ 2 * Real.exp (-φ.φ b y))

theorem **topDens_trans** (hy : y ∈ c.e.source) (hy' : y ∈ c'.e.source) : c.topDens K φ f y = ‖transDet c.e c'.e y‖ ^ 2 * c'.topDens K φ f y

theorem **chartMeas_topDens** {A : Set (CxPts p σ)} (hA : MeasurableSet A) (h : A ⊆ c.e.source) (h' : A ⊆ c'.e.source) : chartMeas c.e (fun y => ENNReal.ofReal (c.topDens K φ f y)) A = chartMeas c'.e (fun y => ENNReal.ofReal (c'.topDens K φ f y)) A

theorem **topMeasure_apply** (c : AlgChart p σ D) {A : Set (CxPts p σ)} (hA : MeasurableSet A) (h : A ⊆ c.e.source) : R.topMeasure K φ f A = ∫⁻ v in c.e '' A, ENNReal.ofReal (c.topDens K φ f (c.e.symm v))

theorem **topMeasure_eq** (R' : AnRead p σ D) : R.topMeasure K φ f = R'.topMeasure K φ f

theorem **topInt_eq** (R' : AnRead p σ D) : R.topInt K φ f = R'.topInt K φ f

theorem **isLocallyFiniteMeasure_topMeasure** (hφ : φ.IsSmooth R) (hf : f ∈ R.holSec (tens K.bundle B)) : IsLocallyFiniteMeasure (R.topMeasure K φ f)

theorem **topInt_lt_top** [CompactSpace (CxPts p σ)] (hφ : φ.IsSmooth R) (hf : f ∈ R.holSec (tens K.bundle B)) : R.topInt K φ f < ⊤

theorem **mem_U** (h : y ∈ (F.refine.chart c).e.source) : y ∈ ptsIn p σ (R.chart c.1).U

theorem **mem_K** (h : y ∈ (F.refine.chart c).e.source) : y ∈ ptsIn p σ (F.K.bundle.U c.2.1)

theorem **mem_L** (h : y ∈ (F.refine.chart c).e.source) (a : κ) : y ∈ ptsIn p σ ((F.L a).U (c.2.2 a))

theorem **atlas_chart** (c : F.Idx) : F.atlas.chart c = (F.refine.chart c).e

theorem **atlas_kap** : F.atlas.kap = Pi.single none 1

theorem **atlas_g** (m : Option κ → ℤ) (c c' : F.Idx) (y : CxPts p σ) : F.atlas.g m c c' y = ∏ o, F.g₀ o c c' y ^ m o

theorem **atlas_ph** (m : Option κ → ℤ) (c : F.Idx) (y : CxPts p σ) : F.atlas.ph m c y = ∑ o, m o * F.ph₀ o c y

theorem **atlas_g_single** (o : Option κ) (c c' : F.Idx) (y : CxPts p σ) : F.atlas.g (Pi.single o 1) c c' y = F.g₀ o c c' y

theorem **atlas_ph_single** (o : Option κ) (c : F.Idx) (y : CxPts p σ) : F.atlas.ph (Pi.single o 1) c y = F.ph₀ o c y

theorem **atlas_g_wt** (a : κ) (c c' : F.Idx) (y : CxPts p σ) : F.atlas.g (wt a) c c' y = gv p σ (F.L a) (c.2.2 a) (c'.2.2 a) y

theorem **atlas_ph_wt** (a : κ) (c : F.Idx) (y : CxPts p σ) : F.atlas.ph (wt a) c y = (F.φ a).φ (c.2.2 a) y

theorem **pos_wt** {a : κ} (h : (F.φ a).SemiPos R) : F.atlas.Pos (wt a) fun _ => 0

theorem **holoOn_iff_atlas** {S : Set (CxPts p σ)} (hS : IsOpen S) {f : CxPts p σ → ℂ} : R.HoloOn S f ↔ ∀ c, DifferentiableOn ℂ (f ∘ (F.atlas.chart c).symm) ((F.atlas.chart c).target ∩ (F.atlas.chart c).symm ⁻¹' S)

theorem **g_self** (m : Option κ → ℤ) (h : y ∈ (F.atlas.chart c).source) : F.atlas.g m c c y = 1

theorem **read_eq** (hs : s ∈ fibSec p σ B) (h : y ∈ (F.atlas.chart c).source) {i : B.ι} (hi : y ∈ ptsIn p σ (B.U i)) : r.read s c y = r.u c i y * s i y

theorem **isSec_read** (hs : s ∈ R.holSec B) : F.atlas.IsSec m (r.read s)

theorem **isSec_read_secVal** (s : B.Sec) : F.atlas.IsSec m (r.read (secVal p σ s))

theorem **isSec_read_val** (x : SecC p σ B) : F.atlas.IsSec m (r.read (SecC.val x))

theorem **eq_of_read_eq** (hs : s ∈ fibSec p σ B) (ht : t ∈ fibSec p σ B) (h : ∀ c, ∀ y ∈ (F.atlas.chart c).source, r.read s c y = r.read t c y) : s = t

theorem **exists_read_eq** {G : F.Idx → CxPts p σ → ℂ} (hG : F.atlas.IsSec m G) : ∃ s ∈ R.holSec B, ∀ c, ∀ y ∈ (F.atlas.chart c).source, r.read s c y = G c y

theorem **Isometric.nsq_read** {φ : Weight p σ B} (h : r.Isometric φ) (s : B.ι → CxPts p σ → ℂ) (y : CxPts p σ) : F.atlas.nsq m (r.read s) y = φ.nsq s y

theorem **read_ofMem** {a : κ} {s : (F.L a).ι → CxPts p σ → ℂ} (hs : s ∈ fibSec p σ (F.L a)) (h : y ∈ (F.atlas.chart c).source) : (ofMem F a).read s c y = s (c.2.2 a) y

theorem **isometric_ofMem** (a : κ) : (ofMem F a).Isometric (F.φ a)

theorem **isometric_canon** : (canon F F.K).Isometric F.φK

theorem **isometric_triv** : (triv F).Isometric (Weight.ofFun fun _ => 0)

@[simp] theorem **read_cast** (h : m = m') : (r.cast h).read = r.read

theorem **Isometric.cast** {φ : Weight p σ B} (h : r.Isometric φ) (e : m = m') : (r.cast e).Isometric φ

theorem **Isometric.tens** {r' : F.Reading A m'} {ψ : Weight p σ A} {φ : Weight p σ B} (h' : r'.Isometric ψ) (h : r.Isometric φ) : (r'.tens r).Isometric (ψ.tens φ)

theorem **read_tens** (r' : F.Reading A m') {s' : A.ι → CxPts p σ → ℂ} (hs' : s' ∈ fibSec p σ A) (hs : s ∈ fibSec p σ B) (h : y ∈ (F.atlas.chart c).source) : (r'.tens r).read (secTens s' s) c y = r'.read s' c y * r.read s c y

theorem **Isometric.zpow** {φ : Weight p σ B} (h : r.Isometric φ) (n : ℤ) : (r.zpow n).Isometric (φ.zpow n)

theorem **u_mul_inv_iso** {w : LB.Iso B A} (hw : w.IsGen) (h : y ∈ (F.atlas.chart c).source) (h' : y ∈ (F.atlas.chart c').source) {a a' : A.ι} {b b' : B.ι} (ha : y ∈ ptsIn p σ (A.U a)) (ha' : y ∈ ptsIn p σ (A.U a')) (hb : y ∈ ptsIn p σ (B.U b)) (hb' : y ∈ ptsIn p σ (B.U b')) : F.atlas.g m c c' y * (r.u c' b' y * (ev p σ _ (w.s (a', b')).val y)⁻¹) = r.u c b y * (ev p σ _ (w.s (a, b)).val y)⁻¹ * gv p σ A a a' y

theorem **Isometric.ofIso** {φ : Weight p σ B} (h : r.Isometric φ) (hA : A.IsLB) (w : LB.Iso B A) (hw : w.IsGen) : (r.ofIso hA w hw).Isometric (φ.ofIso hA w hw)

theorem **Isometric.topDens_eq** {φ : Weight p σ B} (h : r.Isometric φ) (K : CanonLB p D) {f : (CechPic.tens K.bundle B).ι → CxPts p σ → ℂ} (hf : f ∈ R.holSec (CechPic.tens K.bundle B)) (hy : y ∈ (F.atlas.chart c).source) : (F.refine.chart c).topDens K φ f y = 2 ^ D * (F.atlas.nsq (F.atlas.kap + m) (((canon F K).tens r).read f) y * Real.exp (F.atlas.ph F.atlas.kap c y))

theorem **Isometric.topMeasure_eq** {φ : Weight p σ B} (h : r.Isometric φ) (K : CanonLB p D) {f : (CechPic.tens K.bundle B).ι → CxPts p σ → ℂ} (hf : f ∈ R.holSec (CechPic.tens K.bundle B)) : R.topMeasure K φ f = F.atlas.mu.withDensity fun y => ENNReal.ofReal (2 ^ D * F.atlas.nsq (F.atlas.kap + m) (((canon F K).tens r).read f) y)

theorem **Isometric.topInt_eq** {φ : Weight p σ B} (h : r.Isometric φ) (K : CanonLB p D) {f : (CechPic.tens K.bundle B).ι → CxPts p σ → ℂ} (hf : f ∈ R.holSec (CechPic.tens K.bundle B)) : R.topInt K φ f = 2 ^ D * ∫⁻ y, ENNReal.ofReal (F.atlas.nsq (F.atlas.kap + m) (((canon F K).tens r).read f) y) ∂F.atlas.mu

theorem **t2Space_cxPts_of_isProjOver** (h : IsProjOver p) : T2Space (CxPts p σ)

theorem **secondCountableTopology_of_isProjOver** (h : IsProjOver p) : SecondCountableTopology (CxPts p σ)

theorem **fubiniStudyWeight_projPoint** (v : Unimodular ℂ N) (i : Fin (N + 1)) (hi : v.1 i ≠ 0) : (fsO1 k σ N).φ i (projPt σ v) = Real.log (∑ a, ‖v.1 a / v.1 i‖ ^ 2)

theorem **noCommonZero_jac** : Weight.NoCommonZero fun _ : Unit => secVal (PSpace.str k N) σ (K.jacSec (coord k N))

Development.AbelianVarietyTheory.Definitions.Divisors.SemiposAtOfPositive

theorem **NefAt.positive** {D : D7} {Y : ShimX D.what₂.Tau} {A : SlopeTestRemainingChoices Y} (h : NefAt A) (hs : SemiposAt A) : Positive A

theorem **semiposAt_of_famS** {D : D7} {Y : ShimX D.what₂.Tau} {T : Real0 Y} {A : SlopeTestRemainingChoices Y} (hF : ∀ c, (T.F c).FamS) (hM : TieM T A) : SemiposAt A

Development.AbelianVarietyTheory.Definitions.Divisors.SharpOverIff

theorem **sharpOver_iff** (F : SFam g n ζ) (W : Matrix (Fin 8) (Fin 8) ℤ) {T : Scheme.{0}} (P : PolAV T) (u : T ⟶ F.X) : SharpOver F W P u ↔ ∃ c pr, SharpBy F W P u c pr

theorem **isCpl_self** (O : Type) [CommRing O] [IsLocalRing O] : IsCpl (algebraMap O (AdicCompletion (maximalIdeal O) O))

Development.AbelianVarietyTheory.Definitions.Divisors.SimpleTwistedTraceFormulaStable

theorem **baseChangeSph_of_bcCore** : BCCore → BaseChangeSph

theorem **oddInert_iff** (E : Type) [Field E] [NumberField E] [IsCMField E] : OddInert E ↔ ∀ w : Pl E, (2 : 𝒪 E) ∈ w.asIdeal → IsSplitAt w

theorem **bcCoreOdd_of_bcCore** : BCCore → BCCoreOdd

Development.AbelianVarietyTheory.Definitions.Divisors.SlopeTestRemainingChoicesAlt

theorem **SqIn.mono** {σ : K →+* ℂ} {σ' : K' →+* ℂ} (h : SqIn G μ₀ σ) (hσ : Set.range σ ⊆ Set.range σ') : SqIn G μ₀ σ'

theorem **SlopeTestRemainingChoices.atScale** {p : PtAt ex Csz} {CF : ℝ} {G : Grid p CF} {W : G.Weil} {n : ℕ} (a : SlopeTestRemainingChoices p G W n) : a.ch.AtScale CF p.Z.b

Development.AbelianVarietyTheory.Definitions.Divisors.SmallFam

theorem **shiftA_H** (P : Pair p) (u : ℝ) (a b : ℤ) : (shiftA P u).H a b = P.H a b

Development.AbelianVarietyTheory.Definitions.Divisors.SmoothOfRelativeDimensionPspace

theorem **isStandardSmooth_mvPolynomial** (R σ : Type) [CommRing R] [Finite σ] : Algebra.IsStandardSmoothOfRelativeDimension (Nat.card σ) R (MvPolynomial σ R)

theorem **smoothOfRelativeDimension_pspace** (R : Type) [CommRing R] (M : ℕ) : SmoothOfRelativeDimension M (PSpace.str R M)

theorem **geometricallyConnected_pspace** : GeometricallyConnected (PSpace.str k M)

theorem **algebraΓ_eq** : (projSys k M).algebraΓ = algebraSections k M ⊤

theorem **isVariety** (hM : 1 ≤ M) : (projSys k M).IsVariety

theorem **basePointFree** : (projSys k M).BasePointFree

theorem **form_eq** {d : ℕ} {g : MvPolynomial (Fin (M + 1)) k} (hg : g.IsHomogeneous d) : (projSys k M).form hg = ProjSpace.formSec d g hg

theorem **imageIdeal_eq_bot** : (projSys k M).imageIdeal = ⊥

theorem **del_eq_one** : (projSys k M).del = 1

theorem **eq_of_sameImage** {y y' : PSpace k M} (h : (projSys k M).SameImage y y') : y = y'

theorem **noContract** (bd : Set (PSpace k M)) : (projSys k M).NoContract bd

theorem **gen_mem** (i : Fin (M + 1)) : gen k M ∈ chart k M i

theorem **indep** : (projSys k M).Indep

theorem **valR_resPt** {Y : Scheme.{0}} (y : Y) (U : Y.Opens) (hy : y ∈ U) (f : Γ(Y, U)) : valR (resPt y) U f = Y.evaluation U y hy f

theorem **birational** : (projSys k M).Birational

Development.AbelianVarietyTheory.Definitions.Divisors.SmoothSpanLeMspan

theorem **colU_zero** (r : ℕ) : colU (E

theorem **colU_add** (r : ℕ) (x y : Fin a → Cinf E) : colU r (x + y) = colU r x ∗ colU r y

theorem **colU_of_le** {r : ℕ} (hr : a ≤ r) (x : Fin a → Cinf E) : colU r x = 1

theorem **smoothSpan_le_mspan** (π : CuspSub E a) : smoothSpan π ≤ mspan π

theorem **mspan_ne_bot** (π : CuspSub E a) : mspan π ≠ ⊥

theorem **mspan_le_leftInv** (π : CuspSub E a) : mspan π ≤ leftInv E a

theorem **mspan_left** (π : CuspSub E a) {F : GA E a → ℂ} (hF : F ∈ mspan π) (γ : GL (Fin a) E) (g : GA E a) : F (ratGL E a γ ∗ g) = F g

theorem **colConv_integrable** (π : CuspSub E a) {φ : Sm E a} (hφ : φ ∈ π.V) (g : GA E a) (r : ℕ) (f : SchwartzMap (Fin a → Cinf E) ℂ) : Integrable (fun x => f x ∗ φ.1 (g ∗ infGL E a (colU r x)))

Development.AbelianVarietyTheory.Definitions.Divisors.SpecializesGenericImage

theorem **specializes_genericImage** (T : UTrait Xo c) (ξ : Sp T.V ⟶ M.Z) (v : Sp (FractionRing T.V)) : (jK T.V ≫ ξ) v ⤳ (jK T.V ≫ ξ) (IsLocalRing.closedPoint (FractionRing T.V))

theorem **UOrdGe.mem_O** (h : UOrdGe M T ξ x a) : x ∈ M.O ((jK T.V ≫ ξ) (IsLocalRing.closedPoint (FractionRing T.V)))

theorem **UOrdEq.val_ne_zero** (h : UOrdEq M T ξ x a) : M.val (jK T.V ≫ ξ) x ≠ 0

theorem **UOrdEq.mem_O** (h : UOrdEq M T ξ x a) : x ∈ M.O ((jK T.V ≫ ξ) (IsLocalRing.closedPoint (FractionRing T.V)))

theorem **UOrdEq.ordGe** (h : UOrdEq M T ξ x a) : UOrdGe M T ξ x a

theorem **UOrdGe.mono** (h : UOrdGe M T ξ x a) (hba : b ≤ a) : UOrdGe M T ξ x b

theorem **UOrdGe.le_of_ordEq** (h : UOrdGe M T ξ x b) (h' : UOrdEq M T ξ x a) : b ≤ a

theorem **UOrdEq.unique** (h : UOrdEq M T ξ x a) (h' : UOrdEq M T ξ x b) : a = b

theorem **uOrdEq_one** (T : UTrait Xo c) (ξ : Sp T.V ⟶ M.Z) : UOrdEq M T ξ 1 0

theorem **UOrdGe.add** (hx : UOrdGe M T ξ x a) (hy : UOrdGe M T ξ y a) : UOrdGe M T ξ (x + y) a

theorem **UOrdGe.mul** (hx : UOrdGe M T ξ x a) (hy : UOrdGe M T ξ y b) : UOrdGe M T ξ (x ∗ y) (a + b)

theorem **UOrdEq.mul** (hx : UOrdEq M T ξ x a) (hy : UOrdEq M T ξ y b) : UOrdEq M T ξ (x ∗ y) (a + b)

theorem **UOrdGe.ordEq_zero_of_inv** (h0 : x ≠ 0) (hx : UOrdGe M T ξ x 0) (hi : UOrdGe M T ξ x⁻¹ 0) : UOrdEq M T ξ x 0

theorem **ULocTuples.exists_nat** {X X' : Scheme.{0}} {U₀ : M.Z.Opens} {u₀ : U₀.toScheme ⟶ X} {a : X' ⟶ X} {cc : X' ⟶ Xo.XF c} {φ : InvFun (𝒪 E)} {ι : ℕ} (h : ULocTuples M U₀ u₀ a cc φ ι) {T : UTrait Xo c} {ξ : Sp T.V ⟶ M.Z} (hT : UThroughU M U₀ u₀ a cc T ξ) : ∃ a : ℕ, (a : ℝ) = ι ∗ φ.evalZ T.be

Development.AbelianVarietyTheory.Definitions.Divisors.StarMulOfMulStar

theorem **star_mul_conj** {γ₀ γ g : B×} (n : B) (hn : ∀ x : B, n ∗ x = x ∗ n) (h0 : (γ₀ : B) ∗ star (γ₀ : B) = n) (h1 : (γ : B) ∗ star (γ : B) = n) (hg : (g : B) ∗ γ₀ = γ ∗ g) : star (g : B) ∗ γ = γ₀ ∗ star (g : B)

theorem **star_mul_self_comm** {γ₀ γ g : B×} (n : B) (hn : ∀ x : B, n ∗ x = x ∗ n) (h0 : (γ₀ : B) ∗ star (γ₀ : B) = n) (h1 : (γ : B) ∗ star (γ : B) = n) (hg : (g : B) ∗ γ₀ = γ ∗ g) : ((star g ∗ g : B×) : B) ∗ γ₀ = γ₀ ∗ ((star g ∗ g : B×) : B)

theorem **inv_comm** {γ₀ : B} (u : B×) (hu : (u : B) ∗ γ₀ = γ₀ ∗ u) : ((u⁻¹ : B×) : B) ∗ γ₀ = γ₀ ∗ ((u⁻¹ : B×) : B)

@[simp] theorem **ι_liftUnit** (ι : L →ₐ[k] B) (hι : Function.Injective ι) (γ₀ : B) (hcent : ∀ x : B, x ∗ γ₀ = γ₀ ∗ x ↔ x ∈ Set.range ι) (u : B×) (hu : (u : B) ∗ γ₀ = γ₀ ∗ u) : ι ((liftUnit ι hι γ₀ hcent u hu : L×) : L) = u

@[simp] theorem **ι_cocyc** (ι : L →ₐ[k] B) (hι : Function.Injective ι) (γ₀ : B) (hcent : ∀ x : B, x ∗ γ₀ = γ₀ ∗ x ↔ x ∈ Set.range ι) (c : L →ₐ[k] L) (hc : ∀ a, ι (c a) = star (ι a)) (g : B×) (hg : ((star g ∗ g : B×) : B) ∗ γ₀ = γ₀ ∗ ((star g ∗ g : B×) : B)) : ι (((cocyc ι hι γ₀ hcent c hc g hg : fixU k c) : L×) : L) = star (g : B) ∗ g

theorem **mk_eq_mk_of** {c : L →ₐ[k] L} (a b : fixU k c) (t : k×) (y : L×) (h : ((b : L×) : L) = algebraMap k L t ∗ ((a : L×) : L) ∗ ((y : L) ∗ c y)) : (QuotientGroup.mk a : H1 k c) = QuotientGroup.mk b

theorem **StConj.comm** {γ₀ γ : B×} (ν : k×) (h0 : (γ₀ : B) ∗ star (γ₀ : B) = algebraMap k B ν) (h1 : (γ : B) ∗ star (γ : B) = algebraMap k B ν) {g : B×} (hg : (g : B) ∗ γ₀ = γ ∗ g) : ((star g ∗ g : B×) : B) ∗ γ₀ = γ₀ ∗ ((star g ∗ g : B×) : B)

theorem **mk_cocyc_eq** {γ : B×} (g g' : B×) (hg : (g : B) ∗ γ₀ = γ ∗ g) (hg' : (g' : B) ∗ γ₀ = γ ∗ g') (hgc : ((star g ∗ g : B×) : B) ∗ γ₀ = γ₀ ∗ ((star g ∗ g : B×) : B)) (hgc' : ((star g' ∗ g' : B×) : B) ∗ γ₀ = γ₀ ∗ ((star g' ∗ g' : B×) : B)) : (QuotientGroup.mk (cocyc ι hι (γ₀ : B) hcent c hc g hgc) : H1 k c) = QuotientGroup.mk (cocyc ι hι (γ₀ : B) hcent c hc g' hgc')

theorem **invCls_eq** {γ : B×} (h : StConj k γ₀ γ) (g : B×) (hg : (g : B) ∗ γ₀ = γ ∗ g) (hgc : ((star g ∗ g : B×) : B) ∗ γ₀ = γ₀ ∗ ((star g ∗ g : B×) : B)) : invCls ι hι γ₀ hcent c hc h = QuotientGroup.mk (cocyc ι hι (γ₀ : B) hcent c hc g hgc)

theorem **StConj.refl'** (ν : k×) (h0 : (γ₀ : B) ∗ star (γ₀ : B) = algebraMap k B ν) : StConj k γ₀ γ₀

theorem **invCls_self** (h : StConj k γ₀ γ₀) : invCls ι hι γ₀ hcent c hc h = 1

theorem **mul_star_of_star_mul** (x : B×) (n : B) (hn : ∀ y : B, n ∗ y = y ∗ n) (h : star (x : B) ∗ x = n) : (x : B) ∗ star (x : B) = n

@[simp] theorem **coe_locElt** (γ₀ : B×) : ((locElt S γ₀ : (S ⊗[ℚ] B)×) : S ⊗[ℚ] B) = (1 : S) ⊗ₜ[ℚ] (γ₀ : B)

theorem **locMap_cent** (ι : L →ₐ[ℚ] B) (hι : Function.Injective ι) (γ₀ : Bˣ) (hcent : ∀ x : B, x * γ₀ = γ₀ * x ↔ x ∈ Set.range ι) : Function.Injective (locMap S ι) ∧ ∀ y : S ⊗[ℚ] B, y * (locElt S γ₀ : S ⊗[ℚ] B) = (locElt S γ₀ : S ⊗[ℚ] B) * y ↔ y ∈ Set.range (locMap S ι)

theorem **locMap_star** (ι : L →ₐ[ℚ] B) (c : L →ₐ[ℚ] L) (hc : ∀ a, ι (c a) = star (ι a)) [StarRing (S ⊗[ℚ] B)] (hstar : ∀ (s : S) (b : B), star (s ⊗ₜ[ℚ] b) = s ⊗ₜ[ℚ] star b) (a : S ⊗[ℚ] L) : locMap S ι (locInv S c a) = star (locMap S ι a)

Development.AbelianVarietyTheory.Definitions.Divisors.StepsNonvacuous

theorem **jet_const** (J : JetRing ℂ I n) (a : ℂ) : jet J (fun _ ↦ a) = algebraMap ℂ J.B a

theorem **jet_congr** (J : JetRing ℂ I n) {f g : (I → ℂ) → ℂ} (h : f =ᶠ[𝓝 0] g) : jet J f = jet J g

theorem **jet_linear** (J : JetRing ℂ I n) (l : (I → ℂ) →L[ℂ] ℂ) : jet J l = ∑ i, l (Pi.single i 1) • J.t i

theorem **jet_coord** (J : JetRing ℂ I n) (i : I) : jet J (fun z ↦ z i) = J.t i

theorem **t_ne_zero** (J : JetRing ℂ I n) (i : I) : J.t i ≠ 0

theorem **jet_aeval_X** (J : JetRing ℂ I n) (i : I) : MvPolynomial.aeval J.t (MvPolynomial.X i : MvPolynomial I ℂ) = jet J fun z ↦ MvPolynomial.aeval z (MvPolynomial.X i : MvPolynomial I ℂ)

theorem **coord_nonempty** (h1 : JetRingEx) (h2 : JetLift) (h3 : JetBC) (h4 : ModelAlg) (h5 : GermEx) (h6 : CoordEx) (K : Type) [Field K] [NumberField K] (d : ℕ) (δ : K) (io : K →+* ℂ) (hK : Module.finrank ℚ K = 2) (hd : 0 < d) (hδ : δ ^ 2 = -(d : K)) (Λ₀ : PeriodPair) (Ω₀ : ℂ) (hΛ : IsLatOf io 1 Λ₀) (hΩ : Ω₀ ^ 12 = latDisc Λ₀) (E : Type) [Field E] [NumberField E] [IsCMField E] [Algebra K E] (I : Type) [Fintype I] (φ : I → E →+* ℂ) (hφ : IsCMType φ) (hio : ∀ i, (φ i).comp (algebraMap K E) = io) (c : Ideal (OF E)) (y : (Af E)ˣ) (hc : c ≠ ⊥) (n : ℕ) : ∃ (𝔠 : FracId (FR E)) (T : CxPt 𝔠 φ (lattice c y)) (J : JetRing ℂ I n) (Y : Family J T.X T.δ T.low) (G : Germ Y) (ε : ℤˣ), Nonempty (Coord ε G) ∧ ∀ i, T.A i ≠ 0

theorem **steps_nonvacuous** (h : Steps) (h1 : JetRingEx) (h2 : JetLift) (h3 : JetBC) (h4 : ModelAlg) (K : Type) [Field K] [NumberField K] (d : ℕ) (δ : K) (io : K →+* ℂ) (hK : Module.finrank ℚ K = 2) (hd : 0 < d) (hδ : δ ^ 2 = -(d : K)) (Λ₀ : PeriodPair) (Ω₀ : ℂ) (hΛ : IsLatOf io 1 Λ₀) (hΩ : Ω₀ ^ 12 = latDisc Λ₀) (E : Type) [Field E] [NumberField E] [IsCMField E] [Algebra K E] (I : Type) [Fintype I] (φ : I → E →+* ℂ) (hφ : IsCMType φ) (hio : ∀ i, (φ i).comp (algebraMap K E) = io) (c : Ideal (OF E)) (y : (Af E)ˣ) (hc : c ≠ ⊥) (n : ℕ) : ∃ (𝔠 : FracId (FR E)) (T : CxPt 𝔠 φ (lattice c y)) (J : JetRing ℂ I n) (Y : Family J T.X T.δ T.low) (G : Germ Y) (ε : ℤˣ) (C : Coord ε G), (∀ i, jet J (fun z ↦ C.ζ z i) ≠ 0 ∧ ¬ (fun z ↦ C.ζ z i) =ᶠ[𝓝 0] fun _ ↦ 0) ∧ G.e * jet J C.C = algebraMap ℂ J.B T.C

Development.AbelianVarietyTheory.Definitions.Divisors.SubRedMem

theorem **Charts.Hom.ten_tmul** {T T' : K.Obj} (f : T ⟶ T') (d : D) (c : T.C) : Charts.Hom.ten D f (d ⊗ₜ c) = d ⊗ₜ f.φ c

theorem **Charts.Obj.red_tmul** (T : K.Obj) (d : D) (c : T.C) : T.red π (d ⊗ₜ c) = algebraMap ℂ T.C (π d) * c

theorem **_root_.GlobalTheory.AnPer.Charts.Hom.red_ten** {T T' : K.Obj} (f : T ⟶ T') (x : T.ten D) : T'.red π (_root_.GlobalTheory.AnPer.Charts.Hom.ten D f x) = f.φ (T.red π x)

theorem **_root_.GlobalTheory.AnPer.Charts.Hom.ten_mem** {T T' : K.Obj} (f : T ⟶ T') {x : T.ten D} (h : x ∈ T.tenJ π) : _root_.GlobalTheory.AnPer.Charts.Hom.ten D f x ∈ T'.tenJ π

private theorem **sub_one_tmul_red_mem_map_ker** (T : K.Obj) (x : T.ten D) : x - (1 : D) ⊗ₜ T.red π x ∈ (RingHom.ker π).map (Algebra.TensorProduct.includeLeft : D →ₐ[ℂ] D ⊗[ℂ] T.C)

theorem **_root_.GlobalTheory.AnPer.Charts.Obj.tenJ_nil** (hπ : IsNilpotent (RingHom.ker π)) (T : K.Obj) : IsNilpotent (T.tenJ π)

Development.AbelianVarietyTheory.Definitions.Divisors.SubsetSpanBy

theorem **subset_spanBy** {R₁ F₁ : Type} [CommRing R₁] [CommRing F₁] (c : R₁ →+* F₁) (T : Set F₁) : T ⊆ spanBy c T

Development.AbelianVarietyTheory.Definitions.Divisors.Subsingleton

@[simp] theorem **htAct_act** {s : S ⟶ Sp R} (o : Obj b N s) (c : D.O) : (htAct o).act c = o.act.act c

theorem **HTIw.exists_pull** {s' : S' ⟶ Sp R} {s : S ⟶ Sp R} {o' : Obj b N s'} {o : Obj b N s} {w : Ideal (𝒪 D.F)} (C : HTIw o w) {f : S' ⟶ S} (m : ObjMap o' o f) : ∃ C' : HTIw o' w, HTIwMap C' C f

theorem **HTIwMap.subsingleton** {s' : S' ⟶ Sp R} {s : S ⟶ Sp R} {o' : Obj b N s'} {o : Obj b N s} {w : Ideal (𝒪 D.F)} (C' : HTIw o' w) (C : HTIw o w) (f : S' ⟶ S) (m : ObjMap o' o f) : Subsingleton (ChainMap C' C f m.φ)

Development.AbelianVarietyTheory.Definitions.Divisors.SubsingletonComplex

theorem **conjC_apply** (z : ℂ) : conjC z = (starRingEnd ℂ) z

theorem **conjCC_apply** (z : ℂ × ℂ) : conjCC z = ((starRingEnd ℂ) z.1, (starRingEnd ℂ) z.2)

theorem **conjCC_invol** (z : ℂ × ℂ) : conjCC (conjCC z) = z

theorem **mem_normSub_iff** {k : Type} [Field k] {A : Type} [CommRing A] [Algebra k A] (c : A →ₐ[k] A) (a : fixU k c) : a ∈ normSub k c ↔ ∃ (t : kˣ) (x : Aˣ), ((a : Aˣ) : A) = algebraMap k A t * (x * c x)

theorem **subsingleton_complex** : Subsingleton (H1 ℝ conjC)

theorem **u_mem** : u ∈ fixU ℝ conjCC

theorem **u_not_mem** : ⟨u, u_mem⟩ : fixU ℝ conjCC) ∉ normSub ℝ conjCC

theorem **nontrivial_prod** : Nontrivial (H1 ℝ conjCC)

theorem **simT_ne_top** : simT ℝ conjCC ≠ ⊤

Development.AbelianVarietyTheory.Definitions.Divisors.SumExpSt

theorem **LIncl.succ** {K : Type} [Field K] [NumberField K] [StarRing K] {G : Type} [Group G] [Fintype G] {Pl0 : Type} {B : BlockDat K G Pl0} {ℓ : Lvl K G} {Z : Submodule K B.S.Forms} {w : HeightOneSpectrum (𝒪 K)} {J : ℕ} (h : LIncl B ℓ Z w J) : LIncl B ℓ Z w (J + 1)

Development.AbelianVarietyTheory.Definitions.Divisors.SurjectiveOfIsPullbackSqZero

theorem **map_sq** (hI : I ^ 2 = 0) (B : Type) [CommRing B] [Algebra R B] : I.map (algebraMap R B) ^ 2 = 0

theorem **surjective_spMap_sqZero** {B B₀ : Type} [CommRing B] [CommRing B₀] (p : B →+* B₀) (hp : Function.Surjective p) {J : Ideal B} (hJ : J ^ 2 = 0) (hk : ∀ b, p b = 0 → b ∈ J) : Surjective (spMap p)

theorem **surjective_of_isPullback_sqZero** (hI : I ^ 2 = 0) {π : R →+* R₀} (hπ : Function.Surjective π) (hk : RingHom.ker π = I) {s₀ : X₀ ⟶ Sp R₀} {X : Scheme.{0}} {s : X ⟶ Sp R} {i : X₀ ⟶ X} (h : IsPullback i s₀ s (spMap π)) : Surjective i

theorem **cover_of_surjective** {X : Scheme.{0}} {i : X₀ ⟶ X} (hi : Surjective i) (y : ∀ U : AffOp X₀, Sp (S.F.obj U) ⟶ X) (red : ∀ U, spMap (S.p U) ≫ y U = affPt s₀ U.unop ≫ i) (x : X) : ∃ U, x ∈ Set.range (y U).base

theorem **surjective_spMap_p** (U : AffOp X₀) : Surjective (spMap (S.p U))

theorem **Presents.surjective** {X : Scheme.{0}} {s : X ⟶ Sp R} {i : X₀ ⟶ X} (P : S.Presents s i) : Surjective i

theorem **isTheta_zero** (I : Ideal R) {B B₀ B' : Type} [CommRing B] [Algebra R B] [CommRing B₀] [Algebra R₀ B₀] [CommRing B'] [Algebra R B'] (p : B →+* B₀) (α : B →ₐ[R] B') : IsTheta (R₀

theorem **redMor** (hI : I ^ 2 = 0) (hk : RingHom.ker π = I) (s₀ : X₀ ⟶ Sp R₀) : IsMor (divided₂ hI) (s₀ ≫ spMap π) (dp0 R₀) s₀ π (𝟙 X₀) where

theorem **smooth_sec_affine** [Smooth s] (U : X.affineOpens) : Algebra.Smooth R (Sec s U.1)

Development.AbelianVarietyTheory.Definitions.Divisors.SymAtOfThList

theorem **hasTh_exists** (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Sp K)) (ρ : LB.Gen (P.L.pull (eSec P.A))) (hρ : ρ.IsGen) (m : ℕ) (hm : Odd m) (hK : (m : K) ≠ 0) : ∃ c : Fin ((m ^ 2) ^ (2 * P.g)) → K, HasTh P.A P.L ρ m (𝟙 (Sp K)) c ∧ (3 ≤ m → c ≠ 0)

theorem **length_eq** {S : Scheme.{0}} (P : PolAV S) (ρ : LB.Gen (P.L.pull (eSec P.A))) (m : ℕ) {K : Type} [Field K] (hK : (m : K) ≠ 0) (x : Sp K ⟶ S) {θ : SecAt P.A P.L x} {N : ℕ} {pt : Fin N → (Over.mk x ⟶ P.A)} {c : Fin N → K} (H : ThList P.A P.L ρ m x θ pt c) : N = (m ^ 2) ^ (2 * P.g)

theorem **polFib_self** {R : Type} [CommRing R] (P : PolAV (Sp R)) : PolFib P.A P.L P.g (𝟙 _)

theorem **symTup_two** {R : Type} [CommRing R] (a b : R) : symTup ![a, b] = ![(a + b) ^ 2, a * b]

theorem **symTup_perm** {R : Type} [CommRing R] {N : ℕ} (c : Fin N → R) (σ : Equiv.Perm (Fin N)) : symTup (c ∘ σ) = symTup c

theorem **symTup_smul** {R : Type} [CommRing R] {N : ℕ} (c : Fin N → R) (t : R) (j : Fin N) : symTup (fun a => t * c a) j = t ^ N.factorial * symTup c j

theorem **symTup_eq_zero** {K : Type} [Field K] {N : ℕ} (c : Fin N → K) (h : symTup c = 0) : c = 0

theorem **prim_one_two** : Prim (Int.castRingHom ℚ) ![(1 : ℚ), 2]

theorem **not_prim_two** : ¬ Prim (Int.castRingHom ℚ) ![(2 : ℚ)]

theorem **symTup_map** {R R' : Type} [CommRing R] [CommRing R'] (i : R →+* R') {N : ℕ} (c : Fin N → R) (j : Fin N) : symTup (fun a => i (c a)) j = i (symTup c j)

theorem **thG_pull** {S : Scheme.{0}} (P : PolAV S) {ρ : LB.Gen (P.L.pull (eSec P.A))} (hρ : ρ.IsGen) {m : ℕ} {B T : Over S} (β : B ⟶ T) {θ : LB.Sec (LPow P.A P.L ((1 : ℕ) : ℤ) T)} {y : T ⟶ P.A} {c : Γ(T.left, ⊤)} (H : ThG P.A P.L ρ m θ y c) : ThG P.A P.L ρ m (VVRank.pullS P.A _ β θ) (β ≫ y) (β.left.appTop.hom c)

theorem **thMap_exists** (K : Type) [Field K] (P : PolAV (Sp K)) (ρ : LB.Gen (P.L.pull (eSec P.A))) (hρ : ρ.IsGen) (m : ℕ) (hm : Odd m) (hK : (m : K) ≠ 0) {n : ℕ} (c : Fin (n + 1) → K) (hc : HasTh P.A P.L ρ m (𝟙 (Sp K)) c) (h0 : c ≠ 0) : ∃ f : Sp K ⟶ PSpace K n, f ≫ PSpace.str K n = 𝟙 _ ∧ PCrd f (symTup c) ∧ ThMap P.A P.L ρ m (𝟙 _) f

theorem **elliptic** : ∃ (P : PolAV (Sp ℂ)) (ρ : LB.Gen (P.L.pull (eSec P.A))) (c : Fin 81 → ℂ) (f : Sp ℂ ⟶ PSpace ℂ 80), P.g = 1 ∧ ρ.IsGen ∧ PolFib P.A P.L 1 (𝟙 _) ∧ HasTh P.A P.L ρ 3 (𝟙 _) c ∧ c ≠ 0 ∧ PCrd f (symTup c) ∧ ThMap P.A P.L ρ 3 (𝟙 _) f

theorem **thMap_of_ne** {K : Type} [Field K] (P : PolAV (Sp K)) (ρ : LB.Gen (P.L.pull (eSec P.A))) (m : ℕ) (hK : (m : K) ≠ 0) {n : ℕ} (hn : n + 1 ≠ (m ^ 2) ^ (2 * P.g)) (f : Sp K ⟶ PSpace K n) : ThMap P.A P.L ρ m (𝟙 _) f

theorem **thList_pull** {S : Scheme.{0}} (P : PolAV S) {ρ : LB.Gen (P.L.pull (eSec P.A))} (hρ : ρ.IsGen) {m : ℕ} {K F : Type} [Field K] [Field F] (i : K →+* F) {x : Sp K ⟶ S} {θ : SecAt P.A P.L x} {N : ℕ} {pt : Fin N → (Over.mk x ⟶ P.A)} {c : Fin N → K} (H : ThList P.A P.L ρ m x θ pt c) (a : Fin N) : ThG P.A P.L ρ m (SecAt.pull P.A P.L θ i) ((Over.homMk (spMap i) rfl : Over.mk (spMap i ≫ x) ⟶ Over.mk x) ≫ pt a) (GlobalTheory.ZH.fn (i (c a)))

theorem **symAt_of_thList** {K : Type} [Field K] (P : PolAV (Sp K)) {ρ : LB.Gen (P.L.pull (eSec P.A))} (hρ : ρ.IsGen) {m : ℕ} (hm : Odd m) (hK : (m : K) ≠ 0) {θ : SecAt P.A P.L (𝟙 (Sp K))} {N : ℕ} {pt : Fin N → (Over.mk (𝟙 (Sp K)) ⟶ P.A)} {c : Fin N → K} (H : ThList P.A P.L ρ m _ θ pt c) : SymAt P.A P.L ρ m _ θ (symTup c)

theorem **hasTh₂_of_hasTh** {S : Scheme.{0}} (P : PolAV S) {ρ : LB.Gen (P.L.pull (eSec P.A))} {m m' : ℕ} {K : Type} [Field K] {x : Sp K ⟶ S} {N N' : ℕ} {c : Fin N → K} {c' : Fin N' → K} (H : HasTh P.A P.L ρ m x c) (H' : HasTh P.A P.L ρ m' x c') : ∃ r : K, HasTh₂ P.A P.L ρ m m' x c fun a => r * c' a

Development.AbelianVarietyTheory.Definitions.Divisors.SymTupOne

theorem **hasTh_iff** {m : ℕ} {K : Type} [Field K] {x : Sp K ⟶ S} {N : ℕ} {c : Fin N → K} : HasTh G L ρ m x c ↔ HasTh₂ G L ρ m m x c c

theorem **symTup_one** {R : Type} [CommRing R] (c : Fin 1 → R) : symTup c 0 = c 0

Development.AbelianVarietyTheory.Definitions.Divisors.T2SpaceGeneralLinearGroup

theorem **mem_KO_iff** (g : GL n k) : g ∈ KO k n ↔ (∀ i j, g.val i j ∈ 𝒪[k]) ∧ ∀ i j, (g⁻¹).val i j ∈ 𝒪[k]

theorem **isEmbedding_val_GL** : IsEmbedding (fun g : GL n k => g.val)

theorem **isOpen_intMatrix** {m : Type*} [Fintype m] : IsOpen {A : Matrix m n k | ∀ i j, A i j ∈ 𝒪[k]}

theorem **isCompact_intMatrix** {m : Type*} : IsCompact {A : Matrix m n k | ∀ i j, A i j ∈ 𝒪[k]}

theorem **isOpen_KO** : IsOpen (KO k n : Set (GL n k))

theorem **isCompact_KO** : IsCompact (KO k n : Set (GL n k))

@[simp] theorem **diagGL_val** (c : n → kˣ) : (diagGL c).val = diagonal fun i => (c i : k)

theorem **det_diagGL** (c : n → kˣ) : Matrix.GeneralLinearGroup.det (diagGL c) = ∏ i, c i

theorem **hom_eq_one_of_center** (Δ : GL n k →* ℝ≥0) (hc : ∀ g, (∀ h, g * h = h * g) → Δ g = 1) (g : GL n k) : Δ g = 1

theorem **haar_ne_zero_ne_top** (μ : Measure (GL n k)) [μ.IsHaarMeasure] {J : Subgroup (GL n k)} (ho : IsOpen (J : Set (GL n k))) (hK : J ≤ KO k n) : μ J ≠ 0 ∧ μ J ≠ ⊤

theorem **isHaarMeasure_haarVol** {J : Subgroup (GL n k)} (ho : IsOpen (J : Set (GL n k))) (hK : J ≤ KO k n) : (haarVol J).IsHaarMeasure

theorem **haarVol_self** {J : Subgroup (GL n k)} (ho : IsOpen (J : Set (GL n k))) (hK : J ≤ KO k n) : haarVol J J = 1

theorem **eq_haarVol** (μ : Measure (GL n k)) [μ.IsHaarMeasure] {J : Subgroup (GL n k)} (ho : IsOpen (J : Set (GL n k))) (hK : J ≤ KO k n) (h1 : μ J = 1) : μ = haarVol J

Development.AbelianVarietyTheory.Definitions.Divisors.TateProductRigidificationTrivial

theorem **Factor.isFactor** {D D' : DegPre R} (F : Factor D D') : IsFactor D D'

theorem **Wit.isMonHom_pullbackComp** {S S' S'' : Scheme.{0}} (s : S'' ⟶ S') (p : S' ⟶ S) (G : Over S) [GrpObj G] : IsMonHom (M

theorem **Wit.pullbackComp_fst** {S S' S'' : Scheme.{0}} (s : S'' ⟶ S') (p : S' ⟶ S) (G : Over S) : ((Over.pullbackComp s p).hom.app G).left ≫ pullback.fst _ s ≫ pullback.fst G.hom p = pullback.fst G.hom (s ≫ p)

theorem **Wit.toL_surj** (R : Type) [CommRing R] : Function.Surjective (Wit.toL R)

theorem **psi2** (n : ℤ × ℤ) : IsConst (tate2.psi n) (pi0 ^ (n.1 ^ 2 + n.2 ^ 2))

theorem **tateProduct_rigidification_trivial** : IsConst tate2.rigK 1

theorem **tau2** (a b : ℤ × ℤ) : IsConst (tate2.tau a b) (pi0 ^ (2 * a.1 * b.1 + 2 * a.2 * b.2))

theorem **tateProduct_period_value** (a b : ℤ × ℤ) (z J) (hJ : Good tate2.P.L z J) : val (tate2.tau a b) (𝟙 _) z J = gm (𝟙 _) z ⊤ (pi0 ^ (2 * a.1 * b.1 + 2 * a.2 * b.2))

theorem **Wit.fac_e** : Wit.fac.e ≠ 0 ∧ Wit.fac.e ≠ 1

theorem **SamePol.refl** {R : Type} [CommRing R] [IsDomain R] (D : DegPre R) (h : D.TauBimult) : SamePol D D

theorem **Emb.theta_triv** {R : Type} [CommRing R] [IsLocalRing R] {S : Scheme.{0}} {π : S ⟶ Sp R} {G : Over S} [GrpObj G] (E : Emb π G) : E.Theta (Inh.trivLB G.left)

Development.AbelianVarietyTheory.Definitions.Divisors.TestFnConstTermVanishX

theorem **IsPC'.isPCG** {r : ℕ} {ε : Fin r → Bool} {lam : Fin r → ℤ} {μ : Measure (UJ ε)} {X : Type} [NormedAddCommGroup X] [InnerProductSpace ℂ X] {D : Matrix (Fin r) (Fin r) ℂ → (X →L[ℂ] X)} {hD : IsDS ε (fun a => (lam a : ℝ)) D} {f : UJ ε → ℂ} (h : IsPC' ε lam μ D hD f) : haveI

theorem **IsTestArchGX.mem** {fa : ∀ w, GArch D.sp.cm.sig V.gram w → ℂ} (h : IsTestArchGX M μa fa) (w : InfinitePlace D.host.F) : fa w ∈ CcArch D.sp.cm.sig V.gram w

theorem **IsTestFnGX.isPureTensor** {T : TypeDataG V} {U : AuxiliaryPlaceHermitianSpace V} {f : GA D V → ℂ} (h : IsTestFnGX M T U μa μf f) : IsPureTensor D.sp.cm.sig V.gram (frameLat V) f ∧ f ∈ CcGA V

theorem **NonFieldTypeVanishingX.zero** (h : NonFieldTypeVanishingX) (hr : Odd V.rank) {S : D.what₂.Tau → SPar} {M : DSModel V S} {T : TypeDataG V} {U : AuxiliaryPlaceHermitianSpace V} (hH : HypH (k

Development.AbelianVarietyTheory.Definitions.Divisors.TestValOfHhTest

theorem **testVal_of_hhTest** : ∀ r : ℝ, ∀ (D : D7) (Y : DV.ShimX D.what₂.Tau) (Φ : ℝ → Fn Y) (hh : Fn Y), 2 < D.R.f * (Y.N − 1) → LimFam Φ → Propor Y → HhTest Y hh → LatAt Y → PrimeLocalCondition Y → PrimeLocalSplitCondition Y → SiegRead Y → DRAt Y r → TestVal Φ hh r

Development.AbelianVarietyTheory.Definitions.Divisors.ThetaSetupStructureMapComp

theorem **isThetaCompact** : IsThetaCompact F X₁.z i

theorem **div** : X₁.M.HasDiv X₁.S.th

theorem **posW** : PosW W

theorem **diag_pos** (a : Fin 8) : 0 < W a a

theorem **z_eq** {z : Z ⟶ Sp (cycAway n)} (h : i ≫ z = F.s) : X₁.z = z

theorem **isProper_compactificationMap** : IsProper X₁.z

theorem **s_comp** : X₁.M.s ≫ spMap X₁.κ = X₁.π ≫ X₁.z

theorem **projM** : IsProjOver X₁.M.s

theorem **isUnit_level** : IsUnit (n : X₁.o₁)

theorem **point_eq** {c c' : CTrait X₁ p z₁} (h : c.y = c'.y) : spMap c.ι ≫ c.T.y = spMap c'.ι ≫ c'.T.y

theorem **val_eq_valInt** (c : CTrait X₁ p z₁) (a : X₁.M.Z.presheaf.stalk z₁) : c.val a = (c.valInt a : PadicAlgCl p)

theorem **norm_val_le** (c : CTrait X₁ p z₁) (a : X₁.M.Z.presheaf.stalk z₁) : ‖c.val a‖ ≤ 1

theorem **norm_val_lt_iff** (c : CTrait X₁ p z₁) (a : X₁.M.Z.presheaf.stalk z₁) : ‖c.val a‖ < 1 ↔ a ∈ maximalIdeal (X₁.M.Z.presheaf.stalk z₁)

theorem **norm_uniformiser_pos** (c : CTrait X₁ p z₁) : 0 < ‖c.ι (algebraMap c.T.V (FractionRing c.T.V) c.T.ϖ)‖

theorem **scale_pos** (c : CTrait X₁ p z₁) : 0 < c.sc

theorem **exp_neg_sc_mul** (c : CTrait X₁ p z₁) (a : ℕ) : Real.exp (−(c.sc * a)) = ‖c.ι (algebraMap c.T.V (FractionRing c.T.V) c.T.ϖ)‖ ^ a

Development.AbelianVarietyTheory.Definitions.Divisors.ToBaseChangePtsBijective

theorem **map_smul_of_algebraMap** {A A' : Type} [CommRing A] [Algebra R A] [CommRing A'] [Algebra R' A'] (f : A →+* A') (h : ∀ r, f (algebraMap R A r) = algebraMap R' A' (σ r)) (r : R) (a : A) : f (r • a) = σ r • f a

@[simp] theorem **lin_apply** (a : A) : f.lin a = f a

theorem **map_algebraMap** (r : R) : f (algebraMap R A r) = algebraMap R' A' (σ r)

theorem **two_comul** (a : A) : f.two (comul a) = comul (f a)

@[simp] theorem **ofBialgHom_apply** {B : Type} [CommRing B] [Bialgebra R B] (g : A →ₐc[R] B) (a : A) : ofBialgHom g a = g a

@[simp] theorem **toBialgHom_apply** {B : Type} [CommRing B] [Bialgebra R B] (g : SemiBialgHom (RingHom.id R) A B) (a : A) : g.toBialgHom a = g a

theorem **two_two** (g : SemiBialgHom τ A' A'') {ρ : R →+* R''} (k : A →ₛₗ[ρ] A'') (h : ∀ a, k a = g (f a)) (c : A ⊗[R] A) : g.two (f.two c) = TensorProduct.map k k c

@[simp] theorem **comp_apply** (g : SemiBialgHom τ A' A'') (a : A) : g.comp f a = g (f a)

@[simp] theorem **pts_apply** (x : WithConv (A' →ₐ[R'] B)) (a : A) : (f.pts B hB x).ofConv a = x.ofConv (f a)

@[simp] theorem **cot_d** (a : A) : f.cot (dUnit R a) = dUnit R' (f a)

theorem **two_cobOne** (b : A) : f.two (cobOne R A b) = cobOne R' A' (f b)

theorem **cobTwo_two** (c : A ⊗[R] A) : cobTwo R' A' (f.two c) = TensorProduct.map f.lin f.two (cobTwo R A c)

theorem **two_mem** {c : A ⊗[R] A} (hc : c ∈ symCocycles R A) : f.two c ∈ symCocycles R' A'

@[simp] theorem **toBaseChange_apply** (a : A) : toBaseChange R R' A a = (1 : R') ⊗ₜ[R] a

theorem **toBaseChange_pts_bijective** (B : Type) [CommRing B] [Algebra R B] [Algebra R' B] (hB : ∀ r, algebraMap R B r = algebraMap R' B (algebraMap R R' r)) : Function.Bijective ((toBaseChange R R' A).pts B hB)

@[simp] theorem **bcMap_tmul** {C : Type} [CommRing C] [Bialgebra R C] (g : A →ₐc[R] C) (r : R') (a : A) : bcMap R' g (r ⊗ₜ a) = r ⊗ₜ g a

theorem **bc_hom_ext** {g h : R' ⊗[R] A →ₐc[R'] A'} (e : ∀ a, g (1 ⊗ₜ a) = h (1 ⊗ₜ a)) : g = h

@[simp] theorem **SemiBialgHom.liftBC_tmul** (f : SemiBialgHom (algebraMap R R') A A') (r : R') (a : A) : f.liftBC (r ⊗ₜ a) = r • f a

theorem **ext_apply** {s t : SemiHom σ G' G} (h : ∀ n a, s.app n a = t.app n a) : s = t

@[simp] theorem **ofHom_app** (f : G ⟶ H) (n : ℕ) (a : H.A n) : (ofHom f).app n a = f.app n a

@[simp] theorem **toHom_app** (s : SemiHom (RingHom.id O) G H) (n : ℕ) (a : H.A n) : s.toHom.app n a = s.app n a

@[simp] theorem **comp_app** (t : SemiHom τ G'' G') (s : SemiHom σ G' G) (n : ℕ) (a : G.A n) : (t.comp s).app n a = t.app n (s.app n a)

@[simp] theorem **funs_val** (b : G.funs) (n : ℕ) : (s.funs b).1 n = s.app n (b.1 n)

theorem **res_two** (n : ℕ) (c : G.A (n + 1) ⊗[O] G.A (n + 1)) : pullbackTwo (G'.res n) ((s.app (n + 1)).two c) = (s.app n).two (pullbackTwo (G.res n) c)

@[simp] theorem **cocycles_val** (c : G.cocycles) (n : ℕ) : (s.cocycles c).1 n = (s.app n).two (c.1 n)

theorem **res_cot** (n : ℕ) (w : Cot O (G.A (n + 1))) : cotMap (G'.res n) ((s.app (n + 1)).cot w) = (s.app n).cot (cotMap (G.res n) w)

@[simp] theorem **omega_val** (w : G.omega) (n : ℕ) : (s.omega w).1 n = (s.app n).cot (w.1 n)

theorem **cocycles_cob** (b : G.funs) : s.cocycles (G.cob b) = G'.cob (s.funs b)

theorem **omega_dFun** (b : G.funs) : s.omega (G.dFun b) = G'.dFun (s.funs b)

@[simp] theorem **rig_mk** (c : G.cocycles) (w : G.omega) : s.rig (G.rigMk (c, w)) = G'.rigMk (s.cocycles c, s.omega w)

theorem **rig_rigOfOmega** (w : G.omega) : s.rig (G.rigOfOmega w) = G'.rigOfOmega (s.omega w)

theorem **rigToExt_rig** (x : G.RigExt) : G'.rigToExt (s.rig x) = s.vecExt (G.rigToExt x)

theorem **cocycles_cocycles** (c : G.cocycles) : t.cocycles (s.cocycles c) = u.cocycles c

theorem **omega_omega** (w : G.omega) : t.omega (s.omega w) = u.omega w

theorem **rig_rig** (x : G.RigExt) : t.rig (s.rig x) = u.rig x

theorem **comp_rig** (t : SemiHom τ G'' G') (x : G.RigExt) : (t.comp s).rig x = t.rig (s.rig x)

@[simp] theorem **ofHom_cocycles** (f : G ⟶ H) : (ofHom f).cocycles = cocyclesMap f

@[simp] theorem **ofHom_omega** (f : G ⟶ H) : (ofHom f).omega = omegaMap G H f

@[simp] theorem **ofHom_rig** (f : G ⟶ H) : (ofHom f).rig = rigExtMap f

theorem **rig_naturality** (s' : SemiHom σ H' H) (f : G ⟶ H) (f' : G' ⟶ H') (h : ∀ n a, s.app n (f.app n a) = f'.app n (s'.app n a)) (x : H.RigExt) : s.rig (rigExtMap f x) = rigExtMap f' (s'.rig x)

theorem **omega_naturality** (s' : SemiHom σ H' H) (f : G ⟶ H) (f' : G' ⟶ H') (h : ∀ n a, s.app n (f.app n a) = f'.app n (s'.app n a)) (w : H.omega) : s.omega (omegaMap G H f w) = omegaMap G' H' f' (s'.omega w)

@[simp] theorem **toBase_app** (n : ℕ) (a : G.A n) : (G.toBase O').app n a = (1 : O') ⊗ₜ[O] a

@[simp] theorem **baseChangeMap_app** (f : G ⟶ H) (n : ℕ) : (baseChangeMap O' f).app n = bcMap O' (f.app n)

theorem **baseChange_hom_ext** {G' : PDivGroup p O'} {f g : G' ⟶ G.baseChange O'} (h : ∀ n a, f.app n (1 ⊗ₜ a) = g.app n (1 ⊗ₜ a)) : f = g

theorem **baseChangeMap_add** (f g : G ⟶ H) : baseChangeMap O' (f + g) = baseChangeMap O' f + baseChangeMap O' g

@[simp] theorem **SemiHom.lift_app** {G' : PDivGroup p O'} (s : SemiHom (algebraMap O O') G' G) (n : ℕ) (r : O') (a : G.A n) : s.lift.app n (r ⊗ₜ a) = r • s.app n a

Development.AbelianVarietyTheory.Definitions.Divisors.ToBidualDualMapEigenImageInclusion

@[reassoc (attr

theorem **mono_toBidual_of_mono** (i : G ⟶ F) [Mono i] [Mono (cartierDualToBidual F)] : Mono (cartierDualToBidual G)

theorem **mono_toBidual_of_isVectorBundle** {r : ℕ} (h : IsVB F r) : Mono (cartierDualToBidual F)

theorem **isRegularFn_of_isUnit** {D : Γ(X, ⊤)} (h : IsUnit D) : IsRegularFn D

theorem **isRegularFn_of_ne_zero** [IsIntegral X] {D : Γ(X, ⊤)} (h : D ≠ 0) : IsRegularFn D

theorem **IsRegularFn.eq_zero_of_smul** {D : Γ(X, ⊤)} (h : IsRegularFn D) [Mono (cartierDualToBidual F)] {V : X.Opens} {s : Γ(F, V)} (e : resG X (le_top : V ≤ ⊤) D • s = 0) : s = 0

theorem **IsRegularFn.hom_eq_zero** {D : Γ(X, ⊤)} (h : IsRegularFn D) [Mono (cartierDualToBidual F)] {φ : G ⟶ F} (e : D • φ = 0) : φ = 0

theorem **isTorsionFree_of_mono_toBidual** [IsIntegral X] [Mono (cartierDualToBidual F)] : IsTorsionFree F

theorem **dualApp_eigenEval** {V W : X.Opens} (s : Γ(F, V)) (h : W ≤ V) (l : Γ(eigen (dualAct ρ) σ, W)) : dualApp ((eigenEval ρ σ).app V s) h l = dualApp ((eigenι (dualAct ρ) σ).app W l) le_rfl (resM F h s)

theorem **eigenEval_eq** : eigenEval ρ σ = eigenπ ρ σ ≫ cartierDualToBidual _ ≫ dualMap (dualQuotIso ρ σ).inv

@[reassoc (attr

theorem **isTorsionFree_eigenIm** [IsIntegral X] : IsTorsionFree (eigenIm ρ σ)

theorem **eigenImπ_app_eq_zero_iff** {V : X.Opens} {s : Γ(F, V)} : (eigenImπ ρ σ).app V s = 0 ↔ ∀ (W : X.Opens) (h : W ≤ V) (l : Γ(dual F, W)), l ∈ (eigenSub (dualAct ρ) σ).sec W → dualApp l le_rfl (resM F h s) = 0

theorem **act_eigenImπ** (a : O) : act ρ a ≫ eigenImπ ρ σ = σ a • eigenImπ ρ σ

theorem **isEquivariant_eigenImπ** : IsEquivariant ρ (scalarAct (eigenIm ρ σ) σ) (eigenImπ ρ σ)

@[reassoc (attr

@[reassoc] theorem **quotToIm_ι** : quotToIm ρ σ ≫ eigenImι ρ σ = cartierDualToBidual _ ≫ dualMap (dualQuotIso ρ σ).inv

theorem **comp_eq_zero_of_comp_eigenImπ** [Mono (cartierDualToBidual G)] {φ : F ⟶ G} (h : ∀ a, act ρ a ≫ φ = σ a • φ) {k : H ⟶ F} (hk : k ≫ eigenImπ ρ σ = 0) : k ≫ φ = 0

@[reassoc (attr

@[reassoc (attr

@[reassoc] theorem **quotToIm_eigenImMap** {φ : F ⟶ G} (h : IsEquivariant ρ ρ' φ) : quotToIm ρ σ ≫ eigenImMap σ h = eigenQuotMap σ h ≫ quotToIm ρ' σ

@[reassoc] theorem **eigenImMap_ι** {φ : F ⟶ G} (h : IsEquivariant ρ ρ' φ) : eigenImMap σ h ≫ eigenImι ρ' σ = eigenImι ρ σ ≫ dualMap (eigenMap σ h.dual)

@[reassoc (attr

theorem **toBidual_dualMap_eigenImage_inclusion** : (cartierDualToBidual _ ≫ dualMap (eigenImι ρ σ)) ≫ dualMap (eigenImπ ρ σ) = eigenι (dualAct ρ) σ

@[reassoc (attr

theorem **eigenImι_eq** : eigenImι ρ σ = cartierDualToBidual _ ≫ dualMap (dualImIso ρ σ).inv

theorem **isIso_eigenImι** {r : ℕ} (h : IsVB (eigenIm ρ σ) r) : IsIso (eigenImι ρ σ)

theorem **isVB_eigen_dualAct** {r : ℕ} (h : IsVB (eigenIm ρ σ) r) : IsVB (eigen (dualAct ρ) σ) r

theorem **isIso_eigenImι_of_epi** {r : ℕ} (h : IsVB F r) [Epi (dualMap (eigenι (dualAct ρ) σ))] : IsIso (eigenImι ρ σ)

theorem **isIso_quotToIm_iff** : IsIso (quotToIm ρ σ) ↔ Mono (cartierDualToBidual (eigenQuot ρ σ))

theorem **isIso_quotToIm** {r : ℕ} (h : IsVB (eigenQuot ρ σ) r) : IsIso (quotToIm ρ σ)

theorem **isIso_quotToIm_of_isIso** [Mono (cartierDualToBidual F)] [IsIso (eigenι ρ σ ≫ eigenπ ρ σ)] : IsIso (quotToIm ρ σ)

theorem **isIso_quotToIm_of_split** {k κ : Type} [Field k] [Algebra k Γ(X, ⊤)] [Fintype κ] {τ : κ → (O →+* k)} (hτ : IsSplitBy τ) [Mono (cartierDualToBidual F)] (j : κ) : IsIso (quotToIm ρ (scal X (τ j)))

theorem **isIso_eigenι_eigenImπ_of_split** {k κ : Type} [Field k] [Algebra k Γ(X, ⊤)] [Fintype κ] {τ : κ → (O →+* k)} (hτ : IsSplitBy τ) [Mono (cartierDualToBidual F)] (j : κ) : IsIso (eigenι ρ (scal X (τ j)) ≫ eigenImπ ρ (scal X (τ j)))

theorem **isIso_eigenπ_of_actsThrough** (h : ActsThrough ρ σ) : IsIso (eigenπ ρ σ)

theorem **isIso_eigenImπ_of_actsThrough** [Mono (cartierDualToBidual F)] (h : ActsThrough ρ σ) : IsIso (eigenImπ ρ σ)

theorem **isZero_eigenIm_of_actsThrough** {τ : O →+* Γ(X, ⊤)} (h : ActsThrough ρ τ) {a : O} (ha : IsRegularFn (τ a - σ a)) : IsZero (eigenIm ρ σ)

theorem **isEigenProj_of_tensor** {R : Type} [CommRing R] (β : R →+* Γ(X, ⊤)) (χ : O →+* R) (t : R ⊗[ℤ] O →+* End F) (ht : ∀ c a, act t (c ⊗ₜ a) = β c • act ρ a) (v : R ⊗[ℤ] O →+ R) (hv : ∀ c a, v (c ⊗ₜ a) = c * χ a) {x : R ⊗[ℤ] O} (hx : ∀ a, (1 ⊗ₜ a) * x = χ a • x) : IsEigenProj ρ (β.comp χ) (act t x) (β (v x)) where

theorem **exists_isEigenProj_of_tensor** (ρ : O →+* End F) {R : Type} [CommRing R] (β : R →+* Γ(X, ⊤)) (χ : O →+* R) (v : R ⊗[ℤ] O →+ R) (hv : ∀ c a, v (c ⊗ₜ a) = c * χ a) {x : R ⊗[ℤ] O} (hx : ∀ a, (1 ⊗ₜ a) * x = χ a • x) : ∃ ε, IsEigenProj ρ (β.comp χ) ε (β (v x))

theorem **isEigenProj_idem** {k κ : Type} [Field k] [Algebra k Γ(X, ⊤)] [DecidableEq κ] {τ : κ → (O →+* k)} (hτ : IsSplitBy τ) (j : κ) : IsEigenProj ρ (scal X (τ j)) (act (tensorAct k ρ) (hτ.idem j)) 1

theorem **comp_eq_smul** {φ : F ⟶ G} (hφ : ∀ a, act ρ a ≫ φ = σ a • φ) : ε ≫ φ = D • φ

theorem **comp_eq_smul'** {φ : G ⟶ F} (hφ : ∀ a, φ ≫ act ρ a = σ a • φ) : φ ≫ ε = D • φ

theorem **isEquivariant** : IsEquivariant ρ ρ ε

theorem **comp_act** (a : O) : ε ≫ act ρ a = σ a • ε

theorem **comp_self** : ε ≫ ε = D • ε

theorem **dual** : IsEigenProj (dualAct ρ) σ (dualMap ε) D where

theorem **pullback** (f : Y ⟶ X) : IsEigenProj (pullAct ρ f) (f.appTop.hom.comp σ) ((Scheme.Modules.pullback f).map ε) (f.appTop D) where

theorem **of_iso** {ρ' : O →+* End G} {φ : F ≅ G} (hφ : IsEquivariant ρ ρ' φ.hom) : IsEigenProj ρ' σ (φ.inv ≫ ε ≫ φ.hom) D where

theorem **comp_eigenImπ_eq_zero** (hD : IsRegularFn D) {k : G ⟶ F} (hk : k ≫ ε = 0) : k ≫ eigenImπ ρ σ = 0

@[reassoc (attr

theorem **imToSub_ι_eigenImπ** : h.imToSub ≫ eigenι ρ σ ≫ eigenImπ ρ σ = D • 𝟙 _

theorem **ι_eigenImπ_imToSub** : eigenι ρ σ ≫ eigenImπ ρ σ ≫ h.imToSub = D • 𝟙 _

theorem **quotToIm_imToSub_ι_eigenπ** : quotToIm ρ σ ≫ h.imToSub ≫ eigenι ρ σ ≫ eigenπ ρ σ = D • 𝟙 _

theorem **comp_eigenImπ_eq_zero_iff** {k : G ⟶ F} : k ≫ eigenImπ ρ σ = 0 ↔ k ≫ ε = 0

theorem **eigenImπ_app_eq_zero_iff** {V : X.Opens} {s : Γ(F, V)} : (eigenImπ ρ σ).app V s = 0 ↔ ε.app V s = 0

theorem **quotToIm_app_eq_zero_iff** {V : X.Opens} {s : Γ(eigenQuot ρ σ, V)} : (quotToIm ρ σ).app V s = 0 ↔ resG X (le_top : V ≤ ⊤) D • s = 0

theorem **mono_imToSub** : Mono h.imToSub

theorem **mono_ι_eigenImπ** : Mono (eigenι ρ σ ≫ eigenImπ ρ σ)

@[reassoc (attr

theorem **isQuasicoherent_eigenIm** [F.IsQuasicoherent] : (eigenIm ρ σ).IsQuasicoherent

theorem **isFiniteQC_eigenIm** [IsFiniteQC F] : IsFiniteQC (eigenIm ρ σ)

@[reassoc (attr

theorem **pullbackImHom_comp** : h.pullbackImHom f hf ≫ (h.pullback f).imToSub ≫ eigenι _ _ ≫ (Scheme.Modules.pullback f).map (eigenImπ ρ σ) = (f.appTop D : Γ(Y, ⊤)) • 𝟙 _

theorem **pullbackImHom_app_eq_zero_iff** {V : Y.Opens} {s : Γ((Scheme.Modules.pullback f).obj (eigenIm ρ σ), V)} : (h.pullbackImHom f hf).app V s = 0 ↔ resG Y (le_top : V ≤ ⊤) (f.appTop D) • s = 0

theorem **isIso_pullbackImHom** {r : ℕ} (hr : IsVB (eigenIm ρ σ) r) : IsIso (h.pullbackImHom f hf)

theorem **act_twoAct_X** : act (twoAct F G) Polynomial.X = 2 • (biprod.snd ≫ biprod.inr)

theorem **act_comp_of_X** {ρ : ℤ[X] →+* End F} {φ : F ⟶ G} (h : act ρ Polynomial.X ≫ φ = 0) (a : ℤ[X]) : act ρ a ≫ φ = zeroChar X a • φ

theorem **isEigenProj_twoAct** : IsEigenProj (twoAct F G) (zeroChar X) (2 • (biprod.fst ≫ biprod.inl)) 2 where

theorem **actsThrough_pull_twoAct** (f : Y ⟶ X) (hY : (2 : Γ(Y, ⊤)) = 0) : ActsThrough (pullAct (twoAct F G) f) (f.appTop.hom.comp (zeroChar X))

theorem **not_isIso_quotToIm_twoAct** {r s : ℕ} (hF : IsVB F r) (hG : IsVB G (s + 1)) (h : IsRegularFn (2 : Γ(X, ⊤))) (f : Y ⟶ X) [Nonempty Y] (hY : (2 : Γ(Y, ⊤)) = 0) : ¬IsIso (quotToIm (twoAct F G) (zeroChar X))

Development.AbelianVarietyTheory.Definitions.Divisors.ToBidualNaturalityDualSheafTranspose

theorem **isSheaf_homPresheaf** : (homPresheaf F G).IsSheaf

theorem **homApp_resM** (φ : Γ(homMod F G, V)) (h : W ≤ V) {T : X.Opens} (h' : T ≤ W) (s : Γ(F, W)) : homApp φ (h'.trans h) (resM F h' s) = resM G h' (homApp φ h s)

@[simp] theorem **homApp_homMk** (φ : ∀ W : X.Opens, W ≤ V → (Γ(F, W) →ₗ[Γ(X, W)] Γ(G, W))) (hφ) (h : W ≤ V) : homApp (homMk φ hφ) h = φ W h

@[ext] theorem **hom_ext** {φ ψ : Γ(homMod F G, V)} (e : ∀ (W : X.Opens) (h : W ≤ V) (s : Γ(F, W)), homApp φ h s = homApp ψ h s) : φ = ψ

@[simp] theorem **homApp_zero** (h : W ≤ V) : homApp (0 : Γ(homMod F G, V)) h = 0

@[simp] theorem **homApp_add** (φ ψ : Γ(homMod F G, V)) (h : W ≤ V) : homApp (φ + ψ) h = homApp φ h + homApp ψ h

@[simp] theorem **homApp_smul** (a : Γ(X, V)) (φ : Γ(homMod F G, V)) (h : W ≤ V) : homApp (a • φ) h = resG X h a • homApp φ h

theorem **homApp_sum** {κ : Type*} (t : Finset κ) (φ : κ → Γ(homMod F G, V)) (h : W ≤ V) : homApp (∑ k ∈ t, φ k) h = ∑ k ∈ t, homApp (φ k) h

@[simp] theorem **homApp_res** (φ : Γ(homMod F G, V)) (h : W ≤ V) {T : X.Opens} (h' : T ≤ W) : homApp (resM (homMod F G) h φ) h' = homApp φ (h'.trans h)

@[simp] theorem **homGlobalEquiv_app** (φ : Γ(homMod F G, ⊤)) (s : Γ(F, V)) : (homGlobalEquiv F G φ).app V s = homApp φ le_top s

@[simp] theorem **homApp_homMap** {F' G' : X.Modules} (α : F' ⟶ F) (β : G ⟶ G') (φ : Γ(homMod F G, V)) (h : W ≤ V) (s : Γ(F', W)) : homApp ((homMap α β).app V φ) h s = β.app W (homApp φ h (α.app W s))

theorem **dualApp_resM** (l : Γ(dual F, V)) (h : W ≤ V) {T : X.Opens} (h' : T ≤ W) (s : Γ(F, W)) : dualApp l (h'.trans h) (resM F h' s) = resG X h' (dualApp l h s)

@[simp] theorem **dualApp_dualMk** (l : ∀ W : X.Opens, W ≤ V → (Γ(F, W) →ₗ[Γ(X, W)] Γ(X, W))) (hl) (h : W ≤ V) : dualApp (dualMk l hl) h = l W h

@[ext] theorem **dual_ext** {l l' : Γ(dual F, V)} (e : ∀ (W : X.Opens) (h : W ≤ V) (s : Γ(F, W)), dualApp l h s = dualApp l' h s) : l = l'

@[simp] theorem **dualApp_zero** (h : W ≤ V) : dualApp (0 : Γ(dual F, V)) h = 0

@[simp] theorem **dualApp_add** (l l' : Γ(dual F, V)) (h : W ≤ V) : dualApp (l + l') h = dualApp l h + dualApp l' h

@[simp] theorem **dualApp_smul** (a : Γ(X, V)) (l : Γ(dual F, V)) (h : W ≤ V) : dualApp (a • l) h = resG X h a • dualApp l h

theorem **dualApp_sum** {κ : Type*} (t : Finset κ) (l : κ → Γ(dual F, V)) (h : W ≤ V) : dualApp (∑ k ∈ t, l k) h = ∑ k ∈ t, dualApp (l k) h

@[simp] theorem **dualApp_res** (l : Γ(dual F, V)) (h : W ≤ V) {T : X.Opens} (h' : T ≤ W) : dualApp (resM (dual F) h l) h' = dualApp l (h'.trans h)

@[simp] theorem **dualApp_dualMap** (φ : F ⟶ G) (l : Γ(dual G, V)) (h : W ≤ V) (s : Γ(F, W)) : dualApp ((dualMap φ).app V l) h s = dualApp l h (φ.app W s)

@[simp] theorem **dualMap_id** : dualMap (𝟙 F) = 𝟙 (dual F)

theorem **dualMap_comp** (φ : F ⟶ G) (ψ : G ⟶ H) : dualMap (φ ≫ ψ) = dualMap ψ ≫ dualMap φ

theorem **dualMap_add** (φ ψ : F ⟶ G) : dualMap (φ + ψ) = dualMap φ + dualMap ψ

@[simp] theorem **dualMap_zero** : dualMap (0 : F ⟶ G) = 0

@[simp] theorem **dualMap_smulHom** (a : Γ(X, ⊤)) : dualMap (smulHom F a) = smulHom (dual F) a

@[simp] theorem **dualSheafSectionApply_toBidual** (s : Γ(F, V)) (h : W ≤ V) (l : Γ(dual F, W)) : dualApp ((cartierDualToBidual F).app V s) h l = dualApp l le_rfl (resM F h s)

@[reassoc] theorem **toBidual_naturality_dualSheafTranspose** (φ : F ⟶ G) : φ ≫ cartierDualToBidual G = cartierDualToBidual F ≫ dualMap (dualMap φ)

@[simp] theorem **dualApp_coord** (he : IsFrame e) (i : ι) (h : V ≤ U) (s : Γ(F, V)) : dualApp (he.coord i) h s = (he.basis h).repr s i

theorem **dualApp_coord_frame** [DecidableEq ι] (he : IsFrame e) (i j : ι) (h : V ≤ U) : dualApp (he.coord i) h (resM F h (e j)) = if j = i then 1 else 0

theorem **dual_ext** (he : IsFrame e) (h : V ≤ U) {l l' : Γ(dual F, V)} (H : ∀ i, dualApp l le_rfl (resM F h (e i)) = dualApp l' le_rfl (resM F h (e i))) : l = l'

private theorem **dualApp_sum_smul_coord_eq** (he : IsFrame e) (h : V ≤ U) (c : ι → Γ(X, V)) (j : ι) : dualApp (∑ i, c i • resM (VB.dual F) h (he.coord i)) le_rfl (resM F h (e j)) = c j

theorem **dual** (he : IsFrame e) : IsFrame he.coord

theorem **dual_basis_repr** (he : IsFrame e) (h : V ≤ U) (l : Γ(VB.dual F, V)) (i : ι) : (he.dual.basis h).repr l i = dualApp l le_rfl (resM F h (e i))

theorem **toBidual_app** (he : IsFrame e) (i : ι) : (cartierDualToBidual F).app U (e i) = he.dual.coord i

theorem **IsVB.dual** (h : IsVB F r) : IsVB (dual F) r

theorem **isIso_of_frames** (φ : F ⟶ G) (h : ∀ x : X, ∃ (U : X.Opens) (e : ι → Γ(F, U)), x ∈ U ∧ IsFrame e ∧ IsFrame fun i => φ.app U (e i)) : IsIso φ

theorem **IsVB.isIso_toBidual_of_local_frames** (h : IsVB F r) : IsIso (cartierDualToBidual F)

theorem **resG_det** (he : IsFrame e) (h : V ≤ U) (h' : W ≤ V) (s : ι → Γ(F, V)) : resG X h' ((he.basis h).det s) = (he.basis (h'.trans h)).det fun i => resM F h' (s i)

theorem **val_detUnit** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') : (he.detUnit he' h h' : Γ(X, V)) = (he.basis h).det fun i => resM F h' (e' i)

theorem **res_detUnit** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') (hW : W ≤ V) : Semi.res hW (he.detUnit he' h h') = he.detUnit he' (hW.trans h) (hW.trans h')

theorem **detUnit_res** (he : IsFrame e) (he' : IsFrame e') {T T' : X.Opens} (hT : T ≤ U) (hT' : T' ≤ U') (h : V ≤ T) (h' : V ≤ T') : (he.res hT).detUnit (he'.res hT') h h' = he.detUnit he' (h.trans hT) (h'.trans hT')

theorem **detUnit_self** (he : IsFrame e) (h h' : V ≤ U) : he.detUnit he h h' = 1

theorem **detUnit_mul** (he : IsFrame e) (he' : IsFrame e') (he'' : IsFrame e'') (h : V ≤ U) (h' : V ≤ U') (h'' : V ≤ U'') : he.detUnit he' h h' * he'.detUnit he'' h' h'' = he.detUnit he'' h h''

theorem **det_eq_detUnit_mul** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') (s : ι → Γ(F, V)) : (he.basis h).det s = he.detUnit he' h h' * (he'.basis h').det s

theorem **detUnit_dual** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') : he.dual.detUnit he'.dual h h' = (he.detUnit he' h h')⁻¹

private theorem **repr_map_iso_app_eq_repr** (he : IsFrame e) (φ : F ⟶ G) [IsIso φ] (h : V ≤ U) (s : Γ(F, V)) (i : ι) : ((he.map_iso φ).basis h).repr (φ.app V s) i = (he.basis h).repr s i

theorem **detUnit_map_iso** (he : IsFrame e) (he' : IsFrame e') (φ : F ⟶ G) [IsIso φ] (h : V ≤ U) (h' : V ≤ U') : (he.map_iso φ).detUnit (he'.map_iso φ) h h' = he.detUnit he' h h'

private theorem **pullbackSections_resM** [F.IsQuasicoherent] (f : Y ⟶ X) (h : V ≤ U) {T : Y.Opens} (hT : T ≤ f ⁻¹ᵁ V) (x : Γ(F, U)) : QCoh.pullbackSections f F hT (resM F h x) = resM _ (hT.trans (f.preimage_mono h)) (QCoh.pullbackSections f F (le_refl (f ⁻¹ᵁ U)) x)

theorem **pullback_basis_repr** [F.IsQuasicoherent] (he : IsFrame e) (f : Y ⟶ X) (h : V ≤ U) {T : Y.Opens} (hT : T ≤ f ⁻¹ᵁ V) (s : Γ(F, V)) (i : ι) : ((he.pullback f).basis (hT.trans (f.preimage_mono h))).repr (QCoh.pullbackSections f F hT s) i = f.appLE V T hT ((he.basis h).repr s i)

theorem **val_detUnit_pullback** [F.IsQuasicoherent] (he : IsFrame e) (he' : IsFrame e') (f : Y ⟶ X) (h : V ≤ U) (h' : V ≤ U') {T : Y.Opens} (hT : T ≤ f ⁻¹ᵁ V) : (((he.pullback f).detUnit (he'.pullback f) (hT.trans (f.preimage_mono h)) (hT.trans (f.preimage_mono h')) : Γ(Y, T)ˣ) : Γ(Y, T)) = (f.appLE V T hT).hom (he.detUnit he' h h' : Γ(X, V))

theorem **isLB_det_iff** : (det F r).IsLB ↔ IsVB F r

theorem **IsVB.isLB_det** (h : IsVB F r) : (det F r).IsLB

theorem **det_sec_ext** {t t' : Γ(toMod (det F r), V)} (h : ∀ c, detCoord t c = detCoord t' c) : t = t'

@[simp] theorem **detCoord_add** (t t' : Γ(toMod (det F r), V)) (c : Chart F r) : detCoord (t + t') c = detCoord t c + detCoord t' c

@[simp] theorem **detCoord_smul** (a : Γ(X, V)) (t : Γ(toMod (det F r), V)) (c : Chart F r) : detCoord (a • t) c = resG X inf_le_left a * detCoord t c

@[simp] theorem **detCoord_zero** (c : Chart F r) : detCoord (0 : Γ(toMod (det F r), V)) c = 0

theorem **detCoord_wedgeFun** (s : Fin r → Γ(F, V)) (c : Chart F r) : detCoord (wedgeFun s) c = (c.isFrame.basis (inf_le_right : V ⊓ c.U ≤ c.U)).det fun i => resM F inf_le_left (s i)

theorem **detCoord_wedge** (s : Fin r → Γ(F, V)) (c : Chart F r) : detCoord (wedge F V r s) c = (c.isFrame.basis (inf_le_right : V ⊓ c.U ≤ c.U)).det fun i => resM F inf_le_left (s i)

theorem **resM_wedge** (h : W ≤ V) (s : Fin r → Γ(F, V)) : resM _ h (wedge F V r s) = wedge F W r fun i => resM F h (s i)

theorem **wedge_frame** (h : IsVB F r) (c : Chart F r) : wedge F c.U r c.e = LB.chartSec h.isLB_det c

theorem **IsFrame.wedge** (h : IsVB F r) {e : Fin r → Γ(F, U)} (he : IsFrame e) : IsFrame fun _ : Fin 1 => wedge F U r e

theorem **lbIso_det_of_frames** {L : LB X} (hL : L.IsLB) (e : ∀ i, Fin r → Γ(F, L.U i)) (he : ∀ i, IsFrame (e i)) (hg : ∀ i j, L.g i j = (he i).detUnit (he j) inf_le_left inf_le_right) : IsVB F r ∧ LBIso L (det F r)

theorem **IsVB.lbIso_det_of_iso** (h : IsVB F r) (φ : F ≅ G) : LBIso (det F r) (det G r)

theorem **IsVB.lbIso_det_dual** (h : IsVB F r) : LBIso (CechPic.dual (det F r)) (det (VB.dual F) r)

theorem **IsVB.lbIso_det_pullback** (h : IsVB F r) (f : Y ⟶ X) : LBIso ((det F r).pull f) (det ((Scheme.Modules.pullback f).obj F) r)

theorem **lbIso_det_toMod** {L : LB X} (hL : L.IsLB) : LBIso L (det (toMod L) 1)

theorem **IsFrame.cls_det_eq_zero** {e : Fin r → Γ(F, ⊤)} (he : IsFrame e) (h : (det F r).IsLB) : CechPic.cls _ h = 0

Development.AbelianVarietyTheory.Definitions.Divisors.ToIThCapDegOfAddNsmulDivP

theorem **natCast_ne_zero** : (p : O) ≠ 0

theorem **natCast_mem_maximalIdeal** : (p : O) ∈ maximalIdeal O

theorem **natCast_ramIdx** : (ramIdx O p : ℕ∞) = IsDiscreteValuationRing.addVal O p

theorem **ramIdx_ne_zero** : ramIdx O p ≠ 0

theorem **charP_residueField_padicInt** (p : ℕ) [Fact p.Prime] : CharP (ResidueField ℤ_[p]) p

theorem **exists_isPullback_gen** (K : Type) [Field K] [Algebra O K] [IsFractionRing O K] : ∃ q : Z.gen.toScheme ⟶ Sp K, IsPullback Z.gen.ι q Z.str (spMap (algebraMap O K))

@[simp] theorem **cDQ_toQ** (D : Z.VDivZ) : Z.cDQ (toQ D) = firstChernPic (Z.vCls D)

@[simp] theorem **vDegQ_toQ** (D : Z.VDivZ) : Z.vDegQ (toQ D) = Z.vDeg D

theorem **capDeg_vp_toQ** (p : ℕ) (D : Z.VDivZ) (Q : PicSym Z.X) : Z.capDeg_vp p (toQ D) Q = Z.vDeg D Q / ramIdx O p

@[simp] theorem **vPullQ_toQ** {Z₁ : Model O} {f : Z₁.X ⟶ Z.X} (hf : f ≫ Z.str = Z₁.str) (D : Z.VDivZ) : vPullQ hf (toQ D) = toQ (vPull hf D)

theorem **divisorOfPrime_eq_ramificationIdx_smul** {ϖ : O} (hϖ : Irreducible ϖ) : Z.divP p = ramIdx O p • Z.div ϖ hϖ.ne_zero

theorem **toQ_div** (g : O) (hg : g ≠ 0) : toQ (Z.div g hg) = (((IsDiscreteValuationRing.addVal O g).toNat : ℚ) / ramIdx O p) • toQ (Z.divP p)

theorem **toITh_capDeg_eq_zero** (D : Z.VDivQ) {n : ℕ} (hn : n ≠ Z.relDim) {x : PicSym Z.X} (hx : x ∈ symDeg (SmallPic Z.X) n) : (Z.toITh p).capDeg D x = 0

theorem **FlagPair.toFPair_ofSMul_Dv** (fl : Flag E N) (i : Fin N) : ((FlagPair.ofSMul fl p (natCast_ne_zero O p)).toFPair p).Dv i = (Z.toITh p).divP

@[simp] theorem **relDim_base** : (base O).relDim = 0

@[simp] theorem **relDim_toBase** : Z.toBase.relDim = Z.relDim

theorem **chiLen_twist_effConst** {g : O} (hg : g ≠ 0) {M : LB Z.X} (hM : M.IsLB) (h : ∀ n, 0 < n → Subsingleton (coh (twist (structureModule Z.X) M) n)) [Module.Free O (coh (twist (structureModule Z.X) M) 0)] : chiLen O (twist (divMod (Z.effConst g hg).1) M) = (IsDiscreteValuationRing.addVal O g).toNat * Module.finrank O (coh (twist (structureModule Z.X) M) 0)

theorem **isNormalScheme_projSpace** : IsNormalScheme (projSpace O N).X

theorem **algebraTop_projSpace** : (projSpace O N).algebraTop = algebraSections O N ⊤

theorem **chiLen_projSpace** {g : O} (hg : g ≠ 0) {j : ℕ} (hj : j ≤ N + 1) : chiLenPic O (divMod ((projSpace O (N + 1)).effConst g hg).1) (-(j : ℤ) • cls (formalTwistingSheaf O (N + 1)) (O1_isLB O _)) = if j = 0 then ((IsDiscreteValuationRing.addVal O g).toNat : ℤ) else 0

theorem **vDeg_div_projSpace** {g : O} (hg : g ≠ 0) : (projSpace O (N + 1)).vDeg ((projSpace O (N + 1)).div g hg) (c1 (formalTwistingSheaf O (N + 1)) (O1_isLB O _) ^ (N + 1)) = (IsDiscreteValuationRing.addVal O g).toNat

theorem **capDeg_divP_projSpace** : ((projSpace O (N + 1)).toITh p).capDeg ((projSpace O (N + 1)).toITh p).divP (c1 (formalTwistingSheaf O (N + 1)) (O1_isLB O _) ^ (N + 1)) = 1

theorem **capDeg_divP_line** : ((projSpace O 1).toITh p).capDeg ((projSpace O 1).toITh p).divP (c1 (formalTwistingSheaf O 1) (O1_isLB O 1)) = 1

theorem **fibreDeg_line** : (projSpace O 1).toBase.fibreDeg (c1 (formalTwistingSheaf O 1) (O1_isLB O 1)) = 1

Development.AbelianVarietyTheory.Definitions.Divisors.ToKnownTheoremsOf

theorem **Hypotheses2.p18** (_h : Hypotheses2) : GlobalTheory.Published.FinitenessOfHermitianClasses

theorem **Hypotheses2.q2** (_h : Hypotheses2) : GlobalTheory.Published.CanonicalExtensionRankOne

theorem **Hypotheses2.q12** (_h : Hypotheses2) : GlobalTheory.Published.ProperIntersectionLength

theorem **Hypotheses2.toClassicalTheoremsOf** (h : Hypotheses2) (hM : GlobalTheory.Published.GoodChernWeil → GlobalTheory.Published.MumfordProportionality) : Hypotheses₂

Development.AbelianVarietyTheory.Definitions.Divisors.ToKnownTheoremsX

theorem **HypothesesXH.ks** (h : HypothesesXH) : GlobalTheory.Published.ToroidalForFanKS

theorem **HypothesesXH.lie** (h : HypothesesXH) : GlobalTheory.Published.SemiAbelianLieSeminegative

theorem **HypothesesXH.green** (h : HypothesesXH) : GlobalTheory.Published.GreenCuspidal

theorem **HypothesesXH.sep** (h : HypothesesXH) : GlobalTheory.Published.PointSeparationPair

Development.AbelianVarietyTheory.Definitions.Divisors.ToNatLengthEqRangeAddRange

theorem **length_eq_range_add_range** {A B C : Type*} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] [Module R A] [Module R B] [Module R C] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Function.Exact f g) : Module.length R B = Module.length R (LinearMap.range f) + Module.length R (LinearMap.range g)

theorem **toNat_length_eq_range_add_range** {A B C : Type*} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] [Module R A] [Module R B] [Module R C] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Function.Exact f g) (hB : Module.length R B ≠ ⊤) : ((Module.length R B).toNat : ℤ) = (Module.length R (LinearMap.range f)).toNat + (Module.length R (LinearMap.range g)).toNat

theorem **length_ne_top_of_exact** {A B C : Type*} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] [Module R A] [Module R B] [Module R C] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Function.Exact f g) (hA : Module.length R A ≠ ⊤) (hC : Module.length R C ≠ ⊤) : Module.length R B ≠ ⊤

theorem **chiLen_eq_sum_range** (N : ℕ) (hN : ∀ n, N ≤ n → Subsingleton (coh F n)) : chiLen R F = ∑ n ∈ Finset.range N, (-1 : ℤ) ^ n * (Module.length R (coh F n)).toNat

theorem **chiLen_eq_of_linearEquiv** [Algebra R Γ(Y, ⊤)] {H : Y.Modules} (e : ∀ n, coh F n ≃ₗ[R] coh H n) : chiLen R F = chiLen R H

theorem **chiLen_congr** (e : F ≅ G) : chiLen R F = chiLen R G

theorem **chiLen_eq_zero_of_isZero** (h : IsZero F) : chiLen R F = 0

theorem **chiLen_eq_chi** (k : Type) [Field k] [Algebra k Γ(X, ⊤)] (F : X.Modules) : chiLen k F = chi k F

theorem **chiLen_eq_of_surjective** (k : Type) [CommRing k] [Algebra R k] [Algebra k Γ(X, ⊤)] [IsScalarTower R k Γ(X, ⊤)] (h : Function.Surjective (algebraMap R k)) (F : X.Modules) : chiLen R F = chiLen k F

theorem **of_iso** (h : FiniteLenCoh R F) (e : F ≅ G) : FiniteLenCoh R G where

theorem **of_shortExact₂** (hS : S.ShortExact) (h₁ : FiniteLenCoh R S.X₁) (h₃ : FiniteLenCoh R S.X₃) : FiniteLenCoh R S.X₂ where

theorem **of_shortExact₃** (hS : S.ShortExact) (h₁ : FiniteLenCoh R S.X₁) (h₂ : FiniteLenCoh R S.X₂) : FiniteLenCoh R S.X₃ where

theorem **chiLen_shortExact** (hS : S.ShortExact) (h₁ : FiniteLenCoh R S.X₁) (h₃ : FiniteLenCoh R S.X₃) : chiLen R S.X₂ = chiLen R S.X₁ + chiLen R S.X₃

@[simp] theorem **chiLenPic_cls** (hM : M.IsLB) : chiLenPic R F (cls M hM) = chiLen R (twist F M)

theorem **chiLenPic_congr** (e : F ≅ G) : chiLenPic R F = chiLenPic R G

theorem **chiLenPic_twist** (hM : M.IsLB) (c : Pic X) : chiLenPic R (twist F M) c = chiLenPic R F (cls M hM + c)

theorem **chiLenPic_eq_zero_of_isZero** (h : IsZero F) : chiLenPic R F = 0

theorem **interLen_eq_multiDiff** {ι : Type} [Fintype ι] {L : ι → LB X} (hL : ∀ i, (L i).IsLB) : interLen R F L = multiDiff (fun i => cls (L i) (hL i)) (chiLenPic R F) 0

theorem **chiLenPic_pushforward** (k : Type) [Field k] [Algebra R k] [Algebra k Γ(Y, ⊤)] (hk : Function.Surjective (algebraMap R k)) (i : Y ⟶ X) [IsClosedImmersion i] (hi : ∀ r : R, i.appTop (algebraMap R Γ(X, ⊤) r) = algebraMap k Γ(Y, ⊤) (algebraMap R k r)) (H : Y.Modules) : chiLenPic R ((Scheme.Modules.pushforward i).obj H) = chiPic k H ∘ pull i

theorem **FiniteLenTwists.of_iso** (h : FiniteLenTwists R F) (e : F ≅ G) : FiniteLenTwists R G

theorem **FiniteLenTwists.twist** (h : FiniteLenTwists R F) (hM : M.IsLB) : FiniteLenTwists R (twist F M)

theorem **FiniteLenTwists.of_shortExact₃** (hS : S.ShortExact) (h₁ : FiniteLenTwists R S.X₁) (h₂ : FiniteLenTwists R S.X₂) : FiniteLenTwists R S.X₃

theorem **chiLenPic_shortExact** (hS : S.ShortExact) (h₁ : FiniteLenTwists R S.X₁) (h₃ : FiniteLenTwists R S.X₃) : chiLenPic R S.X₂ = chiLenPic R S.X₁ + chiLenPic R S.X₃

theorem **bwdDiff_chiLenPic** {L : LB X} (hL : L.IsLB) (s : L.Sec) [Mono (cutHom F hL s)] (h : FiniteLenTwists R F) : FiniteLenTwists R (cut F hL s) ∧ bwdDiff (cls L hL) (chiLenPic R F) = chiLenPic R (cut F hL s)

theorem **diffVanish_chiLenPic** [IsLocallyNoetherian X] {A : LB X} (hA : IsAmpleLB A) [IsFiniteQC F] (h : FiniteLenTwists R F) {n : ℕ} (hn : suppDim F < n) : DiffVanish n (chiLenPic R F)

theorem **exists_smulHom_pow_eq_zero** [CompactSpace X] [IsFiniteQC F] (a : Γ(X, ⊤)) (h : ∀ x ∈ support F, x ∉ X.basicOpen a) : ∃ m : ℕ, smulHom F (a ^ m) = 0

theorem **finiteLenTwists_of_support** [IsDomain R] [IsNoetherianRing R] [Ring.KrullDimLE 1 R] {f : X ⟶ Sp R} (hf : ∀ r, algebraMap R Γ(X, ⊤) r = f.appTop ((Scheme.ΓSpecIso (.of R)).inv r)) (hp : IsProjOver f) [IsFiniteQC F] {n : ℕ} (hn : suppDim F < n) {a : R} (ha : a ≠ 0) (h : ∀ x ∈ support F, x ∉ X.basicOpen (algebraMap R Γ(X, ⊤) a)) : FiniteLenTwists R F

theorem **symOf_apply** (g : G) : symOf g = SymmetricAlgebra.ι ℚ _ ((1 : ℚ) ⊗ₜ[ℤ] g)

@[simp] theorem **symLift_symOf** {A : Type*} [CommRing A] [Algebra ℚ A] (f : G →+ A) (g : G) : symLift f (symOf g) = f g

@[simp] theorem **symMap_symOf** (f : G →+ H) (g : G) : symMap f (symOf g) = symOf (f g)

theorem **span_prod_symOf** : Submodule.span ℚ (Set.range fun s : Multiset G => (s.map symOf).prod) = (⊤ : Submodule ℚ (SymQ G))

theorem **symQ_linearMap_ext** {N : Type*} [AddCommGroup N] [Module ℚ N] {l l' : SymQ G →ₗ[ℚ] N} (h : ∀ (n : ℕ) (g : Fin n → G), l (∏ i, symOf (g i)) = l' (∏ i, symOf (g i))) : l = l'

theorem **exists_topForm** (h : DiffVanish (d + 1) φ) : ∃ l : SymQ G →ₗ[ℚ] ℚ, ∀ (ι : Type) [Fintype ι] (g : ι → G), l (∏ i, symOf (g i)) = if Fintype.card ι = d then ((multiDiff g φ 0 : ℤ) : ℚ) else 0

theorem **topForm_prod** (h : DiffVanish (d + 1) φ) {ι : Type} [Fintype ι] (g : ι → G) : topForm h (∏ i, symOf (g i)) = if Fintype.card ι = d then ((multiDiff g φ 0 : ℤ) : ℚ) else 0

theorem **topForm_eq_zero** (h : DiffVanish (d + 1) φ) {n : ℕ} (hn : n ≠ d) {x : SymQ G} (hx : x ∈ symDeg G n) : topForm h x = 0

theorem **topForm_eq_add** {χ : G → ℤ} (h : DiffVanish (d + 1) χ) (h₁ : DiffVanish (d + 1) φ) (h₂ : DiffVanish (d + 1) ψ) (a : G) (e : ∀ x, χ x = φ (a + x) + ψ x) : topForm h = topForm h₁ + topForm h₂

@[simp] theorem **smallPicEquiv_smallCls** (L : LB X) (hL : L.IsLB) : smallPicEquiv X (smallCls L hL) = cls L hL

theorem **smallCls_surj** (c : SmallPic X) : ∃ L hL, smallCls L hL = c

theorem **smallCls_eq_iff** {L M : LB X} (hL : L.IsLB) (hM : M.IsLB) : smallCls L hL = smallCls M hM ↔ LBIso L M

theorem **smallPull_smallCls** (f : Y ⟶ X) {L : LB X} (hL : L.IsLB) (h : (L.pull f).IsLB) : smallPull f (smallCls L hL) = smallCls (L.pull f) h

@[simp] theorem **firstChernPic_class** (L : LB X) (hL : L.IsLB) : firstChernPic (cls L hL) = c1 L hL

theorem **firstChernTensor** {L M : LB X} (hL : L.IsLB) (hM : M.IsLB) (h : (tens L M).IsLB) : c1 (tens L M) h = c1 L hL + c1 M hM

theorem **firstChern_trivial** (h : (LB.triv X).IsLB) : c1 (LB.triv X) h = 0

theorem **firstChernCongr** {L M : LB X} (hL : L.IsLB) (hM : M.IsLB) (h : LBIso L M) : c1 L hL = c1 M hM

theorem **pullback_firstChern** (f : Y ⟶ X) {L : LB X} (hL : L.IsLB) (h : (L.pull f).IsLB) : picSymPull f (c1 L hL) = c1 (L.pull f) h

theorem **picSymPull_c1Pic** (f : Y ⟶ X) (c : Pic X) : picSymPull f (firstChernPic c) = firstChernPic (pull f c)

theorem **picForm_prod** (h : DiffVanish (d + 1) φ) {ι : Type} [Fintype ι] (g : ι → Pic X) : picForm h (∏ i, firstChernPic (g i)) = if Fintype.card ι = d then ((multiDiff g φ 0 : ℤ) : ℚ) else 0

theorem **picSym_linearMap_ext** {N : Type*} [AddCommGroup N] [Module ℚ N] {l l' : PicSym X →ₗ[ℚ] N} (h : ∀ (n : ℕ) (g : Fin n → Pic X), l (∏ i, firstChernPic (g i)) = l' (∏ i, firstChernPic (g i))) : l = l'

theorem **picForm_eq_add** {χ : Pic X → ℤ} (h : DiffVanish (d + 1) χ) (h₁ : DiffVanish (d + 1) φ) (h₂ : DiffVanish (d + 1) ψ) (a : Pic X) (e : ∀ x, χ x = φ (a + x) + ψ x) : picForm h = picForm h₁ + picForm h₂

theorem **picForm_comap** (f : Y ⟶ X) {χ : Pic Y → ℤ} (h : DiffVanish (d + 1) χ) (x : PicSym X) : picForm (h.comap (pull f)) x = picForm h (picSymPull f x)

theorem **picForm_eq_zero** (h : DiffVanish (d + 1) φ) {n : ℕ} (hn : n ≠ d) {x : PicSym X} (hx : x ∈ symDeg (SmallPic X) n) : picForm h x = 0

Development.AbelianVarietyTheory.Definitions.Divisors.ToRingActOpOfRingAct

theorem **IsRAct.of_oAct** {O : Type} [CommRing O] {A : Over S} [GrpObj A] (i : OAct O A) : IsRAct A i.act

theorem **IsRAct.op_of_ringAct** {A : Over S} [GrpObj A] [IsCommMonObj A] (ρ : Out.RingAct A O) : IsRAct A (fun a : Oᵐᵒᵖ => ρ.act a.unop)

@[simp] theorem **IsRAct.toRingAct_act** {A : Over S} [GrpObj A] [IsCommMonObj A] {act : Oᵐᵒᵖ → (A ⟶ A)} (h : IsRAct A act) (a : O) : h.toRingAct.act a = act (MulOpposite.op a)

theorem **IsRAct.toRingAct_op_of_ringAct** {A : Over S} [GrpObj A] [IsCommMonObj A] (ρ : Out.RingAct A O) (a : O) : (IsRAct.op_of_ringAct ρ).toRingAct.act a = ρ.act a

@[simp] theorem **IsRAct.toOAct_act** {O : Type} [CommRing O] {A : Over S} [GrpObj A] {act : O → (A ⟶ A)} (h : IsRAct A act) : h.toOAct.act = act

theorem **IsRAct.act_mul_comm** {O : Type} [CommRing O] {A : Over S} [GrpObj A] {act : O → (A ⟶ A)} (h : IsRAct A act) (a b : O) : act (a * b) = act b ≫ act a

theorem **IsRAct.comp_comm** {O : Type} [CommRing O] {A : Over S} [GrpObj A] {act : O → (A ⟶ A)} (h : IsRAct A act) (a b : O) : act a ≫ act b = act b ≫ act a

Development.AbelianVarietyTheory.Definitions.Divisors.ToSpanSingletonConstGenInjective

theorem **fibreDim_eq** (n : ℕ) (q : PrimeSpectrum O) : fibreDim (G.A n) q = p ^ (n * G.height)

theorem **mulBy_pow_self** (n : ℕ) (a : G.A n) : mulBy O (G.A n) (p ^ n) a = algebraMap O _ (counit a)

theorem **pts_pow_self** (n : ℕ) {B : Type*} [CommRing B] [Algebra O B] (x : WithConv (G.A n →ₐ[O] B)) : x ^ p ^ n = 1

theorem **hom_pow_self** (n : ℕ) {B : Type*} [CommRing B] [Bialgebra O B] (x : WithConv (G.A n →ₐc[O] B)) : x ^ p ^ n = 1

theorem **exists_comp_res_iff** (n : ℕ) {B : Type*} [CommRing B] [Algebra O B] (x : G.A (n + 1) →ₐ[O] B) : (∃ y : G.A n →ₐ[O] B, y.comp (G.res n : G.A (n + 1) →ₐ[O] G.A n) = x) ↔ toConv x ^ p ^ n = 1

@[simp] theorem **mulP_res** (n : ℕ) (a : G.A (n + 1)) : G.mulP n (G.res n a) = mulBy O _ p a

@[simp] theorem **res_mulP** (n : ℕ) (a : G.A n) : G.res n (G.mulP n a) = mulBy O _ p a

theorem **mulP_comp_res** (n : ℕ) : (G.mulP n).comp (G.res n) = mulBy O _ p

theorem **res_comp_mulP** (n : ℕ) : (G.res n).comp (G.mulP n) = mulBy O _ p

theorem **exists_res_le** {m n : ℕ} (h : m ≤ n) : ∃ f : G.A n →ₐc[O] G.A m, Function.Surjective f ∧ RingHom.ker f = torsionIdeal O _ (p ^ m)

theorem **dual_mulP_surjective** (n : ℕ) : Function.Surjective (CartierDual.map (G.mulP n))

theorem **ker_dual_mulP** (n : ℕ) : RingHom.ker (CartierDual.map (G.mulP n)) = torsionIdeal O (CartierDual O (G.A (n + 1))) (p ^ n)

theorem **dual_mulP** (n : ℕ) : G.dual.mulP n = CartierDual.map (G.res n)

@[ext] theorem **hom_ext** {f g : G ⟶ H} (h : ∀ n, f.app n = g.app n) : f = g

@[simp] theorem **id_app** (n : ℕ) : Hom.app (𝟙 G) n = BialgHom.id O _

@[simp] theorem **comp_app** (f : G ⟶ H) (g : H ⟶ K) (n : ℕ) : Hom.app (f ≫ g) n = (f.app n).comp (g.app n)

theorem **res_app** (f : G ⟶ H) (n : ℕ) (a : H.A (n + 1)) : G.res n (f.app (n + 1) a) = f.app n (H.res n a)

theorem **app_mulP** (f : G ⟶ H) (n : ℕ) (a : H.A n) : f.app (n + 1) (H.mulP n a) = G.mulP n (f.app n a)

theorem **hom_ext_conv** {f g : G ⟶ H} (h : ∀ n, f.conv n = g.conv n) : f = g

theorem **Hom.comm_conv** (f : G ⟶ H) (n : ℕ) : postcomp _ (G.res n) (f.conv (n + 1)) = precomp _ (H.res n) (f.conv n)

@[simp] theorem **zero_conv** (n : ℕ) : Hom.conv (0 : G ⟶ H) n = 1

@[simp] theorem **add_conv** (f g : G ⟶ H) (n : ℕ) : Hom.conv (f + g) n = f.conv n ∗ g.conv n

@[simp] theorem **smul_conv** (a : ℤ_[p]) (f : G ⟶ H) (n : ℕ) : Hom.conv (a • f) n = f.conv n ^ digit n a

theorem **comp_conv_left** (f : G ⟶ H) (g : H ⟶ K) (n : ℕ) : Hom.conv (f ≫ g) n = precomp (G.A n) (g.app n) (f.conv n)

theorem **comp_conv_right** (f : G ⟶ H) (g : H ⟶ K) (n : ℕ) : Hom.conv (f ≫ g) n = postcomp (K.A n) (f.app n) (g.conv n)

theorem **natCast_smul_id_app** (G : PDivGroup p O) (k n : ℕ) : Hom.app ((k : ℤ_[p]) • 𝟙 G) n = mulBy O _ k

theorem **pow_smul_conv** (f : G ⟶ H) (n : ℕ) : Hom.conv ((p : ℤ_[p]) ^ n • f) n = 1

theorem **Action.eq_padicAction** (ρ : Action ℤ_[p] G) : ρ = G.padicAction

theorem **Action.map_algebraMap** {Λ : Type*} [Semiring Λ] [Algebra ℤ_[p] Λ] (ρ : Action Λ G) (a : ℤ_[p]) : ρ (algebraMap ℤ_[p] Λ a) = algebraMap ℤ_[p] (End G) a

@[simp] theorem **dualHom_app** (f : G ⟶ H) (n : ℕ) : (dualHom G H f).app n = CartierDual.map (f.app n)

@[simp] theorem **dualHom_id** : dualHom G G (𝟙 G) = 𝟙 G.dual

theorem **dualHom_comp** (f : G ⟶ H) (g : H ⟶ K) : dualHom G K (f ≫ g) = dualHom H K g ≫ dualHom G H f

theorem **dualHom_dualHom_bidual_inv** (f : G ⟶ H) : dualHom _ _ (dualHom G H f) ≫ H.bidual.inv = G.bidual.inv ≫ f

theorem **bidual_hom_dualHom_dualHom** (f : G ⟶ H) : G.bidual.hom ≫ dualHom _ _ (dualHom G H f) = f ≫ H.bidual.hom

theorem **dualHom_bidual_inv** : dualHom _ _ G.bidual.inv = G.dual.bidual.hom

theorem **dualHom_bijective** : Function.Bijective (dualHom G H)

@[simp] theorem **dualHomEquiv_apply** (f : G ⟶ H) : dualHomEquiv G H f = dualHom G H f

theorem **ext_conv** {x y : G.TateOver B} (h : ∀ n, x.conv n = y.conv n) : x = y

theorem **compat_conv** (x : G.TateOver B) (n : ℕ) : pointsMap B (G.mulP n) (x.conv (n + 1)) = x.conv n

theorem **pt_mulP** (x : G.TateOver B) (n : ℕ) (a : G.A n) : x.pt (n + 1) (G.mulP n a) = x.pt n a

theorem **conv_succ_pow** (x : G.TateOver B) (n : ℕ) : x.conv (n + 1) ^ p = pointsMap B (G.res n) (x.conv n)

@[simp] theorem **zero_conv** (n : ℕ) : conv (0 : G.TateOver B) n = 1

@[simp] theorem **add_conv** (x y : G.TateOver B) (n : ℕ) : conv (x + y) n = x.conv n ∗ y.conv n

@[simp] theorem **smul_conv** (a : ℤ_[p]) (x : G.TateOver B) (n : ℕ) : conv (a • x) n = x.conv n ^ digit n a

@[simp] theorem **tateMap_pt** (f : G ⟶ H) (x : G.TateOver B) (n : ℕ) (a : H.A n) : (tateMap G H B f x).pt n a = x.pt n (f.app n a)

@[simp] theorem **tateMap_id** (x : G.TateOver B) : tateMap G G B (𝟙 G) x = x

theorem **tateMap_comp** (f : G ⟶ H) (g : H ⟶ K) (x : G.TateOver B) : tateMap G K B (f ≫ g) x = tateMap H K B g (tateMap G H B f x)

@[simp] theorem **tateCongr_apply** (e : G ≅ H) (x : G.TateOver B) : tateCongr B e x = tateMap G H B e.hom x

@[simp] theorem **tateEnd_apply** (f : End G) : tateEnd G B f = tateMap G G B f

@[simp] theorem **tatePush_pt** (g : B →ₐ[O] B') (x : G.TateOver B) (n : ℕ) (a : G.A n) : (G.tatePush g x).pt n a = g (x.pt n a)

theorem **tatePush_bijective** [IsIntegrallyClosedIn O B] : Function.Bijective (G.tatePush (Algebra.ofId O B))

theorem **pow_smul_cot** (n : ℕ) (w : Cot O (G.A n)) : (p : O) ^ n • w = 0

theorem **digit_smul_cot** [Fact p.Prime] (φ : ℤ_[p] →+* O) (a : ℤ_[p]) (n : ℕ) (w : Cot O (G.A n)) : digit n a • w = φ a • w

@[simp] theorem **omegaMap_val** (f : G ⟶ H) (w : H.omega) (n : ℕ) : (omegaMap G H f w).1 n = cotMap (f.app n) (w.1 n)

@[simp] theorem **omegaMap_id** : omegaMap G G (𝟙 G) = LinearMap.id

theorem **omegaMap_comp** (f : G ⟶ H) (g : H ⟶ K) : omegaMap G K (f ≫ g) = omegaMap G H f ∘ₗ omegaMap H K g

theorem **omegaMap_smul** (φ : ℤ_[p] →+* O) (a : ℤ_[p]) (f : G ⟶ H) : omegaMap G H (a • f) = φ a • omegaMap G H f

@[simp] theorem **lieMap_apply** (f : G ⟶ H) (x : G.Lie) (w : H.omega) : lieMap G H f x w = x (omegaMap G H f w)

@[simp] theorem **lieMap_id** : lieMap G G (𝟙 G) = LinearMap.id

theorem **lieMap_comp** (f : G ⟶ H) (g : H ⟶ K) : lieMap G K (f ≫ g) = lieMap H K g ∘ₗ lieMap G H f

theorem **lieMap_smul** (φ : ℤ_[p] →+* O) (a : ℤ_[p]) (f : G ⟶ H) : lieMap G H (a • f) = φ a • lieMap G H f

@[simp] theorem **lieEnd_apply** (f : End G) : lieEnd G f = lieMap G G f

theorem **omegaDualMap_apply** (f : G ⟶ H) : omegaDualMap G H f = omegaMap H.dual G.dual (dualHom G H f)

@[simp] theorem **omegaDualMap_id** : omegaDualMap G G (𝟙 G) = LinearMap.id

theorem **omegaDualMap_comp** (f : G ⟶ H) (g : H ⟶ K) : omegaDualMap G K (f ≫ g) = omegaDualMap H K g ∘ₗ omegaDualMap G H f

@[simp] theorem **omegaDualEnd_apply** (f : End G) : omegaDualEnd G f = omegaDualMap G G f

@[simp] theorem **onTate_apply** (B : Type) [CommRing B] [Algebra O B] (a : Λ) : ρ.onTate B a = tateMap G G B (ρ a)

@[simp] theorem **onTateAlg_apply** [Algebra ℤ_[p] Λ] (B : Type) [CommRing B] [Algebra O B] (a : Λ) : ρ.onTateAlg B a = tateMap G G B (ρ a)

@[simp] theorem **onLie_apply** (a : Λ) : ρ.onLie a = lieMap G G (ρ a)

@[simp] theorem **onOmegaDual_apply** (a : Λ) : ρ.onOmegaDual a = omegaDualMap G G (ρ a)

theorem **muT_pow_succ** (n : ℕ) : muT O (p ^ n) ^ p ^ (n + 1) = 1

@[simp] theorem **muRes_muT** (n : ℕ) : muRes p O n (muT O _) = muT O _

@[simp] theorem **mu_mulP_muT** (n : ℕ) : (mu p O).mulP n (muT O _) = muT O (p ^ (n + 1)) ^ p

theorem **constFun_res** (n : ℕ) (φ : (const p O).A (n + 1)) (k : ℕ) : constFun p O n ((const p O).res n φ) k = constFun p O (n + 1) φ (p * k : ℕ)

theorem **constFun_mulP** (n : ℕ) (φ : (const p O).A n) (k : ℕ) : constFun p O (n + 1) ((const p O).mulP n φ) k = constFun p O n φ k

theorem **muRoot_pow_self** (x : (mu p O).TateOver B) (n : ℕ) : muRoot x n ^ p ^ n = 1

theorem **muRoot_succ_pow** (x : (mu p O).TateOver B) (n : ℕ) : muRoot x (n + 1) ^ p = muRoot x n

theorem **muRoot_ext** {x y : (mu p O).TateOver B} (h : ∀ n, muRoot x n = muRoot y n) : x = y

@[simp] theorem **muRoot_zero** (n : ℕ) : muRoot (0 : (mu p O).TateOver B) n = 1

@[simp] theorem **muRoot_add** (x y : (mu p O).TateOver B) (n : ℕ) : muRoot (x + y) n = muRoot x n * muRoot y n

@[simp] theorem **muRoot_smul** (a : ℤ_[p]) (x : (mu p O).TateOver B) (n : ℕ) : muRoot (a • x) n = muRoot x n ^ digit n a

theorem **pow_self_of_compat** {ζ : ℕ → B} (h0 : ζ 0 = 1) (h : ∀ n, ζ (n + 1) ^ p = ζ n) (n : ℕ) : ζ n ^ p ^ n = 1

@[simp] theorem **muRoot_muTate** (ζ : ℕ → B) (h0 : ζ 0 = 1) (h : ∀ n, ζ (n + 1) ^ p = ζ n) (n : ℕ) : muRoot (muTate O ζ h0 h) n = ζ n

@[simp] theorem **apply_charFun** (χ : WithConv (CartierDual O A →ₐ[O] O)) (φ : CartierDual O A) : φ (charFun O A χ) = χ φ

theorem **isGroupLikeElem_charFun** (χ : WithConv (CartierDual O A →ₐ[O] O)) : IsGroupLikeElem O (charFun O A χ)

@[simp] theorem **evalPoint_apply** (u : A) (hu : IsGroupLikeElem O u) (φ : CartierDual O A) : evalPoint u hu φ = φ u

@[simp] theorem **apply_charElem** (χ : G.dual.Tate) (n : ℕ) (φ : CartierDual O (G.A n)) : φ (charElem χ n) = χ.pt n φ

theorem **isGroupLikeElem_charElem** (χ : G.dual.Tate) (n : ℕ) : IsGroupLikeElem O (charElem χ n)

theorem **charElem_pow_self** (χ : G.dual.Tate) (n : ℕ) : charElem χ n ^ p ^ n = 1

@[simp] theorem **res_charElem** (χ : G.dual.Tate) (n : ℕ) : G.res n (charElem χ (n + 1)) = charElem χ n

@[simp] theorem **charElem_zero** (n : ℕ) : charElem (0 : G.dual.Tate) n = 1

@[simp] theorem **charElem_add** (χ ψ : G.dual.Tate) (n : ℕ) : charElem (χ + ψ) n = charElem χ n * charElem ψ n

@[simp] theorem **charElem_smul** (a : ℤ_[p]) (χ : G.dual.Tate) (n : ℕ) : charElem (a • χ) n = charElem χ n ^ digit n a

@[simp] theorem **charElem_tateMap** (f : G ⟶ H) (χ : H.dual.Tate) (n : ℕ) : charElem (tateMap _ _ O (dualHom G H f) χ) n = f.app n (charElem χ n)

theorem **hom_mu_ext** {f g : G ⟶ mu p O} (h : ∀ n, f.app n (muT O _) = g.app n (muT O _)) : f = g

theorem **add_app_groupLike** (f g : G ⟶ H) (n : ℕ) {u : H.A n} (hu : IsGroupLikeElem O u) : Hom.app (f + g) n u = f.app n u * g.app n u

theorem **smul_app_groupLike** (a : ℤ_[p]) (f : G ⟶ H) (n : ℕ) {u : H.A n} (hu : IsGroupLikeElem O u) : Hom.app (a • f) n u = f.app n u ^ digit n a

theorem **toMuHom_app_muT** (χ : G.dual.Tate) (n : ℕ) : (toMuHom χ).app n (muT O (p ^ n)) = charElem χ n

@[simp] theorem **charHom_app_muT** (χ : G.dual.Tate) (n : ℕ) : (charHom G χ).app n (muT O _) = charElem χ n

@[simp] theorem **charHom_symm_pt** (f : G ⟶ mu p O) (n : ℕ) (φ : CartierDual O (G.A n)) : ((charHom G).symm f).pt n φ = φ (f.app n (muT O _))

theorem **charHom_tateMap** (f : G ⟶ H) (χ : H.dual.Tate) : charHom G (tateMap _ _ O (dualHom G H f) χ) = f ≫ charHom H χ

@[simp] theorem **dlog_val** (χ : G.dual.Tate) (n : ℕ) : (dlog G χ).1 n = dUnit O (charElem χ n)

theorem **dlog_eq_omegaMap** (χ : G.dual.Tate) : dlog G χ = omegaMap G (mu p O) (charHom G χ) (dlogT p O)

theorem **dlog_smul** (φ : ℤ_[p] →+* O) (a : ℤ_[p]) (χ : G.dual.Tate) : dlog G (a • χ) = φ a • dlog G χ

theorem **dlog_tateMap** (f : G ⟶ H) (χ : H.dual.Tate) : dlog G (tateMap _ _ O (dualHom G H f) χ) = omegaMap G H f (dlog H χ)

@[simp] theorem **dlogLinear_tmul** [Algebra ℤ_[p] O] (c : O) (χ : G.dual.Tate) : dlogLinear G (c ⊗ₜ χ) = c • dlog G χ

theorem **constGen_pt** (n : ℕ) (φ : (const p O).A n) : (constGen p O).pt n φ = constFun p O n φ 1

@[simp] theorem **charHom_constGen** : charHom (mu p O) (constGen p O) = 𝟙 _

@[simp] theorem **dlog_constGen** : dlog (mu p O) (constGen p O) = dlogT p O

@[simp] theorem **dualChar_app_muT** (t : G.Tate) (n : ℕ) : (dualChar G t).app n (muT O _) = toConv (t.pt n).toLinearMap

theorem **constHom_eq** (t : G.Tate) : constHom G t = dualHom G.dual (mu p O) (dualChar G t) ≫ G.bidual.inv

theorem **constFun_constHom** (t : G.Tate) (n k : ℕ) (a : G.A n) : constFun p O n ((constHom G t).app n a) k = (t.conv n ^ k) a

@[simp] theorem **tateMap_constHom** (t : G.Tate) : tateMap _ G O (constHom G t) (constGen p O) = t

theorem **constHom_symm_apply** (f : const p O ⟶ G) : (constHom G).symm f = tateMap _ G O f (constGen p O)

theorem **constHom_tateMap** (f : G ⟶ H) (t : G.Tate) : constHom H (tateMap G H O f t) = constHom G t ≫ f

theorem **dualHom_constHom** (t : G.Tate) : dualHom _ G (constHom G t) ≫ (mu p O).bidual.inv = dualChar G t

theorem **hodgeTate_apply** (t : G.Tate) : hodgeTate G t = dlog G.dual (tateMap G G.dual.dual O G.bidual.hom t)

theorem **hodgeTate_eq_omegaMap** (t : G.Tate) : hodgeTate G t = omegaMap G.dual (mu p O) (dualChar G t) (dlogT p O)

theorem **hodgeTate_val** (t : G.Tate) (n : ℕ) : (hodgeTate G t).1 n = dUnit O (toConv (t.pt n).toLinearMap : CartierDual O (G.A n))

theorem **hodgeTate_smul** (φ : ℤ_[p] →+* O) (a : ℤ_[p]) (t : G.Tate) : hodgeTate G (a • t) = φ a • hodgeTate G t

theorem **hodgeTate_tateMap** (f : G ⟶ H) (t : G.Tate) : hodgeTate H (tateMap G H O f t) = omegaDualMap G H f (hodgeTate G t)

theorem **hodgeTate_eq_omegaDualMap** (t : G.Tate) : hodgeTate G t = omegaDualMap _ G (constHom G t) (constDlogT p O)

@[simp] theorem **hodgeTateLinear_tmul** [Algebra ℤ_[p] O] (c : O) (t : G.Tate) : hodgeTateLinear G (c ⊗ₜ t) = c • hodgeTate G t

theorem **constFun_epsHom** (ζ : ℕ → O) (h0 : ζ 0 = 1) (h : ∀ n, ζ (n + 1) ^ p = ζ n) (n k : ℕ) : constFun p O n ((epsHom ζ h0 h).app n (muT O _)) k = ζ n ^ k

theorem **pairing_apply** (t : G.TateOver B) (χ : G.dual.Tate) : pairing G B t χ = tateMap G (mu p O) B (charHom G χ) t

@[simp] theorem **muRoot_pairing** (t : G.TateOver B) (χ : G.dual.Tate) (n : ℕ) : muRoot (pairing G B t χ) n = t.pt n (charElem χ n)

theorem **pairing_tateMap** (f : G ⟶ H) (t : G.TateOver B) (χ : H.dual.Tate) : pairing H B (tateMap G H B f t) χ = pairing G B t (tateMap _ _ O (dualHom G H f) χ)

theorem **constHom_comp_charHom** (t : G.Tate) (χ : G.dual.Tate) : constHom G t ≫ charHom G χ = constHom (mu p O) (pairing G O t χ)

theorem **const_cot_eq_zero** (n : ℕ) (w : Cot O ((const p O).A n)) : w = 0

theorem **mu_cot_span** (n : ℕ) (w : Cot O ((mu p O).A n)) : ∃ c : O, c • dUnit O (muT O (p ^ n)) = w

theorem **one_add_inr_pow** (n : ℕ) : (1 + TrivSqZeroExt.inr 1 : TrivSqZeroExt O (O ⧸ Ideal.span {(p : O) ^ n})) ^ p ^ n = 1

@[simp] theorem **muDer_muT** (n : ℕ) : muDer p O n (dUnit O (muT O (p ^ n))) = 1

theorem **smul_dUnit_muT_eq_zero_iff** (n : ℕ) (c : O) : c • dUnit O (muT O (p ^ n)) = 0 ↔ (p : O) ^ n ∣ c

theorem **dlogT_ne_zero** (hp : ¬ IsUnit (p : O)) : dlogT p O ≠ 0

theorem **toSpanSingleton_dlogT_bijective** [IsAdicComplete (Ideal.span {(p : O)}) O] : Function.Bijective (LinearMap.toSpanSingleton O _ (dlogT p O))

@[simp] theorem **muLieGen_dlogT** [IsAdicComplete (Ideal.span {(p : O)}) O] : muLieGen p O (dlogT p O) = 1

theorem **lieMap_charHom** [IsAdicComplete (Ideal.span {(p : O)}) O] {G : PDivGroup p O} (χ : G.dual.Tate) (x : G.Lie) : lieMap G (mu p O) (charHom G χ) x = x (dlog G χ) • muLieGen p O

theorem **smul_constGen_pt** (a : ℤ_[p]) (n : ℕ) (φ : (const p O).A n) : (a • constGen p O).pt n φ = constFun p O n φ (PadicInt.toZModPow n a)

theorem **toSpanSingleton_constGen_injective** [Nontrivial O] : Function.Injective (LinearMap.toSpanSingleton ℤ_[p] _ (constGen p O))

Development.AbelianVarietyTheory.Definitions.Divisors.TorsTransportPrinc

theorem **int_iff** (x : K) : (∀ v : HeightOneSpectrum (𝒪 K), Valued.v ((algebraMap K (Af K) x) v) ≤ 1) ↔ x ∈ (1 : FractionalIdeal (𝒪 K)⁰ K)

theorem **exists_isLatOf** (d : ℕ) (δ : K) (ι : K →+* ℂ) (h2 : Module.finrank ℚ K = 2) (hd : 0 < d) (hδ : δ ^ 2 = -(d : K)) : ∃ Λ : PeriodPair, IsLatOf ι 1 Λ

theorem **torsTransport_princ** (ι : K →+* ℂ) (α : Kˣ) (Λ₀ : PeriodPair) (Ω₀ : ℂ) (hΩ : Ω₀ ≠ 0) (hΛ : IsLatOf ι 1 Λ₀) : ∃ Λ₁, TorsTransport ι (RingEquiv.refl ℂ) (princId α) Λ₀ Λ₁ Ω₀ (ι α * Ω₀)

theorem **conj_scalars** (E : Type) [Field E] [NumberField E] {I : Type} [Fintype I] (d s : ℕ) (κ : I → ℕ) (a a' Ω S : ℂ) (ha : a ≠ 0) (ha' : a' ≠ 0) (hΩ : Ω ≠ 0) : coordIn E d s κ Ω S = coordIn E d s κ (a * Ω) ((((a * a')⁻¹) ^ (∑ i, κ i))⁻¹ * (a ^ (2 * s * Fintype.card I + ∑ i, κ i) * a'⁻¹ ^ (∑ i, κ i) * S))

theorem **exponents_add** {E : Type} [Field E] [NumberField E] {I : Type} [Fintype I] (d s l : ℕ) (κ : I → ℕ) (ƒ : Ideal (OF E)) (z : ClassGroup (OF E) → ℚ) (hc : c_vp ƒ l ≠ 0) : betaBound_vp d s κ ƒ z l = 12 * (((cE_vp l z : ℝ) + ((2 * s * Fintype.card I : ℕ) : ℝ) * c_vp ƒ l + ((∑ i, κ i : ℕ) : ℝ) * (c_vp ƒ l + 1 + if l = 2 then 1 else 0)) + ((∑ i, κ i : ℕ) : ℝ) * (c_vp ƒ l + if l = d then 1 / 2 else 0) + ((2 * s * Fintype.card I + 2 * ∑ i, κ i : ℕ) : ℝ) * (a0 ƒ l : ℝ))

theorem **sz_inv** (l : ℕ) [Fact l.Prime] (lam : algebraicClosure ℚ ℂ →+* ℂ_[l]) (r : ℝ) : Sz l lam r ((l : ℂ)⁻¹) ↔ 1 ≤ r

theorem **oneOver_one** (l : ℕ) : OneOver (E

theorem **isCMType_ext** {E : Type} [Field E] [NumberField E] [Algebra K E] (d : ℕ) (δ : K) (ι : K →+* ℂ) (h2 : Module.finrank ℚ K = 2) (hd : 0 < d) (hδ : δ ^ 2 = -(d : K)) : IsCMType (fun i : {ψ : E →+* ℂ // ψ.comp (algebraMap K E) = ι} => i.1)

theorem **latDatum_inhabited** : ∃ (K : Type) (_ : Field K) (_ : NumberField K) (d : ℕ) (δ : K) (ι : K →+* ℂ), Module.finrank ℚ K = 2 ∧ 0 < d ∧ δ ^ 2 = -(d : K) ∧ ∃ (t : (Af K)ˣ) (Λ₀ Λ₁ : PeriodPair) (Ω₀ Ω₁ : ℂ), Ω₀ ^ 12 = latDisc Λ₀ ∧ Ω₁ ≠ Ω₀ ∧ TorsTransport ι (RingEquiv.refl ℂ) t Λ₀ Λ₁ Ω₀ Ω₁ ∧ ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (_ : Algebra K E) (I : Type) (_ : Fintype I) (φ : I → E →+* ℂ) (ƒ : Ideal (OF E)), ∀ s, 1 ≤ s → IsLatDatum ι φ s ƒ ∅

Development.AbelianVarietyTheory.Definitions.Divisors.TorsionTowersOfDivisibleGroup

@[simp] theorem **spOver_id** (B : Type) [CommRing B] [Algebra O B] : spOver (AlgHom.id O B) = 𝟙 (SpAlg O B)

theorem **spOver_left** {B C : Type} [CommRing B] [Algebra O B] [CommRing C] [Algebra O C] (φ : B →ₐ[O] C) : (spOver φ).left = spMap (φ : B →+* C)

@[simp] theorem **ptsDown_left** (m : GrpMor κ u A κ' u' A' ρ (spMap σ) φ) (B : Type) [CommRing B] [Algebra O B] [Algebra O' B] (hB : ∀ r, algebraMap O B r = algebraMap O' B (σ r)) (z : SpAlg O' B ⟶ A') : (ptsDown m B hB z).left = z.left ≫ φ

theorem **grpMor_of_isGrpPullback** (h : IsGrpPullback A' A (spMap σ) φ) : GrpMor (dp0 O) (𝟙 _) A (dp0 O') (𝟙 _) A' σ (spMap σ) φ

theorem **ptsDown_bijective** (m : GrpMor κ u A κ' u' A' ρ (spMap σ) φ) (h : IsPullback φ A'.hom A.hom (spMap σ)) (B : Type) [CommRing B] [Algebra O B] [Algebra O' B] (hB : ∀ r, algebraMap O B r = algebraMap O' B (σ r)) : Function.Bijective (ptsDown m B hB)

theorem **gmCoord_comp** (φ : B →ₐ[O] C) (z : SpAlg O B ⟶ Gm (Sp O)) : gmCoord (spOver φ ≫ z) = φ (gmCoord z)

theorem **gmCoord_ext** {x y : SpAlg O B ⟶ Gm (Sp O)} (h : gmCoord x = gmCoord y) : x = y

@[simp] theorem **gmCoord_gmPoint** (u : Bˣ) : gmCoord (gmPoint O u) = u

theorem **gmCoord_muIncl** {n : ℕ} {B : Type} [CommRing B] [Algebra O B] (φ : (mu p O).A n →ₐ[O] B) : gmCoord (spOver φ ≫ muIncl p O n) = φ (muT O (p ^ n))

theorem **gmCoord_muTower** {n : ℕ} {B : Type} [CommRing B] [Algebra O B] (φ : (mu p O).A n →ₐ[O] B) : gmCoord (spOver φ ≫ (muTower p O).y n) = φ (muT O (p ^ n))

@[simp] theorem **pts_apply** (φ : WithConv (G.A n →ₐ[O] B)) : t.pts n B φ = spOver φ.ofConv ≫ t.y n

theorem **pts_injective** : Function.Injective (t.pts n B)

theorem **exists_iff** (z : SpAlg O B ⟶ A) : (∃ φ : G.A n →ₐ[O] B, spOver φ ≫ t.y n = z) ↔ z ^ p ^ n = 1

theorem **range_pts** : Set.range (t.pts n B) = {z | z ^ p ^ n = 1}

theorem **y_pow** : t.y n ^ p ^ n = 1

@[simp] theorem **lift_y** (z : SpAlg O B ⟶ A) (hz : z ^ p ^ n = 1) : spOver (t.lift n z hz) ≫ t.y n = z

theorem **eq_one_iff** (φ : G.A n →ₐ[O] B) : spOver φ ≫ t.y n = 1 ↔ ∀ a, φ a = algebraMap O B (counit a)

theorem **mulBy_y** (k : ℕ) : spOver (mulBy O (G.A n) k : G.A n →ₐ[O] G.A n) ≫ t.y n = t.y n ^ k

theorem **mulP_y** : spOver (G.mulP n : G.A n →ₐ[O] G.A (n + 1)) ≫ t.y n = t.y (n + 1) ^ p

theorem **ringHom_ext** {n : ℕ} {B : Type} [CommRing B] {f g : G.A n →+* B} (h : spMap f ≫ (t.y n).left = spMap g ≫ (t.y n).left) : f = g

theorem **exists_ringHom** (m : GrpMor κ u A κ' u' A' ρ (spMap σ) φ) (n : ℕ) : ∃ g : G.A n →+* G'.A n, (∀ r, g (algebraMap O _ r) = algebraMap O' _ (σ r)) ∧ spMap g ≫ (t.y n).left = (t'.y n).left ≫ φ

theorem **pts_eq_ptsDown** (m : GrpMor κ u A κ' u' A' ρ (spMap σ) φ) {n : ℕ} {g : G.A n →+* G'.A n} (hg : spMap g ≫ (t.y n).left = (t'.y n).left ≫ φ) {B : Type} [CommRing B] [Algebra O B] [Algebra O' B] (hB : ∀ r, algebraMap O B r = algebraMap O' B (σ r)) (x : WithConv (G'.A n →ₐ[O'] B)) (x₀ : WithConv (G.A n →ₐ[O] B)) : t.pts n B x₀ = ptsDown m B hB (t'.pts n B x) ↔ ∀ a, x₀.ofConv a = x.ofConv (g a)

theorem **semiOf_y** (m : GrpMor κ u A κ' u' A' ρ (spMap σ) φ) (n : ℕ) : spMap ((t'.semiOf t m).app n).toRingHom ≫ (t.y n).left = (t'.y n).left ≫ φ

theorem **eq_semiOf_iff** (m : GrpMor κ u A κ' u' A' ρ (spMap σ) φ) (s : SemiHom σ G' G) : s = t'.semiOf t m ↔ ∀ n, spMap (s.app n).toRingHom ≫ (t.y n).left = (t'.y n).left ≫ φ

theorem **semiOf_semiOf** {τ : O' →+* O''} {υ : O →+* O''} {G'' : PDivGroup p O''} {A'' : Over (Sp O'')} [GrpObj A''] {ψ : A''.left ⟶ A'.left} (t'' : G''.TorsionTower A'') {s : SemiHom σ G' G} {s' : SemiHom τ G'' G'} {s'' : SemiHom υ G'' G} (h : ∀ n, spMap (s.app n).toRingHom ≫ (t.y n).left = (t'.y n).left ≫ φ) (h' : ∀ n, spMap (s'.app n).toRingHom ≫ (t'.y n).left = (t''.y n).left ≫ ψ) (h'' : ∀ n, spMap (s''.app n).toRingHom ≫ (t.y n).left = (t''.y n).left ≫ ψ ≫ φ) (n : ℕ) (a : G.A n) : s''.app n a = s'.app n (s.app n a)

theorem **_root_.GlobalTheory.PDiv.grpMor_of_hom** (f : A ⟶ A₁) [IsMonHom f] : GrpMor (dp0 O) (𝟙 _) A₁ (dp0 O) (𝟙 _) A (RingHom.id O) (spMap (RingHom.id O)) f.left

theorem **map_y** (f : A ⟶ A₁) [IsMonHom f] (n : ℕ) : spOver ((t.map t₁ f).app n : H.A n →ₐ[O] G.A n) ≫ t₁.y n = t.y n ≫ f

theorem **hom_ext_y** {g g' : G ⟶ H} (h : ∀ n, spOver (g.app n : H.A n →ₐ[O] G.A n) ≫ t₁.y n = spOver (g'.app n : H.A n →ₐ[O] G.A n) ≫ t₁.y n) : g = g'

theorem **eq_map_iff** (f : A ⟶ A₁) [IsMonHom f] (g : G ⟶ H) : g = t.map t₁ f ↔ ∀ n, spOver (g.app n : H.A n →ₐ[O] G.A n) ≫ t₁.y n = t.y n ≫ f

theorem **map_congr** {f g : A ⟶ A₁} [IsMonHom f] [IsMonHom g] (e : f = g) : t.map t₁ f = t.map t₁ g

@[simp] theorem **map_id** : t.map t (𝟙 A) = 𝟙 G

theorem **map_comp** (f : A ⟶ A₁) (g : A₁ ⟶ A₂) [IsMonHom f] [IsMonHom g] : t.map t₂ (f ≫ g) = t.map t₁ f ≫ t₁.map t₂ g

theorem **map_mul** (f g : A ⟶ A₁) [IsMonHom f] [IsMonHom g] [IsMonHom (f * g)] : t.map t₁ (f * g) = t.map t₁ f + t.map t₁ g

theorem **map_one** [IsMonHom (1 : A ⟶ A₁)] : t.map t₁ 1 = 0

theorem **existsUnique_iso** (t' : H.TorsionTower A) : ∃! e : G ≅ H, ∀ n, spOver (e.hom.app n : H.A n →ₐ[O] G.A n) ≫ t'.y n = t.y n

theorem **semi_map** {σ : O →+* O'} {G' H' : PDivGroup p O'} {A' A₁' : Over (Sp O')} [GrpObj A'] [GrpObj A₁'] (t' : G'.TorsionTower A') (t₁' : H'.TorsionTower A₁') {φ : A'.left ⟶ A.left} {φ₁ : A₁'.left ⟶ A₁.left} {s : SemiHom σ G' G} {s₁ : SemiHom σ H' H} (h : ∀ n, spMap (s.app n).toRingHom ≫ (t.y n).left = (t'.y n).left ≫ φ) (h₁ : ∀ n, spMap (s₁.app n).toRingHom ≫ (t₁.y n).left = (t₁'.y n).left ≫ φ₁) (f : A ⟶ A₁) (f' : A' ⟶ A₁') [IsMonHom f] [IsMonHom f'] (e : φ ≫ f.left = f'.left ≫ φ₁) (n : ℕ) (a : H.A n) : s.app n ((t.map t₁ f).app n a) = (t'.map t₁' f').app n (s₁.app n a)

theorem **map_pow_id** (k : ℕ) [IsMonHom (𝟙 A ^ k)] : t.map t (𝟙 A ^ k) = (k : ℤ_[p]) • 𝟙 G

theorem **ringAct_apply** [IsCommMonObj A] {Λ : Type} [Ring Λ] (a : RingAct A Λ) (x : Λ) [IsMonHom (a.act x)] : t.ringAct a x = t.map t (a.act x)

theorem **act_apply** {Λ : Type} [CommRing Λ] (a : OAct Λ A) (x : Λ) [IsMonHom (a.act x)] : t.act a x = t.map t (a.act x)

@[simp] theorem **_root_.GlobalTheory.PDivGroup.Action.padicExtend_tmul** {Λ : Type} [Ring Λ] (ρ : Action Λ G) (c : ℤ_[p]) (x : Λ) : ρ.padicExtend (c ⊗ₜ x) = c • ρ x

theorem **baseChangeY_left** (n : ℕ) : (t.baseChangeY h n).left ≫ φ = spMap (toBaseChange O O' (G.A n)).toRingHom ≫ (t.y n).left

theorem **ptsDown_baseChangeY** (n : ℕ) (B : Type) [CommRing B] [Algebra O B] [Algebra O' B] (hB : ∀ r, algebraMap O B r = algebraMap O' B (algebraMap O O' r)) (x : WithConv (O' ⊗[O] G.A n →ₐ[O'] B)) : ptsDown (grpMor_of_isGrpPullback h) B hB (spOver x.ofConv ≫ t.baseChangeY h n) = t.pts n B ((toBaseChange O O' (G.A n)).pts B hB x)

theorem **toBase_y** (n : ℕ) : spMap ((G.toBase O').app n).toRingHom ≫ (t.y n).left = ((t.baseChange h).y n).left ≫ φ

theorem **semiHom_baseChange** : (t.baseChange h).semiHom t (grpMor_of_isGrpPullback h) = G.toBase O'

theorem **baseChangeIso_hom** {G' : PDivGroup p O'} (t' : G'.TorsionTower A') : (t.baseChangeIso h t').hom = (t'.semiHom t (grpMor_of_isGrpPullback h)).lift

@[simp] theorem **_root_.GlobalTheory.PDiv.fst_epsTo** (B : Type) [CommRing B] [Algebra O B] (z : DualNumber O) : (epsTo O B z).fst = algebraMap O B z.fst

@[simp] theorem **_root_.GlobalTheory.PDiv.snd_epsTo** (B : Type) [CommRing B] [Algebra O B] (z : DualNumber O) : (epsTo O B z).snd = algebraMap O B z.snd

theorem **_root_.GlobalTheory.PDiv.epsScale_eq** (c : O) : GrpTangent.epsScale c = spOver (TrivSqZeroExt.map (c • (LinearMap.id : O →ₗ[O] O)))

theorem **_root_.GlobalTheory.PDiv.epsRedPow_pow** {v : epsR O ⟶ A} (hv : IsTangent v) (n : ℕ) : epsRedPow p n v ^ p ^ n = 1

theorem **tanPt_y** (hv : IsTangent v) (n : ℕ) : spOver (t.tanPt hv n) ≫ t.y n = epsRedPow p n v

theorem **fst_tanPt** (hv : IsTangent v) (n : ℕ) (a : G.A n) : (t.tanPt hv n a).fst = algebraMap O _ (counit a)

@[simp] theorem **tanDer_d** (hv : IsTangent v) (n : ℕ) (a : G.A n) : t.tanDer hv n (dUnit O a) = (t.tanPt hv n a).snd

theorem **lieCorr_iff** (hv : IsTangent v) (x : G.Lie) : t.LieCorr x v ↔ ∀ (n : ℕ) (w : G.omega), Ideal.Quotient.mk _ (x w) = t.tanDer hv n (w.1 n)

theorem **lieCorr_zero** : t.LieCorr 0 1

theorem **LieCorr.add** (h : t.LieCorr x v) (h' : t.LieCorr x' v') : t.LieCorr (x + x') (v ∗ v')

theorem **LieCorr.smul** (h : t.LieCorr x v) (c : O) : t.LieCorr (c • x) (GrpTangent.epsScale c ≫ v)

theorem **LieCorr.map** [Fact p.Prime] (h : t.LieCorr x v) (f : A ⟶ A₁) [IsMonHom f] : t₁.LieCorr (lieMap G H (t.map t₁ f) x) (v ≫ f)

theorem **LieCorr.semi_mod** {σ : O →+* O'} {G' : PDivGroup p O'} {A' : Over (Sp O')} [GrpObj A'] {t' : G'.TorsionTower A'} {φ : A'.left ⟶ A.left} {s : SemiHom σ G' G} (hs : ∀ n, spMap (s.app n).toRingHom ≫ (t.y n).left = (t'.y n).left ≫ φ) {x' : G'.Lie} {v' : epsR O' ⟶ A'} (h : t.LieCorr x v) (h' : t'.LieCorr x' v') (e : v'.left ≫ φ = spMap (ReciprocityTheory.ChartRestriction.G.Chart.dualMapR σ) ≫ v.left) (w : G.omega) (n : ℕ) : (Ideal.Quotient.mk _ (x' (s.omega w)) : ModPow p O' n) = Ideal.Quotient.mk _ (σ (x w))

theorem **LieCorr.semi** {σ : O →+* O'} [IsHausdorff (Ideal.span {(p : O')}) O'] {G' : PDivGroup p O'} {A' : Over (Sp O')} [GrpObj A'] {t' : G'.TorsionTower A'} {φ : A'.left ⟶ A.left} {s : SemiHom σ G' G} (hs : ∀ n, spMap (s.app n).toRingHom ≫ (t.y n).left = (t'.y n).left ≫ φ) {x' : G'.Lie} {v' : epsR O' ⟶ A'} (h : t.LieCorr x v) (h' : t'.LieCorr x' v') (e : v'.left ≫ φ = spMap (ReciprocityTheory.ChartRestriction.G.Chart.dualMapR σ) ≫ v.left) (w : G.omega) : x' (s.omega w) = σ (x w)

theorem **LieCorr.unique_left** [IsHausdorff (Ideal.span {(p : O)}) O] (h : t.LieCorr x v) (h' : t.LieCorr x' v) : x = x'

theorem **tanDer_succ** (hv : IsTangent v) (n : ℕ) (w : Cot O (G.A (n + 1))) (c : O) (hc : Ideal.Quotient.mk _ c = t.tanDer hv (n + 1) w) : Ideal.Quotient.mk _ c = t.tanDer hv n (cotMap (G.res n) w)

theorem **existsUnique_lieCorr** [IsAdicComplete (Ideal.span {(p : O)}) O] (hv : IsTangent v) : ∃! x : G.Lie, t.LieCorr x v

theorem **lieCorr_muLieGen** [IsAdicComplete (Ideal.span {(p : O)}) O] : (muTower p O).LieCorr (muLieGen p O) (Diag.lieVec ((Int.castAddHom O).comp (AddEquiv.ulift : ULift.{0} ℤ ≃+ ℤ).toAddMonoidHom))

Development.AbelianVarietyTheory.Definitions.Divisors.TorusMemOfSepAtBTate

theorem **fromBase_id** {V : Type} [CommRing V] [IsDomain V] (Q : DegPre V) (τ₀ : Gens Q) (h : ∀ a b, (τ₀ a b).IsGen) : FromBase Q τ₀ (RingHom.id V)

theorem **gm_field** {W : Type} [Field W] {Z : Scheme.{0}} (r : Z ⟶ Sp W) (z : Z) (U : (Sp W).Opens) (h : r z ∈ U) (a : Γ(Sp W, U)ˣ) : ∃ x : Wˣ, gm r z U a = gm r z ⊤ (unitOn x)

theorem **gm_spMap** {W V : Type} [CommRing W] [CommRing V] (f : W →+* V) (z : Sp V) (x : Wˣ) : gm (spMap f) z ⊤ (unitOn x) = gm (𝟙 (Sp V)) z ⊤ (unitOn (Units.map f.toMonoidHom x))

theorem **gm_unitOn_inj** {V : Type} [CommRing V] [IsDomain V] (z : Sp V) {x y : Vˣ} (h : gm (𝟙 (Sp V)) z ⊤ (unitOn x) = gm (𝟙 (Sp V)) z ⊤ (unitOn y)) : x = y

theorem **mem_range_of_fromBase** {W : Type} [Field W] (ιb : W →+* InhTate.R) (u : ℤ → ℤ → InhTate.Rˣ) (h : FromBase tatePre (tGens u) ιb) (a b : ℤ) : (u a b : InhTate.R) ∈ ιb.range

theorem **fromBase_tate_rat** (x : ℤ → ℤ → ℚˣ) : FromBase tatePre (tGens fun a b => Units.map (algebraMap ℚ InhTate.R).toMonoidHom (x a b)) (algebraMap ℚ InhTate.R)

theorem **not_fromBase_tate** : ¬ FromBase tatePre (tGens fun _ _ => onePlus) (algebraMap ℚ InhTate.R)

theorem **lbIso_two** {X Y Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) (φ : Y ⟶ X) (r : Z ⟶ Y) : LBIso (L.pull (r ≫ φ)) ((L.pull φ).pull r)

theorem **sepAtB_iff_of_range** {V : Type} [CommRing V] [IsDomain V] (Q : DegPre V) (τ₀ : Gens Q) {m N : ℕ} (s : Fin m ⊕ Fin N → V) {K : Type} [Field K] (ev : V →+* K) (k₀ : Subfield K) (h : ∀ v, ev v ∈ k₀) : SepAtB Q τ₀ s ev k₀ ↔ ∀ i, ev (s (.inl i)) = 0

theorem **sepAtB_tate** (τ₀ : Gens tatePre) : SepAtB tatePre τ₀ (Sum.elim ![PowerSeries.X] ![onePlus.val]) (PowerSeries.constantCoeff (R

theorem **torus_mem_of_sepAtB_tate** (x : ℤ → ℤ → ℚˣ) {m N : ℕ} (s : Fin m ⊕ Fin N → InhTate.R) {K : Type} [Field K] [CharZero K] (ev : InhTate.R →+* K) (k₀ : Subfield K) (h : SepAtB tatePre (tGens fun a b => Units.map (algebraMap ℚ InhTate.R).toMonoidHom (x a b)) s ev k₀) (i : Fin N) : ev (s (.inr i)) ∈ k₀

theorem **sepB_two_of_sep** {Xo : Moduli Dw l} {c : Coeff l} {φ : InvFun (𝒪 E)} {S : Scheme.{0}} {j : Xo.XF c ⟶ S} (h : Sep Xo c φ j) (V : Type) [CommRing V] [IsDomain V] [IsLocalRing V] (C : MonoFam Xo c φ V) (hV : GlobalTheory.Mum.Base V) (ξ : Sp V ⟶ S) (hξ : C.y ≫ j = jK V ≫ ξ) (τ₀ : Gens C.P.D.toPre) (K : Type) [Field K] [IsAlgClosed K] (ev : V →+* K) (k₀ : Subfield K) (hk : IsAlgClosed k₀) (hc : ∀ p, (spMap ev ≫ ξ) p = ξ (closedPoint V)) (hd : DefOver k₀ (spMap ev ≫ ξ)) : SepAtB C.P.D.toPre τ₀ (N

theorem **uPoinStX_of_uStB** (h : UStB) : GlobalTheory.UPoinc.UPoinStX

theorem **stalkMap_gm** {X Y Z : Scheme.{0}} (r : Z ⟶ Y) (f : Y ⟶ X) (z : Z) (U : X.Opens) (h : f (r z) ∈ U) (a : Γ(X, U)ˣ) : Units.map (r.stalkMap z).hom.toMonoidHom (gm f (r z) U a) = gm (r ≫ f) z U a

theorem **mem_of_fromBase** {W : Type} [CommRing W] (ιb : W →+* InhTate.R) (u : ℤ → ℤ → InhTate.Rˣ) (h : FromBase tatePre (tGens u) ιb) {K : Type} [Field K] (ev : InhTate.R →+* K) (k₀ : Subfield K) (hk : ∀ a, ev (ιb a) ∈ k₀) (a b : ℤ) : ev (u a b) ∈ k₀

theorem **exists_coords** {Xo : Moduli Dw l} {c : Coeff l} {φ : InvFun (𝒪 E)} {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] (C : MonoFam Xo c φ V) : ∃ (N : ℕ) (s : Fin C.m ⊕ Fin N → V) (ε : Module.Basis (Fin C.m ⊕ Fin N) ℤ (balForms H.bar C.P.D.toPre.X)) (τ₀ : Gens C.P.D.toPre), (∀ i, s (.inl i) = C.s i) ∧ (∀ i, (ε (.inl i) : LinearMap.BilinForm ℤ _) = C.δ i) ∧ (∀ k, IsUnit (s (.inr k))) ∧ TauIn C.P.D.toPre τ₀ (fun k => s (finSumFinEquiv.symm k)) (fun k => (l.M' : ℤ) • (ε (finSumFinEquiv.symm k) : LinearMap.BilinForm ℤ C.P.D.toPre.X))

theorem **not_sepB_point** {Xo : Moduli Dw l} {c : Coeff l} {φ : InvFun (𝒪 E)} {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] (C : MonoFam Xo c φ V) (hV : GlobalTheory.Mum.Base V) (hm : 1 ≤ C.m) : ¬ SepB Xo c φ (S

theorem **exists_not_sepB** : ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : IsCMField E) (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (φ : InvFun (𝒪 E)), IsPolFan H.bar (min Dw.p Dw.q) φ ∧ NoSelfInt H.bar (min Dw.p Dw.q) φ ∧ ¬ SepB Xo c φ (S

theorem **sepAtB_closed** {Xo : Moduli Dw l} {c : Coeff l} {φ : InvFun (𝒪 E)} {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] (C : MonoFam Xo c φ V) {N : ℕ} (s : Fin C.m ⊕ Fin N → V) (τ₀ : Gens C.P.D.toPre) (hs : ∀ i, s (.inl i) = C.s i) : SepAtB C.P.D.toPre τ₀ s (residue V) ⊤

Development.AbelianVarietyTheory.Definitions.Divisors.TorusNoLeClassNoRankOne

theorem **isUA_mul** {g h : Matrix (Fin n) (Fin n) (RA F E)} (hg : IsUA c G g) (hh : IsUA c G h) : IsUA c G (g * h)

theorem **isUA_inv** {g : Matrix (Fin n) (Fin n) (RA F E)} (hg : IsUA c G g) : IsUA c G g⁻¹

theorem **IsKM_maps** {k : Matrix (Fin n) (Fin n) (RA F E)} (hk : IsKM c G L M k) {x : Fin n → RA F E} (hx : x ∈ hat F L) : k *ᵥ x ∈ hat F L

theorem **isKM_mul** {k h : Matrix (Fin n) (Fin n) (RA F E)} (hk : IsKM c G L M k) (hh : IsKM c G L M h) : IsKM c G L M (k * h)

theorem **isKM_inv** {k : Matrix (Fin n) (Fin n) (RA F E)} (hk : IsKM c G L M k) : IsKM c G L M k⁻¹

theorem **isKM_neg_one** : IsKM c G L 2 (-1)

theorem **ratM_mul** (a b : Matrix (Fin n) (Fin n) E) : ratM F (a * b) = ratM F a * ratM F b

theorem **ratM_one** : ratM F (1 : Matrix (Fin n) (Fin n) E) = 1

theorem **uEquiv_refl** (g : Matrix (Fin n) (Fin n) (RA F E)) : UEquiv c G L M g g

theorem **uEquiv_trans** {a b d : Matrix (Fin n) (Fin n) (RA F E)} (h : UEquiv c G L M a b) (h' : UEquiv c G L M b d) : UEquiv c G L M a d

theorem **one_le_classNo** : 1 ≤ classNo c G L M

theorem **one_le_torusNo** : 1 ≤ torusNo c G L M

theorem **isTA_det** (hG : G.det ≠ 0) {g : Matrix (Fin n) (Fin n) (RA F E)} (hg : IsUA c G g) : IsTA c g.det

theorem **isKM_of_dvd** {M' : ℕ} (h : M ∣ M') {k : Matrix (Fin n) (Fin n) (RA F E)} (hk : IsKM c G L M' k) : IsKM c G L M k

theorem **one_tmul_injective** : Function.Injective fun x : E => (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] x

theorem **ratM_det** (γ : Matrix (Fin n) (Fin n) E) : (ratM F γ).det = (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ γ.det

theorem **isUA_of_isTA** (G : Matrix (Fin 1) (Fin 1) E) {t : RA F E} (ht : IsTA c t) : IsUA c G !![t]

theorem **torusNo_le_classNo_rank_one** (G : Matrix (Fin 1) (Fin 1) E) (hG : G.det ≠ 0) (L : Set (Fin 1 → E)) : torusNo c G L M ≤ classNo c G L M

theorem **classNo_rank_zero** (G : Matrix (Fin 0) (Fin 0) E) (L : Set (Fin 0 → E)) : classNo c G L M = 1

theorem **uEquiv_symm** {a b : Matrix (Fin n) (Fin n) (RA F E)} (h : UEquiv c G L M a b) : UEquiv c G L M b a

Development.AbelianVarietyTheory.Definitions.Divisors.TupleReadBaseTupleEqFamily

theorem **IndefiniteRowSideFullHecke.face_eq** (P : IndefiniteRowSideFullHecke ex) (Dcf Spf) (Del h0 : ℕ) (Csz : ℝ) : P.face Dcf Spf Del h0 Csz = faceT P.toRowSideFamily Dcf Spf (P.zsc Del h0 Csz)

theorem **nonempty_lift** (p : PtAt ex Csz) : Nonempty (Fin D.R.m)

theorem **phEff_one** (i : Fin D.R.m) : phEff p.R i = 1

theorem **tupleRead_baseTuple_eq_family** : p.tupT D.what₂.mu0 = p.P.A3.A.μ0

theorem **tw_iff** (w : D.what₂.places.Pl) : D.plEq w ∈ p.R.twPl ↔ D.what₂.muAt w ≠ D.what₂.mu0

theorem **realised_ofReal** (h : RealRecPins Qp Qr) : (ofReal P Del h0 hDel hh0 Dcf Spf gamDvd B Qp Qr).Realised

theorem **ofReal_S** (b : Bool) : (ofReal P Del h0 hDel hh0 Dcf Spf gamDvd B Qp Qr).S b = (B.Q.Rl.toCx3 B.Q.F).data B.Q.F.cr b

theorem **ofReal_N** (C CH : ℝ) : (ofReal P Del h0 hDel hh0 Dcf Spf gamDvd B Qp Qr).N C CH = B.Q.Rl.num B.Q.F C CH

theorem **exportRow_ofReal** : (ofReal P Del h0 hDel hh0 Dcf Spf gamDvd B Qp Qr).ExportRow ↔ Formalism.IndefR.NX.NMjT B.Q B.Q.cG B.Q.Sx B.Q.nexp ex.nch

theorem **rows_ofReal** (hE : B.Q.Rl.toRealDat.ExAt) (hI : InB B) (hB : RowsB B) (hP : PrimRows3 B.Q.Rl B.Q.F) (hs : P.A3.fam3.TypesSep3) (hk : P.A3.fam3.Ok3 1) (hf : B.Q.FramesEx) (hd : DivRows3 D P.A3 P.P0 P.ψ0 P.Ps P.ψs P.Zs P.S0 P.S1 ex.p0 P.C P.NuC P.I) (hψ : PsiPPTie D P.P0 P.ψ0 P.Ps P.ψs P.I.eG) (h0' : D.what₂.D0 = (P.A3.A.fam.R P.A3.A.fam.μ0).dZ (P.A3.A.fam.lvl P.P0 P.ψ0) (P.Zs.Z P.A3.A.fam.μ0)) (hN : Formalism.IndefR.NX.NMjT B.Q B.Q.cG B.Q.Sx B.Q.nexp ex.nch) (hG : Formalism.BasicNotions.GoodFTAll D) (hw : ∀ w, (B.Q.Rl.toCx3 B.Q.F).W1 B.Q.F.cr (B.Q.Rl.num B.Q.F 0 0) w → w ∈ B.Q.Rl.twPl) (ht : (ofReal P Del h0 hDel hh0 Dcf Spf gamDvd B Qp Qr).Ties) : (ofReal P Del h0 hDel hh0 Dcf Spf gamDvd B Qp Qr).Rows

theorem **induction** {Q : PtAt ex Csz → Prop} (hQ : ∀ (Rc : RecT ex) (S : Formalism.L2.SG.SpectralTiesReal.STiesR Rc) (B : RealTB Rc p.P.toRowSideFamily p.Dcf p.Spf (p.P.zsc p.Del p.h0 Csz) S) (Qp : RealPins B.Q) (Qr : RealRec B.Q), RealRecPins Qp Qr → Q (ofReal p.P p.Del p.h0 p.hDel p.hh0 p.Dcf p.Spf p.gamDvd B Qp Qr)) : Q p

theorem **tiedRec** : TiedRec ex

theorem **sp** : HEq (p.Spf p.P.Zs) p.P.Sp

theorem **col** : ∃ sEE : (p.R.toCx3 p.F).EE →ₗ*[ℂ] (p.F.fc (p.R.toCx3 p.F).zs).Sp.HH, HEq ((p.R.toCx3 p.F).colOf p.F.cr ((p.R.toCx3 p.F).colDef (fun u t => (zetaEps (p.R.toCx3 p.F) p.F.cr sEE u t)⁻¹) sEE)) p.P.C

theorem **trSem** : (∃ tr, p.R.TrISem p.F tr ∧ TrISepWith p.P.A3 tr) → p.R.TrISem p.F p.P.A3.trI ∧ p.P.A3.fam3.TrISep

theorem **exists_frame** : ∃ Rc : RecT ex, p.R.C = Rc.T.n.cf ∧ HEq p.Tp Rc.T.n.tp ∧ p.R.ψ = Rc.T.theta.W.ψ

theorem **valLL0** : Formalism.L2.Host.ValLL0 D p.R.C

theorem **psiChoice** : Formalism.L2.Host.PsiChoiceCompatible D p.R.C ex.nch p.Tp

theorem **Ties.small** {p : PtAt ex Csz} (h : p.Ties) : GlobalTheory.FinalJoin2.FieldSmall D L

theorem **Ties.eSer** {p : PtAt ex Csz} (h : p.Ties) : GlobalTheory.GLq.ESerTie D

theorem **Ties.mem_xd_iff** {p : PtAt ex Csz} (h : p.Ties) (i : D.ntw.Tw) (j : Fin D.R.m) (b : Bool) (a : D.nexp.Dat i ((j : ℕ) + 1) b) : a ∈ ex.xd.xd0 i _ b ↔ ∀ u : D.nexp.U, D.nexp.actG i _ b u (D.nexp.th i _ b a) = ((D.nexp.psiU i b u : (FractionRing (D.ntw.O i))ˣ) : FractionRing (D.ntw.O i)) • D.nexp.th i _ b a

theorem **EvPt.of_forall** {eps : ℝ} {Fm : FamP eps} {Q : ∀ (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x), Pt Fm x hx i → Prop} (h : ∀ x hx i p, p.Rows → Q x hx i p) : EvPt Fm Q

theorem **EvPt.and** {eps : ℝ} {Fm : FamP eps} {Q Q' : ∀ (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x), Pt Fm x hx i → Prop} (h : EvPt Fm Q) (h' : EvPt Fm Q') : EvPt Fm fun x hx i p => Q x hx i p ∧ Q' x hx i p

theorem **not_evPt_false** {eps : ℝ} {Fm : FamP eps} (h : ∀ x0, ∃ (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x) (p : Pt Fm x hx i), x0 ≤ x ∧ p.Rows) : ¬ EvPt Fm fun _ _ _ _ => False

theorem **Fun0.toric** (h : Fun0) : ToricFunctorialityCondition

Development.AbelianVarietyTheory.Definitions.Divisors.TwOrbMEqTwOrbUOfNormal

theorem **avg_const_fun** {H : Type} [Group H] (c : ℂ) : GlobalTheory.BCT.avg (fun _ : H => c) = c

theorem **orbSum_of_twCent_eq_top** (θ : G →* G) (K : Subgroup G) (δ : G) (h : GlobalTheory.BCT.twCent θ δ = ⊤) : ∃ x : G, ∀ (w : G → ℝ) (f : G → ℂ), GlobalTheory.LKLoc.orbSum θ K δ w f = (w x : ℂ) * f δ

theorem **locPart_of_normal** (T K : Subgroup G) [hK : K.Normal] (x : G) : GlobalTheory.LKLoc.locPart T K x = T ⊓ K

private lemma **volRat_self'** (V : Subgroup G) : GlobalTheory.LKLoc.volRat V V = 1

theorem **twOrbM_eq_twOrbU_of_normal** (θ : G →* G) (K : Subgroup G) [K.Normal] (δ : G) (f : G → ℂ) : GlobalTheory.LKLoc.twOrbM θ K δ f = GlobalTheory.LKLoc.twOrbU θ K δ (GlobalTheory.BCT.twCent θ δ ⊓ K) f

theorem **sigKQ_apply_inclK** (r : ℕ) (a : K) : sigKQ p K r (inclK p K r a) = inclK p K r a

theorem **sigKQ_apply_inclQ** (r : ℕ) (b : Qr p r) : sigKQ p K r (inclQ p K r b) = inclQ p K r (frobQ p r b)

theorem **frobQ_degree_one** : frobQ p 1 = AlgHom.id ℚ_[p] (Qr p 1)

theorem **sigRes_degree_one** (n : ℕ) : sigRes p K n 1 = MonoidHom.id _

private lemma **gl_fin_one_mul_comm** {R : Type} [CommRing R] (a b : GL (Fin 1) R) : a * b = b * a

theorem **normal_of_gl_one** {R : Type} [CommRing R] (H : Subgroup (GL (Fin 1) R)) : H.Normal

theorem **TOHT_gl_one** (m r : ℕ) (Kw : Fin m → IntermediateField ℚ_[p] (PadicAlgCl p)) (δ : GHT p m 1 r Kw) (φ : GHT p m 1 r Kw → ℂ) : TOHT p m 1 r Kw δ φ = TOHTU p m 1 r Kw δ (twCentHT p m 1 r Kw δ ⊓ KHT p m 1 r Kw) φ

Development.AbelianVarietyTheory.Definitions.Divisors.TwistAtEach

theorem **TwistSum.congr** {X₁ : GSet} {z₁ : X₁.M.Z} {act : I → (X₁.ρ.P.A ⟶ X₁.ρ.P.A)} {x₀ : X₁.S.Gam} {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] {ev : X₁.M.Z.presheaf.stalk z₁ →+* V} {X : Type} [AddCommGroup X] {t : X → V} {δ : I → X → X} {m : GMem X₁ z₁ act V ev X t δ} {C : Type} {lab : C → X₁.S.Gam} {T : X → C → V} {sg : ℤˣ} {c : C} {τ : X₁.S.Gam} {e e' : ℕ} (he : X₁.S.ζ ^ e = X₁.S.ζ ^ e') (h : TwistSum X₁ z₁ act x₀ m lab T sg c τ e) : TwistSum X₁ z₁ act x₀ m lab T sg c τ e'

Development.AbelianVarietyTheory.Definitions.Divisors.TwistedPlaceChoiceLiesOverPair

theorem **lambdaGrid_apply** (G : GridData Lc sc) (l : ℕ) (j a : Fin D.R.m) : (lamOf G).lam0 l j a = Lc.C.mu0 j * Lc.C.delAt l / G.ch.chiH (a.1 + 1)

theorem **lamOf_hasInfQ** (G : GridData Lc sc) (a : Fin D.R.m) : ((lamOf G).chi a).HasInfQ (embPsi D.host) fun g => ((lamOf G).c a g : ℚ)

theorem **lamOf_teich** (G : GridData Lc sc) : (lamOf G).Teich

theorem **FieldHas.map** {G : GridData Lc sc} {K K' : Type} [Field K] [Field K'] {e' : K' →+* ℂ} (j : K →+* K') (h : FieldHas G (e'.comp j)) : FieldHas G e'

theorem **exists_fieldHas** (G : GridData Lc sc) : ∃ M : IntermediateField ℚ ℂ, FiniteDimensional ℚ M ∧ FieldHas G M.val.toRingHom

theorem **TwistedPlaceChoiceLiesOverPair.inW1** {D : D7} {c : NChoice D} (h : TwistedPlaceChoiceLiesOverPair D c) {w : D.what₂.places.Pl} (hw : GlobalTheory.IndefiniteRowSideFullHecke.GoodPairSystems.Twisted D w) : D.ntw.NonzeroIndexPlace (c.twI w)

theorem **GridFit.fieldLL** {p : PtAt ex Csz} {CF : ℝ} {G : Grid p CF} (h : GridFit p G) : FieldHas G D.what₂.embLL

theorem **GridFit.of_twistAt** {p : PtAt ex Csz} {CF : ℝ} {G : Grid p CF} {xd v0 rep} (h : GlobalTheory.EX.TwistAt L ex.nch xd v0 rep) (h1 : ValLL0 D p.R.C) (h2 : FieldHas G (embLL0 D)) (h3 : Formalism.L2.Host.PsiChoiceCompatible D p.R.C ex.nch p.Tp) (h4 : GlobalTheory.LamTie.IsLamTieFull D.ntw (lamOf G)) : GridFit p G

theorem **evPt_and** {eps : ℝ} {Fm : FamP eps} {P Q : ∀ (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x), Pt Fm x hx i → Prop} (hP : EvPt Fm P) (hQ : EvPt Fm Q) : EvPt Fm fun x hx i p => P x hx i p ∧ Q x hx i p

theorem **evPt_imp** {eps : ℝ} {Fm : FamP eps} {P Q : ∀ (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x), Pt Fm x hx i → Prop} (hP : EvPt Fm P) (h : ∀ x hx i p, p.Rows → P x hx i p → Q x hx i p) : EvPt Fm Q

theorem **Lf.mono** {Comparison₂ Tie' : TieT} {Q Q' : PropT} (h : Lf Comparison₂ Q) (h' : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Y : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Y cX} {Csz : ℝ} (p : PtAt ex Csz) (CF : ℝ) (d : FunIn p CF) (C CH : ℝ), p.Rows → Tie' p CF d → Comparison₂ p CF d ∧ (Q p CF d C CH → Q' p CF d C CH)) : Lf Tie' Q'

theorem **Lf.and** {Comparison₂ : TieT} {Q Q' : PropT} (h : Lf Comparison₂ Q) (h' : Lf Comparison₂ Q') : Lf Comparison₂ fun p CF d C CH => Q p CF d C CH ∧ Q' p CF d C CH

theorem **Lf.of_noTie** {Q : PropT} (h : Lf trivialTieCondition Q) (Comparison₂ : TieT) : Lf Comparison₂ Q

theorem **fun0Fit_of_leaves** (Fld : FldT) (Comparison₂ : TieT) (hB : Bundle) (hS : Scale) (hE : Ex Fld Comparison₂) (h1 : MM Comparison₂) (h2 : OneMap Comparison₂) (h3 : Ker Comparison₂) (h4 : FP Comparison₂) (h5 : IPa Comparison₂) (h6 : IPb) (h7 : OneSq Comparison₂) (h8 : Den Comparison₂) (h9 : Int Comparison₂) (h10 : Arch Comparison₂) (h11 : Thb Comparison₂) (h12 : Wt Comparison₂) (h13 : DesH) (h14 : RankGapBound) (h15 : RamifiedSumBound) (h16 : DesPol) : Fun0Fit Fld

Development.AbelianVarietyTheory.Definitions.Divisors.TwistedPlacesOfCondition

theorem **Hypotheses₂.eisenstein** (h : Hypotheses₂) : GlobalTheory.Published.SiegelEisensteinContinuation

theorem **serClause_of_rec** (Rc : Formalism.L2.RecT ex) : SerClause D

theorem **TiesA.of_real** {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (Q : RealT Rc P Dcf Spf Zscf S Ks) (hP : RealPins Q) (ok : P.A3.fam3.Ok3 1) (hex : Q.Rl.toRealDat.ExAt) (par : Nonempty (ParameterChoice' D L)) (cc : D.host.ccK = ⊤) (ty : GlobalTheory.IndefiniteRowSideFullHecke.CommutingLevelOperators.GoodTypes Q.Rl.toRealDat) (junk : GlobalTheory.Hk3.InertJunk P.A3) (w1 : ∀ w, (Q.Rl.toCx3 Q.F).W1 Q.F.cr (Q.Rl.num Q.F 0 0) w → w ∈ Q.Rl.twPl) (nV : 8 * D.R.f * D.R.r + 4 ≤ D.R.nV) : TiesA Q.Rl Q.F

theorem **TiesG.of_real** (B : RealTB Rc P Dcf Spf Zscf S) (hB : RowsB B) (hU : B.Q.Rl.toRealDat.LUnif) (hC : GlobalTheory.TwistField.CoverField B.T B.Q.Rl.toRealDat) : TiesG B.T B.Q.Rl B.κ B.v₀ B.O

theorem **TiesA.twistedPlaces_of_condition** (h : TiesA R F) (C CH : ℝ) : ∀ w, (R.toCx3 F).W1 F.cr (R.num F C CH) w → w ∈ R.twPl

theorem **TiesA.char3** (h : TiesA R F) : R.Char3 F

theorem **TiesG.phaseTie** {T : CoreJ D} [Algebra T.Q.c.F D.what₂.LL] {R : RealDat3 D A3 T.Q.c.F} {κ : ∀ i : Fin D.R.m, WeightFamilies.Wt D ((i : ℕ) + 1)} {v₀ : ∀ (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), (R.toRealDat.MW i σ).Vec} {O : ∀ i, OMDat T.toCore R.toRealDat i Unit (fun _ => A3.A.μ0)} (h : TiesG T R κ v₀ O) (i : Fin D.R.m) : PhaseTie T R.toRealDat κ v₀ O (fun i => cj (twAt D.what₂.embLL A3.A.v0) (R.phs i)) i

Development.AbelianVarietyTheory.Definitions.Divisors.TwoClassFrameHasLevelSetting

theorem **embZ_ring** : Function.Injective (embZ l) ∧ ∃ L : IntermediateField ℚ_[l] (PadicAlgCl l), FiniteDimensional ℚ_[l] L ∧ Set.range (embZ l) = {x | x ∈ L ∧ ‖x‖ ≤ 1}

theorem **twoFrame_ok** (N : ℕ) : (twoFrame l N).Ok where

theorem **act_one** (N : ℕ) (mu : Unit → ℕ → ℕ) : act l N mu 1 = 1

theorem **act_mul** (N : ℕ) (mu : Unit → ℕ → ℕ) (g h : GL (Fin N) ℤ_[l]) : act l N mu (g * h) = act l N mu g * act l N mu h

theorem **twoFrame_isEig** (N : ℕ) (mu : Unit → ℕ → ℕ) (v : NablaT (PadicAlgCl l) N mu) (hv : v ≠ 0) (e : ℤˣ) : (twoFrame l N).IsEig mu fun _ => ((e : ℤ) : PadicAlgCl l)

theorem **twoFrame_isEig_neg** (N : ℕ) : (twoFrame l N).IsEig (fun _ _ => 0) fun _ => -1

theorem **exists_over** {ι : Type} {N : ℕ} (F : Frame l ι N) (hF : F.Ok) (θ : F.I → PadicAlgCl l) (hθ : ∀ i, ‖θ i‖ ≤ 1) : ∃ m : Ideal F.T, m.IsPrime ∧ F.Over θ m

theorem **hasLevelSetting** {ι : Type} [Fintype ι] [DecidableEq ι] {N : ℕ} (F : Frame l ι N) (hF : F.Ok) (lam : ι → ℕ → ℕ) (θ : F.I → PadicAlgCl l) (hθ : ∀ i, ‖θ i‖ ≤ 1) : HasLevelSetting l F lam θ

theorem **twoClassFrame_hasLevelSetting** (N : ℕ) (lam : Unit → ℕ → ℕ) : HasLevelSetting l (twoFrame l N) lam fun _ => -1

theorem **scFrame_ok** (ev : ℕ → ℤ) : (scFrame l ev).Ok

theorem **cc_tprod** (x : Unit → Formalism.GradedWeylDivisibility.dualWeyl (PadicAlgCl l) 3 3 (res 3 (fun _ : ℕ => 0))) : cc l (PiTensorProduct.tprod _ x) = MvPolynomial.coeff 0 (x ()).1

theorem **cc_one** : cc l (oneVec l 3) = 1

theorem **one_mem_int** : (1 : MvPolynomial (Fin 3 × Fin 3) ℤ_[l]) ∈ Formalism.GradedWeylDivisibility.dualWeyl ℤ_[l] 3 3 (res 3 (fun _ : ℕ => 0))

theorem **th_occurs** (ev : ℕ → ℤ) : (scReads l ev).Occurs ⊤ (th l ev)

theorem **hasBlock_yes** : (scReads l evYes).HasBlock (th l evYes)

theorem **root_of_hasBlock** (ev : ℕ → ℤ) (θ : Fin 3 → PadicAlgCl l) (h : (scReads l ev).HasBlock θ) : 1 - θ 0 + 2 * θ 1 - 8 * θ 2 = 0

theorem **not_hasBlock_no** : ¬ (scReads l evNo).HasBlock (th l evNo)

theorem **not_hasBlock_zero** (ev : ℕ → ℤ) : ¬ (scReads l ev).HasBlock 0

theorem **allPlaces_reads** (E Ω : Type) [Field E] [NumberField E] [Field Ω] [Algebra E Ω] [IsAlgClosure E Ω] : (allPlaces l E Ω).Reads 1 (fun _ _ => 1) fun _ => Polynomial.X - Polynomial.C 1

Development.AbelianVarietyTheory.Definitions.Divisors.UApproxAtZero

theorem **val_eq_valInt** (t : UCTrait X₁ p z₁) (a : X₁.M.Z.presheaf.stalk z₁) : t.val a = (valInt t a : PadicAlgCl p)

theorem **norm_val_le** (t : UCTrait X₁ p z₁) (a : X₁.M.Z.presheaf.stalk z₁) : ‖t.val a‖ ≤ 1

theorem **sc_pos** (t : UCTrait X₁ p z₁) : 0 < t.sc

theorem **uApproxAt_zero** (φ : GlobalTheory.LocM.InvFun (𝒪 E)) : UApproxAt X₁ φ 0

Development.AbelianVarietyTheory.Definitions.Divisors.UnifOfSpaceEq

theorem **Laws.evalWD** (_ : Laws D j Xo c U dict) : EvalWD (K

theorem **Laws.unTest** (_ : Laws D j Xo c U dict) : ∀ σ, GoodEmb c σ → ∀ k x, FromTest (K

theorem **Laws.xiAbs** (h : Laws D j Xo c U dict) : ∀ σ, GoodEmb c σ → ∀ k (τ : D.host.E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = xiR D

theorem **unif_of_space_eq** {D : D7} {r : ℕ} {C : Type} {X : C → Data D.host.E r} (V W : HSp D) (h : V = W) (dict : GA D V → Σ k, Frame (X k).Ψ r) (M : ℕ) (Kf : Subgroup (GA D V)) (hK : IsLevelK V M Kf) (hd : ∃ th hr, (∀ k τ, Formalism.MainNotions.StandardLattice.IsInfEmb τ.1 (th k τ)) ∧ DictIsUnif V dict Kf th hr) (hu : ∀ g τ k, ‖dmin (X (dict g).1).Gm (dict g).2 τ k‖ = 1) : ∃ Kf th hr, IsLevelK W M Kf ∧ (∀ k τ, Formalism.MainNotions.StandardLattice.IsInfEmb τ.1 (th k τ)) ∧ DictIsUnif W (fun g => dict (h ▸ g)) Kf th hr ∧ ∀ (g : GA D W) τ k, ‖dmin (X (dict (h ▸ g)).1).Gm (dict (h ▸ g)).2 τ k‖ = 1

theorem **laws_core** (D : D7) (T : CoreJ D) (j : ℕ) (h : LawsJ T) (hj : InR D j) : Laws D j (T.Q.Xo j) T.Q.c (T.U j) (coreDict T.toCore j) where

theorem **one_le_multBd** {ε : ℝ} (hε : 0 < ε) : 1 ≤ multBd E r ε

Development.AbelianVarietyTheory.Definitions.Divisors.UnipVal

theorem **weyl_val** : ((weyl c ν : GL (ν ⊕ ν) E) : Matrix (ν ⊕ ν) (ν ⊕ ν) E) = stdJ ν E

theorem **unip_val** (b : Matrix ν ν E) (hb : (b.map c)ᵀ = b) : ((unip c b hb : GL (ν ⊕ ν) E) : Matrix (ν ⊕ ν) (ν ⊕ ν) E) = fromBlocks 1 b 0 1

theorem **mat_inv** (g : UG c (stdJ ν E)) : mat c g⁻¹ = -(stdJ ν E * ((mat c g).map c)ᵀ * stdJ ν E)

theorem **mem_sieg** {g : UG c (stdJ ν E)} : g ∈ sieg c ν ↔ (mat c g).toBlocks₂₁ = 0

Development.AbelianVarietyTheory.Definitions.Divisors.UnitaryWeightExponentVectorSignatureWeightMuZero

theorem **le_pull_chart** {w : Sp S ⟶ Z'} {j : PUnit → L.ι} (hz : ⊤ ≤ (w ≫ f) ⁻¹ᵁ L.U (j PUnit.unit)) : ⊤ ≤ w ⁻¹ᵁ (L.pull f).U j

theorem **ev_pull_g** (j j' : PUnit → L.ι) (w : Sp S ⟶ Z') (hw : ⊤ ≤ w ⁻¹ᵁ ((L.pull f).U j ⊓ (L.pull f).U j')) (hh : ⊤ ≤ (w ≫ f) ⁻¹ᵁ (L.U (j PUnit.unit) ⊓ L.U (j' PUnit.unit))) : ev w _ hw ((L.pull f).g j j').val = ev (w ≫ f) _ hh (L.g (j PUnit.unit) (j' PUnit.unit)).val

theorem **exists_tri_of_le** {v v' : Fin m → N} (h : ∀ i, initSpan R v' i ≤ initSpan R v i) : ∃ b : Matrix (Fin m) (Fin m) R, b.BlockTriangular id ∧ ∀ i, v' i = ∑ j, b j i • v j

theorem **exists_tri_of_eq** {v v' : Fin m → N} (hv : LinearIndependent R v) (h : ∀ i, initSpan R v' i = initSpan R v i) : ∃ b : Matrix (Fin m) (Fin m) R, b.BlockTriangular id ∧ (∀ i, IsUnit (b i i)) ∧ ∀ i, v' i = ∑ j, b j i • v j

theorem **linearIndependent_of_tri** {v v' : Fin m → N} (hv : LinearIndependent R v) {b : Matrix (Fin m) (Fin m) R} (hb : b.BlockTriangular id) (hu : ∀ i, IsUnit (b i i)) (e : ∀ i, v' i = ∑ j, b j i • v j) : LinearIndependent R v'

theorem **frBor_baseChange** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} {z z' : FrPt Xo c T} (h : FrBor t y y') (hz : FrMap φ y z) (hz' : FrMap φ y' z') : FrBor (fun i => Units.map (φ : S →* T) (t i)) z z'

theorem **exists_frMap** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y : FrPt Xo c S) : ∃ z, FrMap φ y z

theorem **frMap_refl** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : FrMap (AlgHom.id c.F S) y y

theorem **frMap_unique** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] {φ : S →ₐ[c.F] T} {y : FrPt Xo c S} {z z' : FrPt Xo c T} (h : FrMap φ y z) (h' : FrMap φ y z') : z = z'

theorem **spMap_algId** {S : Type} [CommRing S] [Algebra c.F S] {Z : Scheme.{0}} (z : Sp S ⟶ Z) : spMap (AlgHom.id c.F S).toRingHom ≫ z = z

theorem **spMap_algComp** {S S' S'' : Type} [CommRing S] [Algebra c.F S] [CommRing S'] [Algebra c.F S'] [CommRing S''] [Algebra c.F S''] (ψ : S →ₐ[c.F] S') (χ : S' →ₐ[c.F] S'') {Z : Scheme.{0}} (z : Sp S ⟶ Z) : spMap (χ.comp ψ).toRingHom ≫ z = spMap χ.toRingHom ≫ spMap ψ.toRingHom ≫ z

theorem **ChartFr.over** {k : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {Q : FrPt Xo c S} (h : F.ChartFr k z hz Q) : frX Q ≫ Y.j = z ≫ π

theorem **ChartFr.frVec_eq** {k : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {Q : FrPt Xo c S} (h : F.ChartFr k z hz Q) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (hQ : frX Q ≫ Y.j = x) (σ : E →+* ℂ) : frVec Y Q hQ σ = F.vecAt k z hz hx σ

theorem **ChartFr.unique** {k : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {Q Q' : FrPt Xo c S} (h : F.ChartFr k z hz Q) (h' : F.ChartFr k z hz Q') : Q = Q'

theorem **ChartFr.frMap** {k : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {Q : FrPt Xo c S} (h : F.ChartFr k z hz Q) {ψ : S →ₐ[c.F] S'} {Q' : FrPt Xo c S'} (hQ : FrMap ψ Q Q') {z' : Sp S' ⟶ Z} (ez : spMap ψ.toRingHom ≫ z = z') (hz' : ⊤ ≤ z' ⁻¹ᵁ F.U k) : F.ChartFr k z' hz' Q'

theorem **ChartFr.frBor** {k k' : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {hz' : ⊤ ≤ z ⁻¹ᵁ F.U k'} {Q Q' : FrPt Xo c S} (h : F.ChartFr k z hz Q) (h' : F.ChartFr k' z hz' Q') (hh : ⊤ ≤ z ⁻¹ᵁ (F.U k ⊓ F.U k')) : FrBor (F.TAt k k' z hh) Q Q'

theorem **TopEigen.exists_chartFr** (hT : F.TopEigen) {y : Sp S ⟶ Xo.XF c} (hy : y ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) {z : Sp S ⟶ Z} (hx : z ≫ π = y ≫ Y.j) (k : F.J) (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) : ∃ Q : FrPt Xo c S, F.ChartFr k z hz Q

theorem **Lies.frMap** {z : Sp S ⟶ Z} {P : FrPt Xo c S} (h : F.Lies z P) {ψ : S →ₐ[c.F] S'} {P' : FrPt Xo c S'} (hP : FrMap ψ P P') : F.Lies (spMap ψ.toRingHom ≫ z) P'

theorem **Lies.frBor** {z : Sp S ⟶ Z} {P P₁ : FrPt Xo c S} (h : F.Lies z P) {t : Ix Dw → Sˣ} (hB : FrBor t P P₁) : F.Lies z P₁

theorem **ChartFr.lies** {k : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {Q : FrPt Xo c S} (h : F.ChartFr k z hz Q) : F.Lies z Q

theorem **Lies.initSpan_eq** {z : Sp S ⟶ Z} {P : FrPt Xo c S} (h : F.Lies z P) {k : F.J} (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {x : Sp S ⟶ Y.X} (hx : z ≫ π = x) (hP : frX P ≫ Y.j = x) (σ : E →+* ℂ) (i : ℕ) : initSpan S (frVec Y P hP σ) i = F.step z hx σ i

theorem **ChartFr.exists_frBor** {k : F.J} {z : Sp S ⟶ Z} {hz : ⊤ ≤ z ⁻¹ᵁ F.U k} {Q P : FrPt Xo c S} (h : F.ChartFr k z hz Q) (hP : F.Lies z P) : ∃ t, FrBor t Q P

theorem **TopEigen.mem_step_top_iff** (hT : F.TopEigen) {V K : Type} [CommRing V] [IsLocalRing V] [CommRing K] [Algebra c.F K] (φ : V →+* K) (hφ : Function.Injective φ) (z : Sp V ⟶ Z) {x : Sp V ⟶ Y.X} (hx : z ≫ π = x) {y : Sp K ⟶ Xo.XF c} (hy : y ≫ pullback.snd _ _ = spMap (algebraMap c.F K)) (e : spMap φ ≫ x = y ≫ Y.j) (σ : E →+* ℂ) (v : TanM x Y.t.G) : v ∈ F.step z hx σ (Dw.mult σ) ↔ tanPull φ e v ∈ eigLie c (y ≫ Y.j) σ

theorem **TopEigen.diag_unit** (hT : F.TopEigen) {V K : Type} [CommRing V] [IsLocalRing V] [CommRing K] [Algebra c.F K] (φ : V →+* K) (hφ : Function.Injective φ) (z : Sp V ⟶ Z) {k : F.J} (hz : ⊤ ≤ z ⁻¹ᵁ F.U k) {P Q : FrPt Xo c K} (hP : F.Lies (spMap φ ≫ z) P) (e : spMap φ ≫ z ≫ π = frX P ≫ Y.j) (σ : E →+* ℂ) (ub : Fin (Dw.mult σ) → TanM (z ≫ π) Y.t.G) (hub : ∀ i, tanPull φ e (ub i) = frVec Y P rfl σ i) (hsp : ∀ v, tanPull φ e v ∈ eigLie c (frX P ≫ Y.j) σ → ∃ a : Fin (Dw.mult σ) → V, ∑ i, a i • ub i = v) (hQ : F.ChartFr k (spMap φ ≫ z) (le_preimage_comp _ hz) Q) {t : Ix Dw → Kˣ} (ht : FrBor t Q P) : (∀ i, initSpan V ub i = F.step z rfl σ i) ∧ ∀ i, ∃ u : Vˣ, φ u = t ⟨σ, i⟩

theorem **LL_single_one** (F : Frames Y π) (ix : Ix Dw) : F.LL (Pi.single ix 1) = GlobalTheory.CechPic.dual (F.line ix)

theorem **mult_of_mem** (τ : Psi H) : Dw.mult τ.1 = Dw.p

theorem **mult_conj** (τ : Psi H) : Dw.mult (conjEmb τ.1) = Dw.q

theorem **exOf_none** (la : Weight Dw) : exOf Dw la 0 none = 0

theorem **exOf_big** (la : Weight Dw) (τ : Psi H) (i : Fin (Dw.mult τ.1)) : exOf Dw la 0 ⟨τ.1, i⟩ = la τ ⟨i, by have

theorem **exOf_small** (la : Weight Dw) (τ : Psi H) (i : Fin (Dw.mult (conjEmb τ.1))) : exOf Dw la 0 ⟨conjEmb τ.1, i⟩ = -la τ ⟨Dw.p + (Dw.q - 1 - i), by have

theorem **exOf_ext** {la : Weight Dw} {e : Ix Dw → ℤ} (h0 : e none = 0) (h1 : ∀ (τ : Psi H) (i : Fin (Dw.mult τ.1)) hi, la τ ⟨i, hi⟩ = e ⟨τ.1, i⟩) (h2 : ∀ (τ : Psi H) (i : Fin (Dw.mult (conjEmb τ.1))) hi, -la τ ⟨Dw.p + (Dw.q - 1 - i), hi⟩ = e ⟨conjEmb τ.1, i⟩) : exOf Dw la 0 = e

theorem **unitaryWeightExponentVector_signatureWeightMuZero** : exOf Dw (mu0 Dw) 0 = fun ix => ix.elim 0 fun _ => 1

theorem **isLB_Llam** (la : Weight Dw) : (X.Llam la).IsLB

theorem **lies_ptY** (P : FrPt Xo c S) : X.F.Lies (X.ptY P) P

theorem **eq_ptY** {P : FrPt Xo c S} {z : Sp S ⟶ X.YY} (h : X.F.Lies z P) : z = X.ptY P

theorem **pt_inc** (P : FrPt Xo c S) : X.pt P ≫ X.inc = X.ptY P

theorem **pt_snd** (P : FrPt Xo c S) : X.pt P ≫ pullback.snd _ _ = frX P

theorem **ptY_frMap** {ψ : S →ₐ[c.F] S'} {P : FrPt Xo c S} {P' : FrPt Xo c S'} (h : FrMap ψ P P') : spMap ψ.toRingHom ≫ X.ptY P = X.ptY P'

theorem **pt_frMap** {ψ : S →ₐ[c.F] S'} {P : FrPt Xo c S} {P' : FrPt Xo c S'} (h : FrMap ψ P P') : spMap ψ.toRingHom ≫ X.pt P = X.pt P'

theorem **ptY_frBor** {t : Ix Dw → Sˣ} {P P' : FrPt Xo c S} (h : FrBor t P P') : X.ptY P = X.ptY P'

theorem **pt_frBor** {t : Ix Dw → Sˣ} {P P' : FrPt Xo c S} (h : FrBor t P P') : X.pt P = X.pt P'

theorem **exists_chartFr** (w : Sp S ⟶ X.Yo) (hw : w ≫ pullback.snd _ _ ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (k : X.F.J) (hz : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k) : ∃ Q : FrPt Xo c S, X.F.ChartFr k (w ≫ X.inc) hz Q ∧ X.pt Q = w

theorem **points** [IsLocalRing S] : (∀ w : Sp S ⟶ X.Yo, w ≫ pullback.snd _ _ ≫ pullback.snd _ _ = spMap (algebraMap c.F S) ↔ ∃ P, X.pt P = w) ∧ ∀ P P' : FrPt Xo c S, X.pt P = X.pt P' ↔ ∃ t, FrBor t P P'

theorem **le_pull** {w : Sp S ⟶ X.Yo} {k : X.F.J} (hz : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k) : ⊤ ≤ w ⁻¹ᵁ ((X.F.LL e).pull X.inc).U (fun _ => k)

theorem **coef_resO** (s : (X.F.LL e).Sec) (k : X.F.J) (w : Sp S ⟶ X.Yo) (hz : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k) : coef (resO s) k w hz = ev (w ≫ X.inc) (X.F.U k) hz (s.f k)

theorem **coef_change** (s : X.SecO e) (k k' : X.F.J) (w : Sp S ⟶ X.Yo) (hz : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k) (hz' : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k') (hh : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ (X.F.U k ⊓ X.F.U k')) : coef s k w hz = (∏ ix, X.F.TAt k k' (w ≫ X.inc) hh ix ^ (-e ix)).val * coef s k' w hz'

theorem **coef_spMap** (s : X.SecO e) (k : X.F.J) (φ : S →+* S') (w : Sp S ⟶ X.Yo) (hz : ⊤ ≤ (w ≫ X.inc) ⁻¹ᵁ X.F.U k) {w' : Sp S' ⟶ X.Yo} (ew : spMap φ ≫ w = w') (hz' : ⊤ ≤ (w' ≫ X.inc) ⁻¹ᵁ X.F.U k) : coef s k w' hz' = φ (coef s k w hz)

theorem **exists_chartFr_frBor** (P : FrPt Xo c S) (k : X.F.J) (hz : ⊤ ≤ (X.pt P ≫ X.inc) ⁻¹ᵁ X.F.U k) : ∃ (Q : FrPt Xo c S) (t : Ix Dw → Sˣ), X.F.ChartFr k (X.pt P ≫ X.inc) hz Q ∧ FrBor t Q P

theorem **isValAt_of_chart** (s : X.SecO e) {P Q : FrPt Xo c S} {k : X.F.J} {hz : ⊤ ≤ (X.pt P ≫ X.inc) ⁻¹ᵁ X.F.U k} {t : Ix Dw → Sˣ} (hQ : X.F.ChartFr k (X.pt P ≫ X.inc) hz Q) (ht : FrBor t Q P) : IsValAt s P ((∏ ix, t ix ^ e ix).val ∗ coef s k (X.pt P) hz)

theorem **existsUnique_isValAt** (s : X.SecO e) (P : FrPt Xo c S) (k : X.F.J) (hz : ⊤ ≤ (X.pt P ≫ X.inc) ⁻¹ᵁ X.F.U k) : ∃! a, IsValAt s P a

theorem **IsValAt.frMap** {s : X.SecO e} {P : FrPt Xo c S} {a : S} (h : IsValAt s P a) {k : X.F.J} (hz : ⊤ ≤ (X.pt P ≫ X.inc) ⁻¹ᵁ X.F.U k) {ψ : S →ₐ[c.F] S'} {P' : FrPt Xo c S'} (hP : FrMap ψ P P') : IsValAt s P' (ψ a)

theorem **existsUnique_isVal** (s : X.SecO e) (P : FrPt Xo c S) : ∃! a, IsVal s P a

theorem **isVal_val** (s : X.SecO e) (P : FrPt Xo c S) : IsVal s P (val s P)

theorem **IsVal.eq_val** {s : X.SecO e} {P : FrPt Xo c S} {a : S} (h : IsVal s P a) : a = val s P

theorem **val_eq** (s : X.SecO e) {P Q : FrPt Xo c S} {k : X.F.J} {hz : ⊤ ≤ (X.pt P ≫ X.inc) ⁻¹ᵁ X.F.U k} {t : Ix Dw → Sˣ} (hQ : X.F.ChartFr k (X.pt P ≫ X.inc) hz Q) (ht : FrBor t Q P) : val s P = (∏ ix, t ix ^ e ix).val ∗ coef s k (X.pt P) hz

theorem **val_chartFr** (s : X.SecO e) {Q : FrPt Xo c S} {k : X.F.J} {hz : ⊤ ≤ (X.pt Q ≫ X.inc) ⁻¹ᵁ X.F.U k} (hQ : X.F.ChartFr k (X.pt Q ≫ X.inc) hz Q) : val s Q = coef s k (X.pt Q) hz

theorem **val_frMap** (s : X.SecO e) {ψ : S →ₐ[c.F] S'} {P : FrPt Xo c S} {P' : FrPt Xo c S'} (h : FrMap ψ P P') : val s P' = ψ (val s P)

theorem **val_frBor** (s : X.SecO e) {t : Ix Dw → Sˣ} {P P₁ : FrPt Xo c S} (h : FrBor t P P₁) : val s P₁ = (∏ ix, t ix ^ e ix).val ∗ val s P

theorem **toRule_mem_secZ** (s : X.SecO e) : toRule s ∈ secZ Xo c e

theorem **evR_toRule** (s : X.SecO e) (P : FrPt Xo c S) (h : FromTest (K

theorem **evR_toRule_of_isReduced** [_root_.IsReduced S] (s : X.SecO e) (P : FrPt Xo c S) : evR Xo c (toRule s) P = val s P

theorem **val_add** {s s₁ s₂ : X.SecO e} (h : ∀ j, s.f j = s₁.f j + s₂.f j) (P : FrPt Xo c S) : val s P = val s₁ P + val s₂ P

theorem **val_mul** {e₁ e₂ : Ix Dw → ℤ} {s : X.SecO (e₁ + e₂)} {s₁ : X.SecO e₁} {s₂ : X.SecO e₂} (h : ∀ j, s.f j = (show Γ(X.Yo, ((X.F.LL (e₁ + e₂)).pull X.inc).U j) from s₁.f j) ∗ (show Γ(X.Yo, ((X.F.LL (e₁ + e₂)).pull X.inc).U j) from s₂.f j)) (P : FrPt Xo c S) : val s P = val s₁ P ∗ val s₂ P

theorem **eq_of_val_eq** {s s' : X.SecO e} (h : ∀ (S : Type) [CommRing S] [Algebra c.F S] (P : FrPt Xo c S), val s P = val s' P) : s = s'

theorem **Llam_single_big** (τ : Psi H) (i : Fin (Dw.mult τ.1)) (hi : (i : ℕ) < Dw.p + Dw.q) : X.Llam (Pi.single τ (Pi.single ⟨i, hi⟩ (-1))) = X.gr τ.1 i

theorem **Llam_single_small** (τ : Psi H) (i : Fin (Dw.mult (conjEmb τ.1))) (hi : Dw.p + (Dw.q - 1 - i) < Dw.p + Dw.q) : X.Llam (Pi.single τ (Pi.single ⟨_, hi⟩ (-1))) = GlobalTheory.CechPic.dual (X.gr (conjEmb τ.1) i)

Development.AbelianVarietyTheory.Definitions.Divisors.VolMeasureOfAdd

theorem **CompHS.cval_mem_fibSec** (f : H0 p B O) : cval σ f ∈ fibSec p σ B

theorem **divVal_mem** (ω : H0 p (LB.div B L) O) : divVal σ ω ∈ fibSec p σ (tens B (dual L))

theorem **volDens_eq** (a : AlgChart p σ D) {y : CxPts p σ} (hy : y ∈ a.e.source) {i : K.bundle.ι} {j : L.ι} (hi : y ∈ ptsIn p σ (K.bundle.U i)) (hj : y ∈ ptsIn p σ (L.U j)) : volDens K φ ω a y = 2 ^ D ∗ (‖rv σ a.isFrame (K.isFrame i) y ∗ cval σ ω (i, j) y‖ ^ 2 / Real.exp (-φ.φ j y))

theorem **volDens_pos** (hω : NowhereZero ω) (a : AlgChart p σ D) {y : CxPts p σ} (hy : y ∈ a.e.source) (hO : y ∈ ptsIn p σ O) : 0 < volDens K φ ω a y

theorem **volMeasure_compl** : volMeasure K φ ω (ptsIn p σ O)ᶜ = 0

theorem **volMeasure_apply** (a : AlgChart p σ D) {A : Set (CxPts p σ)} (hA : MeasurableSet A) (h : A ⊆ a.e.source) : volMeasure K φ ω A = ∫⁻ v in a.e '' (A ∩ ptsIn p σ O), ENNReal.ofReal (volDens K φ ω a (a.e.symm v))

theorem **volMeasure_of_add** {s : ℝ} (h : ∀ j, ∀ y ∈ ptsIn p σ O, y ∈ ptsIn p σ (L.U j) → φ'.φ j y = φ.φ j y + s) : volMeasure K φ' ω = ENNReal.ofReal (Real.exp s) • volMeasure K φ ω

theorem **volMeasure_congr** (h : ∀ j, ∀ y ∈ ptsIn p σ O, y ∈ ptsIn p σ (L.U j) → φ'.φ j y = φ.φ j y) : volMeasure K φ' ω = volMeasure K φ ω

theorem **volMeasure_pos** (hω : NowhereZero ω) (hφ : ∀ j, ContinuousOn (φ.φ j) (ptsIn p σ O ∩ ptsIn p σ (L.U j))) {y : CxPts p σ} (hy : y ∈ ptsIn p σ O) : 0 < volMeasure K φ ω univ

theorem **goodPt_of_good** (h : GoodEmb c σ) (y : CxPts p σ) : (goodPt y).1 = y.1

theorem **goodPt_of_not** (h : ¬ GoodEmb c σ) (y : CxPts p σ) : (goodPt y).1 = spMap (starRingEnd ℂ) ≫ y.1

theorem **goodPt_mem** {y : CxPts p σ} {W : Y.Opens} (h : y ∈ ptsIn p σ W) : goodPt y ∈ ptsIn p (goodOf c σ) W

theorem **norm_ev_goodPt** {y : CxPts p σ} {W : Y.Opens} (h : y ∈ ptsIn p σ W) (f : Γ(Y, W)) : ‖ev p _ W f (goodPt y)‖ = ‖ev p σ W f y‖

theorem **goodPt_map** {q : T ⟶ Sp c.F} {f : T ⟶ Y} {z : CxPts q σ} {y : CxPts p σ} (h : y.1 = z.1 ≫ f) : (goodPt y).1 = (goodPt z).1 ≫ f

theorem **goodPt_conj** {y : CxPts p σ} {y' : CxPts p ((starRingEnd ℂ).comp σ)} (h : y'.1 = spMap (starRingEnd ℂ) ≫ y.1) : (goodPt y').1 = (goodPt y).1

theorem **isConj_atGood** (φ : ∀ τ, Weight p τ B) (σ : c.F →+* ℂ) : (atGood (φ (goodOf c σ))).IsConj (atGood (φ (goodOf c ((starRingEnd ℂ).comp σ))))

theorem **diagChar_ne_zero** (e : Ix Dw → ℤ) (t : Ix Dw → (CAt σ)ˣ) : diagChar e t ≠ 0

theorem **diagChar_mul** (e : Ix Dw → ℤ) (t t' : Ix Dw → (CAt σ)ˣ) : diagChar e (t ∗ t') = diagChar e t ∗ diagChar e t'

theorem **diagChar_add** (e e' : Ix Dw → ℤ) (t : Ix Dw → (CAt σ)ˣ) : diagChar (e + e') t = diagChar e t ∗ diagChar e' t

theorem **diagChar_one** (e : Ix Dw → ℤ) : diagChar (σ

theorem **diagChar_zero** (t : Ix Dw → (CAt σ)ˣ) : diagChar 0 t = 1

theorem **homFac_pos** (la : AllWt.Weight Dw) (k : U.Comp) (w : (U.X k).dom) : 0 < homFac U x la k w

theorem **homFac_add** (a b : AllWt.Weight Dw) (k : U.Comp) (w : (U.X k).dom) : homFac U x (a + b) k w = homFac U x a k w ∗ homFac U x b k w

theorem **homN_eq_homFac** (la : AllWt.Weight Dw) (f : Rul Xo c) (k : U.Comp) (w : (U.X k).dom) : homN U x (wtOf Dw la 0) f ⟨k, w.1⟩ = homFac U x la k w ∗ ‖toC (evR Xo c f (U.un k w))‖

theorem **genNorm_pos** (la : AllWt.Weight Dw) (k : U.Comp) (w : (U.X k).dom) (t : Ix Dw → (CAt σ)ˣ) : 0 < genNorm U x la k w t

theorem **genNorm_add** (a b : AllWt.Weight Dw) (k : U.Comp) (w : (U.X k).dom) (t : Ix Dw → (CAt σ)ˣ) : genNorm U x (a + b) k w t = genNorm U x a k w t * genNorm U x b k w t

theorem **homFac_frBor** (hσ : GoodEmb c σ) (hU : IsUnf U) (la : AllWt.Weight Dw) {k k' : U.Comp} {w : (U.X k).dom} {w' : (U.X k').dom} {t : Ix Dw → (CAt σ)ˣ} (ht : FrBor t (U.un k w) (U.un k' w')) : homFac U x la k' w' * ‖diagChar (exOf Dw la 0) t‖ = homFac U x la k w

Development.AbelianVarietyTheory.Definitions.Divisors.VolumeBallOne

theorem **volume_ball_one** : volume (Metric.ball (0 : V 1) 1) = ENNReal.ofReal Real.pi

theorem **topInt_pos** [SmoothOfRelativeDimension D p] [SecondCountableTopology (CxPts p σ)] {B : LB Y} (K : CanonLB p D) {φ : Weight p σ B} {f : (tens K.bundle B).ι → CxPts p σ → ℂ} {R : AnRead p σ D} (hφ : φ.IsSmooth R) (hf : f ∈ R.holSec (tens K.bundle B)) {j : (tens K.bundle B).ι} {y : CxPts p σ} (h0 : f j y ≠ 0) : 0 < R.topInt K φ f

theorem **topInt_jac** (N : ℕ) (K : CanonLB (PSpace.str k N) N) (R : AnRead (PSpace.str k N) σ N) : 0 < R.topInt K ((ProjAn.fsO1 k σ N).pow (N + 1)) (secVal _ σ (K.jacSec (coord k N))) ∧ R.topInt K ((ProjAn.fsO1 k σ N).pow (N + 1)) (secVal _ σ (K.jacSec (coord k N))) < ⊤

theorem **exists_std** (y : CxPts (CanonLB.affStr k D) σ) : ∃ c : AlgChart (CanonLB.affStr k D) σ D, y ∈ c.e.source ∧ ∀ z ∈ c.e.source, ∀ a, c.e z a = X a z

theorem **holo_X** (R : AnRead (CanonLB.affStr k D) σ D) (a : Fin D) : R.HoloOn univ (X a)

theorem **not_holo_conj** (R : AnRead (CanonLB.affStr k D) σ D) (a : Fin D) : ¬ R.HoloOn univ fun z => starRingEnd ℂ (X a z)

theorem **smooth_sq** (R : AnRead (CanonLB.affStr k D) σ D) (t : ℝ) : (Weight.ofFun (sq t)).IsSmooth R

theorem **levi_sq** (c : AlgChart (CanonLB.affStr k D) σ D) (t : ℝ) {y} (hy : y ∈ c.e.source) (ξ : Fin D → ℂ) : (ReciprocityTheory.L2DbarEstimates.leviV D (fun v => (sq t (c.e.symm v) : ℂ)) (c.e y) ξ).re = t * ∑ a, Complex.normSq (∑ j, ReciprocityTheory.L2DbarEstimates.wd D j (X a ∘ c.e.symm) (c.e y) * ξ j)

theorem **semiPos_sq** (R : AnRead (CanonLB.affStr k D) σ D) : (Weight.ofFun (sq 1)).SemiPos R

theorem **not_semiPos_sq** (a : Fin D) : ¬ (Weight.ofFun (sq (-1))).SemiPos (AnRead.all (CanonLB.affStr k D) σ D)

Development.AbelianVarietyTheory.Definitions.Divisors.WedgeNrmLeOfWedgeNormLe

theorem **Reframes.val_eq** [IsLocalRing S] {φ : K →ₐ[c.F] S} {y : FrPt Xo c K} {t : Ix Dw → Sˣ} {z : FrPt Xo c S} (h : Reframes φ y t z) {e : Ix Dw → ℤ} {s : RulK Xo c K} (hs : s ∈ secK Xo c K e) : val φ z s = ((∏ i, t i ^ e i : Sˣ) : S) * φ (val (AlgHom.id c.F K) y s)

theorem **Reframes.evR_eq** [IsLocalRing S] {φ : K →ₐ[c.F] S} {y : FrPt Xo c K} {t : Ix Dw → Sˣ} {z : FrPt Xo c S} (h : Reframes φ y t z) {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) : evR Xo c f z = ((∏ i, t i ^ e i : Sˣ) : S) * φ (evR Xo c f y)

theorem **Reframes.liesOn** {φ : K →ₐ[c.F] S} {y : FrPt Xo c K} {t : Ix Dw → Sˣ} {z : FrPt Xo c S} (h : Reframes φ y t z) {Z : Comp Xo c} (hy : LiesOn Z y) : LiesOn Z z

theorem **nrmOf_nonneg** (la : HeightOneSpectrum (𝒪 K)) (w : EReal) : 0 ≤ nrmOf la w

theorem **nrmOf_anti** (la : HeightOneSpectrum (𝒪 K)) {w w' : EReal} (hw : w ≠ ⊥) (h : w ≤ w') : nrmOf la w' ≤ nrmOf la w

theorem **abs_apply** (a : F.T.Fr) : F.abs a = nrmOf v.maximalIdeal (nval F.T.V F.e a)

theorem **abs_φ** (a : K) : F.abs (F.φ a) = v a

theorem **abs_add_le_max** (a b : F.T.Fr) : F.abs (a + b) ≤ max (F.abs a) (F.abs b)

theorem **abs_φ_le_one** (a : 𝒪 K) : F.abs (F.φ a) ≤ 1

theorem **ofReal_abs** (a : F.T.Fr) : ENNReal.ofReal (F.abs a) = normOfOrd (Ideal.absNorm v.maximalIdeal.asIdeal) (nval F.T.V F.e a)

theorem **nrm_nonneg** (f : Rul Xo c) : 0 ≤ F.nrm f

theorem **nrm_mul** (f g : Rul Xo c) : F.nrm (f * g) = F.nrm f * F.nrm g

theorem **fac_pos** (e : Ix Dw → ℤ) : 0 < F.fac e

theorem **fac_add** (e e' : Ix Dw → ℤ) : F.fac (e + e') = F.fac e * F.fac e'

theorem **fac_zero** : F.fac 0 = 1

theorem **nrm_eq** {e : Ix Dw → ℤ} {f : Rul Xo c} (hf : f ∈ secZ Xo c e) : F.nrm f = F.fac e * v (evR Xo c f y)

theorem **exists_mem_valSet** : ∃ (la : HeightOneSpectrum (𝒪 c.F)) (e' : ℕ), OverLa la F.T.V F.io e' ∧ (fun f => nval F.T.V e' (evI F.pt f)) ∈ valSet Xo c (algCls Xo c) la

theorem **nrm_le_one** {C : Submodule c.F (Rul Xo c)} {f : C} (hf : f ∈ supLat C) : F.nrm f ≤ 1

theorem **bddAbove_nrm** (C : Submodule c.F (Rul Xo c)) : BddAbove (Set.range fun w : supLat C => F.nrm w.1)

theorem **nrm_le_latMax** {C : Submodule c.F (Rul Xo c)} {f : C} (hf : f ∈ supLat C) : F.nrm f ≤ F.latMax C

theorem **latMax_le_iff** {C : Submodule c.F (Rul Xo c)} {B : ℝ} : F.latMax C ≤ B ↔ ∀ f ∈ supLat C, F.nrm f.1 ≤ B

theorem **latMax_mem** (C : Submodule c.F (Rul Xo c)) : 0 ≤ F.latMax C ∧ F.latMax C ≤ 1

theorem **latMax_le_of_span** {C : Submodule c.F (Rul Xo c)} {κ : Type} {g : κ → C} (hg : supLat C ≤ Submodule.span (𝒪 c.F) (Set.range g)) {B : ℝ} (hB : 0 ≤ B) (h : ∀ k, F.nrm (g k) ≤ B) : F.latMax C ≤ B

theorem **wedgeNrm_spec** {s : ι → RulK Xo c K} (h : F.wedgeNrm s ≠ ⊤) : (∃ o d, IsWedgeVal (psiSet H) o F.φ F.pt.1 s d) ∧ ∀ o d, IsWedgeVal (psiSet H) o F.φ F.pt.1 s d → F.wedgeNrm s = ENNReal.ofReal (F.abs d)

theorem **wedgeNrm_eq** (h : KSIso (psiSet H) F.pt.1) {e : Ix Dw → ℤ} {s : ι → RulK Xo c K} (hs : ∀ i, s i ∈ secK Xo c K e) {o : ι ≃ TIx Dw (psiSet H)} {d : F.T.Fr} (hd : IsWedgeVal (psiSet H) o F.φ F.pt.1 s d) : F.wedgeNrm s = ENNReal.ofReal (F.abs d)

theorem **finite_support_log** {a : K} (ha : a ≠ 0) : (Function.support fun v : FinitePlace K => Real.log (v a)).Finite

theorem **finsum_log_natCast** {p : ℕ} (hp : p ≠ 0) : ∑ᶠ v : FinitePlace K, Real.log (v (p : K)) = -(Module.finrank ℚ K * Real.log p)

theorem **finite_support_primeWt** {S : Finset ℕ} (hS : 0 ∉ S) (b : ℕ → ℝ) : (Function.support (primeWt (K

theorem **finsum_primeWt** {S : Finset ℕ} (hS : 0 ∉ S) (b : ℕ → ℝ) : (∑ᶠ v, primeWt (K

theorem **exp_primeWt** {S : Finset ℕ} (b : ℕ → ℝ) (v : FinitePlace K) {p : ℕ} (hp : p ∈ S) (h0 : p ≠ 0) (h : ∀ q ∈ S, q ≠ p → v (q : K) = 1) : Real.exp (primeWt S b v) = v (p : K) ^ (-b p)

theorem **primeWt_eq_zero** {S : Finset ℕ} (b : ℕ → ℝ) (v : FinitePlace K) (h : ∀ q ∈ S, v (q : K) = 1) : primeWt S b v = 0

theorem **abs_eq_rpow** {p : ℕ} (hp : p ≠ 0) (h : (p : F.T.V) ∈ IsLocalRing.maximalIdeal F.T.V) {a : F.T.Fr} {w : ℝ} (hw : nval F.T.V (ramIdx F.T p) a = w) : F.abs a = v (p : K) ^ w ∧ v (p : K) < 1

theorem **exists_prime** : ∃ p : ℕ, p.Prime ∧ (p : F.T.V) ∈ IsLocalRing.maximalIdeal F.T.V ∧ ∀ q : ℕ, q.Prime → q ≠ p → v (q : K) = 1

theorem **wedgeNrm_le_of_wedgeNorm_le** {ι : Type} [Fintype ι] {s : ι → RulK Xo c K} {p : ℕ} (hp : 1 < p) (h : (p : F.T.V) ∈ IsLocalRing.maximalIdeal F.T.V) {b : ℝ} (hb : wedgeNorm (psiSet H) p F.φ F.pt s ≤ ENNReal.ofReal ((p : ℝ) ^ b)) : F.wedgeNrm s ≤ ENNReal.ofReal (v (p : K) ^ (-b))

theorem **norm_goodHom** (τ : K →+* ℂ) (a : K) : ‖toC (goodHom c τ a)‖ = ‖τ a‖

theorem **InfFr.nonempty** (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) (y : FrPt Xo c K) (τ : K →+* ℂ) : Nonempty (InfFr U y τ)

theorem **mem_framesOn** {Z : Comp Xo c} (hy : LiesOn Z y) : F.pt ∈ framesOn (U (baseEmb c τ)) Z

theorem **nrm_nonneg** (G : WTZ E) (f : Rul Xo c) : 0 ≤ F.nrm x G f

theorem **fac_pos** (G : WTZ E) : 0 < F.fac x G

theorem **fac_add** {G G' : WTZ E} (hG : Even G.2) (hG' : Even G'.2) : F.fac x (G + G') = F.fac x G * F.fac x G'

theorem **nrm_eq** {G : WTZ E} {f : Rul Xo c} (hf : f ∈ secZ Xo c (exZ Dw G)) : F.nrm x G f = F.fac x G * ‖τ (evR Xo c f y)‖

theorem **bergCusp_le_iff** {G : WTZ E} {σ : c.F →+* ℂ} {z : Σ k, FD.Frame ((U (goodOf c σ)).X k).Ψ r} {B : ℝ} (hB : 0 ≤ B) : bergCusp U x V dict Z G σ z ≤ ENNReal.ofReal B ↔ ∀ f ∈ cuspOn U Z G, homN (U (goodOf c σ)) x G f z ^ 2 ≤ B * petOn U x V dict Z G σ f ^ 2

theorem **bergCusp_le_bergOn** {G : WTZ E} {σ : c.F →+* ℂ} (hd : ∀ g, (dict (goodOf c σ) g).2 ∈ ((U (goodOf c σ)).X (dict (goodOf c σ) g).1).dom) (z : FPt (U (goodOf c σ)).X) : bergCusp U x V dict Z G σ ⟨z.1, z.2.1⟩ ≤ bergOn (U (goodOf c σ)) (V (goodOf c σ)) (dict (goodOf c σ)) Z G z

theorem **latExt_le_secK** {K : Type} [Field K] [Algebra c.F K] {la : AllWt.Weight Dw} {s : RulK Xo c K} (hs : s ∈ latExt U Z K la) : s ∈ secK Xo c K (exOf Dw la 0)

theorem **FinFr.abs_val_le_one** {K : Type} [Field K] [NumberField K] [Algebra c.F K] {y : FrPt Xo c K} {v : FinitePlace K} (F : FinFr y v) {la : AllWt.Weight Dw} {s : RulK Xo c K} (hs : s ∈ latExt U Z K la) : F.abs (val F.φ F.pt.1 s) ≤ 1

theorem **WtPar.kap_add_tauWt** : (kap Dw + dimFlag Dw • P.u₁) + P.u₂ + P.tauWt = (dimFlag Dw + 3) • P.u₁

theorem **JacFinAt.mono** {X X' : ℝ} (h : JacFinAt U Z P y X) (e : X ≤ X') : JacFinAt U Z P y X'

theorem **TauPtAt.mono** {X X' : ℝ} (h : TauPtAt U x V dict Z P y X) (e : X ≤ X') : TauPtAt U x V dict Z P y X'

theorem **JacFinAt.of_primes** (S : Finset ℕ) (hS : ∀ p ∈ S, p.Prime) (b : ℕ → ℝ) (h : ∀ s : Fin (dimFlag Dw) → RulK Xo c K, (∀ i, s i ∈ latExt U Z K P.u₁) → (∀ i, val (AlgHom.id c.F K) y (s i) = 0) → ∀ v (F : FinFr y v) (p : ℕ), p.Prime → (p : F.T.V) ∈ IsLocalRing.maximalIdeal F.T.V → wedgeNorm (psiSet H) p F.φ F.pt s ≤ ENNReal.ofReal ((p : ℝ) ^ (if p ∈ S then b p else 0))) : JacFinAt U Z P y (∑ p ∈ S, b p * Real.log p)

theorem **TauPtAt.latMax_pos** {X : ℝ} (h : TauPtAt U x V dict Z P y X) {v : FinitePlace K} (F : FinFr y v) : 0 < F.latMax (Cusp Z U P.u₁)

theorem **JacFin.mono** {X X' : ℝ} (h : JacFin U Z P X) (e : X ≤ X') : JacFin U Z P X'

theorem **TauPt.mono** {X X' : ℝ} (h : TauPt U x V dict Z P X) (e : X ≤ X') : TauPt U x V dict Z P X'

theorem **SupBdWt.mono** {la : AllWt.Weight Dw} {β β' : ℝ} (h : SupBdWt U x V dict Z la β) (e : β ≤ β') : SupBdWt U x V dict Z la β'

theorem **SupBd.mono** {β β' : ℝ} (h : SupBd U x V dict Z P β) (e : β ≤ β') : SupBd U x V dict Z P β'

theorem **supBd_iff_wt_zero** {β : ℝ} : SupBd U x V dict Z P β ↔ SupBdWt U x V dict Z (P.wt 0) β

theorem **supBdWt_iff** {la : AllWt.Weight Dw} {β : ℝ} (hβ : 0 ≤ β) : SupBdWt U x V dict Z la β ↔ ∀ σ, ∀ z ∈ framesOn (U (goodOf c σ)) Z, bergCusp U x V dict Z (wtOf Dw la 0) σ z ≤ ENNReal.ofReal β

theorem **SupBdWt.of_bergOn** {la : AllWt.Weight Dw} {β : ℝ} (hβ : 0 ≤ β) (hd : ∀ σ, GoodEmb c σ → ∀ g, (dict σ g).2 ∈ ((U σ).X (dict σ g).1).dom) (h : ∀ σ, GoodEmb c σ → ∀ z : FPt (U σ).X, ((z.1, z.2.1) : Σ k, FD.Frame ((U σ).X k).Ψ r) ∈ framesOn (U σ) Z → bergOn (U σ) (V σ) (dict σ) Z (wtOf Dw la 0) z ≤ ENNReal.ofReal β) : SupBdWt U x V dict Z la β

theorem **SupBdWt.nrm_sq_le** {la : AllWt.Weight Dw} {β : ℝ} (h : SupBdWt U x V dict Z la β) {K : Type} [Field K] [Algebra c.F K] {y : FrPt Xo c K} (hy : LiesOn Z y) {ρ : K →+* ℂ} (F : InfFr U y ρ) {f : Rul Xo c} (hf : f ∈ Cusp Z U la) : F.nrm x (wtOf Dw la 0) f ^ 2 ≤ β * petOn U x V dict Z (wtOf Dw la 0) (ρ.comp (algebraMap c.F K)) f ^ 2

Development.AbelianVarietyTheory.Definitions.Divisors.WeilRangeOfAll

theorem **range_cases** (n m : ℕ) : 2 * n < m ∨ (n < m ∧ m ≤ 2 * n) ∨ m ≤ n

theorem **weilRange_of_all** (h : GlobalTheory.Published.SiegelWeilAnisotropicWitt) : WeilRange

theorem **midRange_of_all** (h : GlobalTheory.Published.SiegelWeilAnisotropicWitt) : MidRange

theorem **lowRange_of_all** (h : GlobalTheory.Published.SiegelWeilAnisotropicWitt) : LowRange

theorem **all_of_ranges** (h₁ : WeilRange) (h₂ : MidRange) (h₃ : LowRange) : GlobalTheory.Published.SiegelWeilAnisotropicWitt

theorem **all_iff_ranges** : GlobalTheory.Published.SiegelWeilAnisotropicWitt ↔ WeilRange ∧ MidRange ∧ LowRange

Development.AbelianVarietyTheory.Definitions.Divisors.WhiskerLeftEqLift

theorem **isOn_top** {L : LB X} (hL : L.IsLB) : IsOn ⊤ L

theorem **fnBij_id** : FnBij (𝟙 X)

theorem **AddCls.pull** {T T' : Over S} {c : Pic (A ⊗ T).left} (h : AddCls A c) (u : T' ⟶ T) : AddCls A (CechPic.pull (A ◁ u).left c)

theorem **addF_map_val** {T T' : Over S} (u : T' ⟶ T) (c : (addF A).obj (Opposite.op T)) : ((addF A).map u.op c).1 = CechPic.pull (A ◁ u).left c.1

theorem **famCls_surj** {T : Over S} (c : (addF A).obj (Opposite.op T)) : ∃ (N : LB (A ⊗ T).left) (h : Fam A N), famCls h = c

theorem **whiskerLeft_eq_lift** {T T' : Over S} (u : T' ⟶ T) : A ◁ u = lift (fst A T') (snd A T' ≫ u)

theorem **same_iff_map** {T U Z : Over S} {N : LB (A ⊗ T).left} {M : LB (A ⊗ U).left} (hN : Fam A N) (hM : Fam A M) (y : Z ⟶ T) (u : Z ⟶ U) : Same A N M y u ↔ (addF A).map y.op (famCls hN) = (addF A).map u.op (famCls hM)

Development.AbelianVarietyTheory.Definitions.Divisors.XiMulNu

theorem **_root_.GlobalTheory.Sandwich.xi_mul_nu** (ε : PreTilt O p) : _root_.GlobalTheory.Sandwich.xi ε * _root_.GlobalTheory.Sandwich.nu ε = _root_.GlobalTheory.Sandwich.mu ε

theorem **_root_.GlobalTheory.Sandwich.phiInv_mu** (ε : PreTilt O p) : _root_.GlobalTheory.Sandwich.phiInv p O (_root_.GlobalTheory.Sandwich.mu ε) = _root_.GlobalTheory.Sandwich.nu ε

Development.AbelianVarietyTheory.Definitions.Divisors.ZeroMemSolSet

theorem **_root_.GlobalTheory.HCA.apOp_zero_one** (u : _root_.ZetaZeros.D.EllReg.E d → ℂ) (x : _root_.ZetaZeros.D.EllReg.E d) : _root_.GlobalTheory.HCA.apOp 0 (fun _ _ _ => 1) u x = u x

theorem **_root_.GlobalTheory.HCA.zero_mem_solSet** {ι : Type} (P : _root_.ZetaZeros.D.EllReg.Sys d ι) (U : Set (_root_.ZetaZeros.D.EllReg.E d)) : (0 : U →ᵇ ℂ) ∈ _root_.GlobalTheory.HCA.solSet P U

Development.AbelianVarietyTheory.Definitions.Divisors.ZetaActRefl

theorem **wB_eq** (N : ℕ) (x : Fin 2 → ZMod N) (τ : ℍ) : wB N x τ = (EisensteinSeries.G2 τ + (tauPair τ).weierstrassP (tors N τ x)) / (2 * Real.pi * Complex.I) ^ 2

theorem **gact_one** (N : ℕ) (α : GL (Fin 2) (ZMod N)) (γ : SL(2, ℤ)) : gact N 1 α γ = α * redGL N γ

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.AdelLeftInvAndContinuous

theorem **dict_invariant** : ∀ (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (r : ℕ) (C : Ctx Xo c σ V r), C.OK → ∀ dict, C.IsDict dict → ∀ Kf : Subgroup (GA D V), IsLevelK V l.M Kf → (∀ γ ∈ GQ V, ∀ g, ∃ γ' ∈ (C.M.X (dict g).1).Γ, dict (γ * g) = ⟨(dict g).1, lact γ' (dict g).2⟩) ∧ (∀ k ∈ Kf, ∀ g, ∃ γ' ∈ (C.M.X (dict g).1).Γ, dict (g * k) = ⟨(dict g).1, lact γ' (dict g).2⟩) ∧ (∀ g, (dict g).2 ∈ (C.M.X (dict g).1).dom) ∧ (∀ g τ i, ‖dmin (C.M.X (dict g).1).Gm (dict g).2 τ i‖ = 1) ∧ ∀ g k γ, C.Mark g k γ → ∃ δ ∈ (C.M.X k).Γ, dict g = ⟨k, lact δ (C.fr (γ * g) k)⟩

theorem **frame_rmul_lact** {E : Type} [Field E] [NumberField E] [IsCMField E] {Ψ : Finset (E →+* ℂ)} {r : ℕ} (γ : GL (Fin r) E) (x b : Frame Ψ r) : rmul (lact γ x) b = lact γ (rmul x b)

theorem **dict_eq_of_mark** (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (r : ℕ) (C : Ctx Xo c σ V r) (hC : C.OK) (dict : GA D V → Σ k, Frame (C.M.X k).Ψ r) (hd : C.IsDict dict) (g : GA D V) (k : Fin C.F.n) (γ : GA D V) (hm : C.Mark g k γ) : ∃ δ ∈ (C.M.X k).Γ, dict g = ⟨k, lact δ (C.fr (γ * g) k)⟩

theorem **dict_real_equivariant_onto** : ∀ (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (r : ℕ) (C : Ctx Xo c σ V r), C.OK → ∀ dict, C.IsDict dict → (∀ g, ∀ h ∈ GinfC V, ∃ γ' ∈ (C.M.X (dict g).1).Γ, dict (g * h) = ⟨(dict g).1, lact γ' (rmul (dict g).2 (archMat V C.th C.hr (fun k => C.fr 1 k) h (dict g).1))⟩) ∧ ∀ k, ∀ x ∈ (C.M.X k).dom, ∃ g, ∃ γ ∈ (C.M.X k).Γ, ∃ b ∈ Bor (C.M.X k).Ψ r, dict g = ⟨k, rmul (lact γ x) b⟩

theorem **dictIsUnif_of_isDict** : ∀ (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (r : ℕ) (C : Ctx Xo c σ V r), C.OK → ∀ dict, C.IsDict dict → ∀ Kf : Subgroup (GA D V), IsLevelK V l.M Kf → DictIsUnif V dict Kf C.th C.hr

theorem **adel_leftInv_and_continuous** : ∀ (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (r : ℕ) (C : Ctx Xo c σ V r), C.OK → ∀ dict, C.IsDict dict → ∀ mu : GA D V → ℂ, (∀ g g', mu (g * g') = mu g * mu g') → (∀ γ ∈ GQ V, mu γ = 1) → Continuous mu → ∀ (la : ∀ k, Wt (C.M.X k).Ψ r) (P : FDSec C.M.X), IsHolCusp C.M.X la P → (∀ γ ∈ GQ V, ∀ g, adel V dict mu P (γ * g) = adel V dict mu P g) ∧ Continuous (adel V dict mu P)

theorem **exists_pkgLaws** : ∀ (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (r : ℕ) (C : Ctx Xo c σ V r), C.OK → ∀ Kf : Subgroup (GA D V), IsLevelK V l.M Kf → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → (∀ a, χ (idF D a) = 1) → ∃ un dict mu, C.IsDict dict ∧ GlobalTheory.FrameUnif.FramesOK C.e C.F C.M un ∧ PkgLaws V C.hr C.ι Kf χ (C.unf un) dict mu C.x₀

theorem **not_iotaOK_zero** {D : D7} {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} (Dw : Datum H) (V : HSp D) (hV : 0 < V.rank) : ¬ IotaOK Dw V 0

theorem **isRat_mul** {D : D7} (V : HSp D) {a b : GA D V} {a' b' : GL (Fin V.rank) D.host.E} (ha : IsRat V a a') (hb : IsRat V b b') : IsRat V (a * b) (a' * b')

theorem **exists_ctx_dict_pkg** (D : D7) (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) : ∃ (l : Level (hostOfRecord D)) (Xo : Moduli (datumW D j hj) l) (c : Coeff l) (σ : c.F →+* ℂ) (C : Ctx Xo c σ (D.sp.W j) (D.sp.W j).rank) (Kf : Subgroup (GA D (D.sp.W j))), C.OK ∧ 3 ≤ (D.sp.W j).rank ∧ 0 < C.F.n ∧ IsLevelK (D.sp.W j) l.M Kf ∧ ∃ un dict mu, C.IsDict dict ∧ PkgLaws (D.sp.W j) C.hr C.ι Kf 1 (C.unf un) dict mu C.x₀

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.CmHostRootComplexEqPmSqrt

theorem **exists_curve_model** (H : Host E) : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (P : Out.PolAV (Sp (𝒪 M))) (C : Curve H) (φ : C.P.A.left ⟶ P.A.left), IsGrpPullback C.P.A P.A (spMap ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M))) φ

theorem **cmHost_rootComplex_eq_pm_sqrt** (H : Host E) : H.rt0 = Complex.I * Real.sqrt H.d ∨ H.rt0 = -(Complex.I * Real.sqrt H.d)

theorem **cmHost_rootZero_im_ne_zero** (H : Host E) : H.rt0.im ≠ 0

theorem **map_wE** (τ : E →+* ℂ) (hτ : τ (H.rt : E) = H.rt0) (s : ℤ × ℤ) : τ (wE H s : E) = wC H s

theorem **wC_injective** (H : Host E) : Function.Injective (wC H)

theorem **pt_eq_iff** (C : Curve H) (z z' : ℂ) : C.pt z = C.pt z' ↔ ∃ s, z - z' = wC H s

theorem **exists_wC_mul** (H : Host E) (s a : ℤ × ℤ) : ∃ b, wC H s * wC H a = wC H b

theorem **spMap_refl** : spMap ((RingEquiv.refl ℂ : ℂ ≃+* ℂ) : ℂ →+* ℂ) = 𝟙 _

theorem **actsBy_refl** (C : Curve H) (n : ℕ) : C.ActsBy (RingEquiv.refl ℂ) (𝟙 _) n (1, 0)

theorem **fixesCurve_refl** (H : Host E) (n : ℕ) : FixesCurve H (RingEquiv.refl ℂ) n

theorem **curveOver_top** (H : Host E) (n : ℕ) : CurveOver H ⊤ n

theorem **actsBy_unique** {C : Curve H} {τ : ℂ ≃+* ℂ} {f : C.P.A.left ⟶ C.P.A.left} {n : ℕ} (hn : n ≠ 0) {s s' : ℤ × ℤ} (h : C.ActsBy τ f n s) (h' : C.ActsBy τ f n s') : (n : ℤ) ∣ s.1 - s'.1 ∧ (n : ℤ) ∣ s.2 - s'.2

theorem **not_actsBy_neg** (C : Curve H) : ¬ C.ActsBy (RingEquiv.refl ℂ) (𝟙 _) 3 (-1, 0)

theorem **actsBy_of_dvd** {C : Curve H} {τ : ℂ ≃+* ℂ} {f : C.P.A.left ⟶ C.P.A.left} {m n : ℕ} (hn : n ≠ 0) (hmn : m ∣ n) {s : ℤ × ℤ} (h : C.ActsBy τ f n s) : C.ActsBy τ f m s

theorem **pt_wC** (C : Curve H) (a : ℤ × ℤ) : C.pt (wC H a) = η[C.P.A].left

theorem **actsBy_one** {C : Curve H} {τ : ℂ ≃+* ℂ} {f : C.P.A.left ⟶ C.P.A.left} (h : IsGrpPullback C.P.A C.P.A (spMap (τ : ℂ →+* ℂ)) f) (s : ℤ × ℤ) : C.ActsBy τ f 1 s

theorem **fixesCurve_zero_iff** (τ : ℂ ≃+* ℂ) : FixesCurve H τ 0 ↔ FixesCurve H τ 1

theorem **exists_curveOver_one** (H : Host E) : ∃ F : IntermediateField ℚ ℂ, FiniteDimensional ℚ F ∧ CurveOver H F 1

theorem **sign_lat** {ρ : ℂ} (hρ : ρ.im ≠ 0) {a : D.La} (ha : a ∈ lat hP ρ) (h0 : a ≠ 0) : 0 < ρ.im * D.EF (H.rt • a) a

theorem **signs** : (∀ a ∈ lat hP H.rt0, a ≠ 0 → 0 < H.rt0.im * D.EF (H.rt • a) a) ∧ ∀ a ∈ lat hP (-H.rt0), a ≠ 0 → H.rt0.im * D.EF (H.rt • a) a < 0

theorem **splitLat_of_q_zero** (D : Datum H) (hq : D.q = 0) : SplitLat D ∧ ∀ (g : ℕ) (ty : Fin g → (E →+* ℂ)) (P : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty P), IsTypeFn H D.p D.q ty → IsSplit hP

theorem **exists_splitLat** : ∃ D : Datum CM.LatModel.host, D.p = 1 ∧ D.q = 0 ∧ SplitLat D

theorem **neg_of_p_zero** (D : Datum H) (hp : D.p = 0) (a : D.La) (h0 : a ≠ 0) : H.rt0.im * D.EF (H.rt • a) a < 0

theorem **splitLat_sum** (D D' : Datum H) (hq : D.q = 0) (hp : D'.p = 0) : ∃ D'' : Datum H, D''.p = D.p ∧ D''.q = D'.q ∧ SplitLat D''

theorem **exists_splitLat_indefinite** : ∃ D : Datum CM.LatModel.host, D.p = 1 ∧ D.q = 1 ∧ SplitLat D

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.CoeSelfDualLatticeInFrameCoordinatesEq

theorem **cF_apply_eq_complexConj** (D : D7) (x : D.host.E) : cF D x = IsCMField.complexConj D.host.E x

theorem **mem_latC_iff_symm_mem** (V : HSp D) (v : Fin V.rank → D.host.E) : v ∈ latC V ↔ V.frame.equivFun.symm v ∈ V.L0

theorem **coe_selfDualLatticeInFrameCoordinates_eq** (V : HSp D) : (latC V : Set (Fin V.rank → D.host.E)) = Formalism.MainNotions.StandardLattice.L0c D V

theorem **isLevelK_iff_finCong** (V : HSp D) (N : ℕ) (Kf : Subgroup (GA D V)) : IsLevelK V N Kf ↔ ∀ k, k ∈ Kf ↔ k ∈ GfC V ∧ FinCong V N k

theorem **isRat_one_one** (V : HSp D) : IsRat V 1 1

theorem **exists_isRat_of_mem_GQ** (V : HSp D) {γ : GA D V} (h : γ ∈ GQ V) : ∃ γ', IsRat V γ γ'

theorem **trace_model_eq_trace_hT** (V : HSp D) {r : ℕ} (hr : V.rank = r) (u v : Fin V.rank → D.host.E) : Algebra.trace ℚ D.host.E (-(-D.host.xi0) * herm (GlobalTheory.RecDict.gmW V hr) (u ∘ Fin.cast hr.symm) (v ∘ Fin.cast hr.symm)) = -Algebra.trace ℚ D.host.E (D.host.xi0 * V.hT (V.frame.equivFun.symm v) (V.frame.equivFun.symm u))

theorem **isom_iff_model** (Dw : Datum H) (V : HSp D) {r : ℕ} (hr : V.rank = r) (ι : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) : (∀ x y, ((Dw.EF x y : ℤ) : ℚ) = Algebra.trace ℚ D.host.E (D.host.xi0 * V.hT (V.frame.equivFun.symm (ι x)) (V.frame.equivFun.symm (ι y)))) ↔ ∀ x y, ((Dw.EF x y : ℤ) : ℚ) = Algebra.trace ℚ D.host.E (-(-D.host.xi0) * herm (GlobalTheory.RecDict.gmW V hr) (ι x ∘ Fin.cast hr.symm) (ι y ∘ Fin.cast hr.symm))

theorem **ok_iff_recOK** : C.OK ↔ GlobalTheory.RecDict.RecOK C.M V C.hr C.ι C.th C.x₀ ∧ (H.rt : D.host.E) = D.host.sqrtMinusD ∧ H.rt0 = D.host.φ₀ D.host.sqrtMinusD ∧ 1 ≤ Dw.p + Dw.q ∧ l.A = 1 ∧ GoodEmb c σ ∧ ∀ τ, Formalism.MainNotions.StandardLattice.IsInfEmb τ (C.θ τ)

theorem **markSpec_iff_mark** (h : GA D V) (p : Σ k, Frame (C.M.X k).Ψ r) : GlobalTheory.RecDict.MarkSpec C.M V C.hr C.ι C.th C.x₀ h p ↔ ∃ γ, C.Mark h p.1 γ ∧ p.2 = C.fr (γ * h) p.1

theorem **isDict_iff_markSpec** (dict : GA D V → Σ k, Frame (C.M.X k).Ψ r) : C.IsDict dict ↔ (∀ h, GlobalTheory.RecDict.MarkSpec C.M V C.hr C.ι C.th C.x₀ h (dict h)) ∧ ∀ k, ∃ h, dict h = ⟨k, C.fr 1 k⟩

theorem **unf_eq_toPj** (un : ∀ k, (C.M.X k).dom → FrPt Xo c (CAt σ)) : C.unf un = (GlobalTheory.FrameUnif.unfOf C.e C.F C.M un).toPj

theorem **mat_fr_one** (k : Fin C.F.n) (τ : (C.M.X k).Ψ) : mat (C.fr 1 k) τ = C.x₀ τ.1

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.ExistsCtxOkOfPins

theorem **sqrt_im_pos** (D : D7) : 0 < (D.host.φ₀ D.host.sqrtMinusD).im

theorem **domFor_one** (D : D7) (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) (hI : DatInt D j) : DomFor D j ((1 : ℤ) • intForm (efQ D j) hI)

theorem **sgnW_eq_one** (D : D7) (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) : sgnW D j = 1

theorem **latHat_eq_hat** (V : HSp D) : latHat V = GlobalTheory.AdLat.hat D.host.F (latC V : Set (Fin V.rank → D.host.E))

theorem **latC_fg** (V : HSp D) : (latC V).FG

theorem **latC_span** (V : HSp D) : Submodule.span D.host.E (latC V : Set (Fin V.rank → D.host.E)) = ⊤

theorem **finVec_mem_gLat_one** (V : HSp D) (v : Fin V.rank → D.host.E) : finVec V v ∈ gLat V 1 ↔ v ∈ latC V

theorem **iotaOK_at_lift** (D : D7) (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) : IotaOK (datumW D j hj) (D.sp.W j) (iotaW D j) where

theorem **hT_coord** (V : HSp D) (x y : Fin V.rank → D.host.E) : V.hT (V.frame.equivFun.symm y) (V.frame.equivFun.symm x) = -herm (-V.gramᵀ) x y

theorem **gram_det_ne** (V : HSp D) : V.gram.det ≠ 0

theorem **latG_one_span** (V : HSp D) : Submodule.span D.host.E (latG V 1 : Set (Fin V.rank → D.host.E)) = ⊤

theorem **isom_model** {V : HSp D} {ι : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)} (h : IotaOK Dw V ι) (x y : Dw.La) : ((Dw.EF x y : ℤ) : ℚ) = Algebra.trace ℚ D.host.E (-(-D.host.xi0) * herm (-V.gramᵀ) (ι x) (ι y))

theorem **exists_ctx_ok** (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (V : HSp D) (ι : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (hι : IotaOK Dw V ι) (hpq : 1 ≤ Dw.p + Dw.q) (hA : l.A = 1) (hσ : GoodEmb c σ) : ∃ C : Ctx Xo c σ V V.rank, C.OK ∧ C.ι = ι

theorem **exists_ctx_ok_at_lift** (D : D7) (j : ℕ) (hj : 1 ≤ j ∧ j ≤ D.R.m) : ∃ (l : Level (hostOfRecord D)) (Xo : Moduli (datumW D j hj) l) (c : Coeff l) (σ : c.F →+* ℂ) (C : Ctx Xo c σ (D.sp.W j) (D.sp.W j).rank), C.OK ∧ 3 ≤ (D.sp.W j).rank ∧ 0 < C.F.n ∧ l.M = D.host.g 0 * D.host.g 1

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.FrameRingGlRing

theorem **unit_rel** : Ideal.Quotient.mk (rel m R) (MvPolynomial.X (Sum.inr (Sum.inr true))) * Ideal.Quotient.mk (rel m R) (MvPolynomial.X (Sum.inr (Sum.inr false))) = 1

theorem **det_rel** (τ : J) : (glX m R τ).det * Ideal.Quotient.mk (rel m R) (MvPolynomial.X (Sum.inr (Sum.inl τ))) = 1

theorem **frameRing_glRing** : FrameRing m (glπ (K

theorem **FrameRing.injective** {A : Type} [CommRing A] [Algebra K A] {π : R →ₐ[K] A} {X : ∀ τ, Matrix (Fin (m τ)) (Fin (m τ)) A} {s : Aˣ} (h : FrameRing m π X s) : Function.Injective π

theorem **FrameRing.unit_not_mem** [Nontrivial R] {A : Type} [CommRing A] [Algebra K A] {π : R →ₐ[K] A} {X : ∀ τ, Matrix (Fin (m τ)) (Fin (m τ)) A} {s : Aˣ} (h : FrameRing m π X s) (r : R) : π r ≠ s

theorem **FrameRing.triRing** {A : Type} [CommRing A] [Algebra K A] {π : R →ₐ[K] A} {X : ∀ τ, Matrix (Fin (m τ)) (Fin (m τ)) A} {s : Aˣ} (h : FrameRing m π X s) : let I : Ideal A

theorem **spMap_id** (S : Type) [CommRing S] : spMap (RingHom.id S) = 𝟙 (Sp S)

theorem **secGen_frPt** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : SecGen y.x y.s

theorem **chart_le** {S T : Type} [CommRing S] [CommRing T] (x : Sp S ⟶ Xo.X) (x' : Sp T ⟶ Xo.X) (pm : Gx x' ⟶ Gx x) (h : pm ≫ (snd (Over.mk x) Xo.univ.t.G).left = (snd (Over.mk x') Xo.univ.t.G).left) (i : Fin 1 → Xo.univ.t.PP.L.ι) : (PPx x').U i ≤ pm ⁻¹ᵁ (PPx x).U i

theorem **scal_one** {S : Type} [CommRing S] (x : Sp S ⟶ Xo.X) (i : (PPx x).ι) : scal x 1 i = 1

theorem **frMap_iff** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y : FrPt Xo c S) (z : FrPt Xo c T) : FrMap φ y z ↔ z.x = spMap φ.toRingHom ≫ y.x ∧ (∀ τ i, z.u τ i = spMap (dualMap φ.toRingHom) ≫ y.u τ i) ∧ SecBC φ.toRingHom 1 y.x z.x y.s z.s

theorem **frBor_iff** {S : Type} [CommRing S] [Algebra c.F S] (t : Ix Dw → Sˣ) (y y' : FrPt Xo c S) : FrBor t y y' ↔ ∃ g : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) S, (∀ τ i, g τ i i = (t (some (τ, i)) : S)) ∧ (∀ τ i j, i < j → g τ i j = 0) ∧ FrChg g (t none : S) y y'

theorem **appLE_id_apply** (X : Scheme.{0}) (U : X.Opens) (h : U ≤ 𝟙 X ⁻¹ᵁ U) (z : Γ(X, U)) : (Scheme.Hom.appLE (𝟙 X) U U h).hom z = z

theorem **secBC_id_iff** {S : Type} [CommRing S] (x : Sp S ⟶ Xo.X) (a : S) (s s' : LB.Sec (PPx x)) : SecBC (RingHom.id S) a x x s s' ↔ ∀ i, s'.f i = scal x a i * s.f i

private theorem **secBC_unique_of_secGen** {S : Type} [CommRing S] (x x' : Sp S ⟶ Xo.X) (hx : x' = x) {a b : S} (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')) (ha : SecBC (RingHom.id S) a x x' s s') (hb : SecBC (RingHom.id S) b x x' s s') (hgen : SecGen x s) : a = b

theorem **frChg_unique** {S : Type} [CommRing S] [Algebra c.F S] {g g' : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) S} {a a' : S} {y y' : FrPt Xo c S} (h : FrChg g a y y') (h' : FrChg g' a' y y') : g = g' ∧ a = a'

private theorem **exists_secBC_of_secGen** {S : Type} [CommRing S] (x x' : Sp S ⟶ Xo.X) (hx : x' = x) (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')) (hgen : SecGen x s) : ∃ a, SecBC (RingHom.id S) a x x' s s'

theorem **exists_frChg** {S : Type} [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (h : y'.x = y.x) : ∃ g a, FrChg g a y y'

theorem **scal_mul** {S : Type} [CommRing S] (x : Sp S ⟶ Xo.X) (a b : S) (i : (PPx x).ι) : scal x (a * b) i = scal x a i * scal x b i

private theorem **isUnit_of_secBC_of_secGen** {S : Type} [CommRing S] (x x' : Sp S ⟶ Xo.X) (hx : x' = x) {a : S} (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')) (hs : SecBC (RingHom.id S) a x x' s s') (hgen : SecGen x s) (hgen' : SecGen x' s') : IsUnit a

theorem **FrChg.isUnit** {S : Type} [CommRing S] [Algebra c.F S] {g : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) S} {a : S} {y y' : FrPt Xo c S} (h : FrChg g a y y') : IsUnit a

theorem **OpenChart.eq_lift** {R : Type} [CommRing R] [Algebra c.F R] {j : Sp R ⟶ Xo.XF c} {y₀ : FrPt Xo c R} (h : OpenChart j y₀) : j = pullback.lift y₀.x (spMap (algebraMap c.F R)) (by rw [y₀.x_over, sOf]; simp [spMap])

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.NegBaseElementImNegBaseEmbedding

theorem **dictLat_iff** {c : Coeff l} {σ : c.F →+* ℂ} {g : ℕ} (e : Fin g ≃ Σ τ : D.host.E →+* ℂ, Fin (Dw.mult τ)) (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) (fun j => relab c σ (e j).1)) (M : GlobalTheory.FrameUnif.ModelsOf F r) (un : ∀ k, (M.X k).dom → FrPt Xo c (CAt σ)) (hun : ∀ k x, GlobalTheory.FrameUnif.FrameOK e F M k x (un k x)) (V : HSp D) (hr : V.rank = r) (io : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (th : ∀ k, (M.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (hR : RecOK M V hr io th x₀) (dict : GA D V → Σ k, Frame (M.X k).Ψ r) : DictLat ({ Comp

theorem **markSpec_left** (M : GlobalTheory.FrameUnif.ModelsOf F r) (V : HSp D) (hr : V.rank = r) (io : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (th : ∀ k, (M.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (h : GA D V) (p : Σ k, Frame (M.X k).Ψ r) (δ : GA D V) (hδ : δ ∈ GQ V) : MarkSpec M V hr io th x₀ (δ * h) p ↔ MarkSpec M V hr io th x₀ h p

theorem **herm_gmW** (V : HSp D) (hr : V.rank = r) (x y : Fin V.rank → D.host.E) : herm (gmW V hr) (x ∘ Fin.cast hr.symm) (y ∘ Fin.cast hr.symm) = −V.hT (V.frame.equivFun.symm y) (V.frame.equivFun.symm x)

theorem **recOK_isom** (M : GlobalTheory.FrameUnif.ModelsOf F r) (V : HSp D) (hr : V.rank = r) (io : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (th : ∀ k, (M.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (hR : RecOK M V hr io th x₀) (x y : Dw.La) : ((Dw.EF x y : ℤ) : ℚ) = Algebra.trace ℚ D.host.E (D.host.xi0 * V.hT (V.frame.equivFun.symm (io x)) (V.frame.equivFun.symm (io y)))

theorem **recOK_latIsom** (M : GlobalTheory.FrameUnif.ModelsOf F r) (V : HSp D) (hr : V.rank = r) (io : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (th : ∀ k, (M.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (hR : RecOK M V hr io th x₀) : LatIsom V Dw.EF io

theorem **recOK_gram** (M : GlobalTheory.FrameUnif.ModelsOf F r) (V : HSp D) (hr : V.rank = r) (io : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (th : ∀ k, (M.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (hR : RecOK M V hr io th x₀) (k : Fin F.n) : (M.X k).Gm = gmW V hr

theorem **neg_baseElement_im_neg_baseEmbedding** (D : D7) (τ : D.host.E →+* ℂ) (hτ : τ D.host.sqrtMinusD = D.host.φ₀ D.host.sqrtMinusD) : (τ (−D.host.xi0)).im < 0

theorem **th_conj** {τ : D.host.E →+* ℂ} {θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ} (h : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ) (z : Ainf D.host.F ⊗[D.host.F] D.host.E) : θ (sigR (Ainf D.host.F) D.sp.cm.sig z) = starRingEnd ℂ (θ z)

theorem **cT_map** {A : Type} [CommRing A] {n : Type} (s : A →+* A) (φ : A →+* ℂ) (h : ∀ z, φ (s z) = starRingEnd ℂ (φ z)) (M : Matrix n n A) : (Formalism.UG.cT s M).map φ = (M.map φ).conjTranspose

theorem **uni_map** {A : Type} [CommRing A] {n : Type} [Fintype n] (s : A →+* A) (φ : A →+* ℂ) (h : ∀ z, φ (s z) = starRingEnd ℂ (φ z)) (M J : Matrix n n A) (hM : Formalism.UG.cT s M * J * M = J) (J' M' : Matrix n n ℂ) (hJ : J.map φ = J') (hM' : M' = M.map φ) : M'.conjTranspose * J' * M' = J'

theorem **archM_unitary** (V : HSp D) {τ : D.host.E →+* ℂ} {θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ} (h : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ) (g : GA D V) : (archM V θ g).conjTranspose * V.gram.map τ * archM V θ g = V.gram.map τ

theorem **gram_eq** {E : Type} [Field E] [NumberField E] [IsCMField E] (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (x : Matrix (Fin r) (Fin r) ℂ) : gram τ G x = (x.conjTranspose * (G.map τ).transpose * x).transpose

theorem **markSpec_orthonormal** (M : GlobalTheory.FrameUnif.ModelsOf F r) (V : HSp D) (hr : V.rank = r) (io : Dw.La →ₗ[𝒪 D.host.E] (Fin V.rank → D.host.E)) (th : ∀ k, (M.X k).Ψ → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ) (hR : RecOK M V hr io th x₀) (h : GA D V) (p : Σ k, Frame (M.X k).Ψ r) (hp : MarkSpec M V hr io th x₀ h p) (τ : (M.X p.1).Ψ) : gram τ.1 (M.X p.1).Gm (mat p.2 τ) = Matrix.diagonal fun a : Fin r => ((sgn (M.X p.1).p0 (a : ℕ) : ℝ) : ℂ)

theorem **exists_base_frame** (M : GlobalTheory.FrameUnif.ModelsOf F r) (V : HSp D) (hr : V.rank = r) (hB : GlobalTheory.FrameUnif.ModelBase M) (hW : ∀ k, ModelW V hr (M.X k)) : ∃ x₀ : (D.host.E →+* ℂ) → Matrix (Fin r) (Fin r) ℂ, ∀ k, IsOrthogonalFrame (M.X k) x₀

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.PermutesRefl

theorem **permutes_refl** (F : FamilyUnif Xo σ ty) : ConjugationPermutesPeriods F (RingEquiv.refl ℂ) id

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.PowFourComplementaryModEight

theorem **perFr_add** (X : Data E r) (x : Frame X.Ψ r) (v w : Fin r → E) (τ : E →+* ℂ) (i : ℕ) : perFr X x (v + w) τ i = perFr X x v τ i + perFr X x w τ i

theorem **exists_isPerFr** {g : ℕ} (e : Fin g ≃ Σ τ : E →+* ℂ, Fin (Dw.mult τ)) (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) (fun j => relab c σ (e j).1)) (M : ModelsOf F r) (k : Fin F.n) : ∃ Pf, IsPerFr e F M k Pf

theorem **normConst_pos** {g : ℕ} (ty : Fin g → (E →+* ℂ)) {ξ₀ : E} (h : ξ₀ ≠ 0) : 0 < normConst ty ξ₀

theorem **isNormUn_ne** {g : ℕ} {e : Fin g ≃ Σ τ : E →+* ℂ, Fin (Dw.mult τ)} {F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) (fun j => relab c σ (e j).1)} {M : ModelsOf F r} {k : Fin F.n} (T : ThetaData e F M k) (un : (M.X k).dom → FrPt Xo c (CAt σ)) (h : IsNormUn T un) (x : (M.X k).dom) : ∃ κ : ℂ, κ ≠ 0 ∧ IsThetaPt T (T.co.symm x.1) κ (un x)

theorem **pow_four_complementary_modEight** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : (∃ v : 𝒪 E, H.w0 ^ 4 + (1 − H.w0) ^ 4 = 1 + 8 ∗ v) ∧ ∃ v : 𝒪 E, H.w0 ^ 4 ∗ (1 − H.w0) ^ 4 = 8 ∗ v

theorem **sign_genus_one** (t : ℂ) (ht : 0 < t.im) : (hermForm (g

theorem **models_of_latModel** {g : ℕ} {ty : Fin g → (E →+* ℂ)} {σ' : KA l →+* ℂ} (F : FamilyUnif Xo σ' ty) (Ψ : Finset (E →+* ℂ)) (hΨ : ∀ τ, τ ∈ Ψ ↔ τ (H.rt : E) = H.rt0) (hcm : ∀ τ : E →+* ℂ, τ ∈ Ψ ↔ ComplexEmbedding.conjugate τ ∉ Ψ) (Gm : Matrix (Fin r) (Fin r) E) (hG : ∀ a b, cc (Gm a b) = Gm b a) (hdet : Gm.det ≠ 0) (hrk : r = Dw.p + Dw.q) (hsig : ∀ τ : Ψ, ∃ P : Matrix (Fin r) (Fin r) ℂ, P.det ≠ 0 ∧ gram τ.1 Gm P = Matrix.diagonal fun a : Fin r => ((sgn Dw.q (a : ℕ) : ℝ) : ℂ)) (ξ₀ : E) (hξ : cc ξ₀ = −ξ₀) (hξ0 : ξ₀ ≠ 0) (hneg : ∀ τ ∈ Ψ, (τ ξ₀).im < 0) (hL : ∀ i, LatModel l Ψ Gm ξ₀ Dw.q (F.Di i)) : ∃ M : ModelsOf F r, ModelBase M ∧ NegXi M

theorem **thetaData_of** {g : ℕ} (e : Fin g ≃ Σ τ : E →+* ℂ, Fin (Dw.mult τ)) (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) (fun j => relab c σ (e j).1)) (M : ModelsOf F r) (k : Fin F.n) (Pf : Frame (M.X k).Ψ r → ((F.Di k).La →+ (Fin g → ℂ))) (per : IsPerFr e F M k Pf) (m : ℕ) (co : (Fin m → ℂ) ≃L[ℂ] Frame (M.X k).Ψ r) (ho : IsOpen (M.X k).dom) (hc : IsPreconnected (M.X k).dom) (holPer : IsHolPeriods (F.Di k) (fun j => relab c σ (e j).1) (co ⁻¹' (M.X k).dom) (fun b => Pf (co b))) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] (F.Di k).La) (sympl : ∀ a a', (F.Di k).EF (bas a) (bas a') = −omegaStd a a') (adapt : (∀ i, ∃ v, H.w0 ^ 4 • bas (Pi.single (Sum.inl i) 1) = bas (Pi.single (Sum.inl i) 1) + (8 : ℕ) • v) ∧ ∀ i, ∃ v, H.w0 ^ 4 • bas (Pi.single (Sum.inr i) 1) = (8 : ℕ) • v) (h ta : (Fin m → ℂ) → Matrix (Fin g) (Fin g) ℂ) (norm : ∀ b ∈ co ⁻¹' (M.X k).dom, IsUnit (h b).det ∧ (ta b

Development.AbelianVarietyTheory.Definitions.ExistenceOfLattices.SmulBasisMemHatIff

theorem **_root_.GlobalTheory.AdLat.Model.integral_iff** (x : F) : (∀ v, algebraMap F (FiniteAdeleRing (𝒪 F) F) x v ∈ v.adicCompletionIntegers F) ↔ x ∈ (algebraMap (𝒪 F) F).range

theorem **ext_tmul** {n : ℕ} (φ : (Fin n → E) →ₗ[F] F) (a : FiniteAdeleRing (𝒪 F) F) (x : Fin n → E) : ext φ (fun i => a ⊗ₜ x i) = algebraMap F _ (φ x) ∗ a

theorem **functional_integral** {n : ℕ} (L : Set (Fin n → E)) (φ : (Fin n → E) →ₗ[F] F) (hφ : ∀ x ∈ L, φ x ∈ (algebraMap (𝒪 F) F).range) {w : Fin n → E} (hw : fv F w ∈ hat F L) : φ w ∈ (algebraMap (𝒪 F) F).range

theorem **fv_mem_hat** {n : ℕ} {L : Set (Fin n → E)} {x : Fin n → E} (hx : x ∈ L) : fv F x ∈ hat F L

theorem **latOf_one_freeLat** {n : ℕ} {ι : Type} (b : Module.Basis ι F (Fin n → E)) : latOf F 1 (freeLat b) = freeLat b

theorem **smul_basis_mem_hat_iff** {n : ℕ} {ι : Type} (b : Module.Basis ι F (Fin n → E)) (j : ι) (t : F) : fv F (t • b j) ∈ hat F (freeLat b) ↔ t ∈ (algebraMap (𝒪 F) F).range

theorem **_root_.GlobalTheory.AdLat.Model.sg_tmul** (c : E ≃ₐ[F] E) (a : FiniteAdeleRing (𝒪 F) F) (x : E) : _root_.GlobalTheory.AdLat.sg (FiniteAdeleRing (𝒪 F) F) c (a ⊗ₜ x) = a ⊗ₜ c x

theorem **hermA_fv** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (x y : Fin n → E) : hermA c G (fv F x) (fv F y) = (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ ∑ a, ∑ b, x a ∗ G a b ∗ c (y b)

theorem **_root_.GlobalTheory.AdLat.Model.hermA_smul_left** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (t : _root_.GlobalTheory.AdLat.RA F E) (x y : Fin n → _root_.GlobalTheory.AdLat.RA F E) : _root_.GlobalTheory.AdLat.hermA c G (t • x) y = t ∗ _root_.GlobalTheory.AdLat.hermA c G x y

theorem **_root_.GlobalTheory.AdLat.Model.hermA_smul_right** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (t : _root_.GlobalTheory.AdLat.RA F E) (x y : Fin n → _root_.GlobalTheory.AdLat.RA F E) : _root_.GlobalTheory.AdLat.hermA c G x (t • y) = _root_.GlobalTheory.AdLat.sg _ c t ∗ _root_.GlobalTheory.AdLat.hermA c G x y

theorem **sg_sg** (c : E ≃ₐ[F] E) (hc : ∀ x, c (c x) = x) (z : RA F E) : sg _ c (sg _ c z) = z

theorem **_root_.GlobalTheory.AdLat.Model.hermA_symm** (c : E ≃ₐ[F] E) (hc : ∀ x, c (c x) = x) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (hG : ∀ a b, c (G a b) = G b a) (x y : Fin n → _root_.GlobalTheory.AdLat.RA F E) : _root_.GlobalTheory.AdLat.sg _ c (_root_.GlobalTheory.AdLat.hermA c G x y) = _root_.GlobalTheory.AdLat.hermA c G y x

theorem **fv_mulVec** {n : ℕ} (γ : Matrix (Fin n) (Fin n) E) (x : Fin n → E) : (γ.map fun e => ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ e : RA F E)).mulVec (fv F x) = fv F (γ.mulVec x)

theorem **latOf_rational** {n : ℕ} (γ : Matrix (Fin n) (Fin n) E) (L : Set (Fin n → E)) : latOf F (γ.map fun e => ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ e : RA F E)) L = latOf F 1 (γ.mulVec '' L)

Development.AbelianVarietyTheory.Definitions.Foundations.AbsCharNotUnitary

theorem **inf_ne** (a : (InfiniteAdeleRing F)ˣ) (w : InfinitePlace F) : (a : InfiniteAdeleRing F) w ≠ 0

theorem **logInf_mul** (a b : (InfiniteAdeleRing F)ˣ) : logInf F (a ∗ b) = logInf F a + logInf F b

theorem **logInf_one** : logInf F 1 = 0

theorem **continuous_logInf** : Continuous (logInf F)

theorem **logFin_mul** (t s : (FiniteAdeleRing (𝒪 F) F)ˣ) : logFin F (t * s) = logFin F t + logFin F s

theorem **logFin_one** : logFin F 1 = 0

theorem **logFin_eventually** : ∀ᶠ t in nhds (1 : (FiniteAdeleRing (𝒪 F) F)ˣ), logFin F t = 0

theorem **logNorm_mul** (a b : Ideles F) : logNorm F (a * b) = logNorm F a + logNorm F b

theorem **logInf_principal** (x : Fˣ) : logInf F (infPart F (diagUnits F x)) = Real.log ((|Algebra.norm ℚ (x : F)| : ℚ) : ℝ)

theorem **logNorm_principal** (x : Fˣ) : logNorm F (diagUnits F x) = 0

theorem **absHom_val** (s : ℂ) (a : Ideles F) : ((absHom F s a : ℂˣ) : ℂ) = Complex.exp (s * logNorm F a)

theorem **continuous_logNorm** : Continuous (logNorm F)

theorem **continuous_absHom** (s : ℂ) : Continuous (absHom F s)

theorem **idealOf_local** (v : HeightOneSpectrum (𝒪 F)) (x : (v.adicCompletion F)ˣ) (n : ℤ) (hx : Valued.v (x : v.adicCompletion F) = WithZero.exp (-n)) : GlobalTheory.Idele.idealOf (finPart F (localIdele F v x)) = (v.asIdeal : FractionalIdeal (𝒪 F)⁰ F) ^ n

theorem **logNorm_local** (v : HeightOneSpectrum (𝒪 F)) (x : (v.adicCompletion F)ˣ) (n : ℤ) (hx : Valued.v (x : v.adicCompletion F) = WithZero.exp (-n)) : logNorm F (localIdele F v x) = -(n * Real.log (Ideal.absNorm v.asIdeal : ℕ))

theorem **infDiag_parts** (k : Fˣ) : infPart F (infDiag k) = infPart F (diagUnits F k) ∧ finPart F (infDiag k) = 1

theorem **logNorm_infDiag** (k : Fˣ) : logNorm F (infDiag k) = ∑ φ : F →+* ℂ, Real.log ‖φ (k : F)‖

theorem **absChar_isAbsPow** (s : ℝ) : (absChar F s).IsAbsPow s

theorem **absChar_not_unitary** : ¬ (absChar F 1).IsUnitary

theorem **absChar_unitary** (t : ℝ) : (absChar F (t * Complex.I)).IsUnitary

theorem **unitaryNorm** {χ : HChar F} (h : χ.IsUnitary) (s : ℝ) : χ.IsUnitaryNormOf (χ * absChar F s) s

theorem **isCond_top** {χ : HChar F} (h : ∀ v, χ.UnrAt v) : χ.IsCond ⊤

theorem **tendsto_cut** (a : FiniteAdeleRing (𝒪 F) F) : Filter.Tendsto (cut a) Filter.atTop (nhds a)

theorem **prod_local** (k : Fˣ) (T : Finset (HeightOneSpectrum (𝒪 F))) : ((infDiag k * ∏ v ∈ T, localIdele F v (atFin v k) : Ideles F) : AdeleRing (𝒪 F) F) = (((diagUnits F k : Ideles F) : AdeleRing (𝒪 F) F).1, cut ((diagUnits F k : Ideles F) : AdeleRing (𝒪 F) F).2 T)

theorem **tendsto_prod_local** (k : Fˣ) : Filter.Tendsto (fun T : Finset (HeightOneSpectrum (𝒪 F)) => infDiag k * ∏ v ∈ T, localIdele F v (atFin v k)) Filter.atTop (nhds (diagUnits F k))

theorem **locK_unr** (χ : HChar F) {v : HeightOneSpectrum (𝒪 F)} (h : χ.UnrAt v) (k : Fˣ) : χ.locK v k = χ.val v ^ ordF v k

theorem **finite_ordF** (k : Fˣ) : {v : HeightOneSpectrum (𝒪 F) | ordF v k ≠ 0}.Finite

theorem **productRule** (χ : HChar F) : χ.ProductRule

theorem **principal_unr** (χ : HChar F) (h : ∀ v, χ.UnrAt v) (k : Fˣ) : ∏ᶠ v : HeightOneSpectrum (𝒪 F), χ.val v ^ ordF v k = (χ.inf k)⁻¹

theorem **exists_over** (K : Type) [Field K] [NumberField K] {p : ℕ} (hp : p.Prime) : ∃ v : HeightOneSpectrum (𝒪 K), ((p : ℕ) : 𝒪 K) ∈ v.asIdeal

theorem **nonempty_piBar** : Nonempty (PiBar H)

theorem **exists_faithful** {p : ℕ} (hp : p.Prime) : ∃ ω : (ZMod p)ˣ →* ℂˣ, Function.Injective ω

theorem **nonempty_omg** : Nonempty (Omg H)

theorem **isEtaAlg_of** {χ : HChar H.E} (h : IsEtaT H χ) : IsEtaAlg H (χ * absChar H.E ((1 / 2 : ℝ) : ℂ))

theorem **isEtaAlg_inf** {χ : HChar H.E} (h : IsEtaAlg H χ) (k : H.Eˣ) : ((χ.inf k : ℂˣ) : ℂ) * ∏ g : R.Psi, ((‖embPsi H g (k : H.E)‖ : ℝ) : ℂ) = (∏ g : R.Psi, embPsi H g (k : H.E)) * ∏ φ : H.E →+* ℂ, ((‖φ (k : H.E)‖ : ℝ) : ℂ) ^ (((1 / 2 : ℝ)) : ℂ)

theorem **not_isDelta_one** {p : ℕ} (hp : p.Prime) : ¬ IsDelta (H

theorem **not_isEtaT_one** : ¬ IsEtaT H 1

Development.AbelianVarietyTheory.Definitions.Foundations.ActWBOfInvariant

theorem **Gamma_le_Gamma0** {l M : ℕ} (h : l ∣ M) : CongruenceSubgroup.Gamma M ≤ CongruenceSubgroup.Gamma0 l

theorem **crtF_mod_left** (m N : ℕ) : ((crtF m N : ℤ) : ZMod m) = 0

theorem **crtF_mod_right** {m N : ℕ} [NeZero N] (h : Nat.Coprime m N) : ((crtF m N : ℤ) : ZMod N) = 1

theorem **sigN_mem_Gamma** (m N : ℕ) : sigN m N ∈ CongruenceSubgroup.Gamma m

theorem **det_wMat** (l N : ℕ) : (wMat l N).det = N

theorem **exists_wMat_mul** {l N : ℕ} [NeZero N] (h : Nat.Coprime l N) {γ : SL(2, ℤ)} (hγ : γ ∈ Gam N l) : ∃ γ' ∈ Gam N l, wMat l N * (γ : Matrix (Fin 2) (Fin 2) ℤ) = (γ' : Matrix (Fin 2) (Fin 2) ℤ) * wMat l N

theorem **vecMul_gR** (l N : ℕ) [Fact l.Prime] (hN : (N : ZMod l) ≠ 0) (r₀ : ZMod l) : (![r₀, 1] : Fin 2 → ZMod l) ᵥ* ((GlobalTheory.ZetaB.gR l N r₀ : Matrix (Fin 2) (Fin 2) ℤ).map (Int.castRingHom (ZMod l))) = ![0, 1]

theorem **mem_lnFn** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (n : ℕ) (b : X.Fn l) : b ∈ lnFn X l n ↔ ∀ u, b u ∈ (idl l) ^ n • (⊤ : Submodule ℤ_[l] (X.Khat l))

theorem **act_mem_lnFn** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] {n : ℕ} (m : Matrix (Fin 2) (Fin 2) ℤ) {b : X.Fn l} (hb : b ∈ lnFn X l n) : X.act l m b ∈ lnFn X l n

theorem **kmap_clsHat** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (φ : X.K ≃+* X.K) (a : X.K) : X.kmap l φ (GlobalTheory.ZetaB.clsHat X l a) = GlobalTheory.ZetaB.clsHat X l (φ a)

theorem **toGQ_coe_SL** (γ : SL(2, ℤ)) : toGQ (γ : Matrix (Fin 2) (Fin 2) ℤ) = ofSL γ

theorem **act_act_SL** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (γ γ' : SL(2, ℤ)) (b : X.Fn l) : X.act l (γ' : Matrix (Fin 2) (Fin 2) ℤ) (X.act l (γ : Matrix (Fin 2) (Fin 2) ℤ) b) = X.act l ((γ * γ' : SL(2, ℤ)) : Matrix (Fin 2) (Fin 2) ℤ) b

theorem **InvMod.act_congr** {X : Tower Ω ε} {l : ℕ} [Fact l.Prime] {n : ℕ} {H : Subgroup SL(2, ℤ)} {b : X.Fn l} (hb : InvMod X l n H b) {m m' : SL(2, ℤ)} (h : m * m'⁻¹ ∈ H) : X.act l (m : Matrix (Fin 2) (Fin 2) ℤ) b = X.act l (m' : Matrix (Fin 2) (Fin 2) ℤ) b ∈ lnFn X l n

theorem **invMod_secT** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] {n M : ℕ} (hM : l ^ n ∣ M) (w : ℕ) {a : X.K} (ha : a ∈ topL X M) : InvMod X l n (CongruenceSubgroup.Gamma M) (secT X l w a)

theorem **InvMod.wB** {X : Tower Ω ε} {l N : ℕ} [Fact l.Prime] [NeZero N] (h : Nat.Coprime l N) {n : ℕ} {b : X.Fn l} (hb : InvMod X l n (Gam N l) b) : InvMod X l n (Gam N l) (wB X l N b)

theorem **cycM_apply** [IsAlgClosure ℚ Ω] (M : ℕ) [NeZero M] (s : Ω ≃ₐ[ℚ] Ω) {z : Ω} (hz : z ^ M = 1) : s z = z ^ (cycM M s : ZMod M).val

theorem **repM_apply** [IsAlgClosure ℚ Ω] (M : ℕ) [NeZero M] (y : (ZMod M)ˣ) {z : Ω} (hz : z ^ M = 1) : repM (Ω

theorem **shArg_mem_fixMu** [IsAlgClosure ℚ Ω] (M : ℕ) [NeZero M] (s : Ω ≃ₐ[ℚ] Ω) (y : (ZMod M)ˣ) : shArg M s y ∈ fixMu Ω M

Development.AbelianVarietyTheory.Definitions.Foundations.AestronglyMeasurableQuotMeasureIffLift

@[simp] theorem **IsSubrep.coe_rep_apply** (hW : IsSubrep π W) (g : G) (v : W) : (hW.rep g v : H) = π g v

theorem **IsSubrep.completeSpace** [CompleteSpace H] (hW : IsSubrep π W) : CompleteSpace W

theorem **isSubrep_top** : IsSubrep π ⊤

theorem **IsSubrep.inf** (hW : IsSubrep π W) (hU : IsSubrep π U) : IsSubrep π (W ⊓ U)

theorem **isSubrep_topologicalClosure** (h : ∀ g, ∀ v ∈ W, π g v ∈ W) : IsSubrep π W.topologicalClosure

theorem **isSubrep_closure_sSup** {S : Set (Submodule ℂ H)} (h : ∀ W ∈ S, IsSubrep π W) : IsSubrep π (sSup S).topologicalClosure

theorem **isSubrep_disc** : IsSubrep π (disc π)

theorem **IsIrredSubrep.le_disc** (hW : IsIrredSubrep π W) : W ≤ disc π

theorem **disc_le** (hU : IsClosed (U : Set H)) (h : ∀ W, IsIrredSubrep π W → W ≤ U) : disc π ≤ U

theorem **IsSubrep.isSubrep_rep_iff** (hW : IsSubrep π W) {T : Submodule ℂ W} : IsSubrep hW.rep T ↔ IsSubrep π (T.map W.subtype)

theorem **IsSubrep.isIrredSubrep_iff** (hW : IsSubrep π W) : IsIrredSubrep π W ↔ IsTopIrreducible hW.rep

theorem **isTopIrreducible_iff_top** : IsTopIrreducible π ↔ IsIrredSubrep π ⊤

theorem **IsTopIrreducible.disc_eq_top** (h : IsTopIrreducible π) : disc π = ⊤

theorem **isIrredSubrep_span_singleton** {v : H} (hv : v ≠ 0) (h : ∀ g, ∃ c : ℂ, π g v = c • v) : IsIrredSubrep π (ℂ • v)

theorem **IsSubrep.isUnitaryRep** (hW : IsSubrep π W) (hπ : IsUnitaryRep π) : IsUnitaryRep hW.rep

theorem **IsSubrep.orthogonal** (hW : IsSubrep π W) (hπ : IsUnitaryRep π) : IsSubrep π W⊥

theorem **IsSubrep.starProjection_apply** (hW : IsSubrep π W) (hπ : IsUnitaryRep π) [W.HasOrthogonalProjection] (g : G) (v : H) : W.starProjection (π g v) = π g (W.starProjection v)

theorem **IsSubrep.map_mem_of_isRepOp** (hW : IsSubrep π W) {T : H →L[ℂ] H} (hT : IsRepOp π μ f T) {v : H} (hv : v ∈ W) : T v ∈ W

theorem **IsSubrep.coe_repOp_rep** (hW : IsSubrep π W) {T : H →L[ℂ] H} (hT : IsRepOp π μ f T) : IsRepOp hW.rep μ f (repOp hW.rep μ f) ∧ ∀ v : W, (repOp hW.rep μ f v : H) = T v

theorem **isRepOp_of_integrable** [SecondCountableTopology G] [OpensMeasurableSpace G] (hπ : IsUnitaryRep π) (hf : Integrable f μ) : IsRepOp π μ f (repOp π μ f)

theorem **mk_eq_iff** {x y : G} : mk Γ x = mk Γ y ↔ ∃ γ : Γ, γ * x = y

@[simp] theorem **mk_mul** (γ : Γ) (g : G) : mk Γ (γ * g) = mk Γ g

@[simp] theorem **rmul_mk** (g x : G) : rmul Γ g (mk Γ x) = mk Γ (x * g)

@[simp] theorem **rmul_one** (q : Quot Γ) : rmul Γ 1 q = q

theorem **rmul_mul** (g h : G) (q : Quot Γ) : rmul Γ (g * h) q = rmul Γ h (rmul Γ g q)

theorem **leftInv_lift** {X : Type*} (F : Quot Γ → X) : LeftInv Γ (lift Γ F)

theorem **LeftInv.eq_of_mk_eq** {X : Type*} {φ : G → X} (h : LeftInv Γ φ) {x y : G} (e : mk Γ x = mk Γ y) : φ x = φ y

theorem **LeftInv.descend_mk** {X : Type*} {φ : G → X} (h : LeftInv Γ φ) (g : G) : descend Γ φ (mk Γ g) = φ g

theorem **LeftInv.lift_descend** {X : Type*} {φ : G → X} (h : LeftInv Γ φ) : lift Γ (descend Γ φ) = φ

@[simp] theorem **descend_lift** {X : Type*} (F : Quot Γ → X) : descend Γ (lift Γ F) = F

theorem **LeftInv.mul_right** {X : Type*} {φ : G → X} (h : LeftInv Γ φ) (g : G) : LeftInv Γ fun x => φ (x * g)

theorem **leftInv_orbitSum** (f : G → ℂ) : LeftInv Γ (orbitSum Γ f)

theorem **kernel_eq_orbitSum** (f : G → ℂ) (x : G) : kernel Γ f x = orbitSum Γ fun y => f (x⁻¹ * y)

@[simp] theorem **starFn_starFn** (f : G → ℂ) : starFn (starFn f) = f

theorem **kernel_starFn** (f : G → ℂ) (x y : G) : kernel Γ (starFn f) x y = conj (kernel Γ f y x)

theorem **leftInv_kernel_right** (f : G → ℂ) (x : G) : LeftInv Γ (kernel Γ f x)

theorem **leftInv_kernel_left** (f : G → ℂ) (y : G) : LeftInv Γ fun x => kernel Γ f x y

@[simp] theorem **kernelQ_mk** (f : G → ℂ) (x y : G) : kernelQ Γ f (mk Γ x) (mk Γ y) = kernel Γ f x y

theorem **hasCompactSupport_starFn** {f : G → ℂ} (hf : HasCompactSupport f) : HasCompactSupport (starFn f)

theorem **finite_mem_compact** [T2Space G] [DiscreteTopology Γ] {C : Set G} (hC : IsCompact C) : {γ : Γ | (γ : G) ∈ C}.Finite

theorem **finite_support_kernel** [T2Space G] [DiscreteTopology Γ] {f : G → ℂ} (hf : HasCompactSupport f) (x y : G) : (Function.support fun γ : Γ => f (x⁻¹ * γ * y)).Finite

theorem **hasSum_kernel** [T2Space G] [DiscreteTopology Γ] {f : G → ℂ} (hf : HasCompactSupport f) (x y : G) : HasSum (fun γ : Γ => f (x⁻¹ * γ * y)) (kernel Γ f x y)

theorem **measurableSet_stdDom** : MeasurableSet (stdDom Γ)

theorem **existsUnique_stdDom** (x : G) : ∃! γ : Γ, (γ : G) * x ∈ stdDom Γ

theorem **isFundamentalDomain_stdDom** (μ : Measure G) : IsFundamentalDomain Γ (stdDom Γ) μ

@[simp] theorem **mk_sec** (q : Quot Γ) : mk Γ (sec Γ q) = q

theorem **sec_mem** (q : Quot Γ) : sec Γ q ∈ stdDom Γ

theorem **sec_eq** {q : Quot Γ} {x : G} (hx : x ∈ stdDom Γ) (e : mk Γ x = q) : sec Γ q = x

theorem **measurable_mk** : Measurable (mk Γ)

theorem **measurableEmbedding_sec** : MeasurableEmbedding (sec Γ) where

theorem **measurable_rmul_uncurry** : Measurable fun p : Quot Γ × G => rmul Γ p.2 p.1

theorem **measurable_rmul** (g : G) : Measurable (rmul Γ g)

theorem **measurePreserving_sec** : MeasurePreserving (sec Γ) (quotMeasure Γ μ) (μ.restrict (stdDom Γ))

theorem **quotMeasure_smul** (c : ℝ≥0∞) : quotMeasure Γ (c • μ) = c • quotMeasure Γ μ

theorem **quotMeasure_eq** (h : IsFundamentalDomain Γ 𝓕 μ) : quotMeasure Γ μ = (μ.restrict 𝓕).map (mk Γ)

theorem **measurePreserving_mk** (h : IsFundamentalDomain Γ 𝓕 μ) : MeasurePreserving (mk Γ) (μ.restrict 𝓕) (quotMeasure Γ μ)

theorem **quotMeasure_univ** (h : IsFundamentalDomain Γ 𝓕 μ) : quotMeasure Γ μ univ = μ 𝓕

theorem **ae_quotMeasure_iff** {p : Quot Γ → Prop} : (∀ᵐ q ∂quotMeasure Γ μ, p q) ↔ ∀ᵐ g ∂μ, p (mk Γ g)

theorem **quasiMeasurePreserving_mk** : Measure.QuasiMeasurePreserving (mk Γ) μ (quotMeasure Γ μ)

theorem **lintegral_quotMeasure** (h : IsFundamentalDomain Γ 𝓕 μ) (F : Quot Γ → ℝ≥0∞) : ∫⁻ q, F q ∂quotMeasure Γ μ = ∫⁻ g in 𝓕, F (mk Γ g) ∂μ

theorem **integral_quotMeasure** {X : Type*} [NormedAddCommGroup X] [NormedSpace ℝ X] (h : IsFundamentalDomain Γ 𝓕 μ) (F : Quot Γ → X) : ∫ q, F q ∂quotMeasure Γ μ = ∫ g in 𝓕, F (mk Γ g) ∂μ

theorem **aestronglyMeasurable_quotMeasure_iff** (h : IsFundamentalDomain Γ 𝓕 μ) {F : Quot Γ → ℂ} : AEStronglyMeasurable F (quotMeasure Γ μ) ↔ AEStronglyMeasurable (lift Γ F) (μ.restrict 𝓕)

theorem **aestronglyMeasurable_quotMeasure_iff_lift** {F : Quot Γ → ℂ} : AEStronglyMeasurable F (quotMeasure Γ μ) ↔ AEStronglyMeasurable (lift Γ F) μ

theorem **memLp_quotMeasure_iff** (h : IsFundamentalDomain Γ 𝓕 μ) {F : Quot Γ → ℂ} {p : ℝ≥0∞} : MemLp F p (quotMeasure Γ μ) ↔ MemLp (lift Γ F) p (μ.restrict 𝓕)

theorem **lintegral_eq_lintegral_tsum** {f : G → ℝ≥0∞} (hf : AEMeasurable f μ) : ∫⁻ g, f g ∂μ = ∫⁻ q, ∑' γ : Γ, f (γ * q.out) ∂quotMeasure Γ μ

theorem **integral_eq_integral_orbitSum** {f : G → ℂ} (hf : Integrable f μ) : ∫ g, f g ∂μ = ∫ q, descend Γ (orbitSum Γ f) q ∂quotMeasure Γ μ

theorem **quotMeasure_le** {A : Set (Quot Γ)} (hA : MeasurableSet A) {T : Set G} (h : A ⊆ mk Γ '' T) : quotMeasure Γ μ A ≤ μ T

theorem **measurePreserving_rmul** [μ.IsMulRightInvariant] (g : G) : MeasurePreserving (rmul Γ g) (quotMeasure Γ μ) (quotMeasure Γ μ)

theorem **memL2_lift** (F : L2 Γ μ) : MemL2Quotient Γ μ (lift Γ F)

@[simp] theorem **toL2_lift** (F : L2 Γ μ) : (memL2_lift F).toL2 = F

theorem **MemL2Quotient.coeFn_toL2** (h : MemL2Quotient Γ μ φ) : h.toL2 =ᵐ[quotMeasure Γ μ] descend Γ φ

theorem **MemL2Quotient.integrable_mul_conj** (h : MemL2Quotient Γ μ φ) (h' : MemL2Quotient Γ μ ψ) : Integrable (fun q => descend Γ φ q * conj (descend Γ ψ q)) (quotMeasure Γ μ)

theorem **MemL2Quotient.inner_toL2_eq_theta_pairing** (h : MemL2Quotient Γ μ φ) (h' : MemL2Quotient Γ μ ψ) : ⟪h'.toL2, h.toL2⟫_ℂ = pet Γ μ φ ψ

theorem **LeftInv.convR** (h : LeftInv Γ φ) (f : G → ℂ) : LeftInv Γ (convR μ f φ)

theorem **convR_eq** (f φ : G → ℂ) (x : G) : convR μ f φ x = ∫ y, f (x⁻¹ * y) * φ y ∂μ

theorem **memL2_iff_fundamental_domain** (h : IsFundamentalDomain Γ 𝓕 μ) : MemL2Quotient Γ μ φ ↔ LeftInv Γ φ ∧ MemLp φ 2 (μ.restrict 𝓕)

theorem **pet_eq** (h : IsFundamentalDomain Γ 𝓕 μ) (hφ : LeftInv Γ φ) (hψ : LeftInv Γ ψ) : pet Γ μ φ ψ = ∫ g in 𝓕, φ g * conj (ψ g) ∂μ

theorem **MemL2Quotient.lift_toL2_ae_eq** (h : MemL2Quotient Γ μ φ) : lift Γ h.toL2 =ᵐ[μ] φ

theorem **l2DiscreteQuotient_ext_iff_lift_ae_eq** {F F' : L2 Γ μ} : F = F' ↔ lift Γ F =ᵐ[μ] lift Γ F'

theorem **MemL2Quotient.toL2_eq_iff_ae_eq** (h : MemL2Quotient Γ μ φ) (h' : MemL2Quotient Γ μ ψ) : h.toL2 = h'.toL2 ↔ φ =ᵐ[μ] ψ

theorem **coeFn_R** (g : G) (F : L2 Γ μ) : R Γ μ g F =ᵐ[quotMeasure Γ μ] fun q => F (rmul Γ g q)

theorem **lift_R** (g : G) (F : L2 Γ μ) : lift Γ (R Γ μ g F) =ᵐ[μ] fun x => lift Γ F (x * g)

theorem **norm_R** (g : G) (F : L2 Γ μ) : ‖R Γ μ g F‖ = ‖F‖

theorem **MemL2Quotient.mul_right** (h : MemL2Quotient Γ μ φ) (g : G) : MemL2Quotient Γ μ fun x => φ (x * g)

theorem **right_translate_toL2** (h : MemL2Quotient Γ μ φ) (g : G) : R Γ μ g h.toL2 = (h.mul_right g).toL2

theorem **memL2_const** [IsFiniteMeasure (quotMeasure Γ μ)] (c : ℂ) : MemL2Quotient Γ μ fun _ => c

theorem **isIrredSubrep_const** [IsFiniteMeasure (quotMeasure Γ μ)] [NeZero μ] : IsIrredSubrep (R Γ μ) (ℂ • (memL2_const (Γ

theorem **continuous_R** (F : L2 Γ μ) : Continuous fun g => R Γ μ g F

theorem **isUnitaryRep_R** : IsUnitaryRep (R Γ μ)

theorem **isRepOp_Rop** (hf : Integrable f μ) : IsRepOp (R Γ μ) μ f (Rop Γ μ f)

theorem **IsSubrep.Rop_mem** {W : Submodule ℂ (L2 Γ μ)} (hW : IsSubrep (R Γ μ) W) (hf : Integrable f μ) {F : L2 Γ μ} (hF : F ∈ W) : Rop Γ μ f F ∈ W

theorem **adjoint_Rop** [μ.IsOpenPosMeasure] (hf : Continuous f) (hc : HasCompactSupport f) : ContinuousLinearMap.adjoint (Rop Γ μ f) = Rop Γ μ (starFn f)

theorem **isUnitaryRep_RG** [μ.IsHaarMeasure] [μ.IsMulRightInvariant] : IsUnitaryRep (RG V μ)

theorem **leftInv_of_isAutFun** {φ : GA D V → ℂ} (h : IsAutFun V φ) : LeftInv (GQ V) φ

theorem **isVolPair_volPair** (h : ∃ p : Measure (GA D V) × Set (GA D V), IsVolPair D V p.1 p.2) : IsVolPair D V (volPair V).1 (volPair V).2

theorem **muQ_eq** (h : ∃ p : Measure (GA D V) × Set (GA D V), IsVolPair D V p.1 p.2) : muQ V = autMeasure V (volPair V).1

Development.AbelianVarietyTheory.Definitions.Foundations.AltCoeffRho

theorem **altCoeff_rho** {ι : Type} [Fintype ι] [DecidableEq ι] {N : ℕ} (P : MvPolynomial (ι × Fin N) ℝ) : altCoeff (fun _ => rho N) P = Formalism.DegreeFormula.degFl ι N P

Development.AbelianVarietyTheory.Definitions.Foundations.AnalyticOnNhdTsumNpow

theorem **term_mk** {Φ : G → X} (h : LeftInv (P ⊓ Γ) Φ) (g : G) (γ : Γ) : term P Γ Φ g (mk _ γ) = Φ (γ * g)

theorem **term_mul_left** {Φ : G → X} (h : LeftInv (P ⊓ Γ) Φ) (γ : Γ) (g : G) (q : Cosets P Γ) : term P Γ Φ (γ * g) q = term P Γ Φ g (rmulEquiv P Γ γ q)

theorem **leftInv_cosetSum** (h : LeftInv (P ⊓ Γ) Φ) : LeftInv Γ (cosetSum P Γ Φ)

theorem **cosetConv_mul_left** (h : LeftInv (P ⊓ Γ) Φ) {γ : G} (hγ : γ ∈ Γ) : CosetConv P Γ Φ (γ * g) ↔ CosetConv P Γ Φ g

theorem **cosetSum_mul_right** (Φ : G → ℂ) (k g : G) : cosetSum P Γ (fun x => Φ (x * k)) g = cosetSum P Γ Φ (g * k)

theorem **cosetConv_mul_right** (Φ : G → ℂ) (k g : G) : CosetConv P Γ (fun x => Φ (x * k)) g ↔ CosetConv P Γ Φ (g * k)

theorem **cosetSum_eq_tsum** (h : LeftInv (P ⊓ Γ) Φ) {ι : Type} {r : ι → Γ} (hr : Function.Bijective fun i => mk (P.subgroupOf Γ) (r i)) (g : G) : cosetSum P Γ Φ g = ∑' i, Φ (r i * g)

theorem **cosetConv_iff** (h : LeftInv (P ⊓ Γ) Φ) {ι : Type} {r : ι → Γ} (hr : Function.Bijective fun i => mk (P.subgroupOf Γ) (r i)) : CosetConv P Γ Φ g ↔ Summable fun i => ‖Φ (r i * g)‖

theorem **CosetConv.hasSum** (h : CosetConv P Γ Φ g) : HasSum (term P Γ Φ g) (cosetSum P Γ Φ g)

theorem **CosetConv.of_norm_le** (h : CosetConv P Γ Ψ g) {C : ℝ} (e : ∀ x, ‖Φ x‖ ≤ C * ‖Ψ x‖) : CosetConv P Γ Φ g

theorem **CosetConv.add** (h : CosetConv P Γ Φ g) (h' : CosetConv P Γ Ψ g) : CosetConv P Γ (Φ + Ψ) g

theorem **CosetConv.smul** (h : CosetConv P Γ Φ g) (c : ℂ) : CosetConv P Γ (c • Φ) g

theorem **cosetSum_add** (h : CosetConv P Γ Φ g) (h' : CosetConv P Γ Ψ g) : cosetSum P Γ (Φ + Ψ) g = cosetSum P Γ Φ g + cosetSum P Γ Ψ g

theorem **cosetSum_smul** (c : ℂ) (Φ : G → ℂ) (g : G) : cosetSum P Γ (c • Φ) g = c * cosetSum P Γ Φ g

theorem **cosetSum_of_le** (e : Γ ≤ P) (h : LeftInv (P ⊓ Γ) Φ) (g : G) : CosetConv P Γ Φ g ∧ cosetSum P Γ Φ g = Φ g

theorem **cosetSum_indicator** (g : G) : LeftInv (P ⊓ Γ) (indicator ((P ⊓ Γ : Subgroup G) : Set G) (1 : G → ℂ)) ∧ CosetConv P Γ (indicator ((P ⊓ Γ : Subgroup G) : Set G) 1) g ∧ cosetSum P Γ (indicator ((P ⊓ Γ : Subgroup G) : Set G) 1) g = indicator (Γ : Set G) 1 g

theorem **rpow_le_add** {x a b t : ℝ} (hx : 0 < x) (h1 : a ≤ t) (h2 : t ≤ b) : x ^ t ≤ x ^ a + x ^ b

theorem **differentiable_npow** {r : ℝ≥0} (hr : 0 < r) : Differentiable ℂ (npow r)

theorem **analyticOnNhd_tsum_npow** {ι : Type} {f : ι → ℂ} {a : ι → ℝ≥0} (ha : ∀ i, 0 < a i) {σ : ℝ} {s₀ : ℂ} (h : ∀ s : ℂ, σ < s.re → Summable fun i => ‖f i * npow (a i) (s - s₀)‖) : AnalyticOnNhd ℂ (fun s => ∑' i, f i * npow (a i) (s - s₀)) (rightOf σ)

theorem **exists_majorant** {S : Set ℂ} {C : Set G} (hS : IsCompact S) (hσ : S ⊆ rightOf σ) (hC : IsCompact C) : ∃ u : Cosets P Γ → ℝ, Summable u ∧ ∀ q, ∀ s ∈ S, ∀ g ∈ C, ‖term P Γ (Φ s) g q‖ ≤ u q

theorem **cosetConv_of_spherical** {s : ℂ} (h : σ < s.re) (g : G) : CosetConv P Γ (Φ s) g

theorem **continuousOn_cosetSum** [WeaklyLocallyCompactSpace G] : ContinuousOn (fun x : ℂ × G => cosetSum P Γ (Φ x.1) x.2) (rightOf σ ×ˢ univ)

theorem **analyticOnNhd_cosetSum** (hd : ∀ g, DifferentiableOn ℂ (Φ · g) (rightOf σ)) (g : G) : AnalyticOnNhd ℂ (fun s => cosetSum P Γ (Φ s) g) (rightOf σ)

Development.AbelianVarietyTheory.Definitions.Foundations.ApplyEqOfMemAdjoin

theorem **InZl.mem_adjoin** {l n : ℕ} {F : Ω} (h : InZl ε l n F) : F ∈ IntermediateField.adjoin ℚ {zt ε l n}

theorem **IsNTr.iff_isNT** {l n m : ℕ} [NeZero l] {F F' : Ω} (hm : 1 ≤ m) (hmn : m ≤ n) (hF : InZl ε l n F) : IsNTr ε l n m F F' ↔ IsNT ε l n m F F'

theorem **IsNTr.apply_eq_of_mem_adjoin** {z F : Ω} (hF : F ∈ IntermediateField.adjoin ℚ {z}) {s t : Ω ≃ₐ[ℚ] Ω} (h : s z = t z) : s F = t F

theorem **IsNTr.unique** {l n m : ℕ} [NeZero l] {F F' F'' : Ω} (h : IsNTr ε l n m F F') (h' : IsNTr ε l n m F F'') : F' = F''

theorem **IsNTr.add** {l n m : ℕ} [NeZero l] {F F' G G' : Ω} (h : IsNTr ε l n m F F') (h' : IsNTr ε l n m G G') : IsNTr ε l n m (F + G) (F' + G')

theorem **KT.mem_def** (X : GlobalTheory.KZ.Tower Ω ε) (L : ℕ) (f : X.K) : f ∈ KT X L ↔ X.sl (ofSL (ModularGroup.T ^ L)) f = f

theorem **IsCusp.isPlaceIn** {A : ValuationSubring Ω} {T : GlobalTheory.K5.Tower p Ωl} {j : Ω →+* Ωl} (h : IsCusp ε A T j) : IsPlaceIn ε A T j

theorem **IsPlaceIn.unique** {A A' : ValuationSubring Ω} {T T' : GlobalTheory.K5.Tower p Ωl} {j : Ω →+* Ωl} (h : IsPlaceIn ε A T j) (h' : IsPlaceIn ε A' T' j) : A = A'

theorem **IsPlaceIn.mem_nonunits** (hc : IsClo p Ωl) {A : ValuationSubring Ω} {T : GlobalTheory.K5.Tower p Ωl} {j : Ω →+* Ωl} (h : IsPlaceIn ε A T j) : ((p : ℕ) : Ω) ∈ A.nonunits

theorem **IsPlaceIn.stab** {A : ValuationSubring Ω} {T : GlobalTheory.K5.Tower p Ωl} {j : Ω →+* Ωl} (h : IsPlaceIn ε A T j) {g : Gal p Ωl} {s : Ω ≃ₐ[ℚ] Ω} (hs : ∀ x, g (j x) = j (s x)) : Stab A s

theorem **IsIo.isExpIn** (h : IsIo X A T j n ψ io) (hj : IsPlaceIn ε A T j) (hψ : IsReb T n ψ) : IsExpIn X A T j n ψ io

theorem **IsExpIn.isIo** (h : IsExpIn X A T j n ψ io) (hg : GalTr X T j n io) : IsIo X A T j n ψ io

theorem **IsExpIn.par** (h : IsExpIn X A T j n ψ io) : ψ (qO p) = T.qq n

theorem **GalTr.coeff_sl_gal** (X : GlobalTheory.KZ.Tower Ω ε) (s : Ω ≃ₐ[ℚ] Ω) (b : ℕ) (y : X.K) (r : ℚ) : (X.exp (X.sl (ofSL (ModularGroup.T ^ b)) (X.gal s y))).coeff r = eOm ε (r * b) * s ((X.exp y).coeff r)

Development.AbelianVarietyTheory.Definitions.Foundations.AuxiliaryFrameNormalizedSpaces

theorem **level_split_E** : ∀ p ∈ B.host.levelPrimes, (placesOver B.host.E p).ncard = Module.finrank ℚ B.host.E

theorem **g_split** : ∀ k, (placesOver B.host.E (B.host.g k)).ncard = Module.finrank ℚ B.host.E

theorem **pde_split** : (Ideal.primesOver (Ideal.span {(B.host.pde : ℤ)}) (𝒪 B.host.E)).ncard = Module.finrank ℚ B.host.E ∧ (Ideal.primesOver (Ideal.span {(B.host.pde' : ℤ)}) (𝒪 B.host.E)).ncard = Module.finrank ℚ B.host.E

theorem **spacesNorm_T** : (spacesNorm R cm T hT pz hpz).T = T

theorem **spacesNorm_Y_sig** (j : ℕ) (τ : E →+* ℂ) : ((spacesNorm R cm T hT pz hpz).Y j).sig τ = sigYD R j

theorem **spacesNorm_z_norm** : ((spacesNorm R cm T hT pz hpz).X R.m).hT (spacesNorm R cm T hT pz hpz).z (spacesNorm R cm T hT pz hpz).z = (T : E)

theorem **diag_pair** {N n c : ℕ} (h : n + 2 * c ≤ N) (a b a' b' : E) (i j : Fin c) : diagForm N (cut N (n + c)) (Pi.single ((n + i, by omega) : Fin N) a + Pi.single ((n + c + i, by omega) : Fin N) b) (Pi.single ((n + j, by omega) : Fin N) a' + Pi.single ((n + c + j, by omega) : Fin N) b') = if i = j then star a * a' − star b * b' else 0

theorem **auxiliary_frame_normalized_spaces** (j : ℕ) (hj : j < R.m) : HasHyperbolicAuxiliaryFrame (spacesNorm R cm T hT pz hpz) j

theorem **exists_finset_capitulation** (E : Type) [Field E] [NumberField E] (τ : E →+* ℂ) : ∃ S : Finset ℂ, (∀ z ∈ S, IsAlgebraic ℚ z) ∧ ∀ (K : IntermediateField ℚ ℂ) [FiniteDimensional ℚ K], (∀ z ∈ S, z ∈ K) → (∀ x, τ x ∈ K) → ∀ a : Ideal (𝒪 E), ∃ β : K, Submodule.span (𝒪 K) {β} = Submodule.span (𝒪 K) {x : K | ∃ a ∈ a, (x : ℂ) = τ (a : E)}

Development.AbelianVarietyTheory.Definitions.Foundations.CanonicalExponents

theorem **deg_span** {ι : Type} [Fintype ι] [DecidableEq ι] {b : Basis ι K V} {X : InfinitePlace K → Matrix ι ι ℂ} {N : (K →+* ℂ) → V → ℝ} (h : IsGram b X N) {j : ℕ} (hj : 0 < j) (e : Fin j → V) (he : LinearIndependent K e) : deg N (Submodule.span (𝒪 K) (Set.range e)) = -(1 / (2 * (Formalism.HN.nPl K : ℝ))) * Real.log (∏ v, (Formalism.HN.gramPl b X v e).det.re)

theorem **deg_two** : deg (fun (σ : K →+* ℂ) (y : K) => ‖σ y‖) (Submodule.span (𝒪 K) {(2 : K)}) = -Real.log 2

theorem **bar_gap** (a : (E →+* ℂ) × ℕ → ℤ) (m : ℤ) (τ : Psi H) (hp : 0 < Dw.p) (hq : 0 < Dw.q) : weightOf Dw a m τ (Dw.p - 1, by omega) - weightOf Dw a m τ (Dw.p, by omega) = a (τ.1, Dw.p - 1) + a (conjEmb τ.1, Dw.q - 1) + 2 * m

theorem **small_antitone** (la : Weight Dw) (m : ℤ) (τ : Psi H) (h : ∀ i j : Fin (Dw.p + Dw.q), Dw.p ≤ i → i ≤ j → la τ j ≤ la τ i) (i j : ℕ) (hij : i ≤ j) (hj : j < Dw.q) : (wtOf Dw la m).1 (conjEmb τ.1, j) ≤ (wtOf Dw la m).1 (conjEmb τ.1, i)

theorem **canonical_exponents** (τ : Psi H) (i j : ℕ) (hi : i < Dw.p) (hj : j < Dw.q) : (wtOf Dw (fun _ k => if (k : ℕ) < Dw.p then (Dw.q : ℤ) else -(Dw.p : ℤ)) 0).1 (τ.1, i) = Dw.q ∧ (wtOf Dw (fun _ k => if (k : ℕ) < Dw.p then (Dw.q : ℤ) else -(Dw.p : ℤ)) 0).1 (conjEmb τ.1, j) = Dw.p

theorem **modelWt_eq_sub** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (G : WTZ E) (hG : Even G.2) : ∃ a n a' n', Dom a n ∧ Dom a' n' ∧ ∀ k, laU U a' n' k = modelWt U G k + laU U a n k

theorem **chi_modelWt** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (h : ∀ a n, Dom a n → ∀ k, ∀ γ ∈ (U.X k).Γ, (U.X k).χ (laU U a n k) γ = 1) (G : WTZ E) (hG : Even G.2) (k : U.Comp) : ∀ γ ∈ (U.X k).Γ, (U.X k).χ (modelWt U G k) γ = 1

Development.AbelianVarietyTheory.Definitions.Foundations.CdScaleTieIff

theorem **trueVol_pos** (cm : CMData F E) {I : Type} [Fintype I] (τ : I → (E →+* ℂ)) (cd : CDHaar) (W : HSpace E) : 0 < trueVol cm τ cd W

theorem **trueVol_spec** (cm : CMData F E) {I : Type} [Fintype I] (τ : I → (E →+* ℂ)) (cd : CDHaar) (W : HSpace E) (h : ∃ x : ℝ, 0 < x ∧ VolIs cm τ cd W x) : VolIs cm τ cd W (trueVol cm τ cd W)

theorem **trueScale_pos** (cd : CDHaar) (n : ℕ) (ε : Fin n → Bool) : 0 < trueScale cd n ε

theorem **volTieG_iff** (D : D7) : Formalism.L2.Theta.VolTieG D ↔ ∀ W : HSpace D.host.E, W.SigOK → VolIs D.sp.cm (Formalism.BasicNotions.TH.tauE D) D.cd W (D.vol W)

theorem **cdScaleTie_iff** (D : D7) : cdScaleTie D ↔ CdScaleIs D.cd D.s.cdScale

Development.AbelianVarietyTheory.Definitions.Foundations.ClassMassBotBotOfInfinite

theorem **card_fibre_mul** (hle : K' ≤ K) (g : G) (k : K) : Nat.card {t : K ⧸ K'.subgroupOf K // (DoubleCoset.mk Γ K' (g * (t.out : K)) : ClassSet Γ K') = DoubleCoset.mk Γ K' (g * k)} * Nat.card (stabAt Γ K' (g * k)) = Nat.card (stabAt Γ K g)

theorem **classMass_of_le** (h : FiniteClasses Γ K) (hle : K' ≤ K) (hi : K'.relIndex K ≠ 0) : classMass Γ K' = K'.relIndex K * classMass Γ K

theorem **classMass_finite** [Finite G] (Γ K : Subgroup G) : classMass Γ K = (Nat.card G : ℝ) / (Nat.card Γ * Nat.card K)

theorem **classMass_swap** : classMass (Subgroup.zpowers (Equiv.swap (0 : Fin 3) 1)) (Subgroup.zpowers (Equiv.swap (0 : Fin 3) 1)) = 3 / 2

theorem **classMass_bot_bot_of_infinite** [Infinite G] : classMass (⊥ : Subgroup G) ⊥ = 0

theorem **unitaryMass_of_le** (hσ : IsCMExt F E σ) (hJ : IsHermND σ J) (hd : IsTotDef σ J) {K K' : Subgroup (HA (Af F) σ J)} (hc : IsCompact (K : Set (HA (Af F) σ J))) (ho' : IsOpen (K' : Set (HA (Af F) σ J))) (hle : K' ≤ K) : K'.relIndex K ≠ 0 ∧ unitaryMass σ J K' = K'.relIndex K * unitaryMass σ J K

theorem **exists_section** {π : HA (AF F) σ J →* HA (Af F) σ J} (h : IsFinProj σ J π) : ∃ s : HA (Af F) σ J → HA (AF F) σ J, Continuous s ∧ ∀ b, π (s b) = b

theorem **isOpen_of_comap** {π : HA (AF F) σ J →* HA (Af F) σ J} (h : IsFinProj σ J π) {L : Subgroup (HA (Af F) σ J)} (ho : IsOpen (L.comap π : Set (HA (AF F) σ J))) : IsOpen (L : Set (HA (Af F) σ J))

theorem **two_le_stabOrder_std** (hσ : IsCMExt F E σ) (hJ : IsHermND σ J) (hd : IsTotDef σ J) (hM : 0 < M) (c : ClassSet (ratIn (Af F) σ J) (stdLevel σ J)) : 2 ≤ stabOrder _ _ c

theorem **unitaryMass_map_eq** {D : D7} (V : HSp D) {π : GA D V →* HA (Af D.host.F) D.sp.cm.sig V.gram} (h : IsFinProj D.sp.cm.sig V.gram π) (K : Subgroup (GA D V)) : unitaryMass D.sp.cm.sig V.gram (K.map π) = classMass (GQ V) (K ⊔ GinfC V)

theorem **finiteClasses_record** {D : D7} (V : HSp D) (hσ : IsCMExt D.host.F D.host.E D.sp.cm.sig) (hJ : IsHermND D.sp.cm.sig V.gram) (hd : IsTotDef D.sp.cm.sig V.gram) {K : Subgroup (GA D V)} (hc : IsCompact (K : Set (GA D V))) (ho : IsOpen ((K ⊔ GinfC V : Subgroup (GA D V)) : Set (GA D V))) : FiniteClasses (GQ V) (K ⊔ GinfC V)

Development.AbelianVarietyTheory.Definitions.Foundations.ClassMassEqSumOfBijective

@[simp] theorem **stabOrder_mk** (Γ K : Subgroup G) (g : G) : stabOrder Γ K (DoubleCoset.mk Γ K g) = Nat.card (stabAt Γ K g)

theorem **one_mem_stabAt** (Γ K : Subgroup G) (g : G) : (1 : Γ) ∈ stabAt Γ K g

theorem **FiniteClasses.stabOrder_pos** (h : FiniteClasses Γ K) (c : ClassSet Γ K) : 0 < stabOrder Γ K c

theorem **classMass_eq_sum** [Fintype (ClassSet Γ K)] : classMass Γ K = ∑ c : ClassSet Γ K, ((stabOrder Γ K c : ℝ))⁻¹

theorem **classMass_eq_sum_of_bijective** {ι : Type*} [Fintype ι] (g : ι → G) (hg : Function.Bijective fun i => (DoubleCoset.mk Γ K (g i) : ClassSet Γ K)) : classMass Γ K = ∑ i, ((Nat.card (stabAt Γ K (g i)) : ℝ))⁻¹

theorem **classMass_eq_finsum_out** (Γ K : Subgroup G) : classMass Γ K = ∑ᶠ c : ClassSet Γ K, ((Nat.card (stabAt Γ K c.out) : ℝ))⁻¹

theorem **FiniteClasses.inv_card_le** (h : FiniteClasses Γ K) (g : G) : ((Nat.card (stabAt Γ K g) : ℝ))⁻¹ ≤ classMass Γ K

theorem **FiniteClasses.classMass_pos** (h : FiniteClasses Γ K) : 0 < classMass Γ K

theorem **classMass_of_isNeat** [Finite (ClassSet Γ K)] (h : IsNeat Γ K) : classMass Γ K = Nat.card (ClassSet Γ K)

theorem **FiniteClasses.of_le** (h : FiniteClasses Γ K) (hle : K' ≤ K) (hi : K'.relIndex K ≠ 0) : FiniteClasses Γ K'

theorem **FiniteClasses.of_ge** (h : FiniteClasses Γ K') (hle : K' ≤ K) (hi : K'.relIndex K ≠ 0) : FiniteClasses Γ K

theorem **relIndex_ne_zero_of_isOpen** [TopologicalSpace G] [IsTopologicalGroup G] (hK : IsCompact (K : Set G)) (hK' : IsOpen (K' : Set G)) : K'.relIndex K ≠ 0

theorem **FiniteClasses.of_isCompact_isOpen** [TopologicalSpace G] [IsTopologicalGroup G] (h : FiniteClasses Γ K) (hc : IsCompact (K : Set G)) (ho : IsOpen (K : Set G)) (hc' : IsCompact (K' : Set G)) (ho' : IsOpen (K' : Set G)) : FiniteClasses Γ K'

theorem **bijective_classMap** (hπ : Function.Surjective π) : Function.Bijective fun c : ClassSet Γ (L.comap π) => (DoubleCoset.mk (Γ.map π) L (π c.out) : ClassSet (Γ.map π) L)

theorem **nonempty_stabAt_equiv** (hΓ : Set.InjOn π Γ) (g : G) : Nonempty (stabAt Γ (L.comap π) g ≃ stabAt (Γ.map π) L (π g))

theorem **classMass_comap** (hπ : Function.Surjective π) (hΓ : Set.InjOn π Γ) : classMass Γ (L.comap π) = classMass (Γ.map π) L

theorem **finiteClasses_comap** (hπ : Function.Surjective π) (hΓ : Set.InjOn π Γ) : FiniteClasses Γ (L.comap π) ↔ FiniteClasses (Γ.map π) L

theorem **FiniteClasses.measure_toReal** [MeasurableSpace G] [MeasurableMul G] [Countable Γ] (h : FiniteClasses Γ K) (μ : Measure G) [μ.IsMulLeftInvariant] (hK : MeasurableSet (K : Set G)) (hK1 : μ K = 1) {𝓕 : Set G} (h𝓕 : IsFundamentalDomain Γ 𝓕 μ) : (μ 𝓕).toReal = classMass Γ K

theorem **swapTens_sig** (x : R ⊗[F] E) : swapTens F E R (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ x) = Formalism.UG.sigR σ R (swapTens F E R x)

theorem **exists_swapPts** : ∃ e : HA R σ J ≃* UPts σ J R, ∀ g, (e g : GL (Fin M) (E ⊗[F] R)) = bc (swapTens F E R : R ⊗[F] E →+* E ⊗[F] R) (HA.mat R σ J g)

theorem **coe_swapPts** (g : HA R σ J) : (swapPts R σ J g : GL (Fin M) (E ⊗[F] R)) = bc (swapTens F E R : R ⊗[F] E →+* E ⊗[F] R) (HA.mat R σ J g)

theorem **map_swapPts_rat** : (ratIn R σ J).map (swapPts R σ J : HA R σ J →* UPts σ J R) = (ratPts σ J R).subgroupOf (UPts σ J R)

theorem **unitaryMass_comap_swapPts** (L : Subgroup (UPts σ J (Af F))) : unitaryMass σ J (L.comap (swapPts (Af F) σ J : HA (Af F) σ J →* UPts σ J (Af F))) = classMass ((ratPts σ J (Af F)).subgroupOf (UPts σ J (Af F))) L

theorem **mem_stabGL_stdLat** {A : Type} [CommRing A] (S : Subring A) {n : Type} [Fintype n] [DecidableEq n] (g : GL n A) : g ∈ stabGL (stdLat (n

theorem **mem_stdLevel** (g : HA (Af F) σ J) : g ∈ stdLevel σ J ↔ ∀ i j, (HA.mat (Af F) σ J g).1 i j ∈ intTens F E ∧ ((HA.mat (Af F) σ J g)⁻¹).1 i j ∈ intTens F E

theorem **intTens_open_compact** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] : IsOpen (intTens F E : Set (Af F ⊗[F] E)) ∧ ∃ C : Set (Af F ⊗[F] E), IsCompact C ∧ ∀ w ∈ intTens F E, w ∈ C

theorem **topGroup_fin** : IsTopologicalGroup (HA (Af F) σ J)

theorem **isOpen_stdLevel** : IsOpen (stdLevel σ J : Set (HA (Af F) σ J))

theorem **isCompact_stdLevel** (hJ : J.det ≠ 0) : IsCompact (stdLevel σ J : Set (HA (Af F) σ J))

theorem **finiteClasses_unitary** (hσ : IsCMExt F E σ) (hJ : IsHermND σ J) (hd : IsTotDef σ J) {K : Subgroup (HA (Af F) σ J)} (hc : IsCompact (K : Set (HA (Af F) σ J))) (ho : IsOpen (K : Set (HA (Af F) σ J))) : FiniteClasses (ratIn (Af F) σ J) K

theorem **finiteClasses_lattice** (hσ : IsCMExt F E σ) (hJ : IsHermND σ J) (hd : IsTotDef σ J) (Λ : Submodule (𝒪 E) (Fin M → E)) (hΛ : Module.Finite (𝒪 E) Λ) (hsp : Submodule.span E (Λ : Set (Fin M → E)) = ⊤) : FiniteClasses (ratIn (Af F) σ J) (((levelAfL σ J Λ).subgroupOf (UPts σ J (Af F))).comap (swapPts (Af F) σ J : HA (Af F) σ J →* UPts σ J (Af F)))

theorem **unitaryMass_pos** (hσ : IsCMExt F E σ) (hJ : IsHermND σ J) (hd : IsTotDef σ J) {K : Subgroup (HA (Af F) σ J)} (hc : IsCompact (K : Set (HA (Af F) σ J))) (ho : IsOpen (K : Set (HA (Af F) σ J))) : 0 < unitaryMass σ J K

theorem **parts_spec** : (∀ g : GL (Fin M) (AF F ⊗[F] E), g ∈ HR (AF F) σ J ↔ bc (Algebra.TensorProduct.map (fstA F) (AlgHom.id F E)).toRingHom g ∈ HR (Ainf F) σ J ∧ bc (Algebra.TensorProduct.map (sndA F) (AlgHom.id F E)).toRingHom g ∈ HR (Af F) σ J) ∧ ∀ (a : GL (Fin M) (Ainf F ⊗[F] E)) (b : GL (Fin M) (Af F ⊗[F] E)), ∃ g : GL (Fin M) (AF F ⊗[F] E), bc (Algebra.TensorProduct.map (fstA F) (AlgHom.id F E)).toRingHom g = a ∧ bc (Algebra.TensorProduct.map (sndA F) (AlgHom.id F E)).toRingHom g = b

theorem **existsUnique_isFinProj** : ∃! π, IsFinProj σ J π

theorem **IsFinProj.surjective** (h : IsFinProj σ J π) : Function.Surjective π

theorem **IsFinProj.injOn** (h : IsFinProj σ J π) : Set.InjOn π (ratIn (AF F) σ J)

theorem **IsFinProj.map_ratIn** (h : IsFinProj σ J π) : (ratIn (AF F) σ J).map π = ratIn (Af F) σ J

theorem **IsFinProj.continuous** (h : IsFinProj σ J π) : Continuous π

theorem **IsFinProj.unitaryMass_eq** (h : IsFinProj σ J π) (K : Subgroup (HA (Af F) σ J)) : unitaryMass σ J K = classMass (ratIn (AF F) σ J) (K.comap π)

theorem **IsFinProj.measure_toReal** (h : IsFinProj σ J π) (hσ : IsCMExt F E σ) (hJ : IsHermND σ J) (hd : IsTotDef σ J) {K : Subgroup (HA (Af F) σ J)} (hc : IsCompact (K : Set (HA (Af F) σ J))) (ho : IsOpen (K : Set (HA (Af F) σ J))) (μ : Measure (HA (AF F) σ J)) [μ.IsMulLeftInvariant] (hK1 : μ (K.comap π) = 1) {𝓕 : Set (HA (AF F) σ J)} (h𝓕 : IsFundamentalDomain (HFinIn (AF F) σ J) 𝓕 μ) : (μ 𝓕).toReal = unitaryMass σ J K

theorem **classMass_top_top** (G : Type) [Group G] : classMass (⊤ : Subgroup G) ⊤ = ((Nat.card G : ℝ))⁻¹

theorem **classMass_bot** {G : Type} [Group G] (K : Subgroup G) [K.FiniteIndex] : classMass (⊥ : Subgroup G) K = K.index

theorem **finiteClasses_of_finite** {G : Type} [Group G] [Finite G] (Γ K : Subgroup G) : FiniteClasses Γ K

Development.AbelianVarietyTheory.Definitions.Foundations.CohomologyInputsOfUnitary

theorem **isConj_iff** (s : G ℚ Ω) : IsConj s ↔ IsCC s

theorem **vanishOrd_of** (h : X.LemmaKatoW) : VanishOrd

theorem **cohomology_inputs_of_unitary** (h : U3.Inp) : Inp0

theorem **om_mulVec** (A : Matrix (Fin 2) (Fin 2) (Qbar l)) (x y : Fin 2 → Qbar l) : om (A *ᵥ x) (A *ᵥ y) = A.det * om x y

theorem **om_equiv** (W : R2 Ω l) (s : G ℚ Ω) (x y : Fin 2 → Qbar l) : om (mat W s *ᵥ x) (mat (dual1 W) s *ᵥ y) = eps l s * om x y

theorem **om_add_left** (x y z : Fin 2 → Qbar l) : om (x + y) z = om x z + om y z

theorem **om_add_right** (x y z : Fin 2 → Qbar l) : om z (x + y) = om z x + om z y

theorem **om_sub_left** (x y z : Fin 2 → Qbar l) : om (x − y) z = om x z − om y z

theorem **mat_mul** (W : R2 Ω l) (s t : G ℚ Ω) : mat W (s ∗ t) = mat W s ∗ mat W t

theorem **cup_coc** (W : R2 Ω l) (a b : G ℚ Ω → Fin 2 → Qbar l) (ha : ∀ s t, a (s ∗ t) = a s + mat W s ∗ᵥ a t) (hb : ∀ s t, b (s ∗ t) = b s + mat (dual1 W) s ∗ᵥ b t) (r s t : G ℚ Ω) : eps l r ∗ cup W a b s t + cup W a b r (s ∗ t) = cup W a b (r ∗ s) t + cup W a b r s

theorem **cup_cob** (W : R2 Ω l) (a : Fin 2 → Qbar l) (z : G ℚ Ω → Fin 2 → Qbar l) (hz : ∀ s t, z (s ∗ t) = z s + mat (dual1 W) s ∗ᵥ z t) (s t : G ℚ Ω) : cup W (fun s => mat W s ∗ᵥ a − a) z s t = eps l s ∗ om a (z t) − om a (z (s ∗ t)) + om a (z s)

theorem **ht_cob** (W : R2 Ω l) (a : Fin 2 → Qbar l) (z : G ℚ Ω → Fin 2 → Qbar l) (s : G ℚ Ω) : ht W (fun s => mat W s ∗ᵥ a − a) z (fun s => om a (z s)) s = 0

theorem **gr_of_triv** (W : R2 Ω l) (A : ValuationSubring Ω) (hW : ∀ s ∈ Dec (F

theorem **strZero_of_grZero** (W : R2 Ω l) (A : ValuationSubring Ω) (hW : ∀ s ∈ Dec (F

theorem **eps_eq** (s : G ℚ Ω) : eps l s = algebraMap ℤ_[l] (Qbar l) (cycZ ℚ Ω l s : ℤ_[l])

theorem **cont_int** : Continuous (algebraMap ℤ_[l] (Qbar l))

theorem **cg_refl** (n : ℕ) (a : Qbar l) : Cg n a a

theorem **cycZ_ine** [IsAlgClosure ℚ Ω] {v : Pl} (hl : ((l : ℕ) : 𝒪 ℚ) ∉ v.asIdeal) {A : ValuationSubring Ω} (hA : IsPlaceOver A v) {σ : G ℚ Ω} (hσ : σ ∈ Ine (F

theorem **cyclotomic_two_cocycle** [IsAlgClosure ℚ Ω] (c : Coef l) (n : ℕ) (S : Finset Pl) (hS : ContainsAbove S l) : Coc2 c n S (fun s _ : G ℚ Ω => eps l s) where

theorem **cyclotomic_two_coboundary** (c : Coef l) (n : ℕ) (H : Subgroup (G ℚ Ω)) : Cob2 c n H (fun s _ : G ℚ Ω => eps l s)

theorem **tame_zero** (c : Coef l) (W : R2 Ω l) (y : G ℚ Ω → Fin 2 → Qbar l) (n : ℕ) (w : Pl) (ψ : G ℚ Ω →∗ Multiplicative (ZMod (l ^ n))) (h : ∀ (B : ValuationSubring Ω) (σ : G ℚ Ω), IsPlaceOver B w → IsFrob B (normV w) σ → Cg n 0 (mat W σ − 1).det) : Tame c W y n w ψ

theorem **nu** (u : ℤ_[l]ˣ) : ‖((u : ℤ_[l]) : ℚ_[l])‖ = 1

theorem **lgE_mul** (s t : G ℚ Ω) : lgE l (s ∗ t) = lgE l s + lgE l t

theorem **es_one** [IsAlgClosure ℚ Ω] (A : ValuationSubring Ω) (hA : OverL l A) : ES (1 : R2 Ω l) A

theorem **deriv_rel** (W : R2 Ω l) (n : ℕ) (ψ : G ℚ Ω →∗ Multiplicative (ZMod (l ^ n))) (Y : G ℚ Ω → ZMod (l ^ n) → Fin 2 → Qbar l) (hY : ∀ s t x, Y (s ∗ t) x = Y s x + mat W s ∗ᵥ Y t (x − (ψ s).toAdd)) (s t : G ℚ Ω) : ∃ k : ZMod (l ^ n) → ℕ, ∑ᶠ x, (x.val : Qbar l) • Y (s ∗ t) x + ((l : Qbar l) ^ n) • ∑ᶠ x, (k x : Qbar l) • (mat W s ∗ᵥ Y t x) = ∑ᶠ x, (x.val : Qbar l) • Y s x + mat W s ∗ᵥ (∑ᶠ x, (x.val : Qbar l) • Y t x) + (((ψ s).toAdd).val : Qbar l) • (mat W s ∗ᵥ ∑ᶠ x, Y t x)

Development.AbelianVarietyTheory.Definitions.Foundations.ConstTermOnMulLeftOfNormalizer

theorem **leftInv_integrand** (hφ : LeftInv Γ φ) (x : G) : LeftInv (Γ.subgroupOf N) fun n : N => φ (n ∗ x)

theorem **probMeasure_eq** (μ : Measure N) [μ.IsHaarMeasure] : probMeasure Γ N = (quotMeasure (Γ.subgroupOf N) μ univ)⁻¹ • quotMeasure (Γ.subgroupOf N) μ

theorem **constTermOn_eq** [μ.IsHaarMeasure] (h : IsFundamentalDomain (Γ.subgroupOf N) 𝓕 μ) (hφ : LeftInv Γ φ) (x : G) : constTermOn Γ N φ x = (μ.real 𝓕)⁻¹ • ∫ n in 𝓕, φ (n ∗ x) ∂μ

theorem **constTermOn_eq_of_measure_eq_one** [μ.IsHaarMeasure] (h : IsFundamentalDomain (Γ.subgroupOf N) 𝓕 μ) (h1 : μ 𝓕 = 1) (hφ : LeftInv Γ φ) (x : G) : constTermOn Γ N φ x = ∫ n in 𝓕, φ (n ∗ x) ∂μ

theorem **constTermOn_mul_right** (φ : G → ℂ) (x g : G) : constTermOn Γ N (fun y => φ (y ∗ g)) x = constTermOn Γ N φ (x ∗ g)

theorem **constTermOn_smul** (c : ℂ) (φ : G → ℂ) (x : G) : constTermOn Γ N (c • φ) x = c ∗ constTermOn Γ N φ x

theorem **constTermOn_mul_left_of_normalizer** (hφ : LeftInv Γ φ) {γ : G} (hγ : γ ∈ Γ) (hN : ∀ n, n ∈ N ↔ γ ∗ n ∗ γ⁻¹ ∈ N) (x : G) : constTermOn Γ N φ (γ ∗ x) = constTermOn Γ N φ x

theorem **integrableOn** [μ.IsHaarMeasure] (h : IsFundamentalDomain (Γ.subgroupOf N) 𝓕 μ) (hφ : LeftInv Γ φ) (hφc : Continuous φ) (x : G) : IntegrableOn (fun n : N => φ (n ∗ x)) 𝓕 μ

theorem **measure_ne_top** [μ.IsHaarMeasure] (h : IsFundamentalDomain (Γ.subgroupOf N) 𝓕 μ) : μ 𝓕 ≠ ⊤

theorem **isProbabilityMeasure** : IsProbabilityMeasure (probMeasure Γ N)

theorem **integrable** (hφ : LeftInv Γ φ) (hφc : Continuous φ) (x : G) : Integrable (fun q : Quot (Γ.subgroupOf N) => φ ((q.out : N) ∗ x)) (probMeasure Γ N)

theorem **constTermOn_const** (c : ℂ) (x : G) : constTermOn Γ N (fun _ => c) x = c

theorem **constTermOn_add** (hφ : LeftInv Γ φ) (hφc : Continuous φ) (hψ : LeftInv Γ ψ) (hψc : Continuous ψ) (x : G) : constTermOn Γ N (φ + ψ) x = constTermOn Γ N φ x + constTermOn Γ N ψ x

theorem **constTermOn_eq_zero_iff** [μ.IsHaarMeasure] (h : IsFundamentalDomain (Γ.subgroupOf N) 𝓕 μ) (hφ : LeftInv Γ φ) (x : G) : constTermOn Γ N φ x = 0 ↔ ∫ n in 𝓕, φ (n ∗ x) ∂μ = 0

theorem **isMulRightInvariant** (μ : Measure N) [μ.IsHaarMeasure] : μ.IsMulRightInvariant

theorem **constTermOn_mul_left** (hφ : LeftInv Γ φ) {n : G} (hn : n ∈ N) (x : G) : constTermOn Γ N φ (n ∗ x) = constTermOn Γ N φ x

theorem **continuous_constTermOn** [LocallyCompactSpace G] (hφ : LeftInv Γ φ) (hφc : Continuous φ) : Continuous (constTermOn Γ N φ)

Development.AbelianVarietyTheory.Definitions.Foundations.ConstantOneNotCuspidalOfIsotropic

theorem **exists_lintegral_compact_le** [μ.IsMulLeftInvariant] {C : Set G} (hC : IsCompact C) : ∃ m : ℕ, ∀ ψ : G → ℂ, LeftInv Γ ψ → Continuous ψ → ∫⁻ x in C, ‖ψ x‖ₑ ∂μ ≤ m ∗ ∫⁻ q, ‖descend Γ ψ q‖ₑ ∂quotMeasure Γ μ

theorem **exists_measure_le** [LocallyCompactSpace N] [μ.IsMulLeftInvariant] [SFinite μ] (hc : Cocompact Γ N) {U : Set G} (hU : IsCompact U) : ∃ m : ℕ, ∀ ψ : G → ℂ, LeftInv Γ ψ → Continuous ψ → (∀ x, constTermOn Γ N ψ x = 1) → μ U ≤ m ∗ ∫⁻ q, ‖descend Γ ψ q‖ₑ ∂quotMeasure Γ μ

theorem **constant_one_L2_ne_zero** : constantOneDividedPowerChartLift V μ ≠ 0

theorem **constant_one_not_cuspidal_of_isotropic** (h : 0 < wittIndex V) : constantOneDividedPowerChartLift V μ ∉ L2cusp V μ

theorem **constant_one_cuspidal_iff_anisotropic** : constantOneDividedPowerChartLift V μ ∈ L2cusp V μ ↔ wittIndex V = 0

theorem **L2cusp_plane_ne_top** (μ : Measure (GA D (plane D))) [μ.IsHaarMeasure] : L2cusp (plane D) μ ≠ ⊤

theorem **L2cusp_ne_bot** (h : wittIndex V = 0) : L2cusp V μ ≠ ⊥

theorem **wittIndex_line** (D : D7) : wittIndex (line D) = 0

theorem **isCuspAE_of_aniso** (h : wittIndex V = 0) (F : DividedPowerChartLift V μ) : IsCuspAE V μ F

theorem **rank_minusHyp_plane** (D : D7) : (minusHyp (plane D) 1 (wittIndex_plane D).ge).rank = 0

theorem **isIsoSub_four** (D : D7) : IsIsoSub (four D) (i

theorem **k_columnFlag_four** (D : D7) : (columnFlag (isIsoSub_four D)).k = 2

theorem **wittIndex_four** (D : D7) : wittIndex (four D) = 2

Development.AbelianVarietyTheory.Definitions.Foundations.ContinuousOfMemCcSmooth

theorem **indicator_mem_SB** {X : Type*} [TopologicalSpace X] [T2Space X] {U : Set X} (ho : IsOpen U) (hc : IsCompact U) : U.indicator (1 : X → ℂ) ∈ SB X

theorem **comp_mem_SB** {X Y : Type*} [TopologicalSpace X] [TopologicalSpace Y] {φ : Y → ℂ} (h : φ ∈ SB Y) (e : X ≃ₜ Y) : φ ∘ e ∈ SB X

theorem **starFn_mem_SB** (h : f ∈ SB G) : starFn f ∈ SB G

theorem **exists_biinvariant** (h : f ∈ SB G) : ∃ U : Subgroup G, IsOpen (U : Set G) ∧ ∀ k ∈ U, ∀ g, f (k * g) = f g ∧ f (g * k) = f g

theorem **conv_eq_zero** {y : G} (h : y ∉ tsupport f * tsupport f') : conv μ f f' y = 0

theorem **hasCompactSupport_conv** (hc : HasCompactSupport f) (hc' : HasCompactSupport f') : HasCompactSupport (conv μ f f')

theorem **conv_mem_SB** (h : f ∈ SB G) (h' : f' ∈ SB G) : conv μ f f' ∈ SB G

theorem **starFn_conv** [μ.IsMulLeftInvariant] (f f' : G → ℂ) : starFn (conv μ f f') = conv μ (starFn f') (starFn f)

theorem **conv_assoc** [μ.IsMulLeftInvariant] (hf : Continuous f) (hc : HasCompactSupport f) (hf' : Continuous f') (hc' : HasCompactSupport f') (hf'' : Continuous f'') : conv μ (conv μ f f') f'' = conv μ f (conv μ f' f'')

theorem **typeIdem_coe** (k : K) : typeIdem μ r τ k = (Module.finrank ℂ W : ℂ) * (μ.real K : ℂ)⁻¹ * conj (τ.character (r k))

theorem **typeIdem_of_notMem** {g : G} (h : g ∉ K) : typeIdem μ r τ g = 0

theorem **typeIdem_mul_ker** {k : K} (hk : r k = 1) (g : G) : typeIdem μ r τ (k * g) = typeIdem μ r τ g ∧ typeIdem μ r τ (g * k) = typeIdem μ r τ g

theorem **starFn_typeIdem** [Finite Q] [FiniteDimensional ℂ W] : starFn (typeIdem μ r τ) = typeIdem μ r τ

theorem **typeIdem_mem_SB** [TopologicalSpace G] [IsTopologicalGroup G] (ho : IsOpen (K : Set G)) (hc : IsCompact (K : Set G)) (hr : IsOpen (r.ker : Set K)) : typeIdem μ r τ ∈ SB G

theorem **mem_ccSmooth** : f ∈ CcSmooth c ↔ HasCompactSupport f ∧ ∃ Φ : X → ℂ, ContDiff ℝ ∞ Φ ∧ ∀ g, f g = Φ (c g)

theorem **continuous_of_mem_ccSmooth** (hc : Continuous c) (h : f ∈ CcSmooth c) : Continuous f

theorem **ccSmooth_le** {X' : Type*} [NormedAddCommGroup X'] [NormedSpace ℝ X'] {c' : H → X'} {p : X' → X} (hp : ContDiff ℝ ∞ p) (e : ∀ g, c g = p (c' g)) : CcSmooth c ≤ CcSmooth c'

theorem **exists_bump** [LocallyCompactSpace H] [T2Space H] [FiniteDimensional ℝ X] (hc : Topology.IsInducing c) {g : H} {U : Set H} (hU : U ∈ 𝒩 g) : ∃ φ : H → ℝ, (fun x => (φ x : ℂ)) ∈ CcSmooth c ∧ φ g = 1 ∧ (∀ x, φ x ∈ Icc 0 1) ∧ tsupport φ ⊆ U

theorem **isTempered_of_memLp_two** (hπ : IsUnitaryRep π) {v : V} (hv : v ≠ 0) (h : MemLp (fun g => ⟪v, π g v⟫_ℂ) 2 μ) : IsTempered μ π

theorem **isTempered_of_finite** [IsFiniteMeasure μ] [SecondCountableTopology H] [Nontrivial V] (hπ : IsUnitaryRep π) : IsTempered μ π

theorem **IsCharFn.unique** [LocallyCompactSpace H] [T2Space H] [μ.IsOpenPosMeasure] (hι : Topology.IsInducing (matCoord ι)) {θ θ' : H → ℂ} (h : IsCharFn ι μ π θ) (h' : IsCharFn ι μ π θ') {s : H} (hs : IsRegSS (ι s)) : θ s = θ' s

theorem **IsPseudoCoeff.orb_of_isCharFn** [LocallyCompactSpace H] [T2Space H] [μ.IsOpenPosMeasure] {f : H → ℂ} (h : IsPseudoCoeff ι μ π f) (hι : Topology.IsInducing (matCoord ι)) {θ : H → ℂ} (hθ : IsCharFn ι μ π θ) {s : H} (hs : IsRegSS (ι s)) (hT : IsCompact (cent s : Set H)) (dt : Measure (cent s)) [dt.IsHaarMeasure] [dt.IsMulRightInvariant] : HasOrbInt μ s dt f (conj (θ s) / dt.real univ)

theorem **isPseudoCoeff_char** [IsTopologicalGroup H] [CompactSpace H] [T2Space H] [SecondCountableTopology H] [μ.IsHaarMeasure] (hι : Continuous (matCoord ι)) {χ : H →* Circle} (hχ : (fun g => conj (χ g : ℂ)) ∈ CcMat ι) : IsPseudoCoeff ι μ (charRep χ) fun g => conj (χ g : ℂ) / μ.real univ

Development.AbelianVarietyTheory.Definitions.Foundations.CycMOfNot

theorem **cycM_spec** (M : ℕ) (s : Ω ≃ₐ[ℚ] Ω) (h : ∃ y : (ZMod M)ˣ, ∀ z : Ω, z ^ M = 1 → s z = z ^ (y : ZMod M).val) : ∀ z : Ω, z ^ M = 1 → s z = z ^ (cycM M s : ZMod M).val

theorem **repM_spec** (M : ℕ) (y : (ZMod M)ˣ) (h : ∃ s : Ω ≃ₐ[ℚ] Ω, ∀ z : Ω, z ^ M = 1 → s z = z ^ (y : ZMod M).val) : ∀ z : Ω, z ^ M = 1 → (repM M y : Ω ≃ₐ[ℚ] Ω) z = z ^ (y : ZMod M).val

Development.AbelianVarietyTheory.Definitions.Foundations.CyclotomicDvdExpandOfCoprime

theorem **hasSubst_om** (p : ℕ) (hp : 0 < p) : PowerSeries.HasSubst (((1 + PowerSeries.X : PowerSeries R) ^ p) - 1)

theorem **phi_X** (p : ℕ) (hp : 0 < p) : phi (R

theorem **phi_one_add_X** (p : ℕ) (hp : 0 < p) : phi (R

theorem **phi_eval₂** (p : ℕ) (hp : 0 < p) (P : R[X]) : phi (R

theorem **phi_cyc** (p m : ℕ) [hp : Fact p.Prime] (hm : 1 ≤ m) : phi (R

theorem **hasSubst_pow** (b : ℕ) : PowerSeries.HasSubst (((1 + PowerSeries.X : PowerSeries R) ^ b) - 1)

theorem **FrobLift.witt** (p : ℕ) [hp : Fact p.Prime] (k : Type) [Field k] [CharP k p] [PerfectRing k p] : FrobLift (WittVector p k) p (WittVector.frobeniusEquiv p k) where

theorem **padicHom_ext** (p : ℕ) [Fact p.Prime] [IsHausdorff (Ideal.span {(p : R)}) R] (f g : ℤ_[p] →+* R) : f = g

theorem **ev_cyc_self** (p m : ℕ) : ev R p m (cyc R p m) = 0

theorem **cyc_dvd_level** (p m : ℕ) : cyc R p m ∣ (1 + PowerSeries.X : PowerSeries R) ^ (p ^ m) - 1

theorem **coeffMap_cyc** {S : Type} [CommRing S] (τ : R →+* S) (p m : ℕ) : PowerSeries.map τ (cyc R p m) = cyc S p m

theorem **levMap_ev** {S : Type} [CommRing S] (τ : R →+* S) (p m : ℕ) (h : PowerSeries R) : levMap τ p m (ev R p m h) = ev S p m (PowerSeries.map τ h)

theorem **levMap_id** (p m : ℕ) (x : lev R p m) : levMap (RingHom.id R) p m x = x

theorem **levMap_comp** {S U : Type} [CommRing S] [CommRing U] (τ : R →+* S) (υ : S →+* U) (p m : ℕ) (x : lev R p m) : levMap (υ.comp τ) p m x = levMap υ p m (levMap τ p m x)

theorem **levF_ev** (σ : R ≃+* R) (p m : ℕ) (h : PowerSeries R) : levF σ p m (ev R p m h) = ev R p m (smap σ h)

theorem **sbp_apply** (b : ℤ_[p]) (h : PowerSeries R) : sbp p b h = act p R b h

theorem **sbp_one_add_X** (b : ℤ_[p]) : sbp p b (1 + PowerSeries.X : PowerSeries R) = PowerSeries.binomialSeries R b

theorem **sbp_nat** (b : ℕ) : sbp (R

theorem **sbp_one** (h : PowerSeries R) : sbp p 1 h = h

theorem **sbp_mul** (b c : ℤ_[p]) (h : PowerSeries R) : sbp p (b * c) h = sbp p b (sbp p c h)

theorem **coeffMap_sbp** {S : Type} [CommRing S] [Algebra ℤ_[p] S] (τ : R →+* S) (hτ : τ.comp (algebraMap ℤ_[p] R) = algebraMap ℤ_[p] S) (b : ℤ_[p]) (h : PowerSeries R) : PowerSeries.map τ (sbp p b h) = sbp p b (PowerSeries.map τ h)

theorem **smap_sbp** [IsHausdorff (Ideal.span {(p : R)}) R] (σ : R ≃+* R) (b : ℤ_[p]) (h : PowerSeries R) : smap σ (sbp p b h) = sbp p b (smap σ h)

theorem **ev_binomialSeries** (m : ℕ) (b : ℤ_[p]) (n : ℕ) (hn : b - n ∈ Ideal.span {(p : ℤ_[p]) ^ m}) : ev R p m (PowerSeries.binomialSeries R b) = levZeta R p m ^ n

theorem **cyclotomic_dvd_expand_of_coprime** (N : ℕ) : ∀ n : ℕ, n ≠ 0 → n.Coprime N → Polynomial.cyclotomic N R ∣ Polynomial.expand R n (Polynomial.cyclotomic N R)

theorem **ev_sbp_cyc** (m : ℕ) (b : ℤ_[p]ˣ) : ev R p m (sbp p (b : ℤ_[p]) (cyc R p m)) = 0

theorem **levAct_ev** (m : ℕ) (b : ℤ_[p]ˣ) (h : PowerSeries R) : levAct p m b (ev R p m h) = ev R p m (sbp p (b : ℤ_[p]) h)

theorem **levF_levAct** [IsHausdorff (Ideal.span {(p : R)}) R] (σ : R ≃+* R) (m : ℕ) (b : ℤ_[p]ˣ) (x : lev R p m) : levF σ p m (levAct p m b x) = levAct p m b (levF σ p m x)

Development.AbelianVarietyTheory.Definitions.Foundations.DiagUnitsEqLocalIdele

theorem **localIdele_snd_self** (v : HeightOneSpectrum (𝒪 L)) (x : (v.adicCompletion L)ˣ) : ((localIdele L v x : Ideles L) : AdeleRing (𝒪 L) L).2 v = x

theorem **localIdele_snd_ne** (v w : HeightOneSpectrum (𝒪 L)) (h : w ≠ v) (x : (v.adicCompletion L)ˣ) : ((localIdele L v x : Ideles L) : AdeleRing (𝒪 L) L).2 w = 1

theorem **localIdele_fst** (v : HeightOneSpectrum (𝒪 L)) (x : (v.adicCompletion L)ˣ) : ((localIdele L v x : Ideles L) : AdeleRing (𝒪 L) L).1 = 1

theorem **infIdele_fst_self** (w : InfinitePlace L) (x : (w.Completion)ˣ) : ((infIdele L w x : Ideles L) : AdeleRing (𝒪 L) L).1 w = x

theorem **infIdele_fst_ne** (w u : InfinitePlace L) (h : u ≠ w) (x : (w.Completion)ˣ) : ((infIdele L w x : Ideles L) : AdeleRing (𝒪 L) L).1 u = 1

theorem **infIdele_snd** (w : InfinitePlace L) (x : (w.Completion)ˣ) : ((infIdele L w x : Ideles L) : AdeleRing (𝒪 L) L).2 = 1

theorem **localIdele_injective** (v : HeightOneSpectrum (𝒪 L)) : Function.Injective (localIdele L v)

theorem **infIdele_injective** (w : InfinitePlace L) : Function.Injective (infIdele L w)

theorem **diagUnits_injective** : Function.Injective (diagUnits L)

theorem **diagUnits_snd** (v : HeightOneSpectrum (𝒪 L)) (a : Lˣ) : ((diagUnits L a : Ideles L) : AdeleRing (𝒪 L) L).2 v = algebraMap L (v.adicCompletion L) a

theorem **infIdele_continuous** (w : InfinitePlace L) : Continuous (infIdele L w)

theorem **diagUnits_fst** (w : InfinitePlace L) (a : Lˣ) : ((diagUnits L a : Ideles L) : AdeleRing (𝒪 L) L).1 w = algebraMap L w.Completion a

theorem **finSingle_continuous** (v : HeightOneSpectrum (𝒪 L)) : Continuous (finSingle L v)

theorem **localIdele_continuous** (v : HeightOneSpectrum (𝒪 L)) : Continuous (localIdele L v)

theorem **diagUnits_eq_localIdele** (v : HeightOneSpectrum (𝒪 L)) (a : Lˣ) (x : (v.adicCompletion L)ˣ) (h : diagUnits L a = localIdele L v x) : a = 1

theorem **localIdele_eq_infIdele** (v : HeightOneSpectrum (𝒪 L)) (w : InfinitePlace L) (x : (v.adicCompletion L)ˣ) (y : (w.Completion)ˣ) (h : localIdele L v x = infIdele L w y) : x = 1 ∧ y = 1

theorem **diagUnits_neg_one_ne** : diagUnits L (-1) ≠ 1

Development.AbelianVarietyTheory.Definitions.Foundations.EigenOfTwoStep

theorem **lift_congr** {B : Type} [Semiring B] [Algebra R B] {f g : Letter → B} (h : ∀ T ∈ A, f T = g T) {P : FreeAlgebra R Letter} (hP : P ∈ Algebra.adjoin R (FreeAlgebra.ι R '' A)) : FreeAlgebra.lift R f P = FreeAlgebra.lift R g P

theorem **_root_.GlobalTheory.KZ.SSAdm.lift_sub_smul_mem** {f : _root_.GlobalTheory.KZ.Letter → Module.End R M} {lam : _root_.GlobalTheory.KZ.Letter → R} {W : Submodule R M} (hW : ∀ T ∈ A, ∀ x ∈ W, f T x ∈ W) {x : M} (hx : ∀ T ∈ A, f T x - lam T • x ∈ W) {P : FreeAlgebra R _root_.GlobalTheory.KZ.Letter} (hP : P ∈ Algebra.adjoin R (FreeAlgebra.ι R '' A)) : FreeAlgebra.lift R f P x - FreeAlgebra.lift R lam P • x ∈ W

theorem **eigen_of_two_step** {f : Letter → Module.End R M} {lam : Letter → R} {Z : Submodule R M} (hZ : ∀ T ∈ A, ∀ z ∈ Z, f T z ∈ Z) {P : FreeAlgebra R Letter} (hP : P ∈ Algebra.adjoin R (FreeAlgebra.ι R '' A)) (h1 : FreeAlgebra.lift R lam P = 1) {x : M} (hPx : ∀ T ∈ A, (f T - lam T • (1 : Module.End R M)) (FreeAlgebra.lift R f P x) ∈ Z) (hx : ∀ S ∈ A, ∀ T ∈ A, (f S - lam S • (1 : Module.End R M)) ((f T - lam T • (1 : Module.End R M)) x) ∈ Z) : ∀ T ∈ A, (f T - lam T • (1 : Module.End R M)) x ∈ Z

theorem **SSPotA_congr** {N l w : ℕ} {lam lam' : Letter → ℂ} (h : ∀ T, admitted N l T → lam T = lam' T) : SSPotA N l w lam ↔ SSPotA N l w lam'

theorem **Tower.SSBA.eigen_of_two_step** {X : Tower Ω ε} {N l w : ℕ} [Fact l.Prime] {E : Type} [Field E] [Algebra ℤ_[l] E] {lam : Letter → E} (h : X.SSBA N l w E lam) {x : E ⊗[ℤ_[l]] X.Fn l} (hx : x ∈ X.VB N l w E) (h2 : ∀ S, admitted N l S → ∀ T, admitted N l T → (X.opV N l E S - lam S • (1 : Module.End E (E ⊗[ℤ_[l]] X.Fn l))) ((X.opV N l E T - lam T • (1 : Module.End E (E ⊗[ℤ_[l]] X.Fn l))) x) = 0) : ∀ T, admitted N l T → X.opV N l E T x = lam T • x

Development.AbelianVarietyTheory.Definitions.Foundations.EigenfunctionalZeroProjection

theorem **IsEigenfunctionalOfL2.smul** (h : IsEigenfunctionalOfL2 X A N w j pr lam) (a : Qbar l) : IsEigenfunctionalOfL2 X A N w j (a • pr) (a • lam)

Development.AbelianVarietyTheory.Definitions.Foundations.EisValueAtIffEisPoleLE

theorem **leftInv_of_mem** (hμ : IsHeckeChar D μ) (hΦ : Φ ∈ degPS V a μ s) : LeftInv (siegelQ V) Φ

theorem **EisConv.hasSum** (h : EisConv V Φ g) : HasSum (fun q : SiegelCosets V => Φ ((q.out : GQ (dblSp V)) ∗ g)) (eis V Φ g)

theorem **leftInv_eis** (h : LeftInv (siegelQ V) Φ) : LeftInv (GQ (dblSp V)) (eis V Φ)

theorem **eis_rTr** (k : GA D (dblSp V)) (Φ : GA D (dblSp V) → ℂ) : eis V (rTr _ k Φ) = rTr _ k (eis V Φ)

theorem **eis_add** (h : EisConv V Φ g) (h' : EisConv V Ψ g) : eis V (Φ + Ψ) g = eis V Φ g + eis V Ψ g

theorem **eis_smul** (c : ℂ) (Φ : GA D (dblSp V) → ℂ) : eis V (c • Φ) = c • eis V Φ

theorem **injective_mk_io** : Function.Injective fun γ : GQ V => (mk _ (io V (γ, 1), io_mem_GQ V γ.2 (one_mem _)) : SiegelCosets V)

theorem **IsEisFamily.eqRight** (h : IsEisFamily V Φ) {γ : GA D (dblSp V)} (hγ : γ ∈ GQ (dblSp V)) (g : GA D (dblSp V)) : EqRight (eisSer V Φ (γ ∗ g)) (eisSer V Φ g)

theorem **IsEisFamily.add** (h : IsEisFamily V Φ) (h' : IsEisFamily V Ψ) : IsEisFamily V (Φ + Ψ)

theorem **IsEisFamily.smul** (h : IsEisFamily V Φ) (c : ℂ → ℂ) : IsEisFamily V (c • Φ)

theorem **IsEisFamily.rTr** (h : IsEisFamily V Φ) (x : GA D (dblSp V)) : IsEisFamily V fun s => rTr _ x (Φ s)

theorem **IsEisHolExt.isEisExt** (h : IsEisHolExt V Φ U E) : IsEisExt V Φ U E

theorem **IsEisExt.isMerExt** (h : IsEisExt V Φ U E) (g : GA D (dblSp V)) : IsMerExt (eisSer V Φ g) U (E · g)

theorem **IsEisExt.eisValueAt** (h : IsEisExt V Φ U E) (hs : s_0 ∈ U) (ha : ∀ g, AnalyticAt ℂ (E · g) s_0) : EisValueAt V Φ s_0 (E s_0)

theorem **IsEisHolExt.eisValueAt** (h : IsEisHolExt V Φ U E) (hs : s_0 ∈ U) : EisValueAt V Φ s_0 (E s_0)

theorem **EisValueAt.eisCoeffAt** (h : EisValueAt V Φ s_0 φ) : EisCoeffAt V Φ s_0 0 φ

theorem **eisValueAt_iff_eisPoleLE** : EisValueAt V Φ s_0 φ ↔ EisPoleLE V Φ s_0 0 ∧ EisCoeffAt V Φ s_0 0 φ

theorem **EisCoeffAt.eisLaurent_eq** (h : EisCoeffAt V Φ s_0 n φ) : eisLaurent V Φ s_0 n = φ

theorem **EisValueAt.eisCont_eq** (h : EisValueAt V Φ s_0 φ) : eisCont V Φ s_0 = φ

theorem **EisCoeffAt.unique** (h : EisCoeffAt V Φ s_0 n φ) (h' : EisCoeffAt V Φ s_0 n ψ) : φ = ψ

theorem **EisValueAt.unique** (h : EisValueAt V Φ s_0 φ) (h' : EisValueAt V Φ s_0 ψ) : φ = ψ

theorem **EisCoeffAt.leftInv** (h : EisCoeffAt V Φ s_0 n φ) : LeftInv (GQ (dblSp V)) φ

theorem **EisValueAt.leftInv** (h : EisValueAt V Φ s_0 φ) : LeftInv (GQ (dblSp V)) φ

theorem **IsEisFamily.leftInv_eisLaurent** (h : IsEisFamily V Φ) (s_0 : ℂ) (n : ℤ) : LeftInv (GQ (dblSp V)) (eisLaurent V Φ s_0 n)

theorem **EisContinues.eisCoeffAt** (h : EisContinues V Φ) (s_0 : ℂ) (n : ℤ) : EisCoeffAt V Φ s_0 n (eisLaurent V Φ s_0 n)

theorem **EisCoeffAt.add** (h : EisCoeffAt V Φ s_0 n φ) (h' : EisCoeffAt V Ψ s_0 n ψ) : EisCoeffAt V (Φ + Ψ) s_0 n (φ + ψ)

theorem **EisValueAt.add** (h : EisValueAt V Φ s_0 φ) (h' : EisValueAt V Ψ s_0 ψ) : EisValueAt V (Φ + Ψ) s_0 (φ + ψ)

theorem **EisCoeffAt.smul** (h : EisCoeffAt V Φ s_0 n φ) (c : ℂ) : EisCoeffAt V (c • Φ) s_0 n (c • φ)

theorem **EisValueAt.smul** (h : EisValueAt V Φ s_0 φ) (c : ℂ) : EisValueAt V (c • Φ) s_0 (c • φ)

theorem **EisCoeffAt.rTr** (h : EisCoeffAt V Φ s_0 n φ) (x : GA D (dblSp V)) : EisCoeffAt V (fun s => rTr _ x (Φ s)) s_0 n (rTr _ x φ)

theorem **EisValueAt.rTr** (h : EisValueAt V Φ s_0 φ) (x : GA D (dblSp V)) : EisValueAt V (fun s => rTr _ x (Φ s)) s_0 (rTr _ x φ)

theorem **IsEisFamily.isEisHolExt** : IsEisHolExt V Φ (rightOf (V.rank / 2)) fun s => eis V (Φ s)

theorem **IsEisFamily.eisValueAt** (hs : (V.rank : ℝ) / 2 < s_0.re) : EisValueAt V Φ s_0 (eis V (Φ s_0))

theorem **IsEisFamily.analyticOnNhd_flat** {s_1 : ℂ} (h : IsEisFamily V (flat V a φ s_1)) (g : GA D (dblSp V)) : AnalyticOnNhd ℂ (eisSer V (flat V a φ s_1) g) (rightOf (V.rank / 2))

theorem **isIwasawa_Khat** : IsIwasawa ((absA D).comp (aP V)) (Khat V a)

theorem **indLaw_of_mem** : ∀ s ∈ rightOf (V.rank / 2), IndLaw ((absA D).comp (aP V)) (μ.comp (aP V)) (s + ((V.rank : ℝ) / 2 : ℝ)) (Φ s)

theorem **SphConv.summable** : ∀ τ, (V.rank : ℝ) / 2 < τ → Summable fun q => ((term (siegel V) (GQ (dblSp V)) (iwAbs ((absA D).comp (aP V)) (Khat V a)) 1 q : ℝ≥0) : ℝ) ^ (τ + (V.rank : ℝ) / 2)

theorem **SphConv.isEisFamily** : IsEisFamily V Φ

theorem **SphConv.continuousOn_eis** : ContinuousOn (fun x : ℂ × GA D (dblSp V) => eis V (Φ x.1) x.2) (rightOf (V.rank / 2) ×ˢ univ)

theorem **flat_mem** (s : ℂ) : flat V a φ s_1 s ∈ degPS V a μ s

theorem **continuous_flat** : Continuous fun x : ℂ × GA D (dblSp V) => flat V a φ s_1 x.1 x.2

theorem **SphConv.isEisFamily_flat** (hs : SphConv V a) (hμ : IsHeckeChar D μ) : IsEisFamily V (flat V a φ s_1)

theorem **SphConv.continuous_eis** (hs : SphConv V a) (h : (V.rank : ℝ) / 2 < s_1.re) : Continuous (eis V φ)

theorem **SphConv.eisValueAt_flat** (hs : SphConv V a) (hμ : IsHeckeChar D μ) (h : (V.rank : ℝ) / 2 < s_1.re) : EisValueAt V (flat V a φ s_1) s_1 (eis V φ)

theorem **sph_mem** (s : ℂ) : sph V a s ∈ degPS V a 1 s

theorem **sphConv_iff** : SphConv V a ↔ IsEisFamily V (sph V a) ∧ ∀ σ : ℝ, (V.rank : ℝ) / 2 < σ → 1 ≤ (eis V (sph V a σ) 1).re

theorem **apply_detA** (hμ : IsHeckeChar D μ) {γ : GA D V} (hγ : γ ∈ GQ V) : μ (detA V γ) = 1

theorem **leftInv_dblIntegrand** (hμ : IsHeckeChar D μ) (hE : LeftInv (GQ (dblSp V)) E) (h_1 : LeftInv (GQ V) f_1) (h_2 : LeftInv (GQ V) f_2) : LeftInv ((GQ V).prod (GQ V)) (dblIntegrand V μ E f_1 f_2)

theorem **continuous_detA** : Continuous (detA V)

theorem **continuous_dblIntegrand** (hμ : IsHeckeChar D μ) (hE : Continuous E) (h_1 : Continuous f_1) (h_2 : Continuous f_2) : Continuous (dblIntegrand V μ E f_1 f_2)

theorem **dblIntegrandQ_mk** (h : LeftInv ((GQ V).prod (GQ V)) (dblIntegrand V μ E f_1 f_2)) (g_1 g_2 : GA D V) : dblIntegrandQ V μ E f_1 f_2 (mk (GQ V) g_1, mk (GQ V) g_2) = dblIntegrand V μ E f_1 f_2 (g_1, g_2)

theorem **dblIntegrand_mul** {s : ℂ} (hE : E ∈ degPS V a μ s) (g x : GA D V) : dblIntegrand V μ E f₁ f₂ (x * g, x) = E (io V (g, 1)) * conj (f₁ (x * g)) * f₂ x

theorem **dblInt_smul** (c c₁ c₂ : ℂ) : dblInt V μ m (c • E) (c₁ • f₁) (c₂ • f₂) = c * conj c₁ * c₂ * dblInt V μ m E f₁ f₂

theorem **dblInt_add** (h : DblConv V μ m E f₁ f₂) (h' : DblConv V μ m E' f₁ f₂) : dblInt V μ m (E + E') f₁ f₂ = dblInt V μ m E f₁ f₂ + dblInt V μ m E' f₁ f₂

theorem **dblInt_smul_measure** (c : ℝ≥0) : dblInt V μ (c • m) E f₁ f₂ = (c : ℂ) ^ 2 * dblInt V μ m E f₁ f₂

theorem **dblInt_of_factor** {A B : GA D V → ℂ} (h : ∀ g₁ g₂, E (io V (g₁, g₂)) = A g₁ * conj (B g₂) * μ (detA V g₂)) : dblInt V μ m E f₁ f₂ = Cusp.petersson V m A f₁ * Cusp.petersson V m f₂ B

theorem **DblConv.dblInt_eq** (hc : DblConv V μ m E f₁ f₂) (h : IsFundamentalDomain (GQ V) 𝓕 m) : IntegrableOn (dblIntegrand V μ E f₁ f₂) (𝓕 ×ˢ 𝓕) (m.prod m) ∧ dblInt V μ m E f₁ f₂ = ∫ g in 𝓕 ×ˢ 𝓕, dblIntegrand V μ E f₁ f₂ g ∂m.prod m

theorem **dblConv_iff** (hi : LeftInv ((GQ V).prod (GQ V)) (dblIntegrand V μ E f₁ f₂)) (hm : Measurable (dblIntegrand V μ E f₁ f₂)) (h : IsFundamentalDomain (GQ V) 𝓕 m) : DblConv V μ m E f₁ f₂ ↔ IntegrableOn (dblIntegrand V μ E f₁ f₂) (𝓕 ×ˢ 𝓕) (m.prod m)

theorem **ZetaValueAt.unique** (h : ZetaValueAt V μ m Φ f₁ f₂ s₀ c) (h' : ZetaValueAt V μ m Φ f₁ f₂ s₀ c') : c = c'

theorem **ZetaCoeffAt.unique** (h : ZetaCoeffAt V μ m Φ f₁ f₂ s₀ n c) (h' : ZetaCoeffAt V μ m Φ f₁ f₂ s₀ n c') : c = c'

theorem **eis_smul_indicator** (V : HSp D) (c : ℂ → ℂ) : IsEisFamily V (fun s => c s • indicator (siegelQ V : Set (GA D (dblSp V))) 1) ∧ ∀ s, eis V (c s • indicator (siegelQ V : Set (GA D (dblSp V))) 1) = c s • indicator (GQ (dblSp V) : Set _) 1

Development.AbelianVarietyTheory.Definitions.Foundations.EqPlaceIdemOfIsPrimeIdem

theorem **tens_map_map** {R R' R'' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] [CommRing R''] [Algebra F R''] (φ : R →ₐ[F] R') (ψ : R' →ₐ[F] R'') (y : R ⊗[F] E) : Algebra.TensorProduct.map ψ (AlgHom.id F E) (Algebra.TensorProduct.map φ (AlgHom.id F E) y) = Algebra.TensorProduct.map (ψ.comp φ) (AlgHom.id F E) y

theorem **adTens_eq_one'** {y : AF F ⊗[F] E} (h1 : ∀ w, compInf F E w y = 1) (h2 : ∀ v, compFin F E v y = 1) : y = 1

theorem **mul_idem_iff_mem_range_emb** {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (v : HeightOneSpectrum (𝒪 F)) (ε : AF F) (e0 : ∀ w, evInf F w ε = 0) (e1 : evFin F v ε = 1) (e2 : ∀ u, u ≠ v → evFin F u ε = 0) (g : GAd σ J) : (∀ i j, algebraMap (AF F) (AF F ⊗[F] E) ε * ((matHom _ σ J g).1 - 1) i j = ((matHom _ σ J g).1 - 1) i j) ↔ g ∈ (emb σ J v).range

theorem **adTens_ext** {y z : AF F ⊗[F] E} (h1 : ∀ w, compInf F E w y = compInf F E w z) (h2 : ∀ v, compFin F E v y = compFin F E v z) : y = z

@[simp] theorem **compFin_singleTens_self** (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v ⊗[F] E) : compFin F E v (singleTens E v y) = y

theorem **compFin_singleTens_ne** {v u : HeightOneSpectrum (𝒪 F)} (h : u ≠ v) (y : Fv F v ⊗[F] E) : compFin F E u (singleTens E v y) = 0

@[simp] theorem **compInf_singleTens** (w : InfinitePlace F) (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v ⊗[F] E) : compInf F E w (singleTens E v y) = 0

theorem **compFin_one_tmul** (v : HeightOneSpectrum (𝒪 F)) (x : E) : compFin F E v ((1 : AF F) ⊗ₜ[F] x) = (1 : Fv F v) ⊗ₜ[F] x

theorem **singleTens_one** (v : HeightOneSpectrum (𝒪 F)) : singleTens E v 1 = algebraMap (AF F) (AF F ⊗[F] E) (placeIdem v)

theorem **singleTens_mul** (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v ⊗[F] E) (z : AF F ⊗[F] E) : singleTens E v y * z = singleTens E v (y * compFin F E v z)

theorem **sigR_singleTens** (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v ⊗[F] E) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ (singleTens E v y) = singleTens E v (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ y)

theorem **splitIdem_mul_eq** (h : IsCMExt F E σ) {v : HeightOneSpectrum (𝒪 F)} (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : splitIdem h τ * y = (splitProj τ y ⊗ₜ[F] (1 : E)) * splitIdem h τ

theorem **splitIdemAd_mul** (h : IsCMExt F E σ) {v : HeightOneSpectrum (𝒪 F)} (τ : E →ₐ[F] Fv F v) (z : AF F ⊗[F] E) : splitIdemAd h τ * z = singleTens E v (splitProj τ (compFin F E v z) ⊗ₜ[F] (1 : E)) * splitIdemAd h τ

theorem **isPlaceIdem_placeIdem** (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) : IsPlaceIdem D v (placeIdem v)

theorem **isPlaceIdem_iff** {ε : AF D.host.F} : IsPlaceIdem D v ε ↔ ε = placeIdem v

theorem **suppF_iff_mem_range** {ε : AF D.host.F} (h : IsPlaceIdem D v ε) (g : GA D V) : SuppF V (algebraMap (AF D.host.F) (AE D) ε) g ↔ g ∈ (emb D.sp.cm.sig V.gram v).range

theorem **suppAt_iff_mem_range** {ε : AF D.host.F} (h : IsPlaceIdem D v ε) {e : AE D} (he : e + cA D e = algebraMap (AF D.host.F) (AE D) ε) (g : GA D V) : SuppAt e V g ↔ g ∈ (emb D.sp.cm.sig V.gram v).range

theorem **emb_mem_KsphC** (x : GLoc D.sp.cm.sig V.gram v) : emb D.sp.cm.sig V.gram v x ∈ KsphC V ∅ ↔ x ∈ stabLoc D.sp.cm.sig V.gram (frameLat V) v

theorem **suppF_and_mem_KsphC** {ε : AF D.host.F} (h : IsPlaceIdem D v ε) (g : GA D V) : (SuppF V (algebraMap (AF D.host.F) (AE D) ε) g ∧ g ∈ KsphC V ∅) ↔ g ∈ (stabLoc D.sp.cm.sig V.gram (frameLat V) v).map (emb D.sp.cm.sig V.gram v)

theorem **fj_singleV** (y : Fv D.host.F v) : fj D (singleV D.host.F v y) = singleTens D.host.E v (y ⊗ₜ[D.host.F] (1 : D.host.E))

theorem **pl_eq** (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) : pl D v = algebraMap (AF D.host.F) (AE D) (placeIdem v)

theorem **splitAt_splitIdemAd** (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : SplitAt D v (splitIdemAd (isCMExt_host D) τ)

theorem **isIdemAt_splitIdemAd** (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : IsIdemAt (D

theorem **exists_eq_splitIdemAd** {e : AE D} (he : SplitAt D v e) : ∃ τ : D.host.E →ₐ[D.host.F] Fv D.host.F v, e = splitIdemAd (isCMExt_host D) τ

theorem **isSplitIdem_iff_exists** {e : AE D} : IsSplitIdem D e ↔ ∃ (v : HeightOneSpectrum (𝒪 D.host.F)) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v), e = splitIdemAd (isCMExt_host D) τ

theorem **isIdemAt_iff_exists** {e : AE D} {P : HeightOneSpectrum (𝒪 D.host.E)} : IsIdemAt e P ↔ ∃ (v : HeightOneSpectrum (𝒪 D.host.F)) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v), e = splitIdemAd (isCMExt_host D) τ ∧ P = splitPrime τ

theorem **suppAt_splitIdemAd** (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (g : GA D V) : SuppAt (D

theorem **finPart_fj_mem_iff** (y : Fv D.host.F v) : finPart D (fj D (singleV D.host.F v y)) ∈ intAE D ↔ Valued.v y ≤ 1

theorem **eq_placeIdem_of_isPrimeIdem** {P : HeightOneSpectrum (𝒪 D.host.E)} {e f : AE D} (h : IsPrimeIdem P e f) (hv : P.under (𝒪 D.host.F) = v) : f = algebraMap (AF D.host.F) (AE D) (placeIdem v)

theorem **suppF_iff_of_isPrimeIdem** {P : HeightOneSpectrum (𝒪 D.host.E)} {e f : AE D} (h : IsPrimeIdem P e f) (hv : P.under (𝒪 D.host.F) = v) (g : GA D V) : SuppF V f g ↔ g ∈ (emb D.sp.cm.sig V.gram v).range

Development.AbelianVarietyTheory.Definitions.Foundations.EtaSixProductDifferentiable

lemma **eta_six_ne_zero** (z : ℍ) : eta6 z ≠ 0

lemma **eta_six_product_formula** (z : ℍ) : eta6 z = Periodic.qParam 4 z * (∏' n, (1 - eta_q n z)) ^ 6

lemma **eta6_slash_three_T_eq_I_smul** : eta6 ∣[(3 : ℤ)] ModularGroup.T = (Complex.I : ℂ) • eta6

lemma **sqrt_pow_six** (z : ℂ) : Complex.sqrt z ^ 6 = z ^ 3

lemma **eta_six_slash_S** : eta6 ∣[(3 : ℤ)] ModularGroup.S = (Complex.I : ℂ) • eta6

lemma **eta_six_slash_root_of_unity** (γ : SL(2, ℤ)) : ∃ c : ℂ, c ^ 4 = 1 ∧ eta6 ∣[(3 : ℤ)] γ = c • eta6

lemma **eta6Mult0_eq** {γ : SL(2, ℤ)} {c : ℂ} (h : eta6 ∣[(3 : ℤ)] γ = c • eta6) : eta6MultiplierFunction γ = c

lemma **eta_six_slash_multiplier** (γ : SL(2, ℤ)) : eta6 ∣[(3 : ℤ)] γ = eta6MultiplierFunction γ • eta6

lemma **eta6Mult_T_eq_I** : eta6Mult ModularGroup.T = Complex.I

lemma **eta6Mult_S** : eta6Mult ModularGroup.S = Complex.I

lemma **eta6Mult_neg_one** : eta6Mult (-1) = -1

lemma **mem_eta6Group** (γ : SL(2, ℤ)) : γ ∈ eta6Group ↔ eta6Mult γ = 1

lemma **T_pow_four_mem_eta6Group** : ModularGroup.T ^ 4 ∈ eta6Group

lemma **neg_one_not_mem_eta6Group** : (-1 : SL(2, ℤ)) ∉ eta6Group

lemma **eta_six_holomorphic** : MDiff eta6

lemma **eta_six_zero_at_infinity** : IsZeroAtImInfty eta6

lemma **coe_eta6Form** : (eta6Form : ℍ → ℂ) = eta6

lemma **eta_six_cusp_function** : Set.EqOn (cuspFunction 4 eta6) (fun q ↦ q * (∏' i, (1 - (q ^ 4) ^ (i + 1))) ^ 6) (Metric.ball 0 1)

lemma **eta_six_product_differentiable** : DifferentiableOn ℂ (fun q : ℂ ↦ (∏' i, (1 - (q ^ 4) ^ (i + 1))) ^ 6) (Metric.ball 0 1)

lemma **eta_six_first_coefficient** : (qExpansion 4 eta6).coeff 1 = 1

Development.AbelianVarietyTheory.Definitions.Foundations.EventuallyEqOfIsPreconnected

@[simp] theorem **mem_rightOf** : s ∈ rightOf σ ↔ σ < s.re

theorem **isOpen_rightOf** (σ : ℝ) : IsOpen (rightOf σ)

theorem **rightOf_subset** (e : σ ≤ τ) : rightOf τ ⊆ rightOf σ

theorem **isRightDomain_univ** : IsRightDomain univ

theorem **isRightDomain_rightOf** (σ : ℝ) : IsRightDomain (rightOf σ)

theorem **isRightDomain_re_ge** (σ : ℝ) : IsRightDomain {s | σ ≤ s.re}

theorem **IsRightDomain.inter** (a : IsRightDomain U) (b : IsRightDomain U') : IsRightDomain (U ∩ U')

theorem **IsRightDomain.preimage_add** (a : IsRightDomain U) (z : ℂ) : IsRightDomain ((· + z) ⁻¹' U)

theorem **IsRightDomain.isPreconnected** (a : IsRightDomain U) : IsPreconnected U

@[**refl**] theorem refl (h : ℂ → ℂ) : EqRight h h

theorem **symm** (a : EqRight H h) : EqRight h H

theorem **map₂** (f : ℂ → ℂ → ℂ) (a : EqRight H h) (b : EqRight H' h') : EqRight (fun s => f (H s) (H' s)) fun s => f (h s) (h' s)

theorem **trans** (a : EqRight H h) (b : EqRight h h') : EqRight H h'

theorem **eventuallyEq_of_isPreconnected** (hU : IsPreconnected U) (hσ : rightOf σ ⊆ U) (a : MeromorphicOn H U) (b : MeromorphicOn H' U) (e : EqRight H H') (hs : s ∈ U) : H =ᶠ[𝓝[≠] s] H'

theorem **laurentConst_congr** (a : LaurentConst H s₀ c) (e : H =ᶠ[𝓝[≠] s₀] H') : LaurentConst H' s₀ c

theorem **laurentConst_of_tendsto** (a : Tendsto H (𝓝[≠] s₀) (𝓝 c)) : LaurentConst H s₀ c

theorem **laurentConst_add** (a : LaurentConst H s₀ c) (b : LaurentConst H' s₀ c') : LaurentConst (fun s => H s + H' s) s₀ (c + c')

theorem **laurentConst_const_mul** (z : ℂ) (a : LaurentConst H s₀ c) : LaurentConst (fun s => z * H s) s₀ (z * c)

theorem **laurentConst_unique** (a : LaurentConst H s₀ c) (b : LaurentConst H s₀ c') : c = c'

private theorem **laurentConst_add_poly_eval_inv** {F : ℂ → ℂ} {s₀ c : ℂ} (a : LaurentConst F s₀ c) (q : Polynomial ℂ) (hq : q.coeff 0 = 0) : LaurentConst (fun s => F s + q.eval (s - s₀)⁻¹) s₀ c

theorem **laurentConst_div_pow** (k : ℕ) : ∀ {g : ℂ → ℂ} {p : FormalMultilinearSeries ℂ ℂ ℂ}, HasFPowerSeriesAt g p s₀ → LaurentConst (fun s => g s / (s - s₀) ^ k) s₀ (p.coeff k)

theorem **laurentCoeff_div_pow** {g : ℂ → ℂ} {p : FormalMultilinearSeries ℂ ℂ ℂ} (hg : HasFPowerSeriesAt g p s₀) (i k : ℕ) : LaurentCoeff (fun s => g s / (s - s₀) ^ k) s₀ (i - k) (p.coeff i)

theorem **exists_laurentConst** (a : MeromorphicAt H s₀) : ∃ c, LaurentConst H s₀ c

theorem **exists_laurentCoeff** (a : MeromorphicAt H s₀) (n : ℤ) : ∃ c, LaurentCoeff H s₀ n c

@[simp] theorem **laurentCoeff_zero** : LaurentCoeff H s₀ 0 c ↔ LaurentConst H s₀ c

theorem **LaurentCoeff.congr** (a : LaurentCoeff H s₀ n c) (e : H =ᶠ[𝓝[≠] s₀] H') : LaurentCoeff H' s₀ n c

theorem **IsHolExt.isMerExt** (a : IsHolExt h U H) : IsMerExt h U H

theorem **IsMerExt.isExtAt** (a : IsMerExt h U H) (hs : s₀ ∈ U) : IsExtAt h s₀ H

theorem **IsMerExt.mono** (a : IsMerExt h U H) (b : IsRightDomain U') (e : U' ⊆ U) : IsMerExt h U' H

theorem **isMerExt_univ** : IsMerExt h univ H ↔ Meromorphic H ∧ EqRight H h

theorem **IsMerExt.eventuallyEq_of_isPreconnected** (a : IsMerExt h U H) (e : U' ⊆ U) (hU : IsPreconnected U') (hσ : rightOf σ ⊆ U') (b : MeromorphicOn H' U') (e' : EqRight H' h) (hs : s ∈ U') : H =ᶠ[𝓝[≠] s] H'

theorem **meromorphicAt** (a : IsExtAt h s₀ H) : MeromorphicAt H s₀

theorem **eqRight** (a : IsExtAt h s_0 H) : EqRight H h

theorem **eventuallyEq** (a : IsExtAt h s_0 H) (b : IsExtAt h s_0 H') : H =ᶠ[$\mathcal{N}$[≠] s_0] H'

theorem **congr** (a : IsExtAt h s_0 H) (e : EqRight h h') : IsExtAt h' s_0 H

theorem **map₂** (f : ℂ → ℂ → ℂ) (hf : ∀ s, MeromorphicAt H s → MeromorphicAt H' s → MeromorphicAt (fun s => f (H s) (H' s)) s) (a : IsExtAt h s_0 H) (b : IsExtAt h' s_0 H') : IsExtAt (fun s => f (h s) (h' s)) s_0 fun s => f (H s) (H' s)

theorem **add** (a : IsExtAt h s_0 H) (b : IsExtAt h' s_0 H') : IsExtAt (fun s => h s + h' s) s_0 fun s => H s + H' s

theorem **mul** (a : IsExtAt h s_0 H) (b : IsExtAt h' s_0 H') : IsExtAt (fun s => h s * h' s) s_0 fun s => H s * H' s

theorem **div** (a : IsExtAt h s_0 H) (b : IsExtAt h' s_0 H') : IsExtAt (fun s => h s / h' s) s_0 fun s => H s / H' s

theorem **comp_add** {z : ℂ} (a : IsExtAt h (s_0 + z) H) : IsExtAt (fun s => h (s + z)) s_0 fun s => H (s + z)

theorem **update** (a : IsExtAt h s_0 H) (c : ℂ) : IsExtAt h s_0 (Function.update H s_0 c)

private theorem **isOpen_setOf_meromorphicAt** (H : ℂ → ℂ) : IsOpen {x | MeromorphicAt H x}

private theorem **isOpen_setOf_add_nonneg_real_mem** {W : Set ℂ} {σ : ℝ} (hW : IsOpen W) (hσ : rightOf σ ⊆ W) : IsOpen {x : ℂ | ∀ t : ℝ, 0 ≤ t → x + t ∈ W}

private theorem **exists_isOpen_isMerExt_of_isExtAt** (a : IsExtAt h s_0 H) : ∃ U, IsOpen U ∧ s_0 ∈ U ∧ IsMerExt h U H

theorem **eventually** (a : IsExtAt h s_0 H) : ∀ᶠ s in $\mathcal{N}$ s_0, IsExtAt h s H

theorem **exists_isOpen** (a : IsExtAt h s_0 H) : ∃ U, IsOpen U ∧ s_0 ∈ U ∧ IsMerExt h U H

theorem **isMerExt_self** (a : IsRightDomain U) (b : MeromorphicOn h U) : IsMerExt h U h

theorem **isExtAt_const** (z s_0 : ℂ) : IsExtAt (fun _ => z) s_0 fun _ => z

theorem **IsExtAt.coeffAt_iff** (a : IsExtAt h s_0 H) : CoeffAt h s_0 n c ↔ LaurentCoeff H s_0 n c

theorem **IsExtAt.valueAt_iff** (a : IsExtAt h s_0 H) : ValueAt h s_0 c ↔ LaurentConst H s_0 c

theorem **IsExtAt.holValueAt_iff** (a : IsExtAt h s_0 H) : HolValueAt h s_0 c ↔ Tendsto H ($\mathcal{N}$[≠] s_0) ($\mathcal{N}$ c)

theorem **IsExtAt.poleLE_iff** (a : IsExtAt h s_0 H) : PoleLE h s_0 k ↔ ∃ c, Tendsto (fun s => (s − s_0) ^ k * H s) ($\mathcal{N}$[≠] s_0) ($\mathcal{N}$ c)

theorem **IsExtAt.coeffAt_of_tendsto** (a : IsExtAt h s_0 H) (e : Tendsto (fun s => (s − s_0) ^ k * H s) ($\mathcal{N}$[≠] s_0) ($\mathcal{N}$ c)) : CoeffAt h s_0 (−k) c

theorem **IsExtAt.tendsto_of_poleLE** (a : IsExtAt h s_0 H) (b : PoleLE h s_0 k) (e : CoeffAt h s_0 (−k) c) : Tendsto (fun s => (s − s_0) ^ k * H s) ($\mathcal{N}$[≠] s_0) ($\mathcal{N}$ c)

@[simp] theorem **coeffAt_zero** : CoeffAt h s_0 0 c ↔ ValueAt h s_0 c

theorem **CoeffAt.unique** (a : CoeffAt h s_0 n c) (b : CoeffAt h s_0 n c') : c = c'

theorem **ValueAt.unique** (a : ValueAt h s_0 c) (b : ValueAt h s_0 c') : c = c'

theorem **HolValueAt.valueAt** (a : HolValueAt h s_0 c) : ValueAt h s_0 c

theorem **HolValueAt.unique** (a : HolValueAt h s_0 c) (b : HolValueAt h s_0 c') : c = c'

theorem **holValueAt_iff_poleLE** : HolValueAt h s_0 c ↔ PoleLE h s_0 0 ∧ ValueAt h s_0 c

theorem **IsHolExt.holValueAt** (a : IsHolExt h U H) (hs : s_0 ∈ U) : HolValueAt h s_0 (H s_0)

theorem **holValueAt_self** (a : AnalyticOnNhd ℂ h (rightOf σ)) (hs : σ < s_0.re) : HolValueAt h s_0 (h s_0)

theorem **Continues.holValueAt** (a : Continues h) (hU : IsPreconnected U) (hσ : rightOf σ ⊆ U) (b : AnalyticOnNhd ℂ H U) (e : EqRight H h) (hs : s_0 ∈ U) : HolValueAt h s_0 (H s_0)

theorem **CoeffAt.congr** (a : CoeffAt h s_0 n c) (e : EqRight h h') : CoeffAt h' s_0 n c

theorem **ValueAt.congr** (a : ValueAt h s_0 c) (e : EqRight h h') : ValueAt h' s_0 c

theorem **HolValueAt.congr** (a : HolValueAt h s_0 c) (e : EqRight h h') : HolValueAt h' s_0 c

theorem **PoleLE.congr** (a : PoleLE h s_0 k) (e : EqRight h h') : PoleLE h' s_0 k

theorem **Continues.congr** (a : Continues h) (e : EqRight h h') : Continues h'

theorem **CoeffAt.add** (a : CoeffAt h s_0 n c) (b : CoeffAt h' s_0 n c') : CoeffAt (fun s => h s + h' s) s_0 n (c + c')

theorem **CoeffAt.const_mul** (z : ℂ) (a : CoeffAt h s_0 n c) : CoeffAt (fun s => z * h s) s_0 n (z * c)

theorem **ValueAt.add** (a : ValueAt h s_0 c) (b : ValueAt h' s_0 c') : ValueAt (fun s => h s + h' s) s_0 (c + c')

theorem **ValueAt.const_mul** (z : ℂ) (a : ValueAt h s_0 c) : ValueAt (fun s => z * h s) s_0 (z * c)

theorem **holValueAt_const** (z s_0 : ℂ) : HolValueAt (fun _ => z) s_0 z

theorem **HolValueAt.add** (a : HolValueAt h s_0 c) (b : HolValueAt h' s_0 c') : HolValueAt (fun s => h s + h' s) s_0 (c + c')

theorem **HolValueAt.mul** (a : HolValueAt h s_0 c) (b : HolValueAt h' s_0 c') : HolValueAt (fun s => h s * h' s) s_0 (c * c')

theorem **HolValueAt.div** (a : HolValueAt h s_0 c) (b : HolValueAt h' s_0 c') (h0 : c' ≠ 0) : HolValueAt (fun s => h s / h' s) s_0 (c / c')

theorem **holValueAt_prod** {ι : Type*} (T : Finset ι) {f : ι → ℂ → ℂ} {z : ι → ℂ} (a : ∀ i ∈ T, HolValueAt (f i) s_0 (z i)) : HolValueAt (fun s => ∏ i ∈ T, f i s) s_0 (∏ i ∈ T, z i)

theorem **HolValueAt.comp_add** {z : ℂ} (a : HolValueAt h (s_0 + z) c) : HolValueAt (fun s => h (s + z)) s_0 c

theorem **PoleLE.mul_hol** (a : PoleLE h s_0 k) (e : CoeffAt h s_0 (−k) c) (b : HolValueAt h' s_0 c') : PoleLE (fun s => h s * h' s) s_0 k ∧ CoeffAt (fun s => h s * h' s) s_0 (−k) (c * c')

theorem **isOpen_contDom** (h : ℂ → ℂ) : IsOpen (contDom h)

theorem **IsMerExt.subset_contDom** (a : IsMerExt h U H) : U ⊆ contDom h

theorem **isRightDomain_contDom** (hs : s_0 ∈ contDom h) : IsRightDomain (contDom h)

theorem **CoeffAt.laurent_eq** (a : CoeffAt h s_0 n c) : laurent h s_0 n = c

theorem **coeffAt_laurent** (hs : s_0 ∈ contDom h) (n : ℤ) : CoeffAt h s_0 n (laurent h s_0 n)

theorem **coeffAt_iff** : CoeffAt h s_0 n c ↔ s_0 ∈ contDom h ∧ laurent h s_0 n = c

theorem **valueAt_iff** : ValueAt h s_0 c ↔ s_0 ∈ contDom h ∧ cont h s_0 = c

theorem **ValueAt.cont_eq** (a : ValueAt h s_0 c) : cont h s_0 = c

theorem **IsExtAt.cont_eventuallyEq** (a : IsExtAt h s_0 H) : cont h =ᶠ[$\mathcal{N}$[≠] s_0] H

theorem **meromorphicOn_cont** (h : ℂ → ℂ) : MeromorphicOn (cont h) (contDom h)

theorem **HolValueAt.analyticAt_cont** (a : HolValueAt h s_0 c) : AnalyticAt ℂ (cont h) s_0

theorem **eqOn_cont** (a : AnalyticOnNhd ℂ h (rightOf σ)) : EqOn (cont h) h (rightOf σ)

theorem **isMerExt_cont** (a : AnalyticOnNhd ℂ h (rightOf σ)) : IsMerExt h (contDom h) (cont h)

theorem **continues_iff** (a : AnalyticOnNhd ℂ h (rightOf σ)) : Continues h ↔ contDom h = univ

theorem **isMerExt_riemannZeta** : IsMerExt (fun s => ∑' n : ℕ, 1 / ((n : ℂ) + 1) ^ s) univ riemannZeta

Development.AbelianVarietyTheory.Definitions.Foundations.EventuallyNormFinLeOne

theorem **one_le_htOf** (N : Matrix n n A → ℝ≥0) (g : GL n A) : 1 ≤ htOf N g

theorem **le_htOf** (N : Matrix n n A → ℝ≥0) (g : GL n A) : N (g : Matrix n n A) ≤ htOf N g

theorem **inv_le_htOf** (N : Matrix n n A → ℝ≥0) (g : GL n A) : N ((g⁻¹ : GL n A) : Matrix n n A) ≤ htOf N g

theorem **htOf_inv** (N : Matrix n n A → ℝ≥0) (g : GL n A) : htOf N g⁻¹ = htOf N g

theorem **htOf_eq_one** {N : Matrix n n A → ℝ≥0} {g : GL n A} : htOf N g = 1 ↔ N (g : Matrix n n A) ≤ 1 ∧ N ((g⁻¹ : GL n A) : Matrix n n A) ≤ 1

theorem **htOf_mul_le** {N : Matrix n n A → ℝ≥0} (hN : ∀ a b, N (a * b) ≤ N a * N b) (g h : GL n A) : htOf N (g * h) ≤ htOf N g * htOf N h

theorem **absBar_mul** (x y : FvBar F v) : absBar F v (x * y) = absBar F v x * absBar F v y

theorem **absBar_add_le** (x y : FvBar F v) : absBar F v (x + y) ≤ max (absBar F v x) (absBar F v y)

theorem **absBar_algebraMap** (a : Fv F v) : absBar F v (algebraMap (Fv F v) (FvBar F v) a) = ‖a‖₊

theorem **absBar_neg** (x : FvBar F v) : absBar F v (-x) = absBar F v x

theorem **absBar_eq_zero** {x : FvBar F v} : absBar F v x = 0 ↔ x = 0

theorem **absBar_le_one_of_monic** {x : FvBar F v} (p : Polynomial ℤ) (hp : p.Monic) (hx : p.eval₂ (Int.castRingHom _) x = 0) : absBar F v x ≤ 1

@[simp] theorem **finEmb_tmul** (τ : E →ₐ[F] FvBar F v) (a : Fv F v) (x : E) : finEmb F v τ (a ⊗ₜ[F] x) = algebraMap (Fv F v) (FvBar F v) a * τ x

theorem **le_absFin** (τ : E →ₐ[F] FvBar F v) (y : Fv F v ⊗[F] E) : absBar F v (finEmb F v τ y) ≤ absFin F v y

theorem **absFin_le_iff** {y : Fv F v ⊗[F] E} {c : ℝ≥0} : absFin F v y ≤ c ↔ ∀ τ : E →ₐ[F] FvBar F v, absBar F v (finEmb F v τ y) ≤ c

theorem **absFin_mul_le** (x y : Fv F v ⊗[F] E) : absFin F v (x * y) ≤ absFin F v x * absFin F v y

theorem **absFin_add_le** (x y : Fv F v ⊗[F] E) : absFin F v (x + y) ≤ max (absFin F v x) (absFin F v y)

@[simp] theorem **absFin_zero** : absFin F v (0 : Fv F v ⊗[F] E) = 0

theorem **absFin_neg** (y : Fv F v ⊗[F] E) : absFin F v (-y) = absFin F v y

theorem **absFin_sum_le** {ι : Type} (s : Finset ι) (f : ι → Fv F v ⊗[F] E) {c : ℝ≥0} (h : ∀ i ∈ s, absFin F v (f i) ≤ c) : absFin F v (∑ i ∈ s, f i) ≤ c

theorem **absFin_tmul_one** (a : Fv F v) : absFin F v (a ⊗ₜ[F] (1 : E)) = ‖a‖₊

@[simp] theorem **absFin_one** : absFin F v (1 : Fv F v ⊗[F] E) = 1

theorem **absFin_tmul_le** (a : Fv F v) (x : $\mathcal{O}$ E) : absFin F v (a ⊗ₜ[F] (x : E)) ≤ ‖a‖₊

theorem **normFin_le_iff** {m n : Type} [Fintype m] [Fintype n] {y : Matrix m n (Fv F v ⊗[F] E)} {c : ℝ≥0} : normFin F v y ≤ c ↔ ∀ i j, absFin F v (y i j) ≤ c

theorem **absFin_le_normFin** {m n : Type} [Fintype m] [Fintype n] (y : Matrix m n (Fv F v ⊗[F] E)) (i : m) (j : n) : absFin F v (y i j) ≤ normFin F v y

theorem **normFin_mul_le** {m n p : Type} [Fintype m] [Fintype n] [Fintype p] (x : Matrix m n (Fv F v ⊗[F] E)) (y : Matrix n p (Fv F v ⊗[F] E)) : normFin F v (x * y) ≤ normFin F v x * normFin F v y

theorem **normFin_one_le** {n : Type} [Fintype n] [DecidableEq n] : normFin F v (1 : Matrix n n (Fv F v ⊗[F] E)) ≤ 1

@[simp] theorem **infEmb_tmul** (τ : EmbOver E w) (a : w.Completion) (x : E) : infEmb τ (a ⊗ₜ[F] x) = InfinitePlace.Completion.extensionEmbedding w a * τ.1 x

theorem **opNorm_mul_le** {n : Type} [Fintype n] [DecidableEq n] (x y : Matrix n n ℂ) : opNorm (x * y) ≤ opNorm x * opNorm y

theorem **opNorm_one_le** {n : Type} [Fintype n] [DecidableEq n] : opNorm (1 : Matrix n n ℂ) ≤ 1

theorem **normInf_le_iff** {n : Type} [Fintype n] [DecidableEq n] {y : Matrix n n (w.Completion ⊗[F] E)} {c : ℝ≥0} : normInf F w y ≤ c ↔ ∀ τ : EmbOver E w, opNorm (y.map (infEmb τ)) ≤ c

theorem **normInf_mul_le** {n : Type} [Fintype n] [DecidableEq n] (x y : Matrix n n (w.Completion ⊗[F] E)) : normInf F w (x * y) ≤ normInf F w x * normInf F w y

theorem **normInf_one_le** {n : Type} [Fintype n] [DecidableEq n] : normInf F w (1 : Matrix n n (w.Completion ⊗[F] E)) ≤ 1

theorem **opNorm_map_conj_le** {n : Type} [Fintype n] [DecidableEq n] (y : Matrix n n ℂ) : opNorm (y.map (starRingEnd ℂ)) ≤ opNorm y

theorem **infEmb_eq_or_conj** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (τ τ' : EmbOver E w) : infEmb τ' = infEmb τ ∨ infEmb τ' = (starRingEnd ℂ).comp (infEmb τ)

theorem **normInf_eq_opNorm** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (τ : EmbOver E w) {n : Type} [Fintype n] [DecidableEq n] (y : Matrix n n (w.Completion ⊗[F] E)) : normInf F w y = opNorm (y.map (infEmb τ))

theorem **eventually_absFin_le_one** (z : AF F ⊗[F] E) : ∀ᶠ v in Filter.cofinite, absFin F v (compFin F E v z) ≤ 1

theorem **eventually_normFin_le_one** (y : Matrix n n (AF F ⊗[F] E)) : ∀ᶠ v in Filter.cofinite, normFin F v (y.map (compFin F E v)) ≤ 1

theorem **finite_mulSupport_htFin** (g : GL n (AF F ⊗[F] E)) : (Function.mulSupport fun v => htFin F v (bc (compFin F E v) g)).Finite

theorem **one_le_heightGL** (g : GL n (AF F ⊗[F] E)) : 1 ≤ heightGL F g

theorem **heightGL_inv** (g : GL n (AF F ⊗[F] E)) : heightGL F g⁻¹ = heightGL F g

theorem **heightGL_eq_one_iff** {g : GL n (AF F ⊗[F] E)} : heightGL F g = 1 ↔ (∀ w, htInf F w (bc (compInf F E w) g) = 1) ∧ ∀ v, htFin F v (bc (compFin F E v) g) = 1

@[simp] theorem **heightGL_one** : heightGL F (1 : GL n (AF F ⊗[F] E)) = 1

theorem **heightGL_mul_le** (g h : GL n (AF F ⊗[F] E)) : heightGL F (g * h) ≤ heightGL F g * heightGL F h

theorem **heightGL_mul_of_eq_one** {k : GL n (AF F ⊗[F] E)} (hk : heightGL F k = 1) (g : GL n (AF F ⊗[F] E)) : heightGL F (g * k) = heightGL F g ∧ heightGL F (k * g) = heightGL F g

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsAHEqOfMemRatPts

theorem **JR_blkMat** (J : Matrix (Fin m) (Fin m) E) (J' : Matrix (Fin n) (Fin n) E) : JR (F

theorem **JR_dblGram** (J : Matrix (Fin n) (Fin n) E) : JR (F

theorem **val_detH** {J : Matrix (Fin m) (Fin m) E} (g : HA R σ J) : (detH R σ J g).1 = (matHom R σ J g).1.det

theorem **matHom_ioH** (g g' : HA R σ J) : matHom R σ Q (ioH R σ h (g, g')) = blkGL finSumFinEquiv (matHom R σ J g, matHom R σ J g')

theorem **ioH_one** : ioH R σ h (1, 1) = 1

theorem **ioH_injective** : Function.Injective (ioH R σ h)

theorem **continuous_ioH** [TopologicalSpace R] : Continuous (ioH R σ h)

theorem **ioH_mem_siegelH** {g g' : HA R σ J} : ioH R σ h (g, g') ∈ siegelH R σ Q ↔ g = g'

theorem **coe_ioDiag** (g : HA R σ J) : (ioDiag h g : HA R σ Q) = ioH R σ h (g, g)

theorem **aH_ioDiag** (g : HA R σ J) : aH R σ Q (ioDiag h g) = detH R σ J g

theorem **isUnit_det_JR** (hJ : J.det ≠ 0) : IsUnit (JR (F

theorem **sigR_det_quot_mul_aH** (hJ : J.det ≠ 0) (p : siegelH R σ Q) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ (quotMat finSumFinEquiv (matHom R σ Q p).1).det * (aH R σ Q p).1 = 1

theorem **sigR_invol** (hσ : ∀ x, σ (σ x) = x) (y : R ⊗[F] E) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ y) = y

theorem **conjUnit_detH** {J : Matrix (Fin m) (Fin m) E} (hJ : J.det ≠ 0) (g : HA R σ J) : conjUnit σ (detH R σ J g) = (detH R σ J g)⁻¹

theorem **mapH_ioH** (g g' : HA R σ J) : mapH σ Q φ (ioH R σ h (g, g')) = ioH R' σ h (mapH σ J φ g, mapH σ J φ g')

theorem **mapH_mem_siegelH** {g : HA R σ Q} (hg : g ∈ siegelH R σ Q) : mapH σ Q φ g ∈ siegelH R' σ Q

theorem **aH_mapH** (p : siegelH R σ Q) : aH R' σ Q (mapH σ Q φ p, mapH_mem_siegelH φ p.2) = Units.map (Algebra.TensorProduct.map φ (AlgHom.id F E)).toRingHom.toMonoidHom (aH R σ Q p)

theorem **diagH_mem_siegelH** [Nontrivial R] {γ : UJ (σ : E →+* E) Q} : diagH σ Q R γ ∈ siegelH R σ Q ↔ γ ∈ siegelRat σ Q

theorem **aH_diagH** {γ : UJ (σ : E →+* E) Q} (hγ : γ ∈ siegelRat σ Q) (hγ' : diagH σ Q R γ ∈ siegelH R σ Q) : aH R σ Q ⟨_, hγ'⟩ = Units.map (inclR (F

theorem **ioH_diagH** (γ γ' : UJ (σ : E →+* E) J) : ioH R σ h (diagH σ J R γ, diagH σ J R γ') = diagH σ Q R (ioU (σ : E →+* E) finSumFinEquiv h (γ, γ'))

theorem **proj_ioH** (v : HeightOneSpectrum (𝒪 F)) (g g' : GAd σ J) : proj σ Q v (ioH _ σ h (g, g')) = ioH _ σ h (proj σ J v g, proj σ J v g')

theorem **projArch_ioH** (w : InfinitePlace F) (g g' : GAd σ J) : projArch σ Q w (ioH _ σ h (g, g')) = ioH _ σ h (projArch σ J w g, projArch σ J w g')

theorem **emb_ioH** (v : HeightOneSpectrum (𝒪 F)) (g g' : GLoc σ J v) : emb σ Q v (ioH _ σ h (g, g')) = ioH _ σ h (emb σ J v g, emb σ J v g')

theorem **ioH_mem_ratPts** {g g' : GAd σ J} (hg : g ∈ AdelicU.ratPts σ J) (hg' : g' ∈ AdelicU.ratPts σ J) : ioH _ σ h (g, g') ∈ AdelicU.ratPts σ Q

@[simp] theorem **coe_projP** (v : HeightOneSpectrum (𝒪 F)) (p : siegelH (AF F) σ Q) : (projP v p : GLoc σ Q v) = proj σ Q v p

@[simp] theorem **coe_projArchP** (w : InfinitePlace F) (p : siegelH (AF F) σ Q) : (projArchP w p : GArch σ Q w) = projArch σ Q w p

theorem **aH_projP** (v : HeightOneSpectrum (𝒪 F)) (p : siegelH (AF F) σ Q) : aH _ σ Q (projP v p) = unitFin F E v (aH _ σ Q p)

theorem **aH_projArchP** (w : InfinitePlace F) (p : siegelH (AF F) σ Q) : aH _ σ Q (projArchP w p) = unitInf F E w (aH _ σ Q p)

theorem **mem_siegelH_adelic** {g : GAd σ Q} : g ∈ siegelH (AF F) σ Q ↔ (∀ w, projArch σ Q w g ∈ siegelH _ σ Q) ∧ ∀ v, proj σ Q v g ∈ siegelH _ σ Q

@[simp] theorem **coe_embP** (v : HeightOneSpectrum (𝒪 F)) (p : siegelH (Fv F v) σ Q) : (embP v p : GAd σ Q) = emb σ Q v p

@[simp] theorem **coe_embArchP** (w : InfinitePlace F) (p : siegelH w.Completion σ Q) : (embArchP w p : GAd σ Q) = embArch σ Q w p

theorem **aH_embP** (v : HeightOneSpectrum (𝒪 F)) (p : siegelH (Fv F v) σ Q) : aH _ σ Q (embP v p) = ideleAt F E v (aH _ σ Q p)

theorem **exists_aH_eq_of_mem_ratPts** (p : siegelH (AF F) σ Q) (hp : (p : GAd σ Q) ∈ AdelicU.ratPts σ Q) : ∃ x : Eˣ, aH _ σ Q p = Units.map (inclR (F

theorem **latLoc_sumLat** (L : Submodule (𝒪 E) (Fin m → E)) (L' : Submodule (𝒪 E) (Fin n → E)) (v : HeightOneSpectrum (𝒪 F)) : latLoc F (sumLat finSumFinEquiv L L') v = sumLat finSumFinEquiv (latLoc F L v) (latLoc F L' v)

theorem **mem_KAd** {g : GAd σ Q} : g ∈ KAd σ Q L a ↔ (∀ v, proj σ Q v g ∈ KFin σ Q L v) ∧ ∀ w, projArch σ Q w g ∈ KArch σ Q w (bc (compInf F E w) a)

theorem **emb_mem_KAd** {v : HeightOneSpectrum (𝒪 F)} {k : GLoc σ Q v} (hk : k ∈ KFin σ Q L v) : emb σ Q v k ∈ KAd σ Q L a

theorem **ioH_mem_KFin** {v : HeightOneSpectrum (𝒪 F)} {g g' : GLoc σ J v} : ioH _ σ h (g, g') ∈ KFin σ Q L v ↔ g ∈ stabLoc σ J L v ∧ g' ∈ stabLoc σ J L v

theorem **ioH_mem_KArch** {w : InfinitePlace F} {b : GL (Fin n) (w.Completion ⊗[F] E)} {g g' : GArch σ J w} : ioH _ σ h (g, g') ∈ KArch σ Q w b ↔ g ∈ stabArch σ J w b ∧ g' ∈ stabArch σ J w b

theorem **ioH_mem_KAd** {g g' : GAd σ J} : ioH _ σ h (g, g') ∈ KAd σ Q L a ↔ g ∈ levelAll σ J L a ∧ g' ∈ levelAll σ J L a

theorem **isLatticeFrame_dbl** (ha : IsLatticeFrame L a) : IsLatticeFrame (sumLat finSumFinEquiv L L) (blkGL finSumFinEquiv (a, a))

theorem **isMajorantFrame_dbl** (ha : IsMajorantFrame σ J a) : IsMajorantFrame σ Q (blkGL finSumFinEquiv (a, a))

theorem **isSelfDualAt_dbl** {v : HeightOneSpectrum (𝒪 F)} (hL : IsSelfDualAt σ J L v) : IsSelfDualAt σ Q (sumLat finSumFinEquiv L L) v

theorem **isCompact_levelAll** (h : IsCMExt F E σ) (J : Matrix (Fin m) (Fin m) E) (L : Submodule (𝒪 E) (Fin m → E)) [Submodule.IsLattice E L] (a : GL (Fin m) (AF F ⊗[F] E)) : IsCompact (levelAll σ J L a : Set (GAd σ J))

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsAuxiliaryPlaceNotMemFinset

theorem **archMat_apply** {w : InfinitePlace F} (τ : EmbOver E w) (g : GArch σ J w) : archMat σ J τ g = (matHom _ σ J g).1.map (infEmb τ)

theorem **archMat_eq_archEquiv** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) (g : GArch σ J w) : archMat σ J τ g = (archEquiv J h τ g).1.1

theorem **continuous_infEmb** {w : InfinitePlace F} (τ : EmbOver E w) : Continuous (infEmb τ)

theorem **continuous_archMat** {w : InfinitePlace F} (τ : EmbOver E w) : Continuous (archMat σ J τ)

theorem **continuous_archCoord** (w : InfinitePlace F) : Continuous (archCoord σ J w)

theorem **continuous_infCoord** : Continuous (infCoord σ J)

theorem **isEmbedding_archMat** (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) : Topology.IsEmbedding (matCoord (archMat σ J τ))

theorem **ccArch_eq** (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) : CcArch σ J w = CcMat (archMat σ J τ)

theorem **tensor_eq_infFin** : tensor σ J fa ff = infFin σ J (tensorInf σ J fa) (tensorFin σ J ff)

theorem **infFin_mem_ccAd** {a : GInf σ J → ℂ} {b : GFin σ J → ℂ} (ha : a ∈ CcInf σ J) (hb : b ∈ CcFin σ J) : infFin σ J a b ∈ CcAd σ J

theorem **continuous_of_mem_ccAd** {f : GAd σ J → ℂ} (h : f ∈ CcAd σ J) : Continuous f ∧ HasCompactSupport f

theorem **tensorInf_mem** (h : ∀ w, fa w ∈ CcArch σ J w) : tensorInf σ J fa ∈ CcInf σ J

theorem **unitLoc_mem** (v : HeightOneSpectrum (𝒪 F)) : unitLoc σ J L v ∈ CcLoc σ J v

theorem **isFinFamily_unit** : IsFinFamily σ J L (unitLoc σ J L)

theorem **IsFinFamily.of_lattice** (h : IsFinFamily σ J L ff) (L' : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L'] : IsFinFamily σ J L' ff

theorem **IsFinFamily.finite_mulSupport** (h : IsFinFamily σ J L ff) (g : GFin σ J) : (Function.mulSupport fun v => ff v (projFinLoc σ J v g)).Finite

theorem **IsFinFamily.tensorFin_ne_zero_iff** (h : IsFinFamily σ J L ff) (g : GFin σ J) : tensorFin σ J ff g ≠ 0 ↔ ∀ v, ff v (projFinLoc σ J v g) ≠ 0

theorem **IsFinFamily.tensorFin_mem** (h : IsFinFamily σ J L ff) : tensorFin σ J ff ∈ CcFin σ J

theorem **tensor_mem_ccAd** (ha : ∀ w, fa w ∈ CcArch σ J w) (hf : IsFinFamily σ J L ff) : tensor σ J fa ff ∈ CcAd σ J

theorem **IsPureTensor.mem_ccAd** {f : GAd σ J → ℂ} (h : IsPureTensor σ J L f) : f ∈ CcAd σ J

theorem **IsFinFamily.starFn_tensor** (h : IsFinFamily σ J L ff) : IsFinFamily σ J L (fun v => starFn (ff v)) ∧ starFn (tensor σ J fa ff) = tensor σ J (fun w => starFn (fa w)) fun v => starFn (ff v)

theorem **integral_tensor** {μa : ∀ w, Measure (GArch σ J w)} {μf : ∀ v, Measure (GLoc σ J v)} {μ : Measure (GAd σ J)} (hμ : IsRestrictedProd σ J L μa μf μ) (ha : ∀ w, fa w ∈ CcArch σ J w) (hf : IsFinFamily σ J L ff) : ∫ g, tensor σ J fa ff g ∂μ = (∏ w, ∫ y, fa w y ∂μa w) * ∏ᶠ v, ∫ y, ff v y ∂μf v

theorem **exists_isPureTensor** (h : IsCMExt F E σ) : ∃ f, IsPureTensor σ J L f ∧ f 1 = 1

theorem **strictTorus_nonempty** {k : Type} [Field k] (n : ℕ) (ϖ : kˣ) : (strictTorus n ϖ).Nonempty

theorem **exists_isUnif** (v : HeightOneSpectrum (𝒪 F)) : ∃ ϖ : (Fv F v)ˣ, IsUnif ϖ

theorem **IsUnif.valuation_eq** {v : HeightOneSpectrum (𝒪 F)} {ϖ : (Fv F v)ˣ} (h : IsUnif ϖ) : letI

theorem **red_surjective** : Function.Surjective s.red

theorem **red_eq_one_iff** (k : stabLoc σ J L v) : s.red k = 1 ↔ k.1 ∈ stabLocOne σ J s.emb L

theorem **mem_stabLoc_iff** {g : GLoc σ J v} : letI

theorem **mem_iwahori** {g : GLoc σ J v} : g ∈ s.iwahori ↔ g ∈ stabLoc σ J L v ∧ ∀ i j, j < i → Valued.v ((s.equiv g).1 i j) < 1

theorem **iwahori_le** : s.iwahori ≤ stabLoc σ J L v

theorem **mem_iwahoriCoset** {t : Fin n → (Fv F v)ˣ} {g : GLoc σ J v} : letI

theorem **isOpen_iwahori** : IsOpen (s.iwahori : Set (GLoc σ J v))

theorem **isCompact_iwahori** : IsCompact (s.iwahori : Set (GLoc σ J v))

theorem **cosetFn_mem** (t : Fin n → (Fv F v)ˣ) : s.cosetFn t ∈ CcLoc σ J v

theorem **cosetFn_diag** (t : Fin n → (Fv F v)ˣ) : s.cosetFn t (s.diag t) = 1

theorem **idem_mem** [Submodule.IsLattice E L] : s.idem μ τ ∈ CcLoc σ J v

theorem **starFn_idem** [FiniteDimensional ℂ W] : starFn (s.idem μ τ) = s.idem μ τ

theorem **IsTestFin.isFinFamily** (h : IsTestFin T U μf ff) : IsFinFamily σ J L ff

theorem **IsTestFn.isPureTensor** {f : GAd σ J → ℂ} (h : IsTestFn Ψ D T U μa μf f) : IsPureTensor σ J L f

theorem **exists_isTestFin** (hu : U.u ∉ T.S) (μf : ∀ v, Measure (GLoc σ J v)) (h : ∀ v : T.S, GLoc σ J v → ℂ) (hh : ∀ v : T.S, h v ∈ CcLoc σ J v) {t : Fin n → (Fv F U.u)ˣ} (ht : t ∈ strictTorus n U.ϖ) {rest : ∀ v, GLoc σ J v → ℂ} (hr : IsFinFamily σ J L rest) : ∃ ff, IsTestFin T U μf ff ∧ (∀ v : T.S, ff v = conv (μf v) (conv (μf v) (T.idem μf v) (h v)) (T.idem μf v)) ∧ ff U.u = U.frame.cosetFn t ∧ ∀ v ∉ T.S, v ≠ U.u → ff v = rest v

theorem **IsFinFamily.transfer** {n' : ℕ} {J' : Matrix (Fin n') (Fin n') E} {L' : Submodule (𝒪 E) (Fin n' → E)} (h : IsFinFamily σ J L ff) (j : ∀ v, GLoc σ J v ≃ₜ* GLoc σ J' v) (hj : ∀ᶠ v in cofinite, ∀ g, j v g ∈ stabLoc σ J' L' v ↔ g ∈ stabLoc σ J L v) : IsFinFamily σ J' L' (transfer j ff)

theorem **exists_split_place_notMem** (h : IsCMExt F E σ) (S : Finset (HeightOneSpectrum (𝒪 F))) : ∃ v ∉ S, Nonempty (E →ₐ[F] Fv F v)

theorem **exists_auxiliaryPlace_not_mem_finset** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (hL : ∀ v, IsSelfDualAt σ J L v) (S : Finset (HeightOneSpectrum (𝒪 F))) : ∃ U : UnitaryAuxiliaryPlaceData σ J L, U.u ∉ S

theorem **exists_testData** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (hL : ∀ v, IsSelfDualAt σ J L v) (μf : ∀ v, Measure (GLoc σ J v)) : ∃ (T : TypeData σ J L) (U : UnitaryAuxiliaryPlaceData σ J L) (ff : ∀ v, GLoc σ J v → ℂ), T.S.Nonempty ∧ IsTestFin T U μf ff

theorem **exists_isTestFn_ne_zero** (h : IsCMExt F E σ) (μa : ∀ w, Measure (GArch σ (1 : Matrix (Fin n) (Fin n) E) w)) (μf : ∀ v, Measure (GLoc σ (1 : Matrix (Fin n) (Fin n) E) v)) [∀ w, (μa w).IsHaarMeasure] [∀ v, (μf v).IsHaarMeasure] (Ψ : ∀ w : InfinitePlace F, EmbOver

E w) : ∃ (T : TypeData σ 1 (stdLattice E n)) (U : UnitaryAuxiliaryPlaceData σ 1 (stdLattice E n)) (f : GAd σ 1 → ℂ), T.S.Nonempty ∧ IsTestFn Ψ (fun _ => charRep 1) T U μa μf f ∧ f ≠ 0

theorem **restrGroup_cent** (γ : GAd σ J) : GlobalTheory.RestrMeas.IsRestrGroup (fun w => centMap (projArch σ J w) γ) (fun v => centMap (proj σ J v) γ) fun v => (stabLoc σ J L v).subgroupOf (cent (proj σ J v γ))

theorem **existsUnique_isCentProd** (γ : GAd σ J) (dta : ∀ w, Measure (cent (projArch σ J w γ))) (dtf : ∀ v, Measure (cent (proj σ J v γ))) [∀ w, (dta w).IsHaarMeasure] [∀ v, (dtf v).IsHaarMeasure] (h : ∀ᶠ v in cofinite, dtf v ((stabLoc σ J L v).subgroupOf (cent (proj σ J v γ))) = 1) : ∃! dt, IsCentProd L γ dta dtf dt

theorem **isUnitaryRep_repVia** {H X : Type} [Group H] [TopologicalSpace H] [NormedAddCommGroup X] [InnerProductSpace ℂ X] {m : ℕ} {G : Set (Matrix (Fin m) (Fin m) ℂ)} {ρ : Matrix (Fin m) (Fin m) ℂ → (X →L[ℂ] X)} (h : IsURepOn G ρ) {ι : H →* Matrix (Fin m) (Fin m) ℂ} (hι : ∀ g, ι g ∈ G) (hc : Continuous ι) : IsUnitaryRep (repVia h ι hι)

theorem **isTopIrreducible_repVia** {H X : Type} [Group H] [NormedAddCommGroup X] [InnerProductSpace ℂ X] [Nontrivial X] {m : ℕ} {G : Set (Matrix (Fin m) (Fin m) ℂ)} {ρ : Matrix (Fin m) (Fin m) ℂ → (X →L[ℂ] X)} (h : IsURepOn G ρ) {ι : H →* Matrix (Fin m) (Fin m) ℂ} (hι : ∀ g, ι g ∈ G) (hs : ∀ x ∈ G, ∃ g, ι g = x) (hi : IsIrredOn G ρ) : IsTopIrreducible (repVia h ι hι)

theorem **isHermND_gram** : IsHermND D.sp.cm.sig V.gram

theorem **exists_splitAt** {v : HeightOneSpectrum (𝒪 D.host.F)} (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : ∃ s : SplitAt D.sp.cm.sig V.gram (frameLat V) v, s.emb = τ

theorem **exists_testDataG** (μf : ∀ v, Measure (GLoc D.sp.cm.sig V.gram v)) : ∃ (T : TypeDataG V) (U : AuxiliaryPlaceHermitianSpace V) (ff : ∀ v, GLoc D.sp.cm.sig V.gram v → ℂ), T.S.Nonempty ∧ IsTestFin T U μf ff

theorem **conj_mem_UJset_iff** (ρ : D.what₂.Tau) (x : Matrix (Fin V.rank) (Fin V.rank) ℂ) : M.fr ρ * x * (M.fr ρ)⁻¹ ∈ UJset (M.ε ρ) ↔ x ∈ Formalism.QuantumDeterminants.UH (gramAt D V ρ)

theorem **frameMat_mem** (ρ : D.what₂.Tau) (g : GArch D.sp.cm.sig V.gram (placeOf D ρ)) : frameMat M ρ g ∈ UJset (M.ε ρ)

theorem **dsRep_apply** (ρ : D.what₂.Tau) (g : GArch D.sp.cm.sig V.gram (placeOf D ρ)) : dsRep M ρ g = M.rep ρ (matAt V ρ g)

theorem **isUnitaryRep_dsRep** (ρ : D.what₂.Tau) : IsUnitaryRep (dsRep M ρ)

theorem **isTopIrreducible_dsRep** (ρ : D.what₂.Tau) : IsTopIrreducible (dsRep M ρ)

theorem **IsTestArchG.mem** {fa : ∀ w, GArch D.sp.cm.sig V.gram w → ℂ} (h : IsTestArchG M μa fa) (w : InfinitePlace D.host.F) : fa w ∈ CcArch D.sp.cm.sig V.gram w

theorem **IsTestFnG.isPureTensor** {T : TypeDataG V} {U : AuxiliaryPlaceHermitianSpace V} {f : GA D V → ℂ} (h : IsTestFnG M T U μa μf f) : IsPureTensor D.sp.cm.sig V.gram (frameLat V) f ∧ f ∈ CcGA V

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsBaseTwoWords

theorem **span_image_span** (s : Set K) : Submodule.span (𝒪 K') (f '' (Submodule.span (𝒪 K) s : Set K)) = Submodule.span (𝒪 K') (f '' s)

theorem **extend_span** (s : Set K) : extend f (Submodule.span (𝒪 K) s) = Submodule.span (𝒪 K') (f '' s)

theorem **conjugate_emb** (g : R.GG) : ComplexEmbedding.conjugate (H.emb g) = H.emb (g * R.cc)

theorem **conjugate_emb_apply** (g : R.GG) (x : H.E) : ComplexEmbedding.conjugate (H.emb g) x = H.emb g (H.conj x)

theorem **ex_emb_iff** (g : R.GG) : (∃ t : Tau R, H.emb t = H.emb g) ↔ g ∈ R.Psi

theorem **emb_cm** (σ : H.E →+* ℂ) : (∃ t : Tau R, H.emb t = σ) ↔ ¬ ∃ t : Tau R, H.emb t = ComplexEmbedding.conjugate σ

theorem **sum_emb_bijective** : Function.Bijective (Sum.elim (fun t : Tau R => H.emb t) fun t : Tau R => ComplexEmbedding.conjugate (H.emb t))

theorem **typeExp_emb** (p q : ℕ) (t : Tau R) : typeExp H p q (H.emb t) = p

theorem **typeExp_conj** (p q : ℕ) (t : Tau R) : typeExp H p q (ComplexEmbedding.conjugate (H.emb t)) = q

theorem **typeExp_add** (p q : ℕ) (σ : H.E →+* ℂ) : typeExp H p q σ + typeExp H p q (ComplexEmbedding.conjugate σ) = p + q

theorem **prod_typeExp** {M : Type} [CommMonoid M] (p q : ℕ) (F : (H.E →+* ℂ) → M) : ∏ σ, F σ ^ typeExp H p q σ = ∏ t : Tau R, (F (H.emb t) ^ p * F (ComplexEmbedding.conjugate (H.emb t)) ^ q)

theorem **isAlgebraic_exp** (N : ℕ) : IsAlgebraic ℚ (Complex.exp (2 * Real.pi * Complex.I / N))

theorem **exists_gens** (E : Type) [Field E] [NumberField E] : ∃ S : Finset ℂ, (∀ z ∈ S, IsAlgebraic ℚ z) ∧ ∀ K : IntermediateField ℚ ℂ, (∀ z ∈ S, z ∈ K) → ∀ (τ : E →+* ℂ) x, τ x ∈ K

theorem **isTotallyComplex_of_I** (K : IntermediateField ℚ ℂ) [NumberField K] (h : Complex.I ∈ K) : IsTotallyComplex K

theorem **image_top** {E : Type} [Field E] (K : IntermediateField ℚ ℂ) (𝔞 : Ideal (𝒪 E)) (τ : E →+* ℂ) (e : E → K) (he : ∀ a, (e a : ℂ) = τ a) : extend (toTop K) (Submodule.span (𝒪 K) {x : K | ∃ a ∈ 𝔞, (x : ℂ) = τ (a : E)}) = Submodule.span (𝒪 (⊤ : IntermediateField ℚ K)) {x : (⊤ : IntermediateField ℚ K) | ∃ a ∈ 𝔞, (x : K) = e (a : E)}

theorem **exists_base** : ∀ (B : Pre) (S₀ S₁ S₂ : Finset ℂ), (∀ z ∈ S₀, IsAlgebraic ℚ z) → (∀ z ∈ S₁, IsAlgebraic ℚ z) → (∀ z ∈ S₂, IsAlgebraic ℚ z) → ∃ B' : Base, B'.toPre = B ∧ (∀ z ∈ S₀, ∃ x ∈ B'.Ks, B'.embLL x = z) ∧ (∀ z ∈ S₁, ∃ x ∈ B'.LL0, B'.embLL x = z) ∧ ∀ z ∈ S₂, ∃ x : B'.LL, B'.embLL x = z

theorem **GlobalTheory.Mem.exists_base_two_words** : ∃ B : GlobalTheory.Mem.Base, Fintype.card B.Word = 2 ∧ Function.Injective B.hcW

theorem **GlobalTheory.Mem.nonempty_base** : Nonempty GlobalTheory.Mem.Base

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsCharIdemOfIsSeries

theorem **isUp_iff_isUniUp** (u : GG g N) : IsUp g N u ↔ IsUniUp u.val

theorem **coeff_eDat** (P : Matrix (Fin N) (Fin N) (ZMod g) → Prop) (φ : Matrix (Fin N) (Fin N) (ZMod g) → ZMod g) (ψ : AddChar (ZMod g) ℂ) (x : GG g N) : (eDat g N P φ ψ).coeff x = eps (fun u : GG g N => P u.val) (fun u => ψ (φ u.val)) x

theorem **sum_ite_entN** (M : Matrix (Fin N) (Fin N) (ZMod g)) (a : Fin N) (c : ℕ) : (∑ b : Fin N, if (b : ℕ) = c then M a b else 0) = entN M a c

private theorem **sum_superdiag_eq_sum_entN** (M : Matrix (Fin N) (Fin N) (ZMod g)) (hN : 0 < N) : (∑ a : Fin N, ∑ b : Fin N, if (b : ℕ) = a + 1 then M a b else 0) = ∑ i ∈ Finset.range (N - 1), Formalism.MainNotions.ArithmeticCylinders.entN M i (i + 1)

theorem **coeff_eU** (ψ : AddChar (ZMod g) ℂ) (hψ : ψ ≠ 1) (s k : ℕ) (h : s + k < N) (x : GG g N) : (eU g N ψ).coeff x = (levOfChar ψ hψ N false s k h).e x

private theorem **sum_range_eq_sum_split_of_last_eq_zero** {R : Type} [AddCommMonoid R] (f F : ℕ → R) (s k N : ℕ) (hN : N = k + 2 * s + 1) (hF : ∀ i, F i = if i + 1 ≠ s ∧ ¬(0 < k ∧ i + 1 = s + k) then f i else 0) (hf : f (N - 1) = 0) : ∑ i ∈ Finset.range N, F i = ∑ i ∈ Finset.range (s - 1), f i + (if 1 ≤ s then f (s + k) else 0) + ∑ i ∈ Finset.range (s - 1), f (s + k + 1 + i) + ∑ a ∈ Finset.range (k - 1), f (s + a)

theorem **sum_edge_eq_phiJ** (ψ : AddChar (ZMod g) ℂ) (hψ : ψ ≠ 1) (s k : ℕ) (hN : N = k + 2 * s + 1) (h : s + k < N) (M : Matrix (Fin N) (Fin N) (ZMod g)) : (∑ a, ∑ b, if (levOfChar ψ hψ N true s k h).Edge a b then M a b else 0) = phiJ k s M

theorem **coeff_eJ** (ψ : AddChar (ZMod g) ℂ) (hψ : ψ ≠ 1) (s k : ℕ) (hN : N = k + 2 * s + 1) (h : s + k < N) (x : GG g N) : (eJ g N k s ψ).coeff x = (levOfChar ψ hψ N true s k h).e x

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsCharsPrime

theorem **exists_unif** {T : CoreJ D} (hT : LawsJ T) {j : ℕ} (hj : InR D j) {l : Level T.Q.H} (hA : l.A = 1) (Xo : Moduli (T.Q.Dw j) l) (c : Coeff l) : ∃ (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ D.R.r) (dict : ∀ σ, GA D (D.sp.W j) → Σ k, Frame ((U σ).X k).Ψ D.R.r), (∀ σ, GoodEmb c σ → GlobalTheory.IndefiniteRowSideFullHecke.HalfLatticeTypes.IsUnifAt (U σ) (D.sp.W j) (dict σ) (T.iota j)) ∧ ∀ σ, GoodEmb c σ → ∀ k la, ∀ γ ∈ ((U σ).X k).Γ, ((U σ).X k).χ la γ = 1

theorem **exists_deepLevel** (L : SizeData₂ D) {Q : Par D} (hQ : Q.Pins) (h5 : 5 ≤ D.host.splitBound) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) : ∃ l : Level Q.H, (l.M = levDeep D L ∧ l.A = 1 ∧ Q.l.Dvd l) ∧ IsTypeLevel l

theorem **exists_deepField** (T : CoreJ D) {j : ℕ} (hpos : 1 ≤ (T.Q.Dw j).p + (T.Q.Dw j).q) {l : Level T.Q.H} (Xo : Moduli (T.Q.Dw j) l) {n : ℕ} (hn : n ≠ 0) : ∃ (c : Coeff l) (le : T.Q.c.F ≤ c.F), (letI

theorem **exists_chars_prime** {K : Type} [Field K] {g : ℕ} [Fact g.Prime] {ζ : K} (h : IsPrimitiveRoot ζ (g * (g - 1))) : (∃ ψ : AddChar (ZMod g) K, ψ ≠ 1) ∧ ∃ om : (ZMod g)ˣ →* Kˣ, Function.Injective om

theorem **exists_chars** {K : Type} [Field K] {ζ : K} (h : IsPrimitiveRoot ζ (∏ k, D.host.g k * (D.host.g k - 1))) : (∃ ψ : ∀ v : Sq D.host, AddChar (ZMod (D.host.g v.1)) K, IsGenericFamily ψ) ∧ ∃ om : ∀ k : Fin 2, (ZMod (D.host.g k))ˣ →* Kˣ, IsGeneratorFamily om

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsEqSubOfMemCob

theorem **natCast_ne_zero** : (l : Qbar l) ≠ 0

theorem **norm_natCast_lt_one** : ‖(l : Qbar l)‖ < 1

theorem **pow_mul_lamv** (m : ℕ) (s : G ℚ Ω) : (l : Qbar l) ^ m * lamv l m s = algebraMap ℚ_[l] (Qbar l) ((cycZ ℚ Ω l s : ℤ_[l]ˣ) : ℤ_[l]) - 1

@[simp] theorem **lamv_one** (m : ℕ) : lamv l m (1 : G ℚ Ω) = 0

theorem **lamv_mul** (m : ℕ) (s t : G ℚ Ω) : lamv l m (s * t) = lamv l m s + lamv l m t + (l : Qbar l) ^ m * lamv l m s * lamv l m t

theorem **exists_lamv_eq** {m : ℕ} {s : G ℚ Ω} (hs : cycMod ℚ Ω l m s = 1) : ∃ a : ℤ_[l], ((cycZ ℚ Ω l s : ℤ_[l]ˣ) : ℤ_[l]) = 1 + (l : ℤ_[l]) ^ m * a ∧ lamv l m s = algebraMap ℤ_[l] (Qbar l) a

theorem **norm_le_one_of_mem** {c : Coef l} {a : Qbar l} (ha : a ∈ c.O) : ‖a‖ ≤ 1

theorem **norm_lamv_le_one** {m : ℕ} {s : G ℚ Ω} (hs : cycMod ℚ Ω l m s = 1) : ‖lamv l m s‖ ≤ 1

theorem **sub_mem_cob** {w : Ω} (hw : w ∈ A) : (fun s : G ℚ Ω => j (s w) - j w) ∈ cob c A j

theorem **mem_cob_iff** {f : G ℚ Ω → Qbar l} : f ∈ cob c A j ↔ ∃ (q : ℕ) (o : Fin q → Qbar l) (w : Fin q → Ω), (∀ i, o i ∈ c.O ∧ w i ∈ A) ∧ ∀ s, f s = ∑ i, o i * (j (s (w i)) - j (w i))

theorem **mul_mem_cob** {f : G ℚ Ω → Qbar l} (hf : f ∈ cob c A j) {a : Qbar l} (ha : a ∈ c.O) : (fun s => a * f s) ∈ cob c A j

theorem **apply_one_of_mem_cob** {f : G ℚ Ω → Qbar l} (hf : f ∈ cob c A j) : f 1 = 0

theorem **exists_eq_sub_of_mem_cob** (hj : ReciprocityTheory.Neutral.Cmp.Compat l A j) {f : G ℚ Ω → Qbar l} (hf : f ∈ cob c A j) : ∃ x : Qbar l, ‖x‖ ≤ 1 ∧ ∀ (σ : Qbar l ≃+* Qbar l) (s : G ℚ Ω), (∀ w, j (s w) = σ (j w)) → (∀ a ∈ c.O, σ a = a) → f s = σ x - x

theorem **compat** (H : HTCoIn c A j Z m c₀ y h) : ReciprocityTheory.Neutral.Cmp.Compat l A j

theorem **mono** (H : HTCoIn c A j Z m c₀ y h) {c₁ : ℕ} (hc : c₀ ≤ c₁) : HTCoIn c A j Z m c₁ y h

theorem **add** (H : HTCoIn c A j Z m c₀ y h) (H' : HTCoIn c A j Z' m c₀ y h') : HTCoIn c A j (fun r s x => Z r s x + Z' r s x) m c₀ y (h + h')

theorem **mul_left** (H : HTCoIn c A j Z m c₀ y h) {a : Qbar l} (ha : a ∈ c.O) : HTCoIn c A j (fun r s x => a * Z r s x) m c₀ y (a * h)

theorem **sub** (H : HTCoIn c A j Z m c₀ y h) (H' : HTCoIn c A j Z' m c₀ y h') : HTCoIn c A j (fun r s x => Z r s x - Z' r s x) m c₀ y (h - h')

theorem **congr** (H : HTCoIn c A j Z m c₀ y h) {y' : (ZMod (l ^ m))ˣ} {g : G ℚ Ω → Qbar l} (hg : g ∈ cob c A j) (hZ : ∀ s ∈ Ine (F

theorem **cg_one** (H : HTCoIn c A j Z m c₀ y h) : Cg (m - c₀) (Z m 1 y) 0

theorem **htCoIn_lamv** (hj : ReciprocityTheory.Neutral.Cmp.Compat l A j) {f : G ℚ Ω → Qbar l} (hf : f ∈ cob c A j) (m c₀ : ℕ) (y : (ZMod (l ^ m))ˣ) (h : Qbar l) : HTCoIn c A j (fun r s _ => h * lamv l r s + f s) m c₀ y h

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsFinsetOfEventually

theorem **mem_cyl** {S : Finset V} {A : ∀ w, Set (Ya w)} {B : ∀ v, Set (Yf v)} {x : X} : x ∈ cyl pa pf K S A B ↔ (∀ w, pa w x ∈ A w) ∧ (∀ v ∈ S, pf v x ∈ B v) ∧ ∀ v ∉ S, pf v x ∈ K v

theorem **cyl_self** (S : Finset V) (A : ∀ w, Set (Ya w)) : cyl pa pf K S A K = {x | (∀ w, pa w x ∈ A w) ∧ ∀ v, pf v x ∈ K v}

theorem **cyl_eq_of_subset** {S T : Finset V} (h : S ⊆ T) (hK : ∀ v ∉ T, K v = K' v) (A : ∀ w, Set (Ya w)) (B : ∀ v, Set (Yf v)) : cyl pa pf K' S A B = cyl pa pf K T A fun v => if v ∈ S then B v else K' v

theorem **exists_finset_of_eventually** {p : V → Prop} (h : ∀ᶠ v in cofinite, p v) : ∃ S : Finset V, ∀ v ∉ S, p v

theorem **eventually_of_finset** {p : V → Prop} (S : Finset V) (h : ∀ v ∉ S, p v) : ∀ᶠ v in cofinite, p v

theorem **isCylProd_of_large** (hK' : ∀ v, MeasurableSet (K' v)) (S₀ : Finset V) (h1 : ∀ v ∉ S₀, μf v (K' v) = 1) (hK : ∀ v ∉ S₀, K v = K' v) (h : ∀ S : Finset V, S₀ ⊆ S → ∀ (A : ∀ w, Set (Ya w)) (B : ∀ v, Set (Yf v)), (∀ w, MeasurableSet (A w)) → (∀ v ∈ S, MeasurableSet (B v)) → μ (cyl pa pf K S A B) = (∏ w, μa w (A w)) * ∏ v ∈ S, μf v (B v)) : IsCylProd pa pf K' μa μf μ

theorem **IsCylProd.congr** (h : IsCylProd pa pf K μa μf μ) (hK' : ∀ v, MeasurableSet (K' v)) (hK : ∀ᶠ v in cofinite, K v = K' v) : IsCylProd pa pf K' μa μf μ

theorem **IsCylProd.measure_level** (h : IsCylProd pa pf K μa μf μ) (hK : ∀ v, MeasurableSet (K v)) {A : ∀ w, Set (Ya w)} (hA : ∀ w, MeasurableSet (A w)) : μ {x | (∀ w, pa w x ∈ A w) ∧ ∀ v, pf v x ∈ K v} = (∏ w, μa w (A w)) * ∏ᶠ v, μf v (K v)

theorem **IsCylProd.smul** (h : IsCylProd pa pf K μa μf μ) (hK : ∀ v, MeasurableSet (K v)) (c : W → ℝ≥0∞) (d : V → ℝ≥0∞) (T : Finset V) (hd : ∀ v ∉ T, d v = 1) : IsCylProd pa pf K (fun w => c w • μa w) (fun v => d v • μf v) ((((∏ w, c w) * ∏ v ∈ T, d v) • μ)

theorem **IsCylProd.map** {X' : Type} [MeasurableSpace X'] (h : IsCylProd pa pf K μa μf μ) (e : X ≃ᵐ X') {pa' : ∀ w, X' → Ya w} {pf' : ∀ v, X' → Yf v} (ha : ∀ w x, pa' w (e x) = pa w x) (hf : ∀ v x, pf' v (e x) = pf v x) : IsCylProd pa' pf' K μa μf (μ.map e)

theorem **exists_extend_one** {V' : Type} {j : V' → V} (hj : Function.Injective j) (y : ∀ v', Yf (j v')) : ∃ z : ∀ v, Yf v, (∀ v', z (j v') = y v') ∧ ∀ v ∉ range j, z v = 1

theorem **IsRestrGroup.subgroup** [∀ w, T1Space (Ya w)] [∀ v, T1Space (Yf v)] (h : IsRestrGroup pa pf K) {W' V' : Type} {i : W' → W} {j : V' → V} (hi : Function.Injective i) (hj : Function.Injective j) (H : Subgroup X) (hH : ∀ x, x ∈ H ↔ (∀ w ∉ range i, pa w x = 1) ∧ ∀ v ∉ range j, pf v x = 1) : IsRestrGroup (fun w => (pa (i w)).comp H.subtype) (fun v => (pf (j v)).comp H.subtype) fun v => K (j v)

theorem **IsRestrGroup.noArch** [IsEmpty W] (h : IsRestrGroup pa pf K) : IsRestrGroup (noArch X) pf K

theorem **mem_levelAway** {S : Finset V} {x : X} : x ∈ levelAway pf K S ↔ ∀ v ∉ S, pf v x ∈ K v

theorem **IsRestrGroup.isOpen_levelAway** [IsTopologicalGroup X] (h : IsRestrGroup pa pf K) (S : Finset V) : IsOpen (levelAway pf K S : Set X)

theorem **isHaarProdFin_iff** : IsHaarProdFin pf K μf μ ↔ (∀ v, (μf v).IsHaarMeasure) ∧ μ.IsHaarMeasure ∧ (∀ᶠ v in cofinite, μf v (K v) = 1) ∧ ∀ S : Finset V, (∀ v ∉ S, μf v (K v) = 1) → ∀ B : ∀ v, Set (Yf v), (∀ v ∈ S, MeasurableSet (B v)) → μ {x | (∀ v ∈ S, pf v x ∈ B v) ∧ ∀ v ∉ S, pf v x ∈ K v} = ∏ v ∈ S, μf v (B v)

theorem **IsRestrGroup.measurableSet_cyl** (h : IsRestrGroup pa pf K) {S : Finset V} {A : ∀ w, Set (Ya w)} {B : ∀ v, Set (Yf v)} (hA : ∀ w, MeasurableSet (A w)) (hB : ∀ v ∈ S, MeasurableSet (B v)) : MeasurableSet (cyl (fun w => pa w) (fun v => pf v) (fun v => (K v : Set (Yf v))) S A B)

theorem **IsHaarProd.smul** (h : IsHaarProd pa pf K μa μf μ) (hK : ∀ v, IsOpen (K v : Set (Yf v))) (c : W → ℝ≥0) (d : V → ℝ≥0) (hc : ∀ w, c w ≠ 0) (hd0 : ∀ v, d v ≠ 0) (T : Finset V) (hd : ∀ v ∉ T, d v = 1) : IsHaarProd pa pf K (fun w => c w • μa w) (fun v => d v • μf v) ((((∏ w, c w) * ∏ v ∈ T, d v) • μ)

theorem **IsHaarProd.unique** (hK : ∀ v, IsOpen (K v : Set (Yf v))) (hK' : ∀ v, IsCompact (K v : Set (Yf v))) (h : IsHaarProd pa pf K μa μf μ) (h' : IsHaarProd pa pf K μa μf μ') : μ = μ'

theorem **IsRestrGroup.exists_factor** (h : IsRestrGroup pa pf K) (μa : ∀ w, Measure (Ya w)) (μf : ∀ v, Measure (Yf v)) [∀ w, (μa w).IsHaarMeasure] [∀ v, (μf v).IsHaarMeasure] (ν : Measure X) [ν.IsHaarMeasure] (S : Finset V) : ∃ c : ℝ≥0∞, ∀ (A : ∀ w, Set (Ya w)) (B : ∀ v, Set (Yf v)), (∀ w, MeasurableSet (A w)) → (∀ v ∈ S, MeasurableSet (B v)) → ν (cyl (fun w => pa w) (fun v => pf v) (fun v => (K v : Set (Yf v))) S A B) = c * (((∏ w, μa w (A w)) * ∏ v ∈ S, μf v (B v))

theorem **IsRestrGroup.existsUnique_isHaarProd** (h : IsRestrGroup pa pf K) (μa : ∀ w, Measure (Ya w)) (μf : ∀ v, Measure (Yf v)) [ha : ∀ w, (μa w).IsHaarMeasure] [hf : ∀ v, (μf v).IsHaarMeasure] (h1 : ∀ᶠ v in cofinite, μf v (K v) = 1) : ∃! μ : Measure X, IsHaarProd pa pf K μa μf μ

theorem **IsHaarProd.isMulRightInvariant** (h : IsHaarProd pa pf K μa μf μ) (hX : IsRestrGroup pa pf K) [∀ w, (μa w).IsMulRightInvariant] [∀ v, (μf v).IsMulRightInvariant] : μ.IsMulRightInvariant

theorem **IsHaarProd.prod_away** {S : Finset V} {pa' : ∀ w, X' →* Ya w} {pf' : ∀ v : {v // v ∉ S}, X' →* Yf v} {ν : Measure X'} (h : IsHaarProd pa' pf' (fun v => K v) μa (fun v => μf v) ν) [hf : ∀ v, (μf v).IsHaarMeasure] (hK : ∀ v, IsOpen (K v : Set (Yf v))) (e : ((∀ v : S, Yf v) × X') ≃ₜ* X) (ea : ∀ w p, pa w (e p) = pa' w p.2) (eS : ∀ (v : S) p, pf v (e p) = p.1 v) (ef : ∀ (v : {v // v ∉ S}) p, pf v (e p) = pf' v p.2) : IsHaarProd pa pf K μa μf (((Measure.pi fun v : S => μf v).prod ν).map e)

theorem **IsHaarProdFin.prod_arch** {pf' : ∀ v, X' →* Yf v} {ν : Measure X'} (h : IsHaarProdFin pf' K μf ν) [ha : ∀ w, (μa w).IsHaarMeasure] (e : ((∀ w, Ya w) × X') ≃ₜ* X) (ea : ∀ w p, pa w (e p) = p.1 w) (ef : ∀ v p, pf v (e p) = pf' v p.2) : IsHaarProd pa pf K μa μf (((Measure.pi μa).prod ν).map e)

theorem **IsHaarProd.map** {pa' : ∀ w, X' →* Ya w} {pf' : ∀ v, X' →* Yf v} {ν : Measure X'} (h : IsHaarProd pa' pf' K μa μf ν) (e : X' ≃ₜ* X) (ea : ∀ w x, pa w (e x) = pa' w x) (ef : ∀ v x, pf v (e x) = pf' v x) : IsHaarProd pa pf K μa μf (ν.map e)

@[simp] theorem **normHaar_self** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] (K : Subgroup G) (ho : IsOpen (K : Set G)) (hc : IsCompact (K : Set G)) : normHaar K ho hc K = 1

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsFlatSecRankZero

theorem **fin_mul** {ρ : HeightOneSpectrum (𝒪 K) → ℝ} (h : (Function.support ρ).Finite) (g : HeightOneSpectrum (𝒪 K) → ℝ) : (Function.support fun la => ρ la * g la).Finite

theorem **lineDeg_mul** {r r' : (K →+* ℂ) → ℝ} {ρ ρ' : HeightOneSpectrum (𝒪 K) → ℝ} (hr : ∀ σ, 0 < r σ) (hr' : ∀ σ, 0 < r' σ) (h : (Function.support ρ).Finite) (h' : (Function.support ρ').Finite) : lineDeg (r * r') (ρ + ρ') = lineDeg r ρ + lineDeg r' ρ'

theorem **lineDeg_inv** (r : (K →+* ℂ) → ℝ) (ρ : HeightOneSpectrum (𝒪 K) → ℝ) : lineDeg r⁻¹ (-ρ) = -lineDeg r ρ

theorem **lineDeg_one** : lineDeg (K

theorem **finite_vAdd** {a : K} (ha : a ≠ 0) : (Function.support fun la : HeightOneSpectrum (𝒪 K) => (vAdd la a : ℝ)).Finite

theorem **lineDeg_num** {a : K} (ha : a ≠ 0) : lineDeg (fun σ : K →+* ℂ => ‖σ a‖) (fun la => (vAdd la a : ℝ)) = 0

theorem **sum_vAdd_nat** {n : ℕ} (hn : n ≠ 0) : ∑ᶠ la, (vAdd la (n : K) : ℝ) * lw K la = Real.log n

theorem **vla_span** {x : 𝒪 K} (hx : x ≠ 0) (la : HeightOneSpectrum (𝒪 K)) : (QAt.vla la (Ideal.span {x}) : ℤ) = vAdd la (x : K)

theorem **vAdd_natCast** {ℓ n : ℕ} (hℓ : ℓ.Prime) (hn : n ≠ 0) {la : HeightOneSpectrum (𝒪 K)} (hm : (ℓ : 𝒪 K) ∈ la.asIdeal) : vAdd la (n : K) = padicValNat ℓ n * vAdd la (ℓ : K)

theorem **finite_ord** (σ : E →+* K) : (Function.support fun la : HeightOneSpectrum (𝒪 K) => (ord_vw la (conjDifferent σ) : ℝ)).Finite

theorem **sum_ord** (σ : E →+* K) : ∑ᶠ la, (ord_vw la (conjDifferent σ) : ℝ) * lw K la = lrd E

theorem **parWt_eq_sum** (w : Psi H → ℤ) : parWt Dw w = ∑ τ, (-w τ) • flatWt Dw τ

theorem **flDefect_vp_of_ne** {ℓ : ℕ} (h : ℓ ≠ H.d) : flDefect_vp Dw ℓ = Dw.q * different_vp E ℓ

theorem **flDefect_vp_d** (h : different_vp E H.d = 1 / 2) : flDefect_vp Dw H.d = -(Dw.q * different_vp E H.d)

theorem **flDefect_vp_eq_zero** {ℓ : ℕ} (h : ℓ ≠ H.d) (h' : ¬ ℓ ∣ (discr E).natAbs) : flDefect_vp Dw ℓ = 0

theorem **flDefect_vw_eq** [IsGalois ℚ E] (τ : Psi H) {ℓ : ℕ} (hℓ : ℓ.Prime) {la : HeightOneSpectrum (𝒪 c.F)} (hm : (ℓ : 𝒪 c.F) ∈ la.asIdeal) : flDefect_vw Dw c τ la = (vAdd la (ℓ : c.F) : ℝ) * (flDefect_vp Dw ℓ : ℝ)

theorem **finite_flDefect_vw** (τ : Psi H) : (Function.support (flDefect_vw Dw c τ)).Finite

theorem **sum_flDefect_vw** (τ : Psi H) : ∑ᶠ la, flDefect_vw Dw c τ la * lw c.F la = Dw.q * (lrd E - Real.log H.d)

theorem **idem_mul_idem** (Z : Comp Xo c) : idem Z * idem Z = idem Z

theorem **idem_mem_secZ** (Z : Comp Xo c) : idem Z ∈ secZ Xo c 0

theorem **evR_idem** {Z : Comp Xo c} {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} (h : LiesOn Z y) (ht : Tested y) : evR Xo c (idem Z) y = 1

theorem **idem_mem_suppOn** (Z : Comp Xo c) : idem Z ∈ suppOn Z

theorem **valOn_subset** (Z : Comp Xo c) (la : HeightOneSpectrum (𝒪 c.F)) : valOn Z la ⊆ valSet Xo c (algCls Xo c) la

theorem **isVal_valOn** {Z : Comp Xo c} {la : HeightOneSpectrum (𝒪 c.F)} {v : Rul Xo c → EReal} (hv : v ∈ valOn Z la) : IsVal la v

theorem **mem_valOn_or** {Z : Comp Xo c} {la : HeightOneSpectrum (𝒪 c.F)} {v : Rul Xo c → EReal} (hv : v ∈ valSet Xo c (algCls Xo c) la) : v ∈ valOn Z la ∨ ∀ f ∈ suppOn Z, v f = ⊤

theorem **mem_supLat_iff** {Z : Comp Xo c} {C : Submodule c.F (Rul Xo c)} {f : C} (hf : (f : Rul Xo c) ∈ suppOn Z) : f ∈ supLat C ↔ ∀ la, ∀ v ∈ valOn Z la, 0 ≤ v (f : Rul Xo c)

theorem **val_idem** {Z : Comp Xo c} {la : HeightOneSpectrum (𝒪 c.F)} {v : Rul Xo c → EReal} (hv : v ∈ valOn Z la) : v (idem Z) = 0

theorem **val_rat** {Z : Comp Xo c} {la : HeightOneSpectrum (𝒪 c.F)} {v : Rul Xo c → EReal} (hv : v ∈ valOn Z la) {f : Rul Xo c} {a : ℝ} (h : v f = a) : ∃ b : ℚ, a = b

theorem **det_gram** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (y : Matrix (Fin r) (Fin r) ℂ) : ‖(gram τ G y).det‖ = ‖τ G.det‖ * ‖y.det‖ ^ 2

theorem **pnorm_horizGen** (X : Data E r) (τ : X.Ψ) {y : Frame X.Ψ r} (hy : y ∈ X.dom) : X.pnorm (modelFlat r τ) (horizGen τ) y = Real.sqrt ‖τ.1 X.Gm.det‖

theorem **modelWt_flatWt** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (hσ : GoodEmb c σ) (hU : IsUnf U) (τ : Psi H) (k : U.Comp) (hτ : relab c σ τ.1 ∈ (U.X k).Ψ) : modelWt U (wtOf Dw (flatWt Dw τ) 0) k = modelFlat r ⟨_, hτ⟩

theorem **IsHorizontal.homN_const** {σ : c.F →+* ℂ} {U : Unf Xo c σ r} (hσ : GoodEmb c σ) (hU : IsUnf U) (x : ℝ) {Z : Comp Xo c} {τ : Psi H} {f : Rul Xo c} (h : IsHorizontal U Z τ f) (k : U.Comp) : ∃ b : ℝ, ∀ y, (⟨k, y⟩ : Σ k, Frame (U.X k).Ψ r) ∈ framesOn U Z → homN U x (wtOf Dw (flatWt Dw τ) 0) f ⟨k, y⟩ = b

theorem **homN_zero** (f : Rul Xo c) (z : Σ k, Frame (U.X k).Ψ r) : homN U x 0 f z = ‖rdU U f z.1 z.2‖

theorem **homN_smul** (G : WTZ E) (a : c.F) (f : Rul Xo c) (z : Σ k, Frame (U.X k).Ψ r) : homN U x G (a • f) z = ‖σ a‖ * homN U x G f z

theorem **tested_of_homN_ne** {G : WTZ E} {f : Rul Xo c} {z : Σ k, Frame (U.X k).Ψ r} (hz : z.2 ∈ (U.X z.1).dom) (h : homN U x G f z ≠ 0) : Tested (U.un z.1 ⟨z.2, hz⟩)

theorem **homN_idem** {Z : Comp Xo c} {z : Σ k, Frame (U.X k).Ψ r} (hz : z.2 ∈ (U.X z.1).dom) (h : LiesOn Z (U.un z.1 ⟨z.2, hz⟩)) (ht : Tested (U.un z.1 ⟨z.2, hz⟩)) : homN U x 0 (idem Z) z = 1

theorem **image_eq_singleton_iff** {α β : Type*} {f : α → β} {s : Set α} {a : β} : f '' s = {a} ↔ s.Nonempty ∧ ∀ z ∈ s, f z = a

theorem **smul_mul_inv_smul** {K R : Type} [Field K] [CommRing R] [Algebra K R] {a : K} (ha : a ≠ 0) (f g : R) : a • f * a⁻¹ • g = f * g

theorem **wtOf_add'** (la la' : Weight Dw) : wtOf Dw (la + la') 0 = wtOf Dw la 0 + wtOf Dw la' 0

theorem **exOf_add'** (la la' : Weight Dw) : exOf Dw (la + la') 0 = exOf Dw la 0 + exOf Dw la' 0

theorem **exOf_zero** : exOf Dw 0 0 = 0

theorem **exOf_neg** (la : Weight Dw) : exOf Dw (-la) 0 = -exOf Dw la 0

theorem **sig_mem_suppOn** : s.sig ∈ suppOn Z

theorem **inv_mem_suppOn** : s.inv ∈ suppOn Z

theorem **evR_mul_inv** {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} (h : LiesOn Z y) (ht : Tested y) : evR Xo c s.sig y * evR Xo c s.inv y = 1

theorem **homN_eq** {σ : c.F →+* ℂ} {z} (hz : z ∈ framesOn (U (goodOf c σ)) Z) : homN (U (goodOf c σ)) x (wtOf Dw la 0) s.sig z = s.cInf σ

theorem **val_eq** {w} {v : Rul Xo c → EReal} (hv : v ∈ valOn Z w) : v s.sig = ((δ w + s.rho_vw w : ℝ) : EReal)

theorem **hasPts** : HasPts U Z

theorem **sig_ne_zero** : s.sig ≠ 0

theorem **homN_mul_sig** {G : WTZ E} (hG : Even G.2) (f : Rul Xo c) {σ : c.F →+* ℂ} {z} (hz : z ∈ framesOn (U (goodOf c σ)) Z) : homN (U (goodOf c σ)) x (G + wtOf Dw la 0) (f * s.sig) z = homN (U (goodOf c σ)) x G f z * s.cInf σ

theorem **val_mul_sig** (f : Rul Xo c) {w} {v : Rul Xo c → EReal} (hv : v ∈ valOn Z w) : v (f * s.sig) = v f + ((δ w + s.rho_vw w : ℝ) : EReal)

theorem **exists_rat** (w : HeightOneSpectrum (𝒪 c.F)) : ∃ b : ℚ, δ w + s.rho_vw w = b

theorem **sum_log_eq** : ((∑ᶠ w, Real.log (s.cFin w)) + ∑ σ, Real.log (s.cInf σ)) / (finrank ℚ c.F : ℝ) = -s.deg

theorem **ext** {s t : ConstSec U x Z la δ} (h : s.sig = t.sig) : s = t

theorem **deg_one** (h : HasPts U Z) : (one x h).deg = 0

theorem **deg_smul** (a : c.F) (ha : a ≠ 0) (s : ConstSec U x Z la δ) : (s.smul a ha).deg = s.deg

theorem **deg_mul** (s : ConstSec U x Z la δ) (t : ConstSec U x Z la' δ') : (s.mul t).deg = s.deg + t.deg

theorem **deg_symm** (s : ConstSec U x Z la δ) : s.symm.deg = -s.deg

theorem **deg_relat** (s : ConstSec U x Z la δ) (δ' : HeightOneSpectrum (𝒪 c.F) → ℝ) (h : (Function.support (δ - δ')).Finite) : (s.relat δ' h).deg = s.deg + ∑ᶠ w, (δ w - δ' w) * lw c.F w

theorem **exists_prod** (h : HasPts U Z) (T : Finset ι) (s : ∀ i, ConstSec U x Z (lb i) (ε i)) : ∃ t : ConstSec U x Z (∑ i ∈ T, lb i) (∑ i ∈ T, ε i), t.sig = idem Z * ∏ i ∈ T, (s i).sig ∧ t.inv = idem Z * ∏ i ∈ T, (s i).inv ∧ t.cInf = ∏ i ∈ T, (s i).cInf ∧ t.rho_vw = ∑ i ∈ T, (s i).rho_vw ∧ t.deg = ∑ i ∈ T, (s i).deg

theorem **prod_spec** (h : HasPts U Z) (T : Finset ι) (s : ∀ i, ConstSec U x Z (lb i) (ε i)) : (prod h T s).sig = idem Z * ∏ i ∈ T, (s i).sig ∧ (prod h T s).inv = idem Z * ∏ i ∈ T, (s i).inv ∧ (prod h T s).cInf = ∏ i ∈ T, (s i).cInf ∧ (prod h T s).rho_vw = ∑ i ∈ T, (s i).rho_vw ∧ (prod h T s).deg = ∑ i ∈ T, (s i).deg

theorem **sum_range_toNat** {A : Type} [AddCommGroup A] (a : A) (n : ℤ) : ∑ _i ∈ Finset.range n.toNat, a + ∑ _i ∈ Finset.range (-n).toNat, -a = n • a

theorem **zpow_spec** (s : ConstSec U x Z la δ) (n : ℤ) : (s.zpow n).sig = idem Z * (s.sig ^ n.toNat * s.inv ^ (-n).toNat) ∧ (∀ σ, (s.zpow n).cInf σ = s.cInf σ ^ n) ∧ (s.zpow n).rho_vw = n • s.rho_vw ∧ (s.zpow n).deg = n * s.deg

theorem **deg_eq_toFrame** {τ : Psi H} (s : FlatSec U x Z τ) : s.deg = s.toFrame.deg + Dw.q * (Real.log H.d - lrd E)

theorem **rho_rat** {τ : Psi H} (s : FlatSec U x Z τ) (w : HeightOneSpectrum (𝒪 c.F)) : ∃ b : ℚ, s.rho_vw w = b

theorem **shift_spec** (h : HasPts U Z) (s : ∀ τ, FlatSec U x Z τ) (w : Psi H → ℤ) : (shift h s w).sig = idem Z * ∏ τ, ((s τ).zpow (-w τ)).sig ∧ (∀ σ, (shift h s w).cInf σ = ∏ τ, (s τ).cInf σ ^ (-w τ)) ∧ (∀ v, (shift h s w).rho_vw v = -∑ τ, w τ * (s τ).rho_vw v) ∧ (shift h s w).deg = ∑ τ, w τ * -(s τ).deg

theorem **FlatExistsAt.degFlat_eq** {a : ℝ} (h : FlatExistsAt U x a) (Z : Comp Xo c) (τ : Psi H) : degFlat U x Z τ = -a

theorem **FlatExistsAt.c1_eq** {a : ℝ} (h : FlatExistsAt U x a) (hne : CompNe Xo c) (τ : Psi H) : c1 U x τ = a

theorem **FlatExists.deg_eq** (h : FlatExists U x) {Z : Comp Xo c} {τ : Psi H} (s : FlatSec U x Z τ) (τ' : Psi H) : s.deg = -c1 U x τ'

theorem **FlatExists.nonempty** (h : FlatExists U x) (Z : Comp Xo c) (τ : Psi H) : Nonempty (FlatSec U x Z τ)

theorem **FlatExists.sum_log_eq** (h : FlatExists U x) {Z : Comp Xo c} {τ : Psi H} (s : FlatSec U x Z τ) : ((∑ᶠ w, Real.log (s.cFin w)) + ∑ σ, Real.log (s.cInf σ)) / (finrank ℚ c.F : ℝ) = c1 U x τ

theorem **FlatExists.deg_shift** (h : FlatExists U x) {Z : Comp Xo c} (hp : HasPts U Z) (s : ∀ τ, FlatSec U x Z τ) (w : Psi H → ℤ) : (FlatSec.shift hp s w).deg = ∑ τ, w τ * c1 U x τ

theorem **FlatDegree.c1_eq** {logOm : ℝ} (h : FlatDegree U x logOm) (hne : CompNe Xo c) (τ : Psi H) : c1 U x τ = -(((Dw.p : ℝ) - Dw.q) / 2 * logOm) - Dw.q * (Real.log H.d - lrd E)

theorem **FlatDegree.flatExists** {logOm : ℝ} (h : FlatDegree U x logOm) : FlatExists U x

theorem **exists_flatSec_rank_zero** (hp : Dw.p = 0) (hq : Dw.q = 0) {Z : Comp Xo c} (h : HasPts U Z) (τ : Psi H) {a : c.F} (ha : a ≠ 0) : ∃ s : FlatSec U x Z τ, s.sig = a • idem Z ∧ (∀ σ, s.cInf σ = ‖σ a‖) ∧ (∀ w, s.rho_vw w = vAdd w a) ∧ s.deg = 0

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsGLSubmatrix

theorem **mem_hdual** {S K V : Type} [CommRing S] [Field K] [StarRing K] [Algebra S K] [AddCommGroup V] [Module K V] [Module S V] [IsScalarTower S K V] {B : V →ₗ⋆[K] V →ₗ[K] K} {L : Submodule S V} {x : V} : x ∈ Formalism.HermitianLatticesTheory.dual B L ↔ ∀ y ∈ L, B y x ∈ (1 : Submodule S K)

theorem **orthSum_apply** (V W : HSpace E) (x y : V.V × W.V) : Formalism.E.orthSum V.hT W.hT x y = V.hT x.1 y.1 + W.hT x.2 y.2

theorem **gram_hneg** (V : HSpace E) : (hneg V).gram = -V.gram

theorem **frame_hsum_inl** (V W : HSpace E) (i : Fin V.rank) : (hsum V W).frame (finSumFinEquiv (.inl i)) = (V.frame i, 0)

theorem **frame_hsum_inr** (V W : HSpace E) (i : Fin W.rank) : (hsum V W).frame (finSumFinEquiv (.inr i)) = (0, W.frame i)

theorem **gram_hsum** (V W : HSpace E) : (hsum V W).gram = blkMat finSumFinEquiv V.gram W.gram

theorem **gram_hdbl** (V : HSpace E) : (hdbl V).gram = dblGram finSumFinEquiv V.gram

theorem **sum_pairVec_frame** (V W : HSpace E) (x : Fin V.rank → E) (y : Fin W.rank → E) : ∑ i, pairVec finSumFinEquiv x y i • (hsum V W).frame i = (∑ i, x i • V.frame i, ∑ i, y i • W.frame i)

theorem **frameLat_hsum** (V W : HSpace E) : frameLat (hsum V W) = sumLat finSumFinEquiv (frameLat V) (frameLat W)

theorem **isLattice_frameLat** (V : HSpace E) : Submodule.IsLattice E (frameLat V)

theorem **isIsomEmb_inl** (V W : HSpace E) : HSpace.IsIsomEmb V (hsum V W) (LinearMap.inl E V.V W.V)

theorem **isIsomEmb_inr** (V W : HSpace E) : HSpace.IsIsomEmb W (hsum V W) (LinearMap.inr E V.V W.V)

theorem **exists_GL_submatrix** {A m n : Type} [CommRing A] [Fintype m] [DecidableEq m] [Fintype n] [DecidableEq n] (P : GL m A) (e : n ≃ m) : ∃ P' : GL n A, P'.1 = P.1.submatrix e e

private lemma **cT_submatrix_mul_mul** {m n : Type} [Fintype m] [Fintype n] (σ : ℂ →+* ℂ) (e : n ≃ m) (P J Q : Matrix m m ℂ) : cT σ (P.submatrix e e) * J.submatrix e e * Q.submatrix e e = (cT σ P * J * Q).submatrix e e

theorem **hasSignature_iff** {n : Type} [Fintype n] [DecidableEq n] {J : Matrix n n ℂ} {p q : ℕ} : HasSignature J p q ↔ ∃ (e : Fin p ⊕ Fin q ≃ n) (P : GL n ℂ), cT (starRingEnd ℂ) P.1 * J * P.1 = blkMat e 1 (-1)

private lemma **blkMat_diagonal_eq_diagonal** {m n ι : Type} [DecidableEq m] [DecidableEq n] [DecidableEq ι] (e : m ⊕ n ≃ ι) (d₁ : m → ℂ) (d₂ : n → ℂ) : blkMat e (Matrix.diagonal d₁) (Matrix.diagonal d₂) = Matrix.diagonal (fun i => Sum.elim d₁ d₂ (e.symm i))

private lemma **blkMat_one_neg_one_eq_diagonal** {m n ι : Type} [DecidableEq m] [DecidableEq n] [DecidableEq ι] (e : m ⊕ n ≃ ι) : blkMat e (1 : Matrix m m ℂ) (-1) = Matrix.diagonal (fun i => Sum.elim (fun _ => (1 : ℂ)) (fun _ => -1) (e.symm i))

theorem **sigOK_hneg** {V : HSpace E} (h : V.SigOK) : (hneg V).SigOK

theorem **sigOK_hsum** {V W : HSpace E} (h : V.SigOK) (h' : W.SigOK) : (hsum V W).SigOK

theorem **sigOK_hdbl** {V : HSpace E} (h : V.SigOK) : (hdbl V).SigOK

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsHaarZetaValueAt

theorem **exists_cosetSum_eq_one** {G : Type} [Group G] (P Γ : Subgroup G) : ∃ ψ : G → ℂ, LeftInv (P ⊓ Γ) ψ ∧ ∀ g, CosetConv P Γ ψ g ∧ cosetSum P Γ ψ g = 1

theorem **vol_ne_zero** : (((autMeasure V m).real univ : ℝ) : ℂ) ≠ 0

theorem **exists_family** (c : ℂ → ℂ) : ∃ Φ : ℂ → GA D (dblSp V) → ℂ, (∀ s g, eis V (Φ s) g = c s) ∧ IsZetaFamily V 1 m Φ 1 1 ∧ ∀ s, zeta V 1 m Φ 1 1 s = c s * ((autMeasure V m).real univ : ℝ) ^ 2

theorem **exists_zeta_eq** (h : ℂ → ℂ) : ∃ Φ, IsZetaFamily V 1 m Φ 1 1 ∧ zeta V 1 m Φ 1 1 = h

theorem **exists_zetaValueAt** (s₀ z : ℂ) : ∃ Φ, ZetaContinues V 1 m Φ 1 1 ∧ ZetaValueAt V 1 m Φ 1 1 s₀ z ∧ ZetaCoeffAt V 1 m Φ 1 1 s₀ (-1) 1 ∧ (∀ c, ¬ ZetaHolValueAt V 1 m Φ 1 1 s₀ c) ∧ ∀ c ≠ z, ¬ ZetaValueAt V 1 m Φ 1 1 s₀ c

theorem **exists_not_zetaValueAt** : ∃ Φ, IsZetaFamily V 1 m Φ 1 1 ∧ ∀ s₀ c, ¬ ZetaValueAt V 1 m Φ 1 1 s₀ c

theorem **exists_haar_zetaValueAt** (s₀ z : ℂ) : ∃ (m : Measure (GA D V)) (_ : m.IsHaarMeasure) (Φ : ℂ → GA D (dblSp V) → ℂ), ZetaValueAt V 1 m Φ 1 1 s₀ z ∧ ∀ c ≠ z, ¬ ZetaValueAt V 1 m Φ 1 1 s₀ c

theorem **not_isEisFamily_one** [Infinite (GQ V)] : ¬ IsEisFamily V fun _ _ => 1

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsIsDeepOfField

theorem **exists_isDeepLevel** (L : SizeData₂ D) (hT : LawsJ T) (hj : InR D j) (h5 : 5 ≤ D.host.splitBound) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) {n : ℕ} (hn : n ≠ 0) : ∃ (l : Level T.Q.H) (Xo : Moduli (T.Q.Dw j) l) (c : Coeff l) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ D.R.r) (dict : ∀ σ, GA D (D.sp.W j) → Σ k, Frame ((U σ).X k).Ψ D.R.r), IsDeepLevel L T j Xo c U dict ∧ IsTypeLevel l ∧ ∃ ζ : c.F, IsPrimitiveRoot ζ n

theorem **IsDeepLevel.isDeep** (h : IsDeepLevel L T j Xo c U dict) (hs : Real.log (Module.finrank ℚ c.F) ≤ (numAt Q j).dP ∧ Real.log (rootDiscr c.F) ≤ (numAt Q j).rP) : IsDeep Q T j Xo c U dict

theorem **exists_isDeep_of_field** (hT : LawsJ T) (hj : InR D j) (hl : l.M = levDeep D L ∧ l.A = 1 ∧ T.Q.l.Dvd l) (le : T.Q.c.F ≤ c.F) (hG : letI

theorem **ringCost_le_o** {K : Type} [Field K] [NumberField K] (hs : Real.log (Module.finrank ℚ K) ≤ (numAt Q j).dP ∧ Real.log (rootDiscr K) ≤ (numAt Q j).rP) : GlobalTheory.IndefiniteRowSideFullHecke.HalfLatticeTypes.ringCost K ≤ (numAt Q j).o

theorem **htPrimes_eq_badPrimes** (hM : l.M = levDeep D L) (hA : l.A = 1) (B : ℝ) : htPrimes l B = ↑(badPrimes D L B)

theorem **coe_restrictOr** {φ : M →ₗ[K] M'} {C : Submodule K M} {C' : Submodule K M'} (h : ∀ x ∈ C, φ x ∈ C') (x : C) : (restrictOr φ C C' x : M') = φ x

theorem **mul_mem_cuspW** {G G' : WTZ E} (hG : Even G.2) (hG' : Even G'.2) {f g : Rul Xo c} (hf : f ∈ cuspW Xo c U G) (hg : g ∈ cuspW Xo c U G') : f * g ∈ cuspW Xo c U (G + G')

theorem **levPh_mem_cuspW** {G G' : WTZ E} (hG : Even G.2) (hG' : Even G'.2) (hK : KeepsCusp U (G + G')) (hK' : KeepsCusp U G') {S : Subgroup (GlobalTheory.IndefiniteRowSideFullHecke.LevelStructureAutomorphisms.LevelAut Dw l)} (χ : S →* (c.F)ˣ) {u f : Rul Xo c} (hu : u ∈ cuspW Xo c U G') (hf : f ∈ cuspW Xo c U G) (h : S) : levPh Xo c S χ u f h ∈ cuspW Xo c U (G + G')

theorem **typeOp_mem_cuspW** {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {r : ℕ} {U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r} {G : WTZ D.host.E} (hK : KeepsCusp U G) (hl : IsTypeLevel l) (fr : TypeFrame Dw l) (e : MonoidAlgebra c.F (LevelGroup D.host)) {f : Rul Xo c} (hf : f ∈ cuspW Xo c U G) : typeOp Xo c hl fr e f ∈ cuspW Xo c U G

theorem **inR_succ** (i : Fin D.R.m) : InR D ((i : ℕ) + 1)

theorem **coe_Ph** (hK : KeepsCusp Y.U (gKap T.Q ((i : ℕ) + 1))) (hK' : KeepsCusp Y.U (gTw T.Q ((i : ℕ) + 1))) (f : Y.V (gKapMinus T.Q ((i : ℕ) + 1))) (h : Y.Hb) : (Y.Ph f h : Rul Y.Xo Y.c) = levTr Y.Xo Y.c Y.Hb (chiHR Y.fr Y.ψ) ((f : Rul Y.Xo Y.c) * pull Y.Xo Y.c h (Y.u₀ : Rul Y.Xo Y.c))

theorem **coe_eGood** (hK : KeepsCusp Y.U (gKap T.Q ((i : ℕ) + 1))) (f : Y.V (gKap T.Q ((i : ℕ) + 1))) : (Y.eGood f : Rul Y.Xo Y.c) = goodOp Y.Xo Y.c Y.typeLevel Y.fr Y.om i f

theorem **eGood_eGood** (f : Y.V (gKap T.Q ((i : ℕ) + 1))) : Y.eGood (Y.eGood f) = Y.eGood f

theorem **good_bad_eq** : Y.NN ⊓ (LinearMap.range Y.eGood).restrictScalars (𝒪 Y.c.F) = latGood Y.typeLevel Y.fr Y.U _ Y.ψ Y.om i ∧ Y.NN ⊓ (LinearMap.ker Y.eGood).restrictScalars (𝒪 Y.c.F) = latBad Y.typeLevel Y.fr Y.U _ Y.ψ Y.om i

theorem **facOf_eq** : facOf Q h Y CI C2 = facWith Q Y.U Y.dict h (inR_succ i) CI C2 (mixVals Y.Xo Y.c (bZero Q ((i : ℕ) + 1))) Y.Hb Y.NN Y.Ph Y.eGood (Nat.card (LevelGroup D.host))

theorem **facOf_q** : (facOf Q h Y CI C2).q' = D.R.m - ((i : ℕ) + 1)

theorem **facOf_q_ne_zero_iff** : (facOf Q h Y CI C2).q' ≠ 0 ↔ (i : ℕ) + 1 < D.R.m

theorem **one_le_facOf_g** : 1 ≤ (facOf Q h Y CI C2).g

theorem **facOf_NN_le** : (facOf Q h Y CI C2).NN ≤ (facOf Q h Y CI C2).Mk

theorem **facOf_good_bad** (hG : HasGram (nrmAt Y.U Y.dict (gKap T.Q ((i : ℕ) + 1)))) (hK : KeepsCusp Y.U (gKap T.Q ((i : ℕ) + 1))) (he : IsIdempotentElem (goodIdemTw Y.om i)) : (facOf Q h Y CI C2).good = latCo (nrmAt Y.U Y.dict (gKap T.Q ((i : ℕ) + 1))) (latGood Y.typeLevel Y.fr Y.U _ Y.ψ Y.om i) ∧ (facOf Q h Y CI C2).bad = latCo (nrmAt Y.U Y.dict (gKap T.Q ((i : ℕ) + 1))) (latBad Y.typeLevel Y.fr Y.U _ Y.ψ Y.om i)

theorem **facOf_o_le** (hs : Real.log (Module.finrank ℚ Y.c.F) ≤ (numAt Q ((i : ℕ) + 1)).dP ∧ Real.log (rootDiscr Y.c.F) ≤ (numAt Q ((i : ℕ) + 1)).rP) (ho : (numAt Q ((i : ℕ) + 1)).o ≤ (numAt Q ((i : ℕ) + 1)).PPpp) : (facOf Q h Y CI C2).o' ≤ (pageOf Q).PPpp

theorem **LiftData.exists_deep** (L : SizeData₂ D) (hT : LawsJ T) (i : Fin D.R.m) (h5 : 5 ≤ D.host.splitBound) (h15 : Nat.Coprime 15 (D.host.g 0 * D.host.g 1)) : ∃ Y : LiftData T i, IsDeepLevel L T ((i : ℕ) + 1) Y.Xo Y.c Y.U Y.dict ∧ IsGenericFamily Y.ψ ∧ IsGeneratorFamily Y.om

theorem **LiftData.exists_good_of_field** (hT : LawsJ T) (i : Fin D.R.m) (hi : (i : ℕ) + 1 = D.R.m) {l : Level T.Q.H} (hl : l.M = levDeep D L ∧ l.A = 1 ∧ T.Q.l.Dvd l) (ht : IsTypeLevel l) (Xo : Moduli (T.Q.Dw ((i : ℕ) + 1)) l) (c : Coeff l) (le : T.Q.c.F ≤ c.F) (hG : letI

theorem **isUnifAt_core** (hT : LawsJ T) {j : ℕ} (hj : InR D j) {σ : T.Q.c.F →+* ℂ} (hσ : GoodEmb T.Q.c σ) {V : HSp D} (hV : T.V j σ = V) : GlobalTheory.IndefiniteRowSideFullHecke.HalfLatticeTypes.IsUnifAt (T.U j σ) V (fun g => T.dict j σ (cast (congrArg (GA D) hV.symm) g)) (T.iota j)

theorem **LiftData.ofCore_unif** (hT : LawsJ T) (i : Fin D.R.m) (ψ om u₀) (σ : T.Q.c.F →+* ℂ) (hσ : GoodEmb T.Q.c σ) : GlobalTheory.IndefiniteRowSideFullHecke.HalfLatticeTypes.IsUnifAt ((LiftData.ofCore hT i ψ om u₀).U σ) (D.sp.W ((i : ℕ) + 1)) ((LiftData.ofCore hT i ψ om u₀).dict σ) (T.iota ((i : ℕ) + 1))

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsOfEmbedding

theorem **IsNTr.isPrimitiveRoot_zt** (ε : Ω →+* ℂ) (l n : ℕ) [NeZero l] : IsPrimitiveRoot (zt ε l n) (l ^ n)

theorem **IsNTr.zt_pow** (ε : Ω →+* ℂ) (l m k : ℕ) [NeZero l] : zt ε l (m + k) ^ l ^ k = zt ε l m

theorem **IsNTr.card_filter** (l n m : ℕ) [hl : Fact l.Prime] (hm : 1 ≤ m) (hmn : m ≤ n) : (Finset.univ.filter (fun a : (ZMod (l ^ n))ˣ => (a : ZMod (l ^ n)).val ≡ 1 [MOD l ^ m])).card = l ^ (n - m)

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsPairMass

theorem **wd_zero** (D : D7) (T : Tor D) {a d : ℕ} (G : VC D a d → ℂ) : Wd D T 0 G = G

theorem **pres_vac** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {a d : ℕ} (e0 : VC D a d → ℂ) : Pres D cc δ 1 e0 e0

theorem **spherical_F_ne** {D : D7} {cc : Ideal (𝒪 D.host.K)} {δ t0 : ℂ} (crv : Crv D cc δ) (pt0 : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t0) {a d : ℕ} (had : a ≤ d) (hd : Even d) (e0 : VC D a d → ℂ) (vac : oddE0 D cc δ t0 a d e0) (W : WeilO D cc δ a d e0) (n : D.what₂.Tau → ℕ) (wt) (hfix) (heig) : (spherical crv pt0 had hd e0 vac W n wt hfix heig).F ≠ 0

theorem **mem_intG** {D : D7} {N : ℕ} (k : GA D N) : k ∈ intG D N ↔ IntG D (k : GL (Fin N) (AE D))

theorem **term_mul_int** {D : D7} {cc : Ideal (𝒪 D.host.K)} {δ t0 : ℂ} {Q : ℕ} (P : Pair D cc δ t0 Q) (hF : P.F ∈ TT D (tor0 D cc δ) P.a P.d) (x k : GA D P.a) (hk : k ∈ intG D P.a) (h : GA D P.d) (T : Tup D P.a P.d P.n) : P.term (x * k) h T = P.term x h T

theorem **level_ne_zero** {D : D7} {cc : Ideal (𝒪 D.host.K)} {δ t0 : ℂ} {Q : ℕ} (P : Pair D cc δ t0 Q) : Q ≠ 0

theorem **hf0** (D : D7) (T : Tor D) (z w : VC D 0 0) : Hf D T z w = 0

theorem **iop0** (D : D7) (T : Tor D) (A B : AddSubgroup (VC D 0 0)) (G : VC D 0 0 → ℂ) : Iop D T A B G = G

theorem **taylor0** (D : D7) (α : Mono D 0 0) (c : ℂ) : taylor D (fun _ => c) α = c

theorem **normSq0** (D : D7) (T : Tor D) (A : AddSubgroup (VC D 0 0)) : normSq D T A (fun _ => 1) = 1

theorem **vac0** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ t0 : ℂ) : oddE0 D cc δ t0 0 0 (fun _ => 1)

theorem **eqv0** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) (e0 : VC D 0 0 → ℂ) : oddEq D cc δ 0 0 e0

theorem **gam0** (D : D7) (J : Subgroup (GA D 0)) (g : GA D 0) : gam D J g = 1

theorem **normSumPairRankZeroOfCurve_coord_norm** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ t0 : ℂ) (crv : Crv D cc δ) (pt0 : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t0) : (pair0 D cc δ t0 crv pt0).norm = 1

theorem **exists_twoTorsionPointUnkilled_of_crv** {D : D7} {cc : Ideal (𝒪 D.host.K)} {δ : ℂ} (h : Crv D cc δ) : ∃ t0, GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t0

theorem **vac_mem_TT** {D : D7} {cc : Ideal (𝒪 D.host.K)} {δ : ℂ} (hc : Crv D cc δ) {a d : ℕ} {e0 : VC D a d → ℂ} (he : e0 ∈ Th D (tor0 D cc δ) (stdL D a) (stdL D d)) : e0 ∈ TT D (tor0 D cc δ) a d

theorem **exists_pair_of_unramified** {D : D7} {cc : Ideal (𝒪 D.host.K)} {δ t0 : ℂ} (crv : Crv D cc δ) (pt0 : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t0) {a d : ℕ} (had : a ≤ d) (hd : Even d) (e0 : VC D a d → ℂ) (vac : oddE0 D cc δ t0 a d e0) (W : WeilO D cc δ a d e0) (n : D.what₂.Tau → ℕ) (wt : ∀ τ, 2 * (n τ : ℤ) = W.ep' * W.chV'.n τ − (d : ℤ)) (hu : ∀ u, IntU D u → W.chV'.f u = 1) : ∃ P : Pair D cc δ t0 1, P.a = a ∧ P.d = d ∧ P.n = n ∧ P.F = P.e0 ∧ P.F ≠ 0

theorem **eq_zero_of_zeroAt** (D : D7) {z : AE D} (h : ∀ q : ℕ, q.Prime → ZeroAt D q z) : z = 0

theorem **hecke_eq_one** {D : D7} {N : ℕ} (χ : HeckeU D N) (x : (AE D)ˣ) (hx : x = sgU D x) (hi : ∀ q : ℕ, q.Prime → InertQ D q → ZeroAt D q ((x : AE D) − 1)) : χ.f x = 1

theorem **hfD** (D : D7) (T : Tor D) {d : ℕ} (z w : VC D 0 d) : Hf D T z w = 0

theorem **iopD** (D : D7) (T : Tor D) {d : ℕ} (A B : AddSubgroup (VC D 0 d)) (G : VC D 0 d → ℂ) : Iop D T A B G = G

theorem **taylorD** (D : D7) {d : ℕ} (α : Mono D 0 d) (c : ℂ) : taylor D (fun _ => c) α = c

theorem **normSqD** (D : D7) (T : Tor D) {d : ℕ} (A : AddSubgroup (VC D 0 d)) : normSq D T A (fun _ => 1) = 1

theorem **vacD** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ t0 : ℂ) (d : ℕ) : oddE0 D cc δ t0 0 d (fun _ => 1)

theorem **exists_pair_mass** (D : D7) (d m : ℕ) (hd : Even d) : ∃ (δ t0 : ℂ) (P : Pair D ⊤ δ t0 1), P.a = 0 ∧ P.d = d ∧ (∀ τ, P.n τ = m) ∧ (∀ h T, P.coord h T = 1) ∧ P.norm = ∑ᶠ h : Cls D d (intG D d), ((gam D (intG D d) h.out : ℝ))⁻¹ ∧ 0 < P.norm

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsRootOfIsAlgClosed

@[local simp] theorem **coeff_cm** (c : ℚ → R) (φ : HahnSeries ℚ R) (r : ℚ) : (cm c φ).coeff r = c r * φ.coeff r

theorem **cm_add** (c : ℚ → R) (φ ψ : HahnSeries ℚ R) : cm c (φ + ψ) = cm c φ + cm c ψ

theorem **cm_smul** (c : ℚ → R) (t : R) (φ : HahnSeries ℚ R) : cm c (t • φ) = t • cm c φ

@[local simp] theorem **coeff_qD** [Algebra ℚ R] (φ : HahnSeries ℚ R) (r : ℚ) : (qD φ).coeff r = algebraMap ℚ R r * φ.coeff r

theorem **coeff_scale_mul** {a : ℚ} (ha : 0 < a) (φ : HahnSeries ℚ R) (r : ℚ) : (scale a φ).coeff (a * r) = φ.coeff r

theorem **coeff_scale** {a : ℚ} (ha : 0 < a) (φ : HahnSeries ℚ R) (r : ℚ) : (scale a φ).coeff r = φ.coeff (r / a)

theorem **scale_add** (a : ℚ) (φ ψ : HahnSeries ℚ R) : scale a (φ + ψ) = scale a φ + scale a ψ

theorem **scale_smul** (a : ℚ) (t : R) (φ : HahnSeries ℚ R) : scale a (t • φ) = t • scale a φ

theorem **scale_mul** (a : ℚ) (φ ψ : HahnSeries ℚ R) : scale a (φ * ψ) = scale a φ * scale a ψ

@[local simp] theorem **scale_one** (φ : HahnSeries ℚ R) : scale 1 φ = φ

theorem **coeff_latPart** (N : ℕ) (φ : HahnSeries ℚ R) (r : ℚ) : (latPart N φ).coeff r = if ((N : ℚ) * r).den = 1 then φ.coeff r else 0

theorem **coeff_uqN** (N : ℕ) {q : ℕ} (hq : 0 < q) (φ : HahnSeries ℚ R) (r : ℚ) : (uqN N q φ).coeff r = if ((N : ℚ) * r).den = 1 then φ.coeff (q * r) else 0

theorem **coeff_uq** {q : ℕ} (hq : 0 < q) (φ : HahnSeries ℚ R) (r : ℚ) : (uq q φ).coeff r = if r.den = 1 then φ.coeff (q * r) else 0

theorem **coeff_vq** {q : ℕ} (hq : 0 < q) (φ : HahnSeries ℚ R) (r : ℚ) : (vq q φ).coeff r = φ.coeff (r / q)

theorem **coeff_sub_upper** {a d : ℚ} (h : 0 < a / d) (b : ℚ) (φ : HahnSeries ℚ Ω) (r : ℚ) : (sub ε a b d φ).coeff (r * a / d) = eOm ε (r * b / d) * φ.coeff r

theorem **sub_map_add** (a b d : ℚ) (φ ψ : HahnSeries ℚ Ω) : sub ε a b d (φ + ψ) = sub ε a b d φ + sub ε a b d ψ

theorem **sub_map_smul** (a b d : ℚ) (t : Ω) (φ : HahnSeries ℚ Ω) : sub ε a b d (t • φ) = t • sub ε a b d φ

theorem **ent_div_pos** (g : GQ) (hg : ent g 1 0 = 0) : 0 < ent g 0 0 / ent g 1 1

theorem **exp_sl_upper** (X : Tower Ω ε) (g : GQ) (hg : ent g 1 0 = 0) (f : X.K) : X.exp (X.sl g f) = sub ε (ent g 0 0) (ent g 0 1) (ent g 1 1) (X.exp f)

theorem **eOm_eq_of** {x : ℚ} {c : Ω} (h : ε c = Complex.exp (2 * Real.pi * Complex.I * (x : ℂ))) : eOm ε x = c

theorem **eOm_intCast_div** {q : ℕ} {ζ : Ω} (hζ : ε ζ = Complex.exp (2 * Real.pi * Complex.I / q)) (n : ℤ) : eOm ε ((n : ℚ) / q) = ζ ^ n

theorem **exists_root_of_isAlgClosed** [IsAlgClosed Ω] (ε : Ω →+* ℂ) {q : ℕ} (hq : 0 < q) : ∃ ζ : Ω, ε ζ = Complex.exp (2 * Real.pi * Complex.I / q)

theorem **coeff_hq** {q : ℕ} (hq0 : 0 < q) (φ : HahnSeries ℚ Ω) (r : ℚ) : (hq ε q φ).coeff r = (q : Ω)⁻¹ * (∑ j ∈ Finset.range q, eOm ε (r * j)) * φ.coeff (q * r)

theorem **hq_eq_uq** {q : ℕ} (hq0 : 0 < q) (hζ : ∃ ζ : Ω, ε ζ = Complex.exp (2 * Real.pi * Complex.I / q)) {φ : HahnSeries ℚ Ω} (hφ : IsLat 1 φ) : hq ε q φ = uq q φ

theorem **hq_eq_uq_of_isAlgClosed** [IsAlgClosed Ω] {q : ℕ} (hq0 : 0 < q) {φ : HahnSeries ℚ Ω} (hφ : IsLat 1 φ) : hq ε q φ = uq q φ

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsUniqueIsProdMeasureFinAway

theorem **mem_cyl** {S : Finset κ} {A : ∀ i, Set (H i)} {B : ∀ k, Set (P k)} {g : G} : g ∈ cyl p q K S A B ↔ (∀ i, p i g ∈ A i) ∧ (∀ k ∈ S, q k g ∈ B k) ∧ ∀ k ∉ S, q k g ∈ K k

theorem **cyl_eq_of_subset** {S S' : Finset κ} (h : S ⊆ S') (A : ∀ i, Set (H i)) (B : ∀ k, Set (P k)) : cyl p q K S A B = cyl p q K S' A fun k => if k ∈ S then B k else K k

theorem **prod_ite_eq** {S S' : Finset κ} (h : S ⊆ S') (hS : ∀ k ∉ S, μf k (K k) = 1) (B : ∀ k, Set (P k)) : ∏ k ∈ S', μf k (if k ∈ S then B k else K k) = ∏ k ∈ S, μf k (B k)

theorem **exists_finset_vol_one** (h : ∀ᶠ k in cofinite, μf k (K k) = 1) : ∃ S : Finset κ, ∀ k ∉ S, μf k (K k) = 1

theorem **IsRestrProd.congr** (h : IsRestrProd p q K μi μf μ) {K' : ∀ k, Subgroup (P k)} (hK' : ∀ k, MeasurableSet (K' k : Set (P k))) (e : ∀ᶠ k in cofinite, K k = K' k) : IsRestrProd p q K' μi μf μ

theorem **IsRestrProd.smul_inf** [DecidableEq ι] (h : IsRestrProd p q K μi μf μ) (i : ι) {c : ℝ≥0} (hc : c ≠ 0) : IsRestrProd p q K (Function.update μi i (c • μi i)) μf (c • μ)

theorem **IsRestrProd.smul_fin** [DecidableEq κ] (h : IsRestrProd p q K μi μf μ) (hK : ∀ k, MeasurableSet (K k : Set (P k))) (k : κ) {c : ℝ≥0} (hc : c ≠ 0) : IsRestrProd p q K μi (Function.update μf k (c • μf k)) (c • μ)

theorem **IsRestrGroup.exists_const** (D : IsRestrGroup p q K) (μi : ∀ i, Measure (H i)) (μf : ∀ k, Measure (P k)) [∀ i, (μi i).IsHaarMeasure] [∀ k, (μf k).IsHaarMeasure] (ν : Measure G) [ν.IsHaarMeasure] (S : Finset κ) : ∃ c : ℝ≥0, c ≠ 0 ∧ ∀ A B, (∀ i, MeasurableSet (A i)) → (∀ k ∈ S, MeasurableSet (B k)) → ν (cyl p q K S A B) = c * ((∏ i, μi i (A i)) * ∏ k ∈ S, μf k (B k))

theorem **exists_box_pos** (μi : ∀ i, Measure (H i)) (μf : ∀ k, Measure (P k)) [∀ i, (μi i).IsHaarMeasure] [∀ k, (μf k).IsHaarMeasure] (S : Finset κ) : ∃ (A : ∀ i, Set (H i)) (B : ∀ k, Set (P k)), (∀ i, MeasurableSet (A i)) ∧ (∀ k, MeasurableSet (B k)) ∧ (∏ i, μi i (A i)) * ∏ k ∈ S, μf k (B k) ≠ 0 ∧ (∏ i, μi i (A i)) * ∏ k ∈ S, μf k (B k) ≠ ∞

theorem **IsRestrGroup.exists_smul** (D : IsRestrGroup p q K) (hi : ∀ i, (μi i).IsHaarMeasure) (hf : ∀ k, (μf k).IsHaarMeasure) (h1 : ∀ᶠ k in cofinite, μf k (K k) = 1) (ν : Measure G) [ν.IsHaarMeasure] : ∃ c : ℝ≥0, c ≠ 0 ∧ IsRestrProd p q K μi μf (c • ν)

theorem **IsRestrProd.unique** {μ' : Measure G} (h : IsRestrProd p q K μi μf μ) (h' : IsRestrProd p q K μi μf μ') : μ = μ'

theorem **IsRestrGroup.existsUnique** (D : IsRestrGroup p q K) (hi : ∀ i, (μi i).IsHaarMeasure) (hf : ∀ k, (μf k).IsHaarMeasure) (h1 : ∀ᶠ k in cofinite, μf k (K k) = 1) : ∃! μ, IsRestrProd p q K μi μf μ

theorem **isRestrProd_empty_iff** {G κ : Type} {P : κ → Type} [Group G] [∀ k, Group (P k)] [TopologicalSpace G] [∀ k, TopologicalSpace (P k)] [MeasurableSpace G] [∀ k, MeasurableSpace (P k)] {q : ∀ k, G →* P k} {K : ∀ k, Subgroup (P k)} {μf : ∀ k, Measure (P k)} {μ : Measure G} : IsRestrProd (H

theorem **isProdMeasureFin_iff** {μf : ∀ v, Measure (GLoc σ J v)} {μ : Measure (GFin σ J)} : IsProdMeasureFin σ J L μf μ ↔ (∀ v, (μf v).IsHaarMeasure) ∧ (∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) ∧ μ.IsHaarMeasure ∧ ∀ S : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∉ S, μf v (stabLoc σ J L v) = 1) → ∀ B : ∀ v, Set (GLoc σ J v), (∀ v ∈ S, MeasurableSet (B v)) → μ {g | (∀ v ∈ S, projFinLoc σ J v g ∈ B v) ∧ ∀ v ∉ S, projFinLoc σ J v g ∈ stabLoc σ J L v} = ∏ v ∈ S, μf v (B v)

theorem **comp_glue** {S : Finset (HeightOneSpectrum (𝒪 F))} (a : ∀ w, GArch σ J w) (b : ∀ v : S, GLoc σ J v) : (∀ w, projArch σ J w (ofInf σ J (embArchPi σ J a) * embAt σ J S b) = a w) ∧ (∀ v : S, proj σ J v (ofInf σ J (embArchPi σ J a) * embAt σ J S b) = b v) ∧ ∀ v ∉ S, proj σ J v (ofInf σ J (embArchPi σ J a) * embAt σ J S b) = 1

theorem **isCompact_of_ext** {X : Type} [TopologicalSpace X] {s t : Set X} (h : IsCompact t) (e : ∀ x, x ∈ s ↔ x ∈ t) : IsCompact s

theorem **isRestrGroup_GAd** : IsRestrGroup (projArch σ J) (proj σ J) (stabLoc σ J L) where

theorem **isRestrGroup_GFin** : IsRestrGroup (H

theorem **exists_away** (S : Finset {v // v ∉ T}) (x : ∀ w, GArch σ J w) (y : ∀ v : S, GLoc σ J v.1) : ∃ g ∈ awayPts σ J T, (∀ w, projArch σ J w g = x w) ∧ (∀ v : S, proj σ J v.1 g = y v) ∧ ∀ v ∉ S, proj σ J v.1 g ∈ stabLoc σ J L v

theorem **isClosed_finAwayPts** : IsClosed (finAwayPts σ J T : Set (GAd σ J))

@[simp] theorem **haarLoc_stabLoc** (v : HeightOneSpectrum (𝒪 F)) : haarLoc σ J L v (stabLoc σ J L v) = 1

theorem **exists_isProdMeasure_of_haar** (μ : Measure (GAd σ J)) [μ.IsHaarMeasure] : ∃ μi : ∀ w, Measure (GArch σ J w), IsProdMeasure σ J L μi (haarLoc σ J L) μ

Development.AbelianVarietyTheory.Definitions.Foundations.ExistsUnitaryMulTriangular

@[simp] theorem **evalAt_X** (y : Matrix (Fin N) (Fin N) R) (v : Fin N × Fin N) : evalAt y (X v) = y v.1 v.2

theorem **evalAt_rowAct** (y g : Matrix (Fin N) (Fin N) R) (f : Pol N R) : evalAt y (rowAct R g f) = evalAt (g * y) f

theorem **evalAt_colAct** (y b : Matrix (Fin N) (Fin N) R) (f : Pol N R) : evalAt y (colAct b f) = evalAt (y * b) f

theorem **evalAt_map** (φ : R →+* S) (y : Matrix (Fin N) (Fin N) R) (f : Pol N R) : evalAt (y.map φ) (map φ f) = φ (evalAt y f)

theorem **map_colAct** (φ : R →+* S) (b : Matrix (Fin N) (Fin N) R) (f : Pol N R) : map φ (colAct b f) = colAct (b.map φ) (map φ f)

theorem **colAct_initMinor** {s : ℕ} (I : Fin s → Fin N) (hs : s ≤ N) (b : Matrix (Fin N) (Fin N) R) (hb : b.BlockTriangular id) : colAct b (initMinor R I hs) = diagChar (initW N s) b • initMinor R I hs

theorem **colAct_of_mem** [IsDomain R] {la : Fin N → ℕ} {f : Pol N R} (hf : f ∈ nabla la R) (b : Matrix (Fin N) (Fin N) R) (hb : b.BlockTriangular id) : colAct b f = diagChar la b • f

theorem **colAct_map_of_mem** [IsDomain S] (φ : R →+* S) {la : Fin N → ℕ} {f : Pol N R} (hf : f ∈ nabla la R) (b : Matrix (Fin N) (Fin N) S) (hb : b.BlockTriangular id) : colAct b (map φ f) = diagChar la b • map φ f

theorem **coeff_colAct_diagonal** (t : Fin N → R) (f : Pol N R) (m : Fin N × Fin N →₀ ℕ) : coeff m (colAct (Matrix.diagonal t) f) = (∏ s, t s ^ (Finsupp.weight (colW N N) m s)) * coeff m f

theorem **mem_nabla_iff** [IsDomain R] [Infinite R] {la : Fin N → ℕ} {f : Pol N R} : f ∈ nabla la R ↔ ∀ b : Matrix (Fin N) (Fin N) R, b.BlockTriangular id → (∀ i, b i i ≠ 0) → colAct b f = diagChar la b • f

theorem **colLen_le** (la : Fin N → ℕ) (c : ℕ) : colLen la c ≤ N

theorem **isStandard_canon** (la : Fin N → ℕ) : IsStandard (canon la)

theorem **extremal_mem** (hla : Antitone la) : extremal la R ∈ nabla la R

theorem **evalAt_one_extremal** : evalAt 1 (extremal la R) = 1

theorem **extremal_ne_zero** [Nontrivial R] : extremal la R ≠ 0

theorem **nabla_ne_bot** [Nontrivial R] (hla : Antitone la) : nabla la R ≠ ⊥

theorem **coe_rho** (g : Matrix (Fin N) (Fin N) R) (f : nabla la R) : (rho la R g f : Pol N R) = rowAct R gᵀ f

theorem **coe_baseChange** (φ : R →+* S) (f : nabla la R) : (baseChange la φ f : Pol N S) = map φ f.1

theorem **evalAt_rho** (y g : Matrix (Fin N) (Fin N) R) (f : nabla la R) : evalAt y (rho la R g f : Pol N R) = evalAt (gᵀ * y) f

theorem **rho_one** (f : nabla la R) : rho la R 1 f = f

theorem **rho_smul** (t : R) (g : Matrix (Fin N) (Fin N) R) (f : nabla la R) : rho la R (t • g) f = t ^ size la • rho la R g f

theorem **rho_scalar** (t : R) (f : nabla la R) : rho la R (t • 1) f = t ^ size la • f

theorem **baseChange_rho** (φ : R →+* S) (g : Matrix (Fin N) (Fin N) R) (f : nabla la R) : baseChange la φ (rho la R g f) = rho la S (g.map φ) (baseChange la φ f)

theorem **intVec_spec** (la : Fin N → ℕ) (K : Type) [Field K] [CharZero K] : LinearIndependent K (fun i => map (Int.castRingHom K) (intVec la i)) ∧ Submodule.span K (Set.range fun i => map (Int.castRingHom K) (intVec la i)) = nabla la K

theorem **coe_basis** (la : Fin N → ℕ) (K : Type) [Field K] [CharZero K] (i : Idx la) : (basis la K i : Pol N K) = map (Int.castRingHom K) (intVec la i)

theorem **dim_eq_card** (la : Fin N → ℕ) : dim la = Fintype.card (Idx la)

theorem **dim_pos** {la : Fin N → ℕ} (hla : Antitone la) : 0 < dim la

theorem **rhoU_apply** (u : UN N) : rhoU la u = rho la ℂ u.1

theorem **continuous_evalAt** (f : Pol N ℂ) : Continuous fun y : Matrix (Fin N) (Fin N) ℂ => evalAt y f

theorem **continuous_evalAt_coe** (f : Pol N ℂ) : Continuous fun g : UN N => evalAt g.1 f

theorem **integrable_mul_conj** (f f' : Pol N ℂ) : Integrable (fun g : UN N => evalAt g.1 f * conj (evalAt g.1 f')) (haarU N)

theorem **integrable_norm_sq** (f : Pol N ℂ) : Integrable (fun g : UN N => ‖evalAt g.1 f‖ ^ 2) (haarU N)

theorem **form_self** (f : Pol N ℂ) : form la f f = normSq la f

theorem **normSq_nonneg** (f : Pol N ℂ) : 0 ≤ normSq la f

theorem **conj_form** (f f' : Pol N ℂ) : conj (form la f f') = form la f' f

theorem **form_add_left** (f₁ f₂ f' : Pol N ℂ) : form la (f₁ + f₂) f' = form la f₁ f' + form la f₂ f'

theorem **form_smul_left** (c : ℂ) (f f' : Pol N ℂ) : form la (c • f) f' = c * form la f f'

theorem **form_add_right** (f f₁ f₂ : Pol N ℂ) : form la f (f₁ + f₂) = form la f f₁ + form la f f₂

theorem **form_smul_right** (c : ℂ) (f f' : Pol N ℂ) : form la f (c • f') = conj c * form la f f'

theorem **form_rowAct** (u : UN N) (f f' : Pol N ℂ) : form la (rowAct ℂ u.1 f) (rowAct ℂ u.1 f') = form la f f'

theorem **transpose_mem_unitaryGroup** {u : Matrix (Fin N) (Fin N) ℂ} (hu : u ∈ Matrix.unitaryGroup (Fin N) ℂ) : uᵀ ∈ Matrix.unitaryGroup (Fin N) ℂ

theorem **form_rhoU** (u : UN N) (f f' : nabla la ℂ) : form la (rhoU la u f : Pol N ℂ) (rhoU la u f' : Pol N ℂ) = form la f f'

theorem **normSq_rhoU** (u : UN N) (f : nabla la ℂ) : normSq la (rhoU la u f : Pol N ℂ) = normSq la f

theorem **exists_unitary_mul_triangular** (y : Matrix (Fin N) (Fin N) ℂ) : ∃ (u : UN N) (b : Matrix (Fin N) (Fin N) ℂ), b.BlockTriangular id ∧ y = u.1 * b

theorem **eq_zero_of_evalAt_unitary** {f : Pol N ℂ} (hf : f ∈ nabla la ℂ) (h : ∀ g : UN N, evalAt g.1 f = 0) : f = 0

theorem **normSq_pos** {f : Pol N ℂ} (hf : f ∈ nabla la ℂ) (h0 : f ≠ 0) : 0 < normSq la f

theorem **normSq_le_dim** {f : Pol N ℂ} (h : ∀ g : UN N, ‖evalAt g.1 f‖ ≤ 1) : normSq la f ≤ dim la

theorem **inner_toSpace** (f f' : nabla la ℂ) : ⟪toSpace la f, toSpace la f'⟫_ℂ = form la f'.1 f.1

theorem **norm_toSpace_sq** (f : nabla la ℂ) : ‖toSpace la f‖ ^ 2 = normSq la f.1

theorem **finrank_space** : Module.finrank ℂ (Space la) = dim la

theorem **rhoSpace_toSpace** (u : UN N) (f : nabla la ℂ) : rhoSpace la u (toSpace la f) = toSpace la (rhoU la u f)

theorem **norm_rhoSpace** (u : UN N) (v : Space la) : ‖rhoSpace la u v‖ = ‖v‖

theorem **continuous_of_span_dual** {V T : Type*} [NormedAddCommGroup V] [NormedSpace ℂ V] [FiniteDimensional ℂ V] [TopologicalSpace T] (s : Set (Module.Dual ℂ V)) (hs : Submodule.span ℂ s = ⊤) (F : T → V) (h : ∀ l ∈ s, Continuous fun t => l (F t)) : Continuous F

theorem **evSpace_toSpace** (y : Matrix (Fin N) (Fin N) ℂ) (f : nabla la ℂ) : evSpace la y (toSpace la f) = evalAt y f.1

theorem **span_evSpace** : Submodule.span ℂ (Set.range fun g : UN N => evSpace la g.1) = ⊤

theorem **continuous_rhoSpace** (v : Space la) : Continuous fun u : UN N => rhoSpace la u v

theorem **baseChange_mem_stdLat** (f : nabla la (𝒪 K)) : baseChange la (algebraMap (𝒪 K) K) f ∈ stdLat la K

theorem **mem_stdLat** {f : nabla la K} : f ∈ stdLat la K ↔ ∃ g : nabla la (𝒪 K), baseChange la (algebraMap (𝒪 K) K) g = f

theorem **mem_stdLat_iff_coeff** {f : nabla la K} : f ∈ stdLat la K ↔ ∀ m, coeff m f.1 ∈ (algebraMap (𝒪 K) K).range

theorem **stdLat_eq_span** : stdLat la K = Submodule.span (𝒪 K) (Set.range (basis la K))

theorem **rho_mem_locP** (w : HeightOneSpectrum (𝒪 K)) {g : Matrix (Fin N) (Fin N) K} {t : 𝒪 K} (ht : t ∈ w.asIdeal.primeCompl) {A : Matrix (Fin N) (Fin N) (𝒪 K)} (hA : A.map (algebraMap (𝒪 K) K) = algebraMap (𝒪 K) K t • g) {f : nabla la K} (hf : f ∈ localization₂ w.asIdeal (stdLat la K)) : rho la K g f ∈ localization₂ w.asIdeal (stdLat la K)

theorem **mem_colLat_iff** (B : GL (Fin N) K) (x : Fin N → K) : x ∈ colLat B ↔ ∃ c : Fin N → 𝒪 K, (B : Matrix (Fin N) (Fin N) K) *ᵥ (fun j => (c j : K)) = x

theorem **isLattice_colLat** (B : GL (Fin N) K) : Submodule.IsLattice K (colLat B)

Development.AbelianVarietyTheory.Definitions.Foundations.FacWithCsNonneg

theorem **comp_nonempty** (h : T.Pins) (hj : InR D j) : Nonempty (Comp Xo c)

theorem **facWith_T** : (F).T = tR Q

theorem **facWith_z** : (F).z = zed Q j

theorem **facWith_eps** : (F).ε = epsR Q j

theorem **facWith_U** : (F).U = nU D j

theorem **facWith_A** : (𝐅).A = 26 * (26 * (𝐅).DY ^ 2 + 2 * (𝐅).z * (𝐅).T)

theorem **facWith_main** : (𝐅).main = ∑ τ, ((pp D j : ℝ) - qq D j) * wj D j τ * L.logOm

theorem **facWith_DY** : (𝐅).DY = dimFlag (T.Dw j)

theorem **facWith_rho** : (𝐅).ρ = GlobalTheory.CompHt.rhoB (T.Dw j) (toWt (T.Dw j) (uOne Q j))

theorem **facWith_delPrime** (hv : ∀ Z Z' : Comp Xo c, vZero U Z = vZero U Z') (Z : Comp Xo c) : (𝐅).δ' = delPrime U Z (toWt (T.Dw j) (uOne Q j))

theorem **facWith_sig** : ((𝐅).q' : ℝ) + 1 ≤ (pageOf Q).m ∧ (𝐅).pq = (pageOf Q).k + (pageOf Q).m - 1 - 2 * ((𝐅).q' : ℝ)

theorem **facWith_q** : (𝐅).q' = D.R.m - j

theorem **facWith_cs_nonneg** : 0 ≤ (𝐅).cs

theorem **facWith_muA** (hG : GoodAt U dict (gKapMinus T j)) : (𝐅).muA = slopeW_sq U (xiR D) (fun _ => D.sp.W j) dict (gKapMinus T j)

theorem **facWith_muN** (hG : HasGram (nrmAt U dict (gKap T j))) (hN : NN.FG) : (𝐅).muN = slope_sq (nrmAt U dict (gKap T j)) NN

theorem **facWith_mmax** (hG : GoodAt U dict (gKap T j)) (h0 : latW U (gKap T j) ≠ ⊥) : (𝐅).mmax = slopeMax_sq (nrmAt U dict (gKap T j)) (latW U (gKap T j))

theorem **facWith_good** (hG : HasGram (nrmAt U dict (gKap T j))) : (𝐅).good = latCo (nrmAt U dict (gKap T j)) (NN ⊓ (LinearMap.range e).restrictScalars (𝒪 c.F))

theorem **facWith_bad** (hG : HasGram (nrmAt U dict (gKap T j))) : (𝐅).bad = latCo (nrmAt U dict (gKap T j)) (NN ⊓ (LinearMap.ker e).restrictScalars (𝒪 c.F))

theorem **facWith_muG** (hG : HasGram (nrmAt U dict (gKap T j))) (hN : (NN ⊓ (LinearMap.range e).restrictScalars (𝒪 c.F)).FG) : (𝐅).muG = slope_sq (nrmAt U dict (gKap T j)) (NN ⊓ (LinearMap.range e).restrictScalars (𝒪 c.F))

theorem **facWith_rk** (hG : HasGram (nrmAt U dict (gKap T j))) : rkK (𝐅).NN = rkK NN ∧ rkK (𝐅).bad = rkK (NN ⊓ (LinearMap.ker e).restrictScalars (𝒪 c.F)) ∧ rkK (𝐅).LB = rkK (piLat Hb NN)

theorem **facWith_rkA** (hG : HasGram (nrmAt U dict (gKapMinus T j))) : rkK (𝐅).LA = rankW U (gKapMinus T j)

Development.AbelianVarietyTheory.Definitions.Foundations.ForallEventuallyFinLocIff

theorem **isSmooth_smoothRep** (ρ : Representation ℂ G X) : IsSmooth (smoothRep ρ)

theorem **continuous_embPl** : ∀ v, Continuous (embPl σ J v) | .inl w => continuous_embArch σ J w | .inr v => continuous_emb σ J v @[simp] theorem projPl_embPl_self : ∀ v x, projPl σ J v (embPl σ J v x) = x | .inl w, x => projArch_embArch_self σ J w x | .inr v, x => proj_emb_self σ J v x theorem projPl_embPl_ne : ∀ {u v}, u ≠ v → ∀ x, projPl σ J u (embPl σ J v x) = 1 | .inl _, .inl _, h, x => projArch_embArch_ne σ J (fun e => h (congrArg _ e)) x | .inl u, .inr v, _, x => projArch_emb σ J u v x | .inr u, .inl v, _, x => proj_embArch σ J u v x | .inr _, .inr _, h, x => proj_emb_ne σ J (fun e => h (congrArg _ e)) x theorem forall_eventually_proj_iff (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (a : ∀ v, GLoc σ J v → Prop) : (∀ g : GFin σ J, ∀ᶠ v in cofinite, a v (projFinLoc σ J v g)) ↔ ∀ᶠ v in cofinite, ∀ k ∈ stabLoc σ J L v, a v k

theorem **IsFactorisation.eventually_fixed** (h : IsFactorisation σ J δ ξ π B) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] : ∀ᶠ v in cofinite, ∀ k ∈ stabLoc σ J L v, δ (.inr v) k (ξ (.inr v)) = ξ (.inr v)

theorem **IsFactorisation.apply_embPl** [DecidableEq (Formalism.MainNotions.SchurIrreducible.Pl F)] (h : IsFactorisation σ J δ ξ π B) {y : ∀ v, Hs v} (hy : Restr ξ y) (v : Formalism.MainNotions.SchurIrreducible.Pl F) (g : GPl σ J v) : π (embPl σ J v g) (B y) = B (Function.update y v (δ v g (y v)))

theorem **IsFactorisation.hasCompPl** (h : IsFactorisation σ J δ ξ π B) (v : Formalism.MainNotions.SchurIrreducible.Pl F) : HasCompPl σ J v (δ v) π

theorem **mem_KOff** {S : Finset (HeightOneSpectrum (𝒪 D.host.F))} {g : GfC V} : g ∈ KOff V S ↔ (g : GA D V) ∈ KsphC V S

@[simp] theorem **coe_locFin** (v : HeightOneSpectrum (𝒪 D.host.F)) (x : Gv V v) : (locFin V v x : GA D V) = embV V v x

@[simp] theorem **coe_archInf** (w : InfinitePlace D.host.F) (x : Gw V w) : (archInf V w x : GA D V) = embW V w x

theorem **toInf_mul_toFin** (g : GA D V) : (toInf V g : GA D V) * toFin V g = g

theorem **commute_inf_fin** (x : GinfC V) (y : GfC V) : Commute (x : GA D V) y

@[simp] theorem **toFin_coe** (y : GfC V) : toFin V y = y

@[simp] theorem **toFin_coe_inf** (x : GinfC V) : toFin V x = 1

@[simp] theorem **toInf_coe_fin** (y : GfC V) : toInf V y = 1

@[simp] theorem **toInf_coe** (x : GinfC V) : toInf V x = x

theorem **continuous_toFin** : Continuous (toFin V)

theorem **continuous_toInf** : Continuous (toInf V)

theorem **isIrrUnitary_dsRep** (M : DSModel V S) (τ : D.what₂.Tau) : IsIrrUnitary (dsRep M τ)

theorem **mem_homDS** {θ : InfF D} (hθ : IsInfFam θ) {M : DSModel V S} {ρ : GA D V →* (H →L[ℂ] H)} {U : Submodule ℂ H} {Φ : ContinuousMultilinearMap ℂ M.H H} : Φ ∈ homDS M ρ U ↔ (∀ y, Φ y ∈ U) ∧ ∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ), ∀ y, Φ (Function.update y τ (M.rep τ g (y τ))) = ρ (atTau θ V τ g) (Φ y)

theorem **IsFinInf.isUnitaryRep** (h : IsFinInf V πf πi π) (hf : IsUnitaryRep πf) (hi : IsUnitaryRep πi) : IsUnitaryRep π

theorem **IsFinInf.uEquiv** [CompleteSpace H] (h : IsFinInf V πf πi π) : UEquiv (finInf V πf πi) π

Development.AbelianVarietyTheory.Definitions.Foundations.FoundationsPrimeFact

theorem **powMod_succ** (a m k e : ℕ) : powMod a m (k + 1) e = (powMod a m k (e / 2) ^ 2 * a ^ (e % 2)) % m

theorem **powMod_cast** (a m k : ℕ) : ∀ e : ℕ, e < 2 ^ k → ((powMod a m k e : ℕ) : ZMod m) = (a : ZMod m) ^ e

theorem **mem_of_dvd_prod_pow** {l : List ℕ} (hl : ∀ q ∈ l, q.Prime) {n N q : ℕ} (hn : n ∣ l.prod ^ N) (hq : q.Prime) (h : q ∣ n) : q ∈ l

theorem **prime_of_cert** (p a k N : ℕ) (l : List ℕ) (hl : ∀ q ∈ l, q.Prime) (hp : 1 < p) (hk : p < 2 ^ k) (hN : p - 1 ∣ l.prod ^ N) (h1 : powMod a p k (p - 1) % p = 1) (h2 : ∀ q ∈ l, powMod a p k ((p - 1) / q) % p ≠ 1) : p.Prime

theorem **foundations_prime_fact_ter** : Nat.Prime 10944929

theorem **prime_128860109809** : Nat.Prime 128860109809

theorem **foundations_prime_fact** : Nat.Prime 328347841

theorem **foundations_prime_fact_bis** : Nat.Prime 1395478321

theorem **siteGood_prime_torsion** : Nat.Prime (1 + 599 * (123 * (16 * 571 * 599 * (10944929 * 128860109809) ^ 2)) ^ 2)

theorem **not_exc** {g e l k : ℕ} (hl : 1 < l) (he : 1 ≤ e) (hg : l ^ (4 * k) ≤ g) (h : ¬ l ^ (k + 1) ∣ g ^ e - 1) : ¬ Formalism.CyclotomicNorms.Exc g e l

theorem **bb_disjoint** : Disjoint (Formalism.CyclotomicNorms.BB 9 10944929) (Formalism.CyclotomicNorms.BB 9 128860109809)

theorem **hyp** : HostHyp 571 599

theorem **level_prime** : level.Prime

theorem **cm_nonempty** : (ax hl).cmType.Nonempty

theorem **half_exists** : ∃ 𝔟 : Ideal (𝒪 (ax hl).E), differentIdeal (𝒪 (ax hl).K) (𝒪 (ax hl).E) = 𝔟 * ((ax hl).conj • 𝔟) ∧ 𝔟 ⊔ (ax hl).conj • 𝔟 = ⊤ ∧ ∀ σ : (ax hl).E ≃ₐ[ℚ] (ax hl).E, (ax hl).gal σ ∈ R0.Psi → σ • 𝔟 = 𝔟

theorem **norm_one_add** {R : Ranks} (H : Host R) (c : ℕ) : ((1 + c * rootInt H : 𝒪 H.E) : H.E) * H.conj ((1 + c * rootInt H : 𝒪 H.E) : H.E) = ((1 + H.d * c ^ 2 : ℕ) : H.E)

theorem **one_add_mem** {R : Ranks} (H : Host R) (c : ℕ) : ((1 + c * rootInt H : 𝒪 H.E) : H.E) ∈ H.K

theorem **window** {x s : ℝ} (hx : 0 < x) (hs : 0 < s) : (4 * x ^ s / ((1 : ℕ) : ℝ)) ^ ((1 : ℝ) / s) * (2 : ℝ) ^ (-(2 : ℝ) / s) = x

theorem **gen_norm** : letI

theorem **pre_hcW_injective** : Function.Injective (pre hl).hcW

theorem **GlobalTheory.Mem.exists_pre** : ∃ B : GlobalTheory.Mem.Pre, Fintype.card B.Word = 2 ∧ Function.Injective B.hcW

Development.AbelianVarietyTheory.Definitions.Foundations.FrobLiftWittClosure

theorem **frobLift_witt** (p : ℕ) [hp : Fact p.Prime] (k : Type) [Field k] [CharP k p] [PerfectRing k p] : FrobLift (WittVector p k) p (WittVector.frobeniusEquiv p k) where

theorem **frobLift_witt_closure** (p : ℕ) [hp : Fact p.Prime] : FrobLift (WittVector p (AlgebraicClosure (ZMod p))) p (WittVector.frobeniusEquiv p (AlgebraicClosure (ZMod p)))

theorem **om_order** (hu : (p : R) ∈ nonunits R) : ((om p : PowerSeries R).map (Ideal.Quotient.mk (Ideal.span {(p : R)}))).order.toNat = p

theorem **om_div** (hu : (p : R) ∈ nonunits R) : (om p : PowerSeries R).IsWeierstrassDivisorAt (Ideal.span {(p : R)})

theorem **phi_split** (c : R) (g : PowerSeries R) : phi p hp.out.pos (C c + X * g) = C c + om p * phi p hp.out.pos g

theorem **eq_zero_of_X_pow_dvd** (f : PowerSeries R) (h : ∀ n, X ^ n ∣ f) : f = 0

private lemma **exists_eq_smul_of_map_quotient_eq_zero** (g : R⟦X⟧) (hg : PowerSeries.map (Ideal.Quotient.mk (Ideal.span {(p : R)})) g = 0) : ∃ d : R⟦X⟧, g = (p : R) • d

private lemma **prime_dvd_coeff_one_add_X_pow_sub_one_sub_X_pow** (n : ℕ) : (p : R) ∣ coeff n (((1 : PowerSeries R) + X) ^ p - 1 - X ^ p)

private lemma **map_quotient_one_add_X_pow_sub_one** : PowerSeries.map (Ideal.Quotient.mk (Ideal.span {(p : R)})) ((1 + X) ^ p - 1 : R⟦X⟧) = PowerSeries.map (Ideal.Quotient.mk (Ideal.span {(p : R)})) (X ^ p)

private lemma **map_quotient_phi_eq_map_expand** (f : R⟦X⟧) : PowerSeries.map (Ideal.Quotient.mk (Ideal.span {(p : R)})) (_root_.GlobalTheory.Col.phi p hp.out.pos f) = PowerSeries.map (Ideal.Quotient.mk (Ideal.span {(p : R)})) (expand p hp.out.ne_zero f)

private lemma **exists_phi_sub_expand_eq_smul** (f : R⟦X⟧) : ∃ d : R⟦X⟧, _root_.GlobalTheory.Col.phi p hp.out.pos f - expand p hp.out.ne_zero f = (p : R) • d

private lemma **exists_sum_phi_eq_sum_expand_add_smul** (g : Fin p → R⟦X⟧) : ∃ D : R⟦X⟧, ∑ i, _root_.GlobalTheory.Col.phi p hp.out.pos (g i) * X ^ (i : ℕ) = ∑ i, expand p hp.out.ne_zero (g i) * X ^ (i : ℕ) + (p : R) • D

private lemma **not_dvd_mul_add_sub_of_ne** {n : ℕ} {i j : Fin p} (hij : i ≠ j) (hle : (i : ℕ) ≤ p * n + j) : ¬ p ∣ p * n + j - i

private lemma **coeff_sum_expand_mul_X_pow** (g : Fin p → R⟦X⟧) (n : ℕ) (j : Fin p) : coeff (p * n + j) (∑ i, expand p hp.out.ne_zero (g i) * X ^ (i : ℕ)) = coeff n (g j)

private lemma **sum_expand_mk_coeff_mul_X_pow_eq** (h : R⟦X⟧) : ∑ i : Fin p, expand p hp.out.ne_zero (PowerSeries.mk fun n => coeff (p * n + i) h) * X ^ (i : ℕ) = h

private lemma **eq_zero_of_sum_expand_mul_X_pow_eq_zero** (g : Fin p → R⟦X⟧) (hg : ∑ i, expand p hp.out.ne_zero (g i) * X ^ (i : ℕ) = 0) (j : Fin p) : g j = 0

theorem **_root_.GlobalTheory.Col.Model.indep** (hu : (p : R) ∈ nonunits R) (g : Fin p → PowerSeries R) (hg : ∑ i, _root_.GlobalTheory.Col.phi p hp.out.pos (g i) * X ^ (i : ℕ) = 0) (i : Fin p) : g i = 0

private lemma **mem_span_pow_smul_univ_iff_eq_pow_mul** (K : ℕ) (x : R) : x ∈ ((Ideal.span {(p : R)}) ^ K • ⊤ : Submodule R R) ↔ ∃ c, x = (p : R) ^ K * c

theorem **_root_.GlobalTheory.Col.Model.spans** (hu : (p : R) ∈ nonunits R) (h : PowerSeries R) : ∃ g : Fin p → PowerSeries R, h = ∑ i, _root_.GlobalTheory.Col.phi p hp.out.pos (g i) * X ^ (i : ℕ)

theorem **normOp_eq** (h : PowerSeries R) : normOp p h = Algebra.norm (PowerSeries R) (S

theorem **trOp_eq** (h : PowerSeries R) : trOp p h = Algebra.trace (PowerSeries R) (Up R p) h

theorem **normOp_phi** (hu : (p : R) ∈ nonunits R) (g : PowerSeries R) : normOp p (phi p hp.out.pos g) = g ^ p

theorem **trOp_phi** (hu : (p : R) ∈ nonunits R) (g : PowerSeries R) : trOp p (phi p hp.out.pos g) = p • g

theorem **frobLift_ops** {R : Type} [CommRing R] (p : ℕ) [hp : Fact p.Prime] (σ : R ≃+* R) (F : FrobLift R p σ) (g : PowerSeries R) : normOp p (phi p hp.out.pos g) = g ^ p ∧ trOp p (phi p hp.out.pos g) = p • g

theorem **up_ev** (g : PowerSeries R) : up p m hm (ev R p m g) = ev R p (m + 1) (phi p hp.out.pos g)

theorem **nrm_eq** (x : lev R p (m + 1)) : nrm p m hm x = Algebra.norm (lev R p m) (S

theorem **_root_.GlobalTheory.Col.Model.indepL** (hu : (p : R) ∈ nonunits R) (x : Fin p → _root_.GlobalTheory.Col.lev R p m) (hx : ∑ i, _root_.GlobalTheory.Col.up p m hm (x i) * _root_.GlobalTheory.Col.ev R p (m + 1) (X ^ (i : ℕ)) = 0) (i : Fin p) : x i = 0

theorem **_root_.GlobalTheory.Col.Model.nrm_ev** (hu : (p : R) ∈ nonunits R) (h : PowerSeries R) : _root_.GlobalTheory.Col.nrm p m hm (_root_.GlobalTheory.Col.ev R p (m + 1) h) = _root_.GlobalTheory.Col.ev R p m (_root_.GlobalTheory.Col.normOp p h)

theorem **lev_nontrivial** {R : Type} [CommRing R] (p : ℕ) [hp : Fact p.Prime] (hu : (p : R) ∈ nonunits R) (m : ℕ) : Nontrivial (lev R p (m + 1))

theorem **dd_test** {R : Type} [CommRing R] (p : ℕ) (hp : 0 < p) : dd (phi (R

Development.AbelianVarietyTheory.Definitions.Foundations.GLintEqKOAndPrincEqKOne

theorem **dualConj_val** (A x : GL m K) : (dualConj A x).1 = (A.1 ∗ (x⁻¹).1 ∗ (A⁻¹).1)ᵀ

theorem **eq_dualConj_iff** {A x y : GL m K} : y = dualConj A x ↔ y.1ᵀ ∗ A.1 ∗ x.1 = A.1

theorem **continuous_dualConj_val** [TopologicalSpace K] [IsTopologicalRing K] (A : GL m K) : Continuous fun x : GL m K => (dualConj A x).1

theorem **mem_GLint_iff** {O : Subring K} {x : GL m K} : x ∈ GLint m O ↔ (∀ i j, x.1 i j ∈ O) ∧ ∀ i j, (x⁻¹).1 i j ∈ O

theorem **bc_injective** {A B : Type} [CommRing A] [CommRing B] {φ : A →+∗ B} (hφ : Function.Injective φ) : Function.Injective (bc (n

theorem **bc_residue_surjective** (O : Type) [CommRing O] [IsLocalRing O] : Function.Surjective (bc (n

theorem **splitVec_add** (τ : E →ₐ[F] Fv F v) (y z : Fin n → Fv F v ⊗[F] E) : splitVec τ (y + z) = splitVec τ y + splitVec τ z

theorem **splitVec_smul** (τ : E →ₐ[F] Fv F v) (c : Fv F v ⊗[F] E) (y : Fin n → Fv F v ⊗[F] E) : splitVec τ (c • y) = splitProj τ c • splitVec τ y

theorem **splitVec_mulVec** (τ : E →ₐ[F] Fv F v) (M : Matrix (Fin n) (Fin n) (Fv F v ⊗[F] E)) (y : Fin n → Fv F v ⊗[F] E) : splitVec τ (M *ᵥ y) = M.map (splitProj τ) *ᵥ splitVec τ y

@[simp] theorem **splitVec_splitIncl** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (u : Fin n → Fv F v) : splitVec τ (splitIncl h τ u) = u

@[simp] theorem **splitVec_comp_splitIncl** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (u : Fin n → Fv F v) : splitVec (τ.comp σ) (splitIncl h τ u) = 0

@[simp] theorem **splitVec_splitIncl_comp** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (u : Fin n → Fv F v) : splitVec τ (splitIncl h (τ.comp σ) u) = 0

theorem **splitIncl_splitVec** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (y : Fin n → Fv F v ⊗[F] E) : splitIncl h τ (splitVec τ y) = splitIdem h τ • y

theorem **splitVec_ext** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) {y z : Fin n → Fv F v ⊗[F] E} (h1 : splitVec τ y = splitVec τ z) (h2 : splitVec (τ.comp σ) y = splitVec (τ.comp σ) z) : y = z

@[simp] theorem **splitGram_val** (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) : (splitGram hJ τ).1 = J.map τ

theorem **map_comp_conj** (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) : J.map (τ.comp σ) = (J.map τ)ᵀ

theorem **JR_map_splitProj** (J : Matrix (Fin n) (Fin n) E) (τ : E →ₐ[F] Fv F v) : (JR (Fv F v) J).map (splitProj τ) = J.map τ

theorem **splitProj_hLin** (σ : E →ₐ[F] E) (J : Matrix (Fin n) (Fin n) E) (τ : E →ₐ[F] Fv F v) (y x : Fin n → Fv F v ⊗[F] E) : splitProj τ (hLin σ J (Fv F v) y x) = splitVec (τ.comp σ) x ⬝ᵥ J.map τ *ᵥ splitVec τ y

theorem **splitProj_comp_hLin** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) (y x : Fin n → Fv F v ⊗[F] E) : splitProj (τ.comp σ) (hLin σ J (Fv F v) y x) = splitVec (τ.comp σ) y ⬝ᵥ J.map τ *ᵥ splitVec τ x

theorem **splitHom_val** (τ : E →ₐ[F] Fv F v) (g : GLoc σ J v) : (splitHom σ J τ g).1 = (matHom _ σ J g).1.map (splitProj τ)

theorem **splitVec_matHom_mulVec** (τ : E →ₐ[F] Fv F v) (g : GLoc σ J v) (y : Fin n → Fv F v ⊗[F] E) : splitVec τ ((matHom _ σ J g).1 *ᵥ y) = (splitHom σ J τ g).1 *ᵥ splitVec τ y

theorem **matHom_mulVec_splitIncl** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (g : GLoc σ J v) (u : Fin n → Fv F v) : (matHom _ σ J g).1 *ᵥ splitIncl h τ u = splitIncl h τ ((splitHom σ J τ g).1 *ᵥ u)

theorem **map_unitary** (σ : E →ₐ[F] E) (J : Matrix (Fin n) (Fin n) E) (τ : E →ₐ[F] Fv F v) (M : Matrix (Fin n) (Fin n) (Fv F v ⊗[F] E)) : (cT (sigR _ σ) M ∗ JR (Fv F v) J ∗ M).map (splitProj τ) = (M.map (splitProj (τ.comp σ)))ᵀ ∗ J.map τ ∗ M.map (splitProj τ)

theorem **splitHom_unitary** (τ : E →ₐ[F] Fv F v) (g : GLoc σ J v) : (splitHom σ J (τ.comp σ) g).1ᵀ ∗ J.map τ ∗ (splitHom σ J τ g).1 = J.map τ

theorem **pairing_splitHom** (τ : E →ₐ[F] Fv F v) (g : GLoc σ J v) (u' u : Fin n → Fv F v) : ((splitHom σ J (τ.comp σ) g).1 *ᵥ u') ⬝ᵥ J.map τ *ᵥ ((splitHom σ J τ g).1 *ᵥ u) = u' ⬝ᵥ J.map τ *ᵥ u

theorem **splitHom_comp** (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) (g : GLoc σ J v) : splitHom σ J (τ.comp σ) g = dualConj (splitGram hJ τ) (splitHom σ J τ g)

theorem **splitMat_ext** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) {M N : Matrix (Fin n) (Fin n) (Fv F v ⊗[F] E)} (h1 : M.map (splitProj τ) = N.map (splitProj τ)) (h2 : M.map (splitProj (τ.comp σ)) = N.map (splitProj (τ.comp σ))) : M = N

@[simp] theorem **splitPairMat_map** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (a b : Matrix (Fin n) (Fin n) (Fv F v)) : (splitPairMat h τ a b).map (splitProj τ) = a

@[simp] theorem **splitPairMat_map_comp** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (a b : Matrix (Fin n) (Fin n) (Fv F v)) : (splitPairMat h τ a b).map (splitProj (τ.comp σ)) = b

theorem **splitPairMat_mul_inv** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (x y : GL (Fin n) (Fv F v)) : splitPairMat h τ x.1 y.1 ∗ splitPairMat h τ (x⁻¹).1 (y⁻¹).1 = 1

theorem **splitPairMat_unitary** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) {a b : Matrix (Fin n) (Fin n) (Fv F v)} (e : bᵀ ∗ J.map τ ∗ a = J.map τ) : cT (sigR _ σ) (splitPairMat h τ a b) ∗ JR (Fv F v) J ∗ splitPairMat h τ a b = JR (Fv F v) J

theorem **matHom_splitLift** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) (x : GL (Fin n) (Fv F v)) : matHom _ σ J (splitLift h hJ τ x) = splitPairGL h τ x (dualConj (splitGram hJ τ) x)

theorem **splitHom_splitLift** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) (x : GL (Fin n) (Fv F v)) : splitHom σ J τ (splitLift h hJ τ x) = x

theorem **splitHom_injective** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) : Function.Injective (splitHom σ J τ)

theorem **continuous_splitHom** (τ : E →ₐ[F] Fv F v) : Continuous (splitHom σ J τ)

theorem **continuous_splitLift** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) : Continuous (splitLift h hJ τ)

theorem **splitEquiv_apply** (g : GLoc σ J v) : splitEquiv h hJ τ g = splitHom σ J τ g

theorem **splitEquiv_val** (g : GLoc σ J v) : (splitEquiv h hJ τ g).1 = (matHom _ σ J g).1.map (splitProj τ)

theorem **splitEquiv_symm_apply** (x : GL (Fin n) (Fv F v)) : (splitEquiv h hJ τ).symm x = splitLift h hJ τ x

theorem **splitEquiv_comp** (g : GLoc σ J v) : splitEquiv h hJ (τ.comp σ) g = dualConj (splitGram hJ τ) (splitEquiv h hJ τ g)

theorem **splitEquiv_ratLoc** (γ : GRat σ J) : (splitEquiv h hJ τ (ratLoc σ J v γ)).1 = γ.1.1.map τ

theorem **splitEquivIn_apply** (β : GL (Fin n) (Fv F v)) (g : GLoc σ J v) : splitEquivIn h hJ τ β g = β⁻¹ ∗ splitHom σ J τ g ∗ β

theorem **splitHom_splitEquivIn_symm** (β x : GL (Fin n) (Fv F v)) : splitHom σ J τ ((splitEquivIn h hJ τ β).symm x) = β * x * β⁻¹

theorem **mem_splitLat** {u : Fin n → Fv F v} : u ∈ splitLat τ L ↔ ∃ y ∈ latLoc F L v, splitVec τ y = u

theorem **splitIncl_mem_latLoc_iff** (h : IsCMExt F E σ) {u : Fin n → Fv F v} : splitIncl h τ u ∈ latLoc F L v ↔ u ∈ splitLat τ L

theorem **mem_latLoc_iff_split** (h : IsCMExt F E σ) {y : Fin n → Fv F v ⊗[F] E} : y ∈ latLoc F L v ↔ splitVec τ y ∈ splitLat τ L ∧ splitVec (τ.comp σ) y ∈ splitLat (τ.comp σ) L

theorem **mem_splitLat_stdLattice** (h : IsCMExt F E σ) {u : Fin n → Fv F v} : u ∈ splitLat τ (stdLattice E n) ↔ ∀ i, u i ∈ v.adicCompletionIntegers F

theorem **mem_splitLat_comp_iff** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (hL : IsSelfDualAt σ J L v) {u' : Fin n → Fv F v} : u' ∈ splitLat (τ.comp σ) L ↔ ∀ u ∈ splitLat τ L, u' ⬝ᵥ J.map τ *ᵥ u ∈ v.adicCompletionIntegers F

theorem **splitHom_mulVec_mem** {g : GLoc σ J v} (hg : g ∈ stabLoc σ J L v) {u : Fin n → Fv F v} (hu : u ∈ splitLat τ L) : (splitHom σ J τ g).1 *ᵥ u ∈ splitLat τ L

theorem **splitHom_inv_mulVec** (g : GLoc σ J v) (u : Fin n → Fv F v) : (splitHom σ J τ g⁻¹).1 *ᵥ ((splitHom σ J τ g).1 *ᵥ u) = u

theorem **splitHom_mulVec_inv** (g : GLoc σ J v) (u : Fin n → Fv F v) : (splitHom σ J τ g).1 *ᵥ ((splitHom σ J τ g⁻¹).1 *ᵥ u) = u

theorem **splitHom_mem_stabGL** {g : GLoc σ J v} (hg : g ∈ stabLoc σ J L v) : splitHom σ J τ g ∈ stabGL (splitLat τ L)

theorem **mem_stabLoc_iff_splitLat** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (hL : IsSelfDualAt σ J L v) {g : GLoc σ J v} : g ∈ stabLoc σ J L v ↔ splitHom σ J τ g ∈ stabGL (splitLat τ L)

theorem **isSplitBasis_stdLattice** (h : IsCMExt F E σ) : IsSplitBasis τ (stdLattice E n) 1

theorem **IsLatticeFrame.isSplitBasis** (h : IsCMExt F E σ) {a : GL (Fin n) (AF F ⊗[F] E)} (ha : IsLatticeFrame L a) : IsSplitBasis τ L (bc (splitProj τ : Fv F v ⊗[F] E →+* Fv F v) (bc (compFin F E v) a))

theorem **exists_isSplitBasis** (h : IsCMExt F E σ) [Submodule.IsLattice E L] : ∃ β, IsSplitBasis τ L β

theorem **mem_GLint_iff_mulVec** {x : GL (Fin n) (Fv F v)} : x ∈ GLint (Fin n) (v.adicCompletionIntegers F).toSubring ↔ ∀ u : Fin n → Fv F v, (∀ i, u i ∈ v.adicCompletionIntegers F) ↔ ∀ i, (x.1 *ᵥ u) i ∈ v.adicCompletionIntegers F

theorem **IsSplitBasis.mem_iff** (hβ : IsSplitBasis τ L β) {u : Fin n → Fv F v} : u ∈ splitLat τ L ↔ ∀ i, ((β⁻¹).1 *ᵥ u) i ∈ v.adicCompletionIntegers F

theorem **IsSplitBasis.iff_mem_GLint** (hβ : IsSplitBasis τ L β) {β' : GL (Fin n) (Fv F v)} : IsSplitBasis τ L β' ↔ β⁻¹ * β' ∈ GLint (Fin n) (v.adicCompletionIntegers F).toSubring

theorem **IsSplitBasis.mem_stabGL_iff** (hβ : IsSplitBasis τ L β) {x : GL (Fin n) (Fv F v)} : x ∈ stabGL (splitLat τ L) ↔ β⁻¹ * x * β ∈ GLint (Fin n) (v.adicCompletionIntegers F).toSubring

theorem **mem_stabLoc_iff_GLint** (hL : IsSelfDualAt σ J L v) (hβ : IsSplitBasis τ L β) {g : GLoc σ J v} : g ∈ stabLoc σ J L v ↔ splitEquivIn h hJ τ β g ∈ GLint (Fin n) (v.adicCompletionIntegers F).toSubring

theorem **map_stabLoc** (hL : IsSelfDualAt σ J L v) (hβ : IsSplitBasis τ L β) : (stabLoc σ J L v).map (splitEquivIn h hJ τ β : GLoc σ J v →* GL (Fin n) (Fv F v)) = GLint (Fin n) (v.adicCompletionIntegers F).toSubring

theorem **mem_stabLocOne** {g : GLoc σ J v} : g ∈ stabLocOne σ J τ L ↔ g ∈ stabLoc σ J L v ∧ ∀ u ∈ splitLat τ L, (splitHom σ J τ g).1 *ᵥ u − u ∈ IsLocalRing.maximalIdeal (v.adicCompletionIntegers F) • splitLat τ L

theorem **IsSplitBasis.mem_smul_iff** (hβ : IsSplitBasis τ L β) {u : Fin n → Fv F v} : u ∈ IsLocalRing.maximalIdeal (v.adicCompletionIntegers F) • splitLat τ L ↔ ∀ i, Valued.v (((β⁻¹).1 *ᵥ u) i) < 1

theorem **IsSplitBasis.mem_stabLocOne_iff** (hβ : IsSplitBasis τ L β) {g : GLoc σ J v} : g ∈ stabLocOne σ J τ L ↔ g ∈ stabLoc σ J L v ∧ ∀ i j, Valued.v (((β⁻¹ * splitHom σ J τ g * β).1 − 1) i j) < 1

theorem **bc_residue_eq_one_iff** {y : GL (Fin n) (v.adicCompletionIntegers F)} : bc (IsLocalRing.residue (v.adicCompletionIntegers F)) y = 1 ↔ ∀ i j, Valued.v (((bc (v.adicCompletionIntegers F).subtype y).1 − 1) i j) < 1

theorem **mem_princ_iff** {x : GL (Fin n) (Fv F v)} : x ∈ princ (v.adicCompletionIntegers F).subtype (IsLocalRing.maximalIdeal _) ↔ x ∈ GLint (Fin n) (v.adicCompletionIntegers F).toSubring ∧ ∀ i j, Valued.v ((x.1 − 1) i j) < 1

theorem **bc_stabLocEquiv** (k : stabLoc σ J L v) : bc (v.adicCompletionIntegers F).subtype (stabLocEquiv h hJ hL hβ k) = splitEquivIn h hJ τ β k

theorem **coe_stabLocEquiv** (k : stabLoc σ J L v) (i j : Fin n) : ((stabLocEquiv h hJ hL hβ k).1 i j : Fv F v) = (splitEquivIn h hJ τ β k).1 i j

theorem **splitRed_val** (k : stabLoc σ J L v) : (splitRed h hJ hL hβ k).1 = (stabLocEquiv h hJ hL hβ k).1.map (IsLocalRing.residue _)

theorem **splitRed_surjective** : Function.Surjective (splitRed h hJ hL hβ)

theorem **splitRed_eq_one_iff** (k : stabLoc σ J L v) : splitRed h hJ hL hβ k = 1 ↔ k.1 ∈ stabLocOne σ J τ L

theorem **ker_splitRed** : (splitRed h hJ hL hβ).ker = (stabLocOne σ J τ L).subgroupOf (stabLoc σ J L v)

theorem **dualConj_one_val** {K : Type} [Field K] (x : GL (Fin n) K) : (dualConj 1 x).1 = (x⁻¹).1ᵀ

theorem **IsSplitBasis.dual** (hL : IsSelfDualAt σ J L v) (hβ : IsSplitBasis τ L β) : IsSplitBasis (τ.comp σ) L (dualConj 1 (splitGram hJ τ * β))

theorem **splitEquivIn_comp** (τ : E →ₐ[F] Fv F v) (β : GL (Fin n) (Fv F v)) (g : GLoc σ J v) : splitEquivIn h hJ (τ.comp σ) (dualConj 1 (splitGram hJ τ * β)) g = dualConj 1 (splitEquivIn h hJ τ β g)

theorem **conj_mem_stabLocOne** {k g : GLoc σ J v} (hk : k ∈ stabLoc σ J L v) (hg : g ∈ stabLocOne σ J τ L) : k * g * k⁻¹ ∈ stabLocOne σ J τ L

theorem **isOpen_stabLocOne** [Submodule.IsLattice E L] : IsOpen (stabLocOne σ J τ L : Set (GLoc σ J v))

theorem **GLint_eq_KO_and_princ_eq_KOne** (v : HeightOneSpectrum (𝒪 F)) (n : ℕ) : letI

theorem **chartEmb_apply** (hπ : Chart e v π) (x : D.host.E) : chartEmb hπ x = π (incA D x)

theorem **chart_eq_splitProj** (hπ : Chart e v π) (z : AE D) : π z = splitProj (chartEmb hπ) (compFin D.host.F D.host.E v z)

theorem **rho_eq_splitEquivIn** (hπ : Chart e v π) (V : HSp D) (hJ : IsHermND D.sp.cm.sig V.gram) (β : GL (Fin V.rank) (Fv D.host.F v)) (g : GA D V) : rho V π β g = splitEquivIn (isCMExt_host D) hJ (chartEmb hπ) β (proj D.sp.cm.sig V.gram v g)

theorem **compFin_eq_splitIdem** (he : IsSplitIdem D e) (hπ : Chart e v π) : compFin D.host.F D.host.E v e = splitIdem (isCMExt_host D) (chartEmb hπ)

theorem **exists_integer_conj_eq_neg** (h : IsCMExt F E σ) : ∃ a : 𝒪 E, a ≠ 0 ∧ σ a = −a

theorem **one_lt_infinitePlace** (h : IsCMExt F E σ) {a : E} (ha0 : a ≠ 0) (ha : σ a = -a) (w : InfinitePlace E) : 1 < w (1 + 2 * a)

theorem **not_isUnit_of_one_lt** {x : 𝒪 E} (hx : ∀ w : InfinitePlace E, 1 < w x) : ¬ IsUnit x

theorem **exists_split_place** (h : IsCMExt F E σ) : ∃ v : HeightOneSpectrum (𝒪 F), Nonempty (E →ₐ[F] Fv F v)

theorem **isHermND_one** (σ : E →ₐ[F] E) (n : ℕ) : IsHermND σ (1 : Matrix (Fin n) (Fin n) E)

theorem **exists_split_data** (h : IsCMExt F E σ) (n : ℕ) : ∃ (v : HeightOneSpectrum (𝒪 F)) (τ : E →ₐ[F] Fv F v), IsHermND σ (1 : Matrix (Fin n) (Fin n) E) ∧ IsSelfDualAt σ 1 (stdLattice E n) v ∧ IsSplitBasis τ (stdLattice E n) 1

Development.AbelianVarietyTheory.Definitions.Foundations.HahnPowOrderLeadingCoeff

theorem **hahn_pow_order_leadingCoeff** (s : HahnSeries ℚ Ω) (hs : s ≠ 0) (n : ℕ) : (s ^ n).order = n • s.order ∧ (s ^ n).leadingCoeff = s.leadingCoeff ^ n

private theorem **hahnSeries_algebraMap_mul_coeff** (c : Ω) (x : HahnSeries ℚ Ω) (r : ℚ) : (algebraMap Ω (HahnSeries ℚ Ω) c * x).coeff r = c * x.coeff r

theorem **hahn_transcendental** {s : HahnSeries ℚ Ω} (hs : s ≠ 0) (ho : s.order < 0) : Transcendental Ω s

theorem **mem_lev_iff** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (f : X.K) : f ∈ lev X Γ' ↔ ∀ γ ∈ Γ', X.sl (ofSL γ) f = f

theorem **algebraMap_mem_lev** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (c : Ω) : algebraMap Ω X.K c ∈ lev X Γ'

theorem **jK_mem_lev** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) : X.jK ∈ lev X Γ'

theorem **mdl_le_lev** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (S : Subgroup (Ω ≃ₐ[ℚ] Ω)) : mdl X Γ' S ≤ lev X Γ'

theorem **jK_transcendental** (X : Tower Ω ε) : Transcendental Ω X.jK

theorem **coe_jPoly** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (P : Polynomial Ω) : ((jPoly X Γ' P : lev X Γ') : X.K) = Polynomial.aeval X.jK P

theorem **jPoly_injective** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) : Function.Injective (jPoly X Γ')

theorem **jRat_algebraMap** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (P : Polynomial Ω) : jRat X Γ' (algebraMap (Polynomial Ω) (RatFunc Ω) P) = jPoly X Γ' P

theorem **exp_injective** (X : Tower Ω ε) : Function.Injective X.exp

theorem **exp_algebraMap_coeff** (X : Tower Ω ε) (c : Ω) (r : ℚ) : (X.exp (algebraMap Ω X.K c)).coeff r = if r = 0 then c else 0

theorem **gal_algebraMap** (X : Tower Ω ε) (s : Ω ≃ₐ[ℚ] Ω) (c : Ω) : X.gal s (algebraMap Ω X.K c) = algebraMap Ω X.K (s c)

theorem **gal_jK** (X : Tower Ω ε) (s : Ω ≃ₐ[ℚ] Ω) : X.gal s X.jK = X.jK

theorem **mem_mdl_iff** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (S : Subgroup (Ω ≃ₐ[ℚ] Ω)) (f : X.K) : f ∈ mdl X Γ' S ↔ f ∈ lev X Γ' ∧ ∀ s ∈ S, X.gal s f = f

theorem **algebraMap_mem_mdl** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (S : Subgroup (Ω ≃ₐ[ℚ] Ω)) (c : Ω) (hc : ∀ s ∈ S, s c = c) : algebraMap Ω X.K c ∈ mdl X Γ' S

theorem **jK_mem_mdl** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (S : Subgroup (Ω ≃ₐ[ℚ] Ω)) : X.jK ∈ mdl X Γ' S

theorem **coe_jRat_X** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) : ((jRat X Γ' RatFunc.X : lev X Γ') : X.K) = X.jK

theorem **isScalarTower_poly_rat** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) : IsScalarTower (Polynomial Ω) (RatFunc Ω) (lev X Γ')

theorem **isScalarTower_mdl** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (S : Subgroup (Ω ≃ₐ[ℚ] Ω)) : IsScalarTower (mdl X Γ' S) (lev X Γ') X.K

Development.AbelianVarietyTheory.Definitions.Foundations.HarmonicAllOfCompact

theorem **isFockIsometry_fockEmb** (σ : Type) [DecidableEq σ] : IsFockIP (fockEmb σ)

theorem **not_isFockIP_zero** (σ : Type) [DecidableEq σ] : ¬ IsFockIP (0 : MvPolynomial σ ℂ →ₗ[ℂ] ℂ)

theorem **minusPart_of_deg** {σ : Type} (A : Module.End ℂ (MvPolynomial σ ℂ)) (P : MvPolynomial σ ℂ) (h : P.totalDegree ≤ 1) : minusPart A P = 0

theorem **mem_jointHarm_of_deg** {p q r s : ℕ} (P : Pol p q r s) (h : P.totalDegree ≤ 1) : P ∈ jointHarm p q r s

theorem **harmonic_all_of_compact** (p q r : ℕ) : harm1 p q r 0 = Set.univ

theorem **not_mem_harm2** : (X (.inl 0, .inl 0) * X (.inr 0, .inl 0) : Pol 1 1 1 0) ∉ harm2 1 1 1 0

theorem **subset_genSub** {p q : ℕ} {V : Type} [AddCommGroup V] [Module ℂ V] (act : Mat p q →ₗ[ℂ] Module.End ℂ V) (S : Set V) : S ⊆ genSub act S

theorem **genSub_stable** {p q : ℕ} {V : Type} [AddCommGroup V] [Module ℂ V] (act : Mat p q →ₗ[ℂ] Module.End ℂ V) (S : Set V) (Z : Mat p q) (y : V) (hy : y ∈ genSub act S) : act Z y ∈ genSub act S

theorem **charAct_apply** (p q : ℕ) (c : ℚ) (Z : Mat p q) (v : ℂ) : charAct p q c Z v = (c : ℂ) * Z.trace * v

theorem **mem_wt** (p q : ℕ) (c : ℚ) (μ : Ix p q → ℚ) (v : ℂ) : v ∈ wtSub (charAct p q c) μ ↔ ∀ a, (c : ℂ) * v = μ a * v

theorem **coneTypes_rank_one** (c d : ℚ) : ConeTypes (charMod 1 0 c) (fun _ => d) ↔ c = d

theorem **blattner_lamHalf** : blattner lamHalf (.inl 0) = 1

theorem **not_coneTypes_trivial** : ¬ ConeTypes (charMod 1 1 0) lamHalf

theorem **mat_eq** (Z : Mat 1 0) : Z = Z (.inl 0) (.inl 0) • Eab (.inl 0) (.inl 0)

theorem **vac_monomial** (α : Ln 1 0 1 0 →₀ ℕ) : vac (monomial α 1) = (if α = 0 then (1 : ℂ) else 0) • ((1 : ℂ) ⊗ₜ[ℂ] (1 : ℂ) : TensorProduct ℂ ℂ ℂ)

theorem **intw_vac** : Intw (charMod 1 0 (1 / 2)) (charMod 1 0 (1 / 2)) vac ∧ vac ≠ 0

Development.AbelianVarietyTheory.Definitions.Foundations.HermitianSpaceBaseLatticeEqFrameLattice

theorem **tmul_mem_intLoc** {v : HeightOneSpectrum (𝒪 F)} {a : Fv F v} (ha : a ∈ v.adicCompletionIntegers F) (x : 𝒪 E) : a ⊗ₜ[F] (x : E) ∈ intLoc F E v

theorem **intLoc_induction** {v : HeightOneSpectrum (𝒪 F)} {p : Fv F v ⊗[F] E → Prop} (h1 : ∀ a ∈ v.adicCompletionIntegers F, ∀ x : 𝒪 E, p (a ⊗ₜ[F] (x : E))) (h0 : p 0) (hadd : ∀ y z, p y → p z → p (y + z)) (hneg : ∀ y, p y → p (-y)) {y : Fv F v ⊗[F] E} (hy : y ∈ intLoc F E v) : p y

theorem **absFin_le_one_of_mem_intLoc** {v : HeightOneSpectrum (𝒪 F)} {y : Fv F v ⊗[F] E} (hy : y ∈ intLoc F E v) : absFin F v y ≤ 1

theorem **sigR_mem_intLoc** (σ : E →ₐ[F] E) {v : HeightOneSpectrum (𝒪 F)} {y : Fv F v ⊗[F] E} (hy : y ∈ intLoc F E v) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ y ∈ intLoc F E v

theorem **tmul_mem_latLoc** {L : Submodule (𝒪 E) (Fin n → E)} {v : HeightOneSpectrum (𝒪 F)} {a : Fv F v} (ha : a ∈ v.adicCompletionIntegers F) {x : Fin n → E} (hx : x ∈ L) : (fun i => a ⊗ₜ[F] x i) ∈ latLoc F L v

theorem **latLoc_induction** {L : Submodule (𝒪 E) (Fin n → E)} {v : HeightOneSpectrum (𝒪 F)} {p : (Fin n → Fv F v ⊗[F] E) → Prop} (h1 : ∀ a ∈ v.adicCompletionIntegers F, ∀ x ∈ L, p (fun i => a ⊗ₜ[F] x i)) (h0 : p 0) (hadd : ∀ y z, p y → p z → p (y + z)) (hneg : ∀ y, p y → p (-y)) {y : Fin n → Fv F v ⊗[F] E} (hy : y ∈ latLoc F L v) : p y

theorem **mem_latLoc_stdLattice** {v : HeightOneSpectrum (𝒪 F)} {y : Fin n → Fv F v ⊗[F] E} : y ∈ latLoc F (stdLattice E n) v ↔ ∀ i, y i ∈ intLoc F E v

theorem **mem_latHat_iff** {L : Submodule (𝒪 E) (Fin n → E)} (hL : L.FG) (y : Fin n → Af F ⊗[F] E) : y ∈ Formalism.MainNotions.StandardLattice.latHat F E (L : Set (Fin n → E)) ↔ ∀ v, vecLoc v y ∈ latLoc F L v

theorem **hLin_add_left** (R : Type) [CommRing R] [Algebra F R] (y y' x : Fin n → R ⊗[F] E) : hLin σ J R (y + y') x = hLin σ J R y x + hLin σ J R y' x

theorem **hLin_smul_left** (R : Type) [CommRing R] [Algebra F R] (c : R ⊗[F] E) (y x : Fin n → R ⊗[F] E) : hLin σ J R (c • y) x = c * hLin σ J R y x

theorem **isSelfDualAt_stdLattice_one** (v : HeightOneSpectrum (𝒪 F)) : IsSelfDualAt σ (1 : Matrix (Fin n) (Fin n) E) (stdLattice E n) v

theorem **mem_stabLoc** {v : HeightOneSpectrum (𝒪 F)} {g : GLoc σ J v} : g ∈ stabLoc σ J L v ↔ ∀ y, y ∈ latLoc F L v ↔ (matHom _ σ J g).1 *ᵥ y ∈ latLoc F L v

theorem **mem_levelOpen** {g : GAd σ J} : g ∈ levelOpen σ J L ↔ ∀ v, proj σ J v g ∈ stabLoc σ J L v

theorem **mem_levelOff** {g : GAd σ J} : g ∈ levelOff σ J L S ↔ g ∈ finPts σ J ∧ (∀ v ∈ S, proj σ J v g = 1) ∧ ∀ v ∉ S, proj σ J v g ∈ stabLoc σ J L v

theorem **levelOff_eq** : levelOff σ J L S = finAwayPts σ J S ⊓ levelOpen σ J L

theorem **mem_levelFin** {g : GAd σ J} : g ∈ levelFin σ J L ↔ g ∈ finPts σ J ∧ ∀ v, proj σ J v g ∈ stabLoc σ J L v

theorem **mem_levelOffFin** {g : GFin σ J} : g ∈ levelOffFin σ J L S ↔ (∀ v ∈ S, projFinLoc σ J v g = 1) ∧ ∀ v ∉ S, projFinLoc σ J v g ∈ stabLoc σ J L v

theorem **mem_stabArch** {w : InfinitePlace F} {b : GL (Fin n) (w.Completion ⊗[F] E)} {g : GArch σ J w} : g ∈ stabArch σ J w b ↔ b⁻¹ * matHom _ σ J g * b ∈ UJ (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ) 1

theorem **mem_levelAll** {a : GL (Fin n) (AF F ⊗[F] E)} {g : GAd σ J} : g ∈ levelAll σ J L a ↔ (∀ v, proj σ J v g ∈ stabLoc σ J L v) ∧ ∀ w, projArch σ J w g ∈ stabArch σ J w (bc (compInf F E w) a)

theorem **emb_mem_levelOff** {v : HeightOneSpectrum (𝒪 F)} (hv : v ∉ S) {k : GLoc σ J v} (hk : k ∈ stabLoc σ J L v) : emb σ J v k ∈ levelOff σ J L S

theorem **isLatticeFrame_one** : IsLatticeFrame (stdLattice E n) (1 : GL (Fin n) (AF F ⊗[F] E))

theorem **isMajorantFrame_one** : IsMajorantFrame σ (1 : Matrix (Fin n) (Fin n) E) 1

theorem **vecLoc_mulVec** (v : HeightOneSpectrum (𝒪 F)) (g : GAd σ J) (y : Fin n → Af F ⊗[F] E) : vecLoc v ((Formalism.MainNotions.StandardLattice.finMat σ J g).mulVec y) = (matHom _ σ J (proj σ J v g)).1 *ᵥ vecLoc v y

theorem **mem_levelOpen_iff_latHat** {L : Submodule (𝒪 E) (Fin n → E)} (hL : L.FG) (g : GAd σ J) : g ∈ levelOpen σ J L ↔ ∀ y, y ∈ Formalism.MainNotions.StandardLattice.latHat F E (L : Set (Fin n → E)) ↔ (Formalism.MainNotions.StandardLattice.finMat σ J g).mulVec y ∈ Formalism.MainNotions.StandardLattice.latHat F E (L : Set (Fin n → E))

theorem **height_eq** (g : GAd σ J) : height σ J g = (∏ w, htInf F w (matHom _ σ J (projArch σ J w g))) * ∏ᶠ v, htFin F v (matHom _ σ J (proj σ J v g))

theorem **one_le_height** (g : GAd σ J) : 1 ≤ height σ J g

@[simp] theorem **height_one** : height σ J 1 = 1

theorem **height_inv** (g : GAd σ J) : height σ J g⁻¹ = height σ J g

theorem **height_mul_le** (g h : GAd σ J) : height σ J (g * h) ≤ height σ J g * height σ J h

theorem **htInf_le_height** (w : InfinitePlace F) (g : GAd σ J) : htInf F w (matHom _ σ J (projArch σ J w g)) ≤ height σ J g

theorem **htFin_one** (v : HeightOneSpectrum (𝒪 F)) : htFin F v (1 : GL (Fin n) (Fv F v ⊗[F] E)) = 1

theorem **htInf_one** (w : InfinitePlace F) : htInf F w (1 : GL (Fin n) (w.Completion ⊗[F] E)) = 1

theorem **height_emb** (v : HeightOneSpectrum (𝒪 F)) (x : GLoc σ J v) : height σ J (emb σ J v x) = htFin F v (matHom _ σ J x)

theorem **height_embArch** (w : InfinitePlace F) (x : GArch σ J w) : height σ J (embArch σ J w x) = htInf F w (matHom _ σ J x)

theorem **conj_mul** (a : GL (Fin n) (AF F ⊗[F] E)) (g h : GAd σ J) : a⁻¹ * matHom _ σ J (g * h) * a = (a⁻¹ * matHom _ σ J g * a) * (a⁻¹ * matHom _ σ J h * a)

theorem **normInf_le_one_of_unitary** (h : IsCMExt F E σ) {w : InfinitePlace F} {u : GL (Fin n) (w.Completion ⊗[F] E)} (hu : u ∈ UJ (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ) 1) : normInf F w u.1 ≤ 1

theorem **normFin_conj_le_one** {L : Submodule (𝒪 E) (Fin n → E)} {a : GL (Fin n) (AF F ⊗[F] E)} (ha : IsLatticeFrame L a) {v : HeightOneSpectrum (𝒪 F)} {k : GLoc σ J v} (hk : k ∈ stabLoc σ J L v) : normFin F v ((bc (compFin F E v) a)⁻¹ * matHom _ σ J k * bc (compFin F E v) a).1 ≤ 1

theorem **IsModerate.exists_nonneg** {G X : Type*} [Norm X] {ht : G → ℝ≥0} {f : G → X} (h : IsModerate ht f) : ∃ (C : ℝ) (N : ℕ), 0 ≤ C ∧ ∀ g, ‖f g‖ ≤ C * (ht g : ℝ) ^ N

theorem **IsModerate.comp** {G G' X : Type*} [Norm X] {ht : G → ℝ≥0} {ht' : G' → ℝ≥0} {f : G → X} (h : IsModerate ht f) (φ : G' → G) {c : ℝ≥0} {m : ℕ} (hφ : ∀ g, ht (φ g) ≤ c * ht' g ^ m) : IsModerate ht' (f ∘ φ)

theorem **IsModerate.add** {G X : Type*} [SeminormedAddGroup X] {ht : G → ℝ≥0} (h1 : ∀ g, 1 ≤ ht g) {f f' : G → X} (h : IsModerate ht f) (h' : IsModerate ht f') : IsModerate ht (f + f')

theorem **IsModerate.smul** {G R X : Type*} [SeminormedAddGroup R] [SeminormedAddGroup X] [SMul R X] [NormSMulClass R X] {ht : G → ℝ≥0} {f : G → R} {f' : G → X} (h : IsModerate ht f) (h' : IsModerate ht f') : IsModerate ht (f • f')

theorem **ModerateGrowth.mul_right** {X : Type*} [Norm X] {f : GAd σ J → X} (h : ModerateGrowth σ J f) (x : GAd σ J) : ModerateGrowth σ J fun g => f (g * x)

theorem **ModerateGrowth.mul_left** {X : Type*} [Norm X] {f : GAd σ J → X} (h : ModerateGrowth σ J f) (x : GAd σ J) : ModerateGrowth σ J fun g => f (x * g)

theorem **ModerateGrowth.add** {X : Type*} [SeminormedAddGroup X] {f f' : GAd σ J → X} (h : ModerateGrowth σ J f) (h' : ModerateGrowth σ J f') : ModerateGrowth σ J (f + f')

theorem **mem_frameLat** {E : Type} [Field E] [NumberField E] [StarRing E] (V : HSpace E) (x : Fin V.rank → E) : x ∈ frameLat V ↔ ∑ i, x i • V.frame i ∈ V.L0

theorem **frameLat_fg** {E : Type} [Field E] [NumberField E] [StarRing E] (V : HSpace E) : (frameLat V).FG

theorem **isCMExt_host** (D : D7) : IsCMExt D.host.F D.host.E D.sp.cm.sig

theorem **GQ_eq** : GQ V = ratPts D.sp.cm.sig V.gram

theorem **GfC_eq** : GfC V = finPts D.sp.cm.sig V.gram

theorem **GinfC_eq** : GinfC V = infPts D.sp.cm.sig V.gram

theorem **hermitianSpaceBaseLattice_eq_frameLattice** : Formalism.MainNotions.StandardLattice.L0c D V = (frameLat V : Set _)

theorem **KsphC_eq** (S : Finset (HeightOneSpectrum (𝒪 D.host.F))) : KsphC V S = (levelOff D.sp.cm.sig V.gram (frameLat V) S : Set _)

theorem **isCMExt_of_isCMField** (E : Type) [Field E] [NumberField E] [IsCMField E] : IsCMExt (maximalRealSubfield E) E (IsCMField.complexConj E : E →ₐ[maximalRealSubfield E] E)

Development.AbelianVarietyTheory.Definitions.Foundations.HtCoLOfEq

theorem **htCoL_of_eq** (A : ValuationSubring Ω) (c : Coef l) (j : Ω →+* Qbar l) (Z : ∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l) (m c₀ : ℕ) (y : (ZMod (l ^ m))ˣ) (h : Qbar l) (hZ : ∀ s, Z m s y = h * lamv l m s) : HTCoL A c j Z m c₀ y h

Development.AbelianVarietyTheory.Definitions.Foundations.InnerEqZeroOfNotUEquiv

theorem **mem_homIn** : T ∈ homIn π ρ U ↔ (∀ x, T x ∈ U) ∧ ∀ g x, T (π g x) = ρ g (T x)

theorem **mem_homIn_top** : T ∈ homIn π ρ ⊤ ↔ ∀ g x, T (π g x) = ρ g (T x)

theorem **homIn_mono** (h : U ≤ U') : homIn π ρ U ≤ homIn π ρ U'

theorem **multIn_mono** (h : U ≤ U') : multIn π ρ U ≤ multIn π ρ U'

theorem **multIn_eq_zero_iff** : multIn π ρ U = 0 ↔ homIn π ρ U = ⊥

theorem **multIn_lt_aleph0_iff** : multIn π ρ U < Cardinal.aleph0 ↔ FiniteDimensional ℂ (homIn π ρ U)

theorem **multIn_eq_rank** (hU : IsSubrep ρ U) : multIn π ρ U = Module.rank ℂ (ContIntertwiningMap (.ofMonoidHom π : ContRepresentation ℂ G H) (.ofMonoidHom hU.rep : ContRepresentation ℂ G U))

theorem **multIn_rep** (hU : IsSubrep ρ U) : multIn π hU.rep ⊤ = multIn π ρ U

theorem **UEquiv.refl** (π : G →* (H →L[ℂ] H)) : UEquiv π π

theorem **UEquiv.symm** (h : UEquiv π ρ) : UEquiv ρ π

theorem **UEquiv.trans** (h : UEquiv π ρ) (h' : UEquiv ρ ρ') : UEquiv π ρ'

theorem **UEquiv.comp** (h : UEquiv π ρ) (j : G' →* G) : UEquiv (π.comp j) (ρ.comp j)

theorem **UEquiv.multIn_eq** (h : UEquiv π π') : multIn π ρ U = multIn π' ρ U

theorem **homIn_comp** {j : G' →* G} (hj : Function.Surjective j) : homIn (π.comp j) (ρ.comp j) U = homIn π ρ U

theorem **isSubrep_comp** {j : G' →* G} (hj : Function.Surjective j) : IsSubrep (ρ.comp j) U ↔ IsSubrep ρ U

theorem **UEquiv.isTopIrreducible** (h : UEquiv π ρ) (hπ : IsTopIrreducible π) : IsTopIrreducible ρ

theorem **IsCopy.isIrredSubrep** (h : IsCopy π ρ W) (hπ : IsTopIrreducible π) : IsIrredSubrep ρ W

theorem **isCopy_iff** [CompleteSpace H] : IsCopy π ρ W ↔ ∃ e : H →ₗᵢ[ℂ] K, (∀ g x, e (π g x) = ρ g (e x)) ∧ LinearMap.range e.toLinearMap = W

theorem **apply_mem_isoPart** (hT : T ∈ homIn π ρ U) (x : H) : T x ∈ isoPart π ρ U

theorem **isoPart_le** (hW : IsClosed (W : Set K)) (h : ∀ T ∈ homIn π ρ U, ∀ x, T x ∈ W) : isoPart π ρ U ≤ W

theorem **isoPart_le_self** (hU : IsClosed (U : Set K)) : isoPart π ρ U ≤ U

theorem **isSubrep_isoPart** : IsSubrep ρ (isoPart π ρ U)

theorem **IsIsotypic.refl** (π : G →* (H →L[ℂ] H)) : IsIsotypic π π

theorem **IsIsotypic.homIn_ne_bot** [Nontrivial K] (h : IsIsotypic π ρ) : homIn π ρ ⊤ ≠ ⊥

theorem **norm_unitVec** [Nontrivial H] : ‖unitVec H‖ = 1

theorem **be_def** (T T' : H →L[ℂ] K) : be T T' = ⟪T' (unitVec H), T (unitVec H)⟫_ℂ

theorem **be_conj_symm** (T T' : H →L[ℂ] K) : conj (be T' T) = be T T'

theorem **be_self** (T : H →L[ℂ] K) : be T T = (‖T (unitVec H)‖ : ℂ) ^ 2

theorem **inner_map_map** (hρ : IsUnitaryRep ρ) (g : G) (x y : K) : ⟪ρ g x, ρ g y⟫_ℂ = ⟪x, y⟫_ℂ

theorem **isUnitaryRep_comp** [TopologicalSpace G'] (hρ : IsUnitaryRep ρ) {j : G' →* G} (hj : Continuous j) : IsUnitaryRep (ρ.comp j)

theorem **IsIrrUnitary.comp** [TopologicalSpace G'] (hπ : IsIrrUnitary π) {j : G' →* G} (hc : Continuous j) (hj : Function.Surjective j) : IsIrrUnitary (π.comp j)

theorem **isIrrUnitary_rep** [CompleteSpace K] (hρ : IsUnitaryRep ρ) (hW : IsIrredSubrep ρ W) : IsIrrUnitary hW.1.rep

theorem **isIrrUnitary_charRep** {χ : G →* Circle} (hχ : Continuous χ) : IsIrrUnitary (charRep χ)

theorem **inner_eq_be_of_norm** [CompleteSpace H] (h1 : ∀ g x, ‖π g x‖ = ‖x‖) (h2 : IsTopIrreducible π) (h3 : ∀ g x, ‖ρ g x‖ = ‖x‖) (hT : T ∈ homIn π ρ U) (hT' : T' ∈ homIn π ρ U') (x y : H) : ⟪T' y, T x⟫_ℂ = be T T' * ⟪y, x⟫_ℂ

theorem **inner_eq_be** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) (hT : T ∈ homIn π ρ U) (hT' : T' ∈ homIn π ρ U') (x y : H) : ⟪T' y, T x⟫_ℂ = be T T' * ⟪y, x⟫_ℂ

theorem **inner_apply_eq_be** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) (hT : T ∈ homIn π ρ U) (hT' : T' ∈ homIn π ρ U') (g : G) (x y : H) : ⟪T' y, ρ g (T x)⟫_ℂ = be T T' * ⟪y, π g x⟫_ℂ

theorem **norm_apply_eq** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) (hT : T ∈ homIn π ρ U) (x : H) : ‖T x‖ = ‖T (unitVec H)‖ * ‖x‖

theorem **be_self_pos** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) (hT : T ∈ homIn π ρ U) (h0 : T ≠ 0) : 0 < (be T T).re

theorem **isCopy_range** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) (hT : T ∈ homIn π ρ U) (h0 : T ≠ 0) : IsCopy π ρ (LinearMap.range (T : H →ₗ[ℂ] K))

theorem **multIn_ne_zero_iff** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) : multIn π ρ U ≠ 0 ↔ ∃ W ≤ U, IsCopy π ρ W

theorem **homIn_le_iff** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) : homIn π ρ U ≤ homIn π ρ U' ↔ ∀ W ≤ U, IsCopy π ρ W → W ≤ U'

theorem **homIn_disc** (hπ : IsIrrUnitary π) (hρ : IsUnitaryRep ρ) : homIn π ρ (disc ρ) = homIn π ρ ⊤

theorem **uEquiv_of_multIn_ne_zero** (hπ : IsIrrUnitary π) (hρ : IsIrrUnitary ρ) (h : multIn π ρ ⊤ ≠ 0) : UEquiv π ρ

theorem **multIn_self** (hπ : IsIrrUnitary π) : multIn π π ⊤ = 1

theorem **inner_eq_zero_of_not_uEquiv** (hπ : IsIrrUnitary π) (hπ' : IsIrrUnitary π') (hρ : IsUnitaryRep ρ) (h : ¬ UEquiv π π') (hT : T ∈ homIn π ρ U) {S : H' →L[ℂ] K} (hS : S ∈ homIn π' ρ U') (x : H) (y : H') : ⟪S y, T x⟫_ℂ = 0

theorem **IsIsotypic.uEquiv** [Nontrivial K] (hπ : IsIrrUnitary π) (hπ' : IsIrrUnitary π') (hρ : IsUnitaryRep ρ) (h : IsIsotypic π ρ) (h' : IsIsotypic π' ρ) : UEquiv π π'

theorem **IrrU.cl_eq_iff** {R R' : IrrU G} : R.cl = R'.cl ↔ UEquiv R.π R'.π

@[simp] theorem **UDual.cl_rep** (c : UDual G) : c.rep.cl = c

theorem **IrrU.multIn_cl** (R : IrrU G) (ρ : G →* (K →L[ℂ] K)) (U : Submodule ℂ K) : R.cl.multIn ρ U = multIn R.π ρ U

theorem **IrrU.cl_ofChar_inj** {χ ψ : G →* Circle} (hχ : Continuous χ) (hψ : Continuous ψ) : (IrrU.ofChar χ hχ).cl = (IrrU.ofChar ψ hψ).cl ↔ χ = ψ

@[simp] theorem **toRep_apply** (π : G →* (H →L[ℂ] H)) (g : G) (x : H) : toRep π g x = π g x

theorem **contains_iff** : Contains τ ρ ↔ ∃ f : X →ₗ[ℂ] W, f ≠ 0 ∧ ∀ g x, f (τ g x) = ρ g (f x)

theorem **Contains.of_injective** (h : Contains τ σ) (f : σ.IntertwiningMap ρ) (hf : Function.Injective f) : Contains τ ρ

theorem **contains_trivial_iff** (K : Subgroup G) : Contains (Representation.trivial ℂ K ℂ) (ρ.comp K.subtype) ↔ LGrp.IsSpherical ρ K

theorem **IsIsotypicAlg.refl** (σ : Representation ℂ G V) : IsIsotypicAlg σ σ

theorem **IsIsotypicAlg.exists_injective** [Nontrivial W] (h : IsIsotypicAlg σ ρ) (hσ : σ.IsIrreducible) : ∃ f : σ.IntertwiningMap ρ, Function.Injective f

theorem **IsIsotypicAlg.contains** [Nontrivial W] (h : IsIsotypicAlg σ ρ) (hσ : σ.IsIrreducible) {j : G' →* G} {τ : Representation ℂ G' X} (ht : Contains τ (σ.comp j)) : Contains τ (ρ.comp j)

Development.AbelianVarietyTheory.Definitions.Foundations.IsFirstChernClassOfLineBundle

theorem **kum_choose_spec** {N : ℕ} {ζ θ : Ω} {g : Ω ≃ₐ[K] Ω} (h : ∃ i : ZMod N, g θ = ζ ^ i.val * θ) : g θ = ζ ^ (kum (N

theorem **kum_eq_of** {N : ℕ} [NeZero N] {ζ θ : Ω} (hζ : IsPrimitiveRoot ζ N) (hθ : θ ≠ 0) {g : Ω ≃ₐ[K] Ω} (i : ZMod N) (hi : g θ = ζ ^ i.val * θ) : kum (N

theorem **kum_one** {N : ℕ} [NeZero N] {ζ θ : Ω} (hζ : IsPrimitiveRoot ζ N) (hθ : θ ≠ 0) : kum (N

theorem **inert_one** (B : ValuationSubring Ω) : Inert B (1 : Ω ≃ₐ[K] Ω)

theorem **mem_RY_of_inert** (R : Subring K) {B : ValuationSubring Ω} (hB : Over R B) {g : Ω ≃ₐ[K] Ω} (hg : Inert B g) : g ∈ RY R

theorem **over_top** (R : Subring K) : Over R (⊤ : ValuationSubring Ω)

Development.AbelianVarietyTheory.Definitions.Foundations.IsFockIPOfFockData

theorem **J_eq** {p q r s : ℕ} {T0 : GrpData (p * r + q * s) (p * s + q * r)} {T1 : GrpData p q} {T2 : GrpData r s} (F : FockData T0 T1 T2) : F.J = jointHarm p q r s

theorem **isIntw_iff** {p q r s : ℕ} {T0 : GrpData (p * r + q * s) (p * s + q * r)} {T1 : GrpData p q} {T2 : GrpData r s} (F : FockData T0 T1 T2) (M1 : GKMod p q) (M2 : GKMod r s) (f : Pol p q r s →ₗ[ℂ] TensorProduct ℂ M1.V M2.V) : IsIntw F M1 M2 f ↔ Intw M1 M2 f

theorem **isFockIsometry_of_fockData** {p q r s : ℕ} {T0 : GrpData (p * r + q * s) (p * s + q * r)} {T1 : GrpData p q} {T2 : GrpData r s} (F : FockData T0 T1 T2) : IsFockIP F.ι

Development.AbelianVarietyTheory.Definitions.Foundations.IsHilbertSchmidtOfFiniteDimensional

theorem **hasSum_norm_inner_sq** (e : HilbertBasis ι ℂ H) (x : H) : HasSum (fun i => ‖⟪e i, x⟫_ℂ‖ ^ 2) (‖x‖ ^ 2)

theorem **summable_matrix** (e : HilbertBasis ι ℂ H) (f : HilbertBasis κ ℂ K) {T : H →L[ℂ] K} (h : Summable fun i => ‖T (e i)‖ ^ 2) : Summable fun p : ι × κ => ‖⟪f p.2, T (e p.1)⟫_ℂ‖ ^ 2

theorem **hasSum_adjoint** (e : HilbertBasis ι ℂ H) (f : HilbertBasis κ ℂ K) {T : H →L[ℂ] K} {s : ℝ} (h : HasSum (fun i => ‖T (e i)‖ ^ 2) s) : HasSum (fun j => ‖adjoint T (f j)‖ ^ 2) s

theorem **hasSum_basis** (e : HilbertBasis ι ℂ H) (e' : HilbertBasis κ ℂ H) {T : H →L[ℂ] K} {s : ℝ} (h : HasSum (fun i => ‖T (e i)‖ ^ 2) s) : HasSum (fun j => ‖T (e' j)‖ ^ 2) s

theorem **isHilbertSchmidt_iff** (e : HilbertBasis ι ℂ H) {T : H →L[ℂ] K} : IsHilbertSchmidt T ↔ Summable fun i => ‖T (e i)‖ ^ 2

theorem **IsHilbertSchmidt.hasSum** {T : H →L[ℂ] K} (h : IsHilbertSchmidt T) (e : HilbertBasis ι ℂ H) : HasSum (fun i => ‖T (e i)‖ ^ 2) (hsNormSq T)

theorem **IsHilbertSchmidt.adjoint** {T : H →L[ℂ] K} (h : IsHilbertSchmidt T) : IsHilbertSchmidt (adjoint T)

theorem **IsHilbertSchmidt.of_norm_le** {T : H →L[ℂ] K} {S : H →L[ℂ] L} (h : IsHilbertSchmidt T) (c : ℝ) (hS : ∀ x, ‖S x‖ ≤ c * ‖T x‖) : IsHilbertSchmidt S

theorem **IsHilbertSchmidt.comp_left** {T : H →L[ℂ] K} (h : IsHilbertSchmidt T) (A : K →L[ℂ] L) : IsHilbertSchmidt (A ∘L T)

theorem **IsHilbertSchmidt.comp_right** {T : H →L[ℂ] K} (h : IsHilbertSchmidt T) (A : L →L[ℂ] H) : IsHilbertSchmidt (T ∘L A)

theorem **isHilbertSchmidt_zero** : IsHilbertSchmidt (0 : H →L[ℂ] K)

theorem **IsHilbertSchmidt.add** {S T : H →L[ℂ] K} (hS : IsHilbertSchmidt S) (hT : IsHilbertSchmidt T) : IsHilbertSchmidt (S + T)

theorem **IsHilbertSchmidt.smul** {T : H →L[ℂ] K} (h : IsHilbertSchmidt T) (c : ℂ) : IsHilbertSchmidt (c • T)

theorem **isHilbertSchmidt_of_finiteDimensional** [FiniteDimensional ℂ H] (T : H →L[ℂ] K) : IsHilbertSchmidt T

theorem **exists_hasSum_comp** (e : HilbertBasis ι ℂ H) (f : HilbertBasis κ ℂ K) {B : K →L[ℂ] H} {C : H →L[ℂ] K} (hB : IsHilbertSchmidt B) (hC : IsHilbertSchmidt C) : ∃ s : ℂ, HasSum (fun i => ⟪e i, B (C (e i))⟫_ℂ) s ∧ HasSum (fun j => ⟪f j, C (B (f j))⟫_ℂ) s

theorem **hasSum_trace_comp** (e : HilbertBasis ι ℂ H) {B : K →L[ℂ] H} {C : H →L[ℂ] K} (hB : IsHilbertSchmidt B) (hC : IsHilbertSchmidt C) : HasSum (fun i => ⟪e i, (B ∘L C) (e i)⟫_ℂ) (trace (B ∘L C))

theorem **trace_comp_comm** {B : K →L[ℂ] H} {C : H →L[ℂ] K} (hB : IsHilbertSchmidt B) (hC : IsHilbertSchmidt C) : trace (B ∘L C) = trace (C ∘L B)

theorem **IsTraceClass.hasSum_trace** {T : H →L[ℂ] H} (h : IsTraceClass T) (e : HilbertBasis ι ℂ H) : HasSum (fun i => ⟪e i, T (e i)⟫_ℂ) (trace T)

theorem **IsTraceClass.summable_norm** {T : H →L[ℂ] H} (h : IsTraceClass T) (e : HilbertBasis ι ℂ H) : Summable fun i => ‖⟪e i, T (e i)⟫_ℂ‖

theorem **exists_comp_eq_of_norm_le** {C : H →L[ℂ] K} {A : H →L[ℂ] L} (h : ∀ x, ‖C x‖ ≤ ‖A x‖) [CompleteSpace L] : ∃ U : L →L[ℂ] K, U ∘L A = C

theorem **norm_sqrt_apply** (C : H →L[ℂ] K) (x : H) : ‖(CFC.sqrt (adjoint C ∘L C) : H →L[ℂ] H) x‖ = ‖C x‖

theorem **IsHilbertSchmidt.isTraceClass_comp** {B : K →L[ℂ] H} {C : H →L[ℂ] K} (hB : IsHilbertSchmidt B) (hC : IsHilbertSchmidt C) : IsTraceClass (B ∘L C)

theorem **isTraceClass_zero** : IsTraceClass (0 : H →L[ℂ] H)

theorem **IsTraceClass.add** {S T : H →L[ℂ] H} (hS : IsTraceClass S) (hT : IsTraceClass T) : IsTraceClass (S + T)

theorem **isTraceClass_sum** {α : Type*} (s : Finset α) {T : α → (H →L[ℂ] H)} (h : ∀ a ∈ s, IsTraceClass (T a)) : IsTraceClass (∑ a ∈ s, T a)

theorem **IsTraceClass.comp_comp** {T : H →L[ℂ] H} (h : IsTraceClass T) (A : H →L[ℂ] K) (A' : K →L[ℂ] H) : IsTraceClass (A ∘L T ∘L A')

theorem **IsTraceClass.comp_left** {T : H →L[ℂ] H} (h : IsTraceClass T) (A : H →L[ℂ] H) : IsTraceClass (A ∘L T)

theorem **IsTraceClass.comp_right** {T : H →L[ℂ] H} (h : IsTraceClass T) (A : H →L[ℂ] H) : IsTraceClass (T ∘L A)

theorem **IsTraceClass.smul** {T : H →L[ℂ] H} (h : IsTraceClass T) (c : ℂ) : IsTraceClass (c • T)

theorem **IsTraceClass.adjoint** {T : H →L[ℂ] H} (h : IsTraceClass T) : IsTraceClass (adjoint T)

theorem **IsTraceClass.isHilbertSchmidt** {T : H →L[ℂ] H} (h : IsTraceClass T) : IsHilbertSchmidt T

@[simp] theorem **trace_zero** : trace (0 : H →L[ℂ] H) = 0

theorem **trace_add** {S T : H →L[ℂ] H} (hS : IsTraceClass S) (hT : IsTraceClass T) : trace (S + T) = trace S + trace T

theorem **trace_sum** {α : Type*} (s : Finset α) {T : α → (H →L[ℂ] H)} (h : ∀ a ∈ s, IsTraceClass (T a)) : trace (∑ a ∈ s, T a) = ∑ a ∈ s, trace (T a)

theorem **trace_smul** (c : ℂ) (T : H →L[ℂ] H) : trace (c • T) = c * trace T

theorem **IsTraceClass.trace_comp_comp** {T : H →L[ℂ] H} (h : IsTraceClass T) (A : H →L[ℂ] K) (A' : K →L[ℂ] H) : trace (A ∘L T ∘L A') = trace (T ∘L A' ∘L A)

theorem **IsTraceClass.trace_conj** {T : H →L[ℂ] H} (h : IsTraceClass T) (A : H ≃L[ℂ] K) : trace ((A : H →L[ℂ] K) ∘L T ∘L (A.symm : K →L[ℂ] H)) = trace T

theorem **isTraceClass_iff_abs** (e : HilbertBasis ι ℂ H) {T : H →L[ℂ] H} : IsTraceClass T ↔ Summable fun i => ⟪e i, CFC.abs T (e i)⟫_ℂ

theorem **isTraceClass_of_finiteDimensional** [FiniteDimensional ℂ H] (T : H →L[ℂ] H) : IsTraceClass T

theorem **trace_eq_linearMap_trace** [FiniteDimensional ℂ H] (T : H →L[ℂ] H) : trace T = LinearMap.trace ℂ H (T : H →ₗ[ℂ] H)

theorem **isUnitaryRep_charRep** {χ : G →* Circle} (hχ : Continuous χ) : IsUnitaryRep (charRep χ)

theorem **IsUnitaryRep.inner_apply_left** {π : G →* (H →L[ℂ] H)} (hπ : IsUnitaryRep π) (g : G) (v w : H) : ⟪π g v, w⟫_ℂ = ⟪v, π g⁻¹ w⟫_ℂ

theorem **IsRepOp.repOp_eq** {T : H →L[ℂ] H} (h : IsRepOp π μ f T) : repOp π μ f = T

theorem **IsUnitaryRep.isRepOp_of_integrable** (hπ : IsUnitaryRep π) (hf : Integrable f μ) (hm : ∀ v, AEStronglyMeasurable (fun g => f g • π g v) μ) : IsRepOp π μ f (repOp π μ f)

theorem **IsUnitaryRep.isRepOp** (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) : IsRepOp π μ f (repOp π μ f)

theorem **IsUnitaryRep.repOp_apply** (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) (v : H) : repOp π μ f v = ∫ g, f g • π g v ∂μ

theorem **IsUnitaryRep.inner_repOp** (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) (w v : H) : ⟪w, repOp π μ f v⟫_ℂ = ∫ g, f g * ⟪w, π g v⟫_ℂ ∂μ

theorem **IsUnitaryRep.repOp_add** (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) (hf' : Continuous f') (hc' : HasCompactSupport f') : repOp π μ (f + f') = repOp π μ f + repOp π μ f'

theorem **IsUnitaryRep.repOp_smul** (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) (c : ℂ) : repOp π μ (c • f) = c • repOp π μ f

theorem **IsUnitaryRep.comp_repOp** {π' : G →* (K →L[ℂ] K)} (hπ : IsUnitaryRep π) (hπ' : IsUnitaryRep π') (hf : Continuous f) (hc : HasCompactSupport f) (A : H →L[ℂ] K) (hA : ∀ g v, A (π g v) = π' g (A v)) : A ∘L repOp π μ f = repOp π' μ f ∘L A

theorem **IsUnitaryRep.repTrace_eq_of_equiv** {π' : G →* (K →L[ℂ] K)} (hπ : IsUnitaryRep π) (hπ' : IsUnitaryRep π') (hf : Continuous f) (hc : HasCompactSupport f) (A : H ≃L[ℂ] K) (hA : ∀ g v, A (π g v) = π' g (A v)) (ht : IsTraceClass (repOp π μ f)) : IsTraceClass (repOp π' μ f) ∧ repTrace π' μ f = repTrace π μ f

theorem **IsUnitaryRep.adjoint_repOp** [IsTopologicalGroup G] [BorelSpace G] [μ.IsInvInvariant] (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) : adjoint (repOp π μ f) = repOp π μ fun g => starRingEnd ℂ (f g⁻¹)

theorem **IsUnitaryRep.repOp_convolution** [IsTopologicalGroup G] [BorelSpace G] [μ.IsMulLeftInvariant] (hπ : IsUnitaryRep π) (hf : Continuous f) (hc : HasCompactSupport f) (hf' : Continuous f') (hc' : HasCompactSupport f') : IsRepOp π μ (fun y => ∫ x, f x * f' (x⁻¹ * y) ∂μ) (repOp π μ f ∘L repOp π μ f')

theorem **repTrace_charRep** {χ : G →* Circle} (hχ : Continuous χ) (hf : Continuous f) (hc : HasCompactSupport f) : HasTrace (repOp (charRep χ) μ f) (∫ g, f g * (χ g : ℂ) ∂μ)

theorem **finRankTrace_eq** (T : V →ₗ[ℂ] V) (W : Submodule ℂ V) [FiniteDimensional ℂ W] (hW : ∀ x, T x ∈ W) : finRankTrace T = LinearMap.trace ℂ W (T.restrict fun x _ => hW x)

theorem **finRankTrace_add** (S T : V →ₗ[ℂ] V) [FiniteDimensional ℂ (LinearMap.range S)] [FiniteDimensional ℂ (LinearMap.range T)] : finRankTrace (S + T) = finRankTrace S + finRankTrace T

theorem **finRankTrace_smul** (c : ℂ) (T : V →ₗ[ℂ] V) [FiniteDimensional ℂ (LinearMap.range T)] : finRankTrace (c • T) = c * finRankTrace T

theorem **finRankTrace_eq_of_equiv** {W : Type*} [AddCommGroup W] [Module ℂ W] (e : V ≃ₗ[ℂ] W) {T : V →ₗ[ℂ] V} {T' : W →ₗ[ℂ] W} (h : ∀ v, e (T v) = T' (e v)) [FiniteDimensional ℂ (LinearMap.range T)] : FiniteDimensional ℂ (LinearMap.range T') ∧ finRankTrace T' = finRankTrace T

theorem **hasTrace_of_finiteRank** {H : Type*} [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (T : H →L[ℂ] H) [FiniteDimensional ℂ (LinearMap.range (T : H →ₗ[ℂ] H))] : HasTrace T (finRankTrace (T : H →ₗ[ℂ] H))

theorem **finite_range** {F : X → V} (hF : IsLocallyConstant F) (hc : HasCompactSupport F) : (Set.range F).Finite

theorem **map_lcIntegral** {E : Type*} [NormedAddCommGroup E] [NormedSpace ℂ E] [CompleteSpace E] {F : X → V} (hF : IsLocallyConstant F) (hc : HasCompactSupport F) (l : V →ₗ[ℂ] E) : l (lcIntegral μ F) = ∫ x, l (F x) ∂μ

theorem **eq_of_forall_dual** {v w : V} (h : ∀ l : V →ₗ[ℂ] ℂ, l v = l w) : v = w

theorem **mem_fixed** {ρ : Representation ℂ G V} {K : Subgroup G} {v : V} : v ∈ fixed ρ K ↔ ∀ k ∈ K, ρ k v = v

theorem **_root_.GlobalTheory.PRep.IsSmooth.isLocallyConstant_orbit** [ContinuousMul G] {ρ : Representation ℂ G V} (hρ : IsSmooth ρ) (v : V) : IsLocallyConstant fun g => ρ g v

theorem **isOpen_leftStabilizer** [IsTopologicalGroup G] {f : G → ℂ} (hf : IsLocallyConstant f) (hc : HasCompactSupport f) : IsOpen (leftStabilizer f : Set G)

theorem **_root_.GlobalTheory.PRep.IsSmooth.smoothOp_integrand** (hρ : IsSmooth ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (v : V) : IsLocallyConstant (fun g => f g • ρ g v) ∧ HasCompactSupport fun g => f g • ρ g v

theorem **integrable_map** {E : Type*} [NormedAddCommGroup E] {F : G → V} (hF : IsLocallyConstant F) (hc : HasCompactSupport F) (l : V → E) (hl : l 0 = 0) : Integrable (fun g => l (F g)) μ

theorem **_root_.GlobalTheory.PRep.IsSmooth.smoothOp_apply** (hρ : IsSmooth ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (v : V) : smoothOp ρ μ f v = lcIntegral μ fun g => f g • ρ g v

theorem **_root_.GlobalTheory.PRep.IsSmooth.map_smoothOp** {E : Type*} [NormedAddCommGroup E] [NormedSpace ℂ E] [CompleteSpace E] (hρ : IsSmooth ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (l : V →ₗ[ℂ] E) (v : V) : l (smoothOp ρ μ f v) = ∫ g, f g • l (ρ g v) ∂μ

theorem **_root_.GlobalTheory.PRep.IsSmooth.smoothOp_add** (hρ : IsSmooth ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (hf' : IsLocallyConstant f') (hc' : HasCompactSupport f') : smoothOp ρ μ (f + f') = smoothOp ρ μ f + smoothOp ρ μ f'

theorem **_root_.GlobalTheory.PRep.IsSmooth.smoothOp_smul** (hρ : IsSmooth ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (c : ℂ) : smoothOp ρ μ (c • f) = c • smoothOp ρ μ f

theorem **_root_.GlobalTheory.PRep.IsSmooth.map_smoothOp_of_intertwining** {W : Type} [AddCommGroup W] [Module ℂ W] {ρ' : Representation ℂ G W} (hρ : IsSmooth ρ) (hρ' : IsSmooth ρ') (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (j : V →ₗ[ℂ] W) (hj : ∀ g v, j (ρ g v) = ρ' g (j v)) (v : V) : j (smoothOp ρ μ f v) = smoothOp ρ' μ f (j v)

theorem **_root_.GlobalTheory.PRep.IsSmooth.smoothOp_mem_fixed** [μ.IsMulLeftInvariant] (hρ : IsSmooth ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) {K : Subgroup G} (hK : ∀ k ∈ K, ∀ g, f (k * g) = f g) (v : V) : smoothOp ρ μ f v ∈ fixed ρ K

theorem **_root_.GlobalTheory.PRep.IsAdmissible.finiteDimensional_range_smoothOp** [μ.IsMulLeftInvariant] (hρ : IsAdmissible ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) {K : Subgroup G} (ho : IsOpen (K : Set G)) (hK : IsCompact (K : Set G)) : FiniteDimensional ℂ (LinearMap.range (smoothOp ρ μ f))

theorem **_root_.GlobalTheory.PRep.IsAdmissible.smoothTrace_add** [μ.IsMulLeftInvariant] (hρ : IsAdmissible ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) (hf' : IsLocallyConstant f') (hc' : HasCompactSupport f') {K : Subgroup G} (ho : IsOpen (K : Set G)) (hK : IsCompact (K : Set G)) : smoothTrace ρ μ (f + f') = smoothTrace ρ μ f + smoothTrace ρ μ f'

theorem **_root_.GlobalTheory.PRep.IsAdmissible.smoothTrace_smul** [μ.IsMulLeftInvariant] (hρ : IsAdmissible ρ) (hf : IsLocallyConstant f) (hc : HasCompactSupport f) {K : Subgroup G} (ho : IsOpen (K : Set G)) (hK : IsCompact (K : Set G)) (c : ℂ) : smoothTrace ρ μ (c • f) = c * smoothTrace ρ μ f

theorem **_root_.GlobalTheory.PRep.IsAdmissible.smoothTrace_eq_of_equiv** [μ.IsMulLeftInvariant] {W : Type} [AddCommGroup W] [Module ℂ W] {ρ' : Representation ℂ G W} (hρ : IsAdmissible ρ) (hρ' : IsSmooth ρ') (hf : IsLocallyConstant f) (hc : HasCompactSupport f) {K : Subgroup G} (ho : IsOpen (K : Set G)) (hK : IsCompact (K : Set G)) (e : V ≃ₗ[ℂ] W) (he : ∀ g v, e (ρ g v) = ρ' g (e v)) : smoothTrace ρ' μ f = smoothTrace ρ μ f

theorem **_root_.GlobalTheory.PRep.IsAdmissible.smoothTrace_indicator** [μ.IsMulLeftInvariant] (hρ : IsAdmissible ρ) {K : Subgroup G} (ho : IsOpen (K : Set G)) (hK : IsCompact (K : Set G)) : smoothTrace ρ μ ((K : Set G).indicator fun _ => 1) = μ.real K * Module.finrank ℂ (fixed ρ K)

Development.AbelianVarietyTheory.Definitions.Foundations.IsIntegralIff

theorem **hf_add_left** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (a b y : Fin n → E) : hf L H (a + b) y = hf L H a y + hf L H b y

theorem **hf_sub_left** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (a b y : Fin n → E) : hf L H (a - b) y = hf L H a y - hf L H b y

theorem **hf_smul_left** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (c : E) (a y : Fin n → E) : hf L H (c • a) y = c * hf L H a y

theorem **hf_add_right** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (a b y : Fin n → E) : hf L H y (a + b) = hf L H y a + hf L H y b

theorem **hf_sub_right** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (a b y : Fin n → E) : hf L H y (a - b) = hf L H y a - hf L H y b

theorem **hf_smul_right** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (c : E) (a y : Fin n → E) : hf L H y (c • a) = L.cj c * hf L H y a

theorem **hf_symm** (L : LocE F E) {n : ℕ} {H : Matrix (Fin n) (Fin n) E} (h : IsHerm L H) (x y : Fin n → E) : hf L H y x = L.cj (hf L H x y)

theorem **hf_single** (L : LocE F E) {n : ℕ} (H : Matrix (Fin n) (Fin n) E) (i j : Fin n) : hf L H (Pi.single i 1) (Pi.single j 1) = H i j

theorem **eich_isom** (L : LocE F E) {n : ℕ} {H : Matrix (Fin n) (Fin n) E} (h : IsHerm L H) (x u : Fin n → E) (s : E) (hxx : hf L H x x = 0) (hux : hf L H u x = 0) (hs : s + L.cj s = hf L H u u) (v w : Fin n → E) : hf L H (eich L H x u s v) (eich L H x u s w) = hf L H v w

theorem **eich_inv** (L : LocE F E) {n : ℕ} {H : Matrix (Fin n) (Fin n) E} (h : IsHerm L H) (x u : Fin n → E) (s : E) (hxx : hf L H x x = 0) (hux : hf L H u x = 0) (hs : s + L.cj s = hf L H u u) (v : Fin n → E) : eich L H x (-u) (L.cj s) (eich L H x u s v) = v

theorem **isHermND_one** (L : LocE F E) (n : ℕ) : IsHermND L (1 : Matrix (Fin n) (Fin n) E)

private theorem **cj_mem_O_of_mem_O'** (L : LocE F E) {x : E} (hx : x ∈ L.O) : L.cj x ∈ L.O

private theorem **mem_stdLat_iff_forall_mem_O** (L : LocE F E) (n : ℕ) (y : Fin n → E) : y ∈ stdLat L n ↔ ∀ i, y i ∈ L.O

theorem **isSD_std** (L : LocE F E) (n : ℕ) : IsSD L (1 : Matrix (Fin n) (Fin n) E) (stdLat L n)

theorem **nontrivial** (L : LocE F E) : Nontrivial E

theorem **not_isLat_bot** (L : LocE F E) (n : ℕ) : ¬ IsLat L (⊥ : Submodule L.O (Fin (n + 1) → E))

theorem **isSD_zero** (L : LocE F E) {n : ℕ} {Λ : Submodule L.O (Fin n → E)} (h : IsSD L 0 Λ) : Λ = ⊤

theorem **unipOn_top_bot** {n : ℕ} : unipOn (E

theorem **invVol_self** {G : Type*} [Group G] (K : Subgroup G) : invVol K K = 1

theorem **invVol_of_le** {G : Type*} [Group G] {K0 K : Subgroup G} (h : K ≤ K0) : invVol K0 K = K.relIndex K0

theorem **lead_zero** {N : ℕ} (Z : Matrix (Fin N) (Fin N) E) : lead Z 0 = 1

theorem **lead_last** {N : ℕ} (Z : Matrix (Fin N) (Fin N) E) : lead Z (Fin.last N) = Z.det

theorem **lead_one** {N : ℕ} (Z : Matrix (Fin (N + 1)) (Fin (N + 1)) E) : lead Z ⟨1, by omega⟩ = Z 0 0

theorem **vE_zero** (L : LocE F E) (ϖ : 𝒪[F]) : vE L ϖ 0 = 0

theorem **algebraMap_mem_iff** (L : LocE F E) (a : F) : algebraMap F E a ∈ L.O ↔ a ∈ 𝒪[F]

theorem **vE_pow** (L : LocE F E) (ϖ : 𝒪[F]) (hϖ : Irreducible ϖ) (u : 𝒪[F]ˣ) (k : ℤ) : vE L ϖ (algebraMap F E ((ϖ : F) ^ k * ((u : 𝒪[F]) : F))) = k

theorem **exists_char** (a : F) (ha : a ∉ 𝒪[F]) : ∃ ψ : AddChar F ℂ, (∀ y ∈ 𝒪[F], ψ y = 1) ∧ ψ a ≠ 1

theorem **isIntegral_iff** (a : F) : IsIntegral 𝒪[F] a ↔ a ∈ 𝒪[F]

theorem **nonempty_split** [IsNonarchimedeanLocalField F] : ∃ L : LocE F (F × F), (∀ x, L.cj x = x.swap) ∧ L.O = (𝒪[F]).prod 𝒪[F]

theorem **exists_isSD** [IsNonarchimedeanLocalField F] (n : ℕ) : ∃ (L : LocE F (F × F)) (H : Matrix (Fin n) (Fin n) (F × F)) (Λ : Submodule L.O (Fin n → F × F)), IsHermND L H ∧ IsSD L H Λ

theorem **resQ_eq** [IsNonarchimedeanLocalField F] (ϖ : 𝒪[F]) (h : Irreducible ϖ) : resQ ϖ = Nat.card (IsLocalRing.ResidueField 𝒪[F]) ∧ 2 ≤ resQ ϖ

Development.AbelianVarietyTheory.Definitions.Foundations.IsIwasawaOfIsCompact

theorem **hom_pos** {H : Type} [Group H] (δ : H →* ℝ≥0) (p : H) : 0 < δ p

theorem **iwAbs_pos** (g : G) : 0 < iwAbs δ K g

theorem **iwAbs_mul_of_triv** (h : ∀ p : P, (p : G) ∈ K → δ p = 1) (p : P) {k : G} (hk : k ∈ K) : iwAbs δ K (p * k) = δ p

theorem **iwAbs_mul** (p : P) {k : G} (hk : k ∈ K) : iwAbs δ K (p * k) = δ p

theorem **iwAbs_coe** (p : P) : iwAbs δ K p = δ p

theorem **iwAbs_of_mem** {k : G} (hk : k ∈ K) : iwAbs δ K k = 1

theorem **iwAbs_left** (p : P) (g : G) : iwAbs δ K (p * g) = δ p * iwAbs δ K g

theorem **iwAbs_right** (g : G) {k : G} (hk : k ∈ K) : iwAbs δ K (g * k) = iwAbs δ K g

theorem **eq_iwAbs** {f : G → ℝ≥0} (h1 : f 1 = 1) (hl : ∀ (p : P) g, f (p * g) = δ p * f g) (hr : ∀ g, ∀ k ∈ K, f (g * k) = f g) : f = iwAbs δ K

theorem **triv_of_isCompact** (hP : IsClosed (P : Set G)) (hδ : Continuous δ) (hK : IsCompact (K : Set G)) (p : P) (hp : (p : G) ∈ K) : δ p = 1

theorem **isIwasawa_of_isCompact** (hP : IsClosed (P : Set G)) (hδ : Continuous δ) (hK : IsCompact (K : Set G)) (h : ∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) : IsIwasawa δ K

theorem **continuous_of_prod** (hP : IsClosed (P : Set G)) (hK : IsCompact (K : Set G)) (h : ∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) {X : Type} [TopologicalSpace X] {f : G → X} (hf : Continuous fun x : P × K => f (x.1 * x.2)) : Continuous f

theorem **IsIwasawa.continuous_iwAbs** (h : IsIwasawa δ K) (hP : IsClosed (P : Set G)) (hδ : Continuous δ) (hK : IsCompact (K : Set G)) : Continuous (iwAbs δ K)

theorem **npow_mul** (r r' : ℝ≥0) (t : ℂ) : npow (r * r') t = npow r t * npow r' t

theorem **npow_add** {r : ℝ≥0} (hr : 0 < r) (t t' : ℂ) : npow r (t + t') = npow r t * npow r t'

@[simp] theorem **npow_zero** (r : ℝ≥0) : npow r 0 = 1

@[simp] theorem **one_npow** (t : ℂ) : npow 1 t = 1

theorem **norm_npow** {r : ℝ≥0} (hr : 0 < r) (t : ℂ) : ‖npow r t‖ = (r : ℝ) ^ t.re

theorem **npow_ne_zero** {r : ℝ≥0} (hr : 0 < r) (t : ℂ) : npow r t ≠ 0

theorem **npow_eq_exp** {r : ℝ≥0} (hr : 0 < r) (t : ℂ) : npow r t = Complex.exp (Real.log r * t)

theorem **continuous_npow** {X : Type} [TopologicalSpace X] {f : X → ℝ≥0} (hf : Continuous f) (h : ∀ x, 0 < f x) (t : ℂ) : Continuous fun x => npow (f x) t

theorem **isKFinite_iff** {Φ : G → ℂ} : IsKFinite K Φ ↔ ∃ W : Submodule ℂ (G → ℂ), FiniteDimensional ℂ W ∧ ∀ k ∈ K, (fun g => Φ (g * k)) ∈ W

theorem **IsKFinite.of_invariant** {Φ : G → ℂ} (h : ∀ g, ∀ k ∈ K, Φ (g * k) = Φ g) : IsKFinite K Φ

theorem **IsKFinite.add** {Φ Ψ : G → ℂ} (h : IsKFinite K Φ) (h' : IsKFinite K Ψ) : IsKFinite K (Φ + Ψ)

theorem **IsKFinite.mul_invariant** {Φ f : G → ℂ} (h : IsKFinite K Φ) (hf : ∀ g, ∀ k ∈ K, f (g * k) = f g) : IsKFinite K fun g => Φ g * f g

theorem **IsKFinite.translate** {Φ : G → ℂ} (h : IsKFinite K Φ) {k : G} (hk : k ∈ K) : IsKFinite K fun g => Φ (g * k)

theorem **mem_indSpace** [TopologicalSpace G] {Φ : G → ℂ} : Φ ∈ indSpace δ K χ t ↔ IndLaw δ χ t Φ ∧ Continuous Φ ∧ IsKFinite K Φ

theorem **translate_mem_indSpace** [TopologicalSpace G] [ContinuousMul G] {Φ : G → ℂ} (h : Φ ∈ indSpace δ K χ t) {k : G} (hk : k ∈ K) : (fun g => Φ (g * k)) ∈ indSpace δ K χ t

@[simp] theorem **flatSec_self** (Φ : G → ℂ) (s₀ : ℂ) : flatSec δ K Φ s₀ s₀ = Φ

theorem **flatSec_flatSec** (Φ : G → ℂ) (s₀ s₁ s : ℂ) : flatSec δ K (flatSec δ K Φ s₀ s₁) s₁ s = flatSec δ K Φ s₀ s

theorem **norm_flatSec** (Φ : G → ℂ) (s₀ s : ℂ) (g : G) : ‖flatSec δ K Φ s₀ s g‖ = ‖Φ g‖ * (iwAbs δ K g : ℝ) ^ (s.re - s₀.re)

theorem **IndLaw.ext** (h : ∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) {Φ Ψ : G → ℂ} (h1 : IndLaw δ χ t Φ) (h2 : IndLaw δ χ t Ψ) (e : ∀ k ∈ K, Φ k = Ψ k) : Φ = Ψ

theorem **flatSec_of_mem** (Φ : G → ℂ) (s₀ s : ℂ) {k : G} (hk : k ∈ K) : flatSec δ K Φ s₀ s k = Φ k

theorem **isFlat_flatSec** (Φ : G → ℂ) (s₀ : ℂ) : IsFlat K (flatSec δ K Φ s₀)

theorem **indLaw_flatSec** {Φ : G → ℂ} {ρ s₀ : ℂ} (hΦ : IndLaw δ χ (s₀ + ρ) Φ) (s : ℂ) : IndLaw δ χ (s + ρ) (flatSec δ K Φ s₀ s)

theorem **eq_flatSec** {Φ : ℂ → G → ℂ} {ρ : ℂ} (hΦ : ∀ s, IndLaw δ χ (s + ρ) (Φ s)) (hf : IsFlat K Φ) (s₀ s : ℂ) : Φ s = flatSec δ K (Φ s₀) s₀ s

theorem **norm_le** {Φ : G → ℂ} (hΦ : IndLaw δ χ t Φ) {C : ℝ} (hC : ∀ k ∈ K, ‖Φ k‖ ≤ C) (g : G) : ‖Φ g‖ ≤ C * (iwAbs δ K g : ℝ) ^ t.re

theorem **flatSec_mem** (hP : IsClosed (P : Set G)) (hδ : Continuous δ) (hK : IsCompact (K : Set G)) {Φ : G → ℂ} {ρ s₀ : ℂ} (hΦ : Φ ∈ indSpace δ K χ (s₀ + ρ)) (s : ℂ) : flatSec δ K Φ s₀ s ∈ indSpace δ K χ (s + ρ)

theorem **npow_iwAbs_mem** (hP : IsClosed (P : Set G)) (hδ : Continuous δ) (hK : IsCompact (K : Set G)) (t : ℂ) : (fun g => npow (iwAbs δ K g) t) ∈ indSpace δ K 1 t

theorem **IsRightSmooth.continuous** [TopologicalSpace G] [IsTopologicalGroup G] {Φ : G → ℂ} (h : IsRightSmooth Φ) : Continuous Φ

theorem **IsRightSmooth.isKFinite** [TopologicalSpace G] [IsTopologicalGroup G] {Φ : G → ℂ} (h : IsRightSmooth Φ) (hK : IsCompact (K : Set G)) : IsKFinite K Φ

theorem **IsRightSmooth.mem_indSpace** [TopologicalSpace G] [IsTopologicalGroup G] {Φ : G → ℂ} (h : IsRightSmooth Φ) (hK : IsCompact (K : Set G)) (hl : IndLaw δ χ t Φ) : Φ ∈ indSpace δ K χ t

theorem **conj_coef** (π : H → X → X) (x y : X) (g : H) : starRingEnd ℂ (coef π x y g) = inner ℂ (π g x) y

Development.AbelianVarietyTheory.Definitions.Foundations.IsLamFR2

theorem **IsLamFR2.diam** (h : IsLamFR2 X A N w χ j pr c lam) {d : ℕ} (hd : Nat.Coprime d N) {b : X.Fn l} (hb : b ∈ X.B N l w) : pr (X.diam N l d b) = ((χ d : ℤ_[l]) : Qbar l) • pr b

Development.AbelianVarietyTheory.Definitions.Foundations.IsLamOfLSmulIff

theorem **norm_l_pow** (n : ℕ) : ‖((l : ℕ) : Qbar l) ^ n‖ = ((l : ℝ)⁻¹) ^ n

theorem **cg_mono** {m n : ℕ} (h : m ≤ n) {x y : Qbar l} (hxy : Cg n x y) : Cg m x y

theorem **exists_le_l_pow** (l : ℕ) [Fact l.Prime] (t : ℝ) : ∃ k : ℕ, t ≤ (l : ℝ) ^ k

theorem **l_pow_mul_norm** {k m : ℕ} (h : k ≤ m) : (l : ℝ) ^ k * ‖((l : ℕ) : Qbar l) ^ m‖ = ‖((l : ℕ) : Qbar l) ^ (m - k)‖

theorem **cg_sub_iff** {c n : ℕ} (h : c ≤ n) (x y : Qbar l) : Cg (n - c) x y ↔ ‖x - y‖ ≤ (l : ℝ) ^ c * ‖((l : ℕ) : Qbar l) ^ n‖

theorem **cg_mul_left** {k m : ℕ} (h : k ≤ m) {a x y : Qbar l} (ha : ‖a‖ ≤ (l : ℝ) ^ k) (hxy : Cg m x y) : Cg (m - k) (a * x) (a * y)

theorem **seqN_single** {N : ℕ} (hN : N ≠ 0) (j : Ω →+* Qbar l) (s : ℕ) (a : Ω) : seqN N j (HahnSeries.single ((s : ℚ) / N) a) = Pi.single s (j a)

theorem **finSupp_proj** (s : ℕ) : FinSupp (LinearMap.proj (R

theorem **FinSupp.smul** {lam : (ℕ → Qbar l) →₀[Qbar l] Qbar l} (h : FinSupp lam) (a : Qbar l) : FinSupp (a • lam)

theorem **FinSupp.add** {lam lam' : (ℕ → Qbar l) →₀[Qbar l] Qbar l} (h : FinSupp lam) (h' : FinSupp lam') : FinSupp (lam + lam')

theorem **FinSupp.exists_sum** {lam : (ℕ → Qbar l) →₀[Qbar l] Qbar l} (h : FinSupp lam) : ∃ S : Finset ℕ, ∀ f : ℕ → Qbar l, lam f = ∑ s ∈ S, lam (Pi.single s 1) * f s

theorem **FinSupp.exists_loss** {lam : (ℕ → Qbar l) →₀[Qbar l] Qbar l} (h : FinSupp lam) : ∃ c : ℕ, ∀ n, c ≤ n → ∀ f g : ℕ → Qbar l, (∀ s, Cg n (f s) (g s)) → Cg (n - c) (lam f) (lam g)

theorem **IsLamOfL.smul** (h : IsLamOfL X A N w j pr lam) (a : Qbar l) : IsLamOfL X A N w j (a • pr) (a • lam)

theorem **IsLamOfL.congr** (h : IsLamOfL X A N w j pr lam) {pr' : X.Fn l → Fin 2 → Qbar l} (hp : ∀ b ∈ X.B N l w, pr' b 1 = pr b 1) : IsLamOfL X A N w j pr' lam

Development.AbelianVarietyTheory.Definitions.Foundations.IsLocallyConstantCycMod

theorem **not_gr_of_nzAt** {W : R2 Ω l} {A : ValuationSubring Ω} {y : G ℚ Ω → Fin 2 → Qbar l} (h : NzAt W A y) : ¬ Gr W A y

theorem **es_of_sys** {W : R2 Ω l} {A : ValuationSubring Ω} (h : ∃ c y S', Sys W c y S' ∧ NzAt W A y) : ES W A

theorem **sp_mul** {r : ℕ} {z : G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l} (hz : Layer c A ν r z) (η : (ZMod (l ^ r))ˣ →* Qbar l) {s t : G ℚ Ω} (hs : s ∈ Dec (F

theorem **smul** {Z : ∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l} (h : IwSysC c A ν Z) {a : Qbar l} (ha : a ∈ c.O) : IwSysC c A ν fun r s x => a * Z r s x

theorem **map_cycMod_succ** (r : ℕ) (s : G ℚ Ω) : Units.map (ZMod.castHom (pow_dvd_pow l (Nat.le_succ r)) (ZMod (l ^ r))).toMonoidHom (cycMod ℚ Ω l (r + 1) s) = cycMod ℚ Ω l r s

private lemma **continuous_toZModPow_foundations** {l : ℕ} [Fact l.Prime] (r : ℕ) : Continuous (PadicInt.toZModPow (p

theorem **isLocallyConstant_cycMod** [IsAlgClosure ℚ Ω] (r : ℕ) : IsLocallyConstant (cycMod ℚ Ω l r)

theorem **iwSysC_cobDirac** [IsAlgClosure ℚ Ω] (hO : ∀ s ∈ Dec (F

Development.AbelianVarietyTheory.Definitions.Foundations.IsOneCocycleOnSubgroup

@[simp] theorem **sh_apply** (S : Type*) [MulAction G S] (ρ : Representation R G M) (g : G) (f : S → M) (s : S) : sh S ρ g f s = ρ g (f (g⁻¹ • s))

@[simp] theorem **tw_apply** (χ : G →* Rˣ) (ρ : Representation R G M) (g : G) (m : M) : tw χ ρ g m = (χ g : R) • ρ g m

@[simp] theorem **pull_apply** {S S' : Type*} (π : S' → S) (f : S → M) (s : S') : pull (R

@[simp] theorem **ptw_apply** (S : Type*) (β : M →ₗ[R] M' →ₗ[R] M'') (f : S → M) (f' : S → M') (s : S) : ptw S β f f' s = β (f s) (f' s)

@[simp] theorem **fam_apply** {S : Type*} (f : S → M →ₗ[R] M') (m : S → M) (s : S) : fam f m s = f s (m s)

Development.AbelianVarietyTheory.Definitions.Foundations.IsPrimitiveRootTopZeta

theorem **mem_fixMu_iff** {M : ℕ} (s : Ω ≃ₐ[ℚ] Ω) : s ∈ fixMu Ω M ↔ ∀ z : Ω, z ^ M = 1 → s z = z

theorem **kumH_apply** {F Ω' : Type*} [Field F] [Field Ω'] [Algebra F Ω'] {N : ℕ} [NeZero N] {ζ θ : Ω'} (hζ : IsPrimitiveRoot ζ N) (hfix : ∀ g : Ω' ≃ₐ[F] Ω', g ζ = ζ) {w : F} (hw0 : w ≠ 0) (hw : θ ^ N = algebraMap F Ω' w) (g : Ω' ≃ₐ[F] Ω') : (kumH (F

theorem **eps_eOm** [IsAlgClosed Ω] (q : ℚ) : ε (eOm ε q) = Complex.exp (2 * Real.pi * Complex.I * (q : ℂ))

theorem **isPrimitiveRoot_eOm** [IsAlgClosed Ω] (ε : Ω →+* ℂ) {m : ℕ} (hm : m ≠ 0) : IsPrimitiveRoot (eOm ε (1 / (m : ℚ))) m

theorem **isPrimitiveRoot_topZeta** [IsAlgClosed Ω] (X : Tower Ω ε) (M : ℕ) {l : ℕ} (hl : l ≠ 0) (n : ℕ) : IsPrimitiveRoot (topZeta X M l n) (l ^ n)

theorem **topGal_topZeta** [IsAlgClosed Ω] (X : Tower Ω ε) {M l n : ℕ} (hl : l ≠ 0) (hM : l ^ n ∣ M) (g : topGal X M) : g (topZeta X M l n) = topZeta X M l n

theorem **topRoot_pow** (X : Tower Ω ε) (M : ℕ) {m : ℕ} (hm : m ≠ 0) {w : X.K} (hw : w ∈ topL X M) : topRoot X M m w ^ m = algebraMap (topL X M) (topBar X M) (w, hw)

theorem **topChar_apply** [IsAlgClosed Ω] (X : Tower Ω ε) {M l n : ℕ} (hl : l ≠ 0) (hM : l ^ n ∣ M) {w : X.K} (hw : w ∈ topF X M) (hw0 : w ≠ 0) (g : topGal X M) : (topChar X M l n w g).toAdd = kum (K

theorem **isTopEta_topEta** (X : Tower Ω ε) (M : ℕ) {m : ℕ} {x y : topGal X M →* Multiplicative (ZMod m)} (h : ∃ η, IsTopEta X M x y η) : IsTopEta X M x y (topEta X M x y)

theorem **gal_mul_apply** (X : Tower Ω ε) (s t : Ω ≃ₐ[ℚ] Ω) (f : X.K) : X.gal (s * t) f = X.gal s (X.gal t f)

theorem **gal_one_apply** (X : Tower Ω ε) (f : X.K) : X.gal 1 f = f

theorem **gal_mem_topL** [IsAlgClosed Ω] (X : Tower Ω ε) {M : ℕ} (hM : M ≠ 0) {σ : Ω ≃ₐ[ℚ] Ω} (hσ : σ ∈ fixMu Ω M) {f : X.K} (hf : f ∈ topL X M) : X.gal σ f ∈ topL X M

theorem **exists_isTopLift** [IsAlgClosed Ω] (X : Tower Ω ε) {M : ℕ} (hM : M ≠ 0) {σ : Ω ≃ₐ[ℚ] Ω} (hσ : σ ∈ fixMu Ω M) : ∃ φ, IsTopLift X M σ φ

theorem **isTopLift_topLift** [IsAlgClosed Ω] (X : Tower Ω ε) {M : ℕ} (hM : M ≠ 0) {σ : Ω ≃ₐ[ℚ] Ω} (hσ : σ ∈ fixMu Ω M) : IsTopLift X M σ (topLift X M σ)

theorem **IsTopLift.inv_mul_mem_geo** {X : Tower Ω ε} {M : ℕ} {σ : Ω ≃ₐ[ℚ] Ω} {φ ψ : topGal X M} (hφ : IsTopLift X M σ φ) (hψ : IsTopLift X M σ ψ) : φ⁻¹ * ψ ∈ geo (topF X M) (topL X M) (topBar X M)

theorem **topT_spec** {l N : ℕ} [NeZero N] (h : Nat.Coprime l N) (s : ℕ) : (l : ZMod N) ^ s * (topT l s N : ZMod N) = 1

theorem **wkTop_mem** (X : Tower Ω ε) (l n r N : ℕ) (x : ZMod N) (c d : ℕ) (σ : Ω ≃ₐ[ℚ] Ω) : wkTop X l n r N x c d σ ∈ topL X (l ^ (r + n) * N)

theorem **pow_dvd_topLevel** (l n r N : ℕ) : l ^ n ∣ l ^ (r + n) * N

Development.AbelianVarietyTheory.Definitions.Foundations.IsProbabilityMeasureMuPart

theorem **norm_realize_sq** {ι : Type} [Fintype ι] [DecidableEq ι] (b : Basis ι K V) (X : InfinitePlace K → Matrix ι ι ℂ) (x : V) : ‖Formalism.HermitianLatticesTheory.realize b X x‖ ^ 2 = ∑ v, Formalism.HermitianLatticesTheory.normAt b X v x ^ 2

theorem **IsGram.norm_realize** {ι κ : Type} [Fintype ι] [DecidableEq ι] [Fintype κ] [DecidableEq κ] {b : Basis ι K V} {X : InfinitePlace K → Matrix ι ι ℂ} {b' : Basis κ K V} {X' : InfinitePlace K → Matrix κ κ ℂ} {N : (K →+* ℂ) → V → ℝ} (h : IsGram b X N) (h' : IsGram b' X' N) (x : V) : ‖Formalism.HermitianLatticesTheory.realize b X x‖ = ‖Formalism.HermitianLatticesTheory.realize b' X' x‖

theorem **IsGram.rk_deg** {ι κ : Type} [Fintype ι] [DecidableEq ι] [Fintype κ] [DecidableEq κ] {b : Basis ι K V} {X : InfinitePlace K → Matrix ι ι ℂ} {b' : Basis κ K V} {X' : InfinitePlace K → Matrix κ κ ℂ} {N : (K →+* ℂ) → V → ℝ} (h : IsGram b X N) (h' : IsGram b' X' N) {L : Submodule (𝒪 K) V} (hL : L.FG) : Formalism.HN.rk (Formalism.HermitianLatticesTheory.realize b X) L = Formalism.HN.rk (Formalism.HermitianLatticesTheory.realize b' X') L ∧ Formalism.HN.deg (Formalism.HermitianLatticesTheory.realize b X) L = Formalism.HN.deg (Formalism.HermitianLatticesTheory.realize b' X') L

theorem **IsGram.hasGram** {ι : Type} [Fintype ι] [DecidableEq ι] {b : Basis ι K V} {X : InfinitePlace K → Matrix ι ι ℂ} {N : (K →+* ℂ) → V → ℝ} (h : IsGram b X N) : HasGram N

theorem **slope_sq_eq** {ι : Type} [Fintype ι] [DecidableEq ι] {b : Basis ι K V} {X : InfinitePlace K → Matrix ι ι ℂ} {N : (K →+* ℂ) → V → ℝ} (h : IsGram b X N) {L : Submodule (𝒪 K) V} (hL : L.FG) : slope_sq N L = Formalism.HN.muDef b X L

theorem **slopeMax_slopeMin_eq** {ι : Type} [Fintype ι] [DecidableEq ι] {b : Basis ι K V} {X : InfinitePlace K → Matrix ι ι ℂ} {N : (K →+* ℂ) → V → ℝ} (h : IsGram b X N) {L : Submodule (𝒪 K) V} (hL : L.FG) (h0 : L ≠ ⊥) : slopeMax_sq N L = Formalism.HN.muMaxDef b X L ∧ slopeMin_sq N L = Formalism.HN.muMinDef b X L

theorem **deg_bot** (N : (K →+* ℂ) → V → ℝ) : deg N (⊥ : Submodule (𝒪 K) V) = 0

theorem **slope_sq_line** [IsTotallyComplex K] (x : K) (hx : x ≠ 0) : slope_sq (fun (σ : K →+* ℂ) (y : K) => ‖σ y‖) (Submodule.span (𝒪 K) {x}) = -((1 / (Formalism.HN.nPl K : ℝ)) * ∑ v : InfinitePlace K, 2 * Real.log (v x))

theorem **wtOf_snd** (la : Weight Dw) (m : ℤ) : (wtOf Dw la m).2 = 2 * m

theorem **even_wtOf** (la : Weight Dw) (m : ℤ) : Even (wtOf Dw la m).2

theorem **conjEmb_conjEmb** (τ : E →+* ℂ) : conjEmb (conjEmb τ) = τ

theorem **conj_not_mem** (τ : Psi H) : ¬ conjEmb τ.1 (H.rt : E) = H.rt0

theorem **wtOf_big** (la : Weight Dw) (m : ℤ) (τ : Psi H) (i : ℕ) (hi : i < Dw.p) : (wtOf Dw la m).1 (τ.1, i) = la τ (i, by omega) - m

theorem **wtOf_small** (la : Weight Dw) (m : ℤ) (τ : Psi H) (i : ℕ) (hi : i < Dw.q) : (wtOf Dw la m).1 (conjEmb τ.1, i) = -la τ (Dw.p + (Dw.q - 1 - i), by omega) - m

theorem **wtOf_add** (la la' : Weight Dw) (m m' : ℤ) : wtOf Dw (la + la') (m + m') = wtOf Dw la m + wtOf Dw la' m'

theorem **wtOf_zero** : wtOf Dw 0 0 = 0

theorem **exOf_add** (la la' : Weight Dw) (m m' : ℤ) : exOf Dw (la + la') (m + m') = exOf Dw la m + exOf Dw la' m'

theorem **exZ_wtZ** (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : exZ Dw (wtZ a n) = fun i => (ex Dw a n i : ℤ)

theorem **weightOf_wtOf** (la : Weight Dw) (m : ℤ) : weightOf Dw (wtOf Dw la m).1 m = la

theorem **conj_mem** {τ : E →+* ℂ} (h : ¬ τ (H.rt : E) = H.rt0) : conjEmb τ (H.rt : E) = H.rt0

theorem **exOf_weightOf** (a : (E →+* ℂ) × ℕ → ℤ) (m : ℤ) : exOf Dw (weightOf Dw a m) m = exZ Dw (a, 2 * m)

theorem **mem_secZ** {e : Ix Dw → ℤ} {f : Rul Xo c} : f ∈ secZ Xo c e ↔ IsNatural (K

theorem **mem_secZ_exZ** {G : WTZ E} {f : Rul Xo c} : f ∈ secZ Xo c (exZ Dw G) ↔ GlobalTheory.IndefiniteRowSideFullHecke.ZeroWeightSections.IsSecZ G f

theorem **secZ_nat** (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : secZ Xo c (fun i => (ex Dw a n i : ℤ)) = sec Xo c a n

theorem **mul_mem_secZ** {e e' : Ix Dw → ℤ} {f g : Rul Xo c} (hf : f ∈ secZ Xo c e) (hg : g ∈ secZ Xo c e') : f * g ∈ secZ Xo c (e + e')

theorem **secZ_mul_le** (e e' : Ix Dw → ℤ) : secZ Xo c e * secZ Xo c e' ≤ secZ Xo c (e + e')

theorem **one_mem_secZ** : (1 : Rul Xo c) ∈ secZ Xo c 0

theorem **inv_mem_secZ** {e : Ix Dw → ℤ} {u : (Rul Xo c)ˣ} (hu : (u : Rul Xo c) ∈ secZ Xo c e) : ((u⁻¹ : (Rul Xo c)ˣ) : Rul Xo c) ∈ secZ Xo c (-e)

theorem **modelWt_wtZ** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : modelWt U (wtZ a n) = laU U a n

theorem **modelWt_add** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) {G G' : WTZ E} (hG : Even G.2) (hG' : Even G'.2) (k : U.Comp) : modelWt U (G + G') k = modelWt U G k + modelWt U G' k

theorem **relab_injective** (σ : c.F →+* ℂ) : Function.Injective (relab (E

theorem **relab_conj** (σ : c.F →+* ℂ) (τ : E →+* ℂ) : relab c σ (conjEmb τ) = conjEmb (relab c σ τ)

theorem **modelWt_wtOf_big** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (la : Weight Dw) (m : ℤ) (k : U.Comp) (hq : (U.X k).p0 = Dw.q) (τ : Psi H) (hτ : relab c σ τ.1 ∈ (U.X k).Ψ) (i : ℕ) (hi : i < Dw.p) (hr : Dw.q + i < r) : modelWt U (wtOf Dw la m) k ⟨_, hτ⟩ ⟨_, hr⟩ = la τ ⟨i, by omega⟩

theorem **modelWt_wtOf_small** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (la : Weight Dw) (m : ℤ) (k : U.Comp) (hq : (U.X k).p0 = Dw.q) (τ : Psi H) (hτ : relab c σ τ.1 ∈ (U.X k).Ψ) (j : ℕ) (hj : j < Dw.q) (hr : j < r) : modelWt U (wtOf Dw la m) k ⟨_, hτ⟩ ⟨_, hr⟩ = la τ ⟨Dw.p + j, by omega⟩

theorem **mem_cuspW** {G : WTZ E} {f : Rul Xo c} : f ∈ cuspW Xo c U G ↔ f ∈ secZ Xo c (exZ Dw G) ∧ ∀ σ : c.F →+* ℂ, GoodEmb c σ → rdU (U σ) f ∈ holSub (U σ).X (modelWt (U σ) G)

theorem **cuspW_le_secZ** (G : WTZ E) : cuspW Xo c U G ≤ secZ Xo c (exZ Dw G)

theorem **cuspW_wtZ** (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : cuspW Xo c U (wtZ a n) = cusp Xo c U a n

theorem **modelWt_congr** {σ : c.F →+* ℂ} (U : Unf Xo c σ r) (hσ : GoodEmb c σ) (hU : IsUnf U) {G G' : WTZ E} (h : exZ Dw G = exZ Dw G') : modelWt U G = modelWt U G'

theorem **cuspZ_weightOf** (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) (a : (E →+* ℂ) × ℕ → ℕ) (m : ℕ) : cuspZ Xo c U (weightOf Dw (fun x => (a x : ℤ)) m) m = cusp Xo c U a (2 * m)

theorem **liesOff_iff** (Z : Comp Xo c) {S : Type} [Field S] [Algebra c.F S] (y : FrPt Xo c S) : LiesOff Z y ↔ ¬ LiesOn Z y

theorem **mem_suppOn** {Z : Comp Xo c} {f : Rul Xo c} : f ∈ suppOn Z ↔ ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), LiesOff Z y → evR Xo c f y = 0

theorem **mul_mem_suppOn** {Z : Comp Xo c} (g : Rul Xo c) {f : Rul Xo c} (hf : f ∈ suppOn Z) : g * f ∈ suppOn Z

theorem **ptF_genFr** (h : GenFrEx Xo c) (Z : Comp Xo c) (s : Sp (Om Z)) : ptF (genFr h Z) s = genPt Z

theorem **liesOn_genFr** (h : GenFrEx Xo c) (Z : Comp Xo c) : LiesOn Z (genFr h Z)

theorem **liesOff_genFr** (h : GenFrEx Xo c) {Z Z' : Comp Xo c} (hne : Z' ≠ Z) : LiesOff Z (genFr h Z')

theorem **evC_eq_zero** (h : GenFrEx Xo c) {Z Z' : Comp Xo c} (hne : Z' ≠ Z) {f : Rul Xo c} (hf : f ∈ suppOn Z) : evC h Z' f = 0

theorem **subTower** {K R : Type} [Field K] [CommRing R] [Algebra K R] [Algebra (𝒪 K) R] [IsScalarTower (𝒪 K) K R] (C : Submodule K R) : IsScalarTower (𝒪 K) K C

theorem **cuspOn_le** (Z : Comp Xo c) (G : WTZ E) : cuspOn U Z G ≤ cuspW Xo c U G

theorem **mem_supLat** {C : Submodule c.F (Rul Xo c)} {f : C} : f ∈ supLat C ↔ ∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v (f : Rul Xo c)

theorem **mul_mem_supLat** {C C' C'' : Submodule c.F (Rul Xo c)} {f : C} {g : C'} (hf : f ∈ supLat C) (hg : g ∈ supLat C') (h : (f : Rul Xo c) * g ∈ C'') : (⟨_, h⟩ : C'') ∈ supLat C''

theorem **mem_latOn** {Z : Comp Xo c} {G : WTZ E} {f : cuspOn U Z G} : f ∈ latOn U Z G ↔ (⟨f, cuspOn_le U Z G f.2⟩ : cuspW Xo c U G) ∈ latW U G

theorem **ptN_nonneg** (la : ∀ k, Wt (X k).Ψ r) (P : FDSec X) (z : Σ k, Frame (X k).Ψ r) : 0 ≤ ptN la P z

theorem **isProbabilityMeasure_muPart** {A : Set (Σ k, Frame (X k).Ψ r)} (h : muQ1 V (classesOver V dict A) ≠ 0) : IsProbabilityMeasure (muPart V dict A)

theorem **muPart_univ** : muPart V dict Set.univ = muQ1 V

theorem **muPart_univ_le** (A : Set (Σ k, Frame (X k).Ψ r)) : muPart V dict A Set.univ ≤ 1

theorem **petE_smul** (la : ∀ k, Wt (X k).Ψ r) (a : ℂ) (P : FDSec X) : petE V dict μ la (a • P) = ENNReal.ofReal ‖a‖ * petE V dict μ la P

theorem **petE_le_supE** (hμ : μ Set.univ ≤ 1) (hd : ∀ g, (dict g).2 ∈ (X (dict g).1).dom) (la : ∀ k, Wt (X k).Ψ r) (P : FDSec X) : petE V dict μ la P ≤ supE X la P

theorem **petE_ne_top** [Fintype Cp] (hμ : μ Set.univ ≤ 1) (hd : ∀ g, (dict g).2 ∈ (X (dict g).1).dom) {la : ∀ k, Wt (X k).Ψ r} {P : FDSec X} (hP : P ∈ holSub X la) : petE V dict μ la P ≠ ⊤

theorem **bergE_le_iff** {la : ∀ k, Wt (X k).Ψ r} (hfin : ∀ P ∈ holSub X la, petE V dict μ la P ≠ ⊤) (y : FPt X) {B : ENNReal} (hB : B ≠ ⊤) : bergE V dict μ la y ≤ B ↔ ∀ P ∈ holSub X la, ENNReal.ofReal (ptN la P ⟨y.1, y.2.1⟩) ^ 2 ≤ B * petE V dict μ la P ^ 2

theorem **lieFacZ_pos** (x : ℝ) (G : WTZ E) : 0 < lieFacZ Dw x G

theorem **lieFacZ_wtZ** (x : ℝ) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : lieFacZ Dw x (wtZ a n) = lieFac Dw x a

theorem **lieFacZ_add** (x : ℝ) (G G' : WTZ E) : lieFacZ Dw x (G + G') = lieFacZ Dw x G * lieFacZ Dw x G'

theorem **homN_nonneg** (G : WTZ E) (f : Rul Xo c) (z : Σ k, Frame (U.X k).Ψ r) : 0 ≤ homN U x G f z

theorem **rdU_mul** (f g : Rul Xo c) : rdU U (f * g) = rdU U f * rdU U g

theorem **homN_mul** {G G' : WTZ E} (hG : Even G.2) (hG' : Even G'.2) (f g : Rul Xo c) (z : Σ k, Frame (U.X k).Ψ r) (hz : z.2 ∈ (U.X z.1).dom) : homN U x (G + G') (f ∗ g) z = homN U x G f z ∗ homN U x G' g z

theorem **rdU_eq_zero** {Z : Comp Xo c} {f : Rul Xo c} (hf : f ∈ suppOn Z) {z : Σ k, Frame (U.X k).Ψ r} (hz : z ∉ framesOn U Z) : rdU U f z.1 z.2 = 0

theorem **petAt_nonneg** (G : WTZ E) (f : Rul Xo c) : 0 ≤ petAt U x V dict μ G f

theorem **petAt_smul** (G : WTZ E) (a : c.F) (f : Rul Xo c) : petAt U x V dict μ G (a • f) = ‖σ a‖ ∗ petAt U x V dict μ G f

theorem **petAt_eq** (G : WTZ E) (f : Rul Xo c) : petAt U x V dict μ G f = (eLpNorm (fun q : AutQ V => homN U x G f (dict q.out)) 2 μ).toReal

theorem **homN_sq_le** {G : WTZ E} (hμ : μ Set.univ ≤ 1) (hd : ∀ g, (dict g).2 ∈ (U.X (dict g).1).dom) (y : FPt U.X) {B : ℝ} (hB0 : 0 ≤ B) (hB : bergE V dict μ (modelWt U G) y ≤ ENNReal.ofReal B) {f : Rul Xo c} (hf : rdU U f ∈ holSub U.X (modelWt U G)) : homN U x G f (y.1, y.2.1) ^ 2 ≤ B ∗ petAt U x V dict μ G f ^ 2

theorem **petAt_wtZ** {σ : c.F →+∗ ℂ} (V : HSp D) (dict : GA D V → Σ k, Frame ((U σ).X k).Ψ r) (mu : GA D V → ℂ) (hm : MuUnit V dict mu) (a : (E →+∗ ℂ) × ℕ → ℕ) (n : ℕ) (h : JMemL2 V σ (rdC U σ a n) dict mu) {f : Rul Xo c} (hf : f ∈ cusp Xo c U a n) : petAt (U σ) x V dict (muQ1 V) (wtZ a n) f = l2C U σ V dict mu x a n h f

theorem **petW_smul** (G : WTZ E) (σ : c.F →+∗ ℂ) (a : c.F) (f : Rul Xo c) : petW U x V dict G σ (a • f) = ‖σ a‖ ∗ petW U x V dict G σ f

theorem **petOn_smul** (Z : Comp Xo c) (G : WTZ E) (σ : c.F →+∗ ℂ) (a : c.F) (f : Rul Xo c) : petOn U x V dict Z G σ (a • f) = ‖σ a‖ ∗ petOn U x V dict Z G σ f

theorem **degW_eq** (G : WTZ E) : degW U x V dict G = (rankW U G : ℝ) ∗ (slopeW_sq U x V dict G / 2)

Development.AbelianVarietyTheory.Definitions.Foundations.IsRestrictedProdFinAwayIffIsProdMeasureFinAway

theorem **archFinEquiv_apply** (g : GAd σ J) : archFinEquiv σ J g = (fun w => projArch σ J w g, projFin σ J g)

theorem **archFinEquiv_symm_apply** (p : (∀ w, GArch σ J w) × GFin σ J) : (archFinEquiv σ J).symm p = ofInf σ J (embArchPi σ J p.1) ∗ ofFin σ J p.2

theorem **finPtsEquiv_apply** (y : GFin σ J) : (finPtsEquiv σ J y : GAd σ J) = ofFin σ J y

theorem **isRestrictedProd_iff** {μ : Measure (GAd σ J)} : IsRestrictedProd σ J L μa μf μ ↔ (∀ w, (μa w).IsHaarMeasure) ∧ (∀ v, (μf v).IsHaarMeasure) ∧ μ.IsHaarMeasure ∧ (∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) ∧ ∀ S : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∉ S, μf v (stabLoc σ J L v) = 1) → ∀ (A : ∀ w, Set (GArch σ J w)) (B : ∀ v, Set (GLoc σ J v)), (∀ w, MeasurableSet (A w)) → (∀ v ∈ S, MeasurableSet (B v)) → μ {g | (∀ w, projArch σ J w g ∈ A w) ∧ (∀ v ∈ S, proj σ J v g ∈ B v) ∧ ∀ v ∉ S, proj σ J v g ∈ stabLoc σ J L v} = (∏ w, μa w (A w)) ∗ ∏ v ∈ S, μf v (B v)

theorem **isRestrictedProdFin_iff** {μ : Measure (GFin σ J)} : IsRestrictedProdFin σ J L μf μ ↔ (∀ v, (μf v).IsHaarMeasure) ∧ μ.IsHaarMeasure ∧ (∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) ∧ ∀ S : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∉ S, μf v (stabLoc σ J L v) = 1) → ∀ B : ∀ v, Set (GLoc σ J v), (∀ v ∈ S, MeasurableSet (B v)) → μ {g | (∀ v ∈ S, projFinLoc σ J v g ∈ B v) ∧ ∀ v ∉ S, projFinLoc σ J v g ∈ stabLoc σ J L v} = ∏ v ∈ S, μf v (B v)

theorem **isRestrictedProdFinPts_iff** {μ : Measure (finPts σ J)} : IsRestrictedProdFinPts σ J L μf μ ↔ (∀ v, (μf v).IsHaarMeasure) ∧ μ.IsHaarMeasure ∧ (∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) ∧ ∀ S : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∉ S, μf v (stabLoc σ J L v) = 1) → ∀ B : ∀ v, Set (GLoc σ J v), (∀ v ∈ S, MeasurableSet (B v)) → μ {g | (∀ v ∈ S, proj σ J v g ∈ B v) ∧ ∀ v ∉ S, proj σ J v g ∈ stabLoc σ J L v} = ∏ v ∈ S, μf v (B v)

theorem **IsRestrictedProd.haar_arch** {μ : Measure (GAd σ J)} (h : IsRestrictedProd σ J L μa μf μ) (w : InfinitePlace F) : (μa w).IsHaarMeasure

theorem **IsRestrictedProd.haar_loc** {μ : Measure (GAd σ J)} (h : IsRestrictedProd σ J L μa μf μ) (v : HeightOneSpectrum (𝒪 F)) : (μf v).IsHaarMeasure

theorem **IsRestrictedProd.haar** {μ : Measure (GAd σ J)} (h : IsRestrictedProd σ J L μa μf μ) : μ.IsHaarMeasure

theorem **IsRestrictedProd.isCylProd** {μ : Measure (GAd σ J)} (h : IsRestrictedProd σ J L μa μf μ) : IsCylProd (fun w => projArch σ J w) (fun v => proj σ J v) (fun v => (stabLoc σ J L v : Set (GLoc σ J v))) μa μf μ

theorem **IsRestrictedProd.eventually_eq_one** {μ : Measure (GAd σ J)} (h : IsRestrictedProd σ J L μa μf μ) : ∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1

theorem **isHaarProd_iff_isRestrProd** {X W V : Type} {Ya : W → Type} {Yf : V → Type} [Group X] [∀ w, Group (Ya w)] [∀ v, Group (Yf v)] [TopologicalSpace X] [∀ w, TopologicalSpace (Ya w)] [∀ v, TopologicalSpace (Yf v)] [Fintype W] [MeasurableSpace X] [∀ w, MeasurableSpace (Ya w)] [∀ v, MeasurableSpace (Yf v)] {pa : ∀ w, X →∗ Ya w} {pf : ∀ v, X →∗ Yf v} {K : ∀ v, Subgroup (Yf v)} {μa : ∀ w, Measure (Ya w)} {μf : ∀ v, Measure (Yf v)} {μ : Measure X} : IsHaarProd pa pf K μa μf μ ↔ IsRestrProd pa pf K μa μf μ

theorem **isRestrictedProd_iff_isProdMeasure** {μ : Measure (GAd σ J)} : IsRestrictedProd σ J L μa μf μ ↔ IsProdMeasure σ J L μa μf μ

theorem **restrGroup_GFin** : RestrMeas.IsRestrGroup (noArch (GFin σ J)) (projFinLoc σ J) (stabLoc σ J L)

theorem **restrGroup_GAd** : RestrMeas.IsRestrGroup (projArch σ J) (proj σ J) (stabLoc σ J L)

theorem **restrGroup_finPts** : RestrMeas.IsRestrGroup (noArch (finPts σ J)) (fun v => (proj σ J v).comp (finPts σ J).subtype) (stabLoc σ J L)

theorem **restrGroup_awayPts** (S : Finset (HeightOneSpectrum (𝒪 F))) : RestrMeas.IsRestrGroup (fun w => (projArch σ J w).comp (awayPts σ J S).subtype) (fun v : {v // v ∉ S} => (proj σ J v).comp (awayPts σ J S).subtype) fun v => stabLoc σ J L v

theorem **existsUnique_isRestrictedProd** (h : ∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) : ∃! μ, IsRestrictedProd σ J L μa μf μ

theorem **existsUnique_isRestrictedProdFin** (h : ∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) : ∃! μ, IsRestrictedProdFin σ J L μf μ

theorem **existsUnique_isRestrictedProdFinPts** (h : ∀ᶠ v in cofinite, μf v (stabLoc σ J L v) = 1) : ∃! μ, IsRestrictedProdFinPts σ J L μf μ

theorem **existsUnique_isRestrictedProdAway** (h : ∀ᶠ v in cofinite, νf v (stabLoc σ J L v) = 1) : ∃! μ, IsRestrictedProdAway σ J L S μa νf μ

theorem **exists_isRestrictedProd** : ∃ μ, IsRestrictedProd σ J L (fun _ => haar) (haarLoc σ J L) μ

theorem **exists_isRestrictedProd_of_haar** (μ : Measure (GAd σ J)) [μ.IsHaarMeasure] : ∃ μa, IsRestrictedProd σ J L μa (haarLoc σ J L) μ

theorem **IsRestrictedProd.unique** (h : IsRestrictedProd σ J L μa μf μ) (h' : IsRestrictedProd σ J L μa μf μ') : μ = μ'

theorem **IsRestrictedProd.smul** (h : IsRestrictedProd σ J L μa μf μ) (c : InfinitePlace F → ℝ≥0) (d : HeightOneSpectrum (𝒪 F) → ℝ≥0) (hc : ∀ w, c w ≠ 0) (hd0 : ∀ v, d v ≠ 0) (T : Finset (HeightOneSpectrum (𝒪 F))) (hd : ∀ v ∉ T, d v = 1) : IsRestrictedProd σ J L (fun w =>

c w • μa w) (fun v => d v • μf v) ((((∏ w, c w) * ∏ v ∈ T, d v) • μ)

theorem **IsRestrictedProd.smul_arch** [DecidableEq (InfinitePlace F)] (h : IsRestrictedProd σ J L μa μf μ) (w : InfinitePlace F) {x : ℝ≥0} (hx : x ≠ 0) : IsRestrictedProd σ J L (Function.update μa w (x • μa w)) μf (x • μ)

theorem **IsRestrictedProd.measure_level** (h : IsRestrictedProd σ J L μa μf μ) {A : ∀ w, Set (GArch σ J w)} (hA : ∀ w, MeasurableSet (A w)) : μ {g | (∀ w, projArch σ J w g ∈ A w) ∧ g ∈ levelOpen σ J L} = (∏ w, μa w (A w)) * ∏' v, μf v (stabLoc σ J L v)

theorem **IsRestrictedProdFin.measure_level** {ν : Measure (GFin σ J)} (h : IsRestrictedProdFin σ J L μf ν) : ν (levelOffFin σ J L ∅) = ∏' v, μf v (stabLoc σ J L v)

theorem **IsRestrictedProdFinAway.measure_level** {ν : Measure (finAwayPts σ J S)} (h : IsRestrictedProdFinAway σ J L S νf ν) : ν {g | g.1 ∈ levelOff σ J L S} = ∏' v : {v // v ∉ S}, νf v (stabLoc σ J L v)

private theorem **atAwayContEquiv_symm_apply** (p : (∀ v : S, GLoc σ J v) × awayPts σ J S) : (atAwayContEquiv σ J S).symm p = embAt σ J S p.1 * p.2

theorem **IsRestrictedProdAway.prod** [∀ v, (μf v).IsHaarMeasure] {ν : Measure (awayPts σ J S)} (h : IsRestrictedProdAway σ J L S μa (fun v => μf v) ν) : IsRestrictedProd σ J L μa μf (((Measure.pi fun v : S => μf v).prod ν).map (atAwayContEquiv σ J S).symm)

theorem **map_map_symm** {X Y : Type} [Group X] [Group Y] [TopologicalSpace X] [TopologicalSpace Y] [MeasurableSpace X] [MeasurableSpace Y] [BorelSpace X] [BorelSpace Y] (e : X ≃ₜ* Y) (m : Measure Y) : (m.map e.symm).map e = m

theorem **IsRestrictedProd.map_atAway** (h : IsRestrictedProd σ J L μa μf μ) {ν : Measure (awayPts σ J S)} (h' : IsRestrictedProdAway σ J L S μa (fun v => μf v) ν) : μ.map (atAwayContEquiv σ J S) = (Measure.pi fun v : S => μf v).prod ν

theorem **IsRestrictedProd.exists_away** (h : IsRestrictedProd σ J L μa μf μ) (S : Finset (HeightOneSpectrum (𝒪 F))) : ∃ ν, IsRestrictedProdAway σ J L S μa (fun v => μf v) ν ∧ μ.map (atAwayContEquiv σ J S) = (Measure.pi fun v : S => μf v).prod ν

theorem **IsRestrictedProdFin.prod** [∀ w, (μa w).IsHaarMeasure] {ν : Measure (GFin σ J)} (h : IsRestrictedProdFin σ J L μf ν) : IsRestrictedProd σ J L μa μf (((Measure.pi μa).prod ν).map (archFinEquiv σ J).symm)

theorem **IsRestrictedProd.map_archFin** (h : IsRestrictedProd σ J L μa μf μ) {ν : Measure (GFin σ J)} (h' : IsRestrictedProdFin σ J L μf ν) : μ.map (archFinEquiv σ J) = (Measure.pi μa).prod ν

theorem **IsRestrictedProdFin.map_finPts** {ν : Measure (GFin σ J)} (h : IsRestrictedProdFin σ J L μf ν) : IsRestrictedProdFinPts σ J L μf (ν.map (finPtsEquiv σ J))

theorem **IsRestrictedProd.isMulRightInvariant** (h : IsRestrictedProd σ J L μa μf μ) [∀ w, (μa w).IsMulRightInvariant] [∀ v, (μf v).IsMulRightInvariant] : μ.IsMulRightInvariant

Development.AbelianVarietyTheory.Definitions.Foundations.IsRootOne

theorem **isRoot_one** (X : Tower Ω ε) (m : ℕ) (h : ((fun _ => 1 : ℂ → ℂ) : Mero) ≠ 0) : IsRoot X m 1 (fun _ => 1) (fun _ => 1)

Development.AbelianVarietyTheory.Definitions.Foundations.IsSatVecUOfIsSphericalU

theorem **finite_heckeVec** (hK : IsCommensurated K) (hh : h ∈ Hecke ℂ K) (ρ : Representation ℂ G V) (x : V) : (Function.support fun c : G ⧸ K => h c.out • ρ c.out x).Finite

theorem **heckeVec_eq_of_section** (hh : IsBiInv K h) (hx : x ∈ fixed ρ K) (s : G ⧸ K → G) (hs : ∀ c, (s c : G ⧸ K) = c) : heckeVec ρ K h x = ∑' c, h (s c) • ρ (s c) x

theorem **heckeVec_dosetInd** (hx : x ∈ fixed ρ K) {a : G} {R : Finset G} (hR : ∀ γ ∈ R, γ ∈ DoubleCoset.doubleCoset a (K : Set G) K) (hR' : ∀ g ∈ DoubleCoset.doubleCoset a (K : Set G) K, ∃! γ, γ ∈ R ∧ γ⁻¹ * g ∈ K) : heckeVec ρ K (dosetInd ℂ K a) x = ∑ γ ∈ R, ρ γ x

theorem **heckeVec_rightReg** (hK : IsCommensurated K) (hh : h ∈ Hecke ℂ K) (φ : G → W) : heckeVec (rightReg ℂ G W) K h φ = hAct K h φ

theorem **map_heckeVec** (hK : IsCommensurated K) (hh : h ∈ Hecke ℂ K) {ρ' : Representation ℂ G W} (f : V →ₗ[ℂ] W) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (x : V) : f (heckeVec ρ K h x) = heckeVec ρ' K h (f x)

theorem **heckeVec_comp_symm** (e : G ≃* H) {K' : Subgroup H} (hK : ∀ g, g ∈ K ↔ e g ∈ K') {h' : H → ℂ} (hh : IsBiInv K' h') (hx : x ∈ fixed ρ K) : heckeVec (ρ.comp e.symm.toMonoidHom) K' h' x = heckeVec ρ K (h' ∘ e) x

theorem **mem_fixed_comp_symm** (e : G ≃* H) {K' : Subgroup H} (hK : ∀ g, g ∈ K ↔ e g ∈ K') : x ∈ fixed (ρ.comp e.symm.toMonoidHom) K' ↔ x ∈ fixed ρ K

theorem **heckeVec_comp_symm_congr** (e e' : G ≃* H) {K' : Subgroup H} (hK : ∀ g, g ∈ K ↔ e g ∈ K') {k : H} (hk : k ∈ K') (he : ∀ g, e' g = k⁻¹ * e g * k) {h' : H → ℂ} (hh : IsBiInv K' h') (hx : x ∈ fixed ρ K) : heckeVec (ρ.comp e'.symm.toMonoidHom) K' h' x = heckeVec (ρ.comp e.symm.toMonoidHom) K' h' x

theorem **eq_of_satEig_eq** {q n : ℕ} (hq : q ≠ 0) {B B' : Multiset ℂˣ} (hB : B.card = n) (hB' : B'.card = n) (h : ∀ j ≤ n, satEig q n B j = satEig q n B' j) : B = B'

theorem **exists_isUnifV** : ∃ ϖ, IsUnifV E w ϖ

theorem **qV_ne_zero** : qV E w ≠ 0

theorem **mem_Ov_of_isUnifV** {ϖ : w.adicCompletion E} (h : IsUnifV E w ϖ) : ϖ ∈ Ov E w

theorem **Tj_mem_hecke** {ϖ : w.adicCompletion E} (h : IsUnifV E w ϖ) (j : ℕ) : Tj ℂ n (Ov E w) ϖ j ∈ Hecke ℂ (KGL w n)

theorem **IsSatVecGL.unique** (h : IsSatVecGL w ρ x B) (h' : IsSatVecGL w ρ x B') : B = B'

theorem **HasSatGL.exists_vec** (h : HasSatGL w ρ B) : ∃ x, IsSatVecGL w ρ x B

theorem **HasSatGL.unique** (h : HasSatGL w ρ B) (h' : HasSatGL w ρ B') : B = B'

theorem **HasSatGL.card** (h : HasSatGL w ρ B) : B.card = n

theorem **IsSatVecGL.map** (h : IsSatVecGL w ρ x B) {ρ' : Representation ℂ (GL (Fin n) (w.adicCompletion E)) W} (f : V →ₗ[ℂ] W) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (h0 : f x ≠ 0) : IsSatVecGL w ρ' (f x) B

theorem **IsSatVecGL.of_map** {ρ' : Representation ℂ (GL (Fin n) (w.adicCompletion E)) W} (f : V →ₗ[ℂ] W) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (hi : Function.Injective f) (h : IsSatVecGL w ρ' (f x) B) : IsSatVecGL w ρ x B

theorem **isUpperB_one** : IsUpperB E w (1 : GL (Fin n) (w.adicCompletion E))

theorem **chiDelta_one** (α : Fin n → ℂ) : chiDelta E w α (1 : GL (Fin n) (w.adicCompletion E)) = 1

theorem **isSatVecGL_of_isSpherical** {α : Fin n → ℂˣ} {φ : GL (Fin n) (w.adicCompletion E) → ℂ} (h : Formalism.UG.IsSpherical E w (fun i => (α i : ℂ)) φ) (h0 : φ ≠ 0) : IsSatVecGL w (rightReg ℂ _ ℂ) φ (Finset.univ.val.map α)

theorem **hasSatGL_psFunGL** (α : Fin n → ℂˣ) : HasSatGL w (psFunGL w α) (Finset.univ.val.map α)

theorem **HasSatU.exists_vec** (h : HasSatU ϖ q ρ α) : ∃ x, IsSatVecU ϖ q ρ x α

theorem **IsSatVecU.map** (hU : IsUnramInv ϖ σ q) (h : IsSatVecU ϖ q ρ x α) {ρ' : Representation ℂ (GU σ n) W} (f : V →ₗ[ℂ] W) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (h0 : f x ≠ 0) : IsSatVecU ϖ q ρ' (f x) α

theorem **IsSatVecU.of_map** (hU : IsUnramInv ϖ σ q) {ρ' : Representation ℂ (GU σ n) W} (f : V →ₗ[ℂ] W) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (hi : Function.Injective f) (h : IsSatVecU ϖ q ρ' (f x) α) : IsSatVecU ϖ q ρ x α

theorem **isBorelU_one** : IsBorelU (1 : GU σ n)

theorem **chiDeltaU_one** {m : ℕ} (q : ℕ) (α : Fin m → ℂ) : chiDeltaU q α (1 : GU σ n) = 1

theorem **satU_indep** (hU : IsUnramInv ϖ σ q) (hU' : IsUnramInv ϖ' σ q) {h : GU σ n → ℂ} (hh : h ∈ Hecke ℂ (KU σ n)) (α : Fin (n / 2) → ℂˣ) : SatU ϖ q h (fun i => (α i : ℂ)) = SatU ϖ' q h (fun i => (α i : ℂ))

theorem **IsSatVecU.of_unif** (hU : IsUnramInv ϖ σ q) (hU' : IsUnramInv ϖ' σ q) (h : IsSatVecU ϖ q ρ x α) : IsSatVecU ϖ' q ρ x α

theorem **isSatVecU_of_isSphericalU** (hU : IsUnramInv ϖ σ q) {φ : GU σ n → ℂ} (h : IsSphericalU q (fun i => (α i : ℂ)) φ) (h0 : φ ≠ 0) : IsSatVecU ϖ q (rightReg ℂ _ ℂ) φ α

theorem **hasSatU_psFunU** (hU : IsUnramInv ϖ σ q) (α : Fin (n / 2) → ℂˣ) : HasSatU ϖ q (psFunU σ q α) α

Development.AbelianVarietyTheory.Definitions.Foundations.IsScalarTowerMdlOm

theorem **isScalarTower_mdl_om** (X : Tower Ω ε) (Γ' : Subgroup SL(2, ℤ)) (S : Subgroup (Ω ≃ₐ[ℚ] Ω)) : @IsScalarTower (mdl X Γ' S) (lev X Γ') (Om X) Algebra.toSMul Algebra.toSMul Algebra.toSMul

theorem **cR_mod** (l N : ℕ) [Fact l.Prime] (hN : (N : ZMod l) ≠ 0) (r₀ : ZMod l) : ((cR l N r₀ : ℤ) : ZMod l) = -r₀

theorem **_root_.GlobalTheory.ZetaB.eval_clsHat** (X : _root_.GlobalTheory.KZ.Tower Ω ε) (l : ℕ) [Fact l.Prime] (n : ℕ) (a : X.K) : AdicCompletion.eval (_root_.GlobalTheory.KZ.idl l) (X.Kl l) n (_root_.GlobalTheory.ZetaB.clsHat X l a) = X.cls l n (_root_.GlobalTheory.ZetaB.unitOf X a)

theorem **_root_.GlobalTheory.ZetaB.unitOf_val** (X : _root_.GlobalTheory.KZ.Tower Ω ε) {a : X.K} (ha : a ≠ 0) : ((_root_.GlobalTheory.ZetaB.unitOf X a : (X.K)ˣ) : X.K) = a

theorem **_root_.GlobalTheory.ZetaB.unitOf_zero** (X : _root_.GlobalTheory.KZ.Tower Ω ε) : _root_.GlobalTheory.ZetaB.unitOf X 0 = 1

theorem **vecMul_gR** (l N : ℕ) [Fact l.Prime] (hN : (N : ZMod l) ≠ 0) (r₀ : ZMod l) : ![-r₀, -1] ᵥ* (((gR l N r₀ : SL(2, ℤ)) : Matrix (Fin 2) (Fin 2) ℤ).map (Int.cast : ℤ → ZMod l)) = ![0, -1]

Development.AbelianVarietyTheory.Definitions.Foundations.IsTopWTopWOfExists

theorem **mem_geo** (g : Ω ≃ₐ[F] Ω) : g ∈ geo F K Ω ↔ ∀ z : K, g (algebraMap K Ω z) = algebraMap K Ω z

theorem **conj_mem_geo** [Normal F K] (σ : Ω ≃ₐ[F] Ω) {h : Ω ≃ₐ[F] Ω} (hh : h ∈ geo F K Ω) : σ * h * σ⁻¹ ∈ geo F K Ω

theorem **exists_of_fixed** [IsGalois K Ω] {z : Ω} (hz : ∀ g ∈ geo F K Ω, g z = z) : ∃ c : K, algebraMap K Ω c = z

theorem **apply_eq_kum** [NeZero N] (hζ : IsPrimitiveRoot ζ N) {θ : Ω} {w : K} (hw0 : w ≠ 0) (hw : θ ^ N = algebraMap K Ω w) {g : Ω ≃ₐ[F] Ω} (hg : g ∈ geo F K Ω) : g θ = ζ ^ (kum (K

theorem **pow_val_add** [NeZero N] (hζ : IsPrimitiveRoot ζ N) (a b : ZMod N) : ζ ^ (a + b).val = ζ ^ a.val * ζ ^ b.val

theorem **pow_val_mul** [NeZero N] (hζ : IsPrimitiveRoot ζ N) (a b : ZMod N) : ζ ^ (a * b).val = (ζ ^ b.val) ^ a.val

theorem **edge_mul_conj** {Γ : Type*} [Group Γ] (x y : Γ →* Multiplicative (ZMod N)) (η : Γ → ZMod N) (σ τ h : Γ) : edge x y η (σ * τ) h = edge x y η σ (τ * h * τ⁻¹) + edge x y η τ h

theorem **edge_inv** {Γ : Type*} [Group Γ] (x y : Γ →* Multiplicative (ZMod N)) (η : Γ → ZMod N) (σ h : Γ) : edge x y η σ⁻¹ h = η (σ⁻¹ * h * σ) - η h + (x h).toAdd * (y σ).toAdd - (x σ).toAdd * (y h).toAdd

theorem **isTopW_topW_of_exists** {σ : Ω ≃ₐ[F] Ω} (h : ∃ w : K, IsTopW ζ x y η σ w) : IsTopW ζ x y η σ (topW (K

theorem **topW_ne_zero** (σ : Ω ≃ₐ[F] Ω) : topW (K

theorem **unique** [NeZero N] [IsGalois K Ω] (hζ : IsPrimitiveRoot ζ N) {σ : Ω ≃ₐ[F] Ω} {w v : K} (hw : IsTopW ζ x y η σ w) (hv : IsTopW ζ x y η σ v) : ∃ c : K, c ≠ 0 ∧ w = v * c ^ N

Development.AbelianVarietyTheory.Definitions.Foundations.IsUnitaryOfOrder

theorem **exists_unif** (v : HeightOneSpectrum (𝒪 L)) : ∃ k : Lˣ, IsUnifAt v (atFin v k)

theorem **unif_spec** (v : HeightOneSpectrum (𝒪 L)) : IsUnifAt v (atFin v (unif v))

theorem **unrAt_iff** (v : HeightOneSpectrum (𝒪 L)) : χ.UnrAt v ↔ χ.LevelLe v 0

theorem **val_eq** {v : HeightOneSpectrum (𝒪 L)} (h : χ.UnrAt v) (x : (v.adicCompletion L)ˣ) (hx : IsUnifAt v x) : χ.loc v x = χ.val v

@[simp] theorem **loc_mul** (χ χ' : HChar L) (v : HeightOneSpectrum (𝒪 L)) (x) : (χ * χ').loc v x = χ.loc v x * χ'.loc v x

@[simp] theorem **loc_inv** (χ : HChar L) (v : HeightOneSpectrum (𝒪 L)) (x) : (χ⁻¹).loc v x = (χ.loc v x)⁻¹

@[simp] theorem **loc_one** (v : HeightOneSpectrum (𝒪 L)) (x) : (1 : HChar L).loc v x = 1

@[simp] theorem **locInf_mul** (χ χ' : HChar L) (w : InfinitePlace L) (x) : (χ * χ').locInf w x = χ.locInf w x * χ'.locInf w x

@[simp] theorem **locInf_inv** (χ : HChar L) (w : InfinitePlace L) (x) : (χ⁻¹).locInf w x = (χ.locInf w x)⁻¹

@[simp] theorem **val_mul** (χ χ' : HChar L) (v : HeightOneSpectrum (𝒪 L)) : (χ * χ').val v = χ.val v * χ'.val v

@[simp] theorem **val_inv** (χ : HChar L) (v : HeightOneSpectrum (𝒪 L)) : (χ⁻¹).val v = (χ.val v)⁻¹

@[simp] theorem **inf_mul** (χ χ' : HChar L) (k : Lˣ) : (χ * χ').inf k = χ.inf k * χ'.inf k

@[simp] theorem **inf_inv** (χ : HChar L) (k : Lˣ) : (χ⁻¹).inf k = (χ.inf k)⁻¹

theorem **finite_aboveF** (v : HeightOneSpectrum (𝒪 F)) : {P : HeightOneSpectrum (𝒪 L) | P.under (𝒪 F) = v}.Finite

theorem **resFin_apply** (χ : HChar L) (v : HeightOneSpectrum (𝒪 F)) (k : Fˣ) : χ.resFin v k = ∏ P ∈ aboveF (L

theorem **resInf_apply** (χ : HChar L) (w : InfinitePlace F) (k : Fˣ) : χ.resInf w k = ∏ w' ∈ aboveInf (L

theorem **resFin_mul** (χ χ' : HChar L) (v : HeightOneSpectrum (𝒪 F)) (k : Fˣ) : (χ * χ').resFin v k = χ.resFin v k * χ'.resFin v k

theorem **resFin_inv** (χ : HChar L) (v : HeightOneSpectrum (𝒪 F)) (k : Fˣ) : (χ⁻¹).resFin v k = (χ.resFin v k)⁻¹

theorem **resInf_mul** (χ χ' : HChar L) (w : InfinitePlace F) (k : Fˣ) : (χ * χ').resInf w k = χ.resInf w k * χ'.resInf w k

theorem **resInf_inv** (χ : HChar L) (w : InfinitePlace F) (k : Fˣ) : (χ⁻¹).resInf w k = (χ.resInf w k)⁻¹

theorem **TrivOn.mul** {χ χ' : HChar L} (h : χ.TrivOn F) (h' : χ'.TrivOn F) : (χ * χ').TrivOn F

theorem **TrivOn.inv** {χ : HChar L} (h : χ.TrivOn F) : (χ⁻¹).TrivOn F

theorem **ResEta.mul** {χ ξ : HChar L} (h : χ.ResEta F) (h' : ξ.TrivOn F) : (χ * ξ).ResEta F

theorem **ResEta.div** {χ χ' : HChar L} (h : χ.ResEta F) (h' : χ'.ResEta F) : (χ' * χ⁻¹).TrivOn F

theorem **LevelLe.mul** {χ χ' : HChar L} {v : HeightOneSpectrum (𝒪 L)} {n : ℕ} (h : χ.LevelLe v n) (h' : χ'.LevelLe v n) : (χ * χ').LevelLe v n

theorem **LevelLe.inv** {χ : HChar L} {v : HeightOneSpectrum (𝒪 L)} {n : ℕ} (h : χ.LevelLe v n) : (χ⁻¹).LevelLe v n

theorem **UnrAt.mul** {χ χ' : HChar L} {v : HeightOneSpectrum (𝒪 L)} (h : χ.UnrAt v) (h' : χ'.UnrAt v) : (χ * χ').UnrAt v

theorem **UnrAt.inv** {χ : HChar L} {v : HeightOneSpectrum (𝒪 L)} (h : χ.UnrAt v) : (χ⁻¹).UnrAt v

theorem **UnrAt.levelLe** {χ : HChar L} {v : HeightOneSpectrum (𝒪 L)} (h : χ.UnrAt v) (n : ℕ) : χ.LevelLe v n

theorem **IsUnitary.mul** {χ χ' : HChar L} (h : χ.IsUnitary) (h' : χ'.IsUnitary) : (χ * χ').IsUnitary

theorem **IsUnitary.inv** {χ : HChar L} (h : χ.IsUnitary) : (χ⁻¹).IsUnitary

theorem **HasInf.mul_triv** {I : Type} [Fintype I] {φ : I → L →+* ℂ} {n : I → ℤ} {χ ξ : HChar L} (h : χ.HasInf φ n) (h' : ∀ k, ξ.inf k = 1) : (χ * ξ).HasInf φ n

theorem **HasInf.div** {I : Type} [Fintype I] {φ : I → L →+* ℂ} {n : I → ℤ} {χ χ' : HChar L} (h : χ.HasInf φ n) (h' : χ'.HasInf φ n) (k : Lˣ) : (χ' * χ⁻¹).inf k = 1

theorem **HasInfQ.mul_triv** {I : Type} [Fintype I] {φ : I → L →+* ℂ} {a : I → ℚ} {χ ξ : HChar L} (h : χ.HasInfQ φ a) (h' : ∀ k, ξ.inf k = 1) : (χ * ξ).HasInfQ φ a

theorem **HasInfQ.div** {I : Type} [Fintype I] {φ : I → L →+* ℂ} {a : I → ℚ} {χ χ' : HChar L} (h : χ.HasInfQ φ a) (h' : χ'.HasInfQ φ a) (k : Lˣ) : (χ' * χ⁻¹).inf k = 1

theorem **isClsF_one** : IsClsF H (1 : HChar H.E)

theorem **IsEtaT.mul_cls** {χ ξ : HChar H.E} (h : IsEtaT H χ) (hξ : IsClsF H ξ) : IsEtaT H (χ * ξ)

theorem **IsEtaT.div_cls** {χ χ' : HChar H.E} (h : IsEtaT H χ) (h' : IsEtaT H χ') : IsClsF H (χ' * χ⁻¹)

theorem **IsChiPsi.mul_cls** {χ ξ : HChar H.E} (h : IsChiPsi H χ) (hξ : IsClsF H ξ) : IsChiPsi H (χ * ξ)

theorem **IsChiPsi.div_cls** {χ χ' : HChar H.E} (h : IsChiPsi H χ) (h' : IsChiPsi H χ') : IsClsF H (χ' * χ⁻¹)

theorem **IsEtaT.sq_chiPsi** {χ : HChar H.E} (h : IsEtaT H χ) : IsChiPsi H (χ * χ) where

theorem **PiBar.overP** (P : PiBar H) {k : Fin 2} {w : HeightOneSpectrum (𝒪 H.E)} (hw : P.Div k w) : OverP H (H.g k) w

theorem **isUnitAt_nat** {p a : ℕ} {w : HeightOneSpectrum (𝒪 H.E)} (hw : OverP H p w) (ha : a.Coprime p) {x : H.Eˣ} (hx : (x : H.E) = (a : H.E)) : IsUnitAt w (atFin w x)

theorem **IsMu0.mul_cls** {P : PiBar H} {W : Omg H} {e : R.Psi → ℚ} {j : ℕ} {χ ξ : HChar H.E} (h : IsMu0 P W e j χ) (hξ : IsClsF H ξ) : IsMu0 P W e j (χ * ξ) where

theorem **IsMu0.div_cls** {P : PiBar H} {W : Omg H} {e : R.Psi → ℚ} {j : ℕ} {χ χ' : HChar H.E} (h : IsMu0 P W e j χ) (h' : IsMu0 P W e j χ') (hg : ∀ k v, OverP H (H.g k) v → (χ' * χ⁻¹).UnrAt v) : IsClsF H (χ' * χ⁻¹) where

theorem **isPsi_one** (l : ℕ) : IsPsi (H

theorem **HChar.isUnitary_of_order** {L : Type} [Field L] [NumberField L] {χ : HChar L} {n : ℕ} (hn : 0 < n) (h : χ.OrderDvd n) : χ.IsUnitary

theorem **IsDelta.unitary** {p : ℕ} {χ : HChar H.E} (h : IsDelta (H

Development.AbelianVarietyTheory.Definitions.Foundations.KumValUnique

theorem **kumVal_unique** {n : ℕ} {ζ u : Ω} {s : G ℚ Ω} {k k' : ZMod (l ^ n)} (hζ : IsPrimitiveRoot ζ (l ^ n)) (hs : s ζ = ζ) (hu : u ≠ 0) (hk : KumVal n ζ u s k) (hk' : KumVal n ζ u s k') : k = k'

theorem **kumVal_one** (n : ℕ) (ζ : Ω) (s : G ℚ Ω) : KumVal (l

theorem **kumPres_empty** (A : ValuationSubring Ω) {n r : ℕ} {ζ : Ω} (hζ : IsPrimitiveRoot ζ (l ^ n)) (hnr : n ≤ r) (Z : G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l) (hZ : ∀ s ∈ Ine (F

theorem **fix_of_cycMod** [∀ i, HasEnoughRootsOfUnity Ω (l ^ i)] {n r : ℕ} (hnr : n ≤ r) {s : G ℚ Ω} (hs : cycMod ℚ Ω l r s = 1) {t : Ω} (ht : t ^ (l ^ n) = 1) : s t = t

Development.AbelianVarietyTheory.Definitions.Foundations.LiftApplyOfMemAdmissibleSubalgebra

theorem **iota_mem_admissibleSubalgebra** (R : Type) [CommRing R] {N l : ℕ} {T : Letter} (hT : admitted N l T) : FreeAlgebra.ι R T ∈ admissibleLettersSubalgebra R N l

theorem **lift_apply_of_mem_admissibleSubalgebra** {R M : Type} [CommRing R] [AddCommGroup M] [Module R M] {N l : ℕ} {f : Letter → Module.End R M} {lam : Letter → R} {x : M} (hx : ∀ T, admitted N l T → f T x = lam T • x) {P : FreeAlgebra R Letter} (hP : P ∈ admissibleLettersSubalgebra R N l) : FreeAlgebra.lift R f P x = FreeAlgebra.lift R lam P • x

theorem **SSPotB_iff_SSPotA** {N l w : ℕ} {lam : Letter → ℂ} : SSPotB N l w lam ↔ SSPotA N l w lam

theorem **Tower.SSBB_iff_SSBA** (X : Tower Ω ε) {N l w : ℕ} [Fact l.Prime] {E : Type} [Field E] [Algebra ℤ_[l] E] {lam : Letter → E} : X.SSBB N l w E lam ↔ X.SSBA N l w E lam

Development.AbelianVarietyTheory.Definitions.Foundations.LiftSLMapMulInvMem

theorem **eval_liftK** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (n : ℕ) (a : X.Kl l ⧸ ((idl l) ^ n • (⊤ : Submodule ℤ_[l] (X.Kl l)))) : AdicCompletion.eval (idl l) (X.Kl l) n (liftK X l n a) = a

theorem **eval_clsK** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (n : ℕ) {w : X.K} (h : w ≠ 0) : AdicCompletion.eval (idl l) (X.Kl l) n (clsK X l n w) = X.cls l n (Units.mk0 w h)

theorem **clsK_zero** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (n : ℕ) : clsK X l n 0 = 0

theorem **cls_mul** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (n : ℕ) (f g : X.Kˣ) : X.cls l n (f * g) = X.cls l n f + X.cls l n g

theorem **map_liftSL** (M : ℕ) (g : SL(2, ZMod M)) : Matrix.SpecialLinearGroup.map (Int.castRingHom (ZMod M)) (liftSL M g) = g

theorem **liftSL_map_mul_inv_mem** (M : ℕ) (γ : SL(2, ℤ)) : liftSL M (Matrix.SpecialLinearGroup.map (Int.castRingHom (ZMod M)) γ) * γ⁻¹ ∈ CongruenceSubgroup.Gamma M

theorem **exists_place_over** (Ω : Type) [Field Ω] [Algebra ℚ Ω] (l : ℕ) [Fact l.Prime] : ∃ A : ValuationSubring Ω, A.valuation (l : Ω) < 1

theorem **cycMod_surjective** (Ω : Type) [Field Ω] [Algebra ℚ Ω] [IsAlgClosure ℚ Ω] (l : ℕ) [Fact l.Prime] (r : ℕ) : Function.Surjective (cycMod ℚ Ω l r)

theorem **cycMod_repC** [IsAlgClosure ℚ Ω] (l : ℕ) [Fact l.Prime] (r : ℕ) (x : (ZMod (l ^ r))ˣ) : cycMod ℚ Ω l r (repC (Ω

theorem **rowSection_e₁** (l : ℕ) [Fact l.Prime] (u : Fin 2 → ℤ_[l]) : (![1, 0] : Fin 2 → ℤ_[l]) ⬝ᵥ u = u 0

theorem **rowSection_neg_e₂** (l : ℕ) [Fact l.Prime] (u : Fin 2 → ℤ_[l]) : (![0, -1] : Fin 2 → ℤ_[l]) ⬝ᵥ u = -u 1

theorem **galB_section_zero** (X : Tower Ω ε) (l : ℕ) [Fact l.Prime] (s : Ω ≃ₐ[ℚ] Ω) (w : ℕ) (a : X.Khat l) : X.galB l s (fun u => (u 0) ^ w • a) = fun u => (u 0) ^ w • X.kmap l (X.gal s) a

Development.AbelianVarietyTheory.Definitions.Foundations.MapEvalOfFixCoeff

theorem **HTPres.eval_ne_zero** {A : ValuationSubring Ω} {c : Coef l} {m : ℕ} {Z : ∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l} {ζ : Ω} {ι : Type} [Fintype ι] {P : ι → Polynomial Ω} {o : ι → Qbar l} (h : HTPres A c m Z ζ P o) (i : ι) : (P i).eval ζ ≠ 0

theorem **HTPres.ne_zero** {A : ValuationSubring Ω} {c : Coef l} {m : ℕ} {Z : ∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l} {ζ : Ω} {ι : Type} [Fintype ι] {P : ι → Polynomial Ω} {o : ι → Qbar l} (h : HTPres A c m Z ζ P o) (i : ι) : P i ≠ 0

theorem **HTLawAt.mono** {A : ValuationSubring Ω} {c : Coef l} {Z : ∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l} {j : Ω →+* Qbar l} {m₀ m₁ c₀ : ℕ} {Rhs : ∀ m : ℕ, DirichletCharacter (Qbar l) (l ^ m) → Qbar l → Qbar l} (h : HTLawAt A c Z j m₀ c₀ Rhs) (hm : m₀ ≤ m₁) : HTLawAt A c Z j m₁ c₀ Rhs

theorem **tval_coe** (Q : Polynomial (Qbar l)) {N : ℕ} (hN : Q.natDegree < N) (z : Qbar l) : tval (Q : PowerSeries (Qbar l)) N z = Q.eval (z - 1)

theorem **tval_dirac** {y N : ℕ} (hN : y < N) (z : Qbar l) : tval ((1 + PowerSeries.X) ^ y) N z = z ^ y

theorem **mass_dirac** {r : ℕ} {ξ : Qbar l} (hξ : IsPrimitiveRoot ξ (l ^ r)) {x y : ℕ} (hx : x < l ^ r) (hy : y < l ^ r) : mass ((1 + PowerSeries.X) ^ y) r ξ x = if x = y then ((l : ℕ) : Qbar l) ^ r else 0

Development.AbelianVarietyTheory.Definitions.Foundations.MeasurePreservingRmulQuotMeas

theorem **IsQuotMeasure.quotMeas** {ν : Measure (Quot T)} (h : IsQuotMeasure T dt dh ν) : IsQuotMeasure T dt dh (quotMeas T dt dh)

theorem **exists_out_eq** (x : H) : ∃ t : T, (mk T x).out = t * x

theorem **measurable_descend** {X : Type*} [MeasurableSpace X] {φ : H → X} (h : LeftInv T φ) (hm : Measurable φ) : Measurable (descend T φ)

theorem **IsQuotMeasure.smul_left** {ν : Measure (Quot T)} (h : IsQuotMeasure T dt dh ν) (c : ℝ≥0∞) : IsQuotMeasure T dt (c • dh) (c • ν)

theorem **IsQuotMeasure.smul_right** {ν : Measure (Quot T)} (h : IsQuotMeasure T dt dh ν) {c : ℝ≥0∞} (h0 : c ≠ 0) (ht : c ≠ ∞) : IsQuotMeasure T (c • dt) dh (c⁻¹ • ν)

theorem **leftInv_lintegral** [dt.IsMulRightInvariant] (f : H → ℝ≥0∞) : LeftInv T fun x => ∫⁻ t : T, f (t * x) ∂dt

theorem **leftInv_integral** [dt.IsMulRightInvariant] (f : H → ℂ) : LeftInv T fun x => ∫ t : T, f (t * x) ∂dt

theorem **measurable_fiber** [SFinite dt] {f : H → ℝ≥0∞} (hf : Measurable f) : Measurable fun x => ∫⁻ t : T, f (t * x) ∂dt

theorem **IsQuotMeasure.eq_map** {ν : Measure (Quot T)} (h : IsQuotMeasure T dt dh ν) {b : H → ℝ≥0∞} (hb : Measurable b) (h1 : ∀ x, ∫⁻ t : T, b (t * x) ∂dt = 1) : ν = (dh.withDensity b).map (mk T)

theorem **exists_bruhat** (hT : IsClosed (T : Set H)) (dt : Measure T) [dt.IsHaarMeasure] [dt.IsMulRightInvariant] : ∃ b : H → ℝ≥0∞, Measurable b ∧ ∀ x, ∫⁻ t : T, b (t * x) ∂dt = 1

theorem **IsQuotMeasure.unique** {ν ν' : Measure (Quot T)} (h : IsQuotMeasure T dt dh ν) (h' : IsQuotMeasure T dt dh ν') : ν = ν'

theorem **IsQuotMeasure.eq_quotMeas** {ν : Measure (Quot T)} (h : IsQuotMeasure T dt dh ν) : ν = Orb.quotMeas T dt dh

theorem **isQuotMeasure_quotMeas** [SFinite dh] [dh.IsMulLeftInvariant] : IsQuotMeasure T dt dh (quotMeas T dt dh)

theorem **quotMeas_smul_left** (c : ℝ≥0∞) : quotMeas T dt (c • dh) = c • quotMeas T dt dh

theorem **quotMeas_smul_right** {c : ℝ≥0} (h0 : c ≠ 0) : quotMeas T (c • dt) dh = c⁻¹ • quotMeas T dt dh

theorem **measurePreserving_rmul_quotMeas** [dh.IsMulRightInvariant] (g : H) : MeasurePreserving (rmul T g) (quotMeas T dt dh) (quotMeas T dt dh)

theorem **weil_integral** {f : H → ℂ} (hf : Integrable f dh) : (∀ᵐ q ∂quotMeas T dt dh, Integrable (fun t : T => f (t * q.out)) dt) ∧ HasQuotInt T dt dh (fun x => ∫ t : T, f (t * x) ∂dt) (∫ g, f g ∂dh)

theorem **quotMeas_image_lt_top** [IsFiniteMeasureOnCompacts dh] {C : Set H} (hC : IsCompact C) : quotMeas T dt dh (mk T '' C) < ∞

theorem **integrable_descend** [IsFiniteMeasureOnCompacts dh] {φ : H → ℂ} (hi : LeftInv T φ) (hc : Continuous φ) {C : Set H} (hC : IsCompact C) (h0 : ∀ x, φ x ≠ 0 → ∃ t ∈ T, t * x ∈ C) : Integrable (descend T φ) (quotMeas T dt dh)

theorem **quotMeas_of_finite** [IsFiniteMeasure dt] : quotMeas T dt dh = (dt univ)⁻¹ • dh.map (mk T)

theorem **hasQuotInt_of_finite** [IsFiniteMeasure dt] {φ : H → ℂ} (hi : LeftInv T φ) (hφ : Integrable φ dh) : HasQuotInt T dt dh φ ((∫ g, φ g ∂dh) / dt.real univ)

theorem **quotMeas_map** : quotMeas T' (dt.map (subEquiv e h)) (dh.map e) = (quotMeas T dt dh).map (quotMap e h)

theorem **measurableEmbedding_quotMap** : MeasurableEmbedding (quotMap e h)

theorem **hasQuotInt_map** {φ : H' → ℂ} (hi : LeftInv T' φ) (c : ℂ) : (HasQuotInt T' (dt.map (subEquiv e h)) (dh.map e) φ c ↔ HasQuotInt T dt dh (φ ∘ e) c) ∧ quotInt T' (dt.map (subEquiv e h)) (dh.map e) φ = quotInt T dt dh (φ ∘ e)

theorem **mem_cent** {γ t : H} : t ∈ cent γ ↔ t * γ = γ * t

theorem **leftInv_conjFn** {γ : H} {T : Subgroup H} (h : T ≤ cent γ) (f : H → ℂ) : LeftInv T (conjFn γ f)

theorem **orbInt_def** : orbInt dh γ dt f = ∫ q, f (q.out⁻¹ * γ * q.out) ∂quotMeas (cent γ) dt dh

theorem **HasOrbInt.orbInt_eq** (h : HasOrbInt dh γ dt f c) : orbInt dh γ dt f = c

theorem **HasOrbInt.add** (h : HasOrbInt dh γ dt f c) (h' : HasOrbInt dh γ dt f' c') : HasOrbInt dh γ dt (f + f') (c + c')

theorem **HasOrbInt.smul** (h : HasOrbInt dh γ dt f c) (a : ℂ) : HasOrbInt dh γ dt (a • f) (a * c)

theorem **isClosed_cent** (γ : H) : IsClosed (cent γ : Set H)

theorem **isProperOrbit_of_isClosed** [LocallyCompactSpace H] [SecondCountableTopology H] (h : IsClosed (range fun x : H => x⁻¹ * γ * x)) : IsProperOrbit (cent γ) γ

theorem **IsProperOrbit.hasOrbInt** (h : IsProperOrbit (cent γ) γ) (hf : Continuous f) (hc : HasCompactSupport f) : HasOrbInt dh γ dt f (orbInt dh γ dt f)

theorem **hasOrbInt_of_finite** [IsFiniteMeasure dt] (hf : Integrable (conjFn γ f) dh) : HasOrbInt dh γ dt f ((∫ x, f (x⁻¹ * γ * x) ∂dh) / dt.real univ)

theorem **orbInt_smul_left** (a : ℝ≥0) : orbInt (a • dh) γ dt f = a * orbInt dh γ dt f

theorem **hasOrbInt_map** {H' : Type} [Group H'] [TopologicalSpace H'] [IsTopologicalGroup H'] [SecondCountableTopology H'] [LocallyCompactSpace H'] [T2Space H'] [MeasurableSpace H'] [BorelSpace H'] (e : H ≃ₜ* H') {γ' : H'} (h : e γ = γ') (f : H' → ℂ) (c : ℂ) : (HasOrbInt (dh.map e) γ' (dt.map (centEquiv e γ h)) f c ↔ HasOrbInt dh γ dt (f ∘ e) c) ∧ orbInt (dh.map e) γ' (dt.map (centEquiv e γ h)) f = orbInt dh γ dt (f ∘ e)

theorem **hasOrbInt_conj** [dh.IsMulRightInvariant] (g : H) {γ' : H} (h : g * γ * g⁻¹ = γ') (f : H → ℂ) (c : ℂ) : (HasOrbInt dh γ' (dt.map (centEquiv (conjEquiv g) γ h)) f c ↔ HasOrbInt dh γ dt f c) ∧ orbInt dh γ' (dt.map (centEquiv (conjEquiv g) γ h)) f = orbInt dh γ dt f

Development.AbelianVarietyTheory.Definitions.Foundations.MemL2AutomorphicCharacter

theorem **ae_const** {φ : G → ℂ} (hφ : AEStronglyMeasurable φ μ) (h : ∀ g, (fun x => φ (x * g)) =ᵐ[μ] φ) : ∃ c, φ =ᵐ[μ] fun _ => c

theorem **exists_ae_eq_char** {χ : G →* Circle} (hχ : Continuous χ) {F : L2 Γ μ} (h : ∀ g, R Γ μ g F = (χ g : ℂ) • F) : ∃ c : ℂ, lift Γ F =ᵐ[μ] fun x => c * χ x

theorem **lift_smul** (c : ℂ) {F : L2 Γ μ} {φ : G → ℂ} (e : lift Γ F =ᵐ[μ] φ) : lift Γ (c • F : L2 Γ μ) =ᵐ[μ] c • φ

theorem **multIn_char** [NeZero μ] {χ : G →* Circle} (hχ : Continuous χ) (h : MemL2Quotient Γ μ fun x => (χ x : ℂ)) (U : Submodule ℂ (L2 Γ μ)) : (h.toL2 ∈ U → multIn (charRep χ) (R Γ μ) U = 1) ∧ (h.toL2 ∉ U → multIn (charRep χ) (R Γ μ) U = 0)

theorem **multIn_char_eq_zero** [NeZero μ] {χ : G →* Circle} (hχ : Continuous χ) {γ : G} (hγ : γ ∈ Γ) (h : χ γ ≠ 1) (U : Submodule ℂ (L2 Γ μ)) : multIn (charRep χ) (R Γ μ) U = 0

theorem **multTensor_one** {G G₁ G₂ K : Type} [Group G] [Group G₁] [Group G₂] [NormedAddCommGroup K] [InnerProductSpace ℂ K] (j₁ : G₁ →* G) (j₂ : G₂ →* G) (hj : ∀ g, ∃ a b, j₂ a * j₁ b = g) (ρ : G →* (K →L[ℂ] K)) (U : Submodule ℂ K) : multTensor j₁ j₂ (Representation.trivial ℂ G₁ ℂ) (charRep 1) ρ U = multIn (charRep 1) ρ U

theorem **multTensorPi_one_ne_zero** {ι G G₁ K : Type} {Γ : ι → Type} [Fintype ι] [DecidableEq ι] [Group G] [Group G₁] [∀ i, Group (Γ i)] [NormedAddCommGroup K] [InnerProductSpace ℂ K] (j : G₁ →* G) (e : ∀ i, Γ i →* G) {ρ : G →* (K →L[ℂ] K)} {U : Submodule ℂ K} {F : K} (hd : F ∈ U) (hR : ∀ g, ρ g F = F) (h0 : F ≠ 0) : multTensorPi (Hs

theorem **memL2_automorphic_character** (hχ : Continuous χ) (h : ∀ γ ∈ GQ V, χ γ = 1) : MemL2Quotient (GQ V) μ fun x => (χ x : ℂ)

theorem **mDisc_char_eq_one** (hχ : Continuous χ) (h : ∀ γ ∈ GQ V, χ γ = 1) : mDisc V μ (charRep χ) = 1

theorem **mDisc_char_eq_zero** (hχ : Continuous χ) {γ : GA D V} (hγ : γ ∈ GQ V) (h : χ γ ≠ 1) : mDisc V μ (charRep χ) = 0

theorem **mDisc_one** : mDisc V μ (charRep 1) = 1

theorem **mCusp_one_of_aniso** (h : wittIndex V = 0) : mCusp V μ (charRep 1) = 1

theorem **mCusp_one_of_iso** (h : 0 < wittIndex V) : mCusp V μ (charRep 1) = 0

theorem **plane_disc_not_cusp** (μ : Measure (GA D (plane D))) [μ.IsHaarMeasure] [μ.IsMulRightInvariant] : IsDiscreteRep (plane D) μ (charRep 1) ∧ ¬ IsCuspidalRep (plane D) μ (charRep 1)

theorem **right_regular_constant_one** (g : GA D V) : RG V μ g (constantOneDividedPowerChartLift V μ) = constantOneDividedPowerChartLift V μ

theorem **mDiscT_one** : mDiscT V μ (Representation.trivial ℂ (GfC V) ℂ) (charRep 1) = 1

theorem **mCuspT_one_of_aniso** (h : wittIndex V = 0) : mCuspT V μ (Representation.trivial ℂ (GfC V) ℂ) (charRep 1) = 1

theorem **mCuspT_one_of_iso** (h : 0 < wittIndex V) : mCuspT V μ (Representation.trivial ℂ (GfC V) ℂ) (charRep 1) = 0

theorem **mDiscPi_one_ne_zero** : mDiscPi V μ (Hs

theorem **finVec_one** (θ : InfF D) (A : CptF V) : finVec V θ A (charRep 1) = ⊤

theorem **constEmb_mem** (θ : InfF D) (A : CptF V) (h : wittIndex V = 0) : constEmb θ A ∈ formHom V θ A μ (charRep 1)

theorem **constEmb_ne_zero** (θ : InfF D) (A : CptF V) : constEmb θ A ≠ 0

theorem **beF_constEmb** (θ : InfF D) (A : CptF V) (h : wittIndex V = 0) : beF μ (constEmb θ A) (constEmb θ A) = (autMeasure V μ).real Set.univ ∧ beF μ (Complex.I • constEmb θ A) (constEmb θ A) = Complex.I * (autMeasure V μ).real Set.univ

theorem **line_mCusp** (μ : Measure (GA D (line D))) [μ.IsHaarMeasure] [μ.IsMulRightInvariant] : mCusp (line D) μ (charRep 1) = 1

theorem **not_hasLocComp_one** {v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F)} {ψ : Gv V v →* Circle} (h : ψ ≠ 1) : ¬ HasLocComp V v (charRep ψ) (charRep 1)

theorem **multIn_one_euclidean** {G : Type} [Group G] (n : ℕ) : multIn (charRep 1) (1 : G →* (EuclideanSpace ℂ (Fin n) →L[ℂ] EuclideanSpace ℂ (Fin n))) ⊤ = n

theorem **not_isIrrUnitary_two** {G : Type} [Group G] [TopologicalSpace G] : ¬ IsIrrUnitary (1 : G →* (EuclideanSpace ℂ (Fin 2) →L[ℂ] EuclideanSpace ℂ (Fin 2)))

Development.AbelianVarietyTheory.Definitions.Foundations.MemL2OfAutomorphicEmbedding

theorem **isSphericalOff_of_fixed** {S : Finset (HeightOneSpectrum (𝒪 D.host.F))} (h : LGrp.IsSpherical σ (KOff V S)) : IsSphericalOff S σ

theorem **mCusp_eq_Rcusp** : mCusp V μ π = multIn π (Rcusp V μ) ⊤

theorem **mDisc_eq_top** (hπ : IsIrrUnitary π) : mDisc V μ π = multIn π (RG V μ) ⊤

theorem **mDS_congr** {W' : Type} [AddCommGroup W'] [Module ℂ W'] {σ' : Representation ℂ (GfC V) W'} (e : σ.Equiv σ') : mDiscDS μ σ M = mDiscDS μ σ' M ∧ mCuspDS μ σ M = mCuspDS μ σ' M

theorem **lift_add_ae** {F F' : DividedPowerChartLift V μ} {φ ψ : GA D V → ℂ} (e : lift (GQ V) F =ᵐ[μ] φ) (e' : lift (GQ V) F' =ᵐ[μ] ψ) : lift (GQ V) (F + F' : DividedPowerChartLift V μ) =ᵐ[μ] φ + ψ

theorem **lift_smul_ae** (c : ℂ) {F : DividedPowerChartLift V μ} {φ : GA D V → ℂ} (e : lift (GQ V) F =ᵐ[μ] φ) : lift (GQ V) (c • F : DividedPowerChartLift V μ) =ᵐ[μ] c • φ

theorem **exists_toL2_eq_of_ae_eq** {F : DividedPowerChartLift V μ} (hφ : LeftInv (GQ V) φ) (e : lift (GQ V) F =ᵐ[μ] φ) : ∃ h : MemL2Quotient (GQ V) μ φ, h.toL2 = F

theorem **petersson_eq_inner** {F F' : DividedPowerChartLift V μ} (hφ : LeftInv (GQ V) φ) (hψ : LeftInv (GQ V) ψ) (e : lift (GQ V) F =ᵐ[μ] φ) (e' : lift (GQ V) F' =ᵐ[μ] ψ) : petersson V μ φ ψ = ⟪F', F⟫_ℂ

theorem **petersson_rTr_eq_be** (hπ : IsIrrUnitary π) {U U' : Submodule ℂ (DividedPowerChartLift V μ)} {T T' : H →L[ℂ] DividedPowerChartLift V μ} (hT : T ∈ homIn π (RG V μ) U) (hT' : T' ∈ homIn π (RG V μ) U') {x y : H} (hφ : LeftInv (GQ V) φ) (hψ : LeftInv (GQ V) ψ) (e : lift (GQ V) (T x) =ᵐ[μ] φ) (e' : lift (GQ V) (T' y) =ᵐ[μ] ψ) (g : GA D V) : petersson V μ (rTr V g φ) ψ = be T T' * ⟪y, π g x⟫_ℂ

theorem **petersson_rTr_eq_beF** (hπ : IsIrrUnitary π) (h : ι ∈ formHom V θ A μ π) (h' : ι' ∈ formHom V θ A μ π) (g : GA D V) (x y : finVec V θ A π) : petersson V μ (rTr V g (ι x)) (ι' y) = beF μ ι ι' * ⟪(y : H), π g x⟫_ℂ

theorem **petersson_eq_beF** (hπ : IsIrrUnitary π) (h : ι ∈ formHom V θ A μ π) (h' : ι' ∈ formHom V θ A μ π) (x y : finVec V θ A π) : petersson V μ (ι x) (ι' y) = beF μ ι ι' * ⟪y, x⟫_ℂ

theorem **beF_smul_left** (c : ℂ) (ι ι' : X →ₗ[ℂ] (GA D V → ℂ)) : beF μ (c • ι) ι' = c * beF μ ι ι'

theorem **multIn_one_ne_zero** {U : Submodule ℂ (DividedPowerChartLift V μ)} (h : (memL2_const (Γ

theorem **isDiscreteRep_one** : IsDiscreteRep V μ (charRep 1)

Development.AbelianVarietyTheory.Definitions.Foundations.MemL2OfBoundedContinuousInvariant

theorem **mem_extSub_span_lt** {r : ℕ} {C : Matrix (Fin V.rank) (Fin r) E} (i : ℕ) (y : Fin V.rank → R ⊗[F] E) : y ∈ extSub (F

theorem **act_sub_mem_of_mem_span** {S : Set V.V} {A : Submodule E V.V} {g : HA R hc.sig V.gram} (h : ∀ x ∈ S, act hc V g (coordR V R x) − coordR V R x ∈ extSub (F

theorem **map_unipRat** : (𝔤.unipRat D.sp.cm).map (AdelicU.ratEmb D.sp.cm.sig V.gram) = GQ V ⊓ NA 𝔤

theorem **colSpan_le_perp_iff** : colSpan V C ≤ perp V (colSpan V C) ↔ Formalism.UG.cT (D.sp.cm.sig : D.host.E →+* D.host.E) C * V.gram * C = 0

theorem **linearIndependent_vecOf** (h : LinearIndependent D.host.E Cᵀ) : LinearIndependent D.host.E fun k => vecOf V (Cᵀ k)

theorem **NA_frameFlag** (hC : IsIsoFrame V C) : NA (frameFlag hC) = NI V C

theorem **cocompact_single** {W : Submodule D.host.E V.V} (hW : IsIsotropic V W) : Cocompact (GQ V) (NA (.single hW))

theorem **constTerm_eq** {ν : Measure (NA 𝔤)} [ν.IsHaarMeasure] {𝓕 : Set (NA 𝔤)} (h : IsFundamentalDomain (NQ 𝔤) 𝓕 ν) (hφ : LeftInv (GQ V) φ) (x : GA D V) : constTerm 𝔤 φ x = (ν.real 𝓕)⁻¹ • ∫ n in 𝓕, φ (n * x) ∂ν

theorem **isCuspFun_iff** (hφ : LeftInv (GQ V) φ) : IsCuspFun V φ ↔ ∀ (W : Submodule D.host.E V.V) (hW : IsIsotropic V W), constTerm (.single hW) φ = 0

theorem **isCuspFun_of_forall_flag** (hφ : LeftInv (GQ V) φ) (h : ∀ 𝔤 : IsoFlag V, 𝔤.Proper → constTerm 𝔤 φ = 0) : IsCuspFun V φ

theorem **isCuspFun_rTr** (h : IsCuspFun V φ) (k : GA D V) : IsCuspFun V (rTr V k φ)

theorem **mem_cuspCont** : φ ∈ cuspCont V ↔ Continuous φ ∧ LeftInv (GQ V) φ ∧ IsCuspFun V φ

theorem **rTr_mem_cuspCont** (h : φ ∈ cuspCont V) (k : GA D V) : rTr V k φ ∈ cuspCont V

theorem **conj_mem_cuspCont** (h : φ ∈ cuspCont V) : (fun g => conj (φ g)) ∈ cuspCont V

theorem **toL2_mem_cuspidal** [μ.IsMulLeftInvariant] (h : MemL2Quotient (GQ V) μ φ) (hφ : φ ∈ cuspCont V) : h.toL2 ∈ L2cusp V μ

theorem **mem_L2cusp_of_ae_eq** [μ.IsMulLeftInvariant] {F : DividedPowerChartLift V μ} (hφ : φ ∈ cuspCont V) (e : lift (GQ V) F =ᵐ[μ] φ) : F ∈ L2cusp V μ

theorem **L2cusp_le** [μ.IsMulLeftInvariant] {U : Submodule ℂ (DividedPowerChartLift V μ)} (hU : IsClosed (U : Set (DividedPowerChartLift V μ))) (h : ∀ φ ∈ cuspCont V, ∀ h : MemL2Quotient (GQ V) μ φ, h.toL2 ∈ U) : L2cusp V μ ≤ U

theorem **isSubrep_L2cusp** : IsSubrep (RG V μ) (L2cusp V μ)

theorem **isUnitaryRep_Rcusp** : IsUnitaryRep (Rcusp V μ)

theorem **coe_RcuspOp** {f : GA D V → ℂ} (hf : Integrable f μ) (F : L2cusp V μ) : (RcuspOp V μ f F : DividedPowerChartLift V μ) = RopG V μ f F

theorem **exists_isCartanFrame** (hV : V.SigOK) : ∃ A, IsCartanFrame V A

theorem **isKC_mul** {p q : ℕ} {a b : Mat p q} (ha : IsKC a) (hb : IsKC b) : IsKC (a * b)

theorem **mem_Kinf** {g : GA D V} : g ∈ Kinf θ V A ↔ IsKinf θ V A g

theorem **Kinf_le** : Kinf θ V A ≤ GinfC V

theorem **mem_autForms** : φ ∈ autForms θ V K ↔ IsAutFunSm θ V φ ∧ IsKFinite K φ

theorem **mem_cuspForms** : φ ∈ cuspForms θ V K ↔ IsCuspForm θ V φ ∧ IsKFinite K φ

theorem **rTr_fin_mem_autForms** (hθ : IsInfFam θ) (hK : K ≤ GinfC V) (h : φ ∈ autForms θ V K) {g : GA D V} (hg : g ∈ GfC V) : rTr V g φ ∈ autForms θ V K

theorem **rTr_fin_mem_cuspForms** (hθ : IsInfFam θ) (hK : K ≤ GinfC V) (h : φ ∈ cuspForms θ V K) {g : GA D V} (hg : g ∈ GfC V) : rTr V g φ ∈ cuspForms θ V K

theorem **memL2_of_bounded_continuous_invariant** [μ.IsHaarMeasure] (hl : LeftInv (GQ V) φ) (hc : Continuous φ) {C : ℝ} (hC : ∀ g, ‖φ g‖ ≤ C) : MemL2Quotient (GQ V) μ φ

theorem **memL2_of_cusp_form** (hP : Published.CuspRapidDecay) (hθ : IsInfFam θ) [μ.IsHaarMeasure] (h : IsCuspForm θ V φ) : MemL2Quotient (GQ V) μ φ

theorem **lift_cusp_form_toL2** (hP : Published.CuspRapidDecay) (hθ : IsInfFam θ) [μ.IsHaarMeasure] (φ : cuspForms θ V K) : lift (GQ V) (cuspToL2 K μ hP hθ φ : DividedPowerChartLift V μ) =ᵐ[μ] φ

theorem **cusp_form_toL2_injective** (hP : Published.CuspRapidDecay) (hθ : IsInfFam θ) [μ.IsHaarMeasure] : Function.Injective (cuspToL2 (V

theorem **petersson_inner_eq_integral_quotient** (hP : Published.CuspRapidDecay) (hθ : IsInfFam θ) [μ.IsHaarMeasure] (φ ψ : cuspForms θ V K) : ⟪cuspToL2 K μ hP hθ ψ, cuspToL2 K μ hP hθ φ⟫_ℂ = petersson V μ φ ψ

theorem **petersson_eq** [μ.IsMulLeftInvariant] {𝓕 : Set (GA D V)} (h : IsFundamentalDomain (GQ V) 𝓕 μ) (hφ : LeftInv (GQ V) φ) (hψ : LeftInv (GQ V) ψ) : petersson V μ φ ψ = ∫ g in 𝓕, φ g * conj (ψ g) ∂μ

theorem **petersson_conj_symm** (φ ψ : GA D V → ℂ) : conj (petersson V μ ψ φ) = petersson V μ φ ψ

theorem **petersson_smul_left** (c : ℂ) (φ ψ : GA D V → ℂ) : petersson V μ (c • φ) ψ = c * petersson V μ φ ψ

theorem **petersson_add_left** {φ' : GA D V → ℂ} (h : MemL2Quotient (GQ V) μ φ) (h' : MemL2Quotient (GQ V) μ φ') (hψ : MemL2Quotient (GQ V) μ ψ) : petersson V μ (φ + φ') ψ = petersson V μ φ ψ + petersson V μ φ' ψ

theorem **petersson_rTr** [μ.IsHaarMeasure] [μ.IsMulRightInvariant] (h : MemL2Quotient (GQ V) μ φ) (h' : MemL2Quotient (GQ V) μ ψ) (g : GA D V) : petersson V μ (rTr V g φ) (rTr V g ψ) = petersson V μ φ ψ

theorem **petersson_self_pos** [μ.IsHaarMeasure] (h : MemL2Quotient (GQ V) μ φ) (hc : Continuous φ) (h0 : φ ≠ 0) : 0 < (petersson V μ φ φ).re

theorem **le_wittIndex_iff_isIsoSub** {i : ℕ} : i ≤ wittIndex V ↔ ∃ C : Matrix (Fin V.rank) (Fin i) D.host.E, IsIsoSub V C

theorem **wittIndex_eq_zero_iff_isAniso** : wittIndex V = 0 ↔ IsAniso V

theorem **exists_isWittLower_iff** {i : ℕ} : (∃ V' : HSp D, IsWittLower V V' i ∧ V'.SigOK) ↔ i ≤ wittIndex V

theorem **isWittLower_minusHyp** {i : ℕ} (h : i ≤ wittIndex V) : IsWittLower V (minusHyp V i h) i

theorem **rank_minusHyp** {i : ℕ} (h : i ≤ wittIndex V) : (minusHyp V i h).rank + 2 * i = V.rank

theorem **two_mul_wittIndex_le** : 2 * wittIndex V ≤ V.rank

theorem **wittIndex_plane** (D : D7) : wittIndex (plane D) = 1

Development.AbelianVarietyTheory.Definitions.Foundations.MemPuncturedRingIffPowerSeriesPart

theorem **ConvergesOn.mono** {ε ε' : ℝ} {f : ℂ⟦X⟧} (h : ConvergesOn ε f) (hε : ε' ≤ ε) : ConvergesOn ε' f

theorem **ConvergesOn.hasSum** {ε : ℝ} {f : ℂ⟦X⟧} (h : ConvergesOn ε f) {u : ℂ} (hu : ‖u‖ < ε) : HasSum (fun n => PowerSeries.coeff n f * u ^ n) (sum f u)

theorem **convergesOn_iff_le_radius** {ε : ℝ} {f : ℂ⟦X⟧} : ConvergesOn ε f ↔ ENNReal.ofReal ε ≤ radius f

theorem **radius_pos_iff** {f : ℂ⟦X⟧} : 0 < radius f ↔ ∃ ε > 0, ConvergesOn ε f

theorem **hasSum_mul** {f g : ℂ⟦X⟧} {u : ℂ} (hf : Summable fun n => PowerSeries.coeff n f * u ^ n) (hg : Summable fun n => PowerSeries.coeff n g * u ^ n) : HasSum (fun n => PowerSeries.coeff n (f * g) * u ^ n) (sum f u * sum g u)

theorem **hasSum_C** (c u : ℂ) : HasSum (fun n => PowerSeries.coeff n (PowerSeries.C c) * u ^ n) c

theorem **hasSum_X** (u : ℂ) : HasSum (fun n => PowerSeries.coeff n (PowerSeries.X : ℂ⟦X⟧) * u ^ n) u

@[simp] theorem **sum_C** (c u : ℂ) : sum (PowerSeries.C c) u = c

@[simp] theorem **sum_X** (u : ℂ) : sum PowerSeries.X u = u

@[simp] theorem **sum_zero_right** (f : ℂ⟦X⟧) : sum f 0 = PowerSeries.constantCoeff f

theorem **mem_discSeries** {ε : ℝ} {f : ℂ⟦X⟧} : f ∈ discSeries ε ↔ ConvergesOn ε f

theorem **discSeries_anti** : Antitone discSeries

theorem **X_mem_discSeries** (ε : ℝ) : (PowerSeries.X : ℂ⟦X⟧) ∈ discSeries ε

theorem **coe_mem_discSeries** (ε : ℝ) (p : Polynomial ℂ) : (p : ℂ⟦X⟧) ∈ discSeries ε

@[simp] theorem **evalDisc_apply** {ε : ℝ} {u₀ : ℂ} (h : ‖u₀‖ < ε) (f : discSeries ε) : evalDisc ε u₀ h f = sum f u₀

theorem **convergesOn_X_mul** {ε : ℝ} {f : ℂ⟦X⟧} : ConvergesOn ε (PowerSeries.X * f) ↔ ConvergesOn ε f

theorem **convergesOn_X_pow_mul** {ε : ℝ} {f : ℂ⟦X⟧} (n : ℕ) : ConvergesOn ε (PowerSeries.X ^ n * f) ↔ ConvergesOn ε f

theorem **hasFPowerSeriesOnBall_sum** {f : ℂ⟦X⟧} (h : 0 < radius f) : HasFPowerSeriesOnBall (sum f) (FormalMultilinearSeries.ofScalars ℂ fun n => PowerSeries.coeff n f) 0 (radius f)

theorem **ConvergesOn.analyticOnNhd** {ε : ℝ} {f : ℂ⟦X⟧} (h : ConvergesOn ε f) : AnalyticOnNhd ℂ (sum f) (ball 0 ε)

theorem **hasSum_taylor** {ε : ℝ} {F : ℂ → ℂ} (hF : DifferentiableOn ℂ F (ball 0 ε)) {u : ℂ} (hu : ‖u‖ < ε) : HasSum (fun n => PowerSeries.coeff n (taylor F) * u ^ n) (F u)

theorem **taylor_mem_discSeries** {ε : ℝ} {F : ℂ → ℂ} (hF : DifferentiableOn ℂ F (ball 0 ε)) : taylor F ∈ discSeries ε

theorem **sum_taylor** {ε : ℝ} {F : ℂ → ℂ} (hF : DifferentiableOn ℂ F (ball 0 ε)) {u : ℂ} (hu : ‖u‖ < ε) : sum (taylor F) u = F u

theorem **mem_convSeries** {f : ℂ⟦X⟧} : f ∈ convSeries ↔ ∃ ε > 0, f ∈ discSeries ε

theorem **discSeries_le_convSeries** {ε : ℝ} (hε : 0 < ε) : discSeries ε ≤ convSeries

theorem **mem_convSeries_iff_eventually** {f : ℂ⟦X⟧} : f ∈ convSeries ↔ ∀ᶠ u in 𝓝 (0 : ℂ), Summable fun n => PowerSeries.coeff n f * u ^ n

theorem **mem_convSeries_of_summable** {f : ℂ⟦X⟧} {u : ℂ} (hu : u ≠ 0) (h : Summable fun n => PowerSeries.coeff n f * u ^ n) : f ∈ convSeries

theorem **eventually_hasSum** (f : convSeries) : ∀ᶠ u in 𝓝 (0 : ℂ), HasSum (fun n => PowerSeries.coeff n (f : ℂ⟦X⟧) * u ^ n) (sum f u)

theorem **analyticAt_sum** {f : ℂ⟦X⟧} (hf : f ∈ convSeries) : AnalyticAt ℂ (sum f) 0

theorem **taylor_spec** {F : ℂ → ℂ} (hF : AnalyticAt ℂ F 0) : taylor F ∈ convSeries ∧ sum (taylor F) =ᶠ[𝓝 0] F

theorem **eq_zero_of_frequently_sum_eq_zero** {f : ℂ⟦X⟧} (hf : f ∈ convSeries) (h : ∃ᶠ u in 𝓝[≠] (0 : ℂ), sum f u = 0) : f = 0

theorem **germ_apply** (f : convSeries) : germ f = (sum f : Germ (𝓝 (0 : ℂ)) ℂ)

theorem **germ_injective** : Function.Injective germ

theorem **mem_range_germ** {F : ℂ → ℂ} : (F : Germ (𝓝 (0 : ℂ)) ℂ) ∈ germ.range ↔ AnalyticAt ℂ F 0

@[simp] theorem **coe_u** : (u : ℂ⟦X⟧) = PowerSeries.X

theorem **X_pow_mul_mem_convSeries** {f : ℂ⟦X⟧} (n : ℕ) : PowerSeries.X ^ n * f ∈ convSeries ↔ f ∈ convSeries

theorem **isUnit_iff** {f : convSeries} : IsUnit f ↔ PowerSeries.constantCoeff (f : ℂ⟦X⟧) ≠ 0

theorem **u_pow_dvd_iff** {f : convSeries} {n : ℕ} : u ^ n ∣ f ↔ (PowerSeries.X : ℂ⟦X⟧) ^ n ∣ (f : ℂ⟦X⟧)

theorem **irreducible_u** : Irreducible u

theorem **associated_u_pow_order** {f : convSeries} (hf : f ≠ 0) : Associated (u ^ (f : ℂ⟦X⟧).order.toNat) f

theorem **maximalIdeal_eq** : IsLocalRing.maximalIdeal convSeries = Ideal.span {u}

theorem **mem_maximalIdeal** {f : convSeries} : f ∈ IsLocalRing.maximalIdeal convSeries ↔ PowerSeries.constantCoeff (f : ℂ⟦X⟧) = 0

theorem **mem_maximalIdeal_pow** {f : convSeries} {n : ℕ} : f ∈ IsLocalRing.maximalIdeal convSeries ^ n ↔ (f : ℂ⟦X⟧) ∈ (Ideal.span {PowerSeries.X} : Ideal ℂ⟦X⟧) ^ n

@[simp] theorem **evalZero_apply** (f : convSeries) : evalZero f = PowerSeries.constantCoeff (f : ℂ⟦X⟧)

@[simp] theorem **residueFieldEquiv_residue** (f : convSeries) : residueFieldEquiv (IsLocalRing.residue _ f) = PowerSeries.constantCoeff (f : ℂ⟦X⟧)

theorem **eventually_sum_eval** (P : Polynomial convSeries) (a : convSeries) : ∀ᶠ v in 𝓝 (0 : ℂ), sum (P.eval a : convSeries) v = ∑ i ∈ Finset.range (P.natDegree + 1), sum (P.coeff i : convSeries) v * sum a v ^ i

theorem **exists_isRoot** (P : Polynomial convSeries) (z₀ : ℂ) (h₀ : (P.map (evalZero : convSeries →+* ℂ)).eval z₀ = 0) (h₁ : (P.map (evalZero : convSeries →+* ℂ)).derivative.eval z₀ ≠ 0) : ∃ a : convSeries, P.IsRoot a ∧ evalZero a = z₀

@[simp] theorem **quotientMap_mk** (n : ℕ) (f : convSeries) : quotientMap n (Ideal.Quotient.mk _ f) = Ideal.Quotient.mk _ (f : ℂ⟦X⟧)

theorem **quotientMap_bijective** (n : ℕ) : Function.Bijective (quotientMap n)

@[simp] theorem **quotientEquiv_mk** (n : ℕ) (f : convSeries) : quotientEquiv n (Ideal.Quotient.mk _ f) = Ideal.Quotient.mk _ (f : ℂ⟦X⟧)

theorem **factor_quotientMap_evalₐ** {m n : ℕ} (hle : m ≤ n) : (Ideal.Quotient.factorₐ ℂ (Ideal.pow_le_pow_right hle)).comp ((quotientMap n).comp ((AdicCompletion.evalₐ _ n).restrictScalars ℂ)) = (quotientMap m).comp ((AdicCompletion.evalₐ _ m).restrictScalars ℂ)

theorem **mk_completionMap** (n : ℕ) (x : AdicCompletion (IsLocalRing.maximalIdeal convSeries) convSeries) : Ideal.Quotient.mk ((Ideal.span {PowerSeries.X} : Ideal ℂ⟦X⟧) ^ n) (completionMap x) = quotientMap n (AdicCompletion.evalₐ _ n x)

theorem **completionMap_bijective** : Function.Bijective completionMap

@[simp] theorem **completionEquiv_of** (f : convSeries) : completionEquiv (AdicCompletion.of _ _ f) = (f : ℂ⟦X⟧)

theorem **mem_puncturedDisc** {ε : ℝ} {v : ℂ} : v ∈ puncturedDisc ε ↔ v ≠ 0 ∧ ‖v‖ < ε

theorem **isOpen_puncturedDisc** (ε : ℝ) : IsOpen (puncturedDisc ε)

theorem **puncturedDisc_mono** : Monotone puncturedDisc

theorem **puncturedDisc_mem_nhds** {ε : ℝ} (hε : 0 < ε) : puncturedDisc ε ∈ 𝓝[≠] (0 : ℂ)

theorem **eventually_iff_exists_puncturedDisc** {p : ℂ → Prop} : (∀ᶠ v in 𝓝[≠] (0 : ℂ), p v) ↔ ∃ ε > 0, ∀ v ∈ puncturedDisc ε, p v

theorem **hasSum_coe_iff** {g : ℂ⟦X⟧} {v a : ℂ} : HasSum (fun n : ℤ => (g : ℂ((X))).coeff n * v ^ n) a ↔ HasSum (fun n : ℕ => PowerSeries.coeff n g * v ^ n) a

theorem **hasSum_single_mul** {f : ℂ((X))} {v a : ℂ} (hv : v ≠ 0) (m : ℤ) (h : HasSum (fun n : ℤ => f.coeff n * v ^ n) a) : HasSum (fun n : ℤ => (HahnSeries.single m (1 : ℂ) * f).coeff n * v ^ n) (v ^ m * a)

theorem **summable_single_mul_iff** {f : ℂ((X))} {v : ℂ} (hv : v ≠ 0) (m : ℤ) : (Summable fun n : ℤ => (HahnSeries.single m (1 : ℂ) * f).coeff n * v ^ n) ↔ Summable fun n : ℤ => f.coeff n * v ^ n

theorem **summable_single_mul_coe_iff** {g : ℂ⟦X⟧} {v : ℂ} (hv : v ≠ 0) (m : ℤ) : (Summable fun n : ℤ => (HahnSeries.single m (1 : ℂ) * (g : ℂ((X)))).coeff n * v ^ n) ↔ Summable fun n : ℕ => PowerSeries.coeff n g * v ^ n

theorem **hasSum_single_mul_coe** {g : ℂ⟦X⟧} {v : ℂ} (hv : v ≠ 0) (m : ℤ) (h : Summable fun n : ℕ => PowerSeries.coeff n g * v ^ n) : HasSum (fun n : ℤ => (HahnSeries.single m (1 : ℂ) * (g : ℂ((X)))).coeff n * v ^ n) (v ^ m * sum g v)

@[simp] theorem **lsum_coe** (g : ℂ⟦X⟧) (v : ℂ) : lsum (g : ℂ((X))) v = sum g v

theorem **hasSum_mul_laurent** {f g : ℂ((X))} {v : ℂ} (hv : v ≠ 0) (hf : Summable fun n : ℤ => f.coeff n * v ^ n) (hg : Summable fun n : ℤ => g.coeff n * v ^ n) : HasSum (fun n : ℤ => (f * g).coeff n * v ^ n) (lsum f v * lsum g v)

theorem **hasSum_single_laurent** (m : ℤ) (c v : ℂ) : HasSum (fun n : ℤ => (HahnSeries.single m c : ℂ((X))).coeff n * v ^ n) (c * v ^ m)

@[simp] theorem **lsum_single** (m : ℤ) (c v : ℂ) : lsum (HahnSeries.single m c) v = c * v ^ m

@[simp] theorem **lsum_algebraMap** (c v : ℂ) : lsum (algebraMap ℂ ℂ((X))) c) v = c

theorem **mem_puncturedRing** {ε : ℝ} {f : ℂ((X))} : f ∈ puncturedRing ε ↔ ∀ v ∈ puncturedDisc ε, Summable fun n : ℤ => f.coeff n * v ^ n

theorem **puncturedRing_anti** : Antitone puncturedRing

theorem **single_mem_puncturedRing** (ε : ℝ) (m : ℤ) (c : ℂ) : (HahnSeries.single m c : ℂ((X))) ∈ puncturedRing ε

theorem **single_mul_coe_mem_puncturedRing** {ε : ℝ} {g : ℂ⟦X⟧} (m : ℤ) : HahnSeries.single m (1 : ℂ) * (g : ℂ((X))) ∈ puncturedRing ε ↔ g ∈ discSeries ε

theorem **coe_mem_puncturedRing** {ε : ℝ} {g : ℂ⟦X⟧} : (g : ℂ((X))) ∈ puncturedRing ε ↔ g ∈ discSeries ε

theorem **mem_puncturedRing_iff_powerSeriesPart** {ε : ℝ} {f : ℂ((X))} : f ∈ puncturedRing ε ↔ f.powerSeriesPart ∈ discSeries ε

theorem **mem_puncturedRing_iff_exists** {ε : ℝ} {f : ℂ((X))} : f ∈ puncturedRing ε ↔ ∃ n : ℕ, ∃ g ∈ discSeries ε, f = HahnSeries.single (-n : ℤ) (1 : ℂ) * (g : ℂ((X)))

@[simp] theorem **evalAt_apply** {ε : ℝ} {u₀ : ℂ} (h : u₀ ∈ puncturedDisc ε) (f : puncturedRing ε) : evalAt ε u₀ h f = lsum f u₀

theorem **lsum_eq** {ε : ℝ} {f : ℂ((X))} (hf : f ∈ puncturedRing ε) {v : ℂ} (hv : v ∈ puncturedDisc ε) : lsum f v = v ^ f.order * sum f.powerSeriesPart v

theorem **analyticOnNhd_lsum** {ε : ℝ} {f : ℂ((X))} (hf : f ∈ puncturedRing ε) : AnalyticOnNhd ℂ (lsum f) (puncturedDisc ε)

theorem **meromorphicAt_lsum** {ε : ℝ} (hε : 0 < ε) {f : ℂ((X))} (hf : f ∈ puncturedRing ε) : MeromorphicAt (lsum f) 0

theorem **exists_lsum_eq** {ε : ℝ} {F : ℂ → ℂ} (hF : DifferentiableOn ℂ F (puncturedDisc ε)) (h0 : MeromorphicAt F 0) : ∃ f ∈ puncturedRing ε, ∀ v ∈ puncturedDisc ε, lsum f v = F v

theorem **eq_zero_of_frequently_lsum_eq_zero** {ε : ℝ} (hε : 0 < ε) {f : ℂ((X))} (hf : f ∈ puncturedRing ε) (h : ∃ᶠ v in 𝓝[≠] (0 : ℂ), lsum f v = 0) : f = 0

@[simp] theorem **toFunction_apply** {ε : ℝ} (f : puncturedRing ε) (v : puncturedDisc ε) : toFunction ε f v = lsum f v

theorem **toFunction_injective** {ε : ℝ} (hε : 0 < ε) : Function.Injective (toFunction ε)

theorem **range_toFunction** {ε : ℝ} (hε : 0 < ε) : (toFunction ε).range = meromorphicFunctions ε

@[simp] theorem **puncturedEquiv_apply** {ε : ℝ} (hε : 0 < ε) (f : puncturedRing ε) (v : puncturedDisc ε) : (puncturedEquiv hε f : puncturedDisc ε → ℂ) v = lsum f v

@[simp] theorem **coe_algebraMap_discSeries** {ε : ℝ} (g : discSeries ε) : (algebraMap (discSeries ε) (puncturedRing ε) g : ℂ((X))) = ((g : ℂ⟦X⟧) : ℂ((X)))

theorem **mem_convLaurent** {f : ℂ((X))} : f ∈ convLaurent ↔ ∃ ε > 0, f ∈ puncturedRing ε

theorem **puncturedRing_le_convLaurent** {ε : ℝ} (hε : 0 < ε) : puncturedRing ε ≤ convLaurent.toSubalgebra

@[simp] theorem **coe_puncturedIncl** {ε : ℝ} (hε : 0 < ε) (f : puncturedRing ε) : (puncturedIncl hε f : ℂ⸨X⸩) = f

theorem **mem_convLaurent_iff_eventually** {f : ℂ⸨X⸩} : f ∈ convLaurent ↔ ∀ᶠ v in 𝓝[≠] (0 : ℂ), Summable fun n : ℤ => f.coeff n * v ^ n

theorem **coe_mem_convLaurent** {g : ℂ⟦X⟧} : (g : ℂ⸨X⸩) ∈ convLaurent ↔ g ∈ convSeries

theorem **mem_convLaurent_iff_exists** {f : ℂ⸨X⸩} : f ∈ convLaurent ↔ ∃ n : ℕ, ∃ g ∈ convSeries, f = HahnSeries.single (-n : ℤ) (1 : ℂ) * (g : ℂ⸨X⸩)

theorem **mem_convLaurent_of_summable** {f : ℂ⸨X⸩} {v : ℂ} (hv : v ≠ 0) (h : Summable fun n : ℤ => f.coeff n * v ^ n) : f ∈ convLaurent

@[simp] theorem **coe_algebraMap_convSeries** (g : convSeries) : (algebraMap convSeries convLaurent g : ℂ⸨X⸩) = ((g : ℂ⟦X⟧) : ℂ⸨X⸩)

theorem **algebraMap_convSeries_injective** : Function.Injective (algebraMap convSeries convLaurent)

theorem **eventually_hasSum_lsum** (f : convLaurent) : ∀ᶠ v in 𝓝[≠] (0 : ℂ), HasSum (fun n : ℤ => (f : ℂ⸨X⸩).coeff n * v ^ n) (lsum f v)

theorem **eventually_lsum_ne_zero** {f : ℂ⸨X⸩} (hf : f ∈ convLaurent) (h0 : f ≠ 0) : ∀ᶠ v in 𝓝[≠] (0 : ℂ), lsum f v ≠ 0

@[simp] theorem **coeff_expandLaurent_mul** (f : ℂ⸨X⸩) (n : ℤ) : (expandLaurent k hk f).coeff (k * n) = f.coeff n

theorem **coeff_expandLaurent_of_not_dvd** (f : ℂ⸨X⸩) {n : ℤ} (h : ¬ (k : ℤ) ∣ n) : (expandLaurent k hk f).coeff n = 0

@[simp] theorem **expandLaurent_coe** (g : ℂ⟦X⟧) : expandLaurent k hk (g : ℂ⸨X⸩) = (PowerSeries.expand k hk g : ℂ⸨X⸩)

theorem **hasSum_expandLaurent_iff** {f : ℂ⸨X⸩} {v a : ℂ} : HasSum (fun n : ℤ => (expandLaurent k hk f).coeff n * v ^ n) a ↔ HasSum (fun n : ℤ => f.coeff n * (v ^ k) ^ n) a

@[simp] theorem **lsum_expandLaurent** (f : ℂ⸨X⸩) (v : ℂ) : lsum (expandLaurent k hk f) v = lsum f (v ^ k)

@[simp] theorem **sum_expand** (g : ℂ⟦X⟧) (v : ℂ) : sum (PowerSeries.expand k hk g) v = sum g (v ^ k)

theorem **image_pow_puncturedDisc** {ε : ℝ} (hε : 0 ≤ ε) : (fun v : ℂ => v ^ k) '' puncturedDisc ε = puncturedDisc (ε ^ k)

theorem **expandLaurent_mem_puncturedRing** {ε : ℝ} (hε : 0 ≤ ε) {f : ℂ⸨X⸩} : expandLaurent k hk f ∈ puncturedRing ε ↔ f ∈ puncturedRing (ε ^ k)

theorem **expandLaurent_mem_convLaurent** {f : ℂ⸨X⸩} : expandLaurent k hk f ∈ convLaurent ↔ f ∈ convLaurent

theorem **expand_mem_convSeries** {g : ℂ⟦X⟧} : PowerSeries.expand k hk g ∈ convSeries ↔ g ∈ convSeries

@[simp] theorem **coe_expandConv** (g : convSeries) : (expandConv k hk g : ℂ⟦X⟧) = PowerSeries.expand k hk g

@[simp] theorem **expandConv_u** : expandConv k hk u = u ^ k

@[simp] theorem **coe_expandConvLaurent** (f : convLaurent) : (expandConvLaurent k hk f : ℂ⸨X⸩) = expandLaurent k hk f

theorem **eventually_iff_exp** {p : ℂ → Prop} : (∀ᶠ v in 𝓝[≠] (0 : ℂ), p v) ↔ ∃ T : ℝ, ∀ w : ℂ, T < w.im → p (Complex.exp (2 * Real.pi * Complex.I * w))

theorem **exists_chart_of_isLocalRing** {Y : Scheme.{0}} (s : Y ⟶ Spec (.of ℂ)) [LocallyOfFiniteType s] {R : Type} [CommRing R] [IsLocalRing R] [Algebra ℂ R] (x : Spec (.of R) ⟶ Y) (hx : x ≫ s = Spec.map (CommRingCat.ofHom (algebraMap ℂ R))) : ∃ (A : CommRingCat.{0}) (_ : Algebra ℂ A) (_ : Algebra.FiniteType ℂ A) (i : Spec A ⟶ Y) (φ : A →ₐ[ℂ] R), IsOpenImmersion i ∧ i ≫ s = Spec.map (CommRingCat.ofHom (algebraMap ℂ A)) ∧ Spec.map (CommRingCat.ofHom (φ : A →+* R)) ≫ i = x

Development.AbelianVarietyTheory.Definitions.Foundations.MemRingOfInR

theorem **mem_ring_of_inR** {E : Subring ℂ} {f : ℍ → ℂ} (h : InR E f) : f ∈ ring E

theorem **den_zero** {Ω : Type} [Field Ω] : Den (0 : HahnSeries ℚ Ω)

Development.AbelianVarietyTheory.Definitions.Foundations.MemSiegelOfIffRightAct

@[simp] theorem **pairVec_inl** {X : Type} (x : m → X) (y : n → X) (i : m) : pairVec e x y (e (.inl i)) = x i

@[simp] theorem **pairVec_inr** {X : Type} (x : m → X) (y : n → X) (i : n) : pairVec e x y (e (.inr i)) = y i

theorem **pairVec_eta** {X : Type} (w : ι → X) : pairVec e (fun i => w (e (.inl i))) (fun i => w (e (.inr i))) = w

theorem **submatrix_blkMat** (x : Matrix m m A) (y : Matrix n n A) : (blkMat e x y).submatrix e e = fromBlocks x 0 0 y

theorem **submatrix_inj** {M N : Matrix ι ι A} (h : M.submatrix e e = N.submatrix e e) : M = N

theorem **blkMat_map** (f : A → B) (hf : f 0 = 0) (x : Matrix m m A) (y : Matrix n n A) : (blkMat e x y).map f = blkMat e (x.map f) (y.map f)

theorem **cT_blkMat** (σ : A →+* A) (x : Matrix m m A) (y : Matrix n n A) : cT σ (blkMat e x y) = blkMat e (cT σ x) (cT σ y)

theorem **blkMat_inj** {x x' : Matrix m m A} {y y' : Matrix n n A} : blkMat e x y = blkMat e x' y' ↔ x = x' ∧ y = y'

theorem **blkMat_mul** (x x' : Matrix m m A) (y y' : Matrix n n A) : blkMat e x y * blkMat e x' y' = blkMat e (x * x') (y * y')

theorem **mulVec_pairVec** (M : Matrix ι ι A) (x : m → A) (y : n → A) : M *ᵥ pairVec e x y = pairVec e ((M.submatrix e e).toBlocks₁₁ *ᵥ x + (M.submatrix e e).toBlocks₁₂ *ᵥ y) ((M.submatrix e e).toBlocks₂₁ *ᵥ x + (M.submatrix e e).toBlocks₂₂ *ᵥ y)

theorem **pairVec_vecMul** (M : Matrix ι ι A) (x : m → A) (y : n → A) : pairVec e x y ᵥ* M = pairVec e (x ᵥ* (M.submatrix e e).toBlocks₁₁ + y ᵥ* (M.submatrix e e).toBlocks₂₁) (x ᵥ* (M.submatrix e e).toBlocks₁₂ + y ᵥ* (M.submatrix e e).toBlocks₂₂)

theorem **blkMat_mulVec** (x : Matrix m m A) (y : Matrix n n A) (u : m → A) (v : n → A) : blkMat e x y *ᵥ pairVec e u v = pairVec e (x *ᵥ u) (y *ᵥ v)

theorem **blkMat_one** : blkMat e (1 : Matrix m m A) (1 : Matrix n n A) = 1

theorem **det_blkMat** (x : Matrix m m A) (y : Matrix n n A) : (blkMat e x y).det = x.det * y.det

@[simp] theorem **val_blkGL** (g : GL m A × GL n A) : (blkGL e g).1 = blkMat e g.1.1 g.2.1

theorem **blkGL_injective** : Function.Injective (blkGL (A

theorem **bc_blkGL** (f : A →+* B) (x : GL m A) (y : GL n A) : bc f (blkGL e (x, y)) = blkGL e (bc f x, bc f y)

theorem **blkGL_mem_UJ** {σ : A →+* A} {J : Matrix m m A} {J' : Matrix n n A} {x : GL m A} {y : GL n A} : blkGL e (x, y) ∈ UJ σ (blkMat e J J') ↔ x ∈ UJ σ J ∧ y ∈ UJ σ J'

theorem **UJ_neg** (σ : A →+* A) (J : Matrix n n A) : UJ σ (-J) = UJ σ J

theorem **continuous_blkGL** [TopologicalSpace A] : Continuous (blkGL (A

theorem **mem_sumLat** {O X : Type} [CommRing O] [AddCommGroup X] [Module O X] {L : Submodule O (m → X)} {L' : Submodule O (n → X)} {w : ι → X} : w ∈ sumLat e L L' ↔ (fun i => w (e (.inl i))) ∈ L ∧ (fun i => w (e (.inr i))) ∈ L'

theorem **pairVec_mem_sumLat** {O X : Type} [CommRing O] [AddCommGroup X] [Module O X] {L : Submodule O (m → X)} {L' : Submodule O (n → X)} {x : m → X} {y : n → X} : pairVec e x y ∈ sumLat e L L' ↔ x ∈ L ∧ y ∈ L'

theorem **blkGL_mem_stabGL** [Fintype m] [Fintype n] [Fintype ι] [DecidableEq m] [DecidableEq n] [DecidableEq ι] {O : Type} [CommRing O] [Algebra O A] {L : Submodule O (m → A)} {L' : Submodule O (n → A)} {x : GL m A} {y : GL n A} : blkGL e (x, y) ∈ stabGL (sumLat e L L') ↔ x ∈ stabGL L ∧ y ∈ stabGL L'

theorem **pairVec_mem_diagSub** {x y : n → A} : pairVec e x y ∈ diagSub A e ↔ x = y

theorem **pairVec_mem_antiSub** {x y : n → A} : pairVec e x y ∈ antiSub A e ↔ x = -y

theorem **mem_diagSub** {w : ι → A} : w ∈ diagSub A e ↔ ∃ x, w = pairVec e x x

theorem **mem_antiSub** {w : ι → A} : w ∈ antiSub A e ↔ ∃ x, w = pairVec e x (-x)

theorem **isCompl_diagSub_antiSub** [Invertible (2 : A)] : IsCompl (diagSub A e) (antiSub A e)

theorem **leviMat_blkMat** (x y : Matrix n n A) : leviMat e (blkMat e x y) = x

theorem **keepsDiag_blkMat** {x y : Matrix n n A} : KeepsDiag e (blkMat e x y) ↔ y = x

theorem **leviMat_map** (f : A →+* B) (M : Matrix ι ι A) : leviMat e (M.map f) = (leviMat e M).map f

theorem **keepsDiag_iff** {M : Matrix ι ι A} : KeepsDiag e M ↔ ∀ i j, M (e (.inr i)) (e (.inl j)) + M (e (.inr i)) (e (.inr j)) = M (e (.inl i)) (e (.inl j)) + M (e (.inl i)) (e (.inr j))

theorem **KeepsDiag.map** {M : Matrix ι ι A} (h : KeepsDiag e M) (f : A →+* B) : KeepsDiag e (M.map f)

theorem **hform_dblGram** (σ : A →+* A) (J : Matrix n n A) (x y x' y' : n → A) : hform σ (dblGram e J) (pairVec e x y) (pairVec e x' y') = hform σ J x x' - hform σ J y y'

theorem **KeepsDiag.mulVec_diag** {M : Matrix ι ι A} (h : KeepsDiag e M) (x : n → A) : M *ᵥ pairVec e x x = pairVec e (leviMat e M *ᵥ x) (leviMat e M *ᵥ x)

theorem **KeepsDiag.quotMat_eq** {M : Matrix ι ι A} (h : KeepsDiag e M) : quotMat e M = (M.submatrix e e).toBlocks₂₂ - (M.submatrix e e).toBlocks₁₂

theorem **KeepsDiag.mulVec_quot** {M : Matrix ι ι A} (h : KeepsDiag e M) (x y : n → A) (i : n) : (M *ᵥ pairVec e x y) (e (.inl i)) - (M *ᵥ pairVec e x y) (e (.inr i)) = (quotMat e M *ᵥ (x - y)) i

theorem **KeepsDiag.det_eq** {M : Matrix ι ι A} (h : KeepsDiag e M) : M.det = (leviMat e M).det * (quotMat e M).det

theorem **keepsDiag_of_mulVec** {M : Matrix ι ι A} (h : ∀ w ∈ diagSub A e, M *ᵥ w ∈ diagSub A e) : KeepsDiag e M

theorem **mem_stabGL_diagSub** {p : GL ι A} : p ∈ stabGL (diagSub A e) ↔ KeepsDiag e p.1

theorem **leviMat_mul** {M N : Matrix ι ι A} (hM : KeepsDiag e M) (hN : KeepsDiag e N) : leviMat e (M * N) = leviMat e M * leviMat e N

@[simp] theorem **val_leviGL** (p : stabGL (diagSub A e)) : (leviGL e p).1 = leviMat e p.1.1

theorem **quot_levi** {σ : A →+* A} {J : Matrix n n A} {M : Matrix ι ι A} (hM : cT σ M * dblGram e J * M = dblGram e J) (h : KeepsDiag e M) : cT σ (quotMat e M) * J * leviMat e M = J

theorem **sgn_mul_sgn** : blkMat e (1 : Matrix n n A) (-1) * blkMat e (1 : Matrix n n A) (-1) = 1

theorem **det_quot_mul_det_levi** {σ : A →+* A} {J : Matrix n n A} (hJ : IsUnit J.det) {M : Matrix ι ι A} (hM : cT σ M * dblGram e J * M = dblGram e J) (h : KeepsDiag e M) : σ (quotMat e M).det * (leviMat e M).det = 1

theorem **cayley_mulVec** (x y : n → A) : (cayley e).1 *ᵥ pairVec e x y = pairVec e (x + y) (x - y)

theorem **leviElt_mulVec** (g : GL n A × GL n A) (x y : n → A) : (leviElt e g).1 *ᵥ pairVec e (x + y) (x - y) = pairVec e (g.1.1 *ᵥ x + g.2.1 *ᵥ y) (g.1.1 *ᵥ x - g.2.1 *ᵥ y)

theorem **leviElt_diag** (g : GL n A × GL n A) (x : n → A) : (leviElt e g).1 *ᵥ pairVec e x x = pairVec e (g.1.1 *ᵥ x) (g.1.1 *ᵥ x)

theorem **leviElt_anti** (g : GL n A × GL n A) (y : n → A) : (leviElt e g).1 *ᵥ pairVec e y (-y) = pairVec e (g.2.1 *ᵥ y) (-(g.2.1 *ᵥ y))

theorem **leviElt_mem_diag** (g : GL n A × GL n A) : leviElt e g ∈ stabGL (diagSub A e)

theorem **leviElt_mem_anti** (g : GL n A × GL n A) : leviElt e g ∈ stabGL (antiSub A e)

theorem **leviMat_leviElt** (g : GL n A × GL n A) : leviMat e (leviElt e g).1 = g.1.1

theorem **leviElt_mem_UJ** {σ : A →+* A} {J : Matrix n n A} {α β : GL n A} (h1 : cT σ β.1 * J * α.1 = J) (h2 : cT σ α.1 * J * β.1 = J) : leviElt e (α, β) ∈ UJ σ (dblGram e J)

theorem **exists_leviElt_mem_UJ** {σ : A →+* A} (hσ : ∀ x, σ (σ x) = x) {J : Matrix n n A} (hJ : cT σ J = J) (hu : IsUnit J.det) (α : GL n A) : ∃ β : GL n A, leviElt e (α, β) ∈ UJ σ (dblGram e J)

theorem **mem_siegelOf** {g : G} : g ∈ siegelOf e mat ↔ KeepsDiag e (mat g).1

@[simp] theorem **val_leviOf** (p : siegelOf e mat) : (leviOf e mat p).1 = leviMat e (mat p).1

@[simp] theorem **val_aOf** (p : siegelOf e mat) : (aOf e mat p).1 = (leviMat e (mat p).1).det

theorem **mem_unipOf** {g : G} : g ∈ unipOf e mat ↔ g ∈ siegelOf e mat ∧ leviMat e (mat g).1 = 1

theorem **isClosed_siegelOf** [T2Space A] (h : Continuous mat) : IsClosed (siegelOf e mat : Set G)

theorem **continuous_leviOf** (h : Continuous mat) : Continuous (leviOf e mat)

theorem **continuous_aOf** (h : Continuous mat) : Continuous (aOf e mat)

theorem **mem_degPSOf** [TopologicalSpace G] {μ : Aˣ →* Circle} {s : ℂ} {Φ : G → ℂ} : Φ ∈ degPSOf e mat ν K μ s ↔ (∀ (p : siegelOf e mat) (g : G), Φ (p * g) = μ (aOf e mat p) * npow (ν (aOf e mat p)) (s + (Fintype.card n : ℂ) / 2) * Φ g) ∧ Continuous Φ ∧ IsKFinite K Φ

theorem **flatOf_apply** (Φ : G → ℂ) (s₀ s : ℂ) (g : G) : flatOf e mat ν K Φ s₀ s g = Φ g * npow (hgtOf e mat ν K g) (s - s₀)

theorem **hgtOf_pos** (g : G) : 0 < hgtOf e mat ν K g

theorem **IwasawaOf.hgtOf_mul** (h : IwasawaOf e mat ν K) (p : siegelOf e mat) {k : G} (hk : k ∈ K) : hgtOf e mat ν K (p * k) = ν (aOf e mat p)

theorem **iwasawaOf_of_isCompact** (hm : Continuous mat) (hν : Continuous ν) (hK : IsCompact (K : Set G)) (h : ∀ g : G, ∃ p : siegelOf e mat, ∃ k ∈ K, g = p * k) : IwasawaOf e mat ν K

theorem **IwasawaOf.continuous_hgtOf** (h : IwasawaOf e mat ν K) (hm : Continuous mat) (hν : Continuous ν) (hK : IsCompact (K : Set G)) : Continuous (hgtOf e mat ν K)

theorem **IwasawaOf.flatOf_mem** (h : IwasawaOf e mat ν K) (hm : Continuous mat) (hν : Continuous ν) (hK : IsCompact (K : Set G)) {μ : Aˣ →* Circle} {s₀ : ℂ} {Φ : G → ℂ} (hΦ : Φ ∈ degPSOf e mat ν K μ s₀) (s : ℂ) : flatOf e mat ν K Φ s₀ s ∈ degPSOf e mat ν K μ s

theorem **IwasawaOf.spherical_mem** (h : IwasawaOf e mat ν K) (hm : Continuous mat) (hν : Continuous ν) (hK : IsCompact (K : Set G)) (s : ℂ) : (fun g => npow (hgtOf e mat ν K g) (s + (Fintype.card n : ℂ) / 2)) ∈ degPSOf e mat ν K 1 s

theorem **coe_ioU** (e : n ⊕ n ≃ ι) {J : Matrix n n A} {Q : Matrix ι ι A} (h : Q = dblGram e J) (g : UJ σ J × UJ σ J) : (ioU σ e h g : GL ι A) = blkGL e (g.1, g.2)

Development.AbelianVarietyTheory.Definitions.Foundations.MemUnitaryGroupIffIsIsometry

@[simp] theorem **toSesq_apply** (h : HermForm R M) (x : M) : h.toSesq x = h x

@[simp] theorem **mk_apply** (B : SesqForm R M) (hB) (x : M) : (⟨B, hB⟩ : HermForm R M) x = B x

@[**ext**] theorem ext {h h' : HermForm R M} (H : ∀ x y, h x y = h' x y) : h = h'

theorem **toSesq_injective** : Function.Injective (toSesq : HermForm R M → SesqForm R M)

@[simp] theorem **star_apply** (x y : M) : star (h x y) = h y x

@[simp] theorem **conj_apply** (x y : M) : starRingEnd R (h x y) = h y x

theorem **isSelfAdjoint_apply_self** (x : M) : IsSelfAdjoint (h x x)

theorem **add_left** (x y z : M) : h (x + y) z = h x z + h y z

theorem **add_right** (x y z : M) : h x (y + z) = h x y + h x z

theorem **sub_left** (x y z : M) : h (x − y) z = h x z − h y z

theorem **sub_right** (x y z : M) : h x (y − z) = h x y − h x z

theorem **neg_left** (x y : M) : h (−x) y = −h x y

theorem **neg_right** (x y : M) : h x (−y) = −h x y

theorem **zero_left** (y : M) : h 0 y = 0

theorem **zero_right** (x : M) : h x 0 = 0

theorem **smul_left** (a : R) (x y : M) : h (a • x) y = a * h x y

theorem **smul_right** (a : R) (x y : M) : h x (a • y) = star a * h x y

theorem **sum_left** {ι : Type*} (s : Finset ι) (f : ι → M) (y : M) : h (∑ i ∈ s, f i) y = ∑ i ∈ s, h (f i) y

theorem **sum_right** {ι : Type*} (s : Finset ι) (x : M) (f : ι → M) : h x (∑ i ∈ s, f i) = ∑ i ∈ s, h x (f i)

theorem **apply_eq_zero_comm** (x y : M) : h x y = 0 ↔ h y x = 0

theorem **isRefl** : h.toSesq.IsRefl

theorem **isSymm_flip** : h.toSesq.flip.IsSymm

@[simp] theorem **ofFun_apply** (f : M → M → R) (h₁ h₂ h₃) (x y : M) : ofFun f h₁ h₂ h₃ x y = f x y

@[simp] theorem **comap_apply** (f : N →ₗ[R] M) (x y : N) : h.comap f x y = h (f x) (f y)

@[simp] theorem **comap_id** : h.comap LinearMap.id = h

theorem **comap_comp** (f : N →ₗ[R] M) (g : P →ₗ[R] N) : h.comap (f ∘ₗ g) = (h.comap f).comap g

@[simp] theorem **restrict_apply** (W : Submodule R M) (x y : W) : h.restrict W x y = h x y

@[simp] theorem **toDual_apply** (y x : M) : h.toDual y x = h x y

theorem **nondegenerate_iff_left** : h.Nondegenerate ↔ ∀ x, (∀ y, h x y = 0) → x = 0

theorem **nondegenerate_iff_right** : h.Nondegenerate ↔ ∀ y, (∀ x, h x y = 0) → y = 0

theorem **nondegenerate_iff_injective** : h.Nondegenerate ↔ Function.Injective h.toDual

theorem **unimodular_iff** : h.Unimodular ↔ h.Nondegenerate ∧ ∀ φ : Module.Dual R M, ∃ y, ∀ x, h x y = φ x

theorem **Nondegenerate.eq_zero_left** (H : h.Nondegenerate) {x : M} (hx : ∀ y, h x y = 0) : x = 0

theorem **Nondegenerate.eq_zero_right** (H : h.Nondegenerate) {y : M} (hy : ∀ x, h x y = 0) : y = 0

theorem **Nondegenerate.ext_left** (H : h.Nondegenerate) {x x' : M} (hx : ∀ y, h x y = h x' y) : x = x'

theorem **Nondegenerate.ext_right** (H : h.Nondegenerate) {y y' : M} (hy : ∀ x, h x y = h x y') : y = y'

theorem **Unimodular.nondegenerate** (H : h.Unimodular) : h.Nondegenerate

theorem **Unimodular.exists_eq** (H : h.Unimodular) (φ : Module.Dual R M) : ∃ y, ∀ x, h x y = φ x

@[simp] theorem **Unimodular.toDualEquiv_apply** (H : h.Unimodular) (y x : M) : H.toDualEquiv y x = h x y

@[simp] theorem **Unimodular.apply_toDualEquiv_symm** (H : h.Unimodular) (φ : Module.Dual R M) (x : M) : h x (H.toDualEquiv.symm φ) = φ x

theorem **isIsometry_iff_comap** {h₁ : HermForm R M} {h₂ : HermForm R N} {f : M →ₗ[R] N} : h₁.IsIsometry h₂ f ↔ h₂.comap f = h₁

theorem **IsIsometry.id** (h : HermForm R M) : h.IsIsometry h LinearMap.id

theorem **IsIsometry.comp** {h₁ : HermForm R M} {h₂ : HermForm R N} {h₃ : HermForm R P} {g : N →ₗ[R] P} {f : M →ₗ[R] N} (hg : h₂.IsIsometry h₃ g) (hf : h₁.IsIsometry h₂ f) : h₁.IsIsometry h₃ (g ∘ₗ f)

theorem **IsIsometry.injective** {h₁ : HermForm R M} {h₂ : HermForm R N} {f : M →ₗ[R] N} (hf : h₁.IsIsometry h₂ f) (H : h₁.Nondegenerate) : Function.Injective f

theorem **isIsometry_subtype** (h : HermForm R M) (W : Submodule R M) : (h.restrict W).IsIsometry h W.subtype

@[simp] theorem **coe_toLinearEquiv** (f : h₁.IsometryEquiv h₂) : ⇑f.toLinearEquiv = f

@[simp] theorem **coe_mk** (e : M ≃ₗ[R] N) (he) : ⇑(⟨e, he⟩ : h₁.IsometryEquiv h₂) = e

@[**ext**] theorem ext {f g : h₁.IsometryEquiv h₂} (H : ∀ x, f x = g x) : f = g

@[simp] theorem **map_app** (f : h₁.IsometryEquiv h₂) (x y : M) : h₂ (f x) (f y) = h₁ x y

theorem **isIsometry** (f : h₁.IsometryEquiv h₂) : h₁.IsIsometry h₂ f.toLinearEquiv.toLinearMap

@[simp] theorem **refl_apply** (h : HermForm R M) (x : M) : refl h x = x

@[simp] theorem **trans_apply** (f : h₁.IsometryEquiv h₂) (g : h₂.IsometryEquiv h₃) (x : M) : f.trans g x = g (f x)

@[simp] theorem **apply_symm_apply** (f : h₁.IsometryEquiv h₂) (x : N) : f (f.symm x) = x

@[simp] theorem **symm_apply_apply** (f : h₁.IsometryEquiv h₂) (x : M) : f.symm (f x) = x

@[simp] theorem **symm_symm** (f : h₁.IsometryEquiv h₂) : f.symm.symm = f

@[simp] theorem **toLinearEquiv_symm** (f : h₁.IsometryEquiv h₂) : f.symm.toLinearEquiv = f.toLinearEquiv.symm

theorem **map_app_left** (f : h₁.IsometryEquiv h₂) (x : M) (y : N) : h₂ (f x) y = h₁ x (f.symm y)

theorem **map_app_right** (f : h₁.IsometryEquiv h₂) (x : N) (y : M) : h₂ x (f y) = h₁ (f.symm x) y

theorem **nondegenerate** (f : h₁.IsometryEquiv h₂) (H : h₁.Nondegenerate) : h₂.Nondegenerate

theorem **unimodular** (f : h₁.IsometryEquiv h₂) (H : h₁.Unimodular) : h₂.Unimodular

@[**refl**] theorem refl (h : HermForm R M) : h.Isometric h

@[**symm**] theorem symm (H : h₁.Isometric h₂) : h₂.Isometric h₁

@[**trans**] theorem trans (H : h₁.Isometric h₂) (H' : h₂.Isometric h₃) : h₁.Isometric h₃

theorem **mem_unitaryGroup** {g : M ≃ₗ[R] M} : g ∈ h.unitaryGroup ↔ ∀ x y, h (g x) (g y) = h x y

theorem **mem_unitaryGroup_iff_isIsometry** {g : M ≃ₗ[R] M} : g ∈ h.unitaryGroup ↔ h.IsIsometry h g.toLinearMap

theorem **smul_def** (g : h.unitaryGroup) (x : M) : g • x = (g : M ≃ₗ[R] M) x

@[simp] theorem **map_app** (g : h.unitaryGroup) (x y : M) : h ((g : M ≃ₗ[R] M) x) ((g : M ≃ₗ[R] M) y) = h x y

@[simp] theorem **smul_app** (g : h.unitaryGroup) (x y : M) : h (g • x) (g • y) = h x y

theorem **smul_app_left** (g : h.unitaryGroup) (x y : M) : h (g • x) y = h x (g⁻¹ • y)

theorem **smul_app_right** (g : h.unitaryGroup) (x y : M) : h x (g • y) = h (g⁻¹ • x) y

@[**ext**] theorem ext {g g' : h.unitaryGroup} (H : ∀ x, (g : M ≃ₗ[R] M) x = (g' : M ≃ₗ[R] M) x) : g = g'

@[simp] theorem **toIsometryEquiv_apply** (g : h.unitaryGroup) (x : M) : toIsometryEquiv g x = (g : M ≃ₗ[R] M) x

@[simp] theorem **toEnd_apply** (g : h.unitaryGroup) (x : M) : toEnd h g x = (g : M ≃ₗ[R] M) x

theorem **toEnd_injective** : Function.Injective (toEnd h)

@[simp] theorem **IsometryEquiv.coe_toUnitary** (f : h.IsometryEquiv h) : (f.toUnitary : M ≃ₗ[R] M) = f.toLinearEquiv

@[simp] theorem **IsometryEquiv.unitaryCongr_apply** {h₁ : HermForm R M} {h₂ : HermForm R N} (f : h₁.IsometryEquiv h₂) (g : h₁.unitaryGroup) (x : N) : (f.unitaryCongr g : N ≃ₗ[R] N) x = f ((g : M ≃ₗ[R] M) (f.symm x))

@[simp] theorem **IsometryEquiv.unitaryCongr_symm** {h₁ : HermForm R M} {h₂ : HermForm R N} (f : h₁.IsometryEquiv h₂) : f.unitaryCongr.symm = f.symm.unitaryCongr

theorem **Nondegenerate.unimodular** (H : h.Nondegenerate) : h.Unimodular

theorem **unimodular_iff_nondegenerate** : h.Unimodular ↔ h.Nondegenerate

@[simp] theorem **IsIsometry.toUnitary_apply** {f : V →ₗ[K] V} (hf : h.IsIsometry h f) (H : h.Nondegenerate) (x : V) : (hf.toUnitary H : V ≃ₗ[K] V) x = f x

@[simp] theorem **im_apply_self** (x : V) : (h x x).im = 0

@[simp] theorem **ofReal_re_apply_self** (x : V) : ((h x x).re : ℂ) = h x x

theorem **PosDef.nondegenerate** (H : h.PosDef) : h.Nondegenerate

theorem **NegDef.nondegenerate** (H : h.NegDef) : h.Nondegenerate

theorem **HermSpace.unimodular** {K : Type u} [Field K] [StarRing K] (V : HermSpace.{u, v} K) : V.form.Unimodular

theorem **IsStarFixed.of_isScalarTower** (F₀ : Type*) [CommSemiring F₀] [Algebra F₀ F] [Algebra F₀ E] [IsScalarTower F₀ F E] [IsStarFixed F E] : IsStarFixed F₀ E

@[simp] theorem **starAlgEquiv_apply** (x : E) : starAlgEquiv F E x = star x

@[simp] theorem **starAlgEquiv_symm** : (starAlgEquiv F E).symm = starAlgEquiv F E

theorem **IsStarFixed.star_smul** (a : F) (x : E) : star (a • x) = a • star x

theorem **IsStarFixed.isIntegral_star** {F E : Type*} [CommRing F] [CommRing E] [StarRing E] [Algebra F E] [IsStarFixed F E] {x : E} (hx : IsIntegral F x) : IsIntegral F (star x)

theorem **starModule_toAlgebra** {R S : Type*} [CommSemiring R] [CommSemiring S] [StarRing R] [StarRing S] (φ : R →+* S) (hφ : ∀ x, φ (star x) = star (φ x)) : letI

theorem **starModule_of_algebraMap** {R S : Type*} [CommSemiring R] [CommSemiring S] [StarRing R] [StarRing S] [Algebra R S] (hφ : ∀ x, algebraMap R S (star x) = star (algebraMap R S x)) : StarModule R S

theorem **HermForm.smul_right_of_isStarFixed** {F R M : Type*} [CommRing F] [CommRing R] [StarRing R] [Algebra F R] [IsStarFixed F R] [AddCommGroup M] [Module R M] [Module F M] [IsScalarTower F R M] (h : HermForm R M) (a : F) (x y : M) : h x (a • y) = a • h x y

theorem **HermForm.smul_left_of_tower** {S R M : Type*} [CommRing S] [CommRing R] [StarRing R] [Algebra S R] [AddCommGroup M] [Module R M] [Module S M] [IsScalarTower S R M] (h : HermForm R M) (a : S) (x y : M) : h (a • x) y = algebraMap S R a * h x y

theorem **HermForm.smul_right_of_tower** {S R M : Type*} [CommRing S] [CommRing R] [StarRing R] [Algebra S R] [AddCommGroup M] [Module R M] [Module S M] [IsScalarTower S R M] (h : HermForm R M) (a : S) (x y : M) : h x (a • y) = star (algebraMap S R a) * h x y

@[simp] theorem **toTensor_tmul** (e : E) (a : A) : toTensor F E A (tmul F e a) = e ⊗ₜ[F] a

@[simp] theorem **toTensor_symm_tmul** (e : E) (a : A) : (toTensor F E A).symm (e ⊗ₜ[F] a) = tmul F e a

theorem **algebraMap_left** (e : E) : algebraMap E (StarTensor F E A) e = tmul F e 1

theorem **algebraMap_right** (a : A) : algebraMap A (StarTensor F E A) a = tmul F 1 a

theorem **one_def** : (1 : StarTensor F E A) = tmul F 1 1

@[simp] theorem **tmul_mul_tmul** (e e' : E) (a a' : A) : tmul F e a * tmul F e' a' = tmul F (e * e') (a * a')

theorem **tmul_eq_mul** (e : E) (a : A) : tmul F e a = algebraMap E (StarTensor F E A) e * algebraMap A (StarTensor F E A) a

theorem **add_tmul** (e e' : E) (a : A) : tmul F (e + e') a = tmul F e a + tmul F e' a

theorem **tmul_add** (e : E) (a a' : A) : tmul F e (a + a') = tmul F e a + tmul F e a'

@[simp] theorem **zero_tmul** (a : A) : tmul F (0 : E) a = 0

@[simp] theorem **tmul_zero** (e : E) : tmul F e (0 : A) = 0

theorem **smul_tmul** (c : F) (e : E) (a : A) : tmul F (c • e) a = tmul F e (c • a)

@[simp] theorem **left_smul_tmul** (c e : E) (a : A) : c • tmul F e a = tmul F (c * e) a

@[simp] theorem **right_smul_tmul** (c : A) (e : E) (a : A) : c • tmul F e a = tmul F e (c * a)

theorem **induction_on** {motive : StarTensor F E A → Prop} (x : StarTensor F E A) (zero : motive 0) (tmul : ∀ e a, motive (tmul F e a)) (add : ∀ x y, motive x → motive y → motive (x + y)) : motive x

theorem **ringHom_ext** {C : Type*} [Semiring C] {f g : StarTensor F E A →+* C} (hE : ∀ e, f (algebraMap E _ e) = g (algebraMap E _ e)) (hA : ∀ a, f (algebraMap A _ a) = g (algebraMap A _ a)) : f = g

@[simp] theorem **commRight_tmul** (e : E) (a : A) : commRight F E A (tmul F e a) = a ⊗ₜ[F] e

@[simp] theorem **commRight_symm_tmul** (e : E) (a : A) : (commRight F E A).symm (a ⊗ₜ[F] e) = tmul F e a

theorem **algebraMap_left_injective** [Module.Flat F E] (hA : Function.Injective (algebraMap F A)) : Function.Injective (algebraMap E (StarTensor F E A))

theorem **algebraMap_right_injective** [Module.Flat F A] (hE : Function.Injective (algebraMap F E)) : Function.Injective (algebraMap A (StarTensor F E A))

@[simp] theorem **map_tmul** (f : A →ₐ[F] B) (e : E) (a : A) : map E f (tmul F e a) = tmul F e (f a)

@[simp] theorem **map_algebraMap_right** (f : A →ₐ[F] B) (a : A) : map E f (algebraMap A _ a) = algebraMap B _ (f a)

@[simp] theorem **map_id** : map E (AlgHom.id F A) = AlgHom.id E _

theorem **map_comp** (g : B →ₐ[F] C) (f : A →ₐ[F] B) : map E (g.comp f) = (map E g).comp (map E f)

@[simp] theorem **lift_tmul** {D : Type*} [CommRing D] (f : E →+* D) (g : A →+* D) (hfg) (e : E) (a : A) : lift f g hfg (tmul F e a) = f e * g a

@[simp] theorem **lift_algebraMap_left** {D : Type*} [CommRing D] (f : E →+* D) (g : A →+* D) (hfg) (e : E) : lift f g hfg (algebraMap E (StarTensor F E A) e) = f e

@[simp] theorem **lift_algebraMap_right** {D : Type*} [CommRing D] (f : E →+* D) (g : A →+* D) (hfg) (a : A) : lift f g hfg (algebraMap A (StarTensor F E A) a) = g a

@[simp] theorem **rid_tmul** (e : E) (a : F) : rid F E (tmul F e a) = a • e

@[simp] theorem **conjHom_tmul** (e : E) (a : A) : conjHom F E A (tmul F e a) = tmul F (star e) a

@[simp] theorem **star_tmul** (e : E) (a : A) : star (tmul F e a) = tmul F (star e) a

@[simp] theorem **star_algebraMap_left** (e : E) : star (algebraMap E (StarTensor F E A) e) = algebraMap E _ (star e)

theorem **map_star_of** {C : Type*} [CommRing C] [StarRing C] (φ : StarTensor F E A →+* C) (hE : ∀ e, φ (algebraMap E _ (star e)) = star (φ (algebraMap E _ e))) (hA : ∀ a, star (φ (algebraMap A _ a)) = φ (algebraMap A _ a)) (x : StarTensor F E A) : φ (star x) = star (φ x)

theorem **map_star** (f : A →ₐ[F] B) (x : StarTensor F E A) : map E f (star x) = star (map E f x)

theorem **lift_star** {D : Type*} [CommRing D] [StarRing D] (f : E →+* D) (g : A →+* D) (hfg) (hf : ∀ e, f (star e) = star (f e)) (hg : ∀ a, star (g a) = g a) (x : StarTensor F E A) : lift f g hfg (star x) = star (lift f g hfg x)

theorem **rid_star** (x : StarTensor F E F) : rid F E (star x) = star (rid F E x)

theorem **add_star_mem_range** (hE : ∀ e : E, e + star e ∈ (algebraMap F E).range) (x : StarTensor F E A) : x + star x ∈ (algebraMap A (StarTensor F E A)).range

theorem **ringOfIntegers_smul_def** (a : 𝒪 K) (x : V) : a • x = (a : K) • x

theorem **star_eq_complexConj** (x : E) : star x = IsCMField.complexConj E x

theorem **starAlgEquiv_eq_complexConj** : starAlgEquiv (maximalRealSubfield E) E = IsCMField.complexConj E

@[simp] theorem **complexEmbedding_star** (τ : E →+* ℂ) (x : E) : τ (star x) = star (τ x)

theorem **star_eq_self_iff_mem** (x : E) : star x = x ↔ x ∈ maximalRealSubfield E

theorem **add_star_mem** (x : E) : x + star x ∈ maximalRealSubfield E

theorem **mul_star_mem** (x : E) : x * star x ∈ maximalRealSubfield E

theorem **exists_star_ne** : ∃ x : E, star x ≠ x

theorem **HermForm.apply_self_mem** {V : Type*} [AddCommGroup V] [Module E V] (h : HermForm E V) (x : V) : h x x ∈ maximalRealSubfield E

theorem **ringOfIntegers_star_def** (x : 𝒪 E) : star x = IsCMField.ringOfIntegersComplexConj E x

@[simp] theorem **ringOfIntegers_coe_star** (x : 𝒪 E) : algebraMap (𝒪 E) E (star x) = star (algebraMap (𝒪 E) E x)

theorem **ringOfIntegers_star_eq_self_iff** (x : 𝒪 E) : star x = x ↔ x ∈ (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E)).range

theorem **finrank_right** [StrongRankCondition A] : Module.finrank A (CMTensor E A) = 2

theorem **algebraMap_right_injective** : Function.Injective (algebraMap A (CMTensor E A))

theorem **algebraMap_left_injective** [Nontrivial A] : Function.Injective (algebraMap E (CMTensor E A))

theorem **star_eq_self_iff** (x : CMTensor E A) : star x = x ↔ x ∈ (algebraMap A (CMTensor E A)).range

Development.AbelianVarietyTheory.Definitions.Foundations.MultipliableAndSgNeZero

theorem **admissible_example_alt** : admitted 5 ![1 / 3, 0]

theorem **exponent_exact** {c : ℕ} (h : Nat.Coprime c 6) : 12 * ((c ^ 2 - 1) / 12) + 1 = c ^ 2 ∧ 2 * (((c : ℤ) - (c : ℤ) ^ 2) / 2) = (c : ℤ) - (c : ℤ) ^ 2

theorem **ee_ne_zero** (z : ℂ) : ee z ≠ 0

theorem **ee_add** (z w : ℂ) : ee (z + w) = ee z * ee w

theorem **ee_nat_mul** (n : ℕ) (z : ℂ) : ee (n * z) = ee z ^ n

theorem **ee_neg** (z : ℂ) : ee (-z) = (ee z)⁻¹

theorem **norm_ee** (z : ℂ) : ‖ee z‖ = Real.exp (-(2 * Real.pi * z.im))

theorem **norm_ee_lt_one** (τ : ℍ) : ‖ee τ‖ < 1

theorem **ee_eq_one** {a b : ℚ} (τ : ℍ) (h : ee ((a : ℂ) * τ + b) = 1) : a = 0 ∧ ∃ n : ℤ, (n : ℚ) = b

theorem **ne_one_of_not_int** {y : Fin 2 → ℚ} (hy : ¬ (y 0 ∈ Set.range ((↑) : ℤ → ℚ) ∧ y 1 ∈ Set.range ((↑) : ℤ → ℚ))) (τ : ℍ) (m : ℕ) : 1 - ee τ ^ m * ee (pt y τ) ≠ 0 ∧ 1 - ee τ ^ m * (ee (pt y τ))⁻¹ ≠ 0

theorem **hasProd_geom** {q : ℂ} (hq : ‖q‖ < 1) (a : ℂ) (h : ∀ m : ℕ, 1 - q ^ (m + 1) * a ≠ 0) : ∃ P : ℂ, P ≠ 0 ∧ HasProd (fun m : ℕ => 1 - q ^ (m + 1) * a) P

theorem **hasProd_inv** {f : ℕ → ℂ} {P : ℂ} (h : HasProd f P) (hP : P ≠ 0) : HasProd (fun m => (f m)⁻¹) P⁻¹

theorem **hasProd_pow** {f : ℕ → ℂ} {P : ℂ} (h : HasProd f P) (k : ℕ) : HasProd (fun m => (f m) ^ k) (P ^ k)

theorem **ee_pt_pow** (c : ℕ) (x : Fin 2 → ℚ) (τ : ℍ) : ee (pt x τ) ^ c = ee (pt (c • x) τ)

theorem **multipliable_and_sg_ne_zero** {c : ℕ} {x : Fin 2 → ℚ} (h : admitted c x) (τ : ℍ) : Multipliable (fun m : ℕ => cthFac c (ee τ) (ee (pt x τ)) (m + 1)) ∧ 1 - ee (pt x τ) ^ c ≠ 0 ∧ sg c x τ ≠ 0

theorem **sg_translate** (c : ℕ) (x : Fin 2 → ℚ) (τ τ' : ℍ) (h : (τ' : ℂ) = τ + 1) : sg c x τ' = sg c ![x 0, x 0 + x 1] τ

theorem **f_zero_two_eq** (b : ℚ) : f0 2 ![0, b] = -1 / 12

theorem **f_zero_one_third** (b : ℚ) : f0 1 ![4 / 3, b] = 1 / 6

theorem **summable_exp_abs** {a : ℝ} (ha : 0 < a) : Summable fun n : ℤ => Real.exp (-(a * |(n : ℝ)|))

theorem **summable_pow_exp_abs** {a : ℝ} (ha : 0 < a) (k : ℕ) (s : ℝ) : Summable fun n : ℤ => |s + n| ^ k * Real.exp (-(2 * a * |s + n|))

theorem **summable_fTerm** (j : ℕ) (x : Fin 2 → ℚ) (τ : ℍ) : Summable (fTerm j x τ)

Development.AbelianVarietyTheory.Definitions.Foundations.NonemptyAlgHomOfNotIsField

@[simp] theorem **splitProj_tmul** (τ : E →ₐ[F] Fv F v) (a : Fv F v) (x : E) : splitProj τ (a ⊗ₜ[F] x) = a * τ x

theorem **splitProj_inclR** (τ : E →ₐ[F] Fv F v) (x : E) : splitProj τ (inclR (Fv F v) x) = τ x

theorem **splitProj_comp** (τ : E →ₐ[F] Fv F v) (σ : E →ₐ[F] E) (y : Fv F v ⊗[F] E) : splitProj (τ.comp σ) y = splitProj τ (sigR _ σ y)

theorem **continuous_splitProj** (τ : E →ₐ[F] Fv F v) : Continuous (splitProj τ)

theorem **cm_conj_conj** (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

theorem **exists_cm_conj_ne** (h : IsCMExt F E σ) : ∃ x : E, σ x ≠ x

theorem **comp_conj_conj** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : (τ.comp σ).comp σ = τ

theorem **comp_conj_ne** {K : Type} [Field K] [Algebra F K] (h : IsCMExt F E σ) (τ : E →ₐ[F] K) : τ.comp σ ≠ τ

theorem **eq_or_eq_comp_conj** {K : Type} [Field K] [Algebra F K] (h : IsCMExt F E σ) (τ τ' : E →ₐ[F] K) : τ' = τ ∨ τ' = τ.comp σ

theorem **splitProj_comp_sigR** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : splitProj (τ.comp σ) (sigR _ σ y) = splitProj τ y

theorem **splitPair_bijective** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : Function.Bijective ((splitProj τ).prod (splitProj (τ.comp σ)))

theorem **splitAlgEquiv_apply** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : splitAlgEquiv h τ y = (splitProj τ y, splitProj (τ.comp σ) y)

@[simp] theorem **splitProj_symm** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (p : Fv F v × Fv F v) : splitProj τ ((splitAlgEquiv h τ).symm p) = p.1

@[simp] theorem **splitProj_comp_symm** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (p : Fv F v × Fv F v) : splitProj (τ.comp σ) ((splitAlgEquiv h τ).symm p) = p.2

theorem **split_ext** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) {y z : Fv F v ⊗[F] E} (h1 : splitProj τ y = splitProj τ z) (h2 : splitProj (τ.comp σ) y = splitProj (τ.comp σ) z) : y = z

theorem **splitAlgEquiv_sigR** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : splitAlgEquiv h τ (sigR _ σ y) = (splitAlgEquiv h τ y).swap

theorem **continuous_splitAlgEquiv_symm** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : Continuous (splitAlgEquiv h τ).symm

@[simp] theorem **splitProj_splitIdem** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : splitProj τ (splitIdem h τ) = 1

@[simp] theorem **splitProj_comp_splitIdem** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : splitProj (τ.comp σ) (splitIdem h τ) = 0

theorem **splitIdem_mul_self** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : splitIdem h τ * splitIdem h τ = splitIdem h τ

theorem **sigR_splitIdem** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : sigR _ σ (splitIdem h τ) = 1 - splitIdem h τ

theorem **splitIdem_comp** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : splitIdem h (τ.comp σ) = 1 - splitIdem h τ

theorem **splitIdem_mul** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : splitIdem h τ * y = splitProj τ y • splitIdem h τ

theorem **splitProj_eq_zero_iff** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : splitProj τ y = 0 ↔ splitIdem h τ * y = 0

theorem **finEmb_ofId_comp** (τ : E →ₐ[F] Fv F v) (y : Fv F v ⊗[F] E) : finEmb F v ((IsScalarTower.toAlgHom F (Fv F v) (FvBar F v)).comp τ) y = algebraMap (Fv F v) (FvBar F v) (splitProj τ y)

theorem **nnnorm_le_one_iff_mem** {a : Fv F v} : ‖a‖₊ ≤ 1 ↔ a ∈ v.adicCompletionIntegers F

theorem **splitProj_mem** (τ : E →ₐ[F] Fv F v) {y : Fv F v ⊗[F] E} (hy : y ∈ intLoc F E v) : splitProj τ y ∈ v.adicCompletionIntegers F

theorem **mem_intLoc_iff_split** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) {y : Fv F v ⊗[F] E} : y ∈ intLoc F E v ↔ splitProj τ y ∈ v.adicCompletionIntegers F ∧ splitProj (τ.comp σ) y ∈ v.adicCompletionIntegers F

theorem **splitIdem_mem** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : splitIdem h τ ∈ intLoc F E v

theorem **apply_coe_mem** (τ : E →ₐ[F] Fv F v) (x : 𝒪 E) : τ x ∈ v.adicCompletionIntegers F

@[simp] theorem **coe_splitInt** (τ : E →ₐ[F] Fv F v) (x : 𝒪 E) : (splitInt τ x : Fv F v) = τ x

theorem **mem_maximalIdeal_iff_valued** {a : v.adicCompletionIntegers F} : a ∈ IsLocalRing.maximalIdeal (v.adicCompletionIntegers F) ↔ Valued.v (a : Fv F v) < 1

theorem **valued_apply_algebraMap** (τ : E →ₐ[F] Fv F v) (a : F) : Valued.v (τ (algebraMap F E a)) = v.valuation F a

theorem **algebraMap_mem_comap_splitInt** (τ : E →ₐ[F] Fv F v) (a : 𝒪 F) : algebraMap (𝒪 F) (𝒪 E) a ∈ (IsLocalRing.maximalIdeal _).comap (splitInt τ) ↔ a ∈ v.asIdeal

theorem **mem_splitPrime** (τ : E →ₐ[F] Fv F v) (x : 𝒪 E) : x ∈ (splitPrime τ).asIdeal ↔ Valued.v (τ x) < 1

theorem **algebraMap_mem_splitPrime** (τ : E →ₐ[F] Fv F v) (a : 𝒪 F) : algebraMap (𝒪 F) (𝒪 E) a ∈ (splitPrime τ).asIdeal ↔ a ∈ v.asIdeal

theorem **splitPrime_under** (τ : E →ₐ[F] Fv F v) : (splitPrime τ).under (𝒪 F) = v

theorem **mem_splitPrime_comp** (τ : E →ₐ[F] Fv F v) (σ : E →ₐ[F] E) (x : 𝒪 E) : x ∈ (splitPrime (τ.comp σ)).asIdeal ↔ RingOfIntegers.mapRingHom (σ : E →+* E) x ∈ (splitPrime τ).asIdeal

theorem **valued_apply_eq_valuation** (τ : E →ₐ[F] Fv F v) (x : E) : Valued.v (τ x) = (splitPrime τ).valuation E x

theorem **residue_splitInt_surjective** (τ : E →ₐ[F] Fv F v) : Function.Surjective ((IsLocalRing.residue _).comp (splitInt τ))

theorem **splitResidueEquiv_mk** (τ : E →ₐ[F] Fv F v) (x : 𝒪 E) : splitResidueEquiv τ (Ideal.Quotient.mk _ x) = IsLocalRing.residue _ (splitInt τ x)

theorem **splitPrime_comp_ne** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : splitPrime (τ.comp σ) ≠ splitPrime τ

theorem **splitPrime_injective** (h : IsCMExt F E σ) : Function.Injective (splitPrime : (E →ₐ[F] Fv F v) → HeightOneSpectrum (𝒪 E))

theorem **ncard_primesOver_le_two** (h : IsCMExt F E σ) (v : HeightOneSpectrum (𝒪 F)) : (v.asIdeal.primesOver (𝒪 E)).ncard ≤ 2

theorem **primesOver_eq_split** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) : v.asIdeal.primesOver (𝒪 E) = {(splitPrime τ).asIdeal, (splitPrime (τ.comp σ)).asIdeal}

theorem **nonempty_algHom_of_not_isField** {K A : Type} [Field K] [CommRing A] [Algebra K A] (h2 : Module.finrank K A = 2) (hf : ¬ IsField A) : Nonempty (A →ₐ[K] K)

theorem **nonempty_split_iff** (h : IsCMExt F E σ) : Nonempty (E →ₐ[F] Fv F v) ↔ (v.asIdeal.primesOver (𝒪 E)).ncard = 2

theorem **existsUnique_splitPrime_eq** (h : IsCMExt F E σ) {w w' : HeightOneSpectrum (𝒪 E)} (hne : w ≠ w') (hw : w.under (𝒪 F) = v) (hw' : w'.under (𝒪 F) = v) : ∃! τ : E →ₐ[F] Fv F v, splitPrime τ = w

Development.AbelianVarietyTheory.Definitions.Foundations.NonemptyLiftObj

theorem **eisG_scale** {Λ Λ' : AddSubgroup ℂ} {c : ℂ} (hc : c ≠ 0) (h : ∀ z, z ∈ Λ' ↔ c * z ∈ Λ) (k : ℕ) : eisG Λ' k = c ^ (2 * k) * eisG Λ k

theorem **exists_gauss** : ∃ (Λ : AddSubgroup ℂ) (c : ℂ), c ≠ 0 ∧ (∀ z, z ∈ Λ ↔ ∃ m n : ℤ, (m : ℂ) + n * Complex.I = c * z) ∧ g2 Λ = 4 ∧ g3 Λ = 0

theorem **nonempty_place** (p : ℕ) [Fact p.Prime] : Nonempty (Qbar →+* ℂ_[p])

theorem **norm_two** {p : ℕ} [Fact p.Prime] (hp : p ≠ 2) : ‖(2 : ℂ_[p])‖ = 1

theorem **exists_ok** (p : ℕ) [Fact p.Prime] (hp : p ≠ 2) : ∃ g : G1 p, g.Ok ∧ ModelOverW g.lam g.abelian₂ g.C ∧ g.W = ⟨0, 0, 0, -1, 0⟩ ∧ g2 g.abelian₂ = 4 ∧ g3 g.abelian₂ = 0 ∧ ∃ c : ℂ, c ≠ 0 ∧ ∀ z, z ∈ g.abelian₂ ↔ ∃ m n : ℤ, (m : ℂ) + n * Complex.I = c * z

theorem **at_ofRec** (D : D7) {p : ℕ} [Fact p.Prime] (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) (g : GL3Theta.G1 p) : At D cc δ (ofRec g) ↔ GL3Theta.At D cc δ g

theorem **exists_at** (D : D7) (p : ℕ) [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (cc : Ideal (𝒪 D.host.K)) (h1 : IsCurveIdeal D cc) (h3 : Nat.Coprime (Ideal.absNorm cc) p) : ∃ (δ : ℂ) (g : G1 p), At D cc δ g ∧ g.lam = lam

theorem **exists_at_top** (D : D7) (p : ℕ) [Fact p.Prime] : ∃ (δ : ℂ) (g : G1 p), At D ⊤ δ g

theorem **simeq_refl** {σ : Type*} {p : ℕ} [Fact p.Prime] (f : MvPowerSeries σ ℂ_[p]) : Simeq f f

theorem **not_simeq_X** (p : ℕ) [Fact p.Prime] : ¬ Simeq (Xv () : MvPowerSeries Unit ℂ_[p]) 0

theorem **coeff_one_logF** (p : ℕ) [Fact p.Prime] : PowerSeries.coeff 1 (logF p) = 1

theorem **hP_smul** {p : ℕ} [Fact p.Prime] {ι : Type*} [Fintype ι] (a : ℂ_[p] × ℂ_[p]) (x y : ι → ℂ_[p] × ℂ_[p]) : hP (fun k => a * x k) y = a * hP x y ∧ hP x (fun k => a * y k) = a.swap * hP x y

theorem **mem_range_iff** {E : Type} [Field E] [NumberField E] (x : E) : x ∈ (algebraMap (𝒪 E) E).range ↔ IsIntegral ℤ x

theorem **mem_stdLat** {E : Type} [Field E] [NumberField E] {N : ℕ} (x : Fin N → E) : x ∈ stdLat E N ↔ ∀ k, IsIntegral ℤ (x k)

theorem **isSDLat_std** (E : Type) [Field E] [NumberField E] [StarRing E] (N : ℕ) : IsSDLat E N (stdLat E N)

theorem **nonempty_liftObj** (D : D7) (u : D.what₂.Word) (s : D.what₂.Tau → ℂ) (hs : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M), (∀ τ x, embT D τ x ∈ M) ∧ ∀ τ, ∃ t : M, (t : ℂ) = s τ ∧ ellT D M τ = Submodule.span (𝒪 M) {t}) : Nonempty (LiftObj D (frameChain D u))

Development.AbelianVarietyTheory.Definitions.Foundations.NonnegOfMemCStarSet

theorem **exists_basis_co** (h : HasGram N) : ∃ b : Basis (Fin (cDim N)) K V, IsGram b (frameCo N) N ∧ toCo N = b.equivFun ∧ ofCo N = b.equivFun.symm

theorem **cDim_eq** (h : HasGram N) : cDim N = finrank K V

theorem **ofCo_toCo** (h : HasGram N) (x : V) : ofCo N (toCo N x) = x

theorem **toCo_ofCo** (h : HasGram N) (y : Fin (cDim N) → K) : toCo N (ofCo N y) = y

theorem **toCo_bijective** (h : HasGram N) : Function.Bijective (toCo N)

theorem **mapCo_toCo** (h : HasGram N) (φ : V →ₗ[K] W) (x : V) : mapCo N N' φ (toCo N x) = toCo N' (φ x)

theorem **latCo_bijective** (h : HasGram N) : Function.Bijective (latCo N)

theorem **latCo_le_iff** (h : HasGram N) {L L' : Submodule (𝒪 K) V} : latCo N L ≤ latCo N L' ↔ L ≤ L'

theorem **latCo_eq_bot_iff** (h : HasGram N) {L : Submodule (𝒪 K) V} : latCo N L = ⊥ ↔ L = ⊥

theorem **latCo_inf** (h : HasGram N) (L : Submodule (𝒪 K) V) (S : Submodule K V) : latCo N (L ⊓ S.restrictScalars (𝒪 K)) = latCo N L ⊓ (S.map (toCo N)).restrictScalars (𝒪 K)

theorem **range_mapCo** (h : HasGram N) (e : V →ₗ[K] V) : LinearMap.range (mapCo N N e) = (LinearMap.range e).map (toCo N)

theorem **ker_mapCo** (h : HasGram N) (e : V →ₗ[K] V) : LinearMap.ker (mapCo N N e) = (LinearMap.ker e).map (toCo N)

theorem **rkK_latCo** (h : HasGram N) (L : Submodule (𝒪 K) V) : rkK (latCo N L) = rkK L

theorem **zl_latCo** {b : Basis (Fin (cDim N)) K V} (h1 : toCo N = b.equivFun) (L : Submodule (𝒪 K) V) : zl (realize (Pi.basisFun K (Fin (cDim N))) (frameCo N)) (latCo N L) = zl (realize b (frameCo N)) L

theorem **muDef_latCo** (h : HasGram N) {L : Submodule (𝒪 K) V} (hL : L.FG) : muDef (Pi.basisFun K (Fin (cDim N))) (frameCo N) (latCo N L) = slope_sq N L

theorem **muMaxDef_latCo** (h : HasGram N) {L : Submodule (𝒪 K) V} (hL : L.FG) (h0 : L ≠ ⊥) : muMaxDef (Pi.basisFun K (Fin (cDim N))) (frameCo N) (latCo N L) = slopeMax_sq N L

theorem **hasGram_piNorm** {N : (K →+* ℂ) → V → ℝ} (h : HasGram N) : HasGram (piNorm ι N)

theorem **compMean_eq** {g : Comp Xo c → ℝ} (h : ∀ Z Z', g Z = g Z') (Z : Comp Xo c) : compMean g = g Z

theorem **nonneg_of_mem_cStarSet** {t : ℝ} (h : t ∈ cStarSet U x V dict G) : 0 ≤ t

theorem **cStarInf_nonneg** : 0 ≤ cStarInf U x V dict G

theorem **HasPhi.nonneg** {y : IntFr Xo c T io} {φ : ℝ} (h : HasPhi y φ) : 0 ≤ φ

theorem **exists_hasPhi** (y : IntFr Xo c T io) (G : SAB H (Sp T.V)) (t' : SAB H (Sp T.Fr)) (m1 : letI

theorem **refExp_vp_le_iff** {cst : ℝ} {G : WTZ E} {p : ℕ} {a : EReal} : refExp_vp U cst G p ≤ a ↔ ∀ (la : PointsOfFlagModel.abelian₂ p c) (T : CDVR) (io : c.F →+* T.Fr) (e : ℕ), OverLa la.1 T.V io e → ∀ y : IntFr Xo c T io, y.IsAlg → ∀ φ : ℝ, HasPhi y φ → depthAt_vp U G p y ≤ a + ((cst * φ / (vV T.V (p : T.Fr) : ℝ) : ℝ) : EReal)

Development.AbelianVarietyTheory.Definitions.Foundations.NormEulerPolySubOneLe

@[simp] theorem **eulerPoly_zero** (x : ℂ) : eulerPoly 0 x = 1

@[simp] theorem **eulerFac_zero** (x : ℂ) : eulerFac 0 x = 1

theorem **eulerPoly_cons** (b : ℂˣ) (B : Multiset ℂˣ) (x : ℂ) : eulerPoly (b ::ₘ B) x = (1 - (b : ℂ) * x) * eulerPoly B x

@[simp] theorem **eulerFac_singleton** (b : ℂˣ) (x : ℂ) : eulerFac {b} x = (1 - (b : ℂ) * x)⁻¹

theorem **eulerPoly_add** (A B : Multiset ℂˣ) (x : ℂ) : eulerPoly (A + B) x = eulerPoly A x * eulerPoly B x

theorem **eulerFac_add** (A B : Multiset ℂˣ) (x : ℂ) : eulerFac (A + B) x = eulerFac A x * eulerFac B x

theorem **eulerFac_eq_prod** (B : Multiset ℂˣ) (x : ℂ) : eulerFac B x = (B.map fun b : ℂˣ => (1 - (b : ℂ) * x)⁻¹).prod

theorem **eulerFac_map** {ι : Type} (s : Finset ι) (α : ι → ℂˣ) (x : ℂ) : eulerFac (s.val.map α) x = ∏ i ∈ s, (1 - (α i : ℂ) * x)⁻¹

theorem **eulerFac_map_mul** (z : ℂˣ) (B : Multiset ℂˣ) (x : ℂ) : eulerFac (B.map (z * ·)) x = eulerFac B ((z : ℂ) * x)

theorem **eulerPoly_ne_zero** {B : Multiset ℂˣ} {x : ℂ} (h : ∀ b ∈ B, ‖(b : ℂ) * x‖ < 1) : eulerPoly B x ≠ 0

theorem **norm_eulerPoly_sub_one_le** {B : Multiset ℂˣ} {x : ℂ} {y : ℝ} (hy0 : 0 ≤ y) (hy : y ≤ 1) (h : ∀ b ∈ B, ‖(b : ℂ) * x‖ ≤ y) : ‖eulerPoly B x - 1‖ ≤ (2 ^ B.card - 1) * y

theorem **one_lt_absNorm** (w : HeightOneSpectrum (𝒪 K)) : 1 < Ideal.absNorm w.asIdeal

theorem **npow_ne_zero** (w : HeightOneSpectrum (𝒪 K)) (z : ℂ) : npow w z ≠ 0

theorem **npow_add** (w : HeightOneSpectrum (𝒪 K)) (y z : ℂ) : npow w (y + z) = npow w y * npow w z

theorem **norm_npow** (w : HeightOneSpectrum (𝒪 K)) (z : ℂ) : ‖npow w z‖ = ((Ideal.absNorm w.asIdeal : ℕ) : ℝ) ^ z.re

@[simp] theorem **coe_npowU** (w : HeightOneSpectrum (𝒪 K)) (z : ℂ) : (npowU w z : ℂ) = npow w z

theorem **locL_add** (c c' : Fam K) (w : HeightOneSpectrum (𝒪 K)) (s : ℂ) : locL (c + c') w s = locL c w s * locL c' w s

theorem **locL_twist** (c : Fam K) (z : HeightOneSpectrum (𝒪 K) → ℂˣ) (w : HeightOneSpectrum (𝒪 K)) (s : ℂ) : locL (twist c z) w s = ((c w).map fun b : ℂˣ => (1 - (b : ℂ) * (z w : ℂ) * npow w (-s))⁻¹).prod

@[simp] theorem **twist_one** (c : Fam K) : twist c 1 = c

theorem **twist_twist** (c : Fam K) (y z : HeightOneSpectrum (𝒪 K) → ℂˣ) : twist (twist c y) z = twist c (z * y)

theorem **twist_add** (c c' : Fam K) (z : HeightOneSpectrum (𝒪 K) → ℂˣ) : twist (c + c') z = twist c z + twist c' z

theorem **twist_segFam** (y z : HeightOneSpectrum (𝒪 K) → ℂˣ) (t : ℕ) : twist (segFam y t) z = segFam (z * y) t

@[simp] theorem **card_seg** (w : HeightOneSpectrum (𝒪 K)) (x : ℂˣ) (t : ℕ) : (seg w x t).card = t

theorem **locL_segFam** (z : HeightOneSpectrum (𝒪 K) → ℂˣ) (t : ℕ) (w : HeightOneSpectrum (𝒪 K)) (s : ℂ) : locL (segFam z t) w s = ∏ i ∈ Finset.range t, locL (fun u => {z u}) w (s - ((t : ℂ) - 1) / 2 + i)

theorem **locL_segFam'** (z : HeightOneSpectrum (𝒪 K) → ℂˣ) (t : ℕ) (w : HeightOneSpectrum (𝒪 K)) (s : ℂ) : locL (segFam z t) w s = ∏ i ∈ Finset.range t, locL (fun u => {z u}) w (s + ((t : ℂ) - 1) / 2 - i)

theorem **Converges.hasProd_and_ne_zero** (h : Converges T c s) : HasProd (fun w : {w // w ∉ T} => locL c w s) (eulerProd T c s) ∧ eulerProd T c s ≠ 0

theorem **Converges.hasProd** (h : Converges T c s) : HasProd (fun w : {w // w ∉ T} => locL c w s) (eulerProd T c s)

theorem **Converges.multipliable** (h : Converges T c s) : Multipliable fun w : {w // w ∉ T} => locL c w s

theorem **Converges.ne_zero** (h : Converges T c s) : eulerProd T c s ≠ 0

theorem **eulerProd_add** (h : Converges T c s) (h' : Converges T c' s) : eulerProd T (c + c') s = eulerProd T c s * eulerProd T c' s

theorem **eulerProd_congr** (h : ∀ w ∉ T, c w = c' w) (s : ℂ) : eulerProd T c s = eulerProd T c' s

theorem **Converges.congr** (h : Converges T c s) (e : ∀ w ∉ T, c w = c' w) : Converges T c' s

theorem **Converges.eulerProd_eq_inv** (h : Converges T c s) : eulerProd T c s = (∏' w : {w // w ∉ T}, eulerPoly (c w) (npow w.1 (-s)))⁻¹

theorem **Bounded.norm_le** {r : ℕ} {κ σ : ℝ} (h : Bounded T c r κ) (hs : σ ≤ s.re) {w : HeightOneSpectrum (𝒪 K)} (hw : w ∉ T) {b : ℂˣ} (hb : b ∈ c w) : ‖(b : ℂ) * npow w (-s)‖ ≤ ((Ideal.absNorm w.asIdeal : ℕ) : ℝ) ^ (-(σ - κ))

theorem **Bounded.norm_eulerPoly_le** {r : ℕ} {κ σ : ℝ} (h : Bounded T c r κ) (hσ : κ ≤ σ) (hs : σ ≤ s.re) {w : HeightOneSpectrum (𝒪 K)} (hw : w ∉ T) : ‖eulerPoly (c w) (npow w (-s)) - 1‖ ≤ 2 ^ r * ((Ideal.absNorm w.asIdeal : ℕ) : ℝ) ^ (-(σ - κ))

theorem **summable_rpow** (T : Set (HeightOneSpectrum (𝒪 K))) {σ : ℝ} (hσ : 1 < σ) : Summable fun w : {w // w ∉ T} => ((Ideal.absNorm w.1.asIdeal : ℕ) : ℝ) ^ (-σ)

theorem **Bounded.converges** {r : ℕ} {κ : ℝ} (h : Bounded T c r κ) (hs : κ + 1 < s.re) : Converges T c s

theorem **differentiable_eulerPoly** (B : Multiset ℂˣ) (w : HeightOneSpectrum (𝒪 K)) : Differentiable ℂ fun s => eulerPoly B (npow w (-s))

theorem **Bounded.differentiableOn** {r : ℕ} {κ : ℝ} (h : Bounded T c r κ) : DifferentiableOn ℂ (eulerProd T c) {s | κ + 1 < s.re}

theorem **Bounded.twist** {r : ℕ} {κ : ℝ} (h : Bounded T c r κ) {z : HeightOneSpectrum (𝒪 K) → ℂˣ} (hz : ∀ w ∉ T, ‖(z w : ℂ)‖ = 1) : Bounded T (twist c z) r κ

theorem **Bounded.add** {r r' : ℕ} {κ : ℝ} (h : Bounded T c r κ) (h' : Bounded T c' r' κ) : Bounded T (c + c') (r + r') κ

theorem **IsCont.unique** {σ : ℝ} {Λ Λ' : ℂ → ℂ} (h : IsCont T c σ Λ) (h' : IsCont T c σ Λ') : Set.EqOn Λ Λ' {s | σ < s.re}

theorem **heckeFam_of_unrAt** {χ : HChar K} {w : HeightOneSpectrum (𝒪 K)} (h : χ.UnrAt w) : heckeFam χ w = {χ.val w}

theorem **heckeFam_of_not_unrAt** {χ : HChar K} {w : HeightOneSpectrum (𝒪 K)} (h : ¬ χ.UnrAt w) : heckeFam χ w = 0

theorem **locL_heckeFam** (χ : HChar K) (w : HeightOneSpectrum (𝒪 K)) (s : ℂ) : locL (heckeFam χ) w s = GlobalTheory.CMVal.eulerFactor χ s w

theorem **heckeFam_one** : heckeFam (1 : HChar K) = fun _ => {1}

theorem **heckeL_one** (T : Set (HeightOneSpectrum (𝒪 K))) (s : ℂ) : heckeL T 1 s = zetaOff T s

theorem **bounded_heckeFam** (T : Set (HeightOneSpectrum (𝒪 K))) {χ : HChar K} (h : χ.IsUnitary) : Bounded T (heckeFam χ) 1 0

theorem **converges_heckeFam** (T : Set (HeightOneSpectrum (𝒪 K))) {χ : HChar K} (h : χ.IsUnitary) {s : ℂ} (hs : 1 < s.re) : Converges T (heckeFam χ) s

theorem **converges_zeta** (T : Set (HeightOneSpectrum (𝒪 K))) {s : ℂ} (hs : 1 < s.re) : Converges T (fun _ => {1}) s

theorem **mem_aboveF** {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} : w ∈ aboveF (L

theorem **LvTw_eq** (c : Fam E) (m : HeightOneSpectrum (𝒪 E) → ℂˣ) (v : HeightOneSpectrum (𝒪 F)) (s : ℂ) : LvTw c m v s = ∏ w ∈ aboveF (L

theorem **Lv_add** (c c' : Fam E) (v : HeightOneSpectrum (𝒪 F)) (s : ℂ) : Lv (c + c') v s = Lv c v s * Lv c' v s

theorem **hasProd_Lv** {S : Set (HeightOneSpectrum (𝒪 F))} {c : Fam E} {s : ℂ} (h : Converges (above E S) c s) : HasProd (fun v : {v // v ∉ S} => Lv c v.1 s) (LS S c s)

theorem **LvTw_inert** (c : Fam E) (m : HeightOneSpectrum (𝒪 E) → ℂˣ) {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (hv : ∀ u, u.under (𝒪 F) = v ↔ u = w) (hq : Ideal.absNorm w.asIdeal = Ideal.absNorm v.asIdeal ^ 2) (s : ℂ) : LvTw c m v s = eulerFac (c w) ((m w : ℂ)⁻¹ * npow v (-(2 * s)))

theorem **etaU_sq** (v : HeightOneSpectrum (𝒪 F)) : etaU E v ^ 2 = 1

theorem **etaFam_of_unram** {v : HeightOneSpectrum (𝒪 F)} (h : IsUnram (E

theorem **etaFam_add_two_mul** (k j : ℕ) : etaFam (F

theorem **bounded_etaFam** (S : Set (HeightOneSpectrum (𝒪 F))) (k : ℕ) : Bounded S (etaFam E k) 1 0

theorem **converges_etaFam** (S : Set (HeightOneSpectrum (𝒪 F))) (k : ℕ) {s : ℂ} (hs : 1 < s.re) : Converges S (etaFam E k) s

theorem **dv_center** (a t : ℕ) (v : HeightOneSpectrum (𝒪 F)) : dv E a (a + t) v ((t : ℂ) / 2) = ∏ k ∈ Finset.Icc (t + 1) (t + a), locL (etaFam E k) v k

theorem **hasProd_dv** (S : Set (HeightOneSpectrum (𝒪 F))) (a b : ℕ) {s : ℂ} (hs : 0 < s.re) : HasProd (fun v : {v // v ∉ S} => dv E a b v.1 s) (∏ i ∈ Finset.range a, etaL E S (b + i) (2 * s + a - i))

theorem **dSfun_eq** (S : Set (HeightOneSpectrum (𝒪 F))) (a b : ℕ) {s : ℂ} (hs : 0 < s.re) : dSfun E S a b s = ∏ i ∈ Finset.range a, etaL E S (b + i) (2 * s + a - i)

theorem **dS_eq** (S : Set (HeightOneSpectrum (𝒪 F))) (a : ℕ) {t : ℕ} (ht : 1 ≤ t) : dS E S a (a + t) = ∏ k ∈ Finset.Icc (t + 1) (t + a), etaL E S k k ∧ dS E S a (a + t) ≠ 0

Development.AbelianVarietyTheory.Definitions.Foundations.NormOneSevenOneTwo

theorem **normOne_seven_one_two** : NormOne 7 1 2

theorem **normOne_seven_two** : NormOne 7 2 29

theorem **normOne_dvd** {d M p : ℕ} (hd : d % 4 = 3) (h : NormOne d M p) : (M : ℤ) ∣ (p : ℤ) - 1

theorem **not_normOne** : ¬ NormOne 7 2 2

theorem **span_mem_principalRayGroup_top** {K : Type} [Field K] [NumberField K] (a : 𝒪 K) (I : (FractionalIdeal (𝒪 K)⁰ K)ˣ) (h : (I : FractionalIdeal (𝒪 K)⁰ K) = ((Ideal.span {a} : Ideal (𝒪 K)) : FractionalIdeal (𝒪 K)⁰ K)) : I ∈ rayP (⊤ : Ideal (𝒪 K))

theorem **mem_rayI_of_coprime** {K : Type} [Field K] [NumberField K] (q : Ideal (𝒪 K)) (J : Ideal (𝒪 K)) : J ≠ 0 → J ⊔ q = ⊤ → ∀ I : (FractionalIdeal (𝒪 K)⁰ K)ˣ, (I : FractionalIdeal (𝒪 K)⁰ K) = (J : FractionalIdeal (𝒪 K)⁰ K) → I ∈ rayI q

theorem **principalRayGroup_le_coprimeIdealGroup** {K : Type} [Field K] [NumberField K] (q : Ideal (𝒪 K)) : rayP q ≤ rayI q

private theorem **finite_zeros_in_strip_box_of_differentiable** {f : ℂ → ℂ} (hf : Differentiable ℂ f) (hne : ∃ z, f z ≠ 0) (t T : ℝ) : {ρ : ℂ | 0 < ρ.re ∧ ρ.re < 1 ∧ |ρ.im - t| ≤ T ∧ f ρ = 0}.Finite

theorem **zerosIn_finite** {q : ℕ} [NeZero q] (χ : DirichletCharacter ℂ q) (hχ : χ ≠ 1) (t T : ℝ) : (zerosIn χ t T).Finite

private theorem **finite_zeros_in_strip_box_of_differentiable'** {f : ℂ → ℂ} (hf : Differentiable ℂ f) (hne : ∃ z, f z ≠ 0) (t T : ℝ) : {ρ : ℂ | 0 < ρ.re ∧ ρ.re < 1 ∧ |ρ.im - t| ≤ T ∧ f ρ = 0}.Finite

theorem **zerosIn_finite_one** {q : ℕ} [NeZero q] (t T : ℝ) : (zerosIn (1 : DirichletCharacter ℂ q) t T).Finite

theorem **zmult_pos** {q : ℕ} [NeZero q] (χ : DirichletCharacter ℂ q) (hχ : χ ≠ 1) (ρ : ℂ) (h : DirichletCharacter.LFunction χ ρ = 0) : 0 < zmult χ ρ

Development.AbelianVarietyTheory.Definitions.Foundations.NormSumPairRankZeroOfCurveCoordNorm

theorem **inc_inj** (D : D7) : Function.Injective (inc D)

theorem **ratG_eq_ratPts** (D : D7) (N : ℕ) : ratG D N = (Formalism.UG.ratPts (F

theorem **mem_ratG** {D : D7} {N : ℕ} (g : GA D N) : g ∈ ratG D N ↔ ∃ γ, IsRat D (g : GL (Fin N) (AE D)) γ

theorem **mem_intG** {D : D7} {N : ℕ} (k : GA D N) : k ∈ intG D N ↔ IntG D (k : GL (Fin N) (AE D))

theorem **cls_eq_classSet** (D : D7) (N : ℕ) (J : Subgroup (GA D N)) : Cls D N J = Formalism.UG.ClassSet (ratG D N) J

theorem **gam_eq_card_stabAt** (D : D7) {N : ℕ} (J : Subgroup (GA D N)) (g : GA D N) : gam D J g = Nat.card (Formalism.UG.stabAt (ratG D N) J g)

theorem **gam_ne_zero_iff** (D : D7) {N : ℕ} (J : Subgroup (GA D N)) (g : GA D N) : gam D J g ≠ 0 ↔ Finite (Formalism.UG.stabAt (ratG D N) J g)

theorem **Inp.of_conditions** (hA : AdmD D) (hP : aux₃ D) (hM : GL3Theta.modelEx D) (hE : ∀ cc δ t0 N N' e0, StatementOddExactness D cc δ t0 N N' e0) : Inp D

theorem **Pair.even_mul** (P : Pair D cc δ t0 Q) : Even (P.a * P.d)

theorem **Pair.eqv** (P : Pair D cc δ t0 Q) (h : Inp D) : oddEq D cc δ P.a P.d P.e0

theorem **Pair.level_pos** (P : Pair D cc δ t0 Q) : 0 < Q

theorem **Pair.sq_nonneg** (P : Pair D cc δ t0 Q) (h : GA D P.d) : 0 ≤ P.sq h

theorem **Pair.norm_nonneg** (P : Pair D cc δ t0 Q) : 0 ≤ P.norm

theorem **Pair.term_mul_intG** (P : Pair D cc δ t0 Q) (hF : P.F ∈ TT D (tor0 D cc δ) P.a P.d) (x : GA D P.a) {k : GA D P.a} (hk : k ∈ intG D P.a) (h : GA D P.d) (T : Tup D P.a P.d P.n) : P.term (x ∗ k) h T = P.term x h T

theorem **wd_zero** (D : D7) (T : Tor D) {a d : ℕ} (G : VC D a d → ℂ) : Wd D T 0 G = G

theorem **pres_self** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) {a d : ℕ} (e0 : VC D a d → ℂ) : Pres D cc δ 1 e0 e0

theorem **Pair.ofVacuum_F_ne_zero** (crv : Crv D cc δ) (pt0 : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t0) {a d : ℕ} (had : a ≤ d) (hd : Even d) {e0 : VC D a d → ℂ} (vac : oddE0 D cc δ t0 a d e0) (W : WeilO D cc δ a d e0) (n : D.what₂.Tau → ℕ) (wt) (hfix) (heig) : (Pair.ofVacuum crv pt0 had hd vac W n wt hfix heig).F ≠ 0

theorem **exists_crv_top** (D : D7) : ∃ δ : ℂ, Crv D ⊤ δ

theorem **hf0** (D : D7) (T : Tor D) (z w : VC D 0 0 0) : Hf D T z w = 0

theorem **iop0** (D : D7) (T : Tor D) (A B : AddSubgroup (VC D 0 0 0)) (G : VC D 0 0 0 → ℂ) : Iop D T A B G = G

theorem **taylor0** (D : D7) (α : Mono D 0 0) (c : ℂ) : taylor D (fun _ => c) α = c

theorem **normSq0** (D : D7) (T : Tor D) (A : AddSubgroup (VC D 0 0 0)) : normSq D T A (fun _ => 1) = 1

theorem **vac0** (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ t0 : ℂ) : oddE0 D cc δ t0 0 0 (fun _ => 1)

theorem **gam0** (D : D7) (J : Subgroup (GA D 0)) (g : GA D 0) : gam D J g = 1

theorem **normSumPairRankZeroOfCurve_coord_norm** (crv : Crv D cc δ) (pt0 : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t0) : (∀ h T, (pair0 crv pt0).coord h T = 1) ∧ (pair0 crv pt0).norm = 1

Development.AbelianVarietyTheory.Definitions.Foundations.NotCoboundaryConstantOne

theorem **natCast_pow_dvd_iff** (k : ℕ) (f : Lau p) : (p : Lau p) ^ k ∣ f ↔ ∀ n, (p : ℤ_[p]) ^ k ∣ f.coeff n

theorem **dvd_iff_map** (f : Lau p) : (p : Lau p) ∣ f ↔ f.map (PadicInt.toZMod (p

theorem **prime_lau** : Prime (p : Lau p)

theorem **mem_pI_pow** (k : ℕ) (f : Lau p) : f ∈ pI p ^ k ↔ (p : Lau p) ^ k ∣ f

theorem **seq_sub_dvd** (x : AdicCompletion.AdicCauchySequence (pI p) (Lau p)) {m n : ℕ} (h : m ≤ n) : (p : Lau p) ^ m ∣ x.1 n − x.1 m

theorem **exists_not_dvd** (x : AdicCompletion.AdicCauchySequence (pI p) (Lau p)) (hx : AdicCompletion.mk (pI p) (Lau p) x ≠ 0) : ∃ a, ∀ n, a ≤ n → ¬ (p : Lau p) ^ a ∣ x.1 n

theorem **not_isUnit_p** : ¬ IsUnit (p : O0 p)

theorem **isUnit_qO** : IsUnit (qO p)

theorem **isClo_algebraicClosure** : IsClo p (AlgebraicClosure (FractionRing (O0 p))) where

theorem **isDlog_yn** (n : ℕ) : IsDlog p n (yn p n) 1 0

theorem **isDlog_xn** (n : ℕ) : IsDlog p n (xn p n) 0 1

theorem **fPi_yn_xn** (n : ℕ) : fPi p n 0 0 1 0 0 1 = xn p n − 1

theorem **fPi_swap** (n : ℕ) (k l : ℤ) (α β α' β' : On p n) : fPi p n l k α' β' α β = − fPi p n k l α β α' β'

theorem **charZero_of_isClo** (h : IsClo p Ω) : CharZero Ω

theorem **tw_prim_step** {Ω : Type∗} [Field Ω] {ζ z : Ω} {n : ℕ} (hζ : IsPrimitiveRoot ζ (p ^ (n + 1))) (hz : z ^ p = ζ) : IsPrimitiveRoot z (p ^ (n + 2))

theorem **tw_exists_prim_p** (Ω : Type∗) [Field Ω] [IsAlgClosed Ω] [CharZero Ω] : ∃ z : Ω, IsPrimitiveRoot z p

theorem **tw_exists_root_chain** {Ω : Type∗} [Field Ω] [IsAlgClosed Ω] (a : Ω) : ∃ f : ℕ → Ω, f 0 = a ∧ ∀ n, f (n + 1) ^ p = f n

theorem **nonempty_tower** (h : IsClo p Ω) : Nonempty (Tower p Ω)

theorem **qq_pow** (T : Tower p Ω) (n : ℕ) : T.qq n ^ p ^ n = algebraMap (O0 p) Ω (qO p)

theorem **qq_ne_zero** (T : Tower p Ω) (n : ℕ) : T.qq n ≠ 0

theorem **exists_isEmb** (T : Tower p Ω) (n : ℕ) : ∃ ι : On p n →ₐ[O0 p] Ω, IsEmb T n ι

theorem **gms_self** (T : Tower p Ω) (n : ℕ) : Gms T n n = Gn T n

theorem **mem_gms** (T : Tower p Ω) (m s : ℕ) (g : Gal p Ω) : g ∈ Gms T m s ↔ g (T.ζ m) = T.ζ m ∧ g (T.qq s) = T.qq s

theorem **mem_gn** (T : Tower p Ω) (n : ℕ) (g : Gal p Ω) : g ∈ Gn T n ↔ g (T.ζ n) = T.ζ n ∧ g (T.qq n) = T.qq n

theorem **isInt_gal** (g : Gal p Ω) {x : Ω} (hx : IsInt p x) : IsInt p (g x)

theorem **isInt_intCast** (m : ℤ) : IsInt p (m : Ω)

theorem **cong_refl** (k : ℕ) (x : Ω) : Cong p k x x

theorem **cong_symm** {k : ℕ} {x y : Ω} (h : Cong p k x y) : Cong p k y x

theorem **cong_trans** {k : ℕ} {x y z : Ω} (h : Cong p k x y) (h' : Cong p k y z) : Cong p k x z

theorem **cong_add** {k : ℕ} {x y x' y' : Ω} (h : Cong p k x y) (h' : Cong p k x' y') : Cong p k (x + x') (y + y')

theorem **cong_sub** {k : ℕ} {x y x' y' : Ω} (h : Cong p k x y) (h' : Cong p k x' y') : Cong p k (x − x') (y − y')

theorem **cong_mul_right** {k : ℕ} {x y c : Ω} (h : Cong p k x y) (hc : IsInt p c) : Cong p k (x ∗ c) (y ∗ c)

theorem **cong_gal** {k : ℕ} {x y : Ω} (g : Gal p Ω) (h : Cong p k x y) : Cong p k (g x) (g y)

theorem **cong_natCast** {k a b : ℕ} (h : a ≡ b [MOD p ^ k]) : Cong p k (a : Ω) (b : Ω)

theorem **not_cong_one_zero** (h : IsClo p Ω) (k : ℕ) (hk : k ≠ 0) : ¬ Cong p k (1 : Ω) 0

theorem **zeta_pow_eq_iff** (T : Tower p Ω) (k a b : ℕ) : T.ζ k ^ a = T.ζ k ^ b ↔ a ≡ b [MOD p ^ k]

theorem **exists_isKum** (T : Tower p Ω) (n : ℕ) (H : Subgroup (Gal p Ω)) {r : Ω} (hr : r ≠ 0) (hH : ∀ g ∈ H, g (r ^ p ^ n) = r ^ p ^ n) : ∃ a, IsKum T n H r a

theorem **exists_isKum_qq** (T : Tower p Ω) (n : ℕ) : ∃ a, IsKum T n ⊤ (T.qq n) a

theorem **isKum_unique** {T : Tower p Ω} {n : ℕ} {H : Subgroup (Gal p Ω)} {r : Ω} (hr : r ≠ 0) {a b : Gal p Ω → ZMod (p ^ n)} (ha : IsKum T n H r a) (hb : IsKum T n H r b) {g : Gal p Ω} (hg : g ∈ H) : a g = b g

theorem **exists_isCyc** (T : Tower p Ω) (k : ℕ) : ∃ χ : Gal p Ω → ZMod (p ^ k), ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ g).val

theorem **isCyc_unique** {T : Tower p Ω} {k : ℕ} {χ χ' : Gal p Ω → ZMod (p ^ k)} (h : ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ g).val) (h' : ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ' g).val) : χ = χ'

theorem **isCyc_mul** {T : Tower p Ω} {k : ℕ} {χ : Gal p Ω → ZMod (p ^ k)} (h : ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ g).val) (g g' : Gal p Ω) : χ (g * g') = χ g * χ g'

theorem **isCyc_eq_one_iff** {T : Tower p Ω} {k : ℕ} {χ : Gal p Ω → ZMod (p ^ k)} (h : ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ g).val) (g : Gal p Ω) : χ g = 1 ↔ g (T.ζ k) = T.ζ k

theorem **isCyc_pow_totient** {T : Tower p Ω} {k : ℕ} {χ : Gal p Ω → ZMod (p ^ k)} (h : ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ g).val) (g : Gal p Ω) : χ g ^ Nat.totient (p ^ k) = 1

theorem **isKum_mul** {T : Tower p Ω} {n : ℕ} {χ : Gal p Ω → ZMod (p ^ n)} (hχ : ∀ g : Gal p Ω, g (T.ζ n) = T.ζ n ^ (χ g).val) {H : Subgroup (Gal p Ω)} {r : Ω} (hr : r ≠ 0) {a : Gal p Ω → ZMod (p ^ n)} (ha : IsKum T n H r a) {g h : Gal p Ω} (hg : g ∈ H) (hh : h ∈ H) : a (g * h) = a g + χ g * a h

theorem **isKum_cocycle** {T : Tower p Ω} {n : ℕ} {χ : Gal p Ω → ZMod (p ^ n)} (hχ : ∀ g : Gal p Ω, g (T.ζ n) = T.ζ n ^ (χ g).val) {H : Subgroup (Gal p Ω)} {r : Ω} (hr : r ≠ 0) {a : Gal p Ω → ZMod (p ^ n)} (ha : IsKum T n H r a) {g h : Gal p Ω} (hg : g ∈ H) (hh : h ∈ H) : Cong p n (((χ g).val : Ω) ^ 1 * g ((a h).val : Ω) - ((a (g * h)).val : Ω) + ((a g).val : Ω)) 0

theorem **cong_twist** {T : Tower p Ω} {k : ℕ} {χ : Gal p Ω → ZMod (p ^ k)} (hχ : ∀ g : Gal p Ω, g (T.ζ k) = T.ζ k ^ (χ g).val) (j : ℕ) {g : Gal p Ω} (hg : g (T.ζ k) = T.ζ k) {x : Ω} (hx : IsInt p x) : Cong p k (((χ g).val : Ω) ^ j * g x) (g x)

theorem **not_coboundary_constant_one** (h : IsClo p Ω) {k : ℕ} (hk : k ≠ 0) (H : Subgroup (Gal p Ω)) : ¬ Cob1 k H (fun _ => (1 : Ω))

theorem **inhabited_setting** (p : ℕ) [Fact p.Prime] : ∃ (Ω : Type) (_ : Field Ω) (_ : Algebra (O0 p) Ω), IsClo p Ω ∧ ∃ T : Tower p Ω, ∀ n, (∃ ι, IsEmb T n ι) ∧ (∃ χ : Gal p Ω → ZMod (p ^ n), ∀ g : Gal p Ω, g (T.ζ n) = T.ζ n ^ (χ g).val) ∧ ∃ a, IsKum T n ⊤ (T.qq n) a

Development.AbelianVarietyTheory.Definitions.Foundations.NotFlatExistsOfIsEmpty

theorem **exists_vAdd_two_ne** (K : Type) [Field K] [NumberField K] : ∃ w : HeightOneSpectrum (𝒪 K), vAdd w (2 : K) ≠ 0

theorem **exists_constSec** (h1 : N1c) (h2 : GlobalTheory.Published.HasseNormSt) (Xo : Moduli Dw l) (c : Coeff l) (hpq : 1 ≤ Dw.p + Dw.q) (hg : ∀ p : ℕ, p.Prime → l.GoodAt p) (x : ℝ) : ∃ U : ∀ σ : c.F →+* ℂ, Unf Xo c σ (Dw.p + Dw.q), (∀ σ, GoodEmb c σ → IsUnf (U σ)) ∧ Nonempty (Comp Xo c) ∧ ∀ Z : Comp Xo c, ∃ s : ConstSec U x Z 0 0, (∀ σ, s.cInf σ = 2) ∧ (∃ w, s.rho_vw w ≠ 0) ∧ s.deg = 0

theorem **nonempty_psi** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : Nonempty (Psi H)

theorem **not_flatExists_of_isEmpty** (hpq : 1 ≤ Dw.p + Dw.q) (h : ∀ σ, IsEmpty (U σ).Comp) : ¬ FlatExists U x

theorem **flatExistsAt_unique** (hpq : 1 ≤ Dw.p + Dw.q) {a b : ℝ} (ha : FlatExistsAt U x a) (hb : FlatExistsAt U x b) : a = b

theorem **pnorm_modelFlat** (X : Data E r) (τ : X.Ψ) (P : Frame X.Ψ r → ℂ) {y : Frame X.Ψ r} (hy : y ∈ X.dom) : X.pnorm (modelFlat r τ) P y = ‖P y‖ * (‖(mat y τ).det‖ * Real.sqrt ‖τ.1 X.Gm.det‖)

theorem **norm_rdU_sig** (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) {Z : Comp Xo c} {τ : Psi H} (s : FlatSec U x Z τ) (σ : c.F →+* ℂ) (k : (U (goodOf c σ)).Comp) : ∃ (hτ : relab c (goodOf c σ) τ.1 ∈ ((U (goodOf c σ)).X k).Ψ) (a : ℝ), 0 < a ∧ ∀ y, (⟨k, y⟩ : Σ k, Frame ((U (goodOf c σ)).X k).Ψ r) ∈ framesOn (U (goodOf c σ)) Z → ‖rdU (U (goodOf c σ)) s.sig k y‖ = a * ‖horizGen ⟨_, hτ⟩ y‖

theorem **mat_double** {Ψ : Finset (E →+* ℂ)} (y : Frame Ψ r) (τ : Ψ) : mat (rmul (lact 1 y) (twoFr Ψ r)) τ = (2 : ℂ) • mat y τ

theorem **double_mem** {X : Data E r} {y : Frame X.Ψ r} (hy : y ∈ X.dom) : rmul (lact 1 y) (twoFr X.Ψ r) ∈ X.dom

theorem **flatSec_ne_const** (hpq : 1 ≤ Dw.p + Dw.q) (hU : ∀ σ, GoodEmb c σ → IsUnf (U σ)) {Z : Comp Xo c} {τ : Psi H} (s : FlatSec U x Z τ) (a : c.F) : s.sig ≠ a • idem Z

theorem **sum_flatWt** (τ : Psi H) : ∑ ρ : E →+* ℂ, ∑ i : Fin (Dw.mult ρ), (((wtOf Dw (flatWt Dw τ) 0).1 (ρ, i) : ℤ) : ℝ) = Dw.q - Dw.p

theorem **deg_rescale** {Z : Comp Xo c} {τ : Psi H} (s : FlatSec U x Z τ) (x' : ℝ) : (rescale s x').deg = s.deg - ((Dw.p : ℝ) - Dw.q) / 2 * (Real.log (2 * Real.pi * x') - Real.log (2 * Real.pi * x))

theorem **flatExistsAt_rescale** {a : ℝ} (h : FlatExistsAt U x a) (x' : ℝ) : FlatExistsAt U x' (a + ((Dw.p : ℝ) - Dw.q) / 2 * (Real.log (2 * Real.pi * x') - Real.log (2 * Real.pi * x)))

theorem **flDefect_vp_cyclotomic** (h2 : finrank ℚ E = 12) (hd : (discr E).natAbs = 3 ^ 6 * 7 ^ 10) (h7 : H.d = 7) : flDefect_vp Dw 3 = Dw.q / 2 ∧ flDefect_vp Dw 7 = -(Dw.q / 6)

Development.AbelianVarietyTheory.Definitions.Foundations.NotHasParamOfSubsingleton

theorem **constMap_injective** (G : Type) [Group G] : Function.Injective (constMap G)

theorem **chiDelta_rhoGL** (b : GL (Fin n) (w.adicCompletion E)) : chiDelta E w (fun i => (rhoGL w n i : ℂ)) b = 1

theorem **hasSatGL_trivial** : HasSatGL w (Representation.trivial ℂ (GL (Fin n) (w.adicCompletion E)) ℂ) (seg w 1 n)

theorem **not_hasSatGL_trivial_one** : ¬ HasSatGL w (Representation.trivial ℂ (GL (Fin 2) (w.adicCompletion E)) ℂ) {1, 1}

theorem **psFunGL_one_apply** (a : ℂˣ) (t : GL (Fin 1) (w.adicCompletion E)) (ht : IsUnifV E w (t.1 0 0)) (φ : psSpaceGL w (fun _ : Fin 1 => a)) : HasSatGL w (psFunGL w fun _ : Fin 1 => a) {a} ∧ psFunGL w (fun _ : Fin 1 => a) t φ = (a : ℂ) • φ

theorem **hasSatU_trivial** (hU : IsUnramInv ϖ σ q) (hq : q ≠ 0) (n : ℕ) : HasSatU ϖ q (Representation.trivial ℂ (GU σ n) ℂ) (rhoU hq n)

theorem **paramOf_trivial_split** (hv : IsSplit (E

theorem **hasParam_trivial_inert** (hw : IsInertAt F E v w) {c : w.adicCompletion E →+* w.adicCompletion E} {ϖ : w.adicCompletion E} (hU : IsUnramInv ϖ c (qV F v)) {e : GLoc σ J v ≃* GU c n} (he : IsInertChart σ J L h c e) : HasParam σ J L h (Representation.trivial ℂ _ ℂ) (bcParam n (rhoU (qV_ne_zero v) n))

theorem **toCompl_surjective** (h : w.under (𝒪 F) = v) : Function.Surjective (toCompl h)

theorem **isConjAt_unique** {c c' : w.adicCompletion E →+* w.adicCompletion E} (hc : IsConjAt σ h c) (hc' : IsConjAt σ h c') : c = c'

theorem **not_hasParam_id** (hw : IsInertAt F E v w) {V : Type} [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ (GLoc (AlgHom.id F E) J v) V) (B : Multiset ℂˣ) : ¬ HasParam (AlgHom.id F E) J L h ρ B

theorem **not_hasParam_of_subsingleton** {V : Type} [AddCommGroup V] [Module ℂ V] [Subsingleton V] (ρ : Representation ℂ (GLoc σ J v) V) (B : Multiset ℂˣ) : ¬ HasParam σ J L h ρ B

theorem **not_isFrameAt_bot** (a : GL (Fin (n + 1)) (w.adicCompletion E)) : ¬ IsFrameAt w ⊥ a

theorem **locL_trivial** (s : ℂ) : locL (segFam (fun _ : HeightOneSpectrum (𝒪 E) => 1) n) w s = ∏ i ∈ Finset.range n, (1 - npow w (-(s - ((n : ℂ) - 1) / 2 + i)))⁻¹

Development.AbelianVarietyTheory.Definitions.Foundations.NotIsSelfDualAtScaled

theorem **isHermND_hyp** : IsHermND σ (hyp E)

theorem **tmul_mul_inv** {v : HeightOneSpectrum (𝒪 F)} (a : (Fv F v)ˣ) : ((a : Fv F v) ⊗ₜ[F] (1 : E)) ∗ (((a⁻¹ : (Fv F v)ˣ) : Fv F v) ⊗ₜ[F] (1 : E)) = 1

theorem **inv_mul_tmul** {v : HeightOneSpectrum (𝒪 F)} (a : (Fv F v)ˣ) : (((a⁻¹ : (Fv F v)ˣ) : Fv F v) ⊗ₜ[F] (1 : E)) ∗ ((a : Fv F v) ⊗ₜ[F] (1 : E)) = 1

theorem **torMat_mul_inv** {v : HeightOneSpectrum (𝒪 F)} (a : (Fv F v)ˣ) : torMat E a ∗ torMat E a⁻¹ = 1

theorem **normFin_torMat** {v : HeightOneSpectrum (𝒪 F)} (a : (Fv F v)ˣ) : normFin F v (torMat E a) = max ‖(a : Fv F v)‖₊ ‖(a : Fv F v)‖₊⁻¹

theorem **height_emb_tor** {v : HeightOneSpectrum (𝒪 F)} (a : (Fv F v)ˣ) : height σ (hyp E) (emb σ (hyp E) v (tor σ a)) = max ‖(a : Fv F v)‖₊ ‖(a : Fv F v)‖₊⁻¹

theorem **tor_mem_stabLoc** {v : HeightOneSpectrum (𝒪 F)} (a : (Fv F v)ˣ) : tor σ a ∈ stabLoc σ (hyp E) (stdLattice E 2) v ↔ ‖(a : Fv F v)‖₊ = 1

theorem **exists_one_lt** (v : HeightOneSpectrum (𝒪 F)) : ∃ a : (Fv F v)ˣ, 1 < ‖(a : Fv F v)‖

theorem **stabLoc_ne_top** (v : HeightOneSpectrum (𝒪 F)) : stabLoc σ (hyp E) (stdLattice E 2) v ≠ ⊤

theorem **height_unbounded** (c : ℝ) : ∃ g : GAd σ (hyp E), c ≤ height σ (hyp E) g

theorem **not_moderateGrowth_exp** : ¬ ModerateGrowth σ (hyp E) fun g => Real.exp (height σ (hyp E) g)

theorem **isSelfDualAt_hyp** (v : HeightOneSpectrum (𝒪 F)) : IsSelfDualAt σ (hyp E) (stdLattice E 2) v

theorem **height_ofInf_definite** (h : IsCMExt F E σ) {n : ℕ} (w : InfinitePlace F) (x : GArch σ (1 : Matrix (Fin n) (Fin n) E) w) : height σ 1 (embArch σ 1 w x) = 1

theorem **map_map_tens** {R R' R'' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] [CommRing R''] [Algebra F R''] (φ : R →ₐ[F] R') (ψ : R' →ₐ[F] R'') (y : R ⊗[F] E) : Algebra.TensorProduct.map ψ (AlgHom.id F E) (Algebra.TensorProduct.map φ (AlgHom.id F E) y) = Algebra.TensorProduct.map (ψ.comp φ) (AlgHom.id F E) y

theorem **adTens_eq_one** {y : AF F ⊗[F] E} (h1 : ∀ w, compInf F E w y = 1) (h2 : ∀ v, compFin F E v y = 1) : y = 1

theorem **idem_supp_iff** {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (v : HeightOneSpectrum (𝒪 F)) (ε : AF F) (e0 : ∀ w, evInf F w ε = 0) (e1 : evFin F v ε = 1) (e2 : ∀ u, u ≠ v → evFin F u ε = 0) (g : GAd σ J) : (∀ i j, algebraMap (AF F) (AF F ⊗[F] E) ε ∗ ((matHom _ σ J g).1 - 1) i j = ((matHom _ σ J g).1 - 1) i j) ↔ g ∈ (emb σ J v).range

theorem **suppF_iff** {D : D7} (V : HSp D) {v : HeightOneSpectrum (𝒪 D.host.F)} {ε : AF D.host.F} (h : IsPlaceIdem D v ε) (g : GA D V) : Formalism.IndefR.Mult.SuppF V (algebraMap (AF D.host.F) (AE D) ε) g ↔ g ∈ (emb D.sp.cm.sig V.gram v).range

theorem **suppAt_iff** {D : D7} (V : HSp D) {v : HeightOneSpectrum (𝒪 D.host.F)} {ε : AF D.host.F} (h : IsPlaceIdem D v ε) {e : AE D} (he : e + cA D e = algebraMap (AF D.host.F) (AE D) ε) (g : GA D V) : SuppAt e V g ↔ g ∈ (emb D.sp.cm.sig V.gram v).range

theorem **emb_mem_KsphC_iff** {D : D7} (V : HSp D) (v : HeightOneSpectrum (𝒪 D.host.F)) (x : GLoc D.sp.cm.sig V.gram v) : emb D.sp.cm.sig V.gram v x ∈ KsphC V ∅ ↔ x ∈ stabLoc D.sp.cm.sig V.gram (frameLat V) v

theorem **absFin_normalised** (v : HeightOneSpectrum (𝒪 F)) (a : Fv F v) : absFin F v (a ⊗ₜ[F] (1 : E)) = WithZeroMulInt.toNNReal (NumberField.HeightOneSpectrum.absNorm_ne_zero v) (Valued.v a)

theorem **height_rank_zero** (J : Matrix (Fin 0) (Fin 0) E) (g : GAd σ J) : height σ J g = 1

theorem **not_isSelfDualAt_scaled** (v : HeightOneSpectrum (𝒪 F)) (c : F) (hc0 : c ≠ 0) (hc : ‖algebraMap F (Fv F v) c‖ < 1) : ¬ IsSelfDualAt σ (Matrix.of fun _ _ : Fin 1 => algebraMap F E c) (stdLattice E 1) v

theorem **entry_le_opNorm** {n : ℕ} (y : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : ‖y i j‖₊ ≤ opNorm y

theorem **le_height_torArch** {w : InfinitePlace F} (a : w.Completionˣ) : ‖(a : w.Completion)‖₊ ≤ height σ (hyp E) (embArch σ (hyp E) w (torArch σ a))

Development.AbelianVarietyTheory.Definitions.Foundations.NotIsSelfDualETwo

theorem **_root_.GlobalTheory.TestLat.Model.mem_sumLat** (lam : ι → _root_.Formalism.BasicNotions.DomWeight n) (e : Fin N ≃ C × _root_.Formalism.L2.DIV.VVIdx lam) (T : C → Submodule (𝒪 K) (_root_.Formalism.L2.DIV.VV K lam)) (u : Fin N → K) : u ∈ _root_.GlobalTheory.TestLat.sumLat lam e T ↔ ∀ c, _root_.GlobalTheory.TestLat.valAt lam e c u ∈ T c

theorem **valAt_apply** (lam : ι → DomWeight n) (e : Fin N ≃ C × VVIdx lam) (c : C) (u : Fin N → K) : valAt lam e c u = ∑ i, u (e.symm (c, i)) • bVV K lam i

theorem **_root_.GlobalTheory.TestLat.Model.valAt_bijective** (lam : ι → _root_.Formalism.BasicNotions.DomWeight n) (e : Fin N ≃ C × _root_.Formalism.L2.DIV.VVIdx lam) : Function.Bijective fun (u : Fin N → K) (c : C) => _root_.GlobalTheory.TestLat.valAt lam e c u

theorem **_root_.GlobalTheory.TestLat.Model.sumGram_form** [StarRing K] [Fintype C] (lam : ι → _root_.Formalism.BasicNotions.DomWeight n) (e : Fin N ≃ C × _root_.Formalism.L2.DIV.VVIdx lam) (G : ι → Matrix (Fin n) (Fin n) K) (u w : Fin N → K) : star u ⬝ᵥ (_root_.GlobalTheory.TestLat.sumGram lam e G *ᵥ w) = ∑ c, _root_.Formalism.L2.DIV.pairK lam G (_root_.GlobalTheory.TestLat.valAt lam e c u) (_root_.GlobalTheory.TestLat.valAt lam e c w)

theorem **isMetric_one** : IsMetric (1 : Matrix (Fin N) (Fin N) K) fun _ => 1

theorem **isMetric_four** : IsMetric ((4 : K) • (1 : Matrix (Fin 1) (Fin 1) K)) fun _ => (2⁻¹ : ℂ) • (1 : Matrix (Fin 1) (Fin 1) ℂ)

theorem **not_isMetric_zero** (X : InfinitePlace K → Matrix (Fin 1) (Fin 1) ℂ) : ¬ IsMetric (0 : Matrix (Fin 1) (Fin 1) K) X

theorem **isDefGram_one** [StarRing K] : IsDefGram (1 : Matrix (Fin n) (Fin n) K)

theorem **mem_stdLat** (x : Fin n → K) : x ∈ stdLat K n ↔ ∀ a, x a ∈ (algebraMap (𝒪 K) K).range

theorem **isLattice_stdLat** : Submodule.IsLattice K (stdLat K n) where

theorem **map_mem_range** (σ : K →+* K) {x : K} (hx : x ∈ (algebraMap (𝒪 K) K).range) : σ x ∈ (algebraMap (𝒪 K) K).range

theorem **isSelfDualE_std** (σ : K →+* K) : IsSelfDualE σ (1 : Matrix (Fin n) (Fin n) K) (stdLat K n)

theorem **isSelfDualK_std** [StarRing K] : IsSelfDualK (1 : Matrix (Fin n) (Fin n) K) (stdLat K n)

theorem **not_isSelfDualE_two** (σ : K →+* K) : ¬ IsSelfDualE σ (1 : Matrix (Fin 1) (Fin 1) K) (Submodule.span (𝒪 K) {fun _ => (2 : K)})

theorem **spreadLe_wOne** (R : ℝ) : SpreadLe (fun _ : Unit => wOne) R ↔ 1 ≤ R

theorem **bound_one_two** : bound 1 2 1 = 8 * Real.log 2

Development.AbelianVarietyTheory.Definitions.Foundations.NotIsSplitOfIsInertAt

theorem **complMap_rfl** (w : HeightOneSpectrum (𝒪 E)) : complMap (rfl : w.under (𝒪 F) = _) = GlobalTheory.AdeleBaseChange.locMap F w

theorem **complMap_algebraMap** (h : w.under (𝒪 F) = v) (x : F) : complMap h (algebraMap F _ x) = algebraMap E _ (algebraMap F E x)

theorem **complMap_mem** (h : w.under (𝒪 F) = v) {a : Fv F v} (ha : a ∈ v.adicCompletionIntegers F) : complMap h a ∈ Ov E w

@[simp] theorem **toCompl_tmul** (h : w.under (𝒪 F) = v) (a : Fv F v) (x : E) : toCompl h (a ⊗ₜ[F] x) = complMap h a * algebraMap E _ x

theorem **IsFrameAt.inv_mul_mem** {a b : GL (Fin n) (w.adicCompletion E)} (ha : IsFrameAt w L a) (hb : IsFrameAt w L b) : a⁻¹ * b ∈ KGL w n

theorem **IsSplitChart.exists_conj** {h : w.under (𝒪 F) = v} {e e' : GLoc σ J v ≃* GL (Fin n) (w.adicCompletion E)} (he : IsSplitChart σ J L h e) (he' : IsSplitChart σ J L h e') : ∃ k ∈ KGL w n, ∀ g, e' g = k⁻¹ * e g * k

theorem **IsInertChart.exists_conj** {h : w.under (𝒪 F) = v} {c : w.adicCompletion E →+* w.adicCompletion E} {e e' : GLoc σ J v ≃* GU c n} (he : IsInertChart σ J L h c e) (he' : IsInertChart σ J L h c e') : ∃ k ∈ KU c n, ∀ g, e' g = k⁻¹ * e g * k

theorem **not_isSplit_of_isInertAt** (hw : IsInertAt F E v w) : ¬ IsSplit (E

theorem **IsParamVec.ne_zero** (hx : IsParamVec σ J L h ρ x B) : x ≠ 0

theorem **IsParamVec.mem_fixed** (hx : IsParamVec σ J L h ρ x B) : x ∈ fixed ρ (stabLoc σ J L v)

theorem **IsParamVec.card** (hx : IsParamVec σ J L h ρ x B) : B.card = n

theorem **HasParam.isParamVec** (hρ : HasParam σ J L h ρ B) (hx : x ∈ fixed ρ (stabLoc σ J L v)) (h0 : x ≠ 0) : IsParamVec σ J L h ρ x B

theorem **HasParam.exists_vec** (hρ : HasParam σ J L h ρ B) : ∃ x, IsParamVec σ J L h ρ x B

theorem **HasParam.isSpherical** (hρ : HasParam σ J L h ρ B) : GlobalTheory.LGrp.IsSpherical ρ (stabLoc σ J L v)

theorem **IsParamVec.isSatVecGL** (hx : IsParamVec σ J L h ρ x B) (hv : IsSplit (E

theorem **IsSatVecU.of_chart** {c : w.adicCompletion E →+* w.adicCompletion E} {e e' : GLoc σ J v ≃* GU c n} (he : IsInertChart σ J L h c e) (he' : IsInertChart σ J L h c e') {ϖ : w.adicCompletion E} {q : ℕ} {α : Fin (n / 2) → ℂˣ} (hx : IsSatVecU ϖ q (ρ.comp e.symm.toMonoidHom) x α) : IsSatVecU ϖ q (ρ.comp e'.symm.toMonoidHom) x α

theorem **IsParamVec.unique_of_isSplit** (hv : IsSplit (E

theorem **HasParam.hasParam_paramOf** (hρ : HasParam σ J L h ρ B) : HasParam σ J L h ρ (paramOf σ J L h ρ)

theorem **HasParam.paramOf_eq_of_isSplit** (hv : IsSplit (E

theorem **hasParam_psFunGL** (hv : IsSplit (E

theorem **HasParamOff.paramFam** {π : ∀ v, Representation ℂ (GLoc σ J v) (Vl v)} {c : Fam E} (h : HasParamOff σ J L S π c) : HasParamOff σ J L S π (paramFam σ J L π)

theorem **HasParamOff.card** {π : ∀ v, Representation ℂ (GLoc σ J v) (Vl v)} {c : Fam E} (h : HasParamOff σ J L S π c) {w : HeightOneSpectrum (𝒪 E)} (hw : w ∉ above E S) : (c w).card = n

theorem **isUnitary_toHChar** (χ : (AF F)ˣ →* Circle) (h : IsHeckeF F χ) : (toHChar χ h).IsUnitary

theorem **IsEta.isHeckeF** {η : (AF F)ˣ →* Circle} (h : IsEta F E η) : IsHeckeF F η

theorem **toFinAd_tmul** (a : Af F) (x : E) : toFinAd F E (a ⊗ₜ[F] x) = GlobalTheory.AdeleBaseChange.adMap F E a * algebraMap E _ x

theorem **toFinAd_bijective** : Function.Bijective (toFinAd F E)

theorem **finAdEquiv_apply** {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (h : w.under (𝒪 F) = v) (y : Af F ⊗[F] E) : finAdEquiv F E y w = toCompl h (Algebra.TensorProduct.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F v) (AlgHom.id F E) y)

@[simp] theorem **val_mul** : val (χ * ψ) w = val χ w * val ψ w

@[simp] theorem **val_inv** : val χ⁻¹ w = (val χ w)⁻¹

@[simp] theorem **val_one** : val (1 : (AF F ⊗[F] E)ˣ →* Circle) w = 1

theorem **norm_locChar** (x : (w.adicCompletion E)ˣ) : ‖(locChar χ w x : ℂ)‖ = 1

theorem **norm_val** : ‖(val χ w : ℂ)‖ = 1

theorem **UnrAt.mul** (h : UnrAt χ w) (h' : UnrAt ψ w) : UnrAt (χ * ψ) w

theorem **UnrAt.inv** (h : UnrAt χ w) : UnrAt χ⁻¹ w

theorem **UnrAt.locChar_eq_val** (h : UnrAt χ w) {x : (w.adicCompletion E)ˣ} (hx : IsUnifAt w x) : locChar χ w x = val χ w

theorem **charFam_of_unrAt** (h : UnrAt χ w) : charFam χ w = {val χ w}

theorem **bounded_charFam** (T : Set (HeightOneSpectrum (𝒪 E))) (χ : (AF F ⊗[F] E)ˣ →* Circle) : Bounded T (charFam χ) 1 0

theorem **converges_charFam** (T : Set (HeightOneSpectrum (𝒪 E))) (χ : (AF F ⊗[F] E)ˣ →* Circle) {s : ℂ} (hs : 1 < s.re) : Converges T (charFam χ) s

theorem **Bounded.converges_twist_val** {T : Set (HeightOneSpectrum (𝒪 E))} {c : Fam E} {r : ℕ} {κ : ℝ} (h : Bounded T c r κ) (μ : (AF F ⊗[F] E)ˣ →* Circle) {s : ℂ} (hs : κ + 1 < s.re) : Converges T (LFun.twist c (val μ)⁻¹) s

theorem **isHeckeChar_iff** (χ : (AE D)ˣ →* Circle) : IsHeckeChar D χ ↔ IsHecke D.host.F D.host.E χ

theorem **resEta_iff** (χ : (AE D)ˣ →* Circle) (k : ℕ) : Formalism.L2.Theta.ResEta D χ k ↔ ResEta D.host.F D.host.E χ k

Development.AbelianVarietyTheory.Definitions.Foundations.NotIsTrDataGaussianIntOne

theorem **qq_pow** (T : Tower p Ω) (n : ℕ) : T.qq n ^ (p ^ n) = algebraMap (O0 p) Ω (qO p)

theorem **isDerOn_derivation** [Module A M] (D : Derivation A S M) (B : Subalgebra A S) : IsDerOn B (D : S → M)

theorem **isDerOn_zero** (B : Subalgebra A S) : IsDerOn B (fun _ : S => (0 : M))

theorem **isDerOn_mono** {B C : Subalgebra A S} (h : B ≤ C) {D : S → M} (hD : IsDerOn C D) : IsDerOn B D

theorem **isDerOn_algebraMap_mul** {B : Subalgebra A S} {D : S → M} (hD : IsDerOn B D) (a : A) {x : S} (hx : x ∈ B) : D (algebraMap A S a * x) = algebraMap A S a • D x

theorem **isTrData_self** (B : Subalgebra A S) : IsTrData B B id (fun _ : Fin 1 => (1 : S)) (fun _ => 1) 1 where

theorem **isTrData_symm** {B C : Subalgebra A S} {tr : S → S} {n : ℕ} {a b : Fin n → S} {e : S} (h : IsTrData B C tr a b e) : IsTrData B C tr b a e

theorem **isDerOn_derivative** : IsDerOn (⊤ : Subalgebra ℚ (Polynomial ℚ)) (Polynomial.derivative : Polynomial ℚ → Polynomial ℚ) ∧ Polynomial.derivative (Polynomial.X : Polynomial ℚ) = 1

theorem **not_isDerOn_id** : ¬ IsDerOn (⊤ : Subalgebra ℚ (Polynomial ℚ)) (id : Polynomial ℚ → Polynomial ℚ)

theorem **isTrData_prod** : IsTrData (⊥ : Subalgebra ℤ (ℤ × ℤ)) ⊤ (fun x => (x.1 + x.2, x.1 + x.2)) ![((1 : ℤ), (0 : ℤ)), (0, 1)] ![((1 : ℤ), (0 : ℤ)), (0, 1)] 1 where

theorem **isTrData_gaussianInt** : IsTrData (⊥ : Subalgebra ℤ GaussianInt) ⊤ (fun x => x + star x) ![1, ⟨0, 1⟩] ![1, ⟨0, -1⟩] 2 where

theorem **bot_ne_top_gaussianInt** : (⊥ : Subalgebra ℤ GaussianInt) ≠ ⊤

theorem **not_isTrData_gaussianInt_one** {n : ℕ} (a b : Fin n → GaussianInt) : ¬ IsTrData (⊥ : Subalgebra ℤ GaussianInt) ⊤ (fun x => x + star x) a b 1

theorem **coe_zS** (T : Tower p Ω) (n : ℕ) : (zS T n : Ω) = T.ζ n

theorem **coe_qS** (T : Tower p Ω) (n : ℕ) : (qS T n : Ω) = T.qq n

theorem **zS_succ_pow** (T : Tower p Ω) (n : ℕ) : zS T (n + 1) ^ p = zS T n

theorem **qS_succ_pow** (T : Tower p Ω) (n : ℕ) : qS T (n + 1) ^ p = qS T n

theorem **zS_mem** (T : Tower p Ω) (n : ℕ) : zS T n ∈ levelAlg T n

theorem **qS_mem** (T : Tower p Ω) (n : ℕ) : qS T n ∈ levelAlg T n

theorem **levelAlg_mono** (T : Tower p Ω) : Monotone (levelAlg T)

theorem **towerAlg_eq_iSup** (T : Tower p Ω) : towerAlg T = ⨆ n, levelAlg T n

theorem **frac_mk_eq_zero_iff** (x : Ω) : (Submodule.Quotient.mk x : Frac p Ω) = 0 ↔ IsInt p x

theorem **frac_mk_inv_p_ne_zero** (h : IsClo p Ω) : (Submodule.Quotient.mk ((p : Ω)⁻¹) : Frac p Ω) ≠ 0

Development.AbelianVarietyTheory.Definitions.Foundations.OfInfProjInfMulOfFinProjFin

theorem **matHom_apply** (g : HA R σ J) : matHom R σ J g = HA.mat R σ J g

theorem **matHom_injective** : Function.Injective (matHom R σ J)

theorem **isEmbedding_matHom** : Topology.IsEmbedding (matHom R σ J)

theorem **lcsc_GFin** : LocallyCompactSpace (GFin σ J) ∧ SecondCountableTopology (GFin σ J) ∧ T2Space (GFin σ J)

theorem **lcsc_GInf** : LocallyCompactSpace (GInf σ J) ∧ SecondCountableTopology (GInf σ J) ∧ T2Space (GInf σ J)

theorem **lcsc_GLoc** (v : HeightOneSpectrum (𝒪 F)) : LocallyCompactSpace (GLoc σ J v) ∧ SecondCountableTopology (GLoc σ J v) ∧ T2Space (GLoc σ J v)

theorem **lcsc_GArch** (w : InfinitePlace F) : LocallyCompactSpace (GArch σ J w) ∧ SecondCountableTopology (GArch σ J w) ∧ T2Space (GArch σ J w)

theorem **mapH_mapH** {R R' R'' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] [CommRing R''] [Algebra F R''] (φ : R →ₐ[F] R') (ψ : R' →ₐ[F] R'') (g : HA R σ J) : mapH σ J ψ (mapH σ J φ g) = mapH σ J (ψ.comp φ) g

theorem **af_ext** {a b : Af F} (h : ∀ u, Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F u a = Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F u b) : a = b

theorem **evalV_singleV** (v u : HeightOneSpectrum (𝒪 F)) (y : Fv F v) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F u (singleV F v y) = (Pi.single v y : ∀ t : HeightOneSpectrum (𝒪 F), Fv F t) u

theorem **singleFin_mul** (v : HeightOneSpectrum (𝒪 F)) (a b : Fv F v) : singleFin v (a * b) = singleFin v a * singleFin v b

theorem **evalV_singleFin_self** (v : HeightOneSpectrum (𝒪 F)) (a : Fv F v) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F v (singleFin v a) = a

theorem **evalV_singleFin_ne** {v u : HeightOneSpectrum (𝒪 F)} (h : u ≠ v) (a : Fv F v) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.evalV F u (singleFin v a) = 0

theorem **continuous_singleFin** (v : HeightOneSpectrum (𝒪 F)) : Continuous (singleFin v)

theorem **matHom_proj** (v : HeightOneSpectrum (𝒪 F)) (g : GAd σ J) : matHom _ σ J (proj σ J v g) = bc (compFin F E v) (matHom _ σ J g)

theorem **matHom_projArch** (w : InfinitePlace F) (g : GAd σ J) : matHom _ σ J (projArch σ J w g) = bc (compInf F E w) (matHom _ σ J g)

@[simp] theorem **projFinLoc_projFin** (v : HeightOneSpectrum (𝒪 F)) (g : GAd σ J) : projFinLoc σ J v (projFin σ J g) = proj σ J v g

@[simp] theorem **projInfArch_projInf** (w : InfinitePlace F) (g : GAd σ J) : projInfArch σ J w (projInf σ J g) = projArch σ J w g

@[simp] theorem **projFin_ofFin** (y : GFin σ J) : projFin σ J (ofFin σ J y) = y

@[simp] theorem **projInf_ofFin** (y : GFin σ J) : projInf σ J (ofFin σ J y) = 1

@[simp] theorem **projInf_ofInf** (y : GInf σ J) : projInf σ J (ofInf σ J y) = y

@[simp] theorem **projFin_ofInf** (y : GInf σ J) : projFin σ J (ofInf σ J y) = 1

@[simp] theorem **projFinLoc_embLocFin_self** (v : HeightOneSpectrum (𝒪 F)) (x : GLoc σ J v) : projFinLoc σ J v (embLocFin σ J v x) = x

theorem **projFinLoc_embLocFin_ne** {v u : HeightOneSpectrum (𝒪 F)} (h : u ≠ v) (x : GLoc σ J v) : projFinLoc σ J u (embLocFin σ J v x) = 1

@[simp] theorem **projInfArch_embArchInf_self** (w : InfinitePlace F) (x : GArch σ J w) : projInfArch σ J w (embArchInf σ J w x) = x

theorem **projInfArch_embArchInf_ne** {w u : InfinitePlace F} (h : u ≠ w) (x : GArch σ J w) : projInfArch σ J u (embArchInf σ J w x) = 1

@[simp] theorem **proj_ofFin** (v : HeightOneSpectrum (𝒪 F)) (y : GFin σ J) : proj σ J v (ofFin σ J y) = projFinLoc σ J v y

@[simp] theorem **proj_ofInf** (v : HeightOneSpectrum (𝒪 F)) (y : GInf σ J) : proj σ J v (ofInf σ J y) = 1

@[simp] theorem **projArch_ofInf** (w : InfinitePlace F) (y : GInf σ J) : projArch σ J w (ofInf σ J y) = projInfArch σ J w y

@[simp] theorem **projArch_ofFin** (w : InfinitePlace F) (y : GFin σ J) : projArch σ J w (ofFin σ J y) = 1

@[simp] theorem **proj_emb_self** (v : HeightOneSpectrum (𝒪 F)) (x : GLoc σ J v) : proj σ J v (emb σ J v x) = x

theorem **proj_emb_ne** {v u : HeightOneSpectrum (𝒪 F)} (h : u ≠ v) (x : GLoc σ J v) : proj σ J u (emb σ J v x) = 1

@[simp] theorem **projArch_emb** (w : InfinitePlace F) (v : HeightOneSpectrum (𝒪 F)) (x : GLoc σ J v) : projArch σ J w (emb σ J v x) = 1

@[simp] theorem **projInf_emb** (v : HeightOneSpectrum (𝒪 F)) (x : GLoc σ J v) : projInf σ J (emb σ J v x) = 1

@[simp] theorem **projArch_embArch_self** (w : InfinitePlace F) (x : GArch σ J w) : projArch σ J w (embArch σ J w x) = x

theorem **projArch_embArch_ne** {w u : InfinitePlace F} (h : u ≠ w) (x : GArch σ J w) : projArch σ J u (embArch σ J w x) = 1

@[simp] theorem **proj_embArch** (v : HeightOneSpectrum (𝒪 F)) (w : InfinitePlace F) (x : GArch σ J w) : proj σ J v (embArch σ J w x) = 1

@[simp] theorem **projFin_embArch** (w : InfinitePlace F) (x : GArch σ J w) : projFin σ J (embArch σ J w x) = 1

theorem **emb_injective** (v : HeightOneSpectrum (𝒪 F)) : Function.Injective (emb σ J v)

@[simp] theorem **proj_ratEmb** (v : HeightOneSpectrum (𝒪 F)) (γ : GRat σ J) : proj σ J v (ratEmb σ J γ) = ratLoc σ J v γ

@[simp] theorem **projArch_ratEmb** (w : InfinitePlace F) (γ : GRat σ J) : projArch σ J w (ratEmb σ J γ) = ratArch σ J w γ

theorem **ext_inf_fin** {g h : GAd σ J} (h1 : projInf σ J g = projInf σ J h) (h2 : projFin σ J g = projFin σ J h) : g = h

theorem **GFin_ext** {g h : GFin σ J} (e : ∀ v, projFinLoc σ J v g = projFinLoc σ J v h) : g = h

theorem **GInf_ext** {g h : GInf σ J} (e : ∀ w, projInfArch σ J w g = projInfArch σ J w h) : g = h

theorem **GAd_ext** {g h : GAd σ J} (h1 : ∀ w, projArch σ J w g = projArch σ J w h) (h2 : ∀ v, proj σ J v g = proj σ J v h) : g = h

theorem **ofInf_projInf_mul_ofFin_projFin** (g : GAd σ J) : ofInf σ J (projInf σ J g) * ofFin σ J (projFin σ J g) = g

theorem **commute_ofInf_ofFin** (x : GInf σ J) (y : GFin σ J) : Commute (ofInf σ J x) (ofFin σ J y)

theorem **commute_emb** {v : HeightOneSpectrum (𝒪 F)} (x : GLoc σ J v) {g : GAd σ J} (hg : proj σ J v g = 1) : Commute (emb σ J v x) g

theorem **commute_embArch** {w : InfinitePlace F} (x : GArch σ J w) {g : GAd σ J} (hg : projArch σ J w g = 1) : Commute (embArch σ J w x) g

theorem **ratEmb_injective** : Function.Injective (ratEmb σ J)

theorem **continuous_projFin** : Continuous (projFin σ J)

theorem **continuous_projInf** : Continuous (projInf σ J)

theorem **continuous_projFinLoc** (v : HeightOneSpectrum (𝒪 F)) : Continuous (projFinLoc σ J v)

theorem **continuous_projInfArch** (w : InfinitePlace F) : Continuous (projInfArch σ J w)

theorem **continuous_proj** (v : HeightOneSpectrum (𝒪 F)) : Continuous (proj σ J v)

theorem **continuous_projArch** (w : InfinitePlace F) : Continuous (projArch σ J w)

theorem **continuous_ofFin** : Continuous (ofFin σ J)

theorem **continuous_ofInf** : Continuous (ofInf σ J)

theorem **continuous_embLocFin** (v : HeightOneSpectrum (𝒪 F)) : Continuous (embLocFin σ J v)

theorem **continuous_embArchInf** (w : InfinitePlace F) : Continuous (embArchInf σ J w)

theorem **continuous_emb** (v : HeightOneSpectrum (𝒪 F)) : Continuous (emb σ J v)

theorem **continuous_embArch** (w : InfinitePlace F) : Continuous (embArch σ J w)

theorem **isClosedEmbedding_emb** (v : HeightOneSpectrum (𝒪 F)) : Topology.IsClosedEmbedding (emb σ J v)

theorem **mem_finPts** {g : GAd σ J} : g ∈ finPts σ J ↔ projInf σ J g = 1

theorem **mem_infPts** {g : GAd σ J} : g ∈ infPts σ J ↔ projFin σ J g = 1

theorem **mem_finPts_iff_forall** {g : GAd σ J} : g ∈ finPts σ J ↔ ∀ w, projArch σ J w g = 1

theorem **mem_infPts_iff_forall** {g : GAd σ J} : g ∈ infPts σ J ↔ ∀ v, proj σ J v g = 1

theorem **mem_awayPts** {S : Finset (HeightOneSpectrum (𝒪 F))} {g : GAd σ J} : g ∈ awayPts σ J S ↔ ∀ v ∈ S, proj σ J v g = 1

theorem **mem_finAwayPts** {S : Finset (HeightOneSpectrum (𝒪 F))} {g : GAd σ J} : g ∈ finAwayPts σ J S ↔ g ∈ finPts σ J ∧ ∀ v ∈ S, proj σ J v g = 1

theorem **mem_atPts** {S : Finset (HeightOneSpectrum (𝒪 F))} {g : GAd σ J} : g ∈ atPts σ J S ↔ g ∈ finPts σ J ∧ ∀ v ∉ S, proj σ J v g = 1

theorem **range_ratEmb** : (ratEmb σ J).range = ratPts σ J

theorem **range_ofFin** : (ofFin σ J).range = finPts σ J

theorem **range_ofInf** : (ofInf σ J).range = infPts σ J

@[simp] theorem **awayPts_empty** : awayPts σ J ∅ = ⊤

@[simp] theorem **projAt_apply** (S : Finset (HeightOneSpectrum (𝒪 F))) (g : GAd σ J) (v : S) : projAt σ J S g v = proj σ J v g

theorem **embAt_mulSingle** (S : Finset (HeightOneSpectrum (𝒪 F))) [DecidableEq S] (v : S) (y : GLoc σ J v) : embAt σ J S (Pi.mulSingle v y) = emb σ J v y

@[simp] theorem **projAt_embAt** (S : Finset (HeightOneSpectrum (𝒪 F))) (x : ∀ v : S, GLoc σ J v) : projAt σ J S (embAt σ J S x) = x

@[simp] theorem **proj_embAt** (S : Finset (HeightOneSpectrum (𝒪 F))) (x : ∀ v : S, GLoc σ J v) (v : S) : proj σ J v (embAt σ J S x) = x v

theorem **embAt_mem** (S : Finset (HeightOneSpectrum (𝒪 F))) (x : ∀ v : S, GLoc σ J v) : embAt σ J S x ∈ atPts σ J S

theorem **range_embAt** (S : Finset (HeightOneSpectrum (𝒪 F))) : (embAt σ J S).range = atPts σ J S

theorem **range_emb** (v : HeightOneSpectrum (𝒪 F)) : (emb σ J v).range = atPts σ J {v}

theorem **commute_atPts_awayPts** {S : Finset (HeightOneSpectrum (𝒪 F))} {x y : GAd σ J} (hx : x ∈ atPts σ J S) (hy : y ∈ awayPts σ J S) : Commute x y

theorem **embAt_mul_partAway** (S : Finset (HeightOneSpectrum (𝒪 F))) (g : GAd σ J) : embAt σ J S (projAt σ J S g) * partAway σ J S g = g

theorem **atAwayEquiv_symm_apply** (S : Finset (HeightOneSpectrum (𝒪 F))) (p : (∀ v : S, GLoc σ J v) × awayPts σ J S) : (atAwayEquiv σ J S).symm p = embAt σ J S p.1 * p.2

theorem **atAwayEquiv_apply** (S : Finset (HeightOneSpectrum (𝒪 F))) (g : GAd σ J) : atAwayEquiv σ J S g = (projAt σ J S g, partAway σ J S g)

theorem **embArchPi_mulSingle** [DecidableEq (InfinitePlace F)] (w : InfinitePlace F) (y : GArch σ J w) : embArchPi σ J (Pi.mulSingle w y) = embArchInf σ J w y

@[simp] theorem **projInfArch_embArchPi** (x : ∀ w, GArch σ J w) (w : InfinitePlace F) : projInfArch σ J w (embArchPi σ J x) = x w

theorem **continuous_noncommProd** {ι X M : Type} [TopologicalSpace X] [Monoid M] [TopologicalSpace M] [ContinuousMul M] (f : ι → X → M) (hf : ∀ i, Continuous (f i)) (comm : ∀ x, Pairwise fun i j => Commute (f i x) (f j x)) (s : Finset ι) : Continuous fun x =>

s.noncommProd (fun i => f i x) fun _ _ _ _ h => comm x h

theorem **continuous_projAt** (S : Finset (HeightOneSpectrum (𝒪 F))) : Continuous (projAt σ J S)

theorem **continuous_embAt** (S : Finset (HeightOneSpectrum (𝒪 F))) : Continuous (embAt σ J S)

theorem **continuous_partAway** (S : Finset (HeightOneSpectrum (𝒪 F))) : Continuous (partAway σ J S)

theorem **continuous_embArchPi** : Continuous (embArchPi σ J)

theorem **isClosed_finPts** : IsClosed (finPts σ J : Set (GAd σ J))

theorem **isClosed_infPts** : IsClosed (infPts σ J : Set (GAd σ J))

theorem **isClosed_awayPts** (S : Finset (HeightOneSpectrum (𝒪 F))) : IsClosed (awayPts σ J S : Set (GAd σ J))

theorem **infEmb_sigR** (h : IsCMExt F E σ) (τ : EmbOver E w) (y : w.Completion ⊗[F] E) : starRingEnd ℂ (infEmb τ y) = infEmb τ (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ y)

theorem **infEmb_bijective** (h : IsCMExt F E σ) (τ : EmbOver E w) : Function.Bijective (infEmb τ)

theorem **archEquiv_val** (h : IsCMExt F E σ) (τ : EmbOver E w) (g : GArch σ J w) : (archEquiv J h τ g).1.1 = (matHom _ σ J g).1.map (infEmb τ)

theorem **isInfEmb_infProj** (τ : EmbOver E w) : Formalism.MainNotions.StandardLattice.IsInfEmb τ.1 (infProj τ)

theorem **infMat_infProj** (h : IsCMExt F E σ) (τ : EmbOver E w) (g : GAd σ J) : Formalism.MainNotions.StandardLattice.infMat σ J (infProj τ) g = (archEquiv J h τ (projArch σ J w g)).1.1

Development.AbelianVarietyTheory.Definitions.Foundations.OpLDOneSubMemTriv

theorem **lift_congr** {A : Type} [Semiring A] [Algebra R A] {f g : Letter → A} (h : ∀ T, S T → f T = g T) {P : FreeAlgebra R Letter} (hP : P ∈ LetAlg R S) : FreeAlgebra.lift R f P = FreeAlgebra.lift R g P

theorem **exists_isDiam** {N l d : ℕ} (hN : N ≠ 0) (hl : l ≠ 0) (hd : Nat.Coprime d N) : ∃ γ, IsDiam N l d γ

theorem **isDiam_sgD** {N l d : ℕ} (hN : N ≠ 0) (hl : l ≠ 0) (hd : Nat.Coprime d N) : IsDiam N l d (sgD N l d)

theorem **isDiam_sgD_of_adm** {N l q : ℕ} (h : admitted N l (Letter.T q)) : IsDiam N l q (sgD N l q)

theorem **slP_X** (m : Matrix (Fin 2) (Fin 2) ℤ) (i : Fin 2) : slP (R

theorem **polySlash_one** (p : MvPolynomial (Fin 2) R) : slP 1 p = p

theorem **polySlash_polySlash** (m m' : Matrix (Fin 2) (Fin 2) ℤ) (p : MvPolynomial (Fin 2) R) : slP m' (slP m p) = slP (m * m') p

theorem **polySlash_isHomogeneous** (m : Matrix (Fin 2) (Fin 2) ℤ) {p : MvPolynomial (Fin 2) R} {w : ℕ} (hp : p.IsHomogeneous w) : (slP m p).IsHomogeneous w

theorem **eval_slP** (m : Matrix (Fin 2) (Fin 2) ℤ) (u : Fin 2 → R) (p : MvPolynomial (Fin 2) R) : MvPolynomial.eval u (slP m p) = MvPolynomial.eval (m.map (Int.castRingHom R) *ᵥ u) p

theorem **toGQ_sl** (γ : SL(2, ℤ)) : toGQ (γ : Matrix (Fin 2) (Fin 2) ℤ) = ofSL γ

@[simp] theorem **mem_potSub** {F : ℂ → MvPolynomial (Fin 2) R} : F ∈ potSub R w Γ ↔ F ∈ Pot R w Γ

@[simp] theorem **mem_trivSub** {F : ℂ → MvPolynomial (Fin 2) R} : F ∈ trivSub R w Γ ↔ F ∈ Triv R w Γ

Development.AbelianVarietyTheory.Definitions.Foundations.PetEqRelIndexMulPetProd

theorem **le_pmGrp** (Γ : Subgroup SL(2, ℤ)) : Γ ≤ pmGrp Γ

theorem **neg_one_mem_pmGrp** (Γ : Subgroup SL(2, ℤ)) : -1 ∈ pmGrp Γ

theorem **pmGrp_eq_self** {Γ : Subgroup SL(2, ℤ)} (h : -1 ∈ Γ) : pmGrp Γ = Γ

theorem **relIndex_pmGrp_of_notMem** {Γ : Subgroup SL(2, ℤ)} (h : -1 ∉ Γ) : Γ.relIndex (pmGrp Γ) = 2

theorem **neg_one_notMem_lev** {N : ℕ} (l : ℕ) (hN : 3 ≤ N) : -1 ∉ lev N l

theorem **petProd_smul_left** (Γ : Subgroup SL(2, ℤ)) [Γ.FiniteIndex] (k : ℤ) (c : ℂ) (f g : ℍ → ℂ) : petProd Γ k (c • f) g = c * petProd Γ k f g

theorem **petProd_smul_right** (Γ : Subgroup SL(2, ℤ)) [Γ.FiniteIndex] (k : ℤ) (c : ℂ) (f g : ℍ → ℂ) : petProd Γ k f (c • g) = (starRingEnd ℂ) c * petProd Γ k f g

theorem **petProd_level_one** (Γ : Subgroup SL(2, ℤ)) [Γ.FiniteIndex] (k : ℤ) {f g : ℍ → ℂ} (hf : ∀ γ : SL(2, ℤ), f ∣[k] γ = f) (hg : ∀ γ : SL(2, ℤ), g ∣[k] γ = g) : petProd Γ k f g = ((pmGrp Γ).index : ℂ) * ∫ z in ModularGroup.fd, petersson k g f z

Development.AbelianVarietyTheory.Definitions.Foundations.PetIPSmulRight

theorem **mul_eq_cmul** (f g : ℕ → K) : mul f g = GlobalTheory.KummerEisensteinGalois.cmul f g

Development.AbelianVarietyTheory.Definitions.Foundations.PowLtFirstRadius

theorem **_root_.GlobalTheory.Cyc.Model.isom_gal** (g : _root_.Formalism.BasicNotions.Loc.GQ l) : Isometry (g : _root_.Formalism.BasicNotions.Loc.Om l →+* _root_.Formalism.BasicNotions.Loc.Om l)

theorem **extHom_coe** (g : GQ l) (x : Om l) : extHom g (x : PadicComplex l) = ((g x : Om l) : PadicComplex l)

theorem **extHom_cont** (g : GQ l) : Continuous (extHom g)

private theorem **continuous_cyc_extHom** (g : _root_.Formalism.BasicNotions.Loc.GQ l) : Continuous (_root_.GlobalTheory.Cyc.Model.extHom g)

private theorem **cyc_extHom_coe** (g : _root_.Formalism.BasicNotions.Loc.GQ l) (x : _root_.Formalism.BasicNotions.Loc.Om l) : _root_.GlobalTheory.Cyc.Model.extHom g (x : PadicComplex l) = ((g x : _root_.Formalism.BasicNotions.Loc.Om l) : PadicComplex l)

theorem **_root_.GlobalTheory.Cyc.Model.extHom_mul** (g h : _root_.Formalism.BasicNotions.Loc.GQ l) (y : PadicComplex l) : _root_.GlobalTheory.Cyc.Model.extHom (g * h) y = _root_.GlobalTheory.Cyc.Model.extHom g (_root_.GlobalTheory.Cyc.Model.extHom h y)

theorem **_root_.GlobalTheory.Cyc.Model.extHom_one** (y : PadicComplex l) : _root_.GlobalTheory.Cyc.Model.extHom (1 : _root_.Formalism.BasicNotions.Loc.GQ l) y = y

private theorem **norm_cyc_extHom** (g : _root_.Formalism.BasicNotions.Loc.GQ l) (y : PadicComplex l) : ‖_root_.GlobalTheory.Cyc.Model.extHom g y‖ = ‖y‖

theorem **_root_.GlobalTheory.Cyc.Model.isActC_actC** : _root_.GlobalTheory.Cyc.IsActC (_root_.GlobalTheory.Cyc.Model.actC (l

theorem **chi_val** (g : GQ l) : (chi g : ℚ_[l]) = ((chiZ g : ℤ_[l]) : ℚ_[l])

theorem **isCyc_chi** : IsCyc (chi (l

theorem **chi_cont** : Continuous fun g : GQ l => (chi g : ℚ_[l])

theorem **_root_.GlobalTheory.Cyc.Model.cyc_irred** (n : ℕ) : Irreducible (cyclotomic (l ^ (n + 1)) ℚ_[l])

theorem **norm_sub_of_pow_eq** {ζ : Om l} {m a b : ℕ} (hζ : IsPrimitiveRoot ζ (l ^ m)) (h : ζ ^ a = ζ ^ b) : ‖(a : ℚ_[l]) − (b : ℚ_[l])‖ ≤ (l : ℝ) ^ (−(m : ℤ))

theorem **exists_isGen** {χ : GQ l →* ℚ_[l]ˣ} (hχ : IsCyc χ) {j : ℕ} (hj : 1 ≤ j) : ∃ g, IsGen χ (base l) j g

theorem **chi_Hinf** {χ : GQ l →* ℚ_[l]ˣ} (hχ : IsCyc χ) (K : FinExt l) {σ : GQ l} (hσ : σ ∈ Hinf K) : (χ σ : ℚ_[l]) = 1

theorem **pow_lt_first_radius** {m : ℕ} (h : Lev l m) : (l : ℝ) ^ (−(m : ℤ)) < rho1 l

theorem **chiC_one** (σ : GQ l) : chiC chi 1 σ = (((chi σ : ℚ_[l]) : Om l) : PadicComplex l)

theorem **exists_frame** {m : ℕ} (h : Lev l m) : ∃ g, Frame chi (base l) m (Utw (l

theorem **htBasis_twist** (G : Subgroup (GQ l)) : HTBasis actC chi G (Utw (l

theorem **zeta_mem_Xj** (K : FinExt l) (j : ℕ) {ζ : PadicComplex l} (h : IsPrimitiveRoot ζ (l ^ (j + 1))) : ζ ∈ Xj K j

theorem **coc_twist** (G : Subgroup (GQ l)) : CobOn G (twAct actC chi 1 (fun _ => (1 : Matrix (Fin 1) (Fin 1) (PadicComplex l)))) (ball l 1) (fun σ _ => chiC chi 1 σ − 1) ∧ CocOn G (twAct actC chi 1 (fun _ => (1 : Matrix (Fin 1) (Fin 1) (PadicComplex l)))) (ball l 1) (fun σ _ => chiC chi 1 σ − 1)

theorem **isActC_unique** {a b : GQ l →* (PadicComplex l ≃+* PadicComplex l)} (ha : IsActC a) (hb : IsActC b) : a = b

Development.AbelianVarietyTheory.Definitions.Foundations.ReferenceTupleNeBaseTuple

theorem **emb_mul_indep** {L I : Type} [Field L] [CharZero L] [Fintype I] (σ : I → L →+* ℂ) (hσ : Function.Injective σ) (m : I → ℤ) (h : ∀ k : L, k ≠ 0 → ∏ i, σ i k ^ m i = 1) : m = 0

theorem **unit_sq** (z : ℂ) (hz : z ≠ 0) : (z / ((‖z‖ : ℝ) : ℂ)) ^ 2 = z / starRingEnd ℂ z

theorem **hasInf_unique** {L I : Type} [Field L] [NumberField L] [Fintype I] (φ : I → L →+* ℂ) (h1 : Function.Injective φ) (h2 : ∀ i j, φ i ≠ (starRingEnd ℂ).comp (φ j)) {χ : HChar L} {n n' : I → ℤ} (h : χ.HasInf φ n) (h' : χ.HasInf φ n') : n = n'

theorem **inf_eq_one_of_order** {L : Type} [Field L] [NumberField L] [IsTotallyComplex L] {χ : HChar L} {n : ℕ} (hn : 0 < n) (h : χ.OrderDvd n) (k : Lˣ) : χ.inf k = 1

theorem **embTau_inj** (D : D7) : Function.Injective (embTau D)

theorem **embTau_ne_conj** (D : D7) (i j : D.what₂.Tau) : embTau D i ≠ (starRingEnd ℂ).comp (embTau D j)

theorem **hasInfQ_unique** (D : D7) {χ : HChar D.host.E} {a a' : D.what₂.Tau → ℚ} (h : χ.HasInfQ (embTau D) a) (h' : χ.HasInfQ (embTau D) a') : a = a'

theorem **hasInfQ_tau** (D : D7) {χ : HChar D.host.E} {a : D.R.Psi → ℚ} (h : χ.HasInfQ (embPsi D.host) a) : χ.HasInfQ (embTau D) (fun τ => a (D.eTau τ))

theorem **hasInfQ_mul** {L I : Type} [Field L] [NumberField L] [Fintype I] {φ : I → L →+* ℂ} {χ χ' : HChar L} {a a' : I → ℚ} (h : χ.HasInfQ φ a) (h' : χ'.HasInfQ φ a') : (χ * χ').HasInfQ φ (a + a')

theorem **hasInfQ_inv** {L I : Type} [Field L] [NumberField L] [Fintype I] {φ : I → L →+* ℂ} {χ : HChar L} {a : I → ℚ} (h : χ.HasInfQ φ a) : (χ⁻¹).HasInfQ φ (−a)

theorem **tup_unitary** (Tp : TupTied D C nch) (μ : D.what₂.Tup) (j : Fin D.R.m) : (Tp.tup μ j).IsUnitary

theorem **tup_inf** (Tp : TupTied D C nch) (μ : D.what₂.Tup) (j : Fin D.R.m) : (Tp.tup μ j).HasInfQ (embTau D) (fun τ => −(D.hc τ).e j)

theorem **eta_inf** (C : CharFrame D) : C.eta.HasInfQ (embTau D) (fun _ => 1 / 2)

theorem **expo_spec** (c : halfInt D) : c.1.HasInfQ (embTau D) (expo c)

theorem **tupNat_succ** (Tp : TupTied D C nch) (μ : D.what₂.Tup) (j : Fin D.R.m) : tupNat Tp μ (j.1 + 1) = tupEl Tp μ j

theorem **expo_eq** (c : halfInt D) {a : D.what₂.Tau → ℚ} (h : c.1.HasInfQ (embTau D) a) : expo c = a

theorem **expo_tupNat** (Tp : TupTied D C nch) (μ : D.what₂.Tup) (j : Fin D.R.m) (τ : D.what₂.Tau) : expo (tupNat Tp μ (j.1 + 1)) τ = −(D.hc τ).e j

theorem **expo_etaEl** (C : CharFrame D) (τ : D.what₂.Tau) : expo (etaEl C) τ = 1 / 2

theorem **expo_mul** (χ χ' : halfInt D) (τ : D.what₂.Tau) : expo (χ * χ') τ = expo χ τ + expo χ' τ

theorem **charFrame_nonempty** (D : D7) (hη : EtaTExists D.host) (hμ : ∀ P W j, Mu0Exists (H

theorem **infDiag_fst** {L : Type} [Field L] [NumberField L] (k : Lˣ) (w : InfinitePlace L) : ((infDiag k : Ideles L) : AdeleRing (𝒪 L) L).1 w = algebraMap L w.Completion (k : L)

theorem **isFinIdele_finPart** {L : Type} [Field L] [NumberField L] (k : Lˣ) : IsFinIdele (finPart k)

theorem **val_finPart** {L I : Type} [Field L] [NumberField L] [Fintype I] {φ : I → L →+* ℂ} {χ : HChar L} {n : I → ℤ} (h : χ.HasInf φ (fun i => 2 * n i)) (k : Lˣ) : ((χ.1 (finPart k) : ℂˣ) : ℂ) = ∏ i, (φ i k / starRingEnd ℂ (φ i k)) ^ (−n i)

theorem **eDt_finPart** (T : DtTied D C) (k : D.host.Eˣ) : T.eDt (finPart k) = T.eDt (diagUnits D.host.E k)

theorem **chiD_rat** (T : DtTied D C) (k : D.host.Eˣ) : ((D.chiD (T.eDt (diagUnits D.host.E k)) : ℂˣ) : ℂ) = ∏ g : D.R.Psi, (embPsi D.host g k / starRingEnd ℂ (embPsi D.host g k))⁻¹

theorem **muTil_rat** (T : DtTied D C) (j : Fin D.R.m) (e : D.R.Psi → ℤ) (he : ∀ g, eExp D j g = e g) (k : D.host.Eˣ) : ((D.muTil (j.1 + 1) (T.eDt (diagUnits D.host.E k)) : ℂˣ) : ℂ) = ∏ g : D.R.Psi, (embPsi D.host g k / starRingEnd ℂ (embPsi D.host g k)) ^ e g

theorem **exists_place_over** {L : Type} [Field L] [NumberField L] {p : ℕ} (hg : p.Prime) : ∃ v : HeightOneSpectrum (𝒪 L), ((p : ℕ) : 𝒪 L) ∈ v.asIdeal

theorem **reference_tuple_ne_base_tuple** (Tp : TupTied D C nch) : D.what₂.mu0 ≠ D.what₂.muStar

theorem **muTw_of_untwisted** (Tp : TupTied D C nch) (w : D.what₂.places.Pl) (h : D.what₂.muAt w = D.what₂.mu0) : D.what₂.muTw w = if D.what₂.places.resChar w = D.host.pde then D.what₂.muStar' else D.what₂.muStar

Development.AbelianVarietyTheory.Definitions.Foundations.RootOfUnityEightQMemCoeffGaloisSubfield

theorem **card_Tau** {R : Ranks} (H : Host R) : Fintype.card (Tau R) = R.f

theorem **mod_eq_one_of_cond** (h : p % (H.QS * H.q * H.d) = 1) {m : ℕ} (hm : 1 < m) (hdv : m ∣ H.QS * H.q * H.d) : p % m = 1

theorem **four_dvd_QS** : 4 ∣ H.QS

theorem **cls_eq_splitK_of_cond** (h : p % (H.QS * H.q * H.d) = 1) : H.cls p = .splitK

theorem **auxiliary_frame_zero** {R : Ranks} {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] (S : Spaces R F E) : HasHyperbolicAuxiliaryFrame S 0

theorem **level_prime** : ∀ p ∈ B.host.levelPrimes, p.Prime

theorem **level_split** : ∀ p ∈ B.host.levelPrimes, B.host.cls p = .splitK

theorem **level_mod_four** : ∀ p ∈ B.host.levelPrimes, p % 4 = 1

theorem **g_cond** : ∀ k, B.host.g k % (B.host.QS * B.host.q * B.host.d) = 1

theorem **pde_cond** : B.host.pde % (B.host.QS * B.host.q * B.host.d) = 1 ∧ B.host.pde' % (B.host.QS * B.host.q * B.host.d) = 1

theorem **T_prime** : ∀ u, (B.T u).Prime

theorem **T_cond** : ∀ u, B.T u % (B.host.QS * B.host.q * B.host.d) = 1

theorem **auxiliary_frame_exists** : letI

theorem **V_posDef** : letI

theorem **X_posDef** : letI

theorem **conjF_eq** (x : B.host.E) : B.conjF x = B.host.conj x

theorem **zOf_norm** (u : B.Word) : letI

theorem **restRowsFiniteGroupCard_pos** : 0 < B.restRowsFiniteGroupCard

theorem **ramOrder_pos** : 0 < B.ramOrder

theorem **eight_mul_q_dvd_ramOrder** : 8 * B.host.q ∣ B.ramOrder

theorem **eight_mul_dvd_ramOrder** {l : ℕ} (hl : l ∈ B.host.small) : 8 * l ∣ B.ramOrder

theorem **OtoLL_injective** : Function.Injective B.OtoLL

theorem **O_int** (y : B.O) : IsIntegral ℤ (B.OtoLL y)

theorem **O_full** (x : B.LL) (hx : x ∈ B.Ks) (hi : IsIntegral ℤ x) : ∃ y : B.O, B.OtoLL y = x

theorem **O_contains** (x : 𝒪 B.host.E) : ∃ y : B.O, B.OtoLL y = B.embE (x : B.host.E)

theorem **hostField_le_fixedField** (x : B.host.E) : B.embE x ∈ B.LL0

theorem **embE_conj** (x : B.host.E) : B.embE (B.host.conj x) = star (B.embE x)

theorem **roots_of_unity_in_fixed_field** {N : ℕ} (hN : N ∣ B.ramOrder) : ∃ z ∈ B.LL0, B.embLL z = Complex.exp (2 * Real.pi * Complex.I / (N : ℕ))

theorem **OtoLL_intE** (x : 𝒪 B.host.E) : B.OtoLL (B.intE x) = B.embE (x : B.host.E)

theorem **isPrincipal_map_intE** (𝔞 : Ideal (𝒪 B.host.E)) : (𝔞.map B.intE).IsPrincipal

Development.AbelianVarietyTheory.Definitions.Foundations.RowRawRange

lemma **rowRaw_range** (M : Model F E) {n : ℕ} (ZZ : M.Mat n) (j : Fin n) : LinearMap.range ZZ.Z.mulVecLin ≤ LinearMap.ker (rowRaw M ZZ j)

Development.AbelianVarietyTheory.Definitions.Foundations.SamePartIffOfNeZero

theorem **_root_.GlobalTheory.BadShare.bb_iff** {g N l : ℕ} : _root_.GlobalTheory.BadShare.BB g N l ↔ l = g ∨ (l.Prime ∧ ∃ d : ℕ, 1 ≤ d ∧ d ≤ N ∧ g ^ d < l ^ (4 * padicValNat l (g ^ d - 1)))

theorem **bb_of_dvd** {g N l d : ℕ} (hl : l.Prime) (h1 : 1 ≤ d) (h2 : d ≤ N) (h0 : g ^ d - 1 ≠ 0) (h3 : l ∣ g ^ d - 1) (h4 : g ^ d < l ^ 4) : BB g N l

private theorem **pow_eq_pow_of_modEq_orderOf'** {G : Type*} [Monoid G] {x : G} {a b : ℕ} (h : a ≡ b [MOD orderOf x]) : x ^ a = x ^ b

theorem **_root_.GlobalTheory.BadShare.eq_of_pow_eq** {M a : ℕ} (hM : M.Coprime l) {x y : k} (hx : x ^ M = 1) (hy : y ^ M = 1) (hM0 : M ≠ 0) (h : x ^ (l ^ a) = y ^ (l ^ a)) : x = y

private theorem **primeToPart_pow_prime_pow_eq_of_factorization_le** {G : Type*} [Monoid G] {l : ℕ} (hl : l.Prime) {x : G} (hx : IsOfFinOrder x) {a : ℕ} (ha : (orderOf x).factorization l ≤ a) : _root_.Formalism.CyclotomicNorms.primeToPart l x ^ (l ^ a) = x ^ (l ^ a)

private theorem **primeToPart_pow_orderOf_div_prime_pow_eq_one** {G : Type*} [Monoid G] (l : ℕ) (x : G) : _root_.Formalism.CyclotomicNorms.primeToPart l x ^ (orderOf x / l ^ (orderOf x).factorization l) = 1

theorem **_root_.GlobalTheory.BadShare.samePart_iff** (hl : l.Prime) {x y : k} (hx : IsOfFinOrder x) (hy : IsOfFinOrder y) : _root_.GlobalTheory.BadShare.SamePart l x y ↔ ∃ a : ℕ, x ^ (l ^ a) = y ^ (l ^ a)

theorem **_root_.GlobalTheory.BadShare.root_ne_zero** {f : (ZMod g)[X]} (h0 : f.coeff 0 ≠ 0) {x : AlgebraicClosure (ZMod g)} (hx : x ∈ f.aroots (AlgebraicClosure (ZMod g))) : x ≠ 0

private theorem **isOfFinOrder_of_ne_zero_algebraicClosure_zmod** {g : ℕ} [Fact g.Prime] {x : AlgebraicClosure (ZMod g)} (hx : x ≠ 0) : IsOfFinOrder x

theorem **_root_.GlobalTheory.BadShare.samePart_iff_of_ne_zero** {l : ℕ} (hl : l.Prime) {x y : AlgebraicClosure (ZMod g)} (hx : x ≠ 0) (hy : y ≠ 0) : _root_.GlobalTheory.BadShare.SamePart l x y ↔ ∃ a : ℕ, x ^ (l ^ a) = y ^ (l ^ a)

theorem **_root_.GlobalTheory.BadShare.rootsAvoid_iff** {c : (ZMod g)ˣ} {f : (ZMod g)[X]} (h0 : f.coeff 0 ≠ 0) : _root_.GlobalTheory.BadShare.RootsAvoid g c f ↔ ∀ l : ℕ, l.Prime → l ≠ g → ∀ x ∈ f.aroots (AlgebraicClosure (ZMod g)), ∀ a : ℕ, x ^ (l ^ a) ≠ (algebraMap (ZMod g) (AlgebraicClosure (ZMod g)) (c : ZMod g)) ^ (l ^ a)

theorem **sqSet_eq_monicOf** (N : ℕ) : sqSet g N = (Finset.univ.image fun c : Fin N → ZMod g => Formalism.CyclotomicNorms.monicOf c).filter fun f => Squarefree f ∧ f.coeff 0 ≠ 0

theorem **_root_.GlobalTheory.BadShare.mem_sqSet** {N : ℕ} {f : (ZMod g)[X]} : f ∈ _root_.GlobalTheory.BadShare.sqSet g N ↔ f.Monic ∧ f.natDegree = N ∧ Squarefree f ∧ f.coeff 0 ≠ 0

theorem **_root_.GlobalTheory.BadShare.good_iff** {ι : Type} {m N : ℕ} {ch : ι → Fin m → (ZMod g)ˣ} {F : ι → (ZMod g)[X]} : _root_.GlobalTheory.BadShare.Good g N ch F ↔ (∀ v, F v ∈ _root_.GlobalTheory.BadShare.sqSet g N) ∧ (∀ v,

root.GlobalTheory.BadShare.RootsApart g N (F v)) ∧ (∀ v j, _root_.GlobalTheory.BadShare.RootsAvoid g (ch v j) (F v) ∧ _root_.GlobalTheory.BadShare.RootsAvoid g (ch v j)⁻¹ (F v)) ∧ (¬ ∃ a : ℕ, 1 ≤ a ∧ 2 * a ≤ N ∧ ∀ v, _root_.GlobalTheory.BadShare.HasDiv g a (F v)) ∧ ∃ v, _root_.GlobalTheory.BadShare.HasPrimeFactor g N (F v)

theorem **_root_.GlobalTheory.BadShare.mem_goodSet_iff_good** {ι : Type} [Fintype ι] {m N : ℕ} {ch : ι → Fin m → (ZMod g)ˣ} {F : ι → (ZMod g)[X]} : F ∈ _root_.GlobalTheory.BadShare.goodSet g N ch ↔ _root_.GlobalTheory.BadShare.Good g N ch F

theorem **_root_.GlobalTheory.BadShare.torusCard_irreducible** {f : (ZMod g)[X]} (hf : Irreducible f) : _root_.GlobalTheory.BadShare.torusCard g f = g ^ f.natDegree - 1

theorem **_root_.GlobalTheory.BadShare.torusCard_mul** {f h : (ZMod g)[X]} (hf : f ≠ 0) (hh : h ≠ 0) : _root_.GlobalTheory.BadShare.torusCard g (f * h) = _root_.GlobalTheory.BadShare.torusCard g f * _root_.GlobalTheory.BadShare.torusCard g h

theorem **_root_.GlobalTheory.BadShare.mass_nonneg** (N : ℕ) (A : (ZMod g)[X] → Prop) : 0 ≤ _root_.GlobalTheory.BadShare.mass g N A

theorem **_root_.GlobalTheory.BadShare.mass_mono** {N : ℕ} {A B : (ZMod g)[X] → Prop} (h : ∀ f ∈ _root_.GlobalTheory.BadShare.sqSet g N, A f → B f) : _root_.GlobalTheory.BadShare.mass g N A ≤ _root_.GlobalTheory.BadShare.mass g N B

theorem **exists_primitive** (d : ℕ) (hd : d ≠ 0) : ∃ P : (ZMod g)[X], P.Monic ∧ Irreducible P ∧ P.natDegree = d ∧ ∀ x ∈ P.aroots (AlgebraicClosure (ZMod g)), ∀ n : ℕ, x ^ n = 1 ↔ g ^ d - 1 ∣ n

theorem **not_two_primes** {l a n p : ℕ} (hl : l.Prime) (hp : p.Prime) (hp2 : p ≠ 2) (h2 : 2 ∣ n) (h3 : p ∣ n) : ¬ n ∣ l ^ a

Development.AbelianVarietyTheory.Definitions.Foundations.SecondCountableAndT2LocalField

theorem **padic_lt** (p : ℕ) [Fact p.Prime] (x y : ℚ_[p]) : valuation ℚ_[p] x < valuation ℚ_[p] y ↔ ‖x‖ < ‖y‖

theorem **padic_valTop** (p : ℕ) [Fact p.Prime] : IsValuativeTopology ℚ_[p]

theorem **padic_localField** (p : ℕ) [Fact p.Prime] : IsNonarchimedeanLocalField ℚ_[p]

theorem **secondCountable_and_t2_local_field** : SecondCountableTopology k ∧ T2Space k

theorem **nabs_pos** (x : kˣ) : 0 < nabs k x

theorem **nabs_mul** (x y : kˣ) : nabs k (x * y) = nabs k x * nabs k y

theorem **nabs_one** : nabs k 1 = 1

theorem **om_one** {a b : Type*} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (Φ : Matrix b a k → ℂ) : om μ (1 : GL a k) (1 : GL b k) Φ = Φ

theorem **om_mul** {a b : Type*} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g g' : GL a k) (h h' : GL b k) (Φ : Matrix b a k → ℂ) : om μ (g * g') (h * h') Φ = om μ g h (om μ g' h' Φ)

theorem **not_hasConductor_one** (c : kˣ) : ¬ HasConductor (1 : AddChar k Circle) c

theorem **ratCircle_eq_zero** (x : AddCircle (1 : ℚ)) (h : ratCircle x = 0) : x = 0

theorem **exists_hasConductor** : ∃ ψ : AddChar k Circle, HasConductor ψ 1

theorem **KOne_le_KO** (n : Type*) [Fintype n] [DecidableEq n] : KOne k n ≤ KO k n

theorem **IsUnram.isTame** {μ : kˣ →* ℂˣ} (h : IsUnram μ) : IsTame μ

theorem **isOpen_int** : IsOpen (𝒪[k] : Set k)

theorem **isCompact_int** : IsCompact (𝒪[k] : Set k)

theorem **exists_isUnitVol** : ∃ ν : Measure k, IsUnitVol ν

theorem **nabs_vol** (ν : Measure k) (h : IsUnitVol ν) (x : kˣ) : ν (x • (𝒪[k] : Set k)) = ENNReal.ofReal (nabs k x)

theorem **nabs_le_one** (x : kˣ) (hx : valuation k (x : k) ≤ 1) : nabs k x ≤ 1

theorem **volM_int** (ν : Measure k) (h : IsUnitVol ν) (m n : Type*) [Fintype m] [Fintype n] : volM ν m n {x | ∀ i j, x i j ∈ (𝒪[k] : Set k)} = 1

theorem **measurableEmbedding_val** (n : Type*) [Fintype n] [DecidableEq n] : MeasurableEmbedding (fun g : GL n k => g.val)

theorem **indicator_mem_SB** : (𝒪[k] : Set k).indicator (fun _ => (1 : ℂ)) ∈ SB k

theorem **noncompact** : NoncompactSpace k

theorem **not_isSelfDual_one** (ν : Measure k) : ¬ IsSelfDual (1 : AddChar k Circle) ν

theorem **om_mem_SB** {a b : Type*} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) {Φ : Matrix b a k → ℂ} (hΦ : Φ ∈ SB (Matrix b a k)) : om μ g h Φ ∈ SB (Matrix b a k)

Development.AbelianVarietyTheory.Definitions.Foundations.SiegelNeZeroIff

theorem **isOpen_Hs** : IsOpen Hs

theorem **neBot_Hs** : (Filter.codiscreteWithin Hs).NeBot

theorem **germ_eq_of_eqOn** {φ ψ : ℂ → ℂ} (h : ∀ z ∈ Hs, φ z = ψ z) : (φ : Mero) = (ψ : Mero)

theorem **sgC_coe** (c : ℕ) (x : Fin 2 → ℚ) (τ : ℍ) : sgC c x τ = sg c x τ

theorem **germ_sgC_ne_zero** {c : ℕ} {x : Fin 2 → ℚ} (h : admitted c x) : ((sgC c x : ℂ → ℂ) : Mero) ≠ 0

theorem **an_injective** : Function.Injective T.an

theorem **siegel_eq** {c : ℕ} {x : Fin 2 → ℚ} {f : T.K} (hf : T.an f = ((sgC c x : ℂ → ℂ) : Mero)) : siegel T c x = f

theorem **siegel_ne_zero_iff** {c : ℕ} {x : Fin 2 → ℚ} (h : admitted c x) : siegel T c x ≠ 0 ↔ T.an (siegel T c x) = ((sgC c x : ℂ → ℂ) : Mero)

theorem **sg_one** {x : Fin 2 → ℚ} (hx : ¬ (x 0 ∈ Set.range ((↑) : ℤ → ℚ) ∧ x 1 ∈ Set.range ((↑) : ℤ → ℚ))) (τ : ℍ) : sg 1 x τ = 1

theorem **siegel_one** {x : Fin 2 → ℚ} (hx : ¬ (x 0 ∈ Set.range ((↑) : ℤ → ℚ) ∧ x 1 ∈ Set.range ((↑) : ℤ → ℚ))) : siegel T 1 x = 1

theorem **mob_ofSL** (γ : SL(2, ℤ)) (τ : ℍ) : mob (ofSL γ) τ = ((γ • τ : ℍ) : ℂ)

theorem **sl_siegel** {c : ℕ} {x : Fin 2 → ℚ} (γ : SL(2, ℤ)) (hmem : T.an (siegel T c x) = ((sgC c x : ℂ → ℂ) : Mero)) (hγ : ∀ τ : ℍ, sg c x (γ • τ) = sg c (act x γ) τ) : T.sl (ofSL γ) (siegel T c x) = siegel T c (act x γ)

theorem **an_siegel** {c : ℕ} {x : Fin 2 → ℚ} (M : ℕ) (hM : 0 < M) (hd : DifferentiableOn ℂ (sgC c x) Hs) (hmod : ∀ (γ : SL(2, ℤ)) (τ : ℍ), sg c x (γ • τ) = sg c (act x γ) τ) (hlev : ∀ γ ∈ CongruenceSubgroup.Gamma M, ∀ τ : ℍ, sg c (act x γ) τ = sg c x τ) (hexp : ∀ γ : SL(2,

ℤ), ∃ (a : ℤ → Ω) (n0 : ℤ), (∀ n < n0, a n = 0) ∧ ∀ τ : ℍ, HasSum (fun n : ℤ => ε (a n) * Complex.exp (2 * Real.pi * Complex.I * (n : ℂ) * τ / M)) (sg c (act x γ) τ)) : T.an (siegel T c x) = ((sgC c x : ℂ → ℂ) : Mero)

theorem **siegel_zero_index** {c : ℕ} : siegel T c 0 = 0

Development.AbelianVarietyTheory.Definitions.Foundations.SimpleSumRowCol

theorem **inN_tri** {p : ℕ} {g : GL (n ⊕ Fin c) k} (hg : InN p g) : g.val.BlockTriangular (lev p)

theorem **mul_entry** {p : ℕ} {g : GL (n ⊕ Fin c) k} (hg : InN p g) {h : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k} (hh : h.BlockTriangular (lev p)) (r s) (hrs : lev p s ≤ lev p r) : (g.val * h) r s = h r s

theorem **mem_N_iff** {p : ℕ} {g : GL (n ⊕ Fin c) k} : g ∈ N k n c p ↔ InN p g

theorem **rowE_mem** (p : ℕ) (u : n → k) : rowE c p u ∈ N k n c (p + 1)

theorem **colE_mem** (p : ℕ) (β : n → k) : colE c (p + 1) β ∈ N k n c p

theorem **rowE_not_mem** (l : Fin c) (u : n → k) (hu : u ≠ 0) : rowE c (l.val + 1) u ∉ N k n c (l.val + 1)

theorem **simple_entry_mul** {p : ℕ} {g h : GL (n ⊕ Fin c) k} (hg : InN p g) (hh : InN p h) (i j : Fin c) (hij : j.val = i.val + 1) : (g * h).val (Sum.inr i) (Sum.inr j) = g.val (Sum.inr i) (Sum.inr j) + h.val (Sum.inr i) (Sum.inr j)

theorem **simpleSum_mul** {p : ℕ} {g h : GL (n ⊕ Fin c) k} (hg : g ∈ N k n c p) (hh : h ∈ N k n c p) : simpleSum (g * h) = simpleSum g + simpleSum h

theorem **conj_mem** {p : ℕ} {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c p) (g : GL n k) : (embX c g)⁻¹ * u * embX c g ∈ N k n c p

theorem **simpleSum_conj** (u : GL (n ⊕ Fin c) k) (g : GL n k) : simpleSum ((embX c g)⁻¹ * u * embX c g) = simpleSum u

theorem **simpleSum_row_col** (l : Fin c) (hl : l.val + 1 < c) (u β : n → k) : simpleSum (rowE c (l.val + 1) u * colE c (l.val + 1 + 1) β * (rowE c (l.val + 1) u)⁻¹) = u ·ᵥ β

theorem **affPt_block** {n' : Type*} [Fintype n'] [DecidableEq n'] (p : ℕ) (x₀ : Matrix n n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) (r : n) (s : n') : affPt p x₀ e (Sum.inl r) (Sum.inl s) = x₀ r s

theorem **free_last** {n' : Type*} (p : ℕ) (hp : p ≤ c) (r : n ⊕ Fin c) : Free (n

theorem **det_mkN** (p : ℕ) (e : {rs : (n ⊕ Fin c) × (n ⊕ Fin c) // lev p rs.1 < lev p rs.2} → k) : (mkN p e).det = 1

theorem **coord_embedding** (p : ℕ) : MeasurableEmbedding (fun (g : N k n c p) (rs : {rs : (n ⊕ Fin c) × (n ⊕ Fin c) // lev p rs.1 < lev p rs.2}) => (g : GL (n ⊕ Fin c) k).val rs.1.1 rs.1.2) ∧ Function.Surjective (fun (g : N k n c p) (rs : {rs : (n ⊕ Fin c) × (n ⊕ Fin c) // lev p rs.1 < lev p rs.2}) => (g : GL (n ⊕ Fin c) k).val rs.1.1 rs.1.2)

theorem **volN_ne_zero** (ν : Measure k) [SigmaFinite ν] (hν : ν ≠ 0) (p : ℕ) : volN ν n c p ≠ 0

theorem **chi_mul** (ψ : AddChar k Circle) (p : ℕ) {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c p) (h) : chi ψ p (u * h) = (ψ (simpleSum u) : ℂ) * chi ψ p h

theorem **reg_single** {G : Type*} [Group G] (g h : G) (x : ℂ) : reg G g (Finsupp.single h x) = Finsupp.single (g * h) x

theorem **exists_isPos** (ψ : AddChar k Circle) (p : ℕ) : ∃ b : (GL n k →₀ ℂ) →₁[ℂ] (GL (n ⊕ Fin c) k →₀ ℂ) →₁[ℂ] ℂ, IsPos ψ p (reg (GL n k)) (reg (GL (n ⊕ Fin c) k)) b ∧ b ≠ 0

Development.AbelianVarietyTheory.Definitions.Foundations.SmoothPartStable

theorem **twKer_le_comap** (ρ : Representation k G V) (N : Subgroup G) (g : G) (hg : g ∈ Subgroup.normalizer (N : Set G)) : twKer ρ N 1 ≤ (twKer ρ N 1).comap (ρ g)

theorem **smoothPart_stable** (ρ : Representation k G V) (g : G) (v : V) (hv : v ∈ smoothPart ρ) : ρ g v ∈ smoothPart ρ

Development.AbelianVarietyTheory.Definitions.Foundations.StandardFiveNontrivial

theorem **mem_one** (L : Lev g N) : L.Mem 1

theorem **mem_mul** (L : Lev g N) {u w : GG g N} (hu : L.Mem u) (hw : L.Mem w) : L.Mem (u * w)

theorem **entry_mul** (L : Lev g N) {u w : GG g N} (hu : L.Mem u) (hw : L.Mem w) {a b : Fin N} (h : L.Edge a b) : (u * w).val a b = u.val a b + w.val a b

theorem **chi_mul** (L : Lev g N) {u w : GG g N} (hu : L.Mem u) (hw : L.Mem w) : L.chi (u * w) = L.chi u * L.chi w

theorem **mem_inv** (L : Lev g N) {u : GG g N} (hu : L.Mem u) : L.Mem u⁻¹

theorem **norm_chi** (L : Lev g N) (u : GG g N) : ‖L.chi u‖ = 1

theorem **chi_ne** (L : Lev g N) (u : GG g N) : L.chi u ≠ 0

theorem **chi_one** (L : Lev g N) : L.chi 1 = 1

theorem **chi_inv** (L : Lev g N) {u : GG g N} (hu : L.Mem u) : L.chi u⁻¹ = (L.chi u)⁻¹

theorem **isProj_e** (L : Lev g N) : IsProj L.e

theorem **standard_edge_five** (a b : Fin 5) : (levStandard g 5 true 1 2 (by norm_num)).Edge a b ↔ (a, b) = (0, 3) ∨ (a, b) = (1, 2) ∨ (a, b) = (3, 4)

theorem **transv_apply** (a b : Fin N) (h : a ≠ b) (i j : Fin N) : (transv (g

theorem **isUp_transv** (a b : Fin N) (h : a < b) : IsUp g N (transv a b h.ne)

theorem **standard_five_nontrivial** : (∃ u, (levStandard g 5 true 1 2 (by norm_num)).Mem u ∧ (levStandard g 5 true 1 2 (by norm_num)).chi u ≠ 1) ∧ ∃ u, IsUp g 5 u ∧ ¬ (levStandard g 5 true 1 2 (by norm_num)).Mem u

theorem **primeOK_example** : PrimeOK 6529 3 17 ∧ Nat.Prime 6529

theorem **coeff_alg** (e : G → ℂ) (x : G) : (alg e).coeff x = e x

theorem **alg_coeff** (a : MonoidAlgebra ℂ G) : alg (fun x => a.coeff x) = a

theorem **alg_injective** : Function.Injective (alg : (G → ℂ) → MonoidAlgebra ℂ G)

theorem **coeff_mul_eq_conv** (a b : MonoidAlgebra ℂ G) (x : G) : (a * b).coeff x = conv (fun y => a.coeff y) (fun y => b.coeff y) x

theorem **alg_conv** (a b : G → ℂ) : alg (conv a b) = alg a * alg b

theorem **alg_dirac** : alg (dirac : G → ℂ) = 1

theorem **op_eq_asAlgebraHom** {M : Type} [AddCommGroup M] [Module ℂ M] (ρ : Representation ℂ G M) (e : G → ℂ) : op ρ e = ρ.asAlgebraHom (alg e)

theorem **asAlgebraHom_eq_op** {M : Type} [AddCommGroup M] [Module ℂ M] (ρ : Representation ℂ G M) (a : MonoidAlgebra ℂ G) : ρ.asAlgebraHom a = op ρ (fun x => a.coeff x)

theorem **op_conv** {M : Type} [AddCommGroup M] [Module ℂ M] (ρ : Representation ℂ G M) (a b : G → ℂ) : op ρ (conv a b) = op ρ a ∗ op ρ b

theorem **op_dirac** {M : Type} [AddCommGroup M] [Module ℂ M] (ρ : Representation ℂ G M) : op ρ dirac = 1

theorem **conv_assoc** (a b c : G → ℂ) : conv (conv a b) c = conv a (conv b c)

theorem **adj_adj** (e : G → ℂ) : adj (adj e) = e

theorem **adj_conv** (a b : G → ℂ) : adj (conv a b) = conv (adj b) (adj a)

theorem **eq_zero_of_conv_adj** {a : G → ℂ} (h : conv a (adj a) = 0) : a = 0

theorem **alg_sub** (a b : G → ℂ) : alg (a − b) = alg a − alg b

theorem **isProj_of_central** {e : G → ℂ} (he : conv e e = e) (hc : ∀ a, conv e a = conv a e) : IsProj e

theorem **conv_tens** {ι : Type} [Fintype ι] (a b : ι → G → ℂ) : conv (tens a) (tens b) = tens fun v => conv (a v) (b v)

theorem **conv_tens2** {G' : Type} [Group G'] [Fintype G'] (a a' : G → ℂ) (b b' : G' → ℂ) : conv (tens2 a b) (tens2 a' b') = tens2 (conv a a') (conv b b')

theorem **isProj_tens** {ι : Type} [Fintype ι] {a : ι → G → ℂ} (h : ∀ v, IsProj (a v)) : IsProj (tens a)

theorem **isProj_tens2** {G' : Type} [Group G'] [Fintype G'] {a : G → ℂ} {b : G' → ℂ} (ha : IsProj a) (hb : IsProj b) : IsProj (tens2 a b)

theorem **rk_eq_sum_trace** {M : Type} [AddCommGroup M] [Module ℂ M] [FiniteDimensional ℂ M] (ρ : Representation ℂ G M) {e : G → ℂ} (he : conv e e = e) : (rk ρ e : ℂ) = ∑ γ, e γ ∗ LinearMap.trace ℂ M (ρ γ)

theorem **countAt_of_trace_eq_zero** {g₁ g₂ N : ℕ} [Fact g₁.Prime] [Fact g₂.Prime] {ι₁ ι₂ : Type} [Fintype ι₁] [Fintype ι₂] {M : Type} [AddCommGroup M] [Module ℂ M] [FiniteDimensional ℂ M] [Nontrivial M] (ρ : Representation ℂ ((ι₁ → GG g₁ N) × (ι₂ → GG g₂ N)) M) (h : ∀ γ, γ ≠ 1 → AllUnip γ.1 ∨ AllUnip γ.2 → LinearMap.trace ℂ M (ρ γ) = 0) : CountAt ρ 0

theorem **op_reg** (e f : G → ℂ) : op (regFn G) e f = conv e f

theorem **trace_op_reg** (e : G → ℂ) : LinearMap.trace ℂ _ (op (regFn G) e) = Fintype.card G ∗ e 1

theorem **rk_reg** (e : G → ℂ) (he : conv e e = e) : (rk (regFn G) e : ℂ) = Fintype.card G ∗ e 1

theorem **countAt_reg** {g₁ g₂ N : ℕ} [Fact g₁.Prime] [Fact g₂.Prime] {ι₁ ι₂ : Type} [Fintype ι₁] [Fintype ι₂] : CountAt (regFn ((ι₁ → GG g₁ N) × (ι₂ → GG g₂ N))) 0

theorem **isProj_dirac** {G : Type} [Group G] [Fintype G] : IsProj (dirac : G → ℂ)

theorem **finrank_pos_of_countAt** {g₁ g₂ N : ℕ} [Fact g₁.Prime] [Fact g₂.Prime] {ι₁ ι₂ : Type} [Fintype ι₁] [Fintype ι₂] {M : Type} [AddCommGroup M] [Module ℂ M] (ρ : Representation ℂ ((ι₁ → GG g₁ N) × (ι₂ → GG g₂ N)) M) {ep : ℝ} (hep : ep < 1) (h : CountAt ρ ep) : 0 < Module.finrank ℂ M

theorem **not_isSeries_zero** (τ : (ZMod g)[X] → GG g 0 → ℂ) : ¬ IsSeries τ

theorem **char_lineRep** (χ : (ZMod g)ˣ →∗ ℂˣ) (γ : GG g N) : (FDRep.of (lineRep χ)).character γ = ((χ (Matrix.GeneralLinearGroup.det γ) : ℂˣ) : ℂ)

theorem **mem_sqSet_one** {f : (ZMod g)[X]} (hf : f ∈ sqSet g 1) : f = X + C (f.coeff 0) ∧ f.coeff 0 ≠ 0

theorem **exists_isSeries_one** : ∃ τ : (ZMod g)[X] → GG g 1 → ℂ, IsSeries τ

theorem **isNilpotent_of_isUp** {u : GG g N} (hu : IsUp g N u) : IsNilpotent (u.val − 1)

theorem **allUnip_of_tens_e** {ι : Type} [Fintype ι] (L : ι → Lev g N) (γ : ι → GG g N) (h : tens (fun v => (L v).e) γ ≠ 0) : AllUnip γ

Development.AbelianVarietyTheory.Definitions.Foundations.StrictInterlaceIffDescEnum

theorem **mem_entries** {S : SPar} {x : ℚ} : x ∈ entries S ↔ ∃ s, (x, s) ∈ S

theorem **fst_mem_entries** {S : SPar} {x : ℚ × Bool} (h : x ∈ S) : x.1 ∈ entries S

theorem **mem_ofSign** {E : Finset ℚ} {sg : ℚ → Bool} {x : ℚ × Bool} : x ∈ ofSign E sg ↔ x.1 ∈ E ∧ x.2 = sg x.1

@[simp] theorem **mem_plus** {E : Finset ℚ} {x : ℚ × Bool} : x ∈ plus E ↔ x.1 ∈ E ∧ x.2 = true

theorem **mem_shift** {h : ℚ} {S : SPar} {x : ℚ × Bool} : x ∈ shift h S ↔ (x.1 − h, x.2) ∈ S

theorem **mem_negate** {S : SPar} {x : ℚ × Bool} : x ∈ negate S ↔ (−x.1, x.2) ∈ S

theorem **mem_flip** {S : SPar} {x : ℚ × Bool} : x ∈ flip S ↔ (x.1, !x.2) ∈ S

@[simp] theorem **shift_zero** (S : SPar) : shift 0 S = S

@[simp] theorem **entries_union** (S T : SPar) : entries (S ∪ T) = entries S ∪ entries T

@[simp] theorem **entries_ofSign** (E : Finset ℚ) (sg : ℚ → Bool) : entries (ofSign E sg) = E

@[simp] theorem **entries_plus** (E : Finset ℚ) : entries (plus E) = E

theorem **plus_union** (E E' : Finset ℚ) : plus (E ∪ E') = plus E ∪ plus E'

@[simp] theorem **plus_empty** : plus ∅ = ∅

theorem **isRegular_ofSign** (E : Finset ℚ) (sg : ℚ → Bool) : IsRegular (ofSign E sg)

theorem **isRegular_plus** (E : Finset ℚ) : IsRegular (plus E)

theorem **IsRegular.sign_eq** {S : SPar} (h : IsRegular S) {x : ℚ × Bool} (hx : x ∈ S) : sign S x.1 = x.2

theorem **IsRegular.eq_ofSign** {S : SPar} (h : IsRegular S) : S = ofSign (entries S) (sign S)

theorem **isRegular_iff** {S : SPar} : IsRegular S ↔ ∃ E sg, S = ofSign E sg

theorem **IsRegular.card_entries** {S : SPar} (h : IsRegular S) : (entries S).card = S.card

theorem **isRegular_iff_card** {S : SPar} : IsRegular S ↔ (entries S).card = S.card

theorem **IsRegular.union** {S T : SPar} (hS : IsRegular S) (hT : IsRegular T) (h : Disjoint (entries S) (entries T)) : IsRegular (S ∪ T)

theorem **nPlus_add_nMinus** (S : SPar) : nPlus S + nMinus S = S.card

@[simp] theorem **nMinus_plus** (E : Finset ℚ) : nMinus (plus E) = 0

@[simp] theorem **nPlus_plus** (E : Finset ℚ) : nPlus (plus E) = E.card

@[simp] theorem **mem_descList** {E : Finset ℚ} {x : ℚ} : x ∈ descList E ↔ x ∈ E

theorem **pairwise_descList** (E : Finset ℚ) : (descList E).Pairwise (· > ·)

theorem **descList_eq** {E : Finset ℚ} {L : List ℚ} (hL : L.Pairwise (· > ·)) (h : L.toFinset = E) : descList E = L

theorem **strictAnti_descEnum** {n : ℕ} (E : Finset ℚ) (h : E.card = n) : StrictAnti (descEnum E h)

theorem **descEnum_mem** {n : ℕ} (E : Finset ℚ) (h : E.card = n) (i : Fin n) : descEnum E h i ∈ E

theorem **image_descEnum** {n : ℕ} (E : Finset ℚ) (h : E.card = n) : univ.image (descEnum E h) = E

theorem **descEnum_unique** {n : ℕ} {E : Finset ℚ} (h : E.card = n) {f : Fin n → ℚ} (hf : StrictAnti f) (hm : ∀ i, f i ∈ E) : descEnum E h = f

theorem **highestWeight_int** {n : ℕ} {E : Finset ℚ} (h : E.card = n) (hE : InCoset (((n : ℚ) - 1) / 2) E) (i : Fin n) : ∃ z : ℤ, highestWeight E h i = z

theorem **highestWeight_antitone** {n : ℕ} {E : Finset ℚ} (h : E.card = n) {c : ℚ} (hE : InCoset c E) : Antitone (highestWeight E h)

theorem **goodPar_iff** {n p q : ℕ} {S : SPar} : GoodPar n p q S ↔ IsHC n S ∧ signature S = (p, q)

theorem **mem_ncDists** {S : SPar} {d : ℚ} : d ∈ ncDists S ↔ ∃ x ∈ S, ∃ y ∈ S, x.2 ≠ y.2 ∧ |x.1 - y.1| = d

theorem **le_ncGap_iff** {S : SPar} {h : ℚ} : (h : WithTop ℚ) ≤ ncGap S ↔ ∀ x ∈ S, ∀ y ∈ S, x.2 ≠ y.2 → h ≤ |x.1 - y.1|

theorem **ncGap_eq_top_iff** {S : SPar} : ncGap S = ⊤ ↔ ∀ x ∈ S, ∀ y ∈ S, x.2 = y.2

theorem **ncGap_eq_coe_iff** {S : SPar} {g : ℚ} : ncGap S = (g : WithTop ℚ) ↔ (∃ x ∈ S, ∃ y ∈ S, x.2 ≠ y.2 ∧ |x.1 - y.1| = g) ∧ ∀ x ∈ S, ∀ y ∈ S, x.2 ≠ y.2 → g ≤ |x.1 - y.1|

theorem **lt_ncGap_iff** {S : SPar} {h : ℚ} : (h : WithTop ℚ) < ncGap S ↔ ∀ x ∈ S, ∀ y ∈ S, x.2 ≠ y.2 → h < |x.1 - y.1|

@[simp] theorem **ncGap_plus** (E : Finset ℚ) : ncGap (plus E) = ⊤

theorem **mem_block** {c x : ℚ} {t : ℕ} : x ∈ TPar.block c t ↔ ∃ a : ℕ, a < t ∧ x = c + ((t : ℚ) - 1) / 2 - a

@[simp] theorem **card_block** (c : ℚ) (t : ℕ) : (TPar.block c t).card = t

theorem **abs_sub_le_of_mem_block** {c x : ℚ} {t : ℕ} (h : x ∈ TPar.block c t) : |x - c| ≤ ((t : ℚ) - 1) / 2

theorem **neg_mem_block** {c x : ℚ} {t : ℕ} : -x ∈ TPar.block (-c) t ↔ x ∈ TPar.block c t

theorem **sub_mem_block** {c x h : ℚ} {t : ℕ} : x - h ∈ TPar.block c t ↔ x ∈ TPar.block (c + h) t

theorem **inCoset_block** (c : ℚ) (t : ℕ) : InCoset (c + ((t : ℚ) + 1) / 2) (TPar.block c t)

theorem **mem_image_change** {P : ℚ → Prop} [DecidablePred P] {S : SPar} {x : ℚ × Bool} : x ∈ (S.image fun y => (y.1, if P y.1 then !y.2 else y.2)) ↔ (¬ P x.1 ∧ x ∈ S) ∨ (P x.1 ∧ (x.1, !x.2) ∈ S)

theorem **mem_liftPlus** {c : ℚ} {t : ℕ} {S : SPar} {x : ℚ × Bool} : x ∈ liftPlus c t S ↔ (c ≤ x.1 ∧ x ∈ S) ∨ (x.1 < c ∧ (x.1, !x.2) ∈ S) ∨ (x.1 ∈ TPar.block c t ∧ x.2 = true)

theorem **mem_liftMinus** {c : ℚ} {t : ℕ} {S : SPar} {x : ℚ × Bool} : x ∈ liftMinus c t S ↔ (x.1 ≤ c ∧ x ∈ S) ∨ (c < x.1 ∧ (x.1, !x.2) ∈ S) ∨ (x.1 ∈ TPar.block c t ∧ x.2 = true)

theorem **entries_lift** (o : Orientation) (c : ℚ) (t : ℕ) (S : SPar) : entries (lift o c t S) = entries S ∪ TPar.block c t

theorem **liftMinus_eq_negate** (c : ℚ) (t : ℕ) (S : SPar) : liftMinus c t S = negate (liftPlus (-c) t (negate S))

theorem **lift_shift** (o : Orientation) (c h : ℚ) (t : ℕ) (S : SPar) : lift o (c + h) t (shift h S) = shift h (lift o c t S)

theorem **FarFrom.disjoint_block** {c : ℚ} {t : ℕ} {S : SPar} (h : FarFrom c (((t : ℚ) + 1) / 2) S) : Disjoint (entries S) (TPar.block c t)

theorem **IsRegular.lift** {S : SPar} (hS : IsRegular S) (o : Orientation) {c : ℚ} {t : ℕ} (h : Disjoint (entries S) (TPar.block c t)) : IsRegular (lift o c t S)

theorem **IsRegular.card_lift** {S : SPar} (hS : IsRegular S) (o : Orientation) {c : ℚ} {t : ℕ} (h : Disjoint (entries S) (TPar.block c t)) : (SignedHC.lift o c t S).card = S.card + t

theorem **InCoset.lift** {c : ℚ} {t : ℕ} {S : SPar} (h : InCoset (c + ((t : ℚ) + 1) / 2) (entries S)) (o : Orientation) : InCoset (c + ((t : ℚ) + 1) / 2) (entries (lift o c t S))

theorem **nMinus_change** (P : ℚ → Prop) [DecidablePred P] (S : SPar) (E : Finset ℚ) : nMinus ((S.image fun y => (y.1, if P y.1 then !y.2 else y.2)) ∪ plus E) = (S.filter fun x => ¬ P x.1 ∧ x.2 = false).card + (S.filter fun x => P x.1 ∧ x.2 = true).card

theorem **nMinus_liftPlus** (c : ℚ) (t : ℕ) (S : SPar) : nMinus (liftPlus c t S) = (S.filter fun x => c ≤ x.1 ∧ x.2 = false).card + (S.filter fun x => x.1 < c ∧ x.2 = true).card

theorem **nMinus_liftMinus** (c : ℚ) (t : ℕ) (S : SPar) : nMinus (liftMinus c t S) = (S.filter fun x => x.1 ≤ c ∧ x.2 = false).card + (S.filter fun x => c < x.1 ∧ x.2 = true).card

theorem **mem_ofIx_ofBlocks** {p q : ℕ} {l₁ l₂ : List ℚ} (hp : l₁.length = p) (hq : l₂.length = q) {x : ℚ × Bool} : x ∈ ofIx (ofBlocks l₁ l₂ : Ix p q → ℚ) ↔ (x.1 ∈ l₁ ∧ x.2 = true) ∨ (x.1 ∈ l₂ ∧ x.2 = false)

theorem **liftPlus_lamA** (t : ℕ) {al be ga de : List ℚ} (hp : ∀ e ∈ al ++ ga, 0 < e) (hn : ∀ e ∈ be ++ de, e < 0) : liftPlus 0 t (ofIx (lamA al be ga de)) = ofIx (lamA' t al be ga de)

theorem **gapGt_iff** {p q : ℕ} (lam : Ix p q → ℚ) (h : ℚ) : GapGt lam (h : ℝ) ↔ (h : WithTop ℚ) < ncGap (ofIx lam)

theorem **isChain_descList_iff** {E : Finset ℚ} {R : ℚ → ℚ → Prop} : (descList E).IsChain R ↔ ∀ x ∈ E, ∀ y ∈ E, y < x → (∀ z ∈ E, ¬ (y < z ∧ z < x)) → R x y

theorem **HePair.symm** : ∀ {a b : Letter}, HePair a b → HePair b a

theorem **HePair.swap** : ∀ {a b : Letter}, HePair a b → HePair a.swap b.swap

theorem **heInterlace_iff** {S T : SPar} : HeInterlace S T ↔ IsRegular S ∧ IsRegular T ∧ Disjoint (entries S) (entries T) ∧ ∀ x ∈ entries S ∪ entries T, ∀ y ∈ entries S ∪ entries T, y < x → (∀ z ∈ entries S ∪ entries T, ¬ (y < z ∧ z < x)) → HePair (letter S T x) (letter S T y)

theorem **word_eq** {S T : SPar} {L : List ℚ} (hL : L.Pairwise (· > ·)) (h : L.toFinset = entries S ∪ entries T) : word S T = L.map (letter S T)

theorem **letter_left** {S : SPar} (hS : IsRegular S) (T : SPar) {x : ℚ × Bool} (hx : x ∈ S) : letter S T x.1 = if x.2 then .plus else .minus

theorem **letter_right** {S T : SPar} (hT : IsRegular T) (h : Disjoint (entries S) (entries T)) {x : ℚ × Bool} (hx : x ∈ T) : letter S T x.1 = if x.2 then .circPlus else .circMinus

theorem **HeInterlace.flip** {S T : SPar} (h : HeInterlace S T) : HeInterlace (flip S) (flip T)

theorem **HeInterlace.negate** {S T : SPar} (h : HeInterlace S T) : HeInterlace (negate S) (negate T)

theorem **strictAnti_of_interlace** {n : ℕ} {a : Fin (n + 1) → ℚ} {b : Fin n → ℚ} (h : ∀ i : Fin n, b i < a i.castSucc ∧ a i.succ < b i) : StrictAnti a ∧ StrictAnti b

theorem **strictInterlace_iff_descEnum** {n : ℕ} {A B : Finset ℚ} (hA : A.card = n + 1) (hB : B.card = n) : StrictInterlace A B ↔ ∀ i : Fin n, descEnum B hB i < descEnum A hA i.castSucc ∧ descEnum A hA i.succ < descEnum B hB i

theorem **StrictInterlace.card_eq** {A B : Finset ℚ} (h : StrictInterlace A B) : A.card = B.card + 1

theorem **strictInterlace_iff** {A B : Finset ℚ} : StrictInterlace A B ↔ Disjoint A B ∧ A.Nonempty ∧ (∀ y ∈ B, ∃ x ∈ A, y < x) ∧ (∀ y ∈ B, ∃ x ∈ A, x < y) ∧ ∀ x ∈ A ∪ B, ∀ y ∈ A ∪ B, y < x → (∀ z ∈ A ∪ B, ¬ (y < z ∧ z < x)) → (x ∈ A ↔ y ∉ A)

theorem **strictInterlace_iff_he** {A B : Finset ℚ} : StrictInterlace A B ↔ HeInterlace (plus A) (plus B) ∧ A.Nonempty ∧ (∀ y ∈ B, ∃ x ∈ A, y < x) ∧ ∀ y ∈ B, ∃ x ∈ A, x < y

theorem **n_eq** : G.n = G.k + G.m

theorem **a_m** : G.a G.m = G.v 2 + G.ga + 1 / 2

theorem **a_eq_succ_add** (i : ℕ) : G.a i = G.a (i + 1) + G.ga

theorem **a_pred** {i : ℕ} (hi : 1 ≤ i) : G.a (i − 1) = G.a i + G.ga

theorem **a_sub_a** (i j : ℕ) : G.a i − G.a j = ((j : ℚ) − i) * G.ga

theorem **mid_eq** {i : ℕ} (hi : 1 ≤ i) : G.mid i = G.a i + G.ga / 2

theorem **nearMid_of_floor** (h : ∀ i ∈ Icc 1 G.m, G.cen i = ⌊G.mid i⌋) : G.NearMid

theorem **nearMid_of_ceil** (h : ∀ i ∈ Icc 1 G.m, G.cen i = ⌈G.mid i⌉ − 1) : G.NearMid

theorem **mem_cloud** {x : ℚ} : x ∈ G.cloud ↔ ∃ i, 2 ≤ i ∧ i ≤ G.k ∧ x = G.v i

theorem **mem_lows** {x : ℚ} : x ∈ G.lows ↔ ∃ l, 1 ≤ l ∧ l ≤ G.m ∧ x = −(G.b l : ℚ)

theorem **mem_cloudHalf** {x : ℚ} : x ∈ G.cloudHalf ↔ ∃ i, 2 ≤ i ∧ i ≤ G.k ∧ x = G.vHalf i

theorem **mem_tops** {x : ℚ} : x ∈ G.tops ↔ ∃ i, i ≤ G.m ∧ x = G.a i

theorem **mem_lam** {x : ℚ × Bool} : x ∈ G.lam ↔ (∃ i, i ≤ G.m ∧ x = (G.a i, decide (Even (G.m − i)))) ∨ (x.1 ∈ G.cloudHalf ∧ x.2 = true)

theorem **mem_discrete_parameter_iff** {x : ℚ × Bool} : x ∈ G.parD0 ↔ (x.1 ∈ G.cloud ∧ x.2 = true) ∨ ∃ l, 1 ≤ l ∧ l ≤ G.m ∧ x = (−(G.b l : ℚ), decide (Even (l − 1)))

theorem **mem_parD** {i j : ℕ} {x : ℚ × Bool} : x ∈ G.parD i j ↔ (x.1 ∈ G.cloud ∧ x.2 = true) ∨ (∃ l, 1 ≤ l ∧ l ≤ G.m ∧ x = (−(G.b l : ℚ), decide (l ≤ j + 1 ∨ Even (l − j − 1)))) ∨ (∃ l, 1 ≤ l ∧ l ≤ j ∧ x.1 ∈ G.blockM l ∧ x.2 = true) ∨ (∃ l, G.m − i + 1 ≤ l ∧ l ≤ G.m ∧ x.1 ∈ G.blockC l ∧ x.2 = true)

theorem **mem_parPi** {i : ℕ} {x : ℚ × Bool} : x ∈ G.parPi i ↔ (∃ l, l ≤ G.m ∧ x = (G.a l, decide (G.m − i ≤ l ∨ Even (G.m − i − l)))) ∨ (x.1 ∈ G.cloudHalf ∧ x.2 = true) ∨ (∃ l, G.m − i + 1 ≤ l ∧ l ≤ G.m ∧ x.1 ∈ G.blockC' l ∧ x.2 = true)

theorem **entries_lam** : entries G.lam = G.tops ∪ G.cloudHalf

theorem **entries_discrete_parameter** : entries G.parD0 = G.cloud ∪ G.lows

theorem **entries_parD** (i j : ℕ) : entries (G.parD i j) = G.cloud ∪ G.lows ∪ (Icc 1 j).biUnion G.blockM ∪ (Icc (G.m − i + 1) G.m).biUnion G.blockC

theorem **entries_parPi** (i : ℕ) : entries (G.parPi i) = entries G.lam ∪ (Icc (G.m − i + 1) G.m).biUnion G.blockC'

theorem **mem_weightX** {j : ℕ} {x : ℚ} : x ∈ G.weightX j ↔ x ∈ G.tops ∨ x ∈ G.cloudHalf ∨ (∃ i, 1 ≤ i ∧ i ≤ G.m ∧ x ∈ G.blockC' i) ∨ ∃ l, 1 ≤ l ∧ l ≤ j ∧ x ∈ G.blockMPlus l

theorem **parPi_zero** : G.parPi 0 = G.lam

theorem **parD_zero_zero** : G.parD 0 0 = G.parD0

theorem **parPi_m** : G.parPi G.m = G.parX 0

theorem **weightX_succ** (j : ℕ) : G.weightX (j + 1) = G.weightX j ∪ G.blockMPlus (j + 1)

theorem **dpar_eq** {D : D7} (P : TPar D) (τ : D.what₂.Tau) (hk : G.k = D.R.k) (hm : G.m = D.R.m) (ht : G.t = P.t) (hv : ∀ i, P.v τ i = G.v (i.1 + 2)) (hb : ∀ l, P.b τ l = G.b (l.1 + 1)) (he : ∀ l, P.e τ l = G.e (l.1 + 1)) {j : ℕ} (hj : j ≤ G.m) : P.Dpar j τ = shift (if Odd j then P.eta τ else 0) (G.parD 0 j)

theorem **extend_cast** {n : ℕ} (d : ℕ) {f : Fin n → ℚ} (hf : ∀ i, ∃ z : ℤ, f i = z) (i : Fin n) : (extend d f (i.1 + d) : ℚ) = f i

theorem **ofHCParam_spec** {R : Ranks} (P : HCParam R) (s ga : ℕ) (cen : ℕ → ℤ) : (∀ i, P.v i = (ofHCParam P s ga cen).v (i.1 + 2)) ∧ (∀ l, P.b l = (ofHCParam P s ga cen).b (l.1 + 1)) ∧ ∀ l, P.e l = (ofHCParam P s ga cen).e (l.1 + 1)

theorem **exGrid_lam** : exGrid.lam = {(53 / 2, true), (37 / 2, false), (21 / 2, true), (3 / 2, true), (1 / 2, true), (−1 / 2, true)}

theorem **example_discrete_parameter** : exGrid.parD0 = {(2, true), (1, true), (0, true), (−1, true), (−7, false)}

theorem **exGrid_parD** : exGrid.parD 1 1 = plus {15, 14, 13, 2, 1, 0, −1, −3, −4, −5, −7}

theorem **exGrid_parPi** : exGrid.parPi 1 = {(53 / 2, false), (37 / 2, true), (29 / 2, true), (27 / 2, true), (21 / 2, true), (3 / 2, true), (1 / 2, true), (−1 / 2, true)}

theorem **exGrid_signature** : signature exGrid.lam = (5, 1) ∧ signature exGrid.parD0 = (4, 1) ∧ ncGap exGrid.lam = ((8 : ℚ) : WithTop ℚ) ∧ ncGap exGrid.parD0 = ((6 : ℚ) : WithTop ℚ) ∧ exGrid.NearMid

theorem **exGrid_word** : word exGrid.lam exGrid.parD0 = [.plus, .minus, .plus, .circPlus, .plus, .circPlus, .plus, .circPlus, .plus, .circPlus, .circMinus]

theorem **exGrid_interlace** : HeInterlace exGrid.lam exGrid.parD0 ∧ StrictInterlace (exGrid.weightX 2) (entries (exGrid.parD 2 2)) ∧ ¬ StrictInterlace (exGrid.weightX 1) (entries (exGrid.parD 2 2))

theorem **exGrid_lift** : exGrid.parD 1 0 = liftMinus 14 3 (exGrid.parD 0 0) ∧ exGrid.parD 0 1 = liftPlus (−4) 3 (exGrid.parD 0 0) ∧ exGrid.parPi 1 = liftMinus 14 2 (exGrid.parPi 0) ∧ exGrid.parD 1 0 ≠ liftPlus 14 3 (exGrid.parD 0 0) ∧ exGrid.parD 0 1 ≠ liftMinus (−4) 3 (exGrid.parD 0 0)

Development.AbelianVarietyTheory.Definitions.Foundations.SumMulCoPtSbEqSumInv

theorem **coPtSb_one** (N p m k : ℕ) (ψ : DirichletCharacter K (p ^ m)) (c d x : ℤ) (s : ℕ) : coPtSb N p m k ψ c d x 1 s = GlobalTheory.KummerEisensteinGalois.coPtS N p m k ψ c d x s

theorem **ffMu_congr** (N p m k : ℕ) (ψ : DirichletCharacter K (p ^ m)) {a a' : ℤ} (h : a ≡ a' [ZMOD N]) : GlobalTheory.KummerEisensteinGalois.ffMu N p m k ψ a = GlobalTheory.KummerEisensteinGalois.ffMu N p m k ψ a'

theorem **eeMu_congr** (N p m : ℕ) (ψ : DirichletCharacter K (p ^ m)) {b b' : ℤ} (h : b ≡ b' [ZMOD N]) : GlobalTheory.KummerEisensteinGalois.eeMu N p m ψ b = GlobalTheory.KummerEisensteinGalois.eeMu N p m ψ b'

theorem **coPtMu_congr** (N p m k : ℕ) (ψ : DirichletCharacter K (p ^ m)) {a a' b b' : ℤ} (ha : a ≡ a' [ZMOD N]) (hb : b ≡ b' [ZMOD N]) : GlobalTheory.KummerEisensteinGalois.coPtMu N p m k ψ a b = GlobalTheory.KummerEisensteinGalois.coPtMu N p m k ψ a' b'

theorem **coPtSb_congr** (N p m k : ℕ) (ψ : DirichletCharacter K (p ^ m)) (c d : ℤ) {a a' b b' : ℤ} (ha : a ≡ a' [ZMOD N]) (hb : b ≡ b' [ZMOD N]) : coPtSb N p m k ψ c d a b = coPtSb N p m k ψ c d a' b'

theorem **sum_mul_coPtSb_eq_sum_inv** (N p m k : ℕ) [NeZero N] (ψ : DirichletCharacter K (p ^ m)) (χ : DirichletCharacter K N) (c d x : ℤ) (s : ℕ) : ∑ δ : (ZMod N)ˣ, χ (δ : ZMod N) * coPtSb N p m k ψ c d (x * ((δ : ZMod N).val : ℤ)) ((δ : ZMod N).val : ℤ) s = ∑ δ : (ZMod N)ˣ, χ⁻¹ (δ : ZMod N) * coPtSb N p m k ψ c d (x * (((δ⁻¹ : (ZMod N)ˣ) : ZMod N).val : ℤ)) (((δ⁻¹ : (ZMod N)ˣ) : ZMod N).val : ℤ) s

theorem **chiAvg_inv** (N p m k : ℕ) [NeZero N] (ψ : DirichletCharacter K (p ^ m)) (χ : DirichletCharacter K N) (c d x : ℤ) (s : ℕ) : chiAvg N p m k ψ (fun n => χ⁻¹ (n : ZMod N)) c d x s = ∑ δ : (ZMod N)ˣ, χ (δ : ZMod N) * coPtSb N p m k ψ c d (x * ((δ : ZMod N).val : ℤ)) ((δ : ZMod N).val : ℤ) s

theorem **chiAvg_eq_sum_inv** (N p m k : ℕ) [NeZero N] (ψ : DirichletCharacter K (p ^ m)) (χ : DirichletCharacter K N) (c d x : ℤ) (s : ℕ) : chiAvg N p m k ψ (fun n => χ (n : ZMod N)) c d x s = ∑ δ : (ZMod N)ˣ, χ (δ : ZMod N) * coPtSb N p m k ψ c d (x * (((δ⁻¹ : (ZMod N)ˣ) : ZMod N).val : ℤ)) (((δ⁻¹ : (ZMod N)ˣ) : ZMod N).val : ℤ) s

theorem **sum_mul_coPtSb_shift** (N p m k : ℕ) [NeZero N] (ψ : DirichletCharacter K (p ^ m)) (χ : DirichletCharacter K N) (c d x : ℤ) (t : (ZMod N)ˣ) (s : ℕ) : ∑ δ : (ZMod N)ˣ, χ (δ : ZMod N) * coPtSb N p m k ψ c d (((t : ZMod N).val : ℤ) * (x * ((δ : ZMod N).val : ℤ))) (((t : ZMod N).val : ℤ) * ((δ : ZMod N).val : ℤ)) s = (χ (t : ZMod N))⁻¹ * ∑ δ : (ZMod N)ˣ, χ (δ : ZMod N) * coPtSb N p m k ψ c d (x * ((δ : ZMod N).val : ℤ)) ((δ : ZMod N).val : ℤ) s

theorem **chiAvg_shift** (N p m k : ℕ) [NeZero N] (ψ : DirichletCharacter K (p ^ m)) (χ : DirichletCharacter K N) (c d x : ℤ) (t : (ZMod N)ˣ) (s : ℕ) : ∑ δ : (ZMod N)ˣ, (χ (δ : ZMod N))⁻¹ * coPtSb N p m k ψ c d (((t : ZMod N).val : ℤ) * (x * ((δ : ZMod N).val : ℤ))) (((t : ZMod N).val : ℤ) * ((δ : ZMod N).val : ℤ)) s = χ (t : ZMod N) * chiAvg N p m k ψ (fun n => χ (n : ZMod N)) c d x s

Development.AbelianVarietyTheory.Definitions.Foundations.T2SpaceAdeleRing

theorem **hgtH_pos** (g : HA R σ Q) : 0 < hgtH σ Q ν K g

theorem **flatH_apply** (Φ : HA R σ Q → ℂ) (s₀ s : ℂ) (g : HA R σ Q) : flatH σ Q ν K Φ s₀ s g = Φ g * npow (hgtH σ Q ν K g) (s - s₀)

@[simp] theorem **flatH_self** (Φ : HA R σ Q → ℂ) (s₀ : ℂ) : flatH σ Q ν K Φ s₀ s₀ = Φ

theorem **mem_degPSH** {μ : (R ⊗[F] E)ˣ →* Circle} {s : ℂ} {Φ : HA R σ Q → ℂ} : Φ ∈ degPSH σ Q ν K μ s ↔ (∀ (p : siegelH R σ Q) (g : HA R σ Q), Φ (p * g) = μ (aH R σ Q p) * npow (ν (aH R σ Q p)) (s + (n : ℂ) / 2) * Φ g) ∧ Continuous Φ ∧ IsKFinite K Φ

theorem **zetaH_eq** (h : Q = dblGram finSumFinEquiv J) (m : Measure (HA R σ J)) (Φ : HA R σ Q → ℂ) (c : HA R σ J → ℂ) : zetaH σ h m Φ c = ∫ g, Φ (ioH R σ h (g, 1)) * starRingEnd ℂ (c g) ∂m

theorem **zetaConvH_iff** (h : Q = dblGram finSumFinEquiv J) (m : Measure (HA R σ J)) (Φ : HA R σ Q → ℂ) (c : HA R σ J → ℂ) : ZetaConvH σ h m Φ c ↔ Integrable (fun g => Φ (ioH R σ h (g, 1)) * starRingEnd ℂ (c g)) m

theorem **abs_detH** (hν : ∀ x, ν (conjUnit σ x) = ν x) {m : ℕ} {J : Matrix (Fin m) (Fin m) E} (hJ : J.det ≠ 0) (g : HA R σ J) : ν (detH R σ J g) = 1

theorem **apply_ioH_mul** (h : Q = dblGram finSumFinEquiv J) (hν : ∀ x, ν (conjUnit σ x) = ν x) (hJ : J.det ≠ 0) {μ : (R ⊗[F] E)ˣ →* Circle} {s : ℂ} {Φ : HA R σ Q → ℂ} (hΦ : Φ ∈ degPSH σ Q ν K μ s) (x g g' : HA R σ J) : Φ (ioH R σ h (x * g, x * g')) = μ (detH R σ J x) * Φ (ioH R σ h (g, g'))

theorem **continuous_aH** : Continuous (aH R σ Q)

theorem **isClosed_siegelH** : IsClosed (siegelH R σ Q : Set (HA R σ Q))

theorem **abs_aH_eq_one** (p : siegelH R σ Q) (hp : (p : HA R σ Q) ∈ K) : ν (aH R σ Q p) = 1

theorem **hgtH_mul** (p : siegelH R σ Q) {k : HA R σ Q} (hk : k ∈ K) : hgtH σ Q ν K (p * k) = ν (aH R σ Q p)

theorem **hgtH_of_mem** {k : HA R σ Q} (hk : k ∈ K) : hgtH σ Q ν K k = 1

theorem **iwasawaOf** : IwasawaOf finSumFinEquiv (matHom R σ Q) ν K

theorem **hgtH_left** (p : siegelH R σ Q) (g : HA R σ Q) : hgtH σ Q ν K (p * g) = ν (aH R σ Q p) * hgtH σ Q ν K g

theorem **hgtH_right** (g : HA R σ Q) {k : HA R σ Q} (hk : k ∈ K) : hgtH σ Q ν K (g * k) = hgtH σ Q ν K g

theorem **continuous_hgtH** : Continuous (hgtH σ Q ν K)

theorem **flatH_mem** {μ : (R ⊗[F] E)ˣ →* Circle} {s₀ : ℂ} {Φ : HA R σ Q → ℂ} (hΦ : Φ ∈ degPSH σ Q ν K μ s₀) (s : ℂ) : flatH σ Q ν K Φ s₀ s ∈ degPSH σ Q ν K μ s

theorem **spherical_mem** (s : ℂ) : (fun g => npow (hgtH σ Q ν K g) (s + (n : ℂ) / 2)) ∈ degPSH σ Q ν K 1 s

theorem **exists_norm_le** {μ : (R ⊗[F] E)ˣ →* Circle} {s : ℂ} {Φ : HA R σ Q → ℂ} (hΦ : Φ ∈ degPSH σ Q ν K μ s) : ∃ C : ℝ, ∀ g, ‖Φ g‖ ≤ C * (hgtH σ Q ν K g : ℝ) ^ (s.re + (n : ℝ) / 2)

theorem **apply_rat_mul** {μ : (AF F ⊗[F] E)ˣ →* Circle} (hμ : ∀ x : Eˣ, μ (Units.map (inclR (F

theorem **HasIwasawa.fin** (hI : HasIwasawa σ Q (KAd σ Q L a)) (v : HeightOneSpectrum (𝒪 F)) : HasIwasawa σ Q (KFin σ Q L v)

theorem **HasIwasawa.arch** (hI : HasIwasawa σ Q (KAd σ Q L a)) (w : InfinitePlace F) : HasIwasawa σ Q (KArch σ Q w (bc (compInf F E w) a))

theorem **isCompact_KArch** (hc : IsCMExt F E σ) (w : InfinitePlace F) (b : GL (Fin n) (w.Completion ⊗[F] E)) : IsCompact (KArch σ Q w b : Set (GArch σ Q w))

theorem **hgtArch_mul** (hc : IsCMExt F E σ) {w : InfinitePlace F} {b : GL (Fin n) (w.Completion ⊗[F] E)} (p : siegelH w.Completion σ Q) {k : GArch σ Q w} (hk : k ∈ KArch σ Q w b) : hgtArch σ Q w b (p * k) = absEw F E w (aH _ σ Q p)

theorem **isCompact_KFin** (v : HeightOneSpectrum (𝒪 F)) : IsCompact (KFin σ Q L v : Set (GLoc σ Q v))

theorem **isOpen_KFin** (v : HeightOneSpectrum (𝒪 F)) : IsOpen (KFin σ Q L v : Set (GLoc σ Q v))

theorem **isCompact_KAd** (hc : IsCMExt F E σ) : IsCompact (KAd σ Q L a : Set (GAd σ Q))

theorem **t2Space_adeleRing'** : T2Space (AF F)

theorem **hgtFin_mul** {v : HeightOneSpectrum (𝒪 F)} (p : siegelH (Fv F v) σ Q) {k : GLoc σ Q v} (hk : k ∈ KFin σ Q L v) : hgtFin σ Q L v (p ∗ k) = absEv F E v (aH _ σ Q p)

theorem **hgtAd_mul** (hc : IsCMExt F E σ) (p : siegelH (AF F) σ Q) {k : GAd σ Q} (hk : k ∈ KAd σ Q L a) : hgtAd σ Q L a (p ∗ k) = absE F E (aH _ σ Q p)

theorem **AlmostOne.finite** {Φf : ∀ v, GLoc σ Q v → ℂ} (h : AlmostOne σ Q L Φf) (g : GAd σ Q) : (Function.mulSupport fun v => Φf v (proj σ Q v g)).Finite

theorem **finite_mulSupport_hgtFin** (g : GAd σ Q) : (Function.mulSupport fun v => hgtFin σ Q L v (proj σ Q v g)).Finite

theorem **HasIwasawa.hgtAd_eq** (hI : HasIwasawa σ Q (KAd σ Q L a)) (hc : IsCMExt F E σ) (g : GAd σ Q) : hgtAd σ Q L a g = (∏ w, hgtArch σ Q w (bc (compInf F E w) a) (projArch σ Q w g)) ∗ ∏' v, hgtFin σ Q L v (proj σ Q v g)

theorem **isHermND_gram** : IsHermND D.sp.cm.sig V.gram

theorem **exists_isMaj** : ∃ a, IsMaj V a

theorem **matHom_io** (g g' : GA D V) : Formalism.L2.Theta.matHom (dblSp V) (io V (g, g')) = blkGL finSumFinEquiv (Formalism.L2.Theta.matHom V g, Formalism.L2.Theta.matHom V g')

theorem **matHom_ioSum** (W : HSp D) (g : GA D V) (g' : GA D W) : Formalism.L2.Theta.matHom (sumSp V W) (ioSum V W (g, g')) = blkGL finSumFinEquiv (Formalism.L2.Theta.matHom V g, Formalism.L2.Theta.matHom W g')

theorem **io_eq_ioSum** (g : GA D V) (g' : GA D (negSp V)) : ioSum V (negSp V) (g, g') = io V (g, negEquiv V g')

theorem **continuous_io** : Continuous (io V)

theorem **io_injective** : Function.Injective (io V)

theorem **io_mem_GQ** {g g' : GA D V} (h : g ∈ GQ V) (h' : g' ∈ GQ V) : io V (g, g') ∈ GQ (dblSp V)

theorem **io_mem_siegel** {g g' : GA D V} : io V (g, g') ∈ siegel V ↔ g = g'

theorem **Khat_eq** : Khat V a = Kmax (dblSp V) (blkGL finSumFinEquiv (a, a))

theorem **IsMaj.dbl** {V : HSp D} {a : GL (Fin V.rank) (AE D)} (h : IsMaj V a) : IsMaj (dblSp V) (blkGL finSumFinEquiv (a, a))

theorem **io_mem_Khat** {g g' : GA D V} : io V (g, g') ∈ Khat V a ↔ g ∈ Kmax V a ∧ g' ∈ Kmax V a

theorem **isCompact_Khat** : IsCompact (Khat V a : Set (GA D (dblSp V)))

theorem **hgt_mul** (p : siegel V) {k : GA D (dblSp V)} (hk : k ∈ Khat V a) : hgt V a (p ∗ k) = absA D (aP V p)

theorem **mem_degPS** {μ : (AE D)ˣ →∗ Circle} {s : ℂ} {Φ : GA D (dblSp V) → ℂ} : Φ ∈ degPS V a μ s ↔ (∀ (p : siegel V) (g : GA D (dblSp V)), Φ (p ∗ g) = μ (aP V p) ∗ npow (absA D (aP V p)) (s + (V.rank : ℂ) / 2) ∗ Φ g) ∧ Continuous Φ ∧ IsKFinite (Khat V a) Φ

theorem **apply_siegelQ_mul** {μ : (AE D)ˣ →∗ Circle} (hμ : IsHeckeChar D μ) {s : ℂ} {Φ : GA D (dblSp V) → ℂ} (hΦ : Φ ∈ degPS V a μ s) {γ : GA D (dblSp V)} (hγ : γ ∈ siegelQ V) (g : GA D (dblSp V)) : Φ (γ ∗ g) = Φ g

theorem **apply_io_mul** {μ : (AE D)ˣ →∗ Circle} {s : ℂ} {Φ : GA D (dblSp V) → ℂ} (hΦ : Φ ∈ degPS V a μ s) (x g g' : GA D V) : Φ (io V (x ∗ g, x ∗ g')) = μ (detA V x) ∗ Φ (io V (g, g'))

theorem **comp_conjUnit_eq_inv** {μ : (AE D)ˣ →∗ Circle} {k : ℕ} (h : ResEta D μ k) : μ.comp (conjUnit D.sp.cm.sig) = μ⁻¹

theorem **exists_leviH** {R : Type} [CommRing R] [Algebra F R] {σ : E →ₐ[F] E} (hσ : ∀ x, σ (σ x) = x) {Q : Matrix (Fin (n + n)) (Fin (n + n)) E} (h : Q = dblGram finSumFinEquiv J) (hJ : IsHermND σ J) (α : GL (Fin n) (R ⊗[F] E)) : ∃ p : siegelH R σ Q, (p : HA R σ Q) ∈ leviH R σ Q ∧ leviOf finSumFinEquiv (matHom R σ Q) p = α ∧ aH R σ Q p = Matrix.GeneralLinearGroup.det α

Development.AbelianVarietyTheory.Definitions.Foundations.T2SpaceTensorProductFiniteExtension

theorem **norm_map** (φ : R →ₐ[F] R') (x : R ⊗[F] E) : Algebra.norm R' (Algebra.TensorProduct.map φ (AlgHom.id F E) x) = φ (Algebra.norm R x)

theorem **norm_inclR** (x : E) : Algebra.norm R (inclR (F

theorem **norm_sigR** (σ : E →ₐ[F] E) (x : R ⊗[F] E) : Algebra.norm R (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ x) = Algebra.norm R x

theorem **continuous_norm_tens** [TopologicalSpace R] [IsTopologicalRing R] : Continuous (Algebra.norm R : R ⊗[F] E → R)

theorem **isTopologicalRing_tens** [TopologicalSpace R] [IsTopologicalRing R] : IsTopologicalRing (R ⊗[F] E)

theorem **t2Space_tensorProduct_finiteExtension** [TopologicalSpace R] [IsTopologicalRing R] [T2Space R] : T2Space (R ⊗[F] E)

theorem **val_extUnit** (j : R →ₗ[F] R') (hj : ∀ a b, j (a ∗ b) = j a ∗ j b) (y : (R ⊗[F] E)ˣ) : (extUnit j hj y : R' ⊗[F] E) = 1 − tensNU j hj 1 + tensNU j hj y

theorem **continuous_extUnit** [TopologicalSpace R] [IsTopologicalRing R] [TopologicalSpace R'] [IsTopologicalRing R'] (j : R →ₗ[F] R') (hj : ∀ a b, j (a ∗ b) = j a ∗ j b) (hc : Continuous j) : Continuous (extUnit (E

theorem **absEv_apply** (v : HeightOneSpectrum (𝒪 F)) (x : (Fv F v ⊗[F] E)ˣ) : absEv F E v x = ‖Algebra.norm (Fv F v) x.1‖₊

theorem **absEw_apply** (w : InfinitePlace F) (x : (w.Completion ⊗[F] E)ˣ) : absEw F E w x = ‖Algebra.norm w.Completion x.1‖₊ ^ w.mult

theorem **continuous_absEv** (v : HeightOneSpectrum (𝒪 F)) : Continuous (absEv F E v)

theorem **continuous_absEw** (w : InfinitePlace F) : Continuous (absEw F E w)

theorem **absEv_conjUnit** (σ : E →ₐ[F] E) (v : HeightOneSpectrum (𝒪 F)) (x : (Fv F v ⊗[F] E)ˣ) : absEv F E v (conjUnit σ x) = absEv F E v x

theorem **absEw_conjUnit** (σ : E →ₐ[F] E) (w : InfinitePlace F) (x : (w.Completion ⊗[F] E)ˣ) : absEw F E w (conjUnit σ x) = absEw F E w x

theorem **absEv_algebraMap** (v : HeightOneSpectrum (𝒪 F)) (a : (Fv F v)ˣ) : absEv F E v (Units.map (algebraMap (Fv F v) (Fv F v ⊗[F] E)).toMonoidHom a) = ‖a.1‖₊ ^ Module.finrank F E

theorem **algebraMap_norm_eq** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) {R : Type} [CommRing R] [Algebra F R] (x : R ⊗[F] E) : algebraMap R (R ⊗[F] E) (Algebra.norm R x) = x ∗ Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ x

theorem **absEw_eq_sq** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) {w : InfinitePlace F} (τ : EmbOver E w) (x : (w.Completion ⊗[F] E)ˣ) : absEw F E w x = ‖infEmb τ x.1‖₊ ^ 2

theorem **absEv_unitFin** (v : HeightOneSpectrum (𝒪 F)) (x : (AF F ⊗[F] E)ˣ) : absEv F E v (unitFin F E v x) = ‖evFin F v (Algebra.norm (AF F) x.1)‖₊

theorem **absEw_unitInf** (w : InfinitePlace F) (x : (AF F ⊗[F] E)ˣ) : absEw F E w (unitInf F E w x) = ‖evInf F w (Algebra.norm (AF F) x.1)‖₊ ^ w.mult

theorem **absEv_unitFin_eq_one** {v : HeightOneSpectrum (𝒪 F)} {x : (AF F ⊗[F] E)ˣ} (h : evFin F v (Algebra.norm (AF F) x.1) ∈ v.adicCompletionIntegers F) (h' : evFin F v (Algebra.norm (AF F) x⁻¹.1) ∈ v.adicCompletionIntegers F) : absEv F E v (unitFin F E v x) = 1

theorem **finite_mulSupport_absEv** (x : (AF F ⊗[F] E)ˣ) : (Function.mulSupport fun v => absEv F E v (unitFin F E v x)).Finite

theorem **absE_apply** (x : (AF F ⊗[F] E)ˣ) : absE F E x = (∏ w, absEw F E w (unitInf F E w x)) * ∏ᶠ v, absEv F E v (unitFin F E v x)

theorem **absE_conjUnit** (σ : E →ₐ[F] E) (x : (AF F ⊗[F] E)ˣ) : absE F E (conjUnit σ x) = absE F E x

theorem **absE_inclR** (x : Eˣ) : absE F E (Units.map (inclR (F

theorem **continuous_unitFin** (v : HeightOneSpectrum (𝒪 F)) : Continuous (unitFin F E v)

theorem **continuous_unitInf** (w : InfinitePlace F) : Continuous (unitInf F E w)

theorem **continuous_absE** : Continuous (absE F E)

theorem **adele_ext** {x y : AF F ⊗[F] E} (h1 : ∀ w, compInf F E w x = compInf F E w y) (h2 : ∀ v, compFin F E v x = compFin F E v y) : x = y

theorem **idele_ext** {x y : (AF F ⊗[F] E)ˣ} (h1 : ∀ w, unitInf F E w x = unitInf F E w y) (h2 : ∀ v, unitFin F E v x = unitFin F E v y) : x = y

theorem **inFin_mul** (v : HeightOneSpectrum (𝒪 F)) (a b : Fv F v) : inFin v (a * b) = inFin v a * inFin v b

theorem **inInf_mul** (w : InfinitePlace F) (a b : w.Completion) : inInf w (a * b) = inInf w a * inInf w b

theorem **map_extUnit_self** {R R' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] (φ : R' →ₐ[F] R) (j : R →ₗ[F] R') (hj : ∀ a b, j (a * b) = j a * j b) (h : ∀ a, φ (j a) = a) (y : (R ⊗[F] E)ˣ) : Algebra.TensorProduct.map φ (AlgHom.id F E) (extUnit j hj y).1 = y.1

theorem **map_extUnit_zero** {R R' R'' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] [CommRing R''] [Algebra F R''] (φ : R' →ₐ[F] R'') (j : R →ₗ[F] R') (hj : ∀ a b, j (a * b) = j a * j b) (h : ∀ a, φ (j a) = 0) (y : (R ⊗[F] E)ˣ) : Algebra.TensorProduct.map φ (AlgHom.id F E) (extUnit j hj y).1 = 1

@[simp] theorem **unitFin_ideleAt_self** (v : HeightOneSpectrum (𝒪 F)) (x : (Fv F v ⊗[F] E)ˣ) : unitFin F E v (ideleAt F E v x) = x

theorem **unitFin_ideleAt_ne** {v u : HeightOneSpectrum (𝒪 F)} (h : u ≠ v) (x : (Fv F v ⊗[F] E)ˣ) : unitFin F E u (ideleAt F E v x) = 1

@[simp] theorem **unitInf_ideleAt** (w : InfinitePlace F) (v : HeightOneSpectrum (𝒪 F)) (x : (Fv F v ⊗[F] E)ˣ) : unitInf F E w (ideleAt F E v x) = 1

@[simp] theorem **unitInf_ideleAtInf_self** (w : InfinitePlace F) (x : (w.Completion ⊗[F] E)ˣ) : unitInf F E w (ideleAtInf F E w x) = x

theorem **unitInf_ideleAtInf_ne** {w u : InfinitePlace F} (h : u ≠ w) (x : (w.Completion ⊗[F] E)ˣ) : unitInf F E u (ideleAtInf F E w x) = 1

@[simp] theorem **unitFin_ideleAtInf** (v : HeightOneSpectrum (𝒪 F)) (w : InfinitePlace F) (x : (w.Completion ⊗[F] E)ˣ) : unitFin F E v (ideleAtInf F E w x) = 1

theorem **continuous_ideleAt** (v : HeightOneSpectrum (𝒪 F)) : Continuous (ideleAt F E v)

theorem **absE_ideleAt** (v : HeightOneSpectrum (𝒪 F)) (x : (Fv F v ⊗[F] E)ˣ) : absE F E (ideleAt F E v x) = absEv F E v x

theorem **absE_ideleAtInf** (w : InfinitePlace F) (x : (w.Completion ⊗[F] E)ˣ) : absE F E (ideleAtInf F E w x) = absEw F E w x

Development.AbelianVarietyTheory.Definitions.Foundations.ThNSpec

theorem **thN_spec** (X : Tower Ω ε) (A : ValuationSubring Ω) (N l : ℕ) [Fact l.Prime] (n d : ℕ) (b : X.Fn l) (h : ∃ φ : HahnSeries ℚ Ω, IsThB X A N l n (X.act l (sgD N l d : Matrix (Fin 2) (Fin 2) ℤ) b) φ ∧ IsLat 1 φ) : IsThB X A N l n (X.act l (sgD N l d : Matrix (Fin 2) (Fin 2) ℤ) b) (thN X A N l n d b) ∧ IsLat 1 (thN X A N l n d b)

Development.AbelianVarietyTheory.Definitions.Foundations.ThetaExponentIdentities

theorem **le_maxLift** {D : D7} (g : ℕ → ℝ) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : g j ≤ maxLift D g

theorem **maxLift_le_iff** {D : D7} {g : ℕ → ℝ} {a : ℝ} : maxLift D g ≤ a ↔ ∀ j, 1 ≤ j → j ≤ D.R.m → g j ≤ a

theorem **exists_eq_maxLift** (D : D7) (g : ℕ → ℝ) : ∃ j, 1 ≤ j ∧ j ≤ D.R.m ∧ maxLift D g = g j

theorem **pageOf_L** : (pageOf Q).L = L.sc.L

theorem **pageOf_b** : (pageOf Q).b = L.sc.b

theorem **pageOf_lrd** : (pageOf Q).lrd = lrd D

theorem **pageOf_Lc** : (pageOf Q).Lc = L.logDerivativeAtOne

theorem **log_derivative_at_zero** : (pageOf Q).L0 = dlogL D 0

theorem **pageOf_LL0_eq_placeRankTwoOne_LL0** : (pageOf Q).LL0 = placeRankTwoOne.LL0 D L

theorem **ppTsq_nonneg** : 0 ≤ ppTsq Q

theorem **pageOf_PPT_sq** : (pageOf Q).PPT ^ 2 = ppTsq Q

theorem **T_eq** {K : Type} [Field K] [NumberField K] [StarRing K] (F : SlopeBoundLiftData K (pageOf Q)) (h : F.DY = dY D) : F.T = tR Q

theorem **PPp_le** {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : (numAt Q j).PPp ≤ (pageOf Q).PPp

theorem **PPpp_le** {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : (numAt Q j).PPpp ≤ (pageOf Q).PPpp

theorem **leafDiff_pageOf** : LeafDiff (pageOf Q)

theorem **kapMinus_add_twW** (j : ℕ) : kapMinus D j + twW D j = kap D j

theorem **kapMinus_eq** (j : ℕ) (τ : D.what₂.Tau) (i : ℕ) : kapMinus D j τ i = uMinus D j τ i + wAn D j τ

theorem **remW_anchor** (j : ℕ) (τ : D.what₂.Tau) : remW Q j τ (D.R.k − 2) = 0

theorem **uOne_gap** (j : ℕ) (τ : D.what₂.Tau) (i : ℕ) : uOne Q j τ i − uOne Q j τ (i + 1) = gapOne Q j τ i

theorem **remW_gap** (j : ℕ) (τ : D.what₂.Tau) (i : ℕ) : 0 ≤ remW Q j τ i − remW Q j τ (i + 1) ∧ remW Q j τ i − remW Q j τ (i + 1) ≤ tN Q + placeRankTwoOne.dimY D

theorem **uTop_eq** (j : ℕ) (τ : D.what₂.Tau) (i : ℕ) : uTop Q j τ i = uMinus D j τ i − aW D j τ (D.R.k − 2)

theorem **aW_anchor** {j : ℕ} (h1 : 1 ≤ j) (τ : D.what₂.Tau) : aW D j τ (D.R.k − 2) = −canW D j (D.R.k − 2)

theorem **toWt_apply** (v : D.what₂.Tau → ℕ → ℤ) {τ : Psi H} {t : D.what₂.Tau} (h : τ.1 = DV.tauEmb D t) (i : Fin (Dw.p + Dw.q)) : toWt Dw v τ i = v t i

theorem **toWt_add** (v v' : D.what₂.Tau → ℕ → ℤ) : toWt Dw (v + v') = toWt Dw v + toWt Dw v'

theorem **toWt_smul** (n : ℤ) (v : D.what₂.Tau → ℕ → ℤ) : toWt Dw (n • v) = n • toWt Dw v

theorem **psiOf_bijective** {T : Par D} (h : T.Pins) : Function.Bijective (psiOf h.1)

theorem **toWt_psiOf** {T : Par D} (h : T.PinHost) {Dw : Datum T.H} (v : D.what₂.Tau → ℕ → ℤ) (t : D.what₂.Tau) (i : Fin (Dw.p + Dw.q)) : toWt Dw v (psiOf h t) i = v t i

theorem **sig_eq** (h : T.Pins) (hj : InR D j) : (T.Dw j).p = pp D j ∧ (T.Dw j).q = qq D j ∧ (T.Dw j).p + (T.Dw j).q = D.R.r

theorem **dimFlag_eq** (h : T.Pins) (hj : InR D j) : dimFlag (T.Dw j) = placeRankTwoOne.dimY D

theorem **toWt_deW** (h : T.Pins) (hj : InR D j) : toWt (T.Dw j) (fun _ => deW D j) = hodgeWt (T.Dw j)

theorem **toWt_uPr** (h : T.Pins) (hj : InR D j) : toWt (T.Dw j) (uPr Q j) = uPrime (T.Dw j) (toWt (T.Dw j) (uOne Q j))

theorem **vdm_toWt** (h : T.Pins) (hj : InR D j) (v : D.what₂.Tau → ℕ → ℤ) : ((vdm (toWt (T.Dw j) v) : ℤ) : ℝ) = GlobalTheory.IndefiniteRowSideFullHecke.DescentConstants.degW D v

theorem **rhoB_toWt** (h : T.Pins) (hj : InR D j) : GlobalTheory.CompHt.rhoB (T.Dw j) (toWt (T.Dw j) (uOne Q j)) = GlobalTheory.IndefiniteRowSideFullHecke.DescentConstants.rhoB Q j

theorem **thetaExp_add** (v v' : D.what₂.Tau → ℕ → ℤ) : thetaExp D Dw (v + v') = thetaExp D Dw v + plainExp D Dw v'

theorem **thetaExp_kap** (j : ℕ) : thetaExp D Dw (kapMinus D j) + plainExp D Dw (twW D j) = thetaExp D Dw (kap D j)

theorem **thetaExp_kap_eq** (h : T.Pins) (hj : InR D j) (w : WeightFamilies.Wt D j) (hn : w.n = D.R.nV) (hw : Formalism.L2.Geo.wtOf w = DV.kapW D j) : thetaExp D (T.Dw j) (kap D j) = wtZ (wtA w) w.n

theorem **thetaExp_kap_kapTop** (h : T.Pins) (hK : DV.KapDef D) (i : Fin D.R.m) : thetaExp D (T.Dw ((i : ℕ) + 1)) (kap D ((i : ℕ) + 1)) = wtZ (wtA (Formalism.IndefR.FWT.kapTop D i)) (Formalism.IndefR.FWT.kapTop D i).n

Development.AbelianVarietyTheory.Definitions.Foundations.ThetaSeriesTies

theorem **inR_succ** (i : Fin D.R.m) : InR D ((i : ℕ) + 1)

theorem **liftBaseChange_bijective** {F K C V : Type} [Field F] [Field K] [Algebra F K] [AddCommGroup C] [Module F C] [AddCommGroup V] [Module F V] [Module K V] [IsScalarTower F K V] (f : C →ₗ[F] V) (hf : Function.Injective f) {n : ℕ} (b : Fin n → V) (hF : LinearIndependent F b) (hr : Submodule.span F (Set.range b) = LinearMap.range f) (hK : LinearIndependent K b) (hs : Submodule.span K (Set.range b) = ⊤) : Function.Bijective (f.liftBaseChange K)

theorem **cusp_liftBaseChange_bijective** (B : WeightFamilies.Prov D) {j : ℕ} {w : WeightFamilies.Wt D j} {K : Type} [Field K] [Algebra B.KO K] {m : ℕ} (f : B.Cusp j w →ₗ[B.KO] (Fin m → K)) (hf : Function.Injective f) {n : ℕ} (b : Fin n → Fin m → K) (hF : LinearIndependent B.KO b) (hr : Submodule.span B.KO (Set.range b) = LinearMap.range f) (hK : LinearIndependent K b) (hs : Submodule.span K (Set.range b) = ⊤) : Function.Bijective (f.liftBaseChange K)

theorem **traceSpan_le** {B : WeightFamilies.Prov D} {j : ℕ} {w : WeightFamilies.Wt D j} {K : Type} [Field K] [NumberField K] [Algebra B.KO K] {n : ℕ} (z : (Fin n → K) ≃ₗ[K] K ⊗[B.KO] B.Cusp j w) {P : Type} [Group P] [Finite P] (U : Subgroup P) (ψ : P → 𝒪 K) {s : Set (U → B.Cusp j w)} {t : Set (B.Cusp j w)} (hst : ∀ f ∈ s, ∀ u, f u ∈ t) : traceSpan B z U ψ s ≤ spanK z t

theorem **dvd_pair_of_mem** (x : Dw.La) {q : Dw.La} (hq : q ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) : (l.M : ℤ) ∣ Dw.EF x (cSep l • q)

theorem **IsRightMul.mul** {γ δ : LaMod Dw l ≃ₗ[𝒪 E] LaMod Dw l} {n : ℕ} {x x' x'' : Fin n → Dw.La} (h : IsRightMul γ x x') (h' : IsRightMul δ x' x'') : IsRightMul (γ ∗ δ) x x''

theorem **IsOrbitFamily.mem_shortGen** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {bad : ℕ} {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {d : DeltaDat Xo c n} {I : Type} {ℓ : I → (LaMod Dw l ≃ₗ[𝒪 E] LaMod Dw l)} {f : I → Rul Xo c} (hf : IsOrbitFamily Xo c bad a d ℓ f) (nab : NabV Xo c) (u : I) (hu : f u ∈ sec Xo c a n) : f u ∈ shortGen Xo c nab bad n a n d

theorem **orthProj_mem** {K : Type} [Field K] [StarRing K] {n : ℕ} (Hm : Matrix (Fin n) (Fin n) K) (Z : Submodule K (Fin n → K)) (x : Fin n → K) : orthProj Hm Z x ∈ Z

theorem **sub_orthProj_mem_perpOf** {K : Type} [Field K] [StarRing K] {n : ℕ} (Hm : Matrix (Fin n) (Fin n) K) (Z : Submodule K (Fin n → K)) (hd : IsUnit ((subE Z)ᴴ ∗ Hm ∗ subE Z).det) (x : Fin n → K) : x − orthProj Hm Z x ∈ perpOf Hm Z

theorem **IsClassSum.unique** {t : T0} {U : Subgroup G} {ψ : U →* D.what₂.LLˣ} {ph : ℂ} {θ θ' : (Fin (srcDim A i t U ψ) → D.what₂.LL) →ₗ[D.what₂.LL] (Fin (R.nW i) → D.what₂.LL)} (hinj : Function.Injective (adForm R i)) (h : IsClassSum R i t U ψ ph θ) (h' : IsClassSum R i t U ψ ph θ') : θ = θ'

theorem **cutTo_spec** {J : CoreJ D} [Algebra J.Q.c.F D.what₂.LL] {R : RealDat D A J.Q.c.F} {v₀ : ∀ (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), (R.MW i σ).Vec} (fw : FWTie J R κ v₀) (i : Fin D.R.m) : Function.Injective (cutTo fw i) ∧ LinearMap.range (cutTo fw i) = (R.FW i).WK

theorem **toForms_tmul** (h : LawsP T.toTop) (fw : FWTie T.J R κ v₀) (i : Fin D.R.m) {n : ℕ} (z : (Fin n → D.what₂.LL) ≃ₗ[D.what₂.LL] D.what₂.LL ⊗[(recB h.toLawsCore).KO] (recB h.toLawsCore).Cusp ((i : ℕ) + 1) (κ i)) (a : D.what₂.LL) (F : (recB h.toLawsCore).Cusp ((i : ℕ) + 1) (κ i)) : toForms h fw i z (z.symm (a ⊗ₜ F)) = a • cuspTo h fw i F

theorem **toForms_bijective** (h : LawsP T.toTop) (fw : FWTie T.J R κ v₀) (i : Fin D.R.m) {n : ℕ} (z : (Fin n → D.what₂.LL) ≃ₗ[D.what₂.LL] D.what₂.LL ⊗[(recB h.toLawsCore).KO] (recB h.toLawsCore).Cusp ((i : ℕ) + 1) (κ i)) : Function.Bijective (toForms h fw i z)

theorem **OnTie.Th0_le_ThK** (t : OnTie h fw hinj EX p ι wL wF nT O v i z act inc ℓ U ψ ph y) (hn : (κ ι).n = D.R.nV) : O.Th0 v i ≤ O.ThK v i

theorem **OnTie.e_eOp** (t : OnTie h fw hinj EX p ι wL wF nT O v i z act inc ℓ U ψ ph y) (x : Fin (O.nZ i) → D.what₂.LL) : toForms h fw ι z (O.e i x) = (R.FW ι).eOp (fun g => (idemOf A.X).coeff g) (toForms h fw ι z x)

theorem **OnTie.adForm_0'** (t : OnTie h fw hinj EX p ι wL wF nT O v i z act inc ℓ U ψ ph y) (hr : ClassSumRational R ι (R.mu v) U ψ ph) (x : Fin (O.nX v i) → D.what₂.LL) : adForm R ι (toForms h fw ι z (O.θ' v i x)) = ph • R.thF ι (R.mu v) (jVx R ι (R.mu v) (srcForm A ι (R.mu v) U ψ (y x)))

theorem **OnTie.rho_eq_of_actionPin** (t : OnTie h fw hinj EX p ι wL wF nT O v i z act inc ℓ U ψ ph y) (ha : ActionPin h fw ι ℓ) (g : P) (x : Fin (O.nZ i) → D.what₂.LL) : toForms h fw ι z (O.ρ i g x) = (R.FW ι).ρ (inc g) (toForms h fw ι z x)

theorem **charInt_coe** {K : Type} [Field K] [NumberField K] {P : Type} [Group P] (inc : P →* G) (U : Subgroup G) (ψ : U →* Kˣ) (g : P) (hg : inc g ∈ U) : ((charInt inc U ψ g : 𝒪 K) : K) = (ψ ⟨inc g, hg⟩ : Kˣ)

theorem **exists_onTie** (h : LawsP T.toTop) (fw : FWTie T.J R κ v₀) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) {Bs : DefBase D.what₂.LL} {Df : DefSide Bs} {S : IndefData Df} {P : Type} [Group P] [Finite P] {oc : OnConst} (O : OnObj S P oc) (ι : Bs.J → Fin D.R.m) (p : HeightOneSpectrum (𝒪 D.what₂.LL) → ℕ) (wL wF : ∀ i, WeightFamilies.Wt D ((ι i : ℕ) + 1)) (nT : ℕ) (inc : P →* G) (ℓ : ∀ i, G → (LaMod (T.Q.Dw ((ι i : ℕ) + 1)) T.Q.l ≃ₗ[𝒪 D.host.E] LaMod (T.Q.Dw ((ι i : ℕ) + 1)) T.Q.l)) (ρ : ∀ i, Representation (recB h.toLawsCore).KO P ((recB h.toLawsCore).Cusp ((ι i : ℕ) + 1) (κ (ι i)))) (hρ : ∀ i, IsLevAct T.toTop h ((ι i : ℕ) + 1) (κ (ι i)) (fun g => ℓ i (inc g)) fun g F => ρ i g F) (U : Subgroup G) (ψ : U →* D.what₂.LLˣ) (ph : Bs.J → ℂ) : ∃ (O' : OnObj S P oc) (z : ∀ i, (Fin (O'.nZ i) → D.what₂.LL) ≃ₗ[D.what₂.LL] D.what₂.LL ⊗[(recB h.toLawsCore).KO] (recB h.toLawsCore).Cusp ((ι

Development.AbelianVarietyTheory.Definitions.Foundations.ToContinuousLinearMapMemHomIn

theorem **hTensor_coe** (g : G) (a : H₁ ⊗[ℂ] H₂) : hTensor p₁ p₂ π₁ π₂ g a = TensorProduct.mapL (π₁ (p₁ g)) (π₂ (p₂ g)) a

theorem **IsTensorMap.apply** (h : IsTensorMap p₁ p₂ π₁ π₂ π J) (g : G) (a : H₁ ⊗[ℂ] H₂) : π g (J a) = J (TensorProduct.mapL (π₁ (p₁ g)) (π₂ (p₂ g)) a)

theorem **isUnitaryRep_of_dense** [TopologicalSpace G] (h1 : ∀ g v, ‖π g v‖ = ‖v‖) {A : Set H} (hA : (Submodule.span ℂ A).topologicalClosure = ⊤) (h2 : ∀ v ∈ A, Continuous fun g => π g v) : IsUnitaryRep π

theorem **IsTensorMap.dense_span** (h : IsTensorMap p₁ p₂ π₁ π₂ π J) : (Submodule.span ℂ {v | ∃ x y, v = J (x ⊗ₜ y)}).topologicalClosure = ⊤

theorem **IsTensorMap.isUnitaryRep** [TopologicalSpace G] [TopologicalSpace G₁] [TopologicalSpace G₂] (h : IsTensorMap p₁ p₂ π₁ π₂ π J) (hp₁ : Continuous p₁) (hp₂ : Continuous p₂) (h₁ : IsUnitaryRep π₁) (h₂ : IsUnitaryRep π₂) : IsUnitaryRep π

theorem **IsTensorMap.uEquiv** [CompleteSpace H] (h : IsTensorMap p₁ p₂ π₁ π₂ π J) : UEquiv (hTensor p₁ p₂ π₁ π₂) π

theorem **restr_map** {M : ι → Type} {ξ y : ∀ i, M i} {f : ∀ i, M i → M i} (hf : ∀ᶠ i in cofinite, f i (ξ i) = ξ i) (hy : Restr ξ y) : Restr ξ fun i => f i (y i)

theorem **restr_self** {M : ι → Type} (ξ : ∀ i, M i) : Restr ξ ξ

theorem **IsRestrTensor.finite_mulSupport** (h : IsRestrTensor p δ ξ π B) (hy : Restr ξ y) (hy' : Restr ξ y') : (Function.mulSupport fun i => ⟪y' i, y i⟫_ℂ).Finite

theorem **IsRestrTensor.inner_apply** (h : IsRestrTensor p δ ξ π B) (hy : Restr ξ y) (hy' : Restr ξ y') (g : G) : ⟪B y', π g (B y)⟫_ℂ = ∏ᶠ i, ⟪y' i, δ i (p i g) (y i)⟫_ℂ

theorem **IsRestrTensor.ext** (h : IsRestrTensor p δ ξ π B) {v w : H} (e : ∀ y, Restr ξ y → ⟪B y, v⟫_ℂ = ⟪B y, w⟫_ℂ) : v = w

theorem **IsRestrTensor.inner_update** [DecidableEq ι] (h : IsRestrTensor p δ ξ π B) (hy : Restr ξ y) (hy' : Restr ξ y') (i : ι) (x : Hs i) : ⟪B y', B (Function.update y i x)⟫_ℂ = ⟪y' i, x⟫_ℂ * ∏ᶠ (j) (_ : j ≠ i), ⟪y' j, y j⟫_ℂ

theorem **IsRestrTensor.pure_update** [DecidableEq ι] (h : IsRestrTensor p δ ξ π B) (hy : Restr ξ y) (i : ι) (c : ℂ) (a b : Hs i) : B (Function.update y i (c • a + b)) = c • B (Function.update y i a) + B (Function.update y i b)

theorem **pureT_ne_zero** {s y : ∀ i, N i} (hy : Restr s y) (h0 : ∀ i, y i ≠ 0) : pureT s y ≠ 0

theorem **mem_homTensor** : T ∈ homTensor j₁ j₂ σ δ ρ U ↔ (∀ w y, T w y ∈ U) ∧ (∀ g w y, T (σ g w) y = ρ (j₁ g) (T w y)) ∧ ∀ g w y, T w (δ g y) = ρ (j₂ g) (T w y)

theorem **injective_of_mem_homInto** {X : Type} [AddCommGroup X] [Module ℂ X] {τ : Representation ℂ G₁ X} {C : Submodule ℂ X} {f : W →ₗ[ℂ] X} (hσ : σ.IsIrreducible) (hf : f ∈ homInto σ τ C) (h0 : f ≠ 0) : Function.Injective f

theorem **rank_homInto_congr** {W' X : Type} [AddCommGroup W'] [Module ℂ W'] [AddCommGroup X] [Module ℂ X] {σ' : Representation ℂ G₁ W'} (e : σ.Equiv σ') (τ : Representation ℂ G₁ X) (C : Submodule ℂ X) : Module.rank ℂ (homInto σ τ C) = Module.rank ℂ (homInto σ' τ C)

theorem **mem_homPi** : Φ ∈ homPi e δ ρ U ↔ (∀ y, Φ y ∈ U) ∧ ∀ i g y, Φ (Function.update y i (δ i g (y i))) = ρ (e i g) (Φ y)

theorem **mem_homTensorPi** : T ∈ homTensorPi j e σ δ ρ U ↔ (∀ w y, T w y ∈ U) ∧ (∀ g w y, T (σ g w) y = ρ (j g) (T w y)) ∧ ∀ i g w y, T w (Function.update y i (δ i g (y i))) = ρ (e i g) (T w y)

theorem **homTensorPi_mono** (h : U ≤ U') : homTensorPi j e σ δ ρ U ≤ homTensorPi j e σ δ ρ U'

theorem **multTensorPi_congr** {W' : Type} [AddCommGroup W'] [Module ℂ W'] {σ' : Representation ℂ G₁ W'} (f : σ.Equiv σ') : multTensorPi j e σ δ ρ U = multTensorPi j e σ' δ ρ U

theorem **toContinuousLinearMap_mem_homIn** (h : Φ ∈ homPi e δ ρ U) (y : ∀ i, Hs i) (i : ι) : Φ.toContinuousLinearMap y i ∈ homIn (δ i) (ρ.comp (e i)) U

theorem **inner_eq_bePi** (hδ : ∀ i, IsIrrUnitary (δ i)) (hρ : IsUnitaryRep ρ) (h : Φ ∈ homPi e δ ρ U) (h' : Φ' ∈ homPi e δ ρ U') (y y' : ∀ i, Hs i) : ⟪Φ' y', Φ y⟫_ℂ = bePi Φ Φ' * ∏ i, ⟪y' i, y i⟫_ℂ

theorem **bePi_self_pos** (hδ : ∀ i, IsIrrUnitary (δ i)) (hρ : IsUnitaryRep ρ) (h : Φ ∈ homPi e δ ρ U) (h0 : Φ ≠ 0) : 0 < (bePi Φ Φ).re

theorem **isRestrTensor_charRep** {ι G : Type} {Γ : ι → Type} [Group G] [∀ i, Group (Γ i)] (p : ∀ i, G →* Γ i) (χ : ∀ i, Γ i →* Circle) (ψ : G →* Circle) (h : ∀ g, ∀ᶠ i in cofinite, χ i (p i g) = 1) (hψ : ∀ g, (ψ g : ℂ) = ∏ᶠ i, (χ i (p i g) : ℂ)) : IsRestrTensor p (fun i => charRep (χ i)) (fun _ => 1) (charRep ψ) fun y => ∏ᶠ i, y i where

Development.AbelianVarietyTheory.Definitions.Foundations.TracePowApply

theorem **_root_.GlobalTheory.Mem.GelfCheck.rowE_apply** (R : Type) [CommRing R] (r n : ℕ) (a b : Fin r) (P : MvPolynomial (Fin r × Fin n) R) : _root_.GlobalTheory.Mem.Gelf.rowE R r n a b P = ∑ s : Fin n, X (a, s) * pderiv (b, s) P

theorem **_root_.GlobalTheory.Mem.GelfCheck.rowE_X** (R : Type) [CommRing R] (r n : ℕ) (a b c : Fin r) (t : Fin n) : _root_.GlobalTheory.Mem.Gelf.rowE R r n a b (X (c, t)) = if b = c then X (a, t) else 0

theorem **_root_.GlobalTheory.Mem.GelfCheck.rowE_mul** (R : Type) [CommRing R] (r n : ℕ) (a b : Fin r) (P Q : MvPolynomial (Fin r × Fin n) R) : _root_.GlobalTheory.Mem.Gelf.rowE R r n a b (P * Q) = _root_.GlobalTheory.Mem.Gelf.rowE R r n a b P * Q + P * _root_.GlobalTheory.Mem.Gelf.rowE R r n a b Q

theorem **_root_.GlobalTheory.Mem.GelfCheck.rowE_C** (R : Type) [CommRing R] (r n : ℕ) (a b : Fin r) (x : R) : _root_.GlobalTheory.Mem.Gelf.rowE R r n a b (C x) = 0

theorem **_root_.GlobalTheory.Mem.GelfCheck.glRel_rowE** (R : Type) [CommRing R] (r n : ℕ) : _root_.GlobalTheory.Mem.Gelf.GlRel (_root_.GlobalTheory.Mem.Gelf.rowE R r n)

private theorem **gradedWeyl_gen_coeff_one_and_coeff_zero** (R : Type) [CommRing R] (r n : ℕ) (a b : Fin r) (P : MvPolynomial (Fin r × Fin n) R) : (_root_.Formalism.GradedWeylDivisibility.rowGen R a b P).coeff 1 = ∑ s : Fin n, X (a, s) * pderiv (b, s) P ∧ (_root_.Formalism.GradedWeylDivisibility.rowGen R a b P).coeff 0 = P

theorem **_root_.GlobalTheory.Mem.GelfCheck.rowE_eq_rowDiv** (R : Type) [CommRing R] (r n : ℕ) (a b : Fin r) (P : MvPolynomial (Fin r × Fin n) R) : _root_.GlobalTheory.Mem.Gelf.rowE R r n a b P = _root_.Formalism.GradedWeylDivisibility.rowDiv R a b 1 P ∧

root.Formalism.GradedWeylDivisibility.rowDiv R a b 0 P = P

theorem **glRel_iff** {A : Type} [Ring A] {N : ℕ} (E : Matrix (Fin N) (Fin N) A) : GlRel E ↔ Formalism.REAL.GlRel (fun a b => E a b)

theorem **_root_.GlobalTheory.Mem.GelfCheck.gelfA_eq_Amat** {N : ℕ} (μ : Fin N → ℂ) : _root_.GlobalTheory.Mem.Gelf.gelfA μ = _root_.Formalism.REAL.Amat (_root_.Formalism.REAL.lsh μ)

theorem **_root_.GlobalTheory.Mem.GelfCheck.pow_cov** (h : _root_.GlobalTheory.Mem.Gelf.GlRel E) (k : ℕ) (a b c d : Fin N) : E c d * (E ^ k) a b - (E ^ k) a b * E c d = (if d = a then (E ^ k) c b else 0) - (if b = c then (E ^ k) a d else 0)

theorem **upper_kill** (h : GlRel E) (hv : ∀ a b, a < b → E a b v = 0) (k : ℕ) (a b : Fin N) (hab : a < b) : (E ^ k) a b v = 0

theorem **diag_pow** (h : GlRel E) (hw : ∀ a, E a a v = μ a • v) (hv : ∀ a b, a < b → E a b v = 0) (k : ℕ) (a : Fin N) : (E ^ k) a a v = (∑ b, (gelfA μ ^ k) a b) • v

theorem **_root_.GlobalTheory.Mem.GelfCheck.trace_pow_apply** (h : _root_.GlobalTheory.Mem.Gelf.GlRel E) (hw : ∀ a, E a a v = μ a • v) (hv : ∀ a b, a < b → E a b v = 0) (k : ℕ) : (E ^ k).trace v = _root_.GlobalTheory.Mem.Gelf.gelfC μ k • v

theorem **trace_pow_X** (R : Type) [CommRing R] (r n : ℕ) (t : Fin n) (k : ℕ) : (rowE R (r+1) n ^ k).trace (X (0, t)) = gelfC (Pi.single (0 : Fin (r+1)) (1 : R)) k • X (0, t)

Development.AbelianVarietyTheory.Definitions.Foundations.TrueColumnChoicesHeckeCharReferenceEqOne

theorem **HChar.ValuesIn.ofLL0** {D : D7} {χ : HChar D.host.E} {s : ℝ} (h : χ.ValuesIn s (embLL0 D)) : χ.ValuesIn s D.what₂.embLL

theorem **ValLL0.toLL** {D : D7} {C : CharFrame D} (h : ValLL0 D C) : ValLL D C

theorem **TupTied.psi_gen** {D : D7} {C : CharFrame D} {nch : NChoice D} (Tp : TupTied D C nch) (w : D.what₂.places.Pl) : ∃ Q, IsPsi (D.what₂.places.resChar w) Q (Tp.psi w)

theorem **ThetaChrTied.carY** {D : D7} {C : CharFrame D} {nch : NChoice D} {Tp : TupTied D C nch} {X : ThetaX D} (T : ThetaChrTied D C nch Tp X) (μ : D.what₂.Tup) (j : Fin D.R.m) (i : ℕ) : T.eChr (X.carY μ (j.1 + 1) i) = if Even i then Tp.tup μ j * C.eta⁻¹ else Tp.tup μ j

Development.AbelianVarietyTheory.Definitions.Foundations.TupleConstantsNeZero

theorem **IsAwayUnit.of_dvd** {m k : ℕ} {a : 𝒪 K} (h : a ∣ (m : 𝒪 K) ^ k) : IsAwayUnit m (a : K)

theorem **IsAwayUnit.one** (m : ℕ) : IsAwayUnit m (1 : K)

theorem **IsAwayUnit.mul** {m : ℕ} {x x' : K} (h : IsAwayUnit m x) (h' : IsAwayUnit m x') : IsAwayUnit m (x * x')

theorem **vw_eq_zero_of_notMem** (w : Pl K) {a : 𝒪 K} (ha : a ∉ w.asIdeal) : vw w (a : K) = 0

theorem **IsAwayUnit.ne_zero_and_vw** {m : ℕ} {x : K} (h : IsAwayUnit m x) (w : Pl K) (hm : (m : 𝒪 K) ∉ w.asIdeal) : x ≠ 0 ∧ vw w x = 0

theorem **normAlg_mul** {κ X : Type} [Fintype κ] [Fintype X] (G : κ → κ → K) (gam : X → ℕ) (z : X → κ → K) (t : K) : normAlg G gam (fun x k => t * z x k) = t * star t * normAlg G gam z

theorem **Sq.Tuple.tuple_constants_ne_zero** {S : Sq K} {hd y : ℕ} {σ₀ : K →+* ℂ} {a : An (S.p + S.q)} {Z : Co K S.κ} {w : Pl K} {dl : ℝ} (h : S.Tuple σ₀ a Z w dl) (hS : S.OK hd y) : a.c0 ≠ 0 ∧ a.c ≠ 0

theorem **Sq.Twisted.ne_zero** {S : Sq K} {hd y : ℕ} {σ₀ : K →+* ℂ} {w : Pl K} {cw : ℂ} {zA zA' : K} {zT zTd : ℤ → K} {gT : ℤ → ℤ} {n0 : ℤ → ℝ} {er dl : ℝ} {T : Type} {S0 : Finset T} {cc qq : T → ℕ} (h : S.Twisted σ₀ w cw zA zA' zT zTd gT n0 er dl S0 cc qq) (hS : S.OK hd y) : cw ≠ 0

theorem **vw_natCast_nonneg** (w : Pl K) (n : ℕ) : (0 : ℝ) ≤ (vw w (n : K) : ℝ)

theorem **square_ok** : (square K).OK 3 2

theorem **normAlg_coords** (t : ℕ) : normAlg (square K).G (coords K t).gam (coords K t).z = (t : K) ^ 2

theorem **normAlg_coords'** (t : ℕ) : normAlg (square K).G (coords K t).gam (coords K t).z' = 1

theorem **tuple** (σ₀ : K →+* ℂ) (w : Pl K) {n m Q : ℕ} (hn : n ≠ 0) (hm : m ≠ 0) (hQ : Q ≠ 0) : (square K).Tuple σ₀ (numbers n m Q) (coords K (m * Q)) w (vw w ((2 : ℕ) : K) : ℝ)

theorem **numbers_one** : numbers 1 1 1 = (square K).base

theorem **twisted** (σ₀ : K →+* ℂ) (w : Pl K) {n m d Q : ℕ} (hn : n ≠ 0) (hm : m ≠ 0) (hd : d ≠ 0) (hQ : (Q : 𝒪 K) ∉ w.asIdeal) {er : ℝ} (her : (vw w (m : K) : ℝ) ≤ (vw w (d : K) : ℝ) + er) : (square K).Twisted σ₀ w (σ₀ ((n ^ 2 * m * Q : ℕ) : K) * (square K).base.c) 1 1 (fun _ => 1) (fun _ => d) (fun _ => 0) (fun _ => 0) er (vw w ((2 : ℕ) : K) : ℝ) (univ : Finset Unit) (fun _ => 1) (fun _ => n)

theorem **exists_twisted_ne_base** (σ₀ : K →+* ℂ) (w : Pl K) : ∃ (S : Sq K) (cw : ℂ) (dl : ℝ), S.OK 3 2 ∧ 0 ≤ dl ∧ cw = 12 * S.base.c ∧ cw ≠ S.base.c ∧ S.Twisted σ₀ w cw 1 1 (fun _ => 1) (fun _ => 3) (fun _ => 0) (fun _ => 0) 0 dl (univ : Finset Unit) (fun _ => 1) (fun _ => 2)

Development.AbelianVarietyTheory.Definitions.Foundations.UnitarySixTwoHarishChandraParameter

theorem **unitary_six_two_grid_counts** : grid4.NearMid ∧ (∀ i ∈ range 5, ∀ j ∈ range 5, IsRegular (grid4.parD i j) ∧ (grid4.parD i j).card = grid4.rankD i j ∧ nMinus (grid4.parD i j) = (4 - j) / 2) ∧ (∀ i ∈ range 5, IsRegular (grid4.parPi i) ∧ (grid4.parPi i).card = grid4.rankP i ∧ nMinus (grid4.parPi i) = (4 - i + 1) / 2) ∧ (∀ j ∈ range 5, (grid4.weightX j).card = grid4.rankX j)

theorem **unitary_six_two_lift_arrows** : (∀ i ∈ range 4, ∀ j ∈ range 5, grid4.parD (i + 1) j = liftMinus (grid4.cen (4 - i)) 3 (grid4.parD i j) ∧ FarFrom (grid4.cen (4 - i)) 2 (grid4.parD i j)) ∧ (∀ i ∈ range 5, ∀ j ∈ range 4, grid4.parD i (j + 1) = liftPlus (-grid4.e (j + 1)) 3 (grid4.parD i j) ∧ FarFrom (-grid4.e (j + 1)) 3 (grid4.parD i j)) ∧ (∀ i ∈ range 4, grid4.parPi (i + 1) = liftMinus (grid4.cen (4 - i)) 2 (grid4.parPi i) ∧ grid4.parPi (i + 1) ≠ liftPlus (grid4.cen (4 - i)) 2 (grid4.parPi i)) ∧ (∀ j ∈ range 4, grid4.parX (j + 1) = liftPlus (-grid4.e (j + 1)) 4 (grid4.parX j))

theorem **unitary_six_two_interlacing** : ncGap grid4.lam = ((9 : ℚ) : WithTop ℚ) ∧ signature grid4.lam = (6, 2) ∧ signature grid4.parD0 = (5, 2) ∧ HeInterlace grid4.lam grid4.parD0 ∧ ¬ HeInterlace grid4.lam (flip grid4.parD0) ∧ ¬ HeInterlace grid4.lam (grid4.parD 0 1) ∧ ncGap (grid4.parD 2 3) = ⊤ ∧ StrictInterlace (grid4.weightX 4) (entries (grid4.parD 4 4)) ∧ ¬ StrictInterlace (entries (grid4.parD 4 4)) (grid4.weightX 4)

theorem **he_one_one** : ∀ z ∈ Icc (-6 : ℤ) 6, (HeInterlace {(5 / 2, true), (-3 / 2, false)} {((z : ℚ), false)} ↔ z ≤ -2) ∧ (HeInterlace {(5 / 2, false), (-3 / 2, true)} {((z : ℚ), false)} ↔ 3 ≤ z)

theorem **he_two** : ∀ z ∈ Icc (-6 : ℤ) 6, (HeInterlace {(5 / 2, true), (-3 / 2, true)} {((z : ℚ), true)} ↔ -1 ≤ z ∧ z ≤ 2) ∧ (StrictInterlace {5 / 2, -3 / 2} {(z : ℚ)} ↔ -1 ≤ z ∧ z ≤ 2)

theorem **degenerate** : ¬ StrictInterlace ∅ ∅ ∧ StrictInterlace {0} ∅ ∧ ncGap ∅ = ⊤ ∧ ncGap {(0, true), (0, false)} = ((0 : ℚ) : WithTop ℚ) ∧ ¬ HeInterlace {(0, true), (0, false)} ∅ ∧ ¬ HeInterlace {(1, true), (0, true)} {(0, true)} ∧ ¬ HeInterlace {(1, true), (0, true)}

⌀

theorem **nMinus_parD** (G : GridData) (i j : ℕ) : nMinus (G.parD i j) = (G.m − j) / 2

theorem **unitary_six_two_harish_chandra_parameter** : GoodPar 8 6 2 grid4.lam

Development.AbelianVarietyTheory.Definitions.Foundations.ValuedAlgebraMapEqOne

theorem **valued_algebraMap_eq_one** (v : HeightOneSpectrum (𝒪 L)) (x : 𝒪 L) (hx : x ∉ v.asIdeal) : Valued.v (algebraMap L (v.adicCompletion L) (algebraMap (𝒪 L) L x)) = 1

Development.AbelianVarietyTheory.Definitions.Foundations.WittIndexEqZeroIff

theorem **mem_perp** {W : Submodule E V.V} {y : V.V} : y ∈ perp V W ↔ ∀ x ∈ W, V.hD y x = 0

theorem **perp_anti** {A B : Submodule E V.V} (h : A ≤ B) : perp V B ≤ perp V A

@[simp] theorem **perp_bot** : perp V ⊥ = ⊤

@[simp] theorem **perp_top** : perp V ⊤ = ⊥

theorem **le_perp_of_le** {A B : Submodule E V.V} (h : A ≤ B) (hB : B ≤ perp V B) : A ≤ perp V A

theorem **coordR_mem** {W : Submodule E V.V} {x : V.V} (hx : x ∈ W) : coordR (F

theorem **extSub_mono** {A B : Submodule E V.V} (h : A ≤ B) : extSub (F

theorem **mulVec_mem_extSub** {W : Submodule E V.V} {r : ℕ} (M : Matrix (Fin V.rank) (Fin r) E) (h : ∀ k, vecOf V (Mᵀ k) ∈ W) (a : Fin r → R ⊗[F] E) : M.map (inclR (F

theorem **map_mulVec_coordR** {r : ℕ} (M : Matrix (Fin r) (Fin V.rank) E) (x : V.V) : M.map (inclR (F

theorem **mem_extSub_iff_of_ker** {W : Submodule E V.V} {r : ℕ} (T : Matrix (Fin r) (Fin V.rank) E) (P : Matrix (Fin V.rank) (Fin r) E) (hP : T * P * T = T) (hW : ∀ x, x ∈ W ↔ (T *ᵥ fun j => V.frame.repr x j) = 0) (y : Fin V.rank → R ⊗[F] E) : y ∈ extSub (F

theorem **exists_cut** (W : Submodule E V.V) : ∃ T : Matrix (Fin V.rank) (Fin V.rank) E, T * 1 * T = T ∧ ∀ x, x ∈ W ↔ (T *ᵥ fun j => V.frame.repr x j) = 0

@[simp] theorem **extSub_bot** : extSub (F

@[simp] theorem **extSub_top** : extSub (F

theorem **coordR_mem_iff** [Nontrivial R] {W : Submodule E V.V} {x : V.V} : coordR (F

@[simp] theorem **mapVec_coordR** (φ : R →ₐ[F] R') (x : V.V) : mapVec V φ (coordR V R x) = coordR V R' x

theorem **mapVec_sub** (φ : R →ₐ[F] R') (y z : Fin V.rank → R ⊗[F] E) : mapVec V φ (y − z) = mapVec V φ y − mapVec V φ z

theorem **map_mulVec_mapVec** (φ : R →ₐ[F] R') (T : Matrix (Fin V.rank) (Fin V.rank) E) (y : Fin V.rank → R ⊗[F] E) : T.map (inclR (F

theorem **mapVec_mem** (φ : R →ₐ[F] R') {W : Submodule E V.V} {y : Fin V.rank → R ⊗[F] E} (hy : y ∈ extSub (F

theorem **repr_map** {ι : Type} (b : Module.Basis ι F E) (φ : R →ₐ[F] R') (u : R ⊗[F] E) (j : ι) : (Algebra.TensorProduct.basis R' b).repr (Algebra.TensorProduct.map φ (AlgHom.id F E) u) j = φ ((Algebra.TensorProduct.basis R b).repr u j)

theorem **hR_eq** (x y : Fin V.rank → R ⊗[F] E) : hR R hc.sig V.gram x y = (fun i => Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R hc.sig (x i)) ⬝ᵥ (V.gram.map (inclR (F

theorem **hR_mulVec** {r : ℕ} (M : Matrix (Fin V.rank) (Fin r) (R ⊗[F] E)) (a : Fin r → R ⊗[F] E) (y : Fin V.rank → R ⊗[F] E) : hR R hc.sig V.gram (M *ᵥ a) y = (fun i => Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R hc.sig (a i)) ⬝ᵥ ((cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R hc.sig) M * V.gram.map (inclR (F

theorem **hR_act** (g : HA R hc.sig V.gram) (x y : Fin V.rank → R ⊗[F] E) : hR R hc.sig V.gram (act hc V g x) (act hc V g y) = hR R hc.sig V.gram x y

theorem **hR_act_sub** (g : HA R hc.sig V.gram) (x y : Fin V.rank → R ⊗[F] E) : hR R hc.sig V.gram x (act hc V g y − y) = hR R hc.sig V.gram (act hc V g⁻¹ x − x) y

theorem **mem_extSub_perp** {W : Submodule E V.V} {y : Fin V.rank → R ⊗[F] E} : y ∈ extSub (F

theorem **mem_parab_iff_maps** : g ∈ parab hc V R W ↔ ∀ y ∈ extSub (F

theorem **mem_parab** : g ∈ parab hc V R W ↔ ∀ x ∈ W, act hc V g (coordR V R x) ∈ extSub (F

@[simp] theorem **parab_bot** : parab hc V R ⊥ = ⊤

theorem **parab_le_perp** : parab hc V R W ≤ parab hc V R (perp V W)

theorem **mem_trivOn_iff_maps** (h : A ≤ B) : g ∈ trivOn hc V R A B ↔ ∀ y ∈ extSub (F

theorem **mem_trivOn** (h : A ≤ B) : g ∈ trivOn hc V R A B ↔ ∀ x ∈ B, act hc V g (coordR V R x) − coordR V R x ∈ extSub (F

theorem **trivOn_le_parab** (h1 : A ≤ W) (h2 : W ≤ B) : trivOn hc V R A B ≤ parab hc V R W

theorem **trivOn_mono** {A' B' : Submodule E V.V} (h1 : A ≤ A') (h2 : A' ≤ B') (h3 : B' ≤ B) : trivOn hc V R A B ≤ trivOn hc V R A' B'

theorem **trivOn_le_trivOn_perp** (h : A ≤ B) : trivOn hc V R A B ≤ trivOn hc V R (perp V B) (perp V A)

theorem **conj_mem_trivOn** {p : HA R hc.sig V.gram} (hA : p ∈ parab hc V R A) (hB : p ∈ parab hc V R B) (hg : g ∈ trivOn hc V R A B) : p * g * p⁻¹ ∈ trivOn hc V R A B

@[simp] theorem **trivOn_bot_top** : trivOn hc V R ⊥ ⊤ = ⊥

theorem **act_mapH** (φ : R →ₐ[F] R') (g : HA R hc.sig V.gram) (y : Fin V.rank → R ⊗[F] E) : act hc V (mapH hc.sig V.gram φ g) (mapVec V φ y) = mapVec V φ (act hc V g y)

theorem **mapH_mem_parab** (φ : R →ₐ[F] R') (hg : g ∈ parab hc V R W) : mapH hc.sig V.gram φ g ∈ parab hc V R' W

theorem **mapH_mem_trivOn** (h : A ≤ B) (φ : R →ₐ[F] R') (hg : g ∈ trivOn hc V R A B) : mapH hc.sig V.gram φ g ∈ trivOn hc V R' A B

theorem **isClosed_act_sub_mem** (W : Submodule E V.V) (y z : Fin V.rank → R ⊗[F] E) : IsClosed {g : HA R hc.sig V.gram | act hc V g y − z ∈ extSub (F

theorem **isClosed_parab** (W : Submodule E V.V) : IsClosed (parab hc V R W : Set (HA R hc.sig V.gram))

theorem **isClosed_trivOn** {A B : Submodule E V.V} (h : A ≤ B) : IsClosed (trivOn hc V R A B : Set (HA R hc.sig V.gram))

theorem **mem_extSub_of_adelic** {y : Fin V.rank → AF F ⊗[F] E} (h : ∀ w, mapVec V (evInf F w) y ∈ extSub (F

theorem **mem_parab_of_adelic** (h : ∀ w, projArch hc.sig V.gram w g ∈ parab hc V w.Completion W) (h' : ∀ v, proj hc.sig V.gram v g ∈ parab hc V (Fv F v) W) : g ∈ parab hc V (AF F) W

theorem **mem_trivOn_of_adelic** (e : A ≤ B) (h : ∀ w, projArch hc.sig V.gram w g ∈ trivOn hc V w.Completion A B) (h' : ∀ v, proj hc.sig V.gram v g ∈ trivOn hc V (Fv F v) A B) : g ∈ trivOn hc V (AF F) A B

theorem **act_diagH** (γ : GRat hc V) (x : V.V) : act hc V (diagH hc.sig V.gram R γ) (coordR V R x) = coordR V R (ratAct hc V γ x)

theorem **diagH_mem_parab** : diagH hc.sig V.gram R γ ∈ parab hc V R W ↔ ∀ x ∈ W, ratAct hc V γ x ∈ W

theorem **diagH_mem_trivOn** (h : A ≤ B) : diagH hc.sig V.gram R γ ∈ trivOn hc V R A B ↔ ∀ x ∈ B, ratAct hc V γ x - x ∈ A

theorem **le_top** (i : Fin (𝔉.k + 1)) : 𝔉.X i ≤ 𝔉.top

theorem **le_succ** (i : Fin 𝔉.k) : 𝔉.X i.castSucc ≤ 𝔉.X i.succ

theorem **le_perp** (i j : Fin (𝔉.k + 1)) : 𝔉.X i ≤ perp V (𝔉.X j)

theorem **isIsotropic** {i : Fin (𝔉.k + 1)} (hi : i ≠ 0) : IsIsotropic V (𝔉.X i)

theorem **Proper.isIsotropic_top** {𝔉 : IsoFlag V} (h : 𝔉.Proper) : IsIsotropic V 𝔉.top

theorem **proper_single** {W : Submodule E V.V} (hW : IsIsotropic V W) : (single hW).Proper

theorem **mem_par** : g ∈ 𝔉.par hc R ↔ ∀ i, g ∈ parab hc V R (𝔉.X i)

theorem **mem_unip** : g ∈ 𝔉.unip hc R ↔ (∀ i : Fin 𝔉.k, g ∈ trivOn hc V R (𝔉.X i.castSucc) (𝔉.X i.succ)) ∧ g ∈ trivOn hc V R 𝔉.top (perp V 𝔉.top)

theorem **mem_unip_iff** : g ∈ 𝔉.unip hc R ↔ (∀ i : Fin 𝔉.k, ∀ x ∈ 𝔉.X i.succ, act hc V g (coordR V R x) - coordR V R x ∈ extSub (F

theorem **unip_le_par** : 𝔉.unip hc R ≤ 𝔉.par hc R

theorem **conj_mem_unip** {p : HA R hc.sig V.gram} (hp : p ∈ 𝔉.par hc R) (hg : g ∈ 𝔉.unip hc R) : p * g * p⁻¹ ∈ 𝔉.unip hc R

@[simp] theorem **par_nil** : (nil V).par hc R = ⊤

@[simp] theorem **unip_nil** : (nil V).unip hc R = ⊥

theorem **par_single** : (single hW).par hc R = parab hc V R W

theorem **unip_single** : (single hW).unip hc R = unipRad hc V R W

theorem **par_le_parab** (i : Fin (𝔉.k + 1)) : 𝔉.par hc R ≤ parab hc V R (𝔉.X i)

theorem **par_le_parab_perp** (i : Fin (𝔉.k + 1)) : 𝔉.par hc R ≤ parab hc V R (perp V (𝔉.X i))

theorem **unip_single_le** {i : Fin (𝔉.k + 1)} (hi : i ≠ 0) : (single (𝔉.isIsotropic hi)).unip hc R ≤ 𝔉.unip hc R

theorem **mapH_mem_par** (φ : R →ₐ[F] R') (hg : g ∈ 𝔉.par hc R) : mapH hc.sig V.gram φ g ∈ 𝔉.par hc R'

theorem **mapH_mem_unip** (φ : R →ₐ[F] R') (hg : g ∈ 𝔉.unip hc R) : mapH hc.sig V.gram φ g ∈ 𝔉.unip hc R'

theorem **mem_parRat** {γ : GRat hc V} : γ ∈ 𝔉.parRat hc ↔ ∀ i, ∀ x ∈ 𝔉.X i, ratAct hc V γ x ∈ 𝔉.X i

theorem **mem_unipRat** {γ : GRat hc V} : γ ∈ 𝔉.unipRat hc ↔ (∀ i : Fin 𝔉.k, ∀ x ∈ 𝔉.X i.succ, ratAct hc V γ x - x ∈ 𝔉.X i.castSucc) ∧ ∀ x ∈ perp V 𝔉.top, ratAct hc V γ x - x ∈ 𝔉.top

theorem **diagH_mem_par** [Nontrivial R] {γ : GRat hc V} : diagH hc.sig V.gram R γ ∈ 𝔉.par hc R ↔ γ ∈ 𝔉.parRat hc

theorem **diagH_mem_unip** [Nontrivial R] {γ : GRat hc V} : diagH hc.sig V.gram R γ ∈ 𝔉.unip hc R ↔ γ ∈ 𝔉.unipRat hc

theorem **isClosed_par** : IsClosed (𝔉.par hc R : Set (HA R hc.sig V.gram))

theorem **isClosed_unip** : IsClosed (𝔉.unip hc R : Set (HA R hc.sig V.gram))

theorem **mem_par_adelic** : g ∈ 𝔉.par hc (AF F) ↔ (∀ w, projArch hc.sig V.gram w g ∈ 𝔉.par hc w.Completion) ∧ ∀ v, proj hc.sig V.gram v g ∈ 𝔉.par hc (Fv F v)

theorem **mem_unip_adelic** : g ∈ 𝔉.unip hc (AF F) ↔ (∀ w, projArch hc.sig V.gram w g ∈ 𝔉.unip hc w.Completion) ∧ ∀ v, proj hc.sig V.gram v g ∈ 𝔉.unip hc (Fv F v)

theorem **emb_mem_unip** {v : HeightOneSpectrum (𝒪 F)} {x : GLoc hc.sig V.gram v} : emb hc.sig V.gram v x ∈ 𝔉.unip hc (AF F) ↔ x ∈ 𝔉.unip hc (Fv F v)

theorem **emb_mem_par** {v : HeightOneSpectrum (𝒪 F)} {x : GLoc hc.sig V.gram v} : emb hc.sig V.gram v x ∈ 𝔉.par hc (AF F) ↔ x ∈ 𝔉.par hc (Fv F v)

theorem **isGreatest_wittIndex** : IsGreatest {d | ∃ W : Submodule E V.V, W ≤ perp V W ∧ Module.finrank E W = d} (wittIndex V)

theorem **finrank_le_wittIndex** {W : Submodule E V.V} (h : W ≤ perp V W) : Module.finrank E W ≤ wittIndex V

theorem **le_wittIndex_iff** {i : ℕ} : i ≤ wittIndex V ↔ ∃ W : Submodule E V.V, W ≤ perp V W ∧ Module.finrank E W = i

theorem **wittIndex_pos_iff** : 0 < wittIndex V ↔ ∃ W, IsIsotropic V W

theorem **wittIndex_eq_zero_iff** : wittIndex V = 0 ↔ ∀ x, V.hD x x = 0 → x = 0

theorem **IsoFlag.k_le_wittIndex** (𝔉 : IsoFlag V) : 𝔉.k ≤ wittIndex V

theorem **exists_proper_iff** : (∃ 𝔉 : IsoFlag V, 𝔉.Proper) ↔ 0 < wittIndex V

Development.AbelianVarietyTheory.Definitions.Foundations.ZetaElementsUnramifiedOrdinaryOfEulerSystemInert

theorem **zetaElementsUnramifiedOrdinary_of_eulerSystem_inertia** (h : ZetaSysUr2) : ZetaUr

theorem **zetaSysU_of_halves2** (Pin : ∀ {Ω : Type} [Field Ω] [Algebra ℚ Ω] {l : ℕ} [Fact l.Prime], ℕ → (ValuationSubring Ω) → R2 Ω l → Coef l → (G ℚ Ω → (Qbar l)ˣ) → (∀ r : ℕ, (ZMod (l ^ r))ˣ → Qbar l) → (∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l) → Prop) (hG : ExistsZetaEulerSystemIwasawaData Pin) (hL : ZetaEulerSystemNonvanishing Pin) : ZetaSysUr2

theorem **zetaUr_of_halves2** (Pin : ∀ {Ω : Type} [Field Ω] [Algebra ℚ Ω] {l : ℕ} [Fact l.Prime], ℕ → (ValuationSubring Ω) → R2 Ω l → Coef l → (G ℚ Ω → (Qbar l)ˣ) → (∀ r : ℕ, (ZMod (l ^ r))ˣ → Qbar l) → (∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l) → Prop) (hG : ExistsZetaEulerSystemIwasawaData Pin) (hL : ZetaEulerSystemNonvanishing Pin) : ZetaUr

Development.AbelianVarietyTheory.Definitions.Foundations.ZetaSysUOfHalves

theorem **zetaUr_of_sys** (h : ZetaSysUr) : ZetaUr

theorem **zetaSysU_of_halves** (Pin : ∀ {Ω : Type} [Field Ω] [Algebra ℚ Ω] {l : ℕ} [Fact l.Prime], ℕ → (ValuationSubring Ω) → R2 Ω l → Coef l → (G ℚ Ω →* (Qbar l)ˣ) → (∀ r : ℕ, (ZMod (l ^ r))ˣ → Qbar l) → (∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l) → Prop) (hG : GlobU Pin) (hL : LocU Pin) : ZetaSysUr

theorem **zetaUr_of_halves** (Pin : ∀ {Ω : Type} [Field Ω] [Algebra ℚ Ω] {l : ℕ} [Fact l.Prime], ℕ → (ValuationSubring Ω) → R2 Ω l → Coef l → (G ℚ Ω →* (Qbar l)ˣ) → (∀ r : ℕ, (ZMod (l ^ r))ˣ → Qbar l) → (∀ r : ℕ, G ℚ Ω → (ZMod (l ^ r))ˣ → Qbar l) → Prop) (hG : GlobU Pin) (hL

: LocU Pin) : ZetaUr

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ActEEqOfConjRelated

theorem **actE_eq_of_conj_related** {Γ W : Type} [Group Γ] [AddCommGroup W] [Module ℂ W] (ρ : Representation ℂ Γ W) (g h : Γ) {x y : W} (e : x = ρ (g⁻¹ * h * g) y) : actE ρ g x = ρ h (actE ρ g y)

theorem **block_frame_conjugation_identity** {H : Type} [Group H] (S : Subgroup H) (g h : S) (a : H) : ((g : H) * a)⁻¹ * h * (g * a) = a⁻¹ * ((g⁻¹ * h * g : S) : H) * a

theorem **inv_aux** {H : Type} [Group H] (g a h : H) : a⁻¹ * (g⁻¹ * h) = (g * a)⁻¹ * h

theorem **real_apply_eq_actE_real_inv_mul** {μ j} (m : MtReal D μ j) (x) (g : ReciprocityTheory.Aut.GQ D.host.E D.R.nV) (h : Formalism.AUTREP.GAf D.host.E D.R.nV) : m.real x h = actE m.wt.ρ g (m.real x ((g : Formalism.AUTREP.GAf D.host.E D.R.nV)⁻¹ * h))

theorem **shift_a** (A : Agree X M rep) (g : Fin D.R.m → ReciprocityTheory.Aut.GQ D.host.E D.R.nV) (j) : (shift A g).a j = g j * A.a j

theorem **real_eq** {M' : MtFam D} (A : Agree X M rep) (A' : Agree X M' rep) (hP : A.PV = A'.PV) (ha : A.a = A'.a) {μ} (hμ : Used0 D μ) (j : Fin D.R.m) (x) (h) : (A'.eW μ hμ j).symm ((M' μ (j.1 + 1) (by omega) (by omega)).real x h) = (A.eW μ hμ j).symm ((M μ (j.1 + 1) (by omega) (by omega)).real x h)

theorem **eq_zero_of_fin** (A : Agree X M rep) {μ} (hμ : Used0 D μ) (j : Fin D.R.m) (x : D.what₂.Mt μ (j.1 + 1)) (hx : ∀ g ∈ GfC (Vc D j), X.fn hμ j x g = 0) : x = 0

theorem **rep_inj** (A : Agree X M rep) (σ σ' : D.what₂.FT) (h : rep σ = rep σ') : σ = σ'

theorem **canon_phi_inj** (j : Fin D.R.m) (Y : GlobalTheory.NPin.RowMod D rep (j.1 + 1)) (h : IsCanon X j rfl Y) : Function.Injective Y.φ

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.AddCharThetaLawRearrangementIdentity

theorem **finitePart_surjective** : Function.Surjective (finP P)

theorem **finitePart_injective** : Function.Injective (finP P)

theorem **ext_psi** (hc : ∀ h ∈ GfC V, c.ρ h = 1) (φ : Bridge.Gf V → c.W) (h : Formalism.AUTREP.GAf D.host.E n) : c.ext φ (P.ψ h) = φ (finP P h)

theorem **mem_levi_flag_of_mem_parabolic** {x : GL (Fin V.rank) (EAf D)} (hx : x ∈ lev D.sp.cm V.gram β (flag1L L)) : x ∈ lev D.sp.cm V.gram β L ∧ levFn β (flag1L L) x = levFn β L x

theorem **rd_ext_eq_zero** (hc : ∀ h ∈ GfC V, c.ρ h = 1) (φ : Bridge.Gf V → c.W) (h : rd P c e (c.ext φ) = 0) : c.ext φ = 0

theorem **adForms_left** {G Γ W : Type} [Group G] [Group Γ] (ι : Γ →* G) [NormedAddCommGroup W] [InnerProductSpace ℂ W] {ρ : Γ →* Module.End ℂ W} {J : Subgroup G} {χ : J →* ℂˣ} {φ : G → W} (hm : φ ∈ Bridge.AdForms ι ρ J χ) (γ : Γ) (h : G) : φ (ι γ * h) = ρ γ (φ h)

theorem **rho_inv_mul** {G M : Type} [Group G] [Monoid M] (ρ : G →* M) (g k : G) (h : ρ k⁻¹ = 1) : ρ (g * k)⁻¹ = ρ g⁻¹

theorem **wachExpansion_smul_compatible** {M N : Type} [AddCommGroup M] [Module ℂ M] [AddCommGroup N] [Module ℂ N] (e : M ≃ₗ[ℂ] N) {a b : Module.End ℂ M} {x y : N} {z : ℂ} (h1 : a = b) (h2 : x = z • y) : a (e.symm x) = z • b (e.symm y)

theorem **addChar_theta_law_rearrangement_identity** {M N : Type} [AddCommGroup M] [Module ℂ M] [AddCommGroup N] [Module ℂ N] (e : M ≃ₗ[ℂ] N) {a : Module.End ℂ M} {b : Module.End ℂ N} (h : ∀ v, e (a v) = b (e v)) {x y : N} {z z' : ℂ} (hz : z' = z) (h2 : x = z • b y) : e.symm x = z' • a (e.symm y)

theorem **exists_section** {A B : Type} [Group A] [Group B] (f : A →* B) (h1 : Function.Injective f) (h2 : Function.Surjective f) : ∃ s : B →* A, (∀ x, f (s x) = x) ∧ ∀ a, s (f a) = a

theorem **rd_left** {J : Subgroup (Bridge.Gf V)} {χ : J →* ℂˣ} {φ : Bridge.Gf V → c.W} (hφ : φ ∈ Bridge.AdForms (Bridge.ratF V) (c.ρ.comp (Formalism.L2.Theta.GQ V).subtype) J χ) (h : ReciprocityTheory.Aut.GQ D.host.E n) (g : Formalism.AUTREP.GAf D.host.E n) : rd P c e (c.ext φ) (h * g) = ρ h (rd P c e (c.ext φ) g)

theorem **map_forms_le** (hL : principalLevelFrame.LevOK P (levS V β L) (levChi V β L)) : (forms c β L).map (rd P c e) ≤ NG.TypeSp ρ (levJ P β L) (levX P β L)

theorem **mem_adForms** {f : Formalism.AUTREP.GAf D.host.E n → W} (hf : f ∈ NG.TypeSp ρ (levJ P β L) (levX P β L)) (s : Bridge.Gf V →* Formalism.AUTREP.GAf D.host.E n) (hs : ∀ x, finP P (s x) = x) : (fun x => e.symm (f (s x))) ∈ Bridge.AdForms (Bridge.ratF V) (c.ρ.comp (Formalism.L2.Theta.GQ V).subtype) (levEnv V β L).J (levEnv V β L).χ

theorem **map_forms** (hL : principalLevelFrame.LevOK P (levS V β L) (levChi V β L)) : (forms c β L).map (rd P c e) = NG.TypeSp ρ (levJ P β L) (levX P β L)

theorem **rat_unit** {V : HSp D} (hg : V.gram = 1) (h : Formalism.AUTREP.GAf D.host.E V.rank) : h ∈ ReciprocityTheory.Aut.GQ D.host.E V.rank ↔ ∃ γ ∈ Formalism.L2.Theta.GQ V, finHom V γ = finHom V ((GlobalTheory.StdFrameModel.unitPsi hg).ψ h)

theorem **levOK_of_std** {V : HSp D} {n : ℕ} {P : RowPsi V n} {hF : D.host.F = maximalRealSubfield D.host.E} (hP : StdFrame hF P) (β : GL (Fin V.rank) (EAf D)) (L : LevDat D.host.E ℂ V.rank) : principalLevelFrame.LevOK P (levS V β L) (levChi V β L)

theorem **amt_eq** (μ : D.what₂.Tup) (j : Fin D.R.m) : (toLiftReal X.R X.F X.tupT X.hm X.hTop X.hinj).Amt μ j = forms (X.R.cV j) (X.R.βV j) (datOf X μ j)

theorem **range_fnL** {μ : D.what₂.Tup} (hμ : Used0 D μ) (j : Fin D.R.m) : LinearMap.range (fnL X hμ j) = (toLiftReal X.R X.F X.tupT X.hm X.hTop X.hinj).Amt μ j

theorem **xi_cast** {C C' : CharFrame D} (h : C = C') (tp : TupTied D C' nch) : (h ▸ tp : TupTied D C nch).xi = tp.xi

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.AlgTAll

theorem **algTAll** : Formalism.L2.KerInert.AlgTAll

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.Alternates

theorem **plc_mem** {fr : Formalism.HC.Frame} (g : fr.GG) : plc g ∈ fr.Psi

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ApartAllEmpty

theorem **apartAll_empty** (w : HeightOneSpectrum (𝒪 E)) : ApartAll (∅ : Finset (HeightOneSpectrum (𝒪 E))) w

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.AssocSetApplyOfPointwise

theorem **cPre_isSheaf** (X : Scheme.{0}) (ι : Type) : Presieve.IsSheaf (Scheme.proetaleTopology.{0}.over X) (cPre E X ι)

theorem **assocSet_nat** {T : Over X} (f : assocSet E Q ξ T) {T'' T' : Over X} (v : T'' ⟶ T') (u : T' ⟶ T) (η : Q.P.obj (op T')) : f.1 T'' (v ≫ u) (Q.P.map v.op η) = (cPre E X ι).map v.op (f.1 T' u η)

theorem **assocPre_isSheaf** (Q : RActPre X Γ) (ξ : Γ →* GL ι E) : Presheaf.IsSheaf (Scheme.proetaleTopology.{0}.over X) (assocPre E Q ξ)

theorem **ptHom_apply** (T' : Over X) (t : T'.left) (y : (ptOver T' t).left) : (ptHom T' t).left.base y = t

theorem **assocSet_apply_of_pointwise** (Q : RActPre X Γ) (ξ : Γ →* GL ι E) {T : Over X} (f : assocSet E Q ξ T) {T' : Over X} (u : T' ⟶ T) (η η' : Q.P.obj (op T')) (g : T'.left → Γ) (h : ∀ t, Q.P.map (ptHom T' t).op η' = (Q.act (g t)).app _ (Q.P.map (ptHom T' t).op η)) (t : T'.left) : f.1 T' u η' t = ((ξ (g t))⁻¹ : GL ι E) : Matrix ι ι E) *ᵥ f.1 T' u η t

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.BaseLatticeSelfDualOfLocalEq

theorem **IsBaseAt.congr** {N : ℕ} {B : GL (Fin N) D.host.E} {Lb Lb' : Submodule (𝒪 D.host.E) (Fin N → D.host.E)} (h : IsBaseAt X0 u w hg B Lb) (e : ∀ w' : HeightOneSpectrum (𝒪 D.host.F), ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal → Formalism.L2.Herm.locL w' Lb' = Formalism.L2.Herm.locL w' Lb) : IsBaseAt X0 u w hg B Lb'

theorem **rho_int** (hs : SplitInt v spl) (x : E ⊗[F] v.adicCompletion F) (hx : x ∈ intV E v) : rho v spl x ∈ v.adicCompletionIntegers F

theorem **mulVec_map_rho** (A : Matrix (Fin N) (Fin N) E) (x : Fin N → E) (a : Fin N) : ((A.map ((rho v spl).comp (inclE (F

theorem **base_free** (hs : SplitInt v spl) (hd : ∀ a : v.adicCompletion F, spl ((1 : E) ⊗ₜ[F] a) = (a, a)) (B : GL (Fin N) E) (Λ : Submodule (𝒪 E) (Fin N → E)) (hΛ : ∀ x, x ∈ Λ ↔ ∀ a, (((B⁻¹ : GL (Fin N) E) : Matrix (Fin N) (Fin N) E) *ᵥ x) a ∈ (algebraMap (𝒪 E) E).range) (t : Fin N → v.adicCompletion F) : (∃ x ∈ Formalism.L2.Herm.locL v Λ, ∀ a, (spl (x a)).1 = t a) ↔ ∀ a, (((bc ((rho v spl).comp (inclE (F

theorem **isBaseAt_free** {N : ℕ} (B : GL (Fin N) D.host.E) (Λ : Submodule (𝒪 D.host.E) (Fin N → D.host.E)) (hΛ : ∀ x, x ∈ Λ ↔ ∀ a, (((B⁻¹ : GL (Fin N) D.host.E) : Matrix (Fin N) (Fin N) D.host.E) *ᵥ x) a ∈ (algebraMap (𝒪 D.host.E) D.host.E).range) : IsBaseAt X0 u w hg B Λ

theorem **reGL_symm_rfl** {a : ℕ} (R : Type) [CommRing R] (x : GL (Fin a) R) : (Formalism.L2.Forms.reGL rfl R).symm x = x

theorem **free_re** {E : Type} [Field E] {a b : ℕ} (h : a = b) (C : GL (Fin b) E) (S : Set E) (x : Fin a → E) : (∀ k, ((C : Matrix (Fin b) (Fin b) E) *ᵥ reV h E x) k ∈ S) ↔ ∀ k, (((((Formalism.L2.Forms.reGL h E).symm C⁻¹)⁻¹ : GL (Fin a) E) : Matrix (Fin a) (Fin a) E) *ᵥ x) k ∈ S

theorem **mem_crdLat_iff** (i : ℕ) (hi : i ≤ D.R.m) (x : Fin (D.sp.X i).rank → D.host.E) : x ∈ Formalism.L2.Herm.crdLat (D.sp.X i) (D.sp.X i).L0 ↔ reV (rankN D i hi) D.host.E x ∈ selfDualLatticeCoords D i hi

theorem **std_of_C** {E : Type} [Field E] (σ : E →+* E) {N : ℕ} (J : Matrix (Fin N) (Fin N) E) (C : GL (Fin N) E) (hC : cT σ (C : Matrix (Fin N) (Fin N) E) * (C : Matrix (Fin N) (Fin N) E) = J) : cT σ ((C⁻¹ : GL (Fin N) E) : Matrix (Fin N) (Fin N) E) * J * ((C⁻¹ : GL (Fin N) E) : Matrix (Fin N) (Fin N) E) = 1

theorem **baseEx_of** (Fo : FormsIn M (i + 1) hi) (Cm : GL (Fin (D.R.N (i + 1))) D.host.E) (h1 : cT (sigD D : D.host.E →+* D.host.E) Cm.1 * Cm.1 = gramN D (i + 1) hi) (h2 : ∀ x : Fin (D.R.N (i + 1)) → D.host.E, x ∈ selfDualLatticeCoords D (i + 1) hi ↔ ∀ a, (Cm.1 *ᵥ x) a ∈ (algebraMap (𝒪 D.host.E) D.host.E).range) (hL : ChartBaseLatticeEqSelfDualAtP u Fo (D.what₂.places.resChar w)) : BaseEx X0 u w hg Fo

theorem **locL_mono** {n : ℕ} (v : HeightOneSpectrum (𝒪 F)) {Λ Λ' : Submodule (𝒪 E) (Fin n → E)} (h : Λ ≤ Λ') : locL v Λ ≤ locL v Λ'

theorem **natVal_one** (v : HeightOneSpectrum (𝒪 F)) (T : ℕ) (hT : (T : 𝒪 F) ∉ v.asIdeal) : Valued.v ((T : ℕ) : v.adicCompletion F) = 1

theorem **natInv_mem** (v : HeightOneSpectrum (𝒪 F)) (T : ℕ) (hT : (T : 𝒪 F) ∉ v.asIdeal) : ((T : ℕ) : v.adicCompletion F)⁻¹ ∈ v.adicCompletionIntegers F

theorem **loc_lattice_eq_of_torsion_away** {n : ℕ} (v : HeightOneSpectrum (𝒪 F)) (Λ Λ' : Submodule (𝒪 E) (Fin n → E)) (T : ℕ) (hT : (T : 𝒪 F) ∉ v.asIdeal) (h1 : Λ ≤ Λ') (h2 : ∀ y ∈ Λ', (T : E) • y ∈ Λ) : locL v Λ' = locL v Λ

theorem **reL_symm_reL** {a b : ℕ} (h : a = b) {E : Type} [Field E] [NumberField E] (L : Submodule (𝒪 E) (Fin a → E)) : reL h.symm (reL h L) = L

theorem **locL_memCh** (H : HermColsCh D) (aT : ∀ u, AtT D u (H.vT u) (H.Fv u)) (hb : SymmetricBaseCompatibility D H aT) (u : D.what₂.Word) (i : ℕ) (h1 : 1 ≤ i) (hm : i ≤ D.R.m) (β : H.Ch u i) (v : HeightOneSpectrum (𝒪 D.host.F)) (hT : (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal) : locL v (H.lev u i h1 hm β).Lb = locL v (crdLat (D.sp.X i) (D.sp.X i).L0)

theorem **base_lattice_self_dual_of_local_eq** (Fo : FormsIn M (i + 1) hi) (p : ℕ) (Lh : Fo.Ch → Submodule (𝒪 D.host.E) (Fin (D.sp.X (i + 1)).rank → D.host.E)) (hcar : ∀ β, (Fo.K β).lev.Lb = reL (rankN D (i + 1) hi) (Lh β)) (hL : ∀ β (w' : HeightOneSpectrum (𝒪 D.host.F)), ((p : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal → locL w' (Lh β) = locL w' (crdLat (D.sp.X (i + 1)) (D.sp.X (i + 1)).L0)) : BaseLatticeSelfDualAbove Fo p

theorem **top_car** (u : D.what₂.Word) (j : ℕ) (hj : j ≤ D.R.m) (h : j = D.R.m) (h1 : 1 ≤ j) : ∃ β' : cols.herm.base.Ch u j, ((cols.forms u).cls j hj (Or.inr h)).lev.Lb = reL (rankN D j hj) (cols.herm.base.lev u j h1 hj β').Lb

theorem **base_lattice_self_dual_of_chain** (u : D.what₂.Word) (dt : MemDt (cols.forms u)) (i : ℕ) (hi : i + 1 ≤ D.R.m) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : BaseLatticeSelfDualAbove (formsInMem (cols.forms u) dt i hi) (D.what₂.places.resChar w)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.BlockRowSubstitutionPreservesComplexSpan

theorem **finrank_le_span** {O Kf W : Type} [CommRing O] [IsDomain O] [Field Kf] [Algebra O Kf] [IsFractionRing O Kf] [AddCommGroup W] [Module Kf W] [Module O W] [IsScalarTower O Kf W] (M : Submodule O W) [FiniteDimensional Kf (Submodule.span Kf (M : Set W))] : finrank O M ≤ finrank Kf (Submodule.span Kf (M : Set W))

theorem **M_le_Eqv** : (H.M : Set H.Fn) ⊆ Eqv H

theorem **span_le_Eqv** : Submodule.span H.Kf (H.M : Set H.Fn) ≤ Eqv H

theorem **eqv_eq_zero** {rep : Cl → H.comp × H.P.Tup} (hc : Cover H rep) {F : H.Fn} (hF : F ∈ Eqv H) (h0 : ∀ c, F (rep c).1 (rep c).2 = 0) : F = 0

theorem **eqv_mem_InvK** {F : H.Fn} (hF : F ∈ Eqv H) (a : H.comp) (L : H.P.Tup) : F a L ∈ InvK H a L

theorem **evR_inj** {rep : Cl → H.comp × H.P.Tup} (hc : Cover H rep) : Function.Injective (evR H rep)

theorem **finrank_le_of_cover** [IsFractionRing H.O H.Kf] {rep : Cl → H.comp × H.P.Tup} (hc : Cover H rep) (hfin : ∀ c, FiniteDimensional H.Kf (InvK H (rep c).1 (rep c).2)) : finrank H.O H.M ≤ ∑ c, finrank H.Kf (InvK H (rep c).1 (rep c).2)

theorem **finDim_FiB** (S : StepIn M i hi G) (β : S.Ch) (hβ : β ∈ chs S) : FiniteDimensional ℂ (FiB S β)

theorem **sum_le_finrank_FiB** (S : StepIn M i hi G) (β : S.Ch) (hπ : FiBPi S β) : ∑ c, finrank ℂ (FiBc S β c) ≤ finrank ℂ (FiB S β)

theorem **coverOf** {S : StepIn M i hi G} {e : D.what₂.LL →+* G.Kf} (H : ∀ β, S.Core e β) (X : KAat.ChX S) (β : S.Ch) : ∃ rep, RepOk S β rep ∧ Cover (Hc S β) rep

theorem **mk_eq_of** (S : StepIn M i hi G) (β : S.Ch) {γ : GL (Fin (D.R.N i)) D.host.E} (hγ : γ ∈ UJ (sigD D : D.host.E →+* D.host.E) (gramN D i hi)) {x x' : UPts (sigD D) (gramN D i hi) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)} {κ : (S.K β).lev.U} (hx : x'.1 = bc (inclE (F

theorem **fiBPi** (S : StepIn M i hi G) (β : S.Ch) : FiBPi S β

theorem **chartRankGe_of** [IsFractionRing G.O G.Kf] (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (H : ∀ β, S.Core e β) (hS : BlockRationalityField) (X : KAat.ChX S) (hI : InvLeOf S e) : ChartRankGe S

theorem **finrank_le_transfer** (φK : Y →+ WK) (hK : ∀ (a : LL) y, φK (a • y) = e a • φK y) (φC : Y →+ V) [FiniteDimensional C V] (hind : ∀ (k : ℕ) (y : Fin k → Y), LinearIndependent LL y → LinearIndependent C (φC ∘ y)) (N : Submodule K WK) (hN : N ≤ Submodule.span K (Set.range φK)) : FiniteDimensional K N ∧ finrank K N ≤ finrank C V

theorem **mapE_smul** {σ : Type} (r : LL) (Q : MvPolynomial σ LL) : MvPolynomial.map e (r • Q) = e r • MvPolynomial.map e Q

theorem **ker_desc** {σ τ : Type} (Ts : Set (MvPolynomial σ Kf →ₗ[Kf] MvPolynomial τ Kf)) (hT : ∀ T ∈ Ts, ∀ Q : MvPolynomial σ LL, ∃ Q' : MvPolynomial τ LL, MvPolynomial.map e Q' = T (MvPolynomial.map e Q)) {ξ : MvPolynomial σ Kf} (hξ : ∀ T ∈ Ts, T ξ = 0) : ξ ∈ Submodule.span Kf (MvPolynomial.map e '' {Q | ∀ T ∈ Ts, T (MvPolynomial.map e Q) = 0})

theorem **whc_map** {σ M : Type} [AddCommMonoid M] [DecidableEq M] (w : σ → M) (n : M) (Q : MvPolynomial σ LL) : MvPolynomial.map e (MvPolynomial.weightedHomogeneousComponent w n Q) = MvPolynomial.weightedHomogeneousComponent w n (MvPolynomial.map e Q)

theorem **isWH_iff** {σ M R : Type} [CommSemiring R] [AddCommMonoid M] [DecidableEq M] (w : σ → M) (n : M) (f : MvPolynomial σ R) : MvPolynomial.IsWeightedHomogeneous w f n ↔ MvPolynomial.weightedHomogeneousComponent w n f = f

theorem **colAddGen_map** {r n : ℕ} (l j : Fin n) (Q : MvPolynomial (Fin r × Fin n) LL) : MvPolynomial.map e (Formalism.GradedWeylDivisibility.colAddGen LL l j Q) = Formalism.GradedWeylDivisibility.colAddGen Kf l j (MvPolynomial.map e Q)

theorem **dualWeyl_desc** {r n : ℕ} (ν : Fin n → ℕ) : Formalism.GradedWeylDivisibility.dualWeyl Kf r n ν ≤ Submodule.span Kf (MvPolynomial.map e '' {Q | MvPolynomial.map e Q ∈ Formalism.GradedWeylDivisibility.dualWeyl Kf r n ν})

theorem **prod_desc** {S : Type} [Fintype S] {σ : S → Type} {τ : Type} (f : ∀ s, σ s → τ) (Ms : ∀ s, Submodule Kf (MvPolynomial (σ s) Kf)) (hM : ∀ s, Ms s ≤ Submodule.span Kf (MvPolynomial.map e '' {Q | MvPolynomial.map e Q ∈ Ms s})) : Submodule.span Kf {Z | ∃ F : ∀ s, MvPolynomial (σ s) Kf, (∀ s, F s ∈ Ms s) ∧ Z = ∏ s, MvPolynomial.rename (f s) (F s)} ≤ Submodule.span Kf (MvPolynomial.map e '' {Q | MvPolynomial.map e Q ∈ Submodule.span Kf {Z | ∃ F : ∀ s, MvPolynomial (σ s) Kf, (∀ s, F s ∈ Ms s) ∧ Z = ∏ s, MvPolynomial.rename (f s) (F s)}})

theorem **vsp_desc** {n' m : ℕ} (lam0 : Fin n' → Fin (m + 1) → ℤ) : Vsp Kf lam0 ≤ Submodule.span Kf (MvPolynomial.map e '' {Q | MvPolynomial.map e Q ∈ Vsp Kf lam0})

theorem **map_smul_e** (r : LL) (Q : YR H LL) : MvPolynomial.map e (r • Q) = e r • MvPolynomial.map e Q

theorem **spanRat_holds** : SpanRat H e

theorem **span_HtupW_le** {L : H.P.Tup} (hL : H.P.IsLatTup L) : Submodule.span H.Kf (H.P.HtupW L : Set H.P.Xsp) ≤ H.P.Vtup

theorem **inv_le_span** {a : H.comp} {L : H.P.Tup} (hL : H.P.IsLatTup L) (h1 : SpanRat H e) (h2 : ActRat H e a L) : InvK H a L ≤ Submodule.span H.Kf (MvPolynomial.map e '' (YInv H e a L : Set (YR H LL)))

theorem **isLatTup_of_repOk** {S : StepIn M i hi G} {β : S.Ch} {rep : {c // S.chOf c = β} → S.cmpt β × (Hc S β).P.Tup} (hok : RepOk S β rep) (c : {c // S.chOf c = β}) : (Hc S β).P.IsLatTup (rep c).2

theorem **actRat_of_core** {S : StepIn M i hi G} {e : D.what₂.LL →+* G.Kf} {β : S.Ch} (H : S.Core e β) (hS : BlockRationalityField) (a : S.cmpt β) (L : (Hc S β).P.Tup) : ActRat (Hc S β) e a L

theorem **actRatOf_holds** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) : ActRatOf S e

theorem **invLeOf_of** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hR : ActRatOf S e) (hV : ValLiftOf S e) : InvLeOf S e

theorem **invLeOf_of_lift** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hV : ValLiftOf S e) : InvLeOf S e

theorem **valLiftOf_of** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (h1 : FormsOnClassOf S) (h2 : CmpLiftOf S e) : ValLiftOf S e

theorem **chartRankGe_of_lift** [IsFractionRing G.O G.Kf] (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (H : ∀ β, S.Core e β) (hS : BlockRationalityField) (X : KAat.ChX S) (hV : ValLiftOf S e) : ChartRankGe S

theorem **cmpJHyp_of_lift** (c : LatCmp D) {I : CmpJIn Tw u μ w h Lv} (Gt : GatTies D) (hS : BlockRationalityField) (core : ∀ j hj β, (I.at' j hj).st.Core (c.embW u w h) β) (ok : ∀ j hj, (I.at' j hj).st.okW w) (twUnit : ∀ j hj β, (I.at' j hj).st.TwUnit w β) (valIntN : ∀ j hj, (I.at' j hj).st.ValIntN (c.embW u w h) w) (iwF : ∀ j hj β, (I.at' j hj).st.IwF β) (hfin : ∀ j, j + 1 ≤ D.R.m → (gatJ Lv (j + 1)).FinClasses) (X : ∀ j hj, KAat.ChX (I.at' j hj).st) (hV : ∀ j hj, ValLiftOf (I.at' j hj).st (c.embW u w h)) : CmpJHyp c I

theorem **cmpJHyp_of_two** (c : LatCmp D) {I : CmpJIn Tw u μ w h Lv} (Gt : GatTies D) (hS : BlockRationalityField) (core : ∀ j hj β, (I.at' j hj).st.Core (c.embW u w h) β) (ok : ∀ j hj, (I.at' j hj).st.okW w) (twUnit : ∀ j hj β, (I.at' j hj).st.TwUnit w β) (valIntN : ∀ j hj, (I.at' j hj).st.ValIntN (c.embW u w h) w) (iwF : ∀ j hj β, (I.at' j hj).st.IwF β) (hfin : ∀ j, j + 1 ≤ D.R.m → (gatJ Lv (j + 1)).FinClasses) (X : ∀ j hj, KAat.ChX (I.at' j hj).st) (h1 : ∀ j hj, FormsOnClassOf (I.at' j hj).st) (h2 : ∀ j hj, CmpLiftOf (I.at' j hj).st (c.embW u w h)) : CmpJHyp c I

theorem **isTr_inv_mul** (emb : ι → E →+* ℂ) (hc : ∀ τ x, emb τ (σ x) = starRingEnd ℂ (emb τ x)) (γ1 γ2 : GL (Fin n) E) (h1 : γ1 ∈ UJ (σ : E →+* E) J) (h2 : γ2 ∈ UJ (σ : E →+* E) J) (z : Formalism.L2.Forms.Frames (gramC J emb)) : IsTr (atInf emb ((γ1⁻¹ * γ2 : GL (Fin n) E) : Matrix (Fin n) (Fin n) E)) (frameTranslate emb hc γ2⁻¹ ((UJ _ _).inv_mem h2) z) (frameTranslate emb hc γ1⁻¹ ((UJ _ _).inv_mem h1) z)

theorem **law_wd** (S : StepLevel F E σ J T f bad) (emb : ι → E →+* ℂ) (hc : ∀ τ x, emb τ (σ x) = starRingEnd ℂ (emb τ x)) (y0 : UPts σ J (FiniteAdeleRing (𝒪 F) F)) (v : Formalism.L2.Forms.Frames (gramC J emb) → ℂ) (hv : LawAt S emb y0 v) (γ1 γ2 : GL (Fin n) E) (h1 : γ1 ∈ UJ (σ : E →+* E) J) (h2 : γ2 ∈ UJ (σ : E →+* E) J) (κ1 κ2 : S.U) (e : bc (inclE (F

theorem **exists_form_of_law** (S : StepLevel F E σ J T f bad) (emb : ι → E →+* ℂ) (hc : ∀ τ x, emb τ (σ x) = starRingEnd ℂ (emb τ x)) (lam : ι → DomWeight n) (y0 : UPts σ J (FiniteAdeleRing (𝒪 F) F)) (v : Formalism.L2.Forms.Frames (gramC J emb) → ℂ) (hV : v ∈ ValC (gramC J emb) lam) (hv : LawAt S emb y0 v) : ∃ g ∈ AdForms S emb lam, g y0 = v

theorem **formsOnClassOf** (S : StepIn M i hi G) : FormsOnClassOf S

theorem **foNat_anti** (lam : DomWeight n) : Antitone lam.natW

theorem **foRow_delSq** (lam : DomWeight n) (g : Matrix (Fin n) (Fin n) ℂ) {q : MvPolynomial (Fin n × Fin n) ℂ} (hq : q ∈ delSq ℂ n lam) : Formalism.GradedWeylDivisibility.rowAct ℂ g q ∈ delSq ℂ n lam

theorem **blockRowSubstitution_X** (g : ι → Matrix (Fin n) (Fin n) ℂ) (x : ι × Fin n × Fin n) : blockRowSubstitutionAlgHom g (X x) = ∑ c : Fin n, C (g x.1 x.2.1 c) * X (x.1, c, x.2.2)

theorem **eval_blockRowSubstitution** (g : ι → Matrix (Fin n) (Fin n) ℂ) (z : ι × Fin n × Fin n → ℂ) (P : MvPolynomial (ι × Fin n × Fin n) ℂ) : eval z (blockRowSubstitutionAlgHom g P) = eval (fun x => ∑ c, g x.1 x.2.1 c * z (x.1, c, x.2.2)) P

theorem **blockRowSubstitution_rename** (g : ι → Matrix (Fin n) (Fin n) ℂ) (τ : ι) (q : MvPolynomial (Fin n × Fin n) ℂ) : blockRowSubstitutionAlgHom g (rename (Prod.mk τ) q) = rename (Prod.mk τ) (Formalism.GradedWeylDivisibility.rowAct ℂ (g τ) q)

theorem **blockRowSubstitution_tensor** (g : ι → Matrix (Fin n) (Fin n) ℂ) (f : ι → MvPolynomial (Fin n × Fin n) ℂ) : blockRowSubstitutionAlgHom g (ten f) = ten fun τ => Formalism.GradedWeylDivisibility.rowAct ℂ (g τ) (f τ)

theorem **blockRowSubstitution_preserves_complexSpan** (lam : ι → DomWeight n) (g : ι → Matrix (Fin n) (Fin n) ℂ) {P : MvPolynomial (ι × Fin n × Fin n) ℂ} (hP : P ∈ CSpan lam) : blockRowSubstitutionAlgHom g P ∈ CSpan lam

theorem **foValC_tr** {Gm : ι → Matrix (Fin n) (Fin n) ℂ} (lam : ι → DomWeight n) (g : ι → Matrix (Fin n) (Fin n) ℂ) (hg : ∀ τ, IsUnit (g τ).det) {v : Formalism.L2.Forms.Frames Gm → ℂ} (hv : v ∈ ValC Gm lam) : (fun y => v (foTr g hg y)) ∈ ValC Gm lam

theorem **foInf_unit** (emb : ι → E →+* ℂ) (γ : GL (Fin n) E) (τ : ι) : IsUnit (atInf emb γ.1 τ).det

theorem **foInf_mul** (emb : ι → E →+* ℂ) (a b : Matrix (Fin n) (Fin n) E) (τ : ι) : atInf emb (a * b) τ = atInf emb a τ * atInf emb b τ

theorem **foInf_one** (emb : ι → E →+* ℂ) (τ : ι) : atInf emb (1 : Matrix (Fin n) (Fin n) E) τ = 1

theorem **foTrR_val** (emb : ι → E →+* ℂ) (γ : GL (Fin n) E) (y : Formalism.L2.Forms.Frames (gramC J emb)) (τ : ι) : (foTrR emb γ y).1 τ = atInf emb γ.1 τ * y.1 τ

theorem **foTrR_mul** (emb : ι → E →+* ℂ) (γ δ : GL (Fin n) E) (y : Formalism.L2.Forms.Frames (gramC J emb)) : foTrR emb (γ * δ) y = foTrR emb γ (foTrR emb δ y)

theorem **foTrR_one** (emb : ι → E →+* ℂ) (y : Formalism.L2.Forms.Frames (gramC J emb)) : foTrR emb 1 y = y

theorem **foIsTr_iff** (emb : ι → E →+* ℂ) (γ : GL (Fin n) E) (y' y : Formalism.L2.Forms.Frames (gramC J emb)) : IsTr (atInf emb γ) y' y ↔ y = foTrR emb γ y'

theorem **nuOf_anti_ev** {m : ℕ} {lam : Fin (m + 1) → ℤ} (h : Antitone lam) : Antitone (nuOf lam)

theorem **vsp_le** (v : Fin P.nV) : Vsp ℂ (P.lam0 v) ≤ Submodule.span ℂ {Y | ∃ f : Fin (P.n' v) → MvPolynomial (Fin (P.m + 1) × Fin (P.m + 1)) ℤ, (∀ s, f s ∈ weyl ℤ (P.m + 1) (P.m + 1) le_rfl (nuOf (lamAt (P.lam0 v) s))) ∧ Y = ∏ s, blockEmb ℂ s (map (Int.castRingHom ℂ) (f s))}

theorem **vtup_le** : VtupR P ℂ ≤ Submodule.span ℂ {X | ∃ f : ∀ v, Fin (P.n' v) → MvPolynomial (Fin (P.m + 1) × Fin (P.m + 1)) ℤ, (∀ v s, f v s ∈ weyl ℤ (P.m + 1) (P.m + 1) le_rfl (nuOf (lamAt (P.lam0 v) s))) ∧ X = ∏ v, rename (fun x => ((v, x) : Var P)) (∏ s, blockEmb ℂ s (map (Int.castRingHom ℂ) (f v s)))}

theorem **blockRowSubstitution_one** (v : Fin P.nV) (s : Fin (P.n' v)) (f : MvPolynomial (Fin (P.m + 1) × Fin (P.m + 1)) ℤ) : map (RingHom.id ℂ) (bl P rfl pos (fun _ => (1 : GL (Fin (P.m + 1)) ℂ)) v s f) = map (Int.castRingHom ℂ) f

theorem **gen_mem** (G : Tau → Matrix (Fin (P.m + 1)) (Fin (P.m + 1)) ℂ) (M : Tau → GL (Fin (P.m + 1)) ℂ) (lam : Tau → DomWeight (P.m + 1)) (hw : WtTie P rfl pos lam) (f : ∀ v, Fin (P.n' v) → MvPolynomial (Fin (P.m + 1) × Fin (P.m + 1)) ℤ) (hf : ∀ v s, f v s ∈ weyl ℤ (P.m + 1) (P.m + 1) le_rfl (nuOf (lamAt (P.lam0 v) s))) : evC P rfl pos G (fun τ => (M τ : Matrix (Fin (P.m + 1)) (Fin (P.m + 1)) ℂ)) (∏ v, rename (fun x => ((v, x) : Var P)) (∏ s, blockEmb ℂ s (map (Int.castRingHom ℂ) (f v s)))) ∈ ValC G lam

theorem **evC_le** (h : P.m + 1 = n) (pos : (Σ v : Fin P.nV, Fin (P.n' v)) ≃ Tau) (G : Tau → Matrix (Fin n) (Fin n) ℂ) (M : Tau → GL (Fin n) ℂ) (lam : Tau → DomWeight n) (hw : WtTie P h pos lam) : ∀ Q ∈ VtupR P ℂ, evC P h pos G (fun τ => (M τ : Matrix (Fin n) (Fin n) ℂ)) Q ∈ ValC G lam

theorem **trN_inj** (D : D7) (i : ℕ) (hi : i ≤ D.R.m) (γ0 : GL (Fin (D.R.N i)) D.host.E) : LinearMap.ker (trN D i hi γ0) = ⊥

theorem **ker_phiC** (S : StepIn M i hi G) (β : S.Ch) (γ0 : GL (Fin (D.R.N i)) D.host.E) : LinearMap.ker (phiC S β γ0) = ⊥

theorem **phiC_mem_ValC** (S : StepIn M i hi G) (β : S.Ch) (hV : EvValLe S β) (γ0 : GL (Fin (D.R.N i)) D.host.E) {Q : XR (Hc S β).P ℂ} (hQ : Q ∈ VtupR (Hc S β).P ℂ) : phiC S β γ0 Q ∈ Formalism.L2.Forms.ValC (Formalism.L2.Forms.gramC (gramN D i hi) (embT D)) (lamTG M i)

theorem **cmpLiftOf_of** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hD : VtupDesc S e) (hV : ∀ β, EvValLe S β) (hI : CmpInvOf S e) : CmpLiftOf S e

theorem **lin_ext_map** {σ : Type} (φ : MvPolynomial σ LL →ₗ[LL] LL) (P : MvPolynomial σ LL) : (MvPolynomial.basisMonomials σ K).constr K (fun m : σ →₀ ℕ => e (φ (MvPolynomial.monomial m 1))) (MvPolynomial.map e P) = e (φ P)

theorem **desc_span** {σ : Type} (V : Submodule LL (MvPolynomial σ LL)) {Q : MvPolynomial σ LL} (hQ : MvPolynomial.map e Q ∈ Submodule.span K (MvPolynomial.map e '' (V : Set (MvPolynomial σ LL)))) : Q ∈ V

theorem **dw_desc** {r n : ℕ} (ν : Fin n → ℕ) {f : MvPolynomial (Fin r × Fin n) LL} (h : MvPolynomial.map e f ∈ Formalism.GradedWeylDivisibility.dualWeyl K r n ν) : f ∈ Formalism.GradedWeylDivisibility.dualWeyl LL r n ν

theorem **dw_span** {r n : ℕ} (ν : Fin n → ℕ) : Formalism.GradedWeylDivisibility.dualWeyl K r n ν ≤ Submodule.span K (MvPolynomial.map e '' (Formalism.GradedWeylDivisibility.dualWeyl LL r n ν : Set (MvPolynomial (Fin r × Fin n) LL)))

theorem **span_prod_le** {S : Type} [Fintype S] {σ : S → Type} {τ : Type} (f : ∀ s, σ s → τ) (Ms : ∀ s, Submodule K (MvPolynomial (σ s) K)) (Ns : ∀ s, Submodule LL (MvPolynomial (σ s) LL)) (hM : ∀ s, Ms s ≤ Submodule.span K (MvPolynomial.map e '' (Ns s : Set (MvPolynomial (σ s) LL)))) : Submodule.span K {Z | ∃ F : ∀ s, MvPolynomial (σ s) K, (∀ s, F s ∈ Ms s) ∧ Z = ∏ s, MvPolynomial.rename (f s) (F s)} ≤ Submodule.span K (MvPolynomial.map e '' (Submodule.span LL {Z | ∃ F : ∀ s, MvPolynomial (σ s) LL, (∀ s, F s ∈ Ns s) ∧ Z = ∏ s, MvPolynomial.rename (f s) (F s)} : Set (MvPolynomial τ LL)))

theorem **vsp_span** {n' m : ℕ} (lam0 : Fin n' → Fin (m + 1) → ℤ) : Vsp K lam0 ≤ Submodule.span K (MvPolynomial.map e '' (Vsp LL lam0 : Set (MvPolynomial (Fin n' × (Fin (m + 1) × Fin (m + 1))) LL)))

theorem **vtup_span** {p : ℕ} {O Kf : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Field Kf] [Algebra O Kf] (P : Places p O Kf) : VtupR P K ≤ Submodule.span K (MvPolynomial.map e '' (VtupR P LL : Set (XR P LL)))

theorem **span_prod_map** {S : Type} [Fintype S] {σ : S → Type} {τ : Type} (f : ∀ s, σ s → τ) (Ns : ∀ s, Submodule LL (MvPolynomial (σ s) LL)) (Ms : ∀ s, Submodule K (MvPolynomial (σ s) K)) (h : ∀ s, ∀ x ∈ Ns s, MvPolynomial.map e x ∈ Ms s) {Z : MvPolynomial τ LL} (hZ : Z ∈ Submodule.span LL {Z | ∃ F : ∀ s, MvPolynomial (σ s) LL, (∀ s, F s ∈ Ns s) ∧ Z = ∏ s, MvPolynomial.rename (f s) (F s)}) : MvPolynomial.map e Z ∈ Submodule.span K {Z | ∃ F : ∀ s, MvPolynomial (σ s) K, (∀ s, F s ∈ Ms s) ∧ Z = ∏ s, MvPolynomial.rename (f s) (F s)}

theorem **vsp_map** {n' m : ℕ} (lam0 : Fin n' → Fin (m + 1) → ℤ) {x : MvPolynomial (Fin n' × (Fin (m + 1) × Fin (m + 1))) LL} (hx : x ∈ Vsp LL lam0) : MvPolynomial.map e x ∈ Vsp K lam0

theorem **vtup_map** {p : ℕ} {O Kf : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Field Kf] [Algebra O Kf] (P : Places p O Kf) {Q : XR P LL} (hQ : Q ∈ VtupR P LL) : MvPolynomial.map e Q ∈ VtupR P K

theorem **vtupDesc_holds** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) : VtupDesc S e

theorem **cmpLiftOf_of_two** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hV : ∀ β, EvValLe S β) (hI : CmpInvOf S e) : CmpLiftOf S e

theorem **tw_triv** (X : DetTw S emb c) {y0 : UPts σ J (FiniteAdeleRing (𝒪 F) F)} {γ : GL (Fin n) E} (hγ : γ ∈ UJ (σ : E →+* E) J) {κ : S.U} (h : y0.1 = bc (inclE (F

theorem **detF_tr** (γ : GL (Fin n) E) {y y' : Formalism.L2.Forms.Frames (gramC J emb)} (hy : IsTr (atInf emb γ) y' y) : detF c y = (∏ τ, (emb τ (γ : Matrix (Fin n) (Fin n) E).det) ^ c) * detF c y'

theorem **foInv_twist** (X : DetTw S emb c) {y0 : UPts σ J (FiniteAdeleRing (𝒪 F) F)} {h : Formalism.L2.Forms.Frames (gramC J emb) → ℂ} (hh : FoInv S emb y0 h) : FoInv S emb y0 (fun z => (detF c z)⁻¹ * h z)

theorem **cmpInvOf_of_raw** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hR : CmpInvRawOf S e) : CmpInvOf S e

theorem **cmpLiftOf_of_raw** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hV : ∀ β, EvValLe S β) (hR : CmpInvRawOf S e) : CmpLiftOf S e

theorem **grp2** {Γ : Type} [Group Γ] (b y k : Γ) (h : y = b * y * k) : b = y * k⁻¹ * y⁻¹

theorem **cmpInvRaw_holds** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) : CmpInvRawOf S e

theorem **cmpLiftOf_of_evVal** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (hV : ∀ β, EvValLe S β) : CmpLiftOf S e

theorem **evValLe_of_wt** (S : StepIn M i hi G) (β : S.Ch) (hw : WtTie (Hc S β).P (S.loc β).rk S.pos (M.lam i)) : EvValLe S β

theorem **cmpLiftOf_holds** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) : CmpLiftOf S e

theorem **latIn_of** {Fo : FormsIn M (j + 1) hj} {Lc : LocIn X u w hg Lv j Fo.Ch} {pos : (Σ v : Fin (Fl.Pj (j + 1)).nV, Fin ((Fl.Pj (j + 1)).n' v)) ≃ D.what₂.Tau} {β : Fo.Ch} (hfr : FrRatJ (Lc.t β) Lc.B) (H : LatTies hj w Fo β (Lc.Lb β)) : LatIn (stepInOf hj Fo Lc pos) w β where

theorem **valIntN_rec** {Fo : FormsIn M (j + 1) hj} {Lc : LocIn X u w hg Lv j Fo.Ch} {pos : (Σ v : Fin (Fl.Pj (j + 1)).nV, Fin ((Fl.Pj (j + 1)).n' v)) ≃ D.what₂.Tau} (e : D.what₂.LL →+* (D.gat u w hg).Kf) (HT : ∀ β, LocTies hj Fo Lc pos e β) (hS : BlockRationalityField) (hfrac : IsFractionRing (D.gat u w hg).O (D.gat u w hg).Kf) (hR0 : ∀ x, x ∈ intW D.what₂.places w ↔ e x ∈ Set.range (algebraMap (D.gat u w hg).O (D.gat u w hg).Kf)) (HL : ∀ β, LatTies hj w Fo β (Lc.Lb β)) : (stepInOf hj Fo Lc pos).ValIntN e w

theorem **epsU_val** (B B' γ : GL (Fin n) E) (k : ℕ) : ((epsU B B' γ k : Eˣ) : E) = (B'.1.det * (B.1.det)⁻¹) ^ k * (γ.1.det)⁻¹

theorem **det_cT** (τ : E →+* E) (M : Matrix (Fin n) (Fin n) E) : (cT τ M).det = τ M.det

theorem **epsU_conj** (hJ : IsUnit J.det) (B B' γ : GL (Fin n) E) (k : ℕ) (hγ : γ ∈ UJ (σ : E →+* E) J) (hB : cT (σ : E →+* E) B.1 * J * B.1 = 1) : σ (epsU B B' γ k : Eˣ) * (epsU B B' γ k : Eˣ) = ((cT (σ : E →+* E) B'.1 * J * B'.1).det) ^ k

theorem **epsU_iw** (S : StepLevel F E σ J T f bad) (A : LocAt F E n G) (hW : LocInt A) (hp : p.Prime) (hT : ¬ p ∣ T) (halg : ∀ v (a : (A.pl v).1.adicCompletion F), A.spl v ((1 : E) ⊗ₜ[F] a) = (a, a)) (hb : ∀ v (t : Fin n → (A.pl v).1.adicCompletion F), (∃ x ∈ Formalism.L2.Herm.locL (A.pl v).1 S.Lb, ∀ a, (A.spl v (x a)).1 = t a) ↔ ∀ a, (((bc (A.phE v) A.B)⁻¹ : GL (Fin n) ((A.pl v).1.adicCompletion F)) : Matrix (Fin n) (Fin n) ((A.pl v).1.adicCompletion F)).mulVec t a ∈ (A.pl v).1.adicCompletionIntegers F) (hJ : IsUnit J.det) (hB : cT (σ : E →+* E) A.B.1 * J * A.B.1 = 1) (Lc : Submodule (𝒪 E) (Fin n → E)) (y : UPts σ J (FiniteAdeleRing (𝒪 F) F)) (hy : ∀ x, x ∈ Formalism.L2.GlobalTheoryTheory.Comp.hat F Lc ↔ ((y.1⁻¹ : GL (Fin n) _) : Matrix (Fin n) (Fin n) _) *ᵥ x ∈ Formalism.L2.GlobalTheoryTheory.Comp.hat F S.Lb) (hle : ∀ x ∈ Lc, x ∈ dualSet (σ : E →+* E) J Lc) (B' : GL (Fin n) E) (hB' : IsLo

theorem **zpow_calc** {K : Type} [Field K] (a b : K) (c : ℤ) (k : ℕ) : (a ^ (-c)) ^ k * ((b⁻¹) ^ (-c)) ^ k = ((b * a⁻¹) ^ k) ^ c

theorem **intW_of_pow** (w : D.what₂.places.Pl) (r : D.what₂.LL) (k : ℕ) (hk : 0 < k) (h1 : r ^ k ∈ intW D.what₂.places w) (h2 : (r ^ k)⁻¹ ∈ intW D.what₂.places w) : r ∈ intW D.what₂.places w ∧ r⁻¹ ∈ intW D.what₂.places w

theorem **exists_pow** (S : StepIn M i hi G) (β : S.Ch) (cl : {c // S.chOf c = β}) : ∃ (b k : ℕ), 0 < k ∧ ∃ (g κ : UPts (sigD D) (gramN D i hi) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (γ : GL (Fin (D.R.N i)) D.host.E), γ ∈ UJ (sigD D : D.host.E →+* D.host.E) (gramN D i hi) ∧ g.1 = bc (inclE (F

theorem **twUnit_of** (S : StepIn M i hi G) (e : D.what₂.LL →+* G.Kf) (w : D.what₂.places.Pl) (β : S.Ch) (H : S.Core e β) (hR0 : ∀ x, x ∈ intW D.what₂.places w ↔ e x ∈ Set.range (algebraMap G.O G.Kf)) (L : LatIn S w β) : S.TwUnit w β

theorem **finClasses_of** (H : ∀ β, S.Core e β) (X : ChX S) : G.FinClasses

theorem **StepTies.chi** (S : StepTies hj Fo Lc Gt c) (β : Fo.Ch) (κ : (Fo.K β).lev.U) : ∃ r ∈ intW D.what₂.places w, r⁻¹ ∈ intW D.what₂.places w ∧ D.what₂.embLL r = ((Fo.K β).lev.chi κ : ℂ)

theorem **StepTies.ties** (S : StepTies hj Fo Lc Gt c) (β : Fo.Ch) : LocTiesR hj Fo Lc (c.embW u w h) β where

theorem **StepTies.lat** (S : StepTies hj Fo Lc Gt c) (β : Fo.Ch) : LatTies hj w Fo β (Lc.Lb β) where

theorem **StepTies.core** (S : StepTies hj Fo Lc Gt c) (K : CmpTied D X0 T Gt c) (ht : Fl.t = D.R.t) (hF : FillTie Gt u w h Fl) (β : Fo.Ch) : (stepInOf hj Fo Lc (Gt.gat_pos u w h)).Core (c.embW u w h) β

theorem **StepTies.valIntN** (S : StepTies hj Fo Lc Gt c) (K : CmpTied D X0 T Gt c) (ht : Fl.t = D.R.t) (hF : FillTie Gt u w h Fl) : (stepInOf hj Fo Lc (Gt.gat_pos u w h)).ValIntN (c.embW u w h) w

theorem **StepTies.twUnit** (S : StepTies hj Fo Lc Gt c) (K : CmpTied D X0 T Gt c) (ht : Fl.t = D.R.t) (hF : FillTie Gt u w h Fl) (β : Fo.Ch) : (stepInOf hj Fo Lc (Gt.gat_pos u w h)).TwUnit w β

theorem **StepTies.chX** (S : StepTies hj Fo Lc Gt c) : ChX (stepInOf hj Fo Lc (Gt.gat_pos u w h))

theorem **HypTiesI.chX** {I : CmpJIn Tw u μ w h Lv} (A : HypTiesI X0 I Gt c) (j : ℕ) (hj : j + 1 ≤ D.R.m) : ChX (I.at' j hj).st

theorem **HypTiesI.core** {I : CmpJIn Tw u μ w h Lv} (A : HypTiesI X0 I Gt c) (K : CmpTied D X0 T Gt c) (j : ℕ) (hj : j + 1 ≤ D.R.m) : ∀ β, (I.at' j hj).st.Core (c.embW u w h) β

theorem **HypTiesI.fin** {I : CmpJIn Tw u μ w h Lv} (A : HypTiesI X0 I Gt c) (K : CmpTied D X0 T Gt c) (j : ℕ) (hj : j + 1 ≤ D.R.m) : (gatJ Lv (j + 1)).FinClasses

theorem **HypTiesI.hyp** {I : CmpJIn Tw u μ w h Lv} (A : HypTiesI X0 I Gt c) (K : CmpTied D X0 T Gt c) : CmpJHyp c I

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CardStabDvdAutU

theorem **mem_autU** {N : Submodule (𝒪 E) (Sp E n)} {γ : UF E n} : γ ∈ autU N ↔ gmap γ.1 N = N

theorem **card_dvd_card_autU** (Λ : LevFam X.E X.K n SG) (θ : LevQFam X.E X.K n Q) (p : DecoF X.E n ST SG Q) (H : Subgroup (FrmG Λ θ)) (hH : ∀ g ∈ H, g • p = p) : Nat.card H ∣ Nat.card (autU p.deco.1.N)

theorem **frmMem_gsd** : (frmMem W sel mu L u Λ θ).gsd = frmGSD (colValF W sel mu u R.m) Λ θ

theorem **cplMem_gsd** : (cplMem W sel mu cst L u m2 b hY).gsd = cplGSD (midGSD W sel mu L u m2 b) (ybCpl W sel mu cst L u m2 b hY)

theorem **levPt_smul** (i : ℕ) (β : (levFam W sel mu cst L u m2 b hY Λ 0 i).β) (g : (levFam W sel mu cst L u m2 b hY Λ 0 i).G β) (d : (levFam W sel mu cst L u m2 b hY Λ 0 i).D β) : levPt i β (g • d) = levGrp i β g • levPt i β d

theorem **levFam_rho** (i : ℕ) (β : (levFam W sel mu cst L u m2 b hY Λ 0 i).β) (g : (levFam W sel mu cst L u m2 b hY Λ 0 i).G β) : (levFam W sel mu cst L u m2 b hY Λ 0 i).ρ β g = frmRho (colValF W sel mu u i) (levG i β) (levQ i β) (levGrp i β g)

theorem **levFam_nab** (i : ℕ) (β : (levFam W sel mu cst L u m2 b hY Λ 0 i).β) (d : (levFam W sel mu cst L u m2 b hY Λ 0 i).D β) : (levFam W sel mu cst L u m2 b hY Λ 0 i).Nab β d = nabAtr X.emb (W.lamTrue u i) (levPt i β d).deco.1.Lsh

theorem **levFam_del** (i : ℕ) (β : (levFam W sel mu cst L u m2 b hY Λ 0 i).β) (d : (levFam W sel mu cst L u m2 b hY Λ 0 i).D β) : (levFam W sel mu cst L u m2 b hY Λ 0 i).Del β d = delAtr X.emb (W.lamTrue u i) (levPt i β d).deco.1.Lsh

theorem **card_stab_dvd_autU** (i : ℕ) (β : (levFam W sel mu cst L u m2 b hY Λ 0 i).β) (d : (levFam W sel mu cst L u m2 b hY Λ 0 i).D β) : Nat.card (MulAction.stabilizer ((levFam W sel mu cst L u m2 b hY Λ 0 i).G β) d) ∣ Nat.card (autU (levPt i β d).deco.1.N)

theorem **levFam_ok** (hF : ∀ j, (chF W sel mu L u j).Ok) (hT : (frmGSD (colValF W sel mu u R.m) Λ 0).Ok) (i : ℕ) : (levFam W sel mu cst L u m2 b hY Λ 0 i).Ok

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ChartBaseRepresentativePenultimateStep

theorem **isPtAd_rep** (H : AdTie A dtU a) (i : ℕ) (hi : i ≤ D.R.m) (c : (D.chTw u μ).cls i) : IsPtAd A a i hi ((St i hi).chOf c) ((St i hi).y c) (A.pt i hi c) where

theorem **exists_isPtAd** (H : AdTie A dtU a) (i : ℕ) (hi : i ≤ D.R.m) (β : (St i hi).Ch) : ∃ x p, IsPtAd A a i hi β x p

theorem **muOf_det** (j : ℕ) (y : Formalism.L2.Herm.unitaryOneFiniteAdeles D.sp.cm) : muOf t u μ dtU j (Matrix.GeneralLinearGroup.det y.1) = (((t.O u μ).mt j (dtU y) : ℂˣ) : ℂ)

theorem **frsF_mem** {j : ℕ} (p : ChPt (R

private theorem **mem_stdAt_iff_forall_intAt** {n : ℕ} (P : HeightOneSpectrum (𝒪 X.E)) (w : Sp X.E n) : w ∈ stdAt X.E n P ↔ ∀ l, IntAt P (w l)

theorem **sameCoset_iff** {j : ℕ} (p : ChPt (R

theorem **sameCoset_trans** {n : ℕ} {P : rowSG Pr sel mu} {x y z : Sp X.E n} (h : SameCoset sel mu cst P x y) (h' : SameCoset sel mu cst P y z) : SameCoset sel mu cst P x z

theorem **sameCoset_symm** {n : ℕ} {P : rowSG Pr sel mu} {x y : Sp X.E n} (h : SameCoset sel mu cst P x y) : SameCoset sel mu cst P y x

theorem **mem_of_ybSet** {j : ℕ} (p : ChPt (R

theorem **chartBaseRepresentative_penultimate_step** (β : (St (t.m2 + 1) (cpl_lt t μ).le).Ch) (P : SB t μ) : y0At A (t.m2 + 1) (cpl_lt t μ).le β P = A.y0 β P

theorem **cosetOf_rep** (H : AdTie A dtU a) (c : (D.chTw u μ).cls (t.m2 + 1)) : cosetOf A (t.m2 + 1) (cpl_lt t μ).le ((St _ (cpl_lt t μ).le).chOf c) (A.pt _ (cpl_lt t μ).le c) = A.yb c

theorem **xiPt_rep** (H : AdTie A dtU a) (i : ℕ) (hi : i + 1 ≤ D.R.m) (c : (D.chTw u μ).cls i) (c' : (D.chTw u μ).cls (i + 1)) (z : HomE t.X.E (D.R.N i) (D.R.N (i + 1))) : t.seam.σ₀ (xiPt A i hi ((St i _).chOf c) ((St (i + 1) hi).chOf c') (A.pt i (Nat.le_of_succ_le hi) c) (A.pt (i + 1) hi c') z) = (t.rd u μ).Xi i c c' z

theorem **xiAd_rep** (H : AdTie A dtU a) (i : ℕ) (hi : i + 1 ≤ D.R.m) (hA : PtAway A a i hi) (c : (D.chTw u μ).cls i) (c' : (D.chTw u μ).cls (i + 1)) (z : HomE D.host.E (D.R.N i) (D.R.N (i + 1))) : xiAd A a i hi ((St i _).chOf c) ((St (i + 1) hi).chOf c') ((St i _).y c) ((St (i + 1) hi).y c') z = (t.rd u μ).Xi i c c' (seamHom t.seam (inFr (toStdCo (D.sp.cm.sig : D.host.E →+* D.host.E) (gramN D i (Nat.le_of_succ_le hi)) (A.crd i (Nat.le_of_succ_le hi))) (toStdCo (D.sp.cm.sig : D.host.E →+* D.host.E) (gramN D (i + 1) hi) (A.crd (i + 1) hi)) z))

theorem **exists_det_eq** {B : Type} [CommRing B] {n : ℕ} (σ : B →+* B) (J : Matrix (Fin n) (Fin n) B) (hJ : IsUnit J.det) (g : GL (Fin n) B) (hg : g ∈ UJ σ J) : ∃ y ∈ UJ σ (1 : Matrix (Fin 1) (Fin 1) B), Matrix.GeneralLinearGroup.det y = Matrix.GeneralLinearGroup.det g

theorem **adLawAt_of_clauses** (Hp : AdPins A dtU) (H : AdTie A dtU a) (hE : PtEx A a) (hA : ∀ i hi, PtAway A a i hi) (hL : ∀ i hi, LevAt A dtU a i hi) (hR : RatLaw A a) (hS : SizeLaw A a) (hM : MuLaw t u μ dtU) (hI : ∀ i hi β β' x x' z, IntOff (t.O u μ).S (xiAd A a i hi β β' x x' z)) : AdLawAt A dtU a (xiAd A a) (muOf t u μ dtU) where

theorem **adLawOfTie_of_clauses** (hE : UnderTie fun A _ a => PtEx A a) (hA : UnderTie fun A _ a => ∀ i hi, PtAway A a i hi) (hL : UnderTie fun A dtU a => ∀ i hi, LevAt A dtU a i hi) (hR : UnderTie fun A _ a => RatLaw A a) (hS : UnderTie fun A _ a => SizeLaw A a) (hM : UnderTie fun {_ _ _ _ _ _ _ _ _} t u μ _} _ dtU _ => MuLaw t u μ dtU) (hI : UnderTie fun {_ _ _ _ _ _ _ _ _} t u μ _} A _ a => ∀ i hi β β' x x' z, IntOff (t.O u μ).S (xiAd A a i hi β β' x x' z)) : AdLawOfTie

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ChartTies3Site

theorem **formsLev_re** {a b : ℕ} (h : a = b) (σ : E →ₐ[F] E) (J : Matrix (Fin a) (Fin a) E) (J' : Matrix (Fin b) (Fin b) E) (hJ : ∀ x y, J' x y = J (Fin.cast h.symm x) (Fin.cast h.symm y)) {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 F))} (S : Formalism.L2.Forms.StepLevel F E σ J' T f bad) (v : HeightOneSpectrum (𝒪 F)) : ((S.Uloc v).comap (reGL h _).toMonoidHom ≤ levelVL σ J v (reL h.symm S.Lb)) ∧ ((T : 𝒪 F) ∉ v.asIdeal → SelfDualAt σ J v (reL h.symm S.Lb)) ∧ (v ∉ bad → (T : 𝒪 F) ∉ v.asIdeal → levelVL σ J v (reL h.symm S.Lb) ≤ (S.Uloc v).comap (reGL h _).toMonoidHom)

theorem **hypTJC_site** (Z : SiteD X0 u w hg M) (Fl : Fills (D.gat u w hg)) (ht : Fl.t = D.R.t) (hT : ∀ i hi, ¬ D.what₂.places.resChar w ∣ (Z.Fo i hi).T) (i : ℕ) (hi : i + 1 ≤ D.R.m) (β : (Z.Fo i hi).Ch) : HypTJC ((Z.loc Fl ht i hi).t β)

theorem **hypTJ_site** (Z : SiteD X0 u w hg M) (Fl : Fills (D.gat u w hg)) (ht : Fl.t = D.R.t) (hT : ∀ i hi, ¬ D.what₂.places.resChar w ∣ (Z.Fo i hi).T) (hbad : ∀ i hi, ∀ w' ∈ (Z.Fo i hi).bad, ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∉ w'.asIdeal) (i : ℕ) (hi : i + 1 ≤ D.R.m) (β : (Z.Fo i hi).Ch) : HypTJ ((Z.loc Fl ht i hi).t β)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ChiFinOfGlevH

theorem **unit_of_root** {O K : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Field K] [Algebra O K] [IsFractionRing O K] (x : K) (n : ℕ) (hn : 0 < n) (hx : x ^ n = 1) : ∃ o : Oˣ, algebraMap O K (o : O) * x = 1

theorem **exists_char** {G₀ LL O K : Type} [Group G₀] [Field LL] [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Field K] [Algebra O K] [IsFractionRing O K] (emb : LL →+* ℂ) (e : LL →+* K) (χ : G₀ →* ℂˣ) (hrat : ∀ a, ∃ r : LL, emb r = (χ a : ℂ)) (hfin : ∃ n : ℕ, 0 < n ∧ ∀ a, χ a ^ n = 1) : ∃ ψ : G₀ →* Oˣ, ∀ a (r : LL), emb r = (χ a : ℂ) → algebraMap O K (ψ a : O) * e r = 1

theorem **kap_over** (Fo : FormsIn M (i + 1) hi) (β : Fo.Ch) (p : ℕ) (a : awayG D.sp.cm.sig (D.sp.X (i + 1)).gram (ulocOf Fo β) p) (v : HeightOneSpectrum (𝒪 D.host.F)) (hv : ((p : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal) : bc (prV (E

theorem **kap_off** (Fo : FormsIn M (i + 1) hi) (β : Fo.Ch) (p : ℕ) (a : awayG D.sp.cm.sig (D.sp.X (i + 1)).gram (ulocOf Fo β) p) (v : HeightOneSpectrum (𝒪 D.host.F)) (hv : ((p : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal) : bc (prV (E

theorem **chi_kap** (Fo : FormsIn M (i + 1) hi) (β : Fo.Ch) (p : ℕ) (a : awayG D.sp.cm.sig (D.sp.X (i + 1)).gram (ulocOf Fo β) p) (κ : (Fo.K β).lev.U) (hκp : ∀ v : HeightOneSpectrum (𝒪 D.host.F), ((p : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal → bc (prV (E

theorem **psiEx_of** (Fo : FormsIn M (i + 1) hi) (e : D.what₂.LL →+* (D.gat u w hg).Kf) (hfr : IsFractionRing (D.gat u w hg).O (D.gat u w hg).Kf) (hrat : ∀ β, ChiRatAll (Fo.K β).lev D.what₂.embLL) (hfin : ∀ β, ∃ n : ℕ, 0 < n ∧ ∀ κ, (Fo.K β).lev.chi κ ^ n = 1) : PsiEx Fo e

theorem **finite_glk** (v : HeightOneSpectrum (𝒪 F)) : Finite (GL (Fin n) (kv F v))

theorem **chi_fin_of_glevH** {S : StepLevel F E σ J T f bad} {G : GLoc F E σ gp} {H : ∀ (a : Fin 2) (v : HeightOneSpectrum (𝒪 F)), (gp a : 𝒪 F) ∈ v.asIdeal → Subgroup (GL (Fin n) (kv F v))} {lam : ∀ a v h, H a v h →* ℂˣ} (L : GLevH S G H lam) (hgp : ∀ a, gp a ≠ 0) : ∃ m : ℕ, 0 < m ∧ ∀ κ, S.chi κ ^ m = 1

theorem **chi_fin_top** (F : FormsTiedSym D u Ch chOf) (j : ℕ) (hj : j ≤ D.R.m) (h : j = D.R.m) : ∃ m : ℕ, 0 < m ∧ ∀ κ, (F.cls j hj (Or.inr h)).lev.chi κ ^ m = 1

theorem **finMem_of** (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) : ∀ β, ∃ m : ℕ, 0 < m ∧ ∀ κ, ((formsInMem F dt i hi).K β).lev.chi κ ^ m = 1

theorem **chiRat_of_glevH** {S : StepLevel F E σ J T f bad} {G : GLoc F E σ gp} {H : ∀ (a : Fin 2) (v : HeightOneSpectrum (𝒪 F)), (gp a : 𝒪 F) ∈ v.asIdeal → Subgroup (GL (Fin n) (kv F v))} {lam : ∀ a v h, H a v h →* ℂˣ} (L : GLevH S G H lam) (emb : LL →+* ℂ) (hl : ∀ a v h (M : H a v h), ∃ k : LL, emb k = ((lam a v h M : ℂˣ) : ℂ)) : ChiRatAll S emb

theorem **chiRat_top** (F : FormsTiedSym D u Ch chOf) (hM : LamMRat F) (j : ℕ) (hj : j ≤ D.R.m) (h : j = D.R.m) : ChiRatAll (F.cls j hj (Or.inr h)).lev D.what₂.embLL

theorem **ratMem_of** (F : FormsTiedSym D u Ch chOf) (hC : LamCRat F) (hM : LamMRat F) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) : ∀ β, ChiRatAll ((formsInMem F dt i hi).K β).lev D.what₂.embLL

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CoeEqAlgebraMapZeroEquiv

theorem **formRes_smul** (s : X ⟶ Sp k) {U V : X.Opens} (h : V ≤ U) {j : ℕ} (b : Sec s U) (x : OmU s U j) : formRes s h j (b • x) = Sec.res s h b • formRes s h j x

theorem **formRes_refl** (s : X ⟶ Sp k) (U : X.Opens) {j : ℕ} (x : OmU s U j) : formRes s (le_refl U) j x = x

theorem **formRes_formRes** (s : X ⟶ Sp k) {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) {j : ℕ} (x : OmU s U j) : formRes s h' j (formRes s h j x) = formRes s (h'.trans h) j x

theorem **appLE_comp_app** {Y : Scheme.{0}} (g : X ⟶ Y) (f : X' ⟶ X) (U : X.Opens) (V : Y.Opens) (e : U ≤ g ⁻¹ᵁ V) (e' : f ⁻¹ᵁ U ≤ (f ≫ g) ⁻¹ᵁ V) : g.appLE V U e ≫ f.app U = (f ≫ g).appLE V (f ⁻¹ᵁ U) e'

theorem **semiHom_appSec** (hf : f ≫ s = s') (U : X.Opens) : SemiHom (RingHom.id k) (appSec s s' f U)

theorem **formPull_smul** (hf : f ≫ s = s') (U : X.Opens) {j : ℕ} (b : Sec s U) (x : OmU s U j) : formPull hf U j (b • x) = appSec s s' f U b • formPull hf U j x

theorem **formRes_formPull** (hf : f ≫ s = s') {U V : X.Opens} (h : V ≤ U) {j : ℕ} (x : OmU s U j) : formRes s' (f.preimage_mono h) j (formPull hf U j x) = formPull hf V j (formRes s h j x)

theorem **appTop_algebraMap** (hf : f ≫ s = s') (r : k) : letI

theorem **omegaPull_toOmega** (hf : f ≫ s = s') (j : ℕ) (U : X.Opens) (x : OmU s U j) : (omegaPull hf j).val.app (op U) ((toOmega s j).app (op U) x) = (toOmega s' j).app (op (f ⁻¹ᵁ U)) (formPull hf U j x)

theorem **coe_eq_algebraMap_zeroEquiv** (x : Om k B 0) : (x : DR k B) = algebraMap B (DR k B) (exteriorPower.zeroEquiv B Ω[B⁄k] x)

theorem **zeroEquiv_mOm** {k' B' : Type} [CommRing k'] [CommRing B'] [Algebra k' B'] {σ : k →+* k'} {α : B →+* B'} (h : SemiHom σ α) (x : Om k B 0) : exteriorPower.zeroEquiv B' Ω[B'⁄k'] (h.mOm 0 x) = α (exteriorPower.zeroEquiv B Ω[B⁄k] x)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CohomZeroEquivContinuous

theorem **galMap_geomMap** {X Y : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) (q : Y ⟶ Spec (CommRingCat.of k)) (u : X ⟶ Y) (hu : u ≫ q = p) (σ : Ω ≃ₐ[k] Ω) : galMap Ω p σ ≫ geomMap Ω p u hu = geomMap Ω p u hu ≫ galMap Ω q σ

theorem **gal_pull** {X Y : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) (q : Y ⟶ Spec (CommRingCat.of k)) (u : X ⟶ Y) (hu : u ≫ q = p) (σ : Ω ≃ₐ[k] Ω) (n : ℕ) (x : H R n (geom Ω q)) : gal R Ω p n σ (pull R n (geomMap Ω p u hu) x) = pull R n (geomMap Ω p u hu) (gal R Ω q n σ x)

theorem **coefPre_isSheaf** : Presheaf.IsSheaf Scheme.proetaleTopology.{0} (coefPre R)

theorem **cohomZero_nontrivial** (X : Scheme.{0}) [Nonempty X] [Nontrivial R] : Nontrivial (H R 0 X)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CohomZeroEquivSections

theorem **isLisse_pi_coef** (X : Scheme.{0}) (r : ℕ) : IsLisse (∏ᶜ fun _ : Fin r => coef R X)

theorem **isLisse_coef** (X : Scheme.{0}) : IsLisse (coef R X)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CompPowTate

theorem **Datum.gPts_dot** (D : Datum p) {Λ : Type} [CommRing Λ] (g : GL D.Ix Λ) (ν : Λˣ) (hg : (g : Matrix D.Ix D.Ix Λ)ᵀ * D.gramEF.map (Int.castRingHom Λ) * g = (ν : Λ) • D.gramEF.map (Int.castRingHom Λ)) (v w : D.Ix → Λ) : ((g : Matrix D.Ix D.Ix Λ) *ᵥ v) ⬝ᵥ (D.gramEF.map (Int.castRingHom Λ) *ᵥ ((g : Matrix D.Ix D.Ix Λ) *ᵥ w)) = (ν : Λ) * (v ⬝ᵥ (D.gramEF.map (Int.castRingHom Λ) *ᵥ w))

theorem **comp_pow_tate** (u : T₀ ⟶ T) (x : T ⟶ A) (n : ℕ) : u ≫ x ^ n = (u ≫ x) ^ n

theorem **Obj.IsTateL.comp** {o : Obj b N s} {ℓ : ℕ} [Fact ℓ.Prime] {T T' : Over S} {ev : ℕ → (D.Ix → ℤ_[ℓ]) → (T ⟶ o.A)} (h : o.IsTateL ℓ T ev) (u : T' ⟶ T) : o.IsTateL ℓ T' (fun m v => u ≫ ev m v)

theorem **Obj.IsTateL.smul** {o : Obj b N s} {ℓ : ℕ} [Fact ℓ.Prime] {T : Over S} {ev : ℕ → (D.Ix → ℤ_[ℓ]) → (T ⟶ o.A)} (h : o.IsTateL ℓ T ev) (g : D.gPts ℤ_[ℓ]) : o.IsTateL ℓ T (fun m v => ev m (((g : GL D.Ix ℤ_[ℓ]) : Matrix D.Ix D.Ix ℤ_[ℓ]) *ᵥ v))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ConeClZeroBijective

theorem **eq_of_zsmul_add** {A : Type*} [AddCommGroup A] {x y : A} (h : (1 : ℤ) • x + (-1 : ℤ) • y = 0) : x = y

theorem **zsmul_add_cancel** {A : Type*} [AddCommGroup A] (x : A) : (1 : ℤ) • x + (-1 : ℤ) • x = 0

theorem **map_cochain_congr** (M : C ⥤ ModuleCat.{0} R) (f : CatH.Cn M 0) {t t' : ComposableArrows C 0} (h : t = t') {Y : C} (u : t.obj (Fin.last 0) ⟶ Y) (u' : t'.obj (Fin.last 0) ⟶ Y) (hu : HEq u u') : (M.map u).hom (f t) = (M.map u').hom (f t')

theorem **compat_of_d_eq_zero** (M : C ⥤ ModuleCat.{0} R) {f : CatH.Cn M 0} (hf : CatH.d M 0 f = 0) {T T' : C} (φ : T ⟶ T') : (M.map φ).hom (f (ComposableArrows.mk₀ T)) = f (ComposableArrows.mk₀ T')

theorem **BF_self** : ∀ n : ℕ, BF F n n = ⊥ | 0 => rfl | n + 1 => by change (TotF F (n + 1) n).map _ = ⊥ rw [TotF_eq_bot F (Nat.lt_succ_self n), Submodule.map_bot] theorem HypF.toHyp_zero_bijective (n : ℕ) : Function.Bijective (HypF.toHyp F 0 n)

theorem **ofTop_same** {n : ℕ} (c : Term F 0 n) (h : 0 + n = n) : ofTop F n c ⟨(0, n), h⟩ = c

theorem **ofTop_mem_TotF** {n : ℕ} (c : Term F 0 n) : ofTop F n c ∈ TotF F n n

theorem **eq_ofTop** {n : ℕ} {c : Tot F n} (hc : c ∈ TotF F n n) : c = ofTop F n (c ⟨(0, n), zero_add n⟩)

theorem **ofTop_mem_ZF** {n : ℕ} {c : Term F 0 n} (h1 : dH F 0 n c = 0) (h2 : dV F 0 n c = 0) : ofTop F n c ∈ ZF F n n

theorem **dTot_top** {n : ℕ} (c : Tot F n) (h : 0 + (n + 1) = n + 1) : dTot F n c ⟨(0, n + 1), h⟩ = dV F 0 n (c ⟨(0, n), zero_add n⟩)

theorem **coneTerm_apply** {q : ℕ} (ω : Om R B q) (s : ComposableArrows C 0) : coneTerm e q ω s = (SemiHom.of_algHom (e (s.obj (Fin.last 0)))).mOm q ω

theorem **dH_coneTerm** {q : ℕ} (ω : Om R B q) : dH F 0 q (coneTerm e q ω) = 0

theorem **dV_coneTerm** {q : ℕ} (ω : Om R B q) : dV F 0 q (coneTerm e q ω) = coneTerm e (q + 1) (dOm R B q ω)

theorem **coneCl_mem_fil** {n : ℕ} (ω : LinearMap.ker (dOm R B n)) : coneCl e he n ω ∈ Hyp.fil F n n

theorem **coneCl_dOm** {n : ℕ} (η : Om R B n) : coneCl e he (n + 1) ⟨dOm R B n η, dOm_dOm η⟩ = 0

theorem **backOm_coneTerm** {q : ℕ} (ω : Om R B q) : backOm e hb q (coneTerm e q ω (ComposableArrows.mk₀ T₀)) = ω

theorem **coneCl_eq_zero_iff** {n : ℕ} (ω : LinearMap.ker (dOm R B (n + 1))) : coneCl e he (n + 1) ω = 0 ↔ ∃ η, dOm R B n η = ω.1

theorem **coneF_bijective** (h : Limits.IsInitial T₀) (n : ℕ) : Function.Bijective (coneF e he n)

theorem **coneCl_zero_bijective** (h : Limits.IsInitial T₀) : Function.Bijective (coneCl e he 0)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CuspFormLimitTwoPlaces

theorem **sameWtOff_of_eq** (ι : Qbar l ≃+* ℂ) (w₀ v' : HeightOneSpectrum (𝒪 E)) (U U' : Submodule ℂ (UA E a → ℂ)) (h : ∀ τ lam, HasWeightAt U τ lam ↔ HasWeightAt U' τ lam) : SameWtOff ι w₀ v' U U'

theorem **sameWtAt_of_off** (ι : Qbar l ≃+* ℂ) {w₀ v' w : HeightOneSpectrum (𝒪 E)} {U U' : Submodule ℂ (UA E a → ℂ)} (h : SameWtOff ι w₀ v' U U') (hw : ∀ τ : E →+* ℂ, OverPl ι τ w → ∀ u, OverPl ι τ u → u = w) (h₀ : Apart w w₀) (h' : Apart w v') : SameWtAt ι w U U'

theorem **SeedWU2.mono** {Ty : UnitaryAutomorphicTypeClass} {Gen : ℕ → ℕ → Multiset ℂˣ → Prop} {Wt Wt' : SeedWtU} (h : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (l : ℕ) [Fact l.Prime] (a : ℕ) (ι : Qbar l ≃+* ℂ) (w₀ v' : HeightOneSpectrum (𝒪 E)) (U U' : Submodule ℂ (UA E a → ℂ)), Wt' E l a ι w₀ v' U U' → Wt E l a ι w₀ v' U U') (hS : SeedWU2 Ty Gen Wt) : SeedWU2 Ty Gen Wt'

theorem **CongUW2.mono** {Ty : UnitaryAutomorphicTypeClass} {Gen : ℕ → ℕ → Multiset ℂˣ → Prop} {Wt Wt' : SeedWtU} (h : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (l : ℕ) [Fact l.Prime] (a : ℕ) (ι : Qbar l ≃+* ℂ) (w₀ v' : HeightOneSpectrum (𝒪 E)) (U U' : Submodule ℂ (UA E a → ℂ)), Wt' E l a ι w₀ v' U U' → Wt E l a ι w₀ v' U U') (hC : CongUW2 Ty Gen Wt') : CongUW2 Ty Gen Wt

theorem **SeedWU2.of_any** {Ty : UnitaryAutomorphicTypeClass} {Gen : ℕ → ℕ → Multiset ℂˣ → Prop} (Wt : SeedWtU) (hS : SeedWU2 Ty Gen) : SeedWU2 Ty Gen Wt

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.CyclicStrongAtCusp

theorem **unlinked_zero** : Unlinked (0 : Multiset (IrrGL k))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.DegOneIffSurjective

theorem **_root_.GlobalTheory.HeckeSplit.unrAt_iff** (χ : _root_.Formalism.L2.Host.HChar L) (v : HeightOneSpectrum (𝒪 L)) : _root_.GlobalTheory.HeckeSplit.UnrAt χ.1 v ↔ χ.UnrAt v

theorem **_root_.GlobalTheory.HeckeSplit.trivAt_iff** (χ : _root_.Formalism.L2.Host.HChar L) (v : HeightOneSpectrum (𝒪 L)) : _root_.GlobalTheory.HeckeSplit.TrivAt χ.1 v ↔ χ.loc v = 1

@[simp] theorem **_root_.GlobalTheory.HeckeSplit.infPart_infIdele** (x : (InfiniteAdeleRing L)ˣ) : _root_.GlobalTheory.HeckeSplit.infPart L (_root_.GlobalTheory.HeckeSplit.infIdele L x) = x

@[simp] theorem **_root_.GlobalTheory.HeckeSplit.finPart_infIdele** (v : HeightOneSpectrum (𝒪 L)) (x : (InfiniteAdeleRing L)ˣ) : _root_.GlobalTheory.HeckeSplit.finPart L v (_root_.GlobalTheory.HeckeSplit.infIdele L x) = 1

@[simp] theorem **_root_.GlobalTheory.HeckeSplit.finPart_localIdele** (v : HeightOneSpectrum (𝒪 L)) (x : (v.adicCompletion L)ˣ) : _root_.GlobalTheory.HeckeSplit.finPart L v (_root_.HeckeValGen.localIdele L v x) = x

@[simp] theorem **_root_.GlobalTheory.HeckeSplit.infPart_localIdele** (v : HeightOneSpectrum (𝒪 L)) (x : (v.adicCompletion L)ˣ) : _root_.GlobalTheory.HeckeSplit.infPart L (_root_.HeckeValGen.localIdele L v x) = 1

theorem **_root_.GlobalTheory.HeckeSplit.finPart_localIdele_ne** {v w : HeightOneSpectrum (𝒪 L)} (h : w ≠ v) (x : (v.adicCompletion L)ˣ) : _root_.GlobalTheory.HeckeSplit.finPart L w (_root_.HeckeValGen.localIdele L v x) = 1

theorem **IsHeckeChar.mul** {χ ψ : Ideles L →* ℂˣ} (hχ : IsHeckeChar L χ) (hψ : IsHeckeChar L ψ) : IsHeckeChar L (χ * ψ)

theorem **IsHeckeChar.inv** {χ : Ideles L →* ℂˣ} (hχ : IsHeckeChar L χ) : IsHeckeChar L χ⁻¹

theorem **TrivAt.unrAt** {χ : Ideles L →* ℂˣ} {v : HeightOneSpectrum (𝒪 L)} (h : TrivAt χ v) : UnrAt χ v

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.DetIntOfLevel

theorem **det_stab_loc** (v : HeightOneSpectrum (𝒪 F)) (Lb : Submodule (𝒪 E) (Fin n → E)) (hfg : Lb.FG) (hsp : Submodule.span E (Lb : Set (Fin n → E)) = ⊤) (g : GL (Fin n) (E ⊗[F] v.adicCompletion F)) (hg : g ∈ stabGL (latS (F

theorem **det_int_of_level** (S : StepLevel F E σ J T f bad) (x : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (hx : x ∈ S.U) : (x : Matrix (Fin n) (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)).det ∈ intAf F E

theorem **det_norm_one** (hJ : IsUnit J.det) (x : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (hx : x ∈ UPts σ J (FiniteAdeleRing (𝒪 F) F)) : sigR σ (FiniteAdeleRing (𝒪 F) F) (x : Matrix (Fin n) (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)).det ∗ (x : Matrix (Fin n) (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)).det = 1

theorem **scalar_mem_U1Af** (d : (D.host.E ⊗[D.host.F] Af D.host.F)ˣ) (h : sigR D.sp.cm.sig (Af D.host.F) (d : D.host.E ⊗[D.host.F] Af D.host.F) ∗ d = 1) : Matrix.GeneralLinearGroup.scalar (Fin 1) d ∈ unitaryOneFiniteAdeles D.sp.cm

theorem **scalar_stab** {A : Type} [CommRing A] (R : Subring A) (d : Aˣ) (h1 : (d : A) ∈ R) (h2 : ((d⁻¹ : Aˣ) : A) ∈ R) : Matrix.GeneralLinearGroup.scalar (Fin 1) d ∈ stabGL (stdLat (n

theorem **scalar_mem_U1int** (d : (D.host.E ⊗[D.host.F] Af D.host.F)ˣ) (h : sigR D.sp.cm.sig (Af D.host.F) (d : D.host.E ⊗[D.host.F] Af D.host.F) * d = 1) (h1 : (d : D.host.E ⊗[D.host.F] Af D.host.F) ∈ intAf D.host.F D.host.E) (h2 : ((d⁻¹ : (D.host.E ⊗[D.host.F] Af D.host.F)ˣ) : D.host.E ⊗[D.host.F] Af D.host.F) ∈ intAf D.host.F D.host.E) : Matrix.GeneralLinearGroup.scalar (Fin 1) d ∈ IntegralUnitGroup D.sp.cm

theorem **det_detU** (i : ℕ) (hi : i ≤ D.R.m) (x : UPts (sigD D) (gramN D i hi) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : Matrix.GeneralLinearGroup.det (detU D i hi x : GL (Fin 1) (D.host.E ⊗[D.host.F] Af D.host.F)) = Matrix.GeneralLinearGroup.det (x : GL (Fin (D.R.N i)) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.DifferentiableAtDirD

theorem **repr_scale_self** (t : ℂ) (α : Alt n) (S : Finset (Fin (n + n))) : (bas n).repr (PullType.scale n t t α) S = t ^ S.card * (bas n).repr α S

theorem **hasDeg_iff_scale** (k : ℕ) (α : Alt n) : HasDeg k α ↔ PullType.scale n 2 2 α = (2 : ℂ) ^ k • α

theorem **HasDeg.mul** {k l : ℕ} {α β : Alt n} (hα : HasDeg k α) (hβ : HasDeg l β) : HasDeg (k + l) (α * β)

theorem **HasDeg.zero** (k : ℕ) : HasDeg k (0 : Alt n)

theorem **HasDeg.add** {k : ℕ} {α β : Alt n} (hα : HasDeg k α) (hβ : HasDeg k β) : HasDeg k (α + β)

theorem **HasDeg.smul** {k : ℕ} {α : Alt n} (c : ℂ) (hα : HasDeg k α) : HasDeg k (c • α)

theorem **HasDeg.sum** {k : ℕ} {ι : Type} {t : Finset ι} {α : ι → Alt n} (h : ∀ i ∈ t, HasDeg k (α i)) : HasDeg k (∑ i ∈ t, α i)

theorem **HasDeg.pull** {k : ℕ} {α : Alt m} (L : V n →L[ℂ] V m) (hα : HasDeg k α) : HasDeg k (pull L α)

theorem **hasDeg_bas** (S : Finset (Fin (n + n))) : HasDeg S.card (bas n S)

theorem **hasDeg_algebraMap** (c : ℂ) : HasDeg 0 (algebraMap ℂ (Alt n) c)

theorem **HasDeg.eq_algebraMap** {α : Alt n} (h : HasDeg 0 α) : α = algebraMap ℂ _ ((bas n).repr α ∅)

theorem **scale_ι** (t : ℂ) (u : Fin n ⊕ Fin n → ℂ) : PullType.scale n t t (ExteriorAlgebra.ι ℂ u) = t • ExteriorAlgebra.ι ℂ u

theorem **hasDeg_ι** (u : Fin n ⊕ Fin n → ℂ) : HasDeg 1 (ExteriorAlgebra.ι ℂ u)

theorem **hasDeg_iff_mem** (k : ℕ) (α : Alt n) : HasDeg k α ↔ α ∈ ⋀[ℂ]^k (Fin n ⊕ Fin n → ℂ)

theorem **repr_degPart** (k : ℕ) (α : Alt n) (S : Finset (Fin (n + n))) : (bas n).repr (degPart n k α) S = if S.card = k then (bas n).repr α S else 0

theorem **hasDeg_degPart** (k : ℕ) (α : Alt n) : HasDeg k (degPart n k α)

theorem **HasDeg.degPart_eq** {j : ℕ} {α : Alt n} (h : HasDeg j α) (k : ℕ) : degPart n k α = if j = k then α else 0

theorem **degPart_pull** (L : V n →L[ℂ] V m) (k : ℕ) (α : Alt m) : degPart n k (pull L α) = pull L (degPart m k α)

theorem **degPart_ι_mul** (u : Fin n ⊕ Fin n → ℂ) (k : ℕ) (α : Alt n) : degPart n (k + 1) (ExteriorAlgebra.ι ℂ u * α) = ExteriorAlgebra.ι ℂ u * degPart n k α ∧ degPart n 0 (ExteriorAlgebra.ι ℂ u * α) = 0

theorem **par_ι** (u : Fin n ⊕ Fin n → ℂ) : par n (ExteriorAlgebra.ι ℂ u) = -ExteriorAlgebra.ι ℂ u

theorem **HasDeg.par_eq** {k : ℕ} {α : Alt n} (h : HasDeg k α) : par n α = (-1 : ℂ) ^ k • α

theorem **par_par** (α : Alt n) : par n (par n α) = α

theorem **ι_mul_eq** (u : Fin n ⊕ Fin n → ℂ) (α : Alt n) : ExteriorAlgebra.ι ℂ u * α = par n α * ExteriorAlgebra.ι ℂ u

theorem **HasDeg.mul_comm** {k l : ℕ} {α β : Alt n} (hα : HasDeg k α) (hβ : HasDeg l β) : α * β = (-1 : ℂ) ^ (k * l) • (β * α)

theorem **mul_mem_holGe** {p q : ℕ} {α β : Alt n} (hα : α ∈ holGe n p) (hβ : β ∈ holGe n q) : α * β ∈ holGe n (p + q)

theorem **holCount_le** (S : Finset (Fin (n + n))) : holCount S ≤ n

theorem **dirD_sum_mul** (i : Fin n ⊕ Fin n) {ι : Type} [Fintype ι] {f : ι → V n → ℂ} (hf : ∀ p, DifferentiableAt ℝ (f p) v) (c : ι → ℂ) : dirD i (fun x => ∑ p, c p • f p x) v = ∑ p, c p • dirD i (f p) v

theorem **dirD_zero** (i : Fin n ⊕ Fin n) : dirD i (fun _ => 0) v = 0

theorem **dirD_comm** {f : V n → ℂ} (hf : ContDiffAt ℝ 2 f v) (i j : Fin n ⊕ Fin n) : dirD i (dirD j f) v = dirD j (dirD i f) v

theorem **differentiableAt_dirD** {f : V n → ℂ} (hf : ContDiffAt ℝ 2 f v) (i : Fin n ⊕ Fin n) : DifferentiableAt ℝ (dirD i f) v

theorem **repr_linear** (L : Alt n →ₗ[ℂ] Alt n) (α : Alt n) (S : Finset (Fin (n + n))) : (bas n).repr (L α) S = ∑ T, (bas n).repr (L (bas n T)) S • (bas n).repr α T

theorem **coefwise_dirD_linear** (i : Fin n ⊕ Fin n) (L : Alt n →ₗ[ℂ] Alt n) {F : V n → Alt n} (hF : ∀ S, DifferentiableAt ℝ (fun x => (bas n).repr (F x) S) v) : coefwise (fun f => dirD i f v) (fun x => L (F x)) = L (coefwise (fun f => dirD i f v) F)

theorem **opAt_opAt** (s t : Fin n → Fin n ⊕ Fin n) {F : V n → Alt n} (hF : ∀ S, ContDiffAt ℝ 2 (fun x => (bas n).repr (F x) S) v) : opAt s (opAt t F) v = ∑ a, ∑ b, ExteriorAlgebra.ι ℂ (Pi.single (s a) 1) * ExteriorAlgebra.ι ℂ (Pi.single (t b) 1) * coefwise (fun f => dirD (s a) (dirD (t b) f) v) F

theorem **opAt_anticomm** (s t : Fin n → Fin n ⊕ Fin n) {F : V n → Alt n} (hF : ∀ S, ContDiffAt ℝ 2 (fun x => (bas n).repr (F x) S) v) : opAt s (opAt t F) v + opAt t (opAt s F) v = 0

theorem **delAt_dbarAt_sq** {F : V n → Alt n} (hF : ∀ S, ContDiffAt ℝ 2 (fun x => (bas n).repr (F x) S) v) : delAt (delAt F) v = 0 ∧ dbarAt (dbarAt F) v = 0 ∧ delAt (dbarAt F) v + dbarAt (delAt F) v = 0

theorem **delAt_dbarAt_mul** {F G : V n → Alt n} (hF : ∀ S, DifferentiableAt ℝ (fun x => (bas n).repr (F x) S) v) (hG : ∀ S, DifferentiableAt ℝ (fun x => (bas n).repr (G x) S) v) : delAt (fun x => F x * G x) v = delAt F v * G v + par n (F v) * delAt G v ∧ dbarAt (fun x => F x * G x) v = dbarAt F v * G v + par n (F v) * dbarAt G v

theorem **opAt_par** (s : Fin n → Fin n ⊕ Fin n) {F : V n → Alt n} (hF : ∀ S, DifferentiableAt ℝ (fun x => (bas n).repr (F x) S) v) : opAt s (fun x => par n (F x)) v = -par n (opAt s F v)

theorem **degPart_opAt** (s : Fin n → Fin n ⊕ Fin n) (k : ℕ) {F : V n → Alt n} (hF : ∀ S, DifferentiableAt ℝ (fun x => (bas n).repr (F x) S) v) : degPart n (k + 1) (opAt s F v) = opAt s (fun x => degPart n k (F x)) v ∧ degPart n 0 (opAt s F v) = 0

theorem **hasDeg_opAt** (s : Fin n → Fin n ⊕ Fin n) {k : ℕ} {F : V n → Alt n} (hF : ∀ x, HasDeg k (F x)) : HasDeg (k + 1) (opAt s F v)

theorem **opAt_mem_holGe** (s : Fin n → Fin n ⊕ Fin n) {p : ℕ} {F : V n → Alt n} (hF : ∀ x, F x ∈ holGe n p) : opAt s F v ∈ holGe n p

@[simp] theorem **parFld_apply** (α : A.Fld) (x : A.Pt) : A.parFld α x = par n (α x)

@[simp] theorem **degFld_apply** (k : ℕ) (α : A.Fld) (x : A.Pt) : A.degFld k α x = degPart n k (α x)

@[simp] theorem **mem_forms** {α : A.Fld} : α ∈ A.forms ↔ A.IsForm α

theorem **loc_linear** (L : Alt n →ₗ[ℂ] Alt n) (α : A.Fld) (i : A.ι) : A.loc (fun x => L (α x)) i = fun v => L (A.loc α i v)

theorem **IsForm.linear** {α : A.Fld} (hα : A.IsForm α) (L : Alt n →ₗ[ℂ] Alt n) (hL : ∀ (T : V n →L[ℂ] V n) (β : Alt n), L (pull T β) = pull T (L β)) : A.IsForm fun x => L (α x)

theorem **IsForm.parFld** {α : A.Fld} (hα : A.IsForm α) : A.IsForm (A.parFld α)

theorem **IsForm.degFld** {α : A.Fld} (hα : A.IsForm α) (k : ℕ) : A.IsForm (A.degFld k α)

theorem **IsForm.dExt** {α : A.Fld} (hα : A.IsForm α) : A.IsForm (A.dExt α)

theorem **dExt_add** {α β : A.Fld} (hα : A.IsForm α) (hβ : A.IsForm β) : A.dExt (α + β) = A.dExt α + A.dExt β

theorem **dExt_smul** (c : ℂ) {α : A.Fld} (hα : A.IsForm α) : A.dExt (c • α) = c • A.dExt α

@[simp] theorem **coe_dForms** (α : A.forms) : (A.dForms α : A.Fld) = A.dExt α

@[simp] theorem **coe_parForms** (α : A.forms) : (A.parForms α : A.Fld) = A.parFld α

@[simp] theorem **dExt_zero** : A.dExt 0 = 0

@[simp] theorem **dExt_one** : A.dExt 1 = 0

theorem **loc_op_eventuallyEq** (T : (V n → Alt n) → V n → Alt n) (α : A.Fld) (x : A.Pt) : A.loc (fun y => T (A.loc α y.1.1) y.crd) x.1.1 =ᶠ[𝒩 x.crd] T (A.loc α x.1.1)

theorem **del_dbar_sq** {α : A.Fld} (hα : A.IsForm α) : A.del (A.del α) = 0 ∧ A.dbar (A.dbar α) = 0 ∧ A.del (A.dbar α) + A.dbar (A.del α) = 0

theorem **dExt_dExt** {α : A.Fld} (hα : A.IsForm α) : A.dExt (A.dExt α) = 0

theorem **del_dbar_mul** {α β : A.Fld} (hα : A.IsForm α) (hβ : A.IsForm β) : A.del (α ∗ β) = A.del α ∗ β + A.parFld α ∗ A.del β ∧ A.dbar (α ∗ β) = A.dbar α ∗ β + A.parFld α ∗ A.dbar β

theorem **dExt_mul** {α β : A.Fld} (hα : A.IsForm α) (hβ : A.IsForm β) : A.dExt (α ∗ β) = A.dExt α ∗ β + A.parFld α ∗ A.dExt β

theorem **dExt_apply** (α : A.Fld) (x : A.Pt) : A.dExt α x = opAt .inl (A.loc α x.1.1) x.crd + opAt .inr (A.loc α x.1.1) x.crd

theorem **dExt_parFld** {α : A.Fld} (hα : A.IsForm α) : A.dExt (A.parFld α) = -A.parFld (A.dExt α)

theorem **degFld_dExt** {α : A.Fld} (hα : A.IsForm α) (k : ℕ) : A.degFld (k + 1) (A.dExt α) = A.dExt (A.degFld k α) ∧ A.degFld 0 (A.dExt α) = 0

theorem **isDga** : GlobalTheory.PD.IsDga A.parForms A.dForms where

theorem **loc_prop** {P : Alt n → Prop} (h₀ : P 0) {α : A.Fld} (hα : ∀ x, P (α x)) (i : A.ι) (v : V n) : P (A.loc α i v)

theorem **IsDeg.dExt** {k : ℕ} {α : A.Fld} (hα : A.IsDeg k α) : A.IsDeg (k + 1) (A.dExt α)

theorem **IsDeg.mul** {k l : ℕ} {α β : A.Fld} (hα : A.IsDeg k α) (hβ : A.IsDeg l β) : A.IsDeg (k + l) (α ∗ β)

theorem **isDeg_degFld** (k : ℕ) (α : A.Fld) : A.IsDeg k (A.degFld k α)

theorem **IsDeg.degFld_eq** {k : ℕ} {α : A.Fld} (hα : A.IsDeg k α) : A.degFld k α = α

theorem **IsDeg.parFld_eq** {k : ℕ} {α : A.Fld} (hα : A.IsDeg k α) : A.parFld α = (-1 : ℂ) ^ k • α

theorem **IsDeg.mul_comm** {k l : ℕ} {α β : A.Fld} (hα : A.IsDeg k α) (hβ : A.IsDeg l β) : α ∗ β = (-1 : ℂ) ^ (k ∗ l) • (β ∗ α)

theorem **typeGe_iff** {p : ℕ} {α : A.Fld} : A.TypeGe p α ↔ ∀ x, α x ∈ holGe n p

theorem **TypeGe.mul** {p q : ℕ} {α β : A.Fld} (hα : A.TypeGe p α) (hβ : A.TypeGe q β) : A.TypeGe (p + q) (α ∗ β)

theorem **TypeGe.dExt** {p : ℕ} {α : A.Fld} (hα : A.TypeGe p α) : A.TypeGe p (A.dExt α)

theorem **TypeGe.eq_zero** {p : ℕ} {α : A.Fld} (hα : A.TypeGe p α) (hp : n < p) : α = 0

theorem **isHolo_id** : A.IsHolo A id

theorem **IsHolo.comp** {f : M' → M} {g : M'' → M'} (hf : A.IsHolo A' f) (hg : A'.IsHolo A'' g) : A.IsHolo A'' (f ∘ g)

@[simp] theorem **pullbackAlg_apply** (f : M' → M) (α : A.Fld) : A.pullbackAlg A' f α = A.pullback A' f α

theorem **IsForm.pullback** {f : M' → M} (hf : A.IsHolo A' f) {α : A.Fld} (hα : A.IsForm α) : A'.IsForm (A.pullback A' f α)

theorem **dExt_pullback** {f : M' → M} (hf : A.IsHolo A' f) {α : A.Fld} (hα : A.IsForm α) : A'.dExt (A.pullback A' f α) = A.pullback A' f (A.dExt α)

theorem **IsDeg.pullback** (f : M' → M) {k : ℕ} {α : A.Fld} (hα : A.IsDeg k α) : A'.IsDeg k (A.pullback A' f α)

theorem **TypeGe.pullback** (f : M' → M) {p : ℕ} {α : A.Fld} (hα : A.TypeGe p α) : A'.TypeGe p (A.pullback A' f α)

theorem **pullback_id** {α : A.Fld} (hα : A.IsForm α) : A.pullback A id α = α

theorem **pullback_comp** {f : M' → M} {g : M'' → M'} (hf : A.IsHolo A' f) (hg : A'.IsHolo A'' g) (α : A.Fld) : A.pullback A'' (f ∘ g) α = A'.pullback A'' g (A.pullback A' f α)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.DivisionOfDeterminantCharacter

theorem **Lacunary.mono** {J K : Finset ℤ} (h : Lacunary J) (hK : K ⊆ J) : Lacunary K

theorem **lacunary_of_growth** {J : Finset ℤ} (hpos : ∀ x ∈ J, 0 < x) (hg : ∀ x ∈ J, ∀ y ∈ J, x < y → (J.card : ℤ) ∗ x < y) : Lacunary J

theorem **not_lacunary_three** (k : ℤ) : ¬ Lacunary ({k, k + 1, k + 2} : Finset ℤ)

theorem **strictMono_jump** {a : ℕ} (lam : DomWeight a) : StrictMono fun k : Fin a => ((k : ℕ) : ℤ) - lam.w k

theorem **card_wtJumps** {a : ℕ} (lam : DomWeight a) : (wtJumps lam).card = a

theorem **wtJumps_val** {a : ℕ} (lam : DomWeight a) : (wtJumps lam).val.map (fun n : ℤ => (n : ℂ)) = (infOfWeight lam).map (fun z => ((a : ℂ) - 1) / 2 - z)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.DualIsConjByOne

theorem **dualIsConjBy_one** (π : CuspSub L a) : DualIsConjBy π 1 ↔ DualIsConj π

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExactFormsLeClosedForms

theorem **mem_closedForms** {k : ℕ} {α : A.Fld} : α ∈ A.closedForms k ↔ A.IsForm α ∧ A.IsDeg k α ∧ A.dExt α = 0

theorem **mem_exactForms** {k : ℕ} {α : A.Fld} : α ∈ A.exactForms k ↔ A.IsDeg k α ∧ ∃ η, A.IsForm η ∧ α = A.dExt η

theorem **mem_exactForms_succ** {k : ℕ} {α : A.Fld} : α ∈ A.exactForms (k + 1) ↔ ∃ η, A.IsForm η ∧ A.IsDeg k η ∧ α = A.dExt η

theorem **exactForms_zero** : A.exactForms 0 = ⊥

theorem **exactForms_le_closedForms** (k : ℕ) : A.exactForms k ≤ A.closedForms k

theorem **mk_surjective** (k : ℕ) : Function.Surjective (mk A k)

theorem **mk_eq_mk** {k : ℕ} (α β : A.closedForms k) : mk A k α = mk A k β ↔ (α - β : A.Fld) ∈ A.exactForms k

theorem **mk_eq_zero** {k : ℕ} (α : A.closedForms k) : mk A k α = 0 ↔ (α : A.Fld) ∈ A.exactForms k

@[elab_as_elim] theorem **induction_on** {k : ℕ} {P : A.HSm k → Prop} (c : A.HSm k) (h : ∀ α, P (mk A k α)) : P c

theorem **mem_fil** {p k : ℕ} {c : A.HSm k} : c ∈ fil A p k ↔ ∃ α : A.closedForms k, A.TypeGe p α ∧ mk A k α = c

theorem **fil_anti** {p q k : ℕ} (h : p ≤ q) : fil A q k ≤ fil A p k

theorem **fil_zero** (k : ℕ) : fil A 0 k = ⊤

theorem **fil_eq_bot** {p : ℕ} (hp : n < p) (k : ℕ) : fil A p k = ⊥

theorem **exact_mul** {k l : ℕ} {α β : A.Fld} (hα : α ∈ A.exactForms k) (hβ : β ∈ A.closedForms l) : α * β ∈ A.exactForms (k + l)

theorem **mul_exact** {k l : ℕ} {α β : A.Fld} (hα : α ∈ A.closedForms k) (hβ : β ∈ A.exactForms l) : α * β ∈ A.exactForms (k + l)

theorem **cup_mk** {k l : ℕ} (α : A.closedForms k) (β : A.closedForms l) : cup A k l (mk A k α) (mk A l β) = mk A (k + l) (mulClosed A k l α β)

theorem **cup_mem_fil** {p q k l : ℕ} {c : A.HSm k} {c' : A.HSm l} (hc : c ∈ fil A p k) (hc' : c' ∈ fil A q l) : cup A k l c c' ∈ fil A (p + q) (k + l)

theorem **one_cup** {k : ℕ} (c : A.HSm k) : cup A 0 k (one A) c = (Nat.zero_add k).symm ▸ c

theorem **cup_one** {k : ℕ} (c : A.HSm k) : cup A k 0 c (one A) = c

theorem **mulClosed_comm** {k l : ℕ} (α : A.closedForms k) (β : A.closedForms l) : (mulClosed A k l α β : A.Fld) = (-1 : ℂ) ^ (k * l) • (mulClosed A l k β α : A.Fld)

theorem **pullback_exact** {f : M' → M} (hf : A.IsHolo A' f) {k : ℕ} {α : A.Fld} (hα : α ∈ A.exactForms k) : A.pullback A' f α ∈ A'.exactForms k

theorem **pull_mk** {f : M' → M} (hf : A.IsHolo A' f) {k : ℕ} (α : A.closedForms k) : pull hf k (mk A k α) = mk A' k (pullClosed hf k α)

theorem **pull_id** (k : ℕ) : pull (A.isHolo_id) k = LinearMap.id

theorem **pull_comp** {f : M' → M} {g : M'' → M'} (hf : A.IsHolo A' f) (hg : A'.IsHolo A'' g) (k : ℕ) : pull (hf.comp hg) k = pull hg k ∘ₗ pull hf k

theorem **pull_cup** {f : M' → M} (hf : A.IsHolo A' f) {k l : ℕ} (c : A.HSm k) (c' : A.HSm l) : pull hf (k + l) (cup A k l c c') = cup A' k l (pull hf k c) (pull hf l c')

theorem **pull_mem_fil** {f : M' → M} (hf : A.IsHolo A' f) {p k : ℕ} {c : A.HSm k} (hc : c ∈ fil A p k) : pull hf k c ∈ fil A' p k

theorem **hasDeg_zero_dim** (α : Alt 0) : HasDeg 0 α

theorem **eq_pt₀** (x : (std 0).Pt) : x = pt₀

theorem **isForm_std₀** (α : (std 0).Fld) : (std 0).IsForm α

theorem **dExt_std₀** (α : (std 0).Fld) : (std 0).dExt α = 0

theorem **mem_closedForms_std₀** (α : (std 0).Fld) : α ∈ (std 0).closedForms 0

theorem **HSm.std₀Equiv_one** : HSm.std₀Equiv (HSm.one (std 0)) = 1

theorem **HSm.finrank_std₀** : Module.finrank ℂ ((std 0).HSm 0) = 1

theorem **isForm_zFld** : (std 1).IsForm zFld

theorem **dExt_zFld** (x : (std 1).Pt) : (std 1).dExt zFld x = dz 0

theorem **dExt_zFld_ne_zero** : (std 1).dExt zFld ≠ 0

theorem **isHolo_algAtlas** {p : X ⇢ Sp k} {D : ℕ} [SmoothOfRelativeDimension D p] {σ : k →+* ℂ} {p' : Y ⇢ Sp k'} {D' : ℕ} [SmoothOfRelativeDimension D' p'] {σ' : k' →+* ℂ} {f : Y ⇢ X} {m : CxPts p' σ' → CxPts p σ} (hm : ∀ z, (m z).1 = z.1 ≫ f) : (algAtlas p D σ).IsHolo (algAtlas p' D' σ') m

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExistsConstantCoeffOne

theorem **CurveData.exists_of_cm** (h2 : Formalism.L2.Th1.CM.CM2) (h3 : Formalism.L2.Th1.CM.CM3) (h4 : Formalism.L2.Th1.CM.CM4) (d : ℕ) (L : PeriodPair) (hsq : Squarefree d) (hd4 : d % 4 = 3) (hcm : Formalism.L2.Th1.CM.HasCMBy d L) (l : ℕ) [Fact l.Prime] (h5 : 5 ≤ l) (hd : ¬ l ∣ d) (hin : ¬ ∃ a : ZMod l, a ^ 2 = -((d : ℕ) : ZMod l)) : ∃ (g2 g3 s2 ε : MaximalUnramifiedExtension l) (σ : MaximalUnramifiedExtension l →+* MaximalUnramifiedExtension l) (F : MvPowerSeries (Fin 2) (MaximalUnramifiedExtension l)), CurveData l g2 g3 s2 ε σ F ∧ ‖(g2 : 𝕜_[l]) ^ 3 - 27 * (g3 : 𝕜_[l]) ^ 2‖ = 1

theorem **CurveData.exists_five** [Fact (Nat.Prime 5)] (h2 : Formalism.L2.Th1.CM.CM2) (h3 : Formalism.L2.Th1.CM.CM3) (h4 : Formalism.L2.Th1.CM.CM4) : ∃ (g2 g3 s2 ε : MaximalUnramifiedExtension 5) (σ : MaximalUnramifiedExtension 5 →+* MaximalUnramifiedExtension 5) (F : MvPowerSeries (Fin 2) (MaximalUnramifiedExtension 5)), CurveData 5 g2 g3 s2 ε σ F ∧ ‖(g2 : 𝕜_[5]) ^ 3 - 27 * (g3 : 𝕜_[5]) ^ 2‖ = 1

theorem **HypPair.exists_unramified** (l : ℕ) [Fact l.Prime] (h5 : 5 ≤ l) (σ : MaximalUnramifiedExtension l →+* MaximalUnramifiedExtension l) (hσ : UnrOK (MaximalUnramifiedExtension l) l σ) (E : Type) [Fintype E] [DecidableEq E] : ∃ f f' : E → (Bool × E) → MaximalUnramifiedExtension l, HypPair σ f f'

theorem **HypPair.exists_lift** (l : ℕ) [Fact l.Prime] (h5 : 5 ≤ l) (E : Type) [Fintype E] [DecidableEq E] : ∃ σ : MaximalUnramifiedExtension l →+* MaximalUnramifiedExtension l, UnrOK (MaximalUnramifiedExtension l) l σ ∧ ∃ f f' : E → (Bool × E) → MaximalUnramifiedExtension l, HypPair σ f f'

theorem **HypPair.symm** {K : Type} [NormedField K] {σ : K →+* K} {E n : Type} [DecidableEq E] [Fintype n] {f f' : E → n → K} (h : HypPair σ f f') : HypPair σ f' f

theorem **HostLog.iff_phSum** (l : ℕ) [Fact l.Prime] (g2 g3 s2 ε : MaximalUnramifiedExtension l) {n : Type} [Fintype n] (θ : MvPowerSeries n ℂ_[l]) : HostLog l g2 g3 s2 ε θ ↔ Formalism.DworkPowerSeriesWeights.Integral (Formalism.DworkPowerSeriesWeights.mvExp (Formalism.L2.Th1.E.phSum (Formalism.L2.Th1.Top.phL l g2 g3 s2 ε) : MvPowerSeries n ℂ_[l]) * (MvPowerSeries.subst (fun i : n => (at1 i (lgF ℂ_[l] l) : MvPowerSeries n ℂ_[l])) θ : MvPowerSeries n ℂ_[l]))

theorem **HostLog.constantCoeff_subst** {K σ τ : Type} [Field K] [Finite σ] {a : σ → MvPowerSeries τ K} (ha : ∀ s, MvPowerSeries.constantCoeff (a s) = 0) (f : MvPowerSeries σ K) : MvPowerSeries.constantCoeff (MvPowerSeries.subst a f : MvPowerSeries τ K) = MvPowerSeries.constantCoeff f

theorem **HostLog.constantCoeff_factor** (l : ℕ) [Fact l.Prime] (g2 g3 s2 ε : MaximalUnramifiedExtension l) (n : Type) [Fintype n] : MvPowerSeries.constantCoeff (Formalism.DworkPowerSeriesWeights.mvExp (∑ i : n, at1 i (PowerSeries.map (MaximalUnramifiedExtension l).subtype (phFF l g2 g3 s2 ε)) : MvPowerSeries n ℂ_[l])) = 1

theorem **HostLog.norm_constantCoeff_le** {l : ℕ} [Fact l.Prime] {g2 g3 s2 ε : MaximalUnramifiedExtension l} {n : Type} [Fintype n] {θ : MvPowerSeries n ℂ_[l]} (h : HostLog l g2 g3 s2 ε θ) : ‖MvPowerSeries.constantCoeff θ‖ ≤ 1

theorem **HostLog.exists_constantCoeff_one** (l : ℕ) [Fact l.Prime] (g2 g3 s2 ε : MaximalUnramifiedExtension l) (n : Type) [Fintype n] : ∃ θ : MvPowerSeries n ℂ_[l], MvPowerSeries.constantCoeff θ = 1 ∧ HostLog l g2 g3 s2 ε θ

theorem **thOrth.hasSubst** (D : D7) {N N' r : ℕ} {p : ℕ} [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (c : ℂ_[p]) (b : Fin r → Matrix (Fin N') (Fin N) D.host.E) : MvPowerSeries.HasSubst (fun k : D.what₂.Tau × Fin N' × Fin N => (∑ i, MvPowerSeries.C (c * lamC lam (embT D k.1 (b i k.2.1 k.2.2))) * MvPowerSeries.X i : MvPowerSeries (Fin r) ℂ_[p]))

theorem **thOrth.apply_X** (D : D7) {N N' r : ℕ} {p : ℕ} [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (c : ℂ_[p]) (b : Fin r → Matrix (Fin N') (Fin N) D.host.E) (k : D.what₂.Tau × Fin N' × Fin N) : thOrth D lam c b (MvPowerSeries.X k) = ∑ i, MvPowerSeries.C (c * lamC lam (embT D k.1 (b i k.2.1 k.2.2))) * MvPowerSeries.X i

theorem **thOrth.constantCoeff_eq** (D : D7) {N N' r : ℕ} {p : ℕ} [Fact p.Prime] (lam : Qbar →+* ℂ_[p]) (c : ℂ_[p]) (b : Fin r → Matrix (Fin N') (Fin N) D.host.E) (G : MvPowerSeries (D.what₂.Tau × Fin N' × Fin N) ℂ_[p]) : MvPowerSeries.constantCoeff (thOrth D lam c b G) = MvPowerSeries.constantCoeff G

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExistsMidLevIff

theorem **congAt_one** (P : HeightOneSpectrum (𝒪 E)) (n : ℕ) : CongAt (F

theorem **nearAt_self** (P : HeightOneSpectrum (𝒪 E)) (n : ℕ) (g : GL (Fin N) E) : NearAt (F

theorem **nearAt_rat_mul** (P : HeightOneSpectrum (𝒪 E)) (n : ℕ) (γ g : GL (Fin N) E) (x : GL (Fin N) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) : NearAt P n (ratA F γ * x) (γ * g) ↔ NearAt P n x g

theorem **mvL_mul** (g h : GL (Fin N) E) (Λ : Submodule (𝒪 E) (Fin N → E)) : mvL (g * h) Λ = mvL g (mvL h Λ)

theorem **latCo_eq_latK** (J : Matrix (Fin N) (Fin N) E) (co : GL (Fin N) E) (h : toLE co = stdCo (sp.cm.sig : E →+* E) J) (a : 𝒪 E) (y : GL (Fin N) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (Lb : Submodule (𝒪 E) (Fin N → E)) : latCo sp J co a y Lb = latK sp J a y Lb

theorem **latCo_rat** (J : Matrix (Fin N) (Fin N) E) (co γ : GL (Fin N) E) (hγ : γ ∈ UJ (sp.cm.sig : E →+* E) J) (a : 𝒪 E) (y : GL (Fin N) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (Lb : Submodule (𝒪 E) (Fin N → E)) : latCo sp J co a (ratA F γ * y) Lb = mvL (co * γ * co⁻¹) (latCo sp J co a y Lb)

theorem **latCo_mul** (J : Matrix (Fin N) (Fin N) E) (co γ : GL (Fin N) E) (hγ : γ ∈ UJ (sp.cm.sig : E →+* E) J) (a : 𝒪 E) (y : GL (Fin N) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (Lb : Submodule (𝒪 E) (Fin N → E)) : latCo sp J (co * γ) a y Lb = latCo sp J co a (ratA F γ * y) Lb

theorem **latK_eq_map** (J : Matrix (Fin N) (Fin N) E) (co : GL (Fin N) E) (a : 𝒪 E) (y : GL (Fin N) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (Lb : Submodule (𝒪 E) (Fin N → E)) : latK sp J a y Lb = (latCo sp J co a y Lb).map ((toStdCo (sp.cm.sig : E →+* E) J co).toLinearMap.restrictScalars (𝒪 E))

theorem **mem_seamP** (P : HeightOneSpectrum (𝒪 X.E)) (x : 𝒪 X.E) : RingOfIntegers.mapRingEquiv s.iE x ∈ (seamP s P).asIdeal ↔ x ∈ P.asIdeal

theorem **seamGL_mulVec** {n : ℕ} (g : Sp X.E n ≃ₗ[X.E] Sp X.E n) (x : Sp X.E n) : ((seamGL s g : GL (Fin n) D.host.E) : Matrix (Fin n) (Fin n) D.host.E) *ᵥ (fun l => s.iE (x l)) = fun l => s.iE (g x l)

theorem **seamHom_apply** {a b : ℕ} (x : HomE D.host.E a b) (w : Sp X.E a) (l : Fin b) : s.iE (seamHom s x w l) = x (fun k => s.iE (w k)) l

theorem **exists_midLev_iff** (j : ℕ) (P : rowSG Pr sel mu) (h : AutY X R j) (g : Sp X.E (R.N j) ≃ₗ[X.E] Sp X.E (R.N j)) : (∃ z, MidLev Ch sel mu cst j P h g z) ↔ g ∈ prCat (Pr

theorem **mul_toMoved** {x x₁ : UPts σ J (FiniteAdeleRing (𝒪 F) F)} (e : x₁.1 = x.1 * κ.1) : (x₁ * basePtU k).1 = (x * basePtU k).1 * (toMoved k κ).1

theorem **det_toMoved** : Matrix.GeneralLinearGroup.det (toMoved k κ).1 = Matrix.GeneralLinearGroup.det κ.1

theorem **prV_mul_basePt** {x x₁ : UPts σ J (FiniteAdeleRing (𝒪 F) F)} (v : HeightOneSpectrum (𝒪 F)) (e : bc (prV (E

theorem **y_eq** {β : St.Ch} (c : {c // St.chOf c = β}) : (St.K β).y c = St.y c.1

theorem **move_y** (k : St.Base) (c : C.cls i) : (St.move k).y c = St.y c * basePtU (k (St.chOf c))

theorem **exists_isAdIso** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] : ∃ ι : AfE F E ≃+* FiniteAdeleRing (𝒪 E) E, IsAdIso ι

theorem **IsAdIso.val_le** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {ι : AfE F E ≃+* FiniteAdeleRing (𝒪 E) E} (hι : IsAdIso ι) {P : HeightOneSpectrum (𝒪 E)} {z : AfE F E} (h : IntAtA (F

theorem **isChainTop_iff** (r : KerRead C X) (σ : X.K →+* ℂ) (v : Pr.Word) (m2 : ℕ) : r.IsChainTop Ch sel mu cst L σ v m2 ↔ r.SS = Sall sel mu v ∧ r.lev = levC sel mu cst v ∧ ∃ pt ix yb pT, IsChainTopW Ch sel mu cst L r σ v m2 pt ix yb pT

theorem **m_eq** : D.R.m = t.m2 + 2

theorem **cpl_lt** : t.m2 + 1 < D.R.m

theorem **top_le** : t.m2 + 1 + 1 ≤ D.R.m

theorem **mem_ssRec** (P : HeightOneSpectrum (𝒪 t.X.E)) : seamP t.seam P ∈ ssRec t u μ ↔ P ∈ (t.rd u μ).SS

theorem **AdTie.isChainTop** (H : AdTie A dtU a) : (t.rd u μ).IsChainTop t.Ch t.sel (t.tup μ) (t.cst μ) (t.idx μ) t.seam.σ₀ (t.word u) t.m2

theorem **AdTie.N_eq** (H : AdTie A dtU a) (i : ℕ) (hi : i ≤ D.R.m) (c : (D.chTw u μ).cls i) : (t.rd u μ).N i c = (A.pt i hi c).deco.1.N

theorem **AdTie.latK_eq** (H : AdTie A dtU a) (i : ℕ) (hi : i ≤ D.R.m) (c : (D.chTw u μ).cls i) : latK D.sp (gramN D i hi) a ((St i hi).y c).1 ((St i hi).K ((St i hi).chOf c)).lev.Lb = (t.seam.lat ((t.rd u μ).N i c)).map ((toStdCo (D.sp.cm.sig : D.host.E →+* D.host.E) (gramN D i hi) (A.crd i hi)).toLinearMap.restrictScalars (𝒪 D.host.E))

theorem **AdLawAt.mu_ne** {XiAd : AdDatum St} {muT : ℕ → (AfE D.host.F D.host.E)ˣ → ℂ} (H : AdLawAt A dtU a XiAd muT) {i : ℕ} (hi : i < D.R.m) {x : (AfE D.host.F D.host.E)ˣ} (hx : x ∈ U1 (sigD D)) : muT (i + 1) x ≠ 0

theorem **AdPins.unmove_iff** {k : ChainBase St} (B : AdData t u μ (moveCharts St k)) : AdPins B.unmove dtU ↔ AdPins B dtU

theorem **adLawAtB_one_iff** {XiAd : AdDatum St} {muT : ℕ → (AfE D.host.F D.host.E)ˣ → ℂ} : AdLawAtB A dtU a (fun _ _ _ => 1) XiAd muT ↔ AdLawAt A dtU a XiAd muT

theorem **latK_basePt** {i : ℕ} {hi : i ≤ D.R.m} {β : (St i hi).Ch} (k : basePts ((St i hi).K β).lev) (x : GL (Fin (D.R.N i)) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : latK D.sp (gramN D i hi) a (x ∗ k.1) ((St i hi).K β).lev.Lb = latK D.sp (gramN D i hi) a x ((St i hi).K β).lev.Lb

theorem **AdLawAt.unmove** {k : ChainBase St} {B : AdData t u μ (moveCharts St k)} {Xi : AdDatum (moveCharts St k)} {muT : ℕ → (AfE D.host.F D.host.E)ˣ → ℂ} (H : AdLawAt B dtU a Xi muT) : AdLawAtB B.unmove dtU a k Xi.unmove muT where

theorem **AdLaw.unmove** {k : ChainBase St} {B : AdData t u μ (moveCharts St k)} (H : AdLaw B dtU a) : AdLawB B.unmove dtU a k

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExistsReadingLaws

@[simp] theorem **Irr.ofDet_π** (n : ℕ) (χ : kˣ →* ℂˣ) (h : IsSmoothChar χ) : (Irr.ofDet n χ h).π = detRep (Fin n) χ

@[simp] theorem **Irr.toIrrRep_ρ** {n : ℕ} (π : Irr k n) : π.toIrrRep.ρ = π.π

@[simp] theorem **Irr.ofIrrRep_π** {n : ℕ} (σ : IrrRep (GL (Fin n) k)) : (Irr.ofIrrRep σ).π = σ.ρ

theorem **Irr.toIrrRep_equiv_iff** {n : ℕ} (π π' : Irr k n) : π.toIrrRep ≈ π'.toIrrRep ↔ Nonempty (π.π.Equiv π'.π)

theorem **exists_readAt** (T : WDTie N A) (i : N.Loc) (w : PlE H) (rc : ∀ n, Irr (w.adicCompletion H.E) n → WD (E

theorem **exists_reading_laws** (T : WDTie N A) (rc : ∀ (w : PlE H) n, Irr (w.adicCompletion H.E) n → WD (E

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExistsUnifPow

theorem **chRat_ofChiD** (h : ChiDRat D) {i : ℕ} {hi : i ≤ D.R.m} {T : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} {S : StepLevel D.host.F D.host.E (sigD D) (gramN D i hi) T D.R.f bad} (Y : DetTie S) (c : ℤ) (x : UPts (sigD D) (gramN D i hi) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ r : D.what₂.LL, D.what₂.embLL r = (((DetTw.ofChiD Y c).ch x : ℂˣ) : ℂ)

theorem **chRat_mem** (h : ChiDRat D) {u : D.what₂.Word} {Ch : ℕ → Type} {chOf : ∀ i, (D.mem u).C.cls i → Ch i} (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) : ∀ β cl, ∃ r : D.what₂.LL, D.what₂.embLL r = (((formsInMem F dt i hi).tw β).ch (((formsInMem F dt i hi).K β).y cl) : ℂ)

theorem **fillRec** (ex : Extra D L hT Gi Gl Kc Dx cX) : ∀ u w hg, FillTie hT u w hg (Gl.Fl u w hg)

theorem **ide_snd_mul** (a b : Ideles L) (v : HeightOneSpectrum (𝒪 L)) : ((a ∗ b : Ideles L) : AdeleRing (𝒪 L) L).2 v = (a : AdeleRing (𝒪 L) L).2 v ∗ (b : AdeleRing (𝒪 L) L).2 v

theorem **ide_snd_inv_mul** (a : Ideles L) (v : HeightOneSpectrum (𝒪 L)) : ((a⁻¹ : Ideles L) : AdeleRing (𝒪 L) L).2 v ∗ (a : AdeleRing (𝒪 L) L).2 v = 1

theorem **ide_finite** (a : Ideles L) : {v : HeightOneSpectrum (𝒪 L) | ¬ Valued.v ((a : AdeleRing (𝒪 L) L).2 v) = 1}.Finite

theorem **exists_unif_pow** (v : HeightOneSpectrum (𝒪 L)) (y : v.adicCompletion L) (hy : y ≠ 0) : ∃ n : ℤ, Valued.v (((atFin v (unif v)) ^ n : (v.adicCompletion L)ˣ) : v.adicCompletion L) = Valued.v y

theorem **ide_values** {Gp : Type} [CommGroup Gp] (f : Ideles L →* Gp) (hU : ∀ a : Ideles L, (∀ v, Valued.v ((a : AdeleRing (𝒪 L) L).2 v) = 1) → f a = 1) (S : Subgroup Gp) (hS : ∀ v, f (localIdele L v (atFin v (unif v))) ∈ S) (a : Ideles L) : f a ∈ S

theorem **chiDRat_of** {C : CharFrame D} (T : DtTied D C) (hV : C.chiPsi.ValuesIn 0 D.what₂.embLL) : ChiDRat D

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExistsUniqueVec

theorem **RealAt.fire** : ThetaFireRow (toLiftReal X.R X.F X.tupT X.hm X.hTop X.hinj) X.IL

theorem **used_mu** (i : D.nd.Loc) : Used0 D (D.nd.mu i)

theorem **RealAt.existsUnique_vec** {μ} (hμ : Used0 D μ) (j : Fin D.R.m) (f : D.what₂.Mt μ (j.1 + 1)) : ∃! y, X.R.extV j (X.tupT μ) y = X.fn hμ j f

theorem **RealAt.fn_injective** {μ} (hμ : Used0 D μ) (j : Fin D.R.m) : Function.Injective (X.fn hμ j)

theorem **OnGood.mono** {X : RealAt D cf nch tp} {i : D.nd.Loc} {p q : ∀ (j : Fin D.R.m) (t : X.T), (X.R.HV j t).H → Prop} (h : OnGood X i p) (hpq : ∀ j t y, p j t y → q j t y) : OnGood X i q

theorem **Agree.iso_rep** {X : RealAt D cf nch tp} {M : MtFam D} {rep : D.what₂.FT → NG.FinTyp D.host.E D.R.r} (A : Agree X M rep) (σ : D.what₂.FT) : (rep σ).Iso (X.rep σ)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ExistsUnitConjugateNormTwoPowerTwoAdic

theorem **isIntegral_jInv** : Formalism.L2.Th1.CM.JInvariantIntegrality

theorem **norm_jInv_le_one** (d : ℕ) (L : PeriodPair) (hsq : Squarefree d) (hmod : d % 4 = 3) (hcm : Formalism.L2.Th1.CM.HasCMBy d L) (p : ℕ) [Fact p.Prime] (ι : ℂ ≃+* ℂ_[p]) : ‖ι (Formalism.L2.Th1.CM.jInv L)‖ ≤ 1

theorem **twist** : Twist

theorem **deuringScale_of** (hT : Twist) : DeuringScale

theorem **deuring_lattice_homothety_two_adic** : DeuringScale

theorem **cubic_norm_two** : ‖(2 : ℂ_[2])‖ = 1 / 2

theorem **cubic_norm_sixteen** : ‖(16 : ℂ_[2])‖ = 1 / 16

theorem **integral_root_cubic_X_add_sixteen_two_adic** : RootCubic

theorem **unit_root_cubic_forces_unit_jInvariant** : UnitOfRoot

theorem **norm_three_padicComplex_two** : ‖(3 : ℂ_[2])‖ = 1

theorem **disc_ne_zero** (L : PeriodPair) : disc L ≠ 0

theorem **disc_of_mul** (L M : PeriodPair) (π : ℂ) (hπ : π ≠ 0) (hM : ∀ z, z ∈ M.lattice ↔ π ∗ z ∈ L.lattice) : disc M = π ^ 12 ∗ disc L

theorem **jUnit_of** (hW : TwoIsog) (hN : NormPow) (hC : Chain) (hR : RootCubic) (hU : UnitOfRoot) (hP : UnitChoice) : JUnit

theorem **full_of** (hS : DeuringScale) (hJ : JUnit) : Full

theorem **full_of_steps** (hT : DeuringScale) (hW : TwoIsog) (hN : NormPow) (hC : Chain) (hR : RootCubic) (hU : UnitOfRoot) (hP : UnitChoice) : Full

theorem **full_of_open** (hW : TwoIsog) (hN : NormPow) (hC : Chain) (hP : UnitChoice) : Full

lemma **G_split** (L L' : PeriodPair) (t : ℂ) (h2t : 2 * t ∈ L.lattice) (hL' : ∀ z, z ∈ L'.lattice ↔ (z ∈ L.lattice ∨ z - t ∈ L.lattice)) (ht : t ∉ L.lattice) (k : ℕ) (hk : 2 < k) : L'.G k = L.G k + L.sumInvPow t k

lemma **hasDerivAt_P** (L : PeriodPair) {z : ℂ} (hz : z ∉ L.lattice) : HasDerivAt L.weierstrassP (L.derivWeierstrassP z) z

lemma **deriv_two** (L : PeriodPair) {z : ℂ} (hz : z ∉ L.lattice) : deriv L.derivWeierstrassP z = 6 * L.weierstrassP z ^ 2 - L.g₂ / 2

lemma **deriv_three** (L : PeriodPair) {z : ℂ} (hz : z ∉ L.lattice) : deriv (deriv L.derivWeierstrassP) z = 12 * L.weierstrassP z * L.derivWeierstrassP z

lemma **deriv_four** (L : PeriodPair) {z : ℂ} (hz : z ∉ L.lattice) : deriv (deriv (deriv L.derivWeierstrassP)) z = 12 * L.derivWeierstrassP z ^ 2 + 12 * L.weierstrassP z * (6 * L.weierstrassP z ^ 2 - L.g₂ / 2)

lemma **sumInvPow_four** (L : PeriodPair) {t : ℂ} (ht : t ∉ L.lattice) : L.sumInvPow t 4 = (6 * L.weierstrassP t ^ 2 - L.g₂ / 2) / 6

lemma **sumInvPow_six** (L : PeriodPair) {t : ℂ} (ht : t ∉ L.lattice) : L.sumInvPow t 6 = (12 * L.derivWeierstrassP t ^ 2 + 12 * L.weierstrassP t * (6 * L.weierstrassP t ^ 2 - L.g₂ / 2)) / 120

lemma **alg** (e g2 g3 : ℂ) (h : g3 = 4 * e ^ 3 - g2 * e) (hd : g2 ^ 3 - 27 * g3 ^ 2 ≠ 0) : ((60 * e ^ 2 - 4 * g2) ^ 3 - 27 * (g3 + 84 * e ^ 3 - 7 * g2 * e) ^ 2 + 16 * (g2 ^ 3 - 27 * g3 ^ 2)) ^ 3 = 1728 * g2 ^ 3 / (g2 ^ 3 - 27 * g3 ^ 2) * ((60 * e ^ 2 - 4 * g2) ^ 3 - 27 * (g3 + 84 * e ^ 3 - 7 * g2 * e) ^ 2) * (g2 ^ 3 - 27 * g3 ^ 2) ^ 2

theorem **twoIsogeny_discriminant_ratio_cubic** : TwoIsog

theorem **norm_two_lt_one** : ‖(2 : ℂ_[2])‖ < 1

theorem **norm_odd_eq_one** (n : ℤ) (hn : Odd n) : ‖(n : ℂ_[2])‖ = 1

theorem **omegaK_add_conj** (d : ℕ) : omegaK d + conj (omegaK d) = 1

theorem **omegaK_sq_four** (d : ℕ) : 4 * omegaK d ^ 2 = 4 * omegaK d - (1 + d)

theorem **exists_unit_conjugate_norm_two_power_two_adic** : UnitChoice

theorem **full_of_normPow_chain** (hN : NormPow) (hC : Chain) : Full

theorem **stable_lattice_chain_successive_index_two** : Chain

theorem **full_of_normPow** (hN : NormPow) : Full

theorem **isPrincipal_pow_prime_over_two** : NormPow

theorem **goodModelTwo** : Formalism.L2.Th1.Top.GoodModelTwo

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.FiniteResidueFieldPadicInt

@[ext] theorem **Lat.ext** {L M : Lat O K r} (h : L.1 = M.1) : L = M

theorem **glMap_one** (L : Submodule O (Fin r → K)) : glMap 1 L = L

theorem **glMap_mul** (g h : GL (Fin r) K) (L : Submodule O (Fin r → K)) : glMap (g * h) L = glMap g (glMap h L)

theorem **glMap_le_glMap_iff** (g : GL (Fin r) K) (M L : Submodule O (Fin r → K)) : glMap g M ≤ glMap g L ↔ M ≤ L

theorem **glMap_smul** (g : GL (Fin r) K) (I : Ideal O) (L : Submodule O (Fin r → K)) : glMap g (I • L) = I • glMap g L

theorem **relIndex_glMap** (g : GL (Fin r) K) (M L : Submodule O (Fin r → K)) : (glMap g M).toAddSubgroup.relIndex (glMap g L).toAddSubgroup = M.toAddSubgroup.relIndex L.toAddSubgroup

theorem **glMap_isLattice** (g : GL (Fin r) K) (L : Lat O K r) : (glMap g L.1).IsLattice K where

@[simp] theorem **glLat_val** (g : GL (Fin r) K) (L : Lat O K r) : (glLat g L).1 = glMap g L.1

theorem **smul_def** (g : GL (Fin r) K) (L : Lat O K r) : g • L = glLat g L

@[simp] theorem **maxMul_val** (L : Lat O K r) : (maxMul L).1 = IsLocalRing.maximalIdeal O • L.1

theorem **pow_smul_eq_bot** {Q : Type} [AddCommGroup Q] [Module O Q] [Finite Q] (N : ℕ) : ∀ S : Submodule O Q, Nat.card S ≤ N + 1 → IsLocalRing.maximalIdeal O ^ N • S = ⊥

theorem **one_lt_qRes** : 1 < qRes O

theorem **two_le_qRes** : 2 ≤ qRes O

theorem **subIdx_finite** (n : ℕ) (L : Lat O K r) : (subIdx n L).Finite

theorem **subElem_finite** (k : ℕ) (L : Lat O K r) : (subElem k L).Finite

theorem **subIdx_zero** (L : Lat O K r) : subIdx 0 L = {L}

theorem **subElem_zero** (L : Lat O K r) : subElem 0 L = {L}

theorem **relIndex_maxMul** (L : Lat O K r) : (maxMul L).1.toAddSubgroup.relIndex L.1.toAddSubgroup = qRes O ^ r

theorem **subElem_top** (L : Lat O K r) : subElem r L = {maxMul L}

theorem **subElem_eq_empty** {k : ℕ} (hk : r < k) (L : Lat O K r) : subElem k L = ∅

theorem **maxMul_ne** (hr : 0 < r) (L : Lat O K r) : maxMul L ≠ L

theorem **smul_mem_subIdx_iff** (g : GL (Fin r) K) (n : ℕ) (M L : Lat O K r) : g • M ∈ subIdx n (g • L) ↔ M ∈ subIdx n L

theorem **smul_mem_subElem_iff** (g : GL (Fin r) K) (k : ℕ) (M L : Lat O K r) : g • M ∈ subElem k (g • L) ↔ M ∈ subElem k L

theorem **heckeAll_eq_sum** (n : ℕ) (φ : Lat O K r → E) (L : Lat O K r) : heckeAll n φ L = ∑ M ∈ (subIdx_finite n L).toFinset, φ M

theorem **heckeElem_eq_sum** (k : ℕ) (φ : Lat O K r → E) (L : Lat O K r) : heckeElem k φ L = ∑ M ∈ (subElem_finite k L).toFinset, φ M

theorem **heckeAll_zero** (φ : Lat O K r → E) : heckeAll 0 φ = φ

theorem **heckeElem_zero** (φ : Lat O K r → E) : heckeElem 0 φ = φ

theorem **heckeElem_top** (φ : Lat O K r → E) (L : Lat O K r) : heckeElem r φ L = φ (maxMul L)

theorem **heckeElem_eq_zero** {k : ℕ} (hk : r < k) (φ : Lat O K r → E) (L : Lat O K r) : heckeElem k φ L = 0

theorem **heckeAll_smul** (g : GL (Fin r) K) (n : ℕ) (φ : Lat O K r → E) (L : Lat O K r) : heckeAll n φ (g • L) = heckeAll n (fun M => φ (g • M)) L

theorem **heckeElem_smul** (g : GL (Fin r) K) (k : ℕ) (φ : Lat O K r → E) (L : Lat O K r) : heckeElem k φ (g • L) = heckeElem k (fun M => φ (g • M)) L

theorem **finite_residueField_padicInt** (p : ℕ) [Fact p.Prime] : Finite (IsLocalRing.ResidueField ℤ_[p])

theorem **qRes_padicInt** (p : ℕ) [Fact p.Prime] [Finite (IsLocalRing.ResidueField ℤ_[p])] : qRes ℤ_[p] = p

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.FinrankPartTrivialMulRel

theorem **mem_fix** {v : V} : v ∈ fix π K ↔ ∀ k ∈ K, π k v = v

theorem **apply_eq_of_coset** {v : V} (hv : v ∈ fix π K) {x y : G} (h : (x : G ⧸ K) = y) : π x v = π y v

theorem **sum_mem_fix** {n : ℕ} {z : Fin n → G} (hz : IsHeckeFam K z) {v : V} (hv : v ∈ fix π K) : (∑ i, π (z i) v) ∈ fix π K

theorem **coe_op** {n : ℕ} {z : Fin n → G} (hz : IsHeckeFam K z) (v : fix π K) : ((op π hz v : fix π K) : V) = ∑ i, π (z i) v

theorem **isHeckeFam_one** (K : Subgroup G) : IsHeckeFam K (fun _ : Fin 1 => (1 : G))

theorem **op_one** : op π (isHeckeFam_one K) = LinearMap.id

theorem **mem_part** {φ : fix π K →ₗ[E] M} : φ ∈ part π K R ↔ ∀ ⦃n : ℕ⦄ (z : Fin n → G) (hz : IsHeckeFam K z) (T : Module.End E M), R z hz T → φ ∘ₗ op π hz = T ∘ₗ φ

theorem **mem_part_iff_apply** {φ : fix π K →ₗ[E] M} : φ ∈ part π K R ↔ ∀ ⦃n : ℕ⦄ (z : Fin n → G) (hz : IsHeckeFam K z) (T : Module.End E M), R z hz T → ∀ v, φ (op π hz v) = T (φ v)

theorem **part_apply_op** (φ : part π K R) {n : ℕ} {z : Fin n → G} {hz : IsHeckeFam K z} {T : Module.End E M} (hT : R z hz T) (v : fix π K) : (φ : fix π K →ₗ[E] M) (op π hz v) = T ((φ : fix π K →ₗ[E] M) v)

theorem **mem_part_ofFun** {t : ∀ {n : ℕ} (z : Fin n → G), IsHeckeFam K z → Module.End E M} {φ : fix π K →ₗ[E] M} : φ ∈ part π K (HeckeRel.ofFun t) ↔ ∀ ⦃n : ℕ⦄ (z : Fin n → G) (hz : IsHeckeFam K z), φ ∘ₗ op π hz = t z hz ∘ₗ φ

theorem **Intertw.comp** {R : HeckeRel E K M} {R' : HeckeRel E K M'} {R'' : HeckeRel E K M''} {f : M →ₗ[E] M'} {g : M' →ₗ[E] M''} (hf : Intertw R R' f) (hg : Intertw R' R'' g) : Intertw R R'' (g ∘ₗ f)

@[simp] theorem **coe_map** {R' : HeckeRel E K M'} (f : M →ₗ[E] M') (hf : Intertw R R' f) (φ : part π K R) : ((map π f hf φ : part π K R') : fix π K →ₗ[E] M') = f ∘ₗ (φ : fix π K →ₗ[E] M)

theorem **Commutes.intertw** {f : Module.End E M} (hf : Commutes R f) : Intertw R R f

@[simp] theorem **coe_endo** (f : Module.End E M) (hf : Commutes R f) (φ : part π K R) : ((endo π f hf φ : part π K R) : fix π K →ₗ[E] M) = f ∘ₗ (φ : fix π K →ₗ[E] M)

@[simp] theorem **coe_partRep** {Γ : Type} [Group Γ] (ρ : Representation E Γ M) (hρ : ∀ γ, Commutes R (ρ γ)) (γ : Γ) (φ : part π K R) : ((partRep π ρ hρ γ φ : part π K R) : fix π K →ₗ[E] M) = ρ γ ∘ₗ (φ : fix π K →ₗ[E] M)

theorem **fix_trivial** : fix (Representation.trivial E G E) K = ⊤

theorem **op_trivial** {n : ℕ} {z : Fin n → G} (hz : IsHeckeFam K z) : op (Representation.trivial E G E) hz = (n : E) • LinearMap.id

theorem **part_trivial_mulRel** : part (Representation.trivial E G E) K (mulRel E K M) = ⊤

theorem **finrank_part_trivial_mulRel** : Module.finrank E (part (Representation.trivial E G E) K (mulRel E K E)) = 1

theorem **part_zeroRel** {V : Type} [AddCommGroup V] [Module E V] (π : Representation E G V) : part π K (zeroRel E K M) = ⊥

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.FirstChernClassMemHodgeFiltration

theorem **isDPLog_unique** {B : Type} [CommRing B] {J : Ideal B} (δ : DividedPowers J) {v a a' : B} (h : IsDPLog δ v a) (h' : IsDPLog δ v a') : a = a'

theorem **isDPLog_mem** {B : Type} [CommRing B] {J : Ideal B} (δ : DividedPowers J) {v a : B} (h : IsDPLog δ v a) : a ∈ J

theorem **isDPLog_eps** : IsDPLog Ex.δJ (1 + DualNumber.eps) DualNumber.eps

theorem **firstChernClass_mem_hodgeFiltration** {E : LB L} {z : H L 2} (h : IsFirstChernClassOfLineBundle E z) : z ∈ Fil L 2 1

theorem **isNilpotent_of_mem** (T : Thick L) {x : T.B} (hx : x ∈ T.J) : IsNilpotent x

theorem **exists_unit_lift** (T : Thick L) (v : (T.B ⧸ T.J)ˣ) : ∃ w : T.Bˣ, Units.map (Ideal.Quotient.mk T.J).toMonoidHom w = v

theorem **firstChernClass_logComponent** (i : L.ι) : IsFirstChernClassOfLineBundle (LB.ofComp i : LB L) 0

theorem **ratio_eq** {B : Type} [CommRing B] {i j : Fin 2} {q : Fin 2 → B} (a c : Bˣ) (ha : q i = a) (hc : q j = c) : ratio i j q = a * c⁻¹

theorem **ratio_rel** {B : Type} [CommRing B] (i j : Fin 2) (q q' : Fin 2 → B) (h : rel B q q') : ratio i j q = ratio i j q'

theorem **degOne_not_trivial** : ¬ ∃ s : Fin 2 → ∀ {B : CommRingCat.{0}}, lineL.Pt.obj B → Bˣ, (∀ α {B B' : CommRingCat.{0}} (φ : B ⟶ B') (y : lineL.Pt.obj B), lineL.In α y → s α (lineL.Pt.map φ y) = Units.map φ.hom.toMonoidHom (s α y)) ∧ ∀ α β {B : CommRingCat.{0}} (y : lineL.Pt.obj B), lineL.In α y → lineL.In β y → s α y = degOne.g α β y * s β y

theorem **cupC_one_one_model** {R : Type} [CommRing R] {C : Type} [Category.{0} C] {F : C ⥤ ModuleCat.{0} R} (μ : Mult F) (f g : Cn F 1) (s : ComposableArrows C 2) : cupC μ (show 1 + 1 = 2 from rfl) f g s = μ.mul _ ((F.map (s.map' 1 2)).hom (f ((fr_mono 1 2 (by omega)).functor ⋙ s))) ((F.map (s.map' 2 2)).hom (g ((bk_mono 1 1 2 (by omega)).functor ⋙ s)))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.FourthPowerTheta

theorem **data** {a₀ : ℕ} {b₀ c₀ : ℤ} {dK : ℕ} (ha : 0 < a₀) (hdK : 0 < dK) (lam : ℂ) (hlam : lam ≠ 0) (hcm : CmOK a₀ b₀ c₀ dK) : Squarefree dK ∧ dK % 4 = 3 ∧ Formalism.L2.Th1.CM.HasCMBy dK (cmL a₀ b₀ dK ha hdK lam hlam)

theorem **jets_comb** {K : Type} [NormedField K] [IsUltrametricDist K] {X κ : Type} [Fintype κ] (r : κ → K) (sg : κ → MvPowerSeries X K) (hr : ∀ k, ‖r k‖ ≤ 1) (hsg : ∀ k, JetsIntegral (sg k)) : JetsIntegral (∑ k, MvPowerSeries.C (r k) * sg k)

theorem **shape_paired_of** (hT : TensStd) (hG : GramTransfer) (hI : IotaOK) (hF : FrameCompose) (hS : ShapeLin) : ShapeLoc

theorem **jets_odd_of** (hT : TensStd) (hG : GramTransfer) (hF : FrameCompose) (hJ : JetsLin) : IntOpLoc

theorem **tens_std** (p : ℕ) [Fact p.Prime] (L : PeriodPair) (ι : ℂ ≃+* ℂ_[p]) (ω₀ : ℂ) (n : Type) [Fintype n] [DecidableEq n] (h : Matrix n n ℂ) (Λ : AddSubgroup (n → ℂ)) (α : (n → ℂ) → ℂ) (β : (Fin 4 × n → ℂ) → ℂ) (g : (Fin 4 × n → ℂ) ≃ₗ[ℂ] (Fin 4 × n → ℂ)) (hD : StdData L ω₀ h Λ α β g) (hln : ¬ p ∣ relIdx (stdLat L (Fin 4 × n)) (imgLat (g : (Fin 4 × n → ℂ) →ₗ[ℂ] (Fin 4 × n → ℂ)) (pow4 Λ))) (ε : ℂ) (hε : ‖ι ε‖ ≤ 1) (θ : (n → ℂ) → ℂ) (hθ : θ ∈ Th (matH h) α Λ) (hten : Formalism.L2.Th1.H.tens4 θ = ε • kernel4 L ω₀ Λ β g) (t : n → ℂ) (N : ℕ) (hN : 0 < N) (hlN : ¬ p ∣ N) (hNt : (N : ℂ) • t ∈ Λ) : ∃ (κ : Type) (_ : Fintype κ) (r : κ → ℂ_[p]) (s : κ → Fin 4 × n → ℂ) (N' : ℕ), (∀ k, ‖r k‖ ≤ 1) ∧ 0 < N' ∧ ¬ p ∣ N' ∧ (∀ k w, (N' : ℂ) * s k w ∈ L.lattice) ∧ linSub (fun w w' => ι (g.symm (Pi.single w' 1) w)) (Formalism.L2.Th1.G.tens4 (MvPowerSeries.map (ι : ℂ →+* ℂ_[p]) (taylorMv (opU (matH h) t θ)))

theorem **conj_omegaK** (d : ℕ) : conj (omegaK d) = (1 - Complex.I * ((Real.sqrt d : ℝ) : ℂ)) / 2

theorem **half_notMem** {a : ℕ} {b c : ℤ} {d : ℕ} (ha : 0 < a) (hd : 0 < d) (lam : ℂ) (hlam : lam ≠ 0) (hOK : CmOK a b c d) (sg : ℤ) (hsg : sg = 1 ∨ sg = -1) : (1 + (sg : ℂ) * (Complex.I * ((Real.sqrt d : ℝ) : ℂ))) / 2 * (lam * (a : ℂ) / 2) ∉ (cmL a b d ha hd lam hlam).lattice

theorem **fourth_power_theta** : TensZ

theorem **jets_linear** : JetsLinZ

theorem **jets_two_of** (hT : TensZ) (hG : GramTransfer) (hF : FrameCompose) (hJ : JetsLinZ) : IntOpTwoLoc

theorem **gram_transfer** : GramTransfer

theorem **fourth_power_std** : TensStd

theorem **jetsOdd** : IntOpLoc

theorem **jetsTwo** : IntOpTwoLoc

theorem **shape** (hE : EmbSep) : ShapeEigenLoc

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.FracZeroOfPos

theorem **frac_surjective** (f : Fin d → R) (x : LC f M) : ∃ (m : M) (k : ℕ), x = frac f m k

theorem **frac_prod_smul** (f : Fin d → R) (m : M) (k : ℕ) : frac f ((∏ l, f l) • m) (k + 1) = frac f m k

theorem **frac_zero_of_pos** (f : Fin d → R) (hd : 0 < d) (m : M) : frac f m 0 = 0

theorem **top_form** (hρ : IsRes s e ρ) (ω : Om ℂ B d) : ∃ h : B, ω = h • exteriorPower.ιMulti B d fun j => D ℂ B (s j)

theorem **apply_frac** (hρ : IsRes s e ρ) (k : ℕ) (h : B) : ρ (frac s (h • exteriorPower.ιMulti B d fun j => D ℂ B (s j)) (k + 1)) = MvPowerSeries.coeff (Finsupp.equivFunOnFinite.symm fun _ => k) (e h)

theorem **unique** {ρ' : LC s (Om ℂ B d) →ₗ[ℂ] ℂ} (hρ : IsRes s e ρ) (hρ' : IsRes s e ρ') : ρ = ρ'

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.GaloisActionWeightSpectralE1Apply

@[simp] theorem **castHQ_refl** {E : Spc} {n i : ℤ} (x : HQ l E n i) : castHQ l (E

theorem **pullHQ_map_id** (E : Spc) (n i : ℤ) : pullHQ l (𝟙 E) n i = LinearMap.id

theorem **pullHQ_map_comp** {E E' E'' : Spc} (f : E ⟶ E') (g : E' ⟶ E'') (n i : ℤ) : pullHQ l (f ≫ g) n i = pullHQ l f n i ∘ₗ pullHQ l g n i

theorem **mem_idx** {m : ℕ} {a : ℤ} {s : ℕ} : s ∈ idx m a ↔ -a ≤ (s : ℤ) ∧ a + 2 * (s : ℤ) < m

@[simp] theorem **castStr_refl** {t : ℕ} {n i : ℤ} (x : HQ l (sch.obj (Ystr J t)) n i) : castStr l J (rfl : t = t) (rfl : n = n) (rfl : i = i) x = x

theorem **str_succ_of_mem** {m : ℕ} {a : ℤ} {s : ℕ} (h : s ∈ idx m a) : str a s + 1 = str (a + 1) s

theorem **str_succ_of_mem'** {m : ℕ} {a : ℤ} {s : ℕ} (h : s ∈ idx m (a + 1)) : str a (s + 1) = str (a + 1) s + 1

theorem **str_N** (a : ℤ) (s : ℕ) : str a (s + 1) = str (a + 2) s

theorem **d1gys_deg** (b : ℤ) (s : ℕ) : b - 2 * ((s + 1 : ℕ) : ℤ) + 2 = b - 2 * (s : ℤ)

theorem **d1gys_tw** (w : ℤ) (s : ℕ) : w - ((s + 1 : ℕ) : ℤ) + 1 = w - (s : ℤ)

theorem **galS_Yι** (σ : Ω ≃ₐ[k] Ω) (S : Finset ι) : galS Ω p I σ S ≫ Yι (geomFam Ω p I) S = Yι (geomFam Ω p I) S ≫ GlobalTheory.Et.galMap Ω p σ

theorem **galS_one** (S : Finset ι) : galS Ω p I 1 S = 𝟙 _

theorem **galS_mul** (σ τ : Ω ≃ₐ[k] Ω) (S : Finset ι) : galS Ω p I (σ * τ) S = galS Ω p I σ S ≫ galS Ω p I τ S

theorem **inj_galStr** (σ : Ω ≃ₐ[k] Ω) {t : ℕ} (S : Idx ι t) : inj _ S ≫ galStr Ω p I σ t = galS Ω p I σ S.1 ≫ inj _ S

theorem **galStr_one** (t : ℕ) : galStr Ω p I 1 t = 𝟙 _

theorem **galStr_mul** (σ τ : Ω ≃ₐ[k] Ω) (t : ℕ) : galStr Ω p I (σ * τ) t = galStr Ω p I σ t ≫ galStr Ω p I τ t

theorem **galStr_Ystrπ** (σ : Ω ≃ₐ[k] Ω) (t : ℕ) : galStr Ω p I σ t ≫ Ystrπ _ t = Ystrπ _ t ≫ GlobalTheory.Et.galMap Ω p σ

theorem **δ_galStr** (σ : Ω ≃ₐ[k] Ω) (t : ℕ) (j : Fin (t + 2)) : δ _ t j ≫ galStr Ω p I σ t = galStr Ω p I σ (t + 1) ≫ δ _ t j

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.GlobSpecOfExists

theorem **star_det_mul** {E : Type} [Field E] [StarRing E] (γ : UF E n) : star ((LinearEquiv.det γ.1 : Eˣ) : E) * ((LinearEquiv.det γ.1 : Eˣ) : E) = 1

theorem **sigma_pow_mem** (v : Fin P.nV) (s : ℕ) (x : P.W v) : ∃ y : P.W v, (P.σ v ^ s) (algebraMap (P.W v) (P.Qq v) x) = algebraMap (P.W v) (P.Qq v) y

theorem **prod_mem** (T : PlTie X P) {δ : X.E} (hδ : star δ * δ = 1) : (∏ τ, T.eK (X.emb τ δ)) ∈ (algebraMap O Kf).range

theorem **detTw_mem** (T : PlTie X P) (c : ℤ) (γ : UF X.E n) : T.detTw c γ ∈ (algebraMap O Kf).range

theorem **nV_pos** (T : PlTie X P) (hp : p.Prime) : 0 < P.nV

theorem **PlTie.nV_pos_of_prime** {X : Fields} {P : Places p O Kf} (T : PlTie X P) (hp : p.Prime) : 0 < P.nV

theorem **PlTie.WtEq.unique** {X : Fields} {P : Places p O Kf} {T : PlTie X P} (hp : p.Prime) {n : ℕ} {hn : n = P.m + 1} {lam : X.Tau → DomWeight n} {c c' : ℤ} (h : T.WtEq hn lam c) (h' : T.WtEq hn lam c') : c = c'

theorem **PlTie.wtEq_withWt** {X : Fields} {P : Places p O Kf} (T : PlTie X P) (m' : ℕ) (lam : X.Tau → DomWeight (m' + 1)) (c : ℤ) (dom' : ∀ v, IsDominant fun s i => (lam (T.pos (v, s))).w i + c) : (T.withWt m' (fun v s i => (lam (T.pos (v, s))).w i + c) dom').WtEq (n

theorem **comp_mk_eq_iff** (d d' : S.D) : (Quot.mk _ d : Y.Comp) = Quot.mk _ d' ↔ ∃ g : S.G, Y.Aw (g • d) d'

theorem **exists_aw_rep** (d : S.D) : ∃ g : S.G, Y.Aw (g • d) (Y.rep (Quot.mk _ d))

theorem **finite_comp** (h : S.Ok) : Finite Y.Comp

theorem **exists_chi** (hp : p.Prime) (c : ℤ) : ∃ χ : S.G →* Oˣ, ∀ g, algebraMap O Kf (χ g : O) = T.eK ((Y.chr g : X.Kˣ) : X.K) * T.detTw c (Y.u g)

theorem **ChiLift.eq_chiLift** (hp : p.Prime) {c : ℤ} (x : Y.ChiLift T c) : x = Y.chiLift T hp c

theorem **glob_spec_of_exists** (a : Y.Comp) (L : P.Tup) (h : ∃ d, Y.Aw (Y.rep a) d ∧ T.tup hn (Y.N d) = L) : Y.Aw (Y.rep a) (Y.glob T hn a L) ∧ T.tup hn (Y.N (Y.glob T hn a L)) = L

theorem **aw_glob** (a : Y.Comp) (L : P.Tup) : Y.Aw (Y.rep a) (Y.glob T hn a L)

theorem **nbrAt_le** {v : Fin P.nV} {i : ℕ} {d d' : S.D} (h : d' ∈ Y.nbrAt T v i d) : (conjPrime₂ (T.pr v)).asIdeal • Y.N d' ≤ Y.N d ∧ (T.pr v).asIdeal • Y.N d ≤ Y.N d'

theorem **tup_lev_smul** {c : ℤ} (x : Y.ChiLift T c) (a : Y.Comp) (g : Y.Gam (Y.rep a)) (d : S.D) : T.tup hn (Y.N (g.1 • d)) = (Y.lev T hn x a).smul g (T.tup hn (Y.N d))

theorem **theta_lev_smul** {c : ℤ} (x : Y.ChiLift T c) (V : ValTie T hn lam c) (φ : S.EqFun) (a : Y.Comp) (g : Y.Gam (Y.rep a)) (d : S.D) : V.θ (Y.eV (φ.1 (g.1 • d))) = (((Y.lev T hn x a).χ g : Oˣ) : O) • (Y.lev T hn x a).actX g (V.θ (Y.eV (φ.1 d)))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.GoodIdempotentFixesOfOnlySeriesOfLabelActs

theorem **an_iff** {D : D7} : (T2.SumRepresentations.StrongMultiplicityOneCuspidalSums D.nd D.naut ↔ IsobaricCuspidalStrongMultiplicityOne D.nd D.naut) ∧ (T2.SumRepresentations.IsobaricMultiplicityOne D.nd D.naut ↔ IsobaricDiscreteSelfDualStrongMultiplicityOne D.nd D.naut)

theorem **analyticRigidity_of_admissible** {D : D7} (h : SumRepresentations.adm5 D) : T2.SumRepresentations.StrongMultiplicityOneCuspidalSums D.nd D.naut ∧ T2.SumRepresentations.IsobaricMultiplicityOne D.nd D.naut

theorem **LamUsed.at** {D : D7} (h : LamUsed D) {μ} (hμ : μ = D.what₂.mu0 ∨ ∃ w, μ = D.what₂.muAt w) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : LamPos D μ j

theorem **LamUsed.loc** {D : D7} (h : LamUsed D) (i : D.nd.Loc) : LamPos D (D.nd.mu i) (D.nd.j i)

theorem **goodIdempotentFixes_of_onlySeriesOfLabelActs** {D : D7} (h : GlobalTheory.TypeLeaf.SerLoc D) : GoodIdempotentFixesLocalisationForms D

theorem **goodIdempotent_of_goodProjector** {D : D7} (h : ArithmeticCylinders.Conclusion4 D) : GoodIdempotentFixesLocalisationForms D

theorem **partsGive_of** {D : D7} (h1 : SumRepresentations.AdmAt D → SumRepresentations.ReducedFamilyProposition D) (h2 : SumRepresentations.AdmAt D → RowSize D → HeckeSplitOne D → (∀ i : D.nd.Loc, LamPos D (D.nd.mu i) (D.nd.j i)) → ThetaOne D → SumRepresentations.ReducedParameterProposition D) : PartsGive D

theorem **hypO** : SumRepresentations.hypO D

theorem **c1** : GlobalDatumCompatibility D

theorem **bc** : SphericalInertData.Conclusion1 D

theorem **lam** : LamUsed D

theorem **isot** : SumRepresentations.adm8 D

theorem **ser** : GlobalTheory.TypeLeaf.SerLoc D

theorem **typeNZ** : ∀ j t, GlobalTheory.DownLift.TypeNZ I.X.R j t

theorem **e0loc** : GoodIdempotentFixesLocalisationForms D

theorem **mtPar** : MtParTie D.nd D.naut D.ntypes

theorem **adm1** : SumRepresentations.adm1 D

theorem **admAt** (a : RowsA D) (b : RowsB D) : SumRepresentations.AdmAt D

theorem **Conclusion1** (h : PartsGive D) (a : RowsA D) (b : RowsB D) : SumRepresentations.Conclusion1 D

theorem **nonempty_inputs_iff** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} : Nonempty (Inputs hL D) ↔ ∃ T : Tied hL D, Named T

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.HasPartAnd

theorem **StdTie.intro** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {l0 : I.Law0 l} {Z : TC.Later B (I.cols l)} {O : Obj (I.rec7 l Z)} {t : KerEntTied O.ex} (h : StdAt I l l0 Z O t) : StdTie (I.rec7 l Z) O t

theorem **StdTie.ind** {R : ∀ (D : D7) (O : Obj D), KerEntTied O.ex → Prop} (h : ∀ (B : GlobalTheory.Mem.Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (l0 : I.Law0 l) (Z : TC.Later B (I.cols l)) (O : Obj (I.rec7 l Z)) (t : KerEntTied O.ex), StdAt I l l0 Z O t → R (I.rec7 l Z) O t) {D : D7} {O : Obj D} {t : KerEntTied O.ex} (hD : StdTie D O t) : R D O t

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.InstFintypeHodgeBidegree

@[simp] theorem **coe_wedgeOm** {q q' Q : ℕ} (h : q + q' = Q) (x : Om R B q) (y : Om R B q') : (wedgeOm R B h x y : DR R B) = x ∗ y

theorem **dOm_wedgeOm** {q q' Q : ℕ} (h : q + q' = Q) (x : Om R B q) (y : Om R B q') : dOm R B Q (wedgeOm R B h x y) = wedgeOm R B (show q + 1 + q' = Q + 1 by omega) (dOm R B q x) y + (-1 : ℤ) ^ q • wedgeOm R B (show q + (q' + 1) = Q + 1 by omega) x (dOm R B q' y)

theorem **mOm_wedgeOm** {R' B' : Type} [CommRing R'] [CommRing B'] [Algebra R' B'] {σ : R →+* R'} {α : B →+* B'} (e : SemiHom σ α) {q q' Q : ℕ} (h : q + q' = Q) (x : Om R B q) (y : Om R B q') : e.mOm Q (wedgeOm R B h x y) = wedgeOm R' B' h (e.mOm q x) (e.mOm q' y)

@[simp] theorem **coe_oneOm** : (oneOm R B : DR R B) = 1

theorem **dOm_oneOm** : dOm R B 0 (oneOm R B) = 0

theorem **cupTerm_apply** {a a' A q q' Q : ℕ} (ha : a + a' = A) (hq : q + q' = Q) (f : Term F a q) (g : Term F a' q') (s : ComposableArrows C A) : cupTerm F ha hq f g s = wedgeF F hq _ (ev (F

theorem **ev_cupTerm** {a a' A q q' Q K : ℕ} (ha : a + a' = A) (hq : q + q' = Q) (f : Term F a q) (g : Term F a' q') (s : ComposableArrows C K) (p : Fin (A + 1) → Fin (K + 1)) (hp : Monotone p) (j : Fin (K + 1)) (hj : p (Fin.last A) ≤ j) : ev (F

theorem **wedgeF_ev_congr** {m n k q q' Q : ℕ} (hq : q + q' = Q) (s : ComposableArrows C k) (j : Fin (k + 1)) (f : Term F m q) (g : Term F n q') {p p' : Fin (m + 1) → Fin (k + 1)} {hp : Monotone p} {hp' : Monotone p'} {r r' : Fin (n + 1) → Fin (k + 1)} {hr : Monotone r} {hr' : Monotone r'} {h1 h1' h2 h2'} (e1 : ∀ a, (p a : ℕ) = p' a) (e2 : ∀ b, (r b : ℕ) = r' b) : wedgeF F hq _ (ev s p hp j h1 f) (ev s r hr j h2 g) = wedgeF F hq _ (ev s p' hp' j h1' f) (ev s r' hr' j h2' g)

theorem **dH_cupTerm** {m n k q q' Q : ℕ} (h : m + n = k) (hq : q + q' = Q) (f : Term F m q) (g : Term F n q') : dH F k Q (cupTerm F h hq f g) = cupTerm F (show m + 1 + n = k + 1 by omega) hq (dH F m q f) g + ((-1 : ℤ) ^ m) • cupTerm F (show m + (n + 1) = k + 1 by omega) hq f (dH F n q' g)

theorem **dOm_ev** {n k q : ℕ} (s : ComposableArrows C k) (p : Fin (n + 1) → Fin (k + 1)) (hp : Monotone p) (j : Fin (k + 1)) (h : p (Fin.last n) ≤ j) (f : Term F n q) : dOm R (F.obj (s.obj j)) q (ev s p hp j h f) = ev s p hp j h (dV F n q f)

theorem **dV_cupTerm** {m n k q q' Q : ℕ} (h : m + n = k) (hq : q + q' = Q) (f : Term F m q) (g : Term F n q') : dV F k Q (cupTerm F h hq f g) = cupTerm F h (show q + 1 + q' = Q + 1 by omega) (dV F m q f) g + ((-1 : ℤ) ^ q) • cupTerm F h (show q + (q' + 1) = Q + 1 by omega) f (dV F n q' g)

theorem **ev_self** {M : C ⥤ ModuleCat.{0} R} {m : ℕ} (s : ComposableArrows C m) (p : Fin (m + 1) → Fin (m + 1)) (hp : Monotone p) (e : ∀ a, (p a : ℕ) = a) (h : p (Fin.last m) ≤ Fin.last m) (f : Cn M m) : ev s p hp (Fin.last m) h f = f s

theorem **coe_ev_oneTerm** {k : ℕ} (s : ComposableArrows C k) (p : Fin 1 → Fin (k + 1)) (hp : Monotone p) (j : Fin (k + 1)) (h : p (Fin.last 0) ≤ j) : (ev s p hp j h (oneTerm F) : (OmF F 0).obj _).1 = 1

theorem **oneTerm_cupTerm** {a q : ℕ} (ha : 0 + a = a) (hq : 0 + q = q) (f : Term F a q) : cupTerm F ha hq (oneTerm F) f = f

theorem **Tot.sg_same** {a q n : ℕ} (h : a + q = n) (c : Term F a q) : Tot.sg F h c ⟨(a, q), h⟩ = c

theorem **Tot.sg_of_ne** {a q n : ℕ} (h : a + q = n) (c : Term F a q) (b : Bidegree n) (hb : b.1.2 ≠ q) : Tot.sg F h c b = 0

theorem **Tot.sum_sg** {n : ℕ} (c : Tot F n) : ∑ b : Bidegree n, Tot.sg F b.2 (c b) = c

theorem **Tot.induction** {n : ℕ} {P : Tot F n → Prop} (h0 : P 0) (ha : ∀ x y, P x → P y → P (x + y)) (hs : ∀ (a q : ℕ) (h : a + q = n) (c : Term F a q), P (Tot.sg F h c)) (c : Tot F n) : P c

theorem **dTot_sg** {a q n : ℕ} (h : a + q = n) (c : Term F a q) : dTot F n (Tot.sg F h c) = Tot.sg F (show a + 1 + q = n + 1 by omega) (dH F a q c) + (-1 : ℤ) ^ a • Tot.sg F (show a + (q + 1) = n + 1 by omega) (dV F a q c)

theorem **Tot.cup_apply** {m n N : ℕ} (h : m + n = N) (x : Tot F m) (y : Tot F n) : Tot.cup F h x y = ∑ b : Bidegree m, ∑ b' : Bidegree n, (-1 : ℤ) ^ (b.1.2 * b'.1.1) • Tot.sg F (show b.1.1 + b'.1.1 + (b.1.2 + b'.1.2) = N by have

theorem **Tot.cup_sg** {m n N a q a' q' A Q : ℕ} (h : m + n = N) (h₁ : a + q = m) (h₂ : a' + q' = n) (ha : a + a' = A) (hq : q + q' = Q) (h₃ : A + Q = N) (f : Term F a q) (g : Term F a' q') : Tot.cup F h (Tot.sg F h₁ f) (Tot.sg F h₂ g) = (-1 : ℤ) ^ (q * a') • Tot.sg F h₃ (cupTerm F ha hq f g)

theorem **dTot_cup** {m n N : ℕ} (h : m + n = N) (x : Tot F m) (y : Tot F n) : dTot F N (Tot.cup F h x y) = Tot.cup F (show m + 1 + n = N + 1 by omega) (dTot F m x) y + (-1 : ℤ) ^ m • Tot.cup F (show m + (n + 1) = N + 1 by omega) x (dTot F n y)

theorem **Tot.one_cup** {n : ℕ} (h : 0 + n = n) (x : Tot F n) : Tot.cup F h (Tot.one F) x = x

theorem **Tot.one_mem** : Tot.one F ∈ ZTot F 0

theorem **Tot.cup_mem_ZTot** {m n N : ℕ} (h : m + n = N) {x : Tot F m} {y : Tot F n} (hx : x ∈ ZTot F m) (hy : y ∈ ZTot F n) : Tot.cup F h x y ∈ ZTot F N

theorem **Tot.cup_mem_BTot_left** {m n N : ℕ} (h : m + n = N) {x : Tot F m} {y : Tot F n} (hx : x ∈ BTot F m) (hy : y ∈ ZTot F n) : Tot.cup F h x y ∈ BTot F N

theorem **Tot.cup_mem_BTot_right** {m n N : ℕ} (h : m + n = N) {x : Tot F m} {y : Tot F n} (hx : x ∈ ZTot F m) (hy : y ∈ BTot F n) : Tot.cup F h x y ∈ BTot F N

theorem **Hyp.one_cup** {n : ℕ} (h : 0 + n = n) (x : Hyp F n) : Hyp.cup F h (Hyp.one F) x = x

theorem **Hyp.one_ne_zero** (T : C) [Nontrivial (F.obj T)] : Hyp.one F ≠ 0

theorem **cupTerm_coneTerm** {m n N : ℕ} (h : m + n = N) (ω : Om R B m) (η : Om R B n) : cupTerm F rfl h (coneTerm e m ω) (coneTerm e n η) = coneTerm e N (wedgeOm R B h ω η)

theorem **coneCl_cup** {m n N : ℕ} (h : m + n = N) (ω : LinearMap.ker (dOm R B m)) (η : LinearMap.ker (dOm R B n)) : Hyp.cup F h (coneCl e he m ω) (coneCl e he n η) = coneCl e he N (wedgeCl h ω η)

theorem **HdR.one_cup** (h : 0 + n = n) (x : HdR s n) : HdR.cup s h (HdR.one s) x = x

theorem **HdR.ofForm_cup** {B : Type} [CommRing B] [Algebra k B] (h : a + b = n) (ω : LinearMap.ker (dOm k B a)) (η : LinearMap.ker (dOm k B b)) : HdR.cup (algStr k B) h (HdR.ofForm a ω) (HdR.ofForm b η) = HdR.ofForm n (wedgeCl h ω η)

theorem **coe_functionsZero_symm** {B : Type} [CommRing B] [Algebra k B] (x : B) : (((fn0 (R

theorem **HdR.pointEquiv_apply** (x : k) : HdR.pointEquiv k x = HdR.ofForm 0 ((fn0 k).symm x, dOm_zero_self k _)

theorem **HdR.pointEquiv_cup** (x y : k) : HdR.cup (algStr k k) (zero_add 0) (HdR.pointEquiv k x) (HdR.pointEquiv k y) = HdR.pointEquiv k (x * y)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.InvMemLatMonoidOfMemAwaySupp

theorem **mem_heckeLevel_inv** {K : Subgroup Γ} {h u : Γ} : u ∈ heckeLevel K h⁻¹ ↔ u ∈ K ∧ h * u * h⁻¹ ∈ K

theorem **heckeLevel_inv_of_comm** (K : Subgroup Γ) {h : Γ} (hc : ∀ u, h * u = u * h) : heckeLevel K h⁻¹ = K

theorem **cosetTrace_cosetTr_eq_hAct** {K' K : Subgroup Γ} {g : Γ} (hle : K' ≤ heckeLevel K g) (hfin : K'.relIndex K ≠ 0) {W : Type} [AddCommGroup W] [Module ℂ W] (φ : Γ ⧸ K → W) (x : Γ) : cosetTrace (hle.trans (heckeLevel_le K g)) (φ ∘ cosetTr K' K g fun _ hu => conj_mem_of_mem_heckeLevel (hle hu)) (x : Γ ⧸ K) = K'.relIndex (heckeLevel K g) • Formalism.UG.hAct K (Formalism.UG.dosetInd ℂ K g) (fun y => φ y) x

theorem **matEnd_zs** {N : ℕ} (X : GenModel gb N) {k : ℕ} (α : Matrix (Fin k) (Fin k) G₀.O) : (GlobalTheory.KS.matEnd X.univ.act α).left ≫ zs X k = zs X k

theorem **firstLeg_overComplex** : C.c1 k ≫ zs X k = zs X' k

theorem **secondLeg_overComplex** : C.c2 k ≫ zs X k = zs X' k

theorem **twistedSecondLeg_overComplex** (α : Matrix (Fin k) (Fin k) G₀.O) : C.c2Tw α ≫ zs X k = zs X' k

theorem **twistedSecondLeg_one** : C.c2Tw (1 : Matrix (Fin k) (Fin k) G₀.O) = C.c2 k

theorem **opTw_eq** [IsFinite (C.c1 k)] [Etale (C.c1 k)] (α : Matrix (Fin k) (Fin k) G₀.O) (a b : ℕ) : C.opTw α a b = C.op k a b ∘ₗ twistPull X α a b

theorem **mulBy_secondLeg_projection** (X : GenModel gb N) (c : ℕ) (i : Fin k) : (mulBy X c).c2 k ≫ (GlobalTheory.KS.pr X.univ.A i).left = (GlobalTheory.KS.pr X.univ.A i ^ c).left

theorem **IsForget.isHeckeLift** {k' : ℕ} {X' : GenModel gb (N * k')} {C : Corr X X'} (h : C.IsForget) : IsHeckeLift X X' (k' • LinearMap.id) (k' ^ 2) C.q1 C.φ1

theorem **powMap_id_eq** {S : Scheme.{0}} (A : Over S) (h : 𝟙 A.left ≫ A.hom = A.hom ≫ 𝟙 S) (k : ℕ) : GlobalTheory.KS.powMap (𝟙 S) (𝟙 A.left) h k = 𝟙 _

theorem **corrOp_of_eq_id** {Y : Scheme.{0}} {s : Y ⟶ Sp ℂ} {c1 c2 : Y ⟶ Y} [IsFinite c1] [Etale c1] (e : c1 = 𝟙 Y) (h1 : c1 ≫ s = s) (h2 : c2 ≫ s = s) (a b : ℕ) : corrOp c1 c2 h1 h2 a b = hodgePull h2 a b

theorem **gTateIso_ev_nsmul** {k : Type} [Field k] {Q : Type} [Group Q] {act : G₀.O → Q → Q} {e : ℕ → Q → Q → kˣ → Prop} (I : GTateIso G₀ k Q act e) (n c : ℕ) (x : G₀.La) : I.ev n (c • x) = I.ev n x ^ c

theorem **mem_latMonoid_of_mem_KN** {N : ℕ} {h : G.GAf} (hh : h ∈ L.KN N) : h ∈ latMonoid L

theorem **smul_mem_laHat** {a : AfP p} (ha : IsIntAf a) {v : AfP p ⊗[ℚ] VQ D} (hv : v ∈ LaHat D) : a • v ∈ LaHat D

theorem **GActsAt.mem_latMonoid** {S : Scheme.{0}} {s : S ⟶ Sp R} {N N' : ℕ} {o' : GObj gb N' s} {o'' : GObj gb N s} {k : Type} [Field k] {t : Sp k ⟶ S} {I' : GTIsoAt o' t} {I'' : GTIsoAt o'' t} {ψ : o'.A ⟶ o''.A} [IsMonHom ψ] {h : G.GAf} (H : GActsAt L I' I'' ψ 1 h) : h ∈ latMonoid L

theorem **GActsAt.comp** {S : Scheme.{0}} {s : S ⟶ Sp R} {N₁ N₂ N₃ : ℕ} {o₁ : GObj gb N₁ s} {o₂ : GObj gb N₂ s} {o₃ : GObj gb N₃ s} {k : Type} [Field k] {t : Sp k ⟶ S} {I₁ : GTIsoAt o₁ t} {I₂ : GTIsoAt o₂ t} {I₃ : GTIsoAt o₃ t} {ψ₁ : o₁.A ⟶ o₂.A} {ψ₂ : o₂.A ⟶ o₃.A} [IsMonHom ψ₁] [IsMonHom ψ₂] {h₁ h₂ : G.GAf} (H₁ : GActsAt L I₁ I₂ ψ₁ 1 h₁) (H₂ : GActsAt L I₂ I₃ ψ₂ 1 h₂) : GActsAt L I₁ I₃ (ψ₁ ≫ ψ₂) 1 (h₂ ∗ h₁)

theorem **actG_scalarAf** (a : (AfP p)ˣ) (v : AfP p ⊗[ℚ] VQ D) : L.actG (scalarAf G a) v = (a : AfP p) • v

theorem **scalarAf_mem_latMonoid** {a : (AfP p)ˣ} (ha : IsIntAf (a : AfP p)) : scalarAf G a ∈ latMonoid L

theorem **scalarAf_comm** (a : (AfP p)ˣ) (u : G.GAf) : scalarAf G a ∗ u = u ∗ scalarAf G a

theorem **actG_natScalar** (c : ℕ) (hc : c ≠ 0) (v : AfP p ⊗[ℚ] VQ D) : L.actG (scalarAf G (natUnit p c hc)) v = c • v

theorem **isGIsogWit_mul** {S : Scheme.{0}} {s : S ⟶ Sp R} {N M : ℕ} (e : N = M) (o : GObj gb N s) {h : G.GAf} {m c : ℕ} (hm : m.Coprime p) (hc : c.Coprime p) (hh : ∀ v, (m : ℤ) • L.actG h v = c • v) : IsGIsogWit L h (relevelObj e o) o (𝟙 o.A ^ c) (𝟙 o.A ^ c) m (c ∗ c)

theorem **isGIsogWit_refl** {S : Scheme.{0}} {s : S ⟶ Sp R} {N : ℕ} (o : GObj gb N s) : IsGIsogWit L 1 o o (𝟙 o.A) (𝟙 o.A) 1 1

theorem **isGIsogWit_forgetMul** {S : Scheme.{0}} {s : S ⟶ Sp R} {N k' : ℕ} (hN : 0 < N) (hk : 0 < k') (o : GObj gb (N ∗ k') s) : IsGIsogWit L 1 o (o.forgetMul hN hk) (𝟙 o.A) (𝟙 o.A) 1 1

theorem **relIndex_KN_ne_zero** {N N' : ℕ} (hN : 0 < N) (hN' : 0 < N') : (L.KN N').relIndex (L.KN N) ≠ 0

theorem **hkIndex_ne_zero** {N N' : ℕ} (hN : 0 < N) (hN' : 0 < N') (h : G.GAf) : hkIndex L N N' h ≠ 0

theorem **IsAdelicHecke.hkIndex_ne_zero** {h : G.GAf} {k' : ℕ} {X : GenModel gb N} {X' : GenModel gb (N ∗ k')} {C : Corr X X'} (H : IsAdelicHecke L h C) : hkIndex L N (N ∗ k') h ≠ 0

theorem **IsAdelicHecke.conj_mem** {h : G.GAf} {k' : ℕ} {X : GenModel gb N} {X' : GenModel gb (N ∗ k')} {C : Corr X X'} (H : IsAdelicHecke L h C) {u : G.GAf} (hu : u ∈ L.KN (N ∗ k')) : h ∗ u ∗ h⁻¹ ∈ L.KN N

theorem **KN_mul_le** (N k' : ℕ) : L.KN (N ∗ k') ≤ L.KN N

theorem **exists_isAdelicHecke_one** {k' : ℕ} (hp : (N ∗ k').Coprime p) (X : GenModel gb N) (X' : GenModel gb (N ∗ k')) : ∃ C : Corr X X', IsAdelicHecke L 1 C ∧ C.q1 = C.q2

theorem **normalised_model_eq_hAct** {N N' : ℕ} (hN : 0 < N) (hN' : 0 < N') {h : G.GAf} (hle : L.KN N' ≤ heckeLevel (L.KN N) h⁻¹) {W : Type} [AddCommGroup W] [Module ℂ W] (φ : G.GAf ⧸ L.KN N → W) (x : G.GAf) : ((hkIndex L N N' h : ℂ))⁻¹ • cosetTrace (hle.trans (heckeLevel_le _ _)) (φ ∘ cosetTr (L.KN N') (L.KN N) h⁻¹ fun _ hu => conj_mem_of_mem_heckeLevel (hle hu)) (x : G.GAf ⧸ L.KN N) = Formalism.UG.hAct (L.KN N) (Formalism.UG.dosetInd ℂ (L.KN N) h⁻¹) (fun y => φ y) x

theorem **mulPull_one** {G₀ : GDatum} {gb : GBase G₀ ℂ} (X : GenModel gb N) (k a b : ℕ) : mulPull X k 1 a b = 1

theorem **isHeckeOpZ_of_mul** (hN : N.Coprime p) (X : GenModel gb N) (k : ℕ) {h : G.GAf} {c : ℕ} (hc : c.Coprime p) (hcomm : ∀ u, h ∗ u = u ∗ h) (hh : ∀ v, L.actG h v = c • v) (a b : ℕ) : IsHeckeOpZ L X k h a b (mulPull X k c a b)

theorem **isHeckeOpZ_one** (hN : N.Coprime p) (X : GenModel gb N) (k a b : ℕ) : IsHeckeOpZ L X k 1 a b 1

theorem **isHeckeOpZ_natScalar** (hN : N.Coprime p) (X : GenModel gb N) (k : ℕ) {c : ℕ} (hc0 : c ≠ 0) (hc : c.Coprime p) (a b : ℕ) : IsHeckeOpZ L X k (scalarAf G (natUnit p c hc0)) a b (mulPull X k c a b)

theorem **IsHeckeOpZ.isHeckeOpFun** {X : GenModel gb N} {k : ℕ} {h : G.GAf} {a b : ℕ} {T : Module.End ℂ (hodgeH (zs X k) a b)} (H : IsHeckeOpZ L X k h a b T) : IsHeckeOpFun L X k (Formalism.UG.dosetInd ℂ (L.KN N) h⁻¹) a b T

theorem **isHeckeOpFun_zero** (X : GenModel gb N) (k a b : ℕ) : IsHeckeOpFun L X k 0 a b 0

theorem **IsHeckeOpFun.smul** {X : GenModel gb N} {k : ℕ} {f : G.GAf → ℂ} {a b : ℕ} {T : Module.End ℂ (hodgeH (zs X k) a b)} (H : IsHeckeOpFun L X k f a b T) (z : ℂ) : IsHeckeOpFun L X k (z • f) a b (z • T)

theorem **IsHeckeOpFun.add** {X : GenModel gb N} {k : ℕ} {f f' : G.GAf → ℂ} {a b : ℕ} {T T' : Module.End ℂ (hodgeH (zs X k) a b)} (H : IsHeckeOpFun L X k f a b T) (H' : IsHeckeOpFun L X k f' a b T') : IsHeckeOpFun L X k (f + f') a b (T + T')

theorem **IsHeckeOpFun.mem_hecke** {X : GenModel gb N} {k : ℕ} {f : G.GAf → ℂ} {a b : ℕ} {T : Module.End ℂ (hodgeH (zs X k) a b)} (H : IsHeckeOpFun L X k f a b T) : f ∈ Formalism.UG.Hecke ℂ (L.KN N)

theorem **mem_awayGrp** {S : Finset ℕ} {g : G.GAf} : g ∈ awayGrp G S ↔ ∀ v : AwayP p, (∃ q ∈ S, q.Prime ∧ (q : ℤ) ∈ v.1.asIdeal) → ubcMap (AfP.eval v) g = 1

theorem **scalarAf_mem_awayGrp** {S : Finset ℕ} {a : (AfP p)ˣ} (ha : ∀ v : AwayP p, (∃ q ∈ S, q.Prime ∧ (q : ℤ) ∈ v.1.asIdeal) → AfP.eval v (a : AfP p) = 1) : scalarAf G a ∈ awayGrp G S

theorem **eval_locUnit_same** (v : AwayP p) (x : (v.1.adicCompletion ℚ)ˣ) : AfP.eval v (locUnit v x : AfP p) = x

theorem **eval_locUnit_of_ne** {v w : AwayP p} (hw : w ≠ v) (x : (v.1.adicCompletion ℚ)ˣ) : AfP.eval w (locUnit v x : AfP p) = 1

theorem **isIntAf_of_scalarAf_mem** {a : (AfP p)ˣ} (ha : scalarAf G a ∈ latMonoid L) : IsIntAf (a : AfP p)

theorem **scalarAf_locUnit_spec** {S : Finset ℕ} {v : AwayP p} (hv : ∀ q ∈ S, q.Prime → (q : ℤ) ∉ v.1.asIdeal) {x : (v.1.adicCompletion ℚ)ˣ} (hx : (x : v.1.adicCompletion ℚ) ∈ v.1.adicCompletionIntegers ℚ) (hx' : ((x⁻¹ : (v.1.adicCompletion ℚ)ˣ) : v.1.adicCompletion ℚ) ∉ v.1.adicCompletionIntegers ℚ) (N : ℕ) : scalarAf G (locUnit v x) ∈ awayGrp G S ∧ scalarAf G (locUnit v x) ∈ latMonoid L ∧ scalarAf G (locUnit v x) ∉ L.KN N

theorem **mem_heckeAway** {N : ℕ} {S : Finset ℕ} {f : G.GAf → ℂ} : f ∈ heckeAway L N S ↔ f ∈ Formalism.UG.Hecke ℂ (L.KN N) ∧ ∀ g, f g ≠ 0 → g ∈ awaySupp L N S

theorem **dosetInd_mem_heckeAway** {N : ℕ} {S : Finset ℕ} {h : G.GAf} (hS : h ∈ awayGrp G S) (hΔ : h ∈ latMonoid L) : Formalism.UG.dosetInd ℂ (L.KN N) h⁻¹ ∈ heckeAway L N S

theorem **inv_mem_latMonoid_of_mem_awaySupp** {N : ℕ} {S : Finset ℕ} {g : G.GAf} (hg : g ∈ awaySupp L N S) : g⁻¹ ∈ latMonoid L

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsCompletionAt

theorem **IsWCorner.rH_eq** {πw : Λ →+* Matrix (Fin n) (Fin n) Ow} {E : Λ} {G : PDivGroup p T} {ρ : PDivGroup.Action Λ G} {H : VarpiDivModule p Ow T} {iH : H.G ⟶ G} {rH : G ⟶ H.G} (h : IsWCorner πw E ρ H iH rH) : (ρ E : G ⟶ G) ≫ rH = rH

theorem **IsWCorner.iH_eq** {πw : Λ →+* Matrix (Fin n) (Fin n) Ow} {E : Λ} {G : PDivGroup p T} {ρ : PDivGroup.Action Λ G} {H : VarpiDivModule p Ow T} {iH : H.G ⟶ G} {rH : G ⟶ H.G} (h : IsWCorner πw E ρ H iH rH) : iH ≫ (ρ E : G ⟶ G) = iH

theorem **IsWCorner.corner_congr** {πw : Λ →+* Matrix (Fin n) (Fin n) Ow} {E : Λ} {G : PDivGroup p T} {ρ : PDivGroup.Action Λ G} {H : VarpiDivModule p Ow T} {iH : H.G ⟶ G} {rH : G ⟶ H.G} (h : IsWCorner πw E ρ H iH rH) {c c' : Λ} (hc : πw c = πw c') : iH ≫ (ρ c : G ⟶ G) ≫ rH = iH ≫ (ρ c' : G ⟶ G) ≫ rH

theorem **iH_rH** (w : WPartOf πw A a H) : w.iH ≫ w.rH = 𝟙 H.G

theorem **rH_iH** (w : WPartOf πw A a H) : w.rH ≫ w.iH = ((w.t.ringAct a).padicExtend w.E : w.G ⟶ w.G)

theorem **π_E** (w : WPartOf πw A a H) : πw.π w.E = unitE Ow D.m

theorem **ι_eq** (w : WPartOf πw A a H) (x : Ow) (c : ℤ_[p] ⊗[ℤ] D.O) (hc : πw.π c = Matrix.scalar (Fin D.m) x) : (H.ι x : H.G ⟶ H.G) = w.iH ≫ ((w.t.ringAct a).padicExtend c : w.G ⟶ w.G) ≫ w.rH

theorem **TowerInduced.unique** {G : PDivGroup p T} {G' : PDivGroup p T'} {A : Over (Sp T)} [GrpObj A] {A' : Over (Sp T')} [GrpObj A'] {t : G.TorsionTower A} {t' : G'.TorsionTower A'} {φ : A'.left ⟶ A.left} {u v : PDivGroup.SemiHom σ G' G} (hu : TowerInduced t t' φ u) (hv : TowerInduced t t' φ v) : u = v

theorem **TowerInduced.exists** {G : PDivGroup p T} {G' : PDivGroup p T'} {A : Over (Sp T)} [GrpObj A] {A' : Over (Sp T')} [GrpObj A'] (t : G.TorsionTower A) (t' : G'.TorsionTower A') {φ : A'.left ⟶ A.left} (h : IsGrpPullback A' A (spMap σ) φ) : ∃ u : PDivGroup.SemiHom σ G' G, TowerInduced t t' φ u

theorem **IsInducedOf.unique** {w : WPartOf πw A a H} {w' : WPartOf πw A' a' H'} {φ : A'.left ⟶ A.left} {s s' : PDivGroup.SemiHom σ H'.G H.G} (h : IsInducedOf w w' φ s) (h' : IsInducedOf w w' φ s') : s = s'

theorem **IsInducedOf.exists** (w : WPartOf πw A a H) (w' : WPartOf πw A' a' H') {φ : A'.left ⟶ A.left} (h : IsGrpPullback A' A (spMap σ) φ) : ∃ s : PDivGroup.SemiHom σ H'.G H.G, IsInducedOf w w' φ s

theorem **IsInduced.exists** (w : WPart πw o H) (w' : WPart πw o' H') (m : ObjMap o' o (spMap σ)) : ∃ s : PDivGroup.SemiHom σ H'.G H.G, IsInduced w w' m s

theorem **IsInduced.unique** {w : WPart πw o H} {w' : WPart πw o' H'} {m : ObjMap o' o (spMap σ)} {s s' : PDivGroup.SemiHom σ H'.G H.G} (h : IsInduced w w' m s) (h' : IsInduced w w' m s') : s = s'

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsContinuousForgetForget

theorem **locallyCoverDense_forget** : (MorphismProperty.Over.forget @Etale ⊤ X).LocallyCoverDense (Scheme.overGrothendieckTopology @Etale X)

theorem **mem_small_iff** {U : X.Etale} (S : Sieve U) : S ∈ X.smallEtaleTopology U ↔ S.functorPushforward (Scheme.Etale.forget X) ∈ Scheme.etaleTopology.over X ((Scheme.Etale.forget X).obj U)

theorem **bc_forget** : bc f ≫ Scheme.Etale.forget Y = Scheme.Etale.forget X ≫ Over.pullback f

theorem **bc_coverPreserving** : CoverPreserving X.smallEtaleTopology Y.smallEtaleTopology (bc f) where

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsDPMorphismBot

theorem **le_id** (T : Thick L) : T.J ≤ T.J.comap (RingHom.id T.B)

theorem **le_comp** {T T' T'' : Thick L} (f : Hom T T') (g : Hom T' T'') : T.J ≤ T''.J.comap (g.φ.comp f.φ)

theorem **qmap_id** (T : Thick L) (h) : Ideal.quotientMap T.J (RingHom.id T.B) h = RingHom.id _

theorem **qmap_comp** {T T' T'' : Thick L} (f : Hom T T') (g : Hom T' T'') (h) : Ideal.quotientMap T''.J (g.φ.comp f.φ) h = (Ideal.quotientMap T''.J g.φ g.hle).comp (Ideal.quotientMap T'.J f.φ f.hle)

theorem **i_mem** (T : Thick L) {x : b.R} (hx : x ∈ b.I) : T.i x ∈ T.J

theorem **dpow_i** (T : Thick L) (n : ℕ) {x : b.R} (hx : x ∈ b.I) : T.δ.dpow n (T.i x) = T.i (b.γ.dpow n x)

theorem **pdPow_zero** (T : Thick L) : T.pdPow 0 = ⊤

theorem **pdPow_anti** (T : Thick L) {r r' : ℕ} (h : r ≤ r') : T.pdPow r' ≤ T.pdPow r

theorem **J_le_pdPow_one** (T : Thick L) : T.J ≤ T.pdPow 1

theorem **fil_zero** (n : ℕ) : Fil L n 0 = Set.univ

theorem **fil_anti** (n : ℕ) {r r' : ℕ} (h : r ≤ r') : Fil L n r' ⊆ Fil L n r

theorem **prod_pow** {M : Type} [CommMonoid M] (z : L'.ι → M) : ∏ i, ∏ j, z j ^ m.e i j = (∏ j, z j) ^ m.m

theorem **isDPMorphism_bot** {A B : Type} [CommRing A] [DecidableEq A] [CommRing B] {J : Ideal B} (δ : DividedPowers J) (f : A →+* B) : IsDPMorphism (dividedPowersBot A) δ f

theorem **node_covers** (k : Type) [CommRing k] (τ : k) : (node k τ).Covers

theorem **J_sq** : J ^ 2 = 0

theorem **T_J_ne_bot** : T.J ≠ ⊥

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsGrothendieckAbelianPresheaf

theorem **exists_shift** {Z : A} (F : A) (n : ℕ) : ∃ (I Q : A) (i : F ⟶ I) (q : I ⟶ Q) (h : i ≫ q = 0) (hS : (ShortComplex.mk i q h).ShortExact), Injective I ∧ ∀ x : Ext.{w} Z F (n + 1), ∃ y : Ext.{w} Z Q n, y.comp hS.extClass rfl = x

theorem **exists_epi_comp_eq_zero** (n : ℕ) {P Y : A} (x : Ext.{w} P Y (n + 1)) : ∃ (P' : A) (p : P' ⟶ P), Epi p ∧ (Ext.mk₀ p).comp x (zero_add _) = 0

theorem **ext_eq_zero_of_class** (Y : A) (Pr : A → Prop) (h : ∀ {P P' : A} (p : P' ⟶ P), Epi p → Pr P → ∃ (Q : A) (q : Q ⟶ P'), Epi q ∧ Pr (kernel (q ≫ p)) ∧ ∀ z : kernel (q ≫ p) ⟶ Y, ∃ z' : Q ⟶ Y, kernel.ι (q ≫ p) ≫ z' = z) (n : ℕ) {P : A} (hP : Pr P) (x : Ext.{w} P Y (n + 1)) : x = 0

theorem **extMap_apply** {Z : A} {Z' : B} (a : Z' ⟶ Φ.obj Z) (F : A) (n : ℕ) (x : Ext.{w} Z F n) : extMap.{w, w'} Φ a F n x = (Ext.mk₀ a).comp (x.mapExactFunctor Φ) (zero_add n)

theorem **extMap_bijective** {Z : A} {Z' : B} (a : Z' ⟶ Φ.obj Z) (ha : ∀ F : A, Function.Bijective fun g : Z ⟶ F => a ≫ Φ.map g) (hI : ∀ (I : A), Injective I → ∀ (n : ℕ) (x : Ext.{w'} Z' (Φ.obj I) (n + 1)), x = 0) (n : ℕ) (F : A) : Function.Bijective (extMap.{w, w'} Φ a F n)

theorem **mapExactFunctor_natTrans** (Ψ : A ⥤ B) [Ψ.Additive] [PreservesFiniteLimits Ψ] [PreservesFiniteColimits Ψ] (τ : Φ ⟶ Ψ) (n : ℕ) {Z F : A} (x : Ext.{w} Z F n) : (x.mapExactFunctor.{w, w'} Φ).comp (Ext.mk₀ (τ.app F)) (add_zero n) = (Ext.mk₀ (τ.app Z)).comp (x.mapExactFunctor.{w, w'} Ψ) (zero_add n)

theorem **isGrothendieckAbelian_sheaf** (R : Type 1) [CommRing R] : IsGrothendieckAbelian.{1} (Sheaf J (ModuleCat.{1} R))

theorem **pointwiseProjective_free** (T : Cᵒᵖ ⥤ Type 1) : PointwiseProjective (T ⋙ ModuleCat.free R)

theorem **presheaf_ext_eq_zero** (I : Cᵒᵖ ⥤ ModuleCat.{1} S) [Injective I] (n : ℕ) {P : Cᵒᵖ ⥤ ModuleCat.{1} R} (hP : PointwiseProjective P) (x : Ext.{1} P (I ⋙ ModuleCat.restrictScalars.{1} f) (n + 1)) : x = 0

theorem **Limp.epi** {T : ShortComplex (Sheaf J (ModuleCat.{1} R))} (hT : T.ShortExact) (h : Limp J T.X₁) : Epi T.g.hom

theorem **injective_obj** (I : Sheaf J (ModuleCat.{1} R)) [Injective I] : Injective I.obj

theorem **Limp.ext_eq_zero** (n : ℕ) {P : Cᵒᵖ ⥤ ModuleCat.{1} R} (hP : PointwiseProjective P) {F : Sheaf J (ModuleCat.{1} R)} (h : Limp J F) (x : Ext.{1} ((presheafToSheaf J _).obj P) F (n + 1)) : x = 0

theorem **ext_res_eq_zero** (I : Sheaf J (ModuleCat.{1} S)) (hI : Injective I) (n : ℕ) {P : Cᵒᵖ ⥤ ModuleCat.{1} R} (hP : PointwiseProjective P) (x : Ext.{1} ((presheafToSheaf J _).obj P) ((res J f).obj I) (n + 1)) : x = 0

theorem **freeEquiv_comp** (T : Cᵒᵖ ⥤ Type 1) {F G : Sheaf J (ModuleCat.{1} R)} (g : (freeSheaf J R).obj T ⟶ F) (u : F ⟶ G) : freeEquiv J T G (g ≫ u) = freeEquiv J T F g ≫ Functor.whiskerRight u.hom (forget _)

theorem **freeEquiv_resEquiv** (T : Cᵒᵖ ⥤ Type 1) (G : Sheaf J (ModuleCat.{1} S)) (g : (freeSheaf J S).obj T ⟶ G) : freeEquiv J T ((res J f).obj G) (resEquiv J f T G g) = freeEquiv J T G g

theorem **freeUnit_comp** (T : Cᵒᵖ ⥤ Type 1) {G : Sheaf J (ModuleCat.{1} S)} (g : (freeSheaf J S).obj T ⟶ G) : freeUnit J f T ≫ (res J f).map g = resEquiv J f T G g

theorem **cmp_apply** (T : Cᵒᵖ ⥤ Type 1) (G : Sheaf J (ModuleCat.{1} S)) (n : ℕ) (y : Ext.{1} ((freeSheaf J S).obj T) G n) : cmp J f T G n y = (Ext.mk₀ (freeUnit J f T)).comp (y.mapExactFunctor (res J f)) (zero_add n)

theorem **cmp_change** (u : F ⟶ (res J f).obj G) (T : Cᵒᵖ ⥤ Type 1) (n : ℕ) (x : Ext.{1} ((freeSheaf J R).obj T) F n) : cmp J f T G n (change J f u T n x) = x.comp (Ext.mk₀ u) (add_zero n)

theorem **change_mk₀** (u : F ⟶ (res J f).obj G) (T : Cᵒᵖ ⥤ Type 1) (g : (freeSheaf J R).obj T ⟶ F) : change J f u T 0 (Ext.mk₀ g) = Ext.mk₀ ((resEquiv J f T G).symm (g ≫ u))

theorem **change_postcomp_left** (v : F ⟶ F') (u : F' ⟶ (res J f).obj G) (T : Cᵒᵖ ⥤ Type 1) (n : ℕ) (x : Ext.{1} ((freeSheaf J R).obj T) F n) : change J f u T n (x.comp (Ext.mk₀ v) (add_zero n)) = change J f (v ≫ u) T n x

theorem **change_postcomp_right** (u : F ⟶ (res J f).obj G) (v : G ⟶ G') (T : Cᵒᵖ ⥤ Type 1) (n : ℕ) (x : Ext.{1} ((freeSheaf J R).obj T) F n) : (change J f u T n x).comp (Ext.mk₀ v) (add_zero n) = change J f (u ≫ (res J f).map v) T n x

theorem **change_smul** (u : F ⟶ (res J f).obj G) (T : Cᵒᵖ ⥤ Type 1) (n : ℕ) (r : R) (x : Ext.{1} ((freeSheaf J R).obj T) F n) : change J f u T n (r • x) = f r • change J f u T n x

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsHeckeMapLevRefl

theorem **levAct_spec** (g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) : M.levAct g ≫ M.s = M.s ∧ ObjLevMap (M.univAct g) M.univ (M.levAct g)

@[simp] theorem **levAct_comp_s** (g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) : M.levAct g ≫ M.s = M.s

theorem **eq_levAct** (g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) (f : M.X ⟶ M.X) (h₁ : f ≫ M.s = M.s) (h₂ : ObjLevMap (M.univAct g) M.univ f) : f = M.levAct g

@[simp] theorem **levAct_one** : M.levAct 1 = 𝟙 M.X

theorem **levAct_mul** (h g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) : M.levAct (h * g) = M.levAct h ≫ M.levAct g

@[simp] theorem **levAct_inv_comp** (g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) : M.levAct g⁻¹ ≫ M.levAct g = 𝟙 M.X

@[simp] theorem **levAct_comp_inv** (g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) : M.levAct g ≫ M.levAct g⁻¹ = 𝟙 M.X

@[simp] theorem **levAct_forget** (F₀ : RepFam b N) (g : GL (Fin D.n) (𝒪 D.F ⧸ w ^ m)) : M.levAct g ≫ M.forget F₀ = M.forget F₀

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsLocalRingIntegralClosure

theorem **isLocalRing_integralClosure** (O Ω : Type) [CommRing O] [HenselianLocalRing O] [Field Ω] [Algebra O Ω] : IsLocalRing (integralClosure O Ω)

@[simp] theorem **galInt_apply** (σ : Ω ≃ₐ[K] Ω) (x : integralClosure O Ω) : ((galInt O K σ x : integralClosure O Ω) : Ω) = σ x

theorem **galInt_algebraMap** (σ : Ω ≃ₐ[K] Ω) (r : O) : galInt O K σ (algebraMap O (integralClosure O Ω) r) = algebraMap O (integralClosure O Ω) r

theorem **galInt_mul** (σ τ : Ω ≃ₐ[K] Ω) (x : integralClosure O Ω) : galInt O K (σ * τ) x = galInt O K σ (galInt O K τ x)

theorem **galInt_one** (x : integralClosure O Ω) : galInt O K (1 : Ω ≃ₐ[K] Ω) x = x

theorem **galRes_residue** (σ : Ω ≃ₐ[K] Ω) (x : integralClosure O Ω) : galRes σ (residue _ x) = residue _ (galInt O K σ x)

theorem **ringHom_tower** : CommRingCat.ofHom (algebraMap O K) ≫ CommRingCat.ofHom (algebraMap K Ω) = CommRingCat.ofHom (algebraMap O (integralClosure O Ω)) ≫ CommRingCat.ofHom (integralClosure O Ω).val.toRingHom

@[reassoc] theorem **jGen_fst** : jGen K Ω p ≫ pullback.fst _ _ = pullback.fst _ _ ≫ pullback.fst _ _

@[reassoc] theorem **jGen_snd** : jGen K Ω p ≫ intStr Ω p = pullback.snd _ _ ≫ Spec.map (CommRingCat.ofHom (integralClosure O Ω).val.toRingHom)

@[reassoc] theorem **galIntMap_fst** (σ : Ω ≃ₐ[K] Ω) : galIntMap p σ ≫ pullback.fst _ _ = pullback.fst _ _

@[reassoc] theorem **galIntMap_snd** (σ : Ω ≃ₐ[K] Ω) : galIntMap p σ ≫ intStr Ω p = intStr Ω p ≫ Spec.map (CommRingCat.ofHom (galInt O K σ).toRingHom)

@[reassoc] theorem **galSpMap_i** (σ : Ω ≃ₐ[K] Ω) : galSpMap p σ ≫ iSp Ω p = iSp Ω p ≫ galIntMap p σ

theorem **galGen_j** (σ : Ω ≃ₐ[K] Ω) : GlobalTheory.Et.galMap Ω (genStr K p) σ ≫ jGen K Ω p = jGen K Ω p ≫ galIntMap p σ

theorem **iSp_galIntMap** {σ : Ω ≃ₐ[K] Ω} (hσ : σ ∈ inertia O K Ω) : iSp Ω p ≫ galIntMap p σ = iSp Ω p

theorem **galC_natural** {F G : Shv R} (φ : F ⟶ G) (σ : Ω ≃ₐ[K] Ω) : (pullC R (galSpMap p σ)).map (nbCMap R (iSp Ω p) (jGen K Ω p) φ) ≫ galC R p G σ = galC R p F σ ≫ nbCMap R (iSp Ω p) (jGen K Ω p) φ

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsMatrixForm

theorem **pull_id** (n : ℕ) (X : Scheme.{0}) : pull R n (𝟙 X) = LinearMap.id

theorem **pull_comp** (n : ℕ) {X Y Z : Scheme.{0}} (f : X ⟶ Y) (g : Y ⟶ Z) : pull R n (f ≫ g) = (pull R n f).comp (pull R n g)

@[reassoc] theorem **galMap_fst** (σ : Ω ≃ₐ[k] Ω) : galMap Ω p σ ≫ pullback.fst _ _ = pullback.fst _ _

@[reassoc] theorem **galMap_snd** (σ : Ω ≃ₐ[k] Ω) : galMap Ω p σ ≫ pullback.snd _ _ = pullback.snd _ _ ≫ Spec.map (CommRingCat.ofHom (σ : Ω →+* Ω))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsMonHomIdPow

theorem **comp_pow'** {S : Scheme.{0}} {X Y M : Over S} [MonObj M] (f : X ⟶ Y) (g : Y ⟶ M) (n : ℕ) : f ≫ g ^ n = (f ≫ g) ^ n

theorem **pow_comp'** {S : Scheme.{0}} {G G' Z : Over S} [MonObj G] [MonObj G'] (φ : G ⟶ G') [IsMonHom φ] (x : Z ⟶ G) (n : ℕ) : (x ^ n) ≫ φ = (x ≫ φ) ^ n

theorem **isMonHom_id_pow** {S : Scheme.{0}} (A : Over S) [GrpObj A] [IsCommMonObj A] (n : ℕ) : IsMonHom (𝟙 A ^ n)

theorem **isIsogWit_mul** {S : Scheme.{0}} {s : S ⟶ Sp R} {N : ℕ} (o : Obj b N s) (c : ℕ) (hc : c.Coprime p) : IsIsogWit L 1 o o (𝟙 o.A ^ c) (𝟙 o.A ^ c) c (c * c)

theorem **isHeckeWit_mul** {N : ℕ} (M : Model b N) (c : ℕ) (hc : c.Coprime p) : IsHeckeWit L 1 M M (𝟙 M.X) M.univ (𝟙 _ ^ c) (𝟙 _ ^ c) c (c * c) (GlobalTheory.UPel.HT.EtLev.LevForget.idMap M.univ)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsRegAlgTwistOfHalfIntegral

theorem **not_halfIntegral_of_isRegAlg** {α : CG K 1} (h : α.IsRegAlg) : ¬ HalfIntegral α

theorem **infOfWeight_liftWeight** {n : ℕ} (lam : DomWeight n) (t : ℤ) (ht : ∀ k, t ≤ lam.w k) : infOfWeight (liftWeight lam t ht) = (infOfWeight lam).map (· + (((n : ℂ) + 1) / 2 - t)) + (Multiset.range (n + 1)).map fun j : ℕ => -(n : ℂ) / 2 + (j : ℂ) - (n : ℂ) / 2

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsSepClosedResidueFieldPowerSeriesComplex

theorem **henselian** (L : StrictLoc X x) : HenselianLocalRing L.A

theorem **isSepClosed** (L : StrictLoc X x) : IsSepClosed (ResidueField L.A)

@[reassoc] theorem **gfibAut_fst** {T : Scheme.{0}} (g : T ⟶ Spec (CommRingCat.of O)) (σ : K ≃ₐ[O] K) : gfibAut K g σ ≫ pullback.fst _ _ = pullback.fst _ _

@[reassoc] theorem **gfibAut_snd** {T : Scheme.{0}} (g : T ⟶ Spec (CommRingCat.of O)) (σ : K ≃ₐ[O] K) : gfibAut K g σ ≫ pullback.snd _ _ = pullback.snd _ _ ≫ Spec.map (CommRingCat.ofHom (σ : K →+* K))

theorem **gfibAut_one** {T : Scheme.{0}} (g : T ⟶ Spec (CommRingCat.of O)) : gfibAut K g 1 = 𝟙 _

theorem **gfibAut_mul** {T : Scheme.{0}} (g : T ⟶ Spec (CommRingCat.of O)) (σ τ : K ≃ₐ[O] K) : gfibAut K g (σ * τ) = gfibAut K g σ ≫ gfibAut K g τ

@[reassoc] theorem **gfibMap_fst** {T T' : Scheme.{0}} {g : T ⟶ Spec (CommRingCat.of O)} {g' : T' ⟶ Spec (CommRingCat.of O)} (u : T ⟶ T') (hu : u ≫ g' = g) : gfibMap K u hu ≫ pullback.fst _ _ = pullback.fst _ _ ≫ u

@[reassoc] theorem **gfibMap_snd** {T T' : Scheme.{0}} {g : T ⟶ Spec (CommRingCat.of O)} {g' : T' ⟶ Spec (CommRingCat.of O)} (u : T ⟶ T') (hu : u ≫ g' = g) : gfibMap K u hu ≫ pullback.snd _ _ = pullback.snd _ _

theorem **gfibMap_id** {T : Scheme.{0}} (g : T ⟶ Spec (CommRingCat.of O)) : gfibMap K (𝟙 T) (Category.id_comp g) = 𝟙 _

theorem **gfibMap_comp** {T T' T'' : Scheme.{0}} {g : T ⟶ Spec (CommRingCat.of O)} {g' : T' ⟶ Spec (CommRingCat.of O)} {g'' : T'' ⟶ Spec (CommRingCat.of O)} (u : T ⟶ T') (hu : u ≫ g' = g) (v : T' ⟶ T'') (hv : v ≫ g'' = g') : gfibMap K (u ≫ v) (by rw [Category.assoc, hv, hu]) = gfibMap K u hu ≫ gfibMap K v hv

theorem **gfibMap_aut** {T T' : Scheme.{0}} {g : T ⟶ Spec (CommRingCat.of O)} {g' : T' ⟶ Spec (CommRingCat.of O)} (u : T ⟶ T') (hu : u ≫ g' = g) (σ : K ≃ₐ[O] K) : gfibAut K g σ ≫ gfibMap K u hu = gfibMap K u hu ≫ gfibAut K g' σ

theorem **gfibPull_rep** {T T' : Scheme.{0}} {g : T ⟶ Spec (CommRingCat.of O)} {g' : T' ⟶ Spec (CommRingCat.of O)} (u : T ⟶ T') (hu : u ≫ g' = g) (n : ℕ) (σ : K ≃ₐ[O] K) : (gfibPull R K u hu n).comp (gfibRep R K g' n σ) = (gfibRep R K g n σ).comp (gfibPull R K u hu n)

theorem **stalkPull_inertia** (f : X ⟶ Y) (hf : f ≫ q = p) {x : SpecialPoint p} {y : SpecialPoint q} (L : StrictLoc X x.1) (M : StrictLoc Y y.1) (h : L.scheme ⟶ M.scheme) (hh : h ≫ M.toX = L.toX ≫ f) (n : ℕ) (σ : K ≃ₐ[O] K) : (stalkPull R K f hf L M h hh n).comp (inertia R K q n y M σ) = (inertia R K p n x L σ).comp (stalkPull R K f hf L M h hh n)

theorem **stalkPull_id** {x : SpecialPoint p} (L : StrictLoc X x.1) (n : ℕ) : stalkPull R K (𝟙 X) (Category.id_comp p) L L (𝟙 _) (by rw [Category.id_comp, Category.comp_id]) n = LinearMap.id

theorem **stalkPull_comp** {Z : Scheme.{0}} {r : Z ⟶ Spec (CommRingCat.of O)} (f : X ⟶ Y) (hf : f ≫ q = p) (g : Y ⟶ Z) (hg : g ≫ r = q) {x : SpecialPoint p} {y : SpecialPoint q} {z : SpecialPoint r} (L : StrictLoc X x.1) (M : StrictLoc Y y.1) (N : StrictLoc Z z.1) (h : L.scheme ⟶ M.scheme) (hh : h ≫ M.toX = L.toX ≫ f) (k : M.scheme ⟶ N.scheme) (hk : k ≫ N.toX = M.toX ≫ g) (n : ℕ) : stalkPull R K (f ≫ g) (by rw [Category.assoc, hg, hf]) L N (h ≫ k) (by rw [Category.assoc, hk, ← Category.assoc, hh, Category.assoc]) n = (stalkPull R K f hf L M h hh n).comp (stalkPull R K g hg M N k hk n)

theorem **pushAut_congr** {X' : Scheme.{0}} (π : X' ⟶ X) {g g' : X' ⟶ X'} (e : g = g') (hg : g ≫ π = π) (hg' : g' ≫ π = π) (n : ℕ) (x : SpecialPoint p) (L : StrictLoc X x.1) : pushAut R K p π g hg n x L = pushAut R K p π g' hg' n x L

theorem **pushAut_id** {X' : Scheme.{0}} (π : X' ⟶ X) (n : ℕ) (x : SpecialPoint p) (L : StrictLoc X x.1) : pushAut R K p π (𝟙 _) (Category.id_comp _) n x L = LinearMap.id

theorem **pushAut_comp** {X' : Scheme.{0}} (π : X' ⟶ X) (g g' : X' ⟶ X') (hg : g ≫ π = π) (hg' : g' ≫ π = π) (n : ℕ) (x : SpecialPoint p) (L : StrictLoc X x.1) : pushAut R K p π (g ≫ g') (by rw [Category.assoc, hg', hg]) n x L = (pushAut R K p π g hg n x L).comp (pushAut R K p π g' hg' n x L)

theorem **pushAut_inertia** {X' : Scheme.{0}} (π : X' ⟶ X) (g : X' ⟶ X') (hg : g ≫ π = π) (n : ℕ) (x : SpecialPoint p) (L : StrictLoc X x.1) (σ : K ≃ₐ[O] K) : (pushAut R K p π g hg n x L).comp (pushInertia R K p π n x L σ) = (pushInertia R K p π n x L σ).comp (pushAut R K p π g hg n x L)

theorem **pushAct_apply** {X' : Scheme.{0}} (π : X' ⟶ X) {G : Type} [Group G] (ρ : G →* End X') (hρ : ∀ g, ρ g ≫ π = π) (n : ℕ) (x : SpecialPoint p) (L : StrictLoc X x.1) (g : G) : pushAct R K p π ρ hρ n x L g = pushAut R K p π (ρ g⁻¹) (hρ _) n x L

theorem **pushAct_inertia** {X' : Scheme.{0}} (π : X' ⟶ X) {G : Type} [Group G] (ρ : G →* End X') (hρ : ∀ g, ρ g ≫ π = π) (n : ℕ) (x : SpecialPoint p) (L : StrictLoc X x.1) (g : G) (σ : K ≃ₐ[O] K) : (pushAct R K p π ρ hρ n x L g).comp (pushInertia R K p π n x L σ) = (pushInertia R K p π n x L σ).comp (pushAct R K p π ρ hρ n x L g)

theorem **henselian_powerSeries_complex** : HenselianLocalRing (PowerSeries ℂ)

theorem **isSepClosed_residueField_powerSeries_complex** : IsSepClosed (ResidueField (PowerSeries ℂ))

theorem **faithfulSMul_powerSeries_complex** : FaithfulSMul (PowerSeries ℂ) (AlgebraicClosure (FractionRing (PowerSeries ℂ)))

theorem **isAlgebraic_powerSeries_complex** : Algebra.IsAlgebraic (PowerSeries ℂ) (AlgebraicClosure (FractionRing (PowerSeries ℂ)))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsStrictlySemistable

theorem **_root_.GlobalTheory.LogCris.evalAt_res** {B : Type} [CommRing B] {X : Scheme.{0}} (y : Spec (CommRingCat.of B) ⟶ X) {U V : X.Opens} (hV : V ≤ U) (h : ⊤ ≤ y ⁻¹ᵁ U) (h' : ⊤ ≤ y ⁻¹ᵁ V) (a : Γ(X, U)) : _root_.GlobalTheory.Cris.evalAt y V h' (X.presheaf.map (homOfLE hV).op a) = _root_.GlobalTheory.Cris.evalAt y U h a

private theorem **gammaSpecIso_hom_appLE_spec_map** {A C : CommRingCat.{0}} (χ : A ⟶ C) (e : (⊤ : (Spec C).Opens) ≤ Spec.map χ ⁻¹ᵁ ⊤) (b : Γ(Spec A, ⊤)) : (Scheme.ΓSpecIso C).hom ((Spec.map χ).appLE ⊤ ⊤ e b) = χ ((Scheme.ΓSpecIso A).hom b)

theorem **_root_.GlobalTheory.LogCris.evalAt_map** {B B' : CommRingCat.{0}} {X : Scheme.{0}} (φ : B ⟶ B') (y : Spec B ⟶ X) {U : X.Opens} (h : ⊤ ≤ y ⁻¹ᵁ U) (h' : ⊤ ≤ (Spec.map φ ≫ y) ⁻¹ᵁ U) (a : Γ(X, U)) : _root_.GlobalTheory.Cris.evalAt (B

theorem **_root_.GlobalTheory.LogCris.evalAt_base** {B : Type} [CommRing B] {X : Scheme.{0}} (q : X ⟶ Spec (CommRingCat.of O)) {k : Type} [CommRing k] (κ : O →+* k) (ψ : k →+* B) (y : Spec (CommRingCat.of B) ⟶ X) (hy : y ≫ q = Spec.map (CommRingCat.ofHom (ψ.comp κ))) {U : X.Opens} (h : ⊤ ≤ y ⁻¹ᵁ U) (a : O) : _root_.GlobalTheory.Cris.evalAt y U h ((q.appLE ⊤ U le_top).hom ((Scheme.ΓSpecIso (CommRingCat.of O)).inv.hom a)) = ψ (κ a)

theorem **_root_.GlobalTheory.LogCris.le_preimage_inf** {B : Type} [CommRing B] {X : Scheme.{0}} (y : Spec (CommRingCat.of B) ⟶ X) {U V : X.Opens} (h : ⊤ ≤ y ⁻¹ᵁ U) (h' : ⊤ ≤ y ⁻¹ᵁ V) : ⊤ ≤ y ⁻¹ᵁ (U ⊓ V)

theorem **le_preimage_map** {B B' : CommRingCat.{0}} {X : Scheme.{0}} (φ : B ⟶ B') (y : Spec B ⟶ X) {U : X.Opens} (h : ⊤ ≤ y ⁻¹ᵁ U) : ⊤ ≤ (Spec.map φ ≫ y) ⁻¹ᵁ U

theorem **isStrictlySemistable** (C : SSAtlas ϖ X q ι A) : GlobalTheory.Et.IsStrictlySemistable ϖ q

theorem **isSSFun_logFun** (C : SSAtlas ϖ X q ι A) : IsSSFun ϖ X q κ (C.logFun κ)

theorem **logFun_In** (C : SSAtlas ϖ X q ι A) (α : A) {B : CommRingCat.{0}} (y : (C.logFun κ).Pt.obj B) : (C.logFun κ).In α y ↔ ⊤ ≤ (show PtOver X q κ B from y).1.2 ⁻¹ᵁ C.U α

theorem **logFun_f** (C : SSAtlas ϖ X q ι A) (α : A) (i : ι) {B : CommRingCat.{0}} (y : PtOver X q κ B) (h : ⊤ ≤ y.1.2 ⁻¹ᵁ C.U α) : (C.logFun κ).f α i (B

theorem **logFun_g** (C : SSAtlas ϖ X q ι A) (α β : A) (i : ι) {B : CommRingCat.{0}} (y : PtOver X q κ B) (h : ⊤ ≤ y.1.2 ⁻¹ᵁ (C.U α ⊓ C.U β)) : ((C.logFun κ).g α β i (B

theorem **_root_.GlobalTheory.LogCris.model_mul_eq_zero** {m r : ℕ} (hϖ : ϖ ∈ nonZeroDivisors O) (g : _root_.GlobalTheory.Et.ssModel O ϖ m r) (h : algebraMap O _ ϖ * g = 0) : g = 0

theorem **model_X_mem** {m r : ℕ} (hϖ : ϖ ∈ nonZeroDivisors O) (j : Fin m) (hj : (j : ℕ) < r) : Ideal.Quotient.mk _ (MvPolynomial.X j) ∈ nonZeroDivisors (GlobalTheory.Et.ssModel O ϖ m r)

theorem **model_appLE** (a : O) : ((modelStr ϖ r).appLE ⊤ ⊤ le_top).hom ((Scheme.ΓSpecIso (CommRingCat.of O)).inv.hom a) = (Scheme.ΓSpecIso (CommRingCat.of (GlobalTheory.Et.ssModel O ϖ r r))).inv.hom (algebraMap O _ a)

theorem **isSSFun_model** (hr : 1 ≤ r) (hϖ : ϖ ∈ nonZeroDivisors O) {k : Type} [CommRing k] (κ : O →+* k) : IsSSFun ϖ (Spec (CommRingCat.of (GlobalTheory.Et.ssModel O ϖ r r))) (modelStr ϖ r) κ (modelFun ϖ r hr hϖ κ)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsStrictlySemistableAlt

theorem **Mu.ext'** {A : Type} [CommRing A] {x y : Mu n A} (h : (((Additive.toMul (show Additive (rootsOfUnity n A) from x) : rootsOfUnity n A) : Aˣ) : A) = (((Additive.toMul (show Additive (rootsOfUnity n A) from y) : rootsOfUnity n A) : Aˣ) : A)) : x = y

theorem **Mu.map_id** (A : Type) [CommRing A] : Mu.map n (RingHom.id A) = LinearMap.id

theorem **Mu.map_comp** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (φ : A →+* B) (ψ : B →+* C) : Mu.map n (ψ.comp φ) = (Mu.map n ψ).comp (Mu.map n φ)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsStrictlySemistableModel

theorem **eq_one_of_isIso_codiag** {S : Scheme.{0}} (pt : S) {m : ℕ} (h : IsIso (Sigma.desc (fun _ : Fin m => 𝟙 S) : ∐ (fun _ : Fin m => S) ⟶ S)) : m = 1

theorem **isTransfer_id** (X : Scheme.{0}) : IsTransfer R (𝟙 X) (𝟙 _)

theorem **isStrictlySemistable_model** (h1 : 1 ≤ r) (h2 : r ≤ m) : IsStrictlySemistable ϖ (Spec.map (CommRingCat.ofHom (algebraMap O (ssModel O ϖ m r))))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsSumOfSelf

theorem **Base.uniq** (H : Base) : GlobalTheory.TheRec.RecUnique

theorem **SatMatch.wdOf_eq** {sat : HeightOneSpectrum (𝒪 E) → Multiset ℂˣ → Prop} {ι : Qbar l ≃+* ℂ} {r : G E Ω →* GL (Fin a) (Qbar l)} (H : SatMatch sat ι r) : ∀ᶠ w : HeightOneSpectrum (𝒪 E) in Filter.cofinite, ∃ B, sat w B ∧ ∀ (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) (A : ValuationSubring Ω) (h : IsPlaceOver A w), wdOf h hl ι r = (WD.lpUnr h (B.map fun b => ((b⁻¹ : ℂˣ) : ℂ))).lpTw ((1 - (a : ℝ)) / 2)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsTemperedGLOfHasSatGL

theorem **mem_integer_iff** (x : w.adicCompletion E) : x ∈ 𝒪[w.adicCompletion E] ↔ x ∈ Formalism.UG.Ov (F

theorem **range_eq** {K : Type} [Field K] {A B : Subring K} (h : A = B) (n : ℕ) : (Formalism.UG.bc (n

theorem **kgl_eq_ko** (n : ℕ) : GlobalTheory.LFun.KGL w n = GlobalTheory.SplitWeil.KO (w.adicCompletion E) (Fin n)

theorem **isSpherical_of_hasSatGL** (n : ℕ) {V : Type} [AddCommGroup V] [Module ℂ V] {ρ : Representation ℂ (GL (Fin n) (w.adicCompletion E)) V} {B : Multiset ℂˣ} (h : GlobalTheory.LFun.HasSatGL w ρ B) : GlobalTheory.SmGL.IsSpherical ρ

theorem **isTemperedGL_of_hasSatGL** (hT : GlobalTheory.CohomologicalConjSelfDualTempered.CohomologicalCsdTemperedUnr) (L : Type) [Field L] [NumberField L] [IsCMField L] (n : ℕ) (hn : 2 ≤ n) (π : CuspSub L n) (hc : π.IsCsd) (hcoh : GlobalTheory.GKCoh.IsCohomological π) (y :

HeightOneSpectrum (𝒪 L)) (σ : SmIrr (GL (Fin n) (y.adicCompletion L))) (hσ : π.IsLocComp y σ) {B : Multiset ℂˣ} (hB : GlobalTheory.LFun.HasSatGL y σ.ρ B) : GlobalTheory.LocGL.IsTemperedGL σ.ρ

theorem **isSpherical_trivial** (n : ℕ) : GlobalTheory.SmGL.IsSpherical (1 : Representation ℂ (GL (Fin n) (w.adicCompletion E)) ℂ)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsWeight

theorem **InScope.all** {s : Level} {π : CuspSub E a} {χ : HChar (maximalRealSubfield E)} (h : InScope s l π χ) : InScope .all l π χ

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IsZeroFreeSchOfIsEmpty

theorem **_root_.GlobalTheory.EtC.Tr.transfer_spec** {Z X : Scheme.{0}} {f : Z ⟶ X} (h : ∃ t, _root_.GlobalTheory.EtC.Tr.IsTransfer R f t) : _root_.GlobalTheory.EtC.Tr.IsTransfer R f (_root_.GlobalTheory.EtC.Tr.transfer R f)

theorem **isZero_freeSch_of_isEmpty** (X : Scheme.{0}) [IsEmpty X] : IsZero (freeSch R X)

theorem **eq_one_of_isIso_codiag** {S : Scheme.{0}} (pt : S) {m : ℕ} (h : IsIso (Sigma.desc (fun _ : Fin m => 𝟙 S) : ∐ (fun _ : Fin m => S) ⟶ S)) : m = 1

theorem **isTransfer_id** (X : Scheme.{0}) : IsTransfer R (𝟙 X) (𝟙 _)

theorem **eq_id_of_isTransfer_id** {X : Scheme.{0}} {t : freeSch R X ⟶ freeSch R X} (ht : IsTransfer R (𝟙 X) t) : t = 𝟙 _

theorem **transfer_id** (X : Scheme.{0}) : transfer R (𝟙 X) = 𝟙 _

lemma **_root_.GlobalTheory.EtC.Tr.chainMap_free** {X Z : Scheme.{0}} (g : X ⟶ Z) : _root_.GlobalTheory.EtC.Tr.chainMap R ((_root_.GlobalTheory.EtC.free R).map (yoneda.map g)) = (_root_.GlobalTheory.EtC.chains R).map (_root_.GlobalTheory.EtC.sch.map g)

theorem **_root_.GlobalTheory.EtC.Tr.along_id** (X : Scheme.{0}) (F : _root_.GlobalTheory.EtC.Shv R) (n : ℤ) (x : _root_.GlobalTheory.EtC.H R (_root_.GlobalTheory.EtC.sch.obj X) F n) : _root_.GlobalTheory.EtC.Tr.along R F n (𝟙 _) x = x

theorem **_root_.GlobalTheory.EtC.Tr.along_free** {X Z : Scheme.{0}} (g : X ⟶ Z) (F : _root_.GlobalTheory.EtC.Shv R) (n : ℤ) : _root_.GlobalTheory.EtC.Tr.along R F n ((_root_.GlobalTheory.EtC.free R).map (yoneda.map g)) = _root_.GlobalTheory.EtC.pull R (_root_.GlobalTheory.EtC.sch.map g) F n

theorem **trH_id** (X : Scheme.{0}) (F : Shv R) (n : ℤ) (x : H R (sch.obj X) F n) : trH R (𝟙 X) F n x = x

theorem **trHt_id** (q : ℕ) (X : Scheme.{0}) (n i : ℤ) (x : Ht q (sch.obj X) n i) : trHt q (𝟙 X) n i x = x

theorem **_root_.GlobalTheory.EtC.Tr.trHl_spec** {Z X : Scheme.{0}} {f : Z ⟶ X} {n i : ℤ} (h : _root_.GlobalTheory.EtC.Tr.RedCompat l f n i) : _root_.GlobalTheory.EtC.Tr.IsTrHl l f n i (_root_.GlobalTheory.EtC.Tr.trHl l f n i)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.IwahoriAdmissibleSet

theorem **hLin_apply** {ι : Type} (s : Finset ι) (c : ι → ℂ) (f : ι → HeckeFn U) (g : G) : (hLin s c f).toFun g = ∑ i ∈ s, c i * (f i).toFun g

theorem **nodup_sList** (S : Finset (Fin a)) : (sList S).Nodup

theorem **mem_sList** {S : Finset (Fin a)} {c : Fin a} : c ∈ sList S ↔ c ∈ S

theorem **val_wS** (ϖ : kˣ) (S : Finset (Fin a)) : (wS k ϖ S : Matrix (Fin a) (Fin a) k) = (cyc S).permMatrix k * diagonal fun c => (dEnt k ϖ S c : k)

theorem **phiS_toFun** (ϖ : kˣ) (S : Finset (Fin a)) : (phiS k ϖ S).toFun = dcFn (iwahori k (Fin a)) (wS k ϖ S)

theorem **phiS_self** (ϖ : kˣ) (S : Finset (Fin a)) : (phiS k ϖ S).toFun (wS k ϖ S) = 1

theorem **kot_apply** (ϖ : kˣ) (g : GL (Fin a) k) : (kot (a

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.JetStepEqZeroExample

theorem **jetInVV_trueWt** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) : JetInVV X.K (trueWt hc i) (trueWt hc (i + 1)) (nuEnd hc i)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.JumpWindow

theorem **JointTB.mono** {D : D7} {HW : HWOf D.loc} {F : FonOf D.loc D.locA HW} {i : D.nd.Loc} {r : D.nd.X i → (GE D.nd →* GL (Fin D.R.nV) (D.nd.O i))} {π : D.nd.X i} {M0 h0 cB cB' : ℕ} (hle : cB ≤ cB') (h : JointTB D HW F i r π M0 h0 cB) : JointTB D HW F i r π M0 h0 cB'

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.KaehlerOneFormsToExtensionDeRhamMulSelf

theorem **_root_.GlobalTheory.W1.kaehlerOneFormsToExtensionDeRham_mul_self** (ω : Ω[K⁄k]) : _root_.GlobalTheory.W1.map1 k K L ω * _root_.GlobalTheory.W1.map1 k K L ω = 0

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.KernelCharactersEqReference

theorem **prod_inf_mul_finPart** (a : Ideles L) : ∃ y : V w : InfinitePlace L, (w.Completion)ˣ, (∏ w, infIdele L w (y w)) * finPartI L a = a

theorem **isFin_finPart** (a : Ideles L) : IsFinIdele (finPartI L a)

theorem **isUnitary_of_orderDvd** {χ : HChar L} {n : ℕ} (hn : 0 < n) (h : χ.OrderDvd n) : χ.IsUnitary

theorem **eDt_finPart** (a : Ideles D.host.E) : Dt.eDt (finPartI D.host.E a) = Dt.eDt a

theorem **kernelCharacters_eq_reference** : MtIs Dt Tp D.what₂.mu0 D.muTil

theorem **mtIs_unique** {μ : D.what₂.Tup} {mt mt' : ℕ → D.Dt →* ℂˣ} (h : MtIs Dt Tp μ mt) (h' : MtIs Dt Tp μ mt') (j : Fin D.R.m) : mt (j.1 + 1) = mt' (j.1 + 1)

theorem **exists_mtIs_iff** (hm : 0 < D.R.m) (μ : D.what₂.Tup) : (∃ mt, MtIs Dt Tp μ mt) ↔ ∀ a, Dt.eDt a = 1 → (Tp.xi μ).1 (finPartI D.host.E a) = 1

theorem **not_mtIs** (hm : 0 < D.R.m) {μ : D.what₂.Tup} {v : HeightOneSpectrum (𝒪 D.host.E)} {x : (v.adicCompletion D.host.E)ˣ} (hx : Dt.eDt (localIdele D.host.E v x) = 1) (hξ : (Tp.xi μ).loc v x ≠ 1) (mt : ℕ → D.Dt →* ℂˣ) : ¬ MtIs Dt Tp μ mt

theorem **tup_eq_of_mtIs** (hm : 0 < D.R.m) {μ μ' : D.what₂.Tup} {mt : ℕ → D.Dt →* ℂˣ} (h : MtIs Dt Tp μ mt) (h' : MtIs Dt Tp μ' mt) (a : Ideles D.host.E) (ha : IsFinIdele a) : (Tp.xi μ).1 a = (Tp.xi μ').1 a

theorem **mtType_of_mtIs** {μ : D.what₂.Tup} {mt : ℕ → D.Dt →* ℂˣ} (h : MtIs Dt Tp μ mt) (u : D.what₂.Word) : MtType u mt

theorem **mtType_muTil** (u : D.what₂.Word) : MtType u D.muTil

theorem **mtType_word** {u u' : D.what₂.Word} {mt : ℕ → D.Dt →* ℂˣ} (h : MtType u mt) : MtType u' mt

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.KernelRationalReferenceTuple

theorem **nu_step_old** (D : D7) (u : D.what₂.Word) (μ : D.what₂.Tup) (i : ℕ) (hi : i + 1 < D.R.m) (τ : D.what₂.Tau) (r : ℕ) : DV.nuRow D u (i + 2) τ r − DV.nuRow D u (i + 1) τ r = (((D.chTw u μ).sSharp (i + 1) τ : ℤ) : ℝ) − (((D.chTw u μ).sDiv i τ : ℤ) : ℝ)

theorem **nu_step_new** (D : D7) (u : D.what₂.Word) (μ : D.what₂.Tup) (i : ℕ) (hi : i + 1 < D.R.m) (τ : D.what₂.Tau) (a : ℕ) (ha : a < D.R.t + 1) : DV.nuRow D u (i + 2) τ (D.R.N i + a) = (((D.chTw u μ).sSharp (i + 1) τ : ℤ) : ℝ) − (((D.chTw u μ).sDiv i τ : ℤ) : ℝ)

theorem **N_pos** (D : D7) (i : ℕ) : 0 < D.R.N i

theorem **lamW_anti** (u : D.what₂.Word) (i : ℕ) (τ : D.what₂.Tau) : Antitone (lamW D u i τ)

theorem **sSharp_indep** (u : D.what₂.Word) (μ μ' : D.what₂.Tup) (i : ℕ) (h : i < D.R.m) (τ : D.what₂.Tau) : (D.chTw u μ).sSharp i τ = (D.chTw u μ').sSharp i τ

theorem **sDiv_indep** (u : D.what₂.Word) (μ μ' : D.what₂.Tup) (i : ℕ) (h : i < D.R.m) (τ : D.what₂.Tau) : (D.chTw u μ).sDiv i τ = (D.chTw u μ').sDiv i τ

theorem **nuT_real** (u : D.what₂.Word) (i : ℕ) (hi : i < D.R.m) (τ : D.what₂.Tau) (r : Fin (D.R.N i)) : ((nuT D u i τ r : ℕ) : ℝ) = DV.nuRow D u (i + 1) τ r

theorem **clause1** (u : D.what₂.Word) (i : ℕ) (hi : i < D.R.m) (τ : D.what₂.Tau) (r : Fin (D.R.N i)) : (lam hA u i τ).natW r + nn D u i τ = nuT D u i τ r

theorem **last_real** (u : D.what₂.Word) (μ : D.what₂.Tup) (i : ℕ) (hi : i + 1 < D.R.m) (τ : D.what₂.Tau) : ((nuT D u (i + 1) τ (lastI D (i + 1)) : ℕ) : ℝ) = (((D.chTw u μ).sSharp (i + 1) τ : ℤ) : ℝ) − (((D.chTw u μ).sDiv i τ : ℤ) : ℝ)

theorem **wtTie_meet** {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {K : Formalism.MainNotions.G.Consts} {Y : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl K Y cX} (t : KerEntTied ex) : t.WtTie (lam hA) (shh D) (nn D)

theorem **antitone_kernelChain_nuT** (D : D7) : ∀ u i τ, i < D.R.m → Antitone (nuT D u i τ)

theorem **wtTie_exists** {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {K : Formalism.MainNotions.G.Consts} {Y : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl K Y cX} (t : KerEntTied ex) : ∃ lam sh n, t.WtTie lam sh n

theorem **vqTieU_of** (t : KerEntTied ex) {lam : ∀ (_ : D.what₂.Word) (i : ℕ), D.what₂.Tau → DomWeight (D.R.N i)} {sh : D.what₂.Word → ℕ → D.what₂.Tau → ℤ} (h : t.VQTie lam sh) : VQTieU t lam sh

theorem **kmapRat_of_vqU** (t : KerEntTied ex) (lam : ∀ (_ : D.what₂.Word) (i : ℕ), D.what₂.Tau → DomWeight (D.R.N i)) (sh : D.what₂.Word → ℕ → D.what₂.Tau → ℤ) (n : D.what₂.Word → ℕ → D.what₂.Tau → ℕ) (hVQ : VQTieU t lam sh) (hW : t.WtTie lam sh n) (hRK : t.RK) : OrbitPlaces.KmapRat t.ent

theorem **kmapRat_of_ex** (t : KerEntTied ex) (h : ∃ lam sh n, VQTieU t lam sh ∧ t.WtTie lam sh n) (hRK : t.RK) : OrbitPlaces.KmapRat t.ent

theorem **kmapRat_of_vq** (t : KerEntTied ex) (hVQ : VQTieU t (lam (antitone_kernelChain_nuT D)) (shh D)) (hRK : t.RK) : OrbitPlaces.KmapRat t.ent

theorem **kmapRat_of_vqAll** (t : KerEntTied ex) (hVQ : t.VQTie (lam (antitone_kernelChain_nuT D)) (shh D)) (hRK : t.RK) : OrbitPlaces.KmapRat t.ent

theorem **emb_mem_Fi** (M : MemberForms LL embLL Pl Tau hc C A P) (i : ℕ) (x : (M.lat i).V) : M.emb i x ∈ M.Fi i

theorem **fi_of_kRat** (M : MemberForms LL embLL Pl Tau hc C A P) (i : ℕ) (h : M.KRat i) : ∀ φ ∈ M.Fi i, C.K i φ ∈ M.Fi (i + 1)

theorem **K_heq** {R : Ranks} {ι : Type} [Fintype ι] {hc : ι → HCParam R} {C1 C2 : ColumnChain R ι hc} (h : C1 = C2) (i : ℕ) {f1 : C1.cls i → C1.Y i → ℂ} {f2 : C2.cls i → C2.Y i → ℂ} (hf : HEq f1 f2) : HEq (C1.K i f1) (C2.K i f2)

theorem **kernelRational_referenceTuple** (D : D7) (T : TwForms D) (u : D.what₂.Word) (i : ℕ) (hi : i < D.R.m) (h : (D.fm u).KRat i) : (T.fm u D.what₂.mu0).KRat i

theorem **kFi_of_kRat** (D : D7) (T : TwForms D) (h0 : ∀ u, ∀ i < D.R.m, (D.fm u).KRat i) (h1 : ∀ u μ, DV.Used D μ → ∀ i < D.R.m, (T.fm u μ).KRat i) : OrbitPlaces.KFi D T

theorem **natW_succ'** (hA : ∀ u i τ, i < D.R.m → Antitone (nuT D u i τ)) (u : D.what₂.Word) (μ : D.what₂.Tup) (i : ℕ) (hi : i < D.R.m) (τ : D.what₂.Tau) (r : Fin (D.R.N (i + 1))) : (lam hA u (i + 1) τ).natW r = extNu (D.R.N (i + 1)) (fun τ r => (lam hA u i τ).natW r + nn D u i τ) τ r

theorem **natW_succ** (hA : ∀ u i τ, i < D.R.m → Antitone (nuT D u i τ)) (u : D.what₂.Word) (i : ℕ) (hi : i < D.R.m) (τ : D.what₂.Tau) (r : Fin (D.R.N (i + 1))) : (lam hA u (i + 1) τ).natW r = extNu (D.R.N (i + 1)) (nuT D u i) τ r

theorem **VV_eq_WtPol** {K : Type} [Field K] [Infinite K] {ι : Type} [Fintype ι] {n : ℕ} (lam : ι → DomWeight n) : Formalism.L2.DIV.VV K lam = WtPol K (fun τ r => (lam τ).natW r)

theorem **prod_zpow_split** {ι : Type} [Fintype ι] (d : ι → ℂ) (hd : ∀ τ, d τ ≠ 0) (s : ι → ℤ) (k : ι → ℕ) : ∏ τ, d τ ^ ((k τ : ℤ) − s τ) = (∏ τ, d τ ^ (−(s τ))) * ∏ τ, d τ ^ k τ

theorem **evFr_map** (D : D7) {N : ℕ} {K : Type} [Field K] (σ : K →+* ℂ) (Q : MvPolynomial (GL3Theta.FrV D N) K) (y : D.what₂.Tau → Matrix (Fin N) (Fin N) ℂ) : GL3Theta.evFr D (MvPolynomial.map σ Q) y = MvPolynomial.eval₂ σ (fun k : GL3Theta.FrV D N => y k.1 k.2.1 k.2.2) Q

theorem **range_evC** {D : D7} (O : ∀ u μ, GL3Theta.LiftObj D (D.chTw u μ)) (u : D.what₂.Word) (μ : D.what₂.Tup) (i : ℕ) : Set.range (evC O u μ i) = Set.range (evP O u μ i)

theorem **vqTie_kerEntC** (hD : WtDiv) (t : KerEntTied ex) (P : Examples₂.EntPl D) (B : PlaceMatrixFamily D) (hK : KerEntHyp P t.O B Y ex.tw) (he : t.ent = kerEntC P t.O B hK) : t.VQTie (lam (antitone_kernelChain_nuT D)) (shh D)

theorem **kmapRat_kerEntC** (hD : WtDiv) (t : KerEntTied ex) (P : Examples₂.EntPl D) (B : PlaceMatrixFamily D) (hK : KerEntHyp P t.O B Y ex.tw) (he : t.ent = kerEntC P t.O B hK) (hRK : t.RK) : OrbitPlaces.KmapRat t.ent

theorem **gdet_dvd_sq** {n : ℕ} (I : Fin n → Fin n) (hs : n ≤ n) : gdet K n ∣ Formalism.GradedWeylDivisibility.initMinor K I hs

theorem **gdet_dvd_initMinor** {n s : ℕ} (I : Fin s → Fin n) (hs : s ≤ n) (he : s = n) : gdet K n ∣ Formalism.GradedWeylDivisibility.initMinor K I hs

theorem **gdet_pow_dvd_bidet** {n N : ℕ} (sz : Fin N → ℕ) (hsz : ∀ i, sz i ≤ n) (I : ∀ i, Fin (sz i) → Fin n) (k : ℕ) (hk : k ≤ (Finset.univ.filter fun i => sz i = n).card) : gdet K n ^ k ∣ Formalism.GradedWeylDivisibility.bideterminant K sz hsz I

theorem **shapeDeg_last** {n N : ℕ} (hn : 0 < n) (sz : Fin N → ℕ) (hsz : ∀ i, sz i ≤ n) : Formalism.GradedWeylDivisibility.shapeDeg sz ((n − 1, by omega) : Fin n) = (Finset.univ.filter fun i => sz i = n).card

theorem **gdet_pow_dvd_dualWeyl** {n : ℕ} (μ : Fin n → ℕ) (k : ℕ) (hk : ∀ r, k ≤ μ r) : ∀ F ∈ Formalism.GradedWeylDivisibility.dualWeyl K n n μ, gdet K n ^ k ∣ F

theorem **blockEmb_gdet** {ι : Type} {n : ℕ} (τ : ι) : Formalism.GradedWeylDivisibility.blockEmb K τ (gdet K n) = frDet K n τ

theorem **frDet_ne_zero'** {ι : Type} {a : ℕ} (τ : ι) : frDet K a τ ≠ 0

theorem **dk_dvd_wtPol** [Infinite K] {ι : Type} [Fintype ι] {n : ℕ} (ν : ι → Fin n → ℕ) (k : ι → ℕ) (hk : ∀ τ r, k τ ≤ ν τ r) : ∀ Q ∈ WtPol K ν, (∏ τ, frDet K n τ ^ k τ) ∣ Q

theorem **frameWeightPolyDivByDet_proof** : WtDiv

theorem **vqTie_kerEntC'** (t : KerEntTied ex) (P : Examples₂.EntPl D) (B : PlaceMatrixFamily D) (hK : KerEntHyp P t.O B Y ex.tw) (he : t.ent = kerEntC P t.O B hK) : t.VQTie (lam (antitone_kernelChain_nuT D)) (shh D)

theorem **kmapRat_kerEntC'** (t : KerEntTied ex) (P : Examples₂.EntPl D) (B : PlaceMatrixFamily D) (hK : KerEntHyp P t.O B Y ex.tw) (he : t.ent = kerEntC P t.O B hK) (hRK : t.RK) : OrbitPlaces.KmapRat t.ent

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.LocSplitProdRight

theorem **locSplit_prod_left** {G H : Type} [Group G] [Group H] (A : Subgroup G) (B : Subgroup H) : LocSplit (A.prod B) ((⊤ : Subgroup G).prod ⊥)

theorem **LocSplit.map** {G H : Type} [Group G] [Group H] {J L : Subgroup G} (h : LocSplit J L) (e : G ≃* H) : LocSplit (J.map e.toMonoidHom) (L.map e.toMonoidHom)

theorem **LocSplit.conj_mem** {G : Type} [Group G] {J L : Subgroup G} (h : LocSplit J L) {k l : G} (hk : k ∈ J) (hl : l ∈ L) : k * l * k⁻¹ ∈ L

theorem **sum_mul_reps** (ρ : Γ →* Module.End ℂ W) {J L : Subgroup G} {χ : J →* ℂˣ} (hχ : ∀ k : J, (k : G) ∈ L → χ k = 1) {x : G} {A B : Finset G} (hA : IsDC J x A) (hB : IsDC J x B) (hAL : ∀ a ∈ A, a ∈ L) (hBL : ∀ b ∈ B, b ∈ L) {F : G → W} (hF : F ∈ AdForms ι ρ J χ) {k : G} (hkJ : k ∈ J) (hkL : k ∈ L) (h : G) : ∑ a ∈ A, F (h * k * a) = ∑ b ∈ B, F (h * b)

theorem **dcOp_eq_of_isDC** (ρ : Γ →* Module.End ℂ W) {J L : Subgroup G} {χ : J →* ℂˣ} (hχ : ∀ k : J, (k : G) ∈ L → χ k = 1) {x : G} {A B : Finset G} (hA : IsDC J x A) (hB : IsDC J x B) (hAL : ∀ a ∈ A, a ∈ L) (hBL : ∀ b ∈ B, b ∈ L) {F : G → W} (hF : F ∈ AdForms ι ρ J χ) : dcOp A F = dcOp B F

theorem **LocSplit.dcOp_mem** (ρ : Γ →* Module.End ℂ W) {J L : Subgroup G} {χ : J →* ℂˣ} (hs : LocSplit J L) (hχ : ∀ k : J, (k : G) ∈ L → χ k = 1) {x : G} {A : Finset G} (hA : IsDC J x A) (hAL : ∀ a ∈ A, a ∈ L) {F : G → W} (hF : F ∈ AdForms ι ρ J χ) : dcOp A F ∈ AdForms ι ρ J χ

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.MatchOfMatchOffTEmpty

theorem **nonSplitSet_empty** : NonSplitSet (∅ : Finset (HeightOneSpectrum (𝒪 E)))

theorem **matchOffT_of_match** {U : Submodule ℂ (UA E a → ℂ)} {π : CuspSub E a} (h : Match U π) (T : Finset (HeightOneSpectrum (𝒪 E))) : MatchOffT U π T

theorem **descOffT_of_desc** (h : Desc) : DescOffT

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.MemMoveLevUloc

theorem **mem_basePts_iff** (k : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) : k ∈ basePts S ↔ k ∈ UPts σ J (FiniteAdeleRing (𝒪 F) F) ∧ (∀ v, bc (prV (E

theorem **basePt_loc** (v : HeightOneSpectrum (𝒪 F)) : bc (prV (E

theorem **conj_mem_iff** {G : Type} [Group G] {H : Subgroup G} {a : G} (ha : a ∈ H) (x : G) : a * x * a⁻¹ ∈ H ↔ x ∈ H

theorem **mem_moveLev** (x : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) : x ∈ (moveLev k).U ↔ k.1 * x * k.1⁻¹ ∈ S.U

theorem **moveLev_chi** (x : (moveLev k).U) : (moveLev k).chi x = S.chi (k.1 * x.1 * k.1⁻¹, x.2)

theorem **chi_toMoved** (κ : S.U) : (moveLev k).chi (toMoved k κ) = S.chi κ

theorem **basePt_hatStab** : HatStab S.Lb k.1

theorem **latAd_basePt** [StarRing E] (x : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) : latAd (x * k.1) S.Lb = latAd x S.Lb

theorem **cls_move_iff** (x y : UPts σ J (FiniteAdeleRing (𝒪 F) F)) : (Quotient.mk'' (x * basePtU k) : Cls (moveLev k)) = Quotient.mk'' (y * basePtU k) ↔ (Quotient.mk'' x : Cls S) = Quotient.mk'' y

theorem **mem_hat_move_iff** (y : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (x : Fin n → E ⊗[F] FiniteAdeleRing (𝒪 F) F) : (y⁻¹).1 *ᵥ x ∈ latS (F

theorem **map_conj_move** (y : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) : (moveLev k).U.map (MulAut.conj (y * k.1)).toMonoidHom = S.U.map (MulAut.conj y).toMonoidHom

theorem **bij_move** {cl : Type} {y : cl → UPts σ J (FiniteAdeleRing (𝒪 F) F)} (h : Function.Bijective fun c => (Quotient.mk'' (y c) : Cls S)) : Function.Bijective fun c => (Quotient.mk'' (y c * basePtU k) : Cls (moveLev k))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.MemberChainFormsLevelPrimeEqMember

theorem **sameU_of** (Fo : FormsIn M (i + 1) hi) (β : Fo.Ch) : SameUlocJ hi (Fo.K β).lev (ulocOf Fo β)

theorem **rank_of** (X0 : Formalism.L2.Herm.Herm0 D) (Fl : Fills (D.gat u w hg)) (ht : Fl.t = D.R.t) (i : ℕ) (hi : i < D.R.m) : (D.gat u w hg).P.m + (i + 1) * (Fl.t + 1) + 1 = (D.sp.X (i + 1)).rank

theorem **lv_empty** (j : ℕ) (hj : D.R.m < j) : IsEmpty ((Z.lv Fl ht).comp j)

theorem **loc_B** (i : ℕ) (hi : i + 1 ≤ D.R.m) : (Z.loc Fl ht i hi).B = (Z.S i hi).B

theorem **loc_psi** (i : ℕ) (hi : i + 1 ≤ D.R.m) : (Z.loc Fl ht i hi).ψ = (Z.S i hi).ψ

theorem **loc_Uloc** (i : ℕ) (hi : i + 1 ≤ D.R.m) : (Z.loc Fl ht i hi).Uloc = fun β => ulocOf (Z.Fo i hi) β

theorem **loc_Lb** (i : ℕ) (hi : i + 1 ≤ D.R.m) : (Z.loc Fl ht i hi).Lb = fun β => lbOf (Z.Fo i hi) β

theorem **frRat** (i : ℕ) (hi : i + 1 ≤ D.R.m) (β : (Z.Fo i hi).Ch) : FrRatJ ((Z.loc Fl ht i hi).t β) (Z.loc Fl ht i hi).B

theorem **sameU** (i : ℕ) (hi : i + 1 ≤ D.R.m) (β : (Z.Fo i hi).Ch) : SameUlocJ hi ((Z.Fo i hi).K β).lev ((Z.loc Fl ht i hi).Uloc β)

theorem **SiteD.inMem_st** (Z : SiteD X0 u w hg (D.fm u)) (Tw : TwForms D) (Fl : Fills (D.gat u w hg)) (ht : Fl.t = D.R.t) (Gt : GatTies D) (j : ℕ) (hj : j + 1 ≤ D.R.m) : ((Z.inMem Tw Fl ht Gt).at' j hj).st = Z.st Fl ht Gt j hj

theorem **finite_compG_re** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {a b : ℕ} (h : a = b) (σ : E →ₐ[F] E) (J : Matrix (Fin a) (Fin a) E) (J' : Matrix (Fin b) (Fin b) E) (hJ : ∀ x y, J' x y = J (Fin.cast h.symm x) (Fin.cast h.symm y)) (Uloc' : ∀ v : HeightOneSpectrum (𝒪 F), Subgroup (GL (Fin b) (E ⊗[F] v.adicCompletion F))) (p : ℕ) (hf : Finite (compG σ J' Uloc' p)) : Finite (compG σ J (fun v => (Uloc' v).comap (reGL h _).toMonoidHom) p)

theorem **fin_of** (Fo : FormsIn M (i + 1) hi) (β : Fo.Ch) (p : ℕ) : Finite (compG D.sp.cm.sig (D.sp.X (i + 1)).gram (ulocOf Fo β) p)

theorem **formsInMem_top** (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) (h : i + 1 = D.R.m) : formsInMem F dt i hi = formsInOne (F.val (i + 1) hi (Or.inr h)) (dt.top (i + 1) hi h)

theorem **formsInMem_mid** (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) (h : i + 1 + 1 ≤ D.R.m) : formsInMem F dt i hi = formsInMid F (i + 1) (one_le i) h (dt.mid (i + 1) (one_le i) h)

theorem **formsInMem_fin** [∀ i, Finite (Ch i)] (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) : Finite (formsInMem F dt i hi).Ch

theorem **fl_t** (ex : Extra D L hT Gi Gl K Dx cX) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : (Gl.Fl u w hg).t = D.R.t

theorem **StepD.ofEx_std** {Fo : FormsIn M (i + 1) hi} {e : D.what₂.LL →+* (D.gat u w hg).Kf} (hB : BaseEx X0 u w hg Fo) (hψ : PsiEx Fo e) : cT (sigD D : D.host.E →+* D.host.E) (reGL (rankN D (i + 1) hi) D.host.E (StepD.ofEx hB hψ).B).1 * gramN D (i + 1) hi * (reGL (rankN D (i + 1) hi) D.host.E (StepD.ofEx hB hψ).B).1 = 1

theorem **StepD.ofEx_psi** {Fo : FormsIn M (i + 1) hi} {e : D.what₂.LL →+* (D.gat u w hg).Kf} (hB : BaseEx X0 u w hg Fo) (hψ : PsiEx Fo e) (β : Fo.Ch) : UnitCharacterChartCompatibility hi (Fo.K β).lev e ((StepD.ofEx hB hψ).ψ β)

theorem **fillOK_of_empty** (hE : ∀ j, D.R.m < j → IsEmpty (Lv.comp j)) : FillOK μ Lv

theorem **fillOK_of_nonempty** (hN : ∀ j, D.R.m < j → Nonempty ((D.chTw u μ).cls j)) : FillOK μ Lv

theorem **labHyp_of** {I : CmpJIn Tw u μ w h Lv} (X : LabIn I) : LabHyp I

theorem **labHyp_atInM** (Z : AtInM X0 u w h Lv) (Tw : TwForms D) (Gt : GatTies D) (hZ : ChartLocalForms Z.Fo Z.Lc) (hF : FillOK D.what₂.mu0 Lv) : LabHyp (Z.I Tw Gt) where

theorem **fl_t_eq** (ex : Extra D L hT Gi Gl K Dx cX) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : (Gl.Fl u w hg).t = D.R.t

theorem **fl_mm_eq** (ex : Extra D L hT Gi Gl K Dx cX) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : (Gl.Fl u w hg).mm = D.R.m

theorem **memberChainForms_levelPrime_eq_member** {D : D7} {u : D.what₂.Word} {Ch : ℕ → Type} {chOf : ∀ i, (D.mem u).C.cls i → Ch i} (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) : (formsInMem F dt i hi).T = (D.mem u).A.T

theorem **reL_symm** {a b : ℕ} (h : a = b) {E : Type} [Field E] [NumberField E] (L : Submodule (𝒪 E) (Fin b → E)) : reL h (reL h.symm L) = L

theorem **formsInMem_bad** {u : D.what₂.Word} {Ch : ℕ → Type} {chOf : ∀ i, (D.mem u).C.cls i → Ch i} (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) : (formsInMem F dt i hi).bad = badG D

theorem **formsInMem_ok** {u : D.what₂.Word} {Ch : ℕ → Type} {chOf : ∀ i, (D.mem u).C.cls i → Ch i} (F : FormsTiedSym D u Ch chOf) (dt : MemDt F) (i : ℕ) (hi : i + 1 ≤ D.R.m) (w : D.what₂.places.Pl) : (formsInMem F dt i hi).okW w

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.MonodromyZeroIffRestricts

theorem **restricts_iff_wdRel** (x : WD h) (x' : WD h') : Restricts x x' ↔ WDRel (RingEquiv.refl Ω) x x'

theorem **restricts_symm** {x : WD h} {x' : WD h'} : Restricts x x' → Restricts x' x

theorem **lpDim_eq** {x : WD h} {x' : WD h'} (H : Restricts x x') : x'.lpDim = x.lpDim

theorem **monodromyZero_iff_restricts** {x : WD h} {x' : WD h'} (H : Restricts x x') : x'.mono0 ↔ x.mono0

theorem **not_restricts_sp** (c : Fin 2 → ℂˣ) : ¬ Restricts (WD.lpSp h 2) (WD.mk (WDRep.diag h' c))

theorem **ineTriv_mk** {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (a : WDRep h V) : IneTriv (WD.mk a) ↔ ∀ τ : Ine (F

theorem **unr_iff** (x : WD h) : x.unr ↔ x.mono0 ∧ IneTriv x

theorem **ineTriv_sp** : IneTriv (WD.lpSp h 2) ∧ ¬ (WD.lpSp h 2).unr

theorem **ineTriv_ofChar** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : IneTriv (WD.lpOfChar h χ hχ) ↔ ∀ τ : Ine (F

theorem **continuous_restr** : Continuous (AlgEquiv.restrictScalarsHom K : G L Ω →* G K Ω)

theorem **normV_pow** : normV v' = normV v ^ v'.asIdeal.inertiaDeg (𝒪 K)

theorem **actsAs_restr** {σ : G L Ω} {a b : ℕ} (H : ActsAs A (normV v') σ a b) : ActsAs A (normV v) (σ.restrictScalars K : G K Ω) (v'.asIdeal.inertiaDeg (𝒪 K) * a) (v'.asIdeal.inertiaDeg (𝒪 K) * b)

theorem **deg_weilRes** (σ : Weil h') : deg (weilRes h h' σ) = v'.asIdeal.inertiaDeg (𝒪 K) * deg σ

theorem **restricts_mk_iff** {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (a : WDRep h V) (x' : WD h') : Restricts (WD.mk a) x' ↔ x' = WD.mk (repRes h' a)

theorem **existsUnique_restricts** (x : WD h) : ∃! x' : WD h', Restricts x x'

theorem **restricts_diag** {n : ℕ} (c : Fin n → ℂˣ) : Restricts (WD.mk (WDRep.diag h c)) (WD.mk (WDRep.diag h' fun i => c i ^ v'.asIdeal.inertiaDeg (𝒪 K)))

theorem **restricts_sp** (n : ℕ) : Restricts (WD.lpSp h n) (WD.lpSp h' n)

theorem **IneTriv.restricts** {x : WD h} {x' : WD h'} (H : Restricts x x') (hx : IneTriv x) : IneTriv x'

theorem **resTo_resTo** {K L M Ω : Type} [Field K] [Field L] [Field M] [Field Ω] [Algebra K Ω] [Algebra L Ω] [Algebra M Ω] [Algebra K L] [Algebra L M] [Algebra K M] [IsScalarTower K L Ω] [IsScalarTower L M Ω] [IsScalarTower K M Ω] {X : Type} [Monoid X] (r : G K Ω →* X) : resTo M (resTo L r) = resTo M r

theorem **irr_one** {Γ k : Type} [Group Γ] [Field k] (r : Γ →* GL (Fin 1) k) : Irr r

theorem **not_irr_two** {Γ k : Type} [Group Γ] [Field k] : ¬ Irr (1 : Γ →* GL (Fin 2) k)

theorem **irr_std** {k : Type} [Field k] : Irr (MonoidHom.id (GL (Fin 2) k))

theorem **regPol_triv** : RegPol (CuspSub.triv E (le_refl 1))

theorem **RepOf.regPol** {π : CuspSub E a} {ι : Qbar l ≃+* ℂ} {r : G E Ω →* GL (Fin a) (Qbar l)} (H : RepOf π ι r) : RegPol π

theorem **exists_repOf** (H : GlobalTheory.Published.PolarizedGaloisRepPublished) (ha : 1 ≤ a) {π : CuspSub E a} (hπ : RegPol π) (ι : Qbar l ≃+* ℂ) : ∃ r : G E Ω →* GL (Fin a) (Qbar l), RepOf π ι r

theorem **exists_isAut** (H : GlobalTheory.Published.PolarizedGaloisRepPublished) (ι : Qbar l ≃+* ℂ) : ∃ r : G E Ω →* GL (Fin 1) (Qbar l), IsAut ι r

theorem **not_repOf_cyc** (ι : Qbar l ≃+* ℂ) {w : HeightOneSpectrum (𝒪 E)} (hl : ((l : ℕ) : 𝒪 E) ∉ w.asIdeal) {A : ValuationSubring Ω} (h : IsPlaceOver A w) {rc} (hrc : IsTheRec h rc) : ¬ RepOf (CuspSub.triv E (le_refl 1)) ι ((Matrix.GeneralLinearGroup.scalar (Fin 1)).comp (cyc E Ω l))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NatCastMemOfPlace

theorem **DatLoc.comp** {Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) (f : ℂ → ℂ) : DatLoc D cc SS cl N N' fun x => f (Ξ x)

theorem **DatLoc.const_mul** {Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) (k : ℂ) : DatLoc D cc SS cl N N' fun x => k * Ξ x

theorem **DatLoc.add** {Ξ Ξ' : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) (h' : DatLoc D cc SS cl N N' Ξ') : DatLoc D cc SS cl N N' fun x => Ξ x + Ξ' x

theorem **DatLoc.mul** {Ξ Ξ' : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) (h' : DatLoc D cc SS cl N N' Ξ') : DatLoc D cc SS cl N N' fun x => Ξ x * Ξ' x

theorem **DatLoc.of_locP_eq** {M : Submodule (𝒪 D.host.E) (Fin a → D.host.E)} {M' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)} {Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) (hM : ∀ P ∈ SS, Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (aaOf D cc) M M') = Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (aaOf D cc) N N')) : DatLoc D cc SS cl M M' Ξ

theorem **DatLoc.add_halfBO** {Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) (x m : Formalism.L2.Col.HomE D.host.E a b) (hm : m ∈ halfBO SS (perO (aaOf D cc) N N') cl) : Ξ (x + m) = Ξ x

theorem **DatLoc.invOn** {Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ} (h : DatLoc D cc SS cl N N' Ξ) {ι : Type} [Fintype ι] (emb : ι → D.host.E →+* ℂ) (δ : ℂ) (hinj : Function.Injective (embH (a

theorem **KapLawLt.congr** {Th Th' : ∀ j, X.Lt j → X.Lt (j + 1) → MvPowerSeries (X.ZA j) Kc} (h : KapLawLt X Kc Th) (he : ∀ j < X.mm, ∀ (La : X.Lt j) (La' : X.Lt (j + 1)), (X.Pj j).IsLatTup La.2 → (X.Pj (j + 1)).IsLatTup La'.2 → Th' j La La' = Th j La La') : KapLawLt X Kc Th'

theorem **KapLawLt.stab** {Th : ∀ j, X.Lt j → X.Lt (j + 1) → MvPowerSeries (X.ZA j) Kc} (h : KapLawLt X Kc Th) {j : ℕ} (hj : j < X.mm) (c : X.Lv.comp j) (L : (X.Pj j).Tup) (La' : X.Lt (j + 1)) (ε : (X.Lv.lev j c).Γ) (hL : (X.Pj j).IsLatTup L) (hL' : (X.Pj (j + 1)).IsLatTup La'.2) (hε : (X.Lv.lev j c).smul ε L = L) : MvPowerSeries.C ((kapG X Kc j c ε)⁻¹) * SA X Kc j (gamAt X j c ε) (Th j (c, L) La') = Th j (c, L) La'

theorem **frecG_eq** (c : ℂ_[p]) (j : ℕ) (La : X.Lt j) (La' : X.Lt (j + 1)) : FrecG E pos fd B χ lam SS cl Ξ₀ muT c j La La' = linSub (fun x y => c * subBG E pos fd j La La' x y) (pPow X j La La' • serP lam fun z => (scal0 X fd χ muT j La La' * (Base.chiFac χ _ _ (fd.kerN j La) (fd.kerN (j + 1) La'))⁻¹) * kerTh B χ SS cl (fd.kerN j La) (fd.kerN (j + 1) La') (xiM X fd Ξ₀ j La La') z)

theorem **pPow_eq_prod** (j : ℕ) (La : X.Lt j) (La' : X.Lt (j + 1)) : pPow X j La La' = ∏ v : X.Pl, (p : ℂ_[p]) ^ (X.vm v (j + 1) * (len X v (j + 1) La'.2 - len X v j La.2))

theorem **twG_eq_tw** (v : X.Pl) (s : Fin (X.G.P.n' v)) : letI

theorem **LamPos.even** (h : LamPos X E pos lam ρ) (v : X.Pl) (s : Fin (X.G.P.n' v)) (hs : ¬ IsOddPos s) (x : D.host.E) : TaylorLogMetric.lamC lam (TaylorLogMetric.embT D (pos ⟨v, s⟩) x) = twG X E v s (ρ v x)

theorem **LamPos.odd** (h : LamPos X E pos lam ρ) (v : X.Pl) (s : Fin (X.G.P.n' v)) (hs : IsOddPos s) (x : D.host.E) : TaylorLogMetric.lamC lam (TaylorLogMetric.embT D (pos ⟨v, s⟩) x) = twG X E v s (ρ v (D.host.conj x))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NegOnePowBoundaryTermsCancel

theorem **mult_mul** (T : Thick L) (x y : T.B) : (mult L).mul T x y = x * y

theorem **ev_apply** {n k : ℕ} (s : ComposableArrows C k) (p : Fin (n + 1) → Fin (k + 1)) (hp : Monotone p) (j : Fin (k + 1)) (h : p (Fin.last n) ≤ j) (x : Cn F n) : ev s p hp j h x = (F.map (s.map (homOfLE h))).hom (x (hp.functor ⋙ s))

theorem **ev_congr** {n k : ℕ} (s : ComposableArrows C k) (p q : Fin (n + 1) → Fin (k + 1)) (hp : Monotone p) (hq : Monotone q) (hpq : ∀ a, (p a : ℕ) = q a) (j : Fin (k + 1)) (h : p (Fin.last n) ≤ j) (h' : q (Fin.last n) ≤ j) (x : Cn F n) : ev s p hp j h x = ev s q hq j h' x

theorem **map_ev** {n k : ℕ} (s : ComposableArrows C k) (p : Fin (n + 1) → Fin (k + 1)) (hp : Monotone p) {j j' : Fin (k + 1)} (h : p (Fin.last n) ≤ j) (hj : j ≤ j') (x : Cn F n) : (F.map (s.map (homOfLE hj))).hom (ev s p hp j h x) = ev s p hp j' (h.trans hj) x

theorem **ev_sub** {n m k : ℕ} (s : ComposableArrows C k) (q : Fin (m + 1) → Fin (k + 1)) (hq : Monotone q) (p : Fin (n + 1) → Fin (m + 1)) (hp : Monotone p) (j : Fin (m + 1)) (h : p (Fin.last n) ≤ j) (x : Cn F n) : ev (hq.functor ⋙ s) p hp j h x = ev s (q ∘ p) (hq.comp hp) (q j) (hq h) x

theorem **d_apply_ev** {n : ℕ} (x : Cn F n) (s : ComposableArrows C (n + 1)) : d F n x s = ∑ i : Fin (n + 2), (-1 : ℤ) ^ (i : ℕ) • ev s (Fin.succAboveOrderEmb i) (Fin.succAboveOrderEmb i).monotone (Fin.last _) (Fin.le_last _) x

theorem **ev_d** {n k : ℕ} (s : ComposableArrows C k) (p : Fin (n + 2) → Fin (k + 1)) (hp : Monotone p) (j : Fin (k + 1)) (h : p (Fin.last (n + 1)) ≤ j) (x : Cn F n) : ev s p hp j h (d F n x) = ∑ i : Fin (n + 2), (-1 : ℤ) ^ (i : ℕ) • ev s (p ∘ Fin.succAboveOrderEmb i) (hp.comp (Fin.succAboveOrderEmb i).monotone) j ((hp (Fin.le_last _)).trans h) x

theorem **fr_mono** (m k : ℕ) (h : m ≤ k) : Monotone (fr m k h)

theorem **bk_mono** (m n k : ℕ) (h : m + n ≤ k) : Monotone (bk m n k h)

@[simp] theorem **val_fr** (m k : ℕ) (h : m ≤ k) (a : Fin (m + 1)) : (fr m k h a : ℕ) = a

@[simp] theorem **val_bk** (m n k : ℕ) (h : m + n ≤ k) (b : Fin (n + 1)) : (bk m n k h b : ℕ) = m + b

theorem **val_succAbove_lt** {n : ℕ} (i : Fin (n + 2)) (a : Fin (n + 1)) (h : (a : ℕ) < i) : (i.succAbove a : ℕ) = a

theorem **val_succAbove_ge** {n : ℕ} (i : Fin (n + 2)) (a : Fin (n + 1)) (h : (i : ℕ) ≤ a) : (i.succAbove a : ℕ) = a + 1

theorem **cupC_apply** {m n k : ℕ} (h : m + n = k) (f : Cn F m) (g : Cn F n) (s : ComposableArrows C k) : cupC μ h f g s = μ.mul _ (ev s (fr m k (by omega)) (fr_mono _ _ _) (Fin.last k) (Fin.le_last _) f) (ev s (bk m n k h.le) (bk_mono _ _ _ _) (Fin.last k) (Fin.le_last _) g)

theorem **map_cupC_sub** {m n k K : ℕ} (h : m + n = k) (f : Cn F m) (g : Cn F n) (s : ComposableArrows C K) (p : Fin (k + 1) → Fin (K + 1)) (hp : Monotone p) (j : Fin (K + 1)) (hj : p (Fin.last k) ≤ j) : (F.map (s.map (homOfLE hj))).hom (cupC μ h f g (hp.functor ⋙ s)) = μ.mul _ (ev s (p ∘ fr m k (by omega)) (hp.comp (fr_mono _ _ _)) j ((hp (Fin.le_last _)).trans hj) f) (ev s (p ∘ bk m n k h.le) (hp.comp (bk_mono _ _ _ _)) j ((hp (Fin.le_last _)).trans hj) g)

theorem **mul_ev_congr** {m n k : ℕ} (s : ComposableArrows C k) (j : Fin (k + 1)) (f : Cn F m) (g : Cn F n) {p p' : Fin (m + 1) → Fin (k + 1)} {hp : Monotone p} {hp' : Monotone p'} {q q' : Fin (n + 1) → Fin (k + 1)} {hq : Monotone q} {hq' : Monotone q'} {h1 h1' h2 h2'} (e1 : ∀ a, (p a : ℕ) = p' a) (e2 : ∀ b, (q b : ℕ) = q' b) : μ.mul _ (ev s p hp j h1 f) (ev s q hq j h2 g) = μ.mul _ (ev s p' hp' j h1' f) (ev s q' hq' j h2' g)

theorem **add_rearrange_of_cancel_middle** {M : Type*} [AddCommGroup M] {A B C c d D : M} (h1 : A = C) (h2 : B = D) (h3 : c + d = 0) : A + B = C + c + (d + D)

theorem **neg_one_pow_boundary_terms_cancel** {M : Type*} [AddCommGroup M] (m : ℕ) {X Y : M} (h : X = Y) : (-1 : ℤ) ^ (m + 1) • X + (-1 : ℤ) ^ m • ((-1 : ℤ) ^ 0 • Y) = 0

theorem **d_cupC** {m n k : ℕ} (h : m + n = k) (f : Cn F m) (g : Cn F n) : d F k (cupC μ h f g) = cupC μ (show m + 1 + n = k + 1 by omega) (d F m f) g + ((-1 : ℤ) ^ m) • cupC μ (show m + (n + 1) = k + 1 by omega) f (d F n g)

theorem **cupC_mem_ker** {m n k : ℕ} (h : m + n = k) {f : Cn F m} {g : Cn F n} (hf : d F m f = 0) (hg : d F n g = 0) : d F k (cupC μ h f g) = 0

theorem **cupC_mem_left** {m n k : ℕ} (h : m + n = k) {f : Cn F m} {g : Cn F n} (hf : f ∈ Bn F m) (hg : d F n g = 0) : cupC μ h f g ∈ Bn F k

theorem **cupC_mem_right** {m n k : ℕ} (h : m + n = k) {f : Cn F m} {g : Cn F n} (hf : d F m f = 0) (hg : g ∈ Bn F n) : cupC μ h f g ∈ Bn F k

@[simp] theorem **cupZ_val** {m n k : ℕ} (h : m + n = k) (f : LinearMap.ker (d F m)) (g : LinearMap.ker (d F n)) : (cupZ μ h f g : Cn F k) = cupC μ h f g

theorem **cup_mk** {m n k : ℕ} (h : m + n = k) (f : LinearMap.ker (d F m)) (g : LinearMap.ker (d F n)) : cup μ h (Submodule.Quotient.mk f) (Submodule.Quotient.mk g) = Submodule.Quotient.mk (cupZ μ h f g)

theorem **d_oneC** : d F 0 (oneC μ) = 0

theorem **oneH_ne_zero** (s : ComposableArrows C 0) (h : μ.one (s.obj (Fin.last 0)) ≠ 0) : oneH μ ≠ 0

theorem **τ_ev** (M : Mor F F' σ) {n k : ℕ} (s : ComposableArrows C' k) (p : Fin (n + 1) → Fin (k + 1)) (hp : Monotone p) (j : Fin (k + 1)) (h : p (Fin.last n) ≤ j) (x : Cn F n) : M.τ _ (ev (s ⋙ M.G) p hp j h x) = ev s p hp j h (M.pullC n x)

theorem **pullC_cupC** (M : Mor F F' σ) {μ : Mult F} {μ' : Mult F'} (hM : M.IsMult μ μ') {m n k : ℕ} (h : m + n = k) (f : Cn F m) (g : Cn F n) : M.pullC k (cupC μ h f g) = cupC μ' h (M.pullC m f) (M.pullC n g)

theorem **pull_cup** (M : Mor F F' σ) {μ : Mult F} {μ' : Mult F'} (hM : M.IsMult μ μ') {m n k : ℕ} (h : m + n = k) (x : H F m) (y : H F n) : M.pull k (cup μ h x y) = cup μ' h (M.pull m x) (M.pull n y)

theorem **Mor.pull_cup** {b' : Base} {L' : b'.Fun} {σ : b.R →+* b'.R} (M : Mor L L' σ) {m n k : ℕ} (h : m + n = k) (x : H L m) (y : H L n) : M.pull k (cup L h x y) = cup L' h (M.pull m x) (M.pull n y)

theorem **isDPLog_one** {B : Type} [CommRing B] {J : Ideal B} (δ : DividedPowers J) : IsDPLog δ 1 0

theorem **firstChernClass_trivialBundle** : IsFirstChernClassOfLineBundle (LB.triv : LB L) 0

theorem **Mor.pull_eq** {b' : Base} {L' : b'.Fun} {σ : b.R →+* b'.R} (M : Mor L L' σ) (n : ℕ) : M.pull n = M.catMor.pull n

theorem **IsFirstChernClassOfLineBundle.pull** {b' : Base} {L' : b'.Fun} {σ : b.R →+* b'.R} (M : Mor L L' σ) {E : LB L} {z : H L 2} (hz : IsFirstChernClassOfLineBundle E z) : IsFirstChernClassOfLineBundle (E.pull M) (M.pull 2 z)

theorem **one_node_ne_zero** : (one Ex.nodeL : H Ex.nodeL 0) ≠ 0

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NormMapOfFinOrder

theorem **norm_map_of_finOrder** {K : Type} [Field K] (σ : K →+* ℂ) {ζ : Kˣ} (h : IsOfFinOrder ζ) : ‖σ (ζ : K)‖ = 1

theorem **norm_muInvDet** {X : Fields} {Pr : Primes X} {R : Ranks} {Ch : Chars X Pr R.m} {sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)} {S : Set ℕ} {mu : Tup X.F X.E X.K S} {cst : LocCst X (rowSG Pr sel mu)} {L : ChIdx X Pr R (QE sel (mu : Chr X.F X.E X.K))} {G : ChGlob X R.m} {m2 : ℕ} {e : Fin R.m → X.Tau → ℚ} {eta : Formalism.L2.Host.HChar X.E} (hX : X.Ok) (hPr : Pr.Ok) (hsel : IsSel X sel) (hS : ∀ i, Pr.g i ∈ S) (ht : Ch.Tied G.σ₀ e G.mu0 eta) (hL : L.Ok Ch sel mu cst G m2) {j : ℕ} (hj : j < R.m) {n : ℕ} (N : Submodule (𝒪 X.E) (Sp X.E n)) (fr : Frs X.E n (rowSG Pr sel mu)) : ‖G.σ₀ (muInvDet sel (rowSG Pr sel mu) (mUof Ch mu j) (muAt Ch sel mu cst j) N fr)‖ = 1

theorem **norm_chiLat** {D : D7} {X : Fields} {s : Seam D X} {χ : TaylorLogMetric.IdE D → ℂ} (hX : X.Ok) (hs : s.Ok) (hχ : TaylorLogMetric.IsChi D χ) {n : ℕ} {N : Submodule (𝒪 X.E) (Sp X.E n)} (hN : IsSdLat N) : ‖chiLat s χ n N‖ = 1

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NotDvdCondN

theorem **inertUnr_iff** (D : D7) (p : ℕ) : InertUnr D p ↔ Ne p 2 ∧ p ≠ D.host.q ∧ p ≠ D.host.d ∧ jacobiSym (-(D.host.d : ℤ)) p ≠ 1

theorem **InertUnr.ne_two** {D : D7} {p : ℕ} (h : InertUnr D p) : p ≠ 2

theorem **InertUnr.not_dvd_condN** {D : D7} {p : ℕ} (h : InertUnr D p) (hp : p.Prime) : ¬ p ∣ D.host.condN

theorem **IsRowB.lt_orderOf** {D : D7} {L : SizeData₂ D} {p : ℕ} (h : IsRowB L p) : 4 * D.host.Lw < orderOf (DV.frobG D p).2

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NotGoodGenericSign

theorem **nb_congr** {E : Type} [Field E] [NumberField E] [StarRing E] {n : ℕ} {ST SG : Finset (HeightOneSpectrum (𝒪 E))} {P : Ideal (𝒪 E)} {s y : List ℕ} (f : inType P s ≃ₗ[𝒪 E] inType P y) (d : DecoSet E n ST SG) : d.inertSet P s = d.inertSet P y

theorem **op_zero_or** {K : Type} [Field K] [NumberField K] [StarRing K] {E : Type} [Field E] [NumberField E] [StarRing E] {n : ℕ} {ST SG : Finset (HeightOneSpectrum (𝒪 E))} {S : Col.GSD K} (c : Col.DecoPts S E n ST SG) (P : HeightOneSpectrum (𝒪 E)) (s : List ℕ) : (Col.heckeI c P.asIdeal s).op = 0 ∨ ∃ y, Col.InertTypeOk n y ∧ (Col.heckeI c P.asIdeal s).op = (Col.heckeI c P.asIdeal y).op

theorem **csd_zero_or** {K : Type} [Field K] [NumberField K] [StarRing K] {E : Type} [Field E] [NumberField E] [StarRing E] {n : ℕ} {ST SG' : Finset (HeightOneSpectrum (𝒪 E))} (S : Col.GSD K) (h : S.Ok) (c : Col.DecoPts S E n ST SG') (SG : Finset (HeightOneSpectrum (𝒪 E))) (S0 : Set (HeightOneSpectrum (𝒪 E))) (Q : offSubI ST SG SG' S0) (s : List ℕ) : heckeCSDI S h c SG S0 (Q, s) = 0 ∨ ∃ y, Col.InertTypeOk n y ∧ heckeCSDI S h c SG S0 (Q, s) = heckeCSDI S h c SG S0 (Q, y)

theorem **rank_rows** (R : Ranks) : R.N R.m − 1 = R.r + R.m ∗ R.t

theorem **inertJunk_col** {X : Col.Fields} {Pr : Col.Primes X} {R : Ranks} (W : Col.Wts X Pr.Word R) (Ch : Col.Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S : Set ℕ} (hX : X.Ok) (v0 : InfinitePlace X.K) (TU : Finset (Tup X.F X.E X.K S)) (Xg : Finset (Col.G0 X.E Pr.vGs R.r → X.K)) (S0 : Set (HeightOneSpectrum (𝒪 X.E))) (xiVal : Tup X.F X.E X.K S → HeightOneSpectrum (𝒪 X.E) → X.K×) (trIv : Fin R.m → (Col.LevOffI ∅ Pr.vGs S0 × List ℕ → X.K) → Col.LevOffI ∅ Pr.vGs S0 × List ℕ → X.K) : InertJunk (colIn3 W Ch sel hX v0 TU Xg S0 xiVal trIv) where

theorem **not_inertJunk** {K : Type} [Field K] [NumberField K] [StarRing K] {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {m : ℕ} (A : FamIn K G Pl0 T m) (t : T) [Nontrivial (A.MtildeCSD t).Forms] : ¬ InertJunk ((A, fun _ => Set.univ, fun _ k => if k.2 = [0] then (2 : K) • 1 else 1, fun _ _ _ => 1, fun _ e => e) : FamIn3 K G Pl0 Unit (List ℕ) T m)

theorem **goodGeneric_linear** {K : Type} [Field K] [CharZero K] {G : Type} [Group G] [Fintype G] (ψ : G →* K×) (P : Subgroup G) : GoodGeneric {fun g => (ψ g : K)} P (ψ.comp P.subtype)

private lemma **chiIdem_mul_single** {K : Type} [Field K] {G : Type} [Group G] [Fintype G] (χ : G → K) (hχ : ∀ a b, χ (a * b) = χ a * χ b) (h : G) (c : K) : chiIdem χ * MonoidAlgebra.single h c = (c * χ h) • chiIdem χ

private lemma **chiIdem_mul_lvlIdem** {K : Type} [Field K] {G : Type} [Group G] [Fintype G] (χ : G → K) (hχ : ∀ a b, χ (a * b) = χ a * χ b) (P : Subgroup G) [Fintype P] (ψ : P →* K×) : chiIdem χ * lvlIdem P ψ = (∑ h : P, (Nat.card P : K)⁻¹ * ((ψ h⁻¹ : K×) : K) * χ h) • chiIdem χ

private lemma **sum_char_subgroup_eq_zero_of_eq_neg_one** {K : Type} [Field K] [CharZero K] {G : Type} [Group G] [Fintype G] (χ : G → K) (hχ : ∀ a b, χ (a * b) = χ a * χ b) (P : Subgroup G) [Fintype P] (a : P) (ha : χ a = -1) : ∑ h : P, χ h = 0

theorem **not_goodGeneric_sign** : ¬ GoodGeneric (K

theorem **trISepWith_self** : TrISepWith A3 A3.trI ↔ A3.fam3.TrISep

theorem **trISepWith_id** : TrISepWith A3 fun _ e => e

theorem **labW_single** (j : Fin D.R.m) {I : Type} (pl : I → Pl D.host.E) (d : I → List ℤ) (c : Pl D.host.E → List ℤ → ℂ) (i : I) (h : IsLab (Wc D j) (pl i) (d i)) : labW j pl (fun i => Finsupp.single (d i) 1) c i = c (pl i) (d i)

theorem **liftDict_eigen** {R : RealDat3 D A3 Ks} {j : Fin D.R.m} {t : T} {pl cV cW} (H : LiftDict R j t pl cV cW) (h) (hh : IsOpsV R.toRealDat j t h) (x) (s : (A3.liftI j t).OpIx → D.what₂.LL) (hx : ∀ i, (A3.liftI j t).opAll i x = s i • x) (i) : h (pl i) (cV i) (R.jV j t x) = D.what₂.embLL (s i) • R.jV j t x

theorem **rowDict_eigen** {R : RealDat3 D A3 Ks} {t : T} {pl cX c0} (H : RowDict R t pl cX c0) (h) (hh : IsOpsX R.toRealDat t h) (x) (s : (A3.rowI t).OpIx → D.what₂.LL) (hx : ∀ i, (A3.rowI t).opAll i x = s i • x) (i) : h (pl i) (cX i) (R.jX t x) = D.what₂.embLL (s i) • R.jX t x

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NotIsobaricRigid

theorem **sign_ne** (ε : ℤ×) : (0 : ℂ) ∉ ({((ε : ℤ) : ℂ)} : Multiset ℂ)

theorem **sign_inv** (ε : ℤ×) : ((ε : ℤ) : ℂ)⁻¹ = ((ε : ℤ) : ℂ)

theorem **sg_spec** (N : NData R H what₂) (ε : ℤ×) (w : PlE H) : (sg N ε w).unr ∧ (sg N ε w).frobEig = {((ε : ℤ) : ℂ)} ∧ (sg N ε w).lpDim = 1

theorem **sg_mul** (N : NData R H what₂) (ε δ : ℤ×) (w : PlE H) : sg N (ε * δ) w = (sg N ε w).lpChar (sg N δ w)

theorem **labTame** : LabTame N (wdAut N A₀ τ γ)

theorem **rankOneUnitary** : RankOneUnitary (tie N A₀ τ γ)

theorem **char_sum** (a b c : WD h) : (a.lpSum b).lpChar c = (a.lpChar c).lpSum (b.lpChar c)

theorem **char_tw_left** (a c : WD h) (s : ℝ) : (a.lpTw s).lpChar c = (a.lpChar c).lpTw s

theorem **char_tw_right** (a c : WD h) (s : ℝ) : a.lpChar (c.lpTw s) = (a.lpChar c).lpTw s

theorem **char_assoc** (a b c : WD h) : (a.lpChar b).lpChar c = a.lpChar (b.lpChar c)

theorem **char_one** (a : WD h) : a.lpChar (WD.lpUnr h {1}) = a

theorem **tie_lpUnr** (T : WDTie N A) (w : PlE H) (m : Multiset ℂ) (hm : (0 : ℂ) ∉ m) : T.e w (A.lpUnr w m) = WD.lpUnr _ m

theorem **charLaws_of_tie** (T : WDTie N A) : CharLaws N A

theorem **exists_not_mono0** (T : WDTie N A) (w : PlE H) : ∃ a : A.LP w, ¬ A.mono0 a ∧ A.tempered a ∧ A.lpDim a = 2

theorem **not_isobaricRigid** {a : ℕ} (ha : 1 ≤ a) (π π' : A.CG a) {s : ℝ} (hs : s ≠ 0) (h : ∀ w, A.locG π' w = A.lpTw (A.locG π w) s) : ¬ IsobaricRigid N A

theorem **infinite_places** (K : Type) [Field K] [NumberField K] : Infinite (HeightOneSpectrum (𝒪 K))

theorem **ev_inj** (w : PlE H) : Function.Injective (ev w)

theorem **twSum_spec** (w : PlE H) (L : List ((Σ a : ℕ, (wdAut N A₀ τ γ).CG a) × ℝ)) : WD.unr (twSum N (wdAut N A₀ τ γ) L w) ∧ WD.frobEig (twSum N (wdAut N A₀ τ γ) L w) = ((L.map (ev w) : List ℂ) : Multiset ℂ)

theorem **isobaricRigid** : IsobaricRigid N (wdAut N A₀ τ γ)

theorem **label_full** (N : NData R H what₂) [IsEmpty N.Loc] (A₀ : NAut N) (τ : ∀ k, Fb (H.g k) (H.g_prime k) →* ℂ) (γ : Fin 2 → PlE H → G H.E N.Om) (k : Fin 2) (w : PlE H) (hL : haveI : Fact (H.g k).Prime

theorem **exists_tie** (H : Host R) (what₂ : Rows.Data R) (h : Examples₂.NHyp H what₂) (τ : ∀ k, Fb (H.g k) (H.g_prime k) →* ℂ) (γ : Fin 2 → PlE H → G H.E (Examples₂.ndToy H what₂ h).Om) : ∃ (A : NAut (Examples₂.ndToy H what₂ h)) (T : WDTie _ A), CharLaws _ A ∧ LabTame _ A ∧ RankOneUnitary T ∧ IsobaricRigid _ A

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.NotSlightlyRegRankOne

theorem **not_slightlyReg_rank_one** (π : CuspSub E 1) : ¬ SlightlyReg π

theorem **triv_oddOr** : OddOrSlightlyReg (CuspSub.triv E (le_refl 1)) ∧ ¬ SlightlyReg (CuspSub.triv E (le_refl 1))

theorem **not_slightlyReg_of_not_hasInfChar** (π : CuspSub E a) (ha : 2 ≤ a) (h : ∀ τ m, ¬ (CG.mk π).HasInfChar τ m) : ¬ SlightlyReg π

theorem **slightlyReg_twist_of** (π : CuspSub E a) (χ : HChar E) (h : SlightlyReg π) : SlightlyReg (π.twist χ)

theorem **slightlyReg_twist** (π : CuspSub E a) (χ : HChar E) : SlightlyReg (π.twist χ) ↔ SlightlyReg π

theorem **oddOrSlightlyReg_twist** (π : CuspSub E a) (χ : HChar E) : OddOrSlightlyReg (π.twist χ) ↔ OddOrSlightlyReg π

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.OddOrSlightlyRegIffOfEven

theorem **OddOrSlightlyReg.of_odd** (π : CuspSub E a) (h : Odd a) : OddOrSlightlyReg π

theorem **OddOrSlightlyReg.of_slightlyReg** {π : CuspSub E a} (h : SlightlyReg π) : OddOrSlightlyReg π

theorem **oddOrSlightlyReg_iff_of_even** (π : CuspSub E a) (h : Even a) : OddOrSlightlyReg π ↔ SlightlyReg π

theorem **SlightlyRegAll.slightlyReg** {π : CuspSub E a} (ha : 2 ≤ a) (h : SlightlyRegAll π) : SlightlyReg π

theorem **slightlyReg_of_weight** (π : CuspSub E a) (τ : E →+* ℂ) (w : Fin a → ℤ) (hw : Antitone w) (h : (CG.mk π).infChar τ = Finset.univ.val.map fun k : Fin a => (w k : ℂ) + (((a : ℂ) - 1) / 2 - (k : ℕ))) (k l : Fin a) (hk : Even (k : ℕ)) (hl : (l : ℕ) = k + 1) (hg : w l < w k) : SlightlyReg π

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.OfNatTrans

theorem **d_pullC** (n : ℕ) (f : Cn F n) : d F' n (m.pullC n f) = m.pullC (n + 1) (d F n f)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.OmegaPullOmegaTraceOfSecTraceOne

theorem **sec_isScalarTower** (hf : f ≫ s = s') (U : X.Opens) : letI

theorem **sec_finite** [IsFinite f] {U : X.Opens} (hU : IsAffineOpen U) : letI

theorem **sec_etale** [IsFinite f] [Etale f] {U : X.Opens} (hU : IsAffineOpen U) : letI

theorem **secTrace_mul** {U : X.Opens} (hU : IsAffineOpen U) (b : Sec s U) (b' : Sec s' (f ⁻¹ᵁ U)) : secTrace s s' hU (appSec s s' f U b ∗ b') = b ∗ secTrace s s' hU b'

theorem **secTrace_res** {U V : X.Opens} (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) (b' : Sec s' (f ⁻¹ᵁ U)) : Sec.res s h (secTrace s s' hU b') = secTrace s s' hV (Sec.res s' (f.preimage_mono h) b')

theorem **formTraceU_smul_formPull** (hf : f ≫ s = s') {U : X.Opens} (hU : IsAffineOpen U) {j : ℕ} (b' : Sec s' (f ⁻¹ᵁ U)) (x : OmU s U j) : formTraceU hf hU j (b' • formPull hf U j x) = secTrace s s' hU b' • x

theorem **formTraceU_smul** (hf : f ≫ s = s') {U : X.Opens} (hU : IsAffineOpen U) {j : ℕ} (b : Sec s U) (y : OmU s' (f ⁻¹ᵁ U) j) : formTraceU hf hU j (appSec s s' f U b • y) = b • formTraceU hf hU j y

theorem **formOm_induction** (hf : f ≫ s = s') {U : X.Opens} (hU : IsAffineOpen U) {j : ℕ} {P : OmU s' (f ⁻¹ᵁ U) j → Prop} (h0 : P 0) (ha : ∀ y z, P y → P z → P (y + z)) (hm : ∀ (b' : Sec s' (f ⁻¹ᵁ U)) (x : OmU s U j), P (b' • formPull hf U j x)) (y : OmU s' (f ⁻¹ᵁ U) j) : P y

theorem **formTraceU_res** (hf : f ≫ s = s') {U V : X.Opens} (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) {j : ℕ} (y : OmU s' (f ⁻¹ᵁ U) j) : formRes s h j (formTraceU hf hU j y) = formTraceU hf hV j (formRes s' (f.preimage_mono h) j y)

theorem **omegaTraceU_toOm** (hf : f ≫ s = s') (j : ℕ) {U : X.Opens} (hU : IsAffineOpen U) (y : OmU s' (f ⁻¹ᵁ U) j) : omegaTraceU hf j hU (toOm s' j (f ⁻¹ᵁ U) y) = toOm s j U (formTraceU hf hU j y)

theorem **omegaTraceU_res** (hf : f ≫ s = s') (j : ℕ) {U V : X.Opens} (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) (y : Γ((Scheme.Modules.pushforward f).obj (Omega s' j), U)) : (Omega s j).presheaf.map (homOfLE h).op (omegaTraceU hf j hU y) = omegaTraceU hf j hV (((Scheme.Modules.pushforward f).obj (Omega s' j)).presheaf.map (homOfLE h).op y)

theorem **omegaTrace_app_toOm** (hf : f ≫ s = s') (j : ℕ) {U : X.Opens} (hU : IsAffineOpen U) (y : OmU s' (f ⁻¹ᵁ U) j) : (omegaTrace hf j).app U (toOm s' j (f ⁻¹ᵁ U) y) = toOm s j U (formTraceU hf hU j y)

theorem **omegaTrace_toOm_smul_formPull** (hf : f ≫ s = s') (j : ℕ) {U : X.Opens} (hU : IsAffineOpen U) (b' : Sec s' (f ⁻¹ᵁ U)) (x : OmU s U j) : (omegaTrace hf j).app U (toOm s' j (f ⁻¹ᵁ U) (b' • formPull hf U j x)) = toOm s j U (secTrace s s' hU b' • x)

theorem **omegaTrace_ext** (hf : f ≫ s = s') (j : ℕ) (T : (Scheme.Modules.pushforward f).obj (Omega s' j) ⟶ Omega s j) (hT : ∀ (U : X.Opens) (hU : IsAffineOpen U) (b' : Sec s' (f ⁻¹ᵁ U)) (x : OmU s U j), T.app U (toOm s' j (f ⁻¹ᵁ U) (b' • formPull hf U j x)) = toOm s j U (secTrace s s' hU b' • x)) : T = omegaTrace hf j

theorem **hodgeTrace_hodgePull** (hf : f ≫ s = s') (j q : ℕ) (x : hodgeH s j q) : letI

theorem **secTrace_appSec_of_bijective** {U : X.Opens} (hU : IsAffineOpen U) (h : Function.Bijective (appSec s s' f U)) (b : Sec s U) : secTrace s s' hU (appSec s s' f U b) = b

theorem **secTrace_id** {U : X.Opens} (hU : IsAffineOpen U) (b : Sec s U) : secTrace (f

theorem **omegaTrace_id** (j : ℕ) : omegaTrace (f

theorem **hodgeTrace_id** (j q : ℕ) (x : hodgeH s j q) : hodgeTrace (f

theorem **nsmul_app** {F G : X.Modules} (φ : F ⟶ G) (d : ℕ) (U : X.Opens) (y : Γ(F, U)) : (d • φ).app U y = d • φ.app U y

theorem **omegaPull_omegaTrace_of_secTrace_one** (hf : f ≫ s = s') (j d : ℕ) (hd : ∀ (U : X.Opens) (hU : IsAffineOpen U), secTrace (f

theorem **hodgeTrace_hodgePull_of_secTrace_one** (hf : f ≫ s = s') (j q d : ℕ) (hd : ∀ (U : X.Opens) (hU : IsAffineOpen U), secTrace (f

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.OverOnePrimeOfCardLeOne

theorem **quadFields_ok** (K : Type) [Field K] [NumberField K] [IsCMField K] (hK : Module.finrank ℚ K = 2) : (quadFields K).Ok

theorem **isCMField_gauss** : IsCMField (CyclotomicField 4 ℚ)

theorem **exists_fields_ok** : ∃ X : Fields, X.Ok ∧ X.f = 1

theorem **Wts.const_ok** {X : Fields} (Word : Type) {R : Ranks} (P : X.Tau → HCParam R) (tops : Finset X.Tau) : (Wts.const Word P tops).Ok

theorem **QDat.empty_ok** (X : Fields) (Pr : Primes X) : (QDat.empty X).Ok ∧ (QDat.empty X).Off Pr

theorem **QDat.overOnePrime_iff** {X : Fields} (Z : QDat X) (p : ℕ) : QDat.OverOnePrime Z p ↔ ∀ P ∈ Z.Q, ∀ P' ∈ Z.Q, ((p : ℕ) : 𝒪 X.E) ∈ P.asIdeal → ((p : ℕ) : 𝒪 X.E) ∈ P'.asIdeal → ∀ x : 𝒪 X.E, (x : X.E) ∈ X.Kq → x ∈ P.asIdeal → x ∈ P'.asIdeal

theorem **QDat.OverOnePrime.conj_notMem** {X : Fields} {Z : QDat X} {p : ℕ} (h : QDat.OverOnePrime Z p) {P P' : HeightOneSpectrum (𝒪 X.E)} (hP : P ∈ Z.Q) (hP' : P' ∈ Z.Q) (h1 : ((p : ℕ) : 𝒪 X.E) ∈ P.asIdeal) (h2 : ((p : ℕ) : 𝒪 X.E) ∈ P'.asIdeal) {x : 𝒪 X.E} (hx : (x : X.E) ∈ X.Kq) (hc : IsCoprime x (X.conjO x)) (hxP : x ∈ P.asIdeal) : X.conjO x ∉ P'.asIdeal

theorem **QDat.overOnePrime_of_card_le_one** {X : Fields} (Z : QDat X) (p : ℕ) (h : Z.Q.card ≤ 1) : QDat.OverOnePrime Z p

theorem **QDat.overOnePrime_of_hasBase** {X : Fields} {Z : QDat X} {p : ℕ} (h : Z.HasBase p) (hf : X.f = 1) : QDat.OverOnePrime Z p

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PolarizedGaloisRepNarrow2

theorem **mem_ineEig** {a : WDRep h V} {χ : Ine (F

theorem **Mono0.mono0Ine** {a : WDRep h V} (ha : a.Mono0) (χ : Ine (F

theorem **Iso.mono0Ine** {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) {χ : Ine (F

theorem **mono0Ine_tw** {a : WDRep h V} {χ : Ine (F

@[simp] theorem **mono0Ine_mk** {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (a : WDRep h V) (χ : Ine (F

theorem **mono0Ine_of_mono0** {x : WD h} (hx : mono0 x) (χ : Ine (F

@[simp] theorem **mono0Ine_lpTw** (x : WD h) (s : ℝ) (χ : Ine (F

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PolarizedGaloisRepNarrowT

theorem **everywhereGood_of_toral** {π : CuspSub E a} (H : ∀ (w : HeightOneSpectrum (𝒪 E)) (σ : SmIrr (GL (Fin a) (w.adicCompletion E))), π.IsLocComp w σ → UpperTheory.Zero.LLC.Toral (IrrGL.of σ.ρ σ.smooth σ.irred)) : EverywhereGood π

theorem **everywhereGood_rank_one** (π : CuspSub E 1) : EverywhereGood π

theorem **EverywhereGood.rc_algNorm_eq** {π : CuspSub E a} (hπ : EverywhereGood π) {w : HeightOneSpectrum (𝒪 E)} {A : ValuationSubring Ω} {h : IsPlaceOver A w} {rc rc' : IrrGL (w.adicCompletion E) → WD h} (hrc : UpperTheory.Zero.LLC.IsRecT h rc) (hrc' : UpperTheory.Zero.LLC.IsRecT h rc') (e : ∀ x, UpperTheory.Zero.LLC.Good x → rc x = rc' x) {σ : SmIrr (GL (Fin a) (w.adicCompletion E))} (hσ : π.IsLocComp w σ) : rc (algNorm σ) = rc' (algNorm σ)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PolarizedPairLineIff

theorem **op_line** (ha : a ≤ 1) (χ : HChar E) (o : Op E a) (φ : (line ha χ).V) : (line ha χ).op o φ = sc χ o • φ

theorem **gen_one** (ha : a ≤ 1) (χ : HChar E) : (gen ha χ).1.1 1 = 1

theorem **eq_smul_gen** (ha : a ≤ 1) (χ : HChar E) (φ : (line ha χ).V) : φ = φ.1.1 1 • gen ha χ

theorem **paired_line_iff** (ha : a ≤ 1) (χ χ' : HChar E) (s : Op E a → Op E a) : Paired (line ha χ) (line ha χ') s ↔ (∀ k, sc χ (.fin k) * sc χ' (s (.fin k)) = 1) ∧ (∀ k, sc χ (.cpt k) * sc χ' (s (.cpt k)) = 1) ∧ ∀ X, sc χ (.lie X) + sc χ' (s (.lie X)) = 0

theorem **toUnits_barK** (k : Kinf E a) : Unitary.toUnits (barK k) = invT a (Cinf E) (Unitary.toUnits k)

theorem **barK_apply** (k : Kinf E a) (i j : Fin a) : (barK k).1 i j = star (k.1 i j)

theorem **det_invT** {R : Type} [CommRing R] (g : GL (Fin a) R) : GeneralLinearGroup.det (invT a R g) = (GeneralLinearGroup.det g)⁻¹

theorem **sc_barK** (χ : HChar E) (k : Kinf E a) : sc χ (.cpt k) * sc χ (.cpt (barK k)) = 1

theorem **paired_invTOp** (ha : a ≤ 1) (χ : HChar E) : Paired (line ha χ) (line ha χ) invTOp

theorem **det_map'** {R S : Type} [CommRing R] [CommRing S] (c : R →+* S) (k : GL (Fin a) R) : GeneralLinearGroup.det (GeneralLinearGroup.map c k) = Units.map c.toMonoidHom (GeneralLinearGroup.det k)

theorem **sc_fin** (μ : HChar L) (k : GL (Fin a) (Af L)) : sc μ (thetaOp (.fin k)) * sc μ (conjOp (.fin k)) = 1

theorem **sc_cpt** (μ : HChar L) (k : Kinf L a) : sc μ (thetaOp (.cpt k)) * sc μ (conjOp (.cpt k)) = 1

theorem **sc_lie** (μ : HChar L) (X : Lie L a) : sc μ (thetaOp (.lie X)) + sc μ (conjOp (.lie X)) = 0

theorem **dualIsConjBy_line_iff** (ha : a ≤ 1) (μ ψ : HChar L) : DualIsConjBy (line ha μ) ψ ↔ IsConjDualBy (line ha μ) ψ

theorem **dualIsConj_line_iff** (ha : a ≤ 1) (μ : HChar L) : DualIsConj (line ha μ) ↔ (line ha μ).IsCsd

theorem **polarizedPair_line_iff** (ha : a ≤ 1) (μ : HChar L) (χ : HChar (maximalRealSubfield L)) : PolarizedPair (line ha μ) χ ↔ IsPolarized (line ha μ) χ

theorem **dualIsConj_abs_iff** (s : ℂ) : DualIsConj (line le_rfl (CuspRep.Model.absC L s)) ↔ s = 0

theorem **polarizedPair_triv** : PolarizedPair (CuspSub.triv L (le_refl 1)) (quadChar L ^ 1)

theorem **not_polarizedPair_one** (π : CuspSub L 1) : ¬ PolarizedPair π 1

theorem **polarizedPair_absPow** (s : ℝ) : ∃ χ, PolarizedPair (line le_rfl (GlobalTheory.HeckeGal.Model.absPow L s)) χ

theorem **dualIsConjBy_resReal** (μ : HChar L) : DualIsConjBy (line le_rfl μ) (compNorm L (GlobalTheory.HeightCounting.Blocks.resReal μ))

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PolyApproxZeroPadic

theorem **linApprox_of_mem** {Γ : Type} [Group Γ] {a : ℕ} {r : Γ →* GL (Fin a) k} {C : Set (Γ →* GL (Fin a) k)} (h : r ∈ C) : LinApprox r C

theorem **PolyApprox.mono** [IsUltrametricDist k] {J : Type} {θ : J → k} {S T : Set (J → k)} (h : PolyApprox θ S) (hST : S ⊆ T) : PolyApprox θ T

theorem **LinApprox.mono** {Γ : Type} [Group Γ] {a : ℕ} {r : Γ →* GL (Fin a) k} {C D : Set (Γ →* GL (Fin a) k)} (h : LinApprox r C) (hCD : C ⊆ D) : LinApprox r D

theorem **aeval_sub_le** {J : Type} {x y : J → k} {δ : ℝ} (hδ : 0 ≤ δ) (hx : ∀ j, ‖x j‖ ≤ 1) (hy : ∀ j, ‖y j‖ ≤ 1) (hxy : ∀ j, ‖x j − y j‖ ≤ δ) (P : MvPolynomial J ℤ) : ‖MvPolynomial.aeval x P‖ ≤ 1 ∧ ‖MvPolynomial.aeval y P‖ ≤ 1 ∧ ‖MvPolynomial.aeval x P − MvPolynomial.aeval y P‖ ≤ δ

theorem **polyApprox_of_uniform** {J : Type} {θ : J → k} {S : Set (J → k)} (hθ : ∀ j, ‖θ j‖ ≤ 1) (h : ∀ ε : ℝ, 0 < ε → ∃ η ∈ S, (∀ j, ‖η j‖ ≤ 1) ∧ ∀ j, ‖θ j − η j‖ ≤ ε) : PolyApprox θ S

theorem **polyApprox_zero_padic** (p : ℕ) [Fact p.Prime] : PolyApprox (fun _ : Unit => (0 : ℚ_[p])) {θ | ∃ n : ℕ, θ = fun _ => (p : ℚ_[p]) ^ (n + 1)} ∧ (fun _ : Unit => (0 : ℚ_[p])) ∉ {θ : Unit → ℚ_[p] | ∃ n : ℕ, θ = fun _ => (p : ℚ_[p]) ^ (n + 1)}

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PreservesFiniteLimitsTensorLeftSheafify

theorem **preservesFiniteLimits_tensorLeft_presheaf** (P : Cᵒᵖ ⥤ ModuleCat.{1} S) (hP : ∀ U, Module.Flat S (P.obj U)) : PreservesFiniteLimits (tensorLeft P)

theorem **preservesFiniteLimits_tensorLeft_sheafify** (P : Cᵒᵖ ⥤ ModuleCat.{1} S) (hP : ∀ U, Module.Flat S (P.obj U)) : PreservesFiniteLimits (tensorLeft ((presheafToSheaf J _).obj P))

theorem **tensorLeft_sheaf_additive** (F : Sheaf J (ModuleCat.{1} S)) : (tensorLeft F).Additive

theorem **tensorLeft_sheaf_linear** (F : Sheaf J (ModuleCat.{1} S)) : letI

theorem **FlatCoef.of_flat** (h : ∀ U : Scheme.{0}, Module.Flat R C(U, R)) : FlatCoef R

theorem **cupExt_add_left** (X : Scheme.{0}) {i j n : ℕ} (h : i + j = n) (a a' : HS R (coef R) i X) (b : HS R (coef R) j X) : cupExt R X h (a + a') b = cupExt R X h a b + cupExt R X h a' b

theorem **cupExt_add_right** (X : Scheme.{0}) {i j n : ℕ} (h : i + j = n) (a : HS R (coef R) i X) (b b' : HS R (coef R) j X) : cupExt R X h a (b + b') = cupExt R X h a b + cupExt R X h a b'

theorem **cupExt_smul_left** (X : Scheme.{0}) {i j n : ℕ} (h : i + j = n) (r : ULift.{1} R) (a : HS R (coef R) i X) (b : HS R (coef R) j X) : cupExt R X h (r • a) b = r • cupExt R X h a b

theorem **cupExt_smul_right** (X : Scheme.{0}) {i j n : ℕ} (h : i + j = n) (r : ULift.{1} R) (a : HS R (coef R) i X) (b : HS R (coef R) j X) : cupExt R X h a (r • b) = r • cupExt R X h a b

theorem **cupN_apply** (X : Scheme.{0}) (i j n : ℕ) (h : i + j = n) (a : H R i X) (b : H R j X) : cupN R X i j n h a b = cupExt R X h a b

theorem **cup_eq_cupN** (X : Scheme.{0}) (i j : ℕ) : cup R X i j = cupN R X i j (i + j) rfl

theorem **freeDiagPre_naturality** {X Y : Scheme.{0}} (f : X ⟶ Y) : ((Functor.whiskeringRight _ _ _).obj (ModuleCat.free (ULift.{1} R))).map (uliftYoneda.{1}.map f) ≫ freeDiagPre R Y = freeDiagPre R X ≫ (((Functor.whiskeringRight _ _ _).obj (ModuleCat.free (ULift.{1} R))).map (uliftYoneda.{1}.map f) ⊗ₘ ((Functor.whiskeringRight _ _ _).obj (ModuleCat.free (ULift.{1} R))).map (uliftYoneda.{1}.map f))

theorem **freeDiag_naturality** {X Y : Scheme.{0}} (f : X ⟶ Y) : (freeOn R).map f ≫ freeDiag R Y = freeDiag R X ≫ ((freeOn R).map f ⊗ₘ (freeOn R).map f)

theorem **mk₀_comp_cupExt** {X Y : Scheme.{0}} (f : X ⟶ Y) {i j n : ℕ} (h : i + j = n) (a : HS R (coef R) i Y) (b : HS R (coef R) j Y) : (Abelian.Ext.mk₀ ((freeOn R).map f)).comp (cupExt R Y h a b) (zero_add n) = cupExt R X h ((Abelian.Ext.mk₀ ((freeOn R).map f)).comp a (zero_add i)) ((Abelian.Ext.mk₀ ((freeOn R).map f)).comp b (zero_add j))

theorem **pull_cupN** {X Y : Scheme.{0}} (f : X ⟶ Y) (i j n : ℕ) (h : i + j = n) (a : H R i Y) (b : H R j Y) : pull R n f (cupN R Y i j n h a b) = cupN R X i j n h (pull R i f a) (pull R j f b)

theorem **pull_cup** {X Y : Scheme.{0}} (f : X ⟶ Y) (i j : ℕ) (a : H R i Y) (b : H R j Y) : pull R (i + j) f (cup R Y i j a b) = cup R X i j (pull R i f a) (pull R j f b)

theorem **cupExt_mk₀** (X : Scheme.{0}) (g g' : (freeOn R).obj X ⟶ coef R) : cupExt R X (rfl : 0 + 0 = 0) (Abelian.Ext.mk₀ g) (Abelian.Ext.mk₀ g') = Abelian.Ext.mk₀ (freeDiag R X ≫ (g ⊗ₘ g') ≫ coefMul R)

theorem **cupN_ofPre** (X : Scheme.{0}) (γ γ' : freePre R X ⟶ coefPre R) : cupN R X 0 0 0 rfl (ofPre R γ) (ofPre R γ') = ofPre R (freeDiagPre R X ≫ (γ ⊗ₘ γ') ≫ coefMulPre R)

theorem **e_ofPre** (X : Scheme.{0}) (γ : freePre R X ⟶ coefPre R) : H0.e R X (ofPre R γ) = (γ.app (op X) (Finsupp.single (ULift.up (𝟙 X)) 1)).down

theorem **homEquiv_map_ofPre** {X : Scheme.{0}} (γ : freePre R X ⟶ coefPre R) : (sheafificationAdjunction Scheme.proetaleTopology.{0} (ModuleCat.{1} (ULift.{1} R))).homEquiv _ _ ((presheafToSheaf _ _).map γ) = γ ≫ toSheafify _ (coefPre R)

theorem **exists_ofPre** {X : Scheme.{0}} (x : H R 0 X) : ∃ γ : freePre R X ⟶ coefPre R, ofPre R γ = x

theorem **e_cupN** (X : Scheme.{0}) (x y : H R 0 X) : H0.e R X (cupN R X 0 0 0 rfl x y) = H0.e R X x ∗ H0.e R X y

theorem **e_cup** (X : Scheme.{0}) (x y : H R 0 X) : H0.e R X (cup R X 0 0 x y) = H0.e R X x ∗ H0.e R X y

theorem **one_eq_ofPre** (X : Scheme.{0}) : one R X = ofPre R (freeAugPre R X ≫ coefOnePre R)

theorem **e_one** (X : Scheme.{0}) : H0.e R X (one R X) = 1

theorem **gal_cup** {k : Type} [Field k] (Ω : Type) [Field Ω] [Algebra k Ω] {X : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) (i j : ℕ) (σ : Ω ≃ₐ[k] Ω) (a : H R i (geom Ω p)) (b : H R j (geom Ω p)) : gal R Ω p (i + j) σ (cup R (geom Ω p) i j a b) = cup R (geom Ω p) i j (gal R Ω p i σ a) (gal R Ω p j σ b)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PullTotMemBTot

theorem **dV_apply** {a q : ℕ} (c : Term F a q) (s : ComposableArrows C a) : dV F a q c s = dOm R (F.obj (s.obj (Fin.last a))) q (c s)

@[simp] theorem **totH_zero** {q : ℕ} (h : 0 + q = n + 1) : totH F n c ⟨(0, q), h⟩ = 0

@[simp] theorem **totH_succ** {a q : ℕ} (h : a + 1 + q = n + 1) : totH F n c ⟨(a + 1, q), h⟩ = dH F a q (c ⟨(a, q), by omega⟩)

@[simp] theorem **totV_zero** {a : ℕ} (h : a + 0 = n + 1) : totV F n c ⟨(a, 0), h⟩ = 0

@[simp] theorem **totV_succ** {a q : ℕ} (h : a + (q + 1) = n + 1) : totV F n c ⟨(a, q + 1), h⟩ = (-1 : ℤ) ^ a • dV F a q (c ⟨(a, q), by omega⟩)

theorem **dTot_apply** (p : Bidegree (n + 1)) : dTot F n c p = totH F n c p + totV F n c p

theorem **Hyp.mk_surjective** (n : ℕ) : Function.Surjective (Hyp.mk F n)

theorem **Hyp.mk_eq_zero** {n : ℕ} {c : ZTot F n} : Hyp.mk F n c = 0 ↔ c.1 ∈ BTot F n

theorem **TotF_anti** {p p' : ℕ} (h : p ≤ p') (n : ℕ) : TotF F p' n ≤ TotF F p n

theorem **TotF_zero** (n : ℕ) : TotF F 0 n = ⊤

theorem **TotF_eq_bot** {p n : ℕ} (h : n < p) : TotF F p n = ⊥

theorem **BF_le_BTot** (p : ℕ) : ∀ n : ℕ, BF F p n ≤ BTot F n | 0 => le_rfl | _ + 1 => LinearMap.map_le_range theorem BF_anti {p p' : ℕ} (h : p ≤ p') : ∀ n : ℕ, BF F p' n ≤ BF F p n | 0 => le_rfl | n + 1 => Submodule.map_mono (TotF_anti F h n) def HypF (p n : ℕ) : Type (max u v)

theorem **HypF.mk_surjective** (p n : ℕ) : Function.Surjective (HypF.mk F p n)

theorem **HypF.mk_eq_zero** {p n : ℕ} {c : ZF F p n} : HypF.mk F p n c = 0 ↔ c.1 ∈ BF F p n

theorem **HypF.toHyp_incl** {p p' : ℕ} (h : p ≤ p') (n : ℕ) (c : HypF F p' n) : HypF.toHyp F p n (HypF.incl F h n c) = HypF.toHyp F p' n c

theorem **Hyp.mem_fil** {p n : ℕ} {x : Hyp F n} : x ∈ Hyp.fil F p n ↔ ∃ c : ZTot F n, c.1 ∈ TotF F p n ∧ Hyp.mk F n c = x

theorem **Hyp.fil_anti** {p p' : ℕ} (h : p ≤ p') (n : ℕ) : Hyp.fil F p' n ≤ Hyp.fil F p n

theorem **Hyp.fil_zero** (n : ℕ) : Hyp.fil F 0 n = ⊤

theorem **Hyp.fil_eq_bot** {p n : ℕ} (h : n < p) : Hyp.fil F p n = ⊥

theorem **dH_lead** {a p n : ℕ} (h : a + p = n) (c : ZF F p n) : dH F a p (c.1 ⟨(a, p), h⟩) = 0

theorem **dH_pullTerm** {a q : ℕ} (c : Term F a q) : dH F' a q (pullTerm m a q c) = pullTerm m (a + 1) q (dH F a q c)

theorem **dV_pullTerm** {a q : ℕ} (c : Term F a q) : dV F' a q (pullTerm m a q c) = pullTerm m a (q + 1) (dV F a q c)

theorem **pullTot_apply** {n : ℕ} (c : Tot F n) (b : Bidegree n) : pullTot m n c b = pullTerm m _ _ (c b)

theorem **dTot_pullTot** {n : ℕ} (c : Tot F n) : dTot F' n (pullTot m n c) = pullTot m (n + 1) (dTot F n c)

theorem **pullTot_mem_ZTot** {n : ℕ} {c : Tot F n} (hc : c ∈ ZTot F n) : pullTot m n c ∈ ZTot F' n

theorem **pullTot_mem_TotF** {p n : ℕ} {c : Tot F n} (hc : c ∈ TotF F p n) : pullTot m n c ∈ TotF F' p n

theorem **pullTot_mem_BTot** : ∀ {n : ℕ} {c : Tot F n}, c ∈ BTot F n → pullTot m n c ∈ BTot F' n | 0, c, hc => by rw [show c = 0 from hc, map_zero]; exact zero_mem _ | _ + 1, _, ⟨e, he⟩ => ⟨_, (dTot_pullTot m e).trans (congrArg _ he)⟩ theorem pullTot_mem_BF {p : ℕ} : ∀ {n : ℕ} {c : Tot F n}, c ∈ BF F p n → pullTot m n c ∈ BF F' p n | 0, c, hc => by rw [show c = 0 from hc, map_zero]; exact zero_mem _ | _ + 1, _, ⟨e, h, he⟩ => ⟨_, pullTot_mem_TotF m h, (dTot_pullTot m e).trans (congrArg _ he)⟩ def pullZ (n : ℕ) : ZTot F n →ₛₗ[σ] ZTot F' n

theorem **HypF.toHyp_pull** {p n : ℕ} (c : HypF F p n) : HypF.toHyp F' p n (HypF.pull m p n c) = Hyp.pull m n (HypF.toHyp F p n c)

theorem **Hyp.pull_mem_fil** {p n : ℕ} {x : Hyp F n} (hx : x ∈ Hyp.fil F p n) : Hyp.pull m n x ∈ Hyp.fil F' p n

theorem **pullTot_id** {n : ℕ} (c : Tot F n) : pullTot (CoefMap.id F) n c = c

theorem **pullTot_comp** (m' : CoefMap τ F' F'') {n : ℕ} (c : Tot F n) : pullTot (m.comp m') n c = pullTot m' n (pullTot m n c)

theorem **Hyp.pull_id** {n : ℕ} (x : Hyp F n) : Hyp.pull (CoefMap.id F) n x = x

theorem **Hyp.pull_comp** (m' : CoefMap τ F' F'') {n : ℕ} (x : Hyp F n) : Hyp.pull (m.comp m') n x = Hyp.pull m' n (Hyp.pull m n x)

theorem **HypF.pull_id** {p n : ℕ} (x : HypF F p n) : HypF.pull (CoefMap.id F) p n x = x

theorem **HypF.pull_comp** (m' : CoefMap τ F' F'') {p n : ℕ} (x : HypF F p n) : HypF.pull (m.comp m') p n x = HypF.pull m' p n (HypF.pull m p n x)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.PushWithIsTranspose

theorem **Ht.cast_add** {q : ℕ} {E : Spc} {n i n' i' : ℤ} (hn : n = n') (hi : i = i') (x y : Ht q E n i) : (x + y).cast hn hi = x.cast hn hi + y.cast hn hi

theorem **Ht.cast_zero** {q : ℕ} {E : Spc} {n i n' i' : ℤ} (hn : n = n') (hi : i = i') : (0 : Ht q E n i).cast hn hi = 0

theorem **Ht.cast_cast** {q : ℕ} {E : Spc} {n i n' i' n'' i'' : ℤ} (hn : n = n') (hi : i = i') (hn' : n' = n'') (hi' : i' = i'') (x : Ht q E n i) : (x.cast hn hi).cast hn' hi' = x.cast (hn.trans hn') (hi.trans hi')

theorem **cup_cast_left** {q : ℕ} {E : Spc} {n i n' i' m j : ℤ} (hn : n = n') (hi : i = i') (a : Ht q E n i) (b : Ht q E m j) : cup q (a.cast hn hi) b = (cup q a b).cast (by rw [hn]) (by rw [hi])

theorem **Ht.cast_self** {E : Spc} {n i : ℤ} (hn : n = n) (hi : i = i) (x : Ht q E n i) : x.cast hn hi = x

theorem **cup_cast_right** {E : Spc} {n i m j m' j' : ℤ} (hm : m = m') (hj : j = j') (a : Ht q E n i) (b : Ht q E m j) : cup q a (b.cast hm hj) = (cup q a b).cast (by rw [hm]) (by rw [hj])

theorem **Ht.castHom_apply** {E : Spc} {n i n' i' : ℤ} (hn : n = n') (hi : i = i') (x : Ht q E n i) : Ht.castHom hn hi x = x.cast hn hi

private theorem **hm** : 2 * (dX : ℤ) - (b + 2 * ((dX : ℤ) - dZ)) = 2 * dZ - b

private theorem **hj** : (dX : ℤ) - (i + ((dX : ℤ) - dZ)) = dZ - i

theorem **form_add** (a a' : Ht q (sch.obj Z) b i) : form f dZ dX tZ b i (a + a') = form f dZ dX tZ b i a + form f dZ dX tZ b i a'

theorem **pairing_add** {d : ℕ} (t : Ht q (sch.obj X) (2 * d) d →+ ZMod q) {n j : ℤ} (c c' : Ht q (sch.obj X) n j) (y : Ht q (sch.obj X) (2 * d - n) (d - j)) : pairing q d t (c + c') y = pairing q d t c y + pairing q d t c' y

theorem **pairing_zero** {d : ℕ} (t : Ht q (sch.obj X) (2 * d) d →+ ZMod q) {n j : ℤ} (y : Ht q (sch.obj X) (2 * d - n) (d - j)) : pairing q d t (0 : Ht q (sch.obj X) n j) y = 0

theorem **pairing_iff** (a : Ht q (sch.obj Z) b i) (c : Ht q (sch.obj X) (b + 2 * ((dX : ℤ) - dZ)) (i + ((dX : ℤ) - dZ))) : (∀ y, form f dZ dX tZ b i a y = pairing q dX tX c y) ↔ ∀ y : Ht q (sch.obj X) (2 * dZ - b) (dZ - i), tX ((cup q c y).cast (by ring) (by ring)) = tZ ((cup q a (pull _ (sch.map f) _ _ y)).cast (by ring) (by ring))

theorem **exists_isTranspose** (hX : IsPerfect q dX tX) : ∃ γ, IsTranspose q f dZ dX tZ tX b i γ

theorem **IsTranspose.unique** (hX : IsPerfect q dX tX) {γ γ' : Ht q (sch.obj Z) b i →+ Ht q (sch.obj X) (b + 2 * ((dX : ℤ) - dZ)) (i + ((dX : ℤ) - dZ))} (h : IsTranspose q f dZ dX tZ tX b i γ) (h' : IsTranspose q f dZ dX tZ tX b i γ') : γ = γ'

theorem **pushWith_isTranspose** (hX : IsPerfect q dX tX) : IsTranspose q f dZ dX tZ tX b i (pushWith q f dZ dX tZ tX b i)

theorem **pushWith_eq_iff** (hX : IsPerfect q dX tX) (γ : Ht q (sch.obj Z) b i →+ Ht q (sch.obj X) (b + 2 * ((dX : ℤ) - dZ)) (i + ((dX : ℤ) - dZ))) : pushWith q f dZ dX tZ tX b i = γ ↔ IsTranspose q f dZ dX tZ tX b i γ

theorem **pushWith_id** {d : ℕ} {t : Ht q (sch.obj X) (2 * d) d →+ ZMod q} (ht : IsPerfect q d t) : pushWith q (𝟙 X) d d t t b i = Ht.castHom (by ring) (by ring)

theorem **tr_spec** (h : ∃ T, IsTraceFam T) : IsTraceFam tr

theorem **gysin_isTranspose** (h : ∃ T, IsTraceFam T) (k : Type) [Field k] [IsAlgClosed k] (q : ℕ) [NeZero q] (hq : IsUnit (q : k)) {Z X : Scheme.{0}} (f : Z ⟶ X) (p : X ⟶ Spec (CommRingCat.of k)) [IsProper p] (dZ dX : ℕ) [SmoothOfRelativeDimension dX p] (b i : ℤ) : IsTranspose q f dZ dX (tr q Z dZ) (tr q X dX) b i (gysin q f dZ dX b i)

theorem **gysin_id** (h : ∃ T, IsTraceFam T) (k : Type) [Field k] [IsAlgClosed k] (q : ℕ) [NeZero q] (hq : IsUnit (q : k)) {X : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of k)) [IsProper p] (d : ℕ) [SmoothOfRelativeDimension d p] (b i : ℤ) : gysin q (𝟙 X) d d b i = Ht.castHom (by ring) (by ring)

theorem **gysinHl_apply** {Z X : Scheme.{0}} (f : Z ⟶ X) (dZ dX : ℕ) (b i : ℤ) (h : ∃ g : Hl l (sch.obj Z) b i →ₗ[ℤ_[l]] Hl l (sch.obj X) (b + 2 * ((dX : ℤ) - dZ)) (i + ((dX : ℤ) - dZ)), ∀ (x : Hl l (sch.obj Z) b i) (k : ℕ), (g x).1 k = gysin (l ^ k) f dZ dX b i (x.1 k)) (x : Hl l (sch.obj Z) b i) (k : ℕ) : (gysinHl l f dZ dX b i x).1 k = gysin (l ^ k) f dZ dX b i (x.1 k)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ReadsMove

theorem **ReadsOne.readsK** {i : ℕ} {hi : i ≤ B.R.m} {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {Ch : Type} {chOf : (I.stp u a i).Cl → Ch} {K : Fam I l Z u a i hi bad chOf} (h : ReadsOne I l Z u a i hi chOf K) : I.ReadsK l Z u a i hi chOf K

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.RealPointsToUnitaryGroupVal

theorem **toF_apply** (γ : G.GQ) : toF G γ = (G.toAf γ, ubc ℚ_[p] γ)

theorem **isOpen_level** (L : G.Level) {Kp : Subgroup G.GQp} (h : IsOpen (Kp : Set G.GQp)) : IsOpen (level G L Kp : Set (Gf G))

theorem **isCompact_level** (L : G.Level) {Kp : Subgroup G.GQp} (h : IsCompact (Kp : Set G.GQp)) : IsCompact (level G L Kp : Set (Gf G))

theorem **mem_piPart** {V : Type} [AddCommGroup V] [Module ℂ V] {π : Representation ℂ (Gf G) V} {Kf : Subgroup (Gf G)} {d : ℕ} {ξ₀ : G.GQ →* Matrix (Fin d) (Fin d) ℂ} {q : ℕ} {φ : GlobalTheory.HkPart.fix π Kf →ₗ[ℂ] coh G Kf ξ₀ q} : φ ∈ piPart G π Kf ξ₀ q ↔ ∀ ⦃n : ℕ⦄ (z : Fin

n → Gf G) (hz : IsHeckeFam Kf z), φ ∘ₗ GlobalTheory.HkPart.op π hz = hecke G hz ξ₀ q ∘ₗ φ

theorem **mem_gam_mk** (x : Gf G) (γ : G.GQ) : γ ∈ gam G Kf (x : Gf G ⧸ Kf) ↔ x⁻¹ * toF G γ * x ∈ Kf

theorem **pt_cover** : letI

theorem **pt_disjoint** : letI

theorem **UJfam_le_UJpm** (ε : T → Fin n → Bool) : UJfam ε ≤ UJpm ε

theorem **Jm_eps** (v : InfinitePlace G.F) : Jm (eps G v) = diagonal (GlobalTheory.HermCongruence.signVec (G.gram_herm v))

theorem **pOf_eps** (v : InfinitePlace G.F) : pOf (eps G v) = sigPosH (G.gram_herm v)

theorem **frame_spec** (v : InfinitePlace G.F) : ((frame G v : Matrix (Fin G.n) (Fin G.n) ℂ))ᴴ * G.gram v * (frame G v : Matrix (Fin G.n) (Fin G.n) ℂ) = Jm (eps G v)

theorem **conjGL_apply** {m : ℕ} (P : GL (Fin m) ℂ) (X : Matrix (Fin m) (Fin m) ℂ) : conjGL P X = ((P⁻¹ : GL (Fin m) ℂ) : Matrix (Fin m) (Fin m) ℂ) * X * P

theorem **conjGL_sim** {m : ℕ} (P : GL (Fin m) ℂ) (h X : Matrix (Fin m) (Fin m) ℂ) (c : ℂ) (hX : Xᴴ * h * X = c • h) : (conjGL P X)ᴴ * ((P : Matrix (Fin m) (Fin m) ℂ)ᴴ * h * P) * conjGL P X = c • ((P : Matrix (Fin m) (Fin m) ℂ)ᴴ * h * P)

theorem **mat_apply** (x : ℝ ⊗[ℚ] G.C) (v : InfinitePlace G.F) : mat G x v = conjGL (frame G v) (G.psiR v x)

theorem **psiR_star** (v : InfinitePlace G.F) (x : ℝ ⊗[ℚ] G.C) : G.psiR v (starT G.star ℝ x).unop = (G.gram v)⁻¹ * (G.psiR v x)ᴴ * G.gram v

theorem **psiR_sim** (g : G.GR) (v : InfinitePlace G.F) : (G.psiR v ((g : (ℝ ⊗[ℚ] G.C)ˣ × ℝˣ).1 : ℝ ⊗[ℚ] G.C))ᴴ * G.gram v * G.psiR v ((g : (ℝ ⊗[ℚ] G.C)ˣ × ℝˣ).1 : ℝ ⊗[ℚ] G.C) = ((((g : (ℝ ⊗[ℚ] G.C)ˣ × ℝˣ).2 : ℝˣ) : ℝ) : ℂ) • G.gram v

theorem **invSqrtAbs_sq** (c : ℝˣ) : ((invSqrtAbs c : ℝˣ) : ℝ) * (invSqrtAbs c : ℝˣ) * c = SignType.sign (c : ℝ)

theorem **normR_val** (g : G.GR) : ((normR G g : (ℝ ⊗[ℚ] G.C)ˣ) : ℝ ⊗[ℚ] G.C) = algebraMap ℝ (ℝ ⊗[ℚ] G.C) ((invSqrtAbs (uSimR ℝ g) : ℝˣ) : ℝ) * ((g : (ℝ ⊗[ℚ] G.C)ˣ × ℝˣ).1 : ℝ ⊗[ℚ] G.C)

theorem **mat_normR_sim** (g : G.GR) (v : InfinitePlace G.F) : (mat G (normR G g : ℝ ⊗[ℚ] G.C) v)ᴴ * Jm (eps G v) * mat G (normR G g : ℝ ⊗[ℚ] G.C) v = (((SignType.sign (((g : (ℝ ⊗[ℚ] G.C)ˣ × ℝˣ).2 : ℝˣ) : ℝ) : ℝ)) : ℂ) • Jm (eps G v)

theorem **realPointsToUnitaryGroup_val** (g : G.GR) (v : InfinitePlace G.F) : ((realPointsToUnitaryGroup G g : (MatT (InfinitePlace G.F) G.n)ˣ) : MatT (InfinitePlace G.F) G.n) v = mat G (normR G g : ℝ ⊗[ℚ] G.C) v

theorem **coe_toUJ** (h : PosSim G) (g : G.GR) : ((toUJ G h g : UJfam (eps G)) : (MatT (InfinitePlace G.F) G.n)ˣ) = realPointsToUnitaryGroup G g

theorem **aniso_of_division** (h : ∀ x : G.C, x ≠ 0 → IsUnit x) : Aniso G

theorem **const_mem_invariants** : (fun _ _ => (1 : ℂ) : Gf G ⧸ Kf → Fin 1 → ℂ) ∈ (funRepY (toF G) Kf (xiTriv G)).invariants

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ReductionUniformizerPoint

theorem **le_ideal** {S : Finset ι} {i : ι} (h : i ∈ S) : I i ≤ ideal I S

theorem **ideal_mono** {S S' : Finset ι} (h : S ⊆ S') : ideal I S ≤ ideal I S'

@[simp] theorem **ideal_empty** : ideal I ∅ = ⊥

@[simp] theorem **ideal_singleton** (i : ι) : ideal I {i} = I i

@[simp] theorem **ker_Yι** (S : Finset ι) : (Yι I S).ker = ideal I S

theorem **incl_Yι** {S S' : Finset ι} (h : S ⊆ S') : incl I h ≫ Yι I S = Yι I S'

theorem **incl_refl** (S : Finset ι) : incl I (subset_refl S) = 𝟙 (Y I S)

theorem **incl_comp** {S S' S'' : Finset ι} (h : S ⊆ S') (h' : S' ⊆ S'') : incl I h' ≫ incl I h = incl I (h.trans h')

theorem **inj_Ystrπ** {s : ℕ} (S : Idx ι s) : inj I S ≫ Ystrπ I s = Yι I S.1

theorem **Ystr_hom_ext** {s : ℕ} {Z : Scheme.{0}} {f g : Ystr I s ⟶ Z} (h : ∀ S : Idx ι s, inj I S ≫ f = inj I S ≫ g) : f = g

theorem **incl_inj_congr** {s : ℕ} {A : Finset ι} {T T' : Idx ι s} (e : T.1 = T'.1) (h : T.1 ⊆ A) (h' : T'.1 ⊆ A) : incl I h ≫ inj I T = incl I h' ≫ inj I T'

theorem **face_subset** {s : ℕ} (S : Idx ι (s + 1)) (j : Fin (s + 2)) : (face S j).1 ⊆ S.1

theorem **orderEmbOfFin_erase** {S : Finset ι} {n : ℕ} (h : S.card = n + 1) (k : Fin (n + 1)) (h' : (S.erase (S.orderEmbOfFin h k)).card = n) (m : Fin n) : (S.erase (S.orderEmbOfFin h k)).orderEmbOfFin h' m = S.orderEmbOfFin h (k.succAbove m)

theorem **face_face** {s : ℕ} (S : Idx ι (s + 2)) {i j : Fin (s + 2)} (hij : i ≤ j) : (face (face S j.succ) i).1 = (face (face S i.castSucc) j).1

theorem **inj_δ** {s : ℕ} (j : Fin (s + 2)) (S : Idx ι (s + 1)) : inj I S ≫ δ I s j = incl I (face_subset S j) ≫ inj I (face S j)

theorem **δ_Ystrπ** (s : ℕ) (j : Fin (s + 2)) : δ I s j ≫ Ystrπ I s = Ystrπ I (s + 1)

theorem **δ_comp_δ** {s : ℕ} {i j : Fin (s + 2)} (hij : i ≤ j) : δ I (s + 1) j.succ ≫ δ I s i = δ I (s + 1) i.castSucc ≫ δ I s j

theorem **toFibre_fst** (S : Finset ι) (h : fibreIdeal p ϖ ≤ ideal I S) : toFibre I p ϖ S h ≫ pullback.fst p (resMap ϖ) = Yι I S

theorem **fibreIdeal_le_ideal** {S : Finset ι} (hS : S.Nonempty) (h : ∀ i, fibreIdeal p ϖ ≤ I i) : fibreIdeal p ϖ ≤ ideal I S

theorem **incl_toFibre** {S S' : Finset ι} (hSS : S ⊆ S') (h : fibreIdeal p ϖ ≤ ideal I S) : incl I hSS ≫ toFibre I p ϖ S h = toFibre I p ϖ S' (h.trans (ideal_mono I hSS))

theorem **prod_le** (i : ι) [Fintype ι] [DecidableEq ι] : ∏ j, I j ≤ I i

theorem **fibreIdeal_le_of_prod** [Fintype ι] (h : ∏ j, I j = fibreIdeal p ϖ) (i : ι) : fibreIdeal p ϖ ≤ ideal I {i}

theorem **ideal_pull** (S : Finset ι) : ideal (pull I f) S = (ideal I S).comap f

theorem **pullIso_hom_fst** (S : Finset ι) : (pullIso I f S).hom ≫ pullback.fst _ _ = Yι (pull I f) S

theorem **pullMap_Yι** (S : Finset ι) : pullMap I f S ≫ Yι I S = Yι (pull I f) S ≫ f

theorem **nonempty_Ystr** {s : ℕ} (S : Idx ι s) (y : Y I S.1) : Nonempty (Ystr I s)

theorem **isEmpty_Ystr** {s : ℕ} (h : IsEmpty (Idx ι s)) : IsEmpty (Ystr I s)

theorem **nonempty_Y_iff** (S : Finset ι) : Nonempty (Y I S) ↔ ((ideal I S).support : Set X).Nonempty

theorem **inj_YstrToFibre** (h : ∀ i, fibreIdeal p ϖ ≤ I i) {s : ℕ} (S : Idx ι s) : inj I S ≫ YstrToFibre I p ϖ h s = toFibre I p ϖ S.1 (fibreIdeal_le_ideal I p ϖ (Finset.card_pos.1 (by rw [S.2]; omega)) h)

theorem **YstrToFibre_fst** (h : ∀ i, fibreIdeal p ϖ ≤ I i) (s : ℕ) : YstrToFibre I p ϖ h s ≫ pullback.fst p (resMap ϖ) = Ystrπ I s

theorem **δ_YstrToFibre** [LinearOrder ι] (h : ∀ i, fibreIdeal p ϖ ≤ I i) (s : ℕ) (j : Fin (s + 2)) : δ I s j ≫ YstrToFibre I p ϖ h s = YstrToFibre I p ϖ h (s + 1)

theorem **incl_pullMap** {S S' : Finset ι} (h : S ⊆ S') : incl (pull I f) h ≫ pullMap I f S = pullMap I f S' ≫ incl I h

theorem **inj_pullStr** {s : ℕ} (S : Idx ι s) : inj (pull I f) S ≫ pullStr I f s = pullMap I f S.1 ≫ inj I S

theorem **pullStr_Ystrπ** (s : ℕ) : pullStr I f s ≫ Ystrπ I s = Ystrπ (pull I f) s ≫ f

theorem **δ_pullStr** [LinearOrder ι] (s : ℕ) (j : Fin (s + 2)) : δ (pull I f) s j ≫ pullStr I f s = pullStr I f (s + 1) ≫ δ I s j

theorem **reduction_uniformizerPoint** (k : Fin 2) : red 0 ϖ (T 0 ϖ k) = 0

theorem **mem_support_compIdeal** (x : XS 0 ϖ) (k : Fin 2) (hx : T 0 ϖ k ∈ x.asIdeal) : x ∈ ((compIdeal 0 ϖ k).support : Set (XS 0 ϖ))

theorem **isEmpty_Ystr_two** : IsEmpty (Ystr (compIdeal 0 ϖ) 2)

theorem **nonempty_Ystr_one** (h : ¬ IsUnit ϖ) : Nonempty (Ystr (compIdeal 0 ϖ) 1)

theorem **orderEmbOfFin_univ_fin** {n : ℕ} (h : (Finset.univ : Finset (Fin n)).card = n) : ((Finset.univ : Finset (Fin n)).orderEmbOfFin h : Fin n → Fin n) = id

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.RegPolOfRepOf

theorem **Near.refl** (x : WD h) : Near x x

theorem **Near.of_eq** {x y : WD h} (e : x = y) : Near x y

theorem **Near.lpDim_eq** {x y : WD h} (n : Near x y) : x.lpDim = y.lpDim

theorem **BCof.repOf** {π : CuspSub K a} {π' : CuspSub M a} (hb : BCof Ω π π') {l : ℕ} [Fact l.Prime] {ι : Qbar l ≃+* ℂ} {r : G K Ω →* GL (Fin a) (Qbar l)} (hr : RepOf π ι r) : RepOf π' ι (resTo M r)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ResidualIrreducibilityFormsEquivalent

theorem **Core.admA** {D : D7} (h : Core D) : SumRepresentations.AdmA D

theorem **Core.a3** {D : D7} (h : Core D) (L : SizeData₂ D) : TwistedPlaces.adm3 L

theorem **Core.isot** {D : D7} (h : Core D) : Rows.IsotInternal D.what₂

theorem **residualIrreducibility_forms_equivalent** (D : D7) : SumRepresentations.adm6 D ↔ ResidualIrreducibleRankR D

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.RingHomFormallyEtaleOfBijective

theorem **ringHom_formallyEtale_of_bijective** {A B : Type} [CommRing A] [CommRing B] (g : A →+* B) (hg : Function.Bijective g) : g.FormallyEtale

theorem **etaleAtPt_of_isIso** (f : X ⟶ Y) [IsIso f] (x : X) : EtaleAtPt f x

theorem **isClosed_stratSet** (j : Fin n) : IsClosed (stratSet C j)

@[simp] theorem **coe_support_stratIdeal** (j : Fin n) : ((stratIdeal C j).support : Set S) = stratSet C j

theorem **stratSet_eq_empty_of_isIso** (j : Fin n) [IsIso (C.φ j)] : stratSet C j = ∅

theorem **stratIdeal_eq_top_of_isIso** (j : Fin n) [IsIso (C.φ j)] : stratIdeal C j = ⊤

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.RkLe

theorem **rkS** {j : ℕ} (hN : RankTie X D j) : (X.Pj j).m + 1 = D.R.N j

theorem **rkT** {j : ℕ} (hN : RankTie X D j) : (X.Pj (j + 1)).m + 1 = D.R.N (j + 1)

theorem **rk_le** {j : ℕ} (hN : RankTie X D j) : D.R.N j ≤ D.R.N (j + 1)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.RowStarCoreAtUOfRowStarCoreU

theorem **HasRepU.mono** {ι : Qbar l ≃+* ℂ} {U : Submodule ℂ (UA E a → ℂ)} {S S' : Finset (HeightOneSpectrum (𝒪 E))} (hS : S ⊆ S') {r : G E Ω →* GL (Fin a) (Qbar l)} (h : HasRepU ι U S r) : HasRepU ι U S' r

theorem **finite_not_apart** (w₀ : HeightOneSpectrum (𝒪 E)) : {w : HeightOneSpectrum (𝒪 E) | ¬ Apart w w₀}.Finite

theorem **mem_nearPlaces** {w w₀ : HeightOneSpectrum (𝒪 E)} : w ∈ nearPlaces w₀ ↔ ¬ Apart w w₀

theorem **isSplitAt_of_not_apart** {w w₀ : HeightOneSpectrum (𝒪 E)} (h₀ : IsSplitAt w₀) (h : ¬ Apart w w₀) : IsSplitAt w

theorem **RowStarCoreAtU.near** (h : RowStarCoreAtU) (ι : Qbar l ≃+* ℂ) (ha : 1 ≤ a) {T : Finset (HeightOneSpectrum (𝒪 E))} (hT : NonSplitSet T) {w₀ : HeightOneSpectrum (𝒪 E)} (hl : ((l : ℕ) : 𝒪 E) ∈ w₀.asIdeal) (h₀ : IsSplitAt w₀) {U : Submodule ℂ (UA E a → ℂ)} (hU : IsAutRepU U) (hu : ∀ w, ¬ IsSplitAt w → w ∉ T → IsUnramU U w) : ∃ r : G E Ω →* GL (Fin a) (Qbar l), HasRepU ι U (nearPlaces w₀) r

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.SameOffRefl

theorem **phiA_incl** (A : LocAt F E n G) (v : Fin G.P.nV) (x : E) : phiA A v (x ⊗ₜ[F] 1) = A.iw v x

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.SectionOfFreeOnUnitComp

theorem **freeOnUnit_comp_bijective** (X : Scheme.{0}) (G : Shv S) : Function.Bijective fun g : (freeOn S).obj X ⟶ G => freeOnUnit φ X ≫ (resShv φ).map g

theorem **resEquivHS_apply** (G : Shv S) (n : ℕ) (X : Scheme.{0}) (y : HS S G n X) : resEquivHS φ G n X y = (Ext.mk₀ (freeOnUnit φ X)).comp (y.mapExactFunctor (resShv φ)) (zero_add n)

theorem **resEquivHS_chgS** {F : Shv R} {G : Shv S} (u : F ⟶ (resShv φ).obj G) (n : ℕ) (X : Scheme.{0}) (x : HS R F n X) : resEquivHS φ G n X (chgS φ u n X x) = x.comp (Ext.mk₀ u) (add_zero n)

theorem **chgS_mk₀** {F : Shv R} {G : Shv S} (u : F ⟶ (resShv φ).obj G) (X : Scheme.{0}) (g : (freeOn R).obj X ⟶ F) (g' : (freeOn S).obj X ⟶ G) (h : freeOnUnit φ X ≫ (resShv φ).map g' = g ≫ u) : chgS φ u 0 X (Ext.mk₀ g) = Ext.mk₀ g'

theorem **sectionOf_comp** (X : Scheme.{0}) {F G : Shv R} (g : (freeOn R).obj X ⟶ F) (u : F ⟶ G) : sectionOf R X G (g ≫ u) = u.hom.app (op X) (sectionOf R X F g)

theorem **sectionOf_freeOnUnit_comp** (X : Scheme.{0}) {G : Shv S} (g : (freeOn S).obj X ⟶ G) : sectionOf R X ((resShv φ).obj G) (freeOnUnit φ X ≫ (resShv φ).map g) = sectionOf S X G g

theorem **coefMap_toSheafify** (U : Scheme.{0}ᵒᵖ) (g : C(U.unop, R)) : (coefMap φ hφ).hom.app U ((toSheafify Scheme.proetaleTopology.{0} (coefPre R)).app U (ULift.up g)) = (toSheafify Scheme.proetaleTopology.{0} (coefPre S)).app U (ULift.up (((φ, hφ) : C(R, S)).comp g))

theorem **toSheafify_e_mk₀** (X : Scheme.{0}) (g : (freeOn R).obj X ⟶ coef R) : (toSheafify Scheme.proetaleTopology.{0} (coefPre R)).app (op X) (ULift.up (H0.e R X (Ext.mk₀ g))) = sectionOf R X (coef R) g

theorem **toSheafify_up_injective** (U : Scheme.{0}ᵒᵖ) {a b : C(U.unop, R)} (h : (toSheafify Scheme.proetaleTopology.{0} (coefPre R)).app U (ULift.up a) = (toSheafify Scheme.proetaleTopology.{0} (coefPre R)).app U (ULift.up b)) : a = b

theorem **e_chg** (X : Scheme.{0}) (x : H R 0 X) : H0.e S X (chg φ hφ 0 X x) = ((φ, hφ) : C(R, S)).comp (H0.e R X x)

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.SpKerColToEqColCpl

theorem **eq_cpl** (hm : m ≠ n + 1) (h : i = n + 1) (b : SpIdx β n m i) : b = .cpl h

theorem **eq_top** (hm : n + 1 < m) (h : i = m) (b : SpIdx β n m i) : b = .top h

theorem **isEmpty_of_lt** (hm : n + 1 < m) (h : m < i) : IsEmpty (SpIdx β n m i)

theorem **spFam_mem_mat** (i : ℕ) (b : (F i).β) (h : i ≤ n) : (spFam F n m C T i).mem (.mat b h) = (F i).mem b

theorem **spFam_mem_cpl** (h : n + 1 = n + 1) : (spFam F n m C T (n + 1)).mem (.cpl h) = C.gsd

theorem **spFam_mem_top** (h : m = m) : (spFam F n m C T m).mem (.top h) = T.gsd

theorem **spFam_ok** (hF : ∀ i, i ≤ n → (F i).Ok) (hC : C.gsd.Ok) (hT : T.gsd.Ok) (i : ℕ) : (spFam F n m C T i).Ok

theorem **spKer_mat** (i : ℕ) (al : (F i).β) (be : (F (i + 1)).β) (h : i ≤ n) (h' : i + 1 ≤ n) : spKer kk k1 k2 i (.mat al h) (.mat be h') = kk i al be

theorem **spKer_cpl** (al : (F n).β) (h : n ≤ n) (h' : n + 1 = n + 1) : spKer kk k1 k2 n (.mat al h) (.cpl h') = k1 al

theorem **spEntry_top** (e : m = n + 1 + 1) (h : n + 1 = n + 1) (h' : m = m) : spEntry kk k1 k2 (n + 1) m e (.cpl h) (.top h') = k2

theorem **spIn_mat** (i : ℕ) (φ : (F i).Forms) (al : (F i).β) (h : i ≤ n) : spIn n m C T i φ (.mat al h) = φ al

theorem **spKer_colTo_low** (hm : n < m) (hF : ∀ i, (F i).Ok) (φ : (F 0).Forms) (i : ℕ) (hi : i ≤ n) (al : (F i).β) : (spKer kk k1 k2).colTo hS i (spIn n m C T 0 φ) (.mat al hi) = kk.colTo hF i φ al

theorem **spKer_colTo_cpl** (hm : n < m) (hF : ∀ i, (F i).Ok) (φ : (F 0).Forms) (h : n + 1 = n + 1) : (spKer kk k1 k2).colTo hS (n + 1) (spIn n m C T 0 φ) (.cpl h) = ∑ al : (F n).β, (k1 al).op (hF n al) (kk.colTo hF n φ al)

theorem **spKer_colTo_top** (hm : m = n + 2) (hF : ∀ i, (F i).Ok) (hC : C.gsd.Ok) (φ : (F 0).Forms) (h : m = m) : (spKer kk k1 k2).colTo hS m (spIn n m C T 0 φ) (.top h) = k2.op hC (∑ al : (F n).β, (k1 al).op (hF n al) (kk.colTo hF n φ al))

theorem **spKer_colTo_eq_colCpl** {F : ℕ → GFam K} {n m : ℕ} (hm : m = n + 2) (hF : ∀ i, (F i).Ok) {b : (F (n + 1)).β} {Y : Cpl ((F (n + 1)).mem b)} {T : MemOn (F m)} (kk : KerFam F) (k1 : ∀ al : (F n).β, ClsKer ((F n).mem al) (MemOn.pairs (F (n + 1)) b Y).gsd) (k2 : ClsKer (MemOn.pairs (F (n + 1)) b Y).gsd T.gsd) (hS : ∀ i, (spFam F n m (MemOn.pairs (F (n + 1)) b Y) T i).Ok) {S₁ : GSD K} (A : ∀ al : (F n).β, CplKer ((F n).mem al) ((F (n + 1)).mem b) S₁ Y) (til : S₁.D → T.D) (e : (F m).V →ₗ[K] S₁.V) (h1 : ∀ al d₀ (d : Y.Pts), (A al).k1 d₀ d.1.1 d.1.2 = (k1 al).k d₀ d) (h2 : ∀ al (d : Y.Pts) d₁, (A al).k2 d.1.1 d.1.2 d₁ = e ∘ₗ k2.k d (til d₁)) (φ : (F 0).Forms) (h : m = m) (d₁ : S₁.D) : e (((spKer kk k1 k2).colTo hS m (spIn n m _ T 0 φ) (.top h)).1 (til d₁)) = (colCpl kk hF n A (hF (n + 1) b) φ).1 d₁

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.SplitLiftOfLiftParam

theorem **upInj_of_lamPos** (i : N.Loc) (h : LamPos N i) : UpInj N i

theorem **locOfPt_of_repOfPt** (i : N.Loc) (h : RepOfPt N A i) : LocOfPt N A i

theorem **count_of_down** (i : N.Loc) (h1 : TypeNZ N A i) (h2 : DownMem N A i) : Count N A i

theorem **pairs_of_thetaOf** (i : N.Loc) (σ : what₂.FT) (P : A.AV i) (h : thetaOf N A i σ P) : Pairs N A i σ P

private theorem **foldr_lpSum_lpChar_comm** {ι : Type} (hL : ParamLaws N A) (χ : A.CG 1) (w : PlE H) (c : ι → ℝ) (b : A.LP w) (L : List ι) : L.foldr (fun k acc => A.lpSum (A.lpTw (A.lpChar b (A.locG χ w)) (c k)) acc) (A.lpZero w) = A.lpChar (L.foldr (fun k acc => A.lpSum (A.lpTw b (c k)) acc) (A.lpZero w)) (A.locG χ w)

theorem **speh_char** (hL : ParamLaws N A) (p : Part A.CG) (χ : A.CG 1) (w : PlE H) : spehLoc N A (p.a, A.tw p.π χ, p.d) w = A.lpChar (spehLoc N A p w) (A.locG χ w)

theorem **iso_char** (hL : ParamLaws N A) (S : List (Part A.CG)) (χ : A.CG 1) (w : PlE H) : isoLoc N A (twistIso N A S χ) w = A.lpChar (isoLoc N A S w) (A.locG χ w)

theorem **iso_append** (S : List (Part A.CG)) (p : Part A.CG) (w : PlE H) : isoLoc N A (S ++ [p]) w = A.lpSum (isoLoc N A S w) (spehLoc N A p w)

theorem **splitLift_of_liftParam** (hL : ParamLaws N A) (i : N.Loc) (h : LiftParam N A i) : SplitLift N A i

private theorem **pairs_of_starProjection_ne_zero** (i : N.Loc) (σ : what₂.FT) (P : A.AV i) (f : what₂.Mj (N.j i)) (hf : f ∈ what₂.isot (N.j i) σ) (hne : (A.typeV i P).starProjection (what₂.thetaT (N.mu i) (N.j i) f) ≠ 0) : Pairs N A i σ P

theorem **thetaOne_of** (h1 : ∀ i, UpInj N i) (h2 : ∀ i, HoweUp N A i) (h3 : ∀ i, TypeOrth N A i) : ThetaOne N A

theorem **iotaO_inj** (i : N.Loc) : Function.Injective (iotaO N A i)

theorem **exists_pt_of_loc** (i : N.Loc) (h : PtLocal N A i) (P P' : A.AV i) (x : N.X i) (hx : A.ptV i P' = some x) (hl : ∀ w, SplitOut N i w → A.locV i P w = A.locV i P' w) : ∃ y, A.ptV i P = some y ∧ ∀ w, SplitOut N i w → heckePol N i y w = heckePol N i x w

theorem **rowSize_of_le** (hq : ∀ w : PlE H, 1 < (normV w : ℕ)) (h : RowSizeLe N A) : RowSize N A

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.StabilisesColumnBaseLatticeIffFirstFactor

theorem **pairInt_of_mem_dualL** (hc : CMData F E) (S : HSpace E) (Λ : Submodule (𝒪 E) (Fin S.rank → E)) {b : Fin S.rank → E} (hb : b ∈ Formalism.L2.Herm.dualL hc S Λ) : ∀ a ∈ Λ, hform (hc.sig : E →+* E) S.gram a b ∈ (algebraMap (𝒪 E) E).range

theorem **selfDualAt_stepLevelQ** {S : HSpace E} {gp' : Fin 2 → ℕ} {T f : ℕ} {Q : Finset (HeightOneSpectrum (𝒪 F))} (hc : CMData F E) (X : Formalism.L2.Herm.StepLevelQ hc S gp' T f Q) (v : HeightOneSpectrum (𝒪 F)) (hT : (T : 𝒪 F) ∉ v.asIdeal) : SelfDualAt hc.sig S.gram v X.Lb

theorem **stab_iff** (w' : HeightOneSpectrum (𝒪 D.host.F)) (h : ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal) (hdual : SelfDualAt D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb) (x : GL (Fin (D.sp.X (i + 1)).rank) (D.host.E ⊗[D.host.F] w'.adicCompletion D.host.F)) (hx : x ∈ UPts D.sp.cm.sig (D.sp.X (i + 1)).gram (w'.adicCompletion D.host.F)) : x ∈ levelVL D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb ↔ (∀ a b, (((t.fr w' h)⁻¹ * bc ((RingHom.fst _ _).comp ((X.g u w hg).spl w' h).toRingHom) x * t.fr w' h : GL (Fin

(D.sp.X (i + 1)).rank) _) : Matrix (Fin (D.sp.X (i + 1)).rank) (Fin (D.sp.X (i + 1)).rank) _) a b ∈ w'.adicCompletionIntegers D.host.F) ∧ ∀ a b, ((((t.fr w' h)⁻¹ * bc ((RingHom.fst _ _).comp ((X.g u w hg).spl w' h).toRingHom) x * t.fr w' h)⁻¹ : GL (Fin (D.sp.X (i + 1)).rank) _) : Matrix (Fin (D.sp.X (i + 1)).rank) (Fin (D.sp.X (i + 1)).rank) _) a b ∈ w'.adicCompletionIntegers D.host

theorem **levelVL_le** (w' : HeightOneSpectrum (𝒪 D.host.F)) : levelVL D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb ≤ UPts D.sp.cm.sig (D.sp.X (i + 1)).gram (w'.adicCompletion D.host.F)

theorem **hypTJC_of_le** (hU : ∀ w' : HeightOneSpectrum (𝒪 D.host.F), ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal → Uloc w' ≤ levelVL D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb) (hdual : ∀ w' : HeightOneSpectrum (𝒪 D.host.F), ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal → SelfDualAt D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb) : HypTJC t

theorem **hypTJ_of_ge** (hU : ∀ w' : HeightOneSpectrum (𝒪 D.host.F), ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal → levelVL D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb ≤ Uloc w') (hdual : ∀ w' : HeightOneSpectrum (𝒪 D.host.F), ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal → SelfDualAt D.sp.cm.sig (D.sp.X (i + 1)).gram w' Lb) : HypTJ t

theorem **T_not_mem** {p T : ℕ} (hp : p.Prime) (hT : ¬ p ∣ T) (w' : HeightOneSpectrum (𝒪 D.host.F)) (h : ((p : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal) : (T : 𝒪 D.host.F) ∉ w'.asIdeal

theorem **hypTJC_stepLevelQ** (hLb : Lb = S.Lb) (hUl : Uloc = S.Uloc) : HypTJC t

theorem **hypTJ_stepLevelQ** (hLb : Lb = S.Lb) (hUl : Uloc = S.Uloc) (hgp : ∀ a, ¬ D.what₂.places.resChar w ∣ gp a) (hQ : ∀ w' ∈ Q, ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∉ w'.asIdeal) : HypTJ t

theorem **away_of_off** (c c' : (D.gat u w hg).comp) (δ : GL (Fin (D.R.N 0)) D.host.E) (hδ : δ ∈ UJ (sigD D : D.host.E →+* D.host.E) (gramN D 0 (Nat.zero_le _))) (hoff : ∀ w' : HeightOneSpectrum (𝒪 D.host.F), ((D.what₂.places.resChar w : ℕ) : 𝒪 D.host.F) ∉ w'.asIdeal → bc (prV (E

theorem **gamI_inj_locAt** (c : (D.gat u w hg).comp) : Function.Injective ((locAt X u w hg B).gamI c)

theorem **gamOnto_locAt** (c : (D.gat u w hg).comp) : GamOnto ((T u).cls0).lev (locAt X u w hg B) c

theorem **compInj_locAt** : CompInj ((T u).cls0).lev (locAt X u w hg B)

theorem **locHypC_locAt** (hB : FrRat X u w hg B) (hTc : HypTC (X.g u w hg)) : LocHypC ((T u).cls0).lev (locAt X u w hg B)

theorem **oneHyp_locAt** (hB : FrRat X u w hg B) (hT : HypT (X.g u w hg)) (hTc : HypTC (X.g u w hg)) : OneHyp ((T u).cls0).toChart.lev (locAt X u w hg B) ((T u).cls0).toChart.y where

theorem **selfDualAt_col** (L : ColLevel R F E sp gp C A clsLat) (v : HeightOneSpectrum (𝒪 F)) (hT : ((A.T : ℕ) : 𝒪 F) ∉ v.asIdeal) : SelfDualAt sp.cm.sig sp.gram0 v L.Lb

theorem **stabilises_columnBaseLattice_iff_firstFactor** (w : HeightOneSpectrum (𝒪 F)) (h : ((p : ℕ) : 𝒪 F) ∈ w.asIdeal) (hdual : SelfDualAt sp.cm.sig sp.gram0 w L.Lb) (x : GL (Fin R.k) (E ⊗[F] w.adicCompletion F)) (hx : x ∈ UPts sp.cm.sig sp.gram0 (w.adicCompletion F)) : x ∈ levelVL sp.cm.sig sp.gram0 w L.Lb ↔ (∀ a b, (((t.fr w h)⁻¹ * bc ((RingHom.fst _ _).comp (t.spl w h).toRingHom) x * t.fr w h : GL (Fin R.k) _) : Matrix (Fin R.k) (Fin R.k) _) a b ∈ w.adicCompletionIntegers F) ∧ (∀ a b, ((((t.fr w h)⁻¹ * bc ((RingHom.fst _ _).comp (t.spl w h).toRingHom) x * t.fr w h)⁻¹ : GL (Fin R.k) _) : Matrix (Fin R.k) (Fin R.k) _) a b ∈ w.adicCompletionIntegers F)

theorem **hypTC_col** (hT : ∀ w : HeightOneSpectrum (𝒪 F), ((p : ℕ) : 𝒪 F) ∈ w.asIdeal → ((A.T : ℕ) : 𝒪 F) ∉ w.asIdeal) : HypTC t

theorem **hypT_col** (hT : ∀ w : HeightOneSpectrum (𝒪 F), ((p : ℕ) : 𝒪 F) ∈ w.asIdeal → ((A.T : ℕ) : 𝒪 F) ∉ w.asIdeal) (hgp : ∀ a (w : HeightOneSpectrum (𝒪 F)), ((p : ℕ) : 𝒪 F) ∈ w.asIdeal → ((gp a : ℕ) : 𝒪 F) ∉ w.asIdeal) : HypT t

theorem **nat_not_mem** {p n : ℕ} (hp : p.Prime) (hn : ¬ p ∣ n) (w' : HeightOneSpectrum (𝒪 D.host.F)) (h : ((p : ℕ) : 𝒪 D.host.F) ∈ w'.asIdeal) : (n : 𝒪 D.host.F) ∉ w'.asIdeal

theorem **not_dvd_g** {u : D.what₂.Word} {w : D.what₂.places.Pl} (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) (a : Fin 2) : ¬ D.what₂.places.resChar w ∣ D.host.g a

theorem **hypTC_rec** : HypTC (X.g u w hg)

theorem **hypT_rec** : HypT (X.g u w hg)

theorem **oneHyp_rec** (K : CmpTied D X T Gt c) (hJ : Joint0 X.lev0 u (T u)) : OneHyp ((T u).cls0).toChart.lev (locAt X u w hg K.B) ((T u).cls0).toChart.y

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.SurjectiveOfLocallySurjective

theorem **projTrace_dualTensorHom** (t : Module.Dual R M ⊗[R] M) : projTrace R M (dualTensorHom R M M t) = contractLeft R M t

theorem **projTrace_smulRight** (ξ : Module.Dual R M) (m : M) : projTrace R M (ξ.smulRight m) = ξ m

theorem **projTrace_eq_trace** [Module.Free R M] (φ : Module.End R M) : projTrace R M φ = LinearMap.trace R M φ

theorem **projTrace_of_isBaseChange** {S N : Type*} [CommRing S] [Algebra R S] [AddCommGroup N] [Module R N] [Module S N] [IsScalarTower R S N] [Module.Finite S N] [Module.Projective S N] {l : M →ₗ[R] N} (hl : IsBaseChange S l) (φ : Module.End R M) (ψ : Module.End S N) (h : ∀ x, ψ (l x) = l (φ x)) : projTrace S N ψ = algebraMap R S (projTrace R M φ)

theorem **projTrace_id_of_rankAtStalk** {d : ℕ} (h : ∀ p, Module.rankAtStalk (R

theorem **algTrace_apply** (b : B') : algTrace B B' b = projTrace B B' (LinearMap.mul B B' b)

theorem **algTrace_eq_trace** [Module.Free B B'] (b : B') : algTrace B B' b = Algebra.trace B B' b

theorem **algTrace_of_isPushout** {C C' : Type*} [CommRing C] [CommRing C'] [Algebra B C] [Algebra C C'] [Algebra B' C'] [Algebra B C'] [IsScalarTower B C C'] [IsScalarTower B B' C'] [Module.Finite C C'] [Module.Projective C C'] [h : Algebra.IsPushout B C B' C'] (b : B') : algTrace C C' (algebraMap B' C' b) = algebraMap B C (algTrace B B' b)

theorem **algTrace_one** {d : ℕ} (h : ∀ p, Module.rankAtStalk (R

theorem **algTrace_algebraMap_of_bijective** (h : Function.Bijective (algebraMap B B')) (b : B) : algTrace B B' (algebraMap B B' b) = b

theorem **semiHom_algebraMap** : SemiHom (RingHom.id k) (algebraMap B B')

theorem **formLin_apply** {j : ℕ} (x : Om k B j) : formLin k B B' j x = (semiHom_algebraMap k B B').mOm j x

theorem **isBaseChange_formLin** [Algebra.FormallyEtale B B'] (j : ℕ) : IsBaseChange B' (formLin k B B' j)

theorem **projective_of_etale** [Module.Finite B B'] [Algebra.Etale B B'] : Module.Projective B B'

theorem **formTrace_smul_formLin** {j : ℕ} (b : B') (x : Om k B j) : formTrace k B B' j (b • formLin k B B' j x) = etTrace B B' b • x

theorem **surjective_of_locally_surjective** {A M N : Type*} [CommRing A] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module A N] (θ : M →ₗ[A] N) (Z : A → Submodule A N) (h1 : ∀ g h, Z g ≤ Z (g * h)) (h2 : ∀ g (m : ℕ) n, g ^ m • n ∈ Z g → n ∈ Z g) (h3 : ∀ t : Set A,

Ideal.span t = ⊤ → ∀ n, (∀ g ∈ t, n ∈ Z g) → n = 0) (h4 : ∀ g y, θ y ∈ Z g → ∃ m : ℕ, g ^ m • y = 0) (h5 : ∀ n, Ideal.span {g | ∃ (m : ℕ) (y : M), g ^ m • n - θ y ∈ Z g} = ⊤) : Function.Surjective θ

theorem **toOm_apply** (U : X.Opens) (x : OmU s U j) : toOm s j U x = (toOmega s j).app (op U) x

theorem **res_toOm** {U V : X.Opens} (h : V ≤ U) (x : OmU s U j) : (Omega s j).presheaf.map (homOfLE h).op (toOm s j U x) = toOm s j V (formRes s h j x)

theorem **exists_formRes_eq_zero** {U : X.Opens} {x : OmU s U j} (h : toOm s j U x = 0) {p : X} (hp : p ∈ U) : ∃ V : X.Opens, ∃ hV : V ≤ U, p ∈ V ∧ formRes s hV j x = 0

theorem **exists_toOm_eq_res** {U : X.Opens} (y : Γ(Omega s j, U)) {p : X} (hp : p ∈ U) : ∃ V : X.Opens, ∃ hV : V ≤ U, p ∈ V ∧ ∃ x : OmU s V j, toOm s j V x = (Omega s j).presheaf.map (homOfLE hV).op y

theorem **formRes_basicOpen** {U : X.Opens} (hU : IsAffineOpen U) (g : Sec s U) : (∀ y : OmU s (X.basicOpen g) j, ∃ (n : ℕ) (x : OmU s U j), Sec.res s (X.basicOpen_le g) g ^ n • y = formRes s (X.basicOpen_le g) j x) ∧ ∀ x : OmU s U j, formRes s (X.basicOpen_le g) j x = 0 → ∃ n : ℕ, g ^ n • x = 0

theorem **span_eq_top_of_forall_mem** {U : X.Opens} (hU : IsAffineOpen U) {t : Set (Sec s U)} (h : ∀ p ∈ U, ∃ g ∈ t, p ∈ X.basicOpen (U

theorem **toOm_injective** {U : X.Opens} (hU : IsAffineOpen U) : Function.Injective (toOm s j U)

theorem **toOm_surjective** {U : X.Opens} (hU : IsAffineOpen U) : Function.Surjective (toOm s j U)

theorem **toOm_bijective** {U : X.Opens} (hU : IsAffineOpen U) : Function.Bijective (toOm s j U)

@[simp] theorem **toOmEquiv_apply** {U : X.Opens} (hU : IsAffineOpen U) (x : OmU s U j) : toOmEquiv s j hU x = toOm s j U x

theorem **section_ext_affine** {U : X.Opens} {a b : Γ(F, U)} (h : ∀ (V : X.Opens) (_ : IsAffineOpen V) (hV : V ≤ U), F.presheaf.map (homOfLE hV).op a = F.presheaf.map (homOfLE hV).op b) : a = b

theorem **isBasis_range_affine** (X : Scheme.{0}) : Opens.IsBasis (Set.range fun U : X.affineOpens ↦ U.1)

theorem **preHomOfAffine_app** (U : X.affineOpens) (x : Γ(G, U.1)) : (preHomOfAffine φ hφ).app (op U.1) x = φ U x

theorem **homOfAffine_app** (U : X.affineOpens) (x : Γ(G, U.1)) : (homOfAffine φ hφ).app U.1 x = φ U x

theorem **hom_ext_affine** {a b : G ⟶ F} (h : ∀ U : X.Opens, IsAffineOpen U → ∀ x, a.app U x = b.app U x) : a = b

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.TcoMem

theorem **tco_mem** (j : Fin d) : algebraMap F K (x j.succ / x 0) ∈ rB k K x {0}

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ThetaSeriesDataHypothesesBelowMaximalLevel

theorem **dGT_zero** (D : D7) (u : D.what₂.Word) (p : ℕ) : dGT D u p 0 = 0

theorem **dGT_spec** {D : D7} {c : LatCmp D} {Tw : TwForms D} {u : D.what₂.Word} {μ : D.what₂.Tup} {w : D.what₂.places.Pl} {h : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)} [Fact (D.what₂.places.resChar w).Prime] {Fl : Fills (D.gat u w h)} {Lv : Levels Fl} {Ch : Chain Lv} (KA : TwTie.KerAtMu D c Tw u μ w h Ch) : ∀ j ≤ Fl.mm, ∀ (a : Lv.comp j) (L : (Fl.Pj j).Tup), (Fl.Pj j).IsLatTup L → padicValNat (D.what₂.places.resChar w) (Nat.card {γ : (Lv.lev j a).Γ // (Lv.lev j a).smul γ L = L}) ≤ dGT D u (D.what₂.places.resChar w) j

theorem **fracAt** {D : D7} (Gl : G.LatInst D) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : IsFractionRing (D.gat u w hg).O (D.gat u w hg).Kf

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ThetaZJetsThree

theorem **selfDual** (H : LatAt a₀ b₀ c₀ dK ha hdK lam hlam h Λ θ p ι) : IsSelfDual (matH h) Λ

theorem **stable** (H : LatAt a₀ b₀ c₀ dK ha hdK lam hlam h Λ θ p ι) : ∀ x ∈ Λ, Formalism.L2.Th1.F.cmTheta b₀ dK • x ∈ Λ

theorem **mem_th** (H : LatAt a₀ b₀ c₀ dK ha hdK lam hlam h Λ θ p ι) : θ ∈ Th (matH h) (alphaQ (Real.sqrt dK / 2) (matH h)) Λ

theorem **locNorm** (H : LatAt a₀ b₀ c₀ dK ha hdK lam hlam h Λ θ p ι) : LocNorm a₀ b₀ dK ha hdK lam hlam h Λ θ p ι

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.ToPtIdPtBijective

theorem **algebraMap_resAlg** (σ : k →+* k') (B : CommAlgCat.{0} k') : algebraMap k (resAlg σ B) = (algebraMap k' B).comp σ

theorem **resHom_apply** (σ : k →+* k') {B B' : CommAlgCat.{0} k'} (f : B ⟶ B') (b : B) : (resHom σ f).hom b = f.hom b

theorem **spMap_comp'** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap (g.comp f) = spMap g ≫ spMap f

theorem **spMap_id'** (A : Type) [CommRing A] : spMap (RingHom.id A) = 𝟙 _

theorem **eq_of_spMap_eq** {A B : Type} [CommRing A] [CommRing B] {f g : A →+* B} (h : spMap f = spMap g) : f = g

@[ext] theorem **hom_ext** {T T' : AffPt s} {φ ψ : T ⟶ T'} (h : φ.1 = ψ.1) : φ = ψ

theorem **pull_ptCoef_congr** {x' : ∀ T : C, Sp (F.obj T) ⟶ X} (hx' : ∀ T, x' T ≫ s = spMap ((algebraMap k' (F.obj T)).comp σ)) (hc' : ∀ ⦃T T' : C⦄ (a : T ⟶ T'), spMap (mp F a : F.obj T →+* F.obj T') ≫ x' T = x' T') (e : ∀ T, x T = x' T) (n : ℕ) : Hyp.pull (ptCoef x hx hc) n = Hyp.pull (ptCoef x' hx' hc') n

theorem **comp_w** {φ : X' ⟶ X} (h : φ ≫ s = s' ≫ spMap σ) (T : AffPt s') : (T.x ≫ φ) ≫ s = spMap ((algebraMap k' T.B).comp σ)

theorem **over_id** {s' : X' ⟶ Sp k} {f : X' ⟶ X} (hf : f ≫ s = s') : f ≫ s = s' ≫ spMap (RingHom.id k)

theorem **ptCoef_self** : ptCoef (s

theorem **ptCoef_comp** {φ : X' ⟶ X} {ψ : X'' ⟶ X'} (h : φ ≫ s = s' ≫ spMap σ) (h' : ψ ≫ s' = s'' ≫ spMap τ) (hx : ∀ T : AffPt s'', ((T.x ≫ ψ) ≫ φ) ≫ s = spMap ((algebraMap k'' T.B).comp (τ.comp σ))) (hc : ∀ ⦃T T' : AffPt s''⦄ (a : T ⟶ T'), spMap (mp (OPt s'') a : (OPt s'').obj T →+* (OPt s'').obj T') ≫ (T.x ≫ ψ) ≫ φ = (T'.x ≫ ψ) ≫ φ) : ptCoef (F

theorem **HdR.pull_id** (n : ℕ) (c : HdR s n) : HdR.pull (f

theorem **HdR.pullSemi_comp** {φ : X' ⟶ X} {ψ : X'' ⟶ X'} (h : φ ≫ s = s' ≫ spMap σ) (h' : ψ ≫ s' = s'' ≫ spMap τ) (h'' : (ψ ≫ φ) ≫ s = s'' ≫ spMap (τ.comp σ)) (n : ℕ) (c : HdR s n) : HdR.pullSemi h'' n c = HdR.pullSemi h' n (HdR.pullSemi h n c)

theorem **HdR.pull_comp** {s' : X' ⟶ Sp k} {s'' : X'' ⟶ Sp k} {f : X' ⟶ X} {g : X'' ⟶ X'} (hf : f ≫ s = s') (hg : g ≫ s' = s'') (n : ℕ) (c : HdR s n) : HdR.pull (f

theorem **HdR.pullSemi_mem_fil** {φ : X' ⟶ X} (h : φ ≫ s = s' ≫ spMap σ) {p n : ℕ} {c : HdR s n} (hc : c ∈ HdR.fil s p n) : HdR.pullSemi h n c ∈ HdR.fil s' p n

theorem **HdR.pull_mem_fil** {s' : X' ⟶ Sp k} {f : X' ⟶ X} (hf : f ≫ s = s') {p n : ℕ} {c : HdR s n} (hc : c ∈ HdR.fil s p n) : HdR.pull hf n c ∈ HdR.fil s' p n

theorem **HdRF.toHdR_pullSemi** {φ : X' ⟶ X} (h : φ ≫ s = s' ≫ spMap σ) {p n : ℕ} (c : HdRF s p n) : HdRF.toHdR s' p n (HdRF.pullSemi h p n c) = HdR.pullSemi h n (HdRF.toHdR s p n c)

theorem **HdR.toAff_mem_fil** {p n : ℕ} {c : HdR s n} (hc : c ∈ HdR.fil s p n) : HdR.toAff s n c ∈ Hyp.fil (OAff s) p n

theorem **HdR.fil_zero** (n : ℕ) : HdR.fil s 0 n = ⊤

theorem **HdR.fil_succ** (n : ℕ) : HdR.fil s (n + 1) n = ⊥

theorem **HdR.fil_anti** {p p' : ℕ} (h : p ≤ p') (n : ℕ) : HdR.fil s p' n ≤ HdR.fil s p n

theorem **spMap_toPt** (T : AffPt (algStr k B)) : spMap (toPt T : B →+* T.B) = T.x

theorem **toPt_comp** ⦃T T' : AffPt (algStr k B)⦄ (φ : T ⟶ T') (b : B) : mp (OPt (algStr k B)) φ (toPt T b) = toPt T' b

theorem **toPt_idPt** (b : B) : toPt (idPt k B) b = b

theorem **toPt_idPt_bijective** : Function.Bijective (toPt (idPt k B))

theorem **HdR.ofForm_mem_fil** {n : ℕ} (ω : LinearMap.ker (dOm k B n)) : HdR.ofForm n ω ∈ HdR.fil (algStr k B) n n

theorem **HdR.ofForm_eq_zero_iff** {n : ℕ} (ω : LinearMap.ker (dOm k B (n + 1))) : HdR.ofForm (n + 1) ω = 0 ↔ ∃ η, dOm k B n η = ω.1

theorem **dOm_zero_self** (x : Om k k 0) : dOm k k 0 x = 0

theorem **HdR.ofForm_tdt_ne_zero** [Nontrivial k] (h2 : (2 : k) = 0) : HdR.ofForm 1 (tdtForm k) ≠ 0

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.TrueWtAddSSNonneg

theorem **jetLin_monomial** (J : (DVar ι a b →₀ ℕ) → K) (μ : ((ι × Fin b × Fin b) ⊕ DVar ι a b) →₀ ℕ) (c : K) : jetLin J (monomial μ c) = monomial (μ.comapDomain Sum.inl Sum.inl_injective.injOn) (c * J (μ.comapDomain Sum.inr Sum.inr_injective.injOn))

theorem **jetMap_sum** {α : Type} (s : Finset α) (r : α → K) (J : α → (DVar ι a b →₀ ℕ) → K) (n : ι → ℕ) : jetMap (∑ x ∈ s, r x • J x) n = ∑ x ∈ s, r x • jetMap (J x) n

theorem **map_subD** {L : Type} [Field L] (s : K →+* L) (P : Formalism.L2.DIV.Pol K ι a) : MvPolynomial.map s (subD K ι a b P) = subD L ι a b (MvPolynomial.map s P)

theorem **map_jetMap** {L : Type} [Field L] (s : K →+* L) (J : (DVar ι a b →₀ ℕ) → K) (J' : (DVar ι a b →₀ ℕ) → L) (r : L) (n : ι → ℕ) (P : Formalism.L2.DIV.Pol K ι a) (h : ∀ μ ∈ (subD K ι a b (P * ∏ τ, frDet K a τ ^ n τ)).support, s (J (μ.comapDomain Sum.inr Sum.inr_injective.injOn)) = r * J' (μ.comapDomain Sum.inr Sum.inr_injective.injOn)) : MvPolynomial.map s (jetMap J n P) = r • jetMap J' 0 (MvPolynomial.map s (P * ∏ τ, frDet K a τ ^ n τ))

theorem **eval_jetMap** (cD : ι → K) (T : (DVar ι a b →₀ ℕ) → K) (Q : Formalism.L2.DIV.Pol K ι a) (y' : ι → Matrix (Fin b) (Fin b) K) : eval (fun v : ι × Fin b × Fin b => y' v.1 v.2.1 v.2.2) (jetMap (fun q => (q.prod fun k n => cD k.1 ^ n) * T q) 0 Q) = ∑ q ∈ (derPol cD Q y').support, coeff q (derPol cD Q y') * T q

theorem **eval_jetMap_jetC** (D : D7) {a b : ℕ} (cD : D.what₂.Tau → ℂ) (G : Formalism.MainNotions.TaylorLogMetric.VC D a b → ℂ) (Q : MvPolynomial (Formalism.MainNotions.GL3Theta.FrV D a) ℂ) (y' : D.what₂.Tau → Matrix (Fin b) (Fin b) ℂ) : eval (fun v : D.what₂.Tau × Fin b × Fin b => y' v.1 v.2.1 v.2.2) (jetMap (jetC cD G) 0 Q) = Formalism.MainNotions.GL3Theta.applyAt0 D (Formalism.MainNotions.GL3Theta.polD D cD Q y') G

theorem **half** (R : Ranks) (j : ℕ) : ∃ z : ℤ, (z : ℚ) = (R.N j : ℚ) / 2

theorem **zOf_spec** (R : Ranks) (e : ℚ) (he : ∃ z : ℤ, e = z) (j : ℕ) : ((zOf (e - (R.N j : ℚ) / 2) : ℤ) : ℚ) = e - (R.N j : ℚ) / 2

theorem **evS_apply** (i : ℕ) (x : (A.F i).V) : A.evS σ i x = A.ev σ i x

theorem **ev_injective** (i : ℕ) : Function.Injective (A.ev σ i)

theorem **ev_sum** (i : ℕ) {J : Type} [Fintype J] (a : J → X.K) (v : J → (A.F i).V) : A.ev σ i (∑ j, a j • v j) = ∑ j, σ (a j) • A.ev σ i (v j)

theorem **emb_injective** (i : ℕ) : Function.Injective (A.emb σ i)

theorem **exists_ext** (i : ℕ) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) : ∃ U : (Fr X σ (R.N i) → ℂ) →ₗ[ℂ] (Fr X σ (R.N (i + 1)) → ℂ), ∀ x, U (A.ev σ i x) = A.ev σ (i + 1) (A.ker i c c' x)

theorem **kerC_ev** (i : ℕ) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (x : (A.F i).V) : A.kerC σ i c c' (A.ev σ i x) = A.ev σ (i + 1) (A.ker i c c' x)

theorem **prod_de_ne_zero** {n : ℕ} (y : Fr X σ n) (e : X.Tau → ℤ) : (∏ τ, deFr y τ ^ e τ) ≠ 0

theorem **Kmap_eq** (hc : X.Tau → HCParam R) (i : ℕ) (hi : i < R.m) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) : (A.chain σ hc).Kmap i c c' = A.kerC σ i c c'

theorem **Kmap_ev** (hc : X.Tau → HCParam R) (i : ℕ) (hi : i < R.m) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (x : (A.F i).V) : (A.chain σ hc).Kmap i c c' (A.ev σ i x) = A.ev σ (i + 1) (A.ker i c c' x)

theorem **unf_sharp_ev** (hc : X.Tau → HCParam R) (i : ℕ) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (x : (A.F i).V) (y : Fr X σ (R.N (i + 1))) : (A.chain σ hc).unf i c c' ((A.chain σ hc).sharp i (A.ev σ i x)) y = (A.rho i c * A.rho' i c')⁻¹ * ((∏ τ, deFr y τ ^ sD hc i τ) * A.ev σ (i + 1) (A.ker i c c' x) y)

theorem **K_emb** (hc : X.Tau → HCParam R) (i : ℕ) (hi : i < R.m) (φ : (A.F i).Forms) : (A.chain σ hc).K i (A.emb σ i φ) = A.emb σ (i + 1) (A.kk.step A.ok i φ)

theorem **chain_emb** (hc : X.Tau → HCParam R) (φ : (A.F 0).Forms) (j : ℕ) (hj : j ≤ R.m) : Formalism.ThetaLiftIndependence.chainOpM (A.chain σ hc).wt (A.chain σ hc).Kmap j (A.emb σ 0 φ) = A.emb σ j (A.kk.colTo A.ok j φ)

theorem **col_emb** (hc : X.Tau → HCParam R) (φ : (A.F 0).Forms) : (A.chain σ hc).col (A.emb σ 0 φ) = A.emb σ R.m (A.kk.colTo A.ok R.m φ)

theorem **trueWt_eq** {Word : Type} (W : Wts X Word R) (u : Word) (i : ℕ) : trueWt (W.hc u) i = W.lamTrue u i

theorem **trueWt_old** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (τ : X.Tau) (a : Fin (R.N (i + 1))) (h : a.1 < R.N i) : (trueWt hc (i + 1) τ).w a = (trueWt hc i τ).w (a.1, h) + sS hc i τ - sD hc i τ

theorem **trueWt_new** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (τ : X.Tau) (a : Fin (R.N (i + 1))) (h : ¬ a.1 < R.N i) : (trueWt hc (i + 1) τ).w a = -sD hc i τ

theorem **trueWt_add_sS_nonneg** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (τ : X.Tau) (a : Fin (R.N i)) : 0 ≤ (trueWt hc i τ).w a + sS hc i τ

theorem **Tied.sharp_ev_poly** {hc : X.Tau → HCParam R} (h : A.Tied hc) (i : ℕ) (hi : i < R.m) (x : (A.F i).V) : ∃ P : MvPolynomial (X.Tau × Fin (R.N i) × Fin (R.N i)) ℂ, ∀ y : Fr X σ (R.N i), (A.chain σ hc).sharp i (A.ev σ i x) y = MvPolynomial.eval (fun v => y.1 v.1 v.2.1 v.2.2) P

theorem **Tied.ev_rho** {hc : X.Tau → HCParam R} (h : A.Tied hc) (i : ℕ) (hi : i ≤ R.m) (b : (A.F i).β) : ∃ (u : (A.F i).G b →* UF X.E (R.N i)) (χ : (A.F i).G b →* X.Kˣ), ∀ g x (y y' : Fr X σ (R.N i)), IsTr (atInf (embAt σ X.emb) (ufGL X.E (R.N i) (u g) : Matrix (Fin (R.N i)) (Fin (R.N i)) X.E)) y' y → A.ev σ i ((A.F i).ρ b g x) y = σ (χ g : X.Kˣ) * A.ev σ i x y'

theorem **exampleDatum_isAdmissible** (X : Fields) {n : ℕ} (lam : X.Tau → DomWeight n) : (exampleGSD X lam).Ok where

theorem **card_stabilizer_units_int** (d : Unit) : Nat.card (MulAction.stabilizer ℤˣ d) = 2

theorem **exists_linearForm_apply_eq_one** (X : Fields) {n : ℕ} (lam : X.Tau → DomWeight n) : ∃ f : VV X.K lam →ₗ[X.K] X.K, f (extV X.K lam) = 1

theorem **exampleChain_compatible** (hc : X.Tau → HCParam R) : (toyExample hc).Tied hc where

theorem **exampleExt_emb_ne_zero** (hc : X.Tau → HCParam R) (i : ℕ) : (toyExample hc).emb σ i (exampleExt hc i) ≠ 0

theorem **exampleChain_K_apply** (hc : X.Tau → HCParam R) (i : ℕ) (hi : i < R.m) (a : ℂ) : ((toyExample hc).chain σ hc).K i (a • (toyExample hc).emb σ i (exampleExt hc i)) = (a / 2) • (toyExample hc).emb σ (i + 1) (exampleExt hc (i + 1))

theorem **exampleChain_col_apply** (hc : X.Tau → HCParam R) : ((toyExample hc).chain σ hc).col ((toyExample hc).emb σ 0 (exampleExt hc 0)) = ((1 / 2 : ℂ) ^ R.m) • (toyExample hc).emb σ R.m (exampleExt hc R.m)

theorem **exampleChain_col_ne_zero** (hc : X.Tau → HCParam R) : ((toyExample hc).chain σ hc).col ((toyExample hc).emb σ 0 (exampleExt hc 0)) ≠ 0

theorem **unfolds_chain** (hc : X.Tau → HCParam R) : A.Unfolds σ hc (A.chain σ hc).unf

theorem **chainWith_self** (hc : X.Tau → HCParam R) : A.chainWith σ hc (A.chain σ hc).unf = A.chain σ hc

theorem **chainWith_unf** (hc : X.Tau → HCParam R) (U : A.UnfFam σ) : (A.chainWith σ hc U).unf = U

theorem **Unfolds.Kmap_ev** {hc : X.Tau → HCParam R} {U : A.UnfFam σ} (h : A.Unfolds σ hc U) (i : ℕ) (hi : i < R.m) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (x : (A.F i).V) : (A.chainWith σ hc U).Kmap i c c' (A.ev σ i x) = A.ev σ (i + 1) (A.ker i c c' x)

theorem **unfolds_iff_Kmap** (hc : X.Tau → HCParam R) (U : A.UnfFam σ) : A.Unfolds σ hc U ↔ ∀ i, i < R.m → ∀ c c' x, (A.chainWith σ hc U).Kmap i c c' (A.ev σ i x) = A.ev σ (i + 1) (A.ker i c c' x)

theorem **Unfolds.K_emb** {hc : X.Tau → HCParam R} {U : A.UnfFam σ} (h : A.Unfolds σ hc U) (i : ℕ) (hi : i < R.m) (φ : (A.F i).Forms) : (A.chainWith σ hc U).K i (A.emb σ i φ) = A.emb σ (i + 1) (A.kk.step A.ok i φ)

theorem **Unfolds.chain_emb** {hc : X.Tau → HCParam R} {U : A.UnfFam σ} (h : A.Unfolds σ hc U) (φ : (A.F 0).Forms) (j : ℕ) (hj : j ≤ R.m) : Formalism.ThetaLiftIndependence.chainOpM (A.chainWith σ hc U).wt (A.chainWith σ hc U).Kmap j (A.emb σ 0 φ) = A.emb σ j (A.kk.colTo A.ok j φ)

theorem **Unfolds.col_emb** {hc : X.Tau → HCParam R} {U : A.UnfFam σ} (h : A.Unfolds σ hc U) (φ : (A.F 0).Forms) : (A.chainWith σ hc U).col (A.emb σ 0 φ) = A.emb σ R.m (A.kk.colTo A.ok R.m φ)

theorem **polFn_apply** {n : ℕ} (Q : Formalism.L2.DIV.Pol ℂ X.Tau n) (y : Fr X σ n) : polFn X σ n Q y = MvPolynomial.eval (fun v => y.1 v.1 v.2.1 v.2.2) Q

theorem **polFn_injective** (n : ℕ) : Function.Injective (polFn X σ n)

theorem **fnPol_polFn** {n : ℕ} (Q : Formalism.L2.DIV.Pol ℂ X.Tau n) : fnPol X σ n (polFn X σ n Q) = Q

theorem **chainJet_unf** (hc : X.Tau → HCParam R) (J : A.JetFam) (i : ℕ) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (Q : Formalism.L2.DIV.Pol ℂ X.Tau (R.N i)) (y' : Fr X σ (R.N (i + 1))) : (A.chainJet σ hc J).unf i c c' (fun y : Fr X σ (R.N i) => MvPolynomial.eval (fun v => y.1 v.1 v.2.1 v.2.2) Q) y' = MvPolynomial.eval (fun v => y'.1 v.1 v.2.1 v.2.2) (J i c c' Q)

theorem **nuEnd_cast** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (τ : X.Tau) : (nuEnd hc i τ : ℤ) = (trueWt hc i τ).detShift + sS hc i τ

theorem **nuEnd_eq** {Pr : Primes X} (W : Wts X Pr.Word R) (u : Pr.Word) {i : ℕ} (hi : i < R.m) : nuEnd (W.hc u) i = Formalism.L2.Ker.nuLast W u i

theorem **trueWt_detShift_succ** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (τ : X.Tau) : (trueWt hc (i + 1) τ).detShift = -sD hc i τ

theorem **Tied.sharp_ev** {hc : X.Tau → HCParam R} (h : A.Tied hc) (i : ℕ) (hi : i < R.m) (x : (A.F i).V) : (A.chain σ hc).sharp i (A.ev σ i x) = polFn X σ _ (A.sharpPol σ hc i x)

theorem **Tied.de_mul_ev** {hc : X.Tau → HCParam R} (h : A.Tied hc) (i : ℕ) (hi : i < R.m) (x : (A.F (i + 1)).V) (y : Fr X σ (R.N (i + 1))) : (∏ τ, deFr y τ ^ sD hc i τ) * A.ev σ (i + 1) x y = polFn X σ _ (MvPolynomial.map σ (A.eV (i + 1) x).1) y

theorem **Tied.unfolds_iff** {hc : X.Tau → HCParam R} (h : A.Tied hc) (J : A.JetFam) : A.Unfolds σ hc (A.unfJet σ J) ↔ A.JetTie σ hc J

theorem **Tied.exists_jetTie** {hc : X.Tau → HCParam R} (h : A.Tied hc) : ∃ J : A.JetFam, A.JetTie σ hc J

theorem **chainTh_unf** (hc : X.Tau → HCParam R) (cD : X.Tau → ℂ) (Th : A.ThFam) (i : ℕ) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (Q : Formalism.L2.DIV.Pol ℂ X.Tau (R.N i)) (y' : Fr X σ (R.N (i + 1))) : (A.chainTh σ hc cD Th).unf i c c' (fun y : Fr X σ (R.N i) => MvPolynomial.eval (fun v => y.1 v.1 v.2.1 v.2.2) Q) y' = ∑ q ∈ (derPol cD Q y'.1).support, MvPolynomial.coeff q (derPol cD Q y'.1) * Formalism.L2.Ker.derivAt0 (Th i c c') q

theorem **jetTie_of_tables** (hc : X.Tau → HCParam R) (cD : X.Tau → ℂ) (Th : A.ThFam) (h : ∀ i, i < R.m → ∀ c c', ∃ T : (Formalism.L2.Ker.DVar X.Tau (R.N i) (R.N (i + 1)) →₀ ℕ) → X.K, ∀ x, (A.eV (i + 1) (A.ker i c c' x)).1 = Formalism.L2.Ker.jetMap T (nuEnd hc i) (A.eV i x).1 ∧ ∀ μ ∈ (Formalism.L2.Ker.subD X.K X.Tau (R.N i) (R.N (i + 1)) ((A.eV i x).1 * ∏ τ, Formalism.L2.Ker.frDet X.K (R.N i) τ ^ nuEnd hc i τ)).support, σ (T (μ.comapDomain Sum.inr Sum.inr_injective.injOn)) = A.rho i c * A.rho' i c' * Formalism.L2.Ker.jetC cD (Th i c c') (μ.comapDomain Sum.inr Sum.inr_injective.injOn)) : A.JetTie σ hc (A.jetTh cD Th)

theorem **Tied.chainTh_Kmap_ev** (i : ℕ) (hi : i < R.m) (c : (A.F i).Cl) (c' : (A.F (i + 1)).Cl) (x : (A.F i).V) : (A.chainTh σ hc cD Th).Kmap i c c' (A.ev σ i x) = A.ev σ (i + 1) (A.ker i c c' x)

theorem **Tied.chainTh_K_emb** (i : ℕ) (hi : i < R.m) (φ : (A.F i).Forms) : (A.chainTh σ hc cD Th).K i (A.emb σ i φ) = A.emb σ (i + 1) (A.kk.step A.ok i φ)

theorem **Tied.chainTh_col_emb** (φ : (A.F 0).Forms) : (A.chainTh σ hc cD Th).col (A.emb σ 0 φ) = A.emb σ R.m (A.kk.colTo A.ok R.m φ)

theorem **ChSrc.liftTie_chainTh** (D : D7) (T : Formalism.MainNotions.TaylorLogMetric.Tor D) (e : Formalism.MainNotions.TaylorLogMetric.FamE D) (X : Fields) (emb : D.what₂.Tau → (X.E →+* X.K)) (A : ChSrc (fieldsOn D X emb) D.R) (σ : X.K →+* ℂ) (u : D.what₂.Word) (cD : D.what₂.Tau → ℂ) (Th : A.ThFam) (O : Formalism.MainNotions.GL3Theta.LiftObj D (A.chainTh σ (D.hcW u) cD Th)) (i : ℕ) (hf : ∀ y, O.frm i y = y.1) (hf' : ∀ y, O.frm (i + 1) y = y.1) (hd : O.dgen = cD) (hT : ∀ c c', O.Th i c c' = Th i c

c') (hK : ∀ c c', Formalism.MainNotions.GL3Theta.IsKer D T e O.S (O.lat i c).1 (O.lat (i + 1) c').1 (O.Th i c c')) : Formalism.MainNotions.GL3Theta.liftTie D T e O i

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.TwChainWtSharpDivDe

theorem **twDen_pos** : 0 < cAof sel mu cst

theorem **twMod_pos** (hT : 0 < Pr.T u) : 0 < cBof sel mu cst u

theorem **mem_divLat_of_loc** {i j : ℕ} (p : ChPt (R

theorem **loc_of_mem_mulLat** (hPr : Pr.Ok) {i j : ℕ} (p : ChPt (R

theorem **locTop_den** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} {y : HomE X.E (R.N (m2 + 1)) (R.N R.m)} {y' : Sp X.E (R.N (m2 + 1))} (h : locTop Ch sel mu cst m2 P y y' ≠ 0) (s : Fin (R.N (m2 + 1))) (l : Fin (R.N R.m)) : IntAt P.1 ((qOf P.1 : X.E) ^ cst.A * y (Pi.single s 1) l)

theorem **locTop_add** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} (hD : P.1 ∉ Pr.vGs → 2 * cst.M ≤ cst.D P) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) {z : HomE X.E (R.N (m2 + 1)) (R.N R.m)} {y' : Sp X.E (R.N (m2 + 1))} (hz : ∀ s l, IntAt P.1 (((qOf P.1 : X.E) ^ cst.B)⁻¹ * z (Pi.single s 1) l)) (hy' : ∀ i, IntAt P.1 ((qOf P.1 : X.E) ^ (if P.1 ∈ Pr.vGs then 1 else cst.M) * y' i)) : locTop Ch sel mu cst m2 P (y + z) y' = locTop Ch sel mu cst m2 P y y'

theorem **twThPt_top** (e : R.m = m2 + 1 + 1) (h : m2 + 1 = m2 + 1) (h' : R.m = R.m) (d : (ybCpl W sel mu cst L u m2 b hY).Pts) (p' : TopPt R sel mu u) : twThPt (Ch

theorem **twChain_unf** (i : ℕ) (x : (twFam W Ch sel mu cst L u m2 b hY P c i).Cl) (x' : (twFam W Ch sel mu cst L u m2 b hY P c (i + 1)).Cl) (Q : Formalism.L2.DIV.Pol ℂ X.Tau (R.N i)) (y' : ChSrc.Fr X A.σ₀ (R.N (i + 1))) : (twChain chi hX hm hF hK hρ).unf i x x' (fun y : ChSrc.Fr X A.σ₀ (R.N i) => MvPolynomial.eval (fun v => y.1 v.1 v.2.1 v.2.2) Q) y' = ∑ q ∈ (derPol A.cD Q y'.1).support, MvPolynomial.coeff q (derPol A.cD Q y'.1) * derivAt0 (twTh chi hX hm hF hK hρ i x x') q

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.TwistEmptyOfChartsEx

theorem **exists_char** {G : Type} [CommGroup G] (H : Subgroup G) (g : G) (hg : g ∉ H) : ∃ η : G →* ℂˣ, (∀ h ∈ H, η h = 1) ∧ η g ≠ 1

theorem **exists_irr** (E : Type) [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) (M N : ℕ) (hM : M ≠ 0) (hN : 0 < N) : ∃ a : P.adicCompletion E, Valued.v (a − 1) ≤ Valued.v (M : P.adicCompletion E) ∧ Valued.v a = 1 ∧ ∀ e : E, a ^ N ≠ algebraMap E _ e

theorem **exists_wild** (P : HeightOneSpectrum (𝒪 E)) (hP : conjPrime₂ P ≠ P) (hu : ∀ w : HeightOneSpectrum (𝒪 E), conjPrime₂ w = P → w.under (𝒪 F) = P.under (𝒪 F)) (N : ℕ) (hN : 0 < N) (M : ℕ) (hM : M ≠ 0) : ∃ (s : (AfE F E)ˣ) (η : (P.adicCompletion E)ˣ →* ℂˣ), sigR hc.sig _ (s : AfE F E) * s = 1 ∧ (∀ v, v ≠ P.under (𝒪 F) → prV (E

theorem **modulus_ne** : modulus t μ ≠ 0

theorem **chi_central** (H : AdTie A dtU a) (h0 : 0 < D.R.m) (β : (St 0 h0.le).Ch) (v₀ : HeightOneSpectrum (𝒪 D.host.F)) (hv : v₀ ∈ (St 0 h0.le).bad) (s : (AfE D.host.F D.host.E)ˣ) (hs : sigR (sigD D) _ (s : AfE D.host.F D.host.E) * s = 1) (hoff : ∀ v, v ≠ v₀ → prV (E

theorem **detAt_eq_one** (P : HeightOneSpectrum (𝒪 E)) (η : (P.adicCompletion E)ˣ →* ℂˣ) (x : GL (Fin n) (AfE F E)) (h : bc (prV (E

theorem **detAt_pt** [StarRing E] {hc : CMData F E} {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} {P : HeightOneSpectrum (𝒪 E)} (hP : P ∈ Q) {η : (P.adicCompletion E)ˣ →* ℂˣ} (hη : ∀ y : (P.adicCompletion E)ˣ, (∃ e : E, (y : P.adicCompletion E) = algebraMap E _ e) → η y = 1) {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) : detAt F n P η g = 1

theorem **detAt_mid** [StarRing E] {hc : CMData F E} {ST SG Q Q' : Finset (HeightOneSpectrum (𝒪 E))} {P : HeightOneSpectrum (𝒪 E)} (hP : P ∈ Q) (hP' : P ∈ Q') {η : (P.adicCompletion E)ˣ →* ℂˣ} (hη : ∀ y : (P.adicCompletion E)ˣ, (∃ e : E, (y : P.adicCompletion E) = algebraMap E _ e) → η y = 1) {y x k : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} {p' : DecoF E n ST SG Q'} (h : IsPtOf hc (y * x * k) p) (h' : IsPtOf hc (y * k) p') : detAt F n P η x = 1

theorem **move_twist_central** (S : StepLevel F E σ J T f bad) (P : HeightOneSpectrum (𝒪 E)) (η : (P.adicCompletion E)ˣ →* ℂˣ) (k : basePts (twistLev S P η)) (s : (AfE F E)ˣ) (hm : Matrix.GeneralLinearGroup.scalar (Fin n) s ∈ (moveLev k).U) (e : (moveLev k).chi ⟨_, hm⟩ = 1) : ∃ hm' : Matrix.GeneralLinearGroup.scalar (Fin n) s ∈ S.U, S.chi ⟨_, hm'⟩ * η (Units.map (adAt F P).toMonoidHom (s ^ n)) = 1

theorem **readsK_twist** {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i ≤ B.R.m) {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {Ch : Type} (chOf : (I.stp u a i).Cl → Ch) (K : Fam I l Z u a i hi bad chOf) {P : HeightOneSpectrum (𝒪 B.host.E)} (hP : P ∈ I.Q a) {η : (P.adicCompletion B.host.E)ˣ →* ℂˣ} (hη : ∀ y : (P.adicCompletion B.host.E)ˣ, (∃ e : B.host.E, (y : P.adicCompletion B.host.E) = algebraMap B.host.E _ e) → η y = 1) (h : I.ReadsK l Z u a i hi chOf K) : I.ReadsK l Z u a i hi chOf fun β => twistChart (D

theorem **twist_empty_of_tieMoved** (h : TieOnAllReadings) {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (L : SizeData₂ (I.rec7 l Z)) (hT : GatTies (I.rec7 l Z)) (Gi : G.Inst (I.rec7 l Z) L) (Gl : G.LatInst (I.rec7 l Z)) (Kc : G.Consts) (Y : DivX (I.rec7 l Z)) (cX : ExtraC) (ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Y cX) (t : KerEntTied ex) (dt : Formalism.L2.Herm.unitaryOneFiniteAdeles (I.rec7 l Z).sp.cm →* (I.rec7 l Z).Dt) (hS : StdData I l Z t dt) (u : B.Word) (a : I.ι) (bad Ch chOf eY K) (hs : SideNoChar I l Z u a bad Ch chOf eY K) (hK : ∀ i hi, I.ReadsK l Z u a i hi (chOf i hi) (K i hi)) : I.Q a = ∅

theorem **twist_empty_of_chartsEx** (h : TieOnAllReadings) (h0 : ReadingsEx) {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (L : SizeData₂ (I.rec7 l Z)) (hT : GatTies (I.rec7 l Z)) (Gi : G.Inst (I.rec7 l Z) L) (Gl : G.LatInst (I.rec7 l Z)) (Kc : G.Consts) (Y : DivX (I.rec7 l Z)) (cX : ExtraC) (ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Y cX) (t : KerEntTied ex) (dt : Formalism.L2.Herm.unitaryOneFiniteAdeles (I.rec7 l Z).sp.cm →* (I.rec7 l Z).Dt) (hS : StdData I l Z t dt) (u : B.Word) (a : I.ι) : I.Q a = ∅

theorem **not_stdData_of_tieMoved** (h : TieOnAllReadings) (h0 : ReadingsEx) {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (L : SizeData₂ (I.rec7 l Z)) (hT : GatTies (I.rec7 l Z)) (Gi : G.Inst (I.rec7 l Z) L) (Gl : G.LatInst (I.rec7 l Z)) (Kc : G.Consts) (Y : DivX (I.rec7 l Z)) (cX : ExtraC) (ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Y cX) (t : KerEntTied ex) (dt : Formalism.L2.Herm.unitaryOneFiniteAdeles (I.rec7 l Z).sp.cm →* (I.rec7 l Z).Dt) : ¬ StdData I l Z t dt

theorem **exists_pt_one** {x : (AfE D.host.F D.host.E)ˣ} (hx : x ∈ U1 (sigD D)) : ∃ y : Formalism.L2.Herm.unitaryOneFiniteAdeles D.sp.cm, Matrix.GeneralLinearGroup.det y.1 = x

theorem **exists_decomp** (i : ℕ) (hi : i ≤ D.R.m) (β : (St i hi).Ch) (x : UPts (sigD D) (gramN D i hi) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ (c : (D.chTw u μ).cls i) (_ : (St i hi).chOf c = β) (γ : GL (Fin (D.R.N i)) D.host.E) (_ : γ ∈ UJ (sigD D : D.host.E →+* D.host.E) (gramN D i hi)) (κ : ((St i hi).K β).lev.U), x.1 = ratA D.host.F γ * (((St i hi).y c).1 * κ.1)

theorem **muT_unique** {Xi Xi' : AdDatum St} {m m' : ℕ → (AfE D.host.F D.host.E)ˣ → ℂ} (H : AdLawAt A dtU a Xi m) (H' : AdLawAt A dtU a Xi' m') {i : ℕ} (hi : i < D.R.m) {x : (AfE D.host.F D.host.E)ˣ} (hx : x ∈ U1 (sigD D)) : m (i + 1) x = m' (i + 1) x

theorem **xiAd_unique** {Xi Xi' : AdDatum St} {m m' : ℕ → (AfE D.host.F D.host.E)ˣ → ℂ} (H : AdLawAt A dtU a Xi m) (H' : AdLawAt A dtU a Xi' m') : Xi = Xi'

theorem **chart_nonempty** (i : ℕ) (hi : i ≤ D.R.m) (β : (St i hi).Ch) : ∃ c, (St i hi).chOf c = β

theorem **not_topLev_neg** {X : Fields} {Pr : Primes X} {R : Ranks} (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S : Set ℕ} (mu : Tup X.F X.E X.K S) (cst : LocCst X (rowSG Pr sel mu)) (j : ℕ) (P : rowSG Pr sel mu) (z : X.K) : ¬ TopLev (R

theorem **intAtA_add** {a b : E ⊗[F] FiniteAdeleRing (𝒪 F) F} (ha : IntAtA (F

theorem **intAtA_mul** {a b : E ⊗[F] FiniteAdeleRing (𝒪 F) F} (ha : IntAtA (F

theorem **intAtA_int** (x : 𝒪 E) : IntAtA (F

theorem **intAtA_sum** {ι : Type} (s : Finset ι) (f : ι → E ⊗[F] FiniteAdeleRing (𝒪 F) F) (h : ∀ i ∈ s, IntAtA (F

theorem **congAt_mul** (n : ℕ) {k k' : GL (Fin N) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)} (h : CongAt P n k) (h' : CongAt P n k') : CongAt P n (k * k')

theorem **sameCoset_equiv** {X : Fields} {Pr : Primes X} (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S : Set ℕ} (mu : Tup X.F X.E X.K S) (cst : LocCst X (rowSG Pr sel mu)) {n : ℕ} (P : rowSG Pr sel mu) : Equivalence (fun y y' : Sp X.E n => SameCoset sel mu cst P y y') where

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.TwistedStartColumnReading

theorem **card_autU** (N : Submodule (𝒪 E) (Sp E n)) : Nat.card (autU N) = Nat.card (Mem.Col2.autO N)

theorem **card_autU_dvd_tilN** (N : Submodule (𝒪 E) (Sp E n)) : Nat.card (autU N) ∣ Nat.card (autU (tilN N))

theorem **exists_card_autU_dvd** {N : Submodule (𝒪 E) (Sp E n)} (hN : IsSdLat N) (k : ℕ) : ∃ N' : Submodule (𝒪 E) (Sp E (n + k)), IsSdLat N' ∧ Nat.card (autU N) ∣ Nat.card (autU N')

theorem **card_autU_dvd_DGam** (hE : PosStar E) (R : Ranks) (hm : 1 ≤ R.m) {i : ℕ} (hi : i ≤ R.m) {N : Submodule (𝒪 E) (Sp E (R.N i))} (hN : IsSdLat N) : Nat.card (autU N) ∣ Mem.Col2.DGam E R

theorem **RhoUnit.norm_rhoPt** (h : RhoUnit Ch sel mu cst σ chi) {i : ℕ} (hi : i < R.m) {u : Pr.Word} (p : ChPt (R

theorem **RhoUnit.norm_rhoPt'** (h : RhoUnit Ch sel mu cst σ chi) {i : ℕ} (hi : i < R.m) {u : Pr.Word} (p : ChPt (R

theorem **rhoUnit_of** {D : D7} {s : Seam D X} {χ : TaylorLogMetric.IdE D → ℂ} {L : ChIdx X Pr R (QE sel (mu : Chr X.F X.E X.K))} {G : ChGlob X R.m} {m2 : ℕ} {e : Fin R.m → X.Tau → ℚ} {eta : Formalism.L2.Host.HChar X.E} (hX : X.Ok) (hPr : Pr.Ok) (hsel : IsSel X sel) (hS : ∀ i, Pr.g i ∈ S) (ht : Ch.Tied s.σ₀ e G.mu0 eta) (hL : L.Ok Ch sel mu cst G m2) (hG : G.σ₀ = s.σ₀) (hs : s.Ok) (hχ : TaylorLogMetric.IsChi D χ) : RhoUnit Ch sel mu cst s.σ₀ (chiLat s χ)

theorem **TwEquiv.kerEquivC** (h : TwEquiv W Ch sel mu cst L A u m2 b hY P c) (hX : X.Ok) (hPr : Pr.Ok) (hm : R.m = m2 + 2) (hc : cst.Ok) (H : TilOk Pr sel mu u) : KerEquivC W Ch sel mu cst L A u m2 b hY P

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.UnitaryRealisationReadingOfIdempotentActsViaLeve

theorem **used_loc** (D : D7) (i : D.nd.Loc) : D.nd.mu i = D.what₂.mu0 ∨ ∃ w, D.nd.mu i = D.what₂.muAt w

theorem **lamPos_of_used** {D : D7} (h : LamUsed D) {μ} (hμ : μ = D.what₂.mu0 ∨ ∃ w, μ = D.what₂.muAt w) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : LamPos D μ j

theorem **lamLoc** {D : D7} (h : LamUsed D) (i : D.nd.Loc) : LamPos D (D.nd.mu i) (D.nd.j i)

theorem **up_injective** {D : D7} {μ} {j : ℕ} (h : LamPos D μ j) (hI : DirectSum.IsInternal (D.what₂.isot j)) : Function.Injective (D.what₂.thetaT μ j)

theorem **rank_of** {D : D7} (hI : SumRepresentations.adm8 D) (h : LamUsed D) (L : SizeData₂ D) : TwistedPlaces.adm3 L

theorem **not_lamUsed** {D : D7} (j : Fin D.R.m) (σ : D.what₂.FT) (hσ : D.what₂.isot (j.1 + 1) σ ≠ ⊥) (h0 : D.what₂.thetaP D.what₂.mu0 (j.1 + 1) = 0) : ¬ LamUsed D

theorem **mtPar_holds** (D : D7) : MtParTie D.nd D.naut D.ntypes

theorem **goodIdempotent_iff_rowProjector** (D : D7) : GoodIdempotentFixesLocalisationForms D ↔ ∀ (i : D.nd.Loc) (σ' : D.ntypes.Par), ∀ F ∈ D.ntypes.MtPar (D.nd.mu i) (D.nd.j i) σ', D.ntypes.goodPar σ' → D.what₂.eMt (D.nd.mu i) (D.nd.j i) F = F

theorem **goodIdempotent_of_goodProjector** {D : D7} (h : ArithmeticCylinders.Conclusion4 D) : GoodIdempotentFixesLocalisationForms D

theorem **partsGive_of_stmt** {D : D7} (h : SumRepresentations.IntegralityFromMetric D) : PartsGive D

theorem **baseChange_preserves_unitary_of** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} (h : PartsGive D) (I : Inputs hL D) (a : RowsA D) (b : RowsB D) : SumRepresentations.Conclusion1 D

theorem **an_of** {D : D7} (h : SumRepresentations.adm5 D) : IsobaricCuspidalStrongMultiplicityOne D.nd D.naut ∧ IsobaricDiscreteSelfDualStrongMultiplicityOne D.nd D.naut

theorem **exists_vec** {μ} (hμ : Formalism.IndefR.RecTie.Used0 D μ) (j : Fin D.R.m) (f : D.what₂.Mt μ (j.1 + 1)) : ∃ y, X.R.extV j (X.tupT μ) y = X.fn hμ j f

theorem **goodVt_zero** (j : Fin D.R.m) (t : X.T) : GlobalTheory.DownLift.GoodVt X.R j t 0

theorem **good_of_fixed** (h : GoodIdempotentReadsAsGoodTypeProjector X) {μ} (hμ : Formalism.IndefR.RecTie.Used0 D μ) (j : Fin D.R.m) (f : D.what₂.Mt μ (j.1 + 1)) (y) (hy : X.R.extV j (X.tupT μ) y = X.fn hμ j f) (hf : D.what₂.eMt μ (j.1 + 1) f = f) : GlobalTheory.DownLift.GoodVt X.R j (X.tupT μ) y

theorem **lamUsed_of_id** (D : D7) (h : ∀ μ j f, D.what₂.thetaP μ j (D.what₂.thetaT μ j f) = f) (h1 : ∀ μ j σ, D.what₂.lamP μ j σ = 1) : LamUsed D

theorem **lamUsed_model** : ∃ D : D7, LamUsed D ∧ PartNe D

theorem **rowsC_of_isEmpty** (D : D7) [h : IsEmpty D.nd.Loc] : RowsC D

theorem **serLoc_of_isEmpty** (D : D7) [h : IsEmpty D.nd.Loc] : GlobalTheory.TypeLeaf.SerLoc D

theorem **definite_of_tied** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} (T : Tied hL D) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : (letI

theorem **loc_of_tied** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} (T : Tied hL D) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (w : D.what₂.places.Pl) (hw : D.what₂.muAt w ≠ D.what₂.mu0) [Nontrivial (T.P.H.cover.B.V j w)] : ∃ i : D.nd.Loc, D.nd.j i = j ∧ D.nd.w i = w

theorem **eq_zero_of_good** {D : D7} {cf : Formalism.L2.Host.CharFrame D} {nch : NChoice D} {tp : Formalism.L2.Host.TupTied D cf nch} (X : GlobalTheory.NJoin.RealAt D cf nch tp) (h : GoodIdempotentReadsAsGoodTypeProjector X) {μ} (hμ : Formalism.IndefR.RecTie.Used0 D μ) (j : Fin D.R.m) (h0 : D.what₂.eMt μ (j.1 + 1) = 0) (f : D.what₂.Mt μ (j.1 + 1)) (y) (hy : X.R.extV j (X.tupT μ) y = X.fn hμ j f) (hg : GlobalTheory.DownLift.GoodVt X.R j (X.tupT μ) y) : f = 0

theorem **loc_of_inputs** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} (I : Inputs hL D) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (σ : D.what₂.FT) (hσ : D.what₂.isot j σ ≠ ⊥) (w : D.what₂.places.Pl) (hw : D.what₂.muAt w ≠ D.what₂.mu0) : ∃ i : D.nd.Loc, D.nd.j i = j ∧ D.nd.w i = w

theorem **rows_model** : ∃ D : D7, LamUsed D ∧ PartNe D ∧ SumRepresentations.adm8 D ∧ ∀ L : SizeData₂ D, TwistedPlaces.adm3 L

theorem **baseChangeConditions_of_tied** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} (T : Tied hL D) : GlobalDatumCompatibility D

theorem **bc_iff_of_tied** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} (T : Tied hL D) : SphericalInertData.Conclusion1 D ↔ ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ π : D.nbc.AU j, SphericalInertData.SphInert D.nd D.naut D.nbc j π → ∃ Pa, SphericalInertData.IsBC D.nd D.naut D.nbc j π Pa

theorem **eC_eq** {D : D7} {cf : Formalism.L2.Host.CharFrame D} {nch : NChoice D} {tp : Formalism.L2.Host.TupTied D cf nch} (X : GlobalTheory.NJoin.RealAt D cf nch tp) : ∃ eR : X.G ≃* G0g D.nd, ∀ g, Formalism.IndefR.Links.RealDat.eC (A

theorem **unitaryRealisationReading_of_idempotent_acts_via_levelGroup** {D : D7} {cf : Formalism.L2.Host.CharFrame D} {nch : NChoice D} {tp : Formalism.L2.Host.TupTied D cf nch} (X : GlobalTheory.NJoin.RealAt D cf nch tp) (h : ∀ μ (hμ : Formalism.IndefR.RecTie.Used0 D μ) (j : Fin D.R.m) (f : D.what₂.Mt μ (j.1 + 1)) (l : (X.R.cV j).W →ₗ[ℂ] ℂ), Formalism.IndefR.Links.gaL (Formalism.IndefR.Links.Vc D j) (X.R.βV j) (X.R.datVt j (X.tupT μ)) X.R.eG (Formalism.IndefR.Links.RealDat.eC (A

theorem **not_rows_of_dup** (D : D7) (ν ν' : D.naut.CG 1) (hne : ν ≠ ν') (h : ∀ w, D.naut.locG ν w = D.naut.locG ν' w) : ¬ Rows D

theorem **lift_ne** {D : D7} (h : LamUsed D) (i : D.nd.Loc) (σ : D.what₂.FT) (hσ : D.what₂.isot (D.nd.j i) σ ≠ ⊥) : (D.what₂.isot (D.nd.j i) σ).map (D.what₂.thetaT (D.nd.mu i) (D.nd.j i)) ≠ ⊥

Development.AbelianVarietyTheory.Definitions.GaloisGeometry.VerticalBoundHeightCompare

theorem **prDR_ι_D** (i : K →ₐ[ℂ] L) (x : K) : prDR i (ExteriorAlgebra.ι K (D ℂ K x)) = ExteriorAlgebra.ι L (D ℂ L (i x))

theorem **fst_mem_prodRing** (i1 i2 : K →ₐ[ℂ] L) {B : Subalgebra ℂ K} {b : K} (hb : b ∈ B) : i1 b ∈ prodRing i1 i2 B

theorem **snd_mem_prodRing** (i1 i2 : K →ₐ[ℂ] L) {B : Subalgebra ℂ K} {b : K} (hb : b ∈ B) : i2 b ∈ prodRing i1 i2 B

theorem **range_prodMap** (i1 i2 : K →ₐ[ℂ] L) (B : Subalgebra ℂ K) : (prodMap i1 i2 B).range = prodRing i1 i2 B

theorem **IsProdField.prodMap_injective** {i1 i2 : K →ₐ[ℂ] L} (hP : IsProdField i1 i2) (B : Subalgebra ℂ K) : Function.Injective (prodMap i1 i2 B)

theorem **IsProdField.coe_prodEquiv** {i1 i2 : K →ₐ[ℂ] L} (hP : IsProdField i1 i2) (B : Subalgebra ℂ K) (z : B ⊗[ℂ] B) : (hP.prodEquiv B z : L) = prodMap i1 i2 B z

theorem **IsProdField.diagMul_spec** {i1 i2 : K →ₐ[ℂ] L} (hP : IsProdField i1 i2) (B : Subalgebra ℂ K) : ∀ b ∈ B, ∀ (h1 : i1 b ∈ prodRing i1 i2 B) (h2 : i2 b ∈ prodRing i1 i2 B), hP.diagMul B ⟨i1 b, h1⟩ = b ∧ hP.diagMul B ⟨i2 b, h2⟩ = b

theorem **IsProdField.eq_diagMul** {i1 i2 : K →ₐ[ℂ] L} (hP : IsProdField i1 i2) (B : Subalgebra ℂ K) (μ : prodRing i1 i2 B →ₐ[ℂ] K) (hμ : ∀ b ∈ B, ∀ (h1 : i1 b ∈ prodRing i1 i2 B) (h2 : i2 b ∈ prodRing i1 i2 B), μ ⟨i1 b, h1⟩ = b ∧ μ ⟨i2 b, h2⟩ = b) : μ = hP.diagMul B

theorem **diagEq_mem_prodRing** (i1 i2 : K →ₐ[ℂ] L) {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} (ht : ∀ l, t l ∈ B) (l : Fin d) : diagEq i1 i2 t l ∈ prodRing i1 i2 B

theorem **IsEtCoord.span_eq_top** {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} (ht : IsEtCoord B t) : Submodule.span K (Set.range fun l => D ℂ K (t l)) = ⊤

theorem **IsEtCoord.finrank_eq** {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} (ht : IsEtCoord B t) : Module.finrank K Ω[K⁄ℂ] = d

theorem **isDiagChart_iff** {i1 i2 : K →ₐ[ℂ] L} (hP : IsProdField i1 i2) {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} (ht : ∀ l, t l ∈ B) (h : L) : IsDiagChart i1 i2 B t h ↔ ∃ hh : h ∈ prodRing i1 i2 B, hP.diagMul B ⟨h, hh⟩ = 1 ∧ ∀ (z : L) (hz : z ∈ prodRing i1 i2 B), hP.diagMul B ⟨z, hz⟩ = 0 → ∃ (N : ℕ) (c : Fin d → L), (∀ l, c l ∈ prodRing i1 i2 B) ∧ h ^ N * z = ∑ l, c l * diagEq i1 i2 t l

theorem **IsDiagChart.ne_zero** {i1 i2 : K →ₐ[ℂ] L} {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} {h : L} (H : IsDiagChart i1 i2 B t h) : h ≠ 0

theorem **prodRing_le_chartRing** (i1 i2 : K →ₐ[ℂ] L) (B : Subalgebra ℂ K) (h : L) : prodRing i1 i2 B ≤ chartRing i1 i2 B h

theorem **ι_mul_listProd** {M : Type} [AddCommGroup M] [Module L M] (m : M) (ms : List M) : ExteriorAlgebra.ι L m * (ms.map (ExteriorAlgebra.ι L)).prod = (ms.map (ExteriorAlgebra.ι L)).prod * ExteriorAlgebra.ι L m ∨ ExteriorAlgebra.ι L m * (ms.map (ExteriorAlgebra.ι L)).prod = -((ms.map (ExteriorAlgebra.ι L)).prod * ExteriorAlgebra.ι L m)

theorem **listProd_shuffle** {M : Type} [AddCommGroup M] [Module L M] {d : ℕ} (u v : Fin d → M) (S : Finset (Fin d)) (js : List (Fin d)) : (js.map fun j => if j ∈ S then ExteriorAlgebra.ι L (u j) else -ExteriorAlgebra.ι L (v j)).prod = (((js.filter (· ∈ S)).map u).map (ExteriorAlgebra.ι L)).prod * (((js.filter (· ∉ S)).map v).map (ExteriorAlgebra.ι L)).prod ∨ (js.map fun j => if j ∈ S then ExteriorAlgebra.ι L (u j) else -ExteriorAlgebra.ι L (v j)).prod = -((((js.filter (· ∈ S)).map u).map (ExteriorAlgebra.ι L)).prod * (((js.filter (· ∉ S)).map v).map (ExteriorAlgebra.ι L)).prod)

theorem **listProd_mem_omB** {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} (ht : ∀ l, t l ∈ B) (js : List (Fin d)) : (js.map fun j => ExteriorAlgebra.ι K (D ℂ K (t j))).prod ∈ GlobalTheory.PD.omB ℂ K B js.length

theorem **diagForm_mem_biForms** (i1 i2 : K →ₐ[ℂ] L) {B : Subalgebra ℂ K} {d : ℕ} {t : Fin d → K} (ht : ∀ l, t l ∈ B) (h : L) (a : ℕ) : diagForm i1 i2 t a ∈ biForms i1 i2 B h a (d - a)

theorem **isProdField_fractionRing** (K : Type) [Field K] [Algebra ℂ K] [IsDomain (K ⊗[ℂ] K)] : IsProdField ((IsScalarTower.toAlgHom ℂ (K ⊗[ℂ] K) (FractionRing (K ⊗[ℂ] K))).comp Algebra.TensorProduct.includeLeft) ((IsScalarTower.toAlgHom ℂ (K ⊗[ℂ] K) (FractionRing (K ⊗[ℂ] K))).comp Algebra.TensorProduct.includeRight)

theorem **IsProdField.isDiagChart_adjoin** {i1 i2 : K →ₐ[ℂ] L} (hP : IsProdField i1 i2) {d : ℕ} (t : Fin d → K) : IsDiagChart i1 i2 (Algebra.adjoin ℂ (Set.range t)) t 1

theorem **isEtCoord_adjoin** {d : ℕ} (t : Fin d → K) [IsFractionRing (Algebra.adjoin ℂ (Set.range t)) K] (hli : LinearIndependent K fun l => D ℂ K (t l)) : IsEtCoord (Algebra.adjoin ℂ (Set.range t)) t

theorem **algebraMap_mem** (d : ℕ) (p : MvPolynomial (Fin d) ℂ) : algebraMap _ (Fn d) p ∈ polyRing d

theorem **isEtCoord** (d : ℕ) : IsEtCoord (polyRing d) (coord d)

theorem **isDomain_tensor** (k : Type) [Field k] (σ : Type) (A : Type) [CommRing A] [IsDomain A] [Algebra k A] : IsDomain (FractionRing (MvPolynomial σ k) ⊗[k] A)

theorem **isProdField** (d : ℕ) : IsProdField (pr1 d) (pr2 d)

theorem **isDiagChart** (d : ℕ) : IsDiagChart (pr1 d) (pr2 d) (polyRing d) (coord d) 1

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.AndLeft

theorem **OverCols.and_left** {Q R : V4.Slot} (h : OverCols Q) : OverCols fun K X f T c => Q K X f T c ∧ R K X f T c

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.AuxiliaryPrimeCoprimeSmallConductor

theorem **auxiliaryPrime_le_conductor_rpow** {A C : ℝ} (hA : 1 ≤ A) (hS : ∀ M : ℕ, 2 ≤ M → ∃ p : ℕ, p.Prime ∧ p % M = 1 ∧ (p : ℝ) ≤ (M : ℝ) ^ A) {R : Ranks} (H : Host R) (hc : H.ccK = ⊤) (h1 : (H.pde : ℝ) ≤ ((H.QS : ℝ) * H.q * H.d) ^ C) (h2 : (H.pde' : ℝ) ≤ ((H.QS : ℝ) * H.q * H.d) ^ C) : (q0H H : ℝ) ≤ ((H.QS : ℝ) * H.q * H.d) ^ ((3 + 2 * C) * A)

theorem **auxiliaryPrime_coprime_smallConductor** {R : Ranks} (H : Host R) : Nat.Coprime (q0H H) H.QS

theorem **hostExQW_of_hostExW** (hS : SmallPrimeOne) {la : ℝ} {P : HostReq} (hP : ∀ (c : ℝ) (s : Scale) (R : Ranks) (H : Host R) (q0 : ℕ) (hc : H.ccK = ⊤), P c s R H q0 → P c s R (relevel H hc (q0H H)) (q0H H)) (hH : HostExW la P) : HostExQW la P

theorem **hostExQ_of_hostEx** (hS : SmallPrimeOne) (hH : HostEx) : HostExQ

theorem **firstTypePrime_lt_rowPrime** (D : D7) (u : D.what₂.Word) : D.host.g 0 < (D.mem u).A.T

theorem **ObjC.and_q0Pt** {Q : PtProp} (h : ObjC Q) : ObjC (Q.and twistingGlobalPointProp)

theorem **objC_withObj_iff** (Q : PtProp) : ObjC (withObj Q) ↔ ObjC (Q.and (TamePt.and (MinkPt.and TwPt)))

theorem **hostWW_of_hostExQW_weightEx** {la : ℝ} {P : HostReq} (hH : HostExQW la P) (hW : WeightEx) : HostWW la P

theorem **hostW_of_hostExQ_weightEx** (hH : HostExQ) (hW : WeightEx) : HostW

theorem **not_objC_false** (hHost : HostW) : ¬ ObjC (fun _ _ _ _ _ _ _ _ _ => False)

theorem **not_kitC_false** : ¬ KitC (fun _ _ _ _ _ _ _ _ _ => False)

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.AuxiliaryPrimeMemAvoidedPrimes

theorem **setTu_large** {eps : ℝ} (Fm : FamP eps) (K : TransportedAlt) (t CH : ℝ) : (setTu Fm t).Large K CH ↔ Fm.Large (thrTu K t) CH

theorem **setTu_cls** (Q : PtProp) {eps : ℝ} (Fm : FamP eps) (t : ℝ) : cls Q (setTu Fm t) ↔ cls Q Fm

theorem **duty_of_cls** (Q : PtProp) {eps : ℝ} (Fm : FamP eps) (h : cls Q Fm) : Duty Fm

theorem **classOf_of_cls** (Q : PtProp) {eps : ℝ} (Fm : FamP eps) (h : cls Q Fm) : classOf Q Fm

theorem **ObjC.imp** {Q Q' : PtProp} (h : ObjC Q) (hQ : ∀ (C : FC) (p : Pt C), p.Has Q → p.Has Q') : ObjC Q'

theorem **HostExQW.imp** {la : ℝ} {P P' : HostReq} (h : HostExQW la P) (hP : ∀ C5 s R H q0, P C5 s R H q0 → P' C5 s R H q0) : HostExQW la P'

theorem **HostWW.imp** {la : ℝ} {P P' : HostReq} (h : HostWW la P) (hP : ∀ C5 s R H q0, P C5 s R H q0 → P' C5 s R H q0) : HostWW la P'

theorem **cond_pos** : 0 < H.QS * H.q * H.d

theorem **absNorm_ccK_ne_zero** : Ideal.absNorm H.ccK ≠ 0

theorem **rPrH_spec** : (rPrH H).Prime ∧ ¬ rPrH H ∣ H.QS * H.q * H.d * (H.pde - 1) * (H.pde' - 1) * Ideal.absNorm H.ccK

theorem **rPrH_le** {p : ℕ} (hp : p.Prime) (h : ¬ p ∣ H.QS * H.q * H.d * (H.pde - 1) * (H.pde' - 1) * Ideal.absNorm H.ccK) : rPrH H ≤ p

theorem **one_lt_rPrH_mul** : 1 < rPrH H * (H.QS * H.q * H.d)

theorem **globalTwisting_spec_fact** : (q0H H).Prime ∧ q0H H % (rPrH H * (H.QS * H.q * H.d)) = 1 ∧ ¬ q0H H ∣ Ideal.absNorm H.ccK

theorem **globalTwisting_le_fact** (hc : H.ccK = ⊤) {p : ℕ} (hp : p.Prime) (h : p % (rPrH H * (H.QS * H.q * H.d)) = 1) : q0H H ≤ p

theorem **mod_cond** : q0H H % (H.QS * H.q * H.d) = 1

theorem **rPrH_dvd** : rPrH H ∣ q0H H - 1

theorem **q0H_ne_pde_and_ne_pde'** : q0H H ≠ H.pde ∧ q0H H ≠ H.pde'

theorem **globalTwisting_congr_fact** {H' : Host R} (h0 : H.QS = H'.QS) (h1 : H.q = H'.q) (h2 : H.d = H'.d) (h3 : H.pde = H'.pde) (h4 : H.pde' = H'.pde') (h5 : Ideal.absNorm H.ccK = Ideal.absNorm H'.ccK) : q0H H = q0H H'

theorem **eq_of_from** {C : FC} (p : Pt C) {R : Ranks} {H : Host R} {a b : R.A → ℤ} {M : ℕ} {c : Fin M → Fin (R.k - 2) → Bool} {u0 : ℕ} (hF : p.From H a b c u0) (h1 : p.D.host.ccK = ⊤) (h2 : H.ccK = ⊤) : q0Av p.D = q0H H

theorem **auxiliaryPrime_mem_avoidedPrimes** (D : D7) : q0Av D ∈ avPrimes D

theorem **not_tameDistinct_of_eq_inv** {r g m : ℕ} [Fact g.Prime] (χ : Fin m → (ZMod g)ˣ) (j : Fin m) (h : χ j = (χ j)⁻¹) : ¬ TameDistinct r g χ

theorem **primeToPart_of_cube** {M : Type} [Monoid M] (l : ℕ) (hl : l.Prime) (h3 : l ≠ 3) (x : M) (hx : x ^ 3 = 1) : Formalism.CyclotomicNorms.primeToPart l x = x

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.BelowFiniteNonempty

theorem **moveHyp_of_notMem** (hPr : Pr.Ok) (h : ∀ w ∈ SG, (Pr.T u : 𝒪 X.E) ∉ w.asIdeal) (P : Pr.vT u) : MoveHyp (Pr.vT u) SG P

theorem **above_finite_nonempty** (hX : X.Ok) (hPr : Pr.Ok) {P : Pr.vT u} (hP : MoveHyp (Pr.vT u) SG P) (d' : DecoSet X.E n (Pr.vT u) SG) : {d₁ : DecoSet X.E n (Pr.vT u) SG | d'.1.IsUp P (Pr.T u : 𝒪 X.E) d₁.1}.Finite ∧ {d₁ : DecoSet X.E n (Pr.vT u) SG | d'.1.IsUp P (Pr.T u : 𝒪 X.E) d₁.1}.Nonempty

theorem **below_eq_dnSet** (hX : X.Ok) (hPr : Pr.Ok) {P : Pr.vT u} (hP : MoveHyp (Pr.vT u) SG P) (d : DecoSet X.E n (Pr.vT u) SG) : {d₀ : DecoSet X.E n (Pr.vT u) SG | d₀.1.IsUp P (Pr.T u : 𝒪 X.E) d.1} = d.dnSet P (Pr.T u : 𝒪 X.E)

theorem **below_finite_nonempty** (hX : X.Ok) (hPr : Pr.Ok) {P : Pr.vT u} (hP : MoveHyp (Pr.vT u) SG P) (d : DecoSet X.E n (Pr.vT u) SG) : {d₀ : DecoSet X.E n (Pr.vT u) SG | d₀.1.IsUp P (Pr.T u : 𝒪 X.E) d.1}.Finite ∧ {d₀ : DecoSet X.E n (Pr.vT u) SG | d₀.1.IsUp P (Pr.T u : 𝒪 X.E) d.1}.Nonempty

theorem **T_notMem_rowSG** (hPr : Pr.Ok) (hsel : IsSel X sel) (hT : Pr.T u ∈ S) : ∀ w ∈ rowSG Pr sel mu, (Pr.T u : 𝒪 X.E) ∉ w.asIdeal

theorem **alcoveAt_of_sizes** {a : ℝ} (h : Sizes Pr W a) (hj : j ≤ R.m) : AlcoveAt W u j

theorem **T_mem** (h : Hyp Z s cc δ χ cD sel S mu cst u) : Z.Pr.T u ∈ S

theorem **g_mem** (h : Hyp Z s cc δ χ cD sel S mu cst u) (k : Fin 2) : Z.Pr.g k ∈ S

theorem **ram_mem** (h : Hyp Z s cc δ χ cD sel S mu cst u) {p : ℕ} (hp : p.Prime) (hd : p ∣ H.host.condN) : p ∈ S

theorem **moveHyp** (h : Hyp Z s cc δ χ cD sel S mu cst u) {SG : Finset (HeightOneSpectrum (𝒪 Z.X.E))} (hSG : SG ⊆ rowSG Z.Pr sel mu) (P : Z.Pr.vT u) : MoveHyp (Z.Pr.vT u) SG P

theorem **alcoveAt** (h : Hyp Z s cc δ χ cD sel S mu cst u) {j : ℕ} (hj : j ≤ Z.R.m) : AlcoveAt Z.W u j

theorem **offBox_indicator** {a b : ℕ} (N : Submodule (𝒪 X.E) (Sp X.E a)) (N' : Submodule (𝒪 X.E) (Sp X.E b)) : InBoxOfRowSelection sel mu cst u N N' fun x => if x ∈ homAt (rowSG Pr sel mu) N N' then (1 : X.K) else 0

theorem **term_congr** {SG₁ SG₂ SG₁' SG₂' : Finset (HeightOneSpectrum (𝒪 X.E))} (d₁ : DecoSet X.E (R.N i) (Pr.vT u) SG₁) (d₂ : DecoSet X.E (R.N i) (Pr.vT u) SG₂) (d₁' : DecoSet X.E (R.N j) (Pr.vT u) SG₁') (d₂' : DecoSet X.E (R.N j) (Pr.vT u) SG₂') (hN : d₁.1.N = d₂.1.N) (hf : d₁.1.f = d₂.1.f) (hN' : d₁'.1.N = d₂'.1.N) (hf' : d₁'.1.f = d₂'.1.f) (fr : Frs X.E (R.N i) (rowSG Pr sel mu)) (fr' : Frs X.E (R.N j) (rowSG Pr sel mu)) (Ξ₀ : HomE X.E (R.N i) (R.N j) → X.K) : term W Ch sel mu cst A u i j d₁ fr d₁' fr' Ξ₀ = term W Ch sel mu cst A u i j d₂ fr d₂' fr' Ξ₀

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.BhattScholzeCrystallineLattices

theorem **Hypotheses2.toU** (P : Hypotheses2) : HypothesesU

theorem **HypothesesU.geometricLemma** (P : HypothesesU) : GlobalTheory.Published.GeometricLemma

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ClassCountBoundAtConstructedPoints

theorem **row_all** {Bd : BdT} {Q : V4.Slot} {Lk : LkT Bd} (hn4 : ClassCountBoundAtConstructedPoints Q Lk) (goodFT : GoodFourierTransformAtConstructedPoints Q Lk) (hRanks : RankConditionAtConstructedPoints Q Lk) (jOn : JInvariantConditionAtConstructedPoints Q Lk) (bu : BundleBoundAtConstructedPoints Q Lk) (rr : RiemannRochAtConstructedPoints Q Lk) (la : LambdaConditionAtConstructedPoints Q Lk) : Row Q Lk fun 𝒜 _ => All 𝒜

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ContFromConst

theorem **ContFrom.mono** {ht : G → ℝ≥0} {σ₀ σ₁ : ℝ} {e : ℂ → G → ℂ} (h : ContFrom ht σ₀ e) (hσ : σ₀ ≤ σ₁) : ContFrom ht σ₁ e

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.CountsIrreducibilityConstants

theorem **d3_a** {C0 : ℝ} {𝒜 : FamClass} {eps : ℝ} {Fm : FamP eps} (h : D3 C0 𝒜 Fm) : 𝒜 Fm

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DecayZero

theorem **decay_zero** : Decay (fun _ : Fin N → AF F => (0 : ℂ))

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DifferentiableAtChart

theorem **ad_ext** {x y : Ad E} (h1 : pInf E x = pInf E y) (h2 : pFin E x = pFin E y) : x = y

theorem **ga_ext** {g h : GA E a} (h1 : GeneralLinearGroup.map (pInf E) g = GeneralLinearGroup.map (pInf E) h) (h2 : GeneralLinearGroup.map (pFin E) g = GeneralLinearGroup.map (pFin E) h) : g = h

@[simp] theorem **pInf_infGL** (y : GL (Fin a) (Cinf E)) : GeneralLinearGroup.map (pInf E) (infGL E a y) = y

@[simp] theorem **pFin_infGL** (y : GL (Fin a) (Cinf E)) : GeneralLinearGroup.map (pFin E) (infGL E a y) = 1

@[simp] theorem **pInf_finGL** (k : GL (Fin a) (Af E)) : GeneralLinearGroup.map (pInf E) (finGL E a k) = 1

theorem **infGL_mul_finGL** (g : GA E a) : infGL E a (GeneralLinearGroup.map (pInf E) g) * finGL E a (GeneralLinearGroup.map (pFin E) g) = g

theorem **infGL_comm** (y : GL (Fin a) (Cinf E)) (k : GL (Fin a) (Af E)) : infGL E a y * finGL E a k = finGL E a k * infGL E a y

theorem **det_infGL** (y : GL (Fin a) (Cinf E)) : GeneralLinearGroup.det (infGL E a y) = infId E (GeneralLinearGroup.det y)

theorem **contDiff_det** : ContDiff ℝ ∞ fun x : Lie E a => (Matrix.of x).det

theorem **isOpen_invSet** : IsOpen (invSet E a)

theorem **val_mem_invSet** (y : GL (Fin a) (Cinf E)) : (toL y.val) ∈ invSet E a

theorem **chart_val** (φ : GA E a → ℂ) (g : GA E a) (y : GL (Fin a) (Cinf E)) : chart φ g (toL y.val) = φ (g * infGL E a y)

theorem **chart_mul** (φ : GA E a → ℂ) (g : GA E a) (y : GL (Fin a) (Cinf E)) : chart φ (g * infGL E a y) = chart φ g ∘ lmul (toL y.val)

theorem **chart_rt_inf** (φ : GA E a → ℂ) (g : GA E a) (y : GL (Fin a) (Cinf E)) : chart (fun x => φ (x * infGL E a y)) g = chart φ g ∘ rmul (toL y.val)

theorem **chart_rt_fin** (φ : GA E a → ℂ) (g : GA E a) (k : GL (Fin a) (Af E)) : chart (fun x => φ (x * finGL E a k)) g = chart φ (g * finGL E a k)

theorem **rt_mem_Sm** (h : GA E a) (φ : GA E a → ℂ) (hφ : φ ∈ Sm E a) : GlobalTheory.PRep.rightReg ℂ (GA E a) ℂ h φ ∈ Sm E a

@[simp] theorem **rt_apply** (h g : GA E a) (φ : Sm E a) : (rt E a h φ).1 g = φ.1 (g * h)

theorem **contDiffAt_chart** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) {x : Lie E a} (hx : x ∈ invSet E a) : ContDiffAt ℝ ∞ (chart φ g) x

theorem **differentiableAt_chart** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (g : GA E a) : DifferentiableAt ℝ (chart φ g) (oneL E a)

theorem **chart_lieD** {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) (X : Lie E a) (g : GA E a) {x : Lie E a} (hx : x ∈ invSet E a) : chart (lieD X φ) g x = fderiv ℝ (chart φ g) x (rmul X x)

theorem **lieD_mem_Sm** (X : Lie E a) {φ : GA E a → ℂ} (hφ : φ ∈ Sm E a) : lieD X φ ∈ Sm E a

@[simp] theorem **lie_apply** (X : Lie E a) (φ : Sm E a) (g : GA E a) : (lie E a X φ).1 g = lieD X φ.1 g

theorem **hasDerivAt_curve** (φ : Sm E a) (g : GA E a) (c : ℝ → GL (Fin a) (Cinf E)) (X : Lie E a) (h0 : c 0 = 1) (hc : HasDerivAt (fun t => toL (c t).val) X 0) : HasDerivAt (fun t => φ.1 (g * infGL E a (c t))) (lieD X φ.1 g) 0

theorem **lie_rt_fin** (X : Lie E a) (k : GL (Fin a) (Af E)) : lie E a X * rt E a (finGL E a k) = rt E a (finGL E a k) * lie E a X

theorem **lie_comm** (X Y : Lie E a) : lie E a X * lie E a Y − lie E a Y * lie E a X = lie E a (toL (Matrix.of X * Matrix.of Y)) − lie E a (toL (Matrix.of Y * Matrix.of X))

theorem **id_idem** (τ : E →+* ℂ) : idRe τ * idRe τ − idIm τ * idIm τ = idRe τ ∧ idRe τ * idIm τ + idIm τ * idRe τ = idIm τ

theorem **lieS_apply** (i j : Fin a) (x : Cinf E) : lieS E a i j x = lie E a (toL (Matrix.single i j x))

theorem **lieS_comm** (i j k l : Fin a) (x y : Cinf E) : lieS E a i j x * lieS E a k l y - lieS E a k l y * lieS E a i j x = (if j = k then lieS E a i l (x * y) else 0) - (if l = i then lieS E a k j (x * y) else 0)

theorem **glRel_Eop** (τ : E →+* ℂ) : Formalism.REAL.GlRel (Eop (a

theorem **cap_comm** (τ : E →+* ℂ) (u : ℂ) (i j : Fin a) : cap τ u * Eop τ i j = Eop τ i j * cap τ u

theorem **cap_highest** (τ : E →+* ℂ) (ν : Fin a → ℂ) (φ : Sm E a) (h : Formalism.REAL.IsHW (Eop τ) ν φ) (u : ℂ) : cap τ u φ = (∏ k : Fin a, (u + (ν k + (((a : ℂ) - 1) / 2 - (k : ℕ))))) • φ

theorem **Zalg_stable** {V : Submodule ℂ (Sm E a)} (hV : ∀ X, ∀ φ ∈ V, lie E a X φ ∈ V) {z : Module.End ℂ (Sm E a)} (hz : z ∈ Zalg E a) : ∀ φ ∈ V, z φ ∈ V

@[simp] theorem **op_val** (o : Op E a) (φ : π.V) : (π.op o φ).1 = act o φ.1

theorem **HasInfChar.unique** {τ : E →+* ℂ} {m m' : Multiset ℂ} (h : π.HasInfChar τ m) (h' : π.HasInfChar τ m') : m = m'

theorem **iso_cap** (T : π.V ≃ₗ[ℂ] π'.V) (hT : ∀ o φ, T (π.op o φ) = π'.op o (T φ)) (τ : E →+* ℂ) (u : ℂ) (φ : π.V) : ∃ h : cap τ u φ.1 ∈ π.V, (T ⟨_, h⟩).1 = cap τ u (T φ).1

theorem **Iso.hasInfChar** (h : Iso π π') {τ : E →+* ℂ} {m : Multiset ℂ} (hm : π.HasInfChar τ m) : π'.HasInfChar τ m

theorem **locRep_eq** (w : HeightOneSpectrum (𝒪 E)) (g : GL (Fin a) (w.adicCompletion E)) : π.locRep w g = π.op (.fin (locGL E a w g))

theorem **isSmooth_finPart** : GlobalTheory.PRep.IsSmooth π.finPart

theorem **isSmooth_locRep** (w : HeightOneSpectrum (𝒪 E)) : GlobalTheory.PRep.IsSmooth (π.locRep w)

theorem **IsLocComp.congr** {w : HeightOneSpectrum (𝒪 E)} {σ σ' : SmIrr (GL (Fin a) (w.adicCompletion E))} (h : π.IsLocComp w σ) (hπ : Iso π π') (hσ : Nonempty (σ.ρ.Equiv σ'.ρ)) : π'.IsLocComp w σ'

theorem **Iso.hasSatGL** (h : Iso π π') {w : HeightOneSpectrum (𝒪 E)} {B : Multiset ℂˣ} (hB : HasSatGL w (π.locRep w) B) : HasSatGL w (π'.locRep w) B

theorem **Iso.hasCentChar** (h : Iso π π') {η : HChar E} (hη : π.HasCentChar η) : π'.HasCentChar η

theorem **Iso.hasFixed** (h : Iso π π') {U : Subgroup (GL (Fin a) (Af E))} (hU : π.HasFixed U) : π'.HasFixed U

theorem **oneFn_ne_zero** : oneFn E a ≠ 0

theorem **lie_oneFn** (X : Lie E a) : lie E a X (oneFn E a) = 0

theorem **act_oneFn** (o : Op E a) : act o (oneFn E a) = opTriv o • oneFn E a

theorem **isStable_const** : IsStable (Submodule.span ℂ {oneFn E a})

theorem **isAutForm_const** {φ : Sm E a} (hφ : φ ∈ Submodule.span ℂ {oneFn E a}) : IsAutForm φ

theorem **cap_oneFn** (τ : E →+* ℂ) (u : ℂ) : cap τ u (oneFn E a) = (∏ k : Fin a, (u + (((a : ℂ) - 1) / 2 - (k : ℕ)))) • oneFn E a

theorem **HasInfChar.infChar_eq** {π : CG E a} {τ : E →+* ℂ} {m : Multiset ℂ} (h : π.HasInfChar τ m) : π.infChar τ = m

theorem **card_infChar** (π : CG E a) (τ : E →+* ℂ) : Multiset.card (π.infChar τ) = a

@[simp] theorem **hasSat_mk** (π : CuspSub E a) (w : HeightOneSpectrum (𝒪 E)) (B : Multiset ℂˣ) : (mk π).HasSat w B ↔ HasSatGL w (π.locRep w) B

theorem **HasSat.unique** {π : CG E a} {w : HeightOneSpectrum (𝒪 E)} {B B' : Multiset ℂˣ} (h : π.HasSat w B) (h' : π.HasSat w B') : B = B'

theorem **HasSat.card** {π : CG E a} {w : HeightOneSpectrum (𝒪 E)} {B : Multiset ℂˣ} (h : π.HasSat w B) : B.card = a

@[simp] theorem **hasCentChar_mk** (π : CuspSub E a) (η : HChar E) : (mk π).HasCentChar η ↔ π.HasCentChar η

@[simp] theorem **hasFixed_mk** (π : CuspSub E a) (U : Subgroup (GL (Fin a) (Af E))) : (mk π).HasFixed U ↔ π.HasFixed U

theorem **isRegAlg_triv** : (mk (CuspSub.triv E (le_refl 1))).IsRegAlg

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DiscreteSeriesCoefficientsCuspidal

theorem **ArithmeticGeometryHyps.localGlobalAboveL_of_hypotheses** (h : ArithmeticGeometryHyps) : GlobalTheory.GalOfCusp.LocalGlobalAboveL

theorem **ArithmeticGeometryHyps.geometricLemma_of_cells** (h : ArithmeticGeometryHyps) (hc : GlobalTheory.Published.BernsteinZelevinskyFiniteness → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k], GlobalTheory.SmGL.CellSemisimplification k) : GlobalTheory.Published.GeometricLemma

theorem **ArithmeticGeometryHyps.geometricLemma** (h : ArithmeticGeometryHyps) : GlobalTheory.Published.GeometricLemma

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DisjointConductorPlacesExcludedSet

theorem **prime_eq_of_overP** {R : Ranks} {H : Formalism.BasicNotions.Host R} {p q : ℕ} (hp : p.Prime) (hq : q.Prime) {v : HeightOneSpectrum (𝒪 H.E)} (h1 : OverP H p v) (h2 : OverP H q v) : p = q

theorem **exists_overP** {R : Ranks} (H : Formalism.BasicNotions.Host R) {p : ℕ} (hp : p.Prime) : ∃ v, OverP H p v

theorem **TupNarrow.unr** {C : CharFrame D} {nch : NChoice D} {Tp : TupTied D C nch} (h : TupNarrow Tp) (μ : D.what₂.Tup) {p : ℕ} (hp : p.Prime) (hS : p ∈ untwistedPrimes D) {v : HeightOneSpectrum (𝒪 D.host.E)} (hv : OverP D.host p v) : (Tp.xi μ).UnrAt v

theorem **not_unr_of_isDelta** {R : Ranks} {H : Formalism.BasicNotions.Host R} {p : ℕ} {χ : HChar H.E} (h : IsDelta (H

theorem **tuples_ne** {C : CharFrame D} {nch : NChoice D} (Tp : TupTied D C nch) : D.what₂.mu0 ≠ D.what₂.muStar ∧ D.what₂.mu0 ≠ D.what₂.muStar' ∧ D.what₂.muStar ≠ D.what₂.muStar'

theorem **TupFamilies.xi** {C : CharFrame D} {nch : NChoice D} (Tp : TupTied D C nch) (h : TupFamilies D) (μ : D.what₂.Tup) : Tp.xi μ = 1 ∨ Tp.xi μ = C.del ∨ Tp.xi μ = C.del' ∨ ∃ w, D.what₂.muAt w ≠ D.what₂.mu0 ∧ Tp.xi μ = C.delAt (D.what₂.places.resChar w) * Tp.psi w

theorem **norm_xi_val** {C : CharFrame D} {nch : NChoice D} (Tp : TupTied D C nch) (μ : D.what₂.Tup) (P : HeightOneSpectrum (𝒪 D.host.E)) : ‖(((Tp.xi μ).val P : ℂˣ) : ℂ)‖ = 1

theorem **nuAt_iff** {C : CharFrame D} {nch : NChoice D} (Tp : TupTied D C nch) (μ : D.what₂.Tup) (P : HeightOneSpectrum (𝒪 D.host.E)) (ν : ℕ → ℂ) : NuAt Tp μ P ν ↔ ∃ c : ℂ, ‖c‖ = 1 ∧ ∀ j : Fin D.R.m, ν (j.1 + 1) = c * ((((C.mu0 j).val P)⁻¹ : ℂˣ) : ℂ)

theorem **NuAt.ratio** {C : CharFrame D} {nch : NChoice D} {Tp : TupTied D C nch} {μ : D.what₂.Tup} {P : HeightOneSpectrum (𝒪 D.host.E)} {ν : ℕ → ℂ} (h : NuAt Tp μ P ν) (i j : Fin D.R.m) : ν (i.1 + 1) / ν (j.1 + 1) = (((C.mu0 j).val P : ℂˣ) : ℂ) / (((C.mu0 i).val P : ℂˣ) : ℂ)

theorem **range_eq_of_faithful** {Γ : Type*} [CommGroup Γ] [Finite Γ] (φ ψ : Γ →* ℂˣ) (hφ : Function.Injective φ) (hψ : Function.Injective ψ) : φ.range = ψ.range

theorem **TameTied.pow** {C : CharFrame D} (h : TameTied C) (k : Fin 2) (v : D.nd.PlG k) : ∃ γ : (ZMod (D.host.g k))ˣ, Subgroup.zpowers γ = ⊤ ∧ ∀ j : Fin D.R.m, D.nd.chiT k v j = γ ^ (2 * (j.1 + 1))

theorem **IsFurtherRealPlace.mem** {L : SizeData₂ D} {v0 : HeightOneSpectrum (𝒪 D.host.F)} (h : IsFurtherRealPlace L v0) : Ideal.absNorm v0.asIdeal ∈ D.host.levelPrimes ∧ ((Ideal.absNorm v0.asIdeal : ℕ) : 𝒪 D.host.F) ∈ v0.asIdeal

theorem **TwistAt.disjoint_conductorPlaces_excludedSet** {L : SizeData₂ D} {nch : NChoice D} {xd : ThData D} {v0 : HeightOneSpectrum (𝒪 D.host.F)} {rep : ClassGroup (𝒪 D.host.K) → Ideal (𝒪 D.host.K)} (h : TwistAt L nch xd v0 rep) {w : D.what₂.places.Pl} (hw : D.what₂.muAt w ≠ D.what₂.mu0) : D.ntw.NonzeroIndexPlace (nch.twI w) ∧ Disjoint (nch.Q w) (D.ntw.S0 (nch.twI w))

theorem **SpaceAt.real** {X : DivX D} {EX : ErrX D} {lev : LevelData D} {nch : NChoice D} {th : ThetaX D} {C : CharFrame D} {Tp : TupTied D C nch} (h : SpaceAt X EX lev nch th C Tp) : Formalism.L2.Theta.RealTied D ∧ Formalism.L2.Herm.VolumeCompatible D

theorem **SpaceAt.base** {X : DivX D} {EX : ErrX D} {lev : LevelData D} {nch : NChoice D} {th : ThetaX D} {C : CharFrame D} {Tp : TupTied D C nch} (h : SpaceAt X EX lev nch th C Tp) : (∀ (u : D.what₂.Word) (pb : 𝒪 D.host.E), (pb : D.host.E) = D.host.conj (EX.piT u : D.host.E) → (lev u).Lb ⊔ Ideal.span {pb} • Submodule.span (𝒪 D.host.E) (dualSet (D.sp.cm.sig : D.host.E →+* D.host.E) D.sp.gram0 (lev u).Lb) = D.sp.crdLat0 (D.sp.X 0).L0) ∧ ∃ dt : Formalism.L2.Herm.unitaryOneFiniteAdeles D.sp.cm →* D.Dt, ∀ (u : D.what₂.Word) (c : (D.mem u).C.cls 0) (x : Formalism.L2.Herm.unitaryOneFiniteAdeles D.sp.cm), Formalism.L2.Herm.IsDetOf D.sp.cm x ((lev u).y c : GL (Fin D.R.k) (D.host.E ⊗[D.host.F] Formalism.L2.Herm.Af D.host.F)) → D.detCls u 0 c = dt x

theorem **TieCoreAt.char** {C : CharFrame D} {Tp : TupTied D C ex.nch} (h : TieCoreAt ex C Tp) : CharAt X ex.errX ex.nch ex.th C Tp

theorem **TieCoreAt.space** {C : CharFrame D} {Tp : TupTied D C ex.nch} (h : TieCoreAt ex C Tp) : SpaceAt X ex.errX ex.lev ex.nch ex.th C Tp

theorem **TieCore.tuples_ne** (h : TieCore D L gatT Gx Gy K X cX ex) : D.what₂.mu0 ≠ D.what₂.muStar ∧ D.what₂.mu0 ≠ D.what₂.muStar' ∧ D.what₂.muStar ≠ D.what₂.muStar'

theorem **TieCore.pde_mem_Qp** (h : TieCore D L gatT Gx Gy K X cX ex) : D.host.pde ∈ X.Qp D.what₂.muStar

theorem **TieCore.used** (h : TieCore D L gatT Gx Gy K X cX ex) (μ : D.what₂.Tup) : DV.Used D μ

theorem **TieCore.base** (h : TieCore D L gatT Gx Gy K X cX ex) (u : D.what₂.Word) : product_first_zeroth_parts.admB D u

theorem **TieCore.jet** (h : TieCore D L gatT Gx Gy K X cX ex) : JetAt D

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DisjointExceptional

theorem **Pt.has_and** {C : FC} (p : Pt C) (Q Q' : PtProp) : p.Has (Q.and Q') ↔ p.Has Q ∧ p.Has Q'

theorem **classOf_and** (Q Q' : PtProp) {eps : ℝ} (Fm : FamP eps) : classOf (Q.and Q') Fm ↔ classOf Q Fm ∧ classOf Q' Fm

theorem **KitExW.imp** {la : ℝ} {Q Q' : PtProp} (h : KitExW la Q) (hQ : ∀ (C : FC) (cD ch : ℝ) (p : Pt C), Val C cD ch p.L → HostArith p.D.host → p.Has Q → p.Has Q') : KitExW la Q'

theorem **KitEx.imp** {Q Q' : PtProp} (h : KitEx Q) (hQ : ∀ (C : FC) (cD ch : ℝ) (p : Pt C), Val C cD ch p.L → HostArith p.D.host → p.Has Q → p.Has Q') : KitEx Q'

theorem **ExFull.kitEx** {Q : PtProp} (h : ExFull Q) : KitEx Q

theorem **Pt.Same.refl** {C : FC} (p : Pt C) : p.Same p

theorem **Pt.Same.weights** {C : FC} {p p' : Pt C} (h : p.Same p') (t : p.D.host.TT) (t' : p'.D.host.TT) (e : (t : p.D.R.A).val = (t' : p'.D.R.A).val) : p.L.aT t = p'.L.aT t' ∧ p.L.bT t = p'.L.bT t'

theorem **Pt.Same.QS_eq** {C : FC} {p p' : Pt C} (h : p.Same p') : p.D.host.QS = p'.D.host.QS

theorem **Pt.Same.art_eq** {C : FC} {p p' : Pt C} (h : p.Same p') : artNum p.D.host.toHostRaw = artNum p'.D.host.toHostRaw

theorem **Pt.Same.rd_eq** {C : FC} {p p' : Pt C} (h : p.Same p') : p.D.host.rdE = p'.D.host.rdE

theorem **Kit.cover_weights** {C : FC} {cD ch : ℝ} {s : Scale} (k : Kit C cD ch s) (i0 : k.Ix) (u0 : ℕ) (hu : u0 < (k.pt i0).D.R.n) : ∃ i, (k.pt i).D.host.u0.val = u0 ∧ ∀ (t : (k.pt i).D.host.TT) (t0 : (k.pt i0).D.host.TT), (t : (k.pt i).D.R.A).val = (t0 : (k.pt i0).D.R.A).val → (k.pt i).L.aT t = (k.pt i0).L.aT t0 ∧ (k.pt i).L.bT t = (k.pt i0).L.bT t0

theorem **Pt.disjoint_exceptional** {C : FC} (p : Pt C) : Disjoint (Formalism.CyclotomicNorms.BB p.D.R.r (p.D.host.g 0)) (Formalism.CyclotomicNorms.BB p.D.R.r (p.D.host.g 1))

theorem **Val.qdRows** {C : FC} {cD ch : ℝ} {D : D7} {L : SizeData₂ D} (h : Val C cD ch L) : QDRowsW D L.sc

theorem **Val.mem_box** {C : FC} {cD ch : ℝ} {D : D7} {L : SizeData₂ D} (h : Val C cD ch L) : G.aL L ∈ G.box D (G.winA D L.sc) ∧ G.bL L ∈ G.box D (G.winB D C.GK L.sc)

theorem **Pt.has_reqPt_iff** {C : FC} {cD ch : ℝ} (p : Pt C) (h : Val C cD ch p.L) : p.Has ReqPt ↔ AuxBlock8.WReq p.ta.Gx C.GK (G.aL p.L) (G.bL p.L) ∧ BoxesW p.D p.L.sc

theorem **classOf_reqPt** {eps : ℝ} (Fm : FamP eps) (h : classOf ReqPt Fm) : (∀ x, Fm.x1 < x → ∀ i : Fm.Ix x, AuxBlock8.admW (Fm.D x i) (Fm.L x i) (Fm.Gx x i) Fm.GK) ∧ BoxesWF Fm ∧ QDRowsWF Fm

theorem **BoxesWF.boxesF** {eps : ℝ} {Fm : FamP eps} (h : BoxesWF Fm) (hJ : Fm.J = 1) : BoxesF Fm

theorem **QDRowsWF.qdRowsF** {eps : ℝ} {Fm : FamP eps} (h : QDRowsWF Fm) (hJ : Fm.J = 1) : QDRowsF Fm

theorem **boxesWF_iff** {eps : ℝ} (Fm : FamP eps) : BoxesWF Fm ↔ BoxesF Fm

theorem **qdRowsWF_iff** {eps : ℝ} (Fm : FamP eps) : QDRowsWF Fm ↔ QDRowsF Fm

theorem **mod_lcm_of_mod_mul** {p N : ℕ} (hN : 0 < N) (h : p % (15 * N) = 1) : p % Nat.lcm 15 N = 1

theorem **mod_of_mod_lcm** {p N : ℕ} (hN : 1 < N) (h : p % Nat.lcm 15 N = 1) : p % 15 = 1 ∧ p % N = 1

theorem **classOf_twPt** {eps : ℝ} (Fm : FamP eps) (h : classOf TwPt Fm) : TwGood Fm

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DivRows3GIff

theorem **divRows3G_iff** (P : RowSideFamily ex) : GlobalTheory.Cp.DivRows3G P ↔ Formalism.L2.DIV.DivRows3G P

theorem **servT_iff** (𝒞 : FamClass) (sel : Formalism.L2.TSel 𝒞) : GlobalTheory.Cp.ServT 𝒞 sel ↔ Formalism.L2.G.ServT.ServT 𝒞 sel

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.DivisionSecondComplementAtConstructedPoints

theorem **row_all** {Bd : BdT} {Q : V4.Slot} {Lk : LkT Bd} (hh : HeightConditionAtConstructedPoints Q Lk) (hInstR : InstanceRankAtConstructedPoints Q Lk) (aC2 : DivisionSecondComplementAtConstructedPoints Q Lk) : Row Q Lk fun $\mathcal{A}$ _ => All $\mathcal{A}$

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.EventuallyAllScaleContextsAndOfBoth

theorem **eventuallyAllScaleContexts_and_of_both** {p q : Ctx → Prop} (hp : Ev p) (hq : Ev q) : Ev fun K => p K ∧ q K

theorem **eventuallyAllScaleContexts_mono_of_imp** {p q : Ctx → Prop} (h : ∀ K, p K → q K) (hp : Ev p) : Ev q

theorem **LargeBaseCondition.and** (h : LargeBaseCondition KP Q R) (h' : LargeBaseCondition KP Q R') : LargeBaseCondition KP Q (sAnd R R')

theorem **LargeBaseCondition.passes** (h : LargeBaseCondition KP Q R) (h1 : V4.ExTrue KP Q) (h2 : V4.Export Q) (h3 : V4.Retwist Q Q') : V4.ExTrue KP (sAnd Q R) ∧ V4.Export (sAnd Q R) ∧ V4.Retwist (sAnd Q R) (sAnd Q' R)

theorem **Row.strengthen** {H : ∀ $\mathcal{A}$: FamClass, RawSel Bd $\mathcal{A}$ → Prop} (hQ : ∀ K X f T c, Q K X f T c → Q' K X f T c) (h : Row Q' Lk H) : Row Q Lk H

theorem **RowL.and** {H H' : ∀ $\mathcal{A}$: FamClass, RawSel Bd $\mathcal{A}$ → Prop} (h : RowL KP Q Q' Lk H) (h' : RowL KP Q Q' Lk H') : RowL KP Q Q' Lk fun $\mathcal{A}$ σ => H $\mathcal{A}$ σ ∧ H' $\mathcal{A}$ σ

theorem **RowL.elim** {H : ∀ $\mathcal{A}$: FamClass, RawSel Bd $\mathcal{A}$ → Prop} (hx : ExBL KP Q Q' Lk) (h : RowL KP Q Q' Lk H) : ∃ R : V4.Slot, LargeBaseCondition KP Q R ∧ ExB (sAnd Q' R) Lk ∧ Row (sAnd Q' R) Lk H

theorem **Row.rowL** {H : ∀ $\mathcal{A}$: FamClass, RawSel Bd $\mathcal{A}$ → Prop} (r : LargeBaseCondition KP Q R) (h : Row Q' Lk H) : RowL KP Q Q' Lk H

theorem **largeBaseProperty_true** : LargeBaseCondition KP Q fun _ _ _ _ _ => True

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ExistsAboveNe

theorem **off_mv** (y : Ys X n) (p : PtY ST SG Q y) (d : DecoSet X.E n ST SG) (h1 : SameOff ST p.1.1.deco.1.N d.1.N) (h2 : p.1.1.deco.1.be = d.1.be) : (offY y).r p (p.mv d)

theorem **ok_repl** {d : Deco X.E n ST SG} (hd : d.Ok) {e : Deco X.E n ST ∅} (he : e.Ok) (hG : ∀ w : SG, localization₂ w.1.asIdeal d.N = localization₂ w.1.asIdeal e.N) : Deco.Ok ((e.N, e.f, d.be) : Deco X.E n ST SG)

theorem **exists_uf** (γ : GL (Fin n) X.E) (h : (γ : Matrix (Fin n) (Fin n) X.E)ᴴ * (γ : Matrix (Fin n) (Fin n) X.E) = 1) : ∃ g : UF X.E n, ufGL X.E n g = γ

theorem **localization_eq_of_sameOff** (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ w ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : $\mathcal{O}$ X.E) ∉ w.asIdeal) {n : ℕ} {N N' : Submodule ($\mathcal{O}$ X.E) (Sp X.E n)} (h : SameOff (Pr.vT u) N N') (w : rowSG Pr sel mu) : localization₂ w.1.asIdeal N = localization₂ w.1.asIdeal N'

theorem **splitAt_off** (hX : X.Ok) (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ w ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : $\mathcal{O}$ X.E) ∉ w.asIdeal) (j : ℕ) (b) (p q : PtAt R sel mu cst u j) : ((splitAt sel mu cst L Tp hX hPr u hQ j).off b).r p q ↔ SameOffPt sel mu cst u p q

theorem **frsF_eq** (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ w ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : $\mathcal{O}$ X.E) ∉ w.asIdeal) {j : ℕ} {p q : PtAt R sel mu cst u j} (h : SameOffPt sel mu cst u p q) : frsF sel mu p.1.1 = frsF sel mu q.1.1

theorem **exists_above_ne** (hX : X.Ok) (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ w ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : $\mathcal{O}$ X.E) ∉ w.asIdeal) (j : ℕ) (b) (P : Pr.vT u) (p : PtAt R sel mu cst u j) : ∃ q, Above (tdatAt sel mu cst L Tp u j) b P p q ∧ SameOffPt sel mu cst u p q ∧ q ≠ p

theorem **act_of_zeta** (R : B.Real i sp M) (ζ : CommGroup.torsion (D.what₂.LL)ˣ) (b) (g : (B.I.S i).G b) (h1 : (B.I.S i).gam b g = 1) (h2 : (B.I.S i).chr b g = ζ.1) (d) (y : M.Val) : R.zeta ζ • R.pt b d = R.hom b g • R.pt b d ∧ M.act (R.zeta ζ) y = M.act (R.hom b g) y

theorem **src_of_kerIs** {sp' : Split (B.J.A (i + 1)) B.J.T} {M' : PtModel B.ch (i + 1)} (R : B.Real i sp M) (R' : B.Real (i + 1) sp' M') (K : KerPt M M') (k : B.Ker i) (h : Dat.KerIs R R' K k) (b b') (g : (B.I.S i).G b) (d d') (F : B.V i) (y : M.Val) (hy : y.1 = B.ev i F) : B.ev (i + 1) (k b b' (g • d) d' ((B.I.S i).rho B.emb (B.lam i) b g F)) = B.ev (i + 1) (k b b' d d' F)

theorem **kerEq_unique** {D : D7} {u : D.what₂.Word} {B : Dat D u} {i : ℕ} (k k' : B.Ker i) (h : Dat.KerEq k) (h' : Dat.KerEq k') : k = k'

theorem **mv_act** {E K : Type} [Field E] [NumberField E] [StarRing E] [Field K] [NumberField K] [StarRing K] {n : ℕ} (S : GStep E K n) (h : S.Free) (b) (g : S.G b) (d) : S.mv b (g • d) = g * S.mv b d

theorem **mv_rep** {E K : Type} [Field E] [NumberField E] [StarRing E] [Field K] [NumberField K] [StarRing K] {n : ℕ} (S : GStep E K n) (h : S.Free) (c : S.Cl) : S.mv c.1 (S.rep c) = 1

theorem **polOf_one** {ι : Type} [Fintype ι] {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (o : Frames G) (ho : ∀ τ, o.1 τ = 1) (s e : ι → ℤ) (h : ∀ τ, 0 ≤ s τ + e τ) (P : MvPolynomial (ι × Fin n × Fin n) ℂ) : MvPolynomial.aeval (id1 (R

theorem **gen** {G G' K V V' : Type} [Group G] [Group G'] [Field K] [AddCommGroup V] [AddCommGroup V'] [Module K V] [Module K V'] (ρ : G →* Module.End K V) (ρ' : G' →* Module.End K V') (k : V →ₗ[K] V') (g a : G) (g' a' : G') : ρ' (g' * a') ∘ₗ k ∘ₗ ρ (g * a)⁻¹ = ρ' g' ∘ₗ (ρ' a' ∘ₗ k ∘ₗ ρ a⁻¹) ∘ₗ ρ g⁻¹

theorem **kk_equiv** (i : ℕ) (h0 : (B.I.S i).Free) (h1 : (B.I.S (i+1)).Free) (b : (B.I.S i).β) (b' : (B.I.S (i+1)).β) (g : (B.I.S i).G b) (g' : (B.I.S (i+1)).G b') (d : (B.I.S i).D b) (d' : (B.I.S (i+1)).D b') : kk B i b b' (g • d) (g' • d') = (B.I.S (i + 1)).rho B.emb (B.lam (i + 1)) b' g' ∘ₗ kk B i b b' d d' ∘ₗ (B.I.S i).rho B.emb (B.lam i) b g⁻¹

theorem **kk_rep** (i : ℕ) (h0 : (B.I.S i).Free) (h1 : (B.I.S (i+1)).Free) (c : (B.I.S i).Cl) (c' : (B.I.S (i+1)).Cl) (F : B.V i) : kk B i c.1 c'.1 ((B.I.S i).rep c) ((B.I.S (i + 1)).rep c') F = MvPolynomial.aeval (id1 (R

theorem **kk_kmap** (i : ℕ) (hs : ∀ τ, 0 ≤ shiftOf (B.lam i) τ + B.ch.C.sSharp i τ) (ht : ∀ τ, shiftOf (B.lam (i+1)) τ = − B.ch.C.sDiv i τ) (c : (B.I.S i).Cl) (c' : (B.I.S (i+1)).Cl) (F : B.V i) : ((I' B).chain B.emb D.what₂.embLL (B' B).hI (D.hcW u)).Kmap i c c' (B.ev i F) = B.ev (i+1) (MvPolynomial.aeval (id1 (R

theorem **kk_eq** (i : ℕ) (h0 : (B.I.S i).Free) (h1 : (B.I.S (i+1)).Free) (hs : ∀ τ, 0 ≤ shiftOf (B.lam i) τ + B.ch.C.sSharp i τ) (ht : ∀ τ, shiftOf (B.lam (i+1)) τ = − B.ch.C.sDiv i τ) : Dat.KerEq (B

theorem **eval_extremal** {R : Type} [CommRing R] {n : ℕ} (lam : DomWeight n) : MvPolynomial.eval (fun v : Fin n × Fin n => if v.1 = v.2 then (1 : R) else 0) (extremal R n lam) = 1

theorem **ext_one** (i : ℕ) : MvPolynomial.aeval (id1 (R

theorem **nonzero** (i : ℕ) (h0 : (B.I.S i).Free) (h1 : (B.I.S (i+1)).Free) (c : (B.I.S i).Cl) (c' : (B.I.S (i+1)).Cl) (hs : ∀ τ, 0 ≤ shiftOf (B.lam i) τ + B.ch.C.sSharp i τ) (ht : ∀ τ, shiftOf (B.lam (i+1)) τ = − B.ch.C.sDiv i τ) : ∃ k : (B' B).Ker i, Dat.KerEq k ∧ k ≠ 0

theorem **shifts** {X : Fields} {R : Ranks} (hc : X.Tau → HCParam R) (i : ℕ) (hi : i < R.m) (τ : X.Tau) : 0 ≤ shiftOf (GlobalTheory.Mem.Col2.lamT hc i) τ + ⌊(hc τ).e ⟨i, hi⟩ - (R.N (i + 1) : ℚ) / 2⌋ ∧ shiftOf (GlobalTheory.Mem.Col2.lamT hc (i+1)) τ = -⌊(hc τ).e ⟨i, hi⟩ - (R.N i : ℚ) / 2⌋

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ExistsBranchingCoordinatesIff

theorem **branchingLaw_ofRecord_iff** (P : Formalism.MainNotions.product_first_zeroth_parts.Pend D u) : ArchimedeanSignatureMatching (Pend.ofRecord P) ↔ Formalism.MainNotions.product_first_zeroth_parts.ArchimedeanSignatureMatching P

theorem **branchingCoordinates_iff** (P : Pend (own D u)) : Formalism.MainNotions.product_first_zeroth_parts.ArchimedeanSignatureMatching P.toRecord ↔ ArchimedeanSignatureMatching P

theorem **exists_branchingCoordinates_iff** : (∃ P : Pend (own D u), ArchimedeanSignatureMatching P) ↔ ∃ P : Formalism.MainNotions.product_first_zeroth_parts.Pend D u, Formalism.MainNotions.product_first_zeroth_parts.ArchimedeanSignatureMatching P

theorem **Pend.interlaceTr** (P : Pend X) (τ : D.what₂.Tau) : Formalism.MainNotions.PrimeDegreeVariant.Interlace (Nm_succ D.R) (lamTr X τ) (sgTr X τ)

theorem **delSq_lamTr** (X : Ch D u) (R : Type) [CommRing R] (τ : D.what₂.Tau) : delSq R _ (lamTr X τ) = delSq R _ (X.fm.lam D.R.m τ)

theorem **delSq_sgTr** (X : Ch D u) (R : Type) [CommRing R] (τ : D.what₂.Tau) : delSq R _ (sgTr X τ) = delSq R _ (X.fm.sg τ)

theorem **normFock_wSh** {n : ℕ} (l : DomWeight n) (c : ℤ) : normFock (wSh l c) = normFock l

theorem **degZ_tr** (X : Ch D u) (τ : D.what₂.Tau) : Formalism.MainNotions.PrimeDegreeVariant.degZ (lamTr X τ) (sgTr X τ) = X.fm.deg τ

theorem **fockF_self_eq** {σ : Type} (f : MvPolynomial σ ℂ) : Formalism.REAL.fockF f f = ((∑ m ∈ f.support, Complex.normSq (f.coeff m) * ((∏ k ∈ m.support, (m k).factorial : ℕ) : ℝ) : ℝ) : ℂ)

theorem **fockF_self_pos** {σ : Type} {f : MvPolynomial σ ℂ} (hf : f ≠ 0) : 0 < (Formalism.REAL.fockF f f).re

private lemma **extremal_ne_zero_global_twisting** {n : ℕ} (l : DomWeight n) : extremal ℂ n l ≠ 0

theorem **normFock_self_pos** {n : ℕ} (l : DomWeight n) {f : MvPolynomial (Fin n × Fin n) ℂ} (hf : f ≠ 0) : 0 < (normFock l f f).re

theorem **ArchimedeanSignatureMatching.quot_le** {P : Pend X} (h : ArchimedeanSignatureMatching P) (τ : D.what₂.Tau) {f : MvPolynomial (Fin (D.R.N D.R.m) × Fin (D.R.N D.R.m)) ℂ} (hf : f ∈ delSq ℂ _ (X.fm.lam D.R.m τ)) (h0 : f ≠ 0) : Real.sqrt ((normFock (X.fm.sg τ) (P.br τ f) (P.br τ f)).re / (normFock (X.fm.lam D.R.m τ) f f).re) ≤ Formalism.MainNotions.PrimeDegreeVariant.brBd (Nm_succ D.R) (lamTr X τ) (sgTr X τ)

theorem **ArchimedeanSignatureMatching.bddAbove** {P : Pend X} (h : ArchimedeanSignatureMatching P) (τ : D.what₂.Tau) : BddAbove ((fun f => Real.sqrt ((normFock (X.fm.sg τ) (P.br τ f) (P.br τ f)).re / (normFock (X.fm.lam D.R.m τ) f f).re)) '' {f | f ∈ delSq ℂ _ (X.fm.lam D.R.m τ) ∧ f ≠ 0})

private lemma **normFock_self_re_nonneg'** {n : ℕ} (l : DomWeight n) (f : MvPolynomial (Fin n × Fin n) ℂ) : 0 ≤ (normFock l f f).re

theorem **ArchimedeanSignatureMatching.br_le** {P : Pend X} (h : ArchimedeanSignatureMatching P) (τ : D.what₂.Tau) {f : MvPolynomial (Fin (D.R.N D.R.m) × Fin (D.R.N D.R.m)) ℂ} (hf : f ∈ delSq ℂ _ (X.fm.lam D.R.m τ)) : (normFock (X.fm.sg τ) (P.br τ f) (P.br τ f)).re ≤ X.fm.beOn τ ^ 2 * (normFock (X.fm.lam D.R.m τ) f f).re

theorem **ArchimedeanSignatureMatching.beOn_pos** {P : Pend X} (h : ArchimedeanSignatureMatching P) (τ : D.what₂.Tau) : 0 < X.fm.beOn τ

theorem **ArchimedeanSignatureMatching.beOn_le** {P : Pend X} (h : ArchimedeanSignatureMatching P) (τ : D.what₂.Tau) : X.fm.beOn τ ≤ Formalism.MainNotions.PrimeDegreeVariant.brBd (Nm_succ D.R) (lamTr X τ) (sgTr X τ)

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ExistsIntegralPendingObjectSymmetricFormsLevel

theorem **gspan_smul** (d : 𝒪 D.what₂.LL) {p} (hp : p ∈ GSpan D lam) : D.what₂.embLL (d : D.what₂.LL) • p ∈ GSpan D lam

theorem **sLat_good** {a : ℝ} (ha : a ≠ 0) {L : Submodule (𝒪 K) (κ → K)} (h1 : L.FG) (h2 : Submodule.span K (L : Set (κ → K)) = ⊤) : (sLat K κ a L).Good

theorem **sLat_norm** {a : ℝ} (ha : a ≠ 0) (L : Submodule (𝒪 K) (κ → K)) (σ : K →+* ℂ) (x : κ → K) : (Formalism.HermitianLatticesTheory.majAt (sLat K κ a L).b (sLat K κ a L).X (InfinitePlace.mk σ) x x).re = (∑ c, (((a ^ 2 : ℝ) : ℂ) • ipUnit) (stdEmb κ σ x c) (stdEmb κ σ x c)).re

theorem **nfAll_pos** {ι : Type} [Fintype ι] {n : ℕ} (l : ι → DomWeight n) {p : MvPolynomial (ι × Fin n × Fin n) ℂ} (hp : p ≠ 0) : 0 < (nfAll l p p).re

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ExistsPowPrincipalFractionalIdeal

@[simp] theorem **coe_imEquiv** (x : LL) : ((imEquiv ι x : imM ι) : ℂ) = ι x

@[simp] theorem **rhoOf_imEquiv** (s : LL →+* ℂ) (x : LL) : rhoOf ι s (imEquiv ι x) = s x

@[simp] theorem **rhoEquiv_apply** (s : LL →+* ℂ) : rhoEquiv ι s = rhoOf ι s

theorem **sum_rhoOf** (f : (imM ι →ₐ[ℚ] ℂ) → ℝ) : ∑ s : LL →+* ℂ, f (rhoOf ι s) = ∑ ρ, f ρ

theorem **card_emb** : Fintype.card (LL →+* ℂ) = Module.finrank ℚ (imM ι)

theorem **lamConj_cases** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) : ∃ (cc' : Ideal (𝒪 D.host.K)) (δ' : ℂ), Crv D cc' δ' ∧ (lamConj M ρ (lamOf D cc δ) = lamOf D cc' δ' ∨ (lamConj M ρ (lamOf D cc δ)).map (starRingEnd ℂ).toAddMonoidHom = lamOf D cc' δ')

theorem **isLat_lamConj** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) : GlobalTheory.ThetaConj.IsLat (lamConj M ρ (lamOf D cc δ))

theorem **area_lamConj_pos** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) : 0 < area (lamConj M ρ (lamOf D cc δ))

theorem **isThetaZero_unique_lattice** (h : Crv D cc δ) {t t' : ℂ} (ht : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t) (ht' : GL3Theta.IsTwoTorsionPointUnkilled D (lamOf D cc δ) t') : t - t' ∈ lamOf D cc δ

theorem **isThetaZero_unique** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) {t t' : ℂ} (ht : GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t) (ht' : GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t') : t - t' ∈ lamConj M ρ (lamOf D cc δ)

theorem **exists_isThetaZero** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) : ∃ t, GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t

theorem **thetaZero_eq_of_isThetaZero** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) {t t' : ℂ} (ht : GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t) (ht' : GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t') : GL3Theta.th0 (lamConj M ρ (lamOf D cc δ)) t = GL3Theta.th0 (lamConj M ρ (lamOf D cc δ)) t'

theorem **normSq1_eq_nrm1Sq** (Λ : AddSubgroup ℂ) (G : ℂ → ℂ) : GL3Theta.normSq1 Λ G = nrm1Sq Λ G

theorem **normSq1_nonneg** (Λ : AddSubgroup ℂ) (G : ℂ → ℂ) : 0 ≤ GL3Theta.normSq1 Λ G

theorem **normSq_thetaZero_pos** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) {t : ℂ} (ht : GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t) : 0 < GL3Theta.normSq1 (lamConj M ρ (lamOf D cc δ)) (GL3Theta.th0 (lamConj M ρ (lamOf D cc δ)) t)

theorem **ellT_ne_bot** (hM : ∀ τ x, embT D τ x ∈ M) (τ : D.what₂.Tau) : ellT D M τ ≠ ⊥

theorem **ne_zero_of_span_eq** (hM : ∀ τ x, embT D τ x ∈ M) {τ : D.what₂.Tau} {a : M} (ha : Submodule.span (𝒪 M) {a} = ellT D M τ) : a ≠ 0

theorem **exp_neg_logMet** {Λ : AddSubgroup ℂ} {ρ : M →ₐ[ℚ] ℂ} (hΛ : 0 < area (lamConj M ρ Λ)) {a : M} (ha : a ≠ 0) : Real.exp (-logMet M Λ ρ a) = area (lamConj M ρ Λ) / (Real.pi * ‖ρ a‖ ^ 2)

theorem **logMet_mul** {Λ : AddSubgroup ℂ} {ρ : M →ₐ[ℚ] ℂ} (hΛ : 0 < area (lamConj M ρ Λ)) {c a : M} (hc : c ≠ 0) (ha : a ≠ 0) : logMet M Λ ρ (c * a) = 2 * Real.log ‖ρ c‖ + logMet M Λ ρ a

theorem **index_span_mul** (c : 𝒪 M) {a : M} (ha : a ≠ 0) : ((Submodule.span (𝒪 M) {c • a}).toAddSubgroup.addSubgroupOf (Submodule.span (𝒪 M) {a}).toAddSubgroup).index = Ideal.absNorm (Ideal.span {c})

theorem **sum_log_norm** (c : 𝒪 M) (hc : c ≠ 0) : ∑ ρ : M →ₐ[ℚ] ℂ, Real.log ‖ρ (c : M)‖ = Real.log (Ideal.absNorm (Ideal.span {c})) : ℕ)

theorem **meanLogOm_eq_of_gen** (hM : ∀ τ x, embT D τ x ∈ M) {Λ : AddSubgroup ℂ} (hΛ : ∀ ρ : M →ₐ[ℚ] ℂ, 0 < area (lamConj M ρ Λ)) {τ : D.what₂.Tau} {a : M} (ha : Submodule.span (𝒪 M) {a} = ellT D M τ) : meanLogOm D Λ M τ = -(∑ ρ : M →ₐ[ℚ] ℂ, logMet M Λ ρ a) / (Module.finrank ℚ M : ℝ)

theorem **perTie_iff** : PerTie D cc δ ↔ ∃ a, ExtraTie D cc δ D.what₂.embLL D.gx a

theorem **th0NormSq_eq** {M : IntermediateField ℚ ℂ} (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) {t : ℂ} (ht : GL3Theta.IsTwoTorsionPointUnkilled D (lamConj M ρ (lamOf D cc δ)) t) : th0NormSq D M (lamOf D cc δ) ρ = GL3Theta.normSq1 (lamConj M ρ (lamOf D cc δ)) (GL3Theta.th0 (lamConj M ρ (lamOf D cc δ)) t)

theorem **th0NormSq_pos** {M : IntermediateField ℚ ℂ} (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) M) (ρ : M →ₐ[ℚ] ℂ) : 0 < th0NormSq D M (lamOf D cc δ) ρ

theorem **extraTie_gxOf** (h : Crv D cc δ) (hM : DataIn D (tor0 D cc δ) (imM ι)) {a : D.what₂.Tau → imM ι} (ha : ∀ τ, Submodule.span (𝒪 (imM ι)) {a τ} = ellT D (imM ι) τ) : ExtraTie D cc δ ι (gxOf h hM a) a

theorem **perTie_withGx_iff** (G : GlobalExtra D.what₂.LL D.what₂.Tau) : PerTie (withGx D G) cc δ ↔ ∃ a, ExtraTie D cc δ D.what₂.embLL G a

theorem **exists_extraTie** (h : Crv D cc δ) (M : IntermediateField ℚ ℂ) [NumberField M] (hM : DataIn D (tor0 D cc δ) M) (ha : ∀ τ, ∃ a : M, Submodule.span (𝒪 M) {a} = ellT D M τ) : ∃ (G : GlobalExtra M D.what₂.Tau) (a : D.what₂.Tau → imM (M.val : M →+* ℂ)), ExtraTie D cc δ (M.val : M →+* ℂ) G a

theorem **ne_zero** (τ : D.what₂.Tau) : a τ ≠ 0

theorem **om_eq** (τ : D.what₂.Tau) (s : LL →+* ℂ) : G.Om τ s = area (lamConj (imM ι) (rhoOf ι s) (lamOf D cc δ)) / (Real.pi * ‖rhoOf ι s (a τ)‖ ^ 2)

theorem **log_om** (τ : D.what₂.Tau) (s : LL →+* ℂ) : Real.log (G.Om τ s) = -logMet (imM ι) (lamOf D cc δ) (rhoOf ι s) (a τ)

theorem **th_sq** (s : LL →+* ℂ) : G.thOne s ^ 2 = th0NormSq D (imM ι) (lamOf D cc δ) (rhoOf ι s)

theorem **logOmMean_eq** (τ : D.what₂.Tau) : G.logOmMean τ = meanLogOm D (lamOf D cc δ) (imM ι) τ

theorem **crv** : Crv D cc δ

theorem **data** : DataIn D (tor0 D cc δ) (coefM D)

theorem **logOmMean_eq** (τ : D.what₂.Tau) : D.gx.logOmMean τ = meanLogOm D (lamOf D cc δ) (coefM D) τ

theorem **th_sq** (s : D.what₂.LL →+* ℂ) : D.gx.thOne s ^ 2 = th0NormSq D (coefM D) (lamOf D cc δ) (rhoOf D.what₂.embLL s)

theorem **sum_dual_family** {r : ℕ} {a a' : Fin r → D.host.E} (h : ∀ t : D.host.E, t = ∑ k, (Algebra.trace D.host.K D.host.E (t * D.host.conj (a' k))) • a k) (τ σ : D.what₂.Tau) : ∑ k, embT D τ (D.host.conj (a' k)) * embT D σ (a k) = if τ = σ then 1 else 0

theorem **mem_ellT_iff** {M : IntermediateField ℚ ℂ} [NumberField M] (hM : ∀ τ x, embT D τ x ∈ M) (τ : D.what₂.Tau) (x : M) : x ∈ ellT D M τ ↔ ∀ y ∈ D.host.AA0, IsIntegral ℤ ((x : ℂ) * embT D τ (D.host.conj y))

theorem **exists_pow_principal_fractionalIdeal** (D : D7) : ∃ (h : ℕ) (α : D.host.E), 0 < h ∧ α ≠ 0 ∧ ((D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E)) ^ h = Submodule.span (𝒪 D.host.E) {α}

theorem **span_inv_eq_ellT** {M : IntermediateField ℚ ℂ} [NumberField M] (hM : ∀ τ x, embT D τ x ∈ M) {h : ℕ} {α : D.host.E} (hh : 0 < h) (hα : α ≠ 0) (hA : ((D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E)) ^ h = Submodule.span (𝒪 D.host.E) {α}) (τ : D.what₂.Tau) (γ : M) (hγ : (γ : ℂ) ^ h = embT D τ (D.host.conj α)) : Submodule.span (𝒪 M) {γ⁻¹} = ellT D M τ

theorem **exists_field_gen** {cc : Ideal (𝒪 D.host.K)} {δ : ℂ} (h : Crv D cc δ) (M₀ : IntermediateField ℚ ℂ) [FiniteDimensional ℚ M₀] : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M), M₀ ≤ M ∧ DataIn D (tor0 D cc δ) M ∧ ∀ τ, ∃ a : M, Submodule.span (𝒪 M) {a} = ellT D M τ

theorem **exists_extraTie_of_crv** {cc : Ideal (𝒪 D.host.K)} {δ : ℂ} (h : Crv D cc δ) (M₀ : IntermediateField ℚ ℂ) [FiniteDimensional ℚ M₀] : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M), M₀ ≤ M ∧ ∃ (G : GlobalExtra M D.what₂.Tau) (a : D.what₂.Tau → imM (M.val : M →+* ℂ)), ExtraTie D cc δ (M.val : M →+* ℂ) G a

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ExistsRecordDat

theorem **val_tauAt** {X : Fields} {Pr : Primes X} (hPr : Pr.Ok) (u : Pr.Word) (hX : X.AbelianUnramifiedExtensionConditions) {w : HeightOneSpectrum (𝒪 X.K)} (hw : ((Pr.T u : ℕ) : 𝒪 X.K) ∈ w.asIdeal) (P : Pr.vT u) (s : Bool) (hs : s = true ↔ seenC Pr u w) (x : X.E) : X.emb (tauAt hPr u w P) (if s then x else star x) ∈ GlobalTheory.LvT.Inh.val0 w ↔ P.1.valuation X.E x ≤ 1

theorem **sees_ofOk** (X : PreX B.toPre) (I : ChainIn B.fields.E B.fields.K B.fields.Tau B.R) (J : AtIn I) (Q : PerIn I) (Z : FmIn J (assemble B C W V S N G).what₂.places) (hI : I.Ok) (hJ : J.Ok) (hQ : Q.Ok) (hT : Q.Over J) (eT : J.T = B.T u) (eS : ∀ P ∈ J.ST, P ∈ (B.primes X).vT u) (eZ : ∀ w (P : J.ST), Z.tauOf w P = tauAt (GlobalTheory.BasePrimes.primes_ok B X).1 u w (P.1, eS P.1 P.2)) (eF : ∀ w, Z.seenP w = true ↔ seenC (B.primes X) u w) : (ofOk B C W V S N G u I J Q Z hI hJ hQ hT).Sees

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ExistsRowValNeZero

theorem **mem_MtildeL_iff** (hX : X.Ok) (hd : Disjoint Pr.vGs (QE sel (mu : Chr X.F X.E X.K))) (z : MtildeV W Ch sel mu P) : z ∈ MtildeL W Ch sel mu P ↔ ∀ c : (stepX Ch sel mu P).Cl, z.1 (PeriodOnStep.rowPt Ch sel mu P c) ∈ nabAt X.emb W.sgTrue ((stepX Ch sel mu P).rep c).1.1.deco.1.N

theorem **span_toMtP_gen** (hX : X.Ok) (σ : X.K →+* ℂ) : Submodule.span ℂ (Set.range fun z : MtildeL W Ch sel mu P => toMtP W Ch sel mu P σ z.1) = ⊤

theorem **rowVal_add** {u : B.Word} {a : I.ι} {G G' : (I.per l u a).Mt} (h : RowVal I l u a G) (h' : RowVal I l u a G') : RowVal I l u a (G + G')

theorem **rowVal_smul** {u : B.Word} {a : I.ι} (r : 𝒪 B.LL) {G : (I.per l u a).Mt} (h : RowVal I l u a G) : RowVal I l u a (B.embLL (r : B.LL) • G)

theorem **rowVal_rowForm_iff** (h : I.Ok) (u : B.Word) (a : I.ι) (z : MtildeV (Pr

theorem **mtE_toMtP** (h : I.Ok) (u : B.Word) (a : I.ι) (z : MtildeV (Pr

theorem **span_toMtP** (a : I.ι) : Submodule.span ℂ (Set.range fun z : MtildeL (Pr

theorem **span_rowVal** (h : I.Ok) (u : B.Word) (a : I.ι) : Submodule.span ℂ {G | RowVal I l u a G} = ⊤

theorem **not_rowVal_smul** {u : B.Word} {a : I.ι} {G : (I.per l u a).Mt} (hG : RowVal I l u a G) (h0 : (I.perIn u a).rd B.fields.emb B.embLL G ≠ 0) {t : ℂ} (ht : t ∉ Set.range B.embLL) : ¬ RowVal I l u a (t • G)

theorem **rowVal_carrier** (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) {V : Type} [AddCommGroup V] [Module ℂ V] (M : Rows.OLat B.O (B.embLL.comp B.OtoLL) V) (e : (I.per l u a).Mt ≃ₗ[ℂ] V) (H : ∀ v ∈ M.gen, RowVal I l u a (e.symm v)) : ∀ F ∈ M.carrier, RowVal I l u a (e.symm F)

theorem **exists_later** (Z : TC.Later B (I.cols l)) (M : ∀ μ, Rows.OLat B.O (B.embLL.comp B.OtoLL) (Z.W.Amb μ)) (e0 : ∀ u, ((I.cols l).mem u).P.Mt ≃ₗ[ℂ] Z.W.Amb (I.cols l).mu0) (eT : ∀ u μ, ((I.cols l).PTw u μ).Mt ≃ₗ[ℂ] Z.W.Amb μ) : ∃ Z' : TC.Later B (I.cols l), Z'.W = reRows Z.W M e0 eT

theorem **rowInt_iff** (Z : TC.Later B (I.cols l)) : RowInt I l Z ↔ (∀ u, ∀ F ∈ (Z.W.Mtilde (I.cols l).mu0).carrier, RowVal I l u I.mu0 ((Z.W.eMt0 u).symm F)) ∧ ∀ u (a : (I.cols l).Tup), ∀ F ∈ (Z.W.Mtilde a).carrier, RowVal I l u a ((Z.W.eMtTw u a).symm F)

theorem **exists_rowInt** (h : I.Ok) (Z : TC.Later B (I.cols l)) : ∃ Z' : TC.Later B (I.cols l), RowInt I l Z' ∧ Z'.W.Amb = Z.W.Amb ∧ Z'.W.Drows = Z.W.Drows

theorem **rd_ne_zero** {u : B.Word} {a : I.ι} {G : (I.per l u a).Mt} (h0 : G ≠ 0) : (I.perIn u a).rd B.fields.emb B.embLL G ≠ 0

theorem **exists_ne_zero** {O : Type} [CommRing O] {φ : O →+* ℂ} {V : Type} [AddCommGroup V] [Module ℂ V] [Nontrivial V] (M : Rows.OLat O φ V) : ∃ v ∈ M.carrier, v ≠ 0

theorem **exists_rowVal_ne_zero** {Z : TC.Later B (I.cols l)} (hZ : RowInt I l Z) (u : B.Word) (a : (I.cols l).Tup) [Nontrivial (Z.W.Amb a)] : ∃ G : (I.per l u a).Mt, G ≠ 0 ∧ RowVal I l u a G

theorem **exists_not_rowInt** {Z : TC.Later B (I.cols l)} (hZ : RowInt I l Z) (a : (I.cols l).Tup) [Nontrivial (Z.W.Amb a)] : ∃ Z' : TC.Later B (I.cols l), ¬ RowInt I l Z' ∧ Z'.W.Amb = Z.W.Amb

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.FamilyClassSevenDutiesAndStableLattices

theorem **familyClass_sevenDuties_and_stableLattices** (QO : ∀ D : D7, Obj D → Prop) {eps : ℝ} (Fm : FamP eps) (h : cls7 QO Fm) : Duty7 Fm ∧ MtStableF Fm

theorem **familyClass_points_satisfy_objectProperty** (QO : ∀ D : D7, Obj D → Prop) {eps : ℝ} (Fm : FamP eps) (h : cls7 QO Fm) : classOf (ptO QO) Fm

theorem **TwAllAt.twAt** {D : D7} {C : NChoice D} {T : ThData D} (h : TwAllAt D C T) : LocalQuadraticParts.TWAt D C T

theorem **twAllAt_of_untwisted** {D : D7} (C : NChoice D) (T : ThData D) (h : ∀ w, D.what₂.muAt w = D.what₂.mu0) : TwAllAt D C T

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.GeometricWeakTiesOfStrong

theorem **geometricWeakTies_of_strong** {CF : ℝ} (h : RL.GeometricBoundCondition Q Lk CF) : GeometricWeakCompatibility Q Lk

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.GoodReductionCondition

theorem **At.of_row** {p : Prm hL} {H : ∀ 𝒜 : FamClass, RawSel Bundle 𝒜 → Prop} (h : Row (IsGoodE p.s) (IsLinked p) H) : At p fun S L => Row S L H

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.GoodTopBot

theorem **_root_.GlobalTheory.Alc.good_top_bot** (R : Type) [CommRing R] (r n : ℕ) : _root_.GlobalTheory.Alc.Good (⊤ : Submodule R (MvPolynomial (Fin r × Fin n) R)) ∧ _root_.GlobalTheory.Alc.Good (⊥ : Submodule R (MvPolynomial (Fin r × Fin n) R))

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.InfiniteGroup

lemma **mem_ratInt** {q : ℕ} [Fact q.Prime] (x : ℚ) : x ∈ ratInt q ↔ ‖(x : ℚ_[q])‖ ≤ 1

lemma **toZp_coe** {q : ℕ} [Fact q.Prime] (x : ratInt q) : ((toZp q x : ℤ_[q]) : ℚ_[q]) = ((x : ℚ) : ℚ_[q])

lemma **mem_std** (x : Fin n → Lc.K) : x ∈ Lc.std n ↔ ∀ a, x a ∈ Lc.O

lemma **inv_mv** (g : GL (Fin n) Lc.K) (x : Fin n → Lc.K) : mvO g⁻¹ (Matrix.mulVec (g : Matrix (Fin n) (Fin n) Lc.K) x) = x

theorem **isAct_actPP** (g : GL (Fin n) Lc.K) (p : LocF.PP Lc n) : LocF.PP.IsAct g p (actPP g p)

private lemma **mulVec_mvO_inv** (g : GL (Fin n) Lc.K) (y : Fin n → Lc.K) : Matrix.mulVec (g : Matrix (Fin n) (Fin n) Lc.K) (mvO g⁻¹ y) = y

theorem **isAct_unique** {g : GL (Fin n) Lc.K} {p p₁ p₂ : LocF.PP Lc n} (h₁ : LocF.PP.IsAct g p p₁) (h₂ : LocF.PP.IsAct g p p₂) : p₁ = p₂

theorem **exists_dil** (a : F×) (p : LocF.PP Lc n) : ∃ p', LocF.PP.IsDil (a : F) p p'

lemma **q_inv_not_mem** (Lc : LocF q F) : ((q : Lc.K))⁻¹ ∉ Lc.O

lemma **dq_inv_mv** (a : Fin n) (x : Fin n → Lc.K) (b : Fin n) : mvO (dq Lc a)⁻¹ x b = if b = a then (q : Lc.K) * x b else x b

private lemma **stdPair_f_apply** (a : Fin n) {x : Fin n → Lc.K} (h : x ∈ Lc.std n) : (stdPair Lc a).1.f x = Lc.res (x a, (mem_std x).1 h a)

theorem **inc_std** (a : Fin n) : (stdPair Lc a).Inc (actPP (dq Lc a) (stdPair Lc a))

private lemma **iter_mvO_dq_inv_apply** (a : Fin n) (k : ℕ) (y : Fin n → Lc.K) (b : Fin n) : (⇑(mvO (dq Lc a)⁻¹))^[k] y b = if b = a then (q : Lc.K) ^ k * y b else y b

private lemma **mem_iter_actPP_stdPair_iff** (a : Fin n) (k : ℕ) (y : Fin n → Lc.K) : y ∈ ((actPP (dq Lc a))^[k] (stdPair Lc a)).1.L ↔ (⇑(mvO (dq Lc a)⁻¹))^[k] y ∈ Lc.std n

theorem **infinite_PP** (Lc : LocF q F) (a : Fin n) : Infinite (LocF.PP Lc n)

theorem **infinite_group** {X : Ch D u} {i : ℕ} (M : PtModel X i) (c : X.C.cls i) (v : X.A.vT) : Infinite M.G

theorem **empty_kernels** (X : Ch D u) [∀ i, IsEmpty (X.C.cls i)] : ∃ (Lv : LevTie X) (Kt : KerTie Lv), ∀ i h, (Kt i h).TgtEq ∧ (Kt i h).DilLaw ∧ (Kt i h).UKLaw

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.IntegralGLSuccEmbeddingMemParahoricTimesPrincipa

theorem **exists_int_approx** (c : ℕ) {a : E} (ha : IntAt P a) : ∃ r : 𝒪 E, IntAt P (((qOf P : E) ^ c)⁻¹ * (a - algebraMap (𝒪 E) E r))

theorem **mem_levExt** {g : GL (Fin N) κ} : g ∈ (levExt h l).H ↔ ∃ j ∈ l.H, diagEmb κ h j = g

theorem **diagEmb_mem_levExt** {j : GL (Fin n) κ} : diagEmb κ h j ∈ (levExt h l).H ↔ j ∈ l.H

theorem **levExt_lam** {g : GL (Fin N) κ} (hg : g ∈ (levExt h l).H) {j : GL (Fin n) κ} (hj : j ∈ l.H) (e : g = diagEmb κ h j) : (levExt h l).lam ⟨g, hg⟩ = l.lam ⟨j, hj⟩

theorem **lev_lam_congr** {l l' : Lev κ K n} (e : l = l') {g g' : GL (Fin n) κ} (e' : g = g') (hg : g ∈ l.H) (hg' : g' ∈ l'.H) : l.lam ⟨g, hg⟩ = l'.lam ⟨g', hg'⟩

theorem **mem_parahoricTimesPrincipalCongruence_iff** {k : GLint E N P} : k ∈ K1 hN l c ↔ ∃ j ∈ levJ l, ∃ κ ∈ prinCong N P c, k = extI hN P j * κ

theorem **integralGLSuccEmbedding_mem_parahoricTimesPrincipalCongruence** {j : GLint E n P} (hj : j ∈ levJ l) : extI hN P j ∈ K1 hN l c

theorem **principalCongruence_le_levelGroup** : prinCong N P c ≤ K1 hN l c

theorem **reductionModPrime_mem_levelExtendDiagonal_of_parahoric** {k : GLint E N P} (hk : k ∈ K1 hN l c) : redGL N P k ∈ (levExt (by omega : n ≤ N) l).H

theorem **parahoricTimesPrincipalCongruence_character_eq** {j : GLint E n P} (hj : j ∈ levJ l) {κ : GLint E N P} (hκ : κ ∈ prinCong N P c) {k : GLint E N P} (hk : k ∈ (levelParahoricTimesPrincipalCongruence hN l hc).H) (e : k = extI hN P j * κ) : (levelParahoricTimesPrincipalCongruence hN l hc).lam ⟨k, hk⟩ = l.lam ⟨redGL n P j, hj⟩

theorem **levelGroup_character_isOfFinOrder** (k : (levelParahoricTimesPrincipalCongruence hN l hc).H) : IsOfFinOrder ((levelParahoricTimesPrincipalCongruence hN l hc).lam k)

theorem **TopDepth.pos** {R : Ranks} {M : ℕ} (c : TopDepth R M) : 0 < c.1

theorem **topGSD_ok** (hX : X.Ok) (P : LevFam X.E X.K R.r Pr.vGs) (u : Pr.Word) {M : ℕ} (c : TopDepth R M) : (topGSD W Ch sel mu P u c).Ok

theorem **tilOk_of** (hPr : Pr.Ok) (u : Pr.Word) (hQ : ∀ P ∈ QE sel (mu : Chr X.F X.E X.K), (Pr.T u : 𝒪 X.E) ∉ P.asIdeal ∧ ∀ i, (Pr.g i : 𝒪 X.E) ∉ P.asIdeal) : TilOk Pr sel mu u

theorem **tilOk_of_avoid** (hPr : Pr.Ok) (hsel : IsSel X sel) (u : Pr.Word) (hS : Pr.T u ∈ S ∧ ∀ i, Pr.g i ∈ S) : TilOk Pr sel mu u

theorem **tilFr_isLiftFr** (H : TilOk Pr sel mu u) (d : RowPt Pr R sel mu) (P : rowSG Pr sel mu) : IsLiftFr (tilQ sel mu u d).1 P (tilFr H d P)

theorem **frsF_tilTop_g** (H : TilOk Pr sel mu u) (d : RowPt Pr R sel mu) (P : rowSG Pr sel mu) (h : P.1 ∈ Pr.vGs) : frsF sel mu (tilTop H d) P = frsT sel mu u d P

theorem **frsF_tilTop_Q** (H : TilOk Pr sel mu u) (d : RowPt Pr R sel mu) (P : rowSG Pr sel mu) (h : P.1 ∉ Pr.vGs) : frsF sel mu (tilTop H d) P = tilFr H d P

theorem **frsF_mem** {j : ℕ} (p : ChPt (R

theorem **rowTy_inG** (w : Pr.vGs) : rowTy Ch sel mu P (inG sel mu w) = rowTyG Ch P w

theorem **rowTy_inQ** (H : TilOk Pr sel mu u) (w : QE sel (mu : Chr X.F X.E X.K)) : rowTy Ch sel mu P (inQ Pr sel mu w) = rowTyQ R (mu : Chr X.F X.E X.K) (QE sel (mu : Chr X.F X.E X.K)) w

theorem **row_mem_g** (g : (rowTy Ch sel mu P).inv.grp) (w : Pr.vGs) : g.1 (inG sel mu w) ∈ (rowTyG Ch P w).H

theorem **row_mem_Q** (H : TilOk Pr sel mu u) (g : (rowTy Ch sel mu P).inv.grp) (w : QE sel (mu : Chr X.F X.E X.K)) : g.1 (inQ Pr sel mu w) ∈ (rowTyQ R (mu : Chr X.F X.E X.K) (QE sel (mu : Chr X.F X.E X.K)) w).H

theorem **redGL_rowPass** (g : FormG (rowTy Ch sel mu P).inv) (d : RowPt Pr R sel mu) (w : rowSG Pr sel mu) : redGL _ w.1 (rowPass g d w) = g.2.1 w

theorem **rowPass_mem** (H : TilOk Pr sel mu u) (g : FormG (rowTy Ch sel mu P).inv) (d : RowPt Pr R sel mu) (w : QE sel (mu : Chr X.F X.E X.K)) : rowPass g d (inQ Pr sel mu w) ∈ levJ (rowTyQ R (mu : Chr X.F X.E X.K) (QE sel (mu : Chr X.F X.E X.K)) w)

theorem **tilEl_smul** (H : TilOk Pr sel mu u) {c : ℕ} (hc : 0 < c) (g : FormG (rowTy Ch sel mu P).inv) (d : RowPt Pr R sel mu) : tilEl H hc g d • tilTop H d = tilTop H (g • d)

theorem **tilEl_chr** (H : TilOk Pr sel mu u) {c : ℕ} (hc : 0 < c) (g : FormG (rowTy Ch sel mu P).inv) (d : RowPt Pr R sel mu) : (topLevG Ch P).chr (tilEl H hc g d).2.1 * (topLevQ R sel mu hc).chr (tilEl H hc g d).2.2 = ((rowTy Ch sel mu P).inv.chr g.2)⁻¹

theorem **tilEl_rho** (H : TilOk Pr sel mu u) {M : ℕ} (c : TopDepth R M) (g : FormG (rowTy Ch sel mu P).inv) (d : RowPt Pr R sel mu) : (topGSD W Ch sel mu P u c).ρ (tilEl H c.pos g d) = evRho W Ch sel mu P u g

theorem **atNL_last** (x : Sp X.E (R.N R.m)) : atNL X.E (R.N R.m) (R.N R.m - 1) x = x (Br.iL (N_m_succ R))

theorem **cond_of_ite_ne_zero** {c : Prop} {i : Decidable c} {a : X.K} (h : @ite _ c i a 0 ≠ 0) : c

theorem **castAdd_single** {n' c' : ℕ} (i : Fin n') : LinearMap.funLeft X.E X.E (Fin.castAdd c') (Pi.single (Fin.castAdd c' i) 1) = (Pi.single i 1 : Sp X.E n')

theorem **entB_top** (hm : R.m = m2 + 2) (q : ℕ) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) (p p' : ℕ) : LocE.entB q R.k (cOf R (m2 + 1)) (R.N (m2 + 1)) 0 (R.N R.m - 1) (fun i => i + 1) (fun i => i + 1) (prXp X.E R ∘ₗ y ∘ₗ LinearMap.funLeft X.E X.E (Fin.castAdd (cOf R (m2 + 1)))) p p' = if h : p < R.N R.m - 1 ∧ p' < R.N (m2 + 1) then y (Pi.single ⟨p', h.2⟩ 1) (Br.iC (N_m_succ R) ⟨p, h.1⟩) else 0

theorem **rank_add_mul_eq_penultimate_rank** (hm : R.m = m2 + 2) : R.r + (R.m - 1) * R.t = R.N (m2 + 1)

theorem **locTop_supp** (hm : R.m = m2 + 2) {P : rowSG Pr sel mu} (hP : P.1 ∈ Pr.vGs) {y : HomE X.E (R.N (m2 + 1)) (R.N R.m)} {y' : Sp X.E (R.N (m2 + 1))} (h : locTop Ch sel mu cst m2 P y y' ≠ 0) (s : Fin (R.N (m2 + 1))) : y (Pi.single s 1) ∈ stdAt X.E (R.N R.m) P.1

theorem **locTop_congr** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∈ Pr.vGs) {y z : HomE X.E (R.N (m2 + 1)) (R.N R.m)} {y' : Sp X.E (R.N (m2 + 1))} (hy' : ∀ i, IntAt P.1 ((qOf P.1 : X.E) ∗ y' i)) (h : ∀ s l, CongQ P.1 (y (Pi.single s 1) l) (z (Pi.single s 1) l)) : locTop Ch sel mu cst m2 P y y' = locTop Ch sel mu cst m2 P z y'

theorem **locTop_inv** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∈ Pr.vGs) (hq : ((qOf P.1 : ℕ) : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) (κ : GLint X.E (R.N R.m) P.1) (hκ : redGL _ P.1 κ = 1) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) {y' : Sp X.E (R.N (m2 + 1))} (hy' : ∀ i, IntAt P.1 ((qOf P.1 : X.E) ∗ y' i)) : locTop Ch sel mu cst m2 P (κ.1.toLinearMap ∘ₗ y) y' = locTop Ch sel mu cst m2 P y y'

theorem **repY_int** {j : ℕ} (p : ChPt (R

theorem **locTop_tilFr** (H : TilOk Pr sel mu u) (d : RowPt Pr R sel mu) (P : rowSG Pr sel mu) (f : Sp X.E (R.N (m2 + 1)) ≃ₗ[X.E] Sp X.E (R.N (m2 + 1))) (x : HomE X.E (R.N (m2 + 1)) (R.N R.m)) (y' : Sp X.E (R.N (m2 + 1))) : locTop Ch sel mu cst m2 P (inFr f (tilFr H d P) x) y' = PhiTop Ch mu cst.sharp m2 P.1 ((d.1.liftFr P).symm.toLinearMap ∘ₗ prXp X.E R ∘ₗ x ∘ₗ f.toLinearMap ∘ₗ LinearMap.funLeft X.E X.E (Fin.castAdd (cOf R (m2 + 1)))) ∗ ftY sel mu cst P ((((Pr.piT u : 𝒪 X.E) : X.E)⁻¹ • atNL X.E (R.N R.m) (R.N R.m − 1)) ∘ₗ x ∘ₗ f.toLinearMap) y'

theorem **XiTop_tilTop** (hX : X.Ok) (hPr : Pr.Ok) (hm : R.m = m2 + 2) (hc : cst.Ok) (H : TilOk Pr sel mu u) (p : ChPt (R

theorem **k2top_tilTop** (hX : X.Ok) (hPr : Pr.Ok) (hm : R.m = m2 + 2) (hc : cst.Ok) (H : TilOk Pr sel mu u) {T : ChTh W sel mu u m2} (hT : TameFr T.top) (p : ChPt (R

theorem **topEquiv_iff_on** : TopEquiv W Ch sel mu cst L T b hY P c ↔ TopEquivOn W Ch sel mu cst L T b hY P c

theorem **TopEquiv.cp2Equiv** (h : TopEquiv W Ch sel mu cst L T b hY P c) (hX : X.Ok) (hPr : Pr.Ok) (hm : R.m = m2 + 2) (hc : cst.Ok) (H : TilOk Pr sel mu u) (hT : TameFr T.top) : Cp2Equiv W Ch sel mu cst L T b hY P

theorem **levelGroup_lastColumn_congruence** {E K : Type} [Field E] [NumberField E] [StarRing E] [Field K] {n N : ℕ} (hN : N = n + 1) {P : HeightOneSpectrum (𝒪 E)} (l : Lev (𝒪 E ⧸ P.asIdeal) K n) (c : ℕ) {k : GLint E N P} (hk : k ∈ K1 hN l c) : ((qOf P : E) ^ c)⁻¹ • (k.1 (Pi.single (Br.iL hN) 1) − Pi.single (Br.iL hN) 1) ∈ stdAt E N P

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.IsRationalUnitaryDeterminantMap

theorem **CMData0.of_cmData** {L : Type} [Field L] [NumberField L] {F : Type} [Field F] [NumberField F] [Algebra F L] {c : L ≃ₐ[F] L} {I : Type} {φ : I → L →+* ℂ} (h : CMData F c φ) : CMData0 F c φ

theorem **cmData0_negTwo** : CMData0 ℚ negTwoConj (fun w : InfinitePlace NegTwoField => w.embedding)

theorem **wtOK_zero** : WtOK H 0

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.JetsAllOfColIn

theorem **exists_cst_one** {Z : Dat} (hZ : Z.Ok) {sel : HeightOneSpectrum (𝒪 Z.X.F) → HeightOneSpectrum (𝒪 Z.X.E)} (hsel : IsSel Z.X sel) (S : Set ℕ) {m0 : Fin Z.R.m → Formalism.L2.Host.HChar Z.X.E} {eta : Formalism.L2.Host.HChar Z.X.E} (ht : Z.Ch.Tied Z.sig0 (fun j τ => (Z.W.row τ).e j) m0 eta) (hsq : ∀ k, ∃ r : Z.X.K, r ^ 2 = (Z.Pr.g k : Z.X.K)) (hω : ∀ k, ∃ ω : Z.X.K, Z.sig0 ω = Complex.exp (2 ∗ Real.pi ∗ Complex.I / (Z.Pr.g k : ℂ))) : ∃ cst : LocCst Z.X (rowSG Z.Pr sel (1 : Tup Z.X.F Z.X.E Z.X.K S)), cst.Ok ∧ MuqTieG (Pr

theorem **exists_hyp** : ∃ (Z : Dat) (H : D7) (s : Seam H Z.X) (δ : ℂ) (χ : TaylorLogMetric.IdE H → ℂ) (cD : H.what₂.Tau → ℂ) (sel : HeightOneSpectrum (𝒪 Z.X.F) → HeightOneSpectrum (𝒪 Z.X.E)) (S : Set ℕ) (cst : LocCst Z.X (rowSG Z.Pr sel (1 : Tup Z.X.F Z.X.E Z.X.K S))), 2 ≤ Fintype.card Z.Pr.Word ∧ Nonempty (Formalism.L2.Ker.Base H ⊤ δ) ∧ ∀ u, Hyp Z s ⊤ δ χ cD sel S 1 cst u

theorem **hyp_of_colIn** (B : GlobalTheory.Mem.Base) (C : Cols B) (W : RowsRest B C) (V : DivStage B C W) (S : SStage B C W) (N : NStage B C W) (G : GJStage B C) {X : PreX B.toPre} (I : ColIn B X) (h : I.Ok) (a : I.ι) (u : B.Word) : Hyp I.dat (seamOf B C W V S N G) ⊤ I.J.δ I.chi I.J.cD I.sel I.S (I.tup a) (I.cst a) u

theorem **jetsAll_of_colIn** (B : GlobalTheory.Mem.Base) (C : Cols B) (W : RowsRest B C) (V : DivStage B C W) (S : SStage B C W) (N : NStage B C W) (G : GJStage B C) {X : PreX B.toPre} (I : ColIn B X) (h : I.Ok) (hJ : I.JetsIn) (a : I.ι) (u : B.Word) {i : ℕ} (hi : i < B.R.m) : JetsAll (Pr

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.JoinEquivEqAtReference

theorem **conjFn_apply** {G : ι → Matrix (Fin n) (Fin n) ℂ} (hG : ∀ τ, (G τ).IsHermitian) (s : ι → ℤ) (P : Pol ℂ ι n) (y : Frames G) : conjFn G s P y = starRingEnd ℂ ((∏ τ, ((G τ)ᵀ ∗ (y.1 τ).map (starRingEnd ℂ)).det ^ s τ) ∗ MvPolynomial.eval (fun x : ι × Fin n × Fin n => ((G x.1)ᵀ ∗ (y.1 x.1).map (starRingEnd ℂ)) x.2.1 x.2.2) P)

theorem **conjFn_actP** {G : ι → Matrix (Fin n) (Fin n) ℂ} (hG : ∀ τ, IsUnit (G τ).det) (lam : ι → DomWeight n) (g : ι → GL (Fin n) ℂ) (hg : ∀ τ, (g τ : Matrix (Fin n) (Fin n) ℂ)ᴴ ∗ G τ ∗ g τ = G τ) (P : Pol ℂ ι n) (y y' : Frames G) (h : IsTr (fun τ => (g τ : Matrix (Fin n) (Fin n) ℂ)) y' y) : conjFn G (shiftOf lam) (actP lam g P) y = conjFn G (shiftOf lam) P y'

theorem **conjFn_injective** {G : ι → Matrix (Fin n) (Fin n) ℂ} (hG : ∀ τ, IsUnit (G τ).det) (s : ι → ℤ) : Function.Injective (conjFn G s)

theorem **reGL_symm_mem** {F E : Type} [Field F] [CommRing E] [Algebra F E] (σ : E →ₐ[F] E) (R : Type) [CommRing R] [Algebra F R] {a b : ℕ} (h : a = b) (J : Matrix (Fin a) (Fin a) E) {x} (hx : x ∈ Formalism.UG.UPts σ (J.reindex (finCongr h) (finCongr h)) R) : (reGL h _).symm x ∈ Formalism.UG.UPts σ J R

theorem **reGL_mem** {F E : Type} [Field F] [CommRing E] [Algebra F E] (σ : E →ₐ[F] E) (R : Type) [CommRing R] [Algebra F R] {a b : ℕ} (h : a = b) (J : Matrix (Fin a) (Fin a) E) {x} (hx : x ∈ Formalism.UG.UPts σ J R) : reGL h _ x ∈ Formalism.UG.UPts σ (J.reindex (finCongr h) (finCongr h)) R

theorem **exists_hasMat** (x : PtX D) : ∃ h, HasMat h x.1

theorem **finHom_eq** {h h' : GA D D.sp.X'} {y} (H : HasMat h y) (H' : HasMat h' y) : finHom D.sp.X' h = finHom D.sp.X' h'

theorem **rowPt_eq** {h : GA D D.sp.X'} {x : PtX D} (H : HasMat h x.1) : (finHom D.sp.X').rangeRestrict h = rowPt x

theorem **rowPt_apply** (x : PtX D) (i j) : Lev.swapAE D ((rowPt x).1.1 i j) = x.1.1 (finCongr (rankP D) i) (finCongr (rankP D) j)

theorem **eq_rowPt** {g : Bridge.Gf D.sp.X'} {x : PtX D} (H : ∀ i j, Lev.swapAE D (g.1.1 i j) = x.1.1 (finCongr (rankP D) i) (finCongr (rankP D) j)) : g = rowPt x

theorem **rowPt_bijective** : Function.Bijective (rowPt (D

theorem **levIs_iff_topLev** {u : D.what₂.Word} (K : ClsPTied u) (J : Set (GA D D.sp.X')) (χ : GA D D.sp.X' → ℂ) : LevIs K.lev J χ ↔ TopLev K J χ

theorem **adFun_ext** {F E ι : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [Fintype ι] {σ : E →ₐ[F] E} {n T f : ℕ} {J : Matrix (Fin n) (Fin n) E} {bad : Set (HeightOneSpectrum (𝒪 F))} {S : StepLevel F E σ J T f bad} {emb : ι → E →+* ℂ} {cl : Type}

{y : cl → Formalism.UG.UPts σ J (FiniteAdeleRing (𝒪 F) F)} (hy : Function.Surjective fun c => (Quotient.mk'' (y c) : Cls S)) {g g'} (hg : g ∈ AdFun S emb) (hg' : g' ∈ AdFun S emb) (h : ∀ c, g (y c) = g' (y c)) : g = g'

theorem **exists_isCoord** (h : R.fam_tie) : ∃ co, IsCoord R co

theorem **IsCoord.exists_equiv** {co} (h : IsCoord R co) : ∃ e : VV ℂ (lamX D) ≃ₗ[ℂ] R.cX.W, ∀ v, co (e v) = v.1

theorem **IsCoord.injective** {co} (h : IsCoord R co) : Function.Injective co

theorem **conjForm_eq_model** (hj : R.models_jX) (co) (t : T) (f : A.fam.amb.FR t) (x : PtX D) : conjForm R co t f x = conjVal D (co ((R.HX t).e (R.jX t f) (rowPt x)))

theorem **strictAnti_unique** {N : ℕ} {S : Finset ℚ} (hS : S.card ≤ N) {c c' : Fin N → ℚ} (hc : StrictAnti c) (hc' : StrictAnti c') (m : ∀ i, c i ∈ S) (m' : ∀ i, c' i ∈ S) : c = c'

theorem **shW_eq_lamX** (u : D.what₂.Word) (τ : D.what₂.Tau) (sg : DomWeight (D.R.N D.R.m − 1)) (h : ∀ i, (sg.w i : ℚ) = sgHC D.R (D.hcW u τ) i − rhoVec (D.R.N D.R.m − 1) i) (a : Fin D.sp.X'.rank) : (shW sg ((D.R.m * (D.R.t + 1) / 2 : ℕ) : ℤ)).w (Fin.cast (rankP D) a) = (lamX D τ).w a

theorem **sgT_eq_lamX** {u : D.what₂.Word} {K : ClsPTied u} (V : ValPTied u K) (τ : D.what₂.Tau) (a : Fin D.sp.X'.rank) : (sgT u τ).w (Fin.cast (rankP D) a) = (lamX D τ).w a

theorem **shiftOf_sgT** {u : D.what₂.Word} {K : ClsPTied u} (V : ValPTied u K) : shiftOf (sgT u) = shiftOf (lamX D)

theorem **exists_coord** : ∃ co : Coord p, IsCoord p.R.toRealDat co

theorem **gramP_eq_one** : gramP D = 1

theorem **conjVal_eq** {u : D.what₂.Word} {K : ClsPTied u} (V : ValPTied u K) (P : Pol ℂ D.what₂.Tau D.sp.X'.rank) : conjVal D P = fnOf _ (shiftOf (sgT u)) 1 (MvPolynomial.map (starRingEnd ℂ) (MvPolynomial.rename (Prod.map id (Prod.map (Fin.cast (rankP D)) (Fin.cast (rankP D)))) P))

theorem **conjVal_injective** : Function.Injective (conjVal D)

theorem **span_jA** (t : p.P.T) {μ : D.what₂.Tup} (h : p.P.I.tup t = μ) : Submodule.span ℂ (Set.range fun i => (Eq.rec (motive

theorem **MtJoinBy.jM_eq** (h : MtJoinBy p Cs tw co e₀ e) (h' : MtJoinBy p Cs tw co e₀' e') (u : D.what₂.Word) (F : D.what₂.Amb D.what₂.mu0) : (Cs.forms u).jM ((e₀ u).symm F) = (Cs.forms u).jM ((e₀' u).symm F)

theorem **MtJoinBy.jM_eq_tw** (h : MtJoinBy p Cs tw co e₀ e) (h' : MtJoinBy p Cs tw co e₀' e') (u : D.what₂.Word) {μ : D.what₂.Tup} (hu : Used0 D μ) (hμ : μ ≠ D.what₂.mu0) (F : D.what₂.Amb μ) : tw.ptE.jM u μ ((e u μ).symm F) = tw.ptE.jM u μ ((e' u μ).symm F)

theorem **jM_injective** (H : Formalism.L2.Rows.EMtTied D Cs tw) (u : D.what₂.Word) : Function.Injective (Cs.forms u).jM

theorem **jM_injective_tw** (H : Formalism.L2.Rows.EMtTied D Cs tw) (u : D.what₂.Word) {μ : D.what₂.Tup} (hμ : μ ≠ D.what₂.mu0) : Function.Injective (tw.ptE.jM u μ)

theorem **MtJoinBy.joinEquiv_eq_at_reference** (h : MtJoinBy p Cs tw co e₀ e) (h' : MtJoinBy p Cs tw co e₀' e') {u : D.what₂.Word} (hj : Function.Injective (Cs.forms u).jM) : e₀ u = e₀' u

theorem **MtJoinBy.eq_tw** (h : MtJoinBy p Cs tw co e₀ e) (h' : MtJoinBy p Cs tw co e₀' e') {u : D.what₂.Word} {μ : D.what₂.Tup} (hu : Used0 D μ) (hμ : μ ≠ D.what₂.mu0) (hj : Function.Injective (tw.ptE.jM u μ)) : e u μ = e' u μ

theorem **eq_one_of_smul** {M N : Type} [AddCommGroup M] [Module ℂ M] [AddCommGroup N] [Module ℂ N] [Nontrivial N] (f : M ≃ₗ[ℂ] N) {c : ℂ} (h : ∀ x, f x = c • f x) : c = 1

theorem **MtJoinBy.not_scale** (h : MtJoinBy p Cs tw co e₀ e) {u : D.what₂.Word} (hj : Function.Injective (Cs.forms u).jM) [Nontrivial (D.what₂.Amb D.what₂.mu0)] {c : ℂ} (hc : c ≠ 1) (hs : ∀ G, e₀' u G = c • e₀ u G) : ¬ MtJoinBy p Cs tw co e₀' e'

theorem **MtJoinBy.not_scale_tw** (h : MtJoinBy p Cs tw co e₀ e) {u : D.what₂.Word} {μ : D.what₂.Tup} (hu : Used0 D μ) (hμ : μ ≠ D.what₂.mu0) (hj : Function.Injective (tw.ptE.jM u μ)) [Nontrivial (D.what₂.Amb μ)] {c : ℂ} (hc : c ≠ 1) (hs : ∀ G, e' u μ G = c • e u μ G) : ¬ MtJoinBy p Cs tw co e₀' e'

theorem **MtJoin.topLev** (h : MtJoin p Cs tw) (u : D.what₂.Word) (t : p.P.T) (ht : p.P.I.tup t = D.what₂.mu0) : TopLev (Cs.forms u).clsP (levS _ p.R.βX (rowDat p t)) (Lev.levChi _ p.R.βX (rowDat p t))

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LamIdealOfNotPrimePow

theorem **two_lt_ell** (t : ℝ) : 2 < 𝓕.ell t

theorem **half_lt_re_of_region** {c : ℝ} (hc : c ≤ 1) {s : ℂ} (hs : 1 − c / 𝓕.ell s.im ≤ s.re) : 1 / 2 < s.re

theorem **idealsIn_finite** (g : D.G) (n : ℕ) : (idealsIn D g n).Finite

theorem **coe_aRay_eq_cnt** (g : D.G) {n : ℕ} (hn : n ≠ 0) : (aRay D g n : ℂ) = D.cnt g n

theorem **aRay_eq_card** (g : D.G) (n : ℕ) : aRay D g n = ((idealsIn_finite D g n).toFinset.card : ℝ)

theorem **lamRay_eq_sum** (g : D.G) (n : ℕ) : lamRay D g n = ∑ I ∈ (idealsIn_finite D g n).toFinset, lamIdeal I

theorem **lamIdeal_prime_pow** (P : Ideal (𝒪 K)) (hP : P.IsPrime) (h0 : P ≠ ⊥) (k : ℕ) (hk : 1 ≤ k) : lamIdeal (P ^ k) = Real.log (Ideal.absNorm P)

theorem **lamIdeal_of_not_prime_pow** {I : Ideal (𝒪 K)} (h : ¬ ∃ (P : Ideal (𝒪 K)) (k : ℕ), P.IsPrime ∧ P ≠ ⊥ ∧ 1 ≤ k ∧ I = P ^ k) : lamIdeal I = 0

theorem **lamIdeal_unit** : lamIdeal (⊤ : Ideal (𝒪 K)) = 0

theorem **ao_nonneg** (n : ℕ) : 0 ≤ ao n

theorem **count_odd** (N : ℕ) : |∑ n ∈ Icc 1 N, ao n − N / 2| ≤ 1 / 2

theorem **tsum_odd** (t : ℝ) (ht : 0 < t) : ∑' n : ℕ, ao n * Real.exp (−(t * n)) = Real.exp (−t) / (1 − Real.exp (−t) ^ 2)

theorem **sinh_bound** (t : ℝ) (ht : 0 < t) (h1 : t ≤ 1) : |Real.exp (−t) / (1 − Real.exp (−t) ^ 2) − (1 / 2) / t| ≤ 4 ^ 10 * t

theorem **aChi_fam** (n : ℕ) : fam.aChi 1 n = if Odd n then 1 else 0

theorem **res_fam** : fam.res 1 = 1 / 2

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LatIntZero

theorem **latInt_zero** (ι : ℂ ≃+* ℂ_[q]) (ω : ℂ_[q]) (Λ : AddSubgroup V) : LatInt ι ω Λ (fun _ => 0)

theorem **latInt_one** (ι : ℂ ≃+* ℂ_[q]) (ω : ℂ_[q]) (Λ : AddSubgroup V) : LatInt ι ω Λ (fun _ => 1)

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LevelChainSharpColumnValuationOwnChainIff

theorem **dia_zero** (v : X.A.vT) (a : (X.A.Fv v)ˣ) : dia X 0 v a = X.A.rhoT v a

theorem **Nm_zero** (v : X.A.vT) : Nm X 0 v = X.A.Nm v

theorem **Gs_zero** (v : X.A.vT) : Gs X 0 v = X.A.gauss v

theorem **MramI_zero** : MramI X 0 = X.A.Mram

theorem **KT_apply** (i : ℕ) (φ : Fn X i) : KT X i φ = X.C.K i φ

theorem **interlaceTr** (h : X.fm.SgInterlace) (τ : D.what₂.Tau) : Formalism.MainNotions.PrimeDegreeVariant.Interlace (Nm_succ D.R) (lamTr X τ) (sgTr X τ)

theorem **everyLevelPart_of_full** (h : Conclusion6 X) : EveryLevelLiftProperties X

theorem **admissibleBasis_of_naturalComparison** (cl : letI

theorem **lam_indep** (j : ℕ) (hj : j ≤ D.R.m) (τ : D.what₂.Tau) (a : Fin (D.R.N j)) : (X.fm.lam j τ).w a = (Y.fm.lam j τ).w a

theorem **Wt_indep** (τ : D.what₂.Tau) : Wt X τ = Wt Y τ

theorem **ETpi_indep** : ETpi X = ETpi Y

theorem **aTop_indep** : X.fm.aTop = Y.fm.aTop

theorem **bTop_indep** : X.fm.bTop = Y.fm.bTop

theorem **levelSetting_indep_chain** (hT : X.A.T = Y.A.T) : HidaSetting X ↔ HidaSetting Y

theorem **emb_mem_Fi** (i : ℕ) (x : Vl X i) : X.fm.emb i x ∈ X.fm.Fi i

theorem **upCommute_of_firstProperty** (h : Conclusion1 X) : UComm X

theorem **Ulist_mem** (h : UComm X) (l : List X.A.vT) {φ : Fn X 0} (hφ : φ ∈ X.fm.Fi 0) : Ulist X l φ ∈ X.fm.Fi 0

theorem **Ulist_perm** (h : UComm X) {l l' : List X.A.vT} (p : l.Perm l') {φ : Fn X 0} (hφ : φ ∈ X.fm.Fi 0) : Ulist X l φ = Ulist X l' φ

theorem **Shared.refl** (X : Ch D u) : SharedBy X X (LinearEquiv.refl ℂ _) (LinearEquiv.refl _ _) (Equiv.refl _) (fun _ => RingEquiv.refl _)

theorem **Shared.refl** (X : Ch D u) : Shared X X (LinearEquiv.refl ℂ _) (LinearEquiv.refl _ _)

theorem **Shared.of_eq** {X Y : Ch D u} (h : Y = X) (e : Fn X 0 ≃ₗ[ℂ] Fn Y 0) (eL : Vl X 0 ≃ₗ[D.what₂.LL] Vl Y 0) (he : ∀ φ, HEq (e φ) φ) (hL : ∀ x, Y.fm.emb 0 (eL x) = e (X.fm.emb 0 x)) : Shared X Y e eL

theorem **overT_iff** (w : D.what₂.places.Pl) : OverT Y w ↔ OverT X w

theorem **S_iff** (w : D.what₂.places.Pl) (x : Vl X 0) : eL x ∈ Y.fm.S 0 w ↔ x ∈ X.fm.S 0 w

theorem **Nm_eq** (v : X.A.vT) {φ : Fn X 0} (hφ : φ ∈ X.fm.Fi 0) : e (Nm X 0 v φ) = Nm Y 0 (ev v) (e φ)

theorem **ramSub_iff** (x : Vl X 0) : eL x ∈ Y.fm.ramSub ↔ x ∈ X.fm.ramSub

theorem **Sram_iff** (w : D.what₂.places.Pl) (x : Vl X 0) : eL x ∈ Sram Y w ↔ x ∈ Sram X w

theorem **VVT_eq** {w : D.what₂.places.Pl} (hw : OverT X w) : VVT Y w = (VVT X w).map ev.toEmbedding

theorem **expT_eq** {w : D.what₂.places.Pl} (hw : OverT X w) : expT Y w = expT X w

theorem **ET_eq** {w : D.what₂.places.Pl} (hw : OverT X w) : ET Y w = ET X w

theorem **Ulist_eq** (hU : UComm X) (l : List X.A.vT) {φ : Fn X 0} (hφ : φ ∈ X.fm.Fi 0) : e (Ulist X l φ) = Ulist Y (l.map ev) (e φ)

theorem **Uprod_eq** (hU : UComm X) {w : D.what₂.places.Pl} (hw : OverT X w) {φ : Fn X 0} (hφ : φ ∈ X.fm.Fi 0) : e (Uprod X 0 w φ) = Uprod Y 0 w (e φ)

theorem **Uprod_pow_eq** (hU : UComm X) {w : D.what₂.places.Pl} (hw : OverT X w) (n : ℕ) {φ : Fn X 0} (hφ : φ ∈ X.fm.Fi 0) : e ((Uprod X 0 w ^ n) φ) = (Uprod Y 0 w ^ n) (e φ)

theorem **Nlam_iff** (hU : UComm X) {w : D.what₂.places.Pl} (hw : OverT X w) (x : Vl X 0) : eL x ∈ Nlam Y w ↔ x ∈ Nlam X w

theorem **TwTie.of_copy** (Tw : TwForms D) (μ : D.what₂.Tup) (h : tw Tw u μ = own D u) (he : ∀ φ, HEq (D.e0Tw u μ φ) φ) : TwTie Tw u μ

theorem **Wt_tw** (Tw : TwForms D) (μ : D.what₂.Tup) (τ : D.what₂.Tau) : Wt (tw Tw u μ) τ = Wtau D u τ

theorem **levelSetting_twist_iff** (Tw : TwForms D) (μ : D.what₂.Tup) : HidaSetting (tw Tw u μ) ↔ Formalism.MainNotions.product_first_zeroth_parts.HidaSetting D u

theorem **OverT_tw** (Tw : TwForms D) (μ : D.what₂.Tup) (w : D.what₂.places.Pl) : OverT (tw Tw u μ) w ↔ Formalism.MainNotions.product_first_zeroth_parts.OverT D u w

theorem **TwTie.ET_eq** {Tw : TwForms D} {μ : D.what₂.Tup} (h : TwTie Tw u μ) {w : D.what₂.places.Pl} (hw : Formalism.MainNotions.product_first_zeroth_parts.OverT D u w) : ET (tw Tw u μ) w = EatT D u w

theorem **TwTie.Nlam_iff** {Tw : TwForms D} {μ : D.what₂.Tup} (h : TwTie Tw u μ) (hU : UComm (own D u)) {w : D.what₂.places.Pl} (hw : Formalism.MainNotions.product_first_zeroth_parts.OverT D u w) (x : ((D.fm u).lat 0).V) : Tw.e0 u μ x ∈ Nlam (tw Tw u μ) w ↔ x ∈ LT.Nlam D u w

theorem **TwTie.Nlam_eq_map** {Tw : TwForms D} {μ : D.what₂.Tup} (h : TwTie Tw u μ) (hU : UComm (own D u)) {w : D.what₂.places.Pl} (hw : Formalism.MainNotions.product_first_zeroth_parts.OverT D u w) : Nlam (tw Tw u μ) w = (LT.Nlam D u w).map ((Tw.e0 u μ).toLinearMap.restrictScalars (Ow D w))

theorem **TwTie.sharpBound_twist_iff** {Tw : TwForms D} {μ : D.what₂.Tup} (h : TwTie Tw u μ) (hU : UComm (own D u)) : Conclusion5Sharp (tw Tw u μ) ↔ TupleChainSharpColumnValuation Tw u μ

theorem **Alcove_own** : Alcove (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.Alcove D u

theorem **levelSetting_chain_iff** : HidaSetting (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.HidaSetting D u

theorem **upDownIncidence_chain_iff** : UpDownIncidenceInverse (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.UpDownIncidenceInverse D u

theorem **stabilizersTrivial_chain_iff** : admS (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.admS D u

theorem **RoundUp_own** : RoundUp (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.RoundUp D u

theorem **RoundDn_own** : RoundDn (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.RoundDn D u

theorem **TrIso_own** : TrIso (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.TrIso D u

theorem **DiaMul_own** : DiaMul (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.DiaMul D u

theorem **admissibleTorus_chain_iff** : admT (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.admT D u

theorem **firstProperty_chain_iff** : Conclusion1 (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.Conclusion1 D u

theorem **secondProperty_chain_iff** : Conclusion2 (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.Conclusion2 D u

theorem **fourthPropertyExtended_chain_iff** : LevelChainUOperatorCommutation (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.LevelChainUOperatorCommutation D u

theorem **levelChainSharpColumnValuation_ownChain_iff** : Conclusion5Sharp (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.Conclusion5Sharp D u

theorem **sixthProperty_chain_iff** : Conclusion6 (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.Conclusion6 D u

theorem **seventhProperty_chain_iff** : division_rows_rational_lattice (own D u) ↔ Formalism.MainNotions.product_first_zeroth_parts.division_rows_rational_lattice D u

theorem **thirdProperty_chain_iff** (C : ℝ) : PointsLemma (own D u) C ↔ Formalism.MainNotions.product_first_zeroth_parts.PointsLemma D u C

theorem **fourthProperty_chain_iff** (C : ℝ) : Conclusion4 (own D u) C ↔ Formalism.MainNotions.product_first_zeroth_parts.Conclusion4 D u C

theorem **fifthProperty_chain_iff** (C : ℝ) : Conclusion5 (own D u) C ↔ Formalism.MainNotions.product_first_zeroth_parts.Conclusion5 D u C

theorem **weightBudget_chain_iff** (L : SizeData₂ D) (C : ℝ) : WeightBudgetBound (own D u) L C ↔ Formalism.MainNotions.product_first_zeroth_parts.WeightBudgetBound D u L C

theorem **OverT_own** (w) : OverT (own D u) w ↔ Formalism.MainNotions.product_first_zeroth_parts.OverT D u w

theorem **BaseDet_own** : BaseDet (own D u) (D.clsLat u) (D.detCls u) ↔ Formalism.MainNotions.product_first_zeroth_parts.admB D u

theorem **Nlam_own** (w) : Nlam (own D u) w = LT.Nlam D u w

theorem **Sram_own** (w) : Sram (own D u) w = LT.Sram D u w

theorem **VVT_own** (w) : VVT (own D u) w = LT.VVT D u w

theorem **expT_own** (w) : expT (own D u) w = LT.expT D u w

theorem **ET_own** (w) : ET (own D u) w = EatT D u w

theorem **Wt_own** (τ) : Wt (own D u) τ = Wtau D u τ

theorem **Uop_own** (i v) : Uop (own D u) i v = LT.Uop D u i v

theorem **Uprod_own** (i w) : Uprod (own D u) i w = LT.Uprod D u i w

theorem **Dop_own** (i v) : Dop (own D u) i v = Formalism.MainNotions.product_first_zeroth_parts.Dop D u i v

theorem **dia_own** (i v a) : dia (own D u) i v a = Formalism.MainNotions.product_first_zeroth_parts.dia D u i v a

theorem **Nm_own** (i v) : Nm (own D u) i v = Formalism.MainNotions.product_first_zeroth_parts.Nm D u i v

theorem **trp_own** (i v) : trp (own D u) i v = Formalism.MainNotions.product_first_zeroth_parts.trp D u i v

theorem **FFop_own** (i v) : FFop (own D u) i v = Formalism.MainNotions.product_first_zeroth_parts.FFop D u i v

theorem **Gs_own** (i v) : Gs (own D u) i v = Formalism.MainNotions.product_first_zeroth_parts.Gs D u i v

theorem **GsD_own** (i v) : GsD (own D u) i v = Formalism.MainNotions.product_first_zeroth_parts.GsD D u i v

theorem **MramI_own** (i) : MramI (own D u) i = Formalism.MainNotions.product_first_zeroth_parts.MramI D u i

theorem **admissibleBasis_chain_iff** : admB (own D u) (D.clsLat u) D.muTil (D.detCls u) ↔ Formalism.MainNotions.product_first_zeroth_parts.admB D u

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LevelGroupNeTopNeBot

theorem **entB_topQ** (hm : R.m = m2 + 2) (q : ℕ) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) (p p' : ℕ) : LocE.entB q R.k (cOf R (m2 + 1)) (R.N (m2 + 1)) 0 (R.N R.m − 1) 0 0 (prXp X.E R ∘ₗ y ∘ₗ LinearMap.funLeft X.E X.E (Fin.castAdd (cOf R (m2 + 1)))) p p' = if h : p < R.N R.m − 1 ∧ p' < R.N (m2 + 1) then y (Pi.single ⟨p', h.2⟩ 1) (Br.iC (N_m_succ R) ⟨p, h.1⟩) else 0

theorem **suppQ_of_ne** (hm : R.m = m2 + 2) {P : rowSG Pr sel mu} (hP : P.1 ∉ Pr.vGs) {y : HomE X.E (R.N (m2 + 1)) (R.N R.m)} {y' : Sp X.E (R.N (m2 + 1))} (h : locTop Ch sel mu cst m2 P y y' ≠ 0) : SuppQ P.1 cst.M y

theorem **move_small** {P : HeightOneSpectrum (𝒪 X.E)} {M e : ℕ} {κ : GLint X.E (R.N R.m) P} (hκ : κ ∈ prinCong _ P (2 * M + e)) {y : HomE X.E (R.N (m2 + 1)) (R.N R.m)} (hy : SuppQ P M y) (s l) : ∃ z, IntAt P z ∧ κ.1 (y (Pi.single s 1)) l = y (Pi.single s 1) l + (qOf P : X.E) ^ (M + e) * z

theorem **suppQ_move** {P : HeightOneSpectrum (𝒪 X.E)} {M e : ℕ} {κ : GLint X.E (R.N R.m) P} (hκ : κ ∈ prinCong _ P (2 * M + e)) {y : HomE X.E (R.N (m2 + 1)) (R.N R.m)} (hy : SuppQ P M y) : SuppQ P M (κ.1.toLinearMap ∘ₗ y)

private theorem **phiQ_congr_of_entB_congr** {E K : Type} [Field E] [NumberField E] [StarRing E] [Field K] {P : HeightOneSpectrum (𝒪 E)} (ξ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (k₀ r c' n' c n : ℕ) {x x' : HomE E (n' + c') (n + c)} (h : ∀ p p', ∃ w, IntAt P w ∧ LocE.entB (qOf P) k₀ c' n' c n 0 0 x' p p' = LocE.entB (qOf P) k₀ c' n' c n 0 0 x p p' + (qOf P : E) * w) (hZ : ∀ p p', ¬ (p < n + c ∧ p' < n' + c') → IntAt P (LocE.entB (qOf P) k₀ c' n' c n 0 0 x p p')) : LocE.PhiQ (qOf P) ξ k₀ r c' n' c n x' = LocE.PhiQ (qOf P) ξ k₀ r c' n' c n x

theorem **locTop_invQ** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∉ Pr.vGs) (hD : 2 * cst.M ≤ cst.D P) (c : TopDepth R cst.M) {κ : GLint X.E (R.N R.m) P.1} (hκ : κ ∈ prinCong _ P.1 c.1) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) {y' : Sp X.E (R.N (m2 + 1))} (hy' : ∀ i, IntAt P.1 ((qOf P.1 : X.E) ^ cst.M * y' i)) : locTop Ch sel mu cst m2 P (κ.1.toLinearMap ∘ₗ y) y' = locTop Ch sel mu cst m2 P y y'

theorem **not_intAt_inv_q** : ¬ IntAt P ((qOf P : E)⁻¹)

private lemma **exists_prinCong_not_mem_stdAt** {c : ℕ} (hc : 0 < c) : ∃ g ∈ prinCong N P c, ((qOf P : E) ^ (c + 1))⁻¹ • (g.1 (Pi.single (Br.iL hN) 1) − Pi.single (Br.iL hN) 1) ∉ stdAt E N P

theorem **levelGroup_succ_lt** {c : ℕ} (hc : 0 < c) : K1 hN l (c + 1) < K1 hN l c

theorem **levelGroup_ne_top_ne_bot** {c : ℕ} (hc : 0 < c) : K1 hN l (c + 1) ≠ ⊤ ∧ K1 hN l c ≠ ⊥

theorem **repY_intQ** {j : ℕ} (p : ChPt (R

theorem **k2top_invQ** (hm : R.m = m2 + 2) (hc : cst.Ok) (hD : ∀ P : rowSG Pr sel mu, P.1 ∉ Pr.vGs → 2 * cst.M ≤ cst.D P) (c : TopDepth R cst.M) {T : ChTh W sel mu u m2} (hT : TameFr T.top) (κ : GLQ X.E (R.N R.m) (QE sel (mu : Chr X.F X.E X.K))) (hκ : ∀ P, κ P ∈ prinCong _ P.1 c.1) (p : ChPt (R

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LevelPrimeColumnValuationsAtBundlePoints

theorem **row_all** {Bd : BdT} {Q : V4.Slot} {Lk : LkT Bd} (aP10 : LevelPrimeColumnValuationsAtBundlePoints Q Lk) (aAtG : ArithmeticConditionAtBundlePoints Q Lk) : Row Q Lk fun 𝒜 _ => All 𝒜

theorem **row_of_all** {Bd : BdT} {Q : V4.Slot} {Lk : LkT Bd} (h : Row Q Lk fun 𝒜 _ => All 𝒜) : LevelPrimeColumnValuationsAtBundlePoints Q Lk ∧ ArithmeticConditionAtBundlePoints Q Lk

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LogCondLeThree

theorem **modRow_host** {s : Scale} {R : Ranks} (H : Host R) (h : Real.log ((H.QS : ℝ) * H.q * H.d) ≤ 3 * s.x ^ (s.eps / 5) * Real.log s.x ^ 2) : ModRow s H.condN H.d

theorem **HostVal.log_cond_le_three** {s : Scale} {C8 : ℝ} {R : Ranks} {H : Host R} (h : HostVal s C8 H) : Real.log ((H.QS : ℝ) * H.q * H.d) ≤ 3 * s.x ^ (s.eps / 5) * Real.log s.x ^ 2

theorem **HostVal.modRow** {s : Scale} {C8 : ℝ} {R : Ranks} {H : Host R} (h : HostVal s C8 H) : ModRow s H.condN H.d

theorem **modRow_of_rowq** {s : Scale} {Y : ℝ} {n q d : ℕ} (hY : PairWin s Y) (hn : 1 ≤ n) (hd : 100 ≤ d) (h : Rowq Y s.eps n q d) : ModRow s (4 * n * q * d) d

theorem **BoxH.realBox** {R : Ranks} {H : Host R} {Y eps : ℝ} (h : BoxH H Y eps) : RealBox H.toHostRaw

theorem **BoxW.realBox** {s : Scale} {R : Ranks} {H : Host R} (h : BoxW s H) : RealBox H.toHostRaw

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.LowerTieTwAt

theorem **rowVal_zero** (u : B.Word) (a : I.ι) : RowVal I l u a 0

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.MemSplitSeqIff

theorem **pairAct_dot** (g : GL (Fin n) R) (x y : (Fin n → R) × (Fin n → R)) : (pairAct R n g x).2 ⬝ᵥ (pairAct R n g y).1 = x.2 ⬝ᵥ y.1

theorem **mem_splitSeq_iff** (l : List ((Fin n → R) × (Fin n → R))) : l ∈ splitSeq R n ↔ ∀ (i j : ℕ) (hi : i < l.length) (hj : j < l.length), l[i].2 ⬝ᵥ l[j].1 = if i = j then 1 else 0

theorem **smul_val** {p : ℕ} (g : GL (Fin n) R) (l : SplitSet R n p) : (g • l).1 = l.1.map (pairAct R n g)

theorem **face_smul** (p : ℕ) (i : Fin (p + 1)) (g : GL (Fin n) R) (l : SplitSet R n (p + 1)) : face R n p i (g • l) = g • face R n p i l

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.NegGram

theorem **neg_rank** (W : HSpace E) : (neg W).rank = W.rank

theorem **neg_gram** (W : HSpace E) : (neg W).gram = -W.gram

theorem **neg_sig** (W : HSpace E) (τ : E →+* ℂ) : (neg W).sig τ = ((W.sig τ).2, (W.sig τ).1)

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.NotLargeBaseConditionDisjoint

theorem **ev_all** {p : Ctx → Prop} (h : ∀ K, p K) : Ev p

theorem **largeBaseProperty_of_eventually** {p : Ctx → ∀ B : Base, Fr B.toPre → TwDat B → Prop} (h : EvLg p) : LargeBaseCondition KP Q fun K X f T _ => p K X.B f T

theorem **largeBaseProperty_of_isAlgebraic** (S₁ S₂ : Ctx → Finset ℂ) (P : Ctx → Finset ℕ) (A : Ctx → ℕ → Finset ℕ) (h₁ : ∀ K, ∀ z ∈ S₁ K, IsAlgebraic ℚ z) (h₂ : ∀ K, ∀ z ∈ S₂ K, IsAlgebraic ℚ z) : LargeBaseCondition KP Q fun K X _ T _ => Has₂ X.B (S₁ K) (S₂ K) ∧ P K ⊆ T.pl ∧ T.Avoids (A K)

theorem **largeBaseProperty_self** : LargeBaseCondition KP Q fun K X f T c => KP X.B f ∧ Q K X f T c

theorem **largeBaseProperty_mono** {R' : V4.Slot} (h : LargeBaseCondition KP Q R) (hR : ∀ K X f T c, R K X f T c → R' K X f T c) : LargeBaseCondition KP Q R'

theorem **exBL_of_exB** (h : ExB Q' Lk) : ExBL KP Q Q' Lk

theorem **not_rowL_false** : ¬ RowL KP Q Q' Lk fun _ _ => False

theorem **not_largeBaseProperty_of_disjoint** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) (hK : GroupRetwisting.BaseCond KP) (hE : V4.ExTrue KP Q) (hd : ∀ K X f T c, Q K X f T c → ¬ R K X f T c) : ¬ LargeBaseCondition KP Q R

theorem **congr_frame** {α : Type*} (g : ∀ P : Pre, Fr P → α) {P P' : Pre} {f : Fr P} {f' : Fr P'} (h : P = P') (hf : HEq f f') : g P f = g P' f'

theorem **cores_of_passes** (hK : GroupRetwisting.BaseCond KP) (h1 : V4.ExTrue KP Q) (h2 : V4.Export Q) (h3 : V4.Retwist Q Q') : EvG fun K => ∃ X f T c, Q' K X f T c

theorem **cores_sAnd** (hK : GroupRetwisting.BaseCond KP) (h1 : V4.ExTrue KP Q) (h2 : V4.Export Q) (h3 : V4.Retwist Q Q') (h : LargeBaseCondition KP Q R) : EvG fun K => ∃ X f T c, sAnd Q' R K X f T c

theorem **not_void** (hH : DC.HostExQ) (hLP : LevelPrime) (h2b : StandardLattice.SiegelMassRatioCM) (hK : GroupRetwisting.BaseCond KP) (h1 : V4.ExTrue KP Q) (h2 : V4.Export Q) (h3 : V4.Retwist Q Q') (h : LargeBaseCondition KP Q R) : ∃ K X f T c, Q' K X f T c ∧ R K X f T c

theorem **baseCond_of_ksCond** (h2b : StandardLattice.SiegelMassRatioCM) (hK : V4.KsCond KP) : GroupRetwisting.BaseCond KP

theorem **exBL_of_false** : ExBL (fun _ _ => False) Q Q' Lk

theorem **exBL_of_disjoint** (hd : ∀ K X f T c, Q' K X f T c → ¬ Q K X f T c) : ExBL KP Q Q' Lk

theorem **largeBaseProperty_conditional** : LargeBaseCondition KP Q fun K X f T c => LgAt KP Q R K → R K X f T c

theorem **exBL_of_good** (h : GroupRetwisting.LargeBaseConditionG KP Q R) (hx : ExB (sAnd Q' R) Lk) : ExBL KP Q Q' Lk

theorem **rowL_of_good** {H : ∀ 𝒜 : FamClass, RawSel Bd 𝒜 → Prop} (h : GroupRetwisting.LargeBaseConditionG KP Q R) (hr : Row (sAnd Q' R) Lk H) : RowL KP Q Q' Lk H

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.NotMtStableOfFree

theorem **goodAllAt_iff** (D : D7) (C : NChoice D) (I : TwIn D) : GoodAllAt D C I ↔ GlobalTheory.TwAll.TwGoodAtAll D C I

theorem **exps** (D : D7) (i : D.ntw.Tw) : ¬ Nat.Coprime (0 : ℤ).natAbs (D.ntw.l i) ∧ Nat.Coprime (-1 : ℤ).natAbs (D.ntw.l i)

theorem **tup_setDiv** (D : D7) (N) (h) : TupUsedAt (setDiv D N h) ↔ TupUsedAt D

theorem **mt_setDiv** (D : D7) (N) (h) : MtStableAt (setDiv D N h) ↔ MtStableAt D

theorem **conj_setDiv** (D : D7) (O : Obj D) (N) (h) : TwConjAt (setDiv D N h) (O.tr N h).nch ↔ TwConjAt D O.nch

theorem **av_setDiv** (D : D7) (N) (h) : AvRec (setDiv D N h) = AvRec D

theorem **ok_setDiv** (D : D7) (N) (h) (I : TwIn D) (i) : TwIn.Ok (D

theorem **one_setDiv** (D : D7) (N) (h) (I : TwIn D) (i b Q k ψ) : TwGoodOne (setDiv D N h) ⟨{I.E with}, I.N0, I.S⟩ i b Q k ψ ↔ TwGoodOne D I i b Q k ψ

theorem **tup_finite** {D : D7} (h : TupUsedAt D) (hf : {w | D.what₂.muAt w ≠ D.what₂.mu0}.Finite) : Finite D.what₂.Tup

theorem **mem_carrier_iff** (M : Rows.OLat O φ V) (x : V) : x ∈ M.carrier ↔ x ∈ AddSubgroup.closure {y | ∃ c : O, ∃ v ∈ M.gen, y = φ c • v}

theorem **smul_mem_sat** (σ : L →+* ℂ) (M : Rows.OLat O φ V) (a : 𝒪 L) (c : O) {v : V} (hv : v ∈ M.gen) : (φ c * σ (a : L)) • v ∈ (sat σ M).carrier

theorem **subset_sat** (σ : L →+* ℂ) (M : Rows.OLat O φ V) : M.carrier ⊆ (sat σ M).carrier

theorem **sat_stable** (σ : L →+* ℂ) (M : Rows.OLat O φ V) (a : 𝒪 L) : ∀ y ∈ (sat σ M).carrier, σ (a : L) • y ∈ (sat σ M).carrier

theorem **range_of_stable** (σ : L →+* ℂ) (M : Rows.OLat O φ V) (hli : LinearIndependent ℂ (fun v : M.gen => (v : V))) {v : V} (hv : v ∈ M.gen) (a : 𝒪 L) (h : ∀ y ∈ M.carrier, σ (a : L) • y ∈ M.carrier) : σ (a : L) ∈ Set.range φ

theorem **mtStable_sat** (D : D7) : MtStableAt (satRec D)

theorem **tup_setMt** (D : D7) (M) : TupUsedAt (setMt D M) ↔ TupUsedAt D

theorem **qd_sat** (D : D7) (h : TupUsedAt D) (hu : ∀ w, D.what₂.muAt w = D.what₂.mu0) (O : Obj (satRec D)) : QD (satRec D) O

theorem **not_mtStable_of_free** (D : D7) (μ : D.what₂.Tup) (hli : LinearIndependent ℂ (fun v : (D.what₂.Mtilde μ).gen => (v : D.what₂.Amb μ))) (hne : (D.what₂.Mtilde μ).gen.Nonempty) (a : 𝒪 D.what₂.LL) (ha : (a : D.what₂.LL) ∉ Set.range D.what₂.OtoLL) : ¬ MtStableAt D

theorem **used_muTw** {D : D7} (h : TupUsedAt D) (w) : DV.Used D (D.what₂.muTw w)

theorem **good_setDiv** (D : D7) (O : Obj D) (N) (h) (N0 S) : GoodAllAt (setDiv D N h) (O.tr N h).nch ⟨twEOf (O.tr N h).xd, N0, S⟩ ↔ GoodAllAt D O.nch ⟨twEOf O.xd, N0, S⟩

theorem **qd_setDiv** (D : D7) (O : Obj D) (N) (h) : QD (setDiv D N h) (O.tr N h) ↔ QD D O

theorem **not_class_of_false** {eps : ℝ} (Fm : FamP eps) : ¬ cls7 (fun _ _ => False) Fm

theorem **twGood_of_twAllF** {eps : ℝ} {Fm : FamP eps} (h : TwAllF Fm) : TwGood Fm

theorem **duty_of_duty7** {eps : ℝ} {Fm : FamP eps} (h : Duty7 Fm) : DC.Duty Fm

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.NullOfFiniteCurve

theorem **null_of_finite_curve** {V K : Type} [AddCommGroup V] [TopologicalSpace V] [IsTopologicalAddGroup V] [LocallyCompactSpace V] [SecondCountableTopology V] [MeasurableSpace V] [BorelSpace V] [TopologicalSpace K] [SecondCountableTopology K] [MeasurableSpace K] [BorelSpace K] (ρ : Measure K) [SFinite ρ] [NullSingletonClass ρ] (hρ : ρ Set.univ ≠ 0) (μ : Measure V) [μ.IsAddHaarMeasure] {Z : Set V} (hZ : MeasurableSet Z) {L : K → V} (hL : Continuous L) (h : ∀ x : V, {t : K | x + L t ∈ Z}.Finite) : μ Z = 0

theorem **qd_smul** (c : R) (x : Fin N → R) : qd R a (c • x) = c ^ 2 * qd R a x

theorem **qd_add** (x y : Fin N → R) : qd R a (x + y) = qd R a x + qd R a y + Bd R a x y

theorem **Bd_symm** (x y : Fin N → R) : Bd R a x y = Bd R a y x

theorem **Bd_add** (x x' y : Fin N → R) : Bd R a (x + x') y = Bd R a x y + Bd R a x' y

theorem **Bd_smul** (c : R) (x y : Fin N → R) : Bd R a (c • x) y = c * Bd R a x y

theorem **Bd_single** (x : Fin N → R) (l : Fin N) (t : R) : Bd R a x (Pi.single l t) = algebraMap F R (2 * a l) * x l * t

theorem **qd_single** (l : Fin N) (t : R) : qd R a (Pi.single l t) = algebraMap F R (a l) * t ^ 2

theorem **Bd_nondeg** (ha : ∀ l, a l ≠ 0) (x : Fin N → R) (h : ∀ y, Bd R a x y = 0) : x = 0

theorem **continuous_qd** [TopologicalSpace R] [IsTopologicalRing R] : Continuous (qd R a)

theorem **continuous_Bd** [TopologicalSpace R] [IsTopologicalRing R] (y : Fin N → R) : Continuous fun x => Bd R a x y

theorem **exists_fin** (ψ : AddChar (AF F) ℂ) [IsTotallyReal F] (hψ : IsCharAF F ψ) (a : Fin N → F) (ha : ∀ l, a l ≠ 0) (hN : 0 < N) (ν : HeightOneSpectrum (𝒪 F)) : ∃ (μ : Measure (Fin N → Fv F v)) (ν : Measure (Fv F v)) (f : Fv F v → Measure (Fin N → Fv F v)), SelfDual μ (fun x y => psiV F ψ v (Bd (Fv F v) a x y)) ∧ SdR (Fv F v) (psiV F ψ v) ν ∧ IsFibreFamily μ (qd (Fv F v) a) id Set.univ ν {c | c ≠ 0} f

theorem **exists_inf** (ψ : AddChar (AF F) ℂ) [IsTotallyReal F] (hψ : IsCharAF F ψ) (a : Fin N → F) (ha : ∀ l, a l ≠ 0) (hN : 0 < N) : ∃ (μ : Measure (Fin N → Ainf F)) (ν : Measure (Ainf F)) (f : Ainf F → Measure (Fin N → Ainf F)), SelfDual μ (fun x y => psiInf F ψ (Bd (Ainf F) a x y)) ∧ SdR (Ainf F) (psiInf F ψ) ν ∧ IsFibreFamily μ (qd (Ainf F) a) id Set.univ ν {c : Ainf F | ∀ w, c w ≠ 0} f

theorem **nonempty_locData** (ψ : AddChar (AF F) ℂ) [IsTotallyReal F] (hψ : IsCharAF F ψ) (a : Fin N → F) (ha : ∀ l, a l ≠ 0) (hN : 0 < N) : Nonempty (LocData ψ a)

theorem **locData_rat** : ∃ ψ : AddChar (AF ℚ) ℂ, IsCharAF ℚ ψ ∧ Nonempty (LocData ψ (fun _ : Fin 6 => (1 : ℚ)))

theorem **exists_isCov1** : ∃ ν : Measure (AF F), IsCov1 ν

theorem **exists_isLam** (ψ : AddChar (AF F) ℂ) [IsTotallyReal F] (hψ : IsCharAF F ψ) : ∃ lam, IsLam ψ lam

theorem **exists_ctx** [hF : IsTotallyReal F] (a : Fin N → F) (ha : ∀ l, a l ≠ 0) (hN : 0 < N) : ∃ ψ lam ν, Ctx ψ lam ν a ∧ Nonempty (LocData ψ a)

theorem **rho_Ub** (ψ : AddChar (AF F) ℂ) (a : Fin N → F) (b : AF F) (x y : Fin N → AF F) (Φ : (Fin N → AF F) → ℂ) : rho ψ a x y (Ub ψ a b Φ) = Ub ψ a b (rho ψ a x (b • x + y) Φ)

theorem **rho_Da** (ψ : AddChar (AF F) ℂ) (a : Fin N → F) (t s : AF F) (h : t * s = 1) (x y : Fin N → AF F) (Φ : (Fin N → AF F) → ℂ) : rho ψ a x y (Da t Φ) = Da t (rho ψ a (t • x) (s • y) Φ)

theorem **count_ratl** (a : Fin N → F) : (Measure.sum fun ξ : {ξ : Fin N → F // ξ ≠ 0} => Measure.dirac (rat (ξ : Fin N → F))) {u | ¬ ∃ β : F, qd (AF F) a u = algebraMap F (AF F) β} = 0

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ObjCOfRecExObjO

theorem **exists_T_of_over** {D : D7} {P : PreCh} {Lv : LevCh P} (h : Over D P Lv) (u : D.what₂.Word) : ∃ u', (D.mem u).A.T = Lv.T u'

theorem **scale_X'_le_of_log** (eps cs : ℝ) (he : 0 < eps) (hc : 0 < cs) : ∃ x0 : ℝ, ∀ (s : Scale) (R : Ranks) (T : ℝ), s.eps = eps → x0 ≤ s.x → RankVal s R → 0 < T → cs * R.m * s.L ≤ Real.log T → s.X' ≤ T

theorem **scale_one_le_w''** (eps : ℝ) (he : 0 < eps) : ∃ x0 : ℝ, ∀ (s : Scale) (R : Ranks), s.eps = eps → x0 ≤ s.x → RankVal s R → 1 ≤ s.w'' R.n

theorem **objC_of_recEx** (QO : ∀ D : D7, Obj D → Prop) (hLP : LevelPrime) (hRec : RecEx QO) : DC.ObjC (DC.withObj (ptO QO))

theorem **objC_of_recEx_objO** (Q : PtProp) (hLP : LevelPrime) (hRec : RecEx (objO Q)) : DC.ObjC (DC.withObj Q)

theorem **not_recEx_false** (hLP : LevelPrime) (hH : DC.HostW) : ¬ RecEx (fun _ _ => False)

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.PairRowIffQdRows

theorem **n_lo** (h : RankVal s R) : s.x ^ (2 + 8 * s.eps) ≤ (R.n : ℝ)

theorem **n_up** (h : RankVal s R) : (R.n : ℝ) ≤ 2 * s.x ^ (2 + 8 * s.eps)

theorem **n_le** (h : RankVal s R) : (R.n : ℝ) ≤ 2 * s.x ^ (2 + 8 * s.eps + s.eps / 100)

theorem **n_le_of_o** (h : RankVal s R) {C : ℝ} (hC : (R.o : ℝ) ≤ C) : (R.n : ℝ) ≤ 2 * C * s.x ^ (2 + 8 * s.eps)

theorem **n_le'** (h : RankVal s R) : (R.n : ℝ) ≤ 2 * s.x ^ ((9 : ℝ) / 4)

theorem **pairWin_self** (s : Scale) : PairWin s s.x

theorem **PairWin.one_lt** {s : Scale} {Y : ℝ} (h : PairWin s Y) : 1 < Y

theorem **PairWin.rpow_ez_le** {s : Scale} {Y : ℝ} (h : PairWin s Y) {c : ℝ} (hc : 0 < c) : s.x ^ (s.ez / c) ≤ Y ^ (s.eps / c)

theorem **PairWin.rpow_le** {s : Scale} {Y : ℝ} (h : PairWin s Y) {a : ℝ} (ha : 0 ≤ a) : Y ^ a ≤ s.x ^ a

theorem **PairWin.log_le** {s : Scale} {Y : ℝ} (h : PairWin s Y) : Real.log Y ≤ Real.log s.x

theorem **PairWin.log_div_le** {s : Scale} {Y : ℝ} (h : PairWin s Y) : Real.log s.x / s.J ≤ Real.log Y

theorem **PairWin.rpow_inv_le** {s : Scale} {Y C : ℝ} (h : PairWin s Y) (hJ : s.J ≤ C) : s.x ^ (1 / C) ≤ Y

theorem **PairWin.z_log_sq_le** {s : Scale} {Y : ℝ} (h : PairWin s Y) : Y ^ (s.eps / 5) * Real.log Y ^ 2 ≤ s.x ^ (s.eps / 5) * Real.log s.x ^ 2

theorem **four_mul_QS_div** {R : Ranks} (H : Host R) : 4 * (H.QS / 4) = H.QS

theorem **pairRow_iff_rowq** (Y eps : ℝ) {R : Ranks} (H : Host R) : PairRow Y eps H ↔ Rowq Y eps (H.QS / 4) H.q H.d

theorem **split_tie** (h : HostVal s C8 H) : ⌊s.x ^ (s.ez / 5)⌋₊ ≤ H.splitBound

theorem **J_le** (h : HostVal s C8 H) : s.J ≤ C8

theorem **o_le** (h : HostVal s C8 H) : (R.o : ℝ) ≤ C8

theorem **QS_le** (h : HostVal s C8 H) : (H.QS : ℝ) ≤ C8 * R.n

theorem **one_le** (h : HostVal s C8 H) : 1 ≤ C8

theorem **eps_div_le_ez** (h : HostVal s C8 H) : s.eps / C8 ≤ s.ez

theorem **log_cond_le** (h : HostVal s C8 H) : Real.log ((H.QS : ℝ) * H.q * H.d) ≤ 2 * s.x ^ (s.eps / 5) * Real.log s.x ^ 2

theorem **d_le** (h : HostVal s C8 H) : (H.d : ℝ) ≤ GlobalTheoryTheory.X2 s.x s.eps

theorem **split** (h : HostVal s C8 H) (p : ℕ) (hp : p.Prime) (hle : (p : ℝ) ≤ s.x ^ (s.ez / 5)) : GlobalTheoryTheory.kronSym (-(H.d : ℤ)) p = 1

theorem **mono** (h : HostVal s C8 H) {C8' : ℝ} (hC : C8 ≤ C8') : HostVal s C8' H

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.PermRepSingle

@[simp] theorem **permRep_single** {X : Type} [MulAction G X] (A : Rep k G) (g : G) (x : X) (a : A.V) : permRep X A g (Finsupp.single x a) = Finsupp.single (g • x) (A.ρ g a)

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.PowEqPowIffModEqOfIsPrimitiveRoot

theorem **isFrob_one** {K : Type} [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)) : IsFrob K p 1

theorem **heckeVal_one** {K : Type} [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)) : heckeVal K (1 : (K ≃ₐ[K] K) →* ℂˣ) p = 1

theorem **absNorm_eq** (p : HeightOneSpectrum (𝒪 ℚ)) : Ideal.absNorm p.asIdeal = (Rat.HeightOneSpectrum.primesEquiv p : ℕ)

theorem **zeta_prod** (s : ℂ) (hs : 1 < s.re) : HasProd (fun 𝔭 : HeightOneSpectrum (𝒪 ℚ) => (1 - heckeVal ℚ (1 : (ℚ ≃ₐ[ℚ] ℚ) →* ℂˣ) 𝔭 * ((Ideal.absNorm 𝔭.asIdeal : ℕ) : ℂ) ^ (-s))⁻¹) (riemannZeta s)

private theorem **algEquiv_self_eq_one** {K : Type} [Field K] (σ : K ≃ₐ[K] K) : σ = 1

private theorem **heckeVal_self_eq_one** {K : Type} [Field K] [NumberField K] (χ : (K ≃ₐ[K] K) →* ℂˣ) (p : HeightOneSpectrum (𝒪 K)) : heckeVal K χ p = 1

private theorem **heckeIdeal_self_eq_one** {K : Type} [Field K] [NumberField K] (χ : (K ≃ₐ[K] K) →* ℂˣ) (I : Ideal (𝒪 K)) : heckeIdeal K χ I = 1

theorem **condNorm_one** {K : Type} [Field K] [NumberField K] (χ : (K ≃ₐ[K] K) →* ℂˣ) : condNorm K χ = 1

theorem **bigQ_rat** : bigQ ℚ ℚ = 1

theorem **pow_eq_pow_iff_modEq_of_isPrimitiveRoot** {R : Type} [CommRing R] [IsDomain R] {ζ : R} {m : ℕ} [NeZero m] (h : IsPrimitiveRoot ζ m) {a b : ℕ} : ζ ^ a = ζ ^ b ↔ a ≡ b [MOD m]

private theorem **mulSupport_heckeVal_pow_count_finite** {K : Type} (L : Type) [Field K] [NumberField K] [Field L] [NumberField L] [Algebra K L] (χ : (L ≃ₐ[K] L) →* ℂˣ) {I : Ideal (𝒪 K)} (hI : I ≠ 0) : (Function.mulSupport fun 𝔭 : HeightOneSpectrum (𝒪 K) => heckeVal L χ 𝔭 ^ (Associates.mk 𝔭.asIdeal).count (Associates.mk I).factors).Finite

theorem **heckeIdeal_mul** (χ : (L ≃ₐ[K] L) →* ℂˣ) {I J : Ideal (𝒪 K)} (hI : I ≠ 0) (hJ : J ≠ 0) : heckeIdeal L χ (I * J) = heckeIdeal L χ I * heckeIdeal L χ J

theorem **heckeIdeal_prime** (χ : (L ≃ₐ[K] L) →* ℂˣ) (𝔮 : HeightOneSpectrum (𝒪 K)) : heckeIdeal L χ 𝔮.asIdeal = heckeVal L χ 𝔮

theorem **heckeIdeal_top** (χ : (L ≃ₐ[K] L) →* ℂˣ) : heckeIdeal L χ 1 = 1

private theorem **galEquivZMod_eq_of_sub_zeta_pow_mem** (m : ℕ) [NeZero m] (F : Type) [Field F] [NumberField F] [IsCyclotomicExtension {m} ℚ F] (𝔭 : HeightOneSpectrum (𝒪 ℚ)) (hp : (Ideal.absNorm 𝔭.asIdeal).Coprime m) (P : Ideal (𝒪 F)) (hPp : P.IsPrime) (hLies : P.LiesOver 𝔭.asIdeal) (σ : F ≃ₐ[ℚ] F) (e : ℕ) (hmem : σ • (IsCyclotomicExtension.zeta_spec m ℚ F).toInteger - (IsCyclotomicExtension.zeta_spec m ℚ F).toInteger ^ e ∈ P) : ((IsCyclotomicExtension.Rat.galEquivZMod m F σ : (ZMod m)ˣ) : ZMod m) = e

theorem **gal_of_isFrob** (𝔭 : HeightOneSpectrum (𝒪 ℚ)) (hp : (Ideal.absNorm 𝔭.asIdeal).Coprime m) {σ : F ≃ₐ[ℚ] F} (h : IsFrob F 𝔭 σ) : ((galEquivZMod m F σ : (ZMod m)ˣ) : ZMod m) = (Ideal.absNorm 𝔭.asIdeal : ℕ)

theorem **unramIn_cyc** (𝔭 : HeightOneSpectrum (𝒪 ℚ)) (hp : (Ideal.absNorm 𝔭.asIdeal).Coprime m) : UnramIn F 𝔭

theorem **heckeVal_galChar** (ψ : DirichletCharacter ℂ m) (𝔭 : HeightOneSpectrum (𝒪 ℚ)) (hp : (Ideal.absNorm 𝔭.asIdeal).Coprime m) : heckeVal F (galChar m F ψ) 𝔭 = ψ (Ideal.absNorm 𝔭.asIdeal : ℕ)

theorem **heckeIdeal_galChar** (ψ : DirichletCharacter ℂ m) (I : Ideal (𝒪 ℚ)) (hI : I ≠ 0) (hp : (Ideal.absNorm I).Coprime m) : heckeIdeal F (galChar m F ψ) I = ψ (Ideal.absNorm I : ℕ)

theorem **absNorm_span_int** (a : ℤ) : Ideal.absNorm (Ideal.span {(a : 𝒪 ℚ)}) = a.natAbs

theorem **exists_int** (α : 𝒪 ℚ) : ∃ a : ℤ, α = a

theorem **int_dvd_of** {a b : ℤ} (h : (a : 𝒪 ℚ) ∣ b) : a ∣ b

theorem **isModulus_galChar** (ψ : DirichletCharacter ℂ m) : IsModulus F (galChar m F ψ) (Ideal.span {((m : ℤ) : 𝒪 ℚ)})

theorem **condNorm_galChar_le** (ψ : DirichletCharacter ℂ m) : condNorm F (galChar m F ψ) ≤ m

theorem **eq_one_of_isModulus** (ψ : DirichletCharacter ℂ m) {𝔪 : Ideal (𝒪 ℚ)} (h : IsModulus F (galChar m F ψ) 𝔪) (c : ℕ) (hc : (1 + Ideal.absNorm 𝔪 * c).Coprime m) : ψ ((1 + Ideal.absNorm 𝔪 * c : ℕ) : ZMod m) = 1

theorem **condNorm_four** (F : Type) [Field F] [NumberField F] [IsCyclotomicExtension {4} ℚ F] (ψ : DirichletCharacter ℂ 4) (hψ : ψ 3 ≠ 1) : condNorm F (galChar 4 F ψ) = 4

theorem **bigQ_four** (F : Type) [Field F] [NumberField F] [IsCyclotomicExtension {4} ℚ F] : bigQ ℚ F = 4

theorem **nums_example** : QNum (Real.exp 1) 5 11 ∧ DNum (Real.exp 1) 5 11 7

theorem **one_le_mult** {N : ℕ} [NeZero N] (χ : DirichletCharacter ℂ N) {ρ : ℂ} (h : ρ ∈ ntZeros χ) : 1 ≤ mult χ ρ

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.RedPDivOfUnit

theorem **CongQ.red** {a b : E} (h : CongQ P a b) : redP P a = redP P b

theorem **CongQ.div_iff** {a b : E} (h : CongQ P a b) : IntAt P (a / (qOf P : E)) ↔ IntAt P (b / (qOf P : E))

theorem **congQ_zero** : CongQ P (0 : E) 0

theorem **unitV_congr** (χ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) {a b : E} (h : redP P a = redP P b) : LocE.unitV χ a = LocE.unitV χ b

theorem **unitV_det_congr** (χ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) {s : ℕ} {A B : Fin s → Fin s → E} (h : ∀ i j, CongQ P (A i j) (B i j)) : LocE.unitV χ (Matrix.det (Matrix.of A)) = LocE.unitV χ (Matrix.det (Matrix.of B))

theorem **reduction_sum_congr** (s : Finset ℕ) {f g : ℕ → E} (h : ∀ i, CongQ P (f i) (g i)) : redP P (∑ i ∈ s, f i) = redP P (∑ i ∈ s, g i)

theorem **PhiG_congr** (sharp : Bool) (k₀ r t j c' n' c n : ℕ) {x y : HomE E (n' + c') (n + c)} (h : ∀ p p', CongQ P (LocE.entB (qOf P) k₀ c' n' c n (fun i => i + 1) (fun i => i + 1) x p p') (LocE.entB (qOf P) k₀ c' n' c n (fun i => i + 1) (fun i => i + 1) y p p')) : LocE.PhiG (qOf P) ψ₀ χ sharp k₀ r t j c' n' c n x = LocE.PhiG (qOf P) ψ₀ χ sharp k₀ r t j c' n' c n y

theorem **PhiGS_congr** (k₀ rS r t j c' n' c n : ℕ) {x y : HomE E (n' + c') (n + c)} (h : ∀ p p', CongQ P (LocE.entB (qOf P) k₀ c' n' c n (fun i => i + 1) (fun i => i + 1) x p p') (LocE.entB (qOf P) k₀ c' n' c n (fun i => i + 1) (fun i => i + 1) y p p')) : LocE.PhiGS (qOf P) ψ₀ χ k₀ rS r t j c' n' c n x = LocE.PhiGS (qOf P) ψ₀ χ k₀ rS r t j c' n' c n y

theorem **entB_zero** (q k₀ c' n' n : ℕ) (hc : c' = 0) (ex : ℕ → ℕ) (x : HomE E (n' + c') (n + 0)) (p p' : ℕ) : LocE.entB q k₀ c' n' 0 n ex ex x p p' = if h : p < n ∧ p' < n' then x (Pi.single (p', by omega) 1) (p, h.1) else 0

theorem **locChar_mul** (q : ℕ) (cq : K) (ξ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) {u : E} (hu : ∀ a, redP P (a * u) = redP P a) (a : E) : LocE.locChar q cq ξ (a * u) = LocE.locChar q cq ξ a

theorem **tameFr_thDistC** (A : Amb X) {a b : ℕ} (Sall SB : Finset (HeightOneSpectrum (𝒪 X.E))) (D : ℕ) (mU : HeightOneSpectrum (𝒪 X.E) → X.Kˣ) (cq : SB → X.K) (ξ : ∀ P : SB, (𝒪 X.E ⧸ P.1.asIdeal)ˣ →* X.Kˣ) (lam : X.Tau → DomWeight a) (lam' : X.Tau → DomWeight b) (n : X.Tau → ℕ) : TameFr (thDistC A Sall SB D sel mU (fun P => LocE.locChar (qOf P.1) (cq P) (ξ P)) lam lam' n)

theorem **tameFr_chTh** (A : Amb X) (u : Pr.Word) (m2 : ℕ) : TameFr (chTh W Ch sel mu cst A u m2).top

theorem **qOf_notMem_sq** (hX : X.Ok) (hPr : Pr.Ok) {w : HeightOneSpectrum (𝒪 X.E)} (hw : w ∈ Pr.vGs) : ((qOf w : ℕ) : 𝒪 X.E) ∉ w.asIdeal ^ 2

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.RedPEqZeroOfDiv

theorem **intAt_zero** : IntAt P (0 : E)

theorem **intAt_one** : IntAt P (1 : E)

theorem **intAt_neg** {a : E} (ha : IntAt P a) : IntAt P (-a)

theorem **intAt_sub** {a b : E} (ha : IntAt P a) (hb : IntAt P b) : IntAt P (a - b)

theorem **intAt_pow** {a : E} (ha : IntAt P a) (k : ℕ) : IntAt P (a ^ k)

theorem **intAt_nat** (k : ℕ) : IntAt P (k : E)

theorem **intAt_sum** {ι : Type} (s : Finset ι) {f : ι → E} (h : ∀ i ∈ s, IntAt P (f i)) : IntAt P (∑ i ∈ s, f i)

theorem **reduction_one₂** : redP P (1 : E) = 1

theorem **reduction_zero** : redP P (0 : E) = 0

theorem **reduction_sub** {a b : E} (ha : IntAt P a) (hb : IntAt P b) : redP P (a - b) = redP P a - redP P b

theorem **reduction_sum₂** {ι : Type} (s : Finset ι) {f : ι → E} (h : ∀ i ∈ s, IntAt P (f i)) : redP P (∑ i ∈ s, f i) = ∑ i ∈ s, redP P (f i)

theorem **det_int** {ι : Type} [Fintype ι] [DecidableEq ι] {A : Matrix ι ι E} (h : ∀ i j, IntAt P (A i j)) : IntAt P A.det ∧ redP P A.det = (A.map (redP P)).det

theorem **qOf_mem** : ((qOf P : ℕ) : 𝒪 E) ∈ P.asIdeal

theorem **qOf_ne_zero** : ((qOf P : ℕ) : E) ≠ 0

theorem **reduction_absNorm_eq_zero** : redP P ((qOf P : ℕ) : E) = 0

theorem **redP_eq_zero_of_div** {a : E} (h : IntAt P (a / (qOf P : E))) : redP P a = 0

theorem **intAt_div_of_redP** (hq : ((qOf P : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) {a : E} (ha : IntAt P a) (h0 : redP P a = 0) : IntAt P (a / (qOf P : E))

theorem **reduction_mul_of_reduction_eq_one** {u : E} (hu0 : u ≠ 0) (hu : IntAt P u) (hu' : IntAt P u⁻¹) (h1 : redP P u = 1) (a : E) : redP P (a * u) = redP P a

theorem **mem_stdAt** {x : Sp E n} : x ∈ stdAt E n P ↔ ∀ i, IntAt P (x i)

theorem **single_mem_stdAt** (j : Fin n) : (Pi.single j 1 : Sp E n) ∈ stdAt E n P

theorem **smul_mem_locP** {N : Submodule (𝒪 E) (Sp E n)} {a : E} (ha : IntAt P a) {x : Sp E n} (hx : x ∈ localization₂ P.asIdeal N) : a • x ∈ localization₂ P.asIdeal N

theorem **apply_eq_sum** {m : ℕ} (f : Sp E n →ₗ[E] Sp E m) (x : Sp E n) (i : Fin m) : f x i = ∑ j, x j * f (Pi.single j 1) i

theorem **glint_mem** (k : GLint E n P) {x : Sp E n} (hx : x ∈ stdAt E n P) : k.1 x ∈ stdAt E n P

theorem **glint_entry** (k : GLint E n P) (i j : Fin n) : IntAt P (k.1 (Pi.single j 1) i)

theorem **redV_apply** (k : GLint E n P) {x : Sp E n} (hx : x ∈ stdAt E n P) : redV P (k.1 x) = (redMat k).mulVec (redV P x)

theorem **redV_single** (j : Fin n) : redV P (Pi.single j 1 : Sp E n) = Pi.single j 1

theorem **redMat_eq** (k : GLint E n P) {A : Matrix (Fin n) (Fin n) (𝒪 E ⧸ P.asIdeal)} (h : ∀ x ∈ stdAt E n P, redV P (k.1 x) = A.mulVec (redV P x)) : redMat k = A

theorem **redGL_val** (k : GLint E n P) : (redGL n P k : Matrix (Fin n) (Fin n) (𝒪 E ⧸ P.asIdeal)) = redMat k

theorem **redV_redGL** (k : GLint E n P) {x : Sp E n} (hx : x ∈ stdAt E n P) : redV P (k.1 x) = (redGL n P k).1.mulVec (redV P x)

theorem **redGL_eq** (k : GLint E n P) {A : GL (Fin n) (𝒪 E ⧸ P.asIdeal)} (h : ∀ x ∈ stdAt E n P, redV P (k.1 x) = A.1.mulVec (redV P x)) : redGL n P k = A

theorem **det_glint** (k : GLint E n P) : IntAt P ((LinearEquiv.det k.1 : Eˣ) : E) ∧ redP P ((LinearEquiv.det k.1 : Eˣ) : E) = (redMat k).det

theorem **redP_mul_det** (k : GLint E n P) (hk : redGL n P k = 1) (a : E) : redP P (a * ((LinearEquiv.det k.1 : Eˣ) : E)) = redP P a

theorem **mem_prinCong** {c : ℕ} {k : GLint E n P} : k ∈ prinCong n P c ↔ ∀ x ∈ stdAt E n P, ((qOf P : E) ^ c)⁻¹ • (k.1 x - x) ∈ stdAt E n P

theorem **redGL_of_prinCong** {c : ℕ} (hc : 0 < c) {k : GLint E n P} (hk : k ∈ prinCong n P c) : redGL n P k = 1

theorem **prinCong_anti** {c c' : ℕ} (h : c ≤ c') : prinCong n P c' ≤ prinCong n P c

theorem **prH_apply** (x : Sp E N) (i : Fin n) : prH E hN x i = x (Br.iC hN i)

theorem **idx_cases** (l : Fin N) : (∃ i, l = Br.iC hN i) ∨ l = Br.iL hN

theorem **ext_prH** {x y : Sp E N} (h1 : prH E hN x = prH E hN y) (h2 : x (Br.iL hN) = y (Br.iL hN)) : x = y

theorem **prH_sumL** (g : Sp E n →ₗ[E] Sp E n) (t : E) (x : Sp E N) : prH E hN (sumL hN g t x) = g (prH E hN x)

theorem **sumL_last** (g : Sp E n →ₗ[E] Sp E n) (t : E) (x : Sp E N) : sumL hN g t x (Br.iL hN) = t * x (Br.iL hN)

theorem **prH_sumE** (g : Sp E n ≃ₗ[E] Sp E n) (t : Eˣ) (x : Sp E N) : prH E hN (sumE hN g t x) = g (prH E hN x)

theorem **sumE_last** (g : Sp E n ≃ₗ[E] Sp E n) (t : Eˣ) (x : Sp E N) : sumE hN g t x (Br.iL hN) = t * x (Br.iL hN)

theorem **prH_sumE_symm** (g : Sp E n ≃ₗ[E] Sp E n) (t : Eˣ) (x : Sp E N) : prH E hN ((sumE hN g t).symm x) = g.symm (prH E hN x)

theorem **sumE_symm_last** (g : Sp E n ≃ₗ[E] Sp E n) (t : Eˣ) (x : Sp E N) : (sumE hN g t).symm x (Br.iL hN) = (t : E)⁻¹ * x (Br.iL hN)

theorem **sumE_mul** (g g' : Sp E n ≃ₗ[E] Sp E n) (t t' : Eˣ) : sumE hN (g * g') (t * t') = sumE hN g t * sumE hN g' t'

theorem **sumE_one** : sumE hN (1 : Sp E n ≃ₗ[E] Sp E n) 1 = 1

theorem **extUH_val** (γ : UF E n) : (extUH hN γ).1 = sumE hN γ.1 1

theorem **extGH_apply** {SG : Finset (HeightOneSpectrum (𝒪 E))} (h : GLG E n SG) (w : SG) : extGH hN h w = diagEmb _ (by omega : n ≤ N) (h w)

theorem **mem_stdAt_succ** {x : Sp E N} : x ∈ stdAt E N P ↔ prH E hN x ∈ stdAt E n P ∧ IntAt P (x (Br.iL hN))

theorem **extI_val** (j : GLint E n P) : (extI hN P j).1 = sumE hN j.1 1

theorem **diagMat_mulVec** {κ : Type} [CommRing κ] (A : Matrix (Fin n) (Fin n) κ) (v : Fin N → κ) : (∀ i, (diagMat n N A).mulVec v (Br.iC hN i) = A.mulVec (fun j => v (Br.iC hN j)) i) ∧ (diagMat n N A).mulVec v (Br.iL hN) = v (Br.iL hN)

theorem **redGL_extI** (j : GLint E n P) : redGL N P (extI hN P j) = diagEmb _ (by omega : n ≤ N) (redGL n P j)

theorem **be_smul_int** {d : Deco E n ST SG} (hd : d.Ok) (w : SG) {a : E} (ha : IntAt w.1 a) {x : Sp E n} (hx : x ∈ d.at w.1) : d.be w (a • x) = redP w.1 a • d.be w x

theorem **be_sum** {d : Deco E n ST SG} (hd : d.Ok) (w : SG) {ι : Type} (s : Finset ι) {f : ι → Sp E n} (h : ∀ i ∈ s, f i ∈ d.at w.1) : d.be w (∑ i ∈ s, f i) = ∑ i ∈ s, d.be w (f i)

theorem **exists_liftFr** {d : Deco E n ST SG} (hd : d.Ok) (w : SG) : ∃ f, IsLiftFr d w f

theorem **liftFr_spec** {d : Deco E n ST SG} (hd : d.Ok) (w : SG) : IsLiftFr d w (d.liftFr w)

theorem **redGL_passFr** {d : Deco E n ST SG} {w : SG} (γ : Sp E n ≃ₗ[E] Sp E n) (h : GLG E n SG) {f f' : Sp E n ≃ₗ[E] Sp E n} (hf : IsLiftFr d w f) (hf' : IsLiftFr (d.act γ h) w f') : redGL n w.1 (passFr γ h hf hf') = h w

theorem **exists_of_isLiftFr** {d : Deco E n ST SG} {w : SG} {f f' : Sp E n ≃ₗ[E] Sp E n} (hf : IsLiftFr d w f) (hf' : IsLiftFr d w f') : ∃ κ : GLint E n w.1, redGL n w.1 κ = 1 ∧ f = f' * κ.1

theorem **isLiftFr_tilH** (d : DecoSet E n ∅ SG) {pi : 𝒪 E} (hT : ∀ P : ST, cO E pi ∉ P.1.asIdeal) (hG : ∀ w : SG, pi ∉ w.1.asIdeal) (w : SG) {f : Sp E n ≃ₗ[E] Sp E n} (hf : IsLiftFr d.1 w f) (t : Eˣ) (ht : (t : E) = (pi : E)) : IsLiftFr (DecoSet.tilH (ST

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ReferenceChainEvaluationAtBundlePoints

theorem **row_all** {Bd : BdT} {Q : V4.Slot} {Lk : LkT Bd} (aPt0 : ReferenceChainEvaluationAtBundlePoints Q Lk) (aPtTw : TwistedPointAtBundlePoints Q Lk) (aPer0 : PeriodAtZeroAtBundlePoints Q Lk) (aPerTw : TwistedPeriodAtBundlePoints Q Lk) (tie : Row_tie Q Lk) : Row Q Lk fun 𝒜 _ => All 𝒜

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.SetUnipotentTranslate

theorem **Over.host_heq** {D : D7} {P : PreCh} {Lv : LevCh P} (h : Over D P Lv) : D.R = P.R ∧ HEq D.host Lv.host

theorem **Over.field** {D : D7} {P : PreCh} {Lv : LevCh P} (h : Over D P Lv) : HEq D.host.toHostAx P.H.toHostAx

theorem **downTo_tr** (μ : D.what₂.Tup) (j : ℕ) : (sideTr N h S).downTo μ j = S.downTo μ j

theorem **iota_tr** (μ : D.what₂.Tup) : (sideTr N h S).iota μ = S.iota μ

theorem **Qsp_tr** (μ : D.what₂.Tup) : (sideTr N h S).Qsp μ = S.Qsp μ

theorem **RRset_tr** (μ : D.what₂.Tup) : (sideTr N h S).RRset μ = S.RRset μ

theorem **iiStar_tr** : (sideTr N h S).iiStar = S.iiStar

theorem **kapOf_tr** {n : ℕ} (ty : Fin n → D.what₂.FT) : (sideTr N h S).kapOf ty = S.kapOf ty

theorem **rrDef_tr** (H : S.RRDef) : (sideTr N h S).RRDef

theorem **kapSharp_tr** : kapSharp (setDiv D N h) = kapSharp D

theorem **VV_tr** : G.VV (instTr N h X) K a b p = G.VV X K a b p

theorem **tau_translate** : G.tauP (instTr N h X) K a b p = G.tauP X K a b p

theorem **ioP_instTr** : G.ioP (instTr N h X) K a b p = G.ioP X K a b p

theorem **gainP_instTr** : G.gainP (instTr N h X) K a b p j = G.gainP X K a b p j

theorem **TTP_tr** : G.TTP (instTr N h X) K a b p = G.TTP X K a b p

theorem **fills_tr** (w : D.what₂.places.Pl) (H : G.ParametersFill X K w) : G.ParametersFill (instTr N h X) K w

theorem **NP_tr** (Y : G.LatInst D) (u : D.what₂.Word) (w : D.what₂.places.Pl) (hg : GIdx D.host (D.mem u).A.T (D.what₂.places.resChar w)) : G.NP (instTr N h X) (latTr N h Y) K u w hg = G.NP X Y K u w hg

theorem **bk_tr** (i : D.ntw.Tw) (s : D.ntw.Sig i) (k : ℕ) (ψ : BlochKatoReduction.Gen.GE D.nd →* (D.ntw.Ok i k)ˣ) : BlochKatoReduction.Gen.bk (BlochKatoReduction.bk0 (setDiv D N h)) i s k ψ = BlochKatoReduction.Gen.bk (BlochKatoReduction.bk0 D) i s k ψ

theorem **sLen_tr** (i : D.ntw.Tw) (s : D.ntw.Sig i) (k : ℕ) (Q : Finset (PlE D.host)) (ψ : BlochKatoReduction.Gen.GE D.nd →* (D.ntw.Ok i k)ˣ) : BlochKatoReduction.Gen.sLen (BlochKatoReduction.bk0 (setDiv D N h)) i s k Q ψ = BlochKatoReduction.Gen.sLen (BlochKatoReduction.bk0 D) i s k Q ψ

theorem **twAt_tr** {x : NChoice D} {y : ThData D} (H : LocalQuadraticParts.TWAt D x y) : LocalQuadraticParts.TWAt (setDiv D N h) (nchTr N h x) (xdTr N h y)

theorem **ptO_objO** (Q : PtProp) : ptO (objO Q) = Q

theorem **objO_ptO** (QO : ∀ D : D7, Obj D → Prop) : objO (ptO QO) = QO

theorem **has_ptO** {C : FC} (p : Pt C) (QO : ∀ D : D7, Obj D → Prop) : p.Has (ptO QO) ↔ QO p.D (Obj.ofPt p)

theorem **pt_D** : p.pt.D = setDiv D (nPrime p.L) p.ht

theorem **pt_sc** : p.pt.L.sc = p.L.sc

theorem **ofPt_pt** : Obj.ofPt p.pt = (Y.obj p).tr (nPrime p.L) p.ht

theorem **pt_wtEq** {φ : Formalism.HC.Frame} {a b : φ.A → ℤ} (H : WtEq p.L a b) : WtEq p.pt.L a b

theorem **pt_val** {cD ch : ℝ} (H : Val C cD ch p.L) : Val C cD ch p.pt.L

theorem **pt_canon** (H : CanonInst p.ta.Gx) : CanonInst p.pt.ta.Gx

theorem **pt_has** {s : Scale} {cD : ℝ} {Q : PtProp} (hR : RecOK s C cD D) (hT : LocalQuadraticParts.TWAt D Y.nch Y.xd) (h0 : DC.GlobalTwistingFacts p.L) (hQ : p.pt.Has Q) : p.pt.Has (DC.withObj Q)

theorem **RecEx.imp** {QO QO' : ∀ D : D7, Obj D → Prop} (h : RecEx QO) (hQ : ∀ D O, QO D O → QO' D O) : RecEx QO'

theorem **obj_ofPt** {C : FC} (p : Pt C) : (Lay.ofPt p).obj (LedPart.ofPt p) = Obj.ofPt p

theorem **pt_ofPt** {C : FC} (p : Pt C) : (LedPart.ofPt p).pt.D.div.Nu = p.D.div.Nu ∧ (LedPart.ofPt p).pt.D.div.h = p.D.div.h

theorem **nonempty_ledPart_iff** (C : FC) : (∃ (D : D7) (Y : Lay D), Nonempty (LedPart C Y)) ↔ Nonempty (Pt C)

theorem **leastOff_eq_rPrH** {R : Ranks} (H : Host R) (hc : H.ccK = ⊤) : leastOff (H.QS * H.q * H.d * (H.pde - 1) * (H.pde' - 1)) = DC.rPrH H

theorem **chGood_of_host** (eps C5 c6 Cb cD ch : ℝ) (C : FC) (he : 0 < eps) (he1 : eps < 1 / 52) (hC : Std C) (hcD : 0 ≤ cD) (hch : 402 ≤ ch) : ∃ x0 : ℝ, ∀ (s : Scale) (R : Ranks) (H : Host R) (a b : R.A → ℤ) (M : ℕ) (c : Fin M → Fin (R.k - 2) → Bool) (u0 : ℕ), s.eps = eps → x0 ≤ s.x → RankVal s R → H.CT2 = C.CT2 → HostGood s C.P.Cpde C5 c6 Cb H (DC.q0H H) → WinOK s C.GK H a b → M = ⌈3 * GamS C s R ^ 2 * s.x ^ C.P.CF⌉₊ → Function.Injective c → ∃ (hcW : Fin M → Tau R → HCParam R) (w0 : Fin M) (h1 : _) (h2 : _) (h3 : _) (h4 : _) (h5 : _) (h6 : _) (h7 : _) (h8 : _), ChGood s C C5 c6 Cb cD ch (@PreCh.mk R (setUnipotentTranslate H u0) h1 h2 h3 (Fin M) _ w0 hcW h4 h5 h6 h7 h8) a b c

theorem **exists_chGoodW** {la : ℝ} {Q : HostReq} (hH : DC.HostExQW la Q) (eps : ℝ) (he : 0 < eps) (he1 : eps < 1 / 52) : ∃ CT2 C8 C5 c6 Cb : ℝ, 1 ≤ C8 ∧ 0 < C5 ∧ 0 < c6 ∧ 0 < Cb ∧ ∃ S : Scales la, ∀ C : FC, Std C → C.CT2 = CT2 → C.P.Cpde = C8 → ∀ cD ≥ (0 : ℝ), ∀ ch ≥ (402 : ℝ), ∃ x0 : ℝ, ∀ s : Scale, s.eps = eps → x0 ≤ s.x → S.Mem s → ∀ u0 : ℕ, ∃ (P : PreCh) (a b : P.R.A → ℤ) (c : P.Word → Fin (P.R.k - 2) → Bool), ChGood s C C5 c6 Cb cD ch P a b c ∧ P.H.u0 = P.R.ofNat u0 ∧ ∃ v : P.R.A, Q C5 s P.R { P.H with u0

theorem **exists_chGood** (hH : DC.HostExQ) (eps : ℝ) (he : 0 < eps) (he1 : eps < 1 / 52) : ∃ CT2 C8 C5 c6 Cb : ℝ, 1 ≤ C8 ∧ 0 < C5 ∧ 0 < c6 ∧ 0 < Cb ∧ ∃ la : ℝ, 1 < la ∧ ∃ S : Scales la, ∀ C : FC, Std C → C.CT2 = CT2 → C.P.Cpde = C8 → ∀ cD ≥ (0 : ℝ), ∀ ch ≥ (402 : ℝ), ∃ x0 : ℝ, ∀ s : Scale, s.eps = eps → x0 ≤ s.x → S.Mem s → ∀ u0 : ℕ, ∃ (P : PreCh) (a b : P.R.A → ℤ) (c : P.Word → Fin (P.R.k - 2) → Bool), ChGood s C C5 c6 Cb cD ch P a b c ∧ P.H.u0 = P.R.ofNat u0

theorem **exists_chGood_unb** (hH : DC.HostExQ) (eps : ℝ) (he : 0 < eps) (he1 : eps < 1 / 52) : ∃ CT2 C8 C5 c6 Cb : ℝ, 1 ≤ C8 ∧ 0 < C5 ∧ 0 < c6 ∧ 0 < Cb ∧ ∀ C : FC, Std C → C.CT2 = CT2 → C.P.Cpde = C8 → ∀ cD ≥ (0 : ℝ), ∀ ch ≥ (402 : ℝ), ∀ y : ℝ, ∃ s : Scale, s.eps = eps ∧ y ≤ s.x ∧ ∀ u0 : ℕ, ∃ (P : PreCh) (a b : P.R.A → ℤ) (c : P.Word → Fin (P.R.k - 2) → Bool), ChGood s C C5 c6 Cb cD ch P a b c ∧ P.H.u0 = P.R.ofNat u0

theorem **ChGood.level** {s : Scale} {C : FC} {C5 c6 Cb cD ch : ℝ} {P : PreCh} {a b : P.R.A → ℤ} {c : P.Word → Fin (P.R.k - 2) → Bool} (h : ChGood s C C5 c6 Cb cD ch P a b c) : P.H.levelPrimes = {P.H.g 0, P.H.g 1, P.H.pde, P.H.pde', DC.q0H P.H} ∧ (DC.q0H P.H : ℝ) ≤ ((P.H.QS : ℝ) * P.H.q * P.H.d) ^ C.P.Cpde ∧ BoxW s P.H ∧ QDq s.x s.eps C5 P.H.q P.H.d

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.SmulCommClassGalIntegers

theorem **normOne_iff** {d : ℕ} (hd : d % 4 = 3) (M p : ℕ) : NormOne d M p ↔ GlobalTheory.Mem.An.NormOne d M p

theorem **isFrob_iff** (𝔭 : HeightOneSpectrum (𝒪 K)) (σ : L ≃ₐ[K] L) : IsFrob L 𝔭 σ ↔ ∃ P : Ideal (𝒪 L), P.IsPrime ∧ P.LiesOver 𝔭.asIdeal ∧ IsArithFrobAt (𝒪 K) σ P

theorem **inInertia_iff** (𝔭 : HeightOneSpectrum (𝒪 K)) (σ : L ≃ₐ[K] L) : InInertia L 𝔭 σ ↔ ∃ P : Ideal (𝒪 L), P.IsPrime ∧ P.LiesOver 𝔭.asIdeal ∧ σ ∈ P.inertia (L ≃ₐ[K] L)

theorem **exists_isFrob** [IsGalois K L] (𝔭 : HeightOneSpectrum (𝒪 K)) : ∃ σ : L ≃ₐ[K] L, IsFrob L 𝔭 σ

theorem **heckeVal_eq** [IsGalois K L] (χ : (L ≃ₐ[K] L) →* ℂˣ) (𝔭 : HeightOneSpectrum (𝒪 K)) (hχ : ∀ τ, InInertia L 𝔭 τ → χ τ = 1) {σ : L ≃ₐ[K] L} (hσ : IsFrob L 𝔭 σ) : heckeVal L χ 𝔭 = (χ σ : ℂˣ)

theorem **heckeVal_eq_zero** (χ : (L ≃ₐ[K] L) →* ℂˣ) (𝔭 : HeightOneSpectrum (𝒪 K)) (hχ : ¬ ∀ τ, InInertia L 𝔭 τ → χ τ = 1) : heckeVal L χ 𝔭 = 0

theorem **heckeVal_of_unramIn** [IsGalois K L] (χ : (L ≃ₐ[K] L) →* ℂˣ) {𝔭 : HeightOneSpectrum (𝒪 K)} (hu : UnramIn L 𝔭) {σ : L ≃ₐ[K] L} (hσ : IsFrob L 𝔭 σ) : heckeVal L χ 𝔭 = (χ σ : ℂˣ)

theorem **heckeVal_one** [IsGalois K L] (𝔭 : HeightOneSpectrum (𝒪 K)) : heckeVal L (1 : (L ≃ₐ[K] L) →* ℂˣ) 𝔭 = 1

theorem **condNorm_one** [IsGalois K L] : condNorm L (1 : (L ≃ₐ[K] L) →* ℂˣ) = 1

theorem **heckeZero_iff** (χ : (L ≃ₐ[K] L) →* ℂˣ) {G : ℂ → ℂ} (hG : DifferentiableOn ℂ G {1}ᶜ) (hp : ∀ s : ℂ, 1 < s.re → HasProd (fun 𝔭 : HeightOneSpectrum (𝒪 K) => (1 − heckeVal L χ 𝔭 * ((Ideal.absNorm 𝔭.asIdeal : ℕ) : ℂ) ^ (−s))⁻¹) (G s)) (β : ℂ) : HeckeZero L χ β ↔ β ≠ 1 ∧ G β = 0

theorem **absNorm_rat** (p : HeightOneSpectrum (𝒪 ℚ)) : Ideal.absNorm p.asIdeal = (Rat.HeightOneSpectrum.primesEquiv p : ℕ)

theorem **heckeZero_rat_one_iff** (L : Type) [Field L] [NumberField L] [Algebra ℚ L] [IsGalois ℚ L] (β : ℂ) : HeckeZero L (1 : (L ≃ₐ[ℚ] L) →* ℂˣ) β ↔ riemannZeta β = 0

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ThinScalesIffHostBound

theorem **isCharH_iff_isCharED** (D : D7) {N : ℕ} (χ : DirichletCharacter ℂ N) : IsCharH D.host χ ↔ GlobalTheoryTheory.IsCharED D χ

theorem **isCharH_iff** (H : Host R) {N : ℕ} (χ : DirichletCharacter ℂ N) : IsCharH H χ ↔ H.IsCharE χ

theorem **thinScales_iff_hostBound** (s : Scale) (c : ℝ) (H : Host R) : ThinOK s c H ↔ H.ChebyshevCharacterBound s.x s.eps c

theorem **BoxW.what₂** {s : Scale} {H : Host R} (h : BoxW s H) : ∃ Y : ℝ, PairWin s Y ∧ PairRow Y s.eps H

theorem **BoxW.boxesW** {s : Scale} {D : D7} (h : BoxW s D.host) : BoxesW D s

theorem **BoxW.qdRowsW** {s : Scale} {D : D7} (h : BoxW s D.host) : QDRowsW D s

theorem **BoxH.real_ne_zero** {H : Host R} {Y eps : ℝ} (h : BoxH H Y eps) {N : ℕ} [NeZero N] (χ : DirichletCharacter ℂ N) (hχ : IsCharH H χ) (hr : GlobalTheoryTheory.IsRealChar χ) (hN : H.q ∣ N ∨ H.d ∣ N) (β : ℝ) (hβ : 1 − 1 / 400 < β) : DirichletCharacter.LFunction χ β ≠ 0

theorem **BoxW.real_ne_zero** {s : Scale} {H : Host R} (h : BoxW s H) {N : ℕ} [NeZero N] (χ : DirichletCharacter ℂ N) (hχ : H.IsCharE χ) (hr : GlobalTheoryTheory.IsRealChar χ) (hN : H.q ∣ N ∨ H.d ∣ N) (β : ℝ) (hβ : 1 − 1 / 400 < β) : DirichletCharacter.LFunction χ β ≠ 0

private theorem **one_lt_modulus** (H : Host R) : 1 < H.QS * H.q * H.d

theorem **HostGood.pde_mod** {s : Scale} {C8 C5 c6 Cb : ℝ} {H : Host R} {q0 : ℕ} (h : HostGood s C8 C5 c6 Cb H q0) : H.pde % (H.QS * H.q * H.d) = 1 ∧ H.pde' % (H.QS * H.q * H.d) = 1

theorem **HostExW.imp** {la : ℝ} {P P' : HostReq} (h : HostExW la P) (hP : ∀ c s R H q0, P c s R H q0 → P' c s R H q0) : HostExW la P'

theorem **HostExP.imp** {P P' : HostReq} (h : HostExP P) (hP : ∀ c s R H q0, P c s R H q0 → P' c s R H q0) : HostExP P'

theorem **HostExP.hostEx** {P : HostReq} (h : HostExP P) : HostEx

theorem **Pt.From.words** (h : p.From H a b c u0) : ∃ eW : p.D.what₂.Word ≃ Fin M, ∀ u l, wordBits p.D u l = if h : l < R.k − 2 then c (eW u) (l, h) else false

theorem **Pt.From.field** (h : p.From H a b c u0) : HEq p.D.host.toHostAx H.toHostAx

theorem **Pt.From.congr_field** (h : p.From H a b c u0) {α : Sort*} (f : ∀ φ : Formalism.HC.Frame, Formalism.HC.HostAx φ → α) : f p.D.R.toFrame p.D.host.toHostAx = f R.toFrame H.toHostAx

theorem **Pt.From.QS_eq** (h : p.From H a b c u0) : p.D.host.QS = H.QS

theorem **Pt.From.art_eq** (h : p.From H a b c u0) : artNum p.D.host.toHostRaw = artNum H.toHostRaw

theorem **Pt.From.rd_eq** (h : p.From H a b c u0) : p.D.host.rdE = H.rdE

theorem **Pt.From.cls_eq** (h : p.From H a b c u0) : p.D.host.cls = H.cls

theorem **Pt.From.isCharED_iff** (h : p.From H a b c u0) {N : ℕ} (χ : DirichletCharacter ℂ N) : GlobalTheoryTheory.IsCharED p.D χ ↔ IsCharH H χ

theorem **Pt.From.chThin** (h : p.From H a b c u0) {s : Scale} {c6 : ℝ} (hT : ThinOK s c6 H) : ChThin_at p.D s.x s.eps c6

theorem **Pt.From.boxE** (h : p.From H a b c u0) {Y eps : ℝ} (hB : BoxH H Y eps) : GlobalTheoryTheory.BoxE p.D Y eps

theorem **Pt.From.pairRow** (h : p.From H a b c u0) {Y eps : ℝ} (hr : PairRow Y eps H) : PairRow Y eps p.D.host

theorem **Pt.From.boxW** (h : p.From H a b c u0) {s : Scale} (hB : BoxW s H) : BoxW s p.D.host

theorem **Pt.From.boxesW** (h : p.From H a b c u0) {s : Scale} (hB : BoxW s H) : BoxesW p.D s

theorem **Pt.From.qdRowsW** (h : p.From H a b c u0) {s : Scale} (hB : BoxW s H) : QDRowsW p.D s

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.TorIsoId

theorem **torIso_id** (K : IntermediateField ℚ ℂ) (q : ℕ) (L : PeriodPair) : TorIso K q L L id where

theorem **transp_id** (L : PeriodPair) : Transp (RingEquiv.refl ℂ) L L id where

Development.AbelianVarietyTheory.Definitions.GlobalTwisting.ValidForConstructedPointClassesEventuallyWorking

theorem **evGd_of_ev** {p : Ctx → Prop} (hp : Ev p) : EvGd p

theorem **large_of_sup** {C : FC} {K K' : TransportedAlt} {CH : ℝ} (h : C.Large (K.sup K') CH) : C.Large K CH ∧ C.Large K' CH

theorem **ev_and** {p q : Ctx → Prop} (hp : EvGd p) (hq : EvGd q) : EvGd fun K => p K ∧ q K

theorem **ev_mono** {p q : Ctx → Prop} (h : ∀ K, p K → q K) (hp : EvGd p) : EvGd q

theorem **ParEv.and** {p q : Par → Prop} (hp : ParEv p) (hq : ParEv q) : ParEv fun κ => p κ ∧ q κ

theorem **ParEv.mono** {p q : Par → Prop} (h : ∀ κ, p κ → q κ) (hp : ParEv p) : ParEv q

theorem **ParEv.of_all** {p : Par → Prop} (h : ∀ κ, p κ) : ParEv p

theorem **ParEv.choose** {α : Type} [Nonempty α] {r : Par → α → Prop} (h : ParEv fun κ => ∃ a, r κ a) : ∃ f : Par → α, ParEv fun κ => r κ (f κ)

theorem **EvP.and** {p q : Par → Ctx → Prop} (hp : EvP p) (hq : EvP q) : EvP fun κ K => p κ K ∧ q κ K

theorem **EvP.mono** {p q : Par → Ctx → Prop} (h : ∀ κ K, κ.Of K → p κ K → q κ K) (hp : EvP p) : EvP q

theorem **EvP.of_all** {p : Par → Ctx → Prop} (h : ∀ κ K, κ.Of K → p κ K) : EvP p

theorem **EvP.of_evGd** {p : Ctx → Prop} (hp : EvGd p) : EvP fun _ K => p K

theorem **EvP.of_ev** {p : Ctx → Prop} (hp : Ev p) : EvP fun _ K => p K

theorem **recExP_const_iff** (Q0 : ∀ D : D7, Obj D → Prop) : (RecExP fun _ => Q0) ↔ RecEx Q0

theorem **Link.csz** {κ : Par} {eps : ℝ} {Fm : FamP eps} (h : Link κ Fm) : Fm.Csz = κ.C.Csz

theorem **Link.host** {κ : Par} {eps : ℝ} {Fm : FamP eps} (h : Link κ Fm) : max Fm.Co Fm.J = κ.C.P.Cpde

theorem **familyClassWithFurtherProperties_mono** {Q0 Q0' : ∀ D : D7, Obj D → Prop} (h : ∀ D O, Q0 D O → Q0' D O) {eps : ℝ} (Fm : FamP eps) (hc : DQ.cls7 Q0 Fm) : DQ.cls7 Q0' Fm

theorem **sevenDuties_of_class** {Q0 : Par → ∀ D : D7, Obj D → Prop} {eps : ℝ} {Fm : FamP eps} (h : clsP Q0 Fm) : DQ.Duty7 Fm ∧ DQ.MtStableF Fm

theorem **cls_at** {E : Par → BdP Bd} {eps : ℝ} (Fm : FamP eps) (hc : cls Q Lk E Fm) (x : ℝ) (hx : Fm.x1 < x) (i : Fm.Ix x) : ∃ b : Bd (Fm.D x i) (objAt Fm x hx i) Fm.Csz, Built Q Lk (Classical.choose hc) _ _ _ b ∧ E (Classical.choose hc) _ _ _ b

theorem **pick_isSel** (E : Par → BdP Bd) : IsSel Q Lk E (cls Q Lk E) (pick Q Lk E)

theorem **Built.ind** {κ : Par} {R : BdP Bd} (h : ∀ (K : Ctx) (β : BPt K) (c : ℝ) (b : Bd β.D β.O c), κ.Of K → β.Is Q → Lk K β c b → R β.D β.O c b) {D : D7} {O : Obj D} {c : ℝ} {b : Bd D O c} (hb : Built Q Lk κ D O c b) : R D O c b

theorem **Built.intro** {κ : Par} {K : Ctx} {β : BPt K} {c : ℝ} {b : Bd β.D β.O c} (h0 : κ.Of K) (h1 : β.Is Q) (h2 : Lk K β c b) : Built Q Lk κ β.D β.O c b

theorem **RowE.and** {E E' : Par → BdP Bd} (h : RowE Q Lk E) (h' : RowE Q Lk E') : RowE Q Lk fun κ D O c b => E κ D O c b ∧ E' κ D O c b

theorem **RowE.imp** {E E' : Par → BdP Bd} (h : RowE Q Lk E) (hi : ∀ (κ : Par) (K : Ctx) (β : BPt K) (c : ℝ) (b : Bd β.D β.O c), κ.Of K → β.Is Q → Lk K β c b → E κ _ _ c b → E' κ _ _ c b) : RowE Q Lk E'

theorem **RowE.of_all** {E : Par → BdP Bd} (h : ∀ (κ : Par) (K : Ctx) (β : BPt K) (c : ℝ) (b : Bd β.D β.O c), κ.Of K → β.Is Q → Lk K β c b → E κ _ _ c b) : RowE Q Lk E

theorem **IsSel.mono** {E E' : Par → BdP Bd} (h : ∀ κ D O c b, E κ D O c b → E' κ D O c b) {𝒜 : FamClass} {σ : RawSel Bd 𝒜} (s : IsSel Q Lk E 𝒜 σ) : IsSel Q Lk E' 𝒜 σ

theorem **Row.and** {H H' : ∀ 𝒜 : FamClass, RawSel Bd 𝒜 → Prop} (h : Row Q Lk H) (h' : Row Q Lk H') : Row Q Lk fun 𝒜 σ => H 𝒜 σ ∧ H' 𝒜 σ

theorem **Row.imp** {H H' : ∀ 𝒜 : FamClass, RawSel Bd 𝒜 → Prop} (h : Row Q Lk H) (hi : ∀ (𝒜 : FamClass) (σ : RawSel Bd 𝒜), IsSel Q Lk (fun _ _ _ _ _ => True) 𝒜 σ → H 𝒜 σ → H' 𝒜 σ) : Row Q Lk H'

theorem **Row.of_sel** {H : ∀ 𝒜 : FamClass, RawSel Bd 𝒜 → Prop} (h : ∀ (𝒜 : FamClass) (σ : RawSel Bd 𝒜), IsSel Q Lk (fun _ _ _ _ _ => True) 𝒜 σ → H 𝒜 σ) : Row Q Lk H

theorem **RowE.row** {R : BdP Bd} (h : RowE Q Lk fun _ => R) : Row Q Lk (AtAll R)

theorem **Row.map** (hQ : ∀ K X f T c, Q K X f T c → Q' K X f T c) (hL : ∀ (K : Ctx) (β : BPt K) (c : ℝ) (b : Bd β.D β.O c), β.Is Q → Lk K β c b → Lk' K β c (π _ _ c b)) {H : ∀ 𝒜 : FamClass, RawSel Bd' 𝒜 → Prop} (h : Row Q' Lk' H) : Row Q Lk fun 𝒜 σ => H 𝒜 (RawSel.map π σ)

theorem **Row0.row** {H : FamClass → Prop} (h : Row0 Q H) (Lk : LkT Bd) : Row Q Lk fun 𝒜 _ => H 𝒜

theorem **Row0.of_all** {H : FamClass → Prop} (h : ∀ 𝒜 : FamClass, H 𝒜) : Row0 Q H

theorem **QOf₂** {E : Par → BdP Bd} (hx : ExB Q Lk) (he : RowE Q Lk E) : EvP fun κ K => ∀ β : BPt K, β.Is Q → QOf Q Lk E κ β.D β.O

theorem **Row.elim** {H : ∀ 𝒜 : FamClass, RawSel Bd 𝒜 → Prop} (hx : ExB Q Lk) (h : Row Q Lk H) : ∃ E : Par → BdP Bd, (EvP fun κ K => ∀ β : BPt K, β.Is Q → QOf Q Lk E κ β.D β.O) ∧ H (cls Q Lk E) (pick Q Lk E)

theorem **w3_a** {𝒜 : FamClass} {eps : ℝ} {Fm : FamP eps} (h : W3 𝒜 Fm) : 𝒜 Fm

theorem **evaluation_constantFamily** {α : Type} {R : α → ∀ D : D7, Obj D → Prop} (a : Par → α) (h : EvP fun κ K => ∀ β : BPt K, β.Is Q → R (a κ) β.D β.O) : Row0 Q (EvCC R)

theorem **validForConstructedPointClasses_eventuallyWorkingClass** {R : ∀ D : D7, Obj D → Prop} (h : EvP fun _ K => ∀ β : BPt K, β.Is Q → R β.D β.O) : Row0 Q (EvC R)

Development.AbelianVarietyTheory.Definitions.HeightFunctions.ConclOfInputs4

theorem **concl_of_inputs4** {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} (large₂ : SizeData₂ D) (X : Prov D B P) (K : KAdm) (H : KHyp) (Ch Ah Cq Aq C2 A2 CR CT : ℝ) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (S : Scal B) (ka wL wF : WeightFamilies.Wt D j) (T nP nW : ℕ) (h0 : ℝ) (nus : HeightOneSpectrum (𝒪 B.KO) → ℝ) (h : Inputs4 large₂ (X j h1 h2) K H Ch Ah Cq Aq C2 A2 CR CT S ka wL wF T nP nW h0 nus) : Concl (X j h1 h2) S ka wL wF T nus ((X j h1 h2).rk ka (B.Mlat j ka) * D.R.f * D.R.m * large₂.sc.b * large₂.de2)

Development.AbelianVarietyTheory.Definitions.HeightFunctions.FiniteDimensionalOfSz

theorem **finiteDimensional_of_sz** {F : IntermediateField ℚ ℂ} {a b : ℝ} (h : Sz F a b) : FiniteDimensional ℚ F

Development.AbelianVarietyTheory.Definitions.HeightFunctions.GrowthConstantBoundGcR

theorem **growthConstantBound_gcR** (large₂ : SizeData₂ D) (B : WeightFamilies.Prov D) : GrowthConstantBound large₂ B (gcR large₂ B)

Development.AbelianVarietyTheory.Definitions.HeightFunctions.IntegralOfMemIntAE

theorem **barF_tmul** (a : Af D.host.F) (x : D.host.E) : barF D (a ⊗ₜ x) = a ⊗ₜ star x

theorem **barF_barF** (x : AfE D) : barF D (barF D x) = x

theorem **normOne_one** : NormOne (1 : (AfE D)ˣ)

theorem **normOne_neg_one** : NormOne (-1 : (AfE D)ˣ)

theorem **NormOne.mul'** {x y : (AfE D)ˣ} (hx : NormOne x) (hy : NormOne y) : NormOne (x * y)

theorem **NormOne.inv_eq** {x : (AfE D)ˣ} (hx : NormOne x) : ((x⁻¹ : (AfE D)ˣ) : AfE D) = barF D x

theorem **NormOne.inv'** {x : (AfE D)ˣ} (hx : NormOne x) : NormOne x⁻¹

theorem **normOne_div** (y : (AfE D)ˣ) : NormOne (y * (Units.map (barF D : AfE D →* AfE D) y)⁻¹)

theorem **congOne_of_dvd** {M N : ℕ} (h : M ∣ N) {x : (AfE D)ˣ} (hx : CongOne N x) : CongOne M x

theorem **not_xiOK_one** (p : ℕ) : ¬ XiOK D p 1

theorem **not_xiOK_level_one** (xi : (AfE D)ˣ →* ℂˣ) : ¬ XiOK D 1 xi

theorem **xiOK_inv** {p : ℕ} {xi : (AfE D)ˣ →* ℂˣ} (h : XiOK D p xi) : XiOK D p xi⁻¹ where

theorem **xi_neg_one** {p : ℕ} {xi : (AfE D)ˣ →* ℂˣ} (h : XiOK D p xi) : xi (-1) = 1

theorem **primeClass_zero** {φ : Formalism.HC.Frame} (H : Formalism.HC.HostRaw φ) : primeClass H 0 = PrimeClass.splitK

theorem **prime_of_fieldOK** {p : ℕ} (h : FieldOK D p) : p.Prime ∧ p % 15 = 1 ∧ ∀ i, p ≠ D.host.g i

theorem **zeta_isIntegral** {n : ℕ} (hn : 0 < n) : IsIntegral ℚ (zeta n)

theorem **exists_field** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) {n p : ℕ} (hn : 0 < n) (hp : 0 < p) : ∃ F₀ : IntermediateField ℚ ℂ, FiniteDimensional ℚ F₀ ∧ ∀ F, F₀ ≤ F → zeta p ∈ F ∧ GlobalTheory.CompRat.CurveOver H F n

theorem **integral_of_mem_intAE** {x : D.host.E} (hx : (1 : Af D.host.F) ⊗ₜ[D.host.F] x ∈ intAE D) : ∃ y : 𝒪 D.host.E, (y : D.host.E) = x

theorem **not_congOne_neg_one** {N : ℕ} (hN : 3 ≤ N) : ¬ CongOne N (-1 : (AfE D)ˣ)

theorem **exists_deltaOK** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (h1 : 1 ≤ Dw.p + Dw.q) (hA : l.A = 1) (p : ℕ) (hp : p.Prime) (h5 : 5 ≤ p) (hc : p.Coprime l.M) (hj : jacobiSym (-(H.d : ℤ)) p = 1) : ∃ (I : Ideal (𝒪 E)) (c' : Coeff l), c.F ≤ c'.F ∧ ∀ c'' : Coeff l, c'.F ≤ c''.F → ∀ n : ℕ, ∃ d : DeltaDat Xo c'' n, DeltaOK d p I

theorem **fieldOK_of_stages** (X : GlobalTheory.Mem.Stages) {p : ℕ} (hp : p = X.toD7.host.pde ∨ p = X.toD7.host.pde') (hz : zeta p ∈ FO X.toD7) (hc : GlobalTheory.CompRat.CurveOver (hostOfRecord X.toD7) (FO X.toD7) (X.toD7.host.g 0 * X.toD7.host.g 1 * p)) (hk : OKPsi X.toD7) (hv : VolTieG X.toD7) (h0 : 0 ∉ X.toD7.host.levelPrimes) : FieldOK X.toD7 p where

Development.AbelianVarietyTheory.Definitions.HeightFunctions.NormComparisonConstants

theorem **fieldOK_iff** (D : D7) (p : ℕ) : FieldOK D p ↔ FieldOK0 D p ∧ Formalism.L2.Theta.VolTieG D

Development.AbelianVarietyTheory.Definitions.HeightFunctions.XiValIffXiAt

theorem **xiVal_iff_xiAt** (T : Full D) (j n : ℕ) : T.XiVal j n ↔ XiAt T.J j (T.dδ j n) T.xi

theorem **chiOK_one** (D : D7) : ChiOK D (fun _ => 1)

Development.AbelianVarietyTheory.Definitions.HermitianLattices.ExistsZsmulMemOfIsLattice

theorem **exists_zsmul_mem_of_isLattice** {Λ : Submodule (𝒪 K) (Fin r → K)} (hΛ : IsLattice Λ) (v : Fin r → K) : ∃ n : ℤ, n ≠ 0 ∧ n • v ∈ Λ

theorem **isLatticeEmb_subtype** {Λ : Submodule (𝒪 K) (Fin r → K)} (hΛ : IsLattice Λ) : IsLatticeEmb Λ.subtype

theorem **isLatticeEmb_of_span** [Module.Finite (𝒪 K) M] {φ : M →ₗ[𝒪 K] (Fin r → K)} (hi : Function.Injective φ) (hs : Submodule.span K (Set.range φ) = ⊤) : IsLatticeEmb φ

theorem **exists_zsmul** (h : IsLatticeEmb φ) (v : Fin r → K) : ∃ n : ℤ, n ≠ 0 ∧ ∃ x, φ x = n • v

theorem **finite** (h : IsLatticeEmb φ) : Module.Finite (𝒪 K) M

theorem **isLocalizedModule_int** (h : IsLatticeEmb φ) : IsLocalizedModule ℤ⁰ (φ.restrictScalars ℤ) where

theorem **isLocalizedModule** (h : IsLatticeEmb φ) : IsLocalizedModule (𝒪 K)⁰ φ where

theorem **isBaseChange_rat** (h : IsLatticeEmb φ) : IsBaseChange ℚ (φ.restrictScalars ℤ)

theorem **isBaseChange** (h : IsLatticeEmb φ) : IsBaseChange K φ

theorem **finrank_eq** (h : IsLatticeEmb φ) : Module.finrank (𝒪 K) M = r

theorem **finrank_int_eq** (h : IsLatticeEmb φ) : Module.finrank ℤ M = r * Module.finrank ℚ K

theorem **bilin_ext** (h : IsLatticeEmb φ) {B B' : LinearMap.BilinForm ℚ (Fin r → K)} (hB : ∀ x y, B (φ x) (φ y) = B' (φ x) (φ y)) : B = B'

theorem **matrix_ext** {s : ℕ} (h : IsLatticeEmb φ) {A A' : Matrix (Fin s) (Fin r) K} (hA : ∀ x, A *ᵥ φ x = A' *ᵥ φ x) : A = A'

theorem **exists_matrix** {s : ℕ} (h : IsLatticeEmb φ) (g : M →ₗ[𝒪 K] (Fin s → K)) : ∃ A : Matrix (Fin s) (Fin r) K, ∀ x, A *ᵥ φ x = g x

theorem **existsUnique_GL** {φ' : M' →ₗ[𝒪 K] (Fin r → K)} (h : IsLatticeEmb φ) (h' : IsLatticeEmb φ') (g : M ≃ₗ[𝒪 K] M') : ∃! γ : GL (Fin r) K, ∀ x, (γ : Matrix (Fin r) (Fin r) K) *ᵥ φ x = φ' (g x)

theorem **comp_GL** (h : IsLatticeEmb φ) (γ : GL (Fin r) K) : IsLatticeEmb (((γ : Matrix (Fin r) (Fin r) K).mulVecLin.restrictScalars (𝒪 K)) ∘ₗ φ)

@[simp] theorem **extend_apply** (h : IsLatticeEmb φ) (β : LinearMap.BilinForm ℤ M) (x y : M) : h.extend β (φ x) (φ y) = β x y

theorem **extend_isAlt** (h : IsLatticeEmb φ) {β : LinearMap.BilinForm ℤ M} (hβ : ∀ x, β x x = 0) : (h.extend β).IsAlt

theorem **extend_nondegenerate_iff** (h : IsLatticeEmb φ) (β : LinearMap.BilinForm ℤ M) : (h.extend β).Nondegenerate ↔ Function.Injective β

theorem **exists_isLatticeEmb** [Module.Finite (𝒪 K) M] [Module.IsTorsionFree (𝒪 K) M] (hr : Module.finrank (𝒪 K) M = r) : ∃ φ : M →ₗ[𝒪 K] (Fin r → K), IsLatticeEmb φ

Development.AbelianVarietyTheory.Definitions.HermitianLattices.TraceFormBilinSeparatingLeftIff

theorem **eq_zero_of_forall_trace_mul** {K : Type} [Field K] [NumberField K] {t : K} (h : ∀ a : K, Algebra.trace ℚ K (a * t) = 0) : t = 0

theorem **herm_single_single** (G : Matrix (Fin r) (Fin r) E) (a b : Fin r) : herm G (Pi.single a 1) (Pi.single b 1) = G a b

theorem **herm_injective** {G G' : Matrix (Fin r) (Fin r) E} (h : ∀ x y, herm G x y = herm G' x y) : G = G'

theorem **herm_eq_dotProduct** (G : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : herm G x y = x ⬝ᵥ G *ᵥ fun b => cc (y b)

theorem **herm_mulVec** {s : ℕ} (G : Matrix (Fin r) (Fin r) E) (γ : Matrix (Fin r) (Fin s) E) (x y : Fin s → E) : herm G (γ *ᵥ x) (γ *ᵥ y) = herm (γᵀ * G * γ.map cc) x y

theorem **EFf_neg_eq_trace** (ξ : E) (G : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : EFf (-ξ) G x y = Algebra.trace ℚ E (ξ * herm G x y)

theorem **EFf_smul_eq_trace** (ξ0 : E) (G : Matrix (Fin r) (Fin r) E) (a : E) (x y : Fin r → E) : EFf ξ0 G (a • x) y = Algebra.trace ℚ E (a * (-ξ0 * herm G x y))

theorem **traceFormBilin_isConjAdj** (ξ0 : E) (G : Matrix (Fin r) (Fin r) E) : IsConjAdj (traceFormBilin ξ0 G)

theorem **EFf_swap** {ξ0 : E} (hξ : cc ξ0 = -ξ0) {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, cc (G a b) = G b a) (x y : Fin r → E) : EFf ξ0 G y x = -EFf ξ0 G x y

theorem **EFf_self** {ξ0 : E} (hξ : cc ξ0 = -ξ0) {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, cc (G a b) = G b a) (x : Fin r → E) : EFf ξ0 G x x = 0

theorem **traceFormBilin_isAlt** {ξ0 : E} (hξ : cc ξ0 = -ξ0) {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, cc (G a b) = G b a) : (traceFormBilin ξ0 G).IsAlt

theorem **herm_eq_of_EFf_smul** {s : ℕ} {ξ0 : E} (h0 : ξ0 ≠ 0) {G : Matrix (Fin r) (Fin r) E} {G' : Matrix (Fin s) (Fin s) E} {x y : Fin r → E} {x' y' : Fin s → E} (h : ∀ a : E, EFf ξ0 G (a • x) y = EFf ξ0 G' (a • x') y') : herm G x y = herm G' x' y'

theorem **traceFormBilin_injective** {ξ0 : E} (h0 : ξ0 ≠ 0) : Function.Injective (traceFormBilin ξ0 : Matrix (Fin r) (Fin r) E → _)

theorem **traceFormBilin_isAlt_iff** {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) (G : Matrix (Fin r) (Fin r) E) : (traceFormBilin ξ0 G).IsAlt ↔ ∀ a b, cc (G a b) = G b a

theorem **forall_EFf_eq_zero_iff** {ξ0 : E} (h0 : ξ0 ≠ 0) (G : Matrix (Fin r) (Fin r) E) (x : Fin r → E) : (∀ y, EFf ξ0 G x y = 0) ↔ ∀ y, herm G x y = 0

theorem **traceFormBilin_separatingLeft_iff** {ξ0 : E} (h0 : ξ0 ≠ 0) (G : Matrix (Fin r) (Fin r) E) : (traceFormBilin ξ0 G).SeparatingLeft ↔ G.det ≠ 0

theorem **traceFormBilin_nondegenerate_iff** {ξ0 : E} (h0 : ξ0 ≠ 0) (G : Matrix (Fin r) (Fin r) E) : (traceFormBilin ξ0 G).Nondegenerate ↔ G.det ≠ 0

theorem **EFf_mulVec_iff** {s : ℕ} {ξ0 : E} (h0 : ξ0 ≠ 0) (G : Matrix (Fin r) (Fin r) E) (G' : Matrix (Fin s) (Fin s) E) (γ : Matrix (Fin r) (Fin s) E) : (∀ x y, EFf ξ0 G (γ *ᵥ x) (γ *ᵥ y) = EFf ξ0 G' x y) ↔ ∀ x y, herm G (γ *ᵥ x) (γ *ᵥ y) = herm G' x y

theorem **trace_mul_traceRep** {K : Type} [Field K] [NumberField K] (f : Module.Dual ℚ K) (a : K) : Algebra.trace ℚ K (a * traceRep f) = f a

theorem **trace_skewGram** {ξ0 : E} (h0 : ξ0 ≠ 0) (B : LinearMap.BilinForm ℚ (Fin r → E)) (a b : Fin r) (t : E) : Algebra.trace ℚ E (t * (-ξ0 * skewGram ξ0 B a b)) = B (Pi.single a t) (Pi.single b 1)

theorem **IsConjAdj.ext** {B B' : LinearMap.BilinForm ℚ (Fin r → E)} (h : IsConjAdj B) (h' : IsConjAdj B') (hh : ∀ a b t, B (Pi.single a t) (Pi.single b 1) = B' (Pi.single a t) (Pi.single b 1)) : B = B'

theorem **traceFormBilin_skewGram** {ξ0 : E} (h0 : ξ0 ≠ 0) {B : LinearMap.BilinForm ℚ (Fin r → E)} (hB : IsConjAdj B) : traceFormBilin ξ0 (skewGram ξ0 B) = B

theorem **EFf_skewGram** {ξ0 : E} (h0 : ξ0 ≠ 0) {B : LinearMap.BilinForm ℚ (Fin r → E)} (hB : IsConjAdj B) (x y : Fin r → E) : EFf ξ0 (skewGram ξ0 B) x y = B x y

theorem **skewGram_traceFormBilin** {ξ0 : E} (h0 : ξ0 ≠ 0) (G : Matrix (Fin r) (Fin r) E) : skewGram ξ0 (traceFormBilin ξ0 G) = G

theorem **eq_skewGram** {ξ0 : E} (h0 : ξ0 ≠ 0) {B : LinearMap.BilinForm ℚ (Fin r → E)} {G : Matrix (Fin r) (Fin r) E} (h : ∀ x y, B x y = EFf ξ0 G x y) : G = skewGram ξ0 B

theorem **skewGram_herm** {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) {B : LinearMap.BilinForm ℚ (Fin r → E)} (hB : IsConjAdj B) (halt : B.IsAlt) (a b : Fin r) : cc (skewGram ξ0 B a b) = skewGram ξ0 B b a

theorem **skewGram_herm_iff** {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) {B : LinearMap.BilinForm ℚ (Fin r → E)} (hB : IsConjAdj B) : (∀ a b, cc (skewGram ξ0 B a b) = skewGram ξ0 B b a) ↔ B.IsAlt

theorem **skewGram_det_ne_zero_iff** {ξ0 : E} (h0 : ξ0 ≠ 0) {B : LinearMap.BilinForm ℚ (Fin r → E)} (hB : IsConjAdj B) : (skewGram ξ0 B).det ≠ 0 ↔ B.Nondegenerate

theorem **existsUnique_herm_EFf** {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) {B : LinearMap.BilinForm ℚ (Fin r → E)} (hB : IsConjAdj B) (halt : B.IsAlt) : ∃! G : Matrix (Fin r) (Fin r) E, (∀ a b, cc (G a b) = G b a) ∧ ∀ x y, B x y = EFf ξ0 G x y

theorem **preserves_iff_herm** {ξ0 : E} (h0 : ξ0 ≠ 0) {B : LinearMap.BilinForm ℚ (Fin r → E)} {G : Matrix (Fin r) (Fin r) E} (hG : ∀ x y, B x y = EFf ξ0 G x y) (γ : GL (Fin r) E) : (∀ x y, B ((γ : Matrix (Fin r) (Fin r) E) *ᵥ x) ((γ : Matrix (Fin r) (Fin r) E) *ᵥ y) = B x y) ↔ ∀ x y, herm G ((γ : Matrix (Fin r) (Fin r) E) *ᵥ x) ((γ : Matrix (Fin r) (Fin r) E) *ᵥ y) = herm G x y

theorem **smul_skewGram_mul** {c : E} (hc : c ≠ 0) (ξ0 : E) (B : LinearMap.BilinForm ℚ (Fin r → E)) : c • skewGram (c * ξ0) B = skewGram ξ0 B

theorem **coe_ringOfIntegersComplexConj** (a : 𝓞 E) : ((IsCMField.ringOfIntegersComplexConj E a : 𝓞 E) : E) = cc (a : E)

theorem **extend_isConjAdj** (h : IsLatticeEmb φ) {β : LinearMap.BilinForm ℤ M} (hβ : IsIntConjAdj E β) : IsConjAdj (h.extend β)

theorem **EFf_gram** (h : IsLatticeEmb φ) {ξ0 : E} (h0 : ξ0 ≠ 0) {β : LinearMap.BilinForm ℤ M} (hβ : IsIntConjAdj E β) (x y : M) : EFf ξ0 (h.gram ξ0 β) (φ x) (φ y) = β x y

theorem **eq_gram** (h : IsLatticeEmb φ) {ξ0 : E} (h0 : ξ0 ≠ 0) {β : LinearMap.BilinForm ℤ M} {G : Matrix (Fin r) (Fin r) E} (hG : ∀ x y, EFf ξ0 G (φ x) (φ y) = β x y) : G = h.gram ξ0 β

theorem **gram_herm** (h : IsLatticeEmb φ) {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) {β : LinearMap.BilinForm ℤ M} (hβ : IsIntConjAdj E β) (halt : ∀ x, β x x = 0) (a b : Fin r) : cc (h.gram ξ0 β a b) = h.gram ξ0 β b a

theorem **gram_det_ne_zero_iff** (h : IsLatticeEmb φ) {ξ0 : E} (h0 : ξ0 ≠ 0) {β : LinearMap.BilinForm ℤ M} (hβ : IsIntConjAdj E β) : (h.gram ξ0 β).det ≠ 0 ↔ Function.Injective β

theorem **EF_int_of_eq** {ξ0 : E} {β : LinearMap.BilinForm ℤ M} {G : Matrix (Fin r) (Fin r) E} (hG : ∀ x y, EFf ξ0 G (φ x) (φ y) = β x y) : ∀ x ∈ LinearMap.range φ, ∀ y ∈ LinearMap.range φ, ∃ k : ℤ, EFf ξ0 G x y = k

theorem **EF_perf_iff** (h : IsLatticeEmb φ) {ξ0 : E} {β : LinearMap.BilinForm ℤ M} {G : Matrix (Fin r) (Fin r) E} (hG : ∀ x y, EFf ξ0 G (φ x) (φ y) = β x y) : (∀ f : LinearMap.range φ →+ ℤ, ∃ x ∈ LinearMap.range φ, ∀ y : LinearMap.range φ, (f y : ℚ) = EFf ξ0 G x y) ↔ Function.Surjective β

theorem **gram_eq_of_mulVec** {s : ℕ} {φ' : M' →ₗ[𝒪 E] (Fin s → E)} (h : IsLatticeEmb φ) (h' : IsLatticeEmb φ') {ξ0 : E} (h0 : ξ0 ≠ 0) {β : LinearMap.BilinForm ℤ M} {β' : LinearMap.BilinForm ℤ M'} (hβ' : IsIntConjAdj E β') {g : M → M'} (hg : ∀ x y, β' (g x) (g y) = β x y) {γ : Matrix (Fin s) (Fin r) E} (hγ : ∀ x, γ *ᵥ φ x = φ' (g x)) : h.gram ξ0 β = γᵀ * h'.gram ξ0 β' * γ.map cc

theorem **isUStab_iff** (h : IsLatticeEmb φ) {ξ0 : E} (h0 : ξ0 ≠ 0) {β : LinearMap.BilinForm ℤ M} {G : Matrix (Fin r) (Fin r) E} (hG : ∀ x y, EFf ξ0 G (φ x) (φ y) = β x y) (γ : GL (Fin r) E) : IsUStab G (LinearMap.range φ) γ ↔ ∃ g : M ≃ₗ[𝒪 E] M, (∀ x, (γ : Matrix (Fin r) (Fin r) E) *ᵥ φ x = φ (g x)) ∧ ∀ x y, β (g x) (g y) = β x y

theorem **latticeEF_apply** (x y : Λ) : ((latticeEF ξ0 G Λ hint x y : ℤ) : 𝕂) = EFf ξ0 G x y

theorem **latticeEF_isIntConjAdj** : IsIntConjAdj E (latticeEF ξ0 G Λ hint)

theorem **latticeEF_self** (hξ : cc ξ0 = -ξ0) (hG : ∀ a b, cc (G a b) = G b a) (x : Λ) : latticeEF ξ0 G Λ hint x x = 0

theorem **gram_latticeEF** (hΛ : IsLattice Λ) (h0 : ξ0 ≠ 0) : (isLatticeEmb_subtype hΛ).gram ξ0 (latticeEF ξ0 G Λ hint) = G

theorem **latticeEF_bijective_iff** (hΛ : IsLattice Λ) (h0 : ξ0 ≠ 0) : Function.Bijective (latticeEF ξ0 G Λ hint) ↔ G.det ≠ 0 ∧ ∀ f : Λ →+ ℤ, ∃ x ∈ Λ, ∀ y : Λ, (f y : 𝕂) = EFf ξ0 G x y

theorem **exists_lattice_EFf** [Module.Finite (𝒪 E) M] [Module.IsTorsionFree (𝒪 E) M] (hr : Module.finrank (𝒪 E) M = r) {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) {β : LinearMap.BilinForm ℤ M} (halt : ∀ x, β x x = 0) (hβ : IsIntConjAdj E β) (hu : Function.Bijective β) : ∃ (φ : M →ₗ[𝒪 E] (Fin r → E)) (G : Matrix (Fin r) (Fin r) E), Function.Injective φ ∧ IsLattice (LinearMap.range φ) ∧ (∀ a b, cc (G a b) = G b a) ∧ G.det ≠ 0 ∧ (∀ x y, EFf ξ0 G (φ x) (φ y) = β x y) ∧ (∀ x ∈ LinearMap.range φ, ∀ y ∈ LinearMap.range φ, ∃ k : ℤ, EFf ξ0 G x y = k) ∧ ∀ f : LinearMap.range φ →+ ℤ, ∃ x ∈ LinearMap.range φ, ∀ y : LinearMap.range φ, (f y : 𝕂) = EFf ξ0 G x y

theorem **host_bar_eq** (H : Host E) (a : 𝒪 E) : H.bar a = IsCMField.ringOfIntegersComplexConj E a

theorem **datum_isIntConjAdj** (D : Datum H) : IsIntConjAdj E D.EF

theorem **datum_exists_isLatticeEmb** (D : Datum H) : ∃ φ : D.La →ₗ[𝒪 E] (Fin (D.p + D.q) → E), IsLatticeEmb φ

theorem **datumEmb_isLatticeEmb** (D : Datum H) : IsLatticeEmb (datumEmb D)

theorem **datumLat_isLattice** (D : Datum H) : IsLattice (datumLat D)

@[simp] theorem **coe_datumEquivLat** (D : Datum H) (x : D.La) : (datumEquivLat D x : Fin (D.p + D.q) → E) = datumEmb D x

theorem **datum_gram_spec** (D : Datum H) {φ : D.La →ₗ[𝒪 E] (Fin r → E)} (h : IsLatticeEmb φ) {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) : (∀ a b, cc (h.gram ξ0 D.EF a b) = h.gram ξ0 D.EF b a) ∧ (h.gram ξ0 D.EF).det ≠ 0 ∧ (∀ x y, EFf ξ0 (h.gram ξ0 D.EF) (φ x) (φ y) = D.EF x y) ∧ (∀ x ∈ LinearMap.range φ, ∀ y ∈ LinearMap.range φ, ∃ k : ℤ, EFf ξ0 (h.gram ξ0 D.EF) x y = k) ∧ ∀ f : LinearMap.range φ →+ ℤ, ∃ x ∈ LinearMap.range φ, ∀ y : LinearMap.range φ, (f y : 𝕂) = EFf ξ0 (h.gram ξ0 D.EF) x y

theorem **datum_exists_lattice_EFf** (D : Datum H) {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) : ∃ (Λ : Submodule (𝒪 E) (Fin (D.p + D.q) → E)) (e : D.La ≃ₗ[𝒪 E] Λ) (G : Matrix (Fin (D.p + D.q)) (Fin (D.p + D.q)) E), IsLattice Λ ∧ (∀ a b, cc (G a b) = G b a) ∧ G.det ≠ 0 ∧ (∀ x y, EFf ξ0 G (e x) (e y) = D.EF x y) ∧ (∀ x ∈ Λ, ∀ y ∈ Λ, ∃ k : ℤ, EFf ξ0 G x y = k) ∧ ∀ f : Λ →+ ℤ, ∃ x ∈ Λ, ∀ y : Λ, (f y : 𝕂) = EFf ξ0 G x y

theorem **datum_rank_eq** (D : Datum H) {φ : D.La →ₗ[𝒪 E] (Fin r → E)} (h : IsLatticeEmb φ) : r = D.p + D.q

theorem **datumGram_herm** (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) (a b : Fin (D.p + D.q)) : cc (datumGram D ξ0 a b) = datumGram D ξ0 b a

theorem **datumGram_det_ne_zero** (h0 : ξ0 ≠ 0) : (datumGram D ξ0).det ≠ 0

theorem **EFf_datumGram** (h0 : ξ0 ≠ 0) (x y : D.La) : EFf ξ0 (datumGram D ξ0) (datumEmb D x) (datumEmb D y) = D.EF x y

theorem **eq_datumGram** (h0 : ξ0 ≠ 0) {G : Matrix (Fin (D.p + D.q)) (Fin (D.p + D.q)) E} (hG : ∀ x y, EFf ξ0 G (datumEmb D x) (datumEmb D y) = D.EF x y) : G = datumGram D ξ0

theorem **datum_EF_int** (h0 : ξ0 ≠ 0) : ∀ x ∈ datumLat D, ∀ y ∈ datumLat D, ∃ k : ℤ, EFf ξ0 (datumGram D ξ0) x y = k

theorem **datum_EF_perf** (h0 : ξ0 ≠ 0) : ∀ f : datumLat D →+ ℤ, ∃ x ∈ datumLat D, ∀ y : datumLat D, (f y : 𝕂) = EFf ξ0 (datumGram D ξ0) x y

theorem **traceFormDual_datumLat** (h0 : ξ0 ≠ 0) : traceFormDual ξ0 (datumGram D ξ0) (datumLat D) = datumLat D

theorem **datum_isUStab_iff** (h0 : ξ0 ≠ 0) (γ : GL (Fin (D.p + D.q)) E) : IsUStab (datumGram D ξ0) (datumLat D) γ ↔ ∃ g : D.La ≃ₗ[𝒪 E] D.La, (∀ x, (γ : Matrix (Fin (D.p + D.q)) (Fin (D.p + D.q)) E) *ᵥ datumEmb D x = datumEmb D (g x)) ∧ ∀ x y, D.EF (g x) (g y) = D.EF x y

@[simp] theorem **val_smul** (a : 𝒪 E) (f : ConjDual E) (x : 𝒪 E) : val (a • f) x = val f (IsCMField.ringOfIntegersComplexConj E a * x)

@[**ext**] theorem ext {f g : ConjDual E} (h : ∀ x, val f x = val g x) : f = g

@[simp] theorem **hypForm_apply** (u v : 𝒪 E × ConjDual E) : hypForm E u v = ConjDual.val v.2 u.1 - ConjDual.val u.2 v.1

theorem **hypForm_self** (u : 𝒪 E × ConjDual E) : hypForm E u u = 0

theorem **hypForm_isIntConjAdj** : IsIntConjAdj E (hypForm E)

theorem **hypForm_bijective** : Function.Bijective (hypForm E)

theorem **finrank_hyp** : Module.finrank (𝒪 E) (𝒪 E × ConjDual E) = 2

theorem **exists_perfect_lattice_rank_two** {ξ0 : E} (h0 : ξ0 ≠ 0) (hξ : cc ξ0 = -ξ0) : ∃ (Λ : Submodule (𝒪 E) (Fin 2 → E)) (G : Matrix (Fin 2) (Fin 2) E), IsLattice Λ ∧ (∀ a b, cc (G a b) = G b a) ∧ G.det ≠ 0 ∧ (∀ x ∈ Λ, ∀ y ∈ Λ, ∃ k : ℤ, EFf ξ0 G x y = k) ∧ ∀ f : Λ →+ ℤ, ∃ x ∈ Λ, ∀ y : Λ, (f y : 𝕂) = EFf ξ0 G x y

Development.AbelianVarietyTheory.Definitions.Jets.CmLatticeCommutatorPairingPowEqOneOfGoodLift

theorem **logS_eq** : (logS : ℂ_[q]⟦X⟧) = PowerSeries.log ℂ_[q]

theorem **mulS_eq** (a : ℤ_[q]) : mulS a = PowerSeries.map ((algebraMap ℚ_[q] ℂ_[q]).comp PadicInt.Coe.ringHom) (binomialSeries ℤ_[q] a) - 1

theorem **eis_eq** {d : ℕ} {L : PeriodPair} (h : IsOK d L) {k : ℕ} (hk : 0 < k) : eis d k = L.G (2 * k)

theorem **_root_.GlobalTheory.MomOne.Basic.discr_eq** {d : ℕ} {L : PeriodPair} (h : _root_.GlobalTheory.MomOne.IsOK d L) : _root_.GlobalTheory.MomOne.discr d = L.g₂ ^ 3 - 27 * L.g₃ ^ 2 ∧ _root_.GlobalTheory.MomOne.discr d ≠ 0

theorem **_root_.GlobalTheory.MomOne.Basic.isOK_okPair** (d : ℕ) (hd : 0 < d) : _root_.GlobalTheory.MomOne.IsOK d (_root_.GlobalTheory.MomOne.Basic.okPair d hd)

theorem **mk_le_of_denseRange** {A X : Type} [MetricSpace X] (f : A → X) (hf : DenseRange f) (hA : #A ≤ 𝔠) : #X ≤ 𝔠

private theorem **cardinal_le_seq_of_denseRange** {X Y : Type} [TopologicalSpace X] [FrechetUrysohnSpace X] [T2Space X] (i : Y → X) (hi : DenseRange i) : #X ≤ #(ℕ → Y)

private theorem **cardinal_padicComplex_eq_continuum** (q : ℕ) [Fact q.Prime] : #ℂ_[q] = 𝔠

theorem **_root_.GlobalTheory.MomOne.Basic.nonempty_iso** (q : ℕ) [Fact q.Prime] : Nonempty (ℂ ≃+* ℂ_[q])

theorem **exists_ok_of** (ι : ℂ ≃+* ℂ_[q]) (W : WeierstrassCurve ℂ_[q]) (h1 : ‖W.a₁‖ ≤ 1) (h2 : ‖W.a₂‖ ≤ 1) (h3 : ‖W.a₃‖ ≤ 1) (h4 : ‖W.a₄‖ ≤ 1) (h6 : ‖W.a₆‖ ≤ 1) (hΔ : ‖W.Δ‖ = 1) : ∃ (L : PeriodPair) (c : ℂ), Mdl.Ok (ι, L, c, W)

theorem **_root_.GlobalTheory.MomOne.Basic.norm_nat_eq_one** {m : ℕ} (h : Nat.Coprime q m) : ‖(m : ℂ_[q])‖ = 1

theorem **exists_ok** (q : ℕ) [Fact q.Prime] : ∃ M : Mdl q, M.Ok

theorem **hasDerivAt_u** (M : Mdl q) (hc : M.c ≠ 0) : HasDerivAt M.u M.c 0

theorem **exists_chart** (M : Mdl q) (h : M.Ok) : ∃ ℓ, M.Chart ℓ ∧ hat M.ι ℓ M.u = X

theorem **exists_model_chart** (q : ℕ) [Fact q.Prime] : ∃ (M : Mdl q) (ℓ : ℂ_[q]⟦X⟧), M.Chart ℓ

theorem **_root_.GlobalTheory.MomOne.Basic.cmLatticeCommutatorPairing_pow_eq_one_of_good_lift** (ι : ℂ ≃+* ℂ_[q]) {d n s : ℕ} (hd : 0 < d) {y : ℂ} (h : _root_.GlobalTheory.MomOne.Good ι d n s y) : ι (_root_.GlobalTheory.MomOne.kap1 d 1 y) ^ (q ^ n) = 1

theorem **exists_good** [Fact (Nat.Prime 2)] (ι : ℂ ≃+* ℂ_[2]) : ∃ y, Good ι 7 1 1 y ∧ kap1 7 1 y = -1

theorem **shiftS_X** (ζ : ℂ_[q]) : shiftS ζ X = C ζ * (1 + X) - 1

theorem **_root_.GlobalTheory.MomOne.Basic.cmLatticeHermitianForm_mul_left** (d : ℕ) (a z w : ℂ) : _root_.GlobalTheory.MomOne.h1 d (a * z) w = a * _root_.GlobalTheory.MomOne.h1 d z w

theorem **riemannForm_im_eq_neg_one** {d : ℕ} (hd : 0 < d) : (h1 d 1 (wK d)).im = -1

theorem **_root_.GlobalTheory.MomOne.Basic.shiftS_one** (f : ℂ_[q]⟦X⟧) : _root_.GlobalTheory.MomOne.shiftS 1 f = f

Development.AbelianVarietyTheory.Definitions.Jets.CmQuadraticFormEqAlternatingForm

theorem **T0_pos** (Tp : ParamsT D L Pm) : 0 < Tp.T0

Development.AbelianVarietyTheory.Definitions.Jets.ColValMonomial

theorem **Pt1.germ_zero** (q : ℕ) {r : PS V 1 y.d} (hr : JetSer.IsSer1 y.x _ (y.ρ.s (fun _ => y.j)).val r) : y.Germ q 0 0

theorem **Pt1.Red.mono** {y : Pt1 E H V F} {r : ℕ} {b : Module.Basis (Fin r) ℤ y.P.D.toPre.X} {Cc Cc' M₀ : ℝ} (h : y.Red r b Cc M₀) (hC : Cc ≤ Cc') : y.Red r b Cc' M₀

theorem **Pt1.Transl.mul** {y : Pt1 E H V F} {q : ℕ} {c : 𝒪 E} {s h : Fam y.P.D.toPre F q} (ht : y.Transl q c s h) (c' : 𝒪 E) : y.Transl q (c' * c) s h

theorem **Pt1.colVal_monomial** (mo : Λ × Fin n →₀ ℕ) (c : F) (f : Fin n → PS F 1 y.d) : y.colVal h q e (MvPolynomial.monomial mo c) f = c * ∏ b, (letI

theorem **Pt1.colVal_zero** (f : Fin n → PS F 1 y.d) : y.colVal h q e (0 : MvPolynomial (Λ × Fin n) F) f = 0

theorem **Pt1.colVal_add** (P P' : MvPolynomial (Λ × Fin n) F) (f : Fin n → PS F 1 y.d) : y.colVal h q e (P + P') f = y.colVal h q e P f + y.colVal h q e P' f

theorem **Pt1.colVal_smul** (c : F) (P : MvPolynomial (Λ × Fin n) F) (f : Fin n → PS F 1 y.d) : y.colVal h q e (c • P) f = c * y.colVal h q e P f

theorem **Pt1.colVal_sum** {I : Type} (S : Finset I) (c : I → F) (P : I → MvPolynomial (Λ × Fin n) F) (f : Fin n → PS F 1 y.d) : y.colVal h q e (∑ i ∈ S, c i • P i) f = ∑ i ∈ S, c i * y.colVal h q e (P i) f

theorem **minorMonomial_zero** (k : Type) [CommRing k] {T : Type} [Fintype T] [DecidableEq T] {r : T → ℕ} {n : ℕ} (col : ∀ τ, Fin (r τ) → Fin n) : minorsP k col (fun _ _ => 0) = 1

Development.AbelianVarietyTheory.Definitions.Jets.CubWeilOneIffPairOne

theorem **cubWeil_one_iff_pairOne** : GlobalTheory.UU.CubWeil c N x y 1 ↔ x ^ N = 1 ∧ y ^ N = 1 ∧ PairOne c N x y

theorem **PairOne.cubWeil** (h : PairOne c N x y) (hx : x ^ N = 1) (hy : y ^ N = 1) : GlobalTheory.UU.CubWeil c N x y 1

theorem **PairOne.of_cubWeil** (h : GlobalTheory.UU.CubWeil c N x y 1) : PairOne c N x y

theorem **PairOne.swap** (h : PairOne c N x y) : PairOne c N y x

theorem **PairOne.precomp** (h : PairOne c N x y) (hx : x ^ N = 1) (hy : y ^ N = 1) {T' : Over S} (σ : T' ⟶ T) : PairOne c N (σ ≫ x) (σ ≫ y)

theorem **pairOne_trivCub** (N : ℕ) (x y : T ⟶ G) : PairOne (GlobalTheory.UU.trivCub G) N x y

theorem **not_pairOne_signCube** : ¬ PairOne GlobalTheory.UU.SignCube.cub 2 (GlobalTheory.UU.SignCube.p₁ (𝟙_ _)) (GlobalTheory.UU.SignCube.p₂ (𝟙_ _))

theorem **AlphaPt.pt_zero_eq_one** {H : Over S} [GrpObj H] (α : AlphaPt H κ l) (B : Type) [CommRing B] [Algebra k B] (h : (0 : B) ^ l = 0) : α.pt B 0 h = 1

theorem **AlphaPt.pt_natCast_mul** {H : Over S} [GrpObj H] (α : AlphaPt H κ l) (hl : l ≠ 0) (B : Type) [CommRing B] [Algebra k B] (t : B) (h : t ^ l = 0) (n : ℕ) (hn : ((n : B) * t) ^ l = 0) : α.pt B ((n : B) * t) hn = α.pt B t h ^ n

theorem **AlphaPt.pt_pow_eq_one** [CharP k l] {H : Over S} [GrpObj H] (α : AlphaPt H κ l) (hl : l ≠ 0) (B : Type) [CommRing B] [Algebra k B] (t : B) (h : t ^ l = 0) : α.pt B t h ^ l = 1

theorem **AlphaPt.pairsTriv_iff_cubWeil** [CharP k l] {L : RigLB G} (c : CubicalStr G L) (α : AlphaPt G κ l) (hl : l ≠ 0) : α.PairsTriv c ↔ ∀ (B : Type) [CommRing B] [Algebra k B] [Module.Finite k B] (aug : B →ₐ[k] k), (∀ b, aug b = 0 → IsNilpotent b) → ∀ (s : B) (_ : s ^ 2 = 0) (h : s ^ l = 0) (y : overAt κ B ⟶ G), Formalism.L2.Pj.homAt κ aug ≫ y = 1 → y ^ l = 1 → GlobalTheory.UU.CubWeil c l (α.pt B s h) y 1

theorem **eps_pow_eq_zero_iff** : (DualNumber.eps : DualNumber k) ^ l = 0 ↔ 2 ≤ l

theorem **nonDegAt_of_le_one** {L : RigLB G} (c : CubicalStr G L) (κ : Sp k ⟶ S) (hl : l ≤ 1) : NonDegAt c κ l

theorem **alphaAdd_pt_eq_one_iff** (κ : Sp k ⟶ S) (B : Type) [CommRing B] [Algebra k B] (t : B) (h : t ^ l = 0) : (alphaAdd κ l).pt B t h = 1 ↔ t = 0

theorem **not_tanZero_alphaAdd** (κ : Sp k ⟶ S) (hl : 2 ≤ l) : ¬ (alphaAdd κ l).TanZero

theorem **AlphaPt.pairsTriv_trivCub** (α : AlphaPt G κ l) : α.PairsTriv (GlobalTheory.UU.trivCub G)

theorem **not_nonDegAt_alphaAdd** (κ : Sp k ⟶ S) (hl : 2 ≤ l) : ¬ NonDegAt (GlobalTheory.UU.trivCub (GlobalTheory.IndefiniteRowSideFullHecke.SemiAbelianAxisTangent.vecObj S 1)) κ l

Development.AbelianVarietyTheory.Definitions.Jets.DepthNumbersThresholdEq

theorem **PhiC.conjIn** {N : NumDatum} {l : ℕ} (h : PhiC N l) : ConjIn N l

theorem **even_ordP_of_not_conjIn** {N : NumDatum} {l : ℕ} (hI : Inert N l) (hC : ¬ ConjIn N l) : Even (N.ordP l)

theorem **ne_two_of_not_small** {N : NumDatum} {l : ℕ} (h : l ∉ N.H.small) : l ≠ 2

Development.AbelianVarietyTheory.Definitions.Jets.EmbShortThLeThetaPlain

theorem **spanK_mono** {s t : Set C} (h : s ⊆ t) : spanK z s ≤ spanK z t

theorem **mem_spanK** {s : Set C} {f : C} (hf : f ∈ s) : z.symm (1 ⊗ₜ[F] f) ∈ spanK z s

theorem **spanK_fg_of_fg** [Module (𝒪 F) C] [IsScalarTower (𝒪 F) F C] {N : Submodule (𝒪 F) C} (hN : N.FG) : (spanK z (N : Set C)).FG

theorem **runScalar_kappa_ne_zero** (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) {p : ℕ} (hp : 0 < p) (ka wL wF : WeightFamilies.Wt D j) (T : ℕ) : (scalOfP B hinj EX p ka wL wF T).sK ≠ 0

theorem **shortTh_fg_of_Thc_le** (X : WeightAlgebra.PJ D B P6 j) (w : WeightFamilies.Wt D j) {s : 𝒪 B.KO} (hs : s ≠ 0) (hle : X.Thc s w ≤ B.Mlat j w) (hM : (B.Mlat j w).FG) : (X.shortTh w).FG

theorem **shortTh_fg** (X : WeightAlgebra.PJ D B P6 j) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) {p : ℕ} (hp : 0 < p) (ka wL wF : WeightFamilies.Wt D j) (T : ℕ) (hM : Submodule.IsLattice B.KO (B.Mlat j ka)) (h2 : WeightAlgebra.H2 X (scalOfP B hinj EX p ka wL wF T) ka wL wF) : (X.shortTh ka).FG

theorem **ProductsShort.mem** {X : WeightAlgebra.PJ D B P6 j} {ka wL wF : WeightFamilies.Wt D j} {T : ℕ} (h : ProductsShort X ka wL wF T) {F : B.Cusp j ka} (hF : X.inc ka F ∈ thetaPlain X ka wL wF T) : F ∈ X.shortTh ka

theorem **FormTieK.toFormTie** {X : WeightAlgebra.PJ D B P6 j} {hinj : Function.Injective D.what₂.OtoLL} {EX : ErrX D} {p : ℕ} {ka wL wF : WeightFamilies.Wt D j} {T : ℕ} {O : OnObj S P oc} {v : HeightOneSpectrum (𝒪 K)} {i : Bs.J} {z : (Fin (O.nZ i) → K) ≃ₗ[K] K ⊗[B.KO] B.Cusp j ka} {act : P → B.Cusp j ka → B.Cusp j ka} (h : FormTieK X hinj EX p ka wL wF T O v i z act) (hfg : (X.shortTh ka).FG) (hpr : ProductsShort X ka wL wF T) : FormTie X hinj EX p ka wL wF T O v i z act

theorem **FormTieK.rho_tmul** {X : WeightAlgebra.PJ D B P6 j} {hinj : Function.Injective D.what₂.OtoLL} {EX : ErrX D} {p : ℕ} {ka wL wF : WeightFamilies.Wt D j} {T : ℕ} {O : OnObj S P oc} {v : HeightOneSpectrum (𝒪 K)} {i : Bs.J} {z : (Fin (O.nZ i) → K) ≃ₗ[K] K ⊗[B.KO] B.Cusp j ka} {act : P → B.Cusp j ka → B.Cusp j ka} (h : FormTieK X hinj EX p ka wL wF T O v i z act) (g : P) (a : K) (F : B.Cusp j ka) : O.ρ i g (z.symm (a ⊗ₜ F)) = z.symm (a ⊗ₜ act g F)

theorem **exists_formTieK** {B : WeightFamilies.Prov D} {P6 : ModuliIntegralCharts.Prov D} (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) {K : Type} [Field K] [NumberField K] [StarRing K] [Algebra B.KO K] {Bs : DefBase K} {Df : DefSide Bs} {S : IndefData Df} {P : Type} [Group P] [Finite P] {oc : OnConst} (O : OnObj S P oc) (jj : Bs.J → ℕ) (X : HeightOneSpectrum (𝒪 K) → ∀ i, WeightAlgebra.PJ D B P6 (jj i)) (ρ : HeightOneSpectrum (𝒪 K) → ℕ) (ka wL wF : ∀ i, WeightFamilies.Wt D (jj i)) (T : ℕ) (ρ : ∀ i, Representation B.KO P (B.Cusp (jj i) (ka i))) : ∃ (O' : OnObj S P oc) (z : ∀ i, (Fin (O'.nZ i) → K) ≃ₗ[K] K ⊗[B.KO] B.Cusp (jj i) (ka i)), (∀ i, O'.nZ i = Module.finrank B.KO (B.Cusp (jj i) (ka i))) ∧ O'.pde = O.pde ∧ O'.U = O.U ∧ ∀ v i, FormTieK (X v i) hinj EX (p v) (ka i) (wL i) (wF i) T O' v i (z i) (fun g F => ρ i g F)

theorem **OneCover.eq_withCoeff** {T : Top D} {j : ℕ} (h : OneCover T j) (n n' : ℕ) : ∃ φ, T.dδ j n' = withCoeff (T.dδ j n) φ

Development.AbelianVarietyTheory.Definitions.Jets.ExistsSecTieFOfTies

theorem **IsSymAct.pow_eq_one** {N : ℕ} {T : Over X} {b : T ⟶ A} {s s' : LB.Sec (LPow A L.L (N : ℤ) T)} (h : IsSymAct A L N b s s') : b ^ N = 1

theorem **IsTransPull.pow_eq_one** {φ : A ⟶ A} {N : ℕ} {T : Over X} {s : LB.Sec (LPow A L.L 1 T)} {b : T ⟶ A} {hh : LB.Sec (LPow A L.L (N : ℤ) T)} (h : IsTransPull A L φ N s b hh) : b ^ N = 1

theorem **PtTieF.base** {x : TFK (R

theorem **SecTieF.base** {q : ℕ} {σ : P.U.toUnifPre.SecGF F q} {hh : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L (q : ℤ) (ptO Xo xF))} (h : SecTieF I P F xF q σ hh) : xF = spMap (algebraMap T.Fr F) » (letI

theorem **nrm_zero** (Dw : Datum H) (c : Coeff l) (α : 𝒪 E) : nrm Dw c 0 α = 1

theorem **runNum_cc** (D : D7) (L : SizeData₂ D) {j : ℕ} (w : WeightFamilies.Wt D j) (C2 A2 CR : ℝ) (h1 : w.LastPos) (h2 : D.R.f * D.R.r * (D.R.r + 1) ≤ w.n) (h3 : ∀ i, 4 * WeightAlgebra.DF w + 4 ≤ D.host.g i) (h4 : rr0 D ≤ WeightFamilies.RRv D CR) (h5 : (2 * D.R.f * D.R.m : ℝ) ^ 4 * w.n + 2 * (2 * D.R.f * D.R.m : ℝ) ^ 3 ≤ WeightFamilies.PPv D L C2 A2) (h6 : ∀ i, (WeightAlgebra.DF w : ℝ) * (2 * Real.sqrt (2 * D.R.f * D.R.m)) < (D.host.g i : ℝ)) : RunNum D w (WeightAlgebra.cc D L C2 A2 CR w)

Development.AbelianVarietyTheory.Definitions.Jets.FactorNonneg

theorem **Box.strictAnti** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) : StrictAnti L

theorem **prod_erase_eq** {n : ℕ} (f : Fin (n + 1) → ℝ) (l : Fin (n + 1)) : ∏ j ∈ univ.erase l, f j = ∏ j : Fin n, f (l.succAbove j)

theorem **factor_nonneg** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) (l : Fin (n + 1)) (j : Fin n) (e : ℤ) (he0 : 0 ≤ e) (he1 : e ≤ 1) : 0 ≤ ((M j : ℝ) - e - L l) / ((L (l.succAbove j) : ℝ) - L l)

theorem **placeWeight_nonneg** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) (l : Fin (n + 1)) : 0 ≤ wP L M l

theorem **wM_nonneg** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) (l : Fin (n + 1)) : 0 ≤ wM L M l

theorem **denom_ne_zero** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) (l : Fin (n + 1)) : ∏ j ∈ univ.erase l, ((L j : ℝ) - (L l : ℝ)) ≠ 0

theorem **weight_eq_abs_prod_div_prod** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) (β : ℝ) (u : Fin (n + 1) → ℝ) (ρ : Fin n → ℝ) (hu : ∀ j, (L j : ℝ) = u j - (n : ℝ) / 2 - β) (hρ : ∀ j, (M j : ℝ) = ρ j - ((n : ℝ) - 1) / 2 - β) (l : Fin (n + 1)) : wP L M l = (∏ i, |u l + 1 / 2 - ρ i|) / ∏ i ∈ univ.erase l, |u l - u i|

theorem **wM_eq_abs** {n : ℕ} {L : Fin (n + 1) → ℤ} {M : Fin n → ℤ} (h : Box L M) (β : ℝ) (u : Fin (n + 1) → ℝ) (ρ : Fin n → ℝ) (hu : ∀ j, (L j : ℝ) = u j - (n : ℝ) / 2 - β) (hρ : ∀ j, (M j : ℝ) = ρ j - ((n : ℝ) - 1) / 2 - β) (l : Fin (n + 1)) : wM L M l = (∏ i, |u l - 1 / 2 - ρ i|) / ∏ i ∈ univ.erase l, |u l - u i|

theorem **example_wP** : Box ![3, -1] ![1] ∧ wP ![3, -1] ![1] 0 = 3 / 4 ∧ wP ![3, -1] ![1] 1 = 1 / 4

theorem **example_wM** : wM ![3, -1] ![1] 0 = 1 / 2 ∧ wM ![3, -1] ![1] 1 = 1 / 2

theorem **rank_one** (L : Fin 1 → ℤ) (M : Fin 0 → ℤ) : wP L M 0 = 1 ∧ wM L M 0 = 1

Development.AbelianVarietyTheory.Definitions.Jets.IsAppBox

theorem **isApp_box** {R : Type} [CommRing R] [IsDomain R] {D D₂ D' : DegPre R} {n₁ n₂ : ℕ} (pD₂ : IsPowDeg D₂ D n₂) (pD : IsPowDeg D' D (n₁ + n₂)) (F : Type) [Field F] [Algebra (FractionRing R) F] (k : ℕ) (a : Fin n₁ → Fam D F k) (u : Fin n₂ → Fam D F k) (sb : Fam D₂ F k) (S : Fam D' F k) (h₂ : IsBoxFam pD₂ F k sb u) (h : IsBoxFam pD F k S (Fin.append a u)) : IsApp pD₂ pD F k a sb S

Development.AbelianVarietyTheory.Definitions.Jets.NotRunTiesZeroSteps

theorem **runTie_retie** (O : OnObj S P oc) (ι : Bs.J → ℕ) (hι : ∀ j, InR D (ι j)) (hs : ∀ a : 𝒪 D.host.E, star a = T.Q.H.bar a) (hc : ∀ a : 𝒪 D.host.E, star (algebraMap (𝒪 D.host.E) K a) = algebraMap (𝒪 D.host.E) K (T.Q.H.bar a)) : RunTie T EX Tp hK (retie T EX Tp hK O ι) ι where

theorem **tiePrimes** (h : RunTie T EX Tp hK O ι) : DepthCond.TiePrimes O (numOf D L)

theorem **tieFK** (h : RunTie T EX Tp hK O ι) : DepthCond.TieFK O (numOf D L)

theorem **tieT** (h : RunTie T EX Tp hK O ι) : O.T = (((depOf T EX Tp hK).Q.T : ℕ) : ℝ)

theorem **b_eq** (h : RunTie T EX Tp hK O ι) : O.b = L.sc.b

theorem **wt_eq'** (h : RunTie T EX Tp hK O ι) (j : Bs.J) : O.wt j = 2 ∗ O.f ∗ O.b

theorem **th_zero** (h : RunTie T EX Tp hK O ι) : O.th 0 = 0 ∧ O.th 1 = 0

theorem **not_runTies_zeroSteps** (h : RunTie T EX Tp hK O ι) : ¬ RunTie T EX Tp hK ({ O with T

theorem **numOf_w** (L : SizeData₂ D) {j : ℕ} (hj : InR D j) (z : RPl D.R) : (numOf D L).w j z = wj D j (plOf D z)

theorem **numOf_aPlus** (L : SizeData₂ D) (j : ℕ) (z : RPl D.R) (i : ℕ) : (numOf D L).aPlus j z i = DV.aPlusQ D j (plOf D z) i

theorem **numOf_aMinus** (L : SizeData₂ D) (j : ℕ) (z : RPl D.R) (i : ℕ) : (numOf D L).aMinus j z i = DV.aMinusQ D j (plOf D z) i

theorem **eTau_plOf_false** (t : D.R.A) : (D.eTau (plOf D (false, t)) : D.R.GG) = D.R.tau1 t

theorem **eTau_plOf_true** (t : D.R.A) : (D.eTau (plOf D (true, t)) : D.R.GG) = D.R.tau2 t

theorem **localExponent_small_prime** (T : Core D) (EX : ErrX D) (Tp : ParamsT D L Pm) (hK : DV.KapDef D) {j : ℕ} (w : WeightFamilies.Wt D j) {p : ℕ} (hp : p ∈ D.host.small) : (depOf T EX Tp hK).Q.cP (WeightAlgebra.DF w) p = padicValNat p D.host.QS ∗ c2 D w

theorem **localExponent_two** (T : Core D) (EX : ErrX D) (Tp : ParamsT D L Pm) (hK : DV.KapDef D) {j : ℕ} (w : WeightFamilies.Wt D j) : (depOf T EX Tp hK).Q.cP (WeightAlgebra.DF w) 2 = padicValNat 2 D.host.QS ∗ c2 D w

theorem **hostMap_conj** (T : Core D) (a : 𝒪 D.host.E) : star (hostMap D a) = hostMap D (T.Q.H.bar a)

theorem **intMap_integral** (T : Core D) : letI

theorem **exists_runTie** (T : Core D) (EX : ErrX D) (Tp : ParamsT D L Pm) (hK : DV.KapDef D) {Bs : DefBase D.what₂.LL} {Df : DefSide Bs} {S : IndefData Df} {P : Type} [Group P] [Finite P] {oc : OnConst} (O : OnObj S P oc) (ι : Bs.J → ℕ) (hι : ∀ j, InR D (ι j)) : letI

theorem **exists_runTie_toy** (T : Core D) (EX : ErrX D) (Tp : ParamsT D L Pm) (hK : DV.KapDef D) (v0 : InfinitePlace D.what₂.LL) (Tw0 : Finset (HeightOneSpectrum (𝒪 D.what₂.LL))) (j : ℕ) (hj : InR D j) : letI

theorem **inR_one** (D : D7) : InR D 1

Development.AbelianVarietyTheory.Definitions.Jets.PowerSeriesHasSumAt

theorem **nMul_zero** {R : Type∗} [CommRing R] (F : FormalGroup R) : nMul F 0 = 0

theorem **nMul_succ** {R : Type∗} [CommRing R] (F : FormalGroup R) (n : ℕ) : nMul F (n + 1) = MvPowerSeries.subst (![nMul F n, PowerSeries.X] : Fin 2 → MvPowerSeries Unit R) F.toPowerSeries

Development.AbelianVarietyTheory.Definitions.Jets.SurjectiveAppTopSqCube

theorem **surjective_appTop_sq_cube** {R : Type} [CommRing R] [IsDomain R] {I : Ideal R} {D : DegData R I} {A : PolAV (Spec (CommRingCat.of (FractionRing R)))} {G : Over (Spec (CommRingCat.of R))} [GrpObj G] [IsCommMonObj G] (U : UnifPre I D A G) : Function.Surjective (G ⊗ G).hom.appTop.hom ∧ Function.Surjective (G ⊗ G ⊗ G).hom.appTop.hom

theorem **Pt1.exists_frameSer** : ∃ r : PS V 1 y.d, JetSer.IsSer1 y.x _ (y.ρ.s (fun _ => y.j)).val r

theorem **Pt1.Transl.smul** {y : Pt1 E H V F} {q : ℕ} (hq : 1 ≤ q) {c : 𝒪 E} {s h : Fam y.P.D.toPre F q} (ht : y.Transl q c s h) {ξ : F} {n : ℕ} (hn : 1 ≤ n) (hξ : ξ ^ n = 1) : y.Transl q c s (ξ • h)

Development.AbelianVarietyTheory.Definitions.KeyFormula.GeomIntSelf

theorem **per_add** {g : ℕ} (τ : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g ⊕ Fin g → ℤ) : per τ (a + b) = per τ a + per τ b

Development.AbelianVarietyTheory.Definitions.KeyFormula.KeyRootOfKeyFormula

theorem **keyFormula_iff** : KeyFormula ↔ ∃ κ : Rule, Homog (fun _ => 1) κ ∧ Funct κ ∧ UnitSp κ ∧ ThetaOne κ ∧ ArchNorm (fun _ => 1) κ

Development.AbelianVarietyTheory.Definitions.KeyFormula.KeyRuleOneOfKeyFormula

theorem **smul_one_row** {g : ℕ} (c : ℂ) (i : Fin g) : c • (1 : Matrix (Fin g) (Fin g) ℂ) i = Pi.single i c

theorem **normalAt_one_iff** (x : KeyIn ℂ) (τ : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (a b : Fin x.P.g → ℂ) : NormalAt x τ 1 a b 1 ↔ ∃ u : HolUnif x.P.A x.P.L x.P.g, τ.IsSymm ∧ (τ.map Complex.im).PosDef ∧ (∀ z, z ∈ u.Λ ↔ ∃ m n : Fin x.P.g → ℤ, z = (fun i => (m i : ℂ)) + τ.mulVec (fun i => (n i : ℂ))) ∧ u.IsRig x.rig ∧ u.IsFn x.th (thetaChar τ a b) ∧ ∀ i, u.IsDiff (Pi.single i (2 ∗ Real.pi ∗ Complex.I)⁻¹) (x.v i)

theorem **NormalAt.mem_siegelH** {x : KeyIn ℂ} {τ M : Matrix (Fin x.P.g) (Fin x.P.g) ℂ} {a b : Fin x.P.g → ℂ} {c : ℂ} (h : NormalAt x τ M a b c) : τ ∈ ReciprocityTheory.PTheta.SiegelH (Fin x.P.g)

theorem **ClauseH.mono** {R : Type} [CommRing R] {N : ℕ → ℕ} {g₀ g₁ : ℕ} {κ : KeyIn R → Rˣ} (h : ClauseH R N g₀ κ) (hg : g₀ ≤ g₁) : ClauseH R N g₁ κ

theorem **KeyRule.mono** {N : ℕ → ℕ} {g₀ g₁ : ℕ} {S S' : ∀ g : ℕ, (Fin g → ℂ) → (Fin g → ℂ) → Prop} (h : KeyRule N g₀ S) (hg : g₀ ≤ g₁) (hS : ∀ g a b, g₁ ≤ g → S' g a b → S g a b) : KeyRule N g₁ S'

theorem **keyRule_one_of_keyFormula** (h : KeyFormula) : KeyRule (fun _ => 1) 1 charHalf

theorem **chiAt_iff_thetaLaw** (g : ℕ) (sp : Matrix (Fin g ⊕ Fin g) (Fin g ⊕ Fin g) ℤ) : ChiAt g sp ↔ ∀ n n' : Fin g → ℤ, ∃ k k' : Fin g → ℤ, (∀ i, k i = 0 ∨ k i = 1) ∧ (∀ i, k' i = 0 ∨ k' i = 1) ∧ ThetaLaw g sp n n' k k'

Development.AbelianVarietyTheory.Definitions.LatticeFrames.AdmissiblePeriodOfPeriodInt

theorem **vGam_eq** (w : D.what₂.places.Pl) : vGam D u w = WeilJacquetShalikaLaw.vDG D u w

theorem **colGe_iff** (w : D.what₂.places.Pl) (x : ((D.fm u).lat 0).V) (s : ℝ) : ColGe D u w x s ↔ WeilJacquetShalikaLaw.ColGe0 D u w x s

theorem **nprime_iff** (L : SizeData₂ D) : DV.NprimeDef D L ↔ ∀ u, (D.div.Nu u).L = shift D u L.supp (fun w => ⌊DV.ur D w (L.sg u w)⌋) (L.NuLat u)

theorem **admissiblePeriod_of_periodInt** (h : ∀ u, PeriodInt D u) : WeightDegreeSizes.PeriodClassSumWeyl D

theorem **second_hypothesis_of_lifting** (L : SizeData₂ D) (hμ : ∀ w, D.what₂.muAt w = D.what₂.mu0) (h : ∀ u w, PerGeOn D u (L.NuLat u) w (((⌊DV.ur D w (L.sg u w)⌋ : ℤ) : ℝ)) : LocalQuadraticParts.adm2 D L

theorem **valGe_emb** (w : D.what₂.places.Pl) (y : D.what₂.LL) (s : ℝ) : ValGe D w (D.what₂.embLL y) s ↔ (y = 0 ∨ s ≤ (vAdd (D.plEq w) y : ℝ))

theorem **vAdd_resChar** (w : D.what₂.places.Pl) : vAdd (D.plEq w) ((D.what₂.places.resChar w : ℕ) : D.what₂.LL) = (D.what₂.places.ramIdx w : ℕ)

theorem **valGe_resChar** (w : D.what₂.places.Pl) (s : ℝ) : ValGe D w ((D.what₂.places.resChar w : ℕ) : ℂ) s ↔ s ≤ (D.what₂.places.ramIdx w : ℝ)

theorem **vAdd_inv** (w : D.what₂.places.Pl) (y : D.what₂.LL) : vAdd (D.plEq w) y⁻¹ = − vAdd (D.plEq w) y

theorem **not_valGe_inv** (w : D.what₂.places.Pl) : ¬ ValGe D w ((D.what₂.places.resChar w : ℕ) : ℂ)⁻¹ 0

theorem **not_valGe_of_notMem** (w : D.what₂.places.Pl) {z : ℂ} (hz : z ∉ Set.range D.what₂.embLL) (s : ℝ) : ¬ ValGe D w z s

theorem **valGe_ceil** (w : D.what₂.places.Pl) (z : ℂ) (s : ℝ) : ValGe D w z s ↔ ValGe D w z ((⌈s⌉ : ℤ) : ℝ)

theorem **valGe_mono** {w : D.what₂.places.Pl} {z : ℂ} {s t : ℝ} (hst : t ≤ s) (h : ValGe D w z s) : ValGe D w z t

theorem **valGe_mul** {w : D.what₂.places.Pl} {z z' : ℂ} {s t : ℝ} (h : ValGe D w z s) (h' : ValGe D w z' t) : ValGe D w (z * z') (s + t)

theorem **valGe_add** {w : D.what₂.places.Pl} {z z' : ℂ} {s : ℝ} (h : ValGe D w z s) (h' : ValGe D w z' s) : ValGe D w (z + z') s

theorem **valGe_of_mem_intW** {w : D.what₂.places.Pl} (a : intW D.what₂.places w) : ValGe D w (D.what₂.embLL a) 0

theorem **pairInt_of** (h1 : PtInt D u) (h2 : PeriodInt D u) (w : D.what₂.places.Pl) : PairInt D u w

theorem **perGeOn_of** {N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V} {w : D.what₂.places.Pl} {s : ℝ} (h1 : ColGeOn D u N w s) (h2 : PairInt D u w) : PerGeOn D u N w ((D.what₂.places.ramIdx w : ℝ) * (s − vGam D u w))

theorem **per_smul** (a : D.what₂.LL) (φ : (D.div.Nu u).V) (F : D.what₂.Amb D.what₂.mu0) : D.what₂.perCol u D.what₂.mu0 (D.jN u (a • φ)) F = D.what₂.embLL a * D.what₂.perCol u D.what₂.mu0 (D.jN u φ) F

theorem **per_add** (φ ψ : (D.div.Nu u).V) (F : D.what₂.Amb D.what₂.mu0) : D.what₂.perCol u D.what₂.mu0 (D.jN u (φ + ψ)) F = D.what₂.perCol u D.what₂.mu0 (D.jN u φ) F + D.what₂.perCol u D.what₂.mu0 (D.jN u ψ) F

theorem **int_of_valGe** {z : ℂ} (h : ∀ w, ValGe D w z 0) (w0 : D.what₂.places.Pl) : ∃ y : 𝒪 D.what₂.LL, D.what₂.embLL (y : D.what₂.LL) = z

theorem **valGe_of_int** (w : D.what₂.places.Pl) (y : 𝒪 D.what₂.LL) : ValGe D w (D.what₂.embLL (y : D.what₂.LL)) 0

theorem **nonempty_place** : Nonempty D.what₂.places.Pl

theorem **perIntOn_iff** (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PerIntOn D u N ↔ ∀ w, PerGeOn D u N w 0

theorem **count_shift** (S : Finset D.what₂.places.Pl) (n : D.what₂.places.Pl → ℤ) (w : D.what₂.places.Pl) : FractionalIdeal.count D.what₂.LL (D.plEq w) (∏ w ∈ S, ((D.plEq w).asIdeal : FractionalIdeal (nonZeroDivisors (𝒪 D.what₂.LL)) D.what₂.LL) ^ (−(n w))) = if w ∈ S then − n w else 0

theorem **vAdd_eq_count** (w : D.what₂.places.Pl) {y : D.what₂.LL} (hy : y ≠ 0) : vAdd (D.plEq w) y = FractionalIdeal.count D.what₂.LL (D.plEq w) (FractionalIdeal.spanSingleton (nonZeroDivisors (𝒪 D.what₂.LL)) y)

theorem **perIntOn_shift_iff** (S : Finset D.what₂.places.Pl) (n : D.what₂.places.Pl → ℤ) (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PerIntOn D u (shift D u S n N) ↔ ∀ w, PerGeOn D u N w (((if w ∈ S then n w else 0 : ℤ)) : ℝ)

theorem **gen_mem** (μ : D.what₂.Tup) {v : D.what₂.Amb μ} (hv : v ∈ (D.what₂.Mtilde μ).gen) : v ∈ (D.what₂.Mtilde μ).carrier

theorem **perIntOnAmb_iff** (A : AmbForm D (RowSide.sharp D)) (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PerIntOnAmb D u A N ↔ PerIntOn D u N

theorem **le_zero_of_functional** {V : Type} [AddCommGroup V] [Module D.what₂.LL V] [Module (𝒪 D.what₂.LL) V] (Λ : Submodule (𝒪 D.what₂.LL) V) (l : V →ₗ[D.what₂.LL] D.what₂.LL) (hl : ∀ x ∈ Λ, ∃ c : 𝒪 D.what₂.LL, l x = (c : D.what₂.LL)) (w : D.what₂.places.Pl) {a : D.what₂.LL} {x : V} (hx : x ∈ locW D.what₂.places w Λ) (h1 : l (a • x) = 1) {s : ℝ} (ha : a = 0 ∨ (D.what₂.places.ramIdx w : ℝ) * s ≤ (vAdd (D.plEq w) a : ℝ)) : s ≤ 0

theorem **col_one** {R : Ranks} {Tau : Type} [Fintype Tau] (hc : Tau → HCParam R) (n : ℕ) (hn : 0 < n) (φ : FK Unit) : (chainN hc (oneCls n hn)).col φ = φ

theorem **fam_per** (u : W) (φ : Unit → H.E) (F : En 1) : (DR).what₂.perCol u (DR).what₂.mu0 ((DR).jN u φ) F = z * H.φ₀ (φ ()) * F 0

theorem **fam_Mt** (F : En 1) : F ∈ ((DR).what₂.Mtilde (DR).what₂.mu0).carrier ↔ ∃ c : 𝒪 H.E, F 0 = H.φ₀ (c : H.E)

theorem **mem_std** (K : Type) [Field K] [NumberField K] [StarRing K] (x : Unit → K) : x ∈ (stdHLat K Unit).L ↔ ∃ c : 𝒪 K, x () = (c : K)

theorem **fam_perGeOn** (u : W) (w : (DR).what₂.places.Pl) (s : ℝ) : PerGeOn DR u (DV.MramL DR u) w s ↔ ValGe DR w z s

theorem **fam_ptInt** (u : W) : PtInt DR u

theorem **fam_periodInt** (u : W) : PeriodInt DR u ↔ ∀ w : (DR).what₂.places.Pl, ValGe DR w z (−(((DR).what₂.places.ramIdx w : ℝ) * vGam DR u w))

theorem **fam_colGeOn** (u : W) (w : (DR).what₂.places.Pl) (s : ℝ) : ColGeOn DR u (DV.MramL DR u) w s ↔ s ≤ 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.AmbIsTrueOfAmbTrue

theorem **map_mem_span_image** {K : Type} [Field K] (σ : K →+* ℂ) {V W : Type} [AddCommGroup V] [Module K V] [AddCommGroup W] [Module ℂ W] (j : V →ₛₗ[σ] W) (S : Set V) {x : V} (hx : x ∈ Submodule.span K S) : j x ∈ Submodule.span ℂ (j '' S)

theorem **span_image_eq_top** {K : Type} [Field K] (σ : K →+* ℂ) {V W : Type} [AddCommGroup V] [Module K V] [AddCommGroup W] [Module ℂ W] (j : V →ₛₗ[σ] W) {ι : Type} (b : Module.Basis ι K V) (S : Set V) (hS : Submodule.span K S = ⊤) (hb : Submodule.span ℂ (Set.range fun i

=> j (b i)) = ⊤) : Submodule.span ℂ (j '' S) = ⊤

theorem **rowsDef_ex** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (hX : B.fields.Ok), Nonempty (RowsDef I l hX)

theorem **ambTrue_ex** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (hX : B.fields.Ok), ∀ (Z : TC.Later B (I.cols l)) (e : ∀ a : (I.cols l).Tup, MtCx I a ≃ₗ[ℂ] Z.W.Amb a), DefTrue I l hX Z.W e → ∃ A : AmbForm (I.rec7 l Z) (RowSide.sharp (I.rec7 l Z)), AmbTrue I l hX Z e A

theorem **exists_choices** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X), StagesInhab.HasLetters B.toPre ∧ I.Ok ∧ TC.Pins B X ∧ I.IsTw ∧ ColIn.JetsIn I

theorem **exists_builtCols** : ∃ D : D7, BuiltCols D

theorem **isRowEmb_rowJ** (h : I.Ok) (u : B.Word) : IsRowEmb I l u (rowJ I l h u)

theorem **isRowEmb_unique** {u : B.Word} {j j'} (h : IsRowEmb I l u j) (h' : IsRowEmb I l u j') : j = j'

theorem **isRowEmb_injective** {u : B.Word} {j} (h : IsRowEmb I l u j) : Function.Injective j

theorem **isPerTrue_unique** {u : B.Word} {j e e'} (h : IsPerTrue I l u j e) (h' : IsPerTrue I l u j e') : e = e'

theorem **exists_isPerTrue** (h : I.Ok) (u : B.Word) : ∃ e, IsPerTrue I l u (rowJ I l h u) e

theorem **ambIsTrue_of_ambTrue** (Z : TC.Later B (I.cols l)) (e) (hD : Df.DefTrue I l (GlobalTheory.Mem.FieldsLaws.fields_ok B) Z.W e) (A : AmbForm (I.rec7 l Z) (RowSide.sharp (I.rec7 l Z))) (hA : Df.AmbTrue I l (GlobalTheory.Mem.FieldsLaws.fields_ok B) Z e A) : AmbIsTrue I l Z A

theorem **exists_ambIsTrue** : ∃ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (A : AmbForm (I.rec7 l Z) (RowSide.sharp (I.rec7 l Z))), I.Ok ∧ TC.Pins B X ∧ I.IsTw ∧ ColIn.JetsIn I ∧ BuiltCols (I.rec7 l Z) ∧ AmbIsTrue I l Z A ∧ ∀ u, ∃ j e, IsRowEmb I l u j ∧ IsPerTrue I l u j e

theorem **ambIsTrue_jAmb** (h : I.Ok) (Z : TC.Later B (I.cols l)) (A : AmbForm (I.rec7 l Z) (RowSide.sharp (I.rec7 l Z))) (hA : AmbIsTrue I l Z A) (w w' : B.Word) (z : (MtTrue I).V) : (I.rec7 l Z).eMt w (rowJ I l h w z) = (I.rec7 l Z).eMt w' (rowJ I l h w' z)

theorem **exists_isRowEmb** (u : B.Word) : ∃ j, IsRowEmb I l u j

Development.AbelianVarietyTheory.Definitions.LatticeFrames.BaseMatrixGlApply

theorem **coordHom_val** (κ : GA D V) : (coordHom V pr β γ βγ γβ κ).val = γ ∗ (mt V κ).map pr ∗ β

theorem **eq_coordHom_iff** (κ : GA D V) (g : GL Y K) : (mt V κ).map pr = β ∗ g.val ∗ γ ↔ g = coordHom V pr β γ βγ γβ κ

theorem **GX.matAt_iff** (κ : GA D (D.sp.X 0)) (g : GL (XIx (Fin D.R.k) (Fin D.R.t) 0) H.k) : H.MatAt κ g ↔ g = H.mat0 κ

theorem **inLev0_iff** (κ : GA D (D.sp.X 0)) : InLev0 o G κ ↔ GlobalTheory.FinPl.Moves (D.sp.X 0) κ (a.x 0).L (a.x 0).L (fun _ => True) ∧ (∀ i, o.mat0 i κ • o.Q 0 i = o.Q 0 i) ∧ ∀ PP, embX ((G PP).s + 1) ((G PP).mat0 κ) ∈ (G PP).lev0.J

theorem **mem_lev0Grp** (κ : GA D (D.sp.X 0)) : κ ∈ levelZeroSubgroup o G ↔ InLev0 o G κ

theorem **InLev0.mul** {κ κ' : GA D (D.sp.X 0)} (h : InLev0 o G κ) (h' : InLev0 o G κ') : InLev0 o G (κ ∗ κ')

theorem **InLev0.inv** {κ : GA D (D.sp.X 0)} (h : InLev0 o G κ) : InLev0 o G κ⁻¹

theorem **InLev0.smul_ker** {κ : GA D (D.sp.X 0)} (h : InLev0 o G κ) (i : ι) : o.mat0 i κ • (o.Q 0 i).ker = (o.Q 0 i).ker

theorem **Arith.of_local** (unifT : ∀ i, valuation k ((o.p i).pr (incA D (((D.mem u).A.T : ℕ) : D.host.E))) = valuation k ((o.q.1 : 𝒪[k]) : k)) (oneG : ∀ PP, valuation (G PP).k ((G PP).q.pr (incA D (((D.mem u).A.T : ℕ) : D.host.E)) - 1) < 1) (unifG : ∀ PP, valuation (G PP).k ((G PP).q.pr (incA D ((D.host.g 0 ∗ D.host.g 1 : ℕ) : D.host.E))) = valuation (G PP).k (((G PP).S.ϖ : (G PP).kˣ) : (G PP).k)) (under : ∀ PP : a.c.Pg, (G PP).q.v = PP.1.under (𝒪 D.host.F)) (unr : ∀ v, v ∉ a.c.S → CharUnramAt D a.c.η v ∧ ∀ j, 1 ≤ j → j ≤ D.R.m → CharUnramAt D (a.c.μ j) v) (lz : a.Lz.SigOK ∧ a.Lz.IsPosDef) : Arith o G

Development.AbelianVarietyTheory.Definitions.LatticeFrames.BuiltGIntro

theorem **builtG_intro** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (h : I.Ok) (hJ : I.JetsIn) (l0 : I.Law0 l) (EX : ErrX (I.rec7 l Z)) (e : EX.piG = X.piG) : BuiltG (I.rec7 l Z) EX

theorem **builtG_ind** {Q : ∀ D : D7, ErrX D → Prop} (H : ∀ (B : GlobalTheory.Mem.Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)), I.Ok → I.JetsIn → I.Law0 l → ∀ EX : ErrX (I.rec7 l Z), EX.piG = X.piG → Q (I.rec7 l Z) EX) {D : D7} {EX : ErrX D} (h : BuiltG D EX) : Q D EX

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ChainParSucc

theorem **chainPar_succ** (c : Com D) (B : GlobalTheory.LFun.Fam D.host.E) (k : ℕ) : chainPar c B (k + 1) = stepPar (c.chS (k + 1)) (c.chT (k + 1)) D.R.t (chainPar c B k)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ChartIndexInjectiveOffLastStep

theorem **cosetIn_eq_empty** {j : ℕ} (h : j + 1 ≠ B.R.m) (p : PtAt B.R I.sel (I.tup a) (I.cst a) u j) : cosetIn p = fun _ => ∅

theorem **chartIndex_injective_offLastStep** (hA : A.Is) {j : ℕ} (hj : j ≤ B.R.m) (h : j + 1 ≠ B.R.m) : Function.Injective (A.b0 j)

theorem **subsingleton_top** (hA : A.Is) : Subsingleton (Ch B.R.m)

theorem **finite_charts** (hA : A.Is) {j : ℕ} (hj : j ≤ B.R.m) : Finite (Ch j)

theorem **kb_top_ne_one** (hI : I.Ok) (hA : A.Is) [NeZero (B.R.N B.R.m - 1)] (P : I.Q a) (hpi : ((X.piT u : 𝒪 B.host.E) : B.host.E) ≠ 1) (β : Ch B.R.m) : A.kb B.R.m β ≠ 1

theorem **bad_finite** (hA : A.Is) (hG : (Formalism.L2.Forms.badG (I.rec7 l Z)).Finite) {j : ℕ} (hj : j ≤ B.R.m) : (A.bad j).Finite

theorem **mem_Q_of_underQ** (hI : I.Ok) {a' : I.ι} {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ I.Q a') (h : I.UnderQ a (P.under (𝒪 B.host.F))) : P ∈ I.Q a

theorem **exists_top_base** (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) : ∃ (k : TC.GA B (B.R.N B.R.m)) (p : PtAt B.R I.sel (I.tup a) (I.cst a) u B.R.m), I.TopPt l Z u a k ∧ TC.PtOf k (extTo I.SQ p.1.1) ∧ p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N B.R.m) ∧ ∀ v : HeightOneSpectrum (𝒪 B.host.F), (B.T u : 𝒪 B.host.F) ∉ v.asIdeal → ¬ I.UnderQ a v → Formalism.UG.bc (Formalism.UG.prV (E

theorem **w_unique** (h : PinColN A ψ θ x vac c₀ kom dd w Ram rd cK) (h' : PinColT A ψ θ x vac c₀ kom' dd' w' Ram' rd' cK) (j : ℕ) (hj : j < B.R.m) (α : Ch j) (γ : Ch (j + 1)) : w j α γ = w' j α γ

theorem **norm_cK** (h : PinColT A ψ θ x vac c₀ kom dd w Ram rd cK) (j : ℕ) (hj : j < B.R.m) (α : Ch j) (γ : Ch (j + 1)) : ‖cK j‖ = |volLev (x j).vol (A.U j α)| * ‖w j α γ‖ * ‖c₀ j‖

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ChartNonempty

theorem **mt_mul** (g h : GA D V) : mt V (g * h) = mt V g * mt V h

theorem **mt_one** : mt V (1 : GA D V) = 1

theorem **finM_mul** (g h : GA D V) : finM V hn (g * h) = finM V hn g * finM V hn h

theorem **finM_one** : finM V hn (1 : GA D V) = 1

theorem **infM_eq** (g : GA D V) (τ : D.what₂.Tau) : infM θ V hn g τ = ((mt V g).map ((θ τ).comp (infPart D))).reindex (finCongr hn) (finCongr hn)

theorem **infM_mul** (g h : GA D V) (τ : D.what₂.Tau) : infM θ V hn (g * h) τ = infM θ V hn g τ * infM θ V hn h τ

theorem **infM_one** (τ : D.what₂.Tau) : infM θ V hn (1 : GA D V) τ = 1

theorem **infM_unit** (g : GA D V) (τ : D.what₂.Tau) : IsUnit (infM θ V hn g τ).det

theorem **toEAf_conj** (a : AE D) : toEAf D (cA D a) = Formalism.UG.sigR (sigD D) (FiniteAdeleRing (𝒪 D.host.F) D.host.F) (toEAf D a)

theorem **toEAf_inc** (e : D.host.E) : toEAf D (incA D e) = inclE (F

theorem **finGL_mem** (g : GA D V) : finGL V hn g ∈ UPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)

theorem **exists_dec** (hy : Function.Surjective (fun c => (Quotient.mk'' (yc c) : Cls S))) (g : GA D V) : ∃ p, IsDec S yc (finM V hn g) p

theorem **rdS_res** (hy : Function.Surjective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC _ (embT D))) (g : GA D V) : rdS θ V hn S yc (fun c => Ψ (yc c)) y g = Ψ (finGL V hn g, finGL_mem V hn g) (trFr θ V hn g y)

theorem **rdS_left** (hθ : IsInfFam θ) (hy : Function.Surjective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC _ (embT D))) (γ : GA D V) (hγ : γ ∈ GQ V) (g : GA D V) : rdS θ V hn S yc (fun c => Ψ (yc c)) y (γ * g) = rdS θ V hn S yc (fun c => Ψ (yc c)) y g

theorem **rdS_right_inf** (ψ : cl → Frames (gramC (V.gram.reindex (finCongr hn) (finCongr hn)) (embT D)) → ℂ) (y : Frames (gramC _ (embT D))) (g k : GA D V) (hk : finM V hn k = 1) : rdS θ V hn S yc ψ y (g * k) = rdS θ V hn S yc ψ (trFr θ V hn k y) g

theorem **rdS_right_lev** (hy : Function.Surjective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC _ (embT D))) (g k : GA D V) (κ : S.U) (hk : finM V hn k = κ.1.1) (hk' : ∀ τ, infM θ V hn k τ = 1) : rdS θ V hn S yc (fun c => Ψ (yc c)) y (g * k) = (S.chi κ : ℂ) * rdS θ V hn S yc (fun c => Ψ (yc c)) y g

theorem **rdS_at_rep** (hy : Function.Surjective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC _ (embT D))) (g : GA D V) (c : cl) (hk : finM V hn g = (yc c).1.1) (hk' : ∀ τ, infM θ V hn g τ = 1) : rdS θ V hn S yc (fun c => Ψ (yc c)) y g = Ψ (yc c) y

theorem **rdS_add** (ψ ψ' : cl → Frames (gramC (V.gram.reindex (finCongr hn) (finCongr hn)) (embT D)) → ℂ) (y) (g : GA D V) : rdS θ V hn S yc (ψ + ψ') y g = rdS θ V hn S yc ψ y g + rdS θ V hn S yc ψ' y g

theorem **chart_rd** {i : ℕ} {hi : i ≤ D.R.m} (Q : Chart D u i hi) (θ : InfFam D) (ψ : (D.mem u).C.cls i → (D.mem u).C.Y i → ℂ) (Ψ) (hΨ : Ψ ∈ AdFun Q.K.lev (embT D)) (hψ : ∀ c z, ψ (Q.inc c) (Q.eY.symm z) = Ψ (Q.K.y c) z) (y : (D.mem u).C.Y i) (g : GA D (D.sp.X i)) : Q.rd θ ψ y g = Ψ (finGL _ (rankN D i hi) g, finGL_mem _ (rankN D i hi) g) (trFr θ _ (rankN D i hi) g (Q.eY y))

theorem **chart_nonempty** {i : ℕ} {hi : i ≤ D.R.m} (Q : Chart D u i hi) : Nonempty Q.cl

theorem **mid_left** (i : ℕ) (h1 : 1 ≤ i) (h2 : i + 1 ≤ D.R.m) (β : Ch i) (θ : InfFam D) (hθ : IsInfFam θ) (ψ : (D.mem u).C.cls i → (D.mem u).C.Y i → ℂ) (hψ : ψ ∈ (D.fm u).Fi i) (y) (γ) (hγ : γ ∈ GQ (D.sp.X i)) (g) : rd F θ i (le_of_succ_m h2) β ψ y (γ * g) = rd F θ i (le_of_succ_m h2) β ψ y g

theorem **end_left** (i : ℕ) (hi : i ≤ D.R.m) (h : i = 0 ∨ i = D.R.m) (β : Ch i) (θ : InfFam D) (hθ : IsInfFam θ) (ψ : (D.mem u).C.cls i → (D.mem u).C.Y i → ℂ) (hψ : ψ ∈ (D.fm u).Fi i) (y) (γ) (hγ : γ ∈ GQ (D.sp.X i)) (g) : rd F θ i hi β ψ y (γ * g) = rd F θ i hi β ψ y g

theorem **rdP_left** (θ : InfFam D) (hθ : IsInfFam θ) (z : (D.mem u).P.Wrow) (hz : z ∈ (D.fm u).ptC) (y) (γ) (hγ : γ ∈ GQ D.sp.X') (g) : rdP F θ z y (γ * g) = rdP F θ z y g

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ColHeightSharpWith

theorem **sizeRowsS_of** {N : NumDatum} {δ : Code N} {T : ℕ} (h : SizeRows N δ T) : SizeRowsS 9 N δ

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ColMatchAll

theorem **CplData.phi_apply** (c : CplData I u a) (h : I.Ok) (hX : B.fields.Ok) (x : (RamH I u hX).V) (z : RowsK I a) : c.phi h hX x z = c.per h hX x.1 z

Development.AbelianVarietyTheory.Definitions.LatticeFrames.CombAtSingleZero

theorem **rank_pos_of_le** (D : D7) {j : ℕ} (h : j ≤ D.R.m) : 0 < (D.sp.X j).rank

theorem **lastLine_rank** (d : Dat D u) : d.Lz.rank = 1

theorem **combAt_single_zero** (P : Pl D → Prop) {S T : HSp D} (c : ℂ) : CombAt P (Finsupp.single (0 : WO S T) c)

theorem **comb_single_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single 0 1) f = f

Development.AbelianVarietyTheory.Definitions.LatticeFrames.CombSmulLeft

theorem **root_ne** {z : ℂ} (h : ∃ n : ℕ, 0 < n ∧ z ^ n = 1) : z ≠ 0

theorem **resc_resc** (w : VTup t) (s s' : ∀ j, t.Ch j → ℂ) : resc (resc w s) s' = resc w (fun j α => s' j α * s j α)

theorem **twist_eq** (w : VTup t) (s : ∀ j, t.Ch j → ℂ) (hs : ∀ j α, s j α ^ 2 = 1) : twist w s = resc w s

theorem **of_qpinS** {v : VCom c} {w : VTup t} (h : QPinS v w) : QPinU v w

theorem **qpinU_resc** {v : VCom c} {w : VTup t} (h : QPinU v w) (s' : ∀ j, t.Ch j → ℂ) (hs' : ∀ j α, ∃ n : ℕ, 0 < n ∧ s' j α ^ n = 1) : QPinU v (resc w s')

theorem **comb_lc** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (e : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T e f = Finsupp.linearCombination ℂ (fun x => heisO ψ S T x f) e

theorem **comb_smul_left** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (a : ℂ) (e : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (a • e) f = a • comb ψ S T e f

theorem **trc_lc** {V : HSp D} (e : GA D V →₀ ℂ) (f : GA D V → ℂ) : trc e f = Finsupp.linearCombination ℂ (fun g => rTrans g f) e

theorem **trc_smul_left** {V : HSp D} (a : ℂ) (e : GA D V →₀ ℂ) (f : GA D V → ℂ) : trc (a • e) f = a • trc e f

theorem **biComb_smul2** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (a b : ℂ) (e : WO A S →₀ ℂ) (e' : WO S A' →₀ ℂ) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' (a • e) (b • e') C = (a * b) • biComb ψ A S A' e e' C

theorem **biLift_move** {A S A' : HSp D} (oma : Om A S) (omb : Om S A') (vA : Vol A) (vS : Vol S) (φ : GA D A → ℂ) (a : ℂ) (C : WO A S → WO S A' → ℂ) : biLift oma omb vA vS (a • φ) C = a • biLift oma omb vA vS φ C

theorem **biLift_smul'** {A S A' : HSp D} (oma : Om A S) (omb : Om S A') (vA : Vol A) (vS : Vol S) (φ : GA D A → ℂ) (a : ℂ) (C : WO A S → WO S A' → ℂ) : biLift oma omb vA vS φ (a • C) = a • biLift oma omb vA vS φ C

theorem **hand_smul_fun** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (a : ℂ) (Φ₁ : WO B.V J.W → ℂ) (Φ₂ : WO J.W B'.V → ℂ) (φ : GA D B.A → ℂ) (g : GA D B.V) (h₁ : GA D J.W) (h₂ : GA D B'.V) : Chain.hand s₁ s₂ l vL Φ₁ Φ₂ (a • φ) g h₁ h₂ = a • Chain.hand s₁ s₂ l vL Φ₁ Φ₂ φ g h₁ h₂

theorem **sum_supp** {G : Type} (a : ℂ) (e : G →₀ ℂ) (F : G → ℂ) : ∑ g ∈ (a • e).support, (a • e) g * F g = ∑ g ∈ e.support, e g * (a * F g)

theorem **contAut_smul** {V : HSp D} (a : ℂ) {φ : GA D V → ℂ} (h : IsContAut V φ) : IsContAut V (a • φ)

theorem **qpin_free** {v : VCom c} {w : VTup t} (h : QPin v w) : QPin v (free w)

theorem **inner** (s : ∀ j, t.Ch j → ℂ) (hs : ∀ j α, s j α ≠ 0) (w : VTup t) (j : ℕ) (β : t.Ch (j + 1 + 1)) (φ : t.Ch j → GA D (c.X j) → ℂ) (C) : (∑ α, ∑ γ, c.bl j (t.kom j) (t.kom (j + 1)) (φ α) (biComb c.ψ _ _ _ ((resc w s).dQ j α γ) ((resc w s).dQ (j + 1) γ β) C)) = (s (j + 1 + 1) β)⁻¹ • ∑ α, ∑ γ, c.bl j (t.kom j) (t.kom (j + 1)) (s j α • φ α) (biComb c.ψ _ _ _ (w.dQ j α γ) (w.dQ (j + 1) γ β) C)

theorem **qlast_resc** (v : VCom c) (w : VTup t) (s : ∀ j, t.Ch j → ℂ) (hs : ∀ j α, s j α ≠ 0) (H : QLast v (free w)) : QLast v (resc w s)

theorem **qlaw_resc** (v : VCom c) (w : VTup t) (s : ∀ j, t.Ch j → ℂ) (hs : ∀ j α, s j α ≠ 0) (H : QLaw v (free w)) : QLaw v (resc w s)

theorem **qfin_resc** (v : VCom c) (w : VTup t) (s : ∀ j, t.Ch j → ℂ) (H : QFin v w) : QFin v (resc w s)

theorem **laws_of_qpinU** (v : VCom c) (H : ∀ w : VTup t, QPin v w → QLaw v w ∧ QLast v w ∧ QFin v w) (w : VTup t) (h : QPinU v w) : QLaw v w ∧ QLast v w ∧ QFin v w

theorem **qpinU_iff** {v : VCom c} {w : VTup t} : QPinU v w ↔ ∃ s : ∀ j, t.Ch j → ℂ, (∀ j α, ∃ n : ℕ, 0 < n ∧ s j α ^ n = 1) ∧ QPin v (resc w s)

theorem **norm_dQ** (w : VTup t) (s : ∀ j, t.Ch j → ℂ) (hs : ∀ j α, ∃ n : ℕ, 0 < n ∧ s j α ^ n = 1) (j α γ x) : ‖(resc w s).dQ j α γ x‖ = ‖w.dQ j α γ x‖

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ConclusionZero

theorem **of_all** (D : D7) (h : SiegelWeilAll D) : SiegelWeilFock D

theorem **resEta_one_dbl** (A : HSp D) {χ : (AE D)ˣ →* Circle} {k : ℕ} (hr : ResEta D χ k) : ResEta D 1 (dblSp A).rank

theorem **hyp** (A B : HSp D) (hA : HSpace.SigOK A) (hB : HSpace.SigOK B) (ha : 1 ≤ A.rank) (hab : A.rank < B.rank) : Hyp A B

theorem **hyp_definite** (D : D7) {n m : ℕ} (hn : 1 ≤ n) (hm : n < m) : ∃ A B : HSp D, A.rank = n ∧ B.rank = m ∧ wittIndex B = 0 ∧ Hyp A B

theorem **hyp_indefinite** (D : D7) : ∃ A B : HSp D, 1 ≤ A.rank ∧ A.rank < B.rank ∧ B.rank ≤ 2 * A.rank ∧ A.rank + wittIndex B < B.rank ∧ (∀ τ, B.sig τ = (8, 1)) ∧ HSpace.SigOK B ∧ Hyp A B

theorem **conclusion_zero** (A B : HSp D) (hab : A.rank < B.rank) (omh : WRep (dblSp A) B) (a : GL (Fin A.rank) (AE D)) (ν : Measure (WO A B)) (𝓕 : Set (WO A B)) (mH : Measure (GA D B)) : Eis.EisValueAt A (flat A a (swSec A B omh ν 𝓕 0) (sZero A B)) (sZero A B) (ThetaMean.thetaMean omh mH 0)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ContinuesConst

theorem **Dominated.refl** {G : Type} [Group G] (K : Subgroup G) (φ : G → ℂ) : Dominated K φ φ

theorem **exists_isLevel** (T : HSp D) : ∃ a, IsLevel T (AdelicU.frameLat T) a

theorem **lev_frameLat** (T : HSp D) (a : GL (Fin T.rank) (AE D)) : lev T (AdelicU.frameLat T) a = Kmax T a

theorem **Dominated.const_mul** {G : Type} [Group G] (K : Subgroup G) (φ : G → ℂ) (a : ℂ) : Dominated K φ fun g => a * φ g

theorem **Dominated.eq_zero** {G : Type} [Group G] {K : Subgroup G} {ψ : G → ℂ} (h : Dominated K (fun _ => 0) ψ) (g : G) : ψ g = 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.CornerCoordinateAutomatic

theorem **used_base** (P : RowSideFamily ex) : Formalism.IndefR.RecTie.Used0 D (P.I.tup P.A3.A.μ0)

theorem **mu_eq_of_heq** (P : RowSideFamily ex) {Zs : DefSub P.A3.fam3.amb3} (h : Zs = P.Zs) (S : IndefData (P.A3.side3 P.P0 P.ψ0 Zs)) (hS : HEq S P.S0) : S.μ = P.S0.μ

theorem **used_all** (W : RealPack ex) (t : W.P.T) : Formalism.IndefR.RecTie.Used0 D (W.P.I.tup t)

theorem **dcPtG_iff** (W : RealPack ex) : DcPtG W.P ↔ RowsHoldDcPt3 W.P

theorem **hypAt_iff** (W : RealPack ex) : HypAt W ↔ Nonempty (Formalism.L2.SG.JoinSI.JoinSI W.R W.S W.Q) ∧ RowsHoldDcPt3 W.P ∧ W.Q.Rl.Char3 W.Q.F ∧ FramesOKg2 W.Q.Rl

theorem **hypAt_of** (W : RealPack ex) (hJ : Nonempty (Formalism.L2.SG.JoinSI.JoinSI W.R W.S W.Q)) (hP : RowsHoldDcPt3 W.P) (hE : W.Q.Rl.toRealDat.ExAt) (hF : W.Q.FramesEx) : HypAt W

theorem **period_rational** {P : RowSideFamily ex} (h : DcPtG P) (u : D.what₂.Word) (x : Fin (P.Dc.a (P.I.wU u)) → D.what₂.LL) (F : Fin ((P.A3.side3 P.Ps P.ψs P.Zs).nV P.A3.A.μ0) → D.what₂.LL) : D.what₂.perCol u (P.I.tup P.A3.A.μ0) (D.jN u (P.I.eNu u x)) (P.I.jA P.A3.A.μ0 ((rV0 D P.A3.A P.Ps P.ψs P.A3.A.μ0).mulVec F)) ∈ Set.range D.what₂.embLL

theorem **cornerCoordinate_automatic** (W : RealPack ex) : (cxW W).CoordinatesInGoodPart W.Q.F.cr W.Q.F.cv true

theorem **tiesRest_of_cxTies** (W : RealPack ex) (h : CxTies (cxW W) W.Q.F.cr W.Q.F.cv (W.Q.F.eV (cxW W).zs)) : TiesRest W

Development.AbelianVarietyTheory.Definitions.LatticeFrames.CxTiesOf

theorem **cxTies_of** {W : RealPack ex} (hH : HypAt W) (hLv : LevHyp W) (hR : TiesRest W) : CxTies (cxW W) W.Q.F.cr W.Q.F.cv (W.Q.F.eV (cxW W).zs)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.DblVacAtInhabited

theorem **exists_isSkewOne** (D : D7) (n : ℕ) : ∃ κ 𝒞, IsSkewOne (D

theorem **exists_split** (D : D7) : ∃ q : FinPl D, Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **exists_dbl_vacAt** (A B : HSp D) (hA : A.SigOK) (hB : B.SigOK) (h1 : 1 ≤ A.rank) (h2 : A.rank + 2 ≤ B.rank) (hw : wittIndex B = 0 ∨ A.rank + wittIndex B < B.rank) : ∃ ψ χB omh a ν 𝒴 mH κ 𝒞 f, Dbl ψ A B χB omh a ν 𝒴 mH κ 𝒞 f ∧ swSec A B omh ν 𝒴 f 1 = 1 ∧ sch ψ A B ν 𝒴 f 0 = 1 ∧ ∀ q, Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **dbl_vacAt_inhabited** (D : D7) {n t : ℕ} (hn : 1 ≤ n) (ht : 2 ≤ t) : ∃ (A B : HSp D) (ψ : _) (χB : _) (omh : _) (a : _) (ν : _) (𝒴 : _) (mH : _) (κ : _) (𝒞 : _) (f : _) (q : _), A.rank = n ∧ B.rank = n + t ∧ Dbl ψ A B χB omh a ν 𝒴 mH κ 𝒞 f ∧ swSec A B omh ν 𝒴 f 1 = 1 ∧ VacAt ψ A B χB omh (frameLat B) q f

theorem **exists_isWeyl** (A : HSp D) : ∃ w, IsWeyl A w ∧ lowC A w = 1

theorem **lowC_one** (A : HSp D) : lowC A 1 = 0

theorem **lowC_mul_nC** (A : HSp D) (g : GA D (dblSp A)) (c : skewA D A.rank) : lowC A (g ∗ (dblDec A).nC c) = lowC A g

theorem **chr_gramOf** (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (x : Mx D B.rank A.rank) (c : skewA D A.rank) : chr ψ (gramOf B x) c = ψ ((dblDec A).qS B c x)

theorem **laws_of_formulas** {ψ : AddChar (AF D.host.F) ℂ} {A B : HSp D} {χB : (AE D)ˣ →∗ Circle} {omh : WRep (dblSp A) B} {om₀ : WRep (zeroSp D) B} {ν : Measure (WO A B)} {𝒴 : Set (WO A B)} {f : WO (dblSp A) B → ℂ} (h : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.FormulasS (dblDec A) ψ χB omh om₀ ν 𝒴 f) : (∀ (m : GL (Fin A.rank) (AE D)) x, sch ψ A B ν 𝒴 (omh ((dblDec A).levi m, 1) f) x = ((χB⁻¹ (Matrix.GeneralLinearGroup.det m) : Circle) : ℂ) ∗ npow (absA D (Matrix.GeneralLinearGroup.det m)) ((B.rank : ℂ) / 2) ∗ sch ψ A B ν 𝒴 f (x · m.1)) ∧ ∀ c x, sch ψ A B ν 𝒴 (omh ((dblDec A).nC c, 1) f) x = chr ψ (gramOf B x) c ∗ sch ψ A B ν 𝒴 f x

theorem **chr_add** (ψ : AddChar (AF D.host.F) ℂ) {n : ℕ} (T : Mx D n n) (c c' : skewA D n) : chr ψ T (c + c') = chr ψ T c ∗ chr ψ T c'

theorem **nsAt_of_norm** {par : LFun.Fam D.host.E} (χ : (AE D)ˣ →∗ Circle) {t : ℕ} {q : FinPl D} (h : ∀ w : HeightOneSpectrum (𝒪 D.host.E), w.under (𝒪 D.host.F) = q → ∀ β ∈ par w, ∃ j : ℤ, |j| < (t : ℤ) - 1 ∧ ‖(β : ℂ)‖ = ((Ideal.absNorm w.asIdeal : ℕ) : ℝ) ^ ((j : ℝ) / 2)) : NSAt par χ t q

theorem **not_nsAt** (χ : (AE D)ˣ →∗ Circle) (t : ℕ) (w : HeightOneSpectrum (𝒪 D.host.E)) : ∃ par : LFun.Fam D.host.E, ¬ NSAt par χ t (w.under (𝒪 D.host.F))

theorem **nsAt_one** (A : HSp D) (χ : (AE D)ˣ →∗ Circle) {t : ℕ} (ht : A.rank < t) {q : FinPl D} (hq : Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Definitions.LatticeFrames.EvalAtOneExtremalMul

theorem **evalAt_lft** (g y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : evalAt y (lft g F) = evalAt (fun τ => g τ ∗ y τ) F

private lemma **mvZ_one_right_X_eq_sum** {κ : Type} (h : κ → Matrix (Fin b) (Fin b) ℂ) (τ : κ) (β : Fin b) (p : Fin a) : mvZ (fun _ => 1) h (X (τ, β, p) : FP κ a b) = ∑ i, C (h τ i β) ∗ X (τ, i, p)

theorem **mvZ_subY** (hab : a ≤ b) (h y' : ι → Matrix (Fin b) (Fin b) ℂ) (F : PY ι a) : mvZ (fun _ => 1) h (subY hab y' F) = subY hab (fun τ => h τ ∗ y' τ) F

theorem **subY_eq_subD** (hab : a ≤ b) (y' : ι → Matrix (Fin b) (Fin b) ℂ) (F : PY ι a) : subY hab y' F = aeval (Sum.elim (fun x : ι × Fin b × Fin b => (C (y' x.1 x.2.1 x.2.2) : FP ι a b)) X) (subD ℂ ι a b F)

theorem **jetL_monomial** (c : ι → ℂ) (G : VCx ι a b → ℂ) (m : ι × Fin b × Fin a →₀ ℕ) : jetL c G (monomial m 1) = jetC c G m

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsBuiltAndNot

theorem **under_inj** (u : B.Word) (P Q : (B.primes X).vT u) (e : P.1.under (𝒪 B.host.F) = Q.1.under (𝒪 B.host.F)) : P = Q

theorem **builtG_errOf** (I : TC B X) (h : I.Ok) (hJ : I.JetsIn) (l : I.Laws) (Z : TC.Later B (I.cols l)) : BuiltG (I.rec7 l Z) (errOf I l Z)

theorem **exists_builtG** : ∃ (D : D7) (EX : ErrX D), BuiltG D EX

theorem **cong** {D : D7} {EX : ErrX D} (h : BuiltG D EX) (i : Fin 2) : ((D.host.QS ∗ D.host.q ∗ D.host.d : ℕ) : 𝒪 D.host.E) ∣ EX.piG i - 1

theorem **nonempty_cls** {D : D7} {EX : ErrX D} (h : BuiltG D EX) (u : D.what₂.Word) : Nonempty ((D.mem u).C.cls 0)

theorem **congr_piG** {D : D7} {EX EX' : ErrX D} (h : BuiltG D EX) (e : EX'.piG = EX.piG) : BuiltG D EX'

theorem **not_builtG_neg** {D : D7} {EX : ErrX D} (h : BuiltG D EX) : ¬ BuiltG D (negG EX)

theorem **exists_built_and_not** : ∃ (D : D7) (EX EX' : ErrX D), BuiltG D EX ∧ ¬ BuiltG D EX'

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsDatOfTower

theorem **reOm_apply** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →∗ Circle} {om : Om S T} (h : IsWeilDatumO ψ S T χS χT om) (π : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)) (h₁ h₂) (p f) : reOm h π h₁ h₂ p f = om p (π f) + (f - π f)

theorem **reOm_theta** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →∗ Circle} {om : Om S T} (h : IsWeilDatumO ψ S T χS χT om) (π : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)) (h₁ h₂) (p f) (hf : IsThetaFunO ψ S T f) : reOm h π h₁ h₂ p f = om p f

theorem **reOm_weil** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →∗ Circle} {om : Om S T} (h : IsWeilDatumO ψ S T χS χT om) (π : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)) (h₁ h₂) : IsWeilDatumO ψ S T χS χT (reOm h π h₁ h₂) where

theorem **kerDat_free** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {x : DefSp S} {y : DefSp T} (k : KerDat S T) (hk : k.Is ψ θ x y) (N : Submodule ℂ (WO S T → ℂ)) (hN : ∀ g ∈ N, IsThetaFunO ψ S T g → g = 0) (τ : N →ₗ[ℂ] (WO S T → ℂ)) (hτ : ∀ g, IsThetaFunO ψ S T (τ g)) : ∃ k' : KerDat S T, k'.Is ψ θ x y ∧ k'.vac = k.vac ∧ k'.c = k.c ∧ (∀ p f, IsThetaFunO ψ S T f → k'.om p f = k.om p f) ∧ ∀ p (g : N), k'.om p g = k.om p (τ g) + (g - τ g)

theorem **exists_congr** (S V : HSp D) (hS : S.SigOK) (hV : V.SigOK) (hr : S.rank = V.rank) (hs : ∀ τ, S.sig τ = V.sig τ) : ∃ M : Matrix (Fin V.rank) (Fin S.rank) D.host.E, Mᴴ ∗ V.gram ∗ M = S.gram

theorem **block_frame_conjugation_identity** {n k a b : Type} [Fintype n] [Fintype k] {E : Type} [CommRing E] [StarRing E] (M : Matrix n k E) (G : Matrix n n E) (x : Matrix k a E) (y : Matrix k b E) : (M ∗ x)ᴴ ∗ G ∗ (M ∗ y) = xᴴ ∗ (Mᴴ ∗ G ∗ M) ∗ y

theorem **nonempty_orthDec** (V A B : HSp D) (hV : V.SigOK) (hA : A.SigOK) (hB : B.SigOK) (hr : A.rank + B.rank = V.rank) (hs : ∀ τ, A.sig τ + B.sig τ = V.sig τ) : Nonempty (OrthDec V A B)

theorem **nonempty_hypDec** (V A : HSp D) (ρ : ℕ) (hV : V.SigOK) (hA : A.SigOK) (hr : A.rank + 2 ∗ ρ = V.rank) (hs : ∀ τ, A.sig τ + (ρ, ρ) = V.sig τ) : Nonempty (HypDec V A ρ)

theorem **sigOK_negLine** : (GlobalTheory.Doubling.hneg (Examples₂.posSpace (E

theorem **isLineVec_nLine** {c : D.host.E} (hc : c ≠ 0) : IsLineVec (nLine D) (fun _ => c)

theorem **rank_arith** (R : Ranks) {j : ℕ} (h : j < R.m) : R.rankY j = R.N j + (R.m − 1 − j)

theorem **exists_besAt** {c : Com D} {s : PiLocalization₂ D} {j ρ : ℕ} (h : j + 2 ∗ ρ + 2 = D.R.m) (hs : s.Is c j) : ∃ (d : HypDec s.V (GlobalTheory.Doubling.hsum (D.sp.X j) (nLine D)) ρ), Nonempty (VolN d)

theorem **nonempty_jacAt** {c : Com D} {s : PiLocalization₂ D} {j ρ : ℕ} (h : j + 2 ∗ ρ + 2 = D.R.m) (hs : s.Is c (j + 1)) : Nonempty (JacAt s.V (D.sp.X (j + 1)) ρ)

theorem **nonempty_top** (D : D7) (n : ℕ) (h : n = D.R.m) : Nonempty (OrthDec (D.sp.X n) D.sp.X' (Examples₂.posSpace (E

theorem **exists_step** {ψ : AddChar (AF D.host.F) ℂ} {V V₀ W W₀ : HSp D} {a b : ℕ} (dV : HypDec V V₀ a) (dW : HypDec W W₀ b) (ρ σ : ℕ) {χV χW : (AE D)ˣ →∗ Circle} {om : Om V W} (h : IsWeilDatumO ψ V W χV χW om) (vW : Vol W) : ∃ s : GlobalTheory.Unf.Step ψ V V₀ W W₀ ρ σ, s.om = om ∧ s.vW.μ = vW.μ ∧ s.vW.𝓕 = vW.𝓕

theorem **exists_dat** (g : Geo D) (u : D.what₂.Word) (p : ∀ j, GlobalTheory.ChainOnto.Step g.c (g.st j) (g.st (j + 1))) (pT : ∀ j, j + 2 ∗ 0 + 2 = D.R.m → GlobalTheory.ChainOnto.Step g.c (g.st (j + 1)) g.top.st) (hc : g.c.Is) (hst : ∀ j, j < D.R.m → (g.st j).Is g.c j) (htop : g.top.st.Is g.c D.R.m) (hp : ∀ j, j + 1 < D.R.m → (p j).Is (j + 1)) (hpT : ∀ j h, (pT j h).Is D.R.m) (t : ∀ j, GA D (g.st j).V) (ht : ∀ j, t j ∈ GfC (g.st j).V) (Ram : ∀ j, Submodule ℂ (Col D u j)) (S : ∀ j, Col D u j →ₗ[ℂ] Col D u j) (rd : ∀ j, Col D u j → (D.mem u).C.Y j → GA D (D.sp.X j) → ℂ) (eT : (D.mem u).P.Mt → g.top.st.CC g.c) : ∃ d : Dat D u, d.Base ∧ d.toGeo = g ∧ d.Ram = Ram ∧ d.S = S ∧ d.rd = rd ∧ HEq d.t t ∧ HEq d.eT eT ∧ ∀ j, j < D.R.m → (d.k j).vac ≠ 0 ∧ (d.k j).c ≠ 0

theorem **exists_stAt** (s : PiLocalization₂ D) (V : HSp D) (h : s.V = V) : ∃ a : StAt V, a.st = s

theorem **step_cast** {c : Com D} {s₁ s₂ s₁' s₂' : PiLocalization₂ D} (h₁ : s₁ = s₁') (h₂ : s₂ = s₂') (p : GlobalTheory.ChainOnto.Step c s₁ s₂) : ∃ p' : GlobalTheory.ChainOnto.Step c s₁' s₂', ∀ j, p.Is j → p'.Is j

theorem **exists_dat_of_tower** (W : GlobalTheory.Core.Rk.Tower D) (hW : W.Is) (u : D.what₂.Word) (Ram : ∀ j, Submodule ℂ (Col D u j)) (S : ∀ j, Col D u j →ₗ[ℂ] Col D u j) (rd : ∀ j, Col D u j → (D.mem u).C.Y j → GA D (D.sp.X j) → ℂ) : ∃ d : Dat D u, d.Base ∧ d.c = comOf W ∧ (∀ j, d.st j = stageY W j) ∧ d.top.st = stageX W ∧ d.Ram = Ram ∧ d.S = S ∧ d.rd = rd ∧ ∀ j, j < D.R.m → (d.k j).vac ≠ 0 ∧ (d.k j).c ≠ 0

theorem **exists_dat_laws** (W : GlobalTheory.Core.Rk.Tower D) (hW : W.Is) (u : D.what₂.Word) : ∃ d : Dat D u, d.Base ∧ d.RdLaw ∧ d.Stab ∧ d.Comm ∧ ∀ j ψ y, d.rd j ψ y ≠ 0

theorem **exists_dat_not_comm** (W : GlobalTheory.Core.Rk.Tower D) (hW : W.Is) (u : D.what₂.Word) (h : (D.mem u).A.Mram ≠ ⊥) : ∃ d : Dat D u, d.Base ∧ ¬ d.Comm

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsDblClosed

theorem **nonempty_frame** (k : Type) [Field k] {N : ℕ} (Y : Type) [Fintype Y] [DecidableEq Y] (h : Fintype.card Y = N) : Nonempty (GlobalTheory.LocMod.Frame k N Y)

theorem **exists_dbl** (p : Place D k) {σ : ℕ} (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] {s : k} (hs : s ≠ 0) (F₀ : GlobalTheory.LocMod.Frame k B.A.rank n₀) (F₁ : GlobalTheory.LocMod.Frame k J.W₀.rank n₁) (F₂ : GlobalTheory.LocMod.Frame k B'.A.rank n₂) : ∃ F : Dbl p B J B' n₀ n₁ n₂ (2 ∗ σ), F.s = s ∧ F.F₀ = F₀ ∧ F.F₁ = F₁ ∧ F.F₂ = F₂

theorem **Dbl_c** (p : Place D k) {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} {n₀ n₁ n₂ : Type} [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] {c : ℕ} (F : Dbl p B J B' n₀ n₁ n₂ c) : c = 2 ∗ σ

theorem **exists_dbl_closed** (D : D7) : ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (B : BesLev D 1) (J : JacLev D 1) (B' : BesLev D 0), Nonempty (StepBF ψ B J) ∧ Nonempty (StepFB (σ

theorem **exists_isOne** : ∃ Φ : SB (Matrix Y Y' k), IsOne Φ ∧ Φ.1 0 = 1

theorem **isCombAt_one** (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) : p.IsCombAt ψ FS FT (Finsupp.single 0 1) (Set.indicator {x | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1)

theorem **toEAf_surj** (D : D7) : Function.Surjective (toEAf D)

theorem **pi_congr** (p : Place D k) {a b : Formalism.L2.Theta.AE D} (h : toEAf D a = toEAf D b) : p.π a = p.π b

theorem **exists_movedTo** (p : Place D k) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) {M M' : Matrix (Fin V.rank) (Fin V.rank) k} (hM : M ∗ M' = 1) : ∃ g, SuppAt p.e V g ∧ MovedTo p V g β M

theorem **movedTo_unique** (p : Place D k) (V : HSp D) (g : GA D V) (β : GL (Fin V.rank) (Lev.EAf D)) {M M' : Matrix (Fin V.rank) (Fin V.rank) k} (h : MovedTo p V g β M) (h' : MovedTo p V g β M') : M = M'

theorem **exists_ord** (κ l τ j N : ℕ) (h : Fintype.card (Ix (Fin κ ⊕ Fin l) (Fin τ) j) = N) : ∃ e : Ix (Fin κ ⊕ Fin l) (Fin τ) j ≃ Fin N, GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.zPos e

theorem **exists_gAt** (d : Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (unitVol : IsUnitVol ν) (q : GlobalTheory.FinPl.SplitPt D k) (idem : Formalism.L2.NTheta.IsIdemAt q.e PP) (unit : q.UnitAA) (s : ℕ) (hs : s + 1 + 1 = D.R.m) (S : SymData k s D.R.k) (char : (ofSplit q).IsChar d.c.ψ S.ψ) (mu : ∀ j, 1 ≤ j → j ≤ D.R.m → (ofSplit q).IsComp (d.c.chS j)⁻¹ (S.μ j)) (FX : ∀ j, j ≤ s + 1 → GlobalTheory.LocMod.Frame k (D.sp.X j).rank (XIx (Fin D.R.k) (Fin D.R.t) j)) (FT : GlobalTheory.LocMod.Frame k D.sp.X'.rank (Ix (Fin D.R.k ⊕ Fin (s + 1)) (Fin D.R.t) (s + 2))) (FM : GlobalTheory.LocMod.Frame k (D.sp.X (s + 1 + 1)).rank (Ix (Fin D.R.k ⊕ Fin (s + 1)) (Fin D.R.t) (s + 2) ⊕ Unit)) (c : k) (hc : valuation k c = 1) (hxi : valuation k (ofSplit q).xi = 1) (hz_unit : ∀ j ρ h, valuat

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsDoubleDescentRanksOneThreeFive

theorem **isCO_sph** : IsCO (sph k n).J

theorem **om_ind** {μ : kˣ →∗ ℂˣ} (hμ : IsUnram μ) {g : GL n' k} {h : GL n k} (hg : g ∈ KO k n') (hh : h ∈ KO k n) : om μ g h (ind k n n') = ind k n n'

theorem **hasTypes_ind** {μ : kˣ →* ℂˣ} (hμ : IsUnram μ) : HasTypes μ (sph k n') (sph k n) (ind k n n')

theorem **covers_ind** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (μ : kˣ →* ℂˣ) : Covers 1 (KO k (n ⊕ Fin (c + 1))) {1} fun h => GlobalTheory.StepData.flat ψ ν μ (ind k (n ⊕ Fin (c + 1)) (n' ⊕ Fin (c + 1 + 1))) 1 h

theorem **xi_ind** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (μ : kˣ →* ℂˣ) : xi ψ ν μ (ind k (n ⊕ Fin (c + 1)) (n' ⊕ Fin (c + 1 + 1))) (KO k _) 1 1 1 = ind k n n'

theorem **isOneComb_one** (ψ : AddChar k Circle) (ν : Measure k) (L : Level (GL (n ⊕ Fin c) k)) (β : GL (n ⊕ Fin c) k) : IsOneComb ψ ν 1 L β (Finsupp.single β 1)

theorem **isOneComb_two** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (hc : 1 < c) : IsOneComb ψ ν 2 (sph k (n ⊕ Fin c)) 1 (Finsupp.single 1 1)

theorem **exists_haar** (ν : Measure k) [ν.IsAddHaarMeasure] (n : Type) [Fintype n] [DecidableEq n] : ∃ m : Measure (GL n k), m.IsHaarMeasure ∧ m (KO k n) = 1

theorem **isOpen_ker** {μ : kˣ →* ℂˣ} (hμ : IsUnram μ) : IsOpen (μ.ker : Set kˣ)

theorem **exists_doubleDescent_of_parity** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {σ : ℕ} (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (h₁ : B.A.rank % 2 = J.W₀.rank % 2) (h₂ : J.W₀.rank % 2 = B'.A.rank % 2) : ∃ S : Dbl2 ψ σ, S.B = B ∧ S.J = J ∧ S.B' = B'

theorem **even_im_hermitian_sub** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} (S : Dbl2 ψ σ) : S.B.A.rank % 2 = S.J.W₀.rank % 2 ∧ S.J.W₀.rank % 2 = S.B'.A.rank % 2

theorem **exists_doubleDescent_ranks_one_three_five** (D : D7) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (S : Dbl2 ψ 0), IsStdChar D ψ (-1) ∧ S.B.A.rank = 1 ∧ S.J.W₀.rank = 3 ∧ S.B'.A.rank = 5

theorem **exists_gl** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (V : HSp D) {Y : Type} [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k V.rank Y) : ∃ gl : GL Y k →* GA D V, ∀ g, (ofSplit q).IsLoc F (gl g) g

theorem **exists_qstep_at** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S : Dbl2 ψ 0) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {ψk : AddChar k Circle} (hk : (ofSplit q).IsChar ψ ψk) (hc : HasConductor ψk 1) {μ₁ μ₂ : kˣ →* ℂˣ} (c₁ : (ofSplit q).IsComp S.χ₁⁻¹ μ₁) (c₂ : (ofSplit q).IsComp S.χ₂⁻¹ μ₂) (u₁ : IsUnram μ₁) (u₂ : IsUnram μ₂) : ∃ Q : QStep S, Q.q.ν = q.ν ∧ IsSph Q

theorem **infinite_qstep** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S : Dbl2 ψ 0) : {ν | ∃ Q : QStep S, Q.q.ν = ν ∧ IsSph Q}.Infinite

theorem **exists_family** (D : D7) (r : ℕ) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (S : Dbl2 ψ 0) (Q : Fin r → QStep S), IsStdChar D ψ (-1) ∧ S.B.A.rank = 1 ∧ S.J.W₀.rank = 3 ∧ S.B'.A.rank = 5 ∧ Function.Injective (fun i => (Q i).q.ν) ∧ ∀ i, IsSph (Q i)

theorem **volJ_ne_zero** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {S : Dbl2 ψ σ} (Q : QStep S) : Q.volJ ≠ 0

theorem **w_ne_zero** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {S : Dbl2 ψ σ} (Q : QStep S) (γ : Q.A₁) (β : Q.A₂) : Q.w₁ γ ≠ 0 ∧ Q.w₂ β ≠ 0

theorem **c_eq** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {S : Dbl2 ψ σ} (Q : QStep S) : Q.c = 2 * σ

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsFoldr

theorem **mat_iff** (G : GLoc v) (j a : ℕ) (ha : j + a = G.s + 1) (g : GA D (c.V j)) (x) : G.Mat j a ha g x ↔ x = (matM G j a ha).toHomUnits g

theorem **glam_unique** {n : ℕ} (G : Fin n → GLoc v) (j : ℕ) (g : GA D (c.V j)) {a a' : ℂ} (h : GLam G j g a) (h' : GLam G j g a') : a = a'

theorem **glam_exists** {n : ℕ} {G : Fin n → GLoc v} (P : GPin G) (j : ℕ) (hj : j < D.R.m) (g : GA D (c.V j)) (hg : g ∈ v.JG j) : ∃ a, a ≠ 0 ∧ GLam G j g a

theorem **glam_mul** {n : ℕ} (G : Fin n → GLoc v) (j : ℕ) {g g' : GA D (c.V j)} {a a' : ℂ} (h : GLam G j g a) (h' : GLam G j g' a') : GLam G j (g * g') (a * a')

theorem **exists_foldr** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) {ι : Type} (e : ι → WO S T →₀ ℂ) (l : List ι) : ∃ d : WO S T →₀ ℂ, ∀ f, comb ψ S T d f = l.foldr (fun i f' => comb ψ S T (e i) f') f

theorem **exists_pin** {n : ℕ} (G : Fin n → GLoc v) (inj : Function.Injective fun i => (G i).q.ν) : ∃ gP JG eG eGT dG, GPin fun i => move (G i) gP JG eG eGT dG

theorem **exists_gLoc** (v : VCom c) [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (unitVol : IsUnitVol ν) (q : GlobalTheory.FinPl.SplitPt D k) (unit : q.UnitAA) (s : ℕ) (hs : s + 1 + 1 = D.R.m) (S : SymData k s D.R.k) (char : (ofSplit q).IsChar c.ψ S.ψ) (FX : ∀ j, j ≤ s + 1 → GlobalTheory.LocMod.Frame k (c.X j).rank (XIx (Fin D.R.k) (Fin D.R.t) j)) (FT : GlobalTheory.LocMod.Frame k c.T.rank (Ix (Fin D.R.k ⊕ Fin (s + 1)) (Fin D.R.t) (s + 2))) (FM : GlobalTheory.LocMod.Frame k (c.X (s + 1 + 1)).rank (Ix (Fin D.R.k ⊕ Fin (s + 1)) (Fin D.R.t) (s + 2) ⊕ Unit)) (a₀ : k) (hc : valuation k a₀ = 1) (hxi : valuation k (ofSplit q).xi = 1) (hz_unit : ∀ j ρ h, valuation k ((ofSplit q).hz (c.B j ρ h)) = 1) (z₀ : Fin c.Lz.rank → k) (last : FM.β = fromCols ((c.oT (s + 1 + 1) hs).iA.map (ofSplit q).pr * FT.β) (Matrix.of fun r _ => ((c.oT (s + 1 + 1) hs).iB.map (ofSplit q).pr *ᵥ z₀) r)) (tau : ∀ h, (v.τ

theorem **val_det** {a : Type} [Fintype a] [DecidableEq a] {g : GL a k} (hg : g ∈ KO k a) : valuation k ((Matrix.GeneralLinearGroup.det g : kˣ) : k) = 1

theorem **iter_congr** {r t : Type} [Fintype r] [DecidableEq r] [Fintype t] [DecidableEq t] (P : TypeDatum k r) {μ μ' : ℕ → kˣ →* ℂˣ} (h : ∀ j, IsTame (μ j)) (h' : ∀ j, IsTame (μ' j)) (e : ∀ j (x : kˣ), valuation k (x : k) = 1 → μ j x = μ' j x) (j : ℕ) : P.iter t μ h j = P.iter t μ' h' j

theorem **withMu_same** {s κ : ℕ} (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t] (μ : ℕ → kˣ →* ℂˣ) (h : ∀ j, IsTame (μ j)) (e : ∀ j (x : kˣ), valuation k (x : k) = 1 → μ j x = S.μ j x) : ((withMu S μ h).T t).lev = (S.T t).lev ∧ ((withMu S μ h).T t).Phi = (S.T t).Phi ∧ ((withMu S μ h).T t).PhiTop = (S.T t).PhiTop

theorem **_root_.GlobalTheory.SeeSaw.GModel.phi_sb** (G : _root_.GlobalTheory.SeeSaw.GLoc v) : (∀ j a H, (G.S.T (Fin D.R.t)).Phi j a H ∈ _root_.GlobalTheory.SplitWeil.SB _) ∧ (G.S.T (Fin D.R.t)).PhiTop ∈ _root_.GlobalTheory.SplitWeil.SB _

theorem **volLast_ne** (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) : Measure.map (c.jac j 0 h).d.emb0 (c.x (j + 1)).vol.μ ≠ 0

theorem **haar_map** {S S₀ : HSp D} (d : HypDec S S₀ 0) (μ : Measure (GA D S₀)) [μ.IsHaarMeasure] : (μ.map d.emb0).IsHaarMeasure

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsFramedAdapted

theorem **besVec_ne_zero** (B : SBes X h u ψ A) : B.d.i₀ *ᵥ vecB B.o B.ζ ≠ 0

theorem **besScale_unique** (B : SBes X h u ψ A) {μ μ' : D.host.E} (h₁ : ∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = μ * X.ιE (X.zR u (Fin.cast (rank_eq X h) k))) (h₂ : ∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = μ' * X.ιE (X.zR u (Fin.cast (rank_eq X h) k))) : μ = μ'

theorem **Framed.lines** {B : SBes X h u ψ A} (hB : Framed B) (a : Fin D.R.rS) : ∃ s s' : D.host.E, (∀ k, B.d.e k a = s * X.ιE (X.eR a (Fin.cast (rank_eq X h) k))) ∧ ∀ k, B.d.e' k a = s' * X.ιE (X.fR a (Fin.cast (rank_eq X h) k))

theorem **Framed.e'_eq** {A' : Formalism.BasicNotions.TH.HSp D} {u' : D.what₂.Word} {B : SBes X h u ψ A} {B' : SBes X h u' ψ A'} (hB : Framed B) (hB' : Framed B') (e : B'.d.e = B.d.e) : B'.d.e' = B.d.e'

theorem **Framed.shear_eq_zero** {B B' : SBes X h u ψ A} (hB : Framed B) (hB' : Framed B') {c : Matrix (Fin D.R.rS) (Fin D.R.rS) D.host.E} (e : B'.d.e = B.d.e) (e' : B'.d.e' = B.d.e' + B.d.e * c) : c = 0

theorem **exists_framed_adapted** (F₀ : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F₀) (hE : IdE X) (hψ : IsStdChar D ψ (-1)) (μ : D.host.E) (hμ : μ ≠ 0) (vA : Vol (D.sp.X 0)) : ∃ B : SBes X h u ψ (D.sp.X 0), Framed B ∧ B.Adapted tr.wt ∧ ∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = μ * X.ιE (X.zR u (Fin.cast (rank_eq X h) k))

theorem **Placed.adapted** (h : Placed f B) : B.Adapted (W.S.tr u).wt

theorem **Placed.levTrace** (h : Placed f B) : LevTrace f B

theorem **InPlace.common** (h : InPlace f B) : ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (ι : Type) (v : VCom f.c) (o : TDat f.c k ι) (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), f.c.Is ∧ v.Is ∧ VTie f v ∧ ShapeOff v o ∧ GPin G ∧ GRead f G pl ∧ BotTie f B v o G ∧ ∃ μ lam s : D.host.E, μ ≠ 0 ∧ μ * lam = -D.host.xi0 ^ (D.R.m / 2) ∧ (s = (W.S.X.B.g : D.host.E) ∨ s = -(W.S.X.B.g : D.host.E)) ∧ (∀ j hj, v.τ j hj *ᵥ o.ζ₀ = lam • (f.c.bes j 0 hj).ζ) ∧ (∀ r, (B.d.i₀ *ᵥ vecB B.o B.ζ) r = μ * W.S.X.ιE (W.S.X.zR u (Fin.cast (rank_eq W.S.X W.Q.instEq) r))) ∧ (∀ r (a : Fin D.R.rS), B.d.e r a = μ * s ^ (D.R.rS - (a : ℕ)) * W.S.X.ιE (W.S.X.eR a (Fin.cast (rank_eq W.S.X W.Q.instEq) r))) ∧ ∀ r (a : Fin D.R.rS), B.d.e' r a = (star (μ * s ^ (D.R.rS - (a : ℕ))))⁻¹ * W.S.X.ιE (W.S.X.fR a (Fin.cast (rank_eq W.S.X W.Q.instEq) r))

theorem **shift_tOf** (a₀ a : GA D (D.sp.X 0)) : (iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u) * tOf f B a₀ a = embA B.d B.o (a₀ * a) * (show GA D (D.sp.Y 0) from f.g.b 0)

theorem **tOf_eq_iff** (a₀ a : GA D (D.sp.X 0)) (t : GA D (D.sp.Y 0)) : tOf f B a₀ a = t ↔ embA B.d B.o (a₀ * a) * (show GA D (D.sp.Y 0) from f.g.b 0) = (iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u) * t

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsIsComOfOf

theorem **comb_conv** (c c' : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (conv ψ S T c c') f = comb ψ S T c (comb ψ S T c' f)

theorem **comb_unit** (f : WO S T → ℂ) : comb ψ S T (Finsupp.single 0 1) f = f

theorem **comb_convList** {α : Type} (l : List α) (c : α → WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (convList ψ S T (l.map c)) f = l.foldr (fun i f' => comb ψ S T (c i) f') f

theorem **mem_support_conv** {c c' : WO S T →₀ ℂ} {w : WO S T} (hw : w ∈ (conv ψ S T c c').support) : ∃ x ∈ c.support, ∃ y ∈ c'.support, w = x + y

theorem **support_convList** {p : WO S T → Prop} (h0 : p 0) (hadd : ∀ x y, p x → p y → p (x + y)) {l : List (WO S T →₀ ℂ)} (hl : ∀ c ∈ l, ∀ x ∈ c.support, p x) : ∀ w ∈ (convList ψ S T l).support, p w

theorem **mem_symLevel** {Pg : Finset (Pl D.host.E)} {ψ₀ : ∀ PP : Pl D.host.E, AddChar (kP PP) ℂ} (g : Formalism.L2.DIV.G0 Pg D.R.r) : g ∈ (symLevel Pg ψ₀).H ↔ ∀ PP, g PP ∈ Formalism.L2.Col.HPsymF (kP PP.1) D.R.r D.R.k D.R.rS

theorem **symLevel_chi** {Pg : Finset (Pl D.host.E)} {ψ₀ : ∀ PP : Pl D.host.E, AddChar (kP PP) ℂ} (g : (symLevel Pg ψ₀).H) : (symLevel Pg ψ₀).chi g = ∏ PP : Pg, Formalism.L2.Col.chiSymF (ψ₀ PP.1) D.R.r D.R.k D.R.rS (g.1 PP, (mem_symLevel g.1).1 g.2 PP)

theorem **norm** : star ζ₀ ·ᵥ a.Lz.gram *ᵥ ζ₀ = star π * π

theorem **line** (h : π ≠ 0) : IsLineVec a.Lz ζ₀

theorem **vec** (n) (h : n = D.R.m) (r) : vecB (a.oT n h) ζ₀ r = if r.val + 1 = (D.sp.X n).rank then π else 0

theorem **unitLine_spec** (D : D7) : (unitLine D).rank = 1 ∧ (unitLine D).gram = 1 ∧ (unitLine D).SigOK ∧ (unitLine D).IsPosDef

theorem **chiC_apply** (i P x) : ((chiC I i P x : ℂˣ) : ℂ) = B.embLL ((Formalism.L2.Col.chiP I.Ch P i x : B.LLˣ) : B.LL)

theorem **IsComOf.nonempty** {c} (H : IsComOf I l Z c) : c.Pg.Nonempty

theorem **BaseCoset.intAt** (H : BaseCoset A) {j : ℕ} (hj : j + 1 = B.R.m) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup I.mu0)) (hP : P.1 ∈ (B.primes X).vGs) (i : Fin (B.R.N j)) : Formalism.L2.Col.IntAt P.1 (baseY A j P i - if i.val = B.R.k then ((Formalism.L2.Col.qOf P.1 : B.fields.E))⁻¹ else 0)

theorem **weil** {j : ℕ} (hj : j < B.R.m) : IsWeilDatumO a.c.ψ _ _ (a.c.μ (j + 1)) (a.c.μ (j + 1)) (a.k j).om

theorem **norm** : star o.ζ₀ ·ᵥ a.Lz.gram *ᵥ o.ζ₀ = ((B.T v : ℕ) : B.host.E)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsIsFundU

theorem **exists_isFundU** {ρ : ℕ} (v : VolU D ρ) : ∃ 𝓊, IsFundU v 𝓊 ∧ v.ν 𝓊 = 1

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsIsParamFunOne

theorem **exists_cartan** {m : Type} [Fintype m] [DecidableEq m] {n : ℕ} (e : Fin n ≃ m) {C : Matrix m (Fin n) ℂ} {J : Matrix m m ℂ} {G : Matrix (Fin n) (Fin n) ℂ} (hJ : J * J = 1) (hJ' : Jᴴ = J) (hG : IsUnit G.det) (h : Cᴴ * J * C = G) : ∃ s, IsCartanInv G s ∧ C * s = J * C

theorem **gramAt_unit** (V : HSp D) (τ : D.what₂.Tau) : IsUnit (gramAt D V τ).det

theorem **exists_real** {θ : InfFam D} (hθ : IsInfFam θ) (A B : HSp D) (hA : A.SigOK) (h1 : 1 ≤ A.rank) {t : ℕ} (ht : 1 ≤ t) (hB : B.rank = A.rank + t) : ∃ sA sB cA a, HidaSetting θ A B sA sB t cA ∧ IsMaj A a ∧ ∀ τ, (compAt θ τ a.1)ᴴ * (gramAt D A τ * sA τ) * compAt θ τ a.1 = 1

theorem **isSplit_two** (v : FinPl D) (h2 : (2 : 𝒪 D.host.F) ∈ v.asIdeal) : Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **isSphOff_of_vac** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {sS : RealMat S} {sT : RealMat T} {LS : Lat S} {LT : Lat T} {Φ : WO S T → ℂ} (h : IsVacO ψ θ S T sS sT LS LT Φ) (Sf : Finset (FinPl D)) : IsFockVec ψ θ S T sS sT Φ ∧ IsSphOff ψ S T LS LT Sf Φ ∧ Φ ≠ 0

theorem **exists_pair_unram** (A B : HSp D) (hA : A.SigOK) (hB : B.SigOK) (h1 : 1 ≤ A.rank) {t : ℕ} (ht : 1 ≤ t) (hr : B.rank = A.rank + t) (LA : Lat A) (LB : Lat B) (hLA : IsSelfDualC A LA) (hLB : IsSelfDualC B LB) : ∃ ψ θ χA χB sA sB cA a om ν ℱ Φ, Pair ψ θ A B χA χB sA sB t cA a om ν ℱ ∧ IsFockVec ψ θ A B sA sB Φ ∧ wcoef om ν ℱ Φ Φ 1 = 1 ∧ ∀ Sf, Unram ψ A B χB LA LB Sf Φ Φ

theorem **exists_param_one** (V : HSp D) {G : Type} [Group G] (v : FinPl D) (j : Gv V v →* G) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v) : ∃ B, LFun.IsParamVec D.sp.cm.sig V.gram (frameLat V) h ((PRep.rightReg ℂ G ℂ).comp j) (fun _ => 1) B

theorem **exists_isSphCoef_one** (V : HSp D) (v : FinPl D) : ∃ par, IsSphCoef V (frameLat V) v par fun _ => 1

theorem **exists_isParamFun_one** (V : HSp D) : ∃ par, ∀ Sf : Finset (FinPl D), LFun.IsParamFun D.sp.cm.sig V.gram (frameLat V) (Sf : Set (FinPl D)) (fun _ => (1 : ℂ)) par ∧ SphFun V (frameLat V) Sf fun _ => 1

theorem **measurable_wcoef** (hom : IsWeilDatumO ψ A B χA χB om) (hν : IsCovolOne (ratL B A) ν 𝓕) (h₁ : IsThetaFunO ψ A B Φ₁) (h₂ : IsThetaFunO ψ A B Φ₂) : StronglyMeasurable (wcoef om ν 𝓕 Φ₁ Φ₂)

theorem **bounded_wcoef** (hom : IsWeilDatumO ψ A B χA χB om) (hν : IsCovolOne (ratL B A) ν 𝓕) (h₁ : IsThetaFunO ψ A B Φ₁) (h₂ : IsThetaFunO ψ A B Φ₂) : ∃ C : ℝ, ∀ g, ‖wcoef om ν 𝓕 Φ₁ Φ₂ g‖ ≤ C

theorem **openOff_empty** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {A B : HSp D} {χA χB : (AE D)ˣ →* Circle} {sA : RealMat A} {sB : RealMat B} {t : ℕ} {cA : CptF A} {a : GL (Fin A.rank) (AE D)} {om : WRep A B} {ν : Measure (WO A B)} {𝓕 : Set (WO A B)} (h : Pair ψ θ A B χA χB sA sB t cA a om ν 𝓕) (hq : ∀ τ, (A.sig τ).2 = 0) : openOff A (frameLat A) ∅ = Kmax A a

theorem **heckeForm_hypotheses** (D : D7) {n t : ℕ} (hn : 1 ≤ n) (ht : 1 ≤ t) : ∃ (A B : HSp D) (ψ : _) (θ : _) (χA χB : _) (sA : _) (sB : _) (cA : _) (a : _) (om : _) (ν : _) (𝓕 : _) (Φ : _) (m : Measure (GA D A)) (par : _), A.rank = n ∧ Pair ψ θ A B χA χB sA sB t cA a om ν 𝓕 ∧ wittIndex B = 0 ∧ IsFockVec ψ θ A B sA sB Φ ∧ wcoef om ν 𝓕 Φ Φ 1 = 1 ∧ Unram ψ A B χB (frameLat A) (frameLat B) ∅ Φ Φ ∧ m.IsHaarMeasure ∧ (fun _ => (1 : ℂ)) ∈ A0 θ A cA ∧ LFun.IsParamFun D.sp.cm.sig A.gram (frameLat A) ((∅ : Finset (FinPl D)) : Set (FinPl D)) (fun _ => (1 : ℂ)) par ∧ SphFun A (frameLat A) ∅ (fun _ => 1) ∧ ∀ s, IntegrableOn (integrand om m ν 𝓕 (frameLat A) a Φ Φ (fun _ => 1) (fun _ => 1) s) (openOff A (frameLat A) ∅) m

theorem **up_smul** {A B : HSp D} (om : WRep A B) (m : Measure (GA D A)) (Φ : WO A B → ℂ) (c : ℂ) (u : GA D A → ℂ) : up om m Φ (c • u) = c • up om m Φ u

theorem **up_eq_conj_lift** {A B : HSp D} (om : WRep A B) (m : Measure (GA D A)) (Φ : WO A B → ℂ) (u : GA D A → ℂ) (h : GA D B) : up om m Φ u h = conj (GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.lift m (fun g h => conj (thetaKO om g h Φ)) u h)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsLatInd

theorem **up_zero** (q : GlobalTheory.FinPl.SplitPt D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) : (ofSplit q).up FS FT 0 = 0

theorem **exists_latInd** (q : GlobalTheory.FinPl.SplitPt D k) (hU : q.UnitAA) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hM : IsSelfDualC S M) (hM' : IsSelfDualC T M') : ∃ Φ₀ : SB (Matrix Y Y' k), Φ₀.1 = Set.indicator {x | (ofSplit q).up FS FT x ∈ latO S T (M : Set _) (M' : Set _) (· = q.v)} (fun _ => 1)

theorem **exists_locPl** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (q : GlobalTheory.FinPl.SplitPt D k) (hU : q.UnitAA) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (Φ : SB (Matrix Y Y' k)) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hM : IsSelfDualC S M) (hM' : IsSelfDualC T M') : ∃ e o h1 h2 h3, ((k, q, hU, Y, Y', FS, FT, Φ, e, h1, o, h2, h3) : LocPl ψ S T M M').q.v = q.v

theorem **comb_comb** (L : LocPl ψ S T M M') {Tm} (hT : (ofSplit L.q).IsModel ψ L.FS L.FT Tm) (Φ₀ : SB (Matrix L.Y L.Y' L.k)) (h₀ : Φ₀.1 = Set.indicator {x | (ofSplit L.q).up L.FS L.FT x ∈ latO S T (M : Set _) (M' : Set _) (· = L.q.v)} (fun _ => 1)) : Tm L.Φ = comb ψ S T L.e (comb ψ S T L.o (Tm Φ₀))

theorem **o_ne_zero** (hψ : IsStdChar D ψ (-1)) (hM : IsSelfDualC S M) (hM' : IsSelfDualC T M') (L : LocPl ψ S T M M') : L.o ≠ 0

theorem **e_ne_zero** (hψ : IsStdChar D ψ (-1)) (L : LocPl ψ S T M M') (hΦ : L.Φ ≠ 0) : L.e ≠ 0

theorem **prodDat_one** (L : LocPl ψ S T M M') (x : Matrix L.Y L.Y' L.k) : prodDat (fun _ : Fin 1 => L) ((ofSplit L.q).up L.FS L.FT x) = L.Φ.1 x

theorem **prodDat_zero** (L : Fin 0 → LocPl ψ S T M M') (Λ : AddSubgroup (WO S T)) : prodDat L = halfInd 0 Λ

theorem **e_mul_e** (L L' : LocPl ψ S T M M') (h : L.q.v ≠ L'.q.v) : L.q.e * L'.q.e = 0

theorem **prodDat_sum** {n : ℕ} (L : Fin n → LocPl ψ S T M M') (inj : Function.Injective fun i => (L i).q.v) (x : ∀ i, Matrix (L i).Y (L i).Y' (L i).k) : prodDat L (∑ i, (ofSplit (L i).q).up (L i).FS (L i).FT (x i)) = ∏ i, (L i).Φ.1 (x i)

theorem **infinite_unit** (D : D7) : {v : GlobalTheory.FinPl.Pl D | ∃ p : GlobalTheory.FinPl.SplitPt D (Fv D.host.F v), p.v = v ∧ p.UnitAA}.Infinite

theorem **exists_family** (hψ : IsStdChar D ψ (-1)) (hM : IsSelfDualC S M) (hM' : IsSelfDualC T M') (n : ℕ) : ∃ L : Fin n → LocPl ψ S T M M', (Function.Injective fun i => (L i).q.v) ∧ prodDat L 0 = 1 ∧ ∀ i, (L i).e ≠ 0 ∧ (L i).o ≠ 0

theorem **exists_closed** (D : D7) (S T : HSp D) (n : ℕ) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsStdChar D ψ (-1) ∧ IsSelfDualC S M ∧ IsSelfDualC T M' ∧ ∃ L : Fin n → LocPl ψ S T M M', (Function.Injective fun i => (L i).q.v) ∧ prodDat L 0 = 1 ∧ ∀ i, (L i).e ≠ 0 ∧ (L i).o ≠ 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsPlacesSingleton

theorem **jpq_eq_one** {A : HSp D} (hp : A.IsPosDef) (τ : D.what₂.Tau) : Formalism.UG.Jpq (A.sig (tauE D τ)).1 (A.sig (tauE D τ)).2 = 1

theorem **isCartanInv_one** {A : HSp D} (hA : A.SigOK) (hp : A.IsPosDef) (τ : D.what₂.Tau) : IsCartanInv (gramAt D A τ) 1

theorem **exists_defin** {θ : InfFam D} (hθ : IsInfFam θ) {A : HSp D} (hA : A.SigOK) (hp : A.IsPosDef) : ∃ cA a, Defin θ A cA a

theorem **exists_places** (A : HSp D) : ∃ β, ∀ Z : Finset (Formalism.L2.DIV.Pl D.host.E), (∀ w ∈ Z, Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **exists_places_singleton** (A : HSp D) : ∃ β w, Places A β {w}

theorem **bcParam_rho** {q : ℕ} (hq : q ≠ 0) (n : ℕ) : GlobalTheory.LGrp.bcParam n (LFun.Model.rhoU hq n) = GlobalTheory.IndefiniteRowSideFullHecke.SphericalThetaInertPlace.str (Units.mk0 (q : ℂ) (by exact_mod_cast hq)) n

theorem **seg_eq_str** {E : Type} [Field E] [NumberField E] {w : HeightOneSpectrum (𝒪 E)} {q : ℕ} (hq : q ≠ 0) (h : Ideal.absNorm w.asIdeal = q ^ 2) (n : ℕ) : LFun.seg w 1 n = GlobalTheory.IndefiniteRowSideFullHecke.SphericalThetaInertPlace.str (Units.mk0 (q : ℂ) (by exact_mod_cast hq)) n

theorem **isParamFun_one** (V : HSp D) (S : Set (FinPl D)) : LFun.IsParamFun D.sp.cm.sig V.gram (frameLat V) S (fun _ => (1 : ℂ)) (LFun.segFam 1 V.rank)

theorem **form_one** (θ : InfFam D) {A : HSp D} (hA : IsAniso A) (h5 : 5 ≤ A.rank) (cA : CptF A) (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)) : Form θ A cA β Z 1 0 A.rank (fun _ => 1) (LFun.segFam 1 A.rank)

theorem **unrOff_pow** {η : (AE D)ˣ →* Circle} (hη : UnramEverywhere D η) (n : ℕ) : GlobalTheory.StepLaw.UnrOff (η ^ n) ∅

theorem **detPlus_neg** {m n : ℕ} (a : Matrix (Fin n) (Fin n) ℂ) (b : Matrix (Fin m) (Fin m) ℂ) : GlobalTheory.WeilParts.detPlus 1 (-1) a b = 1

theorem **exists_occ** {θ : InfFam D} (hθ : IsInfFam θ) {A W : HSp D} (hA : A.SigOK) (hp : A.IsPosDef) (hW : W.SigOK) (hn : ∀ τ, IsCartanInv (gramAt D W τ) (-1)) : ∃ ψ η om Φ, GlobalTheory.WeilParts.IsEtaTilde D η ∧ IsWeilDatumO ψ A W (η ^ A.rank) (η ^ W.rank) om ∧ Φ ≠ 0 ∧ ∀ β, Formalism.L2.DIV.IsFrame β (frameLat A) → ∀ Z (m : Measure (GA D A)), m.IsHaarMeasure → Occ ψ θ (fun _ => -1) om m β Z (fun _ => 1) Φ

theorem **isCartanInv_neg** {W : HSp D} (hW : W.SigOK) (hp : W.IsPosDef) (τ : D.what₂.Tau) : IsCartanInv (gramAt D (GlobalTheory.NegSpace.neg W) τ) (-1)

theorem **hypotheses** (D : D7) (n r : ℕ) : ∃ (ψ : _) (θ : _) (A W : HSp D) (χA χW : _) (om : WRep A W) (cA : _) (a : _) (β : _) (w : _) (m : Measure (GA D A)) (sW : RealMat W) (Φ : _), A.rank = n ∧ W.rank = r ∧ Defin θ A cA a ∧ W.SigOK ∧ (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) ∧ IsWeilDatumO ψ A W χA χW om ∧ Places A β {w} ∧ m.IsHaarMeasure ∧ Occ ψ θ sW om m β {w} (fun _ => 1) Φ ∧ (r = 0 → 5 ≤ n → Form θ A cA β {w} χW W.rank n (fun _ => 1) (LFun.segFam 1 n))

theorem **lev_prinDat** (A : HSp D) (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)) : Formalism.L2.DIV.lev D.sp.cm A.gram β (prinDat A Z) = Formalism.L2.DIV.Kg D.sp.cm A.gram β Z

theorem **dblInt_delta** {ψ : AddChar (AF D.host.F) ℂ} {A W : HSp D} {χ : (AE D)ˣ →* Circle} {om : WRep A W} {omh : WRep (dblSp A) W} (h : GlobalTheory.IndefiniteRowSideFullHecke.SphericalZero.ThetaClause ψ A W χ om omh) (m : Measure (GA D A)) [m.IsHaarMeasure] (u : GA D A → ℂ) {f₁ f₂ : WO A W → ℂ} (h₁ : IsThetaFunO ψ A W f₁) (h₂ : IsThetaFunO ψ A W f₂) (x : GA D W) : Eis.dblInt A χ⁻¹ m (fun g => thetaKO omh g x (delta A W f₁ f₂)) (fun g => conj (u g)) (fun g => conj (u g)) = up om m f₁ u x * conj (up om m f₂ u x)

theorem **noLift_of_kills** {ψ : AddChar (AF D.host.F) ℂ} {A W : HSp D} {χ : (AE D)ˣ →* Circle} {om : WRep A W} {omh : WRep (dblSp A) W} (h : GlobalTheory.IndefiniteRowSideFullHecke.SphericalZero.ThetaClause ψ A W χ om omh) {m : Measure (GA D A)} [m.IsHaarMeasure] {u : GA D A → ℂ} (hk : Kills ψ χ omh m u) : NoLift ψ om m u

theorem **gap_of_form** {θ : InfFam D} {A : HSp D} {cA : CptF A} {β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)} {Z : Finset (Formalism.L2.DIV.Pl D.host.E)} {χ : (AE D)ˣ →* Circle} {r t : ℕ} {u : GA D A → ℂ} {par : LFun.Fam D.host.E} (h : Form θ A cA β Z χ r t u par) : r + 5 ≤ t

theorem **not_occ_zero** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {A W : HSp D} (hA : A.SigOK) (hp : A.IsPosDef) (sW : RealMat W) (om : WRep A W) (m : Measure (GA D A)) (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)) (Φ : WO A W → ℂ) : ¬ Occ ψ θ sW om m β Z 0 Φ

theorem **not_form_one** {θ : InfFam D} {A : HSp D} {cA : CptF A} {β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)} {Z : Finset (Formalism.L2.DIV.Pl D.host.E)} {χ : (AE D)ˣ →* Circle} {r t : ℕ} {par : LFun.Fam D.host.E} (hr : 0 < r) (hn : A.rank = r + t) : ¬ Form θ A cA β Z χ r t (fun _ => 1) par

theorem **form_ne_zero** {θ : InfFam D} {A : HSp D} {cA : CptF A} {β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)} {Z : Finset (Formalism.L2.DIV.Pl D.host.E)} {χ : (AE D)ˣ →* Circle} {r t : ℕ} {par : LFun.Fam D.host.E} {u : GA D A → ℂ} (h : Form θ A cA β Z χ r t u par) : u ≠ 0 ∧ A.rank = r + t

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsRamNeZero

theorem **pair_empty** {n : Type} [IsEmpty n] : IsEmpty (Pair k n)

theorem **offDat_empty** (d : Dat D u) (j : ℕ) (hj : (D.sp.X j).rank = 0) [Nonempty ι] : IsEmpty (OffDat d k ι)

theorem **HasFrames.nonempty** {p : SplitPt D k} (h : HasFrames p) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (hn : Fintype.card n = S.rank) : Nonempty (Frame p S n)

theorem **offDat_nonempty** (d : Dat D u) (p : ι → SplitPt D k) (hp : ∀ i, HasFrames (p i)) (I : List ι) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (hX : ∀ j, 0 < (D.sp.X j).rank) (hL : d.Lz.rank = 1) : ∃ o : OffDat d k ι, o.p = p ∧ o.I = I ∧ o.χ = χ

theorem **symplO_kap** (H : PinOff o) (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) (S : HSp D) (x y : WO S d.Lz) : symplO _ S (kap (o.τ j h) S x) (kap (o.τ j h) S y) = symplO S d.Lz x y

theorem **atT_off** (H : PinOff o) {v : Pl D} (hv : o.atT v) : ¬ onS d v

theorem **finite_index** (H : PinOff o) : Finite ι

theorem **withS_iff** (S : ∀ j, Col D u j →ₗ[ℂ] Col D u j) : ((withS d S).Supp1 ↔ d.Supp1) ∧ ((withS d S).Supp2 ↔ d.Supp2) ∧ ((withS d S).Base ↔ d.Base)

theorem **valS_of** (R : RdT o) (S : ∀ j, Col D u j →ₗ[ℂ] Col D u j) (ε : ℕ → ι → 𝓀[k]ˣ) (hS : ∀ j, j ≤ D.R.m → ∀ ψ ∈ d.Ram j, ∀ y, d.rd j (S j ψ) y = o.I.foldr (fun i φ => sOp (o.β j i) (o.Q j i) o.q (resChar (o.χ i) (o.hχ i) o.q) (ε j i) φ) (d.rd j ψ y)) (hV : ∀ j σ h, ValAd o j σ h (ε j)) : (withS d S).ValS

theorem **norm_tau** (H : PinOff o) (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) : ∃ lam : D.host.E, o.τ j h *ᵥ o.ζ₀ = lam • (d.bes j 0 h).ζ ∧ star lam * lam * (star (d.bes j 0 h).ζ ·ᵥ (d.bes j 0 h).L.gram *ᵥ (d.bes j 0 h).ζ) = -(D.host.xi0 * star D.host.xi0) * (star o.ζ₀ ·ᵥ d.Lz.gram *ᵥ o.ζ₀)

theorem **exists_ram_ne_zero** (R : RdT o) {ψ : Col D u 0} (hψ : ψ ∈ (D.mem u).A.Mram) (h0 : ψ ≠ 0) : ∃ y, d.rd 0 ψ y ≠ 0 ∧ o.Ram 0 (d.rd 0 ψ y)

theorem **isS_unique** (R : RdT o) {j : ℕ} (hj : j ≤ D.R.m) {ψ ψ' : Col D u j} (h : ψ ∈ d.Ram j) (h' : ψ' ∈ d.Ram j) (e : ∀ y, d.rd j ψ y = d.rd j ψ' y) : ψ = ψ'

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsStage

theorem **mem_S** {L : LevCh D Unit} (hL : L.Is ()) (θ ψ P) (k : Fin 2) (Q : Pl D.host.E) (hk : ((D.host.g k : ℕ) : 𝒪 D.host.E) ∈ Q.asIdeal) : Q.under (𝒪 D.host.F) ∈ (comL L θ ψ P).S

theorem **comL_is** {L : LevCh D Unit} (hL : L.Is ()) {θ : InfFam D} (hθ : IsInfFam θ) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (P : LevelP L.Pg D.R.r ℂ) : (comL L θ ψ P).Is

theorem **pg_nonempty** {L : LevCh D Unit} (hL : L.Is ()) : L.Pg.Nonempty

theorem **exists_com** (C : Formalism.L2.Host.CharFrame D) : ∃ c : Com D, c.Pg.Nonempty ∧ ∀ P : LevelP c.Pg D.R.r ℂ, Com.Is { c with P

theorem **exists_haar** (V : HSp D) : ∃ (ν : Measure (GA D V)) (𝒢 : Set (GA D V)), ν.IsHaarMeasure ∧ IsFundamentalDomain (GQ V) 𝒢 ν

theorem **exists_stage** (hN : GlobalTheory.Published.DiscreteSeriesExists) (c : Com D) (j : ℕ) (hj : j ≤ D.R.m) : ∃ s : PiLocalization₂ D, s.V = D.sp.Y j ∧ s.Is c j

theorem **exists_reach** (hN : GlobalTheory.Published.DiscreteSeriesExists) (c : Com D) (hstep : ∀ k < D.R.m, ∀ s s' : PiLocalization₂ D, s.Is c k → s'.Is c (k + 1) → ∃ p : Step c s s', p.Is (k + 1)) (s₀ : PiLocalization₂ D) (h₀ : s₀.Is c 0) : ∀ k ≤ D.R.m, ∃ s : PiLocalization₂ D, s.Is c k ∧ ∃ θ, Reach c s₀ k s θ

theorem **it_ne_zero** {c : Com D} {s s' : PiLocalization₂ D} {p : Step c s s'} {j : ℕ} (h : p.Is j) : p.It ≠ 0

theorem **chainPar_two** (c : Com D) (B : GlobalTheory.LFun.Fam D.host.E) (w) : (chainPar c B 2 w).card = (B w).card + 2 * D.R.t

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsSumFrame

theorem **exists_sumFrame** {S A L : HSp D} (o : OrthDec S A L) {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) (β : Frame p A (Fin A.rank)) (γ : Frame p S n) : ∃ β₀ : Frame p S (Fin A.rank ⊕ Fin 1), IsSumFrame o ζ β β₀

theorem **exists_pairDat** {v : Pl D} (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) {S A L : HSp D} (o : OrthDec S A L) {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) (hA : 0 < A.rank) : ∃ a : PairDat (fun _ : Unit => Model.splitPt τ) A S (Fin A.rank), IsSumFrame o ζ (a.α ()) (a.α₀ ())

theorem **exists_data** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ (v : Pl D) (p : Unit → SplitPt D (Fv D.host.F v)) (B : BesLev D 1) (J : JacLev D 1) (B' : BesLev D 0) (_ : StepBF ψ B J) (_ : StepFB ψ J B') (_ : Link B B'.L B'.ζ) (Λ : DblLat B J B') (a : PairDat p B.A B.V₀ (Fin B.A.rank)) (_ : MidDat p J.W₀ (Fin J.W₀.rank)) (a' : PairDat p B'.A B'.V₀ (Fin B'.A.rank)), 0 < B.A.rank ∧ 0 < J.W₀.rank ∧ 0 < B'.A.rank ∧ IsSelfDualC _ Λ.L₀ ∧ IsSelfDualC _ Λ.M₀ ∧ IsSelfDualC _ Λ.L₁ ∧ IsSelfDualC _ Λ.M₁ ∧ IsSelfDualC _ Λ.L₂ ∧ IsSelfDualC _ Λ.M₂ ∧ IsSumFrame B.o B.ζ (a.α ()) (a.α₀ ()) ∧ IsSumFrame B'.o B'.ζ (a'.α ()) (a'.α₀ ())

theorem **collapse_hyp** (A B : Type) [Group A] [Group B] [TopologicalSpace A] [TopologicalSpace B] [IsTopologicalGroup A] [IsTopologicalGroup B] [T2Space A] [T2Space B] [CompactSpace B] (R₁ : Subgroup A) (R₂ : Subgroup B) (h₁ : IsClosed (R₁ : Set A)) (h₂ : IsClosed (R₂ : Set B)) : let G₁ : Subgroup (A × B)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsTransportAdelicCorrection

theorem **exists_finM** (V : HSp D) {n : ℕ} (hn : V.rank = n) (y : Formalism.UG.UPts (Formalism.L2.Forms.sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ a ∈ GfC V, GlobalTheory.AdForm.finM V hn a = y.1.1

theorem **exists_transport_adelic_correction** {X : HidaSetting D} {F : Formalism.L2.Forms.Forms0 D u} (tr : Formalism.L2.SG.ColDict.Transport X u F) : ∃ a, IsTransportAdelicCorrection tr a

theorem **exists_symOrd'** {r : ℕ} (ρ k : ℕ) (h : ρ + k ≤ r) : ∃ π : Equiv.Perm (Fin r), IsSymOrd ρ k π

theorem **exists_isSymLev** (Pg : Finset (Pl D.host.E)) : ∃ P : LevelP Pg D.R.r ℂ, IsSymLev P

theorem **exists_isTypePlaces** : ∃ Pg : Finset (Pl D.host.E), GlobalTheory.Core.Mi.IsTypePlaces Pg

theorem **not_isTypePlaces_empty** : ¬ GlobalTheory.Core.Mi.IsTypePlaces (D

theorem **pinBot_of_matches** (h0 : 0 + 2 * D.R.rS + 2 = D.R.m) (B : SBes S.X S.inst u d.c.ψ (D.sp.X 0)) (hB : B.lev = d.B 0 D.R.rS h0) (hA : B.Adapted (S.tr u).wt) (t : GA D (D.sp.Y 0)) (ht : Matches S.X S.inst (d.lev 0) (levChi (D.sp.Y 0) d.bot.β d.bot.L) t) (hrd : d.rd 0 = (chart0 (F₀ u)).rd d.c.θ) : ∃ t₀ : GA D (D.sp.Y 0), PinBot S { d with t

theorem **level_side** (B : SBes S.X S.inst u d.c.ψ (D.sp.X 0)) (a₀ a : GA D (D.sp.X 0)) (ha : a ∈ Subgroup.center (GA D (D.sp.X 0))) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (x t₀ : GA D (D.sp.Y 0)) (h : Matches S.X S.inst J χ (x⁻¹ * embA B.d B.o (a₀ * a) * t₀)) (k : GA D (D.sp.X 0)) : t₀⁻¹ * embA B.d B.o k * t₀ ∈ J ↔ x⁻¹ * embA B.d B.o (a₀ * k * a₀⁻¹) * x ∈ rowJ S.X S.inst

theorem **rank_ok** : 0 + 2 * D.R.rS + 2 = D.R.m

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ExistsUnitarySupportedAtPlaceOfMulEqIdem

theorem **Place.isChar_ofSplitPt** (q : GlobalTheory.FinPl.SplitPt D k) {ψ : AddChar (AF D.host.F) ℂ} {χ ψk : AddChar k Circle} (hχ : q.IsChar ψ χ) : (Place.ofSplitPt q).IsChar ψ ψk ↔ ∀ x, ψk x = χ ((Place.ofSplitPt q).xi * x)

theorem **Place.down_ofSplitPt** {q : GlobalTheory.FinPl.SplitPt D k} {S T : HSp D} {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (α : GlobalTheory.FinPl.Frame q S n') (β : GlobalTheory.FinPl.Frame q T n) (t : WO S T) : (Place.ofSplitPt q).down (Frame.ofFinPl α) (Frame.ofFinPl β) t = GlobalTheory.FinPl.rd α β t

theorem **Place.isLoc_ofSplitPt** {q : GlobalTheory.FinPl.SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (F : GlobalTheory.FinPl.Frame q S n) (g : GL n k) : (Place.ofSplitPt q).IsLoc (Frame.ofFinPl F) (F.gl g) g

theorem **exists_place** {e : AE D} (he : IsSplitIdem D e) : ∃ v : HeightOneSpectrum (𝒪 D.host.F), letI

theorem **nonempty_place** (D : D7) : ∃ v : HeightOneSpectrum (𝒪 D.host.F), letI

theorem **e_ι** (x : k) : p.e * p.ι x = p.ι x

theorem **exists_isChar** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ ψk, p.IsChar ψ ψk

theorem **up_half** (x) : IsHalf p.e (p.up FS FT x)

theorem **up_add** (x y) : p.up FS FT (x + y) = p.up FS FT x + p.up FS FT y

theorem **down_add** (x y) : p.down FS FT (x + y) = p.down FS FT x + p.down FS FT y

theorem **down_up** (x) : p.down FS FT (p.up FS FT x) = x

theorem **up_down** {t : WO S T} (ht : IsHalf p.e t) : p.up FS FT (p.down FS FT t) = t

theorem **up_cont** : Continuous (p.up FS FT)

theorem **down_cont** : Continuous (p.down FS FT)

theorem **half_add** {e : AE D} {x y : WO S T} (hy : IsHalf e y) : IsHalf e (x + y) ↔ IsHalf e x

theorem **ext_apply** (Φ) (t) : ext p FS FT Φ t = if IsHalf p.e t then Φ.1 (p.down FS FT t) else 0

theorem **lat_compact** : IsCompact (((latO S T ΛS ΛT fun _ => True).comap (upH p FS FT) : AddSubgroup _) : Set (Matrix Y Y' k))

theorem **lat_open** : IsOpen (((latO S T ΛS ΛT fun _ => True).comap (upH p FS FT) : AddSubgroup _) : Set (Matrix Y Y' k))

theorem **ext_sb** (Φ) : IsSBHalf p.e (latO S T ΛS ΛT fun _ => True) (ext p FS FT Φ)

theorem **exists_isModel** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ Tm, p.IsModel ψ FS FT Tm ∧ Tm ≠ 0

theorem **exists_isComp** (χ : (AE D)ˣ →* Circle) : ∃ μ, p.IsComp χ μ

theorem **card_frame** {N : ℕ} {Z : Type} [Fintype Z] [DecidableEq Z] (F : Frame k N Z) : N = Fintype.card Z

theorem **exists_scaled** {N : ℕ} {Z Z' : Type} [Fintype Z] [DecidableEq Z] [Fintype Z'] [DecidableEq Z'] (F : Frame k N Z) (σ : Z' ≃ Z) (t : Z' → k) (ht : ∀ z, t z ≠ 0) : ∃ F' : Frame k N Z', ∀ r z, F'.β r z = t z * F.β r (σ z)

theorem **exists_witt** {V V₀ : HSp D} {σ : ℕ} (d : HypDec V V₀ σ) (G₀ : Frame k V₀.rank Y) : ∃ F : Frame k V.rank (Y ⊕ (Fin σ ⊕ Fin σ)), F.β = fromCols (d.i₀.map p.pr ∗ G₀.β) (fromCols (d.e.map p.pr) (d.e'.map p.pr))

theorem **exists_scaled'** {N : ℕ} {Z Z' : Type} [Fintype Z] [DecidableEq Z] [Fintype Z'] [DecidableEq Z'] (F : Frame k N Z) (τ : Z → Z') (hτ : Function.Injective τ) (hc : Fintype.card Z = Fintype.card Z') (t : Z → k) (ht : ∀ z, t z ≠ 0) : ∃ F' : Frame k N Z', ∀ r z, F'.β r (τ z) = t z ∗ F.β r z

theorem **xi_ne** : p.xi ≠ 0

theorem **hz_ne** {ρ : ℕ} (B : BesLev D ρ) : p.hz B ≠ 0

theorem **exists_core** {ρ : ℕ} (B : BesLev D ρ) {s : k} (hs : s ≠ 0) (FA : Frame k B.A.rank Y) : ∃ F, p.IsCoreFrame B s FA F

theorem **exists_jac** {ρ : ℕ} (J : JacLev D ρ) {s hh : k} (hs : s ≠ 0) (hh0 : hh ≠ 0) (F₀ : Frame k J.W₀.rank Y) : ∃ F : Frame k J.W.rank (Y ⊕ Fin (2 ∗ ρ)), p.IsJacFrame J s hh F₀ F

theorem **exists_bes** {ρ : ℕ} (B : BesLev D ρ) {s : k} (hs : s ≠ 0) (FA : Frame k B.A.rank Y) : ∃ F : Frame k B.V.rank (Y ⊕ Fin (2 ∗ ρ + 1)), p.IsBesFrame B s FA F

theorem **exists_unitary_supportedAt_place_of_mul_eq_idem** (a b : Mx D S.rank S.rank) (hba : b ∗ a = p.e • (1 : Mx D S.rank S.rank)) : ∃ g : GA D S, SuppAt p.e S g ∧ (mt S g).map p.π = a.map p.π

theorem **exists_isLoc** (F : Frame k S.rank Y) (gk : GL Y k) : ∃ g, p.IsLoc F g gk

theorem **hyp_flatBF** {ψ : AddChar (AF D.host.F) ℂ} {ρ : ℕ} (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) {s : k} (hs : s ≠ 0) (FA : Frame k B.A.rank Y') (G₀ : Frame k J.W₀.rank Y) : ∃ (ψk : AddChar k Circle) (F₀ : Frame k B.V₀.rank (Y' ⊕ Unit)) (FV : Frame k B.V.rank (Y' ⊕ Fin (2 ∗ ρ + 1))) (FW : Frame k J.W.rank (Y ⊕ Fin (2 ∗ ρ))), p.IsChar ψ ψk ∧ p.IsCoreFrame B s FA F₀ ∧ p.IsBesFrame B s FA FV ∧ p.IsJacFrame J s (p.hz B) G₀ FW ∧ ∃ Tm, p.IsModel ψ FV FW Tm ∧ Tm ≠ 0

theorem **hyp_flatFB** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) {s : k} (hs : s ≠ 0) (G₀ : Frame k J.W₀.rank Y') (FA : Frame k B.A.rank Y) : ∃ (ψk : AddChar k Circle) (FW : Frame k J.W.rank (Y' ⊕ Fin (2 ∗ σ + 1 + 1))) (F₀ : Frame k B.V₀.rank (Y ⊕ Unit)) (FV : Frame k B.V.rank (Y ⊕ Fin (2 ∗ σ + 1))), p.IsChar ψ ψk ∧ p.IsJacFrame J s (−p.xi ^ 2 ∗ p.hz B) G₀ FW ∧ p.IsCoreFrame B (s ∗ p.xi) FA F₀ ∧ p.IsBesFrame B (s ∗ p.xi) FA FV ∧ ∃ Tm, p.IsModel ψ FW FV Tm ∧ Tm ≠ 0

theorem **hyp_contr** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) {ρ σ : ℕ} (B : BesLev D ρ) (B' : BesLev D σ) {s : k} (hs : s ≠ 0) {Z : Type} [Fintype Z] [DecidableEq Z] (FA : Frame k B.A.rank Y') (FA' : Frame k B'.A.rank Z) (F : Frame k S.rank Y) : ∃ (ψk : AddChar k Circle) (F₀ : Frame k B.V₀.rank (Y' ⊕ Unit)) (F₀' : Frame k B'.V₀.rank (Z ⊕ Unit)), p.IsChar ψ ψk ∧ p.IsCoreFrame B s FA F₀ ∧ p.IsCoreFrame B' (s ∗ p.xi) FA' F₀' ∧ (∃ T₁, p.IsModel ψ F₀ F T₁ ∧ T₁ ≠ 0) ∧ ∃ T₂, p.IsModel ψ F F₀' T₂ ∧ T₂ ≠ 0

theorem **exists_haar** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] : ∃ ν : Measure k, ν.IsAddHaarMeasure

theorem **Witness.isModel_smul** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {FS : Frame k S.rank Y'} {FT : Frame k T.rank Y} {Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)} (h : p.IsModel ψ FS FT Tm) (c : ℂ) : p.IsModel ψ FS FT (c • Tm)

theorem **Witness.exists_prod** {X X' U U' : Type} [TopologicalSpace X] [TopologicalSpace X'] (T₁ : SB X →ₗ[ℂ] (U → ℂ)) (T₂ : SB X' →ₗ[ℂ] (U' → ℂ)) (n₁ : T₁ ≠ 0) (n₂ : T₂ ≠ 0) : ∃ Λ : SB (X × X') →ₗ[ℂ] (U → U' → ℂ), Λ ≠ 0 ∧ ∀ (Ξ₁ : SB X) (Ξ₂ : SB X') Ξ, (∀ q, Ξ.1 q = Ξ₁.1 q.1 ∗ Ξ₂.1 q.2) → ∀ u v, Λ Ξ u v = T₁ Ξ₁ u ∗ T₂ Ξ₂ v

theorem **Witness.exists_isBiModel** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) {A S A' : HSp D} {n n' n'' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (FA : Frame k A.rank n') (FS : Frame k S.rank n) (FA' : Frame k A'.rank n'') : ∃ Λ, p.IsBiModel ψ FA FS FA' Λ ∧ Λ ≠ 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.FinAfLocalIdele

theorem **exists_vG_over** {X : Fields} {Pr : Primes X} (hX : X.Ok) (hPr : Pr.Ok) (k : Fin 2) (v : HeightOneSpectrum (𝒪 X.F)) (h : ((Pr.g k : ℕ) : 𝒪 X.F) ∈ v.asIdeal) : ∃ w ∈ Pr.vG k, w.under (𝒪 X.F) = v

theorem **piBar_std** (k : Fin 2) : ∃ y : 𝒪 B.host.K, ((y : B.host.K) : B.host.E) = (B.fields.conjO (X.piG k) : B.host.E) ∧ y ∈ (I.C.P.pb k).asIdeal

theorem **om_std** (G : GLoc (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) (I.rec7 l Z).host.g) (hG : I.IdG l Z G) (k : Fin 2) (v : HeightOneSpectrum (𝒪 B.host.F)) (h : (B.host.g k : 𝒪 B.host.F) ∈ v.asIdeal) (a n : ℕ) (hn : n.Coprime (B.host.g k)) (ha : a < B.R.m) : G.chiB k v h a (n : kv B.host.F v) = ((I.C.W.om k (ZMod.unitOfCoprime n hn) ^ (2 ∗ (a + 1)) : ℂˣ) : ℂ)

theorem **idem_comp** {e : AfE F E} {P : HeightOneSpectrum (𝒪 E)} (he : e ∗ e = e) (h : ∀ x : 𝒪 E, x ≠ 0 → (x ∈ P.asIdeal ↔ e ∗ (((x : E)⁻¹) ⊗ₜ[F] (1 : FiniteAdeleRing (𝒪 F) F)) ∉ RInt F E)) : afIso F E e P = 1 ∧ ∀ Q, Q ≠ P → afIso F E e Q = 0

theorem **finAf_localIdele** {e : AfE F E} {P : HeightOneSpectrum (𝒪 E)} (he : e ∗ e = e) (h : ∀ x : 𝒪 E, x ≠ 0 → (x ∈ P.asIdeal ↔ e ∗ (((x : E)⁻¹) ⊗ₜ[F] (1 : FiniteAdeleRing (𝒪 F) F)) ∉ RInt F E)) (k : Eˣ) : (finAf F E (localIdele E P (atFin P k)) : AfE F E) = 1 + e ∗ ((k : E) ⊗ₜ[F] (1 : FiniteAdeleRing (𝒪 F) F) − 1)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.FiniteDimensionalMram

theorem **rank_map_field** {K L : Type} [Field K] [Field L] (f : K →+* L) {m n : Type} [Fintype m] [Fintype n] (A : Matrix m n K) : (A.map f).rank = A.rank

theorem **rank_toMatrix** {K V W ι κ : Type} [Field K] [AddCommGroup V] [Module K V] [AddCommGroup W] [Module K W] [Fintype ι] [DecidableEq ι] [Fintype κ] (b : Basis ι K V) (c : Basis κ K W) (φ : V →ₗ[K] W) : (LinearMap.toMatrix b c φ).rank = finrank K (LinearMap.range φ)

theorem **colTo_succ** (j : ℕ) : colTo C (j + 1) = (C.Klin j).comp (colTo C j)

theorem **colTo_eq** (j : ℕ) (φ : C.cls 0 → C.Y 0 → ℂ) : colTo C j φ = Formalism.ThetaLiftIndependence.chainOpM C.wt C.Kmap j φ

theorem **PhiC_apply** (φ : A.Mram) (F : P.Mt) : PhiC C A P φ F = P.PtilCol (φ : C.cls 0 → C.Y 0 → ℂ) F

theorem **finiteDimensional_Mram** : FiniteDimensional ℂ A.Mram

theorem **rkC_add_ker** : rkC C A P + finrank ℂ (LinearMap.ker (PhiC C A P)) = A.du

theorem **rkC_le** : rkC C A P ≤ A.du

theorem **rkc_iff_ker** : RKC C A P ↔ 2 ∗ finrank ℂ (LinearMap.ker (PhiC C A P)) < A.du

theorem **dualH_muMax** {K : Type} [Field K] [NumberField K] [StarRing K] (A : HLat K) : (dualH A).muMax = A.muMaxDual

theorem **isPhi_Phi** (h : PhiRat D Am u) : IsPhi D Am u (Phi D Am u)

theorem **isPhi_unique** {Φ Ψ} (h : IsPhi D Am u Φ) (h' : IsPhi D Am u Ψ) : Φ = Ψ

theorem **IsPhi.eq_Phi** {Φ} (h : IsPhi D Am u Φ) : Φ = Phi D Am u

theorem **phiInt_iff_exists** (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PhiInt D Am u N ↔ ∃ Φ, IsPhi D Am u Φ ∧ ∀ x ∈ N, Φ x ∈ (MtDual D Am).L

theorem **phiInt_iff** (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PhiInt D Am u N ↔ PhiRat D Am u ∧ ∀ x ∈ N, ∀ z ∈ (Mt D Am).L, ∃ a : 𝒪 D.what₂.LL, D.what₂.embLL (a : D.what₂.LL) = D.what₂.perCol u D.what₂.mu0 (D.jN u x) (Am.jAmb D.what₂.mu0 z)

theorem **htPhiLe_iff_exists** (h : ℝ) : HtPhiLe D Am u h ↔ ∃ Φ, IsPhi D Am u Φ ∧ (Φ ≠ 0 → Formalism.HN.htMap (D.div.Nu u).b (D.div.Nu u).X (MtDual D Am).b (MtDual D Am).X Φ ≤ h)

theorem **finrank_Nu** : finrank D.what₂.LL (D.div.Nu u).V = dU D u

theorem **toMatrix_PhiC** {Φ} (h : IsPhi D Am u Φ) : LinearMap.toMatrix (basN D u) (basM D Am u).dualBasis (PhiC (D.mem u).C (D.mem u).A (D.mem u).P) = (LinearMap.toMatrix (D.div.Nu u).b (MtDual D Am).b Φ).map D.what₂.embLL

theorem **rkL_eq** (h : PhiRat D Am u) : rkL D Am u = rkC (D.mem u).C (D.mem u).A (D.mem u).P

theorem **rkl_iff** : RKL D Am u ↔ PhiRat D Am u ∧ RK D u

theorem **RKL.phi_ne_zero** (h : RKL D Am u) : Phi D Am u ≠ 0

theorem **RKL.htPhi_le** (h : RKL D Am u) {t : ℝ} (ht : HtPhiLe D Am u t) : htPhi D Am u ≤ t

theorem **PhiInt.mono** {N N' : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V} (h : PhiInt D Am u N) (hle : N' ≤ N) : PhiInt D Am u N'

Development.AbelianVarietyTheory.Definitions.LatticeFrames.FiniteOfPlaces

theorem **exists_one_place** (D : D7) (Z : Finset (Pl D)) : ∃ (v : Pl D) (p : SplitPt D (Fv D.host.F v)), v ∉ Z ∧ Places (fun _ : Unit => p) [()] (· = v)

theorem **finite_of_places** {p : ι → SplitPt D k} {I : List ι} {P : Pl D → Prop} (h : Places p I P) : {x | P x}.Finite

theorem **not_places_of_eq** {p : ι → SplitPt D k} {I : List ι} {P : Pl D → Prop} {i j : ι} (hij : i ≠ j) (hv : (p i).v = (p j).v) : ¬ Places p I P

theorem **places_empty** (p : Empty → SplitPt D k) : Places p [] fun _ => False

theorem **reps_nil** (Ψ : ι → Matrix n n' k → ℂ) (P : Pl D → Prop) : Reps ψ α β LS LT Ψ [] P (Finsupp.single 0 1)

theorem **vac_zero** (P : Pl D → Prop) : Vac ψ S T LS LT P 0

theorem **sb_of_reps** {Ψ : ι → Matrix n n' k → ℂ} {i : ι} {I : List ι} {P : Pl D → Prop} {c : WO S T →₀ ℂ} (h : Reps ψ α β LS LT Ψ (i :: I) P c) : Ψ i ∈ SB (Matrix n n' k) ∧ IsSelfDualC S LS ∧ IsSelfDualC T LT

theorem **symplOH_kap** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ ∗ (L.gram ∗ τ) = (-(D.host.xi0 ∗ star D.host.xi0)) • L''.gram) (S : HSp D) (x y : WO S L'') : symplOH L S (kap τ S x) (kap τ S y) = symplOH S L'' x y

theorem **kap_add** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (S : HSp D) (x y : WO S L'') : kap τ S (x + y) = kap τ S x + kap τ S y

theorem **lastBi_heis** (ψ : AddChar (AF D.host.F) ℂ) {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ ∗ (L.gram ∗ τ) = (-(D.host.xi0 ∗ star D.host.xi0)) • L''.gram) (a : WO A S) (x : WO S X) (Ξ Φ₂) (u v) : heisO ψ A S a (fun u' => heisO ψ S X x (lastBi o o'' τ Ξ Φ₂ u') v) u = lastBi o o'' τ (heisO ψ V₀ S ((o.resS S).symm (a, kap τ S (o''.resT S x).2)) Ξ) (heisO ψ S T (o''.resT S x).1 Φ₂) u v

theorem **smallPair_zero** {V₀ A L S T : HSp D} (o : OrthDec V₀ A L) (om₁ : Om V₀ S) (omT : Om S T) (vA : Vol A) (vS : Vol S) (vT : Vol T) (Ξ Φ₂ F) : smallPair o om₁ omT vA vS vT 0 Ξ Φ₂ F = 0

theorem **smallPair_eq_Q** {ψ : AddChar (AF D.host.F) ℂ} {T : HSp D} {B : BesLev D 0} (t : Last ψ T B) (o : OrthDec B.V B.A B.L) (ho : ∀ a, o.emb (a, 1) = embA B.d B.o a) (φ Φ₁ Φ₂ F) : smallPair o t.s.om t.om B.vA t.s.vW t.vT φ Φ₁ Φ₂ F = t.Q φ Φ₁ Φ₂ F

Development.AbelianVarietyTheory.Definitions.LatticeFrames.FinrankLeRankOnAdd

theorem **corankP_le_hsDev** (D : D7) (u : D.what₂.Word) : (corankP D u : ℝ) ≤ GlobalTheory.Core.Rk.hsDev D u

theorem **finrank_le_rankOn_add** (D : D7) (u : D.what₂.Word) (M : Submodule ℂ (D.s.Cu u)) : Module.finrank ℂ M ≤ rankOn D u M + corankP D u

theorem **shareAt_iff** {D : D7} {u : D.what₂.Word} {e : ℝ} : ShareAt D u e ↔ GlobalTheory.Core.Rk.DiagS D u e

theorem **fullAt_of** {D : D7} {u : D.what₂.Word} {L C c : ℝ} (hL : 0 ≤ L) (hC : 0 ≤ C) (hc : c ≤ 1) (h : SDiagAt D u (gamL D L C) c) : FullAt D u L C c

theorem **ShareAt.pu_ne_zero** {D : D7} {u : D.what₂.Word} {e : ℝ} (h : ShareAt D u e) (he : e < 1) (hd : 0 < D.duS u) : D.s.Pu u ≠ 0

theorem **shareL_lt_one** {D : D7} {L C' Cl Cg cb Cu cT CT C c : ℝ} (h : GlobalTheory.Core.Deg.HypL D L C' Cl Cg cb Cu cT CT) (h8 : 8 ≤ C') (hC : 1 ≤ C) (hc : 1 / 2 ≤ c) : shareL D L C c < 1

Development.AbelianVarietyTheory.Definitions.LatticeFrames.GammaRangeOfSizeHypotheses

theorem **pow_five_mul_f_pos** (D : D7) : (0 : ℝ) < (D.R.m : ℝ) ^ 5 ∗ D.R.f

theorem **gamRange_iff** {D : D7} (large₂ : SizeData₂ D) : GamRange D large₂.sc.L ↔ 1 ≤ large₂.CGam ∧ 8 ∗ large₂.CGam ∗ ((D.R.m : ℝ) ^ 5 ∗ D.R.f ∗ large₂.sc.L) ≤ D.R.k

theorem **gammaRange_of_sizeHypotheses** {D : D7} {large₂ : SizeData₂ D} {C' Cl Cg cb Cu cT CT : ℝ} (h1 : 1 ≤ large₂.CGam) (h : StandingHypotheses D large₂ C' Cl Cg cb Cu cT CT) : GamRange D large₂.sc.L

theorem **dataNW_of_window** {D : D7} {C0 : ℝ} (h3 : 1 ≤ C0) (hd : StandingNumericalData D C0) : GlobalTheory.Core.Deg.DataNW D

theorem **hd_of_window** {D : D7} {large₂ : SizeData₂ D} {C' Cl Cg cb Cu cT CT C0 : ℝ} (h : Cnt.Hyp0 D large₂ C' Cl Cg cb Cu cT CT C0) (tf : TFrameAt D) : HD D large₂.sc.L C' Cl Cg cb Cu cT CT

theorem **sDiagG_of_sDiag** {c : ℝ} (h : SDiag c) : SDiagG c

theorem **sDiagAtLed_of_sDiagG** {c : ℝ} (h : SDiagG c) : SDiagAtLed c

theorem **sDiagOn_of_sDiagAtLed** {c : ℝ} (h : SDiagAtLed c) (S : Set ℝ) : SDiagOn S c

theorem **sDiagOn_of_sDiag** {c : ℝ} (h : SDiag c) (S : Set ℝ) : SDiagOn S c

theorem **SDiagOn.mono** {S S' : Set ℝ} {c : ℝ} (hS : S ⊆ S') (h : SDiagOn S' c) : SDiagOn S c

theorem **OneOn.mono** {S S' : Set ℝ} {gen : Bool} (hS : S ⊆ S') (h : OneOn S' gen) : OneOn S gen

theorem **OneAllOn.mono** {S S' : Set ℝ} (hS : S ⊆ S') (h : OneAllOn S') : OneAllOn S

theorem **oneAllOn_of_oneAll** (h : OneAll) (S : Set ℝ) : OneAllOn S

theorem **oneOn_of_oneLed** {gen : Bool} (h : OneLed gen) (S : Set ℝ) : OneOn S gen

Development.AbelianVarietyTheory.Definitions.LatticeFrames.HeckeFormNSOfHeckeForm

theorem **heckeFormNS_of_heckeForm** (h : HeckeForm D) : HeckeFormNS D

Development.AbelianVarietyTheory.Definitions.LatticeFrames.HtPsiLeIffExists

theorem **isPsi_Psi** (h : PsiRat D u) : IsPsi D u (Psi D u)

theorem **IsPsi.unique** {Ψ Ψ'} (h : IsPsi D u Ψ) (h' : IsPsi D u Ψ') : Ψ = Ψ'

theorem **IsPsi.eq_Psi** {Ψ} (h : IsPsi D u Ψ) : Ψ = Psi D u

theorem **psiInt_iff_exists** (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PsiInt D u N ↔ ∃ Ψ, IsPsi D u Ψ ∧ ∀ x ∈ N, Ψ x ∈ (Pt D u).L

theorem **psiInt_iff** (N : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V) : PsiInt D u N ↔ PsiRat D u ∧ ∀ x ∈ N, ∃ y ∈ (Pt D u).L, (D.fm u).ptEmb y = PsiC (D.mem u).C (D.mem u).A (D.mem u).P (D.eMu u (D.jN u x))

theorem **PsiInt.mono** {N N' : Submodule (𝒪 D.what₂.LL) (D.div.Nu u).V} (h : PsiInt D u N) (hle : N' ≤ N) : PsiInt D u N'

theorem **htPsiLe_iff_exists** (h : ℝ) : HtPsiLe D u h ↔ ∃ Ψ, IsPsi D u Ψ ∧ (Ψ ≠ 0 → Formalism.HN.htMap (D.div.Nu u).b (D.div.Nu u).X (Pt D u).b (Pt D u).X Ψ ≤ h)

theorem **RKPsi.psi_ne_zero** (h : RKPsi D u) : Psi D u ≠ 0

theorem **RKPsi.psiRat** (h : RKPsi D u) : PsiRat D u

theorem **RKPsi.htPsi_le** (h : RKPsi D u) {t : ℝ} (ht : HtPsiLe D u t) : htPsi D u ≤ t

theorem **IsPer.unique** {e e'} (h : IsPer D u Am e) (h' : IsPer D u Am e') : e = e'

theorem **IsPer.isPhi_comp** {e} (he : IsPer D u Am e) {Ψ} (h : IsPsi D u Ψ) : IsPhi D Am u (e.toLinearMap ∘ₗ Ψ)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.HypOver

theorem **rk_share** {D : D7} {u : D.what₂.Word} (h : RK D u) {ρ : ℝ} (hρ : ρ ≤ 1 / 2) : RKShare ρ D u

theorem **someAt_mono** (F : Pts) {Q Q' : ℝ → ∀ D : D7, D.what₂.Word → Prop} (x₁ : ℝ) (hq : ∀ x, x₁ ≤ x → ∀ i u, Q x (F.D x i) u → Q' x (F.D x i) u) (h : F.SomeAt Q) : F.SomeAt Q'

theorem **win_of_room** (F : Pts) (ε : ℝ) (h : F.RKRoom ε) : F.RKWin

theorem **weak_of_room** (F : Pts) {ε : ℝ} (hε : 0 < ε) (h : F.RKRoom ε) : F.RKWeak ε

theorem **xu_pos** (D : D7) (u : D.what₂.Word) : 0 < (D.mem u).A.Xu

theorem **fk_pos** (R : Ranks) : (0 : ℝ) < (R.f : ℝ) * R.k

theorem **log_tstar** (D : D7) (u : D.what₂.Word) : Real.log (D.mem u).A.Tstar = Real.log (D.mem u).A.Xu - 2 / ((D.R.f : ℝ) * D.R.k) * Real.log (D.mem u).A.GamP

theorem **log_T_lt** (D : D7) (u : D.what₂.Word) : Real.log ((D.mem u).A.T : ℝ) < Real.log (D.mem u).A.Xu + 1 / ((D.R.f : ℝ) * D.R.k)

theorem **le_log_alt** {D : D7} (L : SizeData₂ D) (u : D.what₂.Word) : Real.log (D.mem u).A.Xu - 2 * L.CGam * (D.R.m : ℝ) ^ 5 / D.R.k * L.sc.L ≤ Real.log ((D.mem u).A.T : ℝ)

theorem **sameTops_iff** {D D' : D7} (L : SizeData₂ D) (L' : SizeData₂ D') (hk : D.R.k = D'.R.k) : SameTops D D' ↔ ∀ (t : D.host.TT) (t' : D'.host.TT), (t : D.R.A).val = (t' : D'.R.A).val → L.aT t = L'.aT t' ∧ L.bT t = L'.bT t'

theorem **win_of_all** {ε : ℝ} (F : FamCore ε) (h : (ptsOf F).RKAll) : (ptsOf F).RKWin

theorem **exists_of_win** {ε : ℝ} (F : FamCore ε) (h : (ptsOf F).RKWin) : ∃ x₀ : ℝ, ∀ x, x₀ ≤ x → x ∈ F.Win → ∃ (i : F.Ix x) (u : (F.D x i).what₂.Word), RK (F.D x i) u

theorem **tH** (t : R0.A) : THyp (Ht t)

theorem **colTo_one** (j : ℕ) : colTo (chainN hc (pkg N)) j (fun _ _ => 1) = fun _ _ => ((N : ℂ) + 1) ^ j

theorem **rk_le_one** : rkC (chainN hc (pkg N)) (atTN hc (pkg N) T Dst hD 1 one_pos 2 one_lt_two) (periodN hc (pkg N) (En 1) (perS N s)) ≤ 1

theorem **rk_zero** : rkC (chainN hc (pkg N)) (atTN hc (pkg N) T Dst hD 1 one_pos 2 one_lt_two) (periodN hc (pkg N) (En 1) (perS N 0)) = 0

theorem **rk_one** : rkC (chainN hc (pkg N)) (atTN hc (pkg N) T Dst hD 1 one_pos 2 one_lt_two) (periodN hc (pkg N) (En 1) (perS N 1)) = 1

theorem **baseDegree_ne_two** : T0 ≠ 2

theorem **recd_dU** (t c N s u) : dU (recd t c N s) u = N + 1

theorem **columnRank_eq_one_of_rankOne** (t c N u) : rkC ((recd t c N 1).mem u).C ((recd t c N 1).mem u).A ((recd t c N 1).mem u).P = 1

theorem **columnRank_eq_zero_of_rankZero** (t c N u) : rkC ((recd t c N 0).mem u).C ((recd t c N 0).mem u).A ((recd t c N 0).mem u).P = 0

theorem **rankCondition_rankOne_iff_eq_zero** (t c N u) : RK (recd t c N 1) u ↔ N = 0

theorem **not_rankCondition_rankZero** (t c N u) : ¬ RK (recd t c N 0) u

theorem **recd_share** (t c N u) (ρ : ℝ) : RKShare ρ (recd t c N 1) u ↔ ρ * ((N : ℝ) + 1) ≤ 1

theorem **recd_sameTops** (t c N s t' c' N' s') : SameTops (recd t c N s) (recd t' c' N' s') ↔ c = c'

theorem **val_cast** (u₀ : ℕ) (h : u₀ < R0.n) : (R0.ofNat u₀).val = u₀

theorem **tIn_ev** (D : R0.A → D7) (hm : ∀ t, (D t).R.m = 2) (v : ∀ t, (D t).what₂.Word) {ε : ℝ} (hε : 0 ≤ ε) : ∃ CR CT : ℝ, ∀ x, 1 ≤ x → ∀ t, TIn (D t) (v t) (CR * x ^ ε) (CT * (D t).R.m * x ^ ε)

theorem **sameWord_symm** {D D' : D7} {u : D.what₂.Word} {u' : D'.what₂.Word} (h : SameWord D u D' u') : SameWord D' u' D u

theorem **sameWord_trans** {D D' D'' : D7} {u : D.what₂.Word} {u' : D'.what₂.Word} {u'' : D''.what₂.Word} (h : SameWord D u D' u') (h' : SameWord D' u' D'' u'') : SameWord D u D'' u''

theorem **sameTops_symm** {D D' : D7} (h : SameTops D D') : SameTops D' D

theorem **sameTops_trans** {D D' D'' : D7} (hh : SameHost D D') (h : SameTops D D') (h' : SameTops D' D'') : SameTops D D''

theorem **sameWord_self_iff** (D : D7) (u u' : D.what₂.Word) : SameWord D u D u' ↔ D.hcW u = D.hcW u'

theorem **over_mono_eps** {F : Pts} {Tr : ℝ → D7 → Prop} {ε ε' : ℝ} (h : F.RKRoomOver Tr ε) (hε : ε ≤ ε') : F.RKRoomOver Tr ε'

theorem **over_threshold** {F : Pts} {Tr : ℝ → D7 → Prop} {ε : ℝ} (h : F.RKRoomOver Tr ε) : ∃ CR CT : ℝ, F.SomeAt fun x D u => Real.log (D.mem u).A.Xu ≤ (CR + CT * D.R.m) * x ^ ε

theorem **log_T_le** (D : D7) (u : D.what₂.Word) : Real.log ((D.mem u).A.T : ℝ) ≤ Real.log (D.mem u).A.Xu + 1

theorem **exists_big** : ∃ q : ℝ → LvP, ∀ x, Real.exp x < (q x).T

theorem **ev_lt** (a b : ℝ) {δ : ℝ} (hδ : δ < 1) : ∃ x₁ : ℝ, ∀ x, x₁ ≤ x → a * x ^ δ + b < x

theorem **lower_end** (p : LvP) (q : ℝ → LvP) (hq : ∀ x, Real.exp x < (q x).T) {ε : ℝ} (hε : ε < 1 / 52) : ¬ (famQ q).RKWeakOver (cls p 0 1) ε

theorem **win_not_room** (p : LvP) {ε : ℝ} (h0 : 0 < ε) (hε : ε < 1 / 52) : ∃ F : Pts, F.RKAll ∧ F.RKWin ∧ ¬ F.RKRoom ε ∧ ¬ F.RKRoomOver (cls p 0 1) ε

theorem **one_translate** (p p' : LvP) (ε : ℝ) : ¬ (fam p (fun _ => 0) 1).RKWeakOver (fun _ D => D = recd p' 0 0 0 1) ε

theorem **two_weights** (p p' : LvP) {ε : ℝ} (hε : 0 ≤ ε) : ¬ (famTwo p).RKRoomOver (cls p' 0 1) ε ∧ (famTwo p).RKRoomOver (fun x D => cls p' 0 1 x D ∨ cls p' 1 1 x D) ε

theorem **half_over** (p p' : LvP) {ε : ℝ} (hε : 0 < ε) : (fam p (fun _ => 0) 0).RKWeakOver (clsN p' fun _ => 1) ε ∧ ¬ (fam p (fun _ => 0) 0).RKRoomOver (clsN p' fun _ => 1) ε

theorem **thin_over** (p p' : LvP) {ε : ℝ} (h52 : ε < 1 / 52) : ¬ (fam p (fun _ => 0) 1).RKWeakOver (clsN p' fun x => ⌈x⌉₊) ε

theorem **free_class** (p : LvP) {ε : ℝ} (hε : 0 ≤ ε) : (fam p (fun _ => 0) 0).RKRoomOver (fun _ _ => True) ε

theorem **win_not_all** (p : LvP) : (famSome p).RKWin ∧ ¬ (famSome p).RKAll

theorem **hyp_over** (p p' : LvP) : HypAll (fun _ => Pts) (fun _ F => F = fam p (fun _ => 0) 0) (fun _ F => F) (fun ε F => F.RKRoomOver (cls p' 0 1) ε) ∧ ¬ HypEx (fun _ => Pts) (fun _ F => F = fam p (fun _ => 0) 1) (fun _ F => F) (fun ε F => F.RKRoomOver (cls p' 0 0) ε)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.InfiniteQAtOfIs

theorem **blk_mul** (g h : GL (n ⊕ Fin 0) k) : (g * h).val.toBlocks₁₁ = g.val.toBlocks₁₁ * h.val.toBlocks₁₁

theorem **box_stable** {g : GL (n ⊕ Fin 0) k} (hg : g ∈ KO k _) (a : kˣ) {y : n → k} (hy : y ∈ box n a) : g.val.toBlocks₁₁ *ᵥ y ∈ box n a

theorem **sameOrb_mk** {J : Subgroup (GL (n ⊕ Fin 0) k)} (hJ : J ≤ KO k _) (a : kˣ) (y : n → k) (yb : (n → k) ⧸ box n a) : SameOrb J (box n a) (Submodule.Quotient.mk y) yb ↔ ∃ g ∈ J, Submodule.Quotient.mk (g.val.toBlocks₁₁ *ᵥ y) = yb

theorem **sameOrb_equiv** {J : Subgroup (GL (n ⊕ Fin 0) k)} (hJ : J ≤ KO k _) (a : kˣ) : Equivalence (SameOrb J (box n a))

theorem **card_orb_ne_zero** {J : Subgroup (GL (n ⊕ Fin 0) k)} (hJ : J ≤ KO k _) (hc : IsCompact (J : Set (GL (n ⊕ Fin 0) k))) (a : kˣ) (x : (n → k) ⧸ box n a) : Nat.card {yb | SameOrb J (box n a) x yb} ≠ 0

theorem **orb_zero** {J : Subgroup (GL (n ⊕ Fin 0) k)} (hJ : J ≤ KO k _) (a : kˣ) : {yb | SameOrb J (box n a) (Submodule.Quotient.mk 0) yb} = {Submodule.Quotient.mk 0}

theorem **prX_KO** {c : ℕ} : prX (Cgrp (KO k (n ⊕ Fin c)) 1) = KO k n

theorem **ind_apply** {a b : Type} (x : Matrix a b k) : ind k a b x = if IntM x then 1 else 0

theorem **ind_SB** (a b : Type) [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] : ind k a b ∈ SB (Matrix a b k)

theorem **sph_reg** (μ : ℕ → kˣ →* ℂˣ) : (sphChain X Z s μ).Regular

theorem **sph_psi'** (μ : ℕ → kˣ →* ℂˣ) (x₀ y) : (sphChain X Z s μ).Psi 1 x₀ y = if IntM x₀ ∧ ∀ i, y i ∈ 𝒪[k] then 1 else 0

theorem **sph_psi** (μ : ℕ → kˣ →* ℂˣ) (x₀ y) : (sphChain X Z s μ).Psi 1 x₀ y ≠ 0 ↔ IntM x₀ ∧ ∀ i, y i ∈ 𝒪[k]

theorem **mem_box_one** {y : n → k} : y ∈ box n (1 : kˣ) ↔ ∀ i, y i ∈ 𝒪[k]

theorem **sph_adapt** (μ : ℕ → kˣ →* ℂˣ) : (sphChain X Z s μ).Adapted (fun _ _ _ => {1}) (box _ 1) (box _ 1)

theorem **sph_pen** (μ : ℕ → kˣ →* ℂˣ) : (sphChain X Z s μ).xiPen (box _ 1) 1 (Submodule.Quotient.mk 0) = ind k _ _

theorem **sph_top** (μ : ℕ → kˣ →* ℂˣ) : (sphChain X Z s μ).xiTop (box _ 1) ψ ν (-1) (Submodule.Quotient.mk 0) = ind k _ _

theorem **sph_reps** (μ : ℕ → kˣ →* ℂˣ) : (sphChain X Z s μ).IsRepSystem (fun _ _ _ => {1}) ψ ν

theorem **isOneComb_sph** {c P : ℕ} (h1 : 1 ≤ P) (h2 : P ≤ c) : IsOneComb ψ ν P (sph k (n ⊕ Fin c)) 1 (Finsupp.single 1 1)

theorem **ae_int** (a : D.host.E) : ∀ᶠ x : GlobalTheory.FinPl.Pl D in Filter.cofinite, ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.v = x → (ofSplit q).pr a ∈ 𝒪[k]

theorem **exists_inv** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (f : D.host.E →+* k) {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (M : Matrix (Fin N) Y D.host.E) (F : GlobalTheory.LocMod.Frame k N Y) (h : F.β = M.map f) : ∃ M' : Matrix Y (Fin N) D.host.E, M * M' = 1

theorem **ae_frame** (S : HSp D) (L : GlobalTheory.FinPl.Lat S) (hL : Submodule.IsLattice D.host.E L) {Y : Type} [Fintype Y] [DecidableEq Y] (M : Matrix (Fin S.rank) Y D.host.E) : ∀ᶠ x : GlobalTheory.FinPl.Pl D in Filter.cofinite, ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.v = x → ∀ F : GlobalTheory.LocMod.Frame k S.rank Y, F.β = M.map (ofSplit q).pr → ∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((F.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S L (· = q.v)

theorem **rat_bes** {ρ : ℕ} (B : BesLev D ρ) (n c : ℕ) : ∃ M : Matrix (Fin B.V.rank) (Fin B.A.rank ⊕ Fin c) D.host.E, ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (F : GlobalTheory.LocMod.Frame k B.V.rank (Fin B.A.rank ⊕ Fin c)), p.IsBesFrame B (p.xi ^ n) (Frame.ofGL 1) F → F.β = M.map p.pr

theorem **rat_jac** {ρ : ℕ} (B : BesLev D ρ) (J : JacLev D ρ) (n c : ℕ) : ∃ M : Matrix (Fin J.W.rank) (Fin J.W₀.rank ⊕ Fin c) D.host.E, ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (F : GlobalTheory.LocMod.Frame k J.W.rank (Fin J.W₀.rank ⊕ Fin c)), p.IsJacFrame J (p.xi ^ n) (p.hz B) (Frame.ofGL 1) F → F.β = M.map p.pr

theorem **idx_ne_zero** (Q : QAt v χ) (j : ℕ) (h : j ≤ D.R.m) : Q.idx j ≠ 0

theorem **wt_ne_zero** (Q : QAt v χ) : (∀ y, Q.wt (Q.s + 1) y ≠ 0) ∧ ∀ u, Q.wt (Q.s + 1 + 1) u ≠ 0

theorem **pin_ne_zero** {t : Tup c} {w : VTup t} (h : QPin v w) : (∀ j, j + 2 ≤ D.R.m → w.cQ j ≠ 0) ∧ ∀ n hn β, w.εT n hn β ≠ 0

theorem **exists_last** (p : Place D k) {X T L : HSp D} (o : OrthDec X T L) (hL : L.rank = 1) {Z : Type} [Fintype Z] [DecidableEq Z] (FT : Frame k T.rank Z) {a : k} (ha : a ≠ 0) : ∃ FM : Frame k X.rank (Z ⊕ Unit), FM.β = fromCols (o.iA.map p.pr * FT.β) (Matrix.of fun r _ => (o.iB.map p.pr *ᵥ fun _ => a) r)

theorem **exists_qAt** (v : VCom c) (χ : ℕ → (AE D)ˣ →* Circle) (q : GlobalTheory.FinPl.SplitPt D k) (unit : q.UnitAA) {ψk : AddChar k Circle} (hk : (ofSplit q).IsChar c.ψ ψk) (hc : HasConductor ψk 1) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, j < D.R.m → (ofSplit q).IsComp (χ j)⁻¹ (μ (j + 1)) ∧ IsUnram (μ (j + 1))) (hL : c.Lz.rank = 1) (hτ : ∀ j h, v.τ j h ≠ 0) (lB : ∀ j ρ h (F : GlobalTheory.LocMod.Frame k (c.V j).rank (Fin (c.X j).rank ⊕ Fin (2 * ρ + 1))), (ofSplit q).IsBesFrame (c.B j ρ h) ((ofSplit q).xi ^ (j / 2)) (Frame.ofGL 1) F → Spans q F (v.N j)) (lJ : ∀ j ρ h (F : GlobalTheory.LocMod.Frame k (c.V (j + 1)).rank (Fin (c.X (j + 1)).rank ⊕ Fin (2 * ρ))), (ofSplit q).IsJacFrame (c.J j ρ h) ((ofSplit q).xi ^ (j / 2)) ((ofSplit q).hz (c.B j ρ h)) (Frame.ofGL 1) F → Spans q F (v.N (j + 1))) (latT : Spans q (Frame.ofGL 1) v.NT) : ∃ Q : QAt v χ, Q.q.v = q.v ∧ IsSphAt Q

theorem **qpin_of_sph** (Q : QAt v χ) (hQ : IsSphAt Q) {t : Tup c} [∀ j, Unique (t.Ch j)] (w : VTup t) (kom : ∀ j, j < D.R.m → IsWeilDatumO c.ψ _ _ (χ j) (χ j) (t.kom j)) (mid : ∀ j σ h, ∃ χz : (AE D)ˣ →* Circle, (t.sBF j (σ + 1) h).χV = χ j * χz ∧ (t.sBF j (σ + 1) h).χW = χ j ∧ (t.sFB j σ h).χV = χ (j + 1) ∧ (t.sFB j σ h).χW = χ (j + 1) * χz) (lastc : ∀ j h, (t.sBF j 0 h).χW = χ j) : QPin v { w with Q

theorem **sum_orbits** (Q : QAt v χ) (f : ((Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂) → ℂ) (h1 : ∀ x y, SameOrb (Q.T.lev (Q.s + 1) 0 rfl).J (box _ Q.c₂) x y → f x = f y) (h2 : ∀ x, f x ≠ 0 → x ∈ Q.T.coupIdx Q.A (box _ Q.c₁) (box _ Q.c₂)) : (Q.idx (Q.s + 1) : ℂ) * ∑ᶠ x, f x = ∑ y, Q.wt (Q.s + 1) y * f (Q.cos y)

theorem **infinite_qAt** (hψ : IsStdChar D c.ψ (-1)) (v : VCom c) (χ : ℕ → (AE D)ˣ →* Circle) (hχ : ∀ j, j < D.R.m → Continuous (χ j)) (hL : c.Lz.rank = 1) (hτ : ∀ j h, v.τ j h ≠ 0) (hN : ∀ j, j < D.R.m → Submodule.IsLattice D.host.E (v.N j)) (hT : Submodule.IsLattice D.host.E v.NT) : {x | ∃ Q : QAt v χ, Q.q.v = x ∧ IsSphAt Q}.Infinite

theorem **infinite_qAt_of_is** (hc : c.Is) (hv : v.Is) {t : Tup c} (hW : WeilOK (t

theorem **infinite_qpin** (hc : c.Is) (hv : v.Is) {t : Tup c} (hW : WeilOK (t

theorem **exists_tup** (hψ : IsStdChar D c.ψ (-1)) (t₀ : Tup c) (hp : ∀ j, j ≤ D.R.m → (c.X j).rank % 2 = (c.X 0).rank % 2) : ∃ (χ : ℕ → (AE D)ˣ →* Circle) (t : Tup c), (∀ j, Continuous (χ j)) ∧ (∀ j, Nonempty (Unique (t.Ch j))) ∧ (∀ j, j < D.R.m → IsWeilDatumO c.ψ _ _ (χ j) (χ j) (t.kom j)) ∧ (∀ j σ h, ∃ χz : (AE D)ˣ →* Circle, (t.sBF j (σ + 1) h).χV = χ j * χz ∧ (t.sBF j (σ + 1) h).χW = χ j ∧ (t.sFB j σ h).χV = χ (j + 1) ∧ (t.sFB j σ h).χW = χ (j + 1) * χz) ∧ ∀ j h, (t.sBF j 0 h).χW = χ j

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IntOfKeeps

theorem **nat_mul_int** {x : Pl D} (m : ℕ) {b : AE D} (hb : b ∈ intAt D (· = x)) : (m : AE D) * b ∈ intAt D (· = x)

theorem **int_of_keeps** {S : HSp D} {L : Lat S} (hL : IsSelfDualC S L) (hr : 0 < S.rank) {x : Pl D} {t : AE D} (h : ∀ y ∈ latV S L (· = x), t • y ∈ latV S L (· = x)) : t * idem D (· = x) ∈ intAt D (· = x)

theorem **isU_of_norm** {x : Pl D} {t : AE D} (ht : t * idem D (· = x) ∈ intAt D (· = x)) (hn : IsU x (t * cA D t)) : IsU x t

theorem **norm_tau** {E : Type} [Field E] [StarRing E] {m n : ℕ} {G : Matrix (Fin m) (Fin m) E} {G' : Matrix (Fin n) (Fin n) E} {τ : Matrix (Fin m) (Fin n) E} {t lam : E} {ζ : Fin m → E} {ζ₀ : Fin n → E} (h : τᴴ * (G * τ) = t • G') (hτ : τ *ᵥ ζ₀ = lam • ζ) : lam * star lam * (star ζ ⬝ᵥ G *ᵥ ζ) = t * (star ζ₀ ⬝ᵥ G' *ᵥ ζ₀)

theorem **alg_norm** {R : Type} [CommRing R] {A A' B B' X X' e ec l l' h k : R} (hN : l * l' * h = -(X * -X) * k) (g1 : A * A' = 1) (g2 : B * B' = 1) (g3 : X * X' = 1) : A' * -X' * e * B * l' * h * (B' * - -X' * ec * A * l * h) = -(e * ec * h * k)

theorem **alg_gen** {R : Type} [CommRing R] {A' B X X' e e' l li l' h k a : R} (hN : l * l' * h = -(X * -X) * k) (ee : e' * e = 1) (g3 : X * X' = 1) (g4 : l * li = 1) : -(A' * -X' * e * B * l' * h * (e' * a)) = k * (A' * (X * B) * li * a)

theorem **rank_one** {Z S : HSp D} {ζ₀ : Fin Z.rank → D.host.E} {LZ : Lat Z} {LS : Lat S} (hLZ : IsSelfDualC Z LZ) (hLS : IsSelfDualC S LS) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ y, y ∈ latA Z LZ (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV Z LZ (· = x)) (hμ : ∀ a ∈ idAt D (· = x), a • ratV ζ₀ ∈ latV Z LZ (· = x) ↔ a ∈ intAt D (· = x)) {s' : Fin S.rank → AE D} (hs' : s' ∈ latV S LS (· = x)) : ∃ w ∈ latO S Z (LS : Set _) (LZ : Set _) (· = x), ∀ y, (show Matrix (Fin Z.rank) (Fin S.rank) (AE D) from w) *ᵥ y = ((α : AE D) * hermAt S y s') • ratV ζ₀

theorem **top_keeps** {L L'' S : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) {ζ : Fin L.rank → D.host.E} {ζ₀ : Fin L''.rank → D.host.E} {lam : D.host.E} (hτ : τ *ᵥ ζ₀ = lam • ζ) {LL : Lat L} {LZ : Lat L''} {LS : Lat S} (hLL : IsSelfDualC L LL) (hLZ : IsSelfDualC L'' LZ) (hLS : IsSelfDualC S LS) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hμ : ∀ a ∈ idAt D (· = x), a • ratV ζ₀ ∈ latV L'' LZ (· = x) ↔ a ∈ intAt D (· = x)) (hk : ∀ w ∈ latO S L'' (LS : Set _) (LZ : Set _) (· = x), kap τ S w ∈ latO L S (LL : Set _) (LS : Set _) (· = x)) {e : AE D} (hl : (e * idem D (· = x)) • ratV ζ ∈ latV L LL (· = x)) : ∀ s' ∈ latV S LS (· = x), (((α⁻¹ : (AE D)ˣ) : AE D) * incA D (-D.host.xi0⁻¹) * e * cA D α * cA D (incA D lam) * incA D (star ζ ⬝ᵥ L.gram *ᵥ ζ)) • s' ∈ latV S LS (· = x)

theorem **top_line** {L L'' S : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) {ζ : Fin L.rank → D.host.E} {ζ₀ : Fin L''.rank → D.host.E} (hζ : IsLineVec L ζ) (hζ₀ : IsLineVec L'' ζ₀) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : τ *ᵥ ζ₀ = lam • ζ) (hsim : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) {LL : Lat L} {LZ : Lat L''} {LS : Lat S} (hLL : IsSelfDualC L LL) (hLZ : IsSelfDualC L'' LZ) (hLS : IsSelfDualC S LS) (hr : 0 < S.rank) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hμ : ∀ a ∈ idAt D (· = x), a • ratV ζ₀ ∈ latV L'' LZ (· = x) ↔ a ∈ intAt D (· = x)) (hk : ∀ w ∈ latO S L'' (LS : Set _) (LZ : Set _) (· = x), kap τ S w ∈ latO L S (LL : Set _) (LS : Set _) (· = x)) : ∀ a ∈ idAt D (· = x), a • ratV ζ ∈ latV L LL (· = x) ↔ (((α * (xiU D * cU α)⁻¹ * inU lam hlam)⁻¹ : (AE D)ˣ)

theorem **last_iff** {V₀ W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (z : Fin V₀.rank → D.host.E) {M : Lat V₀} {LW : Lat W} (hM : IsSelfDualC V₀ M) {x : Pl D} (hLW : IsSelfDualC W LW) {α m : (AE D)ˣ} (r : Fin ρ) (hr : r.1 + 1 = ρ) (hα : ∀ y, y ∈ latA W LW (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV W LW (· = x)) (hW : ∀ a ∈ idAt D (· = x), colMul d.jY r a ∈ latV W LW (· = x) ↔ cA D ((xiU D * cU α * m : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) {a : AE D} (ha : a ∈ idAt D (· = x)) : d.incAdj V₀ (a • yZero V₀ ρ z) ∈ latO V₀ W (M : Set _) (LW : Set _) (· = x) ↔ ((m : AE D) * a) • ratV z ∈ latV V₀ M (· = x)

theorem **dual_line** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} (hL : IsSelfDualC S L) {x : Pl D} (hm : MonoAt d L L₀ x) (i : Fin ρ) {b : (AE D)ˣ} (hX : ∀ a ∈ idAt D (· = x), colMul d.jX i a ∈ latV S L (· = x) ↔ ((b⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : ∀ a ∈ idAt D (· = x), colMul d.jY i a ∈ latV S L (· = x) ↔ cA D (b : AE D) * a ∈ intAt D (· = x)

theorem **step_line** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} {LV : Lat B.V} {LV₀ : Lat B.V₀} {LW : Lat J.W} {LW₀ : Lat J.W₀} {LV' : Lat B'.V} {LV₀' : Lat B'.V₀} {x : Pl D} {α m : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (h₁ : GoodBF ψ B J LV LV₀ LW LW₀ x) (h₂ : AdaptFB ψ J B' LW LW₀ LV' LV₀' x) (hz : ∀ a ∈ idAt D (· = x), a • ratV (vecB B'.o B'.ζ) ∈ latV B'.V₀ LV₀' (· = x) ↔ ((m⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : ∀ a ∈ idAt D (· = x), a • ratV (vecB B.o B.ζ) ∈ latV B.V₀ LV₀ (· = x) ↔ (((α * (xiU D * cU α)⁻¹ * m)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **core_line** (hc : c.Is) (hs : ShapeOff v o) (hp : TopPrim v o) (hr : ∀ j, j + 2 * 0 + 2 = D.R.m → 0 < (c.X (j + 1)).rank) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : ∀ j h, v.τ j h *ᵥ o.ζ₀ = lam • (c.bes j 0 h).ζ) {x : Pl D} (hx : x ∉ v.tP) {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (ρ : ℕ) : ∀ j (h : j + 2 * ρ + 2 = D.R.m), ∀ a ∈ idAt D (· = x), a • ratV (vecB (c.bes j ρ h).o (c.bes j ρ h).ζ) ∈ latV (c.bes j ρ h).V₀ (o.M j ρ h) (· = x) ↔ ((((α * (xiU D * cU α)⁻¹) ^ (ρ + 1) * inU lam hlam)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IntegrandSZero

theorem **integrand_sZero** {A B : HSp D} (om : WRep A B) (m : Measure (GA D A)) (ν : Measure (WO A B)) (ℱ : Set (WO A B)) (L : Lat A) (a : GL (Fin A.rank) (AE D)) (Φ₁ Φ₂ : WO A B → ℂ) (u₁ u₂ : GA D A → ℂ) (g : GA D A) : integrand om m ν ℱ L a Φ₁ Φ₂ u₁ u₂ (sZero A B) g = wcoef om ν ℱ Φ₁ Φ₂ g * rcoef m u₁ u₂ g

theorem **one_mem_openOff** (A : HSp D) (L : Lat A) (Sf : Finset (FinPl D)) : (1 : GA D A) ∈ openOff A L Sf

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsCompactParahoric

theorem **isReps_central** {g : GL (A ⊕ D) k} (hg : ∀ j, j * g = g * j) : IsReps g {1} (fun _ => 1)

theorem **isTOp_central** {V : Type} [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (A ⊕ D) k) V) (ξ : kˣ →* ℂˣ) {g : GL (A ⊕ D) k} (hg : ∀ j, j * g = g * j) {x : V} (hx : x ∈ typeSpace π ξ) (y : V) : IsTOp π ξ g x y ↔ y = π g x

theorem **comm_one** (j g : GL (Empty ⊕ Unit) k) : j * g = g * j

theorem **det_one_block** (g : GL (Empty ⊕ Unit) k) : g.1.det = g.1 (.inr ()) (.inr ())

theorem **one_mem_type** (χ : kˣ →* ℂˣ) : (1 : ℂ) ∈ typeSpace (detRep (Empty ⊕ Unit) χ) χ

theorem **isGen_rank_one** (χ : kˣ →* ℂˣ) (ϖ : kˣ) (b : ℂˣ) : IsGen (detRep (Empty ⊕ Unit) χ) χ ϖ {b} (1 : ℂ) ↔ b = χ ϖ

theorem **isCoType_rank_one** (χ : kˣ →* ℂˣ) : IsCoType (detRep (Empty ⊕ Unit) χ) χ LinearMap.id

theorem **omTwo_self** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (u : Matrix b a k → ℂ) : omTwo μ μ g h u = om μ⁻¹ g h u

theorem **omTwo_apply** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μS μT : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (u : Matrix b a k → ℂ) (x : Matrix b a k) : omTwo μS μT g h u x = (((μT (GeneralLinearGroup.det g))⁻¹ * μS (GeneralLinearGroup.det h) : ℂˣ) : ℂ) * om 1 g h u x

theorem **liftS_diag** (ξ : kˣ →* ℂˣ) (ψ : Matrix D D' k → ℂ) (y : Matrix D D' k) (hy : ∀ i j, y i j ∈ 𝒪[k]) : liftS ξ ψ (fromBlocks (1 : Matrix A A k) 0 0 y) = ψ y

theorem **segL_one** (ϖ : kˣ) (z : ℂˣ) : segL ϖ z 1 = {z}

theorem **exists_reps** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] (J : Subgroup G) (ho : IsOpen (J : Set G)) (hc : IsCompact (J : Set G)) (g : G) : ∃ R : Finset G, (∀ j ∈ R, j ∈ J) ∧ ∀ j ∈ J, ∃! j', j' ∈ R ∧ (j' * g)⁻¹ * (j * g) ∈ J

theorem **isOpen_parahoric** : IsOpen (parahoric k A D : Set (GL (A ⊕ D) k))

theorem **isCompact_parahoric** : IsCompact (parahoric k A D : Set (GL (A ⊕ D) k))

theorem **det_lower_ne** {j : GL (A ⊕ D) k} (hj : j ∈ parahoric k A D) : j.1.toBlocks₂₂.det ≠ 0

theorem **exists_isReps** (g : GL (A ⊕ D) k) : ∃ R d, IsReps g R d

theorem **isTOp_unique** {V : Type} [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (A ⊕ D) k) V) (ξ : kˣ →* ℂˣ) (g : GL (A ⊕ D) k) {x y y' : V} (h : IsTOp π ξ g x y) (h' : IsTOp π ξ g x y') : y = y'

theorem **exists_isRepsK** (g : GL D k) : ∃ R, IsRepsK g R

theorem **exists_isHgt** {n : Type} [Fintype n] [DecidableEq n] {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) : ∃ H : GL n k → ℝ, IsHgt ϖ H

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsFinCombSingleZero

theorem **isFinComb_single_zero** {S T : HSp D} (a : ℂ) : IsFinComb (Finsupp.single (0 : WO S T) a)

theorem **comb_single** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (u : WO S T) (a : ℂ) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single u a) f = a • heisO ψ S T u f

theorem **exists_onb** {n : ℕ} {G : Matrix (Fin n) (Fin n) ℂ} (h : G.PosDef) : ∃ B : Matrix (Fin n) (Fin n) ℂ, Bᴴ * G * B = 1

theorem **posDef_neg_of_sig** {n : ℕ} (G : Matrix (Fin n) (Fin n) ℂ) (h : Formalism.UG.HasSignature G 0 n) : (-G).PosDef

theorem **exists_cartan_pm** {n : ℕ} (G : Matrix (Fin n) (Fin n) ℂ) (h : Formalism.UG.HasSignature G n 0 ∨ Formalism.UG.HasSignature G 0 n) : ∃ s B : Matrix (Fin n) (Fin n) ℂ, IsCartanInv G s ∧ (s = 1 ∨ s = -1) ∧ ∃ p, IsONBasis G s B p

theorem **coCpt_of_isDefinite** {V : HSp D} (hs : V.SigOK) (hd : V.IsDefinite) : CoCpt V

theorem **exists_defSp** {V : HSp D} (hs : V.SigOK) (hd : V.IsDefinite) : ∃ x : DefSp V, x.Is

theorem **exists_defSp_X** (D : D7) (j : ℕ) (hj : j ≤ D.R.m) : ∃ x : DefSp (D.sp.X j), x.Is

theorem **vac_ne_zero** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {sS sT LS LT} {f : WO S T → ℂ} (h : IsVacO ψ θ S T sS sT LS LT f) : f ≠ 0

theorem **exists_kerDat** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {S T : HSp D} {x : DefSp S} {y : DefSp T} (hx : x.Is) (hy : y.Is) : ∃ k : KerDat S T, k.Is ψ θ x y ∧ k.vac ≠ 0 ∧ k.c = Finsupp.single 0 1

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsGLExistsUnique

theorem **exists_unram** (ϖ : kˣ) (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (c : ℂˣ) : ∃ η : kˣ →* ℂˣ, IsUnram η ∧ η ϖ = c

theorem **exists_psi** (P : HeightOneSpectrum (𝒪 E)) (q D : ℕ) (ι : E →+* k) (ϖ : kˣ) (σ₀ : K →+* ℂ) (psi : E → K) (hc : IsCompl P q ι ϖ) (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (hD : 1 ≤ D) (hadd : ∀ x y, IntAt P ((q : E) ^ D * x) → IntAt P ((q : E) ^ D * y) → psi (x + y) = psi x * psi y) (hint : ∀ x, IntAt P x → psi x = 1) (hne : psi ((q : E)⁻¹) ≠ 1) : ∃ ψ : AddChar k Circle, HasConductor ψ 1 ∧ ∀ a, IntAt P ((q : E) ^ D * a) → (ψ (ι a) : ℂ) = σ₀ (psi a)

theorem **exists_delta** (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (ϖ : kˣ) (hc : IsCompl P q ι ϖ) (χ : (𝒪 E ⧸ P.asIdeal)ˣ →* ℂˣ) : ∃ δ : kˣ →* ℂˣ, IsTame δ ∧ ∀ a (u : kˣ), IntAt P a → ∀ hu : IsUnit (redP P a), (u : k) = ι a → δ u = χ hu.unit

theorem **exists_atTwist** (s : ℕ) (hP : P.1 ∉ Pr.vGs) (hs : R.m = s + 2) (hok : cst.Ok) (hmu : ∀ i, i < R.m → cst.muq (i + 1) P ≠ 0) (hq : ((qOf P.1 : ℕ) : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) : ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (ι : X.E →+* k) (W : TwData k), AtTwist (R

theorem **sg_sq** (h : AtTwist (R

theorem **not_isUnram** (h : AtTwist (R

theorem **isGL_existsUnique** (j c : ℕ) (hc : cOf R j = c) (γ : AutY X R j) : ∃! g : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin c) k, IsGL ι γ g

theorem **valuation_spec** (h : AtTwist (R

theorem **mu_agree** (h : AtTwist (R

theorem **nabs_agree** (h : AtTwist (R

theorem **toLoc_eq** {A B : Type} {a b : ℕ} (e : A ≃ Fin a) (e' : B ≃ Fin b) (x : (Fin b → X.E) →ₗ[X.E] (Fin a → X.E)) : toLoc ι e e' x = ((LinearMap.toMatrix' x).map ι).submatrix e e'

theorem **toLoc_mul** {A B C : Type} [Fintype B] {a b c : ℕ} (e : A ≃ Fin a) (e' : B ≃ Fin b) (e'' : C ≃ Fin c) (x : (Fin b → X.E) →ₗ[X.E] (Fin a → X.E)) (y : (Fin c → X.E) →ₗ[X.E] (Fin b → X.E)) : toLoc ι e e' x ∗ toLoc ι e' e'' y = toLoc ι e e'' (x ∘ₗ y)

theorem **det_toLoc** {A : Type} [Fintype A] [DecidableEq A] {a : ℕ} (e : A ≃ Fin a) (γ : (Fin a → X.E) ≃ₗ[X.E] (Fin a → X.E)) (g : GL A k) (hg : g.val = toLoc ι e e γ.toLinearMap) : ((Matrix.GeneralLinearGroup.det g : kˣ) : k) = ι (LinearEquiv.det γ : X.Eˣ)

theorem **om_agree** (h : AtTwist (R

theorem **twData_closed** : ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (W : TwData k), ¬ IsUnram W.δ ∧ (∀ j, W.η j ≠ 1) ∧ W.sg = -1

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsOpenKerUnramMulTame

theorem **isOpen_ker_unram_mul_tame** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {η δ : kˣ →* ℂˣ} (hη : IsUnram η) (hδ : IsTame δ) : IsOpen ((η ∗ δ).ker : Set kˣ)

theorem **T_regular** : (G.T δ hδ η).Regular

theorem **T_laws** (hη : ∀ j, IsUnram (η j)) : (G.T δ hδ η).Laws

theorem **T_le** (j d : ℕ) (h : j + d = G.s + 1) : ((G.T δ hδ η).lev j d h).J ≤ KO G.k _

theorem **T_leTop** : (G.T δ hδ η).levTop.J ≤ KO G.k _

theorem **T_openTop** : IsOpen ((G.T δ hδ η).levTop.J : Set (GL (Ix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t) (G.s + 2)) G.k))

theorem **reg** : (G.T E.δ E.tame E.η).Regular

theorem **laws** : (G.T E.δ E.tame E.η).Laws

theorem **le** : ∀ j d h, ((G.T E.δ E.tame E.η).lev j d h).J ≤ KO G.k _

theorem **leT** : (G.T E.δ E.tame E.η).levTop.J ≤ KO G.k _

theorem **openT** : IsOpen ((G.T E.δ E.tame E.η).levTop.J : Set (GL (Ix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t) (G.s + 2)) G.k))

theorem **mu** : ∀ j, j ≤ G.s + 1 → (ofSplit G.q).IsComp (χ j)⁻¹ (E.η (j + 1) ∗ E.δ) ∧ IsOpen ((E.η (j + 1) ∗ E.δ).ker : Set G.kˣ)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsPairResThetaInt

theorem **residueAt_zero** (s₀ : ℂ) : ResidueAt (fun _ : ℂ => (0 : ℂ)) s₀ 0

theorem **IsPairRes.eq_zero** {S T' : HSp D} {ω' : GA D S × GA D T' →* Module.End ℂ (WO S T' → ℂ)} {m : Measure (GA D T')} {Ψ : WO S T' → ℂ} {B : GA D S → ℂ} (h : IsPairRes ω' m Ψ B) {g : GA D S} (h0 : ∀ x, thetaKO ω' g x Ψ = 0) : B g = 0

theorem **IsPairRes.eq_of_kernel** {S T' : HSp D} {ω' : GA D S × GA D T' →* Module.End ℂ (WO S T' → ℂ)} {m : Measure (GA D T')} {Ψ : WO S T' → ℂ} {B : GA D S → ℂ} (h : IsPairRes ω' m Ψ B) {g g' : GA D S} (h0 : ∀ x, thetaKO ω' g x Ψ = thetaKO ω' g' x Ψ) : B g = B g'

theorem **isPairRes_thetaInt** {S T' : HSp D} {ω' : GA D S × GA D T' →* Module.End ℂ (WO S T' → ℂ)} {m : Measure (GA D T')} {Ψ : WO S T' → ℂ} (h : ∀ g, ThetaPairConv ω' m Ψ 1 g) (κ : ℂ) : IsPairRes ω' m Ψ (κ • thetaInt ω' m Ψ)

theorem **met_of** (h : FirstTermWeak) (A T T' : HSp D) (t : ℕ) (d : HypDec T T' t) (hA : HSpace.SigOK A) (hT : HSpace.SigOK T) (hr : T.rank + T'.rank = 2 ∗ A.rank) (hlt : T'.rank < T.rank) (h0 : 0 < T'.rank) (hc : IsAniso T' ∨ ∃ (T₀ : HSp D) (r' : ℕ) (_ : HypDec T' T₀ r'), IsAniso T₀ ∧ 0 < r') : Met A T T'

theorem **isAniso_pos** (D : D7) (n : ℕ) : IsAniso (posSp D n)

theorem **met_aniso** (h : FirstTermWeak) (D : D7) : Met (posSp D 2) (hsum (posSp D 1) (hypSp D 1)) (posSp D 1)

theorem **met_iso** (h : FirstTermWeak) (D : D7) : Met (posSp D 4) (hsum (hsum (posSp D 1) (hypSp D 1)) (hypSp D 1)) (hsum (posSp D 1) (hypSp D 1))

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsScaleWithBoundedGamma

theorem **TrueF.elim** {Q : D7 → Prop} (h : ∀ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z), I.IsTw → IdE w.S.X → Q (I.rec7 l Z)) {D : D7} (hD : TrueF D) : Q D

theorem **Row.mono** {E E' : D7 → ℝ → ℝ → Prop} (hE : ∀ D L C', E' D L C' → E D L C') {Q : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} (h : Row E Q) : Row E' Q

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsStageTrivialTwist

theorem **eq_triv** {Pg : Finset (Pl D.host.E)} {r : ℕ} (P : LevelP Pg r ℂ) (h : P.H = ⊥) : P = LevelP.triv _ _ _

theorem **trivialTwist_isChain** {c : Com D} (hc : c.Is) (h : c.P.H = ⊥) (ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) : (tw0 c ξ).Is c where

theorem **isStage_trivialTwist** {c : Com D} (h : c.P.H = ⊥) {j : ℕ} {s : PiLocalization₂ D} (hs : s.Is c j) : ∃ ξ, IsStage c (tw0 c ξ) j s

theorem **isStep_trivialTwist** {c : Com D} {j : ℕ} {s s' : PiLocalization₂ D} {p : Step c s s'} (hp : p.Is j) (ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) : IsStep (tw0 c ξ) j p

theorem **twist_is** {c : Com D} (hc : c.Is) {Q : Finset (Pl D.host.E)} (hQ : Q ⊆ c.Pg) {ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ} (hξ : ∀ PP ∈ Q, ξ PP ≠ 1) (hχ : ∀ j : Fin D.R.m, ∀ PP ∈ Q, c.chi ((j : ℕ) + 1) PP = (ξ PP)⁻¹) : ((c.Pg \ Q, Q, c.chi, ξ) : Twist D).Is (comQ c Q ξ) where

theorem **isStage_twStage** {c : Com D} (w : Twist D) {j : ℕ} {s : PiLocalization₂ D} (hs : s.Is c j) : ∃ hr, IsStage c w j (twStage w j s hr)

theorem **one_lt_norm** (PP : Pl D.host.E) : (1 : ℝ) < (Ideal.absNorm PP.asIdeal : ℝ)

theorem **zinv_pos** (PP : Pl D.host.E) (a b : ℕ) : 0 < zinv PP a b

theorem **zinv_mul** (PP : Pl D.host.E) {a b c : ℕ} (h : a ≤ b) (h' : b ≤ c) : zinv PP a b ∗ zinv PP b c = zinv PP a c

theorem **idx_vol** (PP : Pl D.host.E) (k : ℕ) : idxQ PP (k + 1) ∗ volOne PP (k + 1) ∗ zinv PP D.R.t ((k + 1) ∗ D.R.t) = zinv PP D.R.t (D.R.r + (k + 1) ∗ D.R.t)

theorem **const_eq** {c : Com D} {s s' : PiLocalization₂ D} (w : Twist D) (k : ℕ) (p : Step c s s') (m : Measure (WO s.V s'.V)) (X : Set (WO s.V s'.V)) (mR : Measure (GinfC s.V)) (φ : PolF D s'.V.rank s.V.rank) (v : s.M.Vec) (L H H₀ : ℂ) (hH : H₀ = H ∗ ∏ PP ∈ w.Q, blockL

c k PP) (h0 : dS w.S₀ s s' ≠ 0) (hd : dS (c.S : Set _) s s' = dS w.S₀ s s' * ((∏ PP ∈ w.Q, zinv PP D.R.t (D.R.r + (k + 1) * D.R.t) : ℝ) : ℂ)) : (GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.volQ s'.V s'.ν : ℂ) * (L * H / dS (c.S : Set _) s s') * locQ c w k * ∫ g, zReal p m X φ v g ∂mR = kapOf w k p m X mR φ v H₀ * L

theorem **card_qPar** (c : Com D) (PP : Pl D.host.E) (x : (AE D)ˣ) (k : ℕ) : (qPar c PP x k).card = Fintype.card (GlobalTheory.IndefiniteRowSideFullHecke.VolumeFlatSection.Blocks k D.R.t)

theorem **locQ_zero** (c : Com D) (w : Twist D) : locQ c w 0 = 1

theorem **stepLam_of** (h1 : SoftQ D) (h2 : UpParQ D) (h3 : UpQ D) (h4 : QOfVal D) (h5 : UpVal D) (h6 : StepExact D) (h7 : LFull D) (h8 : Restore D) : StepLam D

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsSumFrame

theorem **handBF_eq** {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (φ : GA D B.A → ℂ) (Ξ : WO B.V₀ J.W₀ → ℂ) (h : GA D J.W₀) (w : WO J.W₀ L') : handBF s l φ Ξ h w = star (liftA B.o s.om₀ B.vA φ Ξ h (l.κ J.W₀ w))

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsTransportAdelicCorrection

theorem **rd_zero** {Ch : ℕ → Type} {chOf : ∀ i, (D.mem u).C.cls i → Ch i} (F : Formalism.L2.Forms.FormsTiedSym D u Ch chOf) (θ : InfFam D) (β : Ch 0) : GlobalTheory.AdForm.rd F θ 0 (Nat.zero_le _) β = (chart0 F.to0).rd θ

Development.AbelianVarietyTheory.Definitions.LatticeFrames.IsTrueWith

theorem **IsTrueWith.ind** {P : TP} {R : ∀ D : D7, Obj D → Prop} (h : ∀ K B f T (w : ColW B f T) (Z : w.Lt) (O : Obj (w.D Z)) (p : TruePt K w Z O), P K B f T w Z O p → R (w.D Z) O) {D : D7} {O : Obj D} (hD : IsTrueWith P D O) : R D O

theorem **IsTrue.ind** {R : ∀ D : D7, Obj D → Prop} (h : ∀ K B f T (w : ColW B f T) (Z : w.Lt) (O : Obj (w.D Z)), TruePt K w Z O → R (w.D Z) O) {D : D7} {O : Obj D} (hD : IsTrue D O) : R D O

theorem **IsTrue.selB** {D : D7} {O : Obj D} (h : IsTrue D O) : Nonempty (SelB O.ex)

theorem **TrueF.selBF** {eps : ℝ} {Fm : FamP eps} (h : TrueF Fm) : SelBF Fm

Development.AbelianVarietyTheory.Definitions.LatticeFrames.JointSmooth

theorem **idx_step** {j σ m : ℕ} (h : j + 2 * (σ + 1) + 2 = m) : j + 1 + 1 + 2 * σ + 2 = m

Development.AbelianVarietyTheory.Definitions.LatticeFrames.KerInclOn

theorem **EvAll.some** {F : Pts} {Q : ℝ → ∀ D : D7, D.what₂.Word → Prop} (h : F.EvAll Q) : F.EvSome Q

theorem **EvAll.and** {F : Pts} {Q Q' : ℝ → ∀ D : D7, D.what₂.Word → Prop} (h : F.EvAll Q) (h' : F.EvAll Q') : F.EvAll fun x D u => Q x D u ∧ Q' x D u

theorem **EvSome.rkAll** {F : Pts} (h : F.EvSome fun _ D u => RK D u) : F.RKAll

Development.AbelianVarietyTheory.Definitions.LatticeFrames.LatNotSelfDual

theorem **trace_iff** {ζ₀ : Fin a.Lz.rank → D.host.E} {PP : Formalism.L2.DIV.Pl D.host.E} (H : GX a ζ₀ PP) (g : GL (XIx (Fin D.R.k) (Fin D.R.t) 0) H.k) : embX (H.s + 1) g ∈ H.lev0.J ↔ g ∈ colGroup H.k D.R.k (Fin D.R.t)

theorem **inLev0_iff_unip** (κ : GA D (D.sp.X 0)) : InLev0 o G κ ↔ GlobalTheory.FinPl.Moves (D.sp.X 0) κ (a.x 0).L (a.x 0).L (fun _ => True) ∧ (∀ i, o.mat0 i κ • o.Q 0 i = o.Q 0 i) ∧ ∀ PP, (G PP).mat0 κ ∈ colGroup (G PP).k D.R.k (Fin D.R.t)

theorem **under_auto** {ζ₀ : Fin a.Lz.rank → D.host.E} {PP : Formalism.L2.DIV.Pl D.host.E} (H : GX a ζ₀ PP) : H.q.v = PP.under (𝒪 D.host.F)

theorem **val_lt_one_iff** {K : Type} [Field K] [ValuativeRel K] [TopologicalSpace K] [IsNonarchimedeanLocalField K] (q : GlobalTheory.FinPl.SplitPt D K) {PP : Formalism.L2.DIV.Pl D.host.E} (h : Formalism.L2.NTheta.IsIdemAt q.e PP) (x : 𝒪 D.host.E) : valuation K (q.pr (incA D (x : D.host.E))) < 1 ↔ x ∈ PP.asIdeal

theorem **oneG_iff** {ζ₀ : Fin a.Lz.rank → D.host.E} {PP : Formalism.L2.DIV.Pl D.host.E} (H : GX a ζ₀ PP) (T : ℕ) : valuation H.k (H.q.pr (incA D ((T : ℕ) : D.host.E)) - 1) < 1 ↔ ((T : 𝒪 D.host.E) - 1) ∈ PP.asIdeal

theorem **mem_of_arith** (h : Arith o G) (PP : a.c.Pg) : ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∈ PP.1.asIdeal

theorem **T_sub_one** (B : D7) (v : B.what₂.Word) (P : Ideal (𝒪 B.host.E)) (hP : ((B.host.g 0 * B.host.g 1 : ℕ) : 𝒪 B.host.E) ∈ P) : (((B.mem v).A.T : ℕ) : 𝒪 B.host.E) - 1 ∈ P

theorem **val_le_one** (x : 𝒪 D.host.E) : valuation K (q.pr (incA D (x : D.host.E))) ≤ 1

theorem **val_le_one_of** (h : Formalism.L2.NTheta.IsIdemAt q.e PP) {y : D.host.E} (hy : PP.valuation D.host.E y ≤ 1) : valuation K (q.pr (incA D y)) ≤ 1

theorem **val_eq_unif** (h : Formalism.L2.NTheta.IsIdemAt q.e PP) (ϖ : Unif K) {x : 𝒪 D.host.E} (h1 : x ∈ PP.asIdeal) (h2 : x ∉ PP.asIdeal ^ 2) : valuation K (q.pr (incA D (x : D.host.E))) = valuation K ((ϖ.1 : 𝒪[K]) : K)

theorem **unif_val** (ϖ : Unif K) {π : Kˣ} (h : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif π) : valuation K ((ϖ.1 : 𝒪[K]) : K) = valuation K (π : K)

theorem **unifG_of** {ζ₀ : Fin a.Lz.rank → D.host.E} {PP : Formalism.L2.DIV.Pl D.host.E} (H : GX a ζ₀ PP) (h1 : ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) (h2 : ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal ^ 2) : valuation H.k (H.q.pr (incA D ((D.host.g 0 * D.host.g 1 : ℕ) : D.host.E))) = valuation H.k ((H.S.ϖ : H.kˣ) : H.k)

theorem **exists_prime** {K : Type} [Field K] [ValuativeRel K] [TopologicalSpace K] [IsNonarchimedeanLocalField K] (q : GlobalTheory.FinPl.SplitPt D K) : ∃ PP : Formalism.L2.DIV.Pl D.host.E, Formalism.L2.NTheta.IsIdemAt q.e PP

theorem **notMem_sq** {p : ℕ} (h : GlobalTheory.IndefiniteRowSideFullHecke.SplitColumnClasses.SplitsCompletely D.host.E p) (P : HeightOneSpectrum (𝒪 D.host.E)) (hP : (p : 𝒪 D.host.E) ∈ P.asIdeal) : (p : 𝒪 D.host.E) ∉ P.asIdeal ^ 2

theorem **unifT_of** (p : GlobalTheory.FinPl.SplitPt D k) (ϖ : Unif k) (h : (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∈ p.v.asIdeal) : valuation k (p.pr (incA D (((D.mem u).A.T : ℕ) : D.host.E))) = valuation k ((ϖ.1 : 𝒪[k]) : k)

theorem **g_notMem_sq** (P : HeightOneSpectrum (𝒪 D.host.E)) : ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∉ P.asIdeal ^ 2

theorem **arith_iff** (hT : ∀ i, (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∈ (o.p i).v.asIdeal) (hG : ∀ PP ∈ a.c.Pg, ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) : Arith o G ↔ (∀ v, v ∉ a.c.S → CharUnramAt D a.c.η v ∧ ∀ j, 1 ≤ j → j ≤ D.R.m → CharUnramAt D (a.c.μ j) v) ∧ a.Lz.SigOK ∧ a.Lz.IsPosDef

theorem **comp_map** {F E : Type} [Field F] [Field E] [Algebra F E] {R R' R'' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] [CommRing R''] [Algebra F R''] (φ : R →ₐ[F] R') (ψ : R' →ₐ[F] R'') (z : R ⊗[F] E) : Algebra.TensorProduct.map (ψ.comp φ) (AlgHom.id F E) z = Algebra.TensorProduct.map ψ (AlgHom.id F E) (Algebra.TensorProduct.map φ (AlgHom.id F E) z)

theorem **charUnramAt_of** {χ : (AE D)ˣ →* Circle} {v : HeightOneSpectrum (𝒪 D.host.F)} (h : ∀ w : HeightOneSpectrum (𝒪 D.host.E), w.under (𝒪 D.host.F) = v → GlobalTheory.LFun.UnrAt χ w) : CharUnramAt D χ v

theorem **arith_of_pins** (hc : a.c.Is) (hP : GlobalTheory.Core.Mi.IsTypePlaces a.c.Pg) (ho : PinOffX o) (hL : a.Lz.SigOK ∧ a.Lz.IsPosDef) : Arith o G

theorem **lat_not_selfDual** {D : D7} {u : D.what₂.Word} (F : Forms0 D u) (L : Submodule (𝒪 D.host.E) (Fin (D.sp.X 0).rank → D.host.E)) (sd : IsSelfDualC (D.sp.X 0) L) (lat : ∀ z : Fin (D.sp.X 0).rank → D.host.E, z ∈ L ↔ (fun i => z (Fin.cast (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)).symm i)) ∈ F.cls0.lev.Lb) : False

Development.AbelianVarietyTheory.Definitions.LatticeFrames.LatticeFramesOfHyps

theorem **Hypotheses3.q8** (h : Hypotheses3) : ∀ D : Formalism.BasicNotions.D7, GlobalTheory.Core.Rk.SW.HeckeFormNS D

theorem **Hypotheses3.of_classicalTheorems2** (h : Hypotheses2) : Hypotheses3

theorem **Hypotheses3.toHypotheses2** (h : Hypotheses3) : Hypotheses2

Development.AbelianVarietyTheory.Definitions.LatticeFrames.LevChiEqOne

theorem **chOf_onto** (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) : Function.Surjective (A.chOf i)

theorem **chart_ext** (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (α β : Ch i) (hb : A.b0 i α = A.b0 i β) (hc : cosetIn (A.p0 i α) = cosetIn (A.p0 i β)) : α = β

theorem **chi_pow** (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) : ∃ n : ℕ, 0 < n ∧ ∀ g, A.chi i β g ^ n = 1

theorem **norm_chi** (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) (g : A.U i β) : ‖((A.chi i β g : ℂˣ) : ℂ)‖ = 1

theorem **lev_chi_eq_one** (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) (κ : (A.K i hi β).lev.U) (h : ∀ v, v ∈ Formalism.L2.Forms.badG (I.rec7 l Z) ∨ I.UnderQ a v → Formalism.UG.bc (Formalism.UG.prV (E

theorem **PinColN.w_ne** (h : PinColN A ψ θ x vac c₀ kom dd w Ram rd cK) (j : ℕ) (hj : j < B.R.m) (α : Ch j) (γ : Ch (j + 1)) : w j α γ ≠ 0 ∧ volLev (x j).vol (A.U j α) ≠ 0 ∧ c₀ j ≠ 0

theorem **PinColT.vacT** (hX : PinX I l Z ψ θ x vac c₀) (h : PinColT A ψ θ x vac c₀ kom dd w Ram rd cK) (j : ℕ) (hj : j + 1 ≤ B.R.m) : VacT I l Z θ j hj (c₀ j) (A.chr (j + 1)) (kom j) (vac j)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.LevelLawOfPairLaw

theorem **inFlag_one** : s.InFlag 1

theorem **inFlag_mul** {κ κ' : GA D s.V} (h : s.InFlag κ) (h' : s.InFlag κ') : s.InFlag (κ * κ')

theorem **ch_ne_zero** (κ : GA D s.V) : s.ch κ ≠ 0

theorem **ch_mul** {κ κ' : GA D s.V} (h : s.InFlag κ) (h' : s.InFlag κ') : s.ch (κ * κ') = s.ch κ * s.ch κ'

theorem **ch_one** : s.ch 1 = 1

theorem **moved_one** (y : s.CC c) : Moved 1 y y

theorem **moved_unique** {κ : GA D s.V} {y a b : s.CC c} (ha : Moved κ y a) (hb : Moved κ y b) : a = b

theorem **moved_mul** {κ κ' : GA D s.V} (h : s.InFlag κ) (h' : s.InFlag κ') {y a b : s.CC c} (ha : Moved κ' y a) (hb : Moved κ a b) : Moved (κ * κ') y b

theorem **inFlag_iff** (κ : GA D s.V) : s.InFlag κ ↔ Lev.IsLevelLift s.V s.β s.L (rTup s.β s.L (Lev.finE s.V κ)) κ

theorem **ch_eq_levChi** {j : ℕ} (hs : s.Is c j) (hP : c.P.H = ⊥) (κ : GA D s.V) : s.ch κ = Lev.levChi s.V s.β s.L κ

theorem **rTup_eq_one** {j : ℕ} (hs : s.Is c j) (hP : c.P.H = ⊥) {κ : GA D s.V} (hκ : κ ∈ Lev.levG s.V s.β s.L) : rTup s.β s.L (Lev.finE s.V κ) = 1

theorem **sameBlock_of_one** {j j' : ℕ} (hs : s.Is c j) (hs' : s'.Is c j') {κ : GA D s.V} {κ' : GA D s'.V} (h : rTup s.β s.L (Lev.finE s.V κ) = 1) (h' : rTup s'.β s'.L (Lev.finE s'.V κ') = 1) : SameBlock s s' κ κ'

theorem **exists_sameBlock** {j j' : ℕ} (hs : s.Is c j) (hs' : s'.Is c j') (hl : Lev.LiftsExist s'.V s'.β s'.L) (κ : GA D s.V) : ∃ κ', s'.InFlag κ' ∧ SameBlock s s' κ κ'

theorem **levelLaw_of_pairLaw** (h : PairLaw D) : ForStep D fun c _ s s' p => c.P.H = ⊥ → ∀ φ, VecLaw p.ω (Lev.levG s.V s.β s.L) (Lev.levChi s.V s.β s.L) (Lev.levG s'.V s'.β s'.L) (fun κ => (Lev.levChi s'.V s'.β s'.L κ)⁻¹) (p.pt φ)

theorem **steps_of_is** (W : GlobalTheory.Core.Rk.Tower D) (hW : W.Is) (hQ : W.Q () = ∅) : (∀ k, k + 1 < D.R.m → (stepY W k).Is (k + 1)) ∧ (stepX W).Is D.R.m

theorem **lev_of_Q** (W : GlobalTheory.Core.Rk.Tower D) (hQ : W.Q () = ∅) : (∀ j, ∃ hr ξ, (stageY W j).L = Lev.datY (stageY W j).V.rank D.R.r D.R.t j hr (comOf W).Pg ∅ (comOf W).P (comOf W).chi ξ Lev.Frm.B) ∧ (∃ hr ξ, (stageX W).L = Lev.datY (stageX W).V.rank D.R.r D.R.t D.R.m hr (comOf W).Pg ∅ (comOf W).P (comOf W).chi ξ Lev.Frm.B) ∧ (comOf W).P.H = ⊥

theorem **q_empty** (W : GlobalTheory.Core.Rk.Tower D) (hW : W.Is) : W.Q () = ∅

theorem **steps_of_is'** (W : GlobalTheory.Core.Rk.Tower D) (hW : W.Is) : (∀ k, k + 1 < D.R.m → (stepY W k).Is (k + 1)) ∧ (stepX W).Is D.R.m

Development.AbelianVarietyTheory.Definitions.LatticeFrames.LineLatAt

theorem **cI_step** (D : D7) {j : ℕ} (h : j + 1 + 1 ≤ D.R.m) : FractionalIdeal.spanSingleton _ D.host.xi0 * cI D j = D.host.AA * D.host.AAbar⁻¹ * cI D (j + 1 + 1)

theorem **idV_mono** {a b : FI D} (h : a ≤ b) (v : Pl D) : idV a v ≤ idV b v

theorem **idV_zero** (v : Pl D) : idV (0 : FI D) v = ⊥

theorem **YPart.Sk.withT** {ζ₀ : Fin a.Lz.rank → D.host.E} {τ} (h : b.Sk ζ₀ τ) (t : ∀ j, GA D ((a.geo b.bot b.mid).st j).V) (ht : ∀ j, t j ∈ GfC _) : (b.withT t).Sk ζ₀ τ where

theorem **CoreLat.withT** {β₀ M} (h : CoreLat o b β₀ M) (t : ∀ j, GA D ((a.geo b.bot b.mid).st j).V) : CoreLat o (b.withT t) β₀ M

Development.AbelianVarietyTheory.Definitions.LatticeFrames.LinkSimil

theorem **jointCont** (D : D7) : JointCont D

theorem **quasiFam_of_weil** {S L : HSp D} {χS χL} {om : Om S L} (hom : IsWeilDatumO ψ S L χS χL om) {f : WO S L → ℂ} (hf : IsThetaFunO ψ S L f) {φ : GA D S → ℂ} (hφ : IsContAut S φ) : IsQuasiFam ψ S L fun h w => star (φ h) * om (h, 1) f w

theorem **exists_quasiFam** (D : D7) (m : ℕ) (hW : ∀ ψ, IsStdChar D ψ (-1) → ∃ χS χL om, IsWeilDatumO ψ (posSp D m) (posSp D 1) χS χL om) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (Y : GA D (posSp D m) → WO (posSp D m) (posSp D 1) → ℂ), IsStdChar D ψ (-1) ∧ IsQuasiFam ψ _ _ Y ∧ Y 1 ≠ 0

theorem **trace_inverseGram_conj_smul_eq** {R : Type} [CommRing R] {m n : ℕ} (c : R) (a : Matrix (Fin m) (Fin m) R) (b : Matrix (Fin m) (Fin n) R) (g : Matrix (Fin n) (Fin n) R) (y : Matrix (Fin n) (Fin m) R) : (a · (b · ((c • g) · y))).trace = c * (a · (b · (g · y))).trace

theorem **hO_ofAdj** (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (c : D.host.E) (hτ : τᴴ · (L.gram · τ) = c • L'.gram) (x y : WO S L') : hO L S (ofAdj L S ((rat τ : Mx D L.rank L'.rank) · x)) (ofAdj L S ((rat τ : Mx D L.rank L'.rank) · y)) = incA D (c / (D.host.xi0 * star D.host.xi0)) * hO S L' y x

theorem **normSq_xi_ne_zero** : D.host.xi0 * star D.host.xi0 ≠ 0

theorem **sympl_link** (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (hτ : τᴴ · (L.gram · τ) = (D.host.xi0 * star D.host.xi0) • L'.gram) (x y : WO S L') : symplOH L S (ofAdj L S ((rat τ : Mx D L.rank L'.rank) · x)) (ofAdj L S ((rat τ : Mx D L.rank L'.rank) · y)) = -symplOH S L' x y

theorem **sympl_last** (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (hτ : τᴴ · (L.gram · τ) = (-(D.host.xi0 * star D.host.xi0)) • L'.gram) (x y : WO S L') : symplOH L S (ofAdj L S ((rat τ : Mx D L.rank L'.rank) · x)) (ofAdj L S ((rat τ : Mx D L.rank L'.rank) · y)) = symplOH S L' x y

theorem **mt_inv_ofAdj** (L W : HSp D) (n : GA D W) (z : Mx D L.rank W.rank) : mt W n⁻¹ · ofAdj L W z = ofAdj L W (z · mt W n)

theorem **incAdj_eq** {W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (L : HSp D) (y : Mx D L.rank ρ) : d.incAdj L y = ofAdj L W (y · d.pX)

theorem **projection_mul_ofAdjoint** {W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (L : HSp D) (z : Mx D L.rank W.rank) : d.p0 · ofAdj L W z = ofAdj L W₀ (z · d.j0)

theorem **tau_zEll** {ρ σ : ℕ} {B : BesLev D σ} {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') : (rat l.τ : Mx D B.L.rank L'.rank) · zEll L' ρ ζ' = yZero B.L ρ B.ζ

theorem **link_point** {ρ σ : ℕ} {B : BesLev D σ} {L' W W₀ : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (d : HypDec W W₀ ρ) (n : GA D W) : ptT d B.L (yZero B.L ρ B.ζ) n = l.κ W₀ (ptS d L' (zEll L' ρ ζ') n)

theorem **link_centre** {ρ σ : ℕ} {B : BesLev D σ} {L' W W₀ : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (hτ : l.τᴴ · (B.L.gram · l.τ) = (D.host.xi0 * star D.host.xi0) • L'.gram) (d : HypDec W W₀ ρ) (n : GA D W) : cenT d B.L (yZero B.L ρ B.ζ) n = -cenS d L' (zEll L' ρ ζ') n

theorem **link_simil** {σ : ℕ} {B : BesLev D σ} {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (h : IsLineVec L' ζ') (l : Link B L' ζ') : l.τᴴ · (B.L.gram · l.τ) = (D.host.xi0 * star D.host.xi0) • L'.gram

Development.AbelianVarietyTheory.Definitions.LatticeFrames.MemLineIff

theorem **TopPrim.line** (H : TopPrim v o) {n : ℕ} (h : n = D.R.m) {x : GlobalTheory.FinPl.Pl D} (hx : x ∉ v.tP) {LT : Lat c.T} {LZ : Lat c.Lz} (ho : OrthAt (c.oT n h) (c.x n).L LT LZ x) {a : AE D} (ha : a ∈ idAt D (· = x)) : a • ratV o.ζ₀ ∈ latV c.Lz LZ (· = x) ↔ a ∈ intAt D (· = x)

theorem **PlacedPrim.placed** (h : PlacedPrim f B) : Placed f B

theorem **InPosition.placed** (h : InPosition f B) : Placed f B

Development.AbelianVarietyTheory.Definitions.LatticeFrames.MiPinned0OfGap

theorem **finX_of_gap** : Formalism.L2.SG.PoincareAmplitudeLowerBounds.FinXC

theorem **World0.ofGap_C** (w : World0 I l Z) (h : GapHyp (I.rec7 l Z)) : (w.ofGap h).C = w.C

theorem **World0.ofGap_toSTies** (w : World0 I l Z) (h : GapHyp (I.rec7 l Z)) : (w.ofGap h).S.toSTies = w.S.toSTies

theorem **World0.bare_ofGap** (w : World0 I l Z) (h : GapHyp (I.rec7 l Z)) : (w.ofGap h).bare = w

theorem **World0.of_forall_world** {Q : World0 I l Z → Prop} (hQ : ∀ w : World I l Z, Q w.bare) (h : GapHyp (I.rec7 l Z)) (w : World0 I l Z) : Q w

theorem **miPinned0_of_gap** {u : (I.rec7 l Z).what₂.Word} (hQ : ∀ w : World I l Z, MIPinned w u) (h : GapHyp (I.rec7 l Z)) (w : World0 I l Z) : MIPinned0 w u

theorem **trueD_iff_of_gap** {D : D7} (h : GapHyp D) : TrueD D ↔ IsTrueColumnDatum D

theorem **IsTrueColumnDatum.of_forall_trueD** {Q : D7 → Prop} (hQ : ∀ D, TrueD D → Q D) {D : D7} (hD : IsTrueColumnDatum D) (h : GapHyp D) : Q D

Development.AbelianVarietyTheory.Definitions.LatticeFrames.MiPinnedWithFinIff

theorem **ontoAll** (h : W.Onto L.P) : OntoAll D u L.θ

theorem **TrueD.elim** {Q : D7 → Prop} (h : ∀ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)), World I l Z → Q (I.rec7 l Z)) {D : D7} (hD : TrueD D) : Q D

theorem **TrueD.sTies** {D : D7} (hD : TrueD D) : ∃ S : STies D, FinX S.Y ∧ Formalism.L2.SG.Pf.LocCP.LocTied D

theorem **TrueD.bare** {D : D7} (hD : TrueD D) : IsTrueColumnDatum D

theorem **trueD_iff** {D : D7} (h : ∀ T : STies D, FinX T.Y) : TrueD D ↔ IsTrueColumnDatum D

theorem **MITower.ker** (h : MITower D u) : MIKerTower D u

theorem **kerIncl_of_tower** (h : MIKerTower D u) (h' : OntoTower D) : KerIncl D u

theorem **miPinned_withFin_iff** {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : GlobalTheory.Mem.Col2.TC B X} {l : I.Laws} {Z : GlobalTheory.Mem.Col2.TC.Later B (I.cols l)} (w : World0 I l Z) (hF : FinX w.S.Y) (u : (I.rec7 l Z).what₂.Word) : MIPinned (w.withFin hF) u ↔ MIPinned0 w u

Development.AbelianVarietyTheory.Definitions.LatticeFrames.NonemptyOffDatOfTrueD

theorem **mem_GinfC_of_all** (hP : ∀ v, P v) {S : HSp D} {g : GA D S} (hg : g ∈ grpOff S P) : g ∈ GinfC S

Development.AbelianVarietyTheory.Definitions.LatticeFrames.NotBadPrimeBase

theorem **idx_top** {j m : ℕ} (h : j + 2 * 0 + 2 = m) : m - 1 = j + 1

theorem **GeoMeas.cIOf_ne_zero** (h : GeoMeas R g) (t : T) (j : ℕ) (hj : j < D.R.m) : cIOf R P g t j ≠ 0

theorem **LevG.one_mem** {x : GA D D.sp.X'} {J : Set (GA D D.sp.X')} {χ : GA D D.sp.X' → ℂ} (h : LevG K x J χ) : 1 ∈ J ∧ χ 1 = 1

theorem **levG_one_iff** (K₀ : ClsPTied u) (J : Set (GA D D.sp.X')) (χ : GA D D.sp.X' → ℂ) : LevG K₀.toG 1 J χ ↔ GlobalTheory.Core.Rk.Top.TopLev K₀ J χ

theorem **readsG_zero** {S : D.what₂.Tau → SPar} {MX : DSModel D.sp.X' S} (eW : Pm.Wrow ≃ₗ[ℂ] (M.clsP → FrP D → ℂ)) (j : SmT MX →ₗ[ℂ] (FrP D → ℂ)) (x : GA D D.sp.X') : ReadsG K eW j x 0 0

theorem **not_badPrime_base** (W : RealPack ex) (u : D.what₂.Word) (p : ℕ) : ¬ badPrime W u W.P.A3.A.μ0 p

theorem **exists_badPrime** (W : RealPack ex) (u : D.what₂.Word) (t : W.P.T) (h : ((pk W).Q t).Nonempty) : ∃ p : ℕ, p.Prime ∧ badPrime W u t p

theorem **badPrime_notMem** (W : RealPack ex) (u : D.what₂.Word) (w : HeightOneSpectrum (𝒪 D.what₂.LL)) (p : ℕ) (hp : p.Prime) (h : badPrime W u (W.Q.Rl.mu w) p) : (p : 𝒪 D.what₂.LL) ∉ w.asIdeal

theorem **reads_zero** (W : RealPack ex) (u : D.what₂.Word) (j : SmT (pk W).MX →ₗ[ℂ] (FrP D → ℂ)) (x : GA D D.sp.X') (μ : D.what₂.Tup) : Reads W u j x μ 0 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.NotInhAbsOfGamLe

theorem **TrueE.trueD** {D : D7} (h : TrueE D) : TrueD D

theorem **TrueE.elim** {Q : D7 → Prop} (h : ∀ (B : Base) (X : PreX B.toPre) (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z), IdE w.S.X → Q (I.rec7 l Z)) {D : D7} (hD : TrueE D) : Q D

theorem **Row.of_noE** {Q : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} (h : Row TrivialRankCondition Q) : Row E Q

theorem **Row.and** {Q : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} {Q' : ∀ D : D7, D.what₂.Word → Prop} (h : Row E Q) (h' : Row E fun D u _ _ => Q' D u) : Row E fun D u L C => Q D u L C ∧ Q' D u

theorem **rowKer_of** (h : RowMI E) (h' : RowOnto E) : RowKer E

theorem **rkFam_of** (h : RKRowG E) : RKFam E

theorem **Row.of_imp** {E' : D7 → ℝ → ℝ → Prop} (hE : ∀ D L C', E' D L C' → E D L C') {Q : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} (h : Row E Q) : Row E' Q

theorem **RKFam.of_imp** {E' : D7 → ℝ → ℝ → Prop} (hE : ∀ D L C', E' D L C' → E D L C') (h : RKFam E) : RKFam E'

theorem **Inh.of_imp** {E' : D7 → ℝ → ℝ → Prop} (hE : ∀ D L C', E D L C' → E' D L C') {P : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} (h : Inh E P) : Inh E' P

theorem **Inh.reach** {P Q : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} (h : Inh E P) (h' : Row E fun D u L C => P D u L C → Q D u L C) : ∃ (D : D7) (L C : ℝ), TrueE D ∧ 1 ≤ C ∧ ∀ u, Q D u L C

theorem **Inh.not_row_false** {P : ∀ D : D7, D.what₂.Word → ℝ → ℝ → Prop} (h : Inh E P) : ¬ Row E fun D u L C => P D u L C → False

theorem **InhG.rk** (h : InhG E) (h' : RKRowG E) : ∃ D : D7, TrueE D ∧ ∀ u, RK D u

theorem **GapE.nodeGap** {D : D7} {L C' : ℝ} (h : GapE D L C') : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D

theorem **GapE.of_params** {D : D7} {large₂ : SizeData₂ D} (P : ParameterChoice' D large₂) (L C' : ℝ) : GapE D L C'

theorem **one_le_mfL** {D : D7} {L C' Cl Cg cb Cu cT CT : ℝ} (hH : Deg.HypL D L C' Cl Cg cb Cu cT CT) : 1 ≤ mfL D L ∧ 16 ≤ D.R.k ∧ 1 ≤ D.R.f

theorem **LedE.gamLe** {D : D7} {L C' : ℝ} (h : LedE D L C') : GamLe D L C'

theorem **GamLe.le_k** {D : D7} {L C' Cl Cg cb Cu cT CT : ℝ} (h : GamLe D L C') (hH : Deg.HypL D L C' Cl Cg cb Cu cT CT) (u : D.what₂.Word) : 8 * Real.log (D.mem u).A.GamP ≤ D.R.k

theorem **GamLe.log_lt** {D : D7} {L C C' Cl Cg cb Cu cT CT : ℝ} (h : GamLe D L C') (hH : Deg.HypL D L C' Cl Cg cb Cu cT CT) (hC : 1 ≤ C) (u : D.what₂.Word) : Real.log (D.mem u).A.GamP < C * D.R.f * D.R.k * L

theorem **RKRowAbs.of_gamLe** (hE : ∀ D L C', E D L C' → GamLe D L C') : RKRowAbs E

theorem **not_inhAbs_of_gamLe** (hE : ∀ D L C', E D L C' → GamLe D L C') : ¬ InhAbs E

theorem **GamLe.code_room** {D : D7} {L C' Cl Cg cb Cu cT CT : ℝ} (h : GamLe D L C') (hH : Deg.HypL D L C' Cl Cg cb Cu cT CT) (u : D.what₂.Word) : C' * (D.mem u).A.GamP ^ 2 ≤ 2 ^ (D.R.k - 3)

theorem **CodeE.one_lt_card** {D : D7} {L C' : ℝ} (h : CodeE D L C') (hC : 1 ≤ C') : 1 < Fintype.card D.what₂.Word

theorem **RowsInh.gap** (h : RowsInh) : InhG GapE ∧ InhG TrivialRankCondition ∧ InhG LedE

Development.AbelianVarietyTheory.Definitions.LatticeFrames.NotLineLatAtZero

theorem **tmul_mul_incA** (α : AF D.host.F) (x z : D.host.E) : (α ⊗ₜ[D.host.F] x : AE D) * incA D z = α ⊗ₜ[D.host.F] (x * z)

theorem **latV_lineLat** (L : HSp D) (ζ : Fin L.rank → D.host.E) (𝔠 : FI D) (v : Pl D) : latV L (lineLat L ζ 𝔠) (· = v) = (idV 𝔠 v).map (LinearMap.toSpanSingleton (AE D) (Fin L.rank → AE D) (ratV ζ)).toAddMonoidHom

theorem **lineLatAt_lineLat** (L : HSp D) {ζ : Fin L.rank → D.host.E} (hζ : ζ ≠ 0) (𝔠 : FI D) (v : Pl D) : LineLatAt L ζ 𝔠 (lineLat L ζ 𝔠) v

theorem **idV_le_idAt** (𝔞 : FI D) (v : Pl D) : ∀ a ∈ idV 𝔞 v, a ∈ idAt D (· = v)

theorem **one_tmul_one_ne** (F A E : Type) [Field F] [Field A] [Field E] [Algebra F A] [Algebra F E] : (1 : A) ⊗ₜ[F] (1 : E) ≠ 0

theorem **idV_ne_bot** {𝔞 : FI D} (h : 𝔞 ≠ 0) (v : Pl D) : idV 𝔞 v ≠ ⊥

theorem **not_lineLatAt_zero** (L : HSp D) {ζ : Fin L.rank → D.host.E} (hζ : ζ ≠ 0) {𝔠 : FI D} (h : 𝔠 ≠ 0) (v : Pl D) : ¬ LineLatAt L ζ 𝔠 (lineLat L ζ 0) v

theorem **AA_ne** (hP : FactPrincipalDatum) : D.host.AA ≠ 0 ∧ D.host.AAbar ≠ 0 ∧ D.host.xi0 ≠ 0

theorem **cols_odd_even** (hP : FactPrincipalDatum) {j σ : ℕ} (h : j + 2 * (σ + 1) + 2 = D.R.m) : (∀ i, i < σ → D.host.AA⁻¹ * bI D j (σ + 1) i = D.host.AA * bI D (j + 1 + 1) σ i) ∧ D.host.AA⁻¹ * bI D j (σ + 1) σ = D.host.AA * cI D (j + 1 + 1)

theorem **norm_link** (x y : D.host.E) (j : ℕ) (T : D.host.E) (hl : x = D.host.xi0 * star D.host.xi0 * y) (hx : x * (D.host.xi0 * star D.host.xi0) ^ (j / 2) = -T) : y * (D.host.xi0 * star D.host.xi0) ^ ((j + 1 + 1) / 2) = -T

theorem **tau_norm** {ζ₀ : Fin a.Lz.rank → D.host.E} {τ} (h : b.Sk ζ₀ τ) (j : ℕ) (hj : j + 2 * 0 + 2 = D.R.m) : star (τ j hj *ᵥ ζ₀) ⬝ᵥ (b.bes j 0 hj).L.gram *ᵥ (τ j hj *ᵥ ζ₀) = -(D.host.xi0 * star D.host.xi0) * (((D.mem u).A.T : ℕ) : D.host.E)

theorem **exists_sc** {k : Type} [Field k] [ValuativeRel k] {ξ c : k} (hξ : valuation k ξ = 1) (hc : valuation k c = 1) (s : ℕ) : ∃ sc : ℕ → k, (∀ j, valuation k (sc j) = 1) ∧ (∀ j, sc (j + 1 + 1) = sc j * ξ) ∧ -(ξ * sc s) = c * ξ ^ ((s + 1 + 1) / 2)

theorem **lineId_of_split** {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M : Lat S) (LA : Lat A) (LL : Lat L) (c : FI D) (v : Pl D) (h : LineLatAt L ζ c LL v) (ho : OrthAt o M LA LL v) : LineId o ζ M c v

Development.AbelianVarietyTheory.Definitions.LatticeFrames.NotRedOfMem

theorem **not_red_of_mem** (ι : ℂ ≃+* ℂ_[p]) (L : PeriodPair) (r : ℂ) {z : ℂ} (h : z ∈ L.lattice) : ¬ Red ι L r z

Development.AbelianVarietyTheory.Definitions.LatticeFrames.NotUnitRatioBase

theorem **levG_exists** (x : GA D D.sp.X') : ∃ J χ, LevG K x J χ ∧ ∀ κ : K.lev.U, ∃ h, HasMat h κ.1 ∧ x⁻¹ * h * x ∈ J

theorem **levG_type** {x : GA D D.sp.X'} {J : Set (GA D D.sp.X')} {χ : GA D D.sp.X' → ℂ} (hL : LevG K x J χ) (f : GA D D.sp.X' → ℂ) (hf : ∀ k ∈ J, ∀ g, f (g * k) = χ k * f g) (g h : GA D D.sp.X') (κ : K.lev.U) (hk : HasMat h κ.1) : f (g * h * x) = star ((K.lev.chi κ : ℂˣ) : ℂ) * f (g * x)

theorem **y_inj** : Function.Injective fun c => (K.y c).1

theorem **readsG_exists** (G : Pm.Mt) : ∃ f, ReadsG K eW j x G f

theorem **ReadsG.period_eq** {G G' : Pm.Mt} {f : SmT MX →ₗ[ℂ] (GA D D.sp.X' → ℂ)} (h : ReadsG K eW j x G f) (h' : ReadsG K eW j x G' f) (w : Pm.Wrow) (hw : ∀ c, eW w c ∈ LinearMap.range j) : Pm.period w G = Pm.period w G'

theorem **not_readsG_zero** {G : Pm.Mt} (w : Pm.Wrow) (hw : ∀ c, eW w c ∈ LinearMap.range j) (hG : Pm.period w G ≠ 0) : ¬ ReadsG K eW j x G 0

theorem **ReadsG.scalar** {G : Pm.Mt} {f : SmT MX →ₗ[ℂ] (GA D D.sp.X' → ℂ)} {a : ℂ} (h : ReadsG K eW j x G f) (h' : ReadsG K eW (a • j) x G f) (w : Pm.Wrow) (hw : ∀ c, eW w c ∈ LinearMap.range j) (hG : Pm.period w G ≠ 0) : a = 1

theorem **chars_agree** {t : T} {r : RowCh R P u g t} (h : r.Is) : (∀ j ρ hj, ((r.sBF j ρ hj).χV, (r.sBF j ρ hj).χW) = Lift.chrStep (R.toLevCh.mu t) R.eta (j + 1)) ∧ ∀ j σ hj, ((r.sFB j σ hj).χV, (r.sFB j σ hj).χW) = Lift.chrStep (R.toLevCh.mu t) R.eta (j + 1 + 1)

theorem **isStepVector_trivial_of_chain** (hv : R.chain_vY) (t : T) (j : ℕ) (hj : j + 1 < D.R.m) : IsStepVecOf R.ψ R.θ (Yc D j) (Yc D (j + 1)) (cart (R.MY j)) (cart (R.MY (j + 1))) (R.ΛY j) (R.ΛY (j + 1)) (R.βY j) (R.βY (j + 1)) (R.datYt t j R.triv) (R.chi (j + 1)) (R.Q t) (R.xiR t) (R.vac j) (R.vY t j)

theorem **GeoMeas.common** (h : GeoMeas R g) (t t' : T) : ENNReal.ofReal (g.κ D.R.m * volLev R g t (D.R.m - 1)) • R.νX t = ENNReal.ofReal (g.κ D.R.m * volLev R g t' (D.R.m - 1)) • R.νX t'

theorem **not_geoMeas** (t : T) (h0 : R.νX t = 0) : ¬ GeoMeas R g

theorem **colCh_nonempty** (C : ColumnChain D.R D.what₂.Tau (D.hcW u)) : ∃ k : ColCh u C, ∀ j, Nonempty (k.Ch j)

theorem **geo_nonempty** (bes : ∀ j ρ, j + 2 * ρ + 2 = D.R.m → BesAt (Yc D j) (D.sp.X j) ρ) (jac : ∀ j ρ, j + 2 * ρ + 2 = D.R.m → JacAt (Yc D (j + 1)) (D.sp.X (j + 1)) ρ) (lnk : ∀ j σ h v, Link ((bes j (σ + 1) h).lev v) (bes (j + 1 + 1) σ (idx_step h)).L (bes (j + 1 + 1) σ (idx_step h)).ζ) (Lz : HSp D) (o : OrthDec (D.sp.X D.R.m) D.sp.X' Lz) : Nonempty (Geo D)

theorem **reads_exists** (u : D.what₂.Word) (j : SmT (pk W).MX →ₗ[ℂ] (FrP D → ℂ)) (x : GA D D.sp.X') (μ : D.what₂.Tup) (G : D.what₂.Amb μ) : ∃ f, Reads W u j x μ G f

theorem **tailOf_eq** (u : D.what₂.Word) (j : SmT (pk W).MX →ₗ[ℂ] (FrP D → ℂ)) (x : GA D D.sp.X') (μ : D.what₂.Tup) (h : μ ≠ D.what₂.mu0) (ψ : (D.chTw u μ).cls D.R.m → (D.chTw u μ).Y D.R.m → ℂ) (G : D.what₂.Amb μ) (f : SmT (pk W).MX →ₗ[ℂ] (GA D D.sp.X' → ℂ)) (hG : Reads W u j x μ G f) (hψ : ∀ c, (W.R.T.tw.pt.val u μ h).eW ((D.PTw u μ).lowerMap ((D.PTw u μ).evz ψ)) c ∈ LinearMap.range j) : tailOf W u j x μ ψ f = (D.PTw u μ).Ptil ψ ((D.eMtTw u μ).symm G)

theorem **not_topReadAt_zero** (u : D.what₂.Word) (x : GA D D.sp.X') : ¬ TopReadAt W u 0 x

theorem **not_unitRatio_base** (u : D.what₂.Word) (p : ℕ) (hp : p.Prime) : ¬ UnitRatio (badPrime W u W.P.A3.A.μ0) (p : ℂ) 1

Development.AbelianVarietyTheory.Definitions.LatticeFrames.OnFramesRdApply

theorem **inN_rank** (θ : InfFam D) (V : HSp D) (B) (g : GA D V) : inN θ V B rfl g = inB θ V B g

theorem **castP_rank** {p q : ℕ} (P : PolF D q p) : castP (D

theorem **castP_bijective** {p q a b : ℕ} (hp : p = a) (hq : q = b) : Function.Bijective (castP (D

theorem **RealC.toD_r** {c : Comm D} (ρ : RealC c) (j hj g) : ρ.toD.r j hj g = c.r j g

theorem **RealC.toD_rT** {c : Comm D} (ρ : RealC c) (g) : ρ.toD.rT g = inB c.θ c.T ρ.BT g

theorem **RealC.Is.toD** {c : Comm D} {ρ : RealC c} (h : ρ.Is) : ρ.toD.Is where

theorem **ClsDat.Is.toK** {c : Comm D} {t : Tup c} {A : ClsDat t} {ρ : RealC c} (hA : A.Is ρ) : (A.toK ρ).Is

theorem **ClsDat.Is.toT** {c : Comm D} {t : Tup c} {A : ClsDat t} {ρ : RealC c} (hA : A.Is ρ) : (A.toT ρ).Is

theorem **ClsPin.toD** {c : Comm D} {𝒯 : Type} {t : 𝒯 → Tup c} (h : ClsPin t) : ClsPinD t

theorem **Tup.onFrames_rd** (t : Tup c) {j : ℕ} (hj : j ≤ D.R.m) (β ψ) (z : FrN D (N j)) : (t.onFrames fr).rd j β ψ z = t.rd j β ψ ((fr j hj).symm z)

theorem **Tup.onFrames_rd_apply** (t : Tup c) {j : ℕ} (hj : j ≤ D.R.m) (β ψ) (y : c.Fr j) : (t.onFrames fr).rd j β ψ (fr j hj y) = t.rd j β ψ y

theorem **Comm.onFrames_is** (N : ℕ → ℕ) : (c.onFrames N).Is ↔ c.Is

theorem **Tup.onFrames_is** (t : Tup c) : (t.onFrames fr).Is ↔ t.Is

theorem **KerX.of_onFrames** (h : KerX fun s => (t s).onFrames fr) : KerX t

theorem **PerX.of_onFrames** (h : PerX fun s => (t s).onFrames fr) : PerX t

theorem **RdX.of_onFrames** (h : RdX fun s => (t s).onFrames fr) : RdX t

theorem **ClsPinF.laws** {c : Comm D} {𝒯 : Type} {t : 𝒯 → Tup c} (hc : c.Is) (ht : ∀ s, (t s).Is) (h : ClsPinF t) : ((∀ (c : Comm D) (t : 𝒯 → Tup c), c.Is → (∀ s, (t s).Is) → ClsPinD t → KerX t) → KerX t) ∧ ((∀ (c : Comm D) (t : 𝒯 → Tup c), c.Is → (∀ s, (t s).Is) → ClsPinD t → PerX t) → PerX t) ∧ ((∀ (c : Comm D) (t : 𝒯 → Tup c), c.Is → (∀ s, (t s).Is) → ClsPinD t → RdX t) → RdX t)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.OntoOfSurjectiveDescents

theorem **multT_anti** {V : HSp D} {S : D.what₂.Tau → SPar} (θ : InfFam D) (M : DSModel V S) {J J' : Set (GA D V)} {χ χ' : GA D V → ℂ} (hJ : J ⊆ J') (hχ : ∀ k ∈ J, χ k = χ' k) : MultT θ M J' χ' ≤ MultT θ M J χ

theorem **multT_withP_le** {V : HSp D} {S : D.what₂.Tau → SPar} (θ : InfFam D) (M : DSModel V S) (β : GL (Fin V.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ V.rank) (P : LevelP L.Pg L.r ℂ) : MultT θ M (levS V β (withP L P)) (levChi V β (withP L P)) ≤ MultT θ M (levS V β (flag1L L)) (levChi V β (flag1L L))

theorem **cp_le** (P : W.Lv) (j : ℕ) : W.CP P j ≤ W.CC j

theorem **cxp_le** (P : W.Lv) : W.CXP P ≤ W.CX

theorem **cxp_triv** : W.CXP W.triv = W.CX

theorem **exists_isDn** (P : W.Lv) {T} (hT : T ∈ W.CXP P) : ∃ T₀ ∈ W.CC 0, W.IsDn T T₀

theorem **isDn_unique** {T T₀ T₁} (h : W.IsDn T T₀) (h' : W.IsDn T T₁) : T₀ = T₁

theorem **dnLev_triv** : W.DnLev W.triv

theorem **onto_triv_iff** : W.Onto W.triv ↔ W.Onto1

theorem **dnFrom_surj** (n : ℕ) (h : ∀ k < n, Function.Surjective (W.ioY k)) : Function.Surjective (Lift.dnFrom W.CC W.ioY n)

theorem **onto_of_surjective_descents** (h : W.StepsOnto) : W.Onto1

theorem **onto_iff_bot** (h : W.ioX = 0) (P : W.Lv) : W.Onto P ↔ W.CP P 0 = ⊥

theorem **xi_one** (h : W.Pre) : W.xiH () = 1

theorem **Q_empty** (h : W.Pre) : W.Q () = ∅

theorem **vX_ne** (h : W.Pre) (τ : D.what₂.Tau) : W.vX ≠ 0

theorem **dnFrom_congr** (io io' : ∀ k, W.CC (k + 1) →ₗ[ℂ] W.CC k) (n : ℕ) (h : ∀ k < n, io k = io' k) : Lift.dnFrom W.CC io n = Lift.dnFrom W.CC io' n

theorem **nonempty_tower** (D : D7) (L : LevCh D Unit) (hY : ∀ j, Nonempty (DSModel (Yc D j) (parB (Ptw D) (min j D.R.m)))) (hX : Nonempty (DSModel D.sp.X' (parB (Ptw D) D.R.m))) : ∃ W : Tower D, W.toLevCh = L ∧ IsInfFam W.θ ∧ IsStdChar D W.ψ (-1) ∧ ¬ W.Pre

theorem **levAt_is** {L : LevCh D T} {t : T} (h : L.Is t) : (levAt L t).Is ()

theorem **isIo_ofEq** {G H S Sinf Ds Dt : Type} [MeasurableSpace H] [AddCommGroup Sinf] [Module ℂ Sinf] [AddCommGroup Ds] [Module ℂ Ds] [AddCommGroup Dt] [Module ℂ Dt] {ν : Measure H} {ℱ : Set H} {th : G → H → S → ℂ} {pt : Sinf → S} {It : Sinf →ₗ[ℂ] Ds →ₗ[ℂ] Dt} {C C' : Submodule ℂ (Ds →ₗ[ℂ] (H → ℂ))} {Ct : Submodule ℂ (Dt →ₗ[ℂ] (G → ℂ))} {io : C →ₗ[ℂ] Ct} (e : C' = C) (h : Lift.IsIo ν ℱ th pt It C Ct io) : Lift.IsIo ν ℱ th pt It C' Ct (io ∘ₗ (LinearEquiv.ofEq _ _ e).toLinearMap)

theorem **ofReal_is** (R : RealDat D A Ks) (hT : R.TopMult) (hl : R.toLevCh.Is A.μ0) (hm : R.IsModels) (hc : R.IsChain) : (ofReal R hT A.μ0).Is

theorem **exists_is_of_char** (R : RealDat D A Ks) (F : Face D A) (h : R.Char F) : ∃ W : Tower D, W.Is

theorem **ofReal_isDn** (R : RealDat D A Ks) (hT : R.TopMult) (t : T) (x : (R.HX t).H) : (ofReal R hT t).IsDn (R.cX.toMap R.prX ((R.HX t).e x)) (R.thDn t x)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.PairLeIffPairHtLe

theorem **nrm_nonneg** (A : HLat K) (v : InfinitePlace K) (x : A.V) : 0 ≤ nrm A v x

theorem **Norming.nrm_le** {B M : HLat K} {pr : B.V →ₗ[K] Module.Dual K M.V} (hn : Norming B M pr) (w : B.V) (v : InfinitePlace K) {C : ℝ} (hC : 0 ≤ C) (h : ∀ y, v (pr w y) ≤ C * nrm M v y) : nrm B v w ≤ C

theorem **Reads.exists_nrm_le** {N B A M : HLat K} {Ψ : N.V →ₗ[K] B.V} {Φ : A.V →ₗ[K] Module.Dual K M.V} (hr : Reads N B A M Ψ Φ) : ∃ e : N.V →ₗ[K] A.V, IsoAll N A e ∧ ∀ x v (C : ℝ), 0 ≤ C → (∀ y, v (Φ (e x) y) ≤ C * nrm M v y) → nrm B v (Ψ x) ≤ C

theorem **valLe_of_reads** {N B A M : HLat K} {Ψ : N.V →ₗ[K] B.V} {Φ : A.V →ₗ[K] Module.Dual K M.V} {h : ℝ} (hr : Reads N B A M Ψ Φ) (hp : PairLe A M Φ h) : ValLe N B Ψ h

theorem **dot_apply** (x y : Fin n → K) : dot K n x y = ∑ i, x i * y i

theorem **nrm_sq** (v : InfinitePlace K) (x : Fin n → K) : nrm (lat K n) v x ^ 2 = ∑ i, ‖v.embedding (x i)‖ ^ 2

theorem **norming_dot** [IsGalois ℚ K] : Norming (lat K n) (lat K n) (dot K n)

theorem **reads_dot** [IsGalois ℚ K] {m : ℕ} (Ψ : (lat K n).V →ₗ[K] (lat K m).V) : Reads (lat K n) (lat K m) (lat K n) (lat K m) Ψ (dot K m ∘ₗ Ψ)

theorem **pairLe_dot** : PairLe (lat K n) (lat K n) (dot K n) 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.PinColTOfPinColL

theorem **live_id** {Ch₀ : ℕ → Type} (A : TwDat I l Z u a Ch₀) : A.Live (fun _ => id)

theorem **live_reached** {Ch₀ : ℕ → Type} (A : TwDat I l Z u a Ch₀) : A.Live (Ch

theorem **reached_mem** {Ch₀ Ch : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ι : ∀ j, Ch j → Ch₀ j} (h : A.Live ι) : ∀ j (β : Ch₀ j), Reached A j β → β ∈ Set.range (ι j) | 0, β, _ => h.first β | j + 1, β, Or.inl e => by have key : ∀ n, B.R.m = n → Function.Surjective (ι n)

theorem **ramL_first** {Ch₀ Ch : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ι : ∀ j, Ch j → Ch₀ j} (h : A.Live ι) (φ) : A.RamL ι 0 φ ↔ RamT 0 φ

theorem **ramL_last** {Ch₀ Ch : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ι : ∀ j, Ch j → Ch₀ j} (h : A.Live ι) (φ) : A.RamL ι B.R.m φ ↔ RamT B.R.m φ

theorem **pinColL_of_pinColT** {Ch₀ Ch : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ι : ∀ j, Ch j → Ch₀ j} (hι : A.Live ι) {ψ θ x vac c₀ kom dd w Ram rd cK} (h : PinColT A ψ θ x vac c₀ kom dd w Ram rd cK) : PinColL A ι ψ θ x vac c₀ kom (fun j α γ => dd j (ι j α) (ι (j + 1) γ)) (fun j α γ => w j (ι j α) (ι (j + 1) γ)) (fun j => {φ | A.RamL ι j φ}) (fun j β => rd j (ι j β)) cK

theorem **pinColT_of_pinColL** {Ch₀ : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ψ θ x vac c₀ kom dd w Ram rd cK} (h : PinColL A (fun _ => id) ψ θ x vac c₀ kom dd w Ram rd cK) (hc : ∀ j, j < B.R.m → c₀ j ≠ 0) (hv : ∀ j, j < B.R.m → ∀ α, volLev (x j).vol (A.U j α) ≠ 0) : ∃ w', PinColT A ψ θ x vac c₀ kom dd w' Ram rd cK

theorem **w_ne_zero** {Ch₀ Ch : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ι : ∀ j, Ch j → Ch₀ j} {ψ θ x vac c₀ kom dd w Ram rd cK} (h : PinColL A ι ψ θ x vac c₀ kom dd w Ram rd cK) (j) (hj : j + 1 ≤ B.R.m) (hK : cK j ≠ 0) (α γ) (hz : ∃ z, A.xi j hj (ι j α) (ι (j + 1) γ) z ≠ 0) : w j α γ ≠ 0

theorem **pinTopL_withTop** (h : IsEmbTop (D

theorem **pinChL_withTop** (ε : ℂ) {c₀} {Ch₀ : ℕ → Type} {A : TwDat I l Z u a Ch₀} {ι w} (h : PinChL R g k c₀ A ι w) : PinChL R g (withTop R g P t k ε) c₀ A ι w

theorem **pinTopL_lev** {UT ε} (h : PinTopL R g P (t

theorem **top_subsingleton** {Ch₀ : ℕ → Type} {A : TwDat I l Z u a Ch₀} (h : A.Is) (β β' : Ch₀ B.R.m) : β = β'

theorem **gram_pos** {E : Type} [Field E] [NumberField E] [StarRing E] (hs : ∀ x : E, IsIntegral ℤ x → IsIntegral ℤ (star x)) (n : ℕ) : (Examples₂.posSpace hs n).gram = 1

theorem **rk_top** (D : D7) : D.sp.X'.rank + 1 = (D.sp.X D.R.m).rank

theorem **iXm_eq** (hi : ∀ (a : Fin D.sp.X'.rank) (b : Fin (D.sp.X D.R.m).rank), (a : ℕ) = b → D.sp.iX' (D.sp.X'.frame a) = (D.sp.X D.R.m).frame b) (b a) : iXm D b a = if (a : ℕ) = b then 1 else 0

theorem **isEmbTop_topDec** (hG : (D.sp.X D.R.m).gram = 1) (hG' : D.sp.X'.gram = 1) (hi : ∀ (a : Fin D.sp.X'.rank) (b : Fin (D.sp.X D.R.m).rank), (a : ℕ) = b → D.sp.iX' (D.sp.X'.frame a) = (D.sp.X D.R.m).frame b) (c : D.host.E) : IsEmbTop (D

theorem **exists_emb** (hI : I.Ok) (hP : TC.Pins B X) : ∃ (Lz : HSp (I.rec7 l Z)) (oT : ∀ n, n = (I.rec7 l Z).R.m → GlobalTheory.MixModel.OrthDec ((I.rec7 l Z).sp.X n) (I.rec7 l Z).sp.X' Lz), IsEmbTop (D

theorem **exists_pinTopL** (hI : I.Ok) (hP : TC.Pins B X) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (k : ColCh (D

Development.AbelianVarietyTheory.Definitions.LatticeFrames.PlacedP

theorem **PlacedP.placed** (h : PlacedP f B) : Placed f B

Development.AbelianVarietyTheory.Definitions.LatticeFrames.QuasiOfFam

theorem **quasi_of_fam** {S L : HSp D} {Y : GA D S → WO S L → ℂ} (hY : IsQuasiFam ψ S L Y) (h : GA D S) : Quasi ψ S L (Y h)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.RankConditionRankOneIffEqZero

theorem **sameTops_iff₂** {D D' : D7} (L : SizeData₂ D) (L' : SizeData₂ D') (hk : D.R.k = D'.R.k) : SameTops D D' ↔ ∀ (t : D.host.TT) (t' : D'.host.TT), (t : D.R.A).val = (t' : D'.R.A).val → L.aT t = L'.aT t' ∧ L.bT t = L'.bT t'

theorem **Pts.ofFam_covered** {ε : ℝ} (F : FamCore ε) : (Pts.ofFam F).Covered

theorem **Pts.ofFam_rkWin** {ε : ℝ} (F : FamCore ε) (h : (Pts.ofFam F).RKAll) : (Pts.ofFam F).RKWin

theorem **exists_lvP** (N : ℕ) : ∃ p : LvP, N < p.T

theorem **tH** (p : LvP) (t : R0.A) : THyp (Ht p t)

theorem **LvP.ne_two** (p : LvP) : p.T ≠ 2

theorem **LvP.ne_pde** (p : LvP) : p.T ≠ H0.pde ∧ p.T ≠ H0.pde'

theorem **colTo_one** (j : ℕ) : colTo (chainN hc (pkg N)) j (fun _ _ => 1) = fun _ _ => ((N : ℂ) + 1) ^ j

theorem **rk_le_one** : rkC (chainN hc (pkg N)) (atTN hc (pkg N) T Dst hD 1 one_pos 2 one_lt_two) (periodN hc (pkg N) (En 1) (perS N s)) ≤ 1

theorem **rk_zero** : rkC (chainN hc (pkg N)) (atTN hc (pkg N) T Dst hD 1 one_pos 2 one_lt_two) (periodN hc (pkg N) (En 1) (perS N 0)) = 0

theorem **rk_one** : rkC (chainN hc (pkg N)) (atTN hc (pkg N) T Dst hD 1 one_pos 2 one_lt_two) (periodN hc (pkg N) (En 1) (perS N 1)) = 1

theorem **recd_dU** (t c N s u) : dU (recd p t c N s) u = N + 1

theorem **columnRank_eq_one_of_rankOne** (t c N u) : rkC ((recd p t c N 1).mem u).C ((recd p t c N 1).mem u).A ((recd p t c N 1).mem u).P = 1

theorem **columnRank_eq_zero_of_rankZero** (t c N u) : rkC ((recd p t c N 0).mem u).C ((recd p t c N 0).mem u).A ((recd p t c N 0).mem u).P = 0

theorem **rankCondition_rankOne_iff_eq_zero** (t c N u) : RK (recd p t c N 1) u ↔ N = 0

theorem **not_rankCondition_rankZero** (t c N u) : ¬ RK (recd p t c N 0) u

theorem **recd_share** (t c N u) (ρ : ℝ) : RKShare ρ (recd p t c N 1) u ↔ ρ * ((N : ℝ) + 1) ≤ 1

theorem **recd_sameHost** (t c N s p' t' c' N' s') : SameHost (recd p t c N s) (recd p' t' c' N' s')

theorem **recd_sameTops** (t c N s p' t' c' N' s') : SameTops (recd p t c N s) (recd p' t' c' N' s') ↔ c = c'

theorem **recd_sameWord** (t c N s u p' t' c' N' s' u') : SameWord (recd p t c N s) u (recd p' t' c' N' s') u' ↔ t = t' ∧ c = c'

theorem **val_cast** (u₀ : ℕ) (h : u₀ < R0.n) : (R0.ofNat u₀ : R0.A).val = u₀

theorem **exists_const** (a b : ℝ) {ε : ℝ} (hε : 0 ≤ ε) : ∃ C : ℝ, ∀ x : ℝ, 1 ≤ x → a − C * x ^ ε ≤ b ∧ b ≤ C * (2 : ℕ) * x ^ ε

theorem **fam_someAt** (N : ℝ → ℕ) (s : ℂ) {Q : ℝ → ∀ D : D7, D.what₂.Word → Prop} (x₀ : ℝ) (h : ∀ x, x₀ ≤ x → ∀ t, Q x (recd p t 0 (N x) s) ()) : (fam p N s).SomeAt Q

theorem **fam_yes** {ε : ℝ} (hε : 0 < ε) : (fam p (fun _ => 0) 1).RKAll ∧ (fam p (fun _ => 0) 1).RKWin ∧ (fam p (fun _ => 0) 1).RKRoom ε ∧ (fam p (fun _ => 0) 1).RKWeak ε

theorem **fam_zero** (N : ℝ → ℕ) (ε : ℝ) : ¬ (fam p N 0).RKAll ∧ ¬ (fam p N 0).RKWin ∧ ¬ (fam p N 0).RKRoom ε ∧ ¬ (fam p N 0).RKWeak ε

theorem **fam_half** {ε : ℝ} (hε : 0 < ε) : (fam p (fun _ => 1) 1).RKWeak ε ∧ ¬ (fam p (fun _ => 1) 1).RKWin

theorem **fam_over** (p' : LvP) (s : ℂ) {ε : ℝ} (hε : 0 ≤ ε) : (fam p (fun _ => 0) s).RKRoomOver (cls p' 0 1) ε

theorem **fam_not_over** (p' : LvP) (s : ℂ) (ε : ℝ) : ¬ (fam p (fun _ => 0) s).RKRoomOver (cls p' 0 0) ε ∧ ¬ (fam p (fun _ => 0) s).RKRoomOver (cls p' 1 1) ε

theorem **room_used** {ε : ℝ} (hε : 0 ≤ ε) : (∃ (F : Pts) (Tr : ℝ → D7 → Prop), F.RKRoomOver Tr ε ∧ ¬ F.RKWin ∧ ∀ x i u D' u', Tr x D' → (1 + 1 / (((F.D x i).R.f : ℝ) * (F.D x i).R.k)) * ((F.D x i).mem u).A.Xu < ((D'.mem u').A.T : ℝ)) ∧ (∃ (F : Pts) (Tr : ℝ → D7 → Prop), F.RKRoomOver Tr ε ∧ ¬ F.RKWin ∧ ∀ x i u D' u', Tr x D' → ((D'.mem u').A.T : ℝ) < ((F.D x i).mem u).A.Tstar)

theorem **not_over_of_large** {ε : ℝ} (hε : ε < 1) (q : ℝ → LvP) (hq : ∀ x, Real.exp x < (q x).T) : ¬ (fam p (fun _ => 0) 1).RKRoomOver (fun x => cls (q x) 0 1 x) ε

Development.AbelianVarietyTheory.Definitions.LatticeFrames.RatioOKOfRootOfUnity

theorem **served_holds** : Served W

theorem **not_mem_Q_base** (PP : HeightOneSpectrum (𝒪 D.host.E)) : PP ∉ (pk W).Q W.P.A3.A.μ0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.RealSizeNonneg

theorem **realSize_nonneg** (θ : InfFam D) {m n : ℕ} (x : Mx D m n) : 0 ≤ realSize θ x

theorem **maj_nonneg** (θ : InfFam D) {m n : ℕ} (K : Set (Mx D m n)) (N : ℕ) (x : Mx D m n) : 0 ≤ maj θ K N x

theorem **isSobForm_eval** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) : IsSobForm ψ θ S T fun f => f 0 where

Development.AbelianVarietyTheory.Definitions.LatticeFrames.RealisedChainTorusEq

theorem **SameHost.condN_eq** {D D' : D7} (h : SameHost D D') : D.host.condN = D'.host.condN

theorem **SameHost.refl** (D : D7) : SameHost D D

theorem **SameHost.symm** {D D' : D7} (h : SameHost D D') : SameHost D' D

theorem **SameHost.trans** {D D' D'' : D7} (h : SameHost D D') (h' : SameHost D' D'') : SameHost D D''

theorem **SameHost.exists_top** {D D' : D7} (h : SameHost D D') (t : D.host.TT) : ∃ t' : D'.host.TT, (t : D.R.A).val = (t' : D'.R.A).val

theorem **SameTops.refl** (D : D7) : SameTops D D

theorem **SameWord.refl** (D : D7) (u : D.what₂.Word) : SameWord D u D u

theorem **SameWord.exists_place** {D D' : D7} {u : D.what₂.Word} {u' : D'.what₂.Word} (h : SameWord D u D' u') (τ : D.what₂.Tau) : ∃ τ' : D'.what₂.Tau, HEq (D.eTau τ) (D'.eTau τ') ∧ HEq (D.hcW u τ) (D'.hcW u' τ')

theorem **SameWord.sameTops** {D D' : D7} {u : D.what₂.Word} {u' : D'.what₂.Word} (h : SameWord D u D' u') : SameTops D D'

theorem **ne_of_forall** {Ix : ℝ → Type} (h : ∀ x, Nonempty (Ix x)) : ∀ y, ∃ x, y ≤ x ∧ Nonempty (Ix x)

theorem **SomeAt.mono** {Q Q' : ℝ → ∀ D : D7, D.what₂.Word → Prop} (h : F.SomeAt Q) (x₁ : ℝ) (hq : ∀ x, x₁ ≤ x → ∀ i u, Q x (F.D x i) u → Q' x (F.D x i) u) : F.SomeAt Q'

theorem **SomeAt.exists** {Q : ℝ → ∀ D : D7, D.what₂.Word → Prop} (h : F.SomeAt Q) : ∃ x₀ : ℝ, ∀ x, x₀ ≤ x → Nonempty (F.Ix x) → ∃ (i : F.Ix x) (u : (F.D x i).what₂.Word), Q x (F.D x i) u

theorem **SomeAt.exists_ge** {Q : ℝ → ∀ D : D7, D.what₂.Word → Prop} (h : F.SomeAt Q) (y : ℝ) : ∃ x, y ≤ x ∧ ∃ (i : F.Ix x) (u : (F.D x i).what₂.Word), Q x (F.D x i) u

theorem **SomeAt.covered** {Q : ℝ → ∀ D : D7, D.what₂.Word → Prop} (h : F.SomeAt Q) : F.Covered

theorem **RKAll.rkWin** (h : F.RKAll) (hc : F.Covered) : F.RKWin

theorem **RKRoom.rkWin** {ε : ℝ} (h : F.RKRoom ε) : F.RKWin

theorem **RKRoom.over** {ε : ℝ} {Tr : ℝ → D7 → Prop} (h : F.RKRoom ε) (hT : F.Within Tr) : F.RKRoomOver Tr ε

theorem **RKWeak.over** {ε : ℝ} {Tr : ℝ → D7 → Prop} (h : F.RKWeak ε) (hT : F.Within Tr) : F.RKWeakOver Tr ε

theorem **RKRoomOver.mono** {ε : ℝ} {Tr Tr' : ℝ → D7 → Prop} (h : F.RKRoomOver Tr ε) (hT : ∀ x D, Tr x D → Tr' x D) : F.RKRoomOver Tr' ε

theorem **RKWeakOver.mono** {ε : ℝ} {Tr Tr' : ℝ → D7 → Prop} (h : F.RKWeakOver Tr ε) (hT : ∀ x D, Tr x D → Tr' x D) : F.RKWeakOver Tr' ε

theorem **not_rkRoomOver** {ε : ℝ} {Tr : ℝ → D7 → Prop} (hT : ∀ x D u, Tr x D → ¬ RK D u) : ¬ F.RKRoomOver Tr ε

theorem **RK.share** {D : D7} {u : D.what₂.Word} (h : RK D u) {ρ : ℝ} (hρ : ρ ≤ 1 / 2) : RKShare ρ D u

theorem **room_le_and_share_le** {ε : ℝ} (hε : 0 < ε) (C : ℝ) : ∃ x₁ : ℝ, ∀ x, x₁ ≤ x → C * x ^ ε ≤ x ^ (13 * ε) ∧ x ^ (-(12 * ε)) ≤ 1 / 2

theorem **Xu_pos** (D : D7) (u : D.what₂.Word) : 0 < (D.mem u).A.Xu

theorem **f_mul_k_pos** (R : Ranks) : (0 : ℝ) < (R.f : ℝ) * R.k

theorem **T_lt_cap** (D : D7) (u : D.what₂.Word) : ((D.mem u).A.T : ℝ) < (1 + 1 / ((D.R.f : ℝ) * D.R.k)) * (D.mem u).A.Xu

theorem **le_log_alt** (D : D7) (u : D.what₂.Word) : Real.log (D.mem u).A.Xu - 2 * Real.log (D.mem u).A.GamP / ((D.R.f : ℝ) * D.R.k) ≤ Real.log ((D.mem u).A.T : ℝ)

theorem **RKWin.rkRoom** {ε : ℝ} (h : F.RKWin) (CR CT x₁ : ℝ) (hb : ∀ x, x₁ ≤ x → ∀ (i : F.Ix x) u, 2 * Real.log ((F.D x i).mem u).A.GamP ≤ CR * x ^ ε * (((F.D x i).R.f : ℝ) * (F.D x i).R.k) ∧ Real.log (((F.D x i).mem u).A.T : ℝ) ≤ CT * (F.D x i).R.m * x ^ ε) : F.RKRoom ε

theorem **RKRoom.rkWeak** {ε : ℝ} (h : F.RKRoom ε) (hε : 0 < ε) : F.RKWeak ε

theorem **RKRoomOver.rkWeakOver** {ε : ℝ} {Tr : ℝ → D7 → Prop} (h : F.RKRoomOver Tr ε) (hε : 0 < ε) : F.RKWeakOver Tr ε

theorem **hypEx_of_hypAll** (h : HypAll Fam Cl pts Form) (hi : ClsInhabited Fam Cl) : HypEx Fam Cl pts Form

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ReflectionBoostMatrixConjTransposeMulSelf

theorem **exists_inv** (Φ : SB X) (A : AddSubgroup X) (hA : IsCompact (A : Set X)) : ∃ N : ℕ, 0 < N ∧ ∀ a ∈ A, ∀ x, Φ.1 (x + N • a) = Φ.1 x

theorem **exists_supp** {K : Set X} (hK : IsCompact K) (A : AddSubgroup X) (hA : IsOpen (A : Set X)) (h : ∀ x, ∃ n : ℕ, 0 < n ∧ n • x ∈ A) : ∃ N : ℕ, 0 < N ∧ ∀ x ∈ K, N • x ∈ A

theorem **chart_one** {e : AE D} (h1 : e * e = e) {v : HeightOneSpectrum (𝒪 D.host.F)} {π : AE D →+* Fv D.host.F v} (hπ : GlobalTheory.SplitHecke.Chart e v π) : π e = 1

theorem **chart_cont** {e : AE D} (he : IsSplitIdem D e) {v : HeightOneSpectrum (𝒪 D.host.F)} {π : AE D →+* Fv D.host.F v} (hπ : GlobalTheory.SplitHecke.Chart e v π) : Continuous π

theorem **single_add** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (a b : Fv F v) : singleV F v (a + b) = singleV F v a + singleV F v b

theorem **sec_cont** (e : AE D) (v : HeightOneSpectrum (𝒪 D.host.F)) : Continuous (sec e v)

theorem **chart_sec** {e : AE D} (h1 : e * e = e) {v : HeightOneSpectrum (𝒪 D.host.F)} {π : AE D →+* Fv D.host.F v} (hπ : GlobalTheory.SplitHecke.Chart e v π) (a) : π (sec e v a) = a

theorem **sec_chart** {e : AE D} (h1 : e * e = e) {v : HeightOneSpectrum (𝒪 D.host.F)} {π : AE D →+* Fv D.host.F v} (hπ : GlobalTheory.SplitHecke.Chart e v π) (x) : sec e v (π x) = e * x

theorem **exists_idem** (D : D7) : ∃ e, IsSplitIdem D e

theorem **isCompact_half** {e : AE D} (he : IsSplitIdem D e) {S T : HSp D} {ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) : IsCompact {t : WO S T | t ∈ (latO S T ΛS ΛT fun _ => True) ∧ IsHalf e t}

theorem **units_mul_relation_rearrange_commRing** {R : Type} [CommRing R] {n : ℕ} (e e' f : R) (G Gi a a' b b' : Matrix (Fin n) (Fin n) R) (h1 : e * e = e) (h1' : e' * e' = e') (h2 : e * e' = 0) (hf : f = 1 - e - e') (hba : b * a = e • (1 : Matrix (Fin n) (Fin n) R)) (hab' : a' * b' = e' • (1 : Matrix (Fin n) (Fin n) R)) (g1 : ∀ x : Matrix (Fin n) (Fin n) R, Gi * (G * x) = x) (g2 : ∀ x : Matrix (Fin n) (Fin n) R, G * (Gi * x) = x) : (e' • a' + e • (G * b * Gi) + f • 1) * (G * (e • a + e' • (Gi * (b' * G)) + f • 1)) = G

theorem **reflectionBoost_matrix_conjTranspose_mul_self** (e e' f : AE D) (a b : Mx D S.rank S.rank) (h1 : e * e = e) (h2 : e * e' = 0) (c1 : cA D e = e') (hf : f = 1 - e - e') (hba : b * a = e • (1 : Mx D S.rank S.rank)) : adj (e • a + e' • (giA S * (adj b * gA S)) + f • 1) * (gA S * (e • a + e' • (giA S * (adj b * gA S)) + f • 1)) = gA S

Development.AbelianVarietyTheory.Definitions.LatticeFrames.RescOne

theorem **resc_one** (w : VTup t) : resc w (fun _ _ => 1) = w

theorem **QPinU.of** {v : VCom c} {w : VTup t} (h : QPin v w) : QPinU v w

Development.AbelianVarietyTheory.Definitions.LatticeFrames.SameOrbEquivalence

theorem **corner_mul** (g h : GL (n ⊕ Fin 0) k) : (g * h).val.toBlocks₁₁ = g.val.toBlocks₁₁ * h.val.toBlocks₁₁

theorem **corner_one** : (1 : GL (n ⊕ Fin 0) k).val.toBlocks₁₁ = 1

theorem **mulVec_mem_box** {A : Matrix n n k} (hA : IntM A) {c : kˣ} {y : n → k} (hy : y ∈ box n c) : A *ᵥ y ∈ box n c

theorem **sameOrb_equivalence** (J : Subgroup (GL (n ⊕ Fin 0) k)) (hJ : J ≤ KO k _) (c : kˣ) : Equivalence (SameOrb J (box n c))

theorem **lev_zero** (d : ℕ) (h : 0 + d = G.s + 1) : ((G.T δ hδ η).lev 0 d h).J = KO G.k _ ∧ ∀ x, ((G.T δ hδ η).lev 0 d h).lam x = 1

theorem **isOneComb_zero** (ρ d : ℕ) (hρ : 0 + 2 * ρ + 2 = D.R.m) (h : 0 + (d + 1) = G.s + 1) : IsOneComb G.ψk G.ν (ρ + 1) ((G.T δ hδ η).lev 0 (d + 1) h) 1 (Finsupp.single 1 1)

theorem **piA_finite** (hfin : ∀ j d h, (A j d h).Finite) (j : ℕ) : Finite (PiA G A j)

theorem **chT_finite** (hfin : ∀ j d h, (A j d h).Finite) (j : ℕ) : Finite (ChT G δ hδ η A M j)

theorem **mT_spec** (j : ℕ) : (mT G j).IsHaarMeasure ∧ mT G j (KO G.k (XIx (Fin D.R.k) (Fin D.R.t) j) : Set _) = 1

theorem **combOf_spec** (hstd : IsStdChar D c.ψ (-1)) {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame G.k S.rank Y') (FT : Frame G.k T.rank Y) (Ξ : Matrix Y Y' G.k → ℂ) : (ofSplit G.q).IsCombAt c.ψ FS FT (combOf G hstd FS FT Ξ) Ξ

theorem **repT_mem** {D : D7} {c : Comm D} {v : VCom c} {B : GlobalTheory.Fam.Base c} {PP : Formalism.L2.DIV.Pl D.host.E} (G : QGeo v B PP) (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ) (A : RepFam G.k G.s (XIx (Fin D.R.k) (Fin D.R.t))) (M : ℕ) (j d : ℕ) (h : j + d = G.s) (α : ChT G δ hδ η A M j) : repT G δ hδ η A M j d h α ∈ A j d h

theorem **exists_ext_of_reps** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle), c.Is → ∀ (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ), (∀ j, IsUnram (η j)) → (∀ j, j ≤ G.s + 1 → (ofSplit G.q).IsComp (χ j)⁻¹ (η (j + 1) * δ)) → ∀ A, (G.T δ hδ η).IsRepSystem A G.ψk G.ν → ∀ M : ℕ, (G.T δ hδ η).Adapted A (box _ (G.ϖ ^ M)⁻¹) (box _ (G.ϖ ^ M)) → ∃ E : QExt G χ, E.δ = δ ∧ E.η = η ∧ E.A = A ∧ E.M = M ∧ ∀ y, ∃ z ∈ box (XIx (Fin D.R.k) (Fin D.R.t) (G.s + 1)) (G.ϖ ^ E.M)⁻¹, Submodule.Quotient.mk z = E.cos y

Development.AbelianVarietyTheory.Definitions.LatticeFrames.SingleMemGluedIff

theorem **pr_mem_of_intAt** (p : SplitPt D k) {a : AE D} (ha : finPart D a ∈ intAE D) : p.pr a ∈ 𝒪[k]

theorem **eq_sec_add** (p : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = p.v)) : a = p.sec (p.pr a) + cA D (p.sec (p.pr (cA D a)))

theorem **mem_intAt_iff** (p : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = p.v)) : a ∈ intAt D (· = p.v) ↔ p.pr a ∈ 𝒪[k] ∧ p.pr (cA D a) ∈ 𝒪[k]

theorem **single_mem_glued_iff** {n : Type} [DecidableEq n] (q : Unif k) (P : Pair k n) (t : k) : (Pi.single (Sum.inr 0) t : n ⊕ Fin 1 → k) ∈ gluedS q 1 P.ker P.vec ↔ t ∈ 𝒪[k]

theorem **pair_sumFrame** {p : SplitPt D k} {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) {n : Type} [Fintype n] [DecidableEq n] {β : Frame p A n} {β₀ : Frame p S (n ⊕ Fin 1)} (hs : IsSumFrame o ζ β β₀) (x : n ⊕ Fin 1 → k) (t : k) : ∑ i, ∑ j, (β₀.b *ᵥ x) i * p.pr (cA D (incA D (S.gram i j))) * (t * p.pr (incA D (star (vecB o ζ j)))) = t * (p.pr (incA D (star ζ ⬝ᵥ L.gram *ᵥ ζ)) * x (Sum.inr 0))

theorem **line_of_glued** {p : SplitPt D k} {ρ : ℕ} {B : BesLev D ρ} {M : Lat B.V₀} {q : Unif k} {n : Type} [Fintype n] [DecidableEq n] {β : Frame p B.A n} {β₀ : Frame p B.V₀ (n ⊕ Fin 1)} {P : Pair k n} (hM : IsSelfDualC B.V₀ M) (hg : GluedCore B M q β β₀ P) {a : AE D} (ha : a ∈ idAt D (· = p.v)) : a • ratV (vecB B.o B.ζ) ∈ latV B.V₀ M (· = p.v) ↔ a ∈ intAt D (· = p.v)

theorem **isU_of_val** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) {t : D.host.E} (h : valuation k (p.pr (incA D t)) = 1) (h' : valuation k (p.pr (incA D (star t))) = 1) : IsU p.v (incA D t)

theorem **prim_line** {ψ : AddChar (AF D.host.F) ℂ} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} (hL : IsSelfDualC S L) {x : Pl D} (hp : PrimAt ψ d L x) {cc b : (AE D)ˣ} (hψ : ∀ s ∈ idAt D (· = x), (∀ o ∈ intAt D (· = x), psiE ψ (s * o) = 1) ↔ ((cc⁻¹ : (AE D)ˣ) : AE D) * s ∈ intAt D (· = x)) {i j : Fin ρ} (hij : j.1 = i.1 + 1) (hj : ∀ a ∈ idAt D (· = x), colMul d.jX j a ∈ latV S L (· = x) ↔ ((b⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : ∀ a ∈ idAt D (· = x), colMul d.jX i a ∈ latV S L (· = x) ↔ (((cc * b)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **iso_step** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} {LV : Lat B.V} {LV₀ : Lat B.V₀} {LW : Lat J.W} {LW₀ : Lat J.W₀} {LV' : Lat B'.V} {LV₀' : Lat B'.V₀} {x : Pl D} {α m cc : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hψ : ∀ s ∈ idAt D (· = x), (∀ o ∈ intAt D (· = x), psiE ψ (s * o) = 1) ↔ ((cc⁻¹ : (AE D)ˣ) : AE D) * s ∈ intAt D (· = x)) (h₁ : GoodBF ψ B J LV LV₀ LW LW₀ x) (h₂ : AdaptFB ψ J B' LW LW₀ LV' LV₀' x) (hz : ∀ a ∈ idAt D (· = x), a • ratV (vecB B'.o B'.ζ) ∈ latV B'.V₀ LV₀' (· = x) ↔ ((m⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) (n : ℕ) : ∀ i : Fin (σ + 1), i.1 + n = σ → ∀ a ∈ idAt D (· = x), colMul B.d.jX i a ∈ latV B.V LV (· = x) ↔ (((cc ^ n * (α * (α * m)))⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **core_line_all** (hc : c.Is) (hs : ShapeOff v o) (hp : TopPrim v o) (hr : ∀ j, j + 2 * 0 + 2 = D.R.m → 0 < (c.X (j + 1)).rank) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : ∀ j h, v.τ j h *ᵥ o.ζ₀ = lam • (c.bes j 0 h).ζ) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (ρ j : ℕ) (h : j + 2 * ρ + 2 = D.R.m) : ∀ a ∈ idAt

D (· = x), a • ratV (vecB (c.bes j ρ h).o (c.bes j ρ h).ζ) ∈ latV (c.bes j ρ h).V₀ (o.M j ρ h) (· = x) ↔ ((((α ∗ (xiU D ∗ cU α)⁻¹) ^ (ρ + 1) ∗ inU lam hlam)⁻¹ : (AE D)ˣ) : AE D) ∗ a ∈ intAt D (· = x)

theorem **iso_line_all** (hc : c.Is) (hs : ShapeOff v o) (hp : TopPrim v o) (hr : ∀ j, j + 2 ∗ 0 + 2 = D.R.m → 0 < (c.X (j + 1)).rank) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : ∀ j h, v.τ j h ∗ᵥ o.ζ₀ = lam • (c.bes j 0 h).ζ) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hψ : ∀ s ∈ idAt D (· = x), (∀ o ∈ intAt D (· = x), psiE c.ψ (s ∗ o) = 1) ↔ (((xiU D ∗ α ∗ cU α)⁻¹ : (AE D)ˣ) : AE D) ∗ s ∈ intAt D (· = x)) (ρ j : ℕ) (h : j + 2 ∗ ρ + 2 = D.R.m) (i : Fin ρ) : ∀ a ∈ idAt D (· = x), colMul (c.bes j ρ h).d.jX i a ∈ latV (c.V j) (v.N j) (· = x) ↔ ((((xiU D ∗ α ∗ cU α) ^ (ρ − i.1) ∗ ((α ∗ (xiU D ∗ cU α)⁻¹) ^ (ρ + 1) ∗ inU lam hlam))⁻¹ : (AE D)ˣ) : AE D) ∗ a ∈ intAt D (· = x)

theorem **topPrim_iff** {c : Comm D} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} (v : VCom c) (o : TDat c k ι) : TopPrim v o ↔ PrimTop (c

theorem **placedPrim_iff** : PlacedPrim f B ↔ PlacedP f B

theorem **two_rS** (D : D7) : 0 + 2 ∗ D.R.rS + 2 = D.R.m

theorem **rank_pos** (f : FamCh W u) (j : ℕ) (h : j + 2 ∗ 0 + 2 = D.R.m) : 0 < (f.c.X (j + 1)).rank

theorem **mem_moved** {v : VCom f.c} (ht : VTie f v) {x : GlobalTheory.FinPl.Pl D} (hg : ∀ PP ∈ (pk W).Pg, x ≠ PP.under (𝒪 D.host.F)) {y : Fin (Yc D 0).rank → AE D} (hy : ∀ i, y i ∈ idAt D (· = x)) : mt (Yc D 0) (f.g.b 0)⁻¹ ∗ᵥ y ∈ latV (Yc D 0) ((pk W).ΛY 0) (· = x) ↔ y ∈ latV (Yc D 0) (v.N 0) (· = x)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.SingleMemIdAt

theorem **mem_int_iff** {v : Pl D} (a : Fv D.host.F v) : a ∈ 𝒪[Fv D.host.F v] ↔ Valued.v a ≤ 1

theorem **exists_splitPt** (D : D7) : ∃ v : Pl D, Nonempty (SplitPt D (Fv D.host.F v))

theorem **isHermND** (V : HSp D) : IsHermND D.sp.cm.sig V.gram

theorem **gl_injective** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {p : SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : Frame p S n) : Function.Injective β.gl

theorem **eq_single** {v : Pl D} {a : AE D} (ha : a ∈ idAt D (· = v)) : a = singleTens D.host.E v (compAt D v a)

theorem **compAt_eq** (u : Pl D) (z : AE D) : compAt D u z = GlobalTheory.AdelicU.compFin D.host.F D.host.E u z

theorem **single_mem_idAt** (v : Pl D) (y : Fv D.host.F v ⊗[D.host.F] D.host.E) : singleTens D.host.E v y ∈ idAt D (· = v)

theorem **mem_idAt_iff** {v : Pl D} {a : AE D} : a ∈ idAt D (· = v) ↔ algebraMap (AF D.host.F) (AE D) (placeIdem v) ∗ a = a

theorem **mem_grpAt_iff** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (g : GA D S) : g ∈ grpAt S (· = p.v) ↔ SuppAt p.e S g

theorem **tame** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) : Tame D p.v

theorem **latV_le_int** (S : HSp D) (L : Lat S) (hL : ∀ x ∈ L, ∀ i, ∃ b : 𝒪 D.host.E, (b : D.host.E) = x i) (v : Pl D) : latV S L (· = v) ≤ AddSubgroup.pi Set.univ fun _ => intGrp D v

theorem **single_mem_latV** (S : HSp D) (L : Lat S) (v : Pl D) (i : Fin S.rank) (hL : ∀ b : 𝒪 D.host.E, Pi.single i (b : D.host.E) ∈ L) : intAE D ≤ (latV S L (· = v)).comap ((AddMonoidHom.single (fun _ : Fin S.rank => AE D) i).comp ((singleTens D.host.E v).toAddMonoidHom.comp (GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D v).toAddMonoidHom))

theorem **mem_latV_std** (S : HSp D) (L : Lat S) (hL : ∀ x, x ∈ L ↔ ∀ i, ∃ b : 𝒪 D.host.E, (b : D.host.E) = x i) (v : Pl D) (y : Fin S.rank → AE D) : y ∈ latV S L (· = v) ↔ ∀ i, y i ∈ intAt D (· = v)

theorem **isLat_std** {v : Pl D} (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (S : HSp D) (L : Lat S) (hL : ∀ x, x ∈ L ↔ ∀ i, ∃ b : 𝒪 D.host.E, (b : D.host.E) = x i) : (frame τ S 1).IsLat L (Submodule.pi Set.univ fun _ => (1 : Submodule 𝒪[Fv D.host.F v] (Fv D.host.F v)))

theorem **integrand_dual** {V V₀ W : HSp D} {ρ : ℕ} (d : HypDec V V₀ ρ) (ψ : AddChar (AF D.host.F) ℂ) (f : WO V W → ℂ) (x a a' : Mx D W.rank ρ) (w₀ : WO V₀ W) : heisO ψ V W (d.extY W a') (heisO ψ V W (d.extX W x) (heisO ψ V W (d.ext0 W w₀) (heisO ψ V W (d.extY W a) f))) 0 = ψ (symplOH V W (d.extX W x) (d.extY W a) + symplOH V W (d.extX W x) (d.extY W a)) ∗ heisO ψ V W (d.extY W (a' + a)) (heisO ψ V W (d.extX W x) (heisO ψ V W (d.ext0 W w₀) f)) 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.SobContId

theorem **sobCont_id** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) {S T : HSp D} (μ : Measure (WO S T)) (ℱ : Set (WO S T)) : Sobolev₂ ψ θ μ ℱ μ ℱ id

Development.AbelianVarietyTheory.Definitions.LatticeFrames.SphericalEisensteinSeriesOnGroup

theorem **Lvl.K_frameLat** (V : HSp D) (a : GL (Fin V.rank) (AE D)) : Lvl.K ((frameLat V, a) : Lvl V) = Kmax V a

theorem **Lvl.good_frameLat** (V : HSp D) {a : GL (Fin V.rank) (AE D)} (ha : IsMaj V a) : Lvl.Good ((frameLat V, a) : Lvl V)

theorem **Lvl.exists_good** (V : HSp D) : ∃ l : Lvl V, l.Good ∧ l.K = Kmax V l.c

theorem **Lvl.Good.isCompact_K** {V : HSp D} {l : Lvl V} (h : l.Good) : IsCompact (l.K : Set (GA D V))

theorem **Lvl.form_frame** {V : HSp D} (l : Lvl V) : (adj l.c.1 · (l.form · l.c.1) : Mx D V.rank V.rank) = 1

theorem **sphericalSection_maximalCompact** (b : GL (Fin T.rank) (AE D)) : sphericalSectionIwasawa d (Kmax T b) = sphericalSection d b

theorem **conjG_I_smul_one** : conjG n (Complex.I • 1) = Complex.I • 1

theorem **projS_of_invariant** {T : HSp D} {K : Subgroup (GA D T)} (ν : Measure K) [IsProbabilityMeasure ν] {Φ : Mx D T.rank n → ℂ} (h : ∀ k ∈ K, ∀ x : Mx D T.rank n, Φ (mt T k · x) = Φ x) : projS ν Φ = Φ

Development.AbelianVarietyTheory.Definitions.LatticeFrames.StepVanOfStepX

theorem **exists_haar_fin** (V : HSp D) : ∃ μ : Measure (GfC V), μ.IsHaarMeasure

theorem **exists_haar_sub** (V : HSp D) (R : Subgroup (GA D V)) (hR : IsClosed (R : Set (GA D V))) : ∃ μ : Measure (R.subgroupOf (GfC V)), μ.IsHaarMeasure

theorem **vol_ext** {V : HSp D} {a b : Vol V} (h : a.μ = b.μ) (h' : a.ℱ = b.ℱ) : a = b

theorem **vol_μ** {j : ℕ} (h : j < D.R.m) : (vol d hb j).μ = (d.st j).ν

theorem **vol_F** {j : ℕ} (h : j < D.R.m) : (vol d hb j).ℱ = (d.st j).fd

theorem **comm_is** (hT : CoCpt D.sp.X') (hg : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hs : GlobalTheory.ChainOnto.ForStep D fun _ _ s s' p => GlobalTheory.Core.Mi.ItSpans s.M s'.M p.It) : (comm d hb).Is where

theorem **tup_is** (hst : d.Stab) : (tup d hb).Is where

theorem **trc_single** {V : HSp D} (g : GA D V) (f : GA D V → ℂ) : trc (Finsupp.single g 1) f = rTrans g f

theorem **trc_one** {V : HSp D} (f : GA D V → ℂ) : trc (Finsupp.single 1 1) f = f

theorem **per_eq** (j ρ h ψ G y v) : (tup d hb).per j ρ h ψ G y v = (d.B j ρ h).per d.c.ψ (d.rd j ψ y) (rTrans (d.t j) (G.1 v))

theorem **two_eq** (j φ C β) : (tup d hb).two j (fun _ => φ) C β = d.two j φ C

theorem **realEq** : RealEq fun _ : Unit => tup d hb

theorem **kerX** (h : d.KerLift) : KerX fun _ : Unit => tup d hb

theorem **rdX** (h : d.RdLaw) : RdX fun _ : Unit => tup d hb

theorem **perX** (h : d.PerLast) (ho : d.TopOnto) : PerX fun _ : Unit => tup d hb

theorem **valLastX** (h : d.ValLastS) : ValLastX fun _ : Unit => tup d hb

theorem **exists_of_lawsT** (l : d.LawsT) (hg : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hs : GlobalTheory.ChainOnto.ForStep D fun _ _ s s' p => GlobalTheory.Core.Mi.ItSpans s.M s'.M p.It) : ∃ (c : Comm D) (t : Tup c), c.Is ∧ t.Is ∧ c.X = D.sp.X ∧ (∀ j, Nonempty (t.Ch j)) ∧ t.cP ≠ 0 ∧ (∀ j, t.cV j ≠ 0 ∧ t.cK j ≠ 0) ∧ RealEq (fun _ : Unit => t) ∧ KerX (fun _ : Unit => t) ∧ RdX (fun _ : Unit => t) ∧ PerX (fun _ : Unit => t) ∧ ValLastX fun _ : Unit => t

theorem **up_mem** {t : Tup c} (ht : t.Is) {ψ : t.Col 0} (hψ : ψ ∈ t.Ram 0) : ∀ i, 2 * i ≤ D.R.m → t.up i ψ ∈ t.Ram (2 * i)

theorem **down** (ht : ∀ s, (t s).Is) (hS : StepX t) (n : ℕ) (h₀ : 0 + 2 * n + 2 = D.R.m) (I : Type) [Fintype I] (s : I → ℱ) (ψ : ∀ i, (t (s i)).Col 0) (hψ : ∀ i, ψ i ∈ (t (s i)).Ram 0) : ∀ k ρ (h : 2 * k + 2 * ρ + 2 = D.R.m) (α : I → ℂ) (G : ∀ i, (t (s i)).CC (2 * k)), (∀ y v, ∑ i, α i * (∏ l ∈ Finset.range k, (t (s i)).cV (2 * l) * (t (s i)).cK (2 * l) * (t (s i)).cK (2 * l + 1)) * (t (s i)).per (2 * k) ρ h ((t (s i)).up k (ψ i)) (G i) y v = 0) → ∀ y v, ∑ i, α i * (∏ j ∈ Finset.range (2 * k), (t (s i)).cI j) * (t (s i)).per 0 n h₀ (ψ i) ((t (s i)).dn (2 * k) (G i)) y v = 0

theorem **mainX_of_steps** (ht : ∀ s, (t s).Is) (hS : StepX t) (hL : LastX t) : MainX t

theorem **stepVan_of_stepX** (h : StepX fun _ : Unit => tup d hb) : d.StepVan

theorem **locFin** (hg : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hLL : GlobalTheory.ChainOnto.LevelLaw D) (h1 : d.Supp1) : (tup d hb).LocFin

Development.AbelianVarietyTheory.Definitions.LatticeFrames.SumConsecutiveRanks

theorem **errF_le_errA** (D : D7) : 0 ≤ errF D ∧ errF D ≤ errA D

theorem **sum_consecutive_ranks** (k T : ℝ) (n : ℕ) : ∑ j ∈ Finset.range n, (k + j * T) * (k + (j + 1) * T) = n * k ^ 2 + n ^ 2 * k * T + (n ^ 3 - n) / 3 * T ^ 2

theorem **errStep_le_errF** (D : D7) : errStep D ≤ errF D

theorem **colGe_of_le** (w : D.what₂.places.Pl) (x : ((D.fm u).lat 0).V) {s t : ℝ} (hst : s ≤ t) (h : Int.ColGe D u w x t) : Int.ColGe D u w x s

theorem **colA_of_colStep** (h : ColStep D EX u) : ColA D EX u

theorem **colCoarse_of_colA** (h : ColA D EX u) : ColCoarse D EX u

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ThetaIdentityOfMapsAll

theorem **rank_eq** (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) : (D.sp.Y 0).rank = X.B.I.n

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ToOneSucc

theorem **toOne_one** (ν : Measure k) (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (l : A → ℂ) : toOne ν σ 1 l = l

theorem **toOne_succ** (ν : Measure k) (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (p : ℕ) (l : A → ℂ) : toOne ν σ (p + 2) l = toOne ν σ (p + 1) (exDown ν (p + 1) σ l)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.TopLevExists

theorem **unit_transfer** {A B m n : Type} [CommRing A] [CommRing B] [Fintype m] [Fintype n] (e : m ≃ n) (s : A →+* B) (σ : A →+* A) (τ : B →+* B) (hs : ∀ x, s (σ x) = τ (s x)) (J : Matrix n n A) (J' : Matrix m m B) (hJ : ∀ a b, J' a b = s (J (e a) (e b))) (y : Matrix n n A) (g : Matrix m m B) (hg : ∀ a b, g a b = s (y (e a) (e b))) (hy : Formalism.UG.cT σ y * J * y = J) : Formalism.UG.cT τ g * J' * g = J'

theorem **swap_inj** : Function.Injective (Lev.swapAE D)

theorem **hasMat_mat** {h : GA D D.sp.X'} {y y'} (h1 : HasMat h y) (h2 : HasMat h y') : y = y'

theorem **hasMat_pt** {h h' : GA D D.sp.X'} {y} (h1 : HasMat h y) (h2 : HasMat h' y) : h = h'

theorem **hasMat_exists** (y : Formalism.UG.UPts (sigD D) (gramP D) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ h : GA D D.sp.X', HasMat h y.1

theorem **hasMat_one** : HasMat (1 : GA D D.sp.X') 1

theorem **topLev_exists** (K : ClsPTied u) : ∃ J χ, TopLev K J χ ∧ 1 ∈ J ∧ χ 1 = 1 ∧ ∀ κ : K.lev.U, ∃ h ∈ J, HasMat h κ.1

theorem **isJ_smul** {S : D.what₂.Tau → SPar} {M : DSModel D.sp.X' S} {j : SmT M →ₗ[ℂ] (FrP D → ℂ)} (h : IsJ u M j) {a : ℂ} (ha : a ≠ 0) : IsJ u M (a • j)

theorem **multT_bot** {V : HSp D} {S : D.what₂.Tau → SPar} (θ) (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) (hχ : ∀ U : Subgroup (GA D V), IsOpen (U : Set (GA D V)) → ∃ k ∈ U, k ∈ GfC V ∧ k ∈ J ∧ χ k ≠ 1) : MultT θ M J χ = ⊥

theorem **period_zero** {K : ClsPTied u} (V : ValPTied u K) {S : D.what₂.Tau → SPar} (θ) (M : DSModel D.sp.X' S) (j : SmT M →ₗ[ℂ] (FrP D → ℂ)) (J : Set (GA D D.sp.X')) (χ : GA D D.sp.X' → ℂ) (hχ : ∀ U : Subgroup (GA D D.sp.X'), IsOpen (U : Set (GA D D.sp.X')) → ∃ k ∈ U, k ∈ GfC D.sp.X' ∧ k ∈ J ∧ χ k ≠ 1) (eT : (D.mem u).P.Mt → MultT θ M J χ) (hf : TopFormula V M j fun F => (eT F).1) (F c v) : (D.mem u).P.period (V.eW.symm (by classical exact Pi.single c (j v))) F = 0

theorem **valC_ne_bot** {ι : Type} [Fintype ι] {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (lam : ι → DomWeight n) : ValC G lam ≠ ⊥

theorem **not_isJ_zero** {S : D.what₂.Tau → SPar} (M : DSModel D.sp.X' S) : ¬ IsJ u M 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.TraceInverseGramConjSmulEq

theorem **tau_zEll** (σ : ℕ) : (rat l.τ : Mx D B.L.rank L'.rank) · zEll L' σ ζ' = yZero B.L σ B.ζ

theorem **kappa_mul** (S : HSp D) (w : WO S L') (g : GA D S) : l.κ S (w · mt S g) = mt S g⁻¹ · l.κ S w

theorem **kappa_mul_jacobiEmbedding** {W W₀ : HSp D} {σ : ℕ} (d : HypDec W W₀ σ) (x : WO W L') : l.κ W₀ (x · d.j0) = d.p0 · l.κ W x

theorem **kappa_extX** {W W₀ : HSp D} {σ : ℕ} (d : HypDec W W₀ σ) : l.κ W (d.extX L' (zEll L' σ ζ')) = d.incAdj B.L (yZero B.L σ B.ζ)

theorem **ptT_eq** {W W₀ : HSp D} {σ : ℕ} (d : HypDec W W₀ σ) (n : GA D W) : ptT d B.L (yZero B.L σ B.ζ) n = l.κ W₀ (ptS d L' (zEll L' σ ζ') n)

theorem **tau_gram** (h : IsLineVec L' ζ') : l.τᴴ ∗ (B.L.gram ∗ l.τ) = (D.host.xi0 ∗ star D.host.xi0) • L'.gram

theorem **trace_inverseGram_conj_smul_eq** {R : Type} [CommRing R] {a b c : ℕ} (τ : Matrix (Fin a) (Fin b) R) (x : Matrix (Fin b) (Fin c) R) (g : Matrix (Fin c) (Fin c) R) (y : Matrix (Fin c) (Fin b) R) (σ : Matrix (Fin b) (Fin a) R) (k : Matrix (Fin a) (Fin a) R) (k' : Matrix (Fin b) (Fin b) R) {r s t : R} (e : σ ∗ (k ∗ τ) = t • k') (e1 : s ∗ (r ∗ t) = 1) : r • s • (τ ∗ (x ∗ (g ∗ (y ∗ (σ ∗ k))))).trace = (g ∗ (y ∗ (k' ∗ x))).trace

theorem **symplOH_kappa** (h : IsLineVec L' ζ') (S : HSp D) (w w' : WO S L') : symplOH B.L S (l.κ S w) (l.κ S w') = −symplOH S L' w w'

theorem **cenT_eq** (h : IsLineVec L' ζ') {W W₀ : HSp D} {σ : ℕ} (d : HypDec W W₀ σ) (n : GA D W) : cenT d B.L (yZero B.L σ B.ζ) n = −cenS d L' (zEll L' σ ζ') n

theorem **iter_zero** {ψ : AddChar (AF D.host.F) ℂ} {T : HSp D} {σ : ℕ} {B₁ : BesLev D σ} (c : Chain ψ T σ B₁) : c.Iter (fun _ => 0) 0

theorem **iter_smul** {ψ : AddChar (AF D.host.F) ℂ} {T : HSp D} {σ : ℕ} {B₁ : BesLev D σ} (c : Chain ψ T σ B₁) (a : ℂ) : ∀ q z, c.Iter q z → c.Iter (fun h => a ∗ q h) (a ∗ z)

theorem **chain_inhabited** (D : D7) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (T : HSp D) (B : BesLev D 1) (c : Chain ψ T 1 B), 0 < B.A.rank ∧ 0 < T.rank ∧ ∃ Φ : c.Data, c.IsTheta Φ ∧ GlobalTheory.Unf.Inhab.AllNonzero c Φ

theorem **jac_link** {ψ : AddChar (AF D.host.F) ℂ} (hψ : ∀ a, ψ (−a) = star (ψ a)) (h : IsLineVec L' ζ') {W W₀ : HSp D} {σ : ℕ} (d : HypDec W W₀ σ) (n : GA D W) (f : WO B.L W₀ → ℂ) : star (jacS ψ d L' (zEll L' σ ζ') n (fun w => star (f (l.κ W₀ w)))) 0) = jacT ψ d B.L (yZero B.L σ B.ζ) n f 0

theorem **semiMeas_ne_zero** {G H : Type} [Group G] [Group H] [MeasurableSpace G] [MeasurableSpace H] [MeasurableMul₂ G] (N : Subgroup G) (s : H →∗ G) (hs : Measurable s) (μ : Measure H) (ν : Measure N) [SFinite ν] (hμ : μ ≠ 0) (hν : ν ≠ 0) : semiMeas N s μ ν ≠ 0

theorem **sFinite_volN** {S S₀ : HSp D} {σ : ℕ} {d : HypDec S S₀ σ} (v : VolN d) : SFinite v.ν ∧ v.ν ≠ 0

theorem **jac_muR_ne_zero** (J : JacLev D ρ) : J.μR ≠ 0

theorem **bes_muR_ne_zero** (B : BesLev D ρ) : B.μR ≠ 0

Development.AbelianVarietyTheory.Definitions.LatticeFrames.TransferOn

theorem **TransferOn.mono** {D : D7} {C C' : HeightOneSpectrum (𝒪 D.host.F) → Prop} (h : TransferOn D C) (hC : ∀ v, C' v → C v) : TransferOn D C'

Development.AbelianVarietyTheory.Definitions.LatticeFrames.TrueColumnTupleTrivialEqOne

theorem **SelBF.gTiedF** {eps : ℝ} {Fm : FamP eps} (h : SelBF Fm) : Formalism.L2.G.GroupCompatibilityFamily Fm

theorem **SelBJF.selBF** {eps : ℝ} {Fm : FamP eps} (h : SelBJF Fm) : SelBF Fm

theorem **SelC.Joints.sig_piG_mem** {s : SelC ex} (J : s.Joints) (k : Fin 2) : Forms.sigD D ((s.cols.forms D.u₀).gl.piG k : D.host.E) ∈ D.host.K

theorem **SelC.Joints.not_mem_primes_referenceTuple** {s : SelC ex} (J : s.Joints) (p : ℕ) : p ∉ Y.Qp D.what₂.mu0

theorem **SelB.Joints.heckeChar_isUnitary_of_joints** {s : SelB ex} (J : s.Joints) (j : Fin D.R.m) : (s.ker.glob.mu0 j).IsUnitary

theorem **SelB.Joints.trueColumn_tuple_trivial_eq_one** {s : SelB ex} (J : s.Joints) : s.ker.tup D.what₂.mu0 = 1

theorem **SelB.Joints.transport_xi_eq** {s : SelB ex} (J : s.Joints) {μ μ' : D.what₂.Tup} (h : s.ker.tup μ = s.ker.tup μ') : Host.CT.transport s.ker.seam.iE (s.tp.xi μ) = Host.CT.transport s.ker.seam.iE (s.tp.xi μ')

theorem **SelB.Joints.div_plMap** {s : SelB ex} (J : s.Joints) {i : Fin 2} {P : HeightOneSpectrum (𝒪 s.ker.X.E)} (h : P ∈ s.ker.Pr.vG i) : s.cf.P.Div i (Host.CT.plMap s.ker.seam.iE P)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.UnderEqOfConj

theorem **not_two_primes** {I : Ideal (𝒪 D.host.E)} (hI : I ≠ ⊤) {p q : ℕ} (hp : p.Prime) (hq : q.Prime) (hne : p ≠ q) (h1 : (p : 𝒪 D.host.E) ∈ I) (h2 : (q : 𝒪 D.host.E) ∈ I) : False

theorem **char_mem** {P : Pl D.host.E} {q : ℕ} (h : Nat.card (kP P) = q) : (q : 𝒪 D.host.E) ∈ P.asIdeal

theorem **nat_mem_of_conj** {P P' : Pl D.host.E} (h : Lev.IsConjPl (D

theorem **g_mem_of_div** (C : Formalism.L2.Host.CharFrame D) {k : Fin 2} {PP : Pl D.host.E} (h : C.P.Div k PP) : ((D.host.g k : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal

theorem **disjoint_Pg_Q** (L : LevCh D T) {μ0 : T} (hL : L.Is μ0) (t : T) : Disjoint L.Pg (L.Q t)

theorem **rangeX_iff** : RangeX R ↔ Formalism.IndefR.RecTie.TopMultH R

theorem **comR_S** (t : T) : (comR R t).S = GlobalTheory.IndefiniteRowSideFullHecke.LabelSettingScalars.badPl R t

theorem **mem_S** (hL : R.toLevCh.Is A.μ0) (t : T) (k : Fin 2) (P : Pl D.host.E) (hk : ((D.host.g k : ℕ) : 𝒪 D.host.E) ∈ P.asIdeal) : P.under (𝒪 D.host.F) ∈ (comR R t).S

theorem **stages** (hL : R.toLevCh.Is A.μ0) (hM : R.IsModels) (t : T) : (∀ j < D.R.m, IsStage (comR R t) (twR R t) j (stY R t j)) ∧ IsStage (comR R t) (twR R t) D.R.m (stX R t)

theorem **steps** (hC : R.IsChain) (h : RangeX R) (t : T) : (∀ k, k + 1 < D.R.m → IsStep (twR R t) (k + 1) (stepY R t k)) ∧ IsStep (twR R t) D.R.m (stepX R h t)

theorem **twist_of** (hL : R.toLevCh.Is A.μ0) (t : T) (hc : (comR R t).Is) (hne : ∀ PP ∈ R.Q t, R.xiR t PP ≠ 1) : (twR R t).Is (comR R t)

theorem **stepX_toCC** (h : RangeX R) (t : T) (x : (R.HX t).H) : PiTensorProduct.lift.symm ((stepX R h t).io (toCC R h t x)).1 = GlobalTheory.IndefiniteRowSideFullHecke.LabelSettingScalars.topStep R t x

theorem **toCC_val** (h : RangeX R) (t : T) (x : (R.HX t).H) (v) (g) : PiLocalization₂.val (toCC R h t x) v g = R.prX (R.cX.ext ((R.HX t).e x) g) v

theorem **toCC_surj** (h : RangeX R) (t : T) : Function.Surjective (toCC R h t)

theorem **isInnerProduct_of_isMet** (t : T) (k : ℕ) {ipD ip} (hm : IsMet (comR R t) (stY R t k) ipD ip) (x y : (stY R t k).CC (comR R t)) : GlobalTheory.IndefiniteRowSideFullHecke.LabelSettingScalars.IsIP R t k (PiTensorProduct.lift.symm x.1) (PiTensorProduct.lift.symm y.1) (ip y x)

theorem **ipDict_of** (hv : ∀ k < D.R.m, ∃ v : (R.MY k).Vec, ∏ τ, ‹(v τ : (R.MY k).H τ), (v τ : (R.MY k).H τ)›_ℂ ≠ 0) : IPDict R

theorem **stepExactNS_of** (h : StepExact D) : StepExactNS D

theorem **Q_base** (hL : R.toLevCh.Is A.μ0) : R.Q A.μ0 = ∅

theorem **isUnitAt_of** (P : Pl D.host.E) (a : 𝒪 D.host.E) (k : D.host.Eˣ) (u : (kP P)ˣ) (hk : (k : D.host.E) = a) (hu : (u : kP P) = Ideal.Quotient.mk _ a) : Formalism.L2.Host.IsUnitAt P (Formalism.L2.Host.atFin P k)

theorem **under_eq_of_conj** {P P' : Pl D.host.E} (h : Lev.IsConjPl (D

theorem **com_is** (hL : R.toLevCh.Is A.μ0) (hM : R.IsModels) (hC : R.IsChain) (t : T) : (comR R t).Is

theorem **xiR_ne** (hL : R.toLevCh.Is A.μ0) (t : T) : ∀ PP ∈ R.Q t, R.xiR t PP ≠ 1

theorem **rChain** (hL : R.toLevCh.Is A.μ0) (hM : R.IsModels) (hC : R.IsChain) (h : RangeX R) : RChain R h

theorem **zinv_pos** (PP : Pl D.host.E) (a b : ℕ) : 0 < zinv PP a b

theorem **rVol_zero** (h : RVol R) (t : T) : GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.volQ (Yc D 0) (R.νY t 0) = GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.volQ (Yc D 0) (R.νY A.μ0 0)

Development.AbelianVarietyTheory.Definitions.LatticeFrames.UnfoldingConstantsPos

theorem **MIMap.miKer** {0 : Desc D u} (h : MIMap D u 0) : MIKer D u 0

theorem **OntoAll.ontoP** {0 : Desc D u} (h : OntoAll D u 0) : OntoP D u 0

theorem **kerIncl_of** {0 : Desc D u} (h : MIKer D u 0) (h' : OntoP D u 0) : KerIncl D u

theorem **d0_eq** : d0 D u = D.duS u

Development.AbelianVarietyTheory.Definitions.LatticeFrames.ValueDoubleExact

theorem **SplitPt.sec_mul** (p : SplitPt D k) (a b : k) : p.sec (a * b) = p.sec a * p.sec b

theorem **SplitPt.sec_one** (p : SplitPt D k) : p.sec 1 = p.e

theorem **SplitPt.idele_val** (p : SplitPt D k) (a : kˣ) : (p.idele a : AE D) = 1 - p.e + p.sec a

theorem **SplitPt.pr_idele** (p : SplitPt D k) (a : kˣ) : p.pr (p.idele a) = a

theorem **IsGlueConst.norm** {q : Unif k} {I : List ι} {N r : ℕ} {μ : Fin r → ((AE D)ˣ →* Circle)} {c : ℂ} (h : IsGlueConst p q I N μ c) : ∃ a : ℤ, ‖c‖ = (‖(resCard k : ℂ) ^ N - 1‖ * √(resCard k : ℝ) ^ a) ^ I.length

theorem **IsGlueConst.ne_zero** {q : Unif k} {I : List ι} {N r : ℕ} {μ : Fin r → ((AE D)ˣ →* Circle)} {c : ℂ} (h : IsGlueConst p q I N μ c) (hN : 0 < N) : c ≠ 0

theorem **sgnRes_eq** (s : ℤˣ) : sgnRes k s = 1 ∨ sgnRes k s = -1

theorem **ValueDoubleExact.valueDouble** (h : ValueDoubleExact D) : ValueDouble D

theorem **ValueLastExact.valueLast** (h : ValueLastExact D) : ValueLast D

Development.AbelianVarietyTheory.Definitions.LatticeFrames.VanishingGramTermsBoundParams

theorem **hypL_of** {D : D7} {large₂ : SizeData₂ D} {C' Cl Cg cb Cu cT CT : ℝ} (h : StandingHypotheses D large₂ C' Cl Cg cb Cu cT CT) : HypL D large₂.sc.L C' Cl Cg cb Cu cT CT

Development.AbelianVarietyTheory.Definitions.LatticeFrames.XiEqXiOffMul

theorem **twist_one** (w : VTup t) : twist w (fun _ _ => 1) = w

theorem **QPinS.of** {v : VCom c} {w : VTup t} (h : QPin v w) : QPinS v w

theorem **exists_xOrd** (R : Ranks) (j : ℕ) : ∃ e : XIx (Fin R.k) (Fin R.t) j ≃ Fin (R.N j), GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.xPos e

theorem **exists_topOrd** (R : Ranks) (s : ℕ) (hs : s + 1 + 1 = R.m) : ∃ e : Ix (Fin R.k ⊕ Fin (s + 1)) (Fin R.t) (s + 2) ⊕ Unit ≃ Fin (R.N (s + 1 + 1)), GlobalTheory.SymBridge.IsOrd (Sum.elim GlobalTheory.SymBridge.zPos fun _ => R.N R.m - 1) e

theorem **xi_eq_xiOff_mul** (i : ℕ) (hi : i + 1 ≤ B.R.m) (α : Ch₀ i) (γ : Ch₀ (i + 1)) (z) : A.xi i hi α γ z = xiOff A i hi α γ z * B.embLL (∏ P ∈ Finset.univ.filter (fun P : BadPr I a => ¬ P.1 ∈ (B.primes X).vGs), locXi A i α γ P (homOf I l Z i hi z))

theorem **BaseCommon.moved** (h : BaseCommon A) : BaseMoved A

theorem **ownVec_mem** (P : BadPr I a) (hP : P.1 ∉ (B.primes X).vGs) (hc : GlobalTheory.SymBridge.IsCompl (E

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.DenseGenericOfFlat

theorem **dense_generic_of_flat** [IsDomain o] (s : S ⟶ Sp o) [Flat s] : Dense {x : S | (s x).asIdeal = ⊥}

theorem **dense_generic_spMap** {O : Type} [CommRing O] [IsDomain O] (β : o →+* O) (hβ : Function.Injective β) : Dense {x : Sp O | (spMap β x).asIdeal = ⊥}

theorem **hom_ext** (F : SFam h k ζ) {s' : S ⟶ Sp o} (c : SiegPre h k ζ s') (f f' : S ⟶ F.X) (hs : f ≫ F.s = f' ≫ F.s) (h₁ : SiegPB c F.univ f) (h₂ : SiegPB c F.univ f') : f = f'

theorem **existsUnique** (F : SFam h k ζ) {s' : S ⟶ Sp o} (c : SiegPre h k ζ s') (hr : IsReduced S) (hd : Dense {x : S | (s' x).asIdeal = ⊥}) (hc : c.Principal) : ∃! f : S ⟶ F.X, f ≫ F.s = s' ∧ SiegPB c F.univ f

theorem **exists_point** (F : SFam h k ζ) {O : Type} [CommRing O] [IsDomain O] (β : o →+* O) (hβ : Function.Injective β) (c : SiegPre h k ζ (spMap β)) (hc : c.Principal) : ∃ x : Sp O ⟶ F.X, x ≫ F.s = spMap β ∧ SiegPB c F.univ x

theorem **nonempty** (F : SFam h k ζ) {O : Type} [CommRing O] [IsDomain O] (β : o →+* O) (hβ : Function.Injective β) (c : SiegPre h k ζ (spMap β)) (hc : c.Principal) : Nonempty F.X

theorem **isReduced_and_dense** [IsDomain o] (F : SFam h k ζ) : IsReduced F.X ∧ Dense {x : F.X | (F.s x).asIdeal = ⊥}

theorem **exists_iso** [IsDomain o] (F F' : SFam h k ζ) : ∃ e : F.X ≅ F'.X, e.hom ≫ F'.s = F.s ∧ SiegPB F.univ F'.univ e.hom

theorem **exists_baseChange** (F : SFam h k ζ) {B : Type} [CommRing B] {ζ' : B} (j : o →+* B) (hi : Function.Injective j) (hj : j ζ = ζ') : ∃ (F' : SFam h k ζ') (π : F'.X ⟶ F.X), IsPullback π F'.s F.s (spMap j) ∧ SiegPB F'.univ F.univ π

theorem **siegRep** {p M : ℕ} (F : SFam h M (zetaZP p M)) (sm : SmoothOfRelativeDimension (h * (h + 1) / 2) F.s) (qc : QuasiCompact F.s) (pr : F.univ.Principal) (pts unr) : GlobalTheory.Models.SiegRep ((F.X, F.s, sm, qc, F.univ, pr, pts, unr) : SiegelFamily h p M)

theorem **SFam.exists_sRep_siegRep** {p M : ℕ} (hM : M ≠ 0) (F : SFam h M (zetaZP p M)) (Ag : SiegelFamily h p M) (H : SRep Ag) : ∃ Ag' : SiegelFamily h p M, SRep Ag' ∧ GlobalTheory.Models.SiegRep Ag'

theorem **PELRep.exists_point** {D : Datum H} {a : J → 𝒪 E} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] D.La} {ρ : o} {t : TObj g k J b s} (R : PELRep D a bas ρ t) {K : Type} [Field K] (β : o →+* K) (hβ : Function.Injective β) (t' : TObj g k J b (spMap β)) (τ' : SAB H (Sp K)) (h' : IsPEL D a bas ρ t' τ') : ∃ x : Sp K ⟶ S, x ≫ s = spMap β ∧ TMap t' t x

theorem **exists_hom_of_basis** {G : Type} [CommGroup G] {n : ℕ} (p : Fin n → G) : ∃ q : (Fin n → ℤ) → G, (∀ x y, q (x + y) = q x * q y) ∧ ∀ i, q (Pi.single i 1) = p i

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsAdmissibleKernelScheme

theorem **pointwise_iff** : ToroidalHeckeCompatibility (H

theorem **exists_pt** (X : Scheme.{0}) (g : X) : ∃ (k : Type) (_ : Field k) (_ : IsAlgClosed k) (x : Sp k ⟶ X), Set.range x = {g}

theorem **lift_closed** {T N B : Scheme.{0}} [IsReduced T] (x : T ⟶ B) (ν : N ⟶ B) [IsClosedImmersion ν] (h : Set.range x ⊆ Set.range ν) : ∃ l : T ⟶ N, l ≫ ν = x

theorem **set_ext** {G : Over S} {K K' : Set G.left} (h : ∀ (k : Type) [Field k] [IsAlgClosed k] (x : Sp k ⟶ S) (z : Over.mk x ⟶ G), Set.range z.left ⊆ K ↔ Set.range z.left ⊆ K') : K = K'

theorem **kerSet_unique** {G G' : Over S} [GrpObj G'] {ψ : G ⟶ G'} {K K' : Set G.left} (h : KerSet ψ K) (h' : KerSet ψ K') : K = K'

theorem **kerSet_preimage** {G G' : Over S} [GrpObj G'] [IsSeparated G'.hom] (ψ : G ⟶ G') : KerSet ψ (ψ.left ⁻¹' Set.range (η[G'] : 𝟙_ (Over S) ⟶ G').left)

theorem **etSet_of_kerSet** {G G' : Over S} [GrpObj G'] [IsSeparated G'.hom] {ψ : G ⟶ G'} [IsFinite ψ.left] [Etale ψ.left] {K : Set G.left} (h : KerSet ψ K) : EtSet G K

theorem **isHecke_iff** (o o' : Ob D l s) (P : Ideal (𝒪 E)) (p i : ℕ) (ψ : o.t.G ⟶ o'.t.G) : IsHecke o o' P p i ψ ↔ IsIsog o o' p ψ ∧ ∃ K, KerSet ψ K ∧ HeckeSet D o.t P p i K ∧ EtSet o.t.G K ∧ HasQuot o p K

theorem **range_subset_torsSet** (o : Ob D l s) (p : ℕ) (k : Type) [Field k] [IsAlgClosed k] (x : Sp k ⟶ S) (z : Over.mk x ⟶ o.t.G) : Set.range z.left ⊆ torsSet o p ↔ z ≫ o.t.act.act (p : 𝒪 E) = 1

theorem **heckeKer_tors** (o : Ob D l s) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) (k : Type) [Field k] [IsAlgClosed k] (x : Sp k ⟶ S) : o.t.HeckeKer (Over.mk x) D P p 0 (torsSub o.t (Over.mk x) p)

theorem **heckeSet_torsSet** (o : Ob D l s) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) : HeckeSet D o.t P p 0 (torsSet o p)

theorem **eq_torsSub** (o : Ob D l s) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) (k : Type) [Field k] [IsAlgClosed k] (x : Sp k ⟶ S) (C : Subgroup (Over.mk x ⟶ o.t.G)) (hC : o.t.HeckeKer (Over.mk x) D P p 0 C) : C = torsSub o.t (Over.mk x) p

theorem **eq_torsSet** (o : Ob D l s) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) {K : Set o.t.G.left} (hK : HeckeSet D o.t P p 0 K) : K = torsSet o p

theorem **torsSet_pull** {S' : Scheme.{0}} {s' : S' ⟶ Sp (KA l)} (o' : Ob D l s') (o : Ob D l s) {f : S' ⟶ S} (n : ObjMap o' o f) (p : ℕ) : torsSet o' p = n.toSABMap.φ ⁻¹' torsSet o p

theorem **isHecke_mul** [IsLocallyNoetherian S] (o : Ob D l s) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) : IsHecke o o P p 0 (o.t.act.act (p : 𝒪 E))

theorem **etSet_torsSet** [IsLocallyNoetherian S] (o : Ob D l s) {p : ℕ} (hp : 0 < p) : EtSet o.t.G (torsSet o p)

theorem **kerZero_carries** (Xo : Moduli D l) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) (k : Type) [Field k] [IsAlgClosed k] (y : Sp k ⟶ Xo.X) : (kerZero Xo P hp).Carries y y (torsSub Xo.univ.t (Over.mk y) p)

theorem **exists_kerScheme** (hr : 1 ≤ D.p + D.q) (l : Level H) (P : Ideal (𝒪 E)) {p : ℕ} (hp : 0 < p) : ∃ (Xo : Moduli D l) (Z : KerScheme Xo P p 0) (z : Sp ℂ ⟶ Z.Hk) (y : Sp ℂ ⟶ Xo.X) (C : Subgroup (Over.mk y ⟶ Xo.univ.t.G)), Z.Carries z y C ∧ Nat.card C = p ^ (2 * (Module.finrank ℚ E / 2 * (D.p + D.q)))

theorem **exists_adm** (l : Level H) : ∃ (P : Ideal (𝒪 E)) (p : ℕ), admitted l P p

theorem **exists_admissible_kernel_scheme** (hr : 1 ≤ D.p + D.q) (l : Level H) : ∃ (P : Ideal (𝒪 E)) (p : ℕ) (Xo : Moduli D l) (Z : KerScheme Xo P p 0) (z : Sp ℂ ⟶ Z.Hk) (y : Sp ℂ ⟶ Xo.X) (C : Subgroup (Over.mk y ⟶ Xo.univ.t.G)), admitted l P p ∧ Z.Carries z y C ∧ C ≠ ⊥

theorem **smooth** (Z : KerScheme Xo 𝔓 p i) : Smooth (Z.π₁ ≫ Xo.s)

theorem **isReduced** (Z : KerScheme Xo 𝔓 p i) : IsReduced Z.Hk

theorem **comp_π₁** (Z : KerScheme Xo 𝔓 p i) [IsReduced S] {o : Ob D l s} {h : S ⟶ Z.Hk} (hh : h ≫ Z.π₁ ≫ Xo.s = s) (n : ObjMap o Z.o₁ h) {f : S ⟶ Xo.X} (hf : f ≫ Xo.s = s) (m : ObjMap o Xo.univ f) : h ≫ Z.π₁ = f

theorem **exists_iso** (Z Z' : KerScheme Xo 𝔓 p i) : ∃ e : Z.Hk ≅ Z'.Hk, e.hom ≫ Z'.π₁ = Z.π₁ ∧ ∃ n : ObjMap Z.o₁ Z'.o₁ e.hom, Z.K = n.toSABMap.φ ⁻¹' Z'.K

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsArBaseAway

theorem **isE8_W8** : IsE8 W8

theorem **exists_arBase_away** {F : Type} [Field F] [NumberField F] (φ : F →+* ℂ) {n : ℕ} (hn : n ≠ 0) : ∃ em : Localization.Away (n : 𝒪 F) →+* ℂ, ArBase em ∧ em.comp (algebraMap (𝒪 F) (Localization.Away (n : 𝒪 F))) = φ.comp (algebraMap (𝒪 F) F)

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsGeometricPoint

theorem **coe_rt** (S A : ℕ) : (RZ.rt H S A : ℂ) = H.rt0

theorem **coe_ze** (S A : ℕ) : (RZ.ze H S A : ℂ) = zeta A

theorem **rt_sq** (S A : ℕ) : RZ.rt H S A ^ 2 = -(H.d : RZ H S A)

theorem **fullObj_of_obj** {D : Datum H} {l : Level H} {n : ℕ} (β : RZ H n l.A →+* KA l) (hβ : IsInc (RZ H n l.A).subtype (KA l).val.toRingHom β) {S : Scheme.{0}} {s : S ⟶ Sp (KA l)} (o : Obj D l (baseKA l) s) : FullObj D (s ≫ spMap β) o.t o.etaA

theorem **exists_roots** {n N : ℕ} {k : Type} [Field k] [IsAlgClosed k] [CharZero k] (β : RZ H n N →+* k) : ∃ ζ : ℕ → kˣ, (∀ m, 0 < m → IsPrimitiveRoot (ζ m) m) ∧ (∀ a b, 0 < a → 0 < b → ζ (a * b) ^ b = ζ a) ∧ (ζ N : k) = β (RZ.ze H n N)

theorem **weilIs_zero** {S : Scheme.{0}} (T : Over S) (m : ℕ) : Out.WeilIs (𝟙_ (Over S)) (UU.trivRig (𝟙_ (Over S))).L m (1 : T ⟶ _) 1 1

theorem **fullObj_zero** {n N : ℕ} {S : Scheme.{0}} (s : S ⟶ Sp (RZ H n N)) : FullObj (Test.zeroDatum H) s (Inhab.zeroTriple H S) (fun _ => 1)

theorem **not_fullObj_zero** {D : Datum H} [Nontrivial D.La] {n N : ℕ} {S : Scheme.{0}} (s : S ⟶ Sp (RZ H n N)) (lev) (x : Sp ℂ ⟶ S) : ¬ FullObj D s (Inhab.zeroTriple H S) lev

theorem **siegClause_of_top** (T : Top Xo n N) : SiegClause D T.sM T.tM T.lev

theorem **gs_one** {X Y : Scheme.{0}} (f : Y ⟶ X) (y : Y) (U : X.Opens) (h : f y ∈ U) : gs f y U 1 = 1

theorem **mem_wchart_triv** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (j) (y : Y) : y ∈ LB.wchart (Inh.trivLB X) φ j

theorem **lin_one** {T T' : Over Z} (h : T' ⟶ T) : Lin G (Inh.trivLB G.left) N h (fun _ : Fin 1 => oneSec _ _) (fun _ => 1) (oneSec _ _)

theorem **basisAlong_one** {T T' : Over Z} (h : T' ⟶ T) (hb : Function.Bijective (fst T' G).left.appTop.hom) : BasisAlong G (Inh.trivLB G.left) N h (fun _ : Fin 1 => oneSec _ _)

theorem **ampFrames_triv** [IsAffine Z] [IsProper G.hom] [Flat G.hom] [GeometricallyIntegral G.hom] : Nonempty (AmpFrames G (Inh.trivLB G.left) N)

theorem **zOn_self** {O : Type} [CommRing O] {Z : Scheme.{0}} (s : Z ⟶ Sp O) (t : SAB H Z) (m : ℕ) (hn : Geo.Tor.IsNormalScheme Z) (hp : IsProjOver s) (hf : Flat s) (hq : IsQProjOver (t.G.hom ≫ s)) (hg : Good t) (W : Z.Opens) (hW : Dense (W : Set Z)) (ha : Nonempty (GlobalTheory.LN.AmpFrames t.G t.PP.L (m ^ 2))) : ∃ X : ZOn s t m, X.Z = Z ∧ X.W = W

theorem **zOn_zero** (O : Type) [CommRing O] [IsDomain O] [IsIntegrallyClosed O] (m : ℕ) (W : (Sp O).Opens) (hW : (W : Set (Sp O)).Nonempty) : ∃ X : ZOn (𝟙 (Sp O)) (Inhab.zeroTriple H (Sp O)) m, X.Z = Sp O ∧ X.W = W

theorem **range_j_subset** {O : Type} [CommRing O] {MM : Scheme.{0}} {sM : MM ⟶ Sp O} {tM : SAB H MM} {m : ℕ} (X : ZOn sM tM m) (h : IsProper tM.G.hom) : Set.range X.j ⊆ abLocus X.t.G

theorem **exists_map_of_famCore** {l : Level H} (Xo : Moduli D l) {MM : Scheme.{0}} {sM : MM ⟶ Sp (RZ H n l.A)} {tM : SAB H MM} {lev} (F : FamCore D sM tM lev) (β : RZ H n l.A →+* KA l) (hβ : IsInc (RZ H n l.A).subtype (KA l).val.toRingHom β) (hf : Flat (spMap β)) (hr : IsReduced Xo.X) : ∃! f : Xo.X ⟶ MM, f ≫ sM = Xo.s ≫ spMap β ∧ ∃ ν : SABMap Xo.univ.t tM f, ∀ a, (Xo.univ.etaA a).left ≫ ν.φ = f ≫ (lev a).left

theorem **nonempty_of_famCore_zero** [IsNoetherianRing (RZ H n N)] {MM : Scheme.{0}} {sM : MM ⟶ Sp (RZ H n N)} {tM : SAB H MM} {lev} (F : FamCore (Test.zeroDatum H) sM tM lev) : ∃ f : Sp (RZ H n N) ⟶ MM, f ≫ sM = 𝟙 _

theorem **torsion_eq_one** (h : FullObj (Test.zeroDatum H) s t lv) {k : Type} [Field k] [IsAlgClosed k] [CharZero k] (x : Sp k ⟶ S) {m : ℕ} (hm : 0 < m) (g : Over.mk x ⟶ t.G) (hg : g ^ m = 1) : g = 1

theorem **point_eq_one** (h₁ : Out.Laws) (h₂ : Out.Laws2) (h : FullObj (Test.zeroDatum H) s t lv) {k : Type} [Field k] [IsAlgClosed k] [CharZero k] (z : Sp k ⟶ t.G.left) : z = z ≫ t.G.hom ≫ eSec t.G

theorem **exists_geometric_point_over** {X : Scheme.{0}} (K : Type) [Field K] (u : Sp K ⟶ X) : ∃ (k : Type) (_ : Field k) (_ : IsAlgClosed k) (z : Sp k ⟶ X) (q : Sp k) (q' : Sp K), z q = u q'

theorem **exists_geom** {X : Scheme.{0}} (y : X) : ∃ (k : Type) (_ : Field k) (_ : IsAlgClosed k) (z : Sp k ⟶ X) (q : Sp k), z q = y

theorem **exists_geom_specializes** {X : Scheme.{0}} {R : Type} [CommRing R] [IsDomain R] [CharZero R] (f : X ⟶ Sp R) [Flat f] (y : X) : ∃ (k : Type) (_ : Field k) (_ : IsAlgClosed k) (_ : CharZero k) (z : Sp k ⟶ X) (q : Sp k), z q ⤳ y

theorem **hom_comp_eSec** (h₁ : Out.Laws) (h₂ : Out.Laws2) (h : FullObj (Test.zeroDatum H) s t lv) [IsReduced S] [Flat s] : t.G.hom ≫ eSec t.G = 𝟙 _

theorem **lbIso_triv** {X Y : Scheme.{0}} (L : LB X) (φ : X ⟶ Y) (c : LB.Gen L) (hc : c.IsGen) : LBIso L ((Out.Inh.trivLB Y).pull φ)

theorem **famCore_zero** (h₁ : Out.Laws) (h₂ : Out.Laws2) (H : Host E) (n N : ℕ) : FamCore (Test.zeroDatum H) (𝟙 (Sp (RZ H n N))) (Inhab.zeroTriple H _) (fun _ => 1)

theorem **zOn_of_top** (T : Top Xo n N) (m : ℕ) (a : GlobalTheory.LN.AmpFrames T.t.G T.t.PP.L (m ^ 2)) : ∃ X : ZOn T.sM T.tM m, X.Z = T.Z

theorem **zOn_int** (m : ℕ) : ∃ X : ZOn (𝟙 (Sp ℤ)) (Inhab.zeroTriple HostModel.host (Sp ℤ)) m, X.W = PrimeSpectrum.basicOpen (2 : ℤ)

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsIsE8IsSharpBasis

theorem **isE8_W8** : IsE8 W8

theorem **not_isE8_one** : ¬ IsE8 1

theorem **arBase_int** : ArBase (Int.castRingHom ℂ)

theorem **not_arBase_field** {K : Type} [Field K] (em : K →+* ℂ) : ¬ ArBase em

theorem **arBase_away** : ArBase (awayEmb φ hn)

theorem **halves_inl** (g : ℕ) (a : Fin g) : (halves g (.inl a) : ℕ) = a

theorem **halves_inr** (g : ℕ) (a : Fin g) : (halves g (.inr a) : ℕ) = g + a

theorem **sum_halves** (g : ℕ) (i j : Fin (2 * g)) : ∑ a : Fin g, ((Pi.single i 1 : Fin (2 * g) → ℤ) (halves g (.inl a)) * (Pi.single j 1 : Fin (2 * g) → ℤ) (halves g (.inr a)) - (Pi.single i 1 : Fin (2 * g) → ℤ) (halves g (.inr a)) * (Pi.single j 1 : Fin (2 * g) → ℤ) (halves g (.inl a))) = Out.stdSymp g i j

theorem **dual_sum** (y : Fin 8 → ℤ) (u : Fin 8) : ∑ v, E8 u v * ∑ w, E8inv v w * y w = y u

theorem **sharp_form** (x x' : Fin (2 * (8 * h)) → ℤ) : ∑ u, ∑ v, E8 u v * omegaStd (sharpBas h x u) (sharpBas h x' v) = ∑ a : Fin (8 * h), (x (halves _ (.inl a)) * x' (halves _ (.inr a)) - x (halves _ (.inr a)) * x' (halves _ (.inl a)))

theorem **isSharpBasis_sharpBas** : IsSharpBasis W8 (sharpBas h)

theorem **exists_isE8_isSharpBasis** : ∃ W : Matrix (Fin 8) (Fin 8) ℤ, IsE8 W ∧ ∀ h : ℕ, ∃ bas : (Fin (2 * (8 * h)) → ℤ) ≃ₗ[ℤ] (Fin 8 → (Fin h ⊕ Fin h → ℤ)), IsSharpBasis W bas

theorem **eq_eight_mul** {g h : ℕ} (bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] (Fin 8 → (Fin h ⊕ Fin h → ℤ))) : g = 8 * h

theorem **act_mul_act_of_isPEL** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {D : Datum H} {a : J → 𝒪 E} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] D.La} {ρ : o} {t : TObj g k J b s} {τ : SAB H S} (ht : IsPEL D a bas ρ t τ) {w w' : J} (hw : a w + a w' = 1) : t.act w * t.act w' = 𝟙 t.A

theorem **act_one_of_isSharp** {h : ℕ} {W : Matrix (Fin 8) (Fin 8) ℤ} {a : J → Matrix (Fin 8) (Fin 8) ℤ} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] (Fin 8 → (Fin h ⊕ Fin h → ℤ))} {t : TObj g k J b s} {c : SiegPre h k (b.ζ : o) s} (ht : IsSharp W a bas t c) {j : J} (hj : a j = 1) : t.act j = 𝟙 t.A

theorem **corr_comm_pull** {X X' Y : Scheme.{0}} (f : X ⟶ X') {K : Type} [Fintype K] (φ φ' : K → (Y ⟶ X)) (ε : K → ℤ) (a : LB.Gen (LB.LinComb ((trivLB X').pull f) φ ε)) (b : LB.Gen (LB.LinComb ((trivLB X').pull f) φ' ε)) (h h' : Y ⟶ Y) (hh : h = 𝟙 Y) (hh' : h' = 𝟙 Y) (c : LB.Gen (LB.LinComb ((trivLB X').pull f) (Y

theorem **isGen_one** {X : Scheme.{0}} (L : LB X) (hL : ∀ i j, L.g i j = 1) : ((fun _ => 1) : L.Gen).IsGen

theorem **weil_pull_zero** {X : Scheme.{0}} (f : (𝟙_ (Over S)).left ⟶ X) (n : ℤ) {T : Over S} (x y : T ⟶ 𝟙_ (Over S)) : Out.WeilIs (𝟙_ (Over S)) ((trivLB X).pull f) n x y 1

theorem **relAmple_unit** (M : LB (𝟙_ (Over S)).left) (hg : ∀ i j, M.g i j = 1) (i₀ : M.ι) (hi₀ : M.U i₀ = ⊤) : Out.RelAmple (𝟙_ (Over S)) M

theorem **isSharp_zero** (W : Matrix (Fin 8) (Fin 8) ℤ) (a : J → Matrix (Fin 8) (Fin 8) ℤ) (bas) (b : TBase o k) (s : S ⟶ Sp o) : IsSharp W a bas (zeroObj J b s) (zeroSieg (b.ζ : o) s)

theorem **zeroSieg_principal** (ζ : o) (s : S ⟶ Sp o) : (zeroSieg (k

theorem **sharpClass_zero** (W : Matrix (Fin 8) (Fin 8) ℤ) (a : J → Matrix (Fin 8) (Fin 8) ℤ) (bas) (b : TBase o k) (em : o →+* ℂ) : SharpClass (h

theorem **isPEL_zero** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (a : J → 𝒪 E) (bas) (ρ : ℂ) (hk : k ≠ 0) (s : S ⟶ Sp ℂ) : IsPEL (Test.zeroDatum H) a bas ρ (zeroObj J (baseC k hk) s) (Models.Inhab.zeroTriple H S)

theorem **pelClass_zero** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (a : J → 𝒪 E) (bas) (ρ : ℂ) (hk : k ≠ 0) : PELClass (Test.zeroDatum H) a bas ρ (RingHom.id ℂ) (zeroObj J (baseC k hk) _)

theorem **pt_act_of_pelClass** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {D : Datum H} {g : ℕ} {a : J → 𝒪 E} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] D.La} {ρ : o} {b : TBase o k} {em : o →+* ℂ} {t : TObj g k J b (spMap em)} (ht : PELClass D a bas ρ em t) (j : J) (i : Fin (2 * g)) : t.pt i ≫ t.act j = (List.ofFn fun i' => t.pt i' ^ (bas.symm (a j • bas (Pi.single i 1)) i')).prod

theorem **act_comp_of_isSharp** {h g : ℕ} {W : Matrix (Fin 8) (Fin 8) ℤ} {a : J → Matrix (Fin 8) (Fin 8) ℤ} {b : TBase o k} {s : S ⟶ Sp o} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] (Fin 8 → (Fin h ⊕ Fin h → ℤ))} {t : TObj g k J b s} {c : SiegPre h k (b.ζ : o) s} (ht : IsSharp W a bas t c) {j j' l : J} (hl : a l = a j' * a j) : t.act j ≫ t.act j' = t.act l

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsPelRepIsPEL

theorem **isIso_of_points** [IsReduced S] (A : Over S) [GrpObj A] (h : IsAbelianScheme A) (hp : ∀ (K : Type) [Field K] [IsAlgClosed K] (x : Sp K ⟶ S) (y : Over.mk x ⟶ A), y = 1) : IsIso A.hom

theorem **siegPB_zero** {B : Type} [CommRing B] {ζ' : B} {s' : S' ⟶ Sp B} (c : SiegPre 0 k ζ' s') [IsIso c.A.hom] (ζ : o) (s : S ⟶ Sp o) (f : S' ⟶ S) : SiegPB c (zeroSieg k ζ s) f

theorem **tors_eq_one** {ζ : o} (hζ : IsUnit ζ) {s' : S ⟶ Sp o} (c : SiegPre 0 k ζ s') (K : Type) [Field K] [IsAlgClosed K] (x : Sp K ⟶ S) (y : Over.mk x ⟶ c.A) (hy : y ^ k = 1) : y = 1

theorem **isIso_of_count** (H : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Sp K)) [GrpObj B] (g n : ℕ), IsAbelianScheme B → SmoothOfRelativeDimension g B.hom → (n : K) ≠ 0 → Nat.card {x : (𝟙_ (Over (Sp K)) ⟶ B) // x ^ n = 1} = n ^ (2 * g)) {ζ : o} (hζ : IsUnit ζ) (hku : IsUnit (k : o)) (hk : 2 ≤ k) {s' : S ⟶ Sp o} (c : SiegPre 0 k ζ s') [IsReduced S] : IsIso c.A.hom

theorem **hom_ext** {h : ℕ} {ζ : o} (F : SFam h k ζ) {s' : S ⟶ Sp o} (c : SiegPre h k ζ s') (f f' : S ⟶ F.X) (hs : f ≫ F.s = f' ≫ F.s) (h₁ : SiegPB c F.univ f) (h₂ : SiegPB c F.univ f') : f = f'

theorem **zp_hom_ext** (p : ℕ) {R : Type} [CommRing R] (f f' : ZP p 0 →+* R) : f = f'

theorem **exists_sRep** (p : ℕ) : ∃ Ag : SiegelFamily 0 p 0, SRep Ag ∧ Nonempty Ag.X

theorem **exists_arith** (n e p : ℕ) (hn : n.Prime) (hp : p.Prime) (hnp : n ≠ p) (he : 1 ≤ e) : (∃ (o : Type) (_ : CommRing o) (b : TBase o (n ^ e)) (em : o →+* ℂ) (F : SFam 0 (n ^ e) (b.ζ : o)), ArBase em ∧ Nonempty F.X) ∧ ∃ Ag : SiegelFamily 0 p (n ^ e), SRep Ag ∧ Nonempty Ag.X

theorem **exists_section** [IsDomain o] {ζ : o} (F : SFam 0 k ζ) : ∃ f : Sp o ⟶ F.X, f ≫ F.s = 𝟙 _

theorem **exists_pelRep_isPEL** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) {J : Type} (a : J → 𝒪 E) (ρ : ℂ) (hk : k ≠ 0) : ∃ (t : TObj 0 k J (baseC k hk) (𝟙 (Sp ℂ))) (τ : SAB H (Sp ℂ)) (lev : _), PELRep (Test.zeroDatum H) a (zeroBas H) ρ t ∧ IsPEL (Test.zeroDatum H) a (zeroBas H) ρ τ τ ∧ LevUnder (Test.zeroDatum H) a (zeroBas H) t τ lev

theorem **nonempty_of_pelRep** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) {J : Type} (a : J → 𝒪 E) (ρ : ℂ) (hk : k ≠ 0) {s : S ⟶ Sp ℂ} (t : TObj 0 k J (baseC k hk) s) (h : PELRep (Test.zeroDatum H) a (zeroBas H) ρ t) : Nonempty S

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsRetractTensorTriple

theorem **topExists_iff_amp** : TopExists ↔ ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H), 1 ≤ D.p + D.q → ∀ (Xo : ∀ l : Level H, Moduli D l) (n : ℕ), AuxiliaryInertOddPrime H n → ∃ J : ℕ, 1 ≤ J ∧ ∃ T : Top Xo n (n ^ J), T.Amp

theorem **isProper_tensorTriple** : IsProper (tensorTriple H A hA).G.hom

theorem **exists_retract_tensorTriple** : ∃ (i : A ⟶ (tensorTriple H A hA).G) (r : (tensorTriple H A hA).G ⟶ A), i ≫ r = 𝟙 A

theorem **flat_projSpace** (R : Type) [CommRing R] (N : ℕ) : Flat (PSpace.str R N)

theorem **isNormalScheme_projSpace** (R : Type) [CommRing R] [IsDomain R] [UniqueFactorizationMonoid R] (N : ℕ) : Geo.Tor.IsNormalScheme (PSpace R N)

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.ExistsTwoPoints

theorem **secCoord_iff** {σ : Type} (e : σ ≃ Fin (N + 1)) (φ : Y ⟶ PSpace R N) (c : R →+* A) (x : Sp A ⟶ Y) (hx : (x ≫ φ) ≫ PSpace.str R N = spMap c) (v : σ → A) : SecCoord ((01 R N).pull φ) (fun a => Cartier.secPull (coord R N (e a)) φ) x v ↔ ∃ hv : Ideal.span (Set.range (v ∘ e.symm)) = ⊤, x ≫ φ = pt c (v ∘ e.symm) hv

theorem **range_comp_symm** {σ τ : Type} (e : σ ≃ τ) (v : σ → A) : Set.range (v ∘ e.symm) = Set.range v

theorem **projRep_pspace** (R : Type) [CommRing R] (N : ℕ) : ProjRep (⊥ : Ideal (MvPolynomial (Fin (N + 1)) R)) (PSpace.str R N) ((01 R N).pull (𝟙 _)) (fun a => Cartier.secPull (coord R N a) (𝟙 _))

theorem **pt_mem_cosetOpen** {β : K →+* A} {v : Ix g (N' ^ 2) → A} (hv : Ideal.span (Set.range (v ∘ e.symm)) = ⊤) (q : Sp A) : pt β _ hv q ∈ cosetOpen K g n N' e ↔ Coset n N' v q.asIdeal

theorem **exists_two_points** {g n N' M : ℕ} (e : Ix g (N' ^ 2) ≃ Fin (M + 1)) (a : Ix g (N' ^ 2)) (ha : a ≠ 0) : ∃ x y : Sp ℚ ⟶ (mmBot ℚ g n N' e).MM, x ≠ y

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.FiniteHomogeneousInf

theorem **finite_homogeneous_inf** {K σ : Type} [Field K] [Finite σ] (I : Ideal (MvPolynomial σ K)) (e : ℕ) : Module.Finite K ↥(MvPolynomial.homogeneousSubmodule σ K e ⊓ I.restrictScalars K)

theorem **dimDeg_eq_zero_iff** {K σ : Type} [Field K] [Finite σ] (I : Ideal (MvPolynomial σ K)) (e : ℕ) : dimDeg I e = 0 ↔ ∀ F ∈ I, F.IsHomogeneous e → F = 0

theorem **veryDense_primes** (R : Type) [CommRing R] : VeryDense {p : Ideal R | p.IsPrime}

theorem **veryDense_maximals** (R : Type) [CommRing R] [IsJacobsonRing R] : VeryDense {p : Ideal R | p.IsMaximal}

theorem **vanIdeal_unit_one** (K : Type) [Field K] (d : ℕ) : vanIdeal d (fun (_ : Unit) (_ : Unit) => (1 : K)) = ⊥

theorem **cutBy_bot** (K : Type) [Field K] (d : ℕ) : CutBy (⊥ : Ideal (MvPolynomial Unit K)) 0 d (fun (_ : Unit) (_ : Unit) => (1 : K))

theorem **restrict_restrict** {X : Scheme.{0}} {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (a : Γ(X, U)) : resG X h' (resG X h a) = resG X (h'.trans h) a

theorem **secCoord_one** {X : Scheme.{0}} {I : Type} [Unique I] {R : Type} [CommRing R] (x : Sp R ⟶ X) (v : I → R) : SecCoord (Out.Inh.trivLB X) (fun _ : I => unitSec X) x v ↔ IsUnit (v default)

theorem **isQProjOver_id** (R : Type) [CommRing R] : IsQProjOver (𝟙 (Sp R))

theorem **projRep_point** (R : Type) [CommRing R] : ProjRep (⊥ : Ideal (MvPolynomial Unit R)) (𝟙 (Sp R)) (Out.Inh.trivLB _) (fun _ => OnePoint.unitSec _)

theorem **inMM_zero_iff** (n N' : ℕ) {R : Type} [CommRing R] (β : o ⊗+* R) (v : Ix 0 (N' ^ 2) → R) (a : Ix 0 (N' ^ 2)) : InMM n N' (⊥ : Ideal (MvPolynomial (Ix 0 (N' ^ 2)) o)) β v ↔ IsUnit (v a)

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.GoodClass

theorem **TBase.complex_ζ** (k : ℕ) (hk : k ≠ 0) : ((TBase.complex k hk).ζ : ℂ) = Complex.exp (2 * Real.pi * Complex.I / k)

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.H0IsIffNonemptyLinearEquiv

theorem **symB_iff_exists_gen** {A : Over (Sp K)} [GrpObj A] (M : LB A.left) : SymB M ↔ ∃ u : LB.Gen (LB.LinComb M (K

theorem **symB_iff_sym** {A : Over (Sp K)} [GrpObj A] {M : LB A.left} (hM : M.IsLB) : SymB M ↔ Out.Sym A M

theorem **sectionsFinDim_lbPow_O1_projSpace** (N e : ℕ) : SectionsHaveFiniteDimension (PSpace.str K (N + 1)) (lbPow (O1 K (N + 1)) e) (Module.finrank K (homogeneousSubmodule (Fin (N + 2)) K e))

theorem **sectionsFinDim_lbPow_O1_projLine** (e : ℕ) : SectionsHaveFiniteDimension (PSpace.str K 1) (lbPow (O1 K 1) e) (e + 1)

theorem **vol_line** : Vol (PSpace.str K 1) (O1 K 1) 1 1

theorem **not_vol_line_two** : ¬ Vol (PSpace.str K 1) (O1 K 1) 1 2

theorem **vAmp_coord** (N : ℕ) : VAmp (PSpace.str K N) (O1 K N) (coord K N)

theorem **not_vAmp_one** : ¬ VAmp (PSpace.str K 1) (ProjSpace.O1 K 1) (fun _ : Unit => ProjSpace.coord K 1 0)

theorem **torsBasis_gm** {k : ℕ} [NeZero k] {ζ : Kˣ} (hζ : IsPrimitiveRoot ζ k) : TorsBasis (Gm (Sp K)) k ![(gmPts K).symm ζ]

theorem **actsBy_gm_pow** {k : ℕ} [NeZero k] {ζ : Kˣ} (hζ : IsPrimitiveRoot ζ k) (n : ℕ) : ActsBy ![(gmPts K).symm ζ] (𝟙 (Gm (Sp K)) ^ n) !![(n : ZMod k)]

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.IsSeparatedOfIsQProjOver

theorem **isSeparated_of_isQProjOver** {X : Scheme.{0}} {f : X ⟶ Sp o} (h : IsQProjOver f) : IsSeparated f

theorem **SFam.isSeparated** {h : ℕ} {ζ : o} (F : SFam h k ζ) : IsSeparated F.s

theorem **SFam.eq_constPt** {h : ℕ} {ζ : o} (F : SFam h k ζ) (K : Type) [Field K] (x : Sp (DualNumber K) ⟶ F.X) (c : SiegPre h k ζ (x ≫ F.s)) (hs : x ≫ F.s = constPt x ≫ F.s) (h₁ : SiegPB c F.univ x) (h₂ : SiegPB c F.univ (constPt x)) : x = constPt x

theorem **SFam.sRep** {h p M : ℕ} (F : SFam h M (zetaZP p M)) (sm : SmoothOfRelativeDimension (h * (h + 1) / 2) F.s) (qc : QuasiCompact F.s) (pr : F.univ.Principal) (pts unr) : SRep ((F.X, F.s, sm, qc, F.univ, pr, pts, unr) : SiegelFamily h p M)

theorem **PELRep.isSeparated** {D : Datum H} {a : J → 𝒪 E} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] D.La} {ρ : o} {t : TObj g k J b s} (h : PELRep D a bas ρ t) : IsSeparated s

theorem **LevUnder.lev_zero** {D : Datum H} {a : J → 𝒪 E} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] D.La} {t : TObj g k J b s} {τ : SAB H S} {lev : D.La → (𝟙_ (Over S) ⟶ τ.G)} (h : LevUnder D a bas t τ lev) : lev 0 = 1

theorem **LevUnder.lev_bas_pow** {D : Datum H} {a : J → 𝒪 E} {bas : (Fin (2 * g) → ℤ) ≃ₗ[ℤ] D.La} {t : TObj g k J b s} {τ : SAB H S} {lev : D.La → (𝟙_ (Over S) ⟶ τ.G)} (h : LevUnder D a bas t τ lev) (i : Fin (2 * g)) : lev (bas (Pi.single i 1)) ^ k = 1

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.NotGoodClassZeroSucc

theorem **lbIso_of_g** {X : Scheme.{0}} (L M : LB X) (hL : ∀ i j, L.g i j = 1) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **zero_prim** {J : Type} (b : TBase o k) (s : S ⟶ Sp o) (w w' : J) : (zeroObj J b s).Prim w w'

theorem **zero_tmap** {J : Type} (b : TBase o k) (s : S ⟶ Sp o) (s' : S' ⟶ Sp o) (f : S' ⟶ S) : TMap (zeroObj J b s') (zeroObj J b s) f

theorem **zero_hasTh** {J : Type} {N : ℕ} (hN : Odd N) (b : TBase o (N ^ 2)) {R : Type} [CommRing R] (β : o ⊗+* R) (x : Sp R ⟶ S) (s : S ⟶ Sp o) : (zeroObj J b s).HasTh x (fun _ => 1)

theorem **zero_fromC** {J : Type} (b : TBase ℂ k) : fromSet (RingHom.id ℂ) (fun _ => True) (zeroObj J b (spMap (RingHom.id ℂ))) = Set.univ

theorem **goodClass_empty** {g N : ℕ} {J : Type} {b : TBase o (N ^ 2)} (em : o ⊗+* ℂ) (Mat κ) (w w' : J) (d : ℕ) : GoodClass (g

theorem **odd_of_thetaFam** {g n N : ℕ} {J : Type} {b : TBase o (N ^ 2)} {em : o ⊗+* ℂ} {C : TObj g (N ^ 2) J b (spMap em) → Prop} (F : ThetaFam n em C) : Odd N

theorem **isIso_of_dim_zero** (A : Over S) [GrpObj A] (h : IsAbelianScheme A) (hd : SmoothOfRelativeDimension 0 A.hom) : IsIso A.hom

theorem **lbIso_of_gen** {X : Scheme.{0}} (L M : LB X) (F : X ⟶ X) (hF : F = 𝟙 X) (t : LB.Gen (L.pull F)) (ht : t.IsGen) (hM : ∀ i j, M.g i j = 1) : LBIso L M

theorem **tmap_zero** {J : Type} {b : TBase o k} (s : S ⟶ Sp o) {s' : S' ⟶ Sp o} (t : TObj 0 k J b s') (f : S' ⟶ S) : TMap t (zeroObj J b s) f

theorem **zero_hasTh_unit** {J : Type} {N : ℕ} (hN : Odd N) (b : TBase o (N ^ 2)) {R : Type} [CommRing R] (β : o →+* R) (x : Sp R ⟶ S) (s : S ⟶ Sp o) (v : Ix 0 (N ^ 2) → R) (c : Rˣ) (hv : ∀ a, v a = c) : (zeroObj J b s).HasTh x v

theorem **linComb_g_one** {X Y : Scheme.{0}} (M : LB X) (hg : ∀ i j, M.g i j = 1) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (j j' : (LB.LinComb M φ ε).ι) : (LB.LinComb M φ ε).g j j' = 1

theorem **zero_shape** {J : Type} (b : TBase o k) (s : S ⟶ Sp o) (Mat κ) : (zeroObj J b s).Shape Mat κ

theorem **linComb_iso_triv** {X X' Y : Scheme.{0}} {L : LB X'} (hL : L.IsLB) {F : X' ⟶ X} (H : LBIso L ((trivLB X).pull F)) {K : Type} [Fintype K] (ψ : K → (Y ⟶ X')) (ε : K → ℤ) {M : LB Y} (hM : M.IsLB) (hg : ∀ i j, M.g i j = 1) : LBIso (LB.LinComb L ψ ε) M

theorem **shape_prim_of_dim_zero** {J : Type} {b : TBase o k} {R : Type} [CommRing R] {s : Sp R ⟶ Sp o} (t : TObj 0 k J b s) (Mat κ) (w w' : J) : t.Shape Mat κ ∧ t.Prim w w'

theorem **goodClass_zero** {N : ℕ} (hN : Odd N) {J : Type} (b : TBase o (N ^ 2)) (em : o →+* ℂ) (Mat κ) (w w' : J) : GoodClass (g

theorem **odd_of_goodClass** {g N : ℕ} {J : Type} {b : TBase o (N ^ 2)} {em : o →+* ℂ} {C : TObj g (N ^ 2) J b (spMap em) → Prop} {Mat κ} {w w' : J} {d : ℕ} (h : GoodClass em C Mat κ w w' d) (t) (ht : C t) : Odd N

theorem **nonempty_of_thetaFam** {n N : ℕ} {J : Type} {b : TBase o (N ^ 2)} {em : o →+* ℂ} {C : TObj 0 (N ^ 2) J b (spMap em) → Prop} (F : ThetaFam n em C) (t) (ht : C t) : Nonempty F.MM

theorem **not_goodClass_zero_succ** {N : ℕ} {J : Type} {b : TBase o (N ^ 2)} {em : o →+* ℂ} {C : TObj 0 (N ^ 2) J b (spMap em) → Prop} {Mat κ} {w w' : J} {d : ℕ} (hne : ∀ (t : TObj 0 (N ^ 2) J b (spMap em)) v, t.HasTh (𝟙 _) v → v 0 ≠ 0) (t) (ht : C t) : ¬ GoodClass em C Mat κ w w' (d + 1)

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.NotIsGaloisCoverTrivial

private lemma **spMap_comp_moduli_spaces** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap (g.comp f) = spMap g ≫ spMap f

private lemma **spMap_surjective'** {A B : Type} [CommRing A] [CommRing B] (y : Sp B ⟶ Sp A) : ∃ f : A →+* B, spMap f = y

private lemma **spMap_inj_lemma_3** {A B : Type} [CommRing A] [CommRing B] {f g : A →+* B} (h : spMap f = spMap g) : f = g

theorem **isGaloisCover_field** (K L : Type) [Field K] [Field L] [Algebra K L] [FiniteDimensional K L] [IsGalois K L] : IsGaloisCover (L ≃ₐ[K] L) (fun σ => spMap (σ : L →+* L)) (spMap (algebraMap K L)) where

theorem **isGaloisCover_complex** : IsGaloisCover (ℂ ≃ₐ[ℝ] ℂ) (fun σ => spMap (σ : ℂ →+* ℂ)) (spMap (algebraMap ℝ ℂ)) ∧ Nat.card (ℂ ≃ₐ[ℝ] ℂ) = 2

theorem **not_isGaloisCover_trivial** : ¬ IsGaloisCover ℤˣ (fun _ => 𝟙 (Sp ℂ)) (𝟙 (Sp ℂ))

theorem **dirMap_fst** {K : Type} [Field K] {n : ℕ} (lam : (Fin n → K) →ₗ[K] K) (x : TrivSqZeroExt K (Fin n → K)) : (dirMap lam x).fst = x.fst ∧ (dirMap lam x).snd = lam x.snd

theorem **densePrimes_univ_iff** (R : Type) [CommRing R] : DenseP (Set.univ : Set (Ideal R)) ↔ _root_.IsReduced R

theorem **not_denseP_two** : ¬ DenseP ({Ideal.span {2}} : Set (Ideal ℤ))

theorem **odd_of_wfam** {o : Type} [CommRing o] {g : ℕ} {J : Type} {n N' : ℕ} {b : TBase o (N' ^ 2)} {em : o →+* ℂ} {C : TObj g (N' ^ 2) J b (spMap em) → Prop} {Mat : J → Matrix (Fin (2 * g)) (Fin (2 * g)) ℤ} {κ : ℕ} (W : WFam n em C Mat κ) : Odd N'

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.NotPelRepOfTwoPoints

theorem **translT_zero** {T : Over S} (x : T ⟶ (zeroPolAV S).A) : (Out.translT (zeroPolAV S).A x).left = 𝟙 _

theorem **exists_gen** {X : Scheme.{0}} (L : LB X) (hL : ∀ i j, L.g i j = 1) : ∃ t : L.Gen, t.IsGen ∧ ∀ i, t.s i = 1

theorem **corr_comm** {X Y : Scheme.{0}} (L : LB X) [Unique L.ι] {K : Type} [Fintype K] (φ φ' : K → (Y ⟶ X)) (ε : K → ℤ) (a : LB.Gen (LB.LinComb L φ ε)) (b : LB.Gen (LB.LinComb L φ' ε)) (h h' : Y ⟶ Y) (hh : h = 𝟙 Y) (hh' : h' = 𝟙 Y) (c : LB.Gen (LB.LinComb L (Y

theorem **weilIs_zero** (L : LB (zeroPolAV S).A.left) [Unique L.ι] (hL : ∀ i j, L.g i j = 1) (n : ℤ) {T : Over S} (x y : T ⟶ (zeroPolAV S).A) : Out.WeilIs (zeroPolAV S).A L n x y 1

theorem **principal_zeroSieg** (k : ℕ) {B : Type} [CommRing B] (ζ : B) (s : S ⟶ Sp B) : (zeroSieg k ζ s).Principal

theorem **pelRep_zero** [_root_.IsReduced o] (D : Datum H) (a : J → 𝒪 E) (bas : (Fin (2 * 0) → ℤ) ≃ₗ[ℤ] D.La) (ρ : o) (b : TBase o k) : PELRep D a bas ρ (zeroObj J b (𝟙 (Sp o))) where

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.SectionsHaveFiniteDimension

theorem **SectionsHaveFiniteDimension.le** {p : X ⟶ Sp K} {M : LB X} {r r' : ℕ} (h : SectionsHaveFiniteDimension p M r) (h' : SectionsHaveFiniteDimension p M r') : r ≤ r'

theorem **SectionsHaveFiniteDimension.unique** {p : X ⟶ Sp K} {M : LB X} {r r' : ℕ} (h : SectionsHaveFiniteDimension p M r) (h' : SectionsHaveFiniteDimension p M r') : r = r'

theorem **Vol.unique** {p : X ⟶ Sp K} {M : LB X} {g : ℕ} {δ δ' : ℝ} (h : Vol p M g δ) (h' : Vol p M g δ') : δ = δ'

theorem **SectionsHaveFiniteDimension.one_of_isIso** (p : X ⟶ Sp K) [IsIso p] {M : LB X} [Unique M.ι] (hU : M.U default = ⊤) (hg : ∀ i j, M.g i j = 1) : SectionsHaveFiniteDimension p M 1

theorem **SectionsHaveFiniteDimension.trivial_point** (K : Type) [Field K] : SectionsHaveFiniteDimension (𝟙 (Sp K)) (Out.Inh.trivLB (Sp K)) 1

theorem **Vol.trivial_point** (K : Type) [Field K] : Vol (𝟙 (Sp K)) (Out.Inh.trivLB (Sp K)) 0 1

theorem **vAmp_iff_veryAmpleBy** (p : X ⟶ Sp K) (M : LB X) {m : ℕ} (s : Fin (m + 1) → M.Sec) : VAmp p M s ↔ AbLaw.VeryAmpleBy p M s

theorem **VAmp.comp_surjective** {p : X ⟶ Sp K} {M : LB X} {σ τ : Type} {th : σ → M.Sec} (h : VAmp p M th) {e : τ → σ} (he : Function.Surjective e) : VAmp p M (fun i => th (e i))

theorem **vAmp_comp_equiv** (p : X ⟶ Sp K) (M : LB X) {σ τ : Type} (th : σ → M.Sec) (e : τ ≃ σ) : VAmp p M (fun i => th (e i)) ↔ VAmp p M th

theorem **vAmp_iff_veryAmpleBy_equiv** (p : X ⟶ Sp K) (M : LB X) {σ : Type} (th : σ → M.Sec) {m : ℕ} (e : Fin (m + 1) ≃ σ) : VAmp p M th ↔ AbLaw.VeryAmpleBy p M (fun i => th (e i))

theorem **VAmp.exists_veryAmpleBy** {p : X ⟶ Sp K} {M : LB X} {σ : Type} [Finite σ] [Nonempty σ] {th : σ → M.Sec} (h : VAmp p M th) : ∃ (m : ℕ) (e : Fin (m + 1) ≃ σ), AbLaw.VeryAmpleBy p M (fun i => th (e i))

theorem **VAmp.trivial_point** (K : Type) [Field K] : VAmp (𝟙 (Sp K)) (Out.Inh.trivLB (Sp K)) (fun _ : Fin 1 => AbLaw.oneSec (Sp K))

theorem **MapDeg.finrank_eq** {ψ : X ⟶ Y} {D : ℕ} (h : MapDeg ψ D) (U : Y.Opens) (hU : IsAffineOpen U) (hne : (U : Set Y).Nonempty) : letI

theorem **MapDeg.of_isFinite** {ψ : X ⟶ Y} [IsFinite ψ] {D : ℕ} (h : ∀ U : Y.Opens, IsAffineOpen U → (U : Set Y).Nonempty → letI

theorem **MapDeg.unique** [Nonempty Y] {ψ : X ⟶ Y} {D D' : ℕ} (h : MapDeg ψ D) (h' : MapDeg ψ D') : D = D'

theorem **MapDeg.id_one** (X : Scheme.{0}) : MapDeg (𝟙 X) 1

theorem **TorsBasis.pos** (h : TorsBasis A k pt) : 0 < k

theorem **TorsBasis.pow_eq_one** (h : TorsBasis A k pt) (i : Fin m) : pt i ^ k = 1

theorem **TorsBasis.existsUnique** (h : TorsBasis A k pt) {y : Out.baseOver (Sp K) ⟶ A} (hy : y ^ k = 1) : ∃! c : Fin m → ZMod k, Out.linComb pt (fun i => (c i).val) = y

theorem **ActsBy.pos** {f : A ⟶ A} {Mat : Matrix (Fin m) (Fin m) (ZMod k)} (h : ActsBy pt f Mat) : 0 < k

theorem **ActsBy.comp_eq** {f : A ⟶ A} {Mat : Matrix (Fin m) (Fin m) (ZMod k)} (h : ActsBy pt f Mat) (i : Fin m) : pt i ≫ f = Out.linComb pt (fun i' => (Mat i' i).val)

theorem **ActsBy.id_one** [IsCommMonObj A] (hk : 0 < k) (hp : ∀ i, pt i ^ k = 1) : ActsBy pt (𝟙 A) (1 : Matrix (Fin m) (Fin m) (ZMod k))

theorem **ActsBy.comp** [IsCommMonObj A] (hp : ∀ i, pt i ^ k = 1) {f g : A ⟶ A} [IsMonHom g] {F G : Matrix (Fin m) (Fin m) (ZMod k)} (hf : ActsBy pt f F) (hg : ActsBy pt g G) : ActsBy pt (f ≫ g) (G * F)

theorem **TorsBasis.of_level** [IsAlgClosed K] (P : Out.PolAV (Sp K)) {k : ℕ} (hk : 0 < k) {ze : Γ(Sp K, ⊤)ˣ} (lev : Out.Level P k ze) : TorsBasis P.A k lev.pt

theorem **FormIs.isMonHom** {P : Out.PolAV (Sp K)} {M : LB P.A.left} {s : P.A ⟶ P.A} (h : FormIs P M s) : IsMonHom s

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.SystemPrimesDistinctLevelsGeThree

theorem **isProper_zeroTriple** (S : Scheme.{0}) : IsProper (zeroTriple H S).G.hom

theorem **abLocus_eq_univ** {S : Scheme.{0}} (G : Over S) (h : IsProper G.hom) : abLocus G = Set.univ

theorem **abOpen_and_splitFib** {S : Scheme.{0}} (t : SAB H S) (h : IsProper t.G.hom) : AbOpen t.G ∧ SplitFib t.G

theorem **good_zeroTriple** (S : Scheme.{0}) : Good (zeroTriple H S)

theorem **exists_pull** {S S' : Scheme.{0}} (f : S' ⟶ S) (t : SAB H S) : ∃ t' : SAB H S', Nonempty (SABMap t' t f)

theorem **above_refl** (C : Cpt c t₀ β) : ∃ ν : SABMap t₀ t₀ (𝟙 U), ν.φ = 𝟙 _ ∧ Nonempty (C.Above C (RingHom.id O) (𝟙 U) ν) ∧ C.ExtendsAut (Iso.refl U) ν

theorem **abLocus_open_part** (C : Cpt c t₀ β) : abLocus t₀.G = Set.univ

theorem **exists_above_of_finite** {O F : Type} [CommRing O] [Field F] (β : O →+* F) {X X' : Scheme.{0}} (s : X ⟶ Sp O) (g : X' ⟶ X) [IsFinite g] (t : SAB H X) (hn : Geo.Tor.IsNormalScheme X) (hn' : Geo.Tor.IsNormalScheme X') (hp : IsProjOver s) (hf : Flat s) (hf' : Flat (g ≫ s)) (ha : IsProper t.G.hom) : ∃ (t₀ : SAB H (pullback s (spMap β))) (t₀' : SAB H (pullback (g ≫ s) (spMap β))) (C : Cpt (pullback.snd s (spMap β)) t₀ β) (C' : Cpt (pullback.snd (g ≫ s) (spMap β)) t₀' β) (r : pullback (g ≫ s) (spMap β) ⟶ pullback s (spMap β)) (ν : SABMap t₀' t₀ r) (a : C'.Above C (RingHom.id O) r ν), C.X = X ∧ C'.X = X' ∧ HEq a.π g ∧ HEq C.t t

theorem **isProjOver_id** (R : Type) [CommRing R] : IsProjOver (𝟙 (Sp R))

theorem **exists_gaussian** : IsDedFrac (algebraMap ℤ ℚ) ∧ ∃ (U U' : Scheme.{0}) (c : U ⟶ Sp ℚ) (c' : U' ⟶ Sp ℚ) (t₀ : SAB HostModel.host U) (t₀' : SAB HostModel.host U') (C : Cpt c t₀ (algebraMap ℤ ℚ)) (C' : Cpt c' t₀' (algebraMap ℤ ℚ)) (r : U' ⟶ U) (ν : SABMap t₀' t₀ r) (a : C'.Above C (RingHom.id ℤ) r ν), C.X = Sp ℤ ∧ C'.X = Sp GaussianInt ∧ HEq a.π (spMap (algebraMap ℤ GaussianInt)) ∧ Good C.t

theorem **system_primes_distinct_levels_ge_three** {D : Datum H} {Xo : ∀ l : Level H, Moduli D l} (S : System Xo) : S.n true ≠ S.n false ∧ ∀ i, 3 ≤ S.n i ^ S.J i

theorem **isPrimeToPart_unique** {l₀ l₀' l : Level H} {p : ℕ} (h : IsPrimeToPart l₀ l p) (h' : IsPrimeToPart l₀' l p) : l₀ = l₀'

theorem **exists_isInc** (l : Level H) (n : ℕ) : ∃ β : RZ H n l.A →+* KA l, IsInc (RZ H n l.A).subtype (KA l).val.toRingHom β

Development.AbelianVarietyTheory.Definitions.ModuliSpaces.TopAmpExistsIffTopExists

theorem **isProjOver_powObj** {R : Type} [CommRing R] (A : Over (Sp R)) (h : IsProjOver A.hom) : ∀ n, IsProjOver (Out.powObj A n).hom | 0 => Inhab.isProjOver_id R | n + 1 => ProjSpace.isProjOver_pullback h (isProjOver_powObj A h n) end Model section Zero variable {D : Datum H} {n N : ℕ} open RankZero end Zero theorem topAmpExists_iff_topAmpSt : TopAmpExists ↔ GlobalTheory.LN.TopAmpSt

theorem **topAmpExists_iff_topExists** : TopAmpExists ↔ TopExists

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.AcrisTorsionFreeComplete

theorem **IsStd.of_tied** {P : LocPrime l} {AI : LocAinf l P} (T : LocTied l P AI) : IsStd P AI

theorem **locAinf_actC** {P : LocPrime l} (AI : LocAinf l P) (g : GQ l) (x : ℂ_[l]) : AI.actC g x = Per.σC g x

theorem **locAinf_actT** {P : LocPrime l} (AI : LocAinf l P) (g : GQ l) (x : OCb l) : AI.actT g x = Per.σF g x

theorem **locAinf_actW** {P : LocPrime l} (AI : LocAinf l P) (g : GQ l) (a : Ainf l) : AI.actW g a = Per.σA g a

theorem **locAinf_xi** {P : LocPrime l} (AI : LocAinf l P) : AI.ξ = xi AI.ε

theorem **locAinf_mu** {P : LocPrime l} (AI : LocAinf l P) : AI.μ = mu AI.ε

theorem **isUnit_l_BdRp** : IsUnit ((l : ℕ) : Per.BdRp l)

theorem **isUnit_l_Bcris** : IsUnit ((l : ℕ) : Per.Bcris l)

theorem **plusLift_alg** {S : Type} [CommRing S] (f : Per.Acris l →+* S) (h : IsUnit ((l : ℕ) : S)) (x : Per.Acris l) : plusLift f h (algebraMap _ _ x) = f x

theorem **isUnit_l_BcrisP** : IsUnit ((l : ℕ) : BcrisP l)

theorem **plusToB_alg** (x : Per.Acris l) : plusToB l (algebraMap _ _ x) = algebraMap (Per.Acris l) (Per.Bcris l) x

theorem **phiP_alg** (x : Per.Acris l) : phiP l (algebraMap _ _ x) = algebraMap _ _ (phiA l x)

theorem **sigP_alg** (g : Per.Gq l) (x : Per.Acris l) : sigP g (algebraMap _ _ x) = algebraMap _ _ (sigA g x)

theorem **plus_ext** {S : Type} [CommRing S] {f f' : BcrisP l →+* S} (h : ∀ x : Per.Acris l, f (algebraMap _ _ x) = f' (algebraMap _ _ x)) : f = f'

theorem **plusToB_phi** (x : BcrisP l) : plusToB l (phiP l x) = Per.φB l (plusToB l x)

theorem **plusToB_sig** (g : Per.Gq l) (x : BcrisP l) : plusToB l (sigP g x) = Per.σB g (plusToB l x)

theorem **jP_alg** (x : Per.Acris l) : C.jP (algebraMap _ _ x) = C.j x

theorem **ι_plusToB** (x : BcrisP l) : C.ι (plusToB l x) = algebraMap (Per.BdRp l) (Per.BdR l) (C.jP x)

theorem **ker_theta** : RingHom.ker (Per.θ l) = Ideal.span {xi C.ε}

theorem **acris_torsionFree_complete** : (∀ x : Per.Acris l, ((l : ℕ) : Per.Acris l) * x = 0 → x = 0) ∧ IsAdicComplete (Ideal.span {((l : ℕ) : Per.Acris l)}) (Per.Acris l)

theorem **thetaD_onto_ker** : Function.Surjective (Per.θD l) ∧ RingHom.ker (Per.θD l) = Per.FilP l 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.AdIPowLe

theorem **adI_pow_le** {B B' : Obj l R} (f : Hom B B') (m : ℕ) : adI B ^ m ≤ (adI B' ^ m).comap f.f

theorem **qmap_mk** {B B' : Obj l R} (f : Hom B B') (m : ℕ) (a : B.A) : qmap f m (Ideal.Quotient.mk _ a) = Ideal.Quotient.mk _ (f.f a)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.AdmQAdmissibleConjugatePairPlaces

theorem **exists_end_reads** {K M A ι : Type} [Field K] [AddCommGroup M] [Module K M] [AddCommGroup A] [Module ℂ A] {e : K →+* ℂ} (r : M →ₛₗ[e] A) (hr : Function.Injective r) (S : ι → A →ₗ[ℂ] A) : ∃ T : Module.End K M, ∀ f, (∃ g, ∀ a, r g = S a (r f)) → ∀ a, r (T f) = S a (r f)

theorem **hecke_apply** (k : ℕ) (P : Pl D.host.E) (R : Finset (GA D (Wc D j))) (z : (AE D)ˣ) (F : GA D (Wc D j) → ℂ) (x : GA D (Wc D j)) : ρ.hecke i j b k P R z F x = (((ρ.B i j b).mu z : Circle) : ℂ)⁻¹ * (((normV P : ℕ) : ℂ) ^ (k * ((Vc D j).rank / 2)))⁻¹ * ∑ γ ∈ R, F (x * γ)

theorem **heckeRead_iff** : HeckeRead rd ↔ ∀ k P, HkIdx i k P → ∀ f, HeckeAt rd k P f (Y.hkF i _ b k P f)

theorem **heckeRead_iff_tie** : HeckeRead rd ↔ HeckeTie rd ∧ RatHecke rd

theorem **HeckeTie.heckeRead** (h : HeckeTie rd) (hR : RatHecke rd) : HeckeRead rd

theorem **HeckeTie.eq** (h : HeckeTie rd) (hinj : Function.Injective rd) {k P} (hk : HkIdx i k P) {α R z} (d : ρ.IsHeckeDatum i j b k P α R z) {f g : Y.Fm i ((j : ℕ) + 1) b} (hg : HeckeAt rd k P f g) : Y.hkF i _ b k P f = g

theorem **ratHecke_of_range** (h1 : ∀ k P, HkIdx i k P → ∀ α R z, ρ.IsHeckeDatum i j b k P α R z → ∀ f, ρ.hecke i j b k P R z (rd f) ∈ Set.range rd) (h2 : ∀ k P, HkIdx i k P → ∀ α R z α' R' z', ρ.IsHeckeDatum i j b k P α R z → ρ.IsHeckeDatum i j b k P α' R' z' → ∀ f, ρ.hecke i j b k P R z (rd f) = ρ.hecke i j b k P R' z' (rd f)) : RatHecke rd

theorem **exists_hecke** (hinj : Function.Injective rd) (k : ℕ) (P : Pl D.host.E) : ∃ T : Module.End (FractionRing (X.O i)) (Y.Fm i ((j : ℕ) + 1) b), ∀ f, (∃ g, HeckeAt rd k P f g) → HeckeAt rd k P f (T f)

theorem **CellTie.injective_rdK** {hρ : ρ.Good} (T : CellTie hρ Y rd) (k : ℕ) : Function.Injective (rdK rd hρ k)

theorem **admQ_admissibleConjugatePairPlaces** (i : X.Tw) (Q : Finset (Pl D.host.E)) (M : ℕ) : AdmQ X i (admissibleConjugatePairPlaces X i Q M) M

theorem **InRange.twSet** {i : X.Tw} {Q : Finset (Pl D.host.E)} (h : InRange X i Q) : TWSet X i Q

theorem **twSet_of_twOK** {i : X.Tw} {k : ℕ} {Q : Finset (Pl D.host.E)} {ψ} (h : TwOK (X

theorem **delU_off** (hQ : TWSet X i Q) (σ : G D.host.E N.Om) (v : ɪQ) (hv : v.1 ∉ admissibleConjugatePairPlaces X i Q M) : Multiplicative.toAdd (Y.delU i Q M σ) v = 0

theorem **delU_on** (σ : G D.host.E N.Om) (v : ɪQ) (hv : v.1 ∈ admissibleConjugatePairPlaces X i Q M) : Multiplicative.toAdd (Y.delU i Q M σ) v = Multiplicative.toAdd (GalChar.delOf D.host X.cc (X.l i) M (GalChar.pair D.host v.1) σ)

theorem **delU_eq_coord** (hQ : TWSet X i Q) (σ : G D.host.E N.Om) (v : ɪQ) : Multiplicative.toAdd (Y.delU i Q M σ) v = Multiplicative.toAdd (coord Y i (admissibleConjugatePairPlaces X i Q M) M v.1 σ)

theorem **delU_eq_spread** (hQ : TWSet X i Q) {ch : Set (Pl D.host.E)} (hch : IsChoice ch) (σ : G D.host.E N.Om) : Multiplicative.toAdd (Y.delU i Q M σ) = spread Y i Q M ch fun P => Multiplicative.toAdd (coord Y i (admissibleConjugatePairPlaces X i Q M) M P.1 σ)

theorem **delU_eq_one** (hQ : TWSet X i Q) (h0 : admissibleConjugatePairPlaces X i Q M = ∅) : Y.delU i Q M = 1

theorem **heckeTie_withHecke** (i j b) : HeckeTie (Y

theorem **withHecke_hkF_zero** (i n b P) : (withHecke rd hF).hkF i n b 0 P = Y.hkF i n b 0 P

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.AdmissibleEmptyOfGood

theorem **realGL_rat** {θ : InfFam D} (hθ : IsInfFam θ) {V : HSp D} {γ : GQ V} {γE : GL (Fin V.rank) D.host.E} (h : Bridge.IsRatMat V γ γE) : realGL θ V γ = embGL (tauE D) γE

theorem **series_apply** (F : Frame S T) (ω : PairRep S T) (f : WO S T → ℂ) (h : GA D S) (P : PolF D T.rank S.rank) (g : GA D T) : series F ω f h P g = ω (h, g) (F.fock f P) 0

theorem **mem_lawForms** {G Γ W : Type} [Group G] [Group Γ] [AddCommGroup W] [Module ℂ W] {ι : Γ →* G} {ρ : Γ →* Module.End ℂ W} {J : Subgroup G} {χ : J →* ℂˣ} {F : G → W} : F ∈ lawForms ι ρ J χ ↔ ∀ (γ : Γ) (h : G) (k : J), F (ι γ * h * k) = (χ k : ℂ) • ρ γ (F h)

theorem **admissible_empty_of_good** {B : Base S T} (h : B.Good) : B.admitted ∅

theorem **admitted.mono** {B : Base S T} {Q Q' : Finset (Pl D.host.E)} (h : B.admitted Q) (hQ : Q' ⊆ Q) : B.admitted Q'

theorem **vecAt_empty** (φ : ∀ PP : ɪ(∅ : Finset (Pl D.host.E)), Matrix (Fin T.rank) (Fin S.rank) (kP PP.1) → ℂ) : B.vecAt ∅ φ = B.vecBase

theorem **isThetaAt_vecAt** {B : Base S T} (h : B.Good) {Q : Finset (Pl D.host.E)} (hQ : B.admitted Q) (φ : ∀ PP : ɪQ, Matrix (Fin T.rank) (Fin S.rank) (kP PP.1) → ℂ) : IsThetaAt B.F.ψ (symplO S T) (heisO B.F.ψ S T) (halfIdem B.G.Pg (B.de.Q ∪ Q)) B.F.lat 1 B.F.f₀ (B.F.datumFn B.G (B.de.Q ∪ Q) (B.locFn Q φ)) (B.vecAt Q φ)

theorem **isThetaFunO_fock_vecAt** {B : Base S T} (h : B.Good) {Q : Finset (Pl D.host.E)} (hQ : B.admitted Q) (φ : ∀ PP : ɪQ, Matrix (Fin T.rank) (Fin S.rank) (kP PP.1) → ℂ) (P : PolF D T.rank S.rank) : IsThetaFunO B.F.ψ S T (B.F.fock (B.vecAt Q φ) P)

theorem **xiCoef_apply** (ω : PairRep S T) (u : UnivTwist D C) (Q : Finset (Pl D.host.E)) (Φ : ∀ PP : Pl D.host.E, C → Matrix (Fin T.rank) (Fin S.rank) (kP PP) → ℚ) (h : GA D S) (P : PolF D T.rank S.rank) (a : ɪQ → C) (g : GA D T) : B.xiCoef ω u Q Φ h P a g = series B.F ω (B.vecFam Q Φ (a − Multiplicative.toAdd (detChar T (u.delQ Q) g))) h P g

theorem **isAdmFam_univFam** (u : UnivTwist D C) (Q : Finset (Pl D.host.E)) (β : ∀ PP : Pl D.host.E, DetBases (kP PP) T.rank S.rank) : IsAdmFam u Q (univFam u β)

theorem **finPt_mem_levJ** (ψ : Twist D Circle) {k : GA D S} (hk : k ∈ GfC S) : finPt S k ∈ B.levJ ψ ↔ k ∈ Lev.levG S B.F.βS (B.levS ψ)

theorem **levLam_finPt** (ψ : Twist D Circle) {k : GA D S} (hk : finPt S k ∈ B.levJ ψ) : ((B.levLam ψ (_, hk) : ℂˣ) : ℂ) = Lev.levChi S B.F.βS (B.levS ψ) k

theorem **mem_formsC** {ψ : Twist D Circle} {f : Bridge.Gf S → PolF D T.rank S.rank} : f ∈ B.formsC ψ ↔ (∀ h, f h ∈ B.valSpace) ∧ B.HasLaw ψ f

theorem **mem_mtLat** {ψ : Twist D Circle} {f : Bridge.Gf S → PolF D T.rank S.rank} : f ∈ B.mtLat σ emb O ψ ↔ f ∈ B.formsC ψ ∧ ∀ h, f h ∈ B.coefLat σ emb O h

theorem **classSum_apply** (ω : PairRep S T) (ψ : Twist D Circle) {I : Type} [Fintype I] (h : I → GA D S) (f : Bridge.Gf S → PolF D T.rank S.rank) : B.classSum ω ψ h f = ∑ i, ((Nat.card (Bridge.stab (Bridge.ratF S) (B.levJ ψ) (finPt S (h i))) : ℕ) : ℂ)⁻¹ • series B.F ω (B.vecFlat ψ) (h i) (f (finPt S (h i)))

theorem **act_mem_valSpace** (g : GA D S) {P : PolF D T.rank S.rank} (h : P ∈ B.valSpace) : B.act g P ∈ B.valSpace

theorem **mem_latAt_iff** (hF : B.F.Good) {γ : GQ S} {γE : GL (Fin S.rank) D.host.E} (hγ : Bridge.IsRatMat S γ γE) (h k : Bridge.Gf S) (hk : Formalism.UG.bc (Lev.swapAE D) k.1 ∈ St B.F.βS) (x : Fin S.rank → D.host.E) : x ∈ B.latAt (Bridge.ratF S γ ∗ h ∗ k) ↔ (γE⁻¹).val ∗ᵥ x ∈ B.latAt h

theorem **act_mem_coefLat** (hF : B.F.Good) (hc : IsCoefRing σ emb O) (γ : GQ S) (h k : Bridge.Gf S) (hk : Formalism.UG.bc (Lev.swapAE D) k.1 ∈ St B.F.βS) {P : PolF D T.rank S.rank} (hP : P ∈ B.coefLat σ emb O h) : B.act γ P ∈ B.coefLat σ emb O (Bridge.ratF S γ ∗ h ∗ k)

theorem **mem_St_of_mem_levJ** {ψ : Twist D Circle} {k : Bridge.Gf S} (hk : k ∈ B.levJ ψ) : Formalism.UG.bc (Lev.swapAE D) k.1 ∈ St B.F.βS

theorem **vecFlat_triv** : B.vecFlat (Twist.triv Circle ∅) = B.vecBase

theorem **gen_triv** (ω : PairRep S T) (h : GA D S) (P : PolF D T.rank S.rank) : B.gen ω (Twist.triv Circle ∅) h P = B.theta ω h P

theorem **genSet_triv** (ω : PairRep S T) (h : GA D S) (P : PolF D T.rank S.rank) : B.genSet ω (Twist.triv Circle ∅) h P = {B.theta ω h P}

theorem **xiCoef_empty** {C : Type} [AddCommGroup C] [Fintype C] (ω : PairRep S T) (u : UnivTwist D C) (Φ : ∀ PP : Pl D.host.E, C → Matrix (Fin T.rank) (Fin S.rank) (kP PP) → ℚ) (h : GA D S) (P : PolF D T.rank S.rank) (a : ɪ(∅ : Finset (Pl D.host.E)) → C) : B.xiCoef ω u ∅ Φ h P a = B.theta ω h P

theorem **exists_onb_posDef** {n : ℕ} {G : Matrix (Fin n) (Fin n) ℂ} (hG : G.IsHermitian) (hd : IsUnit G.det) (hs : Formalism.UG.HasSignature G n 0) : ∃ B : GL (Fin n) ℂ, IsONBasis G 1 B.val n

theorem **exists_frame_good** (S T : HSp D) (hS : S.SigOK) (hpos : S.IsPosDef) (hT : T.SigOK) (hr : T.rank ≤ S.rank) : ∃ F : Frame S T, F.Good

theorem **exists_base_good** (D : D7) (k : ℕ) : ∃ (S T : HSp D) (B : Base S T), S.rank = 2 ∗ k + 2 ∧ T.rank = 2 ∗ k + 1 ∧ B.Good ∧ B.admitted ∅ ∧ ∀ τ, B.w.ap τ 0 = 1

theorem **vec_empty** {F : Frame S T} (hF : F.Good) : F.vec (TypeDat.empty D T.rank) ∅ (fun _ => 1) = F.f₀

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.AevalRootEqZeroIff

theorem **_root_.GlobalTheory.UTr.IsCsr.conj_mul_coeff_zero** {P : B[X]} (h : _root_.GlobalTheory.UTr.IsCsr s P) : s (P.coeff 0) ∗ P.coeff 0 = 1

theorem **isCsr_X_sub_C** [Nontrivial B] {u : B} : IsCsr s (X − C u) ↔ s u ∗ u = 1

theorem **IsCsr.mul** {P Q : B[X]} (hP : IsCsr s P) (hQ : IsCsr s Q) : IsCsr s (P ∗ Q)

theorem **IsCsr.map** {P : B[X]} (hP : IsCsr s P) (φ : B →+∗ B') (hφ : ∀ b, φ (s b) = s' (φ b)) : IsCsr s' (P.map φ)

theorem **_root_.GlobalTheory.UTr.aeval_root_eq_zero_iff** {q : B[X]} : aeval (_root_.GlobalTheory.UTr.root P) q = 0 ↔ P.1 ∣ q

theorem **_root_.GlobalTheory.UTr.aeval_root** : aeval (_root_.GlobalTheory.UTr.root P) P.1 = 0

theorem **_root_.GlobalTheory.UTr.exists_aeval_root** (z : _root_.GlobalTheory.UTr.Alg P) : ∃ q : B[X], aeval (_root_.GlobalTheory.UTr.root P) q = z

theorem **Alg.induction** {p : Alg P → Prop} (hC : ∀ b, p (algebraMap B _ b)) (hr : p (root P)) (hadd : ∀ y z, p y → p z → p (y + z)) (hmul : ∀ y z, p y → p z → p (y ∗ z)) (z : Alg P) : p z

theorem **Alg.ringHom_ext** {S : Type} [Ring S] {f g : Alg P →+∗ S} (hC : ∀ b, f (algebraMap B _ b) = g (algebraMap B _ b)) (hr : f (root P) = g (root P)) : f = g

@[simp] theorem **val_gamma** : (gamma P : Alg P) = root P

theorem **_root_.GlobalTheory.UTr.eval₂_gamma_inv** : P.1.eval₂ ((algebraMap B (_root_.GlobalTheory.UTr.Alg P)).comp s) ((_root_.GlobalTheory.UTr.gamma P)⁻¹ : (_root_.GlobalTheory.UTr.Alg P)ˣ) = 0

@[simp] theorem **_root_.GlobalTheory.UTr.conj_algebraMap** (b : B) : _root_.GlobalTheory.UTr.conj P (algebraMap B _ b) = algebraMap B _ (s b)

@[simp] theorem **_root_.GlobalTheory.UTr.conj_root** : _root_.GlobalTheory.UTr.conj P (_root_.GlobalTheory.UTr.root P) = ((_root_.GlobalTheory.UTr.gamma P)⁻¹ : (_root_.GlobalTheory.UTr.Alg P)ˣ)

theorem **root_mul_conj_root** : root P ∗ conj P (root P) = 1

theorem **conj_smul** (b : B) (z : Alg P) : conj P (b • z) = s b • conj P z

theorem **_root_.GlobalTheory.UTr.conj_conj** (hs : ∀ b, s (s b) = b) (z : _root_.GlobalTheory.UTr.Alg P) : _root_.GlobalTheory.UTr.conj P (_root_.GlobalTheory.UTr.conj P z) = z

@[simp] theorem **val_normU** (z : (Alg P)ˣ) : (normU P z : Alg P) = z ∗ conj P z

theorem **mem_tor** {z : (Alg P)ˣ} : z ∈ tor P ↔ (z : Alg P) ∗ conj P z = 1

theorem **gamma_mem_tor** : gamma P ∈ tor P

theorem **_root_.GlobalTheory.UTr.mem_plus** {x : (_root_.GlobalTheory.UTr.Alg P)ˣ} : x ∈ _root_.GlobalTheory.UTr.plus P ↔ _root_.GlobalTheory.UTr.conj P x = x

theorem **_root_.GlobalTheory.UTr.normU_mem_plus** (hs : ∀ b, s (s b) = b) (z : (_root_.GlobalTheory.UTr.Alg P)ˣ) : _root_.GlobalTheory.UTr.normU P z ∈ _root_.GlobalTheory.UTr.plus P

@[simp] theorem **algMap_algebraMap** (b : B) : algMap φ h (algebraMap B _ b) = algebraMap B' _ (φ b)

@[simp] theorem **algMap_root** : algMap φ h (root P) = root Q

theorem **algMap_conj** (z : Alg P) : algMap φ h (conj P z) = conj Q (algMap φ h z)

@[simp] theorem **coe_torMap** (z : tor P) : ((torMap φ h hφ z).1 : Alg Q) = algMap φ h (z.1 : Alg P)

theorem **herm_mul_mul** (x u y z : Alg P) : herm P x (u * y) (u * z) = herm P (x * (u * conj P u)) y z

@[simp] theorem **_root_.GlobalTheory.UTr.evalMat_root** (g : Matrix n n B) (hg : aeval g P.1 = 0) : _root_.GlobalTheory.UTr.evalMat P g hg (_root_.GlobalTheory.UTr.root P) = g

theorem **eq_evalMat** {g : Matrix n n B} (hg : aeval g P.1 = 0) (f : Alg P →ₐ[B] Matrix n n B) (hf : f (root P) = g) (z : Alg P) : f z = evalMat P g hg z

theorem **_root_.GlobalTheory.UTr.cT_evalMat** (g : _root_.Formalism.UG.UJ s J) (hg : aeval g.1.1 P.1 = 0) (z : _root_.GlobalTheory.UTr.Alg P) : _root_.Formalism.UG.cT s (_root_.GlobalTheory.UTr.evalMat P _ hg z) * J = J * _root_.GlobalTheory.UTr.evalMat P _ hg (_root_.GlobalTheory.UTr.conj P z)

@[simp] theorem **coe_torToU** (g : UJ s J) (hg : aeval g.1.1 P.1 = 0) (z : tor P) : (torToU P g hg z).1.1 = evalMat P _ hg (z.1 : Alg P)

theorem **torToU_gamma** (g : UJ s J) (hg : aeval g.1.1 P.1 = 0) : torToU P g hg (gamma P, gamma_mem_tor P) = g

theorem **hform_mulVec** (J g : Matrix n n B) (u v : n → B) : hform s J (g *ᵥ u) (g *ᵥ v) = hform s (cT s g * J * g) u v

theorem **hform_single** (J : Matrix n n B) (i j : n) : hform s J (Pi.single i 1) (Pi.single j 1) = J i j

theorem **mulMat_mulVec** (i : Alg P ≃ₗ[B] (n → B)) (z y : Alg P) : mulMat i z *ᵥ i y = i (z * y)

theorem **eq_mulMat** (i : Alg P ≃ₗ[B] (n → B)) {z : Alg P} {g : Matrix n n B} (h : ∀ y, g *ᵥ i y = i (z * y)) : g = mulMat i z

theorem **aeval_mulMat_root** (i : Alg P ≃ₗ[B] (n → B)) : aeval (mulMat i (root P)) P.1 = 0

theorem **IsCyc.aeval_eq_zero** {g : Matrix n n B} (h : IsCyc P g) : aeval g P.1 = 0

theorem **evalMat_eq_mulMat** {g : Matrix n n B} (hg : aeval g P.1 = 0) {i : Alg P ≃ₗ[B] (n → B)} (hi : mulMat i (root P) = g) (z : Alg P) : evalMat P g hg z = mulMat i z

theorem **_root_.GlobalTheory.UTr.aeval_map_map** (φ : B →+* B') (g : Matrix n n B) (q : B[X]) : aeval (g.map φ) (q.map φ) = (aeval g q).map φ

theorem **evalMat_algMap** {Q : Csr s'} (φ : B →+* B') (h : Q.1 ∣ P.1.map φ) {g : Matrix n n B} (hg : aeval g P.1 = 0) (hg' : aeval (g.map φ) Q.1 = 0) (z : Alg P) : evalMat Q _ hg' (algMap φ h z) = (evalMat P g hg z).map φ

theorem **IsCyc.map** {g : Matrix n n B} (h : IsCyc P g) (φ : B →+* B') {Q : Csr s'} (hQ : Q.1 = P.1.map φ) : IsCyc Q (g.map φ)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ApplyLayers

theorem **root_pow** {p : ℕ} (hp : p.Prime) {ζ : LL} (h : ∃ j : ℕ, IsPrimitiveRoot ζ (p ^ j)) (a : ℕ) : ∃ j : ℕ, IsPrimitiveRoot (ζ ^ a) (p ^ j)

theorem **levOf_root** (W : LocWach PD LL O v n) (i : ZMod n) {ζ : LL} (h : ∃ j : ℕ, IsPrimitiveRoot ζ (PD.p ^ j)) : levOf W i ζ = W.lev i ζ

theorem **levOf_cases** (W : LocWach PD LL O v n) (i : ZMod n) (ζ : LL) : ∃ ζ', levOf W i ζ = W.lev i ζ'

theorem **nOf_crys** (W : LocWach PD LL O v n) {T : LocLat G Qp LL O} (h : IsCrys PD T.ρ) : nOf W T = W.N T

theorem **nOf_ind** (W : LocWach PD LL O v n) (T : LocLat G Qp LL O) (p : Submodule (LetA O n) (W.DD T) → Prop) (h1 : p (W.N T)) (h0 : p ⊥) : p (nOf W T)

theorem **p_prime** (P : LocPrime l) (K : FinExt l) : (P.PD K).p.Prime

theorem **ioW** : IoW (Γ

theorem **latL_N** {T : LocLat ɪ(GK K) ℚ_[l] ɪLL.1 ɪ(OL LL)} (hT : IsCrys (P.PD K) T.ρ) : (LatL H F h hL).N T = (H.W K s LL h).N T

theorem **apply_layers** (p : ∀ (Γ : GamS P K s LL) (R : WSer Γ) (I : IoS AI R) (E : WEt I), NS E → Prop) (hp : ∀ (Γ : GamS P K s LL) (R : WSer Γ) (I : IoS AI R), IoW I → ∀ (E : WEt I) (Mo : NS E), p Γ R I E Mo) : p (GamL H h) (SerL H F h hL) (IoL H F h hL) (EtL H F h hL) (LatL H F h hL)

theorem **lev_eq_evL** (W : LocWach PD L O v n) (i : ZMod n) (ζ : L) (a : LetA O n) : W.lev i ζ a = evL W i ζ (a i)

theorem **evL_X** (W : LocWach PD L O v n) (i : ZMod n) {ζ : L} (h : ∃ j : ℕ, IsPrimitiveRoot ζ (PD.p ^ j)) : evL W i ζ X = C ζ * exp L − 1

theorem **levLaws_of** (W : LocWach PD L O v n) (hp : PD.p = l) (c : W.Gam → ℤ_[l]) (hc : ∀ g, algebraMap ℤ_[l] O (c g) = (W.χΓ g : O)) (hγ : ∀ g, W.γ₀ g = GlobalTheory.Fon.gam O (c g)) (hv : ∀ (c : ℤ_[l]) (j a : ℕ), (j : WithTop ℚ) ≤ v (algebraMap ℤ_[l] L c − (a : L)) → (l : ℤ_[l]) ^ j ∣ c − a) (H : EvLaw l (O

theorem **levLaws** : LevLaws (H.W K s LL h)

theorem **laws** : Laws H F h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.BddGoodOfBddAll

theorem **inV_iff_val** (T : CDVR) (x : T.Fr) : InV T x ↔ valV T.V x ≤ 1

theorem **inV_iff_vV** (T : CDVR) {x : T.Fr} (hx : x ≠ 0) : InV T x ↔ 0 ≤ vV T.V x

theorem **inV_iff_nval** (T : CDVR) {e : ℕ} (he : 0 < e) (x : T.Fr) : InV T x ↔ 0 ≤ nval T.V e x

theorem **bddGood_of_bddAll** {κ : c.F} {f : Rul Xo c} (h : BddAll Xo c la κ f) : BddGood Xo c la κ f

theorem **bddAll_iff_smul** (κ : c.F) (f : Rul Xo c) : BddAll Xo c la κ f ↔ BddAll Xo c la 1 (κ • f)

theorem **bddGood_iff_smul** (κ : c.F) (f : Rul Xo c) : BddGood Xo c la κ f ↔ BddGood Xo c la 1 (κ • f)

theorem **evI_one** {T : CDVR} {io : c.F →+* T.Fr} (y : IntFr Xo c T io) : evI y 1 = 1

private theorem **evI_add_non_galois_descent** {T : CDVR} {io : c.F →+* T.Fr} (y : IntFr Xo c T io) (f g : Rul Xo c) : evI y (f + g) = evI y f + evI y g

theorem **bddAll_add** {κ : c.F} {f g : Rul Xo c} (hf : BddAll Xo c la κ f) (hg : BddAll Xo c la κ g) : BddAll Xo c la κ (f + g)

private theorem **evI_mul_non_galois_descent** {T : CDVR} {io : c.F →+* T.Fr} (y : IntFr Xo c T io) (f g : Rul Xo c) : evI y (f * g) = evI y f * evI y g

theorem **bddAll_mul** {κ μ : c.F} {f g : Rul Xo c} (hf : BddAll Xo c la κ f) (hg : BddAll Xo c la μ g) : BddAll Xo c la (κ * μ) (f * g)

theorem **inV_io** {T : CDVR} {io : c.F →+* T.Fr} {e : ℕ} (ho : OverLa la T.V io e) {a : c.F} (ha : a ≠ 0) : InV T (io a) ↔ 0 ≤ vLa la a

theorem **smul_mem_supSet** {W : Submodule c.F (Rul Xo c)} {a : c.F} (ha : 0 ≤ vLa la a) {f : Rul Xo c} (hf : f ∈ supSet Xo c la W) : a • f ∈ supSet Xo c la W

theorem **add_mem_supSet** {W : Submodule c.F (Rul Xo c)} {f g : Rul Xo c} (hf : f ∈ supSet Xo c la W) (hg : g ∈ supSet Xo c la W) : f + g ∈ supSet Xo c la W

theorem **bddAll_one_iff** (hne : IntFrNe Xo c (fun _ _ y => y.IsAlg)) {κ : c.F} (hκ : κ ≠ 0) : BddAll Xo c la κ 1 ↔ 0 ≤ vLa la κ

private theorem **vV_mul'** (T : CDVR) {a b : T.Fr} (ha : a ≠ 0) (hb : b ≠ 0) : vV T.V (a * b) = vV T.V a + vV T.V b

private theorem **inV_mul_iff_vLa_add_nval_nonneg** {T : CDVR} {io : c.F →+* T.Fr} {e : ℕ} (ho : OverLa la T.V io e) {κ : c.F} (hκ : κ ≠ 0) (z : T.Fr) : InV T (io κ * z) ↔ 0 ≤ (((vLa la κ : ℤ) : ℝ) : EReal) + nval T.V e z

theorem **bddAll_iff_vAtV** {κ : c.F} (hκ : κ ≠ 0) (f : Rul Xo c) : BddAll Xo c la κ f ↔ ∀ y : PtV Xo c (fun _ _ y => y.IsAlg) Unit (fun _ _ _ _ => True) la, 0 ≤ (((vLa la κ : ℤ) : ℝ) : EReal) + vAtV y f

theorem **bddGood_iff_vAtV** {κ : c.F} (hκ : κ ≠ 0) (f : Rul Xo c) : BddGood Xo c la κ f ↔ ∀ y : PtV Xo c (fun _ _ y => y.IsAlg ∧ Good y) Unit (fun _ _ _ _ => True) la, 0 ≤ (((vLa la κ : ℤ) : ℝ) : EReal) + vAtV y f

theorem **good_congr** {T : CDVR} {io : c.F →+* T.Fr} {y y' : IntFr Xo c T io} (h : (letI

theorem **good_iff_abelian** {T : CDVR} {io : c.F →+* T.Fr} (y : IntFr Xo c T io) : Good y ↔ ∃ (G : SAB H (Sp T.V)) (t' : SAB H (Sp T.Fr)) (_ : SABMap t' Xo.univ.t (letI

theorem **bddGood_one_iff** (hne : IntFrNe Xo c (fun _ _ y => y.IsAlg ∧ Good y)) {κ : c.F} (hκ : κ ≠ 0) : BddGood Xo c la κ 1 ↔ 0 ≤ vLa la κ

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.BoundAtTwoAndThreeForFamily

theorem **BoundAtTwoAndThreeForFamily.of_all** {eps : ℝ} {Fm : FamP eps} (h : BoundAtTwoThreeFamily Fm) : BoundAtTwoAndThreeForFamily Fm

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.BoundAtTwoThreeFamily

theorem **midF_mono** (L : SizeData₂ D) {C C' : ℝ} (h : C ≤ C') (h0 : 0 ≤ C') : midF L C ≤ midF L C'

theorem **starF_of** (L : SizeData₂ D) (ρ : FamilyRepresentationConstants.RhoFam D) (C : ℝ) (h23 : BoundAtTwoThree L ρ C) (hS : StarS L ρ C) : StarF L ρ C

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CharpolyEqOfIsRowPar

theorem **mem_plB** (Pl : PlSide D) (w : D.what₂.places.Pl) (z : Zbar (D.what₂.places.resChar w)) : Pl.ι w (z : Qbar (D.what₂.places.resChar w)) ∈ Pl.plB w

theorem **plRed_eq** (Pl : PlSide D) (w : D.what₂.places.Pl) (z : Zbar (D.what₂.places.resChar w)) : plRed Pl w z = Formalism.L2.NT.redThrough (Pl.plB w) (Pl.plκ w) (Pl.ι w z)

theorem **teichAt_of_iotaOk** {Pl : PlSide D} (h : IotaOk D Pl) (w : D.what₂.places.Pl) : TeichAt (Pl.ι w) w (plRed Pl w)

theorem **resTied_of_plSide** {Rw : RowSide D} {Pl : PlSide D} (h : IotaOk D Pl) (hr : ∀ w, D.what₂.muAt w ≠ D.what₂.mu0 → ∀ σ π r, D.nd.goodRow σ → OnW D σ → IsRowPar Rw σ π → IsGaloisRepOf D π _ (Pl.ι w) r → IsSSRep (D.ntypes.rbar σ w) ∧ ∀ g, ∃ p : (Zbar (D.what₂.places.resChar w))[X], p.map (Zbar (D.what₂.places.resChar w)).subtype = ((r g : GL (Fin D.R.r) (Qbar (D.what₂.places.resChar w))) : Matrix (Fin D.R.r) (Fin D.R.r) (Qbar (D.what₂.places.resChar w))).charpoly ∧ p.map (plRed Pl w) = ((D.ntypes.rbar σ w g : GL (Fin D.R.r) (FbAt D w)) : Matrix (Fin D.R.r) (Fin D.R.r) (FbAt D w)).charpoly) : ResTied D Rw

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CharpolySmulOfProd

theorem **Reading.toC_apply** (ρ : Reading X i b) (x : X.O0 i) : ρ.toC x = ρ.ι (toQ ρ.κ x)

theorem **Reading.toC_injective** (ρ : Reading X i b) : Function.Injective ρ.toC

theorem **Reading.nonempty** (i : X.Tw) (b : Bool) : Nonempty (Reading X i b)

theorem **Reading.Induces.resChar** {ρ : Reading X i b} {w : D.what₂.places.Pl} (h : ρ.Induces w) : D.what₂.places.resChar w = X.l i

theorem **muBase_eq** (l : ℕ) (j : Fin D.R.m) : muBase C l j = (hframe C).mu0 j * GlobalTheory.GalChar.delAt D.host (hframe C) l

theorem **eExp_int** (j : Fin D.R.m) (τ : D.R.Psi) : ∃ z : ℤ, eExp D j τ = z

theorem **shiftChar_unr** (v : PlE D.host) : (shiftChar C).UnrAt v

theorem **shiftChar_val** (v : PlE D.host) : (((shiftChar C).val v : ℂˣ) : ℂ) = ((C.eta.val v : ℂˣ) : ℂ)⁻¹ * ((normV v : ℕ) : ℂ) ^ (((D.R.t : ℝ) / 2 : ℝ) : ℂ)

theorem **blockChar_val** (l : ℕ) (j : Fin D.R.m) (v : PlE D.host) : (((blockChar C l j).val v : ℂˣ) : ℂ) = (((muBase C l j).val v : ℂˣ) : ℂ) * ((C.eta.val v : ℂˣ) : ℂ)⁻¹ * ((normV v : ℕ) : ℂ) ^ (((D.R.r : ℝ) / 2 : ℝ) : ℂ)

theorem **exists_shift** (ha : IsGlobalArtin art) (hC : ValLL0 D C) (ρ : Reading X i b) : ∃ (ν : GalE D →* (X.O0 i)ˣ) (n : _), IsGal art ρ.κ ρ.ι (shiftChar C) n ν

theorem **exists_blockChar** (ha : IsGlobalArtin art) (hC : ValLL0 D C) (ρ : Reading X i b) (j : Fin D.R.m) : ∃ (χ : GalE D →* (X.O0 i)ˣ) (n : _), IsGal art ρ.κ ρ.ι (blockChar C (X.l i) j) n χ

theorem **card_rowPar** {R : Ranks} (P : HCParam R) : Multiset.card (rowPar P) = R.r

theorem **param_eq** {R : Ranks} (P : HCParam R) (j : Fin R.m) : GlobalTheory.IndefiniteRowSideFullHecke.JumpDictionary.param P j = rowPar P + GlobalTheory.IndefiniteRowSideFullHecke.JumpDictionary.run P j

theorem **two_mul_nV_half** (R : Ranks) : 2 * (R.nV / 2) = R.r + R.t

theorem **two_mul_half** (R : Ranks) {k : ℕ} (hk : k ≤ R.r) : 2 * (k * (R.r - k) / 2) = k * (R.r - k)

theorem **normV_ne_zero** (P : PlE D.host) : ((normV P : ℕ) : ℂ) ≠ 0

theorem **heckeVal_zero** : heckeVal μ π 0 P = 1

theorem **card_satake** (h : D.naut.unr (D.naut.locG π P)) : Multiset.card (satake μ π P) = D.R.r

theorem **heckeVal_top** (h : D.naut.unr (D.naut.locG π P)) : heckeVal μ π D.R.r P = (satake μ π P).prod

theorem **pow_mul_heckeVal** {k : ℕ} (hk : k ≤ D.R.r) : ((normV P : ℕ) : ℂ) ^ (k * (k - 1) / 2) * heckeVal μ π k P = ((satake μ π P).map fun α => ((normV P : ℕ) : ℂ) ^ ((D.R.r - 1) / 2) * α).esymm k

theorem **geometric_eigenvalue** (C : CharFrame D) (l : ℕ) (j : Fin D.R.m) (a : ℂ) : (a * ((normV P : ℕ) : ℂ) ^ (((1 - (D.R.r : ℝ)) / 2 : ℝ) : ℂ) * (((shiftChar C).val P : ℂˣ) : ℂ)⁻¹ * (((blockChar C l j).val P : ℂˣ) : ℂ) * ((normV P : ℕ) : ℂ) ^ D.R.t)⁻¹ = ((normV P : ℕ) :

ℂ) ^ ((D.R.r − 1) / 2) ∗ (a⁻¹ ∗ (((muBase C l j).val P : ℂˣ) : ℂ)⁻¹ ∗ (((normV P : ℕ) : ℂ) ^ (D.R.nV / 2))⁻¹)

theorem **charpoly_smul_of_prod** {n : ℕ} (M : Matrix (Fin n) (Fin n) ℂ) (m : Multiset ℂ) (c : ℂ) (h : M.charpoly = (m.map fun a => X − C a).prod) : (c • M).charpoly = (m.map fun a => X − C (a ∗ c)).prod

theorem **isGaloisRepOf_conj** {a : ℕ} {π : D.naut.CG a} {l : ℕ} [Fact l.Prime] {ι : Qbar l ≃+* ℂ} {r r' : GalE D →* GL (Fin a) (Qbar l)} (h : IsGaloisRepOf D π l ι r) (A : GL (Fin a) (Qbar l)) (hA : ∀ g, r' g = A ∗ r g ∗ A⁻¹) : IsGaloisRepOf D π l ι r'

theorem **isTripleAt_of_conj** (good : D.nd.goodRow σ) (occurs : D.what₂.isot (X.jS i s) σ ≠ ⊥) (param : IsParam σ π) (gal : IsGaloisRepOf D π (X.l i) ρ.ι r) (shift : ∃ n, IsGal art ρ.κ ρ.ι (shiftChar C) n ν) (A : GL (Fin D.R.r) (Qbar (X.l i))) (rho : ∀ g, ((X.rhoS i s g : GL (Fin D.R.r) (X.O0 i)) : Matrix (Fin D.R.r) (Fin D.R.r) (X.O0 i)).map (toQ ρ.κ) = toQ ρ.κ ((ν g : (X.O0 i)ˣ) : X.O0 i) • ((A ∗ r g ∗ A⁻¹ : GL (Fin D.R.r) (Qbar (X.l i))) : Matrix (Fin D.R.r) (Fin D.R.r) (Qbar (X.l i)))) (hecke : ∀ k P, k ≤ D.R.r → IsSplitPl D.host P → P ∉ X.S0 i → ∃ x : X.O0 i, algebraMap (X.O0 i) (X.O i) x = X.hEig i s k P ∧ ρ.toC x = heckeVal (muBase C (X.l i) (jIdx s)) π k P) (induces : ρ.Induces w) (residual : IsResidual ρ.ι r σ w red) : IsTripleAt C art ρ s

theorem **jIdx_add_one** (s : X.Sig i) : (jIdx s).1 + 1 = X.jS i s

theorem **IsChars.chiS** {ρ : Reading X i (X.side i s)} (h : IsChars C art ρ) : ∃ n, IsGal art ρ.κ ρ.ι (blockChar C (X.l i) (jIdx s)) n (Formalism.MainNotions.BlochKatoReduction.Gen.chiS i s)

theorem **IsTripleWith.hEig_zero** (h : IsTripleWith C art ρ s σ π r ν w red) {P : PlE D.host} (hP : IsSplitPl D.host P) (hS : P ∉ X.S0 i) : X.hEig i s 0 P = 1

theorem **IsTripleWith.unr_of_notMem** (h : IsTripleWith C art ρ s σ π r ν w red) {P : PlE D.host} (hS : P ∉ X.S0 i) : D.naut.unr (D.naut.locG π P)

theorem **IsTripleWith.charpoly_frob** (ha : IsGlobalArtin art) (h : IsTripleWith C art ρ s σ π r ν w red) {P : PlE D.host} (hl : P ∉ placesOver D.host.E (X.l i)) (hg : ∀ k, P ∉ placesOver D.host.E (D.host.g k)) {B : ValuationSubring D.nd.Om} (hB : IsPlaceOver B P) {g : GalE D} (hF : IsFrob (F

theorem **IsTripleFor.triple** (h : IsTripleFor C art Y) {i : X.Tw} (hi : X.NonzeroIndexPlace i) (s : X.Sig i) : ∃ ρ : Reading X i (X.side i s), IsChars C art ρ ∧ IsTripleAt C art ρ s

theorem **IsTripleOf.allGood** (h : IsTripleOf Y) : GlobalTheory.TwSys.AllGood Y

theorem **IsTripleOf.sig_all** (h : IsTripleOf Y) : Y.sig_all

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ConjTupRefl

theorem **ser_vP** (i : Fin D.R.m) (t : T0) (u : ∀ ρ, lowT (R.MW i (sg0 D A)) ρ) (h : GA D (Vc D i)) (w : (R.cV i).W) (g : GA D (Wc D i)) : ser R i t (R.vP i t) u h w g = thetaKO (R.ωP i t) h g (R.ptP i t (R.ioP i (PiTensorProduct.tprod ℂ u) w))

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ConnPhiExtensionGeneratorEqZeta

theorem **ulim_sub_dvd** {s : ℕ → K⟦X⟧} (h : ∀ n, (X : K⟦X⟧) ^ (n + 1) ∣ s (n + 1) − s n) (n : ℕ) : (X : K⟦X⟧) ^ (n + 1) ∣ ulim s − s n

theorem **mem_uPow** {m : ℕ} {v : Vec K a ι} : v ∈ uPow K a ι m ↔ ∀ τ i, (X : K⟦X⟧) ^ m ∣ v τ i

theorem **uPow_anti** {m n : ℕ} (h : m ≤ n) : uPow K a ι n ≤ uPow K a ι m

theorem **eq_zero_of_mem_uPow** {v : Vec K a ι} (h : ∀ m, v ∈ uPow K a ι (m + 1)) : v = 0

theorem **mem_uPow_one** {v : Vec K a ι} : v ∈ uPow K a ι 1 ↔ red v = 0

theorem **mem_uPow_iff_exists** {m : ℕ} {v : Vec K a ι} : v ∈ uPow K a ι m ↔ ∃ w : Vec K a ι, v = (X : K⟦X⟧) ^ m • w

theorem **X_smul_mem_uPow** (w : Vec K a ι) : (X : K⟦X⟧) • w ∈ uPow K a ι 1

theorem **dvd_mulVec** [Fintype ι] {c : K⟦X⟧} (Y : Matrix κ ι K⟦X⟧) {v : ι → K⟦X⟧} (hv : ∀ i, c ∣ v i) (j : κ) : c ∣ (Y ∗ᵥ v) j

theorem **X_pow_dvd_frob** (l : ℕ) [Fact l.Prime] {m : ℕ} {f : K⟦X⟧} (h : (X : K⟦X⟧) ^ (m + 1) ∣ f) : (X : K⟦X⟧) ^ (m + 2) ∣ frob K l f

theorem **Raises.pow_mem** {T : Module.End K (Vec K a ι)} (hT : Raises T) {w : Vec K a ι} (hw : w ∈ uPow K a ι 1) (n : ℕ) : (T ^ n) w ∈ uPow K a ι (n + 1)

theorem **geom_sub_mem** {T : Module.End K (Vec K a ι)} (hT : Raises T) {w : Vec K a ι} (hw : w ∈ uPow K a ι 1) (n : ℕ) : geom T w − (∑ j ∈ Finset.range n, T ^ j) w ∈ uPow K a ι (n + 1)

theorem **geom_mem** {T : Module.End K (Vec K a ι)} (hT : Raises T) {w : Vec K a ι} (hw : w ∈ uPow K a ι 1) : geom T w ∈ uPow K a ι 1

theorem **geom_eq** {T : Module.End K (Vec K a ι)} (hT : Raises T) {w : Vec K a ι} (hw : w ∈ uPow K a ι 1) : geom T w = w + T (geom T w)

@[simp] theorem **cst_apply** (d : ZMod a → ι → K) (τ : ZMod a) (i : ι) : cst d τ i = C (d τ i)

@[simp] theorem **red_cst** (d : ZMod a → ι → K) : red (cst d) = d

@[simp] theorem **derV_apply** (v : Vec K a ι) (τ : ZMod a) (i : ι) : derV v τ i = PowerSeries.derivative K (v τ i)

theorem **eq_sum_bas** [Fintype ι] [DecidableEq ι] (v : Vec K a ι) : v = ∑ i, (fun τ => v τ i) • (bas i : Vec K a ι)

theorem **cst_single** [DecidableEq ι] (i : ι) : cst (fun _ : ZMod a => (Pi.single i 1 : ι → K)) = bas i

theorem **raises_phi** : Raises P.phi

theorem **raises_smul_phi** (c : K) : Raises (c • P.phi)

theorem **_root_.GlobalTheory.BK.Raises.eq_zero** {T : Module.End K (Vec K a ι)} (hT : Raises T) {α : Type} {σ : α → α} (hσ : Function.Surjective σ) {δ : α → Vec K a ι} (h1 : ∀ x, δ x ∈ uPow K a ι 1) (h : ∀ x, δ (σ x) = T (δ x)) (x : α) : δ x = 0

theorem **xiSeq_zero** (τ : ZMod a) : P.xiSeq 0 τ = 1

theorem **xiSeq_succ** (n : ℕ) (τ : ZMod a) : P.xiSeq (n + 1) τ = P.mx τ ∗ (P.xiSeq n (τ − 1)).map (frob K l) ∗ ((P.mxInv τ).map constantCoeff).map C

theorem **xiSeq_dvd** (n : ℕ) (τ : ZMod a) (i j : ι) : (X : K⟦X⟧) ^ (n + 1) ∣ P.xiSeq (n + 1) τ i j − P.xiSeq n τ i j

theorem **xiMat_sub_dvd** (n : ℕ) (τ : ZMod a) (i j : ι) : (X : K⟦X⟧) ^ (n + 1) ∣ P.xiMat τ i j − P.xiSeq n τ i j

theorem **xiMat_map_constantCoeff** (τ : ZMod a) : (P.xiMat τ).map constantCoeff = 1

theorem **mx_mul_xiMat** (τ : ZMod a) : P.mx τ ∗ (P.xiMat (τ − 1)).map (frob K l) = P.xiMat τ ∗ (P.mx0 τ).map C

theorem **isUnit_det_xiMat** (τ : ZMod a) : IsUnit (P.xiMat τ).det

theorem **xiExt_apply** (w : Vec K a ι) (τ : ZMod a) : P.xiExt w τ = P.xiMat τ ∗ᵥ w τ

theorem **xiExt_smul_tot** (f : ZMod a → K⟦X⟧) (w : Vec K a ι) : P.xiExt (f • w) = f • P.xiExt w

theorem **xi_apply** (d : P.D) : P.xi d = P.xiExt (cst d)

theorem **red_xiExt** (w : Vec K a ι) : red (P.xiExt w) = red w

@[simp] theorem **red_xi** (d : P.D) : red (P.xi d) = d

theorem **phi_xiExt** (w : Vec K a ι) : P.phi (P.xiExt w) = P.xiExt fun τ => (P.mx0 τ).map C *ᵥ frobV l (w (τ - 1))

theorem **phi_xi** (d : P.D) : P.phi (P.xi d) = P.xi (P.phiD d)

theorem **map_xi** {Q : PhiMod O K l e a κ} {g : Vec K a ι → Vec K a κ} (hg : ∀ v, g (P.phi v) = Q.phi (g v)) (d : P.D) : g (P.xi d) = Q.xi (red (g (P.xi d)))

theorem **xi_unique** {s : P.D → Vec K a ι} (h0 : ∀ d, red (s d) = d) (h : ∀ d, P.phi (s d) = s (P.phiD d)) (d : P.D) : s d = P.xi d

theorem **conn_apply** (v : Vec K a ι) : P.conn v = (X : K⟦X⟧) • P.nabla v

theorem **nabla_xiExt** (w : Vec K a ι) : P.nabla (P.xiExt w) = P.xiExt (derV w)

theorem **conn_xiExt** (w : Vec K a ι) : P.conn (P.xiExt w) = P.xiExt fun τ i => N K (w τ i)

@[simp] theorem **nabla_xi** (d : P.D) : P.nabla (P.xi d) = 0

@[simp] theorem **conn_xi** (d : P.D) : P.conn (P.xi d) = 0

theorem **nabla_smul_tot** (f : ZMod a → K⟦X⟧) (v : Vec K a ι) : P.nabla (f • v) = derTot f • v + f • P.nabla v

theorem **conn_smul_tot** (f : ZMod a → K⟦X⟧) (v : Vec K a ι) : P.conn (f • v) = NTot f • v + f • P.conn v

theorem **conn_smul** (f : K⟦X⟧) (v : Vec K a ι) : P.conn (f • v) = N K f • v + f • P.conn v

theorem **conn_mem_uPow** (v : Vec K a ι) : P.conn v ∈ uPow K a ι 1

theorem **conn_phi** (v : Vec K a ι) : P.conn (P.phi v) = (l : K) • P.phi (P.conn v)

theorem **map_conn** {Q : PhiMod O K l e a κ} (g : Vec K a ι →+ Vec K a κ) (σ : (ZMod a → K⟦X⟧) → ZMod a → K⟦X⟧) (hσ : ∀ f, σ (NTot f) = NTot (σ f)) (hs : ∀ f v, g (f • v) = σ f • g v) (hg : ∀ v, g (P.phi v) = Q.phi (g v)) (v : Vec K a ι) : g (P.conn v) = Q.conn (g v)

theorem **bas_mem_M** (i : ι) : bas i ∈ P.M

theorem **phi_bas** (i : ι) (τ : ZMod a) : P.phi (bas i) τ = (P.mx τ).col i

theorem **eq_sum_phi_bas** (v : Vec K a ι) : v = ∑ i, (fun τ => (P.mxInv τ *ᵥ v τ) i) • P.phi (bas i)

theorem **smul_tot_mem_Mext** {B : Subalgebra O K⟦X⟧} {f : ZMod a → K⟦X⟧} (hf : ∀ τ, f τ ∈ B) {v : Vec K a ι} (hv : v ∈ P.Mext B) : f • v ∈ P.Mext B

theorem **smul_tot_mem_ext** {B : Subalgebra O K⟦X⟧} {f : ZMod a → K⟦X⟧} (hf : ∀ τ, f τ ∈ B) {v : Vec K a ι} (hv : v ∈ P.ext B) : f • v ∈ P.ext B

theorem **Mext_mono** {B B' : Subalgebra O K⟦X⟧} (h : B ≤ B') {v : Vec K a ι} (hv : v ∈ P.Mext B) : v ∈ P.Mext B'

theorem **ext_mono** {B B' : Subalgebra O K⟦X⟧} (h : B ≤ B') {v : Vec K a ι} (hv : v ∈ P.ext B) : v ∈ P.ext B'

theorem **lphi_apply** (v : Vec K a ι) : P.lphi v = (l : K) • P.phi v

theorem **raises_lphi** : Raises P.lphi

theorem **lphi_conn_mem** (y : Vec K a ι) : P.lphi (P.conn y) ∈ uPow K a ι 1

theorem **zeta_mem_uPow** (y : Vec K a ι) : P.zeta y ∈ uPow K a ι 1

theorem **zeta_eq** (y : Vec K a ι) : P.zeta y = P.lphi (P.zeta y - P.conn y)

theorem **zeta_unique** {y z : Vec K a ι} (h1 : z ∈ uPow K a ι 1) (h : z = P.lphi (z - P.conn y)) : z = P.zeta y

theorem **zeta_add_sum_mem** (y : Vec K a ι) (m : ℕ) : P.zeta y + ∑ n ∈ Finset.range m, (P.lphi ^ (n + 1)) (P.conn y) ∈ uPow K a ι (m + 1)

theorem **zeta_phi_sub** (m : Vec K a ι) : P.zeta (P.phi m - m) = P.conn (P.phi m)

theorem **map_zeta** {Q : PhiMod O K l e a κ} (g : Vec K a ι →ₗ[K] Vec K a κ) (h1 : ∀ v ∈ uPow K a ι 1, g v ∈ uPow K a κ 1) (hg : ∀ v, g (P.phi v) = Q.phi (g v)) (hN : ∀ v, g (P.conn v) = Q.conn (g v)) (y : Vec K a ι) : g (P.zeta y) = Q.zeta (g y)

theorem **app_smul_tot** (f : ZMod a → K⟦X⟧) (v : Vec K a ι) : F.app (f • v) = f • F.app v

theorem **app_mem_uPow** {m : ℕ} {v : Vec K a ι} (hv : v ∈ uPow K a ι m) : F.app v ∈ uPow K a κ m

theorem **app_xi** (d : P.D) : F.app (P.xi d) = Q.xi (F.onD d)

theorem **app_conn** (v : Vec K a ι) : F.app (P.conn v) = Q.conn (F.app v)

theorem **app_nabla** (v : Vec K a ι) : F.app (P.nabla v) = Q.nabla (F.app v)

theorem **app_zeta** (y : Vec K a ι) : F.app (P.zeta y) = Q.zeta (F.app y)

theorem **plus_phi_join** (v : Vec K a ι) (f : ZMod a → K⟦X⟧) : (P.plus hy).phi (join v f) = join (P.phi v + frobTot K a l f • y) (frobTot K a l f)

theorem **plusInl_app** (v : Vec K a ι) : (P.plusInl hy).app v = join v 0

theorem **phi_extension_generator_eq_add** : (P.plus hy).phi xVec = xVec + (P.plusInl hy).app y

theorem **conn_phi_extension_generator_eq_zeta** : (P.plus hy).conn ((P.plus hy).phi xVec) = (P.plusInl hy).app (P.zeta y)

theorem **plus_conn_xVec_eq** : (P.plus hy).conn xVec = (P.plusInl hy).app (P.zeta y - P.conn y)

theorem **plusProj_app** (v : Vec K a ι) (f : ZMod a → K⟦X⟧) : (P.plusProj hy).app (join v f) = fun τ _ => f τ

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CoreEq

theorem **TwJoin.Sb_eq** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {nch : NChoice D} {xd : ThData D} {first : Prop} {G : Type} [iG : Group G] [fG : Fintype G] {Pl0 I0 T : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m} {C : Formalism.L2.Pj.CoreJ D} [aC : Algebra C.Q.c.F D.what₂.LL] {R : RealDat3 D A3 C.Q.c.F} {F : Face3 D A3} {cG : G0 R.Pg D.R.r ≃* G0g D.nd} {geo : GlobalTheory.RealGeo.GeoHalf C R.toRealDat fun _ => 1} {S : RatDat D.ntw} {YS : NExpRealS R cG S} {cf : CharFrame D} (a : TwJoin hL D nch xd first R F cG geo S YS cf) : a.X.R.Sb = R.Sb

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CrisToODToOCr

theorem **isCrisP_crisP** : GlobalTheory.MemLoc.IsCrisP (crisP l)

theorem **eq_crisP** {j : Per.Acris l →+* Per.BdRp l} (h : GlobalTheory.MemLoc.IsCrisP j) : j = crisP l

theorem **crisP_ainf** (x : Ainf l) : crisP l (ainfToAcris l x) = ainfToBdRp l x

theorem **complete_BdRp** {d : Ainf l} (hd : Ideal.span {d} = RingHom.ker (Per.θ l)) : IsAdicComplete (Ideal.span {ainfToBdRp l d}) (Per.BdRp l)

theorem **crisToOD_toOCr** (x : OA l O) : crisToOD l O (toOCr l O x) = toOD l O x

theorem **isUnit_l_OD** : IsUnit (l : OD l O)

theorem **iAl_of_mem** {f : K⟦X⟧} (h : f ∈ A O K) (h' : f ∈ invL l (A O K)) : ϖ.iAl O K ⟨f, h'⟩ = toOD l O (ϖ.iA O K ⟨f, h⟩)

theorem **complete_OD** : IsAdicComplete (Ideal.span {ϖ.iAl O K (EIn l e _)}) (OD l O)

theorem **iHatRing_ofAl** (f : invL l (A O K)) : ϖ.iHatRing O K (ofAl O K l e f) = ϖ.iAl O K f

theorem **iHat_ofAl** (f : invL l (A O K)) : ϖ.iHat O K (ofAl O K l e f) = ϖ.iAl O K f

theorem **iHat_ofA** (f : A O K) : ϖ.iHat O K (ofA O K l e f) = toOD l O (ϖ.iA O K f)

theorem **iSD_of_mem** {f : K⟦X⟧} (h : f ∈ S O K e) (h' : f ∈ invL l (S O K e)) : ϖ.iSD O K ⟨f, h'⟩ = crisToOD l O (ϖ.iS O K ⟨f, h⟩)

theorem **iD_ofA** (f : ZMod a → A O K) : ϖ.iD t K (fun τ => ofA O K l e (f τ)) = toOD l O (ϖ.iTot t K f)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CuspLawsOfLawsLit

theorem **dualCl_ofChar** (χ ψ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) (e : ψ = χ⁻¹) : DualCl (IrrGL.ofChar χ hχ) (IrrGL.ofChar ψ hψ)

theorem **dualCl_fst** {x y : IrrGL k} (H : DualCl x y) : y.1 = x.1

theorem **dualCl_unique** {x y y' : IrrGL k} (H : DualCl x y) (H' : DualCl x y') : y = y'

theorem **segDual_unique** {Δ Γ Γ' : Segment k} (H : SegDual Δ Γ) (H' : SegDual Δ Γ') : Γ = Γ'

theorem **segDual_ofChar** (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) (hχ' : IsSmoothChar χ⁻¹) (l : ℕ) (hl : 0 < l) : SegDual (Segment.ofChar χ hχ l hl) ((Segment.ofChar χ⁻¹ hχ' l hl).nu (1 - (l : ℂ)))

theorem **not_segDual_self** : ¬ SegDual (Segment.ofChar (1 : kˣ →* ℂˣ) isSmoothChar_one 2 two_pos) (.ofChar 1 isSmoothChar_one 2 two_pos)

theorem **isLExt_self** (H : LawsMin a ha e rc) : IsLExt rc rc

theorem **eq_of_isLExt** (H : IsLExt r rc) {x : IrrGL (w.adicCompletion E)} (hx : x.IsCusp) : rc x = r x

theorem **cuspLaws_of_lawsLit** (H : LawsLit a ha e rc) (hgap : ∀ ψ : AddChar (w.adicCompletion E) Circle, Continuous ψ → ψ ≠ 1 → ∀ x y : IrrGL (w.adicCompletion E), x.IsCusp → y.IsCusp → y.1 + 1 < x.1 → HasPairGamma ψ x.rep y.rep (e ψ ((rc x).lpChar (rc y)))) : CuspLaws a ha e rc where

theorem **one_of_cuspLaws** (H : CuspLaws a ha e r) (χ : (w.adicCompletion E)ˣ →* ℂˣ) (hχ : IsSmoothChar χ) : r (.ofChar χ hχ) = WD.lpOfChar h (χ.comp a) (ha.smooth hχ)

theorem **not_cuspLaws_const** (y : WD h) : ¬ CuspLaws a ha e fun _ => y

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CuspReadGIffCuspReadT

theorem **cuspReadG_iff_cuspReadT** (rc : RecMap A) : CuspReadG rc ↔ GlobalTheory.RecT.CuspReadT rc

theorem **cuspReadG_of_cgTrueT** (hT : UpperTheory.Zero.LLC.RecTExists) (D : D7) (T : WDTie D.nd D.naut) (C : GlobalTheory.HeightCounting.CGTrueT hT D T) : CuspReadG (GlobalTheory.RecT.recMapT T fun h => ParLawsT.locRecT hT h)

theorem **cuspReadG_of_family** (rc : GlobalTheory.RecT.RecFam) (D : D7) (T : WDTie D.nd D.naut) (C : GlobalTheory.RecT.CGTrueT rc D T) : CuspReadG (GlobalTheory.RecT.recMapT T rc)

theorem **good_rank_one** {w : PlE H} (σ : LocIrr w 1) : UpperTheory.Zero.LLC.Good (IrrGL.of σ.ρ σ.smooth σ.irred)

theorem **rank_one** {rc : RecMap A} (h : CuspReadG rc) (P : CuspSub H.E 1) (hc : P.IsCsd) : ∃ π : A.CG 1, A.csd π ∧ (∀ w σ, P.IsLocComp w σ → A.locG π w = rc w 1 σ) ∧ ∀ τ m, P.HasInfChar τ m → A.infG π τ = m

theorem **isCsd_triv** : (CuspSub.triv H.E (le_refl 1)).IsCsd

theorem **exists_member** {rc : RecMap A} (h : CuspReadG rc) : ∃ π : A.CG 1, A.csd π ∧ ∀ w, ∃ σ, (CuspSub.triv H.E (le_refl 1)).IsLocComp w σ ∧ A.locG π w = rc w 1 σ

theorem **not_cuspReadG** (rc : RecMap A) (h : ∀ π : A.CG 1, ¬ A.csd π) : ¬ CuspReadG rc

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CuspSubsetBaseForms

theorem **idCov_rd** (nab : NabV (T.Q.Xo j) T.Q.c) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (f : T.R j) : (idCov T j σ nab).rd a f = T.phiJ j σ a f

theorem **idCov_ratForms** (nab : NabV (T.Q.Xo j) T.Q.c) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) : (idCov T j σ nab).ratForms a n = baseForms T j σ a n

theorem **idCov_intForms** (nab : NabV (T.Q.Xo j) T.Q.c) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (S : Set (HeightOneSpectrum (𝒪 T.Q.c.F))) : (idCov T j σ nab).intForms a n S = baseInt T j σ a n S

theorem **idCov_isUnif** (h : LawsJ T) (hj : Formalism.L2.Pj.InR D j) (hσ : GoodEmb T.Q.c σ) (nab : NabV (T.Q.Xo j) T.Q.c) : (idCov T j σ nab).IsUnif

theorem **idCov_isPinned** (h : LawsJ T) (hj : Formalism.L2.Pj.InR D j) (hσ : GoodEmb T.Q.c σ) (nab : NabV (T.Q.Xo j) T.Q.c) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) : (idCov T j σ nab).IsPinned a n

theorem **rd_apply** (C : Cover T j σ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (f₂ : Rul C.d.Xo₂ C.d.c₂) (g : GA D (T.V j σ)) : C.rd a f₂ g = C.cc₂ (C.dict₂ g).1 * ((lieFac (T.Q.Dw j) (xiR D) a : ℂ) * T.mu j σ g * rdU C.U₂ f₂ (C.dict₂ g).1 (C.dict₂ g).2)

theorem **isUnif_withC** (C : Cover T j σ) (cc : C.U₂.Comp → ℂ) : (withC C cc).IsUnif ↔ C.IsUnif

theorem **pinned_withC** (C : Cover T j σ) {a : (D.host.E →+* ℂ) × ℕ → ℕ} {n : ℕ} (h : C.IsPinned a n) (cc : C.U₂.Comp → ℂ) (hc : CUnit cc) (hag : ∀ g, (∃ f ∈ sec (T.Q.Xo j) T.Q.c a n, T.phiJ j σ a f g ≠ 0) → cc (C.dict₂ g).1 = C.cc₂ (C.dict₂ g).1) : (withC C cc).IsPinned a n

theorem **pinned_unique** (C : Cover T j σ) {a : (D.host.E →+* ℂ) × ℕ → ℕ} {n : ℕ} (h : C.IsPinned a n) (cc : C.U₂.Comp → ℂ) (h' : (withC C cc).IsPinned a n) {f : T.R j} (hf : f ∈ sec (T.Q.Xo j) T.Q.c a n) {f₂ : Rul C.d.Xo₂ C.d.c₂} (hf₂ : f₂ ∈ sec C.d.Xo₂ C.d.c₂ a n) (hp : C.IsPull f f₂) {g : GA D (T.V j σ)} (hg : T.phiJ j σ a f g ≠ 0) : cc (C.dict₂ g).1 = C.cc₂ (C.dict₂ g).1

theorem **cusp_subset_baseForms** (hσ : GoodEmb T.Q.c σ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) : T.phiJ j σ a '' (T.cuspA j a n : Set (T.R j)) ⊆ baseForms T j σ a n

theorem **zero_mem_ratForms** (C : Cover T j σ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) : 0 ∈ C.ratForms a n

theorem **add_mem_ratForms** (C : Cover T j σ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) {φ ψ : GA D (T.V j σ) → ℂ} (hφ : φ ∈ C.ratForms a n) (hψ : ψ ∈ C.ratForms a n) : φ + ψ ∈ C.ratForms a n

theorem **smul_mem_ratForms** (C : Cover T j σ) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : C.d.c₂.F) {φ : GA D (T.V j σ) → ℂ} (hφ : φ ∈ C.ratForms a n) : C.σ₂ x • φ ∈ C.ratForms a n

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.CyclotomicFrobeniusPowerSeriesXEqMulQuotient

theorem **ps_hom_ext** {R B : Type} [CommRing R] [CommRing B] {f g : R⟦X⟧ →+* B} (hC : ∀ r, f (C r) = g (C r)) (hX : f X = g X) (hsep : ∀ b : B, (∀ n, f X ^ n ∣ b) → b = 0) : f = g

theorem **eq_zero_of_pow_dvd** {R : Type} [CommRing R] {a : R⟦X⟧} (ha : constantCoeff a = 0) (b : R⟦X⟧) (h : ∀ n, a ^ n ∣ b) : b = 0

@[simp] theorem **cyclotomicFrobeniusPowerSeries_X** : phi0 l R X = (1 + X) ^ l − 1

@[simp] theorem **cyclotomicFrobeniusPowerSeries_C** (r : R) : phi0 l R (C r) = C r

theorem **X_mul_qq** : X * qq l R = (1 + X) ^ l − 1

theorem **cyclotomicFrobeniusPowerSeries_X_eq_mul_quotient** : phi0 l R X = X * qq l R

theorem **cyclotomicFrobeniusPowerSeries_pow_apply_X** (m : ℕ) : (phi0 l R ^ m) X = (1 + X) ^ (l ^ m) − 1

theorem **Xs_eq** (s : ℕ) : Xs l R s = (phi0 l R ^ (s − 1)) X

theorem **cyclotomicFrobenius_iteratedVariable_eq_mul** (s : ℕ) : phi0 l R (Xs l R s) = Xs l R s * Qs l R s

@[simp] theorem **phiS_X** (Fr : R →+* R) : phiS l Fr X = (1 + X) ^ l − 1

@[simp] theorem **phiS_C** (Fr : R →+* R) (r : R) : phiS l Fr (C r) = C (Fr r)

@[simp] theorem **gam_X** (c : ℤ_[l]) : gam R c X = binomialSeries R c − 1

@[simp] theorem **gam_C** (c : ℤ_[l]) (r : R) : gam R c (C r) = C r

theorem **constantCoeff_gam_X** (c : ℤ_[l]) : constantCoeff (gam R c X) = 0

theorem **coeff_one_gam_X** (c : ℤ_[l]) : coeff 1 (gam R c X) = algebraMap ℤ_[l] R c

theorem **map_binomialSeries** {R' : Type} [CommRing R'] [Algebra ℤ_[l] R'] (h : R →+* R') (hh : ∀ z : ℤ_[l], h (algebraMap _ _ z) = algebraMap _ _ z) (c : ℤ_[l]) : PowerSeries.map h (binomialSeries R c) = binomialSeries R' c

theorem **map_gam** {R' : Type} [CommRing R'] [Algebra ℤ_[l] R'] (h : R →+* R') (hh : ∀ z : ℤ_[l], h (algebraMap _ _ z) = algebraMap _ _ z) (c : ℤ_[l]) (a : R⟦X⟧) : PowerSeries.map h (gam R c a) = gam R' c (PowerSeries.map h a)

private theorem **coeff_ode** (f : ℤ_[l]⟦X⟧) (n : ℕ) : coeff n ((1 + X) * d⁄dX ℤ_[l] f) = coeff (n + 1) f * (n + 1) + coeff n f * n

private theorem **ode_binomial** (c : ℤ_[l]) : (1 + X) * d⁄dX ℤ_[l] (binomialSeries ℤ_[l] c) = C c * binomialSeries ℤ_[l] c

private theorem **ode_unique** (c : ℤ_[l]) {f g : ℤ_[l]⟦X⟧} (hf : (1 + X) * d⁄dX ℤ_[l] f = C c * f) (hg : (1 + X) * d⁄dX ℤ_[l] g = C c * g) (h0 : constantCoeff f = constantCoeff g) : f = g

theorem **gam_binomialSeries** (c c' : ℤ_[l]) : gam R c (binomialSeries R c') = binomialSeries R (c * c')

theorem **algHom_ext_X** {f g : R⟦X⟧ →ₐ[R] R⟦X⟧} (h : f X = g X) (h0 : constantCoeff (f X) = 0) : f = g

theorem **gam_mul** (c c' : ℤ_[l]) (a : R⟦X⟧) : gam R (c * c') a = gam R c (gam R c' a)

@[simp] theorem **gam_one** (a : R⟦X⟧) : gam R (1 : ℤ_[l]) a = a

theorem **gam_l** : gam R (l : ℤ_[l]) = phi0 l R

theorem **cyclotomicFrobenius_comm_gammaAction** (c : ℤ_[l]) (a : R⟦X⟧) : phi0 l R (gam R c a) = gam R c (phi0 l R a)

@[simp] theorem **gamAut_apply** (c : ℤ_[l]ˣ) (a : R⟦X⟧) : gamAut R c a = gam R (c : ℤ_[l]) a

theorem **gam_X_eq** (c : ℤ_[l]) : gam R c X = X * vser R c

theorem **constantCoeff_vser** (c : ℤ_[l]) : constantCoeff (vser R c) = algebraMap ℤ_[l] R c

theorem **isUnit_vser** (c : ℤ_[l]ˣ) : IsUnit (vser R (c : ℤ_[l]))

theorem **eq_zero_of_l_pow_dvd** {B : Type} [CommRing B] [IsHausdorff (Ideal.span {(l : B)}) B] (b : B) (h : ∀ k, (l : B) ^ k ∣ b) : b = 0

theorem **isUnit_of_l_dvd** {B : Type} [CommRing B] [LComplete l B] {u x : B} (hu : IsUnit u) (h : (l : B) ∣ x − u) : IsUnit x

theorem **ofLoc_alg** (a : R⟦X⟧) : ofLoc l R (algebraMap _ _ a) = ιE l R a

theorem **algebraMap_completedLocalization** (r : R) : algebraMap R (completedLocalizedPowerSeriesRing l R) r = ιE l R (C r)

theorem **isUnit_ιE_X** : IsUnit (ιE l R X)

private theorem **mem_iff'** (S : Type) [CommRing S] (k : ℕ) (x : AdicCompletion (Ideal.span {(l : S)}) S) : (l : AdicCompletion (Ideal.span {(l : S)}) S) ^ k ∣ x ↔ AdicCompletion.eval (Ideal.span {(l : S)}) S k x = 0

private theorem **mem_iff** (k : ℕ) (x : completedLocalizedPowerSeriesRing l R) : (l : completedLocalizedPowerSeriesRing l R) ^ k ∣ x ↔ AdicCompletion.eval (Ideal.span {(l : Localization.Away (X : R⟦X⟧))}) _ k x = 0

theorem **exists_loc** (x : completedLocalizedPowerSeriesRing l R) (k : ℕ) : ∃ y, (l : completedLocalizedPowerSeriesRing l R) ^ k ∣ x − ofLoc l R y

theorem **completedLocalizedPowerSeries_dense_clearDenominators** (x : completedLocalizedPowerSeriesRing l R) (k : ℕ) : ∃ (a : R⟦X⟧) (m : ℕ), (l : completedLocalizedPowerSeriesRing l R) ^ k ∣ x * ιE l R (X ^ m) − ιE l R a

theorem **prime_pow_dvd_of_dvd_completedLocalization** (a : R⟦X⟧) (k : ℕ) (h : (l : completedLocalizedPowerSeriesRing l R) ^ k ∣ ιE l R a) : (l : R⟦X⟧) ^ k ∣ a

theorem **ιE_injective** [IsHausdorff (Ideal.span {(l : R)}) R] : Function.Injective (ιE l R)

theorem **completedLocalizedPowerSeries_ringHom_ext** {B : Type} [CommRing B] [IsHausdorff (Ideal.span {(l : B)}) B] {f g : completedLocalizedPowerSeriesRing l R →+* B} (h : ∀ a, f (ιE l R a) = g (ιE l R a)) : f = g

@[simp] theorem **liftE_ιE** {B : Type} [CommRing B] [LComplete l B] (f : R⟦X⟧ →+* B) (hf : IsUnit (f X)) (a : R⟦X⟧) : liftE (l

@[simp] theorem **mapE_ιE** {R' : Type} [CommRing R'] (h : R →+* R') (a : R⟦X⟧) : mapE h (ιE l R a) = ιE l R' (PowerSeries.map h a)

theorem **isUnit_ιE_phi** : IsUnit (ιE l R ((1 + X) ^ l − 1))

@[simp] theorem **phiES_ιE** (Fr : R →+* R) (a : R⟦X⟧) : phiES Fr (ιE l R a) = ιE l R (phiS l Fr a)

@[simp] theorem **completedLocalizedLinearFrobenius_comp** (a : R⟦X⟧) : completedLocalizedLinearFrobenius l R (ιE l R a) = ιE l R (phi0 l R a)

theorem **isUnit_ιE_phi_pow** (m : ℕ) : IsUnit (ιE l R ((phi0 l R ^ m) X))

theorem **isUnit_ιE_Xs** (s : ℕ) : IsUnit (ιE l R (Xs l R s))

theorem **isUnit_ιE_Qs** (s : ℕ) : IsUnit (ιE l R (Qs l R s))

@[simp] theorem **completedLocalizedGammaAction_comp** (c : ℤ_[l]ˣ) (a : R⟦X⟧) : completedLocalizedGammaAction R c (ιE l R a) = ιE l R (gam R (c : ℤ_[l]) a)

theorem **EE_isUnit_π** : IsUnit (algebraMap (LetA O n) (EE l O n) letπ)

theorem **EE_inj** [IsHausdorff (Ideal.span {(l : O)}) O] : Function.Injective (algebraMap (LetA O n) (EE l O n))

private theorem **pi_dvd** {ι : Type} {B : Type} [CommRing B] {x y : ι → B} : x ∣ y ↔ ∀ i, x i ∣ y i

theorem **EE_dense** [NeZero n] (x : EE l O n) (k : ℕ) : ∃ (a : LetA O n) (m : ℕ), x * algebraMap (LetA O n) (EE l O n) (letπ ^ m) - algebraMap (LetA O n) (EE l O n) a ∈ Ideal.span {(l : EE l O n) ^ k}

theorem **EE_sat** (a : LetA O n) (k : ℕ) (h : algebraMap (LetA O n) (EE l O n) a ∈ Ideal.span {(l : EE l O n) ^ k}) : ∃ b : LetA O n, a = (l : LetA O n) ^ k * b

theorem **phiEE_ext** (a : LetA O n) : phiEE l O n (algebraMap (LetA O n) _ a) = algebraMap (LetA O n) _ (phiA l O n a)

theorem **gamEE_ext** (c : ℤ_[l]ˣ) (a : LetA O n) : gamEE O n c (algebraMap (LetA O n) _ a) = algebraMap (LetA O n) _ (gamA O n (c : ℤ_[l]) a)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.DecompositionInvariantsCondition

theorem **epJ_eq** {D : D7} (h : AdmissibleParameterJ.ConclusionJa D) : epJ D = Exists.choose h

theorem **canon_ourX** (L : SizeData₂ D) (HW : HWOf D.loc) (R : KillingConditions.RowK) (r : FamilyRepresentationConstants.RFam D) (ρ : FamilyRepresentationConstants.RhoFam D) (e' : D.nd.Loc → HeightOneSpectrum (𝒪 D.host.E) → ℕ) (exc : Finset ℕ) (I : TwIn D) (h : GlobalTheory.PtCore.PtFam D r ρ) : Canon L HW R (ourX L HW R r ρ e' exc I)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.DeltaStructureOfWittLengthTwoSection

theorem **map_cc** (f : A →+* B) (a b : A) : f (cc l a b) = cc l (f a) (f b)

theorem **l_mul_cc** [hl : Fact l.Prime] (a b : A) : (l : A) * cc l a b = (a + b) ^ l - a ^ l - b ^ l

theorem **mapF_add** (f : A →+* B) (x y : W2 l A) : mapF f (x + y) = mapF f x + mapF f y

theorem **mapF_mul** (f : A →+* B) (x y : W2 l A) : mapF f (x * y) = mapF f x * mapF f y

theorem **gh_add** (x y : W2 l A) : gh (x + y) = gh x + gh y

theorem **gh_mul** (x y : W2 l A) : gh (x * y) = gh x * gh y

theorem **ext_gh** {σ : Type} {x y : W2 l (MvPolynomial σ ℤ)} (h0 : x.fst = y.fst) (h1 : gh x = gh y) : x = y

theorem **exists_lift** (x : W2 l A) : ∃ y : W2 l (MvPolynomial A ℤ), mapF (MvPolynomial.eval₂Hom (Int.castRingHom A) id) y = x

theorem **mem_δLocus** {D : Delta l A} {D' : Delta l B} {f : A →+* B} {a : A} : a ∈ δLocus D D' f ↔ f (D.δ a) = D'.δ (f a)

theorem **polyDelta_C** (D : Delta l A) (a : A) : (polyDelta D).δ (MvPolynomial.C a) = MvPolynomial.C (D.δ a)

theorem **polyDelta_X** (D : Delta l A) (i : ℕ) : (polyDelta D).δ (MvPolynomial.X i) = MvPolynomial.X (i + 1)

theorem **IsCompl.self** {L : Type} [CommRing L] (J : Ideal L) [h : IsAdicComplete J L] : IsCompl (RingHom.id L) J

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.DetCharacterMemPrincipalCongruenceFixedVectors

theorem **Sys.IsOf.has_iff** {S : Sys g} {L : LabelDatum g} (h : S.IsOf L) {n : ℕ} {V : Type} [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (Fin n) ℚ_[g]) V) (f : (ZMod g)[X]) : Has S π f ↔ HasLab L π f

theorem **eSerTie_iff** (D : D7) : ESerTie D ↔ ∃ L, LabTie D L

theorem **LabTie.sysTie** {D : D7} {L : ∀ k, LabelDatum (D.host.g k)} {S : ∀ k, Sys (D.host.g k)} (h : LabTie D L) (hS : ∀ k, (S k).IsOf (L k)) : SysTie D S

theorem **alg_sum** {G U : Type} [Group G] [Fintype G] [AddCommGroup U] [Module ℂ U] (ρ : Representation ℂ G U) (a : MonoidAlgebra ℂ G) (w : U) : ρ.asAlgebraHom a w = ∑ x, a.coeff x • ρ x w

theorem **det_blocks** (a b : ℕ) (x : Gn g a) (y : Gn g b) : det (blk a b (a + b) x y) = det x * det y

theorem **det_rad** {a b : ℕ} {u : Gn g (a + b)} (h0 : InPar a (u : Mt g (a + b))) (h1 : tl a (u : Mt g (a + b)) = 1) (h2 : br a b (u : Mt g (a + b)) = 1) : det u = 1

theorem **alg_ev** (χ : (ZMod g)ˣ →* ℂˣ) {n : ℕ} (a : MonoidAlgebra ℂ (Gn g n)) (v : ℂ) : (Model.chRep (χ.comp (det (n

theorem **charIn_iff** (S : Sys g) (χ : (ZMod g)ˣ →* ℂˣ) (b : ℕ) (h : (ZMod g)[X]) : CharIn S χ b h ↔ ev χ (S.e b h) = 1

theorem **exists_charIn** (S : Sys g) (χ : (ZMod g)ˣ →* ℂˣ) (b : ℕ) : ∃ h, CharIn S χ b h ∧ h.Monic ∧ h.natDegree = b ∧ h.coeff 0 ≠ 0

theorem **charIn_mul** (S : Sys g) (χ : (ZMod g)ˣ →* ℂˣ) {a b : ℕ} {f f' : (ZMod g)[X]} (h : CharIn S χ a f) (h' : CharIn S χ b f') : CharIn S χ (a + b) (f * f')

theorem **charIn_pow** (S : Sys g) (χ : (ZMod g)ˣ →* ℂˣ) {h : (ZMod g)[X]} (h1 : CharIn S χ 1 h) (b : ℕ) : CharIn S χ b (h ^ b)

theorem **tamePart_tame** (χ : (ZMod g)ˣ →* ℂˣ) : TamePart (Model.tame χ) χ

theorem **det_inc** (k : GL (Fin n) ℤ_[g]) : μ (det (inc g n k)) = χ (det (red g n k))

theorem **detCharacter_mem_principalCongruenceFixedVectors** (v : ℂ) : v ∈ fixK1 (Model.detRep n μ)

theorem **det_lift** (x : Gn g n) (v : ℂ) : Model.detRep n μ (lift g n x) v = χ (det x) * v

theorem **actK_det** (a : MonoidAlgebra ℂ (Gn g n)) (v : ℂ) : actK (Model.detRep n μ) a v = ev χ a * v

theorem **has_det** (S : Sys g) (f : (ZMod g)[X]) : Has S (Model.detRep n μ) f ↔ CharIn S χ n f

theorem **irrSm_det** : IrrSm (Rep.of (Model.detRep n μ))

theorem **isParVec_det** (r t : ℕ) (v : ℂ) : IsParVec (Model.detRep (r + t) μ) r t χ v

theorem **actsThrough_det** (r t : ℕ) : ActsThrough (Rep.of (Model.detRep (r + t) μ)) r t χ (⊤ : Submodule ℂ ℂ) fun a => ev χ a • LinearMap.id

theorem **act_eq_of_actsThrough_det** {r t : ℕ} {χ' : (ZMod g)ˣ →* ℂˣ} {M : Type} [AddCommGroup M] [Module ℂ M] {W : Submodule ℂ M} {act : MonoidAlgebra ℂ (Gn g r) → Module.End ℂ M} (H : ActsThrough (Rep.of (Model.detRep (r + t) μ)) r t χ' W act) {a : MonoidAlgebra ℂ (Gn g

r)} (ha : a ∈ Subalgebra.center ℂ (MonoidAlgebra ℂ (Gn g r))) {F : M} (hF : F ∈ W) : act a F = ev χ a • F

theorem **prod_linear** (τ : Formalism.CyclotomicNorms.Fbar g →* ℂ) (c : ZMod g) : ((Formalism.CyclotomicNorms.rootsBar (X − C c)).map fun x => X − C (τ x)).prod = X − C (τ (algebraMap (ZMod g) (Formalism.CyclotomicNorms.Fbar g) c))

theorem **exists_serParam** (e : ∀ n : ℕ, (ZMod g)[X] → MonoidAlgebra ℂ (Gn g n)) (τ : Formalism.CyclotomicNorms.Fbar g →* ℂ) (h : ∀ n (π : QRep g n), IrrSm π → ∀ f f', (∃ v ∈ fixK1 π.ρ, actK π.ρ (e n f) v ≠ 0) → (∃ v ∈ fixK1 π.ρ, actK π.ρ (e n f') v ≠ 0) → f = f') : ∃ ine, SerParam e τ ine

theorem **isClassMap_mk** : IsClassMap (LP

theorem **ActsThrough.exists_parVec** {n r t : ℕ} {π : QRep g n} {χ : (ZMod g)ˣ →* ℂˣ} {M : Type} [AddCommGroup M] [Module ℂ M] {W : Submodule ℂ M} {act : MonoidAlgebra ℂ (Gn g r) → Module.End ℂ M} (H : ActsThrough π r t χ W act) {F : M} (hF : F ∈ W) {a : MonoidAlgebra ℂ (Gn g r)} (ha : a ∈ Subalgebra.center ℂ (MonoidAlgebra ℂ (Gn g r))) (h0 : act a F ≠ 0) : ∃ u, IsParVec π.ρ r t χ u ∧ actTop π.ρ r t a u ≠ 0

theorem **Sys.IsOf.charIn_one** {S : Sys g} {L : LabelDatum g} (h : S.IsOf L) {χ : (ZMod g)ˣ →* ℂˣ} {c : ZMod g} (hc : IsCharLab L χ c) (χ' : (ZMod g)ˣ →* ℂˣ) : CharIn S χ' 1 (X − C c) ↔ χ' = χ

theorem **Sys.IsOf.charIn** {S : Sys g} {L : LabelDatum g} (h : S.IsOf L) {χ : (ZMod g)ˣ →* ℂˣ} {c : ZMod g} (hc : IsCharLab L χ c) (b : ℕ) : CharIn S χ b ((X − C c) ^ b)

theorem **Loc.not_isOf_coarse** (hg : g ≠ 2) (L : LabelDatum g) : ¬ (Loc.coarse g).IsOf L

theorem **LabParam.char** {D : D7} {L : ∀ k, LabelDatum (D.host.g k)} {rc : Corr D} (h : LabParam D L rc) (k : Fin 2) (v : D.nd.PlG k) (χ : (ZMod (D.host.g k))ˣ →* ℂˣ) (u : (ZMod (D.host.g k))ˣ) (hu : algebraMap (ZMod (D.host.g k)) (Formalism.CyclotomicNorms.Fbar (D.host.g k)) u = (L k).gen 1) : D.naut.inePol (rc k v 1 (Rep.of (Model.detRep 1 (Model.tame χ)))) (D.ntypes.tg k v) = X − C ((χ u : ℂˣ) : ℂ)

theorem **locType_of_serParam** {D : D7} {S : ∀ k, Sys (D.host.g k)} {rc : Corr D} (hC : LocComp D rc) (hP : ∀ k v, SerParam (S k).e (D.ntypes.teichC k) fun n π => D.naut.inePol (rc k v n π) (D.ntypes.tg k v)) : LocType D S

theorem **locType_of_ties** {D : D7} {L : ∀ k, LabelDatum (D.host.g k)} {S : ∀ k, Sys (D.host.g k)} {rc : Corr D} (hL : LabTie D L) (hS : ∀ k, (S k).IsOf (L k)) (hC : LocComp D rc) (hP : LabParam D L rc) : LocType D S

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.DieuUnifDiagNotSurjective

theorem **onLie_eq** (H : VarpiDivModule p O S) (a : O) : H.ι.onLie a = algebraMap O S a • LinearMap.id

theorem **ι_algebraMap** [Algebra ℤ_[p] O] (H : VarpiDivModule p O S) (a : ℤ_[p]) : H.ι (algebraMap ℤ_[p] O a) = algebraMap ℤ_[p] (End H.G) a

@[simp] theorem **id_f** (H : VarpiDivModule p O S) : Hom.f (𝟙 H) = 𝟙 H.G

@[simp] theorem **comp_f** {H H' H'' : VarpiDivModule p O S} (f : H ⟶ H') (g : H' ⟶ H'') : Hom.f (f ≫ g) = f.f ≫ g.f

@[ext] theorem **hom_ext** {H H' : VarpiDivModule p O S} {f g : H ⟶ H'} (h : f.f = g.f) : f = g

theorem **hasOHeight_iff** [Algebra ℤ_[p] O] [Module.Free ℤ_[p] O] [Module.Finite ℤ_[p] O] (H : VarpiDivModule p O S) (n : ℕ) : H.HasOHeight n ↔ H.G.height = n ∗ Module.finrank ℤ_[p] O

theorem **padicAction_onLie** [Algebra ℤ_[p] S] (G : PDivGroup p S) (a : ℤ_[p]) (x : G.Lie) : (PDivGroup.padicAction G).onLie a x = algebraMap ℤ_[p] S a • x

@[simp] theorem **ofPDiv_G** [Algebra ℤ_[p] S] (G : PDivGroup p S) (h : IsNilpotent (p : S)) : (ofPDiv p G h).G = G

theorem **ofPDiv_hasOHeight** [Algebra ℤ_[p] S] (G : PDivGroup p S) (h : IsNilpotent (p : S)) : (ofPDiv p G h).HasOHeight G.height

theorem **natCast_resRing** : ((p : ℕ) : ResRing p) = 0

theorem **muMod_hasOHeight** : (muMod p).HasOHeight 1

theorem **unifDiag_apply** (ϖ : R) (i j : Fin n) : unifDiag ϖ i j = if i = j then (if i = 0 then ϖ else 1) else 0

theorem **unifDiag_mul_coUnifDiag** (ϖ : R) : (unifDiag ϖ : Matrix (Fin n) (Fin n) R) ∗ coUnifDiag ϖ = ϖ • 1

theorem **coUnifDiag_mul_unifDiag** (ϖ : R) : (coUnifDiag ϖ : Matrix (Fin n) (Fin n) R) ∗ unifDiag ϖ = ϖ • 1

theorem **unifDiag_mem** (ϖ : R) : unifDiag ϖ ∈ BsetInt n ϖ

theorem **exists_mul_eq_smul** {ϖ : R} {β : Matrix (Fin n) (Fin n) R} (hβ : β ∈ BsetInt n ϖ) : ∃ γ : Matrix (Fin n) (Fin n) R, β ∗ γ = ϖ • 1 ∧ γ ∗ β = ϖ • 1

@[simp] theorem **sigmaGL_val** (σ : R ≃+* R) (g : GL (Fin n) R) : (sigmaGL σ g).val = (g.val).map σ

@[simp] theorem **sigmaConj_one** (σ : R ≃+* R) (β : Matrix (Fin n) (Fin n) R) : sigmaConj σ 1 β = β

theorem **sigmaConj_mul** (σ : R ≃+* R) (g h : GL (Fin n) R) (β : Matrix (Fin n) (Fin n) R) : sigmaConj σ (g ∗ h) β = sigmaConj σ g (sigmaConj σ h β)

theorem **sigmaConj_mem** (σ : R ≃+* R) (g : GL (Fin n) R) {ϖ : R} {β : Matrix (Fin n) (Fin n) R} (hβ : β ∈ BsetInt n ϖ) : sigmaConj σ g β ∈ BsetInt n ϖ

@[simp] theorem **dieuF_apply** (σ : R ≃+* R) (β : Matrix (Fin n) (Fin n) R) (x : Fin n → R) : dieuF σ β x = β ∗ᵥ fun i => σ (x i)

@[simp] theorem **dieuV_apply** (σ : R ≃+* R) (γ : Matrix (Fin n) (Fin n) R) (y : Fin n → R) (i : Fin n) : dieuV σ γ y i = σ.symm ((γ ∗ᵥ y) i)

theorem **mul_coMat** {ϖ : R} (β : BsetInt n ϖ) : β.1 ∗ coMat β = ϖ • 1

theorem **coMat_mul** {ϖ : R} (β : BsetInt n ϖ) : coMat β ∗ β.1 = ϖ • 1

@[simp] theorem **dieu_F** (σ : R ≃+* R) {ϖ : R} (hϖ : σ ϖ = ϖ) (β : BsetInt n ϖ) (x : Fin n → R) : (dieu σ hϖ β).F x = β.1 ∗ᵥ fun i => σ (x i)

theorem **finrank_dieu** [Nontrivial R] (σ : R ≃+* R) {ϖ : R} (hϖ : σ ϖ = ϖ) (β : BsetInt n ϖ) : Module.finrank R (dieu σ hϖ β).M = n

theorem **dieu_unifDiag_F_zero** (σ : R ≃+* R) {ϖ : R} (hϖ : σ ϖ = ϖ) : (dieu σ hϖ ⟨unifDiag ϖ, unifDiag_mem ϖ⟩).F (Pi.single (0 : Fin n) (1 : R)) = ϖ • Pi.single (0 : Fin n) (1 : R)

theorem **dieu_unifDiag_F_ne** (σ : R ≃+* R) {ϖ : R} (hϖ : σ ϖ = ϖ) {i : Fin n} (hi : i ≠ 0) : (dieu σ hϖ ⟨unifDiag ϖ, unifDiag_mem ϖ⟩).F (Pi.single i (1 : R)) = Pi.single i (1 : R)

theorem **dieu_unifDiag_not_surjective** (σ : R ≃+* R) {ϖ : R} (hϖ : σ ϖ = ϖ) (hu : ¬ IsUnit ϖ) : ¬ Function.Surjective (dieu σ hϖ ⟨unifDiag (n

theorem **RelDieudonne.V_comm_of_F_comm** {D D' : RelDieudonne R σ ϖ} (e : D.M ≃ₗ[R] D'.M) (hF : ∀ x, e (D.F x) = D'.F (e x)) (hreg : ∀ y : D'.M, ϖ • y = 0 → y = 0) (x : D.M) : e (D.V x) = D'.V (e x)

@[simp] theorem **glEquiv_apply** (g : GL (Fin n) R) (x : Fin n → R) : glEquiv g x = g.val *ᵥ x

theorem **sigma_mulVec** (σ : R ≃+* R) (g : Matrix (Fin n) (Fin n) R) (x : Fin n → R) : (fun i => σ ((g *ᵥ x) i)) = g.map σ *ᵥ fun i => σ (x i)

theorem **nonempty_iso_of_rel** (hϖ : σ ϖ = ϖ) (hreg : ϖ ∈ nonZeroDivisors R) {β β' : BsetInt n ϖ} (h : (sigmaSetoid n σ ϖ) β β') : Nonempty (RelDieudonne.Iso (dieu σ hϖ β) (dieu σ hϖ β'))

theorem **det_unifDiag** (ϖ : R) : (unifDiag ϖ : Matrix (Fin n) (Fin n) R).det = ϖ

theorem **exists_mem_Bset** (K : Type) [Field K] [Algebra R K] (ϖ : R) (h : algebraMap R K ϖ ≠ 0) : ∃ g ∈ Bset n K ϖ, g.val = (unifDiag ϖ).map (algebraMap R K)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.EmbedsLevelZeroDepthZero

theorem **levelRep_apply** (ρ : Representation ℂ G V) {P J : Subgroup G} (hP : P ≤ Subgroup.normalizer (J : Set G)) (q : P →* M) (hq : Function.Surjective q) (hk : ∀ p, q p = 1 → (p : G) ∈ J) (p : P) (v : fixed ρ J) : (levelRep ρ hP q hq hk (q p) v : V) = ρ p v

theorem **redK_apply** (g : KO k n) (i j : n) : (redK k n g : Matrix n n 𝒦[k]) i j = IsLocalRing.residue 𝒪[k] (((g : GL n k) : Matrix n n k) i j, StepData.intM_of_mem_KO g.2 i j)

theorem **redK_eq_iff** (g : KO k n) (i j : n) (x : 𝒪[k]) : (redK k n g : Matrix n n 𝒦[k]) i j = IsLocalRing.residue 𝒪[k] x ↔ valuation k (((g : GL n k) : Matrix n n k) i j - x) < 1

theorem **redK_eq_zero_iff** (g : KO k n) (i j : n) : (redK k n g : Matrix n n 𝒦[k]) i j = 0 ↔ valuation k (((g : GL n k) : Matrix n n k) i j) < 1

theorem **redK_surjective** : Function.Surjective (redK k n)

theorem **redK_eq_one_iff** (g : KO k n) : redK k n g = 1 ↔ (g : GL n k) ∈ KOne k n

theorem **mem_levelPre** {H : Subgroup (GL n 𝒦[k])} {g : GL n k} : g ∈ levelPre H ↔ ∃ h : g ∈ KO k n, redK k n ⟨g, h⟩ ∈ H

theorem **coe_mem_levelPre** {H : Subgroup (GL n 𝒦[k])} (g : KO k n) : (g : GL n k) ∈ levelPre H ↔ redK k n g ∈ H

theorem **levelPre_le_KO** (H : Subgroup (GL n 𝒦[k])) : levelPre H ≤ KO k n

theorem **KOne_le_levelPre** (H : Subgroup (GL n 𝒦[k])) : KOne k n ≤ levelPre H

theorem **levelPre_bot** : levelPre (⊥ : Subgroup (GL n 𝒦[k])) = KOne k n

theorem **levelPre_top** : levelPre (⊤ : Subgroup (GL n 𝒦[k])) = KO k n

theorem **levelPre_mono** {H H' : Subgroup (GL n 𝒦[k])} (h : H ≤ H') : levelPre H ≤ levelPre H'

theorem **isOpen_levelPre** (H : Subgroup (GL n 𝒦[k])) : IsOpen (levelPre H : Set (GL n k))

theorem **isCompact_levelPre** (H : Subgroup (GL n 𝒦[k])) : IsCompact (levelPre H : Set (GL n k))

theorem **levelPre_le_normalizer** {H U : Subgroup (GL n 𝒦[k])} (h : H ≤ Subgroup.normalizer (U : Set (GL n 𝒦[k]))) : levelPre H ≤ Subgroup.normalizer (levelPre U : Set (GL n k))

theorem **KO_le_normalizer** : KO k n ≤ Subgroup.normalizer (KOne k n : Set (GL n k))

@[simp] theorem **levelZero_apply** (π : Representation ℂ (GL n k) V) (g : KO k n) (v : fixed π (KOne k n)) : (levelZero π (redK k n g) v : V) = π g v

theorem **mem_parahoric** {e : (Σ a, m a) ≃ n} {g : GL n k} : g ∈ parahoric k e ↔ g ∈ KO k n ∧ ∀ a b (i : m a) (j : m b), b < a → valuation k ((g : Matrix n n k) (e ⟨a, i⟩) (e ⟨b, j⟩)) < 1

theorem **mem_proRad** {e : (Σ a, m a) ≃ n} {g : GL n k} : g ∈ proRad k e ↔ g ∈ parahoric k e ∧ ∀ a (i j : m a), valuation k ((g : Matrix n n k) (e ⟨a, i⟩) (e ⟨a, j⟩) - (1 : Matrix (m a) (m a) k) i j) < 1

theorem **proRad_le_parahoric** (e : (Σ a, m a) ≃ n) : proRad k e ≤ parahoric k e

theorem **parahoric_le_normalizer** (e : (Σ a, m a) ≃ n) : parahoric k e ≤ Subgroup.normalizer (proRad k e : Set (GL n k))

theorem **parahoricProj_eq_one_iff** (e : (Σ a, m a) ≃ n) (p : parahoric k e) : parahoricProj k e p = 1 ↔ (p : GL n k) ∈ proRad k e

theorem **parahoricProj_surjective** (e : (Σ a, m a) ≃ n) : Function.Surjective (parahoricProj k e)

theorem **redLevi_apply** (g : leviK k m) (a : α) : redLevi k m g a = redK k (m a) ⟨g.1 a, g.2 a trivial⟩

theorem **redLevi_surjective** : Function.Surjective (redLevi k m)

theorem **redLevi_eq_one_iff** (g : leviK k m) : redLevi k m g = 1 ↔ (g : Levi k m) ∈ leviKOne k m

theorem **leviK_le_normalizer** : leviK k m ≤ Subgroup.normalizer (leviKOne k m : Set (Levi k m))

theorem **leviEmb_mem_KO** (e : (Σ a, m a) ≃ n) {g : Levi k m} (hg : g ∈ leviK k m) : leviEmb k e g ∈ KO k n

theorem **redK_leviEmb** (e : (Σ a, m a) ≃ n) (g : leviK k m) : redK k n ⟨leviEmb k e g, leviEmb_mem_KO e g.2⟩ = leviEmb 𝒦[k] e (redLevi k m g)

theorem **leviEmb_mem_parahoric** (e : (Σ a, m a) ≃ n) (g : leviK k m) : leviEmb k e g ∈ parahoric k e

theorem **parahoricProj_leviEmb** (e : (Σ a, m a) ≃ n) (g : leviK k m) : parahoricProj k e ⟨leviEmb k e g, leviEmb_mem_parahoric e g⟩ = redLevi k m g

theorem **KOne_le_proRad** (e : (Σ a, m a) ≃ n) : KOne k n ≤ proRad k e

theorem **jLevi_le_KOne** (e : (Σ a, m a) ≃ n) : jLevi k e ≤ KOne k n

theorem **jMinus_le_proRad** (e : (Σ a, m a) ≃ n) : jMinus k e ≤ proRad k e

theorem **jLevi_le_proRad** (e : (Σ a, m a) ≃ n) : jLevi k e ≤ proRad k e

theorem **jPlus_le_proRad** (e : (Σ a, m a) ≃ n) : jPlus k e ≤ proRad k e

@[simp] theorem **parahoricInv_apply** (e : (Σ a, m a) ≃ n) (π : Representation ℂ (GL n k) V) (p : parahoric k e) (v : fixed π (proRad k e)) : (parahoricInv e π (parahoricProj k e p) v : V) = π p v

@[simp] theorem **parahoricInv_redLevi** (e : (Σ a, m a) ≃ n) (π : Representation ℂ (GL n k) V) (g : leviK k m) (v : fixed π (proRad k e)) : (parahoricInv e π (redLevi k m g) v : V) = π (leviEmb k e g) v

@[simp] theorem **leviLevelZero_apply** (τ : Representation ℂ (Levi k m) W) (g : leviK k m) (v : fixed τ (leviKOne k m)) : (leviLevelZero τ (redLevi k m g) v : W) = τ g v

@[simp] theorem **toJacquet_apply** (e : (Σ a, m a) ≃ n) (π : Representation ℂ (GL n k) V) (v : fixed π (proRad k e)) : (toJacquet e π v : Jacquet π (unipRad k e)) = Submodule.Quotient.mk (v : V)

theorem **detChar_of_mem_KO** {ν : kˣ →* ℂˣ} (hν : IsUnram ν) {g : GL n k} (hg : g ∈ KO k n) : detChar n ν g = 1

theorem **deltaHalf_of_mem_leviK** {g : Levi k m} (hg : g ∈ leviK k m) : deltaHalf k m g = 1

theorem **redProd_surjective** : Function.Surjective (redProd k a b)

@[simp] theorem **prodLevelZero_apply** (τ : Representation ℂ (GL a k × GL b k) W) (g : (KO k a).prod (KO k b)) (v : fixed τ ((KOne k a).prod (KOne k b))) : (prodLevelZero τ (redProd k a b g) v : W) = τ g v

@[simp] theorem **parahoricInv₂_apply** (e : a ⊕ b ≃ n) (π : Representation ℂ (GL n k) V) (g : (KO k a).prod (KO k b)) (v : fixed π (proRad k (two e))) : (parahoricInv₂ e π (redProd k a b g) v : V) = π (emb₂ k e g) v

@[simp] theorem **toJacquet₂_apply** (e : a ⊕ b ≃ n) (π : Representation ℂ (GL n k) V) (v : fixed π (proRad k (two e))) : (toJacquet₂ e π v : Jacquet π (unipRad k (two e))) = Submodule.Quotient.mk (v : V)

theorem **unipRad_eq_of_le** (e : (Σ a, m a) ≃ Fin N) {c : Fin N → ℕ} (hc : ∀ i j, c j ≤ c i ↔ blk e j ≤ blk e i) : unipRad R e = FiniteGL.unipRadical R c

theorem **unipRad_eq_unipRadical** (e : (Σ a, Fin (d a)) ≃ Fin N) : unipRad R e = FiniteGL.unipRadical R (Parabolic.blockOf e)

theorem **unipRad_two_eq** {r t : ℕ} (hN : r + t = N) : unipRad R (two (stdSum hN)) = FiniteGL.unipRadical R (FiniteGL.twoBlocks N r)

theorem **coe_leviEquiv_symm** (e : (Σ a, Fin (d a)) ≃ Fin N) (x : Levi R fun a => Fin (d a)) : ((Parabolic.leviEquiv e).symm x : GL (Fin N) R) = leviEmb R e x

theorem **coe_leviTwo_symm** {r t : ℕ} (hN : r + t = N) {c : Fin N → ℕ} (hc : Parabolic.IsTwoBlock r c) (x : GL (Fin r) R × GL (Fin t) R) : ((Parabolic.leviTwo hN hc).symm x : GL (Fin N) R) = emb₂ R (stdSum hN) x

theorem **valuation_eq_one_iff** {x : kˣ} : valuation k (x : k) = 1 ↔ ∃ u, unitIncl k u = x

theorem **IsTameLift.mul** {χ' ψ' : kˣ →* ℂˣ} {χ ψ : 𝒪[k]ˣ →* ℂˣ} (h : IsTameLift χ' χ) (h' : IsTameLift ψ' ψ) : IsTameLift (χ' * ψ') (χ * ψ)

theorem **IsTameLift.detChar_eq** {χ' : kˣ →* ℂˣ} {χ : 𝒪[k]ˣ →* ℂˣ} (h : IsTameLift χ' χ) (g : KO k n) : detChar n χ' g = detChar n χ (redK k n g)

theorem **IsTameLift.isTame** {χ' : kˣ →* ℂˣ} {χ : 𝒪[k]ˣ →* ℂˣ} (h : IsTameLift χ' χ) : IsTame χ'

theorem **isSmoothChar_of_isTame** {χ' : kˣ →* ℂˣ} (h : IsTame χ') : IsSmoothChar χ'

theorem **exists_uniformizer** : ∃ ϖ : kˣ, valuation k (ϖ : k) = uniformizer k

theorem **exists_isTameLift** (χ : 𝒪[k]ˣ →* ℂˣ) {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (c : ℂˣ) : ∃ χ' : kˣ →* ℂˣ, IsTameLift χ' χ ∧ χ' ϖ = c

theorem **mem_ZK** {g : GL n k} : g ∈ ZK k n ↔ ∃ z : kˣ, ∃ x ∈ KO k n, GeneralLinearGroup.scalar n z * x = g

theorem **KO_le_ZK** : KO k n ≤ ZK k n

theorem **scalar_mem_ZK** (z : kˣ) : GeneralLinearGroup.scalar n z ∈ ZK k n

theorem **scalar_mem_KO** (u : 𝒪[k]ˣ) : GeneralLinearGroup.scalar n (unitIncl k u) ∈ KO k n

theorem **redK_scalar** (u : 𝒪[k]ˣ) : redK k n ⟨_, scalar_mem_KO u⟩ = GeneralLinearGroup.scalar n (unitRes k u)

theorem **extendsCentral_iff** [Nonempty n] {ω : kˣ →* ℂˣ} {ρ : Representation ℂ (GL n 𝒪[k]) W} : ExtendsCentral ω ρ ↔ ∀ (u : 𝒪[k]ˣ) (w : W), ρ (GeneralLinearGroup.scalar n (unitRes k u)) w = (ω (unitIncl k u) : ℂ) • w

theorem **IsTameLift.extendsCentral** [Nonempty n] {ω : kˣ →* ℂˣ} {ω₀ : 𝒪[k]ˣ →* ℂˣ} (h : IsTameLift ω ω₀) {ρ : Representation ℂ (GL n 𝒪[k]) W} (hρ : HasCentralChar ρ ω₀) : ExtendsCentral ω ρ

theorem **exists_hasCentralChar** {R : Type} [CommRing R] [FiniteDimensional ℂ W] {ρ : Representation ℂ (GL n R) W} (h : ρ.IsIrreducible) : ∃ ω, HasCentralChar ρ ω

theorem **extZK_apply** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) (z : kˣ) (g : KO k n) (x : ZK k n) (hx : GeneralLinearGroup.scalar n z * g = x) (w : W) : extZK ρ ω h x w = (ω z : ℂ) • ρ (redK k n g) w

@[simp] theorem **extZK_KO** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) (g : KO k n) (w : W) : extZK ρ ω h ⟨g, KO_le_ZK g.2⟩ w = ρ (redK k n g) w

@[simp] theorem **extZK_scalar** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) (z : kˣ) (w : W) : extZK ρ ω h ⟨_, scalar_mem_ZK z⟩ w = (ω z : ℂ) • w

@[simp] theorem **depthZero_apply** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) (g x : GL n k) (f : DepthZeroSp ρ ω h) : (depthZero ρ ω h g f).1.1.1 x = f.1.1.1 (x * g)

theorem **isSmooth_depthZero** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) : IsSmooth (depthZero ρ ω h)

theorem **hasCentralChar_depthZero** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) : HasCentralChar (depthZero ρ ω h) ω

theorem **canonFun_mul** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) (w : W) (g : GL n k) (x : ZK k n) : canonFun ρ ω h w (g * x) = canonFun ρ ω h (extZK ρ ω h x w) g

theorem **canon_injective** (ρ : Representation ℂ (GL n 𝒪[k]) W) (ω : kˣ →* ℂˣ) (h : ExtendsCentral ω ρ) : Function.Injective (canon ρ ω h)

theorem **isUnram_mul** {ν μ : kˣ →* ℂˣ} (h : IsUnram ν) (h' : IsUnram μ) : IsUnram (ν * μ)

theorem **isUnram_inv** {ν : kˣ →* ℂˣ} (h : IsUnram ν) : IsUnram ν⁻¹

theorem **InertEquiv.of_equiv** {π : Representation ℂ (GL n k) V} {σ : Representation ℂ (GL n k) W} (e : π.Equiv σ) : InertEquiv π σ

theorem **InertEquiv.symm** {π : Representation ℂ (GL n k) V} {σ : Representation ℂ (GL n k) W} (h : InertEquiv π σ) : InertEquiv σ π

theorem **InertEquiv.trans** {X : Type} [AddCommGroup X] [Module ℂ X] {π : Representation ℂ (GL n k) V} {σ : Representation ℂ (GL n k) W} {τ : Representation ℂ (GL n k) X} (h : InertEquiv π σ) (h' : InertEquiv σ τ) : InertEquiv π τ

theorem **InertEquiv.levelZero** {π : Representation ℂ (GL n k) V} {σ : Representation ℂ (GL n k) W} (h : InertEquiv π σ) : Nonempty ((levelZero σ).Equiv (levelZero π))

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsAlgebraMapEqRoot

theorem **Ezpow_mem_A** {i : ℤ} (hi : 0 ≤ i) : Ezpow K l e i ∈ A O K

theorem **rankOneDesc_act** (j : ℤ) (χ : TameGal K l e a →* O) (g : TameGal K l e a) (v : Vec K a Unit) (τ : ZMod a) : (rankOneDesc j χ).act g v τ () = C (algebraMap O K (χ g)) * rescale (g.root τ) (v (τ - g.shift) ())

theorem **exists_EN_bas_rankOne** (j : ℤ) : ∃ γ ∈ S O K e, (rankOne O K l e fun _ : ZMod a => j).EN (bas ()) = (E K l e * γ - (j : K⟦X⟧) * N K (E K l e)) • bas ()

theorem **exists_conn_eq** {m : Vec K a ι} (hm : m ∈ D.MM) : ∃ w ∈ D.MM, D.conn m = (X : K⟦X⟧) • w

theorem **conn_mem_MM** {m : Vec K a ι} (hm : m ∈ D.MM) : D.conn m ∈ D.MM

theorem **EN_mem_Man** {m : Vec K a ι} (hm : m ∈ D.Man) : D.EN m ∈ D.Man

theorem **EN_mem_divE** {n : ℕ} {y : Vec K a ι} (hy : y ∈ divE l e n D.Man) : D.EN y ∈ divE l e n D.Man

theorem **conn_bas_mem** (i : ι) : D.conn (bas i) ∈ D.Mext (RpiE O K l e)

theorem **hatTot_conn** [CompleteSpace K] {v : Vec K a ι} (hv : v ∈ D.Mext (RpiE O K l e)) : D.hatTot (D.conn v) = D.connHat (D.hatTot v)

theorem **smul_mem_Mext** {B : Subalgebra O K⟦X⟧} (hB : A O K ≤ B) {m : Vec K a ι} (hm : m ∈ D.ext B) : E K l e ^ η • m ∈ D.Mext B

@[simp] theorem **Lc_cls** (v : Vec K a ι) : D.Lc (D.cls v) = D.cls (D.EN v)

theorem **Lc_mem_PolEta** {n : ℕ} {w : D.Pol} (hw : w ∈ D.PolEta n) : D.Lc w ∈ D.PolEta n

theorem **actPol_Lc** (g : TameGal K l e a) (w : D.Pol) : D.desc.actPol g (D.Lc w) = D.Lc (D.desc.actPol g w)

theorem **Lc_mem_Zl** {w : D.Pol} (hw : w ∈ D.Zl) : D.Lc w ∈ D.Zl

@[simp] theorem **coe_LcZ** (w : D.Zl) : (D.LcZ w : D.Pol) = D.Lc w

theorem **Mon_eq** (y : Vec K a ι) : D.Mon y = D.Lc (D.cls y) - D.cls (E K l e • D.zeta y)

theorem **Mon_add_phi_sub** (y : Vec K a ι) {m : Vec K a ι} (hm : m ∈ D.Man) : D.Mon (y + (D.phi m - m)) = D.Mon y

theorem **Mon_mem_Zl_of_zeta_mem** {y : Vec K a ι} (hy : y ∈ D.Yl) (hz : D.zeta y ∈ D.ext (Rpi O K l e)) : D.Mon y ∈ D.Zl

theorem **_root_.GlobalTheory.BK.BKHom.onPol_Lc** {D : BKDat O K l e a ι η H} {D' : BKDat O K l e a κ η' H'} (F : BKHom D D') (w : D.Pol) : F.onPol (D.Lc w) = D'.Lc (F.onPol w)

theorem **mem_invE_of_mem_Y** (hy : y ∈ D.Y) : y ∈ D.Mext (invE l e (A O K))

theorem **plusMod_upper** (hy : y ∈ D.Y) : ∀ v ∈ (plusMod hy).L, E K l e ^ η • v ∈ (plusMod hy).M

theorem **plusMod_lower** (hy : y ∈ D.Y) : ∀ v ∈ (plusMod hy).M, E K l e ^ (H + η) • v ∈ (plusMod hy).L

theorem **norm_card** : ‖(Nat.card (TameGal K l e a) : K)‖ = 1

theorem **avg_mem_inv** (v : Vec K a ι) : Δ.avg v ∈ Δ.inv

theorem **map_avg** {V : Type} [AddCommGroup V] [Module K V] (f : Vec K a ι →ₗ[K] V) {v : Vec K a ι} (h : ∀ g, f (Δ.act g v) = f v) : f (Δ.avg v) = f v

theorem **avg_of_mem_inv** {v : Vec K a ι} (hv : v ∈ Δ.inv) : Δ.avg v = v

theorem **avg_mem** (W : Submodule O (Vec K a ι)) {v : Vec K a ι} (h : ∀ g, Δ.act g v ∈ W) : Δ.avg v ∈ W

theorem **map_Y** (η : ℕ) : (Δ.Y η).map (P.cls.restrictScalars O) = Δ.Z η

theorem **Ezpow_mem_invL_A_iff** {i : ℤ} : Ezpow K l e i ∈ invL l (A O K) ↔ 0 ≤ i

theorem **mem_shiftE_of_le** {P : PhiMod O K l e a ι} {i k : ℤ} (h : i ≤ k) {v : Vec K a ι} (hv : v ∈ shiftE l e k (P.Mext (invL l (A O K)))) : v ∈ shiftE l e i (P.Mext (invL l (A O K)))

theorem **filL_rankOne** (j i : ℤ) : (rankOne O K l e fun _ : ZMod a => j).filL i = shiftE l e (if i ≤ j then j else j + 1) ((rankOne O K l e fun _ : ZMod a => j).Mext (invL l (A O K)))

theorem **jumps_rankOne** (j : ℤ) : (rankOne O K l e fun _ : ZMod a => j).jumps = {j}

theorem **Lc_rankOneDat** (χ : TameGal K l e a →* O) : (rankOneDat O K l e a η H j h₁ h₂ χ).Lc ((rankOneDat O K l e a η H j h₁ h₂ χ).cls (Ezpow K l e (-1) • bas ())) = (((j + 1) * e * l : ℤ) : K) • (rankOneDat O K l e a η H j h₁ h₂ χ).cls (Ezpow K l e (-1) • bas ())

theorem **cls_mem_Z_rankOneDat** (hη : 1 ≤ η) : (rankOneDat O K l e a η H j h₁ h₂ 1).cls (Ezpow K l e (-1) • bas ()) ∈ (rankOneDat O K l e a η H j h₁ h₂ 1).Z ∧ (rankOneDat O K l e a η H j h₁ h₂ 1).cls (Ezpow K l e (-1) • bas ()) ≠ 0

theorem **isOfType_rankOneDat** [CompleteSpace K] {j : ℤ} (h₁ : -(η : ℤ) ≤ j) (h₂ : j ≤ H) : (rankOneDat O K l e a η H j h₁ h₂ 1).IsOfType {0} {j} where

theorem **exists_algebraMap_eq_root** (g : TameGal K l e 1) : ∃ c : O, algebraMap O K c = g.root 0

theorem **algebraMap_omegaOne** (g : TameGal K l e 1) : algebraMap O K (omegaOne O g) = g.root 0

theorem **mem_chars_rankOneDesc** (j : ℤ) (c : ZMod e) : c ∈ (rankOneDesc (K

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsEqAlgebraMapAddPowMul

theorem **mem_A_of_E_mul_mem** {h : K⟦X⟧} (hh : h ∈ Rpi O K l e) (hE : E K l e * h ∈ A O K) : h ∈ A O K

theorem **mem_A_of_E_pow_mul_mem** (m : ℕ) {h : K⟦X⟧} (hh : h ∈ Rpi O K l e) (hE : E K l e ^ m * h ∈ A O K) : h ∈ A O K

theorem **algebraMap_AhatE_injective** : Function.Injective (algebraMap (Ahat O K l e) (AhatE O K l e))

theorem **hat_E_pow_mul** (n : ℕ) {f : K⟦X⟧} (hf : f ∈ RpiE O K l e) : hat O l e (E K l e ^ n * f) = algebraMap _ _ (Ehat O K l e ^ n) * hat O l e f

theorem **hat_sub** {f g : K⟦X⟧} (hf : f ∈ RpiE O K l e) (hg : g ∈ RpiE O K l e) : hat O l e (f - g) = hat O l e f - hat O l e g

theorem **hat_mem_range_iff** {f : K⟦X⟧} (hf : f ∈ RpiE O K l e) : (∃ x : Ahat O K l e, algebraMap _ _ x = hat O l e f) ↔ f ∈ Rpi O K l e

theorem **exists_eq_algebraMap_add_pow_mul** {B : Type} [CommRing B] (s : B) (n : ℕ) (x : AdicCompletion (Ideal.span {s}) B) : ∃ (g : B) (x' : AdicCompletion (Ideal.span {s}) B), x = algebraMap B _ g + algebraMap B _ s ^ n * x'

theorem **toHat_derivative** (f : Rpi O K l e) : toHat O K l e (_, derivative_mem_R l f.2) = dHat O K l e (toHat O K l e f)

theorem **N_mem_RpiE** {f : K⟦X⟧} (hf : f ∈ RpiE O K l e) : N K f ∈ RpiE O K l e

theorem **hat_N_of_mem_Rpi** {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) : hat O l e (N K f) = NHatE O K l e (hat O l e f)

theorem **hat_N** {f : K⟦X⟧} (hf : f ∈ RpiE O K l e) : hat O l e (N K f) = NHatE O K l e (hat O l e f)

@[simp] theorem **Ezpow_natCast** (n : ℕ) : Ezpow K l e n = E K l e ^ n

theorem **Ezpow_add** (i j : ℤ) : Ezpow K l e (i + j) = Ezpow K l e i * Ezpow K l e j

theorem **E_pow_mul_Ezpow_neg** (n : ℕ) : E K l e ^ n * Ezpow K l e (-n) = 1

theorem **mem_divE** {n : ℕ} {W : Submodule R (Vec K a ι)} {y : Vec K a ι} : y ∈ divE l e n W ↔ E K l e ^ n • y ∈ W

theorem **mem_shiftE** {i : ℤ} {W : Submodule R (Vec K a ι)} {y : Vec K a ι} : y ∈ shiftE l e i W ↔ Ezpow K l e (-i) • y ∈ W

@[simp] theorem **mulV_apply** (f : K⟦X⟧) (v : Vec K a ι) : mulV K a ι f v = f • v

theorem **mem_Mext_invE_of_mem_divE** {B : Subalgebra O K⟦X⟧} [HasU B] {n : ℕ} {y : Vec K a ι} (hy : E K l e ^ n • y ∈ P.Mext B) : y ∈ P.Mext (invE l e B)

theorem **mem_Man** {v : Vec K a ι} : v ∈ P.Man ↔ v ∈ P.Mext (Rpi O K l e)

theorem **smul_mem_Man** {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) {v : Vec K a ι} (hv : v ∈ P.Man) : f • v ∈ P.Man

theorem **cls_eq_zero_iff** {v : Vec K a ι} : P.cls v = 0 ↔ v ∈ P.Man

@[simp] theorem **mulPol_cls** (f : Rpi O K l e) (v : Vec K a ι) : P.mulPol f (P.cls v) = P.cls ((f : K⟦X⟧) • v)

@[simp] theorem **vPol_cls** (v : Vec K a ι) : P.vPol (P.cls v) = P.cls ((X : K⟦X⟧) ^ e • v)

theorem **EN_apply** (v : Vec K a ι) : P.EN v = E K l e • P.conn v

theorem **Mon_apply** (y : Vec K a ι) : P.Mon y = P.cls (P.EN y - E K l e • P.zeta y)

theorem **eisenstein_x_eq_neg** {y : Vec K a ι} (hy : y ∈ P.Mext (invE l e (A O K))) : (P.plus hy).EN xVec = -(P.plusInl hy).app (P.EN y - E K l e • P.zeta y)

theorem **Mon_eq_zero_iff** {y : Vec K a ι} (hy : y ∈ P.Mext (invE l e (A O K))) : P.Mon y = 0 ↔ (P.plus hy).EN xVec ∈ (P.plus hy).Man

theorem **Mon_phi_sub** (m : Vec K a ι) : P.Mon (P.phi m - m) = -P.cls (P.EN m)

theorem **E_pow_smul_EN** (n : ℕ) (y : Vec K a ι) : E K l e ^ n • P.EN y = P.EN (E K l e ^ n • y) - ((n : K⟦X⟧) * N K (E K l e)) • E K l e ^ n • y

theorem **M_le_Man** {v : Vec K a ι} (hv : v ∈ P.M) : v ∈ P.Man

theorem **cls_mem_PolEta** {η : ℕ} {y : Vec K a ι} (hy : y ∈ P.EM η) : P.cls y ∈ P.PolEta η

theorem **mem_M_of_mem_Man** {η : ℕ} {y : Vec K a ι} (hy : y ∈ P.EM η) (h : y ∈ P.Man) : y ∈ P.M

theorem **mem_Mhat** {w : ZMod a → ι → AhatE O K l e} : w ∈ P.Mhat ↔ ∀ τ i, ∃ x : Ahat O K l e, algebraMap _ _ x = w τ i

theorem **hatTot_mem_Mhat_iff** {v : Vec K a ι} (hv : v ∈ P.Mext (RpiE O K l e)) : P.hatTot v ∈ P.Mhat ↔ v ∈ P.Man

theorem **hatTot_sub_mem_Mhat_iff** {v w : Vec K a ι} (hv : v ∈ P.Mext (RpiE O K l e)) (hw : w ∈ P.Mext (RpiE O K l e)) : P.hatTot v - P.hatTot w ∈ P.Mhat ↔ P.cls v = P.cls w

theorem **smul_hatTot_mem_Mhat_iff** (η : ℕ) {v : Vec K a ι} (hv : v ∈ P.Mext (RpiE O K l e)) : Ehat O K l e ^ η • P.hatTot v ∈ P.Mhat ↔ v ∈ divE l e η P.Man

theorem **exists_sub_hatTot_mem_Mhat** (η : ℕ) {w : ZMod a → ι → AhatE O K l e} (hw : Ehat O K l e ^ η • w ∈ P.Mhat) : ∃ v ∈ divE l e η (P.Mext (invL l (A O K))), w - P.hatTot v ∈ P.Mhat

theorem **conn_apply_eq** (v : Vec K a ι) (τ : ZMod a) (i : ι) : P.conn v τ i = N K (v τ i) + ∑ j, v τ j * P.conn (bas j) τ i

theorem **hatTot_conn** (hN : ∀ j, P.conn (bas j) ∈ P.Mext (RpiE O K l e)) {v : Vec K a ι} (hv : v ∈ P.Mext (RpiE O K l e)) : P.hatTot (P.conn v) = P.connHat (P.hatTot v)

@[simp] theorem **onPol_cls** (v : Vec K a ι) : F.onPol (P.cls v) = Q.cls (F.app v)

theorem **app_EN** (v : Vec K a ι) : F.app (P.EN v) = Q.EN (F.app v)

theorem **onPol_Mon** (y : Vec K a ι) : F.onPol (P.Mon y) = Q.Mon (F.app y)

theorem **app_mem_EM** {η : ℕ} {y : Vec K a ι} (hy : y ∈ P.EM η) : F.app y ∈ Q.EM η

theorem **onPol_mem_PolEta** {η : ℕ} {w : P.Pol} (hw : w ∈ P.PolEta η) : F.onPol w ∈ Q.PolEta η

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsEqMulAddOfIsAdicComplete

theorem **exists_eq_mul_add_of_isAdicComplete** {R : Type} [CommRing R] (t : R) [IsAdicComplete (Ideal.span {t}) R] (T : R →+ R) (hT : ∀ x, T (t * x) = t * T x) (w : R) : ∃ x, x = t * T x + w

theorem **pow_mul_factorial_dvd** (l k : ℕ) (hl : 0 < l) : l ^ k * k.factorial ∣ (l * k).factorial

theorem **exists_mul_frob_E** : ∃ c ∈ S O K e, c * frob K l (E K l e) = l

theorem **E_mul_N_Ezpow** (j : ℤ) : E K l e * N K (Ezpow K l e j) = (j : K⟦X⟧) * N K (E K l e) * Ezpow K l e j

theorem **map_Ezpow** (f : K⟦X⟧ →+* K⟦X⟧) (hf : f (E K l e) = E K l e) (j : ℤ) : f (Ezpow K l e j) = Ezpow K l e j

theorem **Ezpow_neg_mul** (j : ℤ) : Ezpow K l e (-j) * Ezpow K l e j = 1

theorem **mx_rankOne_eq** (j : ZMod a → ℤ) (τ : ZMod a) (i k : Unit) : (rankOne O K l e j).mx τ i k = Ezpow K l e (j τ)

theorem **mxInv_rankOne_eq** (j : ZMod a → ℤ) (τ : ZMod a) (i k : Unit) : (rankOne O K l e j).mxInv τ i k = Ezpow K l e (-j τ)

theorem **phi_bas_rankOne** (j : ℤ) : (rankOne O K l e fun _ : ZMod a => j).phi (bas ()) = Ezpow K l e j • bas ()

theorem **mem_ext_rankOne** (j : ℤ) {B : Subalgebra O K⟦X⟧} {v : Vec K a Unit} : v ∈ (rankOne O K l e fun _ : ZMod a => j).ext B ↔ ∀ τ, Ezpow K l e (-j) * v τ () ∈ B

theorem **exists_nabla_rankOne** (j : ℤ) : ∃ γ ∈ S O K e, (rankOne O K l e fun _ : ZMod a => j).nabla (Ezpow K l e j • bas ()) = γ • Ezpow K l e j • bas ()

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsGenIsUnitSub

theorem **idem_mul_self** (τ : ZMod a) : idem c τ * idem c τ = idem c τ

theorem **idem_mul_of_ne** {τ σ : ZMod a} (h : τ ≠ σ) : idem c τ * idem c σ = 0

theorem **sum_idem** : ∑ τ, idem c τ = 1

theorem **idem_mul_sum** (x : ZMod a → R) (τ : ZMod a) : idem c τ * ∑ σ, idem c σ * x σ = idem c τ * x τ

theorem **eq_sum_idem** (x : ZMod a → O) : c x = ∑ τ, idem c τ * algebraMap O R (x τ)

theorem **glue_apply** (x : ZMod a → B) : glue c f x = ∑ τ, idem c τ * f (x τ)

theorem **glue_const** (b : B) : glue c f (fun _ => b) = f b

theorem **idem_mul_glue** (x : ZMod a → B) (τ : ZMod a) : idem c τ * glue c f x = idem c τ * f (x τ)

theorem **map_glue** {B' R' : Type} [CommRing B'] [CommRing R'] [Algebra O B'] [Algebra O R'] {c' : (ZMod a → O) →ₐ[O] R'} {f' : B' →ₐ[O] R'} (s : R →+* R') (k : ZMod a) (hs : ∀ τ, s (idem c τ) = idem c' (τ + k)) {x : ZMod a → B} {y : ZMod a → B'} (h : ∀ τ, s (f (x (τ − k))) = f' (y τ)) : s (glue c f x) = glue c' f' y

theorem **tensor_hom_ext** {Z W J : Type} [CommRing Z] [CommRing W] [Algebra Z O] [Algebra Z W] {v : J → W} (hv : Submodule.span Z (Set.range v) = ⊤) (f g : O ⊗[Z] W →ₐ[O] R) (h : ∀ i, f (1 ⊗ₜ v i) = g (1 ⊗ₜ v i)) : f = g

theorem **bijective_lift** {Z W J I : Type} [CommRing Z] [CommRing W] [Algebra Z O] [Algebra Z W] [Fintype I] (emb : I → W →ₐ[Z] O) {v : J → W} (hv : Submodule.span Z (Set.range v) = ⊤) (t : W) (ht : ∀ j, v j = 0 ∨ ∃ k : ℕ, v j = t ^ k) (hu : ∀ i j, i ≠ j → IsUnit (emb i t − emb j t)) (f : Polynomial Z) (hfm : f.Monic) (hfd : f.natDegree ≤ Fintype.card I) (h0 : Polynomial.aeval t f = 0) (hroot : ∀ i, Polynomial.aeval (emb i t) f = 0) : Function.Bijective (Algebra.TensorProduct.lift (Algebra.ofId O (I → O)) (AlgHom.pi emb) fun _ _ => Commute.all _ _)

theorem **sigmaNeg_zero** : sigmaNeg l a 0 = 1

theorem **sigmaNeg_mul** (τ : ZMod a) : sigmaNeg l a τ * sigma l a = sigmaNeg l a (τ − 1)

theorem **mul_sigmaNeg** (τ : ZMod a) : sigma l a * sigmaNeg l a τ = sigmaNeg l a (τ − 1)

theorem **exists_monic** (t : Wq l a) : ∃ f : Polynomial ℤ_[l], f.Monic ∧ f.natDegree = a ∧ ∀ τ, Polynomial.aeval (sigmaNeg l a τ t) f = 0

theorem **exists_gen_isUnit_sub** : ∃ z : Fq l a, (∀ x : Fq l a, x ≠ 0 → ∃ k : ℕ, x = z ^ k) ∧ ∀ τ σ : ZMod a, τ ≠ σ → IsUnit (sigmaNeg l a τ (WittVector.teichmuller l z) − sigmaNeg l a σ (WittVector.teichmuller l z))

theorem **emb_zero** (w : Wq l a) : emb t 0 w = t w

theorem **emb_sigma** (τ : ZMod a) (w : Wq l a) : emb t τ (sigma l a w) = emb t (τ − 1) w

theorem **splitHom_tmul** (o : O) (w : Wq l a) (τ : ZMod a) : splitHom t (o ⊗ₜ w) τ = o * emb t τ w

theorem **splitHom_bijective** : Function.Bijective (splitHom t)

theorem **frobOA_tmul** (o : O) (w : Ainf l) : frobOA l O (o ⊗ₜ w) = o ⊗ₜ WittVector.frobenius w

theorem **galOA_tmul** (g : GQ l) (o : O) (w : Ainf l) : galOA O g (o ⊗ₜ w) = o ⊗ₜ Per.σA g w

theorem **galOW_tmul** (g : GQ l) (o : O) (w : WCb l) : galOW O g (o ⊗ₜ w) = o ⊗ₜ galW l g w

theorem **inclO_tmul** (o : O) (w : Ainf l) : inclO l O (o ⊗ₜ w) = o ⊗ₜ inclW l w

theorem **frobOW_tmul** (o : O) (w : WCb l) : frobOW l O (o ⊗ₜ w) = o ⊗ₜ frobW l w

theorem **galOW_one** (x : OW l O) : galOW O 1 x = x

theorem **galOW_mul** (g h : GQ l) (x : OW l O) : galOW O (g * h) x = galOW O g (galOW O h x)

theorem **frobOW_inclO** (x : OA l O) : frobOW l O (inclO l O x) = inclO l O (frobOA l O x)

theorem **galOW_inclO** (g : GQ l) (x : OA l O) : galOW O g (inclO l O x) = inclO l O (galOA O g x)

theorem **galOW_frobOW** (g : GQ l) (x : OW l O) : galOW O g (frobOW l O x) = frobOW l O (galOW O g x)

theorem **inclO_injective** [Module.Free ℤ_[l] O] : Function.Injective (inclO l O)

theorem **constA_splitHom** (x : O ⊗[ℤ_[l]] Wq l a) : constA t (splitHom t x) = Algebra.TensorProduct.map (AlgHom.id O _) (wA l a) x

theorem **constA_emb** (w : Wq l a) : constA t (fun τ => emb t τ w) = 1 ⊗ₜ wA l a w

theorem **frobOA_constA** (x : ZMod a → O) : frobOA l O (constA t x) = constA t fun τ => x (τ − 1)

theorem **frobOA_idem** (τ : ZMod a) : frobOA l O (idem (constA t) τ) = idem (constA t) (τ + 1)

theorem **inclO_idem** (τ : ZMod a) : inclO l O (idem (constA t) τ) = idem (constW t) τ

theorem **frobOW_idem** (τ : ZMod a) : frobOW l O (idem (constW t) τ) = idem (constW t) (τ + 1)

theorem **galOA_constA** {g : GQ l} (hg : ∀ x ∈ Kunr l a, g x = x) (x : ZMod a → O) : galOA O g (constA t x) = constA t x

theorem **coe_frobAc** (f : ZMod a → A O K) (τ : ZMod a) : (frobAc l f τ : K⟦X⟧) = frobTot _ a l (fun σ => (f σ : K⟦X⟧)) τ

theorem **coeff_toAc** (x : (Wq l a)⟦X⟧) (τ : ZMod a) (n : ℕ) : coeff n (toAc t K x τ : K⟦X⟧) = algebraMap O K (emb t τ (coeff n x))

theorem **exists_eq_sum_toAc** {J : Type} [Fintype J] (o : Module.Basis J ℤ_[l] O) (f : ZMod a → A O K) : ∃ x : J → (Wq l a)⟦X⟧, f = ∑ j, o j • toAc t K (x j)

theorem **complete_teichR** : IsAdicComplete (Ideal.span {ϖ.teichR O}) (OA l O)

theorem **sep_teichR** (x : OA l O) (h : ∀ n, ϖ.teichR O ^ n ∣ x) : x = 0

theorem **exists_evO** : ∃ f : O⟦X⟧ →ₐ[O] OA l O, f X = ϖ.teichR O

theorem **evO_X** : ϖ.evO O X = ϖ.teichR O

theorem **iA_equivA** (f : O⟦X⟧) : ϖ.iA O K (equivA _ _ f) = ϖ.evO O f

theorem **iA_u** : ϖ.iA O K (uIn _) = ϖ.teichR O

theorem **iA_E** : ϖ.iA O K (EIn l e _) = 1 ⊗ₜ ϖ.ev a (eis l a e)

theorem **iTot_u** : ϖ.iTot t K (fun _ => uIn _) = 1 ⊗ₜ ϖ.teich

theorem **iTot_E** : ϖ.iTot t K (fun _ => EIn l e _) = 1 ⊗ₜ ϖ.ev a (eis l a e)

theorem **iTot_toAc** (x : (Wq l a)⟦X⟧) : ϖ.iTot t K (toAc t K x) = 1 ⊗ₜ ϖ.ev a x

theorem **iA_frob** (f : A O K) : ϖ.iA O K ⟨frob _ l f, frob_mem_A l f.2⟩ = frobOA l O (ϖ.iA O K f)

theorem **iTot_frob** (f : ZMod a → A O K) : ϖ.iTot t K (frobAc l f) = frobOA l O (ϖ.iTot t K f)

theorem **galOA_teichR** (g : GQ l) : galOA O g (ϖ.teichR O) = (1 + 1 ⊗ₜ ϖ.mu g) * ϖ.teichR O

theorem **galOA_iTot** {g : GQ l} (hg : g ∈ ϖ.GfieldInf a) (f : ZMod a → A O K) : galOA O g (ϖ.iTot t K f) = ϖ.iTot t K f

theorem **isUnit_inclO_teichR** : IsUnit (inclO l O (ϖ.teichR O))

theorem **isUnit_inclO_iA_E** : IsUnit (inclO l O (ϖ.iA O K (EIn l e _)))

theorem **iE_of_mem** {f : K⟦X⟧} (h : f ∈ A O K) (h' : f ∈ invE l e (A O K)) : ϖ.iE O K ⟨f, h'⟩ = inclO l O (ϖ.iA O K ⟨f, h⟩)

theorem **iTot_injective** : Function.Injective (ϖ.iTot t K)

theorem **iW_of_mem** (f : ZMod a → A O K) : ϖ.iW t K (fun τ => ⟨f τ, le_away _ _ (f τ).2⟩) = inclO l O (ϖ.iTot t K f)

theorem **iW_injective** : Function.Injective (ϖ.iW t K)

theorem **galOW_iW** {g : GQ l} (hg : g ∈ ϖ.GfieldInf a) (f : ZMod a → invE l e (A O K)) : galOW O g (ϖ.iW t K f) = ϖ.iW t K f

theorem **intToOC_injective** (LL : FinExt l) : Function.Injective (intToOC LL)

theorem **intToOC_algebraMap** (LL : FinExt l) (z : ℤ_[l]) : intToOC LL (algebraMap ℤ_[l] ɪ(OL LL) z) = Per.θ l (algebraMap ℤ_[l] (Ainf l) z)

theorem **exists_intToOC_eq** (w : Wq l a) : ∃ y, intToOC LL y = Per.θ l (wA l a w)

theorem **intToOC_thetaW** (w : Wq l a) : intToOC LL (thetaW a LL w) = Per.θ l (wA l a w)

theorem **inclO_eq** : inclO l ɪ(OL LL) = inclR LL

theorem **frobOW_eq** : frobOW l ɪ(OL LL) = φRW LL

theorem **galOW_eq** {P : LocPrime l} (AI : LocAinf l P) (g : GQ l) : galOW ɪ(OL LL) g = AI.gRW LL g

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsIsClassSystem

theorem **Frame.Good.posDef** {F : Frame S T} (h : F.Good) (τ : D.what₂.Tau) : (gramAt D S τ).PosDef

theorem **exists_isClassSystem** (B : Base S T) (hF : B.F.Good) (ψ : Twist D Circle) : ∃ (m : ℕ) (h : Fin m → GA D S), IsClassSystem S (B.levJ ψ) h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsIsScalar

theorem **evW_smul** (j : Fin D.R.m) (c : D.what₂.LL) (x : S.WL ((j : ℕ) + 1)) : evW YS v₀ j (c • x) = D.what₂.embLL c • evW YS v₀ j x

theorem **evK_tmul** (a : NX.KB S i b) (f : NX.FmK S ((j : ℕ) + 1)) : evK YS v₀ hp i j b (a ⊗ₜ f) = ρ.e i b a • evW YS v₀ j (Subtype.val f)

theorem **realRd_tmul** (ν : ℂ) (a : NX.KB S i b) (f : NX.FmK S ((j : ℕ) + 1)) : realRd YS v₀ hp i j b ν ((YS.eF i _ b (Nat.le_add_left _ _) j.2).symm (a ⊗ₜ[S.Ks] f : NX.KB S i b ⊗[S.Ks] _)) = (ν⁻¹ * ρ.e i b a) • evW YS v₀ j (Subtype.val f)

theorem **realRd_apply** (ν : ℂ) (f : D.nexp.Fm i ((j : ℕ) + 1) b) : realRd YS v₀ hp i j b ν f = ν⁻¹ • realRd YS v₀ hp i j b 1 f

theorem **heckeAt_realRd** {ρ : Setup D.ntw} (hp : ρ.Good) (i : D.ntw.Tw) (j : Fin D.R.m) (b : Bool) {ν : ℂ} (hν : ν ≠ 0) (k P f g) : HeckeAt (realRd YS v₀ hp i j b ν) k P f g ↔ HeckeAt (realRd YS v₀ hp i j b 1) k P f g

theorem **exists_isScalar** {ρ : Setup D.ntw} (hp : ρ.Good) (i : D.ntw.Tw) (j : Fin D.R.m) (b : Bool) : ∃ ν, IsScalar YS v₀ hp i j b ν

theorem **heckeRead** {i j b} (h : RatHecke (Z.rd i j b)) : HeckeRead (Z.rd i j b)

theorem **delU_ine** (i : D.ntw.Tw) (Q : Finset (Pl D.host.E)) (M : ℕ) (hQ : Formalism.MainNotions.NAdm.AdmQ D.ntw i Q M) (P : Pl D.host.E) (hP : P ∈ Q) : (∀ A : ValuationSubring D.nd.Om, IsPlaceOver A P → Function.Surjective fun σ : Ine (F

theorem **delUTie** (hK : GlobalTheory.TwSys.K1 D.ntw) : GlobalTheory.TwSys.DelUTie D.nexp

theorem **rat** {i j b} (h : RatRow YS v₀ Z.good i j b) : Rat Z.good (Z.rd i j b)

theorem **thU_empty** (i : D.ntw.Tw) (j : ℕ) (b : Bool) (M : ℕ) (d : D.nexp.Dat i j b) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) : D.nexp.thU i j b ∅ M d = D.nexp.th i j b d ⊗ₜ[D.ntw.O i] 1

theorem **exists_dat** {i j b} (h : RatRow YS v₀ Z.good i j b) {s g P} (hd : Z.ρ.IsDatum i j b s g P) : ∃ d, (Z.tie i j b).cls d = g ∧ (Z.tie i j b).coef d = P ∧ Z.rd i j b (D.nexp.th i _ b d) = Z.ρ.theta i j b g P

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsIterateDerivative

theorem **PiSys.isGalois_field** (he : e ∣ l ^ a - 1) : IsGalois ℚ_[l] ɪ(ϖ.field a).1

theorem **frobMat_mul_inv** : frobMat t ϖ P * frobMatInv t ϖ P = 1

theorem **frobMatInv_mul** : frobMatInv t ϖ P * frobMat t ϖ P = 1

theorem **phiW_apply** (v : ι → OW l O) : phiW t ϖ P v = frobMat t ϖ P *ᵥ fun i => frobOW l O (v i)

theorem **phiW_smul** (r : OW l O) (v : ι → OW l O) : phiW t ϖ P (r • v) = frobOW l O r • phiW t ϖ P v

theorem **mem_Treal** {v : ι → OW l O} : v ∈ Treal t ϖ P ↔ phiW t ϖ P v = v

theorem **galCoef_phiW** {g : GQ l} (hg : g ∈ ϖ.GfieldInf a) (v : ι → OW l O) : galCoef O ι g (phiW t ϖ P v) = phiW t ϖ P (galCoef O ι g v)

theorem **galCoef_mem_Treal** {g : GQ l} (hg : g ∈ ϖ.GfieldInf a) {v : ι → OW l O} (hv : v ∈ Treal t ϖ P) : galCoef O ι g v ∈ Treal t ϖ P

theorem **mem_ME_of_mem_M** {v : Vec K a ι} (hv : v ∈ P.M) : v ∈ ME P

theorem **mem_ME_of_mem_L** {v : Vec K a ι} (hv : v ∈ P.L) : v ∈ ME P

theorem **toW_apply** (v : ME P) (i : ι) : toW t ϖ P v i = ϖ.iW t K fun τ => ⟨v.1 τ i, mem_tot.1 v.2 τ i⟩

theorem **toW_injective** : Function.Injective (toW t ϖ P)

theorem **toW_smul** (f : ZMod a → invE l e (A O K)) (v w : ME P) (h : ∀ τ, w.1 τ = (f τ : K⟦X⟧) • v.1 τ) : toW t ϖ P w = ϖ.iW t K f • toW t ϖ P v

theorem **toW_of_mx** (v : ME P) (m : ι → ZMod a → A O K) (h : ∀ τ, v.1 τ = P.mx τ *ᵥ fun j => m j τ) : toW t ϖ P v = frobMat t ϖ P *ᵥ fun j => inclO l O (ϖ.iTot t K (m j))

theorem **toW_phi** {v : Vec K a ι} (hv : v ∈ P.M) : toW t ϖ P (P.phi v, mem_ME_of_mem_L (P.phi_mem_L hv)) = phiW t ϖ P (toW t ϖ P ⟨v, mem_ME_of_mem_M hv⟩)

theorem **galCoef_mem_MA** (g : GQ l) {v : ι → OW l O} (hv : v ∈ MA P) : galCoef O ι g v ∈ MA P

theorem **inclL_apply** (c : ι → OA l O) : inclL t ϖ P c = frobMat t ϖ P *ᵥ fun i => inclO l O (c i)

theorem **inclL_injective** : Function.Injective (inclL t ϖ P)

theorem **toW_mem_MA** {v : Vec K a ι} (hv : v ∈ P.M) : toW t ϖ P ⟨v, mem_ME_of_mem_M hv⟩ ∈ MA P

theorem **toW_mem_LA** {v : Vec K a ι} (hv : v ∈ P.L) : toW t ϖ P ⟨v, mem_ME_of_mem_L hv⟩ ∈ LA t ϖ P

theorem **mx_mulVec_coordL** (B : Subalgebra O K⟦X⟧) (v : P.ext B) (τ : ZMod a) : (P.mx τ *ᵥ fun i => (coordL P B v i τ : K⟦X⟧)) = v.1 τ

theorem **toCris_apply** (m : P.MM) (i : ι) : toCris t ϖ P m i = ϖ.iC t K (coordL P (S O K e) m i)

theorem **toBcris_apply** (m : P.MMl) (i : ι) : toBcris t ϖ P m i = ϖ.iB t K (coordL P (invL l (S O K e)) m i)

theorem **toW_toCris_of_mem_L** {v : Vec K a ι} (hv : v ∈ P.L) (hv' : v ∈ P.MM) : toW t ϖ P ⟨v, mem_ME_of_mem_L hv⟩ = inclL t ϖ P (fun i => ϖ.iTot t K (coordL P _ ⟨v, hv⟩ i)) ∧ toCris t ϖ P ⟨v, hv'⟩ = fun i => toOCr l O (ϖ.iTot t K (coordL P _ ⟨v, hv⟩ i))

theorem **homW_apply** (F : BK.Hom P Q) (v : ι → OW l O) : homW t ϖ F v = (homMat t ϖ F).map (inclO l O) *ᵥ v

theorem **homW_toW** (F : BK.Hom P Q) (v : ME P) : homW t ϖ F (toW t ϖ P v) = toW t ϖ Q ⟨F.app v, F.app_mem_Mext (le_away _ _) v.2⟩

theorem **homMat_mul_frobMat** (F : BK.Hom P Q) : (homMat t ϖ F).map (inclO l O) ∗ frobMat t ϖ P = frobMat t ϖ Q ∗ ((homMat t ϖ F).map (inclO l O)).map (frobOW l O)

theorem **homW_phiW** (F : BK.Hom P Q) (v : ι → OW l O) : homW t ϖ F (phiW t ϖ P v) = phiW t ϖ Q (homW t ϖ F v)

theorem **coe_homT** (F : BK.Hom P Q) (v : Treal t ϖ P) : (homT t ϖ F v : κ → OW l O) = homW t ϖ F v

theorem **homW_injective** : Function.Injective (homW t ϖ : BK.Hom P Q → _)

theorem **homW_comp** (G : BK.Hom Q R) (F : BK.Hom P Q) (v : ι → OW l O) : homW t ϖ (G.comp F) v = homW t ϖ G (homW t ϖ F v)

theorem **homW_scalar** (c : O) (v : ι → OW l O) : homW t ϖ (BK.Hom.scalar P c) v = c • v

theorem **frob_fixed** (T : LocLat G ℚ_[l] K O) : (∀ y : TW (OW l O) T, ∃ b, TWmap (frobOW l O) T b − b = y) ∧ (∀ b : TW (OW l O) T, TWmap (frobOW l O) T b = b ↔ ∃ x : ɪT.L.T, b = (1 : OW l O) ⊗ₜ[O] x) ∧ Function.Injective fun x : ɪT.L.T => (1 : OW l O) ⊗ₜ[O] x

theorem **j_ofT** (x : ɪT.L.T) : I.j (I.ofT x) = (1 : OW l O) ⊗ₜ[O] x

theorem **ofT_injective** : Function.Injective I.ofT

theorem **range_ofT** : LinearMap.range I.ofT = Treal t ϖ P

theorem **coe_equivT** (x : ɪT.L.T) : (I.equivT x : ι → OW l O) = I.ofT x

theorem **exists_phiW_sub** (y : ι → OW l O) : ∃ v, phiW t ϖ P v − v = y

theorem **j_act** (s : GQ l) (g : G) (v : ι → OW l O) : I.j (I.act s g v) = TWact (galOW O s) T g (I.j v)

theorem **act_ofT** (s : GQ l) (g : G) (x : ɪT.L.T) : I.act s g (I.ofT x) = I.ofT (T.actT g x)

theorem **act_smul** (s : GQ l) (g : G) (r : OW l O) (v : ι → OW l O) : I.act s g (r • v) = galOW O s r • I.act s g v

theorem **act_phiW** (s : GQ l) (g : G) (v : ι → OW l O) : I.act s g (phiW t ϖ P v) = phiW t ϖ P (I.act s g v)

theorem **act_mul** (s s' : GQ l) (g g' : G) (v : ι → OW l O) : I.act (s ∗ s') (g ∗ g') v = I.act s g (I.act s' g' v)

theorem **IsLMat.unique** {s : GQ l} {g : G} {Γ Γ' : Matrix ι ι (OA l O)} (h : I.IsLMat s g Γ) (h' : I.IsLMat s g Γ') : Γ = Γ'

theorem **exists_isLMat_iff** (s : GQ l) (g : G) : (∃ Γ, I.IsLMat s g Γ) ↔ ∀ v ∈ LA t ϖ P, I.act s g v ∈ LA t ϖ P

theorem **IsMapOf.unique** {P : PhiMod O K l e a ι} {Q : PhiMod O K l e a κ} {T T' : LocLat G ℚ_[l] K O} {t : Wq l a →ₐ[ℤ_[l]] O} {ϖ : PiSys l e} {I : Ident t ϖ P T} {I' : Ident t ϖ Q T'} {f : LatHom T.L T'.L} {F F' : BK.Hom P Q} (h : IsMapOf I I' f F) (h' : IsMapOf I I' f F') : F = F'

theorem **IsMapOf.homW_ofT** {P : PhiMod O K l e a ι} {Q : PhiMod O K l e a κ} {T T' : LocLat G ℚ_[l] K O} {t : Wq l a →ₐ[ℤ_[l]] O} {ϖ : PiSys l e} {I : Ident t ϖ P T} {I' : Ident t ϖ Q T'} {f : LatHom T.L T'.L} {F : BK.Hom P Q} (h : IsMapOf I I' f F) (x : ɪT.L.T) : homW t ϖ F (I.ofT x) = I'.ofT ⟨f.f x, f.T_le _ x.2⟩

theorem **kappa_tmul** (T : LocLat G ℚ_[l] K O) (w : WCb l) (x : ɪT.L.T) : letI

theorem **kappa_bijective** (T : LocLat G ℚ_[l] K O) : letI

theorem **Ident.card_eq** {G : Type} [Group G] [TopologicalSpace G] {t : Wq l a →ₐ[ℤ_[l]] O} {ϖ : PiSys l e} {P : PhiMod O K l e a ι} {T : LocLat G ℚ_[l] K O} (I : Ident t ϖ P T) : Fintype.card ι = Module.finrank O ɪT.L.T

theorem **map_smul** {T : Lat a O K ϖ} (hT : C T) (c : O) : F.map hT hT (latSmul T c) = BK.Hom.scalar _ c

theorem **map_id** {T : Lat a O K ϖ} (hT : C T) : F.map hT hT (latId T) = BK.Hom.scalar _ 1

theorem **map_comp** {T T' T'' : Lat a O K ϖ} (hT : C T) (hT' : C T') (hT'' : C T'') (g : LatHom T'.L T''.L) (f : LatHom T.L T'.L) : F.map hT hT'' (latComp g f) = (F.map hT' hT'' g).comp (F.map hT hT' f)

theorem **N_apply** (v : Vec K a ι) : D.N v = (X : K⟦X⟧) • D.nabla v

theorem **desc_zero** (v : Vec K a ι) : D.desc 0 v = v

theorem **desc_succ** (i : ℕ) (v : Vec K a ι) : D.desc (i + 1) v = D.desc i (D.N v − i • v)

theorem **N_mem** {m : Vec K a ι} (hm : m ∈ P.MM) : D.N m ∈ P.MM

theorem **desc_mem** (i : ℕ) {m : Vec K a ι} (hm : m ∈ P.MM) : D.desc i m ∈ P.MM

theorem **mx_unitMod** (τ : ZMod a) : (unitMod l a e O K ι).mx τ = 1

theorem **mxInv_unitMod** (τ : ZMod a) : (unitMod l a e O K ι).mxInv τ = 1

theorem **mem_ext_unitMod** {B : Subalgebra O K⟦X⟧} {v : Vec K a ι} : v ∈ (unitMod l a e O K ι).ext B ↔ ∀ τ i, v τ i ∈ B

theorem **frobMat_unitMod** : frobMat t ϖ (unitMod l a e O K ι) = 1

theorem **unitConn_frobCompat** : (unitConn (O

theorem **exists_iterate_derivative** {f : K⟦X⟧} (hf : f ∈ S O K e) (r : ℕ) : ∃ i₀ : ℕ, ∀ i ≥ i₀, ∃ h ∈ S O K e, (PowerSeries.derivative K)^[i] f = (l : K⟦X⟧) ^ r ∗ h

theorem **unitConn_topNilp** : (unitConn (O

theorem **unitConn_desc** (i : ℕ) (m : Vec K a ι) (τ : ZMod a) (j : ι) : (unitConn (O

theorem **unitIdent_j** (T : LocLat G ℚ_[l] K O) (b : Module.Basis ι O ɪT.L.T) (v : ι → OW l O) : (unitIdent t ϖ T b).j v = ∑ i, v i ⊗ₜ[O] b i

theorem **unitIdent_act** (T : LocLat G ℚ_[l] K O) (b : Module.Basis ι O ɪT.L.T) (s : GQ l) {g : G} (hg : ∀ x, T.ρ g x = x) (v : ι → OW l O) : (unitIdent t ϖ T b).act s g v = galCoef O ι s v

theorem **kappa_sum** (T : LocLat G ℚ_[l] K O) (b : Module.Basis ι O ɪT.L.T) {J : Type} [Fintype J] (o : J → O) (x : J × ι → (Wq l a)⟦X⟧) (m : Vec K a ι) (hm : m ∈ (unitMod l a e O K ι).M) (h : ∀ i, (fun τ => ⟨⟨m τ i, mem_tot.1 hm τ i⟩ : A O K⟩) = ∑ k, o k • toAc t K (x (k, i))) : letI

theorem **isBK_unitIdent** {T : Lat a O K ϖ} (hT : IsTriv T) (b : Module.Basis ι O ɪT.L.T) : IsBK (unitIdent t ϖ T b)

theorem **homMat_constHom** (Y : Matrix κ ι O) (i : κ) (j : ι) : (homMat t ϖ (constHom (K

theorem **isMapOf_constHom** {T T' : LocLat G ℚ_[l] K O} (b : Module.Basis ι O ɪT.L.T) (b' : Module.Basis κ O ɪT'.L.T) (f : LatHom T.L T'.L) : IsMapOf (unitIdent t ϖ T b) (unitIdent t ϖ T' b') f (constHom l a e (LinearMap.toMatrix b b' ((f.f.restrictScalars O).restrict f.T_le)))

theorem **isLMat_unitIdent** (T : LocLat G ℚ_[l] K O) (b : Module.Basis ι O ɪT.L.T) (s : GQ l) {g : G} (hg : ∀ x, T.ρ g x = x) : (unitIdent t ϖ T b).IsLMat s g 1

theorem **stable_unitIdent** {T : Lat a O K ϖ} (hT : IsTriv T) (b : Module.Basis ι O ɪT.L.T) : (unitIdent t ϖ T b).Stable

theorem **toCris_unitMod** (v : (unitMod l a e O K ι).MM) (j : ι) : toCris t ϖ _ v j = ∑ τ, idem (constC t) τ * ϖ.iS O K (v.1 τ j, mem_ext_unitMod.1 v.2 τ j)

theorem **galOCr_idem** (g : ɪ(GK (ϖ.field a))) (τ : ZMod a) : galOCr O g (idem (constC t) τ) = idem (constC t) τ

theorem **taylor_unitIdent** {T : Lat a O K ϖ} (hT : IsTriv T) (b : Module.Basis ι O ɪT.L.T) : (unitIdent t ϖ T b).Taylor (unitConn l a e ι)

theorem **natural_constHom** (Y : Matrix κ ι O) : (unitConn (K

theorem **isTriv_unitLat** : IsTriv (unitLat ɪ(GK (ϖ.field a)) : Lat a O K ϖ)

theorem **zlLat_eq** (T : LocLat G ℚ_[l] ɪLL.1 ɪ(OL LL)) : zlLat T = Formalism.MainNotions.LocalSituation.zlMod LL T

theorem **actZ_eq** (T : LocLat G ℚ_[l] ɪLL.1 ɪ(OL LL)) : actZ T = fun g => Formalism.MainNotions.LocalSituation.actTZ LL T g

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsLocExtFinrankEq

theorem **isSmooth_one** {L : Type} [Field L] [NumberField L] [Algebra L Ω] [IsAlgClosure L Ω] {w : HeightOneSpectrum (𝒪 L)} (h : IsPlaceOver A w) : GlobalTheory.WeilDeligne.IsSmooth (1 : Weil h →* ℂˣ)

theorem **exists_locExt_finrank_eq** (h : IsPlaceOver A v) (d : ℕ) (hd : 0 < d) : ∃ E : LocExt h, Module.finrank K E.L = d

theorem **BTerm.exists_of_data** (hc : Continuous ψ) (h1 : ψ ≠ 1) (E : LocExt h) (χ : (E.v'.adicCompletion E.L)ˣ →* ℂˣ) (hχ : IsSmoothChar χ) (n : ℤ) : ∃ t : BTerm h ψ, t.E = E ∧ HEq t.χ χ ∧ t.n = n ∧ t.ψ' ≠ 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsOfLayers

theorem **exists_of_layers** (P : LocPrime l) (AI : LocAinf l P) (h : ∀ (K : FinExt l) (s : ℕ) (LL : FinExt l), Good K s LL → Nonempty (Layers P AI K s LL)) (ht : ∀ K : FinExt l, P.ιB (AI.ιA AI.μ) - P.ιB P.tB ∈ (P.PD K).Fil 2) (hπ : ∀ K : FinExt l, ∃ u ∈ (P.PD K).Fil 0, P.ιB (AI.ιA AI.π₁) * u = 1) : ∃ H : LocHW l P, ∃ F : LocFon l P AI H, Nonempty (LocCmpT l P AI H F)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsOfLocalLanglands

theorem **q_pos** : (0 : ℝ) < resCard k

theorem **nabs_pos'** (x : kˣ) : 0 < nabs k x

theorem **cond_eq** {c c' : kˣ} (hc : HasConductor ψ c) (hc' : HasConductor ψ c') : nabs k c = nabs k c'

theorem **hasExponent_mul_absPow** (s : ℝ) {a : ℕ} : HasExponent (χ * absPow k s) a ↔ HasExponent χ a

theorem **sum_shift** {c : kˣ} {a : ℕ} {R : Finset kˣ} (hR : IsShellReps c a R) : (resCard k : ℂ) ^ (-(a : ℂ) / 2) * ∑ r ∈ R, ((((χ * absPow k ((-(1 / 2) : ℝ) : ℂ)) r)⁻¹ : ℂˣ) : ℂ) * (ψ (r : k) : ℂ) = (Real.sqrt (nabs k c) : ℂ) * ∑ r ∈ R, (((χ r)⁻¹ : ℂˣ) : ℂ) * (ψ (r : k) : ℂ)

theorem **isTateMeas_iff** {c : kˣ} (hc : HasConductor ψ c) {t : ℝ} (ht : 0 < t) {e : ℂ} : IsTateMeas ψ t χ (((t * Real.sqrt (nabs k c) : ℝ) : ℂ) * e) ↔ IsTateEps ψ (χ * absPow k ((-(1 / 2) : ℝ) : ℂ)) e

theorem **existsUnique_isTateMeas** (hψ : Continuous ψ) (h1 : ψ ≠ 1) (hχ : IsSmoothChar χ) {t : ℝ} (ht : 0 < t) : (∃! e : ℂ, IsTateMeas ψ t χ e) ∧ ∀ e, IsTateMeas ψ t χ e → e ≠ 0

theorem **isTateMeas_of_isUnram** {c : kˣ} (hc : HasConductor ψ c) (hχ : IsUnram χ) (t : ℝ) : IsTateMeas ψ t χ (((t * nabs k c : ℝ) : ℂ) * ((χ c)⁻¹ : ℂˣ))

theorem **condAbs_eq** {c : kˣ} (hc : HasConductor ψ c) : condAbs ψ = nabs k c

theorem **condAbs_pos** (ψ : AddChar k Circle) : 0 < condAbs ψ

theorem **lpTw_lpOfChar** {a : Weil h →* (v.adicCompletion K)ˣ} (ha : IsLocalArtin h a) {χ : (v.adicCompletion K)ˣ →* ℂˣ} (hχ : IsSmoothChar χ) (s : ℝ) : (WD.lpOfChar h (χ.comp a) (ha.smooth hχ)).lpTw s = WD.lpOfChar h ((χ * absPow _ s).comp a) (ha.smooth (hχ.mul (isSmoothChar_absPow _)))

theorem **induces_dim** {x : WD h'} {y : WD h} (H : Induces x y) : y.lpDim = Module.finrank K L * x.lpDim

theorem **induces_tw** {x : WD h'} {y : WD h} (H : Induces x y) (s : ℝ) : Induces (x.lpTw s) (y.lpTw s)

theorem **isDeligneMeas_measOf** {ε : EpsFamily A} (H : IsDeligneEps ε) : IsDeligneMeas (measOf ε)

theorem **isDeligneEps_centreOf** {ε : EpsMeasFamily A} (H : IsDeligneMeas ε) : IsDeligneEps (centreOf ε)

theorem **exists_iff** : (∃ ε : EpsMeasFamily A, IsDeligneMeas ε) ↔ ∃ ε : EpsFamily A, IsDeligneEps ε

theorem **exists_of_localLanglands** (hL : GlobalTheory.Published.LocalLanglandsGL) (E : Type) [Field E] [NumberField E] [Algebra E Ω] [IsAlgClosure E Ω] (w : HeightOneSpectrum (𝒪 E)) (h : IsPlaceOver A w) : (∃ a, IsLocalArtin h a) ∧ ∃ ε : EpsMeasFamily A, IsDeligneMeas ε

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsOpenSubgroup

theorem **exists_open_subgroup** {S : Set (Ine (F

theorem **IsTm.eq_one** {χ : Ine (F

theorem **IsTm.of_le** {χ χ' ψ : Ine (F

theorem **exists_isTm** {g : ℕ} [Fact g.Prime] (h : IsPlaceOver A w) (hw : normV w = g) {γ : G E Ω} (hγ : TameGenAt A γ) {m : ℕ} (hm : 0 < m) (hc : m.Coprime g) {z : ℂ} (hz : z ^ m = 1) : ∃ χ : Ine (F

theorem **exists_genFam** : ∃ γ, GenFam D γ

theorem **normV_eq** {k : Fin 2} {w : PlE D.host} (hw : w ∈ placesOver D.host.E (D.host.g k)) : normV w = D.host.g k

theorem **exists_order** {g : ℕ} [Fact g.Prime] {x : Fbar g} (hx : x ≠ 0) : ∃ m : ℕ, 0 < m ∧ m.Coprime g ∧ x ^ m = 1

theorem **exists_tameOm** (γ : Fin 2 → PlE D.host → GE D.nd) (hγ : GenFam D γ) : ∃ Ω : TameOm D, ∀ k : Fin 2, ∀ w ∈ placesOver D.host.E (D.host.g k), ∀ x, x ≠ 0 → Ω.om k w x (γ k w) = D.ntypes.teichC k x

theorem **nonempty_tameOm** : Nonempty (TameOm D)

theorem **ineRep_of_tie** : IneRep D

theorem **tameLP_iff** {w : PlE D.host} (L : D.naut.LP w) : TameLP D L ↔ (T.e w L).tame

theorem **labDepth_of_tie** {γ} (hγ : GenFam D γ) (h : LabAll T γ) : LabDepth D

theorem **labGen_of_tie** {γ} (hγ : GenFam D γ) (h : LabAll T γ) : LabGen D

theorem **exists_ties** (D : D7) (hE : IsEmpty D.nd.Loc) : ∃ D' : D7, Nonempty (GlobalTheory.RowTie.WDTie D'.nd D'.naut) ∧ IneRep D' ∧ LabDepth D' ∧ LabGen D'

theorem **heckeEig_eq_top** (i : D.nd.Loc) [Subsingleton (D.nd.X i)] (x : D.nd.X i) : heckeEig (D.nd.T i) (D.nd.act i) x = ⊤

theorem **nonempty_sec** (i : D.nd.Loc) (x : D.nd.X i) (hx : ∀ P, D.naut.ptV i P = some x) : Nonempty (Sec D i)

theorem **sec_ne_zero** {i : D.nd.Loc} (S : Sec D i) : S.Φ ≠ 0

theorem **nonempty_cenO** (S : ∀ i, Sec D i) (hpt : ∀ i P, D.naut.ptV i P ≠ none) (a : ∀ i, ↥(Subalgebra.center (D.nd.O i) (MonoidAlgebra (D.nd.O i) (G0g D.nd))) →ₐ[D.nd.O i] D.nd.O i) (ha : ∀ i z F, D.ntypes.onMt (D.nd.mu i) (D.nd.j i) (PairedCharacterCondition.mapCoef (iotaO D.nd D.naut i) ((z : ↥(Subalgebra.center (D.nd.O i) (MonoidAlgebra (D.nd.O i) (G0g D.nd)))) : MonoidAlgebra (D.nd.O i) (G0g D.nd))) F = iotaO D.nd D.naut i (a i z) • F) : Nonempty (CenO D)

theorem **mk_of** (P : PairedCharacterCondition.Pend D) : GlobalTheory.PendOf.mk P.lz P.eSer_eq P.eSer_LL P.zetaPrimitiveRoot (omOf P) P.inertiaCharpoly_eigenvalues P.inertiaCharpoly_character (cenOf P) P.zL_stable P.prMt_type P.Par_surj = P

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsPresOfRealD

theorem **φ_mem** (T : LocLat G Qp LL O) (x : ↥(DΔ W T)) : W.φD T x ∈ DΔ W T

theorem **γ_mem** (T : LocLat G Qp LL O) (g : W.Gam) (x : ↥(DΔ W T)) : W.γD T g x ∈ DΔ W T

theorem **Pres.j_φ'** (p : Pres W F ι τ γ) {T A q} (h : Onto T A q) (i : ZMod n) (x : ↥(DΔ W T)) : p.j T A q i ⟨W.φD T x, φ_mem W T x⟩ = (F.D A).φ i (p.j T A q (i - 1) x)

theorem **Pres.j_γ'** (p : Pres W F ι τ γ) {T A q} (h : Onto T A q) (i : ZMod n) (x : ↥(DΔ W T)) : p.j T A q i ⟨W.γD T γ x, γ_mem W T γ x⟩ = (F.D A).γ i (p.j T A q i x)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsRealisesNeOne

theorem **matches_one** : Matches (1 : G L Ω →* ℂˣ) (1 : HeckeCharacter L)

theorem **continuous_resG** (L : Type) [Field L] [Algebra L Ω] [Algebra k L] [IsScalarTower k L Ω] : Continuous (resG k L Ω)

theorem **exists_pow_eq_one** (χ : G L Ω →* ℂˣ) (op : IsOpen (χ.ker : Set (G L Ω))) : ∃ N : ℕ, 0 < N ∧ ∀ σ, χ σ ^ N = 1

theorem **exists_matches** (χ : G L Ω →* ℂˣ) (op : IsOpen (χ.ker : Set (G L Ω))) : ∃ X : HeckeCharacter L, Matches χ X ∧ ∃ N : ℕ, 0 < N ∧ X ^ N = 1

theorem **hk_spec** (χ : G L Ω →* ℂˣ) (op : IsOpen (χ.ker : Set (G L Ω))) : Matches χ (hk χ) ∧ ∃ N : ℕ, 0 < N ∧ hk χ ^ N = 1

theorem **resG_injective** : Function.Injective (resG k L Ω)

theorem **extendByZero_restrict** (χ : G L Ω →* ℂˣ) (τ : G L Ω) : ext0 k χ (resG k L Ω τ) = χ τ

theorem **gInd_twist** (χ : G L Ω →* ℂˣ) (ξ : G k Ω →* ℂˣ) (σ : G k Ω) : gInd k (χ * ξ.comp (resG k L Ω)) σ = ξ σ * gInd k χ σ

theorem **gInd_self** (χ : G L Ω →* ℂˣ) (σ : G L Ω) : gInd L χ σ = χ σ

theorem **trW_char** (h : IsPlaceOver A w) (ξ : Weil h →* ℂˣ) (hξ : GlobalTheory.WeilDeligne.IsSmooth ξ) (σ : Weil h) : trW (WD.lpOfChar h ξ hξ) σ = ξ σ

theorem **under_self** (u : HeightOneSpectrum (𝒪 L)) : u.under (𝒪 L) = u

theorem **smooth_localChar** (h : IsPlaceOver A w) (χ : G L Ω →* ℂˣ) (op : IsOpen (χ.ker : Set (G L Ω))) : IsSmoothChar (localChar (hk χ) w)

theorem **fcorr_self** (h : IsPlaceOver A w) : FCorr L w (LocExt.self h) 1 (RingEquiv.refl _) where

theorem **realises_self** (h : IsPlaceOver A w) (χ : G L Ω →* ℂˣ) (op : IsOpen (χ.ker : Set (G L Ω))) (n : ℤ) : Realises (GTerm.self χ op n) (fun u => selfTerm h χ op n u.1 ((under_self u.1).symm.trans u.2)) (fun _ => 1) (fun _ => RingEquiv.refl _)

theorem **GRel.gtw** {gl : List (GTerm k Ω)} (H : GRel gl) (ξ : G k Ω →* ℂˣ) (hξ : IsOpen (ξ.ker : Set (G k Ω))) : GRel (gtw gl ξ hξ)

theorem **gRel_self_iff** (χ : G k Ω →* ℂˣ) (op : IsOpen (χ.ker : Set (G k Ω))) : GRel [GTerm.self χ op 1] ↔ χ = 1

theorem **fcorr_locExt** {A : ValuationSubring Ω} {w : HeightOneSpectrum (𝒪 k)} {h : IsPlaceOver A w} (E : LocExt h) : FCorr E.L E.v' E 1 (RingEquiv.refl _) where

theorem **exists_gterm_proper** {A : ValuationSubring Ω} {v : HeightOneSpectrum (𝒪 k)} (h : IsPlaceOver A v) (d : ℕ) (hd : 0 < d) (n : ℤ) : ∃ t : GTerm k Ω, Module.finrank k t.L = d ∧ t.χ ≠ 1 ∧ t.n = n

theorem **exists_realises_ne_one** {A : ValuationSubring Ω} {v : HeightOneSpectrum (𝒪 k)} (h : IsPlaceOver A v) (n : ℤ) : ∃ (t : GTerm k Ω) (w : HeightOneSpectrum (𝒪 t.L)) (B : ValuationSubring Ω) (h' : IsPlaceOver B w), t.χ ≠ 1 ∧ ∃ b g i, Realises (h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsSmallStable

theorem **hasLevelTrace_line** (χ : G →* ℂˣ) {K : Subgroup G} (hK : (K : Set G) ⊆ χ.ker) (A : ℂ →ₗ[ℂ] ℂ) (c : ℂ) : HasLevelTrace (ofChar χ) K A c ↔ c = A 1

theorem **prod_pow_eq_norm** {s : k' ≃ₐ[k] k'} (hs : ∀ t, t ∈ Subgroup.zpowers s) (x : k') : ∏ j : Fin (Module.finrank k k'), (s ^ (j : ℕ)) x = algebraMap k k' (Algebra.norm k x)

theorem **det_galAct** (s : k' ≃ₐ[k] k') {n : ℕ} (g : GL (Fin n) k') : (galAct s n g : Matrix (Fin n) (Fin n) k').det = s (g : Matrix (Fin n) (Fin n) k').det

theorem **det_norm** {s : k' ≃ₐ[k] k'} (hs : ∀ t, t ∈ Subgroup.zpowers s) {n : ℕ} (g : GL (Fin n) k') : (norm k s g : Matrix (Fin n) (Fin n) k').det = algebraMap k k' (Algebra.norm k (g : Matrix (Fin n) (Fin n) k').det)

theorem **det_of_charpoly** {s : k' ≃ₐ[k] k'} (hs : ∀ t, t ∈ Subgroup.zpowers s) {n : ℕ} {g : GL (Fin n) k'} {γ : GL (Fin n) k} (h : (γ : Matrix (Fin n) (Fin n) k).charpoly.map (algebraMap k k') = (norm k s g : Matrix (Fin n) (Fin n) k').charpoly) : GeneralLinearGroup.det γ = Units.map (Algebra.norm k : k' →* k) (GeneralLinearGroup.det g)

theorem **isLBC_det** {s : k' ≃ₐ[k] k'} (hs : ∀ t, t ∈ Subgroup.zpowers s) (n : ℕ) {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) (hN : IsSmoothChar (χ.comp (Units.map (Algebra.norm k : k' →* k)))) : IsLBC s (detRep (Fin n) χ) (detRep (Fin n) (χ.comp (Units.map (Algebra.norm k : k' →* k))))

theorem **exists_small** (n : ℕ) (U : Set (GL (Fin n) k)) (hU : U ∈ 𝒩 (1 : GL (Fin n) k)) : ∃ K : Subgroup (GL (Fin n) k), IsOpen (K : Set (GL (Fin n) k)) ∧ IsCompact (K : Set (GL (Fin n) k)) ∧ (K : Set (GL (Fin n) k)) ⊆ U

theorem **exists_small_stable** (hc : ∀ t : k' ≃ₐ[k] k', Continuous t) (s : k' ≃ₐ[k] k') (n : ℕ) (U : Set (GL (Fin n) k')) (hU : U ∈ 𝒩 (1 : GL (Fin n) k')) : ∃ K : Subgroup (GL (Fin n) k'), IsOpen (K : Set (GL (Fin n) k')) ∧ IsCompact (K : Set (GL (Fin n) k')) ∧ IsStable s K ∧ (K : Set (GL (Fin n) k')) ⊆ U

theorem **eq_of_isLBC** (hc : ∀ t : k' ≃ₐ[k] k', Continuous t) {s : k' ≃ₐ[k] k'} (hs : ∀ t, t ∈ Subgroup.zpowers s) {χ : kˣ →* ℂˣ} {ψ : k'ˣ →* ℂˣ} (h : IsLBC s (detRep (Fin 1) χ) (detRep (Fin 1) ψ)) : ψ = χ.comp (Units.map (Algebra.norm k : k' →* k))

theorem **isLBC_absPow_iff** (s t : ℝ) : IsLBC (1 : k ≃ₐ[k] k) (detRep (Fin 1) (absPow k s)) (detRep (Fin 1) (absPow k t)) ↔ s = t

theorem **pow_eq_norm** {s : k' ≃ₐ[k] k'} {P : Representation ℂ (GL (Fin n) k') W} {I : W →ₗ[ℂ] W} (hI : IsTwInt k s P I) (g : GL (Fin n) k') : (P g * I) ^ Module.finrank k k' = P (norm k s g)

theorem **CharVal_unique** {π : Representation ℂ (GL (Fin n) k) V} (hπ : IsSmooth π) {γ : GL (Fin n) k} {c c' : ℂ} (h : CharVal π γ c) (h' : CharVal π γ c') : c = c'

theorem **TwCharVal_unique** [FiniteDimensional k k'] (hc : ∀ t : k' ≃ₐ[k] k', Continuous t) {s : k' ≃ₐ[k] k'} {P : Representation ℂ (GL (Fin n) k') W} (hP : IsSmooth P) {I : W →ₗ[ℂ] W} {g : GL (Fin n) k'} {c c' : ℂ} (h : TwCharVal k s P I g c) (h' : TwCharVal k s P I g c') : c = c'

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsSplitPlace

theorem **emb_surjective** (hw : (p : 𝒪 F) ∈ w.asIdeal) (h2 : (p : 𝒪 F) ∉ w.asIdeal ^ 2) (hN : Ideal.absNorm w.asIdeal = p) : Function.Surjective (Formalism.BasicNotions.Examples₂.LocF.emb w p hw)

theorem **nonempty_ident** (D : D7) : Nonempty (Ident D)

theorem **ident_unique** (D : D7) (I J : Ident D) : I = J

theorem **cls_eq_of_iso** {D : D7} (I : Ident D) (k : Fin 2) (v : D.nd.PlG k) {n : ℕ} (σ τ : GlobalTheory.SmGL.IrrRep (GL (Fin n) ((D.naut.plGw k v).adicCompletion D.host.E))) (h : Nonempty (pull I k v σ ≅ pull I k v τ)) : GlobalTheory.LocLanglands.cls σ = GlobalTheory.LocLanglands.cls τ

theorem **exists_isRc** (hL : GlobalTheory.Published.LocalLanglandsGL) {D : D7} (T : GlobalTheory.RowTie.WDTie D.nd D.naut) (I : Ident D) : ∃ rc, IsRc hL T I rc

theorem **isRc_dim** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {T : GlobalTheory.RowTie.WDTie D.nd D.naut} {I : Ident D} {rc} (h : IsRc hL T I rc) (k v n σ) : D.naut.lpDim (rc k v n (pull I k v σ)) = n

theorem **param_eq** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {T : GlobalTheory.RowTie.WDTie D.nd D.naut} {I : Ident D} {rc} (h : IsRc hL T I rc) {rep cp} (hc : IsComp I rep cp) {loc : D.what₂.FT → ∀ w, D.naut.LP w} {sat} (hr : GlobalTheory.HeightCounting.RowTrue hL D rep (fun σ w => T.e w (loc σ w)) sat) (σ k v) : loc σ (D.naut.plGw k v) = rc k v D.R.r (cp σ k v)

theorem **map_surj** {D : D7} (I : Ident D) (k v) (n : ℕ) : Function.Surjective (Matrix.GeneralLinearGroup.map (n

theorem **irrSm_pull** {D : D7} (I : Ident D) (k v) {n : ℕ} (σ) : GlobalTheory.SerSys.IrrSm (pull I k v (n

theorem **emb_le** {F : Type} [Field F] [NumberField F] (w : HeightOneSpectrum (𝒪 F)) (p : ℕ) [Fact p.Prime] (f : ℚ_[p] →+* w.adicCompletion F) (hf : Continuous f) (t : ℚ_[p]) (ht : ‖t‖ ≤ 1) : ‖f t‖ ≤ 1

theorem **exists_iso** {L : Type} [Field L] [NumberField L] [IsCMField L] (s : SplitPlace L) (p : ℕ) [hp : Fact p.Prime] (hw : (p : 𝒪 L) ∈ s.w.asIdeal) (I : ℚ_[p] ≃+* s.w.adicCompletion L) (hI : Continuous I) : ∃ i : s.v.adicCompletion (maximalRealSubfield L) ≃+* s.w.adicCompletion L, Continuous i ∧ ∀ x : L, i (s.i x) = algebraMap L (s.w.adicCompletion L) x

theorem **exists_splitPlace** (D : D7) (k v) : ∃ s : SplitPlace D.host.E, s.w = D.naut.plGw k v

theorem **isComp_const** (D : D7) (I : Ident D) : IsComp I (fun _ => constTyp D.host.E D.R.r) fun _ k v => pull I k v (GlobalTheory.SmGL.IrrRep.ofChar 1 GlobalTheory.SmGL.isSmoothChar_one)

theorem **not_isComp_zero** (D : D7) (I : Ident D) (rep) (σ : D.what₂.FT) (k : Fin 2) (v : D.nd.PlG k) (cp) (h : IsComp I rep cp) : Nontrivial (cp σ k v)

theorem **exists_pull** {D : D7} (I : Ident D) (k v) {n : ℕ} (π : Rep.{0} ℂ (GL (Fin n) ℚ_[D.host.g k])) (h : GlobalTheory.SerSys.IrrSm π) : ∃ σ, Nonempty (π ≅ pull I k v σ)

theorem **isRc_unique** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {T : GlobalTheory.RowTie.WDTie D.nd D.naut} {I : Ident D} {rc rc'} (h : IsRc hL T I rc) (h' : IsRc hL T I rc') (k v n π) (hπ : GlobalTheory.SerSys.IrrSm π) : rc k v n π = rc' k v n π

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ExistsSumMonomialAdd

theorem **eq_sum_monomial_add** (x : K⟦X⟧) (N : ℕ) : x = ∑ n ∈ Finset.range N, monomial n (coeff n x) + X ^ N * PowerSeries.mk fun n => coeff (n + N) x

theorem **exists_sum_monomial_add** {x : K⟦X⟧} (hx : x ∈ S 0 K e) (m : ℕ) : ∃ N₀, ∀ N ≥ N₀, ∃ h ∈ S 0 K e, x = ∑ n ∈ Finset.range N, monomial n (coeff n x) + (l : K⟦X⟧) ^ m * h

theorem **monomial_mem_invL** (n : ℕ) (c : K) : monomial n c ∈ invL l (A 0 K)

theorem **sameJets_Sb** : SameJets (Subalgebra.inclusion (inf_le_left : Sb 0 K l e ≤ S 0 K e)).toRingHom (l : Sb 0 K l e) where

theorem **ringHom_ext_S** (htf : ∀ x : R, (l : R) * x = 0 → x = 0) (hs : ∀ x : R, (∀ n, (l : R) ^ n ∣ x) → x = 0) (g g' : S 0 K e →+* R) (h : ∀ x : A 0 K, g (x, A_le_S l e x.2) = g' (x, A_le_S l e x.2)) : g = g'

theorem **exists_extend_S** (htf : ∀ x : R, (l : R) * x = 0 → x = 0) [IsAdicComplete (Ideal.span {(l : R)}) R] (f : A 0 K →ₐ[0] R) (hd : ∀ q : ℕ, ∃ d, (q.factorial : R) * d = f (uIn _) ^ (e * q)) : ∃ g : S 0 K e →ₐ[0] R, ∀ x : A 0 K, g (x, A_le_S l e x.2) = f x

theorem **coeff_descN** (i : ℕ) (f : K⟦X⟧) (n : ℕ) : coeff n (descN K i f) = (n.descFactorial i : K) * coeff n f

theorem **descN_mem_S** (i : ℕ) {f : K⟦X⟧} (hf : f ∈ S 0 K e) : descN K i f ∈ S 0 K e

theorem **phiA_eq_frobA_pt** (w : Ainf l) : Per.φA l w = GlobalTheory.Cyc.IoMap.frobA l w

theorem **phiA_zl_pt** (z : ℤ_[l]) : Per.φA l (algebraMap ℤ_[l] (Ainf l) z) = algebraMap ℤ_[l] (Ainf l) z

theorem **acris_nat_cancel** {n : ℕ} (hn : n ≠ 0) {x : Per.Acris l} (h : (n : Per.Acris l) * x = 0) : x = 0

theorem **exists_dpow** {x : Ainf l} (hx : Per.θ l x = 0) (n : ℕ) : ∃ y : Per.Acris l, (n.factorial : Per.Acris l) * y = ainfToAcris l x ^ n

theorem **factorial_mul_dpow** (x : Ainf l) (hx : Per.θ l x = 0) (n : ℕ) : (n.factorial : Per.Acris l) * dpow x hx n = ainfToAcris l x ^ n

theorem **eq_dpow** {x : Ainf l} (hx : Per.θ l x = 0) {n : ℕ} {y : Per.Acris l} (h : (n.factorial : Per.Acris l) * y = ainfToAcris l x ^ n) : y = dpow x hx n

theorem **dpow_zero** (x : Ainf l) (hx : Per.θ l x = 0) : dpow x hx 0 = 1

theorem **dpow_one** (x : Ainf l) (hx : Per.θ l x = 0) : dpow x hx 1 = ainfToAcris l x

theorem **PiSys.exists_dpow_teich** {e : ℕ} (ϖ : PiSys l e) (q : ℕ) : ∃ d : Per.Acris l, (q.factorial : Per.Acris l) * d = ainfToAcris l ϖ.teich ^ (e * q)

theorem **toOCr_tmul** (o : O) (w : Ainf l) : toOCr l O (o ⊗ₜ w) = o ⊗ₜ ainfToAcris l w

theorem **frobOCr_toOCr** (x : OA l O) : frobOCr l O (toOCr l O x) = toOCr l O (frobOA l O x)

theorem **galOCr_toOCr** (g : GQ l) (x : OA l O) : galOCr O g (toOCr l O x) = toOCr l O (galOA O g x)

theorem **frobOB_toOB** (x : OCr l O) : frobOB l O (toOB l O x) = toOB l O (frobOCr l O x)

theorem **galOB_toOB** (g : GQ l) (x : OCr l O) : galOB O g (toOB l O x) = toOB l O (galOCr O g x)

theorem **isUnit_l_OB** : IsUnit (l : OB l O)

theorem **complete_OCr** : IsAdicComplete (Ideal.span {(l : OCr l O)}) (OCr l O)

theorem **sep_OCr** (x : OCr l O) (h : ∀ n, (l : OCr l O) ^ n ∣ x) : x = 0

theorem **nat_cancel_OCr** {n : ℕ} (hn : n ≠ 0) {x : OCr l O} (h : (n : OCr l O) * x = 0) : x = 0

theorem **exists_iS** : ∃ g : S O K e →ₐ[O] OCr l O, ∀ x : A O K, g ⟨x, A_le_S l e x.2⟩ = toOCr l O (ϖ.iA O K x)

theorem **iS_of_mem** {f : K⟦X⟧} (h : f ∈ A O K) (h' : f ∈ S O K e) : ϖ.iS O K ⟨f, h'⟩ = toOCr l O (ϖ.iA O K ⟨f, h⟩)

theorem **iS_u** : ϖ.iS O K (uIn _) = toOCr l O (ϖ.teichR O)

theorem **iS_frob** (f : S O K e) : ϖ.iS O K ⟨frob K l f, frob_mem_S l f.2⟩ = frobOCr l O (ϖ.iS O K f)

theorem **iSl_of_mem** {f : K⟦X⟧} (h : f ∈ S O K e) (h' : f ∈ invL l (S O K e)) : ϖ.iSl O K ⟨f, h'⟩ = toOB l O (ϖ.iS O K ⟨f, h⟩)

theorem **iC_of_mem** (f : ZMod a → A O K) : ϖ.iC t K (fun τ => ⟨f τ, A_le_S l e (f τ).2⟩) = toOCr l O (ϖ.iTot t K f)

theorem **iB_of_mem** (f : ZMod a → S O K e) : ϖ.iB t K (fun τ => ⟨f τ, le_away _ _ (f τ).2⟩) = toOB l O (ϖ.iC t K f)

theorem **galOCr_iS_monomial** (g : GQ l) (n : ℕ) (c : K) (h : monomial n c ∈ S O K e) : galOCr O g (ϖ.iS O K ⟨_, h⟩) = (1 ⊗ₜ ainfToAcris l (ϖ.mu g) + 1) ^ n * ϖ.iS O K ⟨_, h⟩

theorem **sum_muDiv_descFactorial** (g : ɪ(GK (ϖ.field a))) {n M : ℕ} (h : n < M) : ∑ i ∈ Finset.range M, ϖ.muDiv a O g i * (n.descFactorial i : OCr l O) = (1 ⊗ₜ ainfToAcris l (ϖ.mu g) + 1) ^ n

theorem **taylor_S** (g : ɪ(GK (ϖ.field a))) (f : S O K e) (r : ℕ) : ∃ M₀, ∀ M ≥ M₀, (l : OCr l O) ^ r ∣ galOCr O g (ϖ.iS O K f) - ∑ i ∈ Finset.range M, ϖ.muDiv a O g i * ϖ.iS O K ⟨descN K i f, descN_mem_S e i f.2⟩

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.FieldCompatibleIffWdRel

theorem **weilRel_refl_iff** (σ σ' : Weil h) : WeilRel (RingEquiv.refl Ω) σ σ' ↔ σ = σ'

theorem **weilRel_conj_iff** (τ σ σ' : Weil h) : WeilRel ((τ : G E Ω) : Ω ≃ₐ[E] Ω).toRingEquiv σ σ' ↔ σ' = τ * σ * τ⁻¹

theorem **wdRel_refl_iff** (y y' : WD h) : WDRel (RingEquiv.refl Ω) y y' ↔ y = y'

theorem **repRel_refl_iff** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (x x' : IrrGL k) : RepRel (RingEquiv.refl k) x x' ↔ x = x'

theorem **placeIso_weil** (τ : Weil h) : PlaceIso A w A w ((τ : G E Ω) : Ω ≃ₐ[E] Ω).toRingEquiv (RingHom.id E) (RingEquiv.refl _)

theorem **exists_placeIso_ne** (h : IsPlaceOver A w) : ∃ φ : Ω ≃+* Ω, φ ≠ RingEquiv.refl Ω ∧ PlaceIso A w A w φ (RingHom.id E) (RingEquiv.refl _)

theorem **wdRel_weil** (τ : Weil h) (hτ : deg τ = 0) (y : WD h) : WDRel ((τ : G E Ω) : Ω ≃ₐ[E] Ω).toRingEquiv y y

theorem **fieldCompatible_iff** {E' Ω' : Type} [Field E'] [NumberField E'] [Field Ω'] [Algebra E' Ω'] [IsAlgClosure E' Ω'] {A' : ValuationSubring Ω'} {w' : HeightOneSpectrum (𝒪 E')} {h' : IsPlaceOver A' w'} (rc : IrrGL (w.adicCompletion E) → WD h) (rc' : IrrGL (w'.adicCompletion E') → WD h') : FieldCompatible rc rc' ↔ ∀ φ j i, PlaceIso A w A' w' φ j i → ∀ x x', RepRel i x x' → ∃ (b : WDRep h (Fin x.1 → ℂ)) (b' : WDRep h' (Fin x.1 → ℂ)), WD.mk b = rc x ∧ WD.mk b' = rc' x' ∧ b.N = b'.N ∧ ∀ (σ : Weil h) (σ' : Weil h'), WeilRel φ σ σ' → b.ρ σ = b'.ρ σ'

theorem **fieldCompatible_iff_wdRel** {E' Ω' : Type} [Field E'] [NumberField E'] [Field Ω'] [Algebra E' Ω'] [IsAlgClosure E' Ω'] {A' : ValuationSubring Ω'} {w' : HeightOneSpectrum (𝒪 E')} {h' : IsPlaceOver A' w'} (rc : IrrGL (w.adicCompletion E) → WD h) (rc' : IrrGL (w'.adicCompletion E') → WD h') (hd : ∀ x, (rc x).lpDim = x.1) : FieldCompatible rc rc' ↔ ∀ φ j i, PlaceIso A w A' w' φ j i → ∀ x x', RepRel i x x' → WDRel φ (rc x) (rc' x')

theorem **exists_isRecLit** (H : GlobalTheory.Published.LocalLanglandsGL) : ∃ rc, IsRecLit h rc

theorem **not_lawsMin_const** {a : Weil h →* (w.adicCompletion E)ˣ} {ha : IsLocalArtin h a} {e : AddChar (w.adicCompletion E) Circle → WD h → ℂˣ} (y : WD h) : ¬ LawsMin a ha e fun _ => y

theorem **segSum_nil** (rc : IrrGL (w.adicCompletion E) → WD h) : segSum rc [] = WD.lpZero h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.FiltrationZeroMembership

theorem **mem_Fil_iff** (K : FinExt l) {a : ℤ} {x : P.R.BdR} : x ∈ Fil P a ↔ x ∈ (P.PD K).Fil a

theorem **Fil_zero_mem** (a : ℤ) : (0 : P.R.BdR) ∈ Fil P a

theorem **Fil_add** {a : ℤ} {x y : P.R.BdR} (hx : x ∈ Fil P a) (hy : y ∈ Fil P a) : x + y ∈ Fil P a

theorem **Fil_neg** {a : ℤ} {x : P.R.BdR} (hx : x ∈ Fil P a) : -x ∈ Fil P a

theorem **Fil_sum** {a : ℤ} {ι : Type*} {s : Finset ι} {f : ι → P.R.BdR} (h : ∀ i ∈ s, f i ∈ Fil P a) : ∑ i ∈ s, f i ∈ Fil P a

theorem **Fil_mul** {a b : ℤ} {x y : P.R.BdR} (hx : x ∈ Fil P a) (hy : y ∈ Fil P b) : x * y ∈ Fil P (a + b)

theorem **Fil_mul_left** {a : ℤ} {x y : P.R.BdR} (hx : x ∈ Fil P 0) (hy : y ∈ Fil P a) : x * y ∈ Fil P a

theorem **Fil_mul_right** {a : ℤ} {x y : P.R.BdR} (hx : x ∈ Fil P a) (hy : y ∈ Fil P 0) : x * y ∈ Fil P a

theorem **one_mem_Fil** : (1 : P.R.BdR) ∈ Fil P 0

theorem **Fil_act** (g : GQ l) {a : ℤ} {x : P.R.BdR} (hx : x ∈ Fil P a) : P.actQdR g x ∈ Fil P a

theorem **mem_S0_of_sum_eq** {x : P.R.BdR} : x ∈ S0 P ↔ x ∈ Fil P 0

theorem **om_mem_Fil** (x : Om l) : algebraMap (Om l) P.R.BdR x ∈ Fil P 0

theorem **kd_mem_Fil** (c : ɪ(MaximalUnramifiedExtension l)) : kd P c ∈ Fil P 0

theorem **q_mem_Fil** (q : ℚ_[l]) : algebraMap ℚ_[l] P.R.BdR q ∈ Fil P 0

theorem **iota_injective** : Function.Injective P.ιB

theorem **iota_act** (g : GQ l) (b : P.R.B) : P.ιB (P.actQ g b) = P.actQdR g (P.ιB b)

theorem **iota_kur** (c : ɪ(MaximalUnramifiedExtension l)) : P.ιB (algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B c) = kd P c

theorem **fr_zero** (c : ɪ(MaximalUnramifiedExtension l)) : fr l 0 c = c

theorem **fr_add** (a b : ℕ) (c : ɪ(MaximalUnramifiedExtension l)) : fr l (a + b) c = fr l a (fr l b c)

theorem **fr_one** (c : ɪ(MaximalUnramifiedExtension l)) : fr l 1 c = frobUr l c

theorem **fr_q** (m : ℕ) (q : ℚ_[l]) : fr l m (algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) q) = algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) q

theorem **fr_comm** {r : ℕ} {c : ɪ(MaximalUnramifiedExtension l)} (h : fr l r c = c) (m : ℕ) : fr l r (fr l m c) = fr l m c

theorem **fr_mul** {r : ℕ} {c : ɪ(MaximalUnramifiedExtension l)} (h : fr l r c = c) (k : ℕ) : fr l (r * k) c = c

theorem **fr_mod** {r : ℕ} {c : ɪ(MaximalUnramifiedExtension l)} (h : fr l r c = c) (m : ℕ) : fr l (m % r) c = fr l m c

theorem **phi_iter_kur** (m : ℕ) (c : ɪ(MaximalUnramifiedExtension l)) : P.φB^[m] (algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B c) = algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B (fr l m c)

theorem **evHom_apply** (m : ℕ) (b : P.R.B) : evHom P m b = ev P m b

theorem **ev_kur** (m : ℕ) (c : ɪ(MaximalUnramifiedExtension l)) : ev P m (algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B c) = kd P (fr l m c)

theorem **ev_act** (m : ℕ) (g : GQ l) (b : P.R.B) : ev P m (P.actQ g b) = P.actQdR g (ev P m b)

theorem **act_kd** {K : FinExt l} (g : ɪ(GK K)) {c : ɪ(MaximalUnramifiedExtension l)} (hc : (c : Om l) ∈ K.1) : P.actQdR (g : GQ l) (kd P c) = kd P c

theorem **act_kur** {K : FinExt l} (g : ɪ(GK K)) {c : ɪ(MaximalUnramifiedExtension l)} (hc : (c : Om l) ∈ K.1) : P.actQ (g : GQ l) (algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B c) = algebraMap ɪ(MaximalUnramifiedExtension l) P.R.B c

theorem **act_ks** {K : FinExt l} (g : ɪ(GK K)) {c : ɪ(MaximalUnramifiedExtension l)} (hc : (c : Om l) ∈ K.1) : actSt P (g : GQ l) (ks P c) = ks P c

theorem **tz_add** (a b : ℤ) : tz P (a + b) = tz P a * tz P b

theorem **tz_zero** : tz P 0 = 1

theorem **tz_one** : tz P 1 = P.ιB P.tB

theorem **fund** {b : P.R.B} (h : P.φB b = b) (h0 : P.ιB b ∈ Fil P 0) : ∃ c : ℚ_[l], algebraMap ℚ_[l] P.R.B c = b

theorem **IsRat.resK** {n : Type} [Fintype n] [DecidableEq n] {K K' : FinExt l} (hK : K.1 ≤ K'.1) {r : ℕ} {ρ : ɪ(GK K) →* GL n ɪ(MaximalUnramifiedExtension l)} (h : IsRat r ρ) : IsRat r (resK hK ρ) where

theorem **AdmDR.resK** {n : Type} [Fintype n] [DecidableEq n] {K K' : FinExt l} (hK : K.1 ≤ K'.1) {ρ : ɪ(GK K) →* GL n ɪ(MaximalUnramifiedExtension l)} (h : AdmDR P ρ) : AdmDR P (resK hK ρ)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.FinCuspOne

theorem **cuspOn_bot** (π : Representation ℂ (GL (Fin n) k) V) : CuspOn π ⊥

theorem **finCusp_one** {F : Type} [Field F] {W : Type} [AddCommGroup W] [Module ℂ W] (τ : Representation ℂ (GL (Fin 1) F) W) : FinCusp τ

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.FiniteCocycles

@[**ext**] theorem ext {D D' : GDet n A G} (h : ∀ (B : Type) [CommRing B] [Algebra A B] (x), D.D B x = D'.D B x) : D = D'

theorem **TopFG.phiFin** (l : ℕ) [NeZero l] {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [CompactSpace G] (h : TopFG G) : PhiFin l G

theorem **PhiFin.of_surjective** {l : ℕ} {G G' : Type} [Group G] [TopologicalSpace G] [Group G'] [TopologicalSpace G'] (h : PhiFin l G) (p : G →* G') (hc : Continuous p) (hs : Function.Surjective p) : PhiFin l G'

theorem **PhiFin.of_finite** (l : ℕ) [NeZero l] (G : Type) [Group G] [Finite G] [TopologicalSpace G] : PhiFin l G

theorem **PhiFin.finite_cocycles** {l : ℕ} [Fact l.Prime] {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [CompactSpace G] (hG : PhiFin l G) {M : Type} [AddCommGroup M] [Finite M] (hl : ∀ m : M, l • m = 0) (H : Subgroup G) (hH : IsOpen (H : Set G)) (L R : G → M → M) (hL : ∀ h ∈ H, ∀ m, L h m = m) (hR : ∀ h ∈ H, ∀ m, R h m = m) : {f : G → M | IsLocallyConstant f ∧ ∀ g h, f (g * h) = L g (f h) + R h (f g)}.Finite

theorem **Frame.l_eq_zero** (Φ : Frame) : (Φ.l : Φ.k) = 0

theorem **PolGp.finite_cocycles** {Φ : Frame} (G : PolGp Φ) (i i' : Fin Φ.n) : {b : G.Γ → Φ.k | IsLocallyConstant b ∧ ∀ g h, b (g * h) = ((G.chi i g : Φ.kˣ) : Φ.k) * b h + b g * ((G.chi i' h : Φ.kˣ) : Φ.k)}.Finite

theorem **PolGp.star_single** (j : Φ.O →+* B) (g : G.Γ) (a : B) : star G j (MonoidAlgebra.single g a) = MonoidAlgebra.single (G.sig g)⁻¹ (a * j ((G.mu g : Φ.Oˣ) : Φ.O))

theorem **PolGp.star_zero** (j : Φ.O →+* B) : star G j (0 : MonoidAlgebra B G.Γ) = 0

theorem **PolGp.star_add** (j : Φ.O →+* B) (x y : MonoidAlgebra B G.Γ) : star G j (x + y) = star G j x + star G j y

theorem **PolGp.star_mul** (j : Φ.O →+* B) (x y : MonoidAlgebra B G.Γ) : star G j (x * y) = star G j y * star G j x

theorem **PolGp.star_star** (j : Φ.O →+* B) (x : MonoidAlgebra B G.Γ) : star G j (star G j x) = x

theorem **map_ofRep** {n : ℕ} {A A' G : Type} [CommRing A] [CommRing A'] [Group G] (ρ : G →* Matrix (Fin n) (Fin n) A) (f : A →+* A') : (GDet.ofRep ρ).map f = GDet.ofRep (f.mapMatrix.toMonoidHom.comp ρ)

theorem **chi_ofRep** {n : ℕ} {A G : Type} [CommRing A] [Group G] (ρ : G →* Matrix (Fin n) (Fin n) A) (y : MonoidAlgebra A G) : (GDet.ofRep ρ).chi y = (GDet.repAlg ρ A y).charpoly

theorem **tr_ofRep** {n : ℕ} {A G : Type} [CommRing A] [Nontrivial A] [Group G] (ρ : G →* Matrix (Fin (n+1)) (Fin (n+1)) A) (y : MonoidAlgebra A G) : (GDet.ofRep ρ).tr y = (GDet.repAlg ρ A y).trace

theorem **adic** {X R : Type} [TopologicalSpace X] [DiscreteTopology X] [CommRing R] (I : Ideal R) (f : X → R) : AdicCont I f

theorem **three_mem** : ((3 : ℕ) : ℤ_[3]) ∈ maximalIdeal ℤ_[3]

theorem **three_k** : (3 : ResidueField ℤ_[3]) = 0

theorem **neg_one_ne** : (-1 : ResidueField ℤ_[3]) ≠ 1

theorem **tri** (B : Type) [CommRing B] (hB : (3 : B) = 0) (g : S3) : ((rhoZ g 1 0 : ℤ) : B) = 0 ∧ ((rhoZ g 0 0 : ℤ) : B) = 1 ∧ ((rhoZ g 1 1 : ℤ) : B) = ((sgn g : ℤ) : B)

theorem **det_char3** (K : Type) [CommRing K] (h3 : (3 : K) = 0) : GDet.ofRep (rho K) = GDet.ofRep (diagRep fun i => (chs K i).comp sgn)

theorem **rep_gp** (A : Type) [CommRing A] (g : S3) : GDet.repAlg (rho A) A (gp A g) = rho A g

theorem **rep_sc** (A : Type) [CommRing A] (a : A) : GDet.repAlg (rho A) A (sc S3 a) = a • 1

theorem **charpoly_inv** (g : S3) : (rhoZ g⁻¹).charpoly = (rhoZ g).charpoly

theorem **o_zero** (A : Type) [CommRing A] (a b : ℤˣ) : (rho A (inc a) - ((chs A 0 a : Aˣ) : A) • 1) ∗ (rho A (inc b) - ((chs A 1 b : Aˣ) : A) • 1) = 0

theorem **rep_oEl** (g : Fin 2 → ℤˣ) : GDet.repAlg (rho ℤ_[3]) ℤ_[3] (oEl pol () (chs ℤ_[3]) g) = 0

theorem **j_eq** : (MvPowerSeries.constantCoeff : MvPowerSeries (Fin 0) ℤ_[3] →+* ℤ_[3]).comp MvPowerSeries.C = RingHom.id _

theorem **star_gp** (j : ℤ_[3] →+* ℤ_[3]) (g : S3) : star pol j (gp ℤ_[3] g) = gp ℤ_[3] g⁻¹

theorem **res_chs** (i : Fin 2) (g : ℤˣ) : residue ℤ_[3] ((chs ℤ_[3] i g : ℤ_[3]ˣ) : ℤ_[3]) = ((chs (ResidueField ℤ_[3]) i g : (ResidueField ℤ_[3])ˣ) : ResidueField ℤ_[3])

theorem **lam_inv** (i : ℕ) (g : S3) : (GDet.ofRep (rho ℤ_[3])).lam i (gp ℤ_[3] g⁻¹) = (GDet.ofRep (rho ℤ_[3])).lam i (gp ℤ_[3] g)

theorem **one_sub** (A : Type) [CommRing A] : gp A (1 : S3) - 1 = 0

theorem **rep_one_sub** (A : Type) [CommRing A] : GDet.repAlg (rho A) A (gp A (1 : S3) - 1) = 0

theorem **tr_zero** (x y : MonoidAlgebra ℤ_[3] S3) (hy : GDet.repAlg (rho ℤ_[3]) ℤ_[3] y = 0) : (GDet.ofRep (rho ℤ_[3])).tr (x ∗ y) = 0

theorem **exists_pair** : ∃ (Φ : Frame) (G : PolGp Φ), Φ.n = 2 ∧ Nonempty Φ.Vl ∧ Nonempty Φ.St ∧ Nat.card G.Γ = 6 ∧ Nonempty (Pair G)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.FiniteDimensionalSpan

theorem **Via.lat_fg** {L : Submodule (e.comp (algebraMap R (FractionRing R))).range A} (h : L.FG) : (Via.lat L).FG

theorem **inSpan_fg** {L : Submodule R V} (h : L.FG) : (inSpan L).FG

theorem **span_inSpan** (L : Submodule R V) : Submodule.span (FractionRing R) (inSpan L : Set (Submodule.span (FractionRing R) (L : Set V))) = ⊤

theorem **finiteDimensional_span** {L : Submodule R V} (h : L.FG) : FiniteDimensional (FractionRing R) (Submodule.span (FractionRing R) (L : Set V))

theorem **exists_lift** {M : Type} [AddCommGroup M] [Module R M] (g : V →ₛₗ[e] A) (r : M →ₛₗ[e.comp (algebraMap R (FractionRing R))] A) (hr : Function.Injective r) (hdiv : ∀ a : R, a ≠ 0 → ∀ m : M, ∃ m', a • m' = m) : ∃ φ : V →ₗ[R] M, ∀ v, g v ∈ Set.range r → r (φ v) = g v

theorem **divisible_baseChange** (Y : NExp D.R D.host D.what₂ D.LL0 N X) (i : X.Tw) (n : ℕ) (b : Bool) (k : ℕ) (a : X.Ok i k) (ha : a ≠ 0) (m : X.Ok i k ⊗[X.O i] Y.Fm i n b) : ∃ m', a • m' = m

theorem **finite_dvd_normProd** (s : Finset (Pl D.host.E)) : {v : Pl D.host.E | ((normProd s : ℕ) : 𝒪 D.host.E) ∈ v.asIdeal}.Finite

theorem **lat_fg** (hL : ρ.LatFG) (i : X.Tw) : ∀ n b k Q ψ, (lat ρ i n b k Q ψ).FG | 0, _, _, _, _ => Submodule.fg_bot | n + 1, b, k, Q, ψ => by rw [lat] split_ifs with h · exact Via.lat_fg (hL i ⟨n, h.1⟩ b k Q ψ h.2) · exact Submodule.fg_bot abbrev MtV (ρ : Setup X) (i : X.Tw) (n : ℕ) (b : Bool) (k : ℕ) (Q : Finset (Pl D.host.E)) (ψ : G D.host.E N.Om →* (X.Ok i k)ˣ) : Type

theorem **exists_thP** (i : X.Tw) (j : Fin D.R.m) (b : Bool) (k : ℕ) (Q : Finset (Pl D.host.E)) (ψ : G D.host.E N.Om →* (X.Ok i k)ˣ) : ∃ φ : MtV ρ i ((j : ℕ) + 1) b k Q ψ →ₗ[X.Ok i k] X.Ok i k ⊗[X.O i] Y.Fm i ((j : ℕ) + 1) b, ∀ f, ρ.sum i j b k Q ψ (rdV ρ i _ b k Q ψ f) ∈ Set.range (rdK (rd i j b) hρ k) → rdK (rd i j b) hρ k (φ f) = ρ.sum i j b k Q ψ (rdV ρ i _ b k Q ψ f)

theorem **thPsi_one** (i : X.Tw) : ∀ n b k (d : Dat rd i n b), thPsi hρ rd i n b k ∅ 1 d = {(1 : X.Ok i k) ⊗ₜ[X.O i] th rd i n b d} | 0, _, _, d => d.elim | n + 1, b, k, d => by ext x rw [thPsi, Set.mem_preimage, Setup.gens_one, Set.mem_singleton_iff, Set.mem_singleton_iff, ← ((hF i ⟨n, d.1⟩ b).injective_rdK hρ k).eq_iff, rdK_tmul, map_one, one_smul, ← d.2.rat.choose_spec] rfl end Model variable (hF : ∀ i j b, Faithful (rd i j b)) (hL : ρ.LatFG) def withSeries : NExp D.R D.host D.what₂ D.LL0 N X

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.FinrankMaxUnramifiedSubfieldPadicFinExtPos

theorem **dickson_hom** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (x : A) (N : ℕ) : (dickson 1 (1 : B) N).eval (f x) = f ((dickson 1 (1 : A) N).eval x)

theorem **root_mem_Kur** (y : Om l) (hy : y ^ (ordCompl[l] H.condN) = 1) : y ∈ (MaximalUnramifiedExtension l).toSubfield

theorem **theta_mem_Kur** : J (thL H l) ∈ MaximalUnramifiedExtension l

theorem **adjoin_gen_top** {p : ℕ} (hp : p.Prime) (hpS : ¬ p ∣ H.QS) : IntermediateField.adjoin ℚ ({thL H p, H.sqq, H.sqd} : Set H.E) = ⊤

theorem **image_le** (hlS : ¬ l ∣ H.QS) (M : IntermediateField ℚ_[l] (Om l)) (h1 : J (thL H l) ∈ M) (h2 : J (H.sqq) ∈ M) (h3 : J (H.sqd) ∈ M) (x : H.E) : J x ∈ M

theorem **liesOver_of_mem** {E : Type} [Field E] [NumberField E] {p : ℕ} (hp : p.Prime) {v : HeightOneSpectrum (𝒪 E)} (hv : v ∈ placesOver E p) : v.asIdeal.LiesOver (Ideal.span {(p : ℤ)})

theorem **ramIdx_eq_one** {R : Ranks} (H : Host R) {l : ℕ} (hl : l.Prime) (hlN : ¬ l ∣ H.condN) (v : HeightOneSpectrum (𝒪 H.E)) (hv : v ∈ placesOver H.E l) : v.asIdeal.ramificationIdx ℤ = 1

theorem **image_mem_Kur** (hlN : ¬ l ∣ H.condN) (x : H.E) : J x ∈ MaximalUnramifiedExtension l

theorem **kw_le_Kur** (hlN : ¬ l ∣ H.condN) (Kw : FinExt l) (hE : ∀ x : H.E, J x ∈ Kw.1) (hdense : (Kw.1 : Set (Om l)) ⊆ closure (Set.range J)) : Kw.1 ≤ MaximalUnramifiedExtension l

theorem **kw_le_Kur_of_ne** (i : D.nd.Loc) (hlN : ¬ D.nd.l i ∣ D.host.condN) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : w ∈ placesOver D.host.E (D.nd.l i)) : (D.nbr i w hw).Kw.1 ≤ Formalism.BasicNotions.Loc.MaximalUnramifiedExtension (D.nd.l i)

theorem **FilV_anti'** (a : ℤ) : FilV PD (LL

theorem **mu_mul** (c d : K) (x : V ⊗[Qp] BdR) : μ (c ∗ d) x = μ c (μ d x)

theorem **mu_add** (c d : K) (x : V ⊗[Qp] BdR) : μ (c + d) x = μ c x + μ d x

theorem **mu_alg** (q : Qp) (x : V ⊗[Qp] BdR) : μ (algebraMap Qp K q) x = algebraMap Qp LL q • x

theorem **mu_pow** (c : K) (n : ℕ) (x : V ⊗[Qp] BdR) : μ (c ^ n) x = ((μ c) ^ n) x

theorem **mu_aeval** (c : K) (p : Qp[X]) (x : V ⊗[Qp] BdR) : aeval (μ c) (p.map (algebraMap Qp LL)) x = μ (aeval c p) x

theorem **mu_mem** (c : K) (a : ℤ) (x : V ⊗[Qp] BdR) (hx : x ∈ DdR PD ρ ⊓ FilV PD (LL

theorem **aeval_mem** {W : Type} [AddCommGroup W] [Module LL W] (M : Module.End LL W) (U : Submodule LL W) (h : ∀ x ∈ U, M x ∈ U) (q : LL[X]) (x : W) (hx : x ∈ U) : aeval M q x ∈ U

theorem **jump_of_total** [CharZero Qp] [FiniteDimensional Qp K] (hs : Nat.card (K →ₐ[Qp] LL) = finrank Qp K) (a : ℤ) (h : DdR PD ρ ⊓ FilV PD (LL

theorem **Ks_mono** (K : FinExt l) {s s' : ℕ} (h : s ≤ s') : Ks K s ≤ Ks K s'

theorem **good_one** {K LL : FinExt l} {s : ℕ} (h : Good K s LL) (h2 : l ≠ 2) (hK : K.1 ≤ Ks K 1) : Good K 1 LL

theorem **card_emb** (K LL : FinExt l) (h : ∀ (τ : ↥K.1 →ₐ[ℚ_[l]] Om l) (x : ↥K.1), τ x ∈ LL.1) : Nat.card (↥K.1 →ₐ[ℚ_[l]] ↥LL.1) = finrank ℚ_[l] ↥K.1

theorem **finrank_sup_adjoin** (F : IntermediateField ℚ_[l] (Om l)) (y : Om l) : finrank ℚ_[l] ↥(F ⊔ IntermediateField.adjoin ℚ_[l] {y}) = finrank ℚ_[l] ↥F * (minpoly F y).natDegree

theorem **Ks_eq** (K : FinExt l) (s : ℕ) {z : Om l} (hz : IsPrimitiveRoot z (l ^ s)) : Ks K s = K0 K ⊔ IntermediateField.adjoin ℚ_[l] {z}

theorem **finrank_maxUnramifiedSubfield_padicFinExt_pos** (K : FinExt l) : 0 < finrank ℚ_[l] ↥(K0 K)

theorem **le_Ks_one_of_Kur** (K : FinExt l) (h : K.1 ≤ MaximalUnramifiedExtension l) : K.1 ≤ Ks K 1

theorem **cTB_le** (l h t N e : ℕ) (hl : l.Prime) (U : Finset ℤ) (U' : Set ℤ) (hU : U' ⊆ U) (hN : U.card ≤ N) (hb : ∀ u ∈ U, -(h : ℤ) ≤ u ∧ u + 1 ≤ h) (he : e < t) (hth : t ≤ h) : LocalSituation.cTB l 1 e 0 (LocalSituation.cLL l (LocalSituation.mN l 1 (h + h)) (LocalSituation.mN l 1 (h + h) - 1) U' 0 ((e : ℤ) + 1)) ≤ cBnum l h t N

theorem **vZ_le_log** (l : ℕ) [Fact l.Prime] (h : ℕ) (u : ℤ) (hu : -(h : ℤ) ≤ u ∧ u ≤ h) : LocalSituation.vZ l u ≤ Nat.log l h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.GaloisExportOfMatrixDetBounds

theorem **ordHyp_X** (l M : ℕ) : OrdHyp O l M 1 1 0 (MvPolynomial.X (Sum.inl 0) + MvPolynomial.X (Sum.inr 0))

theorem **addVal_le_of_ordHyp** {l M R v : ℕ} (hl : l ≠ 0) {f : MvPolynomial (Fin R ⊕ Fin R) O} (h : OrdHyp O l M R 0 v f) : IsDiscreteValuationRing.addVal O (f.coeff 0) ≤ v

theorem **not_ordHyp_X** {l M v : ℕ} (hl : l ≠ 0) : ¬ OrdHyp O l M 1 0 v (MvPolynomial.X (Sum.inr 0))

theorem **exists_isCoord** (Y : NExp D.R D.host D.what₂ D.LL0 N X) (i : X.Tw) (M : ℕ) : ∃ d, IsCoord Y i M d

theorem **toLam_surjective** (i : X.Tw) (M : ℕ) (ι : Type) [Fintype ι] : Function.Surjective (toLam X i M ι)

theorem **galois_export_of_matrix_det_bounds** (bkP : BKPh X) (hS : ThmS bkP) (hS4 : SelmerLengthsByDeterminant bkP) (h3 : ResidualBoundOddPrimes bkP) (hC : ThmC bkP) : GaloisSideTwistedSelmerAuxiliaryPrimes bkP

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.GoodIdempotentFixesMemGoodTypeSpace

theorem **isEigSys_iff** {ev : ℕ → PlE D.host → FractionRing (X.O i)} : IsEigSys Y i j b ev ↔ Eigsp Y i j b ev ⊓ V0 Y i j b ≠ ⊥

theorem **IsEigSys.congr** {ev ev' : ℕ → PlE D.host → FractionRing (X.O i)} (h : IsEigSys Y i j b ev) (e : ∀ k P, HkIdx i k P → ev k P = ev' k P) : IsEigSys Y i j b ev'

theorem **good_idempotent_fixes_mem_good_type_space** {x : Y.Fm i j b} (hx : x ∈ V0 Y i j b) : eG Y i j b x = x

theorem **bijOn_iff** : Set.BijOn (sysOf (X

theorem **SigTiedAt.bijOn** (h : SigTiedAt Y i) : Set.BijOn (sysOf (X

theorem **SigTiedAt.goodE** (h : SigTiedAt Y i) (s : X.Sig i) : goodE Y i s

theorem **SigTiedAt.mem'** (h : SigTiedAt Y i) (s : X.Sig i) {k : ℕ} {P : PlE D.host} (hk : k ≤ D.R.r) (hP : IsSplitPl D.host P) (hS : P ∉ X.S0 i) : ∃ x : X.O0 i, algebraMap (X.O0 i) (X.O i) x = X.hEig i s k P

theorem **SigTiedAt.hEig_one** (h : SigTiedAt Y i) (s : X.Sig i) {P : PlE D.host} (hP : IsSplitPl D.host P) (hS : P ∉ X.S0 i) {σ : G D.host.E N.Om} (hσ : IsFrob (F

theorem **fracMap_algebraMap** (i : X.Tw) (x : X.O0 i) : fracMap X i (algebraMap (X.O0 i) (FractionRing (X.O0 i)) x) = algebraMap (X.O i) (FractionRing (X.O i)) (algebraMap (X.O0 i) (X.O i) x)

theorem **liftIdx_jS** (s : X.Sig i) : liftIdx (Finset.mem_Icc.2 (X.jS_range i s)) = jIdx s

theorem **EigParAt.full** (h : EigParAt C Rw Y i) (iC : FractionRing (X.O i) →+* ℂ) (hC : ∀ x, iC (embO X i b x) = embLL0 D x) {j : ℕ} (hj : j ∈ Finset.Icc 1 D.R.m) {ev : ℕ → PlE D.host → FractionRing (X.O i)} (hev : IsEigSys Y i j b ev) : ∃ σ π, D.nd.goodRow σ ∧ D.what₂.isot j σ ≠ ⊥ ∧ IsRowPar Rw σ π ∧ ∀ k P, HkIdx i k P → iC (ev k P) = heckeVal (muBase C (X.l i) (liftIdx hj)) π k P

theorem **EigParAt.evidence** (h : EigParAt C Rw Y i) (ρ : Reading X i b) {j : ℕ} (hj : j ∈ Finset.Icc 1 D.R.m) {ev : ℕ → PlE D.host → X.O0 i} (hev : IsEigSys Y i j b fun k P => algebraMap (X.O i) (FractionRing (X.O i)) (algebraMap (X.O0 i) (X.O i) (ev k P))) : ∃ σ π, D.nd.goodRow σ ∧ D.what₂.isot j σ ≠ ⊥ ∧ IsRowPar Rw σ π ∧ ∀ k P, HkIdx i k P → ρ.toC (ev k P) = heckeVal (muBase C (X.l i) (liftIdx hj)) π k P

theorem **EigParAt.exists_row** (h : EigParAt C Rw Y i) (hS : SigTiedAt Y i) (s : X.Sig i) (ρ : Reading X i (X.side i s)) : ∃ σ π, D.nd.goodRow σ ∧ D.what₂.isot (X.jS i s) σ ≠ ⊥ ∧ IsRowPar Rw σ π ∧ ∀ k P, k ≤ D.R.r → IsSplitPl D.host P → P ∉ X.S0 i → ∃ x : X.O0 i, algebraMap (X.O0 i) (X.O i) x = X.hEig i s k P ∧ ρ.toC x = heckeVal (muBase C (X.l i) (jIdx s)) π k P

theorem **norm_int_le** {L : Type} [Field L] [NumberField L] {l : ℕ} [Fact l.Prime] (a : L →+* Qbar l) (x : 𝓞 L) : ‖a x‖ ≤ 1

theorem **ball_of_int** {L : Type} [Field L] [NumberField L] {l : ℕ} [Fact l.Prime] (a b : L →+* Qbar l) (h : ∀ x : 𝓞 L, ‖a x‖ < 1 ↔ ‖b x‖ < 1) (x : L) : ‖a x‖ ≤ 1 ↔ ‖b x‖ ≤ 1

theorem **val_pos_iff** (w : D.what₂.places.Pl) {x : D.what₂.LL} (hx : x ≠ 0) : 0 < D.what₂.places.val w x ↔ ¬ 0 ≤ D.what₂.places.val w x⁻¹

theorem **IsoInduces.lt_iff** {l : ℕ} [Fact l.Prime] {ι : Qbar l ≃+* ℂ} {w : D.what₂.places.Pl} (h : IsoInduces ι w) (x : D.what₂.LL) : 0 < D.what₂.places.val w x ↔ ‖ι.symm (D.what₂.embLL x)‖ < 1

theorem **resChar_eq** {w : D.what₂.places.Pl} {l : ℕ} (hl : l.Prime) (h : 0 < D.what₂.places.val w (l : D.what₂.LL)) : D.what₂.places.resChar w = l

theorem **norm_lt_iff** (ρ : Reading X i b) (z : FractionRing (X.O0 i)) : (∃ y ∈ IsLocalRing.maximalIdeal (X.O0 i), algebraMap (X.O0 i) (FractionRing (X.O0 i)) y = z) ↔ ‖ρ.κ z‖ < 1

theorem **exists_reading_of_induces** {b' : Bool} (ρ : Reading X i b') {ι : Qbar (X.l i) ≃+* ℂ} {w : D.what₂.places.Pl} (h : IsoInduces ι w) (hw : ∀ x : ↥D.LL0, 0 ≤ D.what₂.places.val w x ↔ ‖ρ.κ (embSide X i b x)‖ ≤ 1) : ∃ ρ' : Reading X i b, ρ'.ι = ι

theorem **ResTied.exists_reading** {Rw : RowSide D} (h : ResTied D Rw) (hs : ∀ x : ɪD.LL0, star (x : D.what₂.LL) ∈ D.LL0) (hi : X.NonzeroIndexPlace i) (b : Bool) : ∃ (ρ : Reading X i b) (w : D.what₂.places.Pl) (red : Zbar (X.l i) →+* FbAt D w), ρ.Induces w ∧ ResAt Rw ρ.ɪ w red

theorem **mem_invariant_forms** (i : D.ntw.Tw) (j : ℕ) (b : Bool) (x : (ex D).Fm i j b) : x ∈ V0 (ex D) i j b

theorem **coeff_charpolyRev_scalar** {A : Type} [CommRing A] (n k : ℕ) (c : A) : (c • (1 : Matrix (Fin n) (Fin n) A)).charpolyRev.coeff k = (-1) ^ k * (n.choose k : A) * c ^ k

theorem **tmat_eq** (i : D.ntw.Tw) (s : (tw D).Sig i) (g : GalE D) : Tmat (X

theorem **hkFrob** (i : D.ntw.Tw) (s : (tw D).Sig i) : HkFrob (X

theorem **sigTiedAt** (i : D.ntw.Tw) : SigTiedAt (X

theorem **exists_sigTied** : ∃ (X : NTw D.R D.host D.what₂ D.LL0 D.nd) (Y : NExp D.R D.host D.what₂ D.LL0 D.nd X) (e : X.Tw ≃ D.ntw.Tw), (∀ i, X.NonzeroIndexPlace i ↔ D.ntw.NonzeroIndexPlace (e i)) ∧ (∀ i, Fintype.card (X.Sig i) = D.R.m * 2) ∧ (∀ i, SigTiedAt Y i) ∧ SigTiedOf Y

theorem **not_sigTiedAt_of_isEmpty** {X : NTw D.R D.host D.what₂ D.LL0 D.nd} {Y : NExp D.R D.host D.what₂ D.LL0 D.nd X} {i : X.Tw} [IsEmpty (X.Sig i)] {j : ℕ} (hj : j ∈ Finset.Icc 1 D.R.m) {b ev} (h : IsEigSys Y i j b ev) : ¬ SigTiedAt Y i

theorem **not_eigParAt** (C : CharFrame D) (Rw : RowSide D) (h : ∀ σ, ¬ D.nd.goodRow σ) (i : D.ntw.Tw) : ¬ EigParAt C Rw (X

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.GoodSetupCountEqZeroOffBadPlaces

theorem **embLL0_labLL0** (τ : D.what₂.Tau) : (embLL0 D).comp (labLL0 D τ) = tauE D τ

theorem **Good.weil_base** {ρ : Setup X} (h : ρ.Good) (i : X.Tw) (j : Fin D.R.m) (b : Bool) : (ρ.B i j b).IsWeil 1 (ρ.om i j b 1)

theorem **eO_injective** : Function.Injective (ρ.eO i b)

theorem **eOk_injective** (k : ℕ) : Function.Injective (ρ.eOk i b k)

theorem **Good.isCoefRing** {ρ : Setup X} (h : ρ.Good) (i : X.Tw) (b : Bool) : IsCoefRing (embLL0 D) (labLL0 D) (ρ.eO i b).range

theorem **Good.range_le** {ρ : Setup X} (h : ρ.Good) (i : X.Tw) (b : Bool) (k : ℕ) : (ρ.eO i b).range ≤ (ρ.eOk i b k).range

theorem **Good.isCoefRing_k** {ρ : Setup X} (h : ρ.Good) (i : X.Tw) (b : Bool) (k : ℕ) : IsCoefRing (embLL0 D) (labLL0 D) (ρ.eOk i b k).range

theorem **gens_one** (k : ℕ) (h : GA D (Vc D j)) (P : PolF D (Wc D j).rank (Vc D j).rank) : ρ.gens i j b k ∅ 1 h P = {ρ.theta i j b h P}

theorem **xi_of_empty** (Y : NExp D.R D.host D.what₂ D.LL0 N X) {Q : Finset (Pl D.host.E)} {M : ℕ} (hQ : half (ρ.ch i) (admissibleConjugatePairPlaces X i Q M) = ∅) (h : GA D (Vc D j)) (P : PolF D (Wc D j).rank (Vc D j).rank) (a) : ρ.xi i j b Y Q M h P a = ρ.theta i j b h P

theorem **twOK_one** (i : X.Tw) (k : ℕ) : TwOK i k ∅ (1 : G D.host.E N.Om →* (X.Ok i k)ˣ)

theorem **rd_smul** (a : X.O i) (f : Y.Fm i ((j : ℕ) + 1) b) : rd (a • f) = ρ.eO i b a • rd f

theorem **rdK_tmul** (hρ : ρ.Good) (k : ℕ) (a : X.Ok i k) (f : Y.Fm i ((j : ℕ) + 1) b) : rdK rd hρ k (a ⊗ₜ f) = ρ.eOk i b k a • rd f

theorem **Faithful.injective** (h : Faithful rd) : Function.Injective rd

theorem **Faithful.injective_rdK** (h : Faithful rd) (hρ : ρ.Good) (k : ℕ) : Function.Injective (rdK rd hρ k)

theorem **thU_of_empty** {Q : Finset (Pl D.host.E)} {M : ℕ} (hQ : InRange X i Q) (h0 : half (ρ.ch i) (admissibleConjugatePairPlaces X i Q M) = ∅) (d : Y.Dat i ((j : ℕ) + 1) b) : Y.thU i _ b Q M d = Y.th i _ b d ⊗ₜ[X.O i] 1

theorem **thU_empty** (M : ℕ) (d : Y.Dat i ((j : ℕ) + 1) b) : Y.thU i _ b ∅ M d = Y.th i _ b d ⊗ₜ[X.O i] 1

theorem **thU_of_rat** (hR : RatUniv rd) {Q : Finset (Pl D.host.E)} (M : ℕ) (hQ : InRange X i Q) {d : Y.Dat i ((j : ℕ) + 1) b} (hd : Disjoint Q (Y.lev i _ b d)) : ∃ x, (∀ a, rd (x a) = ρ.xi i j b Y Q M (T.cls d) (T.coef d) a) ∧ Y.thU i _ b Q M d = grpSum Y (ρ.ch i) Q M x

theorem **rdV_mem** {k Q ψ} (h : TwOK i k Q ψ) {f : Y.MtV i ((j : ℕ) + 1) b k Q ψ} : f ∈ Y.MtLat i _ b k Q ψ ↔ T.rdV k Q ψ f ∈ ρ.mt i j b k Q ψ

theorem **nonempty_dat** : Nonempty (Y.Dat i ((j : ℕ) + 1) b)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.H1MapSubgroup

theorem **repOn_map** (f : B →ɪ[Λ] B') (g : Γ) (m : Fin r → B) : f.compLeft (Fin r) (repOn ρ κ B g m) = repOn ρ κ B' g (f.compLeft (Fin r) m)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.HasPairGammaZeroIff

theorem **exists_isSelfDualFor** [MeasurableSpace k] [BorelSpace k] {ψ : AddChar k Circle} {c : kˣ} (hψ : HasConductor ψ c) : ∃ μ : Measure k, IsSelfDualFor ψ μ

theorem **arg_three_one** (g : GL (Fin 1) k) (x : Fin 1 → Fin 1 → k) : (emb₂ k (stdSum (rfl : 1 + (1 + 1) = 3)) (g, 1) * lowUnip rfl (padRow x) * emb₂ k (stdSum (rfl : 1 + (1 + 1) = 3)) (1, longElt (1 + 1) k)).val = !![g.val 0 0, 0, 0; x 0 0, 0, 1; 0, 1, 0]

theorem **lowUnip_zero** {m r n : ℕ} (hn : m + r = n) : lowUnip hn (0 : Matrix (Fin r) (Fin m) k) = 1

theorem **longElt_one** : longElt (0 + 1) k = 1

theorem **hasPairGamma_succ_iff** {m : ℕ} {ψ : AddChar k Circle} {c : kˣ} (hψ : HasConductor ψ c) (π : Representation ℂ (GL (Fin (m + 1)) k) V) (τ : Representation ℂ (GL (Fin m) k) V') (e : ℂ) : HasPairGamma ψ π τ e ↔ HasPairEps ψ π τ e

theorem **hasPairGamma_zero_iff** {j : ℕ} {ψ : AddChar k Circle} {c : kˣ} (hψ : HasConductor ψ c) (π : Representation ℂ (GL (Fin (0 + (j + 1))) k) V) (hπ : IsGeneric ψ π) (e : ℂ) : HasPairGamma ψ π (detRep (Fin 0) 1) e ↔ e = 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.Imports

theorem **rel_one** {B : Obj l ɪ(OL K)} (u : B.A) (f : Hom bk.prism B) : ht.Rel u f f 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.InstTameDescentExponentDataPadicThree

theorem **pow_pow_mod** (ζ : rootsOfUnity e K) (n : ℕ) : ζ ^ l ^ (n % a) = ζ ^ l ^ n

theorem **powL_apply** (s : ZMod a) (ζ : rootsOfUnity e K) : powL l s ζ = ζ ^ l ^ s.val

@[simp] theorem **powL_zero** (ζ : rootsOfUnity e K) : powL l (0 : ZMod a) ζ = ζ

theorem **powL_add** (s t : ZMod a) (ζ : rootsOfUnity e K) : powL l (s + t) ζ = powL l s (powL l t ζ)

theorem **powL_one** (ζ : rootsOfUnity e K) : powL l (1 : ZMod a) ζ = ζ ^ l

@[simp] theorem **shift_one** : shift (1 : TameGal K l e a) = 0

theorem **shift_mul** (g h : TameGal K l e a) : shift (g * h) = shift g + shift h

@[simp] theorem **shift_inertia** (ζ : rootsOfUnity e K) : shift (inertia ζ : TameGal K l e a) = 0

@[simp] theorem **shift_fr** : shift (fr : TameGal K l e a) = 1

@[simp] theorem **root_one** (τ : ZMod a) : root (1 : TameGal K l e a) τ = 1

@[simp] theorem **root_fr** (τ : ZMod a) : root (fr : TameGal K l e a) τ = 1

@[simp] theorem **root_inertia_zero** (ζ : rootsOfUnity e K) : root (inertia ζ : TameGal K l e a) 0 = (ζ : Kˣ)

theorem **root_mul** (g h : TameGal K l e a) (τ : ZMod a) : root (g * h) τ = root g τ * root h (τ - shift g)

theorem **root_sub_one** (g : TameGal K l e a) (τ : ZMod a) : root g (τ - 1) = root g τ ^ l

theorem **root_pow** (g : TameGal K l e a) (τ : ZMod a) : root g τ ^ e = 1

theorem **fr_mul_inertia** (ζ : rootsOfUnity e K) : (fr : TameGal K l e a) * inertia ζ = inertia (ζ ^ l) * fr

theorem **root_ne_zero** (g : TameGal K l e a) (τ : ZMod a) : root g τ ≠ 0

theorem **norm_root** (g : TameGal K l e a) (τ : ZMod a) : ‖root g τ‖ = 1

theorem **card** : Nat.card (TameGal K l e a) = e * a

theorem **not_dvd_card** : ¬ l ∣ Nat.card (TameGal K l e a)

theorem **ringAct_apply** (g : TameGal K l e a) (f : ZMod a → K⟦X⟧) (τ : ZMod a) : ringAct g f τ = rescale (g.root τ) (f (τ - g.shift))

theorem **ringAct_frobTot** (g : TameGal K l e a) (f : ZMod a → K⟦X⟧) : ringAct g (frobTot K a l f) = frobTot K a l (ringAct g f)

theorem **ringAct_NTot** (g : TameGal K l e a) (f : ZMod a → K⟦X⟧) : ringAct g (NTot f) = NTot (ringAct g f)

theorem **ringAct_X** (g : TameGal K l e a) : ringAct g (fun _ => X) = fun τ => C (g.root τ) * X

theorem **ringAct_X_pow** (g : TameGal K l e a) : ringAct g (fun _ => X ^ e) = fun _ => X ^ e

theorem **ringAct_E** (g : TameGal K l e a) : ringAct g (fun _ => E K l e) = fun _ => E K l e

theorem **join_eta** (w : Vec K a (ι ⊕ Unit)) : join (fun τ i => w τ (.inl i)) (fun τ => w τ (.inr ())) = w

theorem **join_add** (v v' : Vec K a ι) (f f' : ZMod a → K⟦X⟧) : join (v + v') (f + f') = join v f + join v' f'

theorem **join_smul** (c : K) (v : Vec K a ι) (f : ZMod a → K⟦X⟧) : join (c • v) (c • f) = c • join v f

theorem **join_smul_tot** (c : ZMod a → K⟦X⟧) (v : Vec K a ι) (f : ZMod a → K⟦X⟧) : join (c • v) (c * f) = c • join v f

theorem **PhiMod.join_mem_Mext** {O : Type} [CommRing O] [IsUltrametricDist K] [Algebra O K] [IsUnitBall O K] {l e : ℕ} [Fintype ι] [DecidableEq ι] (P : PhiMod O K l e a ι) {y : Vec K a ι} (hy : y ∈ P.Mext (invE l e (A O K))) {B : Subalgebra O K⟦X⟧} {v : Vec K a ι} {f : ZMod a → K⟦X⟧} : join v f ∈ (P.plus hy).Mext B ↔ v ∈ P.Mext B ∧ ∀ τ, f τ ∈ B

theorem **norm_coeff_rescale** [NeZero e] {c : K} (hc : c ^ e = 1) (f : K⟦X⟧) (n : ℕ) : ‖coeff n (rescale c f)‖ = ‖coeff n f‖

theorem **rot_A** [NeZero e] : Rot e (A O K)

theorem **rot_S** [NeZero e] : Rot e (S O K e)

theorem **rot_R** [NeZero e] (k : ℝ) : Rot e (R O K l k)

theorem **Rot.invL** {B : Subalgebra O K⟦X⟧} (h : Rot e B) : Rot e (invL l B)

theorem **Rot.invE** {B : Subalgebra O K⟦X⟧} [HasU B] (h : Rot e B) : Rot e (invE l e B)

theorem **Rot.ringAct_mem** {B : Subalgebra O K⟦X⟧} (h : Rot e B) (g : TameGal K l e a) {f : ZMod a → K⟦X⟧} (hf : ∀ τ, f τ ∈ B) (τ : ZMod a) : ringAct g f τ ∈ B

theorem **matAct_apply** (g : TameGal K l e a) (G : ZMod a → Matrix ι ι K⟦X⟧) (v : Vec K a ι) (τ : ZMod a) : matAct g G v τ = G τ *ᵥ fun i => rescale (g.root τ) (v (τ - g.shift) i)

theorem **act_E_pow_smul** (n : ℕ) (v : Vec K a ι) : Δ.act g (E K l e ^ n • v) = E K l e ^ n • Δ.act g v

theorem **act_E_smul** (v : Vec K a ι) : Δ.act g (E K l e • v) = E K l e • Δ.act g v

theorem **act_X_pow_smul** (v : Vec K a ι) : Δ.act g ((X : K⟦X⟧) ^ e • v) = (X : K⟦X⟧) ^ e • Δ.act g v

theorem **act_mem_uPow** {m : ℕ} {v : Vec K a ι} (hv : v ∈ uPow K a ι m) : Δ.act g v ∈ uPow K a ι m

theorem **act_mem_Mext** {B : Subalgebra O K⟦X⟧} (hA : A O K ≤ B) (hB : Rot e B) {v : Vec K a ι} (hv : v ∈ P.Mext B) : Δ.act g v ∈ P.Mext B

theorem **act_mem_ext** {B : Subalgebra O K⟦X⟧} (hA : A O K ≤ B) (hB : Rot e B) {v : Vec K a ι} (hv : v ∈ P.ext B) : Δ.act g v ∈ P.ext B

theorem **act_conn** (v : Vec K a ι) : Δ.act g (P.conn v) = P.conn (Δ.act g v)

theorem **act_zeta** (y : Vec K a ι) : Δ.act g (P.zeta y) = P.zeta (Δ.act g y)

theorem **red_act_cst_red** (v : Vec K a ι) : red (Δ.act g (cst (red v))) = red (Δ.act g v)

theorem **red_act** (v : Vec K a ι) : red (Δ.act g v) = Δ.actD g (red v)

theorem **actD_phiD** (d : P.D) : Δ.actD g (P.phiD d) = P.phiD (Δ.actD g d)

theorem **act_xi** (d : P.D) : Δ.act g (P.xi d) = P.xi (Δ.actD g d)

theorem **plusAct_join** (v : Vec K a ι) (f : ZMod a → K⟦X⟧) : Δ.plusAct g (join v f) = join (Δ.act g v) (ringAct g f)

theorem **plus_act_join** [NeZero e] {y : Vec K a ι} (hy : y ∈ P.Mext (invE l e (A O K))) (hinv : y ∈ Δ.inv) (v : Vec K a ι) (f : ZMod a → K⟦X⟧) : (Δ.plus hy hinv).act g (join v f) = join (Δ.act g v) (ringAct g f)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IntFormsSubset

theorem **intForms_subset** (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n : ℕ) (S : Set (HeightOneSpectrum (𝒪 C.d.c₂.F))) : C.intForms a n S ⊆ C.ratForms a n

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IntegrandPeriodic

theorem **cA_inc** (e : D.host.E) : cA D (incA D e) = incA D (star e)

theorem **hO_eq_trace** (x y : WO V W) : hO V W x y = Matrix.trace ((V.gram⁻¹).map (incA D) * ((y.map (cA D))ᵀ * W.gram.map (incA D) * x))

theorem **hO_ratMul** (C : Matrix (Fin W.rank) (Fin W.rank) D.host.E) (u v : WO V W) : hO V W (ratMul C u) (ratMul C v) = Matrix.trace ((V.gram⁻¹).map (incA D) ∗ ((v.map (cA D))ᵀ ∗ (Cᴴ ∗ W.gram ∗ C).map (incA D) ∗ u))

theorem **gram_eq** {Y : Lag W} {C} (h : IsLagFrame W Y C) : C.1ᴴ ∗ W.gram ∗ C.1 = C.2ᴴ ∗ W.gram ∗ C.2

theorem **ratMul_add** (C : Matrix (Fin W.rank) (Fin W.rank) D.host.E) (γ) (u : WO V W) : ratMul C (ratW W V γ + u) = ratW W V (C ∗ γ) + ratMul C u

theorem **integrand_periodic** (hψ : ∀ x, ‖ψ x‖ = 1) {Y : Lag W} {C} (h : IsLagFrame W Y C) {f₁ f₂ : WO V W → ℂ} (h₁ : IsThetaFunO ψ V W f₁) (h₂ : IsThetaFunO ψ V W f₂) (γ) (u : WO V W) : f₁ (ratMul C.1 (ratW W V γ + u)) ∗ star (f₂ (ratMul C.2 (ratW W V γ + u))) = f₁ (ratMul C.1 u) ∗ star (f₂ (ratMul C.2 u))

theorem **swPair_rank_zero** {νV 𝓗 νW 𝓖 μ 𝓕 fr} (hs : SWSetup ψ V W χV χW om νV 𝓗 νW 𝓖 μ 𝓕 fr) (h : W.rank = 0) (f₁ f₂ : WO V W → ℂ) : HasSum (fun Y : Lag W => ((νV 𝓗).toReal : ℂ) ∗ lagTerm μ 𝓕 (fr Y) f₁ f₂) (∫ h in 𝓗, thetaKO om h 1 f₁ ∗ star (thetaKO om h 1 f₂) ∂νV)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.InvertibleSpanSingleton

theorem **Delta.δ_zero** {A : Type} [CommRing A] (D : Delta l A) : D.δ 0 = 0

@[simp] theorem **Delta.ofLift_φ** (l : ℕ) [Fact l.Prime] {A : Type} [CommRing A] (φ : A →+∗ A) (htf : ∀ x : A, (l : A) ∗ x = 0 → x = 0) (h : ∀ a, ∃ b, φ a = a ^ l + (l : A) ∗ b) : (Delta.ofLift l φ htf h).φ = φ

theorem **Delta.map_φ_of_map_δ** {A B : Type} [CommRing A] [CommRing B] (D : Delta l A) (D' : Delta l B) (f : A →+∗ B) (h : ∀ a, f (D.δ a) = D'.δ (f a)) (a : A) : f (D.φ a) = D'.φ (f a)

theorem **Delta.map_δ_of_map_φ** {A B : Type} [CommRing A] [CommRing B] (D : Delta l A) (D' : Delta l B) (htf : ∀ x : B, (l : B) ∗ x = 0 → x = 0) (f : A →+∗ B) (h : ∀ a, f (D.φ a) = D'.φ (f a)) (a : A) : f (D.δ a) = D'.δ (f a)

theorem **Hom.f_φ** {P P' : Obj l R} (f : Hom P P') (a : P.A) : f.f (P.D.φ a) = P'.D.φ (f.f a)

theorem **Hom.ext'** {P P' : Obj l R} {f g : Hom P P'} (h : f.f = g.f) : f = g

theorem **Delta.l_mem_of_isUnit** (hl : 0 < l) {A : Type} [CommRing A] (D : Delta l A) {q : A} (hq : IsUnit (D.δ q)) : (l : A) ∈ Ideal.span {q} ⊔ (Ideal.span {q}).map D.φ

theorem **invertible_span_singleton** {A : Type} [CommRing A] {q : A} (hq : q ∈ nonZeroDivisors A) : Module.Invertible A ↥(Ideal.span {q})

theorem **Obj.crys_quot_nontrivial** (l : ℕ) [Fact l.Prime] : Nontrivial ((Obj.crys l).A ⧸ (Obj.crys l).I)

theorem **semiBC_self** {A B : Type} [CommRing A] [CommRing B] (f : A →+∗ B) : SemiBC f.toSemilinearMap

theorem **FCrys.unit_effective** (l : ℕ) (R : Type) [CommRing R] : (FCrys.unit l R).Effective

theorem **FCrys.unit_crys_Φ_ne_zero** (l : ℕ) [hl : Fact l.Prime] : (FCrys.unit l ℤ_[l]).Φ (Obj.crys l) ((l : ℤ_[l]), Ideal.mem_span_singleton_self _) (1 : ℤ_[l]) ≠ 0

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IotaInjOfTensorInj

theorem **exists_core_facts** (l : ℕ) [Fact l.Prime] : ∃ C : Core l, C.PhiInj ∧ C.IotaInj ∧ C.InvdR

theorem **iotaInj_of_tensorInj** (C : Core l) (h : C.TensorInj) : C.IotaInj

theorem **sigma_tB** (C : Core l) (g : GQ l) : Per.σB g C.tB = C.qB (((C.χ g : ℤ_[l]ˣ) : ℤ_[l]) : ℚ_[l]) ∗ C.tB

theorem **phi_tB** (C : Core l) : Per.φB l C.tB = ((l : ℕ) : Per.Bcris l) ∗ C.tB

theorem **inv_t_not_plus** (C : Core l) (x : BcrisP l) : plusToB l x ∗ C.tB ≠ 1

theorem **qB_of_fits** (h : Fits T C) (x : ℚ_[l]) : T.eB (algebraMap ℚ_[l] P.R.B x) = C.qB x

theorem **fund_of_fits** (h : Fits T C) : C.Fund

theorem **invB_of_fits** (h : Fits T C) : C.InvB

theorem **isField_of_isStd** (h : GlobalTheory.Per.IsStd P AI) : IsField P.R.BdR

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsAdicCompleteOfPowLe

theorem **liftInj_spec** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+∗ C) (g : B →+∗ C) (hg : Function.Injective g) (h : ∀ x, ∃ y, g y = f x) (x : A) : g (liftInj f g hg h x) = f x

theorem **isAdicComplete_of_pow_le** {A : Type} [CommRing A] {I J : Ideal A} {N : ℕ} (h : I ^ (N + 1) ≤ J) (hJ : J ≤ I) [IsAdicComplete I A] : IsAdicComplete J A where

theorem **ringHom_ext_of_sep** {R B : Type} [CommRing R] [CommRing B] (f g : R⟦X⟧ →+∗ B) (hC : ∀ r, f (C r) = g (C r)) (hX : f X = g X) (hs : ∀ x : B, (∀ n, f X ^ n ∣ x) → x = 0) : f = g

theorem **mem_resSub** {x : OCb l} : x ∈ resSub l a ↔ x ^ l ^ a = x

theorem **one_lt_pow_l** : 1 < l ^ a

theorem **isUnit_resSub** (x : ↥(resSub l a)) : IsUnit x ∨ x = 0

theorem **Fq.val_injective** : Function.Injective (Fq.val l a)

theorem **Fq.pow_card** (x : Fq l a) : x ^ l ^ a = x

theorem **finite_sol** (R : Type) [CommRing R] [IsDomain R] : {x : R | x ^ l ^ a = x}.Finite

theorem **exists_gen** : ∃ (z : Fq l a) (ω : Om l), IsPrimitiveRoot ω (l ^ a − 1) ∧ IsTiltOf ω (l ^ a − 1) (Fq.val l a z) ∧ ∀ x : Fq l a, x ≠ 0 → ∃ k : ℕ, x = z ^ k

theorem **card_Fq** : Nat.card (Fq l a) = l ^ a

theorem **wA_injective** : Function.Injective (wA l a)

theorem **wA_teich** (x : Fq l a) : wA l a (WittVector.teichmuller l x) = WittVector.teichmuller l (Fq.val l a x)

theorem **coeff_sigma** (w : Wq l a) (n : ℕ) : (sigma l a w).coeff n = w.coeff n ^ l

theorem **coeff_sigma_pow** (k : ℕ) (w : Wq l a) (n : ℕ) : ((sigma l a ^ k) w).coeff n = w.coeff n ^ l ^ k

theorem **sigma_pow_eq_one** : sigma l a ^ a = 1

theorem **sigma_pow_teich** (k : ℕ) (x : Fq l a) : (sigma l a ^ k) (WittVector.teichmuller l x) = WittVector.teichmuller l x ^ l ^ k

theorem **wA_sigma** (w : Wq l a) : wA l a (sigma l a w) = WittVector.frobenius (wA l a w)

theorem **span_teich** : Submodule.span ℤ_[l] (Set.range (WittVector.teichmuller l : Fq l a → Wq l a)) = ⊤

theorem **dvd_of_not_isUnit** {w : Wq l a} (h : ¬ IsUnit w) : (l : Wq l a) ∣ w

theorem **Kunr_le_Kur** : Kunr l a ≤ MaximalUnramifiedExtension l

theorem **seq_pow** (n : ℕ) : ϖ.seq n ^ l ^ n = ϖ.seq 0

theorem **seq_pow_mul** (n : ℕ) : ϖ.seq n ^ (l ^ n * e) = -(l : Om l)

theorem **seq_ne_zero** [NeZero e] (n : ℕ) : ϖ.seq n ≠ 0

theorem **norm_seq_lt_one** (n : ℕ) : ‖(ϖ.seq n : 𝕆_[l])‖ < 1

@[simp] theorem **coe_seqO** (n : ℕ) : (ϖ.seqO n : 𝕆_[l]) = algebraMap (Om l) 𝕆_[l] (ϖ.seq n)

theorem **seqO_succ** (n : ℕ) : ϖ.seqO (n + 1) ^ l = ϖ.seqO n

theorem **seqO_zero** : ϖ.seqO 0 ^ e = -(l : OC l)

theorem **coeff_flat** (n : ℕ) : PreTilt.coeff n ϖ.flat = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : OC l)}) (ϖ.seqO n)

theorem **untilt_flat** : ϖ.flat.untilt = ϖ.seqO 0

theorem **theta_teich** : Per.θ l ϖ.teich = ϖ.seqO 0

theorem **flat_ne_zero** [NeZero e] : ϖ.flat ≠ 0

theorem **not_isUnit_seqO** (n : ℕ) : ¬ IsUnit (ϖ.seqO n)

theorem **not_isUnit_teich** : ¬ IsUnit ϖ.teich

theorem **isUnit_inclW_teich** [NeZero e] : IsUnit (inclW l ϖ.teich)

theorem **complete_teich** : IsAdicComplete (Ideal.span {ϖ.teich}) (Ainf l)

theorem **sep_teich** (x : Ainf l) (h : ∀ n, ϖ.teich ^ n ∣ x) : x = 0

theorem **pi_mem_field** : ϖ.seq 0 ∈ (ϖ.field a).1

theorem **Kunr_le_field** : Kunr l a ≤ (ϖ.field a).1

theorem **gal_pi** (g : ↥(GK (ϖ.field a))) : (g : GQ l) (ϖ.seq 0) = ϖ.seq 0

theorem **field_le_fieldInf** : (ϖ.field a).1 ≤ ϖ.fieldInf a

theorem **seq_mem_fieldInf** (n : ℕ) : ϖ.seq n ∈ ϖ.fieldInf a

theorem **fixing_fieldInf_le** : (ϖ.fieldInf a).fixingSubgroup ≤ GK (ϖ.field a)

theorem **exists_ev** : ∃ f : (Wq l a)⟦X⟧ →+* Ainf l, f X = ϖ.teich ∧ ∀ w, f (C w) = wA l a w

@[simp] theorem **ev_X** : ϖ.ev a X = ϖ.teich

@[simp] theorem **ev_C** (w : Wq l a) : ϖ.ev a (C w) = wA l a w

theorem **eq_ev** (f : (Wq l a)⟦X⟧ →+* Ainf l) (hX : f X = ϖ.teich) (hC : ∀ w, f (C w) = wA l a w) : f = ϖ.ev a

@[simp] theorem **frobS_X** : frobS l a X = X ^ l

@[simp] theorem **frobS_C** (w : Wq l a) : frobS l a (C w) = C (sigma l a w)

theorem **dvd_of_map_eq_zero** (g : (Wq l a)⟦X⟧) (h : PowerSeries.map (WittVector.constantCoeff (p

theorem **l_eq_zero** : (l : (Fq l a)⟦X⟧) = 0

theorem **l_ne_zero** : (l : (Wq l a)⟦X⟧) ≠ 0

theorem **frobS_eq** (x : (Wq l a)⟦X⟧) : ∃ b, frobS l a x = x ^ l + (l : (Wq l a)⟦X⟧) * b

theorem **constantCoeff_eis** : constantCoeff (eis l a e) = (l : Wq l a)

theorem **dvd_or_unit** (x : (Wq l a)⟦X⟧) : eis l a e ∣ x ∨ ∃ (m : ℕ) (v y : (Wq l a)⟦X⟧), IsUnit v ∧ x = v * X ^ m + eis l a e * y

theorem **complete_eis** : IsAdicComplete (Ideal.span {(l : (Wq l a)⟦X⟧)} ⊔ Ideal.span {eis l a e}) (Wq l a)⟦X⟧

theorem **l_mem_eis** : (l : (Wq l a)⟦X⟧) ∈ Ideal.span {eis l a e} ⊔ (Ideal.span {eis l a e}).map (deltaS l a).φ

theorem **theta_ev_eis** : Per.θ l (ϖ.ev a (eis l a e)) = 0

theorem **theta_ev_eq_zero_iff** (x : (Wq l a)⟦X⟧) : Per.θ l (ϖ.ev a x) = 0 ↔ eis l a e ∣ x

theorem **expand** {l : ℕ} [Fact l.Prime] (F : IntermediateField ℚ_[l] (PadicAlgCl l)) (π : PadicAlgCl l) {e : ℕ} (he : e ≠ 0) (hπ : π ^ e = -(l : PadicAlgCl l)) (x : PadicAlgCl l) (hx : x ∈ F ⊔ IntermediateField.adjoin ℚ_[l] {π}) : ∃ c : ℕ → PadicAlgCl l, (∀ i, c i ∈ F) ∧ x = ∑ i ∈ Finset.range e, c i * π ^ i

theorem **norm_l_lt** {l : ℕ} [Fact l.Prime] : ‖(l : PadicAlgCl l)‖ < 1

theorem **coeff_le_one** {l : ℕ} [hl : Fact l.Prime] (π : PadicAlgCl l) {e : ℕ} (he : e ≠ 0) (hπ : π ^ e = -(l : PadicAlgCl l)) (c : ℕ → PadicAlgCl l) (hc : ∀ i, c i ≠ 0 → ∃ n : ℤ, ‖c i‖ = ‖(l : PadicAlgCl l)‖ ^ n) (hx : ‖∑ i ∈ Finset.range e, c i * π ^ i‖ ≤ 1) (i : ℕ) (hi : i < e) : ‖c i‖ ≤ 1

theorem **norm_l_lt'** {l : ℕ} [Fact l.Prime] : ‖(l : 𝕆_[l])‖ < 1

theorem **mem_int_iff** {l : ℕ} [Fact l.Prime] (x : 𝕆_[l]) : x ∈ PadicComplexInt l ↔ ‖x‖ ≤ 1

theorem **hom_padicInt** {l : ℕ} [Fact l.Prime] (f : ℤ_[l] →+* 𝕆_[l]) (hf : ∀ z, ‖f z‖ ≤ 1) (z : ℤ_[l]) : f z = algebraMap ℚ_[l] 𝕆_[l] (z : ℚ_[l])

theorem **frac** {l : ℕ} [hl : Fact l.Prime] {W : Type} [CommRing W] (g : ℤ_[l] →+* W) (φ : W →+* 𝕆_[l]) (hφ : ∀ w, ‖φ w‖ ≤ 1) (S : Set (PadicAlgCl l)) (hS : ∀ ζ ∈ S, ∃ w, φ w = algebraMap (PadicAlgCl l) 𝕆_[l] ζ) (c : PadicAlgCl l) (hc : c ∈ IntermediateField.adjoin ℚ_[l] S) : ∃ (n : ℕ) (w : W), φ w = (l : 𝕆_[l]) ^ n * algebraMap (PadicAlgCl l) 𝕆_[l] c

theorem **unram** {l : ℕ} [hl : Fact l.Prime] {W : Type} [CommRing W] (hW : ∀ w : W, w ≠ 0 → ∃ (m : ℕ) (u : Wˣ), w = (l : W) ^ m * u) (φ : W →+* 𝕆_[l]) (hφ : ∀ w, ‖φ w‖ ≤ 1) (c : PadicAlgCl l) (hc : ∃ (n : ℕ) (w : W), φ w = (l : 𝕆_[l]) ^ n * algebraMap (PadicAlgCl l) 𝕆_[l] c) : (c ≠ 0 → ∃ n : ℤ, ‖c‖ = ‖(l : PadicAlgCl l)‖ ^ n) ∧ (‖c‖ ≤ 1 → ∃ w, φ w = algebraMap (PadicAlgCl l) 𝕆_[l] c)

theorem **theta_seq** {l : ℕ} [hl : Fact l.Prime] (s : ℕ → Formalism.BasicNotions.Loc.OC l) (hs : ∀ n, s (n + 1) ^ l = s n) (t : Formalism.BasicNotions.Loc.OCb l) (ht : ∀ n, t.1 n = Ideal.Quotient.mk (Ideal.span {(l : Formalism.BasicNotions.Loc.OC l)}) (s n)) : Formalism.Per.θ l (WittVector.teichmuller l t) = s 0

theorem **root_lift** {l : ℕ} [hl : Fact l.Prime] (a : ℕ) [NeZero a] (ζ : PadicAlgCl l) (hζ : ζ ^ l ^ a = ζ) : ∃ w : Wq l a, ((Formalism.Per.θ l (wA l a w) : Formalism.Per.OC l) : 𝕆_[l]) = algebraMap (PadicAlgCl l) 𝕆_[l] ζ

theorem **wq_unit** {l : ℕ} [hl : Fact l.Prime] (a : ℕ) [NeZero a] (w : Wq l a) (hw : w ≠ 0) : ∃ (m : ℕ) (u : (Wq l a)ˣ), w = (l : Wq l a) ^ m * u

theorem **Bridge.exists_powerSeries_eval_eq** : ∀ {l : ℕ} [hl : Fact (Nat.Prime l)] (a : ℕ) [inst : NeZero a] {e : ℕ} [NeZero e] (ϖ : GlobalTheory.BKReal.PiSys l e) (x : ɪ(Formalism.BasicNotions.Loc.OL (GlobalTheory.BKReal.PiSys.field a ϖ))), ∃ (y : PowerSeries (GlobalTheory.BKReal.Wq l a)), (Formalism.Per.θ l) ((GlobalTheory.BKReal.PiSys.ev a ϖ) y) = (GlobalTheory.Prism.intToOC (GlobalTheory.BKReal.PiSys.field a ϖ)) x

theorem **exists_theta_ev_eq** (x : ɪ(OL (ϖ.field a))) : ∃ y, Per.θ l (ϖ.ev a y) = intToOC (ϖ.field a) x

theorem **quotToOC_mk** (x : (Wq l a)⟦X⟧) : ϖ.quotToOC a (Ideal.Quotient.mk _ x) = Per.θ l (ϖ.ev a x)

theorem **quotToOC_injective** : Function.Injective (ϖ.quotToOC a)

theorem **quotToOC_strMap** (x : ɪ(OL (ϖ.field a))) : ϖ.quotToOC a (ϖ.strMap a x) = intToOC (ϖ.field a) x

theorem **ev_frobS** (x : (Wq l a)⟦X⟧) : ϖ.ev a (frobS l a x) = WittVector.frobenius (ϖ.ev a x)

theorem **toAinf_f** (x : (Wq l a)⟦X⟧) : (ϖ.toAinf a).f x = ϖ.ev a x

theorem **span_ev_eis** : Ideal.span {ϖ.ev a (eis l a e)} = RingHom.ker (Per.θ l)

theorem **isUnit_inclW_ev_eis** : IsUnit (inclW l (ϖ.ev a (eis l a e)))

theorem **coe_ratio** (g : GQ l) (n : ℕ) : (ϖ.ratio g n : ℂ_[l]) = algebraMap (Om l) ℂ_[l] (g (ϖ.seq n) / ϖ.seq n)

theorem **ratio_succ** (g : GQ l) (n : ℕ) : ϖ.ratio g (n + 1) ^ l = ϖ.ratio g n

theorem **coeff_kummer** (g : GQ l) (n : ℕ) : PreTilt.coeff n (ϖ.kummer g) = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : OC l)}) (ϖ.ratio g n)

theorem **gal_flat** (g : GQ l) : Per.σF g ϖ.flat = ϖ.kummer g * ϖ.flat

theorem **untilt_kummer** (g : GQ l) : (ϖ.kummer g).untilt = ϖ.ratio g 0

theorem **gal_teich** (g : GQ l) : Per.σA g ϖ.teich = (1 + ϖ.mu g) * ϖ.teich

theorem **theta_mu** {g : GQ l} (hg : g (ϖ.seq 0) = ϖ.seq 0) : Per.θ l (ϖ.mu g) = 0

theorem **existsUnique_isExponent** (ζ : ℕ → Om l) (h0 : ζ 0 = 1) (hs : ∀ n, ζ (n + 1) ^ l = ζ n) (h1 : ζ 1 ≠ 1) {g : GQ l} (hg : g (ϖ.seq 0) = ϖ.seq 0) : ∃! x, ϖ.IsExponent ζ g x

theorem **IsExponent.coeff_kummer** {ϖ : PiSys l e} {ζ : ℕ → Om l} {g : GQ l} {x : ℤ_[l]} (h : ϖ.IsExponent ζ g x) (n : ℕ) (z : OC l) (hz : (z : ℂ_[l]) = algebraMap (Om l) ℂ_[l] (ζ n)) : PreTilt.coeff n (ϖ.kummer g) = Ideal.Quotient.mk (Ideal.span {((l : ℕ) : OC l)}) z ^ (PadicInt.toZModPow n x).val

theorem **gal_wA** {g : GQ l} (hg : ∀ x ∈ Kunr l a, g x = x) (w : Wq l a) : Per.σA g (wA l a w) = wA l a w

theorem **PiSys.gal_ev** {e : ℕ} [NeZero e] (ϖ : PiSys l e) {g : GQ l} (hg : g ∈ ϖ.GfieldInf a) (x : (Wq l a)⟦X⟧) : Per.σA g (ϖ.ev a x) = ϖ.ev a x

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsCharsOfReading

theorem **toC_eq_of_mem** (ρ ρ' : Reading X i b) {x : X.O0 i} {y : ɪD.LL0} (h : ρ.toC x = embLL0 D y) : ρ'.toC x = embLL0 D y

theorem **almostIn_blockChar** (hC : ValLL0 D C) (l : ℕ) (j : Fin D.R.m) : GlobalTheory.GalChar.AlmostIn (embLL0 D) (blockChar C l j)

theorem **isGal_of_reading** (ha : IsGlobalArtin art) (hC : ValLL0 D C) (ρ ρ' : Reading X i b) {j : Fin D.R.m} {χ : GalE D →* (X.O0 i)ˣ} {n} (h : IsGal art ρ.κ ρ.ι (blockChar C (X.l i) j) n χ) : ∃ n', IsGal art ρ'.κ ρ'.ι (blockChar C (X.l i) j) n' χ

theorem **notMem_l** {k : Fin 2} (hk : X.l i ≠ D.host.g k) (v : D.nd.PlG k) : ((X.l i : ℕ) : 𝒪 D.host.E) ∉ (D.naut.plGw k v).asIdeal

theorem **isChars_of_reading** (ha : IsGlobalArtin art) (hC : ValLL0 D C) (ρ ρ' : Reading X i b) (h : IsChars C art ρ) : IsChars C art ρ'

theorem **exists_tameTie** (ha : IsGlobalArtin art) (hP : ∀ k v, C.P.Div k (D.naut.plGw k v)) : ∃ c, TameTie C art c

theorem **rd_conj** (i : D.ntw.Tw) (x : ɪD.LL0) : (rd D i).ι ((rd D i).κ (D.ntw.emb0b i x)) = starRingEnd ℂ (embLL0 D x)

theorem **del_val** (l : ℕ) : (C.delAt l).ValuesIn 0 (embLL0 D)

theorem **chars_nonempty** (i : D.ntw.Tw) : Nonempty (Chars D C art i)

theorem **isTripleFor_ex** (ha' : IsGlobalArtin art') (hT : TameTie C art' D.nd.chiT) : IsTripleFor C art' (ex D C K)

theorem **exists_isTripleOf** (D : D7) [IsEmpty D.nd.Loc] (C : CharFrame D) {art : Ideles D.host.E →* Gab D.host.E D.nd.Om} (ha : IsGlobalArtin art) (hC : ValLL0 D C) (hs : ∀ x : ɪD.LL0, star (x : D.what₂.LL) ∈ D.LL0) (hT : TameTie C art D.nd.chiT) : ∃ (X : NTw D.R D.host D.what₂ D.LL0 D.nd) (Y : NExp D.R D.host D.what₂ D.LL0 D.nd X) (e : X.Tw ≃ D.ntw.Tw), (∀ i, X.NonzeroIndexPlace i ↔ D.ntw.NonzeroIndexPlace (e i)) ∧ IsTripleOf Y

theorem **not_isTripleOf** {X : NTw D.R D.host D.what₂ D.LL0 D.nd} (Y : NExp D.R D.host D.what₂ D.LL0 D.nd X) {i : X.Tw} (hi : X.NonzeroIndexPlace i) (s : X.Sig i) {P : PlE D.host} (hP : IsSplitPl D.host P) (hS : P ∉ X.S0 i) (h : X.hEig i s 0 P ≠ 1) : ¬ IsTripleOf Y

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsCuspidalIffCompositions

theorem **blockDiag_iff** {b : Fin N → ℕ} {M : Matrix (Fin N) (Fin N) k} : (∀ i j, b i ≠ b j → M i j = 0) ↔ M.BlockTriangular b ∧ M.BlockTriangular (OrderDual.toDual ∘ b)

theorem **mem_levi** {b : Fin N → ℕ} {g : GL (Fin N) k} : g ∈ levi k b ↔ ∀ i j, b i ≠ b j → (g : Matrix (Fin N) (Fin N) k) i j = 0

theorem **levi_congr** {b b' : Fin N → ℕ} (h : ∀ i j, b i = b j ↔ b' i = b' j) : levi k b = levi k b'

theorem **levi_le_blockTri** (b : Fin N → ℕ) : levi k b ≤ blockTri b

theorem **unipRadical_le_blockTri** (b : Fin N → ℕ) : unipRadical k b ≤ blockTri b

theorem **diagPart_apply** (b : Fin N → ℕ) (M : Matrix (Fin N) (Fin N) k) (i j : Fin N) : diagPart b M i j = if b i = b j then M i j else 0

theorem **diagPart_mul** {b : Fin N → ℕ} {X Y : Matrix (Fin N) (Fin N) k} (hX : X.BlockTriangular b) (hY : Y.BlockTriangular b) : diagPart b (X * Y) = diagPart b X * diagPart b Y

theorem **leviProj_apply** (b : Fin N → ℕ) (p : blockTri (κ

theorem **leviProj_inclusion** (b : Fin N → ℕ) (m : levi k b) : leviProj b (Subgroup.inclusion (levi_le_blockTri b) m) = m

theorem **leviProj_eq_one_iff** (b : Fin N → ℕ) (p : blockTri (κ

theorem **ker_leviProj** (b : Fin N → ℕ) : (leviProj (k

theorem **blockTri_le_normalizer** (b : Fin N → ℕ) : blockTri b ≤ Subgroup.normalizer ((unipRadical k b : Subgroup (GL (Fin N) k)) : Set (GL (Fin N) k))

theorem **inclusion_mem_normalizer** (a c : Fin N → ℕ) (m : ↥(levi k a ⊓ levi k c)) : Subgroup.inclusion (inf_le_left : levi k a ⊓ levi k c ≤ levi k a) m ∈ Subgroup.normalizer (((unipRadical k c).subgroupOf (levi k a) : Subgroup (levi k a)) : Set (levi k a))

theorem **permGL_apply** (w : Equiv.Perm (Fin N)) (i j : Fin N) : ((permGL k w : GL (Fin N) k) : Matrix (Fin N) (Fin N) k) i j = if i = w j then 1 else 0

theorem **conj_permGL_apply** (w : Equiv.Perm (Fin N)) (g : GL (Fin N) k) (i j : Fin N) : ((permGL k w * g * (permGL k w)⁻¹ : GL (Fin N) k) : Matrix (Fin N) (Fin N) k) (w i) (w j) = (g : Matrix (Fin N) (Fin N) k) i j

theorem **conj_mem_iff** {w : Equiv.Perm (Fin N)} {g : GL (Fin N) k} (R : Fin N → Fin N → Prop) (c : Matrix (Fin N) (Fin N) k) (hc : ∀ i j, c (w i) (w j) = c i j) : (∀ i j, R i j → ((permGL k w * g * (permGL k w)⁻¹ : GL (Fin N) k) : Matrix (Fin N) (Fin N) k) i j = c i j) ↔ ∀ i j, R (w i) (w j) → (g : Matrix (Fin N) (Fin N) k) i j = c i j

theorem **conj_mem_blockTri** {b : Fin N → ℕ} {w : Equiv.Perm (Fin N)} {g : GL (Fin N) k} : permGL k w * g * (permGL k w)⁻¹ ∈ blockTri b ↔ g ∈ blockTri (b ∘ w)

theorem **conj_mem_levi** {b : Fin N → ℕ} {w : Equiv.Perm (Fin N)} {g : GL (Fin N) k} : permGL k w * g * (permGL k w)⁻¹ ∈ levi k b ↔ g ∈ levi k (b ∘ w)

theorem **conj_mem_unipRadical** {b : Fin N → ℕ} {w : Equiv.Perm (Fin N)} {g : GL (Fin N) k} : permGL k w * g * (permGL k w)⁻¹ ∈ unipRadical k b ↔ g ∈ unipRadical k (b ∘ w)

theorem **comp_symm_comp** (b : Fin N → ℕ) (w : Equiv.Perm (Fin N)) : (b ∘ w.symm) ∘ w = b

@[simp] theorem **blockOf_apply** (e : (Σ a, Fin (d a)) ≃ Fin N) (x : Σ a, Fin (d a)) : blockOf e (e x) = x.1

theorem **leviEquiv_apply** (e : (Σ a, Fin (d a)) ≃ Fin N) (g : levi k (blockOf e)) (a : Fin h) (i j : Fin (d a)) : ((leviEquiv e g a : GL (Fin (d a)) k) : Matrix (Fin (d a)) (Fin (d a)) k) i j = ((g : GL (Fin N) k) : Matrix (Fin N) (Fin N) k) (e ⟨a, i⟩) (e ⟨a, j⟩)

theorem **stdEquiv_val** (d : Fin h → ℕ) (hd : ∑ a, d a = N) (a : Fin h) (i : Fin (d a)) : (stdEquiv d hd ⟨a, i⟩ : ℕ) = ∑ c ∈ Finset.Iio a, d c + i

theorem **blockOf_stdEquiv_two** {r t : ℕ} (hd : ∑ c, (![r, t] : Fin 2 → ℕ) c = N) : blockOf (stdEquiv ![r, t] hd) = twoBlocks N r

theorem **isTwoBlock_twoBlocks** (N r : ℕ) : IsTwoBlock r (twoBlocks N r)

@[simp] theorem **twoEquiv_inl** (hN : r + t = N) (i : Fin r) : (twoEquiv hN (Sum.inl i) : ℕ) = i

@[simp] theorem **twoEquiv_inr** (hN : r + t = N) (i : Fin t) : (twoEquiv hN (Sum.inr i) : ℕ) = r + i

theorem **mem_fixed** {ρ : Representation ℂ G V} {U : Subgroup G} {v : V} : v ∈ fixed ρ U ↔ ∀ u ∈ U, ρ u v = v

theorem **bijective_fixed_mkQ** (ρ : Representation ℂ G V) (U : Subgroup G) [Finite U] : Function.Bijective ((twKer ρ U 1).mkQ ∘ₗ (fixed ρ U).subtype)

theorem **Embeds.refl** (ρ : Representation ℂ G V) : Embeds ρ ρ

theorem **Embeds.trans** {ρ : Representation ℂ G V} {σ : Representation ℂ G W} {τ : Representation ℂ G X} (h : Embeds ρ σ) (h' : Embeds σ τ) : Embeds ρ τ

theorem **Embeds.of_equiv** {ρ : Representation ℂ G V} {σ : Representation ℂ G W} (e : ρ.Equiv σ) : Embeds ρ σ

theorem **isIrreducible_of_equiv** {ρ : Representation ℂ G V} {σ : Representation ℂ G W} (e : ρ.Equiv σ) (h : ρ.IsIrreducible) : σ.IsIrreducible

@[simp] theorem **extTprod_tmul** (ρ : Representation ℂ G V) (σ : Representation ℂ H W) (x : G × H) (v : V) (w : W) : extTprod ρ σ x (v ⊗ₜ w) = ρ x.1 v ⊗ₜ σ x.2 w

@[simp] theorem **extTensor_tprod** {ι : Type} {Γ Y : ι → Type} [∀ a, Group (Γ a)] [∀ a, AddCommGroup (Y a)] [∀ a, Module ℂ (Y a)] (ρ : ∀ a, Representation ℂ (Γ a) (Y a)) (x : ∀ a, Γ a) (v : ∀ a, Y a) : extTensor ρ x (PiTensorProduct.tprod ℂ v) = PiTensorProduct.tprod ℂ fun a => ρ a (x a) (v a)

theorem **cls_eq_iff** [FiniteDimensional ℂ V] [FiniteDimensional ℂ W] (ρ : Representation ℂ G V) (σ : Representation ℂ G W) : cls ρ = cls σ ↔ Nonempty (ρ.Equiv σ)

@[simp] theorem **hcRes_apply** (b : Fin N → ℕ) (ρ : Representation ℂ (GL (Fin N) k) V) (m : levi k b) (v : fixed ρ (unipRadical k b)) : (hcRes b ρ m v : V) = ρ m v

@[simp] theorem **inflate_apply** (b : Fin N → ℕ) (σ : Representation ℂ (levi k b) W) (p : blockTri (κ

theorem **mem_indSpace** {b : Fin N → ℕ} {σ : Representation ℂ (levi k b) W} {f : GL (Fin N) k → W} : f ∈ IndSpace b σ ↔ ∀ (p : blockTri (κ

@[simp] theorem **hcInd_apply** (b : Fin N → ℕ) (σ : Representation ℂ (levi k b) W) (g x : GL (Fin N) k) (f : IndSpace b σ) : (hcInd b σ g f : GL (Fin N) k → W) x = (f : GL (Fin N) k → W) (x * g)

@[simp] theorem **hcResMap_apply** (b : Fin N → ℕ) {ρ : Representation ℂ (GL (Fin N) k) V} {σ : Representation ℂ (GL (Fin N) k) W} (f : ρ.IntertwiningMap σ) (v : fixed ρ (unipRadical k b)) : (hcResMap b f v : W) = f v

@[simp] theorem **hcIndMap_apply** (b : Fin N → ℕ) {ρ : Representation ℂ (levi k b) V} {σ : Representation ℂ (levi k b) W} (f : ρ.IntertwiningMap σ) (φ : IndSpace b ρ) (x : GL (Fin N) k) : (hcIndMap b f φ : GL (Fin N) k → W) x = f ((φ : GL (Fin N) k → V) x)

theorem **Embeds.hcRes** (b : Fin N → ℕ) {ρ : Representation ℂ (GL (Fin N) k) V} {σ : Representation ℂ (GL (Fin N) k) W} (h : Embeds ρ σ) : Embeds (hcRes b ρ) (hcRes b σ)

theorem **Embeds.hcInd** (b : Fin N → ℕ) {ρ : Representation ℂ (levi k b) V} {σ : Representation ℂ (levi k b) W} (h : Embeds ρ σ) : Embeds (hcInd b ρ) (hcInd b σ)

@[simp] theorem **toHcInd_apply** (b : Fin N → ℕ) (ρ : Representation ℂ (GL (Fin N) k) V) (σ : Representation ℂ (levi k b) W) (L : V →ₗ[ℂ] W) (hL : ∀ (p : blockTri (κ

theorem **toHcInd_injective** (b : Fin N → ℕ) (ρ : Representation ℂ (GL (Fin N) k) V) (σ : Representation ℂ (levi k b) W) (L : V →ₗ[ℂ] W) (hL : ∀ (p : blockTri (κ

@[simp] theorem **relRes_apply** (a c : Fin N → ℕ) (ρ : Representation ℂ (levi k a) V) (m : ↥(levi k a ⊓ levi k c)) (v : fixed ρ ((unipRadical k c).subgroupOf (levi k a))) : (relRes a c ρ m v : V) = ρ (m, m.2.1) v

@[simp] theorem **relInd_apply** (a c : Fin N → ℕ) (σ : Representation ℂ ↥(levi k a ⊓ levi k c) W) (g x : levi k a) (f : RelIndSpace a c σ) : (relInd a c σ g f : levi k a → W) x = (f : levi k a → W) (x * g)

@[simp] theorem **relTwist_apply** (a b : Fin N → ℕ) (w : Equiv.Perm (Fin N)) (σ : Representation ℂ ↥(levi k a ⊓ levi k (b ∘ w.symm)) W) (x : ↥(levi k b ⊓ levi k (a ∘ w))) : relTwist a b w σ x = σ (relConj a b w x)

@[simp] theorem **leviTensor_tprod** (e : (Σ a, Fin (d a)) ≃ Fin N) (ρ : ∀ a, Representation ℂ (GL (Fin (d a)) k) (Y a)) (m : levi k (blockOf e)) (v : ∀ a, Y a) : leviTensor e ρ m (PiTensorProduct.tprod ℂ v) = PiTensorProduct.tprod ℂ fun a => ρ a (leviEquiv e m a) (v a)

theorem **fixed_eq_bot_iff** {ρ : Representation ℂ (GL (Fin N) k) V} {U : Subgroup (GL (Fin N) k)} : fixed ρ U = ⊥ ↔ ∀ v : V, (∀ u ∈ U, ρ u v = v) → v = 0

theorem **isCuspidal_iff_forall** (ρ : Representation ℂ (GL (Fin N) k) V) : IsCuspidal ρ ↔ ∀ b : Fin N → ℕ, (∃ i j, b i ≠ b j) → fixed ρ (unipRadical k b) = ⊥

theorem **Embeds.isCuspidal** {ρ : Representation ℂ (GL (Fin N) k) V} {σ : Representation ℂ (GL (Fin N) k) W} (h : Embeds ρ σ) (hσ : IsCuspidal σ) : IsCuspidal ρ

theorem **HasCuspSupport.sum_fst** {κ : Representation ℂ (GL (Fin N) k) V} {s : Multiset (GradedClass k)} (h : HasCuspSupport κ s) : (s.map Sigma.fst).sum = N

theorem **HasCuspSupport.of_mem** {κ : Representation ℂ (GL (Fin N) k) V} {s : Multiset (GradedClass k)} (h : HasCuspSupport κ s) {n : ℕ} [FiniteDimensional ℂ W] {σ : Representation ℂ (GL (Fin n) k) W} (hσ : ((n, cls σ) : GradedClass k) ∈ s) : 0 < n ∧ σ.IsIrreducible ∧ IsCuspidal σ

theorem **HasCuspSupport.of_equiv** {κ : Representation ℂ (GL (Fin N) k) V} {κ' : Representation ℂ (GL (Fin N) k) W} {s : Multiset (GradedClass k)} (h : HasCuspSupport κ s) (e : κ.Equiv κ') : HasCuspSupport κ' s

theorem **hasCuspSupport_self** [FiniteDimensional ℂ V] (hN : 0 < N) {κ : Representation ℂ (GL (Fin N) k) V} (hi : κ.IsIrreducible) (hc : IsCuspidal κ) : HasCuspSupport κ {(N, cls κ)}

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsDiscreteUOfIsDiscrete

theorem **isDiscreteU_of_isDiscrete** {V : Submodule ℂ (Sm E a)} (h : IsDiscrete V) : IsDiscreteU V

theorem **inInd_iff** (h : ∑ i, n i = N) (π : ∀ i, CuspSub E (n i)) (φ : Sm E N) : InInd h (fun i => (π i).V) φ ↔ φ ∈ indForms h π

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsDualDualRep

theorem **mod_l_eq_one** {M n : ℕ} (hM : 1 ≤ M) (hn : n % (2 * X.l i ^ M) = 1) : n % X.l i = 1

theorem **FirstPlace.firstHalf** (h : FirstPlace i s) : FirstHalf i s

theorem **IsCj.unique** {g g' g'' : GE N} (h : IsCj X g g') (h' : IsCj X g g'') : g' = g''

theorem **isCj_one** : IsCj X (1 : GE N) 1

theorem **IsCj.mul** {g g' h h' : GE N} (hg : IsCj X g g') (hh : IsCj X h h') : IsCj X (g * h) (g' * h')

theorem **IsDual.unique** {ρ ρ' : GE N →* GL (Fin D.R.r) (X.O0 i)} (h : IsDual i s ρ) (h' : IsDual i s ρ') : ρ = ρ'

theorem **form_swap** {K : Type} [Field K] [Algebra (X.O0 i) K] [TopologicalSpace K] [DiscreteTopology K] {ρ ρ' : GE N →* GL (Fin D.R.r) (X.O0 i)} (x : Z1 (red i K ρ)) (y : Z1 (red i K ρ')) (g h g' h' : GE N) : form x y g' h' g h = −form x y g h g' h'

theorem **isDual_dualRep** : IsDual i s (dualRep i s)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsEmptyOfNotAll

theorem **elim** (a : TwJoin hL D nch xd first R F cG geo S YS cf) : R.Char3 F ∧ R.C = cf ∧ GlobalTheory.RealGeo.InGeo geo ∧ Formalism.L2.Host.ValLL0 D cf ∧ XdAll xd ∧ (∀ i j b (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (x : D.nexp.Fm i j b), x ∈ D.nexp.Bsup i j b ↔ YS.eF i j b h1 h2 x ∈ NX.Bsup S i j b) ∧ (∃ H : NExpRealH R cG S YS, GlobalTheory.ExportPin.Reads R cG S YS H.hL) ∧ Nonempty (ExportTie YS geo.v₀) ∧ (∀ g : G, cG (R.eG g) ∈ D.nexp.U ↔ g ∈ F.P0) ∧ 100 ≤ D.R.r

theorem **isEmpty_of_not_all** (h : ¬ XdAll xd) : IsEmpty (TwJoin hL D nch xd first R F cG geo S YS cf)

theorem **isEmpty_of_ne** (h : R.C ≠ cf) : IsEmpty (TwJoin hL D nch xd first R F cG geo S YS cf)

theorem **psi_of_grp** (h : GrpTie D A3.A F.P0 F.ψ0 F.P1 F.ψ1 (R.eG.trans cG)) (Z : ExportTie YS geo.v₀) (i b) (g : F.P0) (hg : cG (R.eG g) ∈ D.nexp.U) : Z.ρ.e i b (D.nexp.psiU i b (_, hg) : (FractionRing (D.ntw.O i))ˣ) = (D.what₂.embLL (F.ψ0 g : D.what₂.LLˣ))⁻¹

theorem **low_of_tied** {Fr : TwFrame D} (h : NTwTied D Fr) (hq : D.host.q ∣ (NumberField.discr D.host.E).natAbs) (i) (P : PlE D.host) (hP : ((D.host.QS * D.host.q * Ideal.absNorm D.host.ccK * (D.host.g 0 * D.host.g 1 * D.host.pde * D.host.pde') : ℕ) : 𝒪 D.host.E) ∈ P.asIdeal) : P ∈ D.ntw.S0 i

theorem **conj_of_lev** {P P' : HeightOneSpectrum (𝒪 D.host.E)} (h : Formalism.IndefR.Lev.IsConjPl P P') : Formalism.BasicNotions.IsConjPl D.host P P'

theorem **baseTup_of** {X : RealSide D} {E : IntegralHeckeForms.XD D} (Y : SerSide X E) (i : D.ntw.Tw) (t : X.T) (ht : X.R.xiH t = X.R.C.delAt (D.ntw.l i)) : BaseTup Y i

theorem **q_dvd_discr** (D : D7) : D.host.q ∣ (NumberField.discr D.host.E).natAbs

theorem **base_of_untw** {hH hV hs} (Y : SerSide (realOf t hH hV hs) (twEOf xd)) (hT : T = TwDat.untw B T.pl) (v : B.places.Pl) (hv : (w.D Z).what₂.muAt v ≠ (w.D Z).what₂.mu0) : BaseTup Y (nch.twI v)

theorem **wd_of_built** {D : D7} (b : NBuilt hL D) : ∃ wd : GlobalTheory.RowTie.WDTie D.nd D.naut, Nonempty (CGTrue hL D wd) ∧ LabTrue D wd

theorem **gal_of** (e : NE hL K B f T w Z Wt p0 t nch n xd c) (h3 : GlobalTheory.Triple.IsTriple (w.D Z)) (hr : 100 ≤ B.R.r) (hg : ∀ k, 2 * B.R.r ≤ B.host.g k) : ∃ g : GalSide (w.st Z).toD7, g.cf = TC.cf w.I w.l Z ∧ g.lam.C = TC.cf w.I w.l Z ∧ Nonempty (CGTrue hL (w.st Z).toD7 g.wd)

theorem **hk_of_built** {D : D7} (b : NBuilt hL D) : (∀ i j s k P (x : D.ntw.B i j s), 1 ≤ j → j ≤ D.R.m → IntegralHeckeForms.HkIdx i k P → D.nexp.embB i j s (D.ntw.hk i j s k P x) = D.nexp.hkF i j s k P (D.nexp.embB i j s x)) ∧ ∀ i (s : D.ntw.Sig i) P, D.ntw.hEig i s 0 P = 1

theorem **nonempty_of_more** {hH hV hs} (e : NE hL K B f T w Z Wt p0 t nch n xd c) (hF : FieldOk (realOf t hH hV hs)) (h3 : GlobalTheory.Triple.IsTriple (w.D Z)) (hr : 100 ≤ B.R.r) (hg : ∀ k, 2 * B.R.r ≤ B.host.g k) (H : NExpRealH t.r.z.R t.r.z.cG _ c.row.nexp) (hR : GlobalTheory.ExportPin.Reads t.r.z.R t.r.z.cG _ c.row.nexp H.hL) (hd : ∀ i b P, P ∉ (w.D Z).ntw.S0 i → ∀ σ : Formalism.ConjugateSelfDual.G (w.D Z).host.E (w.D Z).nd.Om, Formalism.ConjugateSelfDual.IsFrob (F

theorem **twOf_xdAll** (a : TwOf hL K B f T w Z Wt p0 t nch n xd c) : XdAll xd

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsHermPairSelf

theorem **one_mem** (N m : ℕ) : (1 : GL (Fin N) k) ∈ colSet k N m ∧ (1 : GL (Fin N) k) ∈ cornerSet k N m

theorem **isHermPair_self** {G : Type} [Group G] {π : Representation ℂ G V} (h : IsUnitarisable π) : IsHermPair π π

theorem **isAvg_bot** {X : Type} [AddCommGroup X] (u : Representation ℂ (Multiplicative X) V) : IsAvg u ⊥ (fun _ => 1) 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsHermitianOfMemRealScalars

@[simp] theorem **PhiGL_val** : (PhiGL n R).val = Phi n R

@[simp] theorem **PhiGL_inv_val** : (PhiGL n R)⁻¹.val = (Phi n R)ᵀ

theorem **Phi_inv** : (Phi n R)⁻¹ = (Phi n R)ᵀ

theorem **thetaGL_val** (g : GL (Fin n) R) : (thetaGL g).val = theta g.val

theorem **thetaGL_thetaGL** (g : GL (Fin n) R) : thetaGL (thetaGL g) = g

theorem **thetaGL_mul** (g h : GL (Fin n) R) : thetaGL (g * h) = thetaGL g * thetaGL h

theorem **star_PhiGL** : star (PhiGL n R) = (PhiGL n R)⁻¹

theorem **pred_le** (V : Submodule k (Fin n → k)) : F.pred V ≤ V

theorem **le_pred** {V V' : Submodule k (Fin n → k)} (h : V' ∈ F.carrier) (hlt : V' < V) : V' ≤ F.pred V

theorem **mem_lieQ** {X : Matrix (Fin n) (Fin n) k} : X ∈ F.lieQ ↔ ∀ V ∈ F.carrier, ∀ v ∈ V, X *ᵥ v ∈ V

theorem **mem_lieN** {X : Matrix (Fin n) (Fin n) k} : X ∈ F.lieN ↔ ∀ V ∈ F.carrier, ∀ v ∈ V, X *ᵥ v ∈ F.pred V

theorem **lieN_le_lieQ** : F.lieN ≤ F.lieQ

theorem **one_mem_lieQ** : (1 : Matrix (Fin n) (Fin n) k) ∈ F.lieQ

theorem **mul_mem_lieQ** {X Y : Matrix (Fin n) (Fin n) k} (hX : X ∈ F.lieQ) (hY : Y ∈ F.lieQ) : X * Y ∈ F.lieQ

theorem **mulVec_mem_pred** {X : Matrix (Fin n) (Fin n) k} (hX : X ∈ F.lieQ) (V : Submodule k (Fin n → k)) {v : Fin n → k} (hv : v ∈ F.pred V) : X *ᵥ v ∈ F.pred V

theorem **lieQ_mul_lieN** {X Y : Matrix (Fin n) (Fin n) k} (hX : X ∈ F.lieQ) (hY : Y ∈ F.lieN) : X * Y ∈ F.lieN

theorem **lieN_mul_lieQ** {X Y : Matrix (Fin n) (Fin n) k} (hX : X ∈ F.lieN) (hY : Y ∈ F.lieQ) : X * Y ∈ F.lieN

theorem **unip_le_stab** : F.unip ≤ F.stab

theorem **mem_stab_iff_map** {g : GL (Fin n) k} : g ∈ F.stab ↔ ∀ V ∈ F.carrier, V.map g.val.mulVecLin = V

theorem **conj_mem_lieN** {g : GL (Fin n) k} (hg : g ∈ F.stab) {X : Matrix (Fin n) (Fin n) k} (hX : X ∈ F.lieN) : g.val * X * g⁻¹.val ∈ F.lieN

theorem **conj_mem_unip** {g u : GL (Fin n) k} (hg : g ∈ F.stab) (hu : u ∈ F.unip) : g * u * g⁻¹ ∈ F.unip

theorem **lieQ_triv** : (triv n k).lieQ = ⊤

theorem **stab_triv** : (triv n k).stab = ⊤

theorem **mem_lieQ_single** {V : Submodule k (Fin n → k)} {X : Matrix (Fin n) (Fin n) k} : X ∈ (single V).lieQ ↔ ∀ v ∈ V, X *ᵥ v ∈ V

theorem **mem_stab_iff** {g : GL (Fin n) k} : g ∈ F.stab ↔ g.val ∈ F.lieQ

theorem **mem_unip_iff** {g : GL (Fin n) k} : g ∈ F.unip ↔ g.val − 1 ∈ F.lieN

theorem **isNilpotent_of_mem_lieN** {X : Matrix (Fin n) (Fin n) k} (hX : X ∈ F.lieN) : IsNilpotent X

theorem **pred_mem_and_lt** {V : Submodule k (Fin n → k)} (hV : V ≠ ⊥) : F.pred V ∈ F.carrier ∧ F.pred V < V

theorem **mem_lieN_iff_trace** {X : Matrix (Fin n) (Fin n) k} : X ∈ F.lieN ↔ ∀ Y ∈ F.lieQ, (X * Y).trace = 0

theorem **mem_lieN_iff_trace_stab** [Infinite k] {X : Matrix (Fin n) (Fin n) k} : X ∈ F.lieN ↔ ∀ g ∈ F.stab, (X * g.val).trace = 0

theorem **isClosed_lieQ** : IsClosed (F.lieQ : Set (Matrix (Fin n) (Fin n) ℂ))

theorem **isClosed_lieN** : IsClosed (F.lieN : Set (Matrix (Fin n) (Fin n) ℂ))

theorem **isClosed_stab** : IsClosed (F.stab : Set (GL (Fin n) ℂ))

theorem **isClosed_unip** : IsClosed (F.unip : Set (GL (Fin n) ℂ))

theorem **mulVec_dot_star** (X : Matrix (Fin n) (Fin n) ℂ) (v w : Fin n → ℂ) : (X *ᵥ v) ⬝ᵥ star w = v ⬝ᵥ star (Xᴴ *ᵥ w)

theorem **dot_star_comm_zero** {v w : Fin n → ℂ} (h : v ⬝ᵥ star w = 0) : w ⬝ᵥ star v = 0

theorem **perp_anti** {V V' : Submodule ℂ (Fin n → ℂ)} (h : V ≤ V') : perp V' ≤ perp V

theorem **eq_zero_of_dot_star** {v : Fin n → ℂ} (h : v ⬝ᵥ star v = 0) : v = 0

theorem **eq_zero_of_mem_perp** {V : Submodule ℂ (Fin n → ℂ)} {v : Fin n → ℂ} (h : v ∈ V) (h' : v ∈ perp V) : v = 0

theorem **sup_perp** (V : Submodule ℂ (Fin n → ℂ)) : V ⊔ perp V = ⊤

theorem **perp_perp** (V : Submodule ℂ (Fin n → ℂ)) : perp (perp V) = V

theorem **pred_sup_summand** (F : PFlag n ℂ) (V : Submodule ℂ (Fin n → ℂ)) : F.pred V ⊔ F.summand V = V

theorem **le_of_summand_le** {U V : Submodule ℂ (Fin n → ℂ)} (hV : V ∈ F.carrier) (h : ∀ V' ∈ F.carrier, V' ≤ V → F.summand V' ≤ U) : V ≤ U

theorem **iSup_summand** (F : PFlag n ℂ) : ⨆ V ∈ F.carrier, F.summand V = ⊤

theorem **summand_le_perp** {V V' : Submodule ℂ (Fin n → ℂ)} (hV : V ∈ F.carrier) (h : V < V') : F.summand V' ≤ perp V

theorem **mem_lieQ_and_star_iff** {X : Matrix (Fin n) (Fin n) ℂ} : X ∈ F.lieQ ∧ Xᴴ ∈ F.lieQ ↔ ∀ V ∈ F.carrier, ∀ v ∈ F.summand V, X *ᵥ v ∈ F.summand V

theorem **preserves_of_mem_realScalars** {X : Matrix (Fin n) (Fin n) ℂ} (hX : X ∈ F.realScalars) : ∀ V ∈ F.carrier, ∀ v ∈ F.summand V, X *ᵥ v ∈ F.summand V

theorem **isHermitian_of_mem_realScalars** {X : Matrix (Fin n) (Fin n) ℂ} (hX : X ∈ F.realScalars) : X.IsHermitian

theorem **commute_of_commute_separable** {n : ℕ} {A g h : Matrix (Fin n) (Fin n) ℂ} (hA : A.charpoly.Separable) (hg : Commute A g) (hh : Commute A h) : Commute g h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsLevelProjLevelProj

@[simp] theorem **galAct_apply** (s : k' ≃ₐ[k] k') {n : ℕ} (g : GL (Fin n) k') (i j : Fin n) : (galAct s n g : Matrix (Fin n) (Fin n) k') i j = s ((g : Matrix (Fin n) (Fin n) k') i j)

@[simp] theorem **galAct_one** (n : ℕ) : galAct (1 : k' ≃ₐ[k] k') n = MonoidHom.id _

theorem **galAct_mul** (s t : k' ≃ₐ[k] k') (n : ℕ) : galAct (s * t) n = (galAct s n).comp (galAct t n)

@[simp] theorem **incl_apply** {n : ℕ} (γ : GL (Fin n) k) (i j : Fin n) : (incl k' n γ : Matrix (Fin n) (Fin n) k') i j = algebraMap k k' ((γ : Matrix (Fin n) (Fin n) k) i j)

@[simp] theorem **galAct_incl** (s : k' ≃ₐ[k] k') {n : ℕ} (γ : GL (Fin n) k) : galAct s n (incl k' n γ) = incl k' n γ

theorem **isLevelProj_id** (π : Representation ℂ G V) (K : Subgroup G) : IsLevelProj π K LinearMap.id LinearMap.id

theorem **hasLevelTrace_id** (π : Representation ℂ G V) (K : Subgroup G) [FiniteDimensional ℂ (fixed π K)] : HasLevelTrace π K LinearMap.id (Module.finrank ℂ (fixed π K))

theorem **IsLevelProj.unique** (hd : Disjoint (fixed π K) (twKer π K 1)) {T T' : fixed π K →ₗ[ℂ] fixed π K} (hT : IsLevelProj π K A T) (hT' : IsLevelProj π K A T') : T = T'

theorem **HasLevelTrace.unique** (hd : Disjoint (fixed π K) (twKer π K 1)) {c c' : ℂ} (h : HasLevelTrace π K A c) (h' : HasLevelTrace π K A c') : c = c'

theorem **IsTwInt.isUnit** {s : k' ≃ₐ[k] k'} {P : Representation ℂ (GL (Fin n) k') W} {I : W →ₗ[ℂ] W} (hI : IsTwInt k s P I) : IsUnit I

theorem **IsTwInt.conj** {s : k' ≃ₐ[k] k'} {P : Representation ℂ (GL (Fin n) k') W} {I : W →ₗ[ℂ] W} (hI : IsTwInt k s P I) (g : GL (Fin n) k') : P (galAct s n g) = I * P g * ((hI.isUnit.unit⁻¹ : (W →ₗ[ℂ] W)ˣ) : W →ₗ[ℂ] W)

theorem **IsTwInt.mem_fixed** {s : k' ≃ₐ[k] k'} {P : Representation ℂ (GL (Fin n) k') W} {I : W →ₗ[ℂ] W} (hI : IsTwInt k s P I) {K : Subgroup (GL (Fin n) k')} (hK : IsStable s K) {v : W} (hv : v ∈ fixed P K) : I v ∈ fixed P K

theorem **isTwInt_of_triv** (s : k' ≃ₐ[k] k') (hd : 0 < Module.finrank k k') {P : Representation ℂ (GL (Fin n) k') ℂ} (hP : ∀ g, P g = LinearMap.id) : IsTwInt k s P LinearMap.id

theorem **hasLevelTrace_of_triv** {G : Type} [Group G] {P : Representation ℂ G ℂ} (hP : ∀ g, P g = LinearMap.id) (K : Subgroup G) : HasLevelTrace P K LinearMap.id 1

theorem **charVal_of_triv** {P : Representation ℂ (GL (Fin n) k) ℂ} (hP : ∀ g, P g = LinearMap.id) (γ : GL (Fin n) k) : CharVal P γ 1

theorem **twCharVal_of_triv** (s : k' ≃ₐ[k] k') {P : Representation ℂ (GL (Fin n) k') ℂ} (hP : ∀ g, P g = LinearMap.id) (g' : GL (Fin n) k') : TwCharVal k s P LinearMap.id g' 1

theorem **ofChar_one_apply** {G : Type} [Group G] (g : G) : SmGL.ofChar (1 : G →* ℂˣ) g = LinearMap.id

theorem **isLBC_one** (s : k' ≃ₐ[k] k') (hd : 0 < Module.finrank k k') : IsLBC s (SmGL.ofChar (1 : GL (Fin n) k →* ℂˣ)) (SmGL.ofChar (1 : GL (Fin n) k' →* ℂˣ))

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsPlaceModelPlaceMap

theorem **isDetRoot_detRoot** (V : HSp D) (g : GA D V) : IsDetRoot V g (detRoot V g)

theorem **detChar_eq** {V : HSp D} {χ : (AE D)ˣ →* A} (hF : ∀ a, χ (idF D a) = 1) {g : GA D V} {z : (AE D)ˣ} (hz : IsDetRoot V g z) : detChar V χ g = χ z

theorem **detChar_mul** {V : HSp D} {χ : (AE D)ˣ →* A} (hF : ∀ a, χ (idF D a) = 1) (g g' : GA D V) : detChar V χ (g * g') = detChar V χ g * detChar V χ g'

theorem **detChar_rat** {V : HSp D} {χ : (AE D)ˣ →* A} (hF : ∀ a, χ (idF D a) = 1) (hE : ∀ x : D.host.Eˣ, χ (Units.map (incA D).toMonoidHom x) = 1) {γ : GA D V} (hγ : γ ∈ GQ V) : detChar V χ γ = 1

@[simp] theorem **detChar_one** (V : HSp D) (g : GA D V) : detChar V (1 : (AE D)ˣ →* A) g = 1

theorem **muVal_detChar** (V : HSp D) {χ : (AE D)ˣ →* Circle} (hF : ∀ a, χ (idF D a) = 1) : Formalism.L2.Pj.MuVal V χ fun g => ((detChar V χ g : Circle) : ℂ)

theorem **twistBy_apply** (V : HSp D) (χ : (AE D)ˣ →* Circle) (φ : GA D V → ℂ) (g : GA D V) : twistBy V χ φ g = ((detChar V χ g : Circle) : ℂ) * φ g

@[simp] theorem **twistBy_one** (V : HSp D) (φ : GA D V → ℂ) : twistBy V 1 φ = φ

theorem **idemAt_eq** {e : AE D} {PP : Pl D.host.E} (h : IsIdemAt e PP) : idemAt PP = e

theorem **idem_eq** {e : AE D} {P : Finset (Pl D.host.E)} (h : IsIdemSum e P) : idem P = e

theorem **SplitSet.isIdemSum** {P : Finset (Pl D.host.E)} (h : SplitSet P) : IsIdemSum (idem P) P

theorem **SplitSet.isIdemAt** {P : Finset (Pl D.host.E)} (h : SplitSet P) {PP : Pl D.host.E} (hPP : PP ∈ P) : IsIdemAt (idemAt PP) PP

@[simp] theorem **idem_empty** : idem (∅ : Finset (Pl D.host.E)) = 0

theorem **splitSet_empty** : SplitSet (∅ : Finset (Pl D.host.E))

theorem **SplitSet.mul_eq_zero** {P : Finset (Pl D.host.E)} (h : SplitSet P) {i j : Pl D.host.E} (hi : i ∈ P) (hj : j ∈ P) (hij : j ≠ i) : idemAt i * idemAt j = 0

theorem **SplitSet.mul_conj** {P : Finset (Pl D.host.E)} (h : SplitSet P) {i j : Pl D.host.E} (hi : i ∈ P) (hj : j ∈ P) : idemAt i * cA D (idemAt j) = 0

theorem **SplitSet.mono** {P P' : Finset (Pl D.host.E)} (h : SplitSet P) (hP : P' ⊆ P) : SplitSet P'

theorem **Twist.triv_good** {Q : Finset (Pl D.host.E)} (h : SplitSet Q) : (Twist.triv A Q).Good where

theorem **Twist.Good.isHeckeChar** {t : Twist D Circle} (h : t.Good) : IsHeckeChar D t.chi

theorem **UnivTwist.good_empty** (u : UnivTwist D C) : u.Good ∅

theorem **UnivTwist.Good.mono** {u : UnivTwist D C} {Q Q' : Finset (Pl D.host.E)} (h : u.Good Q) (hQ : Q' ⊆ Q) : u.Good Q'

theorem **UnivTwist.Good.delQ_real** {u : UnivTwist D C} {Q : Finset (Pl D.host.E)} (h : u.Good Q) (a : (AF D.host.F)ˣ) : u.delQ Q (idF D a) = 1

theorem **PlacesOK.mono** {Pg Q Q' : Finset (Pl D.host.E)} (h : PlacesOK Pg Q) (hQ : Q' ⊆ Q) : PlacesOK Pg Q'

theorem **placesOK_empty** : PlacesOK (∅ : Finset (Pl D.host.E)) ∅

theorem **datumFn_add** (Q : Finset (Pl D.host.E)) (ω ω' : Reds D T.rank S.rank → ℂ) : F.datumFn G Q (ω + ω') = F.datumFn G Q ω + F.datumFn G Q ω'

theorem **datumFn_smul** (Q : Finset (Pl D.host.E)) (c : ℂ) (ω : Reds D T.rank S.rank → ℂ) : F.datumFn G Q (c • ω) = c • F.datumFn G Q ω

theorem **PlacesOK.isOrthSum** {Pg Q : Finset (Pl D.host.E)} (h : PlacesOK Pg Q) : GlobalTheory.PlaceModelSum.IsOrthSum (halfIdem Pg Q) ∧ halfIdem Pg Q * cA D (halfIdem Pg Q) = 0

theorem **isSBHalf_mul** {m n : ℕ} {e : AE D} {Λ : AddSubgroup (Mx D m n)} {Φ Φ' : Mx D m n → ℂ} (h : IsSBHalf e Λ Φ) (h' : IsSBHalf e Λ Φ') : IsSBHalf e Λ fun x => Φ x * Φ' x

theorem **isPlaceModel_placeMap** (hF : F.Good) (h : PlacesOK G.Pg Q) : IsPlaceModel F.ψ (symplO S T) (heisO F.ψ S T) (halfIdem G.Pg Q) F.lat 1 F.f₀ (F.placeMap (halfIdem G.Pg Q))

theorem **isSBHalf_datumFn** (hF : F.Good) (h : PlacesOK G.Pg Q) (hω : ∀ y y', (∀ PP ∈ Q, y PP = y' PP) → ω y = ω y') : IsSBHalf (halfIdem G.Pg Q) F.lat (F.datumFn G Q ω)

theorem **isThetaAt_vec** (hF : F.Good) (h : PlacesOK G.Pg Q) (hω : ∀ y y', (∀ PP ∈ Q, y PP = y' PP) → ω y = ω y') : IsThetaAt F.ψ (symplO S T) (heisO F.ψ S T) (halfIdem G.Pg Q) F.lat 1 F.f₀ (F.datumFn G Q ω) (F.vec G Q ω)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsPotentialPhiSub

theorem **xi_unit** (d : (unitMod l a e O K ι).D) : (unitMod l a e O K ι).xi d = cst d

theorem **nabla_bas** (i : ι) : (unitMod l a e O K ι).nabla (bas i) = 0

theorem **phi_bas_unit** (i : ι) : (unitMod l a e O K ι).phi (bas i) = bas i

theorem **unitDesc_act** (γ : TameGal K l e a) (v : Vec K a ι) (τ : ZMod a) (i : ι) : (unitDesc l a e O K ι).act γ v τ i = ringAct γ (fun σ => v σ i) τ

theorem **galOW_iW_of_covers** {h : GQ l} {γ : TameGal K l e a} (hc : Covers K t ϖ h γ) (f f' : ZMod a → invE l e (A O K)) (hf : ∀ τ, (f' τ : K⟦X⟧) = ringAct γ (fun σ => (f σ : K⟦X⟧)) τ) : galOW O h (ϖ.iW t K f) = ϖ.iW t K f'

theorem **galOB_iB_const** {h : GQ l} (hh : h ∈ GK (ϖ.field a)) (f : ZMod a → invL l (S O K e)) (c : ZMod a → K) (hf : ∀ τ, (f τ : K⟦X⟧) = C (c τ)) : galOB O h (ϖ.iB t K f) = ϖ.iB t K f

theorem **coordEq_apply** (R : Type) [CommRing R] [Algebra O R] (T : LocLat G 𝕆_[l] K O) (b : Module.Basis ι O ɪT.L.T) (v : ι → R) : coordEq R T b v = ∑ i, v i ⊗ₜ[O] b i

theorem **isCrisVal_unique** {P : PhiMod O K l e a ι} {T : LocLat G 𝕆_[l] K O} {jB : (ι → OB l O) ≃ₗ[OB l O] TW (OB l O) T} {w : ι → OW l O} {b b' : TW (OB l O) T} (h : IsCrisVal t ϖ P jB w b) (h' : IsCrisVal t ϖ P jB w b') : b = b'

theorem **isPotential_phi_sub** {s : G →* GQ l} {η H : ℕ} {D : BKDat O K l e a ι η H} {T : LocLat G 𝕆_[l] K O} (R : GalDat t ϖ s D T) {m : Vec K a ι} (hm : m ∈ D.Minv) : R.IsPotential (D.phi m − m) 0

theorem **classOf_phi_sub** {s : G →* GQ l} {η H : ℕ} {D : BKDat O K l e a ι η H} {T : LocLat G 𝕆_[l] K O} (R : GalDat t ϖ s D T) {m : Vec K a ι} (hm : m ∈ D.Minv) : R.ClassOf (D.phi m − m) 0

theorem **covers_one** {g : GQ l} (hg : g ∈ ϖ.GfieldInf a) : Covers K t ϖ g (1 : TameGal K l e a)

theorem **fix_Kur** {F : FinExt l} {g : ɪ(GK F)} (hg : g ∈ IK F) : (g : GQ l) ∈ (MaximalUnramifiedExtension l).fixingSubgroup

theorem **isGood_of_trivial** (Pl : LocPrime l) {LL : FinExt l} (T : QLat LL) (hT : ∀ g x, T.ρ g x = x) : IsGood a ϖ Pl T {0} {0}

theorem **finrank_unitLat** : Module.finrank O ɪ(unitLat G : LocLat G 𝕆_[l] K O).L.T = 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsProjBot

theorem **isProj_bot** {X : Type} [AddCommGroup X] (u : Representation ℂ (Multiplicative X) V) : IsProj u ⊥ (fun _ => 1) 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsPsCharOne

theorem **map_alt** {B : Type*} [CommRing B] (f : A →+* B) (T : M → A) : ∀ (n : ℕ) (x : Fin n → M), f (alt T n x) = alt (fun g => f (T g)) n x | 0, _ => by simp [alt] | n + 1, x => by simp [alt, map_alt f T n] end GlobalTheory.PsChar

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsRatScalarMulIff

theorem **isScale_iff** (v₀ : ∀ (j : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), (Rl.MW j σ).Vec) {j : Fin D.R.m} (B : Base (Vc D j) (Wc D j)) (c : ℂ) : IsScale Rl v₀ B c ↔ ∃ gramR c0R c0L, Bridge.FamTie Rl.B Rl.pl (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E)) Rl.Q (labLL D) (fun τ => Bridge.lamRdef (Ptw D) τ) (fun j τ => Bridge.lamLdef (Vc D j).rank ((j : ℕ) + 1) τ) gramR c0R c0L ∧ ∃ u : ∀ τ, lowT (Rl.MW j (sg0 D A3.A)) τ, (∀ τ, (u τ : (Rl.MW j (sg0 D A3.A)).sm τ) = v₀ j _ τ) ∧ IsScaleAt Rl B u (c0L j) c

theorem **IsScaleAt.ioP_eq** {j : Fin D.R.m} {B : Base (Vc D j) (Wc D j)} {u : ∀ τ, lowT (Rl.MW j (sg0 D A3.A)) τ} {c0 : VsV D j D.what₂.LL →ₛₗ[D.what₂.embLL] (Rl.cV j).W} {c : ℂ} (h : IsScaleAt Rl B u c0 c) {H : Submodule ℂ (PolF D (Wc D j).rank (Vc D j).rank)} (hinj : ∀ P ∈ H, harmFn (Wc D j).rank B.w B.F.BS P = 0 → P = 0) {F : VV D.what₂.LL (FWT.lamV D j)} {P : PolF D (Wc D j).rank (Vc D j).rank} (hP : P ∈ H) (hio : Rl.ioP j (PiTensorProduct.tprod ℂ u) (c0 F) ∈ H) (hPF : harmFn (Wc D j).rank B.w B.F.BS P = coefFn D.what₂.embLL (FWT.lamV D j) (F : Formalism.L2.DIV.Pol D.what₂.LL D.what₂.Tau (Vc D j).rank)) : Rl.ioP j (PiTensorProduct.tprod ℂ u) (c0 F) = c • P

theorem **isNorm_of_root** {v₀ : ∀ (j : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), (Rl.MW j σ).Vec} {j : Fin D.R.m} {B : Base (Vc D j) (Wc D j)} {c ζ : ℂ} (hc : IsScale Rl v₀ B c) {n : ℕ} (hn : n ≠ 0) (hζ : ζ ^ n = 1) (h0 : ζ * cj (twAt D.what₂.embLL A3.A.v0) (Rl.phs j) * c ≠ 0) : IsNorm Rl v₀ B (ζ * cj (twAt D.what₂.embLL A3.A.v0) (Rl.phs j) * c)

theorem **RealPin.phase_mul_thetaKO** {cG : G0 Rl.Pg D.R.r ≃* G0g D.nd} {ρ : Setup D.ntw} (h : RealPin Rl cG ρ) (i : D.ntw.Tw) {j : Fin D.R.m} (b : Bool) {u : ∀ τ, lowT (Rl.MW j (sg0 D A3.A)) τ} {w : (Rl.cV j).W} {P : PolF D (Wc D j).rank (Vc D j).rank} {c : ℂ} (hP : Rl.ioP j (PiTensorProduct.tprod ℂ u) w = c • P) (ph : ℂ) (om : PairRep (Vc D j) (Wc D j)) (s : GA D (Vc D j)) (f : WO (Vc D j) (Wc D j) → ℂ) : (fun g => ph * thetaKO om s g (harmVec Rl.toRealDat j u w f)) = (ph * c) • series (ρ.B i j b).F om f s P

theorem **isRat_mul** {C : Cover T ((j : ℕ) + 1) σ} {κ : WeightFamilies.Wt D ((j : ℕ) + 1)} {φ : GA D (Wc D j) → ℂ} (h : IsRat T j hV C κ φ) {a : ℂ} (ha : a ∈ C.σ₂.fieldRange) : IsRat T j hV C κ fun g => a * φ g

theorem **seriesRatOn_mul** {C : Cover T ((j : ℕ) + 1) σ} {κ : WeightFamilies.Wt D ((j : ℕ) + 1)} {u : ∀ τ, lowT (R.MW j (sg0 D A)) τ} {c0 : VsV D j D.what₂.LL →ₛₗ[D.what₂.embLL] (R.cV j).W} {ph : ℂ} (h : SeriesRatOn T R j hV C κ u c0 ph) {a : ℂ} (ha : a ∈ C.σ₂.fieldRange) : SeriesRatOn T R j hV C κ u c0 (a * ph)

theorem **IsRatPhase.mul** {κ : WeightFamilies.Wt D ((j : ℕ) + 1)} {u : ∀ τ, lowT (R.MW j (sg0 D A)) τ} {c0 : VsV D j D.what₂.LL →ₛₗ[D.what₂.embLL] (R.cV j).W} {ph : ℂ} (h : IsRatPhase T R j hV κ u c0 ph) {a : ℂ} (ha : a ∈ (embLL0 D).fieldRange) : IsRatPhase T R j hV κ u c0 (a * ph)

theorem **isRatScalar_iff** (ν : ℂ) : IsRatScalar T Rl κ v₀ hV B ν ↔ ∃ gramR c0R c0L, Bridge.FamTie Rl.B Rl.pl (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E)) Rl.Q (labLL D) (fun τ => Bridge.lamRdef (Ptw D) τ) (fun j τ => Bridge.lamLdef (Vc D j).rank ((j : ℕ) + 1) τ) gramR c0R c0L ∧ ∃ u : ∀ τ, lowT (Rl.MW j (sg0 D A3.A)) τ, (∀ τ, (u τ : (Rl.MW j (sg0 D A3.A)).sm τ) = v₀ j _ τ) ∧ ∃ c ph : ℂ, c ≠ 0 ∧ ‖ph‖ = 1 ∧ ν = ph * c ∧ (∀ F : VV D.what₂.LL (FWT.lamV D j), harmFn (Wc D j).rank B.w B.F.BS (Rl.ioP j (PiTensorProduct.tprod ℂ u) (c0L j F)) = c • coefFn D.what₂.embLL (FWT.lamV D j) (F : Formalism.L2.DIV.Pol D.what₂.LL D.what₂.Tau (Vc D j).rank)) ∧ ∀ C : Cover T

((j : ℕ) + 1) T.σ₀, GoodCover T Rl.toRealDat j C (κ j).n → (embLL0 D).fieldRange ≤ C.σ₂.fieldRange → SeriesRatOn T Rl.toRealDat j hV C (κ j) u (c0L j) ph

theorem **IsRatScalar.intro** {gramR c0R c0L} (h : Bridge.FamTie Rl.B Rl.pl (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E)) Rl.Q (labLL D) (fun τ => Bridge.lamRdef (Ptw D) τ) (fun j τ => Bridge.lamLdef (Vc D j).rank ((j : ℕ) + 1) τ) gramR c0R c0L) {u : ∀ τ, lowT (Rl.MW j (sg0 D A3.A)) τ} (hu : ∀ τ, (u τ : (Rl.MW j (sg0 D A3.A)).sm τ) = v₀ j _ τ) {c ph : ℂ} (hc : c ≠ 0) (hph : ‖ph‖ = 1) (hs : IsScaleAt Rl B u (c0L j) c) (hr : IsRatPhase T Rl.toRealDat j hV (κ j) u (c0L j) ph) : IsRatScalar T Rl κ v₀ hV B (ph ∗ c)

theorem **IsRatScalar.isScale** {ν : ℂ} (h : IsRatScalar T Rl κ v₀ hV B ν) : ∃ c ph : ℂ, c ≠ 0 ∧ ‖ph‖ = 1 ∧ ν = ph ∗ c ∧ IsScale Rl v₀ B c

theorem **IsRatScalar.ne_zero** {ν : ℂ} (h : IsRatScalar T Rl κ v₀ hV B ν) : ν ≠ 0

theorem **IsRatScalar.mul** {ν : ℂ} (h : IsRatScalar T Rl κ v₀ hV B ν) {a : ℂ} (ha : a ∈ (embLL0 D).fieldRange) (h0 : ‖a‖ = 1) : IsRatScalar T Rl κ v₀ hV B (a ∗ ν)

theorem **IsRatScalar.mul_right** {ν : ℂ} (h : IsRatScalar T Rl κ v₀ hV B ν) {a : ℂ} (ha : a ∈ (embLL0 D).fieldRange) (h0 : ‖a‖ = 1) : IsRatScalar T Rl κ v₀ hV B (ν ∗ a)

theorem **isRatScalar_mul_iff** {ν a : ℂ} (ha : a ∈ (embLL0 D).fieldRange) (h0 : ‖a‖ = 1) : IsRatScalar T Rl κ v₀ hV B (a ∗ ν) ↔ IsRatScalar T Rl κ v₀ hV B ν

theorem **exists_isRatPhase** (row : SeriesRat) {F : Face D A3.A} (hT : LawsJ T) (h0 : Rl.toRealDat.toLevCh.Is A3.A.μ0) (hg : Rl.toRealDat.IsGrp F) (hf : Rl.toRealDat.IsFam) (hm : Rl.toRealDat.IsModels) (ht : Rl.toRealDat.IsThP) (hi : TieIn T Rl.toRealDat κ v₀) (hp : GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalLatticeThetaValues.FramePin T Rl.toRealDat j T.σ₀ hV) (hstd : VStd D j) {gramR c0R c0L} (h : Bridge.FamTie Rl.B Rl.pl (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E)) Rl.Q (labLL D) (fun τ => Bridge.lamRdef (Ptw D) τ) (fun j τ => Bridge.lamLdef (Vc D j).rank ((j : ℕ) + 1) τ) gramR c0R c0L) {u : ∀ τ, lowT (Rl.MW j (sg0 D A3.A)) τ} (hu : ∀ τ, (u τ : (Rl.MW j (sg0 D A3.A)).sm τ) = v₀ j _ τ) : ∃ ph : ℂ, ‖ph‖ = 1 ∧ IsRatPhase T Rl.toRealDat j hV (κ j) u (c0L j) ph

theorem **exists_isRatScalar** (row : SeriesRat) {F : Face D A3.A} (hT : LawsJ T) (h0 : Rl.toRealDat.toLevCh.Is A3.A.μ0) (hg : Rl.toRealDat.IsGrp F) (hf : Rl.toRealDat.IsFam) (hm : Rl.toRealDat.IsModels) (ht : Rl.toRealDat.IsThP) (hi : TieIn T Rl.toRealDat κ v₀) (hp : GlobalTheory.IndefiniteRowSideFullHecke.OrthonormalLatticeThetaValues.FramePin T Rl.toRealDat j T.σ₀ hV) (hstd : VStd D j) {c : ℂ} (hc : c ≠ 0) (hs : IsScale Rl v₀ B c) : ∃ ph : ℂ, ‖ph‖ = 1 ∧ IsRatScalar T Rl κ v₀ hV B (ph ∗ c)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsRecTOfIsRec

theorem **isRecT_of_isRec** (hT : GlobalTheory.Published.DepthZeroTypeInertia) {rc : IrrGL (w.adicCompletion E) → WD h} (hrec : IsRec h rc) : IsRecT h rc

theorem **recTExists_of** (hL : GlobalTheory.Published.LocalLanglandsGL) (hT : GlobalTheory.Published.DepthZeroTypeInertia) : RecTExists

theorem **good_of_toral** {x : IrrGL k} (hx : Toral x) : Good x

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsRegularOne

theorem **shift_univ** (deg : M →* Multiplicative (ι → ℤ)) (ρ : Representation ℂ M V) (μ : ι → ℤ) (g : M) : shift V μ ∘ₗ univ deg ρ g = univ deg ρ g ∘ₗ shift V μ

theorem **isRegular_one** (n : ℕ) : IsRegular k n fun _ => 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsRhoPtOfSplitting

theorem **locE_apply** {k : ℕ} (M : Formalism.MainNotions.N.GE D.nd →* Matrix (Fin k) (Fin k) (D.nd.O i)) (g : Br.G D.nbr i w hw) (a b : Fin k) : locE D i w hw L' h M g a b = inc D i w hw L' h (Br.loc D.nbr i w hw M g a b)

theorem **chiE_apply** (χ : Formalism.MainNotions.N.GE D.nd →* (D.nd.O i)ˣ) (g : Br.G D.nbr i w hw) : ((chiE D i w hw L' h χ g : (ɪL'.1)ˣ) : ɪL'.1) = inc D i w hw L' h ((Br.locChi D.nbr i w hw χ g : (Br.LL D.nbr i w hw)ˣ) : Br.LL D.nbr i w hw)

theorem **nonempty_embE** : Nonempty (Br.Kw D.nbr i w hw →ₐ[ℚ_[D.nd.l i]] ɪL'.1)

theorem **EpsAll.exists_generator** {D : D7} {i : D.nd.Loc} (h : EpsAll D i) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : w ∈ placesOver D.host.E (D.nd.l i)) : ∃ g ∈ Ine (F

theorem **attachedIn_self** (a : D.nd.Loc) (π : D.nd.X a) (s : D.ntw.Sig (D.ntw.twOf a)) : AttachedIn D a π _ s ↔ GlobalTheory.IndefiniteRowSideFullHecke.LatticeConditions.Attached D a π s

theorem **AttachedIn.jS_side** {D : D7} {a : D.nd.Loc} {π : D.nd.X a} {j : D.ntw.Tw} {s : D.ntw.Sig j} (h : AttachedIn D a π j s) : D.ntw.jS j s = D.nd.j a ∧ D.ntw.side j s = D.ntw.twSide a

theorem **SelCore.bound** {D : D7} {C : NChoice D} {ρ : ∀ i : D.nd.Loc, D.nd.X i → (Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.r) (D.nd.O i))} (h : SelCore D C ρ) (i : D.nd.Loc) (b : ℝ) (hb : 0 ≤ b) (hs : ∀ s : D.ntw.Sig (C.twI (D.nd.w i)), BlochKatoReduction.Gen.goodS (BlochKatoReduction.bk0 D) (C.twI (D.nd.w i)) s → D.ntw.jS _ s = D.nd.j i → D.ntw.side _ s = C.sd (D.nd.w i) → BlochKatoReduction.Gen.sLen (BlochKatoReduction.bk0 D) (C.twI (D.nd.w i)) s (C.k' (D.nd.w i)) (C.Q (D.nd.w i)) (psiG D C (D.nd.w i)) ≤ ENNReal.ofReal b) : ∀ π ∈ D.nd.Zgood i, Br.sFin D.loc D.nbr i (ρ i) π ∧ (Br.sPi D.loc D.nbr i (ρ i) π : ℝ) ≤ b

theorem **GoodPtRow.exists_attached** {D : D7} (hG : GoodPtRow D) (hT : TriplesGood D) (a : D.nd.Loc) (π : D.nd.X a) (hπ : π ∈ D.nd.Zgood a) : ∃ s, GlobalTheory.IndefiniteRowSideFullHecke.LatticeConditions.Attached D a π s ∧ BlochKatoReduction.Gen.goodS (BlochKatoReduction.bk0 D) (D.ntw.twOf a) s

theorem **sameTwistAt_iff** (a : D.nd.Loc) (j : D.ntw.Tw) (k : ℕ) (ψ : Formalism.MainNotions.N.GE D.nd →* (D.ntw.Ok j k)ˣ) : SameTwistAt D a j k ψ ↔ D.ntw.twOf a = j ∧ ∃ f : D.ntw.O0 j →+* D.nd.O a, HEq f (D.ntw.twist₂ a) ∧ GlobalTheory.IndefiniteRowSideFullHecke.LatticeConditions.SameChar f (BlochKatoReduction.Gen.phiK D.ntw j k) (D.ntw.psiL a) ψ

theorem **SameTwist.twOf_eq** {D : D7} {C : NChoice D} (h : SameTwist D C) (a : D.nd.Loc) : D.ntw.twOf a = C.twI (D.nd.w a)

theorem **SameTwist.exists_heq** {D : D7} {C : NChoice D} (h : SameTwist D C) (a : D.nd.Loc) : ∃ f : D.ntw.O0 (C.twI (D.nd.w a)) →+* D.nd.O a, HEq f (D.ntw.twist₂ a) ∧ GlobalTheory.IndefiniteRowSideFullHecke.LatticeConditions.SameChar f (BlochKatoReduction.Gen.phiK D.ntw (C.twI (D.nd.w a)) (C.k' (D.nd.w a))) (D.ntw.psiL a) (psiG D C (D.nd.w a))

theorem **isRhoPt_of_splitting** (i : D.nd.Loc) (π : D.nd.X i) (r : Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.nV) (D.nd.O i)) (ρ : Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.r) (D.nd.O i)) (hr : IsHeckeGalois D.nd i π r) (h : BlockRepresentationsSemisimple.IsComplement D.nd i r ρ) : LatticeConditions.IsRhoPt D i π ρ

theorem **RhoFrob.of_splitting** (i : D.nd.Loc) (π : D.nd.X i) (r : Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.nV) (D.nd.O i)) (ρ : Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.r) (D.nd.O i)) (hr : IsHeckeGalois D.nd i π r) (h : BlockRepresentationsSemisimple.IsComplement D.nd i r ρ) : RhoFrob D i π ρ

theorem **multV_cycPow_line** [Algebra.IsSeparable Qp K] [FiniteDimensional Qp LL] (hF : IsField BdR) (hs : ∃ s ∈ PD.Fil (-1), PD.ι PD.t * s = 1) (σ : Representation LL G (Fin 1 → LL)) (e : ℕ) (h : ∀ g u, σ g u 0 = ((algebraMap Qp LL (PD.χ g : Qp))⁻¹) ^ e * u 0) (τ : K →ₐ[Qp] LL) (a : ℤ) : multV PD σ τ a = if a = e then 1 else 0

theorem **BlockE.of_splitting** (i : D.nd.Loc) (r : D.nd.X i → (Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.nV) (D.nd.O i))) (ρ : D.nd.X i → (Formalism.MainNotions.N.GE D.nd →* GL (Fin D.R.r) (D.nd.O i))) (π : D.nd.X i) (hc : BlockRepresentationsSemisimple.IsComplement D.nd i (r π) (ρ π)) : BlockE D i r ρ π

theorem **frob** (h : PtFam D r ρ) : ∀ i, ∀ π ∈ D.nd.Z i, RhoFrob D i π (ρ i π)

theorem **pt** (h : PtFam D r ρ) : ∀ i, ∀ π ∈ D.nd.Z i, GlobalTheory.IndefiniteRowSideFullHecke.LatticeConditions.IsRhoPt D i π (ρ i π)

theorem **block** (h : PtFam D r ρ) : ∀ i, ∀ π ∈ D.nd.Z i, BlockE D i (r i) (ρ i) π

theorem **exists_extend** (hC : GlobalTheory.IndefiniteRowSideFullHecke.BlockRepresentationsSemisimple.Complement_at D) (hr : ∀ i, GoodR D.nd i (r i)) : ∃ ρ, PtFam D r ρ

theorem **exists_pair** (hC : GlobalTheory.IndefiniteRowSideFullHecke.BlockRepresentationsSemisimple.Complement_at D) (hr : ∀ i, ∃ r, GoodR D.nd i r) : ∃ r ρ, PtFam D r ρ

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsSplitOfMemAdmissibleConjugatePairPlaces

theorem **exists_recip_good** (h : GlobalTheory.Published.GlobalArtinReciprocity) (D : D7) (Ω : Type) [Field Ω] [Algebra D.host.E Ω] [IsAlgClosure D.host.E Ω] : ∃ a : Recip D Ω, a.Good

theorem **descend_toAb** {φ : G D.host.E Ω →* A} (h : IsOpen (φ.ker : Set (G D.host.E Ω))) (σ : G D.host.E Ω) : descend φ (toAb D.host.E Ω σ) = φ σ

theorem **Recip.pull_eq** (a : Recip D Ω) {φ : G D.host.E Ω →* A} (h : IsOpen (φ.ker : Set (G D.host.E Ω))) {x : (AE D)ˣ} {σ : G D.host.E Ω} (hx : a.sym x = toAb D.host.E Ω σ) : a.pull φ x = φ σ

theorem **Recip.exists_sym** (a : Recip D Ω) (x : (AE D)ˣ) : ∃ σ, a.sym x = toAb D.host.E Ω σ

@[simp] theorem **Recip.pull_one** (a : Recip D Ω) : a.pull (1 : G D.host.E Ω →* A) = 1

theorem **Recip.pull_comp** (a : Recip D Ω) {A' : Type} [CommGroup A'] {φ : G D.host.E Ω →* A} (h : IsOpen (φ.ker : Set (G D.host.E Ω))) (f : A →* A') : a.pull (f.comp φ) = f.comp (a.pull φ)

theorem **unitPart_of_pow** {z : ℂˣ} {n : ℕ} (hn : n ≠ 0) (h : z ^ n = 1) : ((unitPart z : Circle) : ℂ) = z

theorem **isResAt_resOf** {A : Type} [CommGroup A] {χ : (AE D)ˣ →* A} {P : Pl D.host.E} (h : ∃ c, IsResAt χ P c) : IsResAt χ P (resOf χ P)

theorem **mem_half** {ch : Set (Pl D.host.E)} {Q : Finset (Pl D.host.E)} {P : Pl D.host.E} : P ∈ half ch Q ↔ P ∈ Q ∧ P ∈ ch

@[simp] theorem **half_empty** (ch : Set (Pl D.host.E)) : half ch ∅ = ∅

theorem **charOf_eq** {ψ : G D.host.E Ω →* Rˣ} (h : IsOpen (ψ.ker : Set (G D.host.E Ω))) {x : (AE D)ˣ} {σ : G D.host.E Ω} (hx : a.sym x = toAb D.host.E Ω σ) {n : ℕ} (hn : n ≠ 0) (hψ : ψ σ ^ n = 1) : ((charOf a f ψ x : Circle) : ℂ) = f (ψ σ : Rˣ)

@[simp] theorem **charOf_one** : charOf a f 1 = 1

theorem **mem_admissibleConjugatePairPlaces_iff** {i : X.Tw} {Q : Finset (Pl D.host.E)} {M : ℕ} {P : Pl D.host.E} : P ∈ admissibleConjugatePairPlaces X i Q M ↔ P ∈ Q ∧ ∃ P' ∈ Q, IsConjPl D.host P P' ∧ Formalism.MainNotions.NAdm.AdmQ X i {P, P'} M

theorem **admissibleConjugatePairPlaces_subset** (i : X.Tw) (Q : Finset (Pl D.host.E)) (M : ℕ) : admissibleConjugatePairPlaces X i Q M ⊆ Q

@[simp] theorem **admissibleConjugatePairPlaces_empty** (i : X.Tw) (M : ℕ) : admissibleConjugatePairPlaces X i ∅ M = ∅

theorem **conj_mem_admissibleConjugatePairPlaces** {i : X.Tw} {Q : Finset (Pl D.host.E)} {M : ℕ} {P P' : Pl D.host.E} (h : P ∈ admissibleConjugatePairPlaces X i Q M) (hc : IsConjPl D.host P P') : P' ∈ admissibleConjugatePairPlaces X i Q M

theorem **ne_of_split** {P P' : Pl D.host.E} (h : IsSplitPl D.host P) (hc : IsConjPl D.host P P') : P ≠ P'

theorem **isSplit_of_mem_admissibleConjugatePairPlaces** {i : X.Tw} {Q : Finset (Pl D.host.E)} {M : ℕ} {P : Pl D.host.E} (h : P ∈ admissibleConjugatePairPlaces X i Q M) : IsSplitPl D.host P

theorem **coord_apply** {P : Pl D.host.E} (h : P ∈ Q) (σ : G D.host.E N.Om) : Multiplicative.toAdd (coord Y i Q M P σ) = Multiplicative.toAdd (Y.delU i Q M σ) ⟨P, h⟩

theorem **spread_apply** (x : ↥(half ch (admissibleConjugatePairPlaces X i Q M)) → ZMod (X.l i ^ M)) (v : ↥Q) : spread Y i Q M ch x v = ∑ P, weight Y i (admissibleConjugatePairPlaces X i Q M) M ⟨P.1, (mem_half.1 P.2).1⟩ v.1 * x P

theorem **spread_of_not_mem** (x : ↥(half ch (admissibleConjugatePairPlaces X i Q M)) → ZMod (X.l i ^ M)) (v : ↥Q) (hv : v.1 ∉ admissibleConjugatePairPlaces X i Q M) : spread Y i Q M ch x v = 0

theorem **spread_of_mem** (hch : IsChoice ch) (x : ↥(half ch (admissibleConjugatePairPlaces X i Q M)) → ZMod (X.l i ^ M)) (v : ↥Q) (hv : v.1 ∈ half ch (admissibleConjugatePairPlaces X i Q M)) : spread Y i Q M ch x v = x ⟨v.1, hv⟩

theorem **spread_coord** (hch : IsChoice ch) (σ : G D.host.E N.Om) (v : ↥Q) : spread Y i Q M ch (fun P => Multiplicative.toAdd (coord Y i (admissibleConjugatePairPlaces X i Q M) M P.1 σ)) v = Multiplicative.toAdd (coord Y i (admissibleConjugatePairPlaces X i Q M) M v.1 σ)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsStratHom

theorem **E_ne_zero** : bk.E ≠ 0

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsTheRecOfIsRec

theorem **RecUnique.isTheRec** (H : RecUnique) {h : IsPlaceOver A w} {rc : IrrGL (w.adicCompletion E) → WD h} (hrc : IsRecLit h rc) : IsTheRec h rc

theorem **RecUnique.isTheRec_of_isRec** (H : RecUnique) {h : IsPlaceOver A w} {rc : IrrGL (w.adicCompletion E) → WD h} (hrc : IsRec h rc) : IsTheRec h rc

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsUnitarisableOfCharOne

theorem **IsEssSqInt.tw** {x : IrrGL k} (h : x.IsEssSqInt) (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (x.tw χ hχ).IsEssSqInt

theorem **IsEssSqInt.nu** {x : IrrGL k} (h : x.IsEssSqInt) (s : ℂ) : (x.nu s).IsEssSqInt

theorem **IsBoxSum.of_equiv** (h : IsBoxSum x π) {σ' : Representation ℂ (GL (Fin N) k) W} (φ : π.Equiv σ') : IsBoxSum x σ'

theorem **IsBoxSum.sum_fst** (h : IsBoxSum x π) : ∑ a, (x a).1 = N

theorem **isBoxSum_single** {y : IrrGL k} {Δ : Segment k} (hΔ : IsLSeg Δ y.rep) : IsBoxSum (fun _ : Fin 1 => y) y.rep

theorem **IsGenSpeh.of_equiv** {π₀ : IrrGL k} {π : Representation ℂ (GL (Fin N) k) V} (h : IsGenSpeh π₀ t π) {σ : Representation ℂ (GL (Fin N) k) W} (φ : π.Equiv σ) : IsGenSpeh π₀ t σ

theorem **IsGenSpeh.mul_eq** {π₀ : IrrGL k} {π : Representation ℂ (GL (Fin N) k) V} (h : IsGenSpeh π₀ t π) : t * π₀.1 = N

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.IsWeilDatumOOneOfRankZero

theorem **SWSetup.posDef** {νV 𝓚 νW 𝒢 μ ℱ fr} (h : SWSetup ψ V W χV χW om νV 𝓚 νW 𝒢 μ ℱ fr) (τ : D.host.E →+* ℂ) : (V.gram.map τ).PosDef

theorem **exists_lagFrame** (W : HSp D) (Y : Lag W) : ∃ C, IsLagFrame W Y C

theorem **isLagFrame_diag** (W : HSp D) : IsLagFrame W (Lag.diag W) (1, 1)

theorem **ratMul_one** {a c : ℕ} (u : Mx D a c) : ratMul 1 u = u

theorem **lagTerm_one_one** (μ : Measure (WO V W)) (ℱ : Set (WO V W)) (f₁ f₂ : WO V W → ℂ) : lagTerm μ ℱ (1, 1) f₁ f₂ = thetaIP μ ℱ f₂ f₁

theorem **exists_haar_fundDomain** (V : HSp D) : ∃ (ν : Measure (GA D V)) (𝒢 : Set (GA D V)), ν.IsHaarMeasure ∧ IsFundamentalDomain (GQ V) 𝒢 ν

theorem **exists_swSetup** (hw : IsWeilDatumO ψ V W χV χW om) (hs : V.SigOK) (hp : V.IsPosDef) (hb : 2 * W.rank < V.rank) : ∃ νV 𝓚 νW 𝒢 μ ℱ fr, SWSetup ψ V W χV χW om νV 𝓚 νW 𝒢 μ ℱ fr

theorem **exists_hecke_resEta** (D : D7) (n : ℕ) : ∃ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ ∧ ResEta D χ n

theorem **exists_swSetup_of_weilExists** (hex : Formalism.IndefR.AutomorphicFormsHyp.WeilExistsO D) (V W : HSp D) (hs : V.SigOK) (hp : V.IsPosDef) (hb : 2 * W.rank < V.rank) : ∃ ψ χV χW om νV 𝓚 νW 𝒢 μ ℱ fr, SWSetup ψ V W χV χW om νV 𝓚 νW 𝒢 μ ℱ fr

theorem **exists_swSetup_unitForm** (hex : Formalism.IndefR.AutomorphicFormsHyp.WeilExistsO D) (W : HSp D) (n : ℕ) (hb : 2 * W.rank < n) : ∃ ψ χV χW om νV 𝓚 νW 𝒢 μ ℱ fr, SWSetup (D

theorem **isWeilDatumO_one_of_rank_zero** (hψ : IsStdChar D ψ (-1)) (h1 : IsHeckeChar D χV) (h2 : ResEta D χV V.rank) (h : W.rank = 0) : IsWeilDatumO ψ V W χV 1 1

theorem **exists_swSetup_of_rank_zero** (V W : HSp D) (hs : V.SigOK) (hp : V.IsPosDef) (hV : 0 < V.rank) (h : W.rank = 0) : ∃ ψ χV χW om νV 𝓚 νW 𝒢 μ ℱ fr, SWSetup ψ V W χV χW om νV 𝓚 νW 𝒢 μ ℱ fr

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.JumpSetDistinctOfRealPlaceFacts

theorem **hc_u_anti** : ∀ (i j : ℕ) (hi : i < R.k) (hj : j < R.k), i < j → P.u ⟨j, hj⟩ < P.u ⟨i, hi⟩

theorem **hc_v_anti** (a a' : Fin (R.k - 1)) (h : a.1 < a'.1) : P.v a' < P.v a

theorem **hc_b_mono** : ∀ (i j : ℕ) (hi : i < R.m) (hj : j < R.m), i < j → P.b ⟨i, hi⟩ < P.b ⟨j, hj⟩

theorem **hc_b_mono'** (i i' : Fin R.m) (h : i.1 < i'.1) : P.b i < P.b i'

theorem **hc_b_le** (i i' : Fin R.m) (h : i.1 ≤ i'.1) : P.b i ≤ P.b i'

theorem **hc_low_pos** (jj : Fin R.m) (i : Fin R.m) (h : i.1 ≤ jj.1) : ((R.t : ℚ) + 3) / 2 ≤ P.e jj - P.b i

theorem **hc_low_neg** (jj : Fin R.m) (i : Fin R.m) (h : jj.1 < i.1) : ((R.t : ℚ) + 3) / 2 ≤ P.b i - P.e jj

theorem **sz_eq** (j : ℕ) (ρ : D.what₂.Tau) (h : j - 1 < D.R.m) : sz D j ρ = (D.hc ρ).e ⟨j - 1, h⟩

theorem **wj_eq** (j : ℕ) (ρ : D.what₂.Tau) : wj D j ρ = ((sz D j ρ : ℚ) : ℝ)

theorem **b'_pos** (L : SizeData₂ D) : 0 < L.sc.b'

theorem **w''_le_top** (L : SizeData₂ D) : L.sc.w'' D.R.n ≤ L.sc.b' + L.sc.b' / L.sc.x

theorem **sz_win** {L : SizeData₂ D} (hP : AdmPar L) (j : ℕ) (hj1 : 1 ≤ j) (hjm : j ≤ D.R.m) (ρ : D.what₂.Tau) : L.sc.w'' D.R.n ≤ ((sz D j ρ : ℚ) : ℝ) ∧ ((sz D j ρ : ℚ) : ℝ) ≤ L.sc.b + L.etaSz + 1

theorem **jFacts_of** {L : SizeData₂ D} (hP : AdmPar L) (j : ℕ) (hj1 : 1 ≤ j) (hjm : j ≤ D.R.m) (ρ : D.what₂.Tau) : JFacts L j ρ

lemma **jp_cast_sub** (t c : ℕ) (h : c < t) : ((t - 1 - c : ℕ) : ℚ) = (t : ℚ) - 1 - c

lemma **jp_t_one** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) : (1 : ℚ) ≤ D.R.t

lemma **jp_s_gt** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) : (D.R.t : ℚ) < sz D j ρ

lemma **jp_b_pos** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (i : Fin D.R.m) : 0 < (D.hc ρ).b i

lemma **jp_low_cases** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (i : Fin D.R.m) : ((D.R.t : ℚ) + 3) / 2 ≤ sz D j ρ - (D.hc ρ).b i ∨ sz D j ρ - (D.hc ρ).b i ≤ -(((D.R.t : ℚ) + 3) / 2)

lemma **jp_block_iff** (q : ℚ) : q ∈ blockJ D ↔ ∃ c : ℕ, c < D.R.t ∧ q = c

lemma **jp_refl_block** (q : ℚ) (hq : q ∈ blockJ D) : refl D q ∈ blockJ D

lemma **jp_refl_refl** (q : ℚ) : refl D (refl D q) = q

lemma **jp_block_not_cloud** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (q : ℚ) (hq : q ∈ blockJ D) : q ∉ cloudJ D j ρ

lemma **jp_block_not_low** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (q : ℚ) (hq : q ∈ blockJ D) : q ∉ lowJ D j ρ

theorem **block_mem_iff** (a : ℤ) : ((a : ℚ) ∈ blockJ D ↔ 0 ≤ a ∧ a < (D.R.t : ℤ))

theorem **block_not_mem_JW** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (up : Bool) (q : ℚ) (hq : q ∈ blockJ D) : q ∉ JW D up j ρ

theorem **block_sub_JJ** (up : Bool) (j : ℕ) (ρ : D.what₂.Tau) (q : ℚ) (hq : q ∈ blockJ D) : q ∈ JJ D up j ρ

theorem **JJ_iff_JW** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (up : Bool) (q : ℚ) (hq : q ∉ blockJ D) : q ∈ JJ D up j ρ ↔ q ∈ JW D up j ρ

lemma **jp_nodup** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) : (PPm D j ρ).Nodup

lemma **jp_mem_iff** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (q : ℚ) : q - sz D j ρ - c0 D ∈ PPm D j ρ ↔ q ∈ Jp D j ρ

theorem **count_PPm** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (q : ℚ) : (PPm D j ρ).count (q - sz D j ρ - c0 D) = if q ∈ Jp D j ρ then 1 else 0

lemma **realPlace_jumpCenter_nonneg** (F : JFacts L j ρ) : 0 ≤ c0 D

lemma **realPlace_t_add_one_lt_size** (F : JFacts L j ρ) : (D.R.t : ℚ) + 1 < sz D j ρ

lemma **realPlace_etaSize_lt_size** (F : JFacts L j ρ) : L.etaSz < ((sz D j ρ : ℚ) : ℝ)

lemma **jb_cloud** (F : JFacts L j ρ) {q : ℚ} (hq : q ∈ cloudJ D j ρ) : sz D j ρ ≤ q − c0 D ∧ ((q − c0 D : ℚ) : ℝ) ≤ ((sz D j ρ : ℚ) : ℝ) + L.Sc

lemma **jb_low** (F : JFacts L j ρ) {q : ℚ} (hq : q ∈ lowJ D j ρ) : ((D.R.t : ℚ) + 3) / 2 ≤ |q − c0 D| ∧ ((|q − c0 D| : ℚ) : ℝ) ≤ L.etaSz

lemma **jb_block** {q : ℚ} (hq : q ∈ blockJ D) : |q − c0 D| ≤ c0 D

lemma **jb_notin_cl** (F : JFacts L j ρ) {x : ℚ} (hx : |x − c0 D| ≤ ((D.R.t : ℚ) + 1) / 2) : x ∉ cloudJ D j ρ ∪ lowJ D j ρ

lemma **jb_notin_Jp** (F : JFacts L j ρ) {x : ℚ} (hx : |x − c0 D| = ((D.R.t : ℚ) + 1) / 2) : x ∉ Jp D j ρ

lemma **jb_refl_abs** (q : ℚ) : |refl D q − c0 D| = |q − c0 D|

lemma **jb_Jp_bound** (F : JFacts L j ρ) {q : ℚ} (hq : q ∈ Jp D j ρ) : ((|q − c0 D| : ℚ) : ℝ) ≤ L.sc.b + L.Sc + L.etaSz + 1

lemma **jb_cloud_eq** : cloudJ D j ρ = Set.range (fun a : Fin (D.R.k − 1) => sz D j ρ + (D.hc ρ).v a + c0 D)

lemma **jb_low_eq** : lowJ D j ρ = Set.range (fun a : Fin D.R.m => sz D j ρ − (D.hc ρ).b a + c0 D)

lemma **jb_block_eq** : blockJ D = Set.range (fun c : Fin D.R.t => ((c : ℕ) : ℚ))

lemma **jb_cloud_inj** (F : JFacts L j ρ) : Function.Injective (fun a : Fin (D.R.k − 1) => sz D j ρ + (D.hc ρ).v a + c0 D)

lemma **jb_low_inj** (F : JFacts L j ρ) : Function.Injective (fun a : Fin D.R.m => sz D j ρ − (D.hc ρ).b a + c0 D)

lemma **jb_block_inj** : Function.Injective (fun c : Fin D.R.t => ((c : ℕ) : ℚ))

theorem **jumpSetBound_of_realPlaceFacts** {L : SizeData₂ D} (h : ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ ρ, JFacts L j ρ) : CloudCentreDistance L

theorem **jumpSetCloud_of_realPlaceFacts** {L : SizeData₂ D} (h : ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ ρ, JFacts L j ρ) : jcCondition L

theorem **jumpSetEta_of_realPlaceFacts** {L : SizeData₂ D} (h : ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ ρ, JFacts L j ρ) (hpar : L.etaSz + D.R.t + L.Sc ≤ L.sc.b) (ht : 1 ≤ D.R.t) (hS : 0 < L.Sc) (he : 0 < L.etaSz) : JumpAbsoluteBound L

theorem **jumpSetGap_of_realPlaceFacts** {L : SizeData₂ D} (h : ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ ρ, JFacts L j ρ) : ThickPlaceCloudSizes L

lemma **jd_inj** {n : ℕ} (f : Fin n → ℚ) (h : ∀ a a' : Fin n, a.1 < a'.1 → f a ≠ f a') : Function.Injective f

lemma **jd_card_lt** {m j : ℕ} (hj : j ≤ m) : (Finset.univ.filter (fun a : Fin m => a.1 < j)).card = j

lemma **jd_card_ge** {m j : ℕ} (hj : j ≤ m) : (Finset.univ.filter (fun a : Fin m => j ≤ a.1)).card = m − j

lemma **jd_ncard** {n : ℕ} (f : Fin n → ℚ) (hf : Function.Injective f) (p : Fin n → Prop) [DecidablePred p] (S : Set ℚ) (hS : ∀ q, q ∈ S ↔ ∃ a, p a ∧ q = f a) : S.ncard = (Finset.univ.filter p).card

lemma **jd_ncard_union** {n₁ n₂ : ℕ} (f : Fin n₁ → ℚ) (g : Fin n₂ → ℚ) (hf : Function.Injective f) (hg : Function.Injective g) (p : Fin n₂ → Prop) [DecidablePred p] (hfg : ∀ a b, f a ≠ g b) (S : Set ℚ) (hS : ∀ q, q ∈ S ↔ (∃ a, q = f a) ∨ ∃ b, p b ∧ q = g b) : S.ncard = n₁ + (Finset.univ.filter p).card

lemma **cloudJumpSet_true** : cloudT D true j ρ = cloudJ D j ρ

lemma **jd_cloudT_f** : cloudT D false j ρ = reflS D (cloudJ D j ρ)

lemma **lowJumpSet_true** : lowT D true j ρ = lowJ D j ρ

lemma **jd_lowT_f** : lowT D false j ρ = reflS D (lowJ D j ρ)

lemma **jumpUnion_true** : JW D true j ρ = cloudJ D j ρ ∪ lowJ D j ρ

lemma **jd_JW_f** : JW D false j ρ = reflS D (cloudJ D j ρ) ∪ reflS D (lowJ D j ρ)

lemma **one_le_jump_parameter** (F : JFacts L j ρ) : (1 : ℚ) ≤ D.R.t

lemma **jumpCount_t_add_one_lt_size** (F : JFacts L j ρ) : (D.R.t : ℚ) + 1 < sz D j ρ

lemma **jd_b** (F : JFacts L j ρ) (a : Fin D.R.m) : 0 < (D.hc ρ).b a

lemma **jd_lneg** (F : JFacts L j ρ) (a : Fin D.R.m) (h : j ≤ a.1) : sz D j ρ − (D.hc ρ).b a + c0 D ≤ −2

lemma **jd_lpos** (F : JFacts L j ρ) (a : Fin D.R.m) (h : a.1 < j) : (D.R.t : ℚ) + 1 ≤ sz D j ρ − (D.hc ρ).b a + c0 D

lemma **jd_cl** (F : JFacts L j ρ) (a : Fin (D.R.k − 1)) : (D.R.t : ℚ) + 1 < sz D j ρ + (D.hc ρ).v a + c0 D

lemma **jd_cl_low** (F : JFacts L j ρ) (a : Fin (D.R.k − 1)) (a' : Fin D.R.m) : sz D j ρ − (D.hc ρ).b a' + c0 D < sz D j ρ + (D.hc ρ).v a + c0 D

theorem **jumpSetDistinct_of_realPlaceFacts** {L : SizeData₂ D} (h : ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ ρ, JFacts L j ρ) : JumpLowsSignCount D

theorem **jumpSetFinal_of_realPlaceFacts** {L : SizeData₂ D} (h : ∀ j, 1 ≤ j → j ≤ D.R.m → ∀ ρ, JFacts L j ρ) : DualCloudSizes L

theorem **cc_mul_cc'** (φ : Formalism.HC.Frame) : φ.cc * φ.cc = 1

theorem **mul_cc_mem_Psi'** {φ : Formalism.HC.Frame} {g : φ.GG} (hg : g ∉ φ.Psi) : g * φ.cc ∈ φ.Psi

theorem **exists_placeOf** (g : D.R.GG) : ∃ ρ : D.what₂.Tau, PlaceOf D ρ g

theorem **count_key** {L : SizeData₂ D} {j : ℕ} {ρ : D.what₂.Tau} (hF : JFacts L j ρ) (J : Multiset ℤ) (sg : ℚ) (cτ c : ℤ) (up : Bool) (hsU : up = true → sg = 1) (hsD : up = false → sg = −1) (hJ : J.map (fun n : ℤ => (n : ℚ)) = (PPm D j ρ).map (fun p => sg * p + cτ)) (hc : (c : ℚ) = sg * (−sz D j ρ) − c0 D + cτ) (a : ℤ) : ((J.count (a + c) : ℕ) : ℤ) = if (a : ℚ) ∈ JJ D up j ρ then 1 else 0

theorem **ite_ne_zero_iff'** (P : Prop) [Decidable P] : ((if P then (1 : ℤ) else 0) ≠ 0) ↔ P

theorem **dualS_int** (S : Set ℚ) (a : ℤ) : ((a : ℚ) ∈ dualS S ↔ ((−a − 1 : ℤ) : ℚ) ∈ S)

theorem **jumpLemma_a_of_admissible** {L : SizeData₂ D} (hP : AdmPar L) (hG2 : DeRhamFamilyAway D) (hCr : CrysIffWD D.nd D.naut) (hG2g : DeRhamFamilyTypePrimes D) (hDi : AdmDict D) (hRd : AdmRead D) : ConclusionJa D

theorem **stmtJ_holds** (D : D7) : JumpBoundClosedForm D

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.KillingFormDiffCartanInvolution

theorem **cartan_involution_apply** (g : GP a b R) : th0 g = star g⁻¹

@[simp] theorem **cartan_involution_involutive** (g : GP a b R) : th0 (th0 g) = g

@[simp] theorem **thetaG_thetaG** (g : GP a b R) : thetaG (thetaG g) = g

theorem **cartan_involution_frobenius** : th0 (PhiG a b R) = PhiG a b R

theorem **mat_apply** (g : GR a b) : mat g = (g.1.val, g.2.val)

theorem **mat_star** (g : GR a b) : mat (star g) = star (mat g)

theorem **mat_injective** : Function.Injective (mat : GR a b → LieG a b)

@[simp] theorem **mat_mul_inv** (g : GR a b) : mat g * mat g⁻¹ = 1

@[simp] theorem **mat_inv_mul** (g : GR a b) : mat g⁻¹ * mat g = 1

theorem **mem_KR** {k : GR a b} : k ∈ KR a b ↔ th0 k = k

theorem **mem_KR_iff_star** {k : GR a b} : k ∈ KR a b ↔ star k = k⁻¹

theorem **mem_KR_iff_mat** {k : GR a b} : k ∈ KR a b ↔ mat k * star (mat k) = 1

theorem **mem_KR_iff_unitary** {k : GR a b} : k ∈ KR a b ↔ k.1.val ∈ unitaryGroup (Fin a) ℂ ∧ k.2.val ∈ unitaryGroup (Fin b) ℂ

theorem **PhiG_mem_KR** : PhiG a b ℂ ∈ KR a b

theorem **isCompact_unitary_GL** (n : ℕ) : IsCompact {k : GL (Fin n) ℂ | k.val ∈ unitaryGroup (Fin n) ℂ}

theorem **isCompact_KR** : IsCompact (KR a b : Set (GR a b))

theorem **mem_pSp** {X : LieG a b} : X ∈ pSp a b ↔ star X = X

theorem **AdLie_apply** (g : GR a b) (X : LieG a b) : AdLie g X = mat g * X * mat g⁻¹

theorem **diff_cartan_involution_apply** (X : LieG a b) : dth0 X = -star X

theorem **AdLie_mul** (g h : GR a b) : AdLie (g * h) = AdLie g ∘ₗ AdLie h

@[simp] theorem **AdLie_one** : AdLie (1 : GR a b) = LinearMap.id

theorem **AdT_eq** (δ : GR a b) : AdT δ = AdLie (toTh0 δ) ∘ₗ dth0

theorem **AdLie_mem_pSp** {k : GR a b} (hk : k ∈ KR a b) {X : LieG a b} (hX : X ∈ pSp a b) : AdLie k X ∈ pSp a b

theorem **AdT_mem_pSp** {k : GR a b} (hk : k ∈ KR a b) {X : LieG a b} (hX : X ∈ pSp a b) : AdT k X ∈ pSp a b

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LabFixedOfTypesTrue

theorem **labFixed_withLoc** {D : D7} (L : LocAll (lay0 D)) : LabFixed (withLoc L) ↔ LabFixed D

theorem **labFixed_iff** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {cf : CharFrame D} {nch : NChoice D} {tp : TupTied D cf nch} (a : NAt hL D cf nch tp) : LabFixed D ↔ LabFixed a.nt.D

theorem **labFixed_of_typesTrue** {D : D7} {P : PiBar D.host} {W : Omg D.host} (h : TypesTrue D.ntypes P W) : LabFixed D

theorem **lab_of_rows** {hL : GlobalTheory.Published.LocalLanglandsGL} {w : ColW B f T} {Z : w.Lt} {Wt : Set B.places.Pl} {p0 : BaseLevelRowSide (w.D Z)} (t : RowsT K w Z Wt p0) {nch : NChoice (w.D Z)} {n : NchT w Z Wt nch} (c : NCore hL t n) : LabFixed c.a.nt.D

theorem **isGenSys_gen** (g : ℕ) [Fact g.Prime] (L : GlobalTheory.GLq.LabelDatum g) : Formalism.L2.NT.IsGenSys L.gen

theorem **tame_of_pin** {hL : GlobalTheory.Published.LocalLanglandsGL} {D : D7} {cf : CharFrame D} {nch : NChoice D} {tp : TupTied D cf nch} {a : NAt hL D cf nch tp} (h : OPin a) (k : Fin 2) (v : a.nt.D.nd.PlG k) : @Formalism.L2.NT.TameCoupled _ _ _ _ _ (a.nt.D.nd.plOm.1 (a.nt.D.naut.plGw k v)) _ (a.nt.D.host.g_prime k) (a.nt.A.grp.kv k v) (a.nt.D.ntypes.tg k v) (@GlobalTheory.GLq.LabelDatum.gen _ (a.nt.D.host.g_prime k) (@GlobalTheory.Mem.TypePl.labDat (a.nt.D.host.g k) (a.nt.D.host.g_prime k)))

theorem **opin_iff** {hL : GlobalTheory.Published.LocalLanglandsGL} {w : ColW B f T} {Z : w.Lt} {Wt : Set B.places.Pl} {p0 : BaseLevelRowSide (w.D Z)} (t : RowsT K w Z Wt p0) {nch : NChoice (w.D Z)} {n : NchT w Z Wt nch} (c : NCore hL t n) : OPin c.a ↔ (SumRepresentations.hypO (w.D Z) ∧ TakenAll (w.D Z)) ∧ (∀ i, ClassSum (c.a.all.gal.mt.loc i)) ∧ ∀ k, c.a.nt.A.grp.ζ k = @GlobalTheory.GLq.LabelDatum.gen _ (c.a.nt.D.host.g_prime k) (@GlobalTheory.Mem.TypePl.labDat (c.a.nt.D.host.g k) (c.a.nt.D.host.g_prime k))

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LabelDatum

theorem **hcL_nil** (n : ℕ) : hcL ([] : List (Pc g)) n = fun _ => 1

theorem **hcL_cons** (p : Pc g) (L : List (Pc g)) (n : ℕ) : hcL (p :: L) n = hc p.1 (degL L) p.2 (hcL L (degL L)) n

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.Langlands

theorem **Laws.toLit** {a : Weil h →* (w.adicCompletion E)ˣ} {ha : IsLocalArtin h a} {ε : EpsFamily A} {rc : IrrGL (w.adicCompletion E) → WD h} (H : Laws a ha ε rc) : LawsLit a ha (ε E w h) rc

theorem **IsRec.toLit** {rc : IrrGL (w.adicCompletion E) → WD h} (H : IsRec h rc) : IsRecLit h rc

theorem **PlaceIso.refl** : PlaceIso A w A w (RingEquiv.refl Ω) (RingHom.id E) (RingEquiv.refl _)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LatIndexEqOneOfMem

theorem **latIndex_eq_one_of_mem** (W : HSp D) (g : GA D W) (hg : g ∈ KsphC W ∅) (hg' : g⁻¹ ∈ KsphC W ∅) : latIndex W g = 1

theorem **coef_one** (R : RealDat D A Ks) (i : Fin D.R.m) (t : T) (h : VecNe R i t) (μ) (ℱ) (hμ : IsCovolOne (ratL (Wc D i) (Vc D i)) μ ℱ) : coef R i t μ ℱ 1 = 1

theorem **pureTensor_one** (R : RealDat D A Ks) (i : Fin D.R.m) (t : T) : R.ptP i t 1 = R.vP i t

theorem **coef_mul_lev** (R : RealDat D A Ks) (i : Fin D.R.m) (t : T) (h : LevFix R i t) (μ) (ℱ) (g k : GA D (Wc D i)) (hk : k ∈ levW R i) : coef R i t μ ℱ (g * k) = coef R i t μ ℱ g

theorem **goodMod_ne** (R : RealDat D A Ks) (i : Fin D.R.m) (t : T) (h : GoodMod R i t) (μ) (ℱ) (hμ : IsCovolOne (ratL (Wc D i) (Vc D i)) μ ℱ) (v f) (hf : IsFinIdem D v f) (hP : ∀ P, P ∈ R.Pg v P ∈ R.Q t → ∀ e f', IsPrimeIdem P e f' → f' ≠ f) (k) (hk : SuppF (Wc D i) f k) : coef R i t μ ℱ k ≠ 0

theorem **trivAt_one** (V : HSp D) (f : AE D) : TrivAt V f 1 ∧ SuppF V f 1

theorem **schur_hyp_scalar** (ψ : AddChar (AF D.host.F) ℂ) (S T' : HSp D) (c : ℂ) (w : WO S T') (f : WO S T' → ℂ) : (c • (LinearMap.id : Module.End ℂ (WO S T' → ℂ))) (heisO ψ S T' w f) = heisO ψ S T' w ((c • (LinearMap.id : Module.End ℂ (WO S T' → ℂ))) f)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LatticeMembershipByPowerSeries

theorem **algebraMap_eq** (c : k) : algebraMap k k⟦X⟧ c = HahnSeries.C c

theorem **smul_eq** (c : k) (x : k⟦X⟧) : c • x = algebraMap k k⟦X⟧ c * x

theorem **vv_eq** : vv k = ((PowerSeries.X : k⟦X⟧) : k⟦X⟧)

theorem **vv_ne** : vv k ≠ 0

theorem **powerSeries_oneAddConstMulX_mul_inv** (c : k) : (1 + PowerSeries.C c * PowerSeries.X) * inv1 k c = 1

theorem **hs** (c : k) : PowerSeries.HasSubst (PowerSeries.X * inv1 k c : k⟦X⟧)

theorem **sb_X** (c : k) : sb k c PowerSeries.X = PowerSeries.X * inv1 k c

theorem **sb_C** (c d : k) : sb k c (PowerSeries.C d) = PowerSeries.C d

theorem **sb_key** (c : k) : sb k c (PowerSeries.X * inv1 k (-c)) = PowerSeries.X

theorem **sb_sb** (c : k) (f : k⟦X⟧) : sb k c (sb k (-c) f) = f

theorem **gE_coe** (a : k⟦X⟧) : gE k (a : k⸨X⸩) = ((sb k 1 a : k⟦X⟧) : k⸨X⸩)

theorem **mem_lat** {a : ℤ} {y : k⸨X⸩} : y ∈ lat k a ↔ ∃ b : k⟦X⟧, (b : k⸨X⸩) * vv k ^ a = y

theorem **sh_lat** (a b : ℤ) : sh a (lat k b) = lat k (a + b)

theorem **isLat_lat** (a : ℤ) : IsLat (lat k a)

theorem **nab_apply** (g : k⸨X⸩ →ₗ[k] k⸨X⸩) (y : k⸨X⸩) : nab g y = (vv k)⁻¹ * (g y - y)

theorem **sb_sub** (a : k⟦X⟧) : ∃ b : k⟦X⟧, sb k 1 a - a = PowerSeries.X * b

theorem **nab_coe** (a : k⟦X⟧) : ∃ b : k⟦X⟧, nab (lin (gA k)) (a : k⸨X⸩) = b

theorem **g_vinv** : gE k (vv k)⁻¹ = (vv k)⁻¹ + 1

theorem **nab_vinv** : nab (lin (gA k)) (vv k)⁻¹ = (vv k)⁻¹

theorem **decomp** {y : k⸨X⸩} (hy : y ∈ lat k (-1)) : ∃ (c : k) (b : k⟦X⟧), y = c • (vv k)⁻¹ + b

theorem **heckeAlgebra_mapsTo_localisationLatticeSpan** {y : k⸨X⸩} : y ∈ lat k 0 ↔ ∃ b : k⟦X⟧, (b : k⸨X⸩) = y

theorem **vinv_mem** : (vv k)⁻¹ ∈ lat k (-1)

theorem **coe_mem** (b : k⟦X⟧) : (b : k⸨X⸩) ∈ lat k (-1)

theorem **vinv_eq** : (vv k)⁻¹ = HahnSeries.single (-1) 1

theorem **coeff_decomp** (c : k) (b : k⟦X⟧) : (c • (vv k)⁻¹ + (b : k⸨X⸩)).coeff (-1) = c

theorem **step** (x c : k) (b : k⟦X⟧) : ∃ b' : k⟦X⟧, (nab (lin (gA k)) - algebraMap k (Module.End k k⸨X⸩) x) (c • (vv k)⁻¹ + b) = ((1 - x) * c) • (vv k)⁻¹ + b'

theorem **steps** (x : k) (m : ℕ) : ∀ (c : k) (b : k⟦X⟧), ∃ b' : k⟦X⟧, ((nab (lin (gA k)) - algebraMap k (Module.End k k⸨X⸩) x) ^ m) (c • (vv k)⁻¹ + b) = ((1 - x) ^ m * c) • (vv k)⁻¹ + b'

theorem **condC** : CondC (lin (gA k)) (lat k (-1))

theorem **sen_one** : sen (lin (gA k)) (lat k (-1)) 1 = 1

theorem **sen_ne** {x : k} (hx : x ≠ 1) : sen (lin (gA k)) (lat k (-1)) x = 0

theorem **pull_lat** (p : ℕ) [Fact p.Prime] (a : ℤ) : pull (lin (ringModel p).φ) (lat (ZMod p) a) = lat (ZMod p) (p * a)

theorem **condAB** (p : ℕ) [Fact p.Prime] (n : ℕ) (a : ℤ) : CondAB (modModel p n).φ (fun _ => (1 - p) * a) (fun _ => (p - 1) * a) (fun _ => lat (ZMod p) a)

theorem **condC_model** (p : ℕ) [Fact p.Prime] (n : ℕ) (i : ZMod n) : CondC ((modModel p n).γ i) (lat (ZMod p) (-1))

theorem **sen_model** (p : ℕ) [Fact p.Prime] (n : ℕ) (i : ZMod n) (x : ZMod p) : sen ((modModel p n).γ i) (lat (ZMod p) (-1)) x = if x = 1 then 1 else 0

theorem **sg_test** : sg 5 (fun _ : ZMod 1 => 9) 0 = 2

theorem **gmult_top_bot** {U : Type} [AddCommGroup U] [Module k U] (T : Module.End k U) (x : k) : gmult T ⊤ ⊥ x = Module.finrank k (T.maxGenEigenspace x)

theorem **isEmpty_ringPG_zero** : IsEmpty (RingPG k 0)

theorem **condAB_var** (p : ℕ) [Fact p.Prime] (n : ℕ) (a : ZMod n → ℤ) : CondAB (modModel p n).φ (fun i => a i - p * a (i - 1)) (fun i => p * a (i - 1) - a i) (fun i => lat (ZMod p) (a i))

theorem **sg_sharp** : sg 5 (fun _ : ZMod 3 => 4) 0 = 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LayersOfExists

theorem **fixRoot_of** {p : ℕ} {ζ : LL} (h : ∃ j : ℕ, IsPrimitiveRoot ζ (p ^ j)) : fixRoot p ζ = ζ

theorem **fixRoot_not** {p : ℕ} {ζ : LL} (h : ¬ ∃ j : ℕ, IsPrimitiveRoot ζ (p ^ j)) : fixRoot p ζ = 1

theorem **fixRoot_one** (p : ℕ) : fixRoot p (1 : LL) = 1

theorem **prim_pow** {p : ℕ} (hp : p.Prime) {ζ : LL} (h : ∃ j : ℕ, IsPrimitiveRoot ζ (p ^ j)) (a : ℕ) : ∃ j : ℕ, IsPrimitiveRoot (ζ ^ a) (p ^ j)

theorem **tmul_cancel** (hO : Function.Injective (algebraMap O LL)) (W : LocWach PD LL O v n) (e₀ : ɪW.one.L.T) (h₀ : Submodule.span O {e₀} = ⊤) {r r' : RW} (h : (r ⊗ₜ[O] e₀ : TW RW W.one) = r' ⊗ₜ[O] e₀) : r = r'

theorem **ioE_φ** (hO : Function.Injective (algebraMap O LL)) (E : EtTie W (O ⊗[Zl] Ai) RW inclR φR gR πA) (a : W.EE) : E.ioE (W.φE a) = φR (E.ioE a)

theorem **ioE_γ** (hO : Function.Injective (algebraMap O LL)) (E : EtTie W (O ⊗[Zl] Ai) RW inclR φR gR πA) (g : G) (a : W.EE) : E.ioE (W.γE (W.resΓ g) a) = gR g (E.ioE a)

theorem **Ncut_of** {T : LocLat G Qp LL O} (h : IsCrys PD T.ρ) : Ncut W T = W.N T

theorem **Ncut_not** {T : LocLat G Qp LL O} (h : ¬ IsCrys PD T.ρ) : Ncut W T = ⊥

theorem **Ncut_le** (W : LocWach PD LL O v n) (T : LocLat G Qp LL O) : Ncut W T ≤ W.N T

theorem **exists_layers** [FiniteDimensional Qp LL] (hO : Function.Injective (algebraMap O LL)) (C : CmpTie W E ιdR p1 tS zS) {zl : PowerSeries Zl →+* LetA O n} (hproj : ∀ T, @Module.Projective (PowerSeries Zl) _ ɪ(W.N T) _ (Module.compHom _ zl)) (hz : ∀ (ι : LL →ₐ[Qp] BdR) (ζ : LL), ι ζ = zS → ∃ j : ℕ, IsPrimitiveRoot ζ (PD.p ^ j)) {s : ℕ} (hs : W.s = s) : ∃ (Γ : WGam PD LL O v n s) (R : WSer Γ) (I : WIo R (O ⊗[Zl] Ai) RW inclR φR gR πA) (E' : WEt I) (Mo : WN E' zl), Γ.emb = W.emb ∧ I.io = E.io ∧ Nonempty (WCl E') ∧ Nonempty (WCmp Mo ιdR p1 tS zS)

theorem **zeta_pow** (P : LocPrime l) (s : ℕ) : P.ζQ s ^ (l ^ s) = 1

theorem **layers_of_exists** (P : LocPrime l) (AI : LocAinf l P) (H : LocHW l P) (F : LocFon l P AI H) (C : LocCmpT l P AI H F) (K : FinExt l) (s : ℕ) (LL : FinExt l) (g : Good K s LL) : Nonempty (Layers P AI K s LL)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LbcOfCharOne

theorem **lbc_ofChar_one** (f : v.adicCompletion K →+* u.adicCompletion L) (hf : Continuous f) (hK : ∀ z : K, f (algebraMap K _ z) = algebraMap L _ (algebraMap K L z)) (hd : letI : Algebra (v.adicCompletion K) (u.adicCompletion L)

theorem **lbc_deg** (x : IrrGL (v.adicCompletion K)) (X : IrrGL (u.adicCompletion L)) (h : lbc K L v u x X) (hs : ∀ f : v.adicCompletion K →+* u.adicCompletion L, letI : Algebra (v.adicCompletion K) (u.adicCompletion L)

theorem **feeds** (hS : GlobalTheory.Published.ScholzeFunctorialExtension) : ∃ σ : SigFam, ∀ (K L Ω : Type) [Field K] [NumberField K] [Field L] [NumberField L] [Field Ω] [Algebra K Ω] [IsAlgClosure K Ω] [Algebra L Ω] [IsAlgClosure L Ω] [Algebra K L] [IsScalarTower K L Ω] (A : ValuationSubring Ω) (v : HeightOneSpectrum (𝒪 K)) (u : HeightOneSpectrum (𝒪 L)) (h : IsPlaceOver A v) (h' : IsPlaceOver A u) (x : IrrGL (v.adicCompletion K)) (X : IrrGL (u.adicCompletion L)), x.IsCusp → lbc K L v u x X → (∀ f : v.adicCompletion K →+* u.adicCompletion L, letI : Algebra (v.adicCompletion K) (u.adicCompletion L)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LengthPosOfPosDef

theorem **Laws.of_char** {R : RealDat D A Ks} {F : Face D A} (h : R.Char F) : Laws R F

theorem **Laws.of_charTh** {R : RealDat D A Ks} {F : Face D A} (h : GlobalTheory.ThetaClauses.CharTh R F) : Laws R F

theorem **Laws.posDef** {R : RealDat D A Ks} {F : Face D A} (h : Laws R F) (i : Fin D.R.m) : (Vc D i).SigOK ∧ (Vc D i).IsPosDef

theorem **length_pos_of_posDef** {V : HSp D} (hs : V.SigOK) (hp : V.IsPosDef) (τ : D.host.E →+* ℂ) (hτ : ∀ a, τ (star a) = (starRingEnd ℂ) (τ a)) (x : V.V) (hx : x ≠ 0) : ∃ c : ℝ, 0 < c ∧ τ (V.hT x x) = (c : ℂ)

theorem **exists_lift_model** {X Z : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup Z] [Module ℂ Z] (M : HMod Z) {U : Submodule ℂ Z} {ip : Z → Z → ℂ} (H : M.Is U ip) (f : X →ₗ[ℂ] Z) (hm : ∀ x, f x ∈ U) : ∃ th : X →ₗ[ℂ] M.H, ∀ x, M.e (th x) = f x

theorem **Laws.exists_isRaw** {R : RealDat D A Ks} {F : Face D A} (h : Laws R F) (i : Fin D.R.m) (t : T) : ∃ th, IsRaw R i t th

theorem **Laws.isRaw_unique** {R : RealDat D A Ks} {F : Face D A} (h : Laws R F) {i : Fin D.R.m} {t : T} {th th' : (R.HV i t).H →ₗ[ℂ] (R.HW i (sg0 D A)).H} (h₁ : IsRaw R i t th) (h₂ : IsRaw R i t th') : th = th'

theorem **InnerPos.innerNe** {R : RealDat D A Ks} {i : Fin D.R.m} {t : T} {S : Finset (HeightOneSpectrum (𝒪 D.host.F))} (h : InnerPos R i t S) : InnerNe R i t S

theorem **thetaInner_smul_left** {S T' : HSp D} (μ : Measure (WO S T')) (𝓕 : Set (WO S T')) (a : ℂ) (f f' : WO S T' → ℂ) : thetaIP μ 𝓕 (a • f) f' = star a * thetaIP μ 𝓕 f f'

theorem **thetaInner_smul_right** {S T' : HSp D} (μ : Measure (WO S T')) (𝓕 : Set (WO S T')) (a : ℂ) (f f' : WO S T' → ℂ) : thetaIP μ 𝓕 f (a • f') = a * thetaIP μ 𝓕 f f'

theorem **upLift_smul** {V W : HSp D} (om : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)) (ν : Measure (GA D W)) (𝒢 : Set (GA D W)) (f : WO V W → ℂ) (a : ℂ) (φ : GA D W → ℂ) (h : GA D V) : upLift om ν 𝒢 f (a • φ) h = a * upLift om ν 𝒢 f φ h

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LocalisationCoefficientExtensionIsWildInertiaCyc

theorem **exists_heq** {p q : ℕ} [Fact p.Prime] [Fact q.Prime] (h : p = q) (z : ℤ_[p]) : ∃ z' : ℤ_[q], HEq z z'

theorem **iotaO_injective** (N : NData R H what₂) (A : NAut N) (i : N.Loc) : Function.Injective (iotaO N A i)

theorem **φ_injective** (a : LocExt N₀ N₁ A₀ A₁ i₀ i₁) : Function.Injective a.φ

theorem **s0_eq** (a : LocExt N₀ N₁ A₀ A₁ i₀ i₁) : N₀.s0 i₀ = N₁.s0 i₁

theorem **heckePol_map** (a : LocExt N₀ N₁ A₀ A₁ i₀ i₁) (x : N₁.X i₁) (w : PlE H) (hw : SplitOut N₁ i₁ w) : heckePol N₁ i₁ x w = (heckePol N₀ i₀ (a.e x) w).map a.φ

theorem **gen_heq** {p q : ℕ} [Fact p.Prime] [Fact q.Prime] (h : p = q) (s : ℕ) (z : ℤ_[p]ˣ) (z' : ℤ_[q]ˣ) (hz : HEq z z') : (∃ u : ℤ_[p]ˣ, (z : ℤ_[p]) = 1 + (p : ℤ_[p]) ^ s * (u : ℤ_[p])) ↔ ∃ u : ℤ_[q]ˣ, (z' : ℤ_[q]) = 1 + (q : ℤ_[q]) ^ s * (u : ℤ_[q])

theorem **localisationCoefficientExtension_isWildInertiaCyclotomic_iff** (a : LocExt N₀ N₁ A₀ A₁ i₀ i₁) (g : GE N₁) : IsWildInertiaCyclotomicElement N₁ i₁ g ↔ IsWildInertiaCyclotomicElement N₀ i₀ (a.γ g)

theorem **contains_ext** (a : LocExt N₀ N₁ A₀ A₁ i₀ i₁) (S : Set ℂ) (h : Contains N₀ A₀ i₀ S) : Contains N₁ A₁ i₁ S

theorem **takenAfter_of_takenWild** (N : NData R H what₂) (i : N.Loc) (rep : N.X i → (GE N →* GL (Fin R.nV) (N.O i))) (ρ : N.X i → (GE N →* GL (Fin R.r) (N.O i))) (h : TakenWild N i rep ρ) : GlobalTheory.IndefiniteRowSideFullHecke.CongruenceModuleMatrices.TakenAfter N i rep ρ

theorem **takenWild_of_chosen** (N : NData R H what₂) (A : NAut N) (i : N.Loc) (S : Set ℂ) (hC : Chosen N A i S) (hS : Contains N A i S) : ∃ r ρ, GoodR N i r ∧ TakenWild N i r ρ

theorem **isBounded_umax** : IsBounded (Umax L n)

theorem **isInvForm_triv** : IsInvForm (trivWt L n) (innerₛₗ ℂ : ℂ →ₗ⋆[ℂ] ℂ →ₗ[ℂ] ℂ)

theorem **exists_isClassReps** (J : Subgroup (GAf L n)) (T : Finset (GAf L n)) (hT : ∀ g : GAf L n, ∃ t ∈ T, ∃ (γ : GQ L n) (κ : GAf L n), κ ∈ J ∧ g = (γ : GAf L n) * t * κ) : ∃ T', IsClassReps J T'

theorem **card_class** (J : Subgroup (GAf L n)) (t κ : GAf L n) (δ : GQ L n) (hκ : κ ∈ J) : Nat.card {γ : GQ L n // ((δ : GAf L n) * t * κ)⁻¹ * (γ : GAf L n) * ((δ : GAf L n) * t * κ) ∈ J} = Nat.card {γ : GQ L n // t⁻¹ * (γ : GAf L n) * t ∈ J}

theorem **classSum_indep** (a : NG.AlgWt L n) (β : a.W →ₗ⋆[ℂ] a.W →ₗ[ℂ] ℂ) (hβ : IsInvForm a β) (J : Subgroup (GAf L n)) (χ : J →* ℂˣ) (hχ : ∀ κ, starRingEnd ℂ (χ κ : ℂ) * (χ κ : ℂ) = 1) (T T' : Finset (GAf L n)) (hT : IsClassReps J T) (hT' : IsClassReps J T') (f f' : GAf L n → a.W) (hf : f ∈ NG.TypeSp a.ρ J χ) (hf' : f' ∈ NG.TypeSp a.ρ J χ) : classSum a β J T f f' = classSum a β J T' f f'

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.LubinTateSeriesEqOfCoeffOneEq

theorem **resCard_Kb** : resCard Kb = 2

theorem **bij** : Function.Bijective (algebraMap ℤ_[2] 𝔦(OL Kb))

theorem **irr** : Irreducible (2 : 𝔦(OL Kb))

theorem **coeff_cyc** (k : ℕ) (hk : k ≠ 0) (c : ℤ_[2]) : PowerSeries.coeff k (GlobalTheory.Cyc.cyc 2 𝔦(OL Kb) c) = algebraMap ℤ_[2] 𝔦(OL Kb) (Ring.choose c k)

theorem **commGroup_mul_eq_mul_mul_div** (O : Type) [CommRing O] [Algebra ℤ_[2] O] (c : ℤ_[2]) : PowerSeries.C 2 * GlobalTheory.Cyc.cyc 2 O c + GlobalTheory.Cyc.cyc 2 O c ^ 2 = PowerSeries.subst (PowerSeries.C 2 * PowerSeries.X + PowerSeries.X ^ 2 : PowerSeries O) (GlobalTheory.Cyc.cyc 2 O c)

theorem **endo** (c : ℤ_[2]) : IsLTEndo Kb 2 (algebraMap ℤ_[2] 𝔦(OL Kb) c) (GlobalTheory.Cyc.cyc 2 𝔦(OL Kb) c)

theorem **norm_lt** {ζ : R} {n : ℕ} (h : ζ ^ 2 ^ n = 1) : ‖ζ - 1‖ < 1

theorem **hasSum_C** [CompleteSpace R] {ζ : R} {n : ℕ} (h : ζ ^ 2 ^ n = 1) (c : ℤ_[2]) : HasSum (fun k => algebraMap ℚ_[2] R ((Ring.choose c k : ℤ_[2]) : ℚ_[2]) * (ζ - 1) ^ k) (ζ ^ (PadicInt.toZModPow n c).val)

theorem **hasSum_Om** {ζ : Om 2} {n : ℕ} (h : ζ ^ 2 ^ n = 1) (c : ℤ_[2]) : HasSum (fun k => algebraMap ℚ_[2] (Om 2) ((Ring.choose c k : ℤ_[2]) : ℚ_[2]) * (ζ - 1) ^ k) (ζ ^ (PadicInt.toZModPow n c).val)

theorem **exists_roots** : ∃ ζ : ℕ → Om 2, ζ 0 = 1 ∧ ζ 1 = -1 ∧ ∀ n, ζ (n + 1) ^ 2 = ζ n

theorem **ltData** : ∃ z, LTData Kb cycK 2 z

theorem **two_le_resCard** (K : FinExt l) : 2 ≤ resCard K

theorem **lubinTateSeries_eq_of_coeff_one_eq** {O : Type} [CommRing O] [IsDomain O] {π : O} (hπ : Irreducible π) {q : ℕ} (hq : 2 ≤ q) {φ ψ : PowerSeries O} (h0 : PowerSeries.constantCoeff φ = 0) (h0' : PowerSeries.constantCoeff ψ = 0) (h1 : PowerSeries.coeff 1 φ = PowerSeries.coeff 1 ψ) (h : PowerSeries.C π * φ + φ ^ q = PowerSeries.subst (PowerSeries.C π * PowerSeries.X + PowerSeries.X ^ q : PowerSeries O) φ) (h' : PowerSeries.C π * ψ + ψ ^ q = PowerSeries.subst (PowerSeries.C π * PowerSeries.X + PowerSeries.X ^ q : PowerSeries O) ψ) : φ = ψ

theorem **IsLTEndo_unique** {K : FinExt l} {π a : ɪ(OL K)} (hπ : Irreducible π) {φ ψ : PowerSeries ɪ(OL K)} (h : IsLTEndo K π a φ) (h' : IsLTEndo K π a ψ) : φ = ψ

theorem **IsLTEndo_one** (K : FinExt l) (π : ɪ(OL K)) : IsLTEndo K π 1 PowerSeries.X

theorem **fixed_of_chi_eq_one** {K : FinExt l} {χ : ɪ(GK K) →* (ɪK.1)ˣ} {π : ɪ(OL K)} {z : ℕ → Om l} (h : LTData K χ π z) (g : ɪ(GK K)) (hg : χ g = 1) (n : ℕ) : (g : GQ l) (z n) = z n

theorem **not_ltData_one** (z : ℕ → Om 2) : ¬ LTData Kb 1 2 z

theorem **cycK_ne_one** : cycK ≠ 1

theorem **lineRep_apply** {l : ℕ} [Fact l.Prime] {K LL : FinExt l} (χ : ɪ(GK K) →* (ɪLL.1)ˣ) (g : ɪ(GK K)) (v : Fin 1 → ɪLL.1) : lineRep χ g v = (χ g : ɪLL.1) • v

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MapCSmulMulVec

theorem **valTop_zero** (l : ℕ) {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] : valTop l (0 : O) = ⊤

theorem **valTop_of_ne** (l : ℕ) {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] {c : O} (hc : c ≠ 0) : valTop l c = (val l c : ℝ)

theorem **val_natCast_self** (l : ℕ) {O : Type} [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] (h0 : (l : O) ≠ 0) (h1 : ¬ IsUnit (l : O)) : val l (l : O) = 1

theorem **isUnit_γX_div_X** : IsUnit (1 + (R.κ : O⟦X⟧) * R.E)

theorem **φit_zero** (c : ZMod n) : S.φit c 0 = LinearMap.id

theorem **φit_succ** (c : ZMod n) (j : ℕ) : S.φit c (j + 1) = (S.φl (c + (j : ZMod n) + 1)).comp (S.φit c j)

theorem **φ0it_zero** (c : ZMod n) : S.φ0it c 0 = 1

theorem **φ0it_succ** (c : ZMod n) (j : ℕ) : S.φ0it c (j + 1) = S.φ0 (c + (j : ZMod n) + 1) * S.φ0it c j

theorem **map_C_smul_mulVec** (c : O) (B : Matrix (Fin r') (Fin r) O⟦X⟧) (x : Fin r → R.EE) : (((C c : O⟦X⟧) • B).map R.ι).mulVec x = c • (B.map R.ι).mulVec x

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MatKOSmulOne

theorem **matKO_mk** (A : Matrix (Fin k) (Fin k) O) (v : Fin k → FractionRing O) : matKO A (Submodule.Quotient.mk v) = Submodule.Quotient.mk ((A.map (algebraMap O (FractionRing O))).mulVec v)

theorem **matKO_mul** (A B : Matrix (Fin k) (Fin k) O) (x : KOG O k) : matKO (A * B) x = matKO A (matKO B x)

theorem **matKO_smul_one** (a : O) (x : KOG O k) : matKO (a • (1 : Matrix (Fin k) (Fin k) O)) x = a • x

theorem **WKG_eq_matKO** {Γ : Type} [Group Γ] (M : Γ →* Matrix (Fin k) (Fin k) O) (g : Γ) (x : KOG O k) : WKG M g x = matKO (M g) x

theorem **mem_ccOff_of** {Xs : Set (D.nd.X i)} {π : D.nd.X i} {f : D.nd.X i → D.nd.O i} (hf : f ∈ D.nd.T i) (h : ∀ x ∈ Xs, x ∉ D.nd.Z i → f x = 0) : f π ∈ ccOff i Xs π

theorem **ccOff_univ** (π : D.nd.X i) : ccOff i Set.univ π = ccAt (D.nd.T i) (D.nd.Z i) π

theorem **vccOff_univ** (π : D.nd.X i) : vccOff i Set.univ π = D.nd.vcc i π

theorem **ccOff_anti** {Xs Xs' : Set (D.nd.X i)} (h : Xs ⊆ Xs') (π : D.nd.X i) : ccOff i Xs' π ≤ ccOff i Xs π

theorem **ccOff_eq_top** {Xs : Set (D.nd.X i)} (h : ∀ x ∈ Xs, x ∈ D.nd.Z i) (π : D.nd.X i) : ccOff i Xs π = ⊤

theorem **ccOff_ne_bot** (Xs : Set (D.nd.X i)) {π : D.nd.X i} (hπ : π ∈ D.nd.Z i) : ccOff i Xs π ≠ ⊥

theorem **inR'_one** : InR' D.nd r S 1

theorem **inY_yg** (γ : GE D.nd) : InY D.nd r S (yg D.nd r S γ)

theorem **yg_eq** (γ : GE D.nd) (x' : S.X') : yg D.nd r S γ x' = (rTwO D.nd i r (S.p x') γ - 1) * Eel D.nd S x'

theorem **cocW_unram** {k : ℕ} (ψ : (S.X' → MatN D.nd i) → Matrix (Fin k) (Fin k) (D.nd.O i)) (hU : UnrOut i r Xs) {c : GE D.nd → KOG (D.nd.O i) k} (hc : c ∈ cocW i r S Xs ψ) (v) (h1 : v ∉ placesOver D.host.E (D.nd.l i)) (h2 : v ∉ D.nd.S i) (h3 : v ∉ D.nd.Q i) (γ) (hγ : γ ∈ Ine (F

theorem **cocW_empty** {k : ℕ} (ψ : (S.X' → MatN D.nd i) → Matrix (Fin k) (Fin k) (D.nd.O i)) : cocW i r S ∅ ψ = ⊥

theorem **cocW_mono** {k : ℕ} (ψ : (S.X' → MatN D.nd i) → Matrix (Fin k) (Fin k) (D.nd.O i)) {Xs Xs' : Set (D.nd.X i)} (h : Xs ⊆ Xs') : cocW i r S Xs ψ ≤ cocW i r S Xs' ψ

theorem **cocW_cocycle** {k : ℕ} (ψ : (S.X' → MatN D.nd i) → Matrix (Fin k) (Fin k) (D.nd.O i)) (M : GE D.nd →* Matrix (Fin k) (Fin k) (D.nd.O i)) (hψ : ∀ g, M g = (((chiU D.nd i 1 g)⁻¹ : (D.nd.O i)ˣ) : D.nd.O i) • ψ fun x' => rO D.nd i r (S.p x') g) {c : GE D.nd → KOG (D.nd.O i) k} (hc : c ∈ cocW i r S Xs ψ) (g h : GE D.nd) : c (g * h) = c g + WKG M g (c h)

theorem **cocW_quad** {k : ℕ} (ψ : (S.X' → MatN D.nd i) → Matrix (Fin k) (Fin k) (D.nd.O i)) {v} (hq : QuadAt i r Xs v) {c : GE D.nd → KOG (D.nd.O i) k} (hc : c ∈ cocW i r S Xs ψ) (γ) (hγ : γ ∈ Ine (F

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MatMapTmul

theorem **matMap_tmul** (n : ℕ) (A B : Type) [CommRing A] [CommRing B] [Algebra A B] (b : B) (M : Matrix (Fin n) (Fin n) A) : matMap n A B (b ⊗ₜ M) = b • M.map (algebraMap A B)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MeetsReferenceOfMeetsAll

theorem **heckeVal_mul** (μ δ : HChar D.host.E) (π : D.naut.CG D.R.r) (k : ℕ) (P : PlE D.host) : heckeVal (μ * δ) π k P = (((δ.val P : ℂˣ) : ℂ)⁻¹) ^ k * heckeVal μ π k P

theorem **eig_eq_heckeVal** (h : EigTrue D cf Rw Sb eig) (j : Fin D.R.m) {σ} (h1 : D.what₂.isot (j.1 + 1) σ ≠ ⊥) (h2 : D.nd.goodRow σ) {π} (hπ : GlobalTheory.EigTie.IsRowPar Rw σ π) {P} (hP : IsSplitPl D.host P) (hS : P ∉ Sb) {k} (hk : 1 ≤ k) (hk' : k ≤ D.R.r) : D.what₂.embLL (eig (j.1 + 1) σ k P : D.what₂.LL) = heckeVal (cf.mu0 j) π k P

theorem **eig_twist** (h : EigTrue D cf Rw Sb eig) (l : ℕ) (j : Fin D.R.m) {σ} (h1 : D.what₂.isot (j.1 + 1) σ ≠ ⊥) (h2 : D.nd.goodRow σ) {π} (hπ : GlobalTheory.EigTie.IsRowPar Rw σ π) {P} (hP : IsSplitPl D.host P) (hS : P ∉ Sb) {k} (hk : 1 ≤ k) (hk' : k ≤ D.R.r) : (((((cf.delAt l).val P : ℂˣ) : ℂ)⁻¹) ^ k * D.what₂.embLL (eig (j.1 + 1) σ k P : D.what₂.LL) = heckeVal (muBase cf l j) π k P

theorem **eig_top** (h : EigTrue D cf Rw Sb eig) (hR : RowParam D Rw) (j : Fin D.R.m) {σ} (h1 : D.what₂.isot (j.1 + 1) σ ≠ ⊥) (h2 : D.nd.goodRow σ) {P} (hP : IsSplitPl D.host P) (hS : P ∉ Sb) : ∃ π, GlobalTheory.EigTie.IsRowPar Rw σ π ∧ D.what₂.embLL (eig (j.1 + 1) σ D.R.r P : D.what₂.LL) = (satake (cf.mu0 j) π P).prod

theorem **eig_unique** {eig'} (h : EigTrue D cf Rw Sb eig) (h' : EigTrue D cf Rw Sb eig') (j : Fin D.R.m) {σ} (h1 : D.what₂.isot (j.1 + 1) σ ≠ ⊥) (h2 : D.nd.goodRow σ) {P} (hP : IsSplitPl D.host P) (hS : P ∉ Sb) {k} (hk : 1 ≤ k) (hk' : k ≤ D.R.r) : eig (j.1 + 1) σ k P = eig' (j.1 + 1) σ k P

theorem **exists_eig** (hR : RowParam D Rw) (hS : ∀ k, placesOver D.host.E (D.host.g k) ⊆ (Sb : Set (PlE D.host))) (hv : ∀ (j : Fin D.R.m) σ π, D.what₂.isot (j.1 + 1) σ ≠ ⊥ → D.nd.goodRow σ → GlobalTheory.EigTie.IsRowPar Rw σ π → ∀ k P, IsSplitPl D.host P → P ∉ Sb → ∃ x : ↥D.LL0, D.what₂.embLL (x : D.what₂.LL) = heckeVal (cf.mu0 j) π k P) : ∃ eig, EigTrue D cf Rw Sb eig

theorem **eigOcc_scalar** (S : RatDat D.ntw) (θ) (c : ℕ → ℕ → PlE D.host → ↥D.LL0) (hne : ∀ j, 1 ≤ j → j ≤ D.R.m → goodL D S θ j ≠ ⊥) : EigOcc D S θ (fun j k P => ((c j k P : ↥D.LL0) : D.what₂.LL) • 1) Sb (fun j _ k P => c j k P)

theorem **not_eigOcc** (S : RatDat D.ntw) (θ) (hL) {j} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) {σ} (ho : D.what₂.isot j σ ≠ ⊥) (hg : D.nd.goodRow σ) (hb : goodL D S θ j = ⊥) : ¬ EigOcc D S θ hL Sb eig

theorem **meetsReference_of_meetsAll** (h : MeetsAll D) : GlobalTheory.TripTie.MeetsP0 D

theorem **sig_all_of** (hN : NoFormsOff D) (hM : MeetsAll D) (hC : CommAll D) (hall : ∀ i, ∀ j ∈ Finset.Icc 1 D.R.m, ∀ b ev, GlobalTheory.EigTie.IsEigSys D.nexp i j b ev → ∃ s : D.ntw.Sig i, D.ntw.jS i s = j ∧ D.ntw.side i s = b ∧ ∀ k P, HkIdx i k P → GlobalTheory.EigTie.sigEv s k P = ev k P) : D.nexp.sig_all

theorem **noFormsOff** : NoFormsOff (GlobalTheory.SupLaw.Model.rec D A s)

theorem **meetsAll** : MeetsAll (GlobalTheory.SupLaw.Model.rec D A s)

theorem **commAll** (t) : CommAll (GlobalTheory.SupLaw.Model.rec D (scal D t) s)

theorem **fullAll** : FullAll (GlobalTheory.SupLaw.Model.rec D A s)

theorem **all_laws** : ∃ (D' : D7) (_ : D'.ntw.Tw ≃ D.ntw.Tw), (∀ i, ∀ j ∈ Finset.Icc 1 D'.R.m, ∀ b, Nontrivial (D'.nexp.Fm i j b)) ∧ NoFormsOff D' ∧ MeetsAll D' ∧ CommAll D' ∧ FullAll D'

theorem **not_commAll** (i : D.ntw.Tw) (k P) (hP : HkIdx (D

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MemBallOfGaugeLt

theorem **_root_.GlobalTheory.BSP.ball_mono** {r s : ℝ} (h : r ≤ s) : _root_.GlobalTheory.BSP.ball l u ρ r ≤ _root_.GlobalTheory.BSP.ball l u ρ s

theorem **_root_.GlobalTheory.BSP.mul_mem_ball** {r s : ℝ} (hr : 0 ≤ r) {x y : _root_.GlobalTheory.BSP.PreDisc l u ρ} (hx : x ∈ _root_.GlobalTheory.BSP.ball l u ρ r) (hy : y ∈ _root_.GlobalTheory.BSP.ball l u ρ s) : x * y ∈ _root_.GlobalTheory.BSP.ball l u ρ (r * s)

theorem **exists_mem_ball** (y : PreDisc l u ρ) : ∃ r : ℝ, 0 < r ∧ y ∈ ball l u ρ r

theorem **_root_.GlobalTheory.BSP.gauge_nonneg** (y : _root_.GlobalTheory.BSP.PreDisc l u ρ) : 0 ≤ _root_.GlobalTheory.BSP.gauge l u ρ y

theorem **_root_.GlobalTheory.BSP.gauge_le** {y : _root_.GlobalTheory.BSP.PreDisc l u ρ} {r : ℝ} (hr : 0 < r) (h : y ∈ _root_.GlobalTheory.BSP.ball l u ρ r) : _root_.GlobalTheory.BSP.gauge l u ρ y ≤ r

private lemma **preDisc_exists_eq_algebraMap_mul_invSelf_pow** (y : _root_.GlobalTheory.BSP.PreDisc l u ρ) : ∃ (a : A) (n : ℕ), y = algebraMap A _ a * IsLocalization.Away.invSelf (l : A) ^ n

private lemma **preDisc_exists_pos_mem_ball** (y : _root_.GlobalTheory.BSP.PreDisc l u ρ) : ∃ r : ℝ, 0 < r ∧ y ∈ _root_.GlobalTheory.BSP.ball l u ρ r

theorem **_root_.GlobalTheory.BSP.mem_ball_of_gauge_lt** {y : _root_.GlobalTheory.BSP.PreDisc l u ρ} {r : ℝ} (h : _root_.GlobalTheory.BSP.gauge l u ρ y < r) : y ∈ _root_.GlobalTheory.BSP.ball l u ρ r

theorem **_root_.GlobalTheory.BSP.gauge_add_le_max** (x y : _root_.GlobalTheory.BSP.PreDisc l u ρ) : _root_.GlobalTheory.BSP.gauge l u ρ (x + y) ≤ max (_root_.GlobalTheory.BSP.gauge l u ρ x) (_root_.GlobalTheory.BSP.gauge l u ρ y)

theorem **norm_preDisc** (y : PreDisc l u ρ) : ‖y‖ = gauge l u ρ y

theorem **_root_.GlobalTheory.BSP.norm_ofPre** (y : _root_.GlobalTheory.BSP.PreDisc l u ρ) : ‖_root_.GlobalTheory.BSP.ofPre l u ρ y‖ = _root_.GlobalTheory.BSP.gauge l u ρ y

theorem **_root_.GlobalTheory.BSP.denseRange_ofPre** : DenseRange (_root_.GlobalTheory.BSP.ofPre l u ρ)

theorem **_root_.GlobalTheory.BSP.isUnit_l_disc** : IsUnit ((l : ℕ) : _root_.GlobalTheory.BSP.Disc l u ρ)

theorem **_root_.GlobalTheory.BSP.isDiscMap_id** : _root_.GlobalTheory.BSP.IsDiscMap (RingHom.id A) (RingHom.id (_root_.GlobalTheory.BSP.Disc l u ρ))

theorem **_root_.GlobalTheory.BSP.norm_algebraMap_le** (a : A) : ‖algebraMap A (_root_.GlobalTheory.BSP.PreDisc l u ρ) a‖ ≤ 1

theorem **_root_.GlobalTheory.BSP.norm_toDisc_le** (a : A) : ‖_root_.GlobalTheory.BSP.toDisc l u ρ a‖ ≤ 1

theorem **_root_.GlobalTheory.BSP.preMap_alg** (f : A →+* A') (a : A) : _root_.GlobalTheory.BSP.preMap l u ρ u' ρ' f (algebraMap A _ a) = algebraMap A' _ (f a)

theorem **_root_.GlobalTheory.BSP.preMap_inv** (f : A →+* A') : _root_.GlobalTheory.BSP.preMap l u ρ u' ρ' f (IsLocalization.Away.invSelf (l : A)) = IsLocalization.Away.invSelf (l : A')

private lemma **preDisc_norm_pow_le_norm_pow** (x : _root_.GlobalTheory.BSP.PreDisc l u ρ) (k : ℕ) : ‖x ^ k‖ ≤ ‖x‖ ^ k

theorem **_root_.GlobalTheory.BSP.norm_preMap_le** [hl : Fact l.Prime] (f : A →+* A') (h : ‖_root_.GlobalTheory.BSP.toDisc l u' ρ' (f u)‖ ≤ ρ) (y : _root_.GlobalTheory.BSP.PreDisc l u ρ) : ‖_root_.GlobalTheory.BSP.preMap l u ρ u' ρ' f y‖ ≤ ‖y‖

theorem **_root_.GlobalTheory.BSP.isDiscMap_map** [Fact l.Prime] (f : A →+* A') (h : ‖_root_.GlobalTheory.BSP.toDisc l u' ρ' (f u)‖ ≤ ρ) : _root_.GlobalTheory.BSP.IsDiscMap f (_root_.GlobalTheory.BSP.Disc.map f h)

theorem **_root_.GlobalTheory.BSP.IsDiscMap.unique** {f : A →+* A'} {F G : _root_.GlobalTheory.BSP.Disc l u ρ →+* _root_.GlobalTheory.BSP.Disc l u' ρ'} (hF : _root_.GlobalTheory.BSP.IsDiscMap f F) (hG : _root_.GlobalTheory.BSP.IsDiscMap f G) : F = G

theorem **_root_.GlobalTheory.BSP.norm_toDisc_pow_le** (k : ℕ) (hρ : 0 < ρ) : ‖_root_.GlobalTheory.BSP.toDisc l u ρ (u ^ k)‖ ≤ (ρ : ℝ) ^ k

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MemCentIffTwConj

theorem **Phi_apply'** (i j : Fin n) : Phi n R i j = if j = i.rev then (-1) ^ (i : ℕ) else 0

theorem **Phi_mul_transpose** : Phi n R ∗ (Phi n R)ᵀ = 1

theorem **transpose_mul_Phi** : (Phi n R)ᵀ ∗ Phi n R = 1

theorem **Phi_conjTranspose** : (Phi n R)ᴴ = (Phi n R)ᵀ

theorem **Phi_transpose** : (Phi n R)ᵀ = ((-1 : R) ^ (n - 1)) • Phi n R

theorem **theta_one** : theta (1 : Matrix (Fin n) (Fin n) R) = 1

theorem **theta_mul** (g h : Matrix (Fin n) (Fin n) R) : theta (g ∗ h) = theta g ∗ theta h

theorem **theta_mul_theta_inv** {g : Matrix (Fin n) (Fin n) R} (hg : IsUnit g) : theta g ∗ theta g⁻¹ = 1

theorem **theta_inv_eq** {g : Matrix (Fin n) (Fin n) R} (hg : IsUnit g) : (theta g)⁻¹ = Phi n R ∗ gᴴ ∗ (Phi n R)ᵀ

theorem **theta_theta** {g : Matrix (Fin n) (Fin n) R} (hg : IsUnit g) : theta (theta g) = g

theorem **nrm_twConj** {x δ : Matrix (Fin n) (Fin n) R} (hx : IsUnit x) : nrm (twConj x δ) = x⁻¹ ∗ nrm δ ∗ x

theorem **adj_eq** {δ g : Matrix (Fin n) (Fin n) R} (hδ : IsUnit δ) (hg : IsUnit g) : adj δ g = (δ ∗ theta g ∗ δ⁻¹)⁻¹

theorem **mem_cent_iff_twConj** {δ g : Matrix (Fin n) (Fin n) R} (hg : IsUnit g) : g ∈ cent δ ↔ twConj g δ = δ

theorem **Phi_two** : Phi 2 ℂ = !![0, 1; -1, 0]

theorem **inv_diag** : (!![2, 0; 0, 1] : Matrix (Fin 2) (Fin 2) ℂ)⁻¹ = !![1/2, 0; 0, 1]

theorem **nrm_diag** : nrm (!![2, 0; 0, 1] : Matrix (Fin 2) (Fin 2) ℂ) = Matrix.diagonal ![2, 1/2]

theorem **reg_diag** : Reg (!![2, 0; 0, 1] : Matrix (Fin 2) (Fin 2) ℂ)

theorem **not_reg_one** : ¬ Reg (1 : Matrix (Fin 2) (Fin 2) ℂ)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MemEurOfTeichmullerSeries

theorem **cyc_const** (c : ℤ_[l]) : PowerSeries.constantCoeff (cyc l O c) = 0

theorem **cyc_eq** (c : ℤ_[l]) : cyc l O c = PowerSeries.binomialSeries O c - 1

theorem **act_X** (c : ℤ_[l]) : act l O c PowerSeries.X = cyc l O c

theorem **act_eq** (c : ℤ_[l]) : act l O c = Fon.gam O c

theorem **phi_eq** : phi l O = Fon.phi0 l O

theorem **cyc_natCast** (k : ℕ) : cyc l O (k : ℤ_[l]) = (1 + PowerSeries.X) ^ k - 1

theorem **act_act** (c c' : ℤ_[l]) (a : PowerSeries O) : act l O c (act l O c' a) = act l O (c ∗ c') a

theorem **act_one** (a : PowerSeries O) : act l O 1 a = a

theorem **phi_act** (c : ℤ_[l]) (a : PowerSeries O) : phi l O (act l O c a) = act l O c (phi l O a)

theorem **phi_X** : phi l O PowerSeries.X = (1 + PowerSeries.X) ^ l - 1

theorem **phi_pow_X** (s : ℕ) : (phi l O ^ (s - 1)) PowerSeries.X = XX l O s

theorem **phi_XX** (s : ℕ) : phi l O (XX l O s) = XX l O s ∗ QQ l O s

theorem **act_X_unit** (c : ℤ_[l]ˣ) : ∃ v, IsUnit v ∧ act l O c PowerSeries.X = PowerSeries.X ∗ v

theorem **phi_X_mod** : ∃ r, phi l O PowerSeries.X = PowerSeries.X ^ l + (l : PowerSeries O) ∗ r

private theorem **dvd_iff_eval** (S : Type) [CommRing S] (k : ℕ) (x : AdicCompletion (Ideal.span {(l : S)}) S) : (l : AdicCompletion (Ideal.span {(l : S)}) S) ^ k ∣ x ↔ AdicCompletion.eval (Ideal.span {(l : S)}) S k x = 0

theorem **isUnit_toEE_π** : IsUnit (toEE l O n letπ)

theorem **EE_dense** (x : EE l O n) (k : ℕ) : ∃ (a : LetA O n) (m : ℕ), (l : EE l O n) ^ k ∣ x ∗ toEE l O n (letπ ^ m) - toEE l O n a

theorem **EE_hom_ext** {B : Type} [CommRing B] [IsHausdorff (Ideal.span {(l : B)}) B] {f g : EE l O n →+* B} (h : ∀ a, f (toEE l O n a) = g (toEE l O n a)) : f = g

@[simp] theorem **liftEE_toEE** {B : Type} [CommRing B] [Fon.LComplete l B] (f : LetA O n →+* B) (hf : IsUnit (f letπ)) (a : LetA O n) : liftEE (l

@[simp] theorem **extEE_toEE** (f : LetA O n →ₐ[O] LetA O n) (hf : IsUnit (toEE l O n (f letπ))) (a : LetA O n) : extEE f hf (toEE l O n a) = toEE l O n (f a)

theorem **phiEE_toEE** (a : LetA O n) : phiEE l O n (toEE l O n a) = toEE l O n (letφ (phi l O) a)

private theorem **actEE1_toEE** (c : ℤ_[l]ˣ) (a : LetA O n) : actEE1 l O n c (toEE l O n a) = toEE l O n (letγ (act l O c) a)

theorem **actEE_toEE** (c : ℤ_[l]ˣ) (a : LetA O n) : actEE l O n c (toEE l O n a) = toEE l O n (letγ (act l O c) a)

theorem **toPiEE_bijective** [NeZero n] : Function.Bijective (toPiEE l O n)

theorem **mem_clW** {S : Set (WCb l)} {x : WCb l} : x ∈ clW l S ↔ ∀ k : ℕ, ∃ y ∈ S, (l : WCb l) ^ k ∣ x - y

theorem **mem_clAlg** {S : Subalgebra ℤ_[l] (WCb l)} {x : WCb l} : x ∈ clAlg S ↔ ∀ k : ℕ, ∃ y ∈ S, (l : WCb l) ^ k ∣ x - y

theorem **le_clAlg** (S : Subalgebra ℤ_[l] (WCb l)) : S ≤ clAlg S

theorem **clAlg_clAlg** (S : Subalgebra ℤ_[l] (WCb l)) : clAlg (clAlg S) = clAlg S

theorem **map_mem_clAlg** {S S' : Subalgebra ℤ_[l] (WCb l)} (σ : WCb l →+* WCb l) (h : ∀ x ∈ S, σ x ∈ clAlg S') {x : WCb l} (hx : x ∈ clAlg S) : σ x ∈ clAlg S'

theorem **inv_mem_clAlg** {S : Subalgebra ℤ_[l] (WCb l)} {r w : WCb l} (hr : r ∈ S) (hw : (1 - l * r) * w = 1) : w ∈ clAlg S

theorem **IsIo.isUnit_X** {e : OCb l} {io : PowerSeries ℤ_[l] →ₐ[ℤ_[l]] Ainf l} (h : IsIo e io) (he : e ≠ 1) : IsUnit (inclW l (io PowerSeries.X))

theorem **εroot_pow** {P : LocPrime l} (AI : LocAinf l P) (s : ℕ) : AI.εroot s ^ l ^ s = AI.ε

theorem **εroot_ne_one** {P : LocPrime l} (AI : LocAinf l P) (s : ℕ) : AI.εroot s ≠ 1

theorem **εroot_sub_one_not_unit** {P : LocPrime l} (AI : LocAinf l P) (s : ℕ) : ¬ IsUnit (AI.εroot s - 1)

private theorem **coeff_mem** {A : Type} [CommRing A] [Algebra ℤ_[l] A] (S : Subalgebra ℤ_[l] A) {a : A} (ha : a ∈ S) (m k i : ℕ) : (Polynomial.X ^ m - Polynomial.C a * Polynomial.X ^ k : Polynomial A).coeff i ∈ S

theorem **io_mem_E0** (a : PowerSeries ℤ_[l]) : inclW l (io a) ∈ E0 io

theorem **inv_X_mem_E0** {w : WCb l} (hw : inclW l (io PowerSeries.X) * w = 1) : w ∈ E0 io

theorem **clAlg_E0** : clAlg (E0 io) = E0 io

theorem **clAlg_Eur** : clAlg (Eur io) = Eur io

theorem **simpleRoots_subset_Eur** : simpleRoots io ⊆ Eur io

theorem **E0_le_Eur** : E0 io ≤ Eur io

theorem **map_mem_E0** (σ : WCb l →+* WCb l) (f : PowerSeries ℤ_[l] → PowerSeries ℤ_[l]) (hσ : ∀ a, σ (inclW l (io a)) = inclW l (io (f a))) (hw : ∃ w ∈ E0 io, inclW l (io (f PowerSeries.X)) * w = 1) {x : WCb l} (hx : x ∈ E0 io) : σ x ∈ E0 io

theorem **map_mem_E0_act** (hX : IsUnit (inclW l (io PowerSeries.X))) (σ : WCb l →+* WCb l) (c : ℤ_[l]ˣ) (hσ : ∀ a, σ (inclW l (io a)) = inclW l (io (act l ℤ_[l] c a))) {x : WCb l} (hx : x ∈ E0 io) : σ x ∈ E0 io

theorem **map_mem_E0_phi** (hX : IsUnit (inclW l (io PowerSeries.X))) (σ : WCb l →+* WCb l) (hσ : ∀ a, σ (inclW l (io a)) = inclW l (io (phi l ℤ_[l] a))) {x : WCb l} (hx : x ∈ E0 io) : σ x ∈ E0 io

theorem **map_mem_Eur** (σ : WCb l →ₐ[ℤ_[l]] WCb l) (hσ : ∀ x ∈ E0 io, σ x ∈ E0 io) {x : WCb l} (hx : x ∈ Eur io) : σ x ∈ Eur io

theorem **teichmuller_mem_Eur** {a : Cb l} {d : ℕ} (hd : 0 < d) (ha : a ^ l ^ d = a) : WittVector.teichmuller l a ∈ Eur io

theorem **mem_Eur_of_teichmuller_series** (a : ℕ → Cb l) (ha : ∀ n, ∃ d, 0 < d ∧ a n ^ l ^ d = a n) (x : WCb l) (hx : ∀ k, (l : WCb l) ^ k ∣ x - ∑ n ∈ Finset.range k, (l : WCb l) ^ n * WittVector.teichmuller l (a n)) : x ∈ Eur io

private theorem **map_smul_gen** {O R M : Type} [CommRing O] [CommRing R] [Algebra O R] [AddCommGroup M] [Module O M] (f : R →ₐ[O] R) (g : M →ₗ[O] M) (r : R) (y : R ⊗[O] M) : TensorProduct.map f.toLinearMap g (r • y) = f r • TensorProduct.map f.toLinearMap g y

private theorem **smul_mem_gen** {Z O W M : Type} [CommRing Z] [CommRing O] [CommRing W] [Algebra Z O] [Algebra Z W] [AddCommGroup M] [Module O M] (S : Subalgebra Z W) (o : O) {w : W} (hw : w ∈ S) {z : (O ⊗[Z] W) ⊗[O] M} (hz : z ∈ Submodule.span O {z | ∃ w ∈ S, ∃ t : M, z = ((1 : O) ⊗ₜ[Z] w) ⊗ₜ[O] t}) : (o ⊗ₜ[Z] w : O ⊗[Z] W) • z ∈ Submodule.span O {z | ∃ w ∈ S, ∃ t : M, z = ((1 : O) ⊗ₜ[Z] w) ⊗ₜ[O] t}

private theorem **map_map_gen** {Z O W M : Type} [CommRing Z] [CommRing O] [CommRing W] [Algebra Z O] [Algebra Z W] [AddCommGroup M] [Module O M] {a b c d : W →ₐ[Z] W} {f g h k : M →ₗ[O] M} (h1 : ∀ w, a (b w) = c (d w)) (h2 : ∀ t, f (g t) = h (k t)) (y : (O ⊗[Z] W) ⊗[O] M) : TensorProduct.map (Algebra.TensorProduct.map (AlgHom.id O O) a).toLinearMap f (TensorProduct.map (Algebra.TensorProduct.map (AlgHom.id O O) b).toLinearMap g y) = TensorProduct.map (Algebra.TensorProduct.map (AlgHom.id O O) c).toLinearMap h (TensorProduct.map (Algebra.TensorProduct.map (AlgHom.id O O) d).toLinearMap k y)

private theorem **map_comm_gen** {O R M M' : Type} [CommRing O] [CommRing R] [Algebra O R] [AddCommGroup M] [Module O M] [AddCommGroup M'] [Module O M'] (a : R →ₐ[O] R) (f : M →ₗ[O] M') (g : M →ₗ[O] M) (g' : M' →ₗ[O] M') (h : ∀ t, f (g t) = g' (f t)) (y : R ⊗[O] M) : TensorProduct.map a.toLinearMap g' (TensorProduct.map LinearMap.id f y) = TensorProduct.map LinearMap.id f (TensorProduct.map a.toLinearMap g y)

theorem **tmul_mem_EurT** {w : WCb l} (hw : w ∈ Eur io) (t : ɪT.L.T) : ((1 : ɪ(OL LL)) ⊗ₜ[ℤ_[l]] w) ⊗ₜ[ɪ(OL LL)] t ∈ EurT LL io T

theorem **map_mem_EurT** {T' : LocLat G ℚ_[l] ɪLL.1 ɪ(OL LL)} (σ : WCb l →ₐ[ℤ_[l]] WCb l) (hσ : ∀ x ∈ Eur io, σ x ∈ Eur io) (f : ɪT.L.T →ₗ[ɪ(OL LL)] ɪT'.L.T) {z : TW (RW LL) T} (hz : z ∈ EurT LL io T) : TensorProduct.map (Algebra.TensorProduct.map (AlgHom.id ɪ(OL LL) ɪ(OL LL)) σ).toLinearMap f z ∈ EurT LL io T'

theorem **smul_mem_EurT** (o : ɪ(OL LL)) {w : WCb l} (hw : w ∈ Eur io) {z : TW (RW LL) T} (hz : z ∈ EurT LL io T) : (o ⊗ₜ[ℤ_[l]] w : RW LL) • z ∈ EurT LL io T

theorem **mem_FonD** {z : TW (RW LL) T} : z ∈ FonD AI K LL io T ↔ z ∈ EurT LL io T ∧ ∀ g : ɪ(GK K), P.χQ (g : GQ l) = 1 → TWact (AI.gRW LL (g : GQ l)) T g z = z

theorem **tmul_mem_FonD** {w : WCb l} (hw : w ∈ Eur io) (hwH : ∀ g : ɪ(GK K), P.χQ (g : GQ l) = 1 → AI.actWCbHom (g : GQ l) w = w) {t : ɪT.L.T} (ht : ∀ g : ɪ(GK K), P.χQ (g : GQ l) = 1 → T.actT g t = t) : ((1 : ɪ(OL LL)) ⊗ₜ[ℤ_[l]] w) ⊗ₜ[ɪ(OL LL)] t ∈ FonD AI K LL io T

theorem **smul_mem_FonD** (o : ɪ(OL LL)) {w : WCb l} (hw : w ∈ Eur io) (hwH : ∀ g : ɪ(GK K), P.χQ (g : GQ l) = 1 → AI.actWCbHom (g : GQ l) w = w) {z : TW (RW LL) T} (hz : z ∈ FonD AI K LL io T) : (o ⊗ₜ[ℤ_[l]] w : RW LL) • z ∈ FonD AI K LL io T

theorem **phi_mem_FonD** (hφ : ∀ x ∈ E0 io, frobW l x ∈ E0 io) {z : TW (RW LL) T} (hz : z ∈ FonD AI K LL io T) : TWmap (φRW LL) T z ∈ FonD AI K LL io T

theorem **act_mem_EurT** (g : ɪ(GK K)) (hg : ∀ x ∈ E0 io, AI.actWCbHom (g : GQ l) x ∈ E0 io) {z : TW (RW LL) T} (hz : z ∈ EurT LL io T) : TWact (AI.gRW LL (g : GQ l)) T g z ∈ EurT LL io T

private theorem **act_act_TW** {g h h' : ɪ(GK K)} (e : h * g = g * h') (z : TW (RW LL) T) : TWact (AI.gRW LL (h : GQ l)) T h (TWact (AI.gRW LL (g : GQ l)) T g z) = TWact (AI.gRW LL (g : GQ l)) T g (TWact (AI.gRW LL (h' : GQ l)) T h' z)

theorem **act_mem_FonD** (g : ɪ(GK K)) (hg : ∀ x ∈ E0 io, AI.actWCbHom (g : GQ l) x ∈ E0 io) {z : TW (RW LL) T} (hz : z ∈ FonD AI K LL io T) : TWact (AI.gRW LL (g : GQ l)) T g z ∈ FonD AI K LL io T

theorem **map_mem_FonD** {T' : LocLat ɪ(GK K) ℚ_[l] ɪLL.1 ɪ(OL LL)} (f : ɪT.L.T →ₗ[ɪ(OL LL)] ɪT'.L.T) (hf : ∀ g t, f (T.actT g t) = T'.actT g (f t)) {z : TW (RW LL) T} (hz : z ∈ FonD AI K LL io T) : TensorProduct.map (LinearMap.id : RW LL →ₗ[ɪ(OL LL)] RW LL) f z ∈ FonD AI K LL io T'

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MemRootsBarXSubC

theorem **exists_red** : Nonempty (ZC →+* Fbar g)

theorem **N_pos** {d : ℕ} (hd : 0 < d) : 0 < g ^ d - 1

theorem **N_ne** {d : ℕ} (hd : 0 < d) : ((g ^ d - 1 : ℕ) : Fbar g) ≠ 0

theorem **exp_int** (n : ℕ) : IsIntegral ℤ (Complex.exp (2 * Real.pi * Complex.I / n))

theorem **ez_prim** {n : ℕ} (hn : 0 < n) : IsPrimitiveRoot (ez n) n

theorem **gen_prim** {d : ℕ} (hd : 0 < d) : IsPrimitiveRoot (red g (ez (g ^ d - 1))) (g ^ d - 1)

theorem **ez_pow** {m n : ℕ} (hn : 0 < n) (h : m ∣ n) : ez n ^ (n / m) = ez m

theorem **N_dvd** (d e : ℕ) : g ^ d - 1 ∣ g ^ (d * e) - 1

theorem **lifts_unique** {x : Fbar g} {ω ω' : ZC} (h : Lifts g x ω) (h' : Lifts g x ω') : ω = ω'

theorem **exists_level** {x : Fbar g} (hx : x ≠ 0) : ∃ d, 0 < d ∧ x ^ (g ^ d - 1) = 1

theorem **exists_lifts** {x : Fbar g} (hx : x ≠ 0) : ∃ ω, Lifts g x ω

theorem **lifts_mul** {x y : Fbar g} {ω ω' : ZC} (h : Lifts g x ω) (h' : Lifts g y ω') : Lifts g (x * y) (ω * ω')

theorem **tf_spec** {x : Fbar g} (hx : x ≠ 0) : Lifts g x (tf g x)

theorem **tf_zero** : tf g 0 = 0

theorem **teichM_inj** (x y : Fbar g) (hx : x ≠ 0) (hy : y ≠ 0) (h : teichM g x = teichM g y) : x = y

theorem **thetaOf_gen_pow** (L : LabelDatum g) {d : ℕ} (hd : 0 < d) {x : Fbar g} (hx : x ^ (g ^ d - 1) = 1) (n : ℕ) : thetaOf L d x (L.gen d ^ n) = L.teich x ^ n

theorem **thetaOf_norm** (L : LabelDatum g) {d e : ℕ} (hd : 0 < d) (he : 0 < e) {x : Fbar g} (hx : x ^ (g ^ d - 1) = 1) (n : ℕ) : thetaOf L (d * e) x (L.gen (d * e) ^ n) = thetaOf L d x ((L.gen (d * e) ^ n) ^ ((g ^ (d * e) - 1) / (g ^ d - 1)))

theorem **cuspOf_prod** (L : LabelDatum g) (Q : List ((ZMod g)[X] × Pc g)) (h : ∀ p ∈ Q, IsCuspOf L p.1 p.2.2) : (Q.map Prod.fst).prod.Monic ∧ (Q.map Prod.fst).prod.natDegree = degL (Q.map Prod.snd) ∧ (Q.map Prod.fst).prod.coeff 0 ≠ 0

theorem **eSerC_zero** (L : LabelDatum g) (n : ℕ) (f : (ZMod g)[X]) (hf : ¬ (f.Monic ∧ f.natDegree = n ∧ f.coeff 0 ≠ 0)) : eSerC L n f = 0

theorem **toG_coe** {n : ℕ} (x : Gn g n) : toG (x : Mt g n) = x

theorem **tl_self** {n : ℕ} (M : Mt g n) : tl n M = M

theorem **hc_full** {n : ℕ} (α : Gn g n → ℂ) (hα : IsClass α) : hc n 0 α (fun _ => 1) n = α

theorem **isIrr_one** {G : Type} [Group G] [Fintype G] : IsIrr (fun _ : G => (1 : ℂ))

theorem **mem_rootsBar_X_sub_C** {a : ZMod g} {z : Fbar g} (hz : z ∈ rootsBar (X - C a)) : z = algebraMap (ZMod g) (Fbar g) a

theorem **isCuspOf_one** (L : LabelDatum g) : IsCuspOf L (X - C 1) (fun _ : Gn g 1 => (1 : ℂ))

theorem **hasLabel_one** (L : LabelDatum g) : HasLabel L (fun _ : Gn g 1 => (1 : ℂ)) (X - C 1)

theorem **label_unique** (hG : Green) (L : LabelDatum g) {d : ℕ} (hE : ∀ χ χ' : Gn g d → ℂ, IsCusp χ → IsCusp χ' → (∀ y : Gn g d, Irreducible (y : Mt g d).charpoly → χ y = χ' y) → χ = χ') {P P' : (ZMod g)[X]} {χ : Gn g d → ℂ} (h : IsCuspOf L P χ) (h' : IsCuspOf L P' χ) : P = P'

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.MkMemAdmissibleAtIffCocycle

theorem **mem_block** {w u : ℤ} {t : ℕ} : u ∈ block w t ↔ w ≤ u ∧ u < w + t

theorem **card_block** (w : ℤ) (t : ℕ) : Multiset.card (block w t) = t

theorem **card_U** (i : ZMod n) : Multiset.card (D.U i) = r

theorem **card_Ut** (i : ZMod n) : Multiset.card (D.Ut i) = r + D.t

theorem **mem_Ut** (i : ZMod n) {u : ℤ} (hu : u ∈ D.Ut i) : 1 ≤ u ∧ u ≤ (D.h i : ℤ) - 1

theorem **exData_sg** (i : ZMod 2) : exData.sg 41 i = 1

theorem **exData_T1_T2_T3_fortyOne** : exData.T1 41 ∧ exData.T2 41 ∧ exData.T3 41

theorem **SameFactors.refl** (h : IsFiniteLength R M) : SameFactors R M M

theorem **SameFactors.symm** (h : SameFactors R M N) : SameFactors R N M

theorem **SameFactors.trans** (h : SameFactors R M N) (h' : SameFactors R N P) : SameFactors R M P

theorem **SameFactors.of_equiv** (h : IsFiniteLength R M) (e : M ≃ₗ[R] N) : SameFactors R M N

theorem **SameFactors.length_eq** (h : SameFactors R M N) : Module.length R M = Module.length R N

@[simp] theorem **modRep_mk** (π : Representation O G M) (c : O) (g : G) (x : M) : modRep π c g (Submodule.Quotient.mk x) = Submodule.Quotient.mk (π g x)

@[simp] theorem **piRep_apply** {ι : Type} {N : ι → Type} [∀ j, AddCommGroup (N j)] [∀ j, Module O (N j)] (π : ∀ j, Representation O G (N j)) (g : G) (x : ∀ j, N j) (j : ι) : piRep π g x j = π j g (x j)

theorem **twistRep_apply** (π : Representation O G M) (χ : G →* Oˣ) (g : G) (x : M) : twistRep π χ g x = (χ g : O) • π g x

theorem **coefRep_tmul** (π : Representation O G M) (X : Type) [AddCommGroup X] [Module O X] (g : G) (m : M) (x : X) : coefRep π X g (m ⊗ₜ[O] x) = π g m ⊗ₜ[O] x

theorem **twist_twCoef** (W : Representation O G M) (χ : G →* Oˣ) (X : Type) [AddCommGroup X] [Module O X] : twistRep (twCoef W χ X) χ = coefRep W X

theorem **smul_tensor_eq_zero** {X : Type} [AddCommGroup X] [Module O X] {c : O} (hc : ∀ x : X, c • x = 0) (y : M ⊗[O] X) : c • y = 0

@[simp] theorem **cycMap_mk** {c : O} (hc : ∀ m : M, c • m = 0) (m : M) (y : O) : cycMap hc m (Ideal.Quotient.mk _ y) = y • m

theorem **cocycle_mul** {π : Representation O G M} (z : ↥(Z1 π ⊤)) (g h : G) : (z : G → M) (g * h) = (z : G → M) g + π g ((z : G → M) h)

theorem **cocycle_one** {π : Representation O G M} (z : ↥(Z1 π ⊤)) : (z : G → M) 1 = 0

theorem **extRep_apply** (π : Representation O G M) {c : O} (hc : ∀ m : M, c • m = 0) (z : ↥(Z1 π ⊤)) (g : G) (m : M) (y : O) : extRep π hc z g (m, Ideal.Quotient.mk _ y) = (π g m + y • (z : G → M) g, Ideal.Quotient.mk _ y)

theorem **Ea_apply** (z : ↥(Z1 (twCoef W χ X) ⊤)) (g : G) (m : M ⊗[O] X) (y : O) : Ea W χ hc z g (m, Ideal.Quotient.mk _ y) = (coefRep W X g m + ((χ g : O) * y) • (z : G → M ⊗[O] X) g, Ideal.Quotient.mk _ ((χ g : O) * y))

theorem **Ea_inl** (z : ↥(Z1 (twCoef W χ X) ⊤)) (g : G) (m : M ⊗[O] X) : Ea W χ hc z g (m, 0) = (coefRep W X g m, 0)

theorem **Ea_gen** (z : ɪ(Z1 (twCoef W χ X) ⊤)) (g : G) : Ea W χ hc z g (0, 1) = (χ g : O) • (((0 : M ⊗[O] X), (1 : O ⧸ Ideal.span {c})) + ((z : G → M ⊗[O] X) g, 0))

theorem **Ea_snd** (z : ɪ(Z1 (twCoef W χ X) ⊤)) (g : G) (x : (M ⊗[O] X) × (O ⧸ Ideal.span {c})) : (Ea W χ hc z g x).2 = (χ g : O) • x.2

theorem **Ea_equiv_of_class_eq** {z z' : ɪ(Z1 (twCoef W χ X) ⊤)} (h : (Submodule.Quotient.mk z : H1 (twCoef W χ X) ⊤) = Submodule.Quotient.mk z') : ∃ e : ((M ⊗[O] X) × (O ⧸ Ideal.span {c})) ≃ₗ[O] ((M ⊗[O] X) × (O ⧸ Ideal.span {c})), (∀ g x, e (Ea W χ hc z g x) = Ea W χ hc z' g (e x)) ∧ (∀ m, e (m, 0) = (m, 0)) ∧ ∀ x, (e x).2 = x.2

theorem **mk_mem_unr_of_trivial** {π : Representation O G M} (hπ : ∀ g m, π g m = m) (I : Subgroup G) (z : ɪ(Z1 π ⊤)) : (Submodule.Quotient.mk z : H1 π ⊤) ∈ unr π I ↔ ∀ g ∈ I, (z : G → M) g = 0

theorem **toAmb_injective** (π : Representation O G M) : Function.Injective (toAmb π)

@[simp] theorem **chiLL_apply** (χ : G →* Oˣ) (g : G) : (chiLL LL χ g : LL) = algebraMap O LL (χ g)

@[simp] theorem **latRep_apply** (L : Lattice (O

theorem **latRep_twLattice** (L : Lattice (O

theorem **charLattice_equiv** (hinj : Function.Injective (algebraMap O LL)) (χ : G →* Oˣ) : ∃ e : O ≃ₗ[O] ɪ(charLattice hinj χ).T, (∀ a, (e a : LL) = algebraMap O LL a) ∧ ∀ g a, e (scalRep O χ g a) = latRep (charLattice hinj χ) g (e a)

@[simp] theorem **chi_one** : Γ.chi 1 = Γ.χ₁

theorem **chi_succ** (c : ℕ) : Γ.chi (c + 1) = Γ.chi c * Γ.ε⁻¹

theorem **Full.withLat** {T : LocLat G Qp LL O} (h : Full PD emb Γ t Ut T) (L : Lattice (O

theorem **SumFull.withLat** {μ : ℕ} {T : LocLat G Qp LL O} (h : SumFull PD emb Γ t Ut μ T) (L : Lattice (O

theorem **Full.sumFull** {T : LocLat G Qp LL O} (h : Full PD emb Γ t Ut T) : SumFull PD emb Γ t Ut 1 T

theorem **Admissible.of_equiv** {M M' : Type} [AddCommGroup M] [Module O M] [AddCommGroup M'] [Module O M'] {π : Representation O G M} {π' : Representation O G M'} (h : Admissible PD emb Γ t Ut π) (e : M ≃ₗ[O] M') (he : ∀ g x, e (π g x) = π' g (e x)) : Admissible PD emb Γ t Ut π'

theorem **admissible_zero** [IsTopologicalRing LL] {M : Type} [AddCommGroup M] [Module O M] [Subsingleton M] (π : Representation O G M) : Admissible PD emb Γ t Ut π

theorem **admissible_quotient** {μ : ℕ} {T : LocLat G Qp LL O} (hT : SumFull PD emb Γ t Ut μ T) (L₂ : Lattice (O

theorem **mk_mem_admissibleAt_iff_cocycle** {a : ℕ} (ha : ∀ x : X, Γ.lam ^ a • x = 0) (z : ɪ(Z1 (twCoef W Γ.χ₁ X) ⊤)) : Submodule.Quotient.mk z ∈ admAt PD emb Γ t Ut W X a ha ↔ Admissible PD emb Γ t Ut (Ea W Γ.χ₁ ha z)

theorem **tensAmb_tmul** (c : ɪ(latticeCocycles (twLattice L χ⁻¹))) (x : X) : tensAmb L χ X (Submodule.Quotient.mk c ⊗ₜ[O] x) = Submodule.Quotient.mk (tensCoc L χ X c x)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NatCastMemPlaceAt

theorem **norm_int_le_one** (a : 𝒪 F) : ‖φ (a : F)‖ ≤ 1

theorem **norm_p_lt_one** : ‖(p : ℂ_[p])‖ < 1

theorem **mem_placeAt** [NumberField F] (a : 𝒪 F) : a ∈ (placeAt φ).asIdeal ↔ ‖φ (a : F)‖ < 1

theorem **natCast_mem_placeAt** [NumberField F] : ((p : ℕ) : 𝒪 F) ∈ (placeAt φ).asIdeal

theorem **mem_kurC** (a : F) : φ a ∈ kurC φ

theorem **kur_le_kurC** : Formalism.L2.Th1.CM.MaximalUnramifiedExtension p ≤ kurC φ

theorem **root_mem_kurC** {ζ : ℂ_[p]} {n : ℕ} (hn : ¬ p ∣ n) (h : ζ ^ n = 1) : ζ ∈ kurC φ

theorem **kurC_congr** {F' : Type} [Field F'] {φ' : F' →+* ℂ_[p]} (h : φ.fieldRange = φ'.fieldRange) : kurC φ = kurC φ'

theorem **toCp_algebraMap** (a : F) : toCp φ (algebraMap F (PAt φ) a) = φ a

theorem **valC_add** (z : FrPt Xo c (PAt φ)) (f g : Rul Xo c) : valC z (f + g) = valC z f + valC z g

theorem **valC_smul** (z : FrPt Xo c (PAt φ)) (a : c.F) (f : Rul Xo c) : valC z (a • f) = φ a * valC z f

theorem **valC_zero** (z : FrPt Xo c (PAt φ)) : valC z 0 = 0

theorem **IsReadPt.valC** {T : CDVR} {io : c.F →+* T.Fr} {k : T.Fr →+* ℂ_[p]} {h : IsReading φ T io k} {y : IntFr Xo c T io} {u : Ix Dw → (PAt φ)ˣ} {z : FrPt Xo c (PAt φ)} (hz : IsReadPt h y u z) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) : valC z f = toCp φ (wchar (ex Dw a n) u) * k (evI y f)

theorem **norm_wchar** {u : Ix Dw → (PAt φ)ˣ} (hu : ∀ a, ‖toCp φ (u a)‖ = 1) (e : Ix Dw → ℕ) : ‖toCp φ (wchar e u)‖ = 1

theorem **PtC.exists_valC** (z : PtC Xo c φ) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : ∃ (T : CDVR) (io : c.F →+* T.Fr) (k : T.Fr →+* ℂ_[p]) (_ : IsReading φ T io k) (y : IntFr Xo c T io) (u : ℂ_[p]), y.IsAlg ∧ ‖u‖ = 1 ∧ ∀ f ∈ sec Xo c a n, valC z.1 f = u * k (evI y f)

theorem **PtC.exists_of_reading** {T : CDVR} {io : c.F →+* T.Fr} {k : T.Fr →+* ℂ_[p]} (h : IsReading φ T io k) (y : IntFr Xo c T io) (hy : y.IsAlg) : ∃ z : PtC Xo c φ, ∀ a n, ∀ f ∈ sec Xo c a n, valC z.1 f = k (evI y f)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NearbyCyclesDegreeZeroEquiv

theorem **spec_str** {C D : Type} [CommRing C] [Algebra K C] [CommRing D] [Algebra K D] (f : C →ₐ[K] D) : Spec.map (CommRingCat.ofHom (f : C →+* D)) ≫ str K C = str K D

theorem **geomMap_spec_id** (C : Type) [CommRing C] [Algebra K C] : GlobalTheory.Et.geomMap Ω (str K C) (Spec.map (CommRingCat.ofHom ((AlgHom.id K C : C →ₐ[K] C) : C →+* C))) (spec_str K (AlgHom.id K C)) = 𝟙 _

theorem **mapG_id** (i : ℕ) (C : Type) [CommRing C] [Algebra K C] : mapG Λ K Ω i (AlgHom.id K C) = LinearMap.id

theorem **mapG_comp** (i : ℕ) {C D E : Type} [CommRing C] [Algebra K C] [CommRing D] [Algebra K D] [CommRing E] [Algebra K E] (g : D →ₐ[K] E) (f : C →ₐ[K] D) : mapG Λ K Ω i (g.comp f) = (mapG Λ K Ω i g).comp (mapG Λ K Ω i f)

theorem **nonempty_geom** (C : Type) [CommRing C] [Algebra K C] (ψ : C →ₐ[K] Ω) : Nonempty (GlobalTheory.Et.geom Ω (str K C))

theorem **nontrivial_HG_zero** [Nontrivial Λ] (C : Type) [CommRing C] [Algebra K C] (ψ : C →ₐ[K] Ω) : Nontrivial (HG Λ K Ω 0 C)

theorem **map_id** (i : ℕ) (A : Type) [CommRing A] [Algebra O A] : map Λ K Ω O i (AlgHom.id O A) = LinearMap.id

theorem **map_comp** (i : ℕ) {A B D : Type} [CommRing A] [Algebra O A] [CommRing B] [Algebra O B] [CommRing D] [Algebra O D] (g : B →ₐ[O] D) (f : A →ₐ[O] B) : map Λ K Ω O i (g.comp f) = (map Λ K Ω O i g).comp (map Λ K Ω O i f)

theorem **coe_aut_one** (A : Type) [CommRing A] [Algebra O A] : ((1 : A ≃ₐ[O] A) : A →ₐ[O] A) = AlgHom.id O A

theorem **coe_aut_mul** {A : Type} [CommRing A] [Algebra O A] (e e' : A ≃ₐ[O] A) : ((e * e' : A ≃ₐ[O] A) : A →ₐ[O] A) = (e : A →ₐ[O] A).comp (e' : A →ₐ[O] A)

theorem **nontrivial_H_zero_self** [Nontrivial Λ] [IsAlgClosed Ω] : Nontrivial (H Λ K Ω O 0 O)

theorem **subsingleton_geom_self** : Subsingleton (GlobalTheory.Et.geom Ω (str K (K ⊗[O] O)))

theorem **nonempty_equiv_self** : Nonempty (H Λ K Ω O 0 O ≃ₗ[Λ] Λ)

theorem **gal_self** (σ : Ω ≃ₐ[K] Ω) : gal Λ K Ω O 0 O σ = 1

theorem **smapG_id** (i : ℕ) (C : Type) [CommRing C] [Algebra K C] (hτ : (RingHom.id Ω).comp (algebraMap K Ω) = (algebraMap K Ω).comp (RingHom.id K)) (hf : (RingHom.id C).comp (algebraMap K C) = (algebraMap K C).comp (RingHom.id K)) : smapG Λ i (RingHom.id K) (RingHom.id Ω) hτ (RingHom.id C) hf = LinearMap.id

theorem **smapG_congr** (i : ℕ) {C D : Type} [CommRing C] [Algebra K C] [CommRing D] [Algebra K' D] {κ : K →+* K'} {τ : Ω →+* Ω'} {hτ : τ.comp (algebraMap K Ω) = (algebraMap K' Ω').comp κ} {f f' : C →+* D} (hf : f.comp (algebraMap K C) = (algebraMap K' D).comp κ) (h : f = f') : smapG Λ i κ τ hτ f hf = smapG Λ i κ τ hτ f' (h ▸ hf)

theorem **smapG_comp** (i : ℕ) {K'' Ω'' : Type} [Field K''] [Field Ω''] [Algebra K'' Ω''] {C D E : Type} [CommRing C] [Algebra K C] [CommRing D] [Algebra K' D] [CommRing E] [Algebra K'' E] (κ : K →+* K') (τ : Ω →+* Ω') (hτ : τ.comp (algebraMap K Ω) = (algebraMap K' Ω').comp κ) (f : C →+* D) (hf : f.comp (algebraMap K C) = (algebraMap K' D).comp κ) (κ' : K' →+* K'') (τ' : Ω' →+* Ω'') (hτ' : τ'.comp (algebraMap K' Ω') = (algebraMap K'' Ω'').comp κ') (g : D →+* E) (hg : g.comp (algebraMap K' D) = (algebraMap K'' E).comp κ') : smapG Λ i (κ'.comp κ) (τ'.comp τ) (by rw [RingHom.comp_assoc, hτ, ← RingHom.comp_assoc, hτ', RingHom.comp_assoc]) (g.comp f) (by rw [RingHom.comp_assoc, hf, ← RingHom.comp_assoc, hg, RingHom.comp_assoc]) = (smapG Λ i κ' τ' hτ' g hg).comp (smapG Λ i κ τ hτ f hf)

theorem **tensor_ringHom_ext** {A B : Type} [CommRing A] [Algebra O A] [Ring B] {f g : K ⊗[O] A →+* B} (h : ∀ k a, f (k ⊗ₜ a) = g (k ⊗ₜ a)) : f = g

theorem **tmap_tmul** {A A' : Type} [CommRing A] [Algebra O A] [CommRing A'] [Algebra O' A'] (o : O →+* O') (κ : K →+* K') (hκ : κ.comp (algebraMap O K) = (algebraMap O' K').comp o) (φ : A →+* A') (hφ : φ.comp (algebraMap O A) = (algebraMap O' A').comp o) (k : K) (a : A) : tmap o κ hκ φ hφ (k ⊗ₜ a) = κ k ⊗ₜ φ a

theorem **tmap_algebraMap** {A A' : Type} [CommRing A] [Algebra O A] [CommRing A'] [Algebra O' A'] (o : O →+* O') (κ : K →+* K') (hκ : κ.comp (algebraMap O K) = (algebraMap O' K').comp o) (φ : A →+* A') (hφ : φ.comp (algebraMap O A) = (algebraMap O' A').comp o) : (tmap o κ hκ φ hφ).comp (algebraMap K (K ⊗[O] A)) = (algebraMap K' (K' ⊗[O'] A')).comp κ

theorem **tmap_id** {A : Type} [CommRing A] [Algebra O A] (hκ : (RingHom.id K).comp (algebraMap O K) = (algebraMap O K).comp (RingHom.id O)) (hφ : (RingHom.id A).comp (algebraMap O A) = (algebraMap O A).comp (RingHom.id O)) : tmap (RingHom.id O) (RingHom.id K) hκ (RingHom.id A) hφ = RingHom.id _

theorem **tmap_comp** {K'' : Type} [Field K''] {O'' : Type} [CommRing O''] [Algebra O'' K''] {A A' A'' : Type} [CommRing A] [Algebra O A] [CommRing A'] [Algebra O' A'] [CommRing A''] [Algebra O'' A''] (o : O →+* O') (κ : K →+* K') (hκ : κ.comp (algebraMap O K) = (algebraMap O' K').comp o) (φ : A →+* A') (hφ : φ.comp (algebraMap O A) = (algebraMap O' A').comp o) (o' : O' →+* O'') (κ' : K' →+* K'') (hκ' : κ'.comp (algebraMap O' K') = (algebraMap O'' K'').comp o') (φ' : A' →+* A'') (hφ' : φ'.comp (algebraMap O' A') = (algebraMap O'' A'').comp o') : tmap (o'.comp o) (κ'.comp κ) (by rw [RingHom.comp_assoc, hκ, ← RingHom.comp_assoc, hκ', RingHom.comp_assoc]) (φ'.comp φ) (by rw [RingHom.comp_assoc, hφ, ← RingHom.comp_assoc, hφ', RingHom.comp_assoc]) = (tmap o' κ' hκ' φ' hφ').comp (tmap o κ hκ φ hφ)

theorem **smap_congr** (i : ℕ) {A A' : Type} [CommRing A] [Algebra O A] [CommRing A'] [Algebra O' A'] {o₁ o₂ : O →+* O'} {κ₁ κ₂ : K →+* K'} {τ₁ τ₂ : Ω →+* Ω'} {φ₁ φ₂ : A →+* A'} {hκ₁ : κ₁.comp (algebraMap O K) = (algebraMap O' K').comp o₁} {hτ₁ : τ₁.comp (algebraMap K Ω) = (algebraMap K' Ω').comp κ₁} {hφ₁ : φ₁.comp (algebraMap O A) = (algebraMap O' A').comp o₁} {hκ₂ : κ₂.comp (algebraMap O K) = (algebraMap O' K').comp o₂} {hτ₂ : τ₂.comp (algebraMap K Ω) = (algebraMap K' Ω').comp κ₂} {hφ₂ : φ₂.comp (algebraMap O A) = (algebraMap O' A').comp o₂} (e₁ : o₁ = o₂) (e₂ : κ₁ = κ₂) (e₃ : τ₁ = τ₂) (e₄ : φ₁ = φ₂) : smap Λ i o₁ κ₁ hκ₁ τ₁ hτ₁ φ₁ hφ₁ = smap Λ i o₂ κ₂ hκ₂ τ₂ hτ₂ φ₂ hφ₂

theorem **smap_id** (i : ℕ) (A : Type) [CommRing A] [Algebra O A] (hκ : (RingHom.id K).comp (algebraMap O K) = (algebraMap O K).comp (RingHom.id O)) (hτ : (RingHom.id Ω).comp (algebraMap K Ω) = (algebraMap K Ω).comp (RingHom.id K)) (hφ : (RingHom.id A).comp (algebraMap O A) = (algebraMap O A).comp (RingHom.id O)) : smap Λ i (RingHom.id O) (RingHom.id K) hκ (RingHom.id Ω) hτ (RingHom.id A) hφ = LinearMap.id

theorem **smap_comp** (i : ℕ) {K'' Ω'' : Type} [Field K''] [Field Ω''] [Algebra K'' Ω''] {O'' : Type} [CommRing O''] [Algebra O'' K''] {A A' A'' : Type} [CommRing A] [Algebra O A] [CommRing A'] [Algebra O' A'] [CommRing A''] [Algebra O'' A''] (o : O →+* O') (κ : K →+* K') (hκ : κ.comp (algebraMap O K) = (algebraMap O' K').comp o) (τ : Ω →+* Ω') (hτ : τ.comp (algebraMap K Ω) = (algebraMap K' Ω').comp κ) (φ : A →+* A') (hφ : φ.comp (algebraMap O A) = (algebraMap O' A').comp o) (o' : O' →+* O'') (κ' : K' →+* K'') (hκ' : κ'.comp (algebraMap O' K') = (algebraMap O'' K'').comp o') (τ' : Ω' →+* Ω'') (hτ' : τ'.comp (algebraMap K' Ω') = (algebraMap K'' Ω'').comp κ') (φ' : A' →+* A'') (hφ' : φ'.comp (algebraMap O' A') = (algebraMap O'' A'').comp o') : smap Λ i (o'.comp o) (κ'.comp κ) (by rw [RingHom.comp_assoc, hκ, ← RingHom.comp_assoc, hκ', RingHom.comp_assoc]) (τ'.comp τ) (by rw [RingHom.comp_assoc,

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NegOneNotMemNatLat

theorem **secRow_eLast** {R : Type*} [CommRing R] : secRow (eLast R a) = 1

theorem **secRow_eq_updateRow** {R : Type*} [CommRing R] {n : ℕ} (y : Fin (n + 1) → R) : secRow y = (1 : Matrix (Fin (n + 1)) (Fin (n + 1)) R).updateRow (Fin.last n) y

theorem **det_secRow** {R : Type*} [CommRing R] {n : ℕ} (y : Fin (n + 1) → R) : (secRow y).det = y (Fin.last n)

theorem **eLast_vecMul_secRow** {R : Type*} [CommRing R] {n : ℕ} (y : Fin (n + 1) → R) : Matrix.vecMul (eLast R (n + 1)) (secRow y) = y

theorem **secRow_mul_secInv** {n : ℕ} (y : Fin (n + 1) → Cinf E) (hy : ∀ w, y (Fin.last n) w ≠ 0) : secRow y * (secInv y : Matrix (Fin (n + 1)) (Fin (n + 1)) (Cinf E)) = 1

theorem **vecMul_secInv** {n : ℕ} (y : Fin (n + 1) → Cinf E) (hy : ∀ w, y (Fin.last n) w ≠ 0) : Matrix.vecMul y (secInv y : Matrix (Fin (n + 1)) (Fin (n + 1)) (Cinf E)) = eLast (Cinf E) (n + 1)

theorem **secInv_eLast** : secInv (eLast (Cinf E) a) = 1

theorem **rowAct_ratGL_mul** (ξ : Fin a → E) (γ : GL (Fin a) E) (g : GA E a) : rowAct ξ (ratGL E a γ * g) = rowAct (Matrix.vecMul ξ (γ : Matrix (Fin a) (Fin a) E)) g

theorem **detPow_mul** (x : ℝ) (g h : GA E a) : detPow E a x (g * h) = detPow E a x g * detPow E a x h

theorem **detPow_pos** (x : ℝ) (g : GA E a) : 0 < detPow E a x g

theorem **detPow_ratGL** (x : ℝ) (γ : GL (Fin a) E) : detPow E a x (ratGL E a γ) = 1

theorem **norm_absPow_val** (x : ℝ) (v : HeightOneSpectrum (𝒪 E)) : (((GlobalTheory.HeckeGal.Model.absPow E x).val v : ℂˣ) : ℂ) = (Ideal.absNorm v.asIdeal : ℂ) ^ (-(x : ℂ))

theorem **thetaSum_ratGL_mul** (φ φ' : GA E a → ℂ) (Φ : (Fin a → Cinf E) → ℂ) (N : ℕ) (hφ : ∀ γ g, φ (ratGL E a γ * g) = φ g) (hφ' : ∀ γ g, φ' (ratGL E a γ * g) = φ' g) (γ : GL (Fin a) E) (g : GA E a) : thetaSum φ φ' Φ N (ratGL E a γ * g) = thetaSum φ φ' Φ N g

theorem **indN_eLast** (N : ℕ) : indN N (eLast (Af E) a) = 1

theorem **neg_one_notMem_natLat** (N : ℕ) (v : HeightOneSpectrum (𝒪 E)) (hv : (N : 𝒪 E) ∈ v.asIdeal) : (-1 : Af E) ∉ natLat E N

theorem **indN_zero** {n : ℕ} (N : ℕ) (v : HeightOneSpectrum (𝒪 E)) (hv : (N : 𝒪 E) ∈ v.asIdeal) : indN N (0 : Fin (n + 1) → Af E) = 0

theorem **pInf_rowAct_one** (ξ : Fin a → E) (i : Fin a) (w : InfinitePlace E) : pInf E (rowAct ξ 1 i) w = w.embedding (ξ i)

theorem **thetaSum_point** {n : ℕ} (φ φ' : GA E (n + 1) → ℂ) (N : ℕ) : thetaSum φ φ' (fun y => if y = eLast (Cinf E) (n + 1) then 1 else 0) N 1 = φ 1 * starRingEnd ℂ (φ' 1)

theorem **zint_dirac** {n : ℕ} (φ φ' : GA E (n + 1) → ℂ) (N : ℕ) (x : ℝ) : Zint (Measure.dirac 1) Set.univ φ φ' (fun y => if y = eLast (Cinf E) (n + 1) then 1 else 0) N x = φ 1 * starRingEnd ℂ (φ' 1)

theorem **pFin_eq** : GlobalTheory.RSJ.pFin E = GlobalTheory.CuspRep.pFin E

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NonemptyEquivMultiProdOne

private theorem **twKer_le_comap_intertwining** {G : Type} [Group G] {X Y : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup Y] [Module ℂ Y] {ρ : Representation ℂ G X} {τ : Representation ℂ G Y} (f : ρ.IntertwiningMap τ) (N : Subgroup G) : twKer ρ N 1 ≤ (twKer τ N 1).comap f.toLinearMap

theorem **isCuspidal_of_equiv** {π : Representation ℂ (GL n k) V} {σ : Representation ℂ (GL n k) W} (φ : π.Equiv σ) (h : IsCuspidal π) : IsCuspidal σ

theorem **isCuspidal_twist** {π : Representation ℂ (GL n k) V} (h : IsCuspidal π) {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) : IsCuspidal (twist π (detChar n χ))

theorem **isSmoothChar_absPow** (s : ℂ) : IsSmoothChar (absPow k s)

theorem **absPow_ofReal_eq_one** {s : ℝ} : absPow k s = 1 ↔ s = 0

private theorem **ofChar_cl_congr** (e : (Σ _ : Fin 1, n) ≃ n) (g : GL n k) : g ∈ parab k e

private theorem **leviProj_fin_one** (e : (Σ _ : Fin 1, n) ≃ n) (he : ∀ a i, e (a, i) = i) (p : parab k e) (a : Fin 1) : leviProj k e p a = (p : GL n k)

private theorem **deltaHalf_fin_one_eq_one** (g : Levi k fun _ : Fin 1 => n) : deltaHalf k (fun _ : Fin 1 => n) g = 1

theorem **nonempty_equiv_multiProd_one** {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) (e : (Σ _ : Fin 1, n) ≃ n) (he : ∀ a i, e (a, i) = i) : Nonempty (π.Equiv (multiProd e fun _ : Fin 1 => π))

theorem **of_eq_of** {N : ℕ} {π : Representation ℂ (GL (Fin N) k) V} {hs : IsSmooth π} {hi : π.IsIrreducible} {σ : Representation ℂ (GL (Fin N) k) W} {hs' : IsSmooth σ} {hi' : σ.IsIrreducible} : of π hs hi = of σ hs' hi' ↔ Nonempty (π.Equiv σ)

@[simp] theorem **of_rep** (x : IrrGL k) : of x.rep x.2.rep.smooth x.2.rep.irr = x

@[simp] theorem **tw_fst** (x : IrrGL k) (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (x.tw χ hχ).1 = x.1

@[simp] theorem **nu_fst** (x : IrrGL k) (s : ℂ) : (x.nu s).1 = x.1

theorem **tw_of** {N : ℕ} (π : Representation ℂ (GL (Fin N) k) V) (hs : IsSmooth π) (hi : π.IsIrreducible) (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (of π hs hi).tw χ hχ = of (twist π (detChar _ χ)) (isSmooth_twist hs hχ.detChar) (isIrreducible_twist hi _)

private theorem **clTwist_congr** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] (c : IrrCl G) {χ ψ : G →* ℂˣ} (h : χ = ψ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : clTwist c χ hχ = clTwist c ψ hψ

private theorem **detChar_mul'** (m : Type) [Fintype m] [DecidableEq m] (χ ψ : kˣ →* ℂˣ) : detChar m (χ * ψ) = detChar m χ * detChar m ψ

theorem **tw_tw** (x : IrrGL k) (χ ψ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : (x.tw χ hχ).tw ψ hψ = x.tw (χ * ψ) (hχ.mul hψ)

theorem **tw_one** (x : IrrGL k) (h : IsSmoothChar (1 : kˣ →* ℂˣ)) : x.tw 1 h = x

private theorem **tw_congr'** (x : IrrGL k) {χ ψ : kˣ →* ℂˣ} (h : χ = ψ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : x.tw χ hχ = x.tw ψ hψ

@[simp] theorem **nu_nu** (x : IrrGL k) (s t : ℂ) : (x.nu s).nu t = x.nu (s + t)

@[simp] theorem **nu_zero** (x : IrrGL k) : x.nu 0 = x

private theorem **mem_parab_fin_one** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] {χ ψ : G →* ℂˣ} (h : χ = ψ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : (IrrRep.ofChar χ hχ).cl = (IrrRep.ofChar ψ hψ).cl

theorem **tw_ofChar** (χ ψ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : (ofChar χ hχ).tw ψ hψ = ofChar (χ * ψ) (hχ.mul hψ)

private theorem **det_scalar_fin_one'** (x : kˣ) : GeneralLinearGroup.det (GeneralLinearGroup.scalar (Fin 1) x) = x

theorem **ofChar_inj** {χ ψ : kˣ →* ℂˣ} {hχ : IsSmoothChar χ} {hψ : IsSmoothChar ψ} : ofChar χ hχ = ofChar ψ hψ ↔ χ = ψ

theorem **nu_ofChar_inj** (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) {s t : ℝ} : (ofChar χ hχ).nu s = (ofChar χ hχ).nu t ↔ s = t

theorem **isCusp_of** {N : ℕ} {π : Representation ℂ (GL (Fin N) k) V} {hs : IsSmooth π} {hi : π.IsIrreducible} : (of π hs hi).IsCusp ↔ 0 < N ∧ IsCuspidal π

theorem **IsCusp.tw** {x : IrrGL k} (h : x.IsCusp) (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (x.tw χ hχ).IsCusp

theorem **IsCusp.nu** {x : IrrGL k} (h : x.IsCusp) (s : ℂ) : (x.nu s).IsCusp

theorem **isCusp_ofChar** (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (ofChar χ hχ).IsCusp

theorem **inert_equivalence** : Equivalence (Inert (k

theorem **inert_nu** (x : IrrGL k) (s : ℂ) : Inert x (x.nu s)

theorem **inert_of** {N : ℕ} {π : Representation ℂ (GL (Fin N) k) V} {hs : IsSmooth π} {hi : π.IsIrreducible} {σ : Representation ℂ (GL (Fin N) k) W} {hs' : IsSmooth σ} {hi' : σ.IsIrreducible} : Inert (of π hs hi) (of σ hs' hi') ↔ InertEquiv π σ

theorem **isSmooth_prodIrr** {h N : ℕ} (x : Fin h → IrrGL k) (hd : ∑ a, (x a).1 = N) : IsSmooth (prodIrr x hd)

theorem **nonempty_equiv_prodIrr_one** (x : IrrGL k) (hd : ∑ _ : Fin 1, x.1 = x.1) : Nonempty (x.rep.Equiv (prodIrr (fun _ : Fin 1 => x) hd))

theorem **HasSupport.isIrreducible** (h : HasSupport π s) : π.IsIrreducible

theorem **HasSupport.isSmooth** (h : HasSupport π s) : IsSmooth π

theorem **HasSupport.isCusp** (h : HasSupport π s) {y : IrrGL k} (hy : y ∈ s) : y.IsCusp

theorem **HasSupport.sum_fst** (h : HasSupport π s) : (s.map Sigma.fst).sum = Fintype.card n

theorem **HasSupport.of_equiv** (h : HasSupport π s) {σ : Representation ℂ (GL n k) W} (φ : π.Equiv σ) : HasSupport σ s

theorem **hasSupport_of_isCuspSupport** {h : ℕ} {d : Fin h → ℕ} {X : Fin h → Type} [∀ a, AddCommGroup (X a)] [∀ a, Module ℂ (X a)] {σ : ∀ a, Representation ℂ (GL (Fin (d a)) k) (X a)} (H : IsCuspSupport π d σ) : HasSupport π (Finset.univ.val.map fun a => IrrGL.of (σ a) (H.2.1 a).2.2.1 (H.2.1 a).2.1)

theorem **hasSupport_of_isSubquotient** {h N : ℕ} {π : Representation ℂ (GL (Fin N) k) V} (hi : π.IsIrreducible) {x : Fin h → IrrGL k} (hc : ∀ a, (x a).IsCusp) {hd : ∑ a, (x a).1 = N} (hs : IsSubquotient π (prodIrr x hd)) : HasSupport π (Finset.univ.val.map x)

theorem **hasSupport_self** {x : IrrGL k} (hx : x.IsCusp) : HasSupport x.rep {x}

theorem **isCusp_first** (Δ : Segment k) : Δ.first.IsCusp

@[simp] theorem **first_ofFirst** (ρ : IrrGL k) (hρ : ρ.IsCusp) (l : ℕ) (hl : 0 < l) : (ofFirst ρ hρ l hl).first = ρ

@[simp] theorem **len_ofFirst** (ρ : IrrGL k) (hρ : ρ.IsCusp) (l : ℕ) (hl : 0 < l) : (ofFirst ρ hρ l hl).len = l

theorem **ext_first** {Δ Δ' : Segment k} (h : Δ.first = Δ'.first) (h' : Δ.len = Δ'.len) : Δ = Δ'

theorem **isCusp_term** (Δ : Segment k) (a : Fin Δ.len) : (Δ.term a).IsCusp

theorem **sum_term_fst** (Δ : Segment k) : ∑ a, (Δ.term a).1 = Δ.deg

theorem **mem_carrier** {Δ : Segment k} {x : IrrGL k} : x ∈ Δ.carrier ↔ ∃ a : ℕ, a < Δ.len ∧ Δ.first.nu a = x

theorem **mem_ms** {Δ : Segment k} {x : IrrGL k} : x ∈ Δ.ms ↔ x ∈ Δ.carrier

@[simp] theorem **card_ms** (Δ : Segment k) : Δ.ms.card = Δ.len

theorem **mem_icc** {ρ : IrrGL k} {hρ : ρ.IsCusp} {c d : ℤ} {h : c ≤ d} {x : IrrGL k} : x ∈ (icc ρ hρ c d h).carrier ↔ ∃ a : ℤ, c ≤ a ∧ a ≤ d ∧ ρ.nu a = x

@[simp] theorem **carrier_single** (ρ : IrrGL k) (hρ : ρ.IsCusp) : (single ρ hρ).carrier = {ρ}

@[simp] theorem **ms_single** (ρ : IrrGL k) (hρ : ρ.IsCusp) : (single ρ hρ).ms = {ρ}

theorem **Linked.symm** {Δ Δ' : Segment k} (h : Linked Δ Δ') : Linked Δ' Δ

theorem **not_linked_self** (Δ : Segment k) : ¬ Linked Δ Δ

theorem **inert_of_mem** {Δ : Segment k} {x : IrrGL k} (h : x ∈ Δ.carrier) : IrrGL.Inert Δ.first x

private theorem **first_mem_carrier'** (Δ : Segment k) : Δ.first ∈ Δ.carrier

theorem **Linked.inert** {Δ Δ' : Segment k} (h : Linked Δ Δ') : IrrGL.Inert Δ.first Δ'.first

theorem **unipB_mul_apply** (u v : unipB k (Fin N)) {i j : Fin N} (hij : (j : ℕ) = i + 1) : ((u * v : unipB k (Fin N)) : GL (Fin N) k).val i j = (u : GL (Fin N) k).val i j + (v : GL (Fin N) k).val i j

@[simp] theorem **msupp_add** (m n : Multiset (Segment k)) : msupp (m + n) = msupp m + msupp n

@[simp] theorem **msupp_singleton** (Δ : Segment k) : msupp {Δ} = Δ.ms

theorem **IsZSeg.of_equiv** (h : IsZSeg Δ π) (φ : π.Equiv σ) : IsZSeg Δ σ

theorem **IsLSeg.of_equiv** (h : IsLSeg Δ π) (φ : π.Equiv σ) : IsLSeg Δ σ

theorem **IsZ.of_equiv** (h : IsZ m π) (φ : π.Equiv σ) : IsZ m σ

theorem **IsL.of_equiv** (h : IsL m π) (φ : π.Equiv σ) : IsL m σ

theorem **IsZSeg.unique** (h : IsZSeg Δ π) (h' : IsZSeg Δ σ) : Nonempty (σ.Equiv π)

theorem **IsLSeg.unique** (h : IsLSeg Δ π) (h' : IsLSeg Δ σ) : Nonempty (σ.Equiv π)

theorem **IsZSeg.hasSupport** (h : IsZSeg Δ π) : HasSupport π Δ.ms

theorem **IsLSeg.hasSupport** (h : IsLSeg Δ π) : HasSupport π Δ.ms

theorem **IsZSeg.deg_eq** (h : IsZSeg Δ π) : Δ.deg = N

theorem **IsLSeg.deg_eq** (h : IsLSeg Δ π) : Δ.deg = N

theorem **IsZ.irr_smooth** (h : IsZ m π) : π.IsIrreducible ∧ IsSmooth π

theorem **IsL.irr_smooth** (h : IsL m π) : π.IsIrreducible ∧ IsSmooth π

theorem **isOrderedForm_one** (Δ : Segment k) : IsOrderedForm (fun _ : Fin 1 => Δ) {Δ}

theorem **IsZSeg.isZ** (h : IsZSeg Δ π) : IsZ {Δ} π

theorem **IsLSeg.isL** (h : IsLSeg Δ π) : IsL {Δ} π

theorem **IsZSeg.cl_eq** {Z : ∀ Δ : Segment k, IrrCl (GL (Fin (Δ.len * Δ.d)) k)} (hZ : IsSegmentClass k Z) {Δ : Segment k} {π : IrrRep (GL (Fin (Δ.len * Δ.d)) k)} (h : IsZSeg Δ π.ρ) : π.cl = Z Δ

private theorem **nonempty_equiv_prodIrr_fin_one** (x : Fin 1 → IrrGL k) (y : IrrGL k) (hx : x 0 = y) (hd : ∑ a, (x a).1 = y.1) : Nonempty (y.rep.Equiv (prodIrr x hd))

theorem **single_spec** {ρ : IrrGL k} (hρ : ρ.IsCusp) : IsZSeg (.single ρ hρ) ρ.rep ∧ IsLSeg (.single ρ hρ) ρ.rep ∧ IsZ {.single ρ hρ} ρ.rep ∧ IsL {.single ρ hρ} ρ.rep

private theorem **rep_apply_eq_det_smul_of_eq_ofChar** {ψ : kˣ →* ℂˣ} (hψ : IsSmoothChar ψ) (y : IrrGL k) (hy : y = IrrGL.ofChar ψ hψ) (g : GL (Fin y.1) k) (w : y.2.rep.V) : y.rep g w = (ψ (GeneralLinearGroup.det g) : ℂ) • w

theorem **embeds_detRep_segment** {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) {l : ℕ} (hl : 0 < l) (hd : ∑ a, ((Segment.ofChar χ hχ l hl).term a).1 = N) : Embeds (detRep (Fin N) (χ * absPow k (((l : ℂ) - 1) / 2))) (prodIrr (Segment.ofChar χ hχ l hl).term hd)

theorem **IsZSeg.equiv_detRep** {χ : kˣ →* ℂˣ} {hχ : IsSmoothChar χ} {l : ℕ} {hl : 0 < l} {π : Representation ℂ (GL (Fin N) k) V} (h : IsZSeg (Segment.ofChar χ hχ l hl) π) : Nonempty ((detRep (Fin N) (χ * absPow k (((l : ℂ) - 1) / 2))).Equiv π)

theorem **hasSupport_detRep** {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) {l : ℕ} (hl : 0 < l) : HasSupport (detRep (Fin l) (χ * absPow k (((l : ℂ) - 1) / 2))) (Segment.ofChar χ hχ l hl).ms

theorem **single_precedes** (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (Segment.single _ (IrrGL.isCusp_ofChar χ hχ)).Precedes (.single _ ((IrrGL.isCusp_ofChar χ hχ).nu 1)) ∧ ¬ (Segment.single _ ((IrrGL.isCusp_ofChar χ hχ).nu 1)).Precedes (.single _ (IrrGL.isCusp_ofChar χ hχ))

theorem **isSpeh_one** (x : IrrGL k) : IsSpeh x 1 x.rep

theorem **isGenericFor_one** (ψ : AddChar k Circle) (χ : kˣ →* ℂˣ) : IsGenericFor ψ (detRep (Fin 1) χ)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NonemptyEquivOfRangeEq

@[simp] theorem **descend_apply** (ρ : Representation ℂ G V) (q : G →* M) (hq : Function.Surjective q) (h : ∀ g, q g = 1 → ρ g = 1) (g : G) : descend ρ q hq h (q g) = ρ g

theorem **mem_subrepIso** {ρ : Representation ℂ G V} {σ : Representation ℂ G W} (e : ρ.Equiv σ) (S : Subrepresentation ρ) (w : W) : w ∈ subrepIso e S ↔ e.symm w ∈ S

theorem **isIrreducible_twist** {ρ : Representation ℂ G V} (h : ρ.IsIrreducible) (χ : G →* ℂˣ) : (twist ρ χ).IsIrreducible

theorem **IsSocle.embeds** {σ : Representation ℂ G W} {π : Representation ℂ G V} (h : IsSocle σ π) : Embeds σ π

theorem **IsSocle.isSubquotient** {σ : Representation ℂ G W} {π : Representation ℂ G V} (h : IsSocle σ π) : IsSubquotient σ π

theorem **IsCosocle.isSubquotient** {σ : Representation ℂ G W} {π : Representation ℂ G V} (h : IsCosocle σ π) : IsSubquotient σ π

theorem **IsSocle.of_equiv** {σ : Representation ℂ G W} {π : Representation ℂ G V} {Y : Type} [AddCommGroup Y] [Module ℂ Y] {σ' : Representation ℂ G X} {π' : Representation ℂ G Y} (h : IsSocle σ π) (a : σ.Equiv σ') (b : π.Equiv π') : IsSocle σ' π'

theorem **IsCosocle.of_equiv** {σ : Representation ℂ G W} {π : Representation ℂ G V} {Y : Type} [AddCommGroup Y] [Module ℂ Y] {σ' : Representation ℂ G X} {π' : Representation ℂ G Y} (h : IsCosocle σ π) (a : σ.Equiv σ') (b : π.Equiv π') : IsCosocle σ' π'

theorem **isSocle_self** {π : Representation ℂ G V} (h : π.IsIrreducible) : IsSocle π π

theorem **isCosocle_self** {π : Representation ℂ G V} (h : π.IsIrreducible) : IsCosocle π π

theorem **isAtom_range** {σ : Representation ℂ G W} {π : Representation ℂ G V} (h : σ.IsIrreducible) (f : σ.IntertwiningMap π) (hf : Function.Injective f) : IsAtom f.range

theorem **isCoatom_ker** {σ : Representation ℂ G W} {π : Representation ℂ G V} (h : σ.IsIrreducible) (f : π.IntertwiningMap σ) (hf : Function.Surjective f) : IsCoatom f.ker

theorem **nonempty_equiv_of_range_eq** {σ : Representation ℂ G W} {τ : Representation ℂ G X} {π : Representation ℂ G V} (f : σ.IntertwiningMap π) (hf : Function.Injective f) (g : τ.IntertwiningMap π) (hg : Function.Injective g) (h : f.range = g.range) : Nonempty (σ.Equiv τ)

theorem **nonempty_equiv_of_ker_eq** {σ : Representation ℂ G W} {τ : Representation ℂ G X} {π : Representation ℂ G V} (f : π.IntertwiningMap σ) (hf : Function.Surjective f) (g : π.IntertwiningMap τ) (hg : Function.Surjective g) (h : f.ker = g.ker) : Nonempty (σ.Equiv τ)

theorem **IsSocle.equiv_of_embeds** {σ : Representation ℂ G W} {τ : Representation ℂ G X} {π : Representation ℂ G V} (h : IsSocle σ π) (hτ : τ.IsIrreducible) (he : Embeds τ π) : Nonempty (τ.Equiv σ)

theorem **IsCosocle.equiv_of_surjective** {σ : Representation ℂ G W} {τ : Representation ℂ G X} {π : Representation ℂ G V} (h : IsCosocle σ π) (hτ : τ.IsIrreducible) (g : π.IntertwiningMap τ) (hg : Function.Surjective g) : Nonempty (τ.Equiv σ)

theorem **IsSocle.unique** {σ : Representation ℂ G W} {τ : Representation ℂ G X} {π : Representation ℂ G V} (h : IsSocle σ π) (h' : IsSocle τ π) : Nonempty (τ.Equiv σ)

theorem **IsCosocle.unique** {σ : Representation ℂ G W} {τ : Representation ℂ G X} {π : Representation ℂ G V} (h : IsCosocle σ π) (h' : IsCosocle τ π) : Nonempty (τ.Equiv σ)

theorem **apply_mem_isotypic** {σ : Representation ℂ G V} {τ : Representation ℂ G W} (f : τ.IntertwiningMap σ) (w : W) : f w ∈ isotypic σ τ

theorem **isotypic_ne_bot_iff** {σ : Representation ℂ G V} {τ : Representation ℂ G W} : isotypic σ τ ≠ ⊥ ↔ ∃ f : τ.IntertwiningMap σ, f ≠ 0

theorem **isotypic_ne_bot_iff_embeds** {σ : Representation ℂ G V} {τ : Representation ℂ G W} (h : τ.IsIrreducible) : isotypic σ τ ≠ ⊥ ↔ Embeds τ σ

theorem **map_mem_isotypic** {σ : Representation ℂ G V} {σ' : Representation ℂ G X} (φ : σ.IntertwiningMap σ') (τ : Representation ℂ G W) {v : V} (hv : v ∈ isotypic σ τ) : φ v ∈ isotypic σ' τ

@[simp] theorem **isotypicMap_apply** {σ : Representation ℂ G V} {σ' : Representation ℂ G X} (φ : σ.IntertwiningMap σ') (τ : Representation ℂ G W) (v : (isotypic σ τ).toSubmodule) : (isotypicMap φ τ v : X) = φ v

theorem **out_apply** (π : Representation ℂ G V) {H : Subgroup G} {U : Subgroup H} {v : V} (hv : ∀ u ∈ U, π (u : H) v = v) (h : H) : π ((h : H ⧸ U).out : H) v = π h v

theorem **exists_normal_fix** {π : Representation ℂ G V} (hπ : IsSmooth π) {H : Subgroup G} (hH : IsCompact (H : Set G)) (v : V) : ∃ U : Subgroup H, U.Normal ∧ U.FiniteIndex ∧ ∀ u ∈ U, π (u : H) v = v

theorem **isCompl_fixed_twKer** {π : Representation ℂ G V} (hπ : IsSmooth π) {H : Subgroup G} (hH : IsCompact (H : Set G)) : IsCompl (fixed π H) (twKer π H 1)

theorem **idem_mem** (v : V) : idem hπ hH v ∈ fixed π H

theorem **idem_of_mem** {v : V} (hv : v ∈ fixed π H) : idem hπ hH v = v

theorem **idem_eq_zero_iff** {v : V} : idem hπ hH v = 0 ↔ v ∈ twKer π H 1

theorem **idem_apply_mem** {h : G} (hh : h ∈ H) (v : V) : idem hπ hH (π h v) = idem hπ hH v

theorem **idem_eq_cosetSum** (U : Subgroup H) [Fintype (H ⧸ U)] {v : V} (hv : ∀ u ∈ U, π (u : H) v = v) : idem hπ hH v = (Fintype.card (H ⧸ U) : ℂ)⁻¹ • ∑ q : H ⧸ U, π (q.out : H) v

theorem **orbitMap_surjective** {V : Type} [AddCommGroup V] [Module ℂ V] {ρ : Representation ℂ G V} (h : ρ.IsIrreducible) {v : V} (hv : v ≠ 0) : Function.Surjective (orbitMap ρ v)

theorem **cl_eq_iff_rep** {σ : IrrRep G} {c : IrrCl G} : σ.cl = c ↔ Nonempty (σ.ρ.Equiv c.rep.ρ)

theorem **irrCl_eq_iff** {c d : IrrCl G} : c = d ↔ Nonempty (c.rep.ρ.Equiv d.rep.ρ)

theorem **clTwist_cl** (σ : IrrRep G) (χ : G →* ℂˣ) (hχ : IsSmoothChar χ) : clTwist σ.cl χ hχ = (irrTwist σ χ hχ).cl

theorem **clTwist_clTwist** (c : IrrCl G) (χ ψ : G →* ℂˣ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : clTwist (clTwist c χ hχ) ψ hψ = clTwist c (χ * ψ) (hχ.mul hψ)

theorem **clTwist_one** (c : IrrCl G) (h : IsSmoothChar (1 : G →* ℂˣ)) : clTwist c 1 h = c

theorem **clTwist_ofChar** (χ ψ : G →* ℂˣ) (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : clTwist (IrrRep.ofChar χ hχ).cl ψ hψ = (IrrRep.ofChar (χ * ψ) (hχ.mul hψ)).cl

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NonemptyEquivProdIrrZero

theorem **nonempty_equiv_parInd_zero** {d : Fin 0 → ℕ} (e : (Σ a, Fin (d a)) ≃ Fin 0) (ρ : Representation ℂ (Levi k fun a => Fin (d a)) V) (L : ℂ ≃ₗ[ℂ] V) : Nonempty ((detRep (Fin 0) (1 : kˣ →* ℂˣ)).Equiv (parInd e ρ))

theorem **nonempty_equiv_prodIrr_zero** (x : Fin 0 → IrrGL k) (hd : ∑ a, (x a).1 = 0) : Nonempty ((detRep (Fin 0) (1 : kˣ →* ℂˣ)).Equiv (prodIrr x hd))

theorem **empty_spec** : IsZ (0 : Multiset (Segment k)) (detRep (Fin 0) (1 : kˣ →* ℂˣ)) ∧ IsL (0 : Multiset (Segment k)) (detRep (Fin 0) (1 : kˣ →* ℂˣ))

theorem **IsL_deg** {N : ℕ} {m : Multiset (Segment k)} {π : Representation ℂ (GL (Fin N) k) V} (h : IsL m π) : (m.map Segment.deg).sum = N

theorem **eq_of_isZSeg_single** {ρ x : IrrGL k} (hρ : ρ.IsCusp) (h : IsZSeg (.single ρ hρ) x.rep) : x = ρ

theorem **eq_of_embeds** {e : (Σ _ : Fin 2, Fin 1) ≃ Fin 2} {χ : Fin 2 → kˣ →* ℂˣ} {ψ : kˣ →* ℂˣ} (h : Embeds (detRep (Fin 2) ψ) (multiProd e fun a => detRep (Fin 1) (χ a))) : χ 1 = χ 0 * absPow k 1

theorem **eq_of_embeds_prodIrr** {c : Fin 2 → kˣ →* ℂˣ} {ψ : kˣ →* ℂˣ} (hc : ∀ a, IsSmoothChar (c a)) {x : Fin 2 → IrrGL k} (hx : x = fun a => IrrGL.ofChar (c a) (hc a)) {hd : ∑ a, (x a).1 = 2} (h : Embeds (detRep (Fin 2) ψ) (prodIrr x hd)) : c 1 = c 0 * absPow k 1

theorem **not_isZ_pair** {χ ψ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) : ¬ IsZ {.single _ (IrrGL.isCusp_ofChar χ hχ), .single _ ((IrrGL.isCusp_ofChar χ hχ).nu 1)} (detRep (Fin 2) ψ)

theorem **not_isGeneric_detRep** (χ : kˣ →* ℂˣ) : ¬ IsGeneric (detRep (Fin 2) χ)

theorem **linked_tests** (χ : kˣ →* ℂˣ) (hχ : IsSmoothChar χ) : (Segment.icc _ (IrrGL.isCusp_ofChar χ hχ) 0 1 (by decide)).Precedes (.icc _ (IrrGL.isCusp_ofChar χ hχ) 1 2 (by decide)) ∧ ¬ (Segment.icc _ (IrrGL.isCusp_ofChar χ hχ) 0 2 (by decide)).Linked (.icc _ (IrrGL.isCusp_ofChar χ hχ) 1 1 (by decide))

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NonemptyExtAnd

theorem **RealSide.space_eq** {D : D7} (X : RealSide D) (j : Fin D.R.m) : X.core.V ((j : ℕ) + 1) X.core.σ₀ = Wc D j

theorem **TwAt.Sb_eq** {H : NExpRealH R cG S YS} {cf : CharFrame D} (a : TwAt hL D nch xd first R F cG geo S YS H cf) : a.X.R.Sb = R.Sb

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NonemptyPtsOfReading

theorem **frBor_one** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum H} {l : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H} {Xo : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Moduli Dw l} {c : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l} {A : Type} [CommRing A] [Algebra c.F A] (w : FrPt Xo c A) : FrBor 1 w w

theorem **isUnitDiag_one** {p : ℕ} [Fact p.Prime] {F ι : Type} [Field F] {φ : F →+* ℂ_[p]} : IsUnitDiag (1 : ι → (PAt φ)ˣ)

theorem **ev_some** {y z} (h : Z.pt y = some z) (f) : T.ev i _ b f y = valAt YS fw j z.1 (toK YS j b f)

theorem **ev_none** {y} (h : Z.pt y = none) (f) : T.ev i _ b f y = 0

theorem **ev_eq_evI** {y z} (h : Z.pt y = some z) : ∃ (Tc : CDVR) (io : T0.Q.c.F →+* Tc.Fr) (k : Tc.Fr →+* ℂ_[D.ntw.l i]) (_ : IsReading (embC YS (T.iC i) b) Tc io k) (y₀ : IntFr (T0.Q.Xo ((j : ℕ) + 1)) T0.Q.c Tc io) (u : ℂ_[D.ntw.l i]), y₀.IsAlg ∧ ‖u‖ = 1 ∧ ∀ (a : NX.KB S i b) (x : NX.FmK S ((j : ℕ) + 1)), T.ev i _ b ((toK YS j b).symm (a ⊗ₜ[S.Ks] x : NX.KB S i b ⊗[S.Ks] _)) y = T.iC i a * (u * k (evI y₀ (cuspOf YS fw j x : T0.R ((j : ℕ) + 1))))

theorem **exists_ev_eq** (z : Pts YS (T.iC i) b ((j : ℕ) + 1)) : ∃ (y : T.Pt i ((j : ℕ) + 1)) (u : ℂ_[D.ntw.l i]), ‖u‖ = 1 ∧ ∀ f, valAt YS fw j z.1 (toK YS j b f) = u * T.ev i _ b f y

theorem **exists_ev_eq_evI** {Tc : CDVR} {io : T0.Q.c.F →+* Tc.Fr} {k : Tc.Fr →+* ℂ_[D.ntw.l i]} (hk : IsReading (embC YS (T.iC i) b) Tc io k) (y₀ : IntFr (T0.Q.Xo ((j : ℕ) + 1)) T0.Q.c Tc io) (hy : y₀.IsAlg) : ∃ (y : T.Pt i ((j : ℕ) + 1)) (u : ℂ_[D.ntw.l i]), ‖u‖ = 1 ∧ ∀ (a : NX.KB S i b) (x : NX.FmK S ((j : ℕ) + 1)), T.iC i a * k (evI y₀ (cuspOf YS fw j x : T0.R ((j : ℕ) + 1))) = u * T.ev i _ b ((toK YS j b).symm (a ⊗ₜ[S.Ks] x : NX.KB S i b ⊗[S.Ks] _)) y

theorem **norm_le_iff** {B : ℝ} (hB : 0 ≤ B) (f : D.nexp.Fm i ((j : ℕ) + 1) b) : (∀ y, ‖T.ev i _ b f y‖ ≤ B) ↔ ∀ z : Pts YS (T.iC i) b ((j : ℕ) + 1), ‖valAt YS fw j z.1 (toK YS j b f)‖ ≤ B

theorem **ball_iff** {B : ℝ} (hB : 0 ≤ B) (f : D.nexp.Fm i ((j : ℕ) + 1) b) : (f ∈ T.FmUr i _ b ∧ ∀ y, ‖T.ev i _ b f y‖ ≤ B) ↔ toK YS j b f ∈ urBall YS fw (T.iC i) b j B

theorem **norm_hkU** {k P} (h : HkIdx i k P) {y z} (hy : Z.pt y = some z) (c) : ‖T.hkU i _ b k P c y‖ = 1

theorem **norm_hkU_all** {k P} (h : HkIdx i k P) (c) (y) : ‖T.hkU i ((j : ℕ) + 1) b k P c y‖ = 1

theorem **hecke** {k P} (h : HkIdx i k P) : ∃ s : T0.Q.c.F, ‖embC YS (T.iC i) b s‖ = 1 ∧ ∀ y z, Z.pt y = some z → ∃ quot : HkKer P.asIdeal (charAt P) k z.1 → FrC YS (T.iC i) b ((j : ℕ) + 1), (∀ C, IsHkQuot P.asIdeal (charAt P) k z.1 C (quot C)) ∧ ∀ f, valAt YS fw j z.1 (toK YS j b (D.nexp.hkF i _ b k P f)) = embC YS (T.iC i) b s * ∑ᶠ C, valAt YS fw j (quot C) (toK YS j b f)

theorem **exists_hkF_eq** {k P} (h : HkIdx i k P) : ∃ s : T0.Q.c.F, ‖embC YS (T.iC i) b s‖ = 1 ∧ ∀ (f : D.nexp.Fm i ((j : ℕ) + 1) b) (g : NX.Fm S i ((j : ℕ) + 1) b), (∀ (z : Pts YS (T.iC i) b ((j : ℕ) + 1)) (quot : HkKer P.asIdeal (charAt P) k z.1 → FrC YS (T.iC i) b ((j : ℕ) + 1)), (∀ C, IsHkQuot P.asIdeal (charAt P) k z.1 C (quot C)) → valAt YS fw j z.1 g = ∑ᶠ C, valAt YS fw j (quot C) (toK YS j b f)) → toK YS j b (D.nexp.hkF i _ b k P f) = sideK YS i b s • g

theorem **hkF_mem_fmUr** {k P} (h : HkIdx i k P) (f : D.nexp.Fm i ((j : ℕ) + 1) b) {g : NX.Fm S i ((j : ℕ) + 1) b} (hg : g ∈ ratForms S _ b (kurAt YS (T.iC i) b)) (hfg : ∀ (z : Pts YS (T.iC i) b ((j : ℕ) + 1)) (quot : HkKer P.asIdeal (charAt P) k z.1 → FrC YS (T.iC i) b ((j : ℕ) + 1)), (∀ C, IsHkQuot P.asIdeal (charAt P) k z.1 C (quot C)) → valAt YS fw j z.1 g = ∑ᶠ C, valAt YS fw j (quot C) (toK YS j b f)) : D.nexp.hkF i _ b k P f ∈ T.FmUr i _ b

theorem **detects** : Detects YS fw (T.iC i) b j

theorem **mem_span_toK** {i : D.ntw.Tw} {j : Fin D.R.m} {b : Bool} {s : Set (D.nexp.Fm i ((j : ℕ) + 1) b)} {s' : Set (NX.Fm S i ((j : ℕ) + 1) b)} (h : ∀ f, f ∈ s ↔ toK YS j b f ∈ s') (f : D.nexp.Fm i ((j : ℕ) + 1) b) : f ∈ Submodule.span (D.ntw.O i) s ↔ toK YS j b f ∈ Submodule.span (D.ntw.O i) s'

theorem **SideTie.mem_bsup_iff** {i : D.ntw.Tw} {j : Fin D.R.m} {b : Bool} (Z : SideTie YS fw T i j b) (f : D.nexp.Fm i ((j : ℕ) + 1) b) : f ∈ D.nexp.Bsup i _ b ↔ toK YS j b f ∈ Submodule.span (D.ntw.O i) (urBall YS fw (T.iC i) b j 1)

theorem **SideTie.supPin_iff** {i : D.ntw.Tw} {j : Fin D.R.m} {b : Bool} (Z : SideTie YS fw T i j b) : SupPin YS i j b ↔ SupDescent YS fw (T.iC i) b j

theorem **SideTie.supDescent** {i : D.ntw.Tw} {j : Fin D.R.m} {b : Bool} (Z : SideTie YS fw T i j b) (h : SupPin YS i j b) : SupDescent YS fw (T.iC i) b j

theorem **ThetaTie.kur_eq** (Z : ThetaTie YS fw T) (h : ∀ x : S.Ks, ∃ y : S.Ks, algebraMap S.Ks D.what₂.LL y = star (algebraMap S.Ks D.what₂.LL x)) (i : D.ntw.Tw) (b : Bool) : T.MaximalUnramifiedExtension i = kurAt YS (T.iC i) b

theorem **ThetaTie.root_mem_kur** (Z : ThetaTie YS fw T) (i : D.ntw.Tw) {ζ : FractionRing (D.ntw.O i)} {n : ℕ} (hn : ¬ D.ntw.l i ∣ n) (h : ζ ^ n = 1) : ζ ∈ T.MaximalUnramifiedExtension i

theorem **extLift_succ** {m : ℕ} {α : ℕ → Type} (f : ∀ n : Fin m, α ((n : ℕ) + 1)) (d : ∀ j, α j) (n : Fin m) : extLift f d ((n : ℕ) + 1) = f n

theorem **extLift_off** {m : ℕ} {α : ℕ → Type} (f : ∀ n : Fin m, α ((n : ℕ) + 1)) (d : ∀ j, α j) {j : ℕ} (h : j ∉ Finset.Icc 1 m) : extLift f d j = d j

theorem **lift_cases** {m : ℕ} {p : ℕ → Prop} (h1 : ∀ n : Fin m, p ((n : ℕ) + 1)) (h2 : ∀ j, j ∉ Finset.Icc 1 m → p j) (j : ℕ) : p j

theorem **atSide_same** {P : Bool → Type} (b : Bool) (z : P b) : atSide b ⟨b, z⟩ : Σ b', P b' = some z

theorem **atSide_other** {P : Bool → Type} {b : Bool} {y : Σ b', P b'} (h : y.1 ≠ b) : atSide b y = none

theorem **eq_of_atSide** {P : Bool → Type} {b : Bool} {y : Σ b', P b'} {z : P b} (h : atSide b y = some z) : y = ⟨b, z⟩

theorem **allMove_fst** (j k : ℕ) (P : PlE D.host) (c) (y : AllPts YS iC j) : (allMove YS iC j k P c y).1 = y.1

theorem **GeoLaws.move** (h : GeoLaws YS fw iC) {j : Fin D.R.m} {k P} (hk : HkIdx i k P) (b : Bool) (z : Pts YS iC b ((j : ℕ) + 1)) : ∃ (e : Fin (kerNum YS iC ((j : ℕ) + 1) k P) ≃ HkKer P.asIdeal (charAt P) k z.1) (q : HkKer P.asIdeal (charAt P) k z.1 → Pts YS iC b ((j : ℕ) + 1)), (∀ C, IsHkQuot P.asIdeal (charAt P) k z.1 C (q C).1) ∧ ∀ c, allMove YS iC _ k P c ⟨b, z⟩ = ⟨b, q (e c)⟩

theorem **GeoLaws.hkUnit_spec** (h : GeoLaws YS fw iC) {j : Fin D.R.m} {k P} (hk : HkIdx i k P) (b : Bool) : ‖hkUnit YS fw iC j b k P‖ = 1 ∧ HeckeIs YS fw iC b j k P (hkUnit YS fw iC j b k P)

theorem **exists_hkUnit_eq** (j : Fin D.R.m) (b : Bool) (k : ℕ) (P : PlE D.host) : ∃ s : T0.Q.c.F, hkUnit YS fw iC j b k P = embC YS iC b s

theorem **allEv_same** {j : Fin D.R.m} {b : Bool} (f : D.nexp.Fm i ((j : ℕ) + 1) b) (z : Pts YS iC b ((j : ℕ) + 1)) : allEv YS fw iC j b f ⟨b, z⟩ = valAt YS fw j z.1 (toK YS j b f)

theorem **allEv_other** {j : Fin D.R.m} {b : Bool} (f : D.nexp.Fm i ((j : ℕ) + 1) b) {y : AllPts YS iC ((j : ℕ) + 1)} (h : y.1 ≠ b) : allEv YS fw iC j b f y = 0

theorem **allEv_ball** {j : Fin D.R.m} {b : Bool} {B : ℝ} (hB : 0 ≤ B) (f : D.nexp.Fm i ((j : ℕ) + 1) b) : (∀ y, ‖allEv YS fw iC j b f y‖ ≤ B) ↔ ∀ z : Pts YS iC b ((j : ℕ) + 1), ‖valAt YS fw j z.1 (toK YS j b f)‖ ≤ B

theorem **evN_succ** (j : Fin D.R.m) : evN YS fw (iC

theorem **mem_fmUrN** (j : Fin D.R.m) (b : Bool) (f) : f ∈ fmUrN YS iC ((j : ℕ) + 1) b ↔ toK YS j b f ∈ ratForms S _ b (kurAt YS iC b)

theorem **hkUnitN_succ** (j : Fin D.R.m) (b : Bool) (k : ℕ) (P : PlE D.host) : hkUnitN YS fw iC ((j : ℕ) + 1) b k P = hkUnit YS fw iC j b k P

theorem **ev_smul** (j : ℕ) (b : Bool) (a : FractionRing (D.ntw.O i)) (f : D.nexp.Fm i j b) (y : AllPts YS iC j) : evN YS fw j b (a • f) y = iC a * evN YS fw j b f y

theorem **ev_inj** (j : ℕ) (b : Bool) : Function.Injective (evN YS fw (iC

theorem **hk_ev** (j : ℕ) (b : Bool) {k : ℕ} {P : PlE D.host} (hk : HkIdx i k P) (f : D.nexp.Fm i j b) (y : AllPts YS iC j) : evN YS fw j b (D.nexp.hkF i j b k P f) y = ∑ c, hkUnitN YS fw iC j b k P * evN YS fw j b f (allMove YS iC j k P c y)

theorem **fmUr_smul** (j : ℕ) (b : Bool) (a : FractionRing (D.ntw.O i)) (f : D.nexp.Fm i j b) (ha : a ∈ kurAt YS iC true ⊓ kurAt YS iC false) (hf : f ∈ fmUrN YS iC j b) : a • f ∈ fmUrN YS iC j b

theorem **fmUr_span** (j : ℕ) (b : Bool) : Submodule.span (FractionRing (D.ntw.O i)) (fmUrN YS iC j b : Set (D.nexp.Fm i j b)) = ⊤

theorem **bsup_def** (j : ℕ) (b : Bool) : D.nexp.Bsup i j b = Submodule.span (D.ntw.O i) {f | f ∈ fmUrN YS iC j b ∧ ∀ y, ‖evN YS fw (iC

theorem **GeoLaws.hk_tie** (h : GeoLaws YS fw iC) (j : Fin D.R.m) (b : Bool) {k P} (hk : HkIdx i k P) : ∃ s : T0.Q.c.F, ‖embC YS iC b s‖ = 1 ∧ ∀ (y : AllPts YS iC ((j : ℕ) + 1)) (z : Pts YS iC b ((j : ℕ) + 1)), atSide b y = some z → ∃ e : Fin (kerNum YS iC ((j : ℕ) + 1) k P) ≃ HkKer P.asIdeal (charAt P) k z.1, ∀ c, ∃ (z' : Pts YS iC b ((j : ℕ) + 1)) (t : Ix (T0.Q.Dw ((j : ℕ) + 1)) → (PAt (embC YS iC b))ˣ) (w : FrC YS iC b ((j : ℕ) + 1)), atSide b (allMove YS iC _ k P c y) = some z' ∧ IsHkQuot P.asIdeal (charAt P) k z.1 (e c) w ∧ IsUnitDiag t ∧ FrBor t z'.1 w ∧ hkUnitN YS fw iC ((j : ℕ) + 1) b k P = embC YS iC b s * wtChar (wtA (κ j)) (κ j).n t

theorem **evN_eq** (j : Fin D.R.m) (b : Bool) (f : D.nexp.Fm i ((j : ℕ) + 1) b) (y : AllPts YS iC ((j : ℕ) + 1)) : evN YS fw ((j : ℕ) + 1) b f y = (atSide b y).elim 0 fun z => valAt YS fw j z.1 (toK YS j b f)

theorem **nonempty_pts_of_reading** {b : Bool} {j : ℕ} {Tc : CDVR} {io : T0.Q.c.F →+* Tc.Fr} {k : Tc.Fr →+* C_[D.ntw.l i]} (hk : IsReading (embC YS iC b) Tc io k) (y₀ : IntFr (T0.Q.Xo j) T0.Q.c Tc io) (hy : y₀.IsAlg) : Nonempty (Pts YS iC b j)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NormLeOneOfMemOL

theorem **galTilt_apply** (g : GQ l) (x : OCb l) : galTilt l g x = Per.σF g x

theorem **galW_inclW** (g : GQ l) (a : Ainf l) : galW l g (inclW l a) = inclW l (Per.σA g a)

theorem **frobW_inclW** (a : Ainf l) : frobW l (inclW l a) = inclW l (WittVector.frobenius a)

theorem **locAinf_actT_eq** {P : LocPrime l} (AI : LocAinf l P) : AI.actT = galTilt l

theorem **locAinf_actWCbHom_eq** {P : LocPrime l} (AI : LocAinf l P) : AI.actWCbHom = galW l

theorem **inclW_injective** : Function.Injective (inclW l)

theorem **ainf_torsionFree** (x : Ainf l) (h : (l : Ainf l) * x = 0) : x = 0

theorem **frobenius_eq_pow_add** (a : Ainf l) : ∃ b, WittVector.frobenius a = a ^ l + (l : Ainf l) * b

theorem **theta_onto** : Function.Surjective (Per.θ l)

theorem **exists_generator** : ∃ ξ : Ainf l, RingHom.ker (Per.θ l) = Ideal.span {ξ} ∧ ξ ≠ 0 ∧ Per.θ l (WittVector.frobenius ξ) = (l : OC l)

theorem **isUnit_inclW_of_span** {d : Ainf l} (hd : Ideal.span {d} = RingHom.ker (Per.θ l)) : IsUnit (inclW l d)

theorem **ainfQuot_mk** (a : Ainf l) : ainfQuot l (Ideal.Quotient.mk _ a) = Per.θ l a

theorem **norm_le_one_of_mem_OL** (K : FinExt l) (x : ɪ(OL K)) : ‖((x : ɪK.1) : Om l)‖ ≤ 1

theorem **coe_intToOC** (K : FinExt l) (x : ɪ(OL K)) : ((intToOC K x : OC l) : ℂ_[l]) = (((x : ɪK.1) : Om l) : ℂ_[l])

theorem **ainfObj_φ_bijective** (K : FinExt l) : Function.Bijective (ainfObj K).D.φ

theorem **isAinf_ainfObj** (K : FinExt l) : IsAinf K (ainfObj K) (ainfId K) where

theorem **mk_eq_str_iff** (H : IsAinf K P e) (x : ɪ(OL K)) (a : P.A) : Ideal.Quotient.mk P.I a = P.str x ↔ Per.θ l (e a) = intToOC K x

theorem **map_I** (H : IsAinf K P e) : P.I.map e = RingHom.ker (Per.θ l)

theorem **exists_generator** (H : IsAinf K P e) : ∃ d : ɪP.I, Ideal.span {(d : P.A)} = P.I ∧ (d : P.A) ≠ 0

theorem **isUnit_inclW** (H : IsAinf K P e) {d : P.A} (hd : Ideal.span {d} = P.I) : IsUnit (inclW l (e d))

theorem **e_galHom** (H : IsAinf K P e) (g : ɪ(GK K)) (a : P.A) : e ((H.galHom g).f a) = Per.σA (g : GQ l) (e a)

theorem **eq_galHom** (H : IsAinf K P e) (g : ɪ(GK K)) (h : Hom P P) (hh : ∀ a, e (h.f a) = Per.σA (g : GQ l) (e a)) : h = H.galHom g

theorem **galHom_one** (H : IsAinf K P e) : H.galHom 1 = Hom.id P

theorem **galHom_mul** (H : IsAinf K P e) (g h : ɪ(GK K)) : H.galHom (g * h) = (H.galHom g).comp (H.galHom h)

theorem **ainfHom_f** (K : FinExt l) (g : ɪ(GK K)) (a : Ainf l) : (ainfHom K g).f a = Per.σA (g : GQ l) a

theorem **exists_isAinf** (K : FinExt l) : ∃ (P : Obj l ɪ(OL K)) (e : P.A ≃+* Ainf l), IsAinf K P e ∧ ∀ g ∈ GK K, ∃ h : Hom P P, ∀ a, e (h.f a) = Per.σA g (e a)

theorem **SemiBC.hom_ext** {A B M N Q : Type} [CommRing A] [CommRing B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module B N] [AddCommGroup Q] [Module B Q] {f : A →+* B} {u : M →ₛₗ[f] N} (h : SemiBC u) {w w' : N →ₗ[B] Q} (hw : ∀ x, w (u x) = w' (u x)) : w = w'

theorem **SemiBC.comp_equiv** {A B M N N' : Type} [CommRing A] [CommRing B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module B N] [AddCommGroup N'] [Module B N'] {f : A →+* B} {u : M →ₛₗ[f] N} (h : SemiBC u) (s : N ≃ₗ[B] N') : SemiBC (s.toLinearMap.comp u)

theorem **exists_isFrac** {A M : Type} [CommRing A] [AddCommGroup M] [Module A M] (d : A) (y : LocalizedModule (Submonoid.powers d) M) : ∃ m k, IsFrac d y m k

theorem **FCrys.frac_agree** {l : ℕ} {R : Type} [CommRing R] (E : FCrys l R) (P : Obj l R) [IsDomain P.A] (d d₀ : ɪP.I) (hd : (d : P.A) ≠ 0) (h0 : (d₀ : P.A) ≠ 0) (x m n : E.M P) (k i : ℕ) (h : IsFrac (d : P.A) (E.Φ P d x) m k) (hn : IsFrac (d₀ : P.A) (E.Φ P d₀ x) n i) : (d₀ : P.A) ^ i • m = (d : P.A) ^ k • n

theorem **smul_eq_of_isUnit** {A B M N : Type} [CommRing A] [CommRing B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module B N] {f : A →+* B} {u : M →ₛₗ[f] N} {d d₀ : A} (hu : IsUnit (f d₀)) {m n : M} {k i : ℕ} (h : d₀ ^ i • m = d ^ k • n) {y : N} (hy : f d₀ ^ i • y = u n) : f d ^ k • y = u m

theorem **eq_of_isFrac** {A : Type} [CommRing A] [IsDomain A] {d : A} (hd : d ≠ 0) (y m : A) (k : ℕ) (h : IsFrac d (algebraMap A (Localization.Away d) y) m k) : m = d ^ k * y

theorem **one_tmul_injective** (T : Type) [AddCommGroup T] [Module ℤ_[l] T] : Function.Injective fun t : T => (1 : WCb l) ⊗ₜ[ℤ_[l]] t

theorem **SemiBC.finite_projective** {A B M N : Type} [CommRing A] [CommRing B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module B N] {f : A →+* B} {u : M →ₛₗ[f] N} (h : SemiBC u) [Module.Finite A M] [Module.Projective A M] : Module.Finite B N ∧ Module.Projective B N

theorem **frobW_fixed_iff** (a : WCb l) : frobW l a = a ↔ ∃ z : ℤ_[l], a = algebraMap ℤ_[l] (WCb l) z

theorem **frobW_map_fixed_iff** {T : Type} [AddCommGroup T] [Module ℤ_[l] T] [Module.Free ℤ_[l] T] [Module.Finite ℤ_[l] T] (y : WCb l ⊗[ℤ_[l]] T) : TensorProduct.map (frobW l).toLinearMap LinearMap.id y = y ↔ ∃ t : T, y = (1 : WCb l) ⊗ₜ[ℤ_[l]] t

theorem **of_generator** (H : IsAinf K P e) (bc : SemiBC j) (d₀ : ɪP.I) (hd : Ideal.span {(d₀ : P.A)} = P.I) (hφ : ∀ (x m : E.M P) (k : ℕ), IsFrac (d₀ : P.A) (E.Φ P d₀ x) m k → inclW l (e d₀) ^ k • TensorProduct.map (frobW l).toLinearMap LinearMap.id (j x) = j m) (hact : ∀ (g : ɪ(GK K)) (x : E.M P), j (E.tr (H.galHom g) x) = TensorProduct.map (galW l (g : GQ l)).toLinearMap (ρ g) (j x)) : Realises K E P e T ρ j

theorem **j_act_of** (H : Realises K E P e T ρ j) (g : ɪ(GK K)) (h : Hom P P) (hh : ∀ a, e (h.f a) = Per.σA (g : GQ l) (e a)) (x : E.M P) : j (E.tr h x) = TensorProduct.map (galW l (g : GQ l)).toLinearMap (ρ g) (j x)

theorem **end_ext** (bc : SemiBC j) (s : WCb l ≃ₐ[ℤ_[l]] WCb l) {r r' : T →ₗ[ℤ_[l]] T} (h : ∀ x, TensorProduct.map s.toLinearMap r (j x) = TensorProduct.map s.toLinearMap r' (j x)) : r = r'

theorem **ρ_unique** (H : Realises K E P e T ρ j) (H' : Realises K E P e T ρ' j) : ρ = ρ'

theorem **ρ_one** (H : Realises K E P e T ρ j) : ρ 1 = LinearMap.id

theorem **ρ_mul** (H : Realises K E P e T ρ j) (g h : ɪ(GK K)) : ρ (g * h) = ρ g ∘ₗ ρ h

theorem **finite** (H : Realises K E P e T ρ j) : Module.Finite ℤ_[l] T

theorem **free** (H : Realises K E P e T ρ j) : Module.Free ℤ_[l] T

theorem **fixed_iff** (H : Realises K E P e T ρ j) (y : WCb l ⊗[ℤ_[l]] T) : TensorProduct.map (frobW l).toLinearMap LinearMap.id y = y ↔ ∃ t : T, y = (1 : WCb l) ⊗ₜ[ℤ_[l]] t

theorem **unitMap_apply** {K : FinExt l} (P : Obj l ɪ(OL K)) (e : P.A ≃+* Ainf l) (a : P.A) : unitMap P e a = inclW l (e a) ⊗ₜ[ℤ_[l]] (1 : ℤ_[l])

theorem **IsAinf.realises_unit** {K : FinExt l} {P : Obj l ɪ(OL K)} {e : P.A ≃+* Ainf l} (H : IsAinf K P e) : Realises K (FCrys.unit l ɪ(OL K)) P e ℤ_[l] (fun _ => LinearMap.id) (unitMap P e) where

theorem **isRealisation_unit** (K : FinExt l) : IsRealisation K (FCrys.unit l ɪ(OL K)) ℤ_[l] fun _ => LinearMap.id

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NormPowLeNormFactorial

theorem **norm_coeff_mul_le** {w : ℕ → ℝ} (hw : ∀ n, 0 ≤ w n) (hm : ∀ i j, w (i + j) ≤ w i * w j) {f g : K⟦X⟧} {a b : ℝ} (ha : 0 ≤ a) (hb : 0 ≤ b) (hf : ∀ n, ‖coeff n f‖ * w n ≤ a) (hg : ∀ n, ‖coeff n g‖ * w n ≤ b) (n : ℕ) : ‖coeff n (f * g)‖ * w n ≤ a * b

theorem **tendsto_coeff_mul** {w : ℕ → ℝ} (hw : ∀ n, 0 ≤ w n) (hm : ∀ i j, w (i + j) ≤ w i * w j) {f g : K⟦X⟧} (hf : Tendsto (fun n => ‖coeff n f‖ * w n) atTop (𝓝 0)) (hg : Tendsto (fun n => ‖coeff n g‖ * w n) atTop (𝓝 0)) : Tendsto (fun n => ‖coeff n (f * g)‖ * w n) atTop (𝓝 0)

theorem **norm_coeff_add_le** {w : ℕ → ℝ} (hw : ∀ n, 0 ≤ w n) (f g : K⟦X⟧) (n : ℕ) : ‖coeff n (f + g)‖ * w n ≤ max (‖coeff n f‖ * w n) (‖coeff n g‖ * w n)

theorem **norm_natCast_le_of_dvd** {a b : ℕ} (h : a ∣ b) : ‖(b : K)‖ ≤ ‖(a : K)‖

theorem **wS_nonneg** (e n : ℕ) : 0 ≤ wS K e n

theorem **wS_le_one** (e n : ℕ) : wS K e n ≤ 1

theorem **wS_add** (e i j : ℕ) : wS K e (i + j) ≤ wS K e i * wS K e j

theorem **wS_anti** (e : ℕ) {i j : ℕ} (h : i ≤ j) : wS K e j ≤ wS K e i

theorem **norm_algebraMap_le** (c : O) : ‖algebraMap O K c‖ ≤ 1

theorem **algebraMap_eq_C** (c : O) : algebraMap O K⟦X⟧ c = C (algebraMap O K c)

theorem **mem_A** {f : K⟦X⟧} : f ∈ A O K ↔ ∀ n, ‖coeff n f‖ ≤ 1

theorem **range_map_eq_A** : (PowerSeries.mapAlgHom (Algebra.ofId O K)).range = A O K

@[simp] theorem **coe_equivA** (f : O⟦X⟧) : (equivA O K f : K⟦X⟧) = PowerSeries.map (algebraMap O K) f

theorem **mem_S** {e : ℕ} {f : K⟦X⟧} : f ∈ S O K e ↔ (∀ n, ‖coeff n f‖ * wS K e n ≤ 1) ∧ Tendsto (fun n => ‖coeff n f‖ * wS K e n) atTop (𝓝 0)

theorem **rad_nonneg** (l : ℕ) (k : ℝ) : 0 ≤ rad l k

theorem **mem_R** {l : ℕ} {k : ℝ} {f : K⟦X⟧} : f ∈ R O K l k ↔ Tendsto (fun n => ‖coeff n f‖ * rad l k ^ n) atTop (𝓝 0)

theorem **le_normR** {l : ℕ} {k : ℝ} {f : K⟦X⟧} (hf : f ∈ R O K l k) (n : ℕ) : ‖coeff n f‖ * rad l k ^ n ≤ normR l k f

theorem **mem_away** {B : Subalgebra O K⟦X⟧} {s : B} {f : K⟦X⟧} : f ∈ away B s ↔ ∃ n : ℕ, (s : K⟦X⟧) ^ n * f ∈ B

theorem **le_away** (B : Subalgebra O K⟦X⟧) (s : B) : B ≤ away B s

@[simp] theorem **coe_algebraMap_away** {B : Subalgebra O K⟦X⟧} {s : B} (f : B) : (algebraMap B (away B s) f : K⟦X⟧) = f

theorem **isLocalization_away** (B : Subalgebra O K⟦X⟧) (s : B) (hs : IsUnit (s : K⟦X⟧)) : IsLocalization.Away s (away B s)

@[simp] theorem **coe_algebraMap_K** {B : Subalgebra O K⟦X⟧} [HasK B] (c : K) : (algebraMap K B c : K⟦X⟧) = C c

theorem **E_mem** (l e : ℕ) (B : Subalgebra O K⟦X⟧) [HasU B] : E K l e ∈ B

@[simp] theorem **coe_uIn** (B : Subalgebra O K⟦X⟧) [HasU B] : (uIn B : K⟦X⟧) = X

@[simp] theorem **coe_EIn** (l e : ℕ) (B : Subalgebra O K⟦X⟧) [HasU B] : (EIn l e B : K⟦X⟧) = E K l e

theorem **away_mono** {B B' : Subalgebra O K⟦X⟧} (h : B ≤ B') {s : B} {s' : B'} (hs : (s : K⟦X⟧) = s') : away B s ≤ away B' s'

theorem **N_apply** (f : K⟦X⟧) : N K f = X * PowerSeries.derivative K f

theorem **coeff_N** (f : K⟦X⟧) (n : ℕ) : coeff n (N K f) = n * coeff n f

@[simp] theorem **frob_X** (l : ℕ) [Fact l.Prime] : frob K l X = X ^ l

@[simp] theorem **frob_C** (l : ℕ) [Fact l.Prime] (c : K) : frob K l (C c) = C c

theorem **coeff_frob** (l : ℕ) [Fact l.Prime] (f : K⟦X⟧) (n : ℕ) : coeff n (frob K l f) = if l ∣ n then coeff (n / l) f else 0

@[simp] theorem **constantCoeff_frob** (l : ℕ) [Fact l.Prime] (f : K⟦X⟧) : constantCoeff (frob K l f) = constantCoeff f

theorem **frob_injective** (l : ℕ) [Fact l.Prime] : Function.Injective (frob K l)

theorem **N_frob** (l : ℕ) [Fact l.Prime] (f : K⟦X⟧) : N K (frob K l f) = (l : K⟦X⟧) * frob K l (N K f)

theorem **N_E** (l e : ℕ) : N K (E K l e) = (e : K⟦X⟧) * X ^ e

theorem **constantCoeff_E** (l e : ℕ) [NeZero e] : constantCoeff (E K l e) = l

theorem **monomial_mem_S** {e n : ℕ} {c : K} (h : ‖c‖ * wS K e n ≤ 1) : monomial n c ∈ S O K e

theorem **frob_mem_A** (l : ℕ) [Fact l.Prime] {f : K⟦X⟧} (hf : f ∈ A O K) : frob K l f ∈ A O K

theorem **frob_bound** (l : ℕ) [Fact l.Prime] {w : ℕ → ℝ} (hw : ∀ n, 0 ≤ w n) (ha : Antitone w) (f : K⟦X⟧) (n : ℕ) : ‖coeff n (frob K l f)‖ * w n ≤ ‖coeff (n / l) f‖ * w (n / l)

theorem **frob_mem_S** (l : ℕ) [Fact l.Prime] {e : ℕ} {f : K⟦X⟧} (hf : f ∈ S O K e) : frob K l f ∈ S O K e

theorem **rad_le_one** (l : ℕ) [Fact l.Prime] {k : ℝ} (hk : 0 ≤ k) : rad l k ≤ 1

theorem **rad_lt_one** (l : ℕ) [Fact l.Prime] {k : ℝ} (hk : 0 < k) : rad l k < 1

theorem **rad_pos** (l : ℕ) [Fact l.Prime] (k : ℝ) : 0 < rad l k

theorem **frob_mem_R** (l : ℕ) [Fact l.Prime] {k : ℝ} (hk : 0 ≤ k) {f : K⟦X⟧} (hf : f ∈ R O K l k) : frob K l f ∈ R O K l k

theorem **R_mono** (l : ℕ) [Fact l.Prime] {k k' : ℝ} (h : k ≤ k') : R O K l k ≤ R O K l k'

theorem **A_le_R** (l : ℕ) [Fact l.Prime] {k : ℝ} (hk : 0 < k) : A O K ≤ R O K l k

theorem **norm_coeff_derivative_le** (f : K⟦X⟧) (n : ℕ) : ‖coeff n (PowerSeries.derivative K f)‖ ≤ ‖coeff (n + 1) f‖

theorem **derivative_mem_A** {f : K⟦X⟧} (hf : f ∈ A O K) : PowerSeries.derivative K f ∈ A O K

theorem **derivative_mem_R** (l : ℕ) [Fact l.Prime] {k : ℝ} {f : K⟦X⟧} (hf : f ∈ R O K l k) : PowerSeries.derivative K f ∈ R O K l k

theorem **derivative_mem_S** {e : ℕ} {f : K⟦X⟧} (hf : f ∈ S O K e) : PowerSeries.derivative K f ∈ S O K e

theorem **derivative_mem_away** {B : Subalgebra O K⟦X⟧} (s : B) (hB : ∀ f ∈ B, PowerSeries.derivative K f ∈ B) {f : K⟦X⟧} (hf : f ∈ away B s) : PowerSeries.derivative K f ∈ away B s

theorem **N_mem** {B : Subalgebra O K⟦X⟧} [HasU B] {f : K⟦X⟧} (hf : PowerSeries.derivative K f ∈ B) : N K f ∈ B

theorem **frob_mem_invL** (l : ℕ) [Fact l.Prime] {B : Subalgebra O K⟦X⟧} (hB : ∀ f ∈ B, frob K l f ∈ B) {f : K⟦X⟧} (hf : f ∈ invL l B) : frob K l f ∈ invL l B

theorem **norm_natCast_eq** (n : ℕ) : ‖(n : K)‖ = ‖(n : ℚ_[l])‖

theorem **norm_l** : ‖(l : K)‖ = (l : ℝ)⁻¹

theorem **natCast_ne_zero'** {n : ℕ} (hn : n ≠ 0) : (n : K) ≠ 0

theorem **l_ne_zero** : (l : K) ≠ 0

theorem **norm_l_lt_one** : ‖(l : K)‖ < 1

theorem **norm_pow_le_norm_factorial** (m : ℕ) : ‖(l : K)‖ ^ m ≤ ‖(m.factorial : K)‖

theorem **norm_factorial_le** (m : ℕ) : ‖(m.factorial : K)‖ ≤ ‖(l : K)‖ ^ (m / l)

theorem **tendsto_wS** (e : ℕ) [NeZero e] : Tendsto (wS K e) atTop (𝓝 0)

theorem **A_le_S** (e : ℕ) [NeZero e] : A O K ≤ S O K e

theorem **rad_pow** (e : ℕ) [NeZero e] : rad l (1 / e) ^ e = ‖(l : K)‖

theorem **S_le_R** (e : ℕ) [NeZero e] : S 0 K e ≤ R 0 K l (1 / e)

theorem **exists_pow_mul_mem** (c : K) : ∃ (n : ℕ) (b : O), (l : K) ^ n * c = algebraMap O K b

theorem **isUnit_l** : IsUnit (l : K⟦X⟧)

theorem **isUnit_E** (e : ℕ) [NeZero e] : IsUnit (E K l e)

theorem **invL_le** {B B' : Subalgebra O K⟦X⟧} [HasK B'] (h : B ≤ B') : invL l B ≤ B'

theorem **Ediv_mem_S** (e : ℕ) [NeZero e] (i : ℕ) : Ediv K l e i ∈ S 0 K e

theorem **mem_Fil** {e j : ℕ} {f : S 0 K e} : f ∈ Fil 0 K l e j ↔ ∀ n : ℕ, ∃ g ∈ FilPre 0 K l e j, ∃ h : S 0 K e, f = g + (l : S 0 K e) ^ n * h

theorem **FilPre_le_Fil** (e j : ℕ) : FilPre 0 K l e j ≤ Fil 0 K l e j

theorem **Ediv_mem_Fil** (e : ℕ) [NeZero e] {j i : ℕ} (h : j ≤ i) : ((Ediv K l e i, Ediv_mem_S l e i) : S 0 K e) ∈ Fil 0 K l e j

theorem **Fil_anti** (e : ℕ) {i j : ℕ} (h : i ≤ j) : Fil 0 K l e j ≤ Fil 0 K l e i

theorem **Fil_zero** (e : ℕ) : Fil 0 K l e 0 = ⊤

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.NotTeichTrivOfIsDelta

theorem **isUnitAt_of_pow_eq_one** {E : Type} [Field E] [NumberField E] {w : HeightOneSpectrum (𝒪 E)} {x : (w.adicCompletion E)ˣ} (hx : x ^ (Ideal.absNorm w.asIdeal − 1) = 1) : IsUnitAt w x

theorem **unrAt_of_tameAt_of_teich** {E : Type} [Field E] [NumberField E] {χ : Formalism.L2.Host.HChar E} {w : HeightOneSpectrum (𝒪 E)} (ht : χ.TameAt w) (h : ∀ x : (w.adicCompletion E)ˣ, x ^ (Ideal.absNorm w.asIdeal − 1) = 1 → χ.loc w x = 1) : χ.UnrAt w

theorem **one** : TeichTriv H 1

theorem **mul** (h : TeichTriv H χ) (h' : TeichTriv H χ') : TeichTriv H (χ * χ')

theorem **inv** (h : TeichTriv H χ) : TeichTriv H χ⁻¹

theorem **of_unr** (h : ∀ w, OverP H H.pde w ∨ OverP H H.pde' w → χ.UnrAt w) : TeichTriv H χ

theorem **pow_eq_one** (h : TeichTriv H χ) {w : HeightOneSpectrum (𝒪 H.E)} (hw : OverP H H.pde w ∨ OverP H H.pde' w) {x : (w.adicCompletion H.E)ˣ} (hx : IsUnitAt w x) : ∃ n : ℕ, χ.loc w x ^ (Ideal.absNorm w.asIdeal ^ n) = 1

theorem **unrAt** (h : TeichTriv H χ) {w : HeightOneSpectrum (𝒪 H.E)} (hw : OverP H H.pde w ∨ OverP H H.pde' w) (ht : χ.TameAt w) : χ.UnrAt w

theorem **not_teichTriv_of_isDelta** {p : ℕ} (hd : IsDelta (H

theorem **Teich.lambda_local_teichmuller** (h : Z.Teich) (l : ℕ) (j a : Fin D.R.m) {w : HeightOneSpectrum (𝒪 D.host.E)} (hw : OverP D.host D.host.pde w ∨ OverP D.host D.host.pde' w) {x : (w.adicCompletion D.host.E)ˣ} (hx : x ^ (Ideal.absNorm w.asIdeal − 1) = 1) : (Z.lam0 l j a).loc w x = (Z.C.mu0 j).loc w x * (Z.C.delAt l).loc w x

theorem **Teich.not_mul_del** (h : Z.Teich) (a : Fin D.R.m) {w : HeightOneSpectrum (𝒪 D.host.E)} (hw : OverP D.host D.host.pde w) : ¬ TeichTriv D.host (Z.chi a * Z.C.del)

theorem **IsLam.range** {i : X.Tw} {ι : ℂ →+* Kbar X i} {lam : X.abelian₂ i} {j a : Fin D.R.m} {s : ℕ} (h : IsLam X Z i ι lam j a s) : 1 ≤ s ∧ s ≤ BlochKatoReduction.Gen.NFun D.R + 1

theorem **isLamTieFull_of_labels** (J A : ∀ i, X.abelian₂ i → Fin D.R.m) (S : ∀ i, X.abelian₂ i → ℕ) (hS : ∀ i, X.NonzeroIndexPlace i → ∀ P, (below P).asIdeal ∣ cond0 D.host Z.w0 → P ∈ X.S0 i) (hι : ∀ i, X.NonzeroIndexPlace i → ∃ ι : ℂ →+* Kbar X i, (∀ x : D.LL0, ι (D.what₂.embLL (x : D.what₂.LL)) = toBar X i (X.emb0 i x)) ∧ ∀ lam : X.abelian₂ i, IsLam X Z i (atSide X (X.sideL i lam) ι) lam (J i lam) (A i lam) (S i lam) ∧ IsThOf X i (atSide X (X.sideL i lam) ι) lam (phi D.host) (S i lam) (Z.kap (J i lam) (A i lam) (S i lam)) (cond0 D.host Z.w0) (Z.lam0 (X.l i) (J i lam) (A i lam))) (hall : ∀ i, X.NonzeroIndexPlace i → ∀ (j a : Fin D.R.m) (s : ℕ) (b : Bool), 1 ≤ s → s ≤ BlochKatoReduction.Gen.NFun D.R + 1 → ∃ lam : X.abelian₂ i, X.sideL i lam = b ∧ J i lam = j ∧ A i lam = a ∧ S i lam = s) : IsLamTieFull X Z

theorem **isLamTie** (h : IsLamTieFull X Z) : IsLamTie X Z

theorem **exists_labels** (h : IsLamTieFull X Z) {i : X.Tw} (hi : X.NonzeroIndexPlace i) (j : Fin D.R.m) {s : ℕ} (b : Bool) (hs : 1 ≤ s) (hs' : s ≤ BlochKatoReduction.Gen.NFun D.R + 1) : ∃ lam : X.abelian₂ i, X.sideL i lam = b ∧ X.jL i lam = j.1 + 1 ∧ X.s2 i lam = 2 * (s : ℤ)

theorem **nonempty** (h : IsLamTieFull X Z) {i : X.Tw} (hi : X.NonzeroIndexPlace i) : Nonempty (X.abelian₂ i)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.OfKillsOne

theorem **Hom.Kills.of_dvd** {i : ZMod n} {f g : O⟦X⟧} (h : F.Kills i f) (hfg : f ∣ g) : F.Kills i g

theorem **Hom.Kills.pow_le** {i : ZMod n} {f : O⟦X⟧} {a a' : ℕ} (h : F.Kills i (f ^ a)) (ha : a ≤ a') : F.Kills i (f ^ a')

theorem **typed_iff** {up : ZMod n → Fin R.e → Bool} {b : ZMod n → Fin R.e → ℕ} {Sn η m h : ℕ} : Typed F up b Sn η m h ↔ TypedOnto F up b Sn η m h ∧ ∀ i, ∃ a : ℕ, F.Kills i ((l : O⟦X⟧) ^ a)

theorem **cHyp_iff** [NeZero n] {b : ZMod n → ℕ} {Sn η m h : ℕ} : CHyp F b Sn η m h ↔ CSet F b Sn η m h ∧ (∀ i, ∃ a : ℕ, F.Kills i ((l : O⟦X⟧) ^ a)) ∧ ∀ i, F.Kills i (R.E ^ aC l (h + η))

theorem **bHyp_iff** [NeZero n] {up : ZMod n → Bool} {sz : ZMod n → ℕ} {b w Sn η m h : ℕ} {A ρ : ZMod n → ℕ} : BHyp F up sz b w Sn η m h A ρ ↔ BSet F up sz b w Sn η m h ∧ CutBounds F sz w Sn η h A ρ

theorem **TypedOnto.jumps_width** {up : ZMod n → Fin R.e → Bool} {b : ZMod n → Fin R.e → ℕ} {Sn η m h : ℕ} (H : TypedOnto F up b Sn η m h) (i : ZMod n) (τ : Fin R.e) : S₁.JumpsIn i τ (Set.Icc (−((if up i τ then η else b i τ + Sn : ℕ) : ℤ)) ((if up i τ then b i τ + Sn else η : ℕ) : ℤ))

theorem **BSet.lt_iff** {up : ZMod n → Bool} {sz : ZMod n → ℕ} {b w Sn η m h : ℕ} (H : BSet F up sz b w Sn η m h) (i : ZMod n) : sz i < b ↔ sz i ≤ w

theorem **CokBound.of_kills_one** (h : ∀ i, F.Kills i 1) (Hw : ZMod n → ℕ) (H : ℕ) : CokBound F Hw H

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.PadicIntGLSmulMemPrincipalCongruenceFixed

theorem **red_surjective** (n : ℕ) : Function.Surjective (red g n)

theorem **mem_principalCongruenceFixedVectors_iff** (π : Representation ℂ (GL (Fin n) ℚ_[g]) V) (v : V) : v ∈ fixK1 π ↔ ∀ k, red g n k = 1 → π (inc g n k) v = v

theorem **lift_red** (π : Representation ℂ (GL (Fin n) ℚ_[g]) V) {v : V} (hv : v ∈ fixK1 π) (k : GL (Fin n) ℤ_[g]) : π (lift g n (red g n k)) v = π (inc g n k) v

theorem **padicIntGL_smul_mem_principalCongruenceFixed** (π : Representation ℂ (GL (Fin n) ℚ_[g]) V) {v : V} (hv : v ∈ fixK1 π) (k : GL (Fin n) ℤ_[g]) : π (inc g n k) v ∈ fixK1 π

theorem **lift_mul** (π : Representation ℂ (GL (Fin n) ℚ_[g]) V) {v : V} (hv : v ∈ fixK1 π) (x y : Gn g n) : π (lift g n (x * y)) v = π (lift g n x) (π (lift g n y) v)

theorem **isIrrRep_iff** {G : Type} [Group G] (π : Representation ℂ G V) : IsIrrRep π ↔ π.IsIrreducible

@[simp] theorem **chRep_apply** {G : Type} [Group G] (φ : G →* ℂˣ) (x : G) (v : ℂ) : chRep φ x v = φ x * v

theorem **isIrrRep_chRep** {G : Type} [Group G] (φ : G →* ℂˣ) : IsIrrRep (chRep φ)

theorem **unitPart_unit** (u : ℤ_[g]ˣ) : unitPart (Units.map (algebraMap ℤ_[g] ℚ_[g]).toMonoidHom u) = u

theorem **tame_inc** (χ : (ZMod g)ˣ →* ℂˣ) (k : GL (Fin n) ℤ_[g]) : tame χ (Matrix.GeneralLinearGroup.det (inc g n k)) = χ (Matrix.GeneralLinearGroup.det (red g n k))

theorem **tame_mem_fixK1** (χ : (ZMod g)ˣ →* ℂˣ) (v : ℂ) : v ∈ fixK1 (detRep n (tame χ))

theorem **tame_lift** (χ : (ZMod g)ˣ →* ℂˣ) (x : Gn g n) (v : ℂ) : detRep n (tame χ) (lift g n x) v = χ (Matrix.GeneralLinearGroup.det x) * v

theorem **det_one'** {R : Type} [CommRing R] (y : GL (Fin 1) R) : ((Matrix.GeneralLinearGroup.det y : Rˣ) : R) = (y : Matrix (Fin 1) (Fin 1) R) 0 0

theorem **det_two'** {R : Type} [CommRing R] (y : GL (Fin 2) R) (h : (y : Matrix (Fin 2) (Fin 2) R) 1 0 = 0) : ((Matrix.GeneralLinearGroup.det y : Rˣ) : R) = (y : Matrix (Fin 2) (Fin 2) R) 0 0 * (y : Matrix (Fin 2) (Fin 2) R) 1 1

theorem **det_surj_one** {R : Type} [CommRing R] (u : Rˣ) : ∃ y : GL (Fin 1) R, Matrix.GeneralLinearGroup.det y = u

theorem **det_blk** (x y : Gn g 1) : Matrix.GeneralLinearGroup.det (blk 1 1 2 x y) = Matrix.GeneralLinearGroup.det x * Matrix.GeneralLinearGroup.det y

theorem **isParVec_tame** (χ χ' : (ZMod g)ˣ →* ℂˣ) (v : ℂ) (hv : v ≠ 0) : IsParVec (detRep 2 (tame χ)) 1 1 χ' v ↔ χ' = χ

theorem **mapsToInd_det** (μ ν₁ ν₂ : ℚ_[g]ˣ →* ℂˣ) : MapsToInd (detRep 2 μ) 1 1 (detRep 1 1) ν₁ ν₂ ↔ ν₁ = μ ∧ ν₂ = μ

theorem **exists_gen** (L : LabelDatum g) : ∃ u : (ZMod g)ˣ, algebraMap (ZMod g) (Formalism.CyclotomicNorms.Fbar g) u = L.gen 1 ∧ orderOf u = g − 1

theorem **isIrr_dch** (χ : (ZMod g)ˣ →* ℂˣ) : IsIrr (dch χ)

theorem **eq_of_ip** {G : Type} [Group G] [Fintype G] {α β : G → ℂ} (ha : IsIrr α) (hb : IsIrr β) (h : ip α β ≠ 0) : α = β

theorem **theta_eq** (L : LabelDatum g) {χ : (ZMod g)ˣ →* ℂˣ} {c : ZMod g} (hc : IsCharLab L χ c) (y : Gn g 1) {z : Formalism.CyclotomicNorms.Fbar g} (hz : z ∈ Formalism.CyclotomicNorms.rootsBar (y : Mt g 1).charpoly) : (-1) ^ (1 − 1) * ∑ i ∈ Finset.range 1, thetaOf L 1 (algebraMap (ZMod g) _ c) (z ^ (g ^ i)) = dch χ y

theorem **isCuspOf_iff** (L : LabelDatum g) {χ : (ZMod g)ˣ →* ℂˣ} {c : ZMod g} (hc : IsCharLab L χ c) (τ : Gn g 1 → ℂ) : IsCuspOf L (X − C c) τ ↔ τ = dch χ

theorem **isClass_dch** (χ : (ZMod g)ˣ →* ℂˣ) : IsClass (dch χ)

theorem **ip_dch** (χ : (ZMod g)ˣ →* ℂˣ) : ip (dch χ) (dch χ) = 1

theorem **eSerC_one** (L : LabelDatum g) {χ : (ZMod g)ˣ →* ℂˣ} {c : ZMod g} (hc : IsCharLab L χ c) : eSerC L 1 (X − C c) = zOf (dch χ)

theorem **hasLab_tame** (L : LabelDatum g) {χ : (ZMod g)ˣ →* ℂˣ} {c : ZMod g} (hc : IsCharLab L χ c) (χ' : (ZMod g)ˣ →* ℂˣ) : HasLab L (detRep 1 (tame χ')) (X − C c) ↔ χ' = χ

theorem **isCharLab_unique** (L : LabelDatum g) {χ : (ZMod g)ˣ →* ℂˣ} {c c' : ZMod g} (h : IsCharLab L χ c) (h' : IsCharLab L χ c') : c = c'

theorem **isCharLab_exists** (L : LabelDatum g) (χ : (ZMod g)ˣ →* ℂˣ) : ∃ c, IsCharLab L χ c

theorem **hasLab_monic** (L : LabelDatum g) (π : Representation ℂ (GL (Fin n) ℚ_[g]) V) (f : (ZMod g)[X]) (h : HasLab L π f) : f.Monic ∧ f.natDegree = n ∧ f.coeff 0 ≠ 0

theorem **mapsToInd_of_too_large** (μ μ' ν₁ ν₂ : ℚ_[g]ˣ →* ℂˣ) : MapsToInd (detRep 1 μ) 1 1 (detRep 1 μ') ν₁ ν₂

theorem **actTop_tame** (χ : (ZMod g)ˣ →* ℂˣ) (x : Gn g 1) (v : ℂ) : actTop (detRep 2 (tame χ)) 1 1 (MonoidAlgebra.single x 1) v = dch χ x * v

theorem **hasLab_inhabited** (χ : (ZMod g)ˣ →* ℂˣ) : ∃ f, HasLab (GLq.Model.datum g) (detRep 1 (tame χ)) f

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.PotentiallySemistableCyclotomicSum

theorem **fr_alg** (m : ℕ) (x : ℚ_[l]) : fr l m (algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) x) = algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) x

theorem **kd_alg** (x : ℚ_[l]) : kd P (algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) x) = algebraMap ℚ_[l] P.R.BdR x

theorem **ks_alg** (x : ℚ_[l]) : ks P (algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) x) = Polynomial.C (algebraMap ℚ_[l] P.R.B x)

theorem **cycSum_val** (K : FinExt l) (a : n → ℤ) (g : ɪ(GK K)) : (cycSum P K a g : Matrix n n ɪ(MaximalUnramifiedExtension l)) = Matrix.diagonal fun j => algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) ((P.χQ (g : GQ l) ^ a j : ℚ_[l]ˣ) : ℚ_[l])

theorem **matD_cycSum** (K : FinExt l) (a : n → ℤ) (m : ℕ) (g : ɪ(GK K)) : matD P (cycSum P K a) m g = Matrix.diagonal fun j => algebraMap ℚ_[l] P.R.BdR ((P.χQ (g : GQ l) ^ a j : ℚ_[l]ˣ) : ℚ_[l])

theorem **isRat_cycSum** {K : FinExt l} {r : ℕ} (hr : 0 < r) (hK : ∀ c : ɪ(MaximalUnramifiedExtension l), fr l r c = c → (c : Om l) ∈ K.1) (a : n → ℤ) : IsRat r (cycSum P K a) where

theorem **isRat_one** (K : FinExt l) (a : n → ℤ) : IsRat 1 (cycSum P K a)

theorem **framed_cycSum** (K : FinExt l) (a : n → ℤ) (r : ℕ) : Framed P (cycSum P K a) (fun _ : Fin r => a)

theorem **actQ_tU** (g : GQ l) (a : ℤ) : P.actQ g ((tU P ^ a : P.R.Bˣ) : P.R.B) = algebraMap ℚ_[l] P.R.B ((P.χQ g ^ a : ℚ_[l]ˣ) : ℚ_[l]) * ((tU P ^ a : P.R.Bˣ) : P.R.B)

theorem **actdR_ɪ** (g : GQ l) (b : P.R.B) : P.actQdR g (P.ɪB b) = P.ɪB (P.actQ g b)

theorem **act_tz** (g : GQ l) (a : ℤ) : P.actQdR g (tz P a) = algebraMap ℚ_[l] P.R.BdR ((P.χQ g ^ a : ℚ_[l]ˣ) : ℚ_[l]) * tz P a

theorem **deRham_of_framed** {K : FinExt l} {r : ℕ} {ρ : ɪ(GK K) →* GL n ɪ(MaximalUnramifiedExtension l)} (hρ : IsRat r ρ) {w : Fin r → n → ℤ} (h : Framed P ρ w) : AdmDR P ρ

theorem **deRham_cyclotomicSum** (K : FinExt l) (a : n → ℤ) : AdmDR P (cycSum P K a)

theorem **potentiallySemistable_cyclotomicSum** (K : FinExt l) (a : n → ℤ) : AdmSt P (cycSum P K a)

theorem **splits_cycSum** (K : FinExt l) (a : Fin 2 → ℤ) (r : ℕ) : Splits r (cycSum P K a)

theorem **not_splits_one** {K : FinExt l} (r : ℕ) (ρ : ɪ(GK K) →* GL (Fin 1) ɪ(MaximalUnramifiedExtension l)) : ¬ Splits r ρ

theorem **one_not_mem** (K : FinExt l) : (1 : P.R.BdR) ∉ (P.PD K).Fil 1

theorem **tz_mem** (K : FinExt l) (d : ℕ) : tz P ((d + 1 : ℕ) : ℤ) ∈ (P.PD K).Fil 1

theorem **no_neg** (K : FinExt l) (u : P.R.BdRˣ) (hu : (u : P.R.BdR) ∈ Fil P 0) (d : ℕ) (h : ∀ g : ɪ(GK K), algebraMap ℚ_[l] P.R.BdR ((P.χQ (g : GQ l) ^ ((d + 1 : ℕ) : ℤ) : ℚ_[l]ˣ) : ℚ_[l]) ∗ P.actQdR (g : GQ l) u = u) : False

theorem **framed_iff** (K : FinExt l) (a b : ℤ) : Framed P (cycSum P K fun _ : Fin 1 => a) (fun (_ : Fin 1) _ => b) ↔ a = b

theorem **cyc_ne_one** (K : FinExt l) : (cycSum P K fun _ : Fin 1 => 1) ≠ 1

theorem **exists_ext** {r : ℕ} (hr : 0 < r) : ∃ K : FinExt l, ∀ c : ɪ(MaximalUnramifiedExtension l), fr l r c = c → (c : Om l) ∈ K.1

theorem **kummer_period** (g : GQ l) : !![ks P (algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) (P.χQ g : ℚ_[l])), ks P (algebraMap ℚ_[l] ɪ(MaximalUnramifiedExtension l) (kummer P g : ℚ_[l]))]; 0, 1] ∗ (!![Polynomial.C ((tU P ^ (-1 : ℤ) : P.R.Bˣ) : P.R.B), -(Polynomial.C ((tU P ^ (-1 : ℤ) : P.R.Bˣ) : P.R.B) ∗ Polynomial.X); 0, 1] : Matrix (Fin 2) (Fin 2) (Bst P)).map (actSt P g) = !![Polynomial.C ((tU P ^ (-1 : ℤ) : P.R.Bˣ) : P.R.B), -(Polynomial.C ((tU P ^ (-1 : ℤ) : P.R.Bˣ) : P.R.B) ∗ Polynomial.X); 0, 1]

theorem **plus_ainf** {AI : LocAinf l P} (T : LocTied l P AI) (a : Ainf l) : Plus T (AI.ιA a)

theorem **exists_model** (l : ℕ) [Fact l.Prime] {r : ℕ} (hr : 0 < r) : ∃ (P : LocPrime l) (AI : LocAinf l P), Nonempty (LocTied l P AI) ∧ ∃ (K : FinExt l) (ρ : ɪ(GK K) →* GL (Fin 1) ɪ(MaximalUnramifiedExtension l)), ρ ≠ 1 ∧ IsRat r ρ ∧ AdmDR P ρ ∧ AdmSt P ρ ∧ Framed P ρ (fun (_ : Fin r) _ => 1) ∧ ¬ Splits r ρ

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.PowerSeriesOneAddConstMulXMulInv

theorem **const_smul_eq** (c : k) (x : k⦅X⦆) : c • x = algebraMap k k⦅X⦆ c ∗ x

theorem **vv_eq_coe_X** : vv k = ((PowerSeries.X : k⟦X⟧) : k⦅X⦆)

theorem **vv_ne_zero** : vv k ≠ 0

theorem **mem_sh_iff_exists** {a : ℤ} {N : Submodule k⟦X⟧ V} {x : V} : x ∈ sh a N ↔ ∃ y ∈ N, vv k ^ a • y = x

theorem **mem_sh_iff_smul_mem** {a : ℤ} {N : Submodule k⟦X⟧ V} {x : V} : x ∈ sh a N ↔ vv k ^ (-a) • x ∈ N

theorem **sh_zero_eq** (N : Submodule k⟦X⟧ V) : sh 0 N = N

theorem **sh_sh_eq** (a b : ℤ) (N : Submodule k⟦X⟧ V) : sh a (sh b N) = sh (a + b) N

theorem **sh_le_sh** (a : ℤ) {N N' : Submodule k⟦X⟧ V} (h : N ≤ N') : sh a N ≤ sh a N'

theorem **sh_le_sh_of_le** {a b : ℤ} (h : b ≤ a) (N : Submodule k⟦X⟧ V) : sh a N ≤ sh b N

theorem **IsLat.shift** {N : Submodule k⟦X⟧ V} (h : IsLat N) (a : ℤ) : IsLat (sh a N)

theorem **pull_le_pull** [Module k V] (f : V →ₗ[k] V) {N N' : Submodule k⟦X⟧ V} (h : N ≤ N') : pull f N ≤ pull f N'

theorem **nab_apply_eq** [Module k V] [IsScalarTower k k⦅X⦆ V] (g : V →ₗ[k] V) (x : V) : nab g x = (vv k)⁻¹ • (g x - x)

theorem **condC_iff_nab_mem** [Module k V] [IsScalarTower k k⦅X⦆ V] [Module k⟦X⟧ V] [IsScalarTower k⟦X⟧ k⦅X⦆ V] (g : V →ₗ[k] V) (N : Submodule k⟦X⟧ V) : CondC g N ↔ ∀ x ∈ N, nab g x ∈ N

theorem **ModPG.span_φ** (D : ModPG R n V) (i : ZMod n) {ι : Type} (b : Module.Basis ι k⦅X⦆ V) : Submodule.span k⦅X⦆ (Set.range fun j => D.φ i (b j)) = ⊤

theorem **ModPG.linearIndependent_φ** (D : ModPG R n V) (i : ZMod n) {ι : Type} (b : Module.Basis ι k⦅X⦆ V) : LinearIndependent k⦅X⦆ fun j => D.φ i (b j)

theorem **summable_sg** {l n : ℕ} [NeZero n] (hl : 2 ≤ l) (h : ZMod n → ℕ) (i : ZMod n) : Summable fun j : ℕ => (h (i + ((j + 1 : ℕ) : ZMod n)) : ℝ) / (l : ℝ) ^ (j + 1)

theorem **extend_coe** (f : k⟦X⟧ →ₐ[k] k⟦X⟧) (hf : f PowerSeries.X ≠ 0) (a : k⟦X⟧) : extend f hf (a : k⦅X⦆) = ((f a : k⟦X⟧) : k⦅X⦆)

theorem **algHom_ext** {f g : k⦅X⦆ →ₐ[k] k⦅X⦆} (h : ∀ a : k⟦X⟧, f (a : k⦅X⦆) = g (a : k⦅X⦆)) (x : k⦅X⦆) : f x = g x

theorem **powerSeries_oneAddConstMulX_mul_inv** (c : k) : (1 + PowerSeries.C c ∗ PowerSeries.X) ∗ inv1 c = 1

theorem **hasSubst_mob** (c : k) : PowerSeries.HasSubst (PowerSeries.X ∗ inv1 c : k⟦X⟧)

theorem **mob_X** (c : k) : mob c PowerSeries.X = PowerSeries.X ∗ inv1 c

theorem **mob_C** (c d : k) : mob c (PowerSeries.C d) = PowerSeries.C d

theorem **mob_X_ne** (c : k) : mob c PowerSeries.X ≠ 0

theorem **mob_mob** (c : k) (f : k⟦X⟧) : mob c (mob (-c) f) = f

theorem **extend_mob_mob** (c : k) (x : k⦅X⦆) : extend (mob c) (mob_X_ne c) (extend (mob (-c)) (mob_X_ne (-c)) x) = x

theorem **gam_bijective** : Function.Bijective (gam k)

theorem **hasSubst_pow** (l : ℕ) [Fact l.Prime] : PowerSeries.HasSubst (PowerSeries.X ^ l : k⟦X⟧)

theorem **frob_X** (l : ℕ) [Fact l.Prime] : frob k l PowerSeries.X = PowerSeries.X ^ l

theorem **frob_X_ne** (l : ℕ) [Fact l.Prime] : frob k l PowerSeries.X ≠ 0

theorem **frob_mob** (l : ℕ) [Fact l.Prime] [CharP k l] (f : k⟦X⟧) : frob k l (mob 1 f) = mob 1 (frob k l f)

@[simp] theorem **lin_apply** (σ : k⦅X⦆ →ₐ[k] k⦅X⦆) (x : k⦅X⦆) : lin σ x = σ x

theorem **mem_stdLat** {x : k⦅X⦆} : x ∈ stdLat k ↔ ∃ a : k⟦X⟧, (a : k⦅X⦆) = x

theorem **isLat_stdLat** : IsLat (stdLat k)

theorem **pull_stdLat** {l : ℕ} (R : RingPG k l) : pull (lin R.φ) (stdLat k) = stdLat k

theorem **condAB_stdLat** {l : ℕ} (R : RingPG k l) (hγ : Function.Bijective R.γ) (n : ℕ) : CondAB (unitMod R hγ n).φ (fun _ => 0) (fun _ => 0) (fun _ => stdLat k)

theorem **condC_stdLat** {l : ℕ} (R : RingPG k l) : CondC (lin R.γ) (stdLat k)

theorem **nab_vv** {l : ℕ} (R : RingPG k l) : ∃ κ : k⟦X⟧, PowerSeries.constantCoeff κ = 1 ∧ nab (lin R.γ) (vv k) = -((κ : k⸨X⸩)) * vv k)

theorem **nab_mem_stdLat** {l : ℕ} (R : RingPG k l) {x : k⸨X⸩} (hx : x ∈ stdLat k) : nab (lin R.γ) x ∈ sh 1 (stdLat k)

theorem **sen_stdLat_zero** {l : ℕ} (R : RingPG k l) : sen (lin R.γ) (stdLat k) 0 = 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.PowerSeriesVecReduceModUMulVec

theorem **mem_vecs** {B : Subalgebra O K⟦X⟧} {v : ι → K⟦X⟧} : v ∈ vecs ι B ↔ ∀ i, v i ∈ B

theorem **mem_tot** {B : Subalgebra O K⟦X⟧} {P : ZMod a → Submodule B (ι → K⟦X⟧)} {v : Vec K a ι} : v ∈ tot P ↔ ∀ τ, v τ ∈ P τ

@[simp] theorem **powerSeriesVecReduceModU_apply** (v : ι → K⟦X⟧) (i : ι) : red0 v i = constantCoeff (v i)

@[simp] theorem **red_apply** (v : Vec K a ι) (τ : ZMod a) : red v τ = red0 (v τ)

@[simp] theorem **frobV_apply** (l : ℕ) [Fact l.Prime] (v : ι → K⟦X⟧) (i : ι) : frobV l v i = frob K l (v i)

@[simp] theorem **frobTot_apply** (l : ℕ) [Fact l.Prime] (f : ZMod a → K⟦X⟧) (τ : ZMod a) : frobTot K a l f τ = frob K l (f (τ - 1))

theorem **powerSeriesVecReduceModU_mulVec** {κ : Type} [Fintype ι] (Y : Matrix κ ι K⟦X⟧) (v : ι → K⟦X⟧) : red0 (Y *ᵥ v) = Y.map constantCoeff *ᵥ red0 v

theorem **mx_mem** (τ : ZMod a) (i j : ι) : P.mx τ i j ∈ invE l e (A O K)

theorem **mxInv_mem** (τ : ZMod a) (i j : ι) : P.mxInv τ i j ∈ invE l e (A O K)

theorem **mx_mul_mxInv** (τ : ZMod a) : P.mx τ * P.mxInv τ = 1

theorem **mxInv_mul_mx** (τ : ZMod a) : P.mxInv τ * P.mx τ = 1

@[simp] theorem **mx_ofMatrix** (Y Y' : ZMod a → Matrix ι ι K⟦X⟧) (hY : ∀ τ i j, Y τ i j ∈ invE l e (A O K)) (hY' : ∀ τ i j, Y' τ i j ∈ invE l e (A O K)) (h : ∀ τ, Y τ * Y' τ = 1) (τ : ZMod a) : (ofMatrix Y Y' hY hY' h).mx τ = Y τ

theorem **mem_extc** {B : Subalgebra O K⟦X⟧} {τ : ZMod a} {v : ι → K⟦X⟧} : v ∈ P.extc B τ ↔ ∃ m ∈ vecs ι B, v = P.mx τ *ᵥ m

theorem **mem_extc_iff** {B : Subalgebra O K⟦X⟧} {τ : ZMod a} {v : ι → K⟦X⟧} : v ∈ P.extc B τ ↔ P.mxInv τ *ᵥ v ∈ vecs ι B

theorem **linearIndependent_col** (B : Subalgebra O K⟦X⟧) (τ : ZMod a) : LinearIndependent B (P.mx τ).col

@[simp] theorem **coe_basisExt** (B : Subalgebra O K⟦X⟧) (τ : ZMod a) (i : ι) : (P.basisExt B τ i : ι → K⟦X⟧) = (P.mx τ).col i

theorem **extc_eq_span** {B : Subalgebra O K⟦X⟧} (h : A O K ≤ B) (τ : ZMod a) : P.extc B τ = Submodule.span B (P.Lcomp τ : Set (ι → K⟦X⟧))

theorem **exists_exponents** [NeZero a] : ∃ α β : ℕ, (∀ v ∈ P.L, E K l e ^ α • v ∈ P.M) ∧ ∀ v ∈ P.M, E K l e ^ β • v ∈ P.L

theorem **mem_uL_iff** {τ : ZMod a} {v : P.Lcomp τ} : v ∈ P.uL τ ↔ red0 (v : ι → K⟦X⟧) = 0

theorem **redQ_injective** (τ : ZMod a) : Function.Injective (P.redQ τ)

theorem **matrix_mul_inv_const_eq_one** (τ : ZMod a) : P.mx0 τ * (P.mxInv τ).map constantCoeff = 1

theorem **inv_const_mul_matrix_eq_one** (τ : ZMod a) : (P.mxInv τ).map constantCoeff * P.mx0 τ = 1

theorem **phiD_apply** (d : P.D) (τ : ZMod a) : P.phiD d τ = P.mx0 τ *ᵥ d (τ - 1)

theorem **phi_apply** (v : Vec K a ι) (τ : ZMod a) : P.phi v τ = P.mx τ *ᵥ frobV l (v (τ - 1))

theorem **phi_smul_tot** (f : ZMod a → K⟦X⟧) (v : Vec K a ι) : P.phi (f • v) = frobTot K a l f • P.phi v

theorem **phi_smul** (f : K⟦X⟧) (v : Vec K a ι) : P.phi (f • v) = frob K l f • P.phi v

theorem **phi_injective** : Function.Injective P.phi

theorem **phi_mem_ext** {B : Subalgebra O K⟦X⟧} (hB : ∀ f ∈ B, frob K l f ∈ B) {v : Vec K a ι} (hv : v ∈ P.Mext B) : P.phi v ∈ P.ext B

theorem **phi_mem_L** {v : Vec K a ι} (hv : v ∈ P.M) : P.phi v ∈ P.L

theorem **L_eq_span** [NeZero a] : P.L = Submodule.span (A O K) (P.phi '' P.M)

theorem **red_phi** (v : Vec K a ι) : red (P.phi v) = P.phiD (red v)

theorem **isLocalizedModule_redQ** (τ : ZMod a) : IsLocalizedModule (Submonoid.powers (l : O)) (P.redQ τ) where

theorem **app_apply** (F : Hom P Q) (v : Vec K a ι) (τ : ZMod a) : F.app v τ = F.mat τ *ᵥ v τ

theorem **red_app** (F : Hom P Q) (v : Vec K a ι) : red (F.app v) = F.onD (red v)

theorem **app_phi** (F : Hom P Q) (v : Vec K a ι) : F.app (P.phi v) = Q.phi (F.app v)

theorem **app_mem_Mext** (F : Hom P Q) {B : Subalgebra O K⟦X⟧} (hB : A O K ≤ B) {v : Vec K a ι} (hv : v ∈ P.Mext B) : F.app v ∈ Q.Mext B

theorem **app_mem_M** (F : Hom P Q) {v : Vec K a ι} (hv : v ∈ P.M) : F.app v ∈ Q.M

theorem **app_mem_ext** (F : Hom P Q) {B : Subalgebra O K⟦X⟧} (hB : A O K ≤ B) {v : Vec K a ι} (hv : v ∈ P.ext B) : F.app v ∈ Q.ext B

theorem **app_scalar** (c : O) (v : Vec K a ι) : (scalar P c).app v = algebraMap O K⟦X⟧ c • v

theorem **app_comp** (G : Hom Q T) (F : Hom P Q) (v : Vec K a ι) : (G.comp F).app v = G.app (F.app v)

theorem **hat_of_mem** {f : K⟦X⟧} (h : f ∈ RpiE O K l e) : hat O l e f = toHatE O K l e ⟨f, h⟩

@[simp] theorem **hat_coe** (f : RpiE O K l e) : hat O l e (f : K⟦X⟧) = toHatE O K l e f

theorem **hat_coe_Rpi** (f : Rpi O K l e) : hat O l e (f : K⟦X⟧) = algebraMap _ _ (toHat O K l e f)

theorem **hat_coe_A** (f : A O K) : hat O l e (f : K⟦X⟧) = algebraMap _ _ (ofA O K l e f)

theorem **hat_add** {f g : K⟦X⟧} (hf : f ∈ RpiE O K l e) (hg : g ∈ RpiE O K l e) : hat O l e (f + g) = hat O l e f + hat O l e g

theorem **hat_mul** {f g : K⟦X⟧} (hf : f ∈ RpiE O K l e) (hg : g ∈ RpiE O K l e) : hat O l e (f * g) = hat O l e f * hat O l e g

theorem **hat_E** : hat O l e (E K l e) = algebraMap _ _ (Ehat O K l e)

theorem **invE_A_le_RpiE** : invE l e (A O K) ≤ RpiE O K l e

theorem **invE_invL_S_le_RpiE** : invE l e (invL l (S O K e)) ≤ RpiE O K l e

theorem **mem_Mhatc** {w : ι → AhatE O K l e} : w ∈ Mhatc O K l e ↔ ∀ i, ∃ x : Ahat O K l e, algebraMap _ _ x = w i

theorem **hatSpan_extc** {B : Subalgebra O K⟦X⟧} (hB : B ≤ Rpi O K l e) (τ : ZMod a) : hatSpan O l e (P.extc B τ) = Submodule.span _ (Set.range fun j => hatV O l e ((P.mx τ).col j))

@[simp] theorem **coe_Eunit** : ((Eunit O K l e : invE l e (A O K)) : K⟦X⟧) = E K l e

theorem **mx_rankOne** {j : ZMod a → ℤ} {τ : ZMod a} {n : ℕ} (h : j τ = n) : (rankOne O K l e j).mx τ () () = E K l e ^ n

theorem **mxInv_rankOne** {j : ZMod a → ℤ} {τ : ZMod a} {n : ℕ} (h : j τ = -n) : (rankOne O K l e j).mxInv τ () () = E K l e ^ n

theorem **mem_Lcomp_rankOne** {j : ZMod a → ℤ} {τ : ZMod a} {n : ℕ} (h : j τ = n) {v : Unit → K⟦X⟧} : v ∈ (rankOne O K l e j).Lcomp τ ↔ ∃ m ∈ A O K, v () = E K l e ^ n * m

theorem **phiD_rankOne** {j : ZMod a → ℤ} {τ : ZMod a} {n : ℕ} (h : j τ = n) (d : (rankOne O K l e j).D) : (rankOne O K l e j).phiD d τ () = (l : K) ^ n * d (τ - 1) ()

theorem **one_notMem_Lcomp_rankOne** (τ : ZMod a) : (fun _ => 1) ∉ (rankOne O K l e fun _ : ZMod a => 1).Lcomp τ

theorem **nontrivial_Ahat** [Fact l.Prime] [NormedAlgebra ℚ_[l] K] [NeZero e] : Nontrivial (Ahat O K l e)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.QuotIntegrableIffCutoff

theorem **apply_out_mk** {T : Subgroup G} {α : Type} {F : G → α} (hF : ∀ t ∈ T, ∀ x, F (t * x) = F x) (x : G) : F (Quotient.out (Quotient.mk (QuotientGroup.rightRel T) x)) = F x

theorem **measurable_comp_out** {T : Subgroup G} {α : Type} [MeasurableSpace α] {F : G → α} (hm : Measurable F) (hF : ∀ t ∈ T, ∀ x, F (t * x) = F x) : Measurable fun c : RCoset T => F (Quotient.out c)

theorem **exists_isCosetCutoff** [LocallyCompactSpace G] [T2Space G] (hT : IsClosed (T : Set G)) [μT.IsHaarMeasure] [μT.IsInvInvariant] : ∃ β : G → NNReal, Continuous β ∧ IsCosetCutoff T μT β

theorem **IsCosetCutoff.isQuotMeasure** [SFinite μ] [μ.IsMulLeftInvariant] [SFinite μT] [μT.IsMulLeftInvariant] [μT.IsInvInvariant] {β : G → NNReal} (hβ : IsCosetCutoff T μT β) : IsQuotMeasure T μ μT ((μ.withDensity fun x => β x).map (Quotient.mk (QuotientGroup.rightRel T)))

theorem **IsQuotMeasure.eq_map** {ν : Measure (RCoset T)} (hν : IsQuotMeasure T μ μT ν) {β : G → NNReal} (hβ : IsCosetCutoff T μT β) : ν = (μ.withDensity fun x => β x).map (Quotient.mk (QuotientGroup.rightRel T))

theorem **isQuotMeasure_quotMeas** (hT : IsClosed (T : Set G)) : IsQuotMeasure T μ μT (quotMeas T μ μT)

theorem **IsQuotMeasure.eq_quotMeas** (hT : IsClosed (T : Set G)) {ν : Measure (RCoset T)} (hν : IsQuotMeasure T μ μT ν) : ν = quotMeas T μ μT

theorem **quotInt_eq_cutoff** (hT : IsClosed (T : Set G)) {β : G → NNReal} (hβ : IsCosetCutoff T μT β) {F : G → ℂ} (hm : Measurable F) (hF : ∀ t ∈ T, ∀ x, F (t * x) = F x) : quotInt T μ μT F = ∫ x, (β x : ℝ) • F x ∂μ

theorem **quotIntegrable_iff_cutoff** (hT : IsClosed (T : Set G)) {β : G → NNReal} (hβ : IsCosetCutoff T μT β) {F : G → ℂ} (hm : Measurable F) (hF : ∀ t ∈ T, ∀ x, F (t * x) = F x) : QuotIntegrable T μ μT F ↔ Integrable (fun x => (β x : ℝ) • F x) μ

theorem **quotInt_of_isCompact** (hT : IsCompact (T : Set G)) {F : G → ℂ} (hm : Measurable F) (hF : ∀ t ∈ T, ∀ x, F (t * x) = F x) : quotInt T μ μT F = ((μT.real Set.univ : ℝ) : ℂ)⁻¹ * ∫ x, F x ∂μ

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.RealisesUnitPow

theorem **semiBC_of_basis** {A B M N J : Type} [CommRing A] [CommRing B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module B N] {f : A →+* B} {u : M →ₛₗ[f] N} (b : Module.Basis J A M) (c : Module.Basis J B N) (h : ∀ i, u (b i) = c i) : SemiBC u

theorem **piMap_single** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) {J : Type} [DecidableEq J] (i : J) : piMap f J (Pi.single i 1) = Pi.single i 1

theorem **semiBC_piMap** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (J : Type) [Fintype J] [DecidableEq J] : SemiBC (piMap f J)

theorem **exists_of_isFrac** {A M : Type} [CommRing A] [AddCommGroup M] [Module A M] {d : A} {y m : M} {k : ℕ} (h : IsFrac d (LocalizedModule.mk y 1) m k) : ∃ r : ℕ, d ^ r • d ^ k • y = d ^ r • m

theorem **unitPowMap_apply** (F : FinExt l) {T J : Type} [AddCommGroup T] [Module ℤ_[l] T] [Fintype J] [DecidableEq J] (β : Module.Basis J ℤ_[l] T) (x : J → Ainf l) : unitPowMap F β x = ∑ i, inclW l (x i) ⊗ₜ[ℤ_[l]] β i

theorem **sum_frob** {T J : Type} [AddCommGroup T] [Module ℤ_[l] T] [Fintype J] (β : J → T) {d : Ainf l} (hd : d ≠ 0) (x m : J → Ainf l) (k r : ℕ) (h : ∀ i, d ^ r * (d ^ k * WittVector.frobenius (x i)) = d ^ r * m i) : inclW l d ^ k • TensorProduct.map (frobW l).toLinearMap LinearMap.id (∑ i, inclW l (x i) ⊗ₜ[ℤ_[l]] β i) = ∑ i, inclW l (m i) ⊗ₜ[ℤ_[l]] β i

theorem **sum_gal** {T J : Type} [AddCommGroup T] [Module ℤ_[l] T] [Fintype J] (β : J → T) (g : GQ l) (ρ : T →ₗ[ℤ_[l]] T) (hρ : ∀ x, ρ x = x) (x : J → Ainf l) : ∑ i, inclW l (Per.σA g (x i)) ⊗ₜ[ℤ_[l]] β i = TensorProduct.map (galW l g).toLinearMap ρ (∑ i, inclW l (x i) ⊗ₜ[ℤ_[l]] β i)

theorem **unitPowMap_frob** (F : FinExt l) {T J : Type} [AddCommGroup T] [Module ℤ_[l] T] [Fintype J] [DecidableEq J] (β : Module.Basis J ℤ_[l] T) (d : Ainf l) (hd : d ≠ 0) (x m : J → Ainf l) (k : ℕ) (h : IsFrac d (LocalizedModule.mk (piMap WittVector.frobenius J x) 1) m k) : inclW l d ^ k • TensorProduct.map (frobW l).toLinearMap LinearMap.id (unitPowMap F β x) = unitPowMap F β m

theorem **unitPowMap_gal** (F : FinExt l) {T J : Type} [AddCommGroup T] [Module ℤ_[l] T] [Fintype J] [DecidableEq J] (β : Module.Basis J ℤ_[l] T) (g : GQ l) (ρ : T →ₗ[ℤ_[l]] T) (hρ : ∀ x, ρ x = x) (x : J → Ainf l) : unitPowMap F β (fun i => Per.σA g (x i)) = TensorProduct.map (galW l g).toLinearMap ρ (unitPowMap F β x)

theorem **realises_unitPow** (F : FinExt l) {T J : Type} [AddCommGroup T] [Module ℤ_[l] T] [Fintype J] [DecidableEq J] (β : Module.Basis J ℤ_[l] T) (ρ : ɪ(GK F) → (T →ₗ[ℤ_[l]] T)) (hρ : ∀ g x, ρ g x = x) : Realises F (unitPow l ɪ(OL F) J) (ainfObj F) (ainfId F) T ρ (unitPowMap F β) where

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.SignAtBasePlaceEqEmbedding

theorem **tensMap_tmul** (r : R →+* R') (κ : K →+ K') (φ : M →+ M') (hκ hφ) (a : K) (x : M) : tensMap r κ φ hκ hφ (a ⊗ₜ x) = κ a ⊗ₜ φ x

theorem **tens_ext** {P : Type} [AddCommGroup P] {f g : K ⊗[R] M →+ P} (h : ∀ a x, f (a ⊗ₜ x) = g (a ⊗ₜ x)) : f = g

theorem **symm_smul** {r : R ≃+* R'} {φ : M ≃+ M'} (hφ : ∀ c x, φ (c • x) = r c • φ x) (c : R') (x : M') : φ.symm (c • x) = r.symm c • φ.symm x

theorem **tensEquiv_tmul** (r : R ≃+* R') (κ : K ≃+ K') (φ : M ≃+ M') (hκ hφ) (a : K) (x : M) : tensEquiv r κ φ hκ hφ (a ⊗ₜ x) = κ a ⊗ₜ φ x

theorem **map_mem_of_span** {f : M →+ M'} {r : R → R'} (hf : ∀ c x, f (c • x) = r c • f x) {s : Set M} {B : Submodule R' M'} (hs : ∀ x ∈ s, f x ∈ B) {x : M} (hx : x ∈ Submodule.span R s) : f x ∈ B

theorem **stab_iff** (q : KsIso ι eK γ A A') : (∀ g x, x ∈ A.WK → A.ρ g x ∈ A.WK) ↔ ∀ g x, x ∈ A'.WK → A'.ρ g x ∈ A'.WK

theorem **KsIso.ofFormsIso_apply** {A : Arith.FormsL LL Ks WL W G} {A' : Arith.FormsL LL Ks' WL' W' G'} (f : NX.FormsIso A A') (x : A.WK) : ((KsIso.ofFormsIso f).φ x : WL') = f.e x

theorem **KsIso.bc_apply** (A : Arith.FormsL L0 Ks WL W G) (x : A.WK) : ((KsIso.bc L A).φ x : NX.BC.BCW Ks L0 L WL) = NX.BC.incl Ks L0 L WL x

theorem **KsIso.bc_j** (A : Arith.FormsL L0 Ks WL W G) (σ : L →+* ℂ) (x : A.WK) : (NX.BC.bc L A).j σ ((KsIso.bc L A).φ x : NX.BC.BCW Ks L0 L WL) = A.j (σ.comp (algebraMap L0 L)) x

theorem **Scal.read'** (s : Scal ι e i i' b κ) {x : ɪLL0} {x' : ɪLL0'} (h : (x' : rows'.LL) = ι (x : what₂.LL)) : fracEquiv κ (NX.BC.rdTw tw i b x) = NX.BC.rdTw tw' i' b x'

theorem **Scal.place** (s : Scal ι e i i' b κ) (q : KsIso ι eK γ (S.A j) (S'.A j')) (c : S.Ks) : fracEquiv κ (NX.ιK S i b c) = NX.ιK S' i' b (eK c)

theorem **fmOf_tmul** (a : NX.KB S i b) (x : NX.FmK S j) : fmOf s q (a ⊗ₜ[S.Ks] x : NX.KB S i b ⊗[S.Ks] NX.FmK S j) = ((show NX.KB S' i' b from fracEquiv κ a) ⊗ₜ[S'.Ks] (show NX.FmK S' j' from q.φ x) : NX.KB S' i' b ⊗[S'.Ks] NX.FmK S' j')

theorem **fmOf_smul** (c : FractionRing (tw.O i)) (x : NX.Fm S i j b) : fmOf s q (c • x) = fracEquiv κ c • fmOf s q x

theorem **fmOf_smulO** (c : tw.O i) (x : NX.Fm S i j b) : fmOf s q (c • x) = κ c • fmOf s q x

theorem **fmOf_mem_Bsup** (x : NX.Fm S i j b) : fmOf s q x ∈ NX.Bsup S' i' j' b ↔ x ∈ NX.Bsup S i j b

theorem **KsIso.rhoStab_iff** : NX.RhoStab S j ↔ NX.RhoStab S' j'

theorem **KsIso.chiInLL0** (he : ∀ x : ɪLL0', ι (e x : what₂.LL) = (x : rows'.LL)) (h : NX.ChiInLL0 S j) : NX.ChiInLL0 S' j'

theorem **fmOf_baseChange** (u : Module.End S.Ks (NX.FmK S j)) (u' : Module.End S'.Ks (NX.FmK S' j')) (hu : ∀ y : NX.FmK S j, q.φ (u y) = u' (q.φ y)) (x : NX.KB S i b ⊗[S.Ks] NX.FmK S j) : fmOf s q (Module.End.baseChangeHom S.Ks (NX.KB S i b) (NX.FmK S j) u x) = Module.End.baseChangeHom S'.Ks (NX.KB S' i' b) (NX.FmK S' j') u' (fmOf s q x)

theorem **fmOf_actG** (hst : NX.RhoStab S j) (hχ : NX.ChiInLL0 S j) (hst' : NX.RhoStab S' j') (hχ' : NX.ChiInLL0 S' j') (g : G0g N) (g' : G0g N') (hg : S'.eG.symm g' = γ (S.eG.symm g)) (x : NX.Fm S i j b) : fmOf s q (NX.actG S i j b hst hχ g x) = NX.actG S' i' j' b hst' hχ' g' (fmOf s q x)

theorem **fmOf_restrict** (u : Module.End what₂.LL (S.WL j)) (u' : Module.End rows'.LL (S'.WL j')) (hs : ∀ x ∈ (S.A j).WK, u x ∈ (S.A j).WK) (hs' : ∀ x ∈ (S'.A j').WK, u' x ∈ (S'.A j').WK) (hu : ∀ y : (S.A j).WK, (q.φ (u y, hs _ y.2) : S'.WL j') = u' (q.φ y)) (x : NX.KB S i b ⊗[S.Ks] NX.FmK S j) : fmOf s q (Module.End.baseChangeHom S.Ks (NX.KB S i b) (NX.FmK S j) ((u.restrictScalars S.Ks).restrict (p

theorem **towKsBC** : letI

theorem **ratDatBC_Ks** : (ratDatBC tw' S ι e he c).Ks = S.Ks

theorem **ratDatBC_G** : (ratDatBC tw' S ι e he c).G = S.G

theorem **ratDatBC_eG** (g : S.G) : (ratDatBC tw' S ι e he c).eG g = c (S.eG g)

theorem **ratDatBC_kKs** (x : S.Ks) : (ratDatBC tw' S ι e he c).kKs x = e.symm (S.kKs x)

theorem **ratDatBC_A** (j : ℕ) : letI

theorem **ratDatBC_rhoStab_iff** (j : ℕ) : NX.RhoStab S j ↔ NX.RhoStab (ratDatBC tw' S ι e he c) j

theorem **ratDatBC_chiInLL0** (j : ℕ) (h : NX.ChiInLL0 S j) : NX.ChiInLL0 (ratDatBC tw' S ι e he c) j

theorem **ksBC_j** (j : ℕ) (σ : rows'.LL →+* ℂ) (x : (S.A j).WK) : ((ratDatBC tw' S ι e he c).A j).j σ ((ksBC tw' S ι e he c j).φ x : (ratDatBC tw' S ι e he c).WL j) = (S.A j).j (σ.comp ι) x

theorem **ringCast_heq** {A B : Type} [iA : CommRing A] [iB : CommRing B] (h : A = B) (hi : HEq iA iB) (x : A) : HEq (ringCast h hi x) x

theorem **fracEquiv_ringCast_heq** {A B : Type} [iA : CommRing A] [iB : CommRing B] (h : A = B) (hi : HEq iA iB) (x : FractionRing A) : HEq (fracEquiv (ringCast h hi) x) x

theorem **heq_ring** {T T' : Type} {O : T → Type} {O' : T' → Type} {c : ∀ i, CommRing (O i)} {c' : ∀ i, CommRing (O' i)} (hT : T = T') (hO : HEq O O') (hc : HEq c c') {i : T} {i' : T'} (hi : HEq i i') : O i = O' i' ∧ HEq (c i) (c' i')

theorem **heq_fracMap** {T T' : Type} {A O : T → Type} {A' O' : T' → Type} [a : ∀ i, CommRing (A i)] [a' : ∀ i, CommRing (A' i)] [c : ∀ i, CommRing (O i)] [c' : ∀ i, CommRing (O' i)] [∀ i, IsDomain (A i)] [∀ i, IsDomain (A' i)] [∀ i, IsDomain (O i)] [∀ i, IsDomain (O' i)] [g : ∀ i, Algebra (A i) (O i)] [g' : ∀ i, Algebra (A' i) (O' i)] (hT : T = T') (hA : HEq A A') (ha : HEq a a') (hO : HEq O O') (hc : HEq c c') (hg : HEq g g') {i : T} {i' : T'} (hi : HEq i i') (n : Function.Injective (algebraMap (A i) (O i))) (n' : Function.Injective (algebraMap (A' i') (O' i'))) {x : FractionRing (A i)} {x' : FractionRing (A' i')} (hx : HEq x x') : HEq (IsFractionRing.map (K

theorem **heq_fam** {T T' : Type} {K : T → Type} {K' : T' → Type} {k : ∀ i, CommRing (K i)} {k' : ∀ i, CommRing (K' i)} {M : T → ℕ → Bool → Type} {M' : T' → ℕ → Bool → Type} {a : ∀ i j b, AddCommGroup (M i j b)} {a' : ∀ i j b, AddCommGroup (M' i j b)} {m : ∀ i j b, Module (FractionRing (K i)) (M i j b)} {m' : ∀ i j b, Module (FractionRing (K' i)) (M' i j b)} (hT : T = T') (hK : HEq K K') (hk : HEq k k') (hM : HEq M M') (ha : HEq a a') (hm : HEq m m') {i : T} {i' : T'} (hi : HEq i i') (j : ℕ) (b : Bool) : M i j b = M' i' j b ∧ HEq (a i j b) (a' i' j b) ∧ ∀ (c : FractionRing (K i)) (c' : FractionRing (K' i')) (x : M i j b) (x' : M' i' j b), HEq c c' → HEq x x' → HEq (c • x) (c' • x')

theorem **heq_grp** {R R' : Ranks} {H : Host R} {H' : Host R'} {P P' : Fin 2 → Type} (hR : R = R') (hH : HEq H H') (hP : P = P') : (∀ kv : (Σ k : Fin 2, P k), GL (Fin R.r) (ZMod (H.g kv.1))) = (∀ kv : (Σ k : Fin 2, P' k), GL (Fin R'.r) (ZMod (H'.g kv.1))) ∧ HEq (inferInstance : Mul (∀ kv : (Σ k : Fin 2, P k), GL (Fin R.r) (ZMod (H.g kv.1)))) (inferInstance : Mul (∀ kv : (Σ k : Fin 2, P' k), GL (Fin R'.r) (ZMod (H'.g kv.1))))

theorem **refl** (D : D7) : CopyTw D D (RingHom.id _) (RingEquiv.refl _) where

theorem **Tw_eq** : D.ntw.Tw = D'.ntw.Tw

theorem **instTau** : HEq D.what₂.instTau D'.what₂.instTau

theorem **freeB_finB** : HEq D.ntw.freeB D'.ntw.freeB ∧ HEq D.ntw.finB D'.ntw.finB

theorem **embLL_comp** : D'.what₂.embLL.comp ι = D.what₂.embLL

theorem **embLL0_eq** (x : ɪD'.LL0) : embLL0 D (e x) = embLL0 D' x

theorem **map_star** (x : D.what₂.LL) : ι (star x) = star (ι x)

theorem **map_embE** {x : D.host.E} {x' : D'.host.E} (hx : HEq x x') : ι (D.embE x) = D'.embE x'

theorem **grpEquiv_heq** (g : G0g D.nd) : HEq (h.grpEquiv g) g

theorem **grpEquiv_eq_iff** {g : G0g D.nd} {g' : G0g D'.nd} : h.grpEquiv g = g' ↔ HEq g g'

theorem **oEquiv_heq** (x : D.ntw.O i) : HEq (h.oEquiv hi x) x

theorem **frac_heq** (x : FractionRing (D.ntw.O i)) : HEq (fracEquiv (h.oEquiv hi) x) x

theorem **sideEmb_heq** (b : Bool) (x : ɪD'.LL0) : HEq (sideEmb D.ntw i b (e x)) (sideEmb D'.ntw i' b x)

theorem **scal** (b : Bool) : Scal ι e i i' b (h.oEquiv hi)

theorem **fmCast_heq** (j : ℕ) (b : Bool) (x : D.nexp.Fm i j b) : HEq (h.fmCast hi j b x) x

theorem **fmCast_smul** (j : ℕ) (b : Bool) (c : FractionRing (D.ntw.O i)) (x : D.nexp.Fm i j b) : h.fmCast hi j b (c • x) = fracEquiv (h.oEquiv hi) c • h.fmCast hi j b x

theorem **eFMove_fmCast** {S : NX.RatDat D.ntw} {S' : NX.RatDat D'.ntw} {eK : S.Ks ≃+* S'.Ks} {γ : S.G ≃* S'.G} {j : ℕ} {b : Bool} (q : KsIso ι eK γ (S.A j) (S'.A j)) (eF : D.nexp.Fm i j b ≃ₗ[FractionRing (D.ntw.O i)] NX.Fm S i j b) (x : D.nexp.Fm i j b) : h.eFMove hi q eF (h.fmCast hi j b x) = fmOf (h.scal hi b) q (eF x)

theorem **fmMove_smul** (c : FractionRing (D.ntw.O i)) (x : NX.Fm S i j b) : fmMove h S hi j b (c • x) = fracEquiv (h.oEquiv hi) c • fmMove h S hi j b x

theorem **fmMove_mem_Bsup** (x : NX.Fm S i j b) : fmMove h S hi j b x ∈ NX.Bsup (h.ratDat S) i' j b ↔ x ∈ NX.Bsup S i j b

theorem **fmMove_actG** (hst : NX.RhoStab S j) (hχ : NX.ChiInLL0 S j) (hst' hχ') (g : G0g D.nd) (x : NX.Fm S i j b) : fmMove h S hi j b (NX.actG S i j b hst hχ g x) = NX.actG (h.ratDat S) i' j b hst' hχ' (h.grpEquiv g) (fmMove h S hi j b x)

theorem **SameModel.refl** (M : Mult.DSModel V P) : SameModel M M

theorem **SameVec.refl** {M : Mult.DSModel V P} (v : M.Vec) : SameVec v v

theorem **SameVec.eq** {M : Mult.DSModel V P} {v v' : M.Vec} (h : SameVec v v') : v = v'

theorem **SameMap.refl** {M : Mult.DSModel V P} (T : Mult.MMap M) : SameMap T T

theorem **SameMap.eq** {M : Mult.DSModel V P} {T T' : Mult.MMap M} (h : SameMap T T') : T = T'

theorem **exists_isCoef** (X : RealSide D) : ∃ c0L, IsCoef X c0L

theorem **sign_at_base_place_eq_embedding** (X : RealSide D) : sg0 D X.A3.A = D.what₂.embLL

theorem **TupExt.refl** (X : RealSide D) : TupExt X X

theorem **Sb** : HEq X.R.Sb X'.R.Sb

theorem **phs** (j : Fin D.R.m) (j' : Fin D'.R.m) : X.R.phs j = X'.R.phs j'

theorem **fixedEmbedding_comp** (X : RealSide D) (X' : RealSide D') : (sg0 D' X'.A3.A).comp ι = sg0 D X.A3.A

theorem **cG_eG** (g : X.Γ) : X'.cG (X'.R.eG (m.γ g)) = h.grpEquiv (X.cG (X.R.eG g))

theorem **kKsOf_spec** {S : NX.RatDat D.ntw} (YS : NX.NExpRealS X.R X.cG S) (j : Fin D.R.m) (x : X'.core.Q.c.F) : ((m.kKsOf (e

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.SmulMemRatForms

theorem **eKs_LL** (a : S.Ks) : algebraMap T0.Q.c.F D.what₂.LL (eKs YS a) = algebraMap S.Ks D.what₂.LL a

theorem **sideK_eKs** (i : D.ntw.Tw) (b : Bool) (a : S.Ks) : sideK YS i b (eKs YS a) = NX.ιK S i b a

theorem **e_cuspOf** (j : Fin D.R.m) (x : NX.FmK S ((j : ℕ) + 1)) : (fw.e j (cuspOf YS fw j x) : Fin (Rl.nW j) → D.what₂.LL) = (YS.fi j).e (Subtype.val x)

theorem **cuspOf_smul** (j : Fin D.R.m) (a : S.Ks) (x : NX.FmK S ((j : ℕ) + 1)) : cuspOf YS fw j (a • x) = eKs YS a • cuspOf YS fw j x

theorem **cuspOf_mem_sec** (j : Fin D.R.m) (x : NX.FmK S ((j : ℕ) + 1)) : (cuspOf YS fw j x : T0.R ((j : ℕ) + 1)) ∈ sec (T0.Q.Xo ((j : ℕ) + 1)) T0.Q.c (wtA (κ j)) (κ j).n

theorem **evW_eq_phiC** (j : Fin D.R.m) (hV : T0.V ((j : ℕ) + 1) (res T0 (sg0 D A3.A)) = Wc D j) (hev : fw.Evaluates) (x : NX.FmK S ((j : ℕ) + 1)) : GlobalTheory.SeriesTie.evW YS v₀ j (Subtype.val x) = phiC T0 hV (wtA (κ j)) (cuspOf YS fw j x : T0.R ((j : ℕ) + 1))

theorem **realRd_eq_phiC** {ρ : GlobalTheory.SeriesTie.Setup D.ntw} (hρ : ρ.Good) (i : D.ntw.Tw) (j : Fin D.R.m) (b : Bool) (ν : ℂ) (hV : T0.V ((j : ℕ) + 1) (res T0 (sg0 D A3.A)) = Wc D j) (hev : fw.Evaluates) (a : NX.KB S i b) (x : NX.FmK S ((j : ℕ) + 1)) : GlobalTheory.SeriesTie.realRd YS v₀ hρ i j b ν ((YS.eF i _ b (Nat.le_add_left _ _) j.2).symm (a ⊛ₜ[S.Ks] x : NX.KB S i b ⊛[S.Ks] _)) = (ν⁻¹ * ρ.e i b a) • phiC T0 hV (wtA (κ j)) (cuspOf YS fw j x : T0.R ((j : ℕ) + 1))

theorem **charAt_mem** (P : PlE D.host) : (charAt P : 𝒪 D.host.E) ∈ P.asIdeal

theorem **charAt_prime** (P : PlE D.host) : (charAt P).Prime

theorem **sideK_mem_kurAt** (a : T0.Q.c.F) : sideK YS i b a ∈ kurAt YS iC b

theorem **kurAt_eq** (h : ∀ x : S.Ks, ∃ y : S.Ks, algebraMap S.Ks D.what₂.LL y = star (algebraMap S.Ks D.what₂.LL x)) : kurAt YS iC true = kurAt YS iC false

theorem **valAt_tmul** (j : Fin D.R.m) (z : FrC YS iC b ((j : ℕ) + 1)) (a : NX.KB S i b) (x : NX.FmK S ((j : ℕ) + 1)) : valAt YS fw j z (a ⊛ₜ[S.Ks] x : NX.KB S i b ⊛[S.Ks] _) = iC a * valC z (cuspOf YS fw j x : T0.R ((j : ℕ) + 1))

theorem **valAt_smul** (j : Fin D.R.m) (z : FrC YS iC b ((j : ℕ) + 1)) (a : FractionRing (D.ntw.O i)) (x : NX.Fm S i ((j : ℕ) + 1) b) : valAt YS fw j z (a • x) = iC a * valAt YS fw j z x

theorem **valAt_frBor** (j : Fin D.R.m) {z z' : FrC YS iC b ((j : ℕ) + 1)} {t : Ix (T0.Q.Dw ((j : ℕ) + 1)) → (PAt (embC YS iC b))ˣ} (h : FrBor t z z') (x : NX.Fm S i ((j : ℕ) + 1) b) : valAt YS fw j z' x = wtChar (wtA (κ j)) (κ j).n t * valAt YS fw j z x

theorem **tmul_mem_ratForms** {j : ℕ} {L : Subfield (FractionRing (D.ntw.O i))} {a : NX.KB S i b} (ha : a ∈ L) (f : NX.FmK S j) : (a ⊛ₜ[S.Ks] f : NX.KB S i b ⊛[S.Ks] _) ∈ ratForms S j b L

theorem **smul_mem_ratForms** {j : ℕ} {L : Subfield (FractionRing (D.ntw.O i))} {a : FractionRing (D.ntw.O i)} (ha : a ∈ L) {x : NX.Fm S i j b} (hx : x ∈ ratForms S j b L) : a • x ∈ ratForms S j b L

theorem **span_ratForms** (j : ℕ) (L : Subfield (FractionRing (D.ntw.O i))) : Submodule.span (FractionRing (D.ntw.O i)) (ratForms S j b L : Set (NX.Fm S i j b)) = ⊤

theorem **ratForms_top** (j : ℕ) : ratForms S j b (⊤ : Subfield (FractionRing (D.ntw.O i))) = ⊤

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.SndDualLiftAlgebraMap

theorem **SameJets.bijective** (H : SameJets ι s) (m : ℕ) : Function.Bijective (jetMap ι s m)

theorem **SameJets.jet_eq_iff** (H : SameJets ι s) (m : ℕ) (x : B) (g : A') : H.jet m x = Ideal.Quotient.mk _ g ↔ ι s ^ m ∣ x - ι g

theorem **SameJets.jet_compat** (H : SameJets ι s) {m n : ℕ} (hle : m ≤ n) : (Ideal.Quotient.factorPow (Ideal.span {s}) hle).comp (H.jet n) = H.jet m

theorem **SameJets.evalₐ_toCompl_eq_iff** (H : SameJets ι s) (m : ℕ) (x : B) (g : A') : AdicCompletion.evalₐ _ m (H.toCompl x) = Ideal.Quotient.mk _ g ↔ ι s ^ m ∣ x - ι g

theorem **SameJets.toCompl_ι** (H : SameJets ι s) (g : A') : H.toCompl (ι g) = algebraMap A' _ g

theorem **derivation_mem_pow** (n : ℕ) {x : A'} (hx : x ∈ I ^ (n + 1)) : D x ∈ I ^ n

theorem **evalₐ_of_succ** {x : AdicCompletion I A'} {n : ℕ} {r : A'} (h : AdicCompletion.evalₐ I (n + 1) x = Ideal.Quotient.mk _ r) : AdicCompletion.evalₐ I n x = Ideal.Quotient.mk _ r

theorem **evalₐ_complDerFun** {x : AdicCompletion I A'} {n : ℕ} {r : A'} (h : AdicCompletion.evalₐ I (n + 1) x = Ideal.Quotient.mk _ r) : AdicCompletion.evalₐ I n (complDerFun I D x) = Ideal.Quotient.mk _ (D r)

theorem **complDerFun_mul** (x y : AdicCompletion I A') : complDerFun I D (x * y) = x * complDerFun I D y + y * complDerFun I D x

theorem **complDerFun_algebraMap** (r : A') : complDerFun I D (algebraMap A' _ r) = algebraMap A' _ (D r)

theorem **evalₐ_complDer** {x : AdicCompletion I A'} {n : ℕ} {r : A'} (h : AdicCompletion.evalₐ I (n + 1) x = Ideal.Quotient.mk _ r) : AdicCompletion.evalₐ I n (complDer I D x) = Ideal.Quotient.mk _ (D r)

theorem **complDer_algebraMap** (r : A') : complDer I D (algebraMap A' _ r) = algebraMap A' _ (D r)

theorem **isUnit_dualHom** (s : A') : IsUnit (dualHom D s s)

theorem **fst_dualLift** (s : A') (x : Localization.Away s) : (dualLift D s x).fst = x

theorem **snd_dualLift_mul** (s : A') (x y : Localization.Away s) : (dualLift D s (x * y)).snd = x * (dualLift D s y).snd + y * (dualLift D s x).snd

theorem **snd_dualLift_algebraMap** (s r : A') : (dualLift D s (algebraMap A' _ r)).snd = algebraMap A' _ (D r)

theorem **awayDer_algebraMap** (s r : A') : awayDer D s (algebraMap A' _ r) = algebraMap A' _ (D r)

theorem **mem_invL_A_iff** {f : K⟦X⟧} : f ∈ invL l (A O K) ↔ ∃ b : ℝ, ∀ n, ‖coeff n f‖ ≤ b

theorem **A_le_Rpi** : A O K ≤ Rpi O K l e

theorem **invL_A_le_Rpi** : invL l (A O K) ≤ Rpi O K l e

theorem **invL_S_le_Rpi** : invL l (S O K e) ≤ Rpi O K l e

theorem **coeff_E_mul** (g : K⟦X⟧) (n : ℕ) : coeff (n + e) (E K l e * g) = coeff n g + l * coeff (n + e) g

theorem **tendsto_add_mul** (n : ℕ) : Tendsto (fun j : ℕ => n + j * e) atTop atTop

theorem **norm_term_eq** (f : K⟦X⟧) (n j : ℕ) : ‖coeff (n + j * e) f * (-(l : K)) ^ j‖ * rad l (1 / e) ^ n = ‖coeff (n + j * e) f‖ * rad l (1 / e) ^ (n + j * e)

theorem **eq_zero_of_E_mul** {g : K⟦X⟧} (hg : g ∈ Rpi O K l e) (h : ∀ n, coeff (n + e) (E K l e * g) = 0) : g = 0

theorem **tail_rec** {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) (n : ℕ) : tail l e f n = coeff n f - l * tail l e f (n + e)

theorem **coeff_sub_E_mul_quotE** {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) (n : ℕ) : coeff (n + e) (f - E K l e * quotE l e f) = 0

theorem **quotE_mem_Rpi** {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) : quotE l e f ∈ Rpi O K l e

theorem **quotE_mem_invL_A** {f : K⟦X⟧} (hf : f ∈ invL l (A O K)) : quotE l e f ∈ invL l (A O K)

theorem **sub_E_mul_quotE_mem** {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) : f - E K l e * quotE l e f ∈ invL l (A O K)

theorem **mem_invL_A_of_E_mul** {h : K⟦X⟧} (hh : h ∈ Rpi O K l e) (hE : E K l e * h ∈ invL l (A O K)) : h ∈ invL l (A O K)

theorem **mem_invL_A_of_E_pow_mul** (m : ℕ) {h : K⟦X⟧} (hh : h ∈ Rpi O K l e) (hE : E K l e ^ m * h ∈ invL l (A O K)) : h ∈ invL l (A O K)

theorem **exists_div_E_pow** (m : ℕ) {f : K⟦X⟧} (hf : f ∈ Rpi O K l e) : ∃ g ∈ invL l (A O K), ∃ h ∈ Rpi O K l e, f = g + E K l e ^ m * h

theorem **sameJets** : SameJets (Subalgebra.inclusion (invL_A_le_Rpi (O

@[simp] theorem **coe_derAl** (f : invL l (A O K)) : (derAl O K l f : K⟦X⟧) = PowerSeries.derivative K f

theorem **dHatE_algebraMap** (x : Ahat O K l e) : dHatE O K l e (algebraMap _ _ x) = algebraMap _ _ (dHat O K l e x)

theorem **evalₐ_toHat_eq_iff** (m : ℕ) (f : Rpi O K l e) (g : invL l (A O K)) : AdicCompletion.evalₐ _ m (toHat O K l e f) = Ideal.Quotient.mk _ g ↔ ∃ h ∈ Rpi O K l e, (f : K⟦X⟧) - g = E K l e ^ m * h

theorem **toHat_eq_ofAl** {f : Rpi O K l e} {g : invL l (A O K)} (h : (f : K⟦X⟧) = g) : toHat O K l e f = ofAl O K l e g

theorem **toHat_E** : toHat O K l e (EIn l e _) = Ehat O K l e

theorem **toHatE_eq** {f : RpiE O K l e} {g : Rpi O K l e} (h : (f : K⟦X⟧) = g) : toHatE O K l e f = algebraMap _ _ (toHat O K l e g)

theorem **invE_le_RpiE** {B : Subalgebra O K⟦X⟧} [HasU B] (h : B ≤ Rpi O K l e) : invE l e B ≤ RpiE O K l e

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.SqEqOneOfFixDat

theorem **odd_r** (D : D7) : Odd D.R.r

theorem **inf_of_baseOf** {B : Base (Vc D i) (Wc D i)} (h : BaseOf R i B) (hL : R.toLevCh.Is A.μ0) (τ : D.what₂.Tau) : HasInfExp D B.mu0 τ ((B.w.c τ : ℚ) - ((B.w.p τ : ℚ) - B.w.q τ) / 2 + 1 / 2)

theorem **frame_good** {F : Face D A} (hC : GlobalTheory.Hk3.Core R F) {B : Base (Vc D i) (Wc D i)} (h : BaseOf R i B) : B.F.Good where

theorem **splitSet_Pg** {F : Face D A} (hC : GlobalTheory.Hk3.Core R F) (i : Fin D.R.m) (t : T0) : SplitSet (R.Pg ∪ R.Q t)

theorem **good_baseAt** {F : Face D A} (hC : GlobalTheory.Hk3.Core R F) (P : LevelP R.Pg (Wc D i).rank ℂ) (hu : UnramOutside (R.mu0 ((i : ℕ) + 1)) R.Pg) (hp : ∀ PP ∈ R.Pg, ∀ PP', PP' = PP ∨ Lev.IsConjPl PP PP' → FractionalIdeal.count D.host.E PP' D.host.AA = 0) : (baseAt R i hC.models.BV (Twist.triv Circle ∅) P).Good

theorem **ap_eq** {B : Base (Vc D i) (Wc D i)} (h : BaseOf R i B) (hB : B.w.Good) (τ : D.what₂.Tau) (a : Fin (Vc D i).rank) (ha : a.val < B.w.p τ) : B.w.ap τ a = (liftWeight D i).ap τ a

theorem **levS_chars** {B : Base (Vc D i) (Wc D i)} (h : BaseOf R i B) (hL : R.toLevCh.Is A.μ0) (hB : B.Good) {ψ : Twist D Circle} (hψ : ψ.Good) (hQ : B.admitted ψ.Q) {t : T0} (ht : IsTupOf R B ψ t) : (B.levS ψ).Pg = (R.datVt i t).Pg ∧ (B.levS ψ).Q = (R.datVt i t).Q ∧ (∀ c, ∀ PP ∈ R.Pg, (B.levS ψ).cg c PP = (R.datVt i t).cg c PP) ∧ ∀ PP ∈ R.Q t, (B.levS ψ).xq PP = (R.datVt i t).xq PP

theorem **not_baseOf_shift** {B : Base (Vc D i) (Wc D i)} (h : BaseOf R i B) (τ : D.what₂.Tau) : ¬ BaseOf R i { B with w

theorem **sq_eq_one_of_fixDat** {S T : HSp D} {B : Base S T} (h : FixDat B (starRingAut : ℂ ≃+* ℂ)) (x : (AE D)ˣ) : B.de.chi x * B.de.chi x = 1

theorem **chi_unique** {B : Base (Vc D i) (Wc D i)} {ψ ψ' : Twist D Circle} {t : T0} (h : IsTupOf R B ψ t) (h' : IsTupOf R B ψ' t) : ψ.chi = ψ'.chi

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.StepJumpDirect

theorem **steps_iff** : Steps ↔ StepCut ∧ StepsNoCut

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.SumTrWEqZeroOfFrobCoset

theorem **exists_pow_frob_comm** (ρ : Representation ℂ (Weil h) V) (hρ : IsSmooth ρ) : ∃ m : ℕ, 0 < m ∧ ∀ σ, Commute (ρ σ) (ρ (frob h) ^ m)

theorem **isSemisimple_apply** (ρ : Representation ℂ (Weil h) V) (hρ : IsSmooth ρ) (hss : ρ.IsSemisimpleRepresentation) (σ : Weil h) : End.IsSemisimple (ρ σ)

@[simp] theorem **wdOfRep_ρ** (ρ : Representation ℂ (Weil h) V) (hρ : IsSmooth ρ) (hss : ρ.IsSemisimpleRepresentation) : (wdOfRep ρ hρ hss).ρ = ρ

@[simp] theorem **wdOfRep_N** (ρ : Representation ℂ (Weil h) V) (hρ : IsSmooth ρ) (hss : ρ.IsSemisimpleRepresentation) : (wdOfRep ρ hρ hss).N = 0

theorem **trace_eq_of_iso** {V' : Type} [AddCommGroup V'] [Module ℂ V'] [FiniteDimensional ℂ V'] {a : WDRep h V} {b : WDRep h V'} (hab : a.Iso b) (σ : Weil h) : LinearMap.trace ℂ V (a.ρ σ) = LinearMap.trace ℂ V' (b.ρ σ)

@[simp] theorem **trW_mk** (a : WDRep h V) (σ : Weil h) : trW (WD.mk a) σ = LinearMap.trace ℂ V (a.ρ σ)

theorem **trW_one** (x : WD h) : trW x 1 = x.lpDim

@[simp] theorem **trW_lpSum** (x y : WD h) (σ : Weil h) : trW (x.lpSum y) σ = trW x σ + trW y σ

@[simp] theorem **trW_lpChar** (x y : WD h) (σ : Weil h) : trW (x.lpChar y) σ = trW x σ * trW y σ

@[simp] theorem **trW_lpZero** (σ : Weil h) : trW (WD.lpZero h) σ = 0

theorem **trW_sumList** (l : List (WD h)) (σ : Weil h) : trW (sumList l) σ = (l.map fun x => trW x σ).sum

@[simp] theorem **trW_lpTwist** (x : WD h) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (σ : Weil h) : trW (x.lpTwist χ hχ) σ = χ σ * trW x σ

@[simp] theorem **trW_lpOfChar** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (σ : Weil h) : trW (WD.lpOfChar h χ hχ) σ = χ σ

theorem **isIrrWD_mk** (a : WDRep h V) : IsIrrWD (WD.mk a) ↔ a.ρ.IsIrreducible

theorem **nontrivial_of_irr** {ρ : Representation ℂ (Weil h) V} (hρ : ρ.IsIrreducible) : Nontrivial V

theorem **N_eq_zero_of_irr** (a : WDRep h V) (hirr : a.ρ.IsIrreducible) : a.N = 0

theorem **IrrW.mono0** (z : IrrW h) : z.1.mono0

theorem **IrrW.lpDim_pos** (z : IrrW h) : 0 < z.1.lpDim

theorem **isIrrWD_lpTwist** {x : WD h} (hx : IsIrrWD x) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : IsIrrWD (x.lpTwist χ hχ)

theorem **isIrrWD_lpOfChar** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : IsIrrWD (WD.lpOfChar h χ hχ)

theorem **linearIndependent_trW** : LinearIndependent ℂ fun z : IrrW h => trW z.1

@[simp] theorem **mem_frobCoset** {r : ℕ} {τ : Weil h} : τ ∈ frobCoset h r ↔ deg τ = -(r : ℤ)

@[simp] theorem **coe_qPowC** (s : ℂ) : (qPowC w s : ℂ) = ((normV w : ℕ) : ℂ) ^ s

theorem **isSmooth_absChar** (s : ℂ) : IsSmooth (absChar h s)

theorem **absChar_apply** (s : ℂ) (σ : Weil h) : (absChar h s σ : ℂ) = ((normV w : ℕ) : ℂ) ^ (s * deg σ)

theorem **absChar_of_mem_frobCoset** (s : ℂ) {r : ℕ} {τ : Weil h} (hτ : τ ∈ frobCoset h r) : (absChar h s τ : ℂ) = ((normV w : ℕ) : ℂ) ^ (-(r * s))

@[simp] theorem **absChar_zero** : absChar h 0 = 1

theorem **absChar_add** (s t : ℂ) : absChar h (s + t) = absChar h s * absChar h t

@[simp] theorem **dim_irr** (z : IrrW h) : dim h (irr z) = z.1.lpDim

@[simp] theorem **tr_irr** (τ : Weil h) (z : IrrW h) : tr τ (irr z) = trW z.1 τ

@[simp] theorem **twist_irr** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (z : IrrW h) : twist χ hχ (irr z) = irr (z.twist χ hχ)

theorem **tr_apply** (τ : Weil h) (x : VR h) : tr τ x = ∑ z ∈ x.support, (x z : ℂ) * trW z.1 τ

theorem **dim_apply** (x : VR h) : dim h x = ∑ z ∈ x.support, x z * (z.1.lpDim : ℚ)

theorem **irr_mem_posCone** (z : IrrW h) : irr z ∈ posCone h

theorem **eq_sum_irr** (x : VR h) : x = ∑ z ∈ x.support, x z • irr z

theorem **tr_one** (x : VR h) : tr 1 x = (dim h x : ℂ)

@[simp] theorem **tr_twist** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (τ : Weil h) (x : VR h) : tr τ (twist χ hχ x) = χ τ * tr τ x

theorem **tr_tw** (s : ℂ) (τ : Weil h) (x : VR h) : tr τ (tw s x) = ((normV w : ℕ) : ℂ) ^ (s * deg τ) * tr τ x

theorem **tr_tw_of_mem_frobCoset** (s : ℂ) {r : ℕ} {τ : Weil h} (hτ : τ ∈ frobCoset h r) (x : VR h) : tr τ (tw s x) = ((normV w : ℕ) : ℂ) ^ (-(r * s)) * tr τ x

theorem **ext_tr** {x y : VR h} (H : ∀ τ, tr τ x = tr τ y) : x = y

@[simp] theorem **dim_twist** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) (x : VR h) : dim h (twist χ hχ x) = dim h x

@[simp] theorem **dim_tw** (s : ℂ) (x : VR h) : dim h (tw s x) = dim h x

@[simp] theorem **tw_zero** (x : VR h) : tw 0 x = x

theorem **tw_tw** (s t : ℂ) (x : VR h) : tw t (tw s x) = tw (s + t) x

theorem **exists_list_tr** (x : WD h) : ∃ l : List (IrrW h), ∀ τ, tr τ (l.map irr).sum = trW x τ

@[simp] theorem **tr_ofWD** (τ : Weil h) (x : WD h) : tr τ (ofWD x) = trW x τ

@[simp] theorem **dim_ofWD** (x : WD h) : dim h (ofWD x) = x.lpDim

@[simp] theorem **ofWD_irr** (z : IrrW h) : ofWD z.1 = irr z

@[simp] theorem **ofWD_lpZero** : ofWD (WD.lpZero h) = 0

@[simp] theorem **ofWD_lpSum** (x y : WD h) : ofWD (x.lpSum y) = ofWD x + ofWD y

@[simp] theorem **ofWD_lpTwist** (x : WD h) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : ofWD (x.lpTwist χ hχ) = twist χ hχ (ofWD x)

@[simp] theorem **ofWD_lpTw** (x : WD h) (s : ℝ) : ofWD (x.lpTw s) = tw s (ofWD x)

@[simp] theorem **tr_ofRep** (τ : Weil h) (ρ : Representation ℂ (Weil h) V) (hρ : IsSmooth ρ) (hss : ρ.IsSemisimpleRepresentation) : tr τ (ofRep ρ hρ hss) = LinearMap.trace ℂ V (ρ τ)

@[simp] theorem **dim_ofRep** (ρ : Representation ℂ (Weil h) V) (hρ : IsSmooth ρ) (hss : ρ.IsSemisimpleRepresentation) : dim h (ofRep ρ hρ hss) = finrank ℂ V

@[simp] theorem **tr_ofChar** (τ : Weil h) (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : tr τ (ofChar χ hχ) = χ τ

@[simp] theorem **ofWD_lpOfChar** (χ : Weil h →* ℂˣ) (hχ : IsSmooth χ) : ofWD (WD.lpOfChar h χ hχ) = ofChar χ hχ

@[simp] theorem **tr_triv** (τ : Weil h) : tr τ (triv h) = 1

@[simp] theorem **dim_triv** : dim h (triv h) = 1

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TameDescentExtendedActionExtraVector

theorem **act_mem_Man** (g : TameGal K l e a) {v : Vec K a ι} (hv : v ∈ P.Man) : Δ.act g v ∈ P.Man

theorem **act_mem_EM** (g : TameGal K l e a) {η : ℕ} {v : Vec K a ι} (hv : v ∈ P.EM η) : Δ.act g v ∈ P.EM η

theorem **act_EN** (g : TameGal K l e a) (v : Vec K a ι) : Δ.act g (P.EN v) = P.EN (Δ.act g v)

@[simp] theorem **actPol_cls** (g : TameGal K l e a) (v : Vec K a ι) : Δ.actPol g (P.cls v) = P.cls (Δ.act g v)

theorem **actPol_Mon** (g : TameGal K l e a) (y : Vec K a ι) : Δ.actPol g (P.Mon y) = P.Mon (Δ.act g y)

theorem **cls_mem_invariants** {v : Vec K a ι} (hv : v ∈ Δ.inv) : P.cls v ∈ Δ.actPol.invariants

theorem **Mon_mem_invariants** {y : Vec K a ι} (hy : y ∈ Δ.inv) : P.Mon y ∈ Δ.actPol.invariants

theorem **mem_invOf** {B : Subalgebra O K⟦X⟧} {W : Submodule B (Vec K a ι)} {v : Vec K a ι} : v ∈ Δ.invOf W ↔ v ∈ W ∧ ∀ g, Δ.act g v = v

theorem **mem_Y'** {η : ℕ} {y : Vec K a ι} : y ∈ Δ.Y' η ↔ ∃ y₀ ∈ Δ.Y η, (X : K⟦X⟧) ^ e • y₀ = y

theorem **Y'_le_Y** (η : ℕ) : Δ.Y' η ≤ Δ.Y η

theorem **mem_Yl_iff** {η : ℕ} {y : Vec K a ι} : y ∈ Δ.Yl η ↔ ∃ n : ℕ, (l : K) ^ n • y ∈ Δ.Y η

theorem **mem_Z** {η : ℕ} {w : P.Pol} : w ∈ Δ.Z η ↔ (∃ y ∈ P.EM η, P.cls y = w) ∧ ∀ g, Δ.actPol g w = w

theorem **mem_Zl_of_mem_Z** {η : ℕ} {w : P.Pol} (hw : w ∈ Δ.Z η) : w ∈ Δ.Zl η

theorem **cls_mem_Z** {η : ℕ} {y : Vec K a ι} (hy : y ∈ Δ.Y η) : P.cls y ∈ Δ.Z η

theorem **vPol_mem_Z** {η : ℕ} {w : P.Pol} (hw : w ∈ Δ.Z η) : P.vPol w ∈ Δ.Z η

@[simp] theorem **actO_apply** (g : TameGal K l e a) (v : Vec K a ι) : Δ.actO g v = Δ.act g v

theorem **coe_actEM** (η : ℕ) (g : TameGal K l e a) (v : (P.EM η).restrictScalars O) : (Δ.actEM η g v : Vec K a ι) = Δ.act g v

theorem **quotToPol_act** (η : ℕ) (g : TameGal K l e a) (w) : quotToPol P η (Δ.actQuot η g w) = Δ.actPol g (quotToPol P η w)

theorem **quotToPol_injective** (η : ℕ) : Function.Injective (quotToPol P η)

theorem **map_invariants_actQuot** (η : ℕ) : (Δ.actQuot η).invariants.map (quotToPol P η) = Δ.Z η

theorem **app_mem_inv** {v : Vec K a ι} (hv : v ∈ Δ₁.inv) : F.app v ∈ Δ₂.inv

theorem **app_mem_Minv** {v : Vec K a ι} (hv : v ∈ Δ₁.Minv) : F.app v ∈ Δ₂.Minv

theorem **app_mem_Y** {η : ℕ} {y : Vec K a ι} (hy : y ∈ Δ₁.Y η) : F.app y ∈ Δ₂.Y η

theorem **app_mem_Y'** {η : ℕ} {y : Vec K a ι} (hy : y ∈ Δ₁.Y' η) : F.app y ∈ Δ₂.Y' η

theorem **onPol_act** (g : TameGal K l e a) (w : P.Pol) : F.onPol (Δ₁.actPol g w) = Δ₂.actPol g (F.onPol w)

theorem **onPol_mem_Z** {η : ℕ} {w : P.Pol} (hw : w ∈ Δ₁.Z η) : F.onPol w ∈ Δ₂.Z η

theorem **onPol_mem_Zl** {η : ℕ} {w : P.Pol} (hw : w ∈ Δ₁.Zl η) : F.onPol w ∈ Δ₂.Zl η

theorem **tameDescent_extended_action_extraVector** (g : TameGal K l e a) : (Δ.plus hy hinv).act g xVec = xVec

theorem **nabla_mem_MM_of_basis** (h : ∀ i, P.nabla (P.phi (bas i)) ∈ P.MM) (h' : ∀ i, P.phi (bas i) ∈ P.MM) {m : Vec K a ι} (hm : m ∈ P.MM) : P.nabla m ∈ P.MM

theorem **EN_mem_Man_of_basis** (h : ∀ i, P.EN (bas i) ∈ P.Man) {m : Vec K a ι} (hm : m ∈ P.Man) : P.EN m ∈ P.Man

theorem **smul_mem_ext_of_basis** {c : K⟦X⟧} (h : ∀ i, c • bas i ∈ P.L) {B : Subalgebra O K⟦X⟧} (hB : A O K ≤ B) {m : Vec K a ι} (hm : m ∈ P.Mext B) : c • m ∈ P.ext B

theorem **smul_mem_Mext_of_basis** {c : K⟦X⟧} (h : ∀ i, c • P.phi (bas i) ∈ P.M) {B : Subalgebra O K⟦X⟧} (hB : A O K ≤ B) {m : Vec K a ι} (hm : m ∈ P.ext B) : c • m ∈ P.Mext B

theorem **bas_inl** (i : ι) : bas (Sum.inl i) = (P.plusInl hy).app (bas i)

theorem **bas_inr** : (bas (Sum.inr ()) : Vec K a (ι ⊕ Unit)) = xVec

theorem **plus_upper** {η : ℕ} (h : ∀ v ∈ P.L, E K l e ^ η • v ∈ P.M) (hη : y ∈ P.EM η) : ∀ v ∈ (P.plus hy).L, E K l e ^ η • v ∈ (P.plus hy).M

theorem **plus_lower** {n : ℕ} (h : ∀ v ∈ P.M, E K l e ^ n • v ∈ P.L) (hn : E K l e ^ n • y ∈ P.L) : ∀ v ∈ (P.plus hy).M, E K l e ^ n • v ∈ (P.plus hy).L

theorem **isExactTriple_plus** [NormedAlgebra ℚ_[l] K] [NeZero e] : IsExactTriple (P.plusInl hy) (P.plusProj hy) where

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ToUnitsCompToCircle

theorem **liftC_cast** (τ : D.what₂.Tau) : (liftC D i τ : ℚ) = ((pS D i τ : ℚ) − (qS D i τ : ℚ)) / 2 − (D.hc τ).e i − 1 / 2

theorem **liftWeight_good** : (liftWeight D i).Good where

theorem **liftWeight_delta** : (liftWeight D i).delta (Wc D i).rank = RealDat.Delta i

theorem **map_loc** {M N : Type} [CommRing M] [CommRing N] (f : M →+* N) (e a : M) : f (1 + e * (a − 1)) = 1 + f e * (f a − 1)

theorem **exists_isLocUnit** {PP : Pl D.host.E} (h : Formalism.L2.NTheta.IsIdemAt (idemAt PP) PP) (u : (kP PP)ˣ) : ∃ x : (AE D)ˣ, IsLocUnit PP u x

theorem **IsLocUnit.eq** {M : Type} [CommGroup M] {χ : (AE D)ˣ →* M} {PP : Pl D.host.E} {c : (kP PP)ˣ →* M} (h : IsResAt χ PP c) {u : (kP PP)ˣ} {x : (AE D)ˣ} (hx : IsLocUnit PP u x) : χ x = c u

theorem **isResAt_unique** {M : Type} [CommGroup M] {χ : (AE D)ˣ →* M} {PP : Pl D.host.E} (h : Formalism.L2.NTheta.IsIdemAt (idemAt PP) PP) {c c' : (kP PP)ˣ →* M} (hc : IsResAt χ PP c) (hc' : IsResAt χ PP c') : c = c'

theorem **IsLocUnit.eq_one** {M : Type} [CommGroup M] {χ : (AE D)ˣ →* M} {Q P : Finset (Pl D.host.E)} (hχ : UnramOutside χ Q) (hP : SplitSet P) (hQ : Q ⊆ P) {PP : Pl D.host.E} (hPP : PP ∈ P) (hn : PP ∉ Q) {u : (kP PP)ˣ} {x : (AE D)ˣ} (hx : IsLocUnit PP u x) : χ x = 1

theorem **eq_res_of_chrEq** {c : (AE D)ˣ →* Circle} {χ : Formalism.L2.Host.HChar D.host.E} (hc : ChrEq c χ) {PP : Pl D.host.E} (h : Formalism.L2.NTheta.IsIdemAt (idemAt PP) PP) (ht : χ.TameAt PP) {ρ : (kP PP)ˣ →* ℂˣ} (hρ : IsRes χ PP ρ) (x : (AE D)ˣ) (hx : (1 − idemAt PP) ∗ (x : AE D) = 1 − idemAt PP) (hi : toEAf D (x : AE D) ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) (u : (kP PP)ˣ) (hu : resS D.host.F PP ⟨_, hi⟩ = u) : ((c x : Circle) : ℂ) = ((ρ u : ℂˣ) : ℂ)

@[simp] theorem **coe_toCircle** {M : Type} [Group M] [Finite M] (c : M →* ℂˣ) (u : M) : ((toCircle c u : Circle) : ℂ) = ((c u : ℂˣ) : ℂ)

theorem **toUnits_comp_toCircle** {M : Type} [Group M] [Finite M] (c : M →* ℂˣ) : Circle.toUnits.comp (toCircle c) = c

theorem **conjTup_of** {B : Base (Vc D i) (Wc D i)} {τ : ℂ ≃+* ℂ} {ψ ψ' : Twist D Circle} {t t' : T0} (h : IsTupOf R B ψ t) (h' : IsTupOf R B ψ' t') (hc : ConjChar τ ψ ψ') (hf : FixDat B τ) : ConjTup R τ t t'

theorem **ConjChar.symm** {τ : ℂ ≃+* ℂ} {ψ ψ' : Twist D Circle} (h : ConjChar τ ψ ψ') : ConjChar τ.symm ψ' ψ

theorem **FixDat.symm** {S T : HSp D} {B : Base S T} {τ : ℂ ≃+* ℂ} (h : FixDat B τ) : FixDat B τ.symm

theorem **FixDat.conjChar** {S T : HSp D} {B : Base S T} {τ : ℂ ≃+* ℂ} (h : FixDat B τ) : ConjChar τ B.de B.de

theorem **ConjChar.res** {τ : ℂ ≃+* ℂ} {ψ ψ' : Twist D Circle} (h : ConjChar τ ψ ψ') (hψ : ψ.Good) (hψ' : ψ'.Good) {PP : Pl D.host.E} (hPP : PP ∈ ψ.Q) (u : (kP PP)ˣ) : ((ψ'.res PP u : Circle) : ℂ) = τ ((ψ.res PP u : Circle) : ℂ)

theorem **exponent** (τ : D.what₂.Tau) : (B.w.c τ : ℚ) − ((B.w.p τ : ℚ) − B.w.q τ) / 2 + 1 / 2 = −(D.hc τ).e i

theorem **fock_eq** (f : WO (Vc D i) (Wc D i) → ℂ) : B.F.fock f = pureTensorO R.ψ R.θ (Vc D i) (Wc D i) (fun _ => 1) (cart (R.MW i (sg0 D A))) (R.BV i) (onb (R.MW i (sg0 D A))) f

theorem **series_eq_ser** (t : T0) (f : WO (Vc D i) (Wc D i) → ℂ) (u : ∀ ρ, lowT (R.MW i (sg0 D A)) ρ) (g : GA D (Vc D i)) (w : (R.cV i).W) : series B.F (R.ωP i t) f g (R.ioP i (PiTensorProduct.tprod ℂ u) w) = ser R i t f u g w

theorem **mu_mul** {ψ : Twist D Circle} {t : T0} (ht : IsTupOf R B ψ t) : B.mu ∗ ψ.chi = R.toLevCh.mu t ((i : ℕ) + 1)

theorem **isWeil** (hω : R.thP_ωP) {ψ : Twist D Circle} {t : T0} (ht : IsTupOf R B ψ t) : B.IsWeil ψ.chi (R.ωP i t)

theorem **muC_eq** {t₀ : T0} (hL : R.toLevCh.Is t₀) (hB : B.Good) {PP : Pl D.host.E} (hPP : PP ∈ R.Pg) : B.G.muC PP = (R.chi ((i : ℕ) + 1) PP)⁻¹

theorem **fix_mu** {t₀ : T0} (hL : R.toLevCh.Is t₀) (hB : B.Good) {τ : ℂ ≃+* ℂ} (hτ : ∀ PP ∈ R.Pg, ∀ u, τ ((R.chi ((i : ℕ) + 1) PP u : ℂˣ) : ℂ) = ((R.chi ((i : ℕ) + 1) PP u : ℂˣ) : ℂ)) : ∀ PP ∈ B.G.Pg, ∀ u, τ ((B.G.mu PP u : Circle) : ℂ) = ((B.G.mu PP u : Circle) : ℂ)

theorem **IsTupOf.xiR_eq** {B : Base (Vc D i) (Wc D i)} {ψ : Twist D Circle} {t t₀ : T0} (ht : IsTupOf R B ψ t) (hL : R.toLevCh.Is t₀) (hB : B.Good) (hψ : ψ.Good) (hQ : B.admitted ψ.Q) {PP : Pl D.host.E} (hPP : PP ∈ R.Q t) : R.xiR t PP = B.de.locC PP ∗ ψ.locC PP

theorem **isResAt_tupTwist** {t₀ : T0} (hL : R.toLevCh.Is t₀) (t : T0) {PP : Pl D.host.E} (hPP : PP ∈ R.Q t) (h : Formalism.L2.NTheta.IsIdemAt (idemAt PP) PP) : IsResAt (tupTwist R t).chi PP ((tupTwist R t).res PP)

theorem **isUnit_of_isONBasis** {k p : ℕ} {J s B : Matrix (Fin k) (Fin k) ℂ} (h : IsONBasis J s B p) : IsUnit B

theorem **baseOf_baseAt** (hB : R.models_BV) (δ : Twist D Circle) (P : LevelP R.Pg (Wc D i).rank ℂ) : BaseOf R i (baseAt R i hB δ P)

theorem **isTupOf_baseAt** (hB : R.models_BV) (t₀ : T0) (P : LevelP R.Pg (Wc D i).rank ℂ) : IsTupOf R (baseAt R i hB (tupTwist R t₀) P) (Twist.triv Circle ∅) t₀

theorem **isTupOf_tupTwist** (hB : R.models_BV) (t : T0) (P : LevelP R.Pg (Wc D i).rank ℂ) : IsTupOf R (baseAt R i hB (Twist.triv Circle ∅) P) (tupTwist R t) t

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TorsionExponentCondition

theorem **one_le_Lg** (L : SizeData₂ D) : 1 ≤ TwistedPlaces.large₂ L

theorem **log_le_Lg** (L : SizeData₂ D) (l k : ℕ) (hl : 2 ≤ l) (hk : (k : ℝ) ≤ 8 ∗ L.sc.b) : (Nat.log l k : ℝ) ≤ 5 ∗ TwistedPlaces.large₂ L

theorem **one_le_b** (L : SizeData₂ D) : 1 ≤ L.sc.b

theorem **sizes** (D : D7) : (1 : ℝ) ≤ D.R.t ∧ (1 : ℝ) ≤ D.R.n ∧ (1 : ℝ) ≤ D.R.nV

theorem **condCB** (h : Canon L HW R X) : CondCB L X 80

theorem **torsion_exponent_condition** (h : Canon L HW R X) : KillingConditions.ConstantColumnBound L X 80

theorem **eta_le** (h : Canon L HW R X) : (X.eta : ℝ) ≤ 2 ∗ L.etaSz ∧ L.etaSz ≤ (X.eta : ℝ) + 1

theorem **not_inert** {l : ℕ} (h1 : TwistedPlaces.SplitK D l) : ¬ TwistedPlaces.InertK D l

theorem **decl** (h : Canon L HW R X) (hJ : AdmissibleParameterJ.ConclusionJa D) (i : D.nd.Loc) (hg : D.nd.l i ∉ TwistedPlaces.gset D) (hr : KillingConditions.RangeAD L X.exc (D.nd.l i)) (hs : TwistedPlaces.SplitK D (D.nd.l i)) (hex : ∃ dc : TruncatedSeriesCountingIdentities.Decl D i, EmbeddingTypes.hypDecl i dc hJ.choose) : EmbeddingTypes.hypDecl i (X.dc i) hJ.choose

theorem **col_empty** (h : Canon L HW R X) (i : D.nd.Loc) (hn : ¬ GlobalTheory.IndefiniteRowSideFullHecke.SplitPlaceTypes.SplitRowPrime D.host (D.nd.l i)) : (X.dc i).col = ∅

theorem **cLL_le** (L : SizeData₂ D) (i : D.nd.Loc) (u : HeightOneSpectrum (𝒪 D.host.E)) (U : Finset ℤ) (U' : Set ℤ) (hU : U' ⊆ U) (hN : U.card ≤ 4 ∗ D.R.n ∗ D.R.nV) (hb : ∀ a ∈ U, −(hOf L : ℤ) ≤ a ∧ a ≤ hOf L) : LocalSituation.cLL (D.nd.l i) (LocalSituation.mN (D.nd.l i) 1 (hOf L + hOf L)) (LocalSituation.mN (D.nd.l i) 1 (hOf L + hOf L) − 1) U' 0 1 ≤ c0of L i u

theorem **window** (L : SizeData₂ D) (hJ : AdmissibleParameterJ.JumpAbsoluteBound L) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (ρ : D.what₂.Tau) (up : Bool) (a : ℤ) (ha : (a : ℚ) ∈ AdmissibleParameterJ.JJ D up j ρ) : (−(⌊2 ∗ L.sc.b⌋₊ : ℤ) ≤ a ∧ a + 1 ≤ (⌊2 ∗ L.sc.b⌋₊ : ℤ)) ∧ D.R.t ≤ ⌊2 ∗ L.sc.b⌋₊

theorem **exists_canon** (L : SizeData₂ D) (HW : HWOf D.loc) (R : KillingConditions.RowK) (I : TwIn D) (exc : Finset ℕ) (hC : GlobalTheory.IndefiniteRowSideFullHecke.BlockRepresentationsSemisimple.Complement_at D) (hr : ∀ i, ∃ r, GoodR D.nd i r) : ∃ X, Canon L HW R X ∧ X.I = I ∧ X.exc = exc

theorem **wild_charpoly** (h1 : DepthZeroG D) (h2 : LgcG D) (i : D.nd.Loc) (r : D.nd.X i → (GE D.nd →* GL (Fin D.R.nV) (D.nd.O i))) (hr : GoodR D.nd i r) (w) (hw : w ∈ D.nd.S i) (hg : ∃ k, w ∈ placesOver D.host.E (D.host.g k)) (x : D.nd.X i) (σ : GE D.nd) (hσ : σ ∈ wildSet D.nd (D.nd.plOm.1 w)) (hI : σ ∈ Ine (F

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TranslateMemCoefLatFn

theorem **fockSubst_one** (p : D.what₂.Tau → ℕ) : fockSubst (m

theorem **fockSubst_mul** (p : D.what₂.Tau → ℕ) (u u' : GLR D n) : fockSubst (m

theorem **fockGL_apply** (w : Weight D n) (B u : GLR D n) (P : PolF D m n) : fockGL m w B u P = (((detFam w.c u)⁻¹ : ℂˣ) : ℂ) • fockSubst w.p (inBasis B u) P

theorem **delta_mem_harm** (w : Weight D n) (B : GLR D n) : w.delta m ∈ harm m w B

theorem **fockGL_mem_harm** (w : Weight D n) (B u : GLR D n) {P : PolF D m n} (h : P ∈ harm m w B) : fockGL m w B u P ∈ harm m w B

theorem **aeval_fockPt_fockSubst** (p : D.what₂.Tau → ℕ) (u y : GLR D n) (P : PolF D m n) : aeval (fockPt p y) (fockSubst p u P) = aeval (fockPt p (u⁻¹ ∗ y)) P

theorem **harmFn_apply** (w : Weight D n) (B x : GLR D n) (P : PolF D m n) : harmFn m w B P x = ((detFam w.c (B⁻¹ ∗ x) : ℂˣ) : ℂ) ∗ aeval (fockPt w.p (B⁻¹ ∗ x)) P

theorem **harmFn_fockGL** (w : Weight D n) (B u x : GLR D n) (P : PolF D m n) : harmFn m w B (fockGL m w B u P) x = harmFn m w B P (u⁻¹ ∗ x)

theorem **coefFn_rhoVr** {emb : D.what₂.Tau → (D.host.E →+* K)} (hemb : ∀ τ, σ.comp (emb τ) = tauE D τ) (lam : D.what₂.Tau → DomWeight n) (γ : GL (Fin n) D.host.E) (F : VV K lam) (x : GLR D n) : coefFn σ lam (rhoVr emb lam γ F : VV K lam) x = coefFn σ lam F ((embGL (tauE D) γ)⁻¹ ∗ x)

theorem **mem_harmLat** {w : Weight D n} {B : GLR D n} {M : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} {P : PolF D m n} : P ∈ harmLat σ emb O m w B M ↔ P ∈ harm m w B ∧ harmFn m w B P ∈ coefLatFn σ emb O w.lam M

theorem **nabAtr_of_mem_iff** (lam : D.what₂.Tau → DomWeight n) (γ : GL (Fin n) D.host.E) {M M' : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} (hM : ∀ x, x ∈ M' ↔ (γ⁻¹).val *ᵥ x ∈ M) : nabAtr emb lam M' = (nabAtr emb lam M).map ((rhoVr emb lam γ).restrictScalars (𝒪 K))

theorem **translate_mem_coefLatFn** (hemb : ∀ τ, σ.comp (emb τ) = tauE D τ) (lam : D.what₂.Tau → DomWeight n) (γ : GL (Fin n) D.host.E) {M M' : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} (hM : ∀ x, x ∈ M' ↔ (γ⁻¹).val *ᵥ x ∈ M) {φ : GLR D n → ℂ} (h : φ ∈ coefLatFn σ emb O lam M) : (fun x => φ ((embGL (tauE D) γ)⁻¹ ∗ x)) ∈ coefLatFn σ emb O lam M'

theorem **fockGL_mem_harmLat** (hemb : ∀ τ, σ.comp (emb τ) = tauE D τ) (w : Weight D n) (B : GLR D n) (γ : GL (Fin n) D.host.E) {M M' : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} (hM : ∀ x, x ∈ M' ↔ (γ⁻¹).val *ᵥ x ∈ M) {P : PolF D m n} (h : P ∈ harmLat σ emb O m w B M) : fockGL m w B (embGL (tauE D) γ) P ∈ harmLat σ emb O m w B M'

theorem **Weight.first_good** {p q : D.what₂.Tau → ℕ} (c : D.what₂.Tau → ℤ) (hp : ∀ τ, 0 < p τ) (h : ∀ τ, p τ + q τ ≤ n) : (Weight.first D n p q c).Good where

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TrivIsCr

theorem **Frame.triv_isCr** (bk : BK K) (d : ℕ) : (Frame.triv bk d).IsCr

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TruncQuotSymmTruncTensor

theorem **red_eq_zero** (A : LocAlg O Or) {a : A.A} (ha : a ∈ maximalIdeal A.A) : A.red a = 0

@[simp] theorem **comp_f** {A B C : LocAlg O Or} (f : A ⟶ B) (g : B ⟶ C) : Hom.f (f ≫ g) = g.f.comp f.f

@[simp] theorem **id_f** (A : LocAlg O Or) : Hom.f (𝟙 A) = AlgHom.id Or A.A

@[simp] theorem **red_algebraMap** (A : LocAlg O Or) (r : Or) : A.red (algebraMap Or A.A r) = residue Or r

theorem **red_comp** {A B : LocAlg O Or} (f : A ⟶ B) (a : A.A) : B.red (f.f a) = A.red a

@[simp] theorem **truncTensor_tmul** (R : LocAlg O Or) (M : Type) [CommRing M] [Algebra R.A M] (k : ℕ) (r : R.A) (x : M) : R.truncTensor M k (Ideal.Quotient.mk _ r ⊗ₜ[R.A] x) = Ideal.Quotient.mk _ r ⊗ₜ[R.A] x

@[simp] theorem **truncTensor_one_tmul** (R : LocAlg O Or) (M : Type) [CommRing M] [Algebra R.A M] (k : ℕ) (x : M) : R.truncTensor M k ((1 : R.Trunc (k + 1)) ⊗ₜ[R.A] x) = (1 : R.Trunc k) ⊗ₜ[R.A] x

@[simp] theorem **truncQuot_mk** (R : LocAlg O Or) (M : Type) [CommRing M] [Algebra R.A M] (k : ℕ) (x : M) : R.truncQuot M k (Ideal.Quotient.mk _ x) = (1 : R.Trunc k) ⊗ₜ[R.A] x

theorem **truncQuot_symm_tmul** (R : LocAlg O Or) (M : Type) [CommRing M] [Algebra R.A M] (k : ℕ) (x : M) : (R.truncQuot M k).symm ((1 : R.Trunc k) ⊗ₜ[R.A] x) = Ideal.Quotient.mk _ x

theorem **truncQuot_symm_truncTensor** (R : LocAlg O Or) (M : Type) [CommRing M] [Algebra R.A M] (k : ℕ) (z : R.Trunc (k + 1) ⊗[R.A] M) : (R.truncQuot M k).symm (R.truncTensor M k z) = Ideal.Quotient.factorPow (R.mapMax M) (Nat.le_succ (k + 1)) ((R.truncQuot M (k + 1)).symm z)

theorem **Push.isPush** {S S' : Type} [CommRing S] [Algebra O S] [CommRing S'] [Algebra O S'] {red : S →+* ResidueField Or} {red' : S' →+* ResidueField Or} {f : S →+* S'} {D : Defo p Hb S red} {D' : Defo p Hb S' red'} (P : Push f D D') : IsPush f D D'

theorem **isPush_iff_nonempty** {S S' : Type} [CommRing S] [Algebra O S] [CommRing S'] [Algebra O S'] {red : S →+* ResidueField Or} {red' : S' →+* ResidueField Or} {f : S →+* S'} {D : Defo p Hb S red} {D' : Defo p Hb S' red'} : IsPush f D D' ↔ Nonempty (Push f D D')

theorem **ProRep.trans** {R : LocAlg O Or} (U : ProRep p Hb R) (k : ℕ) : IsPush (R.truncMap k) (U.univ (k + 1)) (U.univ k)

theorem **IsFullSet.comp_equiv** {S B T J J' : Type} [CommRing S] [CommRing B] [Algebra S B] [CommRing T] [Algebra S T] {P : J → (B →ₐ[S] T)} (h : IsFullSet P) (e : J' ≃ J) : IsFullSet (P ∘ e)

theorem **IsFullSet.comp** {S B T T' J : Type} [CommRing S] [CommRing B] [Algebra S B] [CommRing T] [Algebra S T] [CommRing T'] [Algebra S T'] {P : J → (B →ₐ[S] T)} (h : IsFullSet P) (g : T →ₐ[S] T') : IsFullSet fun i => g.comp (P i)

@[simp] theorem **map_α** (L : Level p G ι ϖ n m T) (g : T →ₐ[S] T') (x) : (L.map g).α x = g.comp (L.α x)

@[simp] theorem **act_α** (L : Level p G ι ϖ n m T) (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) (x) : (L.act g).α x = L.α (g.val.mulVec x)

theorem **map_map** {T'' : Type} [CommRing T''] [Algebra S T''] (L : Level p G ι ϖ n m T) (g : T →ₐ[S] T') (h : T' →ₐ[S] T'') : (L.map g).map h = L.map (h.comp g)

@[simp] theorem **map_id** (L : Level p G ι ϖ n m T) : L.map (AlgHom.id S T) = L

theorem **act_map** (L : Level p G ι ϖ n m T) (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) (h : T →ₐ[S] T') : (L.act g).map h = (L.map h).act g

theorem **act_mul** (L : Level p G ι ϖ n m T) (a b : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) : (L.act a).act b = L.act (a ∗ b)

@[simp] theorem **act_one** (L : Level p G ι ϖ n m T) : L.act 1 = L

theorem **aut_spec** (k : ℕ) (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) : (L.lev k).map (L.aut k g) = (L.lev k).act g

theorem **aut_one** (k : ℕ) : L.aut k 1 = AlgHom.id _ _

theorem **aut_mul** (k : ℕ) (g h : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) : L.aut k (g ∗ h) = (L.aut k g).comp (L.aut k h)

theorem **truncTensor_aut** (k : ℕ) (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) (z : R.Trunc (k + 1) ⊗[R.A] L.Rm) : R.truncTensor L.Rm k (L.aut (k + 1) g z) = L.aut k g (R.truncTensor L.Rm k z)

theorem **succ_strictMono** : StrictMono fun k : ℕ => k + 1

theorem **actTrunc_compat** (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) {k : ℕ} : (Ideal.Quotient.factorPow (R.mapMax L.Rm) (succ_strictMono.monotone k.le_succ)).comp (L.actTrunc g (k + 1)) = L.actTrunc g k

theorem **eq_of_tmul** {x y : L.Rm} (h : ∀ k, (1 : R.Trunc k) ⊗ₜ[R.A] x = (1 : R.Trunc k) ⊗ₜ[R.A] y) : x = y

theorem **tmul_actRing** (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) (k : ℕ) (x : L.Rm) : (1 : R.Trunc k) ⊗ₜ[R.A] L.actRing g x = L.aut k g ((1 : R.Trunc k) ⊗ₜ[R.A] x)

theorem **actRing_one** : L.actRing 1 = RingHom.id _

theorem **actRing_mul** (g h : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) : L.actRing (g * h) = (L.actRing g).comp (L.actRing h)

theorem **actRing_algebraMap** (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) (r : R.A) : L.actRing g (algebraMap R.A L.Rm r) = algebraMap R.A L.Rm r

theorem **smul_def** (g : GL (Fin n) (O ⧸ Ideal.span {ϖ ^ m})) (x : L.Rm) : g • x = L.actRing g x

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TwistedModelAdmissibleCocycles

theorem **_root_.GlobalTheory.Models.exists_ob** {M : Type} [AddCommGroup M] [Module k M] [Module.Finite k M] (π : Representation k G M) : ∃ (A : _root_.GlobalTheory.Models.Ob k G) (e : M ≃ₗ[k] (Fin A.m → k)), ∀ g x, e (π g x) = A.ρ g (e x)

theorem **const_smul_mem_lat** {m : ℕ} (N : Submodule k⟦X⟧ (Vc k m)) (c : k) {x : Vc k m} (hx : x ∈ N) : c • x ∈ N

theorem **mem_pairSpace** {m r : ℕ} {D : ModPG R n (Vc k m)} {Dt : TopData n r} {N : Lats k n m} {b g : ZMod n → Vc k m} : (b, g) ∈ pairSpace D Dt N ↔ InZZ D Dt N b g

theorem **herrD_apply** {m : ℕ} (D : ModPG R n (Vc k m)) (x : ZMod n → Vc k m) : herrD D x = (fun i => D.φ i (x (i - 1)) - x i, fun i => D.γ i (x i) - x i)

theorem **herrB_eq_range** {m : ℕ} (D : ModPG R n (Vc k m)) : herrB D = LinearMap.range (herrD D)

theorem **mem_latB** {m : ℕ} {D : ModPG R n (Vc k m)} {N : Lats k n m} {p : (ZMod n → Vc k m) × (ZMod n → Vc k m)} : p ∈ latB D N ↔ ∃ x : ZMod n → Vc k m, (∀ i, x i ∈ N i) ∧ herrD D x = p

theorem **latB_le_herrB** {m : ℕ} (D : ModPG R n (Vc k m)) (N : Lats k n m) : latB D N ≤ herrB D

theorem **one_le_w_and_wst** {r : ℕ} (Dt : TopData n r) (i : ZMod n) : 1 ≤ Dt.w i ∧ 1 ≤ Dt.wst i

theorem **latB_le_pairSpace** {m r : ℕ} {D : ModPG R n (Vc k m)} {Dt : TopData n r} {N : Lats k n m} (h : ABC D Dt.w Dt.wst N) : latB D N ≤ pairSpace D Dt N

theorem **IsModel.le_maxModel** {F : DFun R n G} {Base : ∀ A : Ob k G, Lats k n A.m → Prop} {A : Ob k G} {N : Lats k n A.m} (h : IsModel F Base A N) (i : ZMod n) : N i ≤ maxModel F Base A i

theorem **maxModel_eq** {F : DFun R n G} {Base : ∀ A : Ob k G, Lats k n A.m → Prop} {A : Ob k G} {N : Lats k n A.m} (h : IsModel F Base A N) (hmax : ∀ N', IsModel F Base A N' → ∀ i, N' i ≤ N i) : maxModel F Base A = N

theorem **IsModel.of_base** {F : DFun R n G} {Base : ∀ A : Ob k G, Lats k n A.m → Prop} {A : Ob k G} {N : Lats k n A.m} (hA : F.Obj A) (h : Base A N) : IsModel F Base A N

theorem **ModTh.laws** {F : DFun R n G} {r : ℕ} {Dt : TopData n r} (M : ModTh F Dt) : Law.All F Dt M.data

theorem **ModTh.data_ofLaws** {F : DFun R n G} {r : ℕ} {Dt : TopData n r} (Y : ModData k G n) (h : Law.All F Dt Y) : (ModTh.ofLaws Y h).data = Y

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.TypePreservedOfAdmissibleIntegral

theorem **zpow_apply'** (i : X.Tw) (Q : Finset (PlE D.host)) (k : ℕ) (ψ : X.RCG i Q →* (X.Ok i k)ˣ) (a : ℤ) (g : X.RCG i Q) : (ψ ^ a) g = (ψ g) ^ a

theorem **not_coprime_mul** (i : X.Tw) (n : ℤ) : ¬ IsCoprime (n * (X.l i : ℤ)) (X.l i : ℤ)

theorem **admissibleBounds_of_all** {N0 : X.Tw → ℕ} (h : admissibleIndexBoundAllPowers Y bkP N0) : TwistingCharacterExistence Y bkP N0

theorem **provedGalois_of_all** {E : ExpDat Y} {N0 : X.Tw → ℕ} {i : X.Tw} (h : goodTwistExistsAllCoprimePowers Y bkP E N0 i) : TwistTheoremAtIndex Y bkP E N0 i

theorem **provedGaloisAll_iff** {E : ExpDat Y} {N0 : X.Tw → ℕ} {i : X.Tw} : goodTwistExistsAllCoprimePowers Y bkP E N0 i ↔ GlobalTheory.IndefiniteRowSideFullHecke.LemmaAtAllGoodPlaces.LemmaAllAt Y bkP E N0 i

theorem **twGoodAt_of_all** {D : D7} {C : NChoice D} {I : TwIn D} (h : TwGoodAtAll D C I) : TwGoodAt D C I

theorem **psiAt_zpow** {Γ A : Type} [Group Γ] [CommGroup A] (b : Bool) (ψ : Γ →* A) (a : ℤ) : BlochKatoReduction.Gen.psiAt b (ψ ^ a) = BlochKatoReduction.Gen.psiAt b ψ ^ a

theorem **twGoodAll_side** {D : D7} {C : NChoice D} {I : TwIn D} (h : TwGoodAtAll D C I) (w) (hw : D.what₂.muAt w ≠ D.what₂.mu0) (b : Bool) (a : ℤ) (ha : IsCoprime a (D.ntw.l (C.twI w) : ℤ)) : TwGoodOne D I (C.twI w) b (C.Q w) (C.k' w) (C.psi' w ^ a)

theorem **zero_mem_ball** (i : X.Tw) (j : ℕ) (b : Bool) {B : ℝ} (hB : 0 ≤ B) : (0 : Y.Fm i j b) ∈ Ball T i j b B

theorem **bsup_eq** (i : X.Tw) (j : ℕ) (b : Bool) : Y.Bsup i j b = Submodule.span (X.O i) (Ball T i j b 1)

theorem **norm_l** (i : X.Tw) : ‖((X.l i : ℕ) : ℂ_[X.l i])‖ = ((X.l i : ℝ))⁻¹

theorem **smul_l_pow** (i : X.Tw) (j : ℕ) (b : Bool) (c : ℕ) (f : Y.Fm i j b) : ((X.l i : X.O i) ^ c) • f = ((X.l i : FractionRing (X.O i)) ^ c) • f

theorem **l_pow_smul_mem** (i : X.Tw) (j : ℕ) (b : Bool) (c : ℕ) (B : ℝ) (f : Y.Fm i j b) (hf : f ∈ Ball T i j b ((X.l i : ℝ) ^ c * B)) : ((X.l i : X.O i) ^ c) • f ∈ Ball T i j b B

theorem **l_pow_smul_span** (i : X.Tw) (j : ℕ) (b : Bool) (c : ℕ) (f : Y.Fm i j b) (hf : f ∈ Submodule.span (X.O i) (Ball T i j b ((X.l i : ℝ) ^ c))) : ((X.l i : X.O i) ^ c) • f ∈ Y.Bsup i j b

theorem **bsup_stable_of_gKeeps** (h : GKeeps T) (i : X.Tw) (hi : X.NonzeroIndexPlace i) (j : ℕ) (hj : j ∈ Finset.Icc 1 D.R.m) (b : Bool) (g) : (Y.Bsup i j b).map ((Y.actG i j b g).toLinearMap.restrictScalars (X.O i)) ≤ Y.Bsup i j b

theorem **bsup_stable_of_hkKeeps** (h : HkKeeps T) (i : X.Tw) (hi : X.NonzeroIndexPlace i) (j : ℕ) (hj : j ∈ Finset.Icc 1 D.R.m) (b : Bool) (k P) (hk : HkIdx i k P) : (Y.Bsup i j b).map (hkO Y i j b k P) ≤ Y.Bsup i j b

theorem **integral_of_univBdd** {E : ExpDat Y} (h : UnivBdd T E) (i : X.Tw) (hi : X.NonzeroIndexPlace i) (j : ℕ) (hj : j ∈ Finset.Icc 1 D.R.m) (b : Bool) : ∀ a ∈ E.Dat0 i j b, ∃ c : ℕ, ∀ Q M, (∀ P ∈ Q, Formalism.BasicNotions.IsSplitPl D.host P) → ClsQ (D

theorem **span_bsup_of_supBdd** (h : SupBdd T) (i : X.Tw) (hi : X.NonzeroIndexPlace i) (j : ℕ) (hj : j ∈ Finset.Icc 1 D.R.m) (b : Bool) : Submodule.span (FractionRing (X.O i)) (Y.Bsup i j b : Set (Y.Fm i j b)) = ⊤

theorem **typePreserved_of_admissibleIntegral** {E : ExpDat Y} (h : IsotypicPartsInternalDirectSum Y E) : TypeKept E

theorem **coeffs_of_range** {O V Δ : Type} [CommRing O] [AddCommGroup V] [Module O V] [AddCommMonoid Δ] (S : Submodule O V) (t : S ⊗[O] AddMonoidAlgebra O Δ) : ∃ Xi : Δ →₀ V, S.subtype.rTensor _ t = Xi.sum (fun δ f => f ⊗ₜ[O] AddMonoidAlgebra.single δ (1 : O)) ∧ ∀ δ, Xi δ ∈ S

theorem **l_ne_zero** (T : NTh D.R D.host D.what₂ D.LL0 N X Y) (i : X.Tw) : ((X.l i : FractionRing (X.O i))) ≠ 0

theorem **inv_smul_span** (i : X.Tw) (j : ℕ) (b : Bool) (c : ℕ) (f : Y.Fm i j b) (hf : f ∈ Y.Bsup i j b) : ((X.l i : FractionRing (X.O i)) ^ c)⁻¹ • f ∈ Submodule.span (X.O i) (Ball T i j b ((X.l i : ℝ) ^ c))

theorem **uniformBound_of_admissibleIntegral** {E : ExpDat Y} (h : IsotypicPartsInternalDirectSum Y E) : UnivBdd T E

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.UnitaryFinitePointsToGLFiniteAdelesIsUnitary

theorem **kappa_tmul** (x : L) (a : FiniteAdeleRing (𝒪 F₀) F₀) : kappa L (x ⊗ₜ a) = GlobalTheory.AdeleBaseChange.adMap F₀ L a * algebraMap L _ x

theorem **kappa_conj** (y : AL L) : kappa L (conjAL L y) = conjFin (IsCMField.complexConj L).toRingEquiv (kappa L y)

theorem **map_conjT** (M : Matrix (Fin n) (Fin n) (AL L)) : (conjT (conjAL L) M).map (kappa L).toRingHom = Formalism.UG.cT (conjFin (IsCMField.complexConj L).toRingEquiv) (M.map (kappa L).toRingHom)

theorem **unitaryFinitePointsToGLFiniteAdeles_isUnitary** (x : GAf L n) : Formalism.UG.cT (conjFin (IsCMField.complexConj L).toRingEquiv) (psi0 L n x).val * (psi0 L n x).val = 1

theorem **kappa_eq** (F G : FinIso L n) : F.κ = G.κ

theorem **j_eq** (F G : FinIso L n) : F.j = G.j

theorem **j_injective** (F : FinIso L n) : Function.Injective F.j

theorem **dict_triv** (F : FinIso L n) : Dict F (trivRep L n) (Submodule.span ℂ {(fun _ => 1 : UA L n → ℂ)})

theorem **not_dict_bot** (F : FinIso L n) (π : NG.AutRep L n) : ¬ Dict F π ⊥

theorem **restrict_translate** (F : FinIso L n) (φ : UA L n → ℂ) (h : GAf L n) : LinearMap.funLeft ℂ ℂ F.j (RU L n (F.j h) φ) = NG.rt L n h (LinearMap.funLeft ℂ ℂ F.j φ)

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.UnivCoefNeZero

theorem **hecke_mul** {χ χ' : (AE D)ˣ →* Circle} (h : IsHeckeChar D χ) (h' : IsHeckeChar D χ') : IsHeckeChar D (χ * χ')

theorem **hecke_inv** {χ : (AE D)ˣ →* Circle} (h : IsHeckeChar D χ) : IsHeckeChar D χ⁻¹

theorem **exists_isWeil** (hP : GlobalTheory.Published.WeilRepresentationUnitaryPair) {B : Base S T} (hB : B.Good) {ψ : Twist D Circle} (hψ : ψ.Good) : ∃ ω, B.IsWeil ψ.chi ω

theorem **not_isWeil_swap** {B : Base S T} (hB : B.Good) (ω : PairRep S T) : ¬ IsWeilDatumO B.F.ψ S T (B.mu * B.eta⁻¹) B.mu ω

theorem **posDef_of_good** {F : Frame S T} (hF : F.Good) (τ : D.what₂.Tau) : (gramAt D S τ).PosDef

theorem **exists_lift** (V : HSp D) (u : Bridge.Gf V) : ∃ k : GA D V, k ∈ GfC V ∧ (finHom V).rangeRestrict k = u

theorem **exists_isClassSystem** (B : Base S T) (hF : B.F.Good) (ψ : Twist D Circle) : ∃ (m : ℕ) (h : Fin m → GA D S), IsClassSystem S (B.levJ ψ) h

theorem **pair_int** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (u : Fin n → E) (hu : ∀ y ∈ Λ, ∃ c : 𝒪 E, algebraMap (𝒪 E) E c = ∑ k, u k * y k) (z : Fin n → E ⊗[F] Formalism.L2.Herm.Af F) (hz : z ∈ Formalism.L2.Herm.hatL (F

theorem **ct_mul** {a b c : ℕ} (A : Matrix (Fin a) (Fin b) (EAf D)) (B : Matrix (Fin b) (Fin c) (EAf D)) : ct (A * B) = ct B * ct A

theorem **ct_map** {a b : ℕ} (A : Matrix (Fin a) (Fin b) D.host.E) : ct (A.map (eToEAf D)) = Aᴴ.map (eToEAf D)

theorem **int_ct** {a b : ℕ} {A : Matrix (Fin a) (Fin b) (EAf D)} (h : IsIntM A) : IsIntM (ct A)

theorem **int_mul** {a b c : ℕ} {A : Matrix (Fin a) (Fin b) (EAf D)} {B : Matrix (Fin b) (Fin c) (EAf D)} (h : IsIntM A) (h' : IsIntM B) : IsIntM (A * B)

theorem **gram_frame_int** (V : HSp D) {Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)} {β : GL (Fin V.rank) (EAf D)} (hV : IsSelfDualC V Λ) (f : IsFrame β Λ) : IsIntM (ct β.val * V.gram.map (eToEAf D) * β.val) ∧ IsIntM ((β⁻¹).val * (V.gram⁻¹).map (eToEAf D) * ct (β⁻¹).val)

theorem **int_adj_of_int** {F : Frame S T} (hF : F.Good) {x : WO S T} (h : IsIntM (frMapO S T F.βS F.βT x)) : IsIntM (frMapAdj S T F.βS F.βT x)

theorem **vac_ne_zero** {F : Frame S T} (hF : F.Good) : F.f₀ ≠ 0

theorem **omegaG_empty** (y : Reds D S.rank T.rank) : omegaG (TypeDat.empty D T.rank).levT (TypeDat.empty D T.rank).muC y = 1

theorem **flat_rectOne** {k : Type} [CommRing k] {r n : ℕ} (h : r ≤ n) (ν : kˣ →* ℂˣ) : flat ν (rectOne k r n) = 1

theorem **vecFlat_ne_zero** {B : Base S T} (hB : B.Good) (hG : B.G = TypeDat.empty D T.rank) (ψ : Twist D Circle) (hQ : B.admitted ψ.Q) : B.vecFlat ψ ≠ 0

theorem **exists_split_off** (D : D7) (R : Finset (Pl D.host.E)) : ∃ (P : Pl D.host.E) (p : ℕ), SplitSet {P} ∧ (p : 𝒪 D.host.E) ∈ P.asIdeal ∧ ∀ Q ∈ R, (p : 𝒪 D.host.E) ∉ Q.asIdeal

theorem **exists_placesOK** (D : D7) : ∃ P : Pl D.host.E, PlacesOK ∅ {P}

theorem **exists_model** (S T : HSp D) (hS : S.SigOK) (hpos : S.IsPosDef) (hT : T.SigOK) (hr : T.rank ≤ S.rank) (he : Even S.rank) (ho : Odd T.rank) (hsum : ∀ τ, (T.sig (tauE D τ)).1 + (T.sig (tauE D τ)).2 = T.rank) : ∃ (B : Base S T) (ψ : Twist D Circle), B.Good ∧ ψ.Good ∧ ψ.Q.Nonempty ∧ B.admitted ψ.Q ∧ B.vecFlat ψ ≠ 0 ∧ (∃ (m : ℕ) (h : Fin m → GA D S), IsClassSystem S (B.levJ ψ) h) ∧ (GlobalTheory.Published.WeilRepresentationUnitaryPair → ∃ ω ω', B.IsWeil 1 ω ∧ B.IsWeil ψ.chi ω')

theorem **univCoef_ne_zero** {k C : Type} [Field k] [Finite k] [AddCommGroup C] (β : DetBases k 1 2) (κ : kˣ →* Multiplicative C) : ∃ c, univCoef β κ c (!![1, 0] : Matrix (Fin 1) (Fin 2) k) ≠ 0

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.UnivOfNotLinearIndependent

theorem **flat_of_isUnit** (ν : kˣ →* Rˣ) {y : Matrix (Fin r) (Fin n) k} {u : kˣ} (h : (topL r y).det = u) : flat ν y = ((ν u : Rˣ) : R)

theorem **flat_of_not_isUnit** (ν : kˣ →* Rˣ) {y : Matrix (Fin r) (Fin n) k} (h : ¬ IsUnit (topL r y).det) : flat ν y = 0

theorem **topL_mul** (hr : r ≤ n) (g : Matrix (Fin r) (Fin r) k) (y : Matrix (Fin r) (Fin n) k) : topL r (g * y) = g * topL r y

theorem **flat_mul** (hr : r ≤ n) (ν : kˣ →* Rˣ) (g : GL (Fin r) k) (y : Matrix (Fin r) (Fin n) k) : flat ν ((g : Matrix (Fin r) (Fin r) k) * y) = ((ν (GeneralLinearGroup.det g) : Rˣ) : R) * flat ν y

theorem **finrank_quot** (U : Planes k r n) : Module.finrank k ((Fin n → k) ⧸ U.1) = r

theorem **dU_ne_zero_iff** (β : DetBases k r n) (U : Planes k r n) (y : Matrix (Fin r) (Fin n) k) : dU β U y ≠ 0 ↔ Compl U.1 y

theorem **dU_mul** (β : DetBases k r n) (U : Planes k r n) (g : Matrix (Fin r) (Fin r) k) (y : Matrix (Fin r) (Fin n) k) : dU β U (g * y) = g.det * dU β U y

theorem **univCoef_eq** (β : DetBases k r n) (κ : kˣ →* Multiplicative C) (c : C) (y : Matrix (Fin r) (Fin n) k) : univCoef β κ c y = ((Nat.card k : ℚ) ^ (r * (n - r)))⁻¹ * Nat.card {U : Planes k r n // ∃ u : kˣ, (u : k) = dU β U y ∧ Multiplicative.toAdd (κ u) = c}

theorem **isAdmissible_univCoef** (β : DetBases k r n) (κ : kˣ →* Multiplicative C) : IsAdmissible κ (univCoef β κ) where

theorem **univ_of_not_linearIndependent** (β : DetBases k r n) (κ : kˣ →* Multiplicative C) {y : Matrix (Fin r) (Fin n) k} (h : ¬ LinearIndependent k y.row) : univ β κ y = 0

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.UnivSupOfIsotypicPartsInternalDirectSum

theorem **supG_of_gKeeps** (h : GKeeps D.nth) : SupG D

theorem **supHk_of_hkKeeps** (h : HkKeeps D.nth) : SupHk D

theorem **univSup_iff** (E : XD D) : UnivSup D E ↔ UnivBdd D.nth E

theorem **univSup_of_isotypicPartsInternalDirectSum** (E : XD D) (h : IsotypicPartsInternalDirectSum D.nexp E) : UnivSup D E

theorem **heckeLaws_of** (h1 : HkComm D) (h2 : OffRange D) : GlobalTheory.TwSys.HeckeLaws D.nexp

theorem **mem_intF** (i : D.ntw.Tw) (j : ℕ) (f : Pt D j → KK D i) : f ∈ LinearMap.range (intF D i j) ↔ ∀ y, f y ∈ (algebraMap (D.ntw.O i) (KK D i)).range

theorem **goodFiltration_eq_top** (i : D.ntw.Tw) (j : ℕ) (b : Bool) : Fm0 (D

theorem **inNonzeroPlaces_recursive_iff** (i : D.ntw.Tw) : (rec D A s).ntw.NonzeroIndexPlace i ↔ D.ntw.NonzeroIndexPlace i

theorem **nontrivial_fm** (i : D.ntw.Tw) (j : ℕ) (hj : j ∈ Finset.Icc 1 D.R.m) (b : Bool) : Nontrivial ((rec D A s).nexp.Fm i j b)

theorem **one_mem_range** : 1 ∈ Finset.Icc 1 D.R.m

theorem **supG** : SupG (rec D A s)

theorem **exists_pow** (i : D.ntw.Tw) (j : ℕ) (f : Pt D j → KK D i) : ∃ c : ℕ, ((D.ntw.l i : D.ntw.O i) ^ c) • f ∈ LinearMap.range (intF D i j)

theorem **supFull** : SupFull (rec D A s)

theorem **offRange** : OffRange (rec D A s)

theorem **univSup_one** : UnivSup (rec D A fun _ _ => 1) (dat D A fun _ _ => 1)

theorem **hkComm_scal** (t) : HkComm (rec D (scal D t) s)

theorem **hkSS_scal** (t) : HkSS (rec D (scal D t) s)

theorem **supHk_nat** : SupHk (rec D (scal D fun _ k => k) s)

theorem **all_laws** : ∃ (D' : D7) (E : XD D') (e : D'.ntw.Tw ≃ D.ntw.Tw), (∀ i, D'.ntw.NonzeroIndexPlace i ↔ D.ntw.NonzeroIndexPlace (e i)) ∧ (∀ i, ∀ j ∈ Finset.Icc 1 D'.R.m, ∀ b, Nontrivial (D'.nexp.Fm i j b) ∧ E.Dat0 i j b = Set.univ ∧ Function.Surjective (D'.nexp.th i j b)) ∧ SupG D' ∧ SupFull D' ∧ SupHk D' ∧ UnivSup D' E ∧ HkComm D' ∧ OffRange D' ∧ HkSS D'

theorem **inv_l_not_mem** (i : D.ntw.Tw) : ((D.ntw.l i : KK D i))⁻¹ ∉ (algebraMap (D.ntw.O i) (KK D i)).range

theorem **not_supHk** (i : D.ntw.Tw) (hi : D.ntw.NonzeroIndexPlace i) (k P) (hk : HkIdx (D

theorem **not_univSup** (i : D.ntw.Tw) (hi : D.ntw.NonzeroIndexPlace i) : ¬ UnivSup (rec D A fun i M => ((D.ntw.l i : KK D i) ^ M)⁻¹) (dat D A _)

theorem **goodPart_ne_bot** (i : D.ntw.Tw) (j : ℕ) (hj : j ∈ Finset.Icc 1 D.R.m) (b : Bool) (he : D.nexp.eAlg i b ≠ 0) : V0 (D

theorem **not_hkComm** (i : D.ntw.Tw) (hi : D.ntw.NonzeroIndexPlace i) (P) (hP : HkIdx (D

theorem **not_hkSS** [Nontrivial (G0g D.nd)] (i : D.ntw.Tw) (hi : D.ntw.NonzeroIndexPlace i) (k P) (hP : HkIdx (D

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.UnramifiedSubfieldLeCyclotomicExt

theorem **_root_.GlobalTheory.NL.Model.conv_finite** (e : PowerSeries L) {u : ℤ → M} {m : ℤ} (hm : ∀ k < m, u k = 0) (k : ℤ) : (Function.support fun j : ℕ => PowerSeries.coeff j e • u (k - (j : ℤ))).Finite

theorem **_root_.GlobalTheory.NL.Model.conv_one** (u : ℤ → M) : _root_.GlobalTheory.NL.conv (1 : PowerSeries L) u = u

theorem **conv_X** (u : ℤ → M) (k : ℤ) : conv (PowerSeries.X : PowerSeries L) u k = u (k - 1)

theorem **conv_single** (e : PowerSeries L) (a : ℤ) (d : M) (k : ℤ) : conv e (Pi.single a d : ℤ → M) k = if a ≤ k then PowerSeries.coeff (k - a).toNat e • d else 0

theorem **_root_.GlobalTheory.NL.Model.mem_latSpan** {ι : Type} (ex : ι → ℤ → M) (x : ι) : ex x ∈ _root_.GlobalTheory.NL.latSpan (LL

private theorem **conv_mem_and_vanish_below** (e : PowerSeries L) (D : Submodule L M) {u : ℤ → M} (hu : ∀ k, u k ∈ D) {m : ℤ} (hm : ∀ k < m, u k = 0) : (∀ k, _root_.GlobalTheory.NL.conv e u k ∈ D) ∧ ∀ k < m, _root_.GlobalTheory.NL.conv e u k = 0

theorem **_root_.GlobalTheory.NL.Model.latSpan_coeff** {ι : Type} (ex : ι → ℤ → M) (D : Submodule L M) (hD : ∀ x k, ex x k ∈ D) (hs : ∀ x, ∃ m : ℤ, ∀ k < m, ex x k = 0) (f : ℤ → M) (hf : f ∈ _root_.GlobalTheory.NL.latSpan (LL

theorem **mem_latSpan_single** (a : ℤ) (f : ℤ → L) : f ∈ latSpan (LL

theorem **Std.full** {D : Submodule L M} {Λ : Submodule L (ℤ → M)} (h : Std D Λ) : Full D Λ

theorem **Std.graded** {D : Submodule L M} {Λ : Submodule L (ℤ → M)} (h : Std D Λ) : Graded Λ

theorem **line_tests** (a : ℤ) : Full (⊤ : Submodule L L) (latSpan (LL

theorem **not_graded** : ¬ Graded (latSpan (LL

theorem **_root_.GlobalTheory.NL.Model.mem_van** (W : _root_.Formalism.BasicNotions.LocWach PD LL O v n) (i : ZMod n) (ζ : LL) (k : ℕ) (a : _root_.Formalism.BasicNotions.LetA O n) : a ∈ _root_.GlobalTheory.NL.van W i ζ k ↔ ∀ j < k, PowerSeries.coeff j (W.lev i ζ a) = 0

theorem **_root_.GlobalTheory.NL.Model.pi_van_one** (W : _root_.Formalism.BasicNotions.LocWach PD LL O v n) (i : ZMod n) : (_root_.Formalism.BasicNotions.letπ : _root_.Formalism.BasicNotions.LetA O n) ∈ _root_.GlobalTheory.NL.van W i 1 1 ∧ (_root_.Formalism.BasicNotions.letπ : _root_.Formalism.BasicNotions.LetA O n) ∉ _root_.GlobalTheory.NL.van W i 1 2

theorem **pi_not_van** (W : LocWach PD LL O v n) (i : ZMod n) (ζ : LL) (j : ℕ) (hζ : IsPrimitiveRoot ζ (PD.p ^ j)) (h1 : ζ ≠ 1) : (letπ : LetA O n) ∉ van W i ζ 1

theorem **_root_.GlobalTheory.NL.Model.FilAt_anti** (T : _root_.Formalism.BasicNotions.LocLat G Qp LL O) (τ : K →ₐ[Qp] LL) (a : ℤ) : _root_.GlobalTheory.NL.FilAt PD T τ (a + 1) ≤ _root_.GlobalTheory.NL.FilAt PD T τ a

theorem **wachExpansion_smul_compatible** (W : LocWach PD LL O v n) (T : LocLat G Qp LL O) (i : ZMod n) (ζ : LL) (e : ɪ(W.N T) → ℤ → T.V ⊗[Qp] B) (hs : ∀ x, ∃ m : ℤ, ∀ k < m, e x k = 0) (he : ∀ x k, W.levN T i ζ x k = ιV PD (LL

theorem **ext_aux** {k E E' Ω : Type} [Field k] [Field E] [Field E'] [Field Ω] [Algebra k E] [Algebra k E'] [Algebra k Ω] [IsAlgClosed Ω] [Algebra.IsAlgebraic k E'] (τ : E →ₐ[k] E') (j : E →ₐ[k] Ω) : ∃ σ : E' →ₐ[k] Ω, ∀ y, σ (τ y) = j y

theorem **_root_.GlobalTheory.NL.Model.ex_mem_Lat** (C : _root_.Formalism.BasicNotions.LocCmpT l P AI H F) (h : _root_.Formalism.BasicNotions.Loc.Good K s LL) (T : _root_.GlobalTheory.NL.Lt K LL) (i : _root_.GlobalTheory.NL.Let K) (ζ : ɪLL.1) (x : ɪ((H.W K s LL h).N T)) : _root_.GlobalTheory.NL.ex C h T i ζ x ∈ _root_.GlobalTheory.NL.Lat C h T i ζ

theorem **_root_.GlobalTheory.NL.Model.Lat_coeff** (C : _root_.Formalism.BasicNotions.LocCmpT l P AI H F) (h : _root_.Formalism.BasicNotions.Loc.Good K s LL) (T : _root_.GlobalTheory.NL.Lt K LL) (i : _root_.GlobalTheory.NL.Let K) (ζ : ɪLL.1) (f : ℤ → T.V ⊗[ℚ_[l]] P.R.B) (hf : f ∈ _root_.GlobalTheory.NL.Lat C h T i ζ) : (∀ k, f k ∈ _root_.GlobalTheory.NL.Di H h T i) ∧ ∃ m : ℤ, ∀ k < m, f k = 0

private theorem **periodTensor_ιV_injective** (T : _root_.Formalism.BasicNotions.LocLat G Qp LL O) : Function.Injective (_root_.Formalism.WeaklyAdmissibleFiltrations.ιV PD : T.V ⊗[Qp] B →ₗ[LL] T.V ⊗[Qp] BdR)

private theorem **levB_smul_eq_powerSeriesSmul_lemma_2** {Zl Ai : Type} [CommRing Zl] [CommRing Ai] [Algebra Zl O] [Algebra Zl Ai] {W : _root_.Formalism.BasicNotions.LocWach PD LL O v n} {RW : Type} [CommRing RW] [Algebra O RW] {inclR : TensorProduct Zl O Ai →ₐ[O] RW} {φR : RW →ₐ[O] RW} {gR : G → RW →ₐ[O] RW} {πA : TensorProduct Zl O Ai} {E : _root_.Formalism.BasicNotions.EtTie W (TensorProduct Zl O Ai) RW inclR φR gR πA} {ιdR : Ai →+* BdR} {p1 : Ai} {tS zS : BdR} (S : _root_.Formalism.BasicNotions.CmpTie W E ιdR p1 tS zS) (T : _root_.Formalism.BasicNotions.LocLat G Qp LL O) (i : ZMod n) (ζ : LL) (a : _root_.Formalism.BasicNotions.LetA O n) (x : ɪ(W.N T)) : S.levB T i ζ (a • x) = _root_.GlobalTheory.NL.conv (W.lev i ζ a) (S.levB T i ζ x)

theorem **_root_.GlobalTheory.NL.Model.ex_smul** (C : _root_.Formalism.BasicNotions.LocCmpT l P AI H F) (h : _root_.Formalism.BasicNotions.Loc.Good K s LL) (T : _root_.GlobalTheory.NL.Lt K LL) (i : _root_.GlobalTheory.NL.Let K) (ζ : ɪLL.1) (a : _root_.GlobalTheory.NL.AA K LL) (x : ɪ((H.W K s LL h).N T)) : _root_.GlobalTheory.NL.ex C h T i ζ (a • x) = _root_.GlobalTheory.NL.conv ((H.W K s LL h).lev i ζ a) (_root_.GlobalTheory.NL.ex C h T i ζ x)

theorem **zetaQ_pow** (P : LocPrime l) (s : ℕ) : P.ζQ s ^ (l ^ s) = 1

theorem **zetaQ_mem** (P : LocPrime l) (K : FinExt l) (s : ℕ) : P.ζQ s ∈ Ks K s

theorem **unramifiedSubfield_le_cyclotomicExt** (K : FinExt l) (s : ℕ) : K0 K ≤ Ks K s

theorem **comp_of_pt** (h : Good K s LL) (i : Let K) (ζ : ɪLL.1) (τ : ɪK.1 →ₐ[ℚ_[l]] ɪLL.1) (hp : Pt H h i ζ τ) : ∃ ι : ɪLL.1 →ₐ[ℚ_[l]] P.R.BdR, Comp H h i ζ τ ι

theorem **exists_pt** (h : Good K s LL) (i : Let K) : ∃ (ζ : ɪLL.1) (τ : ɪK.1 →ₐ[ℚ_[l]] ɪLL.1), Pt H h i ζ τ

theorem **not_pt_one** (h : Good K s LL) (i : Let K) (τ : ɪK.1 →ₐ[ℚ_[l]] ɪLL.1) : ¬ Pt H h i 1 τ

theorem **comp_tie** (C : LocCmpT l P AI H F) (h : Good K s LL) (T : Lt K LL) (hT : IsCrys (P.PD K) T.ρ) (i : Let K) (ζ : ɪLL.1) (τ : ɪK.1 →ₐ[ℚ_[l]] ɪLL.1) (ι : ɪLL.1 →ₐ[ℚ_[l]] P.R.BdR) (hι : Comp H h i ζ τ ι) (x : ɪ((H.W K s LL h).N T)) (k : ℕ) (y : TW (RA LL) T) (hy : TWincl (inclR LL) T y = (inclR LL ((1 : ɪ(OL LL)) ⊗ₜ[ℤ_[l]] AI.π₁)) ^ k • (F.E K s LL h).tie.jD T (x : (H.W K s LL h).DD T)) (a : ℕ) (ha : ∀ r : ℤ, r < -(a : ℤ) → (H.W K s LL h).levN T i ζ x r = 0) : ∃ c : ℤ, ∀ M : ℕ, mulVR T (tS P s ^ a) ((C.C K s LL h).tie.βdR T y) - mulVR T ((P.ιB (AI.ιA AI.π₁)) ^ k) (∑ r ∈ Finset.range M, mulVR T (tS P s ^ r) ((H.W K s LL h).levN T i ζ x ((r : ℤ) - (a : ℤ)))) ∈ compKer T ι ⊔ FilV (P.PD K) (LL

Development.AbelianVarietyTheory.Definitions.NonGaloisDescent.ValIrreducible

private theorem **sigma_smul_eq_conj_smul** (e : K.E) (v : B.A) : B.σ (e • v) = K.c e • B.σ v

theorem **_root_.GlobalTheory.UnitLemma.Basic.trace_sigma** (u : B.A) : Algebra.trace K.E B.A (B.σ u) = K.c (Algebra.trace K.E B.A u)

theorem **_root_.GlobalTheory.UnitLemma.Basic.herm_smul_left** (x y z : B.A) (e : K.E) : B.herm x (e • y) z = e * B.herm x y z

theorem **_root_.GlobalTheory.UnitLemma.Basic.herm_smul_right** (x y z : B.A) (e : K.E) : B.herm x y (e • z) = K.c e * B.herm x y z

theorem **_root_.GlobalTheory.UnitLemma.Basic.herm_symm** (x : B.A) (hx : B.σ x = x) (y z : B.A) : B.herm x z y = K.c (B.herm x y z)

theorem **_root_.GlobalTheory.UnitLemma.Basic.herm_mul_left** (x u y z : B.A) : B.herm x (u * y) z = B.herm x y (B.σ u * z)

theorem **_root_.GlobalTheory.UnitLemma.Basic.herm_norm** (x u y z : B.A) : B.herm (x * (u * B.σ u)) y z = B.herm x (u * y) (u * z)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.AlgebraMapBaseMemMaximalIdealIff

theorem **mem_gaussIdeal_iff** {f : MvPolynomial σ O} : f ∈ gaussIdeal p σ ↔ ∀ m, f.coeff m ∈ p

theorem **C_mem_gaussIdeal_iff** {a : O} : (C a : MvPolynomial σ O) ∈ gaussIdeal p σ ↔ a ∈ p

theorem **comap_C_gaussIdeal** : (gaussIdeal p σ).comap (C : O →+* MvPolynomial σ O) = p

@[simp] theorem **quotientGaussIdealEquiv_mk** (f : MvPolynomial σ O) : quotientGaussIdealEquiv p σ (Ideal.Quotient.mk _ f) = MvPolynomial.map (Ideal.Quotient.mk p) f

theorem **X_notMem_gaussIdeal** [hp : p.IsPrime] (i : σ) : (X i : MvPolynomial σ O) ∉ gaussIdeal p σ

theorem **comap_map_gaussIdeal** {O₂ : Type*} [CommRing O₂] (φ : O →+* O₂) (q : Ideal O₂) (h : p = q.comap φ) : gaussIdeal p σ = (gaussIdeal q σ).comap (MvPolynomial.map φ)

theorem **algebraMap_C** (a : O) : algebraMap (MvPolynomial σ O) (GaussLocal O p σ) (C a) = algebraMap O (GaussLocal O p σ) a

theorem **algebraMap_base_mem_maximalIdeal_iff** (a : O) : algebraMap O (GaussLocal O p σ) a ∈ maximalIdeal (GaussLocal O p σ) ↔ a ∈ p

theorem **isUnit_algebraMap_base_iff** (a : O) : IsUnit (algebraMap O (GaussLocal O p σ) a) ↔ a ∉ p

theorem **isUnit_X** (i : σ) : IsUnit (algebraMap (MvPolynomial σ O) (GaussLocal O p σ) (X i))

theorem **map_eq_maximalIdeal** : p.map (algebraMap O (GaussLocal O p σ)) = maximalIdeal (GaussLocal O p σ)

theorem **maximalIdeal_eq_span** {a : O} (h : p = Ideal.span {a}) : maximalIdeal (GaussLocal O p σ) = Ideal.span {algebraMap O (GaussLocal O p σ) a}

theorem **algebraMap_injective** : Function.Injective (algebraMap (MvPolynomial σ O) (GaussLocal O p σ))

@[simp] theorem **reduction_algebraMap** (f : MvPolynomial σ O) : reduction σ φ hφ (algebraMap _ _ f) = algebraMap _ _ (MvPolynomial.map φ f)

theorem **ker_reduction** : RingHom.ker (reduction σ φ hφ) = maximalIdeal (GaussLocal O p σ)

theorem **lift_reduction_bijective** (hk : ∀ c : k, ∃ a b : O, c = φ a / φ b) : Function.Bijective (ResidueField.lift (reduction σ φ hφ))

theorem **residueField_eq_div** (c : p.ResidueField) : ∃ a b : O, c = algebraMap O p.ResidueField a / algebraMap O p.ResidueField b

@[simp] theorem **residueFieldEquiv_residue** (f : MvPolynomial σ O) : residueFieldEquiv p σ (residue _ (algebraMap _ _ f)) = algebraMap _ _ (MvPolynomial.map (algebraMap O p.ResidueField) f)

@[simp] theorem **residueFieldEquiv'_residue** [IsLocalRing O] (f : MvPolynomial σ O) : residueFieldEquiv' O σ (residue _ (algebraMap _ _ f)) = algebraMap _ _ (MvPolynomial.map (residue O) f)

@[simp] theorem **map_algebraMap** (φ : O →+* O₂) (h : p = q.comap φ) (f : MvPolynomial σ O) : map σ φ h (algebraMap _ _ f) = algebraMap _ _ (MvPolynomial.map φ f)

@[simp] theorem **map_algebraMap_base** (φ : O →+* O₂) (h : p = q.comap φ) (a : O) : map σ φ h (algebraMap O _ a) = algebraMap O₂ _ (φ a)

@[simp] theorem **map_id** : map σ (RingHom.id O) (Ideal.comap_id p).symm = RingHom.id _

theorem **map_bijective_of_isLocalization** : Function.Bijective (map σ (algebraMap O A) (IsLocalization.AtPrime.under_maximalIdeal A p).symm : GaussLocal O p σ →+* GaussLocal' A σ)

theorem **mapLocalization_toRingHom** : (mapLocalization p σ A : GaussLocal O p σ →+* GaussLocal' A σ) = map σ (algebraMap O A) (IsLocalization.AtPrime.under_maximalIdeal A p).symm

@[simp] theorem **equivOfIsLocalization_algebraMap** (f : MvPolynomial σ O) : equivOfIsLocalization p σ A (algebraMap _ _ f) = algebraMap _ _ (MvPolynomial.map (algebraMap O A) f)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.BijectiveSmulChartSec

@[simp] theorem **resG_resG** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (a : Γ(X, U)) : resG X h₂ (resG X h₁ a) = resG X (h₂.trans h₁) a

@[simp] theorem **resG_self** {U : X.Opens} (h : U ≤ U) (a : Γ(X, U)) : resG X h a = a

theorem **val_res** {U V : X.Opens} (h : V ≤ U) (a : Γ(X, U)ˣ) : ((res h a : Γ(X, V)ˣ) : Γ(X, V)) = resG X h (a : Γ(X, U))

theorem **fun_ext_of_cover** {ι : Type*} {V : X.Opens} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) {a b : Γ(X, V)} (h : ∀ k, resG X (hW k) a = resG X (hW k) b) : a = b

theorem **exists_fun_of_cover** {ι : Type*} {V : X.Opens} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) (a : ∀ k, Γ(X, W k)) (h : ∀ k l, resG X (inf_le_left : W k ⊓ W l ≤ W k) (a k) = resG X (inf_le_right : W k ⊓ W l ≤ W l) (a l)) : ∃ b : Γ(X, V), ∀ k, resG X (hW k) b = a k

theorem **le_iSup_inf** {ι : Type*} {U : ι → X.Opens} (hU : ⊔ i, U i = ⊤) (V : X.Opens) : V ≤ ⊔ i, V ⊓ U i

@[simp] theorem **homOfLinear_app** {F G : X.Modules} (φ : ∀ V : X.Opens, Γ(F, V) →ₗ[Γ(X, V)] Γ(G, V)) (hφ : ∀ {V W : X.Opens} (h : W ≤ V) (s : Γ(F, V)), φ W (F.presheaf.map (homOfLE h).op s) = G.presheaf.map (homOfLE h).op (φ V s)) (V : X.Opens) (s : Γ(F, V)) : (homOfLinear φ hφ).app V s = φ V s

theorem **secOn_rel** {V : X.Opens} {s : ∀ i, Γ(X, V ⊓ L.U i)} (hs : s ∈ secOn L V) (i j : L.ι) {W : X.Opens} (hi : W ≤ V ⊓ L.U i) (hj : W ≤ V ⊓ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resG X hi (s i) = resG X hij (L.g i j : Γ(X, L.U i ⊓ L.U j)) * resG X hj (s j)

theorem **secRes_smul** {V W : X.Opens} (h : W ≤ V) (r : Γ(X, V)) (m : secGroup L V) : secRes L h (r • m) = resG X h r • secRes L h m

theorem **isSheaf_secPresheaf** : (secPresheaf L).IsSheaf

theorem **coord_mem** {V : X.Opens} (s : Γ(toMod L, V)) : coord s ∈ secOn L V

@[simp] theorem **coord_mkSec** {V : X.Opens} (f : ∀ i, Γ(X, V ⊓ L.U i)) (hf : f ∈ secOn L V) : coord (mkSec f hf) = f

@[ext] theorem **sec_ext** {V : X.Opens} {s t : Γ(toMod L, V)} (h : ∀ i, coord s i = coord t i) : s = t

@[simp] theorem **coord_zero** {V : X.Opens} (i : L.ι) : coord (0 : Γ(toMod L, V)) i = 0

@[simp] theorem **coord_add** {V : X.Opens} (s t : Γ(toMod L, V)) (i : L.ι) : coord (s + t) i = coord s i + coord t i

@[simp] theorem **coord_neg** {V : X.Opens} (s : Γ(toMod L, V)) (i : L.ι) : coord (-s) i = - coord s i

@[simp] theorem **coord_sub** {V : X.Opens} (s t : Γ(toMod L, V)) (i : L.ι) : coord (s - t) i = coord s i - coord t i

@[simp] theorem **coord_smul** {V : X.Opens} (a : Γ(X, V)) (s : Γ(toMod L, V)) (i : L.ι) : coord (a • s) i = resG X inf_le_left a * coord s i

@[simp] theorem **coord_map** {V W : X.Opens} (h : W ≤ V) (s : Γ(toMod L, V)) (i : L.ι) : coord ((toMod L).presheaf.map (homOfLE h).op s) i = resG X (inf_le_inf_right _ h) (coord s i)

theorem **coord_rel** {V : X.Opens} (s : Γ(toMod L, V)) (i j : L.ι) {W : X.Opens} (hi : W ≤ V ⊓ L.U i) (hj : W ≤ V ⊓ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resG X hi (coord s i) = resG X hij (L.g i j : Γ(X, L.U i ⊓ L.U j)) * resG X hj (coord s j)

theorem **resG_g_self** (hL : L.IsLB) (i : L.ι) {W : X.Opens} (h : W ≤ L.U i ⊓ L.U i) : resG X h (L.g i i : Γ(X, L.U i ⊓ L.U i)) = 1

theorem **resG_cocycle** (hL : L.IsLB) (i j k : L.ι) {W : X.Opens} (h₁ : W ≤ L.U i ⊓ L.U j) (h₂ : W ≤ L.U j ⊓ L.U k) (h₃ : W ≤ L.U i ⊓ L.U k) : resG X h₁ (L.g i j : Γ(X, L.U i ⊓ L.U j)) * resG X h₂ (L.g j k : Γ(X, L.U j ⊓ L.U k)) = resG X h₃ (L.g i k : Γ(X, L.U i ⊓ L.U k))

theorem **f_injective** : Function.Injective (LB.Sec.f : LB.Sec L → ∀ i, Γ(X, L.U i))

@[simp] theorem **zero_f** (i : L.ι) : (0 : LB.Sec L).f i = 0

@[simp] theorem **add_f** (s t : LB.Sec L) (i : L.ι) : (s + t).f i = s.f i + t.f i

@[simp] theorem **neg_f** (s : LB.Sec L) (i : L.ι) : (-s).f i = - s.f i

@[simp] theorem **sub_f** (s t : LB.Sec L) (i : L.ι) : (s - t).f i = s.f i - t.f i

@[simp] theorem **smul_f** (a : Γ(X, ⊤)) (s : LB.Sec L) (i : L.ι) : (a • s).f i = resG X le_top a * s.f i

theorem **sec_rel** (s : LB.Sec L) (i j : L.ι) {W : X.Opens} (hi : W ≤ L.U i) (hj : W ≤ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resG X hi (s.f i) = resG X hij (L.g i j : Γ(X, L.U i ⊓ L.U j)) ∗ resG X hj (s.f j)

@[simp] theorem **coord_ofSec** (s : LB.Sec L) (V : X.Opens) (i : L.ι) : coord (ofSec s V) i = resG X inf_le_right (s.f i)

@[simp] theorem **map_ofSec** (s : LB.Sec L) {V W : X.Opens} (h : W ≤ V) : (toMod L).presheaf.map (homOfLE h).op (ofSec s V) = ofSec s W

@[simp] theorem **globalEquiv_f** (s : Γ(toMod L, ⊤)) (i : L.ι) : (globalEquiv L s).f i = resG X (le_inf le_top le_rfl) (coord s i)

@[simp] theorem **globalEquiv_symm_apply** (s : LB.Sec L) : (globalEquiv L).symm s = ofSec s ⊤

theorem **globalEquivSecM_apply** (s : Γ(toMod (LPow A L₀ a T), ⊤)) : globalEquivSecM A L₀ a T s = SecM.ofSec A (globalEquiv _ s)

theorem **globalEquivSecM_smul** (r : Γ(T.left, ⊤)) (s : Γ(toMod (LPow A L₀ a T), ⊤)) : globalEquivSecM A L₀ a T ((show Γ((T ⊗ A).left, ⊤) from (fst T A).left.appTop r) • s) = r • globalEquivSecM A L₀ a T s

@[simp] theorem **coord_chartSec** (hL : L.IsLB) (i j : L.ι) : coord (chartSec hL i) j = resG X (le_inf inf_le_right inf_le_left) (L.g j i : Γ(X, L.U j ⊓ L.U i))

theorem **chartEquiv_apply** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (s : Γ(toMod L, V)) : chartEquiv hL i h s = resG X (le_inf le_rfl h) (coord s i)

theorem **chartEquiv_symm_apply** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (a : Γ(X, V)) : (chartEquiv hL i h).symm a = a • (toMod L).presheaf.map (homOfLE h).op (chartSec hL i)

theorem **chartEquiv_map** (hL : L.IsLB) (i : L.ι) {V W : X.Opens} (h : V ≤ L.U i) (h' : W ≤ V) (s : Γ(toMod L, V)) : chartEquiv hL i (h'.trans h) ((toMod L).presheaf.map (homOfLE h').op s) = resG X h' (chartEquiv hL i h s)

theorem **chartEquiv_change** (hL : L.IsLB) (i j : L.ι) {V : X.Opens} (hi : V ≤ L.U i) (hj : V ≤ L.U j) (s : Γ(toMod L, V)) : chartEquiv hL i hi s = resG X (le_inf hi hj) (L.g i j : Γ(X, L.U i ⊓ L.U j)) ∗ chartEquiv hL j hj s

theorem **bijective_smul_chartSec** (hL : L.IsLB) (i : L.ι) (V : X.Opens) (h : V ≤ L.U i) : Function.Bijective fun a : Γ(X, V) => a • (toMod L).presheaf.map (homOfLE h).op (chartSec hL i)

theorem **chartIso_hom_app** (hL : L.IsLB) (i : L.ι) (W : (L.U i).toScheme.Opens) (a : Γ(X, (L.U i).ι ''ᵁ W)) : ((chartIso hL i).hom.app W a : Γ(toMod L, (L.U i).ι ''ᵁ W)) = a • (toMod L).presheaf.map (homOfLE ((L.U i).ι_image_le W)).op (chartSec hL i)

theorem **isIso_ofEpi_π.{v**, u} {C : Type u} [Category.{v} C] {J : GrothendieckTopology C} {R : Sheaf J RingCat.{0}} [HasWeakSheafify J AddCommGrpCat.{0}] [J.WEqualsLocallyBijective AddCommGrpCat.{0}] {M : SheafOfModules.{0} R} {I : Type} (e : SheafOfModules.free I ≅ M) : IsIso ((SheafOfModules.free.generatingSections I).ofEpi e.hom).π

theorem **isLocallyFree_of_iso** {F : X.Modules} {ι : Type} {U : ι → X.Opens} (hU : ⨆ i, U i = ⊤) (e : ∀ i, SheafOfModules.free (R

theorem **isLocallyFree_toMod** (hL : L.IsLB) : (toMod L).IsLocallyFree

theorem **isQuasicoherent_toMod** (hL : L.IsLB) : (toMod L).IsQuasicoherent

theorem **isFiniteType_toMod** (hL : L.IsLB) : (toMod L).IsFiniteType

theorem **homCoord_rel** (w : LB.Sec (LB.div M L)) (p p' : M.ι) (q q' : L.ι) {W : X.Opens} (h : W ≤ M.U p ⊓ L.U q) (h' : W ≤ M.U p' ⊓ L.U q') (hp : W ≤ M.U p ⊓ M.U p') (hq : W ≤ L.U q ⊓ L.U q') : resG X h (homCoord w p q) ∗ resG X hq (L.g q q' : Γ(X, L.U q ⊓ L.U q')) = resG X hp (M.g p p' : Γ(X, M.U p ⊓ M.U p')) ∗ resG X h' (homCoord w p' q')

theorem **exists_mul_sec** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V : X.Opens} (s : Γ(toMod L, V)) : ∃ t : Γ(toMod M, V), ∀ (p : M.ι) (q : L.ι) (W : X.Opens) (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q), resG X hp (coord t p) = resG X h (homCoord w p q) ∗ resG X hq (coord s q)

theorem **sec_ext_of_cover** {ι : Type*} {U : ι → X.Opens} (hU : ⨆ i, U i = ⊤) {V : X.Opens} {t t' : Γ(toMod M, V)} (h : ∀ p q, resG X (inf_le_left : V ⊓ M.U p ⊓ U q ≤ V ⊓ M.U p) (coord t p) = resG X inf_le_left (coord t' p)) : t = t'

theorem **coord_mulFun** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V : X.Opens} (s : Γ(toMod L, V)) (p : M.ι) (q : L.ι) {W : X.Opens} (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q) : resG X hp (coord (mulFun hL hM w s) p) = resG X h (homCoord w p q) ∗ resG X hq (coord s q)

theorem **coord_mulFun'** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V : X.Opens} (s : Γ(toMod L, V)) (p : M.ι) (q : L.ι) : resG X (inf_le_left : V ⊓ M.U p ⊓ L.U q ≤ V ⊓ M.U p) (coord (mulFun hL hM w s) p) = resG X (inf_le_inf_right _ inf_le_right) (homCoord w p q) ∗ resG X (inf_le_inf_right _ inf_le_left) (coord s q)

theorem **mulFun_map** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V W : X.Opens} (h : W ≤ V) (s : Γ(toMod L, V)) : mulFun hL hM w ((toMod L).presheaf.map (homOfLE h).op s) = (toMod M).presheaf.map (homOfLE h).op (mulFun hL hM w s)

theorem **coord_homOfSec_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V : X.Opens} (s : Γ(toMod L, V)) (p : M.ι) (q : L.ι) {W : X.Opens} (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q) : resG X hp (coord ((homOfSec hL hM w).app V s) p) = resG X h (homCoord w p q) ∗ resG X hq (coord s q)

theorem **mulFun_mulFun** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) (w' : LB.Sec (LB.div L M)) (h : ∀ p q, homCoord w p q ∗ resG X (le_inf inf_le_right inf_le_left) (homCoord w' q p) = 1) {V : X.Opens} (t : Γ(toMod M, V)) : mulFun hL hM w (mulFun hM hL w' t) = t

theorem **coord_isoOfGen_hom_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) {V : X.Opens} (s : Γ(toMod L, V)) (p : M.ι) (q : L.ι) {W : X.Opens} (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q) : resG X hp (coord ((isoOfGen hL hM w hw).hom.app V s) p) = resG X h ((show Γ(X, M.U p ⊓ L.U q)ˣ from w.s (p, q)) : Γ(X, M.U p ⊓ L.U q)) ∗ resG X hq (coord s q)

theorem **nonempty_iso_of_lbIso** (hL : L.IsLB) (hM : M.IsLB) (h : LBIso L M) : Nonempty (toMod L ≅ toMod M)

theorem **isoOfGen_hom_app_genSec** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) {a : LB.Gen L} {b : LB.Gen M} (ha : a.IsGen) (hb : b.IsGen) (h : LB.Gen.Maps w a b) (V : X.Opens) : (isoOfGen hL hM w hw).hom.app V (ofSec (DVR.genSec a ha) V) = ofSec (DVR.genSec b hb) V

theorem **unitHom_app** (σ : LB.Sec L) (V : X.Opens) (a : Γ(X, V)) : (unitHom σ).app V a = a • ofSec σ V

theorem **bijective_smul_genSec** (hL : ⨆ i, L.U i = ⊤) {t : LB.Gen L} (ht : t.IsGen) (V : X.Opens) : Function.Bijective fun a : Γ(X, V) => a • ofSec (DVR.genSec t ht) V

theorem **unitIso_hom_app** (hL : L.IsLB) (t : LB.Gen L) (ht : t.IsGen) (V : X.Opens) (a : Γ(X, V)) : (unitIso hL t ht).hom.app V a = a • ofSec (DVR.genSec t ht) V

theorem **pull_U_le** (j : (L.pull f).ι) : (L.pull f).U j ≤ f ⁻¹ᵁ L.U (j PUnit.unit)

theorem **pull_U** (j : (L.pull f).ι) : (L.pull f).U j = f ⁻¹ᵁ L.U (j PUnit.unit)

theorem **appLE_resG** {U U' : X.Opens} (h : U' ≤ U) {W : Y.Opens} (e : W ≤ f ⁻¹ᵁ U') (a : Γ(X, U)) : (f.appLE U' W e).hom (resG X h a) = (f.appLE U W (e.trans (f.preimage_mono h))).hom a

@[simp] theorem **resG_appLE** {U : X.Opens} {W W' : Y.Opens} (e : W ≤ f ⁻¹ᵁ U) (h : W' ≤ W) (a : Γ(X, U)) : resG Y h ((f.appLE U W e).hom a) = (f.appLE U W' (h.trans e)).hom a

theorem **resG_app** {U : X.Opens} {W : Y.Opens} (e : W ≤ f ⁻¹ᵁ U) (a : Γ(X, U)) : resG Y e ((f.app U).hom a) = (f.appLE U W e).hom a

theorem **resG_pull_g** (i j : (L.pull f).ι) {W : Y.Opens} (h : W ≤ (L.pull f).U i ⊓ (L.pull f).U j) : resG Y h ((L.pull f).g i j : Γ(Y, (L.pull f).U i ⊓ (L.pull f).U j)) = (f.appLE (L.U (i PUnit.unit) ⊓ L.U (j PUnit.unit)) W (h.trans (inf_le_inf (pull_U_le f i) (pull_U_le f j)))).hom (L.g (i PUnit.unit) (j PUnit.unit) : Γ(X, L.U (i PUnit.unit) ⊓ L.U (j PUnit.unit)))

theorem **pull_inf_le** (V : X.Opens) (j : (L.pull f).ι) : f ⁻¹ᵁ V ⊓ (L.pull f).U j ≤ f ⁻¹ᵁ (V ⊓ L.U (j PUnit.unit))

@[simp] theorem **coord_pullSec** {V : X.Opens} (s : Γ(toMod L, V)) (j : (L.pull f).ι) : coord (pullSec L f V s) j = (f.appLE _ _ (pull_inf_le f V j)).hom (coord s (j PUnit.unit))

theorem **pullSec_map** {V W : X.Opens} (h : W ≤ V) (s : Γ(toMod L, V)) : pullSec L f W ((toMod L).presheaf.map (homOfLE h).op s) = (toMod (L.pull f)).presheaf.map (homOfLE (f.preimage_mono h)).op (pullSec L f V s)

theorem **chartEquiv_pullSec** (hL : L.IsLB) (j : (L.pull f).ι) {V : X.Opens} (h : V ≤ L.U (j PUnit.unit)) (h' : f ⁻¹ᵁ V ≤ (L.pull f).U j) (s : Γ(toMod L, V)) : chartEquiv (Semi.LB.isLB_pull hL f) j h' (pullSec L f V s) = (f.app V).hom (chartEquiv hL (j PUnit.unit) h s)

theorem **pullSec_chartSec** (hL : L.IsLB) (j : (L.pull f).ι) : pullSec L f _ (chartSec hL (j PUnit.unit)) = (toMod (L.pull f)).presheaf.map (homOfLE (pull_U f j).ge).op (chartSec (Semi.LB.isLB_pull hL f) j)

theorem **globalEquiv_pullSec_f** (s : LB.Sec L) (j : (L.pull f).ι) : (globalEquiv (L.pull f) (pullSec L f ⊤ (ofSec s ⊤))).f j = (f.appLE _ _ (pull_U_le f j)).hom (s.f (j PUnit.unit))

theorem **toPushforward_app** (V : X.Opens) (s : Γ(toMod L, V)) : (toPushforward L f).app V s = pullSec L f V s

theorem **coord_trivIso_hom_app** (V : X.Opens) (a : Γ(X, V)) (i : PUnit.{1}) : coord (L

theorem **twoChart_isLB** {U₀ U₁ : X.Opens} (h : U₀ ⊔ U₁ = ⊤) (u : Γ(X, U₀ ⊓ U₁)ˣ) : (twoChart U₀ U₁ u).IsLB where

theorem **mem_secOn_twoChart** {U₀ U₁ : X.Opens} (u : Γ(X, U₀ ⊓ U₁)ˣ) (V : X.Opens) (a : Γ(X, V ⊓ U₀)) (b : Γ(X, V ⊓ U₁)) : (fun i : Bool => match i with | true => a | false => b) ∈ secOn (twoChart U₀ U₁ u) V ↔ resG X (inf_le_left : V ⊓ U₀ ⊓ U₁ ≤ V ⊓ U₀) a = resG X (inf_le_inf_right _ inf_le_right) (u : Γ(X, U₀ ⊓ U₁)) ∗ resG X (inf_le_inf_right _ inf_le_left) b

theorem **twoChart_isGen** {U₀ U₁ : X.Opens} (u : Γ(X, U₀ ⊓ U₁)ˣ) (a : Γ(X, U₀)ˣ) (b : Γ(X, U₁)ˣ) (h : res inf_le_left a = u ∗ res inf_le_right b) : LB.Gen.IsGen (L

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.ExistsAwayFracOfMemHomLoc

theorem **awayFrac_apply** (f : A) (k : ℕ) (m : M) : awayFrac M f k m = LocalizedModule.mk m ⟨f ^ k, k, rfl⟩ : Submonoid.powers f)

theorem **awayFrac_zero** (f : A) : awayFrac M f 0 = LocalizedModule.mkLinearMap (Submonoid.powers f) M

theorem **exists_awayFrac** {f : A} (z : LocalizedModule.Away f M) : ∃ k m, awayFrac M f k m = z

theorem **awayFrac_eq_iff** {f : A} {k j : ℕ} {m m' : M} : awayFrac M f k m = awayFrac M f j m' ↔ ∃ l : ℕ, f ^ (l + j) • m = f ^ (l + k) • m'

theorem **awayFrac_eq_zero_iff** {f : A} {k : ℕ} {m : M} : awayFrac M f k m = 0 ↔ ∃ l : ℕ, f ^ l • m = 0

theorem **awayFrac_add_pow_smul** (f : A) (k j : ℕ) (m : M) : awayFrac M f (k + j) (f ^ j • m) = awayFrac M f k m

theorem **pow_smul_awayFrac** (f : A) (k : ℕ) (m : M) : f ^ k • awayFrac M f k m = awayFrac M f 0 m

theorem **isUnit_away_of_dvd_pow** {f g : A} {k : ℕ} (h : f ∣ g ^ k) : IsUnit (algebraMap A (Module.End A (LocalizedModule.Away g M)) f)

theorem **smul_away_bijective** {f g : A} {k : ℕ} (h : f ∣ g ^ k) : Function.Bijective (LinearMap.lsmul A (LocalizedModule.Away g M) f)

theorem **away_ext** {f : A} {φ ψ : LocalizedModule.Away f M →ₗ[A] N} (h : ∀ k m, φ (awayFrac M f k m) = ψ (awayFrac M f k m)) : φ = ψ

theorem **awayMap_awayFrac** (f : A) (φ : M →ₗ[A] N) (k : ℕ) (m : M) : awayMap f φ (awayFrac M f k m) = awayFrac N f k (φ m)

theorem **awayMap_id** (f : A) : awayMap f (LinearMap.id : M →ₗ[A] M) = LinearMap.id

theorem **awayMap_comp** (f : A) (ψ : N →ₗ[A] P) (φ : M →ₗ[A] N) : awayMap f (ψ ∘ₗ φ) = awayMap f ψ ∘ₗ awayMap f φ

theorem **awayMap_zero** (f : A) : awayMap f (0 : M →ₗ[A] N) = 0

theorem **awayMap_injective_of** {f : A} {φ : M →ₗ[A] N} (h : ∀ m, φ m = 0 → ∃ k : ℕ, f ^ k • m = 0) : Function.Injective (awayMap f φ)

theorem **awayMap_surjective_of** {f : A} {φ : M →ₗ[A] N} (h : ∀ y, ∃ k : ℕ, f ^ k • y ∈ LinearMap.range φ) : Function.Surjective (awayMap f φ)

theorem **awayMap_exact** (f : A) {φ : M →ₗ[A] N} {ψ : N →ₗ[A] P} (h : Function.Exact φ ψ) : Function.Exact (awayMap f φ) (awayMap f ψ)

theorem **awayFrac_injective** {f : A} (h : Function.Injective fun m : M => f • m) (k : ℕ) : Function.Injective (awayFrac M f k)

theorem **awayFrac_zero_bijective** {f : A} (h : Function.Bijective fun m : M => f • m) : Function.Bijective (awayFrac M f 0)

theorem **locRes_awayFrac** {ι : Type} (f : ι → A) {s t : Finset ι} (h : s ⊆ t) {c : A} (hc : ∏ i ∈ t, f i = c ∗ ∏ i ∈ s, f i) (k : ℕ) (m : M) : locRes f M h (awayFrac M _ k m) = awayFrac M _ k (c ^ k • m)

theorem **locRes_awayFrac_zero** {ι : Type} (f : ι → A) {s t : Finset ι} (h : s ⊆ t) (m : M) : locRes f M h (awayFrac M _ 0 m) = awayFrac M _ 0 m

theorem **awayMap_locRes** {ι : Type} (f : ι → A) {s t : Finset ι} (h : s ⊆ t) (φ : M →ₗ[A] N) (z : LocalizedModule.Away (∏ i ∈ s, f i) M) : awayMap _ φ (locRes f M h z) = locRes f N h (awayMap _ φ z)

theorem **xProd_mem** (s : Finset ι) : xProd R s ∈ polyGrading ι R s.card

theorem **xProd_pow_mem** (s : Finset ι) (k : ℕ) : xProd R s ^ k ∈ polyGrading ι R (k ∗ s.card)

theorem **xProd_pow_smul_mem** {s : Finset ι} {n : ℤ} {k : ℕ} {m : M} (hm : m ∈ 𝒜 (n + k ∗ s.card)) (j : ℕ) : xProd R s ^ j • m ∈ 𝒜 (n + (k + j : ℕ) ∗ s.card)

theorem **mem_homLoc** {s : Finset ι} {n : ℤ} {z : LocalizedModule.Away (xProd R s) M} : z ∈ homLoc 𝒜 s n ↔ ∃ k : ℕ, ∃ m ∈ 𝒜 (n + k ∗ s.card), awayFrac M (xProd R s) k m = z

theorem **awayFrac_mem_homLoc** {s : Finset ι} {n d : ℤ} {k : ℕ} {m : M} (hm : m ∈ 𝒜 d) (h : d = n + k ∗ s.card) : awayFrac M (xProd R s) k m ∈ homLoc 𝒜 s n

theorem **exists_awayFrac_of_mem_homLoc** {s : Finset ι} {n : ℤ} {z : LocalizedModule.Away (xProd R s) M} (hz : z ∈ homLoc 𝒜 s n) : ∃ k₀ : ℕ, ∀ k ≥ k₀, ∃ m ∈ 𝒜 (n + k ∗ s.card), awayFrac M (xProd R s) k m = z

theorem **isGradedMap_awayFrac** (s : Finset ι) (k : ℕ) : IsGradedMap (grShift 𝒜 (k * s.card)) (homLoc 𝒜 s) (awayFrac M (xProd R s) k)

theorem **isGradedMap_awayFrac_zero** (s : Finset ι) : IsGradedMap 𝒜 (homLoc 𝒜 s) (awayFrac M (xProd R s) 0)

theorem **isGradedMap_locRes** {s t : Finset ι} (h : s ⊆ t) : IsGradedMap (homLoc 𝒜 s) (homLoc 𝒜 t) (locRes X M h)

theorem **IsGradedMap.awayMap** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (s : Finset ι) : IsGradedMap (homLoc 𝒜 s) (homLoc 𝒩 s) (awayMap (xProd R s) f)

theorem **homLoc_grShift** (d : ℤ) (s : Finset ι) (n : ℤ) : homLoc (grShift 𝒜 d) s n = homLoc 𝒜 s (n + d)

theorem **coe_homLocMk** (s : Finset ι) (n : ℤ) (k : ℕ) (m : 𝒜 (n + k * s.card)) : (homLocMk 𝒜 s n k m : LocalizedModule.Away (xProd R s) M) = awayFrac M (xProd R s) k m

theorem **exists_homLocMk** {s : Finset ι} {n : ℤ} (z : homLoc 𝒜 s n) : ∃ k m, homLocMk 𝒜 s n k m = z

theorem **coe_toHomLoc** (s : Finset ι) (n : ℤ) (m : 𝒜 n) : (toHomLoc 𝒜 s n m : LocalizedModule.Away (xProd R s) M) = awayFrac M (xProd R s) 0 m

theorem **coe_homLocRes** {s t : Finset ι} (h : s ⊆ t) (n : ℤ) (z : homLoc 𝒜 s n) : (homLocRes 𝒜 h n z : LocalizedModule.Away (xProd R t) M) = locRes X M h z

theorem **homLocRes_self** {s : Finset ι} (h : s ⊆ s) (n : ℤ) (z : homLoc 𝒜 s n) : homLocRes 𝒜 h n z = z

theorem **homLocRes_homLocRes** {s t w : Finset ι} (h : s ⊆ t) (h' : t ⊆ w) (n : ℤ) (z : homLoc 𝒜 s n) : homLocRes 𝒜 h' n (homLocRes 𝒜 h n z) = homLocRes 𝒜 (h.trans h') n z

theorem **homLocRes_toHomLoc** {s t : Finset ι} (h : s ⊆ t) (n : ℤ) (m : 𝒜 n) : homLocRes 𝒜 h n (toHomLoc 𝒜 s n m) = toHomLoc 𝒜 t n m

theorem **toHomLoc_injective** {s : Finset ι} (h : Function.Injective fun m : M => xProd R s • m) (n : ℤ) : Function.Injective (toHomLoc 𝒜 s n)

theorem **toHomLoc_bijective** [Decomposition 𝒜] {s : Finset ι} (h : Function.Bijective fun m : M => xProd R s • m) (n : ℤ) : Function.Bijective (toHomLoc 𝒜 s n)

theorem **homLocEmptyEquiv_apply** (n : ℤ) (m : 𝒜 n) : homLocEmptyEquiv ι 𝒜 n m = toHomLoc 𝒜 ∅ n m

theorem **coe_homLocSMul** {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) (s : Finset ι) (n : ℤ) (z : homLoc 𝒜 s n) : (homLocSMul 𝒜 hp s n z : LocalizedModule.Away (xProd R s) M) = p • z.1

theorem **homLocSMul_bijective** [Decomposition 𝒜] {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) {s : Finset ι} {k : ℕ} (h : p ∣ xProd R s ^ k) (n : ℤ) : Function.Bijective (homLocSMul 𝒜 hp s n)

theorem **homLocSMul_X_bijective** [Decomposition 𝒜] {s : Finset ι} {i : ι} (hi : i ∈ s) (n : ℤ) : Function.Bijective (homLocSMul 𝒜 (X_mem_polyGrading i) s n)

theorem **coe_homLocMap** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (s : Finset ι) (n : ℤ) (z : homLoc 𝒜 s n) : (homLocMap hf s n z : LocalizedModule.Away (xProd R s) N) = awayMap _ f z

theorem **homLocMap_homLocMk** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (s : Finset ι) (n : ℤ) (k : ℕ) (m : 𝒜 (n + k * s.card)) : homLocMap hf s n (homLocMk 𝒜 s n k m) = homLocMk 𝒩 s n k (hf.piece _ m)

theorem **homLocMap_toHomLoc** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (s : Finset ι) (n : ℤ) (m : 𝒜 n) : homLocMap hf s n (toHomLoc 𝒜 s n m) = toHomLoc 𝒩 s n (hf.piece n m)

theorem **homLocMap_homLocRes** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) {s t : Finset ι} (h : s ⊆ t) (n : ℤ) (z : homLoc 𝒜 s n) : homLocMap hf t n (homLocRes 𝒜 h n z) = homLocRes 𝒩 h n (homLocMap hf s n z)

theorem **homLocMap_id** (h : IsGradedMap 𝒜 𝒜 (LinearMap.id : M →ₗ[MvPolynomial ι R] M)) (s : Finset ι) (n : ℤ) (z : homLoc 𝒜 s n) : homLocMap h s n z = z

theorem **homLocMap_comp** {g : N →ₗ[MvPolynomial ι R] P} {f : M →ₗ[MvPolynomial ι R] N} (hg : IsGradedMap 𝒩 𝒬 g) (hf : IsGradedMap ℳ 𝒩 f) (s : Finset ι) (n : ℤ) (z : homLoc 𝒜 s n) : homLocMap (hg.comp hf) s n z = homLocMap hg s n (homLocMap hf s n z)

theorem **homLocMap_zero** (h : IsGradedMap 𝒜 𝒩 (0 : M →ₗ[MvPolynomial ι R] N)) (s : Finset ι) (n : ℤ) (z : homLoc 𝒜 s n) : homLocMap h s n z = 0

theorem **homLocMap_injective_of** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) {s : Finset ι} (h : ∀ m, f m = 0 → ∃ k : ℕ, xProd R s ^ k • m = 0) (n : ℤ) : Function.Injective (homLocMap hf s n)

theorem **homLocMap_injective** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (h : Function.Injective f) (s : Finset ι) (n : ℤ) : Function.Injective (homLocMap hf s n)

theorem **homLocMap_surjective_of** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) {s : Finset ι} (h : ∀ y, ∃ k : ℕ, xProd R s ^ k • y ∈ LinearMap.range f) (n : ℤ) : Function.Surjective (homLocMap hf s n)

theorem **homLocMap_surjective** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (h : Function.Surjective f) (s : Finset ι) (n : ℤ) : Function.Surjective (homLocMap hf s n)

theorem **homLocMap_exact** {f : M →ₗ[MvPolynomial ι R] N} {g : N →ₗ[MvPolynomial ι R] P} (hf : IsGradedMap 𝒜 𝒩 f) (hg : IsGradedMap 𝒩 𝒬 g) (h : Function.Exact f g) (s : Finset ι) (n : ℤ) : Function.Exact (homLocMap hf s n) (homLocMap hg s n)

theorem **homLocMap_bijective_of_X** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) {s : Finset ι} {i : ι} (hi : i ∈ s) (h₁ : ∀ m, f m = 0 → ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • m = 0) (h₂ : ∀ y, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ LinearMap.range f) (n : ℤ) : Function.Bijective (homLocMap hf s n)

theorem **homLocLift_pow** (k j : ℕ) (m : 𝒜 (n + k * s.card)) : φ (k + j) ⟨_, xProd_pow_smul_mem 𝒜 m.2 j⟩ = φ k m

theorem **homLocLift_congr** {k j : ℕ} {m : 𝒜 (n + k * s.card)} {m' : 𝒜 (n + j * s.card)} (h : homLocMk 𝒜 s n k m = homLocMk 𝒜 s n j m') : φ k m = φ j m'

theorem **homLocLift_homLocMk** (k : ℕ) (m : 𝒜 (n + k * s.card)) : homLocLift φ hφ (homLocMk 𝒜 s n k m) = φ k m

theorem **homLocLift_injective_iff** : Function.Injective (homLocLift φ hφ) ↔ ∀ k m, φ k m = 0 → ∃ l : ℕ, xProd R s ^ l • (m : M) = 0

theorem **homLocLift_surjective_iff** : Function.Surjective (homLocLift φ hφ) ↔ ∀ t, ∃ k m, φ k m = t

theorem **homLocMap_piProj_bijective** (s : Finset ι) (n : ℤ) : Function.Bijective (LinearMap.pi fun k => homLocMap (isGradedMap_piProj ι 𝒜 k) s n)

theorem **homLocPiEquiv_apply** (s : Finset ι) (n : ℤ) (z : homLoc (grPi 𝒜) s n) (k : κ) : homLocPiEquiv 𝒜 s n z k = homLocMap (isGradedMap_piProj ι 𝒜 k) s n z

theorem **res_grCube** (n : ℤ) {s t : Finset ι} (h : s ⊆ t) : Cube.res (grCube ι 𝒜 n) h = homLocRes 𝒜 h n

theorem **grCubeMap_id** (h : IsGradedMap 𝒜 𝒜 (LinearMap.id : M →ₗ[MvPolynomial ι R] M)) (n : ℤ) : grCubeMap h n = 𝟙 _

theorem **grCubeMap_comp** {g : N →ₗ[MvPolynomial ι R] P} {f : M →ₗ[MvPolynomial ι R] N} (hg : IsGradedMap 𝒩 𝒬 g) (hf : IsGradedMap ℳ 𝒩 f) (n : ℤ) : grCubeMap (hg.comp hf) n = grCubeMap hf n ≫ grCubeMap hg n

theorem **grCubeMap_zero** (h : IsGradedMap 𝒜 𝒩 (0 : M →ₗ[MvPolynomial ι R] N)) (n : ℤ) : grCubeMap h n = 0

theorem **grCubeMap_congr** {f g : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (hg : IsGradedMap 𝒜 𝒩 g) (h : f = g) (n : ℤ) : grCubeMap hf n = grCubeMap hg n

theorem **grCech_eq** (n : ℤ) : grCech ι 𝒜 n = cechComplex (grCube ι 𝒜 n)

theorem **grCech_d** (n : ℤ) (p : ℕ) : (grCech ι 𝒜 n).d p (p + 1) = ModuleCat.ofHom (Cube.d (grCube ι 𝒜 n) (p + 1))

theorem **grCechMap_id** (h : IsGradedMap 𝒜 𝒜 (LinearMap.id : M →ₗ[MvPolynomial ι R] M)) (n : ℤ) : grCechMap h n = 𝟙 _

theorem **grCechMap_zero** (h : IsGradedMap 𝒜 𝒩 (0 : M →ₗ[MvPolynomial ι R] N)) (n : ℤ) : grCechMap h n = 0

theorem **isIso_grCechMap** [Decomposition 𝒜] [Decomposition 𝒩] {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (h₁ : ∀ (i : ι) m, f m = 0 → ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • m = 0) (h₂ : ∀ (i : ι) y, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ LinearMap.range f) (n : ℤ) : IsIso (grCechMap hf n)

theorem **cechHomologyZeroIso_grAug** (n : ℤ) (m : 𝒜 n) (s : {s : Finset ι // s.card = 1}) : ((cechHomologyZeroIso (grCube ι 𝒜 n)).hom.hom (grAug ι 𝒜 n m)).1 s = toHomLoc 𝒜 s.1 n m

theorem **grHMap_grAug** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (n : ℤ) (m : 𝒜 n) : (grHMap hf n 0).hom (grAug ι 𝒜 n m) = grAug ι 𝒩 n (hf.piece n m)

theorem **grAug_bijective_iff** (n : ℤ) : Function.Bijective (grAug ι 𝒜 n) ↔ (cubeComplex (grCube ι 𝒜 n)).ExactAt 0 ∧ (cubeComplex (grCube ι 𝒜 n)).ExactAt 1

theorem **IsGradedMap.shortExact_ker** {𝒩 : ℤ → Submodule R N} {𝒬 : ℤ → Submodule R P} {g : N →ₗ[MvPolynomial ι R] P} (hg : IsGradedMap 𝒩 𝒬 g) (h : Function.Surjective g) : GradedShortExact (grSub 𝒩 (LinearMap.ker g)) 𝒩 𝒬 (LinearMap.ker g).subtype g

theorem **GradedShortExact.homLoc** {𝒜 : ℤ → Submodule R M} {𝒩 : ℤ → Submodule R N} {𝒬 : ℤ → Submodule R P} [PolyGradedSMul ι 𝒜] [PolyGradedSMul ι 𝒩] [PolyGradedSMul ι 𝒬] [Decomposition 𝒜] [Decomposition 𝒩] [Decomposition 𝒬] {f : M →ₗ[MvPolynomial ι R] N} {g : N →ₗ[MvPolynomial ι R] P} (h : GradedShortExact 𝒜 𝒩 𝒬 f g) (s : Finset ι) (n : ℤ) : Function.Injective (homLocMap h.graded_f s n) ∧ Function.Exact (homLocMap h.graded_f s n) (homLocMap h.graded_g s n) ∧ Function.Surjective (homLocMap h.graded_g s n)

theorem **cubes_shortExact** (n : ℤ) (s : Finset ι) : ((h.cubes n).map ((evaluation _ _).obj s)).ShortExact

theorem **cubeComplex_shortExact** (n : ℤ) : ((h.cubes n).map (cubeFunctor ι R)).ShortExact

theorem **cech_shortExact** (n : ℤ) : (h.cech n).ShortExact

theorem **exact_δ_map** (n : ℤ) (p : ℕ) : Function.Exact (h.δ n p).hom (grHMap h.graded_f n (p + 1)).hom

theorem **exact_map_δ** (n : ℤ) (p : ℕ) : Function.Exact (grHMap h.graded_g n p).hom (h.δ n p).hom

theorem **cubeComplex_grPi_exactAt_iff** (n : ℤ) (p : ℕ) : (cubeComplex (grCube ι (grPi 𝒜) n)).ExactAt p ↔ ∀ k, (cubeComplex (grCube ι (𝒜 k) n)).ExactAt p

theorem **xProd_smul_injective** (s : Finset ι) : Function.Injective fun p : MvPolynomial ι R => xProd R s • p

theorem **operatorsCommute_id** : OperatorsCommute (ι

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.ExistsPowMulPowEqOfIsZeroCechH

theorem **presheafRes_sectionsPresheaf** (F : X.Modules) {W W' : X.Opens} (h : W' ≤ W) : presheafRes (sectionsPresheaf R F) h = sectionsOn.res R F h

theorem **isSheaf_sectionsPresheaf** (F : X.Modules) : Presheaf.IsSheaf (Opens.grothendieckTopology X) (sectionsPresheaf R F)

theorem **sectionsOn.res_res** (F : X.Modules) {W W' W'' : X.Opens} (h : W' ≤ W) (h' : W'' ≤ W') (x : sectionsOn F W) : sectionsOn.res R F h' (sectionsOn.res R F h x) = sectionsOn.res R F (h'.trans h) x

theorem **sectionsOn.congrOpens_apply** (F : X.Modules) {W W' : X.Opens} (h : W = W') (x : sectionsOn F W) : sectionsOn.congrOpens R F h x = sectionsOn.res R F h.ge x

theorem **sectionsOn.congrOpens_symm_apply** (F : X.Modules) {W W' : X.Opens} (h : W = W') (x : sectionsOn F W') : (sectionsOn.congrOpens R F h).symm x = sectionsOn.res R F h.le x

theorem **cechCube_def** : cechCube R F V U = presheafCube (sectionsPresheaf R F) V U

theorem **cechCube_obj** (s : Finset ι) : (cechCube R F V U).obj s = ModuleCat.of R (sectionsOn F (V ⊓ ⨅ i ∈ s, U i))

theorem **cechCube_obj'** (s : Finset ι) : (cechCube R F V U).obj s = ModuleCat.of R (sectionsOn F (cechOpen V U s))

theorem **res_cechCube** {s t : Finset ι} (h : s ⊆ t) : Cube.res (cechCube R F V U) h = sectionsOn.res R F (cechOpen_anti V U h)

theorem **cechCubeMap_app** (φ : F ⟶ G) (s : Finset ι) : ((cechCubeMap R V U φ).app s).hom = sectionsOn.map R φ (cechOpen V U s)

theorem **cechCubeMap_smulHom** (r : R) : cechCubeMap R V U (smulHom F (algebraMap R Γ(X, ⊤) r)) = r • 𝟙 (cechCube R F V U)

theorem **cechCubeEmptyEquiv_apply** (x : sectionsOn F V) : cechCubeEmptyEquiv R F V U x = sectionsOn.res R F (cechOpen_le V U ∅) x

theorem **cechCubeRes_app** {V' : X.Opens} {U' : ι → X.Opens} (h : ∀ s, cechOpen V' U' s ≤ cechOpen V U s) (s : Finset ι) : ((cechCubeRes R F h).app s).hom = sectionsOn.res R F (h s)

theorem **cechRestrictions_apply** (x : sectionsOn F V) (s : {s : Finset ι // s.card = 1}) : cechRestrictions R F V U x s = sectionsOn.res R F (cechOpen_le V U s.1) x

theorem **cech_def** : cech R F V U = cechComplex (cechCube R F V U)

theorem **cechH_def** (n : ℕ) : cechH R F V U n = (cech R F V U).homology n

theorem **cech_X** (p : ℕ) : (cech R F V U).X p = ModuleCat.of R (∀ s : {s : Finset ι // s.card = p + 1}, sectionsOn F (V ⊓ ⨅ i ∈ s.1, U i))

theorem **cech_d** (p : ℕ) : (cech R F V U).d p (p + 1) = ModuleCat.ofHom (Cube.d (cechCube R F V U) (p + 1))

theorem **cech_d_apply** (p : ℕ) (x : ∀ s : {s : Finset ι // s.card = p + 1}, sectionsOn F (cechOpen V U s.1)) (t : {s : Finset ι // s.card = p + 1 + 1}) : (show ∀ s : {s : Finset ι // s.card = p + 1 + 1}, sectionsOn F (cechOpen V U s.1) from ((cech R F V U).d p (p + 1)).hom x) t = ∑ i : t.1, cubeSign t.1 i.1 • sectionsOn.res R F (cechOpen_anti V U (Finset.erase_subset i.1 t.1)) (x (Cube.eraseIdx t i))

theorem **cechMap_id** : cechMap R V U (𝟙 F) = 𝟙 _

theorem **cechMap_comp** (φ : F ⟶ G) (ψ : G ⟶ H) : cechMap R V U (φ ≫ ψ) = cechMap R V U φ ≫ cechMap R V U ψ

theorem **cechHMap_id** (n : ℕ) : cechHMap R V U (𝟙 F) n = 𝟙 _

theorem **cechHMap_comp** (φ : F ⟶ G) (ψ : G ⟶ H) (n : ℕ) : cechHMap R V U (φ ≫ ψ) n = cechHMap R V U φ n ≫ cechHMap R V U ψ n

theorem **cechHMap_add** (φ ψ : F ⟶ G) (n : ℕ) : cechHMap R V U (φ + ψ) n = cechHMap R V U φ n + cechHMap R V U ψ n

theorem **cechHMap_cechClass** (φ : F ⟶ G) (n : ℕ) (x : LinearMap.ker (Cube.d (cechCube R F V U) (n + 1))) : (cechHMap R V U φ n).hom (cechClass (cechCube R F V U) n x) = cechClass (cechCube R G V U) n ⟨_, Cube.termMap_mem_ker (cechCubeMap R V U φ) x.2⟩

theorem **cechHMap_smulHom_apply** (r : R) (n : ℕ) (x : cechH R F V U n) : (cechHMap R V U (smulHom F (algebraMap R Γ(X, ⊤) r)) n).hom x = r • x

theorem **cechModulesFunctor_shortExact** (S : ShortComplex X.Modules) (h : ∀ s : Finset ι, s.Nonempty → Function.Injective (S.f.app (cechOpen V U s)) ∧ Function.Exact (S.f.app (cechOpen V U s)) (S.g.app (cechOpen V U s)) ∧ Function.Surjective (S.g.app (cechOpen V U s))) : (S.map (cechModulesFunctor R V U)).ShortExact

theorem **isZero_cech_X** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : IsZero ((cech R F V U).X n)

theorem **isZero_cechH** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : IsZero (cechH R F V U n)

theorem **subsingleton_cechH** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : Subsingleton (cechH R F V U n)

theorem **cechTopIso_hom** (h : ∀ i, U i ≤ V) : (cechTopIso R F h).hom = cechComplexMap (cechCubeRes R F (cechOpen_mono (le_top (a

theorem **cechTopIso_naturality** (h : ∀ i, U i ≤ V) (φ : F ⟶ G) : cechMap R ⊤ U φ ≫ (cechTopIso R G h).hom = (cechTopIso R F h).hom ≫ cechMap R V U φ

theorem **d_cechRestrictions** (x : sectionsOn F V) : Cube.d (cechCube R F V U) 1 (cechRestrictions R F V U x) = 0

theorem **toCechH_apply** (x : sectionsOn F V) : toCechH R F V U x = cechClass (cechCube R F V U) 0 ⟨cechRestrictions R F V U x, d_cechRestrictions R F V U x⟩

theorem **cechHMap_toCechH** (φ : F ⟶ G) (x : sectionsOn F V) : (cechHMap R V U φ 0).hom (toCechH R F V U x) = toCechH R G V U (sectionsOn.map R φ V x)

theorem **cechRestrictions_injective** : Function.Injective (cechRestrictions R F V U)

theorem **cechRestrictions_exact** : Function.Exact (cechRestrictions R F V U) (Cube.d (cechCube R F V U) 1)

theorem **cechCube_exactAt_zero_one** : (cubeComplex (cechCube R F V U)).ExactAt 0 ∧ (cubeComplex (cechCube R F V U)).ExactAt 1

theorem **toCechH_bijective** : Function.Bijective (toCechH R F V U)

theorem **cechHZeroEquiv_symm_apply** (x : sectionsOn F V) : (cechHZeroEquiv R F hc).symm x = toCechH R F V U x

theorem **cechHZeroEquiv_toCechH** (x : sectionsOn F V) : cechHZeroEquiv R F hc (toCechH R F V U x) = x

theorem **cechHZeroEquiv_map** (φ : F ⟶ G) (x : cechH R F V U 0) : cechHZeroEquiv R G hc ((cechHMap R V U φ 0).hom x) = sectionsOn.map R φ V (cechHZeroEquiv R F hc x)

theorem **cechHZeroEquivCohOn_map** (φ : F ⟶ G) (x : cechH R F V U 0) : cechHZeroEquivCohOn R G hc ((cechHMap R V U φ 0).hom x) = cohOn.map R φ V 0 (cechHZeroEquivCohOn R F hc x)

theorem **cechHScalarsEquiv_apply** (n : ℕ) (x : cechH R F V U n) : cechHScalarsEquiv R F V U S n x = (cechCubeScalars R F V U S).homology n x

theorem **cechHScalarsEquiv_smul** (n : ℕ) (r : R) (x : cechH R F V U n) : cechHScalarsEquiv R F V U S n (r • x) = algebraMap R S r • cechHScalarsEquiv R F V U S n x

theorem **cechHScalarsEquiv_map** (φ : F ⟶ G) (n : ℕ) (x : cechH R F V U n) : cechHScalarsEquiv R G V U S n ((cechHMap R V U φ n).hom x) = (cechHMap S V U φ n).hom (cechHScalarsEquiv R F V U S n x)

theorem **cechAbHEquiv_apply** (n : ℕ) (x : cechAbH F.presheaf V U n) : cechAbHEquiv R F V U n x = (cechAbCubeToCube R F V U).homology n x

theorem **cechAbHEquiv_map** (φ : F ⟶ G) (n : ℕ) (x : cechAbH F.presheaf V U n) : cechAbHEquiv R G V U n ((cechAbHMap V U φ.mapPresheaf n).hom x) = (cechHMap R V U φ n).hom (cechAbHEquiv R F V U n x)

theorem **smul_cechAbHEquiv** (r : R) (n : ℕ) (x : cechAbH F.presheaf V U n) : r • cechAbHEquiv R F V U n x = cechAbHEquiv R F V U n ((cechAbHMap V U (smulHom F (algebraMap R Γ(X, ⊤) r)).mapPresheaf n).hom x)

theorem **cechAbHEquiv_toCechAbH** (x : sectionsOn F V) : cechAbHEquiv R F V U 0 (toCechAbH F.presheaf V U (sectionsOn.toΓ F V x)) = toCechH R F V U x

theorem **preimage_cechOpen** (s : Finset ι) : g ⁻¹ᵁ cechOpen V U s = cechOpen (g ⁻¹ᵁ V) (fun i => g ⁻¹ᵁ U i) s

theorem **sectionsOn.toΓ_mapOver** (W : X.Opens) (x : sectionsOn F W) : sectionsOn.toΓ G (g ⁻¹ᵁ W) (sectionsOn.mapOver hσ φ W x) = φ.app W (sectionsOn.toΓ F W x)

theorem **sectionsOn.mapOver_res** {W W' : X.Opens} (h : W' ≤ W) (x : sectionsOn F W) : sectionsOn.mapOver hσ φ W' (sectionsOn.res R F h x) = sectionsOn.res S G (g.preimage_mono h) (sectionsOn.mapOver hσ φ W x)

theorem **cechCubeMapOver_app** (s : Finset ι) (x : sectionsOn F (cechOpen V U s)) : (cechCubeMapOver V U hσ φ).app s x = sectionsOn.res S G (preimage_cechOpen g V U s).ge (sectionsOn.mapOver hσ φ (cechOpen V U s) x)

theorem **cechCubeMapOver_id_bijective** (s : Finset ι) : Function.Bijective ((cechCubeMapOver V U hσ (𝟙 ((Scheme.Modules.pushforward g).obj G))).app s)

theorem **cechCubePushforwardIso_hom_app_apply** (s : Finset ι) (x : sectionsOn G (g ⁻¹ᵁ cechOpen V U s)) : ((cechCubePushforwardIso V U G hg).hom.app s).hom x = sectionsOn.res R G (preimage_cechOpen g V U s).ge x

theorem **cechCubePushforwardIso_naturality** {G' : Y.Modules} (ψ : G ⟶ G') : cechCubeMap R V U ((Scheme.Modules.pushforward g).map ψ) ≫ (cechCubePushforwardIso V U G' hg).hom = (cechCubePushforwardIso V U G hg).hom ≫ cechCubeMap R _ _ ψ

theorem **cechTwo_d** (x : (cech R F V U).X 0) : cechTwoXOne R F V U (((cech R F V U).d 0 1).hom x) = sectionsOn.res R F (inf_le_inf_left V inf_le_right) (cechTwoXZero R F V U x).2 − sectionsOn.res R F (inf_le_inf_left V inf_le_left) (cechTwoXZero R F V U x).1

theorem **cechTwoXZero_cechRestrictions** (x : sectionsOn F V) : cechTwoXZero R F V U (cechRestrictions R F V U x) = (sectionsOn.res R F inf_le_left x, sectionsOn.res R F inf_le_left x)

theorem **isZero_cechTwo_X** {n : ℕ} (h : 2 ≤ n) : IsZero ((cech R F V U).X n)

theorem **isZero_cechH_one_iff_surjective** : IsZero (cechH R F V U 1) ↔ Function.Surjective ((cech R F V U).d 0 1).hom

theorem **res_tildeToSections** {W W' : (Spec A).Opens} (h : W' ≤ W) (m : M) : sectionsOn.res A (tilde M) h (tildeToSections M W m) = tildeToSections M W' m

theorem **basicOpenFamily_apply** (f : ι → A) (i : ι) : basicOpenFamily f i = PrimeSpectrum.basicOpen (f i)

theorem **isLocalizedModule_tildeToSections** (f : A) {W : (Spec A).Opens} (h : W = PrimeSpectrum.basicOpen f) : IsLocalizedModule.Away f (tildeToSections M W)

theorem **cechOpen_basicOpenFamily** (f : ι → A) (s : Finset ι) : cechOpen ⊤ (basicOpenFamily f) s = PrimeSpectrum.basicOpen (∏ i ∈ s, f i)

theorem **iSup_basicOpenFamily_eq_top_iff** (f : ι → A) : ⨆ i, basicOpenFamily f i = ⊤ ↔ Ideal.span (Set.range f) = ⊤

theorem **cechCube_tilde_exactAt** [LinearOrder ι] [Finite ι] {f : ι → A} (hf : Ideal.span (Set.range f) = ⊤) (p : ℕ) : (cubeComplex (cechCube A (tilde M) ⊤ (basicOpenFamily f))).ExactAt p

theorem **isZero_cechH_tilde_succ** [LinearOrder ι] [Finite ι] {f : ι → A} (hf : Ideal.span (Set.range f) = ⊤) (p : ℕ) : IsZero (cechH A (tilde M) ⊤ (basicOpenFamily f) (p + 1))

theorem **cechHZeroTildeEquiv_symm_apply** [LinearOrder ι] {f : ι → A} (hf : Ideal.span (Set.range f) = ⊤) (m : M) : (cechHZeroTildeEquiv M hf).symm m = toCechH A (tilde M) ⊤ (basicOpenFamily f) (tildeToSections M ⊤ m)

theorem **awayTransition_mk_one** {a b : A} (h : a ∣ b) (m : M) : awayTransition M h (LocalizedModule.mk m 1) = LocalizedModule.mk m 1

theorem **sectionsAwayEquiv_tildeToSections** (f : A) {W : (Spec A).Opens} (h : W = PrimeSpectrum.basicOpen f) (m : M) : sectionsAwayEquiv M f h (tildeToSections M W m) = LocalizedModule.mk m 1

theorem **sectionsAwayEquiv_res** {a b : A} (hab : a ∣ b) {W W' : (Spec A).Opens} (h : W = PrimeSpectrum.basicOpen a) (h' : W' = PrimeSpectrum.basicOpen b) (hW : W' ≤ W) (x : sectionsOn (tilde M) W) : sectionsAwayEquiv M b h' (sectionsOn.res A (tilde M) hW x) = awayTransition M hab (sectionsAwayEquiv M a h x)

theorem **twoCharts_zero** : twoCharts f g 0 = PrimeSpectrum.basicOpen f

theorem **twoCharts_one** : twoCharts f g 1 = PrimeSpectrum.basicOpen g

theorem **twoCharts_inf** : ⊤ ⊓ (twoCharts f g 0 ⊓ twoCharts f g 1) = PrimeSpectrum.basicOpen (f * g)

theorem **twoCharts_d** (x : (cech A (tilde M) ⊤ (twoCharts f g)).X 0) : twoChartsXOne M f g (((cech A (tilde M) ⊤ (twoCharts f g)).d 0 1).hom x) = awayTransition M (dvd_mul_left g f) (twoChartsXZero M f g x).2 − awayTransition M (dvd_mul_right f g) (twoChartsXZero M f g x).1

theorem **twoChartsXZero_cechRestrictions** (m : M) : twoChartsXZero M f g (cechRestrictions A (tilde M) ⊤ (twoCharts f g) (tildeToSections M ⊤ m)) = (LocalizedModule.mk m 1, LocalizedModule.mk m 1)

theorem **isZero_cechH_twoCharts_one_iff** : IsZero (cechH A (tilde M) ⊤ (twoCharts f g) 1) ↔ ∀ z, ∃ a b, awayTransition M (dvd_mul_left g f) b − awayTransition M (dvd_mul_right f g) a = z

theorem **span_range_pair_one_sub** : Ideal.span (Set.range ![f, 1 − f]) = ⊤

theorem **iSup_twoCharts_one_sub** : ⨆ i, twoCharts f (1 − f) i = ⊤

theorem **isZero_cechH_twoCharts_succ** (p : ℕ) : IsZero (cechH A (tilde M) ⊤ (twoCharts f (1 − f)) (p + 1))

theorem **isZero_cechH_structure_twoCharts_succ** (p : ℕ) : IsZero (cechH A (structureModule (Spec A)) ⊤ (twoCharts f (1 − f)) (p + 1))

theorem **exists_pow_mul_pow_eq_of_isZero_cechH** (h : IsZero (cechH A (structureModule (Spec A)) ⊤ (twoCharts f g) 1)) : ∃ (n k : ℕ) (a b : A), f ^ n * g ^ k = f ^ (n + 1) * b + g ^ (k + 1) * a

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.FiniteCohPushforwardIffOfIsClosedImmersion

theorem **chi_def** (F : X.Modules) : chi k F = ∑ᶠ n, (−1 : ℤ) ^ n * hdim k F n

theorem **chi_eq_sum_range** (N : ℕ) (hN : ∀ n, N ≤ n → Subsingleton (coh F n)) : chi k F = ∑ n ∈ Finset.range N, (−1 : ℤ) ^ n * hdim k F n

theorem **chi_eq_hdim_zero** (h : ∀ n, 0 < n → Subsingleton (coh F n)) : chi k F = hdim k F 0

theorem **chi_eq_hdim_zero_sub_hdim_one** (h : ∀ n, 2 ≤ n → Subsingleton (coh F n)) : chi k F = hdim k F 0 − hdim k F 1

theorem **chi_eq_zero_of_subsingleton** (h : ∀ n, Subsingleton (coh F n)) : chi k F = 0

theorem **chi_eq_zero_of_isZero** (h : IsZero F) : chi k F = 0

theorem **chi_congr** (e : F ≅ G) : chi k F = chi k G

theorem **chi_eq_of_linearEquiv** [Algebra k Γ(Y, ⊤)] {H : Y.Modules} (e : ∀ n, coh F n ≃ₗ[k] coh H n) : chi k F = chi k H

theorem **of_linearEquiv** [Algebra k Γ(Y, ⊤)] {H : Y.Modules} (h : FiniteCoh k F) (e : ∀ n, coh F n ≃ₗ[k] coh H n) : FiniteCoh k H where

theorem **of_iso** (h : FiniteCoh k F) (e : F ≅ G) : FiniteCoh k G

theorem **of_subsingleton** (h : ∀ n, Subsingleton (coh F n)) : FiniteCoh k F

theorem **of_isZero** (h : IsZero F) : FiniteCoh k F

theorem **of_acyclic** (h₀ : Module.Finite k (coh F 0)) (h : ∀ n, 0 < n → Subsingleton (coh F n)) : FiniteCoh k F where

theorem **finite_support** (h : FiniteCoh k F) : (Function.support fun n ↦ (−1 : ℤ) ^ n * hdim k F n).Finite

theorem **of_shortExact₂** (hS : S.ShortExact) (h₁ : FiniteCoh k S.X₁) (h₃ : FiniteCoh k S.X₃) : FiniteCoh k S.X₂ where

theorem **of_shortExact₃** (hS : S.ShortExact) (h₁ : FiniteCoh k S.X₁) (h₂ : FiniteCoh k S.X₂) : FiniteCoh k S.X₃ where

theorem **of_shortExact₁** (hS : S.ShortExact) (h₂ : FiniteCoh k S.X₂) (h₃ : FiniteCoh k S.X₃) : FiniteCoh k S.X₁ where

theorem **biprod** (hF : FiniteCoh k F) (hG : FiniteCoh k G) : FiniteCoh k (F ⊞ G)

theorem **chi_shortExact** (hS : S.ShortExact) (h₁ : FiniteCoh k S.X₁) (h₃ : FiniteCoh k S.X₃) : chi k S.X₂ = chi k S.X₁ + chi k S.X₃

theorem **chi_shortExact₃** (hS : S.ShortExact) (h₁ : FiniteCoh k S.X₁) (h₂ : FiniteCoh k S.X₂) : chi k S.X₃ = chi k S.X₂ − chi k S.X₁

theorem **chi_shortExact₁** (hS : S.ShortExact) (h₂ : FiniteCoh k S.X₂) (h₃ : FiniteCoh k S.X₃) : chi k S.X₁ = chi k S.X₂ − chi k S.X₃

theorem **chi_biprod** (hF : FiniteCoh k F) (hG : FiniteCoh k G) : chi k (F ⊞ G) = chi k F + chi k G

theorem **exists_forall_subsingleton_coh** (h : ∀ i, FiniteCoh k (E i)) : ∃ N, ∀ i n, N ≤ n → Subsingleton (coh (E i) n)

theorem **FiniteCoh.pi** (h : ∀ i, FiniteCoh k (E i)) : FiniteCoh k (∏ᶜ E) where

theorem **chi_pi** (h : ∀ i, FiniteCoh k (E i)) : chi k (∏ᶜ E) = ∑ i, chi k (E i)

theorem **finiteCoh_of_isAffine_of_finite** [IsAffine X] [Module.Finite k Γ(X, ⊤)] [QCoh.IsFiniteQC F] : FiniteCoh k F

theorem **chi_structureModule_of_isAffine** [IsAffine X] : chi k (structureModule X) = Module.finrank k Γ(X, ⊤)

theorem **pushforward_map_smulHom** (H : Y.Modules) (r : Γ(X, ⊤)) : (Scheme.Modules.pushforward f).map (smulHom H (f.appTop r)) = smulHom ((Scheme.Modules.pushforward f).obj H) r

theorem **pushH'_map** {H H' : Y.Modules} (φ : H ⟶ H') (U : X.Opens) (n : ℕ) (x : Sheaf.H'.{u} ((toAbSheaf X).obj ((Scheme.Modules.pushforward f).obj H)) n U) : pushH' f H' U n (SiteCoh.map ((toAbSheaf X).map ((Scheme.Modules.pushforward f).map φ)) n U x) = SiteCoh.map ((toAbSheaf Y).map φ) n (f ⁻¹ᵁ U) (pushH' f H U n x)

theorem **cohPushMap_bijective_of_isClosedImmersion** [IsClosedImmersion f] (H : Y.Modules) (U : X.Opens) (n : ℕ) : Function.Bijective (cohPushMap R f hf H U n)

theorem **cohPushMap_bijective_of_isAffineHom** [IsAffineHom f] (H : Y.Modules) [H.IsQuasicoherent] (U : X.Opens) (n : ℕ) : Function.Bijective (cohPushMap R f hf H U n)

theorem **finiteCoh_pushforward_iff_of_bijective** (h : ∀ n, Function.Bijective (cohPushMap k f hf H ⊤ n)) : FiniteCoh k ((Scheme.Modules.pushforward f).obj H) ↔ FiniteCoh k H

theorem **chi_pushforward_of_bijective** (h : ∀ n, Function.Bijective (cohPushMap k f hf H ⊤ n)) : chi k ((Scheme.Modules.pushforward f).obj H) = chi k H

theorem **finiteCoh_pushforward_iff_of_isClosedImmersion** [IsClosedImmersion f] : FiniteCoh k ((Scheme.Modules.pushforward f).obj H) ↔ FiniteCoh k H

theorem **chi_pushforward_of_isClosedImmersion** [IsClosedImmersion f] : chi k ((Scheme.Modules.pushforward f).obj H) = chi k H

theorem **finiteCoh_pushforward_iff_of_isAffineHom** [IsAffineHom f] [H.IsQuasicoherent] : FiniteCoh k ((Scheme.Modules.pushforward f).obj H) ↔ FiniteCoh k H

theorem **chi_pushforward_of_isAffineHom** [IsAffineHom f] [H.IsQuasicoherent] : chi k ((Scheme.Modules.pushforward f).obj H) = chi k H

theorem **appTop_algebraMap_along** (R : Type v) [CommRing R] [Algebra R Γ(X, ⊤)] (f : Y ⟶ X) (r : R) : letI

theorem **isIso_unitToPush** (f : Y ⟶ X) (h : ∀ U, Function.Bijective (f.app U)) : IsIso (unitToPush f)

theorem **prodChart_le** (L : ι → LB X) (j : ∀ i, (L i).ι) (i : ι) : prodChart L j ≤ (L i).U (j i)

theorem **mem_prodChart** {L : ι → LB X} {j : ∀ i, (L i).ι} {x : X} : x ∈ prodChart L j ↔ ∀ i, x ∈ (L i).U (j i)

theorem **res_lbProd_g** (L : ι → LB X) (n : ι → ℤ) (j j' : ∀ i, (L i).ι) {W : X.Opens} (h : W ≤ prodChart L j ⊓ prodChart L j') : res h ((lbProd L n).g j j') = ∏ i, res (h.trans (inf_le_inf (prodChart_le L j i) (prodChart_le L j' i))) ((L i).g (j i) (j' i)) ^ n i

theorem **gm_lbProd_g** (L : ι → LB X) (n : ι → ℤ) (r : Y ⟶ X) (y : Y) (j j' : ∀ i, (L i).ι) (hj : r y ∈ prodChart L j) (hj' : r y ∈ prodChart L j') : gm r y (prodChart L j ⊓ prodChart L j') ((lbProd L n).g j j') = ∏ i, gm r y _ ((L i).g (j i) (j' i)) ^ n i

theorem **isLB_lbProd** (hL : ∀ i, (L i).IsLB) (n : ι → ℤ) : (lbProd L n).IsLB where

theorem **isLB_lbInvProd** (hL : ∀ i, (L i).IsLB) (S : Finset ι) : (lbInvProd L S).IsLB

theorem **cls_lbProd_add** (hL : ∀ i, (L i).IsLB) (n m : ι → ℤ) : cls _ (isLB_lbProd hL (n + m)) = cls _ (isLB_lbProd hL n) + cls _ (isLB_lbProd hL m)

theorem **cls_lbProd_single** [DecidableEq ι] (hL : ∀ i, (L i).IsLB) (i : ι) : cls _ (isLB_lbProd hL (Pi.single i 1)) = cls (L i) (hL i)

theorem **cls_lbProd** (hL : ∀ i, (L i).IsLB) (n : ι → ℤ) (h : (lbProd L n).IsLB) : cls _ h = ∑ i, n i • cls (L i) (hL i)

theorem **cls_lbInvProd** (hL : ∀ i, (L i).IsLB) (S : Finset ι) (h : (lbInvProd L S).IsLB) : cls _ h = -∑ i ∈ S, cls (L i) (hL i)

theorem **cls_lbZPow** {M : LB X} (hM : M.IsLB) (n : ℤ) (h : (lbZPow M n).IsLB) : cls _ h = n • cls M hM

theorem **cls_triv** (h : (LB.triv X).IsLB) : cls _ h = 0

theorem **nonempty_twist_iso_of_cls_eq** (F : X.Modules) {M N : LB X} (hM : M.IsLB) (hN : N.IsLB) (h : cls M hM = cls N hN) : Nonempty (twist F M ≅ twist F N)

theorem **nonempty_twist_twist_iso_of_cls_eq** (F : X.Modules) {M N P : LB X} (hM : M.IsLB) (hN : N.IsLB) (hP : P.IsLB) (h : cls P hP = cls M hM + cls N hN) : Nonempty (twist (twist F M) N ≅ twist F P)

theorem **nonempty_twist_iso_of_cls_eq_zero** (F : X.Modules) {M : LB X} (hM : M.IsLB) (h : cls M hM = 0) : Nonempty (twist F M ≅ F)

@[simp] theorem **chiPic_cls** (hM : M.IsLB) : chiPic k F (cls M hM) = chi k (twist F M)

@[simp] theorem **chiPic_zero** : chiPic k F 0 = chi k F

theorem **chiPic_congr** (e : F ≅ G) : chiPic k F = chiPic k G

theorem **chiPic_twist** (hM : M.IsLB) (c : Pic X) : chiPic k (twist F M) c = chiPic k F (cls M hM + c)

theorem **chi_twistZ** (hM : M.IsLB) (n : ℤ) : chi k (twistZ F M n) = chiPic k F (n • cls M hM)

theorem **finiteCoh** (h : FiniteCohTwists k F) : FiniteCoh k F

theorem **of_iso** (h : FiniteCohTwists k F) (e : F ≅ G) : FiniteCohTwists k G

theorem **twist** (h : FiniteCohTwists k F) (hM : M.IsLB) : FiniteCohTwists k (twist F M)

theorem **of_isZero** (h : IsZero F) : FiniteCohTwists k F

theorem **of_shortExact₂** (hS : S.ShortExact) (h₁ : FiniteCohTwists k S.X₁) (h₃ : FiniteCohTwists k S.X₃) : FiniteCohTwists k S.X₂

theorem **of_shortExact₃** (hS : S.ShortExact) (h₁ : FiniteCohTwists k S.X₁) (h₂ : FiniteCohTwists k S.X₂) : FiniteCohTwists k S.X₃

theorem **of_shortExact₁** (hS : S.ShortExact) (h₂ : FiniteCohTwists k S.X₂) (h₃ : FiniteCohTwists k S.X₃) : FiniteCohTwists k S.X₁

theorem **finiteCohTwists_of_isAffine_of_finite** [IsAffine X] [Module.Finite k Γ(X, ⊤)] [QCoh.IsFiniteQC F] : FiniteCohTwists k F

theorem **chi_twist_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) (hM : M.IsLB) (h₁ : FiniteCoh k (twist S.X₁ M)) (h₃ : FiniteCoh k (twist S.X₃ M)) : chi k (twist S.X₂ M) = chi k (twist S.X₁ M) + chi k (twist S.X₃ M)

theorem **chiPic_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) (h₁ : FiniteCohTwists k S.X₁) (h₃ : FiniteCohTwists k S.X₃) : chiPic k S.X₂ = chiPic k S.X₁ + chiPic k S.X₃

theorem **chi_twist_pushforward_of_isClosedImmersion** [IsClosedImmersion f] (M : LB X) : chi k (twist ((Scheme.Modules.pushforward f).obj H) M) = chi k (twist H (M.pull f))

theorem **chi_twist_pushforward_of_isAffineHom** [IsAffineHom f] [H.IsQuasicoherent] (hM : M.IsLB) : chi k (twist ((Scheme.Modules.pushforward f).obj H) M) = chi k (twist H (M.pull f))

theorem **chiPic_pushforward_of_isClosedImmersion** [IsClosedImmersion f] : chiPic k ((Scheme.Modules.pushforward f).obj H) = chiPic k H ∘ pull f

theorem **chiPic_pushforward_of_isAffineHom** [IsAffineHom f] [H.IsQuasicoherent] : chiPic k ((Scheme.Modules.pushforward f).obj H) = chiPic k H ∘ pull f

theorem **FiniteCohTwists.pushforward_of_isClosedImmersion** [IsClosedImmersion f] (h : FiniteCohTwists k H) : FiniteCohTwists k ((Scheme.Modules.pushforward f).obj H)

theorem **FiniteCohTwists.pushforward_of_isAffineHom** [IsAffineHom f] [H.IsQuasicoherent] (h : FiniteCohTwists k H) : FiniteCohTwists k ((Scheme.Modules.pushforward f).obj H)

theorem **interNum_def** (F : X.Modules) (L : ι → LB X) : interNum k F L = ∑ S : Finset ι, (-1 : ℤ) ^ S.card * chi k (twist F (lbInvProd L S))

theorem **interNumOn_def** (f : Z ⟶ X) (L : ι → LB X) : interNumOn k f L = interNum k ((Scheme.Modules.pushforward f).obj (structureModule Z)) L

theorem **interNumSub_def** (I : X.IdealSheafData) (L : ι → LB X) : interNumSub k I L = interNumOn k I.subschemeι L

theorem **interNumRed_def** (Z : Closeds X) (L : ι → LB X) : interNumRed k Z L = interNumSub k (Scheme.IdealSheafData.vanishingIdeal Z) L

theorem **topSelf_def** (M : LB X) (d : ℕ) : topSelfIntersection k M d = interNum k (structureModule X) fun _ : Fin d ↦ M

theorem **interNum_eq_multiDiff** (hL : ∀ i, (L i).IsLB) : interNum k F L = multiDiff (fun i ↦ cls (L i) (hL i)) (chiPic k F) 0

theorem **interNum_twist_eq_multiDiff** (hL : ∀ i, (L i).IsLB) (hM : M.IsLB) : interNum k (twist F M) L = multiDiff (fun i ↦ cls (L i) (hL i)) (chiPic k F) (cls M hM)

theorem **interNum_lbZPow_eq_multiDiff** (hM : M.IsLB) (a : ι → ℤ) : interNum k F (fun i ↦ lbZPow M (a i)) = multiDiff a (fun n ↦ chi k (twistZ F M n)) 0

theorem **interNum_congr** (e : F ≅ G) (L : ι → LB X) : interNum k F L = interNum k G L

theorem **interNum_congr_cls** (hL : ∀ i, (L i).IsLB) (hL' : ∀ i, (L' i).IsLB) (h : ∀ i, cls (L i) (hL i) = cls (L' i) (hL' i)) : interNum k F L = interNum k F L'

theorem **interNum_congr_lbIso** (hL : ∀ i, (L i).IsLB) (hL' : ∀ i, (L' i).IsLB) (h : ∀ i, LBIso (L i) (L' i)) : interNum k F L = interNum k F L'

theorem **interNum_comp_equiv** (hL : ∀ i, (L i).IsLB) (e : κ ≃ ι) : interNum k F (L ∘ e) = interNum k F L

theorem **interNum_comp_perm** (hL : ∀ i, (L i).IsLB) (σ : Equiv.Perm ι) : interNum k F (L ∘ σ) = interNum k F L

@[simp] theorem **interNum_of_isEmpty** [IsEmpty ι] (L : ι → LB X) : interNum k F L = chi k F

theorem **interNum_eq_zero_of_isZero** (h : IsZero F) (hL : ∀ i, (L i).IsLB) : interNum k F L = 0

theorem **isLB_cons** (hM : M.IsLB) (hP : ∀ i, (P i).IsLB) : ∀ i, ((Fin.cons M P : Fin (d + 1) → LB X) i).IsLB

theorem **cls_cons** (hM : M.IsLB) (hP : ∀ i, (P i).IsLB) : (fun i ↦ cls _ (isLB_cons hM hP i)) = (Fin.cons (cls M hM) (fun i ↦ cls (P i) (hP i)) : Fin (d + 1) → Pic X)

theorem **interNum_cons_of_cls_eq_neg** (hM : M.IsLB) (hP : ∀ i, (P i).IsLB) (hN : N.IsLB) (h : cls N hN = -cls M hM) : interNum k F (Fin.cons M P : Fin (d + 1) → LB X) = interNum k F P - interNum k (twist F N) P

theorem **interNum_cons** (hM : M.IsLB) (hP : ∀ i, (P i).IsLB) : interNum k F (Fin.cons M P : Fin (d + 1) → LB X) = interNum k F P - interNum k (twistZ F M (-1)) P

theorem **interNum_twist_cons** (hM : M.IsLB) (hP : ∀ i, (P i).IsLB) : interNum k (twist F M) (Fin.cons M P : Fin (d + 1) → LB X) = interNum k (twist F M) P - interNum k F P

theorem **interNum_cons_of_cls_eq_add** (hM : M.IsLB) (hA : A.IsLB) (hB : B.IsLB) (hP : ∀ i, (P i).IsLB) (h : cls M hM = cls A hA + cls B hB) : interNum k F (Fin.cons M P : Fin (d + 1) → LB X) = interNum k F (Fin.cons A P : Fin (d + 1) → LB X) + interNum k F (Fin.cons B P : Fin (d + 1) → LB X) - interNum k F (Fin.cons A (Fin.cons B P) : Fin (d + 1 + 1) → LB X)

theorem **interNum_one** (hM : M.IsLB) : interNum k F ![M] = chi k F - chi k (twistZ F M (-1))

theorem **interNum_eq_zero_of_cls_eq_zero** (hL : ∀ i, (L i).IsLB) (i : ι) (h : cls (L i) (hL i) = 0) : interNum k F L = 0

theorem **interNum_const_eq_multiDiff** (hM : M.IsLB) : interNum k F (fun _ : ι ↦ M) = multiDiff (fun _ : ι ↦ 1) (fun n ↦ chi k (twistZ F M n)) 0

theorem **interNum_const_eq_sum** (hM : M.IsLB) (d : ℕ) : interNum k F (fun _ : Fin d ↦ M) = ∑ j ∈ Finset.range (d + 1), (-1 : ℤ) ^ j * d.choose j * chi k (twistZ F M (-j))

theorem **interNum_lbZPow_of_polynomial** (hM : M.IsLB) {d : ℕ} (p : Polynomial ℚ) (hp : p.natDegree ≤ d) (h : ∀ n : ℤ, (chi k (twistZ F M n) : ℚ) = p.eval (n : ℚ)) (a : Fin d → ℤ) : (interNum k F (fun i ↦ lbZPow M (a i)) : ℚ) = d.factorial * p.coeff d * ∏ i, (a i : ℚ)

theorem **interNum_const_of_polynomial** (hM : M.IsLB) {d : ℕ} (p : Polynomial ℚ) (hp : p.natDegree ≤ d) (h : ∀ n : ℤ, (chi k (twistZ F M n) : ℚ) = p.eval (n : ℚ)) : (interNum k F (fun _ : Fin d ↦ M) : ℚ) = d.factorial * p.coeff d

@[simp] theorem **topSelf_zero** (M : LB X) : topSelfIntersection k M 0 = chi k (structureModule X)

theorem **topSelf_congr** (hM : M.IsLB) (hN : N.IsLB) (h : LBIso M N) (d : ℕ) : topSelfIntersection k M d = topSelfIntersection k N d

@[simp] theorem **interNumOn_id** (L : ι → LB X) : interNumOn k (𝟙 X) L = interNum k (structureModule X) L

theorem **interNumOn_comp_of_isIso** (g : Y ⟶ Z) (f : Z ⟶ X) [IsIso (unitToPush g)] (L : ι → LB X) : interNumOn k (g ≫ f) L = interNumOn k f L

theorem **interNumOn_eq_interNumSub** (f : Z ⟶ X) [IsClosedImmersion f] (L : ι → LB X) : interNumOn k f L = interNumSub k f.ker L

theorem **interNum_shortExact'** (hS : S.ShortExact) (hL : ∀ i, (L i).IsLB) (h₁ : ∀ T, FiniteCoh k (twist S.X₁ (lbInvProd L T))) (h₃ : ∀ T, FiniteCoh k (twist S.X₃ (lbInvProd L T))) : interNum k S.X₂ L = interNum k S.X₁ L + interNum k S.X₃ L

theorem **interNum_shortExact** (hS : S.ShortExact) (hL : ∀ i, (L i).IsLB) (h₁ : FiniteCohTwists k S.X₁) (h₃ : FiniteCohTwists k S.X₃) : interNum k S.X₂ L = interNum k S.X₁ L + interNum k S.X₃ L

theorem **interNum_shortExact₃** (hS : S.ShortExact) (hL : ∀ i, (L i).IsLB) (h₁ : FiniteCohTwists k S.X₁) (h₂ : FiniteCohTwists k S.X₂) : interNum k S.X₃ L = interNum k S.X₂ L - interNum k S.X₁ L

theorem **interNum_shortExact₁** (hS : S.ShortExact) (hL : ∀ i, (L i).IsLB) (h₂ : FiniteCohTwists k S.X₂) (h₃ : FiniteCohTwists k S.X₃) : interNum k S.X₁ L = interNum k S.X₂ L - interNum k S.X₃ L

theorem **interNum_cokernel** (φ : F ⟶ G) [Mono φ] (hL : ∀ i, (L i).IsLB) (hF : FiniteCohTwists k F) (hG : FiniteCohTwists k G) : interNum k (cokernel φ) L = interNum k G L - interNum k F L

theorem **interNum_eq_of_chiPic_eq** {H : Z.Modules} (h : chiPic k F = chiPic k H ∘ pull f) (hL : ∀ i, (L i).IsLB) : interNum k F L = interNum k H fun i ↦ (L i).pull f

theorem **interNum_pushforward_of_isClosedImmersion** [IsClosedImmersion f] (H : Z.Modules) (hL : ∀ i, (L i).IsLB) : interNum k ((Scheme.Modules.pushforward f).obj H) L = interNum k H fun i ↦ (L i).pull f

theorem **interNum_pushforward_of_isAffineHom** [IsAffineHom f] (H : Z.Modules) [H.IsQuasicoherent] (hL : ∀ i, (L i).IsLB) : interNum k ((Scheme.Modules.pushforward f).obj H) L = interNum k H fun i ↦ (L i).pull f

theorem **interNumOn_eq_interNum_pull** [IsAffineHom f] (hL : ∀ i, (L i).IsLB) : interNumOn k f L = interNum k (structureModule Z) fun i ↦ (L i).pull f

theorem **interNumOn_const_eq_topSelf** [IsAffineHom f] (hM : M.IsLB) (d : ℕ) : interNumOn k f (fun _ : Fin d ↦ M) = topSelfIntersection k (M.pull f) d

theorem **interNumOn_comp** [IsAffineHom f] (g : Y ⟶ Z) [QuasiCompact g] [QuasiSeparated g] (hL : ∀ i, (L i).IsLB) : interNumOn k (g ≫ f) L = interNumOn k g fun i ↦ (L i).pull f

theorem **chi_structureModule_Spec** : chi k (structureModule (Spec A)) = Module.finrank k A

theorem **isScalarTower_algebraΓSpecOf** : letI

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.HdimEqZeroOfSubsingleton

lemma **action_apply** (F : X.Modules) (U : X.Opens) (n : ℕ) (r : Γ(X, ⊤)) (x : Sheaf.H'.{u} ((toAbSheaf X).obj F) n U) : action F U n r x = SiteCoh.map ((toAbSheaf X).map (smulHom F r)) n U x

lemma **toH'_smul** {F : X.Modules} {U : X.Opens} {n : ℕ} (r : R) (x : cohOn F U n) : toH' F U n (r • x) = SiteCoh.map ((toAbSheaf X).map (smulHom F (algebraMap R Γ(X, ⊤) r))) n U (toH' F U n x)

lemma **toH'_map** (φ : F ⟶ G) (x : cohOn F U n) : toH' G U n (map R φ U n x) = SiteCoh.map ((toAbSheaf X).map φ) n U (toH' F U n x)

lemma **toH'_res** (h : V ≤ U) (x : cohOn F U n) : toH' F V n (res R F h n x) = SiteCoh.res ((toAbSheaf X).obj F) n (homOfLE h) (toH' F U n x)

lemma **map_eq_map** (S : Type t) [CommRing S] [Algebra S Γ(X, ⊤)] (φ : F ⟶ G) (x : cohOn F U n) : map R φ U n x = map S φ U n x

lemma **res_eq_res** (S : Type t) [CommRing S] [Algebra S Γ(X, ⊤)] (h : V ≤ U) (x : cohOn F U n) : res R F h n x = res S F h n x

lemma **smul_eq_map** (r : R) (x : cohOn F U n) : r • x = map R (smulHom F (algebraMap R Γ(X, ⊤) r)) U n x

@[simp] lemma **map_smulHom_apply** (r : Γ(X, ⊤)) (x : cohOn F U n) : map R (smulHom F r) U n x = r • x

@[simp] lemma **map_id** : map R (𝟙 F) U n = LinearMap.id

lemma **map_comp** (φ : F ⟶ G) (ψ : G ⟶ H) : map R (φ ≫ ψ) U n = map R ψ U n ∘ₗ map R φ U n

lemma **map_map** (φ : F ⟶ G) (ψ : G ⟶ H) (x : cohOn F U n) : map R ψ U n (map R φ U n x) = map R (φ ≫ ψ) U n x

lemma **map_add_hom** (φ ψ : F ⟶ G) : map R (φ + ψ) U n = map R φ U n + map R ψ U n

@[simp] lemma **map_zero_hom** : map R (0 : F ⟶ G) U n = 0

lemma **map_neg_hom** (φ : F ⟶ G) : map R (-φ) U n = -map R φ U n

lemma **map_sub_hom** (φ ψ : F ⟶ G) : map R (φ - ψ) U n = map R φ U n - map R ψ U n

lemma **map_sum_hom** {ι : Type t} (s : Finset ι) (φ : ι → (F ⟶ G)) : map R (∑ i ∈ s, φ i) U n = ∑ i ∈ s, map R (φ i) U n

lemma **map_smul_hom** (r : Γ(X, ⊤)) (φ : F ⟶ G) : map R (r • φ) U n = r • map R φ U n

@[simp] lemma **res_refl** : res R F (le_refl U) n = LinearMap.id

lemma **res_comp_res** (h : V ≤ U) (h' : W ≤ V) : res R F h' n ∘ₗ res R F h n = res R F (h'.trans h) n

@[simp] lemma **res_res** (h : V ≤ U) (h' : W ≤ V) (x : cohOn F U n) : res R F h' n (res R F h n x) = res R F (h'.trans h) n x

lemma **res_map** (φ : F ⟶ G) (h : V ≤ U) (x : cohOn F U n) : res R G h n (map R φ U n x) = map R φ V n (res R F h n x)

lemma **res_comp_map** (φ : F ⟶ G) (h : V ≤ U) : res R G h n ∘ₗ map R φ U n = map R φ V n ∘ₗ res R F h n

lemma **map_bijective** (φ : F ⟶ G) [IsIso φ] : Function.Bijective (map R φ U n)

lemma **moduleOfRingHom_smul** {R : Type v} [CommRing R] (f : R →+* Γ(X, ⊤)) {F : X.Modules} {U : X.Opens} {n : ℕ} (r : R) (x : cohOn F U n) : letI

lemma **subsingleton_of_isZero** {F : X.Modules} (h : IsZero F) (U : X.Opens) (n : ℕ) : Subsingleton (cohOn F U n)

lemma **subsingleton_of_injective** (F : X.Modules) [Injective ((toAbSheaf X).obj F)] (U : X.Opens) (n : ℕ) : Subsingleton (cohOn F U (n + 1))

lemma **pi_map_bijective** [Finite ι] {F : ι → X.Modules} {c : Fan F} (hc : IsLimit c) : Function.Bijective (LinearMap.pi fun i ↦ map R (c.proj i) U n)

@[simp] lemma **fanEquiv_apply** [Finite ι] {F : ι → X.Modules} {c : Fan F} (hc : IsLimit c) (x : cohOn c.pt U n) (i : ι) : fanEquiv R U n hc x i = map R (c.proj i) U n x

@[simp] lemma **piEquiv_apply** [Finite ι] (F : ι → X.Modules) (x : cohOn (∏ᶜ F) U n) (i : ι) : piEquiv R U n F x i = map R (Pi.π F i) U n x

@[simp] lemma **biproductEquiv_apply** [Finite ι] (F : ι → X.Modules) (x : cohOn (⊕ F) U n) (i : ι) : biproductEquiv R U n F x i = map R (biproduct.π F i) U n x

lemma **action_apply** (F : X.Modules) (U : X.Opens) (r : Γ(X, ⊤)) (x : Γ(F, U)) : action F U r x = restrictGlobal X U r • x

lemma **toΓ_smul** {F : X.Modules} {U : X.Opens} (r : R) (x : sectionsOn F U) : toΓ F U (r • x) = restrictGlobal X U (algebraMap R Γ(X, ⊤) r) • toΓ F U x

@[simp] lemma **toΓ_map** (φ : F ⟶ G) (x : sectionsOn F U) : toΓ G U (map R φ U x) = φ.app U (toΓ F U x)

@[simp] lemma **toΓ_res** (h : V ≤ U) (x : sectionsOn F U) : toΓ F V (res R F h x) = F.presheaf.map (homOfLE h).op (toΓ F U x)

@[simp] lemma **map_id** : map R (𝟙 F) U = LinearMap.id

lemma **map_comp** (φ : F ⟶ G) (ψ : G ⟶ H) : map R (φ ≫ ψ) U = map R ψ U ∘ₗ map R φ U

lemma **map_add_hom** (φ ψ : F ⟶ G) : map R (φ + ψ) U = map R φ U + map R ψ U

@[simp] lemma **res_refl** : res R F (le_refl U) = LinearMap.id

lemma **res_comp_res** (h : V ≤ U) (h' : W ≤ V) : res R F h' ∘ₗ res R F h = res R F (h'.trans h)

lemma **res_map** (φ : F ⟶ G) (h : V ≤ U) (x : sectionsOn F U) : res R G h (map R φ U x) = map R φ V (res R F h x)

@[simp] lemma **equivOfSMul_apply** [Module R Γ(F, U)] (h : ∀ (r : R) (x : Γ(F, U)), r • x = restrictGlobal X U (algebraMap R Γ(X, ⊤) r) • x) (x : sectionsOn F U) : equivOfSMul R F U h x = toΓ F U x

lemma **toΓ_zeroEquiv** (x : cohOn F U 0) : sectionsOn.toΓ F U (zeroEquiv R F U x) = SiteCoh.zeroEquiv ((toAbSheaf X).obj F) U (toH' F U 0 x)

lemma **zeroEquiv_map** (φ : F ⟶ G) (x : cohOn F U 0) : zeroEquiv R G U (map R φ U 0 x) = sectionsOn.map R φ U (zeroEquiv R F U x)

lemma **zeroEquiv_res** (h : V ≤ U) (x : cohOn F U 0) : zeroEquiv R F V (res R F h 0 x) = sectionsOn.res R F h (zeroEquiv R F U x)

lemma **zeroAddEquiv_smul** (r : R) (x : cohOn F U 0) : zeroAddEquiv F U (r • x) = restrictGlobal X U (algebraMap R Γ(X, ⊤) r) • zeroAddEquiv F U x

lemma **zeroEquiv_map** {F G : X.Modules} (φ : F ⟶ G) (x : coh F 0) : zeroEquiv G (cohOn.map Γ(X, ⊤) φ ⊤ 0 x) = φ.app ⊤ (zeroEquiv F x)

lemma **toH_map** (R : Type v) [CommRing R] [Algebra R Γ(X, ⊤)] {F G : X.Modules} (φ : F ⟶ G) (n : ℕ) (x : coh F n) : toH G n (cohOn.map R φ ⊤ n x) = Sheaf.H.map ((toAbSheaf X).map φ) n (toH F n x)

lemma **equiv₀_toH** (F : X.Modules) (x : coh F 0) : Sheaf.H.equiv₀ ((toAbSheaf X).obj F) isTerminalTop (toH F 0 x) = zeroEquiv F x

lemma **algebraΓOfHom_algebraMap** {R : Type u} [CommRing R] (f : X ⟶ Spec (.of R)) (r : R) : letI

lemma **algebraΓOfHom_id** (R : Type u) [CommRing R] : algebraΓOfHom (𝟙 (Spec (.of R))) = algebraΓSpec (.of R)

lemma **hdim_def** (F : X.Modules) : hdim k F n = Module.finrank k (coh F n)

lemma **hdim_eq_rank** [Module.Finite k (coh F n)] : (hdim k F n : Cardinal) = Module.rank k (coh F n)

lemma **hdim_eq_zero_iff** [Module.Finite k (coh F n)] : hdim k F n = 0 ↔ Subsingleton (coh F n)

lemma **hdim_eq_zero_of_subsingleton** [Subsingleton (coh F n)] : hdim k F n = 0

lemma **hdim_congr** (e : F ≅ G) : hdim k F n = hdim k G n

lemma **hdim_biprod** [Module.Finite k (coh F n)] [Module.Finite k (coh G n)] : hdim k (F ⊞ G) n = hdim k F n + hdim k G n

lemma **hdim_pi** {ι : Type t} [Fintype ι] (F : ι → X.Modules) [∀ i, Module.Finite k (coh (F i) n)] : hdim k (∏ᶜ F) n = ∑ i, hdim k (F i) n

lemma **hdim_zero** (F : X.Modules) : hdim k F 0 = Module.finrank k (sectionsOn F ⊤)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.HomologyRestrictScalarsBijective

theorem **_root_.GlobalTheory.iCycles_cechCyclesIso_inv** (A : _root_.GlobalTheory.Cube.{v} ι R) (p : ℕ) (x : LinearMap.ker (_root_.GlobalTheory.Cube.d A (p + 1))) : ((_root_.GlobalTheory.cechComplex A).iCycles p).hom ((_root_.GlobalTheory.cechCyclesIso A p).inv.hom x) = x.1

theorem **_root_.GlobalTheory.cechClass_surjective** (A : _root_.GlobalTheory.Cube.{v} ι R) (p : ℕ) : Function.Surjective (_root_.GlobalTheory.cechClass A p)

theorem **cechClass_eq_zero_iff_prev** (A : Cube.{v} ι R) (q p : ℕ) (h : (ComplexShape.up ℕ).prev p = q) (x : LinearMap.ker (Cube.d A (p + 1))) : cechClass A p x = 0 ↔ ∃ y, ((cechComplex A).d q p).hom y = x.1

theorem **_root_.GlobalTheory.cechClass_succ_eq_zero_iff** (A : _root_.GlobalTheory.Cube.{v} ι R) (p : ℕ) (x : LinearMap.ker (_root_.GlobalTheory.Cube.d A (p + 1 + 1))) : _root_.GlobalTheory.cechClass A (p + 1) x = 0 ↔ ∃ y, _root_.GlobalTheory.Cube.d A (p + 1) y = x.1

private theorem **cechClass_zero_eq_iso_inv** (A : _root_.GlobalTheory.Cube.{v} ι R) : _root_.GlobalTheory.cechClass A 0 = (_root_.GlobalTheory.cechHomologyZeroIso A).inv.hom

private theorem **cechClass_zero_bijective'** (A : _root_.GlobalTheory.Cube.{v} ι R) : Function.Bijective (_root_.GlobalTheory.cechClass A 0)

theorem **_root_.GlobalTheory.cechClass_zero_eq_zero_iff** (A : _root_.GlobalTheory.Cube.{v} ι R) (x : LinearMap.ker (_root_.GlobalTheory.Cube.d A (0 + 1))) : _root_.GlobalTheory.cechClass A 0 x = 0 ↔ x = 0

theorem **_root_.GlobalTheory.cechClass_zero_bijective** (A : _root_.GlobalTheory.Cube.{v} ι R) : Function.Bijective (_root_.GlobalTheory.cechClass A 0)

theorem **cechHomologyZeroIso_cechClass** (A : Cube.{v} ι R) (x : LinearMap.ker (Cube.d A (0 + 1))) : (cechHomologyZeroIso A).hom.hom (cechClass A 0 x) = x

theorem **_root_.GlobalTheory.homologyMap_cechClass** {A B : _root_.GlobalTheory.Cube.{v} ι R} (φ : A ⟶ B) (p : ℕ) (x : LinearMap.ker (_root_.GlobalTheory.Cube.d A (p + 1))) (y : LinearMap.ker (_root_.GlobalTheory.Cube.d B (p + 1))) (h : _root_.GlobalTheory.Cube.termMap φ (p + 1) x.1 = y.1) : (HomologicalComplex.homologyMap (_root_.GlobalTheory.cechComplexMap φ) p).hom (_root_.GlobalTheory.cechClass A p x) = _root_.GlobalTheory.cechClass B p y

theorem **_root_.GlobalTheory.Cube.termMap_mem_ker** {A B : _root_.GlobalTheory.Cube.{v} ι R} (φ : A ⟶ B) {p : ℕ} {x : _root_.GlobalTheory.Cube.Term A p} (h : x ∈ LinearMap.ker (_root_.GlobalTheory.Cube.d A p)) : _root_.GlobalTheory.Cube.termMap φ p x ∈ LinearMap.ker (_root_.GlobalTheory.Cube.d B p)

theorem **_root_.GlobalTheory.cechAugHomology_apply** (A : _root_.GlobalTheory.Cube.{v} ι R) (a : A.obj ∅) : (_root_.GlobalTheory.cechAugHomology A).hom a = _root_.GlobalTheory.cechClass A 0 ⟨_root_.GlobalTheory.Cube.aug A a, _root_.GlobalTheory.Cube.d_aug A a⟩

theorem **term_apply** (p : ℕ) (x : Cube.Term A p) (s : {s : Finset ι // s.card = p}) : ψ.term p x s = ψ.app s.1 (x s)

theorem **_root_.GlobalTheory.Cube.SemiHom.term_d** (p : ℕ) (x : _root_.GlobalTheory.Cube.Term A p) : ψ.term (p + 1) (_root_.GlobalTheory.Cube.d A p x) = _root_.GlobalTheory.Cube.d B p (ψ.term p x)

theorem **term_aug** (a : A.obj ∅) : ψ.term 1 (Cube.aug A a) = Cube.aug B (ψ.app ∅ a)

theorem **_root_.GlobalTheory.Cube.SemiHom.term_bijective** (h : ∀ s : Finset ι, s.Nonempty → Function.Bijective (ψ.app s)) (p : ℕ) : Function.Bijective (ψ.term (p + 1))

theorem **coe_cocycle** (p : ℕ) (x : LinearMap.ker (Cube.d A p)) : (ψ.cocycle p x).1 = ψ.term p x.1

theorem **_root_.GlobalTheory.Cube.SemiHom.cechClass_cocycle_eq_zero** (p : ℕ) (x : LinearMap.ker (_root_.GlobalTheory.Cube.d A (p + 1))) (h : _root_.GlobalTheory.cechClass A p x = 0) : _root_.GlobalTheory.cechClass B p (ψ.cocycle (p + 1) x) = 0

theorem **_root_.GlobalTheory.Cube.SemiHom.homology_cechClass** (p : ℕ) (x : LinearMap.ker (_root_.GlobalTheory.Cube.d A (p + 1))) : ψ.homology p (_root_.GlobalTheory.cechClass A p x) = _root_.GlobalTheory.cechClass B p (ψ.cocycle (p + 1) x)

theorem **homology_ofHom** (φ : A ⟶ A') (p : ℕ) : (ofHom φ).homology p = (HomologicalComplex.homologyMap (cechComplexMap φ) p).hom

theorem **_root_.GlobalTheory.Cube.SemiHom.homology_bijective** (h : ∀ s : Finset ι, s.Nonempty → Function.Bijective (ψ.app s)) (p : ℕ) : Function.Bijective (ψ.homology p)

theorem **_root_.GlobalTheory.Cube.SemiHom.homology_naturality** (ψ' : _root_.GlobalTheory.Cube.SemiHom f A' B') (α : A ⟶ A') (β : B ⟶ B') (h : ∀ s, s.Nonempty → ∀ x, ψ'.app s ((α.app s).hom x) = (β.app s).hom (ψ.app s x)) (p : ℕ) (x : (_root_.GlobalTheory.cechComplex A).homology p) : ψ'.homology p ((HomologicalComplex.homologyMap (_root_.GlobalTheory.cechComplexMap α) p).hom x) = (HomologicalComplex.homologyMap (_root_.GlobalTheory.cechComplexMap β) p).hom (ψ.homology p x)

theorem **homology_homology** {Q : Type*} [Ring Q] {g : S →+* Q} {k : R →+* Q} {C : Cube ι Q} (χ : Cube.SemiHom g B C) (ω : Cube.SemiHom k A C) (h : ∀ s, s.Nonempty → ∀ x, χ.app s (ψ.app s x) = ω.app s x) (p : ℕ) (x : (cechComplex A).homology p) : χ.homology p (ψ.homology p x) = ω.homology p x

theorem **_root_.GlobalTheory.Cube.SemiHom.homology_cechAugHomology** (a : A.obj ∅) : ψ.homology 0 ((_root_.GlobalTheory.cechAugHomology A).hom a) = (_root_.GlobalTheory.cechAugHomology B).hom (ψ.app ∅ a)

theorem **homology_restrictScalars_bijective** (f : R →+* S) (B : Cube.{v} ι S) (p : ℕ) : Function.Bijective ((restrictScalars f B).homology p)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.InterNumConsEqCokernel

theorem **interNum_cons_eq_cokernel** (F : X.Modules) {M : LB X} (hM : M.IsLB) (σ : LB.Sec M) [Mono (cut F hM σ)] (hF : FiniteCohTwists k F) {d : ℕ} {P : Fin d → LB X} (hP : ∀ i, (P i).IsLB) : interNum k F (Fin.cons M P : Fin (d + 1) → LB X) = interNum k (cokernel (cut F hM σ)) P

theorem **interNum_one_eq_hdim** (F : X.Modules) {M : LB X} (hM : M.IsLB) (σ : LB.Sec M) [Mono (cut F hM σ)] (hF : FiniteCohTwists k F) (h : ∀ n, 0 < n → Subsingleton (coh (cokernel (cut F hM σ)) n)) : interNum k F ![M] = hdim k (cokernel (cut F hM σ)) 0

theorem **interNumSub_bot** (L : ι → LB X) : interNumSub k ⊥ L = interNum k (structureModule X) L

theorem **radical_ker** (f : Z ⟶ X) [QuasiCompact f] [IsReduced Z] : f.ker.radical = f.ker

theorem **interNumRed_eq_interNumOn** (f : Z ⟶ X) [IsClosedImmersion f] [IsReduced Z] (C : Closeds X) (h : Set.range f = C) (L : ι → LB X) : interNumRed k C L = interNumOn k f L

theorem **isZero_of_isEmpty** [IsEmpty X] (F : X.Modules) : IsZero F

theorem **interNumSub_top** {L : ι → LB X} (hL : ∀ i, (L i).IsLB) : interNumSub k ⊤ L = 0

theorem **interNumRed_bot** {L : ι → LB X} (hL : ∀ i, (L i).IsLB) : interNumRed k ⊥ L = 0

theorem **plane_values** (F : X.Modules) {M : LB X} (hM : M.IsLB) (h : ∀ n : ℤ, 2 * chi k (twistZ F M n) = (n + 1) * (n + 2)) : interNum k F (fun _ : Fin 2 ↦ M) = 1 ∧ interNum k F (fun _ : Fin 3 ↦ M) = 0 ∧ interNum k F (fun _ : Fin 1 ↦ M) = 1

theorem **dualNumber_example** : letI

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.IsGradedMapLinearCombination

theorem **grProj_sum** (𝓜 : ℤ → Submodule R M) [Decomposition 𝓜] (T : Finset ℤ) (v : ℤ → M) (hv : ∀ j ∈ T, v j ∈ 𝓜 j) {n : ℤ} (hn : n ∈ T) : proj 𝓜 n (∑ j ∈ T, v j) = v n

theorem **exists_sum_grProj** (𝓜 : ℤ → Submodule R M) [Decomposition 𝓜] (m : M) : ∃ T : Finset ℤ, ∑ j ∈ T, proj 𝓜 j m = m

theorem **isInternal_of_sum_eq_zero** (𝓜 : ℤ → Submodule R M) (h₁ : ∀ (T : Finset ℤ) (v : ℤ → M), (∀ j ∈ T, v j ∈ 𝓜 j) → ∑ j ∈ T, v j = 0 → ∀ j ∈ T, v j = 0) (h₂ : ∀ m, m ∈ ⊔ j, 𝓜 j) : IsInternal 𝓜

theorem **weight_one_int** (a : ι →₀ ℕ) : Finsupp.weight (fun _ : ι => (1 : ℤ)) a = (a.degree : ℤ)

theorem **mem_polyGrading_iff** {p : MvPolynomial ι R} {n : ℤ} : p ∈ polyGrading ι R n ↔ ∀ a, coeff a p ≠ 0 → (a.degree : ℤ) = n

theorem **mem_polyGrading_natCast** {p : MvPolynomial ι R} {d : ℕ} : p ∈ polyGrading ι R d ↔ p.IsHomogeneous d

theorem **polyGrading_natCast** (d : ℕ) : polyGrading ι R d = homogeneousSubmodule ι R d

theorem **polyGrading_of_neg** {n : ℤ} (hn : n < 0) : polyGrading ι R n = ⊥

theorem **X_mem_polyGrading** (i : ι) : (X i : MvPolynomial ι R) ∈ polyGrading ι R 1

theorem **monomial_mem_polyGrading** (a : ι →₀ ℕ) (r : R) : monomial a r ∈ polyGrading ι R (a.degree : ℤ)

theorem **grSMul_mem** (𝓜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝓜] {p : MvPolynomial ι R} {d n : ℤ} {m : M} (hp : p ∈ polyGrading ι R d) (hm : m ∈ 𝓜 n) : p • m ∈ 𝓜 (n + d)

theorem **X_smul_mem_gr** (𝓜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝓜] (i : ι) (n : ℤ) (m : M) (hm : m ∈ 𝓜 n) : (X i : MvPolynomial ι R) • m ∈ 𝓜 (n + 1)

theorem **polyGradedSMul_of_X_smul_mem** [IsScalarTower R (MvPolynomial ι R) M] (𝓜 : ℤ → Submodule R M) (hX : ∀ (i : ι) (n : ℤ), ∀ m ∈ 𝓜 n, (X i : MvPolynomial ι R) • m ∈ 𝓜 (n + 1)) : PolyGradedSMul ι 𝓜 where

theorem **polyGradedSMul_iff** [IsScalarTower R (MvPolynomial ι R) M] (𝓜 : ℤ → Submodule R M) : PolyGradedSMul ι 𝓜 ↔ ∀ (i : ι) (n : ℤ), ∀ m ∈ 𝓜 n, (X i : MvPolynomial ι R) • m ∈ 𝓜 (n + 1)

theorem **id** : IsGradedMap 𝓜 𝓜 (LinearMap.id : M →ₗ[MvPolynomial ι R] M)

theorem **zero** : IsGradedMap 𝓜 𝓝 (0 : M →ₗ[MvPolynomial ι R] N)

theorem **comp** {g : N →ₗ[MvPolynomial ι R] P} {f : M →ₗ[MvPolynomial ι R] N} (hg : IsGradedMap 𝓝 𝓠 g) (hf : IsGradedMap 𝓜 𝓝 f) : IsGradedMap 𝓜 𝓠 (g ∘ₗ f)

theorem **add** {f g : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) (hg : IsGradedMap 𝓜 𝓝 g) : IsGradedMap 𝓜 𝓝 (f + g)

theorem **neg** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) : IsGradedMap 𝓜 𝓝 (-f)

theorem **sum** {κ : Type*} (T : Finset κ) {f : κ → M →ₗ[MvPolynomial ι R] N} (hf : ∀ k ∈ T, IsGradedMap 𝓜 𝓝 (f k)) : IsGradedMap 𝓜 𝓝 (∑ k ∈ T, f k)

theorem **coe_piece** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) (n : ℤ) (m : 𝓜 n) : (hf.piece n m : N) = f m

theorem **proj_apply** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) (n : ℤ) (m : M) : proj 𝓝 n (f m) = f (proj 𝓜 n m)

theorem **isHomogeneous_ker** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) : (LinearMap.ker f).IsHomogeneous 𝓜

theorem **isHomogeneous_range** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) : (LinearMap.range f).IsHomogeneous 𝓝

theorem **exists_mem_of_mem_range** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝓜 𝓝 f) {n : ℤ} {y : N} (hy : y ∈ 𝓝 n) (h : y ∈ LinearMap.range f) : ∃ m ∈ 𝓜 n, f m = y

theorem **injective_iff** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 ℳ f) : Function.Injective f ↔ ∀ n, Function.Injective (hf.piece n)

theorem **surjective_iff** {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 ℳ f) : Function.Surjective f ↔ ∀ n, Function.Surjective (hf.piece n)

theorem **exact_piece** {f : M →ₗ[MvPolynomial ι R] N} {g : N →ₗ[MvPolynomial ι R] P} (hf : IsGradedMap 𝒜 ℳ f) (hg : IsGradedMap ℳ 𝒬 g) (h : Function.Exact f g) (n : ℤ) : Function.Exact (hf.piece n) (hg.piece n)

theorem **exact_of_exact_piece** [Decomposition 𝒬] {f : M →ₗ[MvPolynomial ι R] N} {g : N →ₗ[MvPolynomial ι R] P} (hf : IsGradedMap 𝒜 ℳ f) (hg : IsGradedMap ℳ 𝒬 g) (h : ∀ n, Function.Exact (hf.piece n) (hg.piece n)) : Function.Exact f g

theorem **symm** {e : M ≃ₗ[MvPolynomial ι R] N} (he : IsGradedMap 𝒜 ℳ e.toLinearMap) : IsGradedMap ℳ 𝒜 e.symm.toLinearMap

theorem **grShift_apply** (𝒜 : ℤ → Submodule R M) (d n : ℤ) : grShift 𝒜 d n = 𝒜 (n + d)

theorem **grShift_zero** (𝒜 : ℤ → Submodule R M) : grShift 𝒜 0 = 𝒜

theorem **grShift_grShift** (𝒜 : ℤ → Submodule R M) (d e : ℤ) : grShift (grShift 𝒜 d) e = grShift 𝒜 (d + e)

theorem **proj_grShift** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] (d n : ℤ) (m : M) : proj (grShift 𝒜 d) n m = proj 𝒜 (n + d) m

theorem **IsGradedMap.shift** {𝒜 : ℤ → Submodule R M} {ℳ : ℤ → Submodule R N} {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 ℳ f) (d : ℤ) : IsGradedMap (grShift 𝒜 d) (grShift ℳ d) f

theorem **isGradedMap_lsmul** (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) : IsGradedMap 𝒜 (grShift 𝒜 d) (LinearMap.lsmul (MvPolynomial ι R) M p)

theorem **isGradedMap_lsmul'** (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) : IsGradedMap (grShift 𝒜 (-d)) 𝒜 (LinearMap.lsmul (MvPolynomial ι R) M p)

theorem **isGradedMap_toSpanSingleton** (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] {m : M} {d : ℤ} (hm : m ∈ 𝒜 d) : IsGradedMap (grShift (polyGrading ι R) (-d)) 𝒜 (LinearMap.toSpanSingleton (MvPolynomial ι R) M m)

theorem **proj_smul_of_mem** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] [PolyGradedSMul ι 𝒜] {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) (n : ℤ) (m : M) : proj 𝒜 (n + d) (p • m) = p • proj 𝒜 n m

theorem **proj_smul_of_mem_right** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] [PolyGradedSMul ι 𝒜] (p : MvPolynomial ι R) {d : ℤ} {m : M} (hm : m ∈ 𝒜 d) (n : ℤ) : proj 𝒜 n (p • m) = proj (polyGrading ι R) (n - d) p • m

theorem **mem_grSub** {𝒜 : ℤ → Submodule R M} {K : Submodule (MvPolynomial ι R) M} {n : ℤ} {m : K} : m ∈ grSub 𝒜 K n ↔ (m : M) ∈ 𝒜 n

theorem **mem_grQuot** {𝒜 : ℤ → Submodule R M} {K : Submodule (MvPolynomial ι R) M} {n : ℤ} {y : M ⧸ K} : y ∈ grQuot 𝒜 K n ↔ ∃ m ∈ 𝒜 n, Submodule.Quotient.mk m = y

theorem **isGradedMap_subtype** (𝒜 : ℤ → Submodule R M) (K : Submodule (MvPolynomial ι R) M) : IsGradedMap (grSub 𝒜 K) 𝒜 K.subtype

theorem **isGradedMap_mkQ** (𝒜 : ℤ → Submodule R M) (K : Submodule (MvPolynomial ι R) M) : IsGradedMap 𝒜 (grQuot 𝒜 K) K.mkQ

theorem **IsGradedMap.codRestrict** {𝒜 : ℤ → Submodule R M} {ℳ : ℤ → Submodule R N} {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 ℳ f) (K : Submodule (MvPolynomial ι R) N) (h : ∀ m, f m ∈ K) : IsGradedMap 𝒜 (grSub ℳ K) (f.codRestrict K h)

theorem **IsGradedMap.liftQ** {𝒜 : ℤ → Submodule R M} {ℳ : ℤ → Submodule R N} {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 ℳ f) (K : Submodule (MvPolynomial ι R) M) (h : K ≤ LinearMap.ker f) : IsGradedMap (grQuot 𝒜 K) ℳ (K.liftQ f h)

theorem **isInternal_grSub** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] {K : Submodule (MvPolynomial ι R) M} (hK : K.IsHomogeneous 𝒜) : IsInternal (grSub 𝒜 K)

theorem **isInternal_grQuot** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] {K : Submodule (MvPolynomial ι R) M} (hK : K.IsHomogeneous 𝒜) : IsInternal (grQuot 𝒜 K)

theorem **isHomogeneous_span_gr** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] [PolyGradedSMul ι 𝒜] {G : Set M} (hG : ∀ g ∈ G, ∃ d, g ∈ 𝒜 d) : (Submodule.span (MvPolynomial ι R) G).IsHomogeneous 𝒜

theorem **mem_grPi** {𝒜 : ∀ k, ℤ → Submodule R (M k)} {n : ℤ} {m : ∀ k, M k} : m ∈ grPi 𝒜 n ↔ ∀ k, m k ∈ 𝒜 k n

theorem **isGradedMap_piProj** (𝒜 : ∀ k, ℤ → Submodule R (M k)) (k : κ) : IsGradedMap (grPi 𝒜) (𝒜 k) (LinearMap.proj (R

theorem **isGradedMap_piSingle** [DecidableEq κ] (𝒜 : ∀ k, ℤ → Submodule R (M k)) (k : κ) : IsGradedMap (𝒜 k) (grPi 𝒜) (LinearMap.single (MvPolynomial ι R) M k)

theorem **IsGradedMap.pi** {𝒩 : ℤ → Submodule R N} {ℳ : ∀ k, ℤ → Submodule R (M k)} {f : ∀ k, N →ₗ[MvPolynomial ι R] M k} (hf : ∀ k, IsGradedMap 𝒩 (ℳ k) (f k)) : IsGradedMap 𝒩 (grPi ℳ) (LinearMap.pi f)

theorem **isGradedMap_linearCombination** [Fintype κ] (𝒩 : ℤ → Submodule R N) [PolyGradedSMul ι 𝒩] {g : κ → N} {d : κ → ℤ} (hg : ∀ k, g k ∈ 𝒩 (d k)) : IsGradedMap (grFree ι R d) 𝒩 (Fintype.linearCombination (MvPolynomial ι R) g)

theorem **exists_graded_surjection** (𝒩 : ℤ → Submodule R N) [Decomposition 𝒩] [PolyGradedSMul ι 𝒩] [Module.Finite (MvPolynomial ι R) N] : ∃ (r : ℕ) (d : Fin r → ℤ) (f : (Fin r → MvPolynomial ι R) →ₗ[MvPolynomial ι R] N), IsGradedMap (grFree ι R d) 𝒩 f ∧ Function.Surjective f

theorem **component_of** (n : ℤ) (a : A n) : component hx n (of hx n a) = a

theorem **component_of_ne** {n k : ℤ} (h : k ≠ n) (a : A k) : component hx n (of hx k a) = 0

theorem **of_injective** (n : ℤ) : Function.Injective (of hx n)

theorem **lhom_ext** {N : Type*} [AddCommGroup N] [Module R N] {f g : GrSum hx →ₗ[R] N} (h : ∀ n a, f (of hx n a) = g (of hx n a)) : f = g

theorem **desc_of** {N : Type*} [AddCommGroup N] [Module R N] (φ : ∀ n, A n →ₗ[R] N) (n : ℤ) (a : A n) : desc hx φ (of hx n a) = φ n a

theorem **op_of** (i : ι) (n : ℤ) (a : A n) : op hx i (of hx n a) = of hx (n + 1) (x i n a)

theorem **commute_op** (i j : ι) : Commute (op hx i) (op hx j)

theorem **smul_def** (p : MvPolynomial ι R) (m : GrSum hx) : p • m = Wick.commEval R (op hx) (commute_op hx) p m

theorem **X_smul** (i : ι) (m : GrSum hx) : (X i : MvPolynomial ι R) • m = op hx i m

theorem **X_smul_of** (i : ι) (n : ℤ) (a : A n) : (X i : MvPolynomial ι R) • of hx n a = of hx (n + 1) (x i n a)

theorem **mem_grading** {n : ℤ} {m : GrSum hx} : m ∈ grading hx n ↔ ∃ a, of hx n a = m

theorem **of_mem_grading** (n : ℤ) (a : A n) : of hx n a ∈ grading hx n

theorem **coe_pieceEquiv** (n : ℤ) (a : A n) : (pieceEquiv hx n a : GrSum hx) = of hx n a

theorem **proj_grading** (n : ℤ) (m : GrSum hx) : proj (grading hx) n m = of hx n (component hx n m)

theorem **lift_of** (φ : ∀ n, A n →ₗ[R] N) (hφ) (n : ℤ) (a : A n) : lift hx φ hφ (of hx n a) = φ n a

theorem **isGradedMap_lift** (φ : ∀ n, A n →ₗ[R] N) (hφ) (𝒩 : ℤ → Submodule R N) (h : ∀ n a, φ n a ∈ 𝒩 n) : IsGradedMap (grading hx) 𝒩 (lift hx φ hφ)

theorem **map_of** (φ : ∀ n, A n →ₗ[R] B n) (hφ) (n : ℤ) (a : A n) : map hx hy φ hφ (of hx n a) = of hy n (φ n a)

theorem **isGradedMap_map** (φ : ∀ n, A n →ₗ[R] B n) (hφ) : IsGradedMap (grading hx) (grading hy) (map hx hy φ hφ)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.IsIsoCechComplexMapPresheafCubeRes

theorem **cechOpen_def** (s : Finset ι) : cechOpen V U s = V ⊓ ⨅ i ∈ s, U i

theorem **cechOpen_eq_inf** (s : Finset ι) : cechOpen V U s = V ⊓ s.inf U

theorem **le_cechOpen_iff** {W : Opens T} {s : Finset ι} : W ≤ cechOpen V U s ↔ W ≤ V ∧ ∀ i ∈ s, W ≤ U i

theorem **mem_cechOpen** {x : T} {s : Finset ι} : x ∈ cechOpen V U s ↔ x ∈ V ∧ ∀ i ∈ s, x ∈ U i

theorem **cechOpen_le** (s : Finset ι) : cechOpen V U s ≤ V

theorem **cechOpen_le_of_mem** {s : Finset ι} {i : ι} (h : i ∈ s) : cechOpen V U s ≤ U i

theorem **cechOpen_anti** {s t : Finset ι} (h : s ⊆ t) : cechOpen V U t ≤ cechOpen V U s

theorem **cechOpen_empty** : cechOpen V U ∅ = V

theorem **cechOpen_singleton** (i : ι) : cechOpen V U {i} = V ⊓ U i

theorem **cechOpen_insert** [DecidableEq ι] (i : ι) (s : Finset ι) : cechOpen V U (insert i s) = U i ⊓ cechOpen V U s

theorem **cechOpen_pair** [DecidableEq ι] (i j : ι) : cechOpen V U {i, j} = V ⊓ (U i ⊓ U j)

theorem **cechOpen_union** [DecidableEq ι] (s t : Finset ι) : cechOpen V U (s ∪ t) = cechOpen V U s ⊓ cechOpen V U t

theorem **cechOpen_top** (s : Finset ι) : cechOpen ⊤ U s = ⨅ i ∈ s, U i

theorem **cechOpen_of_nonempty** (h : ∀ i, U i ≤ V) {s : Finset ι} (hs : s.Nonempty) : cechOpen V U s = ⨅ i ∈ s, U i

theorem **cechOpen_mono** {V' : Opens T} {U' : ι → Opens T} (hV : V' ≤ V) (hU : ∀ i, U' i ≤ U i) (s : Finset ι) : cechOpen V' U' s ≤ cechOpen V U s

theorem **cechOpen_inf_family** (s : Finset ι) : cechOpen V (fun i => V ⊓ U i) s = cechOpen V U s

theorem **cechOpen_inf_left** (W : Opens T) (s : Finset ι) : cechOpen (W ⊓ V) U s = W ⊓ cechOpen V U s

theorem **cechOpen_map** (e : κ ↪ ι) (s : Finset κ) : cechOpen V U (s.map e) = cechOpen V (fun k => U (e k)) s

theorem **cechOpen_insert_map** [DecidableEq ι] (e : κ ↪ ι) (i : ι) (s : Finset κ) : cechOpen V U (insert i (s.map e)) = cechOpen (V ⊓ U i) (fun k => U (e k)) s

theorem **cechOpen_empty_le_iSup** (h : V ≤ ⨆ i, U i) : cechOpen V U ∅ ≤ ⨆ i, cechOpen V U {i}

theorem **presheafRes_self** {W : Opens T} (h : W ≤ W) (x : P.obj (op W)) : presheafRes P h x = x

theorem **presheafRes_presheafRes** {W W' W'' : Opens T} (h : W' ≤ W) (h' : W'' ≤ W') (x : P.obj (op W)) : presheafRes P h' (presheafRes P h x) = presheafRes P (h'.trans h) x

theorem **presheafRes_bijective** {W W' : Opens T} (h : W' ≤ W) (h' : W ≤ W') : Function.Bijective (presheafRes P h)

theorem **presheafCongr_apply** {W W' : Opens T} (h : W = W') (x : P.obj (op W)) : presheafCongr P h x = presheafRes P h.ge x

theorem **presheafCongr_symm_apply** {W W' : Opens T} (h : W = W') (x : P.obj (op W')) : (presheafCongr P h).symm x = presheafRes P h.le x

theorem **app_presheafRes** (φ : P ⟶ Q) {W W' : Opens T} (h : W' ≤ W) (x : P.obj (op W)) : (φ.app (op W')).hom (presheafRes P h x) = presheafRes Q h ((φ.app (op W)).hom x)

theorem **presheafCube_obj** (s : Finset ι) : (presheafCube P V U).obj s = P.obj (op (cechOpen V U s))

theorem **res_presheafCube** {s t : Finset ι} (h : s ⊆ t) : Cube.res (presheafCube P V U) h = presheafRes P (cechOpen_anti V U h)

theorem **presheafCubeMap_app** (φ : P ⟶ Q) (s : Finset ι) : (presheafCubeMap V U φ).app s = φ.app (op (cechOpen V U s))

theorem **presheafCubeRes_app** (h : ∀ s, cechOpen V' U' s ≤ cechOpen V U s) (s : Finset ι) : ((presheafCubeRes P h).app s).hom = presheafRes P (h s)

theorem **presheafCubeRes_self** (h : ∀ s, cechOpen V U s ≤ cechOpen V U s) : presheafCubeRes P h = 𝟙 _

theorem **presheafCubeRes_comp** (h : ∀ s, cechOpen V' U' s ≤ cechOpen V U s) (h' : ∀ s, cechOpen V'' U'' s ≤ cechOpen V' U' s) : presheafCubeRes P h ≫ presheafCubeRes P h' = presheafCubeRes P fun s => (h' s).trans (h s)

theorem **presheafCubeRes_naturality** (φ : P ⟶ Q) (h : ∀ s, cechOpen V' U' s ≤ cechOpen V U s) : presheafCubeMap V U φ ≫ presheafCubeRes Q h = presheafCubeRes P h ≫ presheafCubeMap V' U' φ

theorem **presheafCubeEmptyEquiv_apply** (x : P.obj (op V)) : presheafCubeEmptyEquiv P V U x = presheafRes P (cechOpen_le V U ∅) x

theorem **presheafCubeLowerFaceIso_hom_app** (s : Finset κ) : ((presheafCubeLowerFaceIso P V U e).hom.app s).hom = presheafRes P (cechOpen_map V U e.toEmbedding s).ge

theorem **presheafCubeUpperFaceIso_hom_app** (s : Finset κ) : ((presheafCubeUpperFaceIso P V U e i).hom.app s).hom = presheafRes P (cechOpen_insert_map V U e.toEmbedding i s).ge

theorem **faceMap_presheafCube** : Cube.faceMap e i (presheafCube P V U) ≫ (presheafCubeUpperFaceIso P V U e i).hom = (presheafCubeLowerFaceIso P V U e).hom ≫ presheafCubeRes P (cechOpen_mono inf_le_left fun _ => le_rfl)

theorem **isZero_presheafCech_X** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : IsZero ((presheafCech P V U).X n)

theorem **isZero_presheafCechH** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : IsZero (presheafCechH P V U n)

theorem **isIso_cechComplexMap_presheafCubeRes** {V' : Opens T} {U' : ι → Opens T} (h : ∀ s, cechOpen V' U' s ≤ cechOpen V U s) (h' : ∀ s, s.Nonempty → cechOpen V U s ≤ cechOpen V' U' s) : IsIso (cechComplexMap (presheafCubeRes P h))

theorem **presheafCechTopIso_hom** (h : ∀ i, U i ≤ V) : (presheafCechTopIso P h).hom = cechComplexMap (presheafCubeRes P (cechOpen_mono (le_top (a

theorem **presheafCechAug_apply** (x : P.obj (op V)) (s : {s : Finset ι // s.card = 1}) : presheafCechAug P V U x s = presheafRes P (cechOpen_le V U s.1) x

theorem **aug_presheafCubeEmptyEquiv** (x : P.obj (op V)) : Cube.aug (presheafCube P V U) (presheafCubeEmptyEquiv P V U x) = presheafCechAug P V U x

theorem **d_presheafCechAug** (x : P.obj (op V)) : Cube.d (presheafCube P V U) 1 (presheafCechAug P V U x) = 0

theorem **presheafToCechH_apply** (x : P.obj (op V)) : presheafToCechH P V U x = cechClass (presheafCube P V U) 0 (presheafCechAug P V U x, d_presheafCechAug P V U x)

theorem **homologyMap_presheafToCechH** (φ : P ⟶ Q) (x : P.obj (op V)) : (HomologicalComplex.homologyMap (cechComplexMap (presheafCubeMap V U φ)) 0).hom (presheafToCechH P V U x) = presheafToCechH Q V U ((φ.app (op V)).hom x)

theorem **homologyMap_res_presheafToCechH** {V' : Opens T} {U' : ι → Opens T} (h : ∀ s, cechOpen V' U' s ≤ cechOpen V U s) (hV : V' ≤ V) (x : P.obj (op V)) : (HomologicalComplex.homologyMap (cechComplexMap (presheafCubeRes P h)) 0).hom (presheafToCechH P V U x) = presheafToCechH P V' U' (presheafRes P hV x)

theorem **presheafRes_eq_of_d_eq_zero** (x : Cube.Term (presheafCube P V U) 1) (hx : Cube.d (presheafCube P V U) 1 x = 0) (i j : ι) {W : Opens T} (hi : W ≤ cechOpen V U {i}) (hj : W ≤ cechOpen V U {j}) : presheafRes P hi (x ({i}, Finset.card_singleton i)) = presheafRes P hj (x ({j}, Finset.card_singleton j))

theorem **presheafCechAug_injective** : Function.Injective (presheafCechAug P V U)

theorem **presheafCechAug_exact** : Function.Exact (presheafCechAug P V U) (Cube.d (presheafCube P V U) 1)

theorem **presheafToCechH_bijective** : Function.Bijective (presheafToCechH P V U)

theorem **isIso_cechAugHomology_presheafCube** : IsIso (cechAugHomology (presheafCube P V U))

theorem **presheafCube_exactAt_zero_one** : (cubeComplex (presheafCube P V U)).ExactAt 0 ∧ (cubeComplex (presheafCube P V U)).ExactAt 1

theorem **presheafCechHZeroEquiv_symm_apply** (x : P.obj (op V)) : (presheafCechHZeroEquiv hP hc).symm x = presheafToCechH P V U x

theorem **cechAbCube_obj** (s : Finset ι) : (cechAbCube P V U).obj s = ModuleCat.of ℤ (P.obj (op (cechOpen V U s)))

theorem **res_cechAbCube** {s t : Finset ι} (h : s ⊆ t) (x : P.obj (op (cechOpen V U s))) : Cube.res (cechAbCube P V U) h x = P.map (homOfLE (cechOpen_anti V U h)).op x

theorem **cechAbCubeMap_app_apply** (φ : P ⟶ Q) (s : Finset ι) (x : P.obj (op (cechOpen V U s))) : ((cechAbCubeMap V U φ).app s).hom x = φ.app (op (cechOpen V U s)) x

theorem **cechAb_X** (p : ℕ) : (cechAb P V U).X p = ModuleCat.of ℤ (∀ s : {s : Finset ι // s.card = p + 1}, P.obj (op (cechOpen V U s.1)))

theorem **cechAb_d** (p : ℕ) : (cechAb P V U).d p (p + 1) = ModuleCat.ofHom (Cube.d (cechAbCube P V U) (p + 1))

theorem **cechAbHMap_id** (n : ℕ) : cechAbHMap V U (𝟙 P) n = 𝟙 _

theorem **cechAbHMap_comp** (φ : P ⟶ Q) (ψ : Q ⟶ Q') (n : ℕ) : cechAbHMap V U (φ ≫ ψ) n = cechAbHMap V U φ n ≫ cechAbHMap V U ψ n

theorem **cechAbHMap_add** (φ ψ : P ⟶ Q) (n : ℕ) : cechAbHMap V U (φ + ψ) n = cechAbHMap V U φ n + cechAbHMap V U ψ n

theorem **cechAbFunctor_shortExact** (S : ShortComplex ((Opens T)ᵒᵖ ⥤ AddCommGrpCat.{v})) (h : ∀ s : Finset ι, s.Nonempty → Function.Injective (S.f.app (op (cechOpen V U s))) ∧ Function.Exact (S.f.app (op (cechOpen V U s))) (S.g.app (op (cechOpen V U s))) ∧ Function.Surjective (S.g.app (op (cechOpen V U s)))) : (S.map (cechAbFunctor V U)).ShortExact

theorem **isZero_cechAb_X** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : IsZero ((cechAb P V U).X n)

theorem **isZero_cechAbH** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : IsZero (cechAbH P V U n)

theorem **subsingleton_cechAbH** [Fintype ι] {n : ℕ} (h : Fintype.card ι ≤ n) : Subsingleton (cechAbH P V U n)

theorem **cechAbHMap_toCechAbH** (φ : P ⟶ Q) (x : P.obj (op V)) : (cechAbHMap V U φ 0).hom (toCechAbH P V U x) = toCechAbH Q V U (φ.app (op V) x)

theorem **isSheaf_comp_abToIntModule** (hP : Presheaf.IsSheaf (Opens.grothendieckTopology T) P) : Presheaf.IsSheaf (Opens.grothendieckTopology T) (P ⋙ abToIntModule)

theorem **cechAbCube_exactAt_zero_one** (hP : Presheaf.IsSheaf (Opens.grothendieckTopology T) P) (hc : V ≤ ⨆ i, U i) : (cubeComplex (cechAbCube P V U)).ExactAt 0 ∧ (cubeComplex (cechAbCube P V U)).ExactAt 1

theorem **cechAbHZeroEquiv_symm_apply** (hP : Presheaf.IsSheaf (Opens.grothendieckTopology T) P) (hc : V ≤ ⨆ i, U i) (x : P.obj (op V)) : (cechAbHZeroEquiv hP hc).symm x = toCechAbH P V U x

theorem **cechAbHZeroEquiv_map** (hP : Presheaf.IsSheaf (Opens.grothendieckTopology T) P) (hQ : Presheaf.IsSheaf (Opens.grothendieckTopology T) Q) (hc : V ≤ ⨆ i, U i) (φ : P ⟶ Q) (x : cechAbH P V U 0) : cechAbHZeroEquiv hQ hc ((cechAbHMap V U φ 0).hom x) = φ.app (op V) (cechAbHZeroEquiv hP hc x)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.IsIsoTermComparisonRestrictScalars

theorem **cubeSign_mul_self** (t : Finset ι) (i : ι) : cubeSign t i * cubeSign t i = 1

theorem **cubeSign_eq_or** (t : Finset ι) (i : ι) : cubeSign t i = 1 ∨ cubeSign t i = -1

theorem **cubeSign_smul_smul** {M : Type*} [AddCommGroup M] (t : Finset ι) (i : ι) (x : M) : cubeSign t i • cubeSign t i • x = x

theorem **cubeSign_smul_eq_zero** {M : Type*} [AddCommGroup M] (t : Finset ι) (i : ι) (x : M) : cubeSign t i • x = 0 ↔ x = 0

theorem **cubeSign_eq_one_of_forall_le** {t : Finset ι} {i : ι} (h : ∀ j ∈ t, i ≤ j) : cubeSign t i = 1

theorem **cubeSign_singleton** (i : ι) : cubeSign {i} i = 1

theorem **cubeSign_erase_of_le** (t : Finset ι) {i j : ι} (h : i ≤ j) : cubeSign (t.erase j) i = cubeSign t i

theorem **cubeSign_erase_of_lt** {t : Finset ι} {i j : ι} (hj : j ∈ t) (h : j < i) : cubeSign (t.erase j) i = - cubeSign t i

theorem **cubeSign_mul_erase** {t : Finset ι} {i j : ι} (hi : i ∈ t) (hj : j ∈ t) (h : i ≠ j) : cubeSign t i * cubeSign (t.erase i) j = -(cubeSign t j * cubeSign (t.erase j) i)

theorem **res_self** (A : Cube.{v} ι R) {s : Finset ι} (h : s ⊆ s) (x : A.obj s) : res A h x = x

theorem **res_res** (A : Cube.{v} ι R) {s t w : Finset ι} (h : s ⊆ t) (h' : t ⊆ w) (x : A.obj s) : res A h' (res A h x) = res A (h.trans h') x

theorem **res_congr** (A : Cube.{v} ι R) (x : ∀ s, A.obj s) {s s' t : Finset ι} (e : s = s') (h : s ⊆ t) (h' : s' ⊆ t) : res A h (x s) = res A h' (x s')

theorem **app_res** {A B : Cube.{v} ι R} (φ : A ⟶ B) {s t : Finset ι} (h : s ⊆ t) (x : A.obj s) : (φ.app t).hom (res A h x) = res B h ((φ.app s).hom x)

theorem **res_mk** (X : Finset ι → ModuleCat.{v} R) (ρ : ∀ {s t : Finset ι}, s ⊆ t → X s →ₗ[R] X t) (ρ_self ρ_comp) {s t : Finset ι} (h : s ⊆ t) : res (mk X ρ ρ_self ρ_comp) h = ρ h

theorem **res_const** (M : ModuleCat.{v} R) {s t : Finset ι} (h : s ⊆ t) (x : M) : res (const ι M) h x = x

theorem **extend_apply_of_eq** (A : Cube.{v} ι R) {p : ℕ} (x : Term A p) {s : Finset ι} (h : s.card = p) : extend A p x s = x ⟨s, h⟩

theorem **extend_apply_of_ne** (A : Cube.{v} ι R) {p : ℕ} (x : Term A p) {s : Finset ι} (h : s.card ≠ p) : extend A p x s = 0

theorem **termMap_apply** {A B : Cube.{v} ι R} (φ : A ⟶ B) (p : ℕ) (x : Term A p) (s : {s : Finset ι // s.card = p}) : termMap φ p x s = (φ.app s.1).hom (x s)

theorem **termZeroEquiv_apply** (A : Cube.{v} ι R) (x : Term A 0) : termZeroEquiv A x = x ⟨∅, rfl⟩

theorem **termZeroEquiv_symm_apply** (A : Cube.{v} ι R) (a : A.obj ∅) (s : {s : Finset ι // s.card = 0}) : (termZeroEquiv A).symm a s = res A (Finset.empty_subset s.1) a

theorem **termOneEquiv_symm_apply** (A : Cube.{v} ι R) (x : Term A 1) (i : ι) : (termOneEquiv A).symm x i = x ⟨{i}, Finset.card_singleton i⟩

theorem **termOneEquiv_apply** (A : Cube.{v} ι R) (y : ∀ i : ι, A.obj {i}) (i : ι) : termOneEquiv A y ⟨{i}, Finset.card_singleton i⟩ = y i

theorem **subsingleton_term** [Fintype ι] (A : Cube.{v} ι R) {p : ℕ} (h : Fintype.card ι < p) : Subsingleton (Term A p)

theorem **aug_apply** (A : Cube.{v} ι R) (a : A.obj ∅) (s : {s : Finset ι // s.card = 1}) : aug A a s = res A (Finset.empty_subset s.1) a

theorem **termMap_aug** {A B : Cube.{v} ι R} (φ : A ⟶ B) (a : A.obj ∅) : termMap φ 1 (aug A a) = aug B ((φ.app ∅).hom a)

theorem **d_apply** (A : Cube.{v} ι R) (p : ℕ) (x : Term A p) (t : {s : Finset ι // s.card = p + 1}) : d A p x t = ∑ i : t.1, cubeSign t.1 i.1 • res A (Finset.erase_subset i.1 t.1) (x (eraseIdx t i))

theorem **totalD_apply** (A : Cube.{v} ι R) (x : ∀ s, A.obj s) (t : Finset ι) : totalD A x t = ∑ i ∈ t, cubeSign t i • res A (Finset.erase_subset i t) (x (t.erase i))

theorem **d_apply_eq_totalD** (A : Cube.{v} ι R) {p : ℕ} (x : Term A p) (y : ∀ s, A.obj s) (h : ∀ s : {s : Finset ι // s.card = p}, x s = y s.1) (t : {s : Finset ι // s.card = p + 1}) : d A p x t = totalD A y t.1

theorem **d_apply_extend** (A : Cube.{v} ι R) {p : ℕ} (x : Term A p) (t : {s : Finset ι // s.card = p + 1}) : d A p x t = ∑ i ∈ t.1, cubeSign t.1 i • res A (Finset.erase_subset i t.1) (extend A p x (t.1.erase i))

theorem **totalD_totalD** (A : Cube.{v} ι R) (x : ∀ s, A.obj s) : totalD A (totalD A x) = 0

theorem **d_d** (A : Cube.{v} ι R) (p : ℕ) (x : Term A p) : d A (p + 1) (d A p x) = 0

theorem **d_comp_d** (A : Cube.{v} ι R) (p : ℕ) : d A (p + 1) ∘ₗ d A p = 0

theorem **termMap_d** {A B : Cube.{v} ι R} (φ : A ⟶ B) (p : ℕ) (x : Term A p) : termMap φ (p + 1) (d A p x) = d B p (termMap φ p x)

theorem **d_zero_apply** (A : Cube.{v} ι R) (x : Term A 0) : d A 0 x = aug A (termZeroEquiv A x)

theorem **d_aug** (A : Cube.{v} ι R) (a : A.obj ∅) : d A 1 (aug A a) = 0

theorem **d_one_apply_pair** (A : Cube.{v} ι R) (x : Term A 1) {i j : ι} (h : i < j) : d A 1 x ⟨{i, j}, Finset.card_pair h.ne⟩ = res A (by simp) (x ⟨{j}, Finset.card_singleton j⟩) - res A (by simp) (x ⟨{i}, Finset.card_singleton i⟩)

theorem **eq_zero_of_d_eq_zero** (A : Cube.{v} ι R) (i : ι) {p : ℕ} (hA : ∀ s : Finset ι, i ∉ s → Function.Injective (res A (Finset.subset_insert i s))) (x : Term A p) (hx : d A p x = 0) (h0 : ∀ s : {s : Finset ι // s.card = p}, i ∈ s.1 → x s = 0) : x = 0

theorem **exact_d_of_bijective** (A : Cube.{v} ι R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → Function.Bijective (res A (Finset.subset_insert i s))) (p : ℕ) : Function.Exact (d A p) (d A (p + 1))

theorem **cubeComplex_X** (A : Cube.{v} ι R) (p : ℕ) : (cubeComplex A).X p = ModuleCat.of R (Cube.Term A p)

theorem **cubeComplex_d** (A : Cube.{v} ι R) (p : ℕ) : (cubeComplex A).d p (p + 1) = ModuleCat.ofHom (Cube.d A p)

theorem **cubeComplexMap_f** {A B : Cube.{v} ι R} (φ : A ⟶ B) (p : ℕ) : (cubeComplexMap φ).f p = ModuleCat.ofHom (Cube.termMap φ p)

theorem **cechComplex_X** (A : Cube.{v} ι R) (p : ℕ) : (cechComplex A).X p = ModuleCat.of R (Cube.Term A (p + 1))

theorem **cechComplex_d** (A : Cube.{v} ι R) (p : ℕ) : (cechComplex A).d p (p + 1) = ModuleCat.ofHom (Cube.d A (p + 1))

theorem **cechComplexMap_f** {A B : Cube.{v} ι R} (φ : A ⟶ B) (p : ℕ) : (cechComplexMap φ).f p = ModuleCat.ofHom (Cube.termMap φ (p + 1))

theorem **cechFunctor_eq** : cechFunctor.{v} ι R = cubeFunctor ι R ⋙ CochainComplex.truncate

theorem **cechAug_f_zero** (A : Cube.{v} ι R) : (cechAug A).f 0 = ModuleCat.ofHom (Cube.aug A)

theorem **cechAug_naturality** {A B : Cube.{v} ι R} (φ : A ⟶ B) : (CochainComplex.single₀ (ModuleCat.{v} R)).map (φ.app ∅) ≫ cechAug B = cechAug A ≫ cechComplexMap φ

theorem **cubeComplex_exactAt_succ_iff** (A : Cube.{v} ι R) (p : ℕ) : (cubeComplex A).ExactAt (p + 1) ↔ Function.Exact (Cube.d A p) (Cube.d A (p + 1))

theorem **cochainComplex_exactAt_zero_iff** (K : CochainComplex (ModuleCat.{v} R) ℕ) : K.ExactAt 0 ↔ Function.Injective (K.d 0 1).hom

theorem **cubeComplex_exactAt_zero_iff'** (A : Cube.{v} ι R) : (cubeComplex A).ExactAt 0 ↔ Function.Injective (Cube.d A 0)

theorem **cubeComplex_exactAt_zero_iff** (A : Cube.{v} ι R) : (cubeComplex A).ExactAt 0 ↔ Function.Injective (Cube.aug A)

theorem **cubeComplex_exactAt_one_iff** (A : Cube.{v} ι R) : (cubeComplex A).ExactAt 1 ↔ Function.Exact (Cube.aug A) (Cube.d A 1)

theorem **cechComplex_exactAt_succ_iff** (A : Cube.{v} ι R) (p : ℕ) : (cechComplex A).ExactAt (p + 1) ↔ Function.Exact (Cube.d A (p + 1)) (Cube.d A (p + 2))

theorem **cechComplex_exactAt_succ_iff_cube** (A : Cube.{v} ι R) (p : ℕ) : (cechComplex A).ExactAt (p + 1) ↔ (cubeComplex A).ExactAt (p + 2)

theorem **cochainCyclesIsoKer_hom_subtype** (K : CochainComplex (ModuleCat.{v} R) ℕ) (p : ℕ) (f : K.X p →ₗ[R] K.X (p + 1)) (hf : K.d p (p + 1) = ModuleCat.ofHom f) : (cochainCyclesIsoKer K p f hf).hom ≫ ModuleCat.ofHom (LinearMap.ker f).subtype = K.iCycles p

theorem **cubeCyclesIso_hom_subtype** (A : Cube.{v} ι R) (p : ℕ) : (cubeCyclesIso A p).hom ≫ ModuleCat.ofHom (LinearMap.ker (Cube.d A p)).subtype = (cubeComplex A).iCycles p

theorem **cechCyclesIso_hom_subtype** (A : Cube.{v} ι R) (p : ℕ) : (cechCyclesIso A p).hom ≫ ModuleCat.ofHom (LinearMap.ker (Cube.d A (p + 1))).subtype = (cechComplex A).iCycles p

theorem **homologyπ_cechHomologyZeroIso_hom** (A : Cube.{v} ι R) : (cechComplex A).homologyπ 0 ≫ (cechHomologyZeroIso A).hom = (cechCyclesIso A 0).hom

theorem **isZero_cubeComplex_X** [Fintype ι] (A : Cube.{v} ι R) {p : ℕ} (h : Fintype.card ι < p) : IsZero ((cubeComplex A).X p)

theorem **isZero_cechComplex_X** [Fintype ι] (A : Cube.{v} ι R) {p : ℕ} (h : Fintype.card ι ≤ p) : IsZero ((cechComplex A).X p)

theorem **isZero_cubeComplex_homology** [Fintype ι] (A : Cube.{v} ι R) {p : ℕ} (h : Fintype.card ι < p) : IsZero ((cubeComplex A).homology p)

theorem **isZero_cechComplex_homology** [Fintype ι] (A : Cube.{v} ι R) {p : ℕ} (h : Fintype.card ι ≤ p) : IsZero ((cechComplex A).homology p)

theorem **Cube.term_shortExact** (S : ShortComplex (Cube.{v} ι R)) (p : ℕ) (hS : ∀ s : Finset ι, s.card = p → (S.map ((evaluation _ _).obj s)).ShortExact) : ((S.map (cubeFunctor ι R)).map (HomologicalComplex.eval _ _ p)).ShortExact

theorem **cubeFunctor_shortExact** (S : ShortComplex (Cube.{v} ι R)) (hS : ∀ s : Finset ι, (S.map ((evaluation _ _).obj s)).ShortExact) : (S.map (cubeFunctor ι R)).ShortExact

theorem **cechFunctor_shortExact** (S : ShortComplex (Cube.{v} ι R)) (hS : ∀ s : Finset ι, s.Nonempty → (S.map ((evaluation _ _).obj s)).ShortExact) : (S.map (cechFunctor ι R)).ShortExact

theorem **isIso_cechComplexMap** {A B : Cube.{v} ι R} (φ : A ⟶ B) (h : ∀ s : Finset ι, s.Nonempty → Function.Bijective (φ.app s).hom) : IsIso (cechComplexMap φ)

theorem **cubeComplex_exactAt_of_bijective** (A : Cube.{v} ι R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → Function.Bijective (Cube.res A (Finset.subset_insert i s))) (p : ℕ) : (cubeComplex A).ExactAt p

theorem **cubeComplex_const_exactAt** [Nonempty ι] (M : ModuleCat.{v} R) (p : ℕ) : (cubeComplex (Cube.const ι M)).ExactAt p

theorem **cubeComplex_const_acyclic** [Nonempty ι] (M : ModuleCat.{v} R) : (cubeComplex (Cube.const ι M)).Acyclic

theorem **Cube.termComparison_apply** (A : Cube.{v} ι R) (p : ℕ) (x : F.obj (ModuleCat.of R (Cube.Term A p))) (s : {s : Finset ι // s.card = p}) : (Cube.termComparison F A p).hom x s = (F.map (Cube.termProj A s)).hom x

theorem **Cube.termComparison_naturality** {A B : Cube.{v} ι R} (φ : A ⟶ B) (p : ℕ) : F.map (ModuleCat.ofHom (Cube.termMap φ p)) ≫ Cube.termComparison F B p = Cube.termComparison F A p ≫ ModuleCat.ofHom (Cube.termMap (Functor.whiskerRight φ F) p)

theorem **Cube.map_aug_termComparison** (A : Cube.{v} ι R) : F.map (ModuleCat.ofHom (Cube.aug A)) ≫ Cube.termComparison F A 1 = ModuleCat.ofHom (Cube.aug (A ⋙ F))

theorem **Cube.d_termProj** (A : Cube.{v} ι R) (p : ℕ) (t : {s : Finset ι // s.card = p + 1}) : ModuleCat.ofHom (Cube.d A p) ≫ Cube.termProj A t = ∑ i : t.1, cubeSign t.1 i.1 • (Cube.termProj A (Cube.eraseIdx t i) ≫ A.map (homOfLE (Finset.erase_subset i.1 t.1)))

theorem **Cube.map_d_termComparison** (A : Cube.{v} ι R) (p : ℕ) : F.map (ModuleCat.ofHom (Cube.d A p)) ≫ Cube.termComparison F A (p + 1) = Cube.termComparison F A p ≫ ModuleCat.ofHom (Cube.d (A ⋙ F) p)

theorem **cubeComparison_f** (A : Cube.{v} ι R) (p : ℕ) : (cubeComparison F A).f p = Cube.termComparison F A p

theorem **cechComparison_f** (A : Cube.{v} ι R) (p : ℕ) : (cechComparison F A).f p = Cube.termComparison F A (p + 1)

theorem **Cube.termComparison_restrictScalars_apply** (A : Cube.{v} ι S) (p : ℕ) (x : Cube.Term A p) (s : {s : Finset ι // s.card = p}) : (Cube.termComparison (ModuleCat.restrictScalars f) A p).hom x s = x s

theorem **Cube.isIso_termComparison_restrictScalars** (A : Cube.{v} ι S) (p : ℕ) : IsIso (Cube.termComparison (ModuleCat.restrictScalars f) A p)

theorem **Cube.d_restrictScalars_apply** (A : Cube.{v} ι S) (p : ℕ) (x : Cube.Term A p) : Cube.d (A ⋙ ModuleCat.restrictScalars f) p x = Cube.d A p x

theorem **Cube.termComparison_extendScalars_tmul** {S : Type u'} [CommRing S] (f : R →+* S) (A : Cube.{v} ι R) (p : ℕ) (c : S) (x : Cube.Term A p) (s : {s : Finset ι // s.card = p}) : (Cube.termComparison (ModuleCat.extendScalars f) A p).hom (c ⊗ₜ[R,f] x) s = c ⊗ₜ[R,f] x s

theorem **Cube.baseChange_obj** (A : Cube.{v} ι R) (s : Finset ι) : (Cube.baseChange S A).obj s = ModuleCat.of S (S ⊗[R] A.obj s)

theorem **Cube.res_baseChange** (A : Cube.{v} ι R) {s t : Finset ι} (h : s ⊆ t) : Cube.res (Cube.baseChange S A) h = (Cube.res A h).baseChange S

theorem **Cube.termBaseChange_tmul** (A : Cube.{v} ι R) (p : ℕ) (c : S) (x : Cube.Term A p) (s : {s : Finset ι // s.card = p}) : Cube.termBaseChange S A p (c ⊗ₜ[R] x) s = c ⊗ₜ[R] x s

theorem **Cube.termBaseChange_bijective** [Finite ι] (A : Cube.{v} ι R) (p : ℕ) : Function.Bijective (Cube.termBaseChange S A p)

theorem **Cube.termBaseChangeEquiv_apply** [Finite ι] (A : Cube.{v} ι R) (p : ℕ) (x : S ⊗[R] Cube.Term A p) : Cube.termBaseChangeEquiv S A p x = Cube.termBaseChange S A p x

theorem **Cube.d_termBaseChange** [LinearOrder ι] (A : Cube.{v} ι R) (p : ℕ) (x : S ⊗[R] Cube.Term A p) : Cube.d (Cube.baseChange S A) p (Cube.termBaseChange S A p x) = Cube.termBaseChange S A (p + 1) ((Cube.d A p).baseChange S x)

theorem **Cube.aug_baseChange** (A : Cube.{v} ι R) (x : S ⊗[R] A.obj ∅) : Cube.aug (Cube.baseChange S A) x = Cube.termBaseChange S A 1 ((Cube.aug A).baseChange S x)

theorem **Cube.finite_term** {R : Type u} [Ring R] [Finite ι] (A : Cube.{v} ι R) (p : ℕ) (h : ∀ s : Finset ι, s.card = p → Module.Finite R (A.obj s)) : Module.Finite R (Cube.Term A p)

theorem **Cube.flat_term** [Finite ι] (A : Cube.{v} ι R) (p : ℕ) (h : ∀ s : Finset ι, s.card = p → Module.Flat R (A.obj s)) : Module.Flat R (Cube.Term A p)

theorem **homology_directSum_bijective** {K : κ → CochainComplex (ModuleCat.{v} R) ℕ} {L : CochainComplex (ModuleCat.{v} R) ℕ} (i : ∀ k, K k ⟶ L) (π : ∀ k, L ⟶ K k) (h₁ : ∀ k, i k ≫ π k = 𝟙 _) (h₂ : ∀ k j, k ≠ j → i k ≫ π j = 0) (p : ℕ) (h₃ : ∀ z : L.X p, ∃ T : Finset κ, ∑ k ∈ T, ((π k ≫ i k).f p).hom z = z) : Function.Bijective (DirectSum.toModule R κ (L.homology p) fun k => (HomologicalComplex.homologyMap (i k) p).hom)

theorem **Cube.directSum_obj** (A : κ → Cube.{v} ι R) (s : Finset ι) : (Cube.directSum A).obj s = ModuleCat.of R (⊕ k, (A k).obj s)

theorem **Cube.res_directSum** (A : κ → Cube.{v} ι R) {s t : Finset ι} (h : s ⊆ t) : Cube.res (Cube.directSum A) h = DirectSum.lmap fun k => Cube.res (A k) h

theorem **Cube.directSumIncl_proj** (A : κ → Cube.{v} ι R) (k : κ) : Cube.directSumIncl A k ≫ Cube.directSumProj A k = 𝟙 _

theorem **Cube.directSumIncl_proj_of_ne** (A : κ → Cube.{v} ι R) {k j : κ} (h : k ≠ j) : Cube.directSumIncl A k ≫ Cube.directSumProj A j = 0

theorem **Cube.exists_sum_directSum** [Finite ι] (A : κ → Cube.{v} ι R) (p : ℕ) (z : Cube.Term (Cube.directSum A) p) : ∃ T : Finset κ, ∑ k ∈ T, Cube.termMap (Cube.directSumProj A k ≫ Cube.directSumIncl A k) p z = z

theorem **Cube.termDirectSum_lof** (A : κ → Cube.{v} ι R) (p : ℕ) (k : κ) (x : Cube.Term (A k) p) : Cube.termDirectSum A p (DirectSum.lof R κ (fun k => Cube.Term (A k) p) k x) = Cube.termMap (Cube.directSumIncl A k) p x

theorem **Cube.termMap_proj_termDirectSum** (A : κ → Cube.{v} ι R) (p : ℕ) (j : κ) (x : ⊕ k, Cube.Term (A k) p) : Cube.termMap (Cube.directSumProj A j) p (Cube.termDirectSum A p x) = x j

theorem **Cube.termDirectSum_bijective** [Finite ι] (A : κ → Cube.{v} ι R) (p : ℕ) : Function.Bijective (Cube.termDirectSum A p)

theorem **Cube.d_termDirectSum** (A : κ → Cube.{v} ι R) (p : ℕ) (x : ⨁ k, Cube.Term (A k) p) : Cube.d (Cube.directSum A) p (Cube.termDirectSum A p x) = Cube.termDirectSum A (p + 1) (DirectSum.lmap (fun k => Cube.d (A k) p) x)

theorem **cubeHomologyDirectSumEquiv_lof** (A : κ → Cube.{v} ι R) (p : ℕ) (k : κ) (x : (cubeComplex (A k)).homology p) : cubeHomologyDirectSumEquiv A p (DirectSum.lof R κ (fun k => (cubeComplex (A k)).homology p) k x) = (HomologicalComplex.homologyMap (cubeComplexMap (Cube.directSumIncl A k)) p).hom x

theorem **cechHomologyDirectSumEquiv_lof** (A : κ → Cube.{v} ι R) (p : ℕ) (k : κ) (x : (cechComplex (A k)).homology p) : cechHomologyDirectSumEquiv A p (DirectSum.lof R κ (fun k => (cechComplex (A k)).homology p) k x) = (HomologicalComplex.homologyMap (cechComplexMap (Cube.directSumIncl A k)) p).hom x

theorem **cubeComplex_directSum_exactAt_iff** (A : κ → Cube.{v} ι R) (p : ℕ) : (cubeComplex (Cube.directSum A)).ExactAt p ↔ ∀ k, (cubeComplex (A k)).ExactAt p

theorem **cechComplex_directSum_exactAt_iff** (A : κ → Cube.{v} ι R) (p : ℕ) : (cechComplex (Cube.directSum A)).ExactAt p ↔ ∀ k, (cechComplex (A k)).ExactAt p

theorem **Cube.termOneFinTwo_apply** (A : Cube.{v} (Fin 2) R) (x : Cube.Term A 1) : Cube.termOneFinTwo A x = (x ⟨{0}, rfl⟩, x ⟨{1}, rfl⟩)

theorem **Cube.termTwoFinTwo_apply** (A : Cube.{v} (Fin 2) R) (x : Cube.Term A 2) : Cube.termTwoFinTwo A x = x ⟨{0, 1}, rfl⟩

theorem **Cube.d_zero_finTwo** (A : Cube.{v} (Fin 2) R) (x : Cube.Term A 0) : Cube.termOneFinTwo A (Cube.d A 0 x) = (Cube.res A (Finset.empty_subset _) (Cube.termZeroEquiv A x), Cube.res A (Finset.empty_subset _) (Cube.termZeroEquiv A x))

theorem **Cube.d_one_finTwo** (A : Cube.{v} (Fin 2) R) (x : Cube.Term A 1) : Cube.termTwoFinTwo A (Cube.d A 1 x) = Cube.res A (by decide) (Cube.termOneFinTwo A x).2 - Cube.res A (by decide) (Cube.termOneFinTwo A x).1

theorem **isZero_cubeComplex_X_finTwo** (A : Cube.{v} (Fin 2) R) {p : ℕ} (h : 3 ≤ p) : IsZero ((cubeComplex A).X p)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.IsLocalizedModuleGermLinearPointOfPrime

theorem **stalkFunctor_map_germ** (φ : F ⟶ G) (x : X) (hx : x ∈ U) (s : Γ(F, U)) : (TopCat.Presheaf.stalkFunctor Ab.{u} x).map φ.mapPresheaf (F.presheaf.germ U x hx s) = G.presheaf.germ U x hx (φ.app U s)

theorem **stalkLinear_germ** (φ : F ⟶ G) (x : X) (hx : x ∈ U) (s : Γ(F, U)) : stalkLinear φ x (F.presheaf.germ U x hx s) = G.presheaf.germ U x hx (φ.app U s)

theorem **stalkLinear_id** (F : X.Modules) (x : X) : stalkLinear (𝟙 F) x = LinearMap.id

theorem **stalkLinear_comp** (φ : F ⟶ G) (ψ : G ⟶ H) (x : X) : stalkLinear (φ ≫ ψ) x = stalkLinear ψ x ∘ₗ stalkLinear φ x

theorem **stalkLinear_add** (φ ψ : F ⟶ G) (x : X) : stalkLinear (φ + ψ) x = stalkLinear φ x + stalkLinear ψ x

theorem **stalkLinear_zero** (x : X) : stalkLinear (0 : F ⟶ G) x = 0

theorem **stalkLinearEquiv_apply** (e : F ≅ G) (x : X) (t : (F.presheaf.stalk x :)) : stalkLinearEquiv e x t = stalkLinear e.hom x t

theorem **exact_iff_stalkLinear** (S : ShortComplex X.Modules) : S.Exact ↔ ∀ x : X, Function.Exact (stalkLinear S.f x) (stalkLinear S.g x)

theorem **mono_iff_stalkLinear_injective** (φ : F ⟶ G) : Mono φ ↔ ∀ x : X, Function.Injective (stalkLinear φ x)

theorem **epi_iff_stalkLinear_surjective** (φ : F ⟶ G) : Epi φ ↔ ∀ x : X, Function.Surjective (stalkLinear φ x)

theorem **stalkLinear_injective** (φ : F ⟶ G) [Mono φ] (x : X) : Function.Injective (stalkLinear φ x)

theorem **stalkLinear_surjective** (φ : F ⟶ G) [Epi φ] (x : X) : Function.Surjective (stalkLinear φ x)

theorem **section_eq_zero_of_germ** (s : Γ(F, U)) (h : ∀ (x : X) (hx : x ∈ U), F.presheaf.germ U x hx s = 0) : s = 0

theorem **primeIdealOf_pointOfPrime** (hU : IsAffineOpen U) (P : PrimeSpectrum Γ(X, U)) : hU.primeIdealOf (pointOfPrime hU P) = P

theorem **pointOfPrime_primeIdealOf** (hU : IsAffineOpen U) (x : U) : pointOfPrime hU (hU.primeIdealOf x) = x

theorem **isLocalizedModule_germLinear_pointOfPrime** (F : X.Modules) [F.IsQuasicoherent] (hU : IsAffineOpen U) (P : Ideal Γ(X, U)) [hP : P.IsPrime] : IsLocalizedModule P.primeCompl (germLinear F (pointOfPrime hU ⟨P, hP⟩))

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.IsQuasiRegularSeqSelfIffLocalizationMaximal

theorem **MonoDeg.sum_eq_sum_finsuppAntidiag** {ι N : Type*} [Fintype ι] [DecidableEq ι] [AddCommMonoid N] (a : ℕ) (g : (ι →₀ ℕ) → N) : ∑ α : MonoDeg ι a, g α.1 = ∑ α ∈ Finset.finsuppAntidiag Finset.univ a, g α

theorem **MonoDeg.card_eq_card_finsuppAntidiag** (ι : Type*) [Fintype ι] [DecidableEq ι] (a : ℕ) : Fintype.card (MonoDeg ι a) = (Finset.finsuppAntidiag (Finset.univ : Finset ι) a).card

theorem **monoVal_eq_prod** [CommMonoid R] [Fintype ι] (f : ι → R) (α : ι →₀ ℕ) : monoVal f α = ∏ i, f i ^ α i

@[simp] theorem **monoVal_zero** [CommMonoid R] (f : ι → R) : monoVal f 0 = 1

@[simp] theorem **monoVal_single** [CommMonoid R] (f : ι → R) (i : ι) (k : ℕ) : monoVal f (Finsupp.single i k) = f i ^ k

theorem **monoVal_add** [CommMonoid R] (f : ι → R) (α β : ι →₀ ℕ) : monoVal f (α + β) = monoVal f α * monoVal f β

theorem **monoVal_mul** [CommMonoid R] (f g : ι → R) (α : ι →₀ ℕ) : monoVal (fun i => f i * g i) α = monoVal f α * monoVal g α

theorem **map_monoVal** [CommMonoid R] {S F : Type*} [CommMonoid S] [FunLike F R S] [MonoidHomClass F R S] (φ : F) (f : ι → R) (α : ι →₀ ℕ) : φ (monoVal f α) = monoVal (fun i => φ (f i)) α

theorem **isUnit_monoVal** [CommMonoid R] {u : ι → R} (hu : ∀ i, IsUnit (u i)) (α : ι →₀ ℕ) : IsUnit (monoVal u α)

theorem **monoVal_eq_eval** [CommSemiring R] (f : ι → R) (α : ι →₀ ℕ) : monoVal f α = MvPolynomial.eval f (MvPolynomial.monomial α 1)

theorem **monoVal_X** [CommSemiring R] (α : ι →₀ ℕ) : monoVal (MvPolynomial.X : ι → MvPolynomial ι R) α = MvPolynomial.monomial α 1

theorem **monoVal_congr** [CommMonoid R] {κ : Type*} (e : ι ≃ κ) (f : κ → R) (a : ℕ) (α : MonoDeg ι a) : monoVal f (MonoDeg.congr e a α).1 = monoVal (f ∘ e) α.1

theorem **monoVal_mem_pow** (f : ι → R) {a : ℕ} (α : MonoDeg ι a) : monoVal f α.1 ∈ Ideal.span (Set.range f) ^ a

theorem **span_range_pow_eq_span_monoVal** (f : ι → R) (a : ℕ) : Ideal.span (Set.range f) ^ a = Ideal.span (Set.range fun α : MonoDeg ι a => monoVal f α.1)

theorem **mem_pow_smul_top_iff** (f : ι → R) (a : ℕ) (x : M) : x ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M) ↔ ∃ c : MonoDeg ι a → M, ∑ α, monoVal f α.1 • c α = x

theorem **mk_surjective** : Function.Surjective (mk I M a)

theorem **mk_eq_zero** (x : ↥(I ^ a • ⊤ : Submodule R M)) : mk I M a x = 0 ↔ (x : M) ∈ (I ^ (a + 1) • ⊤ : Submodule R M)

theorem **mk_eq_mk** (x y : ↥(I ^ a • ⊤ : Submodule R M)) : mk I M a x = mk I M a y ↔ (x : M) - y ∈ (I ^ (a + 1) • ⊤ : Submodule R M)

theorem **pow_succ_smul_le** : (I ^ (a + 1) • ⊤ : Submodule R M) ≤ I ^ a • ⊤

theorem **isTorsionBySet** : Module.IsTorsionBySet R (powGr I M a) I

@[simp] theorem **mk_smul** (r : R) (x : powGr I M a) : Ideal.Quotient.mk I r • x = r • x

theorem **map_mk** {N : Type*} [AddCommGroup N] [Module R N] (g : M →ₗ[R] N) (x : ↥(I ^ a • ⊤ : Submodule R M)) : map I a g (mk I M a x) = mk I N a ⟨g x, Submodule.smul_top_le_comap_smul_top (I ^ a) g x.2⟩

@[simp] theorem **map_id** : map I a (LinearMap.id : M →ₗ[R] M) = LinearMap.id

theorem **map_comp** {N P : Type*} [AddCommGroup N] [Module R N] [AddCommGroup P] [Module R P] (g : M →ₗ[R] N) (h : N →ₗ[R] P) : map I a (h ∘ₗ g) = map I a h ∘ₗ map I a g

@[simp] theorem **toQuot_mk** (x : ↥(I ^ a • ⊤ : Submodule R M)) : toQuot I M a (mk I M a x) = Submodule.Quotient.mk (x : M)

theorem **toQuot_injective** : Function.Injective (toQuot I M a)

theorem **exact_toQuot_factor** : Function.Exact (toQuot I M a) (Submodule.factor (pow_succ_smul_le I M a))

theorem **mulElem_mk** (r : R) (hr : r ∈ I ^ a) (x : M) : mulElem M r hr (Submodule.Quotient.mk x) = mk I M a ⟨r • x, Submodule.smul_mem_smul hr trivial⟩

theorem **monoToPowGr_apply** (f : ι → R) (a : ℕ) (c : MonoDeg ι a → M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) : monoToPowGr M f a c = ∑ α, powGr.mulElem M (monoVal f α.1) (monoVal_mem_pow f α) (c α)

theorem **monoToPowGr_mk** (f : ι → R) (a : ℕ) (c : MonoDeg ι a → M) : monoToPowGr M f a (fun α => Submodule.Quotient.mk (c α)) = powGr.mk _ M a ⟨∑ α, monoVal f α.1 • c α, (mem_pow_smul_top_iff f a _).mpr ⟨c, rfl⟩⟩

theorem **monoToPowGr_single** [DecidableEq ι] (f : ι → R) (a : ℕ) (α : MonoDeg ι a) (x : M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) : monoToPowGr M f a (Pi.single α x) = powGr.mulElem M (monoVal f α.1) (monoVal_mem_pow f α) x

theorem **monoToPowGr_surjective** (f : ι → R) (a : ℕ) : Function.Surjective (monoToPowGr M f a)

theorem **monoToPowGr_injective_iff** (f : ι → R) (a : ℕ) : Function.Injective (monoToPowGr M f a) ↔ ∀ c : MonoDeg ι a → M, ∑ α, monoVal f α.1 • c α ∈ (Ideal.span (Set.range f) ^ (a + 1) • ⊤ : Submodule R M) → ∀ α, c α ∈ (Ideal.span (Set.range f) • ⊤ : Submodule R M)

theorem **isQuasiRegularSeq_iff_injective** (f : ι → R) : IsQuasiRegularSeq M f ↔ ∀ a, Function.Injective (monoToPowGr M f a)

theorem **isQuasiRegularSeq_iff_bijective** (f : ι → R) : IsQuasiRegularSeq M f ↔ ∀ a, Function.Bijective (monoToPowGr M f a)

theorem **isQuasiRegularSeq_iff_finsuppAntidiag** {ι : Type*} [Fintype ι] [DecidableEq ι] (f : ι → R) : IsQuasiRegularSeq M f ↔ ∀ (a : ℕ) (c : (ι →₀ ℕ) → M), ∑ α ∈ Finset.finsuppAntidiag Finset.univ a, monoVal f α • c α ∈ (Ideal.span (Set.range f) ^ (a + 1) • ⊤ : Submodule R M) → ∀ α, α.degree = a → c α ∈ (Ideal.span (Set.range f) • ⊤ : Submodule R M)

theorem **isQuasiRegularSeq_self_iff** (f : ι → R) : IsQuasiRegularSeq R f ↔ ∀ (a : ℕ) (p : MvPolynomial ι R), p.IsHomogeneous a → MvPolynomial.eval f p ∈ Ideal.span (Set.range f) ^ (a + 1) → ∀ α, p.coeff α ∈ Ideal.span (Set.range f)

@[simp] theorem **powGrEquiv_apply** {f : ι → R} (h : IsQuasiRegularSeq M f) (a : ℕ) (c : MonoDeg ι a → M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) : h.powGrEquiv a c = monoToPowGr M f a c

theorem **comp_equiv** {κ : Type*} [Finite κ] {f : ι → R} (h : IsQuasiRegularSeq M f) (e : κ ≃ ι) : IsQuasiRegularSeq M (f ∘ e)

theorem **unit_mul** {f : ι → R} (h : IsQuasiRegularSeq M f) {u : ι → R} (hu : ∀ i, IsUnit (u i)) : IsQuasiRegularSeq M fun i => u i * f i

theorem **of_associated** {f g : ι → R} (h : IsQuasiRegularSeq M f) (hfg : ∀ i, Associated (f i) (g i)) : IsQuasiRegularSeq M g

theorem **of_linearEquiv** {N : Type*} [AddCommGroup N] [Module R N] {f : ι → R} (h : IsQuasiRegularSeq M f) (e : M ≃ₗ[R] N) : IsQuasiRegularSeq N f

theorem **isQuasiRegularSeq_comp_equiv_iff** {κ : Type*} [Finite κ] (f : ι → R) (e : κ ≃ ι) : IsQuasiRegularSeq M (f ∘ e) ↔ IsQuasiRegularSeq M f

theorem **isQuasiRegularSeq_unit_mul_iff** (f : ι → R) {u : ι → R} (hu : ∀ i, IsUnit (u i)) : (IsQuasiRegularSeq M fun i => u i * f i) ↔ IsQuasiRegularSeq M f

theorem **isQuasiRegularSeq_congr** {N : Type*} [AddCommGroup N] [Module R N] (f : ι → R) (e : M ≃ₗ[R] N) : IsQuasiRegularSeq M f ↔ IsQuasiRegularSeq N f

theorem **isQuasiRegularSeq_of_smul_top_eq_top** {f : ι → R} (h : (Ideal.span (Set.range f) • ⊤ : Submodule R M) = ⊤) : IsQuasiRegularSeq M f

theorem **isQuasiRegularSeq_algebraMap_iff** (S : Type*) [CommRing S] [Algebra R S] [Module S M] [IsScalarTower R S M] (f : ι → R) : IsQuasiRegularSeq M (fun i => algebraMap R S (f i)) ↔ IsQuasiRegularSeq M f

theorem **IsQuasiRegularSeq.powGrBasis_apply** {f : ι → R} (h : IsQuasiRegularSeq R f) (a : ℕ) (α : MonoDeg ι a) : h.powGrBasis a α = powGr.mk _ R a ⟨monoVal f α.1, by simpa using monoVal_mem_pow f α⟩

theorem **IsQuasiRegularSeq.free_powGr** {M : Type*} [AddCommGroup M] [Module R M] {f : ι → R} (h : IsQuasiRegularSeq M f) (a : ℕ) [Module.Free (R ⧸ Ideal.span (Set.range f)) (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M))] : Module.Free (R ⧸ Ideal.span (Set.range f)) (powGr (Ideal.span (Set.range f)) M a)

theorem **finite_powGr** {M : Type*} [AddCommGroup M] [Module R M] [Module.Finite R M] (f : ι → R) (a : ℕ) : Module.Finite (R ⧸ Ideal.span (Set.range f)) (powGr (Ideal.span (Set.range f)) M a)

theorem **monoToQuotPow_mk** (f : ι → R) (a : ℕ) (c : MonoDeg ι a → R) : monoToQuotPow f a (fun α => Ideal.Quotient.mk _ (c α)) = Ideal.Quotient.mk _ (∑ α, monoVal f α.1 * c α)

theorem **monoToPowGr_injective_iff_lTensor** (f : ι → R) (a : ℕ) : Function.Injective (monoToPowGr M f a) ↔ Function.Injective ((monoToQuotPow f a).lTensor M)

theorem **isQuasiRegularSeq_iff_lTensor** (f : ι → R) : IsQuasiRegularSeq M f ↔ ∀ a, Function.Injective ((monoToQuotPow f a).lTensor M)

private theorem **injective_lTensor_tensor_iff** (N : Type*) [AddCommGroup N] [Module R N] {P Q : Type*} [AddCommGroup P] [Module R P] [AddCommGroup Q] [Module R Q] (g : P →ₗ[R] Q) : Function.Injective (g.lTensor (N ⊗[R] M)) ↔ Function.Injective ((g.lTensor M).lTensor N)

theorem **IsQuasiRegularSeq.tensor** {f : ι → R} (h : IsQuasiRegularSeq M f) (N : Type*) [AddCommGroup N] [Module R N] [Module.Flat R N] : IsQuasiRegularSeq (N ⊗[R] M) f

theorem **isQuasiRegularSeq_tensor_iff** (f : ι → R) (N : Type*) [AddCommGroup N] [Module R N] [Module.FaithfullyFlat R N] : IsQuasiRegularSeq (N ⊗[R] M) f ↔ IsQuasiRegularSeq M f

theorem **IsQuasiRegularSeq.of_flat_module** {f : ι → R} (h : IsQuasiRegularSeq R f) (N : Type*) [AddCommGroup N] [Module R N] [Module.Flat R N] : IsQuasiRegularSeq N f

theorem **IsQuasiRegularSeq.baseChange** [Module.Flat R S] {f : ι → R} (h : IsQuasiRegularSeq M f) : IsQuasiRegularSeq (S ⊗[R] M) fun i => algebraMap R S (f i)

theorem **IsQuasiRegularSeq.of_flat** [Module.Flat R S] {f : ι → R} (h : IsQuasiRegularSeq R f) : IsQuasiRegularSeq S fun i => algebraMap R S (f i)

theorem **isQuasiRegularSeq_baseChange_iff** [Module.FaithfullyFlat R S] (f : ι → R) : (IsQuasiRegularSeq (S ⊗[R] M) fun i => algebraMap R S (f i)) ↔ IsQuasiRegularSeq M f

theorem **IsQuasiRegularSeq.of_flat_of_isBaseChange** [Module S N] [IsScalarTower R S N] [Module.Flat R S] {g : M →ₗ[R] N} (hg : IsBaseChange S g) {f : ι → R} (h : IsQuasiRegularSeq M f) : IsQuasiRegularSeq N fun i => algebraMap R S (f i)

theorem **IsQuasiRegularSeq.of_isLocalizedModule** [Module S N] [IsScalarTower R S N] (T : Submonoid R) [IsLocalization T S] (g : M →ₗ[R] N) [IsLocalizedModule T g] {f : ι → R} (h : IsQuasiRegularSeq M f) : IsQuasiRegularSeq N fun i => algebraMap R S (f i)

theorem **IsQuasiRegularSeq.of_isLocalizedModule'** (T : Submonoid R) (g : M →ₗ[R] N) [IsLocalizedModule T g] {f : ι → R} (h : IsQuasiRegularSeq M f) : IsQuasiRegularSeq N f

theorem **IsQuasiRegularSeq.of_isLocalization** (T : Submonoid R) [IsLocalization T S] {f : ι → R} (h : IsQuasiRegularSeq R f) : IsQuasiRegularSeq S fun i => algebraMap R S (f i)

theorem **exists_smul_mem_smul_top_of_isLocalizedModule** (T : Submonoid R) (g : M →ₗ[R] N) [IsLocalizedModule T g] (J : Ideal R) {x : M} (hx : g x ∈ (J • ⊤ : Submodule R N)) : ∃ s ∈ T, s • x ∈ (J • ⊤ : Submodule R M)

theorem **IsQuasiRegularSeq.of_isLocalizedModule_maximal** {f : ι → R} (Mₚ : ∀ (P : Ideal R) [P.IsMaximal], Type*) [∀ (P : Ideal R) [P.IsMaximal], AddCommGroup (Mₚ P)] [∀ (P : Ideal R) [P.IsMaximal], Module R (Mₚ P)] (g : ∀ (P : Ideal R) [P.IsMaximal], M →ₗ[R] Mₚ P) [∀ (P : Ideal R) [P.IsMaximal], IsLocalizedModule P.primeCompl (g P)] (h : ∀ (P : Ideal R) [P.IsMaximal], Ideal.span (Set.range f) ≤ P → IsQuasiRegularSeq (Mₚ P) f) : IsQuasiRegularSeq M f

theorem **isQuasiRegularSeq_iff_localization_maximal** (f : ι → R) : IsQuasiRegularSeq M f ↔ ∀ (P : Ideal R) [P.IsMaximal], Ideal.span (Set.range f) ≤ P → IsQuasiRegularSeq (LocalizedModule P.primeCompl M) fun i => algebraMap R (Localization.AtPrime P) (f i)

theorem **isQuasiRegularSeq_self_iff_localization_maximal** (f : ι → R) : IsQuasiRegularSeq R f ↔ ∀ (P : Ideal R) [P.IsMaximal], Ideal.span (Set.range f) ≤ P → IsQuasiRegularSeq (Localization.AtPrime P) fun i => algebraMap R (Localization.AtPrime P) (f i)

theorem **isQuasiRegularSeq_X** (ι : Type*) [Finite ι] : IsQuasiRegularSeq (MvPolynomial ι R) (MvPolynomial.X : ι → MvPolynomial ι R)

theorem **isQuasiRegularSeq_singleton_iff** {M : Type*} [AddCommGroup M] [Module R M] (x : R) : IsQuasiRegularSeq M ![x] ↔ ∀ (a : ℕ) (m m' : M), x ^ a • m = x ^ (a + 1) • m' → ∃ m'', m = x • m''

theorem **isQuasiRegularSeq_singleton** {M : Type*} [AddCommGroup M] [Module R M] {x : R} (hx : IsSMulRegular M x) : IsQuasiRegularSeq M ![x]

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.IsQuasicoherentOfLocallyIso

@[simp] theorem **resM_resM** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (s : Γ(F, U)) : resM F h₂ (resM F h₁ s) = resM F (h₂.trans h₁) s

@[simp] theorem **resM_self** {U : X.Opens} (h : U ≤ U) (s : Γ(F, U)) : resM F h s = s

@[simp] theorem **resM_smul** {U V : X.Opens} (h : V ≤ U) (a : Γ(X, U)) (s : Γ(F, U)) : resM F h (a • s) = resG X h a • resM F h s

theorem **app_resM** (φ : F ⟶ G) {U V : X.Opens} (h : V ≤ U) (s : Γ(F, U)) : φ.app V (resM F h s) = resM G h (φ.app U s)

theorem **sec_ext_of_cover** {ι : Type*} {V : X.Opens} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) {s t : Γ(F, V)} (h : ∀ k, resM F (hW k) s = resM F (hW k) t) : s = t

theorem **exists_sec_of_cover** {ι : Type*} {V : X.Opens} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) (s : ∀ k, Γ(F, W k)) (h : ∀ k l, resM F (inf_le_left : W k ⊓ W l ≤ W k) (s k) = resM F (inf_le_right : W k ⊓ W l ≤ W l) (s l)) : ∃ t : Γ(F, V), ∀ k, resM F (hW k) t = s k

theorem **bijective_app_of_cover** (φ : F ⟶ G) {ι : Type*} {U : ι → X.Opens} (hU : ⨆ k, U k = ⊤) (V : X.Opens) (h : ∀ (k : ι) (W : X.Opens), W ≤ V → W ≤ U k → Function.Bijective (φ.app W)) : Function.Bijective (φ.app V)

theorem **isIso_of_bijective** (φ : F ⟶ G) (h : ∀ V, Function.Bijective (φ.app V)) : IsIso φ

theorem **isIso_of_cover** (φ : F ⟶ G) {ι : Type*} {U : ι → X.Opens} (hU : ⨆ k, U k = ⊤) (h : ∀ (k : ι) (W : X.Opens), W ≤ U k → Function.Bijective (φ.app W)) : IsIso φ

theorem **bijective_smul_of_isUnit** {V : X.Opens} {a : Γ(X, V)} (ha : IsUnit a) : Function.Bijective fun s : Γ(F, V) => a • s

theorem **isFiniteType_of_locally_iso** {ι : Type} {U : ι → X.Opens} (hU : ⨆ k, U k = ⊤) (e : ∀ (k : ι) (W : X.Opens), W ≤ U k → (F.restrict W.ι ≅ G.restrict W.ι)) [hF : F.IsFiniteType] : G.IsFiniteType

theorem **isQuasicoherent_of_locally_iso** {ι : Type} {U : ι → X.Opens} (hU : ⨆ k, U k = ⊤) (e : ∀ k : ι, F.restrict (U k).ι ≅ G.restrict (U k).ι) [F.IsQuasicoherent] : G.IsQuasicoherent

theorem **germ_resM** {V W : X.Opens} (h : W ≤ V) {x : X} (hx : x ∈ W) (s : Γ(F, V)) : F.presheaf.germ W x hx (resM F h s) = F.presheaf.germ V x (h hx) s

theorem **stalkHomOfLinear_germ** {V : X.Opens} (hV : V ≤ U) (hxV : x ∈ V) (s : Γ(F, V)) : stalkHomOfLinear φ hφ hx (F.presheaf.germ V x hxV s) = G.presheaf.germ V x hxV (φ V hV s)

theorem **exists_germ_eq_of_mem** (t : (F.presheaf.stalk x :)) : ∃ (V : X.Opens) (_ : V ≤ U) (hxV : x ∈ V) (s : Γ(F, V)), F.presheaf.germ V x hxV s = t

theorem **symm_resM** {V W : X.Opens} (hV : V ≤ U) (h : W ≤ V) (s : Γ(G, V)) : (φ W (h.trans hV)).symm (resM G h s) = resM F h ((φ V hV).symm s)

theorem **stalkEquivOfLinear_germ** {V : X.Opens} (hV : V ≤ U) (hxV : x ∈ V) (s : Γ(F, V)) : stalkEquivOfLinear φ hφ hx (F.presheaf.germ V x hxV s) = G.presheaf.germ V x hxV (φ V hV s)

theorem **mateHom_naturality** {A B : C} (f : A ⟶ B) : P.map (S.map f) ≫ mateHom adj S T e B = mateHom adj S T e A ≫ T.map (P.map f)

theorem **isIso_mateHom** [S.IsEquivalence] [T.IsEquivalence] (A : C) : IsIso (mateHom adj S T e A)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.LengthCommensurableMapIffOfInjective

theorem **relLength_def** (A B : Submodule R V) : relLength A B = Module.length R (A ⧸ B.submoduleOf A)

theorem **relLength_eq_length_quotient_comap** (A B : Submodule R V) : relLength A B = Module.length R (A ⧸ B.comap A.subtype)

theorem **relLength_eq_length_range** {P : Type*} [AddCommGroup P] [Module R P] {A B : Submodule R V} (g : A →ₗ[R] P) (hg : LinearMap.ker g = B.submoduleOf A) : relLength A B = Module.length R (LinearMap.range g)

theorem **relLength_eq_length_of_surjective** {P : Type*} [AddCommGroup P] [Module R P] {A B : Submodule R V} (g : A →ₗ[R] P) (hg : LinearMap.ker g = B.submoduleOf A) (hs : Function.Surjective g) : relLength A B = Module.length R P

theorem **relLength_eq_length_map_mkQ** (A B : Submodule R V) : relLength A B = Module.length R (A.map B.mkQ)

theorem **relLength_ne_top_iff** {A B : Submodule R V} : relLength A B ≠ ⊤ ↔ IsFiniteLength R (A ⧸ B.submoduleOf A)

theorem **relLength_congr_right** {A B B' : Submodule R V} (h : A ⊓ B = A ⊓ B') : relLength A B = relLength A B'

theorem **relLength_congr_left** {A A' B : Submodule R V} (h : A ⊔ B = A' ⊔ B) : relLength A B = relLength A' B

@[simp] theorem **relLength_inf_right** (A B : Submodule R V) : relLength A (A ⊓ B) = relLength A B

@[simp] theorem **relLength_inf_right'** (A B : Submodule R V) : relLength A (B ⊓ A) = relLength A B

@[simp] theorem **relLength_sup_left** (A B : Submodule R V) : relLength (A ⊔ B) B = relLength A B

@[simp] theorem **relLength_sup_left'** (A B : Submodule R V) : relLength (B ⊔ A) B = relLength A B

theorem **relLength_inf_eq_relLength_sup** (A B : Submodule R V) : relLength A (A ⊓ B) = relLength (A ⊔ B) B

@[simp] theorem **relLength_eq_zero_iff** {A B : Submodule R V} : relLength A B = 0 ↔ A ≤ B

theorem **relLength_eq_zero_of_le** {A B : Submodule R V} (h : A ≤ B) : relLength A B = 0

theorem **relLength_pos_iff** {A B : Submodule R V} : 0 < relLength A B ↔ ¬ A ≤ B

theorem **relLength_self** (A : Submodule R V) : relLength A A = 0

theorem **relLength_bot_left** (B : Submodule R V) : relLength ⊥ B = 0

theorem **relLength_top_right** (A : Submodule R V) : relLength A ⊤ = 0

@[simp] theorem **relLength_bot_right** (A : Submodule R V) : relLength A ⊥ = Module.length R A

@[simp] theorem **relLength_top_left** (B : Submodule R V) : relLength ⊤ B = Module.length R (V ⧸ B)

theorem **relLength_mono_left** {A A' : Submodule R V} (h : A ≤ A') (B : Submodule R V) : relLength A B ≤ relLength A' B

theorem **relLength_anti_right** (A : Submodule R V) {B B' : Submodule R V} (h : B ≤ B') : relLength A B' ≤ relLength A B

@[gcongr] theorem **relLength_le_relLength** {A A' B B' : Submodule R V} (hA : A ≤ A') (hB : B' ≤ B) : relLength A B ≤ relLength A' B'

theorem **monotone_relLength_left** (B : Submodule R V) : Monotone fun A => relLength A B

theorem **antitone_relLength_right** (A : Submodule R V) : Antitone (relLength A)

theorem **relLength_le_length_left** (A B : Submodule R V) : relLength A B ≤ Module.length R A

theorem **relLength_le_length_quotient** (A B : Submodule R V) : relLength A B ≤ Module.length R (V ⧸ B)

theorem **length_eq_of_semilinear_bijective** (f : V →ₛₗ[σ] V') (hf : Function.Bijective f) : Module.length R V = Module.length S V'

theorem **relLength_map_left** (f : V →ₛₗ[σ] V') (A : Submodule R V) (B' : Submodule S V') : relLength (A.map f) B' = relLength A (B'.comap f)

theorem **relLength_map_map** (f : V →ₛₗ[σ] V') (A B : Submodule R V) : relLength (A.map f) (B.map f) = relLength A (B ⊔ LinearMap.ker f)

theorem **relLength_comap_comap** (f : V →ₛₗ[σ] V') (A' B' : Submodule S V') : relLength (A'.comap f) (B'.comap f) = relLength (LinearMap.range f ⊓ A') B'

theorem **relLength_map_of_injective** (f : V →ₛₗ[σ] V') (hf : Function.Injective f) (A B : Submodule R V) : relLength (A.map f) (B.map f) = relLength A B

theorem **relLength_comap_of_le_range** (f : V →ₛₗ[σ] V') {A' : Submodule S V'} (hA : A' ≤ LinearMap.range f) (B' : Submodule S V') : relLength (A'.comap f) (B'.comap f) = relLength A' B'

theorem **relLength_comap_of_surjective** (f : V →ₛₗ[σ] V') (hf : Function.Surjective f) (A' B' : Submodule S V') : relLength (A'.comap f) (B'.comap f) = relLength A' B'

theorem **lengthCommensurable_map_iff_of_injective** (f : V →ₛₗ[σ] V') (hf : Function.Injective f) {A B : Submodule R V} : LengthCommensurable (A.map f) (B.map f) ↔ LengthCommensurable A B

theorem **LengthCommensurable.signedLength_map_of_injective** {A B : Submodule R V} (_h : LengthCommensurable A B) (f : V →ₛₗ[σ] V') (hf : Function.Injective f) : signedLength (A.map f) (B.map f) = signedLength A B

theorem **relLength_map_equiv** (e : V ≃ₛₗ[σ] V') (A B : Submodule R V) : relLength (A.map (e : V →ₛₗ[σ] V')) (B.map (e : V →ₛₗ[σ] V')) = relLength A B

theorem **relLength_comap_equiv** (e : V ≃ₛₗ[σ] V') (A' B' : Submodule S V') : relLength (A'.comap (e : V →ₛₗ[σ] V')) (B'.comap (e : V →ₛₗ[σ] V')) = relLength A' B'

theorem **lengthCommensurable_map_equiv_iff** (e : V ≃ₛₗ[σ] V') {A B : Submodule R V} : LengthCommensurable (A.map (e : V →ₛₗ[σ] V')) (B.map (e : V →ₛₗ[σ] V')) ↔ LengthCommensurable A B

theorem **LengthCommensurable.signedLength_map_equiv** {A B : Submodule R V} (h : LengthCommensurable A B) (e : V ≃ₛₗ[σ] V') : signedLength (A.map (e : V →ₛₗ[σ] V')) (B.map (e : V →ₛₗ[σ] V')) = signedLength A B

theorem **relLength_submoduleOf** {A W : Submodule R V} (h : A ≤ W) (B : Submodule R V) : relLength (A.submoduleOf W) (B.submoduleOf W) = relLength A B

theorem **relLength_map_subtype** (W : Submodule R V) (A B : Submodule R W) : relLength (A.map W.subtype) (B.map W.subtype) = relLength A B

theorem **relLength_map_mkQ** {N B : Submodule R V} (h : N ≤ B) (A : Submodule R V) : relLength (A.map N.mkQ) (B.map N.mkQ) = relLength A B

theorem **relLength_smul_smul** {G : Type*} [Monoid G] [DistribMulAction G V] [SMulCommClass G R V] {a : G} (ha : IsSMulRegular V a) (A B : Submodule R V) : relLength (a • A) (a • B) = relLength A B

theorem **lengthCommensurable_smul_smul_iff** {G : Type*} [Monoid G] [DistribMulAction G V] [SMulCommClass G R V] {a : G} (ha : IsSMulRegular V a) {A B : Submodule R V} : LengthCommensurable (a • A) (a • B) ↔ LengthCommensurable A B

theorem **LengthCommensurable.signedLength_smul_smul** {G : Type*} [Monoid G] [DistribMulAction G V] [SMulCommClass G R V] {a : G} {A B : Submodule R V} (h : LengthCommensurable A B) (ha : IsSMulRegular V a) : signedLength (a • A) (a • B) = signedLength A B

theorem **lengthCommensurable_iff_isFiniteLength** : LengthCommensurable A B ↔ IsFiniteLength R (A ⧸ B.submoduleOf A) ∧ IsFiniteLength R (B ⧸ A.submoduleOf B)

@[refl] theorem **LengthCommensurable.refl** (A : Submodule R V) : LengthCommensurable A A

@[symm] theorem **LengthCommensurable.symm** (h : LengthCommensurable A B) : LengthCommensurable B A

theorem **lengthCommensurable_comm** : LengthCommensurable A B ↔ LengthCommensurable B A

theorem **lengthCommensurable_iff_of_ge** (h : B ≤ A) : LengthCommensurable A B ↔ relLength A B ≠ ⊤

theorem **lengthCommensurable_iff_of_le** (h : A ≤ B) : LengthCommensurable A B ↔ relLength B A ≠ ⊤

theorem **lengthCommensurable_of_length_ne_top** (h : Module.length R V ≠ ⊤) (A B : Submodule R V) : LengthCommensurable A B

theorem **lengthCommensurable_iff_inf** : LengthCommensurable A B ↔ LengthCommensurable A (A ⊓ B) ∧ LengthCommensurable B (A ⊓ B)

theorem **lengthCommensurable_iff_sup** : LengthCommensurable A B ↔ LengthCommensurable (A ⊔ B) B ∧ LengthCommensurable (A ⊔ B) A

theorem **signedLength_eq_of_relLength_eq** {m n : ℕ} (hm : relLength A B = m) (hn : relLength B A = n) : signedLength A B = (m : ℤ) - n

@[simp] theorem **signedLength_self** (A : Submodule R V) : signedLength A A = 0

theorem **exists_signedLength_eq** (h : LengthCommensurable A B) : ∃ m n : ℕ, relLength A B = m ∧ relLength B A = n ∧ signedLength A B = (m : ℤ) - n

theorem **signedLength_swap** (_h : LengthCommensurable A B) : signedLength B A = -signedLength A B

theorem **signedLength_add_swap** (h : LengthCommensurable A B) : signedLength A B + signedLength B A = 0

theorem **signedLength_of_ge** (_h : LengthCommensurable A B) (hBA : B ≤ A) : signedLength A B = (relLength A B).toNat

theorem **signedLength_of_le** (_h : LengthCommensurable A B) (hAB : A ≤ B) : signedLength A B = -((relLength B A).toNat : ℤ)

theorem **coe_toNat_signedLength_of_ge** (h : LengthCommensurable A B) (hBA : B ≤ A) : ((signedLength A B).toNat : ℕ∞) = relLength A B

theorem **signedLength_eq_natCast_iff_of_ge** (h : LengthCommensurable A B) (hBA : B ≤ A) {n : ℕ} : signedLength A B = n ↔ relLength A B = n

theorem **signedLength_le_natCast_iff_of_ge** (h : LengthCommensurable A B) (hBA : B ≤ A) {n : ℕ} : signedLength A B ≤ n ↔ relLength A B ≤ n

theorem **signedLength_nonneg** (h : LengthCommensurable A B) (hBA : B ≤ A) : 0 ≤ signedLength A B

theorem **signedLength_nonpos** (h : LengthCommensurable A B) (hAB : A ≤ B) : signedLength A B ≤ 0

theorem **signedLength_eq_zero_iff_of_ge** (h : LengthCommensurable A B) (hBA : B ≤ A) : signedLength A B = 0 ↔ A = B

theorem **signedLength_le_toNat_relLength** (_h : LengthCommensurable A B) : signedLength A B ≤ (relLength A B).toNat

theorem **signedLength_eq_sub_inf** (h : LengthCommensurable A B) : signedLength A B = signedLength A (A ⊓ B) - signedLength B (A ⊓ B)

theorem **signedLength_eq_sub_sup** (h : LengthCommensurable A B) : signedLength A B = signedLength (A ⊔ B) B - signedLength (A ⊔ B) A

theorem **signedLength_le_signedLength** {A' B' : Submodule R V} (h : LengthCommensurable A B) (h' : LengthCommensurable A' B') (hA : A ≤ A') (hB : B' ≤ B) : signedLength A B ≤ signedLength A' B'

theorem **relLength_top_of_isMaximal** {R : Type*} [CommRing R] (I : Ideal R) [hI : I.IsMaximal] : relLength (⊤ : Ideal R) I = 1

theorem **relLength_int_top_span_prime** {p : ℤ} (hp : Prime p) : relLength (⊤ : Ideal ℤ) (Ideal.span {p}) = 1

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.MemHomogIffIsHomogeneous

@[simp] theorem **map_ι** (f : M →ₗ[R] M') (m : M) : map f (SymmetricAlgebra.ι R M m) = SymmetricAlgebra.ι R M' (f m)

@[simp] theorem **map_id** : map (LinearMap.id : M →ₗ[R] M) = AlgHom.id R _

theorem **map_comp** (g : M' →ₗ[R] M'') (f : M →ₗ[R] M') : map (g ∘ₗ f) = (map g).comp (map f)

theorem **ι_mem_homog** (m : M) : SymmetricAlgebra.ι R M m ∈ homog R M 1

theorem **gradedι_apply** (m : M) : gradedι R M m = DirectSum.of (fun i : ℕ => homog R M i) 1 (SymmetricAlgebra.ι R M m, ι_mem_homog m)

theorem **algHom_mem_pow** {A : Type*} [CommRing A] [Algebra R A] (φ : SymmetricAlgebra R M →ₐ[R] A) (P : Submodule R A) (h : ∀ m, φ (SymmetricAlgebra.ι R M m) ∈ P) {d : ℕ} {x : SymmetricAlgebra R M} (hx : x ∈ homog R M d) : φ x ∈ P ^ d

theorem **map_mem_homog** (f : M →ₗ[R] M') {d : ℕ} {x : SymmetricAlgebra R M} (hx : x ∈ homog R M d) : map f x ∈ homog R M' d

theorem **algHom_eq_pow_mul** {A : Type*} [CommRing A] [Algebra R A] (φ ψ : SymmetricAlgebra R M →ₐ[R] A) (s : A) (h : ∀ m, φ (SymmetricAlgebra.ι R M m) = s * ψ (SymmetricAlgebra.ι R M m)) {d : ℕ} {x : SymmetricAlgebra R M} (hx : x ∈ homog R M d) : φ x = s ^ d * ψ x

theorem **mem_homog_iff_isHomogeneous** {ι : Type*} (b : Basis ι R M) (d : ℕ) (x : SymmetricAlgebra R M) : x ∈ homog R M d ↔ (SymmetricAlgebra.equivMvPolynomial b x).IsHomogeneous d

@[simp] theorem **mapOver_ι** (j : M →ₗ[R] W) (m : M) : mapOver S j (SymmetricAlgebra.ι R M m) = SymmetricAlgebra.ι S W (j m)

@[simp] theorem **baseChangeHom_tmul** (j : M →ₗ[R] W) (s : S) (x : SymmetricAlgebra R M) : baseChangeHom S j (s ⊗ₜ x) = s • mapOver S j x

@[simp] theorem **baseChangeEquiv_tmul** {j : M →ₗ[R] W} (h : IsBaseChange S j) (s : S) (x : SymmetricAlgebra R M) : baseChangeEquiv h (s ⊗ₜ x) = s • mapOver S j x

theorem **isBaseChange_dual** [Module.Finite R M] [Module.Projective R M] {j : M →ₗ[R] W} (h : IsBaseChange S j) : IsBaseChange S h.toDual

theorem **mapOver_mem_homog** (j : M →ₗ[R] W) {d : ℕ} {x : SymmetricAlgebra R M} (hx : x ∈ homog R M d) : mapOver S j x ∈ homog S W d

theorem **mapOver_injective** [Module.Projective R M] {j : M →ₗ[R] W} (h : IsBaseChange S j) (hS : Function.Injective (algebraMap R S)) : Function.Injective (mapOver S j)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.MemSaturatedOrderIsoSubspaceSpanSymmApply

theorem **saturation_eq_comap_torsion** (N : Submodule R M) : saturation N = (Submodule.torsion R (M ⧸ N)).comap N.mkQ

theorem **mem_saturation_iff'** {N : Submodule R M} {x : M} : x ∈ saturation N ↔ ∃ a ∈ R⁰, a • x ∈ N

theorem **mem_saturation_iff** [IsDomain R] {N : Submodule R M} {x : M} : x ∈ saturation N ↔ ∃ a : R, a ≠ 0 ∧ a • x ∈ N

theorem **coe_saturation** [IsDomain R] (N : Submodule R M) : (saturation N : Set M) = {x | ∃ a : R, a ≠ 0 ∧ a • x ∈ N}

theorem **mem_saturation_of_smul_mem** [IsDomain R] {N : Submodule R M} {a : R} {x : M} (ha : a ≠ 0) (hx : a • x ∈ N) : x ∈ saturation N

theorem **le_saturation** (N : Submodule R M) : N ≤ saturation N

theorem **saturation_mono** {N N' : Submodule R M} (h : N ≤ N') : saturation N ≤ saturation N'

theorem **saturation_saturation** (N : Submodule R M) : saturation (saturation N) = saturation N

theorem **saturationClosure_apply** (N : Submodule R M) : saturationClosure R M N = saturation N

theorem **saturationClosure_isClosed_iff** (N : Submodule R M) : (saturationClosure R M).IsClosed N ↔ IsSaturatedSubmodule N

theorem **IsSaturatedSubmodule.saturation_eq** {N : Submodule R M} (h : IsSaturatedSubmodule N) : saturation N = N

theorem **isSaturatedSubmodule_iff_le** {N : Submodule R M} : IsSaturatedSubmodule N ↔ saturation N ≤ N

theorem **isSaturatedSubmodule_iff'** {N : Submodule R M} : IsSaturatedSubmodule N ↔ ∀ a ∈ R⁰, ∀ x : M, a • x ∈ N → x ∈ N

theorem **isSaturatedSubmodule_iff** [IsDomain R] {N : Submodule R M} : IsSaturatedSubmodule N ↔ ∀ a : R, a ≠ 0 → ∀ x : M, a • x ∈ N → x ∈ N

theorem **IsSaturatedSubmodule.mem_of_smul_mem** [IsDomain R] {N : Submodule R M} (h : IsSaturatedSubmodule N) {a : R} (ha : a ≠ 0) {x : M} (hx : a • x ∈ N) : x ∈ N

theorem **IsSaturatedSubmodule.smul_mem_iff** [IsDomain R] {N : Submodule R M} (h : IsSaturatedSubmodule N) {a : R} (ha : a ≠ 0) {x : M} : a • x ∈ N ↔ x ∈ N

theorem **isSaturatedSubmodule_iff_torsion_eq_bot** {N : Submodule R M} : IsSaturatedSubmodule N ↔ Submodule.torsion R (M ⧸ N) = ⊥

theorem **isSaturatedSubmodule_iff_isTorsionFree** [IsDomain R] {N : Submodule R M} : IsSaturatedSubmodule N ↔ Module.IsTorsionFree R (M ⧸ N)

theorem **IsSaturatedSubmodule.isTorsionFree_quotient** [IsDomain R] {N : Submodule R M} (h : IsSaturatedSubmodule N) : Module.IsTorsionFree R (M ⧸ N)

theorem **isSaturatedSubmodule_saturation** (N : Submodule R M) : IsSaturatedSubmodule (saturation N)

theorem **IsSaturatedSubmodule.saturation_le_iff** {N P : Submodule R M} (h : IsSaturatedSubmodule P) : saturation N ≤ P ↔ N ≤ P

theorem **IsSaturatedSubmodule.eq_saturation** {N P : Submodule R M} (h : IsSaturatedSubmodule P) (h₁ : N ≤ P) (h₂ : P ≤ saturation N) : P = saturation N

theorem **saturation_eq_of_le_of_le** {N P : Submodule R M} (h₁ : N ≤ P) (h₂ : P ≤ saturation N) : saturation P = saturation N

theorem **saturation_bot** : saturation (⊥ : Submodule R M) = Submodule.torsion R M

theorem **saturation_top** : saturation (⊤ : Submodule R M) = ⊤

theorem **isSaturatedSubmodule_top** : IsSaturatedSubmodule (⊤ : Submodule R M)

theorem **isSaturatedSubmodule_bot_iff** [IsDomain R] : IsSaturatedSubmodule (⊥ : Submodule R M) ↔ Module.IsTorsionFree R M

theorem **isSaturatedSubmodule_bot** [IsDomain R] [Module.IsTorsionFree R M] : IsSaturatedSubmodule (⊥ : Submodule R M)

theorem **torsion_le_saturation** (N : Submodule R M) : Submodule.torsion R M ≤ saturation N

theorem **saturation_eq_top_iff** {N : Submodule R M} : saturation N = ⊤ ↔ Module.IsTorsion R (M ⧸ N)

theorem **saturation_inf** (N P : Submodule R M) : saturation (N ⊓ P) = saturation N ⊓ saturation P

theorem **saturation_sup_saturation** (N P : Submodule R M) : saturation (saturation N ⊔ saturation P) = saturation (N ⊔ P)

theorem **IsSaturatedSubmodule.inf** {N P : Submodule R M} (hN : IsSaturatedSubmodule N) (hP : IsSaturatedSubmodule P) : IsSaturatedSubmodule (N ⊓ P)

theorem **isSaturatedSubmodule_iInf** {ι : Sort*} {N : ι → Submodule R M} (h : ∀ i, IsSaturatedSubmodule (N i)) : IsSaturatedSubmodule (⨅ i, N i)

theorem **saturation_comap** (f : M →ₗ[R] M') (N : Submodule R M') : saturation (N.comap f) = (saturation N).comap f

theorem **IsSaturatedSubmodule.comap** {N : Submodule R M'} (h : IsSaturatedSubmodule N) (f : M →ₗ[R] M') : IsSaturatedSubmodule (N.comap f)

theorem **isSaturatedSubmodule_ker** [IsDomain R] [Module.IsTorsionFree R M'] (f : M →ₗ[R] M') : IsSaturatedSubmodule (LinearMap.ker f)

theorem **map_saturation_le** (f : M →ₗ[R] M') (N : Submodule R M) : (saturation N).map f ≤ saturation (N.map f)

theorem **map_saturation_of_injective** {f : M →ₗ[R] M'} (hf : Function.Injective f) (N : Submodule R M) : (saturation N).map f = LinearMap.range f ⊓ saturation (N.map f)

theorem **map_saturation_equiv** (e : M ≃ₗ[R] M') (N : Submodule R M) : (saturation N).map (e : M →ₗ[R] M') = saturation (N.map (e : M →ₗ[R] M'))

theorem **isSaturatedSubmodule_map_equiv_iff** (e : M ≃ₗ[R] M') {N : Submodule R M} : IsSaturatedSubmodule (N.map (e : M →ₗ[R] M')) ↔ IsSaturatedSubmodule N

theorem **isSaturatedSubmodule_comap_iff_of_surjective** {f : M →ₗ[R] M'} (hf : Function.Surjective f) {N : Submodule R M'} : IsSaturatedSubmodule (N.comap f) ↔ IsSaturatedSubmodule N

theorem **IsSaturatedSubmodule.map_subtype** {P : Submodule R M} {N : Submodule R P} (hN : IsSaturatedSubmodule N) (hP : IsSaturatedSubmodule P) : IsSaturatedSubmodule (N.map P.subtype)

theorem **isSaturatedSubmodule_of_isCompl** [IsDomain R] [Module.IsTorsionFree R M] {N P : Submodule R M} (h : IsCompl N P) : IsSaturatedSubmodule N

theorem **IsSaturatedSubmodule.inf_smul_top** [IsDomain R] {N : Submodule R M} (h : IsSaturatedSubmodule N) {a : R} (ha : a ≠ 0) : N ⊓ a • (⊤ : Submodule R M) = a • N

theorem **saturation_smul** [IsDomain R] (N : Submodule R M) {a : R} (ha : a ≠ 0) : saturation (a • N) = saturation N

theorem **isTorsion_saturation_quotient** (N : Submodule R M) : Module.IsTorsion R (saturation N ⧸ N.submoduleOf (saturation N))

theorem **isSaturatedSubmodule_int_iff** {A : Type*} [AddCommGroup A] (N : Submodule ℤ A) : IsSaturatedSubmodule N ↔ N.toAddSubgroup.NSMulSaturated

theorem **isSaturatedSubmodule_of_field** {K V : Type*} [Field K] [AddCommGroup V] [Module K V] (W : Submodule K V) : IsSaturatedSubmodule W

theorem **rank_saturation** (N : Submodule R M) : Module.rank R (saturation N) = Module.rank R N

theorem **finrank_saturation** [Module.Finite R M] (N : Submodule R M) : Module.finrank R (saturation N) = Module.finrank R N

theorem **le_saturation_of_rank_le** {N P : Submodule R M} (h : N ≤ P) (hN : Module.rank R N < Cardinal.aleph0) (hr : Module.rank R P ≤ Module.rank R N) : P ≤ saturation N

theorem **le_saturation_iff_finrank_eq** [Module.Finite R M] {N P : Submodule R M} (h : N ≤ P) : P ≤ saturation N ↔ Module.finrank R N = Module.finrank R P

theorem **IsSaturatedSubmodule.eq_of_le_of_finrank_eq** [Module.Finite R M] {N P : Submodule R M} (hN : IsSaturatedSubmodule N) (h : N ≤ P) (hr : Module.finrank R N = Module.finrank R P) : N = P

theorem **saturationIn_def** (M N : Submodule R V) : saturationIn M N = M ⊓ saturation N

theorem **mem_saturationIn_iff** [IsDomain R] {M N : Submodule R V} {x : V} : x ∈ saturationIn M N ↔ x ∈ M ∧ ∃ a : R, a ≠ 0 ∧ a • x ∈ N

theorem **saturationIn_le** (M N : Submodule R V) : saturationIn M N ≤ M

theorem **le_saturationIn** {M N : Submodule R V} (h : N ≤ M) : N ≤ saturationIn M N

theorem **saturationIn_mono** {M M' N N' : Submodule R V} (hM : M ≤ M') (hN : N ≤ N') : saturationIn M N ≤ saturationIn M' N'

theorem **saturationIn_top** (N : Submodule R V) : saturationIn ⊤ N = saturation N

theorem **isSaturatedIn_top_iff** {N : Submodule R V} : IsSaturatedIn ⊤ N ↔ IsSaturatedSubmodule N

theorem **saturationIn_inf** (M N : Submodule R V) : saturationIn M (M ⊓ N) = saturationIn M N

theorem **saturationIn_saturationIn** (M N : Submodule R V) : saturationIn M (saturationIn M N) = saturationIn M N

theorem **isSaturatedIn_saturationIn** (M N : Submodule R V) : IsSaturatedIn M (saturationIn M N)

theorem **saturation_saturationIn** {M N : Submodule R V} (h : N ≤ M) : saturation (saturationIn M N) = saturation N

theorem **map_subtype_saturation** (M : Submodule R V) (N : Submodule R M) : (saturation N).map M.subtype = saturationIn M (N.map M.subtype)

theorem **submoduleOf_saturationIn** (M N : Submodule R V) : (saturationIn M N).submoduleOf M = saturation (N.submoduleOf M)

theorem **map_subtype_saturation_submoduleOf** (M N : Submodule R V) : (saturation (N.submoduleOf M)).map M.subtype = saturationIn M N

theorem **IsSaturatedIn.le** {M N : Submodule R V} (h : IsSaturatedIn M N) : N ≤ M

theorem **IsSaturatedIn.saturationIn_eq** {M N : Submodule R V} (h : IsSaturatedIn M N) : saturationIn M N = N

theorem **isSaturatedIn_iff** {M N : Submodule R V} : IsSaturatedIn M N ↔ N ≤ M ∧ IsSaturatedSubmodule (N.submoduleOf M)

theorem **isSaturatedIn_map_subtype_iff** {M : Submodule R V} {N : Submodule R M} : IsSaturatedIn M (N.map M.subtype) ↔ IsSaturatedSubmodule N

theorem **isSaturatedIn_iff_isTorsionFree** [IsDomain R] {M N : Submodule R V} : IsSaturatedIn M N ↔ N ≤ M ∧ Module.IsTorsionFree R (M ⧸ N.submoduleOf M)

theorem **isSaturatedIn_iff_smul_mem** [IsDomain R] {M N : Submodule R V} : IsSaturatedIn M N ↔ N ≤ M ∧ ∀ a : R, a ≠ 0 → ∀ x ∈ M, a • x ∈ N → x ∈ N

theorem **IsSaturatedSubmodule.isSaturatedIn_inf** {W : Submodule R V} (h : IsSaturatedSubmodule W) (M : Submodule R V) : IsSaturatedIn M (M ⊓ W)

theorem **IsSaturatedIn.trans** {M P N : Submodule R V} (hP : IsSaturatedIn M P) (hN : IsSaturatedIn P N) : IsSaturatedIn M N

theorem **map_saturationIn_of_injective** {f : V →ₗ[R] V'} (hf : Function.Injective f) (M N : Submodule R V) : (saturationIn M N).map f = saturationIn (M.map f) (N.map f)

theorem **rank_saturationIn** [IsDomain R] {M N : Submodule R V} (h : N ≤ M) : Module.rank R (saturationIn M N) = Module.rank R N

theorem **finrank_saturationIn** [IsDomain R] {M N : Submodule R V} [Module.Finite R M] (h : N ≤ M) : Module.finrank R (saturationIn M N) = Module.finrank R N

theorem **mk'_mem_localized'_iff** (N : Submodule R M) (x : M) (s : p) : IsLocalizedModule.mk' f x s ∈ N.localized' S p f ↔ ∃ a ∈ p, a • x ∈ N

theorem **localized'_saturation** (hp : p ≤ R⁰) (N : Submodule R M) : (saturation N).localized' S p f = saturation (N.localized' S p f)

theorem **IsSaturatedSubmodule.localized'** (hp : p ≤ R⁰) {N : Submodule R M} (h : IsSaturatedSubmodule N) : IsSaturatedSubmodule (N.localized' S p f)

theorem **IsSaturatedSubmodule.comap_localized'** (hp : p ≤ R⁰) {N : Submodule R M} (h : IsSaturatedSubmodule N) : ((N.localized' S p f).restrictScalars R).comap f = N

theorem **localized'_saturationIn** (hp : p ≤ R⁰) (P N : Submodule R M) : (saturationIn P N).localized' S p f = saturationIn (P.localized' S p f) (N.localized' S p f)

theorem **localized_saturation_atPrime** [IsDomain R] (P : Ideal R) [P.IsPrime] (N : Submodule R M) : (saturation N).localized P.primeCompl = saturation (N.localized P.primeCompl)

theorem **isSaturatedSubmodule_iff_localized_maximal** [IsDomain R] {N : Submodule R M} : IsSaturatedSubmodule N ↔ ∀ (P : Ideal R) [P.IsMaximal], IsSaturatedSubmodule (N.localized P.primeCompl)

theorem **mem_span_localization_iff** (N : Submodule R V) (x : V) : x ∈ Submodule.span S (N : Set V) ↔ ∃ a ∈ p, a • x ∈ N

theorem **isLocalizedModule_inclusionSpan** (M : Submodule R V) : IsLocalizedModule p (M.inclusionSpan S) where

theorem **span_localization_inf** (M N : Submodule R V) : Submodule.span S ((M ⊓ N : Submodule R V) : Set V) = Submodule.span S (M : Set V) ⊓ Submodule.span S (N : Set V)

theorem **span_localization_inf_restrictScalars** (M : Submodule R V) (W : Submodule S V) : Submodule.span S ((M ⊓ W.restrictScalars R : Submodule R V) : Set V) = Submodule.span S (M : Set V) ⊓ W

theorem **span_localization_saturation** (hp : p ≤ R⁰) (N : Submodule R V) : Submodule.span S (saturation N : Set V) = saturation (Submodule.span S (N : Set V))

theorem **span_localization_saturationIn** (hp : p ≤ R⁰) (M N : Submodule R V) : Submodule.span S (saturationIn M N : Set V) = saturationIn (Submodule.span S (M : Set V)) (Submodule.span S (N : Set V))

theorem **IsSaturatedIn.span_localization** (hp : p ≤ R⁰) {M N : Submodule R V} (h : IsSaturatedIn M N) : IsSaturatedIn (Submodule.span S (M : Set V)) (Submodule.span S (N : Set V))

theorem **saturation_eq_comap_localized'** (N : Submodule R M) : saturation N = ((N.localized' K R⁰ f).restrictScalars R).comap f

theorem **saturatedOrderIsoSubspace_apply** (N : {N : Submodule R M // IsSaturatedSubmodule N}) : saturatedOrderIsoSubspace K f N = Submodule.span K (f '' N.1)

theorem **saturatedOrderIsoSubspace_symm_apply** (W : Submodule K V) : ((saturatedOrderIsoSubspace K f).symm W).1 = (W.restrictScalars R).comap f

theorem **localized'_saturation_fractionRing** (N : Submodule R M) : (saturation N).localized' K R⁰ f = N.localized' K R⁰ f

theorem **saturation_eq_span** (N : Submodule R V) : saturation N = (Submodule.span K (N : Set V)).restrictScalars R

theorem **saturationIn_eq_inf_span** (M N : Submodule R V) : saturationIn M N = M ⊓ (Submodule.span K (N : Set V)).restrictScalars R

theorem **map_subtype_saturation_eq_inf_span** (M : Submodule R V) (N : Submodule R M) : (saturation N).map M.subtype = M ⊓ (Submodule.span K (N.map M.subtype : Set V)).restrictScalars R

theorem **mem_saturation_iff_mem_span** {M : Submodule R V} {N : Submodule R M} {x : M} : x ∈ saturation N ↔ (x : V) ∈ Submodule.span K (N.map M.subtype : Set V)

theorem **isSaturatedSubmodule_restrictScalars** (W : Submodule K V) : IsSaturatedSubmodule (W.restrictScalars R)

theorem **isSaturatedIn_inf_restrictScalars** (M : Submodule R V) (W : Submodule K V) : IsSaturatedIn M (M ⊓ W.restrictScalars R)

theorem **isSaturatedSubmodule_comap_restrictScalars** {M : Type*} [AddCommGroup M] [Module R M] (f : M →ₗ[R] V) (W : Submodule K V) : IsSaturatedSubmodule ((W.restrictScalars R).comap f)

theorem **isSaturatedIn_iff_inf_span_eq** {M N : Submodule R V} : IsSaturatedIn M N ↔ M ⊓ (Submodule.span K (N : Set V)).restrictScalars R = N

theorem **isSaturatedIn_iff_exists_subspace** {M N : Submodule R V} : IsSaturatedIn M N ↔ ∃ W : Submodule K V, M ⊓ W.restrictScalars R = N

theorem **span_inf_restrictScalars** (M : Submodule R V) (W : Submodule K V) : Submodule.span K ((M ⊓ W.restrictScalars R : Submodule R V) : Set V) = Submodule.span K (M : Set V) ⊓ W

theorem **span_inf_restrictScalars_of_le** {M : Submodule R V} {W : Submodule K V} (h : W ≤ Submodule.span K (M : Set V)) : Submodule.span K ((M ⊓ W.restrictScalars R : Submodule R V) : Set V) = W

theorem **span_saturationIn** {M N : Submodule R V} (h : N ≤ M) : Submodule.span K (saturationIn M N : Set V) = Submodule.span K (N : Set V)

theorem **map_saturatedOrderIsoSubspaceSpan_apply** (M : Submodule R V) (N : {N : Submodule R M // IsSaturatedSubmodule N}) : (saturatedOrderIsoSubspaceSpan K M N).map (Submodule.span K (M : Set V)).subtype = Submodule.span K (N.1.map M.subtype : Set V)

theorem **mem_saturatedOrderIsoSubspaceSpan_symm_apply** (M : Submodule R V) (W : Submodule K (Submodule.span K (M : Set V))) (x : M) : x ∈ ((saturatedOrderIsoSubspaceSpan K M).symm W).1 ↔ M.inclusionSpan K x ∈ W

theorem **saturatedInOrderIso_apply** (M : Submodule R V) (N : {N : Submodule R V // IsSaturatedIn M N}) : (saturatedInOrderIso K M N).1 = Submodule.span K (N.1 : Set V)

theorem **saturatedInOrderIso_symm_apply** (M : Submodule R V) (W : Set.Iic (Submodule.span K (M : Set V))) : ((saturatedInOrderIso K M).symm W).1 = M ⊓ W.1.restrictScalars R

theorem **rank_eq_rank_span** (N : Submodule R V) : Module.rank R N = Module.rank K (Submodule.span K (N : Set V))

theorem **rank_inf_restrictScalars** {M : Submodule R V} {W : Submodule K V} (h : W ≤ Submodule.span K (M : Set V)) : Module.rank R (M ⊓ W.restrictScalars R : Submodule R V) = Module.rank K W

theorem **finrank_inf_restrictScalars** {M : Submodule R V} {W : Submodule K V} [FiniteDimensional K W] (h : W ≤ Submodule.span K (M : Set V)) : Module.finrank R (M ⊓ W.restrictScalars R : Submodule R V) = Module.finrank K W

theorem **smul_inf_restrictScalars** (M : Submodule R V) (W : Submodule K V) {c : K} (hc : c ≠ 0) : c • M ⊓ W.restrictScalars R = c • (M ⊓ W.restrictScalars R)

theorem **saturationIn_smul_left** (M N : Submodule R V) {c : K} (hc : c ≠ 0) : saturationIn (c • M) N = c • saturationIn M N

theorem **isSaturatedSubmodule_span_one_two** : IsSaturatedSubmodule (Submodule.span ℤ {((1, 2) : ℤ × ℤ)})

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.PreservesFiniteColimitsTwistFunctor

theorem **secGroup_rel** {V : X.Opens} (s : secGroup F L V) (i j : L.ι) {W : X.Opens} (hi : W ≤ V ⊓ L.U i) (hj : W ≤ V ⊓ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resM F hi (s.1 i) = resG X hij (L.g i j : Γ(X, L.U i ⊓ L.U j)) • resM F hj (s.1 j)

theorem **secRes_smul** {V W : X.Opens} (h : W ≤ V) (r : Γ(X, V)) (m : secGroup F L V) : secRes F L h (r • m) = resG X h r • secRes F L h m

theorem **isSheaf_secPresheaf** : (secPresheaf F L).IsSheaf

theorem **coord_mem** {V : X.Opens} (s : Γ(twist F L, V)) : coord s ∈ secGroup F L V

@[simp] theorem **coord_mkSec** {V : X.Opens} (f : ∀ i, Γ(F, V ⊓ L.U i)) (hf : f ∈ secGroup F L V) : coord (mkSec f hf) = f

@[ext] theorem **sec_ext** {V : X.Opens} {s t : Γ(twist F L, V)} (h : ∀ i, coord s i = coord t i) : s = t

@[simp] theorem **coord_zero** {V : X.Opens} (i : L.ι) : coord (0 : Γ(twist F L, V)) i = 0

@[simp] theorem **coord_add** {V : X.Opens} (s t : Γ(twist F L, V)) (i : L.ι) : coord (s + t) i = coord s i + coord t i

@[simp] theorem **coord_neg** {V : X.Opens} (s : Γ(twist F L, V)) (i : L.ι) : coord (-s) i = - coord s i

@[simp] theorem **coord_sub** {V : X.Opens} (s t : Γ(twist F L, V)) (i : L.ι) : coord (s - t) i = coord s i - coord t i

@[simp] theorem **coord_smul** {V : X.Opens} (a : Γ(X, V)) (s : Γ(twist F L, V)) (i : L.ι) : coord (a • s) i = resG X (inf_le_left : V ⊓ L.U i ≤ V) a • coord s i

@[simp] theorem **coord_resM** {V W : X.Opens} (h : W ≤ V) (s : Γ(twist F L, V)) (i : L.ι) : coord (resM (twist F L) h s) i = resM F (inf_le_inf_right _ h) (coord s i)

theorem **coord_rel** {V : X.Opens} (s : Γ(twist F L, V)) (i j : L.ι) {W : X.Opens} (hi : W ≤ V ⊓ L.U i) (hj : W ≤ V ⊓ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resM F hi (coord s i) = resG X hij (L.g i j : Γ(X, L.U i ⊓ L.U j)) • resM F hj (coord s j)

theorem **sec_ext_of_charts** {ι : Type*} {U : ι → X.Opens} (hU : ⊔ k, U k = ⊤) {V : X.Opens} {s t : Γ(twist F L, V)} (h : ∀ i k, resM F (inf_le_left : V ⊓ L.U i ⊓ U k ≤ V ⊓ L.U i) (coord s i) = resM F inf_le_left (coord t i)) : s = t

@[simp] theorem **coord_map_app** (φ : F ⟶ G) {V : X.Opens} (s : Γ(twist F L, V)) (i : L.ι) : coord ((map L φ).app V s) i = φ.app _ (coord s i)

@[simp] theorem **twistFunctor_obj** (L : LB X) (F : X.Modules) : (twistFunctor L).obj F = twist F L

@[simp] theorem **twistFunctor_map** (L : LB X) (φ : F ⟶ G) : (twistFunctor L).map φ = map L φ

@[simp] theorem **map_id** (L : LB X) (F : X.Modules) : map L (𝟙 F) = 𝟙 (twist F L)

@[reassoc] theorem **map_comp** (L : LB X) (φ : F ⟶ G) (ψ : G ⟶ H) : map L (φ ≫ ψ) = map L φ ≫ map L ψ

theorem **hom_ext** {φ ψ : G ⟶ twist F L} (h : ∀ (V : X.Opens) (s : Γ(G, V)) (i : L.ι), coord (φ.app V s) i = coord (ψ.app V s) i) : φ = ψ

theorem **chartEquiv_apply** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (s : Γ(twist F L, V)) : chartEquiv hL i h s = resM F (le_inf le_rfl h) (coord s i)

theorem **coord_chartEquiv_symm** (hL : L.IsLB) (i j : L.ι) {V : X.Opens} (h : V ≤ L.U i) (t : Γ(F, V)) : coord ((chartEquiv hL i h).symm t) j = resG X (le_inf inf_le_right (inf_le_left.trans h)) (L.g j i : Γ(X, L.U j ⊓ L.U i)) • resM F inf_le_left t

theorem **chartEquiv_resM** (hL : L.IsLB) (i : L.ι) {V W : X.Opens} (h : V ≤ L.U i) (h' : W ≤ V) (s : Γ(twist F L, V)) : chartEquiv hL i (h'.trans h) (resM (twist F L) h' s) = resM F h' (chartEquiv hL i h s)

theorem **chartEquiv_change** (hL : L.IsLB) (i j : L.ι) {V : X.Opens} (hi : V ≤ L.U i) (hj : V ≤ L.U j) (s : Γ(twist F L, V)) : chartEquiv hL i hi s = resG X (le_inf hi hj) (L.g i j : Γ(X, L.U i ⊓ L.U j)) • chartEquiv hL j hj s

theorem **chartEquiv_map_app** (hL : L.IsLB) (φ : F ⟶ G) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (s : Γ(twist F L, V)) : chartEquiv hL i h ((map L φ).app V s) = φ.app V (chartEquiv hL i h s)

theorem **restrictIso_hom_app** (hL : L.IsLB) (i : L.ι) (g : Y ⟶ X) [IsOpenImmersion g] (hg : g.opensRange ≤ L.U i) (W : Y.Opens) (s : Γ(twist F L, g ''ᵁ W)) : ((restrictIso F hL i g hg).hom.app W s : Γ(F, g ''ᵁ W)) = chartEquiv hL i ((g.image_le_opensRange W).trans hg) s

theorem **isQuasicoherent_twist_iff** (hL : L.IsLB) : (twist F L).IsQuasicoherent ↔ F.IsQuasicoherent

theorem **isQuasicoherent_twist** (hL : L.IsLB) [h : F.IsQuasicoherent] : (twist F L).IsQuasicoherent

theorem **isFiniteType_twist_iff** (hL : L.IsLB) : (twist F L).IsFiniteType ↔ F.IsFiniteType

theorem **isFiniteType_twist** (hL : L.IsLB) [h : F.IsFiniteType] : (twist F L).IsFiniteType

theorem **finite_sections_twist_iff** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) : Module.Finite Γ(X, V) Γ(twist F L, V) ↔ Module.Finite Γ(X, V) Γ(F, V)

theorem **stalkEquiv_germ** (hL : L.IsLB) (i : L.ι) {x : X} (hx : x ∈ L.U i) {V : X.Opens} (hxV : x ∈ V) (s : Γ(twist F L, V)) : stalkEquiv F hL i hx ((twist F L).presheaf.germ V x hxV s) = F.presheaf.germ (V ⊓ L.U i) x ⟨hxV, hx⟩ (coord s i)

theorem **nonempty_stalkEquiv** (hL : L.IsLB) (x : X) : Nonempty (((twist F L).presheaf.stalk x :) ≃ₗ[X.presheaf.stalk x] (F.presheaf.stalk x :))

theorem **nontrivial_stalk_twist_iff** (hL : L.IsLB) (x : X) : Nontrivial ((twist F L).presheaf.stalk x :) ↔ Nontrivial (F.presheaf.stalk x :)

theorem **support_twist** (hL : L.IsLB) : QCoh.support (twist F L) = QCoh.support F

theorem **suppDim_twist** (hL : L.IsLB) : QCoh.suppDim (twist F L) = QCoh.suppDim F

theorem **assPoints_twist** (hL : L.IsLB) : QCoh.assPoints (twist F L) = QCoh.assPoints F

theorem **isFiniteQC_twist_iff** (hL : L.IsLB) : QCoh.IsFiniteQC (twist F L) ↔ QCoh.IsFiniteQC F

theorem **isFiniteQC_twist** (hL : L.IsLB) [h : QCoh.IsFiniteQC F] : QCoh.IsFiniteQC (twist F L)

theorem **isFlatOver_twist_iff** (hL : L.IsLB) [F.IsQuasicoherent] {R : Type*} [CommRing R] (ρ : R →+* Γ(X, ⊤)) : QCoh.IsFlatOver (twist F L) ρ ↔ QCoh.IsFlatOver F ρ

@[simp] theorem **coord_unitTwistIso_hom_app** {V : X.Opens} (s : Γ(twist (unitMod X) L, V)) (i : L.ι) : LB.coord ((unitTwistIso L).hom.app V s) i = coord s i

@[simp] theorem **coord_unitTwistIso_inv_app** {V : X.Opens} (s : Γ(toMod L, V)) (i : L.ι) : coord ((unitTwistIso L).inv.app V s) i = LB.coord s i

theorem **coord_tmulFun** {V : X.Opens} (t : Γ(F, V)) (σ : Γ(toMod L, V)) (i : L.ι) : coord (tmulFun t σ) i = LB.coord σ i • resM F inf_le_left t

@[simp] theorem **coord_tmul** {V : X.Opens} (t : Γ(F, V)) (σ : Γ(toMod L, V)) (i : L.ι) : coord (tmul F L V t σ) i = LB.coord σ i • resM F inf_le_left t

theorem **resM_tmul** {V W : X.Opens} (h : W ≤ V) (t : Γ(F, V)) (σ : Γ(toMod L, V)) : resM (twist F L) h (tmul F L V t σ) = tmul F L W (resM F h t) (resM (toMod L) h σ)

theorem **chartEquiv_tmul** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (t : Γ(F, V)) (σ : Γ(toMod L, V)) : chartEquiv hL i h (tmul F L V t σ) = LB.chartEquiv hL i h σ • t

theorem **map_app_tmul** (φ : F ⟶ G) {V : X.Opens} (t : Γ(F, V)) (σ : Γ(toMod L, V)) : (map L φ).app V (tmul F L V t σ) = tmul G L V (φ.app V t) σ

@[simp] theorem **coord_secHom_app** (σ : LB.Sec L) {V : X.Opens} (t : Γ(F, V)) (i : L.ι) : coord ((secHom F σ).app V t) i = resG X (inf_le_right : V ⊓ L.U i ≤ L.U i) (σ.f i) • resM F inf_le_left t

theorem **chartEquiv_secHom_app** (hL : L.IsLB) (σ : LB.Sec L) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (t : Γ(F, V)) : chartEquiv hL i h ((secHom F σ).app V t) = resG X h (σ.f i) • t

@[reassoc] theorem **secHom_naturality** (σ : LB.Sec L) (φ : F ⟶ G) : φ ≫ secHom G σ = secHom F σ ≫ map L φ

theorem **isUnit_resG_f** (σ : LB.Sec L) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (hV : V ≤ Out.nonzeroLocus σ) : IsUnit (resG X h (σ.f i))

theorem **bijective_of_eq_smul** {A B : Type*} {V : X.Opens} {f : A → B} {p : A → Γ(F, V)} {q : B → Γ(F, V)} (hp : Function.Bijective p) (hq : Function.Bijective q) {u : Γ(X, V)} (hu : IsUnit u) (h : ∀ a, q (f a) = u • p a) : Function.Bijective f

theorem **bijective_secHom_app** (hL : L.IsLB) (σ : LB.Sec L) {V : X.Opens} (hV : V ≤ Out.nonzeroLocus σ) : Function.Bijective ((secHom F σ).app V)

theorem **isIso_secHom_genSec** (hL : L.IsLB) (t : LB.Gen L) (ht : t.IsGen) : IsIso (secHom F (DVR.genSec t ht))

theorem **genIso_hom** (hL : L.IsLB) (t : LB.Gen L) (ht : t.IsGen) : (genIso F hL t ht).hom = secHom F (DVR.genSec t ht)

theorem **isGen_one_triv** : LB.Gen.IsGen (L

theorem **coord_trivIso_hom_app** {V : X.Opens} (t : Γ(F, V)) (i : PUnit.{1}) : coord (L

theorem **resG_div_g_mul** (p q : (LB.div N L).ι) {W : X.Opens} (h : W ≤ (LB.div N L).U p ⊓ (LB.div N L).U q) (hL : W ≤ L.U p.2 ⊓ L.U q.2) (hN : W ≤ N.U p.1 ⊓ N.U q.1) : resG X h ((LB.div N L).g p q : Γ(X, (LB.div N L).U p ⊓ (LB.div N L).U q)) * resG X hL (L.g p.2 q.2 : Γ(X, L.U p.2 ⊓ L.U q.2)) = resG X hN (N.g p.1 q.1 : Γ(X, N.U p.1 ⊓ N.U q.1))

theorem **div_U_le_left** (p : (LB.div N L).ι) : (LB.div N L).U p ≤ N.U p.1

theorem **div_U_le_right** (p : (LB.div N L).ι) : (LB.div N L).U p ≤ L.U p.2

theorem **resM_chartEquiv_symm** (hL : L.IsLB) (i : L.ι) {V W : X.Opens} (h : V ≤ L.U i) (h' : W ≤ V) (t : Γ(F, V)) : resM (twist F L) h' ((chartEquiv hL i h).symm t) = (chartEquiv hL i (h'.trans h)).symm (resM F h' t)

theorem **coord_divFun** (hL : L.IsLB) {V : X.Opens} (t : Γ(twist F N, V)) (p : (LB.div N L).ι) : coord (divFun hL t) p = (chartEquiv hL p.2 (inf_le_right.trans (div_U_le_right p))).symm (resM F (inf_le_inf_left V (div_U_le_left p)) (coord t p.1))

theorem **coord_coord_divHom_app** (hL : L.IsLB) {V : X.Opens} (t : Γ(twist F N, V)) (n : N.ι) (l' l : L.ι) {W : X.Opens} (h : W ≤ V ⊓ (LB.div N L).U (n, l') ⊓ L.U l) (hn : W ≤ V ⊓ N.U n) (hl : W ≤ L.U l ⊓ L.U l') : resM F h (coord (coord ((divHom F N hL).app V t) (n, l')) l) = resG X hl (L.g l l' : Γ(X, L.U l ⊓ L.U l')) • resM F hn (coord t n)

theorem **chartEquiv_divHom_app** (hL : L.IsLB) (hN : N.IsLB) (p : (LB.div N L).ι) {V : X.Opens} (h : V ≤ (LB.div N L).U p) (t : Γ(twist F N, V)) : chartEquiv hL p.2 (h.trans (div_U_le_right p)) (chartEquiv (Semi.LB.isLB_div hN hL) p h ((divHom F N hL).app V t)) = chartEquiv hN p.1 (h.trans (div_U_le_left p)) t

theorem **isIso_divHom** (hL : L.IsLB) (hN : N.IsLB) : IsIso (divHom F N hL)

@[reassoc] theorem **divHom_naturality** (hL : L.IsLB) (φ : F ⟶ G) : map N φ ≫ divHom G N hL = divHom F N hL ≫ map (LB.div N L) (map L φ)

theorem **divIso_hom** (hL : L.IsLB) (hN : N.IsLB) : (divIso F hL hN).hom = divHom F N hL

theorem **coord_divIso_inv_app** (hL : L.IsLB) (hN : N.IsLB) {V : X.Opens} (s : Γ(twist (twist F L) (LB.div N L), V)) (n : N.ι) (l : L.ι) {W : X.Opens} (hn : W ≤ V ⊓ N.U n) (h : W ≤ V ⊓ (LB.div N L).U (n, l) ⊓ L.U l) : resM F hn (coord ((divIso F hL hN).inv.app V s) n) = resM F h (coord (coord s (n, l)) l)

theorem **coord_swapCoord** {V : X.Opens} (s : Γ(twist (twist F L) M, V)) (l : L.ι) (m : M.ι) : coord (swapCoord s l) m = resM F (inf_right_comm V (L.U l) (M.U m)).le (coord (coord s m) l)

theorem **coord_coord_swapFun** {V : X.Opens} (s : Γ(twist (twist F L) M, V)) (l : L.ι) (m : M.ι) : coord (coord (swapFun s) l) m = resM F (inf_right_comm V (L.U l) (M.U m)).le (coord (coord s m) l)

@[simp] theorem **coord_coord_swapHom_app** {V : X.Opens} (s : Γ(twist (twist F L) M, V)) (l : L.ι) (m : M.ι) : coord (coord ((swapHom F L M).app V s) l) m = resM F (inf_right_comm V (L.U l) (M.U m)).le (coord (coord s m) l)

theorem **swapHom_swapHom** : swapHom F L M ≫ swapHom F M L = 𝟙 _

theorem **coord_homOfSec_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V : X.Opens} (s : Γ(twist F L, V)) (p : M.ι) (q : L.ι) {W : X.Opens} (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q) : resM F hp (coord ((homOfSec F hL hM w).app V s) p) = resG X h (LB.homCoord w p q) • resM F hq (coord s q)

theorem **chartEquiv_homOfSec_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) (p : M.ι) (q : L.ι) {V : X.Opens} (hp : V ≤ M.U p) (hq : V ≤ L.U q) (s : Γ(twist F L, V)) : chartEquiv hM p hp ((homOfSec F hL hM w).app V s) = resG X (le_inf hp hq) (LB.homCoord w p q) • chartEquiv hL q hq s

@[reassoc] theorem **homOfSec_naturality** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) (φ : F ⟶ G) : map L φ ≫ homOfSec G hL hM w = homOfSec F hL hM w ≫ map M φ

theorem **bijective_homOfSec_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) {V : X.Opens} (hV : V ≤ Out.nonzeroLocus w) : Function.Bijective ((homOfSec F hL hM w).app V)

theorem **isoOfGen_hom** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) : (isoOfGen F hL hM w hw).hom = homOfSec F hL hM (DVR.genSec w hw)

theorem **coord_isoOfGen_hom_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) {V : X.Opens} (s : Γ(twist F L, V)) (p : M.ι) (q : L.ι) {W : X.Opens} (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q) : resM F hp (coord ((isoOfGen F hL hM w hw).hom.app V s) p) = resG X h (isoUnit w p q : Γ(X, M.U p ⊓ L.U q)) • resM F hq (coord s q)

theorem **nonempty_iso_of_lbIso** (hL : L.IsLB) (hM : M.IsLB) (h : LBIso L M) : Nonempty (twist F L ≅ twist F M)

theorem **eq_homOfSec** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) (φ : twist F L ⟶ twist F M) (h : ∀ (V : X.Opens) (s : Γ(twist F L, V)) (p : M.ι) (q : L.ι), resM F (inf_le_left : V ⊓ M.U p ⊓ L.U q ≤ V ⊓ M.U p) (coord (φ.app V s) p) = resG X (inf_le_inf_right _ inf_le_right : V ⊓ M.U p ⊓ L.U q ≤ M.U p ⊓ L.U q) (LB.homCoord w p q) • resM F (inf_le_inf_right _ inf_le_left) (coord s q)) : φ = homOfSec F hL hM w

theorem **isoOfGen_idIso_hom** (hL : L.IsLB) : (isoOfGen F hL hL (idIso L) (idIso_isGen hL)).hom = 𝟙 _

theorem **coord_isoOfGen_inv_app** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Iso L M) (hw : w.IsGen) {V : X.Opens} (t : Γ(twist F M, V)) (p : M.ι) (q : L.ι) {W : X.Opens} (hp : W ≤ V ⊓ M.U p) (hq : W ≤ V ⊓ L.U q) (h : W ≤ M.U p ⊓ L.U q) : resM F hq (coord ((isoOfGen F hL hM w hw).inv.app V t) q) = (((res h (isoUnit w p q))⁻¹ : Γ(X, W)ˣ) : Γ(X, W)) • resM F hp (coord t p)

theorem **homOfSec_comp** (hL : L.IsLB) (hM : M.IsLB) (hN : N.IsLB) (w : LB.Sec (LB.div M L)) (v : LB.Sec (LB.div N M)) (u : LB.Sec (LB.div N L)) (h : ∀ (r : N.ι) (p : M.ι) (q : L.ι), resG X (inf_le_left : N.U r ⊓ L.U q ⊓ M.U p ≤ N.U r ⊓ L.U q) (LB.homCoord u r q) = resG X (inf_le_inf_right _ inf_le_left) (LB.homCoord v r p) * resG X (le_inf inf_le_right (inf_le_left.trans inf_le_right)) (LB.homCoord w p q)) : homOfSec F hL hM w ≫ homOfSec F hM hN v = homOfSec F hL hN u

theorem **homOfSec_unitTwistIso** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) : homOfSec (unitMod X) hL hM w ≫ (unitTwistIso M).hom = (unitTwistIso L).hom ≫ LB.homOfSec hL hM w

theorem **coord_twistTwistIso_hom_app** (hL : L.IsLB) (hM : M.IsLB) (hN : N.IsLB) (w : LB.Gen (LB.div (LB.div N L) M)) (hw : w.IsGen) {V : X.Opens} (s : Γ(twist (twist F L) M, V)) (n : N.ι) (l : L.ι) (m : M.ι) {W : X.Opens} (hn : W ≤ V ⊓ N.U n) (hm : W ≤ V ⊓ M.U m ⊓ L.U l) (h : W ≤ N.U n ⊓ L.U l ⊓ M.U m) : resM F hn (coord ((twistTwistIso F hL hM hN w hw).hom.app V s) n) = resG X h (tripleUnit w n l m : Γ(X, N.U n ⊓ L.U l ⊓ M.U m)) • resM F hm (coord (coord s m) l)

theorem **twistTwistNatIso_hom_app** (hL : L.IsLB) (hM : M.IsLB) (hN : N.IsLB) (w : LB.Gen (LB.div (LB.div N L) M)) (hw : w.IsGen) (F : X.Modules) : (twistTwistNatIso hL hM hN w hw).hom.app F = (twistTwistIso F hL hM hN w hw).hom

@[reassoc] theorem **twistTwistIso_naturality** (hL : L.IsLB) (hM : M.IsLB) (hN : N.IsLB) (w : LB.Gen (LB.div (LB.div N L) M)) (hw : w.IsGen) (φ : F ⟶ G) : map M (map L φ) ≫ (twistTwistIso G hL hM hN w hw).hom = (twistTwistIso F hL hM hN w hw).hom ≫ map N φ

theorem **nonempty_twistTwistIso** (hL : L.IsLB) (hM : M.IsLB) (hN : N.IsLB) (h : LBIso M (LB.div N L)) : Nonempty (twist (twist F L) M ≅ twist F N)

@[simp] theorem **twistEquivalence_functor** (hL : L.IsLB) : (twistEquivalence hL).functor = twistFunctor L

@[simp] theorem **twistEquivalence_inverse** (hL : L.IsLB) : (twistEquivalence hL).inverse = twistFunctor (LB.div (LB.triv X) L)

theorem **isEquivalence_twistFunctor** (hL : L.IsLB) : (twistFunctor L).IsEquivalence

theorem **preservesLimits_twistFunctor** (hL : L.IsLB) : Limits.PreservesLimits (twistFunctor L)

theorem **preservesColimits_twistFunctor** (hL : L.IsLB) : Limits.PreservesColimits (twistFunctor L)

theorem **preservesFiniteLimits_twistFunctor** (hL : L.IsLB) : Limits.PreservesFiniteLimits (twistFunctor L)

theorem **preservesFiniteColimits_twistFunctor** (hL : L.IsLB) : Limits.PreservesFiniteColimits (twistFunctor L)

theorem **exact_map_iff** (hL : L.IsLB) (S : ShortComplex X.Modules) : (S.map (twistFunctor L)).Exact ↔ S.Exact

theorem **shortExact_map** (hL : L.IsLB) {S : ShortComplex X.Modules} (hS : S.ShortExact) : (S.map (twistFunctor L)).ShortExact

theorem **mono_map_iff** (hL : L.IsLB) (φ : F ⟶ G) : Mono (map L φ) ↔ Mono φ

theorem **epi_map_iff** (hL : L.IsLB) (φ : F ⟶ G) : Epi (map L φ) ↔ Epi φ

theorem **isIso_map_iff** (hL : L.IsLB) (φ : F ⟶ G) : IsIso (map L φ) ↔ IsIso φ

theorem **isZero_twist_iff** (hL : L.IsLB) : Limits.IsZero (twist F L) ↔ Limits.IsZero F

theorem **le_pull_inf** (V : X.Opens) (i : L.ι) : f ⁻¹ᵁ (V ⊓ L.U i) ≤ f ⁻¹ᵁ V ⊓ (L.pull f).U fun _ => i

theorem **coord_projIso_hom_app** (G : Y.Modules) {V : X.Opens} (s : Γ(twist G (L.pull f), f ⁻¹ᵁ V)) (i : L.ι) : (coord ((projIso L f G).hom.app V s) i : Γ(G, f ⁻¹ᵁ (V ⊓ L.U i))) = resM G (le_pull_inf f V i) (coord s fun _ => i)

theorem **coord_projIso_inv_app** (G : Y.Modules) {V : X.Opens} (s : Γ(twist ((Scheme.Modules.pushforward f).obj G) L, V)) (j : (L.pull f).ι) : coord (F

theorem **coord_pullbackHom_app_unit** {V : X.Opens} (s : Γ(twist F L, V)) (j : (L.pull f).ι) : coord ((pullbackHom F L f).app (f ⁻¹ᵁ V) (((Scheme.Modules.pullbackPushforwardAdjunction f).unit.app (twist F L)).app V s)) j = resM ((Scheme.Modules.pullback f).obj F) (LB.pull_inf_le f V j) (((Scheme.Modules.pullbackPushforwardAdjunction f).unit.app F).app _ (coord s (j PUnit.unit)))

@[reassoc] theorem **pullbackHom_naturality** (φ : F ⟶ G) : (Scheme.Modules.pullback f).map (map L φ) ≫ pullbackHom G L f = pullbackHom F L f ≫ map (L.pull f) ((Scheme.Modules.pullback f).map φ)

theorem **isIso_pullbackHom** (hL : L.IsLB) : IsIso (pullbackHom F L f)

theorem **pullbackIso_hom** (hL : L.IsLB) : (pullbackIso F f hL).hom = pullbackHom F L f

@[simp] theorem **coord_refineHom_app** {V : X.Opens} (s : Γ(twist F L, V)) (m : M.ι) : coord ((refineHom F r hU hg).app V s) m = resM F (inf_le_inf_left V (hU m)) (coord s (r m))

@[reassoc] theorem **refineHom_naturality** (φ : F ⟶ G) : map L φ ≫ refineHom G r hU hg = refineHom F r hU hg ≫ map M φ

theorem **isIso_refineHom** (hL : L.IsLB) (hM : M.IsLB) : IsIso (refineHom F r hU hg)

theorem **refineIso_hom** (hL : L.IsLB) (hM : M.IsLB) : (refineIso F r hU hg hL hM).hom = refineHom F r hU hg

theorem **lbZPow_natCast** (L : LB X) (n : ℕ) : lbZPow L n = lbPow L n

theorem **twistZ_natCast** (F : X.Modules) (L : LB X) (n : ℕ) : twistZ F L n = twist F (lbPow L n)

theorem **isLB_lbZPow** (hL : L.IsLB) (n : ℤ) : (lbZPow L n).IsLB

theorem **zpow_U_le** (j : (lbZPow L n).ι) : (lbZPow L n).U j ≤ L.U (j PUnit.unit)

theorem **le_zpow_U** (j : (lbZPow L n).ι) : L.U (j PUnit.unit) ≤ (lbZPow L n).U j

theorem **le_zidx_U** (n : ℤ) (i : L.ι) : L.U i ≤ (lbZPow L n).U (zidx L n i)

theorem **res_zpow_g** (i j : (lbZPow L n).ι) {W : X.Opens} (h : W ≤ (lbZPow L n).U i ⊓ (lbZPow L n).U j) (h' : W ≤ L.U (i PUnit.unit) ⊓ L.U (j PUnit.unit)) : res h ((lbZPow L n).g i j) = res h' (L.g (i PUnit.unit) (j PUnit.unit)) ^ n

theorem **zcoord_rel** {V : X.Opens} (s : Γ(twistZ F L n, V)) (i j : L.ι) {W : X.Opens} (hi : W ≤ V ⊓ L.U i) (hj : W ≤ V ⊓ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resM F hi (zcoord s i) = ((res hij (L.g i j) ^ n : Γ(X, W)ˣ) : Γ(X, W)) • resM F hj (zcoord s j)

theorem **coord_eq_zcoord** {V : X.Opens} (s : Γ(twistZ F L n, V)) (j : (lbZPow L n).ι) : coord s j = resM F (inf_le_inf_left V (zpow_U_le j)) (zcoord s (j PUnit.unit))

theorem **zsec_ext** {V : X.Opens} {s t : Γ(twistZ F L n, V)} (h : ∀ i, zcoord s i = zcoord t i) : s = t

@[simp] theorem **zcoord_zmkSec** {V : X.Opens} (f : ∀ i, Γ(F, V ⊓ L.U i)) (hf) (i : L.ι) : zcoord (zmkSec (n

@[simp] theorem **zcoord_zero** {V : X.Opens} (i : L.ι) : zcoord (0 : Γ(twistZ F L n, V)) i = 0

@[simp] theorem **zcoord_add** {V : X.Opens} (s t : Γ(twistZ F L n, V)) (i : L.ι) : zcoord (s + t) i = zcoord s i + zcoord t i

@[simp] theorem **zcoord_smul** {V : X.Opens} (a : Γ(X, V)) (s : Γ(twistZ F L n, V)) (i : L.ι) : zcoord (a • s) i = resG X (inf_le_left : V ⊓ L.U i ≤ V) a • zcoord s i

@[simp] theorem **zcoord_resM** {V W : X.Opens} (h : W ≤ V) (s : Γ(twistZ F L n, V)) (i : L.ι) : zcoord (resM (twistZ F L n) h s) i = resM F (inf_le_inf_right _ h) (zcoord s i)

@[simp] theorem **zcoord_map_app** (φ : F ⟶ G) {V : X.Opens} (s : Γ(twistZ F L n, V)) (i : L.ι) : zcoord ((map (lbZPow L n) φ).app V s) i = φ.app _ (zcoord s i)

@[simp] theorem **zcoord_eqToHom_app** (h : n = k) (e : twistZ F L n = twistZ F L k) {V : X.Opens} (s : Γ(twistZ F L n, V)) (i : L.ι) : zcoord (Scheme.Modules.Hom.app (eqToHom e) V s) i = zcoord s i

theorem **zhom_ext** {φ ψ : G ⟶ twistZ F L n} (h : ∀ (V : X.Opens) (s : Γ(G, V)) (i : L.ι), zcoord (φ.app V s) i = zcoord (ψ.app V s) i) : φ = ψ

theorem **zchartEquiv_apply** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (s : Γ(twistZ F L n, V)) : zchartEquiv hL n i h s = resM F (le_inf le_rfl h) (zcoord s i)

theorem **zchartEquiv_resM** (hL : L.IsLB) (i : L.ι) {V W : X.Opens} (h : V ≤ L.U i) (h' : W ≤ V) (s : Γ(twistZ F L n, V)) : zchartEquiv hL n i (h'.trans h) (resM (twistZ F L n) h' s) = resM F h' (zchartEquiv hL n i h s)

theorem **zchartEquiv_change** (hL : L.IsLB) (i j : L.ι) {V : X.Opens} (hi : V ≤ L.U i) (hj : V ≤ L.U j) (s : Γ(twistZ F L n, V)) : zchartEquiv hL n i hi s = ((res (le_inf hi hj) (L.g i j) ^ n : Γ(X, V)ˣ) : Γ(X, V)) • zchartEquiv hL n j hj s

theorem **zfun_rel** (σ : LB.Sec (lbZPow L m)) (i j : L.ι) {W : X.Opens} (hi : W ≤ L.U i) (hj : W ≤ L.U j) (hij : W ≤ L.U i ⊓ L.U j) : resG X hi (zfun σ i) = ((res hij (L.g i j) ^ m : Γ(X, W)ˣ) : Γ(X, W)) ∗ resG X hj (zfun σ j)

theorem **f_eq_zfun** (σ : LB.Sec (lbZPow L m)) (j : (lbZPow L m).ι) : σ.f j = resG X (zpow_U_le j) (zfun σ (j PUnit.unit))

theorem **zglobal_ext** {σ τ : LB.Sec (lbZPow L m)} (h : ∀ i, zfun σ i = zfun τ i) : σ = τ

@[simp] theorem **zfun_zmkGlobal** (f : ∀ i, Γ(X, L.U i)) (hf) (i : L.ι) : zfun (zmkGlobal (m

@[simp] theorem **zfun_zero** (i : L.ι) : zfun (0 : LB.Sec (lbZPow L m)) i = 0

@[simp] theorem **zfun_add** (σ τ : LB.Sec (lbZPow L m)) (i : L.ι) : zfun (σ + τ) i = zfun σ i + zfun τ i

@[simp] theorem **zfun_smul** (a : Γ(X, ⊤)) (σ : LB.Sec (lbZPow L m)) (i : L.ι) : zfun (a • σ) i = resG X le_top a ∗ zfun σ i

@[simp] theorem **zfun_zone** (i : L.ι) : zfun (zone L) i = 1

@[simp] theorem **zfun_zofSec** (σ : LB.Sec L) (i : L.ι) : zfun (zofSec σ) i = σ.f i

@[simp] theorem **zfun_zmul** (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) (i : L.ι) : zfun (zmul σ τ) i = zfun σ i * zfun τ i

theorem **isUnit_resG_zfun** (σ : LB.Sec (lbZPow L m)) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (hV : V ≤ Out.nonzeroLocus σ) : IsUnit (resG X h (zfun σ i))

theorem **zcoord_mulFun** (σ : LB.Sec (lbZPow L m)) (h : n + m = k) {V : X.Opens} (s : Γ(twistZ F L n, V)) (i : L.ι) : zcoord (mulFun σ h s) i = resG X (inf_le_right : V ⊓ L.U i ≤ L.U i) (zfun σ i) • zcoord s i

@[simp] theorem **zcoord_mulSecOfEq_app** (σ : LB.Sec (lbZPow L m)) (h : n + m = k) {V : X.Opens} (s : Γ(twistZ F L n, V)) (i : L.ι) : zcoord ((mulSecOfEq F σ h).app V s) i = resG X (inf_le_right : V ⊓ L.U i ≤ L.U i) (zfun σ i) • zcoord s i

theorem **zchartEquiv_mulSecOfEq_app** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (h : n + m = k) (i : L.ι) {V : X.Opens} (hi : V ≤ L.U i) (s : Γ(twistZ F L n, V)) : zchartEquiv hL k i hi ((mulSecOfEq F σ h).app V s) = resG X hi (zfun σ i) • zchartEquiv hL n i hi s

@[reassoc] theorem **mulSec_eqToHom** (σ : LB.Sec (lbZPow L m)) (h : n + m = k) : mulSec F σ n ≫ eqToHom (congrArg (twistZ F L) h) = mulSecOfEq F σ h

@[reassoc] theorem **mulSecOfEq_naturality** (σ : LB.Sec (lbZPow L m)) (h : n + m = k) (φ : F ⟶ G) : map (lbZPow L n) φ ≫ mulSecOfEq G σ h = mulSecOfEq F σ h ≫ map (lbZPow L k) φ

theorem **mulSecOfEq_comp_of_zfun** {m'' : ℤ} (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) (ρ : LB.Sec (lbZPow L m'')) (hρ : ∀ i, zfun ρ i = zfun σ i * zfun τ i) (h₁ : n + m = k) (h₂ : k + m' = l) (h₃ : n + m'' = l) : mulSecOfEq F σ h₁ ≫ mulSecOfEq F τ h₂ = mulSecOfEq F ρ h₃

@[reassoc] theorem **mulSecOfEq_comp** (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) (h₁ : n + m = k) (h₂ : k + m' = l) (h₃ : n + (m + m') = l) : mulSecOfEq F σ h₁ ≫ mulSecOfEq F τ h₂ = mulSecOfEq F (zmul σ τ) h₃

@[reassoc] theorem **mulSecOfEq_comm** (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) (h₁ : n + m = k) (h₂ : k + m' = l) (h₃ : n + m' = k') (h₄ : k' + m = l) : mulSecOfEq F σ h₁ ≫ mulSecOfEq F τ h₂ = mulSecOfEq F τ h₃ ≫ mulSecOfEq F σ h₄

theorem **mulSec_comp** (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) (n : ℤ) (h : n + (m + m') = n + m + m'

theorem **mulSec_comm** (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) (n : ℤ) (h : n + m' + m = n + m + m'

@[simp] theorem **mulSecOfEq_zone** (h : n + 0 = n) : mulSecOfEq F (zone L) h = 𝟙 _

@[simp] theorem **mulSecOfEq_zero** (h : n + m = k) : mulSecOfEq F (0 : LB.Sec (lbZPow L m)) h = 0

theorem **mulSecOfEq_add** (σ τ : LB.Sec (lbZPow L m)) (h : n + m = k) : mulSecOfEq F (σ + τ) h = mulSecOfEq F σ h + mulSecOfEq F τ h

theorem **mulSecOfEq_smul_app** (a : Γ(X, ⊤)) (σ : LB.Sec (lbZPow L m)) (h : n + m = k) {V : X.Opens} (s : Γ(twistZ F L n, V)) : (mulSecOfEq F (a • σ) h).app V s = resG X (le_top : V ≤ ⊤) a • (mulSecOfEq F σ h).app V s

theorem **bijective_mulSecOfEq_app** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (h : n + m = k) {V : X.Opens} (hV : V ≤ Out.nonzeroLocus σ) : Function.Bijective ((mulSecOfEq F σ h).app V)

theorem **isGen_zoneGen** : (zoneGen L).IsGen

@[simp] theorem **zcoord_twistZZeroIso_hom_app** (hL : L.IsLB) {V : X.Opens} (t : Γ(F, V)) (i : L.ι) : zcoord ((twistZZeroIso F hL).hom.app V t) i = resM F inf_le_left t

@[simp] theorem **zcoord_twistZOneIso_hom_app** (hL : L.IsLB) {V : X.Opens} (s : Γ(twist F L, V)) (i : L.ι) : zcoord ((twistZOneIso F hL).hom.app V s) i = coord s i

theorem **twistZZeroIso_hom_mulSecOfEq** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (h : 0 + m = m) : (twistZZeroIso F hL).hom ≫ mulSecOfEq F σ h = secHom F σ

theorem **res_div_g** (p q : (LB.div N L).ι) {W : X.Opens} (h : W ≤ (LB.div N L).U p ⊓ (LB.div N L).U q) (hN : W ≤ N.U p.1 ⊓ N.U q.1) (hL : W ≤ L.U p.2 ⊓ L.U q.2) : res h ((LB.div N L).g p q) = res hN (N.g p.1 q.1) * (res hL (L.g p.2 q.2))⁻¹

theorem **zpow_mul_inv_mul_identity_commGroup** {A : Type*} [CommGroup A] (a b c d e : A) (n m : ℤ) : (a * b⁻¹ * d) ^ n * (a * c⁻¹ * e) ^ m = a ^ (n + m) * (b ^ n)⁻¹ * (c ^ m)⁻¹ * (d ^ n * e ^ m)

theorem **res_zaddGen** (p : (LB.div (LB.div (lbZPow L k) (lbZPow L n)) (lbZPow L m)).ι) {W : X.Opens} (h : W ≤ (LB.div (LB.div (lbZPow L k) (lbZPow L n)) (lbZPow L m)).U p) (h₁ : W ≤ L.U (p.1.1 PUnit.unit) ⊓ L.U (p.1.2 PUnit.unit)) (h₂ : W ≤ L.U (p.1.1 PUnit.unit) ⊓ L.U (p.2 PUnit.unit)) : res h ((zaddGen L n m k).s p) = res h₁ (L.g (p.1.1 PUnit.unit) (p.1.2 PUnit.unit)) ^ n * res h₂ (L.g (p.1.1 PUnit.unit) (p.2 PUnit.unit)) ^ m

theorem **isGen_zaddGen** (hL : L.IsLB) (h : n + m = k) : (zaddGen L n m k).IsGen

theorem **zcoord_twistZAddIso_hom_app** (hL : L.IsLB) (h : n + m = k) {V : X.Opens} (s : Γ(twistZ (twistZ F L n) L m, V)) (i : L.ι) : zcoord ((twistZAddIso F hL h).hom.app V s) i = resM F (le_inf le_rfl inf_le_right) (zcoord (zcoord s i) i)

@[reassoc] theorem **twistZAddIso_naturality** (hL : L.IsLB) (h : n + m = k) (φ : F ⟶ G) : map (lbZPow L m) (map (lbZPow L n) φ) ≫ (twistZAddIso G hL h).hom = (twistZAddIso F hL h).hom ≫ map (lbZPow L k) φ

theorem **secHom_twistZAddIso_hom** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (h : n + m = k) : secHom (twistZ F L n) σ ≫ (twistZAddIso F hL h).hom = mulSecOfEq F σ h

theorem **pull_zpow_U_le** (j : ((lbZPow L n).pull f).ι) : ((lbZPow L n).pull f).U j ≤ (lbZPow (L.pull f) n).U (zidx _ n (pidx L f (j PUnit.unit PUnit.unit)))

theorem **res_pull_g** (i j : (L.pull f).ι) {W : Y.Opens} (h : W ≤ (L.pull f).U i ⊓ (L.pull f).U j) (h' : W ≤ f ⁻¹ᵁ (L.U (i PUnit.unit) ⊓ L.U (j PUnit.unit))) : res h ((L.pull f).g i j) = res h' (pullU f _ (L.g (i PUnit.unit) (j PUnit.unit)))

theorem **pull_zpow_g** (i j : ((lbZPow L n).pull f).ι) : ((lbZPow L n).pull f).g i j = res (inf_le_inf (pull_zpow_U_le f i) (pull_zpow_U_le f j)) ((lbZPow (L.pull f) n).g (zidx _ n (pidx L f (i PUnit.unit PUnit.unit))) (zidx _ n (pidx L f (j PUnit.unit PUnit.unit))))

theorem **le_pidx_inf** (V : X.Opens) (i : L.ι) : f ⁻¹ᵁ (V ⊓ L.U i) ≤ f ⁻¹ᵁ V ⊓ (L.pull f).U (pidx L f i)

theorem **zcoord_projZIso_hom_app** (hL : L.IsLB) (G : Y.Modules) {V : X.Opens} (s : Γ(twistZ G (L.pull f) n, f ⁻¹ᵁ V)) (i : L.ι) : (zcoord ((projZIso f hL G n).hom.app V s) i : Γ(G, f ⁻¹ᵁ (V ⊓ L.U i))) = resM G (le_pidx_inf f V i) (zcoord s (pidx L f i))

@[reassoc] theorem **projZIso_naturality** (hL : L.IsLB) {G G' : Y.Modules} (φ : G ⟶ G') (n : ℤ) : (Scheme.Modules.pushforward f).map (map (lbZPow (L.pull f) n) φ) ≫ (projZIso f hL G' n).hom = (projZIso f hL G n).hom ≫ map (lbZPow L n) ((Scheme.Modules.pushforward f).map φ)

@[simp] theorem **zfun_zpull** (σ : LB.Sec (lbZPow L m)) (j : (L.pull f).ι) : zfun (zpull f σ) j = (f.appLE _ _ (LB.pull_U_le f j)).hom (zfun σ (j PUnit.unit))

@[reassoc] theorem **projZIso_mulSecOfEq** (hL : L.IsLB) (G : Y.Modules) (σ : LB.Sec (lbZPow L m)) (h : n + m = k) : (Scheme.Modules.pushforward f).map (mulSecOfEq G (zpull f σ) h) ≫ (projZIso f hL G k).hom = (projZIso f hL G n).hom ≫ mulSecOfEq ((Scheme.Modules.pushforward f).obj G) σ h

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.SubsingletonOfSubsingletonFreeSheaf

lemma **freeSheafHomEquiv_comp** (U : C) {F G : Sheaf J AddCommGrpCat.{v}} (φ : (freeSheaf J).obj U ⟶ F) (f : F ⟶ G) : freeSheafHomEquiv U G (φ ≫ f) = f.hom.app (op U) (freeSheafHomEquiv U F φ)

lemma **freeSheaf_map_tautSection** {U V : C} (i : V ⟶ U) : ((freeSheaf J).map i).hom.app (op V) (tautSection J V) = ((freeSheaf J).obj U).obj.map i.op (tautSection J U)

lemma **freeSheafHomEquiv_map_comp** {U V : C} (i : V ⟶ U) {F : Sheaf J AddCommGrpCat.{v}} (φ : (freeSheaf J).obj U ⟶ F) : freeSheafHomEquiv V F ((freeSheaf J).map i ≫ φ) = F.obj.map i.op (freeSheafHomEquiv U F φ)

lemma **extEquiv_map** {F G : Sheaf J AddCommGrpCat.{v}} (f : F ⟶ G) (n : ℕ) (U : C) (x : F.H' n U) : extEquiv G n U (map f n U x) = (extEquiv F n U x).comp (Ext.mk₀ f) (add_zero n)

lemma **extEquiv_res** (F : Sheaf J AddCommGrpCat.{v}) (n : ℕ) {U V : C} (i : V ⟶ U) (x : F.H' n U) : extEquiv F n V (res F n i x) = (Ext.mk₀ ((freeSheaf J).map i)).comp (extEquiv F n U x) (zero_add n)

lemma **zeroEquiv_mk₀** (F : Sheaf J AddCommGrpCat.{v}) (U : C) (φ : (freeSheaf J).obj U ⟶ F) : zeroEquiv F U ((extEquiv F 0 U).symm (Ext.mk₀ φ)) = freeSheafHomEquiv U F φ

lemma **zeroEquiv_map** {F G : Sheaf J AddCommGrpCat.{v}} (f : F ⟶ G) (U : C) (x : F.H' 0 U) : zeroEquiv G U (map f 0 U x) = f.hom.app (op U) (zeroEquiv F U x)

lemma **zeroEquiv_res** (F : Sheaf J AddCommGrpCat.{v}) {U V : C} (i : V ⟶ U) (x : F.H' 0 U) : zeroEquiv F V (res F 0 i x) = F.obj.map i.op (zeroEquiv F U x)

lemma **terminalEquiv_apply** (F : Sheaf J AddCommGrpCat.{v}) (n : ℕ) {T : C} (hT : IsTerminal T) (x : F.H' n T) : terminalEquiv F n hT x = (Ext.mk₀ (freeSheafIsoConst hT).inv).comp (extEquiv F n T x) (zero_add n)

lemma **terminalEquiv_map** {F G : Sheaf J AddCommGrpCat.{v}} (f : F ⟶ G) (n : ℕ) {T : C} (hT : IsTerminal T) (x : F.H' n T) : terminalEquiv G n hT (map f n T x) = Sheaf.H.map f n (terminalEquiv F n hT x)

lemma **equiv₀_terminalEquiv** (F : Sheaf J AddCommGrpCat.{v}) {T : C} (hT : IsTerminal T) (x : F.H' 0 T) : Sheaf.H.equiv₀ F hT (terminalEquiv F 0 hT x) = zeroEquiv F T x

lemma **subsingleton_of_subsingleton_freeSheaf** (F : Sheaf J AddCommGrpCat.{v}) (n : ℕ) (U : C) [Subsingleton (((freeSheaf J).obj U).obj.obj (op U))] : Subsingleton (F.H' n U)

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.ToAbSheafPreservesFiniteLimits

@[simp] lemma **toAbSheaf_obj_obj** (F : X.Modules) : ((toAbSheaf X).obj F).obj = F.presheaf

@[simp] lemma **toAbSheaf_map_hom_app** {F G : X.Modules} (φ : F ⟶ G) (U : X.Opens) : ((toAbSheaf X).map φ).hom.app (op U) = φ.app U

lemma **restrictGlobal_top** (r : Γ(X, ⊤)) : restrictGlobal X ⊤ r = r

lemma **map_restrictGlobal** {U V : X.Opens} (i : V ⟶ U) (r : Γ(X, ⊤)) : X.presheaf.map i.op (restrictGlobal X U r) = restrictGlobal X V r

@[simp] lemma **smulHom_app_apply** (F : X.Modules) (r : Γ(X, ⊤)) (U : X.Opens) (x : Γ(F, U)) : (smulHom F r).app U x = restrictGlobal X U r • x

@[simp] lemma **smulHom_one** (F : X.Modules) : smulHom F 1 = 𝟙 F

@[simp] lemma **smulHom_zero** (F : X.Modules) : smulHom F 0 = 0

lemma **smulHom_add** (F : X.Modules) (r s : Γ(X, ⊤)) : smulHom F (r + s) = smulHom F r + smulHom F s

lemma **smulHom_mul** (F : X.Modules) (r s : Γ(X, ⊤)) : smulHom F (r * s) = smulHom F s ≫ smulHom F r

@[reassoc] lemma **smulHom_comp** {F G : X.Modules} (φ : F ⟶ G) (r : Γ(X, ⊤)) : smulHom F r ≫ φ = φ ≫ smulHom G r

lemma **smul_hom_def** {F G : X.Modules} (r : Γ(X, ⊤)) (φ : F ⟶ G) : r • φ = φ ≫ smulHom G r

lemma **smulHom_eq_smul_id** (F : X.Modules) (r : Γ(X, ⊤)) : smulHom F r = r • 𝟙 F

@[simp] lemma **smul_app_apply** {F G : X.Modules} (r : Γ(X, ⊤)) (φ : F ⟶ G) (U : X.Opens) (x : Γ(F, U)) : (r • φ).app U x = restrictGlobal X U r • φ.app U x

lemma **toBase_smul** (r : R) (x : WithAction ρ) : toBase ρ (r • x) = ρ (algebraMap R A r) (toBase ρ x)

lemma **toBase_linearMap** (f : M →+ N) (h : ∀ a x, f (ρ a x) = σ a (f x)) (x : WithAction ρ) : toBase σ (linearMap ρ σ R f h x) = f (toBase ρ x)

lemma **toBase_linearEquiv** (f : M ≃+ N) (h : ∀ a x, f (ρ a x) = σ a (f x)) (x : WithAction ρ) : toBase σ (linearEquiv ρ σ R f h x) = f (toBase ρ x)

lemma **map_id_apply** (x : F.H' n U) : map (𝟙 F) n U x = x

lemma **map_comp_apply** (f : F ⟶ G) (g : G ⟶ H) (x : F.H' n U) : map (f ≫ g) n U x = map g n U (map f n U x)

lemma **map_add_apply** (f g : F ⟶ G) (x : F.H' n U) : map (f + g) n U x = map f n U x + map g n U x

lemma **map_zero_apply** (x : F.H' n U) : map (0 : F ⟶ G) n U x = 0

lemma **res_id_apply** (x : F.H' n U) : res F n (𝟙 U) x = x

lemma **res_comp_apply** (i : V ⟶ U) (j : W ⟶ V) (x : F.H' n U) : res F n (j ≫ i) x = res F n j (res F n i x)

lemma **res_map_apply** (f : F ⟶ G) (i : V ⟶ U) (x : F.H' n U) : res G n i (map f n U x) = map f n V (res F n i x)

lemma **endAction_apply** (f : End F) (x : F.H' n U) : endAction F n U f x = map f n U x

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.TopologicalKrullDimLeOfIrreducibleCloseds

theorem **finite_sections** [IsFiniteQC F] (hU : IsAffineOpen U) : Module.Finite Γ(X, U) Γ(F, U)

theorem **exists_affine_mem_le** (x : X) (hx : x ∈ U) : ∃ V : X.Opens, IsAffineOpen V ∧ x ∈ V ∧ V ≤ U

theorem **exists_affine_mem** (x : X) : ∃ V : X.Opens, IsAffineOpen V ∧ x ∈ V

theorem **finite_stalk_of_finite_sections** [F.IsQuasicoherent] (hU : IsAffineOpen U) [Module.Finite Γ(X, U) Γ(F, U)] (x : U) : Module.Finite (X.presheaf.stalk (x : X)) (F.presheaf.stalk (x : X) :)

theorem **finite_sections_basicOpen** [F.IsQuasicoherent] (hU : IsAffineOpen U) [Module.Finite Γ(X, U) Γ(F, U)] (f : Γ(X, U)) : Module.Finite Γ(X, X.basicOpen f) Γ(F, X.basicOpen f)

theorem **IsFiniteQC.of_forall_exists_affine** [F.IsQuasicoherent] (h : ∀ x : X, ∃ U : X.Opens, IsAffineOpen U ∧ x ∈ U ∧ Module.Finite Γ(X, U) Γ(F, U)) : IsFiniteQC F

theorem **isFiniteQC_iff_affineCover** [F.IsQuasicoherent] {ι : Type*} (𝒰 : ι → X.Opens) (hcov : IsOpenCover 𝒰) (haff : ∀ i, IsAffineOpen (𝒰 i)) : IsFiniteQC F ↔ ∀ i, Module.Finite Γ(X, 𝒰 i) Γ(F, 𝒰 i)

theorem **isFiniteQC_iff_finite_top** [IsAffine X] [F.IsQuasicoherent] : IsFiniteQC F ↔ Module.Finite Γ(X, ⊤) Γ(F, ⊤)

theorem **IsFiniteQC.of_epi** (φ : F ⟶ G) [Epi φ] [IsFiniteQC F] [G.IsQuasicoherent] : IsFiniteQC G

theorem **IsFiniteQC.of_iso** (e : F ≅ G) [IsFiniteQC F] : IsFiniteQC G

theorem **isFiniteQC_iff_of_iso** (e : F ≅ G) : IsFiniteQC F ↔ IsFiniteQC G

theorem **isNoetherian_sections** [IsLocallyNoetherian X] [IsFiniteQC F] (hU : IsAffineOpen U) : _root_.IsNoetherian Γ(X, U) Γ(F, U)

theorem **IsFiniteQC.of_mono** [IsLocallyNoetherian X] (φ : F ⟶ G) [Mono φ] [F.IsQuasicoherent] [IsFiniteQC G] : IsFiniteQC F

theorem **IsFiniteQC.of_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) [IsFiniteQC S.X₁] [S.X₂.IsQuasicoherent] [IsFiniteQC S.X₃] : IsFiniteQC S.X₂

theorem **sum_app_app** (h : ∑ i, p i ≫ q i = 𝟙 F) (s : Γ(F, U)) : ∑ i, (q i).app U ((p i).app U s) = s

theorem **isQuasicoherent_of_sum_eq_id** (h : ∑ i, p i ≫ q i = 𝟙 F) [∀ i, (A i).IsQuasicoherent] : F.IsQuasicoherent

theorem **IsFiniteQC.of_sum_eq_id** (h : ∑ i, p i ≫ q i = 𝟙 F) [hA : ∀ i, IsFiniteQC (A i)] : IsFiniteQC F

theorem **IsFiniteQC.of_isZero** (h : IsZero F) : IsFiniteQC F

theorem **IsFiniteQC.of_isColimit_cofan** {ι : Type*} [Finite ι] {A : ι → X.Modules} [∀ i, IsFiniteQC (A i)] {c : Cofan A} (hc : IsColimit c) : IsFiniteQC c.pt

theorem **IsFiniteQC.of_isLimit_fan** {ι : Type*} [Finite ι] {A : ι → X.Modules} [∀ i, IsFiniteQC (A i)] {c : Fan A} (hc : IsLimit c) : IsFiniteQC c.pt

theorem **finite_sections_restrict_iff** (W : Y.Opens) : Module.Finite Γ(Y, W) Γ(F.restrict g, W) ↔ Module.Finite Γ(X, g ''ᵁ W) Γ(F, g ''ᵁ W)

theorem **IsFiniteQC.of_restrict** {ι : Type*} {Y : ι → Scheme.{u}} (g : ∀ i, Y i ⟶ X) [∀ i, IsOpenImmersion (g i)] (hcov : ∀ x : X, ∃ i, x ∈ (g i).opensRange) (h : ∀ i, IsFiniteQC (F.restrict (g i))) : IsFiniteQC F

theorem **algebraMap_Spec_top_bijective** : Function.Bijective (algebraMap R Γ(Spec R, ⊤))

theorem **isFiniteQC_Spec_iff** (M : (Spec R).Modules) [M.IsQuasicoherent] : IsFiniteQC M ↔ Module.Finite R Γ(M, ⊤)

theorem **isFiniteQC_tilde_iff** (M : ModuleCat.{u} R) : IsFiniteQC (tilde M) ↔ Module.Finite R M

theorem **not_isFiniteQC_tilde_rat** : ¬ IsFiniteQC (tilde (R

theorem **IsFiniteQC.of_generatingSections** [F.IsQuasicoherent] (σ : F.GeneratingSections) [σ.IsFiniteType] : IsFiniteQC F

theorem **IsFiniteQC.of_isFiniteType** [F.IsQuasicoherent] [F.IsFiniteType] : IsFiniteQC F

theorem **exists_presentation_over** [F.IsQuasicoherent] (hU : IsAffineOpen U) (s : Set Γ(F.restrict hU.fromSpec, ⊤)) (hs : Submodule.span Γ(X, U) s = ⊤) (t : Set (s →₀ Γ(X, U))) (ht : Submodule.span Γ(X, U) t = LinearMap.ker (Finsupp.linearCombination Γ(X, U) ((↑) : s → Γ(F.restrict hU.fromSpec, ⊤)))) : ∃ P : (F.over U).Presentation, P.generators.I = s ∧ P.relations.I = t

theorem **isFiniteQC_iff_isFiniteType** : IsFiniteQC F ↔ F.IsQuasicoherent ∧ F.IsFiniteType

theorem **IsFiniteQC.of_isFinitePresentation** [F.IsFinitePresentation] : IsFiniteQC F

theorem **IsFiniteQC.isFinitePresentation** [IsLocallyNoetherian X] [IsFiniteQC F] : F.IsFinitePresentation

theorem **isFiniteQC_iff_isFinitePresentation** [IsLocallyNoetherian X] : IsFiniteQC F ↔ F.IsFinitePresentation

theorem **mem_support_iff** {x : X} : x ∈ support F ↔ Nontrivial (F.presheaf.stalk x :)

theorem **notMem_support_iff** {x : X} : x ∉ support F ↔ Subsingleton (F.presheaf.stalk x :)

theorem **mem_support_iff_primeIdealOf** [F.IsQuasicoherent] (hU : IsAffineOpen U) (x : U) : (x : X) ∈ support F ↔ hU.primeIdealOf x ∈ Module.support Γ(X, U) Γ(F, U)

theorem **fromSpec_preimage_support** [F.IsQuasicoherent] (hU : IsAffineOpen U) : hU.fromSpec ⁻¹' support F = Module.support Γ(X, U) Γ(F, U)

theorem **mem_support_iff_annihilator_le** [IsFiniteQC F] (hU : IsAffineOpen U) (x : U) : (x : X) ∈ support F ↔ Module.annihilator Γ(X, U) Γ(F, U) ≤ (hU.primeIdealOf x).asIdeal

theorem **support_inter_eq_zeroLocus** [IsFiniteQC F] (hU : IsAffineOpen U) : support F ∩ U = X.zeroLocus (U

theorem **isClosed_support** [IsFiniteQC F] : IsClosed (support F)

theorem **mem_support_of_specializes** [F.IsQuasicoherent] {x y : X} (h : x ⤳ y) (hx : x ∈ support F) : y ∈ support F

theorem **support_subset_of_mono** (φ : F ⟶ G) [Mono φ] : support F ⊆ support G

theorem **support_subset_of_epi** (φ : F ⟶ G) [Epi φ] : support G ⊆ support F

theorem **support_eq_of_iso** (e : F ≅ G) : support F = support G

theorem **support_subset_union_of_exact** {S : ShortComplex X.Modules} (hS : S.Exact) : support S.X₂ ⊆ support S.X₁ ∪ support S.X₃

theorem **support_eq_union_of_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) : support S.X₂ = support S.X₁ ∪ support S.X₃

theorem **support_biprod** : support (F ⊞ G) = support F ∪ support G

theorem **support_eq_empty_iff** : support F = ∅ ↔ IsZero F

theorem **restrictStalkAddEquiv_germ** {Y : Scheme.{u}} (g : Y ⟶ X) [IsOpenImmersion g] (y : Y) {W : Y.Opens} (hy : y ∈ W) (s : Γ(F.restrict g, W)) : restrictStalkAddEquiv F g y ((F.restrict g).presheaf.germ W y hy s) = F.presheaf.germ (g ''ᵁ W) (g y) (y, hy, rfl) ((F.restrictAppIso g W).hom s)

theorem **restrictStalkAddEquiv_smul** {Y : Scheme.{u}} (g : Y ⟶ X) [IsOpenImmersion g] (y : Y) (r : X.presheaf.stalk (g y)) (t : ((F.restrict g).presheaf.stalk y :)) : restrictStalkAddEquiv F g y (g.stalkMap y r • t) = r • restrictStalkAddEquiv F g y t

theorem **support_restrict** {Y : Scheme.{u}} (g : Y ⟶ X) [IsOpenImmersion g] : support (F.restrict g) = g ⁻¹' support F

theorem **structureModuleSectionsEquiv_res** (h : V ≤ U) (s : Γ(structureModule X, U)) : structureModuleSectionsEquiv V ((structureModule X).presheaf.map (homOfLE h).op s) = X.presheaf.map (homOfLE h).op (structureModuleSectionsEquiv U s)

theorem **germ_smul_germ_one** (x : X) (hx : x ∈ U) (s : Γ(structureModule X, U)) : X.presheaf.germ U x hx (structureModuleSectionsEquiv U s) • (structureModule X).presheaf.germ ⊤ x trivial ((structureModuleSectionsEquiv ⊤).symm 1) = (structureModule X).presheaf.germ U x hx s

theorem **structureModuleStalkEquiv_germ** (x : X) (hx : x ∈ U) (s : Γ(structureModule X, U)) : structureModuleStalkEquiv x ((structureModule X).presheaf.germ U x hx s) = X.presheaf.germ U x hx (structureModuleSectionsEquiv U s)

theorem **support_structureModule** : support (structureModule X) = Set.univ

theorem **support_tilde** {R : CommRingCat.{u}} (M : ModuleCat.{u} R) : support (tilde M) = Module.support R M

theorem **isIrreducible_of_image** {S T : Type*} [TopologicalSpace S] [TopologicalSpace T] {f : S → T} (hf : Topology.IsInducing f) {s : Set S} (h : IsIrreducible (f '' s)) : IsIrreducible s

theorem **topologicalKrullDim_le_of_irreducibleCloseds** {T : Type*} [TopologicalSpace T] {d : WithBot ℕ∞} (h : ∀ Z : IrreducibleCloseds T, topologicalKrullDim (Z : Set T) ≤ d) : topologicalKrullDim T ≤ d

theorem **topologicalKrullDim_mono** {T : Type*} [TopologicalSpace T] {s t : Set T} (h : s ⊆ t) : topologicalKrullDim s ≤ topologicalKrullDim t

theorem **topologicalKrullDim_le_of_isIrreducible** {T : Type*} [TopologicalSpace T] {s : Set T} {d : WithBot ℕ∞} (h : ∀ Z ⊆ s, IsIrreducible Z → topologicalKrullDim Z ≤ d) : topologicalKrullDim s ≤ d

theorem **topologicalKrullDim_union** {T : Type*} [TopologicalSpace T] {s t : Set T} (hs : IsClosed s) (ht : IsClosed t) : topologicalKrullDim (s ∪ t :) = max (topologicalKrullDim s) (topologicalKrullDim t)

theorem **topologicalKrullDim_eq_iSup_closure** {T : Type*} [TopologicalSpace T] [QuasiSober T] {s : Set T} (hs : IsClosed s) : topologicalKrullDim s = ⨆ x ∈ s, topologicalKrullDim (closure {x} : Set T)

theorem **topologicalKrullDim_eq_bot_iff** {T : Type*} [TopologicalSpace T] : topologicalKrullDim T = ⊥ ↔ IsEmpty T

theorem **suppDim_eq_bot_iff** : suppDim F = ⊥ ↔ IsZero F

theorem **suppDim_lt_zero_iff** : suppDim F < 0 ↔ IsZero F

theorem **suppDim_le** : suppDim F ≤ topologicalKrullDim X

theorem **suppDim_le_of_support_subset** (h : support F ⊆ support G) : suppDim F ≤ suppDim G

theorem **suppDim_le_of_mono** (φ : F ⟶ G) [Mono φ] : suppDim F ≤ suppDim G

theorem **suppDim_le_of_epi** (φ : F ⟶ G) [Epi φ] : suppDim G ≤ suppDim F

theorem **suppDim_eq_of_iso** (e : F ≅ G) : suppDim F = suppDim G

theorem **suppDim_eq_max_of_shortExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) [IsFiniteQC S.X₁] [IsFiniteQC S.X₃] : suppDim S.X₂ = max (suppDim S.X₁) (suppDim S.X₃)

theorem **suppDim_eq_iSup_closure** [IsFiniteQC F] : suppDim F = ⨆ x ∈ support F, topologicalKrullDim (closure {x} : Set X)

theorem **suppDim_structureModule** : suppDim (structureModule X) = topologicalKrullDim X

theorem **suppDim_tilde** {R : CommRingCat.{u}} (M : ModuleCat.{u} R) [Module.Finite R M] : suppDim (tilde M) = Module.supportDim R M

theorem **mem_assPoints_iff** {x : X} : x ∈ assPoints F ↔ IsAssociatedPrime (maximalIdeal (X.presheaf.stalk x)) (F.presheaf.stalk x :)

theorem **mem_assPoints_iff_exists_pow** {x : X} : x ∈ assPoints F ↔ ∃ t : (F.presheaf.stalk x :), ∀ r : X.presheaf.stalk x, (∃ n : ℕ, r ^ n • t = 0) ↔ ¬ IsUnit r

theorem **assPoints_restrict** {Y : Scheme.{u}} (g : Y ⟶ X) [IsOpenImmersion g] : assPoints (F.restrict g) = g ⁻¹' assPoints F

theorem **mem_assPoints_iff_exists** [IsLocallyNoetherian X] {x : X} : x ∈ assPoints F ↔ ∃ t : (F.presheaf.stalk x :), ∀ r : X.presheaf.stalk x, r • t = 0 ↔ r ∈ maximalIdeal (X.presheaf.stalk x)

theorem **assPoints_subset_support** : assPoints F ⊆ support F

theorem **assPoints_subset_of_mono** (φ : F ⟶ G) [Mono φ] : assPoints F ⊆ assPoints G

theorem **assPoints_eq_of_iso** (e : F ≅ G) : assPoints F = assPoints G

theorem **assPoints_subset_union_of_exact** {S : ShortComplex X.Modules} (hS : S.Exact) [Mono S.f] : assPoints S.X₂ ⊆ assPoints S.X₁ ∪ assPoints S.X₃

theorem **mem_associatedPrimes_stalk_iff** [F.IsQuasicoherent] (hU : IsAffineOpen U) (x : U) (Q : Ideal (X.presheaf.stalk (x : X))) : Q ∈ associatedPrimes (X.presheaf.stalk (x : X)) (F.presheaf.stalk (x : X) :) ↔ Q.comap (algebraMap Γ(X, U) _) ∈ associatedPrimes Γ(X, U) Γ(F, U)

theorem **mem_assPoints_iff_primeIdealOf** [F.IsQuasicoherent] (hU : IsAffineOpen U) (x : U) : (x : X) ∈ assPoints F ↔ (hU.primeIdealOf x).asIdeal ∈ associatedPrimes Γ(X, U) Γ(F, U)

theorem **pointOfPrime_mem_assPoints_iff** [F.IsQuasicoherent] (hU : IsAffineOpen U) (P : PrimeSpectrum Γ(X, U)) : (pointOfPrime hU P : X) ∈ assPoints F ↔ P.asIdeal ∈ associatedPrimes Γ(X, U) Γ(F, U)

theorem **finite_assPoints_inter_affine** [IsFiniteQC F] (hU : IsAffineOpen U) : (assPoints F ∩ U).Finite

theorem **finite_assPoints_inter_of_isCompact** [IsFiniteQC F] {K : Set X} (hK : IsCompact K) : (assPoints F ∩ K).Finite

theorem **finite_assPoints** [CompactSpace X] [IsFiniteQC F] : (assPoints F).Finite

theorem **exists_mem_assPoints_specializes** [F.IsQuasicoherent] {x : X} (hx : x ∈ support F) : ∃ y ∈ assPoints F, y ⤳ x

theorem **mem_assPoints_of_forall_specializes** [F.IsQuasicoherent] {x : X} (hx : x ∈ support F) (h : ∀ y ∈ support F, y ⤳ x → y = x) : x ∈ assPoints F

theorem **closure_assPoints** [IsFiniteQC F] : closure (assPoints F) = support F

theorem **assPoints_eq_empty_iff** [F.IsQuasicoherent] : assPoints F = ∅ ↔ IsZero F

theorem **section_eq_zero_of_germ_assPoints** [F.IsQuasicoherent] (s : Γ(F, U)) (h : ∀ x ∈ assPoints F, ∀ hx : x ∈ U, F.presheaf.germ U x hx s = 0) : s = 0

theorem **isSMulRegular_sections_iff** [F.IsQuasicoherent] (hU : IsAffineOpen U) (f : Γ(X, U)) : IsSMulRegular Γ(F, U) f ↔ assPoints F ∩ U ⊆ X.basicOpen f

theorem **isSMulRegular_stalk_iff** [F.IsQuasicoherent] (f : Γ(X, U)) : (∀ (x : X) (hx : x ∈ U), IsSMulRegular (F.presheaf.stalk x :) (X.presheaf.germ U x hx f)) ↔ assPoints F ∩ U ⊆ X.basicOpen f

theorem **assPoints_tilde** {R : CommRingCat.{u}} [IsNoetherianRing R] (M : ModuleCat.{u} R) : assPoints (tilde M) = PrimeSpectrum.asIdeal ⁻¹' associatedPrimes R M

theorem **algebraMap_residueField** (x : X) : algebraMap (X.presheaf.stalk x) (X.residueField x) = (X.residue x).hom

theorem **algebraMap_sections_residueField** (x : U) : algebraMap Γ(X, U) (X.residueField (x : X)) = (X.evaluation U x x.2).hom

theorem **fibreMk_apply** (x : X) (t : (F.presheaf.stalk x :)) : fibreMk F x t = 1 ⊗ₜ t

theorem **fibreMk_surjective** (x : X) : Function.Surjective (fibreMk F x)

theorem **fibreEquivQuotient_fibreMk** (x : X) (t : (F.presheaf.stalk x :)) : fibreEquivQuotient F x (fibreMk F x t) = Submodule.Quotient.mk t

theorem **fibreMk_eq_zero_iff** (x : X) (t : (F.presheaf.stalk x :)) : fibreMk F x t = 0 ↔ t ∈ (maximalIdeal (X.presheaf.stalk x) • ⊤ : Submodule (X.presheaf.stalk x) (F.presheaf.stalk x :))

theorem **fibreEval_apply** (x : U) (s : Γ(F, U)) : fibreEval F x s = fibreMk F x (F.presheaf.germ U x x.2 s)

theorem **fibreEval_smul** (x : U) (f : Γ(X, U)) (s : Γ(F, U)) : fibreEval F x (f • s) = X.evaluation U x x.2 f • fibreEval F x s

theorem **fibreEval_res** (h : V ≤ U) (x : V) (s : Γ(F, U)) : fibreEval F x (F.presheaf.map (homOfLE h).op s) = fibreEval F (⟨x, h x.2⟩ : U) s

theorem **isBaseChange_fibreEval** [F.IsQuasicoherent] (hU : IsAffineOpen U) (x : U) : IsBaseChange (X.residueField (x : X)) (fibreEval F x)

theorem **fibreMap_fibreMk** (φ : F ⟶ G) (x : X) (t : (F.presheaf.stalk x :)) : fibreMap φ x (fibreMk F x t) = fibreMk G x (stalkLinear φ x t)

theorem **fibreMap_fibreEval** (φ : F ⟶ G) (x : U) (s : Γ(F, U)) : fibreMap φ x (fibreEval F x s) = fibreEval G x (φ.app U s)

theorem **fibreMap_id** (x : X) : fibreMap (𝟙 F) x = LinearMap.id

theorem **fibreMap_comp** (φ : F ⟶ G) (ψ : G ⟶ H) (x : X) : fibreMap (φ ≫ ψ) x = fibreMap ψ x ∘ₗ fibreMap φ x

theorem **fibreMap_add** (φ ψ : F ⟶ G) (x : X) : fibreMap (φ + ψ) x = fibreMap φ x + fibreMap ψ x

theorem **fibreMap_zero** (x : X) : fibreMap (0 : F ⟶ G) x = 0

theorem **fibreEquiv_apply** (e : F ≅ G) (x : X) (t : fibre F x) : fibreEquiv e x t = fibreMap e.hom x t

theorem **fibreMap_surjective** (φ : F ⟶ G) [Epi φ] (x : X) : Function.Surjective (fibreMap φ x)

theorem **fibreMap_exact** {S : ShortComplex X.Modules} (hS : S.Exact) [Epi S.g] (x : X) : Function.Exact (fibreMap S.f x) (fibreMap S.g x)

theorem **nontrivial_fibre_iff** [IsFiniteQC F] {x : X} : Nontrivial (fibre F x) ↔ x ∈ support F

theorem **map_fibreMk_eq_top_iff** [IsFiniteQC F] (x : X) (N : Submodule (X.presheaf.stalk x) (F.presheaf.stalk x :)) : N.map (fibreMk F x) = ⊤ ↔ N = ⊤

theorem **span_fibreMk_eq_top_iff** [IsFiniteQC F] (x : X) (s : Set (F.presheaf.stalk x :)) : Submodule.span (X.residueField x) (fibreMk F x '' s) = ⊤ ↔ Submodule.span (X.presheaf.stalk x) s = ⊤

theorem **fibreMap_surjective_iff** (φ : F ⟶ G) [IsFiniteQC G] (x : X) : Function.Surjective (fibreMap φ x) ↔ Function.Surjective (stalkLinear φ x)

theorem **epi_iff_fibreMap_surjective** (φ : F ⟶ G) [IsFiniteQC G] : Epi φ ↔ ∀ x : X, Function.Surjective (fibreMap φ x)

theorem **structureModuleFibreEquiv_fibreEval** (x : U) (s : Γ(structureModule X, U)) : structureModuleFibreEquiv (x : X) (fibreEval (structureModule X) x s) = X.evaluation U x x.2 (structureModuleSectionsEquiv U s)

theorem **baseHom_apply** (r : R) : baseHom U ρ r = X.presheaf.map (homOfLE (le_top : U ≤ ⊤)).op (ρ r)

theorem **res_baseHom** (h : V ≤ U) (r : R) : X.presheaf.map (homOfLE h).op (baseHom U ρ r) = baseHom V ρ r

theorem **baseHom_top** : baseHom ⊤ ρ = ρ

theorem **germ_baseHom** (x : X) (hx : x ∈ U) (r : R) : X.presheaf.germ U x hx (baseHom U ρ r) = X.presheaf.germ ⊤ x trivial (ρ r)

theorem **IsFlatOver.flat** (h : IsFlatOver F ρ) (hU : IsAffineOpen U) [m : Module R Γ(F, U)] (hm : ∀ (r : R) (s : Γ(F, U)), r • s = baseHom U ρ r • s) : Module.Flat R Γ(F, U)

theorem **flat_of_isLocalizedModule_of_flat** {A S M N : Type*} [CommRing A] [CommRing S] [Algebra A S] [AddCommGroup M] [Module A M] [Module S M] [IsScalarTower A S M] [AddCommGroup N] [Module A N] [Module S N] [IsScalarTower A S N] (T : Submonoid S) (f : M →ₗ[S] N) [IsLocalizedModule T f] [h : Module.Flat A M] : Module.Flat A N

theorem **flat_sections_iff_stalk** [F.IsQuasicoherent] (hU : IsAffineOpen U) : (letI

theorem **isFlatOver_iff_stalk** [F.IsQuasicoherent] : IsFlatOver F ρ ↔ ∀ x : X, letI

theorem **IsFlatOver.of_forall_exists_affine** [F.IsQuasicoherent] (h : ∀ x : X, ∃ U : X.Opens, IsAffineOpen U ∧ x ∈ U ∧ letI

theorem **isFlatOver_iff_affineCover** [F.IsQuasicoherent] {ι : Type*} (𝒰 : ι → X.Opens) (hcov : IsOpenCover 𝒰) (haff : ∀ i, IsAffineOpen (𝒰 i)) : IsFlatOver F ρ ↔ ∀ i, letI

theorem **sectionsBaseLinear_apply** (φ : F ⟶ G) (s : Γ(F, U)) : sectionsBaseLinear ρ φ U s = φ.app U s

theorem **IsFlatOver.of_sum_eq_id** {ι : Type*} [Fintype ι] {A : ι → X.Modules} (ρ : ∀ i, F ⟶ A i) (q : ∀ i, A i ⟶ F) (h : ∑ i, ρ i ≫ q i = 𝟙 F) (hA : ∀ i, IsFlatOver (A i) ρ) : IsFlatOver F ρ

theorem **IsFlatOver.of_iso** (h : IsFlatOver F ρ) (e : F ≅ G) : IsFlatOver G ρ

theorem **IsFlatOver.biprod** (hF : IsFlatOver F ρ) (hG : IsFlatOver G ρ) : IsFlatOver (F ⊞ G) ρ

theorem **IsFlatOver.isSMulRegular** (h : IsFlatOver F ρ) {r : R} (hr : r ∈ nonZeroDivisors R) (U : X.Opens) : IsSMulRegular Γ(F, U) (baseHom U ρ r)

theorem **IsFlatOver.restrict** (h : IsFlatOver F ρ) {Y : Scheme.{u}} (g : Y ⟶ X) [IsOpenImmersion g] : IsFlatOver (F.restrict g) (g.appTop.hom.comp ρ)

theorem **isFlatOver_of_field** {k : Type*} [Field k] (ρ : k →+* Γ(X, ⊤)) : IsFlatOver F ρ

theorem **isFlatOver_structureModule_iff** : IsFlatOver (structureModule X) ρ ↔ ∀ U : X.Opens, IsAffineOpen U → (baseHom U ρ).Flat

theorem **isFlatOver_structureModule_iff_flat** {R : CommRingCat.{u}} (f : X ⟶ Spec R) : IsFlatOver (structureModule X) (structureRingHom f) ↔ Flat f

theorem **baseHom_structureRingHom** {R : CommRingCat.{u}} (f : X ⟶ Spec R) (U : X.Opens) : baseHom U (structureRingHom f) = ((Scheme.ΓSpecIso R).inv ≫ f.appLE ⊤ U le_top).hom

theorem **isFlatOver_Spec_iff** {R : CommRingCat.{u}} (M : (Spec R).Modules) [M.IsQuasicoherent] : IsFlatOver M (Scheme.ΓSpecIso R).inv.hom ↔ Module.Flat R Γ(M, ⊤)

theorem **isFlatOver_tilde_iff** {R : CommRingCat.{u}} (M : ModuleCat.{u} R) : IsFlatOver (tilde M) (Scheme.ΓSpecIso R).inv.hom ↔ Module.Flat R M

theorem **structureRingHom_id** {R : CommRingCat.{u}} : structureRingHom (𝟙 (Spec R)) = (Scheme.ΓSpecIso R).inv.hom

Development.AbelianVarietyTheory.Definitions.QuasicoherentCohomology.TwistingSheafQuasicoherentLocallyFree

theorem **hom_app_eq_smul** {F : X.Modules} (φ : unitMod X ⟶ F) (V : X.Opens) (a : Γ(X, V)) : φ.app V a = a • F.presheaf.map (homOfLE (le_top : V ≤ ⊤)).op (φ.app ⊤ (1 : Γ(X, ⊤)))

theorem **exists_gen_of_iso** (hL : L.IsLB) (e : unitMod X ≅ toMod L) : ∃ t : LB.Gen L, t.IsGen

theorem **nonempty_iso_iff** (hL : L.IsLB) : Nonempty (unitMod X ≅ toMod L) ↔ ∃ t : LB.Gen L, t.IsGen

theorem **isoOfGen_idIso** (hL : L.IsLB) : (isoOfGen hL hL (idIso L) (idIso_isGen hL)).hom = 𝟙 (toMod L)

theorem **subsingleton_of_isEmpty** [IsEmpty L.ι] (V : X.Opens) : Subsingleton Γ(toMod L, V)

theorem **isEmpty_of_isLB** [IsEmpty L.ι] (hL : L.IsLB) : IsEmpty X

theorem **genericPt_mem** {f : MvPolynomial (Fin 2) R} (hf : f ≠ 0) : genericPt R ∈ Proj.basicOpen (grading R 1) f

theorem **exists_const** (i : Fin 2) (u : Γ(PSpace R 1, chart R 1 i)ˣ) : ∃ c : R, c ≠ 0 ∧ (u : Γ(PSpace R 1, chart R 1 i)) = algebraMap R _ c

theorem **not_exists_gen** : ¬ ∃ t : LB.Gen (O1 R 1), t.IsGen

theorem **twisting_sheaf_not_iso_structure_sheaf** : IsEmpty (unitMod (PSpace R 1) ≅ toMod (O1 R 1))

theorem **twisting_sheaf_not_iso_trivial_bundle** : IsEmpty (toMod (Out.Inh.trivLB (PSpace R 1)) ≅ toMod (O1 R 1))

theorem **twisting_sheaf_quasicoherent_locally_free** : (toMod (O1 R 1)).IsQuasicoherent ∧ (toMod (O1 R 1)).IsLocallyFree

theorem **ofSec_coord_eq_chartSec** (k : Fin 2) : ofSec (ProjSpace.coord R 1 k) (chart R 1 k) = chartSec (O1_isLB R 1) k

theorem **coord_independent** (a b : R) (h : algebraMap R Γ(PSpace R 1, ⊤) a • ofSec (ProjSpace.coord R 1 0) ⊤ + algebraMap R Γ(PSpace R 1, ⊤) b • ofSec (ProjSpace.coord R 1 1) ⊤ = 0) : a = 0 ∧ b = 0

theorem **exists_form** (i : Fin 2) (a : Γ(PSpace R 1, chart R 1 i)) : ∃ (n : ℕ) (f : MvPolynomial (Fin 2) R) (hf : f.IsHomogeneous n), a = formOn (chart R 1 i) i le_rfl n f hf

theorem **negOne_isLB** : (negOne R).IsLB

theorem **subsingleton_negOne** : Subsingleton Γ(toMod (negOne R), ⊤)

Development.AbelianVarietyTheory.Definitions.RegularCharts.NotFamConstZero

theorem **mem_R_of_mem_B** (P : PChart Xo c β) (e : P.Ω) {x : P.Ω} (hx : x ∈ P.B) : x ∈ P.R e

theorem **isUnit_e** (P : PChart Xo c β) {e : P.Ω} (he : e ∈ P.es) : IsUnit (⟨e, mem_R_of_mem_B P e (P.es_mem e he).2⟩ : P.R e)

theorem **isFractionRing_R** (P : PChart Xo c β) (e : P.Ω) : IsFractionRing (P.R e) P.Ω

theorem **map_B_mem** (P : PChart Xo c β) {L : Type} [Field L] [Algebra c.F L] (W : ValuationSubring L) (hO : ∀ a : O, algebraMap c.F L (β a) ∈ W) (e : P.Ω) (ψ : P.R e →ₐ[c.F] L) (hg : ∀ g : P.R e, (g : P.Ω) ∈ P.gens → ψ g ∈ W) {x : P.Ω} (hx : x ∈ P.B) : ψ ⟨x, mem_R_of_mem_B P e hx⟩ ∈ W

theorem **not_fam_zero** (P : PChart Xo c β) (ws : Fin 0 → Rul Xo c) : ¬ P.Fam ws

theorem **not_fam_const_zero** (P : PChart Xo c β) (m : ℕ) : ¬ P.Fam (fun _ : Fin m => (0 : Rul Xo c))

theorem **not_covers_empty** (P : PChart Xo c β) {J : Type} [IsEmpty J] (Q : J → PChart Xo c β) : ¬ Covers Q

theorem **nonempty_of_point** (hbc : FrBC Xo c) (y₀ : FrPt Xo c c.F) (hy : ∀ (T : CDVR) (io : c.F →+* T.Fr), letI

theorem **evR_map** (h : EvLaws Xo c) {f : Rul Xo c} (hf : IsNatural (K

theorem **evR_mem_R** (h : EvLaws Xo c) (P : PChart Xo c β) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) {e : P.Ω} (he : e ∈ P.es) : evR Xo c f P.yg ∈ P.R e

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.AbsOfRange

theorem **abs_of_range** (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT C0 A Cn C1 C : ℝ) (hn : 0 ≤ Cn) (h : ABSr D large₂ C' Cl Cg cb Cu cT CT C0 Cn C1 C) : Formalism.MainNotions.S.UnitaryVolumeMass.stABS D large₂ C' Cl Cg cb Cu cT CT C0 A C1 C ∧ Formalism.MainNotions.S.UnitaryVolumeMass.stABSconv D large₂ C' Cl Cg cb Cu cT CT C0 A C1

theorem **vand_pos** {n : ℕ} {x : Fin n → ℝ} (h : StrictAnti x) : 0 < Formalism.MainNotions.S.vand x

theorem **deg_eq** (D : D7) (h : DegVand D) (W : LocW D.R.k D.R.rS) (l) (A₁ A₂ : LocS W) (h₁ : GoodS D l A₁) (h₂ : GoodS D l A₂) : A₂.deg = A₁.deg

theorem **deg_ratio** (D : D7) (h : DegVand D) (W : LocW D.R.k D.R.rS) (l₁ l₂) (A₁ A₂ : LocS W) (h₁ : GoodS D l₁ A₁) (h₂ : GoodS D l₂ A₂) : A₂.deg / A₁.deg = Formalism.MainNotions.S.vand l₂ / Formalism.MainNotions.S.vand l₁

theorem **sizes_of** (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT C0 : ℝ) (hb : D.host.rdE ≤ large₂.sc.b) (h : HypZ D large₂ C' Cl Cg cb Cu cT CT C0) : Sizes D large₂ Cl Cg CT

theorem **theta_pos** (D : D7) (τ w w') (θ : ℝ) (W : LocW D.R.k D.R.rS) (h : GoodW D τ w w' θ W) : 0 < θ

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.ActSingleNeZero

theorem **isDer_trivial** {n : ℕ} (ε : Fin n → Bool) : IsDer ε (fun _ : Matrix (Fin n) (Fin n) ℂ => ContinuousLinearMap.id ℂ ℂ) ⊤ (fun _ _ => 0)

theorem **act_coe** {n : ℕ} (y : Matrix (Fin n) (Fin n) ℂ) (v : (⊤ : Submodule ℂ (EuclideanSpace ℂ (Fin n)))) : (act y v : EuclideanSpace ℂ (Fin n)) = stdRep y v

theorem **isDer** (n : ℕ) : IsDer (allPos n) stdRep ⊤ (fun a b => act (Matrix.single a b 1))

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.AdaptSelf

theorem **adapt_self** (N : NF H r) : Adapt N.F N.F' N.ϑ N

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.AmpKF0IffGapIn

theorem **aOf_mono** (r : ℕ) {h h' : ℝ} (hh : h ≤ h') : aOf r h ≤ aOf r h'

theorem **gapOK_mono** {k r : ℕ} {h h' : ℝ} (hh : h ≤ h') (H : GapOK k r h) : GapOK k r h'

theorem **exists_gapOK** (k r : ℕ) (hk : 1 ≤ k) : ∃ h : ℝ, GapOK k r h

theorem **exists_gp** (k r : ℕ) (hk : 1 ≤ k) (ε : Fin (2 * r + k + 1) → Bool) : ∃ mu : Fin (2 * r + k + 1) → ℝ, IsHCParam mu ∧ Gp k r ε mu

theorem **gp_gap** {k r : ℕ} {ε : Fin (2 * r + k + 1) → Bool} {mu : Fin (2 * r + k + 1) → ℝ} (H : Gp k r ε mu) : 1 ≤ k ∧ ∀ a b, ε a ≠ ε b → 120 * ((2 * r + k + 1 : ℕ) : ℝ) ^ 3 ≤ |mu a − mu b|

theorem **not_gp_small** : ¬ Gp 1 0 (fun i => decide (i = 0)) (fun i => if i = 0 then 1 / 2 else -1 / 2)

theorem **not_gp_zero** (r : ℕ) (ε : Fin (2 * r + 0 + 1) → Bool) (mu : Fin (2 * r + 0 + 1) → ℝ) : ¬ Gp 0 r ε mu

theorem **gp_of_onePlace** {k r : ℕ} {L : LocBessel k r} {h : ℝ} {μG : Measure (Mt k r)} {w0 : L.Hs} {l : L.Hs → L.Wp} {l' : L.Hs → L.Wq} (H : GlobalTheory.CellAmp.OnePlaceKF L h μG w0 l l') : Gp k r L.ε L.lam

theorem **ampKF0_iff_gapIn** : AmpKF0 ↔ GlobalTheory.GapIn.AmpKF0

theorem **ampKF0_holds** : AmpKF0

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.CellDataExists

theorem **amp_dirac** {k r : ℕ} {Wp Wq : Type} [NormedAddCommGroup Wp] [InnerProductSpace ℂ Wp] [CompleteSpace Wp] [NormedAddCommGroup Wq] [InnerProductSpace ℂ Wq] (R : BranchingSupport.Rep k r) (l : R.H → Wp) (l' : R.H → Wq) (w : R.H) (x : Mt k r) (γ : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ) (η : Wq) : amp R l l' w (Measure.dirac x) γ η = ⟪l' (R.σ (Matrix.of x) w), η⟫_ℂ • l (R.σ (γ * Matrix.of x) w)

theorem **coefF_apply** {k r : ℕ} (L : LocBessel k r) (v v' : L.Hs) (ξ : L.Wp) (ξ' η : L.Wq) (g : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ) : coefF L v v' ξ ξ' g η = (⟪ξ', η⟫_ℂ * ⟪v, L.σ g v'⟫_ℂ) • ξ

private theorem **inv_mul_conj_mul_cancel** {n : Type*} [Fintype n] [DecidableEq n] (A B C D γ a : Matrix n n ℂ) (hγ : IsUnit γ.det) (ha : IsUnit a.det) : A⁻¹ * (a * B)⁻¹ * γ * (γ⁻¹ * a * γ * C) * D = A⁻¹ * B⁻¹ * γ * C * D

theorem **_root_.GlobalTheory.CellAmp.Model.pairMaj_stabAct** {k r : ℕ} (γ a : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ) (hγ : IsUnit γ.det) (ha : IsUnit a.det) (b : ℝ) (p : _root_.Formalism.MainNotions.S.PairSp k r) : _root_.GlobalTheory.CellAmp.pairMaj γ b (_root_.Formalism.MainNotions.S.stabAct γ a p) = _root_.GlobalTheory.CellAmp.pairMaj γ b p

theorem **gap_exists** (k r : ℕ) (hk : 1 ≤ k) : ∃ h : ℝ, Formalism.FourierDecayAlt.ThreeDimData.GapOK k r h ∧ GapCell k r h

theorem **_root_.GlobalTheory.CellAmp.Model.ell_one** (k r : ℕ) (z : Fin (2 * r + k + 1) → ℂ) (hz : _root_.Formalism.QuantumDeterminants.inWb k r z) : _root_.Formalism.QuantumDeterminants.ell k r z 1 = 0

theorem **_root_.GlobalTheory.CellAmp.Model.cellData_open** {k r : ℕ} (L : _root_.Formalism.MainNotions.S.LocBessel k r) (μG : Measure (_root_.Formalism.QuantumDeterminants.Mt k r)) (g : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ) (hg : g ∈ _root_.Formalism.QuantumDeterminants.UH (_root_.Formalism.QuantumDeterminants.Hpk k r)) (hinv : g⁻¹ * g = 1) (hG : 0 < _root_.Formalism.QuantumDeterminants.Gf k r L.z L.z' g) (hm : _root_.Formalism.FourierDecayAlt.C2.IsMonomial (_root_.Formalism.QuantumDeterminants.Zhat k r L.z L.z' g)) (hR : _root_.Formalism.MainNotions.S.RStab k r g = {1}) : _root_.GlobalTheory.CellAmp.CellData L μG g (Measure.dirac (fun i j => (1 : _root_.Formalism.MainNotions.S.MatKR k r) i j)) μG (_root_.Formalism.MainNotions.S.pairMeas L)

theorem **nset_zero** (k : ℕ) : Nset k 0 = {1}

theorem **cellData_exists** {k : ℕ} (hk : 1 ≤ k) (L : LocBessel k 0) (μG : Measure (Mt k 0)) : ∃ g μR νR ν, CellData L μG g μR νR ν

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.CintTranslate

lemma **out_mk** {φ : W → ℂ} {B : AddSubgroup W} (h : ∀ w, ∀ b ∈ B, φ (w + b) = φ w) (w : W) : φ (QuotientAddGroup.mk w : W ⧸ B).out = φ w

lemma **supp_fin** {A B : AddSubgroup W} {φ : W → ℂ} (h : Fine A φ B) : (Function.support fun c : W ⧸ B => φ c.out).Finite

lemma **sum_refine** {A B C : AddSubgroup W} {φ : W → ℂ} (h : Fine A φ B) (hC : C ≤ B) (hi : C.relIndex B ≠ 0) : (∑ᶠ c : W ⧸ C, φ c.out) = (C.relIndex B : ℂ) * ∑ᶠ c : W ⧸ B, φ c.out

lemma **ratio_eq** {A B B' : AddSubgroup W} {φ : W → ℂ} (h : Fine A φ B) (h' : Fine A φ B') : (∑ᶠ c : W ⧸ B, φ c.out) / (B.relIndex A : ℂ) = (∑ᶠ c : W ⧸ B', φ c.out) / (B'.relIndex A : ℂ)

theorem **cint_eq** {A B : AddSubgroup W} {φ : W → ℂ} (h : Fine A φ B) : cint A φ = (∑ᶠ c : W ⧸ B, φ c.out) / (B.relIndex A : ℂ)

lemma **fine_ind** {A B : AddSubgroup W} (hB : B ≤ A) (hi : B.relIndex A ≠ 0) : Fine A (ind (· ∈ B)) B

theorem **cint_ind** {A B : AddSubgroup W} (hB : B ≤ A) (hi : B.relIndex A ≠ 0) : cint A (ind (· ∈ B)) = 1 / (B.relIndex A : ℂ)

theorem **cint_self** (A : AddSubgroup W) : cint A (ind (· ∈ A)) = 1

theorem **cint_even** : cint (⊤ : AddSubgroup ℤ) (ind (· ∈ AddSubgroup.zmultiples (2 : ℤ))) = 1 / 2

theorem **cint_translate** {A : AddSubgroup W} {φ : W → ℂ} (h : Nice A φ) (t : W) : cint A (fun w => φ (t + w)) = cint A φ

lemma **fine_refine** {A B C : AddSubgroup W} {φ : W → ℂ} (h : Fine A φ B) (hC : C ≤ B) (hi : C.relIndex A ≠ 0) : Fine A φ C

theorem **cint_add** {A : AddSubgroup W} {φ ψ : W → ℂ} (h : Nice A φ) (h' : Nice A ψ) : Nice A (φ + ψ) ∧ cint A (φ + ψ) = cint A φ + cint A ψ

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.CmHalfIntegralGeneratorMemAlgebraicClosure

theorem **hpk_eq** : Hpk 2 1 = (!![0,0,0,0,1; 0,1,0,0,0; 0,0,1,0,0; 0,0,0,-1,0; 1,0,0,0,0] : Matrix (Fin 5) (Fin 5) ℂ)

theorem **herm3** (v w : Fin (2 + 1) → ℂ) : herm (Hb 2) v w = star (w (0 : Fin 3)) * v (0 : Fin 3) + star (w (1 : Fin 3)) * v (1 : Fin 3) - star (w (2 : Fin 3)) * v (2 : Fin 3)

theorem **xx_eq** : xx = pEl 2 1 (fun _ => (![1,0,0] : Fin 3 → ℂ)) 0

theorem **xx_inv** : xx⁻¹ = xi

theorem **sw_inv** : sw⁻¹ = sw

theorem **sw_mem** : sw ∈ GG 2 1

theorem **xx_mem** : xx ∈ GG 2 1

theorem **herm5** (v w : Fin (2 * 1 + 2 + 1) → ℂ) : herm (Hpk 2 1) v w = star (w (0 : Fin 5)) * v (4 : Fin 5) + star (w (1 : Fin 5)) * v (1 : Fin 5) + star (w (2 : Fin 5)) * v (2 : Fin 5) - star (w (3 : Fin 5)) * v (3 : Fin 5) + star (w (4 : Fin 5)) * v (0 : Fin 5)

theorem **e0_eq** : bvec (eI 2 1 0) = (![1,0,0,0,0] : Fin 5 → ℂ)

theorem **fvec0** (z) : fvec 2 1 z 0 = (![1,0,0,0,0] : Fin 5 → ℂ)

theorem **fvec1** (z) : fvec 2 1 z 1 = z

theorem **sw_mulVec** (v : Fin (2 * 1 + 2 + 1) → ℂ) : sw *ᵥ v = (![v (4 : Fin 5), v (1 : Fin 5), v (2 : Fin 5), v (3 : Fin 5), v (0 : Fin 5)] : Fin 5 → ℂ)

theorem **zhat_eq** : Zhat 2 1 z0 z1 sw = !![1, 0; 0, -2]

theorem **dmin_one** : Dmin 2 1 z0 z1 sw (1, by omega) = 1

theorem **dmin_two** : Dmin 2 1 z0 z1 sw (Fin.last 2) = -2

theorem **dr_eq** : Dr 2 1 z0 z1 sw = 1

theorem **tz0** : Tz 2 1 z0 = 1

theorem **tz1** : Tz 2 1 z1 = 3

theorem **twoFl** : TwoFl 2 1 z0 z1 sw

theorem **xx_mulVec** (v : Fin (2 * 1 + 2 + 1) → ℂ) : xx *ᵥ v = (![v (0 : Fin 5) - v (1 : Fin 5) - v (4 : Fin 5) / 2, v (1 : Fin 5) + v (4 : Fin 5), v (2 : Fin 5), v (3 : Fin 5), v (4 : Fin 5)] : Fin 5 → ℂ)

theorem **vec_eq** : xx⁻¹ *ᵥ (sw⁻¹ *ᵥ bvec (eI 2 1 0)) = (![-1/2, -1, 0, 0, 1] : Fin 5 → ℂ)

theorem **cmHalfIntegralGenerator_mem_algebraicClosure** : (WithLp.toLp 2 (![0, -1, 0, 0, 0] : Fin 5 → ℂ) : EuclideanSpace ℂ (Fin (2 * 1 + 2 + 1))) ∈ Ppos 2 1 z0 z1 sw xx

theorem **ppos_coord** (p : EuclideanSpace ℂ (Fin (2 * 1 + 2 + 1))) (hp : p ∈ Ppos 2 1 z0 z1 sw xx) : p (0 : Fin 5) = 0 ∧ p (4 : Fin 5) = 0

theorem **proj_eq** : (Ppos 2 1 z0 z1 sw xx).starProjection (WithLp.toLp 2 (![-1/2, -1, 0, 0, 1] : Fin 5 → ℂ)) = WithLp.toLp 2 (![0, -1, 0, 0, 0] : Fin 5 → ℂ)

theorem **yv_eq** : Yv 2 1 z0 z1 sw xx = 1

theorem **one_le_Sg** (k r : ℕ) (z) (x : Mat k r) : 1 ≤ Sg k r z x

theorem **eset_nonempty** : ∃ R₀ : ℝ, (Eset 2 1 z0 z1 sw R₀ (3 / 4)).Nonempty

theorem **deOK** : DeOK 2 1 z0 z1 sw 1

theorem **uEmb_mem** (a : ℂ) : uEmb 0 2 (upper fun _ => a) ∈ GG 0 2

theorem **pOf_mem** (b₀ b₁ s : ℂ) (t₀ t₁ : ℝ) : pOf 0 2 (![fun _ => b₀, fun _ => b₁], fun _ => s, ![t₀, t₁]) ∈ GG 0 2

theorem **cont_upper** (r : ℕ) : Continuous (upper : (UT r → ℂ) → Matrix (Fin r) (Fin r) ℂ)

theorem **_root_.GlobalTheory.Amp.Model.det_upper** {r : ℕ} (x : _root_.GlobalTheory.Amp.UT r → ℂ) : (_root_.GlobalTheory.Amp.upper x).det = 1

theorem **cont_upper_inv** (r : ℕ) : Continuous fun x : UT r → ℂ => (upper x)⁻¹

theorem **_root_.GlobalTheory.Amp.Model.cont_uEmb** (k r : ℕ) : Continuous fun x : _root_.GlobalTheory.Amp.UT r → ℂ => _root_.GlobalTheory.Amp.uEmb k r (_root_.GlobalTheory.Amp.upper x)

theorem **_root_.GlobalTheory.Amp.Model.cont_pOf** (k r : ℕ) : Continuous (_root_.GlobalTheory.Amp.pOf k r)

theorem **_root_.GlobalTheory.Amp.Model.cont_lebN_map** (k r : ℕ) : Continuous fun q : (_root_.GlobalTheory.Amp.UT r → ℂ) × _root_.GlobalTheory.Amp.PPar k r => (Matrix.of.symm (_root_.GlobalTheory.Amp.uEmb k r (_root_.GlobalTheory.Amp.upper q.1) * _root_.GlobalTheory.Amp.pOf k r q.2) : _root_.Formalism.QuantumDeterminants.Mt k r)

theorem **lebN_ne_zero** (k r : ℕ) : lebN k r ≠ 0

theorem **ball_pos_fin** {m : ℕ} (S : Set (Matrix (Fin m) (Fin m) ℂ)) {ρ : ℝ} (h : 0 < ρ) : 0 < volume (Metric.ball (0 : lieE S) ρ) ∧ volume (Metric.ball (0 : lieE S) ρ) < ⊤

theorem **hasDens_unique** {m : ℕ} {S : Set (Matrix (Fin m) (Fin m) ℂ)} {μ} {c c' : ℝ≥0∞} (h : HasDens S μ c) (h' : HasDens S μ c') : c = c'

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.CmStarOfIsCMUnram

theorem **int_smul_eq** (c : 𝒪 E) (x : V) : c • x = (c : E) • x

theorem **PosOn.mono** {h : V →ₗ⋆[E] V →ₗ[E] E} {U U' : Submodule E V} (hU : PosOn h U) (hle : U' ≤ U) : PosOn h U'

theorem **orth_span** (h : V →ₗ⋆[E] V →ₗ[E] E) {M : Submodule (𝒪 E) V} {x : V} (hx : ∀ y ∈ M, h y x = 0) : ∀ y ∈ Submodule.span E (M : Set V), h y x = 0

theorem **ndLat_of_isLattice** {h : V →ₗ⋆[E] V →ₗ[E] E} (hn : h.Nondegenerate) {M : Submodule (𝒪 E) V} (hM : Submodule.IsLattice E M) : NDLat h M

theorem **mem_gL** {S : HSpace E} (Λ : Submodule (𝒪 E) S.V) (g : ℕ) (x : S.V) : x ∈ gL Λ g ↔ ∃ y ∈ Λ, (g : E) • y = x

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.CosetExpUnique

theorem **sqrt_bx** (x eps : ℝ) : Real.sqrt (bx x eps) = Real.exp (x ^ eps / 2)

theorem **Num.facts** {R : Ranks} {x eps : ℝ} {Del h : ℕ} (N : Num R x eps Del h) : Real.exp (x ^ eps / 2) ≤ bpC x eps ∧ bpC x eps ≤ aC x eps ∧ Real.exp (x ^ eps / 2) ≤ bx x eps ∧ bx x eps ≤ aC x eps ∧ 4 * aC x eps ≤ bx x eps ^ 2 ∧ 2 * (R.k : ℝ) ≤ bpC x eps / x ∧ bpC x eps / x ≤ bpC x eps ∧ (5 : ℝ) ≤ R.k ∧ 2 * (R.k : ℝ) ≤ Real.exp (x ^ eps / 2)

theorem **half_k** (R : Ranks) : ∃ z : ℤ, ((R.k : ℚ) - 1) / 2 = z + 1 / 2

theorem **place_rows** (D : D7) (N : Num D.R x eps Del h) (hDel : D.s.Del = Del) (hT : ∀ τ, Top D τ → D.hc τ = hcTop N) (hO : ∀ τ, ¬ Top D τ → D.hc τ = hcOff N) : OneScale D x eps (2 * (D.R.m : ℝ) * h) ∧ Nodes D x eps ∧ (∀ τ (j : ℕ) (hj : j + 1 < D.R.m), (((D.hc τ).b (j + 1, hj) - (D.hc τ).b (j, by omega) : ℚ) : ℝ) = 2 * h) ∧ (∀ τ (i : Fin (D.R.k - 1)), 0 ≤ (((D.hc τ).v i : ℚ) : ℝ) ∧ (((D.hc τ).v i : ℚ) : ℝ) ≤ 4 * (D.R.k : ℝ) * (D.s.Del : ℝ)) ∧ (∀ τ, ¬ Top D τ → 4 * (D.R.k : ℝ) * (D.s.Del : ℝ) ≤ uFirst D τ - vFirst D τ ∧ uFirst D τ - vFirst D τ ≤ 8 * (D.R.k : ℝ) * (D.s.Del : ℝ) ∧ uLast D τ = vLast D τ - 1 / 2)

theorem **rcd_rows** (H : Host R) (A : Arith H) (G : LocalGap H) (N : Num R x eps Del h) (Gam : ℝ) (hG : 1 < Gam) : (∀ τ, Top (rcd H A G N Gam hG) τ → (rcd H A G N Gam hG).hc τ = hcTop (R

theorem **charpoly_eq_of_isIso** (n : ℕ) : P n = 2 ^ n

theorem **natCast_two_pow** (n : ℕ) : ((2 ^ n : ℕ) : ℝ) = P n

theorem **twoPow_pos** (n : ℕ) : 0 < P n

theorem **twoPow_mul_twoPow** (a b : ℕ) : P a * P b = P (a + b)

theorem **twoPow_pow** (a n : ℕ) : P a ^ n = P (a * n)

theorem **P_le** {a b : ℕ} (h : a ≤ b) : P a ≤ P b

theorem **twoPow_rpow** (N : ℕ) (c : ℝ) : P N ^ c = (2 : ℝ) ^ ((N : ℝ) * c)

theorem **twoPow_rpow_of_mul_eq** {N n : ℕ} {c : ℝ} (h : (N : ℝ) * c = n) : P N ^ c = P n

theorem **twoPow_rpow_le** {N n : ℕ} {c : ℝ} (h : (N : ℝ) * c ≤ n) : P N ^ c ≤ P n

theorem **twoPow_le_rpow** {N n : ℕ} {c : ℝ} (h : (n : ℝ) ≤ (N : ℝ) * c) : P n ≤ P N ^ c

theorem **twoPow_le_exp** {n : ℕ} {y : ℝ} (h : (n : ℝ) ≤ y) : P n ≤ Real.exp y

theorem **log_le_of_le_P** {n : ℕ} {y : ℝ} (hy : 0 < y) (h : y ≤ P n) : Real.log y ≤ n

theorem **one_le_P** (n : ℕ) : 1 ≤ P n

theorem **L_small** : P 2800 ^ ((1 : ℝ) / 200) = 16384

theorem **num_small** : Num R0 (P 2800) (1 / 200) 1 9 where

theorem **exists_not_top** (D : D7) : ∃ τ, ¬ Top D τ

theorem **exists_top** (D : D7) (h : D.host.TT.Nonempty) : ∃ τ, Top D τ

theorem **gam_gt** {CG y : ℝ} (R : Ranks) (hC : 0 < CG) (hy : 0 < y) : 1 < Real.exp (CG * (R.m : ℝ) ^ 5 * R.f * y)

theorem **small_model** (D0 : D7) (CG : ℝ) (hC : 0 < CG) : ∃ D : D7, (∃ τ, Top D τ) ∧ (∃ τ, ¬ Top D τ) ∧ OneScale D (P 2800) (1 / 200) 36 ∧ Nodes D (P 2800) (1 / 200) ∧ Labels D (P 2800) (1 / 200) CG

theorem **big_nums** (R : Ranks) (Del h : ℕ) (hm : (R.m : ℝ) = P 119925) (hf : (R.f : ℝ) = P 489601) (hk : (R.k : ℝ) = P 1092000) (ht : R.t = 4 * R.k + 1) (hD : (Del : ℝ) = P 32000000) (hh : (h : ℝ) = P 24000000) : Num R (P 240000) (1 / 200) Del h ∧ (∀ lrd, Real.log 2 ≤ lrd → lrd ≤ P 600 → RankNums (P 240000) (1 / 200) 20 R.m R.f R.k lrd Del) ∧ P 240000 ^ ((1 : ℝ) / 200 / 100) * (R.r : ℝ) ^ 19 * (R.f : ℝ) ^ 5 * P 240000 ^ ((1 : ℝ) / 200) ≤ 2 * h

theorem **big_ranks** : ∃ (R : Ranks) (Del h : ℕ), (R.m : ℝ) = P 119925 ∧ (R.f : ℝ) = P 489601 ∧ (R.k : ℝ) = P 1092000 ∧ R.t = 4 * R.k + 1 ∧ (Del : ℝ) = P 32000000 ∧ (h : ℝ) = P 24000000

theorem **big_model** (D0 : D7) (R : Ranks) (Del h : ℕ) (hm : (R.m : ℝ) = P 119925) (hf : (R.f : ℝ) = P 489601) (hk : (R.k : ℝ) = P 1092000) (ht : R.t = 4 * R.k + 1) (hD : (Del : ℝ) = P 32000000) (hh : (h : ℝ) = P 24000000) (H : Host R) (A : Arith H) (Cg CT : ℝ) (r1 : 2 ≤ H.rdE) (r2 : Real.log H.rdE ≤ P 600) (hg : ∀ i, ((R.k : ℝ) * bx (P 240000) (1 / 200)) ^ 8 ≤ (H.g i : ℝ) ∧ (H.g i : ℝ) ≤ bx (P 240000) (1 / 200) ^ Cg) (hT : bx (P 240000) (1 / 200) ^ ((1 / 2 : ℝ) * (R.m : ℝ)) ≤ (A.T : ℝ) ∧ (A.T : ℝ) ≤ bx (P 240000) (1 / 200) ^ (CT * (R.m : ℝ))) : ∃ D : D7, D.R = R ∧ HEq D.host H ∧ Glob D (P 240000) (1 / 200) 20 Cg CT ∧ Nodes D (P 240000) (1 / 200) ∧ OneScale D (P 240000) (1 / 200) (2 * (R.m : ℝ) * h)

theorem **Glob_one_lt** {D : D7} {x eps CD Cg CT : ℝ} (G : Glob D x eps CD Cg CT) : 1 < x

theorem **OneScale_eta** {D : D7} {x eps eta : ℝ} (O : OneScale D x eps eta) : ((D.R.t : ℝ) + 3) / 2 ≤ eta + 1

theorem **thresh_of_small** {x eps y f k N S eta v d : ℝ} (hx : 1 < x) (hy : 1000 ≤ y) (X3 : y ^ 3 ≤ x ^ (eps / 3)) (X4 : x ^ (eps / 3) ≤ y ^ 4) (hL : x ^ eps = y ^ 10) (b1 : x ≤ Real.exp y) (b2 : x ^ ((11 : ℝ) / 2) ≤ Real.exp y) (f0 : 0 ≤ f) (k0 : 0 ≤ k) (N0 : 0 ≤ N) (S0 : 0 ≤ S) (e0 : 0 ≤ eta) (bf : f ≤ Real.exp y) (bk : k ≤ Real.exp y) (bN : N ≤ Real.exp y) (bS : S ≤ Real.exp y) (be : eta ≤ Real.exp y) (bv : v ≤ Real.exp (y ^ 5)) (hd : 2 * x ^ ((11 : ℝ) / 2) < d) : ThreshNums x eps f k N S eta v d

theorem **exists_datum** (eps CD CG : ℝ) (h0 : 0 < eps) (h1 : eps < 1 / 52) (hCD : 0 ≤ CD) (hCG : 0 < CG) : ∃ x0 : ℝ, ∀ x, x0 ≤ x → ∃ (D : D7) (eta : ℝ) (J : ℕ), J + 1 ≤ D.R.e ∧ D.host.TT.card = 2 ^ (D.R.e − J) ∧ Glob D x eps CD 9 1 ∧ Nodes D x eps ∧ OneScaleDatum D x eps eta J ∧ Labels D x eps CG

theorem **thresh_conseq** {D : D7} {x eps eta : ℝ} (he : 0 < eta) (T : Thresholds D x eps eta) (hD : 0 ≤ (D.s.Del : ℝ)) : (D.R.k : ℝ) * x ^ ((11 : ℝ) / 2) ≤ Real.exp (x ^ (eps / 3)) ∧ 2 * x ^ ((11 : ℝ) / 2) ≤ aC x eps ∧ 2 * bpC x eps ≤ bx x eps ∧ 4 * bpC x eps ≤ aC x eps ∧ 2 * aC x eps ≤ bx x eps ^ 2 ∧ (D.R.N D.R.m : ℝ) + 4 * (4 * (D.R.k : ℝ) * (D.s.Del : ℝ)) + 1 ≤ bpC x eps / x ∧ 0 < eps

theorem **coset_exp_unique** {R : Ranks} {H H' : Host R} {J J' : ℕ} (h : TopsCoset H J) (h' : TopsCoset H' J') : J = J'

theorem **coset_one_le** {R : Ranks} {H : Host R} {J : ℕ} (h : TopsCoset H J) : 1 ≤ J

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.DiagMemCongruencePairs

theorem **mem_spanX** (P : Matrix (Fin n) (Fin (r + 1)) E) (v : Fin n → E) : v ∈ spanX P ↔ ∃ c, eOf P *ᵥ c = v

theorem **mem_perpX** (P : Matrix (Fin n) (Fin (r + 1)) E) (v : Fin n → E) : v ∈ perpX σ J P ↔ (cT σ (eOf P) * J) *ᵥ v = 0

theorem **mem_perpX_iff_form** (P : Matrix (Fin n) (Fin (r + 1)) E) (v : Fin n → E) : v ∈ perpX σ J P ↔ ∀ a : Fin r, form σ J v (fun i => P i a.castSucc) = 0

theorem **mem_mulLat** (g : ℕ) (Λ : Submodule (𝒪 E) (Fin n → E)) (v : Fin n → E) : v ∈ mulLat g Λ ↔ ∃ w ∈ Λ, (g : E) • w = v

theorem **mulLat_le** (g : ℕ) (Λ : Submodule (𝒪 E) (Fin n → E)) : mulLat g Λ ≤ Λ

theorem **mem_latMap** (γ : Matrix (Fin n) (Fin n) E) (Λ : Submodule (𝒪 E) (Fin n → E)) (v : Fin n → E) : v ∈ latMap γ Λ ↔ ∃ w ∈ Λ, γ *ᵥ w = v

theorem **IsRed.mem_iff** {Λ : Submodule (𝒪 E) (Fin n → E)} {P : Matrix (Fin n) (Fin (r + 1)) E} {x : Lz E m} {al : Matrix (Fin m) (Fin n) E} (h : IsRed σ J Jb Λ P x al) (y : Fin m → E) : y ∈ x.1 ↔ ∃ v ∈ latIn Λ (perpX σ J P), al *ᵥ v = y

theorem **col_mem_perpX** {T : E} {P : Matrix (Fin n) (Fin (r + 1)) E} (h : IsFrame σ J T P) (b : Fin (r + 1)) : (fun i => P i b) ∈ perpX σ J P

theorem **zOf_mem_perpX** {T : E} {P : Matrix (Fin n) (Fin (r + 1)) E} (h : IsFrame σ J T P) : zOf P ∈ perpX σ J P

theorem **IsRed.congr** {T : E} {Λ : Submodule (𝒪 E) (Fin n → E)} {P : Matrix (Fin n) (Fin (r + 1)) E} {x : Lz E m} {al bl : Matrix (Fin m) (Fin n) E} (h : IsRed σ J Jb Λ P x al) (hP : IsFrame σ J T P) (hb : ∀ v ∈ perpX σ J P, bl *ᵥ v = al *ᵥ v) : IsRed σ J Jb Λ P x bl

theorem **form_mulVec** {γ : Matrix (Fin n) (Fin n) E} (h : IsUn σ J γ) (v w : Fin n → E) : form σ J (γ *ᵥ v) (γ *ᵥ w) = form σ J v w

theorem **isUnit_det_of_isUn** {γ : Matrix (Fin n) (Fin n) E} (hJ : IsUnit J.det) (h : IsUn σ J γ) : IsUnit γ.det

theorem **mulVec_mem_perpX** {γ : Matrix (Fin n) (Fin n) E} (h : IsUn σ J γ) (P : Matrix (Fin n) (Fin (r + 1)) E) (v : Fin n → E) : γ *ᵥ v ∈ perpX σ J (γ * P) ↔ v ∈ perpX σ J P

theorem **IsRed.comp** {Λ : Submodule (𝒪 E) (Fin n → E)} {P : Matrix (Fin n) (Fin (r + 1)) E} {x y : Lz E m} {al : Matrix (Fin m) (Fin n) E} (h : IsRed σ J Jb Λ P x al) (hJ : IsUnit Jb.det) {γ : Matrix (Fin m) (Fin m) E} (hγ : IsoLz σ Jb γ x y) : IsRed σ J Jb Λ P y (γ * al)

theorem **isUn_one** : IsUn σ J 1

theorem **isUn_mul** {γ δ : Matrix (Fin n) (Fin n) E} (hγ : IsUn σ J γ) (hδ : IsUn σ J δ) : IsUn σ J (γ * δ)

theorem **isUn_of_mul_eq_one** {γ δ : Matrix (Fin n) (Fin n) E} (hγ : IsUn σ J γ) (h : γ * δ = 1) : IsUn σ J δ

theorem **latMap_one** (Λ : Submodule (𝒪 E) (Fin n → E)) : latMap 1 Λ = Λ

theorem **latMap_mul** (γ δ : Matrix (Fin n) (Fin n) E) (Λ : Submodule (𝒪 E) (Fin n → E)) : latMap (γ * δ) Λ = latMap γ (latMap δ Λ)

theorem **IsoRig.mul** {γ δ : Matrix (Fin n) (Fin n) E} {c d e : Rig E n m r} (h : IsoRig σ J γ c d) (k : IsoRig σ J δ d e) : IsoRig σ J (δ * γ) c e

theorem **IsoRig.inv** (hJ : IsUnit J.det) {γ : Matrix (Fin n) (Fin n) E} {c d : Rig E n m r} (h : IsoRig σ J γ c d) : IsoRig σ J γ⁻¹ d c

theorem **IsRed.transport** (hJ : IsUnit J.det) {T : E} {Λ : Submodule (𝒪 E) (Fin n → E)} {P : Matrix (Fin n) (Fin (r + 1)) E} {x : Lz E m} {al bl : Matrix (Fin m) (Fin n) E} (h : IsRed σ J Jb Λ P x al) (hP : IsFrame σ J T P) {γ : Matrix (Fin n) (Fin n) E} (hγ : IsUn σ J γ) (hb : ∀ v ∈ perpX σ J P, bl *ᵥ (γ *ᵥ v) = al *ᵥ v) : IsRed σ J Jb (latMap γ Λ) (γ * P) x bl

theorem **mul_mem_dInv** {i j : ℕ} {a b : E} (ha : a ∈ dInv E i) (hb : b ∈ dInv E j) : a * b ∈ dInv E (i + j)

theorem **UO.entry_mem** {u : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (hu : u ∈ UO E r) (a b : Fin (r + 1)) : u a b ∈ dInv E ((b : ℕ) - a)

theorem **one_mem_UO** : (1 : Matrix (Fin (r + 1)) (Fin (r + 1)) E) ∈ UO E r

theorem **frameLat_mul_le** (P : Matrix (Fin n) (Fin (r + 1)) E) {u : Matrix (Fin (r + 1)) (Fin (r + 1)) E} (hu : u ∈ UO E r) : frameLat (P * u) ≤ frameLat P

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.ExistsHaarSplit

theorem **int_algebraMap** {a : K} (ha : a ∈ 𝒪[K]) : IsIntegral 𝒪[K] (algebraMap K B a)

theorem **mem_of_int** {a : K} (ha : IsIntegral 𝒪[K] a) : a ∈ 𝒪[K]

theorem **int_cj** (cj : B ≃ₐ[K] B) {x : B} (hx : IsIntegral 𝒪[K] x) : IsIntegral 𝒪[K] (cj x)

theorem **mem_of_int'** {a : K} (ha : IsIntegral 𝒪[K] (algebraMap K B a)) : a ∈ 𝒪[K]

theorem **cj_gen** : cj θ = algebraMap K B t - θ

theorem **delta_sq** : (θ - cj θ) ^ 2 = algebraMap K B (t ^ 2 - 4 * n)

theorem **cj_cj** (x : B) : cj (cj x) = x

theorem **sub_cj** (a b : K) : (algebraMap K B a + algebraMap K B b * θ - cj (algebraMap K B a + algebraMap K B b * θ)) * (θ - cj θ) = algebraMap K B (b * (t ^ 2 - 4 * n))

theorem **cj_fix** (x : B) : cj x = x ↔ ∃ a : K, algebraMap K B a = x

theorem **int_gen** : IsIntegral 𝒪[K] θ

theorem **int_delta** : IsIntegral 𝒪[K] (θ - cj θ)

theorem **mem_of_mul_disc** {b : K} (hb : b * (t ^ 2 - 4 * n) ∈ 𝒪[K]) : b ∈ 𝒪[K]

theorem **int_iff** (a b : K) : IsIntegral 𝒪[K] (algebraMap K B a + algebraMap K B b * θ) ↔ a ∈ 𝒪[K] ∧ b ∈ 𝒪[K]

theorem **reduced** : IsReduced B

theorem **trace_eq** [CharZero K] (hr : Module.finrank K B = 2) (x : B) : algebraMap K B (Algebra.trace K B x) = x + cj x

theorem **trace_dual** [CharZero K] (hr : Module.finrank K B = 2) (x : B) : (∀ y : B, IsIntegral 𝒪[K] y → Algebra.trace K B (x * y) ∈ 𝒪[K]) ↔ IsIntegral 𝒪[K] x

theorem **unram** (O : Subring B) (hO : ∀ x, x ∈ O ↔ IsIntegral 𝒪[K] x) (ϖ : 𝒪[K]) (hϖ : Prime ϖ) (x : B) (hx : x ∈ O) (hxx : x * x ∈ Submodule.span O {algebraMap K B ϖ}) : x ∈ Submodule.span O {algebraMap K B ϖ}

theorem **isOpen_int** : IsOpen (𝒪[K] : Set K)

theorem **isCompact_int** : IsCompact (𝒪[K] : Set K)

theorem **secondCountable** : SecondCountableTopology K

theorem **isOpen_coord** (f : B ≃ₗ[K] K × K) (S : Set B) (hS : ∀ x, x ∈ S ↔ (f x).1 ∈ 𝒪[K] ∧ (f x).2 ∈ 𝒪[K]) : IsOpen S

theorem **exists_haar** [MeasurableSpace B] [BorelSpace B] (f : B ≃ₗ[K] K × K) (S : Set B) (hS : ∀ x, x ∈ S ↔ (f x).1 ∈ 𝒪[K] ∧ (f x).2 ∈ 𝒪[K]) : ∃ μ : Measure B, μ.IsAddHaarMeasure ∧ SigmaFinite μ ∧ μ S = 1

theorem **continuous_char** {B : Type} [CommRing B] [TopologicalSpace B] [IsTopologicalAddGroup B] (ψ : AddChar B ℂ) (S : Set B) (hS : IsOpen S) (h0 : (0 : B) ∈ S) (hψ : ∀ x ∈ S, ψ x = 1) : Continuous ψ

theorem **exists_de** (l : ℕ) [Fact l.Prime] [Algebra ℚ_[l] K] (hc : Continuous (algebraMap ℚ_[l] K)) [FiniteDimensional ℚ_[l] K] : ∃ de : K, de ≠ 0 ∧ ∀ x : K, (∀ y ∈ 𝒪[K], ‖Algebra.trace ℚ_[l] K (x * y)‖ ≤ 1) ↔ de * x ∈ 𝒪[K]

theorem **char_eq_one_iff** (l : ℕ) [hl : Fact l.Prime] (x : ℚ_[l]) (z : ℂ) (h : ∃ a : ℚ, (∃ n : ℕ, a.den = l ^ n) ∧ ‖x - (a : ℚ_[l])‖ ≤ 1 ∧ z = Complex.exp (2 * Real.pi * Complex.I * (a : ℂ))) : z = 1 ↔ ‖x‖ ≤ 1

theorem **IsUnitGen.exists_coord** {cj : B ≃ₐ[K] B} {θ : B} {t n : K} (h : IsUnitGen cj θ t n) : ∃ f : B ≃ₗ[K] K × K, ∀ x, IsIntegral 𝒪[K] x ↔ (f x).1 ∈ 𝒪[K] ∧ (f x).2 ∈ 𝒪[K]

theorem **exists_splitG** (L : LocE K B) {l : ℕ} (S : SplitAt K B L.cj l) : ∃ G : SplitG L, G.e = S.e ∧ G.q = S.q

theorem **mem_R_split** [MeasurableSpace B] (L : LocE K B) (G : SplitG L) (μ : Measure B) (x : B) : x ∈ (Model.mk L (some G) μ).R ↔ (G.e x).1 ∈ 𝒪[K] ∧ ((G.e x).1 - (G.e x).2) / (G.q : K) ∈ 𝒪[K]

theorem **exists_haar_split** [MeasurableSpace B] [BorelSpace B] (L : LocE K B) (G : SplitG L) : ∃ μ : Measure B, (Model.mk L (some G) μ).HaarOK

theorem **psiE_eq_one_iff** (y : A.Ep) : A.psiE y = 1 ↔ (haveI

theorem **exists_model** {θ : A.Ep} {t n : A.Fp} (h : IsUnitGen A.cj θ t n) : ∃ (M : Model A.Fp A.Ep) (de : A.Fp), M.OK ∧ de ≠ 0 ∧ (∀ x : A.Fp, (haveI

theorem **exists_isUnitGen** [StarRing E] (hc : Formalism.BasicNotions.CMData F E) (A : LocAt F E hc.sig p) : ∃ θ t n, IsUnitGen A.cj θ t n

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.ExistsIsFolnerUWset

theorem **isUnit_det_Hpk** : IsUnit (Hpk k r).det

theorem **isUnit_det_of_mem_GG** {g : Mat k r} (hg : g ∈ GG k r) : IsUnit g.det

theorem **norm_det_of_mem_GG** {g : Mat k r} (hg : g ∈ GG k r) : ‖g.det‖ = 1

theorem **IsClosedSub.isClosedSubgroup** {S : Set (Mat k r)} (hS : IsClosedSub k r S) : Formalism.FourierDecayAlt.B2.IsClosedSubgroup S

theorem **isClosedSub_of** {S : Set (Mat k r)} (hS : Formalism.FourierDecayAlt.B2.IsClosedSubgroup S) (h : S ⊆ GG k r) : IsClosedSub k r S

theorem **isClosedSub_GG** : IsClosedSub k r (GG k r)

theorem **isClosedSub_Nset** : IsClosedSub k r (Nset k r)

theorem **isClosedSub_UWset** (z : Fin (2 * r + k + 1) → ℂ) : IsClosedSub k r (UWset k r z)

private theorem **inv_eq_Hpk_conjTranspose_of_mem_GG** {g : Mat k r} (hg : g ∈ GG k r) : g⁻¹ = Hpk k r * gᴴ * Hpk k r

theorem **isClosedSub_Sset** {z : Fin (2 * r + k + 1) → ℂ} (hz : inWb k r z) (hT : 0 < Tz k r z) : IsClosedSub k r (Sset k r z)

theorem **haar_unimod_of_biinv** {S : Set (Mat k r)} (hS : IsClosedSub k r S) {μ : Measure (Mt k r)} (h : Formalism.FourierDecayAlt.B.IsHaarOn S μ) : IsHaarOn S μ ∧ IsUnimod S μ

theorem **exists_unimod_GG** : ∃ μ : Measure (Mt k r), IsHaarOn (GG k r) μ ∧ IsUnimod (GG k r) μ

theorem **Tz_zStd** : Tz k r (zStd k r) = 1

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.ExistsIsHaarMeasureSUJ

theorem **dsClass_iff** : DSClass ↔ ∀ n, ClassU n

theorem **isParS_two** : IsParS (![1 / 2, -1 / 2] : Fin 2 → ℝ)

theorem **SUJ_eq** (ε : Fin n → Bool) : SUJ ε = Formalism.MainNotions.BridgeLocGrowth.SUJ ε

theorem **isClosed_SUJ_in_UJ** (ε : Fin n → Bool) : IsClosed (SUJ ε : Set (UJ ε))

theorem **borelSpace_SUJ** (ε : Fin n → Bool) : BorelSpace (SUJ ε)

theorem **measurableSpace_SUJ_eq** (ε : Fin n → Bool) : (inferInstance : MeasurableSpace (SUJ ε)) = Formalism.MainNotions.BridgeLocGrowth.instMeasurableSpaceSUJ ε

theorem **exists_isHaarMeasure_SUJ** (ε : Fin n → Bool) : ∃ μ : Measure (SUJ ε), μ.IsHaarMeasure

theorem **mem_UJset_allPos_iff** (g : Matrix (Fin n) (Fin n) ℂ) : g ∈ UJset (allPos n) ↔ g ∈ Kset (allPos n)

theorem **compactSpace_SUJ_allPos** : CompactSpace (SUJ (allPos n))

theorem **stdRep_irred_KS** (S : Submodule ℂ (EuclideanSpace ℂ (Fin (n + 1)))) (hS : Stable (KS (allPos (n + 1))) stdRep S) : S = ⊥ ∨ S = ⊤

theorem **isSqIntIrrS_stdFam** (lam : Fin (n + 1) → ℝ) : IsSqIntIrrS (allPos (n + 1)) ((stdFam (n + 1)).π lam)

theorem **lowCh_two_split** (x y : ℝ) : (y < x → lowCh ![true, false] ![x, y] = ![x + 1 / 2, y - 1 / 2]) ∧ (x < y → lowCh ![true, false] ![x, y] = ![x - 1 / 2, y + 1 / 2])

theorem **lowCh_two_compact** (x y : ℝ) : (y < x → lowCh (allPos 2) ![x, y] = ![x - 1 / 2, y + 1 / 2]) ∧ (x < y → lowCh (allPos 2) ![x, y] = ![x + 1 / 2, y - 1 / 2])

theorem **EwordS_const** (ε : Fin n → Bool) (c : ℂ) (p : Fin n × Fin n) (w : List (Fin n × Fin n)) : EwordS ε (p :: w) (fun _ => c) = fun _ => 0

theorem **isWtS_trivial_two** {μ : Fin 2 → ℝ} {v : ℂ} (hs : ∑ a, μ a = 0) (hv : v ≠ 0) (h : IsWtS (fun _ : Matrix (Fin 2) (Fin 2) ℂ => (1 : ℂ →L[ℂ] ℂ)) μ v) : μ = 0

theorem **isHCS_trivial_two** : IsHCS (allPos 2) ![1 / 2, -1 / 2] (fun _ : Matrix (Fin 2) (Fin 2) ℂ => (1 : ℂ →L[ℂ] ℂ))

theorem **not_isHCS_trivial_split** : ¬ IsHCS ![true, false] ![1 / 2, -1 / 2] (fun _ : Matrix (Fin 2) (Fin 2) ℂ => (1 : ℂ →L[ℂ] ℂ))

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.ExistsIsSatFlagOne

theorem **exists_smul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (Λ : Submodule (𝓞 E) (Fin n → E)) (y : Fin n → E) (hy : y ∈ Submodule.span E (Λ : Set (Fin n → E))) : ∃ d : 𝓞 E, d ≠ 0 ∧ d • y ∈ Λ

theorem **span_inf** {E : Type} [Field E] [NumberField E] {n : ℕ} (Λ : Submodule (𝓞 E) (Fin n → E)) (hΛ : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) (W : Submodule E (Fin n → E)) : Submodule.span E ((Λ ⊓ W.restrictScalars (𝓞 E) : Submodule (𝓞 E) (Fin n → E)) : Set (Fin n → E)) = W

theorem **isSatFlag_inf** (I : AutInst) (Λ : Submodule (𝓞 I.E) (Fin I.n → I.E)) (hΛ : Submodule.span I.E (Λ : Set (Fin I.n → I.E)) = ⊤) {a : ℕ} (j : Fin a → ℕ) (W : Fin (a + 1) → Submodule I.E (Fin I.n → I.E)) (h0 : W 0 = ⊥) (hm : Monotone W) (hr : ∀ i : Fin a, Module.finrank I.E (W i.succ) = Module.finrank I.E (W i.castSucc) + j i) (hi : ∀ x ∈ W (Fin.last a), ∀ y ∈ W (Fin.last a), Formalism.UG.hform (starRingEnd I.E) I.J x y = 0) : IsSatFlag I Λ j fun i => Λ ⊓ (W i).restrictScalars (𝓞 I.E)

theorem **exists_isSatFlag_one** (I : AutInst) (x : Fin I.n → I.E) (hx : x ≠ 0) (hi : Formalism.UG.hform (starRingEnd I.E) I.J x x = 0) : ∃ S, IsSatFlag I I.L (fun _ : Fin 1 => 1) S

theorem **cIw_one_one** : cIw 1 1 (fun _ : Fin 1 => 1) = Real.pi

theorem **volU_pos** (i : ℕ) : 0 < volU i

theorem **cIw_pos** (p q : ℕ) {a : ℕ} (j : Fin a → ℕ) : 0 < cIw p q j

theorem **rankFac_eq** (E : Type) [Field E] [NumberField E] [StarRing E] (M : ℕ) : rankFac E (M + 1) = Formalism.HermitianLatticesTheory.massRatioRightSide E (M + 1) * (2 * Real.pi ^ (M + 1) / (M.factorial : ℝ) * 2 ^ M) ^ Formalism.HermitianLatticesTheory.fDeg E

theorem **finrank_span_image** {E : Type} [Field E] {n : ℕ} (φ : E →+* ℂ) (s : Set (Fin n → E)) : Module.finrank ℂ (Submodule.span ℂ ((fun y : Fin n → E => fun l => φ (y l)) '' s)) = Module.finrank E (Submodule.span E s)

theorem **finrank_flagAt** (I : AutInst) {a : ℕ} (S : Fin (a + 1) → Submodule (𝓞 I.E) (Fin I.n → I.E)) (v : InfinitePlace I.F) (i : Fin (a + 1)) : Module.finrank ℂ (flagAt I S v i) = Formalism.HN.rkK (S i)

theorem **isIsoFlag_flagAt** (I : AutInst) (Λ : Submodule (𝒪 I.E) (Fin I.n → I.E)) {a : ℕ} (j : Fin a → ℕ) (S : Fin (a + 1) → Submodule (𝒪 I.E) (Fin I.n → I.E)) (h : IsSatFlag I Λ j S) (v : InfinitePlace I.F) : IsIsoFlag (I.ε v) j (flagAt I S v)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.ExistsReduction

theorem **mem_Ksub** (I : AutInst) (x : I.Gf) : x ∈ Ksub I ↔ x ∈ I.Kset

theorem **mulVec_mem** {I : AutInst} {x : I.Gf} (hx : x ∈ I.Kset) {y : Fin I.n → I.E ⊗[I.F] Af I.F} (hy : y ∈ I.Lf) : I.fmat x *ᵥ y ∈ I.Lf

theorem **fmat_mul** (I : AutInst) (x y : I.Gf) : I.fmat (x * y) = I.fmat x * I.fmat y

theorem **fmat_one** (I : AutInst) : I.fmat 1 = 1

theorem **map_one** : ρ.toFun 1 = 1

theorem **map_inv** {x : I.Gf} (hx : x ∈ I.Kset) : ρ.toFun x⁻¹ = (ρ.toFun x)⁻¹

theorem **lift_of_mem** (μf : Measure I.Gf) (a : MonoidAlgebra ℂ Q) {x : I.Gf} (hx : x ∈ I.Kset) : ρ.lift μf a x = a.coeff (ρ.toFun x) / ((μf (I.Kg g)).toReal : ℂ)

theorem **lift_of_notMem** (μf : Measure I.Gf) (a : MonoidAlgebra ℂ Q) {x : I.Gf} (hx : x ∉ I.Kset) : ρ.lift μf a x = 0

theorem **lift_one** (μf : Measure I.Gf) (x : I.Gf) : ρ.lift μf 1 x = (I.Kg g).indicator (fun _ => (((μf (I.Kg g)).toReal : ℂ))⁻¹) x

theorem **lift_inv** (μf : Measure I.Gf) (a : MonoidAlgebra ℂ Q) (ha : IsSA a) (x : I.Gf) : ρ.lift μf a x⁻¹ = star (ρ.lift μf a x)

theorem **tensAlg_coeff** {ι : Type} [Fintype ι] {G : ι → Type} [∀ i, Group (G i)] [∀ i, Finite (G i)] (a : ∀ i, MonoidAlgebra ℂ (G i)) (h : ∀ i, G i) : (tensAlg a).coeff h = ∏ i, (a i).coeff (h i)

theorem **levelafC_nat** (N : ℕ) : levelafC (F

theorem **exists_reduction** (p : ℕ) (hp : p.Prime) (w : HeightOneSpectrum (𝒪 E)) (hw : Ideal.absNorm w.asIdeal = p) : ∃ r : IntAf F E →+* ZMod p, ∀ a, r (psi F E a) = 0 ↔ a ∈ w.asIdeal

theorem **exists_approx** {p : ℕ} (hp : p ≠ 0) (s : IntAf F E) : ∃ (b : 𝒪 E) (t : IntAf F E), s = psi F E b + (p : IntAf F E) * t

theorem **ker_eq_extId** {I : AutInst} {p : ℕ} (hp : p ≠ 0) {w : HeightOneSpectrum (𝒪 I.E)} (r : IntAf I.F I.E →+* ZMod p) (hr : ∀ a, r (psi I.F I.E a) = 0 ↔ a ∈ w.asIdeal) : RingHom.ker r = extId I w.asIdeal

theorem **exists_matrix** {R k L : Type} [CommRing R] [CommRing k] [AddCommGroup L] [Module R L] {N : ℕ} (r : R →+* k) (hr : Function.Surjective r) (c : L →ₗ[r] (Fin N → k)) (hc : Function.Surjective c) (a : Ideal R) (hk : ∀ y, c y = 0 ↔ y ∈ a • (⊤ : Submodule R L)) (T : L →ₗ[R] L) : ∃ M : Matrix (Fin N) (Fin N) k, ∀ y, c (T y) = M *ᵥ c y

theorem **r_psi** (a : 𝒪 I.E) : fr.r (psi I.F I.E a) = 0 ↔ a ∈ w.asIdeal

theorem **absNorm_eq** : Ideal.absNorm w.asIdeal = p

theorem **prime** : p.Prime

theorem **ker_r** : RingHom.ker fr.r = extId I w.asIdeal

theorem **c_eq_zero_iff** (y : I.Lf) : fr.c y = 0 ↔ y ∈ extId I w.asIdeal • (⊤ : Submodule (IntAf I.F I.E) I.Lf)

theorem **card_quot** : Nat.card (I.Lf ⧸ (extId I w.asIdeal • (⊤ : Submodule (IntAf I.F I.E) I.Lf))) = p ^ N

theorem **rank_unique** {N' : ℕ} (fr' : PlFrame I N' p w) : N = N'

theorem **ActsAs.unique** {x : I.Gf} (hx : x ∈ I.Kset) {M M' : Matrix (Fin N) (Fin N) (ZMod p)} (h : fr.ActsAs x M) (h' : fr.ActsAs x M') : M = M'

theorem **ActsAs.mul** {x x' : I.Gf} (hx' : x' ∈ I.Kset) {M M' : Matrix (Fin N) (Fin N) (ZMod p)} (h : fr.ActsAs x M) (h' : fr.ActsAs x' M') : fr.ActsAs (x * x') (M * M')

theorem **actsAs_one** : fr.ActsAs 1 1

theorem **exists_actsAs** {x : I.Gf} (hx : x ∈ I.Kset) : ∃ M, fr.ActsAs x M

theorem **exists_act** : ∃ ρ : I.Gf → GL (Fin N) (ZMod p), (∀ x ∈ I.Kset, fr.ActsAs x (ρ x : Matrix (Fin N) (Fin N) (ZMod p))) ∧ ∀ x ∈ I.Kset, ∀ x' ∈ I.Kset, ρ (x * x') = ρ x * ρ x'

theorem **exists_actsBy** : ∃ ρ : I.Gf → GProd N g Pl, ActsBy fr ρ ∧ ∀ x ∈ I.Kset, ∀ x' ∈ I.Kset, ρ (x * x') = ρ x * ρ x'

theorem **ActsBy.unique** {ρ ρ' : I.Gf → GProd N g Pl} (h : ActsBy fr ρ) (h' : ActsBy fr ρ') {x : I.Gf} (hx : x ∈ I.Kset) : ρ x = ρ' x

theorem **exists_frame** (I : AutInst) (hI : ∀ y, y ∈ I.Lf ↔ ∀ i, y i ∈ IntAf I.F I.E) (p : ℕ) (hp : p.Prime) (w : HeightOneSpectrum (𝒪 I.E)) (hw : Ideal.absNorm w.asIdeal = p) : ∃ fr : PlFrame I I.n p w, ∀ (y : I.Lf) (i : Fin I.n), fr.c y i = fr.r (y.1 i, (hI y.1).1 y.2 i)

theorem **rank_eq_of_std** {I : AutInst} (hI : ∀ y, y ∈ I.Lf ↔ ∀ i, y i ∈ IntAf I.F I.E) {N p : ℕ} {w : HeightOneSpectrum (𝒪 I.E)} (fr : PlFrame I N p w) : N = I.n

theorem **exists_choice** (I : AutInst) (g : Fin 2 → ℕ) (hg : ∀ k, (g k).Prime) (hne : g 0 ≠ g 1) (hsp : ∀ k (v : HeightOneSpectrum (𝒪 I.E)), ((g k : ℕ) : 𝒪 I.E) ∈ v.asIdeal → Ideal.absNorm v.asIdeal = g k ∧ ((g k : ℕ) : 𝒪 I.E) ∉ v.asIdeal ^ 2 ∧ ¬ IsConj I v v) : ∃ (Pl : Fin 2 → Type) (w : ∀ k, Pl k → HeightOneSpectrum (𝒪 I.E)), (∀ k, Nonempty (Pl k)) ∧ IsChoice I g Pl w

theorem **fmat_negOne** (I : AutInst) : I.fmat (negOne I) = -1

theorem **negOne_mem** (I : AutInst) : negOne I ∈ I.Kset

theorem **actsAs_negOne** {I : AutInst} {N p : ℕ} {w : HeightOneSpectrum (𝒪 I.E)} (fr : PlFrame I N p w) : fr.ActsAs (negOne I) (-1)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.FlagPairDefinitePartLattice

theorem **_root_.GlobalTheory.HermLat.mem_perp_iff** (H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V) (S : Submodule K V) (y : V) : y ∈ _root_.GlobalTheory.HermLat.perp H S ↔ ∀ x ∈ S, H.B y x = 0

theorem **_root_.GlobalTheory.HermLat.mem_dual_iff** (H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V) (M : Submodule R V) (y : V) : y ∈ _root_.GlobalTheory.HermLat.dual H M ↔ y ∈ Submodule.span K (M : Set V) ∧ ∀ x ∈ M, H.B y x ∈ (1 : Submodule R K)

theorem **_root_.GlobalTheory.HermLat.isIntegralLat_iff** (H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V) (M : Submodule R V) : _root_.GlobalTheory.HermLat.IsIntegralLat H M ↔ ∀ x ∈ M, ∀ y ∈ M, H.B x y ∈ (1 : Submodule R K)

theorem **_root_.GlobalTheory.HermLat.flip_isSymm** (H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V) : H.B.flip.IsSymm

theorem **_root_.GlobalTheory.HermLat.isLatMod_top_bot_iff** (M : Submodule R V) : _root_.GlobalTheory.HermLat.IsLatMod M (⊤ : Submodule K V) ⊥ ↔ Submodule.IsLattice K M

theorem **_root_.GlobalTheory.HermLat.isUnimod_iff** (H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V) (M : Submodule R V) : _root_.GlobalTheory.HermLat.IsUnimod H M ↔ Submodule.IsLattice K M ∧ _root_.Formalism.HermitianLatticesTheory.IsUnimodular H.B.flip M

theorem **_root_.GlobalTheory.HermLat.mem_isomGrp** (H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V) (s : V ≃ₗ[K] V) : s ∈ _root_.GlobalTheory.HermLat.isomGrp H ↔ ∀ x y, H.B (s x) (s y) = H.B x y

theorem **isometric_equivalence** (H : HermSpace K V) : Equivalence (Isometric (R

theorem **autCard_eq_of_isometric** (H : HermSpace K V) {M M' : Submodule R V} (h : Isometric H M M') : autCard H M = autCard H M'

theorem **genusSum_eq_sum** (H : HermSpace K V) {L : Submodule (𝒪 K) V} {s : Finset (Submodule (𝒪 K) V)} (hs : Formalism.ThetaLiftIndependence.IsClassReps H.B.flip L s) {φ : Submodule (𝒪 K) V → ENNReal} (hφ : ∀ M M', InGenus H L M → Isometric H M M' → φ M = φ M') : genusSum H L φ = ∑ M ∈ s, φ M / autCard H M

theorem **_root_.GlobalTheory.HermLat.autCard_eq_natCard** {H : _root_.Formalism.HermitianLatticeCounting.S6.HermSpace K V} (hH : _root_.GlobalTheory.HermLat.IsTraceDef H) {M : Submodule (𝒪 K) V} (hM : Submodule.IsLattice K M) : (_root_.Formalism.ThetaLiftIndependence.autL H.B.flip M).Finite ∧ 0 < Nat.card (_root_.Formalism.ThetaLiftIndependence.autL H.B.flip M) ∧ _root_.GlobalTheory.HermLat.autCard H M = (Nat.card (_root_.Formalism.ThetaLiftIndependence.autL H.B.flip M) : ENNReal)

theorem **genusMass_eq** {H : HermSpace K V} (hH : IsTraceDef H) {L : Submodule (𝒪 K) V} (hL : Submodule.IsLattice K L) : genusMass H L = ENNReal.ofReal (Formalism.HermitianLatticesTheory.mass H.B.flip L)

theorem **genusMass_ne_zero** {H : HermSpace K V} (hH : IsTraceDef H) {L : Submodule (𝒪 K) V} (hL : Submodule.IsLattice K L) : genusMass H L ≠ 0

theorem **genusMass_ne_top** {H : HermSpace K V} (hH : IsTraceDef H) {L : Submodule (𝒪 K) V} (hL : Submodule.IsLattice K L) : genusMass H L ≠ ⊤

theorem **genusMean_one** {H : HermSpace K V} (hH : IsTraceDef H) {L : Submodule (𝒪 K) V} (hL : Submodule.IsLattice K L) : genusMean H L (fun _ => 1) = 1

theorem **ratLine_isTraceDef** : IsTraceDef ratLine

theorem **ratLine_isLattice** : Submodule.IsLattice ℚ (1 : Submodule (NumberField.RingOfIntegers ℚ) ℚ)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.FractionFieldToGenericStalkAlgebraMapEq

theorem **exists_modFam** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] (hc : Formalism.BasicNotions.CMData F E) {g₁ g₂ : ℕ} (h₁ : g₁.Prime) (h₂ : g₂.Prime) (hne : g₁ ≠ g₂) (A : ∀ p : HeightOneSpectrum (𝒪 F), LocAt F E hc.sig p) (hA : ∀ p, (A p).split.isSome ↔ (resChar p = g₁ ∨ resChar p = g₂)) : ∃ X : ModFam F E hc.sig g₁ g₂, X.A = A

theorem **nonempty_modFam** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] (hc : Formalism.BasicNotions.CMData F E) {g₁ g₂ : ℕ} (h₁ : g₁.Prime) (h₂ : g₂.Prime) (hne : g₁ ≠ g₂) (s₁ : ((Ideal.span {(g₁ : ℤ)}).primesOver (𝒪 E)).ncard = Module.finrank ℚ E) (s₂ : ((Ideal.span {(g₂ : ℤ)}).primesOver (𝒪 E)).ncard = Module.finrank ℚ E) : Nonempty (ModFam F E hc.sig g₁ g₂)

theorem **exists_field_modFam** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : StarRing E) (_ : Algebra F E) (hc : Formalism.BasicNotions.CMData F E) (g₁ g₂ : ℕ), Nonempty (ModFam F E hc.sig g₁ g₂)

theorem **lastOne_iff** {r : ℕ} (w : Pt E (r + 1)) : LastOne w ↔ ∃ l t, w.blocks = l ++ [(t, 1)]

theorem **KtopLoc_last_one** (M : Model F E) (τ : E) {r : ℕ} {w w' : Pt E (r + 1)} {h h' : Pt E r} {t t' : Eˣ} (hw : w.blocks = h.blocks ++ [(t, 1)]) (hw' : w'.blocks = h'.blocks ++ [(t', 1)]) : KtopLoc M τ w w' = if t = t' then M.CuspSpanExtension ((t : E) - τ) h h' else 0

theorem **KtopLoc_mixed** (M : Model F E) (τ : E) {r : ℕ} {w w' : Pt E (r + 1)} (h : ¬ (LastOne w ↔ LastOne w')) : KtopLoc M τ w w' = 0

theorem **KtopLoc_not_last_one** (M : Model F E) (τ : E) {r : ℕ} {w w' : Pt E (r + 1)} (h : ¬ LastOne w) (h' : ¬ LastOne w') : KtopLoc M τ w w' = dLoc M τ w * M.Kk (r + 1) w w' * conj (dLoc M τ w')

theorem **exists_unitAt_eq_inv** (p : HeightOneSpectrum (𝒪 F)) : ∃ m : ℕ, ¬ resChar p ∣ m ∧ uP (F

theorem **cP_ne_zero** (p : HeightOneSpectrum (𝒪 F)) : X.cP p ≠ 0

theorem **tmod_eq** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (t : E) : X.tmod ν s p t = X.tfac ν s p * (X.A p).ιE t

theorem **xmod_eq** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (x : E) : X.xmod ν s p x = X.xmod ν s p 1 * (X.A p).ιE x

theorem **tfac_mul** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) : X.tfac ν s p * X.xmod ν s p 1 = 1

theorem **tmod_mul_xmod** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) {t : E} (ht : t ≠ 0) : X.tmod ν s p t * X.xmod ν s p t⁻¹ = 1

theorem **isUnit_tmod** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (t : Eˣ) : IsUnit (X.tmod ν s p t)

theorem **tmod_add** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (t t' : E) : X.tmod ν s p (t + t') = X.tmod ν s p t + X.tmod ν s p t'

theorem **tmod_sub** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (t t' : E) : X.tmod ν s p (t - t') = X.tmod ν s p t - X.tmod ν s p t'

@[simp] theorem **tmod_zero** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) : X.tmod ν s p 0 = 0

theorem **xmod_sub** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (x x' : E) : X.xmod ν s p (x - x') = X.xmod ν s p x - X.xmod ν s p x'

theorem **tmod_injective** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) : Function.Injective (X.tmod ν s p)

@[simp] theorem **coe_tmodU** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (t : Eˣ) : ((X.tmodU ν s p t : (X.A p).Epˣ) : (X.A p).Ep) = X.tmod ν s p t

theorem **coe_tmodU_inv** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) (t : Eˣ) : (((X.tmodU ν s p t)⁻¹ : (X.A p).Epˣ) : (X.A p).Ep) = X.xmod ν s p (t : E)⁻¹

theorem **tmodU_injective** (ν : ℤ) (s : ℕ) (p : HeightOneSpectrum (𝒪 F)) : Function.Injective (X.tmodU ν s p)

lemma **blocksMod_size** (r : ℕ) (p : HeightOneSpectrum (𝒪 F)) (l : List (Eˣ × ℕ+)) : (X.blocksMod r p l).map (fun b => (b.2 : ℕ)) = l.map (fun b => (b.2 : ℕ))

theorem **blocksMod_concat** (r : ℕ) (p : HeightOneSpectrum (𝒪 F)) (l : List (Eˣ × ℕ+)) (b : Eˣ × ℕ+) : X.blocksMod r p (l ++ [b]) = X.blocksMod r p l ++ [(X.tmodU (nu r (l.map fun c => (c.2 : ℕ)).sum b.2) b.2 p b.1, b.2)]

theorem **ptMod_snoc** (r : ℕ) (p : HeightOneSpectrum (𝒪 F)) {m : ℕ} (h : Pt E m) (t : Eˣ) (s : ℕ+) : X.ptMod r p (h.snoc t s) = (X.ptMod r p h).snoc (X.tmodU (nu r m s) s p t) s

theorem **lastOne_ptMod** (r : ℕ) (p : HeightOneSpectrum (𝒪 F)) {n : ℕ} (w : Pt E (n + 1)) : LastOne (X.ptMod r p w) ↔ LastOne w

@[simp] theorem **coe_LaSub** (ν : ℤ) (s : ℕ) : (X.LaSub ν s : Set E) = X.La ν s

theorem **inv_sub_inv_mem_La'** (ν : ℤ) (s : ℕ) (t t' : Eˣ) : (t : E)⁻¹ - (t' : E)⁻¹ ∈ X.La' ν s ↔ ∀ p, (((X.tmodU ν s p t)⁻¹ : (X.A p).Epˣ) : (X.A p).Ep) - (((X.tmodU ν s p t')⁻¹ : (X.A p).Epˣ) : (X.A p).Ep) ∈ (X.M p).R

theorem **nonempty_place** : Nonempty (HeightOneSpectrum (𝒪 F))

theorem **Ktop_mixed** (r : ℕ) (T T' : F) {w w' : Pt E (r + 1)} (h : ¬ (LastOne w ↔ LastOne w')) : X.Ktop r T T' w w' = 0

theorem **KtopLoc_ptMod_last_one** (r : ℕ) (p : HeightOneSpectrum (𝒪 F)) (τ : (X.A p).Ep) {w w' : Pt E (r + 1)} {h h' : Pt E r} {t t' : Eˣ} (hw : w.blocks = h.blocks ++ [(t, 1)]) (hw' : w'.blocks = h'.blocks ++ [(t', 1)]) : KtopLoc (X.M p) τ (X.ptMod r p w) (X.ptMod r p w') = if t = t' then (X.M p).CuspSpanExtension (X.tmod (nu r r 1) 1 p t - τ) (X.ptMod r p h) (X.ptMod r p h') else 0

theorem **Ktop_of_ne** (r : ℕ) (T T' : F) {w w' : Pt E (r + 1)} {h h' : Pt E r} {t t' : Eˣ} (hw : w.blocks = h.blocks ++ [(t, 1)]) (hw' : w'.blocks = h'.blocks ++ [(t', 1)]) (hne : t ≠ t') : X.Ktop r T T' w w' = 0

theorem **Ktop_eq_finprod** (r : ℕ) (T T' : F) (w w' : Pt E (r + 1)) (h : (Function.mulSupport fun p => KtopLoc (X.M p) (tau0 (X.M p) ((X.A p).ιF T) ((X.A p).ιF T')) (X.ptMod r p w) (X.ptMod r p w')).Finite) : X.Ktop r T T' w w' = ∏ᶠ p, KtopLoc (X.M p) (tau0 (X.M p) ((X.A p).ιF T) ((X.A p).ιF T')) (X.ptMod r p w) (X.ptMod r p w')

@[simp] theorem **Nt_zero** (ν : ℤ) (s : ℕ) : X.Nt ν s 0 = 0

theorem **Nt_of_ne** (ν : ℤ) (s : ℕ) {t : E} (ht : t ≠ 0) : X.Nt ν s t = ∏ᶠ p, ((X.M p).modK (X.tmod ν s p t))⁻¹

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.GaugeUnfold

theorem **gauss_pos** {n : ℕ} (y : Fin n → ℂ) : 0 < gauss y

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.GaussDensDefinite

theorem **lamX_one** (I : AutInst) : lamX I (1, 1) = I.L

theorem **gaussDens_definite** (n : ℕ) : gaussDens (fun _ : Fin n => true) (sgn ℝ false) = 0

theorem **disc_integral** {t : ℝ} (ht : 0 < t) : ∫ y in {y : Fin 1 → ℂ | qEps (fun _ => true) y ≤ t}, gauss y = Real.pi * (1 - Real.exp (-t))

theorem **gaussDens_circle** : gaussDens (fun _ : Fin 1 => true) (sgn ℝ true) = Real.pi * Real.exp (-1)

theorem **isPerpInst_ofSigns** (cm : Formalism.BasicNotions.CMData F E) {n : ℕ} (s : Bool) (ε : Fin n → Bool) : IsPerpInst (ofSigns cm (Fin.cons s ε : Fin (n + 1) → Bool)) (ofSigns cm ε) (RingEquiv.refl E) s (Pi.single (0 : Fin (n + 1)) (1 : E)) (Matrix.of fun (i : Fin (n + 1)) (j : Fin n) => if i = j.succ then (1 : E) else 0) 1

theorem **exists_isPerpInst** (n : ℕ) (s : Bool) : ∃ (I I' : AutInst) (e : I'.E ≃+* I.E) (x : Fin I.n → I.E) (A : Matrix (Fin I.n) (Fin I'.n) I.E) (xf : I.Gf), I.n = n + 1 ∧ IsPerpInst I I' e s x A xf

theorem **gaussSum_ne_zero** {I I' : AutInst} {e : I'.E ≃+* I.E} {s : Bool} {x : Fin I.n → I.E} {A : Matrix (Fin I.n) (Fin I'.n) I.E} {xf : I.Gf} (h : IsPerpInst I I' e s x A xf) : gaussSum I (sgn I.E s) (1, xf) ≠ 0

theorem **const_agree** (E : Type) [Field E] [NumberField E] [StarRing E] (M : ℕ) (hv : 0 < Formalism.HermitianLatticesTheory.vInv E) (hL : Formalism.HermitianLatticesTheory.lValEta E M ≠ 0) : meanConst E M * Formalism.HermitianLatticesTheory.massRatioRightSide E M * (2 ^ M * Real.pi ^ M / ((M - 1).factorial : ℝ)) ^ Formalism.HermitianLatticesTheory.fDeg E = 2 ^ (M * Formalism.HermitianLatticesTheory.fDeg E)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.GlVolEqTsumSlabVol

theorem **star_mem_int** {x : E} (hx : x ∈ (algebraMap (𝒪 E) E).range) : star x ∈ (algebraMap (𝒪 E) E).range

theorem **dual_intLat_one** (u : Fin 1 → E) : (∀ t ∈ intLat E 1, star t ⬝ᵥ u ∈ (algebraMap (𝒪 E) E).range) ↔ u 0 ∈ (algebraMap (𝒪 E) E).range

theorem **colM_gram** (x y : Fin 3 → E) : (colM x)ᴴ * signMat E ![true, false, true] * colM y = Matrix.of fun _ _ => star (x 0) * y 0 - star (x 1) * y 1 + star (x 2) * y 2

theorem **colM_mulVec** (x : Fin 3 → E) (u : Fin 1 → E) : colM x *ᵥ u = fun a => x a * u 0

theorem **isAdapted_rank_three** (cm : Formalism.BasicNotions.CMData F E) : IsAdapted (ofSigns cm ![true, false, true]) (ofSigns cm ![true]) (RingEquiv.refl E) 1 (a

theorem **exists_isAdapted** : ∃ (I I₀ : AutInst) (e : I₀.E ≃+* I.E) (xf : I.Gf) (S : Fin 2 → Submodule (𝒪 I.E) (Fin I.n → I.E)) (Λ : ∀ _ : Fin 1, Submodule (𝒪 I.E) (Fin 1 → I.E)) (x₀ : I₀.Gf) (T T' : ∀ _ : Fin 1, Matrix (Fin I.n) (Fin 1) I.E) (T₀ : Matrix (Fin I.n) (Fin I₀.n) I.E), I.n = 3 ∧ I₀.n = 1 ∧ IsAdapted I I₀ e xf (a

theorem **IsAdapted_flag** {I I₀ : AutInst} {e : I₀.E ≃+* I.E} {xf : I.Gf} {a : ℕ} {j : Fin a → ℕ} {S : Fin (a + 1) → Submodule (𝒪 I.E) (Fin I.n → I.E)} {Λ : ∀ i, Submodule (𝒪 I.E) (Fin (j i) → I.E)} {x₀ : I₀.Gf} {T T' : ∀ i, Matrix (Fin I.n) (Fin (j i)) I.E} {T₀ : Matrix (Fin I.n) (Fin I₀.n) I.E} (h : IsAdapted I I₀ e xf j S Λ x₀ T T' T₀) : S 0 = ⊥ ∧ Monotone S ∧ ∀ k, S k ≤ lamX I (1, xf)

theorem **one_mem_leviGroup** (I : AutInst) (xf : I.Gf) {a : ℕ} (S : Fin (a + 1) → Submodule (𝒪 I.E) (Fin I.n → I.E)) : (1 : I.Ginf) ∈ leviGroup I xf S

theorem **glVol_eq_tsum_slabVol** (E : Type) [Field E] [NumberField E] (j : ℕ) : glVol E j = ∑' c : Quot (fun Λ Λ' : {Λ : Submodule (𝒪 E) (Fin j → E) // Submodule.IsLattice E Λ} => ∃ γ : GL (Fin j) E, ∀ y, (γ : Matrix (Fin j) (Fin j) E) *ᵥ y ∈ Λ'.1 ↔ y ∈ Λ.1), slabVol E j (Quot.out c).1

theorem **mem_dualAt** (E : Type) [Field E] [NumberField E] (j : ℕ) (L : Submodule ℤ (VecInf E j)) (y : VecInf E j) : y ∈ dualAt E j L ↔ ∀ x ∈ L, ∃ z : ℤ, ∑ w, (x w ⬝ᵥ y w).re = z

theorem **_root_.GlobalTheory.FlagBase.Test.latAt_aut_mul** (E : Type) [Field E] [NumberField E] (j : ℕ) (Λ : Submodule (𝒪 E) (Fin j → E)) {γ : _root_.GlobalTheory.FlagMeas.MatInf E j} (hγ : γ ∈ _root_.GlobalTheory.FlagMeas.autGL E j Λ) (g : _root_.GlobalTheory.FlagMeas.MatInf E j) : _root_.GlobalTheory.FlagBase.latAt E j Λ (γ * g) = _root_.GlobalTheory.FlagBase.latAt E j Λ g

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.GlobalRowsOfSizeParameters

theorem **scale_pos** (D : D7) (u : D.what₂.Word) : 0 < (D.mem u).A.GamP ^ (-2 : ℤ) * D.what₂.Dstar

theorem **duWin_of_data** {D : D7} {C0 : ℝ} (h : StandingNumericalData D C0) : DuWin D C0

theorem **duWin_zero_iff** (D : D7) : DuWin D 0 ↔ ∀ u, D.duS u = 0

theorem **duWin_forces** {D : D7} {C0 : ℝ} (h : DuWin D C0) (h0 : C0 ≠ 0) : 1 ≤ C0 ∧ ∀ u, D.duS u ≠ 0

theorem **duWin_three** (D : D7) (hr : ∀ u, (D.duS u : ℝ) = ((D.mem u).A.duStar : ℝ) * (((D.mem u).A.T : ℝ) ^ D.R.k - 1) ^ D.R.f) : DuWin D 3

theorem **recd_duS** (N u) : (recd N).duS u = N + 1

theorem **succ_pred_pow** (T k f : ℕ) (hT : 2 ≤ T) (hk : 0 < k) : ∃ N : ℕ, ((N + 1 : ℕ) : ℝ) = ((1 : ℕ) : ℝ) * ((T : ℝ) ^ k - 1) ^ f

theorem **duWin_nontrivial** : (∃ D : D7, DuWin D 3 ∧ ∀ u, D.duS u ≠ 0) ∧ ∃ D : D7, ¬ DuWin D 3 ∧ ∀ u, D.duS u ≠ 0

theorem **two_le_rd** {R : Ranks} (H : Host R) : 2 ≤ H.rdE

theorem **logrd_le** {D : D7} (L : SizeData₂ D) (SzN : SizeN L) (SzQ : SizeQD L) (h2 : 2 ≤ XX L) (h3 : 5 / szTh L ≤ XX L) : Real.log D.host.rdE ≤ L.sc.x ^ (L.sc.eps / 2)

theorem **globalRows_of_sizeData** {D : D7} (L : SizeData₂ D) (P : ParameterChoice' D L) (Sz : SizesAt L) (h2 : 2 ≤ XX L) (h3 : 5 / szTh L ≤ XX L) (CD : ℝ) (h0 : 0 < CD) (hC : CD ≤ P.CCount) : Glob0 D L.sc.x L.sc.eps CD (P.CYL + 1)

theorem **exists_glob0_of_zeroFreeRegion** (p1 : GlobalTheory.Mem.An.ExplZeta) (p2 : GlobalTheory.Mem.An.ExplChi) (p3 : GlobalTheory.Mem.An.LogDerOne) (p4 : GlobalTheory.Mem.An.ZeroCount) (p5 : Formalism.ZeroFreeDirichlet.ZFRDirichlet) (p6 : GlobalTheory.Mem.An.TZRat) (p7 : GlobalTheory.Mem.An.PageThm) (p8 : GlobalTheory.Mem.An.Jutila) (p9 : GlobalTheory.Mem.An.QuadraticHeckeZeroEstimates) (hR : GlobalTheory.NT.RayField) (eps CD x : ℝ) (he : 0 < eps) (he52 : eps < 1 / 52) (hCD : 0 < CD) : ∃ (y : ℝ) (D : D7), x ≤ y ∧ Glob0 D y eps CD 10 ∧ (∃ τ, Top D τ) ∧ ∃ τ, ¬ Top D τ

theorem **one_lt_of_global_data** {D : D7} {x eps CD Cg : ℝ} (G : Glob0 D x eps CD Cg) : 1 < x

theorem **not_glob0_small** (x eps CD Cg : ℝ) : ¬ Glob0 (recd 0) x eps CD Cg

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.HermitianFormUnitaryOneOne

theorem **isOpIntG_iff_isOpOf** {G H : Type} [Group G] [MeasurableSpace G] [NormedAddCommGroup H] [InnerProductSpace ℂ H] (μ : Measure G) (π : G →* (H →L[ℂ] H)) (φ : G → ℂ) (P : H →L[ℂ] H) : IsOpIntG μ π φ P ↔ IsOpOf μ π φ P

theorem **IsOpOf.unique** {G H : Type} [MeasurableSpace G] [NormedAddCommGroup H] [InnerProductSpace ℂ H] {μ : Measure G} {π : G → (H →L[ℂ] H)} {φ : G → ℂ} {T T' : H →L[ℂ] H} (h : IsOpOf μ π φ T) (h' : IsOpOf μ π φ T') : T = T'

theorem **IsURep.integrable_coef** {G H : Type} [Group G] [TopologicalSpace G] [MeasurableSpace G] [OpensMeasurableSpace G] [NormedAddCommGroup H] [InnerProductSpace ℂ H] {π : G → (H →L[ℂ] H)} (hπ : IsURep π) {μ : Measure G} {φ : G → ℂ} (hφ : Integrable φ μ) (x y : H) : Integrable (fun g => φ g * ⟪x, π g y⟫_ℂ) μ

theorem **IsURep.exists_isOpOf** {G H : Type} [Group G] [TopologicalSpace G] [MeasurableSpace G] [OpensMeasurableSpace G] [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] {π : G → (H →L[ℂ] H)} (hπ : IsURep π) {μ : Measure G} {φ : G → ℂ} (hφ : Integrable φ μ) : ∃ T : H →L[ℂ] H, IsOpOf μ π φ T ∧ ‖T‖ ≤ ∫ g, ‖φ g‖ ∂μ

theorem **IsFdegG.unique** {G H : Type} [MeasurableSpace G] [NormedAddCommGroup H] [InnerProductSpace ℂ H] [Nontrivial H] {μ : Measure G} {σ : G → (H →L[ℂ] H)} {d d' : ℝ} (h : IsFdegG μ σ d) (h' : IsFdegG μ σ d') : d = d'

theorem **isHaarSub_univ** {G : Type} [Group G] [TopologicalSpace G] [MeasurableSpace G] (μ : Measure G) [μ.IsHaarMeasure] : IsHaarSub (Set.univ : Set G) μ

theorem **isHaarSub_map** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (N : Subgroup G) (hN : IsClosed (N : Set G)) (ν : Measure N) [ν.IsHaarMeasure] : IsHaarSub (N : Set G) (ν.map Subtype.val)

@[simp] theorem **coe_nwSub** {n : ℕ} (ε : Fin n → Bool) (W : Submodule ℂ (Fin n → ℂ)) : ((nwSub ε W : Subgroup (UJ ε)) : Set (UJ ε)) = NW ε W

theorem **isClosed_NW** {n : ℕ} (ε : Fin n → Bool) (W : Submodule ℂ (Fin n → ℂ)) : IsClosed (NW ε W)

theorem **exists_isHaarSub_NW** {n : ℕ} (ε : Fin n → Bool) (W : Submodule ℂ (Fin n → ℂ)) : ∃ ν : Measure (UJ ε), IsHaarSub (NW ε W) ν

theorem **slice_apply** (I : AutInst) (f : I.GA → ℂ) (x : I.GA) (v : InfinitePlace I.F) (g : UJ (I.ε v)) : slice I f x v (uMat g) = f (x * (Pi.mulSingle v g, 1))

theorem **Eword_congr** {n : ℕ} (ε : Fin n → Bool) {f f' : Matrix (Fin n) (Fin n) ℂ → ℂ} (h : ∀ g, inUJ ε g → f g = f' g) (w : List (Fin n × Fin n)) : ∀ g, inUJ ε g → Eword ε w f g = Eword ε w f' g

private theorem **exists_uMat_eq_of_inUJ** {n : ℕ} {ε : Fin n → Bool} {g : Matrix (Fin n) (Fin n) ℂ} (hg : inUJ ε g) : ∃ u : UJ ε, uMat u = g

theorem **zOp_eq_gelf** (I : AutInst) (v : InfinitePlace I.F) (m : ℕ) (f : I.GA → ℂ) (x : I.GA) (F : Matrix (Fin I.n) (Fin I.n) ℂ → ℂ) (hF : ∀ g : UJ (I.ε v), F (uMat g) = f (x * (Pi.mulSingle v g, 1))) : zOp I v m f x = gelf (I.ε v) m F 1

theorem **isURep_circRep** : IsURep circRep

theorem **isIrr_circRep** : IsIrr circRep

theorem **isFdegG_circRep** [MeasurableSpace Circle] (μ : Measure Circle) [IsProbabilityMeasure μ] : IsFdegG μ circRep 1

theorem **phG_circHom** (z : Circle) : phG 1 circHom 1 z = star (z : ℂ)

theorem **isOpIntG_circHom** [MeasurableSpace Circle] [BorelSpace Circle] (μ : Measure Circle) [IsProbabilityMeasure μ] : IsOpIntG μ circHom (phG 1 circHom 1) 1 ∧ IsOpOf μ circRep (phG 1 circHom 1) 1

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.IsBesPosOfBesselTied

theorem **kap_symm_tPt** (L A : HSp D) (c : D.host.E) (a : Fin A.rank → AE D) (i : Fin L.rank) (k : Fin A.rank) : (kap L A).symm (tPt L A c a) i k = incA D (-(c * L.hD (L.frame i) (lineVec L))) * a k

theorem **lineNorm_eq** (L : HSp D) : lineNorm L = ∑ i, ∑ k, L.gram k i

theorem **star_lineNorm** (L : HSp D) : star (lineNorm L) = lineNorm L

theorem **tPt_add** (L A : HSp D) (c : D.host.E) (a a' : Fin A.rank → AE D) : tPt L A c (a + a') = tPt L A c a + tPt L A c a'

theorem **tPt_zero** (L A : HSp D) (c : D.host.E) : tPt L A c 0 = 0

theorem **hHom_tPt** {L A : HSp D} (hr : L.rank ≤ A.rank) (c : D.host.E) (a a' : Fin A.rank → AE D) : hHom L A (tPt L A c a') (tPt L A c a) = incA D (c * star c * lineNorm L) * hA A a a'

theorem **cA_hA** (A : HSp D) (a a' : Fin A.rank → AE D) : cA D (hA A a a') = hA A a' a

theorem **cenCo_add** (L : HSp D) (c : D.host.E) (t t' : AE D) : cenCo L c (t + t') = cenCo L c t + cenCo L c t'

theorem **cenCo_zero** (L : HSp D) (c : D.host.E) : cenCo L c 0 = 0

theorem **cenCo_conj** (L : HSp D) (c : D.host.E) (t : AE D) : cenCo L c (cA D t) = -cenCo L c t

theorem **symplH_tPt** {L A : HSp D} (hr : L.rank ≤ A.rank) (c : D.host.E) (a a' : Fin A.rank → AE D) : symplH L A (tPt L A c a) (tPt L A c a') = cenCo L c (-hA A a' a)

theorem **jacOp_mul** (ψ : AddChar (AF D.host.F) ℂ) {L A : HSp D} (hr : L.rank ≤ A.rank) (c : D.host.E) (a a' : Fin A.rank → AE D) (t t' : AE D) (f : WA L A → ℂ) : jacOp ψ L A c a t (jacOp ψ L A c a' t' f) = jacOp ψ L A c (a + a') (t + t' - hA A a' a) f

theorem **symplH_zero_left** (V W : HSp D) (x : WA V W) : symplH V W 0 x = 0

theorem **heis_zero** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (f : WA V W → ℂ) : Heisenberg₂ ψ V W 0 f = f

theorem **jacOp_centre** (ψ : AddChar (AF D.host.F) ℂ) (L A : HSp D) (c : D.host.E) (t : AE D) (f : WA L A → ℂ) : jacOp ψ L A c 0 t f = ψ (cenCo L c t) • f

theorem **IsNegLine.lineNorm_neg** {L : HSp D} (h : IsNegLine L) (τ : D.host.E →+* ℂ) : (τ (lineNorm L)).re < 0

theorem **isIsoSub** : IsIsoSub B d.eM

theorem **even_codim** : Even (B.rank - A.rank)

theorem **hA_aCo** (n : GA D B) (i : Fin A.rank) : hA A (d.aCo n) (vecA A (A.frame i)) = hA B (actV B n d.fTop) (vecA B (ι (A.frame i)))

theorem **theta_eq** (ψ : AddChar (AF D.host.F) ℂ) (Om : GA D A → WA L A → ℂ) (n : GA D B) (g : GA D A) : d.theta ψ Om n g = ψ (cenCo L d.c (d.tCo n)) * Om g (tPt L A d.c (d.aCo n))

theorem **actV_one** (v : B.V) : actV B 1 v = vecA B v

theorem **hD_fTop** (y : B.V) (h : ∀ i, B.hD (d.f i) y = 0) : B.hD d.fTop y = 0

theorem **coords_of** (n : GA D B) (a : Fin A.rank → AE D) (b : Fin d.ρ → AE D) (h : actV B n d.fTop = vecA B d.fTop + ((frameMat B A ι).map (incA D)).mulVec a + ∑ l, b l • vecA B (d.e l)) : d.aCo n = a ∧ d.tCo n = ∑ i : Fin d.ρ, if i.1 + 1 = d.ρ then b i else 0

theorem **jac_of_fix** (ψ : AddChar (AF D.host.F) ℂ) (n : GA D B) (hn : actV B n d.fTop = vecA B d.fTop) (f : WA L A → ℂ) : d.jac ψ n f = f

theorem **theta_one** (ψ : AddChar (AF D.host.F) ℂ) (Om : GA D A → WA L A → ℂ) (g : GA D A) : d.theta ψ Om 1 g = Om g 0

theorem **chr_one** (ψ : AddChar (AF D.host.F) ℂ) : d.chr ψ 1 = 1

theorem **probPair_change** {A B V W : HSp D} (κ : WA A B ≃+ WA V W) (h1 : Continuous κ) (h2 : Continuous κ.symm) (h3 : ∀ x, κ x ∈ ratL V W ↔ x ∈ ratL A B) (m : Measure (WA V W) × Set (WA V W)) (hm : IsProbPair V W m.1 m.2) : ∃ m' : Measure (WA A B) × Set (WA A B), IsProbPair A B m'.1 m'.2 ∧ ∀ φ : WA V W → ℂ, ∫ w in m'.2, φ (κ w) ∂m'.1 = ∫ w in m.2, φ w ∂m.1

theorem **isPureAt_iff** {A B V W : HSp D} {κ : WA A B ≃+ WA V W} {σ} (hκ : IsChange κ σ) (ψ : AddChar (AF D.host.F) ℂ) (ε : AE D) (f : WA V W → ℂ) : IsPureAt ψ V W ε f ↔ IndefiniteRowSideFullHecke.SplittingKernelQuotients.PureAt ψ ε (f ∘ κ)

theorem **isFact_iff** {A B V W : HSp D} {κ : WA A B ≃+ WA V W} {σ} (hκ : IsChange κ σ) (ψ : AddChar (AF D.host.F) ℂ) (f : WA V W → ℂ) : IsFact ψ V W f ↔ IndefiniteRowSideFullHecke.SplittingKernelQuotients.IsPure ψ (f ∘ κ) ∧ IsVacAE ψ V W f

theorem **coefH_smul** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (m) (c : ℂ) (f : WA V W → ℂ) (u : WA V W) : coefH ψ V W m (c • f) u = c * star c * coefH ψ V W m f u

theorem **IsFact.smul** {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {f : WA V W → ℂ} (h : IsFact ψ V W f) (c : ℂ) : IsFact ψ V W (c • f)

theorem **resEta_line** (T : ThetaTied D X) {L : HSp D} (h : L.rank = 1) : ResEta D (T.eChr X.etaT⁻¹) L.rank

theorem **resEta_pow** (T : ThetaTied D X) (n : ℕ) : ResEta D (T.eChr (X.etaT ^ n)) n

theorem **omZ_eq** (Wl : WeilTied D X T) {L A : HSp D} (hL : L.rank = 1) (hA : 1 < A.rank) (χ : X.Chr) (hχ : ResEta D (T.eChr χ) A.rank) (Φ : X.Sch L A) (g : GA D A) : omZ Wl L A Φ g = Wl.evS L A (Wl.om L A X.etaT⁻¹ χ (1, g) Φ)

theorem **isBesPos_of_besselTied** (Bt : BesselTied D X T Wl) (ho : Odd (B.rank - A.rank)) (hr : A.rank ≤ B.rank) (hs : A.SigOK) (hι : HSpace.IsIsomEmb A B ι) (p : X.Pos A B ι) (hp : ∃ u, HEq p (X.posSym u)) : IsBesPos Wl (Bt.dat A B ι ho p) p

theorem **trCusp_one** (F : X.cusp B) : trCusp X B 1 F = F

theorem **isTranslSum_refl** (p : X.Pos A B ι) : IsTranslSum X p p

theorem **IsTranslSum.exists_ne_zero** {p p₀ : X.Pos A B ι} (h : IsTranslSum X p p₀) {σ φ F} (hne : X.Bper A B ι p σ φ F ≠ 0) : ∃ σ' y, X.Bper A B ι p₀ σ' φ (trCusp X B y F) ≠ 0

theorem **isAniso_of_posDef** {V : HSp D} (hs : V.SigOK) (hp : V.IsPosDef) : IsAniso V

theorem **isAniso_Xs** (T : ThetaTied D X) (t : ℕ) {j : ℕ} (hj : j ≤ D.R.m) : IsAniso (X.Xs t j)

theorem **one_lt_rank_Xs** (X : ThetaX D) (t : ℕ) {j : ℕ} (hj : j ≤ D.R.m) : 1 < (X.Xs t j).rank

theorem **PerTied.mem_fact_iff** (P : PerTied Wl) (hr : A.rank < B.rank) : (∀ Φ, Φ ∈ X.fact A B ↔ IndefiniteRowSideFullHecke.SplittingKernelQuotients.IsPure Wl.ψ (Wl.evS A B Φ ∘ kap A B) ∧ IsVacAE Wl.ψ A B (Wl.evS A B Φ)) ∧ ∀ Φ, Φ ∈ X.fact B A ↔ IndefiniteRowSideFullHecke.SplittingKernelQuotients.IsPure Wl.ψ (Wl.evS B A Φ ∘ kapT A B) ∧ IsVacAE Wl.ψ B A (Wl.evS B A Φ)

theorem **FJTied.fj_all** (J : FJTied Wl) (t j : ℕ) (hj : j < D.R.m) (ho : Odd j) (p : X.Pos (X.Xs t j) (X.Ys t j) (X.iXYs t j)) : ∃ (d : FJDat J.line (X.Xs t j) (X.Ys t j) (X.iXYs t j)) (p₀ : X.Pos (X.Xs t j) (X.Ys t j) (X.iXYs t j)), IsFJPos Wl (J.eFJ t j hj ho) d p₀ ∧ IsTranslSum X p p₀

theorem **PerTied.fj_all** (P : PerTied Wl) (t j : ℕ) (hj : j < D.R.m) (ho : Odd j) (p : X.Pos (X.Xs t j) (X.Ys t j) (X.iXYs t j)) : ∃ (d : FJDat P.line (X.Xs t j) (X.Ys t j) (X.iXYs t j)) (p₀ : X.Pos (X.Xs t j) (X.Ys t j) (X.iXYs t j)), IsFJPos Wl (P.eFJ t j hj ho) d p₀ ∧ IsTranslSum X p p₀

theorem **IsFJPos.isLinearMap** {L : HSp D} {k : X.FJd A B ι ≃ₗ[ℂ] X.Sch L A} {d : FJDat L A B ι} {p : X.Pos A B ι} (h : IsFJPos Wl k d p) {μ 𝓕 ν 𝒩} (hv : IsVolPair D A μ 𝓕) (hN : IsUnitN B d.eM ν 𝒩) {s : GA D A → GA D B} (hs : ∀ g, EmbRel B A ι (s g) g) {φ : X.AF A} (hφ : IsAniso A ν φ ∈ X.cusp A) (F : X.cusp B) (hc : ∀ Φ, d.Conv Wl.ψ (omZ Wl L A Φ) μ 𝓕 ν 𝒩 s (T.ev A φ) (T.ev B (F : X.AF B))) : IsLinearMap ℂ fun σ => X.Bper A B ι p σ φ F

theorem **FJTied.isLinearMap** (J : FJTied Wl) (hI : FJInt D) (hN : ∀ (B : HSp D) (ρ : ℕ) (e : Matrix (Fin B.rank) (Fin ρ) D.host.E), IsIsoSub B e → ∃ ν 𝒩, IsUnitN B e ν 𝒩) (t j : ℕ) (hj : j < D.R.m) (ho : Odd j) {μ 𝓕} (hv : IsVolPair D (X.Xs t j) μ 𝓕) (p : X.Pos (X.Xs t j) (X.Ys t j) (X.iXYs t j)) (φ F) : IsLinearMap ℂ fun σ => X.Bj t j p σ φ F

theorem **codim_XYs** (X : ThetaX D) (t : ℕ) {j : ℕ} (hj : j < D.R.m) : (X.Ys t j).rank - (X.Xs t j).rank = D.R.m - 1 - j

theorem **odd_codim_XYs** (X : ThetaX D) (t : ℕ) {j : ℕ} (hj : j < D.R.m) (he : Even j) : Odd ((X.Ys t j).rank - (X.Xs t j).rank)

theorem **coefH_eq_thCoef** (ψ : AddChar (AF D.host.F) ℂ) (V W : HSp D) (m : Measure (WA V W) × Set (WA V W)) (f : WA V W → ℂ) (u : WA V W) : coefH ψ V W m f u = GlobalTheory.Mem.Th.thCoef ψ V W m.1 m.2 f u

theorem **isBesPos_iff_besGlob** (d : BesDat A B ι) (p : X.Pos A B ι) : IsBesPos Wl d p ↔ GlobalTheory.Mem.Th.BesGlob T Wl.ψ d p

theorem **isPureAt_of_all** {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {ε : AE D} {f : WA V W → ℂ} (h : GlobalTheory.Mem.Th.IsPureAt ψ V W ε f) : IsPureAt ψ V W ε f

theorem **isFact_of_isPureFun** {ψ : AddChar (AF D.host.F) ℂ} {V W : HSp D} {f : WA V W → ℂ} (h : GlobalTheory.Mem.Th.IsPureFun ψ V W f) (hv : IsVacAE ψ V W f) : IsFact ψ V W f

theorem **bes_of_realises** (h : Realises Bt Wf Tw Rc) (t j : ℕ) (hj : j < D.R.m) (he : Even j) (d : BesDat (X.Xs t j) (X.Ys t j) (X.iXYs t j)) (hd : (d.y : GA D (X.Ys t j)) = 1) : ∃ p, IsBesPos Wl d p

theorem **bes_only_of_realises** (h : Realises Bt Wf Tw Rc) (t j : ℕ) (hj : j < D.R.m) (he : Even j) (p : X.Pos (X.Xs t j) (X.Ys t j) (X.iXYs t j)) : (∃ d : BesDat (X.Xs t j) (X.Ys t j) (X.iXYs t j), (d.y : GA D (X.Ys t j)) = 1 ∧ IsBesPos Wl d p) ∨ ∃ u, ((_, _, _, p) : Σ (A A' : HSp D) (ι : A.V →ₗ[D.host.E] A'.V), X.Pos A A' ι) = ⟨_, _, _, X.pos1 u⟩

theorem **fact_of_realises** (h : Realises Bt Wf Tw Rc) (hF : FactPure D) (V W : HSp D) : X.fact V W = factSet Wl V W

theorem **negLine_gram** (D : D7) (i j) : (negLine D).gram i j = -1

theorem **isNegLine_negLine** (D : D7) : IsNegLine (negLine D)

theorem **sixSpace_hT** (D : D7) (x y : Fin 6 → D.host.E) : (sixSpace D).hT x y = star (x 0) * y 0 + star (x 1) * y 1 + star (x 2) * y 2 - star (x 3) * y 3 + star (x 4) * y 4 - star (x 5) * y 5

theorem **six_rel** (D : D7) (i j : Fin 2) (x : Fin 2 → D.host.E) : (sixSpace D).hT (eSix D j) (eSix D i) = 0 ∧ (sixSpace D).hT (fSix D j) (fSix D i) = 0 ∧ (sixSpace D).hT (fSix D j) (eSix D i) = (if i = j then 1 else 0) ∧ (sixSpace D).hT (twoToSix D x) (eSix D i) = 0 ∧ (sixSpace D).hT (twoToSix D x) (fSix D i) = 0

theorem **six_isom** (D : D7) : HSpace.IsIsomEmb (Examples₂.posSpace Examples₂.star_isIntegral 2) (sixSpace D) (twoToSix D)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.IsIsotropicFlagUnitaryOneOne

theorem **sphV_zero** : sphV 0 = 1

theorem **volU_one** : volU 1 = 2 * Real.pi

theorem **volU_two** : volU 2 = 8 * Real.pi ^ 3

theorem **dimN_two** : dimN 2 ![1] = 1

theorem **dimN_test** : 9 ^ 2 = (9 - 2 * 3) ^ 2 + 2 * (1 ^ 2 + 2 ^ 2) + 2 * dimN 9 ![1, 2]

theorem **covol_eq** {G : Type} [Group G] [MeasurableSpace G] [MeasurableMul G] (Γ : Subgroup G) [Countable Γ] (μ : Measure G) [μ.IsMulLeftInvariant] (D : Set G) (hD : IsFundamentalDomain Γ D μ) (hm : MeasurableSet D) (hc : ∀ x, ∃ γ ∈ (Γ : Set G), γ * x ∈ D) : covol (Γ : Set G) μ Set.univ = μ D

theorem **isIsotropicFlag_unitary_one_one** : IsIsoFlag e2 ![1] W2

theorem **nM_add** (s t : ℝ) : nM s * nM t = nM (s + t)

theorem **nM_zero** : nM 0 = 1

theorem **nU_add** (s t : ℝ) : nU s * nU t = nU (s + t)

theorem **hs_nU** (t : ℝ) : Real.sqrt (Formalism.REAL.hs (uMat (nU t) - 1)) = |t|

theorem **continuous_nM** : Continuous nM

theorem **continuous_nU** : Continuous nU

theorem **flagVariety_unitary_one_one** : flagN e2 W2 = Set.range nU

theorem **isGaugeSub_flagN** : IsGaugeSub (flagN e2 W2) (dimN 2 ![1]) (Measure.map nU volume)

theorem **mU_act** (a : ℂ) : uMat mU *ᵥ (a • ![1, 1]) = (2 * a) • ![1, 1]

theorem **mU_herm** : (uMat mU)ᴴ = uMat mU

theorem **mU_mem** : mU ∈ flagM e2 W2

theorem **locSize_mU** : locSize ![1] W2 (uMat mU) 0 = 1 / 2

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.IsMirClass

theorem **inMir_one** (n : ℕ) : InMir (1 : Mt g n)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.LamXEquivariant

theorem **dHS_rank_one** (z : ℂ) : dHS (!![z] : Matrix (Fin 1) (Fin 1) ℂ) = |Real.log ‖z‖|

theorem **layerProd_const** (f : ℕ) (τ : Fin 2 → ℝ) (c : ℝ) : layerProd 10 f ![1, 2] τ (fun _ => c) = c ^ ((2 * 3 * 10 - 2 * 3 ^ 2 - (1 + 4)) * f)

theorem **gap_const** : (16 : ℝ) * ((1 / 10 ^ 6)⁻¹) ^ 2 * (4 * 11 + 6) = 8 * 10 ^ 14

theorem **mem_lamX** (x : I.GA) (y : Fin I.n → I.E) : y ∈ lamX I x ↔ I.fmat x.2⁻¹ *ᵥ (fun i => Formalism.UG.inclE (F

theorem **le_lamX_one** : I.L ≤ lamX I 1

theorem **one_mem_GF** : (1 : I.GA) ∈ I.GFset

theorem **lamX_equivariant** (g x : I.GA) (γ : GL (Fin I.n) I.E) (h : I.fmat g.2 = (γ : Matrix (Fin I.n) (Fin I.n) I.E).map (Formalism.UG.inclE (F

theorem **embedding_cases** (w : InfinitePlace I.E) : w.embedding = I.φ (w.comap (algebraMap I.F I.E)) ∨ w.embedding = ComplexEmbedding.conjugate (I.φ (w.comap (algebraMap I.F I.E)))

theorem **frm_frame** (x : I.GA) (w : InfinitePlace I.E) : (frm I x w)ᴴ * I.J.map w.embedding * frm I x w = Formalism.REAL.Jm (I.ε (w.comap (algebraMap I.F I.E)))

theorem **one_mem_nearSet** (x : I.GA) {δ : ℝ} (hδ : 0 ≤ δ) : (1 : I.GA) ∈ nearSet I x δ

theorem **hasType_zero** (x : I.GA) (j : Fin 0 → ℕ) (τ : Fin 0 → ℝ) : HasType I x j τ ↔ stepLt I x (1 / 2) = ⊥

theorem **ph_one** (B : RowsAut) (d : InfinitePlace B.I.F → ℝ) (h : IsDSData B d) : ph B d 1 = ((∏ v, d v : ℝ) : ℂ)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.LatExistsOfSteps

theorem **hconj** (x y : S.V) : star (S.hT x y) = S.hT y x

theorem **hsmulL** (c : E) (x y : S.V) : S.hT (c • x) y = star c * S.hT x y

theorem **mem_latSpan** {r : ℕ} (c : Fin r → FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (v : Fin r → S.V) (y : S.V) : y ∈ latSpan c v ↔ ∃ x : Fin r → E, (∀ i, x i ∈ c i) ∧ y = ∑ i, x i • v i

theorem **latSpan_le** {r : ℕ} (c : Fin r → FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (v : Fin r → S.V) (Λ : Submodule (𝒪 E) S.V) (h : ∀ a, ∀ x ∈ c a, x • v a ∈ Λ) : latSpan c v ≤ Λ

theorem **hfe_of** {r : ℕ} {e f : Fin r → S.V} (hef : ∀ a b, S.hT (e a) (f b) = if a = b then 1 else 0) (a b : Fin r) : S.hT (f a) (e b) = if a = b then 1 else 0

theorem **decomp** {r : ℕ} (e f : Fin r → S.V) (hee : ∀ a b, S.hT (e a) (e b) = 0) (hff : ∀ a b, S.hT (f a) (f b) = 0) (hef : ∀ a b, S.hT (e a) (f b) = if a = b then 1 else 0) (y : S.V) (b : Fin r) : S.hT (e b) (y − ∑ a, S.hT (f a) y • e a − ∑ a, S.hT (e a) y • f a) = 0 ∧ S.hT (f b) (y − ∑ a, S.hT (f a) y • e a − ∑ a, S.hT (e a) y • f a) = 0

theorem **key_split** (Λ : Submodule (𝒪 E) S.V) {r : ℕ} (e f : Fin r → S.V) (P Q : Fin r → FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (hee : ∀ a b, S.hT (e a) (e b) = 0) (hff : ∀ a b, S.hT (f a) (f b) = 0) (hef : ∀ a b, S.hT (e a) (f b) = if a = b then 1 else 0) (hPe : ∀ a, ∀ x ∈ P a, x • e a ∈ Λ) (hQf : ∀ a, ∀ x ∈ Q a, x • f a ∈ Λ) (hfΛ : ∀ a, ∀ y ∈ Λ, S.hT (f a) y ∈ P a) (heΛ : ∀ a, ∀ y ∈ Λ, S.hT (e a) y ∈ Q a) : Λ = latSpan P e ⊔ perpL Λ e f ⊔ latSpan Q f

theorem **orth_span_right** {ι : Type} {v : ι → S.V} {u : S.V} (h : ∀ i, S.hT u (v i) = 0) {y : S.V} (hy : y ∈ Submodule.span E (Set.range v)) : S.hT u y = 0

theorem **orth_span_left** {ι : Type} {v : ι → S.V} {u : S.V} (h : ∀ i, S.hT (v i) u = 0) {x : S.V} (hx : x ∈ Submodule.span E (Set.range v)) : S.hT x u = 0

theorem **iso_span** {ι : Type} {v : ι → S.V} (h : ∀ i j, S.hT (v i) (v j) = 0) {x y : S.V} (hx : x ∈ Submodule.span E (Set.range v)) (hy : y ∈ Submodule.span E (Set.range v)) : S.hT x y = 0

theorem **orth_pair** {u a b y : S.V} (ha : S.hT u a = 0) (hb : S.hT u b = 0) (hy : y ∈ Submodule.span E ({a, b} : Set S.V)) : S.hT u y = 0

theorem **zero_of_dual** {r : ℕ} {e f : Fin r → S.V} (hfe : ∀ a b, S.hT (f a) (e b) = if a = b then 1 else 0) {v : S.V} (hv : v ∈ Submodule.span E (Set.range e)) (h0 : ∀ b, S.hT (f b) v = 0) : v = 0

theorem **mem_of_dual** {Λ : Submodule (𝒪 E) S.V} (hU : Formalism.HermitianLatticesTheory.IsUnimodular S.hT Λ) {x : S.V} (h : ∀ y ∈ Λ, ∃ o : 𝒪 E, algebraMap (𝒪 E) E o = S.hT y x) : x ∈ Λ

theorem **dualE** {Λ : Submodule (𝒪 E) S.V} (hU : Formalism.HermitianLatticesTheory.IsUnimodular S.hT Λ) {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (hI : I ≠ 0) (hst : ∀ x ∈ I, star x ∈ I) {e : S.V} (hIe : ∀ x ∈ I, x • e ∈ Λ) {y : S.V} (hy : y ∈ Λ) : S.hT e y ∈ I⁻¹

theorem **dualF** {Λ : Submodule (𝒪 E) S.V} (hU : Formalism.HermitianLatticesTheory.IsUnimodular S.hT Λ) {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (hI : I ≠ 0) (hst : ∀ x ∈ I, star x ∈ I) {f : S.V} (hf : ∀ y ∈ Λ, S.hT f y ∈ I) {x : E} (hx : x ∈ I⁻¹) : x • f ∈ Λ

theorem **fi_neg** {E : Type} [Field E] [NumberField E] {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} {u : E} (hu : u ∈ I) : -u ∈ I

theorem **fi_sum** {E : Type} [Field E] [NumberField E] {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} {r : ℕ} (g : Fin r → E) (hg : ∀ k, g k ∈ I) : ∑ k, g k ∈ I

theorem **fi_intm** {E : Type} [Field E] [NumberField E] {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (o : 𝒪 E) {t : E} (ht : t ∈ I) : (o : E) * t ∈ I

theorem **genFrame_holds** : GenFrame

theorem **locgen** {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (hI : I ≠ 0) {P : Ideal (𝒪 E)} {c : E} (hcI : c ∈ I) (hcP : c ∉ I * (P : FractionalIdeal (nonZeroDivisors (𝒪 E)) E)) : ∀ x ∈ I, ∃ s : 𝒪 E, s ∉ P ∧ ∃ o : 𝒪 E, (s : E) * x = (o : E) * c

theorem **exists_gen** {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (hI : I ≠ 0) {P : Ideal (𝒪 E)} (hP : P ≠ ⊤) : ∃ c ∈ I, c ∉ I * (P : FractionalIdeal (nonZeroDivisors (𝒪 E)) E)

theorem **mem_locP** {V : Type} [AddCommGroup V] [Module (𝒪 E) V] (P : Ideal (𝒪 E)) [P.IsPrime] (L : Submodule (𝒪 E) V) (x : V) : x ∈ Formalism.HermitianLatticesTheory.localization₂ P L ↔ ∃ s : 𝒪 E, s ∉ P ∧ s • x ∈ L

theorem **localization_le_of_le_localization₂** {V : Type} [AddCommGroup V] [Module (𝒪 E) V] (P : Ideal (𝒪 E)) [P.IsPrime] {A B : Submodule (𝒪 E) V} (h : A ≤ Formalism.HermitianLatticesTheory.localization₂ P B) : Formalism.HermitianLatticesTheory.localization₂ P A ≤ Formalism.HermitianLatticesTheory.localization₂ P B

theorem **le_locP** {V : Type} [AddCommGroup V] [Module (𝒪 E) V] (P : Ideal (𝒪 E)) [hP : P.IsPrime] {A : Submodule (𝒪 E) V} : A ≤ Formalism.HermitianLatticesTheory.localization₂ P A

theorem **localization_latSpan** {S : HSpace E} {r : ℕ} (P : Ideal (𝒪 E)) [P.IsPrime] (I : Fin r → FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (v : Fin r → S.V) (cl : Fin r → E) (hcl : ∀ a, cl a ∈ I a) (hgen : ∀ a, ∀ x ∈ I a, ∃ s : 𝒪 E, s ∉ P ∧ ∃ o : 𝒪 E, (s : E) * x = (o : E) * cl a) : Formalism.HermitianLatticesTheory.localization₂ P (latSpan I v) = Formalism.HermitianLatticesTheory.localization₂ P (Submodule.span (𝒪 E) (Set.range fun a => cl a • v a))

theorem **satBasis_of** (h : LatEx.SatBasisV) : SatBasis

theorem **ddG_of** (hU : Unram) (hSD : LatEx.SplitDiff) : DdG D

theorem **aa_gen_one** (hDd : DdG D) (P : HeightOneSpectrum (𝒪 D.host.E)) (hP : ((gS D : ℕ) : 𝒪 D.host.E) ∈ P.asIdeal) (a : Fin D.R.rS) : (1 : D.host.E) ∈ aaS D a ∧ (1 : D.host.E) ∉ aaS D a * (P.asIdeal : FractionalIdeal (nonZeroDivisors (𝒪 D.host.E)) D.host.E)

theorem **latExists_of_steps** (hSB : SatBasis) (hH : HypRep) (hI : IsoW D) (hDd : DdG D) : Formalism.L2.SG.SplitPrimeNormLemmas.LatExists D

theorem **latExists_of** (hH : HypRep) (hU : Unram) (D : Formalism.BasicNotions.D7) (hI : IsoW D) : Formalism.L2.SG.SplitPrimeNormLemmas.LatExists D

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.LevelMatchIff

theorem **levelMatch_iff** (X : HidaSetting D) {u : D.what₂.Word} {F : Formalism.L2.Forms.Forms0 D u} (Tr : Transport X u F) (hK : X.LevelSubgroupCharacter) : LevelMatch X Tr hK ↔ LatMatchH X u F Tr.wt Tr.ιA Tr.h0 ∧ LevelKH X u F Tr.wt Tr.ιA Tr.h0 hK

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.LinkUnique

theorem **mem_Mq** (x : Fin (D.R.N 0) → D.host.E) : x ∈ Mq wt ↔ (∑ j, x j • qFr wt j) ∈ (VS D).L0

theorem **embA_add** (x y : Fin (D.R.N 0) → D.host.E ⊗[D.host.F] Ad D.host.F) : embA X wt ιA (x + y) = embA X wt ιA x + embA X wt ιA y

theorem **embA_smul** (a : D.host.E ⊗[D.host.F] Ad D.host.F) (x : Fin (D.R.N 0) → D.host.E ⊗[D.host.F] Ad D.host.F) : embA X wt ιA (a • x) = ιA.symm a • embA X wt ιA x

theorem **embA_rep** (h : X.B.I.Gf) (y : UPts (Formalism.L2.Forms.sigD D) (Formalism.L2.Forms.gramN D 0 (Nat.zero_le _)) (Ad D.host.F)) (hr : RepF X wt ιA h y) (x : Fin (D.R.N 0) → D.host.E ⊗[D.host.F] Ad D.host.F) : X.B.I.fmat h ∗ᵥ embA X wt ιA x = embA X wt ιA (y.1.1 ∗ᵥ x)

theorem **embA_rat** (hι : GlobalTheory.St.IsAdeleIso X ιA) (x : Fin (D.R.N 0) → D.host.E) : embA X wt ιA (fun j => inclE (F

theorem **embA_mem** (hι : GlobalTheory.St.IsAdeleIso X ιA) (x : Fin (D.R.N 0) → D.host.E) (hx : x ∈ Mq wt) : embA X wt ιA (fun j => inclE (F

theorem **link_unique** (u : D.what₂.Word) (Fr Fr' : FrQ D wt) (h : FrLink X u wt ιA Fr) (h' : FrLink X u wt ιA Fr') : Fr.eps = Fr'.eps ∧ Fr.psi = Fr'.psi ∧ (∀ i, embA X wt ιA (Fr.b i) − embA X wt ιA (Fr'.b i) ∈ mL X.B.I X.B.g) ∧ ∀ i, embA X wt ιA (Fr.bd i) − embA X wt ιA (Fr'.bd i) ∈ mL X.B.I X.B.g

theorem **level_rank** : X.B.g = D.host.g 0 ∗ D.host.g 1 ∧ D.R.N 0 = D.R.k

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.LocDecOfDecGap

theorem **locDec_of_decOp** (hD : DecOp) (B : RowsAut) (d : InfinitePlace B.I.F → ℝ) (h : ℝ) (hB : IsDSData B d) (hg : GapGe B h) (hh : ∀ v, 8 ∗ (B.I.n : ℝ) ∗ rkR (B.I.ε v) < h) : LocDec B h

theorem **locDec_of_decGap** (hD : GlobalTheory.DSId.DecGap) (B : RowsAut) (d : InfinitePlace B.I.F → ℝ) (h : ℝ) (hB : IsDSData B d) (hg : GapGe B h) (hh : ∀ v, 120 ∗ (B.I.n : ℝ) ∗ ((rkR (B.I.ε v) : ℕ) : ℝ) ^ ((3 : ℝ) / 2) ≤ h) : LocDec B h

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.MatrixOfBesselTwoSidedAction

theorem **of_invM** (x : Mt k r) : (Matrix.of (invM x) : Mat k r) = (Matrix.of x : Mat k r)⁻¹

theorem **invM_invM** (x : Mt k r) (h : IsUnit (Matrix.of x : Mat k r).det) : invM (invM x) = x

theorem **measurable_invM** : Measurable (invM : Mt k r → Mt k r)

theorem **of_rmulM** (a : Mat k r) (x : Mt k r) : (Matrix.of (rmulM a x) : Mat k r) = Matrix.of x ∗ a

theorem **continuous_rmulM** (a : Mat k r) : Continuous (rmulM a : Mt k r → Mt k r)

theorem **matrix_of_besselTwoSidedAction** (γ : Mat k r) (p : Mt k r × Mt k r) : (Matrix.of (act2 γ p) : Mat k r) = Matrix.of p.1 ∗ γ ∗ Matrix.of (invM p.2)

theorem **opn_nonneg** {n : ℕ} (g : Matrix (Fin n) (Fin n) ℂ) : 0 ≤ opn g

theorem **opn_mul_le** {n : ℕ} (g h : Matrix (Fin n) (Fin n) ℂ) : opn (g ∗ h) ≤ opn g ∗ opn h

theorem **opn_one** {n : ℕ} : opn (1 : Matrix (Fin (n + 1)) (Fin (n + 1)) ℂ) = 1

theorem **hPart_one** : hPart k r 1 = 1

theorem **PiS_one** (ψ : Mat k r → ℂ) (π : Mat k r → (Wp →L[ℂ] Wp)) : PiS ψ π 1 = ψ 1 • π 1

theorem **coef_smul_left** (σ : Mat k r → (H →L[ℂ] H)) (c : ℂ) (u v : H) (g : Mat k r) : coef σ (c • u) v g = c ∗ coef σ u v g

theorem **coef_smul_right** (σ : Mat k r → (H →L[ℂ] H)) (c : ℂ) (u v : H) (g : Mat k r) : coef σ u (c • v) g = star c ∗ coef σ u v g

theorem **coef_add_left** (σ : Mat k r → (H →L[ℂ] H)) (u u' v : H) (g : Mat k r) : coef σ (u + u') v g = coef σ u v g + coef σ u' v g

theorem **coef_add_right** (σ : Mat k r → (H →L[ℂ] H)) (u v v' : H) (g : Mat k r) : coef σ u (v + v') g = coef σ u v g + coef σ u v' g

theorem **rankOneInnerOperator_apply** (a : Wp) (b η : Wq) : rk1 a b η = ⟪b, η⟫_ℂ • a

theorem **norm_rankOneInnerOperator** (a : Wp) (b : Wq) : ‖rk1 a b‖ = ‖a‖ ∗ ‖b‖

theorem **mem_DD** (σ : Mat k r → (H →L[ℂ] H)) (w : H) {g : Mat k r} (hg : g ∈ GG k r) : σ g w ∈ DD σ w

theorem **Decay.mono** {σ : Mat k r → (H →L[ℂ] H)} {V V' : Set H} {a : ℝ} (h : Decay σ V a) (hV : V' ⊆ V) : Decay σ V' a

theorem **Greg_subset** (z z' : Fin (2 ∗ r + k + 1) → ℂ) : Greg k r z z' ⊆ GG k r

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.MemConfTOfDetNeZero

theorem **hOf_isHermitian** {a : ℕ} (t : HCo a) : (hOf t).IsHermitian

theorem **mem_dualM** (σ : E →+* E) {a : ℕ} (𝔞 : Fin a → FractionalIdeal (𝒪 E)⁰ E) (T : Matrix (Fin a) (Fin a) E) (x : Fin a → E) : x ∈ dualM σ 𝔞 T ↔ ∀ y ∈ tupLat 𝔞 (1 : Matrix (Fin a) (Fin a) E), form σ T x y ∈ (algebraMap (𝒪 E) E).range

theorem **mem_dualM_iff_mulVec** (σ : E →+* E) {a : ℕ} (𝔞 : Fin a → FractionalIdeal (𝒪 E)⁰ E) (T : Matrix (Fin a) (Fin a) E) (x : Fin a → E) : x ∈ dualM σ 𝔞 T ↔ T ∗ᵥ x ∈ dualM σ 𝔞 1

theorem **latMap_dualM** (σ : E →+* E) {a : ℕ} (𝔞 : Fin a → FractionalIdeal (𝒪 E)⁰ E) {T : Matrix (Fin a) (Fin a) E} (hT : T.det ≠ 0) : latMap T (dualM σ 𝔞 T) = dualM σ 𝔞 1

theorem **relIndex_map_submodule** {R M N : Type} [Ring R] [AddCommGroup M] [AddCommGroup N] [Module R M] [Module R N] (f : M →ₗ[R] N) (hf : Function.Injective f) (A B : Submodule R M) : (A.map f).toAddSubgroup.relIndex (B.map f).toAddSubgroup = A.toAddSubgroup.relIndex B.toAddSubgroup

theorem **cork_eq_of_det_ne_zero** (σ : E →+* E) {a : ℕ} (𝔞 : Fin a → FractionalIdeal (𝒪 E)⁰ E) {T : Matrix (Fin a) (Fin a) E} (hT : T.det ≠ 0) (v : HeightOneSpectrum (𝒪 F)) : cork σ 𝔞 T v = Nat.log (Ideal.absNorm v.asIdeal ^ 2) ((tupLat 𝔞 (1 : Matrix (Fin a) (Fin a) E) ⊔ (v.asIdeal.map (algebraMap (𝒪 F) (𝒪 E))) • dualM σ 𝔞 T).toAddSubgroup.relIndex (dualM σ 𝔞 T).toAddSubgroup)

theorem **mem_tupLat_one** {a : ℕ} (x : Fin a → E) : x ∈ tupLat (fun _ : Fin a => (1 : FractionalIdeal (𝒪 E)⁰ E)) (1 : Matrix (Fin a) (Fin a) E) ↔ ∀ i, x i ∈ (algebraMap (𝒪 E) E).range

theorem **dualM_one_one** (σ : E →+* E) {a : ℕ} : dualM σ (fun _ : Fin a => (1 : FractionalIdeal (𝒪 E)⁰ E)) 1 = tupLat (fun _ : Fin a => (1 : FractionalIdeal (𝒪 E)⁰ E)) (1 : Matrix (Fin a) (Fin a) E)

theorem **cork_one_one** (σ : E →+* E) {a : ℕ} (v : HeightOneSpectrum (𝒪 F)) : cork σ (fun _ : Fin a => (1 : FractionalIdeal (𝒪 E)⁰ E)) 1 v = 0

theorem **cork_one_zero** (σ : E →+* E) (h2 : Module.finrank F E = 2) {a : ℕ} (v : HeightOneSpectrum (𝒪 F)) : cork σ (fun _ : Fin a => (1 : FractionalIdeal (𝒪 E)⁰ E)) 0 v = a

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.NcardImageOfLe

theorem **ncard_image_of_le** {W : Type} [AddCommGroup W] {A B : AddSubgroup W} (h : B ≤ A) (S : Set W) (hS : ∀ x ∈ S, ∀ a ∈ A, x + a ∈ S) : ((QuotientAddGroup.mk : W → W ⧸ B) '' S).ncard = ((QuotientAddGroup.mk : W → W ⧸ A) '' S).ncard ∗ B.relIndex A

theorem **_root_.GlobalTheory.HL.ContModel.relVol_of_le** {W : Type} [AddCommGroup W] {W0 A B : AddSubgroup W} (h : B ≤ A) (hB : B.relIndex W0 ≠ 0) : _root_.GlobalTheory.HL.relVol W0 A = B.relIndex A ∗ _root_.GlobalTheory.HL.relVol W0 B

private theorem **ncard_image_mk_eq_mul_relIndex_of_le** {W : Type} [AddCommGroup W] {A B : AddSubgroup W} (h : B ≤ A) {S : Set W} (hS : ∀ x ∈ S, ∀ a ∈ A, x + a ∈ S) : ((QuotientAddGroup.mk : W → W ⧸ B) '' S).ncard = ((QuotientAddGroup.mk : W → W ⧸ A) '' S).ncard * B.relIndex A

theorem **_root_.GlobalTheory.HL.ContModel.cont_eq** {W : Type} [AddCommGroup W] {W0 A : AddSubgroup W} {S : Set W} (hA : _root_.GlobalTheory.HL.GoodSub W0 A S) : _root_.GlobalTheory.HL.cont W0 S = (((QuotientAddGroup.mk : W → W ⧸ A) '' S).ncard : ℝ) * _root_.GlobalTheory.HL.relVol W0 A

theorem **cont_subgroup** {W : Type} [AddCommGroup W] {W0 A : AddSubgroup W} (h1 : A.relIndex W0 ≠ 0) (h2 : W0.relIndex A ≠ 0) : cont W0 (A : Set W) = relVol W0 A

theorem **cont_self** {W : Type} [AddCommGroup W] (W0 : AddSubgroup W) : cont W0 (W0 : Set W) = 1

theorem **cont_even** : cont (⊤ : AddSubgroup ℤ) (AddSubgroup.zmultiples (2 : ℤ) : Set ℤ) = 1 / 2

theorem **hf_sum_smul** (L : LocE F E) {n m : ℕ} (Hm : Matrix (Fin n) (Fin n) E) (x : Fin m → Fin n → E) (c d : Fin m → E) : hf L Hm (∑ i, c i • x i) (∑ k, d k • x k) = ∑ i, ∑ k, L.cj (d k) * hf L Hm (x i) (x k) * c i

theorem **_root_.GlobalTheory.HL.ContModel.gramT_rmul** (L : _root_.Formalism.MainNotions.S.Loc.LocE F E) {n m : ℕ} (Hm : Matrix (Fin n) (Fin n) E) (x : Fin m → Fin n → E) (A : Matrix (Fin m) (Fin m) E) : _root_.GlobalTheory.HL.gramT L Hm (_root_.GlobalTheory.HL.rmul x A) = _root_.GlobalTheory.HL.cT L A * _root_.GlobalTheory.HL.gramT L Hm x * A

theorem **smul_inf_of_fix** (L : LocE F E) {n : ℕ} (Hm : Matrix (Fin n) (Fin n) E) (Y : Submodule E (Fin n → E)) (g : fixG L Hm Y) (Λ : Submodule L.O (Fin n → E)) : (g • Λ) ⊓ Y.restrictScalars L.O = Λ ⊓ Y.restrictScalars L.O

theorem **sigMass_top** (L : LocE F E) {n : ℕ} (Hm : Matrix (Fin n) (Fin n) E) (Λ0 Mh : Submodule L.O (Fin n → E)) : sigMass L Hm ⊤ Λ0 Mh = if IsSD L Hm Mh then 1 else 0

theorem **_root_.GlobalTheory.HL.ContModel.cj_mem** (L : _root_.Formalism.MainNotions.S.Loc.LocE F E) {x : E} (hx : x ∈ L.O) : L.cj x ∈ L.O

theorem **mem_hermLat** (L : LocE F E) (m : ℕ) (X : Matrix (Fin m) (Fin m) E) : X ∈ hermLat L m ↔ IsHerm L X ∧ IsIntM L X

theorem **hf_single** (L : LocE F E) {n : ℕ} (Hm : Matrix (Fin n) (Fin n) E) (a b : Fin n) : hf L Hm (Pi.single a 1) (Pi.single b 1) = Hm a b

theorem **gramT_std** (L : LocE F E) {n : ℕ} (Hm : Matrix (Fin n) (Fin n) E) : gramT L Hm (fun a => Pi.single a 1) = Hm.transpose

theorem **isPrimT_std** (L : LocE F E) {n : ℕ} (Λ : Submodule L.O (Fin n → E)) (h : ∀ a, (Pi.single a 1 : Fin n → E) ∈ Λ) : IsPrimT L 1 Λ (fun a => Pi.single a 1)

theorem **_root_.GlobalTheory.HL.ContModel.gramT_thick** (L : _root_.Formalism.MainNotions.S.Loc.LocE F E) (ϖ : 𝒪[F]) {n m : ℕ} (Hm : Matrix (Fin n) (Fin n) E) (Ω : Set (Fin m → Fin n → E)) (Γ : Matrix (Fin m) (Fin m) E) (i : ℕ) (hΩ : ∀ y ∈ Ω, _root_.GlobalTheory.HL.gramT L Hm y = Γ) : ∀ x ∈ _root_.GlobalTheory.HL.thick L ϖ Ω i, _root_.GlobalTheory.HL.gramT L Hm x ∈ _root_.GlobalTheory.HL.horb L ϖ Γ i

theorem **primCnt_zero** (L : LocE F E) (ϖ : 𝒪[F]) (n j : ℕ) (Gm : Matrix (Fin 0) (Fin 0) E) : primCnt L ϖ n j Gm = 1

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.OneScaleDatumRelabelRec

theorem **glob_relabelRec** {x eps CD Cg CT : ℝ} : Glob (relabelRec D Lw h) x eps CD Cg CT ↔ Glob D x eps CD Cg CT

theorem **nodes_relabelRec** {x eps : ℝ} : Nodes (relabelRec D Lw h) x eps ↔ Nodes D x eps

theorem **oneScale_relabelRec** {x eps eta : ℝ} : OneScale (relabelRec D Lw h) x eps eta ↔ OneScale D x eps eta

theorem **oneScaleDatum_relabelRec** {x eps eta : ℝ} {J : ℕ} : OneScaleDatum (relabelRec D Lw h) x eps eta J ↔ OneScaleDatum D x eps eta J

theorem **exists_labels_relabelRec** {x eps CG : ℝ} (L : LiveLabels D x eps CG) : ∃ (Lw : ℕ) (h : 2 ≤ Lw), Labels (relabelRec D Lw h) x eps CG

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.OverOfIsTermOver

theorem **tri_refl** {K W : Type} [Semiring K] [AddCommMonoid W] [Module K W] {n : ℕ} (f : Fin n → W) : Tri K f f

theorem **tri_two_iff** {K W : Type} [Semiring K] [AddCommMonoid W] [Module K W] (f g : Fin 2 → W) : Tri K f g ↔ g 0 = f 0 ∧ ∃ c : K, g 1 = c • f 0 + f 1

theorem **img_mul'** (g h : V ≃ₗ[E] V) (Λ : Submodule M.L.O V) : img (g * h) Λ = img g (img h Λ)

theorem **img_one'** (Λ : Submodule M.L.O V) : img (1 : V ≃ₗ[E] V) Λ = Λ

theorem **move_one** (t : Term V M r) : move 1 t = t

theorem **move_mul** (g h : V ≃ₗ[E] V) (t : Term V M r) : move (g * h) t = move g (move h t)

theorem **over_of_isTermOver** {H : HermSp M V} (P : FPair H r) (d : Datum P) (t : Term V M r) (h : IsTermOver P d t) : Over P d t

theorem **over_move_iff** {H : HermSp M V} (P : FPair H r) (d : Datum P) (t : Term V M r) (g : V ≃ₗ[E] V) (hg : IsUnitary H g) : Over P d (move g t) ↔ Over P d t

theorem **norms_of_over** {H : HermSp M V} {P : FPair H r} {d : Datum P} {t : Term V M r} (h : Over P d t) : t.T = P.A.T ∧ t.T' = P.B.T

theorem **mem_NGrp_iff** (H : HermSp M V) (Λ : Submodule M.L.O V) (f : Fin (r + 1) → V) (n : V ≃ₗ[E] V) : n ∈ H.NGrp Λ f ↔ n ∈ NU H f ∧ img n Λ = Λ

theorem **NU_eq_NSet** {H : HermSp M V} (Fr : FrameD H r) : NU H Fr.f = Fr.NSet

theorem **NU_zero** (H : HermSp M V) (f : Fin 1 → V) : NU (r

theorem **NU_nontrivial** {H : HermSp M V} (Fr : FrameD H (r + 1)) : ∃ n ∈ NU H Fr.f, n ≠ 1

theorem **cong_none** (H : HermSp M V) (h : M.split = none) (Λ : Submodule M.L.O V) (f f' : Fin (r + 1) → V) (x : V) : Cong H Λ f f' x ↔ x ∈ Λ

theorem **cong_zero** (H : HermSp M V) (Λ : Submodule M.L.O V) (f f' : Fin 1 → V) (x : V) : Cong (r

theorem **inhabited** (M : Model F E) (hM : M.OK) (m : ℕ) : (pair M m).RelG (theta M) ∧ Over (pair M m) (datum M m) (move (LinearEquiv.neg E) (term M m)) ∧ (∀ d, ¬ IsTermOver (pair M m) d (move (LinearEquiv.neg E) (term M m))) ∧ ∀ d, ¬ Over (pair M m) d { term M m with T

theorem **chg_frameAct** {n : ℕ} (f : Fin n → (VS D).V) (ξ : Matrix (Fin n) (Fin n) (𝒪 D.host.E)) : Chg f (ξ.map (fun a => (a : D.host.E))) (frameAct f ξ)

theorem **ext_one** (p : PlF D) : Ext p (1 : (VS D).V ≃ₗ[D.host.E] (VS D).V) 1

theorem **ext_unique** (p : PlF D) (γ : (VS D).V ≃ₗ[D.host.E] (VS D).V) (g g' : Vp D p ≃ₗ[(D.s.atP p).Ep] Vp D p) (h : Ext p γ g) (h' : Ext p γ g') : g = g'

theorem **moveT_eq** {p : PlF D} {Mp : Loc.Model (D.s.atP p).Fp (D.s.atP p).Ep} (g : Vp D p ≃ₗ[(D.s.atP p).Ep] Vp D p) (t : Loc.Term (Vp D p) Mp D.R.rS) : moveT g t = move g t

theorem **locF_eq** (p : PlF D) {T : D.host.E} (F : LFlag (VS D) D.R.rS T) : locF Y p F = scaled (zOf Y) p (frame F)

theorem **locT_eq** (t : TauC Y u u') (p : PlF D) : locT Y u u' t p = locU (zOf Y) p t.toTerm.lam (lf D (X.Fu u)) (Quot.out t.toTerm.cls).1 (Tu D u) (Tu D u')

theorem **overOrb_iff** {Cl Nu : Type} (Fm : LocFam (zOf Y) Cl Nu) (t : TauC Y u u') (p : PlF D) (d : LocNG.Datum (Fm.FP p)) : OverOrb Fm t p d ↔ Over (Fm.FP p) d (locT Y u u' t p)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.PairBdNonneg

private lemma **vInv_nonneg'** (K : Type) [Field K] [NumberField K] : 0 ≤ _root_.Formalism.HermitianLatticesTheory.vInv K

theorem **_root_.GlobalTheory.DegAsm.pairBd_nonneg** (K : Type) [Field K] [NumberField K] (k g j T x x' : ℕ) : 0 ≤ _root_.GlobalTheory.DegAsm.pairBd K k g j T x x'

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.PosPartUnitaryMul

theorem **stdRep_apply** (y : Matrix (Fin n) (Fin n) ℂ) (v : EuclideanSpace ℂ (Fin n)) (i : Fin n) : stdRep y v i = ∑ j, y i j * v j

theorem **mem_Kset_allPos** {g : Matrix (Fin n) (Fin n) ℂ} (hg : g ∈ Matrix.unitaryGroup (Fin n) ℂ) : g ∈ Formalism.REAL.Kset (allPos n)

theorem **stdRep_irred** (S : Submodule ℂ (EuclideanSpace ℂ (Fin n))) (hS : IsKStableOn (allPos n) stdRep S) : S = ⊥ ∨ S = ⊤

theorem **isTau_std** (n : ℕ) : IsTau (allPos (n + 1)) (-Pi.single (Fin.last n) (1 : ℝ)) stdRep (EuclideanSpace.single (Fin.last n) (1 : ℂ))

theorem **IsZEigen.smooth** {ε : Fin n → Bool} {χ : Fin n → ℂ} {f : Matrix (Fin n) (Fin n) ℂ → ℂ} (h : IsZEigen ε χ f) : Formalism.REAL.SmoothOnG ε f

theorem **IsZFinite.smooth** {ε : Fin n → Bool} {f : Matrix (Fin n) (Fin n) ℂ → ℂ} (h : IsZFinite ε f) : Formalism.REAL.SmoothOnG ε f

theorem **IsZEigen.isZFinite** {ε : Fin n → Bool} {χ : Fin n → ℂ} {f : Matrix (Fin n) (Fin n) ℂ → ℂ} (h : IsZEigen ε χ f) : IsZFinite ε f

theorem **gelfVal_succ** (χ : Fin n → ℂ) (k : ℕ) : gelfVal χ (k + 1) = ∑ a, ((Formalism.REAL.Amat (fun c => χ c + ((n : ℂ) - 1) / 2) ^ k) *ᵥ (fun c => χ c - (Formalism.QuantumDeterminants.rhoStd n c : ℂ))) a

theorem **gelfVal_rho** (k : ℕ) : gelfVal (fun a => (Formalism.QuantumDeterminants.rhoStd n a : ℂ)) (k + 1) = 0

theorem **Eword_const** (ε : Fin n → Bool) (c : ℂ) (p : Fin n × Fin n) (w : List (Fin n × Fin n)) : Formalism.REAL.Eword ε (p :: w) (fun _ => c) = fun _ => 0

theorem **isZEigen_const** (ε : Fin n → Bool) (c : ℂ) : IsZEigen ε (fun a => (Formalism.QuantumDeterminants.rhoStd n a : ℂ)) (fun _ => c)

theorem **isSqIntIrr_std** (n : ℕ) : IsSqIntIrr (allPos (n + 1)) stdRep

theorem **posPart_posDef_sq** {g : Matrix (Fin n) (Fin n) ℂ} (hg : IsUnit g) : (posPart g).PosDef ∧ posPart g * posPart g = gᴴ * g

theorem **posPart_eq_of_sq** {g p : Matrix (Fin n) (Fin n) ℂ} (hp : p.PosDef) (h : p * p = gᴴ * g) : posPart g = p

theorem **posPart_unitary_mul** {k p : Matrix (Fin n) (Fin n) ℂ} (hk : k ∈ Matrix.unitaryGroup (Fin n) ℂ) (hp : p.PosDef) : posPart (k * p) = p

theorem **Psi_unitary_mul** (ε : Fin n → Bool) (lam : Fin n → ℝ) {W : Type} [NormedAddCommGroup W] [InnerProductSpace ℂ W] (τ : Matrix (Fin n) (Fin n) ℂ → (W →L[ℂ] W)) (w : W) {k p : Matrix (Fin n) (Fin n) ℂ} (hk : k ∈ Matrix.unitaryGroup (Fin n) ℂ) (hp : p.PosDef) : Psi ε lam τ w (k * p) = (τ k).comp (ps ε lam τ w p)

theorem **Phi_at_one** (lam : Fin n → ℝ) : Phi lam 1 = 1

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.PrCloudAddPrNode

theorem **HRep.EE_eq_add** (A : HRep k r) (P : Mat k r) (x : Idx k r 1 → A.H) : A.EE P x = A.EEd P x + A.EEo P x

theorem **prCloud_add_prNode** {P : Mat k r} (hT : IsTensDec R true P 1 μ ι p) {x : Idx k r 1 → (R μ).H} (hx : (R μ).SmT x) : prCloud R μ ι p x + prNode R μ ι p x = x

theorem **HRep.expect_le_iff** (A : HRep k r) {W : Type} [NormedAddCommGroup W] (ν : Measure (Mt k r)) (B : A.H → W) (φ : Mat k r → ℝ) {w : A.H} (hw : w ≠ 0) (C : ℝ≥0∞) : A.expect ν B φ w ≤ C ↔ A.mass ν B φ w ≤ C * ‖w‖ₑ ^ 2

theorem **Flag.Iso.expect_one** {F : Flag k r} {A : HRep k r} {B : A.H → F.W} (hB : F.Iso A B) {w : A.H} (hs : w ∈ smoothVecs (k

theorem **InPP.mono** {n : ℕ} {lam μ : Fin n → ℝ} {s t : ℝ} (h : InPP lam s μ) (hst : s ≤ t) : InPP lam t μ

theorem **inPP_self** {n : ℕ} (lam : Fin n → ℝ) {t : ℝ} (ht : 0 ≤ t) : InPP lam t lam

theorem **InPP.abs_le** {n : ℕ} {lam μ : Fin n → ℝ} {t : ℝ} (h : InPP lam t μ) (a : Fin n) : |μ a - lam a| ≤ t

theorem **InPP.gap** {n : ℕ} {lam μ : Fin n → ℝ} {t : ℝ} (h : InPP lam t μ) {a b : Fin n} (hab : a ≠ b) : lam a - lam b - t ≤ μ a - μ b

theorem **InPP.add_wtI** {lam μ : Fin (2 * r + k + 1) → ℝ} {t : ℝ} (h : InPP lam t μ) (dual : Bool) {l : ℕ} (I : Idx k r l) : InPP lam (t + l) (μ + wtI dual I)

theorem **abs_add_le_bStar** {n : ℕ} (μ : Fin n → ℝ) (a : Fin n) : |μ a| + n ≤ bStar μ

theorem **node_gap** (hs : IsShape k r lam bv) (hg : GapAtLeast (sgnCN k r) lam h) {i j : Fin (2 * r + k + 1)} (hi : k - 1 ≤ i.val) (hij : i < j) : h ≤ lam i - lam j

theorem **gap_le_bv** (hk : 1 ≤ k) (hs : IsShape k r lam bv) (hg : GapAtLeast (sgnCN k r) lam h) : h ≤ bv

theorem **IsRange.radius_le** (hr : IsRange k r lam c h q) (hq : 0 ≤ q) : 100 * q ≤ h

theorem **IsRange.mono** {q' : ℝ} (hr : IsRange k r lam c h q') (hq : q ≤ q') : IsRange k r lam c h q

theorem **Setup.mono** {q' : ℝ} (hS : Setup k r c Ch P lam bv h q' R) (hq : q ≤ q') : Setup k r c Ch P lam bv h q R

theorem **Setup.standing** {q' : ℝ} (hS : Setup k r c Ch P lam bv h q' R) (hk : 1 ≤ k) (hCh : 915 ≤ Ch) (hCg : 1 ≤ Cg) (hC₀ : 10 ^ 4 ≤ C₀) (hq : 0 ≤ q) (h1 : q + r + 2 ≤ q') (h2 : Cg * (2 * r + 2 : ℝ) ^ 3 ≤ 100 * q') (h3 : C₀ * ((2 * r + k + 1 : ℕ) : ℝ) ^ 7 ≤ bv) : Standing k r c Ch Cg C₀ P lam bv h q R

theorem **setup** (H : Standing k r c Ch Cg C₀ P lam bv h q R) : Setup k r c Ch P lam bv h q R

theorem **mono** {q' : ℝ} (H : Standing k r c Ch Cg C₀ P lam bv h q' R) (h0 : 0 ≤ q) (hq : q ≤ q') : Standing k r c Ch Cg C₀ P lam bv h q R

theorem **inPP_self** (H : Standing k r c Ch Cg C₀ P lam bv h q R) : InPP lam q lam

theorem **radius_le_gap** (H : Standing k r c Ch Cg C₀ P lam bv h q R) {i j : Fin (2 * r + k + 1)} (hij : i < j) : 100 * q ≤ lam i - lam j

theorem **radius_le_bv** (H : Standing k r c Ch Cg C₀ P lam bv h q R) : 100 * q ≤ h ∧ h ≤ bv

theorem **cube_le_bv** (H : Standing k r c Ch Cg C₀ P lam bv h q R) : 10 ^ 4 * ((2 * r + k + 1 : ℕ) : ℝ) ^ 3 ≤ bv

theorem **gap_ge** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam q μ) {i j : Fin (2 * r + k + 1)} (hij : i < j) : 0.98 * (lam i - lam j) ≤ μ i - μ j

theorem **four_le_gap** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam (q + r + 2) μ) {i j : Fin (2 * r + k + 1)} (hij : i < j) : 4 ≤ μ i - μ j

theorem **isHCParam** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam (q + r + 2) μ) : IsHCParam μ

theorem **isDSAt_add** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam q μ) (dual : Bool) {l : ℕ} (hl : l ≤ r + 1) (I : Idx k r l) : IsHCParam (μ + wtI dual I) ∧ (R (μ + wtI dual I)).IsDSAt P (μ + wtI dual I)

theorem **gapAtLeast** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam q μ) : GapAtLeast (sgnCN k r) μ (0.98 * h)

theorem **entries** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam q μ) (i : Fin (2 * r + k + 1)) : (i.val < k - 1 → -q ≤ μ i ∧ μ i ≤ bv / 8 + q) ∧ (k - 1 ≤ i.val → -(2 * bv) - q ≤ μ i ∧ μ i ≤ -bv + q)

theorem **bStar_le** (H : Standing k r c Ch Cg C₀ P lam bv h q R) (hμ : InPP lam q μ) : bStar μ ≤ 2.02 * bv

theorem **HypJ.QQ_le** {j : ℕ} (hJ : HypJ k r bv j) : QQ r bv ≤ (6 * bv) ^ (2 * (2 * r + 2))

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.RowsMeanC

theorem **fv_cast** (F : FlagG S r T) (a : Fin r) : F.fvec a.castSucc = F.e a

theorem **fv_last** (F : FlagG S r T) : F.fvec (Fin.last r) = F.z

theorem **tu_ne** (D : D7) (u : D.what₂.Word) : ((Tu D u : ℕ) : D.host.E) ≠ 0

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.SecondLabelsNonempty

theorem **SplitLab.of_one** {R : Ranks} {H : Host R} {x eps : ℝ} (h : ⌊x ^ (eps / 5)⌋₊ ≤ H.splitBound) (ho : (R.o : ℝ) ≤ x ^ (eps / 100)) (hQ : (H.QS : ℝ) ≤ 4 * R.n * x ^ (eps / 100)) : SplitLab H x eps

theorem **labels_iff** (x eps CG : ℝ) : Labels D x eps CG ↔ DeadLabels D.host ∧ LiveLabels D x eps CG

theorem **exists_deadLabels** {R : Ranks} (H : Host R) : ∃ (Lw : ℕ) (h : 2 ≤ Lw), DeadLabels (relabel H Lw h)

theorem **TopsCoset.succ_le** {R : Ranks} {H : Host R} {J : ℕ} (h : TopsCoset H J) (hm : 2 * R.m ≤ 2 ^ R.e) : J + 1 ≤ R.e

theorem **TopsCoset.card_mul** {R : Ranks} {H : Host R} {J : ℕ} (h : TopsCoset H J) (hJ : J ≤ R.e) : H.TT.card * 2 ^ J = R.n

theorem **TopsCoset.card_eq** {R : Ranks} {H : Host R} {J : ℕ} (h : TopsCoset H J) (hJ : J ≤ R.e) : H.TT.card = R.n / 2 ^ J

theorem **TopsCoset.card_even** {R : Ranks} {H : Host R} {J : ℕ} (h : TopsCoset H J) (hm : 2 * R.m ≤ 2 ^ R.e) : Even H.TT.card

theorem **two_m_le_two_pow** {R : Ranks} (H : Host R) (h : (R.m : ℝ) ^ 4 ≤ R.f) : 2 * R.m ≤ 2 ^ R.e

theorem **topsCoset_withTops** {R : Ranks} (H : Host R) (c : R.A) : TopsCoset (withTops H (Nat.log 2 R.m) c) (Nat.log 2 R.m) where

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.SerFixOne

theorem **serFix_one** (π : Formalism.L2.NG.FinTyp L n) (s : SplitPlace L) {g : ℕ} : SerFix π s (1 : MonoidAlgebra ℂ (GL (Fin n) (ZMod g)))

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.SmallHostRootDiscriminantLe

theorem **sqrt_bx** (x eps : ℝ) : Real.sqrt (bx x eps) = Real.exp (x ^ eps / 2)

theorem **Num.facts** {R : Ranks} {x eps : ℝ} {Del h : ℕ} (N : Num R x eps Del h) : Real.exp (x ^ eps / 2) ≤ bpC x eps ∧ bpC x eps ≤ aC x eps ∧ Real.exp (x ^ eps / 2) ≤ bx x eps ∧ bx x eps ≤ aC x eps ∧ 4 * aC x eps ≤ bx x eps ^ 2 ∧ 2 * (R.k : ℝ) ≤ bpC x eps / x ∧ bpC x eps / x ≤ bpC x eps ∧ (5 : ℝ) ≤ R.k ∧ 2 * (R.k : ℝ) ≤ Real.exp (x ^ eps / 2)

theorem **half_k** (R : Ranks) : ∃ z : ℤ, ((R.k : ℚ) - 1) / 2 = z + 1 / 2

theorem **place_rows** (D : D7) (N : Num D.R x eps Del h) (hDel : D.s.Del = Del) (hT : ∀ τ, Top D τ → D.hc τ = hcTop N) (hO : ∀ τ, ¬ Top D τ → D.hc τ = hcOff N) : OneScale D x eps (2 * (D.R.m : ℝ) * h) ∧ Nodes D x eps ∧ (∀ τ (j : ℕ) (hj : j + 1 < D.R.m), (((D.hc τ).b (j + 1, hj) - (D.hc τ).b (j, by omega) : ℚ) : ℝ) = 2 * h) ∧ (∀ τ (i : Fin (D.R.k - 1)), 0 ≤ (((D.hc τ).v i : ℚ) : ℝ) ∧ (((D.hc τ).v i : ℚ) : ℝ) ≤ 4 * (D.R.k : ℝ) * (D.s.Del : ℝ)) ∧ (∀ τ, ¬ Top D τ → 4 * (D.R.k : ℝ) * (D.s.Del : ℝ) ≤ uFirst D τ - vFirst D τ ∧ uFirst D τ - vFirst D τ ≤ 8 * (D.R.k : ℝ) * (D.s.Del : ℝ) ∧ uLast D τ = vLast D τ - 1 / 2)

theorem **rcd_rows** (H : Host R) (A : Arith H) (G : LocalGap H) (N : Num R x eps Del h) (Gam : ℝ) (hG : 1 < Gam) : (∀ τ, Top (rcd H A G N Gam hG) τ → (rcd H A G N Gam hG).hc τ = hcTop (R

theorem **card_top** (D : D7) : (Finset.univ.filter (fun τ => Top D τ)).card = D.host.TT.card

theorem **even_of_const** (D : D7) (A B : HCParam D.R) (hT : ∀ τ, Top D τ → D.hc τ = A) (hO : ∀ τ, ¬ Top D τ → D.hc τ = B) (hev : Even D.host.TT.card) : EvenCentres D

theorem **even_card** {H : Host R} {J : ℕ} (h : TopsCoset H J) (hm : 2 * R.m ≤ 2 ^ R.e) : Even H.TT.card

theorem **rcd_datum** (H : Host R) (A : Arith H) (G : LocalGap H) (N : Num R x eps Del h) (Gam : ℝ) (hG : 1 < Gam) {J : ℕ} (hJ : TopsCoset H J) (hm : 2 * R.m ≤ 2 ^ R.e) : OneScale (rcd H A G N Gam hG) x eps (2 * (R.m : ℝ) * h) ∧ Nodes (rcd H A G N Gam hG) x eps ∧ EvenCentres (rcd H A G N Gam hG) ∧ DelInt (rcd H A G N Gam hG)

theorem **charpoly_eq_of_isIso** (n : ℕ) : P n = 2 ^ n

theorem **natCast_two_pow** (n : ℕ) : ((2 ^ n : ℕ) : ℝ) = P n

theorem **twoPow_pos** (n : ℕ) : 0 < P n

theorem **twoPow_mul_twoPow** (a b : ℕ) : P a * P b = P (a + b)

theorem **twoPow_pow** (a n : ℕ) : P a ^ n = P (a * n)

theorem **P_le** {a b : ℕ} (h : a ≤ b) : P a ≤ P b

theorem **twoPow_rpow** (N : ℕ) (c : ℝ) : P N ^ c = (2 : ℝ) ^ ((N : ℝ) * c)

theorem **twoPow_rpow_of_mul_eq** {N n : ℕ} {c : ℝ} (h : (N : ℝ) * c = n) : P N ^ c = P n

theorem **twoPow_rpow_le** {N n : ℕ} {c : ℝ} (h : (N : ℝ) * c ≤ n) : P N ^ c ≤ P n

theorem **twoPow_le_rpow** {N n : ℕ} {c : ℝ} (h : (n : ℝ) ≤ (N : ℝ) * c) : P n ≤ P N ^ c

theorem **twoPow_le_exp** {n : ℕ} {y : ℝ} (h : (n : ℝ) ≤ y) : P n ≤ Real.exp y

theorem **log_le_of_le_P** {n : ℕ} {y : ℝ} (hy : 0 < y) (h : y ≤ P n) : Real.log y ≤ n

theorem **one_le_P** (n : ℕ) : 1 ≤ P n

theorem **L_small** : P 2800 ^ ((1 : ℝ) / 200) = 16384

theorem **num_small** : Num R0 (P 2800) (1 / 200) 1 9 where

theorem **exists_not_top** (D : D7) : ∃ τ, ¬ Top D τ

theorem **exists_top** (D : D7) (h : D.host.TT.Nonempty) : ∃ τ, Top D τ

theorem **gam_gt** {CG y : ℝ} (R : Ranks) (hC : 0 < CG) (hy : 0 < y) : 1 < Real.exp (CG * (R.m : ℝ) ^ 5 * R.f * y)

theorem **small_host_tops_coset** : TopsCoset H1 1 where

theorem **small_host_dead_labels** : DeadLabels H1

theorem **floor_le_six** {y : ℝ} (hy : 0 ≤ y) (h : y ^ 5 ≤ 16384) : ⌊y⌋₊ ≤ 6

private theorem **small_host_split_clause** {x eps : ℝ} (hx : 1 ≤ x) (he : 0 ≤ eps) (h : ⌊x ^ (eps / 5)⌋₊ ≤ 6) : SplitLab H1 x eps

theorem **small_witness** (D0 : D7) (CG : ℝ) (hC : 0 < CG) : ∃ D : D7, D.host.TT.card = 2 ∧ D.R.f = 8 ∧ OneScale D (P 2800) (1 / 200) 36 ∧ Nodes D (P 2800) (1 / 200) ∧ Labels D (P 2800) (1 / 200) CG ∧ TopsCoset D.host 1 ∧ EvenCentres D ∧ DelInt D

theorem **L_coarse** : P 1 ^ (12 : ℝ) = 4096

theorem **X_coarse** : P 1 ^ ((12 : ℝ) / 3) = 16

theorem **Y_coarse** : P 1 ^ ((11 : ℝ) / 2) ≤ 64

theorem **le_exp_of_le** {y z : ℝ} (n : ℕ) (h1 : y ≤ 2 ^ n) (h2 : (n : ℝ) ≤ z) : y ≤ Real.exp z

theorem **num_coarse** : Num R0 (P 1) 12 1 9 where

theorem **small_host_root_discriminant_le** : H1.v ≤ 4096

theorem **a_thresholds** {x eps v : ℝ} (h1 : x ^ ((11 : ℝ) / 2) ≤ bx x eps) (h2 : x ^ ((11 : ℝ) / 2) * v ≤ bx x eps) : aC x eps ≤ bx x eps ^ 2 ∧ aC x eps * v ≤ bx x eps ^ 2

theorem **thresh_coarse** {v : ℝ} (hv : v ≤ 4096) : ThreshNums (P 1) 12 8 8 40 32 36 v 599

theorem **coarse_witness** (D0 : D7) (CG : ℝ) (hC : 0 < CG) : ∃ D : D7, D.host.TT.card = 2 ∧ D.R.f = 8 ∧ OneScaleDatum D (P 1) 12 36 1 ∧ Nodes D (P 1) 12 ∧ Labels D (P 1) 12 CG

theorem **big_nums** (R : Ranks) (Del h : ℕ) (hm : (R.m : ℝ) = P 119925) (hf : (R.f : ℝ) = P 489601) (hk : (R.k : ℝ) = P 1092000) (ht : R.t = 4 * R.k + 1) (hD : (Del : ℝ) = P 32000000) (hh : (h : ℝ) = P 24000000) : Num R (P 240000) (1 / 200) Del h ∧ (∀ lrd, Real.log 2 ≤ lrd → lrd ≤ P 600 → RankNums (P 240000) (1 / 200) 20 R.m R.f R.k lrd Del) ∧ P 240000 ^ ((1 : ℝ) / 200 / 100) * (R.r : ℝ) ^ 19 * (R.f : ℝ) ^ 5 * P 240000 ^ ((1 : ℝ) / 200) ≤ 2 * h

theorem **big_ranks** : ∃ (R : Ranks) (Del h : ℕ), (R.m : ℝ) = P 119925 ∧ (R.f : ℝ) = P 489601 ∧ (R.k : ℝ) = P 1092000 ∧ R.t = 4 * R.k + 1 ∧ (Del : ℝ) = P 32000000 ∧ (h : ℝ) = P 24000000

theorem **big_thresh** (R : Ranks) (Del h : ℕ) (hm : (R.m : ℝ) = P 119925) (hf : (R.f : ℝ) = P 489601) (hk : (R.k : ℝ) = P 1092000) (ht : R.t = 4 * R.k + 1) (hD : (Del : ℝ) = P 32000000) (hh : (h : ℝ) = P 24000000) {v d : ℝ} (hv : v ≤ Real.exp (P 600)) (hd : 2 * P 1320000 < d) : ThreshNums (P 240000) (1 / 200) R.f R.k (R.N R.m) (4 * (R.k : ℝ) * Del) (2 * (R.m : ℝ) * h) v d

theorem **big_witness** (D0 : D7) (R : Ranks) (Del h : ℕ) (hm : (R.m : ℝ) = P 119925) (hf : (R.f : ℝ) = P 489601) (hk : (R.k : ℝ) = P 1092000) (ht : R.t = 4 * R.k + 1) (hD : (Del : ℝ) = P 32000000) (hh : (h : ℝ) = P 24000000) (H : Host R) (A : Arith H) (Cg CT CG : ℝ) (hC : 0 < CG) (J : ℕ) (hJ : TopsCoset H J) (r1 : 2 ≤ H.rdE) (r2 : Real.log H.rdE ≤ P 600) (hd : 2 * P 1320000 < H.d) (hs : SplitLab H (P 240000) (1 / 200)) (hg : ∀ i, ((R.k : ℝ) * bx (P 240000) (1 / 200)) ^ 8 ≤ (H.g i : ℝ) ∧ (H.g i : ℝ) ≤ bx (P 240000) (1 / 200) ^ Cg) (hT : bx (P 240000) (1 / 200) ^ ((1 / 2 : ℝ) * (R.m : ℝ)) ≤ (A.T : ℝ) ∧ (A.T : ℝ) ≤ bx (P 240000) (1 / 200) ^ (CT * (R.m : ℝ))) : ∃ D : D7, D.R = R ∧ HEq D.host H ∧ Glob D (P 240000) (1 / 200) 20 Cg CT ∧ Nodes D (P 240000) (1 / 200) ∧ OneScaleDatum D (P 240000) (1 / 200) (2 * (R.m : ℝ) * h) J ∧ LiveLabels D (P 240000) (1 / 200) CG

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.SubsingletonIdxZero

theorem **IsRange.gap_opp** {lam : Fin (2 * r + k + 1) → ℝ} {c h q : ℝ} (H : IsRange k r lam c h q) {a b : Fin (2 * r + k + 1)} (hab : sgnCN k r a ≠ sgnCN k r b) : 100 * q ≤ |lam a − lam b|

theorem **sgnCN_succ_ne** {a b : Fin (2 * r + k + 1)} (ha : k − 1 ≤ a.val) (hb : b.val = a.val + 1) : sgnCN k r a ≠ sgnCN k r b

theorem **IsRange.le_bv** {lam : Fin (2 * r + k + 1) → ℝ} {bv c h q : ℝ} (hk : 1 ≤ k) (S : IsShape k r lam bv) (H : IsRange k r lam c h q) : 100 * q ≤ bv

theorem **IsRange.le_gap** {lam : Fin (2 * r + k + 1) → ℝ} {bv c h q : ℝ} (S : IsShape k r lam bv) (H : IsRange k r lam c h q) {a b : Fin (2 * r + k + 1)} (hab : a < b) : 100 * q ≤ lam a − lam b

theorem **abs_add_abs_le_dist1** {n : ℕ} (μ lam : Fin n → ℝ) {a b : Fin n} (hab : a ≠ b) : |μ a − lam a| + |μ b − lam b| ≤ dist1 μ lam

theorem **InPP.gap_ge** {lam μ : Fin (2 * r + k + 1) → ℝ} {bv c h q : ℝ} (hμ : InPP lam q μ) (S : IsShape k r lam bv) (H : IsRange k r lam c h q) {a b : Fin (2 * r + k + 1)} (hab : a < b) : 99 / 100 * (lam a − lam b) ≤ μ a − μ b

theorem **InPP.isHCParam** {lam μ : Fin (2 * r + k + 1) → ℝ} {bv c h q : ℝ} (hμ : InPP lam q μ) (S : IsShape k r lam bv) (H : IsRange k r lam c h q) : IsHCParam μ

theorem **detRep_apply** (μ : Fin (2 * r + k + 1) → ℝ) (g : Mat k r) (v : ℂ) : (detRep k r μ).σ g v = g.det ^ ⌊μ 0⌋₊ * v

theorem **contDiff_entry** (i j : Fin (2 * r + k + 1)) : ContDiff ℝ ∞ (fun g : Mat k r => g i j)

theorem **contDiff_det** : ContDiff ℝ ∞ (fun g : Mat k r => g.det)

theorem **detRep_smooth** (μ : Fin (2 * r + k + 1) → ℝ) (v : ℂ) : v ∈ smoothVecs (k

theorem **detRep_dsigW** (μ : Fin (2 * r + k + 1) → ℝ) (w : List (Mat k r)) : ∃ c : ℂ, ∀ v : ℂ, dsigW (k

theorem **subsingleton_idx_top** : Subsingleton (Idx k r (2 * r + k + 1))

theorem **idx_univ** (I : Idx k r (2 * r + k + 1)) : I.1 = Finset.univ

theorem **minor_top** (M : Mat k r) (I : Idx k r (2 * r + k + 1)) : minorIJ M I I = M.det

theorem **isTensDec_det** (P : Mat k r) (hP : IsUnit P.det) (μ : Fin (2 * r + k + 1) → ℝ) (hμ : 0 ≤ μ 0) : IsTensDec (detRep k r) false P (2 * r + k + 1) μ (fun _ y _ => y) (fun I x => x I) where

theorem **isHHNorm_det** (μ : Fin (2 * r + k + 1) → ℝ) : IsHHNorm (detRep k r) false (2 * r + k + 1) μ (fun _ y _ => y) epsI

theorem **idx_top_nonempty** : Nonempty (Idx k r (2 * r + k + 1))

theorem **subsingleton_idx_zero** : Subsingleton (Idx k r 0)

theorem **isTensDec_zero** (A : HRep k r) (dual : Bool) (P : Mat k r) (μ : Fin (2 * r + k + 1) → ℝ) : IsTensDec (fun _ => A) dual P 0 μ (fun _ y _ => y) (fun I x => x I)

theorem **shape_range** : IsShape 1 0 ![-(1001/2), -(1999/2)] 500 ∧ IsRange 1 0 ![-(1001/2), -(1999/2)] 1 499 3

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.SymmetricFlagBasisVectorsApplyCentre

@[simp] theorem **ofHSpace_B** {E : Type} [Field E] [NumberField E] [StarRing E] (S : Formalism.BasicNotions.HSpace E) (x y : S.V) : (ofHSpace S).B x y = S.hT y x

theorem **isUnimod_ofHSpace** {E : Type} [Field E] [NumberField E] [StarRing E] (S : Formalism.BasicNotions.HSpace E) (Λ : Submodule (NumberField.RingOfIntegers E) S.V) : IsUnimod (ofHSpace S) Λ ↔ Submodule.IsLattice E Λ ∧ Formalism.HermitianLatticesTheory.IsUnimodular S.hT Λ

theorem **B_eq_zero_comm** {x y : V} : H.B x y = 0 ↔ H.B y x = 0

theorem **perp_anti** {S S' : Submodule K V} (h : S ≤ S') : perp H S' ≤ perp H S

theorem **perp_sup** (S S' : Submodule K V) : perp H (S ⊔ S') = perp H S ⊓ perp H S'

theorem **mem_perp_span** (s : Set V) (y : V) : y ∈ perp H (Submodule.span K s) ↔ ∀ x ∈ s, H.B y x = 0

theorem **X_eq_span** (F : FlagD H r) : F.X = Submodule.span K (Set.range fun a : Fin r => F.e a.castSucc)

theorem **mem_perp_X** (F : FlagD H r) (y : V) : y ∈ perp H F.X ↔ ∀ a : Fin r, H.B y (F.e a.castSucc) = 0

theorem **X_le_perp** (F : FlagD H r) : F.X ≤ perp H F.X

theorem **z_mem_perp** (F : FlagD H r) : F.z ∈ perp H F.X

theorem **form_flag_vector_self** (F : FlagD H r) : H.B F.z F.z = -F.T

theorem **X_le_xPlus** (F : FlagD H r) : F.X ≤ xPlus F

theorem **z_mem_xPlus** (F : FlagD H r) : F.z ∈ xPlus F

theorem **xPlus_eq_span** (F : FlagD H r) : xPlus F = Submodule.span K (Set.range F.e)

theorem **mem_perp_xPlus** (F : FlagD H r) (y : V) : y ∈ perp H (xPlus F) ↔ y ∈ perp H F.X ∧ H.B y F.z = 0

theorem **X_le_perp_xPlus** (F : FlagD H r) : F.X ≤ perp H (xPlus F)

theorem **eq_zero_of_smul_z_mem** (F : FlagD H r) {a : K} (h : a • F.z ∈ F.X) : a = 0

theorem **B_congr** (F : FlagD H r) {x x₁ y y₁ : V} (hx : x ∈ perp H F.X) (hy : y ∈ perp H F.X) (h1 : x₁ - x ∈ F.X) (h2 : y₁ - y ∈ F.X) : H.B x₁ y₁ = H.B x y

theorem **mem_smulLat** (g : R) (Λ : Submodule R V) (x : V) : x ∈ smulLat g Λ ↔ ∃ y ∈ Λ, g • y = x

theorem **gL_eq_smulLat** {E : Type} [Field E] [NumberField E] [StarRing E] {S : Formalism.BasicNotions.HSpace E} (Λ : Submodule (NumberField.RingOfIntegers E) S.V) (g : ℕ) : gL Λ g = smulLat (g : NumberField.RingOfIntegers E) Λ

theorem **mem_imLat** (s : V →ₗ[K] V) (M : Submodule R V) (x : V) : x ∈ imLat s M ↔ ∃ y ∈ M, s y = x

theorem **mem_latX** (F : FlagD H r) (Λ : Submodule R V) (x : V) : x ∈ latX F Λ ↔ x ∈ Λ ∧ x ∈ F.X

theorem **mem_redLat** (F : FlagD H r) (Λ : Submodule R V) (x : V) : x ∈ redLat F Λ ↔ x ∈ perp H F.X ∧ ∃ a ∈ F.X, x + a ∈ Λ

theorem **xPlus_sup** (F F' : FlagD H r) : xPlus F ⊔ xPlus F' = V0 F F'

theorem **jointSpan_comm** (F F' : FlagD H r) : V0 F' F = V0 F F'

theorem **wSp_comm** (F F' : FlagD H r) : wSp F' F = wSp F F'

theorem **wSp_eq** (F F' : FlagD H r) : wSp F F' = perp H (xPlus F) ⊓ perp H (xPlus F')

theorem **mem_norLift** (F F' : FlagD H r) (y : V) : y ∈ norLift F F' ↔ y ∈ V0 F F' ∧ y ∈ perp H (xPlus F)

theorem **X_le_norLift** (F F' : FlagD H r) : F.X ≤ norLift F F'

theorem **norLift_inf** (F F' : FlagD H r) : norLift F F' ⊓ norLift F' F = V0 F F' ⊓ wSp F F'

theorem **IsTransv.symm** {F F' : FlagD H r} (h : IsTransv F F') : IsTransv F' F

theorem **IsTransv.disjoint_norLift** {F F' : FlagD H r} (h : IsTransv F F') : Disjoint (norLift F F') (norLift F' F)

theorem **IsTransv.disjoint_xPlus** {F F' : FlagD H r} (h : IsTransv F F') : Disjoint (xPlus F) (xPlus F')

theorem **isTransv_iff_gram** (F F' : FlagD H r) : IsTransv F F' ↔ (Gram F F').det ≠ 0

theorem **exists_isComp_iff** (F F' : FlagD H r) (u : V) : (∃ a a', IsComp F F' u a a') ↔ u ∈ V0 F F'

theorem **isComp_iff** (F F' : FlagD H r) (u : V) (a a' : K) : IsComp F F' u a a' ↔ ∃ ξ ∈ F.X, ∃ ξ' ∈ F'.X, u = ξ + a • F.z + ξ' + a' • F'.z

theorem **IsComp.swap** {F F' : FlagD H r} {u : V} {a a' : K} (h : IsComp F F' u a a') : IsComp F' F u a' a

theorem **isComp_zero_iff** (F F' : FlagD H r) (u : V) : IsComp F F' u 0 0 ↔ u ∈ F.X ⊔ F'.X

theorem **IsComp.add** {F F' : FlagD H r} {u v : V} {a a' b b' : K} (h : IsComp F F' u a a') (k : IsComp F F' v b b') : IsComp F F' (u + v) (a + b) (a' + b')

theorem **IsComp.smul** {F F' : FlagD H r} {u : V} {a a' : K} (h : IsComp F F' u a a') (c : K) : IsComp F F' (c • u) (c * a) (c * a')

theorem **IsComp.sub** {F F' : FlagD H r} {u v : V} {a a' b b' : K} (h : IsComp F F' u a a') (k : IsComp F F' v b b') : IsComp F F' (u - v) (a - b) (a' - b')

theorem **IsComp.congr** {F F' : FlagD H r} {u v : V} {a a' : K} (h : IsComp F F' u a a') (k : v - u ∈ F.X ⊔ F'.X) : IsComp F F' v a a'

theorem **IsComp.unique** {F F' : FlagD H r} (hd : Disjoint (xPlus F) (xPlus F')) {u : V} {a a' b b' : K} (h : IsComp F F' u a a') (k : IsComp F F' u b b') : a = b ∧ a' = b'

theorem **exists_isSig** {F F' : FlagD H r} {y y' : V} (hy : y ∈ norLift F F') (hy' : y' ∈ norLift F' F) : ∃ a a', IsSig F F' y y' a a'

theorem **IsSig.congr** {F F' : FlagD H r} {y y' y₁ y₁' : V} {a a' : K} (h : IsSig F F' y y' a a') (h1 : y₁ - y ∈ F.X) (h2 : y₁' - y' ∈ F'.X) : IsSig F F' y₁ y₁' a a'

theorem **IsSig.unique** {F F' : FlagD H r} (hd : Disjoint (xPlus F) (xPlus F')) {y y' : V} {a a' b b' : K} (h : IsSig F F' y y' a a') (k : IsSig F F' y y' b b') : a = b ∧ a' = b'

theorem **IsSig.add** {F F' : FlagD H r} {y y' w w' : V} {a a' b b' : K} (h : IsSig F F' y y' a a') (k : IsSig F F' w w' b b') : IsSig F F' (y + w) (y' + w') (a + b) (a' + b')

theorem **IsSig.smul** {F F' : FlagD H r} {y y' : V} {a a' : K} (h : IsSig F F' y y' a a') (c : K) : IsSig F F' (c • y) (c • y') (c * a) (c * a')

theorem **IsSig.swap** {F F' : FlagD H r} {y y' : V} {a a' : K} (h : IsSig F F' y y' a a') : IsSig F' F y' y (-a') (-a)

theorem **IsCong.refl** (F : FlagD H r) (Λ : Submodule R V) (g : R) : IsCong F F Λ g

theorem **IsCong.symm** {F F' : FlagD H r} {Λ : Submodule R V} {g : R} (h : IsCong F F' Λ g) : IsCong F' F Λ g

theorem **IsCong.trans** {F F' F'' : FlagD H r} {Λ : Submodule R V} {g : R} (h : IsCong F F' Λ g) (k : IsCong F' F'' Λ g) : IsCong F F'' Λ g

theorem **IsPairMap.mem_X_iff** (h : IsPairMap F F' F'' Λ g β) (x : V) : β x ∈ F''.X ↔ x ∈ F'.X

theorem **IsPairMap.mem_perp_iff** (h : IsPairMap F F' F'' Λ g β) (x : V) : β x ∈ perp H F''.X ↔ x ∈ perp H F'.X

theorem **IsPairMap.mem_redLat_iff** (h : IsPairMap F F' F'' Λ g β) (x : V) : β x ∈ redLat F'' Λ ↔ x ∈ redLat F' Λ

theorem **isPairMap_of_mem_iff** (h1 : ∀ x, β x ∈ F''.X ↔ x ∈ F'.X) (h2 : ∀ x, β x ∈ perp H F''.X ↔ x ∈ perp H F'.X) (h3 : ∀ x ∈ perp H F'.X, ∀ y ∈ perp H F'.X, H.B (β x) (β y) = H.B x y) (h4 : ∀ x, β x ∈ redLat F'' Λ ↔ x ∈ redLat F' Λ) (h5 : β F'.z - F''.z ∈ F''.X) (h6 : ∀ l' ∈ Λ ⊓ rs R (perp H F'.X), ∀ l'' ∈ Λ ⊓ rs R (perp H F''.X), β l' - l'' ∈ F''.X → l'' - l' ∈ latX F Λ ⊔ smulLat g Λ) : IsPairMap F F' F'' Λ g β

theorem **IsPairMap.symm** (h : IsPairMap F F' F'' Λ g β) : IsPairMap F F'' F' Λ g β.symm

theorem **IsPairMap.congr** (h : IsPairMap F F' F'' Λ g β) {β₁ : V ≃ₗ[K] V} (h1 : F'.X.map (β₁ : V →ₗ[K] V) = F''.X) (h2 : (perp H F'.X).map (β₁ : V →ₗ[K] V) = perp H F''.X) (he : ∀ x ∈ perp H F'.X, β₁ x - β x ∈ F''.X) : IsPairMap F F' F'' Λ g β₁

theorem **isPairMap_of_aut** (hc : IsCong F F'' Λ g) (hβ : β ∈ isomGrp H) (hΛ : imLat β Λ = Λ) (hX : F'.X.map (β : V →ₗ[K] V) = F''.X) (hz : β F'.z - F''.z ∈ F''.X) (hι : ∀ l' ∈ Λ ⊓ rs R (perp H F'.X), β l' - l' ∈ latX F Λ ⊔ smulLat g Λ) : IsPairMap F F' F'' Λ g β

theorem **isPairMap_refl** (hc : IsCong F F' Λ g) : IsPairMap F F' F' Λ g (LinearEquiv.refl K V)

theorem **fixSp_le** (F F' F'' : FlagD H r) (β : V ≃ₗ[K] V) : fixSp F F' F'' β ≤ wSp F F'

theorem **fixSp_congr** {β₁ : V ≃ₗ[K] V} (he : ∀ x ∈ perp H F'.X, β₁ x - β x ∈ F''.X) : fixSp F F' F'' β₁ = fixSp F F' F'' β

theorem **IsPairMap.fixSp_sup_X** (h : IsPairMap F F' F'' Λ g β) : fixSp F F'' F' β.symm ⊔ F.X = fixSp F F' F'' β ⊔ F.X

theorem **sp_B** (x y : (wittSpace E 2 1).V) : (sp E).B x y = -(star (co E y 0) * co E x 0) - star (co E y 1) * co E x 1 + star (co E y 2) * co E x 2 + star (co E y 3) * co E x 3 + star (co E y 4) * co E x 4

theorem **sp_B_vec** (v w : Fin 5 → ℤ) : (sp E).B (vec E v) (vec E w) = ((-(w 0 * v 0) - w 1 * v 1 + w 2 * v 2 + w 3 * v 3 + w 4 * v 4 : ℤ) : E)

theorem **mem_lat** (x : (wittSpace E 2 1).V) : x ∈ (wittSpace E 2 1).L0 ↔ ∀ i, co E x i ∈ Set.range (algebraMap (𝒪 E) E)

theorem **vec_mem** (v : Fin 5 → ℤ) : vec E v ∈ (wittSpace E 2 1).L0

theorem **fl_X** : (fl E m s h1 h2).X = Submodule.span E {vec E ![1 + 2 * m, 0, 1, 2 * s, 2 * m]}

theorem **symmetricFlagBasisVectors_apply_centre** : (fl E m s h1 h2).z = vec E ![0, 1 + 2 * m, 0, 2 * s, -2 * m]

theorem **fl_z_mem** : (fl E m s h1 h2).z ∈ (wittSpace E 2 1).L0

theorem **fl_latX** : latX (fl E m s h1 h2) (wittSpace E 2 1).L0 = Submodule.span (𝒪 E) {vec E ![1 + 2 * m, 0, 1, 2 * s, 2 * m]}

theorem **vec_sub_mem** (v w u : Fin 5 → ℤ) (h : ∀ i, v i - w i = 2 * u i) : vec E v - vec E w ∈ smulLat ((2 : ℕ) : 𝒪 E) (wittSpace E 2 1).L0

theorem **isCong_fl** : IsCong (flZero E) (fl E m s h1 h2) (wittSpace E 2 1).L0 ((2 : ℕ) : 𝒪 E)

theorem **gval_fl** {s : ℤ} (h2 : s * s = 1) : Gval (flZero E) (fl E 1 s rfl h2) = 32

theorem **isTransv_fl** {s : ℤ} (h2 : s * s = 1) : IsTransv (flZero E) (fl E 1 s rfl h2)

theorem **co_flip** (x : (wittSpace E 2 1).V) (i : Fin 5) : co E (flip E x) i = if i = 3 then -co E x i else co E x i

theorem **flip_vec** (a b c d e : ℤ) : flip E (vec E ![a, b, c, d, e]) = vec E ![a, b, c, -d, e]

theorem **isPairMap_flip** : IsPairMap (flZero E) (flPos E) (flNeg E) (wittSpace E 2 1).L0 ((2 : ℕ) : 𝒪 E) (flip E)

theorem **X_ne** : (flPos E).X ≠ (flNeg E).X

theorem **exists_pair** : ∃ (F F' F'' : FlagD (ofHSpace (wittSpace E 2 1)) 1) (β : (wittSpace E 2 1).V ≃ₗ[E] (wittSpace E 2 1).V), IsUnimod (ofHSpace (wittSpace E 2 1)) (wittSpace E 2 1).L0 ∧ F.z ∈ (wittSpace E 2 1).L0 ∧ F'.z ∈ (wittSpace E 2 1).L0 ∧ F''.z ∈ (wittSpace E 2 1).L0 ∧ F.T = 1 ∧ F'.T = 1 ∧ F''.T = 1 ∧ IsCong F F' (wittSpace E 2 1).L0 ((2 : ℕ) : 𝒪 E) ∧ IsCong F F'' (wittSpace E 2 1).L0 ((2 : ℕ) : 𝒪 E) ∧ Gval F F' ≠ 0 ∧ Gval F F'' ≠ 0 ∧ IsTransv F F' ∧ IsTransv F F'' ∧ F'.X ≠ F''.X ∧ IsPairMap F F' F'' (wittSpace E 2 1).L0 ((2 : ℕ) : 𝒪 E) β

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.TupOfOneMemLabT

theorem **tupOf_apply** (a : ∀ kv : (Σ k : Fin 2, D.nd.PlG k), (Fin D.R.r → ZMod (D.host.g kv.1))) (k : Fin 2) (v : D.nd.PlG k) : tupOf D a k v = Formalism.CyclotomicNorms.monicOf (a ⟨k, v⟩)

theorem **tupOf_injective** : Function.Injective (tupOf D)

theorem **mem_range_tupOf** {t : Tup2 D} : t ∈ Set.range (tupOf D) ↔ ∀ k v, (t k v).Monic ∧ (t k v).natDegree = D.R.r

theorem **mem_labT** {t : Tup2 D} : t ∈ labT D ↔ ∀ k v, IsLabel D k (t k v)

theorem **zTup_coeff** (t : Tup2 D) (g : G0g D.nd) : (zTup D t).coeff g = ∏ kv : (Σ k : Fin 2, D.nd.PlG k), (D.ntypes.eSer kv.1 (t kv.1 kv.2)).coeff (g kv)

theorem **zTup_eq_zero** {t : Tup2 D} (h : t ∉ labT D) : zTup D t = 0

theorem **zTup_mul_self** (t : Tup2 D) : zTup D t * zTup D t = zTup D t

theorem **zTup_mul_of_ne** {t t' : Tup2 D} (h : t ≠ t') : zTup D t * zTup D t' = 0

theorem **zTup_comm** (t : Tup2 D) (a : MonoidAlgebra ℂ (G0g D.nd)) : zTup D t * a = a * zTup D t

theorem **ES_coeff** (S : Set (Tup2 D)) (g : G0g D.nd) : (ES D S).coeff g = ∑ a : (∀ kv : (Σ k : Fin 2, D.nd.PlG k), (Fin D.R.r → ZMod (D.host.g kv.1))), if tupOf D a ∈ S then ∏ kv : (Σ k : Fin 2, D.nd.PlG k), (D.ntypes.eSer kv.1 (Formalism.CyclotomicNorms.monicOf (a kv))).coeff (g kv) else 0

theorem **ES_good** : ES D {t | D.ntypes.goodTuple t} = D.ntypes.e0

theorem **ES_eq_sum_coef** (S : Set (Tup2 D)) : ES D S = ∑ a : (∀ kv : (Σ k : Fin 2, D.nd.PlG k), (Fin D.R.r → ZMod (D.host.g kv.1))), if tupOf D a ∈ S then zTup D (tupOf D a) else 0

theorem **ES_eq_sum** (S : Set (Tup2 D)) : ES D S = ∑ t ∈ (labT D).filter (· ∈ S), zTup D t

theorem **ES_eq_finsum** (S : Set (Tup2 D)) : ES D S = ∑ᶠ t ∈ S, zTup D t

theorem **ES_eq_zSet** (S : Set (Tup2 D)) : ES D S = zSet (ι

theorem **ES_coe** (S : Finset (Tup2 D)) : ES D (S : Set (Tup2 D)) = ∑ t ∈ S, zTup D t

theorem **ES_singleton** (t : Tup2 D) : ES D {t} = zTup D t

theorem **ES_empty** : ES D ∅ = 0

theorem **ES_congr** {S T : Set (Tup2 D)} (h : ∀ t ∈ labT D, t ∈ S ↔ t ∈ T) : ES D S = ES D T

theorem **ES_inter_labT** (S : Set (Tup2 D)) : ES D (S ∩ labT D) = ES D S

theorem **ES_univ** : ES D Set.univ = 1

theorem **sum_zTup** : ∑ t ∈ labT D, zTup D t = 1

theorem **isSeries_zTup** : IsSeries (zTup D)

theorem **ES_mul_zTup** (S : Set (Tup2 D)) (t : Tup2 D) : ES D S * zTup D t = if t ∈ S then zTup D t else 0

theorem **ES_mul** (S T : Set (Tup2 D)) : ES D S * ES D T = ES D (S ∩ T)

theorem **ES_mul_self** (S : Set (Tup2 D)) : ES D S * ES D S = ES D S

theorem **ES_comm** (S : Set (Tup2 D)) (a : MonoidAlgebra ℂ (G0g D.nd)) : ES D S * a = a * ES D S

theorem **ES_mem_center** (S : Set (Tup2 D)) : ES D S ∈ Subalgebra.center ℂ (MonoidAlgebra ℂ (G0g D.nd))

theorem **ES_union_add_inter** (S T : Set (Tup2 D)) : ES D (S ∪ T) + ES D (S ∩ T) = ES D S + ES D T

theorem **ES_union** {S T : Set (Tup2 D)} (h : Disjoint S T) : ES D (S ∪ T) = ES D S + ES D T

theorem **ES_compl** (S : Set (Tup2 D)) : ES D Sᶜ = 1 - ES D S

theorem **ES_bad** : ES D {t | ¬ D.ntypes.goodTuple t} = 1 - D.ntypes.e0

theorem **ES_pi** (A : ∀ k, D.nd.PlG k → Finset (ZMod (D.host.g k))[X]) : ES D {t | ∀ k v, t k v ∈ A k v} = tensAlg (ι

theorem **ES_at** (k : Fin 2) (v : D.nd.PlG k) (f : (ZMod (D.host.g k))[X]) : ES D {t | t k v = f} = atPlace D k v (D.ntypes.eSer k f)

theorem **tupOf_one_mem_labT** : tupOf D (fun _ _ => 1) ∈ labT D

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.UniqGapOfUniq

theorem **uniqGap_of_uniq** (h : PI.BesselMapUniq.BesselMapUniq) : UniqGap

theorem **ampKF_iff** : AmpKF ↔ (Formalism.FourierDecayAlt.ThreeDimData.Inputs → AmpKF0)

Development.AbelianVarietyTheory.Definitions.SemilocalFrames.UnitriangularLatticeStabiliserSubFrameAutomorphi

theorem **gmap_one** (Λ : LV D) : Formalism.HermitianLatticesTheory.gmap (1 : GV D) Λ = Λ

theorem **gmap_mul** (a b : GV D) (Λ : LV D) : Formalism.HermitianLatticesTheory.gmap (a * b) Λ = Formalism.HermitianLatticesTheory.gmap a (Formalism.HermitianLatticesTheory.gmap b Λ)

theorem **adapt_one** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) : Adapt (1 : GV D) F F

theorem **adapt_mapLF** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) (g : GV D) (hg : g ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) : Adapt g F (mapLF g F)

theorem **adapt_zero** {T : D.host.E} {F G : LFlag (VS D) D.R.rS T} {g : GV D} (h : Adapt g F G) : g (frame F 0) = frame G 0

theorem **unitriangular_lattice_stabiliser_sub_frame_automorphisms** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) (Λ : LV D) : NLam Λ F ⊆ Aut0 F Λ

theorem **one_mem_stabL** (Λ : LV D) : (1 : GV D) ∈ stabL F F' Λ

theorem **mem_gmap** (h : GV D) (Λ : LV D) (x) : x ∈ Formalism.HermitianLatticesTheory.gmap h Λ ↔ h.symm x ∈ Λ

theorem **gmap_inf** (h : GV D) (A B : LV D) : Formalism.HermitianLatticesTheory.gmap h (A ⊓ B) = Formalism.HermitianLatticesTheory.gmap h A ⊓ Formalism.HermitianLatticesTheory.gmap h B

theorem **gmap_sup** (h : GV D) (A B : LV D) : Formalism.HermitianLatticesTheory.gmap h (A ⊔ B) = Formalism.HermitianLatticesTheory.gmap h A ⊔ Formalism.HermitianLatticesTheory.gmap h B

theorem **fix_X** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hf : ∀ a, h (frame F a) = frame F a) {x} (hx : x ∈ Xsp F) : h x = x

theorem **mem_X_iff** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hf : ∀ a, h (frame F a) = frame F a) (x) : h x ∈ Xsp F ↔ x ∈ Xsp F

theorem **mem_orth_iff** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hi : h ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) (hf : ∀ a, h (frame F a) = frame F a) (x) : h x ∈ orth (VS D) (Xsp F) ↔ x ∈ orth (VS D) (Xsp F)

theorem **gmap_X** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hf : ∀ a, h (frame F a) = frame F a) : Formalism.HermitianLatticesTheory.gmap h (toO (VS D) (Xsp F)) = toO (VS D) (Xsp F)

theorem **gmap_orth** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hi : h ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) (hf : ∀ a, h (frame F a) = frame F a) : Formalism.HermitianLatticesTheory.gmap h (toO (VS D) (orth (VS D) (Xsp F))) = toO (VS D) (orth (VS D) (Xsp F))

theorem **Lrep_gmap** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hi : h ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) (hf : ∀ a, h (frame F a) = frame F a) (Λ : LV D) : Lrep (Formalism.HermitianLatticesTheory.gmap h Λ) F = Formalism.HermitianLatticesTheory.gmap h (Lrep Λ F)

theorem **Lsub_gmap** {T : D.host.E} (F : LFlag (VS D) D.R.rS T) {h : GV D} (hi : h ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) (hf : ∀ a, h (frame F a) = frame F a) (Λ : LV D) : Lsub (Formalism.HermitianLatticesTheory.gmap h Λ) F = Formalism.HermitianLatticesTheory.gmap h (Lsub Λ F)

theorem **NgX_gmap** {h : GV D} (hh : h ∈ H0 F F') (Λ : LV D) (g : ℕ) : NgX (Formalism.HermitianLatticesTheory.gmap h Λ) g F F' = Formalism.HermitianLatticesTheory.gmap h (NgX Λ g F F')

theorem **gdat_gmap** {h : GV D} (hh : h ∈ H0 F F') (Λ : LV D) : gdat F F' (Formalism.HermitianLatticesTheory.gmap h Λ) = (Formalism.HermitianLatticesTheory.gmap h (gdat F F' Λ).1, Formalism.HermitianLatticesTheory.gmap h (gdat F F' Λ).2.1, {p | (h.symm p.1, h.symm p.2) ∈ (gdat F F' Λ).2.2})

theorem **gdat_congr** {h : GV D} (hh : h ∈ H0 F F') {Λ₁ Λ₂ : LV D} (e : gdat F F' Λ₁ = gdat F F' Λ₂) : gdat F F' (Formalism.HermitianLatticesTheory.gmap h Λ₁) = gdat F F' (Formalism.HermitianLatticesTheory.gmap h Λ₂)

theorem **lrel_equivalence** : Equivalence (LRel F F')

theorem **lrel_gmap** (Λ : LV D) {h : GV D} (hh : h ∈ H0 F F') : LRel F F' Λ (Formalism.HermitianLatticesTheory.gmap h Λ)

theorem **ext_unique** (p : PlF D) (g : GV D) (a b : Vp D p ≃ₗ[(D.s.atP p).Ep] Vp D p) (ha : Ext p g a) (hb : Ext p g b) : a = b

theorem **moves_one'** {K : Type} [Field K] [ValuativeRel K] [TopologicalSpace K] {E : Type} [CommRing E] [Algebra K E] [TopologicalSpace E] [MeasurableSpace E] {M : Loc.Model K E} {V : Type} [AddCommGroup V] [Module E V] {H : Loc.HermSp M V} {r : ℕ} {P : LocNG.FPair H r} (d : LocNG.Datum P) : LocNG.Datum.Moves 1 d d

theorem **moves_one** {p : PlF D} (d : LocNG.Datum (FP p)) : LocNG.Datum.Moves 1 d d

theorem **one_mem_stabY** (y : Dat Z FP) : (1 : GV D) ∈ stabY F F' y

theorem **inGen_self** (y : Dat Z FP) : InGen y y

theorem **weight_le_one** (y : Dat Z FP) : (((stabY F F' y).encard : ℝ≥0∞))⁻¹ ≤ 1

theorem **grel_symm** {y y' : Dat Z FP} (h : GRel F F' y y') : GRel F F' y' y

theorem **perp_of_fix** {p : PlF D} (f : Fin (D.R.rS + 1) → Vp D p) (g : Vp D p ≃ₗ[(D.s.atP p).Ep] Vp D p) (hu : LocNG.IsUnitary (locH Z p) g) (hf : ∀ a, g (f a) = f a) (x : Vp D p) (hx : x ∈ (locH Z p).perp ((locH Z p).X f)) : g x ∈ (locH Z p).perp ((locH Z p).X f)

theorem **h_add_right** (p : PlF D) (x y y' : Vp D p) : (locH Z p).h x (y + y') = (locH Z p).h x y + (locH Z p).h x y'

theorem **h_smul_right** (p : PlF D) (c) (x y : Vp D p) : (locH Z p).h x (c • y) = (Z.M p).L.cj c * (locH Z p).h x y

theorem **ext_unitary** {p : PlF D} {h : GV D} (hh : h ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) {g} (hg : Ext p h g) : LocNG.IsUnitary (locH Z p) g

theorem **grel_equivalence** (hB : Based Z FP Tn T'n F F') : Equivalence (GRel (Z

theorem **grel_out** (hB : Based Z FP Tn T'n F F') (y : Dat Z FP) : GRel F F' (Quot.mk (GRel (Z

theorem **tg_self** (t : TauC Y u u') : TG (lf D (X.Fu u)) t.frame t t.lam

theorem **clsRow_row** (TS : Set (TauC Y u u')) : ClsRow TS (Y.tO u u').rowOf

theorem **clsRow_key** (TS : Set (TauC Y u u')) : ClsRow TS key

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.AwayLiftCompFromZeroRingHom

theorem **mk_mem_quotGrading** (I : Ideal A) {d : ℕ} {a : A} (ha : a ∈ 𝒜 d) : Ideal.Quotient.mk I a ∈ quotGrading 𝒜 I d

theorem **isInternal_quotGrading** {I : Ideal A} (hI : I.IsHomogeneous 𝒜) : DirectSum.IsInternal (quotGrading 𝒜 I)

theorem **awayLift_mk_mul** (φ : A →+* B) {f : A} (hf : IsUnit (φ f)) {d : ℕ} (hd : f ∈ 𝒜 d) (n : ℕ) (a : A) (ha : a ∈ 𝒜 (n • d)) : awayLift 𝒜 φ f hf (Away.mk 𝒜 hd n a ha) * φ f ^ n = φ a

theorem **eq_awayLift** (φ : A →+* B) {f : A} (hf : IsUnit (φ f)) {d : ℕ} (hd : f ∈ 𝒜 d) (g : Away 𝒜 f →+* B) (hg : ∀ n a ha, g (Away.mk 𝒜 hd n a ha) * φ f ^ n = φ a) : g = awayLift 𝒜 φ f hf

theorem **awayLift_comp_fromZeroRingHom** (φ : A →+* B) (f : A) (hf : IsUnit (φ f)) : (awayLift 𝒜 φ f hf).comp (fromZeroRingHom 𝒜 _) = φ.comp (algebraMap (𝒜 0) A)

theorem **map_irrelevant_comp** {φ : A →+* B} (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) (ψ : B →+* C) : (irrelevant 𝒜).toIdeal.map (ψ.comp φ) = ⊤

theorem **ofHom_preimage_basicOpen** (φ : A →+* B) (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) {r : A} {n : ℕ} (hn : 0 < n) (hr : r ∈ 𝒜 n) : ofHom 𝒜 φ hφ ⁻¹ᵁ Proj.basicOpen 𝒜 r = PrimeSpectrum.basicOpen (φ r)

theorem **ofHom_toSpecZero** (φ : A →+* B) (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) : ofHom 𝒜 φ hφ ≫ Proj.toSpecZero 𝒜 = Spec.map (CommRingCat.ofHom (φ.comp (algebraMap (𝒜 0) A)))

theorem **SpecMap_ofHom_eq_awayι** (φ : A →+* B) (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) (ψ : B →+* C) {f : A} {n : ℕ} (hn : 0 < n) (hf : f ∈ 𝒜 n) (hu : IsUnit (ψ (φ f))) : Spec.map (CommRingCat.ofHom ψ) ≫ ofHom 𝒜 φ hφ = Spec.map (CommRingCat.ofHom (awayLift 𝒜 (ψ.comp φ) f hu)) ≫ Proj.awayι 𝒜 f hf hn

theorem **ofHom_eq_awayι** (φ : A →+* B) (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) {f : A} {n : ℕ} (hn : 0 < n) (hf : f ∈ 𝒜 n) (hu : IsUnit (φ f)) : ofHom 𝒜 φ hφ = Spec.map (CommRingCat.ofHom (awayLift 𝒜 φ f hu)) ≫ Proj.awayι 𝒜 f hf hn

theorem **hom_ext** (φ : A →+* B) (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) {Y : Scheme.{u}} {g₁ g₂ : Spec (.of B) ⟶ Y} (h : ∀ (C : Type u) [CommRing C] (ψ : B →+* C) (f : A) (n : ℕ), 0 < n → f ∈ 𝒜 n → IsUnit (ψ (φ f)) → Spec.map (CommRingCat.ofHom ψ) ≫ g₁ = Spec.map (CommRingCat.ofHom ψ) ≫ g₂) : g₁ = g₂

theorem **SpecMap_ofHom** (φ : A →+* B) (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) (ψ : B →+* C) : Spec.map (CommRingCat.ofHom ψ) ≫ ofHom 𝒜 φ hφ = ofHom 𝒜 (ψ.comp φ) (map_irrelevant_comp 𝒜 hφ ψ)

theorem **ofHom_eq_of_scale** {φ φ' : A →+* B} (hφ : (irrelevant 𝒜).toIdeal.map φ = ⊤) (hφ' : (irrelevant 𝒜).toIdeal.map φ' = ⊤) (c : B) (hc : ∀ n, ∀ a ∈ 𝒜 n, φ' a = c ^ n * φ a) : ofHom 𝒜 φ' hφ' = ofHom 𝒜 φ hφ

theorem **map_irrelevant_comp_graded** (g : 𝒜₀ →+*ᵍ ℬ) (hg : irrelevant ℬ ≤ (irrelevant 𝒜₀).map g) {φ : A' →+* B} (hφ : (irrelevant 𝒷).toIdeal.map φ = ⊤) : (irrelevant 𝒜₀).toIdeal.map (φ.comp g.toRingHom) = ⊤

theorem **ofHom_map** (g : 𝒜₀ →+*ᵍ ℬ) (hg : irrelevant ℬ ≤ (irrelevant 𝒜₀).map g) (φ : A' →+* B) (hφ : (irrelevant 𝒷).toIdeal.map φ = ⊤) : ofHom 𝒷 φ hφ ≫ Proj.map g hg = ofHom 𝒜₀ (φ.comp g.toRingHom) (map_irrelevant_comp_graded g hg hφ)

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.ExistsIsAdaptedOfIsLocalRing

@[simp] theorem **basisOfDual_apply** (v : ι → M) (c : ι → Dual R M) (h₁ h₂) (j : ι) : basisOfDual v c h₁ h₂ j = v j

@[simp] theorem **basisOfDual_repr** (v : ι → M) (c : ι → Dual R M) (h₁ h₂) (x : M) (i : ι) : (basisOfDual v c h₁ h₂).repr x i = c i x

theorem **basisMulGL_apply** (b : Basis ι R M) (g : GL ι R) (j : ι) : basisMulGL b g j = ∑ i, (g : Matrix ι ι R) i j • b i

@[simp] theorem **toMatrix_basisMulGL** (b : Basis ι R M) (g : GL ι R) : b.toMatrix (basisMulGL b g) = g

@[simp] theorem **basisMulGL_one** (b : Basis ι R M) : basisMulGL b 1 = b

theorem **basisMulGL_mul** (b : Basis ι R M) (g h : GL ι R) : basisMulGL b (g * h) = basisMulGL (basisMulGL b g) h

@[simp] theorem **coe_basisChange** (b b' : Basis ι R M) : (basisChange b b' : Matrix ι ι R) = b.toMatrix b'

@[simp] theorem **basisMulGL_basisChange** (b b' : Basis ι R M) : basisMulGL b (basisChange b b') = b'

@[simp] theorem **basisChange_basisMulGL** (b : Basis ι R M) (g : GL ι R) : basisChange b (basisMulGL b g) = g

theorem **basisMulGL_repr** (b : Basis ι R M) (g : GL ι R) (x : M) (i : ι) : (basisMulGL b g).repr x i = ∑ j, ((g⁻¹ : GL ι R) : Matrix ι ι R) i j * b.repr x j

theorem **mem_upperTriangularGL** {g : GL ι R} : g ∈ upperTriangularGL ι R ↔ ∀ i j, j < i → (g : Matrix ι ι R) i j = 0

theorem **upperTriangular_mul_diag** {a b : Matrix ι ι R} (ha : a.IsUpperTriangular) (hb : b.IsUpperTriangular) (i : ι) : (a * b) i i = a i i * b i i

@[simp] theorem **upperTriangularGL.coe_diagChar** (i : ι) (g : upperTriangularGL ι R) : (upperTriangularGL.diagChar i g : R) = (g.1 : Matrix ι ι R) i i

theorem **map_mem_upperTriangularGL** {S : Type*} [CommRing S] (f : R →+* S) {g : GL ι R} (hg : g ∈ upperTriangularGL ι R) : Matrix.GeneralLinearGroup.map f g ∈ upperTriangularGL ι S

theorem **mem_basis_flag_iff** (b : Basis (Fin n) R M) (k : Fin (n + 1)) (x : M) : x ∈ b.flag k ↔ ∀ i : Fin n, k ≤ i.castSucc → b.repr x i = 0

theorem **flag_basisMulGL_le** (b : Basis (Fin n) R M) {g : GL (Fin n) R} (hg : g ∈ upperTriangularGL (Fin n) R) (k : Fin (n + 1)) : (basisMulGL b g).flag k ≤ b.flag k

theorem **flag_basisMulGL_eq_iff** (b : Basis (Fin n) R M) (g : GL (Fin n) R) : (basisMulGL b g).flag = b.flag ↔ g ∈ upperTriangularGL (Fin n) R

theorem **flag_eq_flag_iff** (b b' : Basis (Fin n) R M) : b'.flag = b.flag ↔ basisChange b b' ∈ upperTriangularGL (Fin n) R

theorem **monotone** (F : FullFlag R M n) : Monotone F.sub

@[simp] theorem **ofBasis_sub** (b : Basis (Fin n) R M) : (ofBasis b).sub = b.flag

theorem **isAdapted_iff** {F : FullFlag R M n} {b : Basis (Fin n) R M} : F.IsAdapted b ↔ ofBasis b = F

theorem **isAdapted_ofBasis** (b : Basis (Fin n) R M) : (ofBasis b).IsAdapted b

theorem **ofBasis_eq_ofBasis_iff** (b b' : Basis (Fin n) R M) : ofBasis b' = ofBasis b ↔ ∃ g ∈ upperTriangularGL (Fin n) R, basisMulGL b g = b'

theorem **IsAdapted.basisMulGL_iff** {F : FullFlag R M n} {b : Basis (Fin n) R M} (hb : F.IsAdapted b) (g : GL (Fin n) R) : F.IsAdapted (basisMulGL b g) ↔ g ∈ upperTriangularGL (Fin n) R

theorem **exists_retraction** (F : FullFlag R M n) (k : Fin (n + 1)) : ∃ r : M →ₗ[R] M, (∀ x, r x ∈ F.sub k) ∧ ∀ x ∈ F.sub k, r x = x

theorem **exists_isCompl** (F : FullFlag R M n) (k : Fin (n + 1)) : ∃ Q : Submodule R M, IsCompl (F.sub k) Q

theorem **exists_splitting** (F : FullFlag R M n) (i : Fin n) : ∃ (π : M →ₗ[R] F.Step i) (σ : F.Step i →ₗ[R] M), (∀ l, π (σ l) = l) ∧ (∀ x ∈ F.sub i.castSucc, π x = 0) ∧ (∀ l, σ l ∈ F.sub i.succ) ∧ (∀ x ∈ F.sub i.succ, x - σ (π x) ∈ F.sub i.castSucc)

theorem **nonempty_baseChange_step_equiv** (F : FullFlag R M n) (S : Type*) [CommRing S] [Algebra R S] (i : Fin n) : Nonempty (SubQuot ((F.sub i.castSucc).baseChange S) ((F.sub i.succ).baseChange S) ≃ₗ[S] S ⊗[R] F.Step i)

@[simp] theorem **baseChange_sub** (F : FullFlag R M n) (S : Type*) [CommRing S] [Algebra R S] (k : Fin (n + 1)) : (F.baseChange S).sub k = (F.sub k).baseChange S

theorem **ofBasis_baseChange** (b : Basis (Fin n) R M) (S : Type*) [CommRing S] [Algebra R S] : ofBasis (b.baseChange S) = (ofBasis b).baseChange S

@[simp] theorem **map_sub** {M' : Type*} [AddCommGroup M'] [Module R M'] (F : FullFlag R M n) (e : M ≃ₗ[R] M') (k : Fin (n + 1)) : (F.map e).sub k = (F.sub k).map e.toLinearMap

theorem **ofBasis_map** {M' : Type*} [AddCommGroup M'] [Module R M'] (b : Basis (Fin n) R M) (e : M ≃ₗ[R] M') : ofBasis (b.map e) = (ofBasis b).map e

theorem **exists_isAdapted_of_free** (F : FullFlag R M n) (h : ∀ i, Module.Free R (F.Step i)) : ∃ b : Basis (Fin n) R M, F.IsAdapted b

theorem **exists_isAdapted_of_isLocalRing** [IsLocalRing R] (F : FullFlag R M n) : ∃ b : Basis (Fin n) R M, F.IsAdapted b

theorem **free_baseChange_of_algHom** {L S T : Type*} [AddCommGroup L] [Module R L] [CommRing S] [CommRing T] [Algebra R S] [Algebra R T] (φ : S →ₐ[R] T) [Module.Free S (S ⊗[R] L)] : Module.Free T (T ⊗[R] L)

theorem **exists_isAdapted_localization** (F : FullFlag R M n) (p : Ideal R) [hp : p.IsPrime] : ∃ f ∉ p, ∀ (S : Type*) [CommRing S] [Algebra R S], IsUnit (algebraMap R S f) → ∃ b : Basis (Fin n) S (S ⊗[R] M), (F.baseChange S).IsAdapted b

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.ExistsIsAffineOpenOfFiniteOpensProj

theorem **isLocalization_away_of_isInvariant** (G : Type*) [Group G] [Finite G] {B B' : Type*} (A A' : Type*) [CommRing B] [CommRing B'] [CommRing A] [CommRing A'] [Algebra B A] [Algebra B' A'] [Algebra B B'] [Algebra A A'] [MulSemiringAction G A] [MulSemiringAction G A'] [SMulCommClass G B A] [SMulCommClass G B' A'] [FaithfulSMul B A] [FaithfulSMul B' A'] [Algebra.IsInvariant B A G] [Algebra.IsInvariant B' A' G] (hc : ∀ b, algebraMap B' A' (algebraMap B B' b) = algebraMap A A' (algebraMap B A b)) (he : ∀ (g : G) (a : A), algebraMap A A' (g • a) = g • algebraMap A A' a) (r : B) [IsLocalization.Away (algebraMap B A r) A'] : IsLocalization.Away r B'

theorem **IsAffineQuotient.of_forall_exists_isOpenImmersion** [Finite G] {ρ : G →* Aut X} {π : X ⟶ Y} (hπ : ∀ g, (ρ g).hom ≫ π = π) (h : ∀ y : Y, ∃ (X' Y' : Scheme.{u}) (ρ' : G →* Aut X') (π' : X' ⟶ Y') (p : X' ⟶ X) (j : Y' ⟶ Y) (_ : IsOpenImmersion p) (_ : IsOpenImmersion j), y ∈ j.opensRange ∧ p ≫ π = π' ≫ j ∧ π ⁻¹ᵁ j.opensRange = p.opensRange ∧ (∀ g, (ρ' g).hom ≫ p = p ≫ (ρ g).hom) ∧ IsAffineQuotient ρ' π') : IsAffineQuotient ρ π

theorem **mem_invariantSections** {U : X.Opens} {hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U} {s : Γ(X, U)} : s ∈ invariantSections ρ U hU ↔ IsInvariantSection ρ U s

theorem **coe_invariantSectionsRes** {U V : X.Opens} (hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) (hV : ∀ g, V ≤ (ρ g).hom ⁻¹ᵁ V) (e : U ≤ V) (s : invariantSections ρ V hV) : (invariantSectionsRes ρ hU hV e s : Γ(X, U)) = X.presheaf.map (homOfLE e).op s

private theorem **invariantSections_of_stable_open** (U : X.Opens) (hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) : letI

theorem **isLocalization_invariantSections** [Finite G] {U V : X.Opens} (hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) (hV : ∀ g, V ≤ (ρ g).hom ⁻¹ᵁ V) (e : U ≤ V) (hV' : IsAffineOpen V) (r : invariantSections ρ V hV) (hr : X.basicOpen (r : Γ(X, V)) = U) : letI

theorem **StableAffineOpens.toOpens_le** {U V : StableAffineOpens ρ} (f : U ⟶ V) : U.toOpens ≤ V.toOpens

theorem **quotientDiagram_obj** (U : StableAffineOpens ρ) : (quotientDiagram ρ).obj U = Spec (.of (invariantSections ρ U.toOpens U.stable))

theorem **toChart_map** {U V : StableAffineOpens ρ} (f : U ⟶ V) : U.toChart ≫ (quotientDiagram ρ).map f = Spec.map (X.presheaf.map (homOfLE (StableAffineOpens.toOpens_le f)).op) ≫ V.toChart

theorem **isAffineQuotient_toChart** (U : StableAffineOpens ρ) : letI

theorem **exists_basicOpen_eq** {U V : StableAffineOpens ρ} (h : U ≤ V) : ∃ r : invariantSections ρ V.toOpens V.stable, X.basicOpen (r : Γ(X, V.toOpens)) = U.toOpens

theorem **range_map** {U V : StableAffineOpens ρ} (f : U ⟶ V) (r : invariantSections ρ V.toOpens V.stable) (hr : X.basicOpen (r : Γ(X, V.toOpens)) = U.toOpens) : Set.range ((quotientDiagram ρ).map f) = (PrimeSpectrum.basicOpen r : Set (PrimeSpectrum (invariantSections ρ V.toOpens V.stable)))

theorem **exists_le_le** (U V : StableAffineOpens ρ) {x : X} (hU : x ∈ U.toOpens) (hV : x ∈ V.toOpens) : ∃ W : StableAffineOpens ρ, W ≤ U ∧ W ≤ V ∧ x ∈ W.toOpens

theorem **exists_le_le_pullback** (U V : StableAffineOpens ρ) (x : ↑(pullback U.isAffineOpen.fromSpec V.isAffineOpen.fromSpec)) : ∃ (W : StableAffineOpens ρ) (hU : W ≤ U) (_ : W ≤ V) (y : Spec Γ(X, W.toOpens)), Spec.map (X.presheaf.map (homOfLE (toOpens_le (homOfLE hU))).op) y = pullback.fst U.isAffineOpen.fromSpec V.isAffineOpen.fromSpec x

theorem **exists_mem_opensRange_chart** (y : finiteQuotient ρ) : ∃ U, y ∈ (chart ρ U).opensRange

theorem **map_chart** {U V : StableAffineOpens ρ} (f : U ⟶ V) : (quotientDiagram ρ).map f ≫ chart ρ V = chart ρ U

theorem **trans_toChart_chart** {U V : StableAffineOpens ρ} (f : U ⟶ V) : (stableAffineCover ρ).trans f ≫ V.toChart ≫ chart ρ V = U.toChart ≫ chart ρ U

theorem **fromSpec_π** (U : StableAffineOpens ρ) : U.isAffineOpen.fromSpec ≫ π ρ = U.toChart ≫ chart ρ U

theorem **π_preimage_opensRange_chart** (U : StableAffineOpens ρ) : π ρ ⁻¹ᵁ (chart ρ U).opensRange = U.toOpens

theorem **comp_π** (g : G) : (ρ g).hom ≫ π ρ = π ρ

theorem **isAffineQuotient** : IsAffineQuotient ρ (π ρ)

theorem **π_apply_eq_iff** (x y : X) : π ρ x = π ρ y ↔ ∃ g, (ρ g).hom x = y

theorem **isOpenQuotientMap_π** : IsOpenQuotientMap (π ρ)

theorem **app_injective_and_range** (W : (finiteQuotient ρ).Opens) : Function.Injective ((π ρ).app W) ∧ Set.range ((π ρ).app W) = {s | IsInvariantSection ρ (π ρ ⁻¹ᵁ W) s}

theorem **π_desc** (f : X ⟶ Z) (hf : ∀ g, (ρ g).hom ≫ f = f) : π ρ ≫ desc ρ f hf = f

theorem **hom_ext** {a b : finiteQuotient ρ ⟶ Z} (e : π ρ ≫ a = π ρ ≫ b) : a = b

theorem **existsUnique_π_comp_eq** (f : X ⟶ Z) (hf : ∀ g, (ρ g).hom ≫ f = f) : ∃! f' : finiteQuotient ρ ⟶ Z, π ρ ≫ f' = f

theorem **comp_π_eq_iff** {k : Type u} [Field k] (x y : Spec (.of k) ⟶ X) : x ≫ π ρ = y ≫ π ρ ↔ ∃ g, x ≫ (ρ g).hom = y

theorem **exists_comp_π_eq** {k : Type u} [Field k] [IsAlgClosed k] (y : Spec (.of k) ⟶ finiteQuotient ρ) : ∃ x : Spec (.of k) ⟶ X, x ≫ π ρ = y

theorem **isFinite_π_of_isFreeAction** (hρ : IsFreeAction ρ) : IsFinite (π ρ)

theorem **etale_π** (hρ : IsFreeAction ρ) : Etale (π ρ)

theorem **finrank_π** (hρ : IsFreeAction ρ) (y : finiteQuotient ρ) : (π ρ).finrank y = Nat.card G

theorem **isFinite_π** [LocallyOfFiniteType f] : IsFinite (π ρ)

theorem **π_pullbackIso_hom** [Flat t] : π (pullbackAction ρ f hf t) ≫ (pullbackIso ρ f hf t).hom = pullbackπ ρ f hf t

theorem **π_pullbackIsoOfIsFreeAction_hom** (hρ : IsFreeAction ρ) : π (pullbackAction ρ f hf t) ≫ (pullbackIsoOfIsFreeAction ρ f hf t hρ).hom = pullbackπ ρ f hf t

theorem **π_map** (ρ : X ⟶ Y) (hp : ∀ g, ∃ k, (ρ g).hom ≫ ρ = ρ ≫ (τ k).hom) : π ρ ≫ map ρ τ ρ hp = ρ ≫ π τ

theorem **map_id** (hp) : map ρ ρ (𝟙 X) hp = 𝟙 _

theorem **map_comp** {K : Type*} [Group K] [Finite K] (σ : K →* Aut Z) [OrbitsInAffineOpens σ] (ρ : X ⟶ Y) (q : Y ⟶ Z) (hp hq hpq) : map ρ σ (ρ ≫ q) hpq = map ρ τ ρ hp ≫ map τ σ q hq

theorem **π_specIso_hom** (B A : Type u) [CommRing B] [CommRing A] [MulSemiringAction G A] [Algebra B A] [SMulCommClass G B A] [FaithfulSMul B A] [Algebra.IsInvariant B A G] : π (specAction G A) ≫ (specIso B A).hom = Spec.map (CommRingCat.ofHom (algebraMap B A))

theorem **exists_isAffineQuotient_iff** [Finite G] (ρ : G →* Aut X) : (∃ (Y : Scheme.{u}) (π : X ⟶ Y), IsAffineQuotient ρ π) ↔ OrbitsInAffineOpens ρ

theorem **IsAffineQuotient.isAffineOpen_of_preimage** [Finite G] {ρ : G →* Aut X} {π : X ⟶ Y} (h : IsAffineQuotient ρ π) {W : Y.Opens} (hW : IsAffineOpen (π ⁻¹ᵁ W)) : IsAffineOpen W

theorem **exists_isAffineOpen_of_finite_opens_proj** {σ : Type*} {A : Type u} [CommRing A] [SetLike σ A] [AddSubgroupClass σ A] (𝒜 : ℕ → σ) [GradedRing 𝒜] (O : (Proj 𝒜).Opens) (s : Set O.toScheme) (hs : s.Finite) : ∃ U : O.toScheme.Opens, IsAffineOpen U ∧ s ⊆ U

theorem **comp_hom_comp_eq_of_normal** [hφ : φ.range.Normal] (hπ : ∀ k, (ρ.comp φ k).hom ≫ π = π) (g : G) (k : H) : (ρ.comp φ k).hom ≫ (ρ g).hom ≫ π = (ρ g).hom ≫ π

theorem **IsAffineQuotient.comp_residualAction** [Finite H] [φ.range.Normal] (h : IsAffineQuotient (ρ.comp φ) π) (g : G) : π ≫ (h.residualAction g).hom = (ρ g).hom ≫ π

theorem **IsAffineQuotient.comp** (h : IsAffineQuotient (ρ.comp φ) π) {τ : G →* Aut Y} (hτ : ∀ g, π ≫ (τ g).hom = (ρ g).hom ≫ π) {π' : Y ⟶ Z} (h' : IsAffineQuotient τ π') : IsAffineQuotient ρ (π ≫ π') where

theorem **overAction_hom** (ρ : G →* Aut X) (g : G) : (overAction ρ g).hom = (ρ g).hom.left

theorem **overAction_comp** (ρ : G →* Aut X) (g : G) : (overAction ρ g).hom ≫ X.hom = X.hom

theorem **finiteQuotientOver.π_left** : (finiteQuotientOver.π ρ).left = finiteQuotient.π (overAction ρ)

theorem **finiteQuotientOver.isAffineQuotient** : IsAffineQuotient (overAction ρ) (finiteQuotientOver.π ρ).left

theorem **finiteQuotientOver.comp_π** (g : G) : (ρ g).hom ≫ finiteQuotientOver.π ρ = finiteQuotientOver.π ρ

theorem **finiteQuotientOver.existsUnique_π_comp_eq** {T : Over S} (f : X ⟶ T) (hf : ∀ g, (ρ g).hom ≫ f = f) : ∃! f' : finiteQuotientOver ρ ⟶ T, finiteQuotientOver.π ρ ≫ f' = f

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.ExistsIsInvariantSectionMemBasicOpenIff

private theorem **mem_basicOpen_iff_not_mem_primeIdealOf** {U : X.Opens} (hU : IsAffineOpen U) {x : X} (hx : x ∈ U) (f : Γ(X, U)) : x ∈ X.basicOpen f ↔ f ∉ (hU.primeIdealOf ⟨x, hx⟩).asIdeal

theorem **exists_basicOpen_le_of_finite** {U : X.Opens} (hU : IsAffineOpen U) (O : X.Opens) {s : Set X} (hs : s.Finite) (hsU : s ⊆ U) (hsO : s ⊆ O) : ∃ f : Γ(X, U), s ⊆ X.basicOpen f ∧ X.basicOpen f ≤ O

theorem **exists_isAffineOpen_le** [FiniteSubsetsInAffineOpens X] {s : Set X} (hs : s.Finite) (O : X.Opens) (hsO : s ⊆ O) : ∃ U : X.Opens, IsAffineOpen U ∧ s ⊆ U ∧ U ≤ O

theorem **of_isAffineHom** (f : X ⟶ Y) [IsAffineHom f] [FiniteSubsetsInAffineOpens Y] : FiniteSubsetsInAffineOpens X

theorem **of_isOpenImmersion** (f : X ⟶ Y) [IsOpenImmersion f] [FiniteSubsetsInAffineOpens Y] : FiniteSubsetsInAffineOpens X

theorem **of_isImmersion** (f : X ⟶ Y) [IsImmersion f] [FiniteSubsetsInAffineOpens Y] : FiniteSubsetsInAffineOpens X

theorem **exists_proj_basicOpen** {σ : Type*} {A : Type u} [CommRing A] [SetLike σ A] [AddSubgroupClass σ A] (𝒜 : ℕ → σ) [GradedRing 𝒜] {s : Set (Proj 𝒜)} (hs : s.Finite) : ∃ n, 0 < n ∧ ∃ f ∈ 𝒜 n, s ⊆ Proj.basicOpen 𝒜 f

theorem **of_isQProjOver** {R : Type} [CommRing R] {X : Scheme.{0}} {f : X ⟶ ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp R} (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsQProjOver f) : FiniteSubsetsInAffineOpens X

theorem **of_isProjOver** {R : Type} [CommRing R] {X : Scheme.{0}} {f : X ⟶ ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp R} (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsProjOver f) : FiniteSubsetsInAffineOpens X

theorem **of_isQProjOver** [Finite G] {R : Type} [CommRing R] {X : Scheme.{0}} {f : X ⟶ ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp R} (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsQProjOver f) (ρ : G →* Aut X) : OrbitsInAffineOpens ρ

theorem **of_locally_on_base** [Finite G] (ρ : G →* Aut X) (f : X ⟶ Y) (hf : ∀ g, (ρ g).hom ≫ f = f) (h : ∀ y : Y, ∃ W : Y.Opens, y ∈ W ∧ FiniteSubsetsInAffineOpens (f ⁻¹ᵁ W)) : OrbitsInAffineOpens ρ

theorem **of_equivariant** (ρ : G →* Aut X) (τ : G →* Aut Y) [OrbitsInAffineOpens τ] (p : X ⟶ Y) [IsAffineHom p] (hp : ∀ g, (ρ g).hom ≫ p = p ≫ (τ g).hom) : OrbitsInAffineOpens ρ

theorem **of_invariant** (ρ : G →* Aut X) (p : X ⟶ Y) [IsAffineHom p] (hp : ∀ g, (ρ g).hom ≫ p = p) : OrbitsInAffineOpens ρ

theorem **IsAffineQuotient.orbitsInAffineOpens** {ρ : G →* Aut X} {π : X ⟶ Y} (h : IsAffineQuotient ρ π) : OrbitsInAffineOpens ρ

theorem **aut_map_mul_apply** (ρ : G →* Aut X) (g k : G) (x : X) : (ρ (g * k)).hom x = (ρ g).hom ((ρ k).hom x)

theorem **aut_map_one_apply** (ρ : G →* Aut X) (x : X) : (ρ 1).hom x = x

theorem **IsInvariantSection.basicOpen_le** {U : X.Opens} {s : Γ(X, U)} (hs : IsInvariantSection ρ U s) (g : G) : X.basicOpen s ≤ (ρ g).hom ⁻¹ᵁ X.basicOpen s

private theorem **mem_basicOpen_prod_iff** {U : X.Opens} (a : G → Γ(X, U)) (T : Finset G) (x : X) : x ∈ X.basicOpen (∏ g ∈ T, a g) ↔ x ∈ U ∧ ∀ g ∈ T, x ∈ X.basicOpen (a g)

private theorem **appLE_apply_congr'** {f f' : X ⟶ X} (hff : f = f') (U : X.Opens) (e : U ≤ f ⁻¹ᵁ U) (e' : U ≤ f' ⁻¹ᵁ U) (s : Γ(X, U)) : f.appLE U U e s = f'.appLE U U e' s

theorem **exists_isInvariantSection_mem_basicOpen_iff** [Finite G] {U : X.Opens} (hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) (s : Γ(X, U)) : ∃ t : Γ(X, U), IsInvariantSection ρ U t ∧ ∀ x, x ∈ X.basicOpen t ↔ ∀ g, (ρ g).hom x ∈ X.basicOpen s

theorem **exists_isInvariantSection_basicOpen_eq** {U : X.Opens} (hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) (s : Γ(X, U)) (hs : ∀ g, X.basicOpen s ≤ (ρ g).hom ⁻¹ᵁ X.basicOpen s) : ∃ t : Γ(X, U), IsInvariantSection ρ U t ∧ X.basicOpen t = X.basicOpen s

theorem **exists_isInvariantSection_basicOpen_le** {U : X.Opens} (hU : IsAffineOpen U) (hst : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) {x : X} (hx : x ∈ U) (O : X.Opens) (hO : ∀ g, (ρ g).hom x ∈ O) : ∃ t : Γ(X, U), IsInvariantSection ρ U t ∧ x ∈ X.basicOpen t ∧ X.basicOpen t ≤ O

theorem **exists_stable_isAffineOpen** [OrbitsInAffineOpens ρ] (x : X) : ∃ V : X.Opens, IsAffineOpen V ∧ (∀ g, V ≤ (ρ g).hom ⁻¹ᵁ V) ∧ x ∈ V

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.ExistsLiftAddOfIsReducedEq

theorem **isReduced_pullback_toSpec** {X : Scheme.{u}} [IsReduced X] (f : X ⟶ Spec (.of R)) : IsReduced (pullback f (toSpec W))

theorem **lift_addOfIsReduced** (k : Type u) [Field k] [Algebra R k] (a b : specOver R k ⟶ schemeOver W) : lift a b ≫ addOfIsReduced W = fieldAdd W k a b

theorem **isCommMonObj_of_isReduced** : letI

theorem **exists_lift_addOfIsReduced_eq** {A : Type u} [CommRing A] [IsLocalRing A] [Algebra R A] {P Q : Fin 3 → A} (hP : IsUnimodularSol (W.baseChange A) P) (hQ : IsUnimodularSol (W.baseChange A) Q) : ∃ (S : Fin 3 → A) (hS : IsUnimodularSol (W.baseChange A) S), (S = (W.baseChange A).toProjective.addXYZ P Q ∨ S = diagXYZ (W.baseChange A).toProjective P Q) ∧ lift (ptOver W P hP) (ptOver W Q hQ) ≫ addOfIsReduced W = ptOver W S hS

theorem **map_coeffHom** : genericCurve.map (coeffHom W) = W

theorem **map_classifyingHom** [W.IsElliptic] : universalCurve.map (classifyingHom W) = W

theorem **eq_classifyingHom** [W.IsElliptic] {φ : UniversalRing.{u} →+* R} (h : universalCurve.map φ = W) : φ = classifyingHom W

theorem **isPullback_universalCurve** [W.IsElliptic] : IsPullback (mapHom universalCurve (classifyingHom W) (map_classifyingHom W)) (schemeOver W).hom (schemeOver universalCurve).hom (Spec.map (CommRingCat.ofHom (classifyingHom W)))

theorem **one_eq_zeroUnit** : η[schemeOver W] = zeroUnit W

theorem **inv_eq_negOver** : ι[schemeOver W] = negOver W

@[simp] theorem **one_left** : η[schemeOver W].left = zero W

@[simp] theorem **inv_left** : ι[schemeOver W].left = negHom W

theorem **hom_one_left** (T : Over (Spec (.of R))) : (1 : T ⟶ schemeOver W).left = T.hom ≫ zero W

theorem **mul_left_mapHom_universal** {T' : Over (Spec (.of R))} {T : Over (Spec (.of UniversalRing.{u}))} (t : T'.left ⟶ T.left) {x' y' : T' ⟶ schemeOver W} {x y : T ⟶ schemeOver universalCurve} (hx : x'.left ≫ mapHom universalCurve (classifyingHom W) (map_classifyingHom W) = t ≫ x.left) (hy : y'.left ≫ mapHom universalCurve (classifyingHom W) (map_classifyingHom W) = t ≫ y.left) : (x' * y').left ≫ mapHom universalCurve (classifyingHom W) (map_classifyingHom W) = t ≫ (lift x y ≫ addOfIsReduced universalCurve).left

theorem **mul_left_mapHom** {T' : Over (Spec (.of R'))} {T : Over (Spec (.of R))} (t : T'.left ⟶ T.left) {x' y' : T' ⟶ schemeOver W'} {x y : T ⟶ schemeOver W} (hx : x'.left ≫ mapHom W f h = t ≫ x.left) (hy : y'.left ≫ mapHom W f h = t ≫ y.left) : (x' * y').left ≫ mapHom W f h = t ≫ (x * y).left

theorem **isGrpPullback_mapHom** {R R' : Type} [CommRing R] [CommRing R'] (W : WeierstrassCurve R) [W.IsElliptic] {W' : WeierstrassCurve R'} [W'.IsElliptic] (f : R →+* R') (h : W.map f = W') : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsGrpPullback (schemeOver W') (schemeOver W) (Spec.map (CommRingCat.ofHom f)) (mapHom W f h)

theorem **one_eq_zeroOver** : (1 : specOver R A ⟶ schemeOver W) = zeroOver W A

theorem **one_eq_ptOver** : (1 : specOver R A ⟶ schemeOver W) = ptOver W ![0, 1, 0] (isUnimodularSol_zero _)

theorem **ptOver_inv** (hP : IsUnimodularSol (W.baseChange A) P) : (ptOver W P hP)⁻¹ = ptOver W ((W.baseChange A).toProjective.neg P) hP.neg

theorem **exists_ptOver_mul_ptOver_eq** [IsLocalRing A] (hP : IsUnimodularSol (W.baseChange A) P) (hQ : IsUnimodularSol (W.baseChange A) Q) : ∃ (S : Fin 3 → A) (hS : IsUnimodularSol (W.baseChange A) S), (S = (W.baseChange A).toProjective.addXYZ P Q ∨ S = diagXYZ (W.baseChange A).toProjective P Q) ∧ ptOver W P hP * ptOver W Q hQ = ptOver W S hS

theorem **ptOver_mul_ptOver** (hP : IsUnimodularSol (W.baseChange A) P) (hQ : IsUnimodularSol (W.baseChange A) Q) (hT : IsUnimodularSol (W.baseChange A) T) (h : IsSumTriple (W.baseChange A) P Q T) : ptOver W P hP * ptOver W Q hQ = ptOver W T hT

theorem **IsSumTriple.equation** {V : WeierstrassCurve A} [V.IsElliptic] (h : IsSumTriple V P Q T) (hP : IsUnimodularSol V P) (hQ : IsUnimodularSol V Q) : V.toProjective.Equation T

theorem **ptOver_mul_self** (hP : IsUnimodularSol (W.baseChange A) P) (hT : IsUnimodularSol (W.baseChange A) T) (h : Proportional T ((W.baseChange A).toProjective.dblXYZ P)) : ptOver W P hP * ptOver W P hP = ptOver W T hT

theorem **ofProjectivePoint_add** (p q : (W.baseChange k).toProjective.Point) : ofProjectivePoint W k (p + q) = ofProjectivePoint W k p * ofProjectivePoint W k q

@[simp] theorem **projectivePointMulEquiv_ofAdd** (p : (W.baseChange k).toProjective.Point) : projectivePointMulEquiv W k (.ofAdd p) = ofProjectivePoint W k p

theorem **ofAffinePoint_add** (p q : (W.baseChange k).toAffine.Point) : ofAffinePoint W k (p + q) = ofAffinePoint W k p * ofAffinePoint W k q

@[simp] theorem **affinePointMulEquiv_ofAdd** (p : (W.baseChange k).toAffine.Point) : affinePointMulEquiv W k (.ofAdd p) = ofAffinePoint W k p

theorem **ofAffinePoint_zsmul** (n : ℤ) (p : (W.baseChange k).toAffine.Point) : ofAffinePoint W k (n • p) = ofAffinePoint W k p ^ n

private theorem **sample_sol** {a b : ℚ} (h : b ^ 2 + b = a ^ 3 - a) : IsUnimodularSol (sampleCurve.baseChange ℚ) ![a, b, 1]

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.FlatImageOfOverFractionFieldOfValuationRing

theorem **ker_comp_of_schemeDominant** (d : T' ⟶ T) (t : T ⟶ P) [IsSchemeTheoreticallyDominant d] : (d ≫ t).ker = t.ker

theorem **schemeDominant_of_comp** (d : T' ⟶ T) (t : T ⟶ P) [IsSchemeTheoreticallyDominant (d ≫ t)] : IsSchemeTheoreticallyDominant t

theorem **schemeDominant_specMap_iff** {R R' : CommRingCat.{u}} (φ : R ⟶ R') : IsSchemeTheoreticallyDominant (Spec.map φ) ↔ Function.Injective φ.hom

theorem **isIso_of_isClosedImmersion_of_schemeDominant** (i : T ⟶ P) [IsClosedImmersion i] [IsSchemeTheoreticallyDominant i] : IsIso i

theorem **isIntegral_of_schemeDominant** (d : T' ⟶ T) [IsSchemeTheoreticallyDominant d] [QuasiCompact d] [IsIntegral T'] : IsIntegral T

theorem **eq_zero_of_mul_eq_zero_of_flat** {S : Scheme.{u}} [IsAffine S] (s : T ⟶ S) [Flat s] (V : T.Opens) (r : Γ(S, ⊤)) (hr : r ∈ nonZeroDivisors Γ(S, ⊤)) (a : Γ(T, V)) (h : s.appLE ⊤ V le_top r * a = 0) : a = 0

theorem **flat_of_schemeDominant_of_torsionFree_flat** {O : CommRingCat.{u}} (hO : ∀ (M : Type u) [AddCommGroup M] [Module O M], Submodule.torsion O M = ⊥ → Module.Flat O M) (d : T' ⟶ T) (s : T ⟶ Spec O) [IsSchemeTheoreticallyDominant d] [QuasiCompact d] [Flat (d ≫ s)] : Flat s

theorem **flat_of_schemeDominant_of_isDedekindDomain** {O : CommRingCat.{u}} [IsDedekindDomain O] (d : T' ⟶ T) (s : T ⟶ Spec O) [IsSchemeTheoreticallyDominant d] [QuasiCompact d] [Flat (d ≫ s)] : Flat s

theorem **flat_of_schemeDominant_of_isBezout** {O : CommRingCat.{u}} [IsDomain O] [IsBezout O] (d : T' ⟶ T) (s : T ⟶ Spec O) [IsSchemeTheoreticallyDominant d] [QuasiCompact d] [Flat (d ≫ s)] : Flat s

theorem **flat_specMap_localization** {O : Type u} [CommRing O] (M : Submonoid O) (K : Type u) [CommRing K] [Algebra O K] [IsLocalization M K] : Flat (Spec.map (CommRingCat.ofHom (algebraMap O K)))

theorem **mono_specMap_localization** {O : Type u} [CommRing O] (M : Submonoid O) (K : Type u) [CommRing K] [Algebra O K] [IsLocalization M K] : Mono (Spec.map (CommRingCat.ofHom (algebraMap O K)))

theorem **flat_of_over_fractionField** {O K : Type u} [CommRing O] [Field K] [Algebra O K] [IsFractionRing O K] (s : T ⟶ Spec (.of O)) (s' : T ⟶ Spec (.of K)) (h : s' ≫ Spec.map (CommRingCat.ofHom (algebraMap O K)) = s) : Flat s

theorem **ker_le_ker_of_schemeDominant** (t : T ⟶ P) (d : T' ⟶ T) [IsSchemeTheoreticallyDominant d] (w : T' ⟶ W) (h : d ≫ t = w ≫ j) : j.ker ≤ t.ker

theorem **closureLift_imageι** (t : T ⟶ P) (d : T' ⟶ T) [IsSchemeTheoreticallyDominant d] (w : T' ⟶ W) (h : d ≫ t = w ≫ j) : closureLift j t d w h ≫ j.imageι = t

theorem **comp_closureLift** (t : T ⟶ P) (d : T' ⟶ T) [IsSchemeTheoreticallyDominant d] (w : T' ⟶ W) (h : d ≫ t = w ≫ j) : d ≫ closureLift j t d w h = w ≫ j.toImage

theorem **closure_hom_ext** {a b : T ⟶ j.image} (h : a ≫ j.imageι = b ≫ j.imageι) : a = b

theorem **existsUnique_closureLift** (t : T ⟶ P) (d : T' ⟶ T) [IsSchemeTheoreticallyDominant d] (w : T' ⟶ W) (h : d ≫ t = w ≫ j) : ∃! l : T ⟶ j.image, l ≫ j.imageι = t

theorem **existsUnique_closureLift_of_isReduced** (t : T ⟶ P) (d : T' ⟶ T) [IsReduced T] [IsDominant d] (w : T' ⟶ W) (h : d ≫ t = w ≫ j) : ∃! l : T ⟶ j.image, l ≫ j.imageι = t

theorem **existsUnique_closureLift_of_injective** {R R' : CommRingCat.{u}} (φ : R ⟶ R') (hφ : Function.Injective φ.hom) (t : Spec R ⟶ P) (w : Spec R' ⟶ W) (h : Spec.map φ ≫ t = w ≫ j) : ∃! l : Spec R ⟶ j.image, l ≫ j.imageι = t

theorem **exists_closure_fac_iff** (t : T ⟶ P) : (∃ l : T ⟶ j.image, l ≫ j.imageι = t) ↔ j.ker ≤ t.ker

theorem **closureToClosed_comp** (g : W ⟶ Z) (i : Z ⟶ P) [IsClosedImmersion i] (h : g ≫ i = j) : closureToClosed j g i h ≫ i = j.imageι

theorem **toImage_closureToClosed** (g : W ⟶ Z) (i : Z ⟶ P) [IsClosedImmersion i] (h : g ≫ i = j) : j.toImage ≫ closureToClosed j g i h = g

theorem **isIso_imageι_iff** : IsIso j.imageι ↔ IsSchemeTheoreticallyDominant j

theorem **isOpenImmersion_toImage** [IsImmersion j] [QuasiCompact j] : IsOpenImmersion j.toImage

theorem **range_imageι** [QuasiCompact j] : Set.range j.imageι = closure (Set.range j)

theorem **flat_image_of_isDedekindDomain** {O : CommRingCat.{u}} [IsDedekindDomain O] (s : P ⟶ Spec O) [QuasiCompact j] [Flat (j ≫ s)] : Flat (j.imageι ≫ s)

theorem **flat_image_of_isBezout** {O : CommRingCat.{u}} [IsDomain O] [IsBezout O] (s : P ⟶ Spec O) [QuasiCompact j] [Flat (j ≫ s)] : Flat (j.imageι ≫ s)

theorem **flat_image_of_valuationRing** {O : CommRingCat.{u}} [IsDomain O] [ValuationRing O] (s : P ⟶ Spec O) [QuasiCompact j] [Flat (j ≫ s)] : Flat (j.imageι ≫ s)

theorem **flat_image_of_over_fractionField** {O K : Type u} [CommRing O] [IsDedekindDomain O] [Field K] [Algebra O K] [IsFractionRing O K] (s : P ⟶ Spec (.of O)) (s' : W ⟶ Spec (.of K)) (h : s' ≫ Spec.map (CommRingCat.ofHom (algebraMap O K)) = j ≫ s) [QuasiCompact j] : Flat (j.imageι ≫ s)

theorem **flat_image_of_over_fractionField_of_valuationRing** {O K : Type u} [CommRing O] [IsDomain O] [ValuationRing O] [Field K] [Algebra O K] [IsFractionRing O K] (s : P ⟶ Spec (.of O)) (s' : W ⟶ Spec (.of K)) (h : s' ≫ Spec.map (CommRingCat.ofHom (algebraMap O K)) = j ≫ s) [QuasiCompact j] : Flat (j.imageι ≫ s)

theorem **closureMap_imageι** {W' P' : Scheme.{u}} (j' : W' ⟶ P') (w : W ⟶ W') (g : P ⟶ P') (h : j ≫ g = w ≫ j') : closureMap j j' w g h ≫ j'.imageι = j.imageι ≫ g

theorem **toImage_closureMap** {W' P' : Scheme.{u}} (j' : W' ⟶ P') (w : W ⟶ W') (g : P ⟶ P') (h : j ≫ g = w ≫ j') : j.toImage ≫ closureMap j j' w g h = w ≫ j'.toImage

theorem **closureMap_id** : closureMap j j (𝟙 W) (𝟙 P) (by simp) = 𝟙 _

theorem **closureMap_comp** {W' P' W'' P'' : Scheme.{u}} (j' : W' ⟶ P') (j'' : W'' ⟶ P'') (w : W ⟶ W') (g : P ⟶ P') (h : j ≫ g = w ≫ j') (w' : W' ⟶ W'') (g' : P' ⟶ P'') (h' : j' ≫ g' = w' ≫ j'') : closureMap j j' w g h ≫ closureMap j' j'' w' g' h' = closureMap j j'' (w ≫ w') (g ≫ g') (by rw [reassoc_of% h, h', Category.assoc])

theorem **ker_eq_comap_of_flat** {W' P' : Scheme.{u}} {j' : W' ⟶ P'} {w : W' ⟶ W} {g : P' ⟶ P} (H : IsPullback j' w g j) [Flat g] [QuasiCompact j] : j'.ker = j.ker.comap g

theorem **isPullback_closureMap_of_flat** {W' P' : Scheme.{u}} {j' : W' ⟶ P'} {w : W' ⟶ W} {g : P' ⟶ P} (H : IsPullback j' w g j) [Flat g] [QuasiCompact j] : IsPullback j'.imageι (closureMap j' j w g H.w) g j.imageι

theorem **isPullback_toImage_of_isClosedImmersion** {P' : Scheme.{u}} (c : W ⟶ P') (u : P' ⟶ P) [IsClosedImmersion c] [Flat u] [Mono u] (h : c ≫ u = j) [QuasiCompact j] : IsPullback c j.toImage u j.imageι

theorem **range_toImage_of_isClosedImmersion** {P' : Scheme.{u}} (c : W ⟶ P') (u : P' ⟶ P) [IsClosedImmersion c] [Flat u] [Mono u] (h : c ≫ u = j) [QuasiCompact j] : Set.range j.toImage = j.imageι ⁻¹' Set.range u

theorem **closureRestrictIso_hom_ι** (P' : P.Opens) (c : W ⟶ P') [IsClosedImmersion c] (h : c ≫ P'.ι = j) [QuasiCompact j] : (closureRestrictIso j P' c h).hom ≫ (j.imageι ⁻¹ᵁ P').ι = j.toImage

theorem **closureRestrictIso_hom_restrict** (P' : P.Opens) (c : W ⟶ P') [IsClosedImmersion c] (h : c ≫ P'.ι = j) [QuasiCompact j] : (closureRestrictIso j P' c h).hom ≫ j.imageι |_ P' = c

theorem **closureSpecIso_hom_comp** {A B : CommRingCat.{u}} (φ : A ⟶ B) : (closureSpecIso φ).hom ≫ Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (RingHom.ker φ.hom))) = (Spec.map φ).imageι

theorem **graphMap_fst** : graphMap f₁ f₂ u φ h ≫ pullback.fst f₁ f₂ = u

theorem **graphMap_snd** : graphMap f₁ f₂ u φ h ≫ pullback.snd f₁ f₂ = φ

theorem **ι_fst** : ι f₁ f₂ u φ h ≫ pullback.fst f₁ f₂ = fst f₁ f₂ u φ h

theorem **ι_snd** : ι f₁ f₂ u φ h ≫ pullback.snd f₁ f₂ = snd f₁ f₂ u φ h

theorem **incl_ι** : incl f₁ f₂ u φ h ≫ ι f₁ f₂ u φ h = graphMap f₁ f₂ u φ h

theorem **incl_fst** : incl f₁ f₂ u φ h ≫ fst f₁ f₂ u φ h = u

theorem **incl_snd** : incl f₁ f₂ u φ h ≫ snd f₁ f₂ u φ h = φ

theorem **condition** : fst f₁ f₂ u φ h ≫ f₁ = snd f₁ f₂ u φ h ≫ f₂

theorem **range_ι** [QuasiCompact u] [QuasiSeparated f₂] : Set.range (ι f₁ f₂ u φ h) = closure (Set.range (graphMap f₁ f₂ u φ h))

theorem **hom_ext** {a b : T ⟶ graphClosure f₁ f₂ u φ h} (h₁ : a ≫ fst f₁ f₂ u φ h = b ≫ fst f₁ f₂ u φ h) (h₂ : a ≫ snd f₁ f₂ u φ h = b ≫ snd f₁ f₂ u φ h) : a = b

theorem **lift_ι** : lift f₁ f₂ u φ h t₁ t₂ ht d w h₁ h₂ ≫ ι f₁ f₂ u φ h = pullback.lift t₁ t₂ ht

theorem **lift_fst** : lift f₁ f₂ u φ h t₁ t₂ ht d w h₁ h₂ ≫ fst f₁ f₂ u φ h = t₁

theorem **lift_snd** : lift f₁ f₂ u φ h t₁ t₂ ht d w h₁ h₂ ≫ snd f₁ f₂ u φ h = t₂

theorem **comp_lift** : d ≫ lift f₁ f₂ u φ h t₁ t₂ ht d w h₁ h₂ = w ≫ incl f₁ f₂ u φ h

theorem **existsUnique_lift** : ∃! l : T ⟶ graphClosure f₁ f₂ u φ h, l ≫ fst f₁ f₂ u φ h = t₁ ∧ l ≫ snd f₁ f₂ u φ h = t₂

theorem **isProper_fst_comp** [IsProper f₁] [IsProper f₂] : IsProper (fst f₁ f₂ u φ h ≫ f₁)

theorem **isPullback_incl** [Flat u] [Mono u] [QuasiCompact u] [IsSeparated f₂] : IsPullback (𝟙 U) (incl f₁ f₂ u φ h) u (fst f₁ f₂ u φ h)

theorem **range_incl** [Flat u] [Mono u] [QuasiCompact u] [IsSeparated f₂] : Set.range (incl f₁ f₂ u φ h) = fst f₁ f₂ u φ h ⁻¹' Set.range u

theorem **opensRange_incl** [IsOpenImmersion u] [QuasiCompact u] [IsSeparated f₂] : (incl f₁ f₂ u φ h).opensRange = fst f₁ f₂ u φ h ⁻¹ᵁ u.opensRange

theorem **isIso_fst_restrict_of_le** [IsOpenImmersion u] [QuasiCompact u] [IsSeparated f₂] (V : Z₁.Opens) (hV : V ≤ u.opensRange) : IsIso (fst f₁ f₂ u φ h |_ V)

theorem **isIso_fst_of_extension** [IsSchemeTheoreticallyDominant u] [IsSeparated f₂] (ψ : Z₁ ⟶ Z₂) (hψ : f₁ = ψ ≫ f₂) (hu : u ≫ ψ = φ) : IsIso (fst f₁ f₂ u φ h)

theorem **exists_extension_of_isIso_fst** [IsIso (fst f₁ f₂ u φ h)] : ∃ ψ : Z₁ ⟶ Z₂, f₁ = ψ ≫ f₂ ∧ u ≫ ψ = φ

theorem **graphClosure.flat_of_isDedekindDomain** {O : CommRingCat.{u}} [IsDedekindDomain O] (f₁ : Z₁ ⟶ Spec O) (f₂ : Z₂ ⟶ Spec O) (u : U ⟶ Z₁) (φ : U ⟶ Z₂) (h : u ≫ f₁ = φ ≫ f₂) [QuasiCompact u] [QuasiSeparated f₂] [Flat (u ≫ f₁)] : Flat (graphClosure.fst f₁ f₂ u φ h ≫ f₁)

theorem **graphClosure.flat_of_valuationRing** {O : CommRingCat.{u}} [IsDomain O] [ValuationRing O] (f₁ : Z₁ ⟶ Spec O) (f₂ : Z₂ ⟶ Spec O) (u : U ⟶ Z₁) (φ : U ⟶ Z₂) (h : u ≫ f₁ = φ ≫ f₂) [QuasiCompact u] [QuasiSeparated f₂] [Flat (u ≫ f₁)] : Flat (graphClosure.fst f₁ f₂ u φ h ≫ f₁)

theorem **comm** : openPart ≫ str (-4) = partialMap ≫ str (-1)

theorem **isFinite_str** (d : ℤ) : IsFinite (str d)

theorem **isProper_and_not_isIso** : IsProper (graphClosure.fst (str (-4)) (str (-1)) openPart partialMap comm) ∧ IsIso (graphClosure.fst (str (-4)) (str (-1)) openPart partialMap comm |_ openPart.opensRange) ∧ ¬ IsIso (graphClosure.fst (str (-4)) (str (-1)) openPart partialMap comm)

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.GlActionMemBalFormsOfSemilinear

theorem **mem_symForms** {b : Bil Y A} : b ∈ symForms S Y A ↔ ∀ y y', b y y' = b y' y

theorem **mem_symForms_iff_flip** {b : Bil Y A} : b ∈ symForms S Y A ↔ b.flip = b

theorem **mem_symForms_iff_isSymm** {b : Bil Y ℤ} : b ∈ symForms ℤ Y ℤ ↔ LinearMap.BilinForm.IsSymm b

@[simp] theorem **pullback_apply** (f : Y →ₗ[ℤ] Y') (b : Bil Y' A) (y y' : Y) : pullback S A f b y y' = b (f y) (f y')

@[simp] theorem **pullback_id** : pullback S A (LinearMap.id : Y →ₗ[ℤ] Y) = LinearMap.id

theorem **pullback_comp** (f : Y →ₗ[ℤ] Y') (g : Y' →ₗ[ℤ] Y'') : pullback S A (g ∘ₗ f) = pullback S A f ∘ₗ pullback S A g

theorem **pullback_injective** {f : Y →ₗ[ℤ] Y'} (hf : Function.Surjective f) : Function.Injective (pullback S A f)

theorem **pullback_mem_symForms** (f : Y →ₗ[ℤ] Y') {b : Bil Y' A} (hb : b ∈ symForms S Y' A) : pullback S A f b ∈ symForms S Y A

theorem **pullback_mem_symForms_iff** {f : Y →ₗ[ℤ] Y'} (hf : Function.Surjective f) {b : Bil Y' A} : pullback S A f b ∈ symForms S Y A ↔ b ∈ symForms S Y' A

theorem **mem_range_pullback_iff** {f : Y →ₗ[ℤ] Y'} (hf : Function.Surjective f) {b : Bil Y A} (hb : b ∈ symForms S Y A) : b ∈ LinearMap.range (pullback S A f) ↔ LinearMap.ker f ≤ LinearMap.ker b

@[simp] theorem **coe_symPullback** (f : Y →ₗ[ℤ] Y') (b : symForms S Y' A) : (symPullback S A f b : Bil Y A) = pullback S A f b

theorem **symPullback_injective** {f : Y →ₗ[ℤ] Y'} (hf : Function.Surjective f) : Function.Injective (symPullback S A f)

@[simp] theorem **glAction_apply** (g : Y ≃ₗ[ℤ] Y) (b : Bil Y A) (y y' : Y) : glAction S Y A g b y y' = b (g.symm y) (g.symm y')

theorem **glAction_eq_pullback** (g : Y ≃ₗ[ℤ] Y) (b : Bil Y A) : glAction S Y A g b = pullback S A g.symm.toLinearMap b

@[simp] theorem **coe_restrictAction** {Γ : Type*} [Group Γ] (ρ : Γ →* (Y ≃ₗ[ℤ] Y)) (L : Submodule S (Bil Y A)) (hL : ∀ γ, ∀ b ∈ L, glAction S Y A (ρ γ) b ∈ L) (γ : Γ) (b : L) : (restrictAction ρ L hL γ b : Bil Y A) = glAction S Y A (ρ γ) b

theorem **glAction_mem_symForms** (g : Y ≃ₗ[ℤ] Y) {b : Bil Y A} (hb : b ∈ symForms S Y A) : glAction S Y A g b ∈ symForms S Y A

@[simp] theorem **symAction_apply** (g : Y ≃ₗ[ℤ] Y) (b : symForms S Y A) (y y' : Y) : (symAction S Y A g b : Bil Y A) y y' = (b : Bil Y A) (g.symm y) (g.symm y')

theorem **ofTensor_surjective** : Function.Surjective (ofTensor Y)

theorem **ker_ofTensor** : (ofTensor Y).ker = symRel Y

theorem **mk_apply** (y y' : Y) : mk y y' = ofTensor Y (y ⊗ₜ y')

theorem **mk_comm** (y y' : Y) : mk y y' = mk y' y

@[ext high] theorem **hom_ext** {A : Type*} [AddCommGroup A] {f g : SymSq Y →ₗ[ℤ] A} (h : ∀ y y', f (mk y y') = g (mk y y')) : f = g

theorem **span_mk** : Submodule.span ℤ {x : SymSq Y | ∃ y y', x = mk y y'} = ⊤

@[simp] theorem **lift_mk** (b : symForms S Y A) (y y' : Y) : lift S Y A b (mk y y') = (b : Bil Y A) y y'

@[simp] theorem **lift_symm_apply** (l : SymSq Y →ₗ[ℤ] A) (y y' : Y) : ((lift S Y A).symm l : Bil Y A) y y' = l (mk y y')

@[simp] theorem **map_mk** {Y' : Type*} [AddCommGroup Y'] (f : Y →ₗ[ℤ] Y') (y y' : Y) : map f (mk y y') = mk (f y) (f y')

theorem **lift_symPullback** {Y' : Type*} [AddCommGroup Y'] (f : Y →ₗ[ℤ] Y') (b : symForms S Y' A) : lift S Y A (symPullback S A f b) = lift S Y' A b ∘ₗ map f

@[simp] theorem **basis_apply** {ι : Type*} (e : Module.Basis ι ℤ Y) (i j : ι) : basis e s(i, j) = mk (e i) (e j)

theorem **finrank_eq** [Module.Finite ℤ Y] [Module.Free ℤ Y] : Module.finrank ℤ (SymSq Y) = (Module.finrank ℤ Y + 1).choose 2

@[simp] theorem **dualEquiv_apply** [Module.Finite ℤ Y] [Module.Free ℤ Y] (x : SymSq Y) (b : SymLattice Y) : dualEquiv Y x b = lift ℤ Y ℤ b x

theorem **dualEquiv_mk** [Module.Finite ℤ Y] [Module.Free ℤ Y] (y y' : Y) (b : SymLattice Y) : dualEquiv Y (mk y y') b = (b : Bil Y ℤ) y y'

@[simp] theorem **equivSymmetricPower_mk** (y y' : Y) : equivSymmetricPower Y (mk y y') = SymmetricPower.tprod ℤ ![y, y']

theorem **finrank_symLattice** [Module.Finite ℤ Y] [Module.Free ℤ Y] : Module.finrank ℤ (SymLattice Y) = (Module.finrank ℤ Y + 1).choose 2

theorem **IsBalanced.right** {b : Bil Y A} (hb : IsBalanced O b) (a : O) (y y' : Y) : b y (a • y') = b (star a • y) y'

theorem **IsBalanced.smul_smul** {b : Bil Y A} (hb : IsBalanced O b) (w : O) (y y' : Y) : b (w • y) (w • y') = b y ((w * star w) • y')

theorem **IsBalanced.smul_smul_of_norm** {b : Bil Y A} (hb : IsBalanced O b) {w : O} {n : ℤ} (hn : (n : O) = w * star w) (y y' : Y) : b (w • y) (w • y') = n • b y y'

theorem **IsBalanced.add_of_add_star** {b : Bil Y A} (hb : IsBalanced O b) {w : O} (hw : w + star w = 1) (y y' : Y) : b y (w • y') + b (w • y) y' = b y y'

theorem **IsBalanced.smul_mem_ker** {b : Bil Y A} (hb : IsBalanced O b) (a : O) {y : Y} (hy : y ∈ LinearMap.ker b) : a • y ∈ LinearMap.ker b

theorem **IsBalanced.mem_radical** {b : Bil Y A} (hb : IsBalanced O b) {y : Y} : y ∈ hb.radical ↔ b y = 0

theorem **mem_balForms** {b : Bil Y A} : b ∈ balForms S O Y A ↔ b ∈ symForms S Y A ∧ IsBalanced O b

theorem **balForms_le_symForms** : balForms S O Y A ≤ symForms S Y A

theorem **add_self_of_add_star** {b : Bil Y A} (hb : b ∈ balForms S O Y A) {w : O} (hw : w + star w = 1) (y : Y) : b y (w • y) + b y (w • y) = b y y

theorem **smul_smul_self_of_norm** {b : Bil Y A} (hb : b ∈ balForms S O Y A) {w : O} {n : ℤ} (hn : (n : O) = w * star w) (y : Y) : b (w • y) (w • y) = n • b y y

theorem **even_apply_self_of_add_star** {b : Bil Y ℤ} (hb : b ∈ balForms ℤ O Y ℤ) {w : O} (hw : w + star w = 1) (y : Y) : Even (b y y)

theorem **balForms_int** {Y : Type*} [AddCommGroup Y] : balForms S ℤ Y A = symForms S Y A

@[simp] theorem **traceForm_apply** (t : O →ₗ[ℤ] A) (x y : O) : traceForm t x y = t (x * star y)

theorem **traceForm_mem_balForms** {t : O →ₗ[ℤ] A} (ht : ∀ x, t (star x) = t x) : traceForm t ∈ balForms S O O A

theorem **mem_symForms_of_zsmul_mem** [Module.IsTorsionFree ℤ A] {Y : Type*} [AddCommGroup Y] {b : Bil Y A} {n : ℤ} (hn : n ≠ 0) (hb : n • b ∈ symForms S Y A) : b ∈ symForms S Y A

theorem **mem_balForms_of_zsmul_mem** [Module.IsTorsionFree ℤ A] {b : Bil Y A} {n : ℤ} (hn : n ≠ 0) (hb : n • b ∈ balForms S O Y A) : b ∈ balForms S O Y A

@[simp] theorem **coe_balInclusion** (b : BalLattice O Y) : (balInclusion O Y b : Bil Y ℤ) = b

theorem **balInclusion_injective** : Function.Injective (balInclusion O Y)

theorem **mem_range_balInclusion** {b : SymLattice Y} : b ∈ LinearMap.range (balInclusion O Y) ↔ IsBalanced O (b : Bil Y ℤ)

theorem **isSaturated_symForms** (Y : Type*) [AddCommGroup Y] : IsSaturatedSubmodule (symForms ℤ Y ℤ)

theorem **isSaturated_balForms** : IsSaturatedSubmodule (balForms ℤ O Y ℤ)

theorem **isSaturated_range_balInclusion** : IsSaturatedSubmodule (LinearMap.range (balInclusion O Y))

theorem **exists_isCompl_range_balInclusion** [Module.Finite ℤ Y] [Module.Free ℤ Y] : ∃ Q : Submodule ℤ (SymLattice Y), IsCompl (LinearMap.range (balInclusion O Y)) Q

theorem **IsBalanced.pullback_of_semilinear** {O' Z : Type*} [CommRing O'] [StarRing O'] [AddCommGroup Z] [Module O' Z] (σ : O → O') (hσ : ∀ a, σ (star a) = star (σ a)) (f : Y →ₗ[ℤ] Z) (hf : ∀ (a : O) y, f (a • y) = σ a • f y) {b : Bil Z A} (hb : IsBalanced O' b) : IsBalanced O (pullback S A f b)

theorem **pullback_mem_balForms** (f : Y →ₗ[O] Y') {b : Bil Y' A} (hb : b ∈ balForms S O Y' A) : pullback S A (f.restrictScalars ℤ) b ∈ balForms S O Y A

theorem **pullback_mem_balForms_iff** {f : Y →ₗ[O] Y'} (hf : Function.Surjective f) {b : Bil Y' A} : pullback S A (f.restrictScalars ℤ) b ∈ balForms S O Y A ↔ b ∈ balForms S O Y' A

@[simp] theorem **balPullback_apply** (f : Y →ₗ[O] Y') (b : balForms S O Y' A) (y y' : Y) : (balPullback S A f b : Bil Y A) y y' = (b : Bil Y' A) (f y) (f y')

theorem **balPullback_injective** {f : Y →ₗ[O] Y'} (hf : Function.Surjective f) : Function.Injective (balPullback S A f)

@[simp] theorem **autToGL_apply** (g : Y ≃ₗ[O] Y) (y : Y) : autToGL O Y g y = g y

@[simp] theorem **autToGL_symm_apply** (g : Y ≃ₗ[O] Y) (y : Y) : (autToGL O Y g).symm y = g.symm y

theorem **glAction_mem_balForms** (g : Y ≃ₗ[O] Y) {b : Bil Y A} (hb : b ∈ balForms S O Y A) : glAction S Y A (autToGL O Y g) b ∈ balForms S O Y A

theorem **glAction_mem_balForms_of_semilinear** (σ : O ≃+* O) (hσ : ∀ a, σ (star a) = star (σ a)) (g : Y ≃ₗ[ℤ] Y) (hg : ∀ (a : O) y, g (a • y) = σ a • g y) {b : Bil Y A} (hb : b ∈ balForms S O Y A) : glAction S Y A g b ∈ balForms S O Y A

@[simp] theorem **autAction_apply** (g : Y ≃ₗ[O] Y) (b : balForms S O Y A) (y y' : Y) : (autAction S O Y A g b : Bil Y A) y y' = (b : Bil Y A) (g.symm y) (g.symm y')

theorem **balInclusion_autAction** (g : Y ≃ₗ[O] Y) (b : BalLattice O Y) : balInclusion O Y (autAction ℤ O Y ℤ g b) = symAction ℤ Y ℤ (autToGL O Y g) (balInclusion O Y b)

theorem **mem_balRel** {x : SymSq Y} : x ∈ balRel O Y ↔ ∃ n : ℤ, n ≠ 0 ∧ n • x ∈ Submodule.span ℤ {x : SymSq Y | ∃ (a : O) (y y' : Y), x = SymSq.mk (a • y) y' − SymSq.mk y (star a • y')}

theorem **mk_apply** (y y' : Y) : mk O y y' = (balRel O Y).mkQ (SymSq.mk y y')

theorem **mk_mem_balForms** : mk O ∈ balForms ℤ O Y (BalSq O Y)

@[ext high] theorem **hom_ext** {f g : BalSq O Y →ₗ[ℤ] A} (h : ∀ y y', f (mk O y y') = g (mk O y y')) : f = g

@[simp] theorem **lift_mk** [Module.IsTorsionFree ℤ A] (b : balForms S O Y A) (y y' : Y) : lift S O Y A b (mk O y y') = (b : Bil Y A) y y'

@[simp] theorem **lift_symm_apply** [Module.IsTorsionFree ℤ A] (l : BalSq O Y →ₗ[ℤ] A) (y y' : Y) : ((lift S O Y A).symm l : Bil Y A) y y' = l (mk O y y')

@[simp] theorem **dualEquiv_apply** [Module.Finite ℤ Y] (x : BalSq O Y) (b : BalLattice O Y) : dualEquiv O Y x b = lift ℤ O Y ℤ b x

theorem **dualEquiv_mk** [Module.Finite ℤ Y] (y y' : Y) (b : BalLattice O Y) : dualEquiv O Y (mk O y y') b = (b : Bil Y ℤ) y y'

theorem **dualEquiv_mkQ** [Module.Finite ℤ Y] [Module.Free ℤ Y] (x : SymSq Y) : dualEquiv O Y ((balRel O Y).mkQ x) = (balInclusion O Y).dualMap (SymSq.dualEquiv Y x)

theorem **dualMap_balInclusion_surjective** [Module.Finite ℤ Y] [Module.Free ℤ Y] : Function.Surjective (balInclusion O Y).dualMap

theorem **mem_level** {M : ℤ} {L : Submodule ℤ (Bil Y ℤ)} {b : Bil Y ℤ} : b ∈ level M L ↔ ∃ c ∈ L, M • c = b

theorem **smul_mem_level** (M : ℤ) {L : Submodule ℤ (Bil Y ℤ)} {b : Bil Y ℤ} (hb : b ∈ L) : M • b ∈ level M L

theorem **level_le** (M : ℤ) (L : Submodule ℤ (Bil Y ℤ)) : level M L ≤ L

theorem **level_mono** (M : ℤ) {L L' : Submodule ℤ (Bil Y ℤ)} (h : L ≤ L') : level M L ≤ level M L'

@[simp] theorem **level_one** (L : Submodule ℤ (Bil Y ℤ)) : level 1 L = L

theorem **level_mul** (M N : ℤ) (L : Submodule ℤ (Bil Y ℤ)) : level (M * N) L = level M (level N L)

theorem **level_le_level_of_dvd** {M N : ℤ} (h : M ∣ N) (L : Submodule ℤ (Bil Y ℤ)) : level N L ≤ level M L

theorem **mem_level_top** {M : ℤ} {b : Bil Y ℤ} : b ∈ level M ⊤ ↔ ∀ y y', M ∣ b y y'

theorem **mem_level_of_saturated** {M : ℤ} {L : Submodule ℤ (Bil Y ℤ)} (hL : IsSaturatedSubmodule L) {b : Bil Y ℤ} : b ∈ level M L ↔ b ∈ L ∧ ∀ y y', M ∣ b y y'

theorem **mem_level_symForms** {M : ℤ} {b : Bil Y ℤ} : b ∈ level M (symForms ℤ Y ℤ) ↔ b ∈ symForms ℤ Y ℤ ∧ ∀ y y', M ∣ b y y'

theorem **mem_level_balForms** {O : Type*} [CommRing O] [StarRing O] [Module O Y] {M : ℤ} {b : Bil Y ℤ} : b ∈ level M (balForms ℤ O Y ℤ) ↔ b ∈ balForms ℤ O Y ℤ ∧ ∀ y y', M ∣ b y y'

theorem **level_balForms** (O : Type*) [CommRing O] [StarRing O] [Module O Y] (M : ℤ) : level M (balForms ℤ O Y ℤ) = level M (symForms ℤ Y ℤ) ⊓ balForms ℤ O Y ℤ

@[simp] theorem **coe_levelEquiv** {M : ℤ} (hM : M ≠ 0) (L : Submodule ℤ (Bil Y ℤ)) (b : L) : (levelEquiv hM L b : Bil Y ℤ) = M • (b : Bil Y ℤ)

theorem **relIndex_level** [Module.Finite ℤ Y] [Module.Free ℤ Y] (M : ℕ) (L : Submodule ℤ (Bil Y ℤ)) : (level M L).toAddSubgroup.relIndex L.toAddSubgroup = M ^ Module.finrank ℤ L

theorem **pullback_mem_level** (f : Y →ₗ[ℤ] Y') {L : Submodule ℤ (Bil Y ℤ)} {L' : Submodule ℤ (Bil Y' ℤ)} (h : ∀ b ∈ L', pullback ℤ ℤ f b ∈ L) (M : ℤ) {b : Bil Y' ℤ} (hb : b ∈ level M L') : pullback ℤ ℤ f b ∈ level M L

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.IrrelevantLeMapLinearGradedMap

@[simp] theorem **specOverMap_left** (g : A →ₐ[R] B) : (specOverMap g).left = Spec.map (CommRingCat.ofHom (g : A →+* B))

theorem **exists_eq_specOverMap** (x : specOver R B ⟶ specOver R A) : ∃ g : A →ₐ[R] B, x = specOverMap g

theorem **exists_specOver_nhds** (T : Over (Spec (.of R))) {t : T.left} {V : T.left.Opens} (ht : t ∈ V) : ∃ (A : Type u) (_ : CommRing A) (_ : Algebra R A) (j : specOver R A ⟶ T), IsOpenImmersion j.left ∧ t ∈ Set.range j.left ∧ j.left ⁻¹ᵁ V = ⊤

theorem **specOverMap_ptOver** {A B : Type u} [CommRing A] [CommRing B] [Algebra R A] [Algebra R B] (g : A →ₐ[R] B) (P : Fin 3 → A) (h : IsUnimodularSol (W.baseChange A) P) : specOverMap g ≫ ptOver W P h = ptOver W (g ∘ P) (h.baseChange g)

theorem **irrelevant_le_map_linearGradedMap** : irrelevant (grading W') ≤ (irrelevant (grading W)).map (linearGradedMap W L hL h)

theorem **isUnimodularSol_linear** : IsUnimodularSol (W.map ρ) (evalHom W' ρ P hP.equation ∘ L)

theorem **ptHom_linearHom** : ptHom W' ρ P hP ≫ linearHom W L hL h hs = ptHom W ρ (evalHom W' ρ P hP.equation ∘ L) (isUnimodularSol_linear W L h hs ρ P hP)

theorem **linearHom_toSpec** : linearHom W L hL h hs ≫ toSpec W = toSpec W'

theorem **equation_neg** {V : WeierstrassCurve A} {P : Fin 3 → A} (h : V.toProjective.Equation P) : V.toProjective.Equation (V.toProjective.neg P)

theorem **mem_span_neg** (V : WeierstrassCurve A) (P : Fin 3 → A) (i : Fin 3) : P i ∈ Ideal.span (Set.range (V.toProjective.neg P))

theorem **IsUnimodularSol.neg** {V : WeierstrassCurve A} {P : Fin 3 → A} (h : IsUnimodularSol V P) : IsUnimodularSol V (V.toProjective.neg P)

theorem **neg_coord_mem** (i : Fin 3) : (W.baseChange (HomRing W)).toProjective.neg (coord W) i ∈ grading W 1

theorem **ptHom_negHom** {A : Type u} [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) : ptHom W ρ P hP ≫ negHom W = ptHom W ρ ((W.map ρ).toProjective.neg P) hP.neg

theorem **negHom_toSpec** : negHom W ≫ toSpec W = toSpec W

@[simp] theorem **negOver_left** : (negOver W).left = negHom W

theorem **ptOver_negOver** {A : Type u} [CommRing A] [Algebra R A] (P : Fin 3 → A) (hP : IsUnimodularSol (W.baseChange A) P) : ptOver W P hP ≫ negOver W = ptOver W ((W.baseChange A).toProjective.neg P) hP.neg

theorem **negHom_mapHom** {R' : Type u} [CommRing R'] {W' : WeierstrassCurve R'} (f : R →+* R') (h : W.map f = W') : negHom W' ≫ mapHom W f h = mapHom W f h ≫ negHom W

@[simp] theorem **zeroUnit_left** : (zeroUnit W).left = zero W

theorem **toUnit_zeroUnit** (A : Type u) [CommRing A] [Algebra R A] : toUnit (specOver R A) ≫ zeroUnit W = zeroOver W A

theorem **isSumTriple_addXYZ** (hP : V.toProjective.Equation P) (hQ : V.toProjective.Equation Q) : IsSumTriple V P Q (V.toProjective.addXYZ P Q)

theorem **isSumTriple_diagXYZ** (hP : V.toProjective.Equation P) (hQ : V.toProjective.Equation Q) : IsSumTriple V P Q (diagXYZ V.toProjective P Q)

theorem **IsSumTriple.map** (h : IsSumTriple V P Q T) (g : A →+* B) : IsSumTriple (V.map g) (g ∘ P) (g ∘ Q) (g ∘ T)

theorem **IsSumTriple.baseChange** {R : Type*} [CommRing R] [Algebra R A] [Algebra R B] {W : WeierstrassCurve R} (h : IsSumTriple (W.baseChange A) P Q T) (g : A →ₐ[R] B) : IsSumTriple (W.baseChange B) (g ∘ P) (g ∘ Q) (g ∘ T)

theorem **IsSumTriple.equiv_add** {k : Type*} [Field k] {V : WeierstrassCurve k} {P Q T : Fin 3 → k} (h : IsSumTriple V P Q T) (hP : V.toProjective.Nonsingular P) (hQ : V.toProjective.Nonsingular Q) (hT : T ≠ 0) : T ≈ V.toProjective.add P Q

theorem **exists_law_notMem** [V.IsElliptic] (hP : IsUnimodularSol V P) (hQ : IsUnimodularSol V Q) (p : Ideal A) [p.IsPrime] : (∃ i, V.toProjective.addXYZ P Q i ∉ p) ∨ ∃ i, diagXYZ V.toProjective P Q i ∉ p

theorem **IsSumTriple.equation_of_isReduced** {A : Type u} [CommRing A] [_root_.IsReduced A] {V : WeierstrassCurve A} [V.IsElliptic] {P Q T : Fin 3 → A} (h : IsSumTriple V P Q T) (hP : IsUnimodularSol V P) (hQ : IsUnimodularSol V Q) : V.toProjective.Equation T

theorem **fieldAdd_ptOver** {k : Type u} [Field k] [Algebra R k] {P Q T : Fin 3 → k} (hP : IsUnimodularSol (W.baseChange k) P) (hQ : IsUnimodularSol (W.baseChange k) Q) (hT : IsUnimodularSol (W.baseChange k) T) (h : IsSumTriple (W.baseChange k) P Q T) : fieldAdd W k (ptOver W P hP) (ptOver W Q hQ) = ptOver W T hT

theorem **comp_ptOver_eq_fieldAdd** {A : Type u} [CommRing A] [Algebra R A] {P Q T : Fin 3 → A} (hP : IsUnimodularSol (W.baseChange A) P) (hQ : IsUnimodularSol (W.baseChange A) Q) (hT : IsUnimodularSol (W.baseChange A) T) (h : IsSumTriple (W.baseChange A) P Q T) (k : Type u) [Field k] [Algebra R k] (x : specOver R k ⟶ specOver R A) : x ≫ ptOver W T hT = fieldAdd W k (x ≫ ptOver W P hP) (x ≫ ptOver W Q hQ)

theorem **exists_nhds_eq_ptOver** {T : Over (Spec (.of R))} (p q : T ⟶ schemeOver W) (t : T.left) : ∃ (A : Type u) (_ : CommRing A) (_ : Algebra R A) (j : specOver R A ⟶ T) (P Q : Fin 3 → A) (hP : IsUnimodularSol (W.baseChange A) P) (hQ : IsUnimodularSol (W.baseChange A) Q), IsOpenImmersion j.left ∧ t ∈ Set.range j.left ∧ j ≫ p = ptOver W P hP ∧ j ≫ q = ptOver W Q hQ

theorem **exists_sum** {T : Over (Spec (.of R))} [IsReduced T.left] (p q : T ⟶ schemeOver W) : ∃ s : T ⟶ schemeOver W, ∀ (k : Type u) [Field k] [Algebra R k] (x : specOver R k ⟶ T), x ≫ s = fieldAdd W k (x ≫ p) (x ≫ q)

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.IsDomainAndIsIntegrallyClosedLocalizationOfIsInv

theorem **exists_invariant_mem_notMem** (q : Ideal A) [hq : q.IsPrime] (I : Ideal A) (h : ∀ g : G, ¬ I ≤ g • q) : ∃ t ∈ I, (∀ g : G, g • t = t) ∧ t ∉ q

theorem **isIntegrallyClosed_of_injective_of_dvd** {R S : Type*} [CommRing R] [CommRing S] [IsDomain R] [IsDomain S] [IsIntegrallyClosed S] (φ : R →+* S) (hφ : Function.Injective φ) (h : ∀ a b : R, b ≠ 0 → φ b ∣ φ a → b ∣ a) : IsIntegrallyClosed R

theorem **isDomain_and_isIntegrallyClosed_localization_of_isInvariant** (B : Type*) [CommRing B] [Algebra B A] [FaithfulSMul B A] [SMulCommClass G B A] [Algebra.IsInvariant B A G] (hA : ∀ (q : Ideal A) [q.IsPrime], IsDomain (Localization.AtPrime q) ∧

IsIntegrallyClosed (Localization.AtPrime q)) (p : Ideal B) [p.IsPrime] : IsDomain (Localization.AtPrime p) ∧ IsIntegrallyClosed (Localization.AtPrime p)

theorem **isInvariant_of_isBaseChange_of_sum_eq_one** {c : A} (hc : ∑ g : G, g • c = 1) (f : A →ₐ[B] A') (hb : IsBaseChange B' f.toLinearMap) (hf : ∀ (g : G) a, f (g • a) = g • f a) : FaithfulSMul B' A' ∧ Algebra.IsInvariant B' A' G

theorem **exists_apply_eq_apply_smul** (φ ψ : A →+* k) (h : ∀ a, (∀ g : G, g • a = a) → φ a = ψ a) (a : A) : ∃ g : G, ψ a = φ (g • a)

theorem **exists_forall_apply_eq_apply_smul** (φ ψ : A →+* k) (h : ∀ a, (∀ g : G, g • a = a) → φ a = ψ a) : ∃ g : G, ∀ a, ψ a = φ (g • a)

theorem **exists_ringHom_extension_of_isIntegral** {B A k : Type*} [CommRing B] [CommRing A] [Field k] [IsAlgClosed k] [Algebra B A] [FaithfulSMul B A] [Algebra.IsIntegral B A] (φ : B →+* k) : ∃ ψ : A →+* k, ψ.comp (algebraMap B A) = φ

theorem **isAffineQuotient_spec** [Finite G] (B A : Type u) [CommRing B] [CommRing A] [MulSemiringAction G A] [Algebra B A] [SMulCommClass G B A] [FaithfulSMul B A] [Algebra.IsInvariant B A G] : IsAffineQuotient (specAction G A) (Spec.map (CommRingCat.ofHom (algebraMap B A)))

theorem **epi_of_surjective_of_app_injective** (π : X ⟶ Y) [Surjective π] (h : ∀ W, Function.Injective (π.app W)) : Epi π

theorem **isSchemeTheoreticallyDominant_of_cover** (π : X ⟶ Y) [QuasiCompact π] (h : ∀ y : Y, ∃ W : Y.Opens, IsAffineOpen W ∧ y ∈ W ∧ Function.Injective (π.app W)) : IsSchemeTheoreticallyDominant π

theorem **of_cover** [Finite G] (hπ : ∀ g, (ρ g).hom ≫ π = π) (h : ∀ y : Y, ∃ W : Y.Opens, IsAffineOpen W ∧ y ∈ W ∧ IsAffineOpen (π ⁻¹ᵁ W) ∧ Function.Injective (π.app W) ∧ ∀ s : Γ(X, π ⁻¹ᵁ W), IsInvariantSection ρ _ s → s ∈ Set.range (π.app W)) : IsAffineQuotient ρ π

theorem **preimage_le** (W : Y.Opens) (g : G) : π ⁻¹ᵁ W ≤ (ρ g).hom ⁻¹ᵁ π ⁻¹ᵁ W

theorem **range_app** (W : Y.Opens) : Set.range (π.app W) = {s | IsInvariantSection ρ (π ⁻¹ᵁ W) s}

theorem **isSchemeTheoreticallyDominant** : IsSchemeTheoreticallyDominant π

theorem **isInvariant** (W : Y.Opens) : letI

theorem **isReduced** [IsReduced X] : IsReduced Y

theorem **isIntegralHom** : IsIntegralHom π where

theorem **surjective** : Surjective π

theorem **apply_eq_iff** (x y : X) : π x = π y ↔ ∃ g, (ρ g).hom x = y

theorem **isQuotientMap** : Topology.IsQuotientMap π

theorem **isOpenMap** : IsOpenMap π

theorem **exists_preimage_eq** {U : X.Opens} (hU : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) : ∃ W : Y.Opens, π ⁻¹ᵁ W = U

theorem **epi** : Epi π

theorem **exists_comp_eq** (f : X ⟶ Z) (hf : ∀ g, (ρ g).hom ≫ f = f) : ∃ f' : Y ⟶ Z, π ≫ f' = f

theorem **existsUnique_comp_eq** (f : X ⟶ Z) (hf : ∀ g, (ρ g).hom ≫ f = f) : ∃! f' : Y ⟶ Z, π ≫ f' = f

theorem **comp_desc** (f : X ⟶ Z) (hf : ∀ g, (ρ g).hom ≫ f = f) : π ≫ h.desc f hf = f

theorem **hom_ext** {a b : Y ⟶ Z} (e : π ≫ a = π ≫ b) : a = b

theorem **comp_uniqueIso_hom** {Y' : Scheme.{u}} {π' : X ⟶ Y'} (h' : IsAffineQuotient ρ π') : π ≫ (h.uniqueIso h').hom = π'

theorem **comp_uniqueIso_inv** {Y' : Scheme.{u}} {π' : X ⟶ Y'} (h' : IsAffineQuotient ρ π') : π' ≫ (h.uniqueIso h').inv = π

theorem **isIntegral** [IsIntegral X] : IsIntegral Y

theorem **isNormalScheme** [IsNormalScheme X] : IsNormalScheme Y

theorem **isFinite** [LocallyOfFiniteType (π ≫ q)] : IsFinite π

theorem **locallyOfFiniteType** [IsLocallyNoetherian S] [LocallyOfFiniteType (π ≫ q)] : LocallyOfFiniteType q

theorem **quasiCompact** [QuasiCompact (π ≫ q)] : QuasiCompact q

theorem **isSeparated** [IsSeparated (π ≫ q)] : IsSeparated q

theorem **isProper** [IsLocallyNoetherian S] [IsProper (π ≫ q)] : IsProper q

theorem **IsFreeAction.of_isAlgClosed** {ρ : G →* Aut X} (h : ∀ (k : Type u) [Field k] [IsAlgClosed k] (x : Spec (.of k) ⟶ X) (g : G), x ≫ (ρ g).hom = x → g = 1) : IsFreeAction ρ

theorem **IsFreeAction.eq_one** {ρ : G →* Aut X} (h : IsFreeAction ρ) {T : Scheme.{u}} [Nonempty T] (x : T ⟶ X) {g : G} (hx : x ≫ (ρ g).hom = x) : g = 1

theorem **IsFreeAction.of_equivariant** {ρ : G →* Aut X} (h : IsFreeAction ρ) {ρ' : G →* Aut Y} (p : Y ⟶ X) (hp : ∀ g, (ρ' g).hom ≫ p = p ≫ (ρ g).hom) : IsFreeAction ρ'

theorem **isFreeAction_specAction_iff** (A : Type u) [CommRing A] [MulSemiringAction G A] : IsFreeAction (specAction G A) ↔ IsFreeRingAction G A

theorem **specAction_fromSpec** (ρ : G →* Aut X) {U : X.Opens} (hU : IsAffineOpen U) (hst : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) (g : G) : letI

theorem **IsFreeAction.isFreeRingAction** {ρ : G →* Aut X} (h : IsFreeAction ρ) {U : X.Opens} (hU : IsAffineOpen U) (hst : ∀ g, U ≤ (ρ g).hom ⁻¹ᵁ U) : letI

theorem **isPullback_fromSpec_app** (π : X ⟶ Y) {W : Y.Opens} (hW : IsAffineOpen W) (hV : IsAffineOpen (π ⁻¹ᵁ W)) : IsPullback hV.fromSpec (Spec.map (π.app W)) π hW.fromSpec

theorem **isFinite_of_isFreeAction** : IsFinite π where

theorem **etale_of_isFreeAction** : Etale π

theorem **finrank_of_isFreeAction** (y : Y) : π.finrank y = Nat.card G

theorem **of_isPullback_of_flat_or_sum_eq_one** [Fintype G] (hW : ∀ (W : Y.Opens) (_ : IsAffineOpen W) (W' : Y'.Opens) (_ : IsAffineOpen W') (e : W' ≤ t ⁻¹ᵁ W), (t.appLE W W' e).hom.Flat ∨ letI

theorem **of_isPullback** [Flat t] : IsAffineQuotient ρ' π'

theorem **of_isPullback_of_isFreeAction** (hρ : IsFreeAction ρ) : IsAffineQuotient ρ' π'

theorem **pullbackAction_fst** (g : G) : (pullbackAction ρ f hf t g).hom ≫ pullback.fst f t = pullback.fst f t ≫ (ρ g).hom

theorem **pullbackAction_snd** (g : G) : (pullbackAction ρ f hf t g).hom ≫ pullback.snd f t = pullback.snd f t

theorem **isPullback_pullbackMap** (π : X ⟶ Y) (q : Y ⟶ S) (hq : π ≫ q = f) : IsPullback (pullback.fst f t) (pullback.map f t q t π (𝟙 _) (𝟙 _) (by simp [hq]) (by simp)) π (pullback.fst q t)

theorem **IsAffineQuotient.pullbackMap** [Flat t] : IsAffineQuotient (pullbackAction ρ f hf t) (pullback.map f t q t π (𝟙 _) (𝟙 _) (by simp [hq]) (by simp))

theorem **IsAffineQuotient.pullbackMap_of_isFreeAction** (hρ : IsFreeAction ρ) : IsAffineQuotient (pullbackAction ρ f hf t) (pullback.map f t q t π (𝟙 _) (𝟙 _) (by simp [hq]) (by simp))

theorem **_root_.GlobalTheory.exists_specMap_fromSpec_eq_of_mem** (x : Spec (.of k) ⟶ X) {U : X.Opens} (hU : IsAffineOpen U) (hx : x default ∈ U) : ∃ φ : Γ(X, U) ⟶ .of k, Spec.map φ ≫ hU.fromSpec = x

theorem **comp_eq_comp_iff** (x y : Spec (.of k) ⟶ X) : x ≫ π = y ≫ π ↔ ∃ g, x ≫ (ρ g).hom = y

theorem **exists_comp_eq_of_isAlgClosed** [IsAlgClosed k] (y : Spec (.of k) ⟶ Y) : ∃ x : Spec (.of k) ⟶ X, x ≫ π = y

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.IsOpenOfMemIrreducibleComponentsOfFinite

theorem **isNormalScheme_iff** (X : Scheme.{u}) : IsNormalScheme X ↔ ∀ x : X, IsDomain (X.presheaf.stalk x) ∧ IsIntegrallyClosed (X.presheaf.stalk x)

theorem **isNormalScheme_iff_pred** (X : Scheme.{0}) : GlobalTheory.IsNormalScheme X ↔ ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme X

theorem **IsNormalScheme.pred** (X : Scheme.{0}) [h : GlobalTheory.IsNormalScheme X] : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme X

theorem **IsNormalScheme.of_pred** {X : Scheme.{0}} (h : ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsNormalScheme X) : GlobalTheory.IsNormalScheme X

theorem **isDomain_and_isIntegrallyClosed_iff_of_ringEquiv** {R S : Type*} [CommRing R] [CommRing S] (e : R ≃+* S) : (IsDomain R ∧ IsIntegrallyClosed R) ↔ (IsDomain S ∧ IsIntegrallyClosed S)

theorem **of_isIso_stalkMap** {X Y : Scheme.{u}} (f : X ⟶ Y) [∀ x, IsIso (f.stalkMap x)] [IsNormalScheme Y] : IsNormalScheme X

theorem **of_isOpenImmersion** {X Y : Scheme.{u}} (f : X ⟶ Y) [IsOpenImmersion f] [IsNormalScheme Y] : IsNormalScheme X

theorem **of_isIso** {X Y : Scheme.{u}} (f : X ⟶ Y) [IsIso f] [IsNormalScheme X] : IsNormalScheme Y

theorem **iff_of_iso** {X Y : Scheme.{u}} (e : X ≅ Y) : IsNormalScheme X ↔ IsNormalScheme Y

theorem **of_forall_exists_isOpenImmersion** {X : Scheme.{u}} (h : ∀ x : X, ∃ (Y : Scheme.{u}) (f : Y ⟶ X) (_ : IsOpenImmersion f), IsNormalScheme Y ∧ x ∈ Set.range f) : IsNormalScheme X

theorem **of_forall_exists_opens** {X : Scheme.{u}} (h : ∀ x : X, ∃ U : X.Opens, x ∈ U ∧ IsNormalScheme U) : IsNormalScheme X

theorem **of_openCover** {X : Scheme.{u}} (𝒰 : X.OpenCover) [∀ i, IsNormalScheme (𝒰.X i)] : IsNormalScheme X

theorem **iff_of_openCover** {X : Scheme.{u}} (𝒰 : X.OpenCover) : IsNormalScheme X ↔ ∀ i, IsNormalScheme (𝒰.X i)

theorem **iff_of_iSup_eq_top** {X : Scheme.{u}} {ι : Sort*} (U : ι → X.Opens) (hU : iSup U = ⊤) : IsNormalScheme X ↔ ∀ i, IsNormalScheme (U i)

theorem **isIntegral_of_irreducibleSpace** (X : Scheme.{u}) [IsNormalScheme X] [IrreducibleSpace X] : IsIntegral X

theorem **of_isEmpty** (X : Scheme.{u}) [IsEmpty X] : IsNormalScheme X

theorem **spec_iff** (R : CommRingCat.{u}) : IsNormalScheme (Spec R) ↔ ∀ (p : Ideal R) [p.IsPrime], IsDomain (Localization.AtPrime p) ∧ IsIntegrallyClosed (Localization.AtPrime p)

theorem **spec_iff_of_isDomain** (R : CommRingCat.{u}) [IsDomain R] : IsNormalScheme (Spec R) ↔ IsIntegrallyClosed R

theorem **iff_spec_of_isAffineOpen** {X : Scheme.{u}} {U : X.Opens} (hU : IsAffineOpen U) : IsNormalScheme U ↔ IsNormalScheme (Spec Γ(X, U))

theorem **spec_of_isAffineOpen** {X : Scheme.{u}} [IsNormalScheme X] {U : X.Opens} (hU : IsAffineOpen U) : IsNormalScheme (Spec Γ(X, U))

theorem **iff_forall_isAffineOpen** (X : Scheme.{u}) : IsNormalScheme X ↔ ∀ U : X.Opens, IsAffineOpen U → ∀ (p : Ideal Γ(X, U)) [p.IsPrime], IsDomain (Localization.AtPrime p) ∧ IsIntegrallyClosed (Localization.AtPrime p)

theorem **of_isIntegral_of_isIntegrallyClosed** (X : Scheme.{u}) [IsIntegral X] (h : ∀ (U : X.Opens) [Nonempty U], IsAffineOpen U → IsIntegrallyClosed Γ(X, U)) : IsNormalScheme X

theorem **isDiscreteValuationRing_stalk** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] {x : X} (hx : Order.coheight x = 1) : IsDiscreteValuationRing (X.presheaf.stalk x)

theorem **isDiscreteValuationRing_stalk_iff** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] {x : X} : IsDiscreteValuationRing (X.presheaf.stalk x) ↔ Order.coheight x = 1

theorem **eq_of_mem_irreducibleComponents_of_isDomain** {X : Scheme.{u}} {x : X} [IsDomain (X.presheaf.stalk x)] {Z W : Set X} (hZ : Z ∈ irreducibleComponents X) (hW : W ∈ irreducibleComponents X) (hxZ : x ∈ Z) (hxW : x ∈ W) : Z = W

theorem **eq_of_mem_irreducibleComponents** {X : Scheme.{u}} [IsNormalScheme X] {x : X} {Z W : Set X} (hZ : Z ∈ irreducibleComponents X) (hW : W ∈ irreducibleComponents X) (hxZ : x ∈ Z) (hxW : x ∈ W) : Z = W

theorem **pairwiseDisjoint_irreducibleComponents** (X : Scheme.{u}) [IsNormalScheme X] : (irreducibleComponents X).PairwiseDisjoint id

theorem **irreducibleComponent_eq** {X : Scheme.{u}} [IsNormalScheme X] {x : X} {Z : Set X} (hZ : Z ∈ irreducibleComponents X) (hx : x ∈ Z) : irreducibleComponent x = Z

theorem **isOpen_of_mem_irreducibleComponents_of_finite** {X : Scheme.{u}} [IsNormalScheme X] (hX : (irreducibleComponents X).Finite) {Z : Set X} (hZ : Z ∈ irreducibleComponents X) : IsOpen Z

theorem **isOpen_of_mem_irreducibleComponents** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] {Z : Set X} (hZ : Z ∈ irreducibleComponents X) : IsOpen Z

theorem **isClopen_of_mem_irreducibleComponents** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] {Z : Set X} (hZ : Z ∈ irreducibleComponents X) : IsClopen Z

theorem **connectedComponent_eq_irreducibleComponent** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] (x : X) : connectedComponent x = irreducibleComponent x

theorem **isIntegral_of_connectedSpace** (X : Scheme.{u}) [IsNormalScheme X] [IsLocallyNoetherian X] [ConnectedSpace X] : IsIntegral X

theorem **coe_component** (Z : irreducibleComponents X) : (component Z : Set X) = Z.1

theorem **mem_component** {Z : irreducibleComponents X} {x : X} : x ∈ component Z ↔ x ∈ Z.1

theorem **isClosed_component** (Z : irreducibleComponents X) : IsClosed (component Z : Set X)

theorem **iSup_component** : ⨆ Z : irreducibleComponents X, component Z = ⊤

theorem **pairwise_disjoint_component** : Pairwise fun Z W : irreducibleComponents X ↦ Disjoint (component Z) (component W)

theorem **componentCover_X** (Z : irreducibleComponents X) : (componentCover X).X Z = (component Z).toScheme

theorem **componentCover_f** (Z : irreducibleComponents X) : (componentCover X).f Z = (component Z).ι

theorem **nonempty_isColimit_component** : Nonempty (IsColimit (Cofan.mk X fun Z : irreducibleComponents X ↦ (component Z).ι))

theorem **isIntegrallyClosed_of_isAffineOpen** {X : Scheme.{u}} [IsNormalScheme X] {U : X.Opens} (hU : IsAffineOpen U) [IsDomain Γ(X, U)] : IsIntegrallyClosed Γ(X, U)

theorem **exists_germToFunctionField_eq_of_locally** {X : Scheme.{u}} [IsIntegral X] (U : X.Opens) [Nonempty U] (t : X.functionField) (h : ∀ x ∈ U, ∃ (V : X.Opens) (_ : Nonempty V) (s : Γ(X, V)), x ∈ V ∧ V ≤ U ∧ X.germToFunctionField V s = t) : ∃ s : Γ(X, U), X.germToFunctionField U s = t

theorem **iff_isIntegrallyClosed_sections** (X : Scheme.{u}) [IsIntegral X] : IsNormalScheme X ↔ ∀ (U : X.Opens) [Nonempty U], IsIntegrallyClosed Γ(X, U)

theorem **isDomain_and_isIntegrallyClosed_of_isIrreducible** {X : Scheme.{u}} [IsNormalScheme X] {U : X.Opens} (hU : IsIrreducible (U : Set X)) : IsDomain Γ(X, U) ∧ IsIntegrallyClosed Γ(X, U)

theorem **isIntegrallyClosedIn_localization** {A B A_m B_m : Type*} [CommRing A] [CommRing B] [CommRing A_m] [CommRing B_m] [Algebra A B] [Algebra A A_m] [Algebra B B_m] [Algebra A_m B_m] [Algebra A B_m] [IsScalarTower A B B_m] [IsScalarTower A A_m B_m] (M : Submonoid A) [IsLocalization M A_m] [IsLocalization (Algebra.algebraMapSubmonoid B M) B_m] [IsIntegrallyClosedIn A B] : IsIntegrallyClosedIn A_m B_m

theorem **isDomain_and_isIntegrallyClosed_away** {A B C : Type*} [CommRing A] [CommRing B] [CommRing C] [Algebra A B] [Algebra B C] [IsIntegrallyClosedIn A B] (a : A) [IsLocalization.Away (algebraMap A B a) C] [IsDomain C] [IsIntegrallyClosed C] : IsDomain (Localization.Away a) ∧ IsIntegrallyClosed (Localization.Away a)

theorem **exists_isIdempotentElem_basicOpen_eq** {X : Scheme.{u}} {U V W : X.Opens} (hU : V ⊔ W = U) (hVW : Disjoint V W) : ∃ e : Γ(X, U), IsIdempotentElem e ∧ X.basicOpen e = V ∧ X.basicOpen (1 - e) = W

theorem **exists_algebraMap_eq_of_isIdempotentElem** {A B : Type*} [CommRing A] [CommRing B] [Algebra A B] [IsIntegrallyClosedIn A B] {e : B} (he : IsIdempotentElem e) : ∃ a : A, algebraMap A B a = e

theorem **spec_of_isIntegrallyClosedIn** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] {U : X.Opens} (hU : IsCompact (U : Set X)) (A : Type u) [CommRing A] [Algebra A Γ(X, U)] [IsIntegrallyClosedIn A Γ(X, U)] : IsNormalScheme (Spec (.of A))

theorem **spec_sections_of_isCompact** {X : Scheme.{u}} [IsNormalScheme X] [IsLocallyNoetherian X] {U : X.Opens} (hU : IsCompact (U : Set X)) : IsNormalScheme (Spec Γ(X, U))

theorem **spec_of_isIntegrallyClosedIn_of_isIntegral** {X : Scheme.{u}} [IsNormalScheme X] [IsIntegral X] (U : X.Opens) (A : Type u) [CommRing A] [Algebra A Γ(X, U)] [IsIntegrallyClosedIn A Γ(X, U)] : IsNormalScheme (Spec (.of A))

theorem **normalization_of_isIntegral** {X Y : Scheme.{u}} (f : X ⟶ Y) [QuasiCompact f] [QuasiSeparated f] [IsNormalScheme X] [IsIntegral X] : IsNormalScheme f.normalization

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.MemPrimitiveTensorsIffIsPrimitiveElem

theorem **ext_apply** {φ ψ : CartierDual R M} (h : ∀ a, φ a = ψ a) : φ = ψ

@[simp] theorem **toDual_apply** (φ : CartierDual R M) : toDual R M φ = φ.ofConv

@[simp] theorem **toDual_symm_apply** (φ : Module.Dual R M) : (toDual R M).symm φ = toConv φ

@[simp] theorem **evalAt_apply** (a : M) (φ : CartierDual R M) : evalAt R M a φ = φ a

@[simp] theorem **pair_tmul_tmul** (φ : CartierDual R M) (ψ : CartierDual R N) (a : M) (b : N) : pair R M N (φ ⊗ₜ ψ) (a ⊗ₜ b) = φ a * ψ b

theorem **pair_tmul** (φ : CartierDual R M) (ψ : CartierDual R N) (t : M ⊗[R] N) : pair R M N (φ ⊗ₜ ψ) t = LinearMap.mul' R R (TensorProduct.map φ.ofConv ψ.ofConv t)

theorem **pair_bijective** [Module.Finite R M] [Module.Projective R M] [Module.Finite R N] [Module.Projective R N] : Function.Bijective (pair R M N)

@[simp] theorem **pairEquiv_apply** [Module.Finite R M] [Module.Projective R M] [Module.Finite R N] [Module.Projective R N] (x : CartierDual R M ⊗[R] CartierDual R N) : pairEquiv R M N x = pair R M N x

theorem **tensor_ext** [Module.Finite R M] [Module.Projective R M] [Module.Finite R N] [Module.Projective R N] {x y : CartierDual R M ⊗[R] CartierDual R N} (h : ∀ a b, pair R M N x (a ⊗ₜ b) = pair R M N y (a ⊗ₜ b)) : x = y

theorem **eq_of_forall_apply** [Module.Projective R M] {a b : M} (h : ∀ φ : CartierDual R M, φ a = φ b) : a = b

theorem **tensor_eq_of_forall_pair** [Module.Finite R M] [Module.Projective R M] [Module.Finite R N] [Module.Projective R N] {s t : M ⊗[R] N} (h : ∀ φ ψ, pair R M N (φ ⊗ₜ ψ) s = pair R M N (φ ⊗ₜ ψ) t) : s = t

private theorem **pair₃_tmul** (φ : CartierDual R M) (x : CartierDual R N ⊗[R] CartierDual R P) (a : M) (t : N ⊗[R] P) : pair₃ (φ ⊗ₜ x) (a ⊗ₜ t) = φ a * pair R N P x t

private theorem **pair₃_assoc** (x : CartierDual R M ⊗[R] CartierDual R N) (χ : CartierDual R P) (a : M) (b : N) (c : P) : pair₃ (TensorProduct.assoc R _ _ _ (x ⊗ₜ χ)) (a ⊗ₜ (b ⊗ₜ c)) = pair R M N x (a ⊗ₜ b) * χ c

private theorem **pair₃_ext** [Module.Finite R M] [Module.Projective R M] [Module.Finite R N] [Module.Projective R N] [Module.Finite R P] [Module.Projective R P] {x y : CartierDual R M ⊗[R] (CartierDual R N ⊗[R] CartierDual R P)} (h : ∀ a b c, pair₃ x (a ⊗ₜ (b ⊗ₜ c)) = pair₃ y (a ⊗ₜ (b ⊗ₜ c))) : x = y

@[simp] theorem **counit_apply** (φ : CartierDual R H) : counit (R

theorem **pair_comul** (φ : CartierDual R H) : pair R H H (comul φ) = φ.ofConv ∘ₗ LinearMap.mul' R H

@[simp] theorem **pair_comul_tmul** (φ : CartierDual R H) (a b : H) : pair R H H (comul φ) (a ⊗ₜ b) = φ (a * b)

theorem **comul_eq_iff** (φ : CartierDual R H) (x : CartierDual R H ⊗[R] CartierDual R H) : comul φ = x ↔ ∀ a b, pair R H H x (a ⊗ₜ b) = φ (a * b)

theorem **pair_comm** {R M N : Type*} [CommSemiring R] [AddCommMonoid M] [Module R M] [AddCommMonoid N] [Module R N] (x : CartierDual R M ⊗[R] CartierDual R N) (a : M) (b : N) : pair R N M (TensorProduct.comm R _ _ x) (b ⊗ₜ a) = pair R M N x (a ⊗ₜ b)

theorem **mul_apply** (φ ψ : CartierDual R C) (a : C) : (φ * ψ) a = pair R C C (φ ⊗ₜ ψ) (comul a)

theorem **one_apply** (a : C) : (1 : CartierDual R C) a = counit a

theorem **mul'_apply** (x : CartierDual R C ⊗[R] CartierDual R C) (a : C) : (LinearMap.mul' R _ x) a = pair R C C x (comul a)

theorem **pair_mul** (x y : CartierDual R C ⊗[R] CartierDual R C') : toConv (pair R C C' (x * y)) = toConv (pair R C C' x) * toConv (pair R C C' y)

@[simp] theorem **antipode_apply** (φ : CartierDual R H) (a : H) : (antipode R φ : CartierDual R H) a = φ (antipode R a)

@[simp] theorem **transpose_apply** (f : M →ₗ[R] N) (φ : CartierDual R N) (a : M) : transpose f φ a = φ (f a)

theorem **pair_map_transpose** (f : M →ₗ[R] N) (g : P →ₗ[R] Q) (x : CartierDual R N ⊗[R] CartierDual R Q) (t : M ⊗[R] P) : pair R M P (TensorProduct.map (transpose f) (transpose g) x) t = pair R N Q x (TensorProduct.map f g t)

theorem **transpose_injective** {f : M →ₗ[R] N} (hf : Function.Surjective f) : Function.Injective (transpose f)

@[simp] theorem **toBidual_apply** (a : M) (φ : CartierDual R M) : cartierDualToBidual R M a φ = φ a

theorem **toBidual_bijective** [Module.Finite R M] [Module.Projective R M] : Function.Bijective (cartierDualToBidual R M)

theorem **pair_map_toBidual** (t : M ⊗[R] N) (x : CartierDual R M ⊗[R] CartierDual R N) : pair R _ _ (TensorProduct.map (cartierDualToBidual R M) (cartierDualToBidual R N) t) x = pair R M N x t

theorem **transpose_bijective** {f : M →ₗ[R] N} (hf : Function.Bijective f) : Function.Bijective (transpose f)

theorem **transpose_bijective_iff** [Module.Finite R M] [Module.Projective R M] [Module.Finite R N] [Module.Projective R N] (f : M →ₗ[R] N) : Function.Bijective (transpose f) ↔ Function.Bijective f

@[simp] theorem **mapAlgHom_apply** (f : C →ₗc[R] C') (φ : CartierDual R C') (a : C) : mapAlgHom f φ a = φ (f a)

@[simp] theorem **mapCoalgHom_apply** (f : H →ₐ[R] K) (φ : CartierDual R K) (a : H) : mapCoalgHom f φ a = φ (f a)

@[simp] theorem **map_apply** (f : H →ₐc[R] K) (φ : CartierDual R K) (a : H) : map f φ a = φ (f a)

@[simp] theorem **map_id** : map (BialgHom.id R H) = BialgHom.id R _

theorem **map_comp** (g : K →ₐc[R] L) (f : H →ₐc[R] K) : map (g.comp f) = (map f).comp (map g)

theorem **map_injective** : Function.Injective (map : (H →ₐc[R] K) → _)

theorem **map_injective_of_surjective** {f : H →ₐc[R] K} (hf : Function.Surjective f) : Function.Injective (map f)

@[simp] theorem **congr_apply** (e : H ≃ₐc[R] K) (φ : CartierDual R K) (a : H) : congr e φ a = φ (e a)

@[simp] theorem **congr_symm_apply** (e : H ≃ₐc[R] K) (φ : CartierDual R H) (a : K) : (congr e).symm φ a = φ (e.symm a)

@[simp] theorem **bidualHom_apply** (a : H) (φ : CartierDual R H) : bidualHom R H a φ = φ a

@[simp] theorem **bidual_apply** (a : H) (φ : CartierDual R H) : bidual R H a φ = φ a

@[simp] theorem **apply_bidual_symm** (x : CartierDual R (CartierDual R H)) (φ : CartierDual R H) : φ ((bidual R H).symm x) = x φ

theorem **map_map_bidual** (f : H →ₐc[R] K) (a : H) : map (map f) (bidual R H a) = bidual R K (f a)

@[simp] theorem **baseChangeLinear_tmul_tmul** (a b : A) (φ : CartierDual R M) (m : M) : baseChangeLinear R A M (a ⊗ₜ φ) (b ⊗ₜ m) = φ m • (a * b)

theorem **baseChangeLinear_bijective** [Module.Finite R M] [Module.Projective R M] : Function.Bijective (baseChangeLinear R A M)

private theorem **pair_baseChange** (x : CartierDual R M ⊗[R] CartierDual R N) (t : M ⊗[R] N) : pair A (A ⊗[R] M) (A ⊗[R] N) (TensorProduct.map (baseChangeLinear R A M) (baseChangeLinear R A N) (AlgebraTensorModule.tensorTensorTensorComm R A R A A A _ _ ((1 ⊗ₜ[A] 1) ⊗ₜ[R] x))) (AlgebraTensorModule.tensorTensorTensorComm R A R A A A _ _ ((1 ⊗ₜ[A] 1) ⊗ₜ[R] t)) = algebraMap R A (pair R M N x t)

theorem **baseChangeLinear_one** : baseChangeLinear R A C 1 = 1

theorem **baseChangeLinear_mul** (x y : A ⊗[R] CartierDual R C) : baseChangeLinear R A C (x * y) = baseChangeLinear R A C x * baseChangeLinear R A C y

@[simp] theorem **baseChange_tmul_tmul** (a b : A) (φ : CartierDual R H) (h : H) : baseChange R A H (a ⊗ₜ φ) (b ⊗ₜ h) = φ h • (a * b)

theorem **baseChange_map** {K : Type*} [Semiring K] [Bialgebra R K] [Module.Finite R K] [Module.Projective R K] (f : H →ₐc[R] K) (x : A ⊗[R] CartierDual R K) : baseChange R A H (Bialgebra.TensorProduct.map (BialgHom.id A A) (map f) x) = map (Bialgebra.TensorProduct.map (BialgHom.id A A) f) (baseChange R A K x)

private theorem **pair_tensorTensorTensorComm** {R M N : Type*} [CommSemiring R] [AddCommMonoid M] [Module R M] [AddCommMonoid N] [Module R N] (s : CartierDual R M ⊗[R] CartierDual R M) (t : CartierDual R N ⊗[R] CartierDual R N) (a : M) (b : N) (c : M) (d : N) : pair R _ _ (TensorProduct.map ((toDual R _).symm.toLinearMap ∘ₗ pair R M N) ((toDual R _).symm.toLinearMap ∘ₗ pair R M N) (AlgebraTensorModule.tensorTensorTensorComm R R R R _ _ _ _ (s ⊗ₜ t))) ((a ⊗ₜ b) ⊗ₜ (c ⊗ₜ d)) = pair R M M s (a ⊗ₜ c) * pair R N N t (b ⊗ₜ d)

@[simp] theorem **tensorEquiv_apply** (x : CartierDual R H ⊗[R] CartierDual R K) (t : H ⊗[R] K) : tensorEquiv R H K x t = pair R H K x t

@[simp] theorem **tensorToHom_tmul** (n : N) (m : M) (φ : CartierDual R M) : tensorToHom R N M (n ⊗ₜ m) φ = φ m • n

theorem **tensorToHom_apply** (x : N ⊗[R] M) (φ : CartierDual R M) : tensorToHom R N M x φ = TensorProduct.rid R N (LinearMap.lTensor N φ.ofConv x)

theorem **tensorToHom_map** (g : N →ₗ[R] N') (f : M →ₗ[R] M') (x : N ⊗[R] M) (φ : CartierDual R M') : tensorToHom R N' M' (TensorProduct.map g f x) φ = g (tensorToHom R N M x (transpose f φ))

theorem **apply_tensorToHom** (x : N ⊗[R] M) (ψ : CartierDual R N) (φ : CartierDual R M) : ψ (tensorToHom R N M x φ) = pair R N M (ψ ⊗ₜ φ) x

theorem **apply_tensorToHom_comm** (x : N ⊗[R] M) (ψ : CartierDual R N) (φ : CartierDual R M) : φ (tensorToHom R M N (TensorProduct.comm R N M x) ψ) = pair R N M (ψ ⊗ₜ φ) x

theorem **tensorToHom_bijective** [Module.Finite R M] [Module.Projective R M] : Function.Bijective (tensorToHom R N M)

@[simp] theorem **tensorEquivHom_apply** [Module.Finite R M] [Module.Projective R M] (x : N ⊗[R] M) : tensorEquivHom R N M x = tensorToHom R N M x

@[simp] theorem **tensorToHom_tensorEquivHom_symm** [Module.Finite R M] [Module.Projective R M] (f : CartierDual R M →ₗ[R] N) : tensorToHom R N M ((tensorEquivHom R N M).symm f) = f

theorem **eq_of_forall_tensorToHom** [Module.Finite R M] [Module.Projective R M] {x y : N ⊗[R] M} (h : ∀ φ, tensorToHom R N M x φ = tensorToHom R N M y φ) : x = y

theorem **eq_of_forall_tensorToHom_pair** [Module.Finite R M] [Module.Projective R M] [Module.Finite R M'] [Module.Projective R M'] {x y : N ⊗[R] (M ⊗[R] M')} (h : ∀ φ ψ, tensorToHom R N _ x (toConv (pair R M M' (φ ⊗ₜ ψ))) = tensorToHom R N _ y (toConv (pair R M M' (φ ⊗ₜ

ψ)))) : x = y

theorem **baseChangeLinear_tmul_apply** {A : Type*} [CommSemiring A] [Algebra R A] (a : A) (φ : CartierDual R M) (x : A ⊗[R] M) : baseChangeLinear R A M (a ⊗ₜ φ) x = a • tensorToHom R A M x φ

theorem **isGroupLikeElem_iff_mul** (a : C) : IsGroupLikeElem R a ↔ (1 : CartierDual R C) a = 1 ∧ ∀ φ ψ : CartierDual R C, (φ * ψ) a = φ a * ψ a

theorem **isGroupLikeElem_tensor_iff_mul** (x : A ⊗[R] C) : IsGroupLikeElem A x ↔ tensorToHom R A C x 1 = 1 ∧ ∀ φ ψ, tensorToHom R A C x (φ * ψ) = tensorToHom R A C x φ * tensorToHom R A C x ψ

@[simp] theorem **pointsEquiv_symm_apply** (x : GroupLike A (A ⊗[R] C)) (φ : CartierDual R C) : (pointsEquiv R C A).symm x φ = tensorToHom R A C x.val φ

@[simp] theorem **tensorToHom_pointsEquiv** (f : CartierDual R C →ₐ[R] A) (φ : CartierDual R C) : tensorToHom R A C (pointsEquiv R C A f).val φ = f φ

theorem **pointsEquiv_eq_iff** (f : CartierDual R C →ₐ[R] A) (x : A ⊗[R] C) : (pointsEquiv R C A f).val = x ↔ ∀ φ, tensorToHom R A C x φ = f φ

theorem **pointsEquiv_comp** {B : Type*} [CommSemiring B] [Algebra R B] (g : A →ₐ[R] B) (f : CartierDual R C →ₐ[R] A) : (pointsEquiv R C B (g.comp f)).val = LinearMap.rTensor C g.toLinearMap (pointsEquiv R C A f).val

theorem **tensorToHom_mul** {R H A : Type*} [CommSemiring R] [Semiring H] [Algebra R H] [Module.Finite R H] [Module.Projective R H] [Semiring A] [Algebra R A] (x y : A ⊗[R] H) : toConv (tensorToHom R A H (x * y)) = toConv (tensorToHom R A H x) * toConv (tensorToHom R A H y)

@[simp] theorem **pointsMulEquiv_apply** (f : WithConv (CartierDual R H →ₐ[R] A)) : pointsMulEquiv R H A f = pointsEquiv R H A f.ofConv

@[simp] theorem **pointsMulEquiv_symm_apply** (x : GroupLike A (A ⊗[R] H)) : (pointsMulEquiv R H A).symm x = toConv ((pointsEquiv R H A).symm x)

@[simp] theorem **tensorToHom_dualPointsEquiv** (f : H →ₐ[R] A) (a : H) : tensorToHom R A _ (dualPointsEquiv R H A f).val (bidual R H a) = f a

theorem **isGroupLikeElem_dual_iff** {R H : Type*} [CommSemiring R] [Semiring H] [Algebra R H] [Module.Finite R H] [Module.Projective R H] (φ : CartierDual R H) : IsGroupLikeElem R φ ↔ φ 1 = 1 ∧ ∀ a b, φ (a * b) = φ a * φ b

private theorem **tmul_tmul_one_eq** {R A M N : Type*} [CommSemiring R] [CommSemiring A] [Algebra R A] [AddCommMonoid M] [Module R M] [AddCommMonoid N] [Module R N] (a : A) (m : M) (n : N) : (a ⊗ₜ[R] m) ⊗ₜ[A] ((1 : A) ⊗ₜ[R] n) = ((1 : A) ⊗ₜ[R] m) ⊗ₜ[A] (a ⊗ₜ[R] n)

private theorem **distribBaseChange_lTensor_comul** {R A C : Type*} [CommSemiring R] [CommSemiring A] [Algebra R A] [AddCommMonoid C] [Module R C] [Coalgebra R C] (x : A ⊗[R] C) : AlgebraTensorModule.distribBaseChange R A C C (LinearMap.lTensor A comul x) = comul (R

@[simp] theorem **primitiveComul_apply** (a : H) : primitiveComul R H a = a ⊗ₜ[R] 1 + 1 ⊗ₜ[R] a

@[simp] theorem **mem_primitiveElems** {a : H} : a ∈ primitiveElems R H ↔ IsPrimitiveElem R a

theorem **IsPrimitiveElem.map** {a : H} (ha : IsPrimitiveElem R a) (f : H →ₐc[R] K) : IsPrimitiveElem R (f a)

theorem **mem_primitiveTensors** {M : Type*} [AddCommMonoid M] [Module R M] {x : M ⊗[R] H} : x ∈ primitiveTensors R M H ↔ LinearMap.lTensor M comul x = LinearMap.lTensor M (primitiveComul R H) x

theorem **mem_primitiveTensors_iff_isPrimitiveElem** {A : Type*} [CommSemiring A] [Algebra R A] {x : A ⊗[R] H} : x ∈ primitiveTensors R A H ↔ IsPrimitiveElem A x

theorem **mem_primitiveTensors_self** {x : R ⊗[R] H} : x ∈ primitiveTensors R R H ↔ IsPrimitiveElem R (TensorProduct.lid R H x)

theorem **IsPrimitiveElem.counit_eq_zero** {R H : Type*} [CommRing R] [Semiring H] [Bialgebra R H] {a : H} (ha : IsPrimitiveElem R a) : counit (R

theorem **mem_primitiveTensors_iff** {M : Type*} [AddCommMonoid M] [Module R M] (x : M ⊗[R] H) : x ∈ primitiveTensors R M H ↔ ∀ φ ψ : CartierDual R H, tensorToHom R M H x (φ * ψ) = φ 1 • tensorToHom R M H x ψ + ψ 1 • tensorToHom R M H x φ

theorem **isPrimitiveElem_iff_mul** (a : H) : IsPrimitiveElem R a ↔ ∀ φ ψ : CartierDual R H, (φ * ψ) a = φ 1 * ψ a + ψ 1 * φ a

@[simp] theorem **tensorToHom_tangentEquiv** (f : {f : CartierDual R H →ₐ[R] TrivSqZeroExt A M // ∀ φ, (f φ).fst = algebraMap R A (counit φ)}) (φ : CartierDual R H) : tensorToHom R M H (tangentEquiv R H A M f).1 φ = (f.1 φ).snd

@[simp] theorem **snd_tangentEquiv_symm** (x : primitiveTensors R M H) (φ : CartierDual R H) : (((tangentEquiv R H A M).symm x).1 φ).snd = tensorToHom R M H x.1 φ

@[simp] theorem **tensorToHom_dualNumberEquiv** (f : {f : CartierDual R H →ₐ[R] DualNumber A // ∀ φ, (f φ).fst = algebraMap R A (counit φ)}) (φ : CartierDual R H) : tensorToHom R A H (dualNumberEquiv R H A f).1 φ = (f.1 φ).snd

@[simp] theorem **apply_primitiveEquiv** (f : {f : CartierDual R H →ₐ[R] DualNumber R // ∀ φ, (f φ).fst = counit φ}) (φ : CartierDual R H) : φ (primitiveEquiv R H f).1 = (f.1 φ).snd

@[simp] theorem **snd_primitiveEquiv_symm** (a : primitiveElems R H) (φ : CartierDual R H) : (((primitiveEquiv R H).symm a).1 φ).snd = φ a.1

@[simp] theorem **monoidAlgebraEquiv_apply** (φ : CartierDual R (MonoidAlgebra R Γ)) (γ : Γ) : monoidAlgebraEquiv R Γ φ γ = φ (MonoidAlgebra.single γ 1)

@[simp] theorem **monoidAlgebraEquiv_symm_single** (f : Γ → R) (γ : Γ) (r : R) : (monoidAlgebraEquiv R Γ).symm f (MonoidAlgebra.single γ r) = r * f γ

@[simp] theorem **addMonoidAlgebraEquiv_apply** (φ : CartierDual R (AddMonoidAlgebra R Γ)) (γ : Γ) : addMonoidAlgebraEquiv R Γ φ γ = φ (AddMonoidAlgebra.single γ 1)

@[simp] theorem **addMonoidAlgebraEquiv_symm_single** (f : Γ → R) (γ : Γ) (r : R) : (addMonoidAlgebraEquiv R Γ).symm f (AddMonoidAlgebra.single γ r) = r * f γ

theorem **counit_addMonoidAlgebra** [AddMonoid Γ] [Finite Γ] (φ : CartierDual R (AddMonoidAlgebra R Γ)) : counit (R

theorem **antipode_addMonoidAlgebra** [AddGroup Γ] [Finite Γ] (φ : CartierDual R (AddMonoidAlgebra R Γ)) (γ : Γ) : addMonoidAlgebraEquiv R Γ (antipode R φ) γ = addMonoidAlgebraEquiv R Γ φ (-γ)

theorem **isGroupLikeElem_tensor_iff** [Semiring K] [Bialgebra R K] (x : A ⊗[R] K) : IsGroupLikeElem A x ↔ TensorProduct.rid R A (LinearMap.lTensor A counit x) = 1 ∧ LinearMap.lTensor A comul x = Algebra.TensorProduct.map (AlgHom.id R A) (includeLeft : K →ₐ[R] K ⊗[R] K) x * Algebra.TensorProduct.map (AlgHom.id R A) (includeRight : K →ₐ[R] K ⊗[R] K) x

private theorem **productMap_convMul** (x : A ⊗[R] K) (g : A →ₐ[R] B) (y y' : WithConv (K →ₐ[R] B)) : productMap g (y * y').ofConv x = productMap g (productMap y.ofConv y'.ofConv) (LinearMap.lTensor A comul x)

theorem **productMap_convMul_of_isGroupLikeElem** (hx : IsGroupLikeElem A x) (g : A →ₐ[R] B) (y y' : WithConv (K →ₐ[R] B)) : productMap g (y * y').ofConv x = productMap g y.ofConv x * productMap g y'.ofConv x

@[simp] theorem **pairingMap_tmul** (a : H) (b : K) (φ : CartierDual R H) : pairingMap R (a ⊗ₜ[R] b) φ = φ a • b

theorem **apply_pairingMap** (u : H ⊗[R] K) (φ : CartierDual R H) (ψ : CartierDual R K) : ψ (pairingMap R u φ) = pair R H K (φ ⊗ₜ ψ) u

theorem **pairingMap_comm** (u : H ⊗[R] K) : pairingMap R (TensorProduct.comm R H K u) = tensorToHom R H K u

theorem **apply_pairingMap_comm** (u : H ⊗[R] K) (φ : CartierDual R H) (ψ : CartierDual R K) : φ (pairingMap R (TensorProduct.comm R H K u) ψ) = ψ (pairingMap R u φ)

theorem **symm** (hu : IsBicharacter R u) : IsBicharacter R (TensorProduct.comm R H K u)

theorem **one** : IsBicharacter R (1 : H ⊗[R] K)

theorem **mul** (hu : IsBicharacter R u) (hv : IsBicharacter R v) : IsBicharacter R (u * v)

theorem **isGroupLikeElem_iff_pairingMap** (u : H ⊗[R] K) : IsGroupLikeElem H u ↔ counit ∘ₗ pairingMap R u = counit ∧ TensorProduct.map (pairingMap R u) (pairingMap R u) ∘ₗ comul = comul ∘ₗ pairingMap R u

@[simp] theorem **IsBicharacter.toBialgHom_apply** (hu : IsBicharacter R u) (φ : CartierDual R H) : hu.toBialgHom φ = pairingMap R u φ

@[simp] theorem **pairingMapEquiv_apply** (u : H ⊗[R] K) : pairingMapEquiv R H K u = pairingMap R u

@[simp] theorem **pairingMap_pairingMapEquiv_symm** (f : CartierDual R H →ₗ[R] K) : pairingMap R ((pairingMapEquiv R H K).symm f) = f

theorem **isBicharacter_iff_exists_bialgHom** (u : H ⊗[R] K) : IsBicharacter R u ↔ ∃ f : CartierDual R H →ₐc[R] K, ∀ φ, f φ = pairingMap R u φ

@[simp] theorem **bicharacterEquiv_apply** (u : {u : H ⊗[R] K // IsBicharacter R u}) : bicharacterEquiv R H K u = u.2.toBialgHom

@[simp] theorem **pairingMap_bicharacterEquiv_symm** (f : CartierDual R H →ₐc[R] K) (φ : CartierDual R H) : pairingMap R ((bicharacterEquiv R H K).symm f).1 φ = f φ

theorem **pairingMap_comm_bijective_iff** (u : H ⊗[R] K) : Function.Bijective (pairingMap R (TensorProduct.comm R H K u)) ↔ Function.Bijective (pairingMap R u)

theorem **IsPerfectBicharacter.symm** (hu : IsPerfectBicharacter R u) : IsPerfectBicharacter R (TensorProduct.comm R H K u)

@[simp] theorem **IsPerfectBicharacter.toBialgEquiv_apply** (hu : IsPerfectBicharacter R u) (φ : CartierDual R H) : hu.toBialgEquiv φ = pairingMap R u φ

theorem **IsBicharacter.eq_one_of_isGroupLikeElem** (hu : IsBicharacter R u) (h : IsGroupLikeElem R u) : u = 1

theorem **tensorToHom_comm_canonical** : tensorToHom R _ H (TensorProduct.comm R _ _ (canonical R H)) = LinearMap.id

@[simp] theorem **pairingMap_canonical** : pairingMap R (canonical R H) = LinearMap.id

theorem **isPerfectBicharacter_canonical** : IsPerfectBicharacter R (canonical R H)

theorem **isGroupLikeElem_tensor_map** {A B K K' : Type*} [CommSemiring A] [Algebra R A] [CommSemiring B] [Algebra R B] [Semiring K] [Bialgebra R K] [Semiring K'] [Bialgebra R K'] {x : A ⊗[R] K} (hx : IsGroupLikeElem A x) (g : A →ₐ[R] B) (f : K →ₐc[R] K') : IsGroupLikeElem B (Algebra.TensorProduct.map g (f : K →ₐ[R] K') x)

theorem **IsBicharacter.map** {H' K' : Type*} [CommSemiring H'] [Bialgebra R H'] [CommSemiring K'] [Bialgebra R K'] (hu : IsBicharacter R u) (f : H →ₐc[R] H') (g : K →ₐc[R] K') : IsBicharacter R (Algebra.TensorProduct.map (f : H →ₐ[R] H') (g : K →ₐ[R] K') u)

theorem **pairingMap_pairingBaseChange** (u : H ⊗[R] K) (x : A ⊗[R] CartierDual R H) : pairingMap A (pairingBaseChange A u) (baseChangeLinear R A H x) = (pairingMap R u).baseChange A x

theorem **IsBicharacter.baseChange** (hu : IsBicharacter R u) : IsBicharacter A (pairingBaseChange A u)

theorem **IsPerfectBicharacter.baseChange** (hu : IsPerfectBicharacter R u) : IsPerfectBicharacter A (pairingBaseChange A u)

theorem **canonical_zmod_two_ne_one** (R : Type*) [CommRing R] [Nontrivial R] : canonical R (AddMonoidAlgebra R (ZMod 2)) ≠ 1

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.MulLeftCompOfPullbackPointEquivMul

theorem **hom_ext_of_fromSpecStalk** {X Y : Scheme.{u}} {f g : X ⟶ Y} (h : ∀ x, X.fromSpecStalk x ≫ f = X.fromSpecStalk x ≫ g) : f = g

theorem **hom_ext_of_isLocalRing** {A : Type u} [CommRing A] {Y : Scheme.{u}} {f g : Spec (.of A) ⟶ Y} (h : ∀ (B : Type u) [CommRing B] [IsLocalRing B] (φ : A →+* B), Spec.map (CommRingCat.ofHom φ) ≫ f = Spec.map (CommRingCat.ofHom φ) ≫ g) : f = g

theorem **exists_over_hom_of_locally** {T Y : Over (Spec (.of R))} [IsReduced T.left] [IsSeparated Y.hom] (F : ∀ (k : Type u) [Field k] [Algebra R k], (specOver R k ⟶ T) → (specOver R k ⟶ Y)) (H : ∀ t : T.left, ∃ (U : Over (Spec (.of R))) (j : U ⟶ T) (_ : IsOpenImmersion j.left) (f : U ⟶ Y), t ∈ Set.range j.left ∧ ∀ (k : Type u) [Field k] [Algebra R k] (x : specOver R k ⟶ U), x ≫ f = F k (x ≫ j)) : ∃ f : T ⟶ Y, ∀ (k : Type u) [Field k] [Algebra R k] (x : specOver R k ⟶ T), x ≫ f = F k x

@[simp] theorem **pullbackPointEquiv_apply_left** (h : IsPullback φ G'.hom G.hom f) {T : Over S'} (x : T ⟶ G') : (pullbackPointEquiv h T x).left = x.left ≫ φ

theorem **pullbackPointEquiv_mul** [GrpObj G] (h : IsPullback φ G'.hom G.hom f) {T : Over S'} (x y : T ⟶ G') : letI

theorem **isCommMonObj_of_isPullback** [GrpObj G] [IsCommMonObj G] (h : IsPullback φ G'.hom G.hom f) : letI

theorem **mul_left_comp_of_pullbackPointEquiv_mul** [GrpObj G] [GrpObj G'] (h : IsPullback φ G'.hom G.hom f) {T' : Over S'} {T : Over S} (t : T'.left ⟶ T.left) {x' y' : T' ⟶ G'} {x y : T ⟶ G} (hm : pullbackPointEquiv h T' (x' * y') = pullbackPointEquiv h T' x' * pullbackPointEquiv h T' y') (hx : x'.left ≫ φ = t ≫ x.left) (hy : y'.left ≫ φ = t ≫ y.left) : (x' * y').left ≫ φ = t ≫ (x * y).left

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.OpensRangeChartYSupOpensRangeChartZ

theorem **eval₂_smul_of_isHomogeneous** {σ R A : Type*} [CommRing R] [CommRing A] (ρ : R →+* A) {p : MvPolynomial σ R} {n : ℕ} (hp : p.IsHomogeneous n) (c : A) (P : σ → A) : MvPolynomial.eval₂ ρ (c • P) p = c ^ n * MvPolynomial.eval₂ ρ P p

theorem **IsUnimodularSol.of_isUnit** {V : WeierstrassCurve A} {P : Fin 3 → A} (h : V.toProjective.Equation P) (i : Fin 3) (hi : IsUnit (P i)) : IsUnimodularSol V P

theorem **isUnimodularSol_zero** (V : WeierstrassCurve A) : IsUnimodularSol V ![0, 1, 0]

theorem **isUnimodularSol_some** {V : WeierstrassCurve A} {a b : A} (h : V.toAffine.Equation a b) : IsUnimodularSol V ![a, b, 1]

theorem **equation_origin_iff** (V : WeierstrassCurve A) (z w : A) : V.toProjective.Equation ![z, -1, w] ↔ w = z ^ 3 + V.a₁ * z * w + V.a₂ * z ^ 2 * w + V.a₃ * w ^ 2 + V.a₄ * z * w ^ 2 + V.a₆ * w ^ 3

theorem **isUnimodularSol_origin** {V : WeierstrassCurve A} {z w : A} (h : V.toProjective.Equation ![z, -1, w]) : IsUnimodularSol V ![z, -1, w]

theorem **IsUnimodularSol.map** {V : WeierstrassCurve A} {P : Fin 3 → A} (h : IsUnimodularSol V P) (g : A →+* B) : IsUnimodularSol (V.map g) (g ∘ P)

theorem **IsUnimodularSol.map_comp** {R : Type*} [CommRing R] {W : WeierstrassCurve R} {ρ : R →+* A} {P : Fin 3 → A} (h : IsUnimodularSol (W.map ρ) P) (g : A →+* B) : IsUnimodularSol (W.map (g.comp ρ)) (g ∘ P)

theorem **IsUnimodularSol.baseChange** {R : Type*} [CommRing R] [Algebra R A] [Algebra R B] {W : WeierstrassCurve R} {P : Fin 3 → A} (h : IsUnimodularSol (W.baseChange A) P) (g : A →ₐ[R] B) : IsUnimodularSol (W.baseChange B) (g ∘ P)

theorem **IsUnimodularSol.of_baseChange** {R : Type*} [CommRing R] [Algebra R A] {W : WeierstrassCurve R} {P : Fin 3 → A} (h : IsUnimodularSol (W.baseChange A) P) : IsUnimodularSol (W.map (algebraMap R A)) P

theorem **IsUnimodularSol.smul** {V : WeierstrassCurve A} {P : Fin 3 → A} (h : IsUnimodularSol V P) {c : A} (hc : IsUnit c) : IsUnimodularSol V (c • P)

theorem **isHomogeneous_polynomial** : W.toProjective.polynomial.IsHomogeneous 3

theorem **isHomogeneous_cubicIdeal** : (cubicIdeal W).IsHomogeneous (MvPolynomial.homogeneousSubmodule (Fin 3) R)

theorem **mem_grading_iff** {n : ℕ} {a : HomRing W} : a ∈ grading W n ↔ ∃ p : MvPolynomial (Fin 3) R, p.IsHomogeneous n ∧ Ideal.Quotient.mk _ p = a

theorem **coord_mem** (i : Fin 3) : coord W i ∈ grading W 1

theorem **homRing_hom_ext** {A : Type*} [CommRing A] {f g : HomRing W →+* A} (h₀ : ∀ r : R, f (algebraMap R _ r) = g (algebraMap R _ r)) (h₁ : ∀ i, f (coord W i) = g (coord W i)) : f = g

theorem **equation_coord** : (W.baseChange (HomRing W)).toProjective.Equation (coord W)

theorem **irrelevant_eq_span_coord** : (irrelevant (grading W)).toIdeal = Ideal.span (Set.range (coord W))

theorem **evalHom_mk** (p : MvPolynomial (Fin 3) R) : evalHom W ρ P h (Ideal.Quotient.mk _ p) = MvPolynomial.eval₂ ρ P p

@[simp] theorem **evalHom_coord** (i : Fin 3) : evalHom W ρ P h (coord W i) = P i

@[simp] theorem **evalHom_algebraMap** (r : R) : evalHom W ρ P h (algebraMap R _ r) = ρ r

theorem **comp_evalHom** (g : A →+* B) : g.comp (evalHom W ρ P h) = evalHom W (g.comp ρ) (g ∘ P) (h.map g)

theorem **evalHom_smul** (c : A) (h' : (W.map ρ).toProjective.Equation (c • P)) {n : ℕ} {a : HomRing W} (ha : a ∈ grading W n) : evalHom W ρ (c • P) h' a = c ^ n * evalHom W ρ P h a

theorem **map_irrelevant_evalHom** (h : IsUnimodularSol (W.map ρ) P) : (irrelevant (grading W)).toIdeal.map (evalHom W ρ P h.equation) = ⊤

theorem **bijective_algebraMap_grading_zero** : Function.Bijective (algebraMap R (grading W 0))

theorem **over_eq_toSpec** : scheme W ↘ Spec (.of R) = toSpec W

theorem **coordOpen_one_sup_coordOpen_two** : coordOpen W 1 ⊔ coordOpen W 2 = ⊤

theorem **ptHom_toSpec** : ptHom W ρ P h ≫ toSpec W = Spec.map (CommRingCat.ofHom ρ)

theorem **SpecMap_ptHom** (g : A →+* B) : Spec.map (CommRingCat.ofHom g) ≫ ptHom W ρ P h = ptHom W (g.comp ρ) (g ∘ P) (h.map_comp g)

theorem **ptHom_smul** {c : A} (hc : IsUnit c) : ptHom W ρ (c • P) (h.smul hc) = ptHom W ρ P h

theorem **ptHom_preimage_coordOpen** (i : Fin 3) : ptHom W ρ P h ⁻¹ᵁ coordOpen W i = PrimeSpectrum.basicOpen (P i)

theorem **ptHom_congr** {ρ' : R →+* A} {P' : Fin 3 → A} (hρ : ρ = ρ') (hP : P = P') (h' : IsUnimodularSol (W.map ρ') P') : ptHom W ρ P h = ptHom W ρ' P' h'

theorem **pt_eq_ptHom** : pt W P h = ptHom W (algebraMap R A) P h.of_baseChange

theorem **pt_toSpec** : pt W P h ≫ toSpec W = Spec.map (CommRingCat.ofHom (algebraMap R A))

theorem **SpecMap_pt** (g : A →ₐ[R] B) : Spec.map (CommRingCat.ofHom (g : A →+* B)) ≫ pt W P h = pt W (g ∘ P) (h.baseChange g)

theorem **pt_smul** {c : A} (hc : IsUnit c) : pt W (c • P) (h.smul hc) = pt W P h

theorem **pt_preimage_coordOpen** (i : Fin 3) : pt W P h ⁻¹ᵁ coordOpen W i = PrimeSpectrum.basicOpen (P i)

@[simp] theorem **ptOver_left** : (ptOver W P h).left = pt W P h

theorem **pt_congr** {P' : Fin 3 → A} (hP : P = P') (h' : IsUnimodularSol (W.baseChange A) P') : pt W P h = pt W P' h'

theorem **zero_toSpec** : zero W ≫ toSpec W = 𝟙 _

theorem **SpecMap_zero** {A : Type u} [CommRing A] (ρ : R →+* A) : Spec.map (CommRingCat.ofHom ρ) ≫ zero W = ptHom W ρ ![0, 1, 0] (isUnimodularSol_zero _)

theorem **SpecMap_algebraMap_zero** (A : Type u) [CommRing A] [Algebra R A] : Spec.map (CommRingCat.ofHom (algebraMap R A)) ≫ zero W = pt W (A

theorem **zero_preimage_coordOpen_two** : zero W ⁻¹ᵁ coordOpen W 2 = ⊥

theorem **zero_preimage_coordOpen_one** : zero W ⁻¹ᵁ coordOpen W 1 = ⊤

@[simp] theorem **awayCoord_self** : awayCoord W i i = 1

theorem **algebraMap_comp_awayConst** : (algebraMap (AwayRing W i) (Localization.Away (coord W i))).comp (awayConst W i) = (algebraMap (HomRing W) _).comp (algebraMap R (HomRing W))

theorem **algebraMap_comp_awayCoord** : algebraMap (AwayRing W i) (Localization.Away (coord W i)) ∘ awayCoord W i = (IsLocalization.Away.invSelf (coord W i) : Localization.Away (coord W i)) • (algebraMap (HomRing W) _ ∘ coord W)

private theorem **isUnit_invSelf** : IsUnit (IsLocalization.Away.invSelf (coord W i) : Localization.Away (coord W i))

theorem **equation_awayCoord** : (W.map (awayConst W i)).toProjective.Equation (awayCoord W i)

theorem **isUnimodularSol_awayCoord** : IsUnimodularSol (W.map (awayConst W i)) (awayCoord W i)

theorem **evalHom_awayCoord** (n : ℕ) (a : HomRing W) (ha : a ∈ grading W (n • 1)) : evalHom W (awayConst W i) (awayCoord W i) (equation_awayCoord W i) a = Away.mk (grading W) (coord_mem W i) n a ha

theorem **awayRing_hom_ext** {A : Type*} [CommRing A] {f g : AwayRing W i →+* A} (h₀ : ∀ r : R, f (awayConst W i r) = g (awayConst W i r)) (h₁ : ∀ j, f (awayCoord W i j) = g (awayCoord W i j)) : f = g

@[simp] theorem **awayEval_awayConst** (h : (W.map ρ).toProjective.Equation P) (hi : IsUnit (P i)) (r : R) : awayEval W i ρ P h hi (awayConst W i r) = ρ r

theorem **awayEval_awayCoord_mul** (h : (W.map ρ).toProjective.Equation P) (hi : IsUnit (P i)) (j : Fin 3) : awayEval W i ρ P h hi (awayCoord W i j) * P i = P j

theorem **opensRange_awayChart** : (awayChart W i).opensRange = coordOpen W i

theorem **isAffineOpen_coordOpen** : IsAffineOpen (coordOpen W i)

theorem **ptHom_eq_awayChart** (h : IsUnimodularSol (W.map ρ) P) (hi : IsUnit (P i)) : ptHom W ρ P h = Spec.map (CommRingCat.ofHom (awayEval W i ρ P h.equation hi)) ≫ awayChart W i

theorem **awayChart_eq_ptHom** : awayChart W i = ptHom W (awayConst W i) (awayCoord W i) (isUnimodularSol_awayCoord W i)

theorem **SpecMap_awayChart** (g : AwayRing W i →+* A) : Spec.map (CommRingCat.ofHom g) ≫ awayChart W i = ptHom W (g.comp (awayConst W i)) (g ∘ awayCoord W i) ((isUnimodularSol_awayCoord W i).map_comp g)

theorem **awayChart_toSpec** : awayChart W i ≫ toSpec W = Spec.map (CommRingCat.ofHom (awayConst W i))

theorem **isOpenImmersion_ptHom** (h : IsUnimodularSol (W.map ρ) P) (hi : IsUnit (P i)) (hb : Function.Bijective (awayEval W i ρ P h.equation hi)) : IsOpenImmersion (ptHom W ρ P h)

theorem **range_ptHom** (h : IsUnimodularSol (W.map ρ) P) (hi : IsUnit (P i)) (hb : Function.Bijective (awayEval W i ρ P h.equation hi)) : Set.range (ptHom W ρ P h) = coordOpen W i

theorem **affine_equation_map_iff** (ρ : R →+* A) (a b : A) : (W.map ρ).toAffine.Equation a b ↔ W.toAffine.polynomial.eval₂ (eval₂RingHom ρ a) b = 0

theorem **equation_affineX_affineY** : (W.baseChange W.toAffine.CoordinateRing).toAffine.Equation (affineX W) (affineY W)

@[simp] theorem **affineEval_affineX** : affineEval W ρ a b h (affineX W) = a

@[simp] theorem **affineEval_affineY** : affineEval W ρ a b h (affineY W) = b

@[simp] theorem **affineEval_algebraMap** (r : R) : affineEval W ρ a b h (algebraMap R _ r) = ρ r

theorem **affine_hom_ext** {f g : W.toAffine.CoordinateRing →+* A} (h₀ : ∀ r : R, f (algebraMap R _ r) = g (algebraMap R _ r)) (hx : f (affineX W) = g (affineX W)) (hy : f (affineY W) = g (affineY W)) : f = g

@[simp] theorem **coe_affineAeval** [Algebra R A] (a b : A) (h : (W.baseChange A).toAffine.Equation a b) : (affineAeval W a b h : W.toAffine.CoordinateRing →+* A) = affineEval W (algebraMap R A) a b h

theorem **bijective_awayEval_affine** : Function.Bijective (awayEval W 2 (algebraMap R _) ![affineX W, affineY W, 1] (isUnimodularSol_some (equation_affineX_affineY W)).equation isUnit_one)

theorem **opensRange_chartZ** : (chartZ W).opensRange = coordOpen W 2

theorem **chartZ_toSpec** : chartZ W ≫ toSpec W = Spec.map (CommRingCat.ofHom (algebraMap R W.toAffine.CoordinateRing))

theorem **SpecMap_affineEval_chartZ** {A : Type u} [CommRing A] (ρ : R →+* A) (a b : A) (h : (W.map ρ).toAffine.Equation a b) : Spec.map (CommRingCat.ofHom (affineEval W ρ a b h)) ≫ chartZ W = ptHom W ρ ![a, b, 1] (isUnimodularSol_some h)

theorem **SpecMap_affineAeval_chartZ** {A : Type u} [CommRing A] [Algebra R A] (a b : A) (h : (W.baseChange A).toAffine.Equation a b) : Spec.map (CommRingCat.ofHom (affineAeval W a b h : W.toAffine.CoordinateRing →+* A)) ≫ chartZ W = pt W ![a, b, 1] (isUnimodularSol_some h)

theorem **chartZ_eq** : chartZ W = Spec.map (CommRingCat.ofHom (awayEquivAffine W : AwayRing W 2 →+* W.toAffine.CoordinateRing)) ≫ awayChart W 2

theorem **monic_originPolynomial** : (originPolynomial W).Monic

theorem **natDegree_originPolynomial** [Nontrivial R] : (originPolynomial W).natDegree = 3

theorem **origin_equation_map_iff** (ρ : R →+* A) (z w : A) : (W.map ρ).toProjective.Equation ![z, -1, w] ↔ (originPolynomial W).eval₂ (eval₂RingHom ρ w) z = 0

theorem **equation_originZ_originW** : (W.baseChange (OriginRing W)).toProjective.Equation ![originZ W, -1, originW W]

@[simp] theorem **originEval_originZ** : originEval W ρ z w h (originZ W) = z

@[simp] theorem **originEval_originW** : originEval W ρ z w h (originW W) = w

@[simp] theorem **originEval_algebraMap** (r : R) : originEval W ρ z w h (algebraMap R _ r) = ρ r

theorem **origin_hom_ext** {f g : OriginRing W →+* A} (h₀ : ∀ r : R, f (algebraMap R _ r) = g (algebraMap R _ r)) (hz : f (originZ W) = g (originZ W)) (hw : f (originW W) = g (originW W)) : f = g

@[simp] theorem **coe_originAeval** [Algebra R A] (z w : A) (h : (W.baseChange A).toProjective.Equation ![z, -1, w]) : (originAeval W z w h : OriginRing W →+* A) = originEval W (algebraMap R A) z w h

theorem **bijective_awayEval_origin** : Function.Bijective (awayEval W 1 (algebraMap R _) ![originZ W, -1, originW W] (equation_originZ_originW W) isUnit_one.neg)

theorem **opensRange_chartY** : (chartY W).opensRange = coordOpen W 1

theorem **chartY_toSpec** : chartY W ≫ toSpec W = Spec.map (CommRingCat.ofHom (algebraMap R (OriginRing W)))

theorem **SpecMap_originEval_chartY** {A : Type u} [CommRing A] (ρ : R →+* A) (z w : A) (h : (W.map ρ).toProjective.Equation ![z, -1, w]) : Spec.map (CommRingCat.ofHom (originEval W ρ z w h)) ≫ chartY W = ptHom W ρ ![z, -1, w] (isUnimodularSol_origin h)

theorem **SpecMap_originAeval_chartY** {A : Type u} [CommRing A] [Algebra R A] (z w : A) (h : (W.baseChange A).toProjective.Equation ![z, -1, w]) : Spec.map (CommRingCat.ofHom (originAeval W z w h : OriginRing W →+* A)) ≫ chartY W = pt W ![z, -1, w] (isUnimodularSol_origin h)

theorem **chartY_eq** : chartY W = Spec.map (CommRingCat.ofHom (awayEquivOrigin W : AwayRing W 1 →+* OriginRing W)) ≫ awayChart W 1

@[simp] theorem **originAug_originZ** : originAug W (originZ W) = 0

@[simp] theorem **originAug_originW** : originAug W (originW W) = 0

theorem **SpecMap_originAug_chartY** : Spec.map (CommRingCat.ofHom (originAug W : OriginRing W →+* R)) ≫ chartY W = zero W

theorem **opensRange_chartY_sup_opensRange_chartZ** : (chartY W).opensRange ⊔ (chartZ W).opensRange = ⊤

theorem **chartY_preimage_coordOpen_two** : chartY W ⁻¹ᵁ coordOpen W 2 = PrimeSpectrum.basicOpen (originW W)

theorem **chartZ_preimage_coordOpen_one** : chartZ W ⁻¹ᵁ coordOpen W 1 = PrimeSpectrum.basicOpen (affineY W)

theorem **exists_chartY_or_chartZ** (x : scheme W) : (∃ y, chartY W y = x) ∨ ∃ y, chartZ W y = x

theorem **scheme_hom_ext** {T : Scheme.{u}} {f g : scheme W ⟶ T} (hY : chartY W ≫ f = chartY W ≫ g) (hZ : chartZ W ≫ f = chartZ W ≫ g) : f = g

theorem **map_comp_eq** {A : Type*} [CommRing A] {f : R →+* R'} (h : W.map f = W') (ρ : R' →+* A) : W.map (ρ.comp f) = W'.map ρ

theorem **IsUnimodularSol.comp_of_map_eq** {A : Type*} [CommRing A] {f : R →+* R'} (h : W.map f = W') {ρ : R' →+* A} {P : Fin 3 → A} (hP : IsUnimodularSol (W'.map ρ) P) : IsUnimodularSol (W.map (ρ.comp f)) P

@[simp] theorem **homRingMap_coord** (i : Fin 3) : homRingMap W f h (coord W i) = coord W' i

@[simp] theorem **homRingMap_algebraMap** (r : R) : homRingMap W f h (algebraMap R _ r) = algebraMap R' _ (f r)

theorem **homRingMap_mk** (p : MvPolynomial (Fin 3) R) : homRingMap W f h (Ideal.Quotient.mk _ p) = Ideal.Quotient.mk _ (MvPolynomial.map f p)

theorem **irrelevant_le_map_gradedMap** : irrelevant (grading W') ≤ (irrelevant (grading W)).map (gradedMap W f h)

theorem **ptHom_mapHom** {A : Type u} [CommRing A] (ρ : R' →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W'.map ρ) P) : ptHom W' ρ P hP ≫ mapHom W f h = ptHom W (ρ.comp f) P (hP.comp_of_map_eq h)

theorem **mapHom_preimage_coordOpen** (i : Fin 3) : mapHom W f h ⁻¹ᵁ coordOpen W i = coordOpen W' i

theorem **mapHom_toSpec** : mapHom W f h ≫ toSpec W = toSpec W' ≫ Spec.map (CommRingCat.ofHom f)

theorem **zero_mapHom** : zero W' ≫ mapHom W f h = Spec.map (CommRingCat.ofHom f) ≫ zero W

theorem **mapHom_id** : mapHom W (RingHom.id R) W.map_id = 𝟙 _

theorem **pt_mapHom** {A : Type u} [CommRing A] [Algebra R R'] [Algebra R' A] [Algebra R A] [IsScalarTower R R' A] (h : W.map (algebraMap R R') = W') (P : Fin 3 → A) (hP' : IsUnimodularSol (W'.baseChange A) P) (hP : IsUnimodularSol (W.baseChange A) P) : pt W' P hP' ≫ mapHom W (algebraMap R R') h = pt W P hP

theorem **mapHom_comp** (g : R' →+* R'') (h' : W'.map g = W'') : mapHom W' g h' ≫ mapHom W f h = mapHom W (g.comp f) ((map_comp_eq h g).trans h')

@[simp] theorem **awayRingMap_awayConst** (i : Fin 3) (r : R) : awayRingMap W f h i (awayConst W i r) = awayConst W' i (f r)

@[simp] theorem **awayRingMap_awayCoord** (i j : Fin 3) : awayRingMap W f h i (awayCoord W i j) = awayCoord W' i j

theorem **awayChart_mapHom** (i : Fin 3) : awayChart W' i ≫ mapHom W f h = Spec.map (CommRingCat.ofHom (awayRingMap W f h i)) ≫ awayChart W i

theorem **isPushout_awayRingMap** (i : Fin 3) : IsPushout (CommRingCat.ofHom (awayConst W i)) (CommRingCat.ofHom f) (CommRingCat.ofHom (awayRingMap W f h i)) (CommRingCat.ofHom (awayConst W' i))

theorem **isPullback_awayChart** (i : Fin 3) : IsPullback (Spec.map (CommRingCat.ofHom (awayRingMap W f h i))) (awayChart W' i) (awayChart W i) (mapHom W f h)

theorem **isPullback_mapHom** : IsPullback (mapHom W f h) (toSpec W') (toSpec W) (Spec.map (CommRingCat.ofHom f))

@[simp] theorem **affineMap_affineX** : affineMap W f h (affineX W) = affineX W'

@[simp] theorem **affineMap_affineY** : affineMap W f h (affineY W) = affineY W'

@[simp] theorem **affineMap_algebraMap** (r : R) : affineMap W f h (algebraMap R _ r) = algebraMap R' _ (f r)

@[simp] theorem **originMap_originZ** : originMap W f h (originZ W) = originZ W'

@[simp] theorem **originMap_originW** : originMap W f h (originW W) = originW W'

@[simp] theorem **originMap_algebraMap** (r : R) : originMap W f h (algebraMap R _ r) = algebraMap R' _ (f r)

theorem **isPullback_chartZ** : IsPullback (Spec.map (CommRingCat.ofHom (affineMap W f h))) (chartZ W') (chartZ W) (mapHom W f h)

theorem **isPullback_chartY** : IsPullback (Spec.map (CommRingCat.ofHom (originMap W f h))) (chartY W') (chartY W) (mapHom W f h)

theorem **pullbackIso_hom_mapHom** : (pullbackIso W f h).hom ≫ mapHom W f h = pullback.fst _ _

theorem **pullbackIso_hom_toSpec** : (pullbackIso W f h).hom ≫ toSpec W' = pullback.snd _ _

theorem **pullbackIso_inv_fst** : (pullbackIso W f h).inv ≫ pullback.fst _ _ = mapHom W f h

theorem **pullbackIso_inv_snd** : (pullbackIso W f h).inv ≫ pullback.snd _ _ = toSpec W'

theorem **pt_some_ne_zero** [Nontrivial R] {a b : R} (h : W.toAffine.Equation a b) : pt W (A

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.PiTensorLiftCompSingle

@[simp] theorem **piTensorLift_tprod** (f : ∀ s, X s →ₐ[A] S) (x : ∀ s, X s) : piTensorLift f (tprod A x) = ∏ s, f s (x s)

@[simp] theorem **piTensorLift_comp_single** (f : ∀ s, X s →ₐ[A] S) (s : σ) : (piTensorLift f).comp (singleAlgHom s) = f s

theorem **piTensorLift_single** (f : ∀ s, X s →ₐ[A] S) (s : σ) (x : X s) : piTensorLift f (singleAlgHom s x) = f s x

theorem **piTensor_induction** (P : ∀ (B : Type (max u v w)) [CommRing B] [Algebra A B], Prop) (equiv : ∀ (B C : Type (max u v w)) [CommRing B] [Algebra A B] [CommRing C] [Algebra A C], (B ≃ₐ[A] C) → P B → P C) (base : ∀ (B : Type (max u v w)) [CommRing B] [Algebra A B], (A ≃ₐ[A] B) → P B) (σ : Type w) [Fintype σ] : ∀ (X : σ → Type v) [∀ s, CommRing (X s)] [∀ s, Algebra A (X s)], (∀ s (B : Type (max u v w)) [CommRing B] [Algebra A B], P B → P (B ⊗[A] X s)) → P (⨂[A] s, X s)

Development.AbelianVarietyTheory.Definitions.ShimuraVarieties.PowObjGeometricallyConnected

theorem **isMonHom_of_mul_comp** (f : G ⟶ H) (h : ∀ (T : Over S) (x y : T ⟶ G), (x ∗ y) ≫ f = (x ≫ f) ∗ (y ≫ f)) : IsMonHom f where

theorem **isMonHom_mul** [IsCommMonObj H] (f g : G ⟶ H) [IsMonHom f] [IsMonHom g] : IsMonHom (f ∗ g)

theorem **isMonHom_one** : IsMonHom (1 : G ⟶ H)

theorem **isMonHom_zpow** [IsCommMonObj H] (f : G ⟶ H) [IsMonHom f] (n : ℤ) : IsMonHom (f ^ n)

theorem **act_zero** (a : OAct O G) : a.act 0 = 1

theorem **act_comm** (a : OAct O G) (c d : O) : a.act c ≫ a.act d = a.act d ≫ a.act c

@[simp] theorem **comapAct_act** {O₀ : Type} [CommRing O₀] (a : OAct O G) (φ : O₀ →+* O) (c : O₀) : (comapAct a φ).act c = a.act (φ c)

@[simp] theorem **intAct_act** [IsCommMonObj G] (n : ℤ) : (intAct G).act n = (𝟙 G) ^ n

theorem **IsLinearHom.id** (a : OAct O G) : IsLinearHom a a (𝟙 G)

theorem **IsLinearHom.comp** {a : OAct O G} {b : OAct O H} {c : OAct O K} {f : G ⟶ H} {g : H ⟶ K} (hf : IsLinearHom a b f) (hg : IsLinearHom b c g) : IsLinearHom a c (f ≫ g)

theorem **IsLinearHom.act** (a : OAct O G) (c : O) : IsLinearHom a a (a.act c)

theorem **IsLinearHom.mul** [IsCommMonObj H] {a : OAct O G} {b : OAct O H} {f g : G ⟶ H} (hf : IsLinearHom a b f) (hg : IsLinearHom a b g) : IsLinearHom a b (f * g)

theorem **IsLinearHom.one** (a : OAct O G) (b : OAct O H) : IsLinearHom a b (1 : G ⟶ H)

theorem **IsLinearHom.inv** {a : OAct O G} {b : OAct O H} (e : G ≅ H) (he : IsLinearHom a b e.hom) : IsLinearHom b a e.inv

theorem **IsLinearHom.int** [IsCommMonObj G] [IsCommMonObj H] (f : G ⟶ H) [IsMonHom f] : IsLinearHom (intAct G) (intAct H) f

@[simp] theorem **hom_mk** (x : T ⟶ G) : (mk a x).hom = x

@[simp] theorem **mk_hom** (x : Pts a T) : mk a x.hom = x

@[simp] theorem **mk_symm_apply** (x : Pts a T) : (mk a).symm x = x.hom

@[**ext**] theorem ext {x y : Pts a T} (h : x.hom = y.hom) : x = y

@[simp] theorem **hom_zero** : (0 : Pts a T).hom = 1

@[simp] theorem **hom_add** (x y : Pts a T) : (x + y).hom = x.hom * y.hom

@[simp] theorem **hom_neg** (x : Pts a T) : (-x).hom = x.hom⁻¹

@[simp] theorem **hom_sub** (x y : Pts a T) : (x - y).hom = x.hom / y.hom

@[simp] theorem **hom_smul** (r : O) (x : Pts a T) : (r • x).hom = x.hom ≫ a.act r

theorem **mk_one** : mk a (1 : T ⟶ G) = 0

theorem **mk_mul** (x y : T ⟶ G) : mk a (x * y) = mk a x + mk a y

theorem **mk_comp_act** (x : T ⟶ G) (r : O) : mk a (x ≫ a.act r) = r • mk a x

@[simp] theorem **hom_sum** {I : Type*} (s : Finset I) (x : I → Pts a T) : (∑ i ∈ s, x i).hom = ∏ i ∈ s, (x i).hom

@[simp] theorem **hom_pre** (u : T' ⟶ T) (x : Pts a T) : (pre a u x).hom = u ≫ x.hom

@[simp] theorem **pre_id** : pre a (𝟙 T) = LinearMap.id

theorem **pre_comp** (v : T'' ⟶ T') (u : T' ⟶ T) : pre a (v ≫ u) = pre a v ∘ₗ pre a u

@[simp] theorem **hom_post** {f : G ⟶ H} (hf : IsLinearHom a b f) (x : Pts a T) : (post hf T x).hom = x.hom ≫ f

@[simp] theorem **post_id** : post (IsLinearHom.id a) T = LinearMap.id

theorem **post_comp** {f : G ⟶ H} {g : H ⟶ K} (hf : IsLinearHom a b f) (hg : IsLinearHom b c g) : post (hf.comp hg) T = post hg T ∘ₗ post hf T

theorem **pre_post** {f : G ⟶ H} (hf : IsLinearHom a b f) (u : T' ⟶ T) (x : Pts a T) : pre b u (post hf T x) = post hf T' (pre a u x)

@[simp] theorem **hom_postEquiv** (e : G ≅ H) (he : IsLinearHom a b e.hom) (x : Pts a T) : (postEquiv e he T x).hom = x.hom ≫ e.hom

@[simp] theorem **hom_postEquiv_symm** (e : G ≅ H) (he : IsLinearHom a b e.hom) (x : Pts b T) : ((postEquiv e he T).symm x).hom = x.hom ≫ e.inv

theorem **comp_homOfNat** (x : T ⟶ G) : x ≫ homOfNat φ = (φ T (mk a x)).hom

theorem **isLinearHom_homOfNat** : IsLinearHom a b (homOfNat φ) where

theorem **natural_symm** (u : T' ⟶ T) (y : Pts b T) : (h.equiv T').symm (Pts.pre b u y) = (Pts.pre a u).lTensor M ((h.equiv T).symm y)

@[simp] theorem **hom_equiv_tmul** (m : M) (x : Pts a T) : (h.equiv T (m ⊗ₜ x)).hom = x.hom ≫ h.tmul m

theorem **isLinearHom_tmul** (m : M) : IsLinearHom a b (h.tmul m)

@[simp] theorem **tmul_zero** : h.tmul 0 = 1

theorem **tmul_add** (m n : M) : h.tmul (m + n) = h.tmul m * h.tmul n

theorem **tmul_smul** (c : O) (m : M) : h.tmul (c • m) = h.tmul m ≫ b.act c

theorem **tmul_smul'** (c : O) (m : M) : h.tmul (c • m) = a.act c ≫ h.tmul m

@[simp] theorem **hom_tmulLinear** (m : M) : (h.tmulLinear m).hom = h.tmul m

theorem **tmul_sum** {I : Type*} (s : Finset I) (m : I → M) : h.tmul (∑ i ∈ s, m i) = ∏ i ∈ s, h.tmul (m i)

theorem **induction_on** {P : (T ⟶ H) → Prop} (y : T ⟶ H) (one : P 1) (tmul : ∀ (m : M) (x : T ⟶ G), P (x ≫ h.tmul m)) (mul : ∀ y z, P y → P z → P (y * z)) : P y

theorem **hom_ext** {K : Over S} [GrpObj K] {f g : H ⟶ K} [IsMonHom f] [IsMonHom g] (e : ∀ m : M, h.tmul m ≫ f = h.tmul m ≫ g) : f = g

theorem **ext_tmul** {h₁ h₂ : IsTensor H b G a M} (e : ∀ m, h₁.tmul m = h₂.tmul m) : h₁ = h₂

theorem **liftPts_equiv_tmul** {K : Over S} [GrpObj K] [IsCommMonObj K] {c : OAct O K} (f : M →ₗ[O] Pts c G) (hf : ∀ m, IsLinearHom a c (f m).hom) (m : M) (x : Pts a T) : (h.liftPts f hf T (h.equiv T (m ⊗ₜ x))).hom = x.hom ≫ (f m).hom

theorem **liftPts_natural** {K : Over S} [GrpObj K] [IsCommMonObj K] {c : OAct O K} (f : M →ₗ[O] Pts c G) (hf : ∀ m, IsLinearHom a c (f m).hom) (u : T' ⟶ T) (y : Pts b T) : h.liftPts f hf T' (Pts.pre b u y) = Pts.pre c u (h.liftPts f hf T y)

theorem **isLinearHom_lift** {K : Over S} [GrpObj K] [IsCommMonObj K] {c : OAct O K} (f : M →ₗ[O] Pts c G) (hf : ∀ m, IsLinearHom a c (f m).hom) : IsLinearHom b c (h.lift f hf)

theorem **tmul_lift** {K : Over S} [GrpObj K] [IsCommMonObj K] {c : OAct O K} (f : M →ₗ[O] Pts c G) (hf : ∀ m, IsLinearHom a c (f m).hom) (m : M) : h.tmul m ≫ h.lift f hf = (f m).hom

theorem **isLinearHom_map** (φ : M →ₗ[O] M') {f : G ⟶ G'} (hf : IsLinearHom a a' f) : IsLinearHom b b' (h.map h' φ hf)

theorem **tmul_map** (φ : M →ₗ[O] M') {f : G ⟶ G'} (hf : IsLinearHom a a' f) (m : M) : h.tmul m ≫ h.map h' φ hf = f ≫ h'.tmul (φ m)

theorem **hom_equiv_comp_map** (φ : M →ₗ[O] M') {f : G ⟶ G'} (hf : IsLinearHom a a' f) (z : M ⊗[O] Pts a T) : (h.equiv T z).hom ≫ h.map h' φ hf = (h'.equiv T (TensorProduct.map φ (Pts.post hf T) z)).hom

@[simp] theorem **map_id** : h.map h LinearMap.id (IsLinearHom.id a) = 𝟙 H

theorem **map_comp** (φ : M →ₗ[O] M') (ψ : M' →ₗ[O] M'') {f : G ⟶ G'} (hf : IsLinearHom a a' f) {g : G' ⟶ G''} (hg : IsLinearHom a' a'' g) : h.map h'' (ψ ∘ₗ φ) (hf.comp hg) = h.map h' φ hf ≫ h'.map h'' ψ hg

theorem **map_add_left** (φ ψ : M →ₗ[O] M') {f : G ⟶ G'} (hf : IsLinearHom a a' f) : h.map h' (φ + ψ) hf = h.map h' φ hf ∗ h.map h' ψ hf

theorem **map_mul_right** (φ : M →ₗ[O] M') {f g : G ⟶ G'} (hf : IsLinearHom a a' f) (hg : IsLinearHom a a' g) : h.map h' φ (hf.mul hg) = h.map h' φ hf ∗ h.map h' φ hg

@[simp] theorem **map_zero_left** {f : G ⟶ G'} (hf : IsLinearHom a a' f) : h.map h' 0 hf = 1

theorem **map_smul_left** (c : O) (φ : M →ₗ[O] M') {f : G ⟶ G'} (hf : IsLinearHom a a' f) : h.map h' (c • φ) hf = h.map h' φ hf ≫ b'.act c

theorem **isLinearHom_mapLeft** (φ : M →ₗ[O] M') : IsLinearHom b b' (h.mapLeft h' φ)

theorem **tmul_mapLeft** (φ : M →ₗ[O] M') (m : M) : h.tmul m ≫ h.mapLeft h' φ = h'.tmul (φ m)

theorem **hom_equiv_comp_mapLeft** (φ : M →ₗ[O] M') (z : M ⊗[O] Pts a T) : (h.equiv T z).hom ≫ h.mapLeft h' φ = (h'.equiv T (φ.rTensor _ z)).hom

@[simp] theorem **mapLeft_id** : h.mapLeft h LinearMap.id = 𝟙 H

theorem **mapLeft_comp** (φ : M →ₗ[O] M') (ψ : M' →ₗ[O] M'') : h.mapLeft h'' (ψ ∘ₗ φ) = h.mapLeft h' φ ≫ h'.mapLeft h'' ψ

theorem **mapLeft_add** (φ ψ : M →ₗ[O] M') : h.mapLeft h' (φ + ψ) = h.mapLeft h' φ ∗ h.mapLeft h' ψ

@[simp] theorem **mapLeft_zero** : h.mapLeft h' 0 = 1

theorem **mapLeft_smul** (c : O) (φ : M →ₗ[O] M') : h.mapLeft h' (c • φ) = h.mapLeft h' φ ≫ b'.act c

theorem **mapLeft_smul_id** (c : O) : h.mapLeft h (c • LinearMap.id) = b.act c

@[simp] theorem **mapLeftIso_hom** (e : M ≃ₗ[O] M') : (h.mapLeftIso h' e).hom = h.mapLeft h' e

@[simp] theorem **mapLeftIso_inv** (e : M ≃ₗ[O] M') : (h.mapLeftIso h' e).inv = h'.mapLeft h e.symm

@[simp] theorem **endAct_act** (φ : Module.End O M) : h.endAct.act φ = h.mapLeft h φ

theorem **isLinearHom_mapRight** {f : G ⟶ G'} (hf : IsLinearHom a a' f) : IsLinearHom b b' (h.mapRight h' hf)

theorem **tmul_mapRight** {f : G ⟶ G'} (hf : IsLinearHom a a' f) (m : M) : h.tmul m ≫ h.mapRight h' hf = f ≫ h'.tmul m

@[simp] theorem **mapRight_id** : h.mapRight h (IsLinearHom.id a) = 𝟙 H

theorem **mapRight_comp** {f : G ⟶ G'} (hf : IsLinearHom a a' f) {g : G' ⟶ G''} (hg : IsLinearHom a' a'' g) : h.mapRight h'' (hf.comp hg) = h.mapRight h' hf ≫ h'.mapRight h'' hg

theorem **mapRight_mul** {f g : G ⟶ G'} (hf : IsLinearHom a a' f) (hg : IsLinearHom a a' g) : h.mapRight h' (hf.mul hg) = h.mapRight h' hf ∗ h.mapRight h' hg

theorem **mapRight_act** (c : O) : h.mapRight h (IsLinearHom.act a c) = b.act c

theorem **isLinearHom_uniqueIso** : IsLinearHom b b' (h.uniqueIso h').hom

theorem **tmul_uniqueIso_hom** (m : M) : h.tmul m ≫ (h.uniqueIso h').hom = h'.tmul m

theorem **hom_equiv_comp_uniqueIso** (z : M ⊗[O] Pts a T) : (h.equiv T z).hom ≫ (h.uniqueIso h').hom = (h'.equiv T z).hom

theorem **eq_uniqueIso** (f : H ⟶ H') (hf : ∀ (T : Over S) (z : M ⊗[O] Pts a T), (h.equiv T z).hom ≫ f = (h'.equiv T z).hom) : f = (h.uniqueIso h').hom

theorem **existsUnique_iso** : ∃! e : H ≅ H', ∀ (T : Over S) (z : M ⊗[O] Pts a T), (h.equiv T z).hom ≫ e.hom = (h'.equiv T z).hom

@[simp] theorem **congrLeft_tmul** (h : IsTensor H b G a M) (e : M' ≃ₗ[O] M) (m : M') : (h.congrLeft e).tmul m = h.tmul (e m)

@[simp] theorem **congrRight_tmul** (h : IsTensor H b G a M) (e : G' ≅ G) (he : IsLinearHom a' a e.hom) (m : M) : (h.congrRight e he).tmul m = e.hom ≫ h.tmul m

@[simp] theorem **ofIso_tmul** (h : IsTensor H b G a M) (e : H ≅ H') (he : IsLinearHom b b' e.hom) (m : M) : (h.ofIso e he).tmul m = h.tmul m ≫ e.hom

@[simp] theorem **self_tmul** (a : OAct O G) (c : O) : (self a).tmul c = a.act c

theorem **tmul_algAct** (c : O') (m : M) : h.tmul m ≫ (h.algAct O').act c = h.tmul (c • m)

@[simp] theorem **algAct_algebraMap** (c : O) : (h.algAct O').act (algebraMap O O' c) = b.act c

theorem **isLinearHom_algAct** (c : O') : IsLinearHom b b ((h.algAct O').act c)

theorem **comapAct_algAct** : comapAct (h.algAct O') (algebraMap O O') = b

@[simp] theorem **hom_algEquiv** (z : M ⊗[O] Pts a T) : (h.algEquiv O' T z).hom = (h.equiv T z).hom

theorem **isLinearHom_algAct_map** [Module O' M'] [IsScalarTower O O' M'] (h' : IsTensor H' b' G' a' M') (φ : M →ₗ[O'] M') {f : G ⟶ G'} (hf : IsLinearHom a a' f) : IsLinearHom (h.algAct O') (h'.algAct O') (h.map h' (φ.restrictScalars O) hf)

@[simp] theorem **trans_tmul** {K : Over S} [GrpObj K] [IsCommMonObj K] {c : OAct O' K} {N : Type} [AddCommGroup N] [Module O' N] [Module O N] [IsScalarTower O O' N] (k : IsTensor K c H (h.algAct O') N) (n : N) (m : M) : (h.trans k).tmul (n ⊗ₜ m) = h.tmul m ≫ k.tmul n

@[simp] theorem **prodAct_act** (b : OAct O H) (b' : OAct O H') (c : O) : (prodAct b b').act c = b.act c ⊗ₘ b'.act c

theorem **isLinearHom_fst** (b : OAct O H) (b' : OAct O H') : IsLinearHom (prodAct b b') b (fst H H')

theorem **isLinearHom_snd** (b : OAct O H) (b' : OAct O H') : IsLinearHom (prodAct b b') b' (snd H H')

theorem **pow_hom_ext** {n : ℕ} (p : IsPowGrp G' G n) {T : Over S} {x y : T ⟶ G'} (e : ∀ i, x ≫ p.pr i = y ≫ p.pr i) : x = y

theorem **powLift_pr** {n : ℕ} (p : IsPowGrp G' G n) {T : Over S} (x : Fin n → (T ⟶ G)) (i : Fin n) : powLift p x ≫ p.pr i = x i

theorem **powAct_act_pr** {n : ℕ} (p : IsPowGrp G' G n) (a : OAct O G) (c : O) (i : Fin n) : (powAct p a).act c ≫ p.pr i = p.pr i ≫ a.act c

theorem **isLinearHom_pr** {n : ℕ} (p : IsPowGrp G' G n) (a : OAct O G) (i : Fin n) : IsLinearHom (powAct p a) a (p.pr i)

@[simp] theorem **Pts.hom_prodEquiv_symm** (b : OAct O H) (b' : OAct O H') {T : Over S} (y : Pts b T × Pts b' T) : ((Pts.prodEquiv b b' T).symm y).hom = lift y.1.hom y.2.hom

@[simp] theorem **Pts.hom_powEquiv_symm** {n : ℕ} (p : IsPowGrp G' G n) (a : OAct O G) {T : Over S} (y : Fin n → Pts a T) : ((Pts.powEquiv p a T).symm y).hom = powLift p fun i ↦ (y i).hom

@[simp] theorem **prod_tmul** (h : IsTensor H b G a M) (h' : IsTensor H' b' G a M') (m : M) (m' : M') : (h.prod h').tmul (m, m') = CartesianMonoidalCategory.lift (h.tmul m) (h'.tmul m')

theorem **pow_tmul_pr** {n : ℕ} (p : IsPowGrp G' G n) (a : OAct O G) (v : Fin n → O) (i : Fin n) : (pow p a).tmul v ≫ p.pr i = a.act (v i)

theorem **pow_mapLeft_pr** {n k : ℕ} (p : IsPowGrp G' G n) (q : IsPowGrp H G k) (a : OAct O G) (A : Matrix (Fin k) (Fin n) O) (i : Fin k) : (pow p a).mapLeft (pow q a) (Matrix.toLin' A) ≫ q.pr i = ∏ j, p.pr j ≫ a.act (A i j)

theorem **isLinearPullback_pullback** [IsCommMonObj G] (a : OAct O G) (f : S' ⟶ S) : IsLinearPullback (a.pullback f G) a f (pullback.fst G.hom f)

theorem **IsLinearPullback.of_iso** {a₁ : OAct O G₁} {b₁ : OAct O H₁} {a : OAct O G} {f : S' ⟶ S} {φ : G₁.left ⟶ G.left} (hφ : IsLinearPullback a₁ a f φ) (e : H₁ ≅ G₁) (he : IsLinearHom b₁ a₁ e.hom) : IsLinearPullback b₁ a f (e.hom.left ≫ φ) where

@[simp] theorem **overBase_left** (f : S' ⟶ S) {T T' : Over S'} (u : T' ⟶ T) : (overBase f u).left = u.left

@[simp] theorem **Pts.hom_pullbackEquiv_left** (hφ : IsLinearPullback a₁ a f φ) {T : Over S'} (x : Pts a₁ T) : (Pts.pullbackEquiv hφ T x).hom.left = x.hom.left ≫ φ

theorem **Pts.pullbackEquiv_pre** (hφ : IsLinearPullback a₁ a f φ) {T T' : Over S'} (u : T' ⟶ T) (x : Pts a₁ T) : Pts.pullbackEquiv hφ T' (Pts.pre a₁ u x) = Pts.pre a (overBase f u) (Pts.pullbackEquiv hφ T x)

theorem **baseChange_tmul** (h : IsTensor H b G a M) (hφ : IsLinearPullback a₁ a f φ) (hψ : IsLinearPullback b₁ b f ψ) (m : M) : ((h.baseChange hφ hψ).tmul m).left ≫ ψ = φ ≫ (h.tmul m).left

@[simp] theorem **pullback_tmul** (h : IsTensor H b G a M) (f : S' ⟶ S) (m : M) : (h.pullback f).tmul m = (Over.pullback f).map (h.tmul m)

theorem **pullback_mapLeft** {H' : Over S} [GrpObj H'] [IsCommMonObj H'] {b' : OAct O H'} {M' : Type} [AddCommGroup M'] [Module O M'] (h : IsTensor H b G a M) (h' : IsTensor H' b' G a M') (f : S' ⟶ S) (l : M →ₗ[O] M') : (h.pullback f).mapLeft (h'.pullback f) l = (Over.pullback f).map (h.mapLeft h' l)

theorem **exists_isLinearPullback** (h : IsTensor H b G a M) (h₁ : IsTensor H₁ b₁ G₁ a₁ M) (hφ : IsLinearPullback a₁ a f φ) : ∃ ψ : H₁.left ⟶ H.left, IsLinearPullback b₁ b f ψ ∧ ∀ m : M, (h₁.tmul m).left ≫ ψ = φ ≫ (h.tmul m).left

theorem **fixIncl_comp** {K : Over S} (p : K ⟶ K) : fixIncl p ≫ p = fixIncl p

theorem **fixProj_incl** {K : Over S} {p : K ⟶ K} (hp : p ≫ p = p) : fixProj hp ≫ fixIncl p = p

theorem **fixIncl_proj** {K : Over S} {p : K ⟶ K} (hp : p ≫ p = p) : fixIncl p ≫ fixProj hp = 𝟙 _

theorem **isMonHom_fixIncl** {p : K ⟶ K} [IsMonHom p] (hp : p ≫ p = p) : letI

theorem **isMonHom_fixProj** {p : K ⟶ K} [IsMonHom p] (hp : p ≫ p = p) : letI

theorem **isCommMonObj_fix** [IsCommMonObj K] {p : K ⟶ K} [IsMonHom p] (hp : p ≫ p = p) : letI

theorem **isLinearHom_retract_incl** (c : OAct O K) (i : H ⟶ K) (r : K ⟶ H) [IsMonHom i] [IsMonHom r] (hir : i ≫ r = 𝟙 H) (hc : ∀ x, c.act x ≫ r ≫ i = r ≫ i ≫ c.act x) : IsLinearHom (retractAct c i r hir hc) c i

theorem **isLinearHom_retract_proj** (c : OAct O K) (i : H ⟶ K) (r : K ⟶ H) [IsMonHom i] [IsMonHom r] (hir : i ≫ r = 𝟙 H) (hc : ∀ x, c.act x ≫ r ≫ i = r ≫ i ≫ c.act x) : IsLinearHom c (retractAct c i r hir hc) r

@[simp] theorem **IsTensor.ofRetract_tmul** {G : Over S} [GrpObj G] [IsCommMonObj G] [IsCommMonObj K] [IsCommMonObj H] {a : OAct O G} {b : OAct O H} {c : OAct O K} {M N : Type} [AddCommGroup M] [Module O M] [AddCommGroup N] [Module O N] (k : IsTensor K c G a N) (s : M →ₗ[O] N) (t : N →ₗ[O] M) (hst : t ∘ₗ s = LinearMap.id) {i : H ⟶ K} {r : K ⟶ H} (hi : IsLinearHom b c i) (hr : IsLinearHom c b r) (hir : i ≫ r = 𝟙 H) (hri : r ≫ i = k.mapLeft k (s ∘ₗ t)) (m : M) : (k.ofRetract s t hst hi hr hir hri).tmul m = k.tmul (s m) ≫ r

theorem **Summand.nonempty** [Module.Finite O M] [Module.Projective O M] : Nonempty (Summand O M)

theorem **toLin'_matrix** : Matrix.toLin' P.matrix = P.incl ∘ₗ P.proj

theorem **matrix_mul_self** : P.matrix ∗ P.matrix = P.matrix

theorem **idem_pr** (i : Fin P.n) : P.idem a ≫ powPr G i = ∏ j, powPr G j ≫ a.act (P.matrix i j)

theorem **idem_idem** : P.idem a ≫ P.idem a = P.idem a

theorem **objIncl_objProj** : P.objIncl a ≫ P.objProj a = 𝟙 _

theorem **objProj_objIncl** : P.objProj a ≫ P.objIncl a = P.idem a

theorem **act_idem** (x : O) : (powAct (powIsPowGrp G P.n) a).act x ≫ P.objProj a ≫ P.objIncl a = P.objProj a ≫ P.objIncl a ≫ (powAct (powIsPowGrp G P.n) a).act x

theorem **powObj_property** (P : MorphismProperty Scheme.{0}) [P.IsStableUnderComposition] [P.IsStableUnderBaseChange] (h₀ : P (𝟙 S)) (G : Over S) (hG : P G.hom) : ∀ n : ℕ, P (powObj G n).hom | 0 => h₀ | n + 1 => P.comp_mem _ _ (P.pullback_fst G.hom (powObj G n).hom (powObj_property P h₀ G hG n)) hG theorem powObj_geometricallyConnected (G : Over S) [UniversallyOpen G.hom] [GeometricallyConnected G.hom] : ∀ n : ℕ, GeometricallyConnected (powObj G n).hom | 0 => inferInstanceAs (GeometricallyConnected (𝟙 S)) | n + 1 => by have

theorem **tmul_mapLeft_self** (t : M →ₗ[O] O) {m : M} (hm : t m = 1) : h.tmul m ≫ h.mapLeft (self a) t = 𝟙 G

theorem **exists_retract** : ∃ (n : ℕ) (ρ : Retract H (powObj G n)), IsMonHom ρ.i ∧ IsMonHom ρ.r

theorem **property** (P : MorphismProperty Scheme.{0}) [P.IsStableUnderComposition] [P.IsStableUnderBaseChange] (h₀ : P (𝟙 S)) (hP : ∀ {X Y : Scheme.{0}} {f : X ⟶ S} {g : Y ⟶ S} {i : X ⟶ Y} {r : Y ⟶ X}, i ≫ r = 𝟙 X → r ≫ f = g → P g → P f) (hG : P G.hom) : P H.hom

theorem **smooth** [hG : Smooth G.hom] : Smooth H.hom

theorem **etale** [hG : Etale G.hom] : Etale H.hom

theorem **flat** [hG : Flat G.hom] : Flat H.hom

theorem **locallyOfFiniteType** [hG : LocallyOfFiniteType G.hom] : LocallyOfFiniteType H.hom

theorem **locallyOfFinitePresentation** [hG : LocallyOfFinitePresentation G.hom] : LocallyOfFinitePresentation H.hom

theorem **quasiCompact** [hG : QuasiCompact G.hom] : QuasiCompact H.hom

theorem **isSeparated** [hG : IsSeparated G.hom] : IsSeparated H.hom

theorem **isProper** [hG : IsProper G.hom] : IsProper H.hom

theorem **isAffineHom** [hG : IsAffineHom G.hom] : IsAffineHom H.hom

theorem **isFinite** [hG : IsFinite G.hom] : IsFinite H.hom

theorem **geometricallyConnected** [UniversallyOpen G.hom] [GeometricallyConnected G.hom] : GeometricallyConnected H.hom

theorem **isAbelianScheme** (hG : IsAbelianScheme G) : IsAbelianScheme H

theorem **of_isSemiAbelian** (hG : IsSemiAbelian G) : Smooth H.hom ∧ IsSeparated H.hom ∧ QuasiCompact H.hom ∧ GeometricallyConnected H.hom

theorem **abLocus_subset** : abLocus G ⊆ abLocus H

theorem **gmRat_id_ne_one** : (𝟙 gmRat : gmRat ⟶ gmRat) ≠ 1

Development.AbelianVarietyTheory.Definitions.SpanningFfamilies.CoeffCongFactor

theorem **exists_gen_one** {𝔪 : Ideal (𝒪 E)} (h0 : 𝔪 ≠ ⊥) (A : Submodule (𝒪 E) E) (hA : A.FG) : ∃ p ∈ A, A ≤ Submodule.span (𝒪 E) {p} ⊔ 𝔪 • A

theorem **exists_gens** {𝔪 : Ideal (𝒪 E)} (h0 : 𝔪 ≠ ⊥) : ∀ (r : ℕ) (L : Submodule (𝒪 E) (Fin r → E)), L.FG → ∃ β : Fin r → Fin r → E, (∀ i, β i ∈ L) ∧ L ≤ Submodule.span (𝒪 E) (Set.range β) ⊔ 𝔪 • L

theorem **span_of_map** {r : ℕ} {M : Type} [AddCommGroup M] [Module (𝒪 E) M] [Module.Finite (𝒪 E) M] [Module.Projective (𝒪 E) M] (hM : Module.finrank (𝒪 E) M = r) {𝔞 : Ideal (𝒪 E)} {L : Submodule (𝒪 E) (Fin r → E)} (hL : L.FG) (hsp : Submodule.span E (L : Set (Fin r → E)) = ⊤) {δ : M →ₗ[𝒪 E] (Fin r → E)} (hδ : ∀ a, δ a ∈ L) (hon : L ≤ LinearMap.range δ ⊔ 𝔞 • L) {S : Submodule (𝒪 E) M} (h : L ≤ loc 𝔞 (S.map δ ⊔ 𝔞 • L)) : S ⊔ 𝔞 • ⊤ = ⊤

theorem **cong_mono** {𝔞 𝔟 : Ideal (𝒪 E)} (h : 𝔞 ≤ 𝔟) {u v : Fin r → E} (H : Cong X 𝔞 u v) : Cong X 𝔟 u v

theorem **coeff_cong_factor** (D : SW.Level X) {𝔞 𝔟 : Ideal (𝒪 E)} (hm : D.𝔪 = 𝔞 ∗ 𝔟) (hc : IsCoprime 𝔞 𝔟) {c d : Fin r → 𝒪 E} (h : Cong X 𝔟 (∑ j, c j • D.β j) (∑ j, d j • D.β j)) (j) : c j - d j ∈ 𝔟

theorem **exists_cong** {𝔞 : Ideal (𝒪 E)} {N : Submodule (𝒪 E) (Fin r → E)} {v : Fin r → E} (hv : v ∈ loc 𝔞 X.Λ) (h : v ∈ loc 𝔞 (N ⊔ loc 𝔞 (𝔞 • X.Λ))) : ∃ u ∈ N, Cong X 𝔞 v u

theorem **red_iff** (D : SW.Level X) {𝔞 𝔟 : Ideal (𝒪 E)} (hm : D.𝔪 = 𝔞 ∗ 𝔟) (hc : IsCoprime 𝔞 𝔟) {L : Lt E r} (hL : L ∈ D.lats) {y z : Fin r → Fin r → E} (hz : ∀ i, z i ∈ L) (h1 : ∀ i, Cong X 𝔟 (y i) (z i)) (h2 : ∀ i, Cong X 𝔞 (y i) (D.β i)) : (∃ k : GL (Fin r) (𝒪 E ⧸ D.𝔪), (∀ i j, ∃ a ∈ 𝔞, (k : Matrix (Fin r) (Fin r) (𝒪 E ⧸ D.𝔪)) i j - (1 : Matrix (Fin r) (Fin r) (𝒪 E ⧸ D.𝔪)) i j = Ideal.Quotient.mk D.𝔪 a) ∧ D.Red y k) ↔ L ≤ loc 𝔟 (Submodule.span (𝒪 E) (Set.range z) ⊔ 𝔟 • L)

theorem **hbar_eq** (a : 𝒪 E) : H.bar a = bar a

theorem **conjI_mfr** (l : Geo.Tor.Level H) : conjI (Mfr l) = l.Mb

theorem **m_mem** (l : Geo.Tor.Level H) : (l.M : 𝒪 E) ∈ Mfr l

theorem **mfr_ne_top** (l : Geo.Tor.Level H) : Mfr l ≠ ⊤

theorem **mem_congGrp** {𝔪 𝔞 : Ideal (𝒪 E)} (h : 𝔪 ≤ 𝔞) (k : GL (Fin r) (𝒪 E ⧸ 𝔪)) : k ∈ congGrp r h ↔ ∀ i j, ∃ a ∈ 𝔞, (k : Matrix (Fin r) (Fin r) (𝒪 E ⧸ 𝔪)) i j - (1 : Matrix (Fin r) (Fin r) (𝒪 E ⧸ 𝔪)) i j = Ideal.Quotient.mk 𝔪 a

theorem **image_upper** {α : Type} (hn : r ≤ n) (x : Fin n → α) : x '' {b | b.val < r} = Set.range fun i => x (Fin.castLE hn i)

theorem **exists_levTie** (d : DeltaDat Xo c n) (σ : c.F →+* ℂ) (hI : d.l₂.Mb = l.Mb ∗ d.I) (hc : IsCoprime (Mfr l) (Jd d)) (X : Data E (Dw.p + Dw.q)) (hn : Dw.p + Dw.q ≤ n) : ∃ D : SW.Level X, LevTie d σ (spanInd d) D hn

theorem **exists_x0Tie** (d : DeltaDat Xo c n) (hpd : Nat.Coprime d.pd l.M) {X : Data E (Dw.p + Dw.q)} (D : SW.Level X) (hm : D.𝔪 ≤ Mfr l) (hn : Dw.p + Dw.q ≤ n) {L : Lt E (Dw.p + Dw.q)} (hL : L ∈ D.lats) {δ : Dw.La →ₗ[𝒪 E] (Fin (Dw.p + Dw.q) → E)} (hδ : ∀ a, δ a ∈ L) (hon : L ≤ LinearMap.range δ ⊔ Mfr l • L) (rk : Bool) : ∃ x0, IsBaseClassLevelAdapted d D hn x0 δ ∧ AdmX (l

theorem **LevTie.norm_eq_one** {d : DeltaDat Xo c n} {σ : c.F →+* ℂ} {ph : (Fin n → LaJ Xo c d) → c.F} {D : SW.Level X} {hn : r ≤ n} (T : LevTie d σ ph D hn) {L : Lt E r} (hL : L ∈ D.lats) {δ : Dw.La →ₗ[𝒪 E] (Fin r → E)} (hδ : ∀ a, δ a ∈ L) (hon : L ≤ LinearMap.range δ ⊔ Jd d • L) (xs : Fin n → Dw.La) (hx : ∀ i, Cong X (Jd d) (D.β i) (δ (xs (Fin.castLE hn i)))) : ‖σ (ph fun b => Submodule.Quotient.mk (xs b))‖ = 1

theorem **spanInd_laws** (d : DeltaDat Xo c n) : (∀ x x' : Fin n → LaJ Xo c d, (∀ b : Fin n, b.val < Dw.p + Dw.q → x b = x' b) → spanInd d x = spanInd d x') ∧ (∀ x, spanInd d x ≠ 0 → Submodule.span (𝒪 E) (x '' {b | b.val < Dw.p + Dw.q}) = ⊤) ∧ ∀ (k : LaJ Xo c d ≃ₗ[𝒪 E] LaJ Xo c d) x, spanInd d (fun b => k (x b)) = spanInd d x

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.FirstCoordinateAdmissible

theorem **first_coordinate_admissible** : admitted 1 1 (0 : Fam E n N)

theorem **isType_data** {r : ℕ} (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) : IsType X.Ψ

theorem **map_cc** (τ : E →+* ℂ) (t : E) : τ (cc t) = conj (τ t)

theorem **ccT_map** {α β : Type} (τ : E →+* ℂ) (A : Matrix α β E) : (ccT A).map τ = (A.map τ)ᴴ

theorem **isIntegral_cc** {x : E} (hx : IsIntegral ℤ x) : IsIntegral ℤ (cc x)

theorem **ccT_one** {α : Type} [DecidableEq α] : ccT (1 : Matrix α α E) = 1

theorem **ccT_ccT** {α β : Type} (A : Matrix α β E) : ccT (ccT A) = A

theorem **ccT_mul** {α β γ : Type} [Fintype β] (A : Matrix α β E) (B : Matrix β γ E) : ccT (A ∗ B) = ccT B ∗ ccT A

theorem **one_mem_DH** (M : ℕ) : (1 : Matrix (Fin n) (Fin n) E) ∈ DH E n M

theorem **ccT_mem_DH** {M : ℕ} {h : Matrix (Fin n) (Fin n) E} (hh : h ∈ DH E n M) : ccT h ∈ DH E n M

theorem **mul_mem_DH** {M : ℕ} {h g : Matrix (Fin n) (Fin n) E} (hh : h ∈ DH E n M) (hg : g ∈ DH E n M) : h ∗ g ∈ DH E n M

theorem **moves_one** (Λ : Lt E n) : Moves 1 Λ Λ

theorem **moves_mul** {h g : Matrix (Fin n) (Fin n) E} {Λ Λ' Λ'' : Lt E n} (h1 : Moves g Λ Λ') (h2 : Moves h Λ' Λ'') : Moves (h ∗ g) Λ Λ''

theorem **moves_ccT** {h : Matrix (Fin n) (Fin n) E} (hu : ccT h ∗ h = 1) {Λ Λ' : Lt E n} (h1 : Moves h Λ Λ') : Moves (ccT h) Λ' Λ

theorem **cls_eq_iff** {M : ℕ} (Λ Λ' : Vset E n M) : (Quot.mk _ Λ : Cls E n M) = Quot.mk _ Λ' ↔ ∃ h ∈ DH E n M, Moves h Λ.1 Λ'.1

theorem **wt_moves** {M : ℕ} {h : Matrix (Fin n) (Fin n) E} (hh : h ∈ DH E n M) {Λ Λ' : Lt E n} (m : Moves h Λ Λ') : wt M Λ' = wt M Λ

theorem **classSum_eq** {M : ℕ} {f : Lt E n → ℂ} (hf : IsClassFn M f) (s : Cls E n M → Vset E n M) (hs : ∀ c, Quot.mk _ (s c) = c) : classSum M f = ∑ᶠ c : Cls E n M, (wt M (s c).1 : ℂ) ∗ f (s c).1

theorem **classSum_congr** {M : ℕ} {f g : Lt E n → ℂ} (h : ∀ Λ ∈ Vset E n M, f Λ = g Λ) : classSum M f = classSum M g

theorem **expo_eq_trace** (x : Matrix (Fin n) (Fin N) E) (Z : CM Ψ N N) (z z' : CM Ψ n N) : expo Ψ x Z z z' = ∑ τ : Ψ, ((x.map τ.1 ∗ Z τ ∗ (x.map τ.1)ᴴ).trace + (x.map τ.1 ∗ (z τ)ᵀ).trace + (z' τ ∗ (x.map τ.1)ᴴ).trace)

theorem **expo_mul** {h : Matrix (Fin n) (Fin n) E} (hu : ccT h ∗ h = 1) (x : Matrix (Fin n) (Fin N) E) (Z : CM Ψ N N) (z z' : CM Ψ n N) : expo Ψ (h ∗ x) Z z z' = expo Ψ x Z (fun τ => (h.map τ.1)ᵀ ∗ z τ) (fun τ => (h.map τ.1)ᴴ ∗ z' τ)

theorem **swd_zero** (M : ℕ) (φ : Fam E n N) (Z : CM Ψ N N) : swd Ψ M φ Z 0 0 = swDiff Ψ M φ Z

theorem **mem_stdL** {m : ℕ} (x : Fin m → E) : x ∈ stdL E m ↔ ∀ a, IsIntegral ℤ (x a)

theorem **herm_one** {m : ℕ} (x y : Fin m → E) : herm 1 x y = ∑ a, x a ∗ cc (y a)

theorem **stdL_mem_Vset** (n M : ℕ) : stdL E n ∈ Vset E n M

theorem **nat_smul_mem** {m : ℕ} (Λ : Lt E m) (s : ℕ) {v : Fin m → E} (hv : v ∈ Λ) : (s : E) • v ∈ Λ

theorem **mem_of_coprime** {m : ℕ} (Λ : Lt E m) {s R : ℕ} (h : s.Coprime R) {v : Fin m → E} (h1 : (s : E) • v ∈ Λ) (h2 : (R : E) • v ∈ Λ) : v ∈ Λ

theorem **I_mem_Hn** : (fun _ => I • (1 : Matrix (Fin N) (Fin N) ℂ) : CM Ψ N N) ∈ Hn Ψ N

theorem **ccT_J** : ccT (Matrix.J (Fin N) E) = -Matrix.J (Fin N) E

theorem **J_mem_UU** : Matrix.J (Fin N) E ∈ UU E N

theorem **one_mem_UU** : (1 : M2 E N) ∈ UU E N

theorem **transH_one** (φ : Matrix (Fin n) (Fin N) E → ℂ) : TransH Ψ (1 : M2 E N) φ φ

theorem **gidx_stdL** : gidx (stdL E N) = 1

theorem **thH_deg_zero** (φ : Matrix (Fin n) (Fin 0) E → ℂ) (Z : CM Ψ 0 0) (z z' : CM Ψ n 0) : thH Ψ φ Z z z' = φ 0

theorem **weil_deg_zero** (γ : M2 E 0) (φ : Matrix (Fin n) (Fin 0) E → ℂ) : weil Ψ γ φ 0 = φ 0

theorem **eis_deg_zero** (φ : Matrix (Fin n) (Fin 0) E → ℂ) (Z : CM Ψ 0 0) (z z' : CM Ψ n 0) : eis Ψ φ Z z z' = φ 0

theorem **swDiff_deg_zero** (M : ℕ) (φ : Fam E n 0) (Z : CM Ψ 0 0) : swDiff Ψ M φ Z = 0

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.Fischer

theorem **zR_embR** (x : Frame X.Ψ r) (y : Matrix (Fin n) (Fin r) E) (k : Fin n) : zR X idx x (embR X y) k = GlobalTheory.Dict.pv X idx x (y k)

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.IMemSiegel

theorem **isSB_zero** : IsSB (0 : Matrix κ ι K → ℂ)

theorem **I_mem_siegel** : (I • (1 : Matrix ι ι ℂ)) ∈ Siegel ι

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.IntMIffInt

theorem **isMu_one** : IsMu 1

theorem **ph_zero** : Ph 0

theorem **ph_I** : Ph I

theorem **IsMu.mul** {a b : ℂ} (ha : IsMu a) (hb : IsMu b) : IsMu (a * b)

theorem **IsMu.inv** {a : ℂ} (ha : IsMu a) : IsMu a⁻¹

theorem **IsMu.norm_eq_one** {a : ℂ} (ha : IsMu a) : ‖a‖ = 1

theorem **IsMu.ph** {a : ℂ} (ha : IsMu a) : Ph a

theorem **Ph.mul** {a b : ℂ} (ha : Ph a) (hb : Ph b) : Ph (a * b)

theorem **Ph.inv** {a : ℂ} (ha : Ph a) : Ph a⁻¹

theorem **pair_int** {a b : Matrix κ ι ℚ} (ha : ∀ k i, ∃ z : ℤ, a k i = z) (hb : ∀ k i, ∃ z : ℤ, b k i = z) : ∃ z : ℤ, (∑ k, a k ⬝ᵥ b k : ℚ) = z

theorem **intM_iff_int** {x : Matrix κ ι ℚ} : IntM x ↔ ∀ k i, ∃ z : ℤ, x k i = z

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.IsBaseClassLevelAdapted

theorem **nonempty_uw** {σ : c.F →+* ℂ} {U : Unf Xo c σ r} (h : IsUnf U) : Nonempty (UW U)

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.IsWeakOfDatum

theorem **isWeak_of_datum** {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (h : IsWeilDatumO ψ S T χS χT om) : IsWeak ψ S T om

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.NotAdmissibleHalfLatticeFamily

theorem **cc_cc** (x : E) : cc (cc x) = x

theorem **cc_integral** {x : E} (hx : IsIntegral ℤ x) : IsIntegral ℤ (cc x)

theorem **mem_stdL** {m : ℕ} (x : Fin m → E) : x ∈ stdL E m ↔ ∀ a, IsIntegral ℤ (x a)

theorem **herm_one** {m : ℕ} (x y : Fin m → E) : herm 1 x y = ∑ a, x a * cc (y a)

theorem **stdL_mem_Vset** (n M : ℕ) : stdL E n ∈ Vset E n M

theorem **one_mem_DH** (n M : ℕ) : (1 : Matrix (Fin n) (Fin n) E) ∈ DH E n M

theorem **nat_smul_mem** {m : ℕ} (Λ : Lt E m) (s : ℕ) {v : Fin m → E} (hv : v ∈ Λ) : (s : E) • v ∈ Λ

theorem **mem_of_coprime** {m : ℕ} (Λ : Lt E m) {s R : ℕ} (h : s.Coprime R) {v : Fin m → E} (h1 : (s : E) • v ∈ Λ) (h2 : (R : E) • v ∈ Λ) : v ∈ Λ

theorem **admissible_indicator_functions** {n N M R : ℕ} (hM : 0 < M) (hR : 0 < R) : admitted M R (indFam E n N)

theorem **indFam_ne_zero** (n N M : ℕ) : ∃ Λ ∈ Vset E n M, indFam E n N Λ 0 = 1

theorem **I_mem_Hn** (Ψ : Finset (E →+* ℂ)) (N : ℕ) : (fun _ => I • (1 : Matrix (Fin N) (Fin N) ℂ) : CM Ψ N N) ∈ Hn Ψ N

theorem **ccT_J** (N : ℕ) : ccT (Matrix.J (Fin N) E) = -Matrix.J (Fin N) E

theorem **J_mem_UU** (N : ℕ) : Matrix.J (Fin N) E ∈ UU E N

theorem **one_mem_UU** (N : ℕ) : (1 : M2 E N) ∈ UU E N

theorem **transH_one** (Ψ : Finset (E →+* ℂ)) {n N : ℕ} (φ : Matrix (Fin n) (Fin N) E → ℂ) : TransH Ψ (1 : M2 E N) φ φ

theorem **transH_translation** (Ψ : Finset (E →+* ℂ)) {n N : ℕ} (β : Matrix (Fin N) (Fin N) E) (ψ : Matrix (Fin n) (Fin N) E → ℂ) : TransH Ψ (Matrix.fromBlocks 1 β 0 1) (fun x => ψ x * chiH Ψ β x) ψ

theorem **expo_ofQ** (Ψ : Finset (E →+* ℂ)) {n N : ℕ} (m : Matrix (Fin n) (QI E N) ℚ) (Z : CM Ψ N N) (z z' : CM Ψ n N) : ∑ k, (qv (m k) ⬝ᵥ tauOf Ψ Z *ᵥ qv (m k) + 2 * (qv (m k) ⬝ᵥ zetaOf Ψ z z' k)) = 2 * expo Ψ (ofQ m) Z z z'

theorem **transH_levi** (Ψ : Finset (E →+* ℂ)) {n N : ℕ} {A D : Matrix (Fin N) (Fin N) E} (h : ccT D * A = 1) (φ : Matrix (Fin n) (Fin N) E → ℂ) : TransH Ψ (Matrix.fromBlocks A 0 0 D) φ (fun x => (∏ τ : Ψ, τ.1 D.det ^ n) * φ (x * A))

theorem **lag_nontrivial** (N : ℕ) : Nontrivial (Lag E (N + 1))

theorem **admissible_scaled_lattice_family** {n N M R : ℕ} (hM : 0 < M) (hR : 0 < R) : admitted M R (divFam E n N R)

theorem **not_admissible_half_lattice_family** {n N M : ℕ} : ¬ admitted M 1 (divFam E (n + 1) (N + 1) 2)

theorem **exists_isType** : ∃ Ψ : Finset (E →+* ℂ), IsType Ψ ∧ Ψ.Nonempty

theorem **wt_rank_zero** (M : ℕ) (Λ : Lt E 0) : wt M Λ = 1

theorem **neg_one_mem_DH** (n : ℕ) : (-1 : Matrix (Fin n) (Fin n) E) ∈ DH E n 2

theorem **twist_indFam_zero** {n N R : ℕ} (Λ : Lt E n) : twist R (stdL E N) (indFam E n N) Λ 0 = 1

theorem **exists_concrete** : ∃ (_ : IsCMField (CyclotomicField 3 ℚ)) (Ψ : Finset (CyclotomicField 3 ℚ →+* ℂ)), IsType Ψ ∧ Ψ.Nonempty

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.NotLatsOKEven

theorem **latsOK_std** (l : Geo.Tor.Level H) (n : ℕ) (Cl : Type) : LatsOK l.M (fun _ : Cl => ((stdLat E n) : GenLat E n)) (fun _ => l.M)

theorem **not_latsOK_even** {M : ℕ} (hM : Even M) {Cl : Type} (k : Cl) (La : Cl → GenLat E n) (Nc : Cl → ℕ) : ¬ LatsOK M La Nc

theorem **zCard_ne_zero** (Dw : Datum H) {M : ℕ} {Cl : Type} {La : Cl → GenLat E n} {Nc : Cl → ℕ} (h : LatsOK M La Nc) (k : Cl) : (La k).zCard (Nc k) (Dw.p + Dw.q) ≠ 0

theorem **clsSum_std** (σ : c.F →+* ℂ) (X : Data E (Dw.p + Dw.q)) (hr : X.p0 + q0 = Dw.p + Dw.q) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin (Dw.p + Dw.q)) (eL : LtN Dw n ≃ Harm.Var X.Ψ q0 X.p0 (Fin n)) (l₂ : Geo.Tor.Level H) (w : (Fin n → LevQ Dw l₂) → c.F) (P : MvPolynomial (LtN Dw n) c.F) (L : Lt E (Dw.p + Dw.q)) (v : TF n g) (Y : (Fin n → LevQ Dw l₂) → Matrix (Fin n) (Fin (Dw.p + Dw.q)) E) (x : Frame X.Ψ (Dw.p + Dw.q)) : clsSum σ X hr bas idx eL (fun _ : Unit => w) (fun _ => (stdLat E n)) (fun _ => l₂.M) (fun _ => 1) 1 (fun _ => 1) (fun _ => P) L v (fun _ => Y) x = ((Nat.card (Fin n → LevQ Dw l₂) : ℂ))⁻¹ * thP X hr bas idx (gphi X bas (H.w0 : E) (stdLat E n) L v (fun q => σ (w q)) Y) (MvPolynomial.rename eL (MvPolynomial.map σ P)) x

theorem **oneSim_inGam** (l l₂ : Geo.Tor.Level H) : (oneSim Dw).InGam l l₂

theorem **datEq_lin** (d : DeltaDat Xo c n) (ψ : (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) →* c.Fˣ) {w w' : (Fin n → LevQ Dw d.l₂) → c.F} (h : DatEq d ψ w) (h' : DatEq d ψ w') (t : c.F) : DatEq d ψ (w + t • w')

theorem **datEq_char** (d : DeltaDat Xo c n) {ψ ψ' : (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) →* c.Fˣ} {w : (Fin n → LevQ Dw d.l₂) → c.F} (h : DatEq d ψ w) (h' : DatEq d ψ' w) (hw : w ≠ 0) (s : Sim Dw) (hs : s.InGam l d.l₂) (kI : LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) (σq : LevQ Dw d.l₂ ≃ LevQ Dw d.l₂) (h1 : ∀ a : Dw.La, (Submodule.Quotient.mk (s.f d.l₂.M a) : LaI Xo c d) = kI (Submodule.Quotient.mk a)) (h2 : ∀ a : Dw.La, σq (Submodule.Quotient.mk a) = Submodule.Quotient.mk (s.f d.l₂.M a)) : ψ kI = ψ' kI

theorem **exists_datEq** (d : DeltaDat Xo c n) (hI : d.l₂.Mb ≤ d.I) (ψ : (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) →* c.Fˣ) (x₀ : Fin n → LaI Xo c d) (hx : Submodule.span (𝒪 E) (Set.range x₀) = ⊤) : ∃ w : (Fin n → LevQ Dw d.l₂) → c.F, DatEq d ψ w ∧ ∀ (k : LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) (a : Fin n → Dw.La), (∀ b, (Submodule.Quotient.mk (a b) : LaI Xo c d) = k (x₀ b)) → w (fun b => Submodule.Quotient.mk (a b)) = (ψ k : c.Fˣ)

theorem **cvIs_unique** (d : DeltaDat Xo c n) [Algebra d.c₂.F ℂ] {X : Data E r} {bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} {idx : Fin g ≃ X.Ψ × Fin r} {e : Fin g ≃ Lt1 Dw} {L : Lt E r} {cv cv' : (Dw.La →ₗ[𝒪 E] (Fin r → E)) → ℂ} (h : CvIs d X bas idx e L cv) (h' : CvIs d X bas idx e L cv') {x : Frame X.Ψ r} (hx : x ∈ X.dom) {y₂ : FrPt d.Xo₂ d.c₂ ℂ} (A : AnFib X bas idx e x L y₂) {mk : Dw.La →ₗ[𝒪 E] (Fin r → E)} (hm : IsMark d.l₂ L mk) (hA : A.Marked mk) : cv mk = cv' mk

theorem **cvIs_const** (d : DeltaDat Xo c n) [Algebra d.c₂.F ℂ] (X : Data E r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (e : Fin g ≃ Lt1 Dw) (L : Lt E r) (t : ℂ) (h : ∀ y₂ : FrPt d.Xo₂ d.c₂ ℂ, evR d.Xo₂ d.c₂ d.cб y₂ = t) : CvIs d X bas idx e L fun _ => t

Development.AbelianVarietyTheory.Definitions.SpanningFamilies.PolarizationScalarImPos

theorem **isOpen_int** (F : Type) [Field F] [NumberField F] : IsOpen (intAf F)

theorem **trace_int** {a : F} (h : algebraMap F (Af F) a ∈ intAf F) : ∃ n : ℤ, Algebra.trace ℚ F a = n

theorem **int_sub** {a b : Af F} (ha : a ∈ intAf F) (hb : b ∈ intAf F) : a - b ∈ intAf F

theorem **int_add** {a b : Af F} (ha : a ∈ intAf F) (hb : b ∈ intAf F) : a + b ∈ intAf F

theorem **int_zero** : (0 : Af F) ∈ intAf F

theorem **exists_char** (t : Ainf F →+ ℂ) (hc : Continuous t) (hre : ∀ x, (t x).im = 0) (ht : ∀ a : F, t (algebraMap F (Ainf F) a) = ((Algebra.trace ℚ F a : ℚ) : ℂ)) : ∃ ψ : AddChar (AF F) ℂ, IsCharAF F ψ ∧ ∀ x, psiInf F ψ x = Complex.exp (2 * Real.pi * Complex.I * (((-1 : ℤ)) : ℂ) * t x)

theorem **exists_stdChar** (D : D7) : ∃ ψ, IsStdChar D ψ (-1)

theorem **mem_latA** {r : ℕ} {x : Fin r → D.host.E} : x ∈ latA D r ↔ ∀ a, x a ∈ D.host.AA

theorem **AA_fg** : (D.host.AA : Submodule (𝒪 D.host.E) D.host.E).FG

theorem **single_mem** {r : ℕ} (i : Fin r) {b : D.host.E} (hb : b ∈ D.host.AA) : Pi.single i b ∈ latA D r

theorem **latA_span** (r : ℕ) : Submodule.span D.host.E (latA D r : Set (Fin r → D.host.E)) = ⊤

theorem **exists_den** : ∃ n : ℕ, 0 < n ∧ ∀ a ∈ D.host.AA, IsIntegral ℤ ((n : D.host.E) * a)

theorem **herm_diag** {r : ℕ} (ε : Fin r → D.host.E) (x y : Fin r → D.host.E) : herm (Matrix.diagonal ε) x y = ∑ a, x a * ε a * cc (y a)

theorem **selfD** {r : ℕ} (X : Data D.host.E r) (neg : Fin r → Bool) (hG : X.Gm = Matrix.diagonal fun a => Examples₂.sgn (neg a)) (hΛ : X.Λ = latA D r) (hκ : X.κ0 = 2 * D.host.xi0) : SelfD X X.Λ

theorem **polarization_scalar_im_pos** (D : D7) (τ : D.what₂.Tau) : 0 < (tauE D τ D.host.xi0).im

theorem **sp_gram** (D : D7) (p q : ℕ) : (sp D p q).gram = Matrix.diagonal fun a : Fin (p + q) => Examples₂.sgn (decide (p ≤ (a : ℕ)))

theorem **exists_ctx** (D : D7) (n p q : ℕ) (hn : Even n) : ∃ C : Ctx (sp D n 0) (sp D p q), C.X.p0 = p ∧ C.q0 = q ∧ C.X.Λ = latA D (p + q) ∧ C.IsB fun _ => 1

theorem **realPt_rat** (y : Matrix (Fin S.rank) (Fin T.rank) D.host.E) : C.realPt (ratW T S yᵀ) = embR C.X y

theorem **isONFr_xB** (B : C.Bas) (h : C.IsB B) : IsONFr C.X (C.xB B)

theorem **EF_eq** (x y : Matrix (Fin S.rank) (Fin T.rank) D.host.E) : ∑ k, EF C.X (x k) (y k) = -Algebra.trace ℚ D.host.E (D.host.xi0 * GlobalTheory.LatVac.hRat S T xᵀ yᵀ)

Development.AbelianVarietyTheory.Definitions.TeichmullerDepth.IsEigBasisIff

theorem **isEigBasis_iff** {O OE : Type} [CommRing O] [CommRing OE] {d s : ℕ} (act : OE → Fin d → PS O 1 d) (τ : OE →+* O) (w : Fin s → Fin d → O) : IsEigBasis act τ w ↔ EigB (fun a => linPart (act a)) τ w

theorem **lin_eq_linPart** {O : Type} [CommRing O] {d : ℕ} (φ : Fin d → PS O 1 d) : lin φ = linPart φ

theorem **splitEx_one** {R : Type} [CommRing R] (d : ℕ) : SplitEx (1 : Matrix (Fin d) (Fin d) R) (0 : Matrix (Fin 0) (Fin d) R)

Development.AbelianVarietyTheory.Definitions.TeichmullerDepth.PsJetEqToJetOfNonempty

theorem **psJet_eq_toJet_of_nonempty** {n : ℕ} [Nonempty (Fin n × Fin d)] (m : ℕ) (f : PS k n d) : psJet m f = GlobalTheory.JetSer.toJet m f

theorem **psJet_eq_toJet_of_pos** (hd : 0 < d) {n : ℕ} (m : ℕ) (f : PS k (n + 1) d) : psJet m f = GlobalTheory.JetSer.toJet m f

theorem **cubeSer_iff_psJet** (hd : 0 < d) {G : Over (Sp k)} [GrpObj G] [IsCommMonObj G] {L : RigLB G} (c : CubicalStr G L) (x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G) (i : L.L.ι) (Φ : PS k 3 d) : CubeSer c x i Φ ↔ ∀ m : ℕ, ∃ h : (⊤ : (Sp (JetR k (Fin 3 × Fin d) m)).Opens) ≤ (jet3 x m).left ⁻¹ᵁ (Theta G L.L).U (fun _ => i), (Semi.res h (pullU (jet3 x m).left _ (c.t.s fun _ => i))).val = (Scheme.ΓSpecIso (CommRingCat.of (JetR k (Fin 3 × Fin d) m))).inv.hom (psJet m Φ)

theorem **CubeSer.unique** {G : Over (Sp k)} [GrpObj G] [IsCommMonObj G] {L : RigLB G} {c : CubicalStr G L} {x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G} {i : L.L.ι} {Φ Φ' : PS k 3 d} (h : CubeSer c x i Φ) (h' : CubeSer c x i Φ') : Φ = Φ'

theorem **CubeSer.isUnit** {G : Over (Sp k)} [GrpObj G] [IsCommMonObj G] {L : RigLB G} {c : CubicalStr G L} {x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G} {i : L.L.ι} {Φ : PS k 3 d} (h : CubeSer c x i Φ) : IsUnit Φ

theorem **cubeSer_one** {G : Over (Sp k)} [GrpObj G] [IsCommMonObj G] {L : RigLB G} (c : CubicalStr G L) (x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G) (i : L.L.ι) (hc : c.t.s (fun _ => i) = 1) (h : ∀ m, (⊤ : (Sp (JetR k (Fin 3 × Fin d) m)).Opens) ≤ (jet3 x m).left ⁻¹ᵁ (Theta G L.L).U (fun _ => i)) : CubeSer c x i 1

theorem **MapSer.constantCoeff_eq_zero** {G B : Over (Sp k)} {q : G ⟶ B} {x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G} {y : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin g) m) ⟶ B} {ψ : Fin g → PS k 1 d} (h : MapSer q x y ψ) (j : Fin g) : MvPowerSeries.constantCoeff (ψ j) = 0

theorem **mapSer_iff_psJet** {G B : Over (Sp k)} (q : G ⟶ B) (x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G) (y : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin g) m) ⟶ B) (ψ : Fin g → PS k 1 d) : MapSer q x y ψ ↔ (∀ j, MvPowerSeries.constantCoeff (ψ j) = 0) ∧ ∀ m : ℕ, ∃ a : JetR k (Fin 1 × Fin g) m →ₐ[k] JetR k (Fin 1 × Fin d) m, (∀ j, a (zR m ((0 : Fin 1), j)) = psJet m (ψ j)) ∧ x m ≫ q = algOverHom a ≫ y m

theorem **mapSer_id** {G : Over (Sp k)} (x : ∀ m : ℕ, algOver k (JetR k (Fin 1 × Fin d) m) ⟶ G) : MapSer (𝟙 G) x x (pt k d 0)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.AevalEqZeroOfHomogeneousComponent

theorem **exists_smul_mem_of_span_eq_top** {R Ω n : Type*} [CommRing R] [Field Ω] [Finite n] (φ : R →+* Ω) (S : Submodule R (n → R)) (h : Submodule.span Ω ((fun v : n → R => φ ∘ v) '' S) = ⊤) : ∃ r, φ r ≠ 0 ∧ ∀ w, r • w ∈ S

theorem **exists_C_mul_X_pow_mem** {R Ω σ : Type*} [CommRing R] [Field Ω] [IsAlgClosed Ω] [Finite σ] (φ : R →+* Ω) {I : Ideal (MvPolynomial σ R)} (hI : I.IsHomogeneous (homogeneousSubmodule σ R)) (h : ∀ x : σ → Ω, (∀ g ∈ I, eval x (map φ g) = 0) → x = 0) : ∃ r, φ r ≠ 0 ∧ ∃ m : ℕ, ∀ j, C r * X j ^ m ∈ I

theorem **aeval_smul_of_isHomogeneous** {G : Type*} [CommRing G] [Algebra K G] {g : MvPolynomial σ K} {n : ℕ} (hg : g.IsHomogeneous n) (c : G) (ξ : σ → G) : aeval (fun j => c * ξ j) g = c ^ n * aeval ξ g

theorem **aeval_eq_zero_of_homogeneousComponent** {G : Type*} [CommRing G] [Algebra K G] {g : MvPolynomial σ K} {ξ : σ → G} (h : ∀ n, aeval ξ (homogeneousComponent n g) = 0) : aeval ξ g = 0

theorem **aeval_smul_eq_zero** {G : Type*} [CommRing G] [Algebra K G] {p : Ideal (MvPolynomial σ K)} (hp : p.IsHomogeneous (homogeneousSubmodule σ K)) {ξ : σ → G} (hξ : ∀ g ∈ p, aeval ξ g = 0) (c : G) : ∀ g ∈ p, aeval (fun j => c * ξ j) g = 0

theorem **forall_aeval_eq_zero_iff** [Fintype σ] {Ω : Type*} [Field Ω] [IsAlgClosed Ω] [Algebra K Ω] {p : Ideal (MvPolynomial σ K)} (hp : p.IsHomogeneous (homogeneousSubmodule σ K)) (t : β × σ → Ω) : (∀ f ∈ elimIdeal β p, aeval t f = 0) ↔ ∃ x : σ → Ω, x ≠ 0 ∧ (∀ g ∈ p, aeval x g = 0) ∧ ∀ i, ∑ j, t (i, j) * x j = 0

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.AwayToSectionFromZero

theorem **_root_.GlobalTheory.ProjFrac.ext** {U : (Proj 𝒜).Opens} {s t : Γ(Proj 𝒜, U)} (h : ∀ x hx, _root_.GlobalTheory.ProjFrac.ev 𝒜 U x hx s = _root_.GlobalTheory.ProjFrac.ev 𝒜 U x hx t) : s = t

theorem **ev_resG** {U V : (Proj 𝒜).Opens} (h : V ≤ U) (x : Proj 𝒜) (hx : x ∈ V) (s : Γ(Proj 𝒜, U)) : ev 𝒜 V x hx (resG (Proj 𝒜) h s) = ev 𝒜 U x (h hx) s

theorem **mem_basicOpen_iff** {U : (Proj 𝒜).Opens} (s : Γ(Proj 𝒜, U)) (x : Proj 𝒜) (hx : x ∈ U) : x ∈ (Proj 𝒜).basicOpen s ↔ IsUnit (ev 𝒜 U x hx s)

theorem **ev_frac** (x : Proj 𝒜) (hx : x ∈ U) : ev 𝒜 U x hx (frac 𝒜 U a b ha hb hU) = Localization.mk a ⟨b, hU hx⟩

theorem **ev_frac_mul_den** (x : Proj 𝒜) (hx : x ∈ U) : ev 𝒜 U x hx (frac 𝒜 U a b ha hb hU) * algebraMap A _ b = algebraMap A _ a

theorem **resG_frac** (h : V ≤ U) : resG (Proj 𝒜) h (frac 𝒜 U a b ha hb hU) = frac 𝒜 V a b ha hb (h.trans hU)

theorem **_root_.GlobalTheory.ProjFrac.frac_eq** (h : a * b' = a' * b) : _root_.GlobalTheory.ProjFrac.frac 𝒜 U a b ha hb hU = _root_.GlobalTheory.ProjFrac.frac 𝒜 U a' b' ha' hb' hU'

theorem **_root_.GlobalTheory.ProjFrac.frac_mul** : _root_.GlobalTheory.ProjFrac.frac 𝒜 U a b ha hb hU * _root_.GlobalTheory.ProjFrac.frac 𝒜 U a' b' ha' hb' hU' = _root_.GlobalTheory.ProjFrac.frac 𝒜 U (a * a') (b * b') (SetLike.mul_mem_graded ha ha') (SetLike.mul_mem_graded hb hb') (by rw [Proj.basicOpen_mul]; exact le_inf hU hU')

theorem **_root_.GlobalTheory.ProjFrac.frac_mul_eq** {d'' : ℕ} {a'' b'' : A} {ha'' : a'' ∈ 𝒜 d''} {hb'' : b'' ∈ 𝒜 d''} {hU'' : U ≤ Proj.basicOpen 𝒜 b''} (h : a * a' * b'' = a'' * (b * b')) : _root_.GlobalTheory.ProjFrac.frac 𝒜 U a b ha hb hU * _root_.GlobalTheory.ProjFrac.frac 𝒜 U a' b' ha' hb' hU' = _root_.GlobalTheory.ProjFrac.frac 𝒜 U a'' b'' ha'' hb'' hU''

theorem **_root_.GlobalTheory.ProjFrac.frac_add** {a₂ : A} {ha₂ : a₂ ∈ 𝒜 d} : _root_.GlobalTheory.ProjFrac.frac 𝒜 U (a + a₂) b (add_mem ha ha₂) hb hU = _root_.GlobalTheory.ProjFrac.frac 𝒜 U a b ha hb hU + _root_.GlobalTheory.ProjFrac.frac 𝒜 U a₂ b ha₂ hb hU

theorem **_root_.GlobalTheory.ProjFrac.frac_self** : _root_.GlobalTheory.ProjFrac.frac 𝒜 U b b hb hb hU = 1

theorem **_root_.GlobalTheory.ProjFrac.frac_zero** : _root_.GlobalTheory.ProjFrac.frac 𝒜 U 0 b (zero_mem _) hb hU = 0

theorem **_root_.GlobalTheory.ProjFrac.frac_pow** (n : ℕ) : _root_.GlobalTheory.ProjFrac.frac 𝒜 U a b ha hb hU ^ n = _root_.GlobalTheory.ProjFrac.frac 𝒜 U (a ^ n) (b ^ n) (SetLike.pow_mem_graded n ha) (SetLike.pow_mem_graded n hb) (fun x hx h => hU hx (x.isPrime.mem_of_pow_mem n h))

theorem **_root_.GlobalTheory.ProjFrac.basicOpen_frac** : (Proj 𝒜).basicOpen (_root_.GlobalTheory.ProjFrac.frac 𝒜 U a b ha hb hU) = U ⊓ Proj.basicOpen 𝒜 a

theorem **awayToSection_mk** {f a : A} {d : ℕ} (hf : f ∈ 𝒜 d) (n : ℕ) (ha : a ∈ 𝒜 (n • d)) : (Proj.awayToSection 𝒜 f).hom (HomogeneousLocalization.Away.mk 𝒜 hf n a ha) = frac 𝒜 (Proj.basicOpen 𝒜 f) a (f ^ n) ha (SetLike.pow_mem_graded n hf) (fun x hx h => hx (x.isPrime.mem_of_pow_mem n h))

theorem **awayToSection_fromZero** (f : A) (a : 𝒜 0) : (Proj.awayToSection 𝒜 f).hom (HomogeneousLocalization.fromZeroRingHom 𝒜 _ a) = frac 𝒜 (Proj.basicOpen 𝒜 f) a 1 a.2 (SetLike.one_mem_graded 𝒜) (by simp)

theorem **map_app_frac** (U : (Proj 𝒜).Opens) {d : ℕ} (a b : A) (ha : a ∈ 𝒜 d) (hb : b ∈ 𝒜 d) (hU : U ≤ Proj.basicOpen 𝒜 b) : ((Proj.map φ hφ).app U).hom (frac 𝒜 U a b ha hb hU) = frac ℬ (Proj.map φ hφ ⁻¹ᵘ U) (φ a) (φ b) (φ.2 ha) (φ.2 hb) (fun _ hx => hU hx)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.CoeNonzeroLocusCanonicalEqComplRange

theorem **secOnOf_mem** {L : LB X} (t : L.Sec) (V : X.Opens) : secOnOf t V ∈ secOn L V

theorem **bijOn_secOnOf_top** (L : LB X) : Set.BijOn (fun t : L.Sec => secOnOf t ⊤) Set.univ (secOn L ⊤)

theorem **ratOf_secOnOf_self** [IsIntegral X] {L : LB X} (hL : L.IsLB) {s : L.Sec} (hs : IsRegularSec s) (V : X.Opens) [Nonempty V] : ratOf hL s V (secOnOf s V) = 1

theorem **isRegularFn_spec** {R : Type} [CommRing R] {r : R} (hr : r ∈ nonZeroDivisors R) : IsRegularFn (X

theorem **exists_mem** (x : X) : ∃ i, x ∈ D.U i

theorem **trans_spec** (i j : D.ι) : resG X (inf_le_left : D.U i ⊓ D.U j ≤ D.U i) (D.f i) = (D.trans i j : Γ(X, D.U i ⊓ D.U j)) ∗ resG X (inf_le_right : D.U i ⊓ D.U j ≤ D.U j) (D.f j)

theorem **trans_spec_on** (i j : D.ι) {W : X.Opens} (hi : W ≤ D.U i) (hj : W ≤ D.U j) : resG X hi (D.f i) = resG X (le_inf hi hj) (D.trans i j : Γ(X, D.U i ⊓ D.U j)) ∗ resG X hj (D.f j)

theorem **eq_trans_on** (i j : D.ι) {W : X.Opens} (hi : W ≤ D.U i) (hj : W ≤ D.U j) {c : Γ(X, W)} (hc : resG X hi (D.f i) = c ∗ resG X hj (D.f j)) : c = resG X (le_inf hi hj) (D.trans i j : Γ(X, D.U i ⊓ D.U j))

theorem **eq_trans** (i j : D.ι) {u : Γ(X, D.U i ⊓ D.U j)ˣ} (hu : resG X (inf_le_left : D.U i ⊓ D.U j ≤ D.U i) (D.f i) = (u : Γ(X, D.U i ⊓ D.U j)) ∗ resG X (inf_le_right : D.U i ⊓ D.U j ≤ D.U j) (D.f j)) : u = D.trans i j

theorem **isLB_lb** : D.lb.IsLB where

@[simp] theorem **canonical_f** (i : D.ι) : D.canonical.f i = D.f i

theorem **isRegularSec_canonical** : IsRegularSec D.canonical

theorem **associated_germ** (i j : D.ι) (x : X) (hi : x ∈ D.U i) (hj : x ∈ D.U j) : Associated ((X.presheaf.germ _ x hi).hom (D.f i)) ((X.presheaf.germ _ x hj).hom (D.f j))

theorem **exists_unit_germ** (c c' : κ) (x : X) (hx : x ∈ V c) (hx' : x ∈ V c') : ∃ u : (X.presheaf.stalk x)ˣ, X.presheaf.germ (V c) x hx (g c) = u ∗ X.presheaf.germ (V c') x hx' (g c')

theorem **mem_locIdeal_iff** {W : X.affineOpens} {a : Γ(X, W.1)} : a ∈ (locIdeal V g hV hg).ideal W ↔ ∀ c x (hx : x ∈ W.1) (hc : x ∈ V c), (X.presheaf.germ _ x hc).hom (g c) ∣ (X.presheaf.germ _ x hx).hom a

theorem **ideal_locIdeal** (W : X.affineOpens) (c : κ) (h : W.1 ≤ V c) : (locIdeal V g hV hg).ideal W = Ideal.span {resG X h (g c)}

theorem **ideal_idealSheaf** (W : X.affineOpens) (i : D.ι) (h : W.1 ≤ D.U i) : D.idealSheaf.ideal W = Ideal.span {resG X h (D.f i)}

theorem **mem_idealSheaf_iff** {W : X.affineOpens} {a : Γ(X, W.1)} : a ∈ D.idealSheaf.ideal W ↔ ∀ i, resG X (inf_le_right : W.1 ⊓ D.U i ≤ D.U i) (D.f i) ∣ resG X (inf_le_left : W.1 ⊓ D.U i ≤ W.1) a

theorem **exists_affine_le** {V : X.Opens} {x : X} (hx : x ∈ V) : ∃ W : X.affineOpens, x ∈ W.1 ∧ W.1 ≤ V

theorem **stalkId_idealSheaf** (i : D.ι) (x : X) (hx : x ∈ D.U i) : FinalTheorems.stalkId D.idealSheaf x = Ideal.span {(X.presheaf.germ _ x hx).hom (D.f i)}

theorem **eq_idealSheaf** {I : X.IdealSheafData} (h : ∀ x : X, ∃ (W : X.affineOpens) (i : D.ι) (hW : W.1 ≤ D.U i), x ∈ W.1 ∧ I.ideal W = Ideal.span {resG X hW (D.f i)}) : I = D.idealSheaf

theorem **effCartier_idealSheaf** : FinalTheorems.EffCartier D.idealSheaf

theorem **mem_support_iff** (i : D.ι) {x : X} (hx : x ∈ D.U i) : x ∈ D.idealSheaf.support ↔ x ∉ X.basicOpen (D.f i)

theorem **coe_nonzeroLocus_canonical** : (Out.nonzeroLocus D.canonical : Set X) = (D.idealSheaf.support : Set X)ᶜ

theorem **coe_nonzeroLocus_canonical_eq_compl_range** : (Out.nonzeroLocus D.canonical : Set X) = (Set.range D.zeroSchemeι)ᶜ

theorem **dense_nonzeroLocus_canonical** : Dense (Out.nonzeroLocus D.canonical : Set X)

theorem **associated_of_idealSheaf_eq** (h : D.idealSheaf = D'.idealSheaf) (p : D'.ι) (q : D.ι) : Associated (resG X (inf_le_right : D'.U p ⊓ D.U q ≤ D.U q) (D.f q)) (resG X (inf_le_left : D'.U p ⊓ D.U q ≤ D'.U p) (D'.f p))

theorem **existsUnique_iso_of_idealSheaf_eq** (h : D.idealSheaf = D'.idealSheaf) : ∃! w : LB.Iso D.lb D'.lb, w.IsGen ∧ SecMaps w D.canonical D'.canonical

theorem **lbIso_of_idealSheaf_eq** (h : D.idealSheaf = D'.idealSheaf) : LBIso D.lb D'.lb

theorem **idealSheaf_eq_of_secMaps** {w : LB.Iso D.lb D'.lb} (hw : SecMaps w D.canonical D'.canonical) : D.idealSheaf = D'.idealSheaf

theorem **idealSheaf_eq_iff** : D.idealSheaf = D'.idealSheaf ↔ ∃ w : LB.Iso D.lb D'.lb, w.IsGen ∧ SecMaps w D.canonical D'.canonical

theorem **exists_idealSheaf_eq** {I : X.IdealSheafData} (hI : FinalTheorems.EffCartier I) : ∃ D : EffCartierData X, D.idealSheaf = I

theorem **lb_ofSec** {L : LB X} (hL : L.IsLB) (s : L.Sec) (hs : IsRegularSec s) : (ofSec hL s hs).lb = L

theorem **existsUnique_iso_ofSec** {L : LB X} (hL : L.IsLB) (s : L.Sec) (hs : IsRegularSec s) : ∃! w : LB.Iso L (ofSec hL s hs).lb, w.IsGen ∧ SecMaps w s (ofSec hL s hs).canonical

theorem **existsUnique_iso_ofSec'** {L : LB X} (hL : L.IsLB) (s : L.Sec) (hs : IsRegularSec s) : ∃! w : LB.Iso (ofSec hL s hs).lb L, w.IsGen ∧ SecMaps w (ofSec hL s hs).canonical s

theorem **lbIso_ofSec** {L : LB X} (hL : L.IsLB) (s : L.Sec) (hs : IsRegularSec s) : LBIso (ofSec hL s hs).lb L

theorem **ofSec_canonical** : ofSec D.isLB_lb D.canonical D.isRegularSec_canonical = D

theorem **coe_nonzeroLocus_eq_compl_support_ofSec** {L : LB X} (hL : L.IsLB) (s : L.Sec) (hs : IsRegularSec s) : (Out.nonzeroLocus s : Set X) = ((ofSec hL s hs).idealSheaf.support : Set X)ᶜ

theorem **germ_app** (φ : Y ⟶ X) {U : X.Opens} (a : Γ(X, U)) (y : Y) (hy : φ y ∈ U) : (Y.presheaf.germ (φ ⁻¹ᵁ U) y hy).hom ((φ.app U).hom a) = (φ.stalkMap y).hom ((X.presheaf.germ U (φ y) hy).hom a)

theorem **associated_germ_app** (φ : Y ⟶ X) (i j : D.ι) (y : Y) (hi : y ∈ φ ⁻¹ᵁ D.U i) (hj : y ∈ φ ⁻¹ᵁ D.U j) : Associated ((Y.presheaf.germ _ y hi).hom ((φ.app (D.U i)).hom (D.f i))) ((Y.presheaf.germ _ y hj).hom ((φ.app (D.U j)).hom (D.f j)))

theorem **comap_idealSheaf** (φ : Y ⟶ X) : D.idealSheaf.comap φ = locIdeal (fun i => φ ⁻¹ᵁ D.U i) (fun i => (φ.app (D.U i)).hom (D.f i)) (fun y => D.exists_mem (φ y)) (D.associated_germ_app φ)

theorem **ideal_comap_idealSheaf** (φ : Y ⟶ X) (W : Y.affineOpens) (i : D.ι) (h : W.1 ≤ φ ⁻¹ᵁ D.U i) : (D.idealSheaf.comap φ).ideal W = Ideal.span {resG Y h ((φ.app (D.U i)).hom (D.f i))}

theorem **keepsRegular_iff_effCartier** (φ : Y ⟶ X) : D.KeepsRegular φ ↔ FinalTheorems.EffCartier (D.idealSheaf.comap φ)

theorem **keepsRegular_congr** (h : D.idealSheaf = D'.idealSheaf) (φ : Y ⟶ X) : D.KeepsRegular φ ↔ D'.KeepsRegular φ

theorem **keepsRegular_iff_isRegularSec** (φ : Y ⟶ X) : D.KeepsRegular φ ↔ IsRegularSec (secPull D.canonical φ)

theorem **keepsRegular_of_flat** (φ : Y ⟶ X) [Flat φ] : D.KeepsRegular φ

theorem **keepsRegular_of_notMem_support** [IsIntegral Y] (φ : Y ⟶ X) {y : Y} (hy : φ y ∉ D.idealSheaf.support) : D.KeepsRegular φ

theorem **keepsRegular_of_isDominant** [IsIntegral Y] (φ : Y ⟶ X) [IsDominant φ] : D.KeepsRegular φ

theorem **idealSheaf_pull** (φ : Y ⟶ X) (h : D.KeepsRegular φ) : (D.pull φ h).idealSheaf = D.idealSheaf.comap φ

theorem **existsUnique_iso_pull** (φ : Y ⟶ X) (h : D.KeepsRegular φ) : ∃! w : LB.Iso (D.pull φ h).lb (D.lb.pull φ), w.IsGen ∧ SecMaps w (D.pull φ h).canonical (secPull D.canonical φ)

theorem **lbIso_pull** (φ : Y ⟶ X) (h : D.KeepsRegular φ) : LBIso (D.pull φ h).lb (D.lb.pull φ)

theorem **keepsRegular_comp_iff** (φ : Y ⟶ X) (h : D.KeepsRegular φ) (ψ : Z ⟶ Y) : D.KeepsRegular (ψ ≫ φ) ↔ (D.pull φ h).KeepsRegular ψ

theorem **pull_comp** (φ : Y ⟶ X) (h : D.KeepsRegular φ) (ψ : Z ⟶ Y) (h' : (D.pull φ h).KeepsRegular ψ) : D.pull (ψ ≫ φ) ((D.keepsRegular_comp_iff φ h ψ).2 h') = (D.pull φ h).pull ψ h'

theorem **pull_id** : D.pull (𝟙 X) (D.keepsRegular_of_flat _) = D

theorem **resG_add_f** (i : D.ι) (j : D'.ι) {W : X.Opens} (h : W ≤ D.U i ⊓ D'.U j) : resG X h ((D.add D').f (i, j)) = resG X (h.trans inf_le_left) (D.f i) * resG X (h.trans inf_le_right) (D'.f j)

theorem **idealSheaf_add** : (D.add D').idealSheaf = D.idealSheaf * D'.idealSheaf

theorem **res_trans_add** (p q : D.ι × D'.ι) {W : X.Opens} (hp : W ≤ D.U p.1 ⊓ D'.U p.2) (hq : W ≤ D.U q.1 ⊓ D'.U q.2) : res (le_inf hp hq) ((D.add D').trans p q) = res (le_inf (hp.trans inf_le_left) (hq.trans inf_le_left)) (D.trans p.1 q.1) * res (le_inf (hp.trans inf_le_right) (hq.trans inf_le_right)) (D'.trans p.2 q.2)

theorem **res_addGen** (i j : D.ι) (i' j' : D'.ι) {W : X.Opens} (h : W ≤ D.U i ⊓ D'.U i' ⊓ D.U j ⊓ D'.U j') : res h ((D.addGen D').s (((i, i'), j), j')) = res (le_inf (h.trans (inf_le_left.trans (inf_le_left.trans inf_le_left))) (h.trans (inf_le_left.trans inf_le_right))) (D.trans i j) * res (le_inf (h.trans (inf_le_left.trans (inf_le_left.trans inf_le_right))) (h.trans inf_le_right)) (D'.trans i' j')

theorem **addGen_canonical** (i j : D.ι) (i' j' : D'.ι) {W : X.Opens} (h : W ≤ D.U i ⊓ D'.U i' ⊓ D.U j ⊓ D'.U j') : resG X (h.trans (inf_le_left.trans inf_le_left)) ((D.add D').canonical.f (i, i')) = (res h ((D.addGen D').s (((i, i'), j), j')) : Γ(X, W)ˣ) * (resG X (h.trans (inf_le_left.trans inf_le_right)) (D.canonical.f j) * resG X (h.trans inf_le_right) (D'.canonical.f j'))

theorem **isGen_of_mul** {N L M : LB X} {a : L.Sec} {b : M.Sec} {c : N.Sec} (ha : IsRegularSec a) (hb : IsRegularSec b) (w : LB.Gen (LB.div (LB.div N L) M)) (H : ∀ (i : N.ι) (j : L.ι) (k : M.ι) {W : X.Opens} (h : W ≤ N.U i ⊓ L.U j ⊓ M.U k), resG X (h.trans (inf_le_left.trans inf_le_left)) (c.f i) = (res h (w.s ((i, j), k)) : Γ(X, W)ˣ) * (resG X (h.trans (inf_le_left.trans inf_le_right)) (a.f j) * resG X (h.trans inf_le_right) (b.f k))) : w.IsGen

theorem **isGen_addGen** : (D.addGen D').IsGen

theorem **lbIso_add** : LBIso D'.lb (LB.div (D.add D').lb D.lb)

theorem **app_add_f** (φ : Y ⟶ X) (p : D.ι × D'.ι) : (φ.app ((D.add D').U p)).hom ((D.add D').f p) = resG Y (φ.preimage_mono inf_le_left) ((φ.app (D.U p.1)).hom (D.f p.1)) * resG Y (φ.preimage_mono inf_le_right) ((φ.app (D'.U p.2)).hom (D'.f p.2))

theorem **KeepsRegular.add** {D D' : EffCartierData X} {φ : Y ⟶ X} (h : D.KeepsRegular φ) (h' : D'.KeepsRegular φ) : (D.add D').KeepsRegular φ

theorem **pull_add** (φ : Y ⟶ X) (h : D.KeepsRegular φ) (h' : D'.KeepsRegular φ) : (D.add D').pull φ (h.add h') = (D.pull φ h).add (D'.pull φ h')

theorem **idealSheaf_nsmul** (n : ℕ) : (D.nsmul n).idealSheaf = D.idealSheaf ^ n

theorem **existsUnique_iso_nsmul** (n : ℕ) : ∃! w : LB.Iso (D.nsmul n).lb (lbPow D.lb n), w.IsGen ∧ SecMaps w (D.nsmul n).canonical (secPow D.canonical n)

theorem **lbIso_nsmul** (n : ℕ) : LBIso (D.nsmul n).lb (lbPow D.lb n)

theorem **idealSheaf_principal** (a : Γ(X, ⊤)) (ha : IsRegularFn a) : (principal a ha).idealSheaf = Scheme.IdealSheafData.ofIdealTop (Ideal.span {a})

theorem **lb_principal** (a : Γ(X, ⊤)) (ha : IsRegularFn a) : (principal a ha).lb = LB.triv X

theorem **idealSheaf_zero** : (zero X).idealSheaf = ⊤

theorem **mem_poleSecs_iff** (n : ℕ) (V : X.Opens) [Nonempty V] (h : X.functionField) : h ∈ D.poleSecs n V ↔ ∀ (i : D.ι) [Nonempty (V ⊓ D.U i : X.Opens)], ∃ a : Γ(X, V ⊓ D.U i), (X.germToFunctionField _).hom a = (X.germToFunctionField (V ⊓ D.U i)).hom (resG X inf_le_right (D.f i)) ^ n * h

theorem **mem_poleSecs_iff_stalk** (n : ℕ) (V : X.Opens) [Nonempty V] (h : X.functionField) : h ∈ D.poleSecs n V ↔ ∀ (i : D.ι) (x : X) (_ : x ∈ V) (hx : x ∈ D.U i), algebraMap (X.presheaf.stalk x) X.functionField ((X.presheaf.germ _ x hx).hom (D.f i)) ^ n * h ∈ Set.range (algebraMap (X.presheaf.stalk x) X.functionField)

theorem **poleSecs_congr** (h : D.idealSheaf = D'.idealSheaf) (n : ℕ) (V : X.Opens) [Nonempty V] : D.poleSecs n V = D'.poleSecs n V

theorem **mem_poleSecs_zero_iff** (V : X.Opens) [Nonempty V] (h : X.functionField) : h ∈ D.poleSecs 0 V ↔ ∃ a : Γ(X, V), (X.germToFunctionField V).hom a = h

theorem **poleSecs_mono** (V : X.Opens) [Nonempty V] : Monotone fun n => D.poleSecs n V

theorem **mul_mem_poleSecs** {m n : ℕ} {V : X.Opens} [Nonempty V] {h h' : X.functionField} (hh : h ∈ D.poleSecs m V) (hh' : h' ∈ D.poleSecs n V) : h * h' ∈ D.poleSecs (m + n) V

theorem **bijOn_poleSecs_nsmul** (n : ℕ) (V : X.Opens) [Nonempty V] : Set.BijOn (ratOf (D.nsmul n).isLB_lb (D.nsmul n).canonical V) (secOn (D.nsmul n).lb V) (D.poleSecs n V)

theorem **germK_eq_pow_mul_ratOf** (n : ℕ) (V : X.Opens) [Nonempty V] {t : ∀ i, Γ(X, V ⊓ (D.nsmul n).lb.U i)} (ht : t ∈ secOn (D.nsmul n).lb V) (i : D.ι) [hi : Nonempty (V ⊓ D.U i : X.Opens)] : (X.germToFunctionField (V ⊓ D.U i)).hom (t i) = (X.germToFunctionField (V ⊓ D.U i)).hom (resG X inf_le_right (D.f i)) ^ n * ratOf (D.nsmul n).isLB_lb (D.nsmul n).canonical V t

theorem **isLB_lbPow** {X : Scheme.{0}} (D : EffCartierData X) (n : ℕ) : (lbPow D.lb n).IsLB

theorem **bijOn_poleSecs_lbPow** (n : ℕ) (V : X.Opens) [Nonempty V] : Set.BijOn (ratOf (D.isLB_lbPow n) (secPow D.canonical n) V) (secOn (lbPow D.lb n) V) (D.poleSecs n V)

theorem **bijOn_poleSecs_one** (V : X.Opens) [Nonempty V] : Set.BijOn (ratOf D.isLB_lb D.canonical V) (secOn D.lb V) (D.poleSecs 1 V)

theorem **germK_eq_pow_mul_ratOf_lbPow** (n : ℕ) (V : X.Opens) [Nonempty V] {t : ∀ j, Γ(X, V ⊓ (lbPow D.lb n).U j)} (ht : t ∈ secOn (lbPow D.lb n) V) (j : PUnit → D.ι) [hj : Nonempty (V ⊓ (lbPow D.lb n).U j : X.Opens)] : (X.germToFunctionField (V ⊓ (lbPow D.lb n).U j)).hom (t j) = (X.germToFunctionField (V ⊓ (lbPow D.lb n).U j)).hom (resG X (inf_le_right.trans (lbPow_U D.lb n j).le) (D.f (j ⟨⟩))) ^ n * ratOf (D.isLB_lbPow n) (secPow D.canonical n) V t

theorem **bijOn_poleSecs_global** (n : ℕ) : Set.BijOn (fun t => ratOf (D.isLB_lbPow n) (secPow D.canonical n) ⊤ (secOnOf t ⊤)) Set.univ (D.poleSecs n ⊤)

theorem **ratOf_secPow_canonical** (n : ℕ) (V : X.Opens) [Nonempty V] : ratOf (D.isLB_lbPow n) (secPow D.canonical n) V (secOnOf (secPow D.canonical n) V) = 1

theorem **mem_poleSecs_principal_iff** (a : Γ(X, ⊤)) (ha : IsRegularFn a) (n : ℕ) (V : X.Opens) [Nonempty V] (h : X.functionField) : h ∈ (principal a ha).poleSecs n V ↔ ∃ b : Γ(X, V), (X.germToFunctionField V).hom b = (X.germToFunctionField V).hom (resG X le_top a) ^ n * h

theorem **mem_support_ofElement_iff** {R : Type} [CommRing R] (r : R) (hr : r ∈ nonZeroDivisors R) (p : PrimeSpectrum R) : (p : Sp R) ∈ (ofElement r hr).idealSheaf.support ↔ r ∈ p.asIdeal

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.EqZeroOrSingleOfDegreeLtTwo

@[simp] theorem **jacobian_apply** (f : τ → MvPowerSeries σ R) (i : τ) (j : σ) : jacobian f i j = coeff (Finsupp.single j 1) (f i)

@[simp] theorem **jacobian_X** [DecidableEq σ] : jacobian (X : σ → MvPowerSeries σ R) = 1

theorem **le_order_sub** {k : ℕ∞} {f g : MvPowerSeries σ R} (hf : k ≤ f.order) (hg : k ≤ g.order) : k ≤ (f - g).order

theorem **le_order_sum** {ι : Type*} {k : ℕ∞} (s : Finset ι) {f : ι → MvPowerSeries σ R} (hf : ∀ i ∈ s, k ≤ (f i).order) : k ≤ (∑ i ∈ s, f i).order

theorem **coeff_eq_of_le_order_sub** {k : ℕ} {f g : MvPowerSeries σ R} (h : (k : ℕ∞) ≤ (f - g).order) {d : σ →₀ ℕ} (hd : d.degree < k) : coeff d f = coeff d g

theorem **eq_of_forall_le_order_sub** {f g : MvPowerSeries σ R} (h : ∀ k : ℕ, (k : ℕ∞) ≤ (f - g).order) : f = g

theorem **eq_zero_or_single_of_degree_lt_two** {d : σ →₀ ℕ} (hd : d.degree < 2) : d = 0 ∨ ∃ j, d = Finsupp.single j 1

theorem **two_le_order_iff** {f : MvPowerSeries σ R} : 2 ≤ f.order ↔ constantCoeff f = 0 ∧ ∀ j, coeff (Finsupp.single j 1) f = 0

theorem **two_le_order_sub_linear** [Fintype σ] {f : MvPowerSeries σ R} (hf : constantCoeff f = 0) : 2 ≤ (f - ∑ j, coeff (Finsupp.single j 1) f • X j).order

private theorem **Close.one** (k : ℕ) : Close k 0 (1 : MvPowerSeries τ S) 1

private theorem **Close.mul** {k n m : ℕ} {P Q P' Q' : MvPowerSeries τ S} (h : Close k n P Q) (h' : Close k m P' Q') : Close k (n + m) (P * P') (Q * Q')

private theorem **close_pow_of_close_one** {k e : ℕ} {x y : MvPowerSeries τ S} (h : Close k 1 x y) : Close k e (x ^ e) (y ^ e)

private theorem **Close.prod** {k : ℕ} {a b : σ → MvPowerSeries τ S} (h : ∀ s, Close k 1 (a s) (b s)) (d : σ →₀ ℕ) : Close k d.degree (d.prod fun s e => a s ^ e) (d.prod fun s e => b s ^ e)

private theorem **natCast_lt_of_le_add_one_of_add_two_le** {x : ℕ∞} {n m : ℕ} (h : (n : ℕ∞) ≤ x + 1) (hm : m + 2 ≤ n) : (m : ℕ∞) < x

theorem **le_order_subst_sub_subst** [Finite σ] {a b : σ → MvPowerSeries τ S} (ha : ∀ s, constantCoeff (a s) = 0) (hb : ∀ s, constantCoeff (b s) = 0) {k : ℕ} (hab : ∀ s, (k : ℕ∞) ≤ (a s - b s).order) {h : MvPowerSeries σ R} (hh : 2 ≤ h.order) : ((k + 1 : ℕ) : ℕ∞) ≤ (h.subst a - h.subst b).order

theorem **subst_eq_linear_add** [Fintype σ] {a : σ → MvPowerSeries τ S} (ha : HasSubst a) (f : MvPowerSeries σ R) : f.subst a = ∑ j, coeff (Finsupp.single j 1) f • a j + (f - ∑ j, coeff (Finsupp.single j 1) f • X j).subst a

private theorem **coeff_single_subst_eq_zero_of_linear_coeff_zero** [Finite σ] {a : σ → MvPowerSeries τ S} (ha : ∀ s, constantCoeff (a s) = 0) {g : MvPowerSeries σ R} (hg : ∀ j, coeff (Finsupp.single j 1) g = 0) (k : τ) : coeff (Finsupp.single k 1) (g.subst a) = 0

theorem **coeff_single_subst** [Fintype σ] {a : σ → MvPowerSeries τ S} (ha : ∀ s, constantCoeff (a s) = 0) (f : MvPowerSeries σ R) (k : τ) : coeff (Finsupp.single k 1) (f.subst a) = ∑ j, coeff (Finsupp.single j 1) f • coeff (Finsupp.single k 1) (a j)

theorem **jacobian_subst** [Fintype τ] {f : τ → MvPowerSeries σ R} (hf : ∀ s, constantCoeff (f s) = 0) (g : υ → MvPowerSeries τ R) : jacobian (fun i => (g i).subst f) = jacobian g * jacobian f

theorem **existsUnique_fixedPoint** (Φ : (σ → MvPowerSeries τ S) → (σ → MvPowerSeries τ S)) (h0 : ∀ a, (∀ s, constantCoeff (a s) = 0) → ∀ s, constantCoeff (Φ a s) = 0) (hΦ : ∀ a b (k : ℕ), (∀ s, constantCoeff (a s) = 0) → (∀ s, constantCoeff (b s) = 0) → (∀ s, (k : ℕ∞) ≤ (a s - b s).order) → ∀ s, ((k + 1 : ℕ) : ℕ∞) ≤ (Φ a s - Φ b s).order) : ∃! a, (∀ s, constantCoeff (a s) = 0) ∧ Φ a = a

theorem **exists_subst_eq** [Fintype σ] [Fintype τ] [DecidableEq τ] {u : τ → MvPowerSeries σ R} (hu : ∀ i, constantCoeff (u i) = 0) {B : Matrix σ τ R} (hB : jacobian u * B = 1) {c : τ → MvPowerSeries υ R} (hc : ∀ i, constantCoeff (c i) = 0) : ∃ v : σ → MvPowerSeries υ R, (∀ j, constantCoeff (v j) = 0) ∧ ∀ i, (u i).subst v = c i

theorem **exists_subst_inverse** [Fintype σ] [Fintype τ] [DecidableEq σ] [DecidableEq τ] {u : τ → MvPowerSeries σ R} (hu : ∀ i, constantCoeff (u i) = 0) {B : Matrix σ τ R} (hB : jacobian u * B = 1) (hB' : B * jacobian u = 1) : ∃ v : σ → MvPowerSeries τ R, (∀ j, constantCoeff (v j) = 0) ∧ (∀ i, (u i).subst v = X i) ∧ ∀ j, (v j).subst u = X j

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.EventuallyDifferentiableAtOfContDiffAtTwo

theorem **realCLM_map_complex_smul** (L : E →L[ℝ] F) (c : ℂ) (v : E) : L (c • v) = (c.re : ℂ) • L v + (c.im : ℂ) • L (I • v)

theorem **I_smul_I_smul** (v : E) : I • I • v = -v

theorem **linPart_apply** (L : E →L[ℝ] F) (v : E) : linPart L v = (1 / 2 : ℂ) • (L v - I • L (I • v))

theorem **antilinPart_apply** (L : E →L[ℝ] F) (v : E) : antilinPart L v = (1 / 2 : ℂ) • (L v + I • L (I • v))

theorem **linPart_add_antilinPart** (L : E →L[ℝ] F) (v : E) : linPart L v + antilinPart L v = L v

theorem **linPart_apply_of_map_I** {L : E →L[ℝ] F} {v : E} (h : L (I • v) = I • L v) : linPart L v = L v

theorem **antilinPart_apply_of_map_I** {L : E →L[ℝ] F} {v : E} (h : L (I • v) = I • L v) : antilinPart L v = 0

theorem **linPart_restrictScalars** (L : E →L[ℂ] F) : linPart (L.restrictScalars ℝ) = L

theorem **antilinPart_restrictScalars** (L : E →L[ℂ] F) : antilinPart (L.restrictScalars ℝ) = 0

theorem **norm_antilinPart_le** (L : E →L[ℝ] F) : ‖antilinPart L‖ ≤ ‖L‖

theorem **norm_linPart_le** (L : E →L[ℝ] F) : ‖linPart L‖ ≤ ‖L‖

@[simp] theorem **linPartCLM_apply** (L : E →L[ℝ] F) : linPartCLM E F L = linPart L

@[simp] theorem **antilinPartCLM_apply** (L : E →L[ℝ] F) : antilinPartCLM E F L = antilinPart L

theorem **del_apply** (f : E → F) (z v : E) : del f z v = (1 / 2 : ℂ) • (fderiv ℝ f z v - I • fderiv ℝ f z (I • v))

theorem **dbar_apply** (f : E → F) (z v : E) : dbar f z v = (1 / 2 : ℂ) • (fderiv ℝ f z v + I • fderiv ℝ f z (I • v))

theorem **del_add_dbar** (f : E → F) (z v : E) : del f z v + dbar f z v = fderiv ℝ f z v

theorem **del_eq_of_hasFDerivAt** {L : E →L[ℝ] F} (h : HasFDerivAt f L z) : del f z = linPart L

theorem **dbar_eq_of_hasFDerivAt** {L : E →L[ℝ] F} (h : HasFDerivAt f L z) : dbar f z = antilinPart L

theorem **del_congr** (h : f =ᶠ[𝓝 z] g) : del f z = del g z

theorem **dbar_congr** (h : f =ᶠ[𝓝 z] g) : dbar f z = dbar g z

theorem **del_eq_fderiv** (hf : DifferentiableAt ℂ f z) : del f z = fderiv ℂ f z

theorem **dbar_eq_zero** (hf : DifferentiableAt ℂ f z) : dbar f z = 0

@[simp] theorem **del_const** (c : F) (z : E) : del (fun _ => c) z = 0

@[simp] theorem **dbar_const** (c : F) (z : E) : dbar (fun _ => c) z = 0

theorem **del_add** (hf : DifferentiableAt ℝ f z) (hg : DifferentiableAt ℝ g z) : del (fun x => f x + g x) z = del f z + del g z

theorem **dbar_add** (hf : DifferentiableAt ℝ f z) (hg : DifferentiableAt ℝ g z) : dbar (fun x => f x + g x) z = dbar f z + dbar g z

theorem **del_neg** (f : E → F) (z : E) : del (fun x => -f x) z = -del f z

theorem **dbar_neg** (f : E → F) (z : E) : dbar (fun x => -f x) z = -dbar f z

theorem **del_sub** (hf : DifferentiableAt ℝ f z) (hg : DifferentiableAt ℝ g z) : del (fun x => f x - g x) z = del f z - del g z

theorem **dbar_sub** (hf : DifferentiableAt ℝ f z) (hg : DifferentiableAt ℝ g z) : dbar (fun x => f x - g x) z = dbar f z - dbar g z

theorem **del_sum** {ι : Type*} {t : Finset ι} {A : ι → E → F} (h : ∀ i ∈ t, DifferentiableAt ℝ (A i) z) : del (fun x => ∑ i ∈ t, A i x) z = ∑ i ∈ t, del (A i) z

theorem **dbar_sum** {ι : Type*} {t : Finset ι} {A : ι → E → F} (h : ∀ i ∈ t, DifferentiableAt ℝ (A i) z) : dbar (fun x => ∑ i ∈ t, A i x) z = ∑ i ∈ t, dbar (A i) z

theorem **del_smul** {c : E → ℂ} (hc : DifferentiableAt ℝ c z) (hf : DifferentiableAt ℝ f z) (v : E) : del (fun x => c x • f x) z v = c z • del f z v + del c z v • f z

theorem **dbar_smul** {c : E → ℂ} (hc : DifferentiableAt ℝ c z) (hf : DifferentiableAt ℝ f z) (v : E) : dbar (fun x => c x • f x) z v = c z • dbar f z v + dbar c z v • f z

theorem **del_mul** {a b : E → ℂ} (ha : DifferentiableAt ℝ a z) (hb : DifferentiableAt ℝ b z) (v : E) : del (fun x => a x * b x) z v = a z * del b z v + del a z v * b z

theorem **dbar_mul** {a b : E → ℂ} (ha : DifferentiableAt ℝ a z) (hb : DifferentiableAt ℝ b z) (v : E) : dbar (fun x => a x * b x) z v = a z * dbar b z v + dbar a z v * b z

theorem **del_const_smul** (c : ℂ) (hf : DifferentiableAt ℝ f z) : del (fun x => c • f x) z = c • del f z

theorem **dbar_const_smul** (c : ℂ) (hf : DifferentiableAt ℝ f z) : dbar (fun x => c • f x) z = c • dbar f z

theorem **del_clm_comp** (T : F →L[ℂ] G) (hf : DifferentiableAt ℝ f z) (v : E) : del (fun x => T (f x)) z v = T (del f z v)

theorem **dbar_conj** (a : E → ℂ) (z v : E) : dbar (fun x => conj (a x)) z v = conj (del a z v)

theorem **del_conj** (a : E → ℂ) (z v : E) : del (fun x => conj (a x)) z v = conj (dbar a z v)

theorem **dbar_ofReal** (u : E → ℝ) (z v : E) : dbar (fun x => (u x : ℂ)) z v = conj (del (fun x => (u x : ℂ)) z v)

theorem **del_comp** {u : F → G} {φ : E → F} (hu : DifferentiableAt ℝ u (φ z)) (hφ : DifferentiableAt ℝ φ z) (v : E) : del (fun x => u (φ x)) z v = del u (φ z) (del φ z v) + dbar u (φ z) (dbar φ z v)

theorem **del_comp_holo** {u : F → G} {φ : E → F} (hu : DifferentiableAt ℝ u (φ z)) (hφ : DifferentiableAt ℂ φ z) (v : E) : del (fun x => u (φ x)) z v = del u (φ z) (fderiv ℂ φ z v)

theorem **dbar_comp_holo** {u : F → G} {φ : E → F} (hu : DifferentiableAt ℝ u (φ z)) (hφ : DifferentiableAt ℂ φ z) (v : E) : dbar (fun x => u (φ x)) z v = dbar u (φ z) (fderiv ℂ φ z v)

theorem **fderiv_ofReal_apply** {u : E → ℝ} (hu : DifferentiableAt ℝ u z) (v : E) : fderiv ℝ (fun x => (u x : ℂ)) z v = (fderiv ℝ u z v : ℂ)

theorem **del_ofReal_comp** {t : E → ℝ} {h : ℝ → ℝ} {h' : ℝ} (hh : HasDerivAt h h' (t z)) (ht : DifferentiableAt ℝ t z) (v : E) : del (fun x => (h (t x) : ℂ)) z v = h' * del (fun x => (t x : ℂ)) z v

theorem **dbar_ofReal_comp** {t : E → ℝ} {h : ℝ → ℝ} {h' : ℝ} (hh : HasDerivAt h h' (t z)) (ht : DifferentiableAt ℝ t z) (v : E) : dbar (fun x => (h (t x) : ℂ)) z v = h' * dbar (fun x => (t x : ℂ)) z v

theorem **contDiffAt_dbar** {m n : WithTop ℕ∞} (hf : ContDiffAt ℝ n f z) (h : m + 1 ≤ n) : ContDiffAt ℝ m (dbar f) z

theorem **contDiffAt_del** {m n : WithTop ℕ∞} (hf : ContDiffAt ℝ n f z) (h : m + 1 ≤ n) : ContDiffAt ℝ m (del f) z

theorem **differentiableAt_fderiv_of_contDiffAt_two** (hu : ContDiffAt ℝ 2 u z) : DifferentiableAt ℝ (fderiv ℝ u) z

theorem **eventually_differentiableAt_of_contDiffAt_two** {𝕜 : Type*} [NontriviallyNormedField 𝕜] [NormedSpace 𝕜 E] [NormedSpace 𝕜 F] (hu : ContDiffAt 𝕜 2 u z) : ∀ᶠ x in 𝓝 z, DifferentiableAt 𝕜 u x

theorem **differentiableAt_dbar** (hu : ContDiffAt ℝ 2 u z) : DifferentiableAt ℝ (dbar u) z

theorem **differentiableAt_dbar_apply** (hu : ContDiffAt ℝ 2 u z) (w : E) : DifferentiableAt ℝ (fun x => dbar u x w) z

theorem **ddbar_apply** (hu : ContDiffAt ℝ 2 u z) (v w : E) : ddbar u z v w = (1 / 4 : ℂ) • (fderiv ℝ (fderiv ℝ u) z v w + fderiv ℝ (fderiv ℝ u) z (I • v) (I • w) + I • fderiv ℝ (fderiv ℝ u) z v (I • w) − I • fderiv ℝ (fderiv ℝ u) z (I • v) w)

theorem **fderiv_fderiv_apply** (hu : ContDiffAt ℝ 2 u z) (v w : E) : fderiv ℝ (fun x => fderiv ℝ u x w) z v = fderiv ℝ (fderiv ℝ u) z v w

theorem **ddbar_eq_del_dbar** (hu : ContDiffAt ℝ 2 u z) (v w : E) : ddbar u z v w = del (fun x => dbar u x w) z v

theorem **ddbar_eq_dbar_del** (hu : ContDiffAt ℝ 2 u z) (v w : E) : ddbar u z v w = dbar (fun x => del u x v) z w

theorem **ddbar_apply_self** (hu : ContDiffAt ℝ 2 u z) (v : E) : ddbar u z v v = (1 / 4 : ℂ) • (fderiv ℝ (fderiv ℝ u) z v v + fderiv ℝ (fderiv ℝ u) z (I • v) (I • v))

theorem **ddbar_congr** (h : u =ᶠ[𝓝 z] u') : ddbar u z = ddbar u' z

@[simp] theorem **ddbar_const** (c : F) (z : E) : ddbar (fun _ => c) z = 0

theorem **ddbar_add** (hu : ContDiffAt ℝ 2 u z) (hu' : ContDiffAt ℝ 2 u' z) : ddbar (fun x => u x + u' x) z = ddbar u z + ddbar u' z

theorem **ddbar_const_smul** (c : ℂ) (hu : ContDiffAt ℝ 2 u z) : ddbar (fun x => c • u x) z = c • ddbar u z

theorem **ddbar_neg** (u : E → F) (z : E) : ddbar (fun x => −u x) z = −ddbar u z

theorem **ddbar_sub** (hu : ContDiffAt ℝ 2 u z) (hu' : ContDiffAt ℝ 2 u' z) : ddbar (fun x => u x − u' x) z = ddbar u z − ddbar u' z

theorem **ddbar_sum** {ι : Type*} {t : Finset ι} {A : ι → E → F} (h : ∀ i ∈ t, ContDiffAt ℝ 2 (A i) z) : ddbar (fun x => ∑ i ∈ t, A i x) z = ∑ i ∈ t, ddbar (A i) z

theorem **ddbar_eq_zero_of_holomorphic** (hu : ∀ᶠ x in 𝓝 z, DifferentiableAt ℂ u x) : ddbar u z = 0

theorem **ddbar_conj** {a : E → ℂ} (ha : ContDiffAt ℝ 2 a z) (v w : E) : ddbar (fun x => conj (a x)) z v w = conj (ddbar a z w v)

theorem **ddbar_comp_holo** {u : F → G} {φ : E → F} (hu : ContDiffAt ℝ 2 u (φ z)) (hφ : ContDiffAt ℂ 2 φ z) (v w : E) : ddbar (fun x => u (φ x)) z v w = ddbar u (φ z) (fderiv ℂ φ z v) (fderiv ℂ φ z w)

theorem **contDiffAt_ofReal_comp** {n : WithTop ℕ∞} (hu : ContDiffAt ℝ n u z) : ContDiffAt ℝ n (fun x => (u x : ℂ)) z

theorem **leviForm_eq_del_dbar** (hu : ContDiffAt ℝ 2 u z) (v w : E) : leviForm u z v w = del (fun x => dbar (fun y => (u y : ℂ)) x w) z v

theorem **fderiv_fderiv_ofReal_apply** (hu : ContDiffAt ℝ 2 u z) (v w : E) : fderiv ℝ (fderiv ℝ fun x => (u x : ℂ)) z v w = (fderiv ℝ (fderiv ℝ u) z v w : ℂ)

theorem **leviForm_apply** (hu : ContDiffAt ℝ 2 u z) (v w : E) : leviForm u z v w = (1 / 4 : ℂ) * (fderiv ℝ (fderiv ℝ u) z v w + fderiv ℝ (fderiv ℝ u) z (I • v) (I • w) + I * fderiv ℝ (fderiv ℝ u) z v (I • w) − I * fderiv ℝ (fderiv ℝ u) z (I • v) w)

theorem **leviForm_apply_self** (hu : ContDiffAt ℝ 2 u z) (v : E) : leviForm u z v v = ((1 / 4 * (fderiv ℝ (fderiv ℝ u) z v v + fderiv ℝ (fderiv ℝ u) z (I • v) (I • v)) : ℝ) : ℂ)

theorem **leviForm_conj_symm** (hu : ContDiffAt ℝ 2 u z) (v w : E) : conj (leviForm u z v w) = leviForm u z w v

theorem **im_leviForm_apply_self** (hu : ContDiffAt ℝ 2 u z) (v : E) : (leviForm u z v v).im = 0

theorem **leviForm_congr** (h : u =ᶠ[𝓝 z] u') : leviForm u z = leviForm u' z

@[simp] theorem **leviForm_const** (c : ℝ) (z : E) : leviForm (fun _ => c) z = 0

theorem **leviForm_add** (hu : ContDiffAt ℝ 2 u z) (hu' : ContDiffAt ℝ 2 u' z) : leviForm (fun x => u x + u' x) z = leviForm u z + leviForm u' z

theorem **leviForm_sub** (hu : ContDiffAt ℝ 2 u z) (hu' : ContDiffAt ℝ 2 u' z) : leviForm (fun x => u x − u' x) z = leviForm u z − leviForm u' z

theorem **leviForm_neg** (u : E → ℝ) (z : E) : leviForm (fun x => −u x) z = −leviForm u z

theorem **leviForm_const_mul** (c : ℝ) (hu : ContDiffAt ℝ 2 u z) : leviForm (fun x => c * u x) z = (c : ℂ) • leviForm u z

theorem **leviForm_add_const** (c : ℝ) (hu : ContDiffAt ℝ 2 u z) : leviForm (fun x => u x + c) z = leviForm u z

theorem **leviForm_sum** {ι : Type*} {t : Finset ι} {A : ι → E → ℝ} (h : ∀ i ∈ t, ContDiffAt ℝ 2 (A i) z) : leviForm (fun x => ∑ i ∈ t, A i x) z = ∑ i ∈ t, leviForm (A i) z

theorem **leviForm_comp_holo** {u : F → ℝ} {φ : E → F} (hu : ContDiffAt ℝ 2 u (φ z)) (hφ : ContDiffAt ℂ 2 φ z) (v w : E) : leviForm (fun x => u (φ x)) z v w = leviForm u (φ z) (fderiv ℂ φ z v) (fderiv ℂ φ z w)

theorem **leviForm_re_eq_zero** {f : E → ℂ} (hf : ContDiffAt ℂ 2 f z) : leviForm (fun x => (f x).re) z = 0

theorem **leviForm_real_comp** {t : E → ℝ} {h : ℝ → ℝ} (hh : ContDiffAt ℝ 2 h (t z)) (ht : ContDiffAt ℝ 2 t z) (v w : E) : leviForm (fun x => h (t x)) z v w = ((deriv h (t z) : ℝ) : ℂ) * leviForm t z v w + ((deriv (deriv h) (t z) : ℝ) : ℂ) * (del (fun x => (t x : ℂ)) z v * dbar (fun x => (t x : ℂ)) z w)

@[simp] theorem **leviMatrix_apply** {ι : Type*} (u : E → ℝ) (z : E) (b : ι → E) (i j : ι) : leviMatrix u z b i j = leviForm u z (b i) (b j)

theorem **leviMatrix_isHermitian** {ι : Type*} (hu : ContDiffAt ℝ 2 u z) (b : ι → E) : (leviMatrix u z b).IsHermitian

theorem **inner_self_eq_ofReal_norm_sq** (v : H) : inner ℂ v v = ((‖v‖ ^ 2 : ℝ) : ℂ)

theorem **fderiv_normSq_apply** (z v : H) : fderiv ℝ (fun x : H => ((‖x‖ ^ 2 : ℝ) : ℂ)) z v = inner ℂ z v + inner ℂ v z

theorem **del_normSq** (z v : H) : del (fun x : H => ((‖x‖ ^ 2 : ℝ) : ℂ)) z v = inner ℂ z v

theorem **dbar_normSq** (z w : H) : dbar (fun x : H => ((‖x‖ ^ 2 : ℝ) : ℂ)) z w = inner ℂ w z

theorem **leviForm_normSq** (z v w : H) : leviForm (fun x : H => ‖x‖ ^ 2) z v w = inner ℂ w v

theorem **leviForm_comp_normSq** {h : ℝ → ℝ} {z : H} (hh : ContDiffAt ℝ 2 h (‖z‖ ^ 2)) (v w : H) : leviForm (fun x : H => h (‖x‖ ^ 2)) z v w = ((deriv h (‖z‖ ^ 2) : ℝ) : ℂ) * inner ℂ w v + ((deriv (deriv h) (‖z‖ ^ 2) : ℝ) : ℂ) * (inner ℂ z v * inner ℂ w z)

theorem **deriv_log_add_const** (δ : ℝ) : deriv (fun t : ℝ => Real.log (t + δ)) = fun t => (t + δ)⁻¹

theorem **deriv_deriv_log_add_const** (δ t : ℝ) : deriv (deriv fun t : ℝ => Real.log (t + δ)) t = −((t + δ) ^ 2)⁻¹

theorem **leviForm_log_normSq_add** {δ : ℝ} {z : H} (hz : ‖z‖ ^ 2 + δ ≠ 0) (v w : H) : leviForm (fun x : H => Real.log (‖x‖ ^ 2 + δ)) z v w = inner ℂ w v / ((‖z‖ ^ 2 + δ : ℝ) : ℂ) − inner ℂ z v * inner ℂ w z / ((‖z‖ ^ 2 + δ : ℝ) : ℂ) ^ 2

theorem **contDiffAt_log_normSq_add_comp** {δ : ℝ} (hφ : ContDiffAt ℂ 2 φ z) (hz : ‖φ z‖ ^ 2 + δ ≠ 0) : ContDiffAt ℝ 2 (fun x => Real.log (‖φ x‖ ^ 2 + δ)) z

theorem **leviForm_log_normSq_add_comp** {δ : ℝ} (hφ : ContDiffAt ℂ 2 φ z) (hz : ‖φ z‖ ^ 2 + δ ≠ 0) (v w : E) : leviForm (fun x => Real.log (‖φ x‖ ^ 2 + δ)) z v w = inner ℂ (fderiv ℂ φ z w) (fderiv ℂ φ z v) / ((‖φ z‖ ^ 2 + δ : ℝ) : ℂ) - inner ℂ (φ z) (fderiv ℂ φ z v) * inner ℂ (fderiv ℂ φ z w) (φ z) / ((‖φ z‖ ^ 2 + δ : ℝ) : ℂ) ^ 2

theorem **leviForm_log_normSq_comp** (hφ : ContDiffAt ℂ 2 φ z) (hz : φ z ≠ 0) (v w : E) : leviForm (fun x => Real.log (‖φ x‖ ^ 2)) z v w = inner ℂ (fderiv ℂ φ z w) (fderiv ℂ φ z v) / (‖φ z‖ : ℂ) ^ 2 - inner ℂ (φ z) (fderiv ℂ φ z v) * inner ℂ (fderiv ℂ φ z w) (φ z) / (‖φ z‖ : ℂ) ^ 4

theorem **leviForm_log_normSq_eq_zero** {f : E → ℂ} (hf : ContDiffAt ℂ 2 f z) (hz : f z ≠ 0) : leviForm (fun x => Real.log (‖f x‖ ^ 2)) z = 0

theorem **IsPshC2On.mono** (h : IsPshC2On u t) (hst : s ⊆ t) : IsPshC2On u s

theorem **IsPshC2On.contDiffAt** (h : IsPshC2On u s) {z : E} (hz : z ∈ s) : ContDiffAt ℝ 2 u z

theorem **IsPshC2On.nonneg** (h : IsPshC2On u s) {z : E} (hz : z ∈ s) (v : E) : 0 ≤ (leviForm u z v v).re

theorem **IsPshC2On.add** (h : IsPshC2On u s) (h' : IsPshC2On u' s) : IsPshC2On (fun x => u x + u' x) s

theorem **IsPshC2On.const_mul** (h : IsPshC2On u s) {c : ℝ} (hc : 0 ≤ c) : IsPshC2On (fun x => c * u x) s

theorem **isPshC2On_normSq** (s : Set H) : IsPshC2On (fun x : H => ‖x‖ ^ 2) s

theorem **leviForm_one_one_eq_laplacian** {u : ℂ → ℝ} {z : ℂ} (hu : ContDiffAt ℝ 2 u z) : leviForm u z 1 1 = ((Δ u z / 4 : ℝ) : ℂ)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsAlgebraicIndependentGenericPoint

theorem **ringKrullDim_eq_of_isIntegral** (R S : Type*) [CommRing R] [CommRing S] [IsDomain S] [Algebra R S] [FaithfulSMul R S] [Algebra.IsIntegral R S] : ringKrullDim S = ringKrullDim R

theorem **exists_ringKrullDim_eq_and_trdeg_eq** (K A : Type u) [Field K] [CommRing A] [IsDomain A] [Algebra K A] [Algebra.FiniteType K A] : ∃ s : ℕ, ringKrullDim A = s ∧ Algebra.trdeg K A = s

theorem **isPrincipal_of_forall_C_mem** {R : Type*} [CommRing R] [IsDomain R] [UniqueFactorizationMonoid R] (E : Ideal (Polynomial R)) [hE : E.IsPrime] (h : ∀ r, Polynomial.C r ∈ E → r = 0) : E.IsPrincipal

theorem **isPrincipal_of_forall_rename_mem** {K ι ι' : Type*} [Field K] [Finite ι'] (ε : ι ≃ Option ι') (E : Ideal (MvPolynomial ι K)) [E.IsPrime] (h : ∀ g : MvPolynomial ι' K, rename (fun v => ε.symm (some v)) g ∈ E → g = 0) : E.IsPrincipal

theorem **injective_of_constantCoeff** {R S V : Type*} [CommRing R] [CommRing S] (ψ : MvPolynomial V R →+* MvPolynomial V S) (hX : ∀ v, ψ (X v) = X v) (h : ∀ r, constantCoeff (ψ (C r)) = 0 → r = 0) : Function.Injective ψ

theorem **injective_of_eval** {R S V : Type*} [CommRing R] [CommRing S] (ψ : MvPolynomial V R →+* MvPolynomial V S) (hX : ∀ v, ψ (X v) = X v) (c : V → R) (c' : V → S) (hc : ∀ v, ψ (C (c v)) = C (c' v)) (h : ∀ r, eval c' (ψ (C r)) = 0 → r = 0) : Function.Injective ψ

theorem **algebraicIndependent_div** {K F β : Type*} [Field K] [Field F] [Algebra K F] {z : β → F} (hz : AlgebraicIndependent K z) (i₀ : β) : AlgebraicIndependent K fun i : {i // i ≠ i₀} => z i / z i₀

theorem **elimIdeal_ne_bot** {p : Ideal (MvPolynomial σ K)} [hP : p.IsPrime] (hp : p.IsHomogeneous (homogeneousSubmodule σ K)) (hX : ∃ j, X j ∉ p) (hdim : ringKrullDim (MvPolynomial σ K ⧸ p) ≤ (Fintype.card β : ℕ)) : elimIdeal β p ≠ ⊥

theorem **exists_algebraicIndependent_genericPoint** [Finite σ] {p : Ideal (MvPolynomial σ K)} [p.IsPrime] (hdim : ringKrullDim (MvPolynomial σ K ⧸ p) = (Fintype.card β : ℕ)) : ∃ κ : β → σ, AlgebraicIndependent K fun i => genericPoint p (κ i)

theorem **isPrincipal_elimIdeal** [Nonempty β] {p : Ideal (MvPolynomial σ K)} [p.IsPrime] (hdim : ringKrullDim (MvPolynomial σ K ⧸ p) = (Fintype.card β : ℕ)) : (elimIdeal β p).IsPrincipal

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsEventuallyNormRiemannThetaTermLe

theorem **isSiegel_iff_mem_siegelH** [Fintype ι] {Ω : Matrix ι ι ℂ} : IsSiegel Ω ↔ Ω ∈ ReciprocityTheory.PTheta.SiegelH ι

theorem **IsSiegel.mk** {Ω : Matrix ι ι ℂ} (isSymm : Ω.IsSymm) (posDef : (Ω.map Complex.im).PosDef) : IsSiegel Ω

theorem **IsSiegel.isSymm** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) : Ω.IsSymm

theorem **IsSiegel.posDef** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) : (Ω.map Complex.im).PosDef

theorem **IsSiegel.mem_siegelH** [Fintype ι] {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) : Ω ∈ ReciprocityTheory.PTheta.SiegelH ι

theorem **IsSiegel.of_mem_siegelH** [Fintype ι] {Ω : Matrix ι ι ℂ} (h : Ω ∈ ReciprocityTheory.PTheta.SiegelH ι) : IsSiegel Ω

theorem **siegelUpperHalfSpace_eq_siegelH** (ι : Type*) [Fintype ι] : siegelUpperHalfSpace ι = ReciprocityTheory.PTheta.SiegelH ι

@[simp] theorem **mem_siegelUpperHalfSpace** {Ω : Matrix ι ι ℂ} : Ω ∈ siegelUpperHalfSpace ι ↔ IsSiegel Ω

theorem **isHermitian_map_im_of_isSymm** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) : (Ω.map Complex.im).IsHermitian

theorem **IsSiegel.of_dotProduct_pos** [Fintype ι] {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (hpos : ∀ x : ι → ℝ, x ≠ 0 → 0 < x ⬝ᵥ (Ω.map Complex.im *ᵥ x)) : IsSiegel Ω

theorem **IsSiegel.dotProduct_pos** [Fintype ι] {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) {x : ι → ℝ} (hx : x ≠ 0) : 0 < x ⬝ᵥ (Ω.map Complex.im *ᵥ x)

theorem **isSiegel_ofReal_add_I_smul** {X Y : Matrix ι ι ℝ} (hX : X.IsSymm) (hY : Y.PosDef) : IsSiegel (X.map ofReal + I • Y.map ofReal)

theorem **isSiegel_diagonal_iff** [DecidableEq ι] {τ : ι → ℂ} : IsSiegel (diagonal τ) ↔ ∀ i, 0 < (τ i).im

theorem **isSiegel_I_smul_one** [DecidableEq ι] : IsSiegel (I • (1 : Matrix ι ι ℂ))

theorem **isSiegel_unique_iff** [Unique ι] {Ω : Matrix ι ι ℂ} : IsSiegel Ω ↔ 0 < (Ω default default).im

theorem **IsSiegel.smul** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) {q : ℝ} (hq : 0 < q) : IsSiegel ((q : ℂ) • Ω)

theorem **IsSiegel.natCast_smul** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) {q : ℕ} (hq : 0 < q) : IsSiegel ((q : ℂ) • Ω)

theorem **IsSiegel.submatrix** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) {e : κ → ι} (he : Function.Injective e) : IsSiegel (Ω.submatrix e e)

theorem **isSiegel_fromBlocks_iff** [Finite ι] [Finite κ] {Ω₁ : Matrix ι ι ℂ} {Ω₂ : Matrix κ κ ℂ} : IsSiegel (fromBlocks Ω₁ 0 0 Ω₂) ↔ IsSiegel Ω₁ ∧ IsSiegel Ω₂

theorem **isSiegel_example** : IsSiegel !![1 + 2 * I, 1 / 2 + I; 1 / 2 + I, 2 * I]

theorem **posDef_exists_mul_dotProduct_le** {Y : Matrix ι ι ℝ} (hY : Y.PosDef) : ∃ c > 0, ∀ x : ι → ℝ, c * (x ⬝ᵥ x) ≤ x ⬝ᵥ (Y *ᵥ x)

theorem **IsSiegel.exists_eventually_mul_dotProduct_le** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) : ∃ c > 0, ∀ᶠ Ω' in 𝒩 Ω, ∀ x : ι → ℝ, c * (x ⬝ᵥ x) ≤ x ⬝ᵥ (Matrix.map Ω' Complex.im *ᵥ x)

theorem **IsSiegel.eventually_isSiegel** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) : ∀ᶠ Ω' in 𝒩 Ω, Matrix.IsSymm Ω' → IsSiegel Ω'

theorem **riemannThetaTerm_eq_thetaTerm** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω v n = ReciprocityTheory.PTheta.thetaTerm a b v Ω n

theorem **riemannThetaTerm_def** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω v n = cexp (2 * π * I * (((fun i => (n i : ℂ) + a i) ⬝ᵥ (Ω *ᵥ fun i => (n i : ℂ) + a i)) / 2 + (fun i => (n i : ℂ) + a i) ⬝ᵥ fun i => v i + b i))

theorem **riemannTheta_eq_theta** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω v = ReciprocityTheory.PTheta.theta a b v Ω

theorem **riemannTheta_def** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω v = ∑' n : ι → ℤ, riemannThetaTerm a b Ω v n

theorem **riemannThetaTerm_eq_sum** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω v n = cexp (π * I * (∑ i, ∑ j, ((n i : ℂ) + a i) * Ω i j * ((n j : ℂ) + a j)) + 2 * π * I * ∑ i, ((n i : ℂ) + a i) * (v i + b i))

theorem **riemannThetaTerm_zero_zero** (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm 0 0 Ω v n = cexp (2 * π * I * (((fun i => (n i : ℂ)) ⬝ᵥ (Ω *ᵥ fun i => (n i : ℂ))) / 2 + (fun i => (n i : ℂ)) ⬝ᵥ v))

theorem **riemannThetaTerm_ne_zero** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω v n ≠ 0

theorem **norm_riemannThetaTerm** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : ‖riemannThetaTerm a b Ω v n‖ = Real.exp (-π * ((fun i => (n i : ℝ) + a i) ⬝ᵥ (Ω.map Complex.im *ᵥ fun i => (n i : ℝ) + a i)) - 2 * π * ((fun i => (n i : ℝ) + a i) ⬝ᵥ fun i => (v i).im))

theorem **riemannThetaBound_pos** (a : ι → ℝ) (c S : ℝ) (n : ι → ℤ) : 0 < riemannThetaBound a c S n

theorem **norm_riemannThetaTerm_le** (a b : ι → ℝ) {Ω : Matrix ι ι ℂ} {v : ι → ℂ} {c S : ℝ} (hΩ : ∀ x : ι → ℝ, c * (x ⬝ᵥ x) ≤ x ⬝ᵥ (Ω.map Complex.im *ᵥ x)) (hv : ∀ i, |(v i).im| ≤ S) (n : ι → ℤ) : ‖riemannThetaTerm a b Ω v n‖ ≤ riemannThetaBound a c S n

theorem **summable_pi_prod_of_nonneg** {α : ι → Type*} {f : ∀ i, α i → ℝ} (h0 : ∀ i x, 0 ≤ f i x) (hf : ∀ i, Summable (f i)) : Summable fun n : ∀ i, α i => ∏ i, f i (n i)

theorem **summable_exp_neg_shifted_sq** (a S : ℝ) {c : ℝ} (hc : 0 < c) : Summable fun n : ℤ => Real.exp (-π * (c * ((n + a) * (n + a)) - 2 * S * |n + a|))

theorem **summable_riemannThetaBound** (a : ι → ℝ) {c : ℝ} (hc : 0 < c) (S : ℝ) : Summable (riemannThetaBound a c S)

theorem **summable_pow_mul_riemannThetaBound** (a : ι → ℝ) {c : ℝ} (hc : 0 < c) (S : ℝ) (k : ℕ) : Summable fun n : ι → ℤ => (∑ i, |(n i : ℝ) + a i|) ^ k * riemannThetaBound a c S n

theorem **abs_im_apply_le_norm** (v : ι → ℂ) (i : ι) : |(v i).im| ≤ ‖v‖

theorem **IsSiegel.summable_norm_riemannThetaTerm** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : Summable fun n => ‖riemannThetaTerm a b Ω v n‖

theorem **IsSiegel.summable_riemannThetaTerm** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : Summable (riemannThetaTerm a b Ω v)

theorem **IsSiegel.hasSum_riemannTheta** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : HasSum (riemannThetaTerm a b Ω v) (riemannTheta a b Ω v)

theorem **IsSiegel.exists_eventually_norm_riemannThetaTerm_le** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : ∃ c > 0, ∃ S, ∀ᶠ p : (ι → ℂ) × Matrix ι ι ℂ in 𝒩 (v, Ω), ∀ n, ‖riemannThetaTerm a b p.2 p.1 n‖ ≤ riemannThetaBound a c S n

theorem **hasSumLocallyUniformlyOn_riemannTheta** (a b : ι → ℝ) : HasSumLocallyUniformlyOn (fun n (p : (ι → ℂ) × Matrix ι ι ℂ) => riemannThetaTerm a b p.2 p.1 n) (fun p => riemannTheta a b p.2 p.1) (Set.univ ×ˢ ReciprocityTheory.PTheta.SiegelH ι)

theorem **IsSiegel.hasSumLocallyUniformly_riemannTheta** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) : HasSumLocallyUniformly (fun n v => riemannThetaTerm a b Ω v n) (riemannTheta a b Ω)

theorem **continuous_riemannThetaTerm** (a b : ι → ℝ) (n : ι → ℤ) : Continuous fun p : (ι → ℂ) × Matrix ι ι ℂ => riemannThetaTerm a b p.2 p.1 n

theorem **continuousOn_riemannTheta** (a b : ι → ℝ) : ContinuousOn (fun p : (ι → ℂ) × Matrix ι ι ℂ => riemannTheta a b p.2 p.1) (Set.univ ×ˢ ReciprocityTheory.PTheta.SiegelH ι)

@[simp] theorem **riemannThetaForm_apply** (a : ι → ℝ) (n : ι → ℤ) (w : ι → ℂ) : riemannThetaForm a n w = 2 * π * I * ((fun i => (n i : ℂ) + a i) ⬝ᵥ w)

theorem **norm_riemannThetaForm_le** (a : ι → ℝ) (n : ι → ℤ) : ‖riemannThetaForm a n‖ ≤ 2 * π * ∑ i, |(n i : ℝ) + a i|

theorem **hasFDerivAt_riemannThetaTerm** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : HasFDerivAt (fun w => riemannThetaTerm a b Ω w n) (riemannThetaTerm a b Ω v n • riemannThetaForm a n) v

theorem **IsSiegel.summable_norm_riemannThetaTerm_smul_form** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : Summable fun n => ‖riemannThetaTerm a b Ω v n • riemannThetaForm a n‖

theorem **IsSiegel.hasFDerivAt_riemannTheta** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : HasFDerivAt (riemannTheta a b Ω) (∑' n, riemannThetaTerm a b Ω v n • riemannThetaForm a n) v

theorem **IsSiegel.differentiable_riemannTheta** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) : Differentiable ℂ (riemannTheta a b Ω)

theorem **IsSiegel.continuous_riemannTheta** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) : Continuous (riemannTheta a b Ω)

theorem **cexp_two_pi_I_dotProduct_intCast** (n m : ι → ℤ) : cexp (2 * π * I * ((fun i => (n i : ℂ)) ⬝ᵥ fun i => (m i : ℂ))) = 1

theorem **dotProduct_mulVec_comm_of_isSymm** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (u w : ι → ℂ) : u ⬝ᵥ (Ω *ᵥ w) = w ⬝ᵥ (Ω *ᵥ u)

theorem **riemannThetaTerm_add_ofReal** (a b x : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω (v + fun i => (x i : ℂ)) n = riemannThetaTerm a (b + x) Ω v n

theorem **riemannTheta_add_ofReal** (a b x : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω (v + fun i => (x i : ℂ)) = riemannTheta a (b + x) Ω v

theorem **riemannThetaTerm_add** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v w : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω (v + w) n = riemannThetaTerm a b Ω v n * cexp (2 * π * I * ((fun i => (n i : ℂ) + a i) ⬝ᵥ w))

theorem **riemannThetaTerm_add_intCast** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (m n : ι → ℤ) : riemannThetaTerm a b Ω (v + fun i => (m i : ℂ)) n = cexp (2 * π * I * ((fun i => (a i : ℂ)) ⬝ᵥ fun i => (m i : ℂ))) * riemannThetaTerm a b Ω v n

theorem **riemannTheta_add_intCast** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (m : ι → ℤ) : riemannTheta a b Ω (v + fun i => (m i : ℂ)) = cexp (2 * π * I * ((fun i => (a i : ℂ)) ⬝ᵥ fun i => (m i : ℂ))) * riemannTheta a b Ω v

theorem **riemannThetaTerm_add_mulVec_ofReal** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b y : ι → ℝ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω (v + Ω *ᵥ fun i => (y i : ℂ)) n = cexp (-(2 * π * I) * (((fun i => (y i : ℂ)) ⬝ᵥ (Ω *ᵥ fun i => (y i : ℂ))) / 2 + (fun i => (y i : ℂ)) ⬝ᵥ fun i => v i + b i)) * riemannThetaTerm (a + y) b Ω v n

theorem **riemannTheta_add_mulVec_ofReal** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b y : ι → ℝ) (v : ι → ℂ) : riemannTheta a b Ω (v + Ω *ᵥ fun i => (y i : ℂ)) = cexp (-(2 * π * I) * (((fun i => (y i : ℂ)) ⬝ᵥ (Ω *ᵥ fun i => (y i : ℂ))) / 2 + (fun i => (y i : ℂ)) ⬝ᵥ fun i => v i + b i)) * riemannTheta (a + y) b Ω v

theorem **riemannThetaTerm_char_add_intCast** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (k n : ι → ℤ) : riemannThetaTerm (a + fun i => (k i : ℝ)) b Ω v n = riemannThetaTerm a b Ω v (n + k)

theorem **riemannTheta_char_add_intCast_left** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (k : ι → ℤ) : riemannTheta (a + fun i => (k i : ℝ)) b Ω v = riemannTheta a b Ω v

theorem **riemannTheta_char_add_intCast_right** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (m : ι → ℤ) : riemannTheta a (b + fun i => (m i : ℝ)) Ω v = cexp (2 * π * I * ((fun i => (a i : ℂ)) ⬝ᵥ fun i => (m i : ℂ))) * riemannTheta a b Ω v

theorem **riemannTheta_add_mulVec_intCast** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b : ι → ℝ) (v : ι → ℂ) (k : ι → ℤ) : riemannTheta a b Ω (v + Ω *ᵥ fun i => (k i : ℂ)) = cexp (-(2 * π * I) * (((fun i => (k i : ℂ)) ⬝ᵥ (Ω *ᵥ fun i => (k i : ℂ))) / 2 + (fun i => (k i : ℂ)) ⬝ᵥ fun i => v i + b i)) * riemannTheta a b Ω v

theorem **riemannTheta_add_lattice** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b : ι → ℝ) (v : ι → ℂ) (m k : ι → ℤ) : riemannTheta a b Ω (v + ((fun i => (m i : ℂ)) + Ω *ᵥ fun i => (k i : ℂ))) = cexp (2 * π * I * (((fun i => (a i : ℂ)) ⬝ᵥ fun i => (m i : ℂ)) - ((fun i => (b i : ℂ)) ⬝ᵥ fun i => (k i : ℂ)) - ((fun i => (k i : ℂ)) ⬝ᵥ (Ω *ᵥ fun i => (k i : ℂ))) / 2 - (fun i => (k i : ℂ)) ⬝ᵥ v)) * riemannTheta a b Ω v

theorem **riemannTheta_eq_cexp_mul_zero_zero** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b : ι → ℝ) (v : ι → ℂ) : riemannTheta a b Ω v = cexp (2 * π * I * (((fun i => (a i : ℂ)) ⬝ᵥ (Ω *ᵥ fun i => (a i : ℂ))) / 2 + (fun i => (a i : ℂ)) ⬝ᵥ fun i => v i + b i)) * riemannTheta 0 0 Ω (v + (fun i => (b i : ℂ)) + Ω *ᵥ fun i => (a i : ℂ))

theorem **riemannThetaTerm_neg** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω (-v) (-n) = riemannThetaTerm (-a) (-b) Ω v n

theorem **riemannTheta_neg** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω (-v) = riemannTheta (-a) (-b) Ω v

theorem **riemannTheta_zero_zero_neg** (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta 0 0 Ω (-v) = riemannTheta 0 0 Ω v

theorem **riemannTheta_half_neg** (α β : ι → ℤ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta (fun i => (α i : ℝ) / 2) (fun i => (β i : ℝ) / 2) Ω (-v) = (-1) ^ (α ⬝ᵥ β) * riemannTheta (fun i => (α i : ℝ) / 2) (fun i => (β i : ℝ) / 2) Ω v

theorem **integral_Icc_cexp_dotProduct** [DecidableEq ι] (m : ι → ℤ) : ∫ x in Set.Icc (0 : ι → ℝ) 1, cexp (2 * π * I * ((fun i => (m i : ℂ)) ⬝ᵥ fun i => (x i : ℂ))) = if m = 0 then 1 else 0

theorem **integral_Icc_tsum_mul_cexp** {c : (ι → ℤ) → ℂ} (hc : Summable fun m => ‖c m‖) (n : ι → ℤ) : ∫ x in Set.Icc (0 : ι → ℝ) 1, (∑' m, c m * cexp (2 * π * I * ((fun i => (m i : ℂ)) ⬝ᵥ fun i => (x i : ℂ)))) * cexp (-(2 * π * I) * ((fun i => (n i : ℂ)) ⬝ᵥ fun i => (x i : ℂ))) = c n

theorem **riemannTheta_add_ofReal_eq_tsum** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (x : ι → ℝ) : riemannTheta a b Ω (v + fun i => (x i : ℂ)) = cexp (2 * π * I * ((fun i => (a i : ℂ)) ⬝ᵥ fun i => (x i : ℂ))) * ∑' m, riemannThetaTerm a b Ω v m * cexp (2 * π * I * ((fun i => (m i : ℂ)) ⬝ᵥ fun i => (x i : ℂ)))

theorem **IsSiegel.integral_riemannTheta_mul_cexp** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) (n : ι → ℤ) : ∫ x in Set.Icc (0 : ι → ℝ) 1, riemannTheta a b Ω (v + fun i => (x i : ℂ)) * cexp (-(2 * π * I) * ((fun i => (n i : ℂ) + a i) ⬝ᵥ fun i => (x i : ℂ))) = riemannThetaTerm a b Ω v n

theorem **IsSiegel.exists_riemannTheta_add_ofReal_ne_zero** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : ∃ x ∈ Set.Icc (0 : ι → ℝ) 1, riemannTheta a b Ω (v + fun i => (x i : ℂ)) ≠ 0

theorem **IsSiegel.riemannTheta_ne_zero** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) : riemannTheta a b Ω ≠ 0

theorem **riemannTheta_unique_eq_tsum** [Unique ι] (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω v = ∑' n : ℤ, cexp (π * I * (n + a default) ^ 2 * Ω default default + 2 * π * I * (n + a default) * (v default + b default))

theorem **riemannTheta_unique_zero_zero** [Unique ι] (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta 0 0 Ω v = jacobiTheta₂ (v default) (Ω default default)

theorem **riemannTheta_unique_eq_jacobiTheta₂** [Unique ι] (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω v = cexp (π * I * (a default) ^ 2 * Ω default default + 2 * π * I * a default * (v default + b default)) * jacobiTheta₂ (v default + b default + a default * Ω default default) (Ω default default)

theorem **riemannTheta_fin_one** (z τ : ℂ) : riemannTheta 0 0 !![τ] ![z] = jacobiTheta₂ z τ

theorem **riemannTheta_submatrix_equiv** (e : κ ≃ ι) (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta (a ∘ e) (b ∘ e) (Ω.submatrix e e) (v ∘ e) = riemannTheta a b Ω v

theorem **riemannThetaTerm_fromBlocks** (a₁ b₁ : ι → ℝ) (a₂ b₂ : κ → ℝ) (Ω₁ : Matrix ι ι ℂ) (Ω₂ : Matrix κ κ ℂ) (v₁ : ι → ℂ) (v₂ : κ → ℂ) (n₁ : ι → ℤ) (n₂ : κ → ℤ) : riemannThetaTerm (Sum.elim a₁ a₂) (Sum.elim b₁ b₂) (fromBlocks Ω₁ 0 0 Ω₂) (Sum.elim v₁ v₂) (Sum.elim n₁ n₂) = riemannThetaTerm a₁ b₁ Ω₁ v₁ n₁ * riemannThetaTerm a₂ b₂ Ω₂ v₂ n₂

theorem **riemannTheta_fromBlocks** {Ω₁ : Matrix ι ι ℂ} {Ω₂ : Matrix κ κ ℂ} (h₁ : IsSiegel Ω₁) (h₂ : IsSiegel Ω₂) (a₁ b₁ : ι → ℝ) (a₂ b₂ : κ → ℝ) (v₁ : ι → ℂ) (v₂ : κ → ℂ) : riemannTheta (Sum.elim a₁ a₂) (Sum.elim b₁ b₂) (fromBlocks Ω₁ 0 0 Ω₂) (Sum.elim v₁ v₂) = riemannTheta a₁ b₁ Ω₁ v₁ * riemannTheta a₂ b₂ Ω₂ v₂

theorem **riemannTheta_fromBlocks'** {Ω₁ : Matrix ι ι ℂ} {Ω₂ : Matrix κ κ ℂ} (h₁ : IsSiegel Ω₁) (h₂ : IsSiegel Ω₂) (a b : ι ⊕ κ → ℝ) (v : ι ⊕ κ → ℂ) : riemannTheta a b (fromBlocks Ω₁ 0 0 Ω₂) v = riemannTheta (a ∘ Sum.inl) (b ∘ Sum.inl) Ω₁ (v ∘ Sum.inl) * riemannTheta (a ∘ Sum.inr) (b ∘ Sum.inr) Ω₂ (v ∘ Sum.inr)

theorem **summable_norm_thetaTerm_of_mem_siegelH** {τ : Matrix ι ι ℂ} (h : τ ∈ SiegelH ι) (a b : ι → ℝ) (z : ι → ℂ) : Summable fun l => ‖thetaTerm a b z τ l‖

theorem **hasSum_theta_of_mem_siegelH** {τ : Matrix ι ι ℂ} (h : τ ∈ SiegelH ι) (a b : ι → ℝ) (z : ι → ℂ) : HasSum (thetaTerm a b z τ) (theta a b z τ)

theorem **hasSumLocallyUniformlyOn_theta_siegelH** (a b : ι → ℝ) : HasSumLocallyUniformlyOn (fun l (p : (ι → ℂ) × Matrix ι ι ℂ) => thetaTerm a b p.1 p.2 l) (fun p => theta a b p.1 p.2) (Set.univ ×ˢ SiegelH ι)

theorem **differentiable_theta_of_mem_siegelH** {τ : Matrix ι ι ℂ} (h : τ ∈ SiegelH ι) (a b : ι → ℝ) : Differentiable ℂ fun z => theta a b z τ

theorem **exists_theta_ne_zero_of_mem_siegelH** {τ : Matrix ι ι ℂ} (h : τ ∈ SiegelH ι) (a b : ι → ℝ) : ∃ z, theta a b z τ ≠ 0

theorem **integral_theta_mul_eC_of_mem_siegelH** {τ : Matrix ι ι ℂ} (h : τ ∈ SiegelH ι) (a b : ι → ℝ) (z : ι → ℂ) (l : ι → ℤ) : ∫ x in Set.Icc (0 : ι → ℝ) 1, theta a b (z + ofRealVec x) τ * eC (-(shiftVec a l ·ᵥ ofRealVec x)) = thetaTerm a b z τ l

theorem **riemannTheta_add_eq_autFactor_mul** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b : ι → ℝ) (v : ι → ℂ) (k m : ι → ℤ) : riemannTheta a b Ω (v + Ω *ᵥ ofIntVec k + ofIntVec m) = autFactor a b Ω k m v * riemannTheta a b Ω v

theorem **theta_fromBlocks_zero** {τ₁ : Matrix ι ι ℂ} {τ₂ : Matrix κ κ ℂ} (h₁ : τ₁ ∈ SiegelH ι) (h₂ : τ₂ ∈ SiegelH κ) (a₁ b₁ : ι → ℝ) (a₂ b₂ : κ → ℝ) (z₁ : ι → ℂ) (z₂ : κ → ℂ) : theta (Sum.elim a₁ a₂) (Sum.elim b₁ b₂) (Sum.elim z₁ z₂) (fromBlocks τ₁ 0 0 τ₂) = theta a₁ b₁ z₁ τ₁ * theta a₂ b₂ z₂ τ₂

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsFormOfEqRatioPowMul

theorem **exists_formOn** (i : Fin (N + 1)) (u : Γ(PSpace R N, chart R N i)) : ∃ n f, ∃ hf : (f : MvPolynomial (Fin (N + 1)) R).IsHomogeneous n, u = formOn (chart R N i) i le_rfl n f hf

theorem **resG_overlap_injective** (i j : Fin (N + 1)) {U : (PSpace R N).Opens} (hi : U ≤ chart R N i) (hU : U = chart R N i ⊓ chart R N j) : Function.Injective (resG (PSpace R N) hi)

theorem **eq_zero_of_mul_ratio_pow** {i j : Fin (N + 1)} (hij : j ≠ i) (k : ℕ) {U : (PSpace R N).Opens} (hi : U ≤ chart R N i) (hj : U ≤ chart R N j) (hU : U = chart R N i ⊓ chart R N j) (u : Γ(PSpace R N, chart R N i)) (v : Γ(PSpace R N, chart R N j)) (h : resG _ hi u * ratioOn U i hi j ^ (k + 1) = resG _ hj v) : u = 0

theorem **sec_neg_eq_zero** (k : ℕ) (s : (On R (N + 1) (-((k + 1 : ℕ) : ℤ))).Sec) (i : PUnit → Fin (N + 2)) : s.f i = 0

theorem **exists_form_of_eq_ratio_pow_mul** {i j : Fin (N + 1)} (hij : j ≠ i) (n : ℕ) {U : (PSpace R N).Opens} (hi : U ≤ chart R N i) (hj : U ≤ chart R N j) (hU : U = chart R N i ⊓ chart R N j) (u : Γ(PSpace R N, chart R N i)) (v : Γ(PSpace R N, chart R N j)) (h : resG _ hi u = ratioOn U i hi j ^ n * resG _ hj v) : ∃ f, ∃ hf : (f : MvPolynomial (Fin (N + 1)) R).IsHomogeneous n, u = formOn (chart R N i) i le_rfl n f hf

theorem **exists_formSec** (n : ℕ) (s : (lbPow (O1 R (N + 1)) n).Sec) : ∃ f, ∃ hf : (f : MvPolynomial (Fin (N + 2)) R).IsHomogeneous n, s = formSec n f hf

theorem **not_isGen_O1** [Nontrivial R] (t : (O1 R (N + 1)).Gen) : ¬ t.IsGen

theorem **exists_isGen_O1_zero** : ∃ t : (O1 R 0).Gen, t.IsGen

theorem **isIso_str_zero** : IsIso (PSpace.str R 0)

theorem **nilpotent_form** (h : (C 2 * X 0 : MvPolynomial (Fin 2) (ZMod 4)).IsHomogeneous 1) : formSec 1 _ h ≠ formSec 1 0 (isHomogeneous_zero _ _ 1) ∧ nonzeroLocus (formSec 1 _ h) = ⊥

theorem **formSec_X_f** (k : Fin (N + 1)) (j : PUnit → Fin (N + 1)) : (formSec 1 (X k) (isHomogeneous_X R k)).f j = resG _ (On_U R N 1 j).le ((coord R N k).f (j PUnit.unit))

theorem **isEmpty_of_subsingleton** [Subsingleton R] : IsEmpty (PSpace R N)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsLinearIndependentNormLeSuccMin

theorem **natCast_le_rank_span_iff** {T : Set V} {i : ℕ} : (i : Cardinal) ≤ Module.rank K (span K T) ↔ ∃ v : Fin i → V, LinearIndependent K v ∧ ∀ j, v j ∈ T

theorem **mem_succMinRadii_iff** : r ∈ succMinRadii K N S i ↔ 0 ≤ r ∧ ∃ v : Fin i → V, LinearIndependent K v ∧ ∀ j, v j ∈ S ∧ N (v j) ≤ r

theorem **bddBelow_succMinRadii** : BddBelow (succMinRadii K N S i)

theorem **succMinRadii_upper** {r' : ℝ} (h : r ∈ succMinRadii K N S i) (hr : r ≤ r') : r' ∈ succMinRadii K N S i

theorem **succMinRadii_nonempty_iff** : (succMinRadii K N S i).Nonempty ↔ (i : Cardinal) ≤ Module.rank K (span K S)

theorem **succMinOf_nonneg** : 0 ≤ succMinOf K N S i

@[simp] theorem **succMinOf_zero** : succMinOf K N S 0 = 0

theorem **succMinOf_le_of_mem** (h : r ∈ succMinRadii K N S i) : succMinOf K N S i ≤ r

theorem **succMinOf_le_of_linearIndependent** {ι : Type*} [Fintype ι] {v : ι → V} (hv : LinearIndependent K v) (hi : i ≤ Fintype.card ι) (hr : 0 ≤ r) (hS : ∀ k, v k ∈ S) (hN : ∀ k, N (v k) ≤ r) : succMinOf K N S i ≤ r

theorem **mem_succMinRadii_of_lt** (hi : (i : Cardinal) ≤ Module.rank K (span K S)) (h : succMinOf K N S i < r) : r ∈ succMinRadii K N S i

theorem **le_succMinOf** (hi : (i : Cardinal) ≤ Module.rank K (span K S)) {a : ℝ} (h : ∀ r ∈ succMinRadii K N S i, a ≤ r) : a ≤ succMinOf K N S i

theorem **succMinOf_mono** (hij : i ≤ j) (hj : (j : Cardinal) ≤ Module.rank K (span K S)) : succMinOf K N S i ≤ succMinOf K N S j

theorem **succMinRadii_image** {W : Type*} [AddCommGroup W] [Module K W] {M : W → ℝ} (f : V →ₗ[K] W) (hf : Function.Injective f) (hM : ∀ x ∈ S, M (f x) = N x) : succMinRadii K M (f '' S) i = succMinRadii K N S i

theorem **succMinOf_image** {W : Type*} [AddCommGroup W] [Module K W] {M : W → ℝ} (f : V →ₗ[K] W) (hf : Function.Injective f) (hM : ∀ x ∈ S, M (f x) = N x) : succMinOf K M (f '' S) i = succMinOf K N S i

theorem **succMinOf_const_mul** {c : ℝ} (hc : 0 < c) : succMinOf K (fun x => c * N x) S i = c * succMinOf K N S i

theorem **succMinOf_mem** (hfin : ∀ R, {x | x ∈ S ∧ N x ≤ R}.Finite) (hi : (i : Cardinal) ≤ Module.rank K (span K S)) : succMinOf K N S i ∈ succMinRadii K N S i

theorem **succMinOf_pos** (hfin : ∀ R, {x | x ∈ S ∧ N x ≤ R}.Finite) (hN : ∀ x ∈ S, x ≠ 0 → 0 < N x) (h1 : 1 ≤ i) (hi : (i : Cardinal) ≤ Module.rank K (span K S)) : 0 < succMinOf K N S i

theorem **succMinOf_one_le** {x : V} (hx : x ∈ S) (h0 : x ≠ 0) (hN : 0 ≤ N x) : succMinOf K N S 1 ≤ N x

theorem **le_succMinOf_of_forall** {a : ℝ} (h : ∀ x ∈ S, x ≠ 0 → a ≤ N x) (h1 : 1 ≤ i) (hi : (i : Cardinal) ≤ Module.rank K (span K S)) : a ≤ succMinOf K N S i

theorem **finrank_span_eq_finrank_int** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] : finrank ℝ (span ℝ (L : Set E)) = finrank ℤ L

theorem **finrank_span_of_isZLattice** (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] : finrank ℝ (span ℝ (L : Set E)) = finrank ℝ E

theorem **lastMin_eq_succMin_finrank_int** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] : lastMin L = succMin L (finrank ℤ L)

theorem **natCast_le_rank_of_le_finrank** {K W : Type*} [DivisionRing K] [AddCommGroup W] [Module K W] (h : i ≤ finrank K W) : (i : Cardinal) ≤ Module.rank K W

theorem **one_le_rank_span_of_ne_bot** (h : L ≠ ⊥) : ((1 : ℕ) : Cardinal) ≤ Module.rank ℝ (span ℝ (L : Set E))

theorem **finite_mem_and_norm_le** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] (R : ℝ) : {x : E | x ∈ L ∧ ‖x‖ ≤ R}.Finite

theorem **succMin_nonneg** (L : Submodule ℤ E) (i : ℕ) : 0 ≤ succMin L i

@[simp] theorem **succMin_zero** (L : Submodule ℤ E) : succMin L 0 = 0

theorem **exists_linearIndependent_norm_le_succMin** [FiniteDimensional ℝ E] [DiscreteTopology L] (hi : i ≤ finrank ℝ (span ℝ (L : Set E))) : ∃ v : Fin i → E, LinearIndependent ℝ v ∧ ∀ k, v k ∈ L ∧ ‖v k‖ ≤ succMin L i

theorem **succMin_mono** (hij : i ≤ j) (hj : j ≤ finrank ℝ (span ℝ (L : Set E))) : succMin L i ≤ succMin L j

theorem **succMin_mono_of_isZLattice** [DiscreteTopology L] [IsZLattice ℝ L] (hij : i ≤ j) (hj : j ≤ finrank ℝ E) : succMin L i ≤ succMin L j

theorem **succMin_pos** [FiniteDimensional ℝ E] [DiscreteTopology L] (h1 : 1 ≤ i) (hi : i ≤ finrank ℝ (span ℝ (L : Set E))) : 0 < succMin L i

theorem **succMin_pos_of_isZLattice** [FiniteDimensional ℝ E] [DiscreteTopology L] [IsZLattice ℝ L] (h1 : 1 ≤ i) (hi : i ≤ finrank ℝ E) : 0 < succMin L i

theorem **minNorm_le_norm** {x : E} (hx : x ∈ L) (h0 : x ≠ 0) : minNorm L ≤ ‖x‖

theorem **le_minNorm** {a : ℝ} (hL : L ≠ ⊥) (h : ∀ x ∈ L, x ≠ 0 → a ≤ ‖x‖) : a ≤ minNorm L

theorem **succMin_map_linearIsometry** (f : E →ₗᵢ[ℝ] F) (L : Submodule ℤ E) (i : ℕ) : succMin (L.map (f.toLinearMap.restrictScalars ℤ)) i = succMin L i

theorem **succMin_comap_linearIsometry** (f : F →ₗᵢ[ℝ] E) (h : (L : Set E) ⊆ Set.range f) (i : ℕ) : succMin (ZLattice.comap ℝ L f.toLinearMap) i = succMin L i

@[simp] theorem **mem_inSpan** {x : span ℝ (L : Set E)} : x ∈ inSpan L ↔ (x : E) ∈ L

theorem **succMin_inSpan** (L : Submodule ℤ E) (i : ℕ) : succMin (inSpan L) i = succMin L i

theorem **lastMin_inSpan** (L : Submodule ℤ E) : lastMin (inSpan L) = lastMin L

theorem **smul_eq_map** (c : ℝ) (L : Submodule ℤ E) : c • L = L.map ((c • LinearMap.id : E →ₗ[ℝ] E).restrictScalars ℤ)

theorem **mem_smul_iff_inv_smul_mem** {c : ℝ} (hc : c ≠ 0) {x : E} : x ∈ c • L ↔ c⁻¹ • x ∈ L

theorem **succMin_smul** {c : ℝ} (hc : c ≠ 0) (L : Submodule ℤ E) (i : ℕ) : succMin (c • L) i = |c| * succMin L i

theorem **minNorm_smul** {c : ℝ} (hc : c ≠ 0) (L : Submodule ℤ E) : minNorm (c • L) = |c| * minNorm L

theorem **exists_norm_sub_eq_infDist** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] (x : E) : ∃ w ∈ L, ‖x - w‖ = Metric.infDist x (L : Set E)

theorem **coveringRadius_le** (h : ∀ x : E, ∃ w ∈ L, ‖x - w‖ ≤ r) : coveringRadius L ≤ r

theorem **mem_dualLattice** {y : E} : y ∈ dualLattice L ↔ ∀ x ∈ L, ∃ m : ℤ, (m : ℝ) = ⟪x, y⟫_ℝ

theorem **innerₗ_nondegenerate** (E : Type*) [NormedAddCommGroup E] [InnerProductSpace ℝ E] : LinearMap.BilinForm.Nondegenerate (innerₗ E)

theorem **inner_innerDualBasis** {ι : Type*} [Finite ι] [DecidableEq ι] (b : Basis ι ℝ E) (i j : ι) : ⟪innerDualBasis b i, b j⟫_ℝ = if j = i then 1 else 0

theorem **dualLattice_span_basis** {ι : Type*} [Finite ι] [DecidableEq ι] (b : Basis ι ℝ E) : dualLattice (span ℤ (Set.range b)) = span ℤ (Set.range (innerDualBasis b))

theorem **dualLattice_smul** {c : ℝ} (hc : c ≠ 0) (L : Submodule ℤ E) : dualLattice (c • L) = c⁻¹ • dualLattice L

theorem **dualLattice_eq_span** {ι : Type*} [Finite ι] [DecidableEq ι] (b : Basis ι ℤ L) : dualLattice L = span ℤ (Set.range (innerDualBasis (b.ofZLatticeBasis ℝ L)))

@[simp] theorem **dualLattice_dualLattice** : dualLattice (dualLattice L) = L

theorem **det_mul_det_innerDualBasis** {ι : Type*} [Fintype ι] [DecidableEq ι] (o : OrthonormalBasis ι ℝ E) (b : Basis ι ℝ E) : o.toBasis.det b * o.toBasis.det (innerDualBasis b) = 1

theorem **covolume_span_basis** {ι : Type*} [Fintype ι] [DecidableEq ι] (o : OrthonormalBasis ι ℝ E) (b : Basis ι ℝ E) : ZLattice.covolume (span ℤ (Set.range b)) = |o.toBasis.det b|

theorem **covolume_mul_covolume_dualLattice** : ZLattice.covolume L * ZLattice.covolume (dualLattice L) = 1

theorem **covolume_dualLattice** : ZLattice.covolume (dualLattice L) = (ZLattice.covolume L)⁻¹

@[simp] theorem **finrank_eq** : finrank ℝ (QuadSpace Q hQ) = finrank ℝ V

@[simp] theorem **inner_equiv** (x y : V) : ⟪equiv Q hQ x, equiv Q hQ y⟫_ℝ = QuadraticMap.associated (R

@[simp] theorem **norm_equiv_sq** (x : V) : ‖equiv Q hQ x‖ ^ 2 = Q x

@[simp] theorem **equiv_mem_lattice** {L : Submodule ℤ V} {x : V} : equiv Q hQ x ∈ lattice Q hQ L ↔ x ∈ L

theorem **lattice_span_basis** {ι : Type*} (b : Basis ι ℝ V) : lattice Q hQ (span ℤ (Set.range b)) = span ℤ (Set.range (b.map (equiv Q hQ)))

theorem **isZLattice_lattice** {L : Submodule ℤ V} [DiscreteTopology (lattice Q hQ L)] (h : span ℝ (L : Set V) = ⊤) : IsZLattice ℝ (lattice Q hQ L) where

theorem **mem_one_real_iff** {x : ℝ} : x ∈ (1 : Submodule ℤ ℝ) ↔ ∃ m : ℤ, (m : ℝ) = x

theorem **minNorm_one_real** : minNorm (1 : Submodule ℤ ℝ) = 1

theorem **dualLattice_one_real** : dualLattice (1 : Submodule ℤ ℝ) = 1

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsSecNonzeroLocusEqBasicOpen

theorem **exists_sec_of_refinement** {M : LB X} (hM : M.IsLB) {α : Type} (W : α → X.Opens) (i : α → M.ι) (hW : ∀ a, W a ≤ M.U (i a)) (hc : ∀ y : X, ∃ a, y ∈ W a) (t : ∀ a, Γ(X, W a)) (ht : ∀ (y : X) a b, y ∈ W a → y ∈ W b → gs (𝟙 X) y _ (t a) = (gm (𝟙 X) y (M.U (i a) ⊓ M.U

(i b)) (M.g (i a) (i b))).val * gs (𝟙 X) y _ (t b)) : ∃ u : M.Sec, ∀ (y : X) k a, y ∈ M.U k → y ∈ W a → gs (𝟙 X) y _ (u.f k) = (gm (𝟙 X) y (M.U k ⊓ M.U (i a)) (M.g k (i a))).val * gs (𝟙 X) y _ (t a)

theorem **exists_finite_affine_refinement** [CompactSpace X] {M : LB X} (hM : M.IsLB) : ∃ (α : Type) (_ : Fintype α) (W : α → X.Opens) (i : α → M.ι), (∀ a, IsAffineOpen (W a)) ∧ (∀ a, W a ≤ M.U (i a)) ∧ ∀ y : X, ∃ a, y ∈ W a

theorem **exists_pow_mul_gs_eq_zero** {U : X.Opens} (hU : IsCompact (U : Set X)) (f x : Γ(X, U)) (h : ∀ y ∈ X.basicOpen f, gs (𝟙 X) y U x = 0) : ∃ e : ℕ, ∀ y ∈ U, gs (𝟙 X) y U f ^ e * gs (𝟙 X) y U x = 0

theorem **exists_pow_mul_powFn_eq_zero** [CompactSpace X] (hL : L.IsLB) (s : (lbPow L n).Sec) (u : (lbPow L m).Sec) (h : ∀ J, resG X (inf_le_right : nonzeroLocus s ⊓ powChart L J ≤ powChart L J) (powFn u J) = 0) : ∃ k : ℕ, ∀ J, powFn s J ^ k * powFn u J = 0

theorem **exists_sec_eq_pow_mul** [CompactSpace X] [QuasiSeparatedSpace X] (hL : L.IsLB) (s : (lbPow L n).Sec) (t : ∀ J : PUnit → L.ι, Γ(X, nonzeroLocus s ⊓ powChart L J)) (ht : t ∈ secOn (lbPow L m) (nonzeroLocus s)) : ∃ N : ℕ, ∀ k, N ≤ k → ∃ u : (lbPow L (m + n * k)).Sec, ∀ J, resG X (inf_le_right : nonzeroLocus s ⊓ powChart L J ≤ powChart L J) (powFn u J) = resG X (inf_le_right : nonzeroLocus s ⊓ powChart L J ≤ powChart L J) (powFn s J) ^ k * t J

theorem **exists_nonzeroLocus_eq_basicOpen** [CompactSpace X] [QuasiSeparatedSpace X] (hL : L.IsLB) (s : (lbPow L n).Sec) (f : Γ(X, nonzeroLocus s)) : ∃ k, 1 ≤ k ∧ ∃ u : (lbPow L (n * k)).Sec, nonzeroLocus u = X.basicOpen f

theorem **exists_sec_nonzeroLocus_eq_basicOpen** [CompactSpace X] [QuasiSeparatedSpace X] (hL : L.IsLB) (r : Γ(X, ⊤)) (i : L.ι) (hr : X.basicOpen r ≤ L.U i) : ∃ u : L.Sec, nonzeroLocus u = X.basicOpen r

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsSecNonzeroLocusEqOfLbIso

theorem **gs_pow** (r : Z ⟶ X) (z : Z) (U : X.Opens) (h : r z ∈ U) (a : Γ(X, U)) (n : ℕ) : gs r z U (a ^ n) = gs r z U a ^ n

theorem **gs_sub** (r : Z ⟶ X) (z : Z) (U : X.Opens) (h : r z ∈ U) (a b : Γ(X, U)) : gs r z U (a - b) = gs r z U a - gs r z U b

theorem **gs_zero** (r : Z ⟶ X) (z : Z) (U : X.Opens) (h : r z ∈ U) : gs r z U 0 = 0

theorem **isUnit_gs_iff** (r : Z ⟶ X) (z : Z) (U : X.Opens) (h : r z ∈ U) (a : Γ(X, U)) : IsUnit (gs r z U a) ↔ r z ∈ X.basicOpen a

theorem **gs_secOn** {M : LB X} {V : X.Opens} {t : ∀ i, Γ(X, V ⊓ M.U i)} (ht : t ∈ secOn M V) (y : X) (i j : M.ι) (hV : y ∈ V) (hi : y ∈ M.U i) (hj : y ∈ M.U j) : gs (𝟙 X) y _ (t i) = (gm (𝟙 X) y (M.U i ⊓ M.U j) (M.g i j)).val * gs (𝟙 X) y _ (t j)

theorem **mem_secOn_of_germ** {M : LB X} {V : X.Opens} (t : ∀ i, Γ(X, V ⊓ M.U i)) (h : ∀ (y : X) i j, y ∈ V → y ∈ M.U i → y ∈ M.U j → gs (𝟙 X) y _ (t i) = (gm (𝟙 X) y (M.U i ⊓ M.U j) (M.g i j)).val * gs (𝟙 X) y _ (t j)) : t ∈ secOn M V

theorem **basicOpen_le_nonzeroLocus** (s : M.Sec) (k : M.ι) : Y.basicOpen (s.f k) ≤ nonzeroLocus s

theorem **exists_chart_of_mem_nonzeroLocus** {s : M.Sec} {y : Y} (h : y ∈ nonzeroLocus s) : ∃ k, y ∈ M.U k

theorem **mem_nonzeroLocus_iff_pull** (s : M.Sec) (r : Z ⟶ Y) (z : Z) {k : M.ι} (hk : r z ∈ M.U k) : r z ∈ nonzeroLocus s ↔ IsUnit (gs r z _ (s.f k))

theorem **mem_nonzeroLocus_iff_isUnit** (s : M.Sec) {y : Y} {k : M.ι} (hk : y ∈ M.U k) : y ∈ nonzeroLocus s ↔ IsUnit (gs (𝟙 Y) y _ (s.f k))

theorem **nonzeroLocus_inf_chart** (s : M.Sec) (k : M.ι) : nonzeroLocus s ⊓ M.U k = Y.basicOpen (s.f k)

theorem **mem_nonzeroLocus_iff_basicOpen** (s : M.Sec) {y : Y} {k : M.ι} (hk : y ∈ M.U k) : y ∈ nonzeroLocus s ↔ y ∈ Y.basicOpen (s.f k)

theorem **nonzeroLocus_eq_of** (s : M.Sec) (W : Y.Opens) (hW : ∀ y ∈ W, ∃ k, y ∈ M.U k) (h : ∀ (y : Y) k, y ∈ M.U k → (IsUnit (gs (𝟙 Y) y _ (s.f k)) ↔ y ∈ W)) : nonzeroLocus s = W

theorem **nonzeroLocus_secZero** : nonzeroLocus (DVR.SecOps.secZero : M.Sec) = ⊥

theorem **nonzeroLocus_secScale** (c : Γ(Y, ⊤)) (s : M.Sec) : nonzeroLocus (DVR.SecOps.secScale c s) = Y.basicOpen c ⊓ nonzeroLocus s

theorem **nonzeroLocus_genSec** (hM : M.IsLB) (t : M.Gen) (ht : t.IsGen) : nonzeroLocus (DVR.genSec t ht) = ⊤

theorem **isAffineHom_nonzeroLocus_ι** (hM : M.IsLB) (s : M.Sec) : IsAffineHom (nonzeroLocus s).ι

theorem **isAffineOpen_nonzeroLocus_inf** (hM : M.IsLB) (s : M.Sec) {V : Y.Opens} (hV : IsAffineOpen V) : IsAffineOpen (nonzeroLocus s ⊓ V)

theorem **isAffineOpen_nonzeroLocus** [IsAffine Y] (hM : M.IsLB) (s : M.Sec) : IsAffineOpen (nonzeroLocus s)

theorem **nonzeroLocus_secOfIso** {A B : LB Y} {u : LB.Iso A B} (t : B.Sec) (hu : u.IsGen) (hA : A.IsLB) (hB : B.IsLB) : nonzeroLocus (DVR.secOfIso t hu hA hB) = nonzeroLocus t

theorem **exists_sec_nonzeroLocus_eq_of_lbIso** {A B : LB Y} (hA : A.IsLB) (hB : B.IsLB) (H : LBIso A B) (t : B.Sec) : ∃ s : A.Sec, nonzeroLocus s = nonzeroLocus t

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.ExistsSmulMemOfMemSpanLattice

theorem **sesqGram_apply** (B : V →ₛₗ[σ] V →ₛₗ[ComplexEmbedding.conjugate σ] ℂ) {ι : Type*} (e : ι → V) (i j : ι) : sesqGram B e i j = B (e i) (e j)

theorem **sesqGram_of_eq_sum** (B : V →ₛₗ[σ] V →ₛₗ[ComplexEmbedding.conjugate σ] ℂ) {ι κ : Type*} [Fintype ι] {e : ι → V} {f : κ → V} {g : Matrix κ ι K} (hf : ∀ i, f i = ∑ j, g i j • e j) : sesqGram B f = g.map σ * sesqGram B e * (g.map σ)ᴴ

theorem **sesqGram_posDef_of_basis** (B : V →ₛₗ[σ] V →ₛₗ[ComplexEmbedding.conjugate σ] ℂ) {ι κ : Type*} [Fintype ι] [Finite κ] (b : Basis ι K V) (hb : (sesqGram B b).PosDef) {e : κ → V} (he : LinearIndependent K e) : (sesqGram B e).PosDef

theorem **ofBasis_form** {ι : Type*} [Finite ι] (b : Basis ι K V) (B : ∀ σ : K →+* ℂ, V →ₛₗ[σ] V →ₛₗ[ComplexEmbedding.conjugate σ] ℂ) (symm : ∀ σ x y, B σ y x = conj (B σ x y)) (pos : ∀ σ, (sesqGram (B σ) b).PosDef) : (ofBasis b B symm pos).form = B

theorem **ofFun_form** {ι : Type*} [Finite ι] (b : Basis ι K V) (H : (K →+* ℂ) → V → V → ℂ) (add : ∀ σ x x' y, H σ (x + x') y = H σ x y + H σ x' y) (smul : ∀ σ (a : K) x y, H σ (a • x) y = σ a * H σ x y) (symm : ∀ σ x y, H σ y x = conj (H σ x y)) (pos : ∀ σ, (Matrix.of fun i j => H σ (b i) (b j)).PosDef) (σ : K →+* ℂ) (x y : V) : (ofFun b H add smul symm pos).form σ x y = H σ x y

theorem **ofGram_form** {ι : Type*} [Fintype ι] (b : Basis ι K V) (G : (K →+* ℂ) → Matrix ι ι ℂ) (hG : ∀ σ, (G σ).PosDef) (σ : K →+* ℂ) (x y : V) : (ofGram b G hG).form σ x y = ∑ i, ∑ j, σ (b.repr x i) * G σ i j * conj (σ (b.repr y j))

theorem **diag_form** {ι : Type*} [Fintype ι] (d : (K →+* ℂ) → ι → ℝ) (hd : ∀ σ i, 0 < d σ i) (σ : K →+* ℂ) (x y : ι → K) : (diag d hd).form σ x y = ∑ i, (d σ i : ℂ) * (σ (x i) * conj (σ (y i)))

theorem **std_form** {ι : Type*} [Fintype ι] (σ : K →+* ℂ) (x y : ι → K) : (std K ι).form σ x y = ∑ i, σ (x i) * conj (σ (y i))

theorem **stdField_form** (σ : K →+* ℂ) (x y : K) : (stdField K).form σ x y = σ x * conj (σ y)

theorem **form_smul_left** (σ : K →+* ℂ) (a : K) (x y : V) : h.form σ (a • x) y = σ a * h.form σ x y

theorem **form_smul_right** (σ : K →+* ℂ) (a : K) (x y : V) : h.form σ x (a • y) = conj (σ a) * h.form σ x y

theorem **conj_form** (σ : K →+* ℂ) (x y : V) : conj (h.form σ x y) = h.form σ y x

theorem **conjInvariant_diag** {ι : Type*} [Fintype ι] {d : (K →+* ℂ) → ι → ℝ} (hd : ∀ σ i, 0 < d σ i) (hc : ∀ σ, d (ComplexEmbedding.conjugate σ) = d σ) : (diag d hd).ConjInvariant

theorem **conjInvariant_ofGram** {ι : Type*} [Fintype ι] (b : Basis ι K V) {G : (K →+* ℂ) → Matrix ι ι ℂ} (hG : ∀ σ, (G σ).PosDef) (hc : ∀ σ i j, G (ComplexEmbedding.conjugate σ) i j = conj (G σ i j)) : (ofGram b G hG).ConjInvariant

theorem **conjInvariant_std** (ι : Type*) [Fintype ι] : (std K ι).ConjInvariant

theorem **conjInvariant_stdField** : (stdField K).ConjInvariant

theorem **gram_apply** (σ : K →+* ℂ) {ι : Type*} (e : ι → V) (i j : ι) : h.gram σ e i j = h.form σ (e i) (e j)

theorem **isHermitian_gram** (σ : K →+* ℂ) {ι : Type*} (e : ι → V) : (h.gram σ e).IsHermitian

theorem **gram_comp** (σ : K →+* ℂ) {ι κ : Type*} (e : ι → V) (p : κ → ι) : h.gram σ (e ∘ p) = (h.gram σ e).submatrix p p

theorem **gram_of_eq_sum** (σ : K →+* ℂ) {ι κ : Type*} [Fintype ι] {e : ι → V} {f : κ → V} {g : Matrix κ ι K} (hf : ∀ i, f i = ∑ j, g i j • e j) : h.gram σ f = g.map σ ∗ h.gram σ e ∗ (g.map σ)ᴴ

theorem **posDef_gram** (σ : K →+* ℂ) {ι : Type*} [Finite ι] {e : ι → V} (he : LinearIndependent K e) : (h.gram σ e).PosDef

theorem **posSemidef_gram** (σ : K →+* ℂ) {ι : Type*} [Finite ι] (e : ι → V) : (h.gram σ e).PosSemidef

theorem **gram_ofGram** {ι : Type*} [Fintype ι] (b : Basis ι K V) (G : (K →+* ℂ) → Matrix ι ι ℂ) (hG : ∀ σ, (G σ).PosDef) (σ : K →+* ℂ) : (ofGram b G hG).gram σ b = G σ

theorem **gram_diag** {ι : Type*} [Fintype ι] [DecidableEq ι] (d : (K →+* ℂ) → ι → ℝ) (hd : ∀ σ i, 0 < d σ i) (σ : K →+* ℂ) : (diag d hd).gram σ (Pi.basisFun K ι) = Matrix.diagonal fun i => (d σ i : ℂ)

theorem **ofReal_gramDet** (e : ι → V) : (h.gramDet σ e : ℂ) = (h.gram σ e).det

theorem **gramDet_nonneg** (e : ι → V) : 0 ≤ h.gramDet σ e

theorem **gramDet_pos** {e : ι → V} (he : LinearIndependent K e) : 0 < h.gramDet σ e

theorem **gramDet_comp_equiv** (e : ι → V) (p : κ ≃ ι) : h.gramDet σ (e ∘ p) = h.gramDet σ e

theorem **gramDet_of_eq_sum** {e f : ι → V} {g : Matrix ι ι K} (hf : ∀ i, f i = ∑ j, g i j • e j) : h.gramDet σ f = ‖σ g.det‖ ^ 2 ∗ h.gramDet σ e

theorem **gramDet_pos_iff** {e : ι → V} : 0 < h.gramDet σ e ↔ LinearIndependent K e

theorem **gramDet_of_isEmpty** [IsEmpty ι] (e : ι → V) : h.gramDet σ e = 1

theorem **gramDet_diag** (d : (K →+* ℂ) → ι → ℝ) (hd : ∀ σ i, 0 < d σ i) : (diag d hd).gramDet σ (Pi.basisFun K ι) = ∏ i, d σ i

theorem **ConjInvariant.gramDet_conjugate** (hc : h.ConjInvariant) (e : ι → V) : h.gramDet (ComplexEmbedding.conjugate σ) e = h.gramDet σ e

theorem **form_self_nonneg** (σ : K →+* ℂ) (x : V) : 0 ≤ h.form σ x x

theorem **norm_nonneg** (σ : K →+* ℂ) (x : V) : 0 ≤ h.norm σ x

theorem **norm_sq** (σ : K →+* ℂ) (x : V) : h.norm σ x ^ 2 = (h.form σ x x).re

theorem **form_self** (σ : K →+* ℂ) (x : V) : h.form σ x x = ((h.norm σ x ^ 2 : ℝ) : ℂ)

theorem **gramDet_unique** (σ : K →+* ℂ) {ι : Type*} [Fintype ι] [DecidableEq ι] [Unique ι] (e : ι → V) : h.gramDet σ e = h.norm σ (e default) ^ 2

theorem **norm_zero** (σ : K →+* ℂ) : h.norm σ 0 = 0

theorem **norm_pos** (σ : K →+* ℂ) {x : V} (hx : x ≠ 0) : 0 < h.norm σ x

theorem **norm_eq_zero** {σ : K →+* ℂ} {x : V} : h.norm σ x = 0 ↔ x = 0

theorem **norm_smul** (σ : K →+* ℂ) (a : K) (x : V) : h.norm σ (a • x) = ‖σ a‖ ∗ h.norm σ x

theorem **stdField_norm** (σ : K →+* ℂ) (x : K) : (stdField K).norm σ x = ‖σ x‖

theorem **norm_neg** (σ : K →+* ℂ) (x : V) : h.norm σ (-x) = h.norm σ x

theorem **norm_form_le** (σ : K →+* ℂ) (x y : V) : ‖h.form σ x y‖ ≤ h.norm σ x ∗ h.norm σ y

theorem **norm_add_sq** (σ : K →+* ℂ) (x y : V) : h.norm σ (x + y) ^ 2 = h.norm σ x ^ 2 + 2 ∗ (h.form σ x y).re + h.norm σ y ^ 2

theorem **norm_add_le** (σ : K →+* ℂ) (x y : V) : h.norm σ (x + y) ≤ h.norm σ x + h.norm σ y

theorem **ConjInvariant.norm_conjugate** (hc : h.ConjInvariant) (σ : K →+* ℂ) (x : V) : h.norm (ComplexEmbedding.conjugate σ) x = h.norm σ x

theorem **comap_form** (x y : V') : (h.comap f hf).form σ x y = h.form σ (f x) (f y)

theorem **comap_norm** (x : V') : (h.comap f hf).norm σ x = h.norm σ (f x)

theorem **comap_gram** {ι : Type*} (e : ι → V') : (h.comap f hf).gram σ e = h.gram σ (f ∘ e)

theorem **comap_gramDet** {ι : Type*} [Fintype ι] [DecidableEq ι] (e : ι → V') : (h.comap f hf).gramDet σ e = h.gramDet σ (f ∘ e)

theorem **comap_id** : h.comap LinearMap.id Function.injective_id = h

theorem **comap_comap** (g : V'' →ₗ[K] V') (hg : Function.Injective g) : (h.comap f hf).comap g hg = h.comap (f ∘ₗ g) (hf.comp hg)

theorem **ConjInvariant.comap** (hc : h.ConjInvariant) : (h.comap f hf).ConjInvariant

theorem **restrict_form** (W : Submodule K V) (σ : K →+* ℂ) (x y : W) : (h.restrict W).form σ x y = h.form σ x y

theorem **restrict_norm** (W : Submodule K V) (σ : K →+* ℂ) (x : W) : (h.restrict W).norm σ x = h.norm σ x

theorem **restrict_gramDet** (W : Submodule K V) (σ : K →+* ℂ) {ι : Type*} [Fintype ι] [DecidableEq ι] (e : ι → W) : (h.restrict W).gramDet σ e = h.gramDet σ fun i => (e i : V)

theorem **ConjInvariant.restrict** (hc : h.ConjInvariant) (W : Submodule K V) : (h.restrict W).ConjInvariant

theorem **scale_form** (x y : V) : (h.scale c hc).form σ x y = c σ ∗ h.form σ x y

theorem **scale_norm** (x : V) : (h.scale c hc).norm σ x = Real.sqrt (c σ) ∗ h.norm σ x

theorem **scale_gram** {ι : Type*} (e : ι → V) : (h.scale c hc).gram σ e = (c σ : ℂ) • h.gram σ e

theorem **scale_gramDet** {ι : Type*} [Fintype ι] [DecidableEq ι] (e : ι → V) : (h.scale c hc).gramDet σ e = c σ ^ Fintype.card ι ∗ h.gramDet σ e

theorem **ConjInvariant.scale** (hh : h.ConjInvariant) (hcc : ∀ σ, c (ComplexEmbedding.conjugate σ) = c σ) : (h.scale c hc).ConjInvariant

theorem **scaleNormExp_norm** (t : (K →+* ℂ) → ℝ) (σ : K →+* ℂ) (x : V) : (h.scaleNormExp t).norm σ x = Real.exp (t σ) ∗ h.norm σ x

theorem **relIndex_toAddSubgroup_eq_card** {R M : Type*} [Ring R] [AddCommGroup M] [Module R M] (B A : Submodule R M) : B.toAddSubgroup.relIndex A.toAddSubgroup = Nat.card (A ⧸ B.submoduleOf A)

theorem **index_range_eq_natAbs_norm_det** {S M : Type*} [CommRing S] [Module.Free ℤ S] [Module.Finite ℤ S] [AddCommGroup M] [Module S M] [Module.Free S M] [Module.Finite S M] (φ : M →ₗ[S] M) (hφ : Function.Injective φ) : (LinearMap.range φ).toAddSubgroup.index = (Algebra.norm ℤ (LinearMap.det φ)).natAbs

theorem **finrank_lattice_eq_finrank_span** (A : Submodule (𝒪 K) V) : finrank (𝒪 K) A = finrank K (Submodule.span K (A : Set V))

theorem **finiteDimensional_span_of_fg** {A : Submodule (𝒪 K) V} (hA : A.FG) : FiniteDimensional K (Submodule.span K (A : Set V))

theorem **finrank_lattice_pos** {A : Submodule (𝒪 K) V} (hA : A.FG) (h0 : A ≠ ⊥) : 0 < finrank (𝒪 K) A

theorem **finrank_inf_restrictScalars_lt** {A : Submodule (𝒪 K) V} (hA : A.FG) {W : Submodule K V} (hW : ¬ A ≤ W.restrictScalars (𝒪 K)) : finrank (𝒪 K) (A ⊓ W.restrictScalars (𝒪 K) : Submodule (𝒪 K) V) < finrank (𝒪 K) A

theorem **mem_span_of_le_of_finrank_eq** {A B : Submodule (𝒪 K) V} (hA : A.FG) (hBA : B ≤ A) (hr : finrank (𝒪 K) B = finrank (𝒪 K) A) : ∀ x ∈ A, x ∈ Submodule.span K (B : Set V)

theorem **exists_smul_mem_of_mem_span_lattice** {B : Submodule (𝒪 K) V} {x : V} (hx : x ∈ Submodule.span K (B : Set V)) : ∃ a : 𝒪 K, a ≠ 0 ∧ a • x ∈ B

theorem **relIndex_ne_zero_of_le_span** {A B : Submodule (𝒪 K) V} (hA : A.FG) (hAB : ∀ x ∈ A, x ∈ Submodule.span K (B : Set V)) : B.toAddSubgroup.relIndex A.toAddSubgroup ≠ 0

theorem **relIndex_span_eq_natAbs_norm_det** {ι : Type*} [Fintype ι] [DecidableEq ι] {e f : ι → V} (he : LinearIndependent K e) {g : Matrix ι ι (𝒪 K)} (hg : g.det ≠ 0) (hf : ∀ i, f i = ∑ j, g i j • e j) : (Submodule.span (𝒪 K) (Set.range f)).toAddSubgroup.relIndex (Submodule.span (𝒪 K) (Set.range e)).toAddSubgroup = (Algebra.norm ℤ g.det).natAbs

theorem **log_natAbs_norm_eq_sum** {a : 𝒪 K} (ha : a ≠ 0) : Real.log ((Algebra.norm ℤ a).natAbs : ℝ) = ∑ σ : K →+* ℂ, Real.log ‖σ (a : K)‖

theorem **span_eq** (he : IsLatticeFrame A e) : Submodule.span K (Set.range e) = Submodule.span K (A : Set V)

theorem **span_le** (he : IsLatticeFrame A e) : Submodule.span (𝒪 K) (Set.range e) ≤ A

theorem **card_eq** [Fintype ι] (he : IsLatticeFrame A e) : Fintype.card ι = finrank (𝒪 K) A

theorem **comp_equiv** (he : IsLatticeFrame A e) (p : κ ≃ ι) : IsLatticeFrame A (e ∘ p)

theorem **relIndex_ne_zero** (he : IsLatticeFrame A e) (hA : A.FG) : (Submodule.span (𝒪 K) (Set.range e)).toAddSubgroup.relIndex A.toAddSubgroup ≠ 0

theorem **finite_quotient** (he : IsLatticeFrame A e) (hA : A.FG) : Finite (A ⧸ (Submodule.span (𝒪 K) (Set.range e)).submoduleOf A)

theorem **of_le** {B : Submodule (𝒪 K) V} (he : IsLatticeFrame A e) (hAB : A ≤ B) (hB : ∀ x ∈ B, x ∈ Submodule.span K (A : Set V)) : IsLatticeFrame B e

theorem **map** (he : IsLatticeFrame A e) (f : V →ₗ[K] V') (hf : Function.Injective f) : IsLatticeFrame (A.map (f.restrictScalars (𝒪 K))) (f ∘ e) where

theorem **smul** (he : IsLatticeFrame A e) {d : ι → 𝒪 K} (hd : ∀ i, d i ≠ 0) : IsLatticeFrame A fun i => d i • e i

theorem **isLatticeFrame_span** {ι : Type*} {e : ι → V} (he : LinearIndependent K e) : IsLatticeFrame (Submodule.span (𝒪 K) (Set.range e)) e

theorem **isLatticeFrame_iff_card_eq** {A : Submodule (𝒪 K) V} (hA : A.FG) {ι : Type*} [Fintype ι] {e : ι → V} : IsLatticeFrame A e ↔ (∀ i, e i ∈ A) ∧ LinearIndependent K e ∧ Fintype.card ι = finrank (𝒪 K) A

theorem **exists_isLatticeFrame** {A : Submodule (𝒪 K) V} (hA : A.FG) : ∃ e : Fin (finrank (𝒪 K) A) → V, IsLatticeFrame A e

theorem **frameDeg_eq_relIndex** (A : Submodule (𝒪 K) V) (e : ι → V) : h.frameDeg A e = Real.log ((Submodule.span (𝒪 K) (Set.range e)).toAddSubgroup.relIndex A.toAddSubgroup) - ∑ σ : K →+* ℂ, 1 / 2 * Real.log (h.gramDet σ e)

theorem **frameDeg_comp_equiv** (A : Submodule (𝒪 K) V) (e : ι → V) (p : κ ≃ ι) : h.frameDeg A (e ∘ p) = h.frameDeg A e

theorem **frameDeg_eq_of_mem_span** (hA : A.FG) {e f : ι → V} (he : IsLatticeFrame A e) (hf : IsLatticeFrame A f) (hfe : ∀ i, f i ∈ Submodule.span (𝒪 K) (Set.range e)) : h.frameDeg A f = h.frameDeg A e

theorem **frameDeg_eq_frameDeg** (hA : A.FG) {e : ι → V} {f : κ → V} (he : IsLatticeFrame A e) (hf : IsLatticeFrame A f) : h.frameDeg A e = h.frameDeg A f

theorem **arDeg_eq_frameDeg** (hA : A.FG) {e : ι → V} (he : IsLatticeFrame A e) : h.arDeg A = h.frameDeg A e

theorem **arDeg_eq** (hA : A.FG) {e : ι → V} (mem : ∀ i, e i ∈ A) (he : LinearIndependent K e) (hc : Fintype.card ι = finrank (𝒪 K) A) : h.arDeg A = Real.log (Nat.card (A ⧸ (Submodule.span (𝒪 K) (Set.range e)).submoduleOf A)) - ∑ σ : K →+* ℂ, 1 / 2 * Real.log (h.gramDet σ e)

theorem **arDeg_span** {e : ι → V} (he : LinearIndependent K e) : h.arDeg (Submodule.span (𝒪 K) (Set.range e)) = -∑ σ : K →+* ℂ, 1 / 2 * Real.log (h.gramDet σ e)

theorem **arDeg_bot** : h.arDeg ⊥ = 0

theorem **arDeg_span_singleton** {x : V} (hx : x ≠ 0) : h.arDeg (Submodule.span (𝒪 K) {x}) = -∑ σ : K →+* ℂ, Real.log (h.norm σ x)

theorem **arDeg_of_finrank_eq_one** (hA : A.FG) (h1 : finrank (𝒪 K) A = 1) {x : V} (hxA : x ∈ A) (hx : x ≠ 0) : h.arDeg A = Real.log (Nat.card (A ⧸ (Submodule.span (𝒪 K) {x}).submoduleOf A)) - ∑ σ : K →+* ℂ, Real.log (h.norm σ x)

theorem **arDeg_eq_add_log_relIndex** {B : Submodule (𝒪 K) V} (hA : A.FG) (hBA : B ≤ A) (hAB : ∀ x ∈ A, x ∈ Submodule.span K (B : Set V)) : h.arDeg A = h.arDeg B + Real.log (B.toAddSubgroup.relIndex A.toAddSubgroup)

theorem **arDeg_le_of_finrank_eq** {B : Submodule (𝒪 K) V} (hA : A.FG) (hBA : B ≤ A) (hr : finrank (𝒪 K) B = finrank (𝒪 K) A) : h.arDeg B ≤ h.arDeg A

theorem **arDeg_sub_arDeg_of_span_eq** {B : Submodule (𝒪 K) V} (hA : A.FG) (hB : B.FG) (hAB : Submodule.span K (A : Set V) = Submodule.span K (B : Set V)) : h.arDeg A - h.arDeg B = Real.log (B.toAddSubgroup.relIndex A.toAddSubgroup) - Real.log (A.toAddSubgroup.relIndex B.toAddSubgroup)

theorem **arDeg_eq_of_span_eq** (hA : A.FG) {e : ι → V} (he : LinearIndependent K e) (hs : Submodule.span K (Set.range e) = Submodule.span K (A : Set V)) : h.arDeg A = Real.log ((Submodule.span (𝒪 K) (Set.range e)).toAddSubgroup.relIndex A.toAddSubgroup) - Real.log (A.toAddSubgroup.relIndex (Submodule.span (𝒪 K) (Set.range e)).toAddSubgroup) - ∑ σ : K →+* ℂ, 1 / 2 * Real.log (h.gramDet σ e)

theorem **arDeg_comap** (f : V' →ₗ[K] V) (hf : Function.Injective f) {A : Submodule (𝒪 K) V'} (hA : A.FG) : (h.comap f hf).arDeg A = h.arDeg (A.map (f.restrictScalars (𝒪 K)))

theorem **arDeg_restrict** (W : Submodule K V) {A : Submodule (𝒪 K) W} (hA : A.FG) : (h.restrict W).arDeg A = h.arDeg (A.map (W.subtype.restrictScalars (𝒪 K)))

theorem **arDeg_scale** (c : (K →+* ℂ) → ℝ) (hc : ∀ σ, 0 < c σ) (hA : A.FG) : (h.scale c hc).arDeg A = h.arDeg A - finrank (𝒪 K) A / 2 * ∑ σ : K →+* ℂ, Real.log (c σ)

theorem **arDeg_scaleNormExp** (t : (K →+* ℂ) → ℝ) (hA : A.FG) : (h.scaleNormExp t).arDeg A = h.arDeg A - finrank (𝒪 K) A * ∑ σ : K →+* ℂ, t σ

theorem **arDeg_diag** (d : (K →+* ℂ) → ι → ℝ) (hd : ∀ σ i, 0 < d σ i) : (diag d hd).arDeg (Submodule.span (𝒪 K) (Set.range (Pi.basisFun K ι))) = -∑ σ : K →+* ℂ, 1 / 2 * ∑ i, Real.log (d σ i)

theorem **arDeg_std** : (std K ι).arDeg (Submodule.span (𝒪 K) (Set.range (Pi.basisFun K ι))) = 0

theorem **arDeg_stdField_span_singleton** {a : K} (ha : a ≠ 0) : (stdField K).arDeg (Submodule.span (𝒪 K) {a}) = -Real.log |((Algebra.norm ℚ a : ℚ) : ℝ)|

theorem **finrank_mul_degNorm** (A : Submodule (𝒪 K) V) : finrank ℚ K * h.degNorm A = h.arDeg A

theorem **degNorm_bot** : h.degNorm ⊥ = 0

theorem **lcv_sq_eq** (hA : A.FG) {ι : Type*} [Fintype ι] [DecidableEq ι] {e : ι → V} (he : IsLatticeFrame A e) : h.lcv_sq A = (∑ σ : K →+* ℂ, Real.log (h.gramDet σ e)) / finrank ℚ K - 2 / finrank ℚ K * Real.log (Nat.card (A ⧸ (Submodule.span (𝒪 K) (Set.range e)).submoduleOf A))

theorem **deg_sq_eq** (A : Submodule (𝒪 K) V) : h.deg_sq A = 2 * h.degNorm A

theorem **slope_sq_eq** (A : Submodule (𝒪 K) V) : h.slope_sq A = 2 * h.slope A

theorem **arSlope_eq** (A : Submodule (𝒪 K) V) : h.arSlope A = finrank ℚ K * h.slope A

theorem **finrank_mul_slope** (hA : A.FG) (h0 : A ≠ ⊥) : finrank (𝒪 K) A * h.slope A = h.degNorm A

theorem **slope_of_finrank_eq_one** (h1 : finrank (𝒪 K) A = 1) : h.slope A = h.degNorm A

theorem **quotSlope_bot** (A : Submodule (𝒪 K) V) : h.quotSlope A ⊥ = h.slope A

theorem **sub_mul_quotSlope** (hA : A.FG) {W : Submodule K V} (hW : ¬ A ≤ W.restrictScalars (𝒪 K)) : ((finrank (𝒪 K) A : ℝ) - finrank (𝒪 K) (A ⊓ W.restrictScalars (𝒪 K) : Submodule (𝒪 K) V)) * h.quotSlope A (A ⊓ W.restrictScalars (𝒪 K)) = h.degNorm A - h.degNorm (A ⊓ W.restrictScalars (𝒪 K))

theorem **slope_le_muMax** (hBA : B ≤ A) (h0 : B ≠ ⊥) : (h.slope B : EReal) ≤ h.muMax A

theorem **muMax_le_iff** {c : EReal} : h.muMax A ≤ c ↔ ∀ B ≤ A, B ≠ ⊥ → (h.slope B : EReal) ≤ c

theorem **muMax_le_coe_iff** {c : ℝ} : h.muMax A ≤ c ↔ ∀ B ≤ A, B ≠ ⊥ → h.slope B ≤ c

theorem **muMax_mono** (hBA : B ≤ A) : h.muMax B ≤ h.muMax A

theorem **muMax_bot** : h.muMax ⊥ = ⊥

theorem **muMax_ne_bot** (h0 : A ≠ ⊥) : h.muMax A ≠ ⊥

theorem **degNorm_le_of_muMax_le** (hA : A.FG) (hBA : B ≤ A) {c : ℝ} (hc : h.muMax A ≤ c) : h.degNorm B ≤ finrank (𝒪 K) B * c

theorem **muMin_le_quotSlope** {W : Submodule K V} (hW : ¬ A ≤ W.restrictScalars (𝒪 K)) : h.muMin A ≤ h.quotSlope A (A ⊓ W.restrictScalars (𝒪 K))

theorem **le_muMin_iff** {c : EReal} : c ≤ h.muMin A ↔ ∀ W : Submodule K V, ¬ A ≤ W.restrictScalars (𝒪 K) → c ≤ h.quotSlope A (A ⊓ W.restrictScalars (𝒪 K))

theorem **coe_le_muMin_iff** {c : ℝ} : (c : EReal) ≤ h.muMin A ↔ ∀ W : Submodule K V, ¬ A ≤ W.restrictScalars (𝒪 K) → c ≤ h.quotSlope A (A ⊓ W.restrictScalars (𝒪 K))

theorem **muMin_eq_iInf_isSaturatedIn** (A : Submodule (𝒪 K) V) : h.muMin A = ⨅ (B : Submodule (𝒪 K) V) (_ : IsSaturatedIn A B) (_ : B ≠ A), (h.quotSlope A B : EReal)

theorem **muMin_le_slope** (h0 : A ≠ ⊥) : h.muMin A ≤ h.slope A

theorem **muMin_le_muMax** (h0 : A ≠ ⊥) : h.muMin A ≤ h.muMax A

theorem **muMin_bot** : h.muMin ⊥ = ⊤

theorem **muMin_ne_top** (h0 : A ≠ ⊥) : h.muMin A ≠ ⊤

theorem **muMax_of_finrank_eq_one** (hA : A.FG) (h1 : finrank (𝒪 K) A = 1) : h.muMax A = h.slope A

theorem **muMin_of_finrank_eq_one** (hA : A.FG) (h1 : finrank (𝒪 K) A = 1) : h.muMin A = h.slope A

theorem **degNorm_comap** (hA : A'.FG) : (h.comap f hf).degNorm A' = h.degNorm (A'.map (f.restrictScalars (𝒪 K)))

theorem **slope_comap** (hA : A'.FG) : (h.comap f hf).slope A' = h.slope (A'.map (f.restrictScalars (𝒪 K)))

theorem **muMax_comap** (hA : A'.FG) : (h.comap f hf).muMax A' = h.muMax (A'.map (f.restrictScalars (𝒪 K)))

theorem **quotSlope_comap** {B' : Submodule (𝒪 K) V'} (hA : A'.FG) (hB : B'.FG) : (h.comap f hf).quotSlope A' B' = h.quotSlope (A'.map (f.restrictScalars (𝒪 K))) (B'.map (f.restrictScalars (𝒪 K)))

theorem **muMin_comap** (hA : A'.FG) : (h.comap f hf).muMin A' = h.muMin (A'.map (f.restrictScalars (𝒪 K)))

theorem **slope_scaleNormExp** (t : (K →+* ℂ) → ℝ) (hA : A.FG) (h0 : A ≠ ⊥) : (h.scaleNormExp t).slope A = h.slope A - (∑ σ : K →+* ℂ, t σ) / finrank ℚ K

theorem **muMax_sq_le_coe_iff** {c : ℝ} : h.muMax_sq A ≤ c ↔ ∀ B ≤ A, B ≠ ⊥ → h.slope_sq B ≤ c

theorem **muMax_sq_eq_iSup** (A : Submodule (𝒪 K) V) : h.muMax_sq A = ⨆ (B : Submodule (𝒪 K) V) (_ : B ≤ A) (_ : B ≠ ⊥), (h.slope_sq B : EReal)

theorem **coe_le_muMin_sq_iff** {c : ℝ} : (c : EReal) ≤ h.muMin_sq A ↔ ∀ W : Submodule K V, ¬ A ≤ W.restrictScalars (𝒪 K) → c ≤ 2 * h.quotSlope A (A ⊓ W.restrictScalars (𝒪 K))

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.HasSumLocallyUniformlyWeierstrassZetaExcept

theorem **isPreconnected_compl_lattice_sdiff** (s : Set ℂ) : IsPreconnected ((L.lattice : Set ℂ) \ s)ᶜ

theorem **isPreconnected_compl_lattice** : IsPreconnected (L.lattice : Set ℂ)ᶜ

theorem **dense_compl_lattice** : Dense (L.lattice : Set ℂ)ᶜ

theorem **eventually_notMem_lattice** (x : ℂ) : ∀ᶠ z in 𝓝[≠] x, z ∉ L.lattice

theorem **add_notMem_lattice** {L : PeriodPair} {z w : ℂ} (hz : z ∉ L.lattice) (hw : w ∈ L.lattice) : z + w ∉ L.lattice

theorem **eventually_cofinite_le_norm** (R : ℝ) : ∀ᶠ l : L.lattice in cofinite, R ≤ ‖(l : ℂ)‖

theorem **zetaTerm_eq** (l z : ℂ) : zetaTerm l z = 1 / (z − l) + 1 / l + z / l ^ 2

theorem **norm_zetaTerm_le** {r : ℝ} (hr : 0 < r) {z : ℂ} (hz : ‖z‖ < r) {l : ℂ} (h : 2 * r ≤ ‖l‖) : ‖zetaTerm l z‖ ≤ 2 * r ^ 2 * ‖l‖ ^ (−3 : ℝ)

theorem **hasSumLocallyUniformly_weierstrassZetaExcept** (l₀ : ℂ) : HasSumLocallyUniformly (fun (l : L.lattice) (z : ℂ) => if (l : ℂ) = l₀ then 0 else zetaTerm l z) (weierstrassZetaExcept L l₀)

theorem **hasSum_weierstrassZetaExcept** (l₀ z : ℂ) : HasSum (fun l : L.lattice => if (l : ℂ) = l₀ then 0 else zetaTerm l z) (weierstrassZetaExcept L l₀ z)

theorem **weierstrassZetaExcept_of_notMem** {l₀ : ℂ} (h : l₀ ∉ L.lattice) : weierstrassZetaExcept L l₀ = weierstrassZeta L

theorem **hasSumLocallyUniformly_weierstrassZeta** : HasSumLocallyUniformly (fun (l : L.lattice) (z : ℂ) => zetaTerm l z) (weierstrassZeta L)

theorem **hasSum_weierstrassZeta** (z : ℂ) : HasSum (fun l : L.lattice => zetaTerm l z) (weierstrassZeta L z)

theorem **weierstrassZetaExcept_add** {l₀ : ℂ} (h : l₀ ∈ L.lattice) (z : ℂ) : weierstrassZetaExcept L l₀ z + zetaTerm l₀ z = weierstrassZeta L z

theorem **weierstrassZeta_eq** (z : ℂ) : weierstrassZeta L z = 1 / z + ∑' l : L.lattice, if (l : ℂ) = 0 then 0 else (1 / (z − l) + 1 / l + z / l ^ 2)

theorem **hasDerivAt_zetaTerm** {l z : ℂ} (h : z ≠ l) : HasDerivAt (zetaTerm l) (−(1 / (z − l) ^ 2 − 1 / l ^ 2)) z

theorem **hasDerivAt_weierstrassZetaExcept** (l₀ : ℂ) {z : ℂ} (hz : z ∈ ((L.lattice : Set ℂ) \ {l₀})ᶜ) : HasDerivAt (weierstrassZetaExcept L l₀) (−℘[L − l₀] z) z

theorem **analyticOnNhd_weierstrassZetaExcept** (l₀ : ℂ) : AnalyticOnNhd ℂ (weierstrassZetaExcept L l₀) ((L.lattice : Set ℂ) \ {l₀})ᶜ

theorem **analyticAt_weierstrassZetaExcept** (l₀ : ℂ) : AnalyticAt ℂ (weierstrassZetaExcept L l₀) l₀

theorem **hasDerivAt_weierstrassZeta** {z : ℂ} (hz : z ∉ L.lattice) : HasDerivAt (weierstrassZeta L) (−℘[L] z) z

theorem **deriv_weierstrassZeta** {z : ℂ} (hz : z ∉ L.lattice) : deriv (weierstrassZeta L) z = −℘[L] z

theorem **analyticOnNhd_weierstrassZeta** : AnalyticOnNhd ℂ (weierstrassZeta L) (L.lattice : Set ℂ)ᶜ

@[simp] theorem **weierstrassZeta_neg** (z : ℂ) : weierstrassZeta L (−z) = −weierstrassZeta L z

theorem **weierstrassZeta_eq_inv_add** (z : ℂ) : weierstrassZeta L z = 1 / z + weierstrassZetaExcept L 0 z

@[simp] theorem **weierstrassZetaExcept_zero** (l₀ : ℂ) : weierstrassZetaExcept L l₀ 0 = 0

theorem **meromorphic_weierstrassZeta** : Meromorphic (weierstrassZeta L)

theorem **tendsto_sub_mul_weierstrassZeta** {x : ℂ} (hx : x ∈ L.lattice) : Tendsto (fun z => (z − x) * weierstrassZeta L z) (𝓝[≠] x) (𝓝 1)

theorem **quasiPeriod_ω₁** : quasiPeriod L L.ω₁ = 2 * weierstrassZeta L (L.ω₁ / 2)

theorem **quasiPeriod_ω₂** : quasiPeriod L L.ω₂ = 2 * weierstrassZeta L (L.ω₂ / 2)

theorem **eqOn_compl_lattice_of_hasDerivAt** {f g f' : ℂ → ℂ} (hf : ∀ z ∉ L.lattice, HasDerivAt f (f' z) z) (hg : ∀ z ∉ L.lattice, HasDerivAt g (f' z) z) {x : ℂ} (hx : x ∉ L.lattice) (h : f x = g x) {z : ℂ} (hz : z ∉ L.lattice) : f z = g z

theorem **weierstrassZeta_add_of_half_notMem** {w : ℂ} (hw : w ∈ L.lattice) (hw2 : w / 2 ∉ L.lattice) {z : ℂ} (hz : z ∉ L.lattice) : weierstrassZeta L (z + w) = weierstrassZeta L z + 2 * weierstrassZeta L (w / 2)

theorem **weierstrassZeta_add** {w : ℂ} (hw : w ∈ L.lattice) {z : ℂ} (hz : z ∉ L.lattice) : weierstrassZeta L (z + w) = weierstrassZeta L z + quasiPeriod L w

theorem **quasiPeriod_eq_two_mul** {w : ℂ} (hw : w ∈ L.lattice) (hw2 : w / 2 ∉ L.lattice) : quasiPeriod L w = 2 * weierstrassZeta L (w / 2)

theorem **quasiPeriod_eq_weierstrassZetaExcept** {w : ℂ} (hw : w ∈ L.lattice) (h0 : w ≠ 0) : quasiPeriod L w = weierstrassZetaExcept L w w + 2 / w

theorem **isHomothetic_scalePeriods** (c : ℂ) (hc : c ≠ 0) : IsHomothetic c L (scalePeriods L c hc)

theorem **isHomothetic_one_iff** {L L' : PeriodPair} : IsHomothetic 1 L L' ↔ L'.lattice = L.lattice

theorem **ne_zero** (h : IsHomothetic c L L') : c ≠ 0

@[simp] theorem **coe_equiv** (h : IsHomothetic c L L') (x : L.lattice) : (h.equiv x : ℂ) = c * x

theorem **weierstrassP_mul** (h : IsHomothetic c L L') (z : ℂ) : ℘[L'] (c * z) = (c ^ 2)⁻¹ * ℘[L] z

theorem **derivWeierstrassP_mul** (h : IsHomothetic c L L') (z : ℂ) : ℘'[L'] (c * z) = (c ^ 3)⁻¹ * ℘'[L] z

theorem **G_eq** (h : IsHomothetic c L L') (n : ℕ) : L'.G n = (c ^ n)⁻¹ * L.G n

theorem **weierstrassZeta_mul** (h : IsHomothetic c L L') (z : ℂ) : weierstrassZeta L' (c * z) = c⁻¹ * weierstrassZeta L z

theorem **quasiPeriod_mul** (h : IsHomothetic c L L') (w : ℂ) : quasiPeriod L' (c * w) = c⁻¹ * quasiPeriod L w

theorem **weierstrassZeta_congr** {L L' : PeriodPair} (h : L'.lattice = L.lattice) : weierstrassZeta L' = weierstrassZeta L

theorem **quasiPeriod_congr** {L L' : PeriodPair} (h : L'.lattice = L.lattice) : quasiPeriod L' = quasiPeriod L

theorem **quasiPeriod_ω₁_eq_eisenstein** (τ : UpperHalfPlane) (hτ : (τ : ℂ) * L.ω₁ = L.ω₂) : quasiPeriod L L.ω₁ = L.ω₁⁻¹ * G2 τ

theorem **lattice_rotatePeriods** : (rotatePeriods L).lattice = L.lattice

theorem **lattice_swapPeriods** : (swapPeriods L).lattice = L.lattice

theorem **legendre_relation** (h : 0 < (L.ω₂ / L.ω₁).im) : quasiPeriod L L.ω₁ * L.ω₂ − quasiPeriod L L.ω₂ * L.ω₁ = 2 * π * I

theorem **latticeArea_pos** : 0 < latticeArea L

theorem **latticeArea_eq** : latticeArea L = |(conj L.ω₁ * L.ω₂).im|

theorem **latticeArea_congr** {L L' : PeriodPair} (h : L'.lattice = L.lattice) : latticeArea L' = latticeArea L

theorem **IsHomothetic.latticeArea_eq** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') : latticeArea L' = ‖c‖ ^ 2 * latticeArea L

theorem **realLinear_eq_add_conj** (f : ℂ →ₗ[ℝ] ℂ) (w : ℂ) : f w = (f 1 − I * f I) / 2 * w + (f 1 + I * f I) / 2 * conj w

theorem **quasiPeriod_eq** (w : ℂ) : quasiPeriod L w = sTwo L * w + (π / latticeArea L : ℝ) * conj w

theorem **sTwo_unique** {s w : ℂ} (hw : w ≠ 0) (h : quasiPeriod L w = s * w + (π / latticeArea L : ℝ) * conj w) : s = sTwo L

theorem **quasiPeriod_mul_sub** (a b : ℂ) : quasiPeriod L a * b - quasiPeriod L b * a = 2 * π * I * ((conj a * b).im / latticeArea L : ℝ)

theorem **IsHomothetic.sTwo_eq** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') : sTwo L' = (c ^ 2)⁻¹ * sTwo L

theorem **sTwo_congr** {L L' : PeriodPair} (h : L'.lattice = L.lattice) : sTwo L' = sTwo L

theorem **sTwo_eq_eisenstein** (τ : UpperHalfPlane) (h1 : L.ω₁ = 1) (h2 : L.ω₂ = τ) : sTwo L = EisensteinSeries.G2 τ - (π / τ.im : ℝ)

@[simp] theorem **primaryFactor_zero** : primaryFactor 0 = 1

theorem **primaryFactor_eq_zero_iff** {w : ℂ} : primaryFactor w = 0 ↔ w = 1

@[fun_prop] theorem **differentiable_primaryFactor** : Differentiable ℂ primaryFactor

theorem **norm_primaryFactor_sub_one_le** {w : ℂ} (hw : ‖w‖ ≤ 1 / 2) : ‖primaryFactor w - 1‖ ≤ 2 * ‖w‖ ^ 3

theorem **sigmaFactor_eq** (l z : ℂ) : sigmaFactor l z = primaryFactor (z / l)

theorem **differentiable_sigmaFactor** (l : ℂ) : Differentiable ℂ (sigmaFactor l)

theorem **sigmaFactor_eq_zero_iff** {l z : ℂ} : sigmaFactor l z = 0 ↔ l ≠ 0 ∧ z = l

theorem **norm_sigmaFactor_sub_one_le** {r : ℝ} (hr : 0 < r) {z : ℂ} (hz : ‖z‖ ≤ r) {l : ℂ} (h : 2 * r ≤ ‖l‖) : ‖sigmaFactor l z - 1‖ ≤ 2 * r ^ 3 * ‖l‖ ^ (-3 : ℝ)

theorem **logDeriv_sigmaFactor** {l z : ℂ} (h : sigmaFactor l z ≠ 0) : logDeriv (sigmaFactor l) z = if l = 0 then 0 else zetaTerm l z

theorem **hasProdLocallyUniformlyOn_sigmaProd** : HasProdLocallyUniformlyOn (fun (l : L.lattice) z => sigmaFactor l z) (sigmaProd L) Set.univ

theorem **hasProd_sigmaProd** (z : ℂ) : HasProd (fun l : L.lattice => sigmaFactor l z) (sigmaProd L z)

theorem **summable_norm_sigmaFactor_sub_one** (z : ℂ) : Summable fun l : L.lattice => ‖sigmaFactor l z - 1‖

@[fun_prop] theorem **differentiable_sigmaProd** : Differentiable ℂ (sigmaProd L)

@[simp] theorem **sigmaProd_zero** : sigmaProd L 0 = 1

@[simp] theorem **sigmaProd_neg** (z : ℂ) : sigmaProd L (-z) = sigmaProd L z

theorem **sigmaProd_ne_zero** {z : ℂ} (hz : z ∈ ((L.lattice : Set ℂ) \ {0})ᶜ) : sigmaProd L z ≠ 0

theorem **logDeriv_sigmaProd** {z : ℂ} (hz : z ∈ ((L.lattice : Set ℂ) \ {0})ᶜ) : logDeriv (sigmaProd L) z = weierstrassZetaExcept L 0 z

@[fun_prop] theorem **differentiable_weierstrassSigma** : Differentiable ℂ (weierstrassSigma L)

@[fun_prop] theorem **analyticAt_weierstrassSigma** (z : ℂ) : AnalyticAt ℂ (weierstrassSigma L) z

@[fun_prop] theorem **continuous_weierstrassSigma** : Continuous (weierstrassSigma L)

@[simp] theorem **weierstrassSigma_zero** : weierstrassSigma L 0 = 0

@[simp] theorem **weierstrassSigma_neg** (z : ℂ) : weierstrassSigma L (-z) = -weierstrassSigma L z

theorem **hasDerivAt_weierstrassSigma_zero** : HasDerivAt (weierstrassSigma L) 1 0

theorem **weierstrassSigma_ne_zero** {z : ℂ} (hz : z ∉ L.lattice) : weierstrassSigma L z ≠ 0

theorem **hasDerivAt_weierstrassSigma** {z : ℂ} (hz : z ∉ L.lattice) : HasDerivAt (weierstrassSigma L) (weierstrassZeta L z * weierstrassSigma L z) z

theorem **logDeriv_weierstrassSigma** {z : ℂ} (hz : z ∉ L.lattice) : logDeriv (weierstrassSigma L) z = weierstrassZeta L z

theorem **IsHomothetic.sigmaProd_mul** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') (z : ℂ) : sigmaProd L' (c * z) = sigmaProd L z

theorem **IsHomothetic.weierstrassSigma_mul** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') (z : ℂ) : weierstrassSigma L' (c * z) = c * weierstrassSigma L z

theorem **weierstrassSigma_congr** {L L' : PeriodPair} (h : L'.lattice = L.lattice) : weierstrassSigma L' = weierstrassSigma L

theorem **sigmaSign_of_mem** {w : ℂ} (h : w / 2 ∈ L.lattice) : sigmaSign L w = 1

theorem **sigmaSign_of_notMem** {w : ℂ} (h : w / 2 ∉ L.lattice) : sigmaSign L w = -1

@[simp] theorem **sigmaSign_sq** (w : ℂ) : sigmaSign L w ^ 2 = 1

@[simp] theorem **norm_sigmaSign** (w : ℂ) : ‖sigmaSign L w‖ = 1

theorem **sigmaSign_ne_zero** (w : ℂ) : sigmaSign L w ≠ 0

@[simp] theorem **coe_sigmaSignCircle** (w : ℂ) : (sigmaSignCircle L w : ℂ) = sigmaSign L w

@[simp] theorem **sigmaSign_zero** : sigmaSign L 0 = 1

@[simp] theorem **sigmaSign_neg** (w : ℂ) : sigmaSign L (-w) = sigmaSign L w

theorem **IsHomothetic.sigmaSign_mul** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') (w : ℂ) : sigmaSign L' (c * w) = sigmaSign L w

theorem **sigmaSign_int** (m n : ℤ) : sigmaSign L (m * L.ω₁ + n * L.ω₂) = (-1) ^ (m + n + m * n)

theorem **exists_weierstrassSigma_add** {w : ℂ} (hw : w ∈ L.lattice) : ∃ C : ℂ, C ≠ 0 ∧ ∀ z, weierstrassSigma L (z + w) = C * cexp (quasiPeriod L w * (z + w / 2)) * weierstrassSigma L z

theorem **weierstrassSigma_eq_zero_iff** {z : ℂ} : weierstrassSigma L z = 0 ↔ z ∈ L.lattice

theorem **deriv_weierstrassSigma_add_of_law** {w C η x : ℂ} (h : ∀ z, weierstrassSigma L (z + w) = C * cexp (η * (z + w / 2)) * weierstrassSigma L z) (hx : weierstrassSigma L x = 0) : deriv (weierstrassSigma L) (x + w) = C * cexp (η * (x + w / 2)) * deriv (weierstrassSigma L) x

theorem **deriv_weierstrassSigma_ne_zero** {w : ℂ} (hw : w ∈ L.lattice) : deriv (weierstrassSigma L) w ≠ 0

theorem **deriv_weierstrassSigma_neg** (z : ℂ) : deriv (weierstrassSigma L) (-z) = deriv (weierstrassSigma L) z

theorem **weierstrassSigma_add** {w : ℂ} (hw : w ∈ L.lattice) (z : ℂ) : weierstrassSigma L (z + w) = sigmaSign L w * cexp (quasiPeriod L w * (z + w / 2)) * weierstrassSigma L z

theorem **deriv_weierstrassSigma_of_mem** {w : ℂ} (hw : w ∈ L.lattice) : deriv (weierstrassSigma L) w = sigmaSign L w * cexp (quasiPeriod L w * (w / 2))

theorem **analyticOrderAt_weierstrassSigma_of_mem** {w : ℂ} (hw : w ∈ L.lattice) : analyticOrderAt (weierstrassSigma L) w = 1

theorem **analyticOrderAt_weierstrassSigma_of_notMem** {z : ℂ} (hz : z ∉ L.lattice) : analyticOrderAt (weierstrassSigma L) z = 0

theorem **latticeForm_add_left** (u u' v : ℂ) : latticeForm L (u + u') v = latticeForm L u v + latticeForm L u' v

theorem **latticeForm_add_right** (u v v' : ℂ) : latticeForm L u (v + v') = latticeForm L u v + latticeForm L u v'

theorem **latticeForm_mul_left** (a u v : ℂ) : latticeForm L (a * u) v = a * latticeForm L u v

theorem **latticeForm_mul_right** (a u v : ℂ) : latticeForm L u (a * v) = conj a * latticeForm L u v

theorem **latticeForm_eq_inner** (u v : ℂ) : latticeForm L u v = inner ℂ v u / latticeArea L

theorem **conj_latticeForm** (u v : ℂ) : conj (latticeForm L u v) = latticeForm L v u

theorem **latticeForm_self** (u : ℂ) : latticeForm L u u = (‖u‖ ^ 2 / latticeArea L : ℝ)

theorem **im_latticeForm_int** (m n m' n' : ℤ) : (latticeForm L (m * L.ω₁ + n * L.ω₂) (m' * L.ω₁ + n' * L.ω₂)).im = (n * m' - m * n' : ℤ) * (latticeForm L L.ω₂ L.ω₁).im

theorem **abs_im_latticeForm** : |(latticeForm L L.ω₂ L.ω₁).im| = 1

theorem **im_latticeForm_eq_one** (h : 0 < (L.ω₂ / L.ω₁).im) : (latticeForm L L.ω₂ L.ω₁).im = 1

theorem **exists_int_im_latticeForm** {a b : ℂ} (ha : a ∈ L.lattice) (hb : b ∈ L.lattice) : ∃ k : ℤ, (latticeForm L a b).im = k

theorem **IsHomothetic.latticeForm_mul** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') (u v : ℂ) : latticeForm L' (c * u) (c * v) = latticeForm L u v

theorem **quasiPeriod_mul_eq** (z w : ℂ) : quasiPeriod L w * z = sTwo L * w * z + π * latticeForm L z w

@[fun_prop] theorem **differentiable_canonicalSigma** : Differentiable ℂ (canonicalSigma L)

@[simp] theorem **canonicalSigma_neg** (z : ℂ) : canonicalSigma L (-z) = -canonicalSigma L z

@[simp] theorem **canonicalSigma_zero** : canonicalSigma L 0 = 0

theorem **canonicalSigma_eq_zero_iff** {z : ℂ} : canonicalSigma L z = 0 ↔ z ∈ L.lattice

theorem **hasDerivAt_canonicalSigma'** (z : ℂ) : HasDerivAt (canonicalSigma L) ((deriv (weierstrassSigma L) z - sTwo L * z * weierstrassSigma L z) * cexp (-(sTwo L * z ^ 2 / 2))) z

theorem **hasDerivAt_canonicalSigma_zero** : HasDerivAt (canonicalSigma L) 1 0

theorem **hasDerivAt_canonicalSigma** {z : ℂ} (hz : z ∉ L.lattice) : HasDerivAt (canonicalSigma L) ((weierstrassZeta L z - sTwo L * z) * canonicalSigma L z) z

theorem **analyticOrderAt_canonicalSigma_of_mem** {w : ℂ} (hw : w ∈ L.lattice) : analyticOrderAt (canonicalSigma L) w = 1

theorem **IsHomothetic.canonicalSigma_mul** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') (z : ℂ) : canonicalSigma L' (c * z) = c * canonicalSigma L z

theorem **canonicalSigma_add** {w : ℂ} (hw : w ∈ L.lattice) (z : ℂ) : canonicalSigma L (z + w) = sigmaSign L w * cexp (π * latticeForm L z w + π * latticeForm L w w / 2) * canonicalSigma L z

theorem **sigmaSign_add** {a b : ℂ} (ha : a ∈ L.lattice) (hb : b ∈ L.lattice) : sigmaSign L (a + b) = sigmaSign L a * sigmaSign L b * cexp (π * I * (latticeForm L a b).im)

theorem **exists_eq_add_sq_mul_of_even** {f : ℂ → ℂ} (hf : AnalyticAt ℂ f 0) (he : ∀ z, f (-z) = f z) : ∃ h : ℂ → ℂ, AnalyticAt ℂ h 0 ∧ ∀ z, f z = f 0 + z ^ 2 * h z

theorem **eq_const_of_periodic_of_tendsto** {f : ℂ → ℂ} {c : ℂ} (hf : DifferentiableOn ℂ f (L.lattice : Set ℂ)ᶜ) (hper : ∀ z ∉ L.lattice, ∀ w ∈ L.lattice, f (z + w) = f z) (hc : Tendsto f (𝓝[≠] 0) (𝓝 c)) {z : ℂ} (hz : z ∉ L.lattice) : f z = c

theorem **exists_eq_const_of_periodic_of_even** {f N : ℂ → ℂ} (hf : DifferentiableOn ℂ f (L.lattice : Set ℂ)ᶜ) (hper : ∀ z ∉ L.lattice, ∀ w ∈ L.lattice, f (z + w) = f z) (hN : AnalyticAt ℂ N 0) (he : ∀ z, N (-z) = N z) (h0 : N 0 = 0) (hfN : ∀ z ∉ L.lattice, z ^ 2 * f z = N z) : ∃ c, ∀ z ∉ L.lattice, f z = c

theorem **weierstrassP_sub_weierstrassP** {u v : ℂ} (hu : u ∉ L.lattice) (hv : v ∉ L.lattice) : ℘[L] u - ℘[L] v = -(weierstrassSigma L (u + v) * weierstrassSigma L (u - v)) / (weierstrassSigma L u ^ 2 * weierstrassSigma L v ^ 2)

theorem **weierstrassP_sub_weierstrassP_canonicalSigma** {u v : ℂ} (hu : u ∉ L.lattice) (hv : v ∉ L.lattice) : ℘[L] u - ℘[L] v = -(canonicalSigma L (u + v) * canonicalSigma L (u - v)) / (canonicalSigma L u ^ 2 * canonicalSigma L v ^ 2)

theorem **exists_canonicalSigma_eq** : ∃ h : ℂ → ℂ, AnalyticAt ℂ h 0 ∧ ∀ z, canonicalSigma L z = z + z ^ 3 * h z

theorem **canonicalSigma_sub_isBigO** : (fun z => canonicalSigma L z - z) =O[𝓝 0] fun z : ℂ => z ^ 3

theorem **exists_eq_const_of_periodic_of_simple_pole** {f N : ℂ → ℂ} (hf : DifferentiableOn ℂ f (L.lattice : Set ℂ)ᶜ) (hper : ∀ z ∉ L.lattice, ∀ w ∈ L.lattice, f (z + w) = f z) (hN : AnalyticAt ℂ N 0) (hfN : ∀ᶠ z in 𝓝[≠] 0, z * f z = N z) : ∃ c, ∀ z ∉ L.lattice, f z = c

theorem **exists_eq_mul_canonicalSigma** {F : ℂ → ℂ} (hF : Differentiable ℂ F) (hlaw : ∀ w ∈ L.lattice, ∀ z, F (z + w) = sigmaSign L w * cexp (π * latticeForm L z w + π * latticeForm L w w / 2) * F z) : ∃ c : ℂ, ∀ z, F z = c * canonicalSigma L z

theorem **latticeArea_periodsOfUpperHalfPlane** (τ : UpperHalfPlane) : latticeArea (periodsOfUpperHalfPlane τ) = τ.im

theorem **isHomothetic_I_gaussian** : IsHomothetic I (periodsOfUpperHalfPlane UpperHalfPlane.I) (periodsOfUpperHalfPlane UpperHalfPlane.I)

theorem **sTwo_gaussian** : sTwo (periodsOfUpperHalfPlane UpperHalfPlane.I) = 0

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.InvariantDifferentialMulEqDerivX

theorem **w_eq** : wSer W = X ^ 3 + C W.a₁ * X * wSer W + C W.a₂ * X ^ 2 * wSer W + C W.a₃ * wSer W ^ 2 + C W.a₄ * X * wSer W ^ 2 + C W.a₆ * wSer W ^ 3

theorem **w_eq_X_pow_mul** : wSer W = X ^ 3 * wU W ∧ constantCoeff (wU W) = 1

theorem **wU_eq** : wU W = 1 + C W.a₁ * X * wU W + C W.a₂ * X ^ 2 * wU W + C W.a₃ * X ^ 3 * wU W ^ 2 + C W.a₄ * X ^ 4 * wU W ^ 2 + C W.a₆ * X ^ 6 * wU W ^ 3

theorem **coeff_one_wU** : coeff 1 (wU W) = W.a₁

theorem **fw_mul_om** : fw W * om W = 1

theorem **coeff_inv** : constantCoeff (iSer W) = 0 ∧ coeff 1 (iSer W) = -1

theorem **inv_mul_den** : iSer W * (-1 + C W.a₁ * X + C W.a₃ * wSer W) = X

theorem **w_inv_mul_den** : (wSer W).subst (iSer W) * (-1 + C W.a₁ * X + C W.a₃ * wSer W) = wSer W

theorem **map_inv** : PowerSeries.map φ (iSer W) = iSer (W.map φ)

@[simp] theorem **formalGroup_toPowerSeries** : (formalGroup W).toPowerSeries = fgl W

theorem **chord** (i : Fin 2) : (wSer W).subst (X i : MvPowerSeries (Fin 2) R) = lam W * X i + nu W

theorem **third_point** : constantCoeff (uThird W) = 0 ∧ (wSer W).subst (uThird W) = lam W * uThird W + nu W ∧ cThree W * (X 0 + X 1 + uThird W) + cTwo W = 0 ∧ (formalGroup W).toPowerSeries = (iSer W).subst (uThird W)

theorem **subst_X_inv** : (formalGroup W).toPowerSeries.subst (![PowerSeries.X, iSer W] : Fin 2 → PowerSeries R) = 0

theorem **subst_inv** {s : MvPowerSeries σ R} (hs : PowerSeries.HasSubst s) : (formalGroup W).toPowerSeries.subst ![s, (iSer W).subst s] = 0

@[simp] theorem **neg_val** (s : (formalGroup W).Point σ) : (-s).val = (iSer W).subst s.val

theorem **map_zsmul** (φ : R →+* S) (n : ℤ) : PowerSeries.map φ (zSer W n) = zSer (W.map φ) n

theorem **hasSubst_twoPoints** : HasSubst (fun j : Fin 2 => (X (j, 0) : PS R 2 1))

theorem **ev_toTwoPoints** (F : MvPowerSeries (Fin 2) R) {n : ℕ} {a b : Pt R n 1} (ha : constantCoeff (a 0) = 0) (hb : constantCoeff (b 0) = 0) : ev ![a, b] (toTwoPoints F) = F.subst ![a 0, b 0]

theorem **map_substAlgHom** {τ : Type*} {a : τ → MvPowerSeries σ R} (ha : HasSubst a) : (seriesCurve W τ).map (substAlgHom (R

theorem **substAlgHom_comp_point** {τ : Type*} {a : τ → MvPowerSeries σ R} (ha : HasSubst a) {s : MvPowerSeries τ R} (hs : PowerSeries.HasSubst s) : (substAlgHom (R

theorem **equation_point** {s : MvPowerSeries σ R} (hs : PowerSeries.HasSubst s) : (seriesCurve W σ).Equation (point W s)

theorem **neg_point** {s : MvPowerSeries σ R} (hs : PowerSeries.HasSubst s) : (seriesCurve W σ).neg (point W s) = (-1 + C W.a₁ * s + C W.a₃ * (wSer W).subst s) • point W ((iSer W).subst s)

theorem **neg_eq_smul_point** {A N Z : MvPowerSeries (Fin 2) R} (hZ : Z = lam W * A - nu W * N) (hA : (-cTwo W - cThree W * (X 0 + X 1)) * N + cThree W * A = 0) : ![A, (seriesCurve W (Fin 2)).negY ![A, N, Z], Z] = (-N * (-1 + C W.a₁ * uThird W + C W.a₃ * wThird W)) • point W (fgl W)

theorem **originN_eq** : originN (seriesCurve W (Fin 2)) (X 0) ((wSer W).subst (X 0 : MvPowerSeries (Fin 2) R)) (X 1) = lam W * originD (seriesCurve W (Fin 2)) ((wSer W).subst (X 0 : MvPowerSeries (Fin 2) R)) (X 1) ((wSer W).subst (X 0 : MvPowerSeries (Fin 2) R) + lam W * (X 1 - X 0))

theorem **addXYZ_diagXYZ_point_X** : (∃ c : MvPowerSeries (Fin 2) R, (seriesCurve W (Fin 2)).addXYZ (point W (X 0)) (point W (X 1)) = c • point W (fgl W)) ∧ ∃ c : MvPowerSeries (Fin 2) R, IsUnit c ∧ diagXYZ (seriesCurve W (Fin 2)) (point W (X 0)) (point W (X 1)) = c • point W (fgl W)

theorem **addXYZ_diagXYZ_point** {s t : MvPowerSeries σ R} (hs : PowerSeries.HasSubst s) (ht : PowerSeries.HasSubst t) : (∃ c : MvPowerSeries σ R, (seriesCurve W σ).addXYZ (point W s) (point W t) = c • point W ((formalGroup W).toPowerSeries.subst ![s, t])) ∧ ∃ c : MvPowerSeries σ R, IsUnit c ∧ diagXYZ (seriesCurve W σ) (point W s) (point W t) = c • point W ((formalGroup W).toPowerSeries.subst ![s, t])

theorem **nonsingular_point** {s : MvPowerSeries σ R} (hs : PowerSeries.HasSubst s) : ((seriesCurve W σ).map φ).Nonsingular (φ ∘ point W s)

theorem **point_subst_equiv_add** {s t : MvPowerSeries σ R} (hs : PowerSeries.HasSubst s) (ht : PowerSeries.HasSubst t) : φ ∘ point W ((formalGroup W).toPowerSeries.subst ![s, t]) ≈ ((seriesCurve W σ).map φ).add (φ ∘ point W s) (φ ∘ point W t)

theorem **formalGroup_singular** (a : R) : (formalGroup ((a, 0, 0, 0, 0) : WeierstrassCurve R)).toPowerSeries = X 0 + X 1 - C a * X 0 * X 1

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.IsAddFundamentalDomainLatticeDomain

theorem **latticeDomain_isBounded** : Bornology.IsBounded (latticeDomain L)

theorem **existsUnique_add_mem_latticeDomain** (x : E) : ∃! a : L, x + a ∈ latticeDomain L

theorem **measurableSet_latticeDomain** : MeasurableSet (latticeDomain L)

theorem **isAddFundamentalDomain_latticeDomain** (μ : Measure E) : IsAddFundamentalDomain L (latticeDomain L) μ

theorem **measure_latticeDomain** (μ : Measure E) [μ.IsAddHaarMeasure] : μ (latticeDomain L) = ENNReal.ofReal (ZLattice.covolume L μ)

theorem **latticeProb_eq** (μ : Measure E) [μ.IsAddHaarMeasure] : latticeProb L = (ENNReal.ofReal (ZLattice.covolume L μ))⁻¹ • μ.restrict (latticeDomain L)

theorem **integrable_latticeProb** {f : E → G} (hf : Continuous f) : Integrable f (latticeProb L)

theorem **latticeMean_eq_integral** [m : MeasurableSpace E] [hm : BorelSpace E] (f : E → G) : latticeMean L f = ∫ x, f x ∂(latticeProb L)

theorem **latticeMean_eq_latticeDomain** [MeasurableSpace E] [BorelSpace E] (μ : Measure E) [μ.IsAddHaarMeasure] (f : E → G) : latticeMean L f = (ZLattice.covolume L μ)⁻¹ • ∫ x in latticeDomain L, f x ∂μ

theorem **latticeMean_eq** [MeasurableSpace E] [BorelSpace E] (μ : Measure E) [μ.IsAddHaarMeasure] {D : Set E} (hD : IsAddFundamentalDomain L D μ) {f : E → G} (hf : ∀ a ∈ L, ∀ x, f (x + a) = f x) : latticeMean L f = (ZLattice.covolume L μ)⁻¹ • ∫ x in D, f x ∂μ

@[simp] theorem **latticeMean_const** [CompleteSpace G] (c : G) : latticeMean L (fun _ => c) = c

theorem **latticeMean_finset_sum** {ι : Type*} (s : Finset ι) {f : ι → E → G} (hf : ∀ i ∈ s, Continuous (f i)) : latticeMean L (fun x => ∑ i ∈ s, f i x) = ∑ i ∈ s, latticeMean L (f i)

theorem **latticeMean_const_mul** {𝕜 : Type*} [RCLike 𝕜] (c : 𝕜) (f : E → 𝕜) : latticeMean L (fun x => c * f x) = c * latticeMean L f

theorem **latticeMean_nonneg** {f : E → ℝ} (hf : ∀ x, 0 ≤ f x) : 0 ≤ latticeMean L f

theorem **latticeMean_comp_add** {f : E → G} (hf : ∀ a ∈ L, ∀ x, f (x + a) = f x) (w : E) : latticeMean L (fun x => f (x + w)) = latticeMean L f

theorem **eq_zero_of_latticeMean_eq_zero** {f : E → ℝ} (hc : Continuous f) (h0 : ∀ x, 0 ≤ f x) (hf : ∀ a ∈ L, ∀ x, f (x + a) = f x) (h : latticeMean L f = 0) : f = 0

theorem **exists_forall_le_of_periodic** {f : E → ℝ} (hc : Continuous f) (hf : ∀ a ∈ L, ∀ x, f (x + a) = f x) : ∃ x₀, ∀ x, f x ≤ f x₀

theorem **latticeMean_eq_of_periodic** (L' : Submodule ℤ E) [DiscreteTopology L'] [IsZLattice ℝ L'] [CompleteSpace G] {f : E → G} (hc : Continuous f) (hf : ∀ a ∈ L, ∀ x, f (x + a) = f x) (hf' : ∀ a ∈ L', ∀ x, f (x + a) = f x) : latticeMean L f = latticeMean L' f

theorem **latticeMean_comp_equiv** {E' : Type*} [NormedAddCommGroup E'] [NormedSpace ℝ E'] [FiniteDimensional ℝ E'] (L' : Submodule ℤ E') [DiscreteTopology L'] [IsZLattice ℝ L'] (e : E ≃L[ℝ] E') (he : ∀ x, e x ∈ L' ↔ x ∈ L) {f : E' → G} (hf : ∀ a ∈ L', ∀ x, f (x + a) = f x) : latticeMean L (fun x => f (e x)) = latticeMean L' f

@[simp] theorem **mem_piLpLattice** {x : PiLp 2 E} : x ∈ piLpLattice L ↔ ∀ i, x i ∈ L i

@[simp] theorem **mem_prodLpLattice** {x : WithLp 2 (E × E')} : x ∈ prodLpLattice L L' ↔ x.fst ∈ L ∧ x.snd ∈ L'

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.IsAmpleLBOfRelAmpleOfIsAffine

theorem **isGenerated_of_finite** {ι : Type*} [Finite ι] (s : ι → L.Sec) (h : ⊔ i, nonzeroLocus (s i) = ⊤) : IsGenerated L

theorem **IsGenerated.exists_mem** (h : IsGenerated L) (x : X) : ∃ s : L.Sec, x ∈ nonzeroLocus s

theorem **exists_finset_iSup_eq_top** [CompactSpace X] (U : X → X.Opens) (hU : ∀ x, x ∈ U x) : ∃ T : Finset X, ⊔ x : T, U x = ⊤

theorem **isGenerated_iff_forall** [CompactSpace X] : IsGenerated L ↔ ∀ x : X, ∃ s : L.Sec, x ∈ nonzeroLocus s

theorem **IsGenerated.exists_fin_succ** (h : IsGenerated L) : ∃ (N : ℕ) (s : Fin (N + 1) → L.Sec), ⊔ i, nonzeroLocus (s i) = ⊤

theorem **IsGenerated.isLB** (h : IsGenerated L) : L.IsLB

theorem **isGenerated_of_gen** (hL : L.IsLB) (t : L.Gen) (ht : t.IsGen) : IsGenerated L

theorem **isGenerated_triv** (X : Scheme.{0}) : IsGenerated (LB.triv X)

theorem **IsGenerated.pull** (h : IsGenerated L) (f : Y ⟶ X) : IsGenerated (L.pull f)

theorem **IsGenerated.of_lbIso** (h : IsGenerated M) (hL : L.IsLB) (H : LBIso L M) : IsGenerated L

theorem **IsGenerated.pow** (h : IsGenerated L) (k : ℕ) : IsGenerated (lbPow L k)

theorem **IsGenerated.pow_add** {m n : ℕ} (hm : IsGenerated (lbPow L m)) (hn : IsGenerated (lbPow L n)) : IsGenerated (lbPow L (m + n))

theorem **isGenerated_of_isQuasiAffine** [X.IsQuasiAffine] (hL : L.IsLB) : IsGenerated L

theorem **exists_common_pow** {ι : Type*} [Finite ι] (n : ι → ℕ) (hn : ∀ i, 1 ≤ n i) (s : ∀ i, (lbPow L (n i)).Sec) : ∃ N, 1 ≤ N ∧ ∃ t : ι → (lbPow L N).Sec, ∀ i, nonzeroLocus (t i) = nonzeroLocus (s i)

theorem **IsAmpleLB.exists_cover** (h : IsAmpleLB L) : ∃ n, 1 ≤ n ∧ ∃ (N : ℕ) (s : Fin (N + 1) → (lbPow L n).Sec), (⊔ i, nonzeroLocus (s i) = ⊤) ∧ ∀ i, IsAffineOpen (nonzeroLocus (s i))

theorem **IsAmpleLB.exists_cover_mul** (h : IsAmpleLB L) : ∃ n, 1 ≤ n ∧ ∀ k, 1 ≤ k → ∃ (N : ℕ) (s : Fin (N + 1) → (lbPow L (n * k)).Sec), (⊔ i, nonzeroLocus (s i) = ⊤) ∧ ∀ i, IsAffineOpen (nonzeroLocus (s i))

theorem **isAmpleLB_of_cover** (hL : L.IsLB) {n : ℕ} (hn : 1 ≤ n) {ι : Type*} [Finite ι] (s : ι → (lbPow L n).Sec) (hs : ⊔ i, nonzeroLocus (s i) = ⊤) (ha : ∀ i, IsAffineOpen (nonzeroLocus (s i))) : IsAmpleLB L where

theorem **isAmpleLB_iff_cover** : IsAmpleLB L ↔ L.IsLB ∧ ∃ n, 1 ≤ n ∧ ∃ (N : ℕ) (s : Fin (N + 1) → (lbPow L n).Sec), (⊔ i, nonzeroLocus (s i) = ⊤) ∧ ∀ i, IsAffineOpen (nonzeroLocus (s i))

theorem **IsAmpleLB.exists_isGenerated_pow** (h : IsAmpleLB L) : ∃ n, 1 ≤ n ∧ IsGenerated (lbPow L n)

theorem **IsAmpleLB.quasiSeparatedSpace** (h : IsAmpleLB L) : QuasiSeparatedSpace X

theorem **IsAmpleLB.exists_affine_le** (h : IsAmpleLB L) {x : X} {U : X.Opens} (hx : x ∈ U) : ∃ n, 1 ≤ n ∧ ∃ s : (lbPow L n).Sec, x ∈ nonzeroLocus s ∧ nonzeroLocus s ≤ U ∧ IsAffineOpen (nonzeroLocus s)

theorem **isAmpleLB_of_forall_le** [CompactSpace X] (hL : L.IsLB) (h : ∀ (x : X) (U : X.Opens), x ∈ U → ∃ n, 1 ≤ n ∧ ∃ s : (lbPow L n).Sec, x ∈ nonzeroLocus s ∧ nonzeroLocus s ≤ U) : IsAmpleLB L

theorem **IsAmpleLB.of_lbIso** (h : IsAmpleLB M) (hL : L.IsLB) (H : LBIso L M) : IsAmpleLB L

theorem **isAmpleLB_pow_iff** (hL : L.IsLB) {k : ℕ} (hk : 1 ≤ k) : IsAmpleLB (lbPow L k) ↔ IsAmpleLB L

theorem **IsAmpleLB.pull_of_isAffineHom** (h : IsAmpleLB L) (f : Y ⟶ X) [IsAffineHom f] : IsAmpleLB (L.pull f)

theorem **IsAmpleLB.pull_of_isOpenImmersion** (h : IsAmpleLB L) (f : Y ⟶ X) [IsOpenImmersion f] [CompactSpace Y] : IsAmpleLB (L.pull f)

theorem **isAmpleLB_of_relAmple** {S : Scheme.{0}} (A : Over S) {L : LB A.left} (hL : L.IsLB) (h : RelAmple A L) (U : S.Opens) (hU : IsAffineOpen U) : IsAmpleLB (L.pull (A.hom ⁻¹ᵁ U).ι)

theorem **lbIso_pull_id** (hL : L.IsLB) : LBIso (L.pull (𝟙 X)) L

theorem **isAmpleLB_of_relAmple_of_isAffine** {S : Scheme.{0}} [IsAffine S] (A : Over S) {L : LB A.left} (hL : L.IsLB) (h : RelAmple A L) : IsAmpleLB L

theorem **polAV_isAmpleLB** {S : Scheme.{0}} (P : PolAV S) (U : S.Opens) (hU : IsAffineOpen U) : IsAmpleLB (P.L.pull (P.A.hom ⁻¹ᵁ U).ι)

theorem **isAmpleLB_of_isQuasiAffine** [X.IsQuasiAffine] (hL : L.IsLB) : IsAmpleLB L

theorem **isAmpleLB_of_isAffine** [IsAffine X] (hL : L.IsLB) : IsAmpleLB L

theorem **isAmpleLB_triv_iff** : IsAmpleLB (LB.triv X) ↔ X.IsQuasiAffine

theorem **isGenerated_O1** (R : Type) [CommRing R] (N : ℕ) : IsGenerated (ProjSpace.O1 R N)

theorem **isAmple_projective_O_one** (R : Type) [CommRing R] (N : ℕ) : IsAmpleLB (ProjSpace.O1 R N)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.IsRegularFnIffOfIsAffineOpen

theorem **resG_resG** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (a : Γ(X, U)) : resG X h₂ (resG X h₁ a) = resG X (h₂.trans h₁) a

theorem **resG_self** {U : X.Opens} (h : U ≤ U) (a : Γ(X, U)) : resG X h a = a

theorem **val_res** {U V : X.Opens} (h : V ≤ U) (u : Γ(X, U)ˣ) : ((res h u : Γ(X, V)ˣ) : Γ(X, V)) = resG X h (u : Γ(X, U))

theorem **germ_resG** {U V : X.Opens} (h : V ≤ U) (x : X) (hx : x ∈ V) (a : Γ(X, U)) : (X.presheaf.germ V x hx).hom (resG X h a) = (X.presheaf.germ U x (h hx)).hom a

theorem **ext_germ** {U : X.Opens} {a b : Γ(X, U)} (h : ∀ x (hx : x ∈ U), (X.presheaf.germ U x hx).hom a = (X.presheaf.germ U x hx).hom b) : a = b

theorem **app_resG** (φ : Y ⟶ X) {U V : X.Opens} (h : V ≤ U) (a : Γ(X, U)) : (φ.app V).hom (resG X h a) = resG Y (φ.preimage_mono h) ((φ.app U).hom a)

theorem **mem** (ha : IsRegularFn a) : a ∈ nonZeroDivisors Γ(X, U)

theorem **resG** (ha : IsRegularFn a) {V : X.Opens} (h : V ≤ U) : IsRegularFn (resG X h a)

theorem **mul** (ha : IsRegularFn a) (hb : IsRegularFn b) : IsRegularFn (a * b)

theorem **pow** (ha : IsRegularFn a) (n : ℕ) : IsRegularFn (a ^ n)

theorem **of_isUnit** (ha : IsUnit a) : IsRegularFn a

theorem **of_dvd** (hb : IsRegularFn b) (h : a ∣ b) : IsRegularFn a

theorem **mul_right_cancel** (ha : IsRegularFn a) {c d : Γ(X, U)} (h : c * a = d * a) : c = d

theorem **isRegularFn_iff_germ** {U : X.Opens} {a : Γ(X, U)} : IsRegularFn a ↔ ∀ x (hx : x ∈ U), (X.presheaf.germ U x hx).hom a ∈ nonZeroDivisors (X.presheaf.stalk x)

theorem **isRegularFn_of_locally** {U : X.Opens} {a : Γ(X, U)} (h : ∀ x ∈ U, ∃ (V : X.Opens) (hV : V ≤ U), x ∈ V ∧ IsRegularFn (resG X hV a)) : IsRegularFn a

theorem **isRegularFn_iff_of_isAffineOpen** {U : X.Opens} (hU : IsAffineOpen U) {a : Γ(X, U)} : IsRegularFn a ↔ a ∈ nonZeroDivisors Γ(X, U)

theorem **isRegularFn_of_ne_zero** [IsIntegral X] {U : X.Opens} {a : Γ(X, U)} (ha : a ≠ 0) : IsRegularFn a

theorem **IsRegularFn.ne_zero** {U : X.Opens} [hU : Nonempty U] {a : Γ(X, U)} (ha : IsRegularFn a) : a ≠ 0

theorem **isRegularFn_iff_ne_zero** [IsIntegral X] {U : X.Opens} [Nonempty U] {a : Γ(X, U)} : IsRegularFn a ↔ a ≠ 0

theorem **IsRegularFn.app_of_flat** (φ : Y ⟶ X) [Flat φ] {U : X.Opens} {a : Γ(X, U)} (ha : IsRegularFn a) : IsRegularFn ((φ.app U).hom a)

theorem **IsRegularFn.dvd_of_germ** {U : X.Opens} {a b : Γ(X, U)} (ha : IsRegularFn a) (h : ∀ x (hx : x ∈ U), (X.presheaf.germ U x hx).hom a ∣ (X.presheaf.germ U x hx).hom b) : a ∣ b

theorem **IsRegularFn.associated_of_dvd_dvd** {U : X.Opens} {a b : Γ(X, U)} (ha : IsRegularFn a) (h₁ : a ∣ b) (h₂ : b ∣ a) : Associated a b

theorem **IsRegularFn.associated_of_germ** {U : X.Opens} {a b : Γ(X, U)} (ha : IsRegularFn a) (hb : IsRegularFn b) (h : ∀ x (hx : x ∈ U), Ideal.span {(X.presheaf.germ U x hx).hom a} = Ideal.span {(X.presheaf.germ U x hx).hom b}) : Associated a b

theorem **sec_ext** {L : LB X} {s t : L.Sec} (h : ∀ i, s.f i = t.f i) : s = t

theorem **sec_compat** {L : LB X} (s : L.Sec) (i j : L.ι) {W : X.Opens} (hi : W ≤ L.U i) (hj : W ≤ L.U j) : resG X hi (s.f i) = resG X (le_inf hi hj) (L.g i j : Γ(X, L.U i ⊓ L.U j)) * resG X hj (s.f j)

theorem **SecMaps.on** {A B : LB X} {w : LB.Iso A B} {a : A.Sec} {b : B.Sec} (h : SecMaps w a b) (p : B.ι) (q : A.ι) {W : X.Opens} (hp : W ≤ B.U p) (hq : W ≤ A.U q) : resG X hp (b.f p) = resG X (le_inf hp hq) (@id (Γ(X, B.U p ⊓ A.U q)ˣ) (w.s (p, q)) : Γ(X, B.U p ⊓ A.U q)) * resG X hq (a.f q)

theorem **secMaps_genSec_iff** {A B : LB X} (w : LB.Iso A B) (a : A.Gen) (ha : a.IsGen) (b : B.Gen) (hb : b.IsGen) : SecMaps w ⟨fun i => (a.s i).val, fun i j => congrArg Units.val (ha i j)⟩ ⟨fun i => (b.s i).val, fun i j => congrArg Units.val (hb i j)⟩ ↔ LB.Gen.Maps w a b

theorem **SecMaps.isGen** {A B : LB X} {w : LB.Iso A B} {a : A.Sec} {b : B.Sec} (h : SecMaps w a b) (ha : IsRegularSec a) : w.IsGen

theorem **SecMaps.unique** {A B : LB X} {w w' : LB.Iso A B} {a : A.Sec} {b : B.Sec} (h : SecMaps w a b) (h' : SecMaps w' a b) (ha : IsRegularSec a) : w = w'

theorem **SecMaps.eq** {A B : LB X} (hA : A.IsLB) {w : LB.Iso A B} {a : A.Sec} {b b' : B.Sec} (h : SecMaps w a b) (h' : SecMaps w a b') : b = b'

theorem **SecMaps.isRegularSec** {A B : LB X} (hA : A.IsLB) {w : LB.Iso A B} {a : A.Sec} {b : B.Sec} (h : SecMaps w a b) (ha : IsRegularSec a) : IsRegularSec b

theorem **existsUnique_secMaps** {A B : LB X} {a : A.Sec} {b : B.Sec} (ha : IsRegularSec a) (h : ∀ (p : B.ι) (q : A.ι), Associated (resG X (inf_le_right : B.U p ⊓ A.U q ≤ A.U q) (a.f q)) (resG X (inf_le_left : B.U p ⊓ A.U q ≤ B.U p) (b.f p))) : ∃! w : LB.Iso A B, w.IsGen ∧ SecMaps w a b

theorem **mem_nonzeroLocus_iff** {L : LB X} (s : L.Sec) (i : L.ι) {x : X} (hx : x ∈ L.U i) : x ∈ Out.nonzeroLocus s ↔ x ∈ X.basicOpen (s.f i)

theorem **SecMaps.nonzeroLocus_eq** {A B : LB X} (hA : A.IsLB) (hB : B.IsLB) {w : LB.Iso A B} {a : A.Sec} {b : B.Sec} (h : SecMaps w a b) : Out.nonzeroLocus a = Out.nonzeroLocus b

theorem **isRegularSec_of_ne_zero** [IsIntegral X] {L : LB X} (s : L.Sec) {i : L.ι} (hi : s.f i ≠ 0) : IsRegularSec s

theorem **pull_U** (L : LB X) (φ : Y ⟶ X) (j : PUnit → L.ι) : (L.pull φ).U j = φ ⁻¹ᵁ L.U (j ⟨⟩)

theorem **pull_g** (L : LB X) (φ : Y ⟶ X) (j j' : PUnit → L.ι) : (L.pull φ).g j j' = res (LB.wchart_inter_le L (fun _ => φ) j j' ⟨⟩) (pullU φ _ (L.g (j ⟨⟩) (j' ⟨⟩)))

theorem **nonzeroLocus_secPull** {L : LB X} (s : L.Sec) (φ : Y ⟶ X) : Out.nonzeroLocus (secPull s φ) = φ ⁻¹ᵁ Out.nonzeroLocus s

theorem **lbPow_U** (L : LB X) (n : ℕ) (j : PUnit → L.ι) : (lbPow L n).U j = L.U (j ⟨⟩)

theorem **lbPow_g** (L : LB X) (n : ℕ) (j j' : PUnit → L.ι) : (lbPow L n).g j j' = res (inf_le_inf (lbPow_U L n j).le (lbPow_U L n j').le) (L.g (j ⟨⟩) (j' ⟨⟩)) ^ n

theorem **IsRegularSec.secPow** {L : LB X} {s : L.Sec} (hs : IsRegularSec s) (n : ℕ) : IsRegularSec (secPow s n)

theorem **genericPoint_mem** (W : X.Opens) [h : Nonempty W] : genericPoint X ∈ W

theorem **germK_resG** {U V : X.Opens} [Nonempty U] [Nonempty V] (h : V ≤ U) (a : Γ(X, U)) : (X.germToFunctionField V).hom (resG X h a) = (X.germToFunctionField U).hom a

theorem **ext_germK** {U : X.Opens} {a b : Γ(X, U)} (h : ∀ hU : genericPoint X ∈ U, (X.presheaf.germ U _ hU).hom a = (X.presheaf.germ U _ hU).hom b) : a = b

theorem **exists_germK_eq_of_stalk** (U : X.Opens) [Nonempty U] (t : X.functionField) (h : ∀ x ∈ U, t ∈ Set.range (algebraMap (X.presheaf.stalk x) X.functionField)) : ∃ a : Γ(X, U), (X.germToFunctionField U).hom a = t

theorem **mem_ratSecs_iff_stalk** {L : LB X} (s : L.Sec) (V : X.Opens) [Nonempty V] (h : X.functionField) : h ∈ ratSecs s V ↔ ∀ (i : L.ι) (x : X) (_ : x ∈ V) (hx : x ∈ L.U i), algebraMap (X.presheaf.stalk x) X.functionField ((X.presheaf.germ _ x hx).hom (s.f i)) * h ∈ Set.range (algebraMap (X.presheaf.stalk x) X.functionField)

theorem **SecMaps.ratSecs_eq** {A B : LB X} (hA : A.IsLB) (hB : B.IsLB) {w : LB.Iso A B} {a : A.Sec} {b : B.Sec} (hw : SecMaps w a b) (V : X.Opens) [Nonempty V] : ratSecs a V = ratSecs b V

theorem **mem_genericChart** {L : LB X} (hL : L.IsLB) : genericPoint X ∈ L.U (genericChart hL)

theorem **ratOf_smul** {L : LB X} (hL : L.IsLB) (s : L.Sec) (V : X.Opens) [Nonempty V] (c : Γ(X, V)) (t : ∀ i, Γ(X, V ⊓ L.U i)) : ratOf hL s V (fun i => resG X inf_le_left c * t i) = c • ratOf hL s V t

theorem **germK_eq_mul_ratOf** {L : LB X} (hL : L.IsLB) {s : L.Sec} (hs : IsRegularSec s) (V : X.Opens) [Nonempty V] {t : ∀ i, Γ(X, V ⊓ L.U i)} (ht : t ∈ secOn L V) (i : L.ι) [Nonempty (V ⊓ L.U i : X.Opens)] : (X.germToFunctionField _).hom (t i) = (X.germToFunctionField (V ⊓ L.U i)).hom (resG X inf_le_right (s.f i)) * ratOf hL s V t

theorem **bijOn_ratOf** {L : LB X} (hL : L.IsLB) {s : L.Sec} (hs : IsRegularSec s) (V : X.Opens) [Nonempty V] : Set.BijOn (ratOf hL s V) (secOn L V) (ratSecs s V)

theorem **app_subschemeι_eq_zero_iff** (J : Y.IdealSheafData) {V : Y.Opens} (b : Γ(Y, V)) : (J.subschemeι.app V).hom b = 0 ↔ ∀ (W : Y.affineOpens) (h : W.1 ≤ V), resG Y h b ∈ J.ideal W

theorem **comap_le_iff** (I : X.IdealSheafData) (φ : Y ⟶ X) (J : Y.IdealSheafData) : I.comap φ ≤ J ↔ ∀ (U : X.affineOpens), ∀ a ∈ I.ideal U, ∀ (W : Y.affineOpens) (h : W.1 ≤ φ ⁻¹ᵁ U.1), resG Y h ((φ.app U.1).hom a) ∈ J.ideal W

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.IsWeightedHomogeneousOfMemOrbitSpan

theorem **laplacian_apply** (t : T) (a : Fin p) (b : Fin q) (P : Poly R T p q κ) : laplacian t a b P = ∑ k : κ, pderiv (xVar t a k) (pderiv (yVar t b k) P)

@[simp] theorem **mem_harmonicSubmodule** {P : Poly R T p q κ} : P ∈ harmonicSubmodule R T p q κ ↔ IsHarmonic P

theorem **isHarmonic_C** (r : R) : IsHarmonic (C r : Poly R T p q κ)

theorem **map_laplacian** (f : R →+* S) (t : T) (a : Fin p) (b : Fin q) (P : Poly R T p q κ) : MvPolynomial.map f (laplacian t a b P) = laplacian t a b (MvPolynomial.map f P)

theorem **IsHarmonic.map** (f : R →+* S) {P : Poly R T p q κ} (hP : IsHarmonic P) : IsHarmonic (MvPolynomial.map f P)

theorem **isHarmonic_map_iff** {f : R →+* S} (hf : Function.Injective f) {P : Poly R T p q κ} : IsHarmonic (MvPolynomial.map f P) ↔ IsHarmonic P

theorem **rowForm_apply** (t : T) (r : Fin p ⊕ Fin q) (v : κ → R) : rowForm t r v = ∑ k : κ, C (v k) * X (t, r, k)

theorem **xForm_apply** (t : T) (a : Fin p) (ε : κ → R) : (xForm t a ε : Poly R T p q κ) = ∑ k : κ, C (ε k) * X (xVar t a k)

theorem **yForm_apply** (t : T) (b : Fin q) (φ : κ → R) : (yForm t b φ : Poly R T p q κ) = ∑ k : κ, C (φ k) * X (yVar t b k)

theorem **rowForm_single** [DecidableEq κ] (t : T) (r : Fin p ⊕ Fin q) (k : κ) : (rowForm t r (Pi.single k 1) : Poly R T p q κ) = X (t, r, k)

@[simp] theorem **xForm_single** [DecidableEq κ] (t : T) (a : Fin p) (k : κ) : (xForm t a (Pi.single k 1) : Poly R T p q κ) = X (xVar t a k)

@[simp] theorem **yForm_single** [DecidableEq κ] (t : T) (b : Fin q) (k : κ) : (yForm t b (Pi.single k 1) : Poly R T p q κ) = X (yVar t b k)

theorem **eval_xForm** (z : Var T p q κ → R) (t : T) (a : Fin p) (ε : κ → R) : eval z (xForm t a ε) = ε ⬝ᵥ fun k => z (xVar t a k)

theorem **eval_yForm** (z : Var T p q κ → R) (t : T) (b : Fin q) (φ : κ → R) : eval z (yForm t b φ) = φ ⬝ᵥ fun k => z (yVar t b k)

theorem **laplacian_xForm_mul_yForm** (t : T) (a : Fin p) (b : Fin q) (ε φ : κ → R) : laplacian t a b (xForm t a ε * yForm t b φ : Poly R T p q κ) = C (ε ⬝ᵥ φ)

theorem **map_rowForm** (f : R →+* S) (t : T) (r : Fin p ⊕ Fin q) (v : κ → R) : MvPolynomial.map f (rowForm t r v) = rowForm t r (f ∘ v)

theorem **map_xForm** (f : R →+* S) (t : T) (a : Fin p) (ε : κ → R) : MvPolynomial.map f (xForm t a ε : Poly R T p q κ) = xForm t a (f ∘ ε)

theorem **map_yForm** (f : R →+* S) (t : T) (b : Fin q) (φ : κ → R) : MvPolynomial.map f (yForm t b φ : Poly R T p q κ) = yForm t b (f ∘ φ)

theorem **ringHom_xMinor** [Fintype κ'] (F : Poly R T p q κ →+* Poly S T p q κ') (t : T) (i : Fin p) (ε : Fin p → κ → R) (ε' : Fin p → κ' → S) (h : ∀ a b, F (xForm t a (ε b)) = xForm t a (ε' b)) : F (xMinor t i ε) = xMinor t i ε'

theorem **ringHom_yMinor** [Fintype κ'] (F : Poly R T p q κ →+* Poly S T p q κ') (t : T) (i : Fin q) (φ : Fin q → κ → R) (φ' : Fin q → κ' → S) (h : ∀ a b, F (yForm t a (φ b)) = yForm t a (φ' b)) : F (yMinor t i φ) = yMinor t i φ'

theorem **map_minorProd** [Fintype T] (f : R →+* S) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : MvPolynomial.map f (minorProd c c' ε φ) = minorProd c c' (fun t a => f ∘ ε t a) (fun t b => f ∘ φ t b)

theorem **IsPerp.map** (f : R →+* S) {ε : T → Fin p → κ → R} {φ : T → Fin q → κ → R} (h : IsPerp ε φ) : IsPerp (fun t a => f ∘ ε t a) (fun t b => f ∘ φ t b)

theorem **partOf_antitone** {m : ℕ} (c : Fin m → ℕ) : Antitone (partOf c)

theorem **sum_smul_sum_lead** {M : Type*} [AddCommMonoid M] {m : ℕ} (c : Fin m → ℕ) (w : Fin m → M) : ∑ i : Fin m, c i • ∑ a : Fin (i.val + 1), w (lead i a) = ∑ a : Fin m, partOf c a • w a

theorem **sum_partOf** {m : ℕ} (c : Fin m → ℕ) : ∑ a, partOf c a = partSize c

theorem **partOf_eq_add** {m : ℕ} (c : Fin m → ℕ) (i : Fin m) : partOf c i = c i + if h : i.val + 1 < m then partOf c ⟨i.val + 1, h⟩ else 0

theorem **partOf_diffOf** {m : ℕ} {l : Fin m → ℕ} (hl : Antitone l) : partOf (diffOf l) = l

theorem **rowForm_isWeightedHomogeneous** (w : T → Fin p ⊕ Fin q → M) (t : T) (r : Fin p ⊕ Fin q) (v : κ → R) : IsWeightedHomogeneous (rowWeight w) (rowForm t r v) (w t r)

theorem **det_isWeightedHomogeneous** {σ ι : Type*} [Fintype ι] [DecidableEq ι] (w : σ → M) (A : Matrix ι ι (MvPolynomial σ R)) (u : ι → M) (h : ∀ a b, IsWeightedHomogeneous w (A a b) (u a)) : IsWeightedHomogeneous w A.det (∑ a, u a)

theorem **minorProd_isWeightedHomogeneous** [Fintype T] (w : T → Fin p ⊕ Fin q → M) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : IsWeightedHomogeneous (rowWeight w) (minorProd c c' ε φ) (∑ t, ((∑ a, partOf (c t) a • w t (Sum.inl a)) + ∑ b, partOf (c' t) b • w t (Sum.inr b)))

theorem **minorProd_isHomogeneous** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : (minorProd c c' ε φ).IsHomogeneous (typeDegree c c')

@[simp] theorem **colSubst_X_xVar** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (a : Fin p) (k : κ) : colSubst A B (X (xVar t a k) : Poly R T p q κ) = xForm t a (A t k)

@[simp] theorem **colSubst_X_yVar** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (b : Fin q) (k : κ) : colSubst A B (X (yVar t b k) : Poly R T p q κ) = yForm t b fun l => B t l k

theorem **colSubst_xForm** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (a : Fin p) (ε : κ → R) : colSubst A B (xForm t a ε : Poly R T p q κ) = xForm t a (ε ᵥ* A t)

theorem **colSubst_yForm** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (b : Fin q) (φ : κ → R) : colSubst A B (yForm t b φ : Poly R T p q κ) = yForm t b (B t *ᵥ φ)

theorem **colSubst_comp** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (A' : T → Matrix κ' κ'' R) (B' : T → Matrix κ'' κ' R) : (colSubst A' B').comp (colSubst (p

theorem **colSubst_one** [DecidableEq κ] : colSubst (p

theorem **colSubst_minorProd** [Fintype T] (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : colSubst A B (minorProd c c' ε φ) = minorProd c c' (fun t a => ε t a ᵥ* A t) (fun t b => B t *ᵥ φ t b)

theorem **eval_colSubst** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (z : Var T p q κ' → R) (P : Poly R T p q κ) : eval z (colSubst A B P) = eval (fun v => match v with | (t, Sum.inl a, k) => (A t *ᵥ fun l => z (xVar t a l)) k | (t, Sum.inr b, k) => ((fun l => z (yVar t b l)) ᵥ* B t) k) P

theorem **colSubst_isWeightedHomogeneous** {M : Type*} [AddCommMonoid M] (w : T → Fin p ⊕ Fin q → M) (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) {P : Poly R T p q κ} {m : M} (hP : IsWeightedHomogeneous (rowWeight w) P m) : IsWeightedHomogeneous (rowWeight w) (colSubst A B P) m

theorem **covecAct_dotProduct_vecAct** (g : GL κ R) (ε φ : κ → R) : covecAct g ε ⬝ᵥ vecAct g φ = ε ⬝ᵥ φ

theorem **IsPerp.act** {ε : T → Fin p → κ → R} {φ : T → Fin q → κ → R} (h : IsPerp ε φ) (g : T → GL κ R) : IsPerp (fun t a => covecAct (g t) (ε t a)) (fun t b => vecAct (g t) (φ t b))

theorem **colActHom_one** : colActHom (p

theorem **colActHom_mul** (g h : T → GL κ R) : colActHom (p

theorem **colAct_apply** (g : T → GL κ R) (P : Poly R T p q κ) : colAct R T p q κ g P = colSubst (fun t => ((g t)⁻¹ : GL κ R)) (fun t => g t) P

@[simp] theorem **colRep_apply** (g : T → GL κ R) (P : Poly R T p q κ) : colRep R T p q κ g P = colAct R T p q κ g P

theorem **colAct_xForm** (g : T → GL κ R) (t : T) (a : Fin p) (ε : κ → R) : colAct R T p q κ g (xForm t a ε) = xForm t a (covecAct (g t) ε)

theorem **colAct_yForm** (g : T → GL κ R) (t : T) (b : Fin q) (φ : κ → R) : colAct R T p q κ g (yForm t b φ) = yForm t b (vecAct (g t) φ)

theorem **colAct_minorProd** [Fintype T] (g : T → GL κ R) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : colAct R T p q κ g (minorProd c c' ε φ) = minorProd c c' (fun t a => covecAct (g t) (ε t a)) (fun t b => vecAct (g t) (φ t b))

theorem **isPerp_base** (h : p + q ≤ n) : IsPerp (fun _ : T => baseCovec R h) (fun _ => baseVec R h)

theorem **linearIndependent_baseCovec** (h : p + q ≤ n) : LinearIndependent R (baseCovec R h)

theorem **linearIndependent_baseVec** (h : p + q ≤ n) : LinearIndependent R (baseVec R h)

theorem **colAct_basePoly_mem** (g : T → GL (Fin n) R) : colAct R T p q (Fin n) g (basePoly R h c c') ∈ orbitSpan R h c c'

theorem **basePoly_mem** : basePoly R h c c' ∈ orbitSpan R h c c'

theorem **colAct_mem_orbitSpan** (g : T → GL (Fin n) R) {P : Poly R T p q (Fin n)} (hP : P ∈ orbitSpan R h c c') : colAct R T p q (Fin n) g P ∈ orbitSpan R h c c'

theorem **map_basePoly** (f : R →+* S) : MvPolynomial.map f (basePoly R h c c') = basePoly S h c c'

theorem **eval_basePoly** : eval (fun v : Var T p q (Fin n) => Sum.elim (baseCovec R h) (baseVec R h) v.2.1 v.2.2) (basePoly R h c c') = 1

theorem **basePoly_ne_zero** [Nontrivial R] : basePoly R h c c' ≠ 0

theorem **totalDegree_basePoly** [Nontrivial R] : (basePoly R h c c').totalDegree = typeDegree c c'

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.LocusEqTopOfDualPow

theorem **locus_eq_top_of_dual_pow** [IrreducibleSpace X] (hF : IsField Γ(X, ⊤)) {L : LB X} (hL : L.IsLB) {n : ℕ} (hn : 1 ≤ n) (t : (lbPow (CechPic.dual L) n).Sec) (ht : nonzeroLocus t ≠ ⊥) (s : L.Sec) (hs : nonzeroLocus s ≠ ⊥) : nonzeroLocus s = ⊤

theorem **isField_global** : IsField Γ(PSpace K N, ⊤)

theorem **coord_ne_bot** : nonzeroLocus (coord K 1 1) ≠ ⊥

theorem **coord_ne_top** : nonzeroLocus (coord K 1 1) ≠ ⊤

theorem **locus_dual_O1_pow** {n : ℕ} (hn : 1 ≤ n) (t : (lbPow (CechPic.dual (O1 K 1)) n).Sec) : nonzeroLocus t = ⊥

theorem **not_isAmpleLB_dual_O1** : (CechPic.dual (O1 K 1)).IsLB ∧ ¬ IsAmpleLB (CechPic.dual (O1 K 1))

theorem **not_isGenerated_dual_O1** : ¬ IsGenerated (CechPic.dual (O1 K 1))

theorem **not_isAmpleLB_triv** : IsGenerated (LB.triv (PSpace K 1)) ∧ ¬ IsAmpleLB (LB.triv (PSpace K 1))

theorem **ample_not_quasiAffine** : IsAmpleLB (O1 K 1) ∧ ¬ (PSpace K 1).IsQuasiAffine

theorem **not_isAmpleLB_pow_zero** : ¬ IsAmpleLB (lbPow (O1 K 1) 0)

theorem **empty_ample** [IsEmpty X] (L : LB X) : IsAmpleLB L ∧ IsGenerated L

theorem **signData_not_ample** : ¬ IsAmpleLB (signData (Sp ℚ)) ∧ ¬ IsGenerated (signData (Sp ℚ)) ∧ ∃ s : (lbPow (signData (Sp ℚ)) 2).Sec, nonzeroLocus s = ⊤ ∧ IsAffineOpen (nonzeroLocus s)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.MemIffConjugateNotMem

theorem **mem_iff_conjugate_notMem** (Φ : CMType E) (τ : E →+* ℂ) : τ ∈ Φ ↔ ComplexEmbedding.conjugate τ ∉ Φ

theorem **conjugate_mem_iff** (Φ : CMType E) (τ : E →+* ℂ) : ComplexEmbedding.conjugate τ ∈ Φ ↔ τ ∉ Φ

theorem **mem_or_conjugate_mem** (Φ : CMType E) (τ : E →+* ℂ) : τ ∈ Φ ∨ ComplexEmbedding.conjugate τ ∈ Φ

theorem **mem_conjugate** (Φ : CMType E) (τ : E →+* ℂ) : τ ∈ Φ.conjugate ↔ ComplexEmbedding.conjugate τ ∈ Φ

theorem **sum_univ_eq** {M : Type*} [AddCommMonoid M] (g : (E →+* ℂ) → M) : ∑ τ, g τ = ∑ τ : Φ, (g τ + g (ComplexEmbedding.conjugate τ.1))

theorem **two_mul_card** : 2 * Fintype.card Φ = Module.finrank ℚ E

theorem **card_eq_finrank** [IsCMField E] : Fintype.card Φ = Module.finrank ℚ (maximalRealSubfield E)

theorem **trace_eq_sum_two_mul_re** (x : E) : ((Algebra.trace ℚ E x : ℚ) : ℝ) = ∑ τ : Φ, 2 * (τ.1 x).re

theorem **eq_zero_of_sum_re_eq_zero** (δ : Φ → ℂ) (h : ∀ a : 𝒪 E, ∑ τ : Φ, (τ.1 (a : E) * δ τ).re = 0) : δ = 0

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.MemNonzeroLocusIffPowFn

theorem **chartSec_f** (a : ChartPull A' A h) (s : A.Sec) (i : A'.ι) : (chartSec a s).f i = (h.appLE _ _ (a.le i)).hom (s.f (a.idx i))

theorem **gs_chartSec** (a : ChartPull A' A h) (s : A.Sec) (r : Z ⟶ Y') (z : Z) (i : A'.ι) (hz : r z ∈ A'.U i) : gs r z _ ((chartSec a s).f i) = gs (r ≫ h) z _ (s.f (a.idx i))

theorem **nonzeroLocus_chartSec** (a : ChartPull A' A h) (s : A.Sec) (hc : ∀ (y : Y') j, h y ∈ A.U j → ∃ i, y ∈ A'.U i) : nonzeroLocus (chartSec a s) = h ⁻¹ᵁ nonzeroLocus s

theorem **nonzeroLocus_chartSec_id** {B B' : LB Y} (a : ChartPull B' B (𝟙 Y)) (s : B.Sec) (hc : ∀ (y : Y) j, y ∈ B.U j → ∃ i, y ∈ B'.U i) : nonzeroLocus (chartSec a s) = nonzeroLocus s

private theorem **lbIso_g_cocycle_on_inter_of_pairs** (hA : A.IsLB) (p₁ p₂ q₁ q₂ : A.ι) : res (inf_le_left : A.U p₁ ⊓ A.U p₂ ⊓ (A.U q₁ ⊓ A.U q₂) ≤ A.U p₁ ⊓ A.U p₂) (A.g p₁ p₂) = res (inf_le_inf inf_le_left inf_le_left : A.U p₁ ⊓ A.U p₂ ⊓ (A.U q₁ ⊓ A.U q₂) ≤ A.U p₁ ⊓ A.U q₁) (A.g p₁ q₁) * (res (inf_le_inf inf_le_right inf_le_right : A.U p₁ ⊓ A.U p₂ ⊓ (A.U q₁ ⊓ A.U q₂) ≤ A.U p₂ ⊓ A.U q₂) (A.g p₂ q₂))⁻¹ * res (inf_le_right : A.U p₁ ⊓ A.U p₂ ⊓ (A.U q₁ ⊓ A.U q₂) ≤ A.U q₁ ⊓ A.U q₂) (A.g q₁ q₂)

theorem **lbIso_refl** (hA : A.IsLB) : LBIso A A

private theorem **lbIso_of_chartPull_lemma** {B B' : LB Y} (a : ChartPull B' B (𝟙 Y)) {y : Y} {p₁ q₁ : B.ι} {p₂ q₂ : B'.ι} (hp₁ : y ∈ B.U p₁) (hp₂ : y ∈ B'.U p₂) (hq₁ : y ∈ B.U q₁) (hq₂ : y ∈ B'.U q₂) : gm (𝟙 Y) y (B.U p₁ ⊓ B'.U p₂ ⊓ (B.U q₁ ⊓ B'.U q₂)) (res (inf_le_inf inf_le_left inf_le_left : B.U p₁ ⊓ B'.U p₂ ⊓ (B.U q₁ ⊓ B'.U q₂) ≤ B.U p₁ ⊓ B.U q₁) (B.g p₁ q₁) * (res (inf_le_inf inf_le_right inf_le_right : B.U p₁ ⊓ B'.U p₂ ⊓ (B.U q₁ ⊓ B'.U q₂) ≤ B'.U p₂ ⊓ B'.U q₂) (B'.g p₂ q₂))⁻¹) = gm (𝟙 Y) y (B.U p₁ ⊓ B.U (a.idx p₂) ⊓ (B.U q₁ ⊓ B.U (a.idx q₂))) (res (inf_le_inf inf_le_left inf_le_left : B.U p₁ ⊓ B.U (a.idx p₂) ⊓ (B.U q₁ ⊓ B.U (a.idx q₂)) ≤ B.U p₁ ⊓ B.U q₁) (B.g p₁ q₁) * (res (inf_le_inf inf_le_right inf_le_right : B.U p₁ ⊓ B.U (a.idx p₂) ⊓ (B.U q₁ ⊓ B.U (a.idx q₂)) ≤ B.U (a.idx p₂) ⊓ B.U (a.idx q₂)) (B.g (a.idx p₂) (a.idx q₂)))⁻¹)

theorem **lbIso_of_chartPull** {B B' : LB Y} (a : ChartPull B' B (𝟙 Y)) (hB : B.IsLB) : LBIso B' B

theorem **nonzeroLocus_secPull** (s : L.Sec) (f : Y ⟶ X) : nonzeroLocus (secPull s f) = f ⁻¹ᵁ nonzeroLocus s

theorem **nonzeroLocus_pullSec** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (s : (LB.LinComb L φ ε).Sec) (h : Z ⟶ Y) : nonzeroLocus (DVR.pullSec L φ ε s h) = h ⁻¹ᵁ nonzeroLocus s

theorem **nonzeroLocus_mulSec** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (s : (LB.LinComb L φ ε).Sec) (s' : (LB.LinComb L φ' ε').Sec) : nonzeroLocus (DVR.mulSec L φ ε φ' ε' s s') = nonzeroLocus s ⊓ nonzeroLocus s'

theorem **mem_powChart** {J : PUnit → L.ι} {x : X} : x ∈ powChart L J ↔ x ∈ L.U (J PUnit.unit)

theorem **powChart_eq** (J : PUnit → L.ι) : powChart L J = L.U (J PUnit.unit)

theorem **isLB_lbPow** (hL : L.IsLB) (n : ℕ) : (lbPow L n).IsLB

theorem **gm_lbPow_g** (n : ℕ) (r : Z ⟶ X) (z : Z) (J J' : PUnit → L.ι) (hJ : r z ∈ powChart L J) (hJ' : r z ∈ powChart L J') : gm r z ((lbPow L n).U J ⊓ (lbPow L n).U J') ((lbPow L n).g J J') = gm r z (L.U (J PUnit.unit) ⊓ L.U (J' PUnit.unit)) (L.g (J PUnit.unit) (J' PUnit.unit)) ^ n

theorem **gs_powFn_compat** (s : (lbPow L n).Sec) (r : Z ⟶ X) (z : Z) (J J' : PUnit → L.ι) (hJ : r z ∈ powChart L J) (hJ' : r z ∈ powChart L J') : gs r z _ (powFn s J) = (gm r z (L.U (J PUnit.unit) ⊓ L.U (J' PUnit.unit)) (L.g (J PUnit.unit) (J' PUnit.unit))).val ^ n * gs r z _ (powFn s J')

theorem **mem_nonzeroLocus_iff_powFn** (s : (lbPow L n).Sec) {y : X} {J : PUnit → L.ι} (hJ : y ∈ powChart L J) : y ∈ nonzeroLocus s ↔ IsUnit (gs (𝟙 X) y _ (powFn s J))

theorem **nonzeroLocus_pow_eq_of** (s : (lbPow L n).Sec) (W : X.Opens) (hW : ∀ y ∈ W, ∃ J, y ∈ powChart L J) (h : ∀ (y : X) J, y ∈ powChart L J → (IsUnit (gs (𝟙 X) y _ (powFn s J)) ↔ y ∈ W)) : nonzeroLocus s = W

@[simp] theorem **powFn_secMul** (s : (lbPow L m).Sec) (t : (lbPow L n).Sec) (J : PUnit → L.ι) : powFn (secMul s t) J = powFn s J * powFn t J

theorem **nonzeroLocus_secMul** (s : (lbPow L m).Sec) (t : (lbPow L n).Sec) : nonzeroLocus (secMul s t) = nonzeroLocus s ⊓ nonzeroLocus t

@[simp] theorem **powFn_secPow** (s : (lbPow L m).Sec) (k : ℕ) (J : PUnit → L.ι) : powFn (secPow s k) J = powFn s J ^ k

theorem **nonzeroLocus_secPow** (s : (lbPow L m).Sec) {k : ℕ} (hk : 1 ≤ k) : nonzeroLocus (secPow s k) = nonzeroLocus s

@[simp] theorem **powFn_secCast** (h : m = n) (s : (lbPow L m).Sec) (J : PUnit → L.ι) : powFn (secCast h s) J = powFn s J

theorem **nonzeroLocus_secCast** (h : m = n) (s : (lbPow L m).Sec) : nonzeroLocus (secCast h s) = nonzeroLocus s

@[simp] theorem **powFn_secOfFn** (c : Γ(X, ⊤)) (J : PUnit → L.ι) : powFn (secOfFn L c) J = resG X le_top c

theorem **nonzeroLocus_secOfFn** (hL : L.IsLB) (c : Γ(X, ⊤)) : nonzeroLocus (secOfFn L c) = X.basicOpen c

@[simp] theorem **powFn_powSecMk** (n : ℕ) (f : ∀ i, Γ(X, L.U i)) (h) (J : PUnit → L.ι) : powFn (powSecMk n f h) J = resG X (powChart_eq J).le (f (J PUnit.unit))

theorem **nonzeroLocus_powSecMk** (n : ℕ) (f : ∀ i, Γ(X, L.U i)) (h) : nonzeroLocus (powSecMk n f h) = ⨆ i, X.basicOpen (f i)

theorem **powSecFn_compat** (s : (lbPow L n).Sec) (i j : L.ι) : resG X (inf_le_left : L.U i ⊓ L.U j ≤ L.U i) (powSecFn s i) = (L.g i j).val ^ n * resG X (inf_le_right : L.U i ⊓ L.U j ≤ L.U j) (powSecFn s j)

theorem **nonzeroLocus_inf_powSecFn** (s : (lbPow L n).Sec) (i : L.ι) : nonzeroLocus s ⊓ L.U i = X.basicOpen (powSecFn s i)

theorem **nonzeroLocus_secToPow** (s : L.Sec) (hn : 1 ≤ n) : nonzeroLocus (secToPow s n) = nonzeroLocus s

theorem **nonzeroLocus_secOfPowOne** (s : (lbPow L 1).Sec) : nonzeroLocus (secOfPowOne s) = nonzeroLocus s

theorem **nonzeroLocus_secPowPull** (s : (lbPow L n).Sec) (f : Y ⟶ X) : nonzeroLocus (secPowPull s f) = f ⁻¹ᵁ nonzeroLocus s

theorem **nonzeroLocus_secToPowPow** (s : (lbPow L (m * n)).Sec) : nonzeroLocus (secToPowPow s) = nonzeroLocus s

theorem **nonzeroLocus_secOfPowPow** (s : (lbPow (lbPow L m) n).Sec) : nonzeroLocus (secOfPowPow s) = nonzeroLocus s

theorem **nonzeroLocus_secOfFinPow** {f : Y ⟶ X} (hn : 1 ≤ n) (s : (LB.LinComb L (K

theorem **lbIso_lbPow_one** (hL : L.IsLB) : LBIso (lbPow L 1) L

theorem **lbIso_lbPow_lbPow** (hL : L.IsLB) (m n : ℕ) : LBIso (lbPow (lbPow L m) n) (lbPow L (m * n))

theorem **lbIso_lbPow_pull** (hL : L.IsLB) (f : Y ⟶ X) (n : ℕ) : LBIso (lbPow (L.pull f) n) ((lbPow L n).pull f)

theorem **lbIso_lbPow_finPow** (hL : L.IsLB) (f : Y ⟶ X) (n : ℕ) : LBIso (lbPow (L.pull f) n) (LB.LinComb L (K

private theorem **lbIso_lbPow_lemma** {M : LB X} (n : ℕ) (P₁ Q₁ : PUnit → M.ι) (P₂ Q₂ : PUnit → L.ι) (uP : Γ(X, M.U (P₁ PUnit.unit) ⊓ L.U (P₂ PUnit.unit))ˣ) (uQ : Γ(X, M.U (Q₁ PUnit.unit) ⊓ L.U (Q₂ PUnit.unit))ˣ) (hu : res (inf_le_left : M.U (P₁ PUnit.unit) ⊓ L.U (P₂ PUnit.unit) ⊓ (M.U (Q₁ PUnit.unit) ⊓ L.U (Q₂ PUnit.unit)) ≤ M.U (P₁ PUnit.unit) ⊓ L.U (P₂ PUnit.unit)) uP = res (inf_le_inf inf_le_left inf_le_left : M.U (P₁ PUnit.unit) ⊓ L.U (P₂ PUnit.unit) ⊓ (M.U (Q₁ PUnit.unit) ⊓ L.U (Q₂ PUnit.unit)) ≤ M.U (P₁ PUnit.unit) ⊓ M.U (Q₁ PUnit.unit)) (M.g (P₁ PUnit.unit) (Q₁ PUnit.unit)) * (res (inf_le_inf inf_le_right inf_le_right : M.U (P₁ PUnit.unit) ⊓ L.U (P₂ PUnit.unit) ⊓ (M.U (Q₁ PUnit.unit) ⊓ L.U (Q₂ PUnit.unit)) ≤ L.U (P₂ PUnit.unit) ⊓ L.U (Q₂ PUnit.unit)) (L.g (P₂ PUnit.unit) (Q₂ PUnit.unit)))⁻¹ * res (inf_le_right : M.U (P₁ PUnit.unit) ⊓ L.U (P₂ PUnit.unit) ⊓ (M.U (Q₁ PUnit.unit) ⊓ L.U (

theorem **lbIso_lbPow** {M : LB X} (H : LBIso L M) (n : ℕ) : LBIso (lbPow L n) (lbPow M n)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.MemRangeCompIffOfIsClosedImmersion

theorem **valR_eq** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (f : Γ(X, U)) : valR x U f = (valMap x U h).hom f

theorem **valF_eq_valR** {F : Type} [Field F] (x : Sp F ⟶ X) (U : X.Opens) (f : Γ(X, U)) : valF x U f = valR x U f

theorem **valAt'_eq_valR** (x : Sp ℂ ⟶ X) (U : X.Opens) (f : Γ(X, U)) : valAt' x U f = valR x U f

theorem **valAt_eq_valR** (x : Sp ℂ ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)) : valAt x U h f = valR x U f

theorem **le_preimage_mono** (x : Sp C ⟶ X) {U V : X.Opens} (e : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵘ V) : ⊤ ≤ x ⁻¹ᵘ U

theorem **le_preimage_comp** (x : Sp C ⟶ X) (g : X ⟶ Y) {U : Y.Opens} {V : X.Opens} (e : V ≤ g ⁻¹ᵘ U) (h : ⊤ ≤ x ⁻¹ᵘ V) : ⊤ ≤ (x ≫ g) ⁻¹ᵘ U

theorem **appLE_valMap** (x : Sp C ⟶ X) (g : X ⟶ Y) {U : Y.Opens} {V : X.Opens} (e : V ≤ g ⁻¹ᵘ U) (h : ⊤ ≤ x ⁻¹ᵘ V) (h' : ⊤ ≤ (x ≫ g) ⁻¹ᵘ U) : g.appLE U V e ≫ valMap x V h = valMap (x ≫ g) U h'

theorem **res_valMap** (x : Sp C ⟶ X) {U V : X.Opens} (e : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵘ V) (h' : ⊤ ≤ x ⁻¹ᵘ U) : X.presheaf.map (homOfLE e).op ≫ valMap x V h = valMap x U h'

theorem **valR_appLE** (x : Sp C ⟶ X) (g : X ⟶ Y) {U : Y.Opens} {V : X.Opens} (e : V ≤ g ⁻¹ᵘ U) (h : ⊤ ≤ x ⁻¹ᵘ V) (f : Γ(Y, U)) : valR x V ((g.appLE U V e).hom f) = valR (x ≫ g) U f

theorem **valR_comp** (x : Sp C ⟶ X) (g : X ⟶ Y) (U : Y.Opens) (f : Γ(Y, U)) : valR (x ≫ g) U f = valR x (g ⁻¹ᵘ U) ((g.app U).hom f)

theorem **valR_res** (x : Sp C ⟶ X) {U V : X.Opens} (e : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵘ V) (f : Γ(X, U)) : valR x V ((X.presheaf.map (homOfLE e).op).hom f) = valR x U f

theorem **valR_add** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (f g : Γ(X, U)) : valR x U (f + g) = valR x U f + valR x U g

theorem **valR_mul** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (f g : Γ(X, U)) : valR x U (f * g) = valR x U f * valR x U g

theorem **valR_neg** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)) : valR x U (-f) = -valR x U f

theorem **valR_pow** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)) (n : ℕ) : valR x U (f ^ n) = valR x U f ^ n

theorem **valR_one** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) : valR x U 1 = 1

theorem **valR_zero** (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) : valR x U 0 = 0

theorem **le_preimage_top** (x : Sp C ⟶ X) : ⊤ ≤ x ⁻¹ᵘ ⊤

theorem **valMap_specMap** {A : CommRingCat.{0}} (φ : A ⟶ CommRingCat.of C) (h : ⊤ ≤ Spec.map φ ⁻¹ᵘ ⊤) : valMap (Spec.map φ) ⊤ h = (Scheme.ΓSpecIso A).hom ≫ φ

theorem **valR_specMap** {A : CommRingCat.{0}} (φ : A ⟶ CommRingCat.of C) (a : A) : valR (Spec.map φ) ⊤ ((Scheme.ΓSpecIso A).inv.hom a) = φ.hom a

theorem **preimage_apply** {A : CommRingCat.{0}} (x : Sp C ⟶ Spec A) (a : A) : (Spec.preimage x).hom a = valR x ⊤ ((Scheme.ΓSpecIso A).inv.hom a)

theorem **specMap_valMap_fromSpec** {U : X.Opens} (hU : IsAffineOpen U) (x : Sp C ⟶ X) (h : ⊤ ≤ x ⁻¹ᵘ U) : Spec.map (valMap x U h) ≫ hU.fromSpec = x

theorem **le_preimage_specMap_fromSpec** {U : X.Opens} (hU : IsAffineOpen U) (φ : Γ(X, U) ⟶ CommRingCat.of C) : ⊤ ≤ (Spec.map φ ≫ hU.fromSpec) ⁻¹ᵘ U

theorem **valMap_specMap_fromSpec** {U : X.Opens} (hU : IsAffineOpen U) (φ : Γ(X, U) ⟶ CommRingCat.of C) (h : ⊤ ≤ (Spec.map φ ≫ hU.fromSpec) ⁻¹ᵘ U) : valMap (Spec.map φ ≫ hU.fromSpec) U h = φ

theorem **ext_of_valR** {U : X.Opens} (hU : IsAffineOpen U) {x y : Sp C ⟶ X} (hx : ⊤ ≤ x ⁻¹ᵘ U) (hy : ⊤ ≤ y ⁻¹ᵘ U) (H : ∀ f, valR x U f = valR y U f) : x = y

theorem **le_preimage_basicOpen_iff** (x : Sp C ⟶ X) {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)) : ⊤ ≤ x ⁻¹ᵘ X.basicOpen f ↔ IsUnit (valR x U f)

theorem **valR_spMap_comp** {C' : Type} [CommRing C'] (σ : C →+* C') (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)) : valR (spMap σ ≫ x) U f = σ (valR x U f)

theorem **valR_constFn** {R : Type} [CommRing R] (p : X ⟶ Sp R) (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (a : R) : valR x U (constFn p U a) = (Spec.preimage (x ≫ p)).hom a

theorem **valR_constFn_of_over** {R : Type} [CommRing R] (p : X ⟶ Sp R) (σ : R →+* C) (x : Sp C ⟶ X) (hx : x ≫ p = spMap σ) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (a : R) : valR x U (constFn p U a) = σ a

theorem **valR_constFn_of_section** (p : X ⟶ Sp C) (x : Sp C ⟶ X) (hx : x ≫ p = 𝟙 _) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵘ U) (a : C) : valR x U (constFn p U a) = a

theorem **le_preimage_iff** (x : Sp C ⟶ X) (U : X.Opens) : ⊤ ≤ x ⁻¹ᵘ U ↔ x (IsLocalRing.closedPoint C) ∈ U

theorem **exists_isAffineOpen_le** (x : Sp C ⟶ X) {V : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ V) : ∃ U : X.Opens, IsAffineOpen U ∧ ⊤ ≤ x ⁻¹ᵘ U ∧ U ≤ V

theorem **exists_le_preimage_of_iSup_eq_top** (x : Sp C ⟶ X) {ι : Sort*} {U : ι → X.Opens} (hU : iSup U = ⊤) : ∃ i, ⊤ ≤ x ⁻¹ᵘ U i

@[simp] theorem **PtsOver.map_val** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (g : X ⟶ Y) (hg : g ≫ q = p) (s : Sp C ⟶ S) (x : PtsOver p s) : (PtsOver.map g hg s x).1 = x.1 ≫ g

@[simp] theorem **PtsOver.map_id** {S : Scheme.{0}} (p : X ⟶ S) (s : Sp C ⟶ S) (x : PtsOver p s) : PtsOver.map (𝟙 X) (Category.id_comp p) s x = x

theorem **PtsOver.map_map** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} {r : Z ⟶ S} (g : X ⟶ Y) (hg : g ≫ q = p) (k : Y ⟶ Z) (hk : k ≫ r = q) (s : Sp C ⟶ S) (x : PtsOver p s) : PtsOver.map k hk s (PtsOver.map g hg s x) = PtsOver.map (g ≫ k) (by rw [Category.assoc, hk, hg]) s x

theorem **topology_complex** (X : Scheme.{0}) : topology ℂ X = generateFrom {S | ∃ (U : X.Opens) (f : Γ(X, U)) (O : Set ℂ), IsOpen O ∧ S = {x : Sp ℂ ⟶ X | ∃ h : ⊤ ≤ x ⁻¹ᵘ U, valAt x U h f ∈ O}}

theorem **isOpen_basicSet** (U : X.Opens) (f : Γ(X, U)) {O : Set C} (hO : IsOpen O) : IsOpen (basicSet U f O)

theorem **isOpen_setOf_le_preimage** (U : X.Opens) : IsOpen {x : Sp C ⟶ X | ⊤ ≤ x ⁻¹ᵘ U}

theorem **tendsto_iff** {B : Type*} {e : B → (Sp C ⟶ X)} {l : Filter B} {x : Sp C ⟶ X} : Tendsto e l (𝓝 x) ↔ ∀ U : X.Opens, ⊤ ≤ x ⁻¹ᵘ U → (∀ᶠ b in l, ⊤ ≤ e b ⁻¹ᵘ U) ∧ ∀ f : Γ(X, U), Tendsto (fun b => valR (e b) U f) l (𝓝 (valR x U f))

theorem **continuousAt_valR** {U : X.Opens} (f : Γ(X, U)) {x : Sp C ⟶ X} (hx : ⊤ ≤ x ⁻¹ᵘ U) : ContinuousAt (fun y : Sp C ⟶ X => valR y U f) x

theorem **continuousOn_valR** {U : X.Opens} (f : Γ(X, U)) : ContinuousOn (fun y : Sp C ⟶ X => valR y U f) {y | ⊤ ≤ y ⁻¹ᵘ U}

theorem **continuous_valR_top** (f : Γ(X, ⊤)) : Continuous fun y : Sp C ⟶ X => valR y ⊤ f

theorem **continuous_iff** {B : Type*} [TopologicalSpace B] {e : B → (Sp C ⟶ X)} : Continuous e ↔ ∀ U : X.Opens, IsOpen {b | ⊤ ≤ e b ⁻¹ᵘ U} ∧ ∀ f : Γ(X, U), ContinuousOn (fun b => valR (e b) U f) {b | ⊤ ≤ e b ⁻¹ᵘ U}

@[fun_prop] theorem **continuous_comp** (g : X ⟶ Y) : Continuous fun x : Sp C ⟶ X => x ≫ g

theorem **PtsOver.continuous_map** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (g : X ⟶ Y) (hg : g ≫ q = p) (s : Sp C ⟶ S) : Continuous (PtsOver.map g hg s)

@[simp] theorem **isoHomeomorph_apply** (g : X ≅ Y) (x : Sp C ⟶ X) : isoHomeomorph g x = x ≫ g.hom

theorem **isInducing_of_tendsto** {α β : Type*} [TopologicalSpace α] [TopologicalSpace β] {f : α → β} (hf : Continuous f) (H : ∀ (x : α) (l : Filter α), Tendsto f l (𝓝 (f x)) → Tendsto id l (𝓝 x)) : IsInducing f

theorem **PtsOver.isEmbedding_map** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (g : X ⟶ Y) (hg : g ≫ q = p) (s : Sp C ⟶ S) (H : IsEmbedding fun x : Sp C ⟶ X => x ≫ g) : IsEmbedding (PtsOver.map g hg s)

theorem **PtsOver.range_map** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (g : X ⟶ Y) (hg : g ≫ q = p) (s : Sp C ⟶ S) : Set.range (PtsOver.map g hg s) = Subtype.val ⁻¹' Set.range fun x : Sp C ⟶ X => x ≫ g

theorem **range_comp_of_isOpenImmersion** (j : X ⟶ Y) [IsOpenImmersion j] : Set.range (fun x : Sp C ⟶ X => x ≫ j) = {y | ⊤ ≤ y ⁻¹ᵘ j.opensRange}

theorem **isOpenEmbedding_comp** (j : X ⟶ Y) [IsOpenImmersion j] : IsOpenEmbedding fun x : Sp C ⟶ X => x ≫ j

theorem **PtsOver.isOpenEmbedding_map** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (j : X ⟶ Y) [IsOpenImmersion j] (hj : j ≫ q = p) (s : Sp C ⟶ S) : IsOpenEmbedding (PtsOver.map j hj s)

theorem **continuousAt_inverse_of_isUnit** [IsOpenUnits C] {u : C} (hu : IsUnit u) : ContinuousAt Ring.inverse u

theorem **tendsto_valR_of_closure** {B : Type*} {e : B → (Sp C ⟶ X)} {l : Filter B} {x : Sp C ⟶ X} {U : X.Opens} (hx : ⊤ ≤ x ⁻¹ᵘ U) (hl : ∀ᶠ b in l, ⊤ ≤ e b ⁻¹ᵘ U) {s : Set Γ(X, U)} (H : ∀ f ∈ s, Tendsto (fun b => valR (e b) U f) l (𝓝 (valR x U f))) {f : Γ(X, U)} (hf : f ∈ Subring.closure s) : Tendsto (fun b => valR (e b) U f) l (𝓝 (valR x U f))

theorem **tendsto_of_isAffineOpen** {B : Type*} {e : B → (Sp C ⟶ X)} {l : Filter B} {x : Sp C ⟶ X} {U : X.Opens} (hU : IsAffineOpen U) (hx : ⊤ ≤ x ⁻¹ᵘ U) (hl : ∀ᶠ b in l, ⊤ ≤ e b ⁻¹ᵘ U) (H : ∀ f : Γ(X, U), Tendsto (fun b => valR (e b) U f) l (𝓝 (valR x U f))) : Tendsto e l (𝓝 x)

theorem **tendsto_iff_of_generators** {B : Type*} {e : B → (Sp C ⟶ X)} {l : Filter B} {x : Sp C ⟶ X} {U : X.Opens} (hU : IsAffineOpen U) (hx : ⊤ ≤ x ⁻¹ᵘ U) {s : Set Γ(X, U)} (hs : Subring.closure s = ⊤) : Tendsto e l (𝓝 x) ↔ (∀ᶠ b in l, ⊤ ≤ e b ⁻¹ᵘ U) ∧ ∀ f ∈ s, Tendsto (fun b => valR (e b) U f) l (𝓝 (valR x U f))

theorem **PtsOver.tendsto_iff_of_generators** {R : Type} [CommRing R] {p : X ⟶ Sp R} {t : Sp C ⟶ Sp R} {B : Type*} {e : B → PtsOver p t} {l : Filter B} {x : PtsOver p t} {U : X.Opens} (hU : IsAffineOpen U) (hx : ⊤ ≤ x.1 ⁻¹ᵘ U) {s : Set Γ(X, U)} (hs : Subring.closure (Set.range (constFn p U) ∪ s) = ⊤) : Tendsto e l (𝓝 x) ↔ (∀ᶠ b in l, ⊤ ≤ (e b).1 ⁻¹ᵘ U) ∧ ∀ f ∈ s, Tendsto (fun b => valR (e b).1 U f) l (𝓝 (valR x.1 U f))

theorem **continuous_iff_of_generators** {B : Type*} [TopologicalSpace B] {e : B → (Sp C ⟶ X)} {ι : Sort*} {U : ι → X.Opens} (hU : ∀ i, IsAffineOpen (U i)) (hcov : iSup U = ⊤) {s : ∀ i, Set Γ(X, U i)} (hs : ∀ i, Subring.closure (s i) = ⊤) : Continuous e ↔ ∀ i, IsOpen {b | ⊤ ≤ e b ⁻¹ᵘ U i} ∧ ∀ f ∈ s i, ContinuousOn (fun b => valR (e b) (U i) f) {b | ⊤ ≤ e b ⁻¹ᵘ U i}

theorem **PtsOver.continuous_iff_of_generators** {R : Type} [CommRing R] {p : X ⟶ Sp R} {t : Sp C ⟶ Sp R} {B : Type*} [TopologicalSpace B] {e : B → PtsOver p t} {ι : Sort*} {U : ι → X.Opens} (hU : ∀ i, IsAffineOpen (U i)) (hcov : iSup U = ⊤) {s : ∀ i, Set Γ(X, U i)} (hs : ∀ i, Subring.closure (Set.range (constFn p (U i)) ∪ s i) = ⊤) : Continuous e ↔ ∀ i, IsOpen {b | ⊤ ≤ (e b).1 ⁻¹ᵘ U i} ∧ ∀ f ∈ s i, ContinuousOn (fun b => valR (e b).1 (U i) f) {b | ⊤ ≤ (e b).1 ⁻¹ᵘ U i}

theorem **continuous_specMap** (A : CommRingCat.{0}) : Continuous fun φ : A ⟶ CommRingCat.of C => (Spec.map φ : Sp C ⟶ Spec A)

@[simp] theorem **specHomeomorph_apply** (A : CommRingCat.{0}) (x : Sp C ⟶ Spec A) : specHomeomorph C A x = Spec.preimage x

@[simp] theorem **specHomeomorph_symm_apply** (A : CommRingCat.{0}) (φ : A ⟶ CommRingCat.of C) : (specHomeomorph C A).symm φ = Spec.map φ

@[simp] theorem **affineHomeomorph_apply** {U : X.Opens} (hU : IsAffineOpen U) (x : {x : Sp C ⟶ X // ⊤ ≤ x ⁻¹ᵘ U}) : affineHomeomorph hU x = valMap x.1 U x.2

@[simp] theorem **affineHomeomorph_symm_val** {U : X.Opens} (hU : IsAffineOpen U) (φ : Γ(X, U) ⟶ CommRingCat.of C) : ((affineHomeomorph hU).symm φ).1 = Spec.map φ ≫ hU.fromSpec

theorem **continuous_spMap_comp** {C₀ : Type} [CommRing C₀] [TopologicalSpace C₀] [IsLocalRing C₀] (σ : C₀ →+* C) (hσ : Continuous σ) : Continuous fun x : Sp C₀ ⟶ X => spMap σ ≫ x

theorem **isEmbedding_comp_of_isClosedImmersion** (i : X ⟶ Y) [IsClosedImmersion i] : IsEmbedding fun x : Sp C ⟶ X => x ≫ i

theorem **mem_range_comp_iff_of_isClosedImmersion** (i : X ⟶ Y) [IsClosedImmersion i] {V : Y.Opens} (hV : IsAffineOpen V) (y : Sp C ⟶ Y) (hy : ⊤ ≤ y ⁻¹ᵘ V) : y ∈ Set.range (fun x : Sp C ⟶ X => x ≫ i) ↔ ∀ f, (i.app V).hom f = 0 → valR y V f = 0

theorem **isClosedEmbedding_comp** [T1Space C] (i : X ⟶ Y) [IsClosedImmersion i] : IsClosedEmbedding fun x : Sp C ⟶ X => x ≫ i

theorem **isEmbedding_comp** (i : X ⟶ Y) [IsImmersion i] : IsEmbedding fun x : Sp C ⟶ X => x ≫ i

theorem **PtsOver.isClosedEmbedding_map** [T1Space C] {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (i : X ⟶ Y) [IsClosedImmersion i] (hi : i ≫ q = p) (s : Sp C ⟶ S) : IsClosedEmbedding (PtsOver.map i hi s)

theorem **PtsOver.isEmbedding_map_of_isImmersion** {S : Scheme.{0}} {p : X ⟶ S} {q : Y ⟶ S} (i : X ⟶ Y) [IsImmersion i] (hi : i ≫ q = p) (s : Sp C ⟶ S) : IsEmbedding (PtsOver.map i hi s)

theorem **isEmbedding_of_isPullback** {P S : Scheme.{0}} {fst : P ⟶ X} {snd : P ⟶ Y} {f : X ⟶ S} {g : Y ⟶ S} (hP : IsPullback fst snd f g) : IsEmbedding fun z : Sp C ⟶ P => (z ≫ fst, z ≫ snd)

@[simp] theorem **pullbackHomeomorph_val** {P S : Scheme.{0}} {fst : P ⟶ X} {snd : P ⟶ Y} {f : X ⟶ S} {g : Y ⟶ S} (hP : IsPullback fst snd f g) (z : Sp C ⟶ P) : (pullbackHomeomorph hP z).1 = (z ≫ fst, z ≫ snd)

@[simp] theorem **PtsOver.prodHomeomorph_apply** {P S : Scheme.{0}} {fst : P ⟶ X} {snd : P ⟶ Y} {f : X ⟶ S} {g : Y ⟶ S} (hP : IsPullback fst snd f g) (s : Sp C ⟶ S) (z : PtsOver (fst ≫ f) s) : PtsOver.prodHomeomorph hP s z = (PtsOver.map fst rfl s z, PtsOver.map snd hP.w.symm s z)

theorem **ptsOverEquivAlgHom_apply** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) (x : PtsOver (spMap (algebraMap R A)) t) (a : A) : ptsOverEquivAlgHom A ht x a = valR x.1 ⊤ ((Scheme.ΓSpecIso (CommRingCat.of A)).inv.hom a)

@[simp] theorem **ptsOverEquivAlgHom_symm_val** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) (φ : A →ₐ[R] C) : ((ptsOverEquivAlgHom A ht).symm φ).1 = spMap φ.toRingHom

theorem **id_eq_spMap_algebraMap** (C : Type) [CommRing C] : 𝟙 (Sp C) = spMap (algebraMap C C)

theorem **range_coord** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) {σ : Type*} (a : σ → A) (ha : Function.Surjective (MvPolynomial.aeval (R

theorem **isEmbedding_ptsOverEquivAlgHom** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) : IsEmbedding fun x : PtsOver (spMap (algebraMap R A)) t => (ptsOverEquivAlgHom A ht x : A → C)

theorem **isEmbedding_coord** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) {σ : Type*} (a : σ → A) (ha : Function.Surjective (MvPolynomial.aeval (R

theorem **isClosedEmbedding_coord** [T1Space C] {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) {σ : Type*} (a : σ → A) (ha : Function.Surjective (MvPolynomial.aeval (R

theorem **aeval_mk_X** {σ : Type} (I : Ideal (MvPolynomial σ R)) (q : MvPolynomial σ R) : MvPolynomial.aeval (fun i => Ideal.Quotient.mk I (MvPolynomial.X i)) q = Ideal.Quotient.mk I q

theorem **zeroSetHomeomorph_apply** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) {σ : Type} (I : Ideal (MvPolynomial σ R)) (x : PtsOver (spMap (algebraMap R (MvPolynomial σ R ⧸ I))) t) (i : σ) : (zeroSetHomeomorph ht I x : σ → C) i = ptsOverEquivAlgHom _ ht x (Ideal.Quotient.mk I (MvPolynomial.X i))

theorem **affineSpaceHomeomorph_apply** {t : Sp C ⟶ Sp R} (ht : t = spMap (algebraMap R C)) (σ : Type) (x : PtsOver (spMap (algebraMap R (MvPolynomial σ R))) t) (i : σ) : affineSpaceHomeomorph ht σ x i = ptsOverEquivAlgHom _ ht x (MvPolynomial.X i)

theorem **isEmbedding_left** (A : Over (Sp C)) : IsEmbedding (overPt (A

@[simp] theorem **overHomeomorph_val** (A : Over (Sp C)) (x : 𝟙_ (Over (Sp C)) ⟶ A) : (overHomeomorph A x).1 = x.left

@[fun_prop] theorem **continuous_over_comp** {A B : Over (Sp C)} (g : A ⟶ B) : Continuous fun x : 𝟙_ (Over (Sp C)) ⟶ A => x ≫ g

@[simp] theorem **tensorHomeomorph_apply** [IsTopologicalRing C] [IsLocalRing C] [IsOpenUnits C] (A B : Over (Sp C)) (z : 𝟙_ (Over (Sp C)) ⟶ A ⊗ B) : tensorHomeomorph A B z = (z ≫ fst A B, z ≫ snd A B)

@[simp] theorem **tensorHomeomorph_symm_apply** [IsTopologicalRing C] [IsLocalRing C] [IsOpenUnits C] (A B : Over (Sp C)) (q : (𝟙_ (Over (Sp C)) ⟶ A) × (𝟙_ (Over (Sp C)) ⟶ B)) : (tensorHomeomorph A B).symm q = lift q.1 q.2

theorem **continuous_iff_valAt'** {B : Type*} [TopologicalSpace B] {e : B → (Sp ℂ ⟶ X)} : Continuous e ↔ ∀ U : X.Opens, IsOpen {b | ⊤ ≤ e b ⁻¹ᵁ U} ∧ ∀ f : Γ(X, U), ContinuousOn (fun b => valAt' (e b) U f) {b | ⊤ ≤ e b ⁻¹ᵁ U}

theorem **continuous_of_differentiableOn** {B : Type*} [NormedAddCommGroup B] [NormedSpace ℂ B] {e : B → (Sp ℂ ⟶ X)} (H : ∀ (U : X.Opens) (f : Γ(X, U)), IsOpen {b | ⊤ ≤ e b ⁻¹ᵁ U} ∧ DifferentiableOn ℂ (fun b => valAt' (e b) U f) {b | ⊤ ≤ e b ⁻¹ᵁ U}) : Continuous e

@[simp] theorem **conjHomeomorph_apply** (x : Sp ℂ ⟶ X) : conjHomeomorph X x = spMap (starRingEnd ℂ) ≫ x

theorem **valR_conjHomeomorph** (x : Sp ℂ ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (f : Γ(X, U)) : valR (conjHomeomorph X x) U f = starRingEnd ℂ (valR x U f)

theorem **ptsEquivCPt_val** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] (x : Pts p) : (ptsEquivCPt p x).1 = x.1 (IsLocalRing.closedPoint ℂ)

theorem **ptsEquivCPt_mem_iff** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] (x : Pts p) (U : X.Opens) : (ptsEquivCPt p x).1 ∈ U ↔ ⊤ ≤ x.1 ⁻¹ᵁ U

theorem **germ_stalkClosedPointTo_eq_valMap** {C : Type} [CommRing C] [IsLocalRing C] (x : Sp C ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) : X.presheaf.germ U (x (IsLocalRing.closedPoint C)) ((le_preimage_iff x U).1 h) ≫ Scheme.stalkClosedPointTo (R

theorem **valR_eq_val** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] (x : Pts p) (U : X.Opens) (h : (ptsEquivCPt p x).1 ∈ U) (f : Γ(X, U)) : valR x.1 U f = val p f (ptsEquivCPt p x).1 h

theorem **valAt_eq_val** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] (x : Pts p) (U : X.Opens) (h : ⊤ ≤ x.1 ⁻¹ᵁ U) (f : Γ(X, U)) : valAt x.1 U h f = val p f (ptsEquivCPt p x).1 ((ptsEquivCPt_mem_iff p x U).2 h)

theorem **resolutionHomeomorph_val** {K : Type} [Fintype K] [Nonempty K] {ι : K → Type} [∀ k, Fintype (ι k)] {P D : Ideal (MvPolynomial (Σ k, ι k) ℂ)} {hP : IsMultiHomogIdeal P} (w : WData P D hP) (x : Pts (w.π ≫ toSpecC hP)) : (resolutionHomeomorph w x).1 = x.1 (IsLocalRing.closedPoint ℂ)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.NormDetStrongContragredientDomRestrictAnnihilato

theorem **gramVol_nonneg** (v : ι → V) : 0 ≤ gramVol 𝕜 v

theorem **gramVol_sq** (v : ι → V) : ((gramVol 𝕜 v : ℝ) : 𝕜) ^ 2 = (gram 𝕜 v).det

theorem **gramVol_eq_of_sq** {v : ι → V} {r : ℝ} (hr : 0 ≤ r) (h : (r : 𝕜) ^ 2 = (gram 𝕜 v).det) : gramVol 𝕜 v = r

theorem **gramVol_of_orthonormal** {v : ι → V} (hv : Orthonormal 𝕜 v) : gramVol 𝕜 v = 1

theorem **gramVol_of_pairwise_orthogonal** {v : ι → V} (hv : Pairwise fun i j => ⟪v i, v j⟫_𝕜 = 0) : gramVol 𝕜 v = ∏ i, ‖v i‖

@[simp] theorem **gramVol_unique** [Unique ι] (v : ι → V) : gramVol 𝕜 v = ‖v default‖

@[simp] theorem **gramVol_of_isEmpty** [IsEmpty ι] (v : ι → V) : gramVol 𝕜 v = 1

theorem **gram_sum_smul** (v : ι → V) (Y : Matrix ι κ 𝕜) : gram 𝕜 (fun j => ∑ i, Y i j • v i) = Yᴴ * gram 𝕜 v * Y

theorem **gramVol_sum_smul** (v : ι → V) (Y : Matrix ι ι 𝕜) : gramVol 𝕜 (fun j => ∑ i, Y i j • v i) = ‖Y.det‖ * gramVol 𝕜 v

theorem **gramVol_eq_norm_det** (b : OrthonormalBasis ι 𝕜 V) (v : ι → V) : gramVol 𝕜 v = ‖(b.toBasis.toMatrix v).det‖

theorem **norm_det_inner** [FiniteDimensional 𝕜 V] (hι : Fintype.card ι = finrank 𝕜 V) (w v : ι → V) : ‖(Matrix.of fun i j => ⟪w i, v j⟫_𝕜).det‖ = gramVol 𝕜 w * gramVol 𝕜 v

theorem **gramVol_mul_gramVol_of_biorthogonal** [FiniteDimensional 𝕜 V] (hι : Fintype.card ι = finrank 𝕜 V) {w v : ι → V} (h : ∀ i j, ⟪w i, v j⟫_𝕜 = if i = j then 1 else 0) : gramVol 𝕜 w * gramVol 𝕜 v = 1

theorem **normDet_eq_gramVol** (T : V →ₗ[𝕜] W) (b : OrthonormalBasis ι 𝕜 V) : T.normDet = gramVol 𝕜 (fun i => T (b i))

theorem **normDet_mul_normDet_of_inner_eq** [FiniteDimensional 𝕜 W] {S T : V →ₗ[𝕜] W} (h : finrank 𝕜 V = finrank 𝕜 W) (hST : ∀ x y, ⟪S x, T y⟫_𝕜 = ⟪x, y⟫_𝕜) : S.normDet * T.normDet = 1

@[simp] theorem **detNorm_refl** : detNorm (LinearEquiv.refl 𝕜 V) = 1

@[simp] theorem **detNorm_linearIsometryEquiv** (f : V ≃ₗᵢ[𝕜] W) : detNorm f.toLinearEquiv = 1

theorem **inner_invAdjoint_apply** (T : V ≃ₗ[𝕜] W) (x y : V) : ⟪invAdjoint T x, T y⟫_𝕜 = ⟪x, y⟫_𝕜

theorem **detNorm_invAdjoint** (T : V ≃ₗ[𝕜] W) : detNorm (invAdjoint T) = (detNorm T)⁻¹

@[simp] theorem **dualQuotientEquivAnnihilator_apply** {E : Type*} [NormedAddCommGroup E] [NormedSpace 𝕜 E] (U : Submodule 𝕜 E) (φ : StrongDual 𝕜 (E ⧸ U)) (x : E) : (dualQuotientEquivAnnihilator U φ : StrongDual 𝕜 E) x = φ (Submodule.Quotient.mk x)

@[simp] theorem **dualQuotientEquivAnnihilator_symm_apply** {E : Type*} [NormedAddCommGroup E] [NormedSpace 𝕜 E] (U : Submodule 𝕜 E) (ψ : StrongDual.polarSubmodule 𝕜 U) (x : E) : (dualQuotientEquivAnnihilator U).symm ψ (Submodule.Quotient.mk x) = (ψ : StrongDual 𝕜 E) x

@[simp] theorem **dualQuotientAnnihilatorEquiv_mk** (U : Submodule 𝕜 V) [U.HasOrthogonalProjection] (φ : StrongDual 𝕜 V) (x : U) : dualQuotientAnnihilatorEquiv U (Submodule.Quotient.mk φ) x = φ x

@[simp] theorem **rieszEquiv_apply** (x y : V) : rieszEquiv 𝕜 V x y = ⟪x, y⟫_𝕜

@[simp] theorem **inner_rieszEquiv** (x y : V) : ⟪rieszEquiv 𝕜 V x, rieszEquiv 𝕜 V y⟫_𝕜 = ⟪y, x⟫_𝕜

@[simp] theorem **strongDualMap_apply** (T : V →ₗ[𝕜] W) (φ : StrongDual 𝕜 W) (x : V) : strongDualMap T φ x = φ (T x)

@[simp] theorem **strongContragredient_apply** (T : V ≃ₗ[𝕜] W) (φ : StrongDual 𝕜 V) (y : W) : strongContragredient T φ y = φ (T.symm y)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.NormSqRiemannThetaMulExpPeriodic

theorem **riemannThetaTerm_eq_thetaTerm** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) (n : ι → ℤ) : riemannThetaTerm a b Ω v n = ReciprocityTheory.PTheta.thetaTerm a b v Ω n

theorem **riemannTheta_eq_theta** (a b : ι → ℝ) (Ω : Matrix ι ι ℂ) (v : ι → ℂ) : riemannTheta a b Ω v = ReciprocityTheory.PTheta.theta a b v Ω

theorem **isSiegel_iff_mem_siegelH** {Ω : Matrix ι ι ℂ} : IsSiegel Ω ↔ Ω ∈ ReciprocityTheory.PTheta.SiegelH ι

theorem **siegelUpperHalfSpace_eq_siegelH** : siegelUpperHalfSpace ι = ReciprocityTheory.PTheta.SiegelH ι

theorem **not_isSiegel_neg_I** : ¬ IsSiegel !![-I]

theorem **not_isSiegel_indefinite** : ¬ IsSiegel !![I, 2 * I; 2 * I, I]

theorem **not_isSiegel_nonsymm** : ¬ IsSiegel !![I, 1; 0, I]

theorem **riemannTheta_zero_zero_half_period** (τ : ℂ) : riemannTheta 0 0 !![τ] ![(1 + τ) / 2] = 0

theorem **one_le_re_riemannTheta_I_smul_one** [DecidableEq ι] : 1 ≤ (riemannTheta (0 : ι → ℝ) 0 (I • (1 : Matrix ι ι ℂ)) 0).re

theorem **riemannTheta_eq_thetaChar** {g : ℕ} (a b : Fin g → ℝ) (Ω : Matrix (Fin g) (Fin g) ℂ) (v : Fin g → ℂ) : riemannTheta a b Ω v = ReciprocityTheory.ChartRestriction.G.Geo.Tor.thetaChar Ω (fun i => a i) (fun i => b i) v

theorem **riemannTheta_zero_matrix** : riemannTheta 0 0 !![(0 : ℂ)] ![0] = 0

theorem **thetaSummable** : ReciprocityTheory.PTheta.ThetaSummableStatement

theorem **riemannTheta_not_constant** : riemannTheta 0 0 !![I] ![(1 + I) / 2] ≠ riemannTheta 0 0 !![I] 0

theorem **riemannTheta_half_add_tau** (α β : ℤ) (z τ : ℂ) : riemannTheta (fun _ => (α : ℝ) / 2) (fun _ => (β : ℝ) / 2) !![τ] ![z + τ] = (-1) ^ β * cexp (-π * I * τ - 2 * π * I * z) * riemannTheta (fun _ => (α : ℝ) / 2) (fun _ => (β : ℝ) / 2) !![τ] ![z]

theorem **norm_riemannTheta_real_coords** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b x y : ι → ℝ) : ‖riemannTheta a b Ω ((fun i => (x i : ℂ)) + Ω *ᵥ fun i => (y i : ℂ))‖ = Real.exp (π * (y ⬝ᵥ (Ω.map Complex.im *ᵥ y))) * ‖riemannTheta (a + y) (b + x) Ω 0‖

theorem **norm_sq_riemannTheta_mul_exp_periodic** {Ω : Matrix ι ι ℂ} (h : Ω.IsSymm) (a b x y : ι → ℝ) (m k : ι → ℤ) : ‖riemannTheta a b Ω ((fun i => ((x + fun i => (m i : ℝ)) i : ℂ)) + Ω *ᵥ fun i => ((y + fun i => (k i : ℝ)) i : ℂ))‖ ^ 2 * Real.exp (-2 * π * ((y + fun i => (k i : ℝ)) ⬝ᵥ (Ω.map Complex.im *ᵥ (y + fun i => (k i : ℝ))))) = ‖riemannTheta a b Ω ((fun i => (x i : ℂ)) + Ω *ᵥ fun i => (y i : ℂ))‖ ^ 2 * Real.exp (-2 * π * (y ⬝ᵥ (Ω.map Complex.im *ᵥ y)))

theorem **not_isSiegel_zero_nat** : ¬ IsSiegel (0 : Matrix ℕ ℕ ℂ)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.SmulOneLeAndLeSmulOneIffEigenvalues

theorem **notMem_coordDivisor** {S : Finset κ} {w : X → κ → ℂ} {x : X} : x ∉ coordDivisor S w ↔ ∀ j ∈ S, w x j ≠ 0

theorem **mem_coordDivisor_iff_prod_eq_zero** {S : Finset κ} {w : X → κ → ℂ} {x : X} : x ∈ coordDivisor S w ↔ ∏ j ∈ S, w x j = 0

theorem **isClosed_coordDivisor** [TopologicalSpace X] (S : Finset κ) {w : X → κ → ℂ} (hw : ∀ j ∈ S, Continuous fun x => w x j) : IsClosed (coordDivisor S w)

theorem **one_le_logWeight** (S : Finset κ) (z : κ → ℂ) : 1 ≤ lwP₂ S z

theorem **logWeight_pos** (S : Finset κ) (z : κ → ℂ) : 0 < lwP₂ S z

theorem **abs_log_norm_le_logWeight** {S : Finset κ} (z : κ → ℂ) {j : κ} (hj : j ∈ S) : |Real.log ‖z j‖| ≤ lwP₂ S z

theorem **mul_logWeight_pow_le** (S : Finset κ) (z : κ → ℂ) {C C' : ℝ} {N N' : ℕ} (hC : C ≤ C') (hC' : 0 ≤ C') (hN : N ≤ N') : C * lwP₂ S z ^ N ≤ C' * lwP₂ S z ^ N'

theorem **exists_one_le** {f : X → E} (hf : HasLogGrowth S w U f) {K : Set X} (hKU : K ⊆ U) (hK : IsCompact K) : ∃ (C : ℝ) (N : ℕ), 1 ≤ C ∧ ∀ x ∈ K, x ∉ coordDivisor S w → ‖f x‖ ≤ C * lwP₂ S (w x) ^ N

theorem **mono** {f : X → E} (hf : HasLogGrowth S w U f) {V : Set X} (hV : V ⊆ U) : HasLogGrowth S w V f

theorem **of_norm_le** {f : X → E} {g : X → F} (hg : HasLogGrowth S w U g) (h : ∀ x ∈ U, x ∉ coordDivisor S w → ‖f x‖ ≤ ‖g x‖) : HasLogGrowth S w U f

theorem **of_le** {f : X → E} {g : X → ℝ} (hg : HasLogGrowth S w U g) (h : ∀ x ∈ U, x ∉ coordDivisor S w → ‖f x‖ ≤ g x) : HasLogGrowth S w U f

theorem **congr** {f g : X → E} (hg : HasLogGrowth S w U g) (h : ∀ x ∈ U, x ∉ coordDivisor S w → f x = g x) : HasLogGrowth S w U f

theorem **norm** {f : X → E} (hf : HasLogGrowth S w U f) : HasLogGrowth S w U fun x => ‖f x‖

theorem **of_continuousOn** {f : X → E} (hf : ContinuousOn f U) : HasLogGrowth S w U f

theorem **const** (c : E) : HasLogGrowth S w U fun _ => c

theorem **lwP₂** : HasLogGrowth S w U fun x => lwP₂ S (w x)

theorem **log_norm** {j : κ} (hj : j ∈ S) : HasLogGrowth S w U fun x => Real.log ‖w x j‖

theorem **add** {f g : X → E} (hf : HasLogGrowth S w U f) (hg : HasLogGrowth S w U g) : HasLogGrowth S w U fun x => f x + g x

theorem **neg** {f : X → E} (hf : HasLogGrowth S w U f) : HasLogGrowth S w U fun x => -f x

theorem **sub** {f g : X → E} (hf : HasLogGrowth S w U f) (hg : HasLogGrowth S w U g) : HasLogGrowth S w U fun x => f x - g x

theorem **sum** {α : Type*} {s : Finset α} {f : α → X → E} (hf : ∀ a ∈ s, HasLogGrowth S w U (f a)) : HasLogGrowth S w U fun x => ∑ a ∈ s, f a x

theorem **mul** {R : Type*} [SeminormedRing R] {f g : X → R} (hf : HasLogGrowth S w U f) (hg : HasLogGrowth S w U g) : HasLogGrowth S w U fun x => f x * g x

theorem **smul** {𝕜 : Type*} [NormedField 𝕜] [NormedSpace 𝕜 E] {c : X → 𝕜} {f : X → E} (hc : HasLogGrowth S w U c) (hf : HasLogGrowth S w U f) : HasLogGrowth S w U fun x => c x • f x

theorem **prod** {R α : Type*} [SeminormedCommRing R] {s : Finset α} {f : α → X → R} (hf : ∀ a ∈ s, HasLogGrowth S w U (f a)) : HasLogGrowth S w U fun x => ∏ a ∈ s, f a x

theorem **pow** {R : Type*} [SeminormedRing R] {f : X → R} (hf : HasLogGrowth S w U f) (k : ℕ) : HasLogGrowth S w U fun x => f x ^ k

theorem **star** {f : X → ℂ} (hf : HasLogGrowth S w U f) : HasLogGrowth S w U fun x => star (f x)

theorem **comp_of_logWeight** {f : X → E} (hf : HasLogGrowth S w U f) {S' : Finset κ'} {w' : X' → κ' → ℂ} {U' : Set X'} {π : X' → X} (hπ : ContinuousOn π U') (hU : MapsTo π U' U) (hB : ∀ x ∈ U', x ∉ coordDivisor S' w' → π x ∉ coordDivisor S w) (hw : HasLogGrowth S' w' U' fun x => GlobalTheory.lwP₂ S (w (π x))) : HasLogGrowth S' w' U' fun x => f (π x)

theorem **notMem** (h : IsUnitMulMonomialOn S v S' w' U') {x : X'} (hx : x ∈ U') (hB : x ∉ coordDivisor S' w') : x ∉ coordDivisor S v

theorem **hasLogGrowth_logWeight** (h : IsUnitMulMonomialOn S v S' w' U') : HasLogGrowth S' w' U' fun x => lwP₂ S (v x)

theorem **HasLogGrowth.comp** {f : X → E} (hf : HasLogGrowth S w U f) {S' : Finset κ'} {w' : X' → κ' → ℂ} {U' : Set X'} {π : X' → X} (hπ : ContinuousOn π U') (hU : MapsTo π U' U) (hw : IsUnitMulMonomialOn S (fun x => w (π x)) S' w' U') : HasLogGrowth S' w' U' fun x => f (π x)

theorem **IsLogBoundedScalar.pos** (hg : IsLogBoundedScalar S w U g) {x : X} (hx : x ∈ U) (hB : x ∉ coordDivisor S w) : 0 < g x

theorem **isLogBoundedScalar_iff** : IsLogBoundedScalar S w U g ↔ (∀ x ∈ U, x ∉ coordDivisor S w → 0 < g x) ∧ HasLogGrowth S w U g ∧ HasLogGrowth S w U fun x => (g x)⁻¹

theorem **isLogBoundedScalar_iff_abs_log_le** : IsLogBoundedScalar S w U g ↔ (∀ x ∈ U, x ∉ coordDivisor S w → 0 < g x) ∧ ∀ K ⊆ U, IsCompact K → ∃ (C : ℝ) (N : ℕ), ∀ x ∈ K, x ∉ coordDivisor S w → |Real.log (g x)| ≤ C + N * Real.log (lwP₂ S (w x))

theorem **hasLogGrowth** (hg : IsLogBoundedScalar S w U g) : HasLogGrowth S w U g

theorem **hasLogGrowth_inv** (hg : IsLogBoundedScalar S w U g) : HasLogGrowth S w U fun x => (g x)⁻¹

theorem **mono** (hg : IsLogBoundedScalar S w U g) {V : Set X} (hV : V ⊆ U) : IsLogBoundedScalar S w V g

theorem **congr** (hg : IsLogBoundedScalar S w U g) (h : ∀ x ∈ U, x ∉ coordDivisor S w → g' x = g x) : IsLogBoundedScalar S w U g'

theorem **of_continuousOn** (hg : ContinuousOn g U) (h0 : ∀ x ∈ U, 0 < g x) : IsLogBoundedScalar S w U g

theorem **const** {c : ℝ} (hc : 0 < c) : IsLogBoundedScalar S w U fun _ => c

theorem **lwP₂** : IsLogBoundedScalar S w U fun x => lwP₂ S (w x)

theorem **mul** (hg : IsLogBoundedScalar S w U g) (hg' : IsLogBoundedScalar S w U g') : IsLogBoundedScalar S w U fun x => g x * g' x

theorem **inv** (hg : IsLogBoundedScalar S w U g) : IsLogBoundedScalar S w U fun x => (g x)⁻¹

theorem **prod** {α : Type*} {s : Finset α} {f : α → X → ℝ} (hf : ∀ a ∈ s, IsLogBoundedScalar S w U (f a)) : IsLogBoundedScalar S w U fun x => ∏ a ∈ s, f a x

theorem **rpow** (hg : IsLogBoundedScalar S w U g) (t : ℝ) : IsLogBoundedScalar S w U fun x => g x ^ t

theorem **zpow** (hg : IsLogBoundedScalar S w U g) (k : ℤ) : IsLogBoundedScalar S w U fun x => g x ^ k

theorem **pow** (hg : IsLogBoundedScalar S w U g) (k : ℕ) : IsLogBoundedScalar S w U fun x => g x ^ k

theorem **of_pow** {k : ℕ} (hk : k ≠ 0) (h0 : ∀ x ∈ U, x ∉ coordDivisor S w → 0 ≤ g x) (hg : IsLogBoundedScalar S w U fun x => g x ^ k) : IsLogBoundedScalar S w U g

theorem **comp** (hg : IsLogBoundedScalar S w U g) {S' : Finset κ'} {w' : X' → κ' → ℂ} {U' : Set X'} {π : X' → X} (hπ : ContinuousOn π U') (hU : MapsTo π U' U) (hw : IsUnitMulMonomialOn S (fun x => w (π x)) S' w' U') : IsLogBoundedScalar S' w' U' fun x => g (π x)

theorem **congr** {A A' : X → Matrix m n ℂ} (hA : HasLogGrowthEntries S w U A) (h : ∀ x ∈ U, x ∉ coordDivisor S w → A' x = A x) : HasLogGrowthEntries S w U A'

theorem **of_continuousOn** {A : X → Matrix m n ℂ} (hA : ContinuousOn A U) : HasLogGrowthEntries S w U A

theorem **mul** [Fintype m] {A : X → Matrix l m ℂ} {A' : X → Matrix m n ℂ} (hA : HasLogGrowthEntries S w U A) (hA' : HasLogGrowthEntries S w U A') : HasLogGrowthEntries S w U fun x => A x * A' x

theorem **conjTranspose** {A : X → Matrix m n ℂ} (hA : HasLogGrowthEntries S w U A) : HasLogGrowthEntries S w U fun x => (A x)ᴴ

theorem **transpose** {A : X → Matrix m n ℂ} (hA : HasLogGrowthEntries S w U A) : HasLogGrowthEntries S w U fun x => (A x)ᵀ

theorem **submatrix** {m' n' : Type*} {A : X → Matrix m n ℂ} (hA : HasLogGrowthEntries S w U A) (r : m' → m) (c : n' → n) : HasLogGrowthEntries S w U fun x => (A x).submatrix r c

theorem **det** [Fintype n] [DecidableEq n] {A : X → Matrix n n ℂ} (hA : HasLogGrowthEntries S w U A) : HasLogGrowth S w U fun x => (A x).det

theorem **trace** [Fintype n] {A : X → Matrix n n ℂ} (hA : HasLogGrowthEntries S w U A) : HasLogGrowth S w U fun x => (A x).trace

theorem **real_smul_one_le_real_smul_one** [DecidableEq n] {a b : ℝ} (h : a ≤ b) : a • (1 : Matrix n n ℂ) ≤ b • (1 : Matrix n n ℂ)

theorem **inv_smul_one_le_and_le_smul_one_mono** [DecidableEq n] {G : Matrix n n ℂ} {a b : ℝ} (ha : 0 < a) (hab : a ≤ b) (h : a⁻¹ • (1 : Matrix n n ℂ) ≤ G ∧ G ≤ a • (1 : Matrix n n ℂ)) : b⁻¹ • (1 : Matrix n n ℂ) ≤ G ∧ G ≤ b • (1 : Matrix n n ℂ)

theorem **norm_apply_le_of_le_smul_one** [DecidableEq n] {G : Matrix n n ℂ} (hG : G.PosSemidef) {a : ℝ} (h : G ≤ a • (1 : Matrix n n ℂ)) (i j : n) : ‖G i j‖ ≤ a

theorem **inv_smul_one_le_inv_and_inv_le_smul_one** {G : Matrix n n ℂ} {a : ℝ} (ha : 0 < a) (h : a⁻¹ • (1 : Matrix n n ℂ) ≤ G ∧ G ≤ a • (1 : Matrix n n ℂ)) : a⁻¹ • (1 : Matrix n n ℂ) ≤ G⁻¹ ∧ G⁻¹ ≤ a • (1 : Matrix n n ℂ)

theorem **posDef_fromBlocks_zero** {A : Matrix m m ℂ} {D : Matrix n n ℂ} (hA : A.PosDef) (hD : D.PosDef) : (fromBlocks A 0 0 D).PosDef

theorem **schurComplement₂₂_eq_swap** (G : Matrix (m ⊕ n) (m ⊕ n) ℂ) : schurComplement₂₂ G = schurComplement₁₁ (G.submatrix Sum.swap Sum.swap)

theorem **schurComplement₁₁_eq_inv** {G : Matrix (m ⊕ n) (m ⊕ n) ℂ} (hG : G.PosDef) : schurComplement₁₁ G = (G⁻¹.submatrix Sum.inr Sum.inr)⁻¹

theorem **isLeast_schurComplement₁₁** {G : Matrix (m ⊕ n) (m ⊕ n) ℂ} (hG : G.PosDef) (v : n → ℂ) : IsLeast (range fun u : m → ℂ => (star (Sum.elim u v) ⬝ᵥ G *ᵥ Sum.elim u v).re) (star v ⬝ᵥ schurComplement₁₁ G *ᵥ v).re

theorem **compoundMatrix_mul** [Fintype n] (k : ℕ) (A : Matrix l n ℂ) (A' : Matrix n o ℂ) : compoundMatrix k (A * A') = compoundMatrix k A * compoundMatrix k A'

theorem **compoundMatrix_conjTranspose** (k : ℕ) (A : Matrix l n ℂ) : compoundMatrix k Aᴴ = (compoundMatrix k A)ᴴ

theorem **compoundMatrix_one** (k : ℕ) : compoundMatrix k (1 : Matrix n n ℂ) = 1

theorem **compoundMatrix_inv** (k : ℕ) {A : Matrix n n ℂ} (hA : IsUnit A) : (compoundMatrix k A)⁻¹ = compoundMatrix k A⁻¹

theorem **posDef_compoundMatrix** (k : ℕ) {G : Matrix n n ℂ} (hG : G.PosDef) : (compoundMatrix k G).PosDef

theorem **IsLogBounded.posDef** (hG : IsLogBounded S w U G) {x : X} (hx : x ∈ U) (hB : x ∉ coordDivisor S w) : (G x).PosDef

theorem **mono** (hG : IsLogBounded S w U G) {V : Set X} (hV : V ⊆ U) : IsLogBounded S w V G

theorem **congr** (hG : IsLogBounded S w U G) (h : ∀ x ∈ U, x ∉ coordDivisor S w → G' x = G x) : IsLogBounded S w U G'

theorem **of_le** {g : X → ℝ} (hg : HasLogGrowth S w U g) (h : ∀ x ∈ U, x ∉ coordDivisor S w → 0 < g x ∧ (g x)⁻¹ • (1 : Matrix n n ℂ) ≤ G x ∧ G x ≤ g x • (1 : Matrix n n ℂ)) : IsLogBounded S w U G

theorem **hasLogGrowthEntries** (hG : IsLogBounded S w U G) : HasLogGrowthEntries S w U G

theorem **transpose** (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => (G x)ᵀ

theorem **submatrix** (hG : IsLogBounded S w U G) {e : m → n} (he : Function.Injective e) : IsLogBounded S w U fun x => (G x).submatrix e e

theorem **reindex** (hG : IsLogBounded S w U G) (e : n ≃ m) : IsLogBounded S w U fun x => Matrix.reindex e e (G x)

theorem **toBlocks₁₁** {G : X → Matrix (m ⊕ n) (m ⊕ n) ℂ} (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => (G x).toBlocks₁₁

theorem **toBlocks₂₂** {G : X → Matrix (m ⊕ n) (m ⊕ n) ℂ} (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => (G x).toBlocks₂₂

theorem **inv** (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => (G x)⁻¹

theorem **hasLogGrowthEntries_inv** (hG : IsLogBounded S w U G) : HasLogGrowthEntries S w U fun x => (G x)⁻¹

theorem **of_trace** (h0 : ∀ x ∈ U, x ∉ coordDivisor S w → (G x).PosDef) (h1 : HasLogGrowth S w U fun x => (G x).trace) (h2 : HasLogGrowth S w U fun x => (G x)⁻¹.trace) : IsLogBounded S w U G

theorem **isLogBounded_iff_entries** : IsLogBounded S w U G ↔ (∀ x ∈ U, x ∉ coordDivisor S w → (G x).PosDef) ∧ HasLogGrowthEntries S w U G ∧ HasLogGrowthEntries S w U fun x => (G x)⁻¹

theorem **isLogBounded_iff_exists_scalar** : IsLogBounded S w U G ↔ ∃ g : X → ℝ, IsLogBoundedScalar S w U g ∧ ∀ x ∈ U, x ∉ coordDivisor S w → (g x)⁻¹ • (1 : Matrix n n ℂ) ≤ G x ∧ G x ≤ g x • (1 : Matrix n n ℂ)

theorem **continuousOn_matrix_inv** {A : X → Matrix n n ℂ} (hA : ContinuousOn A U) (hu : ∀ x ∈ U, IsUnit (A x)) : ContinuousOn (fun x => (A x)⁻¹) U

theorem **of_continuousOn** (hG : ContinuousOn G U) (h0 : ∀ x ∈ U, (G x).PosDef) : IsLogBounded S w U G

theorem **of_le_of_le** {G₁ G₂ : X → Matrix n n ℂ} (h₁ : IsLogBounded S w U G₁) (h₂ : IsLogBounded S w U G₂) (h : ∀ x ∈ U, x ∉ coordDivisor S w → G₁ x ≤ G x ∧ G x ≤ G₂ x) : IsLogBounded S w U G

theorem **smul** {g : X → ℝ} (hg : IsLogBoundedScalar S w U g) (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => g x • G x

theorem **smul_one** {g : X → ℝ} (hg : IsLogBoundedScalar S w U g) : IsLogBounded S w U fun x => g x • (1 : Matrix n n ℂ)

theorem **diagonal** {g : n → X → ℝ} (hg : ∀ i, IsLogBoundedScalar S w U (g i)) : IsLogBounded S w U fun x => Matrix.diagonal fun i => (g i x : ℂ)

theorem **re_apply_self** (hG : IsLogBounded S w U G) (i : n) : IsLogBoundedScalar S w U fun x => (G x i i).re

theorem **re_det** (hG : IsLogBounded S w U G) : IsLogBoundedScalar S w U fun x => (G x).det.re

theorem **re_det_zpow** (hG : IsLogBounded S w U G) (k : ℤ) : IsLogBoundedScalar S w U fun x => (G x).det.re ^ k

theorem **isLogBounded_of_unique_iff** [Unique n] {g : X → ℝ} : (IsLogBounded S w U fun x => Matrix.of fun _ _ : n => (g x : ℂ)) ↔ IsLogBoundedScalar S w U g

theorem **det_zpow** (hG : IsLogBounded S w U G) (k : ℤ) (p : Type*) [Unique p] [DecidableEq p] : IsLogBounded S w U fun x => Matrix.of fun _ _ : p => (G x).det ^ k

theorem **det** (hG : IsLogBounded S w U G) (p : Type*) [Unique p] [DecidableEq p] : IsLogBounded S w U fun x => Matrix.of fun _ _ : p => (G x).det

theorem **compoundMatrix** {n : Type*} [Fintype n] [LinearOrder n] {G : X → Matrix n n ℂ} (hG : IsLogBounded S w U G) (k : ℕ) : IsLogBounded S w U fun x => compoundMatrix k (G x)

theorem **comp_of_logWeight** (hG : IsLogBounded S w U G) {S' : Finset κ'} {w' : X' → κ' → ℂ} {U' : Set X'} {π : X' → X} (hπ : ContinuousOn π U') (hU : MapsTo π U' U) (hB : ∀ x ∈ U', x ∉ coordDivisor S' w' → π x ∉ coordDivisor S w) (hw : HasLogGrowth S' w' U' fun x => lwP₂ S (w (π x))) : IsLogBounded S' w' U' fun x => G (π x)

theorem **comp** (hG : IsLogBounded S w U G) {S' : Finset κ'} {w' : X' → κ' → ℂ} {U' : Set X'} {π : X' → X} (hπ : ContinuousOn π U') (hU : MapsTo π U' U) (hw : IsUnitMulMonomialOn S (fun x => w (π x)) S' w' U') : IsLogBounded S' w' U' fun x => G (π x)

theorem **conj_of_isLogBounded** (hG : IsLogBounded S w U G) {A : X → Matrix n m ℂ} (hA : IsLogBounded S w U fun x => (A x)ᴴ * A x) : IsLogBounded S w U fun x => (A x)ᴴ * G x * A x

theorem **conj_of_injective** (hG : IsLogBounded S w U G) {A : X → Matrix n m ℂ} (hA : ContinuousOn A U) (hinj : ∀ x ∈ U, Function.Injective (A x).mulVec) : IsLogBounded S w U fun x => (A x)ᴴ * G x * A x

theorem **conj** (hG : IsLogBounded S w U G) {A : X → Matrix n n ℂ} (hA : ContinuousOn A U) (hu : ∀ x ∈ U, IsUnit (A x)) : IsLogBounded S w U fun x => (A x)ᴴ * G x * A x

theorem **fromBlocks** (hH : IsLogBounded S w U H) (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => Matrix.fromBlocks (H x) 0 0 (G x)

theorem **schurComplement₁₁** {G : X → Matrix (m ⊕ n) (m ⊕ n) ℂ} (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => schurComplement₁₁ (G x)

theorem **schurComplement₂₂** {G : X → Matrix (m ⊕ n) (m ⊕ n) ℂ} (hG : IsLogBounded S w U G) : IsLogBounded S w U fun x => schurComplement₂₂ (G x)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.SumFinsuppAntidiagProdPowDivMultiFactorial

theorem **l1Norm_eq_sum_of_subset** {P : MvPolynomial σ K} {s : Finset (σ →₀ ℕ)} (h : P.support ⊆ s) : l1Norm P = ∑ α ∈ s, ‖P.coeff α‖

theorem **l1Norm_nonneg** (P : MvPolynomial σ K) : 0 ≤ l1Norm P

theorem **sum_norm_coeff_le_l1Norm** (P : MvPolynomial σ K) (s : Finset (σ →₀ ℕ)) : ∑ α ∈ s, ‖P.coeff α‖ ≤ l1Norm P

theorem **norm_coeff_le_l1Norm** (P : MvPolynomial σ K) (α : σ →₀ ℕ) : ‖P.coeff α‖ ≤ l1Norm P

@[simp] theorem **l1Norm_zero** : l1Norm (0 : MvPolynomial σ K) = 0

@[simp] theorem **l1Norm_eq_zero** {P : MvPolynomial σ K} : l1Norm P = 0 ↔ P = 0

@[simp] theorem **l1Norm_monomial** (α : σ →₀ ℕ) (c : K) : l1Norm (monomial α c) = ‖c‖

@[simp] theorem **l1Norm_C** (c : K) : l1Norm (C c : MvPolynomial σ K) = ‖c‖

@[simp] theorem **l1Norm_one** [NormOneClass K] : l1Norm (1 : MvPolynomial σ K) = 1

@[simp] theorem **l1Norm_X** [NormOneClass K] (i : σ) : l1Norm (X i : MvPolynomial σ K) = 1

@[simp] theorem **l1Norm_neg** (P : MvPolynomial σ K) : l1Norm (-P) = l1Norm P

theorem **l1Norm_add_le** (P Q : MvPolynomial σ K) : l1Norm (P + Q) ≤ l1Norm P + l1Norm Q

theorem **l1Norm_sum_le** {ι : Type*} (s : Finset ι) (f : ι → MvPolynomial σ K) : l1Norm (∑ i ∈ s, f i) ≤ ∑ i ∈ s, l1Norm (f i)

theorem **l1Norm_mul_le** (P Q : MvPolynomial σ K) : l1Norm (P * Q) ≤ l1Norm P * l1Norm Q

@[simp] theorem **l1RingNorm_apply** (P : MvPolynomial σ K) : l1RingNorm σ K P = l1Norm P

@[simp] theorem **l1RingSeminorm_apply** (P : MvPolynomial σ K) : l1RingSeminorm σ K P = l1Norm P

theorem **l1Norm_smul_le** (c : K) (P : MvPolynomial σ K) : l1Norm (c • P) ≤ ‖c‖ * l1Norm P

theorem **l1Norm_prod_le** [NormOneClass K] {ι : Type*} (s : Finset ι) (f : ι → MvPolynomial σ K) : l1Norm (∏ i ∈ s, f i) ≤ ∏ i ∈ s, l1Norm (f i)

theorem **l1Norm_pow_le** [NormOneClass K] (P : MvPolynomial σ K) (n : ℕ) : l1Norm (P ^ n) ≤ l1Norm P ^ n

theorem **l1Norm_prod_pow_le** [NormOneClass K] {ι : Type*} (s : Finset ι) (f : ι → MvPolynomial σ K) (c : ι → ℕ) (B : ι → ℝ) (h : ∀ i ∈ s, l1Norm (f i) ≤ B i) : l1Norm (∏ i ∈ s, f i ^ c i) ≤ ∏ i ∈ s, B i ^ c i

theorem **l1Norm_rename_of_injective** {f : σ → τ} (hf : Function.Injective f) (P : MvPolynomial σ K) : l1Norm (rename f P) = l1Norm P

theorem **l1Norm_map** {R : Type*} [CommSemiring R] (φ : R →+* K) (P : MvPolynomial σ R) : l1Norm (map φ P) = ∑ α ∈ P.support, ‖φ (P.coeff α)‖

theorem **sum_le_sqrt_card_mul_sqrt_sum_sq** {ι : Type*} (s : Finset ι) (f : ι → ℝ) : ∑ i ∈ s, f i ≤ √(#s) * √(∑ i ∈ s, f i ^ 2)

theorem **l1Norm_linearForm_le_sum** {ι : Type*} (s : Finset ι) (v : ι → σ) (e : ι → K) : l1Norm (∑ k ∈ s, C (e k) * X (v k) : MvPolynomial σ K) ≤ ∑ k ∈ s, ‖e k‖

theorem **l1Norm_linearForm_eq_sum** {ι : Type*} (s : Finset ι) {v : ι → σ} (hv : Function.Injective v) (e : ι → K) : l1Norm (∑ k ∈ s, C (e k) * X (v k) : MvPolynomial σ K) = ∑ k ∈ s, ‖e k‖

theorem **l1Norm_linearForm_le** {ι : Type*} [Fintype ι] (v : ι → σ) (e : ι → K) : l1Norm (∑ k, C (e k) * X (v k) : MvPolynomial σ K) ≤ √(Fintype.card ι) * √(∑ k, ‖e k‖ ^ 2)

theorem **l1Norm_det_le** [NormOneClass K] {ι : Type*} [Fintype ι] [DecidableEq ι] (M : Matrix ι ι (MvPolynomial σ K)) {R : ℝ} (h : ∀ a b, l1Norm (M a b) ≤ R) : l1Norm M.det ≤ (Fintype.card ι).factorial * R ^ Fintype.card ι

theorem **factorial_mul_pow_le** (i : ℕ) {R : ℝ} (hR : 0 ≤ R) : (i.factorial : ℝ) * R ^ i ≤ (i * R) ^ i

theorem **l1Norm_det_le_pow** [NormOneClass K] {ι : Type*} [Fintype ι] [DecidableEq ι] (M : Matrix ι ι (MvPolynomial σ K)) {R : ℝ} (h : ∀ a b, l1Norm (M a b) ≤ R) : l1Norm M.det ≤ (Fintype.card ι * R) ^ Fintype.card ι

theorem **degree_le_totalDegree** {R : Type*} [CommSemiring R] {P : MvPolynomial σ R} {α : σ →₀ ℕ} (h : α ∈ P.support) : α.degree ≤ P.totalDegree

theorem **l1Norm_bind_le_sum_norm_coeff_mul_pow_degree** [NormOneClass K] (f : σ → MvPolynomial τ K) (P : MvPolynomial σ K) {R : ℝ} (h : ∀ i, l1Norm (f i) ≤ R) : l1Norm (bind₁ f P) ≤ ∑ α ∈ P.support, ‖P.coeff α‖ * R ^ α.degree

theorem **sum_norm_coeff_mul_pow_le** (P : MvPolynomial σ K) {R : ℝ} (hR : 0 ≤ R) {d : ℕ} (hd : P.totalDegree ≤ d) : ∑ α ∈ P.support, ‖P.coeff α‖ * R ^ α.degree ≤ l1Norm P * max 1 R ^ d

theorem **sum_norm_coeff_mul_pow_of_isHomogeneous** {P : MvPolynomial σ K} {d : ℕ} (hP : P.IsHomogeneous d) (R : ℝ) : ∑ α ∈ P.support, ‖P.coeff α‖ * R ^ α.degree = l1Norm P * R ^ d

theorem **l1Norm_bind_le_mul_pow_of_isHomogeneous** [NormOneClass K] (f : σ → MvPolynomial τ K) {P : MvPolynomial σ K} {d : ℕ} (hP : P.IsHomogeneous d) {R : ℝ} (h : ∀ i, l1Norm (f i) ≤ R) : l1Norm (bind₁ f P) ≤ l1Norm P * R ^ d

theorem **norm_eval_le_sum** [NormOneClass K] (x : σ → K) (P : MvPolynomial σ K) {R : ℝ} (h : ∀ i, ‖x i‖ ≤ R) : ‖eval x P‖ ≤ ∑ α ∈ P.support, ‖P.coeff α‖ * R ^ α.degree

theorem **norm_eval_le_of_le** [NormOneClass K] (x : σ → K) (P : MvPolynomial σ K) {R : ℝ} (h : ∀ i, ‖x i‖ ≤ R) {d : ℕ} (hd : P.totalDegree ≤ d) : ‖eval x P‖ ≤ l1Norm P * max 1 R ^ d

theorem **norm_eval_le** [NormOneClass K] [Fintype σ] (x : σ → K) (P : MvPolynomial σ K) : ‖eval x P‖ ≤ l1Norm P * max 1 ‖x‖ ^ P.totalDegree

@[simp] theorem **mem_monomialsLE** {d : ℕ} {α : σ →₀ ℕ} : α ∈ monomialsLE σ d ↔ α.degree ≤ d

theorem **card_monomialsLE** (σ : Type*) [Fintype σ] [DecidableEq σ] (d : ℕ) : #(monomialsLE σ d) = (Fintype.card σ + d).choose d

theorem **choose_add_le_one_add_pow** (N d : ℕ) : (N + d).choose d ≤ (1 + d) ^ N

theorem **support_subset_monomialsLE** {R : Type*} [CommSemiring R] {P : MvPolynomial σ R} {d : ℕ} : P.support ⊆ monomialsLE σ d ↔ P.totalDegree ≤ d

theorem **card_support_le_choose** {R : Type*} [CommSemiring R] {P : MvPolynomial σ R} {d : ℕ} (h : P.totalDegree ≤ d) : #P.support ≤ (Fintype.card σ + d).choose d

theorem **ncard_setOf_degree_le** (d : ℕ) : {α : σ →₀ ℕ | α.degree ≤ d}.ncard = (Fintype.card σ + d).choose d

theorem **finrank_restrictTotalDegree** (R : Type*) [CommRing R] [Nontrivial R] (d : ℕ) : Module.finrank R (restrictTotalDegree σ R d) = (Fintype.card σ + d).choose d

theorem **l2Norm_nonneg** (P : MvPolynomial σ K) : 0 ≤ l2Norm P

theorem **fubiniStudy_l2Norm_sq_eq** (P : MvPolynomial σ K) : l2Norm P ^ 2 = ∑ α ∈ P.support, ‖P.coeff α‖ ^ 2

theorem **l2Norm_eq_sqrt_sum_of_subset** {P : MvPolynomial σ K} {s : Finset (σ →₀ ℕ)} (h : P.support ⊆ s) : l2Norm P = √(∑ α ∈ s, ‖P.coeff α‖ ^ 2)

@[simp] theorem **l2Norm_zero** : l2Norm (0 : MvPolynomial σ K) = 0

@[simp] theorem **l2Norm_monomial** (α : σ →₀ ℕ) (c : K) : l2Norm (monomial α c) = ‖c‖

@[simp] theorem **l2Norm_neg** (P : MvPolynomial σ K) : l2Norm (-P) = l2Norm P

theorem **vectorL2Norm_const_mul** [NormMulClass K] (c : K) (P : MvPolynomial σ K) : l2Norm (c • P) = ‖c‖ * l2Norm P

theorem **norm_coeff_le_l2Norm** (P : MvPolynomial σ K) (α : σ →₀ ℕ) : ‖P.coeff α‖ ≤ l2Norm P

@[simp] theorem **l2Norm_eq_zero** {P : MvPolynomial σ K} : l2Norm P = 0 ↔ P = 0

theorem **l2Norm_le_l1Norm** (P : MvPolynomial σ K) : l2Norm P ≤ l1Norm P

theorem **l1Norm_le_sqrt_card_mul_l2Norm** (P : MvPolynomial σ K) : l1Norm P ≤ √(#P.support) * l2Norm P

theorem **l1Norm_le_sqrt_choose_mul_l2Norm** [Fintype σ] {P : MvPolynomial σ K} {d : ℕ} (h : P.totalDegree ≤ d) : l1Norm P ≤ √((Fintype.card σ + d).choose d) * l2Norm P

theorem **supCoeff_nonneg** (P : MvPolynomial σ K) : 0 ≤ supCoeff P

theorem **norm_coeff_le_supCoeff** (P : MvPolynomial σ K) (α : σ →₀ ℕ) : ‖P.coeff α‖ ≤ supCoeff P

theorem **supCoeff_le_iff** {P : MvPolynomial σ K} {B : ℝ} (hB : 0 ≤ B) : supCoeff P ≤ B ↔ ∀ α, ‖P.coeff α‖ ≤ B

@[simp] theorem **supCoeff_zero** : supCoeff (0 : MvPolynomial σ K) = 0

@[simp] theorem **supCoeff_eq_zero** {P : MvPolynomial σ K} : supCoeff P = 0 ↔ P = 0

@[simp] theorem **supCoeff_monomial** (α : σ →₀ ℕ) (c : K) : supCoeff (monomial α c) = ‖c‖

theorem **multiFactorial_pos** (α : σ →₀ ℕ) : 0 < multiFactorial α

@[simp] theorem **multiFactorial_zero** : multiFactorial (0 : σ →₀ ℕ) = 1

@[simp] theorem **multiFactorial_single** (i : σ) (n : ℕ) : multiFactorial (Finsupp.single i n) = n.factorial

theorem **multiFactorial_add_single** (α : σ →₀ ℕ) (i : σ) : multiFactorial (α + Finsupp.single i 1) = (α i + 1) * multiFactorial α

theorem **fockNormSq_eq_sum_of_subset** {P : MvPolynomial σ K} {s : Finset (σ →₀ ℕ)} (h : P.support ⊆ s) : fockNormSq P = ∑ α ∈ s, (multiFactorial α : ℝ) * ‖P.coeff α‖ ^ 2

theorem **fockNormSq_nonneg** (P : MvPolynomial σ K) : 0 ≤ fockNormSq P

@[simp] theorem **fockNormSq_zero** : fockNormSq (0 : MvPolynomial σ K) = 0

@[simp] theorem **fockNormSq_monomial** (α : σ →₀ ℕ) (c : K) : fockNormSq (monomial α c) = multiFactorial α * ‖c‖ ^ 2

theorem **l2Norm_sq_le_fockNormSq** (P : MvPolynomial σ K) : l2Norm P ^ 2 ≤ fockNormSq P

@[simp] theorem **fockNormSq_eq_zero** {P : MvPolynomial σ K} : fockNormSq P = 0 ↔ P = 0

theorem **fockInner_eq_sum_of_subset_left** {P : MvPolynomial σ 𝕜} (Q : MvPolynomial σ 𝕜) {s : Finset (σ →₀ ℕ)} (h : P.support ⊆ s) : fockInner P Q = ∑ α ∈ s, (multiFactorial α : 𝕜) * (P.coeff α * starRingEnd 𝕜 (Q.coeff α))

theorem **fockInner_eq_sum_of_subset_right** (P : MvPolynomial σ 𝕜) {Q : MvPolynomial σ 𝕜} {s : Finset (σ →₀ ℕ)} (h : Q.support ⊆ s) : fockInner P Q = ∑ α ∈ s, (multiFactorial α : 𝕜) * (P.coeff α * starRingEnd 𝕜 (Q.coeff α))

theorem **fockInner_conj_symm** (P Q : MvPolynomial σ 𝕜) : starRingEnd 𝕜 (fockInner P Q) = fockInner Q P

theorem **fockInner_add_left** (P P' Q : MvPolynomial σ 𝕜) : fockInner (P + P') Q = fockInner P Q + fockInner P' Q

theorem **fockInner_smul_left** (c : 𝕜) (P Q : MvPolynomial σ 𝕜) : fockInner (c • P) Q = c * fockInner P Q

theorem **fockInner_add_right** (P Q Q' : MvPolynomial σ 𝕜) : fockInner P (Q + Q') = fockInner P Q + fockInner P Q'

theorem **fockInner_smul_right** (c : 𝕜) (P Q : MvPolynomial σ 𝕜) : fockInner P (c • Q) = starRingEnd 𝕜 c * fockInner P Q

@[simp] theorem **fockInner$_{sl}$_apply** (P Q : MvPolynomial σ 𝕜) : fockInner$_{sl}$ σ 𝕜 P Q = fockInner P Q

theorem **fockInner_self** (P : MvPolynomial σ 𝕜) : fockInner P P = (fockNormSq P : 𝕜)

theorem **fockInner_monomial_left** (α : σ →₀ ℕ) (c : 𝕜) (Q : MvPolynomial σ 𝕜) : fockInner (monomial α c) Q = multiFactorial α * (c * starRingEnd 𝕜 (Q.coeff α))

theorem **fockInner_monomial_right** (P : MvPolynomial σ 𝕜) (α : σ →₀ ℕ) (c : 𝕜) : fockInner P (monomial α c) = multiFactorial α * (P.coeff α * starRingEnd 𝕜 c)

theorem **fockInner_X_mul_left** (i : σ) (P Q : MvPolynomial σ 𝕜) : fockInner (X i * P) Q = fockInner P (pderiv i Q)

theorem **fockInner_pderiv_left** (i : σ) (P Q : MvPolynomial σ 𝕜) : fockInner (pderiv i P) Q = fockInner P (X i * Q)

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.TensorContentEqPolyContentScalarCoeffs

@[simp] theorem **coeffSup_zero** : coeffSup N (0 : ι →₀ M) = 0

theorem **coeffSup_le_iff** (h0 : N 0 = 0) {c : ι →₀ M} {γ : Γ₀} : coeffSup N c ≤ γ ↔ ∀ i, N (c i) ≤ γ

theorem **coeffSup_lt_iff** (h0 : N 0 = 0) {c : ι →₀ M} {γ : Γ₀} (hγ : γ ≠ 0) : coeffSup N c < γ ↔ ∀ i, N (c i) < γ

theorem **le_coeffSup** (h0 : N 0 = 0) (c : ι →₀ M) (i : ι) : N (c i) ≤ coeffSup N c

theorem **exists_mem_support_coeffSup_eq** {c : ι →₀ M} (hc : c ≠ 0) : ∃ i ∈ c.support, coeffSup N c = N (c i)

theorem **exists_coeffSup_eq** [Nonempty ι] (h0 : N 0 = 0) (c : ι →₀ M) : ∃ i, coeffSup N c = N (c i)

theorem **coeffSup_eq_zero_iff** (h0 : N 0 = 0) {c : ι →₀ M} : coeffSup N c = 0 ↔ ∀ i, N (c i) = 0

theorem **coeffSup_single** (h0 : N 0 = 0) (i : ι) (a : M) : coeffSup N (Finsupp.single i a) = N a

theorem **coeffSup_eq_sup_univ** [Fintype ι] (h0 : N 0 = 0) (c : ι →₀ M) : coeffSup N c = Finset.univ.sup fun i => N (c i)

theorem **coeffSup_eq_mul_of_forall** (h0 : N 0 = 0) {M' : Type*} [Zero M'] {N' : M' → Γ₀} (h0' : N' 0 = 0) {c : ι →₀ M} {c' : ι →₀ M'} {γ : Γ₀} (h : ∀ i, N' (c' i) = γ * N (c i)) : coeffSup N' c' = γ * coeffSup N c

theorem **coeffSup_coeff_mul_le** [Add G] (f g : AddMonoidAlgebra R G) : coeffSup v (f * g).coeff ≤ coeffSup v f.coeff * coeffSup v g.coeff

theorem **coeffSup_coeff_mul** [Add G] [UniqueSums G] (f g : AddMonoidAlgebra R G) : coeffSup v (f * g).coeff = coeffSup v f.coeff * coeffSup v g.coeff

theorem **gaussValAlg_apply** (f : AddMonoidAlgebra R G) : gaussValAlg v f = f.coeff.support.sup fun m => v (f.coeff m)

theorem **gaussValAlg_le_iff** {f : AddMonoidAlgebra R G} {γ : Γ₀} : gaussValAlg v f ≤ γ ↔ ∀ m, v (f.coeff m) ≤ γ

@[simp] theorem **gaussValAlg_single** (m : G) (a : R) : gaussValAlg v (AddMonoidAlgebra.single m a) = v a

theorem **exists_gaussValAlg_eq** (f : AddMonoidAlgebra R G) : ∃ m, gaussValAlg v f = v (f.coeff m)

theorem **gaussValAlg_eq_zero_iff** {f : AddMonoidAlgebra R G} : gaussValAlg v f = 0 ↔ ∀ m, v (f.coeff m) = 0

theorem **gaussVal_apply** (f : MvPolynomial σ R) : gaussVal v f = f.support.sup fun m => v (coeff m f)

theorem **gaussVal_mul** (f g : MvPolynomial σ R) : gaussVal v (f * g) = gaussVal v f * gaussVal v g

theorem **gaussVal_le_iff** {f : MvPolynomial σ R} {γ : Γ₀} : gaussVal v f ≤ γ ↔ ∀ m, v (coeff m f) ≤ γ

theorem **gaussVal_lt_iff** {f : MvPolynomial σ R} {γ : Γ₀} (hγ : γ ≠ 0) : gaussVal v f < γ ↔ ∀ m, v (coeff m f) < γ

theorem **le_gaussVal** (f : MvPolynomial σ R) (m : σ →₀ ℕ) : v (coeff m f) ≤ gaussVal v f

theorem **exists_gaussVal_eq** (f : MvPolynomial σ R) : ∃ m, gaussVal v f = v (coeff m f)

theorem **gaussVal_eq_zero_iff_forall** {f : MvPolynomial σ R} : gaussVal v f = 0 ↔ ∀ m, v (coeff m f) = 0

@[simp] theorem **gaussVal_monomial** (m : σ →₀ ℕ) (a : R) : gaussVal v (monomial m a) = v a

@[simp] theorem **gaussVal_C** (a : R) : gaussVal v (C a : MvPolynomial σ R) = v a

@[simp] theorem **gaussVal_X** (i : σ) : gaussVal v (X i : MvPolynomial σ R) = 1

@[simp] theorem **gaussVal_comap_C** : (gaussVal v).comap (C : R →+* MvPolynomial σ R) = v

theorem **gaussVal_smul** (a : R) (f : MvPolynomial σ R) : gaussVal v (a • f) = v a * gaussVal v f

theorem **gaussVal_supp** : (gaussVal v).supp = v.supp.map (C : R →+* MvPolynomial σ R)

theorem **gaussVal_supp_eq_bot_of_supp** (hv : v.supp = ⊥) : (gaussVal (σ

theorem **gaussVal_eq_zero_iff_of_supp** (hv : v.supp = ⊥) {f : MvPolynomial σ R} : gaussVal v f = 0 ↔ f = 0

theorem **gaussVal_sum_C_mul_X** (s : Finset σ) (a : σ → R) : gaussVal v (∑ i ∈ s, C (a i) * X i : MvPolynomial σ R) = s.sup fun i => v (a i)

theorem **gaussVal_map** (w : Valuation S Γ₀) (φ : R →+* S) (f : MvPolynomial σ R) : gaussVal w (map φ f) = gaussVal (w.comap φ) f

theorem **gaussVal_map_of_eq** (w : Valuation S Γ₀) (φ : R →+* S) (h : ∀ a, w (φ a) = v a) (f : MvPolynomial σ R) : gaussVal w (map φ f) = gaussVal v f

theorem **valuation_eval₂_le_gaussVal** (w : Valuation S Γ₀) (φ : R →+* S) (h : ∀ a, w (φ a) ≤ v a) (x : σ → S) (hx : ∀ i, w (x i) ≤ 1) (f : MvPolynomial σ R) : w (eval₂ φ x f) ≤ gaussVal v f

theorem **valuation_eval_le_gaussVal** (x : σ → R) (hx : ∀ i, v (x i) ≤ 1) (f : MvPolynomial σ R) : v (eval x f) ≤ gaussVal v f

theorem **valuation_aeval_le_gaussVal** [Algebra R S] (w : Valuation S Γ₀) (h : ∀ a, w (algebraMap R S a) ≤ v a) (x : σ → S) (hx : ∀ i, w (x i) ≤ 1) (f : MvPolynomial σ R) : w (aeval x f) ≤ gaussVal v f

theorem **gaussVal_aeval_le** (g : σ → MvPolynomial τ R) (hg : ∀ i, gaussVal v (g i) ≤ 1) (f : MvPolynomial σ R) : gaussVal v (aeval g f) ≤ gaussVal v f

theorem **gaussVal_algHom_le** (e : MvPolynomial σ R →ₐ[R] MvPolynomial τ R) (h : ∀ i, gaussVal v (e (X i)) ≤ 1) (f : MvPolynomial σ R) : gaussVal v (e f) ≤ gaussVal v f

theorem **gaussVal_algEquiv** (e : MvPolynomial σ R ≃ₐ[R] MvPolynomial τ R) (h : ∀ i, gaussVal v (e (X i)) ≤ 1) (h' : ∀ j, gaussVal v (e.symm (X j)) ≤ 1) (f : MvPolynomial σ R) : gaussVal v (e f) = gaussVal v f

theorem **gaussVal_rename_le** (e : σ → τ) (f : MvPolynomial σ R) : gaussVal v (rename e f) ≤ gaussVal v f

theorem **gaussVal_rename** {e : σ → τ} (he : Function.Injective e) (f : MvPolynomial σ R) : gaussVal v (rename e f) = gaussVal v f

@[simp] theorem **gaussVal_renameEquiv** (e : σ ≃ τ) (f : MvPolynomial σ R) : gaussVal v (renameEquiv R e f) = gaussVal v f

@[simp] theorem **gaussVal_sumAlgEquiv** (f : MvPolynomial (σ ⊕ τ) R) : gaussVal (gaussVal v) (sumAlgEquiv R σ τ f) = gaussVal v f

@[simp] theorem **gaussVal_sumAlgEquiv_symm** (g : MvPolynomial σ (MvPolynomial τ R)) : gaussVal v ((sumAlgEquiv R σ τ).symm g) = gaussVal (gaussVal v) g

theorem **valuation_supp_eq_bot** : v.supp = ⊥

theorem **gaussVal_supp_eq_bot** : (gaussVal (σ

@[simp] theorem **gaussVal_eq_zero_iff** {f : MvPolynomial σ K} : gaussVal v f = 0 ↔ f = 0

theorem **exists_eq_C_mul_gaussVal_eq_one** {f : MvPolynomial σ K} (hf : f ≠ 0) : ∃ (c : K) (g : MvPolynomial σ K), c ≠ 0 ∧ v c = gaussVal v f ∧ gaussVal v g = 1 ∧ f = C c * g

theorem **gaussVal_le_one_iff** {O : Type*} [CommRing O] [Algebra O K] (hO : v.Integers O) {f : MvPolynomial σ K} : gaussVal v f ≤ 1 ↔ ∃ g : MvPolynomial σ O, map (algebraMap O K) g = f

theorem **gaussVal_lt_one_iff_map_residue** {O : Type*} [CommRing O] [IsLocalRing O] [Algebra O K] (hO : v.Integers O) (g : MvPolynomial σ O) : gaussVal v (map (algebraMap O K) g) < 1 ↔ map (IsLocalRing.residue O) g = 0

theorem **exists_valuation_eval_eq_gaussVal** {O : Type*} [CommRing O] [IsLocalRing O] [Algebra O K] (hO : v.Integers O) [Infinite (IsLocalRing.ResidueField O)] (f d : MvPolynomial σ K) (hd : gaussVal v d = 1) : ∃ x : σ → K, (∀ i, v (x i) ≤ 1) ∧ v (eval x d) = 1 ∧ v (eval x f) = gaussVal v f

@[simp] theorem **gaussValFrac_algebraMap** (f : MvPolynomial σ K) : gaussValFrac v σ F (algebraMap _ F f) = gaussVal v f

theorem **gaussValFrac_div** (f g : MvPolynomial σ K) : gaussValFrac v σ F (algebraMap _ F f / algebraMap _ F g) = gaussVal v f / gaussVal v g

@[simp] theorem **gaussValFrac_comap** : (gaussValFrac v σ F).comap (algebraMap (MvPolynomial σ K) F) = gaussVal v

@[simp] theorem **gaussValFrac_algebraMap_const** [Algebra K F] [IsScalarTower K (MvPolynomial σ K) F] (a : K) : gaussValFrac v σ F (algebraMap K F a) = v a

theorem **eq_gaussValFrac** {w : Valuation F Γ₀} (h : ∀ f : MvPolynomial σ K, w (algebraMap _ F f) = gaussVal v f) : w = gaussValFrac v σ F

theorem **exists_gaussValFrac_eq** (x : F) : ∃ c : K, gaussValFrac v σ F x = v c

theorem **gaussValFrac_le_one_iff** {x : F} : gaussValFrac v σ F x ≤ 1 ↔ ∃ f g : MvPolynomial σ K, gaussVal v f ≤ 1 ∧ gaussVal v g = 1 ∧ x = algebraMap _ F f / algebraMap _ F g

theorem **gaussVal_gaussValFrac_sumAlgEquiv** (f : MvPolynomial (τ ⊕ σ) K) : gaussVal (gaussValFrac v σ F) (map (algebraMap _ F) (sumAlgEquiv K τ σ f)) = gaussVal v f

theorem **le_gaussAddVal_iff** {f : MvPolynomial σ R} {γ : Γ} : γ ≤ gaussAddVal v f ↔ ∀ m, γ ≤ v (coeff m f)

theorem **gaussAddVal_apply** (f : MvPolynomial σ R) : gaussAddVal v f = f.support.inf fun m => v (coeff m f)

theorem **gaussAddVal_le** (f : MvPolynomial σ R) (m : σ →₀ ℕ) : gaussAddVal v f ≤ v (coeff m f)

theorem **exists_gaussAddVal_eq** (f : MvPolynomial σ R) : ∃ m, gaussAddVal v f = v (coeff m f)

theorem **gaussAddVal_mul** (f g : MvPolynomial σ R) : gaussAddVal v (f * g) = gaussAddVal v f + gaussAddVal v g

@[simp] theorem **gaussAddVal_C** (a : R) : gaussAddVal v (C a : MvPolynomial σ R) = v a

@[simp] theorem **gaussAddVal_X** (i : σ) : gaussAddVal v (X i : MvPolynomial σ R) = 0

theorem **gaussAddVal_rename** {e : σ → τ} (he : Function.Injective e) (f : MvPolynomial σ R) : gaussAddVal v (rename e f) = gaussAddVal v f

@[simp] theorem **gaussAddVal_sumAlgEquiv** (f : MvPolynomial (σ ⊕ τ) R) : gaussAddVal (gaussAddVal v) (sumAlgEquiv R σ τ f) = gaussAddVal v f

@[simp] theorem **coe_absValuation** [Nontrivial R] (a : AbsoluteValue R ℝ) (ha : IsNonarchimedean a) (x : R) : (absValuation a ha x : ℝ) = a x

theorem **absValuation_supp** [Nontrivial R] (a : AbsoluteValue R ℝ) (ha : IsNonarchimedean a) : (absValuation a ha).supp = ⊥

@[simp] theorem **valuationAbs_apply** (v : Valuation R ℝ≥0) (hv : v.supp = ⊥) (x : R) : valuationAbs v hv x = v x

theorem **isNonarchimedean_valuationAbs** (v : Valuation R ℝ≥0) (hv : v.supp = ⊥) : IsNonarchimedean (valuationAbs v hv)

theorem **gaussAbs_apply** (f : MvPolynomial σ R) : gaussAbs a ha f = gaussVal (absValuation a ha) f

theorem **isNonarchimedean_gaussAbs** : IsNonarchimedean (gaussAbs (σ

theorem **le_gaussAbs** (f : MvPolynomial σ R) (m : σ →₀ ℕ) : a (coeff m f) ≤ gaussAbs a ha f

theorem **exists_gaussAbs_eq** (f : MvPolynomial σ R) : ∃ m, gaussAbs a ha f = a (coeff m f)

theorem **gaussAbs_le_iff** {f : MvPolynomial σ R} {r : ℝ} : gaussAbs a ha f ≤ r ↔ ∀ m, a (coeff m f) ≤ r

@[simp] theorem **gaussAbs_C** (x : R) : gaussAbs a ha (C x : MvPolynomial σ R) = a x

@[simp] theorem **gaussAbs_X** (i : σ) : gaussAbs a ha (X i : MvPolynomial σ R) = 1

theorem **gaussAbs_rename** {e : σ → τ} (he : Function.Injective e) (f : MvPolynomial σ R) : gaussAbs a ha (rename e f) = gaussAbs a ha f

theorem **gaussAbs_sumAlgEquiv** (f : MvPolynomial (σ ⊕ τ) R) : gaussAbs (gaussAbs a ha) (isNonarchimedean_gaussAbs a ha) (sumAlgEquiv R σ τ f) = gaussAbs a ha f

theorem **gaussAbs_eq_gaussNorm** (f : MvPolynomial σ R) : gaussAbs a ha f = MvPowerSeries.gaussNorm a (fun _ => 1) (f : MvPowerSeries σ R)

@[simp] theorem **gaussAbsFrac_algebraMap** (f : MvPolynomial σ K) : gaussAbsFrac a ha σ F (algebraMap _ F f) = gaussAbs a ha f

theorem **isNonarchimedean_gaussAbsFrac** : IsNonarchimedean (gaussAbsFrac a ha σ F)

theorem **basisVal_le_iff** {x : M} {γ : Γ₀} : basisVal v b x ≤ γ ↔ ∀ i, v (b.repr x i) ≤ γ

theorem **le_basisVal** (x : M) (i : ι) : v (b.repr x i) ≤ basisVal v b x

theorem **basisVal_eq_sup_univ** [Fintype ι] (x : M) : basisVal v b x = Finset.univ.sup fun i => v (b.repr x i)

theorem **exists_basisVal_eq** {x : M} (hx : x ≠ 0) : ∃ i, basisVal v b x = v (b.repr x i)

@[simp] theorem **basisVal_zero** : basisVal v b 0 = 0

@[simp] theorem **basisVal_basis** (i : ι) : basisVal v b (b i) = 1

theorem **basisVal_smul** (a : R) (x : M) : basisVal v b (a • x) = v a * basisVal v b x

@[simp] theorem **basisVal_neg** (x : M) : basisVal v b (-x) = basisVal v b x

theorem **basisVal_add_le** (x y : M) : basisVal v b (x + y) ≤ max (basisVal v b x) (basisVal v b y)

theorem **basisVal_sum_le** {α : Type*} {s : Finset α} {x : α → M} {γ : Γ₀} (h : ∀ a ∈ s, basisVal v b (x a) ≤ γ) : basisVal v b (∑ a ∈ s, x a) ≤ γ

theorem **basisVal_eq_zero_iff** (hv : v.supp = ⊥) {x : M} : basisVal v b x = 0 ↔ x = 0

theorem **basisVal_map_le** (c : Basis κ R N) (φ : M →ₗ[R] N) (h : ∀ i, basisVal v c (φ (b i)) ≤ 1) (x : M) : basisVal v c (φ x) ≤ basisVal v b x

theorem **valuation_linearMap_le_basisVal** (φ : M →ₗ[R] R) (h : ∀ i, v (φ (b i)) ≤ 1) (x : M) : v (φ x) ≤ basisVal v b x

theorem **isGreatest_basisVal** (x : M) : IsGreatest {γ | ∃ φ : M →ₗ[R] R, (∀ y, basisVal v b y ≤ 1 → v (φ y) ≤ 1) ∧ v (φ x) = γ} (basisVal v b x)

theorem **basisVal_eq_of_le_one** (c : Basis κ R M) (h : ∀ i, basisVal v c (b i) ≤ 1) (h' : ∀ j, basisVal v b (c j) ≤ 1) (x : M) : basisVal v b x = basisVal v c x

theorem **basisVal_eq_of_forall_le_one** (hv : ∀ a, v a ≤ 1) (c : Basis κ R M) (x : M) : basisVal v b x = basisVal v c x

theorem **basisVal_basisMonomials** (f : MvPolynomial σ R) : basisVal v (basisMonomials σ R) f = gaussVal v f

theorem **polyContent_le_iff** {c : (σ →₀ ℕ) →₀ M} {γ : Γ₀} : polyContent v b c ≤ γ ↔ ∀ m, basisVal v b (c m) ≤ γ

theorem **polyContent_le_iff_coord** {c : (σ →₀ ℕ) →₀ M} {γ : Γ₀} : polyContent v b c ≤ γ ↔ ∀ m i, v (b.repr (c m) i) ≤ γ

theorem **le_polyContent** (c : (σ →₀ ℕ) →₀ M) (m : σ →₀ ℕ) : basisVal v b (c m) ≤ polyContent v b c

theorem **exists_polyContent_eq** (c : (σ →₀ ℕ) →₀ M) : ∃ m, polyContent v b c = basisVal v b (c m)

theorem **polyContent_smul** (a : R) (c : (σ →₀ ℕ) →₀ M) : polyContent v b (a • c) = v a * polyContent v b c

theorem **polyContent_eq_zero_iff** (hv : v.supp = ⊥) {c : (σ →₀ ℕ) →₀ M} : polyContent v b c = 0 ↔ c = 0

theorem **polyContent_eq_of_le_one** (b' : Basis κ R M) (h : ∀ i, basisVal v b' (b i) ≤ 1) (h' : ∀ j, basisVal v b (b' j) ≤ 1) (c : (σ →₀ ℕ) →₀ M) : polyContent v b c = polyContent v b' c

theorem **basisVal_le_one_iff** (hO : v.Integers O) {x : M} : basisVal v b x ≤ 1 ↔ x ∈ Submodule.span O (Set.range b)

theorem **basisVal_eq_of_span_eq** (hO : v.Integers O) (c : Basis κ R M) (h : Submodule.span O (Set.range b) = Submodule.span O (Set.range c)) (x : M) : basisVal v b x = basisVal v c x

theorem **basisVal_baseChange_eq** (hv : ∀ a : O, v (algebraMap O R a) ≤ 1) (c : Basis κ O M) (x : R ⊗[O] M) : basisVal v (b.baseChange R) x = basisVal v (c.baseChange R) x

theorem **basisVal_baseChange_one_tmul** (y : M) : basisVal v (b.baseChange R) (1 ⊗ₜ y) = basisVal (v.comap (algebraMap O R)) b y

theorem **basisVal_baseChange_rTensor** (w : Valuation S Γ₀) (φ : R →ₐ[O] S) (h : ∀ a, w (φ a) = v a) (x : R ⊗[O] M) : basisVal w (b.baseChange S) (φ.toLinearMap.rTensor M x) = basisVal v (b.baseChange R) x

@[simp] theorem **tensorCoeffs_tmul** (f : MvPolynomial σ R) (y : M) (m : σ →₀ ℕ) : tensorCoeffs (f ⊗ₜ[O] y) m = coeff m f ⊗ₜ[O] y

theorem **repr_tensorCoeffs** (x : MvPolynomial σ R ⊗[O] M) (m : σ →₀ ℕ) (i : ι) : (b.baseChange R).repr (tensorCoeffs x m) i = coeff m ((b.baseChange (MvPolynomial σ R)).repr x i)

theorem **tensorContent_eq_polyContent** (x : MvPolynomial σ R ⊗[O] M) : tensorContent v b x = polyContent v (b.baseChange R) (tensorCoeffs x)

theorem **tensorContent_le_iff** {x : MvPolynomial σ R ⊗[O] M} {γ : Γ₀} : tensorContent v b x ≤ γ ↔ ∀ m, basisVal v (b.baseChange R) (tensorCoeffs x m) ≤ γ

theorem **tensorContent_smul** (f : MvPolynomial σ R) (x : MvPolynomial σ R ⊗[O] M) : tensorContent v b (f • x) = gaussVal v f * tensorContent v b x

theorem **tensorContent_add_le** (x y : MvPolynomial σ R ⊗[O] M) : tensorContent v b (x + y) ≤ max (tensorContent v b x) (tensorContent v b y)

theorem **tensorContent_tmul** (f : MvPolynomial σ R) (y : M) : tensorContent v b (f ⊗ₜ y) = gaussVal v f * basisVal (v.comap (algebraMap O R)) b y

theorem **tensorContent_eq_zero_iff** (hv : v.supp = ⊥) {x : MvPolynomial σ R ⊗[O] M} : tensorContent v b x = 0 ↔ x = 0

theorem **tensorContent_eq_of_basis** (hv : ∀ a : O, v (algebraMap O R a) ≤ 1) (c : Basis κ O M) (x : MvPolynomial σ R ⊗[O] M) : tensorContent v b x = tensorContent v c x

@[simp] theorem **scalarCoeffs_tmul** {N : Type*} [AddCommGroup N] [Module R N] (f : MvPolynomial σ R) (y : N) (m : σ →₀ ℕ) : scalarCoeffs (f ⊗ₜ[R] y) m = coeff m f • y

theorem **tensorContent_eq_polyContent_scalarCoeffs** {N : Type*} [AddCommGroup N] [Module R N] (c : Basis ι R N) (x : MvPolynomial σ R ⊗[R] N) : tensorContent v c x = polyContent v c (scalarCoeffs x)

theorem **le_basisAddVal_iff** {x : M} {γ : Γ} : γ ≤ basisAddVal v b x ↔ ∀ i, γ ≤ v (b.repr x i)

theorem **basisAddVal_eq_inf_univ** [Fintype ι] (x : M) : basisAddVal v b x = Finset.univ.inf fun i => v (b.repr x i)

theorem **basisAddVal_smul** (a : R) (x : M) : basisAddVal v b (a • x) = v a + basisAddVal v b x

theorem **basisAddVal_eq_of_forall_nonneg** (hv : ∀ a, 0 ≤ v a) (c : Basis κ R M) (x : M) : basisAddVal v b x = basisAddVal v c x

theorem **le_polyAddContent_iff** {c : (σ →₀ ℕ) →₀ M} {γ : Γ} : γ ≤ polyAddContent v b c ↔ ∀ m, γ ≤ basisAddVal v b (c m)

theorem **tensorAddContent_eq_polyAddContent** (x : MvPolynomial σ R ⊗[O] M) : tensorAddContent v b x = polyAddContent v (b.baseChange R) (tensorCoeffs x)

theorem **tensorAddContent_smul** (f : MvPolynomial σ R) (x : MvPolynomial σ R ⊗[O] M) : tensorAddContent v b (f • x) = gaussAddVal v f + tensorAddContent v b x

theorem **tensorAddContent_eq_of_basis** (hv : ∀ a : O, 0 ≤ v (algebraMap O R a)) (c : Basis κ O M) (x : MvPolynomial σ R ⊗[O] M) : tensorAddContent v b x = tensorAddContent v c x

theorem **basisVal_gaussValFrac** {O K Γ₀ ι σ F M : Type*} [CommRing O] [Field K] [Algebra O K] [LinearOrderedCommGroupWithZero Γ₀] (v : Valuation K Γ₀) [Field F] [Algebra (MvPolynomial σ K) F] [IsFractionRing (MvPolynomial σ K) F] [Algebra O F] [IsScalarTower O (MvPolynomial σ K) F] [AddCommGroup M] [Module O M] (b : Basis ι O M) (x : MvPolynomial σ K ⊗[O] M) : basisVal (gaussValFrac v σ F) (b.baseChange F) ((IsScalarTower.toAlgHom O (MvPolynomial σ K) F).toLinearMap.rTensor M x) = tensorContent v b x

@[simp] theorem **gaussAddValFrac_algebraMap** {K Γ σ F : Type*} [Field K] [LinearOrderedAddCommGroupWithTop Γ] (v : AddValuation K Γ) [Field F] [Algebra (MvPolynomial σ K) F] [IsFractionRing (MvPolynomial σ K) F] (f : MvPolynomial σ K) : gaussAddValFrac v σ F (algebraMap _ F f) = gaussAddVal v f

theorem **finite_gaussVal_ne_one** {O K σ : Type*} [CommRing O] [IsDedekindDomain O] [Field K] [Algebra O K] [IsFractionRing O K] {f : MvPolynomial σ K} (hf : f ≠ 0) : {p : HeightOneSpectrum O | gaussVal (p.valuation K) f ≠ 1}.Finite

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.TwistingSheafOneChartEqPreimageStandardChart

theorem **X_mem** (i : Fin (N + 1)) : (X i : MvPolynomial (Fin (N + 1)) R) ∈ grading R N 1

theorem **X_pow_mem** (i : Fin (N + 1)) (n : ℕ) : (X i ^ n : MvPolynomial (Fin (N + 1)) R) ∈ grading R N n

theorem **isAffineOpen_chart** (i : Fin (N + 1)) : IsAffineOpen (chart R N i)

theorem **mem_span_X_of_isHomogeneous** {n : ℕ} (hn : 0 < n) {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : f ∈ Ideal.span (Set.range (X : Fin (N + 1) → MvPolynomial _ R))

theorem **irrelevant_le_span_X** : (HomogeneousIdeal.irrelevant (grading R N)).toIdeal ≤ Ideal.span (Set.range (X : Fin (N + 1) → MvPolynomial _ R))

theorem **iSup_chart** : ⨆ i, chart R N i = ⊤

theorem **exists_mem_chart** (x : PSpace R N) : ∃ i, x ∈ chart R N i

theorem **le_basicOpen_X_pow** {U : (PSpace R N).Opens} {i : Fin (N + 1)} (hU : U ≤ chart R N i) (n : ℕ) : U ≤ Proj.basicOpen (grading R N) (X i ^ n)

@[simp] theorem **val_ratioUnit** : (ratioUnit U i j hi hj : Γ(PSpace R N, U)) = ratioOn U i hi j

theorem **inv_ratioUnit** : (ratioUnit U i j hi hj)⁻¹ = ratioUnit U j i hj hi

theorem **resG_ratioOn** (h : V ≤ U) : resG _ h (ratioOn U i hi j) = ratioOn V i (h.trans hi) j

theorem **res_ratioUnit** (h : V ≤ U) : res h (ratioUnit U i j hi hj) = ratioUnit V i j (h.trans hi) (h.trans hj)

theorem **resG_formOn** (h : V ≤ U) (hf : f.IsHomogeneous n) : resG _ h (formOn U i hi n f hf) = formOn V i (h.trans hi) n f hf

@[simp] theorem **ratioOn_self** : ratioOn U i hi i = 1

theorem **ratioOn_mul_ratioOn** : ratioOn U i hi j * ratioOn U j hj k = ratioOn U i hi k

theorem **ratioUnit_self** {hi' : U ≤ chart R N i} : ratioUnit U i i hi hi' = 1

theorem **ratioUnit_mul_ratioUnit** : ratioUnit U i j hi hj * ratioUnit U j k hj hk = ratioUnit U i k hi hk

theorem **basicOpen_ratioOn** : (PSpace R N).basicOpen (ratioOn U i hi j) = U ⊓ chart R N j

theorem **basicOpen_formOn** (hf : f.IsHomogeneous n) : (PSpace R N).basicOpen (formOn U i hi n f hf) = U ⊓ Proj.basicOpen (grading R N) f

theorem **formOn_X** : formOn U i hi 1 (X j) (isHomogeneous_X R j) = ratioOn U i hi j

theorem **formOn_add** (hf : f.IsHomogeneous n) (hg : g.IsHomogeneous n) : formOn U i hi n (f + g) (hf.add hg) = formOn U i hi n f hf + formOn U i hi n g hg

theorem **formOn_zero** : formOn U i hi n 0 (isHomogeneous_zero _ R n) = 0

theorem **formOn_mul** (hf : f.IsHomogeneous n) (hg : g.IsHomogeneous m) : formOn U i hi (n + m) (f * g) (hf.mul hg) = formOn U i hi n f hf * formOn U i hi m g hg

theorem **formOn_eq_ratioOn_pow_mul** (hj : U ≤ chart R N j) (hf : f.IsHomogeneous n) : formOn U i hi n f hf = ratioOn U i hi j ^ n * formOn U j hj n f hf

@[simp] theorem **twistingSheafOne_chart_eq_preimage_standardChart** (i : Fin (N + 1)) : (O1 R N).U i = Proj.basicOpen (grading R N) (X i)

theorem **O1_g_eq_ratioUnit_chart_inf** (i j : Fin (N + 1)) : (O1 R N).g i j = ratioUnit (chart R N i ⊓ chart R N j) i j inf_le_left inf_le_right

theorem **O1_isLB** : (O1 R N).IsLB where

@[simp] theorem **coord_f** (k i : Fin (N + 1)) : (coord R N k).f i = ratio R N i k

theorem **nonzeroLocus_coord** (k : Fin (N + 1)) : nonzeroLocus (coord R N k) = (O1 R N).U k

theorem **iSup_nonzeroLocus_coord** : ⨆ k, nonzeroLocus (coord R N k) = ⊤

theorem **linComb_punit_g** {X Y : Scheme.{0}} (L : LB X) (φ : Y ⟶ X) (n : ℤ) (i j : PUnit → L.ι) : (LB.LinComb L (K

theorem **On_isLB** (n : ℤ) : (On R N n).IsLB

theorem **On_U** (n : ℤ) (j : PUnit → Fin (N + 1)) : (On R N n).U j = chart R N (j PUnit.unit)

theorem **On_g** (n : ℤ) (i j : PUnit → Fin (N + 1)) : (On R N n).g i j = ratioUnit ((On R N n).U i ⊓ (On R N n).U j) (i PUnit.unit) (j PUnit.unit) (inf_le_left.trans (On_U R N n i).le) (inf_le_right.trans (On_U R N n j).le) ^ n

@[simp] theorem **formSec_f** (n : ℕ) (f : MvPolynomial (Fin (N + 1)) R) (hf : f.IsHomogeneous n) (j : PUnit → Fin (N + 1)) : (formSec n f hf).f j = formOn _ (j PUnit.unit) (On_U R N n j).le n f hf

theorem **nonzeroLocus_formSec** (n : ℕ) (f : MvPolynomial (Fin (N + 1)) R) (hf : f.IsHomogeneous n) : nonzeroLocus (formSec n f hf) = Proj.basicOpen (grading R N) f

theorem **algebraMap_sections_eq** (U : (PSpace R N).Opens) : algebraMap R Γ(PSpace R N, U) = ((Scheme.ΓSpecIso (CommRingCat.of R)).inv ≫ (PSpace.str R N).appLE ⊤ U le_top).hom

theorem **resG_algebraMap** {U V : (PSpace R N).Opens} (h : V ≤ U) (r : R) : resG _ h (algebraMap R Γ(PSpace R N, U) r) = algebraMap R Γ(PSpace R N, V) r

theorem **algebraMap_chart** (i : Fin (N + 1)) (r : R) : algebraMap R Γ(PSpace R N, chart R N i) r = (Proj.awayToSection (grading R N) (X i)).hom (constAway R N i r)

theorem **formOn_C** {U : (PSpace R N).Opens} {i : Fin (N + 1)} {hi : U ≤ chart R N i} (r : R) : formOn U i hi 0 (C r) (isHomogeneous_C _ r) = algebraMap R Γ(PSpace R N, U) r

@[simp] theorem **dehom_X_self** (i : Fin (N + 1)) : dehom R N i (X i) = 1

@[simp] theorem **dehom_X_of_ne** (i : Fin (N + 1)) (j : {j // j ≠ i}) : dehom R N i (X j.1) = X j

theorem **awayToPoly_mk** (i : Fin (N + 1)) (n : ℕ) (f : MvPolynomial (Fin (N + 1)) R) (hf : f ∈ grading R N (n • 1)) : awayToPoly R N i (HomogeneousLocalization.Away.mk (grading R N) (X_mem R N i) n f hf) = dehom R N i f

theorem **polyToAway_dehom** (i : Fin (N + 1)) (n : ℕ) (f : MvPolynomial (Fin (N + 1)) R) (hf : f ∈ grading R N (n • 1)) : polyToAway R N i (dehom R N i f) = HomogeneousLocalization.Away.mk (grading R N) (X_mem R N i) n f hf

theorem **chartAlgEquiv_apply** (i : Fin (N + 1)) (p : MvPolynomial {j // j ≠ i} R) : chartAlgEquiv R N i p = (Proj.awayToSection (grading R N) (X i)).hom (polyToAway R N i p)

@[simp] theorem **chartAlgEquiv_X** (i : Fin (N + 1)) (j : {j // j ≠ i}) : chartAlgEquiv R N i (X j) = ratio R N i j.1

theorem **chartAlgEquiv_dehom** (i : Fin (N + 1)) {n : ℕ} {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : chartAlgEquiv R N i (dehom R N i f) = formOn (chart R N i) i le_rfl n f hf

theorem **formOn_eq_aeval** {U : (PSpace R N).Opens} {i : Fin (N + 1)} {hi : U ≤ chart R N i} {n : ℕ} {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : formOn U i hi n f hf = aeval (ratioOn U i hi) f

theorem **formOn_chart_inj** (i : Fin (N + 1)) {n : ℕ} {f g : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) (hg : g.IsHomogeneous n) : formOn (chart R N i) i le_rfl n f hf = formOn (chart R N i) i le_rfl n g hg ↔ f = g

theorem **formSec_inj** {n : ℕ} {f g : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) (hg : g.IsHomogeneous n) : formSec n f hf = formSec n g hg ↔ f = g

theorem **aeval_ratio_surjective** (i : Fin (N + 1)) : Function.Surjective (aeval (R

theorem **opensRange_chartι** (i : Fin (N + 1)) : (chartι R N i).opensRange = chart R N i

theorem **chartι_str** (i : Fin (N + 1)) : chartι R N i ≫ PSpace.str R N = spMap (C : R →+* MvPolynomial {j // j ≠ i} R)

@[simp] theorem **gradedMap_apply** (f : MvPolynomial (Fin (N + 1)) R) : gradedMap N φ f = MvPolynomial.map φ f

theorem **irrelevant_le_map** : HomogeneousIdeal.irrelevant (grading R' N) ≤ (HomogeneousIdeal.irrelevant (grading R N)).map (gradedMap N φ)

theorem **baseChange_preimage_basicOpen** (f : MvPolynomial (Fin (N + 1)) R) : baseChange N φ ⁻¹ᵁ Proj.basicOpen (grading R N) f = Proj.basicOpen (grading R' N) (MvPolynomial.map φ f)

theorem **baseChange_preimage_chart** (i : Fin (N + 1)) : baseChange N φ ⁻¹ᵁ chart R N i = chart R' N i

theorem **baseChange_preimage_le** {U : (PSpace R N).Opens} {i : Fin (N + 1)} (hi : U ≤ chart R N i) : baseChange N φ ⁻¹ᵁ U ≤ chart R' N i

theorem **baseChange_app_ratioOn** {U : (PSpace R N).Opens} {i : Fin (N + 1)} (hi : U ≤ chart R N i) (j : Fin (N + 1)) : ((baseChange N φ).app U).hom (ratioOn U i hi j) = ratioOn (baseChange N φ ⁻¹ᵁ U) i (baseChange_preimage_le φ hi) j

theorem **baseChange_app_formOn** {U : (PSpace R N).Opens} {i : Fin (N + 1)} (hi : U ≤ chart R N i) {n : ℕ} {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : ((baseChange N φ).app U).hom (formOn U i hi n f hf) = formOn (baseChange N φ ⁻¹ᵁ U) i (baseChange_preimage_le φ hi) n (MvPolynomial.map φ f) (hf.map φ)

theorem **pullU_baseChange_ratioUnit** {U : (PSpace R N).Opens} {i j : Fin (N + 1)} (hi : U ≤ chart R N i) (hj : U ≤ chart R N j) : pullU (baseChange N φ) U (ratioUnit U i j hi hj) = ratioUnit (baseChange N φ ⁻¹ᵁ U) i j (baseChange_preimage_le φ hi) (baseChange_preimage_le

φ hj)

theorem **chartι_baseChange** (i : Fin (N + 1)) : chartι R' N i ≫ baseChange N φ = spMap (MvPolynomial.map φ : MvPolynomial {j // j ≠ i} R →+* _) ≫ chartι R N i

theorem **isPullback_baseChange** : IsPullback (baseChange N φ) (PSpace.str R' N) (PSpace.str R N) (spMap φ)

theorem **baseChange_str** : baseChange N φ ≫ PSpace.str R N = PSpace.str R' N ≫ spMap φ

theorem **pull_O1_chart** (j : PUnit → Fin (N + 1)) : ((O1 R N).pull (baseChange N φ)).U j = chart R' N (j PUnit.unit)

theorem **pullback_O1_transition_eq_blockUnit** (i j : PUnit → Fin (N + 1)) : ((O1 R N).pull (baseChange N φ)).g i j = ratioUnit (((O1 R N).pull (baseChange N φ)).U i ⊓ ((O1 R N).pull (baseChange N φ)).U j) (i PUnit.unit) (j PUnit.unit) (inf_le_left.trans (pull_O1_chart N φ i).le) (inf_le_right.trans (pull_O1_chart N φ j).le)

theorem **baseChangeIso_isGen** : (baseChangeIso N φ).IsGen

theorem **lbIso_pull_O1** : LBIso ((O1 R N).pull (baseChange N φ)) (O1 R' N)

theorem **baseChangeIso_coord** (k i : Fin (N + 1)) (j : PUnit → Fin (N + 1)) : resG _ (inf_le_left : (O1 R' N).U i ⊓ ((O1 R N).pull (baseChange N φ)).U j ≤ _) ((coord R' N k).f i) = ((baseChangeIso N φ).s (i, j)).val * resG _ (inf_le_right.trans (LB.wchart_le (O1 R N) (fun _ => baseChange N φ) j PUnit.unit)) (((baseChange N φ).app ((O1 R N).U (j PUnit.unit))).hom ((coord R N k).f (j PUnit.unit)))

theorem **baseChange_id** : baseChange N (RingHom.id R) = 𝟙 (PSpace R N)

theorem **baseChange_comp** {R'' : Type} [CommRing R''] (ψ : R' →+* R'') : baseChange N (ψ.comp φ) = baseChange N ψ ≫ baseChange N φ

Development.AbelianVarietyTheory.Definitions.ThetaMetrics.WeightedHomogeneousComponentMemElimIdeal

theorem **genericForm_eq** (i : β) : genericForm K σ i = ∑ j, C (X j) * X (i, j)

theorem **C_mem_incidenceIdeal** {p : Ideal (MvPolynomial σ K)} {g : MvPolynomial σ K} (hg : g ∈ p) : C g ∈ incidenceIdeal β p

theorem **genericForm_mem_incidenceIdeal** (p : Ideal (MvPolynomial σ K)) (i : β) : genericForm K σ i ∈ incidenceIdeal β p

theorem **incidenceIdeal_le** {p : Ideal (MvPolynomial σ K)} {J : Ideal (MvPolynomial (β × σ) (MvPolynomial σ K))} : incidenceIdeal β p ≤ J ↔ (∀ g ∈ p, C g ∈ J) ∧ ∀ i, genericForm K σ i ∈ J

theorem **mem_elimIdeal_iff_forall** {p : Ideal (MvPolynomial σ K)} {f : MvPolynomial (β × σ) K} : f ∈ elimIdeal β p ↔ ∀ j, ∃ m : ℕ, map C f * C (X j ^ m) ∈ incidenceIdeal β p

theorem **aeval_eq_zero_of_mem_elimIdeal** {G : Type*} [CommRing G] [IsDomain G] [Algebra K G] {p : Ideal (MvPolynomial σ K)} {ξ : σ → G} (hξ : ∀ g ∈ p, aeval ξ g = 0) (hne : ξ ≠ 0) {t : β × σ → G} (ht : ∀ i, ∑ j, t (i, j) * ξ j = 0) {f : MvPolynomial (β × σ) K} (hf : f ∈ elimIdeal β p) : aeval t f = 0

theorem **rename_mem_elimIdeal** (e : β → β') {p : Ideal (MvPolynomial σ K)} {f : MvPolynomial (β × σ) K} (hf : f ∈ elimIdeal β p) : rename (Prod.map e id) f ∈ elimIdeal β' p

theorem **map_rename_elimIdeal** (e : β ≃ β') (p : Ideal (MvPolynomial σ K)) : (elimIdeal β p).map (rename (Prod.map e id) : _ →ₐ[K] _) = elimIdeal β' p

theorem **exists_pow_mul_sub_rename_mem** {R : Type*} [CommRing R] (u : σ → R) (j : σ) (h : MvPolynomial (β × σ) R) : ∃ (m : ℕ) (g : MvPolynomial {v : β × σ // v ∉ elimIndices Set.univ j} R), h * C (u j ^ m) − rename Subtype.val g ∈ Ideal.span (Set.range (linForm u : β → _))

theorem **map_linForm** {R S : Type*} [CommRing R] [CommRing S] (φ : R →+* S) (u : σ → R) (i : β) : map φ (linForm u i) = linForm (fun j => φ (u j)) i

theorem **map_mem_span_linForm** {R S : Type*} [CommRing R] [CommRing S] (φ : R →+* S) {u : σ → R} {h : MvPolynomial (β × σ) R} (hh : h ∈ Ideal.span (Set.range (linForm u : β → _))) : map φ h ∈ Ideal.span (Set.range (linForm (fun j => φ (u j)) : β → _))

theorem **mem_elimIdeal_iff_of_generic** {F : Type*} [Field F] [Algebra K F] {p : Ideal (MvPolynomial σ K)} {ξ : σ → F} (hξ : ∀ g, aeval ξ g = 0 ↔ g ∈ p) (hne : ξ ≠ 0) {f : MvPolynomial (β × σ) K} : f ∈ elimIdeal β p ↔ map (algebraMap K F) f ∈ Ideal.span (Set.range (linForm ξ : β → _))

theorem **aeval_genericPoint** (p : Ideal (MvPolynomial σ K)) (g : MvPolynomial σ K) : aeval (genericPoint p) g = algebraMap (MvPolynomial σ K ⧸ p) _ (Ideal.Quotient.mk p g)

theorem **aeval_genericPoint_eq_zero_iff** (p : Ideal (MvPolynomial σ K)) (g : MvPolynomial σ K) : aeval (genericPoint p) g = 0 ↔ g ∈ p

theorem **genericPoint_ne_zero** {p : Ideal (MvPolynomial σ K)} (hp : ∃ j, X j ∉ p) : genericPoint p ≠ 0

theorem **isPrime_elimIdeal** {p : Ideal (MvPolynomial σ K)} [p.IsPrime] (hp : ∃ j, X j ∉ p) : (elimIdeal β p).IsPrime

theorem **weight_blockWeight** [Fintype σ] (d : β × σ →₀ ℕ) (i : β) : Finsupp.weight blockWeight d i = ∑ j, d (i, j)

theorem **isBlockHomogeneous_iff** [Fintype σ] {f : MvPolynomial (β × σ) K} {a : β → ℕ} : IsBlockHomogeneous f a ↔ ∀ d ∈ f.support, ∀ i, ∑ j, d (i, j) = a i

theorem **IsBlockHomogeneous.mul** {f g : MvPolynomial (β × σ) K} {a b : β → ℕ} (hf : IsBlockHomogeneous f a) (hg : IsBlockHomogeneous g b) : IsBlockHomogeneous (f * g) (a + b)

theorem **IsBlockHomogeneous.unique** {f : MvPolynomial (β × σ) K} {a b : β → ℕ} (h : f ≠ 0) (ha : IsBlockHomogeneous f a) (hb : IsBlockHomogeneous f b) : a = b

theorem **IsBlockHomogeneous.isHomogeneous** [Fintype σ] [Fintype β] {f : MvPolynomial (β × σ) K} {a : β → ℕ} (h : IsBlockHomogeneous f a) : f.IsHomogeneous (∑ i, a i)

theorem **IsBlockHomogeneous.rename** [Fintype σ] (e : β ≃ β') {f : MvPolynomial (β × σ) K} {a : β → ℕ} (h : IsBlockHomogeneous f a) : IsBlockHomogeneous (rename (Prod.map e id) f) (a ∘ e.symm)

theorem **IsBlockHomogeneous.aeval_smul** [Fintype σ] [Fintype β] {G : Type*} [CommRing G] [Algebra K G] {f : MvPolynomial (β × σ) K} {a : β → ℕ} (h : IsBlockHomogeneous f a) (c : β → G) (t : β × σ → G) : aeval (fun v => c v.1 * t v) f = (∏ i, c i ^ a i) * aeval t f

theorem **isBlockHomogeneous_linForm** [Fintype σ] (u : σ → K) (i : β) : IsBlockHomogeneous (linForm u i) (Pi.single i 1)

theorem **map_weightedHomogeneousComponent** {S τ M : Type*} [CommRing S] [AddCommMonoid M] (φ : K →+* S) (w : τ → M) (a : M) (f : MvPolynomial τ K) : map φ (weightedHomogeneousComponent w a f) = weightedHomogeneousComponent w a (map φ f)

theorem **weightedHomogeneousComponent_mem_elimIdeal** [Fintype σ] [Finite β] {p : Ideal (MvPolynomial σ K)} {f : MvPolynomial (β × σ) K} (hf : f ∈ elimIdeal β p) (a : β → ℕ) : weightedHomogeneousComponent blockWeight a f ∈ elimIdeal β p

theorem **exists_eq_C_mul_of_dvd** {f g : MvPolynomial τ K} (hg : g ≠ 0) (h : f ∣ g) (hd : g.totalDegree ≤ f.totalDegree) : ∃ c : K, c ≠ 0 ∧ g = C c * f

theorem **exists_isWeightedHomogeneous_of_dvd** (w : τ → M) {f : MvPolynomial τ K} (h : ∀ a, f ∣ weightedHomogeneousComponent w a f) : ∃ a, IsWeightedHomogeneous w f a

theorem **exists_eq_C_mul_of_span_eq** {f g : MvPolynomial τ K} (h : Ideal.span {f} = Ideal.span {g}) : ∃ c : K, c ≠ 0 ∧ g = C c * f

Development.AbelianVarietyTheory.Definitions.ThetaValues.ArchAtEmptyOfRhoZero

theorem **evG_single** (m : Deg D j) (f : T.R j) : evG h j hj (AddMonoidAlgebra.single m f) = rdRing h j hj f

theorem **evG_smul** (a : T.Q.c.F) (F : AddMonoidAlgebra (T.R j) (Deg D j)) : evG h j hj (a • F) = T.σ₀ a • evG h j hj F

theorem **evG_indep** (u : WeightFamilies.Wt D j) (m : ℕ) (F : Fin m → (recB h.toLawsCore).Cusp j u) (hF : LinearIndependent (recB h.toLawsCore).KO F) : LinearIndependent ℂ fun i => evG h j hj (incG deg (cutR T.C j) u (F i))

theorem **isRdgU_rdgU** (n : SecNorm (recB h.toLawsCore)) : IsRdgU h j hj n (rdgU h j hj n) (Equiv.refl _)

theorem **li_congr** {ι M : Type} [AddCommGroup M] [Module ℂ M] {v v' : ι → M} (h : LinearIndependent ℂ v) (e : ∀ i, v i = v' i) : LinearIndependent ℂ v'

@[simp] theorem **ex_zero** (a : ℕ) (α : Fin n → ℕ) : ex a α 0 = a

@[simp] theorem **ex_succ** (a : ℕ) (α : Fin n → ℕ) (i : Fin n) : ex a α i.succ = α i

theorem **degree_eq** (β : Fin (n + 1) →₀ ℕ) : β.degree = β 0 + tl β

theorem **mono_ex** (a : ℕ) (α : Fin n → ℕ) : (monomial (ex a α) (1 : ℂ)) = (∏ i, X i.succ ^ α i) * X 0 ^ a

theorem **exists_le_sum** (γ : Fin n → ℕ) (a : ℕ) (h : a ≤ ∑ i, γ i) : ∃ α : Fin n → ℕ, α ≤ γ ∧ ∑ i, α i = a

theorem **eval_single_eq_coeff** (p : MvPolynomial (Fin (n + 1)) ℂ) {k : ℕ} (hp : p.IsHomogeneous k) : eval (Pi.single 0 1) p = coeff (ex k 0) p

theorem **mon_eq** (α : Fin (WeightFamilies.dimY D j) → ℕ) (a : ℕ) : R.mon w t α * R.s w t 0 ^ a = φ (monomial (ex a α) 1)

theorem **mono_mem_deep** (u : WeightFamilies.Wt D j) (a : ℕ) (β : Fin (WeightFamilies.dimY D j + 1) →₀ ℕ) (c : ℂ) (hd : β.degree = u.n / 2 + a) (ha : a ≤ tl β) : φ (monomial β c) ∈ R.deep w t u a

theorem **deep_le** (u : WeightFamilies.Wt D j) (a : ℕ) : R.deep w t u a ≤ (supp fun β => a ≤ tl β).map φ.toLinearMap

theorem **anFacts_of_poly** (hφ : Function.Injective φ) (x0 : R.Y) (hx : ∀ p, φ p x0 = eval (Pi.single 0 1) p) (hL : wL.n = 2) (ka wF : WeightFamilies.Wt D j) (hF : wF.n = 0) (T : ℕ) (hT : WeightFamilies.dimY D j ≤ T) (hk : ka.n = 2 * T) : AnFacts R w t ka wF T ⊤ (fun _ : Fin 1 => x0)

theorem **hN_norm** (j : ℕ) (_ : InR D j) (w : WeightFamilies.Wt D j) (σ : B₀.KO →+* ℂ) : IsNormOn (hC B₀ j w) σ (hN B₀ j w σ) where

theorem **hC_fd** (j : ℕ) (_ : InR D j) (w : WeightFamilies.Wt D j) : FiniteDimensional B₀.KO (hC B₀ j w)

theorem **fn_apply** {n : ℕ} (p : MvPolynomial (Fin n) ℂ) (y : Fin n → ℂ) : fn n p y = eval y p

theorem **fn_inj** (n : ℕ) : Function.Injective (fn n)

theorem **evalEmbedding_single** {n : ℕ} {M : Type} [AddCommMonoid M] (m : M) (f : MvPolynomial (Fin n) K) : evP σ n M (AddMonoidAlgebra.single m f) = fn n (MvPolynomial.map σ f)

theorem **evalEmbedding_smul** {n : ℕ} {M : Type} [AddCommMonoid M] (a : K) (F : AddMonoidAlgebra (MvPolynomial (Fin n) K) M) : evP σ n M (a • F) = σ a • evP σ n M F

theorem **map_indep** {n m : ℕ} (F : Fin m → MvPolynomial (Fin n) K) (hF : LinearIndependent K F) : LinearIndependent ℂ fun i => MvPolynomial.map σ (F i)

theorem **exampleEvaluation_span_univ** (σ : B₀.KO →+* ℂ) (u : WeightFamilies.Wt D j) : (exampleEvaluation X₀ hj σ).sp u Set.univ = (homogeneousSubmodule _ ℂ (u.n / 2)).map (fn _).toLinearMap

theorem **toy** (σ : B₀.KO →+* ℂ) (T : ℕ) (hT : WeightFamilies.dimY D j ≤ T) : AnFacts (exampleEvaluation X₀ hj σ) (toyW hj) (fun p => if p.1 = p.2 then 1 else 0) (wscalar D j (2 * T) (even_two_mul T)) (wscalar D j 0 Even.zero) T ⊤ (fun _ : Fin 1 => (Pi.single 0 1 : Fin (WeightFamilies.dimY D j + 1) → ℂ))

theorem **homRel_of_monomials** (σ : K →+* ℂ) (X : Data E r) {m d : ℕ} (hm : 3 < m) (hd : 0 < d) (v : Fin 4 → Frame X.Ψ r → ℂ) (u : Fin m → Frame X.Ψ r → ℂ) (hu : ∀ a, ∃ e : Fin 4 → ℕ, e 0 + e 1 = d ∧ e 2 + e 3 = d ∧ ∀ x, u a x = ∏ k, v k x ^ e k) : HomRel σ X u

theorem **margins_of_definite** (X : Data E r) (hp : X.p0 = 0 ∨ X.p0 = r) (Xa : ArchInM X) (hC : ∃ C : ℝ, ∀ x ∈ X.dom, Xa.ps0 x ≤ C) (hiso : ∀ ξ, ¬ X.IsIso ξ) : Margins X Xa

theorem **relIn_margins_lines** (σ : K →+* ℂ) (h : Module.finrank ℚ E = 4) : ∃ (X : Data E 2) (Xa : ArchInM X) (Fl : FlatM X Xa), Fl.δ = 2 ∧ RelIn σ X Xa ∧ Margins X Xa

theorem **archAt_empty_of_rho_zero** {D : D7} {T : Top D} (h : LawsP T) (j : ℕ) {N : ℕ} {wL wF : WeightFamilies.Wt D j} (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (CR Tt c : ℝ) (h0 : WeightFamilies.rho D j = 0) (k : (T.U j T.σ₀).Comp) : IsEmpty (ArchAt h j w wF Tn (WeightFamilies.c1 D CR Tt c j))

Development.AbelianVarietyTheory.Definitions.ThetaValues.ExistsLawsCoreGamAt

theorem **l1At_nonneg** (P : MvPolynomial ι K) : 0 ≤ l1At σ P

theorem **l1At_eq_zero_iff** (P : MvPolynomial ι K) : l1At σ P = 0 ↔ P = 0

theorem **l1At_eq_sum** (P : MvPolynomial ι K) (s : Finset (ι →₀ ℕ)) (hs : P.support ⊆ s) : l1At σ P = ∑ m ∈ s, ‖σ (P.coeff m)‖

theorem **l1At_add_le** (P Q : MvPolynomial ι K) : l1At σ (P + Q) ≤ l1At σ P + l1At σ Q

theorem **l1At_monomial** (m : ι →₀ ℕ) (a : K) : l1At σ (MvPolynomial.monomial m a) = ‖σ a‖

theorem **l1At_smul** (a : K) (P : MvPolynomial ι K) : l1At σ (a • P) = ‖σ a‖ * l1At σ P

theorem **norm_eval₂_le** (P : MvPolynomial ι K) (z : ι → ℂ) (hz : ∀ i, ‖z i‖ ≤ 1) : ‖MvPolynomial.eval₂ σ z P‖ ≤ l1At σ P

theorem **gamAt_iff** (T : Core D) : GamAt T ↔ ∀ j, InR D j → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → GlobalTheory.FrameNorm.HasPrinc ((⟨(T.U j σ).Comp, (T.U j σ).X, (T.U j σ).un⟩ : Formalism.L2.PjFree.Unf _ _ σ _) T.Q.l.M

theorem **range_nonempty** (T : Core D) : ∃ j, InR D j ∧ ∃ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ

theorem **mem_of_modelBase** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {r g : ℕ} {ty : Fin g → (E →+* ℂ)} {σ' : KA l →+* ℂ} {F : FamilyUnif Xo σ' ty} {M : ModelsOf F r} (hM : ModelBase M) (hA : l.A ∣ l.M) (k : Fin F.n) (γ : GL (Fin r) E) (hγ : IsUStab (M.X k).Gm (M.X k).Λ γ) (h : ∀ x ∈ (M.X k).Λ, ∃ y ∈ (M.X k).Λ, Matrix.mulVec (γ : Matrix (Fin r) (Fin r) E) x − x = (l.M : E) • y) : γ ∈ (M.X k).Γ

theorem **exists_lawsCore_gamAt** {hN : N1c} {Q : Par D} {hpq : ∀ j, 1 ≤ (Q.Dw j).p + (Q.Dw j).q} (L : LawsRemainingRows hN Q hpq) (hQ : Q.Pins) : ∃ T : Core D, T.Q = Q ∧ LawsCore T ∧ GamAt T

theorem **exists_xi** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ∃ ξ : E, cc ξ = -ξ ∧ ξ ≠ 0 ∧ ∀ τ : E →+* ℂ, τ (H.rt : E) = H.rt0 → (τ ξ).im < 0

theorem **exists_gamAt** (T : Core D) (hQ : T.Q.Pins) : ∃ T' : Core D, T'.Q = T.Q ∧ GamAt T' ∧ ∀ j, InR D j → ∀ σ, GoodEmb T'.Q.c σ → IsUnf (T'.U j σ) ∧ Nonempty (T'.U j σ).Comp

Development.AbelianVarietyTheory.Definitions.ThetaValues.ExistsMemRowHighestTotalDegreeEqTwo

theorem **rowAddX_eq_rowAdd** (t : T) (a b : Fin p) (s : R) : (rowAddX t a b s : Poly R T p q κ →ₐ[R] _) = rowAdd t (Sum.inl a) (Sum.inl b) s

theorem **rowAddY_eq_rowAdd** (t : T) (a b : Fin q) (s : R) : (rowAddY t a b s : Poly R T p q κ →ₐ[R] _) = rowAdd t (Sum.inr a) (Sum.inr b) s

theorem **rowAdd_eq_bind₁** [DecidableEq (Var T p q κ)] (t : T) (r r' : Fin p ⊕ Fin q) (s : R) : (rowAdd t r r' s : Poly R T p q κ →ₐ[R] _) = bind₁ fun v => X v + ∑ k : κ, if v = (t, r', k) then C s * X (t, r, k) else 0

theorem **rowAddX_eq_bind₁** [DecidableEq (Var T p q κ)] (t : T) (a b : Fin p) (s : R) : (rowAddX t a b s : Poly R T p q κ →ₐ[R] _) = bind₁ fun v => X v + ∑ k : κ, if v = xVar t b k then C s * X (xVar t a k) else 0

theorem **rowAddY_eq_bind₁** [DecidableEq (Var T p q κ)] (t : T) (a b : Fin q) (s : R) : (rowAddY t a b s : Poly R T p q κ →ₐ[R] _) = bind₁ fun v => X v + ∑ k : κ, if v = yVar t b k then C s * X (yVar t a k) else 0

@[simp] theorem **rowAdd_X_self** (t : T) (r r' : Fin p ⊕ Fin q) (s : R) (k : κ) : rowAdd t r r' s (X (t, r', k)) = X (t, r', k) + C s * X (t, r, k)

theorem **rowAdd_X_of_ne** {t t' : T} (r : Fin p ⊕ Fin q) {r' r'' : Fin p ⊕ Fin q} (s : R) (k : κ) (h : ¬ (t' = t ∧ r'' = r')) : rowAdd t r r' s (X (t', r'', k)) = X (t', r'', k)

theorem **rowAdd_X** [DecidableEq T] (t t' : T) (r r' r'' : Fin p ⊕ Fin q) (s : R) (k : κ) : rowAdd t r r' s (X (t', r'', k)) = X (t', r'', k) + if t' = t ∧ r'' = r' then C s * X (t, r, k) else 0

@[simp] theorem **rowAddX_X_self** (t : T) (a b : Fin p) (s : R) (k : κ) : rowAddX t a b s (X (xVar t b k) : Poly R T p q κ) = X (xVar t b k) + C s * X (xVar t a k)

@[simp] theorem **rowAddY_X_self** (t : T) (a b : Fin q) (s : R) (k : κ) : rowAddY t a b s (X (yVar t b k) : Poly R T p q κ) = X (yVar t b k) + C s * X (yVar t a k)

theorem **rowAddX_X_of_ne** {t t' : T} (a : Fin p) {b b' : Fin p} (s : R) (k : κ) (h : ¬ (t' = t ∧ b' = b)) : rowAddX t a b s (X (xVar t' b' k) : Poly R T p q κ) = X (xVar t' b' k)

theorem **rowAddY_X_of_ne** {t t' : T} (a : Fin q) {b b' : Fin q} (s : R) (k : κ) (h : ¬ (t' = t ∧ b' = b)) : rowAddY t a b s (X (yVar t' b' k) : Poly R T p q κ) = X (yVar t' b' k)

@[simp] theorem **rowAddX_X_yVar** (t t' : T) (a b : Fin p) (b' : Fin q) (s : R) (k : κ) : rowAddX t a b s (X (yVar t' b' k) : Poly R T p q κ) = X (yVar t' b' k)

@[simp] theorem **rowAddY_X_xVar** (t t' : T) (a b : Fin q) (b' : Fin p) (s : R) (k : κ) : rowAddY t a b s (X (xVar t' b' k) : Poly R T p q κ) = X (xVar t' b' k)

theorem **rowAdd_rowForm_self** (t : T) (r r' : Fin p ⊕ Fin q) (s : R) (v : κ → R) : rowAdd t r r' s (rowForm t r' v) = rowForm t r' v + C s * rowForm t r v

theorem **rowAdd_rowForm_of_ne** {t t' : T} (r : Fin p ⊕ Fin q) {r' r'' : Fin p ⊕ Fin q} (s : R) (v : κ → R) (h : ¬ (t' = t ∧ r'' = r')) : rowAdd t r r' s (rowForm t' r'' v) = rowForm t' r'' v

@[simp] theorem **rowAdd_zero** (t : T) (r r' : Fin p ⊕ Fin q) : (rowAdd t r r' (0 : R) : Poly R T p q κ →ₐ[R] _) = AlgHom.id R _

theorem **rowAdd_add** (t : T) {r r' : Fin p ⊕ Fin q} (h : r ≠ r') (s s' : R) : (rowAdd t r r' (s + s') : Poly R T p q κ →ₐ[R] _) = (rowAdd t r r' s).comp (rowAdd t r r' s')

theorem **eval_rowAdd** [DecidableEq T] (z : Var T p q κ → R) (t : T) (r r' : Fin p ⊕ Fin q) (s : R) (P : Poly R T p q κ) : eval z (rowAdd t r r' s P) = eval (fun v => z v + if v.1 = t ∧ v.2.1 = r' then s * z (t, r, v.2.2) else 0) P

theorem **map_rowAdd** (f : R →+* S) (t : T) (r r' : Fin p ⊕ Fin q) (s : R) (P : Poly R T p q κ) : MvPolynomial.map f (rowAdd t r r' s P) = rowAdd t r r' (f s) (MvPolynomial.map f P)

theorem **rowAdd_colSubst** [Fintype κ'] (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) {r r' : Fin p ⊕ Fin q} (h : r.isLeft = r'.isLeft) (s : R) (P : Poly R T p q κ) : rowAdd t r r' s (colSubst A B P) = colSubst A B (rowAdd t r r' s P)

theorem **rowAdd_det_rowForm** {ι : Type*} [Fintype ι] [DecidableEq ι] (t t' : T) {r r' : Fin p ⊕ Fin q} (hr : r ≠ r') (s : R) {ρ : ι → Fin p ⊕ Fin q} (hρ : Function.Injective ρ) (h : r' ∈ Set.range ρ → r ∈ Set.range ρ) (v : ι → κ → R) : rowAdd t r r' s (Matrix.of fun a b => rowForm t' (ρ a) (v b)).det = (Matrix.of fun a b => rowForm t' (ρ a) (v b)).det

theorem **mem_range_lead_of_lt** {m : ℕ} {i a b : Fin m} (h : a < b) (hb : b ∈ Set.range (lead i)) : a ∈ Set.range (lead i)

theorem **RowLT.ne** {r r' : Fin p ⊕ Fin q} (h : RowLT r r') : r ≠ r'

theorem **RowLT.isLeft_eq** {r r' : Fin p ⊕ Fin q} (h : RowLT r r') : r.isLeft = r'.isLeft

theorem **rowAdd_det_xForm** (t t' : T) {r r' : Fin p ⊕ Fin q} (h : RowLT r r') (s : R) (i : Fin p) (v : Fin (i.val + 1) → κ → R) : rowAdd t r r' s (Matrix.of fun a b => xForm t' (lead i a) (v b)).det = (Matrix.of fun a b => xForm t' (lead i a) (v b)).det

theorem **rowAdd_det_yForm** (t t' : T) {r r' : Fin p ⊕ Fin q} (h : RowLT r r') (s : R) (i : Fin q) (v : Fin (i.val + 1) → κ → R) : rowAdd t r r' s (Matrix.of fun a b => yForm t' (lead i a) (v b)).det = (Matrix.of fun a b => yForm t' (lead i a) (v b)).det

theorem **rowAdd_minorProd** [Fintype T] (t : T) {r r' : Fin p ⊕ Fin q} (h : RowLT r r') (s : R) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : rowAdd t r r' s (minorProd c c' ε φ) = minorProd c c' ε φ

theorem **rowAdd_blockPoly** [Fintype T] (t : T) {r r' : Fin p ⊕ Fin q} (h : RowLT r r') (s : R) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (col : T → (i : Fin p) → Fin (i.val + 1) → κ) (col' : T → (i : Fin q) → Fin (i.val + 1) → κ) : rowAdd t r r' s (blockPoly R c c' col col') = blockPoly R c c' col col'

theorem **rowAdd_eq_self_of_forall_int** [IsDomain R] [CharZero R] {t : T} {r r' : Fin p ⊕ Fin q} {P : Poly R T p q κ} (h : ∀ s : ℤ, rowAdd t r r' (s : R) P = P) (s : R) : rowAdd t r r' s P = P

theorem **rowInd_apply** [DecidableEq T] (t : T) (r : Fin p ⊕ Fin q) (v : Var T p q κ) : rowInd t r v = if v.1 = t ∧ v.2.1 = r then 1 else 0

theorem **weight_rowInd** (t : T) (r : Fin p ⊕ Fin q) (m : Var T p q κ →₀ ℕ) : Finsupp.weight (rowInd t r) m = ∑ k : κ, m (t, r, k)

theorem **isWeightedHomogeneous_rowInd_iff** (t : T) (r : Fin p ⊕ Fin q) (P : Poly R T p q κ) (d : ℕ) : IsWeightedHomogeneous (rowInd t r) P d ↔ ∀ m ∈ P.support, ∑ k : κ, m (t, r, k) = d

theorem **weight_rowWeight** [Fintype T] {M : Type*} [AddCommMonoid M] (w : T → Fin p ⊕ Fin q → M) (m : Var T p q κ →₀ ℕ) : Finsupp.weight (rowWeight w) m = ∑ t, ((∑ a, (∑ k, m (xVar t a k)) • w t (Sum.inl a)) + ∑ b, (∑ k, m (yVar t b k)) • w t (Sum.inr b))

theorem **mem_rowHighest_iff** : P ∈ rowHighest R c c' ↔ IsHarmonic P ∧ (∀ t a, IsWeightedHomogeneous (rowInd t (Sum.inl a)) P (partOf (c t) a)) ∧ (∀ t b, IsWeightedHomogeneous (rowInd t (Sum.inr b)) P (partOf (c' t) b)) ∧ (∀ t a b, a < b → ∀ s : ℤ, rowAddX t a b (s : R) P = P) ∧ ∀ t a b, a < b → ∀ s : ℤ, rowAddY t a b (s : R) P = P

theorem **mem_rowHighest_iff_support** : P ∈ rowHighest R c c' ↔ IsHarmonic P ∧ (∀ t a, ∀ m ∈ P.support, ∑ k, m (xVar t a k) = partOf (c t) a) ∧ (∀ t b, ∀ m ∈ P.support, ∑ k, m (yVar t b k) = partOf (c' t) b) ∧ (∀ t a b, a < b → ∀ s : ℤ, rowAddX t a b (s : R) P = P) ∧ ∀ t a b, a < b → ∀ s : ℤ, rowAddY t a b (s : R) P = P

theorem **mem_rowHighest_iff_of_charZero** [IsDomain R] [CharZero R] : P ∈ rowHighest R c c' ↔ IsHarmonic P ∧ (∀ t a, ∀ m ∈ P.support, ∑ k, m (xVar t a k) = partOf (c t) a) ∧ (∀ t b, ∀ m ∈ P.support, ∑ k, m (yVar t b k) = partOf (c' t) b) ∧ (∀ t a b, a < b → ∀ s : R, rowAddX t a b s P = P) ∧ ∀ t a b, a < b → ∀ s : R, rowAddY t a b s P = P

theorem **mem_rowHighest_iff_rowAdd** : P ∈ rowHighest R c c' ↔ IsHarmonic P ∧ (∀ t r, IsWeightedHomogeneous (rowInd t r) P (Sum.elim (partOf (c t)) (partOf (c' t)) r)) ∧ ∀ t r r', RowLT r r' → ∀ s : ℤ, rowAdd t r r' (s : R) P = P

theorem **isHarmonic_of_mem_rowHighest** (hP : P ∈ rowHighest R c c') : IsHarmonic P

theorem **rowAddX_eq_self_of_mem_rowHighest** [IsDomain R] [CharZero R] (hP : P ∈ rowHighest R c c') (t : T) {a b : Fin p} (hab : a < b) (s : R) : rowAddX t a b s P = P

theorem **rowAddY_eq_self_of_mem_rowHighest** [IsDomain R] [CharZero R] (hP : P ∈ rowHighest R c c') (t : T) {a b : Fin q} (hab : a < b) (s : R) : rowAddY t a b s P = P

theorem **rowHighest_le_harmonicSubmodule** : rowHighest R c c' ≤ harmonicSubmodule R T p q κ

theorem **isWeightedHomogeneous_of_mem_rowHighest** [Fintype T] {M : Type*} [AddCommMonoid M] (w : T → Fin p ⊕ Fin q → M) (hP : P ∈ rowHighest R c c') : IsWeightedHomogeneous (rowWeight w) P (∑ t, ((∑ a, partOf (c t) a • w t (Sum.inl a)) + ∑ b, partOf (c' t) b • w t (Sum.inr b)))

theorem **isHomogeneous_of_mem_rowHighest** [Fintype T] (hP : P ∈ rowHighest R c c') : P.IsHomogeneous (typeDegree c c')

theorem **rowHighest_le_homogeneousSubmodule** [Fintype T] : rowHighest R c c' ≤ homogeneousSubmodule (Var T p q κ) R (typeDegree c c')

theorem **mem_rowHighest_of_rowWeight** [Fintype T] (h : IsHarmonic P) (hw : ∀ w : T → Fin p ⊕ Fin q → ℕ, IsWeightedHomogeneous (rowWeight w) P (∑ t, ((∑ a, partOf (c t) a • w t (Sum.inl a)) + ∑ b, partOf (c' t) b • w t (Sum.inr b)))) (hr : ∀ t r r', RowLT r r' → ∀ s : ℤ, rowAdd t r r' (s : R) P = P) : P ∈ rowHighest R c c'

theorem **minorProd_mem_rowHighest** [Fintype T] {ε : T → Fin p → κ → R} {φ : T → Fin q → κ → R} (h : IsPerp ε φ) : minorProd c c' ε φ ∈ rowHighest R c c'

theorem **blockPoly_mem_rowHighest** [Fintype T] (col : T → (i : Fin p) → Fin (i.val + 1) → κ) (col' : T → (i : Fin q) → Fin (i.val + 1) → κ) (h : ∀ t i j a b, c t i ≠ 0 → c' t j ≠ 0 → col t i a ≠ col' t j b) : blockPoly R c c' col col' ∈ rowHighest R c c'

theorem **colSubst_mem_rowHighest** [Fintype κ'] [DecidableEq κ] {A : T → Matrix κ κ' R} {B : T → Matrix κ' κ R} (h : ∀ t, A t ∗ B t = 1) (hP : P ∈ rowHighest R c c') : colSubst A B P ∈ rowHighest R c c'

theorem **colAct_mem_rowHighest** [DecidableEq κ] (g : T → GL κ R) (hP : P ∈ rowHighest R c c') : colAct R T p q κ g P ∈ rowHighest R c c'

theorem **colAct_mem_rowHighest_iff** [DecidableEq κ] (g : T → GL κ R) : colAct R T p q κ g P ∈ rowHighest R c c' ↔ P ∈ rowHighest R c c'

theorem **map_mem_rowHighest** (f : R →+* S) (hP : P ∈ rowHighest R c c') : MvPolynomial.map f P ∈ rowHighest S c c'

theorem **map_mem_rowHighest_iff** {f : R →+* S} (hf : Function.Injective f) : MvPolynomial.map f P ∈ rowHighest S c c' ↔ P ∈ rowHighest R c c'

theorem **basePoly_mem_rowHighest** : basePoly R h c c' ∈ rowHighest R c c'

theorem **orbitSpan_le_rowHighest** : orbitSpan R h c c' ≤ rowHighest R c c'

theorem **rowHighest_ne_bot** [Nontrivial R] : (rowHighest R c c' : Submodule R (Poly R T p q (Fin n))) ≠ ⊥

theorem **exists_mem_rowHighest_totalDegree_eq_two** : ∃ P ∈ rowHighest ℤ (fun (_ : Unit) (_ : Fin 1) => 1) (fun _ (_ : Fin 1) => 1), P = (X (xVar () 0 0) ∗ X (yVar () 0 1) : Poly ℤ Unit 1 1 (Fin 2)) ∧ P.totalDegree = 2

Development.AbelianVarietyTheory.Definitions.ThetaValues.ExistsRootNotMem

theorem **sOf_tw** {ta : c.F ≃+* c.F} (h : Fixes c ta) (S : Type) [CommRing S] [Algebra c.F S] : sOf c (Tw ta S) = sOf c S

theorem **isEigAt_tw** {ta : c.F ≃+* c.F} (h : Fixes c ta) {S : Type} [CommRing S] [Algebra c.F S] (τ : E →+* ℂ) (v : Sp (DualNumber S) ⟶ Xo.univ.t.G.left) : IsEigAt (Xo

theorem **fixes_refl** : Fixes c (RingEquiv.refl c.F)

theorem **Tw.val_algebraMap** {K : Type} [Field K] (ta : K ≃+* K) (S : Type) [CommRing S] [Algebra K S] (a : K) : (algebraMap K (Tw ta S) a).val = algebraMap K S (ta a)

theorem **fixes_of_le** {c c' : Coeff l} (ta : c'.F ≃+* c'.F) (h : ∀ x : c'.F, (x : ℂ) ∈ c.F → ta x = x) : Fixes c' ta

theorem **galRul_smul** {ta : c.F ≃+* c.F} (h : Fixes c ta) (a : c.F) (f : Rul Xo c) : galRul h (a • f) = ta a • galRul h f

theorem **covAut_apply_eF** {n : ℕ} (d : DeltaDat Xo c n) (ta : c.F ≃+* c.F) (a : c.F) : covAut d ta (d.eF a) = d.eF (ta a)

theorem **fixes_covAut** {n : ℕ} (d : DeltaDat Xo c n) (hA : d.l₂.A = 1) {ta : c.F ≃+* c.F} (h : Fixes c ta) : Fixes d.c₂ (covAut d ta)

Development.AbelianVarietyTheory.Definitions.ThetaValues.ExistsUniqueSecIsFnAt

theorem **valAt'_eq_valHom** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)) : valAt' x U f = valHom x U h f

theorem **valAt'_mul** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (f f' : Γ(X, U)) : valAt' x U (f ∗ f') = valAt' x U f ∗ valAt' x U f'

theorem **valAt'_add** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (f f' : Γ(X, U)) : valAt' x U (f + f') = valAt' x U f + valAt' x U f'

theorem **valAt'_sum** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) {I : Type*} (t : Finset I) (f : I → Γ(X, U)) : valAt' x U (∑ i ∈ t, f i) = ∑ i ∈ t, valAt' x U (f i)

theorem **valAt'_unit_zpow** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)ˣ) (k : ℤ) : valAt' x U (f ^ k).val = valAt' x U f.val ^ k

theorem **valAt'_res** {U V : X.Opens} (h : V ≤ U) (hV : ⊤ ≤ x ⁻¹ᵁ V) (f : Γ(X, U)) : valAt' x V ((X.presheaf.map (homOfLE h).op).hom f) = valAt' x U f

theorem **valAt'_app** (φ : X ⟶ Y) (U : Y.Opens) (f : Γ(Y, U)) : valAt' x (φ ⁻¹ᵁ U) ((φ.app U).hom f) = valAt' (x ≫ φ) U f

theorem **valAt'_appLE'** (φ : X ⟶ Y) {U : Y.Opens} {V : X.Opens} (e : V ≤ φ ⁻¹ᵁ U) (hV : ⊤ ≤ x ⁻¹ᵁ V) (f : Γ(Y, U)) : valAt' x V ((φ.appLE U V e).hom f) = valAt' (x ≫ φ) U f

theorem **valAt'_sec** {M : LB X} (s : M.Sec) (i j : M.ι) (hi : ⊤ ≤ x ⁻¹ᵁ M.U i) (hj : ⊤ ≤ x ⁻¹ᵁ M.U j) : valAt' x (M.U i) (s.f i) = valAt' x (M.U i ⊓ M.U j) (M.g i j).val * valAt' x (M.U j) (s.f j)

theorem **valAt'_id_top** (r : Γ(Sp ℂ, ⊤)) : valAt' (𝟙 (Sp ℂ)) ⊤ r = (Scheme.ΓSpecIso (CommRingCat.of ℂ)).hom.hom r

theorem **ext_of_valAt'_of_isIso** [IsIso x] {U : X.Opens} {f f' : Γ(X, U)} (h : ⊤ ≤ x ⁻¹ᵁ U → valAt' x U f = valAt' x U f') : f = f'

theorem **res_appLE** (h : Z ⟶ Y) {U U' : Y.Opens} {V V' : Z.Opens} (hU : U' ≤ U) (hV : V' ≤ V) (e : V ≤ h ⁻¹ᵁ U) (e' : V' ≤ h ⁻¹ᵁ U') (f : Γ(Y, U)) : (Z.presheaf.map (homOfLE hV).op).hom ((h.appLE U V e).hom f) = (h.appLE U' V' e').hom ((Y.presheaf.map (homOfLE hU).op).hom f)

theorem **wchart_le_of_comp** {φ : Y ⟶ X} {ψ : Z ⟶ X} {h : Z ⟶ Y} (e : h ≫ φ = ψ) (j : K' → L.ι) : LB.wchart L (fun _ : K' => ψ) j ≤ h ⁻¹ᵁ LB.wchart L (fun _ : K => φ) (fun _ => j default)

theorem **le_wchart_unique** {W : Scheme.{0}} (x : W ⟶ Y) (φ : Y ⟶ X) (j : K → L.ι) : ⊤ ≤ x ⁻¹ᵁ LB.wchart L (fun _ : K => φ) j ↔ ⊤ ≤ (x ≫ φ) ⁻¹ᵁ L.U (j default)

theorem **valAt'_pullPow** {φ : Y ⟶ X} {ψ : Z ⟶ X} (h : Z ⟶ Y) (e : h ≫ φ = ψ) (a : ℤ) (s : LB.Sec (LB.LinComb L (fun _ : K => φ) (fun _ => a))) (x : Sp ℂ ⟶ Z) (j : K' → L.ι) (hj : ⊤ ≤ x ⁻¹ᵁ LB.wchart L (fun _ : K' => ψ) j) : valAt' x (LB.wchart L (fun _ : K' => ψ) j) ((pullPow L (K'

theorem **pullPow_pullPow** {φ : Y ⟶ X} {ψ : Z ⟶ X} (h : Z ⟶ Y) (e : h ≫ φ = ψ) (h' : Y ⟶ Z) (e' : h' ≫ ψ = φ) (hh : h' ≫ h = 𝟙 Y) (a : ℤ) (s : LB.Sec (LB.LinComb L (fun _ : K => φ) (fun _ => a))) : pullPow L (K'

theorem **fibreIncl_snd** : (fibreIncl A p).left ≫ (snd T A).left = 𝟙 A.left

theorem **snd_fibreIncl** [IsIso p] : (snd T A).left ≫ (fibreIncl A p).left = 𝟙 (T ⊗ A).left

theorem **isIso_snd_left** [IsIso p] : IsIso (snd T A).left

theorem **exAt_snd** (z : Fin g → ℂ) : exAt u p z ≫ (snd T A).left = (u.ex z).left

theorem **exAt_fst** (z : Fin g → ℂ) : exAt u p z ≫ (fst T A).left = p.left

theorem **ex_fibreIncl** (z : Fin g → ℂ) : (u.ex z).left ≫ (fibreIncl A p).left = exAt u p z

theorem **exAt_add_period** (z l : Fin g → ℂ) (hl : l ∈ u.Λ) : exAt u p (z + l) = exAt u p z

theorem **comp_ptSecT** {x : T ⟶ A} {z : Fin g → ℂ} (hx : p ≫ x = u.ex z) : p.left ≫ (ptSecT A x).left = exAt u p z

theorem **exAt_mem_chart_iff** (k : ℤ) (j : Fin 1 → L.ι) (z : Fin g → ℂ) : ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j ↔ ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U (j 0)

theorem **exAt_hol** (U : (T ⊗ A).left.Opens) (f : Γ((T ⊗ A).left, U)) : IsOpen {z | ⊤ ≤ exAt u p z ⁻¹ᵁ U} ∧ DifferentiableOn ℂ (fun z => valAt' (exAt u p z) U f) {z | ⊤ ≤ exAt u p z ⁻¹ᵁ U}

theorem **valAt'_liftFun** {W : (T ⊗ A).left.Opens} {z : Fin g → ℂ} (hW : ⊤ ≤ exAt u p z ⁻¹ᵁ W) (r : Γ(T.left, ⊤)) : valAt' (exAt u p z) W (liftFun A W r) = valAt' p.left ⊤ r

theorem **valAt'_LPow_g** (k : ℤ) (j j' : Fin 1 → L.ι) (z : Fin g → ℂ) (hj : ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j) (hj' : ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j') : valAt' (exAt u p z) ((LPow A L k T).U j ⊓ (LPow A L k T).U j') ((LPow A L k T).g j j').val = valAt' (u.ex z).left (L.U (j 0) ⊓ L.U (j' 0)) (L.g (j 0) (j' 0)).val ^ k

theorem **exists_chart_exAt** (hL : L.IsLB) (k : ℤ) (z : Fin g → ℂ) : ∃ j : Fin 1 → L.ι, ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j

theorem **fr_zpow_ne_zero** {k : ℤ} {j : Fin 1 → L.ι} {z : Fin g → ℂ} (hj : ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j) (m : ℤ) : u.fr (j 0) z ^ m ≠ 0

theorem **IsFnAt.eq_div** {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) {j : Fin 1 → L.ι} {z : Fin g → ℂ} (hj : ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j) : θ z = valAt' (exAt u p z) ((LPow A L k T).U j) (s.f j) / u.fr (j 0) z ^ k

theorem **isFnAt_of_exists_chart** {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : ∀ z, ∃ j : Fin 1 → L.ι, ⊤ ≤ exAt u p z ⁻¹ᵁ (LPow A L k T).U j ∧ valAt' (exAt u p z) ((LPow A L k T).U j) (s.f j) = θ z * u.fr (j 0) z ^ k) : IsFnAt u p k s θ

theorem **IsFnAt.unique** (hL : L.IsLB) {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ θ' : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) (h' : IsFnAt u p k s θ') : θ = θ'

theorem **existsUnique_isFnAt** (hL : L.IsLB) (k : ℤ) (s : LB.Sec (LPow A L k T)) : ∃! θ, IsFnAt u p k s θ

theorem **isFnAt_fnAt** (hL : L.IsLB) (k : ℤ) (s : LB.Sec (LPow A L k T)) : IsFnAt u p k s (fnAt u p hL k s)

theorem **isFnAt_iff_eq_fnAt** (hL : L.IsLB) {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} : IsFnAt u p k s θ ↔ θ = fnAt u p hL k s

theorem **IsFnAt.differentiable** (hL : L.IsLB) {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) : Differentiable ℂ θ

theorem **IsFnAt.add_period** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) (l : u.Λ) (z : Fin g → ℂ) : θ (z + l) = (χ l * Complex.exp (Real.pi * hermForm Hm z l + Real.pi / 2 * hermForm Hm l l)) ^ k * θ z

theorem **IsFnAt.isThetaFn** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {n : ℕ} {s : LB.Sec (LPow A L n T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p n s θ) : u.IsThetaFn Hm χ n θ

theorem **IsFnAt.add** {k : ℤ} {s s' t : LB.Sec (LPow A L k T)} {θ θ' : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) (h' : IsFnAt u p k s' θ') (ht : ∀ j, t.f j = s.f j + s'.f j) : IsFnAt u p k t (θ + θ')

theorem **IsFnAt.liftFun_mul** {k : ℤ} {s t : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) (r : Γ(T.left, ⊤)) (ht : ∀ j, t.f j = liftFun A _ r * s.f j) : IsFnAt u p k t (valAt' p.left ⊤ r • θ)

theorem **IsFnAt.sum_liftFun_mul** {k : ℤ} {I : Type*} (F : Finset I) {s : I → LB.Sec (LPow A L k T)} {t : LB.Sec (LPow A L k T)} {θ : I → (Fin g → ℂ) → ℂ} (h : ∀ i ∈ F, IsFnAt u p k (s i) (θ i)) (r : I → Γ(T.left, ⊤)) (ht : ∀ j, t.f j = ∑ i ∈ F, liftFun A _ (r i) * (s i).f j) : IsFnAt u p k t (∑ i ∈ F, valAt' p.left ⊤ (r i) • θ i)

theorem **isFnAt_zero** (k : ℤ) : IsFnAt u p k (H1.SectionArithmetic.secZero (LPow A L k T)) 0

theorem **fnAtHom_apply** (hL : L.IsLB) (k : ℤ) (s : SecM A L k T) : fnAtHom u p hL k s = fnAt u p hL k (SecM.toSec A s)

theorem **isFnAt_secOfFun** (r : Γ(T.left, ⊤)) : IsFnAt u p 0 (secOfFun A L r) (fun _ => valAt' p.left ⊤ r)

theorem **toFibre_ofFibre** (n : ℕ) (s : (lbPow L n).Sec) : toFibre L p n (ofFibre A L T n s) = s

theorem **ofFibre_toFibre** [IsIso p] (n : ℕ) (s : LB.Sec (LPow A L n T)) : ofFibre A L T n (toFibre L p n s) = s

theorem **isFnAt_iff_isFnPow** (n : ℕ) (s : LB.Sec (LPow A L n T)) (θ : (Fin g → ℂ) → ℂ) : IsFnAt u p n s θ ↔ u.IsFnPow n (toFibre L p n s) θ

theorem **isFnAt_ofFibre_iff** (n : ℕ) (s : (lbPow L n).Sec) (θ : (Fin g → ℂ) → ℂ) : IsFnAt u p n (ofFibre A L T n s) θ ↔ u.IsFnPow n s θ

theorem **existsUnique_sec_isFnAt** [IsIso p] {n : ℕ} {θ : (Fin g → ℂ) → ℂ} (h : ∃! s, u.IsFnPow n s θ) : ∃! s : LB.Sec (LPow A L n T), IsFnAt u p n s θ

theorem **mem_ptChart_iff** {x : T ⟶ A} {z : Fin g → ℂ} (hx : p ≫ x = u.ex z) (j : Fin 1 → L.ι) : ⊤ ≤ p.left ⁻¹ᵁ LB.wchart L (fun _ : Fin 1 => x.left) j ↔ ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U (j 0)

theorem **IsFnAt.valAt'_ptSecT** {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) {x : T ⟶ A} {z : Fin g → ℂ} (hx : p ≫ x = u.ex z) (j : Fin 1 → L.ι) (hj : ⊤ ≤ p.left ⁻¹ᵁ LB.wchart L (fun _ : Fin 1 => x.left) j) : valAt' p.left (LB.wchart L (fun _ : Fin 1 => x.left) j) ((T.left.presheaf.map (homOfLE (val_le A L x j)).op).hom (((ptSecT A x).left.app _).hom (s.f j))) = θ z ∗ u.fr (j 0) z ^ k

theorem **IsFnAt.value_mul_frame** {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) {x : T ⟶ A} {z : Fin g → ℂ} (hx : p ≫ x = u.ex z) {τ : LB.Gen (LB.LinComb L (Y

theorem **IsFnAt.isValueAt_of_isIso** [IsIso p] {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) {x : T ⟶ A} {z : Fin g → ℂ} (hx : p ≫ x = u.ex z) {τ : LB.Gen (LB.LinComb L (Y

theorem **comp_one_eq_ex_zero** : p ≫ (1 : T ⟶ A) = u.ex 0

theorem **valAt'_rigT** {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) (k : ℤ) (j : Fin 1 → L.ι) (hj : ⊤ ≤ (u.ex 0).left ⁻¹ᵁ L.U (j 0)) : valAt' p.left (LB.wchart L (fun _ : Fin 1 => (1 : T ⟶ A).left) j) ((rigT A L rig T k).s j).val = u.fr (j 0) 0 ^ k

theorem **IsFnAt.value_origin** (hL : L.IsLB) {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) {v : Γ(T.left, ⊤)} (hv : IsValueAt A L k s 1 (rigT A L rig T k) v) : valAt' p.left ⊤ v = θ 0

theorem **IsFnAt.isValueAt_origin_iff** [IsIso p] (hL : L.IsLB) {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) {k : ℤ} {s : LB.Sec (LPow A L k T)} {θ : (Fin g → ℂ) → ℂ} (h : IsFnAt u p k s θ) (v : Γ(T.left, ⊤)) : IsValueAt A L k s 1 (rigT A L rig T k) v ↔ valAt' p.left ⊤ v = θ 0

theorem **IsFnAt.twistPull** {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} {hm : ψ ≫ snd T A = m} {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **isFnAt_iff_of_isPullSec** {T' : Over (Sp ℂ)} {p' : 𝟙_ (Over (Sp ℂ)) ⟶ T'} {h : T' ⊗ A ⟶ T ⊗ A} {hh : h ≫ snd T A = snd T' A} {k : ℤ} {s : LB.Sec (LPow A L k T)} {s' : LB.Sec (LPow A L k T')} (hs : IsPullSec A L k h hh s s') (hp : ∀ z, exAt u p' z ≫ h.left = exAt u p z) (θ : (Fin g → ℂ) → ℂ) : IsFnAt u p' k s' θ ↔ IsFnAt u p k s θ

theorem **basePt_left** : basePt.left = 𝟙 (Sp ℂ)

theorem **isFnT_iff** (k : ℤ) (s : LB.Sec (LPow A L k (baseOver (Sp ℂ)))) (θ : (Fin g → ℂ) → ℂ) : IsFnT u k s θ ↔ ∀ (j : Fin 1 → L.ι) z, ⊤ ≤ exT u z ⁻¹ᵁ (LPow A L k (baseOver (Sp ℂ))).U j → valAt' (exT u z) ((LPow A L k (baseOver (Sp ℂ))).U j) (s.f j) = θ z ∗ u.fr (j 0) z ^ k

theorem **exUnit_eq_exT** (z : Fin g → ℂ) : exUnit u z = exT u z

theorem **isFnUnit_iff_isFnT** (k : ℤ) (s : LB.Sec (LPow A L k (𝟙_ (Over (Sp ℂ))))) (θ : (Fin g → ℂ) → ℂ) : IsFnUnit u k s θ ↔ IsFnT u k s θ

theorem **exT_snd** (z : Fin g → ℂ) : exT u z ≫ (snd (baseOver (Sp ℂ)) A).left = (u.ex z).left

theorem **eq_exT_of_snd** {z : Fin g → ℂ} {y : Sp ℂ ⟶ (baseOver (Sp ℂ) ⊗ A).left} (hy : y ≫ (snd (baseOver (Sp ℂ)) A).left = (u.ex z).left) : y = exT u z

theorem **numT_constT** (a : ℂ) : numT (constT a) = a

theorem **constT_numT** (r : Γ((baseOver (Sp ℂ)).left, ⊤)) : constT (numT r) = r

theorem **valAt'_basePt** (r : Γ((baseOver (Sp ℂ)).left, ⊤)) : valAt' basePt.left ⊤ r = numT r

theorem **existsUnique_isFnT** (hL : L.IsLB) (k : ℤ) (s : LB.Sec (LPow A L k (baseOver (Sp ℂ)))) : ∃! θ, IsFnT u k s θ

theorem **IsFnT.differentiable** (hL : L.IsLB) {k : ℤ} {s : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u k s θ) : Differentiable ℂ θ

theorem **IsFnT.add_period** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {k : ℤ} {s : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u k s θ) (l : u.Λ) (z : Fin g → ℂ) : θ (z + l) = (χ l ∗ Complex.exp (Real.pi ∗ hermForm Hm z l + Real.pi / 2 ∗ hermForm Hm l l)) ^ k ∗ θ z

theorem **IsFnT.isThetaFn** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {n : ℕ} {s : LB.Sec (LPow A L n (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u n s θ) : u.IsThetaFn Hm χ n θ

theorem **IsFnT.add** {k : ℤ} {s s' t : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ θ' : (Fin g → ℂ) → ℂ} (h : IsFnT u k s θ) (h' : IsFnT u k s' θ') (ht : ∀ j, t.f j = s.f j + s'.f j) : IsFnT u k t (θ + θ')

theorem **IsFnT.liftFun_mul** {k : ℤ} {s t : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u k s θ) (r : Γ((baseOver (Sp ℂ)).left, ⊤)) (ht : ∀ j, t.f j = liftFun A _ r ∗ s.f j) : IsFnT u k t (numT r • θ)

theorem **IsFnT.sum_liftFun_mul** {k : ℤ} {I : Type∗} (F : Finset I) {s : I → LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {t : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ : I → (Fin g → ℂ) → ℂ} (h : ∀ i ∈ F, IsFnT u k (s i) (θ i)) (r : I → Γ((baseOver (Sp ℂ)).left, ⊤)) (ht : ∀ j, t.f j = ∑ i ∈ F, liftFun A _ (r i) ∗ (s i).f j) : IsFnT u k t (∑ i ∈ F, numT (r i) • θ i)

theorem **IsFnT.isValueAt_origin_iff** (hL : L.IsLB) {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) {k : ℤ} {s : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u k s θ) (v : Γ((baseOver (Sp ℂ)).left, ⊤)) : IsValueAt A L k s 1 (rigT A L rig (baseOver (Sp ℂ)) k) v ↔ numT v = θ 0

theorem **IsFnT.isValueAt_origin** (hL : L.IsLB) {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) {k : ℤ} {s : LB.Sec (LPow A L k (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u k s θ) : IsValueAt A L k s 1 (rigT A L rig (baseOver (Sp ℂ)) k) (constT (θ 0))

theorem **isFnT_iff_isFnPow** (n : ℕ) (s : LB.Sec (LPow A L n (baseOver (Sp ℂ)))) (θ : (Fin g → ℂ) → ℂ) : IsFnT u n s θ ↔ u.IsFnPow n (fibreEquiv L basePt n s) θ

theorem **existsUnique_sec_isFnT** {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} {n : ℕ} (hsec : ∀ θ, u.IsThetaFn Hm χ n θ → ∃! s, u.IsFnPow n s θ) {θ : (Fin g → ℂ) → ℂ} (hθ : u.IsThetaFn Hm χ n θ) : ∃! s : LB.Sec (LPow A L n (baseOver (Sp ℂ))), IsFnT u n s θ

theorem **IsFnT.sec_eq** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {n : ℕ} (hsec : ∀ θ, u.IsThetaFn Hm χ n θ → ∃! s, u.IsFnPow n s θ) {s s' : LB.Sec (LPow A L n (baseOver (Sp ℂ)))} {θ : (Fin g → ℂ) → ℂ} (h : IsFnT u n s θ) (h' : IsFnT u n s' θ) : s = s'

theorem **isFnT_zero** (k : ℤ) : IsFnT u k (H1.SectionArithmetic.secZero (LPow A L k (baseOver (Sp ℂ)))) 0

theorem **isFnT_const** (a : ℂ) : IsFnT u 0 (secOfFun A L (constT a)) (fun _ => a)

theorem **zeroUnif_isAH** : zeroUnif.IsAH 0 (fun _ => 1)

theorem **exists_unif_isFnT** (h : ThetaUnifAV) (P : PolAV (Sp ℂ)) (rig : LB.Gen (P.L.pull (eSec P.A))) (hrig : rig.IsGen) : ∃ (u : HolUnif P.A P.L P.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ), u.IsAH Hm χ ∧ u.IsRig rig ∧ ∀ n : ℕ, (∀ s : LB.Sec (LPow P.A P.L n (baseOver (Sp ℂ))), ∃! θ, u.IsThetaFn Hm χ n θ ∧ IsFnT u n s θ) ∧ ∀ θ, u.IsThetaFn Hm χ n θ → ∃! s : LB.Sec (LPow P.A P.L n (baseOver (Sp ℂ))), IsFnT u n s θ

Development.AbelianVarietyTheory.Definitions.ThetaValues.FlatMemLatIffStdLat

@[simp] theorem **flat_apply** (v : Fin n → EuclideanSpace ℂ ι) (p : Fin n × ι) : flat v p = v p.1 p.2

@[simp] theorem **flat_symm_apply** (w : EuclideanSpace ℂ (Fin n × ι)) (k : Fin n) (i : ι) : (flat (n

theorem **Hf_flat** (v w : Fin n → EuclideanSpace ℂ ι) : Hf (flat v) (flat w) = HC v w

theorem **HC_flat_symm** (v w : EuclideanSpace ℂ (Fin n × ι)) : HC ((flat (n

theorem **norm_sq_flat** (v : Fin n → EuclideanSpace ℂ ι) : ‖flat v‖ ^ 2 = ∑ k, ‖v k‖ ^ 2

theorem **HC_self_re** (v : Fin n → EuclideanSpace ℂ ι) : (HC v v).re = ‖flat v‖ ^ 2

theorem **flat_single** [DecidableEq ι] (k : Fin n) (i : ι) (c : ℂ) : flat (Pi.single k (EuclideanSpace.single i c) : Fin n → EuclideanSpace ℂ ι) = EuclideanSpace.single (k, i) c

theorem **HC_conj** (v w : Fin n → EuclideanSpace ℂ ι) : HC w v = (starRingEnd ℂ) (HC v w)

theorem **Hf_sum_left** {κ : Type*} (s : Finset κ) (v : κ → EuclideanSpace ℂ ι) (w : EuclideanSpace ℂ ι) : Hf (∑ j ∈ s, v j) w = ∑ j ∈ s, Hf (v j) w

theorem **Hf_sum_right** {κ : Type*} (s : Finset κ) (v : EuclideanSpace ℂ ι) (w : κ → EuclideanSpace ℂ ι) : Hf v (∑ j ∈ s, w j) = ∑ j ∈ s, Hf v (w j)

theorem **HC_add_left** (v v' w : Fin n → EuclideanSpace ℂ ι) : HC (v + v') w = HC v w + HC v' w

theorem **HC_add_right** (v w w' : Fin n → EuclideanSpace ℂ ι) : HC v (w + w') = HC v w + HC v w'

theorem **HC_sum_left** {κ : Type*} (s : Finset κ) (v : κ → Fin n → EuclideanSpace ℂ ι) (w : Fin n → EuclideanSpace ℂ ι) : HC (∑ j ∈ s, v j) w = ∑ j ∈ s, HC (v j) w

theorem **HC_sum_right** {κ : Type*} (s : Finset κ) (v : Fin n → EuclideanSpace ℂ ι) (w : κ → Fin n → EuclideanSpace ℂ ι) : HC v (∑ j ∈ s, w j) = ∑ j ∈ s, HC v (w j)

theorem **flat_dir** (l : Letter n dim) : flat (dir l) = EuclideanSpace.single l 1

theorem **flat_eq_eT** (X : PT E) (v : Fin n → Vsp X.dim) : flat v = TiledBoundaries.eT X n v

theorem **flat_symm_eq_eT_symm** (X : PT E) (w : EuclideanSpace ℂ (Fin n × Idx X.dim)) : (flat (n

@[simp] theorem **actL_apply** (α : E) (v : Vsp dim) : actL α v = actE α v

theorem **actE_coe** (α : 𝒪 E) (v : Vsp dim) : actE (α : E) v = act α v

theorem **actE_sum_left** {κ : Type*} (s : Finset κ) (α : κ → E) (v : Vsp dim) : actE (∑ j ∈ s, α j) v = ∑ j ∈ s, actE (α j) v

theorem **actE_sum_right** {κ : Type*} (s : Finset κ) (α : E) (v : κ → Vsp dim) : actE α (∑ j ∈ s, v j) = ∑ j ∈ s, actE α (v j)

theorem **actE_neg_left** (α : E) (v : Vsp dim) : actE (-α) v = -actE α v

theorem **actE_neg_right** (α : E) (v : Vsp dim) : actE α (-v) = -actE α v

theorem **actE_comm** (α β : E) (v : Vsp dim) : actE α (actE β v) = actE β (actE α v)

theorem **actE_intCast** (m : ℤ) (v : Vsp dim) : actE (m : E) v = m • v

theorem **actE_natCast** (m : ℕ) (v : Vsp dim) : actE (m : E) v = m • v

theorem **actE_mem_of_isIntegral** (X : PT E) {α : E} (hα : IsIntegral ℤ α) {x : Vsp X.dim} (hx : x ∈ X.L) : actE α x ∈ X.L

@[simp] theorem **tens_apply** (x : Vsp dim) (v : Fin n → E) (k : Fin n) : tens x v k = actE (v k) x

theorem **tens_add_left** (x y : Vsp dim) (v : Fin n → E) : tens (x + y) v = tens x v + tens y v

theorem **tens_add_right** (x : Vsp dim) (v w : Fin n → E) : tens x (v + w) = tens x v + tens x w

@[simp] theorem **tens_zero_left** (v : Fin n → E) : tens (0 : Vsp dim) v = 0

@[simp] theorem **tens_zero_right** (x : Vsp dim) : tens x (0 : Fin n → E) = 0

theorem **tens_smul_left** (c : ℂ) (x : Vsp dim) (v : Fin n → E) : tens (c • x) v = c • tens x v

theorem **tens_smul_right** (α : E) (x : Vsp dim) (v : Fin n → E) : tens x (α • v) = tens (actE α x) v

theorem **actE_tens** (α : E) (x : Vsp dim) (v : Fin n → E) : (fun k => actE α (tens x v k)) = tens (actE α x) v

theorem **tens_sum_left** {κ : Type*} (s : Finset κ) (x : κ → Vsp dim) (v : Fin n → E) : tens (∑ j ∈ s, x j) v = ∑ j ∈ s, tens (x j) v

theorem **tens_sum_right** {κ : Type*} (s : Finset κ) (x : Vsp dim) (v : κ → Fin n → E) : tens x (∑ j ∈ s, v j) = ∑ j ∈ s, tens x (v j)

theorem **tens_single** (x : Vsp dim) (k : Fin n) : tens x (Pi.single k (1 : E)) = Pi.single k x

theorem **sum_tens_single** (γ : Fin n → Vsp dim) : ∑ k, tens (γ k) (Pi.single k (1 : E)) = γ

theorem **norm_tensorEps** (w₀ : 𝒪 E) (γ : Fin n → Vsp dim) : ‖tensorEps w₀ γ‖ = 1

@[simp] theorem **tensorEps_zero** (w₀ : 𝒪 E) : tensorEps w₀ (0 : Fin n → Vsp dim) = 1

theorem **tensorEps_neg** (w₀ : 𝒪 E) (γ : Fin n → Vsp dim) : tensorEps w₀ (-γ) = tensorEps w₀ γ

theorem **tensorEps_of_int** (w₀ : 𝒪 E) (γ : Fin n → Vsp dim) (m : ℤ) (h : (HC γ fun k => act w₀ (γ k)).im = m) : tensorEps w₀ γ = (-1) ^ m

theorem **tensorEps_eq_prod** (w₀ : 𝒪 E) (γ : Fin n → Vsp dim) : tensorEps w₀ γ = ∏ k, cexp (Real.pi * I * ((Hf (γ k) (act w₀ (γ k))).im : ℂ))

theorem **tensorEps_eq_prod_eps** (X : PT E) (w₀ : 𝒪 E) (heps : ∀ x ∈ X.L, (X.eps x : ℝ) = Real.cos (Real.pi * (Hf x (act w₀ x)).im)) {γ : Fin n → Vsp X.dim} (hγ : ∀ k, γ k ∈ X.L) : tensorEps w₀ γ = ∏ k, (X.eps (γ k) : ℂ)

theorem **Hf_actE_cc** (α : E) (v w : Vsp dim) : Hf (actE α v) w = Hf v (actE (cc α) w)

theorem **Hf_actE_cc'** (α : E) (v w : Vsp dim) : Hf v (actE α w) = Hf (actE (cc α) v) w

theorem **HC_actE** (α : E) (γ γ' : Fin n → Vsp dim) : HC (fun k => actE α (γ k)) γ' = HC γ fun k => actE (cc α) (γ' k)

theorem **contr_apply** (γ : Fin n → Vsp dim) (w : Fin n → E) (i : Idx dim) : contr γ w i = ∑ k, (starRingEnd ℂ) (i.1 (w k)) * γ k i

theorem **contr_add_left** (γ γ' : Fin n → Vsp dim) (w : Fin n → E) : contr (γ + γ') w = contr γ w + contr γ' w

theorem **contr_add_right** (γ : Fin n → Vsp dim) (w w' : Fin n → E) : contr γ (w + w') = contr γ w + contr γ w'

@[simp] theorem **contr_zero_left** (w : Fin n → E) : contr (0 : Fin n → Vsp dim) w = 0

@[simp] theorem **contr_zero_right** (γ : Fin n → Vsp dim) : contr γ (0 : Fin n → E) = 0

theorem **contr_smul_left** (c : ℂ) (γ : Fin n → Vsp dim) (w : Fin n → E) : contr (c • γ) w = c • contr γ w

theorem **contr_zsmul_left** (m : ℤ) (γ : Fin n → Vsp dim) (w : Fin n → E) : contr (m • γ) w = m • contr γ w

theorem **contr_actE_left** (α : E) (γ : Fin n → Vsp dim) (w : Fin n → E) : contr (fun k => actE α (γ k)) w = actE α (contr γ w)

theorem **contr_smul_right** (α : E) (γ : Fin n → Vsp dim) (w : Fin n → E) : contr γ (α • w) = actE (cc α) (contr γ w)

theorem **contr_sum_left** {κ : Type*} (s : Finset κ) (γ : κ → Fin n → Vsp dim) (w : Fin n → E) : contr (∑ j ∈ s, γ j) w = ∑ j ∈ s, contr (γ j) w

theorem **contr_sum_right** {κ : Type*} (s : Finset κ) (γ : Fin n → Vsp dim) (w : κ → Fin n → E) : contr γ (∑ j ∈ s, w j) = ∑ j ∈ s, contr γ (w j)

theorem **contr_single'** (γ : Fin n → Vsp dim) (k : Fin n) (α : E) : contr γ (Pi.single k α) = actE (cc α) (γ k)

@[simp] theorem **contr_single** (γ : Fin n → Vsp dim) (k : Fin n) : contr γ (Pi.single k (1 : E)) = γ k

theorem **contr_tens** (x : Vsp dim) (v w : Fin n → E) : contr (tens x v) w = actE (herm 1 v w) x

theorem **HC_tens_right** (γ : Fin n → Vsp dim) (x : Vsp dim) (w : Fin n → E) : HC γ (tens x w) = Hf (contr γ w) x

theorem **HC_tens_left** (γ : Fin n → Vsp dim) (x : Vsp dim) (w : Fin n → E) : HC (tens x w) γ = Hf x (contr γ w)

theorem **HC_tens_tens** (x y : Vsp dim) (v w : Fin n → E) : HC (tens x v) (tens y w) = Hf (actE (herm 1 v w) x) y

theorem **eq_zero_of_contr_span** {S : Set (Fin n → E)} (hS : Submodule.span E S = ⊤) {γ : Fin n → Vsp dim} (h : ∀ w ∈ S, contr γ w = 0) : γ = 0

theorem **mem_tensorLat** {γ : Fin n → Vsp X.dim} : γ ∈ tensorLat X Λ' ↔ ∀ w ∈ Λ', contr γ w ∈ X.L

theorem **tensorLat_anti** {Λ₁ Λ₂ : Submodule (𝒪 E) (Fin n → E)} (h : Λ₁ ≤ Λ₂) : tensorLat X Λ₂ ≤ tensorLat X Λ₁

theorem **act_mem_tensorLat** (α : 𝒪 E) {γ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') : (fun k => act α (γ k)) ∈ tensorLat X Λ'

theorem **tens_mem_tensorLat** {x : Vsp X.dim} (hx : x ∈ X.L) {v : Fin n → E} (hv : ∀ w ∈ Λ', IsIntegral ℤ (herm 1 v w)) : tens x v ∈ tensorLat X Λ'

theorem **im_HC_tens_int** {γ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') {x : Vsp X.dim} (hx : x ∈ X.L) {w : Fin n → E} (hw : w ∈ Λ') : ∃ m : ℤ, (HC γ (tens x w)).im = m

theorem **mem_tensorLat_span** {S : Set (Fin n → E)} {γ : Fin n → Vsp X.dim} : γ ∈ tensorLat X (Submodule.span (𝒪 E) S) ↔ ∀ w ∈ S, contr γ w ∈ X.L

theorem **mem_stdLat** {v : Fin n → E} : v ∈ stdLat E n ↔ ∀ k, IsIntegral ℤ (v k)

theorem **stdLat_fg** : (stdLat E n).FG

theorem **mem_stdLat_iff_herm** {v : Fin n → E} : v ∈ stdLat E n ↔ ∀ w ∈ stdLat E n, IsIntegral ℤ (herm 1 v w)

theorem **mem_tensorLat_stdLat** {γ : Fin n → Vsp X.dim} : γ ∈ tensorLat X (stdLat E n) ↔ ∀ k, γ k ∈ X.L

theorem **stdLat_eq_pi** : stdLat E n = Submodule.pi Set.univ fun _ => (1 : Submodule (𝒪 E) E)

theorem **mem_tensorLat_pi** {γ : Fin n → Vsp X.dim} : γ ∈ tensorLat X (Submodule.pi Set.univ fun _ => (1 : Submodule (𝒪 E) E)) ↔ ∀ k, γ k ∈ X.L

theorem **flat_mem_lat_iff_stdLat** {γ : Fin n → Vsp X.dim} : flat γ ∈ lat (TiledBoundaries.bT X n) ↔ γ ∈ tensorLat X (stdLat E n)

Development.AbelianVarietyTheory.Definitions.ThetaValues.GalRulGalRul

theorem **allTest** : AllTest Xo c

theorem **valueWD** : ValueWD Xo c

theorem **galRul_one** {ta : c.F ≃+* c.F} (h : Fixes c ta) : galRul (Xo

theorem **galRul_mul** {ta : c.F ≃+* c.F} (h : Fixes c ta) (f g : Rul Xo c) : galRul h (f * g) = galRul h f * galRul h g

theorem **galRul_const** {ta : c.F ≃+* c.F} (h : Fixes c ta) (a : c.F) : galRul h (a • (1 : Rul Xo c)) = ta a • (1 : Rul Xo c)

theorem **fixes_trans** {s t : c.F ≃+* c.F} (hs : Fixes c s) (ht : Fixes c t) : Fixes c (s.trans t)

theorem **galRul_galRul** {s t : c.F ≃+* c.F} (hs : Fixes c s) (ht : Fixes c t) {f : Rul Xo c} (hf : Natural c f) : galRul ht (galRul hs f) = galRul (fixes_trans hs ht) f

theorem **exists_moved** {ta : c.F ≃+* c.F} (h : Fixes c ta) (hta : ta ≠ RingEquiv.refl _) (p : TRing c.F) [Nontrivial p.R] (y : FrPt Xo c p.R) : ∃ f : Rul Xo c, galRul h f ≠ f

theorem **galRul_refl** {f : Rul Xo c} (hf : Natural c f) : galRul fixes_refl f = f

Development.AbelianVarietyTheory.Definitions.ThetaValues.IoOverLeftUnique

theorem **alg_eq** (hle : c.F ≤ c'.F) (S : Type) [CommRing S] [i : Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) : i = resAlg hle S

theorem **nabBC_apply** (hle : c.F ≤ c'.F) (nab : NabV Xo c) {S : Type} [CommRing S] [i : Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S) (k : ℤ) (F) (α) : nabBC hle nab y k F α = nab (y.res hle h) k F α

theorem **nabBC_self** (nab : NabV Xo c) : (nabBC (le_refl c.F) nab : NabV Xo c) = (nab : NabV Xo c)

theorem **rulBC_apply** (hle : c.F ≤ c'.F) (f : Rul Xo c) (p : TRing c'.F) [i : Algebra c.F p.R] (h : AlgOver c c' hle p.R) (y : FrPt Xo c' p.R) : rulBC hle f p y = evR Xo c f (y.res hle h)

theorem **wd** (c : Coeff l) : ValueWD Xo c

theorem **fromT_res** (hle : c.F ≤ c'.F) (p : TRing c'.F) [Algebra c.F p.R] (h : AlgOver c c' hle p.R) (y : FrPt Xo c' p.R) : FromT (y.res hle h)

theorem **rulBC_one** (hle : c.F ≤ c'.F) : rulBC hle (1 : Rul Xo c) = 1

theorem **rulBC_self** {f : Rul Xo c} (hf : Natural c f) : rulBC (le_refl c.F) f = f

theorem **rulBC_natural** (hle : c.F ≤ c'.F) {f : Rul Xo c} (hf : Natural c f) : Natural c' (rulBC hle f)

theorem **evR_rulBC** (hle : c.F ≤ c'.F) {f : Rul Xo c} (hf : Natural c f) {S : Type} [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S) : evR Xo c' (rulBC hle f) y = evR Xo c f (y.res hle h)

theorem **isBC_rulBC** (hle : c.F ≤ c'.F) {f : Rul Xo c} (hf : Natural c f) : IsBC hle (rulBC hle f) ![1] ![f]

theorem **rulBC_mem_sec** (hle : c.F ≤ c'.F) {a} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) : rulBC hle f ∈ sec Xo c' a n

theorem **frOver_iff** (hle : c.F ≤ c'.F) {n : ℕ} (d : DeltaDat Xo c n) {k : Type} [CommRing k] [Algebra c.F k] [Algebra d.c₂.F k] [Algebra c'.F k] [i₂ : Algebra (covCoeff hle d).F k] (h : AlgOver c c' hle k) (h₂ : AlgOver d.c₂ (covCoeff hle d) (le_covCoeff hle d) k) (y₂ : FrPt d.Xo₂ (covCoeff hle d) k) (y : FrPt Xo c' k) : @FrOver E _ _ _ H Dw l Xo c' n (ddBC hle d) k _ _ i₂ y₂ y ↔ FrOver d (y₂.res (le_covCoeff hle d) h₂) (y.res hle h)

theorem **w_ddBC** (hle : c.F ≤ c'.F) {n : ℕ} (d : DeltaDat Xo c n) (x0 : Fin n → Dw.La) (q : Fin n → LevQ Dw d.l₂) : (ddBC hle d).w x0 q = incF d.c₂ (covCoeff hle d) (le_covCoeff hle d) (d.w x0 q)

theorem **isLev_ddBC** (hle : c.F ≤ c'.F) {n : ℕ} (d : DeltaDat Xo c n) (hd : d.IsLev) : (ddBC hle d).IsLev

theorem **ioOver_rename** (hle : c.F ≤ c'.F) {n : ℕ} {Ix : Type} [Fintype Ix] (lam : Ix → Formalism.BasicNotions.DomWeight n) (g : Ix × Fin n × Fin n → LtN Dw n) : IoOver hle lam ((MvPolynomial.rename g).toLinearMap ∘ₗ (Formalism.L2.DIV.VV c.F lam).subtype) ((MvPolynomial.rename g).toLinearMap ∘ₗ (Formalism.L2.DIV.VV c'.F lam).subtype)

theorem **ioOver_self** {n : ℕ} {Ix : Type} [Fintype Ix] (lam : Ix → Formalism.BasicNotions.DomWeight n) (io io' : VVc c lam →ₗ[c.F] MvPolynomial (LtN Dw n) c.F) : IoOver (le_refl c.F) lam io io' ↔ io' = io

theorem **map_mem_dualWeyl** {K K' : Type} [CommRing K] [CommRing K'] (φ : K →+* K') {r n : ℕ} (ν : Fin n → ℕ) {f : MvPolynomial (Fin r × Fin n) K} (hf : f ∈ Formalism.GradedWeylDivisibility.dualWeyl K r n ν) : MvPolynomial.map φ f ∈ Formalism.GradedWeylDivisibility.dualWeyl K' r n ν

theorem **exists_over** (hle : c.F ≤ c'.F) {n : ℕ} {Ix : Type} [Fintype Ix] (lam : Ix → Formalism.BasicNotions.DomWeight n) (v : VVc c lam) : ∃ v' : VVc c' lam, (v' : Formalism.L2.DIV.Pol c'.F Ix n) = MvPolynomial.map (incF c c' hle) (v : Formalism.L2.DIV.Pol c.F Ix n)

theorem **ioOver_left_unique** (hle : c.F ≤ c'.F) {n : ℕ} {Ix : Type} [Fintype Ix] (lam : Ix → Formalism.BasicNotions.DomWeight n) {io₁ io₂ : VVc c lam →ₗ[c.F] MvPolynomial (LtN Dw n) c.F} {io' : VVc c' lam →ₗ[c'.F] MvPolynomial (LtN Dw n) c'.F} (h₁ : IoOver hle lam io₁ io') (h₂ : IoOver hle lam io₂ io') : io₁ = io₂

theorem **isNab_nabBC** [StarRing (𝒪 E)] (hle : c.F ≤ c'.F) (nab : NabV Xo c) (h : IsNab Xo c nab) : IsNab Xo c' (nabBC hle nab)

Development.AbelianVarietyTheory.Definitions.ThetaValues.LeCovCoeff

theorem **le_covCoeff** (hle : c.F ≤ c'.F) {n : ℕ} (d : DeltaDat Xo c n) : d.c₂.F ≤ (covCoeff hle d).F

Development.AbelianVarietyTheory.Definitions.ThetaValues.MemAnalyticAtZeroOfEventuallyDifferentiableAt

theorem **mvPowerSeries_eq_zero_of_pderiv_mem_span** (S : Set (MvPowerSeries σ R)) (h0 : ∀ F ∈ S, MvPowerSeries.constantCoeff F = 0) (hd : ∀ F ∈ S, ∀ i, MvPowerSeries.pderiv R i F ∈ Ideal.span S) : ∀ F ∈ S, F = 0

theorem **dirPolyDeriv_monomial_one** (m : σ →₀ ℕ) (u : σ → W) (F : W → ℂ) (x : W) : ReciprocityTheory.ChartRestriction.G.I2.polyDeriv (MvPolynomial.monomial m 1) u F x = iteratedFDeriv ℂ (Multiset.card (Finsupp.toMultiset m)) F x (fun k => u ((Finsupp.toMultiset m).toList.get (k.cast (Multiset.length_toList _).symm)))

theorem **dirPolyDeriv_eq_sum** (P : MvPolynomial σ ℂ) (u : σ → W) (F : W → ℂ) (x : W) : ReciprocityTheory.ChartRestriction.G.I2.polyDeriv P u F x = ∑ m ∈ P.support, P.coeff m * ReciprocityTheory.ChartRestriction.G.I2.polyDeriv (MvPolynomial.monomial m 1) u F x

theorem **dirPolyDeriv_monomial_eq_lineDerivList** {m : σ →₀ ℕ} {l : List σ} (hl : (l : Multiset σ) = Finsupp.toMultiset m) (u : σ → W) {F : W → ℂ} {x : W} (hF : AnalyticAt ℂ F x) : ReciprocityTheory.ChartRestriction.G.I2.polyDeriv (MvPolynomial.monomial m 1) u F x = lineDerivList ℂ (l.map u) F x

theorem **coeff_taylorDir** (f : W → ℂ) (m : σ →₀ ℕ) : MvPowerSeries.coeff m (taylorDir u f) = ((∏ i, (m i).factorial : ℕ) : ℂ)⁻¹ * ReciprocityTheory.ChartRestriction.G.I2.polyDeriv (MvPolynomial.monomial m 1) u f 0

theorem **cast_prod_factorial_ne_zero** (m : σ →₀ ℕ) : ((∏ i, (m i).factorial : ℕ) : ℂ) ≠ 0

theorem **dirPolyDeriv_monomial_eq_coeff** (f : W → ℂ) (m : σ →₀ ℕ) : ReciprocityTheory.ChartRestriction.G.I2.polyDeriv (MvPolynomial.monomial m 1) u f 0 = ((∏ i, (m i).factorial : ℕ) : ℂ) * MvPowerSeries.coeff m (taylorDir u f)

theorem **dirPolyDeriv_eq_sum_coeff_taylorDir** (P : MvPolynomial σ ℂ) (f : W → ℂ) : ReciprocityTheory.ChartRestriction.G.I2.polyDeriv P u f 0 = ∑ m ∈ P.support, P.coeff m * ((∏ i, (m i).factorial : ℕ) : ℂ) * MvPowerSeries.coeff m (taylorDir u f)

theorem **coeff_taylorDir_eq_lineDerivList** (hf : AnalyticAt ℂ f 0) {m : σ →₀ ℕ} {l : List σ} (hl : (l : Multiset σ) = Finsupp.toMultiset m) : MvPowerSeries.coeff m (taylorDir u f) = ((∏ i, (m i).factorial : ℕ) : ℂ)⁻¹ * lineDerivList ℂ (l.map u) f 0

theorem **taylorDir_congr** (h : f =ᶠ[𝒩 0] g) : taylorDir u f = taylorDir u g

@[simp] theorem **constantCoeff_taylorDir** (f : W → ℂ) : MvPowerSeries.constantCoeff (taylorDir u f) = f 0

theorem **taylorDir_add** (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylorDir u (f + g) = taylorDir u f + taylorDir u g

theorem **taylorDir_smul** (c : ℂ) (hf : AnalyticAt ℂ f 0) : taylorDir u (c • f) = c • taylorDir u f

theorem **analyticAt_fderiv_apply** {x : W} (hf : AnalyticAt ℂ f x) (v : W) : AnalyticAt ℂ (fun z => fderiv ℂ f z v) x

theorem **taylorDir_fderiv** (hf : AnalyticAt ℂ f 0) (i : σ) : taylorDir u (fun z => fderiv ℂ f z (u i)) = MvPowerSeries.pderiv ℂ i (taylorDir u f)

@[simp] theorem **taylorDir_const** (c : ℂ) : taylorDir u (fun _ => c) = MvPowerSeries.C c

@[simp] theorem **taylorDir_zero** : taylorDir u (0 : W → ℂ) = 0

@[simp] theorem **taylorDir_one** : taylorDir u (1 : W → ℂ) = 1

theorem **taylorDir_mul** (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylorDir u (f * g) = taylorDir u f * taylorDir u g

theorem **taylorDir_clm** (ℓ : W →L[ℂ] ℂ) : taylorDir u ℓ = ∑ i, ℓ (u i) • (MvPowerSeries.X i : MvPowerSeries σ ℂ)

theorem **taylorDir_comp_clm** {W' : Type*} [NormedAddCommGroup W'] [NormedSpace ℂ W'] (L : W' →L[ℂ] W) (v : σ → W') (hf : AnalyticAt ℂ f 0) : taylorDir v (fun z => f (L z)) = taylorDir (fun i => L (v i)) f

theorem **coeff_taylorDir_eq_zero_iff** (hu : Submodule.span ℂ (Set.range u) = ⊤) (hf : AnalyticAt ℂ f 0) (n : ℕ) : (∀ m : σ →₀ ℕ, m.degree = n → MvPowerSeries.coeff m (taylorDir u f) = 0) ↔ iteratedFDeriv ℂ n f 0 = 0

theorem **taylorDir_eq_zero_iff** (hu : Submodule.span ℂ (Set.range u) = ⊤) (hf : AnalyticAt ℂ f 0) : taylorDir u f = 0 ↔ f =ᶠ[𝓝 0] 0

@[simp] theorem **mem_analyticAtZero** : f ∈ analyticAtZero W ↔ AnalyticAt ℂ f 0

@[simp] theorem **taylorDirAlgHom_apply** (f : analyticAtZero W) : taylorDirAlgHom u f = taylorDir u f

theorem **taylorDir_neg** (hf : AnalyticAt ℂ f 0) : taylorDir u (-f) = -taylorDir u f

theorem **taylorDir_sub** (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylorDir u (f - g) = taylorDir u f - taylorDir u g

theorem **taylorDir_sum** {κ : Type*} (s : Finset κ) {F : κ → W → ℂ} (hF : ∀ k ∈ s, AnalyticAt ℂ (F k) 0) : taylorDir u (fun z => ∑ k ∈ s, F k z) = ∑ k ∈ s, taylorDir u (F k)

theorem **taylorDir_prod** {κ : Type*} (s : Finset κ) {F : κ → W → ℂ} (hF : ∀ k ∈ s, AnalyticAt ℂ (F k) 0) : taylorDir u (fun z => ∏ k ∈ s, F k z) = ∏ k ∈ s, taylorDir u (F k)

theorem **taylorDir_eval** {κ : Type*} {φ : κ → W → ℂ} (hφ : ∀ k, AnalyticAt ℂ (φ k) 0) (P : MvPolynomial κ ℂ) : taylorDir u (fun z => MvPolynomial.eval (fun k => φ k z) P) = MvPolynomial.aeval (fun k => taylorDir u (φ k)) P

theorem **taylorDir_eq_iff** (hu : Submodule.span ℂ (Set.range u) = ⊤) (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylorDir u f = taylorDir u g ↔ f =ᶠ[𝓝 0] g

theorem **taylorDir_fderiv_sum** (hf : AnalyticAt ℂ f 0) (a : σ → ℂ) : taylorDir u (fun z => fderiv ℂ f z (∑ i, a i • u i)) = ∑ i, a i • MvPowerSeries.pderiv ℂ i (taylorDir u f)

theorem **taylor_eq_taylorDir** (f : (σ → ℂ) → ℂ) : taylor f = taylorDir (fun i => (Pi.single i 1 : σ → ℂ)) f

theorem **coeff_taylor** (f : (σ → ℂ) → ℂ) (m : σ →₀ ℕ) : MvPowerSeries.coeff m (taylor f) = ((∏ i, (m i).factorial : ℕ) : ℂ)⁻¹ * ReciprocityTheory.ChartRestriction.G.I2.polyDeriv (MvPolynomial.monomial m 1) (fun i => (Pi.single i 1 : σ → ℂ)) f 0

theorem **dirPolyDeriv_eq_sum_coeff_taylor** (P : MvPolynomial σ ℂ) (f : (σ → ℂ) → ℂ) : ReciprocityTheory.ChartRestriction.G.I2.polyDeriv P (fun i => (Pi.single i 1 : σ → ℂ)) f 0 = ∑ m ∈ P.support, P.coeff m * ((∏ i, (m i).factorial : ℕ) : ℂ) * MvPowerSeries.coeff m (taylor f)

@[simp] theorem **constantCoeff_taylor** (f : (σ → ℂ) → ℂ) : MvPowerSeries.constantCoeff (taylor f) = f 0

theorem **taylor_add** (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylor (f + g) = taylor f + taylor g

@[simp] theorem **taylor_const** (c : ℂ) : taylor (fun _ : σ → ℂ => c) = MvPowerSeries.C c

@[simp] theorem **taylor_zero** : taylor (0 : (σ → ℂ) → ℂ) = 0

@[simp] theorem **taylor_one** : taylor (1 : (σ → ℂ) → ℂ) = 1

theorem **taylor_mul** (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylor (f * g) = taylor f * taylor g

theorem **taylor_sum** {κ : Type*} (s : Finset κ) {F : κ → (σ → ℂ) → ℂ} (hF : ∀ k ∈ s, AnalyticAt ℂ (F k) 0) : taylor (fun z => ∑ k ∈ s, F k z) = ∑ k ∈ s, taylor (F k)

theorem **taylor_fderiv** (hf : AnalyticAt ℂ f 0) (i : σ) : taylor (fun z => fderiv ℂ f z (Pi.single i 1)) = MvPowerSeries.pderiv ℂ i (taylor f)

theorem **taylor_clm** (ℓ : (σ → ℂ) →L[ℂ] ℂ) : taylor ℓ = ∑ i, ℓ (Pi.single i 1) • (MvPowerSeries.X i : MvPowerSeries σ ℂ)

@[simp] theorem **taylor_coord** (i : σ) : taylor (fun z : σ → ℂ => z i) = MvPowerSeries.X i

theorem **span_range_single_one** : Submodule.span ℂ (Set.range fun i : σ => (Pi.single i 1 : σ → ℂ)) = ⊤

theorem **taylor_eq_iff** (hf : AnalyticAt ℂ f 0) (hg : AnalyticAt ℂ g 0) : taylor f = taylor g ↔ f =ᶠ[𝓝 0] g

@[simp] theorem **taylorAlgHom_apply** (f : analyticAtZero (σ → ℂ)) : taylorAlgHom f = taylor f

theorem **taylor_eval_comp** {κ : Type*} {φ : κ → (σ → ℂ) → ℂ} (hφ : ∀ k, AnalyticAt ℂ (φ k) 0) (P : MvPolynomial κ ℂ) : taylor (fun z => MvPolynomial.eval (fun k => φ k z) P) = MvPolynomial.aeval (fun k => taylor (φ k)) P

theorem **taylor_eval** (P : MvPolynomial σ ℂ) : taylor (fun z : σ → ℂ => MvPolynomial.eval z P) = (P : MvPowerSeries σ ℂ)

theorem **taylor_comp_clm** (L : (σ → ℂ) →L[ℂ] W) {F : W → ℂ} (hF : AnalyticAt ℂ F 0) : taylor (fun z => F (L z)) = taylorDir (fun i => L (Pi.single i 1)) F

theorem **taylorDir_eq_taylor** (u : σ → W) {F : W → ℂ} (hF : AnalyticAt ℂ F 0) : taylorDir u F = taylor fun z : σ → ℂ => F (∑ i, z i • u i)

Development.AbelianVarietyTheory.Definitions.ThetaValues.PolyDerivSmulLetter

theorem **per_apply** (x : Frame X.Ψ r) (v : Fin r → E) (p : X.Ψ × Fin r) : per X x v p = if (p.2 : ℕ) < X.p0 then hermC p.1.1 X.Gm (FD.col (mat x p.1) p.2) (Data.emb p.1.1 v) else ((mat x p.1)⁻¹ *ᵥ Data.emb p.1.1 v) p.2

theorem **per_of_lt** (x : Frame X.Ψ r) (v : Fin r → E) (τ : X.Ψ) {a : Fin r} (h : (a : ℕ) < X.p0) : per X x v (τ, a) = hermC τ.1 X.Gm (FD.col (mat x τ) a) (Data.emb τ.1 v)

theorem **per_of_le** (x : Frame X.Ψ r) (v : Fin r → E) (τ : X.Ψ) {a : Fin r} (h : X.p0 ≤ (a : ℕ)) : per X x v (τ, a) = ((mat x τ)⁻¹ *ᵥ Data.emb τ.1 v) a

theorem **rowType_of_lt** (τ : X.Ψ) {a : Fin r} (h : (a : ℕ) < X.p0) : rowType X (τ, a) = ComplexEmbedding.conjugate τ.1

theorem **rowType_of_le** (τ : X.Ψ) {a : Fin r} (h : X.p0 ≤ (a : ℕ)) : rowType X (τ, a) = τ.1

theorem **per_add** (x : Frame X.Ψ r) (u v : Fin r → E) : per X x (u + v) = per X x u + per X x v

theorem **per_smul** (x : Frame X.Ψ r) (α : E) (v : Fin r → E) (p : X.Ψ × Fin r) : per X x (α • v) p = rowType X p α * per X x v p

theorem **per_ratSmul** (x : Frame X.Ψ r) (q : ℚ) (v : Fin r → E) : per X x (q • v) = q • per X x v

@[simp] theorem **perLin_apply** (x : Frame X.Ψ r) (v : Fin r → E) : perLin X x v = per X x v

@[simp] theorem **per_zero** (x : Frame X.Ψ r) : per X x 0 = 0

theorem **per_neg** (x : Frame X.Ψ r) (v : Fin r → E) : per X x (-v) = -per X x v

theorem **per_zsmul** (x : Frame X.Ψ r) (m : ℤ) (v : Fin r → E) : per X x (m • v) = m • per X x v

theorem **per_sum** {ι : Type*} (x : Frame X.Ψ r) (s : Finset ι) (v : ι → Fin r → E) : per X x (∑ i ∈ s, v i) = ∑ i ∈ s, per X x (v i)

theorem **EF_def** (u v : Fin r → E) : EF X u v = Algebra.trace ℚ E (-(X.κ0 / 2) • herm X.Gm u v)

theorem **im_neg_half_pos** (τ : X.Ψ) : 0 < (τ.1 (-(X.κ0 / 2))).im

theorem **EF_eq_or** : (∀ u v, EF X u v = EFf X.ξ0 X.Gm u v) ∨ ∀ u v, EF X u v = -EFf X.ξ0 X.Gm u v

@[simp] theorem **perMat_apply** (s : Fin g → Fin g ⊕ Fin g) (x : Frame X.Ψ r) (i k : Fin g) : perMat X bas idx s x i k = per X x (bas (Pi.single (s k) 1)) (idx i)

theorem **perMat_mulVec_single** (s : Fin g → Fin g ⊕ Fin g) (x : Frame X.Ψ r) (k : Fin g) : perMat X bas idx s x *ᵥ Pi.single k 1 = per X x (bas (Pi.single (s k) 1)) ∘ idx

theorem **imInv_eq** (x : Frame X.Ψ r) : imInv X bas idx x = ((tauMat X bas idx x).map fun z => (z.im : ℂ))⁻¹

theorem **per_bas** (x : Frame X.Ψ r) (m : Fin g ⊕ Fin g → ℤ) : per X x (bas m) ∘ idx = perMat X bas idx Sum.inl x *ᵥ (fun k => (m (Sum.inl k) : ℂ)) + perMat X bas idx Sum.inr x *ᵥ fun k => (m (Sum.inr k) : ℂ)

theorem **hMat_mulVec_per** {x : Frame X.Ψ r} (h : IsUnit (perMat X bas idx Sum.inl x).det) (m : Fin g ⊕ Fin g → ℤ) : hMat X bas idx x *ᵥ (per X x (bas m) ∘ idx) = (fun k => (m (Sum.inl k) : ℂ)) + tauMat X bas idx x *ᵥ fun k => (m (Sum.inr k) : ℂ)

theorem **canFn_zero** (a b : Fin g → ℝ) (x : Frame X.Ψ r) : canFn X bas idx a b x 0 = thetaChar (tauMat X bas idx x) (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) 0

@[simp] theorem **rowIdx_inl** {p q r : ℕ} (hr : p + q = r) (i : Fin q) : (rowIdx hr (Sum.inl i) : ℕ) = p + i

@[simp] theorem **rowIdx_inr** {p q r : ℕ} (hr : p + q = r) (i : Fin p) : (rowIdx hr (Sum.inr i) : ℕ) = p - 1 - i

@[simp] theorem **rowOf_apply** (τ : X.Ψ) (ρ : Fin q0 ⊕ Fin X.p0) : rowOf X hr (τ, ρ) = (τ, rowIdx hr ρ)

theorem **rowType_rowOf_inl** (τ : X.Ψ) (i : Fin q0) : rowType X (rowOf X hr (τ, Sum.inl i)) = τ.1

theorem **rowType_rowOf_inr** (τ : X.Ψ) (i : Fin X.p0) : rowType X (rowOf X hr (τ, Sum.inr i)) = ComplexEmbedding.conjugate τ.1

@[simp] theorem **wtOf_inl** (c : X.Ψ → Fin q0 → ℕ) (c' : X.Ψ → Fin X.p0 → ℕ) (n : ℕ) (τ : X.Ψ) (i : Fin q0) : wtOf X hr c c' n τ (rowIdx hr (Sum.inl i)) = (Harm.partOf (c τ) i : ℤ) + (n / 2 : ℕ)

@[simp] theorem **wtOf_inr** (c : X.Ψ → Fin q0 → ℕ) (c' : X.Ψ → Fin X.p0 → ℕ) (n : ℕ) (τ : X.Ψ) (i : Fin X.p0) : wtOf X hr c c' n τ (rowIdx hr (Sum.inr i)) = -(Harm.partOf (c' τ) i : ℤ) - (n / 2 : ℕ)

theorem **wtOf_isDominant** (c : X.Ψ → Fin q0 → ℕ) (c' : X.Ψ → Fin X.p0 → ℕ) (n : ℕ) : X.IsDominant (wtOf X hr c c' n)

theorem **thFr_add** (P Q : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) : thFr X hr bas idx n a b (P + Q) = thFr X hr bas idx n a b P + thFr X hr bas idx n a b Q

theorem **thFr_smul** (c : ℂ) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) : thFr X hr bas idx n a b (c • P) = c • thFr X hr bas idx n a b P

@[simp] theorem **thFrLin_apply** (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) : thFrLin X hr bas idx n a b P = thFr X hr bas idx n a b P

@[simp] theorem **thFr_zero** : thFr X hr bas idx n a b 0 = 0

theorem **thFr_one** (x : Frame X.Ψ r) : thFr X hr bas idx n a b 1 x = (hMat X bas idx x).det ^ (n / 2) * ∏ k, thetaChar (tauMat X bas idx x) (fun i => (a k i : ℂ)) (fun i => (b k i : ℂ)) 0

theorem **rowFac_ne_zero** (p : X.Ψ × Fin r) : rowFac X d p ≠ 0

theorem **per_rmul_diagFr** (x : Frame X.Ψ r) (v : Fin r → E) (p : X.Ψ × Fin r) : per X (rmul x (diagFr X d)) v p = rowFac X d p * per X x v p

theorem **perMat_rmul_diagFr** (s : Fin g → Fin g ⊕ Fin g) : perMat X bas idx s (rmul x (diagFr X d)) = Matrix.diagonal (fun i => rowFac X d (idx i)) * perMat X bas idx s x

theorem **hMat_rmul_diagFr** : hMat X bas idx (rmul x (diagFr X d)) = hMat X bas idx x * Matrix.diagonal fun i => (rowFac X d (idx i))⁻¹

theorem **polyDeriv_smul_letter** {σ W : Type*} [NormedAddCommGroup W] [NormedSpace ℂ W] (P : MvPolynomial σ ℂ) (s : σ → ℂ) (t : ℂ) (h : ∀ m ∈ P.support, (m.prod fun l e => s l ^ e) = t) (u : σ → W) (F : W → ℂ) (w : W) : polyDeriv P (fun l => s l • u l) F w = t * polyDeriv P u F w

theorem **vec_mem** (m : Fin 2 ⊕ Fin 2 → ℤ) : vec m ∈ flagData.Λ

theorem **EF_vec** (m m' : Fin 2 ⊕ Fin 2 → ℤ) : EF flagData (vec m) (vec m') = -(omegaStd m m' : ℤ)

theorem **vecLin_bijective** : Function.Bijective vecLin

theorem **isSympl_bas** : IsSympl flagData bas

theorem **hr** : flagData.p0 + 1 = 2

theorem **one_mem** : one ∈ flagData.dom

theorem **per_one** (v : Fin 2 → E4) (τ : flagData.Ψ) : per flagData one v (τ, 0) = conj (τ.1 (v 0)) ∧ per flagData one v (τ, 1) = τ.1 (v 1)

theorem **perMat_one** : perMat flagData bas idx Sum.inl one = 1 ∧ perMat flagData bas idx Sum.inr one = I • (1 : Matrix (Fin 2) (Fin 2) ℂ)

theorem **hMat_one** : hMat flagData bas idx one = 1

theorem **tauMat_one** : tauMat flagData bas idx one = I • (1 : Matrix (Fin 2) (Fin 2) ℂ)

theorem **thetaChar_I_ne_zero** : thetaChar (I • (1 : Matrix (Fin 2) (Fin 2) ℂ)) 0 0 0 ≠ 0

theorem **thFr_one_ne_zero** (n : ℕ) : thFr flagData hr bas idx n 0 0 1 one ≠ 0

Development.AbelianVarietyTheory.Definitions.ThetaValues.PowEqOneOfIsFullLev

theorem **mem_iff** (k : ℕ) (a : Dw.La) : a ∈ (Ideal.span {(k : 𝒪 E)} • ⊤ : Submodule (𝒪 E) Dw.La) ↔ ∃ b, a = k • b

theorem **finite_quot** {O M : Type} [CommRing O] [Module.Finite ℤ O] [AddCommGroup M] [Module O M] [Module.Finite O M] (N : Submodule O M) {k : ℕ} (hk : k ≠ 0) (h : ∀ a : M, k • a ∈ N) : Finite (M ⧸ N)

theorem **finite_latMod** {k : ℕ} (hk : k ≠ 0) : Finite (LatMod Dw k)

theorem **finite_levGrp** {k : ℕ} (hk : k ≠ 0) : Finite (LevGrp Dw k)

theorem **etaExpr_comp** {S : Type} [CommRing S] (k : ℕ) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) (h : LevGrp Dw k) : etaExpr k cf x (fun q => w (h q)) = etaExpr k cf x w

theorem **etaExpr_smul** {S : Type} [CommRing S] {k : ℕ} (hk : k ≠ 0) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) (t : S) : etaExpr k cf x (fun q => t * w q) = t ^ (n * Nat.card (LevGrp Dw k)) * etaExpr k cf x w

theorem **exists_isFullLev** {k : ℕ} (hk : 0 < k) {K : Type} [Field K] [IsAlgClosed K] [Algebra c.F K] (y : FrPt Xo c K) : ∃ ev, IsFullLev Xo k y.x ev ∧ ∀ z : ptO Xo y.x ⟶ Xo.univ.t.G, z ^ k = 1 ↔ ∃ q, ev q = z

theorem **exists_isNullVal** (hL : Laws) {N : ℕ} (hN : Odd N) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (p : ptO Xo y.x ⟶ Xo.univ.t.G) (hp : p ^ (N ^ 2) = 1) : ∃ w, IsNullVal Xo c N y p w

theorem **isNullVal_unique** (hL : Laws) {N : ℕ} (hN : Odd N) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (p : ptO Xo y.x ⟶ Xo.univ.t.G) (hp : p ^ (N ^ 2) = 1) {w w' : S} (h : IsNullVal Xo c N y p w) (h' : IsNullVal Xo c N y p w') : w = w'

theorem **exists_etaValAt** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw (R ^ 4)) {K : Type} [Field K] [IsAlgClosed K] [Algebra c.F K] (y : FrPt Xo c K) : ∃ v, EtaValAt Xo c R cf x y v

theorem **pow_eq_one_of_isFullLev** {k : ℕ} {S : Type} [CommRing S] {x0 : Sp S ⟶ Xo.X} {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (h : IsFullLev Xo k x0 ev) (q : LatMod Dw k) : ev q ^ k = 1

theorem **exists_cover** (hL : Laws) {R : ℕ} (hR : Odd R) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : ∃ (S' : Type) (_ : CommRing S') (_ : Algebra c.F S') (φ : S →ₐ[c.F] S') (y' : FrPt Xo c S') (ev : LatMod Dw (R ^ 4) → (ptO Xo y'.x ⟶ Xo.univ.t.G)) (w : LatMod Dw (R ^ 4) → S'), φ.toRingHom.FaithfullyFlat ∧ FrMap φ y y' ∧ IsFullLev Xo (R ^ 4) y'.x ev ∧ ∀ q, IsNullVal Xo c (R ^ 2) y' (ev q) (w q)

theorem **isEtaRule_zero_columns** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] (cf : J → ℤ) (x : J → Fin 0 → LatMod Dw (R ^ 4)) : IsEtaRule Xo c R cf x fun p _ => ((∑ j, cf j : ℤ) : p.R) ^ Nat.card (LevGrp Dw (R ^ 4))

theorem **exists_isEtaRule_one** (hL : Laws) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw (1 ^ 4)) : ∃ f, IsEtaRule Xo c 1 cf x f

Development.AbelianVarietyTheory.Definitions.ThetaValues.SWEqSOf

theorem **sW_eq_sOf** (B : WeightFfamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) {j : ℕ} (ka wL wF : WeightFamilies.Wt D j) (T : ℕ) : (scalOfP B hinj EX p ka wL wF T).sW = sOf B hinj EX wL

Development.AbelianVarietyTheory.Definitions.TorusSections.ExistsSecTie

theorem **IntSec.ofIntData_0** {Xo : Moduli Dw l} {c : Coeff l} {T : CDVR} {io : c.F →+* T.Fr} {y : letI

private lemma **eSec_comp_eq_comp_eSec_of_isGrpPullback** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {f : S' ⟶ S} {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) : eSec G' ≫ φ = f ≫ eSec G

theorem **exists_fib** (Xo : Moduli Dw l) {K : Type} [Field K] (x : Sp K ⟶ Xo.X) : ∃ A : PolAV (Spec (CommRingCat.of K)), Nonempty (Fib Xo x A)

theorem **Fib.SecTie.pt_eq** {Φ : Fib Xo x A} {xF : Sp F ⟶ Xo.X} {s : LB.Sec (PPx xF)} {th : GlobalTheory.UT.SecA A F 1} (h : Φ.SecTie xF s th) : xF = jFK (R

theorem **Fib.SecTie.unique** {Φ : Fib Xo x A} {xF : Sp F ⟶ Xo.X} {s s' : LB.Sec (PPx xF)} {th : GlobalTheory.UT.SecA A F 1} (h : Φ.SecTie xF s th) (h' : Φ.SecTie xF s' th) (i) : s.f i = s'.f i

theorem **Fib.exists_secTie** (Φ : Fib Xo x A) {xF : Sp F ⟶ Xo.X} (hx : xF = jFK (R

theorem **IntTh.isGen** {R : Type} [CommRing R] [IsDomain R] {I : Ideal R} {D : DegData R I} {A : PolAV (Spec (CommRingCat.of (FractionRing R)))} {G : Over (Spec (CommRingCat.of R))} [GrpObj G] [IsCommMonObj G] {U : UnifPre I D A G} {F : Type} [Field F] [Algebra (FractionRing R) F] {w : GlobalTheory.UT.W0 U} {th : GlobalTheory.UT.SecA A F 1} (h : IntTh U w th) : w.IsGen

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ChainParSucc

theorem **chainPar_succ** (c : Com D) (B : GlobalTheory.LFun.Fam D.host.E) (k : ℕ) : chainPar c B (k + 1) = stepPar (c.χ k) (c.χ k) c.t (chainPar c B k)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ClassSumOfPin

theorem **theta_cast** (j : ℕ) (μ ν : D.what₂.Tup) (h : μ = ν) (x : D.what₂.MtL μ j) : D.what₂.thetaP ν j (D.what₂.jMt ν j (h ▸ x)) = D.what₂.thetaP μ j (D.what₂.jMt μ j x)

theorem **pin_of_export** (p : PtAt ex Csz) (hE : p.ExportRow) (hD : DivRows3 D p.P.A3 p.P.P0 p.P.ψ0 p.P.Ps p.P.ψs p.P.Zs p.P.S0 p.P.S1 ex.p0 p.P.C p.P.NuC p.P.I) : ClassSumPin p

theorem **pin_of_rows** (p : PtAt ex Csz) (h : p.Rows) : ClassSumPin p

theorem **dim_eq** (p : PtAt ex Csz) (i : Fin D.R.m) : (p.P.A3.indefDown p.P.S0).dW i = Module.finrank D.what₂.LL (p.R.toRealDat.WL p.F.lv2 false i)

theorem **exists_heq_fin** {K : Type} {a b : ℕ} (h : a = b) (z : Fin a → K) : ∃ y : Fin b → K, HEq z y

theorem **exists_cast** (j : ℕ) (μ ν : D.what₂.Tup) (h : μ = ν) (x : D.what₂.MtL ν j) : ∃ x' : D.what₂.MtL μ j, x = h ▸ x'

theorem **exists_sum** (p : PtAt ex Csz) (i : Fin D.R.m) (v : Fin (p.R.nW i) → D.what₂.LL) : ∃ (n : ℕ) (a : Fin n → D.what₂.LL) (f : Fin n → FmK p.Sx ((i : ℕ) + 1)), (p.nexp.fi i).e (∑ k, a k • (show ɪ(p.Sx.A ((i : ℕ) + 1)).WK from f k).1) = v

theorem **premises_met** (p : PtAt ex Csz) (w : D.what₂.places.Pl) (i : Fin D.R.m) (hw : D.what₂.muAt w ≠ D.what₂.mu0) (x : D.what₂.MtL (D.what₂.muTw w) ((i : ℕ) + 1)) : ∃ (t : p.P.T) (ht : p.P.I.tup t = D.what₂.muTw w) (c : Fin (lDim D p.P.A3.A p.P.P0 p.P.ψ0 i t) → D.what₂.LL) (y : Fin (Module.finrank D.what₂.LL (p.R.toRealDat.WL p.F.lv2 false i)) → D.what₂.LL) (n : ℕ) (a : Fin n → D.what₂.LL) (f : Fin n → FmK p.Sx ((i : ℕ) + 1)), x = ht ▸ p.P.I.eMt i t c ∧ HEq (((p.P.A3.indefDown p.P.S0).θ i t).mulVec ((lPrj D p.P.A3.A p.P.P0 p.P.ψ0 (p.P.A3.fam3.toSub2 p.P.Zs) i t).mulVec c)) y ∧ (p.nexp.fi i).e (∑ k, a k • (show ɪ(p.Sx.A ((i : ℕ) + 1)).WK from f k).1) = jDL p.R p.F.lv2 i y

theorem **classSum_of_pin** (p : PtAt ex Csz) (hP : ClassSumPin p) (hD : DivRows3 D p.P.A3 p.P.P0 p.P.ψ0 p.P.Ps p.P.ψs p.P.Zs p.P.S0 p.P.S1 ex.p0 p.P.C p.P.NuC p.P.I) (w : D.what₂.places.Pl) (i : Fin D.R.m) (h1 : 1 ≤ (i : ℕ) + 1) (h2 : (i : ℕ) + 1 ≤ D.R.m) (hw : D.what₂.muAt w ≠ D.what₂.mu0) (jD : (Y.MjH ((i : ℕ) + 1)).V → p.Sx.WL ((i : ℕ) + 1)) (hjD : ∀ (x : (Y.MjH ((i : ℕ) + 1)).V) (y : Fin (Module.finrank D.what₂.LL (p.R.toRealDat.WL p.F.lv2 false i)) → D.what₂.LL), HEq ((p.P.JM.eW i).symm x) y → (p.nexp.fi i).e (jD x) = jDL p.R p.F.lv2 i y) (eMj : (Y.MjH ((i : ℕ) + 1)).V → FractionRing (ex.nch.Oe w) ⊗[D.ntw.O (ex.nch.twI w)] D.nexp.Fm (ex.nch.twI w) ((i : ℕ) + 1) (ex.nch.sd w)) (hsum : ∀ (x : (Y.MjH ((i : ℕ) + 1)).V) (n : ℕ) (a : Fin n → D.what₂.LL) (f : Fin n → FmK p.Sx ((i : ℕ) + 1)), jD x = ∑ k, a k • (show ɪ(p.Sx.A ((i : ℕ) + 1)).WK from f k).1 → eMj x = ∑ k, ex.nch.embW w (a k) ⊗ₜ[D

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ConjScalarAtOf

theorem **conjScalarAt_of** (h : ConjScalarThm) (R : RealDat D A Ks) (F : Face D A) (hC : R.Char F) (hψ : IsStdChar D R.ψ (-1)) (hX : ∀ χ ∈ A.X, IsAbsIrrChar χ) : ConjScalarAt R

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ContinuousResTSymm

theorem **cancel₂** {k l m n : ℕ} {a : Mx D k l} {b : Mx D l m} {c : Mx D k m} (h : a · b = c) (x : Mx D m n) : a · (b · x) = c · x

theorem **cancel₃** {j k l m n : ℕ} {a : Mx D j k} {b : Mx D k l} {c : Mx D l m} {e : Mx D j m} (h : a · (b · c) = e) (x : Mx D m n) : a · (b · (c · x)) = e · x

theorem **gram_swap** (S : HSp D) {k l : ℕ} {a : Mx D S.rank k} {b : Mx D S.rank l} {c : Mx D k l} (h : adj a · (gA S · b) = c) : adj b · (gA S · a) = adj c

theorem **gram_rat** (S : HSp D) {k l : ℕ} {a : Matrix (Fin S.rank) (Fin k) D.host.E} {b : Matrix (Fin S.rank) (Fin l) D.host.E} {c : Matrix (Fin k) (Fin l) D.host.E} (h : aᴴ ∗ S.gram ∗ b = c) : adj (rat a) · (gA S · rat b) = rat c

theorem **orth'** : d.iAᴴ ∗ S.gram ∗ d.iB = 0

theorem **qA_iA** : d.qA ∗ d.iA = 1

theorem **qA_iB** : d.qA ∗ d.iB = 0

theorem **qB_iB** : d.qB ∗ d.iB = 1

theorem **qB_iA** : d.qB ∗ d.iA = 0

theorem **iA_qA_add** : d.iA ∗ d.qA + d.iB ∗ d.qB = 1

theorem **pA_eq** : d.pA = giA A · (adj d.jA · gA S)

theorem **pB_eq** : d.pB = giA B · (adj d.jB · gA S)

theorem **gram_AA** : adj d.jA · (gA S · d.jA) = gA A

theorem **gram_BB** : adj d.jB · (gA S · d.jB) = gA B

theorem **gram_BA** : adj d.jB · (gA S · d.jA) = 0

theorem **gram_AB** : adj d.jA · (gA S · d.jB) = 0

@[simp] theorem **pA_jA** : d.pA · d.jA = 1

@[simp] theorem **pA_jB** : d.pA · d.jB = 0

@[simp] theorem **pB_jB** : d.pB · d.jB = 1

@[simp] theorem **pB_jA** : d.pB · d.jA = 0

theorem **sum_eq_one** : d.jA · d.pA + d.jB · d.pB = 1

theorem **gram_images** {I : Mx D S.rank A.rank} {J : Mx D S.rank B.rank} (hI : adj I · (gA S · I) = gA A) (hJ : adj J · (gA S · J) = gA B) (hJI : adj J · (gA S · I) = 0) : adj (I · d.pA + J · d.pB) · (gA S · (I · d.pA + J · d.pB)) = adj d.pA · (gA A · d.pA) + adj d.pB · (gA B · d.pB)

theorem **gA_eq** : gA S = adj d.pA · (gA A · d.pA) + adj d.pB · (gA B · d.pB)

theorem **giA_eq** : giA S = d.jA · (giA A · adj d.jA) + d.jB · (giA B · adj d.jB)

theorem **mt_emb** (a : GA D A) (b : GA D B) : mt S (d.emb (a, b)) = d.jA · mt A a · d.pA + d.jB · mt B b · d.pB

theorem **emb_jA** (a : GA D A) (b : GA D B) : mt S (d.emb (a, b)) · d.jA = d.jA · mt A a

theorem **emb_jB** (a : GA D A) (b : GA D B) : mt S (d.emb (a, b)) · d.jB = d.jB · mt B b

theorem **pA_emb** (a : GA D A) (b : GA D B) : d.pA · mt S (d.emb (a, b)) = mt A a · d.pA

theorem **pB_emb** (a : GA D A) (b : GA D B) : d.pB · mt S (d.emb (a, b)) = mt B b · d.pB

theorem **emb_symm** (a : GA D A) (b : GA D B) : d.symm.emb (b, a) = d.emb (a, b)

theorem **emb_injective** : Function.Injective d.emb

theorem **continuous_emb** : Continuous d.emb

theorem **emb_mem_GQ** {a : GA D A} {b : GA D B} (ha : a ∈ GQ A) (hb : b ∈ GQ B) : d.emb (a, b) ∈ GQ S

theorem **embRel_left** (ι : A.V →ₗ[D.host.E] S.V) (hι : frameMat S A ι = d.iA) (a : GA D A) : EmbRel S A ι (d.emb (a, 1)) a

theorem **embRel_right** (ι : B.V →ₗ[D.host.E] S.V) (hι : frameMat S B ι = d.iB) (b : GA D B) : EmbRel S B ι (d.emb (1, b)) b

theorem **resS_apply** (T : HSp D) (x : WO S T) : d.resS T x = (x · d.jA, x · d.jB)

theorem **resS_symm_apply** (T : HSp D) (u : WO A T) (v : WO B T) : (d.resS T).symm (u, v) = u · d.pA + v · d.pB

theorem **hO_resS** (T : HSp D) (x y : WO S T) : hO S T x y = hO A T (d.resS T x).1 (d.resS T y).1 + hO B T (d.resS T x).2 (d.resS T y).2

theorem **symplOH_resS** (T : HSp D) (x y : WO S T) : symplOH S T x y = symplOH A T (d.resS T x).1 (d.resS T y).1 + symplOH B T (d.resS T x).2 (d.resS T y).2

theorem **symplO_resS** (T : HSp D) (x y : WO S T) : symplO S T x y = symplO A T (d.resS T x).1 (d.resS T y).1 + symplO B T (d.resS T x).2 (d.resS T y).2

theorem **resS_actO** (T : HSp D) (a : GA D A) (b : GA D B) (h : GA D T) (x : WO S T) : d.resS T (actO S T (d.emb (a, b)) h x) = (actO A T a h (d.resS T x).1, actO B T b h (d.resS T x).2)

theorem **mem_ratL_resS** (T : HSp D) (x : WO S T) : x ∈ ratL T S ↔ (d.resS T x).1 ∈ ratL T A ∧ (d.resS T x).2 ∈ ratL T B

theorem **continuous_resS** (T : HSp D) : Continuous (d.resS T)

theorem **continuous_resS_symm** (T : HSp D) : Continuous (d.resS T).symm

theorem **boxS_zero** (T : HSp D) (fA : WO A T → ℂ) (fB : WO B T → ℂ) : d.boxS T fA fB 0 = fA 0 ∗ fB 0

theorem **heisO_boxS** (ψ : AddChar (AF D.host.F) ℂ) (T : HSp D) (fA : WO A T → ℂ) (fB : WO B T → ℂ) (u : WO S T) : heisO ψ S T u (d.boxS T fA fB) = d.boxS T (heisO ψ A T (d.resS T u).1 fA) (heisO ψ B T (d.resS T u).2 fB)

theorem **isThetaFunO_boxS** {ψ : AddChar (AF D.host.F) ℂ} {T : HSp D} {fA : WO A T → ℂ} {fB : WO B T → ℂ} (hA : IsThetaFunO ψ A T fA) (hB : IsThetaFunO ψ B T fB) : IsThetaFunO ψ S T (d.boxS T fA fB)

theorem **resT_apply** (S : HSp D) (x : WO S T) : d.resT S x = (d.pA · x, d.pB · x)

theorem **resT_symm_apply** (S : HSp D) (u : WO S A) (v : WO S B) : (d.resT S).symm (u, v) = d.jA · u + d.jB · v

theorem **hO_resT** (S : HSp D) (x y : WO S T) : hO S T x y = hO S A (d.resT S x).1 (d.resT S y).1 + hO S B (d.resT S x).2 (d.resT S y).2

theorem **symplOH_resT** (S : HSp D) (x y : WO S T) : symplOH S T x y = symplOH S A (d.resT S x).1 (d.resT S y).1 + symplOH S B (d.resT S x).2 (d.resT S y).2

theorem **symplO_resT** (S : HSp D) (x y : WO S T) : symplO S T x y = symplO S A (d.resT S x).1 (d.resT S y).1 + symplO S B (d.resT S x).2 (d.resT S y).2

theorem **resT_actO** (S : HSp D) (g : GA D S) (a : GA D A) (b : GA D B) (x : WO S T) : d.resT S (actO S T g (d.emb (a, b)) x) = (actO S A g a (d.resT S x).1, actO S B g b (d.resT S x).2)

theorem **mem_ratL_resT** (S : HSp D) (x : WO S T) : x ∈ ratL T S ↔ (d.resT S x).1 ∈ ratL A S ∧ (d.resT S x).2 ∈ ratL B S

theorem **continuous_resT** (S : HSp D) : Continuous (d.resT S)

theorem **continuous_resT_symm** (S : HSp D) : Continuous (d.resT S).symm

theorem **boxT_zero** (S : HSp D) (fA : WO S A → ℂ) (fB : WO S B → ℂ) : d.boxT S fA fB 0 = fA 0 * fB 0

theorem **heisO_boxT** (ψ : AddChar (AF D.host.F) ℂ) (S : HSp D) (fA : WO S A → ℂ) (fB : WO S B → ℂ) (u : WO S T) : heisO ψ S T u (d.boxT S fA fB) = d.boxT S (heisO ψ S A (d.resT S u).1 fA) (heisO ψ S B (d.resT S u).2 fB)

theorem **isThetaFunO_boxT** {ψ : AddChar (AF D.host.F) ℂ} {S : HSp D} {fA : WO S A → ℂ} {fB : WO S B → ℂ} (hA : IsThetaFunO ψ S A fA) (hB : IsThetaFunO ψ S B fB) : IsThetaFunO ψ S T (d.boxT S fA fB)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.CovolOneIntegralAdd

theorem **ratL_eq** (V W : HSp D) : ratL V W = ratMx D V.rank W.rank

theorem **ratW_eq** (V W : HSp D) (γ : Matrix (Fin V.rank) (Fin W.rank) D.host.E) : ratW V W γ = rat γ

theorem **mem_ratMx** {x : Mx D m n} : x ∈ ratMx D m n ↔ ∃ γ, rat γ = x

theorem **rat_mem** (γ : Matrix (Fin m) (Fin n) D.host.E) : rat γ ∈ ratMx D m n

theorem **incA_injective** : Function.Injective (incA D)

theorem **rat_injective** : Function.Injective (rat : Matrix (Fin m) (Fin n) D.host.E → Mx D m n)

@[simp] theorem **rat_mul** (M : Matrix (Fin l) (Fin m) D.host.E) (N : Matrix (Fin m) (Fin n) D.host.E) : rat (M * N) = rat M · rat N

@[simp] theorem **rat_add** (M N : Matrix (Fin m) (Fin n) D.host.E) : rat (M + N) = rat M + rat N

@[simp] theorem **rat_sub** (M N : Matrix (Fin m) (Fin n) D.host.E) : rat (M - N) = rat M - rat N

@[simp] theorem **rat_neg** (M : Matrix (Fin m) (Fin n) D.host.E) : rat (-M) = -rat M

@[simp] theorem **rat_zero** : rat (0 : Matrix (Fin m) (Fin n) D.host.E) = 0

@[simp] theorem **rat_one** : rat (1 : Matrix (Fin n) (Fin n) D.host.E) = 1

@[simp] theorem **rat_smul** (a : D.host.E) (M : Matrix (Fin m) (Fin n) D.host.E) : rat (a • M) = incA D a • rat M

theorem **rat_conjTranspose** (M : Matrix (Fin m) (Fin n) D.host.E) : rat Mᴴ = adj (rat M)

theorem **cT_sig** (M : Matrix (Fin m) (Fin n) D.host.E) : Formalism.UG.cT (D.sp.cm.sig : D.host.E →+* D.host.E) M = Mᴴ

@[simp] theorem **adj_mul** (x : Mx D l m) (y : Mx D m n) : adj (x · y) = adj y · adj x

@[simp] theorem **adj_add** (x y : Mx D m n) : adj (x + y) = adj x + adj y

@[simp] theorem **adj_sub** (x y : Mx D m n) : adj (x - y) = adj x - adj y

@[simp] theorem **adj_neg** (x : Mx D m n) : adj (-x) = -adj x

@[simp] theorem **adj_zero** : adj (0 : Mx D m n) = 0

@[simp] theorem **adj_one** : adj (1 : Mx D n n) = 1

@[simp] theorem **adj_adj** (x : Mx D m n) : adj (adj x) = x

@[simp] theorem **adj_smul** (a : AE D) (x : Mx D m n) : adj (a • x) = cA D a • adj x

theorem **trace_adj** (x : Mx D n n) : (adj x).trace = cA D x.trace

@[simp] theorem **cA_half** : cA D (half D) = half D

theorem **half_add_half** : half D + half D = 1

theorem **half_smul_add** (x : Mx D m n) : half D • x + half D • x = x

theorem **continuous_adj** : Continuous (adj : Mx D m n → Mx D n m)

theorem **gram_conjTranspose** : S.gramᴴ = S.gram

theorem **gram_isUnit_det** : IsUnit S.gram.det

theorem **gramInv_conjTranspose** : (S.gram⁻¹)ᴴ = S.gram⁻¹

@[simp] theorem **adj_gA** : adj (gA S) = gA S

@[simp] theorem **adj_giA** : adj (giA S) = giA S

@[simp] theorem **giA_mul_gA** : giA S * gA S = 1

@[simp] theorem **gA_mul_giA** : gA S * giA S = 1

@[simp] theorem **giA_gA_cancel** {n : ℕ} (x : Mx D S.rank n) : giA S · (gA S · x) = x

@[simp] theorem **gA_giA_cancel** {n : ℕ} (x : Mx D S.rank n) : gA S · (giA S · x) = x

theorem **hO_eq_trace** (T : HSp D) (x y : WO S T) : hO S T x y = (giA S · (adj y · (gA T · x))).trace

theorem **mt_unitary** (g : GA D S) : adj (mt S g) * (gA S * mt S g) = gA S

theorem **mt_mul** (g h : GA D S) : mt S (g * h) = mt S g * mt S h

@[simp] theorem **mt_one** : mt S 1 = 1

theorem **mt_inv_mul** (g : GA D S) : mt S g⁻¹ * mt S g = 1

theorem **mt_mul_inv** (g : GA D S) : mt S g * mt S g⁻¹ = 1

theorem **mt_inv** (g : GA D S) : mt S g⁻¹ = giA S * (adj (mt S g) * gA S)

theorem **mt_injective** : Function.Injective (mt S)

@[simp] theorem **mt_ofMat** (M : Mx D S.rank S.rank) (h : adj M * (gA S * M) = gA S) : mt S (ofMat S M h) = M

theorem **mem_GQ_iff** (g : GA D S) : g ∈ GQ S ↔ mt S g ∈ ratMx D S.rank S.rank

theorem **continuous_mt** : Continuous (mt S)

theorem **continuous_of_mt** {X : Type} [TopologicalSpace X] {φ : X → GA D S} (h : Continuous fun x => mt S (φ x)) : Continuous φ

theorem **actO_eq** (T : HSp D) (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g h x = mt T h · x · mt S g⁻¹

theorem **ratMx_mul** {x : Mx D l m} {y : Mx D m n} (hx : x ∈ ratMx D l m) (hy : y ∈ ratMx D m n) : x · y ∈ ratMx D l n

theorem **ratMx_adj** {x : Mx D m n} (hx : x ∈ ratMx D m n) : adj x ∈ ratMx D n m

theorem **ratMx_smul** (a : D.host.E) {x : Mx D m n} (hx : x ∈ ratMx D m n) : incA D a • x ∈ ratMx D m n

theorem **ratMx_one** : (1 : Mx D n n) ∈ ratMx D n n

theorem **lattice_Mx** : IsTopologicalAddGroup (Mx D m n) ∧ LocallyCompactSpace (Mx D m n) ∧ SecondCountableTopology (Mx D m n) ∧ T2Space (Mx D m n) ∧ DiscreteTopology (ratMx D m n) ∧ ∃ K : Set (Mx D m n), IsCompact K ∧ ∀ x : Mx D m n, ∃ γ ∈ ratMx D m n, γ + x ∈ K

theorem **exists_isCovolOne** : ∃ μ ℱ, IsCovolOne (ratMx D m n) μ ℱ

theorem **covolOne_measure_eq** {μ μ' : Measure (Mx D m n)} {ℱ ℱ' : Set (Mx D m n)} (h : IsCovolOne (ratMx D m n) μ ℱ) (h' : IsCovolOne (ratMx D m n) μ' ℱ') : μ' = μ

theorem **covolOne_integral_eq** {μ μ' : Measure (Mx D m n)} {ℱ ℱ' : Set (Mx D m n)} (h : IsCovolOne (ratMx D m n) μ ℱ) (h' : IsCovolOne (ratMx D m n) μ' ℱ') (φ : Mx D m n → ℂ) (hφ : ∀ γ ∈ ratMx D m n, ∀ a, φ (γ + a) = φ a) : ∫ a in ℱ', φ a ∂μ' = ∫ a in ℱ, φ a ∂μ

theorem **covolOne_integrableOn** {μ : Measure (Mx D m n)} {ℱ : Set (Mx D m n)} (h : IsCovolOne (ratMx D m n) μ ℱ) (φ : Mx D m n → ℂ) (hc : Continuous φ) (hφ : ∀ γ ∈ ratMx D m n, ∀ a, φ (γ + a) = φ a) : IntegrableOn φ ℱ μ

theorem **covolOne_integral_add** {μ : Measure (Mx D m n)} {ℱ : Set (Mx D m n)} (h : IsCovolOne (ratMx D m n) μ ℱ) (φ : Mx D m n → ℂ) (hφ : ∀ γ ∈ ratMx D m n, ∀ a, φ (γ + a) = φ a) (u : Mx D m n) : ∫ a in ℱ, φ (a + u) ∂μ = ∫ a in ℱ, φ a ∂μ

theorem **symplOH_eq** (S T : HSp D) (x y : WO S T) : symplOH S T x y = trHalf D (hO S T x y)

theorem **symplO_eq** (S T : HSp D) (x y : WO S T) : symplO S T x y = Algebra.trace (AF D.host.F) (AE D) (incA D D.host.xi0 * hO S T x y)

theorem **symplOH_zero_left** (S T : HSp D) (y : WO S T) : symplOH S T 0 y = 0

theorem **symplOH_zero_right** (S T : HSp D) (x : WO S T) : symplOH S T x 0 = 0

theorem **symplOH_skew** (S T : HSp D) (x y : WO S T) : symplOH S T y x = -symplOH S T x y

theorem **map_mul_eq_zero_left** {B : Type} [CommRing B] (φ : AE D →+* B) {l m n : ℕ} {x : Mx D l m} (hx : x.map φ = 0) (y : Mx D m n) : (x · y).map φ = 0

theorem **map_mul_eq_zero_right** {B : Type} [CommRing B] (φ : AE D →+* B) {l m n : ℕ} (x : Mx D l m) {y : Mx D m n} (hy : y.map φ = 0) : (x · y).map φ = 0

theorem **sandwich_apply** {k l m n : ℕ} (P : Mx D k l) (Q : Mx D m n) (w : Mx D l m) : sandwich P Q w = P · w · Q

theorem **continuous_sandwich** {k l m n : ℕ} (P : Mx D k l) (Q : Mx D m n) : Continuous (sandwich P Q)

theorem **map_sandwich** {B : Type} [CommRing B] (φ : AE D →+* B) {k l m n : ℕ} (P : Mx D k l) (Q : Mx D m n) {w : Mx D l m} (hw : w.map φ = 0) : (sandwich P Q w).map φ = 0

theorem **isThetaFunO_prod** (P₁ : Matrix (Fin T₁.rank) (Fin T.rank) D.host.E) (Q₁ : Matrix (Fin S.rank) (Fin S₁.rank) D.host.E) (P₂ : Matrix (Fin T₂.rank) (Fin T.rank) D.host.E) (Q₂ : Matrix (Fin S.rank) (Fin S₂.rank) D.host.E) (h : ∀ x y : WO S T, symplOH S T x y = symplOH S₁ T₁ (sandwich (rat P₁) (rat Q₁) x) (sandwich (rat P₁) (rat Q₁) y) + symplOH S₂ T₂ (sandwich (rat P₂) (rat Q₂) x) (sandwich (rat P₂) (rat Q₂) y)) {f₁ : WO S₁ T₁ → ℂ} {f₂ : WO S₂ T₂ → ℂ} (h₁ : IsThetaFunO ψ S₁ T₁ f₁) (h₂ : IsThetaFunO ψ S₂ T₂ f₂) : IsThetaFunO ψ S T fun w => f₁ (sandwich (rat P₁) (rat Q₁) w) * f₂ (sandwich (rat P₂) (rat Q₂) w) where

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.DecayAtOfPos

theorem **exists_frame** {r n : ℕ} (h : r + 1 = n) (G : Matrix (Fin n) (Fin n) ℂ) (hG : Gᴴ = G) (I : Matrix (Fin n) (Fin r) ℂ) (Z : Matrix (Fin n) (Fin 1) ℂ) (FA : Matrix (Fin r) (Fin r) ℂ) (hA : IsUnit FA) (εA : Fin r → Bool) (h1 : Iᴴ * G * I = FAᴴ * Formalism.REAL.Jm εA * FA) (h2 : Iᴴ * G * Z = 0) (h3 : Zᴴ * G * Z = 1) : ∃ (ε : Fin n → Bool) (FB : Matrix (Fin n) (Fin n) ℂ), IsUnit FB ∧ FBᴴ * Formalism.REAL.Jm ε * FB = G ∧ ∀ (y : Fin r → ℂ) (a : ℂ), eu (FB *ᵥ (I *ᵥ y + a • fun i => Z i 0)) ^ 2 = eu (FA *ᵥ y) ^ 2 + ‖a‖ ^ 2

theorem **exists_adapted** {Y V : HSp D} (d : HermitianOrthogonalExtensionData Y V 1 1) (A : Fr Y) (B₀ : Fr V) : ∃ B : Fr V, Adapted d A B

theorem **decayAt_of_pos** {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) (hε : ∀ i, M.ε τ i = true) (v a : M.H τ) (N : ℕ) : DecayAt M τ v a N

theorem **exists_pack** {Y V : HSp D} (d : HermitianOrthogonalExtensionData Y V 1 1) {S : D.what₂.Tau → SPar} {S' : D.what₂.Tau → SPar} (M₀ : DSModel Y S') (M : DSModel V S) : ∃ p : Pack M₀ M, Adapted d p.A p.B ∧ p.g₀ ∈ GinfC V ∧ (∀ τ, p.v₀ τ ∈ GlobalTheory.IndefiniteRowSideFullHecke.KTypeExtension.kFin M τ ∧ p.v₀ τ ≠ 0) ∧ ∀ τ, (p.w τ : M₀.H τ) ≠ 0

theorem **exists_base** {Y V : HSp D} (d : HermitianOrthogonalExtensionData Y V 1 1) {S : D.what₂.Tau → SPar} {S' : D.what₂.Tau → SPar} (M₀ : DSModel Y S') (M : DSModel V S) (hε : ∀ τ i, M.ε τ i = true) : ∃ p : Pack M₀ M, p.Base d ∧ KDecay M p.v₀ ∧ ∀ τ, p.v₀ τ ≠ 0

theorem **v_ne_zero** (θ : InfFam D) {Y V : HSp D} (d : HermitianOrthogonalExtensionData Y V 1 1) {S : D.what₂.Tau → SPar} {S' : D.what₂.Tau → SPar} {M₀ : DSModel Y S'} {M : DSModel V S} (p : Pack M₀ M) (hb : p.Base d) {Q : ℝ} {m : Measure (GinfC Y)} {c : ℂ} (h : p.RealAt θ d Q m c) (τ : D.what₂.Tau) : p.v θ τ ≠ 0

theorem **exists_pays** (D : D7) (Q : ℝ) : ∃ N, Pays D Q N

theorem **not_pays_one** (D : D7) (Q : ℝ) : ¬ Pays D Q 1

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.DepthPays

theorem **eu_zero** {n : ℕ} : eu (0 : Fin n → ℂ) = 0

theorem **Long.mono** {θ : InfFam D} {V : HSp D} {A : Fr V} {Λ : Lat V} {x : GA D V} {s s' : ℝ} (h : Long θ A Λ x s) (h0 : 0 ≤ s') (hs : s' ≤ s) : Long θ A Λ x s'

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsFFNeZeroThree

theorem **isEmpty_pair** [IsEmpty n] : IsEmpty (Pair k n)

theorem **isAdj_sumOp** (S S' : Pair k n → Finset (Pair k n)) (w w' : Pair k n → Pair k n → ℂ) (hS : ∀ P Q, Q ∈ S P ↔ P ∈ S' Q) (hw : ∀ P Q, Q ∈ S P → w P Q = (starRingEnd ℂ) (w' Q P)) : IsAdj (sumOp S w) (sumOp S' w')

theorem **isAdj_FF** {ψ : AddChar k Circle} (hψ : SplitWeil.HasConductor ψ 1) (q : Unif k) : IsAdj (FF q (resChar ψ hψ q)) : Module.End ℂ (Form k n ℂ)) (FFStar q (resChar ψ hψ q))

theorem **isAdj_U** (q : Unif k) : IsAdj (U q : Module.End ℂ (Form k n ℂ)) (D q)

theorem **sum_withLat_dia** {A : Type*} [AddCommMonoid A] (c : 𝓀[k]ˣ) (L : Submodule 𝒪[k] (n → k)) (F : Pair k n → A) : ∑ P ∈ withLat L, F (c • P) = ∑ P ∈ withLat L, F P

theorem **FF_dia** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (c : 𝓀[k]ˣ) (φ : Form k n M) : FF q χ (dia c φ) = dia c (FF q χ φ)

theorem **sum_char** {V : Type*} [AddCommGroup V] [Module 𝓀[k] V] [Fintype V] {χ : AddChar 𝓀[k] ℂ} (hχ : χ ≠ 1) (g : Module.Dual 𝓀[k] V) [Decidable (g = 0)] : ∑ x, χ (g x) = if g = 0 then (Fintype.card V : ℂ) else 0

theorem **FFStar_FF** [Fintype n] (q : Unif k) {χ : AddChar 𝓀[k] ℂ} (hχ : χ ≠ 1) (φ : Form k n M) : FFStar q χ (FF q χ φ) = (SplitWeil.resCard k ^ Fintype.card n : ℂ) • φ − tr φ

theorem **sum_stable_eq_zero** {φ : Form k n M} (h : φ ∈ Mram k n M) (s : Finset (Pair k n)) (hs : ∀ (c : 𝓀[k]ˣ) P, c • P ∈ s ↔ P ∈ s) : ∑ P ∈ s, φ P = 0

theorem **tr_eq_zero** {φ : Form k n M} (h : φ ∈ Mram k n M) : tr φ = 0 ∧ tr' φ = 0

theorem **FF_eq_D_Gs** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) {φ : Form k n M} (h : φ ∈ Mram k n M) : FF q χ φ = D q (Gs χ φ)

theorem **FF_ne_zero** [Fintype n] (q : Unif k) {χ : AddChar 𝓀[k] ℂ} (hχ : χ ≠ 1) {φ : Form k n M} (h : φ ∈ Mram k n M) (h0 : FF q χ φ = 0) : φ = 0

theorem **Mram_eq_bot** [Subsingleton 𝓀[k]ˣ] : Mram k n M = ⊥

theorem **colE_smul_gluedLat** [Fintype n] [DecidableEq n] (q : Unif k) {c p : ℕ} (h1 : 1 ≤ p) (hp : p ≤ c) (L : Submodule 𝒪[k] (n → k)) (f : n → k) : RootEx.colE c p (−((q.1 : k) • f)) • gluedLat q c p L f = gluedLat q c p L 0

theorem **glued_injective** (q : Unif k) {c p : ℕ} (h1 : 1 ≤ p) (hp : p ≤ c) : Function.Injective fun P : Pair k n => P.glued q c p

theorem **phiOne_one_dia** [Fintype n] [DecidableEq n] (P : Pair k n) {c : 𝓀[k]ˣ} (hc : c ≠ 1) : phiOne P (c • P) 1 = 0

theorem **exists_FF_ne_zero** [Fintype n] [Nonempty n] {c : 𝓀[k]ˣ} (hc : c ≠ 1) : ∃ (q : Unif k) (ψ : AddChar k Circle) (hψ : SplitWeil.HasConductor ψ 1) (φ : Form k n ℂ), φ ∈ Mram k n ℂ ∧ FF q (resChar ψ hψ q) φ ≠ 0

theorem **exists_FF_ne_zero_three** : letI

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsGlOfEquiv

theorem **ne_zero** : ((q.1 : 𝒪[k]) : k) ≠ 0

theorem **mem_max_iff** {a : 𝒪[k]} : a ∈ maximalIdeal 𝒪[k] ↔ q.1 ∣ a

theorem **mem_max** : q.1 ∈ maximalIdeal 𝒪[k]

@[simp] theorem **smul_inv_smul** {n : Type*} (x : n → k) : q.1 • q.inv • x = x

@[simp] theorem **inv_smul_smul** {n : Type*} (x : n → k) : q.inv • q.1 • x = x

theorem **HasVal.mem** (h : P.HasVal x a) : x ∈ P.lat

theorem **HasVal.unique** (h : P.HasVal x a) (h' : P.HasVal x b) : a = b

theorem **hasVal_fn** (P : Pair k n) (x : P.lat) : P.HasVal x (P.fn x)

theorem **mem_lat_iff** : x ∈ P.lat ↔ ∃ a, P.HasVal x a

theorem **HasVal.add** (h : P.HasVal x a) (h' : P.HasVal y b) : P.HasVal (x + y) (a + b)

theorem **HasVal.sub** (h : P.HasVal x a) (h' : P.HasVal y b) : P.HasVal (x − y) (a − b)

theorem **HasVal.smul** (h : P.HasVal x a) (s : 𝒪[k]) : P.HasVal (s • x) (residue 𝒪[k] s ∗ a)

theorem **ext** (h : ∀ x a, P.HasVal x a ↔ Q.HasVal x a) : P = Q

theorem **exists_hasVal_one** (P : Pair k n) : ∃ e, P.HasVal e 1

theorem **mem_ker** : x ∈ P.ker ↔ P.HasVal x 0

theorem **ker_le** (P : Pair k n) : P.ker ≤ P.lat

theorem **smul_mem_ker** {s : 𝒪[k]} (hs : s ∈ maximalIdeal 𝒪[k]) (hx : x ∈ P.lat) : s • x ∈ P.ker

theorem **ker_isLat** (P : Pair k n) : P.ker.IsLattice k

theorem **hasVal_vec** (P : Pair k n) : P.HasVal P.vec 1

theorem **hasVal_iff** (he : P.HasVal e 1) : P.HasVal x a ↔ ∃ s : 𝒪[k], residue 𝒪[k] s = a ∧ x − s • e ∈ P.ker

theorem **hasVal_one_iff** (he : P.HasVal e 1) : P.HasVal x 1 ↔ x − e ∈ P.ker

theorem **sub_vec_mem** (he : P.HasVal e 1) : e − P.vec ∈ P.ker

theorem **lat_eq** (he : P.HasVal e 1) : P.lat = P.ker ⊔ Submodule.span 𝒪[k] {e}

theorem **ext_ker** (hk : P.ker = Q.ker) (he : P.HasVal e 1) (he' : Q.HasVal e 1) : P = Q

theorem **isGlue_iff** (q : Unif k) {L : Submodule 𝒪[k] (n → k)} {e : n → k} : IsGlue L e ↔ L.IsLattice k ∧ e ∉ L ∧ q.1 • e ∈ L

theorem **Pair.isGlue** {P : Pair k n} {e : n → k} (he : P.HasVal e 1) : IsGlue P.ker e

theorem **IsGlue.smul_mem_iff** {L : Submodule 𝒪[k] (n → k)} {e : n → k} (h : IsGlue L e) {s : 𝒪[k]} : s • e ∈ L ↔ s ∈ maximalIdeal 𝒪[k]

theorem **IsGlue.exists_pair** {L : Submodule 𝒪[k] (n → k)} {e : n → k} (h : IsGlue L e) : ∃ P : Pair k n, P.ker = L ∧ P.HasVal e 1

@[simp] theorem **glue_ker** (h : IsGlue L e) : (glue h).ker = L

theorem **glue_hasVal** (h : IsGlue L e) : (glue h).HasVal e 1

theorem **glue_lat** (h : IsGlue L e) : (glue h).lat = L ⊔ Submodule.span 𝒪[k] {e}

theorem **eq_glue_iff** (h : IsGlue L e) {P : Pair k n} : P = glue h ↔ P.ker = L ∧ P.HasVal e 1

theorem **Pair.glue_self** {P : Pair k n} (he : P.HasVal e 1) : glue (Pair.isGlue he) = P

theorem **glue_eq_glue_iff** (h : IsGlue L e) (h' : IsGlue L' e') : glue h = glue h' ↔ L = L' ∧ e' − e ∈ L

theorem **Pair.toGlue_bijective** : Function.Bijective (Pair.toGlue : Pair k n → GluePair k n)

@[simp] theorem **dia_lat** (c : 𝒦[k]ˣ) (P : Pair k n) : (c • P).lat = P.lat

theorem **dia_hasVal** {c : 𝒦[k]ˣ} {P : Pair k n} {x : n → k} {a : 𝒦[k]} : (c • P).HasVal x a ↔ P.HasVal x ((c⁻¹ : 𝒦[k]ˣ) * a)

@[simp] theorem **dia_ker** (c : 𝒦[k]ˣ) (P : Pair k n) : (c • P).ker = P.ker

theorem **dia_hasVal_one** {c : 𝒦[k]ˣ} {P : Pair k n} {e : n → k} (he : P.HasVal e 1) {s : 𝒪[k]} (hs : residue 𝒪[k] s = (c⁻¹ : 𝒦[k]ˣ)) : (c • P).HasVal (s • e) 1

theorem **hasVal_ev** {P : Pair k n} {x : n → k} (h : x ∈ P.lat) : P.HasVal x (P.ev x)

theorem **HasVal.ev_eq** {P : Pair k n} {x : n → k} {a : 𝒦[k]} (h : P.HasVal x a) : P.ev x = a

theorem **mk_surjective** : Function.Surjective (mk L)

theorem **mk_eq_zero** (q : Unif k) {x : L} : mk L x = 0 ↔ q.inv • (x : n → k) ∈ L

theorem **residue_smul** (s : 𝒪[k]) (x : Red L) : residue 𝒪[k] s • x = s • x

@[simp] theorem **mk_out** (x : Red L) : mk L x.out = x

theorem **finrank_eq** [Fintype n] [L.IsLattice k] : Module.finrank 𝒦[k] (Red L) = Fintype.card n

theorem **card_eq** [Fintype n] [L.IsLattice k] : Nat.card (Red L) = SplitWeil.resCard k ^ Fintype.card n

@[simp] theorem **Pair.red_mk** (P : Pair k n) (x : P.lat) : P.red (Red.mk _ x) = P.fn x

theorem **Pair.red_ne** (P : Pair k n) : P.red ≠ 0

@[simp] theorem **ofDual_lat** (f : {f : Module.Dual 𝒦[k] (Red L) // f ≠ 0}) : (ofDual L f).lat = L

theorem **ofDual_hasVal** (f : {f : Module.Dual 𝒦[k] (Red L) // f ≠ 0}) (x : L) : (ofDual L f).HasVal x (f.1 (Red.mk L x))

theorem **Pair.ofDual_red** (P : Pair k n) : ofDual P.lat ⟨P.red, P.red_ne⟩ = P

theorem **isGlue_inv_smul** {x : L} (hx : Red.mk L x ≠ 0) : IsGlue L (q.inv • (x : n → k))

@[simp] theorem **up_ker** (x : {x : Red L // x ≠ 0}) : (up L q x).ker = L

theorem **up_hasVal_iff** (x : {x : Red L // x ≠ 0}) (y : L) : (up L q x).HasVal (q.inv • (y : n → k)) 1 ↔ Red.mk L y = x.1

theorem **up_mk** {y : L} (hy : Red.mk L y ≠ 0) : up L q ⟨_, hy⟩ = glue (isGlue_inv_smul L q hy)

theorem **finite_lat** (L : Submodule 𝒪[k] (n → k)) : {P : Pair k n | P.lat = L}.Finite

theorem **finite_ker** (L : Submodule 𝒪[k] (n → k)) : {P : Pair k n | P.ker = L}.Finite

@[simp] theorem **mem_withLat** {L : Submodule 𝒪[k] (n → k)} {P : Pair k n} : P ∈ withLat L ↔ P.lat = L

@[simp] theorem **mem_withKer** {L : Submodule 𝒪[k] (n → k)} {P : Pair k n} : P ∈ withKer L ↔ P.ker = L

theorem **sum_withLat** {M : Type*} [AddCommMonoid M] (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] [Fintype {f : Module.Dual 𝒦[k] (Red L) // f ≠ 0}] (F : Pair k n → M) : ∑ P ∈ withLat L, F P = ∑ f, F (ofDual L f)

theorem **sum_withKer** {M : Type*} [AddCommMonoid M] (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (q : Unif k) [Fintype {x : Red L // x ≠ 0}] (F : Pair k n → M) : ∑ P ∈ withKer L, F P = ∑ x, F (up L q x)

theorem **cross_hasVal** (q : Unif k) {P₁ P : Pair k n} (h : P.lat = P₁.ker) {e : n → k} (he : P₁.HasVal e 1) : P.HasVal (q.1 • e) (cross q P₁ P)

theorem **cross_up** (q : Unif k) (P : Pair k n) (x : {x : Red P.lat // x ≠ 0}) : cross q (up P.lat q x) P = P.red x.1

theorem **cross_up_ofDual** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (x : {x : Red L // x ≠ 0}) (f : {f : Module.Dual 𝒦[k] (Red L) // f ≠ 0}) : cross q (up L q x) (ofDual L f) = f.1 x.1

theorem **cross_dia_right** (q : Unif k) {P₁ P : Pair k n} (h : P.lat = P₁.ker) (c : 𝒦[k]ˣ) : cross q P₁ (c • P) = c * cross q P₁ P

theorem **cross_dia_left** (q : Unif k) {P₁ P : Pair k n} (h : P.lat = P₁.ker) (c : 𝒦[k]ˣ) : cross q (c • P₁) P = (c⁻¹ : 𝒦[k]ˣ) * cross q P₁ P

theorem **mem_smul_iff** {g : GL n k} {L : Submodule 𝒪[k] (n → k)} {x : n → k} : x ∈ g • L ↔ g⁻¹ • x ∈ L

@[simp] theorem **smul_lat** (g : GL n k) (P : Pair k n) : (g • P).lat = g • P.lat

theorem **smul_hasVal** {g : GL n k} {P : Pair k n} {x : n → k} {a : 𝒦[k]} : (g • P).HasVal x a ↔ P.HasVal (g⁻¹ • x) a

theorem **HasVal.gl_smul** {P : Pair k n} {x : n → k} {a : 𝒦[k]} (h : P.HasVal x a) (g : GL n k) : (g • P).HasVal (g • x) a

@[simp] theorem **smul_ker** (g : GL n k) (P : Pair k n) : (g • P).ker = g • P.ker

theorem **smul_eq_iff** {g : GL n k} {P : Pair k n} {e : n → k} (he : P.HasVal e 1) : g • P = P ↔ g • P.ker = P.ker ∧ g • e - e ∈ P.ker

theorem **cross_smul** (q : Unif k) {P₁ P : Pair k n} (h : P.lat = P₁.ker) (g : GL n k) : cross q (g • P₁) (g • P) = cross q P₁ P

theorem **exists_gl_of_equiv** {L L' : Submodule 𝒪[k] (n → k)} [L.IsLattice k] [L'.IsLattice k] (e : L ≃ₗ[𝒪[k]] L') : ∃ g : GL n k, ∀ x : L, g • (x : n → k) = e x

theorem **Pair.exists_split** (P : Pair k n) : ∃ (K : Submodule 𝒪[k] P.lat) (e : (K × 𝒪[k]) ≃ₗ[𝒪[k]] P.lat), ∀ z, P.fn (e z) = residue 𝒪[k] z.2

theorem **mem_stdLat** {x : n → k} : x ∈ stdLat k n ↔ ∀ i, x i ∈ 𝒪[k]

theorem **stdPair_hasVal** {i : n} {x : n → k} {a : 𝒦[k]} : (stdPair k i).HasVal x a ↔ ∃ h : ∀ j, x j ∈ 𝒪[k], residue 𝒪[k] (x i, h i) = a

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsHasCls

theorem **HasCls.mem** {k : Type} [CommRing k] [Algebra F k] {J : Matrix (Fin n) (Fin n) E} {γ : HA k σ J} {q : Cls P k} (h : HasCls P k J γ q) : q ∈ XSet P k J

theorem **exists_hasCls** {k : Type} [CommRing k] [Algebra F k] {J : Matrix (Fin n) (Fin n) E} {q : Cls P k} (h : q ∈ XSet P k J) : ∃ γ, HasCls P k J γ q

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsHeightLeTwoMulQuotHeight

theorem **continuous_of_le_mul** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] {q h : G → ℝ} (hh : Tendsto h (𝓝 1) (𝓝 1)) (hpos : ∀ k, 0 < h k) (hinv : ∀ k, h k⁻¹ = h k) (hq : ∀ g k, q (g * k) ≤ q g * h k) : Continuous q

theorem **eventually_absFin_tmul_le_one** (x : E) : ∀ᶠ a in 𝓝 (0 : AF F), ∀ v, absFin F v (compFin F E v (a ⊗ₜ[F] x)) ≤ 1

theorem **eventually_absFin_le_one_nhds** : ∀ᶠ z in 𝓝 (0 : AF F ⊗[F] E), ∀ v, absFin F v (compFin F E v z) ≤ 1

theorem **continuous_infEmb_compInf** (w : InfinitePlace F) (τ : EmbOver E w) : Continuous fun z : AF F ⊗[F] E => infEmb τ (compInf F E w z)

theorem **continuous_opNorm** : Continuous (opNorm : Matrix n n ℂ → ℝ≥0)

theorem **continuous_htInf** (w : InfinitePlace F) : Continuous fun g : GL n (AF F ⊗[F] E) => htInf F w (bc (compInf F E w) g)

theorem **eventually_htFin_eq_one** : ∀ᶠ g in 𝒩 (1 : GL n (AF F ⊗[F] E)), ∀ v, htFin F v (bc (compFin F E v) g) = 1

theorem **continuous_heightGL** : Continuous (heightGL F : GL n (AF F ⊗[F] E) → ℝ≥0)

theorem **htInf_le_heightGL** (g : GL n (AF F ⊗[F] E)) (w : InfinitePlace F) : htInf F w (bc (compInf F E w) g) ≤ heightGL F g

theorem **IsQuotIntL1.zero** : IsQuotIntL1 μR μG 0 0

theorem **IsQuotIntL1.add** (h : IsQuotIntL1 μR μG f z) (h' : IsQuotIntL1 μR μG f' z') : IsQuotIntL1 μR μG (f + f') (z + z')

theorem **IsQuotIntL1.smul** (c : ℂ) (h : IsQuotIntL1 μR μG f z) : IsQuotIntL1 μR μG (c • f) (c * z)

theorem **IsQuotIntL1.conj** (h : IsQuotIntL1 μR μG f z) : IsQuotIntL1 μR μG (fun g => starRingEnd ℂ (f g)) (starRingEnd ℂ z)

theorem **IsQuotIntL1.mul_right** [MeasurableMul G] [μG.IsMulRightInvariant] (h : IsQuotIntL1 μR μG f z) (k : G) : IsQuotIntL1 μR μG (fun g => f (g * k)) z

theorem **IsQuotIntL1.left** [MeasurableMul R] [μR.IsMulRightInvariant] (h : IsQuotIntL1 μR μG f z) {s : G} (hs : s ∈ R) (x : G) : f (s * x) = f x

theorem **integrable_of_lintegral_eq_one** {α : Type} [MeasurableSpace α] {μ : Measure α} {b : α → ℝ≥0} (hm : Measurable b) (h : ∫⁻ x, b x ∂μ = 1) : Integrable (fun x => (b x : ℝ)) μ ∧ ∫ x, (b x : ℝ) ∂μ = 1

theorem **IsQuotIntL1.cutoff** [SFinite μG] [μG.IsMulLeftInvariant] [SFinite μR] [μR.IsInvInvariant] (h : IsQuotIntL1 μR μG f z) {β : G → ℝ≥0} (hβ : IsCosetCutoff R μR β) : Integrable (fun x => (β x : ℝ) • f x) μG ∧ z = ∫ x, (β x : ℝ) • f x ∂μG

theorem **IsQuotIntL1.unique** (hR : IsClosed (R : Set G)) (h : IsQuotIntL1 μR μG f z) (h' : IsQuotIntL1 μR μG f z') : z = z'

theorem **isQuotIntL1_iff** (hR : IsClosed (R : Set G)) (hm : Measurable f) : IsQuotIntL1 μR μG f z ↔ (∀ s ∈ R, ∀ x, f (s * x) = f x) ∧ QuotIntegrable R μG μR f ∧ z = quotInt R μG μR f

theorem **one_le_height** (g : GA D V) : 1 ≤ height V g

theorem **height_pos** (g : GA D V) : 0 < height V g

@[simp] theorem **height_one** : height V 1 = 1

@[simp] theorem **height_inv** (g : GA D V) : height V g⁻¹ = height V g

theorem **height_mul_le** (g h : GA D V) : height V (g * h) ≤ height V g * height V h

theorem **htInf_le_height** (g : GA D V) (w : InfinitePlace D.host.F) : (htInf D.host.F w (bc (compInf D.host.F D.host.E w) (Formalism.L2.Theta.matHom V g)) : ℝ) ≤ height V g

theorem **continuous_height** : Continuous (height V)

theorem **exists_height_le_of_isCompact** {K : Set (GA D V)} (hK : IsCompact K) : ∃ C, ∀ g ∈ K, height V g ≤ C

theorem **quotHeight_le** (g : GA D V) {γ : GA D V} (hγ : γ ∈ GQ V) : quotHeight V g ≤ height V (γ * g)

theorem **quotHeight_le_height** (g : GA D V) : quotHeight V g ≤ height V g

theorem **le_quotHeight_iff** {g : GA D V} {c : ℝ} : c ≤ quotHeight V g ↔ ∀ γ ∈ GQ V, c ≤ height V (γ * g)

theorem **one_le_quotHeight** (g : GA D V) : 1 ≤ quotHeight V g

theorem **quotHeight_pos** (g : GA D V) : 0 < quotHeight V g

theorem **quotHeight_left** {γ : GA D V} (hγ : γ ∈ GQ V) (g : GA D V) : quotHeight V (γ * g) = quotHeight V g

theorem **quotHeight_of_mem** {γ : GA D V} (hγ : γ ∈ GQ V) : quotHeight V γ = 1

theorem **quotHeight_mul_le** (g h : GA D V) : quotHeight V (g * h) ≤ quotHeight V g * height V h

theorem **continuous_quotHeight** : Continuous (quotHeight V)

theorem **exists_height_le_two_mul_quotHeight** (g : GA D V) : ∃ γ ∈ GQ V, height V (γ * g) ≤ 2 * quotHeight V g

theorem **modGrowth_iff_exp** {f : GA D V → X} : ModGrowth V f ↔ ∃ A, ModGrowthExp V A f

theorem **nonneg_of_growth** {f : GA D V → X} {C : ℝ} {A : ℕ} (h : ∀ g, ‖f g‖ ≤ C * height V g ^ A) : 0 ≤ C

theorem **ModGrowthExp.mono** {f : GA D V → X} {A B : ℕ} (h : ModGrowthExp V A f) (hAB : A ≤ B) : ModGrowthExp V B f

theorem **ModGrowthExp.add** {f f' : GA D V → X} {A : ℕ} (h : ModGrowthExp V A f) (h' : ModGrowthExp V A f') : ModGrowthExp V A (f + f')

theorem **ModGrowthExp.mul** {f f' : GA D V → ℂ} {A B : ℕ} (h : ModGrowthExp V A f) (h' : ModGrowthExp V B f') : ModGrowthExp V (A + B) (f * f')

theorem **ModGrowthExp.translate** {f : GA D V → X} {A : ℕ} (h : ModGrowthExp V A f) (a b : GA D V) : ModGrowthExp V A fun g => f (a * g * b)

theorem **norm_le_quotHeight_pow** {f : GA D V → X} (hf : ∀ γ ∈ GQ V, ∀ g, f (γ * g) = f g) {C : ℝ} {A : ℕ} (h : ∀ g, ‖f g‖ ≤ C * height V g ^ A) (g : GA D V) : ‖f g‖ ≤ C * 2 ^ A * quotHeight V g ^ A

theorem **ModGrowth.of_bounded** {f : GA D V → X} {C : ℝ} (h : ∀ g, ‖f g‖ ≤ C) : ModGrowth V f

theorem **ModGrowth.of_le** {Y : Type} [NormedAddCommGroup Y] {f : GA D V → X} {f' : GA D V → Y} (h : ModGrowth V f) (hle : ∀ g, ‖f' g‖ ≤ ‖f g‖) : ModGrowth V f'

theorem **ModGrowth.add** {f f' : GA D V → X} (h : ModGrowth V f) (h' : ModGrowth V f') : ModGrowth V (f + f')

theorem **ModGrowth.mul** {f f' : GA D V → ℂ} (h : ModGrowth V f) (h' : ModGrowth V f') : ModGrowth V (f * f')

theorem **ModGrowth.translate** {f : GA D V → X} (h : ModGrowth V f) (a b : GA D V) : ModGrowth V fun g => f (a * g * b)

theorem **ModGrowth₂.right** {f : GA D V → GA D W → X} (h : ModGrowth₂ V W f) (g : GA D V) : ModGrowth W (f g)

theorem **ModGrowth₂.swap** {f : GA D V → GA D W → X} (h : ModGrowth₂ V W f) : ModGrowth₂ W V fun h g => f g h

theorem **ModGrowth₂.left** {f : GA D V → GA D W → X} (h : ModGrowth₂ V W f) (k : GA D W) : ModGrowth V fun g => f g k

theorem **rapidDec_exp_neg** : RapidDec V fun g => (Real.exp (-quotHeight V g) : ℂ)

theorem **decaySize_le_iff** {N : ℕ} {Φ : GA D V → X} {C : ℝ} (hC : 0 ≤ C) : decaySize V N Φ ≤ ENNReal.ofReal C ↔ ∀ g, ‖Φ g‖ * quotHeight V g ^ N ≤ C

theorem **rapidDec_iff_decaySize** {Φ : GA D V → X} : RapidDec V Φ ↔ (∀ γ ∈ GQ V, ∀ g, Φ (γ ∗ g) = Φ g) ∧ Continuous Φ ∧ ∀ N, decaySize V N Φ ≠ ∞

theorem **RapidDec.left** {Φ : GA D V → X} (h : RapidDec V Φ) {γ : GA D V} (hγ : γ ∈ GQ V) (g : GA D V) : Φ (γ ∗ g) = Φ g

theorem **RapidDec.continuous** {Φ : GA D V → X} (h : RapidDec V Φ) : Continuous Φ

theorem **RapidDec.decay** {Φ : GA D V → X} (h : RapidDec V Φ) (N : ℕ) : ∃ C : ℝ, ∀ g, ‖Φ g‖ ∗ quotHeight V g ^ N ≤ C

theorem **RapidDec.bounded** {Φ : GA D V → X} (h : RapidDec V Φ) : ∃ C : ℝ, ∀ g, ‖Φ g‖ ≤ C

theorem **RapidDec.of_cover** {Φ : GA D V → X} (hl : ∀ γ ∈ GQ V, ∀ g, Φ (γ ∗ g) = Φ g) (hc : Continuous Φ) {S : Set (GA D V)} (hS : ∀ g, ∃ γ ∈ GQ V, γ ∗ g ∈ S) (h : ∀ N : ℕ, ∃ C : ℝ, ∀ g ∈ S, ‖Φ g‖ ∗ height V g ^ N ≤ C) : RapidDec V Φ

theorem **RapidDec.of_isCompact** {Φ : GA D V → X} (hl : ∀ γ ∈ GQ V, ∀ g, Φ (γ ∗ g) = Φ g) (hc : Continuous Φ) {K : Set (GA D V)} (hK : IsCompact K) (hS : ∀ g, ∃ γ ∈ GQ V, γ ∗ g ∈ K) : RapidDec V Φ

theorem **RapidDec.zero** : RapidDec V (0 : GA D V → X)

theorem **RapidDec.add** {Φ Ψ : GA D V → X} (h : RapidDec V Φ) (h' : RapidDec V Ψ) : RapidDec V (Φ + Ψ)

theorem **RapidDec.of_le** {Y : Type} [NormedAddCommGroup Y] {Φ : GA D V → X} {Ψ : GA D V → Y} (h : RapidDec V Φ) (hl : ∀ γ ∈ GQ V, ∀ g, Ψ (γ ∗ g) = Ψ g) (hc : Continuous Ψ) (hle : ∀ g, ‖Ψ g‖ ≤ ‖Φ g‖) : RapidDec V Ψ

theorem **RapidDec.conj** {Φ : GA D V → ℂ} (h : RapidDec V Φ) : RapidDec V fun g => starRingEnd ℂ (Φ g)

theorem **RapidDec.mul_right** {Φ : GA D V → X} (h : RapidDec V Φ) (k : GA D V) : RapidDec V fun g => Φ (g ∗ k)

theorem **RapidDec.modGrowth_mul** {f Φ : GA D V → ℂ} (h : RapidDec V Φ) (hf : ModGrowth V f) (hl : ∀ γ ∈ GQ V, ∀ g, f (γ ∗ g) = f g) (hc : Continuous f) : RapidDec V (f ∗ Φ)

theorem **RapidDec.smul** {Φ : GA D V → ℂ} (c : ℂ) (h : RapidDec V Φ) : RapidDec V (c • Φ)

theorem **RapidDec.integrableOn** {Φ : GA D V → ℂ} (h : RapidDec V Φ) {μ : Measure (GA D V)} {S : Set (GA D V)} (hS : μ S ≠ ∞) : IntegrableOn Φ S μ

theorem **isQuotIntL1_bot** {G : Type} [Group G] [MeasurableSpace G] {μG : Measure G} {f : G → ℂ} {z : ℂ} : IsQuotIntL1 (R

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsIsSph

theorem **exists_isSph** {ε : Fin n → Bool} {b : ℕ} {W₀ : Flag n b} (h : IsMinimalFlag ε W₀) (lam : Fin b → ℝ) : ∃ φ, IsSph (ε

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsMatchOver

theorem **pairVec_colOf** (ξ : Module.Dual ℂ (coefPlus G θ).W) (Ψ : GLf G → WtSp G) : pairVec (coefPlus G θ) ξ (colOf θ Ψ) = scal θ (ξ ∘ₗ (GlobalTheory.CoefModel.toHilb (LaP G) (posDef_dual G)).toLinearMap) Ψ

theorem **fin_mul** (γ : GQ (Xp G)) (γE : UR D.sp.cm (Xp G).gram) (hm : Bridge.IsRatMat (Xp G) γ γE.1) (g k : Bridge.Gf (Xp G)) : Formalism.UG.bc (Lev.swapAE D) (Bridge.ratF (Xp G) γ ∗ g ∗ k).1 = ratIn G γE ∗ Formalism.UG.bc (Lev.swapAE D) g.1 ∗ Formalism.UG.bc (Lev.swapAE D) k.1

theorem **map_rat** (hθ : IsInfFam θ) (V : HSp D) (lam : D.what₂.Tau → DomWeight V.rank) (γ : GQ V) (γE : GL (Fin V.rank) D.host.E) (h : Bridge.IsRatMat V γ γE) (v : VV D.what₂.LL lam) : GlobalTheory.CoefModel.mapVV D.what₂.embLL lam (rhoVr (labLL D) lam γE v) = rhoK lam (invTrK (GlobalTheory.CoefModel.realGL θ V γ)) (GlobalTheory.CoefModel.mapVV D.what₂.embLL lam v)

theorem **colOf_mem** (hθ : IsInfFam θ) {Ψ : GLf G → WtSp G} (hl : ∀ γ g, Ψ (ratIn G γ ∗ g) = rhoW G γ (Ψ g)) (hr : ∀ g, ∀ k ∈ G.Kplus, Ψ (g ∗ k) = Ψ g) : colOf θ Ψ ∈ Cols G θ

theorem **dnFrom_dnVec** (h : R.chain_ioY) (t : T) : ∀ n, n ≤ D.R.m − 1 → ∀ (F : R.CC t n) (v : SmT (R.MY n)), (Lift.dnFrom (R.CC t) (R.ioY t) n F).1 (d.dnVec n v) = d.dnFun t n (F.1 v) | 0, _, _, _ => rfl | n + 1, hn, F, v => (dnFrom_dnVec h t n (by omega) (R.ioY t n F) (R.ItY n (d.polY n) v)).trans (congrArg (d.dnFun t n) (funext fun g => ((h t n hn).eq F (d.polY n) v g).symm)) theorem thDn_w (h : R.IsChain) (t : T) (x : (R.HX t).H) : R.thDn t x d.w = d.lift t x

theorem **rowForm_Th** (h : R.IsChain) (t : T) (x : (R.HX t).H) : rowForm R d.w (R.HE.e (R.Th t x)) = R.e0 (d.lift t x)

theorem **guard_of_qGuard** {R : RealDat D A Ks} (h : Formalism.IndefR.FWT.QGuard R) : Guard R

theorem **guard_of_top** (R : RealDat D A Ks) (h : D.host.ccK = ⊤) : Guard R

theorem **form_apply** (u : R.HE.H) : ch.form u = rowForm R ch.w (R.HE.e u)

theorem **IsBase.withReal** (h : ch.IsBase) (ρ ξ) : (ch.withReal ρ ξ).IsBase

theorem **IsBase.c_eq** (h : ch.IsBase) {i j t} (h0 : SquareNonzero ch.W ch.S.B ch.rm t i j) {c : ℂ} (hc : IsSquareConst ch.W ch.S.B ch.rm t i j c) : ch.c i j t = c

theorem **psiK_eq** (a : Fin ch.n) (u : R.HE.H) : psiK ch a u = Bplus ch.W ch.M (scal ch.W.θ (ch.ξ ∘ₗ (GlobalTheory.CoefModel.toHilb (LaP ch.G) (posDef_dual ch.G)).toLinearMap) (ch.Psi a)) (thetaChi ch.W ch.M ch.ρ (ch.Phi a) (ch.form u))

theorem **col_mem** (a : Fin ch.n) : ch.col a ∈ Cols ch.G ch.W.θ

theorem **psiK_smul** (a : Fin ch.n) (z : ℂ) (u : R.HE.H) : psiK ch a (z • u) = z ∗ psiK ch a u

theorem **psiK_zero** (a : Fin ch.n) : psiK ch a 0 = 0

theorem **psiK_add** {a : Fin ch.n} {u u' : R.HE.H} (h : ConvK ch a u) (h' : ConvK ch a u') : psiK ch a (u + u') = psiK ch a u + psiK ch a u'

theorem **form_Th** (h : R.IsChain) (t : T) (x : (R.HX t).H) : ch.form (R.Th t x) = R.e0 (ch.cv.lift t x)

theorem **realOK_of_inj** (h : ∀ u, (∀ a, psiK ch a u = 0) → u = 0) : ch.RealOK

theorem **Choice.RealOK.inj** (h : ch.RealOK) {ρ' ξ'} (h' : ∀ u, (∀ a, psiK (ch.withReal ρ' ξ') a u = 0) → u = 0) (u : R.HE.H) (hu : ∀ a, psiK ch a u = 0) : u = 0

theorem **Choice.IsBase.is_withReal** (h : ch.IsBase) {ρ' ξ'} (h' : ∀ u, (∀ a, psiK (ch.withReal ρ' ξ') a u = 0) → u = 0) : (ch.withReal ρ' ξ').Is

theorem **cTot_ne_zero_iff** (t : T) : cTot ch t ≠ 0 ↔ ∀ i j, ch.c i j t ≠ 0

theorem **ratio_eq_prod** (t : T) : ratio ch t = ∏ i, ∏ j, sqRatio ch i j t

theorem **ratio_base** (h : cTot ch A.μ0 ≠ 0) : ratio ch A.μ0 = 1

theorem **psiN_eq_iff** (h : cTot ch A.μ0 ≠ 0) (κ : ℂˣ) (z v : ℂ) (t : T) (a : Fin ch.n) (u : R.HE.H) : psiN ch κ a u = (z ∗ ratio ch t)⁻¹ ∗ v ↔ psiK ch a u = (z ∗ cTot ch t)⁻¹ ∗ κ ∗ v

theorem **vK_smul** (t : T) (z : D.what₂.LL) (x : A.fam.amb.FR t) (a : Fin ch.n) : vK ch t (z • x) a = D.what₂.embLL z * vK ch t x a

theorem **vK_add** (t : T) {a : Fin ch.n} (h : ch.S.SumOK t (ch.Phi a)) (x x' : A.fam.amb.FR t) : vK ch t (x + x') a = vK ch t x a + vK ch t x' a

theorem **exists_match_over** (hL : R.toLevCh.Is A.μ0) (hch : R.IsChain) {sc : Consts} {G : GridData R.toLevCh sc} {W : G.Weil} (hW : W.Over R) : ∃ rm : RealMatch W, rm.Over R

theorem **exists_choice** (hL : R.toLevCh.Is A.μ0) (hch : R.IsChain) {sc : Consts} {G : GridData R.toLevCh sc} {W : G.Weil} (hW : W.Over R) (S : SumChoice W Set.univ) (cv : ChainVec R) : ∃ M rm rp n e, 0 < n ∧ M.IsRow ∧ rm.Over R ∧ ∀ ρ ξ c, Guard R → cv.w ≠ 0 → (∀ i j t, IsSquareConst W S.B rm t i j (c i j t)) → ((sc, G, W, M, ρ, ξ, cv, rm, S, rp, n, e, c) : Choice R).IsBase

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsRealFnNeZero

theorem **realFr_apply** (g : GA D V) : realFr θ V B g = B⁻¹ * realGL θ V g * B

theorem **realFr_coe** (g : GA D V) (τ : D.what₂.Tau) : (realFr θ V B g τ : Matrix (Fin V.rank) (Fin V.rank) ℂ) = ((B τ)⁻¹ : GL (Fin V.rank) ℂ).1 * Formalism.L2.Pj.archM V (θ τ) g * (B τ).1

theorem **realFr_one** : realFr θ V 1 = realGL θ V

theorem **rhoFr_apply** (g : GA D V) : rhoFr θ V lam B g = rhoK lam (invTrK (realFr θ V B g))

theorem **rhoFr_one** (g : GA D V) : rhoFr θ V lam 1 g = rhoK lam (invTrK (realGL θ V g))

theorem **rhoK_rhoFr** (g : GA D V) (v : VV ℂ lam) : rhoK lam (invTrK B) (rhoFr θ V lam B g v) = rhoFr θ V lam 1 g (rhoK lam (invTrK B) v)

theorem **realFr_fin** {g : GA D V} (hg : g ∈ GfC V) : realFr θ V B g = 1

theorem **rhoFr_fin** {g : GA D V} (hg : g ∈ GfC V) : rhoFr θ V lam B g = 1

theorem **realFr_toInf** (g : GA D V) : realFr θ V B (toInf V g) = realFr θ V B g

theorem **exists_realFr_eq** (hB : ∀ τ, (B τ).1ᴴ * V.gram.map (tauE D τ) * (B τ).1 = 1) (u : D.what₂.Tau → GL (Fin V.rank) ℂ) (hu : ∀ τ, (u τ).1 ∈ Matrix.unitaryGroup (Fin V.rank) ℂ) : ∃ k : GinfC V, realFr θ V B k = u

theorem **continuous_realFr** (τ : D.what₂.Tau) : Continuous fun g => (realFr θ V B g τ : Matrix (Fin V.rank) (Fin V.rank) ℂ)

theorem **continuous_rhoFr** (Λ : Module.Dual ℂ (VV ℂ lam)) (v : VV ℂ lam) : Continuous fun g => Λ (rhoFr θ V lam B g v)

theorem **rhoAt_apply** (g : GA D (G.V X)) : rhoAt θ N X g = rhoK (wt G X) (invTrK (realAt θ N X g))

theorem **rhoVr_rhoAt** (g : GA D (G.V X)) (v : VV ℂ (wt G X)) : rhoVr (tauE D) (wt G X) (N.fr X) (rhoAt θ N X g v) = rhoFr θ _ (wt G X) 1 g (rhoVr (tauE D) (wt G X) (N.fr X) v)

@[simp] theorem **rhoAt_fin** (y : FinPt G X) : rhoAt θ N X y = 1

theorem **frC_on** (hX : DefNode D.R.m X) (τ : D.what₂.Tau) : (N.frC X τ)ᴴ * (G.V X).gram.map (tauE D τ) * N.frC X τ = 1

theorem **realFn_apply** (g : GA D (G.V X)) : realFn θ N X Λ V g = Λ (rhoAt θ N X g⁻¹ (V (toFin _ g)))

@[simp] theorem **realFn_fin** (y : FinPt G X) : realFn θ N X Λ V y = Λ (V y)

theorem **realFn_one** : realFn θ N X Λ V 1 = Λ (V 1)

theorem **realFn_mul** (g h : GA D (G.V X)) : realFn θ N X Λ V (g * h) = realFn θ N X (Λ ∘ₗ rhoAt θ N X h⁻¹) (fun y => V (y * toFin _ h)) g

theorem **realFn_mul_fin** (g : GA D (G.V X)) (k : FinPt G X) : realFn θ N X Λ V (g * k) = realFn θ N X Λ (fun y => V (y * k)) g

theorem **exists_realFn_ne_zero** : ∃ (Λ : Module.Dual ℂ (VV ℂ (wt G X))) (v : VV ℂ (wt G X)), realFn θ N X Λ (fun _ => v) 1 ≠ 0

theorem **exists_realAt_eq** (hX : DefNode D.R.m X) (u : D.what₂.Tau → GL (Fin (G.V X).rank) ℂ) (hu : ∀ τ, (u τ).1 ∈ Matrix.unitaryGroup (Fin (G.V X).rank) ℂ) : ∃ k : GinfC (G.V X), realAt θ N X k = u

theorem **continuous_rhoAt** (v : VV ℂ (wt G X)) : Continuous fun g => Λ (rhoAt θ N X g v)

theorem **unitaryEmbedding_mul** (h h' : GA D A) : stabilizerEmbedding d (h * h') = stabilizerEmbedding d h * stabilizerEmbedding d h'

theorem **Emb_map_eq_one** {R : Type} [CommRing R] (π : AE D →+* R) {h : GA D A} (hy : (matHom A h).1.map π = 1) : (matHom B (stabilizerEmbedding d h)).1.map π = 1

theorem **Emb_mem_inf** {k : GA D A} (hk : k ∈ GinfC A) : stabilizerEmbedding d k ∈ GinfC B

theorem **Emb_fin** (y : GfC A) : stabilizerEmbedding d y = finEmb d y

theorem **archM_Emb** (h : GA D A) (τ : D.what₂.Tau) : Formalism.L2.Pj.archM B (θ τ) (stabilizerEmbedding d h) * (frameMat B A d.ι).map (tauE D τ) = (frameMat B A d.ι).map (tauE D τ) * Formalism.L2.Pj.archM A (θ τ) h ∧ Formalism.L2.Pj.archM B (θ τ) (stabilizerEmbedding d h) * d.a.map (tauE D τ) = d.a.map (tauE D τ)

theorem **isBlock_conj** {K : Type} [Field K] {N n : ℕ} (hN : N = n + 1) (b' : GL (Fin N) K) (b : GL (Fin n) K) {g' : Matrix (Fin N) (Fin N) K} {g : Matrix (Fin n) (Fin n) K} {P : Matrix (Fin N) (Fin n) K} {z : Matrix (Fin N) (Fin 1) K} (h1 : ∀ i k, b'.1 i (Formalism.L2.Col.Br.iC hN k) = (P * b.1) i k) (h2 : ∀ i, b'.1 i (Formalism.L2.Col.Br.iL hN) = z i 0) (hP : g' * P = P * g) (hz : g' * z = z) : GlobalTheory.BranchFormula.IsBlock hN ((b'⁻¹).1 * g' * b'.1) ((b⁻¹).1 * g * b.1) 1

theorem **isBlock_realAt** (h : GA D (G.V (.grid D.R.m D.R.m))) (τ : D.what₂.Tau) : GlobalTheory.BranchFormula.IsBlock G.sp.topDec.rank.symm (realAt θ N (.top D.R.m) (stabilizerEmbedding G.sp.topDec h) τ).1 (realAt θ N (.grid D.R.m D.R.m) h τ).1 1

theorem **isBlock_invTrK_realAt** (h : GA D (G.V (.grid D.R.m D.R.m))) (τ : D.what₂.Tau) : GlobalTheory.BranchFormula.IsBlock G.sp.topDec.rank.symm (invTrK (realAt θ N (.top D.R.m) (stabilizerEmbedding G.sp.topDec h)) τ).1 (invTrK (realAt θ N (.grid D.R.m D.R.m) h) τ).1 1

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExistsStdIsVacVec

theorem **exists_isVacVec** {V W : HSp D} {S S' : D.what₂.Tau → SPar} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (M : DSModel V S) (M' : DSModel W S') : ∃ f, IsVacVec ψ θ M M' f

theorem **exists_std_isVacVec** {V W : HSp D} {S S' : D.what₂.Tau → SPar} (M : DSModel V S) (M' : DSModel W S') : ∃ ψ θ f, IsStdChar D ψ (-1) ∧ IsInfFam θ ∧ IsVacVec ψ θ M M' f

theorem **not_isVacVec_zero** {V W : HSp D} {S S' : D.what₂.Tau → SPar} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (M : DSModel V S) (M' : DSModel W S') : ¬ IsVacVec ψ θ M M' 0

theorem **exp_of_even** {k : Fin D.R.m} (hk : Even (k : ℕ)) (τ : D.what₂.Tau) : expS D k τ = -(D.hc τ).e k - 1 / 2 ∧ expT D k τ = -(D.hc τ).e k

theorem **exp_of_odd** {k : Fin D.R.m} (hk : ¬ Even (k : ℕ)) (τ : D.what₂.Tau) : expS D k τ = -(D.hc τ).e k ∧ expT D k τ = -(D.hc τ).e k - 1 / 2

theorem **hasInfExp_div** {χ η : (AE D)ˣ →* Circle} {τ : D.what₂.Tau} {a b : ℚ} (hχ : HasInfExp D χ τ a) (hη : HasInfExp D η τ b) : HasInfExp D (χ * η⁻¹) τ (a - b)

theorem **isHeckeChar_div** {χ η : (AE D)ˣ →* Circle} (hχ : IsHeckeChar D χ) (hη : IsHeckeChar D η) : IsHeckeChar D (χ * η⁻¹)

theorem **exists_isStepWeil** (hW : GlobalTheory.Published.WeilRepresentationUnitaryPair) (k : Fin D.R.m) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (V W : HSp D) (hV : V.rank % 2 = ((k : ℕ) + 1) % 2) (hW' : W.rank % 2 = (k : ℕ) % 2) (μ η : (AE D)ˣ →* Circle) (hμ : IsHeckeChar D μ) (hμe : ∀ τ, HasInfExp D μ τ (-(D.hc τ).e k)) (hμr : ResEta D μ 0) (hη : IsHeckeChar D η) (hηe : ∀ τ, HasInfExp D η τ (1 / 2)) (hηr : ResEta D η 1) : ∃ ω, IsStepWeil k ψ V W ω

theorem **rank_parity** (D : D7) (k : ℕ) : (D.R.r + k * D.R.t) % 2 = (k + 1) % 2 ∧ (D.R.r + (k + 1) * D.R.t) % 2 = k % 2

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ExtendByZeroTrivialDecomposition

theorem **stab_compact_open** (W : HSp D) (β : GL (Fin W.rank) (Lev.EAf D)) : IsCompact (Lev.levA W (Formalism.L2.DIV.St β) : Set (GA D W)) ∧ IsOpen (((Lev.levA W (Formalism.L2.DIV.St β)).subgroupOf (GfC W) : Subgroup (GfC W)) : Set (GfC W))

theorem **isHalf_mul** {k l m : ℕ} {e : AE D} (p : Mx D k l) {u : Mx D l m} (hu : IsHalf e u) : IsHalf e (p · u)

theorem **gram_mul** {n k l k' l' : ℕ} (M : Matrix (Fin n) (Fin n) D.host.E) (a : Matrix (Fin n) (Fin k) D.host.E) (b : Matrix (Fin n) (Fin l) D.host.E) (u : Matrix (Fin k) (Fin k') D.host.E) (v : Matrix (Fin l) (Fin l') D.host.E) : (a * u)ᴴ * M * (b * v) = uᴴ * (aᴴ * M * b) * v

theorem **xiPow_ne** (D : D7) (n : ℕ) : (-D.host.xi0) ^ n ≠ 0

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.FiniteDimensionalOfTrivialIsAdmOn

theorem **isLinRed_units** (n : ℕ) : IsLinRed {g : Matrix (Fin n) (Fin n) ℂ | IsUnit g}

theorem **not_isLinRed_univ** : ¬ IsLinRed (Set.univ : Set (Matrix (Fin 1) (Fin 1) ℂ))

theorem **finiteDimensional_of_trivial_isAdmOn** {G : Set (Matrix (Fin n) (Fin n) ℂ)} (h : IsAdmOn G (fun _ => (1 : H →L[ℂ] H))) : FiniteDimensional ℂ H

theorem **analyticOnGroup_entry** (G : Set (Matrix (Fin n) (Fin n) ℂ)) (i j : Fin n) : AnalyticOnGroup G fun g => g i j

theorem **continuousOn_of_analyticOnGroup** {G : Set (Matrix (Fin n) (Fin n) ℂ)} {f : Matrix (Fin n) (Fin n) ℂ → ℂ} (h : AnalyticOnGroup G f) : ContinuousOn f G

theorem **isClosed_maxCpt** {G : Set (Matrix (Fin n) (Fin n) ℂ)} (h : IsLinRed G) : IsClosed (maxCpt G)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.GenInhabited

theorem **cjOm_apply** {S T : HSp D} (om : Om (negSp S) T) (p : GA D S × GA D T) (f : WO S T → ℂ) : cjOm om p f = cj id (om (negGA S p.1, p.2) (cj id f))

theorem **negGA_mem_GQ** (V : HSp D) (γ : GA D V) (h : γ ∈ GQ V) : negGA V γ ∈ GQ (negSp V)

theorem **weil_cjOm** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om (negSp S) T) (hw : IsWeilDatumO ψ (negSp S) T χS χT om) : IsWeilDatumO ψ S T χS⁻¹ χT⁻¹ (cjOm om)

theorem **vac_star** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (Γ : AddSubgroup (WO S T)) (f : WO S T → ℂ) : VacG ψ (negSp S) T Γ f ↔ VacG ψ S T Γ (star f)

theorem **part_succ** {T : Type} {L : LevCh D T} {sc : Consts} {G : GridData L sc} (W : G.Weil) (M : GlobalTheory.FunObj.Vol G) (ρ Φ u) (i : ℕ) (hi : i < D.R.m) : part W M ρ Φ u (i + 1) hi = liftA W M (.right i ∅) (W.datum ρ Φ (i, hi)) (part W M ρ Φ u i hi.le)

theorem **part_congr** {T : Type} {L : LevCh D T} {sc : Consts} {G : GridData L sc} (W : G.Weil) (M : GlobalTheory.FunObj.Vol G) (ρ Φ ρ' Φ' u) (i : ℕ) (hi : i ≤ D.R.m) (h : ∀ j : Fin D.R.m, j.1 < i → ρ j = ρ' j ∧ Φ j = Φ' j) : part W M ρ Φ u i hi = part W M ρ' Φ' u i hi

theorem **conjRead** : ConjRead

theorem **partFixed** (h1 : DepFixed) (h2 : DatumDep) : PartFixed

theorem **exists_gen** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} (B : BesLev D ρ) {W W₀ : HSp D} (dW : HypDec W W₀ ρ) (n : ℕ) (g : Fin n → GA D B.V₀) : ∃ a : Gen ψ B.V B.V₀ W ρ (Fin n), a.Is ∧ a.g = g ∧ a.m = Measure.count ∧ a.x₀ = dW.e ∧ ∀ y, ∀ x ∈ flagN a.d, ‖a.Th y x‖ = 1

theorem **gen_inhabited** (D : D7) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W : HSp D) (a : Gen ψ V V₀ W 1 (Fin 2)), a.Is ∧ V₀.rank = 2 ∧ W.rank = 3 ∧ a.m = Measure.count ∧ ∀ y, ∀ x ∈ flagN a.d, ‖a.Th y x‖ = 1

theorem **lhs_count** {ψ : AddChar (AF D.host.F) ℂ} {V V₀ W : HSp D} {ρ n : ℕ} (a : Gen ψ V V₀ W ρ (Fin n)) (h : a.m = Measure.count) (G Φ) : a.lhs G Φ = ∑ y, ∫ x in a.vN.𝒩, a.Th y x.1 * liftS a.om a.vW.μ a.vW.ℱ G Φ (x.1 * a.d.emb0 (a.g y)) ∂a.vN.ν

theorem **strip_stable** (W : HSp D) (R : Subgroup (GA D W)) (C : Set (GA D W)) {h r b : GA D W} (hh : h ∈ strip W R C) (hr : r ∈ R) (hb : b ∈ GfC W) : r * h * b ∈ strip W R C

theorem **top_of_rungs** (h0 : Rung0) (hB : RungB) (hJ : RungJ) : OnTower fun {_} _ _ _ {G} W M v F Ψ u _ => per0 G M (Ψ 0) u ≠ 0 → ∀ k h, NZB W M v F Ψ u k h

theorem **not_nzb** {T : Type} {L : LevCh D T} {sc : Consts} {G : GridData L sc} (W : G.Weil) (M : GlobalTheory.FunObj.Vol G) (v : CoreVol G) (F : Frames G) (u k h) : ¬ NZB W M v F (fun _ => 0) u k h

theorem **not_nzj** {T : Type} {L : LevCh D T} {sc : Consts} {G : GridData L sc} (W : G.Weil) (M : GlobalTheory.FunObj.Vol G) (v : CoreVol G) (F : Frames G) (u ΓL k h) : ¬ NZJ W M v F (fun _ => 0) u ΓL k h

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.HasConductorMulShift

@[simp] theorem **dia_apply** (c : 𝓀[k]ˣ) (φ : Form k n M) (P : Pair k n) : dia c φ P = φ (c • P)

theorem **sumOp_apply** (S : Pair k n → Finset (Pair k n)) (w : Pair k n → Pair k n → ℂ) (P : Pair k n) : sumOp S w φ P = ∑ Q ∈ S P, w P Q • φ Q

theorem **Nm_apply** [Fintype 𝓀[k]ˣ] (P : Pair k n) : Nm φ P = ∑ c : 𝓀[k]ˣ, φ (c • P)

theorem **mem_Mram** [Fintype 𝓀[k]ˣ] : φ ∈ Mram k n M ↔ ∀ P, ∑ c : 𝓀[k]ˣ, φ (c • P) = 0

theorem **Gs_apply** [Fintype 𝓀[k]ˣ] (P : Pair k n) : Gs χ φ P = ∑ c : 𝓀[k]ˣ, χ c • φ (c • P)

theorem **GsStar_apply** [Fintype 𝓀[k]ˣ] (P : Pair k n) : GsStar χ φ P = ∑ c : 𝓀[k]ˣ, χ (-(c : 𝓀[k])) • φ (c⁻¹ • P)

theorem **trLat_apply** (L : Submodule 𝒪[k] (n → k)) : trLat L φ = ∑ P ∈ withLat L, φ P

theorem **trKer_apply** (L : Submodule 𝒪[k] (n → k)) : trKer L φ = ∑ P ∈ withKer L, φ P

theorem **tr_apply** (P : Pair k n) : tr φ P = ∑ Q ∈ withLat P.lat, φ Q

theorem **tr'_apply** (P : Pair k n) : tr' φ P = ∑ Q ∈ withKer P.ker, φ Q

theorem **Da_apply** (a : 𝓀[k]) (P₁ : Pair k n) : Da q a φ P₁ = ∑ P ∈ withLat P₁.ker with cross q P₁ P = a, φ P

theorem **D_apply** (P₁ : Pair k n) : D q φ P₁ = ∑ P ∈ withLat P₁.ker with cross q P₁ P = 1, φ P

theorem **U_apply** (P : Pair k n) : U q φ P = ∑ P₁ ∈ withKer P.lat with cross q P₁ P = 1, φ P₁

theorem **FF_apply** (P₁ : Pair k n) : FF q χ φ P₁ = ∑ P ∈ withLat P₁.ker, χ (cross q P₁ P) • φ P

theorem **FFStar_apply** (P : Pair k n) : FFStar q χ φ P = ∑ P₁ ∈ withKer P.lat, χ (-cross q P₁ P) • φ P₁

theorem **FF_up** [Fintype {f : Module.Dual 𝓀[k] (Red L) // f ≠ 0}] (x : {x : Red L // x ≠ 0}) : FF q χ φ (up L q x) = ∑ f : {f : Module.Dual 𝓀[k] (Red L) // f ≠ 0}, χ (f.1 x.1) • φ (ofDual L f)

theorem **FFStar_ofDual** [Fintype {x : Red L // x ≠ 0}] (f : {f : Module.Dual 𝓀[k] (Red L) // f ≠ 0}) : FFStar q χ φ (ofDual L f) = ∑ x : {x : Red L // x ≠ 0}, χ (-f.1 x.1) • φ (up L q x)

theorem **dia_sub_mem_Mram** (c : 𝓀[k]ˣ) : dia c φ - φ ∈ Mram k n M

theorem **dia_smul_eq_iff** {c : 𝓀[k]ˣ} {P : Pair k n} : c • P = P ↔ c = 1

theorem **sum_withLat_smul** {A : Type*} [AddCommMonoid A] (g : GL n k) (L : Submodule 𝒪[k] (n → k)) (F : Pair k n → A) : ∑ P ∈ withLat (g • L), F P = ∑ P ∈ withLat L, F (g • P)

theorem **sum_withKer_smul** {A : Type*} [AddCommMonoid A] (g : GL n k) (L : Submodule 𝒪[k] (n → k)) (F : Pair k n → A) : ∑ P ∈ withKer (g • L), F P = ∑ P ∈ withKer L, F (g • P)

@[simp] theorem **transl_apply** (g : GL n k) (P : Pair k n) : transl g φ P = φ (g⁻¹ • P)

theorem **formOf_smul** (P₀ : Pair k n) (v : levelVec π P₀) (g : GL n k) : formOf π P₀ v (g • P₀) = π g v

theorem **formOf_mem** (P₀ : Pair k n) (v : levelVec π P₀) : formOf π P₀ v ∈ EqForm π

theorem **map_mem_EqForm** {T : Module.End ℂ (Form k n M)} (h1 : ∀ (g : GL n k) φ P, T φ (g • P) = T (fun Q => φ (g • Q)) P) (h2 : ∀ (g : GL n k) (φ : Form k n M) P, T (fun Q => π g (φ Q)) P = π g (T φ P)) {φ : Form k n M} (h : φ ∈ EqForm π) : T φ ∈ EqForm π

theorem **dia_smul** (c : 𝓀[k]ˣ) : dia c φ (g • P) = dia c (fun Q => φ (g • Q)) P

theorem **U_smul** : U q φ (g • P) = U q (fun Q => φ (g • Q)) P

theorem **FF_smul** : FF q χ φ (g • P) = FF q χ (fun Q => φ (g • Q)) P

theorem **FFStar_smul** : FFStar q χ φ (g • P) = FFStar q χ (fun Q => φ (g • Q)) P

theorem **dia_mem_EqForm** (c : 𝓀[k]ˣ) (h : φ ∈ EqForm π) : dia c φ ∈ EqForm π

theorem **U_mem_EqForm** (h : φ ∈ EqForm π) : U q φ ∈ EqForm π

theorem **FF_mem_EqForm** (h : φ ∈ EqForm π) : FF q χ φ ∈ EqForm π

theorem **FFStar_mem_EqForm** (h : φ ∈ EqForm π) : FFStar q χ φ ∈ EqForm π

theorem **apply_int_eq_one** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (x : 𝒪[k]) : ψ x = 1

theorem **apply_inv_mul_congr** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (q : Unif k) {s t : 𝒪[k]} (h : residue 𝒪[k] s = residue 𝒪[k] t) : ψ (q.inv * s) = ψ (q.inv * t)

theorem **resChar_residue** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (q : Unif k) (s : 𝒪[k]) : resChar ψ hψ q (residue 𝒪[k] s) = ψ (q.inv * s)

theorem **conj_resChar** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (q : Unif k) (a : 𝓀[k]) : (starRingEnd ℂ) (resChar ψ hψ q a) = resChar ψ hψ q (-a)

theorem **resChar_ne_one** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (q : Unif k) : resChar ψ hψ q ≠ 1

theorem **hasConductor_mulShift** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (u : 𝒪[k]ˣ) : HasConductor (ψ.mulShift ((u : 𝒪[k]) : k)) 1

theorem **maps_iff_hasVal** : Maps P Q x ↔ ∀ y a, P.HasVal y a → Q.HasVal (x *ᵥ y) a

theorem **maps_iff_glue** {e : n' → k} {e' : n → k} (he : P.HasVal e 1) (he' : Q.HasVal e' 1) : Maps P Q x ↔ (∀ y ∈ P.ker, x *ᵥ y ∈ Q.ker) ∧ x *ᵥ e - e' ∈ Q.ker

theorem **maps_dia** (c : 𝓀[k]ˣ) : Maps (c • P) (c • Q) x ↔ Maps P Q x

theorem **phiOne_dia** (c : 𝓀[k]ˣ) : phiOne (c • P) (c • Q) = phiOne P Q

theorem **maps_smul** (g : GL n' k) (h : GL n k) : Maps (g • P) (h • Q) x ↔ Maps P Q ((h⁻¹ : GL n k).val * x * g.val)

theorem **phiOne_smul** (g : GL n' k) (h : GL n k) : phiOne (g • P) (h • Q) x = phiOne P Q ((h⁻¹ : GL n k).val * x * g.val)

theorem **latInd_smul** (g : GL n' k) (h : GL n k) (S : Submodule 𝒪[k] (n' → k)) (L : Submodule 𝒪[k] (n → k)) : latInd (g • S) (h • L) x = latInd S L ((h⁻¹ : GL n k).val * x * g.val)

theorem **sum_smul_avec** (z : Fin c → k) : ∑ i : Fin c, z i • avec k n c (i.val + 1) = Sum.elim (0 : n → k) z

theorem **gluedLat_one** (q : Unif k) (c : ℕ) (L : Submodule 𝒪[k] (n → k)) (f : n → k) : gluedLat q c 1 L f = gluedS q c L f

theorem **mem_gluedLat** {q : Unif k} {p : ℕ} {L : Submodule 𝒪[k] (n → k)} {f : n → k} {y : n ⊕ Fin c → k} : y ∈ gluedLat q c p L f ↔ ∃ s : 𝒪[k], y ∘ Sum.inl - s • f ∈ L ∧ ∀ i : Fin c, (i.val + 1 < p → y (Sum.inr i) ∈ 𝒪[k]) ∧ (i.val + 1 = p → y (Sum.inr i) = q.inv * s) ∧ (p < i.val + 1 → (q.1 : k) * y (Sum.inr i) ∈ 𝒪[k])

theorem **gluedLat_congr** (q : Unif k) (c p : ℕ) {L : Submodule 𝒪[k] (n → k)} {f f' : n → k} (h : f - f' ∈ L) : gluedLat q c p L f = gluedLat q c p L f'

theorem **inX_mem_gluedLat** {q : Unif k} {p : ℕ} (h1 : 1 ≤ p) (hp : p ≤ c) {L : Submodule 𝒪[k] (n → k)} {f x : n → k} : inX k n c x ∈ gluedLat q c p L f ↔ x ∈ L

theorem **embX_smul_gluedLat** (g : GL n k) (q : Unif k) (c p : ℕ) (L : Submodule 𝒪[k] (n → k)) (f : n → k) : RootEx.embX c g • gluedLat q c p L f = gluedLat q c p (g • L) (g • f)

theorem **exists_smul_lat** (L L' : Submodule 𝒪[k] (n → k)) [L.IsLattice k] [L'.IsLattice k] : ∃ g : GL n k, g • L = L'

theorem **scalar_smul_eq** [Fintype n] [DecidableEq n] (g : GL n k) (s : 𝒪[k]ˣ) (hg : ∀ x : n → k, g • x = (s : 𝒪[k]) • x) (P : Pair k n) : g • P = (Units.map (residue 𝒪[k]).toMonoidHom s)⁻¹ • P

theorem **Mram_ne_bot** [Fintype n] [Nonempty n] {c : 𝓀[k]ˣ} (hc : c ≠ 1) : Mram k n ℂ ≠ ⊥

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.HermitianOrthogonalExtensionData

theorem **sgn_false** (V : Formalism.BasicNotions.HSpace E) : sgn false V = V

theorem **neg_hT** (V : Formalism.BasicNotions.HSpace E) (x y : V.V) : (neg V).hT x y = -V.hT x y

theorem **neg_gram** (V : Formalism.BasicNotions.HSpace E) : (neg V).gram = -V.gram

theorem **neg_sig** (V : Formalism.BasicNotions.HSpace E) (τ : E →+* ℂ) : (neg V).sig τ = ((V.sig τ).2, (V.sig τ).1)

theorem **sgn_rank** (ε : Bool) (V : Formalism.BasicNotions.HSpace E) : (sgn ε V).rank = V.rank

theorem **hasSignature_neg** {n : Type} {J : Matrix n n ℂ} {p q : ℕ} (h : Formalism.UG.HasSignature J p q) : Formalism.UG.HasSignature (-J) q p

theorem **sigOK_sgn** (ε : Bool) {V : Formalism.BasicNotions.HSpace E} (h : V.SigOK) : (sgn ε V).SigOK

theorem **hr_sgn** (ε : Bool) (V : HSp D) : HR (AF D.host.F) D.sp.cm.sig (sgn ε V).gram = HR (AF D.host.F) D.sp.cm.sig V.gram

theorem **mt_ofSgn** (ε : Bool) (V : HSp D) (g : GA D (sgn ε V)) : Formalism.L2.Theta.mt V (ofSgn ε V g) = Formalism.L2.Theta.mt (sgn ε V) g

theorem **ofSgn_false** (V : HSp D) (g : GA D V) : ofSgn false V g = g

theorem **continuous_ofSgn** (ε : Bool) (V : HSp D) : Continuous (ofSgn ε V)

theorem **continuous_ofSgn_symm** (ε : Bool) (V : HSp D) : Continuous (ofSgn ε V).symm

theorem **neg_gram_inv** (V : HSp D) : (neg V).gram⁻¹ = -V.gram⁻¹

theorem **hO_neg** (S T : HSp D) (x y : WO S T) : hO (neg S) T x y = -hO S T x y

theorem **symplO_neg** (S T : HSp D) (x y : WO S T) : symplO (neg S) T x y = -symplO S T x y

theorem **isCartanInv_sgn** (ε : Bool) (V : HSp D) (τ : D.what₂.Tau) {s : Matrix (Fin V.rank) (Fin V.rank) ℂ} (h : IsCartanInv (gramAt D V τ) s) : IsCartanInv (gramAt D (sgn ε V) τ) (sgnM ε s)

theorem **mem_princGL** {β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} {I : Ideal (𝒪 E)} {x : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} : x ∈ princGL β I ↔ ∃ y, frGL β y = x ∧ ∀ i j, (y.1 - 1) i j ∈ idS F I

theorem **princGL_top** (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) : princGL β ⊤ = St β

theorem **princGL_mono** (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) {I I' : Ideal (𝒪 E)} (h : I ≤ I') : princGL β I ≤ princGL β I'

theorem **multT_le_cusp** (θ : InfFam D) (J : Set (GA D N.V)) (χ : GA D N.V → ℂ) : MultT θ N.M J χ ≤ N.Cusp θ

theorem **mem_princ** {I : Ideal (𝒪 D.host.E)} {x : GL (Fin N.V.rank) (Lev.EAf D)} : x ∈ N.princ I ↔ x ∈ UA D.sp.cm N.V.gram ∧ ∃ y, frGL N.β y = x ∧ ∀ i j, (y.1 - 1) i j ∈ idS D.host.F I

theorem **princ_top** : N.princ ⊤ = UA D.sp.cm N.V.gram ⊓ St N.β

theorem **princ_mono** {I I' : Ideal (𝒪 D.host.E)} (h : I ≤ I') : N.princ I ≤ N.princ I'

theorem **isWeil_false** (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (ω : WRep S T) : IsWeil ψ false S T χS χT ω ↔ IsWeilDatumO ψ S T χS χT ω

theorem **IsVac.isFinVec** {f : WO S.V T.V → ℂ} (h : IsVac ψ θ ε S T f) : IsFinVec ψ θ ε S T f

theorem **InCosetOf.unique** {I : Ideal (𝒪 D.host.E)} (hI : I ≠ ⊥) {c c' : Coset S T I} {x : WO S.V T.V} (h : InCosetOf S T I c x) (h' : InCosetOf S T I c' x) : c = c'

theorem **exists_inCosetOf** {I : Ideal (𝒪 D.host.E)} (hI : I ≠ ⊥) (x : WO S.V T.V) : (∃ c, InCosetOf S T I c x) ↔ IsIntM (frMapO S.V T.V S.β T.β x)

theorem **finite_coset** {I : Ideal (𝒪 D.host.E)} (hI : I ≠ ⊥) : Finite (Coset S T I)

theorem **isIntw_iff** {V W : HSp D} {S S' : D.what₂.Tau → SPar} (θ : InfFam D) (M : DSModel V S) (M' : DSModel W S') (ω : WRep V W) (pt : PolF D W.rank V.rank → (WO V W → ℂ)) (It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M) : IsIntw θ M M' ω pt It ↔ IsItW θ M M' ω pt It

theorem **WRep.swap_apply** {S T : HSp D} (ω : WRep S T) (h : GA D T) (g : GA D S) : ω.swap (h, g) = ω (g, h)

theorem **ArrS.IsLift.int_eq** {a : ArrS S T} {m : T.MeasDat} {f : WO S.V T.V → ℂ} {io : T.Cusp θ →ₗ[ℂ] S.Cusp θ} (h : a.IsLift ψ θ ε m f io) (F : T.Cusp θ) (P : Pol S T) (v : SmT T.M) : a.int m (fock ψ θ ε S T f P) (F.1 v) = (io F).1 (a.It P v)

theorem **ArrT.IsLift.int_eq** {a : ArrT S T} {m : S.MeasDat} {f : WO S.V T.V → ℂ} {io : S.Cusp θ →ₗ[ℂ] T.Cusp θ} (h : a.IsLift ψ θ ε m f io) (F : S.Cusp θ) (P : Pol S T) (v : SmT S.M) : a.int m (fock ψ θ ε S T f P) (F.1 v) = (io F).1 (a.It P v)

theorem **Node.coe_castCusp** (θ : InfFam D) (h : S = S') (F : S.Cusp θ) : (Node.castCusp θ h F).1 = Node.castMaps h F.1

theorem **Node.castMaps_mem** (θ : InfFam D) (h : S = S') (F : S.Maps) : Node.castMaps h F ∈ S'.Cusp θ ↔ F ∈ S.Cusp θ

theorem **ArrS.is_cast** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (ε : Bool) (hS : S = S') (hT : T = T') (a : ArrS S T) (χS χT : (AE D)ˣ →* Circle) : (a.cast hS hT).Is ψ θ ε χS χT ↔ a.Is ψ θ ε χS χT

theorem **isFinVec_cast** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (ε : Bool) (hS : S = S') (hT : T = T') (f : WO S.V T.V → ℂ) : IsFinVec ψ θ ε S' T' (castFun hS hT f) ↔ IsFinVec ψ θ ε S T f

theorem **ArrS.isLift_cast** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (ε : Bool) (hS : S = S') (hT : T = T') (a : ArrS S T) (m : T.MeasDat) (f : WO S.V T.V → ℂ) (io : T'.Cusp θ →ₗ[ℂ] S'.Cusp θ) : (a.cast hS hT).IsLift ψ θ ε (m.cast hT) (castFun hS hT f) io ↔ a.IsLift ψ θ ε m f ((Node.castCusp θ hS).symm.toLinearMap ∘ₗ io ∘ₗ (Node.castCusp θ hT).toLinearMap)

theorem **ArrS.cast_vac** (hS : S = S') (hT : T = T') (a : ArrS S T) : (a.cast hS hT).vac = castFun hS hT a.vac

theorem **ArrS.cast_It_congr** (hS : S = S') (hT : T = T') {a b : ArrS S T} (h : a.It = b.It) : (a.cast hS hT).It = (b.cast hS hT).It

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsAuxiliaryVectorWittDec

theorem **mulVec_idAt** {m n : ℕ} (M : Mx D m n) {z : Fin n → AE D} (hz : ∀ j, z j ∈ idAt D P) (i : Fin m) : (M *ᵥ z) i ∈ idAt D P

theorem **inK_one** : InK S LS P 1

theorem **inK_mul** {κ κ' : GA D S} (h : InK S LS P κ) (h' : InK S LS P κ') : InK S LS P (κ * κ')

theorem **inK_inv** {κ : GA D S} (h : InK S LS P κ) : InK S LS P κ⁻¹

theorem **CondK.cond** (H : CondK dW p x y ψ LW P h) : Cond dW p x y ψ P h

theorem **condK_mul** (hκ : InK W LW P κ) : CondK dW p x y ψ LW P (h * κ) ↔ CondK dW p x y ψ LW P h

theorem **cosetAt_mul** {R : Subgroup (GA D W)} {v : GlobalTheory.FinPl.Pl D} (hκ : InK W LW (· = v) κ) : CosetAt W R LW v (h ∗ κ) ↔ CosetAt W R LW v h

theorem **cosetAt_of_inK** {R : Subgroup (GA D W)} {v : GlobalTheory.FinPl.Pl D} (hκ : InK W LW (· = v) κ) : CosetAt W R LW v κ

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsHermitianAuxiliaryVectorSystemBesselVectors

theorem **comp_vecMul** (x : l → E) (M : Matrix l m E) : τ ∘ (x ᵥ* M) = (τ ∘ x) ᵥ* M.map τ

theorem **comp_mulVec** (M : Matrix l m E) (x : m → E) : τ ∘ (M *ᵥ x) = M.map τ *ᵥ (τ ∘ x)

theorem **comp_dotProduct** (x y : l → E) : (τ ∘ x) ⬝ᵥ (τ ∘ y) = τ (x ⬝ᵥ y)

theorem **dot_mem** (I : Ideal K) (u x : l → K) {M : Matrix l l K} (h : ∀ i j, M i j ∈ I) : u ⬝ᵥ M *ᵥ x ∈ I

theorem **sandwich_mem** (I : Ideal K) (P : Matrix m l K) (Q : Matrix l m K) {M : Matrix l l K} (h : ∀ i j, M i j ∈ I) (i j : m) : (P ∗ M ∗ Q) i j ∈ I

theorem **map_dot** (φ : E →+* K) (x : l → E) (M : Matrix l m E) (y : m → E) : φ (x ⬝ᵥ M *ᵥ y) = (φ ∘ x) ⬝ᵥ M.map φ *ᵥ (φ ∘ y)

theorem **ext_by_one_mul** {m' n c : Type} [Fintype m'] [Fintype n] [Fintype c] [DecidableEq m] [DecidableEq m'] [DecidableEq c] {J : Matrix m m' K} {P : Matrix m' m K} {G : Matrix m' m' K} {Γ : Matrix m' n K} {g : Matrix n n K} {B : Matrix m c K} (h1 : P ∗ J = 1) (h2 : P ∗ B = 0) (h3 : G ∗ Γ = Γ ∗ g) : (1 + J ∗ (G − 1) ∗ P) ∗ fromCols (J ∗ Γ) B = fromCols (J ∗ Γ) B ∗ (fromBlocks g 0 0 1 : Matrix (n ⊕ c) (n ⊕ c) K)

theorem **map_ext_by_one** {m' : Type} [Fintype m'] [DecidableEq m] [DecidableEq m'] (φ : E →+* K) (J : Matrix m m' E) (P : Matrix m' m E) (G : Matrix m' m' E) : (1 + J ∗ (G − 1) ∗ P).map φ = 1 + J.map φ ∗ (G.map φ − 1) ∗ P.map φ

theorem **smul_dot_inv** [DecidableEq l] (φ : K →+* K) {G G' : Matrix l l K} (h : G' ∗ G = 1) (hG : (G.map φ)ᵀ = G) (b : K) (x : l → K) : (b • (x ᵥ* G)) ⬝ᵥ G' *ᵥ (φ ∘ (b • (x ᵥ* G))) = b ∗ (φ b ∗ x ⬝ᵥ G *ᵥ (φ ∘ x))

theorem **tensAt_mem** (y : Fv D.host.F v ⊗[D.host.F] D.host.E) : tensAt D.host.E v y ∈ locIdeal D v

theorem **trace_of_mem** {z : AE D} (hz : z ∈ locIdeal D v) : Algebra.trace (AF D.host.F) (AE D) z = atPlace v (piW τ z + piW τ (cA D z))

theorem **cA_ratV** {m : ℕ} (x : Fin m → D.host.E) : cA D ∘ ratV x = ratV (star x)

theorem **gA_mul_mt_inv** (g : GA D V) : gA V ∗ mt V g⁻¹ = adj (mt V g) ∗ gA V

theorem **cA_hA** (g : GA D V) (x y : Fin V.rank → D.host.E) : cA D (hA V g x y) = hA V g⁻¹ y x

theorem **incA_hQ** (x y : Fin V.rank → D.host.E) : incA D (hQ V x y) = hA V 1 x y

theorem **star_hQ** (x y : Fin V.rank → D.host.E) : star (hQ V x y) = hQ V y x

theorem **piW_hA** (g : GA D V) (x y : Fin V.rank → D.host.E) : piW τ (hA V g x y) = (τ ∘ (star y ᵥ* V.gram)) ⬝ᵥ (mt V g).map (piW τ) *ᵥ (τ ∘ x)

theorem **hA_locGL_mem** (g : GL ι (Fv D.host.F v)) {x y : Fin V.rank → D.host.E} (h : hQ V x y = 0) : hA V (locGL V τ β g) x y ∈ locIdeal D v

theorem **eq_locGL** {x : GA D V} (hx : ∀ i j, (mt V x − 1) i j ∈ locIdeal D v) {g : GL ι (Fv D.host.F v)} (h : (mt V x).map (piW τ) ∗ cols β = cols β ∗ g.val) : x = locGL V τ β g

theorem **simpleSum_eq_sum_ent** {k : Type} [Field k] (g : GL (n ⊕ Fin c) k) : GlobalTheory.RootEx.simpleSum g = ∑ i ∈ Finset.range (c − 1), ent g i

theorem **ent_inv** {k : Type} [Field k] {p : ℕ} {g : GL (n ⊕ Fin c) k} (hg : g ∈ N k n c p) (i : ℕ) : ent g⁻¹ i = −ent g i

theorem **IsScaled.ne_zero** (h : IsScaled V X τ jX a γ β t) {i : ℕ} (hi : i < c) : t i ≠ 0

theorem **IsHermitianAuxiliaryVectorSystem.row_inl** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) {i : ℕ} (hi : i < c) (u : Fin X.rank → Fv D.host.F v) : (τ ∘ (star (a i) ᵥ* V.gram)) ⬝ᵥ jX.map τ *ᵥ u = 0

theorem **IsHermitianAuxiliaryVectorSystem.row_inr** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) {i l : ℕ} (hi : i < c) (hl : l < c) : (τ ∘ (star (a (c − 1 − i)) ᵥ* V.gram)) ⬝ᵥ (τ ∘ a l) = if l = i then τ (κ i) else 0

theorem **IsHermitianAuxiliaryVectorSystem.piW_hA** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (hβ : IsScaled V X τ jX a γ β t) (g : GL (n ⊕ Fin c) (Fv D.host.F v)) (i j : Fin c) : piW τ (hA V (locGL V τ β g) (a j) (a (c − 1 − i))) = τ (κ i) ∗ t i / t j ∗ g.val (.inr i) (.inr j)

theorem **IsHermitianAuxiliaryVectorSystem.proj_mul** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : (X.gram⁻¹ ∗ (jXᴴ ∗ V.gram)).map τ ∗ jX.map τ = 1

theorem **IsHermitianAuxiliaryVectorSystem.exists_isScaled** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (γ : LatFrame (Fv D.host.F v) n (Fin X.rank)) {t : ℕ → Fv D.host.F v} (ht : ∀ i < c, t i ≠ 0) : ∃ β : LatFrame (Fv D.host.F v) (n ⊕ Fin c) (Fin V.rank), IsScaled V X τ jX a γ β t

theorem **IsHermitianAuxiliaryVectorSystem.proj_mulVec** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) {i : ℕ} (hi : i < c) : (X.gram⁻¹ ∗ (jXᴴ ∗ V.gram)) *ᵥ a i = 0

theorem **IsScaled.cols_eq** (h : IsScaled V X τ jX a γ β t) : cols β = fromCols (jX.map τ ∗ cols γ) (Matrix.of fun y (i : Fin c) => t i ∗ τ (a i y))

theorem **map_piW_rat** {l m : ℕ} (M : Matrix (Fin l) (Fin m) D.host.E) : (rat M).map (piW τ) = M.map τ

theorem **IsHermitianAuxiliaryVectorSystem.eq_locGL_embX** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (hβ : IsAdapted V X τ jX a γ β) (g : GL n (Fv D.host.F v)) {x : GA D V} (hx : mt V x = 1 + rat jX ∘ (mt X (locGL X τ γ g) − 1) ∘ rat (X.gram⁻¹ ∗ (jXᴴ ∗ V.gram))) : x = locGL V τ β (embX c g)

theorem **colN_fin** {m ρ : ℕ} (M : Matrix (Fin m) (Fin ρ) D.host.E) (i : Fin ρ) : colN M i = fun y => M y i

theorem **coef_ne_zero** {p : ℕ} {a : D.host.E} (ha : a ≠ 0) (i : ℕ) : coef p a i ≠ 0

theorem **hQ_colN** {k l : ℕ} (M : Matrix (Fin V.rank) (Fin k) D.host.E) (M' : Matrix (Fin V.rank) (Fin l) D.host.E) {i j : ℕ} (hi : i < k) (hj : j < l) : hQ V (colN M i) (colN M' j) = (M'ᴴ ∗ V.gram ∗ M) ⟨j, hj⟩ ⟨i, hi⟩

theorem **hQ_mulVec** {k l : ℕ} (M : Matrix (Fin V.rank) (Fin k) D.host.E) (M' : Matrix (Fin V.rank) (Fin l) D.host.E) (x y) : hQ V (M *ᵥ x) (M' *ᵥ y) = star y ⬝ᵥ (M'ᴴ ∗ V.gram ∗ M) *ᵥ x

theorem **colN_eq_mulVec** {m k : ℕ} (M : Matrix (Fin m) (Fin k) D.host.E) {i : ℕ} (hi : i < k) : colN M i = M *ᵥ Pi.single ⟨i, hi⟩ 1

theorem **perp_of_gram** {k l : ℕ} {M : Matrix (Fin V.rank) (Fin k) D.host.E} {J : Matrix (Fin V.rank) (Fin l) D.host.E} (h : Jᴴ ∗ V.gram ∗ M = 0) (x) : star (M *ᵥ x) ᵥ* V.gram ᵥ* J = 0

theorem **hQ_XX** {i j : ℕ} (hi : i < ρ) (hj : j < ρ) : hQ V (colN d.e i) (colN d.e j) = 0

theorem **hQ_YY** {i j : ℕ} (hi : i < ρ) (hj : j < ρ) : hQ V (colN d.e' i) (colN d.e' j) = 0

theorem **hQ_XY** {i j : ℕ} (hi : i < ρ) (hj : j < ρ) : hQ V (colN d.e i) (colN d.e' j) = if i = j then 1 else 0

theorem **hQ_YX** {i j : ℕ} (hi : i < ρ) (hj : j < ρ) : hQ V (colN d.e' i) (colN d.e j) = if j = i then 1 else 0

theorem **wittDecomposition_hermitian_e_anisotropicPart** {i : ℕ} (hi : i < ρ) (z) : hQ V (colN d.e i) (d.i₀ *ᵥ z) = 0

theorem **wittDecomposition_hermitian_dual_anisotropicPart** {i : ℕ} (hi : i < ρ) (z) : hQ V (colN d.e' i) (d.i₀ *ᵥ z) = 0

theorem **hQ_0X** {i : ℕ} (hi : i < ρ) (z) : hQ V (d.i₀ *ᵥ z) (colN d.e i) = 0

theorem **hQ_0Y** {i : ℕ} (hi : i < ρ) (z) : hQ V (d.i₀ *ᵥ z) (colN d.e' i) = 0

theorem **hQ_vecB** : hQ V (d.i₀ *ᵥ vecB o ζ) (d.i₀ *ᵥ vecB o ζ) = hQ L ζ ζ

theorem **isHermitianAuxiliaryVectorSystem_besselVectors** (hζ : IsLineVec L ζ) : IsHermitianAuxiliaryVectorSystem (2 * ρ + 1) V A (d.i₀ ∘ o.iA) (besselAuxiliaryVectors d (vecB o ζ)) fun i => if i = ρ then hQ L ζ ζ else 1 where

theorem **isSys_fj** : IsHermitianAuxiliaryVectorSystem (2 * ρ) V V₀ d.i₀ (fourierJacobiVectors d) fun _ => 1 where

theorem **emb_localGL** {γ : LatFrame (Fv D.host.F v) n (Fin V₀.rank)} (hc : c = 2 * ρ) (h : IsAdapted V V₀ τ d.i₀ (fourierJacobiVectors d) γ β) (g : GL n (Fv D.host.F v)) : d.emb0 (locGL V₀ τ γ g) = locGL V τ β (embX c g)

theorem **embA_locGL** (hζ : IsLineVec L ζ) {γ : LatFrame (Fv D.host.F v) n (Fin A.rank)} (hc : c = 2 * ρ + 1) (h : IsAdapted V A τ (d.i₀ ∘ o.iA) (besselAuxiliaryVectors d (vecB o ζ)) γ β) (g : GL n (Fv D.host.F v)) : embA d o (locGL A τ γ g) = locGL V τ β (embX c g)

theorem **IsHermitianAuxiliaryVectorSystem.simple_mem** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (g : GL (n ⊕ Fin c) (Fv D.host.F v)) {i : ℕ} (hi : i + 1 < c) : hA V (locGL V τ β g) (a (i + 1)) (a (c − 1 − i)) ∈ locIdeal D v

theorem **IsHermitianAuxiliaryVectorSystem.piW_simple** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (hβ : IsScaled V X τ jX a γ β t) (g : GL (n ⊕ Fin c) (Fv D.host.F v)) {i : ℕ} (hi : i + 1 < c) : piW τ (hA V (locGL V τ β g) (a (i + 1)) (a (c − 1 − i))) = τ (κ i) * t i / t (i + 1) * ent g i

theorem **IsHermitianAuxiliaryVectorSystem.piW_cA_simple** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (hβ : IsScaled V X τ jX a γ β t) {p : ℕ} {g : GL (n ⊕ Fin c) (Fv D.host.F v)} (hg : g ∈ N _ n c p) {i : ℕ} (hi : i + 1 < c) : piW τ (cA D (hA V (locGL V τ β g) (a (i + 1)) (a (c − 1 − i)))) = −(τ (κ (c − 2 − i)) * t (c − 2 − i) / t (c − 2 − i + 1) * ent g (c − 2 − i))

theorem **IsHermitianAuxiliaryVectorSystem.trace_sum** (H : IsHermitianAuxiliaryVectorSystem c V X jX a κ) (hβ : IsScaled V X τ jX a γ β t) {p : ℕ} {g : GL (n ⊕ Fin c) (Fv D.host.F v)} (hg : g ∈ N _ n c p) {m : ℕ} (hm : m < c ∨ m = 0) : Algebra.trace (AF D.host.F) (AE D) (∑ i ∈ range m, hA V (locGL V τ β g) (a (i + 1)) (a (c − 1 − i))) = atPlace v (∑ i ∈ range m, τ (κ i) * t i / t (i + 1) * ent g i − ∑ i ∈ range m, τ (κ (c − 1 − m + i)) * t (c − 1 − m + i) / t (c − 1 − m + i + 1) * ent g (c − 1 − m + i))

theorem **blockX_apply** (g : GA D V) (i j : Fin ρ) : blockX d g i j = hA V g (colN d.e j) (colN d.e' i)

theorem **coefX_apply** (z : Fin V₀.rank → D.host.E) (g : GA D V) (i : Fin ρ) : coefX d z g i = hA V g (d.i₀ *ᵥ z) (colN d.e' i)

theorem **upSum_blockX** (g : GA D V) : upSum (blockX d g) = ∑ i ∈ range (ρ − 1), hA V g (colN d.e (i + 1)) (colN d.e' i)

theorem **ell_dot** {σ : ℕ} (x : Fin (σ + 1) → AE D) : ell D (σ + 1) ·ᵥ x = x (Fin.last σ)

theorem **simpleSum_eq_sum_hA** (z : Fin V₀.rank → D.host.E) (g : GA D V) : GlobalTheory.MixPer.simpleSum d z g = ∑ i ∈ range ρ, hA V g (besselAuxiliaryVectors d z (i + 1)) (besselAuxiliaryVectors d z (2 * ρ + 1 − 1 − i))

theorem **upSum_eq_sum_hA** (g : GA D V) : upSum (blockX d g) = ∑ i ∈ range (ρ − 1), hA V g (fourierJacobiVectors d (i + 1)) (fourierJacobiVectors d (2 * ρ − 1 − i))

theorem **psiBes_locGL** (hζ : IsLineVec L ζ) {γ : LatFrame (Fv D.host.F v) n (Fin A.rank)} (hc : c = 2 * ρ + 1) (hβ : IsScaled V A τ (d.i₀ ∘ o.iA) (besselAuxiliaryVectors d (vecB o ζ)) γ β t) {p : ℕ} {g : GL (n ⊕ Fin c) (Fv D.host.F v)} (hg : g ∈ N _ n c p) : psiBes ψ d 1 (vecB o ζ) (locGL V τ β g) = psiV _ ψ v (∑ i ∈ range (c − 1), τ (coef (ρ + 1) (−hQ L ζ ζ) i) * t i / t (i + 1) * ent g i)

theorem **hO_vecMulVec** (S T : HSp D) (u u' : Fin T.rank → AE D) (r r' : Fin S.rank → AE D) : hO S T (vecMulVec u r) (vecMulVec u' r') = (cA D ∘ u') ·ᵥ gA T *ᵥ u * r ·ᵥ giA S *ᵥ (cA D ∘ r')

theorem **trHalf_of_mem** {b : D.host.E} (hb : star b = b) {H : AE D} (hH : H ∈ locIdeal D v) {x : Fv D.host.F v} (h1 : piW τ H = x) (h2 : piW τ (cA D H) = −x) : trHalf D (incA D b * H) = atPlace v (τ (D.host.xi0 * b) * x)

theorem **jY_mulVec_ell** {σ : ℕ} (d : HypDec V V₀ (σ + 1)) : d.jY *ᵥ ell D (σ + 1) = ratV (colN d.e' σ)

theorem **cA_mulVec** {l m : ℕ} (M : Mx D l m) (x : Fin m → AE D) : cA D ∘ (M *ᵥ x) = (cA D ∘ x) ᵥ* adj M

theorem **cA_vecMul** {l m : ℕ} (M : Mx D l m) (x : Fin l → AE D) : cA D ∘ (x ᵥ* M) = adj M *ᵥ (cA D ∘ x)

theorem **incA_hQ'** (x y : Fin V.rank → D.host.E) : ratV (star y) ·ᵥ gA V *ᵥ ratV x = incA D (hQ V x y)

theorem **incAdj_yZero** {σ : ℕ} (d : HypDec V V₀ (σ + 1)) (ζ : Fin L.rank → D.host.E) : d.incAdj L (yZero L (σ + 1) ζ) = vecMulVec (ratV (colN d.e' σ)) (incA D D.host.xi0⁻¹ • (ratV (star ζ) ᵥ* gA L))

theorem **cA_inv_dot** (g : GA D V) (u : Fin V.rank → D.host.E) : (cA D ∘ (mt V g⁻¹ *ᵥ ratV u)) ·ᵥ gA V *ᵥ ratV u = hA V g u u

theorem **conj_xi_inv** : cA D (incA D D.host.xi0⁻¹) = incA D (−D.host.xi0⁻¹)

theorem **cenT_eq** {σ : ℕ} (d : HypDec V V₀ (σ + 1)) (ζ : Fin L.rank → D.host.E) (g : GA D V) : cenT d L (yZero L (σ + 1) ζ) g = trHalf D (incA D (−(D.host.xi0⁻¹ * D.host.xi0⁻¹ * hQ L ζ ζ)) * hA V g (colN d.e' σ) (colN d.e' σ))

theorem **cenS_eq** {σ : ℕ} (d : HypDec V V₀ (σ + 1)) (ζ : Fin L.rank → D.host.E) (g : GA D V) : cenS d L (zEll L (σ + 1) ζ) g = trHalf D (incA D (hQ L ζ ζ) * hA V g⁻¹ (colN d.e' σ) (colN d.e' σ))

theorem **psiFlag_mul_locGL** {σ : ℕ} (d : HypDec V V₀ (σ + 1)) {γ : LatFrame (Fv D.host.F v) n (Fin V₀.rank)} {β : LatFrame (Fv D.host.F v) (n ⊕ Fin (2 * (σ + 1))) (Fin V.rank)} (hβ : IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β t) {p : ℕ} {g : GL (n ⊕ Fin (2 * (σ + 1))) (Fv D.host.F v)} (hg : g ∈ N _ n _ p) (b : D.host.E) : psiFlag ψ d (locGL V τ β g) * ψ (atPlace v (τ b * (t σ / t (σ + 1) * ent g σ))) = psiV _ ψ v (∑ i ∈ range (2 * (σ + 1) − 1), τ (coef (σ + 1) b i) * t i / t (i + 1) * ent g i)

theorem **central_locGL** {σ : ℕ} (d : HypDec V V₀ (σ + 1)) {γ : LatFrame (Fv D.host.F v) n (Fin V₀.rank)} {β : LatFrame (Fv D.host.F v) (n ⊕ Fin (2 * (σ + 1))) (Fin V.rank)} (hβ : IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β t) {p : ℕ} {g : GL (n ⊕ Fin (2 * (σ + 1))) (Fv D.host.F v)} (hg : g ∈ N _ n _ p) {b : D.host.E} (hb : star b = b) : trHalf D (incA D b * hA V (locGL V τ β g) (colN d.e' σ) (colN d.e' σ)) = atPlace v (τ (D.host.xi0 * b) * (t σ / t (σ + 1) * ent g σ))

theorem **cmHost_imaginaryGenerator_ne_zero** : D.host.xi0 ≠ 0

theorem **fjCharT_locGL** (ζ : Fin L.rank → D.host.E) {γ : LatFrame (Fv D.host.F v) n (Fin V₀.rank)} (hc : c = 2 * ρ) (hβ : IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β t) {p : ℕ} {g : GL (n ⊕ Fin c) (Fv D.host.F v)} (hg : g ∈ N _ n c p) : fjCharT ψ d ζ (locGL V τ β g) = psiV _ ψ v (∑ i ∈ range (c − 1), τ (coef ρ (D.host.xi0⁻¹ * hQ L ζ ζ) i) * t i / t (i + 1) * ent g i)

theorem **fjCharS_locGL** (ζ : Fin L.rank → D.host.E) {γ : LatFrame (Fv D.host.F v) n (Fin V₀.rank)} (hc : c = 2 * ρ) (hβ : IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β t) {p : ℕ} {g : GL (n ⊕ Fin c) (Fv D.host.F v)} (hg : g ∈ N _ n c p) : fjCharS ψ d ζ (locGL V τ β g) = psiV _ ψ v (∑ i ∈ range (c - 1), τ (coef ρ (-(D.host.xi0 * hQ L ζ ζ)) i) * t i / t (i + 1) * ent g i)

theorem **normScale_succ** {k : Type} [Field k] {θ : k} (hθ : θ ≠ 0) (κ : ℕ → k) (t₀ : k) (i : ℕ) : θ * normScale θ κ t₀ (i + 1) = κ i * normScale θ κ t₀ i

theorem **normScale_ne_zero** {k : Type} [Field k] {θ : k} (hθ : θ ≠ 0) {κ : ℕ → k} (hκ : ∀ i, κ i ≠ 0) {t₀ : k} (h : t₀ ≠ 0) (i : ℕ) : normScale θ κ t₀ i ≠ 0

theorem **eq_normScale** {k : Type} [Field k] {θ : k} (hθ : θ ≠ 0) {κ t : ℕ → k} {c : ℕ} (h : ∀ i, i + 1 < c → θ * t (i + 1) = κ i * t i) {i : ℕ} (hi : i < c) : t i = normScale θ κ (t 0) i

theorem **exists_root** {k : Type} [Field k] {i : ℕ} (hi : i + 1 < c) (x : k) : ∃ g : GL (n ⊕ Fin c) k, (∀ p, g ∈ N k n c p) ∧ ∀ j, ent g j = if j = i then x else 0

theorem **isNormalFrame_of_scale** (hβ : IsScaled V X τ jX a γ β t) (hχ : ∀ g ∈ U, χ (locGL V τ β g) = psiV _ ψ v (∑ i ∈ range (c - 1), κ i * t i / t (i + 1) * ent g i)) (h : ∀ i, i + 1 < c → θ * t (i + 1) = κ i * t i) : IsNormalFrame V X τ jX a γ β U χ (locChar ψ hψ θ)

theorem **scale_of_isNormalFrame** (h1 : locChar ψ hψ θ ≠ 1) (hβ : IsScaled V X τ jX a γ β t) (hχ : ∀ g ∈ U, χ (locGL V τ β g) = psiV _ ψ v (∑ i ∈ range (c - 1), κ i * t i / t (i + 1) * ent g i)) (hU : ∀ g, (∀ p, g ∈ N _ n c p) → g ∈ U) (h : IsNormalFrame V X τ jX a γ β U χ (locChar ψ hψ θ)) {i : ℕ} (hi : i + 1 < c) : θ * t (i + 1) = κ i * t i

theorem **ne_zero_of_locChar_ne_one** (h1 : locChar ψ hψ θ ≠ 1) : θ ≠ 0

theorem **isBesFrame_of_scale** (hζ : IsLineVec L ζ) (hc : c = 2 * ρ + 1) (hβ : IsScaled V A τ (d.i₀ * o.iA) (besselAuxiliaryVectors d (vecB o ζ)) γ β t) (h : ∀ i, i + 1 < c → θ * t (i + 1) = τ (coef (ρ + 1) (-hQ L ζ ζ) i) * t i) : IsBesFrame ψ d o ζ (locChar ψ hψ θ) τ β γ

theorem **IsBesFrame.exists_scale** (h1 : locChar ψ hψ θ ≠ 1) (hζ : IsLineVec L ζ) (h : IsBesFrame ψ d o ζ (locChar ψ hψ θ) τ β γ) : ∃ t₀, IsScaled V A τ (d.i₀ * o.iA) (besselAuxiliaryVectors d (vecB o ζ)) γ β (normScale θ (fun i => τ (coef (ρ + 1) (-hQ L ζ ζ) i)) t₀)

theorem **exists_isBesFrame** (hζ : IsLineVec L ζ) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (hθ : θ ≠ 0) (γ : LatFrame (Fv D.host.F v) n (Fin A.rank)) {t₀ : Fv D.host.F v} (h₀ : t₀ ≠ 0) : ∃ β : LatFrame (Fv D.host.F v) (n ⊕ Fin (2 * ρ + 1)) (Fin V.rank), IsBesFrame ψ d o ζ (locChar ψ hψ θ) τ β γ ∧ IsScaled V A τ (d.i₀ * o.iA) (besselAuxiliaryVectors d (vecB o ζ)) γ β (normScale θ (fun i => τ (coef (ρ + 1) (-hQ L ζ ζ) i)) t₀)

theorem **isFJFrameT_of_scale** (hc : c = 2 * ρ) (hβ : IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β t) (h : ∀ i, i + 1 < c → θ * t (i + 1) = τ (coef ρ (D.host.xi0⁻¹ * hQ L ζ ζ) i) * t i) : IsFJFrameT ψ d ζ (locChar ψ hψ θ) τ β γ

theorem **IsFJFrameT.exists_scale** (h1 : locChar ψ hψ θ ≠ 1) (h : IsFJFrameT ψ d ζ (locChar ψ hψ θ) τ β γ) : ∃ t₀, IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β (normScale θ (fun i => τ (coef ρ (D.host.xi0⁻¹ * hQ L ζ ζ) i)) t₀)

theorem **exists_isFJFrameT** (hζ : IsLineVec L ζ) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (hθ : θ ≠ 0) (γ : LatFrame (Fv D.host.F v) n (Fin V₀.rank)) {t₀ : Fv D.host.F v} (h₀ : t₀ ≠ 0) : ∃ β : LatFrame (Fv D.host.F v) (n ⊕ Fin (2 * ρ)) (Fin V.rank), IsFJFrameT ψ d ζ (locChar ψ hψ θ) τ β γ ∧ IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β (normScale θ (fun i => τ (coef ρ (D.host.xi0⁻¹ * hQ L ζ ζ) i)) t₀)

theorem **colN_smul** {m : ℕ} (y : D.host.E) (M : Matrix (Fin m) (Fin ρ) D.host.E) (i : ℕ) : colN (y • M) i = y • colN M i

theorem **fourierJacobiVectors_source** (i : ℕ) : fourierJacobiVectors (srcDec d) i = (if i < ρ then D.host.xi0 else -D.host.xi0⁻¹) • fourierJacobiVectors d i

theorem **isScaled_srcDec** : IsScaled V V₀ τ (srcDec d).i₀ (fourierJacobiVectors (srcDec d)) γ β t ↔ IsScaled V V₀ τ d.i₀ (fourierJacobiVectors d) γ β fun i => t i * τ (if i < ρ then D.host.xi0 else -D.host.xi0⁻¹)

theorem **isFJFrameS_srcDec_iff** {ζ' : Fin L'.rank → D.host.E} (hz : hQ L ζ ζ = hQ L' ζ' ζ') (ψv : AddChar (Fv D.host.F v) Circle) : IsFJFrameS ψ (srcDec d) ζ' ψv τ β γ ↔ IsFJFrameT ψ d ζ ψv τ β γ

theorem **isFJFrameS_iff_of_norm** {ζ' : Fin L'.rank → D.host.E} (hz : hQ L ζ ζ = D.host.xi0 * star D.host.xi0 * hQ L' ζ' ζ') (ψv : AddChar (Fv D.host.F v) Circle) : IsFJFrameS ψ d ζ' ψv τ β γ ↔ IsFJFrameT ψ d ζ ψv τ β γ

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsLevDatHalfLatInd

theorem **IsLevN.insert** {E : ExpSet D} {N p : ℕ} (hN : IsLevN E N) (hp : p.Prime) (h : p ≠ 2 ∧ Nat.Coprime p E.l ∧ (∀ a, p ≠ E.pde a) ∧ (∀ a, p ≠ E.gp a) ∧ D.host.cls p = .splitK ∧ ¬ p ∣ Ideal.absNorm D.host.ccK) : ∃ N', IsLevN E N' ∧ N ∣ N' ∧ p ∣ N'

theorem **modN_eq_mul** (E : ExpSet D) (N : ℕ) : modN E N = modN E 1 * N

theorem **TowerRat.modN_ne_zero** {E : ExpSet D} {O : Type} [CommRing O] {C : CoefLat E O} {Base : Set (GA D E.W → ℂ)} {L1 FO : Subfield ℂ} (T : TowerRat E C Base L1 FO) : modN E 1 ≠ 0

theorem **halfLatInd_zero** (E : ExpSet D) (e : AE D) : halfLatInd E e 0 = 1

theorem **isLevDat_halfLatInd** {E : ExpSet D} {N : ℕ} {e : AE D} (P1 : Finset (Formalism.L2.DIV.Pl D.host.E)) (hP : ∀ L ∈ P1, ((N : ℕ) : 𝒪 D.host.E) ∈ L.asIdeal) (he : IsIdemSum e (E.Pg ∪ E.Pd ∪ P1)) : IsLevDat E N e (halfLatInd E e) where

theorem **isLevDat_base** (E : ExpSet D) (N : ℕ) : IsLevDat E N E.e0 (halfLatInd E E.e0)

theorem **IsCompFn.const** (E : ExpSet D) (N : ℕ) (z : ℂ) : IsCompFn E N fun _ => z

theorem **IsCompFn.map₂** {E : ExpSet D} {N : ℕ} {c c' : GA D E.W → ℂ} (h : IsCompFn E N c) (h' : IsCompFn E N c') (f : ℂ → ℂ → ℂ) : IsCompFn E N fun g => f (c g) (c' g)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsPointedOfIsHkIsog

theorem **translT_mul** (a b : T ⟶ A) : translT A (a * b) = translT A b ≫ translT A a

theorem **translT_one** : translT A (1 : T ⟶ A) = 𝟙 _

theorem **mem_transKer** {Z : Scheme.{0}} {Ψ : (T ⊗ A).left ⟶ Z} {c : T ⟶ A} : c ∈ transKer A Ψ ↔ (translT A c).left ≫ Ψ = Ψ

theorem **prod_iso** {Z : Scheme.{0}} (r : Z ⟶ (Over.mk x ⊗ A).left) {U : Type} [CommGroup U] (h : (Z ⟶ A.left) → U) : ∏ i, h (r ≫ isoφ A Ψ i) ^ (![k, -1] : Fin 2 → ℤ) i = h (r ≫ (snd (Over.mk x) A).left) ^ k * (h (r ≫ Ψ ≫ (snd (Over.mk x') A).left))⁻¹

theorem **prod_iso'** {U : Type} [CommGroup U] (h : ((Over.mk x ⊗ A).left ⟶ A.left) → U) : ∏ i, h (isoφ A Ψ i) ^ (![k, -1] : Fin 2 → ℤ) i = h (snd (Over.mk x) A).left ^ k * (h (Ψ ≫ (snd (Over.mk x') A).left))⁻¹

theorem **IsCarried.val** (h : IsCarried A L Ψ k β s' t) {Z : Scheme.{0}} (r : Z ⟶ (Over.mk x ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : sv A L t r z J = sv A L s' (r ≫ Ψ) z J * (val β r z J).val

theorem **isCarried_of_val** (h : ∀ z J, Good L.L z J → sv A L t (𝟙 _) z J = sv A L s' (𝟙 _ ≫ Ψ) z J ∗ (val β (𝟙 _) z J).val) : IsCarried A L Ψ k β s' t

theorem **sec_ext** {X : Scheme.{0}} {M : LB X} {a b : LB.Sec M} (h : ∀ j, a.f j = b.f j) : a = b

theorem **IsCarried.unique** (h : IsCarried A L Ψ k β s' t) (h' : IsCarried A L Ψ k β s' t') : t = t'

theorem **exists_isCarried** (hβ : β.IsGen) (s' : LB.Sec (LPow A L.L 1 (Over.mk x'))) : ∃ t, IsCarried A L Ψ k β s' t

theorem **val_desc** (c : Over.mk x ⟶ A) {Z : Scheme.{0}} (r : Z ⟶ (Over.mk x ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : val (LB.mulGen L.L _ _ _ _ β (LB.invGen L.L _ _ (LB.pullGen L.L _ _ β (translT A c).left))) r z J = val β r z J ∗ (val β (r ≫ (translT A c).left) z J)⁻¹

theorem **IsDescLift.val** (h : IsDescLift A L Ψ k β U) {Z : Scheme.{0}} (r : Z ⟶ (Over.mk x ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : uv U r z J = val β r z J ∗ (val β (r ≫ (translT A U.pt).left) z J)⁻¹

theorem **exists_isDescLift** (hβ : β.IsGen) {c : Over.mk x ⟶ A} (hc : c ∈ transKer A Ψ) : ∃ U : ThetaGrp A L.L k (Over.mk x), U.pt = c ∧ IsDescLift A L Ψ k β U

theorem **IsDescLift.unique** {U' : ThetaGrp A L.L k (Over.mk x)} (h : IsDescLift A L Ψ k β U) (h' : IsDescLift A L Ψ k β U') (e : U.pt = U'.pt) : LB.Gen.Corr L.L _ _ _ _ U.φ U'.φ

theorem **act_val** {T : Over S} {U : ThetaGrp A L.L k T} {a b : LB.Sec (LPow A L.L k T)} (h : IsActOf A L.L k U a b) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : sv A L b r z J = (uv U r z J).val ∗ sv A L a (r ≫ (translT A U.pt).left) z J

theorem **exists_isActOf** {T : Over S} (U : ThetaGrp A L.L k T) (a : LB.Sec (LPow A L.L k T)) : ∃ b, IsActOf A L.L k U a b

theorem **sv_sum** {T : Over S} {n : ℕ} {a : LB.Sec (LPow A L.L k T)} {F : Fin n → LB.Sec (LPow A L.L k T)} (h : ∀ j, a.f j = ∑ i, (F i).f j) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : sv A L a r z J = ∑ i, sv A L (F i) r z J

theorem **sum_of_sv** {T : Over S} {n : ℕ} {a : LB.Sec (LPow A L.L k T)} {F : Fin n → LB.Sec (LPow A L.L k T)} (h : ∀ z J, Good L.L z J → sv A L a (𝟙 _) z J = ∑ i, sv A L (F i) (𝟙 _) z J) (j) : a.f j = ∑ i, (F i).f j

theorem **isThetaTrace_iff** {p : ℕ} {s : LB.Sec (LPow A L.L 1 (Over.mk x))} : IsThetaTrace A L Ψ p s s' ↔ ∃ (β : LB.Gen (isoLB A L Ψ ((p : ℤ) ^ 2))) (s₁ : LB.Sec (LPow A L.L ((p : ℤ) ^ 2) (Over.mk x))) (n : ℕ) (c : Fin n → (Over.mk x ⟶ A)), IsRigIso A L Ψ _ β ∧ IsMulPull A L p s s₁ ∧ IsRepsModTors (transKer A Ψ) p c ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (Over.mk x ⊗ A).left) (z : Z) (J), Good L.L z J → sv A L s' (r ≫ Ψ) z J = ∑ i, ((val β (r ≫ (translT A (c i)).left) z J)⁻¹).val ∗ sv A L s₁ (r ≫ (translT A (c i)).left) z J

theorem **sv_smul** {T : Over S} {a : Γ(T.left, ⊤)} {u v : LB.Sec (LPow A L.L k T)} (h : ∀ j, v.f j = liftFun A _ a ∗ u.f j) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : sv A L v r z J = gs (r ≫ (fst T A).left) z ⊤ a ∗ sv A L u r z J

theorem **IsThetaTrace.smul** {p : ℕ} {s u : LB.Sec (LPow A L.L 1 (Over.mk x))} {u' : LB.Sec (LPow A L.L 1 (Over.mk x'))} (h : IsThetaTrace A L Ψ p s s') (a : Γ(Y, ⊤)) (hu : ∀ j, u.f j = liftFun A (T

theorem **val_rigT** (T : Over S) (k : ℤ) {Z : Scheme.{0}} (r : Z ⟶ T.left) (z : Z) {J} (hJ : Good L.L z J) : val (rigT A L.L L.rig T k) r z J = val (L

theorem **isPointed_mulT** (p : ℕ) : IsPointed A (x

theorem **mem_transKer_mulT** [IsCommMonObj A] {T : Over S} (p : ℕ) (c : T ⟶ A) : c ∈ transKer A (mulT A T p).left ↔ c ^ p = 1

theorem **IsMulPull.exists_isRigIso** {p : ℕ} {s : LB.Sec (LPow A L.L 1 (Over.mk x))} {s₁ : LB.Sec (LPow A L.L ((p : ℤ) ^ 2) (Over.mk x))} (h : IsMulPull A L p s s₁) : ∃ β, IsRigIso A L (mulT A (Over.mk x) p).left _ β ∧ IsCarried A L (mulT A (Over.mk x) p).left _ β s s₁

theorem **isThetaTrace_mulT** [IsCommMonObj A] {p : ℕ} {s : LB.Sec (LPow A L.L 1 (Over.mk x))} {s₁ : LB.Sec (LPow A L.L ((p : ℤ) ^ 2) (Over.mk x))} (h : IsMulPull A L p s s₁) : IsThetaTrace A L (mulT A (Over.mk x) p).left p s s

theorem **IsRepsModTors.card** {G : Type} [CommGroup G] {C : Subgroup G} {p n : ℕ} {r : Fin n → G} (h : IsRepsModTors C p r) : Nat.card C = Nat.card {c : G // c ∈ C ∧ c ^ p = 1} ∗ n

theorem **exists_isRepsModTors** {G : Type} [CommGroup G] (C : Subgroup G) [Finite C] (p : ℕ) : ∃ (n : ℕ) (r : Fin n → G), IsRepsModTors C p r

theorem **IsRigIso.val_unique** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {β' : LB.Gen (isoLB A L Ψ k)} (h : IsRigIso A L Ψ k β) (h' : IsRigIso A L Ψ k β') {Z : Scheme.{0}} (r : Z ⟶ (Over.mk x ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : val β r z J = val β' r z J

theorem **IsMulPull.unique** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {p : ℕ} {s : LB.Sec (LPow A L.L 1 (Over.mk x))} {s₁ s₂ : LB.Sec (LPow A L.L ((p : ℤ) ^ 2) (Over.mk x))} (h₁ : IsMulPull A L p s s₁) (h₂ : IsMulPull A L p s s₂) : s₁ = s₂

theorem **IsDescLift.symLift** [IsCommMonObj A] (h : IsDescLift A L Ψ k β U) (hβ : β.IsGen) (hs : L.IsSymmetric) (hneg : (negT A (Over.mk x)).left ≫ Ψ = Ψ ≫ (negT A (Over.mk x')).left) {n : ℕ} (hn : U.pt ^ n = 1) : SymLift A L.L k n U.pt U

theorem **IsThetaTrace.sv_eq_of_mulT** [IsCommMonObj A] (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {p : ℕ} {s u : LB.Sec (LPow A L.L 1 (Over.mk x))} (h : IsThetaTrace A L (mulT A (Over.mk x) p).left p s u) {Z : Scheme.{0}} (r : Z ⟶ (Over.mk x ⊗ A).left) (z : Z) {J} (hJ : Good L.L z J) : sv A L u (r ≫ (mulT A (Over.mk x) p).left) z J = sv A L s (r ≫ (mulT A (Over.mk x) p).left) z J

theorem **gx_hom_ext** {f g : Gx x ⟶ Gx x'} (hf : f ≫ (fst (Over.mk x') Xo.univ.t.G).left = (fst (Over.mk x) Xo.univ.t.G).left) (hg : g ≫ (fst (Over.mk x') Xo.univ.t.G).left = (fst (Over.mk x) Xo.univ.t.G).left) (h : ∀ a : Over.mk x ⟶ Xo.univ.t.G, (lift (𝟙 (Over.mk x)) a).left ≫ f = (lift (𝟙 (Over.mk x)) a).left ≫ g) : f = g

theorem **isPointed_of_isHkIsog** (h : IsHkIsog p x x' C ψ ψp) : IsPointed Xo.univ.t.G (conj x x' ψ)

theorem **transKer_of_isHkIsog** (h : IsHkIsog p x x' C ψ ψp) : transKer Xo.univ.t.G (conj x x' ψ) = C

theorem **card_reps** {P : Ideal (𝒪 E)} {i n : ℕ} (hp : 0 < p) (hC : Xo.univ.t.HeckeKer (Over.mk x) Dw P p i C) {r : Fin n → (Over.mk x ⟶ Xo.univ.t.G)} (hr : IsRepsModTors C p r) : n = Ideal.absNorm P ^ i

theorem **exists_isMulPull** (p : ℕ) {T : Over Xo.X} (s : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 T)) : ∃ s₁, IsMulPull Xo.univ.t.G Xo.univ.t.PP p s s₁

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsSubquotientDetRepParInd

theorem **nabs_pos'** (x : kˣ) : 0 < nabs k x

theorem **nabs_mul'** (x y : kˣ) : nabs k (x ∗ y) = nabs k x ∗ nabs k y

theorem **nabs_eq_one** {x : kˣ} (hx : valuation k (x : k) = 1) : nabs k x = 1

theorem **val_absPow** (s : ℂ) (x : kˣ) : (absPow k s x : ℂ) = ((nabs k x : ℝ) : ℂ) ^ s

theorem **val_absPow_ofReal** (s : ℝ) (x : kˣ) : (absPow k s x : ℂ) = ((nabs k x ^ s : ℝ) : ℂ)

theorem **absPow_add** (s t : ℂ) : absPow k (s + t) = absPow k s ∗ absPow k t

@[simp] theorem **absPow_zero** : absPow k 0 = 1

theorem **absPow_neg** (s : ℂ) : absPow k (-s) = (absPow k s)⁻¹

theorem **isUnram_absPow** (s : ℂ) : IsUnram (absPow k s)

theorem **isSmoothChar_of_isUnram** {χ : kˣ →* ℂˣ} (h : IsUnram χ) : IsSmoothChar χ

theorem **ord_mul** (x y : kˣ) : ord k (x * y) = ord k x + ord k y

theorem **ord_eq_zero** {x : kˣ} (hx : valuation k (x : k) = 1) : ord k x = 0

theorem **ord_eq_one** {x : kˣ} (hx : valuation k (x : k) = uniformizer k) : ord k x = 1

theorem **ord_eq_zero_iff** {x : kˣ} : ord k x = 0 ↔ valuation k (x : k) = 1

theorem **unrChar_apply** (α : ℂˣ) (x : kˣ) : unrChar k α x = α ^ ord k x

theorem **isUnram_unrChar** (α : ℂˣ) : IsUnram (unrChar k α)

theorem **unrChar_uniformizer** (α : ℂˣ) {x : kˣ} (hx : valuation k (x : k) = uniformizer k) : unrChar k α x = α

theorem **eq_unrChar** {χ : kˣ →* ℂˣ} (h : IsUnram χ) {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) : χ = unrChar k (χ ϖ)

@[simp] theorem **detChar_apply** (χ : Rˣ →* ℂˣ) (g : GL n R) : detChar n χ g = χ (GeneralLinearGroup.det g)

theorem **continuous_glDet** [TopologicalSpace R] [IsTopologicalRing R] : Continuous (GeneralLinearGroup.det : GL n R → Rˣ)

theorem **IsSmoothChar.detChar** [TopologicalSpace R] [IsTopologicalRing R] {χ : Rˣ →* ℂˣ} (h : IsSmoothChar χ) : IsSmoothChar (detChar n χ)

@[simp] theorem **torusChar_apply** {n : Type} [Fintype n] (χ : n → Rˣ →* ℂˣ) (t : n → Rˣ) : torusChar χ t = ∏ i, χ i (t i)

theorem **deltaHalf_apply** (g : Levi k m) : deltaHalf k m g = ∏ a, absPow k ((modExp m a : ℂ) / 2) (GeneralLinearGroup.det (g a))

theorem **modulus_pos** (g : Levi k m) : 0 < modulus k m g

theorem **val_deltaHalf** (g : Levi k m) : (deltaHalf k m g : ℂ) = (√(modulus k m g) : ℝ)

theorem **isSmoothChar_deltaHalf** : IsSmoothChar (deltaHalf k m)

theorem **parInd_law** (e : (Σ a, m a) ≃ n) (ρ : Representation ℂ (Levi k m) W) (f : IndSp e ρ) (p : parab k e) (g : GL n k) : f.1.1 (p * g) = (deltaHalf k m (leviProj k e p) : ℂ) • ρ (leviProj k e p) (f.1.1 g)

@[simp] theorem **parInd_apply** (e : (Σ a, m a) ≃ n) (ρ : Representation ℂ (Levi k m) W) (g x : GL n k) (f : IndSp e ρ) : (parInd e ρ g f).1.1 x = f.1.1 (x * g)

theorem **isSmooth_parInd** (e : (Σ a, m a) ≃ n) (ρ : Representation ℂ (Levi k m) W) : IsSmooth (parInd e ρ)

theorem **jac_mk** (e : (Σ a, m a) ≃ n) (π : Representation ℂ (GL n k) V) (g : Levi k m) (v : V) : jac e π g (Submodule.Quotient.mk v) = (((deltaHalf k m g)⁻¹ : ℂˣ) : ℂ) • Submodule.Quotient.mk (π (leviEmb k e g) v)

theorem **isSmooth_jacU** (e : (Σ a, m a) ≃ n) {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jacU e π)

theorem **isSmooth_jac** (e : (Σ a, m a) ≃ n) {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jac e π)

theorem **modExp_opp** (a : α) : modExp (fun a : αᵒᵈ => m (OrderDual.ofDual a)) (OrderDual.toDual a) = -modExp m a

theorem **deltaHalf_opp** (g : Levi k m) : deltaHalf k (fun a : αᵒᵈ => m (OrderDual.ofDual a)) g = (deltaHalf k m g)⁻¹

@[simp] theorem **parIndLower_apply** (e : (Σ a, m a) ≃ n) (ρ : Representation ℂ (Levi k m) W) (g x : GL n k) (f : IndSpLower e ρ) : (parIndLower e ρ g f).1.1 x = f.1.1 (x * g)

theorem **jacLower_eq** (e : (Σ a, m a) ≃ n) (π : Representation ℂ (GL n k) V) (g : Levi k m) : jacLower e π g = jac (opp e) π g

theorem **isSmooth_jacLower** (e : (Σ a, m a) ≃ n) {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jacLower e π)

theorem **deltaHalf_two** (g : GL a k × GL b k) : deltaHalf k (Two a b) (leviTwo k a b g) = absPow k ((Fintype.card b : ℂ) / 2) (GeneralLinearGroup.det g.1) * absPow k (-(Fintype.card a : ℂ) / 2) (GeneralLinearGroup.det g.2)

theorem **isSmooth_jac₂** (e : a ⊕ b ≃ n) {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jac₂ e π)

theorem **isSmooth_jac₂Lower** (e : a ⊕ b ≃ n) {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jac₂Lower e π)

theorem **isSmooth_prod₂** {r t : ℕ} (π₁ : Representation ℂ (GL (Fin r) k) V) (π₂ : Representation ℂ (GL (Fin t) k) W) : IsSmooth (prod₂ π₁ π₂)

theorem **jac₂_detRep** (e : a ⊕ b ≃ n) (χ : kˣ →* ℂˣ) (g : GL a k × GL b k) (x : Jacquet (detRep n χ) (unipRad k (two e))) : jac₂ e (detRep n χ) g x = ((detChar a (χ * absPow k (-(Fintype.card b : ℂ) / 2)) g.1 * detChar b (χ * absPow k ((Fintype.card a : ℂ) / 2)) g.2 : ℂˣ) : ℂ) • x

theorem **twKer_detRep** {R : Type} [CommRing R] (e : (Σ a, m a) ≃ n) (χ : Rˣ →* ℂˣ) : twKer (detRep n χ) (unipRad R e) 1 = ⊥

theorem **modulus_torus** {n : Type} [Fintype n] [LinearOrder n] (t : n → kˣ) : modulus k (fun _ : n => Unit) (leviTorus k n t) = ∏ i, ∏ j with i < j, nabs k (t i / t j)

theorem **jacBU_mk** (π : Representation ℂ (GL n k) V) (t : n → kˣ) (v : V) : jacBU π t (Submodule.Quotient.mk v) = Submodule.Quotient.mk (π (diagT k n t) v)

theorem **isSmooth_jacBU** {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jacBU π)

theorem **isSmooth_jacB** {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) : IsSmooth (jacB π)

theorem **iwahori_le_KO** : iwahori k n ≤ KO k n

theorem **KOne_le_iwahori** : KOne k n ≤ iwahori k n

theorem **mem_iwahori** {g : GL n k} : g ∈ iwahori k n ↔ g ∈ KO k n ∧ ∀ i j, j < i → valuation k ((g : Matrix n n k) i j) < 1

theorem **diagT_mem_iwahori** {t : n → kˣ} (ht : t ∈ torusO k n) : diagT k n t ∈ iwahori k n

theorem **detLevi_apply** (e : (Σ a, m a) ≃ n) (χ : kˣ →* ℂˣ) (g : Levi k m) : detLevi e χ g = ∏ a, (χ * absPow k (-(modExp m a : ℂ) / 2)) (GeneralLinearGroup.det (g a))

theorem **isSubquotient_jac_detRep** (e : (Σ a, m a) ≃ n) (χ : kˣ →* ℂˣ) : IsSubquotient (ofChar (detLevi e χ)) (jac e (detRep n χ))

theorem **isSubquotient_detRep_parInd** (e : (Σ a, m a) ≃ n) {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) : IsSubquotient (detRep n χ) (parInd e (ofChar (detLevi e χ)))

theorem **modExp_fin** {N : ℕ} (X : Fin N → Type) [∀ i, Fintype (X i)] (hX : ∀ i, Fintype.card (X i) = 1) (i : Fin N) : modExp X i = (N : ℤ) - 1 - 2 * (i : ℕ)

theorem **inSupport_detRep** {N : ℕ} (χ : kˣ →* ℂˣ) (i : Fin N) : InSupport (detRep (Fin N) χ) (χ ∗ absPow k (((i : ℕ) : ℂ) - ((N : ℂ) - 1) / 2))

theorem **isSpherical_detRep** {χ : kˣ →* ℂˣ} (h : IsUnram χ) : IsSpherical (detRep n χ)

theorem **hasSatake_detRep** {N : ℕ} {χ : kˣ →* ℂˣ} (h : IsUnram χ) {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) : HasSatake (detRep (Fin N) χ) fun i => χ ϖ ∗ absPow k (((i : ℕ) : ℂ) - ((N : ℂ) - 1) / 2) ϖ

theorem **isCuspidal_detRep_one** {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) : IsCuspidal (detRep (Fin 1) χ)

theorem **not_isCuspidal_detRep_two** (χ : kˣ →* ℂˣ) : ¬ IsCuspidal (detRep (Fin 2) χ)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsThetaTraceMulTIff

theorem **fn_ext_of_surjective** {W Y : Scheme.{0}} (Ψ : W ⟶ Y) [Surjective Ψ] [IsReduced Y] {U : Y.Opens} {a b : Γ(Y, U)} (h : ∀ w, Ψ w ∈ U → gs Ψ w U a = gs Ψ w U b) : a = b

theorem **sec_eq_of_surjective** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {T : Over S} {W : Scheme.{0}} (Ψ : W ⟶ (T ⊗ A).left) [Surjective Ψ] [IsReduced (T ⊗ A).left] {k : ℤ} {u v : LB.Sec (LPow A L.L k T)} (h : ∀ w J, Good L.L w J → sv A L u Ψ w J = sv A L v Ψ w J) : u = v

theorem **isThetaTrace_mulT_iff** {p : ℕ} (hp : 1 ≤ p) {Ω : Type} [Field Ω] (x : Sp Ω ⟶ Xo.X) (s u : LB.Sec (PPx x)) : IsThetaTrace Xo.univ.t.G Xo.univ.t.PP (x

theorem **hkTr_scale_iff** {p : ℕ} (hp : 1 ≤ p) {Ω : Type} [Field Ω] [IsAlgClosed Ω] [Algebra c.F Ω] (y : FrPt Xo c Ω) (u : LB.Sec (PPx y.x)) (hu : ∀ s' : LB.Sec (PPx y.x), ∃! a : Ω, ∀ i, s'.f i = scal y.x a i ∗ u.f i) (a : Ω) (ha : ∀ i, u.f i = scal y.x a i ∗ y.s.f i) : hkTr Xo c p Ω y (scale y u hu) ((conj y.x y.x).symm (mulT Xo.univ.t.G (Over.mk y.x) p).left) ↔ a = 1

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsUnitarisableOfCharIff

@[simp] theorem **twist_apply** (ρ : Representation F G V) (χ : G →* Fˣ) (g : G) (v : V) : twist ρ χ g v = (χ g : F) • ρ g v

theorem **twist_twist** (ρ : Representation F G V) (χ ψ : G →* Fˣ) : twist (twist ρ χ) ψ = twist ρ (ψ ∗ χ)

@[simp] theorem **twist_one** (ρ : Representation F G V) : twist ρ 1 = ρ

@[simp] theorem **ofChar_apply** (χ : G →* Fˣ) (g : G) (x : F) : ofChar χ g x = (χ g : F) ∗ x

theorem **isIrreducible_ofChar** (χ : G →* Fˣ) : (ofChar χ).IsIrreducible

theorem **IsSubquotient.of_surjective** {σ : Representation F G W} {ρ : Representation F G V} (f : ρ.IntertwiningMap σ) (hf : Function.Surjective f) : IsSubquotient σ ρ

theorem **IsSubquotient.refl** (ρ : Representation F G V) : IsSubquotient ρ ρ

theorem **IsSubquotient.trans** {τ : Representation F G U} {σ : Representation F G W} {ρ : Representation F G V} (h₁ : IsSubquotient τ σ) (h₂ : IsSubquotient σ ρ) : IsSubquotient τ ρ

theorem **IsSubquotient.of_injective** {σ : Representation F G W} {ρ : Representation F G V} (f : σ.IntertwiningMap ρ) (hf : Function.Injective f) : IsSubquotient σ ρ

@[simp] theorem **extTensor_tmul** (ρ : Representation F G V) (σ : Representation F H W) (g : G × H) (v : V) (w : W) : extTensor ρ σ g (v ⊗ₜ w) = ρ g.1 v ⊗ₜ σ g.2 w

@[simp] theorem **piTensor_tprod** {α : Type} {Γ X : α → Type} [∀ a, Group (Γ a)] [∀ a, AddCommGroup (X a)] [∀ a, Module F (X a)] (ρ : ∀ a, Representation F (Γ a) (X a)) (g : ∀ a, Γ a) (x : ∀ a, X a) : piTensor ρ g (PiTensorProduct.tprod F x) = PiTensorProduct.tprod F fun a => ρ a (g a) (x a)

theorem **jacquet_mk** (ρ : Representation F G V) (N : Subgroup G) (s : H →* G) (hs : ∀ m, s m ∈ Subgroup.normalizer (N : Set G)) (m : H) (v : V) : jacquet ρ N s hs m (Submodule.Quotient.mk v) = Submodule.Quotient.mk (ρ (s m) v)

theorem **isSmooth_iff_isSmoothRep** {X : Type} [AddCommGroup X] [Module ℂ X] (ρ : Representation ℂ G X) : IsSmooth ρ ↔ IsSmoothRep ρ

theorem **isSmooth_comp** [TopologicalSpace H] {ρ : Representation F G V} (hρ : IsSmooth ρ) (s : H →* G) (hs : Continuous s) : IsSmooth (ρ.comp s : Representation F H V)

theorem **IsSmoothChar.comp** [TopologicalSpace H] {χ : G →* Fˣ} (hχ : IsSmoothChar χ) (s : H →* G) (hs : Continuous s) : IsSmoothChar (χ.comp s)

theorem **IsSmoothChar.inv** {χ : G →* Fˣ} (hχ : IsSmoothChar χ) : IsSmoothChar χ⁻¹

theorem **mem_smoothPart_of_nhds** {ρ : Representation F G V} {v : V} (h : {g : G | ρ g v = v} ∈ nhds (1 : G)) : v ∈ smoothPart ρ

theorem **isSmooth_of_nhds** {ρ : Representation F G V} (h : ∀ v, {g : G | ρ g v = v} ∈ nhds (1 : G)) : IsSmooth ρ

theorem **IsSmoothChar.mul** {χ ψ : G →* Fˣ} (hχ : IsSmoothChar χ) (hψ : IsSmoothChar ψ) : IsSmoothChar (χ ∗ ψ)

theorem **isSmooth_twist** {ρ : Representation F G V} (hρ : IsSmooth ρ) {χ : G →* Fˣ} (hχ : IsSmoothChar χ) : IsSmooth (twist ρ χ)

theorem **isSmooth_ofChar_iff** (χ : G →* Fˣ) : IsSmooth (ofChar χ) ↔ IsSmoothChar χ

theorem **IsSubquotient.isSmooth** {σ : Representation F G W} {ρ : Representation F G V} (h : IsSubquotient σ ρ) (hρ : IsSmooth ρ) : IsSmooth σ

theorem **isSmooth_smoothRep** (ρ : Representation F G V) : IsSmooth (smoothRep ρ)

theorem **isSmooth_jacquet** [TopologicalSpace H] [IsTopologicalGroup H] {ρ : Representation F G V} (hρ : IsSmooth ρ) (N : Subgroup G) (s : H →* G) (hs : ∀ m, s m ∈ Subgroup.normalizer (N : Set G)) (hc : Continuous s) : IsSmooth (jacquet ρ N s hs)

theorem **isSmooth_extTensor** [TopologicalSpace H] [IsTopologicalGroup H] {ρ : Representation F G V} {σ : Representation F H W} (hρ : IsSmooth ρ) (hσ : IsSmooth σ) : IsSmooth (extTensor ρ σ)

theorem **isSmooth_piTensor** {α : Type} [Finite α] {Γ X : α → Type} [∀ a, Group (Γ a)] [∀ a, TopologicalSpace (Γ a)] [∀ a, IsTopologicalGroup (Γ a)] [∀ a, AddCommGroup (X a)] [∀ a, Module F (X a)] {ρ : ∀ a, Representation F (Γ a) (X a)} (hρ : ∀ a, IsSmooth (ρ a)) : IsSmooth (piTensor ρ)

@[simp] theorem **contra_apply** (ρ : Representation F G V) (g : G) (l : smoothPart ρ.dual) (v : V) : (contra ρ g l : Module.Dual F V) v = (l : Module.Dual F V) (ρ g⁻¹ v)

theorem **isSmooth_contra** (ρ : Representation F G V) : IsSmooth (contra ρ)

theorem **isUnitarisable_ofChar_iff** (χ : G →* ℂˣ) : IsUnitarisable (ofChar χ) ↔ IsUnitaryChar χ

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.IsVacFOfIsVacVec

theorem **ptF_ofModel** {V W : HSp D} {S S' : D.what₂.Tau → SPar} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (M : DSModel V S) (M' : DSModel W S') (f : WO V W → ℂ) : ptF ψ θ (Fr.ofModel M) (Fr.ofModel M') f = ptOf ψ θ M M' f

theorem **isVacF_of_isVacVec** {V W : HSp D} {S S' : D.what₂.Tau → SPar} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {M : DSModel V S} {M' : DSModel W S'} {f : WO V W → ℂ} (h : IsVacVec ψ θ M M' f) : IsVacF ψ θ (Fr.ofModel M) (Fr.ofModel M') f

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.LeviEmbMemNormalizerLower

@[simp] theorem **_root_.GlobalTheory.SmGL.blk_apply** (e : (Σ a, m a) ≃ n) (a : α) (i : m a) : _root_.GlobalTheory.SmGL.blk e (e ⟨a, i⟩) = a

theorem **_root_.GlobalTheory.SmGL.mem_parab** {e : (Σ a, m a) ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.parab R e ↔ ∀ i j, _root_.GlobalTheory.SmGL.blk e j < _root_.GlobalTheory.SmGL.blk e i → (g : Matrix n n R) i j = 0

theorem **_root_.GlobalTheory.SmGL.mem_parab_iff** {e : (Σ a, m a) ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.parab R e ↔ ∀ a b (i : m a) (j : m b), b < a → (g : Matrix n n R) (e ⟨a, i⟩) (e ⟨b, j⟩) = 0

@[simp] theorem **leviProj_apply** (e : (Σ a, m a) ≃ n) (p : parab R e) (a : α) (i j : m a) : (leviProj R e p a : Matrix (m a) (m a) R) i j = ((p : GL n R) : Matrix n n R) (e ⟨a, i⟩) (e ⟨a, j⟩)

@[simp] theorem **_root_.GlobalTheory.SmGL.leviEmb_apply_same** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) (a : α) (i j : m a) : (_root_.GlobalTheory.SmGL.leviEmb R e g : Matrix n n R) (e ⟨a, i⟩) (e ⟨a, j⟩) = (g a : Matrix (m a) (m a) R) i j

theorem **_root_.GlobalTheory.SmGL.leviEmb_apply_ne** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) {a b : α} (h : a ≠ b) (i : m a) (j : m b) : (_root_.GlobalTheory.SmGL.leviEmb R e g : Matrix n n R) (e ⟨a, i⟩) (e ⟨b, j⟩) = 0

theorem **_root_.GlobalTheory.SmGL.leviEmb_mem_parab** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) : _root_.GlobalTheory.SmGL.leviEmb R e g ∈ _root_.GlobalTheory.SmGL.parab R e

@[simp] theorem **_root_.GlobalTheory.SmGL.leviProj_leviEmb** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) : _root_.GlobalTheory.SmGL.leviProj R e (_root_.GlobalTheory.SmGL.leviEmb R e g, _root_.GlobalTheory.SmGL.leviEmb_mem_parab R e g) = g

private lemma **bijective_blockPosition_of_equiv_sigma** (e : (Σ a, m a) ≃ n) (a : α) : Function.Bijective fun i : m a => (⟨e ⟨a, i⟩, _root_.GlobalTheory.SmGL.blk_apply e a i⟩ : {k : n // _root_.GlobalTheory.SmGL.blk e k = a})

theorem **_root_.GlobalTheory.SmGL.det_parab** (e : (Σ a, m a) ≃ n) (p : _root_.GlobalTheory.SmGL.parab R e) : GeneralLinearGroup.det (p : GL n R) = ∏ a, GeneralLinearGroup.det (_root_.GlobalTheory.SmGL.leviProj R e p a)

theorem **_root_.GlobalTheory.SmGL.det_leviEmb** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) : GeneralLinearGroup.det (_root_.GlobalTheory.SmGL.leviEmb R e g) = ∏ a, GeneralLinearGroup.det (g a)

theorem **_root_.GlobalTheory.SmGL.mem_unipRad** {e : (Σ a, m a) ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.unipRad R e ↔ g ∈ _root_.GlobalTheory.SmGL.parab R e ∧ ∀ a (i j : m a), (g : Matrix n n R) (e ⟨a, i⟩) (e ⟨a, j⟩) = (1 : Matrix (m a) (m a) R) i j

theorem **_root_.GlobalTheory.SmGL.unipRad_le_parab** (e : (Σ a, m a) ≃ n) : _root_.GlobalTheory.SmGL.unipRad R e ≤ _root_.GlobalTheory.SmGL.parab R e

theorem **_root_.GlobalTheory.SmGL.coe_mem_unipRad** {e : (Σ a, m a) ≃ n} (p : _root_.GlobalTheory.SmGL.parab R e) : (p : GL n R) ∈ _root_.GlobalTheory.SmGL.unipRad R e ↔ _root_.GlobalTheory.SmGL.leviProj R e p = 1

private lemma **conj_mem_unipRad_of_mem_parab** {e : (Σ a, m a) ≃ n} {p u : GL n R} (hp : p ∈ _root_.GlobalTheory.SmGL.parab R e) (hu : u ∈ _root_.GlobalTheory.SmGL.unipRad R e) : p * u * p⁻¹ ∈ _root_.GlobalTheory.SmGL.unipRad R e

theorem **_root_.GlobalTheory.SmGL.parab_le_normalizer** (e : (Σ a, m a) ≃ n) : _root_.GlobalTheory.SmGL.parab R e ≤ Subgroup.normalizer (_root_.GlobalTheory.SmGL.unipRad R e : Set (GL n R))

theorem **_root_.GlobalTheory.SmGL.leviEmb_mem_normalizer** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) : _root_.GlobalTheory.SmGL.leviEmb R e g ∈ Subgroup.normalizer (_root_.GlobalTheory.SmGL.unipRad R e : Set (GL n R))

theorem **_root_.GlobalTheory.SmGL.exists_levi_mul_unip** (e : (Σ a, m a) ≃ n) (p : _root_.GlobalTheory.SmGL.parab R e) : ∃ u ∈ _root_.GlobalTheory.SmGL.unipRad R e, (p : GL n R) = _root_.GlobalTheory.SmGL.leviEmb R e (_root_.GlobalTheory.SmGL.leviProj R e p) * u

theorem **_root_.GlobalTheory.SmGL.det_eq_one_of_mem_unipRad** {e : (Σ a, m a) ≃ n} {u : GL n R} (hu : u ∈ _root_.GlobalTheory.SmGL.unipRad R e) : GeneralLinearGroup.det u = 1

theorem **_root_.GlobalTheory.SmGL.continuous_leviEmb** [TopologicalSpace R] {α : Type} {m : α → Type} [Fintype α] [∀ a, Fintype (m a)] [∀ a, DecidableEq (m a)] (e : (Σ a, m a) ≃ n) : Continuous (_root_.GlobalTheory.SmGL.leviEmb R e)

theorem **_root_.GlobalTheory.SmGL.mem_parabLower_iff** {e : (Σ a, m a) ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.parabLower R e ↔ ∀ a b (i : m a) (j : m b), a < b → (g : Matrix n n R) (e ⟨a, i⟩) (e ⟨b, j⟩) = 0

@[simp] theorem **leviProjLower_apply** (e : (Σ a, m a) ≃ n) (p : parabLower R e) (a : α) (i j : m a) : (leviProjLower R e p a : Matrix (m a) (m a) R) i j = ((p : GL n R) : Matrix n n R) (e ⟨a, i⟩) (e ⟨a, j⟩)

theorem **_root_.GlobalTheory.SmGL.mem_unipRadLower** {e : (Σ a, m a) ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.unipRadLower R e ↔ g ∈ _root_.GlobalTheory.SmGL.parabLower R e ∧ ∀ a (i j : m a), (g : Matrix n n R) (e ⟨a, i⟩) (e ⟨a, j⟩) = (1 : Matrix (m a) (m a) R) i j

theorem **_root_.GlobalTheory.SmGL.leviEmb_mem_parabLower** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) : _root_.GlobalTheory.SmGL.leviEmb R e g ∈ _root_.GlobalTheory.SmGL.parabLower R e

theorem **_root_.GlobalTheory.SmGL.leviEmb_mem_normalizer_lower** (e : (Σ a, m a) ≃ n) (g : _root_.GlobalTheory.SmGL.Levi R m) : _root_.GlobalTheory.SmGL.leviEmb R e g ∈ Subgroup.normalizer (_root_.GlobalTheory.SmGL.unipRadLower R e : Set (GL n R))

theorem **_root_.GlobalTheory.SmGL.stdBlocks_val** {s N : ℕ} (d : Fin s → ℕ) (h : ∑ a, d a = N) (a : Fin s) (i : Fin (d a)) : (_root_.GlobalTheory.SmGL.stdBlocks d h ⟨a, i⟩ : ℕ) = ∑ b : Fin a, d (Fin.castLE a.2.le b) + i

@[simp] theorem **two_false** (e : a ⊕ b ≃ n) (i : a) : two e (false, i) = e (Sum.inl i)

@[simp] theorem **two_true** (e : a ⊕ b ≃ n) (j : b) : two e (true, j) = e (Sum.inr j)

@[simp] theorem **_root_.GlobalTheory.SmGL.stdSum_inl** {r t N : ℕ} (h : r + t = N) (i : Fin r) : (_root_.GlobalTheory.SmGL.stdSum h (Sum.inl i) : ℕ) = i

@[simp] theorem **_root_.GlobalTheory.SmGL.stdSum_inr** {r t N : ℕ} (h : r + t = N) (j : Fin t) : (_root_.GlobalTheory.SmGL.stdSum h (Sum.inr j) : ℕ) = r + j

theorem **_root_.GlobalTheory.SmGL.continuous_leviTwo** [TopologicalSpace R] : Continuous (_root_.GlobalTheory.SmGL.leviTwo R a b)

theorem **_root_.GlobalTheory.SmGL.val_emb₂** (e : a ⊕ b ≃ n) (g : GL a R × GL b R) : (_root_.GlobalTheory.SmGL.emb₂ R e g : Matrix n n R) = Matrix.reindex e e (fromBlocks g.1.val 0 0 g.2.val)

theorem **_root_.GlobalTheory.SmGL.det_emb₂** (e : a ⊕ b ≃ n) (g : GL a R × GL b R) : GeneralLinearGroup.det (_root_.GlobalTheory.SmGL.emb₂ R e g) = GeneralLinearGroup.det g.1 * GeneralLinearGroup.det g.2

theorem **_root_.GlobalTheory.SmGL.mem_parab_two** {e : a ⊕ b ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.parab R (_root_.GlobalTheory.SmGL.two e) ↔ ∀ (j : b) (i : a), (g : Matrix n n R) (e (.inr j)) (e (.inl i)) = 0

theorem **_root_.GlobalTheory.SmGL.mem_parabLower_two** {e : a ⊕ b ≃ n} {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.parabLower R (_root_.GlobalTheory.SmGL.two e) ↔ ∀ (i : a) (j : b), (g : Matrix n n R) (e (.inl i)) (e (.inr j)) = 0

@[simp] theorem **points_apply** (i : n) (u : Unit) : points n (i, u) = i

theorem **_root_.GlobalTheory.SmGL.continuous_leviTorus** [TopologicalSpace R] (n : Type) : Continuous (_root_.GlobalTheory.SmGL.leviTorus R n)

theorem **_root_.GlobalTheory.SmGL.val_diagT** (t : n → Rˣ) : (_root_.GlobalTheory.SmGL.diagT R n t : Matrix n n R) = diagonal fun i => (t i : R)

theorem **_root_.GlobalTheory.SmGL.mem_borel** {n : Type} [Fintype n] [DecidableEq n] [LinearOrder n] {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.borel R n ↔ ∀ i j, j < i → (g : Matrix n n R) i j = 0

theorem **_root_.GlobalTheory.SmGL.mem_unipB** {n : Type} [Fintype n] [DecidableEq n] [LinearOrder n] {g : GL n R} : g ∈ _root_.GlobalTheory.SmGL.unipB R n ↔ (∀ i j, j < i → (g : Matrix n n R) i j = 0) ∧ ∀ i, (g : Matrix n n R) i i = 1

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.LowVanishFockOfLowVanish

theorem **exists_sign_frame** {n : ℕ} {G : Matrix (Fin n) (Fin n) ℂ} (hG : G.IsHermitian) (hu : IsUnit G.det) : ∃ (ε : Fin n → Bool) (A : Matrix (Fin n) (Fin n) ℂ), IsUnit A ∧ Aᴴ * Formalism.REAL.Jm ε * A = G

theorem **nonempty_fr** (V : HSp D) : Nonempty (Fr V)

theorem **isCartanInv_cart** {V : HSp D} (A : Fr V) (τ : D.what₂.Tau) : IsCartanInv (gramAt D V τ) (A.cart τ)

theorem **exists_vac** {V W : HSp D} (A : Fr V) (B : Fr W) {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) : ∃ f, f ≠ 0 ∧ IsVacF ψ θ A B f ∧ ptF ψ θ A B f 1 = f ∧ ∃ U : AddSubgroup (WO V W), IsOpen (U : Set (WO V W)) ∧ f ∈ fockSpan ψ θ A B U

theorem **dense_hyp** (V W : HSp D) : ∃ (A : Fr V) (B : Fr W) (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (μ : Measure (WO V W)) (ℱ : Set (WO V W)) (f : WO V W → ℂ), IsStdChar D ψ (-1) ∧ IsInfFam θ ∧ IsCovolOne (ratL W V) μ ℱ ∧ f ≠ 0 ∧ IsVacF ψ θ A B f

theorem **lieFock_of_all** (h : LieFockAll D) {V W : HSp D} {S S' : D.what₂.Tau → SPar} (M : DSModel V S) (M' : DSModel W S') {θ : InfFam D} {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} {f : WO V W → ℂ} {τ : D.what₂.Tau} {aS aT : ℚ} (hθ : IsInfFam θ) (hω : IsWeilDatumO ψ V W χS χT ω) (hS : HasInfExp D χS τ aS) (hT : HasInfExp D χT τ aT) (hf : IsVacVec ψ θ M M' f) : LieFock ψ θ M M' ω f τ aS aT

theorem **compactFock_of_all** (h : CompactFockAll D) {V W : HSp D} {S S' : D.what₂.Tau → SPar} (M : DSModel V S) (M' : DSModel W S') {θ : InfFam D} {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} {f : WO V W → ℂ} {τ : D.what₂.Tau} {aS aT : ℚ} (hθ : IsInfFam θ) (hω : IsWeilDatumO ψ V W χS χT ω) (hS : HasInfExp D χS τ aS) (hT : HasInfExp D χT τ aT) (hf : IsVacVec ψ θ M M' f) : CompactFock ψ θ M M' ω f τ aS aT

theorem **lowVanishFock_of_lowVanish** (h : GlobalTheory.IndefiniteRowSideFullHecke.LoweredSignatureVanishing.LowVanish D) : LowVanishFock D

theorem **fockS_offdiag** (τ : D.what₂.Tau) : fockS τ ![true, false] ![true] (Matrix.single 0 1 1) (1 : PolF D 1 2) = -(MvPolynomial.X (τ, 0, 0) * MvPolynomial.X (τ, 0, 1)) ∧ fockS τ ![true, false] ![true] (Matrix.single 1 0 1) (1 : PolF D 1 2) = 0

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.MemHalfCosetSelf

theorem **rootsIn_iff** {n : ℕ} (hn : n ≠ 0) : RootsIn σ n ↔ ∃ z : K, IsPrimitiveRoot z n

theorem **LocIn.valuesIn** {χ : HChar D.host.E} (h : LocIn σ χ) : χ.ValuesIn 0 σ

theorem **mem_halfCoset_self** {m n : ℕ} (e : AE D) (Λ : AddSubgroup (Mx D m n)) (k : ℕ) (x₀ : Mx D m n) : x₀ ∈ halfCoset e Λ k x₀

theorem **halfCoset_eq** {m n : ℕ} {e : AE D} {Λ : AddSubgroup (Mx D m n)} {k : ℕ} {x₀ x : Mx D m n} (h : x ∈ halfCoset e Λ k x₀) : halfCoset e Λ k x = halfCoset e Λ k x₀

theorem **isCosetDatum_ind** (t : T) (p : Pair T) (P h e he) : IsCosetDatum 1 (FinDatum.ind G t p P h e he)

theorem **IsCosetDatum.fn_eq** {t : T} {p : Pair T} {k : ℕ} {d d' : FinDatum G t p} (h : IsCosetDatum k d) (h' : IsCosetDatum k d') (he : d.e = d'.e) {x} (hx : d.fn x ≠ 0) (hx' : d'.fn x ≠ 0) : d.fn = d'.fn

theorem **IsCosetExpansion.exists_expDepth** {k τ} (h : IsCosetExpansion S t Φ β k τ) : ∃ τ', IsCosetExpansion S t Φ β (expDepth S t Φ β) τ'

private theorem **finDatum_ext** {t : T} {p : Pair T} {d d' : FinDatum G t p} (hP : d.P = d'.P) (he : d.e = d'.e) (hf : d.fn = d'.fn) (hz : d.z = d'.z) : d = d'

theorem **IsCosetExpansion.coef_eq** {k τ} (h : IsCosetExpansion S t Φ β k τ) {d} (hd : d ∈ τ.support) : ∃ x y, τ d = ∏ w, (S.term t Φ β w).datum (S.colAt t w x) (S.rowAt t w y)

theorem **SumsIn.rationalSums** {A A' WW : HSp D} {X : Tri T A A' WW} (h : SumsIn W σ X) (hi : ∀ t P g, IntegrableOn (X.ker W.ψ W.θ t P g) X.fd X.m) (hi' : ∀ t P g, IntegrableOn (X.ker' W.ψ W.θ t P g) X.fd' X.m') : X.RationalSums W.ψ W.θ σ

private theorem **valuesIn_of_range_subset** {χ : HChar D.host.E} {s : ℝ} (h : χ.ValuesIn s σ) (hσ : Set.range σ ⊆ Set.range σ') : χ.ValuesIn s σ'

theorem **NumbersIn.mono** {S : GridChoice W Set.univ} {μ₀ : T} (h : NumbersIn σ S μ₀) (hσ : Set.range σ ⊆ Set.range σ') : NumbersIn σ' S μ₀ where

theorem **numbersIn_id** (S : GridChoice W Set.univ) (μ₀ : T) : NumbersIn (RingHom.id ℂ) S μ₀ where

theorem **FieldIn.fieldOK** (h : ch.FieldIn σ) (hσ : Set.range σ ⊆ Set.range D.what₂.embLL) : ch.FieldOK

theorem **FieldOK.hasNumberField** (h : ch.FieldOK) : ch.HasNumberField

theorem **FieldOK.isRatAt** (h : ch.FieldOK) {t Φ β} (hβ : β ∈ ch.S.supp t Φ) {τ} (hτ : IsCosetExpansion ch.S.toGridChoice t Φ β (expDepth ch.S.toGridChoice t Φ β) τ) {d} (hd : d ∈ τ.support) (c : ch.S.Classes t Φ β) : (∀ j : Fin D.R.m, ch.S.Z.IsRatAt (d.1 j) (ch.S.ptTop c j) (ch.S.ptTop c (j + 1))) ∧ ∀ i : Fin D.R.m, ch.S.Z.IsRatAt (d.2 i) (ch.S.ptRow c i) (ch.S.ptRow c (i + 1))

theorem **FieldOK.valLL** (h : ch.FieldOK) : Formalism.L2.Host.ValLL D R.C

theorem **FieldOK.fieldCond** (h : ch.FieldOK) (hW : ch.W.Over R) (G' : GlobalTheory.Grid.GridPre R) (h1 : ∀ i, 1 ≤ i → i ≤ D.R.m → G'.chiH i = ch.G.ch.chiH i ∧ G'.chi i = ch.G.ch.chi i) (h2 : G'.toPlaceDat.W R.Pg = ch.G.pl.W0) (h3 : ∀ w ∈ ch.G.pl.W0, G'.toPlaceDat.depth w = depthAt ch.G w) (hL : ∀ i j Λ₀, GlobalTheory.CMVal.IsHeckeL (SqNum.nuH ch.G A.μ0 i j) Λ₀ → ∀ s : ℕ, 1 ≤ s → Λ₀ s ≠ 0) (hB : ∀ i j, ∃ (B B' WW : HSp D) (X : Tri T B B' WW), IsBottomTri ch.W i j X) (hI : ∀ i j (B B' WW : HSp D) (X : Tri T B B' WW), IsBottomTri ch.W i j X → ∀ t, (∀ P g, IntegrableOn (X.ker ch.W.ψ ch.W.θ t P g) X.fd X.m) ∧ ∀ P g, IntegrableOn (X.ker' ch.W.ψ ch.W.θ t P g) X.fd' X.m') : FieldCond G'

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.MemLevelFunOfIsAutFun

theorem **coe_locChar** (ψ : AddChar (AF F) ℂ) (hψ : ∀ x, ‖ψ x‖ = 1) {v : HeightOneSpectrum (𝒪 F)} (a x : Fv F v) : (locChar ψ hψ a x : ℂ) = psiV F ψ v (a * x)

theorem **continuous_locChar** {ψ : AddChar (AF F) ℂ} (hc : Continuous ψ) (hψ : ∀ x, ‖ψ x‖ = 1) {v : HeightOneSpectrum (𝒪 F)} (a : Fv F v) : Continuous (locChar ψ hψ a)

theorem **mat_emb_sub_one** {E : Type} [Field E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (v : HeightOneSpectrum (𝒪 F)) (x : GlobalTheory.AdelicU.GLoc σ J v) (i j : Fin n) : ((GlobalTheory.AdelicU.matHom _ σ J (GlobalTheory.AdelicU.emb σ J v x)).val - 1) i j = tensAt E v (((GlobalTheory.AdelicU.matHom _ σ J x).val - 1) i j)

theorem **cols_apply** (β : LatFrame k ι Y) (y : Y) (i : ι) : cols β y i = vec β i y

theorem **rows_mul_cols** (β : LatFrame k ι Y) : what₂ β * cols β = 1

theorem **cols_mul_rows** (β : LatFrame k ι Y) : cols β * what₂ β = 1

theorem **conj_val** (β : LatFrame k ι Y) (g : GL ι k) : (β.conj g).val = cols β * g.val * what₂ β

theorem **conj_mul_cols** (β : LatFrame k ι Y) (g : GL ι k) : (β.conj g).val * cols β = cols β * g.val

theorem **conj_mulVec_vec** (β : LatFrame k ι Y) (g : GL ι k) (j : ι) : (β.conj g).val *ᵥ vec β j = ∑ r, g.val r j • vec β r

theorem **exists_frame** (hc : Fintype.card ι = Fintype.card Y) (M : Matrix Y ι k) (hM : ∀ u, M *ᵥ u = 0 → u = 0) : ∃ β : LatFrame k ι Y, cols β = M

theorem **continuous_conj** [TopologicalSpace k] [IsTopologicalRing k] (β : LatFrame k ι Y) : Continuous β.conj

theorem **chart_piW** (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : GlobalTheory.SplitHecke.Chart (idem τ) v (piW τ)

theorem **piW_incA** (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (x : D.host.E) : piW τ (incA D x) = τ x

theorem **hermND** (V : HSp D) : IsHermND D.sp.cm.sig V.gram

theorem **proj_locGL** (g : GL ι (Fv D.host.F v)) : GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v (locGL V τ β g) = locPt V τ β g

theorem **proj_locGL_ne** {u : HeightOneSpectrum (𝒪 D.host.F)} (h : u ≠ v) (g : GL ι (Fv D.host.F v)) : GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram u (locGL V τ β g) = 1

theorem **locGL_mem_GfC** (g : GL ι (Fv D.host.F v)) : locGL V τ β g ∈ GfC V

theorem **map_piW** (g : GA D V) : (mt V g).map (piW τ) = (splitHom D.sp.cm.sig V.gram τ (GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g)).val

theorem **map_piW_locGL** (g : GL ι (Fv D.host.F v)) : (mt V (locGL V τ β g)).map (piW τ) = (β.conj g).val

theorem **continuous_locPt** : Continuous (locPt V τ β)

theorem **continuous_locGL** : Continuous (locGL V τ β)

theorem **commute_locGL** (g : GL ι (Fv D.host.F v)) {h : GA D V} (hh : GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v h = 1) : Commute (locGL V τ β g) h

theorem **commute_locGL_locGL** {u : HeightOneSpectrum (𝒪 D.host.F)} (h : v ≠ u) (τ' : D.host.E →ₐ[D.host.F] Fv D.host.F u) {ι' : Type} [Fintype ι'] [DecidableEq ι'] (β' : LatFrame (Fv D.host.F u) ι' (Fin V.rank)) (g : GL ι (Fv D.host.F v)) (g' : GL ι' (Fv D.host.F u)) : Commute (locGL V τ β g) (locGL V τ' β' g')

theorem **mt_locGL_sub_one** (g : GL ι (Fv D.host.F v)) (i j : Fin V.rank) : (mt V (locGL V τ β g) - 1) i j = tensAt D.host.E v (((GlobalTheory.AdelicU.matHom _ D.sp.cm.sig V.gram (locPt V τ β g)).val - 1) i j)

theorem **rTrans_mem_levelFun** (x : GA D V) {f : GA D V → ℂ} (hf : f ∈ levelFun V) : rTrans x f ∈ levelFun V

theorem **isFinStable_levelFun** : IsFinStable V (levelFun V)

theorem **locReg_apply** (g : GL ι (Fv D.host.F v)) (f : GA D V → ℂ) : locReg V τ β g f = rTrans (locGL V τ β g) f

theorem **coe_locRepOn** (U : Submodule ℂ (GA D V → ℂ)) (hU : IsFinStable V U) (g : GL ι (Fv D.host.F v)) (f : U) : (locRepOn V τ β U hU g f : GA D V → ℂ) = locReg V τ β g f

theorem **isOpen_stab** {f : GA D V → ℂ} (hf : f ∈ levelFun V) : IsOpen {g | locReg V τ β g f = f}

theorem **isSmoothRep_locRepOn** {U : Submodule ℂ (GA D V → ℂ)} (hU : IsFinStable V U) (h : U ≤ levelFun V) : IsSmoothRep (locRepOn V τ β U hU)

theorem **downIter_zero** (q : ℕ) (l : A → ℂ) : downIter ν σ q 0 l = l

theorem **downIter_succ** (q d : ℕ) (l : A → ℂ) : downIter ν σ q (d + 1) l = downIter ν σ q d (exDown ν (q + d) σ l)

theorem **upIter_zero** (p : ℕ) (l : A → ℂ) : upIter ν σ p 0 l = l

theorem **upIter_succ** (p d : ℕ) (l : A → ℂ) : upIter ν σ p (d + 1) l = exUp ν (p + d) σ (upIter ν σ p d l)

theorem **exTo_of_ge** {p q : ℕ} (h : q ≤ p) (l : A → ℂ) : exTo ν σ p q l = downIter ν σ q (p - q) l

theorem **exTo_of_le** {p q : ℕ} (h : p ≤ q) (l : A → ℂ) : exTo ν σ p q l = upIter ν σ p (q - p) l

theorem **exTo_self** (p : ℕ) (l : A → ℂ) : exTo ν σ p p l = l

theorem **exTo_comp** (σ' : Representation ℂ (GL (n ⊕ Fin c) k) A') (f : A' → A) (hf : ∀ g x, f (σ' g x) = σ g (f x)) (z : ℂ) (p q : ℕ) (l : A → ℂ) : exTo ν σ' p q (fun x => z * l (f x)) = fun x => z * exTo ν σ p q l (f x)

theorem **exTo_law** (σ' : Representation ℂ (GL (n ⊕ Fin c) k) A') {T : A' → A} (hT : ∀ g x, T (σ' g x) = σ g (T x)) {l : A → ℂ} {l' : A' → ℂ} {z : ℂ} (h : ∀ x, l (T x) = z * l' x) (p q : ℕ) (x : A') : exTo ν σ p q l (T x) = z * exTo ν σ' p q l' x

theorem **exTo_isNEquiv** (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (hσ : IsSmoothRep σ) {p q : ℕ} (hp : 1 ≤ p) (hpc : p ≤ c) (hq : 1 ≤ q) (hqc : q ≤ c) (l : A →ₗ[ℂ] ℂ) (hl : IsNEquiv ψ p σ l) : ∃ l' : A →ₗ[ℂ] ℂ, ⇑l' = exTo ν σ p q l ∧ IsNEquiv ψ q σ l'

theorem **exList_law** (σ' : ∀ i, Representation ℂ (GL (n ⊕ Fin c) (k i)) A') {T : A' → A} (hT : ∀ i g x, T (σ' i g x) = σ i g (T x)) {l : A → ℂ} {l' : A' → ℂ} {z : ℂ} (h : ∀ x, l (T x) = z * l' x) (p q : I → ℕ) (L : List I) (x : A') : exList ν σ p q L l (T x) = z * exList

ν σ' p q L l' x

theorem **exList_self** (p : I → ℕ) (L : List I) (l : A →₀ ℂ) : exList ν σ p p L l = l

theorem **exList_isNEquiv** [DecidableEq I] (hψ : ∀ i, Continuous (ψ i)) (hψ1 : ∀ i, ψ i ≠ 1) (hσ : ∀ i, IsSmoothRep (σ i)) (hcom : ∀ i j, i ≠ j → ∀ g h x, σ i g (σ j h x) = σ j h (σ i g x)) {p q : I → ℕ} (hp : ∀ i, 1 ≤ p i ∧ p i ≤ c) (hq : ∀ i, 1 ≤ q i ∧ q i ≤ c) (l : A →₀[ℂ] ℂ) (hl : ∀ i, IsNEquiv (ψ i) (p i) (σ i) l) {L : List I} (hL : L.Nodup) : ∃ l' : A →₀[ℂ] ℂ, ⇑l' = exList ν σ p q L l ∧ ∀ i, IsNEquiv (ψ i) (if i ∈ L then q i else p i) (σ i) l'

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.MixedModelTargetHeisenbergAnisotropicPart

theorem **symplOH_add_self** (S T : HSp D) (x y : WO S T) : symplOH S T x y + symplOH S T x y = symplO S T x y

theorem **smul_cancel** {R : Type} [CommRing R] {n k l : ℕ} {a c : R} (h : a * c = 1) (P G : Matrix (Fin n) (Fin n) R) (hP : P * G = 1) (y : Matrix (Fin n) (Fin k) R) (b : Matrix (Fin k) (Fin l) R) : a • (P * (c • (G * y) * b)) = y * b

theorem **mixS_eq** (f : WO S T → ℂ) (x : Mx D T.rank ρ) (w₀ : WO S₀ T) : d.mixS T ψ μ ℱ f x w₀ = ∫ a in ℱ, ψ (symplOH S T (d.extY T a) (d.extX T x)) * f (d.extY T a + d.extX T x + d.ext0 T w₀) ∂μ

theorem **mixS_heis_extX** (f : WO S T → ℂ) (u x : Mx D T.rank ρ) : d.mixS T ψ μ ℱ (heisO ψ S T (d.extX T u) f) x = d.mixS T ψ μ ℱ f (x + u)

theorem **mixedModelSource_heisenberg_anisotropic_part** (f : WO S T → ℂ) (u : WO S₀ T) (x : Mx D T.rank ρ) : d.mixS T ψ μ ℱ (heisO ψ S T (d.ext0 T u) f) x = heisO ψ S₀ T u (d.mixS T ψ μ ℱ f x)

theorem **mixS_heis_extY** (hμ : IsCovolOne (ratMx D T.rank ρ) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (y x : Mx D T.rank ρ) (w₀ : WO S₀ T) : d.mixS T ψ μ ℱ (heisO ψ S T (d.extY T y) f) x w₀ = ψ (symplO S T (d.extX T x) (d.extY T y)) * d.mixS T ψ μ ℱ f x w₀

theorem **mixS_quasi** (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (x : Mx D T.rank ρ) (γ : Matrix (Fin T.rank) (Fin S₀.rank) D.host.E) (w₀ : WO S₀ T) : d.mixS T ψ μ ℱ f x (ratW T S₀ γ + w₀) = ψ (-(symplOH S₀ T (ratW T S₀ γ) w₀)) * d.mixS T ψ μ ℱ f x w₀

theorem **mixS_rank_zero** (d : HypDec S S₀ 0) {μ : Measure (Mx D T.rank 0)} {ℱ : Set (Mx D T.rank 0)} (hμ : IsCovolOne (ratMx D T.rank 0) μ ℱ) (f : WO S T → ℂ) (x : Mx D T.rank 0) (w₀ : WO S₀ T) : d.mixS T ψ μ ℱ f x w₀ = f (d.ext0 T w₀)

theorem **mixT_eq** (f : WO S T → ℂ) (y : Mx D S.rank ρ) (w₀ : WO S T₀) : d.mixT S ψ μ ℱ f y w₀ = ∫ a in ℱ, ψ (symplOH S T (d.incX S a) (d.incAdj S y)) * f (d.incX S a + d.incAdj S y + d.inc0 S w₀) ∂μ

theorem **mixT_heis_incAdj** (f : WO S T → ℂ) (u y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (heisO ψ S T (d.incAdj S u) f) y = d.mixT S ψ μ ℱ f (y + u)

theorem **mixedModelTarget_heisenberg_anisotropic_part** (f : WO S T → ℂ) (u : WO S T₀) (y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (heisO ψ S T (d.inc0 S u) f) y = heisO ψ S T₀ u (d.mixT S ψ μ ℱ f y)

theorem **mixT_heis_incX** (hμ : IsCovolOne (ratMx D ρ S.rank) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (ξ : Mx D ρ S.rank) (y : Mx D S.rank ρ) (w₀ : WO S T₀) : d.mixT S ψ μ ℱ (heisO ψ S T (d.incX S ξ) f) y w₀ = ψ (symplO S T (d.incAdj S y) (d.incX S ξ)) * d.mixT S ψ μ ℱ f y w₀

theorem **adjoint_nB_incAdj** (b : Mx D ρ T₀.rank) (y : Mx D S.rank ρ) : incA D D.host.xi0 • (giA S · (adj (d.p0 · (mt T (d.nB b)⁻¹ · d.incAdj S y)) · gA T₀)) = y · b

theorem **nC_injective** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : Function.Injective d.nC

theorem **levi_injective** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : Function.Injective d.levi

theorem **nC_xi_ne_one** (D : D7) : (hypDecThree D).nC (xiSkew D 1) ≠ 1

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.NegNotMemSpecialLinearPartUnitaryOneOne

theorem **uPart_fix_J** {L : Submodule ℂ (Fin n → ℂ)} {g : UJ ε} (hg : g ∈ uPart ε L) {y : Fin n → ℂ} (hy : y ∈ L) : uMat g *ᵥ (Jm ε *ᵥ y) = Jm ε *ᵥ y

theorem **uPart_le** (hW : IsIsoFlag ε j W) : uPart ε (W (Fin.last a)) ≤ leviOne ε W

theorem **glPart_inf_uPart** (hW : IsIsoFlag ε j W) : glPart ε W ⊓ uPart ε (W (Fin.last a)) = ⊥

theorem **glPart_zero** {j : Fin 0 → ℕ} {W : Flag n 0} (hW : IsIsoFlag ε j W) : glPart ε W = ⊥

theorem **uMat_neg** : uMat (neg ε) = -1

theorem **neg_mem_leviOne** : neg ε ∈ leviOne ε W

theorem **neg_mem_glPart_iff** : neg ε ∈ glPart ε W ↔ ∀ y : Fin n → ℂ, (∀ z ∈ W (Fin.last a), star z ⬝ᵥ (Jm ε *ᵥ y) = 0) → y ∈ W (Fin.last a)

theorem **neg_mem_slPart_iff** : neg ε ∈ slPart ε W ↔ neg ε ∈ glPart ε W ∧ ∀ i, Even (Module.finrank ℂ (W i))

theorem **neg_mem_generalLinearPart_unitary_one_one** : neg e2 ∈ glPart e2 W2

theorem **neg_notMem_specialLinearPart_unitary_one_one** : neg e2 ∉ slPart e2 W2

theorem **mU_notMem_glPart** : mU ∉ glPart e2 W2

theorem **finrank_siegel_flag** : Module.finrank ℂ (LinearMap.range emb) = 2

theorem **isIsotropicFlag_siegel** : IsIsoFlag e4 ![2] W4

theorem **neg_mem_specialLinearPart_siegel** : neg e4 ∈ slPart e4 W4 ∧ neg e4 ≠ 1

theorem **reflection_mem_unitaryPart** : refl3 ∈ uPart e3 (Submodule.span ℂ {![1, 1, 0]}) ∧ refl3 ≠ 1 ∧ neg e3 ∉ uPart e3 (Submodule.span ℂ {![1, 1, 0]})

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.NodeYPredEqChainVertexAtRowZeroLast

theorem **exists_gapCond** : ∃ γ, GapCond D γ

theorem **isLeast_gap** : IsLeast {γ | GapCond D γ} (gap D)

theorem **grid_nearMid** (τ : D.what₂.Tau) : (grid D τ).NearMid

theorem **rk_eq** (τ : D.what₂.Tau) (i j : ℕ) : rkV D i j = (grid D τ).rankD i j ∧ rkP D i = (grid D τ).rankP i ∧ rkX D j = (grid D τ).rankX j

theorem **parV_zero** {j : ℕ} (hj : j ≤ D.R.m) : parV D 0 j = parB (Ptw D) j

theorem **parP_m** : parP D D.R.m = parXp D 0

theorem **ideal_ne_bot** : Z.ideal Pg ≠ ⊥

theorem **nodeY_eq_chainVertexAtRowZero_of_lt** {j : ℕ} (h : j < D.R.m) : nodeY R j = row0 R j

theorem **nodeX_eq_chainVertexAtRowZero_of_ge** {j : ℕ} (h : ¬ j + 1 < D.R.m) : nodeX R = row0 R (j + 1)

theorem **nodeX_eq_chainVertexAtRowZero_top** : nodeX R = row0 R D.R.m

theorem **nodeY_pred_eq_chainVertexAtRowZero_last** {j : ℕ} (h : j + 1 = D.R.m) : nodeY R (D.R.m − 1) = row0 R j

theorem **meas_zero** (t : T) (i : ℕ) : G.meas t i 0 = G.measCol i

theorem **th_It** (t t' : T) (i j : ℕ) : (G.th t i j).It = (G.th t' i j).It

theorem **RealVec.vec_zero** {G : GridPre R} (r : G.RealVec) : r.vec 0 = PiTensorProduct.tprod ℂ r.w

theorem **RealVec.vec_succ** {G : GridPre R} (r : G.RealVec) (i : ℕ) : r.vec (i + 1) = (G.X i 0).It (r.pol i) (r.vec i)

theorem **colLift_zero** (t : T) (j : ℕ) (f : ∀ i, WO (G.V (i + 1) j).V (G.V i j).V → ℂ) : G.colLift t j f 0 = LinearMap.id

theorem **colLift_succ** (t : T) (j : ℕ) (f : ∀ i, WO (G.V (i + 1) j).V (G.V i j).V → ℂ) (n : ℕ) : G.colLift t j f (n + 1) = G.ioX t n j (f n) ∘ₗ G.colLift t j f n

theorem **rowLift_zero** (t : T) (i : ℕ) (f : ∀ j, WO (G.V i j).V (G.V i (j + 1)).V → ℂ) : G.rowLift t i f 0 = LinearMap.id

theorem **rowLift_succ** (t : T) (i : ℕ) (f : ∀ j, WO (G.V i j).V (G.V i (j + 1)).V → ℂ) (n : ℕ) : G.rowLift t i f (n + 1) = G.rowLift t i f n ∘ₗ G.ioT t i n (f n)

theorem **chiLift_zero** (Φ : G.Data R) : G.chiLift Φ 0 = LinearMap.id

theorem **chiLift_succ** (Φ : G.Data R) (n : ℕ) : G.chiLift Φ (n + 1) = G.gridInclusionColumnZero n (G.datum Φ n) ∘ₗ G.chiLift Φ n

theorem **chiLift_eq** (t : T) (Φ : G.Data R) (n : ℕ) : G.chiLift Φ n = G.colLift t 0 (G.datum Φ) n

theorem **IsLifts.hasLifts** {G : GridDat R} (h : G.IsLifts) : G.toGridPre.HasLifts

theorem **GridPre.exists_gridDat** {R : RealDat D A Ks} (G : GridPre R) (hG : G.Is) (h : G.HasLifts) : ∃ G' : GridDat R, G'.toGridPre = G ∧ G'.Is

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.NotInSupportOfSubsingleton

theorem **evalOne_bijective** (e : (Σ a, m a) ≃ n) {π : Representation ℂ (GL n k) V} (hπ : IsSmooth π) (ρ : Representation ℂ (Levi k m) W) : Function.Bijective (evalOne e : π.IntertwiningMap (parInd e ρ) → _)

theorem **scalar_of_isSubquotient** {G : Type} [Group G] {σ : Representation ℂ G W} {ρ : Representation ℂ G V} (h : IsSubquotient σ ρ) {g : G} {c : ℂ} (hc : ∀ v, ρ g v = c • v) (w : W) : σ g w = c • w

theorem **nontrivial_of_isIrreducible** {G : Type} [Group G] {σ : Representation ℂ G W} (h : σ.IsIrreducible) : Nontrivial W

theorem **jac_detRep** (e : (Σ a, m a) ≃ n) (χ : kˣ →* ℂˣ) (g : Levi k m) (x : Jacquet (detRep n χ) (unipRad k e)) : jac e (detRep n χ) g x = (detLevi e χ g : ℂ) • x

theorem **inSupport_detRep_iff** {N : ℕ} (χ ψ : kˣ →* ℂˣ) : InSupport (detRep (Fin N) χ) ψ ↔ ∃ i : Fin N, ψ = χ * absPow k (((i : ℕ) : ℂ) − ((N : ℂ) − 1) / 2)

theorem **inSupport_one_iff** (χ ψ : kˣ →* ℂˣ) : InSupport (detRep (Fin 1) χ) ψ ↔ ψ = χ

theorem **not_inSupport_zero** (π : Representation ℂ (GL (Fin 0) k) V) (ψ : kˣ →* ℂˣ) : ¬ InSupport π ψ

theorem **not_inSupport_of_subsingleton** [Subsingleton V] (π : Representation ℂ (GL n k) V) (ψ : kˣ →* ℂˣ) : ¬ InSupport π ψ

theorem **jhMult_ofChar_ne** {G : Type} [Group G] {χ ψ : G →* ℂˣ} (h : χ ≠ ψ) : jhMult (ofChar χ) (ofChar ψ) = 0

theorem **modulus_two** (ϖ : 𝒪[k]) (x : kˣ) (h : Irreducible ϖ) (hx : (x : k) = ϖ) : modulus k (fun _ : Fin 2 => Unit) (leviTorus k (Fin 2) ![x, 1]) = (resCard k : ℝ)⁻¹

theorem **completion** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : letI

theorem **not_inSupport_of_isCuspidal** [Nontrivial n] {π : Representation ℂ (GL n k) V} (h : IsCuspidal π) (ψ : kˣ →* ℂˣ) : ¬ InSupport π ψ

theorem **isCuspSupport_detRep** {N : ℕ} {χ : kˣ →* ℂˣ} (hχ : IsSmoothChar χ) : IsCuspSupport (detRep (Fin N) χ) (fun _ : Fin N => 1) fun i => detRep (Fin 1) (χ * absPow k (((i : ℕ) : ℂ) − ((N : ℂ) − 1) / 2))

theorem **hasSatake_one_unique** {π : Representation ℂ (GL (Fin 1) k) V} {a b : Fin 1 → ℂˣ} (ha : HasSatake π a) (hb : HasSatake π b) {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) : a = b

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.OffQOfSizeRows

theorem **part_m** (M : Vol G) (ρ : G.RealVec) (Φ : G.Datum) (u : GA D (Yc D 0) → ℂ) : part W M ρ Φ u D.R.m le_rfl = thetaChi W M ρ Φ u

theorem **offQ_of_sizeRows** {large₂ : SizeData₂ D} (h : SizeRows large₂) : OffQ D

theorem **dep_vac** (i : ℕ) (h : i < D.R.m) : Dep W (.chi i 0) (W.vac i)

theorem **All.gap** {Q : GP} (h : All Q) : Gap Q

theorem **Gap.ev** {Q : GP} (hs : SizeGap) (h : Gap Q) : Ev Q

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.OneMemNbh

theorem **one_mem_nbh** {m : ℕ} (ε : Fin m → Bool) {δ : ℝ} (h : 0 ≤ δ) : (1 : UJ ε) ∈ nbh ε δ

theorem **exp_entry** (X : Mat 1) : (NormedSpace.exp X) 0 0 = Complex.exp (X 0 0)

theorem **prod_entry** (l : List (Mat 1)) : l.prod 0 0 = (l.map fun X => X 0 0).prod

theorem **smV_chi** (L : Set (Mat 1)) (m : ℤ) (w : ℂ) : SmV L (chi m) w

theorem **dR_chi** (m : ℤ) (X : Mat 1) (w : ℂ) : dR (chi m) X w = m * X 0 0 * w

theorem **tensOne_is** (ε : Fin 1 → Bool) (m : ℤ) : (tensOne ε m).Is ε (fun _ => (m : ℝ)) (chi m)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ParRowZero

theorem **parRow_zero** (D : D7) (C : ℝ) : parRow D C 0 = parB (Ptw D) 0

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.PositionOneOfUnique

theorem **rat_xZeroQ** (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) : rat (xZeroQ dV dW) = dV.extX W (xZero dW)

theorem **rat_xLastQ** (dV : HypDec V V₀ (ρ + 1)) (dW : HypDec W W₀ ρ) (z' : Fin W₀.rank → D.host.E) : rat (xLastQ dV dW z') = dV.extX W (xLast dW z')

theorem **mul_qX_colN** {m : ℕ} (d : HypDec V V₀ ρ) (M : Matrix (Fin m) (Fin ρ) D.host.E) (i : ℕ) : (M * d.qX) *ᵥ colN d.e i = colN M i

theorem **normScale_coef** {θ : Fv D.host.F v} (ρ : ℕ) (a : D.host.E) (t₀ : Fv D.host.F v) {i : ℕ} (hi : i < ρ) : normScale θ (fun i => τ (coef ρ a i)) t₀ i = t₀ / θ ^ i

theorem **xZeroQ_first** {V V₀ W W₀ : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (z) (h : 0 < ρ) (t : Fv D.host.F v) : (xZeroQ dV dW).map τ *ᵥ (t • (τ ∘ besselAuxiliaryVectors dV z 0)) = t • (τ ∘ fourierJacobiVectors dW 0)

theorem **xLastQ_first** {V V₀ W W₀ : HSp D} {σ : ℕ} (dV : HypDec V V₀ (σ + 1)) (dW : HypDec W W₀ σ) (z) (t : Fv D.host.F v) : (xLastQ (srcDec dV) dW z).map τ *ᵥ ((t * τ D.host.xi0) • (τ ∘ fourierJacobiVectors dV 0)) = t • (τ ∘ besselAuxiliaryVectors dW z 0)

theorem **exists_isRule** : ∃ β, IsRule C ψ hψ τ γ β

theorem **IsRule.source** (H : IsRule C ψ hψ τ γ β) (i ρ : ℕ) (h : i + ρ = s) (hz : hQ (C.L i) (C.ζ i) (C.ζ i) = hQ (C.L (i + 1)) (C.ζ (i + 1)) (C.ζ (i + 1))) : IsFJFrameS ψ (srcDec (C.dO i ρ h)) (C.ζ (i + 1)) (weilChar ψ hψ τ) τ (β (2 * i + 1) (2 * ρ) (by omega)) (γ (2 * i + 1))

theorem **vec_ne_zero** {k ι Z : Type} [Field k] [Fintype ι] [DecidableEq ι] [Fintype Z] [DecidableEq Z] (b : LatFrame k ι Z) (i : ι) : vec b i ≠ 0

theorem **IsRule.isScaled_even** (H : IsRule C ψ hψ τ γ β) (h1 : weilChar ψ hψ τ ≠ 1) (i ρ : ℕ) (h : i + ρ = s) : IsScaled _ _ τ ((C.dE i ρ h).i₀ * (C.oE i (Nat.le.intro h)).iA) (besselAuxiliaryVectors (C.dE i ρ h) (C.z₀ i ρ h)) (γ (2 * i)) (β (2 * i) (2 * ρ + 1) (by omega)) (normScale (τ D.host.xi0) (fun l => τ (coef (ρ + 1) (-hQ (C.L i) (C.ζ i) (C.ζ i)) l)) (τ D.host.xi0 ^ ρ))

theorem **IsRule.middle** (H : IsRule C ψ hψ τ γ β) (h1 : weilChar ψ hψ τ ≠ 1) (i ρ : ℕ) (h : i + ρ = s) : vec (β (2 * i) (2 * ρ + 1) (by omega)) (.inr (ρ, by omega)) = τ ∘ C.z i ρ h

theorem **locGLs_mulSingle** [DecidableEq S] {ι : Type} [Fintype ι] [DecidableEq ι] (β : ∀ v : S, LatFrame (Fv D.host.F v) ι (Fin V.rank)) (v : S) (g : GL ι (Fv D.host.F v)) : locGLs V τ β (Pi.mulSingle v g) = locGL V (τ v) (β v) g

theorem **locGLs_mem_GfC** {ι : Type} [Fintype ι] [DecidableEq ι] (β : ∀ v : S, LatFrame (Fv D.host.F v) ι (Fin V.rank)) (g : ∀ v : S, GL ι (Fv D.host.F v)) : locGLs V τ β g ∈ GfC V

theorem **posAt_self** (p : S → ℕ) (l : (GA D V → ℂ) → ℂ) : posAt V τ β ν p p l = l

theorem **positionOne_of_unique** [Unique S] (l : (GA D V → ℂ) → ℂ) : pos1 V τ β ν l = downIter (ν default) (locReg V (τ default) (β default)) 1 (c / 2) l

theorem **posAt_of_isEmpty** [IsEmpty S] (p q : S → ℕ) (l : (GA D V → ℂ) → ℂ) : posAt V τ β ν p q l = l

theorem **posAt_coe** (p q : S → ℕ) (l : (GA D V → ℂ) → ℂ) (F : U) : posAt V τ β ν p q l F = exList ν (fun v => locRepOn V (τ v) (β v) U hU) p q (Finset.univ : Finset S).toList (fun F : U => l F) F

theorem **posAt_congr** (p q : S → ℕ) {l l' : (GA D V → ℂ) → ℂ} (h : ∀ F ∈ U, l F = l' F) {F : GA D V → ℂ} (hF : F ∈ U) : posAt V τ β ν p q l F = posAt V τ β ν p q l' F

theorem **posAt_rTrans** (p q : S → ℕ) (l : (GA D V → ℂ) → ℂ) {x : GA D V} (hx : ∀ v : S, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v x = 1) (F : GA D V → ℂ) : posAt V τ β ν p q l (rTrans x F) = posAt V τ β ν p q (fun F => l (rTrans x F)) F

theorem **posAt_isNEquiv** [∀ v, (ν v).IsAddHaarMeasure] (hU' : U ≤ levelFun V) {ψv : ∀ v : S, AddChar (Fv D.host.F v) Circle} (hψ : ∀ v, Continuous (ψv v)) (hψ1 : ∀ v, ψv v ≠ 1) {p q : S → ℕ} (hp : ∀ v, 1 ≤ p v ∧ p v ≤ c) (hq : ∀ v, 1 ≤ q v ∧ q v ≤ c) {l : (GA D V → ℂ) → ℂ} (l₀ : U →ₗ[ℂ] ℂ) (h₀ : ∀ F : U, l F = l₀ F) (hl : ∀ v, IsNEquiv (ψv v) (p v) (locRepOn V (τ v) (β v) U hU) l₀) : ∃ l₁ : U →ₗ[ℂ] ℂ, (∀ F : U, posAt V τ β ν p q l F = l₁ F) ∧ ∀ v, IsNEquiv (ψv v) (q v) (locRepOn V (τ v) (β v) U hU) l₁

theorem **besPos1_eq** (hc : c = 2 * ρ + 1) (φ : GA D A → ℂ) : besPos1 τ β ν ψ d o ζ μ 𝓕 νN 𝒩 φ = pos1 V τ β ν (besPer ψ 1 d o ζ μ 𝓕 νN 𝒩 φ)

theorem **locChar_ne_one** {F : Type} [Field F] [NumberField F] {ψ : AddChar (AF F) ℂ} (hψ : ∀ x, ‖ψ x‖ = 1) {v : HeightOneSpectrum (𝒪 F)} (h : ∃ y, psiV F ψ v y ≠ 1) {a : Fv F v} (ha : a ≠ 0) : locChar ψ hψ a ≠ 1

theorem **weilChar_spec** {ψ : AddChar (AF D.host.F) ℂ} (h : IsStdChar D ψ (-1)) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : Continuous (weilChar ψ h.1.2.1 τ) ∧ weilChar ψ h.1.2.1 τ ≠ 1

theorem **weilChar_piW** (ψ : AddChar (AF D.host.F) ℂ) (hψ : ∀ x, ‖ψ x‖ = 1) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (x : AE D) : (weilChar ψ hψ τ (piW τ x) : ℂ) = psiE ψ (idem τ * (incA D D.host.xi0 * x))

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ProjExists

theorem **embH_one** (e : Option (Fin n) ≃ Fin (n + 1)) : embH e 1 = 1

theorem **embH_mul** (e : Option (Fin n) ≃ Fin (n + 1)) (h h' : Mat n) : embH e (h * h') = embH e h * embH e h'

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.SqNormIntegralFundamentalDomainEqL2NormSq

@[simp] theorem **wordAct_nil** (x : X) : wordAct A [] x = x

@[simp] theorem **wordAct_cons** (i : ι) (l : List ι) (x : X) : wordAct A (i :: l) x = A i (wordAct A l x)

theorem **wordAct_append** (l l' : List ι) (x : X) : wordAct A (l ++ l') x = wordAct A l (wordAct A l' x)

theorem **wordAct_rel** {Y : Type} {A : ι → X → X} {B : ι → Y → Y} {R : X → Y → Prop} (l : List ι) (h : ∀ i ∈ l, ∀ x y, R x y → R (A i x) (B i y)) {x : X} {y : Y} (hx : R x y) : R (wordAct A l x) (wordAct B l y)

@[simp] theorem **wordSum_zero** (x : X) : wordSum A p 0 x = p x

theorem **wordSum_one** (x : X) : wordSum A p 1 x = p x + ∑ i, p (A i x)

theorem **le_wordSum** {d : ℕ} {l : List ι} (hl : l.length ≤ d) (x : X) : p (wordAct A l x) ≤ wordSum A p d x

theorem **wordSum_mono** {d d' : ℕ} (h : d ≤ d') (x : X) : wordSum A p d x ≤ wordSum A p d' x

theorem **wordSum_ne_top_iff** {d : ℕ} {x : X} : wordSum A p d x ≠ ∞ ↔ ∀ l : List ι, l.length ≤ d → p (wordAct A l x) ≠ ∞

theorem **wordSum_le_add** {d : ℕ} {x y z : X} (h : ∀ l, p (wordAct A l z) ≤ p (wordAct A l x) + p (wordAct A l y)) : wordSum A p d z ≤ wordSum A p d x + wordSum A p d y

theorem **wordSum_eq_mul** {d : ℕ} {x y : X} {c : ℝ≥0∞} (h : ∀ l, p (wordAct A l y) = c * p (wordAct A l x)) : wordSum A p d y = c * wordSum A p d x

theorem **wordSum_letter_le** (d : ℕ) (i : ι) (x : X) : wordSum A p d (A i x) ≤ wordSum A p (d + 1) x

theorem **height_laws** : (∀ g, 1 ≤ height V g) ∧ height V 1 = 1 ∧ (∀ g, height V g⁻¹ = height V g) ∧ (∀ g h, height V (g * h) ≤ height V g * height V h) ∧ Continuous (height V)

theorem **quotHeight_laws** : (∀ g, 1 ≤ quotHeight V g ∧ quotHeight V g ≤ height V g) ∧ (∀ γ ∈ GQ V, ∀ g, quotHeight V (γ * g) = quotHeight V g) ∧ (∀ g h, quotHeight V (g * h) ≤ quotHeight V g * height V h ∧ quotHeight V g / height V h ≤ quotHeight V (g * h)) ∧ Continuous (quotHeight V)

theorem **rWord_const** (l : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin V.rank) ℂ)) (c : ℂ) : ∃ c' : ℂ, rWord θ V l (fun _ => c) = fun _ => c'

theorem **rapidDecSm_const** (hV : ∀ τ, (gramAt D V τ).PosDef) (c : ℂ) : RapidDecSm θ V fun _ => c

theorem **gramAt_posSp** (n : ℕ) (τ : D.what₂.Tau) : gramAt D (posSp D n) τ = 1

theorem **rapidDecSm_one_posSp** (n : ℕ) : RapidDecSm θ (posSp D n) fun _ => 1

theorem **growthSize_le_iff** {C : ℝ} (hC : 0 ≤ C) : growthSize V N φ ≤ ENNReal.ofReal C ↔ ∀ g, ‖φ g‖ ≤ C * height V g ^ N

theorem **modGrowthExp_iff** : ModGrowthExp V N φ ↔ growthSize V N φ ≠ ∞

theorem **growthSize_eq_quot** (hφ : ∀ γ ∈ GQ V, ∀ g, φ (γ * g) = φ g) : growthSize V N φ = ⊔ g, ENNReal.ofReal (‖φ g‖ / quotHeight V g ^ N)

theorem **growthSize_anti** {N' : ℕ} (h : N ≤ N') : growthSize V N' φ ≤ growthSize V N φ

theorem **growthSize_add_le** : growthSize V N (φ + φ') ≤ growthSize V N φ + growthSize V N φ'

theorem **growthSize_smul** (c : ℂ) : growthSize V N (c • φ) = ‖c‖ₑ * growthSize V N φ

theorem **growthSize_const** (c : ℂ) : growthSize V N (fun _ => c) = ‖c‖ₑ

theorem **unifModGrowth_iff** : UnifModGrowth V θ φ ↔ ∃ N, ∀ l, IsLieWord V l → growthSize V N (rWord θ V l φ) ≠ ∞

theorem **linOnSm_rWord** {l : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin V.rank) ℂ)} (hl : IsLieWord V l) : LinOnSm θ V (rWord θ V l)

theorem **linOnSm_glWord** (l : List (D.what₂.Tau × Fin V.rank × Fin V.rank)) : LinOnSm θ V (wordAct (glDer θ V) l)

theorem **growthSob_add_le** (h : φ ∈ SmFun θ V) (h' : φ' ∈ SmFun θ V) (d N : ℕ) : growthSob θ V d N (φ + φ') ≤ growthSob θ V d N φ + growthSob θ V d N φ'

theorem **growthSob_smul** (c : ℂ) (d N : ℕ) : growthSob θ V d N (c • φ) = ‖c‖ₑ * growthSob θ V d N φ

theorem **Rder_mem_allDer** {M : Submodule ℂ (GA D V → ℂ)} (h : φ ∈ allDer θ V M) {τ : D.what₂.Tau} {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) : Rder θ V τ X φ ∈ allDer θ V M

theorem **RderC_mem_allDer** {M : Submodule ℂ (GA D V → ℂ)} (h : φ ∈ allDer θ V M) (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : RderC θ V τ Z φ ∈ allDer θ V M

theorem **exists_growthSob_ne_top** (hs : φ ∈ SmFun θ V) (h : UnifModGrowth V θ φ) : ∃ N, ∀ d, growthSob θ V d N φ ≠ ∞

theorem **rapidDecSm_iff** : RapidDecSm θ V φ ↔ φ ∈ allDer θ V (rapidSub V) ∧ ∃ U : Subgroup (GA D V), IsOpen (U : Set (GA D V)) ∧ ∀ k ∈ U, k ∈ GfC V → ∀ g, φ (g * k) = φ g

theorem **RapidDecSm.rapidDec** (h : RapidDecSm θ V φ) : RapidDec V φ

theorem **RapidDecSm.unifModGrowth** (h : RapidDecSm θ V φ) : UnifModGrowth V θ φ

theorem **RapidDecSm.add** (h : RapidDecSm θ V φ) (h' : RapidDecSm θ V φ') : RapidDecSm θ V (φ + φ')

theorem **RapidDecSm.smul** (c : ℂ) (h : RapidDecSm θ V φ) : RapidDecSm θ V (c • φ)

theorem **Rder_rTr_fin** {k : GA D V} (hk : k ∈ GfC V) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin V.rank) ℂ) (φ : GA D V → ℂ) : Rder θ V τ X (rTr V k φ) = rTr V k (Rder θ V τ X φ)

theorem **RapidDecSm.rder** (h : RapidDecSm θ V φ) {τ : D.what₂.Tau} {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) : RapidDecSm θ V (Rder θ V τ X φ)

theorem **RapidDecSm.rTr_fin** (h : RapidDecSm θ V φ) {k : GA D V} (hk : k ∈ GfC V) : RapidDecSm θ V (rTr V k φ)

theorem **l2O_sq_eq_lintegral** (f : WO V W → ℂ) : l2O μ 𝓕 f ^ 2 = ∫⁻ w in 𝓕, ‖f w‖ₑ ^ 2 ∂μ

theorem **l2O_smul** (c : ℂ) (f : WO V W → ℂ) : l2O μ 𝓕 (c • f) = ‖c‖ₑ * l2O μ 𝓕 f

theorem **l2O_add_le_of_continuous** {f f' : WO V W → ℂ} (h : Continuous f) (h' : Continuous f') : l2O μ 𝓕 (f + f') ≤ l2O μ 𝓕 f + l2O μ 𝓕 f'

theorem **ipO_conj** (f f' : WO V W → ℂ) : conj (ipO μ 𝓕 f f') = ipO μ 𝓕 f' f

theorem **ipO_smul_left** (c : ℂ) (f f' : WO V W → ℂ) : ipO μ 𝓕 (c • f) f' = c * ipO μ 𝓕 f f'

theorem **ipO_smul_right** (c : ℂ) (f f' : WO V W → ℂ) : ipO μ 𝓕 f (c • f') = conj c * ipO μ 𝓕 f f'

theorem **ipO_self** (f : WO V W → ℂ) : ipO μ 𝓕 f f = nrm2 μ 𝓕 f

theorem **sqNormIntegralFundamentalDomain_eq_l2Norm_sq** {f : WO V W → ℂ} (h : Continuous f) : nrm2 μ 𝓕 f = (l2O μ 𝓕 f).toReal ^ 2

theorem **enorm_ipO_le** {f f' : WO V W → ℂ} (h : Continuous f) (h' : Continuous f') : ‖ipO μ 𝓕 f f'‖ₑ ≤ l2O μ 𝓕 f * l2O μ 𝓕 f'

theorem **exists_isCovolOne** : ∃ (μ : Measure (WO V W)) (𝓕 : Set (WO V W)), IsCovolOne (ratL W V) μ 𝓕

theorem **measure_dom** (hμ : μ.IsAddHaarMeasure) (h𝓕 : IsAddFundamentalDomain (ratL W V) 𝓕 μ) : μ 𝓕 ≠ 0 ∧ μ 𝓕 ≠ ∞

theorem **covolOne_unique** {μ' : Measure (WO V W)} {𝓕' : Set (WO V W)} (h : IsCovolOne (ratL W V) μ 𝓕) (h' : IsCovolOne (ratL W V) μ' 𝓕') : μ' = μ

theorem **mul_conj_periodic** (hψ : IsStdChar D ψ (-1)) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') : ∀ γ ∈ ratL W V, ∀ w, f (γ + w) * conj (f' (γ + w)) = f w * conj (f' w)

theorem **ipO_dom_eq** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (h𝓕 : IsAddFundamentalDomain (ratL W V) 𝓕 μ) {𝓕' : Set (WO V W)} (h𝓕' : IsAddFundamentalDomain (ratL W V) 𝓕' μ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') : ipO μ 𝓕 f f' = ipO μ 𝓕' f f'

theorem **memLp_theta** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (h𝓕 : IsAddFundamentalDomain (ratL W V) 𝓕 μ) (h : IsThetaFunO ψ V W f) : MemLp f 2 (μ.restrict 𝓕)

theorem **l2NormOnFundamentalDomain_thetaFunction_ne_top** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) : l2O μ ℱ f ≠ ∞

theorem **integrableOn_mul_conj** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') : IntegrableOn (fun w => f w * conj (f' w)) ℱ μ

theorem **ipO_add_left** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') (h'' : IsThetaFunO ψ V W f'') : ipO μ ℱ (f + f') f'' = ipO μ ℱ f f'' + ipO μ ℱ f' f''

theorem **ipO_add_right** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') (h'' : IsThetaFunO ψ V W f'') : ipO μ ℱ f (f' + f'') = ipO μ ℱ f f' + ipO μ ℱ f f''

theorem **l2O_eq_zero_iff_of_isThetaFunO** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) : l2O μ ℱ f = 0 ↔ f = 0

theorem **ipO_self_eq_zero_iff** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) : ipO μ ℱ f f = 0 ↔ f = 0

theorem **thetaL2_fundamentalDomain_indep** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) {ℱ' : Set (WO V W)} (hℱ' : IsAddFundamentalDomain (ratL W V) ℱ' μ) (h : IsThetaFunO ψ V W f) : l2O μ ℱ f = l2O μ ℱ' f

theorem **ipO_heisO** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') (u : WO V W) : ipO μ ℱ (heisO ψ V W u f) (heisO ψ V W u f') = ipO μ ℱ f f'

theorem **l2O_om_apply_eq_of_isThetaFunO** {χS χT : (AE D)ˣ →* Circle} {om : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} (hom : IsWeilDatumO ψ V W χS χT om) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (p : GA D V × GA D W) : l2O μ ℱ (om p f) = l2O μ ℱ f

theorem **ipO_om** {χS χT : (AE D)ˣ →* Circle} {om : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} (hom : IsWeilDatumO ψ V W χS χT om) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') (p : GA D V × GA D W) : ipO μ ℱ (om p f) (om p f') = ipO μ ℱ f f'

theorem **sobO_zero** : sobO ψ θ μ ℱ 0 f = l2O μ ℱ f

theorem **sobO_mono** {d d' : ℕ} (h : d ≤ d') : sobO ψ θ μ ℱ d f ≤ sobO ψ θ μ ℱ d' f

theorem **sobO_heisDer_le** (d : ℕ) (a : D.what₂.Tau × Fin W.rank × Fin V.rank × Bool) : sobO ψ θ μ ℱ d (heisDer ψ θ a f) ≤ sobO ψ θ μ ℱ (d + 1) f

theorem **dRho_smul** (τ : D.what₂.Tau) (z : Matrix (Fin W.rank) (Fin V.rank) ℂ) (c : ℂ) (f : WO V W → ℂ) : Formalism.IndefR.Datum.dRho (heisO ψ V W) θ τ z (c • f) = c • Formalism.IndefR.Datum.dRho (heisO ψ V W) θ τ z f

theorem **dRho_add** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (τ : D.what₂.Tau) (z : Matrix (Fin W.rank) (Fin V.rank) ℂ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') : Formalism.IndefR.Datum.dRho (heisO ψ V W) θ τ z (f + f') = Formalism.IndefR.Datum.dRho (heisO ψ V W) θ τ z f + Formalism.IndefR.Datum.dRho (heisO ψ V W) θ τ z f'

theorem **theta_word** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (l : List (D.what₂.Tau × Fin W.rank × Fin V.rank × Bool)) (h : IsThetaFunO ψ V W f) : IsThetaFunO ψ V W (wordAct (heisDer ψ θ) l f)

theorem **sobO_ne_top** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (d : ℕ) : sobO ψ θ μ ℱ d f ≠ ∞

theorem **sobO_smul** (d : ℕ) (c : ℂ) : sobO ψ θ μ ℱ d (c • f) = ‖c‖ₑ * sobO ψ θ μ ℱ d f

theorem **sobO_add_le** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (h : IsThetaFunO ψ V W f) (h' : IsThetaFunO ψ V W f') (d : ℕ) : sobO ψ θ μ ℱ d (f + f') ≤ sobO ψ θ μ ℱ d f + sobO ψ θ μ ℱ d f'

theorem **sobO_eq_zero_iff** (hψ : IsStdChar D ψ (-1)) (hμ : μ.IsAddHaarMeasure) (hℱ : IsAddFundamentalDomain (ratL W V) ℱ μ) (h : IsThetaFunO ψ V W f) (d : ℕ) : sobO ψ θ μ ℱ d f = 0 ↔ f = 0

theorem **isTheta_iff_level** : IsThetaFunO ψ V W f ↔ ∃ U : AddSubgroup (WO V W), IsOpen (U : Set (WO V W)) ∧ f ∈ thetaLevel ψ V W U

theorem **thetaLevel_anti** {U U' : AddSubgroup (WO V W)} (h : U ≤ U') : thetaLevel ψ V W U' ≤ thetaLevel ψ V W U

theorem **dRho_mem_thetaLevel** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {U : AddSubgroup (WO V W)} (h : f ∈ thetaLevel ψ V W U) (τ : D.what₂.Tau) (z : Matrix (Fin W.rank) (Fin V.rank) ℂ) : Formalism.IndefR.Datum.dRho (heisO ψ V W) θ τ z f ∈ thetaLevel ψ V W U

theorem **exists_theta_posSp** (D : D7) (m n : ℕ) : ∃ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (μ : Measure (WO (posSp D m) (posSp D n))) (ℱ : Set (WO (posSp D m) (posSp D n))) (U : AddSubgroup (WO (posSp D m) (posSp D n))) (f : WO (posSp D m) (posSp D n) → ℂ), IsStdChar D ψ (-1) ∧ IsInfFam θ ∧ IsCovolOne (ratL (posSp D n) (posSp D m)) μ ℱ ∧ IsOpen (U : Set (WO (posSp D m) (posSp D n))) ∧ f ∈ thetaLevel ψ _ _ U ∧ ipO μ ℱ f f ≠ 0 ∧ ∀ d, sobO ψ θ μ ℱ d f ≠ 0 ∧ sobO ψ θ μ ℱ d f ≠ ∞

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.StandardOrderExtendedBasisApplyLowBlock

theorem **Level.mem_map** {G G' : Type*} [Group G] [Group G'] {φ : G ≃* G'} {P : Level G} {h : G'} : h ∈ (P.map φ).J ↔ φ.symm h ∈ P.J

theorem **Level.map_lam** {G G' : Type*} [Group G] [Group G'] (φ : G ≃* G') (P : Level G) (h : (P.map φ).J) : (P.map φ).lam h = P.lam ⟨φ.symm h, h.2⟩

theorem **extZ_coe** (J : Subgroup (GL a k)) (ρ : J →* ℂˣ) (g : J) : extZ J ρ (g : GL a k).val = ρ g

theorem **extZ_of_not** (J : Subgroup (GL a k)) (ρ : J →* ℂˣ) {M : Matrix a a k} (h : ¬ ∃ g ∈ J, g.val = M) : extZ J ρ M = 0

theorem **extZ_ne_zero_iff** (J : Subgroup (GL a k)) (ρ : J →* ℂˣ) {M : Matrix a a k} : extZ J ρ M ≠ 0 ↔ ∃ g ∈ J, g.val = M

theorem **reGL_val** (e : a ≃ b) (g : GL a k) : (reGL e g).val = g.val.submatrix e.symm e.symm

theorem **reGL_symm_val** (e : a ≃ b) (g : GL b k) : ((reGL e).symm g).val = g.val.submatrix e e

theorem **det_reGL** (e : a ≃ b) (g : GL a k) : Matrix.GeneralLinearGroup.det (reGL e g) = Matrix.GeneralLinearGroup.det g

theorem **embTop_val** (g : GL a k) : (embTop t g).val = fromBlocks g.val 0 0 1

theorem **mul_toBlocks₁₁** (M M' : Matrix (a ⊕ t) (a ⊕ t) k) : (M * M').toBlocks₁₁ = M.toBlocks₁₁ * M'.toBlocks₁₁ + M.toBlocks₁₂ * M'.toBlocks₂₁

theorem **mul_toBlocks₂₁** (M M' : Matrix (a ⊕ t) (a ⊕ t) k) : (M * M').toBlocks₂₁ = M.toBlocks₂₁ * M'.toBlocks₁₁ + M.toBlocks₂₂ * M'.toBlocks₂₁

theorem **ofResidue_lam** (H : Subgroup (GL a 𝒦[k])) (χ : H →* ℂˣ) (u : GL a 𝒪[k]) (hu : Matrix.GeneralLinearGroup.map (n

theorem **isOpen** : IsOpen (P.J : Set (GL a k))

theorem **isCompact** : IsCompact (P.J : Set (GL a k))

theorem **exists_of_congr** {g : GL a k} (hg : g ∈ P.J) {M : Matrix a a k} (h : MaxM (M − g.val)) : ∃ g' ∈ P.J, g'.val = M

theorem **rho_apply** (μ : kˣ →* ℂˣ) (g : P.J) : P.rho μ g = P.lam g * (μ (Matrix.GeneralLinearGroup.det (g : GL a k)))⁻¹

theorem **rho_congr** {μ : kˣ →* ℂˣ} (hμ : IsTame μ) {g g' : P.J} (h : MaxM ((g' : GL a k).val − (g : GL a k).val)) : P.rho μ g' = P.rho μ g

theorem **mem_stepJ** {h : GL (a ⊕ t) k} : h ∈ P.stepJ t ↔ h ∈ KO k (a ⊕ t) ∧ (∃ g ∈ P.J, g.val = h.val.toBlocks₁₁) ∧ IntM h.val.toBlocks₁₂ ∧ MaxM h.val.toBlocks₂₁ ∧ ∃ D ∈ KO k t, D.val = h.val.toBlocks₂₂

theorem **mem_stepJ_iff** {h : GL (a ⊕ t) k} : h ∈ P.stepJ t ↔ h ∈ KO k (a ⊕ t) ∧ (∃ g ∈ P.J, g.val = h.val.toBlocks₁₁) ∧ MaxM h.val.toBlocks₂₁

theorem **corner_val** (h : P.stepJ t) : (P.corner h : GL a k).val = (h : GL (a ⊕ t) k).val.toBlocks₁₁

theorem **corner_mul** (h h' : P.stepJ t) : MaxM ((P.corner (h * h') : GL a k).val − ((P.corner h * P.corner h' : P.J) : GL a k).val)

theorem **step_J** (μ : kˣ →* ℂˣ) (hμ : IsTame μ) : (P.step t μ hμ).J = P.stepJ t

theorem **step_lam** (μ : kˣ →* ℂˣ) (hμ : IsTame μ) (h : P.stepJ t) (g : P.J) (hg : (g : GL a k).val = (h : GL (a ⊕ t) k).val.toBlocks₁₁) : (P.step t μ hμ).lam h = P.lam g * (μ (Matrix.GeneralLinearGroup.det (g : GL a k)))⁻¹ * μ (Matrix.GeneralLinearGroup.det (h : GL (a ⊕ t) k))

theorem **step_lam_eq** (μ : kˣ →* ℂˣ) (hμ : IsTame μ) (h : P.stepJ t) (g : P.J) (hg : (g : GL a k).val = (h : GL (a ⊕ t) k).val.toBlocks₁₁) (D : GL t k) (hD : D.val = (h : GL (a ⊕ t) k).val.toBlocks₂₂) : (P.step t μ hμ).lam h = P.lam g * μ (Matrix.GeneralLinearGroup.det D)

theorem **Phi_of_mem** (μ : kˣ →* ℂˣ) {x : Matrix (a ⊕ t) a k} (g : P.J) (hg : (g : GL a k).val = x.toRows₁) (hx : MaxM x.toRows₂) : P.Phi t μ x = (P.lam g : ℂ) * ((μ (Matrix.GeneralLinearGroup.det (g : GL a k)))⁻¹ : ℂˣ)

theorem **Phi_ne_zero_iff** (μ : kˣ →* ℂˣ) {x : Matrix (a ⊕ t) a k} : P.Phi t μ x ≠ 0 ↔ (∃ g ∈ P.J, g.val = x.toRows₁) ∧ MaxM x.toRows₂

theorem **extZ_add** {μ : kˣ →* ℂˣ} (hμ : IsTame μ) (M : Matrix a a k) {Y : Matrix a a k} (hY : MaxM Y) : extZ P.J (P.rho μ) (M + Y) = extZ P.J (P.rho μ) M

theorem **Phi_add** {μ : kˣ →* ℂˣ} (hμ : IsTame μ) (x : Matrix (a ⊕ t) a k) {y : Matrix (a ⊕ t) a k} (hy : MaxM y) : P.Phi t μ (x + y) = P.Phi t μ x

theorem **intM_of_Phi_ne_zero** (μ : kˣ →* ℂˣ) {x : Matrix (a ⊕ t) a k} (h : P.Phi t μ x ≠ 0) : IntM x

theorem **mem_SB_of_periodic** {f : Matrix b a k → ℂ} (hs : ∀ x, f x ≠ 0 → IntM x) (hp : ∀ x y, MaxM y → f (x + y) = f x) : f ∈ SB (Matrix b a k)

theorem **TypeDatum.Phi_mem_SB** (P : TypeDatum k a) {μ : kˣ →* ℂˣ} (hμ : IsTame μ) : P.Phi t μ ∈ SB (Matrix (a ⊕ t) a k)

theorem **TypeDatum.iter_zero** (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) : P.iter t μ hμ 0 = P.reindex (ixZero r t)

theorem **TypeDatum.iter_succ** (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) : P.iter t μ hμ (j + 1) = ((P.iter t μ hμ j).step t (μ (j + 1)) (hμ _)).reindex (ixSucc r t j)

theorem **TypeDatum.iterPhi_mem_SB** (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) : P.iterPhi t μ hμ j ∈ SB (Matrix (Ix r t (j + 1)) (Ix r t j) k)

theorem **mem_parahoric** {δ : kˣ →* ℂˣ} {hδ : IsTame δ} {h : GL (r ⊕ s) k} : h ∈ (parahoric r s δ hδ).J ↔ h ∈ KO k (r ⊕ s) ∧ MaxM h.val.toBlocks₂₁

theorem **parahoric_lam** {δ : kˣ →* ℂˣ} {hδ : IsTame δ} (h : (parahoric r s δ hδ).J) (D : GL s k) (hD : D.val = (h : GL (r ⊕ s) k).val.toBlocks₂₂) : (parahoric r s δ hδ).lam h = δ (Matrix.GeneralLinearGroup.det D)

theorem **parPhi_of_mem** (δ : kˣ →* ℂˣ) {x : Matrix (r ⊕ s) (r ⊕ s') k} {g : GL r k} (hg : g ∈ KO k r) (e : g.val = x.toBlocks₁₁) (h21 : MaxM x.toBlocks₂₁) (h12 : IntM x.toBlocks₁₂) (h22 : IntM x.toBlocks₂₂) : parPhi δ x = ((δ (Matrix.GeneralLinearGroup.det g))⁻¹ : ℂˣ)

theorem **parPhi_fromBlocks** (δ : kˣ →* ℂˣ) {A : Matrix r r k} {B : Matrix r s' k} {C : Matrix s r k} {D : Matrix s s' k} {g : GL r k} (hg : g ∈ KO k r) (e : g.val = A) (h21 : MaxM C) (h12 : IntM B) (h22 : IntM D) : parPhi δ (fromBlocks A B C D) = ((δ (Matrix.GeneralLinearGroup.det g))⁻¹ : ℂˣ)

theorem **parPhi_eq_Phi** [IsEmpty s'] (δ : kˣ →* ℂˣ) (x : Matrix (r ⊕ s) (r ⊕ s') k) : parPhi δ x = (TypeDatum.full k r).Phi s δ x.toCols₁

theorem **parPhi_ne_zero_iff** (δ : kˣ →* ℂˣ) {x : Matrix (r ⊕ s) (r ⊕ s') k} : parPhi δ x ≠ 0 ↔ (∃ g ∈ KO k r, g.val = x.toBlocks₁₁) ∧ MaxM x.toBlocks₂₁ ∧ IntM x.toBlocks₁₂ ∧ IntM x.toBlocks₂₂

theorem **parPhi_mem_SB** [Fintype s'] {δ : kˣ →* ℂˣ} (hδ : IsTame δ) : parPhi δ ∈ SB (Matrix (r ⊕ s) (r ⊕ s') k)

theorem **conj_apply** (F : LatFrame k ι Y) (g : GL ι k) : F.conj g = F.bas * reGL F.ord g * F.bas⁻¹

theorem **det_conj** (F : LatFrame k ι Y) (g : GL ι k) : Matrix.GeneralLinearGroup.det (F.conj g) = Matrix.GeneralLinearGroup.det g

theorem **coords_mul** (F : LatFrame k ι Y) (F' : LatFrame k ι' Y') (h : GL ι k) (g : GL ι' k) (x : Matrix Y Y' k) : F.coords F' ((F.conj h).val * x * (F'.conj g).val) = h.val * F.coords F' x * g.val

theorem **om_transport** (F : LatFrame k ι Y) (F' : LatFrame k ι' Y') (μ : kˣ →* ℂˣ) (g : GL ι' k) (h : GL ι k) (Φ : Matrix ι ι' k → ℂ) : om μ (F'.conj g) (F.conj h) (F.transport F' Φ) = F.transport F' (om μ g h Φ)

theorem **transport_mem_SB** (F : LatFrame k ι Y) (F' : LatFrame k ι' Y') {Φ : Matrix ι ι' k → ℂ} (hΦ : Φ ∈ SB (Matrix ι ι' k)) : F.transport F' Φ ∈ SB (Matrix Y Y' k)

theorem **mem_latOf** {g : GL Y k} {y : Y → k} : y ∈ latOf g ↔ ∀ i, ((g⁻¹).val *ᵥ y) i ∈ 𝒪[k]

theorem **mem_latOf_one** {y : Y → k} : y ∈ latOf (1 : GL Y k) ↔ ∀ i, y i ∈ 𝒪[k]

theorem **mulVec_mem_latOf** (h : GL Y k) {g : GL Y k} {y : Y → k} : h.val *ᵥ y ∈ latOf (h * g) ↔ y ∈ latOf g

theorem **mem_box** {n : Type*} {c : kˣ} {y : n → k} : y ∈ box n c ↔ ∀ i, valuation k (y i) ≤ valuation k (c : k)

theorem **box_one** : box Y (1 : kˣ) = latOf 1

theorem **mem_stabLat** {g h : GL Y k} : h ∈ stabLat g ↔ g⁻¹ * h * g ∈ KO k Y

theorem **latOf_le_iff** {g g' : GL Y k} : latOf g ≤ latOf g' ↔ IntM (g'⁻¹ * g).val

theorem **mem_stabLat_iff** {g h : GL Y k} : h ∈ stabLat g ↔ latOf (h * g) = latOf g

theorem **isOpen_isCompact_conj** (g : GL Y k) {J : Subgroup (GL Y k)} (ho : IsOpen (J : Set (GL Y k))) (hc : IsCompact (J : Set (GL Y k))) : IsOpen (J.comap (MulAut.conj g⁻¹).toMonoidHom : Set (GL Y k)) ∧ IsCompact (J.comap (MulAut.conj g⁻¹).toMonoidHom : Set (GL Y k))

theorem **TypeDatum.map_conj** (P : TypeDatum k ι) (F : LatFrame k ι Y) : IsOpen ((P.toLevel.map F.conj).J : Set (GL Y k)) ∧ IsCompact ((P.toLevel.map F.conj).J : Set (GL Y k)) ∧ (P.toLevel.map F.conj).J ≤ stabLat F.bas

theorem **normalBasis_val** (ϖ : kˣ) (e : Fin c → ℤ) (z : n → k) : (normalBasis ϖ e z).val = fromBlocks 1 (Matrix.of fun r i => if i.val = 0 then z r * (ϖ : k) ^ e i else 0) 0 (diagonal fun i => (ϖ : k) ^ e i)

theorem **mem_latOf_typeFrame** [ValuativeRel k] (ϖ : kˣ) (ord : ι ≃ n ⊕ Fin c) (y : n ⊕ Fin c → k) : y ∈ latOf (typeFrame ϖ ord).bas ↔ (∀ i, y (.inl i) ∈ 𝒪[k]) ∧ ∀ i : Fin c, valuation k (y (.inr i)) ≤ valuation k ((ϖ : k) ^ (i.val + 1))

theorem **stdOrd_eps** (κ t : Type*) {l : ℕ} (j c : ℕ) (h : j + c = l) (i : κ) : stdOrd κ t j c h (.inl (.inl i)) = .inl (.inl (.inl i))

theorem **standardOrderExtendedBasis_apply_low_block** (κ t : Type*) {l : ℕ} (j c : ℕ) (h : j + c = l) (i : Fin c) : stdOrd κ t j c h (.inl (.inr ⟨i.val, by omega⟩)) = .inr i

theorem **stdOrd_f** (κ t : Type*) {l : ℕ} (j c : ℕ) (h : j + c = l) (i : Fin j) : stdOrd κ t j c h (.inl (.inr ⟨c + i.val, by omega⟩)) = .inl (.inl (.inr i))

theorem **stdOrd_d** (κ t : Type*) {l : ℕ} (j c : ℕ) (h : j + c = l) (p : Fin j × t) : stdOrd κ t j c h (.inr p) = .inl (.inr p)

theorem **mem_Cgrp** {J : Subgroup (GL (n ⊕ Fin c) k)} {h x : GL (n ⊕ Fin c) k} : x ∈ Cgrp J h ↔ x ∈ R k n c 1 ∧ h⁻¹ * x * h ∈ J

theorem **mem_prX** {C : Subgroup (GL (n ⊕ Fin c) k)} {g : GL n k} : g ∈ prX C ↔ ∃ u ∈ N k n c 1, u * embX c g ∈ C

theorem **IsColChar.unique** {ψ : AddChar k Circle} {P : Level (GL (n ⊕ Fin c) k)} {h : GL (n ⊕ Fin c) k} {χ χ' : prX (Cgrp P.J h) →* ℂˣ} (H : IsColChar ψ P h χ) (H' : IsColChar ψ P h χ') : χ = χ'

theorem **xi_eq_zero** {J : Subgroup (GL (n ⊕ Fin c) k)} (b : GL (n ⊕ Fin c) k) {h' : GL (n' ⊕ Fin (c + 1)) k} {h : GL (n ⊕ Fin c) k} (hh : DoubleCoset.mk (R k n c 1) J h ∉ idxAt ψ ν μ Φ J h') : xi ψ ν μ Φ J b h' h = 0

theorem **volN_stabLat** [ν.IsAddHaarMeasure] (b : GL (n ⊕ Fin c) k) : 0 < (volN ν n c 1 ((stabLat b).subgroupOf (N k n c 1))).toReal

theorem **volNat_stabLat** [ν.IsAddHaarMeasure] (b : GL (n ⊕ Fin c) k) : volNat ν b (stabLat b) = 1

theorem **volN_Cgrp** [ν.IsAddHaarMeasure] {J : Subgroup (GL (n ⊕ Fin c) k)} (ho : IsOpen (J : Set (GL (n ⊕ Fin c) k))) (hc : IsCompact (J : Set (GL (n ⊕ Fin c) k))) (h : GL (n ⊕ Fin c) k) : volN ν n c 1 ((Cgrp J h).subgroupOf (N k n c 1)) ≠ 0 ∧ volN ν n c 1 ((Cgrp J h).subgroupOf (N k n c 1)) ≠ ⊤

theorem **volNat_Cgrp_pos** [ν.IsAddHaarMeasure] {J : Subgroup (GL (n ⊕ Fin c) k)} (ho : IsOpen (J : Set (GL (n ⊕ Fin c) k))) (hc : IsCompact (J : Set (GL (n ⊕ Fin c) k))) (b h : GL (n ⊕ Fin c) k) : 0 < volNat ν b (Cgrp J h)

theorem **xi_eq_zero_iff** [ν.IsAddHaarMeasure] {J : Subgroup (GL (n ⊕ Fin c) k)} (ho : IsOpen (J : Set (GL (n ⊕ Fin c) k))) (hc : IsCompact (J : Set (GL (n ⊕ Fin c) k))) (b : GL (n ⊕ Fin c) k) (h' : GL (n' ⊕ Fin (c + 1)) k) (h : GL (n ⊕ Fin c) k) : xi ψ ν μ Φ J b h' h = 0 ↔ flat ψ ν μ Φ h' h = 0

theorem **withCol_blocks** (x : Matrix (n ⊕ Fin 0) (n' ⊕ Fin (0 + 1)) k) : withCol x.toBlocks₁₁ (fun r => x (.inl r) (.inr 0)) = x

theorem **integrable_hat** [ν.IsAddHaarMeasure] (hψ : Continuous ψ) {f : (n → k) → ℂ} (hf : f ∈ SB (n → k)) (η : n → k) : Integrable (fun y => f y * (ψ (ε * (η ⬝ᵥ y)) : ℂ)) (Measure.pi fun _ => ν)

theorem **cosetInd_mem_SB** (c : kˣ) (yb : (n → k) ⧸ box n c) : cosetInd (box n c) yb ∈ SB (n → k)

theorem **xiPen_mk** (H : T.Adapted A S M) {g : GL (X s ⊕ Fin (0 + 1)) k} (hg : g ∈ A s 0 rfl) (y : X (s + 1) → k) (x₀ : Matrix (X (s + 1)) (X s) k) : T.xiPen M g (Submodule.Quotient.mk y) x₀ = T.Psi g x₀ y

theorem **isRepSystem_someReps** : T.IsRepSystem (T.someReps ψ ν) ψ ν

theorem **Chain.ofType_regular** (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (F : ∀ j c, j + c = s + 1 → LatFrame k (Ix r t j) (X j ⊕ Fin c)) (Ftop : LatFrame k (Ix r t (s + 2)) Z) : (Chain.ofType P μ hμ F Ftop).Regular

theorem **Chain.ofTwist_regular** (δ : kˣ →* ℂˣ) (hδ : IsTame δ) (ν : ℕ → kˣ →* ℂˣ) (F : ∀ j c, j + c = s + 1 → LatFrame k (Ix r t j) (X j ⊕ Fin c)) (Ftop : LatFrame k (Ix r t (s + 2)) Z) : (Chain.ofTwist δ hδ ν F Ftop).Regular

theorem **extZ_mul_right** (J : Subgroup (GL a k)) (ρ : J →* ℂˣ) (g g' : J) : extZ J ρ ((g : GL a k).val * (g' : GL a k).val) = extZ J ρ (g : GL a k).val * ρ g'

theorem **eq_mul_of_support** {G S : Type*} [Group G] (act : G → S → S) (hact : ∀ g x, act g⁻¹ (act g x) = x) {f : S → ℂ} {c : G → ℂˣ} (h : ∀ g x, f x ≠ 0 → f (act g x) = c g * f x) (g : G) (x : S) : f (act g x) = c g * f x

theorem **om_of_mem_KO** (μ : kˣ →* ℂˣ) {g : GL a k} (hg : g ∈ KO k a) {h : GL b k} (hh : h ∈ KO k b) (Φ : Matrix b a k → ℂ) (x : Matrix b a k) : om μ g h Φ x = (μ (Matrix.GeneralLinearGroup.det g) : ℂ) * ((μ (Matrix.GeneralLinearGroup.det h))⁻¹ : ℂˣ) * Φ ((h⁻¹).val * x * g.val)

theorem **Phi_mul_right** (g : P.J) (x : Matrix (a ⊕ t) a k) : P.Phi t μ (x * (g : GL a k).val) = P.rho μ g * P.Phi t μ x

theorem **Phi_mul_left** (hμ : IsTame μ) (h : P.stepJ t) (x : Matrix (a ⊕ t) a k) : P.Phi t μ ((h : GL (a ⊕ t) k).val * x) = P.rho μ (P.corner h) * P.Phi t μ x

theorem **hasTypes_Phi** (hμ : IsTame μ) : HasTypes μ P.toLevel (P.step t μ hμ).toLevel (P.Phi t μ)

theorem **parPhi_mul_right** (hδ : IsTame δ) (g : (TypeDatum.full k r).stepJ s') (x : Matrix (r ⊕ s) (r ⊕ s') k) : parPhi δ (x * (g : GL (r ⊕ s') k).val) = (TypeDatum.full k r).rho δ ((TypeDatum.full k r).corner g) * parPhi δ x

theorem **parPhi_mul_left** (hδ : IsTame δ) (g : (TypeDatum.full k r).stepJ s) (x : Matrix (r ⊕ s) (r ⊕ s') k) : parPhi δ ((g : GL (r ⊕ s) k).val * x) = (TypeDatum.full k r).rho δ ((TypeDatum.full k r).corner g) * parPhi δ x

theorem **hasTypes_parPhi** (hδ : IsTame δ) {ν : kˣ →* ℂˣ} (hν : IsUnram ν) : HasTypes (ν * δ) (parahoric r s' δ hδ).toLevel (parahoric r s δ hδ).toLevel (parPhi δ)

theorem **HasTypes.transport** {Y Y' : Type*} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {μ : kˣ →* ℂˣ} {P' : Level (GL a k)} {P : Level (GL b k)} {Φ : Matrix b a k → ℂ} (H : HasTypes μ P' P Φ) (F : LatFrame k b Y) (F' : LatFrame k a Y') : HasTypes μ (P'.map F'.conj) (P.map F.conj) (F.transport F' Φ)

theorem **TypeDatum.hasTypes_iterPhi** {r : Type*} [Fintype r] [DecidableEq r] (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) : HasTypes (μ (j + 1)) (P.iter t μ hμ j).toLevel (P.iter t μ hμ (j + 1)).toLevel (P.iterPhi t μ hμ j)

theorem **Chain.ofType_laws** (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (F : ∀ j c, j + c = s + 1 → LatFrame k (Ix r t j) (X j ⊕ Fin c)) (Ftop : LatFrame k (Ix r t (s + 2)) Z) : (Chain.ofType P μ hμ F Ftop).Laws

theorem **Chain.ofTwist_laws** (δ : kˣ →* ℂˣ) (hδ : IsTame δ) (ν : ℕ → kˣ →* ℂˣ) (hν : ∀ j, IsUnram (ν j)) (F : ∀ j c, j + c = s + 1 → LatFrame k (Ix r t j) (X j ⊕ Fin c)) (Ftop : LatFrame k (Ix r t (s + 2)) Z) : (Chain.ofTwist δ hδ ν F Ftop).Laws

theorem **exists_affPt** {p : ℕ} {x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k} (h : ∀ r s, ¬ Free p (r, s) → x r s = basePt c x.toBlocks₁₁ r s) : ∃ e, affPt p x.toBlocks₁₁ e = x

theorem **opE_ne_zero** [MeasurableSpace k] [BorelSpace k] {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (p : ℕ) {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (hΦ : Φ ∈ SB _) (h01 : ∀ x, Φ x ≠ 0 → Φ x = 1 ∧ IntM x)

{x₀ : Matrix n n' k} {e₀} (he : Φ (affPt p x₀ e₀) ≠ 0) : opE ψ ν p Φ x₀ ≠ 0

theorem **twistChain_entry_ne_zero** [MeasurableSpace k] [BorelSpace k] {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (κ t : Type*) [Fintype κ] [DecidableEq κ] [Fintype t] [DecidableEq t] (s c : ℕ) (h : 0 + 1 + c = s) : DoubleCoset.mk _ _ 1 ∈ (twistChain s t κ (1 : kˣ →* ℂˣ) (fun _ _ => rfl) fun _ => 1).idx ψ ν 0 c h 1 ∧ (twistChain s t κ (1 : kˣ →* ℂˣ) (fun _ _ => rfl) fun _ => 1).entry ψ ν 0 c h 1 1 ≠ 0

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.SymplecticAdjointInclusionZero

theorem **sympl_left** (S T : HSp D) (g : GA D T) (x y : WO S T) : symplOH S T (mt T g · x) (mt T g · y) = symplOH S T x y

theorem **splitT** (h3 : d.pY · mt T n⁻¹ = d.pY) : mt T n⁻¹ · d.incAdj S y = d.incX S (xiT d n y) + d.inc0 S (ptT d S y n) + d.incAdj S y

theorem **symplectic_adjoint_inclusion_zero** (v : WO S T₀) : symplOH S T (d.incAdj S y) (d.inc0 S v) = 0

theorem **cenT_eq** (h3 : d.pY · mt T n⁻¹ = d.pY) : cenT d S y n = −symplOH S T (d.incX S (xiT d n y)) (d.incAdj S y)

theorem **moveT** (h2 : mt T n⁻¹ · d.j0 = d.j0 + d.jX · β) (w₀ : WO S T₀) : mt T n⁻¹ · d.inc0 S w₀ = d.inc0 S w₀ + d.incX S (β · w₀)

theorem **crossT** (h2 : mt T n⁻¹ · d.j0 = d.j0 + d.jX · β) (h3 : d.pY · mt T n⁻¹ = d.pY) (w₀ : WO S T₀) : symplOH S T₀ w₀ (ptT d S y n) = −symplOH S T (d.incX S (β · w₀)) (d.incAdj S y)

theorem **mixT_comp** {μ : Measure (Mx D ρ S.rank)} {ℱ : Set (Mx D ρ S.rank)} (hμ : IsCovolOne (ratMx D ρ S.rank) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (h1 : mt T n⁻¹ · d.jX = d.jX) (h2 : mt T n⁻¹ · d.j0 = d.j0 + d.jX · β) (h3 : d.pY · mt T n⁻¹ = d.pY) : d.mixT S ψ μ ℱ (fun w => f (mt T n⁻¹ · w)) y = jacT ψ d S y n (d.mixT S ψ μ ℱ f y)

theorem **sympl_right** (S T : HSp D) (g : GA D S) (x y : WO S T) : symplOH S T (x · mt S g) (y · mt S g) = symplOH S T x y

theorem **splitS** (h1 : mt S n · d.jX = d.jX) : d.extX T x · mt S n = d.extY T (etaS d n x) + d.extX T x + d.ext0 T (ptS d T x n)

theorem **cenS_eq** (h1 : mt S n · d.jX = d.jX) : cenS d T x n = −symplOH S T (d.extY T (etaS d n x)) (d.extX T x)

theorem **moveS** (h2 : d.p0 · mt S n = d.p0 + β · d.pY) (w₀ : WO S₀ T) : d.ext0 T w₀ · mt S n = d.ext0 T w₀ + d.extY T (w₀ · β)

theorem **crossS** (h1 : mt S n · d.jX = d.jX) (h2 : d.p0 · mt S n = d.p0 + β · d.pY) (w₀ : WO S₀ T) : symplOH S₀ T w₀ (ptS d T x n) = −symplOH S T (d.extY T (w₀ · β)) (d.extX T x)

theorem **mixS_comp** {μ : Measure (Mx D T.rank ρ)} {ℱ : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (h1 : mt S n · d.jX = d.jX) (h2 : d.p0 · mt S n = d.p0 + β · d.pY) (h3 : d.pY · mt S n = d.pY) : d.mixS T ψ μ ℱ (fun w => f (w · mt S n)) x = jacS ψ d T x n (d.mixS T ψ μ ℱ f x)

theorem **om_rat** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {g : GA D S} (hg : g ∈ GQ S) {h : GA D T} (hh : h ∈ GQ T) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hpres : ∀ w, IsThetaFunO ψ S T (heisO ψ S T w f)) (w : WO S T) : om (g, h) f (actO S T g h w) = f w

theorem **psiE_half** (ψ : AddChar (AF D.host.F) ℂ) (t : AF D.host.F) : psiE ψ (t • half D) = ψ t

theorem **psiBes_three** (ψ : AddChar (AF D.host.F) ℂ) (t : AF D.host.F) : ∃ n ∈ flagN (hypDecThree D), psiBes ψ (hypDecThree D) 1 (fun _ => 1) n = ψ t

theorem **unip_two** {R : Type} [CommRing R] (s : R) : ∃ u : GL (Fin 2) R, u ∈ upperUni 2 R ∧ u.1 0 1 = s

theorem **psiUni_two** (ψ : AddChar (AF D.host.F) ℂ) (t : AF D.host.F) : ∃ u ∈ UX D 2, psiUni ψ u = ψ t

theorem **UX_one** (u : UX D 1) : u = 1

theorem **isUnitU_one** : IsUnitU D 1 (Measure.dirac 1)

theorem **flat_one** (ψ : AddChar (AF D.host.F) ℂ) {V V₀ W W₀ A L : HSp D} (dV : HypDec V V₀ 1) (o : OrthDec V₀ A L) (ζ) (dW : HypDec W W₀ 1) (μS ℱS μT ℱT) (Ψ : WO V W → ℂ) : flat ψ dV o ζ dW μS ℱS μT ℱT (Measure.dirac 1) Ψ = evalZero ψ dV o ζ dW μS ℱS μT ℱT Ψ

theorem **fjKer_one** (ψ : AddChar (AF D.host.F) ℂ) {W W₀ V₀ A L : HSp D} {ρ : ℕ} (dW : HypDec W W₀ ρ) (o : OrthDec V₀ A L) (ζ) (om : Om V₀ W₀) (Ξ) (h₀ : GA D A) (h : GA D W₀) : fjKer ψ dW o ζ om Ξ h₀ 1 h = thetaKO om (o.emb (h₀, 1)) h Ξ

theorem **mixT_nB_nC** {ψ : AddChar (AF D.host.F) ℂ} {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {μ : Measure (Mx D ρ S.rank)} {ℱ : Set (Mx D ρ S.rank)} (hμ : IsCovolOne (ratMx D ρ S.rank) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (b : Mx D ρ T₀.rank) (c : skewA D ρ) (y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (fun w => f (mt T (d.nB b * d.nC c)⁻¹ · w)) y = jacT ψ d S y (d.nB b * d.nC c) (d.mixT S ψ μ ℱ f y)

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.TupPrimeEmpty

theorem **OneMapOverL.pins** {ph : ℂ} {i : Fin D.R.m} (h : OneMapOverL T R κ v₀ ph i) : PinsOverL T R κ v₀ i

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.TwistPlacesOffPrimeQZero

theorem **AllIn.numbersIn** {μ₀ : T} (h : AllIn σ S μ₀) : NumbersIn σ S μ₀

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.TypeChainPrincipalEntryNeZero

theorem **isColChar_full** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) : IsColChar ψ (TypeDatum.full k (n ⊕ Fin c)).toLevel 1 1

theorem **hat_coset** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {ε : k} (hε : ε ∈ 𝒪[k]) (y₀ : n → k) {η : n → k} (hη : ∀ i, η i ∈ 𝒪[k]) : hat ψ ν ε (cosetInd (box n 1) (Submodule.Quotient.mk y₀)) η = ψ (ε * (η ·ᵥ y₀))

theorem **opE_ne_zero'** (ψ : AddChar k Circle) (ν : Measure k) [ν.IsAddHaarMeasure] (p : ℕ) {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (hΦ : Φ ∈ SB _) (h01 : ∀ x, Φ x ≠ 0 → Φ x = ψ (ch x)) {x₀ : Matrix n n' k} {e₀} (he : Φ (affPt p x₀ e₀) ≠ 0) : opE ψ ν p Φ x₀ ≠ 0

theorem **wt_ne_zero** (ϖ : kˣ) (i : n ⊕ Fin c) : wt ϖ i ≠ 0

theorem **typeFrame_val** (ϖ : kˣ) (o : ι ≃ n ⊕ Fin c) : (typeFrame ϖ o).bas.val = diagonal (wt ϖ)

theorem **typeFrame_inv** (ϖ : kˣ) (o : ι ≃ n ⊕ Fin c) : ((typeFrame ϖ o).bas⁻¹).val = diagonal fun i => (wt ϖ i)⁻¹

theorem **coords_typeFrame** (ϖ : kˣ) (o : ι ≃ n ⊕ Fin c) (o' : ι' ≃ n' ⊕ Fin c') (x : Matrix (n ⊕ Fin c) (n' ⊕ Fin c') k) (i : ι) (j : ι') : (typeFrame ϖ o).coords (typeFrame ϖ o') x i j = (wt ϖ (o i))⁻¹ * x (o i) (o' j) * wt ϖ (o' j)

theorem **typeChain_entry_ne_zero** (ψ : AddChar k Circle) (ν : Measure k) [ν.IsAddHaarMeasure] (ϖ : kˣ) (s c : ℕ) (h : 0 + 1 + c = s) (P : TypeDatum k (κ ⊕ Fin (s + 1))) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (hP : ∀ g : P.J, (P.rho (μ 1) g : ℂ) = ψ (∑ i : Fin s, (g : GL (κ ⊕ Fin (s + 1)) k).val (.inr i.castSucc) (.inr i.succ) / ϖ)) : DoubleCoset.mk _ _ 1 ∈ (typeChain t ϖ P μ hμ).idx ψ ν 0 c h 1 ∧ (typeChain t ϖ P μ hμ).entry ψ ν 0 c h 1 1 ≠ 0

theorem **typeChain_principal_entry_ne_zero** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (ν : Measure k) [ν.IsAddHaarMeasure] {ϖ : kˣ} (hϖ : ∀ z : k, valuation k z < 1 → valuation k z ≤ valuation k (ϖ : k)) (s c : ℕ) (h : 0 + 1 + c = s) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j,

IsTame (μ j)) : (typeChain t ϖ (TypeDatum.principal k (κ ⊕ Fin (s + 1))) μ hμ).entry ψ ν 0 c h 1 1 ≠ 0

theorem **typeChain_adapted** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) < 1) {s : ℕ} (P : TypeDatum k (κ ⊕ Fin (s + 1))) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (A : RepFam k s (XIx κ t)) (hA : (typeChain t ϖ P μ hμ).RepsInStab A) : (typeChain t ϖ P μ hμ).Adapted A (box _ ϖ⁻¹) (box _ 1)

theorem **coords_surjective** {ι ι' Y Y' : Type*} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (F : LatFrame k ι Y) (F' : LatFrame k ι' Y') : Function.Surjective (F.coords F')

theorem **typeChain_coupIdx** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) < 1) {s : ℕ} (P : TypeDatum k (κ ⊕ Fin (s + 1))) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (A : RepFam k s (XIx κ t)) (hA : (typeChain t ϖ P μ hμ).RepsInStab A) (h1 : 1 ∈ A s 0 rfl) : ∃ yb ∈ (typeChain t ϖ P μ hμ).coupIdx A (box _ ϖ⁻¹) (box _ 1), (typeChain t ϖ P μ hμ).xiPen (box _ 1) 1 yb ≠ 0

theorem **twistChain_reps_zero** [MeasurableSpace k] (ψ : AddChar k Circle) (ν : Measure k) (ϖ : kˣ) (δ : kˣ →* ℂˣ) (hδ : IsTame δ) (ν' : ℕ → kˣ →* ℂˣ) : let T

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.UPartBot

theorem **glPart_le** (W : Flag n a) : glPart ε W ≤ leviOne ε W

theorem **uPart_bot** : uPart ε (⊥ : Submodule ℂ (Fin n → ℂ)) = ⊤

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ValuationDetOfMemKO

theorem **MaxM.intM** {M : Matrix m n k} (h : MaxM M) : IntM M

theorem **intM_zero** : IntM (0 : Matrix m n k)

theorem **maxM_zero** : MaxM (0 : Matrix m n k)

theorem **intM_one** [DecidableEq n] : IntM (1 : Matrix n n k)

theorem **IntM.add** {M M' : Matrix m n k} (h : IntM M) (h' : IntM M') : IntM (M + M')

theorem **IntM.neg** {M : Matrix m n k} (h : IntM M) : IntM (-M)

theorem **MaxM.add** {M M' : Matrix m n k} (h : MaxM M) (h' : MaxM M') : MaxM (M + M')

theorem **MaxM.neg** {M : Matrix m n k} (h : MaxM M) : MaxM (-M)

theorem **IntM.mul** [Fintype m] {M : Matrix l m k} {M' : Matrix m n k} (h : IntM M) (h' : IntM M') : IntM (M * M')

theorem **IntM.mul_maxM** [Fintype m] {M : Matrix l m k} {M' : Matrix m n k} (h : IntM M) (h' : MaxM M') : MaxM (M * M')

theorem **MaxM.mul_intM** [Fintype m] {M : Matrix l m k} {M' : Matrix m n k} (h : MaxM M) (h' : IntM M') : MaxM (M * M')

theorem **mem_KO** {g : GL n k} : g ∈ KO k n ↔ IntM g.val ∧ IntM (g⁻¹).val

theorem **intM_of_mem_KO** {g : GL n k} (h : g ∈ KO k n) : IntM g.val

theorem **KOne_le_KO** : KOne k n ≤ KO k n

theorem **residue_eq_zero_iff_lt** (x : 𝒪[k]) : IsLocalRing.residue 𝒪[k] x = 0 ↔ valuation k (x : k) < 1

theorem **residue_eq_iff_lt** (x y : 𝒪[k]) : IsLocalRing.residue 𝒪[k] x = IsLocalRing.residue 𝒪[k] y ↔ valuation k ((x : k) - y) < 1

theorem **exists_mem_KOne** {M : Matrix n n k} (h : MaxM (M - 1)) : ∃ g ∈ KOne k n, g.val = M

theorem **mem_KOne** {g : GL n k} : g ∈ KOne k n ↔ MaxM (g.val - 1)

theorem **det_of_mem_KOne** {g : GL n k} (h : g ∈ KOne k n) : valuation k (((Matrix.GeneralLinearGroup.det g : kˣ) : k) - 1) < 1

theorem **tame_det_of_mem_KOne** {μ : kˣ →* ℂˣ} (hμ : IsTame μ) {g : GL n k} (h : g ∈ KOne k n) : μ (Matrix.GeneralLinearGroup.det g) = 1

theorem **valuation_det_of_mem_KO** {g : GL n k} (h : g ∈ KO k n) : valuation k ((Matrix.GeneralLinearGroup.det g : kˣ) : k) = 1

theorem **exists_mul_KOne** {g : GL n k} (hg : g ∈ KO k n) {M : Matrix n n k} (h : MaxM (M - g.val)) : ∃ u ∈ KOne k n, (g * u).val = M

theorem **inv_mul_mem_KOne** {g g' : GL n k} (hg : g ∈ KO k n) (h : MaxM (g'.val - g.val)) : g⁻¹ * g' ∈ KOne k n

theorem **intM_fromBlocks** {a' t' : Type*} {A : Matrix a a' k} {B : Matrix a t' k} {C : Matrix t a' k} {D : Matrix t t' k} : IntM (fromBlocks A B C D) ↔ IntM A ∧ IntM B ∧ IntM C ∧ IntM D

theorem **maxM_fromBlocks** {a' t' : Type*} {A : Matrix a a' k} {B : Matrix a t' k} {C : Matrix t a' k} {D : Matrix t t' k} : MaxM (fromBlocks A B C D) ↔ MaxM A ∧ MaxM B ∧ MaxM C ∧ MaxM D

theorem **blockGL_mem_KO** {g : GL a k} {B : Matrix a t k} {D : GL t k} (hg : g ∈ KO k a) (hB : IntM B) (hD : D ∈ KO k t) : blockGL g B D ∈ KO k (a ⊕ t)

theorem **det_blockGL** (g : GL a k) (B : Matrix a t k) (D : GL t k) : Matrix.GeneralLinearGroup.det (blockGL g B D) = Matrix.GeneralLinearGroup.det g * Matrix.GeneralLinearGroup.det D

theorem **blocks_inv** {h : GL (a ⊕ t) k} (hh : h ∈ KO k (a ⊕ t)) (hC : MaxM h.val.toBlocks₂₁) : MaxM (h.val.toBlocks₁₁ * (h⁻¹).val.toBlocks₁₁ - 1) ∧ MaxM (h.val.toBlocks₂₂ * (h⁻¹).val.toBlocks₂₂ - 1) ∧ MaxM (h⁻¹).val.toBlocks₂₁

theorem **exists_mem_KO_of_mul** {A A' : Matrix n n k} (hA : IntM A) (hA' : IntM A') (h : MaxM (A * A' - 1)) : ∃ g ∈ KO k n, g.val = A

theorem **exists_diag_blocks** {h : GL (a ⊕ t) k} (hh : h ∈ KO k (a ⊕ t)) (hC : MaxM h.val.toBlocks₂₁) : (∃ g ∈ KO k a, g.val = h.val.toBlocks₁₁) ∧ ∃ D ∈ KO k t, D.val = h.val.toBlocks₂₂

theorem **tame_det_blocks** {μ : kˣ →* ℂˣ} (hμ : IsTame μ) {h : GL (a ⊕ t) k} (hh : h ∈ KO k (a ⊕ t)) (hC : MaxM h.val.toBlocks₂₁) {g : GL a k} (hg : g.val = h.val.toBlocks₁₁) {D : GL t k} (hD : D.val = h.val.toBlocks₂₂) : μ (Matrix.GeneralLinearGroup.det h) = μ (Matrix.GeneralLinearGroup.det g) * μ (Matrix.GeneralLinearGroup.det D)

theorem **isOpen_KOne** : IsOpen (KOne k n : Set (GL n k))

theorem **volN_ne_zero_ne_top** [MeasurableSpace k] [BorelSpace k] (ν : Measure k) [ν.IsAddHaarMeasure] {c : ℕ} (p : ℕ) {H : Subgroup (GL (n ⊕ Fin c) k)} (ho : IsOpen (H : Set (GL (n ⊕ Fin c) k))) (hc : IsCompact (H : Set (GL (n ⊕ Fin c) k))) : RootEx.volN ν n c p (H.subgroupOf (RootEx.N k n c p)) ≠ 0 ∧ RootEx.volN ν n c p (H.subgroupOf (RootEx.N k n c p)) ≠ ⊤

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.WittCoreInclusionMemCompletedHomLattice

theorem **mxAt_mul** {l m n : ℕ} (A : Mx D l m) {u : Mx D m n} (hu : MxAt P u) : MxAt P (A · u)

theorem **mxAt_mul'** {l m n : ℕ} {u : Mx D l m} (hu : MxAt P u) (B : Mx D m n) : MxAt P (u · B)

theorem **mxAt_iff** {m n : ℕ} {u : Mx D m n} : MxAt P u ↔ idem D P • u = u

theorem **mxAt_idem** {m n : ℕ} (u : Mx D m n) : MxAt P (idem D P • u)

theorem **mxAt_top** {m n : ℕ} {u : Mx D m n} (h : MxAt P u) : MxAt (fun _ => True) u

theorem **finLat_of** {m n : ℕ} {A : AddSubgroup (Mx D m n)} (fin : ∀ a ∈ A, MxAt (fun _ => True) a) (opn : ∃ U : Set (Mx D m n), IsOpen U ∧ (0 : Mx D m n) ∈ U ∧ ∀ a ∈ U, MxAt (fun _ => True) a → a ∈ A) {C : Set (Mx D m n)} (hC : IsCompact C) (h : ∀ a ∈ A, a ∈ C) : FinLat A

theorem **finLat_comap** {k l m n : ℕ} {Γ : AddSubgroup (Mx D m n)} (h : FinLat Γ) (f : Mx D k l →+ Mx D m n) {A : Mx D m k} {B : Mx D l n} {A' : Mx D k m} {B' : Mx D n l} (hf : ∀ a, f a = A · a · B) (hA : A' · A = 1) (hB : B · B' = 1) : FinLat (Γ.comap f)

theorem **sec_fin** {s : (Af D.host.F ⊗[D.host.F] D.host.E) →+ AE D} (s2 : ∀ a, finPart D (s a) = a) (s3 : ∀ a, infPart D (s a) = 0) {a : AE D} (ha : a ∈ idAt D P) (c : AE D) : s (finPart D (c * a)) = c * a

theorem **finLat_latO** {S T : HSp D} {LS : Lat S} {LT : Lat T} (hS : Submodule.IsLattice D.host.E LS) (hT : Submodule.IsLattice D.host.E LT) : FinLat (latO S T (LS : Set _) (LT : Set _) fun _ => True)

theorem **latMul_map** (M : Matrix (Fin T.rank) (Fin S.rank) D.host.E) {a a' : Set D.host.E} (h : ∀ a ∈ a, ∀ x ∈ LS, ∃ a' ∈ a', ∃ x' ∈ LT, M *ᵥ (a • x) = a' • x') {y : Fin S.rank → AE D} (hy : y ∈ latMul S a LS P) : rat M *ᵥ y ∈ latMul T a' LT P

theorem **latO_left** (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (hT' : IsSelfDualC T' LT') (M : Matrix (Fin T'.rank) (Fin T.rank) D.host.E) (hM : ∀ x ∈ LT, M *ᵥ x ∈ LT') {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) P) : rat M · u ∈ latO S T' (LS : Set _) (LT' : Set _) P

theorem **latO_right** (hS : IsSelfDualC S LS) (hS' : IsSelfDualC S' LS') (hT : IsSelfDualC T LT) (M : Matrix (Fin S.rank) (Fin S'.rank) D.host.E) (hM : ∀ x ∈ LS', M *ᵥ x ∈ LS) {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) P) : u · rat M ∈ latO S' T (LS' : Set _) (LT : Set _) P

theorem **latO_inK** (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {κ : GA D T} (hκ : InK T LT P κ) {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) P) : mt T κ · u ∈ latO S T (LS : Set _) (LT : Set _) P

theorem **rat_mem_latO** (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (M : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (hM : ∀ x ∈ LS, M *ᵥ x ∈ LT) {α : D.host.E} (hα : α ∈ D.host.AA) : idem D P • rat (α • M) ∈ latO S T (LS : Set _) (LT : Set _) P

theorem **idem_mem_latO** {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) fun _ => True) : idem D P • u ∈ latO S T (LS : Set _) (LT : Set _) P

theorem **mem_latO_at** {u : WO S T} (h : MxAt P u) : u ∈ latO S T (LS : Set _) (LT : Set _) (fun _ => True) ↔ u ∈ latO S T (LS : Set _) (LT : Set _) P

theorem **psi_eq_one** (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {x u : WO S T} (hx : idem D P • x ∈ latO S T (LS : Set _) (LT : Set _) P) (hu : u ∈ latO S T (LS : Set _) (LT : Set _) P) : ψ (symplO S T x u) = 1

theorem **psi_swap** (x u : WO S T) : ψ (symplO S T x u) = 1 ↔ ψ (symplO S T u x) = 1

theorem **exists_vac** (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) : ∃ f, VacG ψ S T (latO S T (LS : Set _) (LT : Set _) fun _ => True) f ∧ f ≠ 0

theorem **symplO_extX** (a : Mx D T.rank ρ) (u : WO S T) : symplO S T (d.extX T a) u = symplO S T (d.extX T a) (d.extY T (u · d.jY))

theorem **symplO_extY** (a : Mx D T.rank ρ) (u : WO S T) : symplO S T (d.extY T a) u = symplO S T (d.extY T a) (d.extX T (u · d.jX))

theorem **symplO_incX** (a : Mx D ρ T.rank) (u : WO T S) : symplO T S (d.incX T a) u = symplO T S (d.incX T a) (d.incY T (d.pY · u))

theorem **symplO_incY** (a : Mx D ρ T.rank) (u : WO T S) : symplO T S (d.incY T a) u = symplO T S (d.incY T a) (d.incX T (d.pX · u))

theorem **projX_mem** (x : Fin S.rank → D.host.E) (hx : x ∈ splitL d L₀ b) : (d.e * d.qX) *ᵥ x ∈ splitL d L₀ b

theorem **projY_mem** (x : Fin S.rank → D.host.E) (hx : x ∈ splitL d L₀ b) : (d.e' * d.qY) *ᵥ x ∈ splitL d L₀ b

theorem **srcX** {u : WO S T} (hu : u ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P) : d.extX T (u · d.jX) ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P

theorem **srcY** {u : WO S T} (hu : u ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P) : d.extY T (u · d.jY) ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P

theorem **tgtX** {u : WO T S} (hu : u ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P) : d.incX T (d.pX · u) ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P

theorem **tgtY** {u : WO T S} (hu : u ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P) : d.incY T (d.pY · u) ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P

theorem **dualX** (hψ : IsStdChar D ψ (-1)) {a : Mx D T.rank ρ} (ha : MxAt P a) : d.extX T a ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P ↔ ∀ a', d.extY T a' ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P → ψ (symplO S T (d.extX T a) (d.extY T a')) = 1

theorem **dualY** (hψ : IsStdChar D ψ (-1)) {a' : Mx D T.rank ρ} (ha : MxAt P a') : d.extY T a' ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P ↔ ∀ a, d.extX T a ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P → ψ (symplO S T (d.extX T a) (d.extY T a')) = 1

theorem **dualX'** (hψ : IsStdChar D ψ (-1)) {a : Mx D ρ T.rank} (ha : MxAt P a) : d.incX T a ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P ↔ ∀ a', d.incY T a' ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P → ψ (symplO T S (d.incX T a) (d.incY T a')) = 1

theorem **dualY'** (hψ : IsStdChar D ψ (-1)) {a' : Mx D ρ T.rank} (ha : MxAt P a') : d.incY T a' ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P ↔ ∀ a, d.incX T a ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P → ψ (symplO T S (d.incX T a) (d.incY T a')) = 1

theorem **src0** {u : WO S T} (hu : u ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P) : u · d.j0 ∈ latO S₀ T (L₀ : Set _) (LT : Set _) P

theorem **tgt0** {u : WO T S} (hu : u ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P) : d.p0 · u ∈ latO T S₀ (LT : Set _) (L₀ : Set _) P

theorem **wittExtendByZero_mem_completedHomLattice** {w : WO S₀ T} : d.ext0 T w ∈ latO S T (splitL d L₀ b : Set _) (LT : Set _) P ↔ w ∈ latO S₀ T (L₀ : Set _) (LT : Set _) P

theorem **wittCoreInclusion_mem_completedHomLattice** {w : WO T S₀} : d.inc0 T w ∈ latO T S (LT : Set _) (splitL d L₀ b : Set _) P ↔ w ∈ latO T S₀ (LT : Set _) (L₀ : Set _) P

theorem **stdAt_pcs** (v : GlobalTheory.FinPl.Pl D) : StdAt dV dW (pcs dV dW LV₀ LW₀ b c ψ) (splitL dV LV₀ b) (splitL dW LW₀ c) v where

theorem **pcs_A₂** : (pcs dV dW LV₀ LW₀ b c ψ).A₂ = (homL dV dW LV₀ LW₀ b c fun _ => True).comap (dV.extY W)

theorem **is_pcs** (uni : ∀ u : UX D ρ, MxAt (fun _ => True) (u.1.1 - 1) → dV.extX W (x · u.1.1 - x) ∈ homL dV dW LV₀ LW₀ b c (fun _ => True) → psiUni ψ u.1 = 1) : (pcs dV dW LV₀ LW₀ b c ψ).Is dV dW x ψ where

theorem **splits_pcs** : Splits dV dW (pcs dV dW LV₀ LW₀ b c ψ) x y ψ (homL dV dW LV₀ LW₀ b c fun _ => True)

theorem **vac_mem_cls** {f : WO V W → ℂ} (hf : VacG ψ V W (homL dV dW LV₀ LW₀ b c fun _ => True) f) : f ∈ cls dV dW (pcs dV dW LV₀ LW₀ b c ψ) x y ψ

theorem **exists_uni** {ρ : ℕ} {N : Mx D ρ ρ} (h2 : ∀ i j, j ≤ i → N i j = 0) : ∃ u : UX D ρ, u.1.1 = 1 + N

theorem **upSum_one_add** {ρ : ℕ} (N : Mx D ρ ρ) : upSum (1 + N) = upSum N

theorem **condK_one** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} {dV : HypDec V V₀ ρ} {dW : HypDec W W₀ σ} {p : Pieces V₀ W W₀ ρ σ} {x : Mx D W.rank ρ} {y : Mx D V₀.rank σ} {LV : Lat V} {LW : Lat W} {v : GlobalTheory.FinPl.Pl D} (hV : IsSelfDualC V LV) (hW : IsSelfDualC W LW) (hp : p.Is dV dW x ψ) (hs : StdAt dV dW p LV LW v) (hx : idem D (· = v) • dV.extX W x ∈ latO V W (LV : Set _) (LW : Set _) (· = v)) (hy : ∀ ξ, dV.ext0 W (dW.incX V₀ ξ) ∈ latO V W (LV : Set _) (LW : Set _) (· = v) → ψ (symplO V₀ W (dW.incAdj V₀ y) (dW.incX V₀ ξ)) = 1) : CondK dW p x y ψ LW (· = v) 1

theorem **trace_shiftM** (N : Mx D ρ ρ) : (adj (rat (shiftM D ρ)) · N).trace = upSum N

theorem **pairS_jX_jY** (N M : Mx D ρ ρ) : pairS W (dW.jX · N) (dW.jY · M) = (adj M · N).trace

theorem **hO_extX_extY** (a b : Mx D W.rank ρ) : hO V W (dV.extX W a) (dV.extY W b) = pairS W a b

theorem **symplO_shift** (N : Mx D ρ ρ) : symplO V W (dV.extX W (dW.jX · N)) (dV.extY W (dW.jY · (incA D (-D.host.xi0⁻¹) • rat (shiftM D ρ)))) = Algebra.trace (AF D.host.F) (AE D) (upSum N)

theorem **xZero_mem** (h1 : ∀ i, ∀ b ∈ 𝔟 i, α⁻¹ * b ∈ 𝔠 i) : idem D P • dV.extX W (xZero dW) ∈ homL dV dW LV₀ LW₀ 𝔟 𝔠 P

theorem **uni_xZero** (h2 : ∀ i, ∀ w ∈ dualC (𝔟 (i + 1)), α⁻¹ * -D.host.xi0⁻¹ * w ∈ dualC (𝔠 i)) (u : UX D ρ) (h : dV.extX W (xZero dW · u.1.1 - xZero dW) ∈ homL dV dW LV₀ LW₀ 𝔟 𝔠 fun _ => True) : psiUni ψ u.1 = 1

theorem **exists_ideals** (D : D7) : ∃ (α : D.host.E) (𝔟 𝔠 : ℕ → FractionalIdeal (𝒪 D.host.E)⁰ D.host.E), α ∈ D.host.AA ∧ α ≠ 0 ∧ (∀ i, 𝔟 i ≠ 0) ∧ (∀ i, 𝔠 i ≠ 0) ∧ (∀ i, ∀ b ∈ 𝔟 i, α⁻¹ * b ∈ 𝔠 i) ∧ ∀ i, ∀ w ∈ dualC (𝔟 (i + 1)), α⁻¹ * -D.host.xi0⁻¹ * w ∈ dualC (𝔠 i)

theorem **yZero_nsmul** (S : HSp D) (ρ N : ℕ) (z : Fin S.rank → D.host.E) : yZero S ρ (N • z) = N • yZero S ρ z

theorem **inhabitants** (hψ : IsStdChar D ψ (-1)) {V V₀ W W₀ : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) {z₀ : Fin V₀.rank → D.host.E} (hz : z₀ ≠ 0) : ∃ (p : Pieces V₀ W W₀ ρ ρ) (Γ : AddSubgroup (WO V W)) (LV : Lat V) (LV₀ : Lat V₀) (LW : Lat W) (LW₀ : Lat W₀) (N : ℕ), 0 < N ∧ FinLat Γ ∧ p.Is dV dW (xZero dW) ψ ∧ Splits dV dW p (xZero dW) (yZero V₀ ρ (N • z₀)) ψ Γ ∧ (∃ Φ ∈ cls dV dW p (xZero dW) (yZero V₀ ρ (N • z₀)) ψ, Φ ≠ 0) ∧ (∀ v, StdAt dV dW p LV LW v ∧ CondK dW p (xZero dW) (yZero V₀ ρ (N • z₀)) ψ LW (· = v) 1) ∧ ∀ᶠ v in Filter.cofinite, GoodAt dV dW (xZero dW) ψ LV LV₀ LW LW₀ v ∧ (0 < ρ → ∀ a ∈ idAt D (· = v), dW.incAdj V₀ (a • yZero V₀ ρ (N • z₀)) ∈ latO V₀ W (LV₀ : Set _) (LW : Set _) (· = v) ↔ a ∈ intAt D (· = v))

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.WittDecompProjAnisotropicAnisotropicEmbedding

theorem **mulVec_of_kill** {R : Type} [CommRing R] {k l n : ℕ} (J : Matrix (Fin n) (Fin k) R) (A : Matrix (Fin k) (Fin n) R) (Z : Matrix (Fin n) (Fin l) R) (P : Matrix (Fin l) (Fin n) R) (v : Fin n → R) (h : P *ᵥ v = 0) : (J * A + Z * P) *ᵥ v = J *ᵥ (A *ᵥ v)

theorem **mul_trace_smul** {R : Type} [CommRing R] {n : ℕ} {a c : R} (h : a * c = 1) (M : Matrix (Fin n) (Fin n) R) : a * (c • M).trace = M.trace

theorem **eq_of_add_self** {F R : Type} [Field F] [CharZero F] [CommRing R] [Algebra F R] {a b : R} (h : a + a = b + b) : a = b

theorem **mem_skewA** {ρ : ℕ} {c : Mx D ρ ρ} : c ∈ skewA D ρ ↔ adj c = -c

theorem **adj_plus** (b : Mx D ρ S₀.rank) : adj (plus S₀ b) = b · giA S₀

theorem **adj_plus_gA** (b : Mx D ρ S₀.rank) : adj (plus S₀ b) · gA S₀ = b

theorem **gA_plus** (b : Mx D ρ S₀.rank) : gA S₀ · plus S₀ b = adj b

theorem **plus_add** (b b' : Mx D ρ S₀.rank) : plus S₀ (b + b') = plus S₀ b + plus S₀ b'

theorem **plus_neg** (b : Mx D ρ S₀.rank) : plus S₀ (-b) = -plus S₀ b

theorem **plus_zero** : plus S₀ (0 : Mx D ρ S₀.rank) = 0

theorem **adj_mul_plus** (b b' : Mx D ρ S₀.rank) : adj (b · plus S₀ b') = b' · plus S₀ b

theorem **plus_mul_left** (a : Mx D ρ ρ) (b : Mx D ρ S₀.rank) : plus S₀ (a · b) = plus S₀ b · adj a

theorem **plus_mul_inv** (g : GA D S₀) (b : Mx D ρ S₀.rank) : plus S₀ (b · mt S₀ g⁻¹) = mt S₀ g · plus S₀ b

theorem **pX_eq** : d.pX = adj d.jY · gA S

theorem **pY_eq** : d.pY = adj d.jX · gA S

theorem **p0_eq** : d.p0 = giA S₀ · (adj d.j0 · gA S)

theorem **gram_XX** : adj d.jX · (gA S · d.jX) = 0

theorem **gram_YY** : adj d.jY · (gA S · d.jY) = 0

theorem **gram_YX** : adj d.jY · (gA S · d.jX) = 1

theorem **gram_XY** : adj d.jX · (gA S · d.jY) = 1

theorem **gram_00** : adj d.j0 · (gA S · d.j0) = gA S₀

theorem **gram_0X** : adj d.j0 · (gA S · d.jX) = 0

theorem **gram_0Y** : adj d.j0 · (gA S · d.jY) = 0

theorem **wittDecomp_gram_X_anisotropic** : adj d.jX · (gA S · d.j0) = 0

theorem **gramR_Y0_eq_TI** : adj d.jY · (gA S · d.j0) = 0

@[simp] theorem **pX_jX** : d.pX · d.jX = 1

@[simp] theorem **pX_j0** : d.pX · d.j0 = 0

@[simp] theorem **pX_jY** : d.pX · d.jY = 0

@[simp] theorem **pY_jX** : d.pY · d.jX = 0

@[simp] theorem **pY_j0** : d.pY · d.j0 = 0

@[simp] theorem **pY_jY** : d.pY · d.jY = 1

@[simp] theorem **p0_jX** : d.p0 · d.jX = 0

@[simp] theorem **p0_j0** : d.p0 · d.j0 = 1

@[simp] theorem **p0_jY** : d.p0 · d.jY = 0

theorem **sum_eq_one** : d.jX · d.pX + d.j0 · d.p0 + d.jY · d.pY = 1

theorem **sum_eq_one_rat** : d.e * d.qX + d.i₀ * d.q0 + d.e' * d.qY = 1

theorem **mat_ext** {k : ℕ} {x y : Mx D k S.rank} (hX : x · d.jX = y · d.jX) (h0 : x · d.j0 = y · d.j0) (hY : x · d.jY = y · d.jY) : x = y

theorem **ga_ext** {g h : GA D S} (hX : mt S g · d.jX = mt S h · d.jX) (h0 : mt S g · d.j0 = mt S h · d.j0) (hY : mt S g · d.jY = mt S h · d.jY) : g = h

theorem **images_self** : Images S S₀ d.jX d.j0 d.jY

theorem **gram_images** {E : Mx D S.rank ρ} {I : Mx D S.rank S₀.rank} {E' : Mx D S.rank ρ} (h : Images S S₀ E I E') : adj (E · d.pX + I · d.p0 + E' · d.pY) · (gA S · (E · d.pX + I · d.p0 + E' · d.pY)) = adj d.pY · d.pX + adj d.p0 · (gA S₀ · d.p0) + adj d.pX · d.pY

theorem **gA_eq** : gA S = adj d.pY · d.pX + adj d.p0 · (gA S₀ · d.p0) + adj d.pX · d.pY

theorem **giA_eq** : giA S = d.jX · adj d.jY + d.j0 · (giA S₀ · adj d.j0) + d.jY · adj d.jX

theorem **mt_ofImages** : mt S (d.ofImages E I E' h) = E · d.pX + I · d.p0 + E' · d.pY

theorem **ofImages_jX** : mt S (d.ofImages E I E' h) · d.jX = E

theorem **wittDecomp_ofImages_inclAnisotropic** : mt S (d.ofImages E I E' h) · d.j0 = I

theorem **ofImages_jY** : mt S (d.ofImages E I E' h) · d.jY = E'

theorem **ofImages_mem_GQ** (h1 : E ∈ ratMx D _ _) (h2 : I ∈ ratMx D _ _) (h3 : E' ∈ ratMx D _ _) : d.ofImages E I E' h ∈ GQ S

theorem **images_levi** (a : GL (Fin ρ) (AE D)) : Images S S₀ (d.jX · a.1) d.j0 (d.jY · adj (a⁻¹).1) where

theorem **mt_levi** : mt S (d.levi a) = d.jX · a.1 · d.pX + d.j0 · d.p0 + d.jY · adj (a⁻¹).1 · d.pY

theorem **levi_jX** : mt S (d.levi a) · d.jX = d.jX · a.1

theorem **wittDecomp_levi_inclAnisotropic** : mt S (d.levi a) · d.j0 = d.j0

theorem **levi_jY** : mt S (d.levi a) · d.jY = d.jY · adj (a⁻¹).1

theorem **pX_levi** : d.pX · mt S (d.levi a) = a.1 · d.pX

theorem **wittDecomp_projAnisotropic_levi** : d.p0 · mt S (d.levi a) = d.p0

theorem **pY_levi** : d.pY · mt S (d.levi a) = adj (a⁻¹).1 · d.pY

theorem **levi_mem_GQ** {a : GL (Fin ρ) (AE D)} (h : a.1 ∈ ratMx D ρ ρ) (h' : (a⁻¹).1 ∈ ratMx D ρ ρ) : d.levi a ∈ GQ S

theorem **continuous_levi** : Continuous d.levi

theorem **wittDecomp_frameImages_anisotropicEmbedding** (g : GA D S₀) : Images S S₀ d.jX (d.j0 · mt S₀ g) d.jY where

theorem **unitaryMatrix_wittCoreEmbedding** : mt S (d.emb0 g) = d.jX · d.pX + d.j0 · mt S₀ g · d.p0 + d.jY · d.pY

theorem **wittDecomp_anisotropicEmbedding_inclX** : mt S (d.emb0 g) · d.jX = d.jX

theorem **wittDecomp_anisotropicEmbedding_inclAnisotropic** : mt S (d.emb0 g) · d.j0 = d.j0 · mt S₀ g

theorem **wittDecomp_anisotropicEmbedding_inclY** : mt S (d.emb0 g) · d.jY = d.jY

theorem **wittDecomp_projX_anisotropicEmbedding** : d.pX · mt S (d.emb0 g) = d.pX

theorem **wittDecomp_projAnisotropic_anisotropicEmbedding** : d.p0 · mt S (d.emb0 g) = mt S₀ g · d.p0

theorem **wittDecomp_projY_anisotropicEmbedding** : d.pY · mt S (d.emb0 g) = d.pY

theorem **wittDecomp_anisotropicEmbedding_mem_rationalPoints** {g : GA D S₀} (h : g ∈ GQ S₀) : d.emb0 g ∈ GQ S

theorem **continuous_embed** : Continuous d.emb0

theorem **wittDecomp_anisotropicEmbedding_injective** : Function.Injective d.emb0

theorem **embRel_isometric_embedding** (ι : S₀.V →ₗ[D.host.E] S.V) (hι : frameMat S S₀ ι = d.i₀) : EmbRel S S₀ ι (d.emb0 g) g

theorem **images_nB** (b : Mx D ρ S₀.rank) : Images S S₀ d.jX (d.j0 + d.jX · b) (d.jY − d.j0 · plus S₀ b − d.jX · (half D • (b · plus S₀ b)))

theorem **images_nC** (c : skewA D ρ) : Images S S₀ d.jX d.j0 (d.jY + d.jX · c.1)

theorem **nB_jX** : mt S (d.nB b) · d.jX = d.jX

theorem **nB_j0** : mt S (d.nB b) · d.j0 = d.j0 + d.jX · b

theorem **nB_jY** : mt S (d.nB b) · d.jY = d.jY − d.j0 · plus S₀ b − d.jX · (half D • (b · plus S₀ b))

theorem **nC_jX** : mt S (d.nC c) · d.jX = d.jX

theorem **nC_j0** : mt S (d.nC c) · d.j0 = d.j0

theorem **nC_jY** : mt S (d.nC c) · d.jY = d.jY + d.jX · c.1

theorem **pX_nB** : d.pX · mt S (d.nB b) = d.pX + b · d.p0 − (half D • (b · plus S₀ b)) · d.pY

theorem **wittDecomp_projAnisotropic_unipotentMixed** : d.p0 · mt S (d.nB b) = d.p0 − plus S₀ b · d.pY

theorem **pY_nB** : d.pY · mt S (d.nB b) = d.pY

theorem **pX_nC** : d.pX · mt S (d.nC c) = d.pX + c.1 · d.pY

theorem **wittDecomp_projAnisotropic_unipotentSkew** : d.p0 · mt S (d.nC c) = d.p0

theorem **pY_nC** : d.pY · mt S (d.nC c) = d.pY

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.WittDecompSymplInclIsotropicInclAnisotropic

theorem **pairT_adj** (S : HSp D) {ρ : ℕ} (c : AE D) (ξ : Mx D ρ S.rank) (y : Mx D S.rank ρ) : pairT S ξ (c • (adj y · gA S)) = (cA D c • (y · ξ)).trace

theorem **extX_apply** (x : Mx D T.rank ρ) : d.extX T x = x · d.pX

theorem **extY_apply** (x : Mx D T.rank ρ) : d.extY T x = x · d.pY

theorem **matrixExtendByZero_apply** (w : WO S₀ T) : d.ext0 T w = w · d.p0

theorem **hO_coordS** (w w' : WO S T) : hO S T w w' = pairS T (w · d.jX) (w' · d.jY) + hO S₀ T (w · d.j0) (w' · d.j0) + pairS T (w · d.jY) (w' · d.jX)

theorem **hO_ext** (x x' y y' : Mx D T.rank ρ) (w w' : WO S₀ T) : hO S T (d.extX T x + d.ext0 T w + d.extY T y) (d.extX T x' + d.ext0 T w' + d.extY T y') = pairS T x y' + hO S₀ T w w' + pairS T y x'

theorem **symplOH_extY_extY** (y y' : Mx D T.rank ρ) : symplOH S T (d.extY T y) (d.extY T y') = 0

theorem **symplOH_extX_extX** (x x' : Mx D T.rank ρ) : symplOH S T (d.extX T x) (d.extX T x') = 0

theorem **wittDecomp_sympl_extDual_extAnisotropic** (y : Mx D T.rank ρ) (w : WO S₀ T) : symplOH S T (d.extY T y) (d.ext0 T w) = 0

theorem **wittDecomp_sympl_extIsotropic_extAnisotropic** (x : Mx D T.rank ρ) (w : WO S₀ T) : symplOH S T (d.extX T x) (d.ext0 T w) = 0

theorem **wittDecomp_sympl_extAnisotropic_extAnisotropic** (w w' : WO S₀ T) : symplOH S T (d.ext0 T w) (d.ext0 T w') = symplOH S₀ T w w'

theorem **symplOH_extX_extY** (x y : Mx D T.rank ρ) : symplOH S T (d.extX T x) (d.extY T y) = trHalf D (pairS T x y)

theorem **extX_levi** (a : GL (Fin ρ) (AE D)) (x : Mx D T.rank ρ) : d.extX T x · mt S (d.levi a) = d.extX T (x · a.1)

theorem **extX_nC** (c : skewA D ρ) (x : Mx D T.rank ρ) : d.extX T x · mt S (d.nC c) = d.extX T x + d.extY T (x · c.1)

theorem **extX_nB** (b : Mx D ρ S₀.rank) (x : Mx D T.rank ρ) : d.extX T x · mt S (d.nB b) = d.extX T x + d.ext0 T (x · b) − d.extY T (x · (half D • (b · plus S₀ b)))

theorem **wittDecomp_homAction_extAnisotropic_equivariant** (g : GA D S₀) (h : GA D T) (w : WO S₀ T) : actO S T (d.emb0 g) h (d.ext0 T w) = d.ext0 T (actO S₀ T g h w)

theorem **integrandS_eq** (f : WO S T → ℂ) (x a : Mx D T.rank ρ) (w₀ : WO S₀ T) : heisO ψ S T (d.extY T a) (heisO ψ S T (d.extX T x) (heisO ψ S T (d.ext0 T w₀) f)) 0 = ψ (symplOH S T (d.extY T a) (d.extX T x)) ∗ f (d.extY T a + d.extX T x + d.ext0 T w₀)

theorem **integrandS_periodic** {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (x : Mx D T.rank ρ) (w₀ : WO S₀ T) (γ : Mx D T.rank ρ) (hγ : γ ∈ ratMx D T.rank ρ) (a : Mx D T.rank ρ) : heisO ψ S T (d.extY T (γ + a)) (heisO ψ S T (d.extX T x) (heisO ψ S T (d.ext0 T w₀) f)) 0 = heisO ψ S T (d.extY T a) (heisO ψ S T (d.extX T x) (heisO ψ S T (d.ext0 T w₀) f)) 0

theorem **integrableOn_mixS** (hψ : Continuous ψ) {μ : Measure (Mx D T.rank ρ)} {𝓕 : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ 𝓕) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (x : Mx D T.rank ρ) (w₀ : WO S₀ T) : IntegrableOn (fun a => heisO ψ S T (d.extY T a) (heisO ψ S T (d.extX T x) (heisO ψ S T (d.ext0 T w₀) f)) 0) 𝓕 μ

theorem **mixS_indep** {μ μ' : Measure (Mx D T.rank ρ)} {𝓕 𝓕' : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ 𝓕) (hμ' : IsCovolOne (ratMx D T.rank ρ) μ' 𝓕') {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : d.mixS T ψ μ' 𝓕' f = d.mixS T ψ μ 𝓕 f

theorem **incX_apply** (ξ : Mx D ρ S.rank) : d.incX S ξ = d.jX · ξ

theorem **incY_apply** (η : Mx D ρ S.rank) : d.incY S η = d.jY · η

theorem **wittDecomp_inclAnisotropicHom_apply** (w : WO S T₀) : d.inc0 S w = d.j0 · w

theorem **incAdj_apply** (y : Mx D S.rank ρ) : d.incAdj S y = d.incY S (incA D (−D.host.xi0⁻¹) • (adj y · gA S))

theorem **wittDecomp_trace0Unit_mul_conj_neg_inv** : incA D D.host.xi0 ∗ cA D (incA D (−D.host.xi0⁻¹)) = 1

theorem **hO_coordT** (w w' : WO S T) : hO S T w w' = pairT S (d.pX · w) (d.pY · w') + hO S T₀ (d.p0 · w) (d.p0 · w') + pairT S (d.pY · w) (d.pX · w')

theorem **hO_inc** (ξ ξ' η η' : Mx D ρ S.rank) (w w' : WO S T₀) : hO S T (d.incX S ξ + d.inc0 S w + d.incY S η) (d.incX S ξ' + d.inc0 S w' + d.incY S η') = pairT S ξ η' + hO S T₀ w w' + pairT S η ξ'

theorem **symplOH_incX_incX** (ξ ξ' : Mx D ρ S.rank) : symplOH S T (d.incX S ξ) (d.incX S ξ') = 0

theorem **symplOH_incY_incY** (η η' : Mx D ρ S.rank) : symplOH S T (d.incY S η) (d.incY S η') = 0

theorem **wittDecomp_sympl_inclIsotropic_inclAnisotropic** (ξ : Mx D ρ S.rank) (w : WO S T₀) : symplOH S T (d.incX S ξ) (d.inc0 S w) = 0

theorem **wittDecomp_sympl_inclDual_inclAnisotropic** (η : Mx D ρ S.rank) (w : WO S T₀) : symplOH S T (d.incY S η) (d.inc0 S w) = 0

theorem **wittDecomp_sympl_inclAnisotropic_inclAnisotropic** (w w' : WO S T₀) : symplOH S T (d.inc0 S w) (d.inc0 S w') = symplOH S T₀ w w'

theorem **hO_incX_incY** (ξ η : Mx D ρ S.rank) : hO S T (d.incX S ξ) (d.incY S η) = pairT S ξ η

theorem **symplO_incX_incAdj** (ξ : Mx D ρ S.rank) (y : Mx D S.rank ρ) : symplO S T (d.incX S ξ) (d.incAdj S y) = Algebra.trace (AF D.host.F) (AE D) (y · ξ).trace

theorem **actO_incAdj** (g : GA D S) (a : GL (Fin ρ) (AE D)) (y : Mx D S.rank ρ) : actO S T g (d.levi a)⁻¹ (d.incAdj S y) = d.incAdj S (mt S g · y · a.1)

theorem **integrandT_eq** (f : WO S T → ℂ) (y : Mx D S.rank ρ) (ξ : Mx D ρ S.rank) (w₀ : WO S T₀) : heisO ψ S T (d.incX S ξ) (heisO ψ S T (d.incAdj S y) (heisO ψ S T (d.inc0 S w₀) f)) 0 = ψ (symplOH S T (d.incX S ξ) (d.incAdj S y)) ∗ f (d.incX S ξ + d.incAdj S y + d.inc0 S w₀)

theorem **integrandT_periodic** {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (y : Mx D S.rank ρ) (w₀ : WO S T₀) (γ : Mx D ρ S.rank) (hγ : γ ∈ ratMx D ρ S.rank) (ξ : Mx D ρ S.rank) : heisO ψ S T (d.incX S (γ + ξ)) (heisO ψ S T (d.incAdj S y) (heisO ψ S T (d.inc0 S w₀) f)) 0 = heisO ψ S T (d.incX S ξ) (heisO ψ S T (d.incAdj S y) (heisO ψ S T (d.inc0 S w₀) f)) 0

theorem **integrableOn_mixT** (hψ : Continuous ψ) {μ : Measure (Mx D ρ S.rank)} {𝓕 : Set (Mx D ρ S.rank)} (hμ : IsCovolOne (ratMx D ρ S.rank) μ 𝓕) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (y : Mx D S.rank ρ) (w₀ : WO S T₀) : IntegrableOn (fun ξ => heisO ψ S T (d.incX S ξ) (heisO ψ S T (d.incAdj S y) (heisO ψ S T (d.inc0 S w₀) f)) 0) 𝓕 μ

theorem **diagSp_gram** (N : ℕ) (neg : Fin N → Bool) (p q : ℕ) : (diagSp D N neg p q).gram = (Matrix.diagonal fun i => Examples₂.sgn (neg i) : Matrix (Fin N) (Fin N) D.host.E)

theorem **gram_check** {N k l : ℕ} {neg : Fin N → Bool} {p q : ℕ} (a : Matrix (Fin N) (Fin k) D.host.E) (b : Matrix (Fin N) (Fin l) D.host.E) (c : Matrix (Fin k) (Fin l) D.host.E) (h : aᴴ ∗ (Matrix.diagonal fun i => Examples₂.sgn (neg i) : Matrix (Fin N) (Fin N) D.host.E) ∗ b = c) : aᴴ · (show Matrix (Fin N) (Fin N) D.host.E from (diagSp D N neg p q).gram) · b = c

theorem **col_gram** {N : ℕ} (neg : Fin N → Bool) (a b : Matrix (Fin N) (Fin 1) E) (c : Matrix (Fin 1) (Fin 1) E) (h : ∑ k, star (a k 0) ∗ Examples₂.sgn (neg k) ∗ b k 0 = c 0 0) : aᴴ ∗ sgnM E neg ∗ b = c

theorem **plane_gram** (a b : Matrix (Fin 2) (Fin 1) E) (ha : a = !![1; 0]) (hb : b = !![0; 1]) : aᴴ ∗ sgnM E ![false, true] ∗ a = sgnM E ![false] ∧ bᴴ ∗ sgnM E ![false, true] ∗ b = sgnM E ![true] ∧ bᴴ ∗ sgnM E ![false, true] ∗ a = 0

theorem **three_gram** (e e' i : Matrix (Fin 3) (Fin 1) E) (he : e = !![0; 1; 1]) (he' : e' = !![0; 1 / 2; −(1 / 2)]) (hi : i = !![1; 0; 0]) : eᴴ ∗ sgnM E ![false, false, true] ∗ e = 0 ∧ e'ᴴ ∗ sgnM E ![false, false, true] ∗ e' = 0 ∧ e'ᴴ ∗ sgnM E ![false, false, true] ∗ e = 1 ∧ iᴴ ∗ sgnM E ![false, false, true] ∗ i = sgnM E ![false] ∧ iᴴ ∗ sgnM E ![false, false, true] ∗ e = 0 ∧ iᴴ ∗ sgnM E ![false, false, true] ∗ e' = 0

theorem **pairS_plus** (S₀ T : HSp D) {ρ : ℕ} (x : Mx D T.rank ρ) (b b' : Mx D ρ S₀.rank) : pairS T x (x · (b' · plus S₀ b)) = hO S₀ T (x · b) (x · b')

theorem **trHalf_half** (z : AE D) : trHalf D (half D ∗ z) + trHalf D (half D ∗ z) = trHalf D z

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.WittHyperbolicAnisotropicCommuteLevi

theorem **nB_mem_GQ** {b : Mx D ρ S₀.rank} (h : b ∈ ratMx D ρ S₀.rank) : d.nB b ∈ GQ S

theorem **nC_mem_GQ** {c : skewA D ρ} (h : c.1 ∈ ratMx D ρ ρ) : d.nC c ∈ GQ S

theorem **continuous_nB** : Continuous d.nB

theorem **continuous_nC** : Continuous d.nC

theorem **nB_zero** : d.nB 0 = 1

theorem **nC_zero** : d.nC 0 = 1

theorem **nC_add** (c c' : skewA D ρ) : d.nC (c + c') = d.nC c * d.nC c'

theorem **nB_mul_nB** (b₁ b₂ : Mx D ρ S₀.rank) : d.nB b₁ * d.nB b₂ = d.nB (b₁ + b₂) * d.nC (cComm S₀ b₁ b₂)

theorem **nB_inv** : (d.nB b)⁻¹ = d.nB (-b)

theorem **nC_inv** : (d.nC c)⁻¹ = d.nC (-c)

theorem **nC_mul_nB** : d.nC c * d.nB b = d.nB b * d.nC c

theorem **levi_mul_nB** (a : GL (Fin ρ) (AE D)) : d.levi a * d.nB b = d.nB (a.1 · b) * d.levi a

theorem **levi_mul_nC** (a : GL (Fin ρ) (AE D)) : d.levi a * d.nC c = d.nC (skewConj a.1 c) * d.levi a

theorem **wittHyperbolic_anisotropic_conj_nB** (g : GA D S₀) : d.emb0 g * d.nB b = d.nB (b · mt S₀ g⁻¹) * d.emb0 g

theorem **wittHyperbolic_anisotropic_commute_nC** (g : GA D S₀) : d.emb0 g * d.nC c = d.nC c * d.emb0 g

theorem **wittHyperbolic_anisotropic_commute_levi** (g : GA D S₀) (a : GL (Fin ρ) (AE D)) : d.emb0 g * d.levi a = d.levi a * d.emb0 g

theorem **mulVec_perp** {M : Mx D S.rank S.rank} {U : Mx D ρ ρ} {b : Mx D ρ S₀.rank} (h1 : M · d.jX = d.jX · U) (h2 : M · d.j0 = d.jX · b) (v : Fin S.rank → AE D) (h3 : d.pY *ᵥ v = 0) : M *ᵥ v = d.jX *ᵥ ((U · d.pX + b · d.p0) *ᵥ v)

theorem **pY_ratV** {x : Fin S.rank → D.host.E} (hx : d.qY *ᵥ x = 0) : d.pY *ᵥ ratV x = 0

theorem **subOne_mul** {k : ℕ} (g : GA D S) (y : Mx D S.rank k) : subOne S g · y = mt S g · y - y

theorem **mem_NFlag** {g : GA D S} {u : GL (Fin ρ) (AE D)} (hu : u ∈ upperUni ρ (AE D)) {b : Mx D ρ S₀.rank} (hX : mt S g · d.jX = d.jX · u.1) (h0 : mt S g · d.j0 = d.j0 + d.jX · b) : g ∈ NFlag S d.e

theorem **one_mem_upperUni** : (1 : GL (Fin ρ) (AE D)) ∈ upperUni ρ (AE D)

theorem **levi_mem_NFlag** {u : GL (Fin ρ) (AE D)} (hu : u ∈ upperUni ρ (AE D)) : d.levi u ∈ NFlag S d.e

theorem **nB_mem_NFlag** (b : Mx D ρ S₀.rank) : d.nB b ∈ NFlag S d.e

theorem **nC_mem_NFlag** (c : skewA D ρ) : d.nC c ∈ NFlag S d.e

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.WittSmallGroupEmbeddingMemPointwiseStabilizerIso

theorem **mem_fixer** {T : HSp D} {k : ℕ} {x : Mx D T.rank k} {g : GA D T} : g ∈ fixer T x ↔ mt T g · x = x

theorem **embA_apply** (h : GA D A) : embA d o h = d.emb0 (o.emb (h, 1))

theorem **liftS_apply** (ν 𝓕 G f) (g : GA D S) : liftS om ν 𝓕 G f g = ∫ h in 𝓕, om (g, h) f 0 * G h ∂ν

theorem **liftT_apply** (μ 𝓕 φ f) (h : GA D T) : liftT om μ 𝓕 φ f h = ∫ g in 𝓕, om (g, h) f 0 * φ g ∂μ

theorem **psiE_zero** (ψ : AddChar (AF D.host.F) ℂ) : psiE ψ 0 = 1

theorem **psiE_add** (ψ : AddChar (AF D.host.F) ℂ) (x y : AE D) : psiE ψ (x + y) = psiE ψ x * psiE ψ y

theorem **upSum_one** (ρ : ℕ) : upSum (1 : Mx D ρ ρ) = 0

theorem **mem_UX** {ρ : ℕ} {u : GL (Fin ρ) (AE D)} : u ∈ UX D ρ ↔ u ∈ upperUni ρ (AE D)

theorem **ell_vecMul** {ρ : ℕ} {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : ell D ρ ᵥ* u.1 = ell D ρ

theorem **zEll_mul** (T : HSp D) {ρ : ℕ} (z : Fin T.rank → D.host.E) {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : zEll T ρ z · u.1 = zEll T ρ z

theorem **yZero_mul** (S : HSp D) {ρ : ℕ} (z : Fin S.rank → D.host.E) {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : yZero S ρ z · u.1 = yZero S ρ z

theorem **ratV_mulVec** {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) (z : Fin n → D.host.E) : ratV (M *ᵥ z) = rat M *ᵥ ratV z

theorem **yZero_vecB** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ρ : ℕ) (ζ : Fin L.rank → D.host.E) : yZero S₀ ρ (vecB o ζ) = o.jB · yZero L ρ ζ

theorem **blockX_levi** (a : GL (Fin ρ) (AE D)) : blockX d (d.levi a) = a.1

theorem **blockX_of_fix** {g : GA D S} (hg : g ∈ stabX d) : blockX d g = 1

theorem **nB_mem_radX** (b : Mx D ρ S₀.rank) : d.nB b ∈ radX d

theorem **nC_mem_radX** (c : skewA D ρ) : d.nC c ∈ radX d

theorem **radX_le_stabX** : radX d ≤ stabX d

theorem **wittSmallGroupEmbedding_mem_pointwiseStabilizerIsotropic** (g : GA D S₀) : d.emb0 g ∈ stabX d

theorem **coefX_levi** (a : GL (Fin ρ) (AE D)) : coefX d z (d.levi a) = 0

theorem **coefX_nB** (b : Mx D ρ S₀.rank) : coefX d z (d.nB b) = b *ᵥ ratV z

theorem **coefX_nC** (c : skewA D ρ) : coefX d z (d.nC c) = 0

theorem **psiFlag_levi_mul** (u : GL (Fin ρ) (AE D)) {n : GA D S} (hn : n ∈ stabX d) : psiFlag ψ d (d.levi u * n) = psiUni ψ u

theorem **psiBes_levi** (u : GL (Fin ρ) (AE D)) : psiBes ψ d 1 z (d.levi u) = psiUni ψ u

theorem **psiBes_nB** (θ : D.host.E) (b : Mx D ρ S₀.rank) : psiBes ψ d θ z (d.nB b) = psiE ψ (incA D θ * ell D ρ ·ᵥ b *ᵥ ratV z)

theorem **psiBes_nC** (θ : D.host.E) (c : skewA D ρ) : psiBes ψ d θ z (d.nC c) = 1

theorem **levi_xZero** (u : GL (Fin ρ) (AE D)) : mt S (d.levi u) · xZero d = xZero d · u.1

theorem **symplO_yZero** {V₀ W W₀ : HSp D} {ρ : ℕ} (dW : HypDec W W₀ ρ) (z : Fin V₀.rank → D.host.E) (b : Mx D ρ V₀.rank) : symplO V₀ W (dW.incAdj V₀ (yZero V₀ ρ z)) (xZero dW · b) = Algebra.trace (AF D.host.F) (AE D) (ell D ρ ·ᵥ b *ᵥ ratV z)

theorem **psi_yZero** (ψ : AddChar (AF D.host.F) ℂ) {V V₀ W W₀ : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (z : Fin V₀.rank → D.host.E) (b : Mx D ρ V₀.rank) : ψ (symplO V₀ W (dW.incAdj V₀ (yZero V₀ ρ z)) (xZero dW · b)) = psiBes ψ dV 1 z (dV.nB b)

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **levi_inv_incAdj** (a : GL (Fin ρ) (AE D)) : mt T (d.levi a)⁻¹ · d.incAdj S y = d.incAdj S (y · a.1)

theorem **jacT_levi_mul** {a : GL (Fin ρ) (AE D)} (ha : y · a.1 = y) (n : GA D T) : jacT ψ d S y (d.levi a * n) = jacT ψ d S y n

theorem **jacT_one** : jacT ψ d S y 1 = id

theorem **ptT_nB_mul_nC** (b : Mx D ρ T₀.rank) (c : skewA D ρ) : ptT d S y (d.nB b ∗ d.nC c) = plus T₀ b · (incA D (-D.host.xi0⁻¹) • (adj y · gA S))

theorem **cenT_nC** (c : skewA D ρ) : cenT d S y (d.nC c) = d.qT S c y

theorem **incAdj_jB** (y : Mx D L.rank ρ) : d.incAdj S₀ (o.jB · y) = d.incAdj L y · o.pB

theorem **ptT_yZero_vecB** (ζ : Fin L.rank → D.host.E) (n : GA D T) : o.resS T₀ (ptT d S₀ (yZero S₀ ρ (vecB o ζ)) n) = (0, ptT d L (yZero L ρ ζ) n)

theorem **cenT_yZero_vecB** (ζ : Fin L.rank → D.host.E) (n : GA D T) : cenT d S₀ (yZero S₀ ρ (vecB o ζ)) n = cenT d L (yZero L ρ ζ) n

theorem **jacS_levi_mul** {a : GL (Fin ρ) (AE D)} (ha : x · a.1 = x) (n : GA D S) : jacS ψ d T x (d.levi a ∗ n) = jacS ψ d T x n

theorem **jacS_one** : jacS ψ d T x 1 = id

theorem **ptS_nB_mul_nC** (b : Mx D ρ S₀.rank) (c : skewA D ρ) : ptS d T x (d.nB b ∗ d.nC c) = x · b

theorem **cenS_nC** (c : skewA D ρ) : cenS d T x (d.nC c) = d.qS T c x

theorem **qX_e** : d.qX ∗ d.e = 1

theorem **isIsoSub_e** : IsIsoSub V d.e

theorem **superSum_const** (t : AE D) (T : Mx D ρ ρ) (b : Fin ρ → AE D) : superSum (fun _ => t) T b = t ∗ (upSum T + ell D ρ ·ᵥ b)

theorem **upSum_add** (T T' : Mx D ρ ρ) : upSum (T + T') = upSum T + upSum T'

theorem **besChar_toBesDat** {n : GA D V} (hn : n ∈ flagN d) : besChar ψ (toBesDat d o hζ hθ ι hι) n = psiBes ψ d θ (vecB o ζ) n

theorem **exists_blocks** {n : GA D S} (hn : n ∈ flagN d) : ∃ (T : Mx D ρ ρ) (b : Mx D ρ S₀.rank), GlobalTheory.BesselChar.StrictUp T ∧ mt S n · d.jX = d.jX + d.jX · T ∧ mt S n · d.j0 = d.j0 + d.jX · b

theorem **simpleSum_eq** (z : Fin S₀.rank → D.host.E) {n : GA D S} {T : Mx D ρ ρ} {b : Mx D ρ S₀.rank} (h1 : mt S n · d.jX = d.jX + d.jX · T) (h2 : mt S n · d.j0 = d.j0 + d.jX · b) : simpleSum d z n = superSum (fun _ => 1) T (b ∗ᵥ ratV z)

theorem **simpleSum_mul** (z : Fin S₀.rank → D.host.E) {n n' : GA D S} (hn : n ∈ flagN d) (hn' : n' ∈ flagN d) : simpleSum d z (n ∗ n') = simpleSum d z n + simpleSum d z n'

theorem **psiBes_mul** (θ : D.host.E) (z : Fin S₀.rank → D.host.E) {n n' : GA D S} (hn : n ∈ flagN d) (hn' : n' ∈ flagN d) : psiBes ψ d θ z (n ∗ n') = psiBes ψ d θ z n ∗ psiBes ψ d θ z n'

theorem **psiFlag_eq** (n : GA D S) : psiFlag ψ d n = psiBes ψ d 1 0 n

theorem **psiFlag_mul** {n n' : GA D S} (hn : n ∈ flagN d) (hn' : n' ∈ flagN d) : psiFlag ψ d (n ∗ n') = psiFlag ψ d n ∗ psiFlag ψ d n'

Development.AbelianVarietyTheory.Definitions.TwistedInnerProducts.ZeroDegreeOfPinFor

theorem **heckePin_iff** : HeckePin Rl cG S Y ↔ PinFor Rl cG S Y D.nexp.hkF

theorem **covered_of_allPlaces** (hL : Rl.toRealDat.toLevCh.Is A3.A.μ0) (hA : Rl.AllPlaces) : Covered Rl

theorem **j_injective** (hU : Rl.toRealDat.LUnif) (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ) : Function.Injective ((Rl.FW i).j σ)

theorem **reads_unique** (hL : Rl.toRealDat.toLevCh.Is A3.A.μ0) (hF : Rl.toRealDat.IsFam) (hU : Rl.toRealDat.LUnif) (hW : Rl.toRealDat.models_HW) {f f' : ∀ j, ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)} (h : Reads Rl cG S Y f) (h' : Reads Rl cG S Y f') (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (k : ℕ) (P : PlE D.host) (hs : Formalism.BasicNotions.IsSplitPl D.host P) (ho : Formalism.IndefR.Bridge.SplitOff P (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E))) (hk : k ≤ D.R.r) : f j k P = f' j k P

theorem **hkFR_congr** (i : D.ntw.Tw) (j : ℕ) (b : Bool) {f f' : ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)} (hs : HeckeStabR S j f) (hs' : HeckeStabR S j f') (k : ℕ) (P : PlE D.host) (h : f k P = f' k P) : hkFR S i j b f hs k P = hkFR S i j b f' hs' k P

theorem **pinFor_baseChange** (hL : Rl.toRealDat.toLevCh.Is A3.A.μ0) (hF : Rl.toRealDat.IsFam) (hU : Rl.toRealDat.LUnif) (hW : Rl.toRealDat.models_HW) (hC : Covered Rl) (f : ∀ j, ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)) (hs : ∀ j, 1 ≤ j → j ≤ D.R.m → HeckeStabR S j (f j)) (hf : Reads Rl cG S Y f) : PinFor Rl cG S Y (baseChangeOps Rl cG S Y f hs)

theorem **exists_pinFor** (hL : Rl.toRealDat.toLevCh.Is A3.A.μ0) (hF : Rl.toRealDat.IsFam) (hU : Rl.toRealDat.LUnif) (hW : Rl.toRealDat.models_HW) (hA : Rl.AllPlaces) : ∃ hk, PinFor Rl cG S Y hk

theorem **row_of_pin** (hp : HeckePin Rl cG S Y) (f : ∀ j, ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)) (hs : ∀ j, 1 ≤ j → j ≤ D.R.m → HeckeStabR S j (f j)) (hf : Reads Rl cG S Y f) (eW : ∀ j, Module.End D.what₂.LL (S.WL j)) : ∃ H : NExpRealH Rl cG S Y, H.hL = f ∧ NExpHeckeRowX Rl cG S Y H eW ∧ ∀ i b P, ∃ u : (D.ntw.O i)ˣ, H.cHk i b 1 P = algebraMap (D.ntw.O i) (FractionRing (D.ntw.O i)) u

theorem **hkFR_of_smul** (i : D.ntw.Tw) (j : ℕ) (b : Bool) {f : ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)} (hs : HeckeStabR S j f) (k : ℕ) (P : PlE D.host) : (f k P = 0 → hkFR S i j b f hs k P = 0) ∧ (f k P = 1 → hkFR S i j b f hs k P = 1)

theorem **not_pinFor** (hk : ∀ i j b, ℕ → PlE D.host → Module.End (FractionRing (D.ntw.O i)) (D.nexp.Fm i j b)) (f : ∀ j, ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)) (hs : ∀ j, 1 ≤ j → j ≤ D.R.m → HeckeStabR S j (f j)) (hf : Reads Rl cG S Y f) (i : D.ntw.Tw) (P : PlE D.host) (hP : P ∉ D.ntw.S0 i) (hsp : Formalism.BasicNotions.IsSplitPl D.host P) (ho : ¬ Formalism.IndefR.Bridge.SplitOff P (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E))) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (b : Bool) (x : D.nexp.Fm i j b) (hx : x ≠ 0) : ¬ PinFor Rl cG S Y hk

theorem **exists_reads** (h : ∀ (i : Fin D.R.m) (P : PlE D.host), Formalism.IndefR.Bridge.SplitOff P (Rl.Sb : Set (Formalism.L2.DIV.Pl D.host.E)) → ∀ k : ℕ, k ≤ D.R.r → GlobalTheory.IndefiniteRowSideFullHecke.RationalHeckeScalars.RationalHeckeAt Rl.toRealDat i P k) : ∃ (f : ∀ j, ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)) (_ : ∀ j, 1 ≤ j → j ≤ D.R.m → HeckeStabR S j (f j)), Reads Rl cG S Y f

theorem **zero_degree_of_pinFor** (hU : Rl.toRealDat.LUnif) (hW : Rl.toRealDat.models_HW) (hC : Covered Rl) (hk : ∀ i j b, ℕ → PlE D.host → Module.End (FractionRing (D.ntw.O i)) (D.nexp.Fm i j b)) (hp : PinFor Rl cG S Y hk) (f : ∀ j, ℕ → PlE D.host → Module.End D.what₂.LL (S.WL j)) (hs : ∀ j, 1 ≤ j → j ≤ D.R.m → HeckeStabR S j (f j)) (hf : Reads Rl cG S Y f) (i : D.ntw.Tw) (P : PlE D.host) (hP : P ∉ D.ntw.S0 i) (hsp : Formalism.BasicNotions.IsSplitPl D.host P) (j : ℕ) (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) (b : Bool) (x : D.nexp.Fm i j b) : hk i j b 0 P x = x

Development.AbelianVarietyTheory.Definitions.TwoChartAtlas.AdditionLawSecondIdentity

theorem **isFrameG_iff** (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (ph1 ph2 : MvPowerSeries (Fin 3 × Fin d) S) (sy : MvPowerSeries (Fin 2 × Fin d) S) (ad : 𝒪 E → MvPowerSeries (Fin 2 × Fin d) S) (e : ∀ m, FrTyG t x0 d x m) : IsFrameG t x0 d x F act ph1 ph2 sy ad e ↔ (∀ m, (e m).IsGen) ∧ FamPh1 t x0 d x F ph1 e ∧ FamPh2 t x0 d x F ph2 e ∧ FamSy t x0 d x sy e ∧ FamAd t x0 d x act ad e

theorem **JEq.refl** {R : Type} [CommRing R] {σ : Type} (m : ℕ) (f : MvPowerSeries σ R) : JEq m f f

theorem **nuLocC_eq_nuAlg** {T : CDVR} {io : c.F →+* T.Fr} {y : IntFr Xo c T io} (Ch : IntChart T io y) (B : FB T.V (chD Ch.ch) (𝒪 E)) (hF : B.G.F = Ch.FT) (hA : B.A.act = Ch.actT) (e k : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : nuLocC Ch e k a n = letI

Development.AbelianVarietyTheory.Definitions.TwoChartAtlas.TwoChartSt

theorem **toJet_one** {κ : Type} [Finite κ] (m : ℕ) : toJet m (1 : MvPowerSeries κ R) = 1

theorem **toJet_X** {κ : Type} [Finite κ] (m : ℕ) (j : κ) : toJet m (X j : MvPowerSeries κ R) = zR m j

theorem **Related.refl** {d : ℕ} {O : Type} [CommRing O] [StarRing O] (B : FB R d O) : Related B B (pt R (n

Development.AbelianVarietyTheory.Definitions.Uniformization.ExistsInKNeOne

theorem **inK_iff_memKL** (x : T ⟶ B) : InK B L x ↔ MemKL B L x

theorem **inK_iff_locGen** (x : T ⟶ B) : InK B L x ↔ KerPow.LocGen L (K

theorem **kTriv_iff_inK** : KTriv B L ↔ ∀ (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (x : Over.mk t ⟶ B), InK B L x → x = toUnit _ ≫ η[B]

theorem **inK_one** (hL : L.IsLB) (T : Over S) : InK B L (1 : T ⟶ B)

theorem **InK.comp** {x : T ⟶ B} (hx : InK B L x) (u : T' ⟶ T) : InK B L (u ≫ x)

theorem **InK.translate** {x : T ⟶ B} (hx : InK B L x) (y : T ⟶ B) : KerPow.LocGen L (K

theorem **InK.inv** (hL : L.IsLB) {x : T ⟶ B} (hx : InK B L x) : InK B L x⁻¹

theorem **InK.mul** (hL : L.IsLB) {x y : T ⟶ B} (hx : InK B L x) (hy : InK B L y) : InK B L (x ∗ y)

@[simp] theorem **mem_kSubgroup** (hL : L.IsLB) {x : T ⟶ B} : x ∈ kSubgroup B hL T ↔ InK B L x

theorem **InK.eq_one** (hK : KTriv B L) {x : T ⟶ B} (hx : InK B L x) : x = 1

theorem **kTriv_iff_forall** : KTriv B L ↔ ∀ (T : Over S) (x : T ⟶ B), InK B L x → x = 1

theorem **inK_of_push** {S' : Scheme.{0}} {f : S' ⟶ S} {B' : Over S'} [GrpObj B'] {φ : B'.left ⟶ B.left} {L' : LB B'.left} (h : IsGrpPullback B' B f φ) (hL : L.IsLB) (hL' : L'.IsLB) (hw : LBIso L' (L.pull φ)) {Z : Over S'} (x : Z ⟶ B') (hx : InK B L (pushPt h.1.w x)) : InK B' L' x

theorem **inK_triv** (x : T ⟶ B) : InK B (Inh.trivLB B.left) x

Development.AbelianVarietyTheory.Definitions.Uniformization.ExistsUniqueSamePt

theorem **finExt_self** (V : Type) [CommRing V] [IsDomain V] [IsLocalRing V] (h : IsAdicComplete (IsLocalRing.maximalIdeal V) V) : FinExt V V

theorem **CDVR.finExt_self** {V : Type} [CommRing V] [IsDomain V] [IsLocalRing V] (h : CDVR V) : FinExt V V

theorem **cdvr_padicInt** (p : ℕ) [Fact p.Prime] : CDVR ℤ_[p]

theorem **PullB.refl** (D : DegPre R) (F : Type) [Field F] [Algebra (FractionRing R) F] (k : ℕ) (s : Raw D F k) : PullB D F (𝟙 _) k s s

Development.AbelianVarietyTheory.Definitions.Uniformization.ExistsUniqueSamePtRight

theorem **finExt_self** (V : Type) [CommRing V] [IsDomain V] [IsLocalRing V] [h : IsAdicComplete (IsLocalRing.maximalIdeal V) V] : FinExt V V

theorem **cdvr_padicInt** (p : ℕ) [Fact p.Prime] : CDVR ℤ_[p]

theorem **not_finExt_padic** (p : ℕ) [Fact p.Prime] : ¬ FinExt ℤ_[p] ℚ_[p]

theorem **existsUnique_samePt_right** (U : PreU I D A G) (x : TFK (R

theorem **existsUnique_samePt_left** (U : PreU I D A G) (y : TF (R

theorem **pullB_unique** {T : Over (Spec (CommRingCat.of R))} (φ : TF (R

theorem **pullB_id** (k : ℕ) (s : Raw D F k) : PullB D F (𝟙 _) k s s

theorem **gtTors_zero** {g h : GtPtF D F} (hz : h.IsZero) : GtTors D 0 g

theorem **gtTors_one** {g : GtPtF D F} (h : GtTors D 1 g) : g.b = 1

theorem **add_formEq** (a b : D.X) (U : Type) [CommGroup U] (h : ((D.P.A ⊗ D.P.A).left ⟶ D.P.A.left) → U) : ∏ k, h (addφ D a b k) ^ addε k = (∏ k, h ((lift (toUnit _) (fst D.P.A D.P.A)).left ≫ famφT D (𝟙_ _) (a + b) k) ^ DegPre.famε 2 k) ∗ (∏ k, h ((lift (toUnit _) (snd D.P.A D.P.A)).left ≫ famφT D (𝟙_ _) (a − b) k) ^ DegPre.famε 2 k) ∗ (∏ k, h ((lift (toUnit _) (fst D.P.A D.P.A ∗ snd D.P.A D.P.A)).left ≫ famφT D (𝟙_ _) a k) ^ DegPre.famε 1 k)⁻¹ ∗ (∏ k, h ((lift (toUnit _) (fst D.P.A D.P.A ∗ (snd D.P.A D.P.A)⁻¹)).left ≫ famφT D (𝟙_ _) b k) ^ DegPre.famε 1 k)⁻¹

theorem **not_allPre_false** : ¬ AllPre fun _ => False

theorem **pullG_unique** (U : PreU I D A G) {T : Over (Spec (CommRingCat.of R))} (φ : TF (R

theorem **exists_finExt_proper** (V : Type) [CommRing V] [IsDomain V] [IsLocalRing V] (h : CDVR V) : ∃ (W : Type) (_ : CommRing W) (_ : IsLocalRing W) (_ : Algebra V W), FinExt V W ∧ ¬ Function.Surjective (algebraMap V W)

theorem **sval_cs_one** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (r : Z ⟶ Y) (z : Z) (J) : sval (cs φ ε 1) r z J = 1

theorem **expF_one** : (zeroU R I).ExpF F k (cs _ _ 1) (fun _ => cs _ _ 1)

theorem **supp_zeroU** : (zeroU R I).Supp F k PUnit.unit

theorem **pullG_one** {T : Over (Spec (CommRingCat.of R))} (φ : TF (R

theorem **acts_iff_qkToAk** (U : PreU I D A G) {k : ℕ} (h : U.FE F k) (p : QkPt D F k) (x : TFK (R

theorem **samePt_mul** (U : PreU I D A G) {x x' : TFK (R

theorem **fe_zeroUq_iff** (q : (FractionRing R)ˣ) : (zeroUq R I q).FE F (k + 1) ↔ q ^ k = 1

theorem **sval_const** {X Y Z : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {s : LB.Sec (LB.LinComb (trivLB X) φ ε)} {B : Type} [CommRing B] {p : Y ⟶ Spec (CommRingCat.of B)} {a : B} (h : SecConst s (scal p a)) (r : Z ⟶ Y) (z : Z) (J) : sval s r z J = gs (𝟙 Z) z ⊤ (scal (r ≫ p) a)

theorem **gs_scal** {Y Z : Scheme.{0}} {B : Type} [CommRing B] (p : Y ⟶ Spec (CommRingCat.of B)) (a : B) (r : Z ⟶ Y) (z : Z) : gs r z ⊤ (scal p a) = gs (𝟙 Z) z ⊤ (scal (r ≫ p) a)

theorem **expT_zeroU** [IsHausdorff I R] {σ : (zeroU R I).SecGT (𝟙_ _) k} {f} (h : (zeroU R I).ExpT (𝟙_ _) k σ f) {a b : R} (ha : SecConst σ (scal (snd (𝟙_ _) (𝟙_ (Over (Spec (CommRingCat.of R))))).left a)) (hb : SecConst (f PUnit.unit) (scal (snd (𝟙_ _) (𝟙_ (Over (Spec (CommRingCat.of R))))).left b)) : a = b

theorem **expF_zeroU** [IsHausdorff I R] {t s} (h : (zeroU R I).ExpF F k t s) {a b : F} (ha : SecConst t (scal (fst (TFK (R

theorem **acts_zeroU** [IsHausdorff I R] (g : GtPtF (zeroData R I).toPre F) : (zeroU R I).Acts F k PUnit.unit g 1

Development.AbelianVarietyTheory.Definitions.Uniformization.ExistsUniqueTorusCompat

theorem **bcMap_charUnit** (f : S' ⟶ S) (χ : X) : (bcMap f X).appTop.hom (charUnit S X χ).val = (charUnit S' X χ).val

theorem **splitTorus_left_hom_ext** {Z : Scheme.{0}} (a b : Z ⟶ (splitTorus S X).left) (h : a ≫ (splitTorus S X).hom = b ≫ (splitTorus S X).hom) (hχ : ∀ χ, a.appTop.hom (charUnit S X χ).val = b.appTop.hom (charUnit S X χ).val) : a = b

theorem **eq_bcMap** {f : S' ⟶ S} {τ : (splitTorus S' X).left ⟶ (splitTorus S X).left} (h : τ ≫ (splitTorus S X).hom = (splitTorus S' X).hom ≫ f) (hχ : ∀ χ, τ.appTop.hom (charUnit S X χ).val = (charUnit S' X χ).val) : τ = bcMap f X

theorem **bcMap_id** : bcMap (𝟙 S) X = 𝟙 _

theorem **bcMap_comp** (f' : S'' ⟶ S') (f : S' ⟶ S) : bcMap (f' ≫ f) X = bcMap f' X ≫ bcMap f X

theorem **torusCompat_iff** {f : S' ⟶ S} {F : G'.left ⟶ G.left} {i' : splitTorus S' X ⟶ G'} {i : splitTorus S X ⟶ G} : TorusCompat f F i' i ↔ bcMap f X ≫ i.left = i'.left ≫ F

theorem **TorusCompat.refl** (i : splitTorus S X ⟶ G) : TorusCompat (𝟙 S) (𝟙 G.left) i i

theorem **TorusCompat.trans** {f' : S'' ⟶ S'} {f : S' ⟶ S} {F' : G''.left ⟶ G'.left} {F : G'.left ⟶ G.left} {i'' : splitTorus S'' X ⟶ G''} {i' : splitTorus S' X ⟶ G'} {i : splitTorus S X ⟶ G} (h' : TorusCompat f' F' i'' i') (h : TorusCompat f F i' i) : TorusCompat (f' ≫ f) (F' ≫ F) i'' i

theorem **torusCompat_id_iff** {i' i : splitTorus S X ⟶ G} : TorusCompat (𝟙 S) (𝟙 G.left) i' i ↔ i' = i

theorem **TorusCompat.of_comp** {H : Over S} (i : splitTorus S X ⟶ G) (φ : G ⟶ H) : TorusCompat (𝟙 S) φ.left i (i ≫ φ)

theorem **torusCompat_bcMap** (f : S' ⟶ S) : TorusCompat f (bcMap f X) (𝟙 (splitTorus S' X)) (𝟙 (splitTorus S X))

theorem **existsUnique_torusCompat** {f : S' ⟶ S} {F : G'.left ⟶ G.left} (hF : IsPullback F G'.hom G.hom f) (i : splitTorus S X ⟶ G) : ∃! i' : splitTorus S' X ⟶ G', TorusCompat f F i' i

theorem **TorusCompat.unique** {f : S' ⟶ S} {F : G'.left ⟶ G.left} (hF : IsPullback F G'.hom G.hom f) {i : splitTorus S X ⟶ G} {i₁ i₂ : splitTorus S' X ⟶ G'} (h₁ : TorusCompat f F i₁ i) (h₂ : TorusCompat f F i₂ i) : i₁ = i₂

theorem **TorusCompat.of_comp_right** {f' : S'' ⟶ S'} {f : S' ⟶ S} {F' : G''.left ⟶ G'.left} {F : G'.left ⟶ G.left} (hF : IsPullback F G'.hom G.hom f) (hF' : F' ≫ G'.hom = G''.hom ≫ f') {i'' : splitTorus S'' X ⟶ G''} {i' : splitTorus S' X ⟶ G'} {i : splitTorus S X ⟶ G} (h'' : TorusCompat (f' ≫ f) (F' ≫ F) i'' i) (h : TorusCompat f F i' i) : TorusCompat f' F' i'' i'

private theorem **torusCompat_mul_comp** [GrpObj G'] [GrpObj G] {f : S' ⟶ S} {F : G'.left ⟶ G.left} (hF : IsGrpPullback G' G f F) {i' : splitTorus S' X ⟶ G'} {i : splitTorus S X ⟶ G} [IsMonHom i] (h : TorusCompat f F i' i) {T : Over S'} (u v : T ⟶ splitTorus S' X) : (u ∗ v) ≫ i' = (u ≫ i') ∗ (v ≫ i')

theorem **TorusCompat.isMonHom** [GrpObj G'] [GrpObj G] {f : S' ⟶ S} {F : G'.left ⟶ G.left} (hF : IsGrpPullback G' G f F) {i' : splitTorus S' X ⟶ G'} {i : splitTorus S X ⟶ G} [IsMonHom i] (h : TorusCompat f F i' i) : IsMonHom i'

theorem **surjective_red** {A : Type} [CommRing A] {J : Ideal A} (hJ : IsNilpotent J) : Surjective (red J)

Development.AbelianVarietyTheory.Definitions.Uniformization.FinrankComplexReal

@[simp] theorem **sec_mk** (s : LB.Sec N) : (mk p s).sec = s

@[simp] theorem **mk_sec** (s : SecMod p N) : mk p s.sec = s

@[simp] theorem **add_f** (s t : SecMod p N) (j : N.ι) : (s + t).sec.f j = s.sec.f j + t.sec.f j

@[simp] theorem **neg_f** (s : SecMod p N) (j : N.ι) : (-s).sec.f j = - s.sec.f j

@[simp] theorem **smul_f** (c : F) (s : SecMod p N) (j : N.ι) : (c • s).sec.f j = resTop _ (scal p c) ∗ s.sec.f j

@[simp] theorem **zero_f** (j : N.ι) : (0 : SecMod p N).sec.f j = 0

@[**ext**] theorem ext {s t : SecMod p N} (h : ∀ j, s.sec.f j = t.sec.f j) : s = t

theorem **eq_zero_iff** (s : SecMod p N) : s = 0 ↔ ∀ j, s.sec.f j = 0

@[simp] theorem **sub_f** (s t : SecMod p N) (j : N.ι) : (s - t).sec.f j = s.sec.f j - t.sec.f j

theorem **sum_f** {I : Type*} (S : Finset I) (s : I → SecMod p N) (j : N.ι) : (∑ i ∈ S, s i).sec.f j = ∑ i ∈ S, (s i).sec.f j

theorem **sval_add** (s t : SecMod p (LB.LinComb L φ ε)) : sval (s + t).sec r z J = sval s.sec r z J + sval t.sec r z J

theorem **sval_smul** (c : F) (s : SecMod p (LB.LinComb L φ ε)) : sval (c • s).sec r z J = gs r z ⊤ (scal p c) ∗ sval s.sec r z J

theorem **sval_zero** : sval (0 : SecMod p (LB.LinComb L φ ε)).sec r z J = 0

theorem **sval_sum** {I : Type*} (S : Finset I) (s : I → SecMod p (LB.LinComb L φ ε)) : sval (∑ i ∈ S, s i).sec r z J = ∑ i ∈ S, sval (s i).sec r z J

theorem **resTop_top** (c : Γ(Y, ⊤)) : resTop ⊤ c = c

@[simp] theorem **ofFun_f** (a : Γ(Y, ⊤)) (j) : (ofFun p a).sec.f j = a

theorem **smul_ofFun** (c : F) (a : Γ(Y, ⊤)) : c • ofFun p a = ofFun p (scal p c ∗ a)

theorem **ofFun_injective** : Function.Injective (ofFun p)

theorem **ofFun_surjective** : Function.Surjective (ofFun p)

theorem **scal_specMap** {A : Type} [CommRing A] (f : F →+* A) (c : F) : scal (Spec.map (CommRingCat.ofHom f)) c = (Scheme.ΓSpecIso (CommRingCat.of A)).inv.hom (f c)

theorem **one_ne_zero_spec** [Nontrivial F] : ofFun (𝟙 (Spec (CommRingCat.of F))) 1 ≠ 0

theorem **finrank_complex_real** : Module.finrank ℝ (SecMod (Spec.map (CommRingCat.ofHom (algebraMap ℝ ℂ))) (Inh.trivLB (Spec (CommRingCat.of ℂ)))) = 2

theorem **conj_smul** : (Complex.I • ofFun (Spec.map (CommRingCat.ofHom (starRingEnd ℂ))) 1) = ofFun _ ((Scheme.ΓSpecIso (CommRingCat.of ℂ)).inv.hom (-Complex.I))

Development.AbelianVarietyTheory.Definitions.Uniformization.InKLbPowIffPowEqOne

theorem **hasGen_lbPow** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (n : ℕ) : KerPow.HasGen (lbPow L n) φ ε ↔ KerPow.HasGen L φ (fun k => n * ε k)

theorem **inK_lbPow_iff** (hL : L.IsLB) (x : T ⟶ B) (n : ℕ) : InK B (lbPow L n) x ↔ KerPow.LocGen L (K

theorem **inK_lbPow_iff_pow_eq_one** (h : KerPowSt) [IsLocallyNoetherian S] (P : PolAV S) {N : ℕ} (hN : 1 ≤ N) (x : T ⟶ P.A) : InK P.A (lbPow P.L N) x ↔ x ^ N = 1

theorem **inK_lbPow_zero** (hL : L.IsLB) (x : T ⟶ B) : InK B (lbPow L 0) x

theorem **inK_of_local** (x : T ⟶ B) (h : ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ InK B L (overResι V ≫ x)) : InK B L x

theorem **inK_of_hom** {B' : Over S} [GrpObj B'] (F : B' ⟶ B) [IsMonHom F] {L' : LB B'.left} (hL : L.IsLB) (hL' : L'.IsLB) (hw : LBIso L' (L.pull F.left)) (x : T ⟶ B') (hx : InK B L (x ≫ F)) : InK B' L' x

theorem **inK_pull_of_ker** {B' : Over S} [GrpObj B'] (F : B' ⟶ B) [IsMonHom F] (hL : L.IsLB) (x : T ⟶ B') (hx : x ≫ F = 1) : InK B' (L.pull F.left) x

theorem **inK_noCharts** (x : T ⟶ B) : InK B (PEmpty, fun i => i.elim, fun i => i.elim) x

theorem **two_torsion** (H : TorusAlgebraic) (h : KerPowSt) : ∃ (P : PolAV (Sp ℂ)) (x : 𝟙_ (Over (Sp ℂ)) ⟶ P.A), P.g = 1 ∧ ¬ InK P.A P.L x ∧ InK P.A (lbPow P.L 2) x ∧ ¬ InK P.A (lbPow P.L 3) x

Development.AbelianVarietyTheory.Definitions.Uniformization.IsEllipticCurveTwoAdic

theorem **exists_even_one** {S : Scheme.{0}} {G H : Over S} [GrpObj G] [IsCommMonObj G] [GrpObj H] (P : CP G) (i : H ⟶ G) (hi : i = 1) : ∃ σ : CubTriv P.c i, σ.Even

theorem **isCPPullback_cancel** {P : CP G} {P' : CP G'} {P'' : CP G''} {f : S' ⟶ S} {g : S'' ⟶ S'} {f₀ : S'' ⟶ S} {F : G'.left ⟶ G.left} {F' : G''.left ⟶ G'.left} {F₀ : G''.left ⟶ G.left} (h : IsCPPullback P' P f F) (n : IsCPPullback P'' P f₀ F₀) (e₁ : g ≫ f = f₀) (e₂ : F' ≫ F = F₀) (w : F' ≫ G'.hom = G''.hom ≫ g) : IsCPPullback P'' P' g F'

theorem **abMap_q** (F : G'.left ⟶ G.left) : (TQ.abelian G' hG' X).q.left ≫ abMap hG hG' X F = F ≫ (TQ.abelian G hG X).q.left

theorem **abMap_comp** {Y : Scheme.{0}} {F : G'.left ⟶ G.left} {F₁ : G.left ⟶ Y} {F₀ : G'.left ⟶ Y} (h : F ≫ F₁ = F₀) : abMap hG hG' X F ≫ abMap' hG X F₁ = abMap' hG' X F₀

theorem **abMap_pb** {P : CP G} {P' : CP G'} {f : S' ⟶ S} {F : G'.left ⟶ G.left} (h : IsCPPullback P' P f F) : IsCPPullback (TQ.Desc.abelian (X

theorem **abMap'_pb** {P : CP G} {P' : CP G'} {f : S' ⟶ S} {F : G'.left ⟶ G.left} (h : IsCPPullback P' P f F) : IsCPPullback (TQ.Desc.abelian (X

theorem **ab_tor** {f : S' ⟶ S} {F : G'.left ⟶ G.left} (hg : IsGrpPullback G' G f F) : TorusCompat f F (TQ.abelian G' hG' X).i (TQ.abelian G hG X).i

theorem **ab_even** (P : CP G) : ∃ σ : CubTriv P.c (TQ.abelian G hG X).i, σ.Even

theorem **thickMap_fst** : thickMap R I e G ≫ pullback.fst G.hom (thick R I (e + 1)) = pullback.fst G.hom (thick R I e)

theorem **thickMap_snd** : thickMap R I e G ≫ (thickObj R I (e + 1) G).hom = (thickObj R I e G).hom ≫ thickTrans R I e

theorem **isGrpPullback_thickMap** : IsGrpPullback (thickObj R I e G) (thickObj R I (e + 1) G) (thickTrans R I e) (thickMap R I e G)

theorem **isCPPullback_level** [IsCommMonObj G] {P : CP G} {P₀ : CP (thickObj R I e G)} (h₀ : IsLevelCP I e P₀ P) : IsCPPullback P₀ P (thick R I e) (pullback.fst G.hom (thick R I e))

theorem **isCPPullback_levels** [IsCommMonObj G] {P : CP G} {P₀ : CP (thickObj R I e G)} {P₁ : CP (thickObj R I (e + 1) G)} (h₀ : IsLevelCP I e P₀ P) (h₁ : IsLevelCP I (e + 1) P₁ P) : IsCPPullback P₀ P₁ (thickTrans R I e) (thickMap R I e G)

theorem **nonempty_lev_torus** (X : Type) [AddCommGroup X] : Nonempty (Lev I (CP.triv (splitTorus (Spec (CommRingCat.of R)) X)) X e)

theorem **good_of_degPre** (hB : PolBaseChangeSt) (hV : CDVR V) (D : DegPre V) : ∃ (A : PolAV (Spec (CommRingCat.of (FractionRing V)))) (s : HidaSetting V A D.P.A) (t : Tower s), Nonempty t.Alg

theorem **good_of_polAV** (hL : Laws) (hV : CDVR V) (P : PolAV (Spec (CommRingCat.of V))) : ∃ (A : PolAV (Spec (CommRingCat.of (FractionRing V)))) (s : HidaSetting V A P.A) (t : Tower s), Nonempty t.Alg

theorem **zero_padic** (p : ℕ) [Fact p.Prime] : ∃ (s : HidaSetting ℤ_[p] (Inh.zeroPolAV _) (𝟙_ (Over (Spec (CommRingCat.of ℤ_[p]))))) (t : Tower s), Nonempty t.Alg

theorem **not_allSet_false** : ¬ AllSet fun _ => False

theorem **repWt_one** {s : HidaSetting V A G} {t : Tower s} (a : t.Alg) (e : ℕ) : a.RepWt 1 0 e

theorem **cubeSt'** : CubeSt

theorem **exists_hidaSetting_tower_of_cdvr** (hV : CDVR V) (P : PolAV (Spec (CommRingCat.of V))) : ∃ (A : PolAV (Spec (CommRingCat.of (FractionRing V)))) (s : HidaSetting V A P.A) (t : Tower s) (a : t.Alg), a.B = P.A ∧ s.P.L.L = P.L

theorem **elliptic_two_adic** : ∃ (A : PolAV (Spec (CommRingCat.of (FractionRing ℤ_[2])))) (s : HidaSetting ℤ_[2] A (GlobalTheory.WeierstrassCurve.ellPolAV curveTwoAdic).A) (t : Tower s) (a : t.Alg), a.B = (GlobalTheory.WeierstrassCurve.ellPolAV curveTwoAdic).A ∧ ∀ u : s.P.L.L.Gen, ¬ u.IsGen

Development.AbelianVarietyTheory.Definitions.Uniformization.NotExistsEvenTwistSymOne

theorem **TQ.comp_i_q** (Q : TQ G X) {T : Over S} (t : T ⟶ splitTorus S X) : (t ≫ Q.i) ≫ Q.q = 1

theorem **TQ.Desc.cubV_torus** {Q : TQ G X} {P : CP G} (D : Q.Desc P) {T : Over S} {z : T.left} {J} (hJ : Good D.M.L.L z J) (h : Q.q.left ≫ Q.B.hom = G.hom ≫ 𝟙 S) (x : T ⟶ splitTorus S X) (y u : T ⟶ G) : @id (T.left.presheaf.stalk z)ˣ (cubV D.M.c (T

private theorem **comp_one_uniformization** {S : Scheme.{0}} {G : Over S} [GrpObj G] {T T' : Over S} (x : T ⟶ T') : x ≫ (1 : T' ⟶ G) = 1

private theorem **one_comp_of_isMonHom'** {S : Scheme.{0}} {G B : Over S} [GrpObj G] [GrpObj B] (q : G ⟶ B) [IsMonHom q] {T : Over S} : (1 : T ⟶ G) ≫ q = 1

private theorem **prod_mul_inv_four_eq_of_mul_eq** {U : Type} [CommGroup U] {C w a b c d e f g : U} (hv : e * f * g * w = a * b * c * d * C) : w * a⁻¹ * (w * b⁻¹) * (w * c⁻¹) * (w * d⁻¹) = C * (w * e⁻¹) * (w * f⁻¹) * (w * g⁻¹)

private theorem **oneGen_isGen_uniformization** {X Y : Scheme.{0}} (L : LB X) : (LB.oneGen L (Y

private theorem **mul_inv_eq_mul_mul_mul_inv** {U : Type} [CommGroup U] (w m s : U) : w * m⁻¹ = s * (w * (m * s)⁻¹)

private theorem **eq_mul_inv_of_one_eq_mul_inv_mul** {U : Type} [CommGroup U] {a b v : U} (h : 1 = a * b⁻¹ * v) : v = b * a⁻¹

theorem **TQ.Desc.exists_even** {Q : TQ G X} {P : CP G} (D : Q.Desc P) : ∃ σ : CubTriv P.c Q.i, σ.Even

theorem **exists_point_neg_one** : ∃ t : 𝟙_ (Over (Spec (CommRingCat.of ℚ))) ⟶ splitTorus (Spec (CommRingCat.of ℚ)) ℤ, t⁻¹ = t ∧ chV t 1 ≠ 1

private theorem **stalk_neg_one_ne_one_of_over_rat** {Z : Scheme.{0}} (z : Z) (f : Z ⟶ Spec (CommRingCat.of ℚ)) : (-1 : Z.presheaf.stalk z) ≠ 1

theorem **not_exists_even_twistSym_one** : ¬ ∃ σ : CubTriv ((CP.triv (splitTorus (Spec (CommRingCat.of ℚ)) ℤ)).twistSym 1).c (𝟙 _), σ.Even

theorem **isEmpty_desc_twistSym_one** : IsEmpty ((TQ.torus (Spec (CommRingCat.of ℚ)) ℤ).Desc ((CP.triv (splitTorus (Spec (CommRingCat.of ℚ)) ℤ)).twistSym 1))

theorem **CubWeil.sq_eq_one_of_weilIs_inv** (hG : GlobalFnsSt) (hA : IsAbelianScheme G) {L : RigLB G} {c : CubicalStr G L} {N : ℕ} {T : Over S} {x y : T ⟶ G} {ζ : Γ(T.left, ⊤)ˣ} (h : CubWeil c N x y ζ) (h' : WeilIs G L.L N x y ζ⁻¹) : ζ ^ 2 = 1

theorem **TQ.Desc.nonempty_biTriv** {Q : TQ G X} {P : CP G} (D : Q.Desc P) : Nonempty (BiTriv P.c Q.i)

private theorem **gs_id_appLE_eq_gs** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (U : Y.Opens) (h : ⊤ ≤ f ⁻¹ᵁ U) (a : Γ(Y, U)) (z : Z) : gs (𝟙 Z) z ⊤ ((f.appLE U ⊤ h).hom a) = gs f z U a

private theorem **gs_id_appTop_eq_gs** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (a : Γ(Y, ⊤)) (z : Z) : gs (𝟙 Z) z ⊤ (f.appTop.hom a) = gs f z ⊤ a

private theorem **exists_tq_i_eq_of_isAbelianScheme** {R : Type} [CommRing R] [IsDomain R] {I : Ideal R} {D : DegPre R} {e : ℕ} {G : Over (Spec (CommRingCat.of R))} [GrpObj G] [IsCommMonObj G] (g : GtCoord I D e G) (hB : IsAbelianScheme (thickObj R I e D.P.A)) : ∃ Q : TQ (thickObj R I e G) D.X, Q.i = g.i

theorem **nonempty_tq_of_gtCoord** {R : Type} [CommRing R] [IsDomain R] {I : Ideal R} {D : DegPre R} {e : ℕ} {G : Over (Spec (CommRingCat.of R))} [GrpObj G] [IsCommMonObj G] (g : GtCoord I D e G) (hB : IsAbelianScheme (thickObj R I e D.P.A)) : ∃ Q : TQ (thickObj R I e G) D.X, Q.i = g.i

Development.AbelianVarietyTheory.Definitions.Uniformization.PointEqOfSubsingleton

theorem **exists_good_pt** {L : LB X} (hL : L.IsLB) (z : Z) : ∃ J, Good L z J

theorem **scaleGen_isGen** {M : LB Y} (u : Γ(Y, ⊤)ˣ) {t : LB.Gen M} (ht : t.IsGen) : (scaleGen u t).IsGen

theorem **val_scaleGen** {L : LB X} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (u : Γ(Y, ⊤)ˣ) (t : LB.Gen (LB.LinComb L φ ε)) (r : Z ⟶ Y) {z : Z} {J} (hJ : Good L z J) : val (scaleGen u t) r z J = gm r z ⊤ u * val t r z J

theorem **unitGen_isGen** (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (unitGen φ ε).IsGen

theorem **val_unitGen** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (r : Z ⟶ Y) (z : Z) (J) : val (unitGen φ ε) r z J = 1

theorem **gm_top** (r : Z ⟶ Y) (z : Z) (u : Γ(Y, ⊤)ˣ) : gm r z ⊤ u = gm (𝟙 Z) z ⊤ (Units.map r.appTop.hom.toMonoidHom u)

theorem **one_left** (G : Over S) [GrpObj G] (T : Over S) : (1 : T ⟶ G).left = T.hom ≫ eSec G

theorem **cubV_trivCub** {T : Over S} (z : T.left) (J) (x y u : T ⟶ G) : cubV (trivCub G) z J x y u = 1

theorem **val_sym_triv** {Z : Scheme.{0}} (r : Z ⟶ G.left) (z : Z) (J) : val (CP.triv G).sym r z J = 1

theorem **CP.val_sym_one** (P : CP G) {T : Over S} {z : T.left} {J} (hJ : Good P.L.L z J) : val P.sym (1 : T ⟶ G).left z J = 1

theorem **CubTriv.triv_even** (i : H ⟶ G) : CubTriv.Even (P

theorem **toBase_mul_of_isMonHom** {A B : Over S} [GrpObj A] [GrpObj B] (q : A ⟶ B) [IsMonHom q] (hq : q.left ≫ B.hom = A.hom ≫ 𝟙 S) (T : Over S) (x y : T ⟶ A) : toBase hq (x * y) = toBase hq x * toBase hq y

theorem **toBase_mul_pullback** (G : Over S) [GrpObj G] (f : S' ⟶ S) (T : Over S') (x y : T ⟶ (Over.pullback f).obj G) : toBase (A

theorem **IsCPMap.triv** (mul : ∀ (T : Over S') (x y : T ⟶ G'), toBase hw (x * y) = toBase hw x * toBase hw y) : IsCPMap (CP.triv G') (CP.triv G) f F hw (fun _ => 1) where

theorem **gw_idIso** {X Y : Scheme.{0}} (L : LB X) (r : Y ⟶ X) (y : Y) {j k : L.ι} (hj : r y ∈ L.U j) (hk : r y ∈ L.U k) : gw (idIso L) r y j k = gm r y _ (L.g j k)

theorem **idIso_isGen** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) : (idIso L).IsGen

theorem **val_change** {X Y Z : Scheme.{0}} {L : LB X} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} (ht : t.IsGen) (r : Z ⟶ Y) {z : Z} {J J'} (hJ : Good L z J) (hJ' : Good L z J') : val t r z J = (∏ k, gm (r ≫ φ k) z _ (L.g (J (r ≫ φ k)) (J' (r ≫ φ k))) ^ ε k) * val t r z J'

theorem **cubV_toBase_id** {L : RigLB G} (c : CubicalStr G L) (h : 𝟙 G.left ≫ G.hom = G.hom ≫ 𝟙 S) {T : Over S} (z : T.left) (J) (x y u : T ⟶ G) : cubV c (T

theorem **prod_sym** {T : Over S} (x : T ⟶ G) {U : Type} [CommGroup U] (h : (T.left ⟶ G.left) → U) : ∏ k : Fin 2, h (x.left ≫ ![(ι[G]).left, 𝟙 G.left] k) ^ (![1, -1] : Fin 2 → ℤ) k = h (x⁻¹).left * (h x.left)⁻¹

theorem **IsCPMap.id** (P : CP G) : IsCPMap P P (𝟙 S) (𝟙 G.left) (by simp) (idIso P.L.L) where

theorem **one_left_comp** (mul : ∀ (T : Over S') (x y : T ⟶ G'), toBase hw (x * y) = toBase hw x * toBase hw y) (T : Over S') : (1 : T ⟶ G').left ≫ F = (T.hom ≫ f) ≫ eSec G

theorem **inv_left_comp** (mul : ∀ (T : Over S') (x y : T ⟶ G'), toBase hw (x * y) = toBase hw x * toBase hw y) {T : Over S'} (x : T ⟶ G') : (x⁻¹).left ≫ F = (x.left ≫ F) ≫ (ι[G]).left

theorem **IsCPMap.comp** {P : CP G} {P' : CP G'} {P'' : CP G''} {f' : S'' ⟶ S'} {F' : G''.left ⟶ G'.left} {hw' : F' ≫ G'.hom = G''.hom ≫ f'} {w : LB.Iso P'.L.L (P.L.L.pull F)} {w' : LB.Iso P''.L.L (P'.L.L.pull F')} (h : IsCPMap P' P f F hw w) (h' : IsCPMap P'' P' f' F' hw' w') (hw'' : (F' ≫ F) ≫ G.hom = G''.hom ≫ f' ≫ f) {w'' : LB.Iso P''.L.L (P.L.L.pull (F' ≫ F))} (Tw : Tri w w' w'') : IsCPMap P'' P (f' ≫ f) (F' ≫ F) hw'' w''

theorem **IsCPMap.exists_comp** {P : CP G} {P' : CP G'} {P'' : CP G''} {f' : S'' ⟶ S'} {F' : G''.left ⟶ G'.left} {hw' : F' ≫ G'.hom = G''.hom ≫ f'} {w : LB.Iso P'.L.L (P.L.L.pull F)} {w' : LB.Iso P''.L.L (P'.L.L.pull F')} (h : IsCPMap P' P f F hw w) (h' : IsCPMap P'' P' f' F' hw' w') (hw'' : (F' ≫ F) ≫ G.hom = G''.hom ≫ f' ≫ f) : ∃ w'', Tri w w' w'' ∧ IsCPMap P'' P (f' ≫ f) (F' ≫ F) hw'' w''

theorem **cub_alg** {U : Type} [CommGroup U] {C C' a b c d e g h k : U} (H : C = a * b⁻¹ * c⁻¹ * d⁻¹ * e * g * h * k⁻¹ * C') : C * b * c * d * k = a * e * g * h * C'

theorem **sym_cub_alg** {A A' U : Type} [CommGroup A] [CommGroup A'] [CommGroup U] (β : A' →* A) (W : A' → U) {θ : A → A → A → U} {θ' : A' → A' → A' → U} {σ : A → U} {σ' : A' → U} (hθ : ∀ a b d, θ (β a) (β b) (β d) = W (a * b * d) * (W (a * b))⁻¹ * (W (a * d))⁻¹ * (W (b * d))⁻¹ * W a * W b * W d * (W 1)⁻¹ * θ' a b d) (hσ : ∀ a, σ (β a) = W a⁻¹ * (W a)⁻¹ * σ' a) {x y u : A'} (e : θ (β x)⁻¹ (β y)⁻¹ (β u)⁻¹

∗ σ (β x ∗ β y) ∗ σ (β x ∗ β u) ∗ σ (β y ∗ β u) = θ (β x) (β y) (β u) ∗ σ (β x ∗ β y ∗ β u) ∗ σ (β x) ∗ σ (β y) ∗ σ (β u)) : θ' x⁻¹ y⁻¹ u⁻¹ ∗ σ' (x ∗ y) ∗ σ' (x ∗ u) ∗ σ' (y ∗ u) = θ' x y u ∗ σ' (x ∗ y ∗ u) ∗ σ' x ∗ σ' y ∗ σ' u

theorem **exists_isCPMap** (P : CP G) (mul : ∀ (T : Over S') (x y : T ⟶ G'), toBase hw (x ∗ y) = toBase hw x ∗ toBase hw y) (L' : LB G'.left) (hL' : L'.IsLB) (w : LB.Iso L' (P.L.L.pull F)) (hg : w.IsGen) : ∃ (rig : LB.Gen (L'.pull (eSec G'))) (hr : rig.IsGen) (c : CubicalStr G' (L', rig, hL', hr)) (s : _) (h₁ : _) (h₂ : _) (h₃ : _), IsCPMap ⟨(L', rig, hL', hr), c, s, h₁, h₂, h₃⟩ P f F hw w

theorem **pullIso_isGen** {X Y : Scheme.{0}} {L : LB X} (hL : L.IsLB) (F : Y ⟶ X) : (pullIso L F).IsGen

theorem **exists_pull** (P : CP G) (mul : ∀ (T : Over S') (x y : T ⟶ G'), toBase hw (x ∗ y) = toBase hw x ∗ toBase hw y) : ∃ (P' : CP G') (w : _), IsCPMap P' P f F hw w

theorem **exists_isLevelCP** {R : Type} [CommRing R] [IsDomain R] (I : Ideal R) (e : ℕ) {G : Over (Spec (CommRingCat.of R))} [GrpObj G] [IsCommMonObj G] (P : CP G) : ∃ P' : CP (thickObj R I e G), IsLevelCP I e P' P

theorem **isLevelCP_triv** {R : Type} [CommRing R] [IsDomain R] (I : Ideal R) (e : ℕ) (G : Over (Spec (CommRingCat.of R))) [GrpObj G] [IsCommMonObj G] : IsLevelCP I e (CP.triv (thickObj R I e G)) (CP.triv G)

theorem **sym_cub_of_surjective** (h3 : Function.Surjective (G3 G).hom.appTop.hom) {L : RigLB G} (c : CubicalStr G L) (sym : LB.Gen (LB.LinComb L.L (K

theorem **surjective_cube** (hG : GlobalFnsSt) (hA : IsAbelianScheme G) : Function.Surjective (G3 G).hom.appTop.hom

theorem **CubWeil.unique** {ζ ζ' : Γ(T.left, ⊤)ˣ} (h : CubWeil c N x y ζ) (h' : CubWeil c N x y ζ') : ζ = ζ'

theorem **cubWeil_iff_isRatio** {ζ : Γ(T.left, ⊤)ˣ} : CubWeil c N x y ζ ↔ x ^ N = 1 ∧ y ^ N = 1 ∧ WeilCube.IsRatio c N x y ζ

theorem **existsUnique_cubWeil** (c : CubicalStr G L) (hx : x ^ N = 1) (hy : y ^ N = 1) : ∃! ζ, CubWeil c N x y ζ

theorem **CubWeil.symm** {ζ : Γ(T.left, ⊤)ˣ} (h : CubWeil c N x y ζ) : CubWeil c N y x ζ⁻¹

theorem **cubWeil_self** (c : CubicalStr G L) (hx : x ^ N = 1) : CubWeil c N x x 1

theorem **CubWeil.comp** {ζ : Γ(T.left, ⊤)ˣ} (h : CubWeil c N x y ζ) {T' : Over S} (σ : T' ⟶ T) : CubWeil c N (σ ≫ x) (σ ≫ y) (Units.map σ.left.appTop.hom.toMonoidHom ζ)

theorem **cubWeil_iff_weilIs** (hG : GlobalFnsSt) (hA : IsAbelianScheme G) (hx : x ^ N = 1) (hy : y ^ N = 1) (ζ : Γ(T.left, ⊤)ˣ) : CubWeil c N x y ζ ↔ WeilIs G L.L N x y ζ

theorem **cubWeil_triv** (hx : x ^ N = 1) (hy : y ^ N = 1) : CubWeil (trivCub G) N x y 1

theorem **BiTriv.val_one_right** (σ : BiTriv c i) (hJ : Good L.L z J) (t : T ⟶ H) : val σ.s (lift t (1 : T ⟶ G)).left z J = 1

theorem **BiTriv.val_one_left** [IsMonHom i] (σ : BiTriv c i) (hJ : Good L.L z J) (x : T ⟶ G) : val σ.s (lift (1 : T ⟶ H) x).left z J = 1

theorem **BiTriv.cubWeil** [IsMonHom i] (σ : BiTriv c i) {t : T ⟶ H} (ht : t ^ N = 1) (hx : x ^ N = 1) : CubWeil c N (t ≫ i) x 1

theorem **CubTriv.val_one** [IsMonHom i] (σ : CubTriv c i) (hJ : Good L.L z J) : val σ.s (1 : T ⟶ H).left z J = 1

theorem **chV_val** {Z : Over S} (t : Z ⟶ splitTorus S X) (χ : X) : (chV t χ).val = t.left.appTop.hom (charUnit S X χ).val

theorem **chV_mul** {Z : Over S} (t t' : Z ⟶ splitTorus S X) (χ : X) : chV (t ∗ t') χ = chV t χ ∗ chV t' χ

theorem **chV_inv** {Z : Over S} (t : Z ⟶ splitTorus S X) (χ : X) : chV t⁻¹ χ = (chV t χ)⁻¹

theorem **chV_add** {Z : Over S} (t : Z ⟶ splitTorus S X) (χ χ' : X) : chV t (χ + χ') = chV t χ ∗ chV t χ'

theorem **chV_zero** {Z : Over S} (t : Z ⟶ splitTorus S X) : chV t 0 = 1

theorem **gm_charUnit** {Z : Over S} (t : Z ⟶ splitTorus S X) (z : Z.left) (χ : X) : gm t.left z ⊤ (charUnit S X χ) = gm (𝟙 Z.left) z ⊤ (chV t χ)

theorem **CubTriv.val_twist** {L : RigLB G} {c : CubicalStr G L} (σ : CubTriv c i) (χ : X) {T : Over S} (x : T ⟶ splitTorus S X) {z : T.left} {J} (hJ : Good L.L z J) : val (σ.twist χ).s x.left z J = gm (𝟙 T.left) z ⊤ (chV x χ) ∗ val σ.s x.left z J

theorem **CubTriv.twist_even_iff** {P : CP G} {σ : CubTriv P.c i} (hσ : σ.Even) (χ : X) : (σ.twist χ).Even ↔ ∀ (T : Over S) (t : T ⟶ splitTorus S X), chV t (χ + χ) = 1

theorem **CubTriv.twist_even_twistSym** {P : CP (splitTorus S X)} {σ : CubTriv P.c (𝟙 _)} (hσ : σ.Even) (ψ : X) : CubTriv.Even (P

theorem **isWt_one** : IsWt i V 1 0

theorem **isWt_zero** (χ : X) : IsWt i V 0 χ

theorem **IsWt.mul** {u v : Γ(G.left, V)} {χ χ' : X} (hu : IsWt i V u χ) (hv : IsWt i V v χ') : IsWt i V (u ∗ v) (χ + χ')

theorem **IsWt.add** {u v : Γ(G.left, V)} {χ : X} (hu : IsWt i V u χ) (hv : IsWt i V v χ) : IsWt i V (u + v) χ

theorem **IsWt.inv** {u : Γ(G.left, V)ˣ} {χ : X} (hu : IsWt i V u.val χ) : IsWt i V (u⁻¹).val (-χ)

theorem **IsWt.res** {u : Γ(G.left, V)} {χ : X} (hu : IsWt i V u χ) (h : V' ≤ V) : IsWt i V' ((G.left.presheaf.map (homOfLE h).op).hom u) χ

theorem **isWt_top_iff** {u : Γ(G.left, ⊤)} {χ : X} : IsWt i ⊤ u χ ↔ ∀ (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G), ((t ≫ i) ∗ g).left.appTop.hom u = (chV t χ).val ∗ g.left.appTop.hom u

theorem **isWt_of_retraction** (p : G ⟶ splitTorus S X) [IsMonHom p] (hp : i ≫ p = 𝟙 _) (χ : X) : IsWt i ⊤ (chV p χ).val χ

theorem **isWt_charUnit** (χ : X) : IsWt (𝟙 (splitTorus S X)) ⊤ (charUnit S X χ).val χ

theorem **isAbelianScheme_unit** (S : Scheme.{0}) : IsAbelianScheme (𝟙_ (Over S))

theorem **point_eq_of_subsingleton** [Subsingleton X] {Z : Over S} (t t' : Z ⟶ splitTorus S X) : t = t'

theorem **TQ.exists_desc** {G : Over S} [GrpObj G] [IsCommMonObj G] (Q : TQ G X) (M : CP Q.B) : ∃ (P : CP G) (D : Q.Desc P), D.M = M

theorem **TQ.genWt_triv** {G : Over S} [GrpObj G] (Q : TQ G X) (u : Γ(G.left, ⊤)ˣ) {χ : X} (hu : IsWt Q.i ⊤ u.val χ) : Q.GenWt (Inh.trivLB _) (scaleGen u (unitGen _ _)) χ

theorem **exists_point_two** : ∃ t : 𝟙_ (Over (Spec (CommRingCat.of ℚ))) ⟶ splitTorus (Spec (CommRingCat.of ℚ)) ℤ, ∀ n : ℤ, chV t n = 1 → n = 0

Development.AbelianVarietyTheory.Definitions.Uniformization.RankOneModel

theorem **galois_all** : AllU fun {_} _ _ {_} {_} {_} {_} _ _ U => Cl.Galois U

theorem **functorial_all** : AllU fun {_} _ _ {_} {_} {_} {_} _ _ U => Cl.Functorial U

theorem **familyRing_all** : AllU fun {_} _ _ {_} {_} {_} {_} _ _ U => Cl.FamilyRing U

theorem **balanced_all** : AllU fun {_} _ _ {_} {_} {_} {_} _ _ U => Cl.Balanced U

theorem **seriesVal_all** (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] [CharZero V] (hV : IsAdicComplete (maximalIdeal V) V) (D : DegData V (maximalIdeal V)) : Cl.SeriesVal D

theorem **rank_one_model** : ∃ (A : PolAV (Spec (CommRingCat.of (FractionRing (PowerSeries ℚ))))) (G : Over (Spec (CommRingCat.of (PowerSeries ℚ)))) (_ : GrpObj G) (_ : IsCommMonObj G) (D : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (U : Unif (maximalIdeal (PowerSeries ℚ)) D A G), Module.finrank ℤ D.toPre.X = 1 ∧ Cl.Torsion U ∧ Cl.Galois U ∧ Cl.Functorial U ∧ Cl.FamilyRing U ∧ Cl.Balanced U ∧ Cl.SeriesVal D ∧ (∃ (p : QkPt D.toPre (AlgebraicClosure (FractionRing (PowerSeries ℚ))) (3 ^ 2)) (x : _) (w : _), p.y ≠ 0 ∧ QkToAkU U.toUnifPre _ (3 ^ 2) p x ∧ TXFrame (3 ^ 2) p w) ∧ ∃ (D' : DegData (PowerSeries ℚ) (maximalIdeal (PowerSeries ℚ))) (A' : PolAV (Spec (CommRingCat.of (FractionRing (PowerSeries ℚ))))) (G' : Over (Spec (CommRingCat.of (PowerSeries ℚ)))) (_ : GrpObj G') (_ : IsCommMonObj G'), IsSemiAbelian G' ∧ Nonempty (Unif (maximalIdeal (PowerSeries ℚ)) D' A' G') ∧ Nonempty (Is

Development.AbelianVarietyTheory.Definitions.Uniformization.SmulOneSecEqZeroIff

@[simp] theorem **coord_zero** (j : N.ι) : (0 : SecMod p N).sec.f j = 0

@[simp] theorem **coord_add** (s t : SecMod p N) (j : N.ι) : (s + t).sec.f j = s.sec.f j + t.sec.f j

@[simp] theorem **coord_neg** (s : SecMod p N) (j : N.ι) : (-s).sec.f j = - s.sec.f j

@[simp] theorem **coord_smul** (c : F) (s : SecMod p N) (j : N.ι) : (c • s).sec.f j = resTop (N.U j) (scal p c) ∗ s.sec.f j

@[simp] theorem **coord_sub** (s t : SecMod p N) (j : N.ι) : (s - t).sec.f j = s.sec.f j - t.sec.f j

theorem **coord_sum** {I : Type*} (S : Finset I) (s : I → SecMod p N) (j : N.ι) : (∑ i ∈ S, s i).sec.f j = ∑ i ∈ S, (s i).sec.f j

@[simp] theorem **coord_mk** (s : LB.Sec N) (j : N.ι) : (mk p s).sec.f j = s.f j

@[ext] theorem **ext_coord** {s t : SecMod p N} (h : ∀ j, s.sec.f j = t.sec.f j) : s = t

theorem **eq_zero_iff_coord** (s : SecMod p N) : s = 0 ↔ ∀ j, s.sec.f j = 0

theorem **value_zero** : sval (0 : SecMod p (LB.LinComb L φ ε)).sec r z J = 0

theorem **value_add** (s t : SecMod p (LB.LinComb L φ ε)) : sval (s + t).sec r z J = sval s.sec r z J + sval t.sec r z J

theorem **value_smul** (c : F) (s : SecMod p (LB.LinComb L φ ε)) : sval (c • s).sec r z J = gs r z ⊤ (scal p c) ∗ sval s.sec r z J

theorem **value_sum** {I : Type*} (S : Finset I) (s : I → SecMod p (LB.LinComb L φ ε)) : sval (∑ i ∈ S, s i).sec r z J = ∑ i ∈ S, sval (s i).sec r z J

theorem **oneSec_ne_zero** (F : Type) [CommRing F] [Nontrivial F] : oneSec F ≠ 0

theorem **smul_oneSec_eq_zero_iff** (F : Type) [CommRing F] (c : F) : c • oneSec F = 0 ↔ c = 0

Development.AbelianVarietyTheory.Definitions.Uniformization.SvalTwistOne

theorem **Lvl.precomp** {R : Type} [CommRing R] {I : Ideal R} {e : ℕ} {Z Z' : Scheme.{0}} {p : Z ⟶ Spec (CommRingCat.of R)} (h : Lvl I e p) (q : Z' ⟶ Z) : Lvl I e (q ≫ p)

theorem **sval_twist** {W : Type} [CommRing W] [Algebra R W] {k : ℕ} (a : Multiplicative D.X →* Wˣ) (f : ∀ α, SecBT D (TA R W) k α) (α : D.X) {Z : Scheme.{0}} (r : Z ⟶ (TA R W ⊗ D.P.A).left) (z : Z) {J} (hJ : Good D.P.L z J) : sval (twist a f α) r z J = gs r z ⊤ (scal (R

Development.AbelianVarietyTheory.Definitions.UpperBounds.FreeBallIffOfDimXEqZero

theorem **exp_neg_rhX** (h0 : IsOrth X x0) (hx : x ∈ X.dom) : Real.exp (-rhX X x0 x) = ∏ τ : X.Ψ, ‖dmin X.Gm x τ ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ / ‖lminor ((mat x0 τ)⁻¹ ∗ mat x τ) ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ ^ 2

theorem **lminor_inv_mul_ne_zero** (h0 : IsOrth X x0) (hx : x ∈ X.dom) (τ : X.Ψ) : lminor ((mat x0 τ)⁻¹ ∗ mat x τ) ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩ ≠ 0

theorem **rhX_eq_sum_log** (h0 : IsOrth X x0) (hx : x ∈ X.dom) : rhX X x0 x = ∑ τ : X.Ψ, -Real.log (‖dmin X.Gm x τ ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ / ‖lminor ((mat x0 τ)⁻¹ ∗ mat x τ) ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ ^ 2)

theorem **rhX_eq_sum** (h0 : IsOrth X x0) (hx : x ∈ X.dom) : rhX X x0 x = -∑ τ : X.Ψ, Real.log ‖dmin X.Gm x τ ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ + ∑ τ : X.Ψ, Real.log (‖lminor ((mat x0 τ)⁻¹ ∗ mat x τ) ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ ^ 2)

theorem **rhX_nonneg** (h0 : IsOrth X x0) (hx : x ∈ X.dom) : 0 ≤ rhX X x0 x

theorem **neg_log_factor_le_rhX** (h0 : IsOrth X x0) (hx : x ∈ X.dom) (τ : X.Ψ) : -Real.log (‖dmin X.Gm x τ ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ / ‖lminor ((mat x0 τ)⁻¹ ∗ mat x τ) ⟨X.p0, Nat.lt_succ_of_le X.p0_le⟩‖ ^ 2) ≤ rhX X x0 x

theorem **rhX_rmul** (x0 x : Frame X.Ψ r) {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) : rhX X x0 (rmul x b) = rhX X x0 x

theorem **rhX_cpt** (h0 : IsOrth X x0) {z : V (nd X)} (hz : z ∈ OmF X x0) : rhX X x0 (cpt X x0 z) = rhF X x0 z

theorem **rhX_self** (h0 : IsOrth X x0) : rhX X x0 x0 = 0

theorem **measurable_rhX** (h0 : IsOrth X x0) : Measurable (rhX X x0)

theorem **continuousOn_rhX** (h0 : IsOrth X x0) : ContinuousOn (rhX X x0) X.dom

theorem **isOrth_lact_of_isUStab** {γ : GL (Fin r) E} (hγ : IsUStab X.Gm X.Λ γ) (h0 : IsOrth X x0) : IsOrth X (lact γ x0)

theorem **rhX_lact_lact** {γ : GL (Fin r) E} (hγ : IsUStab X.Gm X.Λ γ) (x0 x : Frame X.Ψ r) : rhX X (lact γ x0) (lact γ x) = rhX X x0 x

theorem **rhX_lact** {γ : GL (Fin r) E} (hγ : IsUStab X.Gm X.Λ γ) (x0 x : Frame X.Ψ r) : rhX X x0 (lact γ x) = rhX X (lact γ⁻¹ x0) x

theorem **leviRh_nonneg** (h0 : IsOrth X x0) {z : V (nd X)} (hz : z ∈ OmF X x0) (ξ : Fin (nd X) → ℂ) : 0 ≤ leviRh X x0 z ξ

theorem **dimX_eq_zero_iff** : dimX X = 0 ↔ X.p0 = 0 ∨ X.p0 = r

theorem **one_le_dimX_iff** : 1 ≤ dimX X ↔ 0 < X.p0 ∧ X.p0 < r

theorem **card_mixedEnt** : Fintype.card {e : Ent X // (e.2.1.2 : ℕ) < X.p0 ∧ X.p0 ≤ (e.2.1.1 : ℕ)} = dimX X

theorem **FreeBall.le** (h : FreeBall X G x0 ρ0) : X.Γ ≤ G

theorem **FreeBall.isUStab** (h : FreeBall X G x0 ρ0) {γ : GL (Fin r) E} (hγ : γ ∈ G) : IsUStab X.Gm X.Λ γ

theorem **FreeBall.eq_one** (h : FreeBall X G x0 ρ0) {γ : GL (Fin r) E} (hγ : γ ∈ G) (hx : x ∈ X.dom) (h1 : rhX X x0 x < ρ0) (h2 : rhX X x0 (lact γ x) < ρ0) : γ = 1

theorem **FreeBall.unique** (h : FreeBall X G x0 ρ0) {γ γ' : GL (Fin r) E} (hγ : γ ∈ G) (hγ' : γ' ∈ G) (hx : x ∈ X.dom) (h1 : rhX X x0 (lact γ x) < ρ0) (h2 : rhX X x0 (lact γ' x) < ρ0) : γ = γ'

theorem **FreeBall.mono** (h : FreeBall X G x0 ρ0) (hp : ρ1 ≤ ρ0) : FreeBall X G x0 ρ1

theorem **FreeBall.of_le** (h : FreeBall X G x0 ρ0) (h1 : X.Γ ≤ G') (h2 : G' ≤ G) : FreeBall X G' x0 ρ0

theorem **FreeBall.eq_of_lact_eq** (h : FreeBall X G x0 ρ0) (h0 : IsOrth X x0) (hp : 0 < ρ0) {γ γ' : GL (Fin r) E} (hγ : γ ∈ G) (hγ' : γ' ∈ G) {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (e : lact γ x0 = lact γ' (rmul x0 b)) : γ = γ'

theorem **FreeBall.lact** (h : FreeBall X G x0 ρ0) {g : GL (Fin r) E} (hg : g ∈ G) : FreeBall X G (lact g x0) ρ0

theorem **IsPole.lact** (h : IsPole X G x0 κ ε C δ ψ) {γ : GL (Fin r) E} (hγ : γ ∈ G) (hx : x ∈ X.dom) : ψ (lact γ x) = ψ x

theorem **IsPole.rmul** (h : IsPole X G x0 κ ε C δ ψ) (hx : x ∈ X.dom) {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) : ψ (rmul x b) = ψ x

theorem **IsPole.isInvFn** (h : IsPole X G x0 κ ε C δ ψ) (hG : X.Γ ≤ G) : IsInvFn X ψ

theorem **IsPole.mono** (h : IsPole X G x0 κ ε C δ ψ) (hG : G' ≤ G) (hκ : κ ≤ κ') (hC : C ≤ C') : IsPole X G' x0 κ' ε C' δ ψ

theorem **IsPole.le_log_orbit** (h : IsPole X G x0 κ ε C δ ψ) (hG : ∀ γ ∈ G, IsUStab X.Gm X.Λ γ) {γ : GL (Fin r) E} (hγ : γ ∈ G) (hx : x ∈ X.dom) (hle : rhX X (ReciprocityTheory.ChartRestriction.G.FD.lact γ x0) x ≤ ε) : ψ x ≤ dimX X * Real.log (rhX X (ReciprocityTheory.ChartRestriction.G.FD.lact γ x0) x + δ) + C

theorem **IsPole.centre** (h : IsPole X G x0 κ ε C δ ψ) (h0 : IsOrth X x0) (hε : 0 ≤ ε) : ψ x0 ≤ dimX X * Real.log δ + C

theorem **isPole_zero** (h0 : IsOrth X x0) (hκ : 0 ≤ κ) (hδ : 0 < δ) (hC : 0 ≤ C) (hC' : -(dimX X * Real.log δ) ≤ C) : IsPole X G x0 κ ε C δ fun _ => 0

theorem **blockConst_ite** (a a' : X.Ψ → ℤ) : BlockConst X fun τ i => if (i : ℕ) < X.p0 then a τ else a' τ

theorem **blockConst_iff** : BlockConst X la ↔ ∃ a a' : X.Ψ → ℤ, la = fun τ (i : Fin r) => if (i : ℕ) < X.p0 then a τ else a' τ

theorem **BlockConst.add** (h : BlockConst X la) (h' : BlockConst X la') : BlockConst X (la + la')

theorem **BlockConst.zsmul** (h : BlockConst X la) (k : ℤ) : BlockConst X (k • la)

theorem **BlockConst.nsmul** (h : BlockConst X la) (k : ℕ) : BlockConst X (k • la)

Development.AbelianVarietyTheory.Definitions.UpperBounds.SingleMemIntVec

theorem **_root_.GlobalTheory.Up.Definite.herm_one** (x y : Fin r → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm (1 : Matrix (Fin r) (Fin r) E) x y = ∑ a, x a * _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (y a)

theorem **_root_.GlobalTheory.Up.Definite.map_herm_one_self** (φ : E →+* ℂ) (x : Fin r → E) : φ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm (1 : Matrix (Fin r) (Fin r) E) x x) = ((∑ a, ‖φ (x a)‖ ^ 2 : ℝ) : ℂ)

theorem **_root_.GlobalTheory.Up.Definite.mem_intVec** (x : Fin r → E) : x ∈ _root_.GlobalTheory.Up.Definite.intVec E r ↔ ∀ a, IsIntegral ℤ (x a)

theorem **_root_.GlobalTheory.Up.Definite.single_mem_intVec** (b : Fin r) : (Pi.single b 1 : Fin r → E) ∈ _root_.GlobalTheory.Up.Definite.intVec E r

private theorem **finite_GL_of_scaled_entries_integral_bounded** (c : E) (hc : c ≠ 0) (B : ℝ) : {γ : GL (Fin r) E | ∀ a b, IsIntegral ℤ (c * (γ : Matrix (Fin r) (Fin r) E) a b) ∧ ∀ φ : E →+* ℂ, ‖φ (c * (γ : Matrix (Fin r) (Fin r) E) a b)‖ ≤ B}.Finite

theorem **_root_.GlobalTheory.Up.Definite.finite_small** : {γ : GL (Fin r) E | ∀ a b, IsIntegral ℤ ((γ : Matrix (Fin r) (Fin r) E) a b) ∧ ∀ φ : E →+* ℂ, ‖φ ((γ : Matrix (Fin r) (Fin r) E) a b)‖ ≤ 1}.Finite

theorem **gram_one** (τ : E →+* ℂ) (x : Matrix (Fin r) (Fin r) ℂ) : gram τ (1 : Matrix (Fin r) (Fin r) E) x = (x.map conj)ᴴ * x.map conj

theorem **mem_dom** (x : Frame (data E r).Ψ r) : x ∈ (data E r).dom ↔ ∀ τ, (mat x τ).det ≠ 0

theorem **lact_one** {Ψ : Finset (E →+* ℂ)} (x : Frame Ψ r) : lact 1 x = x

theorem **card_cmType** (X : Data E r) (h : Module.finrank ℚ E = 2) : Fintype.card X.Ψ = 1

Development.AbelianVarietyTheory.Definitions.UpperBounds.SumNormSqWPos

theorem **Wdiv_of_mem** (hx : x ∈ X.dom) : Wdiv X Xa Fl n x = ENNReal.ofReal (Real.exp (-phla X Xa.Fp x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2))

theorem **Wdiv_of_notMem** (hx : x ∉ X.dom) : Wdiv X Xa Fl n x = 0

theorem **sum_normSq_pos** {N : ℕ} {a : Fin N → ℂ} (h : ∃ k, a k ≠ 0) : 0 < ∑ k, ‖a k‖ ^ 2

theorem **sum_normSq_w_pos** (hx : x ∈ X.dom) : 0 < ∑ k, ‖Xa.w k x‖ ^ 2

theorem **Wdiv_eq_exp** (hx : x ∈ X.dom) (hs : ∃ k, Fl.s k x ≠ 0) : Wdiv X Xa Fl n x = ENNReal.ofReal (Real.exp (-Psdiv X Xa Fl n x))

theorem **Wdiv_eq_top** (hn : 1 ≤ n) (hx : x ∈ X.dom) (hs : ∀ k, Fl.s k x = 0) : Wdiv X Xa Fl n x = ⊤

theorem **phla_add_nsmul** (hx : x ∈ X.dom) (la la' : Wt X.Ψ r) (k : ℕ) : phla X (la + k • la') x = phla X la x + k * phla X la' x

theorem **pnorm_sq_eq_exp** (hx : x ∈ X.dom) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) : X.pnorm la P x ^ 2 = Real.exp (-phla X la x) * ‖P x‖ ^ 2

theorem **Wdiv_eq_pnorm** (hx : x ∈ X.dom) : Wdiv X Xa Fl n x = ENNReal.ofReal (Real.exp (-phla X (Xa.Fp + n • Xa.L) x)) * ENNReal.ofReal (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, X.pnorm Xa.L (Fl.s k) x ^ 2) ^ (-((n : ℝ) - 1 / 2))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeLawsOfCubeBundleGenerator

theorem **cube** (h : CubeGen) : CubeSt

theorem **square** (h : CubeGen) : SquareSt

theorem **mulPull** (h : CubeGen) : MulPullSt

theorem **kerPow** (h : CubeGen) : KerPowSt

theorem **weil** (h : CubeGen) : WeilSt

theorem **symLift** (h : CubeGen) : SymLiftSt

theorem **mulA_props** (h : CubeGen) : ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)

theorem **weilNondeg** (h : CubeGen) : WeilNondegSt

theorem **charPair** (h : CubeGen) : CharPairSt

theorem **sec_split** (h : CubeGen) : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n j : ℕ), (n : K) ≠ 0 → 1 ≤ j → Nonempty (SecM P.A P.L ((n ^ 2 * j : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K))) ≃ₗ[Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)] ({x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A // x ^ n = 1} → SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K)))))

theorem **all_of_cubeGen** : (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → CubeSt ∧ SquareSt ∧ MulPullSt ∧ KerPowSt ∧ WeilSt ∧ SymLiftSt ∧ WeilNondegSt ∧ CharPairSt ∧ (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) ∧ (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n j : ℕ), (n : K) ≠ 0 → 1 ≤ j → Nonempty (SecM P.A P.L ((n ^ 2 * j : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K))) ≃ₗ[Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)] ({x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A // x ^ n = 1} → SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K))))))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeLawsOfCubeBundleGeneratorStatement

theorem **_root_.GlobalTheory.T.abelianSchemeLaws_of_cubeBundleGenerator** : (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → CubeSt ∧ SquareSt ∧ MulPullSt ∧ KerPowSt ∧ WeilSt ∧ SymLiftSt ∧ WeilNondegSt ∧ CharPairSt ∧ (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) ∧ (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n j : ℕ), (n : K) ≠ 0 → 1 ≤ j → Nonempty (SecM P.A P.L ((n ^ 2 * j : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K))) ≃ₗ[Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)] ({x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A // x ^ n = 1} → SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K))))))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeLawsUnconditional

theorem **_root_.GlobalTheory.abelianSchemeLaws_all** : _root_.ReciprocityTheory.ChartRestriction.G.Out.Laws ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.Laws2

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeLawsUnconditionalStatement

theorem **_root_.GlobalTheory.T.abelianScheme_laws_unconditional** : Laws ∧ Laws2

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeLawsWithoutCohomology

theorem **polBaseChange_holds** : PolBaseChangeSt

theorem **cubical_extends_holds** : Semi.cubical_extendsSt.{0}

theorem **laws_without_cohomology** : GlobalFnsSt ∧ RigiditySt ∧ PolBaseChangeSt ∧ Semi.lemmaBSt.{0} ∧ Semi.cubical_extendsSt.{0}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeLawsWithoutCohomologyStatement

theorem **_root_.GlobalTheory.T.abelianScheme_laws_without_cohomology** : GlobalFnsSt ∧ RigiditySt ∧ PolBaseChangeSt ∧ Semi.lemmaBSt.{0} ∧ Semi.cubical_extendsSt.{0}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianSchemeTorusCompare

theorem **isAbelianScheme_of_iso** {S : Scheme.{0}} {A B : Over S} [GrpObj A] [GrpObj B] (e : A ≅ B) (he : IsMonHom e.hom) (hB : IsAbelianScheme B) : IsAbelianScheme A

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianVarietyLineBundleSectionsFinite

theorem **finSec_eq_zero_of_cover** {X : Scheme.{0}} {I : Type} (V : I → X.Opens) (hV : ⨆ i, V i = ⊤) {U : X.Opens} (x : Γ(X, U)) (hx : ∀ i, rs X (inf_le_left : U ⊓ V i ≤ U) x = 0) : x = 0

theorem **Perf.exists_relation** {X : Scheme.{0}} {K : Type} [Field K] {f : X ⟶ Spec (CommRingCat.of K)} {M : LB X} (Q : Perf f M) (s : Fin (Q.a + 1) → LB.Sec M) : ∃ c : Fin (Q.a + 1) → K, c ≠ 0 ∧ ∀ i, ∑ k, cst f (M.U i) (c k) * (s k).f i = 0

theorem **sections_finite_of_perf** (hperf : ∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)) (hva : ∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left) (hM : M.IsLB) : ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) * (s k).f i = 0

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AbelianVarietyLineBundleSectionsFiniteStatement

theorem **_root_.GlobalTheory.T.abelianVariety_lineBundle_sections_finite** : (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)) → (∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) → ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) * (s k).f i = 0

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.AcyclicityLemmaDepthDifferentialCokernelsFiniteL

theorem **subsingleton_ext_of_isZero_left** {X : ModuleCat.{0} R} (h : IsZero X) (Y : ModuleCat.{0} R) (n : ℕ) : Subsingleton (Abelian.Ext.{0} X Y n)

theorem **shortExact_submodule** {M : Type} [AddCommGroup M] [Module R M] (N : Submodule R M) : (ModuleCat.shortComplexOfCompEqZero N.subtype N.mkQ (LinearMap.exact_subtype_mkQ N).linearMap_comp_eq_zero).ShortExact

theorem **nonempty_iso_residue** [IsLocalRing R] (M : Type) [AddCommGroup M] [Module R M] [IsSimpleModule R M] : Nonempty (ModuleCat.of R M ≅ ModuleCat.of R (R ⧸ maximalIdeal R))

theorem **subsingleton_ext_of_isFiniteLength** [IsLocalRing R] (Y : ModuleCat.{0} R) (n : ℕ) (hk : Subsingleton (Abelian.Ext.{0} (ModuleCat.of R (R ⧸ maximalIdeal R)) Y n)) (M : Type) [AddCommGroup M] [Module R M] (hM : IsFiniteLength R M) : Subsingleton (Abelian.Ext.{0} (ModuleCat.of R M) Y n)

theorem **length_ext_of_isFiniteLength** [IsLocalRing R] (Y : ModuleCat.{0} R) (g : ℕ) (hk : ∀ n, n ≠ g → Subsingleton (Abelian.Ext.{0} (ModuleCat.of R (R ⧸ maximalIdeal R)) Y n)) (M : Type) [AddCommGroup M] [Module R M] (hM : IsFiniteLength R M) : Module.length R (Abelian.Ext.{0} (ModuleCat.of R M) Y g) = Module.length R M * Module.length R (Abelian.Ext.{0} (ModuleCat.of R (R ⧸ maximalIdeal R)) Y g)

theorem **subsingleton_ext_free** (X : ModuleCat.{0} R) (n : ℕ) (h : Subsingleton (Abelian.Ext.{0} X (ModuleCat.of R R) n)) (F : ModuleCat.{0} R) [Module.Free R F] [Module.Finite R F] : Subsingleton (Abelian.Ext.{0} X F n)

theorem **subsingleton_ext_residue** [IsRegularLocalRing R] (g : ℕ) (hg : ringKrullDim R = g) (n : ℕ) (hn : n ≠ g) : Subsingleton (Abelian.Ext.{0} (ModuleCat.of R (R ⧸ maximalIdeal R)) (ModuleCat.of R R) n)

theorem **opcyclesShortComplex_shortExact** (i : ℕ) (h : IsZero (C.homology i)) : (opcyclesShortComplex C i).ShortExact

theorem **isIso_pOpcycles_zero** : IsIso (C.pOpcycles 0)

theorem **acyclicityLemma_depth_differentialCokernels_finiteLength** [IsLocalRing R] (g : ℕ) (hX : ∀ i n, n < g → Subsingleton (Abelian.Ext.{0} (ModuleCat.of R (R ⧸ maximalIdeal R)) (C.X i) n)) (hH : ∀ i, IsFiniteLength R (C.homology i)) (i : ℕ) : (∀ n, n + i < g → Subsingleton (Abelian.Ext.{0} (ModuleCat.of R (R ⧸ maximalIdeal R)) (C.opcycles i) n)) ∧ (i < g → IsZero (C.homology i))

theorem **length_ext_opcycles** (Y : ModuleCat.{0} R) (hP : ∀ i, Projective (C.X i)) (j : ℕ) (hH : ∀ i < j + 1, IsZero (C.homology i)) : Module.length R (Abelian.Ext.{0} (C.opcycles (j + 1)) Y (j + 1)) = Module.length R (Abelian.Ext.{0} (C.opcycles 0) Y 0 ⧸ LinearMap.range ((Ext.mk₀ (C.fromOpcycles 0 1)).precompOfLinear R Y (zero_add 0)))

theorem **length_ext_zero_quot** : Module.length R (Abelian.Ext.{0} (C.opcycles 0) (ModuleCat.of R R) 0 ⧸ LinearMap.range ((Ext.mk₀ (C.fromOpcycles 0 1)).precompOfLinear R (ModuleCat.of R R) (zero_add 0))) = Module.length R (Module.Dual R (C.X 0) ⧸ LinearMap.range (C.d 0 1).hom.dualMap)

theorem **homology_of_free_complex_of_isFiniteLength** : ∀ (O : Type) [CommRing O] [IsRegularLocalRing O] (g : ℕ), ringKrullDim O = g → ∀ C : CochainComplex (ModuleCat.{0} O) ℕ, (∀ i, Module.Free O (C.X i) ∧ Module.Finite O (C.X i)) → (∀ i, g < i → IsZero (C.X i)) → (∀ i, IsFiniteLength O (C.homology i)) → Nonempty ((Module.Dual O (C.X 0) ⧸ LinearMap.range (C.d 0 1).hom.dualMap) ≃ₗ[O] IsLocalRing.ResidueField O) → (∀ i, i ≠ g → IsZero (C.homology i)) ∧ Module.length O (C.homology g) = 1

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CechHOneVanishesOnOpenOfIsAffineOpen

theorem **le_iSup_inf** {J : Type} {P : J → X.Opens} {W : X.Opens} (h : W ≤ ⨆ k, P k) : W ≤ ⨆ k, W ⊓ P k

theorem **isCoboundary_of_refine** {I J : Type} {O : I → X.Opens} {φ : ∀ a b, SecOn M (O a ⊓ O b)} (hφ : IsCocycle M O φ) (P : J → X.Opens) (τ : J → I) (hP : ∀ k, P k ≤ O (τ k)) (hc : ∀ a, O a ≤ ⨆ k, P k) (ψ' : ∀ k, SecOn M (P k)) (h : ∀ k l, restrict M (inf_le_inf (hP k) (hP l)) (φ (τ k) (τ l)) = restrict M inf_le_left (ψ' k) - restrict M inf_le_right (ψ' l)) : IsCoboundary M O φ

theorem **h1Zero_iff** (hM : M.IsLB) : FirstCohomologyVanishes M ↔ CechHOneVanishesOnOpen M ⊤

theorem **cechHOneVanishesOnOpen_of_cover** {κ : Type} {D : κ → X.Opens} {V : X.Opens} (hD : ∀ i, D i ≤ V) (hc : V ≤ ⨆ i, D i) (h1 : ∀ i, CechHOneVanishesOnOpen M (D i)) (h2 : ∀ θ : ∀ i j, SecOn M (D i ⊓ D j), IsCocycle M D θ → IsCoboundary M D θ) : CechHOneVanishesOnOpen M V

theorem **exists_extend** (hM : M.IsLB) {V U W : X.Opens} (hU : IsAffineOpen U) (hUV : U ≤ V) (i : M.ι) (hi : U ≤ M.U i) (t : Γ(X, V)) (hW : W = U ⊓ X.basicOpen t) (φ : SecOn M W) : ∃ n₀ : ℕ, ∀ n, n₀ ≤ n → ∃ ρ : SecOn M U, restrict M (hW.le.trans inf_le_left) ρ = rs X ((hW.le.trans inf_le_left).trans hUV) t ^ n • φ

theorem **exists_kill** (hM : M.IsLB) {V U W : X.Opens} (hU : IsAffineOpen U) (hUV : U ≤ V) (i : M.ι) (hi : U ≤ M.U i) (t : Γ(X, V)) (hW : W = U ⊓ X.basicOpen t) (φ : SecOn M U) (h : restrict M (hW.le.trans inf_le_left) φ = 0) : ∃ n₀ : ℕ, ∀ n, n₀ ≤ n → rs X hUV t ^ n • φ = 0

theorem **cechHOneVanishesOnOpen_of_isAffineOpen** (hM : M.IsLB) {V : X.Opens} (hT : IsAffineOpen V) : CechHOneVanishesOnOpen M V

theorem **add_eq_of_ordered** {κ G : Type} [LinearOrder κ] [AddCommGroup G] (F : κ → κ → G) (h0 : ∀ a, F a a = 0) (hs : ∀ a b, F b a = -F a b) (h : ∀ a b c, a < b → b < c → F a b + F b c = F a c) (a b c : κ) : F a b + F b c = F a c

theorem **exists_triple** {κ : Type} [LinearOrder κ] {t : Finset κ} (ht : t.card = 3) : ∃ i j k, i < j ∧ j < k ∧ t = {i, j, k}

theorem **exists_pair** {κ : Type} [LinearOrder κ] {t : Finset κ} (ht : t.card = 2) : ∃ i j, i < j ∧ t = {i, j}

theorem **d_two_apply_triple** {κ : Type} [LinearOrder κ] (A : GlobalTheory.Cube.{0} κ R) (x : GlobalTheory.Cube.Term A 2) {i j k : κ} (h : i < j) (h' : j < k) (hc : ({i, j, k} : Finset κ).card = 2 + 1) : GlobalTheory.Cube.d A 2 x ({i, j, k}, hc) = GlobalTheory.Cube.res A (by simp) (x ({j, k}, Finset.card_pair h'.ne)) - GlobalTheory.Cube.res A (by simp) (x ({i, k}, Finset.card_pair (h.trans h').ne)) + GlobalTheory.Cube.res A (by simp [Finset.insert_subset_iff]) (x ({i, j}, Finset.card_pair h.ne))

theorem **le_single** {W : X.Opens} {i : κ} (hi : W ≤ D i) : W ≤ ⊓ l ∈ ({i} : Finset κ), D l

theorem **le_of_mem** {s : Finset κ} {i : κ} (hi : i ∈ s) : ⊓ l ∈ s, D l ≤ D i

theorem **le_pair** {W : X.Opens} {i j : κ} (hi : W ≤ D i) (hj : W ≤ D j) : W ≤ ⊓ l ∈ ({i, j} : Finset κ), D l

theorem **restrict_d_one** (x : GlobalTheory.Cube.Term (secCube f M D) 1) {i j : κ} (h : i < j) {W : X.Opens} (hi : W ≤ D i) (hj : W ≤ D j) : restrict M (le_pair D hi hj) (comp M D f (GlobalTheory.Cube.d _ 1 x) {i, j} (Finset.card_pair h.ne)) = restrict M (le_single D hj) (comp M D f x {j} (Finset.card_singleton j)) - restrict M (le_single D hi) (comp M D f x {i} (Finset.card_singleton i))

theorem **restrict_d_two** (x : GlobalTheory.Cube.Term (secCube f M D) 2) {i j k : κ} (h : i < j) (h' : j < k) (hc : ({i, j, k} : Finset κ).card = 2 + 1) {W : X.Opens} (hi : W ≤ D i) (hj : W ≤ D j) (hk : W ≤ D k) : restrict M (by simp [hi, hj, hk]) (comp M D f (GlobalTheory.Cube.d _ 2 x) {i, j, k} hc) = restrict M (le_pair D hj hk) (comp M D f x {j, k} (Finset.card_pair h'.ne)) - restrict M (le_pair D hi hk) (comp M D f x {i, k} (Finset.card_pair (h.trans h').ne)) + restrict M (le_pair D hi hj) (comp M D f x {i, j} (Finset.card_pair h.ne))

theorem **IsCocycle.diag** {I : Type} {O : I → X.Opens} {φ : ∀ a b, SecOn M (O a ⊓ O b)} (hφ : IsCocycle M O φ) (a : I) : φ a a = 0

theorem **IsCocycle.swap** {I : Type} {O : I → X.Opens} {φ : ∀ a b, SecOn M (O a ⊓ O b)} (hφ : IsCocycle M O φ) (a b : I) : φ a b = -restrict M (le_inf inf_le_right inf_le_left) (φ b a)

theorem **isCoboundary_of_exactAt** (h : (cechLB f M D).ExactAt 1) (φ : ∀ a b, SecOn M (D a ⊓ D b)) (hφ : IsCocycle M D φ) : IsCoboundary M D φ

theorem **exactAt_of_isCoboundary** (h : ∀ φ : ∀ a b, SecOn M (D a ⊓ D b), IsCocycle M D φ → IsCoboundary M D φ) : (cechLB f M D).ExactAt 1

theorem **toCech_apply** {V : X.Opens} (hD : ∀ i, D i ≤ V) (x : SecOnR f M V) (s : {s : Finset κ // s.card = 0 + 1}) (h : ⊓ i ∈ s.1, D i ≤ V) : toCech M f D hD x s = SecOnR.restrict f M h x

theorem **toCech_injective** {V : X.Opens} (hD : ∀ i, D i ≤ V) (hc : V ≤ ⨆ i, D i) : Function.Injective (toCech M f D hD)

theorem **exact_toCech** {V : X.Opens} (hD : ∀ i, D i ≤ V) (hc : V ≤ ⨆ i, D i) : Function.Exact (toCech M f D hD) ((cechLB f M D).d 0 1).hom

@[simp] theorem **kerEquiv_val** {V : X.Opens} (hD : ∀ i, D i ≤ V) (hc : V ≤ ⨆ i, D i) (x : SecOnR f M V) : (kerEquiv M f hD hc x).1 = toCech M f D hD x

theorem **secKerEquiv_val** (hc : ⨆ i, D i = ⊤) (σ : LB.Sec M) (s : {s : Finset κ // s.card = 0 + 1}) : (secKerEquiv M f hc σ).1 s = SecOnR.of f M _ (GlobalTheory.LB.ofSec σ (⊓ i ∈ s.1, D i))

theorem **secKerEquiv_smul** (hc : ⊔ i, D i = ⊤) (r : R) (σ : LB.Sec M) : secKerEquiv M f hc (cst f ⊤ r • σ) = r • secKerEquiv M f hc σ

theorem **exactAt_one_of_h1ZeroOn** {V : X.Opens} (hD : ∀ i, D i ≤ V) (hc : V ≤ ⊔ i, D i) (h : CechHOneVanishesOnOpen M V) : (cechLB f M D).ExactAt 1

theorem **exactAt_one_of_h1Zero** (hM : M.IsLB) (hc : ⊔ i, D i = ⊤) (h : FirstCohomologyVanishes M) : (cechLB f M D).ExactAt 1

theorem **h1ZeroOn_of_exactAt_one** (hM : M.IsLB) {V : X.Opens} (hD : ∀ i, D i ≤ V) (hc : V ≤ ⊔ i, D i) (hA : ∀ i, IsAffineOpen (D i)) (h : (cechLB f M D).ExactAt 1) : CechHOneVanishesOnOpen M V

theorem **h1Zero_of_exactAt_one** (hM : M.IsLB) (hc : ⊔ i, D i = ⊤) (hA : ∀ i, IsAffineOpen (D i)) (h : (cechLB f M D).ExactAt 1) : FirstCohomologyVanishes M

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CoequifiberedDiagramMap

lemma **app_pullOver** (U : Y.Opens) : f.app U ≫ pullOver f g a ha U = g.app U

lemma **map_pullOver** {U V : Y.Opens} (h : U ≤ V) : X.presheaf.map (homOfLE (f.preimage_mono h)).op ≫ pullOver f g a ha U = pullOver f g a ha V ≫ W.presheaf.map (homOfLE (g.preimage_mono h)).op

lemma **mem_invariants** {U : Y.Opens} {x : Γ(X, f ⁻¹ᵁ U)} : x ∈ invariants f g a ha p hp U ↔ pullOver f g a ha U x = pullOver f g p hp U x

lemma **app_mem_invariants** {U : Y.Opens} (y : Γ(Y, U)) : f.app U y ∈ invariants f g a ha p hp U

lemma **map_mem_invariants** {U V : Y.Opens} (h : U ≤ V) {x : Γ(X, f ⁻¹ᵁ V)} (hx : x ∈ invariants f g a ha p hp V) : X.presheaf.map (homOfLE (f.preimage_mono h)).op x ∈ invariants f g a ha p hp U

lemma **isLocalization_preimage** [QuasiCompact f] [QuasiSeparated f] {U : Y.Opens} (hU : IsAffineOpen U) (r : Γ(Y, U)) : letI

lemma **isLocalization_eqLocus** {B T B' T' : Type} [CommRing B] [CommRing T] [CommRing B'] [CommRing T'] [Algebra B B'] [Algebra T T'] (α β : B →+* T) (α' β' : B' →+* T') (hα : ∀ x, α' (algebraMap B B' x) = algebraMap T T' (α x)) (hβ : ∀ x, β' (algebraMap B B' x) = algebraMap T T' (β x)) (r : α.eqLocus β) [IsLocalization.Away (r : B) B'] [IsLocalization.Away (α r) T'] [Algebra (α.eqLocus β) (α'.eqLocus β')] (h : ∀ x, (algebraMap (α.eqLocus β) (α'.eqLocus β') x : B') = algebraMap B B' x) : IsLocalization.Away r (α'.eqLocus β')

lemma **coequifibered_diagramMap** : ((toOpensFunctor Y).op.whiskerLeft (diagramMap f g a ha p hp)).Coequifibered

lemma **ι_toQuot** (U : Y.affineOpens) : (f ⁻¹ᵁ U.1).ι ≫ toQuot f g a ha p hp = (f ⁻¹ᵁ U.1).toSpecΓ ≫ Spec.map (CommRingCat.ofHom (invariants f g a ha p hp U.1).subtype) ≫ (cover f g a ha p hp).f U

lemma **ι_fromQuot** (U : Y.affineOpens) : (cover f g a ha p hp).f U ≫ fromQuot f g a ha p hp = Spec.map ((diagramMap f g a ha p hp).app (.op U.1)) ≫ U.2.fromSpec

lemma **fromQuot_preimage** (U : Y.affineOpens) : fromQuot f g a ha p hp ⁻¹ᵁ U = ((cover f g a ha p hp).f U).opensRange

lemma **toQuot_fromQuot** : toQuot f g a ha p hp ≫ fromQuot f g a ha p hp = f

lemma **of_charts** (P : MorphismProperty Scheme.{0}) [IsZariskiLocalAtTarget P] {ι : Type} (X' Y' : ι → Scheme.{0}) (u : ∀ i, X' i ⟶ X) (v : ∀ i, Y' i ⟶ Y) (f' : ∀ i, X' i ⟶ Y' i) [∀ i, IsOpenImmersion (v i)] (hv : ∀ y, ∃ i z, v i z = y) (h : ∀ i, IsPullback (f' i) (u i) (v i) f) (hP : ∀ i, P (f' i)) : P f

lemma **isPullback_of_charts** {Z Q : Scheme.{0}} (b : W ⟶ Z) (q : X ⟶ Q) (q' : Z ⟶ Q) {ι : Type} (W' X' : ι → Scheme.{0}) (u : ∀ i, W' i ⟶ W) (v : ∀ i, X' i ⟶ X) (a' : ∀ i, W' i ⟶ X' i) [∀ i, IsOpenImmersion (v i)] (hv : ∀ y, ∃ i z, v i z = y) (h : ∀ i, IsPullback (a' i) (u i) (v i) a) (h' : ∀ i, IsPullback (a' i) (u i ≫ b) (v i ≫ q) q') : IsPullback a b q q'

lemma **SpecMap_pullOver** {U : Y.Opens} (hf : IsAffineOpen (f ⁻¹ᵁ U)) (hg : IsAffineOpen (g ⁻¹ᵁ U)) : Spec.map (pullOver f g a ha U) ≫ hf.fromSpec = hg.fromSpec ≫ a

lemma **isPullback_pullOver** {U : Y.Opens} (hf : IsAffineOpen (f ⁻¹ᵁ U)) (hg : IsAffineOpen (g ⁻¹ᵁ U)) : IsPullback (Spec.map (pullOver f g a ha U)) hg.fromSpec hf.fromSpec a

lemma **fromSpec_toQuot** (U : Y.affineOpens) (hf : IsAffineOpen (f ⁻¹ᵁ U.1)) : hf.fromSpec ≫ toQuot f g a ha p hp = Spec.map (CommRingCat.ofHom (invariants f g a ha p hp U.1).subtype) ≫ (cover f g a ha p hp).f U

lemma **isPullback_toQuot** (U : Y.affineOpens) (hf : IsAffineOpen (f ⁻¹ᵁ U.1)) : IsPullback (Spec.map (CommRingCat.ofHom (invariants f g a ha p hp U.1).subtype)) hf.fromSpec ((cover f g a ha p hp).f U) (toQuot f g a ha p hp)

lemma **isPullback_fromQuot** (U : Y.affineOpens) : IsPullback (Spec.map ((diagramMap f g a ha p hp).app (.op U.1))) ((cover f g a ha p hp).f U) U.2.fromSpec (fromQuot f g a ha p hp)

lemma **exists_fromSpec_eq** [IsAffineHom f] (x : X) : ∃ (U : Y.affineOpens) (z : Spec Γ(X, f ⁻¹ᵁ U.1)), (U.2.preimage f).fromSpec z = x

lemma **isFinite_fromQuot** [IsFinite f] [IsLocallyNoetherian Y] : IsFinite (fromQuot f g a ha p hp)

theorem **exists_quotient** [IsFinite f] [IsAffineHom g] [IsLocallyNoetherian Y] (H : ∀ U : Y.affineOpens, (invariants f g a ha p hp U.1).subtype.FaithfullyFlat ∧ IsPushout (CommRingCat.ofHom (invariants f g a ha p hp U.1).subtype) (CommRingCat.ofHom (invariants f g a ha p hp U.1).subtype) (pullOver f g a ha U.1) (pullOver f g p hp U.1)) : ∃ (Q : Scheme.{0}) (q : X ⟶ Q) (π : Q ⟶ Y), q ≫ π = f ∧ IsFinite π ∧ IsFinite q ∧ Flat q ∧ Surjective q ∧ IsPullback a p q q

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CompAutEqOfSerreBaseChange

theorem **comp_aut_eq_of_serre_baseChange** (hB : PolBaseChangeSt) {S S' : Scheme.{0}} (f : S' ⟶ S) (P : PolAV S) (M : ℕ) (hS' : ∀ (P : PolAV S') (σ : P.A ≅ P.A), IsMonHom σ.hom → (∃ c : LB.Gen (LB.LinComb P.L (K

theorem **serre_of_algClosed** : PolBaseChangeSt → GlobalFnsSt → (∀ (X T Y : Scheme.{0}) (p : X ⟶ T) (ε : T ⟶ X) (g : X ⟶ Y), ε ≫ p = 𝟙 T → UniversallyClosed p → (∀ V : T.Opens, Function.Surjective (p.app V).hom) → ∀ (t : T) (y : Y), (∀ x : X, p x = t → g x = y) → ∃ V : T.Opens, t ∈ V ∧ (p ⁻¹ᵁ V).ι ≫ g = (p ⁻¹ᵁ V).ι ≫ p ≫ ε ≫ g) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : ℕ), 3 ≤ M → ∀ σ : P.A ≅ P.A, IsMonHom σ.hom → (∃ c : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CoordOfHasWeight

theorem **gm_res_pullU** (r : Z ⟶ Y') (h : Y' ⟶ Y) (z : Z) {U : Y.Opens} {V : Y'.Opens} (le : V ≤ h ⁻¹ᵁ U) (hz : r z ∈ V) (a : Γ(Y, U)ˣ) : gm r z V (res le (pullU h U a)) = gm (r ≫ h) z U a

theorem **gm_res_pullU'** (h : Y' ⟶ Y) (z : Y') {U : Y.Opens} {V : Y'.Opens} (le : V ≤ h ⁻¹ᵁ U) (hz : z ∈ V) (a : Γ(Y, U)ˣ) : gm (𝟙 Y') z V (res le (pullU h U a)) = gm h z U a

theorem **genPull_isGen** {A' : LB Y'} {A : LB Y} {h : Y' ⟶ Y} (a : ChartPull A' A h) {t : LB.Gen A} (ht : t.IsGen) : (genPull a t).IsGen

theorem **lbIso_refl** {A : LB Y} (hA : A.IsLB) : LBIso A A

theorem **eSec_fst** (G : Over S) [GrpObj G] : eSec ((Over.pullback f).obj G) ≫ pullback.fst G.hom f = (f : (𝟙_ (Over S')).left ⟶ (𝟙_ (Over S)).left) ≫ eSec G

theorem **exists_lamChart** (B : Over S) [GrpObj B] [IsCommMonObj B] (L : LB B.left) : ∃ U, Nonempty (ChartPull (abelian₂ ((Over.pullback f).obj B) (L.pull (pullback.fst B.hom f))) (abelian₂ B L) U)

theorem **translate_comp** (q : G ⟶ B) (i : splitTorus S X ⟶ G) [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) {Z : Over S} (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) : ((t ≫ i) ∗ g) ≫ q = g ≫ q

theorem **translate_mem** (q : G ⟶ B) (i : splitTorus S X ⟶ G) [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) {Z : Over S} (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (W : B.left.Opens) (z : Z.left) (hz : g.left z ∈ q.left ⁻¹ᵁ W) : ((t ≫ i) ∗ g).left z ∈ q.left ⁻¹ᵁ W

theorem **hasWeight_iff** (i : splitTorus S X ⟶ G) (U : G.left.Opens) (u : Γ(G.left, U)ˣ) (χ : X) (hle : (snd (splitTorus S X) G).left ⁻¹ᵁ U ≤ (lift (fst (splitTorus S X) G ≫ i) (snd (splitTorus S X) G) ≫ μ[G]).left ⁻¹ᵁ U) (hT : (snd (splitTorus S X) G).left ⁻¹ᵁ U ≤ (fst (splitTorus S X) G).left ⁻¹ᵁ ⊤) : HasWeight i U u χ ↔ res hle (pullU (lift (fst (splitTorus S X) G ≫ i) (snd (splitTorus S X) G) ≫ μ[G]).left _ u) = res hT (pullU (fst (splitTorus S X) G).left ⊤ (charUnit S X χ)) ∗ pullU (snd (splitTorus S X) G).left _ u

theorem **coord_of_hasWeight** {K : Type} (b : Module.Basis K ℤ X) (q : G ⟶ B) (i : splitTorus S X ⟶ G) [hq : IsMonHom q] [hi : IsMonHom i] (hf : Flat q.left) (hs : Surjective q.left) (ha : IsAffineHom q.left) (hk : IsPullback i (toUnit _) q η[B]) (W : B.left.Opens) (u : K → Γ(G.left, q.left ⁻¹ᵁ W)ˣ) (hu : ∀ k, HasWeight i _ (u k) (b k)) (V : B.left.Opens) (hV : IsAffineOpen V) (h : V ≤ W) : Function.Bijective (AddMonoidAlgebra.liftNCRingHom (q.left.app V).hom ((Units.coeHom _).comp (AddMonoidHom.toMultiplicativeLeft (b.constr ℤ (fun k => Additive.ofMul (res (Scheme.Hom.preimage_mono q.left h) (u k)))).toAddMonoidHom)) (fun _ _ => Commute.all _ _))

theorem **HasWeight.baseChange** {X : Type} [AddCommGroup X] {G : Over S} [GrpObj G] {i : splitTorus S X ⟶ G} {U : G.left.Opens} {u : Γ(G.left, U)ˣ} {χ : X} (hu : HasWeight i U u χ) (i' : splitTorus S' X ⟶ (Over.pullback f).obj G) (τ : (splitTorus S' X).left ⟶ (splitTorus S X).left) (hτ : τ ≫ (splitTorus S X).hom = (splitTorus S' X).hom ≫ f) (hi : i'.left ≫ pullback.fst G.hom f = τ ≫ i.left) (hc : τ.appTop.hom (charUnit S X χ).val = (charUnit S' X χ).val) {U' : ((Over.pullback f).obj G).left.Opens} (le : U' ≤ pullback.fst G.hom f ⁻¹ᵁ U) (hU' : ∀ (Z : Over S') (t : Z ⟶ splitTorus S' X) (g : Z ⟶ (Over.pullback f).obj G) (z : Z.left), g.left z ∈ U' → ((t ≫ i') ∗ g).left z ∈ U') : HasWeight i' U' (res le (pullU (pullback.fst G.hom f) U u)) χ

theorem **chart_le_preimage** (E : SplitExt S) (j : (extBC f E).ChartIdx) : (extBC f E).chart j ≤ pullback.fst E.B.hom f ⁻¹ᵁ E.chart (fun k => j k PUnit.unit)

theorem **isGt_extBC** (E : SplitExt S) (G : Over S) [GrpObj G] (d : E.GtDatum G) : (extBC f E).IsGt ((Over.pullback f).obj G)

theorem **gtBaseChange** : GtBaseChangeSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CorrOfGenericOfIsSeparated

theorem **corr_of_corr_pullGen** {X Y Z : Scheme.{u}} (L : LB X) {I I' : Type u} [Fintype I] [Fintype I'] (φ : I → (Y ⟶ X)) (ε : I → ℤ) (φ' : I' → (Y ⟶ X)) (ε' : I' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (h : Z ⟶ Y) (hinj : ∀ U, Function.Injective (h.app U).hom) (hc : ∀ a b : Y ⟶ X, h ≫ a = h ≫ b → a = b) (H : LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t h) (LB.pullGen L _ _ t' h)) : LB.Gen.Corr L _ _ _ _ t t'

theorem **corr_of_generic_of_isSeparated** (V : Type u) [CommRing V] (K : Type u) [Field K] [Algebra V K] [IsFractionRing V K] (G : Over (Spec (CommRingCat.of V))) [IsSeparated G.hom] (T : Over (Spec (CommRingCat.of V))) [Flat T.hom] (L : LB G.left) {I I' : Type u} [Fintype I] [Fintype I'] (φ : I → (T.left ⟶ G.left)) (ε : I → ℤ) (φ' : I' → (T.left ⟶ G.left)) (ε' : I' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (H : LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t (pullback.fst T.hom (Spec.map (CommRingCat.ofHom (algebraMap V K))))) (LB.pullGen L _ _ t' (pullback.fst T.hom (Spec.map (CommRingCat.ofHom (algebraMap V K)))))) : LB.Gen.Corr L _ _ _ _ t t'

theorem **inl_ne_inr** (x : X) (hx : x ∉ U) : pushout.inl U.ι U.ι ≠ pushout.inr U.ι U.ι

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CubeConclusionIff

theorem **isGen_of_maps** {A B : LB X} (hA : A.IsLB) {w : LB.Iso A B} (hw : w.IsGen) {a : LB.Gen A} {b : LB.Gen B} (h : LB.Gen.Maps w a b) : b.IsGen

theorem **idIso_isGen** {L : LB X} (hL : L.IsLB) : (idIso L).IsGen

theorem **gm_linCombIso** (w : LB.Iso A B) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (p : K → B.ι) (q : K → A.ι) (r : Z ⟶ Y) (z : Z) (hz : r z ∈ LB.wchart B φ p ⊓ LB.wchart A φ q) : gm r z (LB.wchart B φ p ⊓ LB.wchart A φ q) ((LB.linCombIso w φ ε).s (p, q)) = ∏ a, gw w (r ≫ φ a) z (p a) (q a) ^ ε a

theorem **maps_iff_pm** (w : LB.Iso A B) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (a : LB.Gen (LB.LinComb A φ ε)) (b : LB.Gen (LB.LinComb B φ ε)) : LB.Gen.Maps (LB.linCombIso w φ ε) a b ↔ PM w φ ε a b

theorem **PM.pullGen** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {a : LB.Gen (LB.LinComb A φ ε)} {b : LB.Gen (LB.LinComb B φ ε)} (H : PM w φ ε a b) (h : Z ⟶ Y) : PM w (fun k => h ≫ φ k) ε (LB.pullGen A φ ε a h) (LB.pullGen B φ ε b h)

theorem **PM.mulGen** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {a : LB.Gen (LB.LinComb A φ ε)} {b : LB.Gen (LB.LinComb B φ ε)} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {a' : LB.Gen (LB.LinComb A φ' ε')} {b' : LB.Gen (LB.LinComb B φ' ε')} (H : PM w φ ε a b) (H' : PM w φ' ε' a' b') : PM w (Sum.elim φ φ') (Sum.elim ε ε') (LB.mulGen A φ ε φ' ε' a a') (LB.mulGen B φ ε φ' ε' b b')

theorem **PM.oneGen** (w : LB.Iso A B) : PM (Y

theorem **corr_of_pm** (hA : A.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} {a : LB.Gen (LB.LinComb A φ ε)} {b : LB.Gen (LB.LinComb B φ ε)} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {a' : LB.Gen (LB.LinComb A φ' ε')} {b' : LB.Gen (LB.LinComb B φ' ε')} (H : PM w φ ε a b) (H' : PM w φ' ε' a' b') (hS : LB.SameSum φ ε φ' ε') (hc : LB.Gen.Corr A φ ε φ' ε' a a') : LB.Gen.Corr B φ ε φ' ε' b b'

theorem **idIso_maps** {X Y : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) : LB.Gen.Maps (LB.linCombIso (idIso L) φ ε) t t

theorem **cubical_clause_iff** {G' : Over S'} {G : Over S} [GrpObj G'] [IsCommMonObj G'] [GrpObj G] [IsCommMonObj G] (m : CubeMap G' G) (L : RigLB G) (c : CubicalStr G L) (hL : (L.L.pull m.F).IsLB) (hr : (m.rigPull L.L L.rig).IsGen) : (∃ c' : CubicalStr G' (m.rigLB L hL hr), m.IsPullCubical c' c) ↔ ∃ c' : CubicalStr G' (m.rigLB L hL hr), c'.t = m.thetaPull L.L c.t ∧ m.IsPullCubical c' c

theorem **cube_conclusion_iff** {G' : Over S'} {G : Over S} [GrpObj G'] [IsCommMonObj G'] [GrpObj G] [IsCommMonObj G] (m : CubeMap G' G) (L : RigLB G) (c : CubicalStr G L) : (∃ (hL : (L.L.pull m.F).IsLB) (hr : (m.rigPull L.L L.rig).IsGen), (∃ c' : CubicalStr G' (m.rigLB L hL hr), m.IsPullCubical c' c) ∧ (L.IsSymmetric → (m.rigLB L hL hr).IsSymmetric)) ↔ ∃ (hL : (L.L.pull m.F).IsLB) (hr : (m.rigPull L.L L.rig).IsGen) (c' : CubicalStr G' (m.rigLB L hL hr)), c'.t = m.thetaPull L.L c.t ∧ m.IsPullCubical c' c ∧ (L.IsSymmetric → (m.rigLB L hL hr).IsSymmetric)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CubeTheoremAndSectionsFiniteGrowthAbelianVariety

theorem **cube_gen** : ∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen

theorem **sections_finite** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) ∗ (s k).f i = 0

theorem **sections_growth** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), (∀ j : ℕ, Module.Finite Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K))))) ∧ ∃ N c C : ℕ, 1 ≤ N ∧ 1 ≤ c ∧ ∀ t : ℕ, 1 ≤ t → t ^ P.g ≤ c ∗ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L ((N ∗ t : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K)))) ∧ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L ((N ∗ t : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K)))) ≤ C ∗ t ^ P.g

theorem **cube_and_sections** : (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) ∧ (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) ∗ (s k).f i = 0) ∧ (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), (∀ j : ℕ, Module.Finite Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K))))) ∧ ∃ N c C : ℕ, 1 ≤ N ∧ 1 ≤ c ∧ ∀ t : ℕ, 1 ≤ t → t ^ P.g ≤ c ∗ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L ((N ∗ t : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K)))) ∧ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L ((N ∗ t : ℕ) : ℤ) (ba

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CubeTheoremAndSectionsFiniteGrowthAbelianVarietyStatement

theorem **_root_.GlobalTheory.T.cube_theorem_and_sections_finite_growth_abelianVariety** : (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) ∧ (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) ∗ (s k).f i = 0) ∧ (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), (∀ j : ℕ, Module.Finite Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K))))) ∧ ∃ N c C : ℕ, 1 ≤ N ∧ 1 ≤ c ∧ ∀ t : ℕ, 1 ≤ t → t ^ P.g ≤ c ∗ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L ((N ∗ t : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K)))) ∧ Module.finrank Γ((baseOver (Spec (CommR

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CubeTotalComplexAcyclicOfSlicesInductionOnCard

theorem **comap_congr** {g g' : Finset κ → Finset ι} (hg : Monotone g) (hg' : Monotone g') (h : ∀ s, g s = g' s) (A : Cube.{v} ι R) : comap g hg A = comap g' hg' A

theorem **monotone_union_map** (f : κ ↪o ι) (s : Finset ι) : Monotone fun t : Finset κ => s ∪ t.map f.toEmbedding

theorem **slice_empty** (f : κ ↪o ι) (A : Cube.{v} ι R) : slice f ∅ A = lowerFace f A

theorem **mem_part** (e : κ ↪o ι) (u : Finset ι) (k : κ) : k ∈ part e u ↔ e k ∈ u

theorem **monotone_part** (e : κ ↪o ι) : Monotone (part e)

theorem **subsingleton_supp** (A : Cube.{v} ι R) (hP : P.OrdConnected) {s : Finset ι} (hs : s ∉ P) : Subsingleton ((supp A P hP).obj s)

theorem **suppEquiv_res** {s t : Finset ι} (h : s ⊆ t) (hs : s ∈ P) (ht : t ∈ P) (x : (supp A P hP).obj s) : suppEquiv A hP ht (res (supp A P hP) h x) = res A h (suppEquiv A hP hs x)

theorem **supp_ext** {s : Finset ι} {x y : (supp A P hP).obj s} (h : ∀ hs : s ∈ P, suppEquiv A hP hs x = suppEquiv A hP hs y) : x = y

theorem **supp_congr** (A : Cube.{v} ι R) (h : ∀ s, s ∈ P ↔ s ∈ P') : supp A P hP = supp A P' hP'

theorem **comap_supp** {κ : Type} (g : Finset κ → Finset ι) (hg : Monotone g) (A : Cube.{v} ι R) : comap g hg (supp A P hP) = supp (comap g hg A) (g ⁻¹' P) (hP.preimage_mono hg)

theorem **suppEquiv_suppMapApp** (φ : A ⟶ B) {s : Finset ι} (hs : s ∈ P) (hs' : s ∈ P') (x : (supp A P hP).obj s) : suppEquiv B hP' hs' (suppMapApp φ hP hP' s x) = (φ.app s).hom (suppEquiv A hP hs x)

theorem **suppMap_app** (φ : A ⟶ B) (h) (s : Finset ι) : ((suppMap φ hP hP' h).app s).hom = suppMapApp φ hP hP' s

theorem **suppMapApp_bijective** (φ : A ⟶ B) {s : Finset ι} (h : s ∈ P ↔ s ∈ P') (hφ : s ∈ P → Function.Bijective (φ.app s).hom) : Function.Bijective (suppMapApp φ hP hP' s)

theorem **isIso_of_bijective** (φ : A ⟶ B) (h : ∀ s, Function.Bijective (φ.app s).hom) : IsIso φ

theorem **suppIncl_app** (A : Cube.{v} ι R) (hP : P.OrdConnected) (hU : IsUpperSet P) {s : Finset ι} (hs : s ∈ P) : ((suppIncl A hP hU).app s).hom = (suppEquiv A hP hs).toLinearMap

theorem **suppSequence_shortExact** (A : Cube.{v} ι R) (hP₁ : P₁.OrdConnected) (hP : P.OrdConnected) (hP₂ : P₂.OrdConnected) (h : ∀ s, s ∈ P ↔ s ∈ P₁ ∨ s ∈ P₂) (hd : ∀ s ∈ P₁, s ∉ P₂) (hu : ∀ ⦃s t : Finset ι⦄, s ⊆ t → s ∈ P₁ → t ∈ P → t ∈ P₁) (s : Finset ι) : ((suppSequence A hP₁ hP hP₂ h hd hu).map ((evaluation _ _).obj s)).ShortExact

theorem **cubeComplex_acyclic_of_iso** {A B : Cube.{v} ι R} (φ : A ≅ B) (h : (cubeComplex A).Acyclic) : (cubeComplex B).Acyclic

theorem **isZero_cubeComplex_X_of_subsingleton** (A : Cube.{v} ι R) (p : ℕ) (h : ∀ s : Finset ι, s.card = p → Subsingleton (A.obj s)) : IsZero ((cubeComplex A).X p)

theorem **isZero_cubeComplex_of_subsingleton** (A : Cube.{v} ι R) (h : ∀ s : Finset ι, Subsingleton (A.obj s)) : IsZero (cubeComplex A)

theorem **cubeComplex_acyclic_of_subsingleton** (A : Cube.{v} ι R) (h : ∀ s : Finset ι, Subsingleton (A.obj s)) : (cubeComplex A).Acyclic

theorem **isIso_cubeLowerProj_of_subsingleton** (e : κ ↪o ι) (A : Cube.{v} ι R) (h : ∀ u : Finset ι, (∃ j ∈ u, j ∉ Set.range e) → Subsingleton (A.obj u)) : IsIso (cubeLowerProj e A)

theorem **isIso_cubeUpperIncl_of_subsingleton** {e : κ ↪o ι} {i : ι} (he : Set.range e = {i}ᶜ) (A : Cube.{v} ι R) (h : ∀ u : Finset ι, i ∉ u → Subsingleton (A.obj u)) : IsIso (cubeUpperIncl he A)

theorem **isIso_homologyMap_g_of_acyclic** {S : ShortComplex (CochainComplex (ModuleCat.{v} R) ℕ)} (hS : S.ShortExact) (h : S.X₁.Acyclic) (n : ℕ) : IsIso (homologyMap S.g n)

theorem **cubeTotalComplex_acyclic_of_slices_induction_on_card** (n : ℕ) : ∀ (μ : Type) [LinearOrder μ] (f : κ ↪o μ) (I : Finset μ) (_ : ∀ j, j ∈ I ↔ j ∉ Set.range f) (_ : I.card = n) (A : Cube.{v} μ R) (_ : ∀ s ⊆ I, (cubeComplex (Cube.slice f s A)).Acyclic), (cubeComplex A).Acyclic

theorem **cubeComplex_acyclic_of_slices** [Finite ι] (h : IsCompl (Set.range e) (Set.range f)) (A : Cube.{v} μ R) (hA : ∀ s : Finset ι, (cubeComplex (Cube.slice f (s.map e.toEmbedding) A)).Acyclic) : (cubeComplex A).Acyclic

theorem **part_union_left** (h : Disjoint (Set.range e) (Set.range f)) (s : Finset ι) (t : Finset κ) : part e (s.map e.toEmbedding ∪ t.map f.toEmbedding) = s

theorem **part_union_right** (h : Disjoint (Set.range e) (Set.range f)) (s : Finset ι) (t : Finset κ) : part f (s.map e.toEmbedding ∪ t.map f.toEmbedding) = t

theorem **map_part_union_map_part** (h : Codisjoint (Set.range e) (Set.range f)) (u : Finset μ) : (part e u).map e.toEmbedding ∪ (part f u).map f.toEmbedding = u

theorem **isUpperSet_nonempty** : IsUpperSet {s : Finset ι | s.Nonempty}

theorem **ordConnected_nonempty** : Set.OrdConnected {s : Finset ι | s.Nonempty}

theorem **mem_partsIn** {Q : Set (Finset ι)} {T : Set (Finset κ)} {u : Finset μ} : u ∈ partsIn e f Q T ↔ part e u ∈ Q ∧ part f u ∈ T

theorem **ordConnected_partsIn** {Q : Set (Finset ι)} {T : Set (Finset κ)} (hQ : Q.OrdConnected) (hT : T.OrdConnected) : (partsIn e f Q T).OrdConnected

theorem **supp_partsIn_comm** {Q : Set (Finset ι)} {T : Set (Finset κ)} (hQ : Q.OrdConnected) (hT : T.OrdConnected) (A : Cube.{v} μ R) : supp A (partsIn e f Q T) (ordConnected_partsIn e f hQ hT) = supp A (partsIn f e T Q) (ordConnected_partsIn f e hT hQ)

theorem **truncBoth_comm** (A : Cube.{v} μ R) : truncBoth e f A = truncBoth f e A

theorem **slice_supp_partsIn_of_mem** (h : Disjoint (Set.range e) (Set.range f)) {Q : Set (Finset ι)} {T : Set (Finset κ)} (hQ : Q.OrdConnected) (hT : T.OrdConnected) (A : Cube.{v} μ R) {s : Finset ι} (hs : s ∈ Q) : slice f (s.map e.toEmbedding) (supp A (partsIn e f Q T) (ordConnected_partsIn e f hQ hT)) = supp (slice f (s.map e.toEmbedding) A) T hT

theorem **subsingleton_slice_supp_partsIn** (h : Disjoint (Set.range e) (Set.range f)) {Q : Set (Finset ι)} {T : Set (Finset κ)} (hQ : Q.OrdConnected) (hT : T.OrdConnected) (A : Cube.{v} μ R) {s : Finset ι} (hs : s ∉ Q) (t : Finset κ) : Subsingleton ((slice f (s.map e.toEmbedding) (supp A (partsIn e f Q T) (ordConnected_partsIn e f hQ hT))).obj t)

theorem **cubeComplex_trunc₁_exactAt_zero** (A : Cube.{v} ι R) : (cubeComplex (Cube.trunc₁ A)).ExactAt 0

theorem **cubeComplex_trunc₁_acyclic_iff** (A : Cube.{v} ι R) : (cubeComplex (Cube.trunc₁ A)).Acyclic ↔ (cechComplex A).Acyclic

theorem **cubeComplex_supp_partsIn_acyclic** [Finite ι] (h : IsCompl (Set.range e) (Set.range f)) {Q : Set (Finset ι)} {T : Set (Finset κ)} (hQ : Q.OrdConnected) (hT : T.OrdConnected) (A : Cube.{v} μ R) (hA : ∀ s ∈ Q, (cubeComplex (supp (slice f (s.map e.toEmbedding) A) T hT)).Acyclic) : (cubeComplex (supp A (partsIn e f Q T) (ordConnected_partsIn e f hQ hT))).Acyclic

theorem **mem_range_of_notMem_range** (h : IsCompl (Set.range e) (Set.range f)) {j : μ} (hj : j ∉ Set.range e) : j ∈ Set.range f

theorem **Cube.isUpperSet_nonempty_ne_singleton** (k : κ) : IsUpperSet {t : Finset κ | t.Nonempty ∧ t ≠ {k}}

theorem **inlEmb_apply** (i : ι) : inlEmb ι κ i = toLex (Sum.inl i)

theorem **inrEmb_apply** (k : κ) : inrEmb ι κ k = toLex (Sum.inr k)

theorem **isCompl_range_inlEmb_inrEmb** : IsCompl (Set.range (inlEmb ι κ)) (Set.range (inrEmb ι κ))

theorem **part_inlEmb_sumVertex** (s : Finset ι) (t : Finset κ) : part (inlEmb ι κ) (sumVertex s t) = s

theorem **part_inrEmb_sumVertex** (s : Finset ι) (t : Finset κ) : part (inrEmb ι κ) (sumVertex s t) = t

theorem **sumVertex_part** (u : Finset (ι ⊕ₗ κ)) : sumVertex (part (inlEmb ι κ) u) (part (inrEmb ι κ) u) = u

theorem **inl_mem_sumVertex** {s : Finset ι} {t : Finset κ} {i : ι} : toLex (Sum.inl i) ∈ sumVertex s t ↔ i ∈ s

theorem **inr_mem_sumVertex** {s : Finset ι} {t : Finset κ} {k : κ} : toLex (Sum.inr k) ∈ sumVertex s t ↔ k ∈ t

theorem **sumVertex_eq_map_disjSum** (s : Finset ι) (t : Finset κ) : sumVertex s t = (s.disjSum t).map toLex.toEmbedding

theorem **sumVertex_mono** {s s' : Finset ι} {t t' : Finset κ} (hs : s ⊆ s') (ht : t ⊆ t') : sumVertex s t ⊆ sumVertex s' t'

theorem **sliceSnd_eq_slice** (A : Cube.{v} (ι ⊕ₗ κ) R) (s : Finset ι) : sliceSnd A s = slice (inrEmb ι κ) (s.map (inlEmb ι κ).toEmbedding) A

theorem **sliceFst_eq_slice** (A : Cube.{v} (ι ⊕ₗ κ) R) (t : Finset κ) : sliceFst A t = slice (inlEmb ι κ) (t.map (inrEmb ι κ).toEmbedding) A

theorem **sliceSnd_empty** (A : Cube.{v} (ι ⊕ₗ κ) R) : sliceSnd A ∅ = lowerFace (inrEmb ι κ) A

theorem **sliceFst_empty** (A : Cube.{v} (ι ⊕ₗ κ) R) : sliceFst A ∅ = lowerFace (inlEmb ι κ) A

theorem **isUpperSet_partsIn_nonempty** {μ : Type} [LinearOrder μ] (e : ι ↪o μ) (f : κ ↪o μ) : IsUpperSet (partsIn e f {s | s.Nonempty} {t | t.Nonempty})

theorem **trunc₁Incl_bijective** (A : Cube.{v} ι R) {s : Finset ι} (hs : s.Nonempty) : Function.Bijective ((trunc₁Incl A).app s).hom

theorem **subsingleton_trunc₁_empty** (A : Cube.{v} ι R) : Subsingleton ((trunc₁ A).obj ∅)

theorem **trunc₂Incl_bijective** (A : Cube.{v} (ι ⊕ₗ κ) R) {s : Finset ι} {t : Finset κ} (hs : s.Nonempty) (ht : t.Nonempty) : Function.Bijective ((trunc₂Incl A).app (sumVertex s t)).hom

theorem **subsingleton_trunc₂** (A : Cube.{v} (ι ⊕ₗ κ) R) {s : Finset ι} {t : Finset κ} (h : s = ∅ ∨ t = ∅) : Subsingleton ((trunc₂ A).obj (sumVertex s t))

theorem **Cube.lastTwoEmb_inl** (k : κ) : lastTwoEmb ι κ ν (toLex (Sum.inl k)) = toLex (Sum.inl (toLex (Sum.inr k)))

theorem **Cube.lastTwoEmb_inr** (l : ν) : lastTwoEmb ι κ ν (toLex (Sum.inr l)) = toLex (Sum.inr l)

theorem **Cube.firstEmb_apply** (i : ι) : firstEmb ι κ ν i = toLex (Sum.inl (toLex (Sum.inl i)))

theorem **Cube.isCompl_range_lastTwoEmb_firstEmb** : IsCompl (Set.range (lastTwoEmb ι κ ν)) (Set.range (firstEmb ι κ ν))

theorem **Cube.map_lastTwoEmb_union** (s : Finset ι) (t : Finset κ) (w : Finset ν) : (sumVertex t w).map (lastTwoEmb ι κ ν).toEmbedding ∪ s.map (firstEmb ι κ ν).toEmbedding = sumVertex (sumVertex s t) w

theorem **Cube.map_lastTwoEmb** (t : Finset κ) (w : Finset ν) : (sumVertex t w).map (lastTwoEmb ι κ ν).toEmbedding = sumVertex (sumVertex (∅ : Finset ι) t) w

theorem **Cube.sliceSnd_lowerFace_lastTwoEmb** (A : Cube.{v} ((ι ⊕ₗ κ) ⊕ₗ ν) R) (t : Finset κ) : sliceSnd (lowerFace (lastTwoEmb ι κ ν) A) t = sliceSnd A (sumVertex ∅ t)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CubicalExtendsAndExtendGen

theorem **smooth_tensor** (A B : Over S) [Smooth A.hom] [Smooth B.hom] : Smooth (A ⊗ B).hom

theorem **geometricallyConnected_tensor** (A B : Over S) [Smooth A.hom] [Smooth B.hom] [GeometricallyConnected A.hom] [GeometricallyConnected B.hom] : GeometricallyConnected (A ⊗ B).hom

theorem **eSec_comp** : eSec G ≫ G.hom = 𝟙 S

theorem **eSec_three** : eSec (G3 G) = (lift (1 : 𝟙_ (Over S) ⟶ G) (lift (1 : 𝟙_ (Over S) ⟶ G) (1 : 𝟙_ (Over S) ⟶ G))).left

theorem **formEq_origin** : FormEq (K

theorem **pullGen_genPull** {X Y Z : Scheme.{0}} (M : LB X) (c : LB.Gen M) (f : Y ⟶ X) (h : Z ⟶ Y) : LB.pullGen M _ _ (LB.genPull M c f) h = LB.genPull M c (h ≫ f)

theorem **cubical_extends_of_extension** (h₁ : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f)) (h₂ : (∀ (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] (H X : Scheme.{0}) (p : H ⟶ Spec (CommRingCat.of V)) (e : Spec (CommRingCat.of V) ⟶ H), Smooth p → GeometricallyIntegral p → e ≫ p = 𝟙 _ → ∀ (eK : Spec (CommRingCat.of K) ⟶ pullback p (etaK V K)), eK ≫ pullback.fst p (etaK V K) = etaK V K ≫ e → ∀ (L : LB X), L.IsLB → ∀ (I : Type) [Fintype I] (φ : I → (H ⟶ X)) (ε : I → ℤ) (t : LB.Gen (LB.LinComb L (fun i => pullback.fst p (etaK V K) ≫ φ i) ε)) (ρ : LB.Gen (LB.LinComb L (fun i => e ≫ φ i) ε)), t.IsGen → ρ.IsGen → LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t eK) (LB.pullGen L _ _ ρ (etaK V K)) → ∃! c : LB.Gen (LB.LinComb L φ ε), c.IsGen ∧ LB.pullGen L _ _ c (pullback.fst p (et

theorem **extendGen_of_extension** (h₁ : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f)) (h₂ : (∀ (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] (H X : Scheme.{0}) (p : H ⟶ Spec (CommRingCat.of V)) (e : Spec (CommRingCat.of V) ⟶ H), Smooth p → GeometricallyIntegral p → e ≫ p = 𝟙 _ → ∀ (eK : Spec (CommRingCat.of K) ⟶ pullback p (etaK V K)), eK ≫ pullback.fst p (etaK V K) = etaK V K ≫ e → ∀ (L : LB X), L.IsLB → ∀ (I : Type) [Fintype I] (φ : I → (H ⟶ X)) (ε : I → ℤ) (t : LB.Gen (LB.LinComb L (fun i => pullback.fst p (etaK V K) ≫ φ i) ε)) (ρ : LB.Gen (LB.LinComb L (fun i => e ≫ φ i) ε)), t.IsGen → ρ.IsGen → LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t eK) (LB.pullGen L _ _ ρ (etaK V K)) → ∃! c : LB.Gen (LB.LinComb L φ ε), c.IsGen ∧ LB.pullGen L _ _ c (pullback.fst p (etaK V K

theorem **cubical_extends_and_extendGen** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] (H X : Scheme.{0}) (p : H ⟶ Spec (CommRingCat.of V)) (e : Spec (CommRingCat.of V) ⟶ H), Smooth p → GeometricallyIntegral p → e ≫ p = 𝟙 _ → ∀ (eK : Spec (CommRingCat.of K) ⟶ pullback p (etaK V K)), eK ≫ pullback.fst p (etaK V K) = etaK V K ≫ e → ∀ (L : LB X), L.IsLB → ∀ (I : Type) [Fintype I] (φ : I → (H ⟶ X)) (ε : I → ℤ) (t : LB.Gen (LB.LinComb L (fun i => pullback.fst p (etaK V K) ≫ φ i) ε)) (ρ : LB.Gen (LB.LinComb L (fun i => e ≫ φ i) ε)), t.IsGen → ρ.IsGen → LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t eK) (LB.pullGen L _ _ ρ (etaK V K)) → ∃! c : LB.Gen (LB.LinComb L φ ε), c.IsGen ∧ LB.pullGen L _ _ c (pullback.fst p (etaK V K))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.CubicalOfNormalised

theorem **corr_of_val_section** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) {Y : Over S} (hY : Function.Surjective Y.hom.appTop.hom) (e : 𝟙_ (Over S) ⟶ Y) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y.left ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y.left ⟶ X)} {ε' : K' → ℤ} (hF : FormEq φ' ε' φ ε) {u : LB.Gen (LB.LinComb L φ ε)} {u' : LB.Gen (LB.LinComb L φ' ε')} (hu : u.IsGen) (hu' : u'.IsGen) (h : ∀ y J, Good L y J → val u e.left y J = val u' e.left y J) : LB.Gen.Corr L φ ε φ' ε' u u'

theorem **surj_tensor** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {T : Over S} (hT : Function.Surjective T.hom.appTop.hom) : Function.Surjective (A ⊗ T).hom.appTop.hom

theorem **prod_theta** {U : Type} [CommGroup U] (h : ((G3 A).left ⟶ A.left) → U) : ∏ I : thK.{0}, h (thφ A (x3 A) I) ^ thε I = ∏ I : thK.{0}, h ((lift (x3 A 0) (lift (x3 A 1) (x3 A 2))).left ≫ thφ A (x3 A) I) ^ thε I

theorem **oneGen_isGen** {Y : Scheme.{0}} : (LB.oneGen L (Y

theorem **exists_normalised** (hL : L.IsLB) (t : LB.Gen (Theta A L)) (ht : t.IsGen) : ∃ t' : LB.Gen (Theta A L), t'.IsGen ∧ IsNormalised t'

theorem **IsNormalised.pull** {t : LB.Gen (Theta A L)} (ht : IsNormalised t) {T : Over S} (e : 𝟙_ (Over S) ⟶ T) {a b c : T ⟶ A} (ha : e ≫ a = 1) (hb : e ≫ b = 1) (hc : e ≫ c = 1) {y J} (hJ : Good L y J) : val (LB.pullGen L (thφ A (x3 A)) thε t (lift a (lift b c)).left) e.left y J = 1

theorem **cubical_unique** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {M : RigLB A} (c c' : CubicalStr A M) : LB.Gen.Corr M.L _ _ _ _ c.t c'.t

theorem **cube_of_gen** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → CubeSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.EqCompSectionOfSurjectiveAppTop

theorem **_root_.GlobalTheory.eq_comp_section_of_surjective_appTop** {X T Y : Scheme.{0}} [IsAffine Y] (p : X ⟶ T) (ε : T ⟶ X) (g : X ⟶ Y) (h : ε ≫ p = 𝟙 T) (hs : Function.Surjective p.appTop.hom) : g = p ≫ ε ≫ g

theorem **_root_.GlobalTheory.local_rigidity** : ∀ (X T Y : Scheme.{0}) (p : X ⟶ T) (ε : T ⟶ X) (g : X ⟶ Y), ε ≫ p = 𝟙 T → UniversallyClosed p → (∀ V : T.Opens, Function.Surjective (p.app V).hom) → ∀ (t : T) (y : Y), (∀ x : X, p x = t → g x = y) → ∃ V : T.Opens, t ∈ V ∧ (p ⁻¹ᵁ V).ι ≫ g = (p ⁻¹ᵁ V).ι ≫ p ≫ ε ≫ g

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.EqIdOfFixesTorsion

theorem **isGen_of_germs** {φ : K → (Y ⟶ X)} {ε : K → ℤ} (t : LB.Gen (LB.LinComb L φ ε)) (h : ∀ (y : Y) (j j' : K → L.ι), y ∈ LB.wchart L φ j → y ∈ LB.wchart L φ j' → gm (𝟙 Y) y (LB.wchart L φ j) (t.s j) = trF φ ε (𝟙 Y) y j j' ∗ gm (𝟙 Y) y (LB.wchart L φ j') (t.s j')) : t.IsGen

theorem **gm_twistGen** (φ : K → (Y ⟶ X)) (ε : K → ℤ) {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (j : K × Fin 2 → L.ι) (hj : r z ∈ LB.wchart L (fun p : K × Fin 2 => φ p.1 ≫ ![σ, 𝟙 X] p.2) j) : gm r z (LB.wchart L (fun p : K × Fin 2 => φ p.1 ≫ ![σ, 𝟙 X] p.2) j) ((twistGen c φ ε).s j) = ∏ k, gm r z (LB.wchart L (fun b => φ k ≫ ![σ, 𝟙 X] b) (fun b => j (k, b))) ((LB.pullGen L _ _ c (φ k)).s (fun b => j (k, b))) ^ (-ε k)

theorem **twistGen_isGen** (hc : c.IsGen) (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (twistGen c φ ε).IsGen

theorem **val_twistGen** (φ : K → (Y ⟶ X)) (ε : K → ℤ) {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) {J} (hJ : Good L z J) : val (twistGen c φ ε) r z J = (∏ k, val c (r ≫ φ k) z J ^ ε k)⁻¹

theorem **formEq_twist** (φ' : K → (Y' ⟶ X)) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (h : Y ⟶ Y') (hh : ∀ k, h ≫ φ' k = φ k ≫ σ) : FormEq (Sum.elim (fun k => h ≫ φ' k) (fun p : K × Fin 2 => φ p.1 ≫ ![σ, 𝟙 X] p.2)) (Sum.elim ε (fun p : K × Fin 2 => -ε p.1 ∗ ![1, -1] p.2)) φ ε

theorem **exists_twist_sec** (hL : L.IsLB) (hc : c.IsGen) {φ' : K → (Y' ⟶ X)} (φ : K → (Y ⟶ X)) {ε : K → ℤ} (h : Y ⟶ Y') (hh : ∀ k, h ≫ φ' k = φ k ≫ σ) (s : LB.Sec (LB.LinComb L φ' ε)) : ∃ s' : LB.Sec (LB.LinComb L φ ε), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J), Good

L z J → sval s' r z J = sval s (r ≫ h) z J ∗ ((∏ k, val c (r ≫ φ k) z J ^ ε k)⁻¹).val

theorem **exists_twist_gen** (hL : L.IsLB) (hc : c.IsGen) {φ' : K → (Y' ⟶ X)} (φ : K → (Y ⟶ X)) {ε : K → ℤ} (h : Y ⟶ Y') (hh : ∀ k, h ≫ φ' k = φ k ≫ σ) (t : LB.Gen (LB.LinComb L φ' ε)) (ht : t.IsGen) : ∃ t' : LB.Gen (LB.LinComb L φ ε), t'.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J), Good L z J → val t' r z J = val t (r ≫ h) z J ∗ (∏ k, val c (r ≫ φ k) z J ^ ε k)⁻¹

theorem **exists_gen_pull** (M : LB X) (t : LB.Gen M) (ht : t.IsGen) (f : Y ⟶ X) : ∃ u : LB.Gen (M.pull f), u.IsGen

theorem **isUnit_gs_iff** {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (h : r z ∈ U) (a : Γ(Y, U)) : IsUnit (gs r z U a) ↔ r z ∈ Y.basicOpen a

theorem **isUnit_sval_iff** {φ : K → (Y ⟶ X)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb L φ ε)) {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) {J} (hJ : Good L z J) : IsUnit (sval s r z J) ↔ r z ∈ nonzeroLocus s

theorem **eq_id_of_sval** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {ι' : Type} (s : ι' → LB.Sec (LB.LinComb L φ ε)) (b : Y ⟶ B) (f : Y ⟶ Y) (hf : f ≫ b = b) (hcov : ⊔ i, nonzeroLocus (s i) = ⊤) (haff : ∀ i, IsAffineOpen (nonzeroLocus (s i)) ∧ ∃ r : ι' → Γ(Y, nonzeroLocus (s i)), (∀ j k, (Y.presheaf.map (homOfLE (inf_le_right : nonzeroLocus (s i) ⊓ (LB.LinComb L φ ε).U k ≤ (LB.LinComb L φ ε).U k)).op).hom ((s j).f k) = (Y.presheaf.map (homOfLE (inf_le_left : nonzeroLocus (s i) ⊓ (LB.LinComb L φ ε).U k ≤ nonzeroLocus (s i))).op).hom (r j) ∗ (Y.presheaf.map (homOfLE (inf_le_right : nonzeroLocus (s i) ⊓ (LB.LinComb L φ ε).U k ≤ (LB.LinComb L φ ε).U k)).op).hom ((s i).f k)) ∧ Function.Surjective (MvPolynomial.eval₂Hom (b.appLE ⊤ (nonzeroLocus (s i)) le_top).hom r)) (h : ∀ y : Y, ∃ J, Good L y J ∧ ∃ u, ∀ i, sval (s i) f y J = u ∗ sval (s i) (𝟙 Y) y J) : f = 𝟙 Y

theorem **exists_eigenvector** {K R M : Type} [Field K] [IsAlgClosed K] [CommRing R] (e : K ≃+* R) [AddCommGroup M] [Module R M] (hM : Module.finrank R M ≠ 0) (f : M →ₗ[R] M) : ∃ (a : R) (v : M), v ≠ 0 ∧ f v = a • v

theorem **sval_isActOf** {k : ℤ} {θ : ThetaGrp A L k T} {s s' : LB.Sec (LPow A L k T)} (h : IsActOf A L k θ s s') {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : secVal s' r z J = (thetaVal θ.φ r z J).val ∗ secVal s (r ≫ (translT A θ.pt).left) z J

theorem **val_thetaPow** {k : ℤ} (θ : ThetaGrp A L k T) (n : ℕ) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : val (powGen A L k θ n) r z J = ∏ j : Fin n, thetaVal θ.φ (r ≫ (translT A (θ.pt ^ (j : ℕ))).left) z J

theorem **val_symGen** {k : ℤ} (θ : ThetaGrp A L k T) (nu : LB.Gen (negLB A L k T)) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : val (symGen A L k θ nu) r z J = thetaVal θ.φ (r ≫ (translT A θ.pt⁻¹).left) z J ∗ negVal nu r z J ∗ (thetaVal θ.φ (r ≫ (negT A T).left) z J ∗ (negVal nu (r ≫ (translT A θ.pt⁻¹).left) z J)⁻¹)

theorem **sval_add** {k : ℤ} (v w : SecM A L k T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : secVal (SecM.toSec A (v + w)) r z J = secVal (SecM.toSec A v) r z J + secVal (SecM.toSec A w) r z J

theorem **sval_smul** {k : ℤ} (a : Γ(T.left, ⊤)) (v : SecM A L k T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : secVal (SecM.toSec A (a • v)) r z J = gs (r ≫ (fst T A).left) z ⊤ a ∗ secVal (SecM.toSec A v) r z J

theorem **secM_ext** (hL : L.IsLB) {k : ℤ} {v w : SecM A L k T} (h : ∀ y J, Good L y J → secVal (SecM.toSec A v) (𝟙 _) y J = secVal (SecM.toSec A w) (𝟙 _) y J) : v = w

theorem **exists_twist_map** (hL : L.IsLB) (hc : c.IsGen) (k : ℤ) (T : Over S) : ∃ F : SecM A L k T →ₗ[Γ(T.left, ⊤)] SecM A L k T, ∀ (v) (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → secVal (SecM.toSec A (F v)) r z J = secVal (SecM.toSec A v) (r ≫ (T ◁ σ).left) z J ∗ ((val c (r ≫ (snd T A).left) z J ^ k)⁻¹).val

theorem **whisker_translPt** {x : T ⟶ A} (hx : x ≫ σ = x) : T ◁ σ ≫ translPt A x = translPt A x ≫ σ

theorem **whisker_translT** {x : T ⟶ A} (hx : x ≫ σ = x) : translT A x ≫ T ◁ σ = T ◁ σ ≫ translT A x

theorem **whisker_negT** : negT A T ≫ T ◁ σ = T ◁ σ ≫ negT A T

theorem **translT_translPt** (x y : T ⟶ A) : translT A y ≫ translPt A x = translPt A (x ∗ y)

theorem **exists_twist_symLift** [IsCommMonObj A] (hL : L.IsLB) (hc : c.IsGen) {k : ℤ} {n : ℕ} (θ : ThetaGrp A L k T) (hn : θ.pt ^ n = 1) (hx : θ.pt ≫ σ = θ.pt) (hθ : SymLift A L k n θ.pt θ) : ∃ (t : LB.Gen (thetaLB A L k θ.pt)) (ht : t.IsGen), SymLift A L k n θ.pt (θ.pt, t, ht) ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → thetaVal t r z J = thetaVal θ.φ (r ≫ (T ◁ σ).left) z J ∗ (val c (r ≫ (snd T A).left) z J ^ k ∗ val c (r ≫ (translPt A θ.pt).left) z J ^ (-k))⁻¹

theorem **twist_comm** [IsMonHom σ] (hL : L.IsLB) {k : ℤ} {T : Over S} (F : SecM A L k T → SecM A L k T) (hF : ∀ (v) (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → secVal (SecM.toSec A (F v)) r z J = secVal (SecM.toSec A v) (r ≫ (T ◁ σ).left) z J ∗ ((val c (r ≫ (snd T A).left) z J ^ k)⁻¹).val) (θ : ThetaGrp A L k T) (hx : θ.pt ≫ σ = θ.pt) (hθ : ∀ y J, Good L y J → thetaVal θ.φ (𝟙 _) y J = thetaVal θ.φ (𝟙 _ ≫ (T ◁ σ).left) y J ∗ (val c (𝟙 _ ≫ (snd T A).left) y J ^ k ∗ val c (𝟙 _ ≫ (translPt A θ.pt).left) y J ^ (-k))⁻¹) {v v' w' : SecM A L k T} (a₁ : IsActOf A L k θ (SecM.toSec A v) (SecM.toSec A v')) (a₂ : IsActOf A L k θ (SecM.toSec A (F v)) (SecM.toSec A w')) : F v' = w'

theorem **eq_id_of_twist_smul** (hL : L.IsLB) (hS : IsAffineOpen (⊤ : S.Opens)) {n : ℕ} (hV : VeryAmpleAt A L n) (a : Γ((baseOver S).left, ⊤)) (hF : ∀ (v : SecM A L n (baseOver S)) (Z : Scheme.{0}) (r : Z ⟶ (baseOver S ⊗ A).left) (z : Z) (J), Good L z J → secVal (SecM.toSec A (a • v)) r z J = secVal (SecM.toSec A v) (r ≫ (baseOver S ◁ σ).left) z J ∗ ((val c (r ≫ (snd (baseOver S) A).left) z J ^ (n : ℤ))⁻¹).val) : σ = 𝟙 A

theorem **eq_id_of_fixes_torsion** : GlobalFnsSt → SymLiftSt → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ ρ : ThetaGrp P.A P.L k T → (SecM P.A P.L k T ≃ₗ[Γ(T.left, ⊤)] SecM P.A P.L k T), (∀ θ s s', IsActOf P.A P.L k θ s s' ↔ ρ θ (SecM.ofSec P.A s) = SecM.ofSec P.A s') ∧ (∀ θx θy c, CommIsOf P.A P.L k θx θy c → ∀ v, ρ θx (ρ θy v) = (c : Γ(T.left, ⊤)) • ρ θy (ρ θx v)) ∧ (∀ θ θ', θ.pt = θ'.pt → ∃ c : Γ(T.left, ⊤)ˣ, ∀ v, ρ θ' v = (c : Γ(T.left, ⊤)) • ρ θ v) ∧ ∀ θx θy, ∃ θ, θ.pt = θx.pt ∗ θy.pt ∧ ∀ v, ρ θ v = ρ θx (ρ θy v)) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ (θx θy : ThetaGrp P.A P.L k T) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k θx.pt θy.pt c ↔ CommIsOf P.A P.L k θx θy c) ∧ (∀ x : T ⟶ P.A, x ^ k = 1 → Subsingleton T.left → ∃ θ : ThetaGrp P.A P.L k T, θ.pt = x ∧ (Odd k → SymLift P.A P.L k k x θ))) → IrredSt → GlobalSectionsRankAlgClosed → AmpleSt → KerPowSt →

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.EqOneOfPowEqOneOfCongrOne

theorem **one_add_pow_expand** (x : E) (n : ℕ) : ∃ y : E, (1 + x) ^ n = 1 + n • x + n.choose 2 • x ^ 2 + x ^ 3 ∗ y

theorem **one_add_smul_pow_expand** (c : ℕ) (ψ : E) (n : ℕ) : ∃ z : E, (1 + c • ψ) ^ n = 1 + c • (n • ψ + (n.choose 2 ∗ c) • ψ ^ 2 + (c ^ 2) • z)

theorem **step** (htf : ∀ (n : ℕ) (x : E), n ≠ 0 → n • x = 0 → x = 0) {q c l : ℕ} (hq : q.Prime) (hl : l.Prime) (hc0 : c ≠ 0) (hqc : q ∣ c) (h2 : q = 2 → 4 ∣ c) (ψ : E) (h : (1 + c • ψ) ^ l = 1) : ∃ y : E, ψ = q • y

theorem **eq_zero_of_prime** (htf : ∀ (n : ℕ) (x : E), n ≠ 0 → n • x = 0 → x = 0) {q c l : ℕ} (hq : q.Prime) (hsep : ∀ x : E, (∀ k : ℕ, ∃ y : E, x = q ^ k • y) → x = 0) (hl : l.Prime) (hc0 : c ≠ 0) (hqc : q ∣ c) (h2 : q = 2 → 4 ∣ c) (ψ : E) (h : (1 + c • ψ) ^ l = 1) : ψ = 0

theorem **eq_one_of_pow** (htf : ∀ (n : ℕ) (x : E), n ≠ 0 → n • x = 0 → x = 0) {q c : ℕ} (hq : q.Prime) (hsep : ∀ x : E, (∀ k : ℕ, ∃ y : E, x = q ^ k • y) → x = 0) (hc0 : c ≠ 0) (hqc : q ∣ c) (h2 : q = 2 → 4 ∣ c) (m : ℕ) : m ≠ 0 → ∀ ψ : E, (1 + c • ψ) ^ m = 1 → 1 + c • ψ = 1

theorem **eq_one_of_pow_eq_one_of_congr_one** : ∀ (E : Type) [Ring E] (M : ℕ), 3 ≤ M → (∀ (n : ℕ) (x : E), n ≠ 0 → (n : E) * x = 0 → x = 0) → (∀ (q : ℕ) (x : E), 2 ≤ q → (∀ k : ℕ, ∃ y : E, x = (q : E) ^ k * y) → x = 0) → ∀ (σ φ : E) (m : ℕ), m ≠ 0 → σ = 1 + (M : E) * φ → σ ^ m = 1 → σ = 1

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsAddEquivOfSplitTorusIso

theorem **appTop_comp** {A B C : Over S} (a : A ⟶ B) (b : B ⟶ C) (r : Γ(C.left, ⊤)) : (a ≫ b).left.appTop.hom r = a.left.appTop.hom (b.left.appTop.hom r)

theorem **appTop_id** {A : Over S} (r : Γ(A.left, ⊤)) : (𝟙 A : A ⟶ A).left.appTop.hom r = r

theorem **appTop_mul_charUnit** {Z : Over S} (a b : Z ⟶ splitTorus S X) (χ : X) : (a * b).left.appTop.hom (charUnit S X χ).val = a.left.appTop.hom (charUnit S X χ).val * b.left.appTop.hom (charUnit S X χ).val

theorem **charUnit_add** (x y : X) : (charUnit S X (x + y)).val = (charUnit S X x).val * (charUnit S X y).val

theorem **existsUnique_pullback_charUnit** [ConnectedSpace S] (f : splitTorus S X' ⟶ splitTorus S X) [IsMonHom f] (χ : X) : ∃! x : X', f.left.appTop.hom (charUnit S X χ).val = (charUnit S X' x).val

theorem **charUnit_injective** [ConnectedSpace S] {x y : X} (h : (charUnit S X x).val = (charUnit S X y).val) : x = y

theorem **exists_addMonoidHom** [ConnectedSpace S] (f : splitTorus S X' ⟶ splitTorus S X) [IsMonHom f] : ∃ φ : X →+ X', ∀ χ, f.left.appTop.hom (charUnit S X χ).val = (charUnit S X' (φ χ)).val

theorem **exists_addEquiv_of_splitTorus_iso** : ∀ (S : Scheme.{0}) [ConnectedSpace S] (X X' : Type) [AddCommGroup X] [AddCommGroup X'] (θ : splitTorus S X' ≅ splitTorus S X), IsMonHom θ.hom → ∃ xe : X' ≃+ X, ∀ x : X', θ.hom.left.appTop.hom (charUnit S X (xe x)).val = (charUnit S X' x).val

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsBasisMemOfSpanEqTopOfSemilocal

theorem **exists_forall_notMem** {A V J : Type*} [CommRing A] [IsLocalRing A] [Infinite (IsLocalRing.ResidueField A)] [AddCommGroup V] [Module A V] [Finite J] (p : J → Submodule A V) (hp : ∀ i, p i ≠ ⊤) (hm : ∀ i, ∀ a ∈ IsLocalRing.maximalIdeal A, ∀ x, a • x ∈ p i) : ∃ x : V, ∀ i, x ∉ p i

theorem **red_apply** {B M : Type*} [CommRing B] [AddCommGroup M] [Module B M] {n : ℕ} (e : Module.Basis (Fin n) B M) (I : Ideal B) (x : M) (j : Fin n) : red e I x j = Ideal.Quotient.mk I (e.repr x j)

theorem **red_surjective** {B M : Type*} [CommRing B] [AddCommGroup M] [Module B M] {n : ℕ} (e : Module.Basis (Fin n) B M) (I : Ideal B) : Function.Surjective (red e I)

theorem **exists_linearIndependent** {A B M : Type*} [CommRing A] [IsLocalRing A] [Infinite (IsLocalRing.ResidueField A)] [CommRing B] [Algebra A B] [Finite (MaximalSpectrum B)] [AddCommGroup M] [Module B M] [Module A M] [IsScalarTower A B M] {n : ℕ} (e : Module.Basis (Fin n) B M) (N : Submodule A M) (hm : ∀ m : MaximalSpectrum B, (IsLocalRing.maximalIdeal A).map (algebraMap A B) ≤ m.asIdeal) (hN : Submodule.span B (N : Set M) = ⊤) (k : ℕ) (hk : k ≤ n) : ∃ v : Fin k → M, (∀ i, v i ∈ N) ∧ ∀ m : MaximalSpectrum B, LinearIndependent (B ⧸ m.asIdeal) (fun i => red e m.asIdeal (v i))

theorem **exists_basis_mem_of_span_eq_top_of_semilocal** : ∀ (A B M : Type) [CommRing A] [IsLocalRing A] [Infinite (IsLocalRing.ResidueField A)] [CommRing B] [Algebra A B] [Finite (MaximalSpectrum B)] [AddCommGroup M] [Module B M] [Module A M] [IsScalarTower A B M] (n : ℕ) (_ : Module.Basis (Fin n) B M) (N : Submodule A M), (∀ m : MaximalSpectrum B, (IsLocalRing.maximalIdeal A).map (algebraMap A B) ≤ m.asIdeal) → Submodule.span B (N : Set M) = ⊤ → ∃ b : Module.Basis (Fin n) B M, ∀ i, b i ∈ N

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsCommVOfLe

theorem **exists_ratio** {X Y S₀ : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (p : Y ⟶ S₀) (hp : Function.Surjective p.appTop.hom) {u u' : LB.Gen (LB.LinComb M φ ε)} (hu : u.IsGen) (hu' : u'.IsGen) : ∃ a : Γ(S₀, ⊤), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J), Good M z J → (val u r z J * (val u' r z J)⁻¹).val = gs (r ≫ p) z ⊤ a

theorem **lift_translPt** {T T' : Over S} (x : T ⟶ A) (t : T' ⟶ T) (a : T' ⟶ A) : lift t a ≫ translPt A x = (t ≫ x) * a

theorem **lift_translT** {T T' : Over S} (x : T ⟶ A) (t : T' ⟶ T) (a : T' ⟶ A) : lift t a ≫ translT A x = lift t ((t ≫ x) * a)

theorem **prod_theta** {T T' : Over S} (x : T ⟶ A) (t : T' ⟶ T) (a : T' ⟶ A) {U : Type} [CommGroup U] (h : (T'.left ⟶ A.left) → U) : ∏ i : Fin 2, h ((lift t a).left ≫ ![(snd T A).left, (translPt A x).left] i) ^ (![k, -k] i) = h a.left ^ k * h ((t ≫ x) * a).left ^ (-k)

theorem **prod_theta_id** {T : Over S} (x : T ⟶ A) {U : Type} [CommGroup U] (h : ((T ⊗ A).left ⟶ A.left) → U) : ∏ i : Fin 2, h (![(snd T A).left, (translPt A x).left] i) ^ (![k, -k] i) = h (snd T A).left ^ k * h ((fst T A ≫ x) * snd T A).left ^ (-k)

theorem **val_pull_tv** {T T₁ T' : Over S} (θ : ThetaGrp A L k T) (t : T₁ ⟶ T) (a : T₁ ⟶ A) (ρ : T' ⟶ T₁) {z : T'.left} {J} (hJ : Good L z J) : val (LB.pullGen L (K

theorem **val_const** {T T' : Over S} (V : T.left.Opens) (c : Γ(T.left, ⊤)ˣ) (t : T' ⟶ overRes V) (a : T' ⟶ A) (z : T'.left) (J) : val (constGen A L (unitRes V c)) (lift t a).left z J = gm (t ≫ overResι V).left z ⊤ c

theorem **commIs_iff** (hL : L.IsLB) {T : Over S} (V : T.left.Opens) (θx θy : ThetaGrp A L k (overRes V)) (c : Γ(T.left, ⊤)ˣ) : CommIsOf A L k θx θy (unitRes V c) ↔ CommV θx θy (overResι V) c

theorem **exists_restrict** (hL : L.IsLB) {T T₁ : Over S} (θ : ThetaGrp A L k T) (f : T₁ ⟶ T) : ∃ θ₁ : ThetaGrp A L k T₁, θ₁.pt = f ≫ θ.pt ∧ ∀ (T' : Over S) (t : T' ⟶ T₁) (a : T' ⟶ A) (z : T'.left) (J), Good L z J → tv θ₁ t a z J = tv θ (t ≫ f) a z J

theorem **exists_mul** [IsCommMonObj A] (hL : L.IsLB) {T : Over S} (θ θ' : ThetaGrp A L k T) : ∃ θ₁ : ThetaGrp A L k T, θ₁.pt = θ.pt * θ'.pt ∧ ∀ (T' : Over S) (t : T' ⟶ T) (a : T' ⟶ A) (z : T'.left) (J), Good L z J → tv θ₁ t a z J = tv θ t a z J * tv θ' t ((t ≫ θ.pt) * a) z J

theorem **ratio_const** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} (θ θ' : ThetaGrp A L k T) (h : θ.pt = θ'.pt) {T' : Over S} (t : T' ⟶ T) (a b : T' ⟶ A) {z : T'.left} {J} (hJ : Good L z J) : tv θ t a z J * (tv θ' t a z J)⁻¹ = tv θ t b z J * (tv θ' t b z J)⁻¹

theorem **CommV.congr** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T T₀ : Over S} {θx θy θx' θy' : ThetaGrp A L k T} {g : T ⟶ T₀} {c : Γ(T₀.left, ⊤)ˣ} (h : CommV θx θy g c) (hx : θx'.pt = θx.pt) (hy : θy'.pt = θy.pt) : CommV θx' θy' g c

theorem **CommV.restrict** (hL : L.IsLB) {T T₀ T₁ : Over S} {θx θy : ThetaGrp A L k T} {g : T ⟶ T₀} {c : Γ(T₀.left, ⊤)ˣ} (h : CommV θx θy g c) (f : T₁ ⟶ T) : ∃ θx₁ θy₁ : ThetaGrp A L k T₁, θx₁.pt = f ≫ θx.pt ∧ θy₁.pt = f ≫ θy.pt ∧ CommV θx₁ θy₁ (f ≫ g) c

theorem **CommV.mul** [IsCommMonObj A] (hL : L.IsLB) {T T₀ : Over S} {θx θx' θy : ThetaGrp A L k T} {g : T ⟶ T₀} {c c' : Γ(T₀.left, ⊤)ˣ} (h : CommV θx θy g c) (h' : CommV θx' θy g c') : ∃ θ : ThetaGrp A L k T, θ.pt = θx.pt * θx'.pt ∧ CommV θ θy g (c * c')

theorem **exists_commV_of_le** (hL : L.IsLB) {T : Over S} {x y : T ⟶ A} {c : Γ(T.left, ⊤)ˣ} {V W : T.left.Opens} (hle : W ≤ V) (h₁ : ∃ θx θy : ThetaGrp A L k (overRes V), θx.pt = overResι V ≫ x ∧ θy.pt = overResι V ≫ y) (h₂ : ∀ θx θy : ThetaGrp A L k (overRes V), θx.pt =

overResι V ≫ x → θy.pt = overResι V ≫ y → CommIsOf A L k θx θy (unitRes V c)) : ∃ θx θy : ThetaGrp A L k (overRes W), θx.pt = overResι W ≫ x ∧ θy.pt = overResι W ≫ y ∧ CommV θx θy (overResι W) c

theorem **weilIs_mul_left** : GlobalFnsSt → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (x x' y : T ⟶ P.A) (c c' : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k x' y c' → WeilIs P.A P.L k (x ∗ x') y (c ∗ c')

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsEqPowOfTorsionCompEqOne

theorem **_root_.GlobalTheory.DivisibleHom.cancel_of_flat_surjective** {X Y Z : Over S} (e : X ⟶ Y) [Flat e.left] [Surjective e.left] {f g : Y ⟶ Z} (h : e ≫ f = e ≫ g) : f = g

theorem **_root_.GlobalTheory.DivisibleHom.exists_lift_point** {A A' T : Over S} (m : A' ⟶ A) [Flat m.left] [Surjective m.left] (y : T ⟶ A) : ∃ (T' : Over S) (e : T' ⟶ T) (x : T' ⟶ A'), Flat e.left ∧ Surjective e.left ∧ x ≫ m = e ≫ y

theorem **_root_.GlobalTheory.DivisibleHom.isMonHom_of_mul_points** {G H : Over S} [GrpObj G] [GrpObj H] (e : G ⟶ H) (mul : ∀ (T : Over S) (x y : T ⟶ G), (x ∗ y) ≫ e = (x ≫ e) ∗ (y ≫ e)) : IsMonHom e

theorem **_root_.GlobalTheory.DivisibleHom.comp_mulA** {A T : Over S} [GrpObj A] (x : T ⟶ A) (n : ℕ) : x ≫ _root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A n = x ^ n

theorem **_root_.GlobalTheory.DivisibleHom.mulA_comp** {A B : Over S} [GrpObj A] [GrpObj B] (u : A ⟶ B) [IsMonHom u] (n : ℕ) : _root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A n ≫ u = u ^ n

theorem **_root_.GlobalTheory.DivisibleHom.exists_factor** {A B : Over S} [GrpObj A] [IsCommMonObj A] [GrpObj B] (M : ℕ) [QuasiCompact (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M).left] [Flat (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M).left] [Surjective (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M).left] (u : A ⟶ B) [IsMonHom u] (hu : ∀ (T : Over S) (x : T ⟶ A), x ^ M = 1 → x ≫ u = 1) : ∃ φ : A ⟶ B, _root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M ≫ φ = u

theorem **_root_.GlobalTheory.DivisibleHom.isMonHom_of_factor** {A B : Over S} [GrpObj A] [IsCommMonObj A] [GrpObj B] (M : ℕ) [Flat (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M).left] [Surjective (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M).left] (φ : A ⟶ B) [IsMonHom (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA A M ≫ φ)] : IsMonHom φ

theorem **_root_.GlobalTheory.DivisibleHom.exists_eq_pow_of_torsion_comp_eq_one** : (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : _root_.ReciprocityTheory.ChartRestriction.G.Out.PolAV S) (n : ℕ), 1 ≤ n → IsFinite (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA P.A n).left ∧ Flat (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA P.A n).left ∧ Surjective (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (_root_.ReciprocityTheory.ChartRestriction.G.Out.mulA P.A n).left)) → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : _root_.ReciprocityTheory.ChartRestriction.G.Out.PolAV S) (B : Over S) [GrpObj B] [IsCommMonObj B] (M : ℕ), 1 ≤ M → ∀ u : P.A ⟶ B, IsMonHom u → (∀ (T : Over S) (x : T ⟶ P.A), x ^ M = 1 → x ≫ u = 1) → (∃ φ : P.A ⟶ B, IsMonHom φ ∧ u = φ ^ M) ∧ (u ^ M = 1 → u = 1)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsFiniteGradedModuleLineBundleSections

theorem **exists_finite_grCube_hom_bijective** (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (G : Type) [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) [PolyGradedSMul ι 𝒢] [Decomposition 𝒢] (h : ∀ i : ι, Module.Finite (MvPolynomial ι R) (QuotSMulTop ((MvPolynomial.X i : MvPolynomial ι R) − 1) G)) : ∃ K : Submodule (MvPolynomial ι R) G, K.IsHomogeneous 𝒢 ∧ Module.Finite (MvPolynomial ι R) K ∧ ∀ (t : ℤ) (I : Finset ι), I.Nonempty → Function.Bijective (homLocMap (isGradedMap_subtype 𝒢 K) I t)

theorem **exists_finite_graded_module_secCube** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∃ (N : Type) (_ : AddCommGroup N) (_ : Module R N) (_ : Module (MvPolynomial (Fin (m + 1)) R) N) (_ : IsScalarTower R (MvPolynomial (Fin (m + 1)) R) N) (𝒩 : ℤ → Submodule R N) (_ : PolyGradedSMul (Fin (m + 1)) 𝒩) (_ : Decomposition 𝒩), Module.Finite (MvPolynomial (Fin (m + 1)) R) N ∧ ∀ t : ℤ, ∃ φ : grCube (Fin (m + 1)) 𝒩 t ⟶ secCube f (tw M H t) (fun i => nonzeroLocus (s i)), ∀ I : Finset (Fin (m + 1)), I.Nonempty → Function.Bijective (φ.app I).hom

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsFiniteGradedModuleLineBundleSectionsStatement

theorem **_root_.GlobalTheory.T.exists_finite_gradedModule_lineBundleSections** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∃ (N : Type) (_ : AddCommGroup N) (_ : Module R N) (_ : Module (MvPolynomial (Fin (m + 1)) R) N) (_ : IsScalarTower R (MvPolynomial (Fin (m + 1)) R) N) (𝒩 : ℤ → Submodule R N) (_ : PolyGradedSMul (Fin (m + 1)) 𝒩) (_ : Decomposition 𝒩), Module.Finite (MvPolynomial (Fin (m + 1)) R) N ∧ ∀ t : ℤ, ∃ φ : grCube (Fin (m + 1)) 𝒩 t ⟶ secCube f (tw M H t) (fun i => nonzeroLocus (s i)), ∀ I : Finset (Fin (m + 1)), I.Nonempty → Function.Bijective (φ.app I).hom

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGenOfAssociated

theorem **isIrreducible_fibre** (f : X ⟶ Y) [GeometricallyIrreducible f] (p : Y) : IsIrreducible (f ⁻¹' {p})

theorem **MeetsFibres.inf** {f : X ⟶ Y} [GeometricallyIrreducible f] {D D' : X.Opens} (h : MeetsFibres f D) (h' : MeetsFibres f D') : MeetsFibres f (D ⊓ D')

theorem **isLeftRegular_germ** {R : Type} [CommRing R] [IsNoetherianRing R] [IsLocallyNoetherian X] (f : X ⟶ Spec (CommRingCat.of R)) [Flat f] [GeometricallyIntegral f] {D : X.Opens} (hD : MeetsFibres f D) (V : X.Opens) (a : Γ(X, V)) (ha : D ⊓ V ≤ X.basicOpen a) (x : X) (hx : x ∈ V) : IsLeftRegular (gs (𝟙 X) x V a)

theorem **exists_quotient** (W : X.Opens) (f g : Γ(X, W)) (hg : ∀ x ∈ W, IsLeftRegular (gs (𝟙 X) x W g)) (h : ∀ x ∈ W, gs (𝟙 X) x W g ∣ gs (𝟙 X) x W f) : ∃ q : Γ(X, W), ∀ x ∈ W, gs (𝟙 X) x W f = gs (𝟙 X) x W g ∗ gs (𝟙 X) x W q

theorem **exists_gen_of_associated** {L : LB X} (hL : L.IsLB) (φ : Fin 2 → (Y ⟶ X)) (s : LB.Sec (LB.LinComb L (fun _ : Fin 1 => φ 0) (fun _ => 1))) (s' : LB.Sec (LB.LinComb L (fun _ : Fin 1 => φ 1) (fun _ => 1))) (hreg : ∀ y J, Good L y J → IsLeftRegular (sval s' (𝟙 Y) y J)) (hdvd : ∀ y J, Good L y J → sval s (𝟙 Y) y J ∣ sval s' (𝟙 Y) y J ∧ sval s' (𝟙 Y) y J ∣ sval s (𝟙 Y) y J) : ∃ t : LB.Gen (LB.LinComb L φ ![1, -1]), t.IsGen

theorem **exists_gen_cube_pow** (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) {T : Over S} (a b z : T ⟶ G) (k : ℕ) : ∃ u : LB.Gen (LB.LinComb L (K

theorem **exists_gen_three** (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) {T : Over S} (a b c z : T ⟶ G) (m : ℕ) (hc : a ∗ b ^ m ∗ c = 1) (i : Fin 4 → L.ι) (hi : ∀ k (y : T.left), (![a.left, b.left, c.left, (1 : T ⟶ G).left] k) y ∈ L.U (i k)) : ∃ u : LB.Gen (LB.LinComb L (K

theorem **exists_sec_at** (hL : L.IsLB) {T : Over S} (v : T ⟶ B) (w : T ⟶ G) : ∃ s : LB.Sec (LB.LinComb L (fun _ : Fin 1 => w.left) (fun _ => 1)), ∀ (T' : Over S) (ρ : T' ⟶ T) (z : T'.left) (J), Good L z J → sval s ρ.left z J = sval θ (lift (ρ ≫ v) (ρ ≫ w)).left z J

theorem **exists_sec_three** (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) {T : Over S} (v : T ⟶ B) (a b c : T ⟶ G) (m : ℕ) (hc : a ∗ b ^ m ∗ c = 1) (i : Fin 4 → L.ι) (hi : ∀ k (y : T.left), (![a.left, b.left, c.left, (1 : T ⟶ G).left] k) y ∈ L.U (i k)) : ∃ s : LB.Sec (LB.LinComb L (fun _ : Fin 1 => (snd T G).left) (fun _ => ((m + 2 : ℕ) : ℤ))), ∀ (w : T ⟶ G) (z : T.left) (J), Good L z J → Associated (sval s (ptSecT G w).left z J) (sval θ (lift v (a ∗ w)).left z J ∗ sval θ (lift v (b ∗ w)).left z J ^ m ∗ sval θ (lift v (c ∗ w)).left z J)

theorem **meetsFibres_transl** {D : (B ⊗ G).left.Opens} (hD : MeetsFibres (fst B G).left D) {T : Over S} (v : T ⟶ B) (e : T ⟶ G) (f : G ⟶ G) [Surjective f.left] : MeetsFibres (fst T G).left ((lift (fst T G ≫ v) ((fst T G ≫ e) ∗ (snd T G ≫ f))).left ⁻¹ᵁ D)

theorem **associated_sval_comp** {X Y Z Z' : Scheme.{0}} {M : LB X} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb M φ ε)) (r : Z ⟶ Y) (ρ : Z' ⟶ Z) (z : Z') {J : (Z ⟶ X) → M.ι} {J' : (Z' ⟶ X) → M.ι} (hJ : Good M (ρ z) J) (hJ' : Good M z J') : Associated (sval s (ρ ≫ r) z J') ((ρ.stalkMap z).hom (sval s r (ρ z) J))

theorem **SameGerms.comp** {n : ℕ} {T T' : Over S} {x y : T ⟶ G} (H : SameGerms L n x y) (g : T' ⟶ T) : SameGerms L n (g ≫ x) (g ≫ y)

theorem **SameGerms.symm** {n : ℕ} {T : Over S} {x y : T ⟶ G} (H : SameGerms L n x y) : SameGerms L n y x

theorem **dvd_of_sameGerms** [Flat G.hom] [GeometricallyIntegral G.hom] (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) (hD : MeetsFibres (fst B G).left (nonzeroLocus θ)) (m : ℕ) [Surjective (mulA G m).left] {X : Over S} (v : X ⟶ B) (x y a : X ⟶ G) (H : SameGerms L (m + 2) x y) (q : X.left) (J) (hJ : Good L q J) : sval θ (lift v (a ∗ y)).left q J ∣ sval θ (lift v (a ∗ x)).left q J

theorem **eq_of_sameGerms** [Smooth G.hom] [GeometricallyIntegral G.hom] (hL : L.IsLB) (hK : KTriv G L) (t : LB.Gen (Theta G L)) (ht : t.IsGen) (hD : MeetsFibres (fst B G).left (nonzeroLocus θ)) (m : ℕ) [Surjective (mulA G m).left] (R : Type) [CommRing R] [IsNoetherianRing R] (b : Spec (CommRingCat.of R) ⟶ S) (v : Over.mk b ⟶ B) (x y : Over.mk b ⟶ G) (H : SameGerms L (m + 2) x y) : x = y

theorem **associated_of_isValueAt** {n : ℤ} {T : Over S} (s : LB.Sec (LPow G L n T)) (w : T ⟶ G) (τ : LB.Gen (LB.LinComb L (Y

theorem **exists_isValueAt** (hL : L.IsLB) {n : ℤ} {T : Over S} (s : LB.Sec (LPow G L n T)) (w : T ⟶ G) (τ : LB.Gen (LB.LinComb L (Y

theorem **exists_sec_meetsFibres** (hS : SecBaseChangeSt) {S : Scheme.{0}} [IsLocallyNoetherian S] (P : PolAV S) [GeometricallyIntegral P.A.hom] (p : S) : ∃ V : S.Opens, p ∈ V ∧ ∃ θ : LB.Sec (LB.LinComb P.L (fun _ : Fin 1 => (snd (overRes (T

theorem **eq_of_forall_value_eq** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → SecBaseChangeSt → (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 3 ≤ n → ∀ (T : Over S) [IsLocallyNoetherian T.left] (x y : T ⟶ P.A) (τx : LB.Gen (LB.LinComb P.L (Y

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGenOfLocalOnBase

theorem **exists_gen_of_local_on_base** {X Y S : Scheme.{0}} {L : LB X} (hL : L.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (p : Y ⟶ S) (e : S ⟶ Y) (he : e ≫ p = 𝟙 S) (hp : ∀ V : S.Opens, Function.Surjective (pullback.snd p V.ι).appTop.hom) (h0 : ∀ (U : Type) [CommGroup U] (h : (S ⟶ X) → U), ∏ k, h (e ≫ φ k) ^ ε k = 1) (h : ∀ s : S, ∃ U : S.Opens, s ∈ U ∧ ∃ t : LB.Gen (LB.LinComb L (fun k => (p ⁻¹ᵁ U).ι ≫ φ k) ε), t.IsGen) : ∃ u : LB.Gen (LB.LinComb L φ ε), u.IsGen

theorem **exists_gen_zsmul** {X Y : Scheme.{0}} {L : LB X} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) (k : ℤ) : ∃ u : LB.Gen (LB.LinComb L φ (fun i => k ∗ ε i)), u.IsGen

theorem **exists_gen_neg_of_sym** (hG : GlobalFnsSt) {S : Scheme.{0}} {A : Over S} [GrpObj A] (hA : IsAbelianScheme A) {L : LB A.left} (hL : L.IsLB) (hs : Out.Sym A L) : ∃ u : LB.Gen (LB.LinComb L (K

theorem **exists_gen_negLB** : GlobalFnsSt → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ nu : LB.Gen (negLB P.A P.L k T), nu.IsGen

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGenOfMemChart

theorem **exists_gen_of_mem_chart** {X Y : Scheme.{0}} {L : LB X} (hL : L.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (i : K → L.ι) (hi : ∀ k (y : Y), φ k y ∈ L.U (i k)) : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen

theorem **exists_gen_square** {S : Scheme.{0}} {G : Over S} [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) {T : Over S} (a b c : T ⟶ G) (i : Fin 4 → L.ι) (hi : ∀ k (y : T.left), (![(a ∗ b).left, a.left, b.left, (1 : T ⟶ G).left] k) y ∈ L.U (i k)) : ∃ u : LB.Gen (LB.LinComb L (K

theorem **square_local** {S : Scheme.{0}} {G : Over S} [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) (T : Over S) (x y : T ⟶ G) (p : T.left) : ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb L (K

theorem **square_of_cube** : (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → (∀ (S : Scheme.{0}) (P : PolAV S) (T : Over S) (x y : T ⟶ P.A) (p : T.left), ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGenOfSmallExtension

theorem **gr_res** (f : Y ⟶ X) (y : Y) {U V : X.Opens} (hle : V ≤ U) (h : f y ∈ V) (a : Γ(X, U)) : gr f y V h ((X.presheaf.map (homOfLE hle).op).hom a) = gr f y U (hle h) a

theorem **gr_app** (f : Z ⟶ Y) (g : Y ⟶ X) (z : Z) (U : X.Opens) (h : f z ∈ g ⁻¹ᵁ U) (a : Γ(X, U)) : gr f z (g ⁻¹ᵁ U) h ((g.app U).hom a) = gr (f ≫ g) z U h a

theorem **gr_appLE** (f : Z ⟶ Y) (g : Y ⟶ X) (z : Z) (U : X.Opens) (V : Y.Opens) (e : V ≤ g ⁻¹ᵁ U) (h : f z ∈ V) (a : Γ(X, U)) : gr f z V h ((g.appLE U V e).hom a) = gr (f ≫ g) z U (e h) a

theorem **gr_congr** {f g : Y ⟶ X} (e : f = g) (y : Y) (U : X.Opens) (h : f y ∈ U) (a : Γ(X, U)) : gr f y U h a = gr g y U (e ▸ h) a

theorem **gr_ext** {V : X.Opens} {a b : Γ(X, V)} (h : ∀ x (hx : (𝟙 X : X ⟶ X) x ∈ V), gr (𝟙 X) x V hx a = gr (𝟙 X) x V hx b) : a = b

theorem **isUnit_of_gr** {V : X.Opens} {a : Γ(X, V)} (h : ∀ x (hx : (𝟙 X : X ⟶ X) x ∈ V), IsUnit (gr (𝟙 X) x V hx a)) : IsUnit a

theorem **gr_cst** (f : Z ⟶ X) (p : X ⟶ Spec (CommRingCat.of R)) (z : Z) (V : X.Opens) (h : f z ∈ V) (r : R) : gr f z V h (cst p V r) = cstS (f ≫ p) z r

theorem **cstS_comp** {R' : Type} [CommRing R'] (φ : R →+* R') (p : X ⟶ Spec (CommRingCat.of R')) (x : X) (r : R) : cstS (p ≫ Spec.map (CommRingCat.ofHom φ)) x r = cstS p x (φ r)

theorem **app_surj** (hφ : Function.Surjective φ) (H : IsPullback π f' p (Spec.map (CommRingCat.ofHom φ))) {V : X.Opens} (hV : IsAffineOpen V) : Function.Surjective (π.app V).hom

theorem **app_cst** (H : π ≫ p = f' ≫ Spec.map (CommRingCat.ofHom φ)) (V : X.Opens) (r : R) : (π.app V).hom (cst p V r) = cst f' (π ⁻¹ᵁ V) (φ r)

theorem **mem_of_app_eq_zero** (hφ : Function.Surjective φ) (H : IsPullback π f' p (Spec.map (CommRingCat.ofHom φ))) {V : X.Opens} (hV : IsAffineOpen V) (a : Γ(X, V)) (ha : (π.app V).hom a = 0) : a ∈ (RingHom.ker φ).map (cst p V)

theorem **mem_of_flat** [IsLocalRing R] {ε : R} (hε : ε ≠ 0) {B : Type} [CommRing B] (c : R →+* B) (hf : c.Flat) (a : B) (ha : c ε * a = 0) : a ∈ (IsLocalRing.maximalIdeal R).map c

theorem **cst_flat** (p : X ⟶ Spec (CommRingCat.of R)) [Flat p] {V : X.Opens} (hV : IsAffineOpen V) : (cst p V).Flat

theorem **mul_eq_zero_of_mem** {B : Type*} [CommRing B] (c : R →+* B) {I : Ideal R} {ε : R} (h : ∀ a ∈ I, a * ε = 0) {b : B} (hb : b ∈ I.map c) : c ε * b = 0

theorem **gr_cocycle** {M : LB X} (hM : M.IsLB) (f : Z ⟶ X) (z : Z) {i j k : M.ι} (hi : f z ∈ M.U i) (hj : f z ∈ M.U j) (hk : f z ∈ M.U k) : gr f z (M.U i ⊓ M.U j) ⟨hi, hj⟩ (M.g i j).val * gr f z (M.U j ⊓ M.U k) ⟨hj, hk⟩ (M.g j k).val = gr f z (M.U i ⊓ M.U k) ⟨hi, hk⟩ (M.g i k).val

theorem **pull_le** (M : LB X) (F : Y ⟶ X) (i : M.ι) {V : X.Opens} (h : V ≤ M.U i) : F ⁻¹ᵁ V ≤ (M.pull F).U (fun _ => i)

theorem **gr_gen** {M : LB X} {F : Y ⟶ X} {t : LB.Gen (M.pull F)} (ht : t.IsGen) (r : Z ⟶ Y) (z : Z) {i j : M.ι} (hi : F (r z) ∈ M.U i) (hj : F (r z) ∈ M.U j) : gr r z _ (pull_le M F i le_rfl hi) (t.s fun _ => i).val = gr (r ≫ F) z (M.U i ⊓ M.U j) ⟨hi, hj⟩ (M.g i j).val * gr r z _ (pull_le M F j le_rfl hj) (t.s fun _ => j).val

theorem **gen_of_refine** {M : LB X} (hM : M.IsLB) {I : Type} (V : I → X.Opens) (hV : ⊔ a, V a = ⊤) (j : I → M.ι) (hj : ∀ a, V a ≤ M.U (j a)) (w : ∀ a, Γ(X, V a)) (hw : ∀ a, IsUnit (w a)) (H : ∀ a b x (ha : (𝟙 X : X ⟶ X) x ∈ V a) (hb : (𝟙 X : X ⟶ X) x ∈ V b), gr (𝟙 X) x _ ha (w a) = gr (𝟙 X) x (M.U (j a) ⊓ M.U (j b)) ⟨hj a ha, hj b hb⟩ (M.g (j a) (j b)).val * gr (𝟙 X) x _ hb (w b)) : ∃ t : LB.Gen M, t.IsGen

theorem **isUnit_one_add** {B : Type*} [CommRing B] {e : B} (he : e * e = 0) (b : B) : IsUnit (1 + e * b)

theorem **exists_eq_mul** (H : IsPullback π f' p (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {ε}))))) {V : X.Opens} (hV : IsAffineOpen V) (a : Γ(X, V)) (ha : (π.app V).hom a = 0) : ∃ b, a = cst p V ε * b

theorem **isUnit_of_app** (hε : ε * ε = 0) (H : IsPullback π f' p (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {ε}))))) {V : X.Opens} (hV : IsAffineOpen V) (a : Γ(X, V)) (ha : IsUnit ((π.app V).hom a)) : IsUnit a

theorem **exists_lift** {ε : R} (hε : ε * ε = 0) {p : X ⟶ Spec (CommRingCat.of R)} [X.IsSeparated] {π : Y ⟶ X} {f' : Y ⟶ Spec (CommRingCat.of (R ⧸ Ideal.span {ε}))} (H : IsPullback π f' p (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {ε}))))) {M : LB X} (hM : M.IsLB) (t : LB.Gen (M.pull π)) (ht : t.IsGen) : ∃ (I : Type) (V : I → X.Opens) (j : I → M.ι) (hj : ∀ a, V a ≤ M.U (j a)) (u : ∀ a, Γ(X, V a)) (α : ∀ a b, Γ(X, V a ⊓ V b)), ⊔ a, V a = ⊤ ∧ (∀ a, IsAffineOpen (V a)) ∧ (∀ a, IsUnit (u a)) ∧ Rel p ε M V j hj u α

theorem **Rel.cocycle** (H : Rel p ε M V j hj u α) (hε : ε * ε = 0) (hM : M.IsLB) (hu : ∀ a, IsUnit (u a)) (a b d : I) (f : Z ⟶ X) (z : Z) (ha : f z ∈ V a) (hb : f z ∈ V b) (hd : f z ∈ V d) : cstS (f ≫ p) z ε * (gr f z (V a ⊓ V b) ⟨ha, hb⟩ (α a b) + gr f z (V b ⊓ V d) ⟨hb, hd⟩ (α b d) - gr f z (V a ⊓ V d) ⟨ha, hd⟩ (α a d)) = 0

theorem **Rel.gen** (H : Rel p ε M V j hj u α) (hε : ε * ε = 0) (hM : M.IsLB) (hV : ⊔ a, V a = ⊤) (hu : ∀ a, IsUnit (u a)) (β : ∀ a, Γ(X, V a)) (hβ : ∀ a b x (ha : (𝟙 X : X ⟶ X) x ∈ V a) (hb : (𝟙 X : X ⟶ X) x ∈ V b), cstS (𝟙 X ≫ p) x ε * (gr (𝟙 X) x (V a ⊓ V b) ⟨ha, hb⟩ (α a b) - (gr (𝟙 X) x _ ha (β a) - gr (𝟙 X) x _ hb (β b))) = 0) : ∃ t : LB.Gen M, t.IsGen

theorem **Rel.slice** (H : Rel p ε M V j hj u α) (hε : ε * ε = 0) (hV : ⊔ a, V a = ⊤) (hA : ∀ a, IsAffineOpen (V a)) {A A' : Scheme.{0}} (s : A ⟶ X) [IsAffineHom s] {π : A' ⟶ A} {f' : A' ⟶ Spec (CommRingCat.of (R ⧸ Ideal.span {ε}))} (P : IsPullback π f' (s ≫ p) (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {ε}))))) (hf : Function.Surjective f'.appTop.hom) (hu : ∀ a, IsUnit (u a)) (t : LB.Gen (M.pull s)) (ht : t.IsGen) : ∃ β : ∀ a, Γ(A, s ⁻¹ᵁ V a), ∀ a b x (ha : s ((𝟙 A : A ⟶ A) x) ∈ V a) (hb : s ((𝟙 A : A ⟶ A) x) ∈ V b), cstS ((𝟙 A ≫ s) ≫ p) x ε * (gr (𝟙 A ≫ s) x (V a ⊓ V b) ⟨ha, hb⟩ (α a b) - (gr (𝟙 A) x (s ⁻¹ᵁ V a) ha (β a) - gr (𝟙 A) x (s ⁻¹ᵁ V b) hb (β b))) = 0

theorem **app_eq_zero_of_mul** (hε : ε ≠ 0) [Flat p] (w : i ≫ p = f ≫ Spec.map (CommRingCat.ofHom (IsLocalRing.residue R))) {V : X.Opens} (hV : IsAffineOpen V) (a : Γ(X, V)) (ha : cst p V ε * a = 0) : (i.app V).hom a = 0

theorem **mul_eq_zero_of_app** (hm : ∀ a ∈ IsLocalRing.maximalIdeal R, a * ε = 0) (P : IsPullback i f p (Spec.map (CommRingCat.ofHom (IsLocalRing.residue R)))) {V : X.Opens} (hV : IsAffineOpen V) (a : Γ(X, V)) (ha : (i.app V).hom a = 0) : cst p V ε * a = 0

theorem **Rel.cocycle_fibre** (H : Rel p ε M V j hj u α) (hε : ε * ε = 0) (h0 : ε ≠ 0) [Flat p] [X.IsSeparated] (hM : M.IsLB) (hA : ∀ a, IsAffineOpen (V a)) (hu : ∀ a, IsUnit (u a)) (w : i ≫ p = f ≫ Spec.map (CommRingCat.ofHom (IsLocalRing.residue R))) (a b d : I) (y : Y) (ha : (𝟙 Y ≫ i) y ∈ V a) (hb : (𝟙 Y ≫ i) y ∈ V b) (hd : (𝟙 Y ≫ i) y ∈ V d) : gr (𝟙 Y ≫ i) y (V a ⊓ V b) ⟨ha, hb⟩ (α a b) + gr (𝟙 Y ≫ i) y (V b ⊓ V d) ⟨hb, hd⟩ (α b d) = gr (𝟙 Y ≫ i) y (V a ⊓ V d) ⟨ha, hd⟩ (α a d)

theorem **Rel.slice_fibre** (H : Rel p ε M V j hj u α) (hε : ε * ε = 0) (h0 : ε ≠ 0) [X.IsSeparated] (hV : ⊔ a, V a = ⊤) (hA : ∀ a, IsAffineOpen (V a)) {A A' Ak : Scheme.{0}} (s : A ⟶ X) [IsAffineHom s] [Flat (s ≫ p)] {π : A' ⟶ A} {f' : A' ⟶ Spec (CommRingCat.of (R ⧸ Ideal.span {ε}))} (P : IsPullback π f' (s ≫ p) (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {ε}))))) (hf : Function.Surjective f'.appTop.hom) (hu : ∀ a, IsUnit (u a)) (t : LB.Gen (M.pull s)) (ht : t.IsGen) {iA : Ak ⟶ A} {fk : Ak ⟶ Spec (CommRingCat.of (IsLocalRing.ResidueField R))} (w : iA ≫ s ≫ p = fk ≫ Spec.map (CommRingCat.ofHom (IsLocalRing.residue R))) (sk : Ak ⟶ Y) (e : sk ≫ i = iA ≫ s) : ∃ ψ : ∀ a, Γ(Ak, sk ⁻¹ᵁ i ⁻¹ᵁ V a), ∀ a b, (sk.app (i ⁻¹ᵁ V a ⊓ i ⁻¹ᵁ V b)).hom ((i.appLE (V a ⊓ V b) (i ⁻¹ᵁ V a ⊓ i ⁻¹ᵁ V b) (fun _ h => h)).hom (α a b)) = (Ak.presheaf.map (homOfLE (show sk ⁻¹ᵁ (i ⁻¹ᵁ V a ⊓ i ⁻¹ᵁ V b) ≤ s

theorem **Rel.gen_fibre** (H : Rel p ε M V j hj u α) (hε : ε * ε = 0) (hm : ∀ a ∈ IsLocalRing.maximalIdeal R, a * ε = 0) [X.IsSeparated] (hM : M.IsLB) (hV : ⊔ a, V a = ⊤) (hA : ∀ a, IsAffineOpen (V a)) (hu : ∀ a, IsUnit (u a)) (P : IsPullback i f p (Spec.map (CommRingCat.ofHom (IsLocalRing.residue R)))) (ψ : ∀ a, Γ(Y, i ⁻¹ᵁ V a)) (hψ : ∀ a b, (i.appLE (V a ⊓ V b) (i ⁻¹ᵁ V a ⊓ i ⁻¹ᵁ V b) (fun _ h => h)).hom (α a b) = (Y.presheaf.map (homOfLE inf_le_left).op).hom (ψ a) - (Y.presheaf.map (homOfLE inf_le_right).op).hom (ψ b)) : ∃ t : LB.Gen M, t.IsGen

theorem **exists_gen_of_small_extension** : GlobalFnsSt → (∀ (K : Type) [Field K] (X Y : Over (Spec (CommRingCat.of K))) (x₀ : 𝟙_ _ ⟶ X) (y₀ : 𝟙_ _ ⟶ Y), IsSeparated X.hom → QuasiCompact X.hom → IsSeparated Y.hom → QuasiCompact Y.hom → Function.Bijective X.hom.appTop.hom → Function.Bijective Y.hom.appTop.hom → ∀ (I : Type) (V : I → (X ⊗ Y).left.Opens), ⨆ a, V a = ⊤ → ∀ φ : ∀ a b, Γ((X ⊗ Y).left, V a ⊓ V b), (∀ a b d, ((X ⊗ Y).left.presheaf.map (homOfLE (inf_le_left : V a ⊓ V b ⊓ V d ≤ V a ⊓ V b)).op).hom (φ a b) + ((X ⊗ Y).left.presheaf.map (homOfLE (le_inf (inf_le_left.trans inf_le_right) inf_le_right : V a ⊓ V b ⊓ V d ≤ V b ⊓ V d)).op).hom (φ b d) = ((X ⊗ Y).left.presheaf.map (homOfLE (le_inf (inf_le_left.trans inf_le_left) inf_le_right : V a ⊓ V b ⊓ V d ≤ V a ⊓ V d)).op).hom (φ a d)) → (∃ ψ : ∀ a, Γ(X.left, (lift (𝟙 X) (toUnit X ≫ y₀)).left ⁻¹ᵁ V a), ∀ a b, ((lift (𝟙 X) (toUnit X ≫ y₀)).l

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGenTranslEqPairing

theorem **interchange** (p q r s x : A) : θ (p * q) (r * s) x * θ p q x * θ r s x = θ (p * r) (q * s) x * θ p r x * θ q s x

theorem **cubN_mul** (k : ℕ) (a b x : A) : cubN θ k 1 (a * b) x * θ a b x ^ k = θ (a ^ k) (b ^ k) x * cubN θ k 1 a x * cubN θ k 1 b x

theorem **cubN_transl** (k : ℕ) (y w x : A) (hy : y ^ k = 1) : cubN θ k 1 (y * w) x * cubN θ k 1 y w = cubN θ k 1 w x * cubN θ k 1 y (x * w)

theorem **exists_frame** {X Y : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (y₀ : Y) (hY : ∀ y : Y, y = y₀) : ∃ t : LB.Gen (LB.LinComb M φ ε), t.IsGen

theorem **exists_theta** {T : Over S} (p₀ : T.left) (hT : ∀ p : T.left, p = p₀) (n : ℕ) (y : T ⟶ A) (hy : y ^ n = 1) : ∃ θ : ThetaGrp A L.L n T, θ.pt = y ∧ ∀ (Z : Over S) (t : Z ⟶ T) (z : Z.left) (J), Good L.L z J → ∃ s, ∀ a : Z ⟶ A, thetaVal θ t a z J = cubN (cubV c z J) n 1 (t ≫ y) a * s

theorem **exists_triv** {T : Over S} (p₀ : T.left) (hT : ∀ p : T.left, p = p₀) (n : ℕ) (θ : ThetaGrp A L.L n T) : ∃ u : LB.Gen (LB.LinComb L.L (K

theorem **exists_gen_transl_eq_pairing** : GlobalFnsSt → CubeSt → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ (θx θy : ThetaGrp P.A P.L k T) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k θx.pt θy.pt c ↔ CommIsOf P.A P.L k θx θy c) ∧ (∀ x : T ⟶ P.A, x ^ k = 1 → Subsingleton T.left → ∃ θ : ThetaGrp P.A P.L k T, θ.pt = x ∧ (Odd k → SymLift P.A P.L k k x θ))) → WeilSt → ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ) (x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A), x ^ n = 1 → ∃ (u : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGrpObjSplitExtensionOfGenerators

theorem **exists_grpObj** {C : Type*} [Category C] [CartesianMonoidalCategory C] [BraidedCategory C] {G B : C} [GrpObj B] [IsCommMonObj B] (q : G ⟶ B) (P : C → Type) [∀ W, CommGroup (P W)] (n : ∀ {W : C}, (W ⟶ G) → P W) (v : ∀ {W : C}, (W ⟶ B) → (W ⟶ B) → P W) (ext : ∀ {W : C} (x y : W ⟶ G), x ≫ q = y ≫ q → n x = n y → x = y) (v₁ : ∀ {W : C} (x : W ⟶ B), v x 1 = 1) (v₂ : ∀ {W : C} (x y : W ⟶ B), v x y = v y x) (v₃ : ∀ {W : C} (x y w : W ⟶ B), v (x * y) w * v x y = v x (y * w) * v y w) (m : G ⊗ G ⟶ G) (m₁ : ∀ {W : C} (x y : W ⟶ G), (lift x y ≫ m) ≫ q = (x ≫ q) * (y ≫ q)) (m₂ : ∀ {W : C} (x y : W ⟶ G), n (lift x y ≫ m) = n x * n y * (v (x ≫ q) (y ≫ q))⁻¹) (e : 𝟙_ C ⟶ G) (e₁ : ∀ W : C, (toUnit W ≫ e) ≫ q = 1) (e₂ : ∀ W : C, n (toUnit W ≫ e) = 1) (j : G ⟶ G) (j₁ : ∀ {W : C} (x : W ⟶ G), (x ≫ j) ≫ q = (x ≫ q)⁻¹) (j₂ : ∀ {W : C} (x : W ⟶ G), n (x ≫ j) = (n x)⁻¹ * v (x ≫ q)⁻¹ (x ≫ q)) : ∃ _ : GrpOb

theorem **Rep.mul** {σ σ' : Form Y X} {F F' : Vals L Y} (h : Rep L σ F) (h' : Rep L σ' F') : Rep L (fun V _ k => σ V k * σ' V k) (fun Z r z J => F Z r z J * F' Z r z J)

theorem **Rep.inv** {σ : Form Y X} {F : Vals L Y} (h : Rep L σ F) : Rep L (fun V _ k => (σ V k)⁻¹) (fun Z r z J => (F Z r z J)⁻¹)

theorem **Rep.pull** {σ : Form Y X} {F : Vals L Y} (h : Rep L σ F) (f : Y' ⟶ Y) : Rep L (fun V _ k => σ V fun a => k (f ≫ a)) (fun Z r z J => F Z (r ≫ f) z J)

theorem **Rep.congr** {σ σ' : Form Y X} {F : Vals L Y} (h : Rep L σ F) (e : ∀ (V : Type) [CommGroup V] k, σ V k = σ' V k) : Rep L σ' F

theorem **mem_empty** (j : PEmpty → L.ι) (y : Y) : y ∈ LB.wchart L (Y

theorem **Rep.unit** (a : Γ(Y, ⊤)ˣ) : Rep L (fun _ _ _ => 1) (fun _ r z _ => gm r z ⊤ a)

theorem **Rep.exists_unit** (hL : L.IsLB) {F : Vals L Y} (h : Rep L (fun _ _ _ => 1) F) : ∃ a : Γ(Y, ⊤)ˣ, ∀ Z r z J, Good L z J → gm r z ⊤ a = F Z r z J

theorem **compat_iff** {b : Y ⟶ X} (s : ∀ i : L.ι, Γ(Y, b ⁻¹ᵁ L.U i)ˣ) : GenScheme.Compat L b s ↔ ∀ Z (r : Z ⟶ Y) (z : Z) i i', b (r z) ∈ L.U i → b (r z) ∈ L.U i' → gm r z (b ⁻¹ᵁ L.U i') (s i') = gm (r ≫ b) z _ (L.g i i') * gm r z (b ⁻¹ᵁ L.U i) (s i)

theorem **Rep.family** {b : Y ⟶ X} {s : ∀ i : L.ι, Γ(Y, b ⁻¹ᵁ L.U i)ˣ} (hs : GenScheme.Compat L b s) : Rep L (fun _ _ k => (k b)⁻¹) (fun _ r z J => gm r z (b ⁻¹ᵁ L.U (J (r ≫ b))) (s _))

theorem **Rep.exists_family** (hL : L.IsLB) {b : Y ⟶ X} {F : Vals L Y} (h : Rep L (fun _ _ k => (k b)⁻¹) F) : ∃ s : ∀ i : L.ι, Γ(Y, b ⁻¹ᵁ L.U i)ˣ, GenScheme.Compat L b s ∧ ∀ Z r z J, Good L z J → gm r z (b ⁻¹ᵁ L.U (J (r ≫ b))) (s _) = F Z r z J

theorem **rep_rv** (k : κ) (Y : Over S) : Rep (O k).L (fun _ _ h => h (1 : Y ⟶ B).left) (fun Z r z J => rv O k Z (r ≫ Y.hom) z J)

theorem **rep_nv** (hu : ∀ k, GenScheme.Compat (O k).L q.left (u k)) (k : κ) {Y : Over S} (x : Y ⟶ G) : Rep (O k).L (fun _ _ h => (h (x ≫ q).left)⁻¹ * h (1 : Y ⟶ B).left) (fun Z r z J => cv O q u k Z (r ≫ x.left) z J * rv O k Z (r ≫ Y.hom) z J)

theorem **nv_comp** {Y W : Over S} (r : W ⟶ Y) (x : Y ⟶ G) (k z J) : cv O q u k _ (r.left ≫ x.left) z J.1 * rv O k _ (r.left ≫ Y.hom) z J.1 = nv O q u (r ≫ x) k z J

theorem **gm_appLE** {X Y Z : Scheme.{0}} (r : Z ⟶ Y) (g : Y ⟶ X) (z : Z) (U : X.Opens) (V : Y.Opens) (h : V ≤ g ⁻¹ᵁ U) (hz : r z ∈ V) (a : Γ(X, U)ˣ) : gm r z V (GenScheme.pb g U V h a) = gm (r ≫ g) z U a

theorem **exists_hom** {X G Y : Scheme.{0}} {N : κ → LB X} (hN : ∀ k, (N k).IsLB) {q : G ⟶ X} {u : ∀ k (i : (N k).ι), Γ(G, q ⁻¹ᵁ (N k).U i)ˣ} (hU : GenScheme.Univ N q u) (b : Y ⟶ X) (F : ∀ k, Vals (N k) Y) (h : ∀ k, Rep (N k) (fun _ _ h => (h b)⁻¹) (F k)) : ∃ g : Y ⟶ G, g ≫ q = b ∧ ∀ k Z (r : Z ⟶ Y) z J, Good (N k) z J → gm (r ≫ g) z (q ⁻¹ᵁ (N k).U (J ((r ≫ g) ≫ q))) (u k _) = F k Z r z J

theorem **exists_pt** (hU : GenScheme.Univ (fun k => (O k).L) q.left u) {Y : Over S} (b : Y ⟶ B) (F : ∀ k, Vals (O k).L Y.left) (h : ∀ k, Rep (O k).L (fun _ _ h => (h b.left)⁻¹ * h (1 : Y ⟶ B).left) (F k)) : ∃ g : Y ⟶ G, g ≫ q = b ∧ ∀ (W : Over S) (r : W ⟶ Y) k z J, nv O q u (r ≫ g) k z J = F k _ r.left z J.1

theorem **exists_good_at** {X Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) (z : Z) (b : Z ⟶ X) (i : L.ι) (hi : b z ∈ L.U i) : ∃ J, Good L z J ∧ J b = i

theorem **pt_ext** (hu : ∀ k, GenScheme.Compat (O k).L q.left (u k)) (hU : GenScheme.Univ (fun k => (O k).L) q.left u) {W : Over S} (x y : W ⟶ G) (h₁ : x ≫ q = y ≫ q) (h₂ : nv O q u x = nv O q u y) : x = y

theorem **exists_unit_pt** (hU : GenScheme.Univ (fun k => (O k).L) q.left u) {Y : Over S} (a : κ → Γ(Y.left, ⊤)ˣ) : ∃ g : Y ⟶ G, g ≫ q = 1 ∧ ∀ (W : Over S) (r : W ⟶ Y), nv O q u (r ≫ g) = av O a r

theorem **exists_units** (hu : ∀ k, GenScheme.Compat (O k).L q.left (u k)) {Y : Over S} (g : Y ⟶ G) (hg : g ≫ q = 1) : ∃ a : κ → Γ(Y.left, ⊤)ˣ, nv O q u g = av O a (𝟙 Y)

theorem **rep_vv** (ht : ∀ k, (t k).IsGen) (k : κ) {Y : Over S} (x y : Y ⟶ B) : Rep (O k).L (fun _ _ h => h (x * y).left * (h x.left)⁻¹ * (h y.left)⁻¹ * h (1 : Y ⟶ B).left) (fun _ r z J => val (L

theorem **vv_comp** {Y W : Over S} (r : W ⟶ Y) (x y : Y ⟶ B) (k z J) : val (L

theorem **exists_mul** (hu : ∀ k, GenScheme.Compat (O k).L q.left (u k)) (hU : GenScheme.Univ (fun k => (O k).L) q.left u) (ht : ∀ k, (t k).IsGen) : ∃ m : G ⊗ G ⟶ G, ∀ (W : Over S) (x y : W ⟶ G), (lift x y ≫ m) ≫ q = (x ≫ q) * (y ≫ q) ∧ nv O q u (lift x y ≫ m) = nv O q u x * nv O q u y * (vv t (x ≫ q) (y ≫ q))⁻¹

theorem **exists_inv** (hu : ∀ k, GenScheme.Compat (O k).L q.left (u k)) (hU : GenScheme.Univ (fun k => (O k).L) q.left u) (ht : ∀ k, (t k).IsGen) : ∃ j : G ⟶ G, ∀ (W : Over S) (x : W ⟶ G), (x ≫ j) ≫ q = (x ≫ q)⁻¹ ∧ nv O q u (x ≫ j) = (nv O q u x)⁻¹ * vv t (x ≫ q)⁻¹ (x ≫ q)

theorem **gm_char** (t : W ⟶ splitTorus S X) (z : W.left) (χ : X) : gm t.left z ⊤ (charUnit S X χ) = gm (𝟙 W.left) z ⊤ (splitTorusPointsEquiv S X W t (Multiplicative.ofAdd χ))

theorem **hom_ext_basis** {κ M : Type} [CommGroup M] (b : Module.Basis κ ℤ X) {f g : Multiplicative X →* M} (h : ∀ k, f (Multiplicative.ofAdd (b k)) = g (Multiplicative.ofAdd (b k))) : f = g

theorem **constr_basis** {κ M : Type} [CommGroup M] (b : Module.Basis κ ℤ X) (a : κ → M) (k : κ) : AddMonoidHom.toMultiplicativeLeft (b.constr ℤ fun k => Additive.ofMul (a k)).toAddMonoidHom (Multiplicative.ofAdd (b k)) = a k

theorem **exists_law** (hG : GlobalFnsSt) (hB : IsAbelianScheme B) (hp : ∀ k, ∃ t : LB.Gen (abelian₂ B (O k).L), t.IsGen) (hu : ∀ k, GenScheme.Compat (O k).L q.left (u k)) (hU : GenScheme.Univ (fun k => (O k).L) q.left u) {X : Type} [AddCommGroup X] (b : Module.Basis κ ℤ X) : ∃ _ : GrpObj G, IsCommMonObj G ∧ IsMonHom q ∧ nv O q u η[G] = 1 ∧ ∃ i : splitTorus S X ⟶ G, IsMonHom i ∧ IsPullback i (toUnit _) q η[B] ∧ ∀ (W : Over S) (t : W ⟶ splitTorus S X) (g : W ⟶ G), ((t ≫ i) ∗ g) ≫ q = g ≫ q ∧ nv O q u ((t ≫ i) ∗ g) = av O (fun k => charUnit S X (b k)) t ∗ nv O q u g

theorem **gm_of_nv** {W : Over S} {g g' : W ⟶ G} {c : PV O W} (h₁ : g' ≫ q = g ≫ q) (h₂ : nv O q u g' = c ∗ nv O q u g) (k : κ) (z : W.left) (i : (O k).L.ι) (hi : q.left (g.left z) ∈ (O k).L.U i) : ∃ J, gm g'.left z (q.left ⁻¹ᵁ (O k).L.U i) (u k i) = c k z J ∗ gm g.left z (q.left ⁻¹ᵁ (O k).L.U i) (u k i)

theorem **unit_norm** [IsMonHom q] (h : nv O q u η[G] = 1) (k : κ) (z : (𝟙_ (Over S)).left) (i : (O k).L.ι) (hi : eSec B z ∈ (O k).L.U i) : gm (eSec G) z (q.left ⁻¹ᵁ (O k).L.U i) (u k i) ∗ gm (𝟙 (𝟙_ (Over S)).left) z (((O k).L.pull (eSec B)).U fun _ => i) ((O k).rig.s fun _ => i) = 1

theorem **GlobalTheory.exists_grpObj_gtDatum_of_generators** : GlobalFnsSt → ∀ (S : Scheme.{0}) (E : SplitExt S) (G : Scheme.{0}) (q : G ⟶ E.B.left), letI

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsGrpObjSplitExtensionOfGeneratorsStatement

theorem **_root_.GlobalTheory.T.exists_grpObj_splitExtension_of_generators** : GlobalFnsSt → ∀ (S : Scheme.{0}) (E : SplitExt S) (G : Scheme.{0}) (q : G ⟶ E.B.left), letI

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsHasProjectiveDimensionLEOfResidueField

theorem **projectiveDimension_quotient_maximalIdeal_ne_top** (R : Type u) [CommRing R] [IsRegularLocalRing R] : projectiveDimension (ModuleCat.of R (R ⧸ maximalIdeal R)) ≠ ⊤

theorem **exists_hasProjectiveDimensionLE_of_residueField** (R : Type u) [CommRing R] [IsLocalRing R] [IsNoetherianRing R] (hk : projectiveDimension (ModuleCat.of R (R ⧸ maximalIdeal R)) ≠ ⊤) (M : Type u) [AddCommGroup M] [Module R M] [Module.Finite R M] : ∃ n : ℕ, HasProjectiveDimensionLE (ModuleCat.of R M) n

theorem **exists_hasProjectiveDimensionLE_of_isRegularLocalRing** : ∀ (R : Type) [CommRing R] [IsRegularLocalRing R] (M : Type) [AddCommGroup M] [Module R M] [Module.Finite R M], ∃ n : ℕ, HasProjectiveDimensionLE (ModuleCat.of R M) n

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsIdPowEqOneOfIsFiniteOfFlat

@[simp] theorem **postcomp_apply** (h : C →ₐ[R] F) (f : WithConv (B →ₗ[R] C)) (c : B) : postcomp B h f c = h (f c)

theorem **pow_card_eq_one** {ι : Type*} [Fintype ι] (b : Module.Basis ι C F) (τ : F ≃ₐ[C] F) (x : WithConv (B →ₗ[R] F)) (hx : IsUnit x) (u : WithConv (B →ₗ[R] C)) (h : postcomp B (τ.toAlgHom.restrictScalars R) x = x ∗ postcomp B (IsScalarTower.toAlgHom R C F) u) : u ^ Fintype.card ι = 1

theorem **mul_comp_antipode** (u : B →ₐ[R] C) : toConv u ∗ toConv (u.comp (HopfAlgebra.antipodeAlgHom R B)) = 1

theorem **point_pow_card** {ι : Type*} [Fintype ι] (b : Module.Basis ι C (C ⊗[R] B)) (u : B →ₐ[R] C) : toConv u ^ Fintype.card ι = 1

theorem **comp_convPow** (h : C →ₐ[R] F) (f : WithConv (B →ₐ[R] C)) (n : ℕ) : h.comp (f ^ n).ofConv = (toConv (h.comp f.ofConv) ^ n).ofConv

theorem **exists_id_pow_eq_one** [Module.Finite R B] [Module.Flat R B] : ∃ N : ℕ, 1 ≤ N ∧ toConv (AlgHom.id R B) ^ N = 1

theorem **isCocomm_of_mul_comm** {R B : Type*} [CommRing R] [CommRing B] [Bialgebra R B] (h : ∀ f g : WithConv (B →ₐ[R] B ⊗[R] B), f ∗ g = g ∗ f) : Coalgebra.IsCocomm R B

theorem **exists_id_pow_eq_one_of_isFinite_of_flat** (R : Type) [CommRing R] (H : Over (Spec (CommRingCat.of R))) [GrpObj H] [IsCommMonObj H] (hf : IsFinite H.hom) (hfl : Flat H.hom) : ∃ N : ℕ, 1 ≤ N ∧ 𝟙 H ^ N = 1

theorem **main** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (H : Over (Spec (CommRingCat.of R))) [GrpObj H] [IsCommMonObj H], IsFinite H.hom → Flat H.hom → ∃ N : ℕ, 1 ≤ N ∧ 𝟙 H ^ N = 1

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsIsGenOfInvariant

theorem **exists_isGen_pull** {X Y : Scheme.{0}} (M : LB X) (v : LB.Gen M) (hv : v.IsGen) (f : Y ⟶ X) : ∃ c : LB.Gen (M.pull f), c.IsGen

theorem **exists_isGen_of_invariant** {X Y Y' : Scheme.{0}} {L : LB X} (hL : L.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (m : Y' ⟶ Y) {I : Type} (t : I → (Y' ⟶ Y')) (ht : ∀ i, t i ≫ m = m) (hd : ∀ (U : Y.Opens) (f : Γ(Y', m ⁻¹ᵁ U)), (∀ i (e : m ⁻¹ᵁ U ≤ t i ⁻¹ᵁ (m ⁻¹ᵁ U)), ((t i).appLE _ _ e).hom f = f) → ∃! g : Γ(Y, U), (m.app U).hom g = f) (φ' : K → (Y' ⟶ X)) (hφ : ∀ k, φ' k = m ≫ φ k) (u : LB.Gen (LB.LinComb L φ' ε)) (hu : u.IsGen) (hinv : ∀ i y J, Good L y J → val u (t i) y J = val u (𝟙 Y') y J) : ∃ v : LB.Gen (LB.LinComb L φ ε), v.IsGen

theorem **weilNondeg_of_descent** : (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ), (n : K) ≠ 0 → ∀ (U : P.A.left.Opens) (f : Γ(P.A.left, (mulA P.A n).left ⁻¹ᵁ U)), (∀ y : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ P.A, y ^ n = 1 → ∀ (e : (mulA P.A n).left ⁻¹ᵁ U ≤ ((toUnit P.A ≫ y) ∗ 𝟙 P.A).left ⁻¹ᵁ ((mulA P.A n).left ⁻¹ᵁ U)), (((toUnit P.A ≫ y) ∗ 𝟙 P.A).left.appLE _ _ e).hom f = f) → ∃! g : Γ(P.A.left, U), ((mulA P.A n).left.app U).hom g = f) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ) (x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A), x ^ n = 1 → ∃ (u : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsIsSemiAbelianIsGt

theorem **exists_isSemiAbelian_isGt** : GlobalFnsSt → ∀ (S : Scheme.{0}) (E : SplitExt S), ∃ (G : Over S) (_ : GrpObj G), IsSemiAbelian G ∧ E.IsGt G

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsKerIffMinors

theorem **exists_ker_iff_minors** {K : Type} [Field K] {a b : ℕ} (D : Matrix (Fin b) (Fin a) K) : (∃ v : Fin a → K, v ≠ 0 ∧ D.mulVec v = 0) ↔ ∀ σ : Fin a → Fin b, (D.submatrix σ id).det = 0

theorem **nontrivial_ker_iff** {R K : Type} [CommRing R] [Field K] (g : R →+* K) {a b : ℕ} (D : Matrix (Fin b) (Fin a) R) : Nontrivial {v : Fin a → K // (D.map g).mulVec v = 0} ↔ ∀ σ : Fin a → Fin b, (D.submatrix σ id).det ∈ RingHom.ker g

theorem **nontrivial_sec_iff** (N : LB X) : Nontrivial (LB.Sec N) ↔ ∃ (s : LB.Sec N) (i : N.ι), s.f i ≠ 0

theorem **pull_pw** (M : LB X) (π : Y ⟶ X) (t : ℤ) : (pw M t).pull π = pw (M.pull π) t

theorem **rs_ne_zero** [IsIntegral X] {U V : X.Opens} (h : V ≤ U) (hV : (V : Set X).Nonempty) {x : Γ(X, U)} (hx : x ≠ 0) : rs X h x ≠ 0

theorem **nonempty_of_ne_zero** {U : X.Opens} {x : Γ(X, U)} (hx : x ≠ 0) : (U : Set X).Nonempty

theorem **triv_iff** [IsIntegral X] (hX : IsField Γ(X, ⊤)) (N : LB X) (hN : ⊔ i, N.U i = ⊤) : Triv N ↔ Nontrivial (LB.Sec N) ∧ Nontrivial (LB.Sec (pw N (-1)))

theorem **triv_fibre_iff** {R : Type} [CommRing R] {f : X ⟶ Spec (CommRingCat.of R)} {M : LB X} (hM : M.IsLB) (P : Perf f M) (Q : Perf f (pw M (-1))) [IsProper f] [GeometricallyIntegral f] {K : Type} [Field K] (g : R →+* K) {X' : Scheme.{0}} {π : X' ⟶ X} {f' : X' ⟶ Spec (CommRingCat.of K)} (h : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) : Triv (M.pull π) ↔ (∀ σ : Fin P.a → Fin P.b, (P.D₀.submatrix σ id).det ∈ RingHom.ker g) ∧ ∀ σ : Fin Q.a → Fin Q.b, (Q.D₀.submatrix σ id).det ∈ RingHom.ker g

theorem **isClosed_trivLocus** : ∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) (M : LB X), M.IsLB → Nonempty (Perf f M) → Nonempty (Perf f (pw M (-1))) → IsProper f → GeometricallyIntegral f → IsClosed {p : PrimeSpectrum R | ∃ (K : Type) (_ : Field K) (g : R →+* K) (X' : Scheme.{0}) (π : X' ⟶ X) (f' : X' ⟶ Spec (CommRingCat.of K)), RingHom.ker g = p.asIdeal ∧ IsPullback π f' f (Spec.map (CommRingCat.ofHom g)) ∧ Triv (M.pull π)}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsKernelVector

theorem **exists_minor** {K m n : Type} [Field K] [Fintype m] [Fintype n] [DecidableEq n] (A : Matrix m n K) (hA : ∀ w, A.mulVec w = 0 → w = 0) : ∃ ρ : n → m, (A.submatrix ρ id).det ≠ 0

theorem **exists_kernel_vector** {R K : Type} [CommRing R] [Field K] (g₀ : R →+* K) {a b : ℕ} (D : Matrix (Fin b) (Fin a) R) (v : Fin a → K) (hv : v ≠ 0) (hker : ∀ w, (D.map g₀).mulVec w = 0 → ∃ c : K, w = c • v) : ∃ (δ : R) (v₀ : Fin a → R), g₀ δ ≠ 0 ∧ ∀ (R' : Type) [CommRing R'] (g : R →+* R'), IsUnit (g δ) → ∀ w, (D.map g).mulVec w = 0 → ∃ c : R', w = c • (g ∘ v₀)

theorem **mem_basicOpen_rs** {U V : Y.Opens} (h : V ≤ U) (a : Γ(Y, U)) {y : Y} (hy : y ∈ V) : y ∈ Y.basicOpen (rs Y h a) ↔ y ∈ Y.basicOpen a

theorem **isUnit_of_basicOpen** {U : Y.Opens} (a : Γ(Y, U)) (h : ∀ y ∈ U, y ∈ Y.basicOpen a) : IsUnit a

theorem **mem_of_isUnit** {U : Y.Opens} {a : Γ(Y, U)} (h : IsUnit a) {y : Y} (hy : y ∈ U) : y ∈ Y.basicOpen a

theorem **sec_basicOpen** {N : LB Y} (s : LB.Sec N) {i j : N.ι} {y : Y} (hi : y ∈ N.U i) (h : y ∈ Y.basicOpen (s.f j)) : y ∈ Y.basicOpen (s.f i)

theorem **sec_ext** {N : LB Y} (s s' : LB.Sec N) {I : Type} (V : I → Y.Opens) (c : I → N.ι) (le : ∀ i, V i ≤ N.U (c i)) (hV : ∀ y, ∃ i, y ∈ V i) (h : ∀ i, rs Y (le i) (s.f (c i)) = rs Y (le i) (s'.f (c i))) : s = s'

theorem **sec_ratio** {N : LB Y} (hc : ⊔ j, N.U j = ⊤) (s : LB.Sec N) (t : LB.Gen N) (ht : t.IsGen) : ∃ ρ : Γ(Y, ⊤), ∀ j, s.f j = rs Y le_top ρ * (t.s j).val

theorem **app_cst** {f : X ⟶ Spec (CommRingCat.of R)} {g : R →+* R'} {π : X' ⟶ X} {f' : X' ⟶ Spec (CommRingCat.of R')} (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom g)) (W : X.Opens) (r : R) : (π.app W).hom (cst f W r) = cst f' (π ⁻¹ᵁ W) (g r)

theorem **mem_basicOpen_cst** (f : X ⟶ Spec (CommRingCat.of R)) (W : X.Opens) (r : R) {x : X} (hx : x ∈ W) : x ∈ X.basicOpen (cst f W r) ↔ r ∉ (f x).asIdeal

theorem **loc_smul** (π : X' ⟶ X) (f' : X' ⟶ Spec (CommRingCat.of R')) (c : R') (v : Fin P.a → R') (i : P.I) : loc P π f' (c • v) i = cst f' _ c * loc P π f' v i

theorem **exists_mem** (x : X) : ∃ i, x ∈ P.W i

theorem **loc_comp** (v : Fin P.a → R) (i : P.I) : loc P π f' (g ∘ v) i = (π.app (P.W i)).hom (loc₀ P v i)

theorem **triv_of_loc** (v : Fin P.a → R') (hv : (P.D₀.map g).mulVec v = 0) (H : ∀ y : X', ∃ i, y ∈ X'.basicOpen (loc P π f' v i)) : Triv (M.pull π)

theorem **loc_of_triv** (hM : M.IsLB) (H : Triv (M.pull π)) : ∃ v : Fin P.a → R', (P.D₀.map g).mulVec v = 0 ∧ (∀ i, IsUnit (loc P π f' v i)) ∧ (Function.Surjective f'.appTop.hom → ∀ w, (P.D₀.map g).mulVec w = 0 → ∃ c : R', w = c • v)

theorem **seesaw** : ∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) (M : LB X), M.IsLB → Nonempty (Perf f M) → UniversallyClosed f → Surjective f → ∀ (K : Type) [Field K] (g₀ : R →+* K) (X₀ : Scheme.{0}) (π₀ : X₀ ⟶ X) (f₀ : X₀ ⟶ Spec (CommRingCat.of K)), IsPullback π₀ f₀ f (Spec.map (CommRingCat.ofHom g₀)) → Function.Bijective f₀.appTop.hom → Triv (M.pull π₀) → ∃ (r : R) (I : Ideal R), g₀ r ≠ 0 ∧ I.FG ∧ ∀ (R' : Type) [CommRing R'] (g : R →+* R') (X' : Scheme.{0}) (π : X' ⟶ X) (f' : X' ⟶ Spec (CommRingCat.of R')), IsPullback π f' f (Spec.map (CommRingCat.ofHom g)) → IsUnit (g r) → (Triv (M.pull π) ↔ I ≤ RingHom.ker g)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsLineBundleExtension

theorem **genericPoint_mem** {U : X.Opens} {x : X} (hx : x ∈ U) : genericPoint X ∈ U

theorem **algebraMap_germ** {U : X.Opens} {x : X} (hx : x ∈ U) (h : genericPoint X ∈ U) (s : Γ(X, U)) : algebraMap (X.presheaf.stalk x) X.functionField (X.presheaf.germ U x hx s) = X.presheaf.germ U _ h s

theorem **germ_mem** {U : X.Opens} {x : X} (hx : x ∈ U) (h : genericPoint X ∈ U) (s : Γ(X, U)) : X.presheaf.germ U _ h s ∈ localSubring x

theorem **exists_section_of_mem** {x : X} {t : X.functionField} (ht : t ∈ localSubring x) : ∃ (U : X.Opens) (hx : x ∈ U) (s : Γ(X, U)), X.presheaf.germ U _ (genericPoint_mem hx) s = t

theorem **eventually_mem** {x : X} {t : X.functionField} (ht : t ∈ localSubring x) : ∀ᶠ y in nhds x, t ∈ localSubring y

theorem **exists_section** {U : X.Opens} (h : genericPoint X ∈ U) {t : X.functionField} (ht : ∀ x ∈ U, t ∈ localSubring x) : ∃ s : Γ(X, U), X.presheaf.germ U _ h s = t

theorem **exists_pow_mul_mem** {U : X.Opens} (hU : IsAffineOpen U) (a : Γ(X, U)) (h : genericPoint X ∈ U) (ha : genericPoint X ∈ X.basicOpen a) {t : X.functionField} (ht : ∀ y ∈ X.basicOpen a, t ∈ localSubring y) : ∃ n : ℕ, ∀ x ∈ U, X.presheaf.germ U _ h a ^ n * t ∈ localSubring x

theorem **inv_germ_mem** {U : X.Opens} (a : Γ(X, U)) (h : genericPoint X ∈ U) {y : X} (hy : y ∈ X.basicOpen a) : X.presheaf.germ U _ h a ≠ 0 ∧ (X.presheaf.germ U _ h a)⁻¹ ∈ localSubring y

theorem **exists_gcd_fraction** {R K : Type*} [CommRing R] [IsDomain R] [UniqueFactorizationMonoid R] [Field K] [Algebra R K] [IsFractionRing R K] {ι : Type*} [Finite ι] (c : ι → K) (hc : ∀ k, c k ≠ 0) : ∃ h : K, h ≠ 0 ∧ (∀ k, c k / h ∈ (algebraMap R K).range) ∧ ∀ t : K, Nonempty ι → (∀ k, c k * t ∈ (algebraMap R K).range) → h * t ∈ (algebraMap R K).range

theorem **mem_localUnits** {x : X} {t : X.functionFieldˣ} : t ∈ localUnits x ↔ t.val ∈ localSubring x ∧ t.val⁻¹ ∈ localSubring x

theorem **exists_local_equation** [NoetherianSpace X] {T : Set (X.Opens × X.functionField^×)} (hT : Compatible T) (x : X) [UniqueFactorizationMonoid (X.presheaf.stalk x)] : ∃ (W : X.Opens) (h : X.functionField^×), x ∈ W ∧ ∀ p ∈ T, ∀ y, y ∈ W → y ∈ p.1 → h / p.2 ∈ localUnits y

theorem **exists_compatible_cover** [NoetherianSpace X] (hX : ∀ x : X, UniqueFactorizationMonoid (X.presheaf.stalk x)) {T₀ : Set (X.Opens × X.functionField^×)} (h₀ : Compatible T₀) : ∃ T, T₀ ⊆ T ∧ Compatible T ∧ ∀ x, ∃ p ∈ T, x ∈ p.1

theorem **val_unitVal** {U : X.Opens} (h : genericPoint X ∈ U) (a : Γ(X, U)^×) : (unitVal U a).val = X.presheaf.germ U _ h a.val

theorem **unitVal_res** {U V : X.Opens} (hle : V ≤ U) (h : genericPoint X ∈ V) (a : Γ(X, U)^×) : unitVal V (res hle a) = unitVal U a

theorem **unit_ext** {U : X.Opens} {a b : Γ(X, U)^×} (h : genericPoint X ∈ U → unitVal U a = unitVal U b) : a = b

theorem **unitVal_mem** {U : X.Opens} {y : X} (hy : y ∈ U) (a : Γ(X, U)^×) : unitVal U a ∈ localUnits y

theorem **exists_unit** {U : X.Opens} {t : X.functionField^×} (ht : ∀ y ∈ U, t ∈ localUnits y) : ∃ a : Γ(X, U)^×, genericPoint X ∈ U → unitVal U a = t

theorem **imageUnit_res** {U V : Y.Opens} (hle : V ≤ U) (a : Γ(Y, U)^×) : imageUnit j V (res hle a) = res (j.image_mono hle) (imageUnit j U a)

theorem **imageUnit_pull** (U : X.Opens) (a : Γ(X, U)^×) : imageUnit j (j ⁻¹ᵁ U) (pullU j U a) = res (Set.image_preimage_subset j U.1) a

theorem **imageVal_res** {U V : Y.Opens} (hle : V ≤ U) (h : genericPoint X ∈ j ''ᵁ V) (a : Γ(Y, U)^×) : imageVal j V (res hle a) = imageVal j U a

theorem **imageVal_res_pull** {U : X.Opens} {V : Y.Opens} (hle : V ≤ j ⁻¹ᵁ U) (h : genericPoint X ∈ j ''ᵁ V) (a : Γ(X, U)^×) : imageVal j V (res hle (pullU j U a)) = unitVal U a

theorem **image_unit_ext** {V : Y.Opens} {a b : Γ(Y, V)^×} (h : genericPoint X ∈ j ''ᵁ V → imageVal j V a = imageVal j V b) : a = b

theorem **exists_image_unit** {V : Y.Opens} {t : X.functionField^×} (ht : ∀ y ∈ V, t ∈ localUnits (j y)) : ∃ a : Γ(Y, V)^×, genericPoint X ∈ j ''ᵁ V → imageVal j V a = t

theorem **mem_image_inf** {U V : Y.Opens} {x : X} (hU : x ∈ j ''ᵁ U) (hV : x ∈ j ''ᵁ V) : x ∈ j ''ᵁ (U ⊓ V)

theorem **exists_equations** (L : LB Y) (hL : L.IsLB) : ∃ f : L.ι → X.functionField^×, ∀ i k, genericPoint X ∈ j ''ᵁ L.U i → genericPoint X ∈ j ''ᵁ L.U k → imageVal j _ (L.g i k) = f i / f k

theorem **isGen_of_equations** (M N : LB Y) (a : M.ι → X.functionField^×) (b : N.ι → X.functionField^×) (ha : ∀ i k, genericPoint X ∈ j ''ᵁ M.U i → genericPoint X ∈ j ''ᵁ M.U k → imageVal j _ (M.g i k) = a i / a k) (hb : ∀ i k, genericPoint X ∈ j ''ᵁ N.U i → genericPoint X ∈ j ''ᵁ N.U k → imageVal j _ (N.g i k) = b i / b k) (s : ∀ p : M.ι × N.ι, Γ(Y, M.U p.1 ⊓ N.U p.2)^×) (hs : ∀ p, genericPoint X ∈ j ''ᵁ (M.U p.1 ⊓ N.U p.2) → imageVal j _ (s p) = a p.1 / b p.2) : LB.Gen.IsGen ((s) : LB.Iso N M)

theorem **imageVal_pull** (M : LB X) (p q : PUnit → M.ι) (h : genericPoint X ∈ j ''ᵁ (LB.wchart M (fun _ => j) p ⊓ LB.wchart M (fun _ => j) q)) : imageVal j _ ((M.pull j).g p q) = unitVal _ (M.g (p ⟨⟩) (q ⟨⟩))

theorem **exists_lineBundle_extension** (X Y : Scheme.{0}) [IsIntegral X] [IsLocallyNoetherian X] [CompactSpace X] (j : Y ⟶ X) (hj : IsOpenImmersion j) (hX : ∀ x : X, UniqueFactorizationMonoid (X.presheaf.stalk x)) (L : LB Y) (hL : L.IsLB) : ∃ L' : LB X, L'.IsLB ∧ ∃ φ : LB.Iso (L'.pull j) L, φ.IsGen

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsLocalUnitOfWeight

theorem **ev_single** (θ : C →+* R) (φ : Multiplicative X →* R^×) (x : X) (c : C) : ev θ φ (AddMonoidAlgebra.single x c) = θ c * (φ (Multiplicative.ofAdd x)).val

theorem **comp_ev** (f : R →+* R') (θ : C →+* R) (φ : Multiplicative X →* R^×) : f.comp (ev θ φ) = ev (f.comp θ) ((Units.map f.toMonoidHom).comp φ)

theorem **univ_val** (x : X) : (univ X C (Multiplicative.ofAdd x)).val = AddMonoidAlgebra.single x 1

theorem **cst_apply** (c : C) : cst X C c = AddMonoidAlgebra.single 0 c

theorem **ev_univ** : ev (cst X C) (univ X C) = RingHom.id _

theorem **ev_comp_cst** (θ : C →+* R) (φ : Multiplicative X →* R^×) : (ev θ φ).comp (cst X C) = θ

theorem **ev_comp_univ** (θ : C →+* R) (φ : Multiplicative X →* R^×) : (Units.map (ev θ φ).toMonoidHom).comp (univ X C) = φ

theorem **coeff_ev** (δ : C →+* R) (ψ : Multiplicative X →* R^×) (f : AddMonoidAlgebra C X) (x : X) : (ev ((cst X R).comp δ) (univ X R * (Units.map (cst X R).toMonoidHom).comp ψ) f).coeff x = δ (f.coeff x) * (ψ (Multiplicative.ofAdd x)).val

theorem **IsTorsorOn.of_equiv** {α : A →+* C} {ρ : C →+* AddMonoidAlgebra C X} (e : R ≃+* R') (h : IsTorsorOn R' α ρ) : IsTorsorOn R α ρ

theorem **coaction_coeff** {α : A →+* C} {ρ : C →+* AddMonoidAlgebra C X} (h : IsTorsorOn (AddMonoidAlgebra (AddMonoidAlgebra C X) X) α ρ) (c : C) (x : X) : ρ ((ρ c).coeff x) = AddMonoidAlgebra.single x ((ρ c).coeff x)

theorem **exists_homogeneous** {α : A →+* C} (hα : α.FaithfullyFlat) {ρ : C →+* AddMonoidAlgebra C X} (h : ∀ (R : Type) [CommRing R], IsTorsorOn R α ρ) (χ : X) (p : Ideal A) (hp : p ≠ ⊤) : ∃ a ∉ p, ∃ c c' : C, c * c' = α a ∧ ρ c = AddMonoidAlgebra.single χ c

theorem **toSpec_appTop** {R : CommRingCat.{0}} (θ : R →+* Γ(Z, ⊤)) : (Z.toSpecΓ ≫ Spec.map (CommRingCat.ofHom θ)).appTop.hom.comp (Scheme.ΓSpecIso R).inv.hom = θ

theorem **comp_pt** (hU : IsAffineOpen U) (f : Z' ⟶ Z) (θ : Γ(Y, U) →+* Γ(Z, ⊤)) : f ≫ pt hU θ = pt hU (f.appTop.hom.comp θ)

theorem **pt_comp** {B : Scheme.{0}} {V : B.Opens} (hU : IsAffineOpen U) (hV : IsAffineOpen V) (q : Y ⟶ B) (e : U ≤ q ⁻¹ᵁ V) (θ : Γ(Y, U) →+* Γ(Z, ⊤)) : pt hU θ ≫ q = pt hV (θ.comp (q.appLE V U e).hom)

theorem **le_pt_preimage** (hU : IsAffineOpen U) (θ : Γ(Y, U) →+* Γ(Z, ⊤)) : ⊤ ≤ pt hU θ ⁻¹ᵁ U

theorem **pt_appLE** (hU : IsAffineOpen U) (θ : Γ(Y, U) →+* Γ(Z, ⊤)) (e : ⊤ ≤ pt hU θ ⁻¹ᵁ U) : ((pt hU θ).appLE U ⊤ e).hom = θ

theorem **pt_injective** (hU : IsAffineOpen U) {θ θ' : Γ(Y, U) →+* Γ(Z, ⊤)} (h : pt hU θ = pt hU θ') : θ = θ'

theorem **exists_eq_pt** (hU : IsAffineOpen U) (g : Z ⟶ Y) (e : ⊤ ≤ g ⁻¹ᵁ U) : ∃ θ, g = pt hU θ

theorem **toHom_comp** {Z Z' : Over S} (f : Z' ⟶ Z) (t : Z ⟶ splitTorus S X) : toHom (f ≫ t) = (Units.map f.left.appTop.hom.toMonoidHom).comp (toHom t)

theorem **act_comp** [IsMonHom q] (hi : IsPullback i (toUnit _) q η[B]) {Z : Over S} (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) : ((t ≫ i) * g) ≫ q = g ≫ q

theorem **exists_coaction** [IsMonHom q] (hi : IsPullback i (toUnit _) q η[B]) {V : B.left.Opens} (hU : IsAffineOpen (q.left ⁻¹ᵁ V)) : ∃ ρ : Γ(G.left, q.left ⁻¹ᵁ V) →+* AddMonoidAlgebra Γ(G.left, q.left ⁻¹ᵁ V) X, ∀ (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (θ), g.left = pt hU θ → ((t ≫ i) * g).left = pt hU ((ev θ (toHom t)).comp ρ)

theorem **isTorsorOn** [IsMonHom q] [IsMonHom i] (hi : IsPullback i (toUnit _) q η[B]) {V : B.left.Opens} (hV : IsAffineOpen V) (hU : IsAffineOpen (q.left ⁻¹ᵁ V)) {ρ : Γ(G.left, q.left ⁻¹ᵁ V) →+* AddMonoidAlgebra Γ(G.left, q.left ⁻¹ᵁ V) X} (hp : ∀ (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (θ), g.left = pt hU θ → ((t ≫ i) ∗ g).left = pt hU ((ev θ (toHom t)).comp ρ)) (Z : Scheme.{0}) : IsTorsorOn Γ(Z, ⊤) (q.left.appLE V _ le_rfl).hom ρ

theorem **faithfullyFlat_appLE** {Y B : Scheme.{0}} (q : Y ⟶ B) [Flat q] [Surjective q] {V : B.Opens} (hV : IsAffineOpen V) (hU : IsAffineOpen (q ⁻¹ᵁ V)) : (q.appLE V (q ⁻¹ᵁ V) le_rfl).hom.FaithfullyFlat

theorem **exists_local_unit_of_weight** : ∀ (S : Scheme.{0}) (X : Type) [AddCommGroup X] (G B : Over S) [GrpObj G] [GrpObj B] (q : G ⟶ B) (i : splitTorus S X ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → IsAffineHom q.left → ∀ (χ : X) (b : B.left), ∃ (V : B.left.Opens) (_ : b ∈ V) (u : Γ(G.left, q.left ⁻¹ᵁ V)ˣ), ∀ (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (h₁ : ⊤ ≤ ((t ≫ i) ∗ g).left ⁻¹ᵁ (q.left ⁻¹ᵁ V)) (h₂ : ⊤ ≤ g.left ⁻¹ᵁ (q.left ⁻¹ᵁ V)), (((t ≫ i) ∗ g).left.appLE _ ⊤ h₁).hom u.val = t.left.appTop.hom (charUnit S X χ).val ∗ (g.left.appLE _ ⊤ h₂).hom u.val

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsMulEqSmulOneOfMulVecInjective

theorem **exists_mul_eq_smul_one_of_mulVec_injective** {R K : Type} [CommRing R] [Field K] (g : R →+* K) {m l : Type} [Fintype m] [Fintype l] [DecidableEq l] (M : Matrix m l R) (hM : ∀ x : l → K, (M.map g).mulVec x = 0 → x = 0) : ∃ (r : R) (T : Matrix l m R), g r ≠ 0 ∧ T ∗ M = r • (1 : Matrix l l R)

theorem **exists_kernel_certificate** {R K : Type} [CommRing R] [Field K] (g : R →+* K) {a b c : Type} [Fintype a] [Fintype b] [Fintype c] [DecidableEq a] (D₀ : Matrix b a R) (D₁ : Matrix c b R) (h0 : D₁ ∗ D₀ = 0) (hex : ∀ w : b → K, (D₁.map g).mulVec w = 0 → ∃ v, (D₀.map g).mulVec v = w) : ∃ (r : R) (n : ℕ) (B : Matrix a (Fin n) R) (C : Matrix (Fin n) a R) (H : Matrix a b R), g r ≠ 0 ∧ D₀ ∗ B = 0 ∧ C ∗ B = r • (1 : Matrix (Fin n) (Fin n) R) ∧ B ∗ C + H ∗ D₀ = r • (1 : Matrix a a R)

theorem **exists_matrix_basis_ker_of_exact_at_point** : ∀ (R K : Type) [CommRing R] [Field K] (g₀ : R →+* K) (a b c : ℕ) (D₀ : Matrix (Fin b) (Fin a) R) (D₁ : Matrix (Fin c) (Fin b) R), D₁ ∗ D₀ = 0 → (∀ w : Fin b → K, (D₁.map g₀).mulVec w = 0 → ∃ v, (D₀.map g₀).mulVec v = w) → ∃ (r : R) (n : ℕ) (B : Matrix (Fin a) (Fin n) R), g₀ r ≠ 0 ∧ D₀ ∗ B = 0 ∧ ∀ (R' : Type) [CommRing R'] (g : R →+* R'), IsUnit (g r) → ∀ v : Fin a → R', (D₀.map g).mulVec v = 0 → ∃! z : Fin n → R', (B.map g).mulVec z = v

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsNoetherianModelOfSmooth

theorem **exists_common_model** (R B C : Type) [CommRing R] [CommRing B] [CommRing C] [Algebra R B] [Algebra R C] [Algebra.Smooth R B] [Algebra.Smooth R C] : ∃ (A B₁ C₁ : Type) (_ : CommRing A) (_ : Algebra A R) (_ : CommRing B₁) (_ : CommRing C₁) (_ : Algebra A B₁) (_ : Algebra A C₁), IsNoetherianRing A ∧ Algebra.Smooth A B₁ ∧ Algebra.Smooth A C₁ ∧ Nonempty (R ⊗[A] B₁ ≃ₐ[R] B) ∧ Nonempty (R ⊗[A] C₁ ≃ₐ[R] C)

theorem **range_incl_mono** {S T : Subalgebra A R} (h : S ≤ T) : (incl C₁ S).range ≤ (incl C₁ T).range

theorem **exists_fg_mem_range_incl** (x : R ⊗[A] C₁) : ∃ S : Subalgebra A R, S.FG ∧ x ∈ (incl C₁ S).range

theorem **exists_fg_subset_range_incl** (t : Finset (R ⊗[A] C₁)) : ∃ S : Subalgebra A R, S.FG ∧ (t : Set (R ⊗[A] C₁)) ⊆ (incl C₁ S).range

theorem **exists_fg_range_le** [Algebra.FiniteType A B₁] (φ : B₁ →ₐ[A] R ⊗[A] C₁) : ∃ S : Subalgebra A R, S.FG ∧ φ.range ≤ (incl C₁ S).range

theorem **incl_injective** [Module.Flat A C₁] (S : Subalgebra A R) : Function.Injective (incl C₁ S)

theorem **exists_lift** {D E : Type} [CommRing D] [CommRing E] [Algebra A D] [Algebra A E] (φ : B₁ →ₐ[A] E) (i : D →ₐ[A] E) (hi : Function.Injective i) (h : φ.range ≤ i.range) : ∃ ψ : B₁ →ₐ[A] D, ∀ b, i (ψ b) = φ b

theorem **cancel_one_tmul** (S : Subalgebra A R) (x : S ⊗[A] C₁) : Algebra.TensorProduct.cancelBaseChange A S R R C₁ (1 ⊗ₜ x) = incl C₁ S x

theorem **incl_tmul** (S : Subalgebra A R) (s : S) (c : C₁) : incl C₁ S (s ⊗ₜ c) = (s : R) • ((1 : R) ⊗ₜ[A] c)

theorem **incl_smul** (S : Subalgebra A R) (s : S) (x : S ⊗[A] C₁) : incl C₁ S (s • x) = (s : R) • incl C₁ S x

theorem **extend_tmul** (S : Subalgebra A R) {D : Type} [CommRing D] [Algebra A D] [Algebra S D] [IsScalarTower A S D] (ψ : B₁ →ₐ[A] D) (s : S) (b : B₁) : extend S ψ (s ⊗ₜ b) = s • ψ b

theorem **incl_extend** (S : Subalgebra A R) (F : R ⊗[A] B₁ →ₐ[R] R ⊗[A] C₁) (ψ : B₁ →ₐ[A] S ⊗[A] C₁) (h : ∀ b, incl C₁ S (ψ b) = F (1 ⊗ₜ b)) (x : S ⊗[A] B₁) : incl C₁ S (extend S ψ x) = F (incl B₁ S x)

theorem **val_extend** (S : Subalgebra A R) (F : R ⊗[A] B₁ →ₐ[R] R) (ψ : B₁ →ₐ[A] S) (h : ∀ b, S.val (ψ b) = F (1 ⊗ₜ b)) (x : S ⊗[A] B₁) : S.val (extend S ψ x) = F (incl B₁ S x)

theorem **exists_noetherian_model_of_smooth** : ∀ (R B C : Type) [CommRing R] [CommRing B] [CommRing C] [Algebra R B] [Algebra R C] [Algebra.Smooth R B] [Algebra.Smooth R C] (f g : B →ₐ[R] C) (r : B →ₐ[R] R), ∃ (R₀ B₀ C₀ : Type) (_ : CommRing R₀) (_ : IsNoetherianRing R₀) (_ : Algebra R₀ R) (_ : CommRing B₀) (_ : CommRing C₀) (_ : Algebra R₀ B₀) (_ : Algebra R₀ C₀) (f₀ g₀ : B₀ →ₐ[R₀] C₀) (r₀ : B₀ →ₐ[R₀] R₀) (β' : TensorProduct R₀ R B₀ ≃ₐ[R] B) (δ' : TensorProduct R₀ R C₀ ≃ₐ[R] C), Module.Flat R₀ B₀ ∧ Module.Flat R₀ C₀ ∧ ∀ b : B₀, f (β' (1 ⊗ₜ b)) = δ' (1 ⊗ₜ f₀ b) ∧ g (β' (1 ⊗ₜ b)) = δ' (1 ⊗ₜ g₀ b) ∧ r (β' (1 ⊗ₜ b)) = algebraMap R₀ R (r₀ b)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsOddNeZero

theorem **toRing_apply** (u : A ⟶ A) [IsMonHom u] (x : A ⟶ A) : (toRing u).1 (Additive.ofMul x) = Additive.ofMul (x ≫ u)

theorem **natCast_mul_apply** (n : ℕ) (f : homRing A) (x : Additive (A ⟶ A)) : ((n : homRing A) ∗ f).1 x = n • f.1 x

theorem **homRing_ext** {f g : homRing A} (h : f.1 (Additive.ofMul (𝟙 A)) = g.1 (Additive.ofMul (𝟙 A))) : f = g

theorem **toRing_pow** (σ : A ≅ A) [IsMonHom σ.hom] (m : ℕ) : toRing σ.hom ^ m = toRing (iter σ m).hom

theorem **hasGen_iter** {L : LB A.left} (hL : L.IsLB) (σ : A ≅ A) (h : HasGen L (K

theorem **exists_iter_fix** (σ : A ≅ A) [IsMonHom σ.hom] (T : Over S) (n : ℕ) [Finite {x : T ⟶ A // x ^ n = 1}] : ∃ m : ℕ, m ≠ 0 ∧ ∀ x : T ⟶ A, x ^ n = 1 → x ≫ (iter σ m).hom = x

theorem **exists_odd_ne_zero** (K : Type) [Field K] : ∃ n : ℕ, Odd n ∧ 5 ≤ n ∧ (n : K) ≠ 0

theorem **serre_algClosed** : (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) → (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (B : Over S) [GrpObj B] [IsCommMonObj B] (M : ℕ), 1 ≤ M → ∀ u : P.A ⟶ B, IsMonHom u → (∀ (T : Over S) (x : T ⟶ P.A), x ^ M = 1 → x ≫ u = 1) → (∃ φ : P.A ⟶ B, IsMonHom φ ∧ u = φ ^ M) ∧ (u ^ M = 1 → u = 1)) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (q : ℕ), 2 ≤ q → ∀ u : P.A ⟶ P.A, (∀ k : ℕ, ∃ ψ : P.A ⟶ P.A, IsMonHom ψ ∧ u = ψ ^ (q ^ k)) → u = 1) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ), Odd n → 5 ≤ n → (n : K) ≠ 0 → ∀ σ : P.A ≅ P.A, IsMonHom σ.hom → (∃ c : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsOneAddKill

theorem **sec_ext** {V : Z.Opens} {a b : Γ(Z, V)} (h : ∀ z (hz : z ∈ V), (Z.presheaf.germ V z hz).hom a = (Z.presheaf.germ V z hz).hom b) : a = b

theorem **germ_res'** {V W : Z.Opens} (h : V ≤ W) (a : Γ(Z, W)) (z : Z) (hz : z ∈ V) : (Z.presheaf.germ V z hz).hom ((Z.presheaf.map (homOfLE h).op).hom a) = (Z.presheaf.germ W z (h hz)).hom a

theorem **reg_mono** {V W : Z.Opens} (h : V ≤ W) {F : Fn Z} (hF : F ∈ Reg W) : F ∈ Reg V

theorem **reg_congr** {V : Z.Opens} {F G : Fn Z} (hF : F ∈ Reg V) (h : ∀ z ∈ V, F z = G z) : G ∈ Reg V

theorem **reg_glue** {V : Z.Opens} {F : Fn Z} (h : ∀ z ∈ V, ∃ O : Z.Opens, z ∈ O ∧ F ∈ Reg O) : F ∈ Reg V

theorem **reg_inv** {V : Z.Opens} {F : Fn Z} (hF : F ∈ Reg V) (hu : ∀ z ∈ V, IsUnit (F z)) : inv F ∈ Reg V

theorem **IsLoc.le** {D W : Z.Opens} {F : Fn Z} (h : IsLoc D W F) : D ≤ W

theorem **IsLoc.unit** {D W : Z.Opens} {F : Fn Z} (h : IsLoc D W F) {z : Z} (hz : z ∈ D) : IsUnit (F z)

theorem **IsLoc.eq_basicOpen** {D W : Z.Opens} {F : Fn Z} (h : IsLoc D W F) {a : Γ(Z, W)} (ha : ∀ z (hz : z ∈ W), (Z.presheaf.germ W z hz).hom a = F z) : D = Z.basicOpen a

theorem **exists_loc** {W : Z.Opens} {F : Fn Z} (hF : F ∈ Reg W) : ∃ D, IsLoc D W F

theorem **IsLoc.affine** {D W : Z.Opens} {F : Fn Z} (h : IsLoc D W F) (hF : F ∈ Reg W) (hW : IsAffineOpen W) : IsAffineOpen D

theorem **kill** {D W : Z.Opens} {w F : Fn Z} (hW : IsCompact (W : Set Z)) (h : IsLoc D W w) (hw : w ∈ Reg W) (hF : F ∈ Reg W) (h0 : ∀ z ∈ D, F z = 0) : ∃ n : ℕ, ∀ z ∈ W, w z ^ n * F z = 0

theorem **frac** {D W : Z.Opens} {w F : Fn Z} (hW : IsAffineOpen W) (h : IsLoc D W w) (hw : w ∈ Reg W) (hF : F ∈ Reg D) : ∃ n : ℕ, ∃ G ∈ Reg W, ∀ z ∈ D, w z ^ n * F z = G z

theorem **Gen.mono** {D : Z.Opens} {S S' : Set (Fn Z)} (h : Gen D S) (hS : ∀ x ∈ S, ∃ b ∈ Subring.closure S', ∀ z ∈ D, x z = b z) : Gen D S'

theorem **Gen.loc** {D W : Z.Opens} {w : Fn Z} {S : Set (Fn Z)} (hW : IsAffineOpen W) (h : IsLoc D W w) (hw : w ∈ Reg W) (hS : Gen W S) : Gen D (insert (inv w) S)

theorem **local_global** {W : Z.Opens} (B : Subring (Fn Z)) (hB : B ≤ Reg W) {T : Type} (w : T → B) (hw : van B W ⊔ Ideal.span (Set.range w) = ⊤) (O : T → Z.Opens) (hO : ∀ t, IsLoc (O t) W (w t).1) (haff : ∀ t, IsAffineOpen (O t)) (hgen : ∀ t, Gen (O t) (insert (inv (w t).1) B)) : IsAffineOpen W ∧ Gen W B

theorem **exists_one_add_kill** {B : Type} [CommRing B] {T : Type} [Finite T] (u : T → B) (K : Ideal B) (h : ∀ t, ∃ e : ℕ, u t ^ e ∈ K ⊔ Ideal.span (Set.range u) ^ (e + 1)) : ∃ c, c - 1 ∈ Ideal.span (Set.range u) ∧ ∃ d : ℕ, ∀ t, c * u t ^ d ∈ K

theorem **Sc.mono** {V W : Z.Opens} {n : ℕ} {F : Fn Z} (h : Sc U G W n F) (hle : V ≤ W) : Sc U G V n F

theorem **Sc.mul** {W : Z.Opens} {n m : ℕ} {F F' : Fn Z} (h : Sc U G W n F) (h' : Sc U G W m F') : Sc U G W (n + m) (F * F')

theorem **Sc.pow** {W : Z.Opens} {n : ℕ} {F : Fn Z} (h : Sc U G W n F) (d : ℕ) : Sc U G W (n * d) (F ^ d)

theorem **Sc.zero** (W : Z.Opens) (n : ℕ) : Sc U G W n 0

theorem **Sc.exists_loc** {W : Z.Opens} {n : ℕ} {F : Fn Z} (h : Sc U G W n F) : ∃ D, IsLoc D W F

theorem **Sc.ratio** {D W : Z.Opens} {n : ℕ} {F σ : Fn Z} (h : Sc U G W n F) (hσ : Sc U G W n σ) (hD : IsLoc D W σ) : F * inv σ ∈ Reg D

theorem **Sc.extend** [CompactSpace Z] [QuasiSeparatedSpace Z] (hT : ∀ i j, G i * inv (G j) ∈ Reg (U i ⊓ U j)) {W : Z.Opens} {f : Fn Z} (hf : f ∈ Reg ⊤) (hW : IsLoc W ⊤ f) {n : ℕ} {F : Fn Z} (h : Sc U G W n F) : ∃ k : ℕ, ∃ σ, Sc U G ⊤ n σ ∧ ∀ z ∈ W, σ z = f z ^ k * F z

theorem **extra_section** {T : Type} [Finite T] {W₀ : Z.Opens} {N d : ℕ} {s : T → Fn Z} (hs : ∀ j, Sc U G W₀ N (s j)) {D : T → Z.Opens} (hD : ∀ j, IsLoc (D j) W₀ (s j)) (cov : ∀ z ∈ W₀, ∃ j, z ∈ D j) (aff : ∀ j, IsAffineOpen (D j)) {S₀ : Set (Fn Z)} (hS₀ : ∀ x ∈ S₀, x ∈ Reg W₀) (gen : ∀ j, Gen (D j) (S₀ ∪ Set.range fun m => s m * inv (s j))) {τ : Fn Z} (hτ : Sc U G W₀ (N * (d + 1)) τ) {W : Z.Opens} (hW : IsLoc W W₀ τ) : IsAffineOpen W ∧ Gen W (S₀ ∪ Set.range fun p : T × T => s p.2 * s p.1 ^ d * inv τ)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsPullbackCompIsoMonHom

theorem **exists_pullback_comp_iso_monHom** {T S' S : Scheme.{0}} (f : S' ⟶ S) (g : S ⟶ T) (h : S' ⟶ T) (hh : h = f ≫ g) (G : Over T) [GrpObj G] : ∃ e : (Over.pullback h).obj G ≅ (Over.pullback f).obj ((Over.pullback g).obj G), IsMonHom e.hom ∧ e.hom.left ≫ pullback.fst ((Over.pullback g).obj G).hom f ≫ pullback.fst G.hom g = pullback.fst G.hom h

theorem **charUnit_val_eq_appTop** (S : Scheme.{0}) (X : Type) [AddCommGroup X] (χ : X) : (charUnit S X χ).val = (pullback.fst (splitTorusAff ZU X).hom (toZ S)).appTop.hom ((Scheme.ΓSpecIso (torusRing ZU X)).inv.hom (AddMonoidAlgebra.single χ 1))

theorem **splitTorus_baseChange** (S S' : Scheme.{0}) (f : S' ⟶ S) (X : Type) [AddCommGroup X] : ∃ e : splitTorus S' X ≅ (Over.pullback f).obj (splitTorus S X), IsMonHom e.hom ∧ ∀ χ : X, (e.hom.left ≫ pullback.fst (splitTorus S X).hom f).appTop.hom (charUnit S X χ).val = (charUnit S' X χ).val

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsQuotientSchemeOfFiniteFlatSubgroup

lemma **exists_coaction** {R Λ : Type} [CommRing R] [CommRing Λ] [HopfAlgebra R Λ] {A : Over (Spec (CommRingCat.of R))} [GrpObj A] (i : specOver R Λ ⟶ A) {Y : Scheme.{0}} (f : A.left ⟶ Y) (g : (specOver R Λ ⊗ A).left ⟶ Y) (ha : ((fst (specOver R Λ) A ≫ i) * snd (specOver R Λ) A).left ≫ f = g) (hp : (snd (specOver R Λ) A).left ≫ f = g) (U : Y.Opens) (hf : IsAffineOpen (f ⁻¹ᵁ U)) (hg : IsAffineOpen (g ⁻¹ᵁ U)) : ∃ (_ : Algebra R Γ(A.left, f ⁻¹ᵁ U)) (v : specOver R Γ(A.left, f ⁻¹ᵁ U) ⟶ A) (_ : Mono v) (ρ : Γ(A.left, f ⁻¹ᵁ U) →ₐ[R] Γ(A.left, f ⁻¹ᵁ U) ⊗[R] Λ) (e : Γ((specOver R Λ ⊗ A).left, g ⁻¹ᵁ U) ⟶ .of (Γ(A.left, f ⁻¹ᵁ U) ⊗[R] Λ)) (_ : IsIso e), specHom ρ ≫ v = (specHom includeRight ≫ i) * (specHom includeLeft ≫ v) ∧ pullOver f g _ ha U ≫ e = CommRingCat.ofHom ρ.toRingHom ∧ pullOver f g _ hp U ≫ e = CommRingCat.ofHom (includeLeft : Γ(A.left, f ⁻¹ᵁ U) →ₐ[R] Γ(A.left, f ⁻¹ᵁ U) ⊗[R] Λ).toRingHom

lemma **exists_quotient_hopf** (hq : (∀ (R Λ B : Type) [CommRing R] [CommRing Λ] [HopfAlgebra R Λ] [Module.Finite R Λ] [Module.Projective R Λ] [CommRing B] [Algebra R B] (ρ : B →ₐ[R] TensorProduct R B Λ), (∀ b, TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

lemma **exists_hopfAlgebra** (R : Type) [CommRing R] [IsNoetherianRing R] (H : Over (Spec (CommRingCat.of R))) [GrpObj H] [IsFinite H.hom] [Flat H.hom] : ∃ (Λ : Type) (_ : CommRing Λ) (_ : HopfAlgebra R Λ) (_ : Module.Finite R Λ) (_ : Module.Projective R Λ) (e : specOver R Λ ≅ H), IsMonHom e.hom

theorem **exists_quotient_scheme_of_finite_flat_subgroup** : (∀ (R : Type) [CommRing R] [IsNoetherianRing R] (H : Over (Spec (CommRingCat.of R))) [GrpObj H] [IsCommMonObj H], IsFinite H.hom → Flat H.hom → ∃ N : ℕ, 1 ≤ N ∧ ↑ H ^ N = 1) → (∀ (R Λ B : Type)

[CommRing R] [CommRing Λ] [HopfAlgebra R Λ] [Module.Finite R Λ] [Module.Projective R Λ] [CommRing B] [Algebra R B] (ρ : B →ₐ[R] TensorProduct R B Λ), (∀ b, TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSecValueIsUnit

theorem **exists_section_mem** {K : Type} [Field K] [IsAlgClosed K] {Y : Scheme.{0}} (f : Y ⟶ Spec (CommRingCat.of K)) [LocallyOfFiniteType f] (W : Y.Opens) (hW : (W : Set Y).Nonempty) : ∃ σ : Spec (CommRingCat.of K) ⟶ Y, σ ≫ f = 𝟙 _ ∧ ∀ p, σ p ∈ W

theorem **translPt_eq** {T : Over S} (a : T ⟶ A) : translPt A a = (fst T A ≫ a) ∗ snd T A

theorem **ptSecT_translT** {T : Over S} (a x : T ⟶ A) : ptSecT A x ≫ translT A a = ptSecT A (a ∗ x)

theorem **exists_two_points** {K : Type} [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S) [LocallyOfFiniteType A.hom] [IsIntegral (Over.mk t ⊗ A).left] (D : (Over.mk t ⊗ A).left.Opens) (hD : (D : Set (Over.mk t ⊗ A).left).Nonempty) (x c : Over.mk t ⟶ A) : ∃ a : Over.mk t ⟶ A, (∀ p, (ptSecT A (a ∗ x)).left p ∈ D) ∧ ∀ p, (ptSecT A (a⁻¹ ∗ c)).left p ∈ D

theorem **unitAt_of_local** (hL : L.IsLB) {T : Over S} [Subsingleton T.left] {I : Type} [Fintype I] (φ : I → ((T ⊗ A).left ⟶ A.left)) (ε : I → ℤ) (p : T.left) (V : T.left.Opens) (hp : p ∈ V) (c : LB.Gen ((LB.LinComb L φ ε).pull ((fst T A).left ⁻¹ᵁ V).ι)) (hc : c.IsGen) (y : (T ⊗ A).left) : UnitAt L φ ε y

theorem **unitAt_transl** (hL : L.IsLB) {T : Over S} (θ : LB.Sec (LPow A L 1 T)) (k : (LPow A L 1 T).ι) (a : T ⟶ A) (y : (T ⊗ A).left) (hy : (translT A a).left y ∈ (T ⊗ A).left.basicOpen (θ.f k)) : UnitAt L (K

theorem **exists_value_one** (hL : L.IsLB) {T : Over S} [Subsingleton T.left] (n : ℤ) (x : T ⟶ A) (p : T.left) (h : UnitAt L (K

theorem **exists_sec_value_isUnit** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (S : Scheme.{0}) (P : PolAV S) (T : Over S) (x y : T ⟶ P.A) (p : T.left), ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSectionsSpanAffine

theorem **le_pre_top** (φ : X' ⟶ X) {V : X'.Opens} : V ≤ φ ⁻¹ᵁ ⊤

theorem **le_pre_inf** {φ : X' ⟶ X} {V : X'.Opens} {U U' : X.Opens} (h : V ≤ φ ⁻¹ᵁ U) (h' : V ≤ φ ⁻¹ᵁ U') : V ≤ φ ⁻¹ᵁ (U ⊓ U')

theorem **rs_rs** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (x : Γ(X, U)) : rs h₂ (rs h₁ x) = rs (h₂.trans h₁) x

theorem **rs_self** {U : X.Opens} (h : U ≤ U) (x : Γ(X, U)) : rs h x = x

theorem **rs_pa** (φ : X' ⟶ X) {U : X.Opens} {V W : X'.Opens} (h : V ≤ φ ⁻¹ᵁ U) (h₂ : W ≤ V) (x : Γ(X, U)) : rs h₂ (pa φ h x) = pa φ (h₂.trans h) x

theorem **pa_rs** (φ : X' ⟶ X) {U U' : X.Opens} {V : X'.Opens} (h₁ : U' ≤ U) (h : V ≤ φ ⁻¹ᵁ U') (x : Γ(X, U)) : pa φ h (rs h₁ x) = pa φ (h.trans (φ.preimage_mono h₁)) x

theorem **pa_pa** (φ : X' ⟶ X) (ψ : X'' ⟶ X') {U : X.Opens} {V : X'.Opens} {W : X''.Opens} (h : V ≤ φ ⁻¹ᵁ U) (h' : W ≤ ψ ⁻¹ᵁ V) (χ : X'' ⟶ X) (e : ψ ≫ φ = χ) (h'' : W ≤ χ ⁻¹ᵁ U) (x : Γ(X, U)) : pa ψ h' (pa φ h x) = pa χ h'' x

theorem **pa_id** {U V : X.Opens} (h : V ≤ (𝟙 X) ⁻¹ᵁ U) (x : Γ(X, U)) : pa (𝟙 X) h x = rs h x

theorem **pa_bijective** (φ : X' ⟶ X) [IsOpenImmersion φ] {U : X.Opens} {V : X'.Opens} (h : V ≤ φ ⁻¹ᵁ U) (h' : φ ⁻¹ᵁ U ≤ V) (hU : U ≤ φ.opensRange) : Function.Bijective (pa φ h)

theorem **eq_of_cover** {κ : Type} {U : X.Opens} (W : κ → X.Opens) (hW : ∀ a, W a ≤ U) (hc : U ≤ ⨆ a, W a) (x y : Γ(X, U)) (h : ∀ a, rs (hW a) x = rs (hW a) y) : x = y

theorem **glue_cover** {κ : Type} {U : X.Opens} (W : κ → X.Opens) (hW : ∀ a, W a ≤ U) (hc : U ≤ ⨆ a, W a) (x : ∀ a, Γ(X, W a)) (h : ∀ a b, rs (inf_le_left : W a ⊓ W b ≤ W a) (x a) = rs (inf_le_right : W a ⊓ W b ≤ W b) (x b)) : ∃ y : Γ(X, U), ∀ a, rs (hW a) y = x a

theorem **fun_extend** (φ : X' ⟶ X) [IsOpenImmersion φ] (f : Γ(X, ⊤)) (hφ : φ.opensRange = X.basicOpen f) {W : X.Opens} (hW : IsAffineOpen W) {W' : X'.Opens} (h : W' ≤ φ ⁻¹ᵁ W) (h' : φ ⁻¹ᵁ W ≤ W') (t : Γ(X', W')) : ∃ (N : ℕ) (x : Γ(X, W)), pa φ h x = pa φ (le_pre_top φ) f ^ N ∗ t

theorem **fun_kill** (φ : X' ⟶ X) [IsOpenImmersion φ] (f : Γ(X, ⊤)) (hφ : φ.opensRange = X.basicOpen f) {W : X.Opens} (hW : IsCompact (W : Set X)) {W' : X'.Opens} (h : W' ≤ φ ⁻¹ᵁ W) (h' : φ ⁻¹ᵁ W ≤ W') (x : Γ(X, W)) (hx : pa φ h x = 0) : ∃ N : ℕ, rs (le_top : W ≤ ⊤) f ^ N ∗ x = 0

theorem **pow_mul_zero_mono** {R : Type} [CommRing R] {a x : R} {m n : ℕ} (h : a ^ m ∗ x = 0) (hmn : m ≤ n) : a ^ n ∗ x = 0

theorem **exists_fin_cover** [CompactSpace X] (M : LB X) (hM : M.IsLB) : ∃ (κ : Type) (_ : Fintype κ) (W : κ → X.Opens) (c : κ → M.ι), (∀ a, IsAffineOpen (W a)) ∧ (∀ a, W a ≤ M.U (c a)) ∧ ⊤ ≤ ⨆ a, W a

theorem **sec_at** {M : LB X} (s : LB.Sec M) (i j : M.ι) {W : X.Opens} (hi : W ≤ M.U i) (hj : W ≤ M.U j) : rs hi (s.f i) = rs (le_inf hi hj) (M.g i j : Γ(X, M.U i ⊓ M.U j)) ∗ rs hj (s.f j)

theorem **cocycle_at** {M : LB X} (hM : M.IsLB) (i j k : M.ι) {W : X.Opens} (h₁ : W ≤ M.U i ⊓ M.U j) (h₂ : W ≤ M.U j ⊓ M.U k) (h₃ : W ≤ M.U i ⊓ M.U k) : rs h₁ (M.g i j : Γ(X, M.U i ⊓ M.U j)) ∗ rs h₂ (M.g j k : Γ(X, M.U j ⊓ M.U k)) = rs h₃ (M.g i k : Γ(X, M.U i ⊓ M.U k))

theorem **le_iSup_inf** {κ : Type} (U : X.Opens) {W : κ → X.Opens} (hc : ⊤ ≤ ⨆ a, W a) : U ≤ ⨆ a, U ⊓ W a

theorem **sec_kill** [CompactSpace X] (φ : X' ⟶ X) [IsOpenImmersion φ] (f : Γ(X, ⊤)) (hφ : φ.opensRange = X.basicOpen f) (M : LB X) (hM : M.IsLB) (U' : M.ι → X'.Opens) (h : ∀ j, U' j ≤ φ ⁻¹ᵁ M.U j) (h' : ∀ j, φ ⁻¹ᵁ M.U j ≤ U' j) (s : LB.Sec M) (hs : ∀ j, pa φ (h j) (s.f j) = 0) : ∃ N : ℕ, ∀ j, rs (le_top : M.U j ≤ ⊤) f ^ N ∗ s.f j = 0

theorem **sec_extend** [CompactSpace X] [QuasiSeparatedSpace X] (φ : X' ⟶ X) [IsOpenImmersion φ] (f : Γ(X, ⊤)) (hφ : φ.opensRange = X.basicOpen f) (M : LB X) (hM : M.IsLB) (U' : M.ι → X'.Opens) (h : ∀ j, U' j ≤ φ ⁻¹ᵁ M.U j) (h' : ∀ j, φ ⁻¹ᵁ M.U j ≤ U' j) (t : ∀ j, Γ(X', U' j)) (ht : ∀ i j, rs (inf_le_left : U' i ⊓ U' j ≤ U' i) (t i) = pa φ (le_pre_inf (inf_le_left.trans (h i)) (inf_le_right.trans (h j))) (M.g i j : Γ(X, M.U i ⊓ M.U j)) ∗ rs (inf_le_right : U' i ⊓ U' j ≤ U' j) (t j)) : ∃ (N : ℕ) (s : LB.Sec M), ∀ j, pa φ (h j) (s.f j) = pa φ (le_pre_top φ) f ^ N ∗ t j

theorem **pa_linComb_g** {X Y Z : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ψ : K → (Z ⟶ X)) (ε : K → ℤ) (h : Z ⟶ Y) (hψ : ∀ k, h ≫ φ k = ψ k) (i j : K → L.ι) (e : LB.wchart L ψ i ⊓ LB.wchart L ψ j ≤ h ⁻¹ᵁ (LB.wchart L φ i ⊓ LB.wchart L φ j)) : pa h e (LB.lcg L φ ε i j : Γ(Y, LB.wchart L φ i ⊓ LB.wchart L φ j)) = (LB.lcg L ψ ε i j : Γ(Z, LB.wchart L ψ i ⊓ LB.wchart L ψ j))

theorem **map_mem_span** {R R' M M' : Type} [CommRing R] [CommRing R'] [AddCommGroup M] [AddCommGroup M'] [Module R M] [Module R' M'] {σ : R →+* R'} (F : M →ₛₗ[σ] M') {κ : Type} (b : κ → M) (s : M) (hs : s ∈ Submodule.span R (Set.range b)) : F s ∈ Submodule.span R' (Set.range fun i => F (b i))

theorem **pow_smul_mem_of** {R R' M M' : Type} [CommRing R] [CommRing R'] [AddCommGroup M] [AddCommGroup M'] [Module R M] [Module R' M'] {σ : R →+* R'} (F : M →ₛₗ[σ] M') (r : R) (h₁ : ∀ c : R', ∃ (n : ℕ) (d : R), σ d = σ r ^ n ∗ c) (h₂ : ∀ s : M, F s = 0 → ∃ n : ℕ, r ^ n • s

= 0) (W : Submodule R M) (s : M) (hs : F s ∈ Submodule.span R' (F '' W)) : ∃ N : ℕ, r ^ N • s ∈ W

theorem **chart_le** (h : T' ⟶ T) (j : Fin 1 → L.ι) : (LPow A L a T').U j ≤ (h ▷ A).left ⁻¹ᵁ (LPow A L a T).U j

theorem **whisker_snd_left** (h : T' ⟶ T) : (h ▷ A).left ≫ (snd T A).left = (snd T' A).left

theorem **whisker_fst_left** (h : T' ⟶ T) : (h ▷ A).left ≫ (fst T A).left = (fst T' A).left ≫ h.left

theorem **chart_ge** (h : T' ⟶ T) (j : Fin 1 → L.ι) : (h ▷ A).left ⁻¹ᵁ (LPow A L a T).U j ≤ (LPow A L a T').U j

theorem **pa_g** (h : T' ⟶ T) (i j : Fin 1 → L.ι) (e : (LPow A L a T').U i ⊓ (LPow A L a T').U j ≤ (h ▷ A).left ⁻¹ᵁ ((LPow A L a T).U i ⊓ (LPow A L a T).U j)) : pa (h ▷ A).left e ((LPow A L a T).g i j : Γ((T ⊗ A).left, (LPow A L a T).U i ⊓ (LPow A L a T).U j)) = ((LPow A L a T').g i j : Γ((T' ⊗ A).left, (LPow A L a T').U i ⊓ (LPow A L a T').U j))

theorem **liftHom_eq** (W : (T ⊗ A).left.Opens) (r : Γ(T.left, ⊤)) : liftHom A W r = pa (fst T A).left (le_pre_top _) r

theorem **pa_liftHom** (h : T' ⟶ T) {W : (T ⊗ A).left.Opens} {W' : (T' ⊗ A).left.Opens} (e : W' ≤ (h ▷ A).left ⁻¹ᵁ W) (r : Γ(T.left, ⊤)) : pa (h ▷ A).left e (liftHom A W r) = liftHom A W' (pa h.left (le_pre_top _) r)

theorem **pull_apply** (h : T' ⟶ T) (s : SecM A L a T) (j : Fin 1 → L.ι) : (pull A L a h s).1 j = pa (h ▷ A).left (chart_le A L a h j) (s.1 j)

theorem **pull_pull** (h : T' ⟶ T) (k : T'' ⟶ T') (s : SecM A L a T) : pull A L a k (pull A L a h s) = pull A L a (k ≫ h) s

theorem **pull_id** (s : SecM A L a T) : pull A L a (𝟙 T) s = s

theorem **isPullSec_iff** (h : T' ⟶ T) (s : LB.Sec (LPow A L a T)) (s' : LB.Sec (LPow A L a T')) : IsPullSec A L a (h ▷ A) (whiskerRight_snd h A) s s' ↔ SecM.ofSec A s' = pull A L a h (SecM.ofSec A s)

theorem **isPullback_whisker** (h : T' ⟶ T) : IsPullback (h ▷ A).left (fst T' A).left (fst T A).left h.left

theorem **whisker_open** (h : T' ⟶ T) [IsOpenImmersion h.left] : IsOpenImmersion (h ▷ A).left

theorem **whisker_range** (h : T' ⟶ T) [IsOpenImmersion h.left] [IsOpenImmersion (h ▷ A).left] (r : Γ(T.left, ⊤)) (hr : h.left.opensRange = T.left.basicOpen r) : (h ▷ A).left.opensRange = (T ⊗ A).left.basicOpen (pa (fst T A).left (V

theorem **qcqs** [IsProper A.hom] [IsAffine T.left] : CompactSpace (T ⊗ A).left ∧ QuasiSeparatedSpace (T ⊗ A).left

theorem **pull_kill** [IsProper A.hom] [IsAffine T.left] (hL : L.IsLB) (h : T' ⟶ T) [IsOpenImmersion h.left] (r : Γ(T.left, ⊤)) (hr : h.left.opensRange = T.left.basicOpen r) (s : SecM A L a T) (hs : pull A L a h s = 0) : ∃ N : ℕ, r ^ N • s = 0

theorem **pull_extend** [IsProper A.hom] [IsAffine T.left] (hL : L.IsLB) (h : T' ⟶ T) [IsOpenImmersion h.left] (r : Γ(T.left, ⊤)) (hr : h.left.opensRange = T.left.basicOpen r) (t : SecM A L a T') : ∃ (N : ℕ) (s : SecM A L a T), pull A L a h s = pa h.left (V

theorem **pow_smul_mem** [IsProper A.hom] [IsAffine T.left] (hL : L.IsLB) (h : T' ⟶ T) [IsOpenImmersion h.left] (r : Γ(T.left, ⊤)) (hr : h.left.opensRange = T.left.basicOpen r) (W : Submodule Γ(T.left, ⊤) (SecM A L a T)) (s : SecM A L a T) (hs : pull A L a h s ∈ Submodule.span Γ(T'.left, ⊤) (pull A L a h '' W)) : ∃ N : ℕ, r ^ N • s ∈ W

theorem **exists_lift** (h : T' ⟶ T) (V : T.left.Opens) (hV : ∀ x, h.left x ∈ V) : ∃ k : T' ⟶ overRes V, k ≫ overResι V = h

theorem **overResι_range** (V : T.left.Opens) : (overResι V).left.opensRange = V

theorem **isUnit_pa** (h : T' ⟶ T) (r : Γ(T.left, ⊤)) (hr : ∀ x, h.left x ∈ T.left.basicOpen r) : IsUnit (pa h.left (V

theorem **span_of_retract** (e : T'' ⟶ T') (e' : T' ⟶ T'') (he : e' ≫ e = 𝟙 T') (k : T'' ⟶ T) {κ : Type} (b : κ → SecM A L a T) (hb : Submodule.span Γ(T''.left, ⊤) (Set.range fun i => pull A L a k (b i)) = ⊤) : Submodule.span Γ(T'.left, ⊤) (Set.range fun i => pull A L a (e' ≫ k) (b i)) = ⊤

theorem **exists_spec_retract** {V : S.Opens} [IsAffine T'.left] (g : T' ⟶ overRes (T

theorem **local_span** (H : SecBaseChangeSt) (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ) (hn : 1 ≤ n) (p : S) : ∃ V : S.Opens, p ∈ V ∧ ∃ b : Fin (n ^ P.g) → SecM P.A P.L n (overRes (T

theorem **GlobalTheory.exists_sections_span_affine** : SecBaseChangeSt → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → ∀ U : S.Opens, IsAffineOpen U → ∃ (m : ℕ) (b : Fin (m + 1) → LB.Sec (LPow P.A P.L n (overRes (T

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSplitExtDatumOfSplitTorusExtension

theorem **gm_comp** (κ : Z ⟶ Y) (f : Y ⟶ X) (z : Z) (U : X.Opens) (h : f (κ z) ∈ U) (a : Γ(X, U)ˣ) : gm (κ ≫ f) z U a = Units.map (κ.stalkMap z).hom.toMonoidHom (gm f (κ z) U a)

theorem **gm_res_pull** (r : Z ⟶ Y) (f : Y ⟶ X) (z : Z) {U : X.Opens} {V : Y.Opens} (hle : V ≤ f ⁻¹ᵁ U) (h : r z ∈ V) (a : Γ(X, U)ˣ) : gm r z V (res hle (pullU f U a)) = gm (r ≫ f) z U a

theorem **gm_res_pre** (r : Z ⟶ Y) (f : Y ⟶ X) (z : Z) {U : Y.Opens} {V : X.Opens} (hle : f ⁻¹ᵁ V ≤ U) (h : f (r z) ∈ V) (a : Γ(Y, U)ˣ) : gm r z (f ⁻¹ᵁ V) (res hle a) = gm r z U a

theorem **gm_id_res_pull** (f : Y ⟶ X) (y : Y) {U : X.Opens} {V : Y.Opens} (hle : V ≤ f ⁻¹ᵁ U) (h : y ∈ V) (a : Γ(X, U)ˣ) : gm (𝟙 Y) y V (res hle (pullU f U a)) = gm f y U a

theorem **gm_id_pull'** (f : Y ⟶ X) (y : Y) {U : X.Opens} (h : f y ∈ U) (a : Γ(X, U)ˣ) : gm (𝟙 Y) y (f ⁻¹ᵁ U) (pullU f U a) = gm f y U a

theorem **unit_ext_of_cover** (π : Y ⟶ X) [Flat π] [Surjective π] {U : X.Opens} {a b : Γ(X, U)ˣ} (h : ∀ y, π y ∈ U → gm π y U a = gm π y U b) : a = b

theorem **exists_unit_descent** (U : X.left.Opens) (f : Γ(Y.left, π.left ⁻¹ᵁ U)ˣ) (hf : ∀ (Z : Over S) (g₁ g₂ : Z ⟶ Y) (z : Z.left), g₁ ≫ π = g₂ ≫ π → π.left (g₁.left z) ∈ U → π.left (g₂.left z) ∈ U → gm g₁.left z _ f = gm g₂.left z _ f) : ∃ a : Γ(X.left, U)ˣ, pullU π.left U a = f

theorem **exists_gen_descent** (M : LB X.left) (f : ∀ j, Γ(Y.left, π.left ⁻¹ᵁ M.U j)ˣ) (hf : ∀ j (Z : Over S) (g₁ g₂ : Z ⟶ Y) (z : Z.left), g₁ ≫ π = g₂ ≫ π → π.left (g₁.left z) ∈ M.U j → π.left (g₂.left z) ∈ M.U j → gm g₁.left z _ (f j) = gm g₂.left z _ (f j)) (hg : ∀ j j' (y : Y.left), π.left y ∈ M.U j → π.left y ∈ M.U j' → gm (𝟙 _) y _ (f j) = gm π.left y (M.U j ⊓ M.U j') (M.g j j') * gm (𝟙 _) y _ (f j')) : ∃ t : LB.Gen M, t.IsGen

theorem **charGerm_eq** {Z : Over S} (z : Z.left) (t : Z ⟶ splitTorus S X) (χ : X) : charGerm z t (Multiplicative.ofAdd χ) = gm t.left z ⊤ (charUnit S X χ)

theorem **act_comp** {Z : Over S} (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) : ((t ≫ i) * g) ≫ q = g ≫ q

theorem **act_apply** {Z : Over S} (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (z : Z.left) : q.left (((t ≫ i) * g).left z) = q.left (g.left z)

theorem **exists_act** {Z : Over S} {g₁ g₂ : Z ⟶ G} (e : g₁ ≫ q = g₂ ≫ q) : ∃ t : Z ⟶ splitTorus S X, g₁ = (t ≫ i) * g₂

theorem **HasWeight.appLE** {χ : X} {V : B.left.Opens} {u : Γ(G.left, q.left ⁻¹ᵁ V)ˣ} (hu : HasWeight q i χ V u) (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (h₁ : ⊤ ≤ ((t ≫ i) * g).left ⁻¹ᵁ (q.left ⁻¹ᵁ V)) (h₂ : ⊤ ≤ g.left ⁻¹ᵁ (q.left ⁻¹ᵁ V)) : (((t ≫ i) *

g).left.appLE _ ⊤ h₁).hom u.val = t.left.appTop.hom (charUnit S X χ).val ∗ (g.left.appLE _ ⊤ h₂).hom u.val

theorem **hasWeight_of_appLE** {χ : X} {V : B.left.Opens} {u : Γ(G.left, q.left ⁻¹ᵁ V)ˣ} (hu : ∀ (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (h₁ : ⊤ ≤ ((t ≫ i) ∗ g).left ⁻¹ᵁ (q.left ⁻¹ᵁ V)) (h₂ : ⊤ ≤ g.left ⁻¹ᵁ (q.left ⁻¹ᵁ V)), (((t ≫ i) ∗ g).left.appLE _ ⊤ h₁).hom u.val = t.left.appTop.hom (charUnit S X χ).val ∗ (g.left.appLE _ ⊤ h₂).hom u.val) : HasWeight q i χ V u

theorem **psum_eq_prod** (G : Over S) [GrpObj G] {T : Over S} {n : ℕ} (x : Fin n → (T ⟶ G)) (I : Finset (Fin n)) : psum G x I = (((List.finRange n).filter (· ∈ I)).map x).prod

theorem **map_psum** {G G' : Over S} [GrpObj G] [GrpObj G'] {T T' : Over S} (F : (T ⟶ G) →∗ (T' ⟶ G')) {n : ℕ} (x : Fin n → (T ⟶ G)) (I : Finset (Fin n)) : F (psum G x I) = psum G' (fun k => F (x k)) I

theorem **comp_psum** (G : Over S) [GrpObj G] {T T' : Over S} (r : T' ⟶ T) {n : ℕ} (x : Fin n → (T ⟶ G)) (I : Finset (Fin n)) : r ≫ psum G x I = psum G (fun k => r ≫ x k) I

theorem **psum_comp** {G G' : Over S} [GrpObj G] [GrpObj G'] (f : G ⟶ G') [IsMonHom f] {T : Over S} {n : ℕ} (x : Fin n → (T ⟶ G)) (I : Finset (Fin n)) : psum G x I ≫ f = psum G' (fun k => x k ≫ f) I

theorem **psum_mul** (G : Over S) [GrpObj G] [IsCommMonObj G] {T : Over S} {n : ℕ} (x y : Fin n → (T ⟶ G)) (I : Finset (Fin n)) : psum G (fun k => x k ∗ y k) I = psum G x I ∗ psum G y I

theorem **prod_square_eq_one** {G : Over S} [GrpObj G] {T : Over S} {A : Type} [CommGroup A] (c : (T ⟶ G) →∗ A) (x : Fin 2 → (T ⟶ G)) : ∏ I : ULift.{0} (Finset (Fin 2)), c (psum G x I.down) ^ ((-1 : ℤ) ^ (2 - I.down.card)) = 1

theorem **monoidHom_ext_basis** {X K A : Type} [AddCommGroup X] [CommGroup A] (b : Module.Basis K ℤ X) {F F' : Multiplicative X →∗ A} (h : ∀ k, F (Multiplicative.ofAdd (b k)) = F' (Multiplicative.ofAdd (b k))) : F = F'

theorem **inv_eq_div_mul_inv** {A : Type} [CommGroup A] (a b : A) : a⁻¹ = b / a ∗ b⁻¹

theorem **eSec_comp** : eSec G ≫ q.left = eSec B

theorem **isAffineHom_left** [Flat q.left] [Surjective q.left] [QuasiCompact q.left] : IsAffineHom q.left

theorem **exists_isFrame** (χ : X) : ∃ (O : RigLB B) (u : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O.L.U j)ˣ), IsFrame q i χ O u

theorem **aux_add** {A : Type} [CommGroup A] (a a' b b' c c' : A) : a⁻¹ ∗ b ∗ c = a' / a ∗ (b' / b)⁻¹ ∗ (c' / c)⁻¹ ∗ (a'⁻¹ ∗ b' ∗ c')

theorem **prod_erase_one_perm_apply_eq_zero** {A K : Type} [CommGroup A] [Fintype K] (c a : K → A) (ε : K → ℤ) (h : ∏ I, c I ^ ε I = 1) : ∏ I, (c I ∗ a I)⁻¹ ^ ε I = ∏ I, (a I)⁻¹ ^ ε I

theorem **commGroup_weight_cancel_identity** {A : Type} [CommGroup A] (x y a b c : A) : (x ∗ y ∗ a)⁻¹ ∗ (x ∗ b) ∗ (y ∗ c) = a⁻¹ ∗ b ∗ c

theorem **IsFrame.trans_id** {χ : X} {O : RigLB B} {u : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O.L.U j)ˣ} (h : IsFrame q i χ O u) (j j' : O.L.ι) (y : G.left) (h₁ : q.left y ∈ O.L.U j) (h₂ : q.left y ∈ O.L.U j') : gm q.left y (O.L.U j ⊓ O.L.U j') (O.L.g j j') = gm (𝟙 _) y _ (u j') / gm (𝟙 _) y _ (u j)

theorem **IsFrame.exists_gen_zero** {O : RigLB B} {u : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O.L.U j)ˣ} (h : IsFrame q i 0 O u) : ∃ t : LB.Gen O.L, t.IsGen

theorem **IsFrame.exists_iso_add** {χ χ' : X} {O O' O'' : RigLB B} {u : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O.L.U j)ˣ} {u' : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O'.L.U j)ˣ} {u'' : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O''.L.U j)ˣ} (h : IsFrame q i χ O u) (h' : IsFrame q i χ' O' u') (h'' : IsFrame q i (χ + χ') O'' u'') : ∃ t : LB.Iso O'.L (LB.div O''.L O.L), t.IsGen

theorem **cover_tensorHom** : (@Surjective ⊓ @Flat ⊓ @QuasiCompact : MorphismProperty Scheme.{0}) (q ⊗ₘ q).left

theorem **IsFrame.exists_gen_linComb** {χ : X} {O : RigLB B} {u : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O.L.U j)ˣ} (h : IsFrame q i χ O u) {Y Y' : Over S} (π : Y' ⟶ Y) [Flat π.left] [Surjective π.left] [QuasiCompact π.left] {K : Type} [Fintype K] (φ : K → (Y.left ⟶ B.left)) (ψ : K → (Y' ⟶ G)) (ε : K → ℤ) (hψ : ∀ I, (ψ I).left ≫ q.left = π.left ≫ φ I) (hinv : ∀ (Z : Over S) (g₁ g₂ : Z ⟶ Y') (z : Z.left), g₁ ≫ π = g₂ ≫ π → ∃ t : K → (Z ⟶ splitTorus S X), (∀ I, g₁ ≫ ψ I = (t I ≫ i) ∗ (g₂ ≫ ψ I)) ∧ ∏ I, charGerm z (t I) (Multiplicative.ofAdd χ) ^ ε I = 1) : ∃ t : LB.Gen (LB.LinComb O.L φ ε), t.IsGen

theorem **IsFrame.exists_gen_lam** [IsCommMonObj G] {χ : X} {O : RigLB B} {u : ∀ j, Γ(G.left, q.left ⁻¹ᵁ O.L.U j)ˣ} (h : IsFrame q i χ O u) : ∃ t : LB.Gen (abelian₂ B O.L), t.IsGen

theorem **exists_gtDatum** (u : ∀ χ j, Γ(G.left, q.left ⁻¹ᵁ (E.O χ).L.U j)ˣ) (hu : ∀ χ, IsFrame q i χ (E.O χ) (u χ)) : ∃ d : E.GtDatum G, d.q = q ∧ d.i = i

theorem **exists_gtDatum_of_splitTorus_extension** : ∀ (S : Scheme.{0}) (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (G B : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj B], IsAbelianScheme B → ∀ (q : G ⟶ B) (i : splitTorus S X ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → ∃ (O : X → RigLB B) (h₁ : _) (h₂ : _) (h₃ : _), ∀ hB : IsAbelianScheme B, ∃ d : (SplitExt.mk B hB X O h₁ h₂ h₃).GtDatum G, d.q = q ∧ d.i = i

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSplitExtDatumOfSplitTorusExtensionStatement

theorem **_root_.GlobalTheory.T.exists_splitExtDatum_of_splitTorus_extension** : ∀ (S : Scheme.{0}) (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (G B : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj B], IsAbelianScheme B → ∀ (q : G ⟶ B) (i : splitTorus S X ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → ∃ (O : X → RigLB B) (h₁ : _) (h₂ : _) (h₃ : _), ∀ hB : IsAbelianScheme B, ∃ d : (SplitExt.mk B hB X O h₁ h₂ h₃).GtDatum G, d.q = q ∧ d.i = i

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSplitExtIsGtOfTorusExtension

theorem **exists_splitExt_isGt_of_torusExtension** : ∀ (K : Type) [Field K] (G : Over (Spec (CommRingCat.of K))) [GrpObj G] [IsCommMonObj G], Nonempty (TorusExtension G) → ∃ E : SplitExt (Spec (CommRingCat.of K)), E.IsGt G

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSplitTorusPointsMulEquiv

theorem **ΓSpecIso_inv_comp_specMap** {W : Scheme.{0}} {A B : CommRingCat.{0}} (g : W ⟶ Spec B) (f : A ⟶ B) : (Scheme.ΓSpecIso A).inv ≫ (g ≫ Spec.map f).appTop = f ≫ (Scheme.ΓSpecIso B).inv ≫ g.appTop

theorem **map_left_fst** {T : Scheme.{0}} (f : S ⟶ T) {A B : Over T} (g : A ⟶ B) : ((Over.pullback f).map g).left ≫ pullback.fst B.hom f = pullback.fst A.hom f ≫ g.left

theorem **exists_mul_factor** (a b : Z ⟶ splitTorus S X) : ∃ q : Z.left ⟶ (splitTorusAff ZU X ⊗ splitTorusAff ZU X).left, q ≫ (fst _ _ : _ ⟶ splitTorusAff ZU X).left = a.left ≫ torusFst S X ∧ q ≫ (snd _ _ : _ ⟶ splitTorusAff ZU X).left = b.left ≫ torusFst S X ∧ q ≫ μ[splitTorusAff ZU X].left = (a ∗ b).left ≫ torusFst S X

theorem **coord_mul_single** (a b : Z ⟶ splitTorus S X) (x : X) : (coord (a ∗ b)).hom (AddMonoidAlgebra.single x 1) = (coord a).hom (AddMonoidAlgebra.single x 1) ∗ (coord b).hom (AddMonoidAlgebra.single x 1)

theorem **coord_single** (t : Z ⟶ splitTorus S X) (x : X) : (coord t).hom (AddMonoidAlgebra.single x 1) = t.left.appTop.hom (charUnit S X x).val

theorem **toHom_val** (t : Z ⟶ splitTorus S X) (x : X) : (toHom t (Multiplicative.ofAdd x)).val = (coord t).hom (AddMonoidAlgebra.single x 1)

theorem **exists_coord_eq** (φ : torusRing ZU X ⟶ Γ(Z.left, ⊤)) : ∃ t : Z ⟶ splitTorus S X, coord t = φ

theorem **toHom_bijective** : Function.Bijective (toHom : (Z ⟶ splitTorus S X) → _)

theorem **toHom_mul** (a b : Z ⟶ splitTorus S X) : toHom (a ∗ b) = toHom a ∗ toHom b

theorem **splitTorusPointsEquiv_val** (S : Scheme.{0}) (X : Type) [AddCommGroup X] (Z : Over S) (t : Z ⟶ splitTorus S X) (χ : X) : (splitTorusPointsEquiv S X Z t (Multiplicative.ofAdd χ)).val = t.left.appTop.hom (charUnit S X χ).val

theorem **exists_splitTorus_points_mulEquiv** : ∀ (S : Scheme.{0}) (X : Type) [AddCommGroup X] (Z : Over S), ∃ e : (Z ⟶ splitTorus S X) ≃∗ (Multiplicative X →∗ Γ(Z.left, ⊤)ˣ), ∀ (t : Z ⟶ splitTorus S X) (χ : X), (e t (Multiplicative.ofAdd χ)).val = t.left.appTop.hom (charUnit S X χ).val

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsSplitsOfIsAffineOpen

@[simp] theorem **rs_rs** {U V W : Z.Opens} (h : U ≤ V) (h' : V ≤ W) (x : Γ(Z, W)) : rs h (rs h' x) = rs (h.trans h') x

@[simp] theorem **rs_self** {U : Z.Opens} (h : U ≤ U) (x : Γ(Z, U)) : rs h x = x

@[simp] theorem **appLE_rs** {T : Scheme.{0}} (f : T ⟶ Z) {U V : Z.Opens} (h : U ≤ V) {N : T.Opens} (e : N ≤ f ⁻¹ᵁ U) (x : Γ(Z, V)) : (f.appLE U N e).hom (rs h x) = (f.appLE V N (e.trans (f.preimage_mono h))).hom x

@[simp] theorem **rs_appLE** {T : Scheme.{0}} (f : T ⟶ Z) {U : Z.Opens} {N M : T.Opens} (h : M ≤ N) (e : N ≤ f ⁻¹ᵁ U) (x : Γ(Z, U)) : rs h ((f.appLE U N e).hom x) = (f.appLE U M (h.trans e)).hom x

theorem **glue** {J : Type} (O : Z.Opens) (W : J → Z.Opens) (hW : ∀ j, W j ≤ O) (hc : O ≤ iSup W) (sf : ∀ j, Γ(Z, W j)) (h : ∀ j k, rs (inf_le_left : W j ⊓ W k ≤ W j) (sf j) = rs inf_le_right (sf k)) : ∃ s : Γ(Z, O), ∀ j, rs (hW j) s = sf j

theorem **loc** {J : Type} {N : Z.Opens} (W : J → Z.Opens) (hc : N ≤ iSup W) (x y : Γ(Z, N)) (h : ∀ j, rs (inf_le_left : N ⊓ W j ≤ N) x = rs inf_le_left y) : x = y

theorem **exists_isDiff** (hV : ⊔ a, V a = ⊤) {O O' P : Z.Opens} {ψ : ∀ a, Γ(Z, O ⊓ V a)} {ψ' : ∀ a, Γ(Z, O' ⊓ V a)} (h : Splits V φ O ψ) (h' : Splits V φ O' ψ') (hO : P ≤ O) (hO' : P ≤ O') : ∃ c : Γ(Z, P), IsDiff V ψ ψ' hO hO' c

theorem **splits_glue** {J : Type} (O : J → Z.Opens) (ψ : ∀ j a, Γ(Z, O j ⊓ V a)) (h : ∀ j, Splits V φ (O j) (ψ j)) (g : ∀ j, Γ(Z, O j)) (hg : ∀ j k a (M : Z.Opens) (hj : M ≤ O j) (hk : M ≤ O k) (ha : M ≤ V a), rs (le_inf hj ha) (ψ j a) − rs (le_inf hk ha) (ψ k a) = rs hj (g j) − rs hk (g k)) (O₀ : Z.Opens) (hc : O₀ ≤ iSup O) : ∃ χ, Splits V φ O₀ χ

theorem **pull_coboundary** (hV : ⊔ a, V a = ⊤) {T : Scheme.{0}} (f : T ⟶ Z) (ρ : ∀ a, Γ(T, f ⁻¹ᵁ V a)) (hρ : ∀ a b, (f.app (V a ⊓ V b)).hom (φ a b) = rs (f.preimage_mono inf_le_left) (ρ a) − rs (f.preimage_mono inf_le_right) (ρ b)) {J : Type} (O : J → Z.Opens) (ψ : ∀ j a, Γ(Z, O j ⊓ V a)) (h : ∀ j, Splits V φ (O j) (ψ j)) (N : J → T.Opens) (hN : ∀ j, N j ≤ f ⁻¹ᵁ O j) : ∃ x : ∀ j, Γ(T, N j), ∀ j k (P : Z.Opens) (hj : P ≤ O j) (hk : P ≤ O k) (c : Γ(Z, P)), IsDiff V (ψ j) (ψ k) hj hk c → ∀ (M : T.Opens) (hMj : M ≤ N j) (hMk : M ≤ N k) (hM : M ≤ f ⁻¹ᵁ P), (f.appLE P M hM).hom c = rs hMj (x j) − rs hMk (x k)

theorem **exists_splits_of_isAffineOpen** (hV : ⊔ a, V a = ⊤) (hφ : IsCocycle V φ) {O : Z.Opens} (hO : IsAffineOpen O) : ∃ ψ, Splits V φ O ψ

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsUniqueCharacterOfMul

lemma **coeff_inl_mul_inr** (u v : AddMonoidAlgebra A X) (x y : X) : (AddMonoidAlgebra.mapDomainRingHom A (AddMonoidHom.inl X X) u ∗ AddMonoidAlgebra.mapDomainRingHom A (AddMonoidHom.inr X X) v).coeff (x, y) = u.coeff x ∗ v.coeff y

lemma **coeff_diag** (u : AddMonoidAlgebra A X) (x y : X) [Decidable (x = y)] : (AddMonoidAlgebra.mapDomainRingHom A ((AddMonoidHom.id X).prod (AddMonoidHom.id X)) u).coeff (x, y) = if x = y then u.coeff x else 0

lemma **IsMultiplicative.eq_single** {u : AddMonoidAlgebra A X} (hu : IsMultiplicative u) {x : X} (hx : IsUnit (u.coeff x)) : u = single x 1

lemma **exists_coeff_notMem** {u : AddMonoidAlgebra A X} (hu : IsUnit u) (p : Ideal A) (hp : p ≠ ⊤) : ∃ x, u.coeff x ∉ p

lemma **eq_of_single_one_eq** [Nontrivial A] {x y : X} (h : (single x 1 : AddMonoidAlgebra A X) = single y 1) : x = y

lemma **torusPoint_fst** {R : CommRingCat.{0}} (φ : torusRing ZU X ⟶ R) (s : Spec R ⟶ S) : (torusPoint φ s).left ≫ torusFst S X = Spec.map φ

lemma **torus_hom_ext** {Z : Over S} {g g' : Z ⟶ splitTorus S X} (h : g.left ≫ torusFst S X = g'.left ≫ torusFst S X) : g = g'

lemma **pullback_map_fst** {A B : Over (Spec ZU.{0})} (f : A ⟶ B) : ((Over.pullback (toZ S)).map f).left ≫ pullback.fst B.hom (toZ S) = pullback.fst A.hom (toZ S) ≫ f.left

lemma **torusPoint_mul** {R : CommRingCat.{0}} (φ φ' ψ : torusRing ZU X ⟶ R) (s : Spec R ⟶ S) (h : ∀ x : X, ψ.hom (single x 1) = φ.hom (single x 1) ∗ φ'.hom (single x 1)) : torusPoint φ s ∗ torusPoint φ' s = torusPoint ψ s

lemma **prodPoint_fst** {R : CommRingCat.{0}} (φ : torusRing ZU X ⟶ R) (y : Spec R ⟶ Y.left) : prodPoint φ y ≫ (fst (splitTorus S X) Y).left ≫ torusFst S X = Spec.map φ

lemma **prodPoint_snd** {R : CommRingCat.{0}} (φ : torusRing ZU X ⟶ R) (y : Spec R ⟶ Y.left) : prodPoint φ y ≫ (snd (splitTorus S X) Y).left = y

lemma **isPullback_prod** : IsPullback ((fst (splitTorus S X) Y).left ≫ torusFst S X) (snd (splitTorus S X) Y).left (splitTorusAff ZU X).hom (Y.hom ≫ toZ S)

lemma **map_prodPoint** {R R' : CommRingCat.{0}} (g : R ⟶ R') (φ : torusRing ZU X ⟶ R) (y : Spec R ⟶ Y.left) : Spec.map g ≫ prodPoint φ y = prodPoint (φ ≫ g) (Spec.map g ≫ y)

lemma **value_comp** {R R' : CommRingCat.{0}} (g : R ⟶ R') (φ : torusRing ZU X ⟶ R) (y : Spec R ⟶ Y.left) : value φ y ≫ g = value (φ ≫ g) (Spec.map g ≫ y)

lemma **charUnit_val** (χ : X) : (charUnit S X χ).val = (torusFst S X).appTop.hom ((Scheme.ΓSpecIso (torusRing ZU X)).inv.hom (single χ 1))

lemma **value_char** {R : CommRingCat.{0}} (φ : torusRing ZU X ⟶ R) (y : Spec R ⟶ Y.left) (χ : X) : (value φ y).hom ((fst (splitTorus S X) Y).left.appTop.hom (charUnit S X χ).val) = φ.hom (single χ 1)

lemma **IsMulLeft.value_mul** {ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)} (hρ : IsMulLeft ρ) {R : CommRingCat.{0}} (φ φ' ψ : torusRing ZU X ⟶ R) (y : Spec R ⟶ Y.left) (h : ∀ x : X, ψ.hom (single x 1) = φ.hom (single x 1) * φ'.hom (single x 1)) : (value ψ y).hom ρ = (value φ y).hom ρ * (value φ' y).hom ρ

lemma **fromInt_single** (R : CommRingCat.{0}) (x : X) : (fromInt X R).hom (single x 1) = single x 1

lemma **expand_char** {R : CommRingCat.{0}} (y : Spec R ⟶ Y.left) (χ : X) : (expand y).hom ((fst (splitTorus S X) Y).left.appTop.hom (charUnit S X χ).val) = single χ 1

lemma **expand_map** {R R' : CommRingCat.{0}} (g : R ⟶ R') (y : Spec R ⟶ Y.left) (ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)) : (expand (Spec.map g ≫ y)).hom ρ = AddMonoidAlgebra.mapRingHom X g.hom ((expand y).hom ρ)

lemma **IsMulLeft.isMultiplicative** {ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)} (hρ : IsMulLeft ρ) {R : CommRingCat.{0}} (y : Spec R ⟶ Y.left) : IsMultiplicative ((expand y).hom ρ)

lemma **isPushout_grpAlg** (A : Type) [CommRing A] : IsPushout (CommRingCat.ofHom (algebraMap ZU.{0} (AddMonoidAlgebra ZU.{0} X))) (CommRingCat.ofHom (algebraMap ZU.{0} A)) (CommRingCat.ofHom (algebraMap (AddMonoidAlgebra ZU.{0} X) (AddMonoidAlgebra A X))) (CommRingCat.ofHom (algebraMap A (AddMonoidAlgebra A X)))

lemma **isPullback_expand** {R : CommRingCat.{0}} (y : Spec R ⟶ Y.left) : IsPullback (prodPoint (fromInt X R) (Spec.map (const X R) ≫ y)) (Spec.map (const X R)) (snd (splitTorus S X) Y).left y

lemma **ext_of_expand** {f g : Γ((splitTorus S X ⊗ Y).left, ⊤)} (h : ∀ p : Y.left, ∃ (R : CommRingCat.{0}) (y : Spec R ⟶ Y.left), IsOpenImmersion y ∧ p ∈ Set.range y ∧ (expand y).hom f = (expand y).hom g) : f = g

lemma **IsCharNear.residueField** {ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)} {χ : X} {p : Y.left} (h : IsCharNear ρ χ p) : (expand (Y.left.fromSpecResidueField p)).hom ρ = single χ 1

lemma **IsCharNear.unique** {ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)} {χ χ' : X} {p : Y.left} (h : IsCharNear ρ χ p) (h' : IsCharNear ρ χ' p) : χ = χ'

lemma **IsMulLeft.exists_isCharNear** {ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)} (hρ : IsMulLeft ρ) (hu : IsUnit ρ) (p : Y.left) : ∃ χ, IsCharNear ρ χ p

theorem **existsUnique_character_of_mul** (S : Scheme.{0}) (X : Type) [AddCommGroup X] (Y : Over S) [ConnectedSpace Y.left] (ρ : Γ((splitTorus S X ⊗ Y).left, ⊤)ˣ) (hρ : ∀ (Z : Over S) (t t' : Z ⟶ splitTorus S X) (y : Z ⟶ Y), (lift (t * t') y).left.appTop.hom ρ.val = (lift t y).left.appTop.hom ρ.val * (lift t' y).left.appTop.hom ρ.val) : ∃! χ : X, ρ.val = (fst (splitTorus S X) Y).left.appTop.hom (charUnit S X χ).val

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsUniqueDescendOfTorsionInvariant

theorem **isOpenImmersion_of_section_of_etale** {X S : Scheme.{0}} (f : X ⟶ S) [Etale f] (s : S ⟶ X) (hs : s ≫ f = 𝟙 S) : IsOpenImmersion s

theorem **exists_section_of_etale** {K : Type} [Field K] [IsAlgClosed K] {X : Scheme.{0}} (f : X ⟶ Spec (CommRingCat.of K)) [Etale f] (x : X) : ∃ s : Spec (CommRingCat.of K) ⟶ X, s ≫ f = 𝟙 _ ∧ x ∈ Set.range s

theorem **exists_local_rational_torsion** {K : Type} [Field K] [IsAlgClosed K] (A : Over (Spec (CommRingCat.of K))) [GrpObj A] (n : ℕ) [Etale (mulA A n).left] {Z : Over (Spec (CommRingCat.of K))} (d : Z ⟶ A) (hd : d ^ n = 1) (p : Z.left) : ∃ (W : Over (Spec (CommRingCat.of K))) (c : W ⟶ Z) (_ : IsOpenImmersion c.left), p ∈ Set.range c.left ∧ ∃ y : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ A, y ^ n = 1 ∧ c ≫ d = toUnit W ≫ y

theorem **existsUnique_appTop_eq_of_flat** {X Y : Scheme.{0}} (m : X ⟶ Y) [QuasiCompact m] [Surjective m] [Flat m] (f : Γ(X, ⊤)) (h : ∀ {Z : Scheme.{0}} (a b : Z ⟶ X), a ≫ m = b ≫ m → a ≫ TLinPf.mOf f = b ≫ TLinPf.mOf f) : ∃! g : Γ(Y, ⊤), m.appTop.hom g = f

theorem **resLE_appTop_topIso_inv** {X Y : Scheme.{0}} (t : X ⟶ Y) {U : Y.Opens} {V : X.Opens} (e : V ≤ t ⁻¹ᵁ U) (g : Γ(Y, U)) : (t.resLE U V e).appTop.hom (U.topIso.inv.hom g) = V.topIso.inv.hom ((t.appLE U V e).hom g)

theorem **existsUnique_descend_of_torsion_invariant** {K : Type} [Field K] [IsAlgClosed K] (A : Over (Spec (CommRingCat.of K))) [GrpObj A] [IsCommMonObj A] (n : ℕ) [Etale (mulA A n).left] [Surjective (mulA A n).left] [QuasiCompact (mulA A n).left] (U : A.left.Opens) (f : Γ(A.left, (mulA A n).left ⁻¹ᵁ U)) (hf : ∀ y : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ A, y ^ n = 1 → ∀ (e : (mulA A n).left ⁻¹ᵁ U ≤ ((toUnit A ≫ y) * 𝟙 A).left ⁻¹ᵁ ((mulA A n).left ⁻¹ᵁ U)), (((toUnit A ≫ y) * 𝟙 A).left.appLE _ _ e).hom f = f) : ∃! g : Γ(A.left, U), ((mulA A n).left.app U).hom g = f

theorem **torsion_invariant_descends** : (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) → ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ), (n : K) ≠ 0 → ∀ (U : P.A.left.Opens) (f : Γ(P.A.left, (mulA P.A n).left ⁻¹ᵁ U)), (∀ y : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ P.A, y ^ n = 1 → ∀ (e : (mulA P.A n).left ⁻¹ᵁ U ≤ ((toUnit P.A ≫ y) * 𝟙 P.A).left ⁻¹ᵁ ((mulA P.A n).left ⁻¹ᵁ U)), (((toUnit P.A ≫ y) * 𝟙 P.A).left.appLE _ _ e).hom f = f) → ∃! g : Γ(P.A.left, U), ((mulA P.A n).left.app U).hom g = f

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsUniqueGeneratorExtendDiscreteValuationRing

theorem **gm_eq_of_corr** {X Y Y' Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) {I : Type} [Fintype I] {φ : I → (Y ⟶ X)} {φ' : I → (Y' ⟶ X)} {ε : I → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' : LB.Gen (LB.LinComb L φ' ε)} (ht : t.IsGen) (ht' : t'.IsGen) (r : Z ⟶ Y) (r' : Z ⟶ Y') (h : ∀ i, r ≫ φ i = r' ≫ φ' i) (hc : LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t r) (LB.pullGen L _ _ t' r')) (z : Z) (a : I → L.ι) (ha : r z ∈ LB.wchart L φ a) (ha' : r' z ∈ LB.wchart L φ' a) : gm r z (LB.wchart L φ a) (t.s a) = gm r' z (LB.wchart L φ' a) (t'.s a)

theorem **res_pullU_eq_of_corr** {X Y Y' Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) {I : Type} [Fintype I] {φ : I → (Y ⟶ X)} {φ' : I → (Y' ⟶ X)} {ε : I → ℤ} {t : LB.Gen (LB.LinComb L φ ε)} {t' : LB.Gen (LB.LinComb L φ' ε)} (ht : t.IsGen) (ht' : t'.IsGen) (r : Z ⟶ Y) (r' : Z ⟶ Y') (h : ∀ i, r ≫ φ i = r' ≫ φ' i) (hc : LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t r) (LB.pullGen L _ _ t' r')) (a : I → L.ι) {A : Z.Opens} (hA : A ≤ r ⁻¹ᵁ LB.wchart L φ a) (hA' : A ≤ r' ⁻¹ᵁ LB.wchart L φ' a) : r'.appLE (LB.wchart L φ' a) A hA' (LB.ts t' a : Γ(Y', LB.wchart L φ' a)ˣ) = r.appLE (LB.wchart L φ a) A hA (LB.ts t a : Γ(Y, LB.wchart L φ a)ˣ)

theorem **existsUnique_gen_extend** : ∀ (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] (H X : Scheme.{0}) (p : H ⟶ Spec (CommRingCat.of V)) (e : Spec (CommRingCat.of V) ⟶ H), Smooth p → GeometricallyIntegral p → e ≫ p = 𝟙 _ → ∀ (eK : Spec (CommRingCat.of K) ⟶ pullback p (etaK V K)), eK ≫ pullback.fst p (etaK V K) = etaK V K ≫ e → ∀ (L : LB X), L.IsLB → ∀ (I : Type) [Fintype I] (φ : I → (H ⟶ X)) (ε : I → ℤ) (t : LB.Gen (LB.LinComb L (fun i => pullback.fst p (etaK V K) ≫ φ i) ε)) (ρ : LB.Gen (LB.LinComb L (fun i => e ≫ φ i) ε)), t.IsGen → ρ.IsGen → LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t eK) (LB.pullGen L _ _ ρ (etaK V K)) → ∃! c : LB.Gen (LB.LinComb L φ ε), c.IsGen ∧ LB.pullGen L _ _ c (pullback.fst p (etaK V K)) = t

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsUniqueGeneratorExtendDiscreteValuationRingStatement

theorem **_root_.GlobalTheory.T.existsUnique_generator_extend_discreteValuationRing** : ∀ (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] (H X : Scheme.{0}) (p : H ⟶ Spec (CommRingCat.of V)) (e : Spec (CommRingCat.of V) ⟶ H), Smooth p → GeometricallyIntegral p → e ≫ p = 𝟙 _ → ∀ (eK : Spec (CommRingCat.of K) ⟶ pullback p (etaK V K)), eK ≫ pullback.fst p (etaK V K) = etaK V K ≫ e → ∀ (L : LB X), L.IsLB → ∀ (I : Type) [Fintype I] (φ : I → (H ⟶ X)) (ε : I → ℤ) (t : LB.Gen (LB.LinComb L (fun i => pullback.fst p (etaK V K) ≫ φ i) ε)) (ρ : LB.Gen (LB.LinComb L (fun i => e ≫ φ i) ε)), t.IsGen → ρ.IsGen → LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ t eK) (LB.pullGen L _ _ ρ (etaK V K)) → ∃! c : LB.Gen (LB.LinComb L φ ε), c.IsGen ∧ LB.pullGen L _ _ c (pullback.fst p (etaK V K)) = t

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsUniqueSecEqMulGen

theorem **existsUnique_sec_eq_mul_gen** {Y : Scheme.{0}} (M : LB Y) (hc : ⊔ j, M.U j = ⊤) (s : LB.Sec M) (t : LB.Gen M) (ht : t.IsGen) : ∃! v : Γ(Y, ⊤), ∀ j, s.f j = (Y.presheaf.map (homOfLE (le_top : M.U j ≤ ⊤)).op).hom v * (t.s j).val

theorem **powFrame_isGen** {S : Scheme.{0}} (A : Over S) [GrpObj A] (L : LB A.left) {T : Over S} (z : T ⟶ A) (e : LB.Gen (LB.LinComb L (Y

theorem **evValSt** : ∀ S : Scheme.{0}, EvValSt (S

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ExistsUniqueWeilIs

theorem **exists_ratio** {X Y S₀ : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (hF : FormEq φ ε φ' ε') (p : Y ⟶ S₀) (e : S₀ ⟶ Y) (he : e ≫ p = 𝟙 _) (hp : Function.Surjective p.appTop.hom) {u : LB.Gen (LB.LinComb M φ ε)} {u' : LB.Gen (LB.LinComb M φ' ε')} (hu : u.IsGen) (hu' : u'.IsGen) : ∃ c : Γ(S₀, ⊤)ˣ, ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J), Good M z J → val u r z J = val u' r z J * gm (r ≫ p) z ⊤ c

theorem **glue_top** {Y : Scheme.{0}} {ι' : Type} (W : ι' → Y.Opens) (hW : ∀ y, ∃ s, y ∈ W s) (c : ∀ s, Γ((W s).toScheme, ⊤)ˣ) (agree : ∀ s s' (Z : Scheme.{0}) (r : Z ⟶ (W s).toScheme) (r' : Z ⟶ (W s').toScheme) (z : Z), r ≫ (W s).ι = r' ≫ (W s').ι → gm r z ⊤ (c s) = gm r' z ⊤ (c s')) : ∃ a : Γ(Y, ⊤)ˣ, ∀ s (Z : Scheme.{0}) (r : Z ⟶ (W s).toScheme) (z : Z), gm (r ≫ (W s).ι) z ⊤ a = gm r z ⊤ (c s)

theorem **surj_fst** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (T : Over S) : Function.Surjective (fst T A).left.appTop.hom

theorem **lift_translPt** {T Z : Over S} (x : T ⟶ A) (t : Z ⟶ T) (a : Z ⟶ A) : lift t a ≫ translPt A x = (t ≫ x) * a

theorem **lift_translT** {T Z : Over S} (x : T ⟶ A) (t : Z ⟶ T) (a : Z ⟶ A) : lift t a ≫ translT A x = lift t ((t ≫ x) * a)

theorem **prod_theta** {T Z : Over S} (x : T ⟶ A) (t : Z ⟶ T) (a : Z ⟶ A) {U : Type} [CommGroup U] (h : (Z.left ⟶ A.left) → U) : ∏ i : Fin 2, h ((lift t a).left ≫ ![(snd T A).left, (translPt A x).left] i) ^ ![k, -k] i = h a.left ^ k * h ((t ≫ x) * a).left ^ (-k)

theorem **val_transl** {T Z : Over S} (θ : ThetaGrp A L k T) (x : T ⟶ A) (t : Z ⟶ T) (a : Z ⟶ A) {z : Z.left} {J} (hJ : Good L z J) : val (LB.pullGen L (K

theorem **val_prod** {T Z : Over S} (θ θ' : ThetaGrp A L k T) (t : Z ⟶ T) (a : Z ⟶ A) {z : Z.left} {J} (hJ : Good L z J) : val (LB.mulGen L _ _ _ _ θ.φ (LB.pullGen L _ _ θ'.φ (translT A θ.pt).left)) (lift t a).left z J = thetaVal θ t a z J * thetaVal θ' t ((t ≫ θ.pt) * a) z J

theorem **val_const** {T Z : Over S} (c : Γ(T.left, ⊤)ˣ) (t : Z ⟶ T) (a : Z ⟶ A) (z : Z.left) (J) : val (constGen A L c) (lift t a).left z J = gm t.left z ⊤ c

theorem **commIsOf_iff** (hL : L.IsLB) {T : Over S} (θx θy : ThetaGrp A L k T) (c : Γ(T.left, ⊤)ˣ) : CommIsOf A L k θx θy c ↔ ∀ (Z : Over S) (t : Z ⟶ T) (a : Z ⟶ A) (z : Z.left) (J), Good L z J → commVal θx θy t a z J = gm t.left z ⊤ c

theorem **exists_pull** (hL : L.IsLB) {T T' : Over S} (θ : ThetaGrp A L k T) (t : T' ⟶ T) : ∃ θ' : ThetaGrp A L k T', θ'.pt = t ≫ θ.pt

theorem **thetaVal_ratio** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T T' Z : Over S} (θ : ThetaGrp A L k T) (θ' : ThetaGrp A L k T') (t : Z ⟶ T) (t' : Z ⟶ T') (h : t ≫ θ.pt = t' ≫ θ'.pt) (z : Z.left) : ∃ u : (Z.left.presheaf.stalk z)ˣ, ∀ (a : Z ⟶ A) (J), Good L z J → thetaVal θ t a z J = thetaVal θ' t' a z J * u

theorem **commVal_indep** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T T' Z : Over S} (θx θy : ThetaGrp A L k T) (θx' θy' : ThetaGrp A L k T') (t : Z ⟶ T) (t' : Z ⟶ T') (hx : t ≫ θx.pt = t' ≫ θx'.pt) (hy : t ≫ θy.pt = t' ≫ θy'.pt) (a : Z ⟶ A) (z : Z.left) (J) (hJ : Good L z J) : commVal θx θy t a z J = commVal θx' θy' t' a z J

theorem **exists_commIsOf** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} (θx θy : ThetaGrp A L k T) : ∃ c : Γ(T.left, ⊤)ˣ, CommIsOf A L k θx θy c

theorem **gm_unitRes** {T : Over S} (V : T.left.Opens) (c : Γ(T.left, ⊤)ˣ) {Z : Scheme.{0}} (r : Z ⟶ V.toScheme) (z : Z) : gm r z ⊤ (unitRes V c) = gm (r ≫ V.ι) z ⊤ c

theorem **overLE_ι** {T : Over S} {V W : T.left.Opens} (h : V ≤ W) : overLE h ≫ overResι W = overResι V

theorem **weilIs_unique** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} {x y : T ⟶ A} {c c' : Γ(T.left, ⊤)ˣ} (h : WeilIs A L k x y c) (h' : WeilIs A L k x y c') : c = c'

theorem **exists_weilIs** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} {x y : T ⟶ A} (hx : ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp A L k (overRes V), θ.pt = overResι V ≫ x) (hy : ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp A L k (overRes V), θ.pt = overResι V ≫ y) : ∃ c : Γ(T.left, ⊤)ˣ, WeilIs A L k x y c

theorem **existsUnique_weilIs** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResι V ≫ x) → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (x y : T ⟶ P.A), x ^ k = 1 → y ^ k = 1 → ∃! c : Γ(T.left, ⊤)ˣ, WeilIs P.A P.L k x y c

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FiniteFaithfullyFlatKer

theorem **tr_comm** (e : ι → M) (c : ι → M →ₗ[A] A) (h : ∀ m, ∑ i, c i m • e i = m) (φ ψ : M →ₗ[A] M) : tr e c (φ ∘ₗ ψ) = tr e c (ψ ∘ₗ φ)

@[simp] theorem **ev_tmul** (B : Type*) [CommRing B] [Algebra R B] (f : Λ →ₗ[R] R) (b : B) (x : Λ) : ev B f (b ⊗ₜ x) = f x • b

theorem **hit_apply** (f : Λ →ₗ[R] R) (y : Λ) : hit f y = ev Λ f (Coalgebra.comul y)

theorem **shear_tmul** (x y : Λ) : shear R Λ (x ⊗ₜ y) = x • Coalgebra.comul (R

theorem **shear_surjective** : Function.Surjective (shear R Λ)

theorem **shear_hit** (f : Λ →ₗ[R] R) (z : Λ ⊗[R] Λ) : shear R Λ ((hit f).baseChange Λ z) = (hit f).baseChange Λ (shear R Λ z)

theorem **lam_apply** (e : ι → Λ) (c : ι → Λ →ₗ[R] R) (f : Λ →ₗ[R] R) (x : Λ) : lam e c f x = tr e c (hit f ∘ₗ LinearMap.mulLeft R x)

theorem **lam_integral** [Module.Finite R Λ] (e : ι → Λ) (c : ι → Λ →ₗ[R] R) (h : ∀ m, ∑ i, c i m • e i = m) (f : Λ →ₗ[R] R) (x : Λ) : ev Λ (lam e c f) (Coalgebra.comul x) = algebraMap R Λ (lam e c f x)

theorem **sum_lam** (e : ι → Λ) (c : ι → Λ →ₗ[R] R) (h : ∀ m, ∑ i, c i m • e i = m) (x : Λ) : ∑ k, lam e c (c k) (HopfAlgebra.antipode R (e k) * x) = Coalgebra.counit (R

theorem **exists_integrals** (R Λ : Type*) [CommRing R] [CommRing Λ] [HopfAlgebra R Λ] [Module.Finite R Λ] [Module.Projective R Λ] : ∃ (n : ℕ) (l : Fin n → Λ →ₗ[R] R) (T : Fin n → Λ), (∀ k x, ev Λ (l k) (Coalgebra.comul x) = algebraMap R Λ (l k x)) ∧ ∀ x, ∑ k, l k (T k * x) = Coalgebra.counit (R

theorem **ev_mem_equalizer** (hassoc : ∀ b, TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

theorem **exists_dual_family** [Module.Finite R Λ] [Module.Projective R Λ] (hassoc : ∀ b, TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

theorem **finite_faithfullyFlat_ker** [Module.Finite R Λ] [Module.Projective R Λ] (hassoc : ∀ b, TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

theorem **GlobalTheory.HopfQuot.affine_quotient** : (∀ (A B M : Type) [CommRing A] [IsLocalRing A] [Infinite (IsLocalRing.ResidueField A)] [CommRing B] [Algebra A B] [Finite (MaximalSpectrum B)] [AddCommGroup M] [Module B M] [Module A M] [IsScalarTower A B M] (n : ℕ) (_ : Module.Basis (Fin n) B M) (N : Submodule A M), (∀ m : MaximalSpectrum B, (IsLocalRing.maximalIdeal A).map (algebraMap A B) ≤ m.asIdeal) → Submodule.span B (N : Set M) = ⊤ → ∃ b : Module.Basis (Fin n) B M, ∀ i, b i ∈ N) → ∀ (R Λ B : Type) [CommRing R] [CommRing Λ] [HopfAlgebra R Λ] [Module.Finite R Λ] [Module.Projective R Λ] [CommRing B] [Algebra R B] (ρ : B →ₐ[R] TensorProduct R B Λ), (∀ b, TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FiniteOfNotLinearIndependent

theorem **eval₂Hom_mul_of_isHomogeneous** (c : R →+* A) (u : A) (x : σ → A) {F : MvPolynomial σ R} {t : ℕ} (hF : F.IsHomogeneous t) : MvPolynomial.eval₂Hom c (fun j => u * x j) F = u ^ t * MvPolynomial.eval₂Hom c x F

theorem **exists_isHomogeneous_eval₂Hom_eq** (c : R →+* A) (x : σ → A) (i : σ) (hi : x i = 1) (p : MvPolynomial σ R) {t : ℕ} (hp : p.totalDegree ≤ t) : ∃ F : MvPolynomial σ R, F.IsHomogeneous t ∧ MvPolynomial.eval₂Hom c x F = MvPolynomial.eval₂Hom c x p

theorem **rs_eval₂Hom** {R σ : Type} [CommRing R] (f : X ⟶ Spec (CommRingCat.of R)) {U V : X.Opens} (h : V ≤ U) (x : σ → Γ(X, U)) (F : MvPolynomial σ R) : rs X h (MvPolynomial.eval₂Hom (cst f U) x F) = MvPolynomial.eval₂Hom (cst f V) (fun j => rs X h (x j)) F

theorem **sec_eq_zero** [IsIntegral X] {M : LB X} (σ : LB.Sec M) {k₀ : M.ι} {W : X.Opens} (hW : W ≤ M.U k₀) [Nonempty W] (h : rs X hW (σ.f k₀) = 0) (k : M.ι) : σ.f k = 0

theorem **growth_of_veryAmpleBy** {K : Type} [Field K] {X : Scheme.{0}} [IsIntegral X] (f : X ⟶ Spec (CommRingCat.of K)) {H : LB X} {m : ℕ} (s : Fin (m + 1) → LB.Sec H) (hs : VeryAmpleBy f H s) (g t₀ : ℕ) (hgen : ∀ t : ℕ, t₀ ≤ t → ∀ σ : LB.Sec (pw H t), ∃ F : MvPolynomial (Fin (m + 1)) K, F.IsHomogeneous t ∧ ∀ k : H.ι, σ.f k = MvPolynomial.eval₂Hom (cst f (H.U k)) (fun j => (s j).f k) F) (hGK : ∀ (B : Type) [CommRing B] [IsDomain B] [Algebra K B] (r : Fin (m + 1) → B), Function.Surjective (MvPolynomial.aeval r : MvPolynomial (Fin (m + 1)) K →ₐ[K] B) → Algebra.trdeg K B = g → ∃ c C : ℕ, 1 ≤ c ∧ ∀ t : ℕ, 1 ≤ t → t ^ g ≤ c * Module.finrank K ((MvPolynomial.restrictTotalDegree (Fin (m + 1)) K t).map (MvPolynomial.aeval r : MvPolynomial (Fin (m + 1)) K →ₐ[K] B).toLinearMap) ∧ Module.finrank K ((MvPolynomial.restrictTotalDegree (Fin (m + 1)) K t).map (MvPolynomial.aeval r : MvPolynomial (Fin (m +

theorem **finite_of_not_linearIndependent** {R V : Type} [CommRing R] (hR : IsField R) [AddCommGroup V] [Module R V] (n : ℕ) (h : ∀ b : Fin n → V, ¬ LinearIndependent R b) : Module.Finite R V

theorem **finrank_compHom** {K R V : Type} [CommRing K] [CommRing R] (e : K ≃+* R) [AddCommGroup V] [Module R V] : letI

theorem **finite_compHom** {K R V : Type} [CommRing K] [CommRing R] (e : K ≃+* R) [AddCommGroup V] [Module R V] [Module.Finite R V] : letI

theorem **finite_secM** {K : Type} [Field K] (P : PolAV (Spec (CommRingCat.of K))) (hfin : ∀ M : LB P.A.left, M.IsLB → ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) * (s k).f i = 0) (a : ℤ) : Module.Finite Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L a (baseOver (Spec (CommRingCat.of K))))

theorem **wchart_le_one** (j : PUnit → L.ι) : LB.wchart L (fun _ : PUnit => π) j ≤ LB.wchart L (fun _ : Fin 1 => π) (fun _ => j PUnit.unit)

theorem **wchart_le_unit** (j : Fin 1 → L.ι) : LB.wchart L (fun _ : Fin 1 => π) j ≤ LB.wchart L (fun _ : PUnit => π) (fun _ => j 0)

theorem **lcg_pow_pow** (N t : ℕ) (j j' : PUnit → L.ι) : (LB.lcg L (K

theorem **lcg_mul** (N t : ℕ) (j j' : Fin 1 → L.ι) : LB.lcg L (K

theorem **secM_finite_and_growth** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → ∃ n : ℕ, ∀ s : Fin n → LB.Sec M, ∃ c : Fin n → Γ(Spec (CommRingCat.of K), ⊤), c ≠ 0 ∧ ∀ i, ∑ k, (P.A.hom.appLE ⊤ (M.U i) le_top).hom (c k) * (s k).f i = 0) → (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB X) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FinrankOfIsSecBasis

theorem **gs_finsum** (r : Z ⟶ X) (z : Z) (U : X.Opens) (h : r z ∈ U) {n : ℕ} (a : Fin n → Γ(X, U)) : gs r z U (∑ i, a i) = ∑ i, gs r z U (a i)

theorem **gs_cst** (p : X ⟶ Spec (CommRingCat.of R)) (W : X.Opens) (r : Z ⟶ X) (z : Z) (hz : r z ∈ W) (a : R) : gs r z W (GlobalTheory.AbLaw.cst p W a) = Tor.cst (r ≫ p) z a

theorem **cstHom_apply** (p : Y ⟶ Spec (CommRingCat.of R)) (y : Y) (a : R) : cstHom p y a = Tor.cst p y a

theorem **cmb_of_loc** {p : Y ⟶ Spec (CommRingCat.of R)} {N : LB Y} {n : ℕ} {b : Fin n → LB.Sec N} {r : Fin n → R} {s : LB.Sec N} (h : ∀ y, ∃ j, y ∈ N.U j ∧ CmbAt p b r s j y) : Cmb p b r s

theorem **sec_ext_loc** {N : LB Y} {s t : LB.Sec N} (h : ∀ y, ∃ j, y ∈ N.U j ∧ gs (𝟙 Y) y (N.U j) (s.f j) = gs (𝟙 Y) y (N.U j) (t.f j)) : s = t

theorem **exists_rel** {A B : LB Y} (hA : A.IsLB) (hB : B.IsLB) (H : LBIso A B) : ∃ u : ∀ y : Y, A.ι → B.ι → (Y.presheaf.stalk y)ˣ, (∀ s : LB.Sec A, ∃ t : LB.Sec B, ∀ y a b, y ∈ A.U a → y ∈ B.U b → gs (𝟙 Y) y (B.U b) (t.f b) = (u y a b).val * gs (𝟙 Y) y (A.U a) (s.f a)) ∧ (∀ t : LB.Sec B, ∃ s : LB.Sec A, ∀ y a b, y ∈ A.U a → y ∈ B.U b → gs (𝟙 Y) y (B.U b) (t.f b) = (u y a b).val * gs (𝟙 Y) y (A.U a) (s.f a))

theorem **Bas.of_iso** {p : Y ⟶ Spec (CommRingCat.of R)} {A B : LB Y} (hA : A.IsLB) (hB : B.IsLB) (H : LBIso A B) {n : ℕ} {b : Fin n → LB.Sec A} (hb : Bas p A b) : ∃ b' : Fin n → LB.Sec B, Bas p B b'

theorem **gs_pullSec** {A' : LB X'} {A : LB X} {h : X' ⟶ X} (c : ChartPull A' A h) (s : LB.Sec A) (r : Z ⟶ X') (z : Z) (i : A'.ι) (hi : r z ∈ A'.U i) : gs r z (A'.U i) ((pullSec c s).f i) = gs (r ≫ h) z (A.U (c.idx i)) (s.f (c.idx i))

theorem **perf_sec** {f : X ⟶ Spec (CommRingCat.of R)} {M : LB X} (Q : Perf f M) (g : R →+* R') {π : X' ⟶ X} {f' : X' ⟶ Spec (CommRingCat.of R')} (h : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) : ∃ E : {v : Fin Q.a → R' // (Q.D₀.map g).mulVec v = 0} ≃ LB.Sec (M.pull π), ∀ v i (Z : Scheme.{0}) (r : Z ⟶ X') (z : Z), π (r z) ∈ Q.W i → gs r z ((M.pull π).U (fun _ => Q.ch i)) ((E v).f (fun _ => Q.ch i)) = ∑ k, Tor.cst (r ≫ f') z (v.1 k) * gs (r ≫ π) z (Q.W i) (Q.u k i)

theorem **perf_cover** {f : X ⟶ Spec (CommRingCat.of R)} {M : LB X} (Q : Perf f M) (x : X) : ∃ i, x ∈ Q.W i

theorem **perf_bas** {f : X ⟶ Spec (CommRingCat.of R)} {M : LB X} (Q : Perf f M) (g : R →+* R') {π : X' ⟶ X} {f' : X' ⟶ Spec (CommRingCat.of R')} (h : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) {n : ℕ} (B : Matrix (Fin Q.a) (Fin n) R) (hB : Q.D₀ * B = 0) (hu : ∀ v : Fin Q.a → R', (Q.D₀.map g).mulVec v = 0 → ∃! z : Fin n → R', (B.map g).mulVec z = v) : ∃ b : Fin n → LB.Sec (M.pull π), Bas f' _ b

theorem **perf_lift** {W : Scheme.{0}} (L : LB W) {K : Type} [Fintype K] (φ : K → (X ⟶ W)) (ε : K → ℤ) {f : X ⟶ Spec (CommRingCat.of R)} (Q : Perf f (LB.LinComb L φ ε)) (g : R →+* R') (hg : Function.Surjective g) {π : X' ⟶ X} {f' : X' ⟶ Spec (CommRingCat.of R')} (h : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) {n : ℕ} (B : Matrix (Fin Q.a) (Fin n) R) (hB : Q.D₀ * B = 0) (hu : ∀ v : Fin Q.a → R', (Q.D₀.map g).mulVec v = 0 → ∃! z : Fin n → R', (B.map g).mulVec z = v) (sbar : LB.Sec (LB.LinComb L (fun k => π ≫ φ k) ε)) : ∃ s : LB.Sec (LB.LinComb L φ ε), IsPullSecLC L φ ε π s sbar

theorem **cmb_iff** {p : Y ⟶ Spec (CommRingCat.of R)} {N : LB Y} {n : ℕ} (b : Fin n → LB.Sec N) (r : Fin n → R) (s : LB.Sec N) : Cmb p b r s ↔ ∀ j, s.f j = ∑ i, (Y.presheaf.map (homOfLE le_top).op).hom (p.appTop.hom ((Scheme.ΓSpecIso (CommRingCat.of R)).inv.hom (r i))) * (b i).f j

theorem **Bas.exists_unique** {p : Y ⟶ Spec (CommRingCat.of R)} {N : LB Y} {n : ℕ} {b : Fin n → LB.Sec N} (hb : Bas p N b) (s : LB.Sec N) : ∃! r : Fin n → Γ(Spec (CommRingCat.of R), ⊤), ∀ j, s.f j = ∑ i, (Y.presheaf.map (homOfLE le_top).op).hom (p.appTop.hom (r i)) * (b i).f j

theorem **finrank_of_isSecBasis** {S : Scheme.{0}} (A : Over S) [GrpObj A] (L : LB A.left) (a : ℤ) (T : Over S) [Nontrivial Γ(T.left, ⊤)] {n : ℕ} (b : Fin n → LB.Sec (LPow A L a T)) (hb : IsSecBasis _ b) : Module.finrank Γ(T.left, ⊤) (SecM A L a T) = n

theorem **iso_of_gen_three** (L : LB X) (a b c : Y ⟶ X) (m n k : ℤ) (w : LB.Gen (LB.LinComb L (K

theorem **lbIso_refl** {A : LB X} (hA : A.IsLB) : LBIso A A

theorem **translT_translPt** (x y : T ⟶ A) : translT A x ≫ translPt A y = translPt A (y * x)

theorem **translPt_one** : translPt A (1 : T ⟶ A) = snd T A

theorem **translT_comp** (x y : T ⟶ A) (h : y * x = 1) : translT A x ≫ translT A y = 𝟙 _

theorem **isIso_translT_left** (x : T ⟶ A) : IsIso (translT A x).left

theorem **geometric_fibre** (hBC : PolBaseChangeSt) (hTW : ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S) (q : ℕ), 1 ≤ q → ∀ c : Over.mk t ⟶ P.A, ∃ d : Over.mk t ⟶ P.A, d ^ q = c ∧ ∃ w : LB.Gen (LB.LinComb P.L (K

theorem **secTwist_of_perf** : (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)) → (∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ), 1 ≤ n → FirstCohomologyVanishes (pw P.L n)) → (∀ (X X' : Scheme.{0}) (e : X' ⟶ X) (M : LB X) (M' : LB X'), IsIso e → M.IsLB → M'.IsLB → (∃ w : LB.Iso M' (M.pull e), w.IsGen) → FirstCohomologyVanishes M → FirstCohomologyVanishes M') → (∀ (R K : Type) [CommRing R] [Field K] (g₀ : R →+* K) (a b c : ℕ) (D₀ : Ma

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FinrankSectionsLineBundlePowEqPow

theorem **le_of_pow_le_mul_pow** {a b C : ℕ} (hb : 1 ≤ b) (h : ∀ k : ℕ, a ^ k ≤ C * b ^ k) : a ≤ b

theorem **le_of_frequently** {a b g : ℕ} (h : ∀ M : ℕ, ∃ n, M ≤ n ∧ a * n ^ g ≤ b * (n + 8) ^ g) : a ≤ b

theorem **eq_pow_of_growth** (h : ℕ → ℕ) (g : ℕ) (G : ℕ → Prop) (hG : ∀ m, G m ∨ G (m + 1)) (hG0 : ¬ G 0) (h1 : h 1 = 1) (hmono : Monotone h) (hrel : ∀ n j, G n → 1 ≤ j → h (n ^ 2 * j) = h (n ^ 2) * h j) (N c C : ℕ) (hN : 1 ≤ N) (hgr : ∀ t, 1 ≤ t → t ^ g ≤ c * h (N * t) ∧ h (N * t) ≤ C * t ^ g) (j : ℕ) (hj : 1 ≤ j) : h j = j ^ g

theorem **finite_index** {R : Type*} [Field R] {M I : Type*} [AddCommGroup M] [Module R M] [Nontrivial M] [Module.Finite R (I → M)] : Finite I

theorem **finrank_eq_mul** {R : Type*} [CommRing R] (hR : IsField R) {I A M B M' : Type*} [AddCommGroup A] [Module R A] [AddCommGroup M] [Module R M] [AddCommGroup B] [Module R B] [AddCommGroup M'] [Module R M'] [Module.Finite R A] [Module.Finite R M'] (hM : Module.finrank R M = 1) (e : A ≃ₗ[R] (I → M)) (e' : B ≃ₗ[R] (I → M')) : Module.finrank R B = Module.finrank R A * Module.finrank R M'

theorem **sec_ne_zero** {X : Scheme.{0}} [IsIntegral X] {M : LB X} (s : LB.Sec M) {i₀ : M.ι} (h₀ : s.f i₀ ≠ 0) (i : M.ι) [hi : Nonempty (M.U i)] : s.f i ≠ 0

theorem **lpow_g_add** (a b : ℤ) (i j : Fin 1 → L.ι) : (LPow A L (a + b) T).g i j = (show Γ((T ⊗ A).left, (LPow A L (a + b) T).U i ⊓ (LPow A L (a + b) T).U j)ˣ from (LPow A L a T).g i j) * (show Γ((T ⊗ A).left, (LPow A L (a + b) T).U i ⊓ (LPow A L (a + b) T).U j)ˣ from (LPow A L b T).g i j)

theorem **mulSec_injective** [IsIntegral (T ⊗ A).left] (a b : ℤ) (s : SecM A L b T) (hs : s ≠ 0) : Function.Injective (mulSec A L T a b s)

theorem **isField_base** (K : Type) [Field K] : IsField Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)

theorem **finrank_sections_lineBundlePow_eq_pow** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K)))) = 1) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n j : ℕ), (n : K) ≠ 0 → 1 ≤ j → Nonempty (SecM P.A P.L ((n ^ 2 * j : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K))) ≃ₗ[Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)] ({x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A // x ^ n = 1} → SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K)))))) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), (∀ j : ℕ, Module.Finite Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L (j : ℤ) (base

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FinrankSectionsPrincipalPolarisationEqOne

theorem **finrank_sections_principal_polarisation_eq_one** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K)))) = 1

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FinrankTotalDegreeImageGrowth

theorem **mem_boxSpan** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (D : ℕ) (a : Fin s → ℕ) (ha : ∀ i, a i ≤ D) (l : Fin n) : (∏ i, y i ^ a i) * b l ∈ boxSpan K y b D

theorem **finrank_boxSpan_le** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (D : ℕ) : Module.finrank K (boxSpan K y b D) ≤ (D + 1) ^ s * n

theorem **monomial_mul_mem** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (D d : ℕ) (a : Fin s → ℕ) (ha : ∀ i, a i ≤ d) {x : B} (hx : x ∈ boxSpan K y b D) : (∏ i, y i ^ a i) * x ∈ boxSpan K y b (D + d)

theorem **boxSpan_mono** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) {D E : ℕ} (h : D ≤ E) : boxSpan K y b D ≤ boxSpan K y b E

theorem **exists_aeval_mul_mem** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (p : MvPolynomial (Fin s) K) {x : B} (hx : ∃ D, x ∈ boxSpan K y b D) : ∃ D, MvPolynomial.aeval y p * x ∈ boxSpan K y b D

theorem **mul_mem_boxSpan** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (D d : ℕ) (z : B) (hz : ∀ l, z * b l ∈ boxSpan K y b d) {x : B} (hx : x ∈ boxSpan K y b D) : z * x ∈ boxSpan K y b (d + D)

theorem **pow_mul_mem_boxSpan** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (D d : ℕ) (z : B) (hz : ∀ l, z * b l ∈ boxSpan K y b d) (k : ℕ) {x : B} (hx : x ∈ boxSpan K y b D) : z ^ k * x ∈ boxSpan K y b (k * d + D)

theorem **prod_mul_mem_boxSpan** {s n : ℕ} (y : Fin s → B) (b : Fin n → B) (D d : ℕ) {I : Type} (r : I → B) (hr : ∀ i l, r i * b l ∈ boxSpan K y b d) (S : Finset I) (a : I → ℕ) {x : B} (hx : x ∈ boxSpan K y b D) : (∏ i ∈ S, r i ^ a i) * x ∈ boxSpan K y b ((∑ i ∈ S, a i) * d + D)

theorem **upper** {s n m : ℕ} (y : Fin s → B) (b : Fin n → B) (hall : ∀ x : B, ∃ D, x ∈ boxSpan K y b D) (r : Fin m → B) : ∃ C : ℕ, ∀ t : ℕ, 1 ≤ t → Module.finrank K ((MvPolynomial.restrictTotalDegree (Fin m) K t).map (MvPolynomial.aeval r : MvPolynomial (Fin m) K →ₐ[K] B).toLinearMap) ≤ C * t ^ s

theorem **lower** {s m : ℕ} (φ : MvPolynomial (Fin s) K →ₐ[K] B) (hφ : Function.Injective φ) (r : Fin m → B) (hr : Function.Surjective (MvPolynomial.aeval r : MvPolynomial (Fin m) K →ₐ[K] B)) : ∃ c : ℕ, 1 ≤ c ∧ ∀ t : ℕ, t ^ s ≤ c * Module.finrank K ((MvPolynomial.restrictTotalDegree (Fin m) K t).map (MvPolynomial.aeval r : MvPolynomial (Fin m) K →ₐ[K] B).toLinearMap)

theorem **finrank_totalDegree_image_growth** : ∀ (K B : Type) [Field K] [CommRing B] [IsDomain B] [Algebra K B] (m g : ℕ) (r : Fin m → B), Function.Surjective (MvPolynomial.aeval r : MvPolynomial (Fin m) K →ₐ[K] B) → Algebra.trdeg K B = g → ∃ c C : ℕ, 1 ≤ c ∧ ∀ t : ℕ, 1 ≤ t → t ^ g ≤ c * Module.finrank K ((MvPolynomial.restrictTotalDegree (Fin m) K t).map (MvPolynomial.aeval r : MvPolynomial (Fin m) K →ₐ[K] B).toLinearMap) ∧ Module.finrank K ((MvPolynomial.restrictTotalDegree (Fin m) K t).map (MvPolynomial.aeval r : MvPolynomial (Fin m) K →ₐ[K] B).toLinearMap) ≤ C * t ^ g

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FirstCohomologyVanishesDescent

theorem **pre_mono** (h : Y ⟶ X) {U U' : X.Opens} (i : U ≤ U') : h ⁻¹ᵘ U ≤ h ⁻¹ᵘ U'

theorem **pre_inf** (h : Y ⟶ X) (U U' : X.Opens) : h ⁻¹ᵘ U ⊓ h ⁻¹ᵘ U' ≤ h ⁻¹ᵘ (U ⊓ U')

theorem **pre_inf'** (h : Y ⟶ X) (U U' : X.Opens) : h ⁻¹ᵘ (U ⊓ U') ≤ h ⁻¹ᵘ U ⊓ h ⁻¹ᵘ U'

theorem **pre_comp** (g : Z ⟶ Y) (h : Y ⟶ X) {U : X.Opens} {V : Y.Opens} {W : Z.Opens} (e : V ≤ h ⁻¹ᵘ U) (e' : W ≤ g ⁻¹ᵘ V) : W ≤ (g ≫ h) ⁻¹ᵘ U

theorem **pre_pull** (M : LB X) (f : Y ⟶ X) (i j : M.ι) {W : Y.Opens} (e : W ≤ (M.pull f).U (fun _ => i) ⊓ (M.pull f).U (fun _ => j)) : W ≤ f ⁻¹ᵘ (M.U i ⊓ M.U j)

theorem **rs_pb** (h : Y ⟶ X) {U : X.Opens} {W W' : Y.Opens} (e : W ≤ h ⁻¹ᵘ U) (i : W' ≤ W) (x : Γ(X, U)) : rs Y i (pb h e x) = pb h (i.trans e) x

theorem **pb_rs** (h : Y ⟶ X) {U U' : X.Opens} {W : Y.Opens} (e : W ≤ h ⁻¹ᵘ U) (i : U ≤ U') (x : Γ(X, U')) : pb h e (rs X i x) = pb h (e.trans (pre_mono h i)) x

theorem **pb_pb** (g : Z ⟶ Y) (h : Y ⟶ X) {U : X.Opens} {V : Y.Opens} {W : Z.Opens} (e : V ≤ h ⁻¹ᵘ U) (e' : W ≤ g ⁻¹ᵘ V) (x : Γ(X, U)) : pb g e' (pb h e x) = pb (g ≫ h) (pre_comp g h e e') x

theorem **pb_congr** {h h' : Y ⟶ X} (H : h = h') {U : X.Opens} {W : Y.Opens} (e : W ≤ h ⁻¹ᵘ U) (x : Γ(X, U)) : pb h e x = pb h' (H ▸ e) x

theorem **app_eq_pb** (h : Y ⟶ X) (U : X.Opens) (x : Γ(X, U)) : (h.app U).hom x = pb h le_rfl x

theorem **rs_pull_g** (M : LB X) (f : Y ⟶ X) (i j : M.ι) {W : Y.Opens} (e : W ≤ (M.pull f).U (fun _ => i) ⊓ (M.pull f).U (fun _ => j)) : rs Y e ((M.pull f).g (fun _ => i) (fun _ => j)).val = pb f (pre_pull M f i j e) (M.g i j).val

theorem **first_cohomology_vanishes_descent** (f : Y ⟶ X) (M : LB X) (G : Type) [Group G] [Fintype G] (τ : G → (Y ⟶ Y)) (hτ : ∀ y, τ y ≫ f = f) (hmul : ∀ y z, τ y ≫ τ z = τ (y * z)) (hu : IsUnit ((Fintype.card G : ℕ) : Γ(X, ⊤))) (hgal : ∀ (U : X.Opens) (s : Γ(Y, f ⁻¹ᵘ U)), (∀ y (e : f ⁻¹ᵘ U ≤ τ y ⁻¹ᵘ (f ⁻¹ᵘ U)), pb (τ y) e s = s) → ∃! g : Γ(X, U), (f.app U).hom g = s) (H : FirstCohomologyVanishes (M.pull f)) : FirstCohomologyVanishes M

theorem **exists_iso_tw** {X : Scheme.{0}} (L : LB X) (hL : L.IsLB) (f : X ⟶ X) (k n r N t : ℤ) (α : LB.Gen (LB.LinComb L (K

theorem **first_cohomology_vanishes_powers** : (∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) → (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB X) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∀ M : LB X, M.IsLB → ∃ t₀ : ℤ, ∀ t, t₀ ≤ t → FirstCohomologyVanishes (tw M H t)) → (∀ (X X' : Scheme.{0}) (e : X' ⟶ X) (M : LB X) (M' : LB X'), IsIso e → M.IsLB → M'.IsLB → (∃ w : LB.Iso M' (M.pull e), w.IsGen) → FirstCohomologyVanishes M → FirstCohomologyVanishes M') → (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FirstCohomologyVanishesOfIsoOfIsGen

theorem **le_pre** (f : Y ⟶ X) {U U' : X.Opens} {W : Y.Opens} (h : W ≤ f ⁻¹ᵘ U) (h' : U ≤ U') : W ≤ f ⁻¹ᵘ U'

theorem **le_pre_inf** (f : Y ⟶ X) {U U' : X.Opens} {W : Y.Opens} (h : W ≤ f ⁻¹ᵘ U) (h' : W ≤ f ⁻¹ᵘ U') : W ≤ f ⁻¹ᵘ (U ⊓ U')

theorem **rs_pb** (f : Y ⟶ X) {U : X.Opens} {W W' : Y.Opens} (h : W ≤ f ⁻¹ᵘ U) (h' : W' ≤ W) (x : Γ(X, U)) : rs Y h' (pb f h x) = pb f (h'.trans h) x

theorem **pb_rs** (f : Y ⟶ X) {U U' : X.Opens} {W : Y.Opens} (h : W ≤ f ⁻¹ᵘ U) (h' : U ≤ U') (x : Γ(X, U')) : pb f h (rs X h' x) = pb f (le_pre f h h') x

theorem **pb_of_eq_id** (g : X ⟶ X) (hg : g = 𝟙 X) {U W : X.Opens} (h : W ≤ g ⁻¹ᵘ U) (h' : W ≤ U) (x : Γ(X, U)) : pb g h x = rs X h' x

theorem **le_of_comp_id** {f : X ⟶ Y} {e : Y ⟶ X} (hfe : f ≫ e = 𝟙 X) {U W : X.Opens} {V : Y.Opens} (h : W ≤ f ⁻¹ᵘ V) (h' : V ≤ e ⁻¹ᵘ U) : W ≤ U

theorem **pb_pb** {f : X ⟶ Y} {e : Y ⟶ X} (hfe : f ≫ e = 𝟙 X) {U W : X.Opens} {V : Y.Opens} (h : W ≤ f ⁻¹ᵁ V) (h' : V ≤ e ⁻¹ᵁ U) (x : Γ(X, U)) : pb f h (pb e h' x) = rs X (le_of_comp_id hfe h h') x

theorem **pull_U_le** (M : LB X) (e : Y ⟶ X) (j : (M.pull e).ι) : (M.pull e).U j ≤ e ⁻¹ᵁ M.U (j PUnit.unit)

theorem **pull_g** (M : LB X) (e : Y ⟶ X) (j j' : (M.pull e).ι) : ((M.pull e).g j j').val = pb e (le_pre_inf e (inf_le_left.trans (pull_U_le M e j)) (inf_le_right.trans (pull_U_le M e j'))) : (M.pull e).U j ⊓ (M.pull e).U j' ≤ e ⁻¹ᵁ (M.U (j PUnit.unit) ⊓ M.U (j' PUnit.unit))) (M.g (j PUnit.unit) (j' PUnit.unit)).val

theorem **firstCohomologyVanishes_pull_of_isIso** (e : Y ⟶ X) [IsIso e] (M : LB X) (hM : FirstCohomologyVanishes M) : FirstCohomologyVanishes (M.pull e)

theorem **val_res** {U V : X.Opens} (h : V ≤ U) (u : Γ(X, U)ˣ) : (res h u).val = rs X h u.val

theorem **iso_rel** {N M : LB X} (w : LB.Iso M N) (hw : w.IsGen) (i j : N.ι) (k l : M.ι) {W : X.Opens} (hi : W ≤ N.U i) (hj : W ≤ N.U j) (hk : W ≤ M.U k) (hl : W ≤ M.U l) : rs X (le_inf hi hk : W ≤ N.U i ⊓ M.U k) (wu w i k).val * rs X (le_inf hk hl : W ≤ M.U k ⊓ M.U l) (M.g k l).val = rs X (le_inf hi hj : W ≤ N.U i ⊓ N.U j) (N.g i j).val * rs X (le_inf hj hl : W ≤ N.U j ⊓ M.U l) (wu w j l).val

theorem **rs_g_self** {M : LB X} (hM : M.IsLB) (k : M.ι) {W : X.Opens} (h : W ≤ M.U k ⊓ M.U k) : rs X h (M.g k k).val = 1

theorem **rs_inv_mul** {U W : X.Opens} (h : W ≤ U) (u : Γ(X, U)ˣ) : rs X h (u⁻¹).val * rs X h u.val = 1

theorem **firstCohomologyVanishes_of_iso_of_isGen** {N M : LB X} (hN : ⊔ i, N.U i = ⊤) (hM : M.IsLB) (w : LB.Iso M N) (hw : w.IsGen) (h0 : FirstCohomologyVanishes N) : FirstCohomologyVanishes M

theorem **h1Zero_of_isIso** : ∀ (X X' : Scheme.{0}) (e : X' ⟶ X) (M : LB X) (M' : LB X'), IsIso e → M.IsLB → M'.IsLB → (∃ w : LB.Iso M' (M.pull e), w.IsGen) → FirstCohomologyVanishes M → FirstCohomologyVanishes M'

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FlatAndSurjectiveOfIsFiniteAbelianEnd

theorem **strictMono_of_locallyQuasiFinite** {X Y : Scheme.{0}} (f : X ⟶ Y) [LocallyQuasiFinite f] : StrictMono (f : X → Y)

theorem **apply_genericPoint** {X : Scheme.{0}} [IrreducibleSpace X] [IsLocallyNoetherian X] (f : X ⟶ X) [LocallyQuasiFinite f] {o : X} (ho : f o = o) : f (genericPoint X) = genericPoint X

theorem **exists_flat_away** {A B : Type} [CommRing A] [CommRing B] [IsDomain A] [IsNoetherianRing A] (φ : A →+* B) (hφ : φ.Finite) : ∃ r : A, r ≠ 0 ∧ ∀ (C : Type) [CommRing C] [Algebra B C] [IsLocalization.Away (φ r) C], ((algebraMap B C).comp φ).Flat

theorem **exists_open_flat** {X Y : Scheme.{0}} [IrreducibleSpace X] [IsIntegral Y] [IsLocallyNoetherian Y] (f : X ⟶ Y) [IsFinite f] (hf : f (genericPoint X) = genericPoint Y) : ∃ W : X.Opens, (W : Set X).Nonempty ∧ ∀ x ∈ W, (f.stalkMap x).hom.Flat

theorem **flat_and_surjective_of_isFinite_grpEnd** (K : Type) [Field K] [IsAlgClosed K] (G : Over (Spec (CommRingCat.of K))) [GrpObj G] [IsIntegral G.left] [LocallyOfFiniteType G.hom] (φ : G ⟶ G) [IsMonHom φ] [IsFinite φ.left] : Flat φ.left ∧ Surjective φ.left

theorem **flat_and_surjective_of_isFinite_abelianEnd** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → ∀ (K : Type) [Field K] [IsAlgClosed K] (A : Over (Spec (CommRingCat.of K))) [GrpObj A], IsAbelianScheme A → ∀ f : A ⟶ A, IsMonHom f → IsFinite f.left → Flat f.left ∧ Surjective f.left

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FlatOfIsFiniteOfFlatGeometricFibres

theorem **lTensor_map_injective** {R A M : Type*} [CommRing R] [CommRing A] [Algebra R A] [AddCommGroup M] [Module R M] [Module A M] [IsScalarTower R A M] [Module.Flat R M] (p : Ideal R) : Function.Injective ((p.map (algebraMap R A)).subtype.lTensor M)

theorem **free_of_flat_reduction** {R A M : Type*} [CommRing R] [CommRing A] [Algebra R A] [IsNoetherianRing A] [IsLocalRing A] [AddCommGroup M] [Module R M] [Module A M] [IsScalarTower R A M] [Module.Flat R M] [Module.Finite A M] (p : Ideal R) (hp : p.map (algebraMap R A) ≤ maximalIdeal A) [Module.Flat (A ⧸ p.map (algebraMap R A)) ((A ⧸ p.map (algebraMap R A)) ⊗[A] M)] : Module.Free A M

theorem **flat_of_flat_fibres** {R A B : Type} [CommRing R] [CommRing A] [CommRing B] [Algebra R A] [Algebra R B] [Algebra A B] [IsScalarTower R A B] [IsNoetherianRing A] [Module.Finite A B] [Module.Flat R B] (H : ∀ (k C : Type) [Field k] [CommRing C] [Algebra A C] (t : R →+* k) (h : k →+* C), (algebraMap A C).comp (algebraMap R A) = h.comp t → Module.Flat C (C ⊗[A] B)) : Module.Flat A B

theorem **flatOver_of_comp_base** {S₀ : Scheme.{0}} {g : Y ⟶ S₀} {u : S₀ ⟶ S} {t : S' ⟶ S₀} (H : FlatOver f (g ≫ u) (t ≫ u)) : FlatOver f g t

theorem **flatOver_descend** {S'' : Scheme.{0}} {g : Y ⟶ S} {t : S' ⟶ S} (σ : S'' ⟶ S') [Surjective σ] [Flat σ] [QuasiCompact σ] (H : FlatOver f g (σ ≫ t)) : FlatOver f g t

theorem **flat_of_flatOver_fields** (f : X ⟶ Y) (g : Y ⟶ S) (hY : IsLocallyNoetherian Y) (h1 : Flat (f ≫ g)) (h2 : IsFinite f) (H : ∀ (k : Type) [Field k] (t : Spec (CommRingCat.of k) ⟶ S), FlatOver f g t) : Flat f

theorem **flat_of_isFinite_of_flat_geometric_fibres** : ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (X Y : Over S) (f : X ⟶ Y), Flat X.hom → LocallyOfFiniteType Y.hom → IsFinite f.left → (∀ (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S), Flat ((Over.pullback t).map f).left) → Flat f.left

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FlatQuotientIsoOfSameKernel

lemma **_root_.GlobalTheory.FlatQuotient.epi_of_flat_surjective** {X Y : Over S} (f : X ⟶ Y) [Flat f.left] [Surjective f.left] : Epi f

private lemma **whiskerRight_left_of_isStableUnderBaseChange** (P : MorphismProperty Scheme.{0}) [P.IsStableUnderBaseChange] {X Y : Over S} (f : X ⟶ Y) (Z : Over S) (hf : P f.left) : P (f ▷ Z).left

private lemma **whiskerLeft_left_of_isStableUnderBaseChange** (P : MorphismProperty Scheme.{0}) [P.IsStableUnderBaseChange] {X Y : Over S} (Z : Over S) (f : X ⟶ Y) (hf : P f.left) : P (Z ◁ f).left

lemma **_root_.GlobalTheory.FlatQuotient.epi_tensorHom** {X Y X' Y' : Over S} (f : X ⟶ Y) (g : X' ⟶ Y') [hf : Flat f.left] [hf' : Surjective f.left] [hg : Flat g.left] [hg' : Surjective g.left] : Epi (f ⊗ₘ g)

lemma **_root_.GlobalTheory.FlatQuotient.comp_eq_of_kills_kernel** {G B B' T : Over S} [GrpObj G] [GrpObj B] [GrpObj B'] (q : G ⟶ B) (i : T ⟶ G) (f : G ⟶ B') [IsMonHom q] [IsMonHom f] (hker : IsPullback i (toUnit _) q (η[B])) (hf : i ≫ f = toUnit _ ≫ η[B']) {Z : Over S} (g g' : Z ⟶ G) (h : g ≫ q = g' ≫ q) : g ≫ f = g' ≫ f

theorem **_root_.GlobalTheory.FlatQuotient.exists_desc** {G B B' T : Over S} [GrpObj G] [GrpObj B] [GrpObj B'] (q : G ⟶ B) (i : T ⟶ G) (f : G ⟶ B') [IsMonHom q] [IsMonHom f] [Flat q.left] [Surjective q.left] [QuasiCompact q.left] (hker : IsPullback i (toUnit _) q (η[B])) (hf : i ≫ f = toUnit _ ≫ η[B']) : ∃ β : B ⟶ B', IsMonHom β ∧ q ≫ β = f

theorem **_root_.GlobalTheory.flatQuotient_iso_of_same_kernel** : ∀ (S : Scheme.{0}) (G B B' T T' : Over S) [GrpObj G] [GrpObj B] [GrpObj B'] [GrpObj T] [GrpObj T'] (q : G ⟶ B) (i : T ⟶ G) (q' : G ⟶ B') (i' : T' ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → IsMonHom q' → IsMonHom i' → Flat q'.left → Surjective q'.left →

QuasiCompact q'.left → IsPullback i' (toUnit _) q' (η[B']) → i' ≫ q = toUnit _ ≫ η[B] → i ≫ q' = toUnit _ ≫ η[B'] → ∃ β : B' ≅ B, IsMonHom β.hom ∧ q' ≫ β.hom = q

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ForallGeometricallyIntegralOfSmoothOfGeometrical

theorem **forall_geometricallyIntegral_of_smooth_of_geometricallyConnected** : ∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FormallyUnramifiedMulInvertibleInteger

theorem **formallyUnramified_mulA** (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (n : ℕ) (hn : IsUnit (n : Γ(S, ⊤))) : FormallyUnramified (mulA G n).left

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FormallyUnramifiedMulInvertibleIntegerStatement

theorem **_root_.GlobalTheory.T.formallyUnramified_mul_invertible_integer** : ∀ (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (n : ℕ), IsUnit (n : Γ(S, ⊤)) → FormallyUnramified (mulA G n).left

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.FunEqZeroOfAffine

theorem **wchart_le_of_eq** {φ : K → (Y ⟶ X)} {φ' : K → (Y' ⟶ X)} (h : Y' ⟶ Y) (e : ∀ k, h ≫ φ k = φ' k) (j : K → L.ι) : LB.wchart L φ' j ≤ h ⁻¹ᵁ LB.wchart L φ j

theorem **sval_pullTo** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K → (Y' ⟶ X)} (s : LB.Sec (LB.LinComb L φ ε)) (h : Y' ⟶ Y) (e : ∀ k, h ≫ φ k = φ' k) (r : Z ⟶ Y') (z : Z) {J : (Z ⟶ X) → L.ι} (hJ : Good L z J) : sval (pullTo s h e) r z J = sval s (r ≫ h) z J

theorem **exists_gen_pullTo** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K → (Y' ⟶ X)} (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) (h : Y' ⟶ Y) (e : ∀ k, h ≫ φ k = φ' k) : ∃ t' : LB.Gen (LB.LinComb L φ' ε), t'.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) (J), Good L z J → val t' r z J = val t (r ≫ h) z J

theorem **whisker_snd_left** {T T' : Over S} (h : T' ⟶ T) : (h ▷ A).left ≫ (snd T A).left = (snd T' A).left

theorem **sv_pullS** (a : ℤ) {T T' : Over S} (h : T' ⟶ T) (s : LB.Sec (LPow A L a T)) {Z : Scheme.{0}} (r : Z ⟶ (T' ⊗ A).left) (z : Z) {J : (Z ⟶ A.left) → L.ι} (hJ : Good L z J) : sv A (pullS A a h s) r z J = sv A s (r ≫ (h ▷ A).left) z J

theorem **sv_of_isTwistPull** {s : LB.Sec (LPow A L b T)} {s' : LB.Sec (LPow A L a T)} (H : IsTwistPull A L ψ m hm a b α s s') {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J : (Z ⟶ A.left) → L.ι} (hJ : Good L z J) : sv A s' r z J = (val α r z J).val ∗ sv A s (r ≫ ψ.left) z J

theorem **isTwistPull_of_sv** (hL : L.IsLB) (hα : α.IsGen) {s : LB.Sec (LPow A L b T)} {s' : LB.Sec (LPow A L a T)} (H : ∀ y J, Good L y J → sv A s' (𝟙 _) y J = (val α (𝟙 _) y J).val ∗ sv A s (𝟙 _ ≫ ψ.left) y J) : IsTwistPull A L ψ m hm a b α s s'

theorem **exists_isTwistPull** (hL : L.IsLB) (hα : α.IsGen) (s : LB.Sec (LPow A L b T)) : ∃ s' : LB.Sec (LPow A L a T), IsTwistPull A L ψ m hm a b α s s'

theorem **exists_twist_pull** (hL : L.IsLB) (hα : α.IsGen) {T' : Over S} (h : T' ⟶ T) {ψ' : T' ⊗ A ⟶ T' ⊗ A} {m' : T' ⊗ A ⟶ A} (hm' : ψ' ≫ snd T' A = m') (e : ψ' ≫ h ▷ A = h ▷ A ≫ ψ) : ∃ α' : LB.Gen (LB.LinComb L (K

theorem **exists_theta_pull** (hL : L.IsLB) {k : ℤ} {T T' : Over S} (h : T' ⟶ T) (θ : ThetaGrp A L k T) : ∃ θ' : ThetaGrp A L k T', θ'.pt = h ≫ θ.pt ∧ ∀ s s', IsActOf A L k θ s s' → IsActOf A L k θ' (pullS A k h s) (pullS A k h s')

theorem **mulT_comm** {T T' : Over S} (h : T' ⟶ T) (n : ℕ) : mulT A T' n ≫ h ▷ A = h ▷ A ≫ mulT A T n

theorem **isVVGen_pull** (P : PolAV S) (N n : ℕ) {T T' : Over S} (h : T' ⟶ T) {t : LB.Sec (LPow P.A P.L N T)} (ht : IsVVGen P N n t) : IsVVGen P N n (pullS P.A N h t)

theorem **exists_isActOf** (hL : L.IsLB) {k : ℤ} {T : Over S} (θ : ThetaGrp A L k T) (s : LB.Sec (LPow A L k T)) : ∃ s', IsActOf A L k θ s s'

theorem **isActOf_unique** (hL : L.IsLB) {k : ℤ} {T : Over S} {θ : ThetaGrp A L k T} {s s₁ s₂ : LB.Sec (LPow A L k T)} (h₁ : IsActOf A L k θ s s₁) (h₂ : IsActOf A L k θ s s₂) (j) : s₁.f j = s₂.f j

theorem **pullSec_comp** (u' : T'' ⟶ T') (u : T' ⟶ T) (s : Secs A L a T) : pullSec u' (pullSec u s) = pullSec (u' ≫ u) s

theorem **pullSec_id** (hL : L.IsLB) (s : Secs A L a T) : pullSec (𝟙 T) s = s

theorem **eq_pullSec** {u : T' ⟶ T} {s : Secs A L a T} {s' : Secs A L a T'} (h : IsPullSec A L a (u ▷ A) (whiskerRight_snd _ _) s s') : s' = pullSec u s

theorem **pullSec_sum** {κ : Type} [Fintype κ] (u : T' ⟶ T) (c : κ → Γ(T.left, ⊤)) (b : κ → Secs A L a T) : pullSec u (∑ l, c l • b l) = ∑ l, u.left.appTop.hom (c l) • pullSec u (b l)

theorem **sum_f** {κ : Type} (F : Finset κ) (b : κ → Secs A L a T) (j) : LB.Sec.f (∑ l ∈ F, b l : Secs A L a T) j = ∑ l ∈ F, LB.Sec.f (b l) j

theorem **resHom_ι** (u : T' ⟶ T) (V : T.left.Opens) (V' : T'.left.Opens) (e : V' ≤ u.left ⁻¹ᵁ V) : resHom u V V' e ≫ overResι V = overResι V' ≫ u

theorem **inVVOver_iff** (s : Secs P.A P.L N T) : InVVOver P N m s ↔ s ∈ genSpan P N m T

theorem **pullSec_mem_genSpan** (u : T' ⟶ T) {s : Secs P.A P.L N T} (hs : s ∈ genSpan P N m T) : pullSec u s ∈ genSpan P N m T'

theorem **inVV_iff** (s : Secs P.A P.L N T) : InVV P N m s ↔ ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ pullSec (overResι V) s ∈ genSpan P N m (overRes V)

theorem **genSpan_res** (u : T' ⟶ T) {V : T.left.Opens} {V' : T'.left.Opens} (e : V' ≤ u.left ⁻¹ᵁ V) {s : Secs P.A P.L N T} (hs : pullSec (overResι V) s ∈ genSpan P N m (overRes V)) : pullSec (overResι V') (pullSec u s) ∈ genSpan P N m (overRes V')

theorem **secIncl_injective** (L : LB A.left) (a : ℤ) (T : Over S) : Function.Injective (secIncl A L a T)

theorem **span_eq_map** (L : LB A.left) (a : ℤ) (T : Over S) (Q : LB.Sec (LPow A L a T) → Prop) : Submodule.span Γ(T.left, ⊤) {f | ∃ h : f ∈ SecM A L a T, Q (SecM.toSec A ⟨f, h⟩)} = (Submodule.span Γ(T.left, ⊤) {s : Secs A L a T | Q s}).map (secIncl A L a T)

theorem **vvSub_eq_map** (T : Over S) : VVsub P N m T = (VV P N m T).map (secIncl P.A P.L N T)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.GeneratorSchemeProperties

theorem **res_pb** (f : Y ⟶ X) (U : X.Opens) (W W' : Y.Opens) (h : W ≤ f ⁻¹ᵁ U) (h' : W' ≤ W) (a : Γ(X, U)ˣ) : res h' (pb f U W h a) = pb f U W' (h'.trans h) a

theorem **pb_res** (f : Y ⟶ X) (U U' : X.Opens) (W : Y.Opens) (h : W ≤ f ⁻¹ᵁ U') (h' : U' ≤ U) (a : Γ(X, U)ˣ) : pb f U' W h (res h' a) = pb f U W (h.trans (f.preimage_mono h')) a

theorem **pb_pb** (g : Z ⟶ Y) (f : Y ⟶ X) (U : X.Opens) (V : Y.Opens) (W : Z.Opens) (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ g ⁻¹ᵁ V) (a : Γ(X, U)ˣ) : pb g V W h' (pb f U V h a) = pb (g ≫ f) U W (h'.trans (g.preimage_mono h)) a

theorem **pullU_eq** (f : Y ⟶ X) (U : X.Opens) (a : Γ(X, U)ˣ) : pullU f U a = pb f U _ le_rfl a

theorem **res_res** {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (a : Γ(X, U)ˣ) : res h' (res h a) = res (h'.trans h) a

theorem **res_self** {U : X.Opens} (h : U ≤ U) (a : Γ(X, U)ˣ) : res h a = a

theorem **pb_top_val** (f : Y ⟶ X) (h : ⊤ ≤ f ⁻¹ᵁ ⊤) (a : Γ(X, ⊤)ˣ) : (pb f ⊤ ⊤ h a).val = f.appTop.hom a.val

theorem **compat_extend** (hL : L.IsLB) (t : Γ(Z, ⊤)ˣ) : Compat L b (extend L b j hj t)

theorem **res_extend_self** (hL : L.IsLB) (t : Γ(Z, ⊤)ˣ) : res hj (extend L b j hj t j) = t

theorem **eq_extend** {s : ∀ i : L.ι, Γ(Z, b ⁻¹ᵁ L.U i)ˣ} (hs : Compat L b s) (i : L.ι) : s i = extend L b j hj (res hj (s j)) i

theorem **compat_pb** {G : Scheme.{0}} {q : G ⟶ B} {u : ∀ i : L.ι, Γ(G, q ⁻¹ᵁ L.U i)ˣ} (hu : Compat L q u) (g : Z ⟶ G) (hg : g ≫ q = b) : Compat L b fun i => pb g (q ⁻¹ᵁ L.U i) (b ⁻¹ᵁ L.U i) (by rw [← hg]; exact le_rfl) (u i)

theorem **exists_hom** (hN : ∀ k, (N k).IsLB) (hU : Univ N q u) {Z : Scheme.{0}} (b : Z ⟶ B) (j : ∀ k, (N k).ι) (hj : ∀ k, ⊤ ≤ b ⁻¹ᵁ (N k).U (j k)) (t : κ → Γ(Z, ⊤)ˣ) : ∃ (g : Z ⟶ G) (hg : g ≫ q = b), ∀ k, pb g (q ⁻¹ᵁ (N k).U (j k)) ⊤ (by subst hg; exact hj k) (u k (j k)) = t k

theorem **hom_ext** (hu : ∀ k, Compat (N k) q (u k)) (hU : Univ N q u) {Z : Scheme.{0}} (j : ∀ k, (N k).ι) (g g' : Z ⟶ G) (hg : g' ≫ q = g ≫ q) (hj : ∀ k, ⊤ ≤ g ⁻¹ᵁ q ⁻¹ᵁ (N k).U (j k)) (hj' : ∀ k, ⊤ ≤ g' ⁻¹ᵁ q ⁻¹ᵁ (N k).U (j k)) (h : ∀ k, pb g' (q ⁻¹ᵁ (N k).U (j k)) ⊤ (hj' k) (u k (j k)) = pb g (q ⁻¹ᵁ (N k).U (j k)) ⊤ (hj k) (u k (j k))) : g' = g

theorem **torus_hom_ext** {S : Scheme.{0}} {T : Over S} {a a' : T ⟶ splitTorus S (κ →₀ ℤ)} (h : ∀ k, val a (Finsupp.single k 1) = val a' (Finsupp.single k 1)) : a = a'

theorem **exists_torus_hom** {S : Scheme.{0}} {T : Over S} (t : κ → Γ(T.left, ⊤)ˣ) : ∃ a : T ⟶ splitTorus S (κ →₀ ℤ), ∀ k, val a (Finsupp.single k 1) = (t k).val

theorem **exists_isPullback** (hN : ∀ k, (N k).IsLB) (hu : ∀ k, Compat (N k) q (u k)) (hU : Univ N q u) {V : Scheme.{0}} (v : V ⟶ B) (j : ∀ k, (N k).ι) (hj : ∀ k, ⊤ ≤ v ⁻¹ᵁ (N k).U (j k)) : ∃ p : (splitTorus V (κ →₀ ℤ)).left ⟶ G, IsPullback p (splitTorus V (κ →₀ ℤ)).hom q v

theorem **prop_of_torus** [Fintype κ] (P : MorphismProperty Scheme.{0}) [IsZariskiLocalAtTarget P] (hP : ∀ V : Scheme.{0}, P (splitTorus V (κ →₀ ℤ)).hom) (hN : ∀ k, (N k).IsLB) (hu : ∀ k, Compat (N k) q (u k)) (hU : Univ N q u) : P q

theorem **generatorScheme_properties** : ∀ (B : Scheme.{0}) (κ : Type) [Fintype κ] (N : κ → LB B), (∀ k, (N k).IsLB) → ∀ (G : Scheme.{0}) (q : G ⟶ B) (u : ∀ k (i : (N k).ι), Γ(G, q ⁻¹ᵁ (N k).U i)ˣ), ((∀ k i i', res (Scheme.Hom.preimage_mono q (inf_le_right : (N k).U i ⊓ (N k).U i' ≤ (N k).U i')) (u k i') = pullU q _ ((N k).g i i') ∗ res (Scheme.Hom.preimage_mono q (inf_le_left : (N k).U i ⊓ (N k).U i' ≤ (N k).U i)) (u k i)) ∧ ∀ (Z : Scheme.{0}) (b : Z ⟶ B) (s : ∀ k (i : (N k).ι), Γ(Z, b ⁻¹ᵁ (N k).U i)ˣ), (∀ k i i', res (Scheme.Hom.preimage_mono b (inf_le_right : (N k).U i ⊓ (N k).U i' ≤ (N k).U i')) (s k i') = pullU b _ ((N k).g i i') ∗ res (Scheme.Hom.preimage_mono b (inf_le_left : (N k).U i ⊓ (N k).U i' ≤ (N k).U i)) (s k i)) → ∃! g : Z ⟶ G, ∃ hg : g ≫ q = b, ∀ k i, (g.appLE (q ⁻¹ᵁ (N k).U i) (b ⁻¹ᵁ (N k).U i) (by rw [← hg]; exact le_rfl)).hom (u k i).val = (s k i).val) → IsAffineHom q ∧ S

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.GlobalFnsHolds

theorem **globalFns_holds** : GlobalFnsSt

theorem **rigidity_holds** : RigiditySt

theorem **lemmaB_holds** : Semi.lemmaBSt.{0}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.GoodPtOfIrredOfMulPull

theorem **gs_zero** (f : X ⟶ Y) (y : X) (U : Y.Opens) : gs f y U 0 = 0

theorem **exists_ratio** (hL : L.IsLB) {φ : K → (Y ⟶ W)} {ε : K → ℤ} (p : Y ⟶ S₀) (hp : Function.Surjective p.appTop.hom) {u u' : LB.Gen (LB.LinComb L φ ε)} (hu : u.IsGen) (hu' : u'.IsGen) : ∃ a : Γ(S₀, ⊤), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J), Good L z J → (val u r z J).val = gs (r ≫ p) z ⊤ a ∗ (val u' r z J).val

theorem **sec_eq_zero_of_flat** (hL : L.IsLB) {φ : K → (Y ⟶ W)} {ε : K → ℤ} (π : X ⟶ Y) [Flat π] [Surjective π] (s : LB.Sec (LB.LinComb L φ ε)) (h : ∀ y J, Good L y J → sval s π y J = 0) (j : K → L.ι) : s.f j = 0

theorem **exists_gen_descend** (hL : L.IsLB) (π : X ⟶ Y) [Flat π] [Surjective π] {I : Type} (τ : I → (X ⟶ X)) (hτ : ∀ i, τ i ≫ π = π) (hd : ∀ (U : Y.Opens) (f : Γ(X, π ⁻¹ᵁ U)), (∀ i (e : π ⁻¹ᵁ U ≤ τ i ⁻¹ᵁ (π ⁻¹ᵁ U)), ((τ i).appLE _ _ e).hom f = f) → ∃ g, (π.app U).hom g = f) (ψ : K → (Y ⟶ W)) (ε : K → ℤ) (u : LB.Gen (LB.LinComb L (fun k => π ≫ ψ k) ε)) (hu : u.IsGen) (hinv : ∀ i y J, Good L y J → val u (τ i) y J = val u (𝟙 X) y J) : ∃ t : LB.Gen (LB.LinComb L ψ ε), t.IsGen

theorem **exists_invariant_gen** (hL : L.IsLB) (t : G → (Y ⟶ Y)) (t₁ : t 1 = 𝟙 Y) (tm : ∀ a b, t a ≫ t b = t (a ∗ b)) (p : Y ⟶ S₀) (tp : ∀ a, t a ≫ p = p) (hp : Function.Surjective p.appTop.hom) (M : Y ⟶ Y) (n : ℕ) (tM : ∀ a, t a ≫ M = M ≫ t (a ^ n)) (x : G) (u : LB.Gen (LB.LinComb L (K

theorem **toBaseOver_unique** {Z : Over S} (f g : Z ⟶ baseOver S) : f = g

theorem **comp_transl** {Z : Over S} (r : Z ⟶ A) (x : baseOver S ⟶ A) : r ≫ transl A x = (toBaseOver Z ≫ x) ∗ r

theorem **comp_mulA** {Z : Over S} (r : Z ⟶ A) (n : ℕ) : r ≫ mulA A n = r ^ n

theorem **transl_one** : transl A 1 = 𝟙 A

theorem **snd_transl** (x : baseOver S ⟶ A) : snd (baseOver S) A ≫ transl A x = translPt A x

theorem **unitInv_snd** : unitInv A ≫ snd (baseOver S) A = 𝟙 A

theorem **unitInv_mulT** (n : ℕ) : unitInv A ≫ mulT A (baseOver S) n = mulA A n ≫ unitInv A

theorem **ptSecT_snd** {T : Over S} (q : T ⟶ A) : ptSecT A q ≫ snd T A = q

theorem **ptSecT_translT** {T : Over S} (q x : T ⟶ A) : ptSecT A q ≫ translT A x = ptSecT A (x ∗ q)

theorem **ptSecT_mulT** {T : Over S} (q : T ⟶ A) (n : ℕ) : ptSecT A q ≫ mulT A T n = ptSecT A (q ^ n)

theorem **transl_comp** (x y : baseOver S ⟶ A) : transl A x ≫ transl A y = transl A (x ∗ y)

theorem **transl_mulA** (x : baseOver S ⟶ A) (n : ℕ) : transl A x ≫ mulA A n = mulA A n ≫ transl A (x ^ n)

theorem **powVal_act** {k : ℤ} {θ : ThetaGrp A L k T} {s s' : LB.Sec (LPow A L k T)} (h : IsActOf A L k θ s s') (r : Z ⟶ (T ⊗ A).left) {z : Z} {J : (Z ⟶ A.left) → L.ι} (hJ : Good L z J) : ∃ c, powVal s' r z J = c * powVal s (r ≫ (translT A θ.pt).left) z J

theorem **isUnit_powVal** {k : ℤ} {s : LB.Sec (LPow A L k T)} {x : T ⟶ A} {τ : LB.Gen (LB.LinComb L (Y

theorem **exists_frame_of_isUnit** [Subsingleton T.left] {k : ℤ} (s : LB.Sec (LPow A L k T)) (x : T ⟶ A) {z : T.left} {J : (T.left ⟶ A.left) → L.ι} (hJ : Good L z J) (h : IsUnit (powVal s (ptSecT A x).left z J)) : ∃ τ : LB.Gen (LB.LinComb L (Y

theorem **surjective_appTop** (P : PolAV (Spec (CommRingCat.of K))) : Function.Surjective P.A.hom.appTop.hom

theorem **exists_pow_eq** (a : baseOver (Spec (CommRingCat.of K)) ⟶ P.A) : ∃ b : baseOver (Spec (CommRingCat.of K)) ⟶ P.A, b ^ n = a

theorem **thetaGrp_pt_pow** [QuasiCompact (mulA P.A n).left] (α : LB.Gen (LB.LinComb P.L (K

theorem **goodPt_of_irred_of_mulPull** : IrredSt → (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

theorem **goodPt_of_irred** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → IrredSt → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ ρ : ThetaGrp P.A P.L k T → (SecM P.A P.L k T ≃ₗ[Γ(T.left, ⊤)] SecM P.A P.L k T), (∀ θ s s', IsActOf P.A P.L k θ s s' ↔ ρ θ (SecM.ofSec P.A s) = SecM.ofSec P.A s') ∧ (∀ θx θy c, CommIsOf P.A P.L k θx θy c → ∀ v, ρ θx (ρ θy v) = (c : Γ(T.left, ⊤)) • ρ θy (ρ θx v)) ∧ (∀ θ θ', θ.pt = θ'.pt → ∃ c : Γ(T.left, ⊤)ˣ, ∀ v, ρ θ' v = (c : Γ(T.left, ⊤)) • ρ θ v) ∧ ∀ θx θy, ∃ θ, θ.pt = θx.pt * θy.pt ∧ ∀ v, ρ θ v = ρ θx (ρ θy v)) → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), (∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((thetaLB P.A P.L N x).pull ((fst T P.A).left ⁻¹ᵁ V).ι), c.IsGen) ↔ x ^ N = 1) → (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.HasEnoughRootsOfUnityGlobalSections

theorem **existsUnique_pairing_eq_character** {H M : Type} [CommGroup H] [CommMonoid M] (N : ℕ) [NeZero N] [HasEnoughRootsOfUnity M N] [hfin : Finite {x : H // x ^ N = 1}] (W : H → H → Mˣ → Prop) (hex : ∀ x y : H, x ^ N = 1 → y ^ N = 1 → ∃! c, W x y c) (hmul : ∀ x x' y c c', W x y c → W x' y c' → W (x * x') y (c * c')) (hskew : ∀ x y c, W x y c → W y x c⁻¹) (hnd : ∀ x : H, x ^ N = 1 → (∀ y : H, y ^ N = 1 → W x y 1) → x = 1) (χ : {x : H // x ^ N = 1} → Mˣ) (hχ : ∀ x y, χ ⟨x.1 * y.1, by rw [mul_pow, x.2, y.2, _root_.one_mul]⟩ = χ x * χ y) : ∃! b : H, b ^ N = 1 ∧ ∀ x : {x : H // x ^ N = 1}, W x.1 b (χ x)

theorem **finite_torsion_of_isFinite_mul** (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (N : ℕ) [IsFinite (mulA P.A N).left] : Finite {x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A // x ^ N = 1}

theorem **hasEnoughRootsOfUnity_globalSections** (K : Type) [Field K] [IsAlgClosed K] (N : ℕ) (hN : (N : K) ≠ 0) : HasEnoughRootsOfUnity Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) N

theorem **charPair_of_weil** : WeilSt → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ x : T ⟶ P.A, x ^ k = 1 → WeilIs P.A P.L k x x 1) ∧ (∀ (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k y x c⁻¹) ∧ ∀ (T' : Over S) (h : T' ⟶ T) (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom c)) → WeilNondegSt → (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) → CharPairSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.InvariantFunctionsDescend

theorem **hom_ext** {Z : Scheme.{0}} {g h : Z ⟶ Spec (CommRingCat.of (Polynomial ℤ))} (e : g.appTop.hom coord = h.appTop.hom coord) : g = h

theorem **exists_hom** {Z : Scheme.{0}} (f : Γ(Z, ⊤)) : ∃ g : Z ⟶ Spec (CommRingCat.of (Polynomial ℤ)), g.appTop.hom coord = f

theorem **comp_appTop_apply** {X Y Z : Scheme.{0}} (g : X ⟶ Y) (h : Y ⟶ Z) (a : Γ(Z, ⊤)) : (g ≫ h).appTop.hom a = g.appTop.hom (h.appTop.hom a)

theorem **appTop_injective** {X Y : Scheme.{0}} (π : X ⟶ Y) [Epi π] : Function.Injective π.appTop.hom

theorem **exists_appTop_eq** {X Y : Scheme.{0}} (π : X ⟶ Y) [EffectiveEpi π] (f : Γ(X, ⊤)) (hf : ∀ (Z : Scheme.{0}) (g₁ g₂ : Z ⟶ X), g₁ ≫ π = g₂ ≫ π → g₁.appTop.hom f = g₂.appTop.hom f) : ∃ a, π.appTop.hom a = f

theorem **res_bijective** {X : Scheme.{0}} (U : X.Opens) : Function.Bijective (res U).hom

theorem **res_comp_appTop** {X Z : Scheme.{0}} (U : X.Opens) (g : Z ⟶ U.toScheme) (k : Z ⟶ X) (e : k = g ≫ U.ι) (h : ⊤ ≤ k ⁻¹ᵁ U) : res U ≫ g.appTop = k.appLE U ⊤ h

theorem **res_square** {X Y : Scheme.{0}} (π : X ⟶ Y) (U : Y.Opens) : res U ≫ (π ∣_ U).appTop = π.app U ≫ res (π ⁻¹ᵁ U)

theorem **res_square_apply** {X Y : Scheme.{0}} (π : X ⟶ Y) (U : Y.Opens) (a : Γ(Y, U)) : (π ∣_ U).appTop.hom ((res U).hom a) = (res (π ⁻¹ᵁ U)).hom ((π.app U).hom a)

theorem **app_injective** {X Y : Scheme.{0}} (π : X ⟶ Y) [Flat π] [Surjective π] (U : Y.Opens) : Function.Injective (π.app U).hom

theorem **isUnit_of_isUnit_app** {X Y : Scheme.{0}} (π : X ⟶ Y) [Surjective π] (U : Y.Opens) (a : Γ(Y, U)) (ha : IsUnit ((π.app U).hom a)) : IsUnit a

theorem **exists_app_eq** {X Y : Scheme.{0}} (π : X ⟶ Y) [Flat π] [Surjective π] [QuasiCompact π] (U : Y.Opens) (f : Γ(X, π ⁻¹ᵁ U)) (hf : ∀ (Z : Scheme.{0}) (g₁ g₂ : Z ⟶ X) (h₁ : ⊤ ≤ g₁ ⁻¹ᵁ (π ⁻¹ᵁ U)) (h₂ : ⊤ ≤ g₂ ⁻¹ᵁ (π ⁻¹ᵁ U)), g₁ ≫ π = g₂ ≫ π → (g₁.appLE _ ⊤ h₁).hom f = (g₂.appLE _ ⊤ h₂).hom f) : ∃ a, (π.app U).hom a = f

theorem **invariant_functions_descend** : ∀ (S : Scheme.{0}) (G B T : Over S) [GrpObj G] [GrpObj B] [GrpObj T] (q : G ⟶ B) (i : T ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → ∀ (U : B.left.Opens), Function.Injective (q.left.app U).hom ∧ (∀ a : Γ(B.left, U), IsUnit ((q.left.app U).hom a) → IsUnit a) ∧ ∀ f : Γ(G.left, q.left ⁻¹ᵁ U), (∀ (Z : Over S) (t : Z ⟶ T) (g : Z ⟶ G) (h₁ : ⊤ ≤ ((t ≫ i) * g).left ⁻¹ᵁ (q.left ⁻¹ᵁ U)) (h₂ : ⊤ ≤ g.left ⁻¹ᵁ (q.left ⁻¹ᵁ U)), (((t ≫ i) * g).left.appLE _ ⊤ h₁).hom f = (g.left.appLE _ ⊤ h₂).hom f) → ∃ a : Γ(B.left, U), (q.left.app U).hom a = f

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.InverseBundleGeneratorPointOfChartUnitsSections

theorem **isRepresentable_of_charts** (F : Scheme.{0}ᵒᵖ ⥤ Type) (hF : Presieve.IsSheaf Scheme.zariskiTopology F) {J : Type} (X : J → Scheme.{0}) (x : ∀ j, F.obj (op (X j))) (W : ∀ (_ : J) (Z : Scheme.{0}), F.obj (op Z) → Z.Opens) (hW : ∀ j (Z : Scheme.{0}) (y : F.obj (op Z)) (Z' : Scheme.{0}) (a : Z' ⟶ Z), Set.range a ⊆ (W j Z y : Set Z) ↔ ∃ g : Z' ⟶ X j, F.map g.op (x j) = F.map a.op y) (inj : ∀ j (Z : Scheme.{0}) (g g' : Z ⟶ X j), F.map g.op (x j) = F.map g'.op (x j) → g = g') (cov : ∀ (Z : Scheme.{0}) (y : F.obj (op Z)) (z : Z), ∃ j, z ∈ W j Z y) : F.IsRepresentable

theorem **exists_torus** (S : Scheme.{0}) (κ : Type) : ∃ (T : Scheme.{0}) (p : T ⟶ S) (t : κ → Γ(T, ⊤)ˣ), (∀ (Z : Scheme.{0}) (c : Z ⟶ S) (v : κ → Γ(Z, ⊤)ˣ), ∃ g : Z ⟶ T, g ≫ p = c ∧ ∀ k, g.appTop.hom (t k).val = (v k).val) ∧ ∀ (Z : Scheme.{0}) (g g' : Z ⟶ T), g ≫ p = g' ≫ p → (∀ k, g.appTop.hom (t k).val = g'.appTop.hom (t k).val) → g = g'

theorem **pb_val** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (U : Y.Opens) (W : Z.Opens) (h : W ≤ f ⁻¹ᵁ U) (x : Γ(Y, U)ˣ) : (pb f U W h x).val = (f.appLE U W h).hom x.val

theorem **pb_pb** {X Y Z : Scheme.{0}} (g : Z ⟶ Y) (f : Y ⟶ X) (U : X.Opens) (V : Y.Opens) (W : Z.Opens) (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ g ⁻¹ᵁ V) (x : Γ(X, U)ˣ) : pb g V W h' (pb f U V h x) = pb (g ≫ f) U W (h'.trans (g.preimage_mono h)) x

theorem **pb_congr** {Y Z : Scheme.{0}} {f f' : Z ⟶ Y} (e : f = f') (U : Y.Opens) (W : Z.Opens) (h : W ≤ f ⁻¹ᵁ U) (x : Γ(Y, U)ˣ) : pb f U W h x = pb f' U W (e ▸ h) x

theorem **res_eq_pb** {X : Scheme.{0}} {U V : X.Opens} (h : V ≤ U) (x : Γ(X, U)ˣ) : res h x = pb (𝟙 X) U V h x

theorem **pullU_eq_pb** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (U : Y.Opens) (x : Γ(Y, U)ˣ) : pullU f U x = pb f U _ le_rfl x

theorem **pb_id** {X : Scheme.{0}} {U : X.Opens} (h : U ≤ 𝟙 X ⁻¹ᵁ U) (x : Γ(X, U)ˣ) : pb (𝟙 X) U U h x = x

theorem **pb_top** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (h : ⊤ ≤ f ⁻¹ᵁ ⊤) (x : Γ(Y, ⊤)ˣ) : (pb f ⊤ ⊤ h x).val = f.appTop.hom x.val

theorem **pb_id_comp** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (U : Y.Opens) (V W : Z.Opens) (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ 𝟙 Z ⁻¹ᵁ V) (x : Γ(Y, U)ˣ) : pb (𝟙 Z) V W h' (pb f U V h x) = pb f U W (le_trans h' h) x

theorem **pb_comp_id** {Y Z : Scheme.{0}} (f : Z ⟶ Y) (U V : Y.Opens) (W : Z.Opens) (h : V ≤ 𝟙 Y ⁻¹ᵁ U) (h' : W ≤ f ⁻¹ᵁ V) (x : Γ(Y, U)ˣ) : pb f V W h' (pb (𝟙 Y) U V h x) = pb f U W (h'.trans (f.preimage_mono h)) x

theorem **pb_ι_injective** {Y : Scheme.{0}} (W : Y.Opens) (h : ⊤ ≤ W.ι ⁻¹ᵁ W) : Function.Injective (pb W.ι W ⊤ h)

theorem **cover_ext** {Y : Scheme.{0}} (𝒰 : Y.OpenCover) (V : Y.Opens) (x y : Γ(Y, V)ˣ) (h : ∀ i, pb (𝒰.f i) V _ le_rfl x = pb (𝒰.f i) V _ le_rfl y) : x = y

theorem **cover_glue** {Y : Scheme.{0}} (𝒰 : Y.OpenCover) (V : Y.Opens) (t : ∀ i, Γ(𝒰.X i, 𝒰.f i ⁻¹ᵁ V)ˣ) (ht : ∀ i j (Z : Scheme.{0}) (a : Z ⟶ 𝒰.X i) (a' : Z ⟶ 𝒰.X j), a ≫ 𝒰.f i = a' ≫ 𝒰.f j → ∀ h h', pb a _ ⊤ h (t i) = pb a' _ ⊤ h' (t j)) : ∃ s : Γ(Y, V)ˣ, ∀ i, pb (𝒰.f i) V _ le_rfl s = t i

theorem **Pt.compat'** {Z : Scheme.{0}} (y : Pt N Z) (k) (i i') {T : Scheme.{0}} (a : T ⟶ Z) (W : T.Opens) (h : W ≤ a ⁻¹ᵁ y.b ⁻¹ᵁ (N k).U i) (h' : W ≤ a ⁻¹ᵁ y.b ⁻¹ᵁ (N k).U i') : pb a _ W h' (y.s k i') = pb (a ≫ y.b) ((N k).U i ⊓ (N k).U i') W (le_inf h h') ((N k).g i i') ∗ pb a _ W h (y.s k i)

theorem **Pt.ext'** {Z : Scheme.{0}} {x y : Pt N Z} (hb : x.b = y.b) (hs : ∀ k i, x.s k i = pb (𝟙 Z) _ _ (by rw [hb]; exact le_rfl) (y.s k i)) : x = y

theorem **Pt.congr** {Z : Scheme.{0}} {x y : Pt N Z} (e : x = y) (k) (i) {T : Scheme.{0}} (a : T ⟶ Z) (W : T.Opens) (h h') : pb a _ W h (x.s k i) = pb a _ W h' (y.s k i)

theorem **genFunctor_map** {Z Z' : Scheme.{0}} (a : Z' ⟶ Z) (y : Pt N Z) : (genFunctor N).map a.op y = Pt.map a y

theorem **isSheaf_genFunctor** : Presieve.IsSheaf Scheme.zariskiTopology (genFunctor N)

theorem **Pt.inverseBundleGeneratorPoint_ofChartUnits_sections** (hN : ∀ k, (N k).IsLB) {Z : Scheme.{0}} (b : Z ⟶ B) (j : ∀ k, (N k).ι) (hb : ∀ k, ⊤ ≤ b ⁻¹ᵁ (N k).U (j k)) (v : κ → Γ(Z, ⊤)ˣ) (k) {T : Scheme.{0}} (a : T ⟶ Z) (h) : (pb a _ ⊤ h ((Pt.ofChart hN b j hb v).s k (j k))).val = a.appTop.hom (v k).val

theorem **Pt.ext_chart** {Z : Scheme.{0}} {x y : Pt N Z} (hb : x.b = y.b) (j : ∀ k, (N k).ι) (hx : ∀ k, ⊤ ≤ 𝟙 Z ⁻¹ᵁ x.b ⁻¹ᵁ (N k).U (j k)) (hy : ∀ k, ⊤ ≤ 𝟙 Z ⁻¹ᵁ y.b ⁻¹ᵁ (N k).U (j k)) (hs : ∀ k, pb (𝟙 Z) _ ⊤ (hx k) (x.s k (j k)) = pb (𝟙 Z) _ ⊤ (hy k) (y.s k (j k))) : x = y

theorem **isRepresentable_genFunctor** [Fintype κ] (hN : ∀ k, (N k).IsLB) : (genFunctor N).IsRepresentable

theorem **exists_generatorScheme** : ∀ (B : Scheme.{0}) (κ : Type) [Fintype κ] (N : κ → LB B), (∀ k, (N k).IsLB) → ∃ (G : Scheme.{0}) (q : G ⟶ B) (u : ∀ k (i : (N k).ι), Γ(G, q ⁻¹ᵁ (N k).U i)ˣ), (∀ k i i', res (Scheme.Hom.preimage_mono q (inf_le_right : (N k).U i ⊓ (N k).U i' ≤ (N k).U i')) (u k i') = pullU q _ ((N k).g i i') ∗ res (Scheme.Hom.preimage_mono q (inf_le_left : (N k).U i ⊓ (N k).U i' ≤ (N k).U i)) (u k i)) ∧ ∀ (Z : Scheme.{0}) (b : Z ⟶ B) (s : ∀ k (i : (N k).ι), Γ(Z, b ⁻¹ᵁ (N k).U i)ˣ), (∀ k i i', res (Scheme.Hom.preimage_mono b (inf_le_right : (N k).U i ⊓ (N k).U i' ≤ (N k).U i')) (s k i') = pullU b _ ((N k).g i i') ∗ res (Scheme.Hom.preimage_mono b (inf_le_left : (N k).U i ⊓ (N k).U i' ≤ (N k).U i)) (s k i)) → ∃! g : Z ⟶ G, ∃ hg : g ≫ q = b, ∀ k i, (g.appLE (q ⁻¹ᵁ (N k).U i) (b ⁻¹ᵁ (N k).U i) (by rw [← hg]; exact le_rfl)).hom (u k i).val = (s k i).val

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsAbelianSchemeThick

theorem **refines_self** (φ : K → (Y ⟶ X)) (ε : K → ℤ) : Refines (LB.LinComb L φ ε) L φ ε

theorem **Refines.div** {M M' : LB Y} {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (h : Refines M L φ ε) (h' : Refines M' L φ' ε') : Refines (LB.div M M') L (Sum.elim φ φ') (Sum.elim ε fun k => − ε' k)

theorem **Refines.linComb** {M : LB Y} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (h : Refines M L φ ε) (ψ : K' → (Y' ⟶ Y)) (δ : K' → ℤ) : Refines (LB.LinComb M ψ δ) L (fun p : K' × K => ψ p.1 ≫ φ p.2) (fun p => ε p.2 ∗ δ p.1)

theorem **Refines.exists_gen** (hL : L.IsLB) {M : LB Y} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (h : Refines M L φ ε) {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (hF : FormEq φ' ε' φ ε) (t : LB.Gen (LB.LinComb L φ' ε')) (ht : t.IsGen) : ∃ s : LB.Gen M, s.IsGen

theorem **isGen_one** : (LB.oneGen L (Y

theorem **exists_iso** (hL : L.IsLB) {φ φ' : K → (Y ⟶ X)} (h : φ = φ') (ε : K → ℤ) : ∃ w : LB.Iso (LB.LinComb L φ ε) (LB.LinComb L φ' ε), w.IsGen

theorem **gm_appLE** {Z : Scheme.{0}} (f : Z ⟶ Y) (U : Y.Opens) (h : ⊤ ≤ f ⁻¹ᵁ U) (a : Γ(Y, U)ˣ) (z : Z) : gm (𝟙 Z) z ⊤ (Units.map (f.appLE U ⊤ h).hom.toMonoidHom a) = gm f z U a

theorem **bijective_of_basis** {X : Type} [AddCommGroup X] {κ : Type} (b : Module.Basis κ ℤ X) {G B : Over S} [GrpObj G] [GrpObj B] (q : G ⟶ B) (i : splitTorus S X ⟶ G) (hq : IsMonHom q) (hi : IsMonHom i) (hf : Flat q.left) (hs : Surjective q.left) (ha : IsAffineHom q.left) (hk : IsPullback i (toUnit _) q (η[B])) (V : B.left.Opens) (hV : IsAffineOpen V) (u : κ → Γ(G.left, q.left ⁻¹ᵁ V)ˣ) (hu : ∀ (k : κ) (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (z : Z.left), q.left (g.left z) ∈ V → gm ((t ≫ i) ∗ g).left z _ (u k) = gm t.left z ⊤ (charUnit S X (b k)) ∗ gm g.left z _ (u k)) : Function.Bijective (AddMonoidAlgebra.liftNCRingHom (q.left.app V).hom ((Units.coeHom _).comp (AddMonoidHom.toMultiplicativeLeft (b.constr ℤ (fun k => Additive.ofMul (u k))).toAddMonoidHom)) (fun _ _ => Commute.all _ _))

theorem **exists_rig** {G B T : Scheme.{0}} (q : G ⟶ B) (eG : T ⟶ G) (eB : T ⟶ B) (e : eG ≫ q = eB) (M : LB B) (s : ∀ a : M.ι, Γ(G, q ⁻¹ᵁ M.U a)ˣ) (hs : ∀ (Z : Scheme.{0}) (r : Z ⟶ G) (z : Z) (a a' : M.ι), q (r z) ∈ M.U a → q (r z) ∈ M.U a' → gm r z _ (s a') = gm (r ≫ q) z _ (M.g a a') ∗ gm r z _ (s a)) : ∃ ρ : LB.Gen (M.pull eB), ρ.IsGen ∧ ∀ (a : M.ι) (y : T), q (eG y) ∈ M.U a → gm eG y _ (s a) ∗ gm (𝟙 _) y _ (ρ.s fun _ => a) = 1

theorem **isGt_of_units** (E : SplitExt S) {G : Over S} [GrpObj G] (q : G ⟶ E.B) (i : splitTorus S E.X ⟶ G) (hq : IsMonHom q) (hi : IsMonHom i) (hf : Flat q.left) (hs : Surjective q.left) (ha : IsAffineHom q.left) (hk : IsPullback i (toUnit _) q (η[E.B])) (s : ∀ (χ :

E.X) (a : (E.O χ).L.ι), Γ(G.left, q.left ⁻¹ᵁ (E.O χ).L.U a)ˣ) (h₁ : ∀ (χ : E.X) (Z : Scheme.{0}) (r : Z ⟶ G.left) (z : Z) (a a' : (E.O χ).L.ι), q.left (r z) ∈ (E.O χ).L.U a → q.left (r z) ∈ (E.O χ).L.U a' → gm r z _ (s χ a') = gm (r ≫ q.left) z _ ((E.O χ).L.g a a') * gm r z _ (s χ a)) (h₂ : ∀ (χ : E.X) (a : (E.O χ).L.ι) (Z : Over S) (t : Z ⟶ splitTorus S E.X) (g : Z ⟶ G) (z : Z.left), q.left (g.left z) ∈ (E.O χ).L.U a → gm ((t ≫ i) * g).left z _ (s χ a) = gm t.left z ⊤ (charUnit S E.X χ) * gm g.left z _ (s χ a)) (h₃ : ∀ (χ : E.X) (a : (E.O χ).L.ι) (y : (𝟙_ (Over S)).left), q.left (eSec G y) ∈ (E.O χ).L.U a → gm (eSec G) y _ (s χ

theorem **prod_square** {S : Scheme.{0}} {B T : Over S} [GrpObj B] (x : Fin 2 → (T ⟶ B)) {U : Type} [CommGroup U] (F : (T ⟶ B) → U) : ∏ J : ULift.{0} (Finset (Fin 2)), F (psum B x J.down) ^ ((-1 : ℤ) ^ (2 - J.down.card)) = F 1 * (F (x 0))⁻¹ * (F (x 1))⁻¹ * F (x 0 * x 1)

theorem **prod_O** {T : Over (Spec (CommRingCat.of R))} (p : T ⟶ D.P.A) (χ : D.X) {U : Type} [CommGroup U] (h : (T.left ⟶ D.P.A.left) → U) : ∏ k, h (D.Oφ p χ k) ^ Oε k = h ((toUnit T ≫ D.c χ) * p).left * (h (toUnit T ≫ D.c χ).left)⁻¹ * (h p.left)⁻¹ * h (1 : T ⟶ D.P.A).left

theorem **c_add** (T : Over (Spec (CommRingCat.of R))) (χ χ' : D.X) : toUnit T ≫ D.c (χ + χ') = (toUnit T ≫ D.c χ) * (toUnit T ≫ D.c χ')

theorem **c_zero** (T : Over (Spec (CommRingCat.of R))) : toUnit T ≫ D.c 0 = 1

theorem **bundle_add** (χ χ' : D.X) : ∃ t : LB.Iso (bundle I D e χ') (LB.div (bundle I D e (χ + χ')) (bundle I D e χ)), t.IsGen

theorem **bundle_zero** : ∃ t : LB.Gen (bundle I D e 0), t.IsGen

theorem **bundle_primitive** (χ : D.X) : ∃ t : LB.Gen (abelian₂ (thickObj R I e D.P.A) (bundle I D e χ)), t.IsGen

theorem **isAbelianScheme_thick** : IsAbelianScheme (thickObj R I e D.P.A)

theorem **q_Oφ** (χ : D.X) (k : Fin 4) : g.q.left ≫ D.Oφ (D.prBe I e) χ k = D.Oφ g.pG χ k

theorem **mem_chart** (χ : D.X) (a : Fin 4 → D.P.L.ι) (y : (thickObj R I e G).left) (hy : g.q.left y ∈ (bundle I D e χ).U a) : y ∈ LB.wchart D.P.L (D.Oφ g.pG χ) a

theorem **gm_coordUnit** (χ : D.X) (a : (bundle I D e χ).ι) {Z : Scheme.{0}} (r : Z ⟶ (thickObj R I e G).left) (z : Z) (h : g.q.left (r z) ∈ (bundle I D e χ).U a) : gm r z _ (coordUnit g χ a) = gm r z (LB.wchart D.P.L (D.Oφ g.pG χ) a) ((g.t χ).s a)

theorem **coordUnit_trans** (χ : D.X) (Z : Scheme.{0}) (r : Z ⟶ (thickObj R I e G).left) (z : Z) (a a' : (bundle I D e χ).ι) (h : g.q.left (r z) ∈ (bundle I D e χ).U a) (h' : g.q.left (r z) ∈ (bundle I D e χ).U a') : gm r z _ (coordUnit g χ a') = gm (r ≫ g.q.left) z _ ((bundle I D e χ).g a a') * gm r z _ (coordUnit g χ a)

theorem **coordUnit_weight** (χ : D.X) (a : (bundle I D e χ).ι) (Z : Over (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))) (t : Z ⟶ splitTorus _ D.X) (y : Z ⟶ thickObj R I e G) (z : Z.left) (h : g.q.left (y.left z) ∈ (bundle I D e χ).U a) : gm ((t ≫ g.i) * y).left z _ (coordUnit g χ a) = gm t.left z ⊤ (charUnit _ D.X χ) * gm y.left z _ (coordUnit g χ a)

theorem **gtCoord_isGt** : Geo.Tor.GtCoordToGtSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsAffineHomFromSpecResidueField

theorem **exists_pow_mul_eq_zero** {k R : Type} [CommRing k] [CommRing R] (hk : IsField k) (φ : k →+* R) (a : R) (ha : φ.IsIntegralElem a) : ∃ (m : ℕ) (c b : R), IsUnit c ∧ a ^ m * (c + a * b) = 0

theorem **isClosed_basicOpen** {Z : Scheme.{0}} {K : Type} [Field K] (f : Z ⟶ Spec (.of K)) [UniversallyClosed f] (a : Γ(Z, ⊤)) : IsClosed (Z.basicOpen a : Set Z)

theorem **locallyQuasiFinite_of_cover** {Z : Scheme.{0}} {K : Type} [Field K] (f : Z ⟶ Spec (.of K)) [IsProper f] {I : Type} (a : I → Γ(Z, ⊤)) (hc : ⊔ i, Z.basicOpen (a i) = ⊤) (ha : ∀ i, IsAffineOpen (Z.basicOpen (a i))) : LocallyQuasiFinite f

theorem **isAffineHom_fromSpecResidueField** (Y : Scheme.{0}) (y : Y) : IsAffineHom (Y.fromSpecResidueField y)

theorem **isUnit_gs** {Z Y : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (h : r z ∈ U) (a : Γ(Y, U)) : IsUnit (gs r z U a) ↔ r z ∈ Y.basicOpen a

theorem **gs_appLE** {Z Y : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (V : Z.Opens) (hle : V ≤ r ⁻¹ᵁ U) (h : z ∈ V) (a : Γ(Y, U)) : gs (𝟙 Z) z V ((r.appLE U V hle).hom a) = gs r z U a

theorem **gs_pow** {Z Y : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (h : r z ∈ U) (a : Γ(Y, U)) (n : ℕ) : gs r z U (a ^ n) = gs r z U a ^ n

theorem **pow_aux** {R : Type} [CommRing R] (g t : Rˣ) (x : R) (e N : ℕ) : ((g ^ N).val * x) ^ e * ((g ^ e * t)⁻¹).val ^ N = x ^ e * (t⁻¹).val ^ N

theorem **exists_fn** {X W Z : Scheme.{0}} (L : LB X) (hL : L.IsLB) (w : W ⟶ X) (N : ℕ) (s : LB.Sec (LB.LinComb L (K

theorem **isFinite_of_relAmple** {S : Scheme.{0}} (A : Over S) [GrpObj A] [IsProper A.hom] (L : LB A.left) (hL : L.IsLB) (hA : RelAmple A L) (m : A ⟶ A) (e : ℕ) (he : e ≠ 0) (α : LB.Gen (LB.LinComb L (K

theorem **isFinite_mulA** : (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsAffineHomOfFlatQuotientOfAffineKernel

theorem **_root_.GlobalTheory.quasiCompact_descendsAlong_fpqc** : MorphismProperty.DescendsAlong (@QuasiCompact : MorphismProperty Scheme.{u}) (@Surjective ⊓ @Flat ⊓ @QuasiCompact)

theorem **_root_.GlobalTheory.quasiSeparated_descendsAlong_fpqc** : MorphismProperty.DescendsAlong (@QuasiSeparated : MorphismProperty Scheme.{u}) (@Surjective ⊓ @Flat ⊓ @QuasiCompact)

theorem **_root_.GlobalTheory.isAffine_of_isPullback_of_flat_surjective** {X Y S T : Scheme.{u}} {f : T ⟶ S} {g : Y ⟶ X} {p : X ⟶ S} {q : Y ⟶ T} (H : IsPullback g q p f) [IsAffine S] [IsAffine T] [Flat f] [Surjective f] [IsAffine Y] : IsAffine X

theorem **_root_.GlobalTheory.isAffineHom_descendsAlong_fpqc** : MorphismProperty.DescendsAlong (@IsAffineHom : MorphismProperty Scheme.{u}) (@Surjective ⊓ @Flat ⊓ @QuasiCompact)

theorem **_root_.GlobalTheory.isAffineHom_of_flatQuotient_of_affine_kernel** : ∀ (S : Scheme.{0}) (G B T : Over S) [GrpObj G] [GrpObj B] [GrpObj T] (q : G ⟶ B) (i : T ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → IsAffineHom T.hom → IsAffineHom q.left

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsAffineOpenAndBijectiveOfWeightUnits

theorem **exists_isPullback** {C : Type*} [Category C] [CartesianMonoidalCategory C] {T G B W V : C} [GrpObj T] [GrpObj G] [GrpObj B] (q : G ⟶ B) (i : T ⟶ G) [IsMonHom q] [IsMonHom i] (hk : IsPullback i (toUnit _) q η[B]) {j : W ⟶ G} {k : V ⟶ B} {q' : W ⟶ V} (hW : IsPullback j q' q k) [Mono j] (τ : W ⟶ T) (hτ : ∀ (Z : C) (t : Z ⟶ T) (g g' : Z ⟶ W), g' ≫ j = (t ≫ i) * (g ≫ j) → g' ≫ τ = t * (g ≫ τ)) : ∃ a : T ⊗ W ⟶ W, IsPullback (𝟙 _) a (T ◁ q') (lift τ q')

theorem **isAffine_and_bijective** {S : Scheme.{0}} {X : Type} [AddCommGroup X] {W V : Over S} [IsAffine V.left] (τ : W ⟶ splitTorus S X) (q : W ⟶ V) [IsIso (lift τ q)] {A : CommRingCat.{0}} (e : A ≅ Γ(V.left, ⊤)) (f : AddMonoidAlgebra A X →+* Γ(W.left, ⊤)) (hf₁ : ∀ a :

A, f (AddMonoidAlgebra.single 0 a) = q.left.appTop.hom (e.hom.hom a)) (hf₂ : ∀ χ : X, f (AddMonoidAlgebra.single χ 1) = τ.left.appTop.hom (charUnit S X χ).val) : IsAffine W.left ∧ Function.Bijective f

theorem **app_comp_topIso_inv** {Y Z : Scheme.{0}} (f : Y ⟶ Z) (V : Z.Opens) : f.app V ≫ (f ⁻¹ᵁ V).topIso.inv = V.topIso.inv ≫ (f ∣_ V).appTop

theorem **cover_of_isPullback** {P Y Z W : Scheme.{0}} {a : P ⟶ Y} {b : P ⟶ Z} {f : Y ⟶ W} {g : Z ⟶ W} (h : IsPullback a b f g) (hf : (@Surjective ⊓ @Flat ⊓ @QuasiCompact : MorphismProperty Scheme.{0}) f) : (@Surjective ⊓ @Flat ⊓ @QuasiCompact : MorphismProperty Scheme.{0}) b

theorem **isAffineOpen_and_bijective_of_weight_units** : ∀ (S : Scheme.{0}) (X : Type) [AddCommGroup X] (G B : Over S) [GrpObj G] [GrpObj B] (q : G ⟶ B) (i : splitTorus S X ⟶ G), IsMonHom q → IsMonHom i → Flat q.left → Surjective q.left → QuasiCompact q.left → IsPullback i (toUnit _) q (η[B]) → ∀ (V : B.left.Opens), IsAffineOpen V → ∀ u : Multiplicative X →* Γ(G.left, q.left ⁻¹ᵁ V)ˣ, (∀ (χ : X) (Z : Over S) (t : Z ⟶ splitTorus S X) (g : Z ⟶ G) (h₁ : ⊤ ≤ ((t ≫ i) ∗ g).left ⁻¹ᵁ (q.left ⁻¹ᵁ V)) (h₂ : ⊤ ≤ g.left ⁻¹ᵁ (q.left ⁻¹ᵁ V)), (((t ≫ i) ∗ g).left.appLE _ ⊤ h₁).hom (u (Multiplicative.ofAdd χ)).val = t.left.appTop.hom (charUnit S X χ).val ∗ (g.left.appLE _ ⊤ h₂).hom (u (Multiplicative.ofAdd χ)).val) → IsAffineOpen (q.left ⁻¹ᵁ V) ∧ Function.Bijective (AddMonoidAlgebra.liftNCRingHom (q.left.app V).hom ((Units.coeHom _).comp u) (fun _ _ => Commute.all _ _))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsAffineOpenInfOfIsSeparated

theorem **lid_rTensor_lTensor** {R C A B : Type} [CommRing R] [AddCommGroup C] [Module R C] [AddCommGroup A] [Module R A] [AddCommGroup B] [Module R B] (l : C →ₗ[R] R) (f : B →ₗ[R] A) (w : C ⊗[R] B) : TensorProduct.lid R A (l.rTensor A (f.lTensor C w)) = f (TensorProduct.lid R B (l.rTensor B w))

theorem **apply_lid_rTensor** {R C B : Type} [CommRing R] [AddCommGroup C] [Module R C] [AddCommGroup B] [Module R B] (l : C →ₗ[R] R) (f : B →ₗ[R] R) (w : C ⊗[R] B) : f (TensorProduct.lid R B (l.rTensor B w)) = l (TensorProduct.rid R C (f.lTensor C w))

theorem **isAffineOpen_prod** [IsAffine S] {U : X.Opens} {W : Y.Opens} (hU : IsAffineOpen U) (hW : IsAffineOpen W) : IsAffineOpen (p ⁻¹ᵁ U ⊓ q ⁻¹ᵁ W)

theorem **exists_splits_preimage** [IsAffine S] (hR : IsField Γ(S, ⊤)) [CompactSpace X] (hX : ∀ U U' : X.Opens, IsAffineOpen U → IsAffineOpen U' → IsAffineOpen (U ⊓ U')) (hY : ∀ U U' : Y.Opens, IsAffineOpen U → IsAffineOpen U' → IsAffineOpen (U ⊓ U')) (hiY : Function.Surjective iY.appTop.hom) {y : S ⟶ Y} (hy : y ≫ iY = 𝟙 S) {s : X ⟶ Z} (hs : s ≫ p = 𝟙 X) (hs' : s ≫ q = iX ≫ y) (W₀ : Y.affineOpens) (hW₀ : ⊤ ≤ y ⁻¹ᵁ W₀.1) (hV : ⊔ a, V a = ⊤) (hφ : IsCocycle V φ) (ρ : ∀ a, Γ(X, s ⁻¹ᵁ V a)) (hρ : ∀ a b, (s.app (V a ⊓ V b)).hom (φ a b) = rs (s.preimage_mono inf_le_left) (ρ a) − rs (s.preimage_mono inf_le_right) (ρ b)) (W : Y.affineOpens) : ∃ τ, Splits V φ (q ⁻¹ᵁ W.1) τ

theorem **exists_splits_top** [IsAffine S] [Flat iY] [CompactSpace X] [QuasiSeparatedSpace X] (hY : ∀ U U' : Y.Opens, IsAffineOpen U → IsAffineOpen U' → IsAffineOpen (U ⊓ U')) (hiX : Function.Bijective iX.appTop.hom) {t : Y ⟶ Z} (ht : t ≫ q = 𝟙 Y) (hV : ⊔ a, V a = ⊤) (ρ : ∀ a, Γ(Y, t ⁻¹ᵁ V a)) (hρ : ∀ a b, (t.app (V a ⊓ V b)).hom (φ a b) = rs (t.preimage_mono inf_le_left) (ρ a) − rs (t.preimage_mono inf_le_right) (ρ b)) (hτ : ∀ W : Y.affineOpens, ∃ τ, Splits V φ (q ⁻¹ᵁ W.1) τ) : ∃ χ, Splits V φ ⊤ χ

theorem **isAffineOpen_inf_of_isSeparated** {X S : Scheme.{0}} [IsAffine S] (f : X ⟶ S) [IsSeparated f] (U U' : X.Opens) (hU : IsAffineOpen U) (hU' : IsAffineOpen U') : IsAffineOpen (U ⊓ U')

theorem **kuenneth_injective** : ∀ (K : Type) [Field K] (X Y : Over (Spec (CommRingCat.of K))) (x₀ : 𝟙_ _ ⟶ X) (y₀ : 𝟙_ _ ⟶ Y), IsSeparated X.hom → QuasiCompact X.hom → IsSeparated Y.hom → QuasiCompact Y.hom → Function.Bijective X.hom.appTop.hom → Function.Bijective Y.hom.appTop.hom → ∀ (I : Type) (V : I → (X ⊗ Y).left.Opens), ⨆ a, V a = ⊤ → ∀ φ : ∀ a b, Γ((X ⊗ Y).left, V a ⊓ V b), (∀ a b d, ((X ⊗ Y).left.presheaf.map (homOfLE (inf_le_left : V a ⊓ V b ⊓ V d ≤ V a ⊓ V b)).op).hom (φ a b) + ((X ⊗ Y).left.presheaf.map (homOfLE (le_inf (inf_le_left.trans inf_le_right) inf_le_right : V a ⊓ V b ⊓ V d ≤ V b ⊓ V d)).op).hom (φ b d) = ((X ⊗ Y).left.presheaf.map (homOfLE (le_inf (inf_le_left.trans inf_le_left) inf_le_right : V a ⊓ V b ⊓ V d ≤ V a ⊓ V d)).op).hom (φ a d)) → (∃ ψ : ∀ a, Γ(X.left, (lift (𝟙 X) (toUnit X ≫ y₀)).left ⁻¹ᵁ V a), ∀ a b, ((lift (𝟙 X) (toUnit X ≫ y₀)).left.app (V a ⊓ V b)).hom (

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsFieldGlobalFnsSpec

theorem **heis_irred_of_isField** (H : ∀ (K G V : Type) [Field K] [CommGroup G] [Finite G] [AddCommGroup V] [Module K V] [Module.Finite K V] (U : G → V ≃ₗ[K] V) (e : G → G →* Kˣ), (∀ x y v, U x (U y v) = (e x y : K) • U y (U x v)) → Function.Injective e → Module.finrank K V ^ 2 ≤ Nat.card G → ∀ W : Submodule K V, (∀ x v, v ∈ W → U x v ∈ W) → W = ⊥ ∨ W = ⊤) (R G V : Type) [CommRing R] (hR : IsField R) [CommGroup G] [Finite G] [AddCommGroup V] [Module R V] (hpos : 0 < Module.finrank R V) (U : G → V ≃ₗ[R] V) (e : G → G →* Rˣ) (h1 : ∀ x y v, U x (U y v) = (e x y : R) • U y (U x v)) (h2 : Function.Injective e) (h3 : Module.finrank R V ^ 2 ≤ Nat.card G) (W : Submodule R V) (hW : ∀ x v, v ∈ W → U x v ∈ W) : W = ⊥ ∨ W = ⊤

theorem **isField_globalFns_spec** (K : Type) [Field K] : IsField Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)

theorem **irredSt_of_heis** : (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ ρ : ThetaGrp P.A P.L k T → (SecM P.A P.L k T ≃ₗ[Γ(T.left, ⊤)] SecM P.A P.L k T), (∀ θ s s', IsActOf P.A P.L k θ s s' ↔ ρ θ (SecM.ofSec P.A s) = SecM.ofSec P.A s') ∧ (∀ θx θy c, CommIsOf P.A P.L k θx θy c → ∀ v, ρ θx (ρ θy v) = (c : Γ(T.left, ⊤)) • ρ θy (ρ θx v)) ∧ (∀ θ θ', θ.pt = θ'.pt → ∃ c : Γ(T.left, ⊤)ˣ, ∀ v, ρ θ' v = (c : Γ(T.left, ⊤)) • ρ θ v) ∧ ∀ θx θy, ∃ θ, θ.pt = θx.pt ∗ θy.pt ∧ ∀ v, ρ θ v = ρ θx (ρ θy v)) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ (θx θy : ThetaGrp P.A P.L k T) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k θx.pt θy.pt c ↔ CommIsOf P.A P.L k θx θy c) ∧ (∀ x : T ⟶ P.A, x ^ k = 1 → Subsingleton T.left → ∃ θ : ThetaGrp P.A P.L k T, θ.pt = x ∧ (Odd k → SymLift P.A P.L k k x θ))) → (∀ (K G V : Type) [Field K] [CommGroup G] [Finite G] [AddCommGroup V] [Module K V] [Module.Fin

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsIntegralOfSmoothFieldOfConnectedSpace

theorem **isIntegral_of_smooth_field_of_connectedSpace** {K : Type u} [Field K] {Z : Scheme.{u}} (g : Z ⟶ Spec (.of K)) [Smooth g] [ConnectedSpace Z] : IsIntegral Z

theorem **geometricallyIntegral_of_smooth_of_geometricallyConnected_fibres** {X S : Scheme.{u}} (f : X ⟶ S) [Smooth f] [GeometricallyConnected f] : GeometricallyIntegral f

theorem **grpObj_geometricallyIntegral_of_smooth** (S : Scheme.{0}) (H : Over S) [GrpObj H] [Smooth H.hom] [GeometricallyConnected H.hom] : GeometricallyIntegral H.hom

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsIsoAppTopOfIsAffine

theorem **baseChange_model** {R₀ R k B₀ B C₀ C : Type} [CommRing R₀] [CommRing R] [CommRing k] [Algebra R₀ R] [Algebra R k] [Algebra R₀ k] [IsScalarTower R₀ R k] [AddCommGroup B₀] [Module R₀ B₀] [AddCommGroup C₀] [Module R₀ C₀] [AddCommGroup B] [Module R B] [AddCommGroup C] [Module R C] (β : R ⊗[R₀] B₀ ≃ₗ[R] B) (δ : R ⊗[R₀] C₀ ≃ₗ[R] C) (u₀ : B₀ →ₗ[R₀] C₀) (u : B →ₗ[R] C) (h : ∀ b, u (β (1 ⊗ₜ b)) = δ (1 ⊗ₜ u₀

b)) (y : k ⊗[R₀] B₀) : u.baseChange k (β.toLinearMap.baseChange k ((AlgebraTensorModule.cancelBaseChange R₀ R k k B₀).symm y)) = δ.toLinearMap.baseChange k ((AlgebraTensorModule.cancelBaseChange R₀ R k k C₀).symm (u₀.baseChange k y))

theorem **model_apply** {R₀ R B₀ B C₀ C : Type} [CommRing R₀] [CommRing R] [Algebra R₀ R] [AddCommGroup B₀] [Module R₀ B₀] [AddCommGroup C₀] [Module R₀ C₀] [AddCommGroup B] [Module R B] [AddCommGroup C] [Module R C] (β : R ⊗[R₀] B₀ ≃ₗ[R] B) (δ : R ⊗[R₀] C₀ ≃ₗ[R] C) (u₀ : B₀ →ₗ[R₀] C₀) (u : B →ₗ[R] C) (h : ∀ b, u (β (1 ⊗ₜ b)) = δ (1 ⊗ₜ u₀ b)) (y : R ⊗[R₀] B₀) : u (β y) = δ (u₀.baseChange R y)

theorem **eq_zero_of_fibres** (hI : FibreInjective) (hD : SmoothDescent) {R B C : Type} [CommRing R] [CommRing B] [CommRing C] [Algebra R B] [Algebra R C] [Algebra.Smooth R B] [Algebra.Smooth R C] (f g : B →ₐ[R] C) (r : B →ₐ[R] R) (hk : ∀ (k : Type) [Field k] [Algebra R k] (x : k ⊗[R] B), f.toLinearMap.baseChange k x = g.toLinearMap.baseChange k x → r.toLinearMap.baseChange k x = 0 → x = 0) (b : B) (hb : f b = g b) (hr : r b = 0) : b = 0

theorem **exists_eq_of_eq** (i : Spec B ⟶ X) [EffectiveEpi i] {f g : B ⟶ C} (H : IsPullback (Spec.map f) (Spec.map g) i i) (x : B) (hx : f x = g x) : ∃ a : Γ(X, ⊤), (i.appTop ≫ (Scheme.ΓSpecIso B).hom) a = x

theorem **appTop_comp_eq** {p : X ⟶ Spec R} {i : Spec B ⟶ X} {φ : R ⟶ B} (hφ : i ≫ p = Spec.map φ) : p.appTop ≫ i.appTop ≫ (Scheme.ΓSpecIso B).hom = (Scheme.ΓSpecIso R).hom ≫ φ

theorem **eq_zero_of_surjective** {p : X ⟶ Spec R} (hp : Function.Surjective p.appTop) (i : Spec B ⟶ X) [EffectiveEpi i] {f g : B ⟶ C} (H : IsPullback (Spec.map f) (Spec.map g) i i) {φ : R ⟶ B} (hφ : i ≫ p = Spec.map φ) {R' : CommRingCat.{0}} (r : B ⟶ R') (hr : Function.Injective (φ ≫ r)) (x : B) (hx : f x = g x) (hrx : r x = 0) : x = 0

theorem **isPullback_pr** (R k B : Type) [CommRing R] [CommRing k] [CommRing B] [Algebra R k] [Algebra R B] : IsPullback (prR R k B) (prL R k B) (Spec.map (CommRingCat.ofHom (algebraMap R B))) (Spec.map (CommRingCat.ofHom (algebraMap R k)))

theorem **bc_apply** (k : Type) [CommRing k] [Algebra R k] (f : B →ₐ[R] C) (x : k ⊗[R] B) : bc k f x = f.toLinearMap.baseChange k x

theorem **bc_prR** (k : Type) [CommRing k] [Algebra R k] (f : B →ₐ[R] C) : Spec.map (bc k f) ≫ prR R k B = prR R k C ≫ Spec.map (CommRingCat.ofHom f.toRingHom)

theorem **bc_prL** (k : Type) [CommRing k] [Algebra R k] (f : B →ₐ[R] C) : Spec.map (bc k f) ≫ prL R k B = prL R k C

theorem **specMap_algHom** (f : B →ₐ[R] C) : Spec.map (CommRingCat.ofHom f.toRingHom) ≫ Spec.map (CommRingCat.ofHom (algebraMap R B)) = Spec.map (CommRingCat.ofHom (algebraMap R C))

theorem **eq_zero_fibre** (p : X ⟶ Spec (.of R)) [UniversallyClosed p] [GeometricallyIntegral p] (i : Spec (.of B) ⟶ X) [Flat i] [Surjective i] [QuasiCompact i] (f g : B →ₐ[R] C) (H : IsPullback (Spec.map (CommRingCat.ofHom f.toRingHom)) (Spec.map (CommRingCat.ofHom g.toRingHom)) i i) (hφ : i ≫ p = Spec.map (CommRingCat.ofHom (algebraMap R B))) (r : B →ₐ[R] R) (k : Type) [Field k] [Algebra R k] (x : k ⊗[R] B) (hx : f.toLinearMap.baseChange k x = g.toLinearMap.baseChange k x) (hr : r.toLinearMap.baseChange k x = 0) : x = 0

theorem **isIso_appTop_of_cover** (hI : FibreInjective) (hD : SmoothDescent) [Algebra.Smooth R B] [Algebra.Smooth R C] (p : X ⟶ Spec (.of R)) [UniversallyClosed p] [GeometricallyIntegral p] (i : Spec (.of B) ⟶ X) [Flat i] [Surjective i] [QuasiCompact i] (f g : B →ₐ[R] C) (H : IsPullback (Spec.map (CommRingCat.ofHom f.toRingHom)) (Spec.map (CommRingCat.ofHom g.toRingHom)) i i) (hφ : i ≫ p = Spec.map (CommRingCat.ofHom (algebraMap R B))) (r : B →ₐ[R] R) : IsIso p.appTop

theorem **isIso_appTop_of_section** (hI : FibreInjective) (hD : SmoothDescent) (p : X ⟶ S) [IsAffine S] [Smooth p] [IsProper p] [GeometricallyIntegral p] (i : T ⟶ X) [IsAffine T] [Smooth i] [Surjective i] (s : S ⟶ T) (hs : s ≫ i ≫ p = 𝟙 S) : IsIso p.appTop

theorem **isIso_appTop_of_isAffine** (hI : FibreInjective) (hD : SmoothDescent) (p : X ⟶ S) [IsAffine S] [Smooth p] [IsProper p] [GeometricallyIntegral p] : IsIso p.appTop

theorem **isIso_app** (hI : FibreInjective) (hD : SmoothDescent) (p : X ⟶ S) [Smooth p] [IsProper p] [GeometricallyIntegral p] (U : S.Opens) : IsIso (p.app U)

theorem **globalFns_of** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (R₀ R C₀ C₁ : Type) [CommRing R₀] [IsNoetherianRing R₀] [CommRing R] [Algebra R₀ R] [AddCommGroup C₀] [Module R₀ C₀] [AddCommGroup C₁] [Module R₀ C₁] [Module.Flat R₀ C₀] [Module.Flat R₀ C₁] (φ : C₀ →ₗ[R₀] C₁), (∀ (k : Type) [Field k] [Algebra R k] [Algebra R₀ k] [IsScalarTower R₀ R k], Function.Injective (φ.baseChange k)) → Function.Injective (φ.baseChange R)) → (∀ (R B C : Type) [CommRing R] [CommRing B] [CommRing C] [Algebra R B] [Algebra R C] [Algebra.Smooth R B] [Algebra.Smooth R C] (f g : B →ₐ[R] C) (r : B →ₐ[R] R), ∃ (R₀ B₀ C₀ : Type) (_ : CommRing R₀) (_ : IsNoetherianRing R₀) (_ : Algebra R₀ R) (_ : CommRing B₀) (_ : CommRing C₀) (_ : Algebra R₀ B₀) (_ : Algebra R₀ C₀) (f₀ g₀ : B₀ →ₐ[R₀] C₀) (r₀ : B₀ →ₐ[R₀] R₀) (β' : TensorProduct R₀ R B₀ ≃ₐ[R] B) (δ' : Te

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsIsoOfIsClosedImmersionOfThickenings

theorem **_root_.GlobalTheory.ThickeningClosed.ker_ideal_le_ker** {X Z : Scheme.{0}} (ι : Z ⟶ X) {U : X.Opens} (hU : IsAffineOpen U) {D : CommRingCat.{0}} (ψ : Γ(X, U) ⟶ D) (l : Spec D ⟶ Z) (hl : l ≫ ι = Spec.map ψ ≫ hU.fromSpec) : ι.ker.ideal ⟨U, hU⟩ ≤ RingHom.ker ψ.hom

theorem **_root_.GlobalTheory.ThickeningClosed.isOpen_zeroLocusOpen** {X : Scheme.{0}} (K : X.IdealSheafData) : IsOpen (_root_.GlobalTheory.ThickeningClosed.zeroLocusOpen K)

theorem **_root_.GlobalTheory.ThickeningClosed.exists_mul_eq_zero_of_mem** {X : Scheme.{0}} (K : X.IdealSheafData) (U : X.affineOpens) (q : PrimeSpectrum Γ(X, U)) (hq : U.2.fromSpec q ∈ _root_.GlobalTheory.ThickeningClosed.zeroLocusOpen K) {a : Γ(X, U)} (ha : a ∈ K.ideal U) : ∃ s ∉ q.asIdeal, s * a = 0

theorem **_root_.GlobalTheory.ThickeningClosed.mem_of_thickenings** {R : Type} [CommRing R] (I : Ideal R) {X Z : Scheme.{0}} [IsLocallyNoetherian X] (f : X ⟶ Spec (CommRingCat.of R)) (ι : Z ⟶ X) (h : ∀ n : ℕ, ∃ l : pullback f (_root_.ReciprocityTheory.ChartRestriction.G.Out.thick R I n) ⟶ Z, l ≫ ι = pullback.fst f (_root_.ReciprocityTheory.ChartRestriction.G.Out.thick R I n)) (x : X) (hx : I ≤ (f x).asIdeal) : x ∈ _root_.GlobalTheory.ThickeningClosed.zeroLocusOpen ι.ker

theorem **_root_.GlobalTheory.ThickeningClosed.noZeroDivisors_fiber** {R C : Type} [CommRing R] [CommRing C] [Algebra R C] {X : Scheme.{0}} (f : X ⟶ Spec (CommRingCat.of R)) (j : Spec (CommRingCat.of C) ⟶ X) [IsOpenImmersion j] (hj : Spec.map (CommRingCat.ofHom (algebraMap R C)) = j ≫ f) (p : PrimeSpectrum R) [IsIntegral (f.fiber p)] : NoZeroDivisors (p.asIdeal.Fiber C)

theorem **_root_.GlobalTheory.isIso_of_isClosedImmersion_of_thickenings** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (I : Ideal R), I ≤ Ideal.jacobson ⊥ → ∀ (H : Over (Spec (CommRingCat.of R))) [Flat H.hom] [LocallyOfFiniteType H.hom] (_e : 𝟙_ _ ⟶ H), (∀ p, IsIntegral (H.hom.fiber p)) → ∀ (Z : Scheme.{0}) (ι : Z ⟶ H.left) [IsClosedImmersion ι], (∀ n : ℕ, ∃ l : (_root_.ReciprocityTheory.ChartRestriction.G.Out.thickObj R I n H).left ⟶ Z, l ≫ ι = (_root_.ReciprocityTheory.ChartRestriction.G.Out.fstG I n H).left) → IsIso ι

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsLocalizedModuleResBasicOpenOfQcqs

@[simp] theorem **resBasicOpen_apply** (a : Γ(X, V)) (s : SecOn M V) : resBasicOpen M a s = restrict M (X.basicOpen_le a) s

theorem **smul_basicOpen_def** (a b : Γ(X, V)) (s : SecOn M (X.basicOpen a)) : b • s = rs X (X.basicOpen_le a) b • s

theorem **isLocalizedModule_resBasicOpen_of_qcqs** (hM : M.IsLB) (hV : IsCompact (V : Set X)) (hV' : IsQuasiSeparated (V : Set X)) (a : Γ(X, V)) : IsLocalizedModule.Away a (resBasicOpen M a)

theorem **isLocalizedModule_resBasicOpen** (hM : M.IsLB) (hV : IsAffineOpen V) (a : Γ(X, V)) : IsLocalizedModule.Away a (resBasicOpen M a)

theorem **exists_pow_smul_eq_zero** (hM : M.IsLB) (hV : IsCompact (V : Set X)) (a : Γ(X, V)) (h : W ≤ V) (hW : W = X.basicOpen a) (t : SecOn M V) (ht : restrict M h t = 0) : ∃ n : ℕ, a ^ n • t = 0

theorem **exists_pow_smul_eq_restrict** (hM : M.IsLB) (hV : IsCompact (V : Set X)) (hV' : IsQuasiSeparated (V : Set X)) (a : Γ(X, V)) (h : W ≤ V) (hW : W = X.basicOpen a) (s : SecOn M W) : ∃ (n : ℕ) (t : SecOn M V), rs X h a ^ n • s = restrict M h t

theorem **isLocalizedModule_of_ringHom** (hM : M.IsLB) (hV : IsCompact (V : Set X)) (hV' : IsQuasiSeparated (V : Set X)) {A : Type*} [CommRing A] (φ : A →+* Γ(X, V)) (a : A) (h : W ≤ V) (hW : W = X.basicOpen (φ a)) [Module A (SecOn M V)] [Module A (SecOn M W)] (hsV : ∀ (r : A) (s : SecOn M V), r • s = φ r • s) (hsW : ∀ (r : A) (s : SecOn M W), r • s = rs X h (φ r) • s) (l : SecOn M V →ₗ[A] SecOn M W) (hl : ∀ s, l s = restrict M h s) : IsLocalizedModule.Away a l

theorem **isBaseChange_restrict** (hM : M.IsLB) (hV : IsAffineOpen V) (hW : IsAffineOpen W) (h : W ≤ V) [Algebra Γ(X, V) Γ(X, W)] [Module Γ(X, V) (SecOn M W)] [IsScalarTower Γ(X, V) Γ(X, W) (SecOn M W)] (halg : ∀ r, algebraMap Γ(X, V) Γ(X, W) r = rs X h r) (l : SecOn M V →ₗ[Γ(X, V)] SecOn M W) (hl : ∀ s, l s = restrict M h s) : IsBaseChange Γ(X, W) l

theorem **invertible** (hM : M.IsLB) (hV : IsAffineOpen V) : Module.Invertible Γ(X, V) (SecOn M V)

theorem **finite** (hM : M.IsLB) (hV : IsAffineOpen V) : Module.Finite Γ(X, V) (SecOn M V)

theorem **projective** (hM : M.IsLB) (hV : IsAffineOpen V) : Module.Projective Γ(X, V) (SecOn M V)

theorem **finitePresentation** (hM : M.IsLB) (hV : IsAffineOpen V) : Module.FinitePresentation Γ(X, V) (SecOn M V)

theorem **flat** (hM : M.IsLB) (hV : IsAffineOpen V) : Module.Flat Γ(X, V) (SecOn M V)

theorem **isBaseChange_pullLE** (hM : M.IsLB) (π : X' ⟶ X) {V' : X'.Opens} (hV : IsAffineOpen V) (hV' : IsAffineOpen V') (e : V' ≤ π ⁻¹ᵁ V) [Algebra Γ(X, V) Γ(X', V')] [Module Γ(X, V) (SecOn (M.pull π) V')] [IsScalarTower Γ(X, V) Γ(X', V') (SecOn (M.pull π) V')] (halg : ∀ r, algebraMap Γ(X, V) Γ(X', V') r = (π.appLE V V' e).hom r) (l : SecOn M V →ₗ[Γ(X, V)] SecOn (M.pull π) V') (hl : ∀ s, l s = pullLE M π V V' e s) : IsBaseChange Γ(X', V') l

theorem **cst_eq_appLE** (f : X ⟶ Spec (CommRingCat.of R)) (V : X.Opens) : CommRingCat.ofHom (cst f V) = (Scheme.ΓSpecIso (CommRingCat.of R)).inv ≫ f.appLE ⊤ V le_top

theorem **flat_cst** (f : X ⟶ Spec (CommRingCat.of R)) [Flat f] {V : X.Opens} (hV : IsAffineOpen V) : (cst f V).Flat

theorem **isAffineOpen_preimage_of_isPullback** (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) {V : X.Opens} (hV : IsAffineOpen V) : IsAffineOpen (π ⁻¹ᵁ V)

theorem **isPushout_cst** (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) {V : X.Opens} (hV : IsAffineOpen V) {V' : X'.Opens} (e : V' = π ⁻¹ᵁ V) : IsPushout (CommRingCat.ofHom g) (CommRingCat.ofHom (cst f V)) (CommRingCat.ofHom (cst f' V')) (π.appLE V V' e.le)

theorem **isLocalizedModule_restrict** (hM : M.IsLB) (hV : IsCompact (V : Set X)) (hV' : IsQuasiSeparated (V : Set X)) (r : R) (h : W ≤ V) (hW : W = X.basicOpen (cst f V r)) : IsLocalizedModule.Away r (restrict f M h)

theorem **invertible** (hM : M.IsLB) (hV : IsAffineOpen V) : Module.Invertible Γ(X, V) (SecOnR f M V)

theorem **flat_of_flat_cst** (hM : M.IsLB) (hV : IsAffineOpen V) (hf : (cst f V).Flat) : Module.Flat R (SecOnR f M V)

theorem **flat** [Flat f] (hM : M.IsLB) (hV : IsAffineOpen V) : Module.Flat R (SecOnR f M V)

theorem **isBaseChange_pull** (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hV : IsAffineOpen V) {V' : X'.Opens} (e : V' = π ⁻¹ᵁ V) [Module R (SecOnR f' (M.pull π) V')] [IsScalarTower R R' (SecOnR f' (M.pull π) V')] (l : SecOnR f M V →ₗ[R] SecOnR f' (M.pull π) V') (hl : ∀ s, l s = pull M H.w V V' e.le s) : IsBaseChange R' l

@[simp] theorem **baseChangeHom_tmul** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) {V' : X'.Opens} (e : V' ≤ π ⁻¹ᵁ V) (c : R') (s : SecOnR f M V) : baseChangeHom M w V V' e (c ⊗ₜ s) = c • pull M w V V' e s

theorem **baseChangeHom_restrict** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) {V' W' : X'.Opens} (h : W ≤ V) (h' : W' ≤ V') (e : V' ≤ π ⁻¹ᵁ V) (e' : W' ≤ π ⁻¹ᵁ W) (x : R' ⊗[R] SecOnR f M V) : baseChangeHom M w W W' e' ((restrict f M h).baseChange R' x) = restrict f' _ h' (baseChangeHom M w V V' e x)

theorem **baseChangeHom_bijective** (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hV : IsAffineOpen V) {V' : X'.Opens} (e : V' = π ⁻¹ᵁ V) : Function.Bijective (baseChangeHom M H.w V V' e.le)

@[simp] theorem **baseChangeEquiv_tmul** (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hV : IsAffineOpen V) {V' : X'.Opens} (e : V' = π ⁻¹ᵁ V) (c : R') (s : SecOnR f M V) : baseChangeEquiv hM H hV e (c ⊗ₜ s) = c • pull M H.w V V' e.le s

theorem **flat_cechLB_X** [LinearOrder κ] [Finite κ] (f : X ⟶ Spec (CommRingCat.of R)) [Flat f] (hM : M.IsLB) (hD : ∀ s : Finset κ, s.Nonempty → IsAffineOpen (⊓ i ∈ s, D i)) (p : ℕ) : Module.Flat R ((cechLB f M D).X p)

theorem **secCubeBaseChange_app_tmul** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) (s : Finset κ) (c : R') (x : (secCube f M D).obj s) : ((secCubeBaseChange M D w).app s).hom (c ⊗ₜ[R] x) = c • secCubePull M D w s x

theorem **secCubeBaseChange_app_bijective** (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) {s : Finset κ} (hs : IsAffineOpen (⊓ i ∈ s, D i)) : Function.Bijective ((secCubeBaseChange M D H.w).app s).hom

theorem **secTermBaseChange_tmul** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) (p : ℕ) (c : R') (x : GlobalTheory.Cube.Term (secCube f M D) p) (s : {s : Finset κ // s.card = p}) : secTermBaseChange M D w p (c ⊗ₜ[R] x) s = c • secCubePull M D w s.1 (x s)

theorem **d_secTermBaseChange** [LinearOrder κ] (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) (p : ℕ) (x : R' ⊗[R] GlobalTheory.Cube.Term (secCube f M D) p) : GlobalTheory.Cube.d _ p (secTermBaseChange M D w p x) = secTermBaseChange M D w (p + 1) ((GlobalTheory.Cube.d (secCube f M D) p).baseChange R' x)

theorem **secTermBaseChange_bijective** [Finite κ] (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (p : ℕ) (hD : ∀ s : Finset κ, s.card = p → IsAffineOpen (⊓ i ∈ s, D i)) : Function.Bijective (secTermBaseChange M D H.w p)

theorem **cechLBBaseChange_f_tmul** [LinearOrder κ] (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom g)) (p : ℕ) (c : R') (x : (cechLB f M D).X p) (s : {s : Finset κ // s.card = p + 1}) : ((cechLBBaseChange M D w).f p).hom (c ⊗ₜ[R,g] x) s = c • secCubePull M D w s.1 (x s)

theorem **isIso_cechLBBaseChange** [LinearOrder κ] [Finite κ] (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) (hD : ∀ s : Finset κ, s.Nonempty → IsAffineOpen (⊓ i ∈ s, D i)) : IsIso (cechLBBaseChange M D H.w)

@[simp] theorem **cechLBBaseChangeIso_hom** [LinearOrder κ] [Finite κ] (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) (hD : ∀ s : Finset κ, s.Nonempty → IsAffineOpen (⊓ i ∈ s, D i)) : (cechLBBaseChangeIso hM H hD).hom = cechLBBaseChange M D H.w

theorem **isAffineOpen_biInf_preimage** (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) {s : Finset κ} (hs : IsAffineOpen (⊓ i ∈ s, D i)) : IsAffineOpen (⊓ i ∈ s, π ⁻¹ᵁ D i)

theorem **sections_over_affine_open** {X : Scheme.{0}} {M : LB X} (hM : M.IsLB) : (∀ V : X.Opens, IsCompact (V : Set X) → IsQuasiSeparated (V : Set X) → ∀ a : Γ(X, V), IsLocalizedModule.Away a (SecOn.resBasicOpen M a)) ∧ (∀ V : X.Opens, IsAffineOpen V → ∀ a : Γ(X, V), IsLocalizedModule.Away a (SecOn.resBasicOpen M a)) ∧ (∀ V : X.Opens, IsAffineOpen V → Module.Invertible Γ(X, V) (SecOn M V) ∧ Module.FinitePresentation Γ(X, V) (SecOn M V) ∧ Module.Projective Γ(X, V) (SecOn M V)) ∧ (∀ (R : Type) [CommRing R] (f : X ⟶ Spec (CommRingCat.of R)), Flat f → ∀ V : X.Opens, IsAffineOpen V → Module.Flat R (SecOnR f M V)) ∧ (∀ (R R' : Type) [CommRing R] [CommRing R'] [Algebra R R'] (X' : Scheme.{0}) (π : X' ⟶ X) (f : X ⟶ Spec (CommRingCat.of R)) (f' : X' ⟶ Spec (CommRingCat.of R')) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (V : X.Opens), IsAffineOpen V → Function.Bijective

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsPrincipalMapAwayOfHasProjectiveDimensionLE

theorem **nonempty_equiv_prod_of_exact_of_projective** {A : Type*} [CommRing A] {K F M : Type*} [AddCommGroup K] [Module A K] [AddCommGroup F] [Module A F] [AddCommGroup M] [Module A M] [Module.Projective A M] (g : K →ₗ[A] F) (f : F →ₗ[A] M) (hg : Function.Injective g) (hf : Function.Surjective f) (h : Function.Exact g f) : Nonempty (F ≃ₗ[A] K × M)

theorem **isStablyFree_baseChange_of_hasProjectiveDimensionLE** (R : Type) [CommRing R] [IsNoetherianRing R] [IsLocalRing R] (A : Type) [CommRing A] [Algebra R A] [Module.Flat R A] (n : ℕ) : ∀ (M : Type) [AddCommGroup M] [Module R M] [Module.Finite R M], HasProjectiveDimensionLE (ModuleCat.of R M) n → Module.Projective A (A ⊗[R] M) → Module.IsStablyFree A (A ⊗[R] M)

theorem **exists_surjective_exteriorPower_prod** (bN : Module.Basis (Fin n) A N) : ∃ φ : ∧[A]^(n + 1) (M × N) →ₗ[A] M, Function.Surjective φ

theorem **isPrincipal_of_isStablyFree** {A : Type} [CommRing A] (J : Ideal A) [Module.Finite A J] [Module.IsStablyFree A J] : J.IsPrincipal

theorem **isPrincipal_map_away_of_hasProjectiveDimensionLE** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] [IsLocalRing R] [IsDomain R] (x : R) (R' : Type) [CommRing R'] [Algebra R R'] [IsLocalization.Away x R'] (I : Ideal R) (n : ℕ), HasProjectiveDimensionLE (ModuleCat.of R I) n → Module.Projective R' (I.map (algebraMap R R')) → (I.map (algebraMap R R')).IsPrincipal

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsProjByOfIsPullback

theorem **rs_ap** (p : P ⟶ X) {U : X.Opens} {W W' : P.Opens} (h : W ≤ p ⁻¹ᵁ U) (h' : W' ≤ W) (x : Γ(X, U)) : rs P h' (ap p h x) = ap p (h'.trans h) x

theorem **ap_rs** (p : P ⟶ X) {U U' : X.Opens} {W : P.Opens} (h : W ≤ p ⁻¹ᵁ U') (h' : U' ≤ U) (x : Γ(X, U)) : ap p h (rs X h' x) = ap p (h.trans fun _ hy => h' hy) x

theorem **basicOpen_ap** (p : P ⟶ X) {U : X.Opens} {W : P.Opens} (h : W ≤ p ⁻¹ᵁ U) (x : Γ(X, U)) : P.basicOpen (ap p h x) = W ⊓ p ⁻¹ᵁ X.basicOpen x

theorem **ap_cst** (p : P ⟶ X) (f : X ⟶ Spec (CommRingCat.of R)) {U : X.Opens} {W : P.Opens} (h : W ≤ p ⁻¹ᵁ U) (r : R) : ap p h (cst f U r) = cst (p ≫ f) W r

theorem **val_res** {U V : X.Opens} (h : V ≤ U) (u : Γ(X, U)ˣ) : (res h u).val = rs X h u.val

theorem **le_pre** (p : P ⟶ X) {U V : X.Opens} {W : P.Opens} (h : W ≤ p ⁻¹ᵁ U) (h' : W ≤ p ⁻¹ᵁ V) : W ≤ p ⁻¹ᵁ (U ⊓ V)

theorem **pre_left** (p : P ⟶ X) {U V : X.Opens} {W : P.Opens} (h : W ≤ p ⁻¹ᵁ (U ⊓ V)) : W ≤ p ⁻¹ᵁ U

theorem **pre_right** (p : P ⟶ X) {U V : X.Opens} {W : P.Opens} (h : W ≤ p ⁻¹ᵁ (U ⊓ V)) : W ≤ p ⁻¹ᵁ V

theorem **rs_mul** {U : X.Opens} {V : Y.Opens} {W W' : P.Opens} (h₁ : W ≤ p ⁻¹ᵁ U) (h₂ : W ≤ q ⁻¹ᵁ V) (h : W' ≤ W) (x : Γ(X, U)) (y : Γ(Y, V)) : rs P h (ap p h₁ x * ap q h₂ y) = ap p (h.trans h₁) x * ap q (h.trans h₂) y

theorem **ap_g_self** (hH : H.IsLB) (i : H.ι) {W : P.Opens} (h : W ≤ p ⁻¹ᵁ (H.U i ⊓ H.U i)) : ap p h (H.g i i).val = 1

theorem **ap_cocycle** (hH : H.IsLB) (i j k : H.ι) {W : P.Opens} (h₁ : W ≤ p ⁻¹ᵁ (H.U i ⊓ H.U j)) (h₂ : W ≤ p ⁻¹ᵁ (H.U j ⊓ H.U k)) (h₃ : W ≤ p ⁻¹ᵁ (H.U i ⊓ H.U k)) : ap p h₃ (H.g i k).val = ap p h₁ (H.g i j).val * ap p h₂ (H.g j k).val

theorem **ap_ratio** {U V : X.Opens} {x y : Γ(X, V)} {r : Γ(X, U)} (e : rs X (inf_le_right : U ⊓ V ≤ V) x = rs X (inf_le_left : U ⊓ V ≤ U) r * rs X (inf_le_right : U ⊓ V ≤ V) y) {W : P.Opens} (h₁ : W ≤ p ⁻¹ᵁ U) (h₂ : W ≤ p ⁻¹ᵁ V) : ap p h₂ x = ap p h₁ r * ap p h₂ y

theorem **ap_compat** (s : LB.Sec H) (i j : H.ι) {W : P.Opens} (h₁ : W ≤ p ⁻¹ᵁ H.U i) (h₂ : W ≤ p ⁻¹ᵁ H.U j) (h : W ≤ p ⁻¹ᵁ (H.U i ⊓ H.U j)) : ap p h₁ (s.f i) = ap p h (H.g i j).val * ap p h₂ (s.f j)

theorem **isLB_box** (hH : H.IsLB) (hK : K.IsLB) : (box p q H K).IsLB where

theorem **nonzeroLocus_boxSec** (s : LB.Sec H) (t : LB.Sec K) : nonzeroLocus (boxSec p q s t) = p ⁻¹ᵁ nonzeroLocus s ⊓ q ⁻¹ᵁ nonzeroLocus t

theorem **ratio_self** (s : LB.Sec H) (r : Γ(X, nonzeroLocus s)) (h : ∀ k, rs X (inf_le_right : nonzeroLocus s ⊓ H.U k ≤ H.U k) (s.f k) = rs X (inf_le_left : nonzeroLocus s ⊓ H.U k ≤ nonzeroLocus s) r * rs X (inf_le_right : nonzeroLocus s ⊓ H.U k ≤ H.U k) (s.f k)) : r = 1

theorem **isProjBy_of_isPullback** {f : X ⟶ Spec (CommRingCat.of R)} {g : Y ⟶ Spec (CommRingCat.of R)} (sq : IsPullback p q f g) (hf : IsProjBy f) (hg : IsProjBy g) : IsProjBy (p ≫ f)

theorem **GlobalTheory.AbLaw.isProjBy_tensor** : ∀ (R : Type) [CommRing R] (X Y : Over (Spec (CommRingCat.of R))), IsProjBy X.hom → IsProjBy Y.hom → IsProjBy (X ⊗ Y).hom

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsRegularRingOfIsDiscreteValuationRing

theorem **isRegularRing_of_isLocalization** {B : Type} [CommRing B] [IsRegularRing B] (M : Submonoid B) (R : Type) [CommRing R] [Algebra B R] [IsLocalization M R] : IsRegularRing R

theorem **isRegularRing_of_isDiscreteValuationRing** (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] : IsRegularRing V

theorem **isLocalization_away_of_irreducible** (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] {π : V} (hπ : Irreducible π) : IsLocalization.Away π K

theorem **isOpenImmersion_etaK** (V : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (K : Type) [Field K] [Algebra V K] [IsFractionRing V K] : IsOpenImmersion (etaK V K)

theorem **isRegularRing_stalk** (hreg : ∀ (V B : Type) [CommRing V] [IsRegularRing V] [CommRing B] [Algebra V B] [Algebra.Smooth V B], IsRegularRing B) (V : Type) [CommRing V] [IsRegularRing V] {X : Scheme.{0}} (f : X ⟶ Spec (CommRingCat.of V)) [Smooth f] (x : X) : IsRegularRing (X.presheaf.stalk x)

theorem **extend_exists** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (V B : Type) [CommRing V] [IsRegularRing V] [CommRing B] [Algebra V B] [Algebra.Smooth V B], IsRegularRing B) → (∀ (R : Type) [CommRing R]

[IsRegularRing R] [IsLocalRing R] [IsDomain R], UniqueFactorizationMonoid R) → (∀ (X Y : Scheme.{0}) [IsIntegral X] [IsLocallyNoetherian X] [CompactSpace X] (j : Y ⟶ X), IsOpenImmersion j → (∀ x : X, UniqueFactorizationMonoid (X.presheaf.stalk x)) → ∀ L : LB Y, L.IsLB → ∃ L' : LB X, L'.IsLB ∧ ∃ φ : LB.Iso (L'.pull j) L, φ.IsGen) → Semi.extend_existsSt.{0}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsRegularRingOfSmooth

private theorem **isRegularLocalRing_atPrime_of_etale'** (R S : Type*) [CommRing R] [CommRing S] [Algebra R S] [IsNoetherianRing R] [IsNoetherianRing S] [Algebra.Etale R S] (q : Ideal S) [q.IsPrime] (h : IsRegularLocalRing (Localization.AtPrime (q.under R))) : IsRegularLocalRing (Localization.AtPrime q)

theorem **isRegularRing_of_etale** (R S : Type*) [CommRing R] [IsRegularRing R] [CommRing S] [Algebra R S] [Algebra.Etale R S] : IsRegularRing S

theorem **isRegularRing_of_smooth** : ∀ (V B : Type) [CommRing V] [IsRegularRing V] [CommRing B] [Algebra V B] [Algebra.Smooth V B], IsRegularRing B

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.IsUnitIffMem

theorem **le_pre** {Y Z : Scheme.{0}} (D : Y.Opens) (g : Z ⟶ D.toScheme) : ⊤ ≤ (g ≫ D.ι) ⁻¹ᵁ D

theorem **appLE_congr'** {Y Z : Scheme.{0}} {g g' : Z ⟶ Y} (e : g = g') {U : Y.Opens} {V : Z.Opens} (h : V ≤ g ⁻¹ᵁ U) : g.appLE U V h = g'.appLE U V (e ▸ h)

theorem **appTop_chart** {Y Z : Scheme.{0}} (D : Y.Opens) {B : CommRingCat.{0}} (φ : B ⟶ Γ(Y, D)) (g : Z ⟶ D.toScheme) : (g ≫ D.toSpecΓ ≫ Spec.map φ).appTop = (Scheme.ΓSpecIso B).hom ≫ φ ≫ (g ≫ D.ι).appLE D ⊤ (le_pre D g)

theorem **affine_surj** {Y W : Scheme.{0}} (f : Y ⟶ W) [IsAffine W] [IsProper f] [IsLocallyNoetherian W] (D : Y.Opens) {σ : Type} (r : σ → Γ(Y, D)) (hD : D ≤ f ⁻¹ᵁ ⊤) (hval : ∀ (V K : Type) [CommRing V] [IsDomain V] [ValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] (l : Spec (.of V) ⟶ Y) (h : ⊤ ≤ (Spec.map (CommRingCat.ofHom (algebraMap V K)) ≫ l) ⁻¹ᵁ D), (∀ j, ∃ ρ : Γ(Spec (.of V), ⊤), (Spec.map (CommRingCat.ofHom (algebraMap V K))).appTop.hom ρ = ((Spec.map (CommRingCat.ofHom (algebraMap V K)) ≫ l).appLE D ⊤ h).hom (r j)) → ⊤ ≤ l ⁻¹ᵁ D) (hmono : ∀ (Z : Scheme.{0}) [IsLocallyNoetherian Z] (x y : Z ⟶ Y) (hx : ⊤ ≤ x ⁻¹ᵁ D) (hy : ⊤ ≤ y ⁻¹ᵁ D), x ≫ f = y ≫ f → (∀ j, (x.appLE D ⊤ hx).hom (r j) = (y.appLE D ⊤ hy).hom (r j)) → x = y) : IsAffineOpen D ∧ Function.Surjective (MvPolynomial.eval₂Hom (f.appLE ⊤ D hD).hom r)

theorem **isUnit_gs** {Z Y : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (h : r z ∈ U) (a : Γ(Y, U)) : IsUnit (gs r z U a) ↔ r z ∈ Y.basicOpen a

theorem **gs_appLE** {Z' Z Y : Scheme.{0}} (q : Z' ⟶ Z) (r : Z ⟶ Y) (z : Z') (U : Y.Opens) (V : Z.Opens) (hle : V ≤ r ⁻¹ᵁ U) (h : q z ∈ V) (a : Γ(Y, U)) : gs q z V ((r.appLE U V hle).hom a) = gs (q ≫ r) z U a

theorem **gs_appTop** {Z' Z Y : Scheme.{0}} (q : Z' ⟶ Z) (r : Z ⟶ Y) (z : Z') (a : Γ(Y, ⊤)) : gs q z ⊤ (r.appTop.hom a) = gs (q ≫ r) z ⊤ a

theorem **gs_sum** {Z Y : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (h : r z ∈ U) {I : Type} [Fintype I] (a : I → Γ(Y, U)) : gs r z U (∑ i, a i) = ∑ i, gs r z U (a i)

theorem **sec_trans'** {Z Y : Scheme.{0}} {M : LB Y} (s : LB.Sec M) (r : Z ⟶ Y) (z : Z) {j j' : M.ι} (hj : r z ∈ M.U j) (hj' : r z ∈ M.U j') : gs r z (M.U j) (s.f j) = (gm r z _ (M.g j j')).val ∗ gs r z (M.U j') (s.f j')

theorem **isUnit_iff_mem** {Z Y : Scheme.{0}} {M : LB Y} (s : LB.Sec M) (r : Z ⟶ Y) (z : Z) {j : M.ι} (hj : r z ∈ M.U j) : IsUnit (gs r z (M.U j) (s.f j)) ↔ r z ∈ nonzeroLocus s

theorem **exists_ratio** {Y : Scheme.{0}} {M : LB Y} (s t : LB.Sec M) : ∃ ρ : Γ(Y, nonzeroLocus s), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (k : M.ι), r z ∈ nonzeroLocus s → r z ∈ M.U k → gs r z _ (t.f k) = gs r z _ ρ ∗ gs r z _ (s.f k)

theorem **valuative** {Y : Scheme.{0}} {M : LB Y} {I : Type} (s : I → LB.Sec M) (hcov : ⊔ i, nonzeroLocus (s i) = ⊤) (i : I) (ρ : I → Γ(Y, nonzeroLocus (s i))) (hρ : ∀ j (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (k : M.ι), r z ∈ nonzeroLocus (s i) → r z ∈ M.U k → gs r z _ ((s j).f k) = gs r z _ (ρ j) ∗ gs r z _ ((s i).f k)) (V K : Type) [CommRing V] [IsDomain V] [ValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] (l : Spec (.of V) ⟶ Y) (h : ⊤ ≤ (Spec.map (CommRingCat.ofHom (algebraMap V K)) ≫ l) ⁻¹ᵁ nonzeroLocus (s i)) (H : ∀ j, ∃ a : Γ(Spec (.of V), ⊤), (Spec.map (CommRingCat.ofHom (algebraMap V K))).appTop.hom a = ((Spec.map (CommRingCat.ofHom (algebraMap V K)) ≫ l).appLE _ ⊤ h).hom (ρ j)) : ⊤ ≤ l ⁻¹ᵁ nonzeroLocus (s i)

theorem **exists_pull** {X Y Y' : Scheme.{0}} {L : LB X} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb L φ ε)) (h : Y' ⟶ Y) (φ' : K → (Y' ⟶ X)) (e : ∀ k, φ' k = h ≫ φ k) : ∃ s' : LB.Sec (LB.LinComb L φ' ε), (∀ j (l : LB.wchart L φ' j ≤ h ⁻¹ᵁ LB.wchart L φ j), s'.f j = (Y'.presheaf.map (homOfLE l).op).hom ((h.app _).hom (s.f j))) ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) J, Good L z J → sval s' r z J = sval s (r ≫ h) z J

theorem **isUnit_sval** {X Y Z : Scheme.{0}} {L : LB X} {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb L φ ε)) (r : Z ⟶ Y) (z : Z) {J} (hJ : Good L z J) : IsUnit (sval s r z J) ↔ r z ∈ nonzeroLocus s

theorem **formEq_pow** {X Y : Scheme.{0}} (a b : Y ⟶ X) (e : a = b) (n : ℕ) : FormEq (K

theorem **exists_frame** {X Y Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) (w : Y ⟶ X) (n : ℕ) (s : LB.Sec (LB.LinComb L (K

theorem **value_sval** {a : ℤ} {T : Over S} {s : LB.Sec (LPow A L a T)} {x : T ⟶ A} {τ : LB.Gen (LB.LinComb L (Y

theorem **span_sval** {a : ℤ} {T : Over S} {I : Type} [Fintype I] (b : I → LB.Sec (LPow A L a T)) (hb : Submodule.span Γ(T.left, ⊤) (Set.range fun i => (SecM.ofSec A (b i) : SecM A L a T)) = ⊤) (s : LB.Sec (LPow A L a T)) : ∃ c : I → Γ(T.left, ⊤), ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) J, Good L z J → sval (φ

theorem **key** {a : ℤ} {T : Over S} {I : Type} [Fintype I] (b : I → LB.Sec (LPow A L a T)) (T' : Over S) (h : T' ⟶ T) (hb : ∀ b' : I → LB.Sec (LPow A L a T'), (∀ i, IsPullSec A L a (h ▷ A) (whiskerRight_snd h A) (b i) (b' i)) → Submodule.span Γ(T'.left, ⊤) (Set.range fun i => (SecM.ofSec A (b' i) : SecM A L a T')) = ⊤) (s : LB.Sec (LPow A L a T')) : ∃ c : I → Γ(T'.left, ⊤), ∀ (x : T' ⟶ A) (g : T'.left ⟶ (T ⊗ A).left), g ≫ (fst T A).left = h.left → g ≫ (snd T A).left = x.left → ∀ (τ : LB.Gen (LB.LinComb L (Y

theorem **ample** : (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → ∀ U : S.Opens, IsAffineOpen U → ∃ (m : ℕ) (b : Fin (m + 1) → LB.Sec (LPow P.A P.L n (overRes (T

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.Laws3

theorem **laws3** : Laws3

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.Laws5

theorem **_root_.GlobalTheory.laws5** (hG : _root_.ReciprocityTheory.ChartRestriction.G.Out.GlobalFnsSt) (hS : _root_.ReciprocityTheory.ChartRestriction.G.Out.SecBaseChangeSt) : _root_.ReciprocityTheory.ChartRestriction.G.Out.Laws5

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LawsFieldsSeparately

theorem **_root_.GlobalTheory.lawsFieldsSeparately** : _root_.ReciprocityTheory.ChartRestriction.G.Out.GlobalSectionsRankAlgClosed ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.SecBaseChangeSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.AmpleSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.KerPowSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.MulPullSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.MulNSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.CubeSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.SquareSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.RigiditySt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.WeilSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.WeilNondegSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.SymLiftSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.IrredSt ∧ _root_.ReciprocityTheory.ChartRestriction.G.Out.GoodPtSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LawsOfH0EqOneClosedFields

theorem **laws_of_h0_eq_one** : (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K)))) = 1) → Laws ∧ Laws2

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LawsSixthOfUnifUniq

theorem **lawsSixth_of_unifUniq** : GlobalFnsSt → SecBaseChangeSt → AmpleSt → Laws6

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LawsThirdToSixthOfLaws

theorem **laws_third_to_sixth_of_laws** : Laws → DVR.TwistBaseChangeStatement → Laws3 ∧ Laws4 ∧ Laws5 ∧ Laws6

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LinearIndependentOfCommSmul

theorem **_root_.GlobalTheory.linearIndependent_of_comm_smul** {K G V : Type} [Field K] [AddCommGroup V] [Module K V] [Nontrivial V] (A : G → V →ₗ[K] V) (c : G → G → K) (hA : ∀ x, Function.Injective (A x)) (h : ∀ x y v, A x (A y v) = c x y • A y (A x v)) (hc : Function.Injective c) : LinearIndependent K A

theorem **_root_.GlobalTheory.heisenberg_family_irreducible** : ∀ (K G V : Type) [Field K] [CommGroup G] [Finite G] [AddCommGroup V] [Module K V] [Module.Finite K V] (U : G → V ≃ₗ[K] V) (e : G → G →* Kˣ), (∀ x y v, U x (U y v) = (e x y : K) • U y (U x v)) → Function.Injective e → Module.finrank K V ^ 2 ≤ Nat.card G → ∀ W : Submodule K V, (∀ x v, v ∈ W → U x v ∈ W) → W = ⊥ ∨ W = ⊤

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LocGenOfTrivial

theorem **gen_transfer** {M M' : LB Y} (e : M'.ι → M.ι) (hU : ∀ j, M'.U j ≤ M.U (e j)) (hg : ∀ i j (y : Y), y ∈ M'.U i → y ∈ M'.U j → gm (𝟙 Y) y (M'.U i ⊓ M'.U j) (M'.g i j) = gm (𝟙 Y) y (M.U (e i) ⊓ M.U (e j)) (M.g (e i) (e j))) (h : ∃ t : LB.Gen M, t.IsGen) : ∃ t' : LB.Gen M', t'.IsGen

theorem **pull_iff** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (h : Y' ⟶ Y) : (∃ c : LB.Gen ((LB.LinComb L φ ε).pull h), c.IsGen) ↔ HasGen L (fun k => h ≫ φ k) ε

theorem **HasGen.pull** {φ : K → (Y ⟶ X)} {ε : K → ℤ} (H : HasGen L φ ε) (h : Y' ⟶ Y) {φ' : K → (Y' ⟶ X)} (e : ∀ k, h ≫ φ k = φ' k) : HasGen L φ' ε

theorem **HasGen.comb** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {φ'' : K'' → (Y ⟶ X)} {ε'' : K'' → ℤ} (H : HasGen L φ ε) (H' : HasGen L φ' ε') (e : ∀ (U : Type) [CommGroup U] (h : (Y ⟶ X) → U), (∏ k, h (φ k) ^ ε k) * (∏ k, h (φ' k) ^ ε' k)⁻¹ = ∏ k, h (φ'' k) ^ ε'' k) : HasGen L φ'' ε''

theorem **hasGen_of_trivial** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} (e : ∀ (U : Type) [CommGroup U] (h : (Y ⟶ X) → U), ∏ k, h (φ k) ^ ε k = 1) : HasGen L φ ε

theorem **locGen_iff** (π : Y ⟶ B) (φ : K → (Y ⟶ X)) (ε : K → ℤ) : (∀ p : B, ∃ V : B.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb L φ ε).pull (π ⁻¹ᵁ V).ι), c.IsGen) ↔ LocGen L π φ ε

theorem **LocGen.comb** (hL : L.IsLB) {π : Y ⟶ B} {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {φ'' : K'' → (Y ⟶ X)} {ε'' : K'' → ℤ} (H : LocGen L π φ ε) (H' : LocGen L π φ' ε') (e : ∀ (U : Type) [CommGroup U] (h : (Y ⟶ X) → U), (∏ k, h (φ k) ^ ε k) * (∏ k, h (φ' k) ^ ε' k)⁻¹ = ∏ k, h (φ'' k) ^ ε'' k) : LocGen L π φ'' ε''

theorem **locGen_of_trivial** (hL : L.IsLB) {π : Y ⟶ B} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (e : ∀ (U : Type) [CommGroup U] (h : (Y ⟶ X) → U), ∏ k, h (φ k) ^ ε k = 1) : LocGen L π φ ε

theorem **HasGen.of_range** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {W : Y.Opens} (H : HasGen L (fun k => W.ι ≫ φ k) ε) (h : Y' ⟶ Y) (hW : ∀ y, h y ∈ W) {φ' : K → (Y' ⟶ X)} (e : ∀ k, h ≫ φ k = φ' k) : HasGen L φ' ε

theorem **LocGen.baseChange** {π : Y ⟶ B} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (H : LocGen L π φ ε) {π' : Y' ⟶ B'} (h : Y' ⟶ Y) (g : B' ⟶ B) (sq : h ≫ π = π' ≫ g) {φ' : K → (Y' ⟶ X)} (e : ∀ k, h ≫ φ k = φ' k) : LocGen L π' φ' ε

theorem **translPt_one** (T : Over S) : translPt A (1 : T ⟶ A) = snd T A

theorem **whisker_translPt** {T T' : Over S} (g : T' ⟶ T) (x : T ⟶ A) : g ▷ A ≫ translPt A x = translPt A (g ≫ x)

theorem **locGen_pow** {L : LB A.left} (hL : L.IsLB) {T : Over S} (sq : ∀ x y : T ⟶ A, LocGen L (K

theorem **eq_one_of_locGen** {L : LB A.left} (hK : KTriv A L) {T : Over S} (y : T ⟶ A) (H : LocGen L (K

theorem **locGen_theta_iff** {L : LB A.left} (hL : L.IsLB) (hK : KTriv A L) {T : Over S} (sq : ∀ x y : T ⟶ A, LocGen L (K

theorem **exists_theta** {L : LB A.left} {T : Over S} (x : T ⟶ A) (k : ℤ) (V : T.left.Opens) (H : HasGen L (fun i : Fin 2 => ((fst T A).left ⁻¹ᵁ V).ι ≫ ![(snd T A).left, (translPt A x).left] i) ![k, -k]) : ∃ θ : ThetaGrp A L k (overRes V), θ.pt = overResι V ≫ x

theorem **kerPow_of_square** : (∀ (S : Scheme.{0}) (P : PolAV S) (T : Over S) (x y : T ⟶ P.A) (p : T.left), ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LocallyOfFiniteTypeKer

theorem **rel_of_basis** {G U J : Type} [CommGroup G] [CommGroup U] [Fintype J] [DecidableEq J] (W : G → G → U → Prop) (φ : (J → ℤ) → G) (hφ : ∀ a b, φ (a + b) = φ a * φ b) (ex : ∀ a b, ∃! c, W (φ a) (φ b) c) (mul : ∀ x x' y c c', W x y c → W x' y c' → W (x * x') y (c * c')) (skew : ∀ x y c, W x y c → W y x c⁻¹) (β : (J → ℤ) → (J → ℤ) → ℤ) (β₁ : ∀ a a' b, β (a + a') b = β a b + β a' b) (β₂ : ∀ a b b', β a (b + b') = β a b + β a b') (ζ : U) (h : ∀ i j, W (φ (Pi.single i 1)) (φ (Pi.single j 1)) (ζ ^ β (Pi.single i 1) (Pi.single j 1))) (a b : J → ℤ) : W (φ a) (φ b) (ζ ^ β a b)

theorem **eq_section_of_range** {X T : Over S} (t : 𝟙_ (Over S) ⟶ X) [IsOpenImmersion t.left] (u : T ⟶ X) (h : Set.range u.left ⊆ Set.range t.left) : toUnit T ≫ t = u

theorem **exists_geomPoint** (X : Over S) (x : X.left) : ∃ (K : Type) (_ : Field K) (_ : IsAlgClosed K) (s : Spec (CommRingCat.of K) ⟶ S) (u : Over.mk s ⟶ X), Set.range u.left = {x}

theorem **isIso_desc_iff** {I : Type} {X : Over S} (t : I → (𝟙_ (Over S) ⟶ X)) : IsIso (Sigma.desc t) ↔ IsIso (Sigma.desc (f

theorem **isIso_desc_of_lists** {I : Type} {X : Over S} (t : I → (𝟙_ (Over S) ⟶ X)) [FormallyUnramified X.hom] [LocallyOfFiniteType X.hom] (h : ListsGeomPoints t) : IsIso (Sigma.desc t)

theorem **lists_of_isIso_desc** {I : Type} {X : Over S} (t : I → (𝟙_ (Over S) ⟶ X)) (h : IsIso (Sigma.desc t)) : ListsGeomPoints t ∧ FormallyUnramified X.hom

theorem **comp_mulBy** {A T : Over S} [GrpObj A] (x : T ⟶ A) (k : ℕ) : x ≫ mulBy A k = x ^ k

theorem **kerPt_fst** {A T : Over S} [GrpObj A] {k : ℕ} (x : T ⟶ A) (hx : x ^ k = 1) : kerPt x hx ≫ pullback.fst _ _ = x

theorem **ker_ext** {A T : Over S} [GrpObj A] {k : ℕ} {u v : T ⟶ Ker A k} (h : u ≫ pullback.fst _ _ = v ≫ pullback.fst _ _) : u = v

theorem **isKer_iff** {A : Over S} [GrpObj A] {k : ℕ} {I : Type} (x : I → (𝟙_ (Over S) ⟶ A)) (hx : ∀ i, x i ^ k = 1) : IsPullback (Sigma.desc x : constOver S I ⟶ A) (toUnit _) (mulBy A k) η[A] ↔ IsIso (Sigma.desc fun i => kerPt (x i) (hx i))

theorem **lists_kerPt_iff** {A : Over S} [GrpObj A] {k : ℕ} {I : Type} (x : I → (𝟙_ (Over S) ⟶ A)) (hx : ∀ i, x i ^ k = 1) : ListsGeomPoints (fun i => kerPt (x i) (hx i)) ↔ ∀ (K : Type) [Field K] [IsAlgClosed K] (s : Spec (CommRingCat.of K) ⟶ S) (y : Over.mk s ⟶ A), y ^ k = 1 → ∃! i, toUnit _ ≫ x i = y

theorem **locallyOfFiniteType_ker** (A : Over S) [GrpObj A] [LocallyOfFiniteType A.hom] (k : ℕ) : LocallyOfFiniteType (Ker A k).hom

theorem **formallyUnramified_ker** (A : Over S) [GrpObj A] [IsCommMonObj A] {k : ℕ} (hk : IsUnit (k : Γ(S, ⊤))) : FormallyUnramified (Ker A k).hom

theorem **weil_existsUnique** (P : PolAV S) (k : ℕ) {T : Over S} (x y : T ⟶ P.A) (hx : x ^ k = 1) (hy : y ^ k = 1) : ∃! c : Γ(T.left, ⊤)ˣ, WeilIs P.A P.L k x y c

theorem **weil_skew_natural** (P : PolAV S) (k : ℕ) (T : Over S) : (∀ (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k y x c⁻¹) ∧ ∀ (T' : Over S) (h : T' ⟶ T) (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom c)

theorem **halves_inl** (g : ℕ) (i : Fin g) : (halves g (.inl i) : ℕ) = i

theorem **halves_inr** (g : ℕ) (i : Fin g) : (halves g (.inr i) : ℕ) = g + i

theorem **stdForm_single** (g : ℕ) (i j : Fin g ⊕ Fin g) : stdForm (Pi.single i 1) (Pi.single j 1) = stdSymp g (halves g i) (halves g j)

theorem **stdForm_add_left** {g : ℕ} (a a' b : Fin g ⊕ Fin g → ℤ) : stdForm (a + a') b = stdForm a b + stdForm a' b

theorem **stdForm_add_right** {g : ℕ} (a b b' : Fin g ⊕ Fin g → ℤ) : stdForm a (b + b') = stdForm a b + stdForm a b'

theorem **reduce_add** {g : ℕ} (k : ℕ) (a b : Fin g ⊕ Fin g → ℤ) : reduce k (a + b) = reduce k a + reduce k b

theorem **reduce_single** {g : ℕ} (k : ℕ) (i : Fin g ⊕ Fin g) (n : ℤ) : reduce k (Pi.single i n) = Pi.single i (n : ZMod k)

theorem **basisHom_single** {G J : Type} [CommGroup G] [Fintype J] [DecidableEq J] {k : ℕ} [NeZero k] (y : J → G) (hy : ∀ i, y i ^ k = 1) (i : J) : basisHom y hy (Multiplicative.ofAdd (Pi.single i 1)) = y i

theorem **hom_pow** {G J : Type} [Monoid G] {k : ℕ} (φ : Multiplicative (J → ZMod k) →* G) (a : Multiplicative (J → ZMod k)) : φ a ^ k = 1

theorem **hom_eq_basisHom** {G J : Type} [CommGroup G] [Fintype J] [DecidableEq J] {k : ℕ} [NeZero k] (φ : Multiplicative (J → ZMod k) →* G) : basisHom (fun i => φ (Multiplicative.ofAdd (Pi.single i 1))) (fun _ => hom_pow φ _) = φ

theorem **comp_basisHom** {A T : Over S} [GrpObj A] [IsCommMonObj A] {g k : ℕ} [NeZero k] (y : Fin (2 * g) → (𝟙_ (Over S) ⟶ A)) (hy : ∀ i, y (halves g i) ^ k = 1) (h : T ⟶ 𝟙_ (Over S)) (a : Fin g ⊕ Fin g → ZMod k) : h ≫ basisHom (fun i => y (halves g i)) hy (Multiplicative.ofAdd a) = linComb (fun i => h ≫ y i) (fun i => (a ((halves g).symm i)).val)

theorem **indep_span_iff** {A T : Over S} [GrpObj A] [IsCommMonObj A] {g k : ℕ} [NeZero k] (y : Fin (2 * g) → (𝟙_ (Over S) ⟶ A)) (hy : ∀ i, y (halves g i) ^ k = 1) (h : T ⟶ 𝟙_ (Over S)) : ((∀ c c' : Fin (2 * g) → Fin k, linComb (fun i => h ≫ y i) (fun i => (c i : ℕ)) = linComb (fun i => h ≫ y i) (fun i => (c' i : ℕ)) → c = c') ∧ ∀ z : T ⟶ A, z ^ k = 1 → ∃ c : Fin (2 * g) → Fin k, linComb (fun i => h ≫ y i) (fun i => (c i : ℕ)) = z) ↔ ∀ z : T ⟶ A, z ^ k = 1 → ∃! a, toUnit T ≫ basisHom (fun i => y (halves g i)) hy (Multiplicative.ofAdd a) = z

theorem **torsion_halves** (lev : Out.Level P k ζ) (i : Fin P.g ⊕ Fin P.g) : (lev.pt (halves P.g i) : 𝟙_ (Over S) ⟶ P.A) ^ k = 1

theorem **homOfBasis_single** [NeZero k] (lev : Out.Level P k ζ) (i : Fin P.g ⊕ Fin P.g) : homOfBasis lev (Multiplicative.ofAdd (Pi.single i 1)) = lev.pt (halves P.g i)

theorem **pairing_of_basis** [NeZero k] (lev : Out.Level P k ζ) (a b : Fin P.g ⊕ Fin P.g → ℤ) : WeilIs P.A P.L (k : ℤ) (T

theorem **lists_of_iso** (ℓ : IndefiniteRowSideFullHecke.LevelStructureForms.Level P k ζ) (K : Type) [Field K] [IsAlgClosed K] (s : Spec (CommRingCat.of K) ⟶ S) (y : Over.mk s ⟶ P.A) (hy : y ^ k = 1) : ∃! a, toUnit _ ≫ ℓ.pt a = y

theorem **unramified_of_iso** (ℓ : IndefiniteRowSideFullHecke.LevelStructureForms.Level P k ζ) : FormallyUnramified (Ker P.A k).hom

theorem **indep_span_of_iso** [NeZero k] (ℓ : IndefiniteRowSideFullHecke.LevelStructureForms.Level P k ζ) (K : Type) [Field K] [IsAlgClosed K] (s : Spec (CommRingCat.of K) ⟶ S) (h : Over.mk s ⟶ 𝟙_ (Over S)) : (∀ c c' : Fin (2 * P.g) → Fin k, linComb (fun i => h ≫ ℓ.pt (Pi.single ((halves P.g).symm i) 1)) (fun i => (c i : ℕ)) = linComb (fun i => h ≫ ℓ.pt (Pi.single ((halves P.g).symm i) 1)) (fun i => (c' i : ℕ)) → c = c') ∧ ∀ z : Over.mk s ⟶ P.A, z ^ k = 1 → ∃ c : Fin (2 * P.g) → Fin k, linComb (fun i => h ≫ ℓ.pt (Pi.single ((halves P.g).symm i) 1)) (fun i => (c i : ℕ)) = z

theorem **basis_ext** {l l' : Out.Level P k ζ} (h : l.pt = l'.pt) : l = l'

theorem **iso_ext** {l l' : IndefiniteRowSideFullHecke.LevelStructureForms.Level P k ζ} (h : l.hom = l'.hom) : l = l'

theorem **neZero_of_prim** (hk : IsUnit (k : Γ(S, ⊤))) (hζ : IsPrimitiveRoot ζ k) : NeZero k

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.LocallyTrivialOfIsoPull

theorem **gen_restrict_iff** (M : LB Y) (W : Y.Opens) : (∃ c : LB.Gen (M.pull W.ι), c.IsGen) ↔ ∃ d, IsGenOn M W d

theorem **locally_trivial_of_iso_pull** (p : Y ⟶ T) (M : LB Y) {N : LB T} (hN : N.IsLB) (u : LB.Iso (N.pull p) M) (hu : u.IsGen) (t : T) : ∃ V : T.Opens, t ∈ V ∧ ∃ c : LB.Gen (M.pull (p ⁻¹ᵁ V).ι), c.IsGen

theorem **iso_pull_of_locally_trivial** (p : Y ⟶ T) (hp : ∀ V : T.Opens, Function.Bijective (p.app V).hom) {M : LB Y} (hM : M.IsLB) (h : ∀ t : T, ∃ V : T.Opens, t ∈ V ∧ ∃ c : LB.Gen (M.pull (p ⁻¹ᵁ V).ι), c.IsGen) : ∃ N : LB T, N.IsLB ∧ ∃ u : LB.Iso (N.pull p) M, u.IsGen

theorem **bijective_app_of_restrict** {Y T : Scheme.{0}} (p : Y ⟶ T) (V : T.Opens) (h : Function.Bijective (p ∣_ V).appTop.hom) : Function.Bijective (p.app V).hom

theorem **bijective_app_fst** (hG : GlobalFnsSt) {S : Scheme.{0}} {A : Over S} [GrpObj A] (hA : IsAbelianScheme A) (T : Over S) (V : T.left.Opens) : Function.Bijective ((fst T A).left.app V).hom

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.MemRangeOfSumEqZero

theorem **germPull_apply** {Z Y : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (U : Y.Opens) (h : r z ∈ U) (a : Γ(Y, U)) : germPull r z U h a = gs r z U a

theorem **gs_appLE** {Z Y X : Scheme.{0}} (f : Z ⟶ Y) (r : Y ⟶ X) (z : Z) (U : X.Opens) (V : Y.Opens) (hle : V ≤ r ⁻¹ᵁ U) (h : f z ∈ V) (a : Γ(X, U)) : gs f z V ((r.appLE U V hle).hom a) = gs (f ≫ r) z U a

theorem **exists_factor** {X W Y S : Scheme.{0}} [IsIntegral X] [Y.IsSeparated] [IsAffine S] (f : X ⟶ W) (w : W ⟶ Y) (g : X ⟶ S) (hS : IsField Γ(S, ⊤)) (D : W.Opens) (hD : IsAffineOpen D) (z : X) (hz : f z ∈ D) (h : ∀ a : Γ(W, D), ∃ c : Γ(S, ⊤), gs g z ⊤ c = gs f z D a) : ∃ p : S ⟶ Y, f ≫ w = g ≫ p

theorem **exists_monomials** {X W Z : Scheme.{0}} (L : LB X) (hL : L.IsLB) (w : W ⟶ X) (N : ℕ) (s s' : LB.Sec (LB.LinComb L (K

theorem **mem_range_of_sum_eq_zero** {K R : Type} [Field K] [IsAlgClosed K] [CommRing R] [IsDomain R] (κ : K →+* R) {n : ℕ} (c : Fin n → K) (hc : c ≠ 0) (x : R) (h : ∑ k, κ (c k) * x ^ (k : ℕ) = 0) : x ∈ κ.range

theorem **eq_one_of_forall_pow_dvd** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.MemRangeRTensorOfForallDual

theorem **mem_range_rTensor_of_forall_dual** {R C₀ C₁ B : Type} [CommRing R] (hR : IsField R) [AddCommGroup C₀] [Module R C₀] [AddCommGroup C₁] [Module R C₁] [AddCommGroup B] [Module R B] (δ : C₀ →ₗ[R] C₁) (z : C₁ ⊗[R] B) (h : ∀ l : C₁ →ₗ[R] R, l ∘ₗ δ = 0 → TensorProduct.lid R B (l.rTensor B z) = 0) : ∃ y, δ.rTensor B y = z

theorem **appLE_appLE** (f : T ⟶ X) (g : X ⟶ Y) (h : T ⟶ Y) (e : f ≫ g = h) {W : Y.Opens} {U : X.Opens} {N : T.Opens} (h1 : U ≤ g ⁻¹ᵁ W) (h2 : N ≤ f ⁻¹ᵁ U) (h3 : N ≤ h ⁻¹ᵁ W) (x : Γ(Y, W)) : (f.appLE U N h2).hom ((g.appLE W U h1).hom x) = (h.appLE W N h3).hom x

theorem **id_appLE** {U N : X.Opens} (h : N ≤ (𝟙 X : X ⟶ X) ⁻¹ᵁ U) (x : Γ(X, U)) : ((𝟙 X : X ⟶ X).appLE U N h).hom x = rs (show N ≤ U from h) x

theorem **mm_bijective** [IsAffine S] {U : X.Opens} {W : Y.Opens} (hU : IsAffineOpen U) (hW : IsAffineOpen W) : Function.Bijective (mm H U W (p ⁻¹ᵁ U ⊓ q ⁻¹ᵁ W) inf_le_left inf_le_right)

theorem **MM_tmul** (U : ι → X.Opens) (W : Y.Opens) (i : ι) (O : Z.Opens) (hU : O ≤ p ⁻¹ᵁ U i) (hW : O ≤ q ⁻¹ᵁ W) (a : ∀ i, Fn iX (U i)) (b : Fn iY W) : MM H U W i O hU hW (a ⊗ₜ b) = (p.appLE (U i) O hU).hom (a i) * (q.appLE W O hW).hom b

theorem **MM_eq** [Fintype ι] [DecidableEq ι] (U : ι → X.Opens) (W : Y.Opens) (i : ι) (O : Z.Opens) (hU : O ≤ p ⁻¹ᵁ U i) (hW : O ≤ q ⁻¹ᵁ W) (z) : MM H U W i O hU hW z = mm H (U i) W O hU hW (TensorProduct.piLeft _ _ _ z i)

theorem **MM_surj** [IsAffine S] [Finite ι] {U : ι → X.Opens} {W : Y.Opens} (hU : ∀ i, IsAffineOpen (U i)) (hW : IsAffineOpen W) (c : ∀ i, Γ(Z, p ⁻¹ᵁ U i ⊓ q ⁻¹ᵁ W)) : ∃ z, ∀ i, MM H U W i _ inf_le_left inf_le_right z = c i

theorem **MM_inj** [IsAffine S] [Finite ι] {U : ι → X.Opens} {W : Y.Opens} (hU : ∀ i, IsAffineOpen (U i)) (hW : IsAffineOpen W) (z) (h : ∀ i, MM H U W i _ inf_le_left inf_le_right z = 0) : z = 0

theorem **rs_MM** (U : ι → X.Opens) (W : Y.Opens) (i : ι) {O O' : Z.Opens} (h : O' ≤ O) (hU : O ≤ p ⁻¹ᵁ U i) (hW : O ≤ q ⁻¹ᵁ W) (z) : rs h (MM H U W i O hU hW z) = MM H U W i O' (h.trans hU) (h.trans hW) z

theorem **MM_lTensor** (U : ι → X.Opens) {W W' : Y.Opens} (h : W' ≤ W) (i : ι) (O : Z.Opens) (hU : O ≤ p ⁻¹ᵁ U i) (hW : O ≤ q ⁻¹ᵁ W') (z) : MM H U W' i O hU hW ((rsA iY h).toLinearMap.lTensor _ z) = MM H U W i O hU (hW.trans (q.preimage_mono h)) z

theorem **dlt_apply** (iX : X ⟶ S) (U : ι → X.Opens) (x : ∀ i, Fn iX (U i)) (k : ι × ι) : dlt iX U x k = (rs inf_le_left (x k.1) - rs inf_le_right (x k.2) : Γ(X, U k.1 ⊓ U k.2))

theorem **MM_dlt** (U : ι → X.Opens) (W : Y.Opens) (k : ι × ι) (O : Z.Opens) (hU : O ≤ p ⁻¹ᵁ (U k.1 ⊓ U k.2)) (hW : O ≤ q ⁻¹ᵁ W) (z) : MM H (fun k : ι × ι => U k.1 ⊓ U k.2) W k O hU hW ((dlt iX U).rTensor _ z) = MM H U W k.1 O (hU.trans (p.preimage_mono inf_le_left)) hW z - MM H U W k.2 O (hU.trans (p.preimage_mono inf_le_right)) hW z

theorem **section_MM** {y : S ⟶ Y} (hy : y ≫ iY = 𝟙 S) {s : X ⟶ Z} (hs : s ≫ p = 𝟙 X) (hs' : s ≫ q = iX ≫ y) (U : ι → X.Opens) (W : Y.Opens) (e : ⊤ ≤ y ⁻¹ᵁ W) (i : ι) (O : Z.Opens) (hU : O ≤ p ⁻¹ᵁ U i) (hW : O ≤ q ⁻¹ᵁ W) (hO : U i ≤ s ⁻¹ᵁ O) (z) : (s.appLE O (U i) hO).hom (MM H U W i O hU hW z) = TensorProduct.rid _ _ ((evA y hy W e).toLinearMap.lTensor _ z) i

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.MulAFiniteFlatSurjectiveEtale

theorem **isMonHom_mulA** {S : Scheme.{0}} (A : Over S) [GrpObj A] [IsCommMonObj A] (n : ℕ) : IsMonHom (mulA A n)

theorem **isPullback_baseChange** {S T : Scheme.{0}} (t : T ⟶ S) {X Y : Over S} (f : X ⟶ Y) : IsPullback (pullback.fst X.hom t) ((Over.pullback t).map f).left f.left (pullback.fst Y.hom t)

theorem **surjective_of_geomFibres** {S : Scheme.{0}} {X Y : Over S} (f : X ⟶ Y) (h : ∀ (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S), Surjective ((Over.pullback t).map f).left) : Surjective f.left

theorem **mulA_finite_flat_surjective_etale** : (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left) → (∀ (K : Type) [Field K] [IsAlgClosed K] (A : Over (Spec (CommRingCat.of K))) [GrpObj A], IsAbelianScheme A → ∀ f : A ⟶ A, IsMonHom f → IsFinite f.left → Flat f.left ∧ Surjective f.left) → (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (X Y : Over S) (f : X ⟶ Y), Flat X.hom → LocallyOfFiniteType Y.hom → IsFinite f.left → (∀ (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S), Flat ((Over.pullback t).map f).left) → Flat f.left) → (∀ (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (n : ℕ), IsUnit (n : Γ(S, ⊤)) → FormallyUnramified (mulA G n).left) → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.MulPullOfCubeOfSymm

theorem **HasGen.mul** (h : HasGen L φ ε) (h' : HasGen L φ' ε') : HasGen L (Sum.elim φ φ') (Sum.elim ε ε')

theorem **HasGen.inv** (h : HasGen L φ ε) : HasGen L φ (fun k => -ε k)

theorem **HasGen.pull** (h : HasGen L φ ε) (f : Z ⟶ Y) : HasGen L (fun k => f ≫ φ k) ε

theorem **HasGen.of_formEq** (hL : L.IsLB) (h : HasGen L φ ε) (e : FormEq φ ε φ' ε') : HasGen L φ' ε'

theorem **hasGen_pow** (hL : L.IsLB) (hcube : HasGen L (thφ A (x3 A)) thε) {T : Over S} (x : T ⟶ A) (hone : HasGen L (K

theorem **mulPull_of_cube_of_symm** : (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ nu : LB.Gen (negLB P.A P.L k T), nu.IsGen) → (∀ (S : Scheme.{0}) (P : PolAV

S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NatCardEqOfEquivPiField

theorem **natCard_eq_of_equiv_pi_field** {R : Type} [Field R] {U V : Type} [AddCommGroup U] [Module R U] [AddCommGroup V] [Module R V] {X : Type} (e : U ≃ₗ[R] (X → V)) {m : ℕ} (hm : 0 < m) (hU : Module.finrank R U = m) (hV : Module.finrank R V = 1) : Nat.card X = m

theorem **natCard_eq_of_equiv_pi** {R : Type} [CommRing R] (hR : IsField R) {U V : Type} [AddCommGroup U] [Module R U] [AddCommGroup V] [Module R V] {X : Type} (e : U ≃ₗ[R] (X → V)) {m : ℕ} (hm : 0 < m) (hU : Module.finrank R U = m) (hV : Module.finrank R V = 1) : Nat.card X = m

theorem **mulN_of_decomposition** : (∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ), 1 ≤ n → IsFinite (mulA P.A n).left ∧ Flat (mulA P.A n).left ∧ Surjective (mulA P.A n).left ∧ (IsUnit (n : Γ(S, ⊤)) → Etale (mulA P.A n).left)) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n j : ℕ), (n : K) ≠ 0 → 1 ≤ j → Nonempty (SecM P.A P.L ((n ^ 2 * j : ℕ) : ℤ) (baseOver (Spec (CommRingCat.of K))) ≃ₗ[Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤)] ({x : baseOver (Spec (CommRingCat.of K)) ⟶ P.A // x ^ n = 1} → SecM P.A P.L (j : ℤ) (baseOver (Spec (CommRingCat.of K)))))) → GlobalSectionsRankAlgClosed → PolBaseChangeSt → MulNSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NatCastNeZeroOfIsUnit

theorem **res_res** {X : Scheme.{0}} {U V W : X.Opens} (h : U ≤ V) (h' : V ≤ W) (u : Γ(X, W)ˣ) : res h (res h' u) = res (h.trans h') u

theorem **pullU_res** {X Y : Scheme.{0}} (f : Y ⟶ X) {U V : X.Opens} (h : U ≤ V) (u : Γ(X, V)ˣ) : pullU f U (res h u) = res (fun _ hy => h hy : f ⁻¹ᵁ U ≤ f ⁻¹ᵁ V) (pullU f V u)

theorem **DualRefines.pull** {Y Z : Scheme.{0}} {M' M : LB Y} (h : DualRefines M' M) (f : Z ⟶ Y) : DualRefines (M'.pull f) (M.pull f)

theorem **mulN_eq_mulA** (A : Over S) [GrpObj A] (n : ℕ) : mulN A n = mulA A n

theorem **finrank_mulN_eq_card** (A : Over S) [GrpObj A] [IsCommMonObj A] (n : ℕ) [IsFinite (mulN A n).left] [Etale (mulN A n).left] [Surjective (mulN A n).left] {k : Type} [Field k] [IsAlgClosed k] (t : Spec (CommRingCat.of k) ⟶ A.left) (z : Spec (CommRingCat.of k)) : (mulN A n).left.finrank (t z) = Nat.card {y : Over.mk (t ≫ A.hom) ⟶ A // y ^ n = 1}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NonemptyEquivPiOfWeights

theorem **nonempty_equiv_pi_of_weights** {R M N G : Type} [CommRing R] [IsDomain R] [AddCommGroup M] [Module R M] [AddCommGroup N] [Module R N] [CommGroup G] [Finite G] (n : ℕ) [NeZero n] [HasEnoughRootsOfUnity R n] (hn : IsUnit (n : R)) (hG : ∀ g : G, g ^ n = 1) (ρ : Representation R G M) (W : Submodule R M) (hW : ∀ g, ∀ v ∈ W, ρ g v ∈ W) (h : ∀ χ : G →* Rˣ, ∃ f : N →ₗ[R] M, Function.Injective f ∧ ∀ v, (∃ x, f x = v) ↔ v ∈ W ∧ ∀ g, ρ g v = (χ g : R) • v) : Nonempty (W ≃ₗ[R] (G → N))

theorem **algebraMap_vals** (c : Γ(B, ⊤)) (i : Idx L p) : algebraMap Γ(B, ⊤) (Vals L p) c i = gs p i.1.1 ⊤ c

theorem **mul_vals** (v w : Vals L p) (i : Idx L p) : (v * w) i = v i * w i

theorem **formEq_of_dv** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (h : dv φ ε = dv φ' ε') : FormEq φ ε φ' ε'

theorem **dv_support** (D : (Y ⟶ X) →₀ ℤ) : dv (fun k : D.support => k.1) (fun k => D k.1) = D

theorem **dv_comp** {Y' : Scheme.{0}} (h : Y' ⟶ Y) (φ : K → (Y ⟶ X)) (ε : K → ℤ) : dv (fun k => h ≫ φ k) ε = (dv φ ε).mapDomain (h ≫ ·)

theorem **ev_add** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {s t u : LB.Sec (LB.LinComb L φ ε)} (h : ∀ j, u.f j = s.f j + t.f j) : ev p u = ev p s + ev p t

theorem **ev_smul** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {s u : LB.Sec (LB.LinComb L φ ε)} (c : Γ(B, ⊤)) (h : ∀ j, u.f j = (Y.presheaf.map (homOfLE le_top).op).hom (p.appTop.hom c) * s.f j) : ev p u = c • ev p s

theorem **ev_zero** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {s : LB.Sec (LB.LinComb L φ ε)} (h : ∀ j, s.f j = 0) : ev p s = 0

theorem **ev_inj** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} {s t : LB.Sec (LB.LinComb L φ ε)} (h : ev p s = ev p t) (j : K → L.ι) : s.f j = t.f j

theorem **mem_Sg** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} {D : (Y ⟶ X) →₀ ℤ} (h : dv φ ε = D) (v : Vals L p) : v ∈ Sg L p D ↔ ∃ s : LB.Sec (LB.LinComb L φ ε), ev p s = v

theorem **mul_mem_Sg** (hL : L.IsLB) {D E F : (Y ⟶ X) →₀ ℤ} {v w : Vals L p} (hv : v ∈ Sg L p D) (hw : w ∈ Sg L p E) (h : D + E = F) : v * w ∈ Sg L p F

theorem **pl_pl** {h h' h'' : Y ⟶ Y} (hh : h ≫ p = p) (hh' : h' ≫ p = p) (hh'' : h'' ≫ p = p) (e : h ≫ h' = h'') (v : Vals L p) : pl L h hh (pl L h' hh' v) = pl L h'' hh'' v

theorem **pl_id** {h : Y ⟶ Y} (hh : h ≫ p = p) (e : h = 𝟙 Y) (v : Vals L p) : pl L h hh v = v

theorem **pl_mem_Sg** (hL : L.IsLB) {h : Y ⟶ Y} (hh : h ≫ p = p) {D E : (Y ⟶ X) →₀ ℤ} {v : Vals L p} (hv : v ∈ Sg L p D) (e : D.mapDomain (h ≫ ·) = E) : pl L h hh v ∈ Sg L p E

theorem **exists_isG** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) : ∃ u : (Vals L p)ˣ, IsG (dv φ ε) u ∧ ∀ i : Idx L p, u.val i = (val t i.1.2.2 i.1.1 i.1.2.1).val

theorem **IsG.inv** {D : (Y ⟶ X) →₀ ℤ} {u : (Vals L p)ˣ} (h : IsG D u) : IsG (-D) u⁻¹

theorem **IsG.mul** (hL : L.IsLB) {D E : (Y ⟶ X) →₀ ℤ} {u w : (Vals L p)ˣ} (h : IsG D u) (h' : IsG E w) : IsG (D + E) (u * w)

theorem **IsG.pow** (hL : L.IsLB) {D : (Y ⟶ X) →₀ ℤ} {u : (Vals L p)ˣ} (h : IsG D u) (j : ℕ) : IsG ((j + 1) • D) (u ^ (j + 1))

theorem **IsG.pull** (hL : L.IsLB) {h : Y ⟶ Y} (hh : h ≫ p = p) {D : (Y ⟶ X) →₀ ℤ} {u : (Vals L p)ˣ} (hu : IsG D u) : IsG (D.mapDomain (h ≫ ·)) (Units.map (pl L h hh : Vals L p →* Vals L p) u)

theorem **IsG.equiv** (hL : L.IsLB) {D E F : (Y ⟶ X) →₀ ℤ} {u : (Vals L p)ˣ} (hu : IsG D u) (h : E + D = F) : Nonempty (Sg L p E ≃ₗ[Γ(B, ⊤)] Sg L p F)

theorem **pl_equiv** (hL : L.IsLB) {h h' : Y ⟶ Y} (hh : h ≫ p = p) (hh' : h' ≫ p = p) (e : h ≫ h' = 𝟙 Y) (e' : h' ≫ h = 𝟙 Y) {D E : (Y ⟶ X) →₀ ℤ} (hE : D.mapDomain (h ≫ ·) = E) : Nonempty (Sg L p D ≃ₗ[Γ(B, ⊤)] Sg L p E)

theorem **gs_comp** {Z W : Scheme.{0}} (r : Z ⟶ W) (t : W ⟶ Y) (z : Z) (U : Y.Opens) (h : t (r z) ∈ U) (f : Γ(Y, U)) : gs (r ≫ t) z U f = (r.stalkMap z).hom (gs t (r z) U f)

theorem **Descends.of_iso** {A : Scheme.{0}} {q : A ⟶ A} {τ : G → (A ⟶ A)} (H : Descends q τ) (hτq : ∀ g, τ g ≫ q = q) (e : Y ≅ A) {q' : Y ⟶ Y} {τ' : G → (Y ⟶ Y)} (hq : e.inv ≫ q' = q ≫ e.inv) (hτ : ∀ g, e.inv ≫ τ' g = τ g ≫ e.inv) : Descends q' τ'

theorem **Descends.injective** (hL : L.IsLB) {q : Y ⟶ Y} {τ : G → (Y ⟶ Y)} (H : Descends q τ) (hq : q ≫ p = p) {D : (Y ⟶ X) →₀ ℤ} {v : Vals L p} (hv : v ∈ Sg L p D) (h : pl L q hq v = 0) : v = 0

theorem **Descends.exists_mem** (hL : L.IsLB) {q : Y ⟶ Y} {τ : G → (Y ⟶ Y)} (H : Descends q τ) (hq : q ≫ p = p) (hτ : ∀ g, τ g ≫ p = p) (hτq : ∀ g, τ g ≫ q = q) {D : (Y ⟶ X) →₀ ℤ} {v : Vals L p} (hv : v ∈ Sg L p (D.mapDomain (q ≫ ·))) (h : ∀ g, pl L (τ g) (hτ g) v = v) : ∃ w ∈ Sg L p D, pl L q hq w = v

theorem **Descends.weight** (hL : L.IsLB) {q : Y ⟶ Y} {τ : G → (Y ⟶ Y)} (H : Descends q τ) (hq : q ≫ p = p) (hτ : ∀ g, τ g ≫ p = p) (hτq : ∀ g, τ g ≫ q = q) {D E F : (Y ⟶ X) →₀ ℤ} {u : (Vals L p)ˣ} (hu : IsG E u) (χ : G → Γ(B, ⊤)ˣ) (hχ : ∀ g, pl L (τ g) (hτ g) u.val = (χ g).val • u.val) (hF : D.mapDomain (q ≫ ·) + E = F) : ∃ f : Sg L p D →ₗ[Γ(B, ⊤)] Vals L p, Function.Injective f ∧ ∀ v, (∃ x, f x = v) ↔ v ∈ Sg L p F ∧ ∀ g, pl L (τ g) (hτ g) v = (χ g).val • v

theorem **secM_equiv** {L : LB A.left} (hL : L.IsLB) (a : ℤ) : Nonempty (SecM A L a T ≃ₗ[Γ(T.left, ⊤)] Sg L (fst T A).left (Finsupp.single (snd T A).left a))

theorem **translPt_eq** (x : T ⟶ A) : translPt A x = (fst T A ≫ x) ∗ snd T A

theorem **translT_fst'** (x : T ⟶ A) : translT A x ≫ fst T A = fst T A

theorem **translT_snd'** (x : T ⟶ A) : translT A x ≫ snd T A = translPt A x

theorem **mulT_fst** (n : ℕ) : mulT A T n ≫ fst T A = fst T A

theorem **mulT_snd** (n : ℕ) : mulT A T n ≫ snd T A = snd T A ^ n

theorem **translT_translPt** (x y : T ⟶ A) : translT A x ≫ translPt A y = translPt A (y ∗ x)

theorem **translT_translT** (x y : T ⟶ A) : translT A x ≫ translT A y = translT A (y ∗ x)

theorem **translPt_one** : translPt A (1 : T ⟶ A) = snd T A

theorem **translT_one** : translT A (1 : T ⟶ A) = 𝟙 _

theorem **translT_mulT** [IsCommMonObj A] (x : T ⟶ A) (n : ℕ) : translT A x ≫ mulT A T n = mulT A T n ≫ translT A (x ^ n)

theorem **descends_mulT** [IsCommMonObj A] (n : ℕ) (H : ∀ (U : A.left.Opens) (f : Γ(A.left, (mulA A n).left ⁻¹ᵁ U)), (∀ y : 𝟙_ (Over S) ⟶ A, y ^ n = 1 → ∀ (e : (mulA A n).left ⁻¹ᵁ U ≤ ((toUnit A ≫ y) ∗ 𝟙 A).left ⁻¹ᵁ ((mulA A n).left ⁻¹ᵁ U)), (((toUnit A ≫ y) ∗ 𝟙 A).left.appLE _ _ e).hom f = f) → ∃! g : Γ(A.left, U), ((mulA A n).left.app U).hom g = f) : Descends (mulT A (baseOver S) n).left (fun y : {y : baseOver S ⟶ A // y ^ n = 1} => (translT A y.1).left)

theorem **pl_inv** {h : Y ⟶ Y} (hh : h ≫ p = p) {u : (Vals L p)ˣ} {c : Γ(B, ⊤)ˣ} (e : pl L h hh u.val = c.val • u.val) : pl L h hh (u⁻¹).val = (c⁻¹).val • (u⁻¹).val

theorem **sections_mulPull_split** : (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NonemptyGtCoordOfIsGt

theorem **pt_eq** (h : a ≫ q = p ≫ q) (x : P) : q (a x) = q (p x)

theorem **HasGen.of_local** (h : a ≫ q = p ≫ q) {N : LB B} {c : GermFn P} {A : Type} (W : A → B.Opens) (hW : ∀ x, ∃ α, x ∈ W α) (F : N.ι → A → GermFn G) (rep : ∀ j α, ∃ y : Γ(G, q ⁻¹ᵁ N.U j ⊓ q ⁻¹ᵁ W α)ˣ, ∀ (Z : Scheme.{0}) (r : Z ⟶ G) (z : Z), q (r z) ∈ N.U j → q (r z) ∈ W α → gm r z _ y = F j α Z r z) (tr : ∀ (Z : Scheme.{0}) (r : Z ⟶ G) (z : Z) (j j' : N.ι) (α α' : A), q (r z) ∈ N.U j → q (r z) ∈ N.U j' → q (r z) ∈ W α → q (r z) ∈ W α' → F j' α' Z r z = gm (r ≫ q) z _ (N.g j j') ∗ F j α Z r z) (wt : ∀ (Z : Scheme.{0}) (r : Z ⟶ P) (z : Z) (j : N.ι) (α : A), q (p (r z)) ∈ N.U j → q (p (r z)) ∈ W α → F j α Z (r ≫ a) z = c Z r z ∗ F j α Z (r ≫ p) z) : HasGen q a p N c

theorem **gm_gen** {N : LB B} {t : LB.Gen N} (ht : t.IsGen) {Z : Scheme.{0}} (r : Z ⟶ B) (z : Z) {j j' : N.ι} (h : r z ∈ N.U j) (h' : r z ∈ N.U j') : gm r z _ (t.s j) = gm r z _ (N.g j j') ∗ gm r z _ (t.s j')

theorem **gm_iso** {L M : LB B} {w : LB.Iso L M} (hw : w.IsGen) {Z : Scheme.{0}} (r : Z ⟶ B) (z : Z) {j j' : M.ι} {k k' : L.ι} (hj : r z ∈ M.U j) (hj' : r z ∈ M.U j') (hk : r z ∈ L.U k) (hk' : r z ∈ L.U k') : gm r z (M.U j ⊓ L.U k) (w.s (j, k)) = gm r z _ (M.g j j') ∗ (gm r z _ (L.g k k'))⁻¹ ∗ gm r z (M.U j' ⊓ L.U k') (w.s (j', k'))

theorem **HasGen.of_gen** (h : a ≫ q = p ≫ q) {N : LB B} (t : LB.Gen N) (ht : t.IsGen) : HasGen q a p N (fun _ _ _ => 1)

theorem **HasGen.div** (h : a ≫ q = p ≫ q) {L M : LB B} {c c' : GermFn P} (hM : HasGen q a p M c) (hL : HasGen q a p L c') : HasGen q a p (LB.div M L) (fun Z r z => c Z r z ∗ (c' Z r z)⁻¹)

theorem **HasGen.iso** (h : a ≫ q = p ≫ q) {L M : LB B} {c : GermFn P} (hL : HasGen q a p L c) (cov : ∀ x, ∃ k, x ∈ L.U k) (w : LB.Iso L M) (hw : w.IsGen) : HasGen q a p M c

theorem **HasGen.iso'** (h : a ≫ q = p ≫ q) {L M : LB B} {c : GermFn P} (hM : HasGen q a p M c) (cov : ∀ x, ∃ k, x ∈ M.U k) (w : LB.Iso L M) (hw : w.IsGen) : HasGen q a p L c

theorem **HasGen.exists_gen** {X : Scheme.{0}} {L : LB X} {K : Type} [Fintype K] {φ : K → (B ⟶ X)} {ε : K → ℤ} {c : GermFn P} (h : a ≫ q = p ≫ q) (H : HasGen q a p (LB.LinComb L φ ε) c) (φ' : K → (G ⟶ X)) (ε' : K → ℤ) (hφ : ∀ k, φ' k = q ≫ φ k) (hε : ∀ k, ε' k = -ε k) (hL : L.IsLB) : ∃ t : LB.Gen (LB.LinComb L φ' ε'), t.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ P) (z : Z) (J), Good L z J → val t (r ≫ a) z J = c Z r z ∗ val t (r ≫ p) z J

theorem **gm_res_pull_top** {X Y Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (f : Y ⟶ X) {V : Y.Opens} (hz : r z ∈ V) (h : V ≤ f ⁻¹ᵁ ⊤) (c : Γ(X, ⊤)ˣ) : gm r z V (res h (pullU f ⊤ c)) = gm (r ≫ f) z ⊤ c

theorem **charUnit_add** (X : Type) [AddCommGroup X] (χ χ' : X) : charUnit S X (χ + χ') = charUnit S X χ ∗ charUnit S X χ'

theorem **wt_add** (X : Type) [AddCommGroup X] (G : Over S) (χ χ' : X) (Z : Scheme.{0}) (r : Z ⟶ (splitTorus S X ⊗ G).left) (z : Z) : wt X G (χ + χ') Z r z = wt X G χ Z r z ∗ wt X G χ' Z r z

theorem **wt_zero** (X : Type) [AddCommGroup X] (G : Over S) : wt X G 0 = fun _ _ _ => 1

theorem **act_q** : act d ≫ d.q = snd _ _ ≫ d.q

theorem **act_q_left** : (act d).left ≫ d.q.left = (snd (splitTorus S E.X) G).left ≫ d.q.left

theorem **exists_mem_chart** (x : E.B.left) : ∃ j, x ∈ E.chart j

theorem **hasGen_basis** (k : Module.Free.ChooseBasisIndex ℤ E.X) : HasGen d.q.left (act d).left (snd (splitTorus S E.X) G).left (E.O (E.basis k)).L (wt E.X G (E.basis k))

theorem **hasGen** (χ : E.X) : HasGen d.q.left (act d).left (snd (splitTorus S E.X) G).left (E.O χ).L (wt E.X G χ)

theorem **exists_const** (hG : GlobalFnsSt) (f : Γ(G.left, ⊤)) (hf : (act d).left.appTop.hom f = (snd (splitTorus S E.X) G).left.appTop.hom f) : ∃ c, G.hom.appTop.hom c = f

theorem **gen_unique_of_invariant** {X Y P S₀ : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (hF : FormEq φ ε φ' ε') (π : Y ⟶ S₀) (e : S₀ ⟶ Y) (he : e ≫ π = 𝟙 _) (a p : P ⟶ Y) (hp : ∀ f : Γ(Y, ⊤), a.appTop.hom f = p.appTop.hom f → ∃ c, π.appTop.hom c = f) {u : LB.Gen (LB.LinComb M φ ε)} {u' : LB.Gen (LB.LinComb M φ' ε')} (hu : u.IsGen) (hu' : u'.IsGen) (h : ∀ y J, Good M y J → val u e y J = val u' e y J) (hw : ∀ y J, Good M y J →

val u a y J ∗ (val u' a y J)⁻¹ = val u p y J ∗ (val u' p y J)⁻¹) (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J) (hJ : Good M z J) : val u r z J = val u' r z J

theorem **comp_Oφ** {T T' : Over (Spec (CommRingCat.of R))} (h : T' ⟶ T) (p : T ⟶ D.P.A) (χ : D.X) (k : Fin 4) : h.left ≫ D.Oφ p χ k = D.Oφ (h ≫ p) χ k

theorem **prod_Oφ** {T : Over (Spec (CommRingCat.of R))} (p : T ⟶ D.P.A) (χ : D.X) {U : Type} [CommGroup U] (h : (T.left ⟶ D.P.A.left) → U) : ∏ k, h (D.Oφ p χ k) ^ Oinvε k = (h (p ∗ (toUnit T ≫ D.c χ)).left)⁻¹ ∗ h (toUnit T ≫ D.c χ).left ∗ h p.left ∗ (h (1 : T ⟶ D.P.A).left)⁻¹

theorem **exists_normalised** {T S₀ P : Over (Spec (CommRingCat.of R))} (q : T ⟶ D.P.A) (π : T ⟶ S₀) (e : S₀ ⟶ T) (he : e ≫ π = 𝟙 _) (heq : e ≫ q = 1) (a p : P ⟶ T) (hq : a ≫ q = p ≫ q) (hπ : a ≫ π = p ≫ π) (hp : ∀ f : Γ(T.left, ⊤), a.left.appTop.hom f = p.left.appTop.hom f → ∃ c, π.left.appTop.hom c = f) (c : D.X → GermFn P.left) (hc : ∀ χ χ' Z r z, c (χ + χ') Z r z = c χ Z r z ∗ c χ' Z r z) (H : ∀ χ, ∃ t : LB.Gen (LB.LinComb D.P.L (D.Oφ q χ) Oinvε), t.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ P.left) (z : Z) (J), Good D.P.L z J → val t (r ≫ a.left) z J = c χ Z r z ∗ val t (r ≫ p.left) z J) : ∃ t : ∀ χ, LB.Gen (LB.LinComb D.P.L (D.Oφ q χ) Oinvε), (∀ χ, (t χ).IsGen) ∧ (∀ χ χ', LB.Gen.Corr D.P.L _ _ _ _ (LB.mulGen D.P.L _ _ _ _ (t (χ + χ')) (LB.pullGen D.P.L (thφ D.P.A (x3 D.P.A)) thε D.cub.t (lift (toUnit _ ≫ D.c χ) (lift (toUnit _ ≫ D.c χ') q)).left)) (LB.mulGen D.P.L _ _ _ _ (t χ) (t χ'))) ∧ (∀

theorem **unit_fst** {S S' : Scheme.{0}} (f : S' ⟶ S) (A : Over S) [GrpObj A] : η[(Over.pullback f).obj A].left ≫ pullback.fst A.hom f = f ≫ η[A].left

theorem **val_charGen** {S Y Z : Scheme.{0}} (X : Type) [AddCommGroup X] (G : Over S) (L : LB Y) (χ : X) (r : Z ⟶ (splitTorus S X ⊗ G).left) (z : Z) (J : (Z ⟶ Y) → L.ι) : val (charGen X G L χ) r z J = gm (r ≫ (fst (splitTorus S X) G).left) z ⊤ (charUnit S X χ)

theorem **nonempty_gtCoord_of_isGt** : GlobalFnsSt → ∀ (R : Type) [CommRing R] [IsDomain R] (I : Ideal R) (D : DegPre R) (e : ℕ) (G : Over (Spec (CommRingCat.of R))) [GrpObj G] [IsCommMonObj G] (E : SplitExt (Spec (CommRingCat.of (R ⧸ I ^ (e + 1))))) (β : E.B ≅ thickObj R I e D.P.A) (xe : D.X ≃+ E.X), IsMonHom β.hom → (∀ χ : D.X, ∃ w : LB.Iso (E.O (xe χ)).L (LB.LinComb D.P.L (fun k => ((Over.map (thick R I e)).map β.hom ≫ D.prBe I e).left ≫ D.Oφ (𝟙 _) χ k) Oε), w.IsGen) → E.IsGt (thickObj R I e G) → Nonempty (GtCoord I D e G)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NonemptyPerfectSectionsComplexOfSmoothProj

theorem **nonempty_perf_of_smooth_isProjBy** : ∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NonemptyPerfectSectionsComplexOfSmoothProjStatement

theorem **_root_.GlobalTheory.T.nonempty_perfectSectionsComplex_of_smooth_proj** : ∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.NonemptyTorusExtensionOfIsGt

theorem **nonempty_torusExtension_of_isGt** {K : Type} [Field K] (E : SplitExt (Spec (.of K))) (G : Over (Spec (.of K))) [GrpObj G] (h : E.IsGt G) : Nonempty (TorusExtension G)

theorem **isSemiAbelian_of_isGt** : ∀ (S : Scheme.{0}) (E : SplitExt S) (G : Over S) [GrpObj G] [IsCommMonObj G], E.IsGt G → IsSemiAbelian G

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.OneTmulNeZeroOfDomain

theorem **one_tmul_ne_zero_of_domain** {A P k : Type} [CommRing A] [IsDomain A] [AddCommGroup P] [Module A P] [Field k] [Algebra A k] (hinj : Function.Injective (algebraMap A k)) (y : P) (hy : ∀ a : A, a • y = 0 → a = 0) : (1 : k) ⊗ₜ[A] y ≠ 0

theorem **one_tmul_ne_zero** {R₀ P k : Type} [CommRing R₀] [AddCommGroup P] [Module R₀ P] [Field k] [Algebra R₀ k] (p : Ideal R₀) (hk : ∀ a, algebraMap R₀ k a = 0 ↔ a ∈ p) (hP : Module.IsTorsionBySet R₀ P p) (y : P) (hy : ∀ a : R₀, a • y = 0 → a ∈ p) : (1 : k) ⊗ₜ[R₀] y ≠ 0

theorem **one_tmul_eq_zero** {R₀ M C₀ C₁ k : Type} [CommRing R₀] [AddCommGroup M] [Module R₀ M] [AddCommGroup C₀] [Module R₀ C₀] [AddCommGroup C₁] [Module R₀ C₁] [Field k] [Algebra R₀ k] (φ : C₀ →ₗ[R₀] C₁) (hφ : Function.Injective (φ.baseChange k)) (y : M ⊗[R₀] C₀) (hy : φ.lTensor M y = 0) : (1 : k) ⊗ₜ[R₀] y = 0

theorem **exists_of_torsion** {R₀ N C : Type} [CommRing R₀] [AddCommGroup N] [Module R₀ N] [AddCommGroup C] [Module R₀ C] [Module.Flat R₀ C] (p : Ideal R₀) (hp : p.FG) (x : N ⊗[R₀] C) (hx : ∀ a ∈ p, a • x = 0) : ∃ y, (Submodule.torsionBySet R₀ N p).subtype.rTensor C y = x

theorem **injective_baseChange** : ∀ (R₀ R C₀ C₁ : Type) [CommRing R₀] [IsNoetherianRing R₀] [CommRing R] [Algebra R₀ R] [AddCommGroup C₀] [Module R₀ C₀] [AddCommGroup C₁] [Module R₀ C₁] [Module.Flat R₀ C₀] [Module.Flat R₀ C₁] (φ : C₀ →ₗ[R₀] C₁), (∀ (k : Type) [Field k] [Algebra R k] [Algebra R₀ k] [IsScalarTower R₀ R k], Function.Injective (φ.baseChange k)) → Function.Injective (φ.baseChange R)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.PolBaseChangeOfRelAmpleLocal

theorem **lbIso_refl** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) : LBIso L L

theorem **eSec_comm** (h : IsGrpPullback A' A f φ) : eSec A' ≫ φ = f ≫ eSec A

theorem **inv_comm** (h : IsGrpPullback A' A f φ) : (GrpObj.inv : A' ⟶ A').left ≫ φ = φ ≫ (GrpObj.inv : A ⟶ A).left

theorem **rigidified_pull** (h : IsGrpPullback A' A f φ) (hr : Rigidified A L) : Rigidified A' (L.pull φ)

theorem **sym_pull** (h : IsGrpPullback A' A f φ) (hs : Sym A L) : Sym A' (L.pull φ)

theorem **relAmple_restrict** (h : IsPullback φ A'.hom A.hom f) (hL : L.IsLB) (hA : RelAmple A L) (V : S.Opens) (hV : IsAffineOpen V) : RelAmple (Over.mk (A'.hom ∣_ (f ⁻¹ᵁ V))) ((L.pull φ).pull (A'.hom ⁻¹ᵁ (f ⁻¹ᵁ V))).ι)

theorem **polBaseChange_of_relAmple_local** : (∀ (S : Scheme.{0}) (A : Over S) (L : LB A.left), L.IsLB → QuasiCompact A.hom → QuasiSeparated A.hom → (∀ s : S, ∃ (U : S.Opens) (_ : s ∈ U), RelAmple (Over.mk (A.hom ∣_ U)) (L.pull (A.hom ⁻¹ᵁ U)).ι)) → RelAmple A L) → PolBaseChangeSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.PowEqOneOfLocalLift

theorem **theta_pt_pow** {S : Scheme.{0}} (P : PolAV S) (k : ℕ) {T : Over S} (θ : ThetaGrp P.A P.L (k : ℤ) T) : θ.pt ^ k = 1

theorem **pow_eq_one_of_local_lift** {S : Scheme.{0}} (P : PolAV S) (k : ℕ) {T : Over S} (x : T ⟶ P.A) (h : ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L (k : ℤ) (overRes V), θ.pt = overResι V ≫ x) : x ^ k = 1

theorem **mem_tors_of_weilIs** {S : Scheme.{0}} (P : PolAV S) (k : ℕ) {T : Over S} {x y : T ⟶ P.A} {c : Γ(T.left, ⊤)ˣ} (h : WeilIs P.A P.L (k : ℤ) x y c) : x ∈ tors P T k ∧ y ∈ tors P T k

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.PrimitiveCocycle

theorem **prod_lam** {T : Over S} (x y : T ⟶ B) {U : Type} [CommGroup U] (h : (T.left ⟶ B.left) → U) : ∏ I, h ((lift x y).left ≫ lamφ B I) ^ lamε I = h (x ∗ y).left ∗ (h x.left)⁻¹ ∗ (h y.left)⁻¹ ∗ h (1 : T ⟶ B).left

theorem **exists_normalised** {L : LB B.left} (hL : L.IsLB) (t : LB.Gen (LB.LinComb L (lamφ B) lamε)) (ht : t.IsGen) : ∃ u : LB.Gen (LB.LinComb L (lamφ B) lamε), u.IsGen ∧ ∀ (T : Over S) (z : T.left) (J), Good L z J → val₂ u z J (1 : T ⟶ B) 1 = 1

theorem **val_mul_eq** (hG : GlobalFnsSt) (hB : IsAbelianScheme B) {L : LB B.left} (hL : L.IsLB) (u : LB.Gen (LB.LinComb L (lamφ B) lamε)) (hu : u.IsGen) (e : 𝟙_ (Over S) ⟶ G3 B) (a b c d a' b' c' d' : G3 B ⟶ B) (hF : ∀ (V : Type) [CommGroup V] (k : (G3 B ⟶ B) → V), (k (a' ∗ b') ∗ (k a')⁻¹ ∗ (k b')⁻¹ ∗ k 1) ∗ (k (c' ∗ d') ∗ (k c')⁻¹ ∗ (k d')⁻¹ ∗ k 1) = (k (a ∗ b) ∗ (k a)⁻¹ ∗ (k b)⁻¹ ∗ k 1) ∗ (k (c ∗ d) ∗ (k c)⁻¹ ∗ (k d)⁻¹ ∗ k 1)) (ha : e ≫ a = 1) (hb : e ≫ b = 1) (hc : e ≫ c = 1) (hd : e ≫ d = 1) (ha' : e ≫ a' = 1) (hb' : e ≫ b' = 1) (hc' : e ≫ c' = 1) (hd' : e ≫ d' = 1) {T : Over S} (ρ : T ⟶ G3 B) {z : T.left} {J} (hJ : Good L z J) : val₂ u z J (ρ ≫ a) (ρ ≫ b) ∗ val₂ u z J (ρ ≫ c) (ρ ≫ d) = val₂ u z J (ρ ≫ a') (ρ ≫ b') ∗ val₂ u z J (ρ ≫ c') (ρ ≫ d')

theorem **primitive_cocycle** : GlobalFnsSt → ∀ (S : Scheme.{0}) (B : Over S) [GrpObj B] [IsCommMonObj B], IsAbelianScheme B → ∀ (O : RigLB B), (∃ t : LB.Gen (abelian₂ B O.L), t.IsGen) → ∃ t : LB.Gen (abelian₂ B O.L), t.IsGen ∧ ∀ (Z : Over S) (z : Z.left) (J : (Z.left ⟶ B.left) → O.L.ι), Geo.Tor.Good O.L z J → let v : (Z ⟶ B) → (Z ⟶ B) → (Z.left.presheaf.stalk z)ˣ

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.PrincipalPolarisationUniqueSection

theorem **principalPolarisation_unique_section** : GlobalTheory.Published.principalPolarisationOneSection

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.QuasiCompactIffLocallyOfFinitePresentation

theorem **isPullback_torsor** {S : Scheme.{u}} {T G B : Over S} [GrpObj T] [GrpObj G] [GrpObj B] (i : T ⟶ G) (q : G ⟶ B) [IsMonHom q] (h : IsPullback i (toUnit _) q η[B]) : IsPullback ((fst T G ≫ i) ∗ snd T G) (snd T G) q q

theorem **geometricallyConnected_of_isPullback** {X Y X' Y' : Scheme.{u}} {a : X' ⟶ X} {p : X' ⟶ Y'} {q : X ⟶ Y} {s : Y' ⟶ Y} (h : IsPullback a p q s) [Surjective s] [GeometricallyConnected p] : GeometricallyConnected q

theorem **quasiCompact_of_isPullback** {X Y X' Y' : Scheme.{u}} {a : X' ⟶ X} {p : X' ⟶ Y'} {q : X ⟶ Y} {s : Y' ⟶ Y} (h : IsPullback a p q s) [Surjective s] (hs : IsOpenMap s) [QuasiCompact p] : QuasiCompact q

theorem **geometricallyConnected_of_isAlgClosed** {X S : Scheme.{u}} (f : X ⟶ S) (h : ∀ (k : Type u) [Field k] [IsAlgClosed k] (s : Spec (CommRingCat.of k) ⟶ S), ConnectedSpace ₁(pullback f s)) : GeometricallyConnected f

theorem **quasiCompact_iff_locallyOfFinitePresentation** [Module.Finite ℤ X] [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) [hf : Flat q.left] [hs : Surjective q.left] : QuasiCompact q.left ↔ LocallyOfFinitePresentation q.left

theorem **geometricallyConnected_quotient** [Module.Free ℤ X] [IsMonHom q] (hk : IsPullback i (toUnit _) q η[B]) [hs : Surjective q.left] : GeometricallyConnected q.left

theorem **isAbelianScheme_iff** {S : Scheme.{u}} (A : Over S) [GrpObj A] : Semi.IsAbelianScheme A ↔ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsAbelianScheme A

theorem **isShortExact_of_torusExtension** (E : Semi.TorusExtension G) : letI

theorem **isTorusExtension_iff** : IsCommMonObj G ∧ Nonempty (Semi.TorusExtension G) ↔ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsTorusExtension G

theorem **geometricallyConnected_of_torusExtension** (E : Semi.TorusExtension G) : GeometricallyConnected G.hom

theorem **isCommMonObj_of_torusExtension** {k : Type} [Field k] {G : Over (Spec (CommRingCat.of k))} [GrpObj G] (E : Semi.TorusExtension G) : IsCommMonObj G

theorem **nonempty_torusExtension_iff** {k : Type} [Field k] (G : Over (Spec (CommRingCat.of k))) [GrpObj G] : Nonempty (Semi.TorusExtension G) ↔ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsTorusExtension G

theorem **isSemiAbelian_iff** {S : Scheme.{u}} (G : Over S) [GrpObj G] : Semi.IsSemiAbelian G ↔ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsSemiAbelianScheme G

theorem **semiAbelian_notions_equiv** : (∀ (S : Scheme.{u}) (A : Over S) [GrpObj A], Semi.IsAbelianScheme A ↔ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsAbelianScheme A) ∧ (∀ (k : Type u) [Field k] (G : Over (Spec (CommRingCat.of k))) [GrpObj G] (X : Type u) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (B : Over (Spec (CommRingCat.of k))) [GrpObj B] (i : Semi.splitTorus (Spec (CommRingCat.of k)) X ⟶ G) (q : G ⟶ B), (Semi.IsAbelianScheme B ∧ IsMonHom q ∧ Flat q.left ∧ Surjective q.left ∧ QuasiCompact q.left ∧ IsMonHom i ∧ IsPullback i (toUnit _) q (η : 𝟙_ _ ⟶ B)) ↔ (IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsAbelianScheme B ∧ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsShortExact i q)) ∧ (∀ (k : Type u) [Field k] (G : Over (Spec (CommRingCat.of k))) [GrpObj G], IsCommMonObj G ∧ Nonempty (Semi.TorusExtension G) ↔ IndefiniteRowSideFullHecke.SemiAbelianSchemes.IsTorusExte

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.QuotientIsAbelianSchemeOfQuotientScheme

lemma **exists_desc** {X Y Z : Over S} (p : X ⟶ Y) [Flat p.left] [Surjective p.left] [QuasiCompact p.left] (f : X ⟶ Z) (h : ∀ (W : Over S) (a b : W ⟶ X), a ≫ p = b ≫ p → a ≫ f = b ≫ f) : ∃ β : Y ⟶ Z, p ≫ β = f

lemma **flat_tensorHom** {X Y X' Y' : Over S} (f : X ⟶ Y) (g : X' ⟶ Y') [hf : Flat f.left] [hg : Flat g.left] : Flat (f ⊗ₘ g).left

lemma **surjective_tensorHom** {X Y X' Y' : Over S} (f : X ⟶ Y) (g : X' ⟶ Y') [hf : Surjective f.left] [hg : Surjective g.left] : Surjective (f ⊗ₘ g).left

lemma **quasiCompact_tensorHom** {X Y X' Y' : Over S} (f : X ⟶ Y) (g : X' ⟶ Y') [hf : QuasiCompact f.left] [hg : QuasiCompact g.left] : QuasiCompact (f ⊗ₘ g).left

theorem **exists_grpObj** {A H Q : Over S} [GrpObj A] [IsCommMonObj A] [GrpObj H] (i : H ⟶ A) [IsMonHom i] (q : A ⟶ Q) [Flat q.left] [Surjective q.left] [QuasiCompact q.left] (hq : IsPullback ((fst H A ≫ i) ∗ snd H A) (snd H A) q q) : ∃ _ : GrpObj Q, IsCommMonObj Q ∧ IsMonHom q ∧ IsPullback i (toUnit H) q η[Q]

lemma **isReduced_of_flat_of_surjective** {X Y : Scheme.{0}} (p : X ⟶ Y) [Flat p] [Surjective p] [IsReduced X] : IsReduced Y

lemma **geometricallyReduced_of_comp** {X Y Z : Scheme.{0}} (p : X ⟶ Y) (f : Y ⟶ Z) [Flat p] [Surjective p] [h : GeometricallyReduced (p ≫ f)] : GeometricallyReduced f

lemma **geometricallyConnected_of_comp** {X Y Z : Scheme.{0}} (p : X ⟶ Y) (f : Y ⟶ Z) [Surjective p] [h : GeometricallyConnected (p ≫ f)] : GeometricallyConnected f

lemma **smooth_of_grpObj_of_flat** (Q : Over S) [GrpObj Q] [Flat Q.hom] [LocallyOfFinitePresentation Q.hom] [GeometricallyReduced Q.hom] : Smooth Q.hom

theorem **quotient_isAbelianScheme_of_quotientScheme** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (R : Type) [CommRing R] [IsNoetherianRing R] (P : PolAV (Spec (CommRingCat.of R))) (H : Over (Spec (CommRingCat.of R))) [GrpObj H] (i : H ⟶ P.A), IsMonHom i → Mono i → IsFinite H.hom → Flat H.hom → ∃ (Q : Over (Spec (CommRingCat.of R))) (q : P.A ⟶ Q), IsFinite q.left ∧ Flat q.left ∧ Surjective q.left ∧ IsSeparated Q.hom ∧ LocallyOfFiniteType Q.hom ∧ IsPullback ((fst H P.A ≫ i) ∗ snd H P.A) (snd H P.A) q q) → QuotientSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.RangeRestrictCompSubtypeOfOrthogonalOfNe

theorem **nonempty_homology_equiv_of_retract** (S : ShortComplex (ModuleCat.{v} R)) {W : Type*} [AddCommGroup W] [Module R W] (I : W →ₗ[R] S.X₂) (Q : S.X₂ →ₗ[R] W) (hI : ∀ w, S.g.hom (I w) = 0) (hQ : ∀ y, Q (S.f.hom y) = 0) (hQI : ∀ w, Q (I w) = w) (hex : ∀ x, S.g.hom x = 0 → Q x = 0 → ∃ y, S.f.hom y = x) : Nonempty (S.homology ≃ₗ[R] W)

theorem **restrictDeg_apply** (A : Cube.{v} ι R) (p : ℕ) (x : ∀ s, A.obj s) (s : {s : Finset ι // s.card = p}) : restrictDeg A p x s = x s.1

theorem **degProj_apply** (A : Cube.{v} ι R) (p : ℕ) (x : ∀ s, A.obj s) (s : Finset ι) : degProj A p x s = if s.card = p then x s else 0

theorem **degSign_apply** (A : Cube.{v} ι R) (x : ∀ s, A.obj s) (s : Finset ι) : degSign A x s = (-1 : ℤ) ^ s.card • x s

theorem **restrictDeg_extend** (A : Cube.{v} ι R) (p : ℕ) (x : Term A p) : restrictDeg A p (extend A p x) = x

theorem **ext_restrictDeg** {A : Cube.{v} ι R} {x y : ∀ s, A.obj s} (h : ∀ p, restrictDeg A p x = restrictDeg A p y) : x = y

theorem **restrictDeg_degProj** (A : Cube.{v} ι R) (p q : ℕ) (x : ∀ s, A.obj s) : restrictDeg A p (degProj A q x) = if p = q then restrictDeg A p x else 0

theorem **restrictDeg_degSign** (A : Cube.{v} ι R) (p : ℕ) (x : ∀ s, A.obj s) : restrictDeg A p (degSign A x) = (-1 : ℤ) ^ p • restrictDeg A p x

theorem **degProj_mul_degProj** (A : Cube.{v} ι R) (p q : ℕ) : degProj A p ∗ degProj A q = if p = q then degProj A p else 0

theorem **degSign_mul_degSign** (A : Cube.{v} ι R) : degSign A ∗ degSign A = 1

theorem **restrictDeg_totalD_succ** (A : Cube.{v} ι R) (p : ℕ) (x : ∀ s, A.obj s) : restrictDeg A (p + 1) (totalD A x) = d A p (restrictDeg A p x)

theorem **restrictDeg_totalD_zero** (A : Cube.{v} ι R) (x : ∀ s, A.obj s) : restrictDeg A 0 (totalD A x) = 0

theorem **totalD_extend** (A : Cube.{v} ι R) (p : ℕ) (x : Term A p) : totalD A (extend A p x) = extend A (p + 1) (d A p x)

theorem **totalD_mul_degSign** (A : Cube.{v} ι R) : (totalD A : Module.End R _) ∗ degSign A = -(degSign A ∗ totalD A)

theorem **nonempty_cubeHomology_equiv_of_retract** (A : Cube.{v} ι R) (n : ℕ) {W : Type*} [AddCommGroup W] [Module R W] (I : W →ₗ[R] ∀ s, A.obj s) (Q : (∀ s, A.obj s) →ₗ[R] W) (hI : ∀ w, totalD A (I w) = 0) (hPI : ∀ w, degProj A n (I w) = I w) (hQ : ∀ y, Q (totalD A y) = 0) (hQI : ∀ w, Q (I w) = w) (hex : ∀ x, totalD A x = 0 → degProj A n x = x → Q x = 0 → ∃ y, totalD A y = x) : Nonempty ((cubeComplex A).homology n ≃ₗ[R] W)

theorem **nonempty_cubeHomology_equiv_of_homotopy** (A : Cube.{v} ι R) (n : ℕ) {V W : Type*} [AddCommGroup V] [Module R V] [AddCommGroup W] [Module R W] (Φ : V ≃ₗ[R] ∀ s, A.obj s) (T P e h : Module.End R V) (hT : ∀ x, totalD A (Φ x) = Φ (T x)) (hP : ∀ x, degProj A n (Φ x) = Φ (P x)) (hh : 1 - e = T ∗ h + h ∗ T) (hTe : T ∗ e = 0) (heT : e ∗ T = 0) (heP : P ∗ e = e ∗ P) (I : W →ₗ[R] V) (Q : V →ₗ[R] W) (hIQ : ∀ x, I (Q x) = e (P x)) (hQI : ∀ w, Q (I w) = w) : Nonempty ((cubeComplex A).homology n ≃ₗ[R] W)

theorem **exists_comp_comp_eq_of_field** {V W : Type*} [AddCommGroup V] [Module K V] [AddCommGroup W] [Module K W] (f : V →ₗ[K] W) : ∃ g : W →ₗ[K] V, ∀ x, f (g (f x)) = f x

theorem **Cube.exists_contraction** (A : Cube.{v} ι K) : ∃ e h : Module.End K (∀ s, A.obj s), 1 - e = (totalD A : Module.End K _) ∗ h + h ∗ totalD A ∧ (totalD A : Module.End K _) ∗ e = 0 ∧ e ∗ e = e ∧ (∀ p, degProj A p ∗ e = e ∗ degProj A p) ∧ degSign A ∗ e = e ∗ degSign A ∧ degSign A ∗ h = -(h ∗ degSign A)

theorem **tensor_homotopy** {D e h ε : Module.End K V} {D' e' h' : Module.End K V'} (hh : 1 - e = D ∗ h + h ∗ D) (hDe : D ∗ e = 0) (heD : e ∗ D = 0) (hDε : D ∗ ε = -(ε ∗ D)) (hεe : ε ∗ e = e ∗ ε) (hεh : ε ∗ h = -(h ∗ ε)) (hεε : ε ∗ ε = 1) (hh' : 1 - e' = D' ∗ h' + h' ∗ D') : (1 : Module.End K (V ⊗[K] V')) - TensorProduct.map e e' = (TensorProduct.map D 1 + TensorProduct.map ε D') ∗ (TensorProduct.map h 1 + TensorProduct.map (ε ∗ e) h') + (TensorProduct.map h 1 + TensorProduct.map (ε ∗ e) h') ∗ (TensorProduct.map D 1 + TensorProduct.map ε D')

theorem **rangeRestrict_comp_subtype_of_orthogonal** {f : ℕ → Module.End K V} (hf : ∀ p q, f p ∗ f q = if p = q then f p else 0) (p : ℕ) : (f p).rangeRestrict ∘ₗ (LinearMap.range (f p)).subtype = LinearMap.id

theorem **rangeRestrict_comp_subtype_of_orthogonal_of_ne** {f : ℕ → Module.End K V} (hf : ∀ p q, f p ∗ f q = if p = q then f p else 0) {p q : ℕ} (h : p ≠ q) : (f p).rangeRestrict ∘ₗ (LinearMap.range (f q)).subtype = 0

theorem **exists_retraction_tensor_ranges** {f : ℕ → Module.End K V} {g : ℕ → Module.End K V'} (hf : ∀ p q, f p ∗ f q = if p = q then f p else 0) (hg : ∀ p q, g p ∗ g q = if p = q then g p else 0) (S : Finset (ℕ × ℕ)) : ∃ (I : (∀ x : S, LinearMap.range (f x.1.1) ⊗[K] LinearMap.range (g x.1.2)) →ₗ[K] V ⊗[K] V') (Q : V ⊗[K] V' →ₗ[K] ∀ x : S, LinearMap.range (f x.1.1) ⊗[K] LinearMap.range (g x.1.2)), (∀ z, I (Q z) = ∑ x ∈ S, TensorProduct.map (f x.1) (g x.2) z) ∧ ∀ w, Q (I w) = w

theorem **extTensor_obj** (A : Cube.{v} ι K) (B : Cube.{v} κ K) (w : Finset (ι ⊕ₗ κ)) : (extTensor A B).obj w = ModuleCat.of K (A.obj (part (inlEmb ι κ) w) ⊗[K] B.obj (part (inrEmb ι κ) w))

theorem **res_extTensor** (A : Cube.{v} ι K) (B : Cube.{v} κ K) {w w' : Finset (ι ⊕ₗ κ)} (h : w ⊆ w') : res (extTensor A B) h = TensorProduct.map (res A (monotone_part _ h)) (res B (monotone_part _ h))

theorem **sum_parts** {M : Type*} [AddCommMonoid M] (w : Finset (ι ⊕ₗ κ)) (F : ι ⊕ₗ κ → M) : ∑ j ∈ w, F j = ∑ i ∈ part (inlEmb ι κ) w, F (inlEmb ι κ i) + ∑ k ∈ part (inrEmb ι κ) w, F (inrEmb ι κ k)

theorem **card_parts** (w : Finset (ι ⊕ₗ κ)) : w.card = (part (inlEmb ι κ) w).card + (part (inrEmb ι κ) w).card

theorem **cubeSign_inlEmb** (w : Finset (ι ⊕ₗ κ)) (i : ι) : cubeSign w (inlEmb ι κ i) = cubeSign (part (inlEmb ι κ) w) i

theorem **cubeSign_inrEmb** (w : Finset (ι ⊕ₗ κ)) (k : κ) : cubeSign w (inrEmb ι κ k) = (-1) ^ (part (inlEmb ι κ) w).card ∗ cubeSign (part (inrEmb ι κ) w) k

theorem **totalTensorEquiv_tmul** (x : ∀ s, A.obj s) (y : ∀ t, B.obj t) (w : Finset (ι ⊕ₗ κ)) : totalTensorEquiv A B (x ⊗ₜ y) w = x (part (inlEmb ι κ) w) ⊗ₜ[K] y (part (inrEmb ι κ) w)

theorem **totalD_totalTensorEquiv** (z : (∀ s, A.obj s) ⊗[K] (∀ t, B.obj t)) : totalD (extTensor A B) (totalTensorEquiv A B z) = totalTensorEquiv A B ((TensorProduct.map (totalD A) 1 + TensorProduct.map (degSign A) (totalD B) : Module.End K _) z)

theorem **degProj_totalTensorEquiv** (n : ℕ) (z : (∀ s, A.obj s) ⊗[K] (∀ t, B.obj t)) : degProj (extTensor A B) n (totalTensorEquiv A B z) = totalTensorEquiv A B ((∑ x ∈ Finset.HasAntidiagonal.antidiagonal n, TensorProduct.map (degProj A x.1) (degProj B x.2) : Module.End K

_) z)

theorem **nonempty_trunc₂_iso_extTensor** (C : Cube.{v} (ι ⊕ₗ κ) K) (μ : ∀ (s : Finset ι) (t : Finset κ), s.Nonempty → t.Nonempty → A.obj s ⊗[K] B.obj t →ₗ[K] C.obj (sumVertex s t)) (hμ : ∀ s t hs ht, Function.Bijective (μ s t hs ht)) (hnat : ∀ {s s' : Finset ι} {t t' : Finset κ} (hs : s.Nonempty) (ht : t.Nonempty) (h : s ⊆ s') (h' : t ⊆ t') (a : A.obj s) (b : B.obj t), μ s' t' (hs.mono h) (ht.mono h') (res A h a ⊗ₜ res B h' b) = res C (sumVertex_mono h h') (μ s t hs ht (a ⊗ₜ b))) : Nonempty (trunc₂ C ≅ trunc₂ (extTensor A B))

theorem **nonempty_cubeHomology_extTensor_equiv** (A : Cube.{v} ι K) (B : Cube.{v} κ K) (n : ℕ) : Nonempty ((cubeComplex (extTensor A B)).homology n ≃ₗ[K] ∀ x : antidiagonal n, (cubeComplex A).homology x.1.1 ⊗[K] (cubeComplex B).homology x.1.2)

theorem **cube_kuenneth** (A : Cube.{v} ι K) (B : Cube.{v} κ K) (hA : ∀ p, Module.Finite K ((cubeComplex A).homology p)) (hB : ∀ q, Module.Finite K ((cubeComplex B).homology q)) (n : ℕ) : Module.Finite K ((cubeComplex (extTensor A B)).homology n) ∧ finrank K ((cubeComplex (extTensor A B)).homology n) = ∑ x ∈ antidiagonal n, finrank K ((cubeComplex A).homology x.1) * finrank K ((cubeComplex B).homology x.2)

theorem **cech_kuenneth** (A : Cube.{v} ι K) (B : Cube.{v} κ K) (hA : ∀ p, Module.Finite K ((cechComplex A).homology p)) (hB : ∀ q, Module.Finite K ((cechComplex B).homology q)) (n : ℕ) : Module.Finite K ((cubeComplex (trunc₂ (extTensor A B))).homology (n + 2)) ∧ finrank K ((cubeComplex (trunc₂ (extTensor A B))).homology (n + 2)) = ∑ x ∈ antidiagonal n, finrank K ((cechComplex A).homology x.1) * finrank K ((cechComplex B).homology x.2)

theorem **cech_kuenneth_of_bijective** (A : Cube.{v} ι K) (B : Cube.{v} κ K) (C : Cube.{v} (ι ⊕ₗ κ) K) (μ : ∀ (s : Finset ι) (t : Finset κ), s.Nonempty → t.Nonempty → A.obj s ⊗[K] B.obj t →ₗ[K] C.obj (sumVertex s t)) (hμ : ∀ s t hs ht, Function.Bijective (μ s t hs ht)) (hnat : ∀ {s s' : Finset ι} {t t' : Finset κ} (hs : s.Nonempty) (ht : t.Nonempty) (h : s ⊆ s') (h' : t ⊆ t') (a : A.obj s) (b : B.obj t), μ s' t' (hs.mono h) (ht.mono h') (res A h a ⊗ₜ res B h' b) = res C (sumVertex_mono h h') (μ s t hs ht (a ⊗ₜ b))) (hA : ∀ p, Module.Finite K ((cechComplex A).homology p)) (hB : ∀ q, Module.Finite K ((cechComplex B).homology q)) (n : ℕ) : Module.Finite K ((cubeComplex (trunc₂ C)).homology (n + 2)) ∧ finrank K ((cubeComplex (trunc₂ C)).homology (n + 2)) = ∑ x ∈ antidiagonal n, finrank K ((cechComplex A).homology x.1) * finrank K ((cechComplex B).homology x.2)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.RelAmpleOfLocallyOnBase

theorem **glue_pack** {ι : Type} {U : ι → Z.Opens} {G : ι → Fn Z} (hU : ∀ z, ∃ i, z ∈ U i) (hG : ∀ i z, IsUnit (G i z)) [CompactSpace Z] [QuasiSeparatedSpace Z] (hT : ∀ i j, G i * inv (G j) ∈ Reg (U i ⊓ U j)) {R : Type} [CommRing R] (φ : R →+* Fn Z) (hφ : ∀ r, φ r ∈ Reg ⊤) {Λ : Type} [Finite Λ] (a : Λ → R) (ha : Ideal.span (Set.range a) = ⊤) (W : Λ → Z.Opens) (hW : ∀ l, IsLoc (W l) ⊤ (φ (a l))) (H : ∀ l, Pack U G (W l) (insert (inv (φ (a l))) (Set.range φ))) : ∃ N : ℕ, 1 ≤ N ∧ ∃ M : ℕ, ∃ σ : Fin (M + 1) → Fn Z, (∀ t, Sc U G ⊤ N (σ t)) ∧ (∀ z, ∃ t, IsUnit (σ t z)) ∧ ∀ t D, IsLoc D ⊤ (σ t) → IsAffineOpen D ∧ Gen D (Set.range φ ∪ Set.range fun m => σ m * inv (σ t))

theorem **pb_apply** (h : Y ⟶ Z) (F : Fn Z) (y : Y) : pb h F y = (h.stalkMap y).hom (F (h y))

theorem **pb_comp** (h : Y ⟶ Z) (g : Z ⟶ X) (F : Fn X) : pb (h ≫ g) F = pb h (pb g F)

theorem **pb_unit** (h : Y ⟶ Z) (F : Fn Z) (y : Y) : IsUnit (pb h F y) ↔ IsUnit (F (h y))

theorem **pb_inv** (h : Y ⟶ Z) (F : Fn Z) : pb h (inv F) = inv (pb h F)

theorem **gs_id** (V : Z.Opens) (z : Z) (hz : z ∈ V) (a : Γ(Z, V)) : gs (𝟙 Z) z V a = (Z.presheaf.germ V z hz).hom a

theorem **mem_reg_iff** {V : Z.Opens} {F : Fn Z} : F ∈ Reg V ↔ ∃ a : Γ(Z, V), ∀ z ∈ V, gs (𝟙 Z) z V a = F z

theorem **stalkMap_gs** (h : Y ⟶ Z) (g : Z ⟶ X) (y : Y) (V : X.Opens) (hy : g (h y) ∈ V) (a : Γ(X, V)) : (h.stalkMap y).hom (gs g (h y) V a) = gs (h ≫ g) y V a

theorem **reg_gs** (g : Z ⟶ X) (V : X.Opens) (a : Γ(X, V)) : (fun z => gs g z V a) ∈ Reg (g ⁻¹ᵁ V)

theorem **reg_pull** (h : Y ⟶ Z) {V : Z.Opens} {F : Fn Z} (hF : F ∈ Reg V) : pb h F ∈ Reg (h ⁻¹ᵁ V)

theorem **cst_apply** (p : Z ⟶ X) (r : Γ(X, ⊤)) (z : Z) : cst p r z = gs p z ⊤ r

theorem **pb_cst** (h : Y ⟶ Z) (p : Z ⟶ X) (r : Γ(X, ⊤)) : pb h (cst p r) = cst (h ≫ p) r

theorem **reg_cst** (p : Z ⟶ X) (r : Γ(X, ⊤)) : cst p r ∈ Reg ⊤

theorem **loc_cst** (r : Γ(X, ⊤)) : IsLoc (X.basicOpen r) ⊤ (cst (𝟙 X) r)

theorem **pb_inj** {F F' : Fn Z} {y : Y} (e : pb h F y = pb h F' y) : F (h y) = F' (h y)

theorem **pb_surj** (F' : Fn Y) : ∃ F, pb h F = F'

theorem **reg_push** {V : Y.Opens} {F : Fn Z} (hF : pb h F ∈ Reg V) : F ∈ Reg (h ''ᵁ V)

theorem **gen_push** {V : Y.Opens} {S : Set (Fn Z)} (hS : Gen V (pb h '' S)) : Gen (h ''ᵁ V) S

theorem **pack_push** {ι : Type} {U : ι → Z.Opens} {G : ι → Fn Z} {S : Set (Fn Z)} (H : Pack (fun i => h ⁻¹ᵁ U i) (fun i => pb h (G i)) ⊤ (pb h '' S)) : Pack U G (h ''ᵁ ⊤) S

theorem **gen_iff** {R T : Type} [CommRing R] {D : Z.Opens} (ψ : R →+* Γ(Z, D)) (r : T → Γ(Z, D)) (φ : R → Fn Z) (ρ : T → Fn Z) (hψ : ∀ c, ∀ z ∈ D, gs (𝟙 Z) z D (ψ c) = φ c z) (hr : ∀ t, ∀ z ∈ D, gs (𝟙 Z) z D (r t) = ρ t z) : Gen D (Set.range φ ∪ Set.range ρ) ↔ Function.Surjective (MvPolynomial.eval₂Hom ψ r)

theorem **sec_tr** (s : LB.Sec P) (y : Y) {k k' : P.ι} (hk : y ∈ P.U k) (hk' : y ∈ P.U k') : gs (𝟙 Y) y _ (s.f k) = (gm (𝟙 Y) y _ (P.g k k')).val * gs (𝟙 Y) y _ (s.f k')

theorem **loc_sv** (s : LB.Sec P) : IsLoc (nonzeroLocus s) ⊤ (sv γ s)

theorem **ratio_iff** (s t : LB.Sec P) (r : Γ(Y, nonzeroLocus s)) : (∀ k, (Y.presheaf.map (homOfLE (inf_le_right : nonzeroLocus s ⊓ P.U k ≤ P.U k)).op).hom (t.f k) = (Y.presheaf.map (homOfLE (inf_le_left : nonzeroLocus s ⊓ P.U k ≤ nonzeroLocus s)).op).hom r * (Y.presheaf.map (homOfLE (inf_le_right : nonzeroLocus s ⊓ P.U k ≤ P.U k)).op).hom (s.f k)) ↔ ∀ y ∈ nonzeroLocus s, gs (𝟙 Y) y _ r = (sv γ t * inv (sv γ s)) y

theorem **sc_sv** (s : LB.Sec P) : Sc U G ⊤ N (sv (fun y => κ (c y)) s)

theorem **exists_sv** (hP : P.IsLB) {σ : Fn Y} (hσ : Sc U G ⊤ N σ) : ∃ s : LB.Sec P, sv (fun y => κ (c y)) s = σ

theorem **Pack.mono** {ι : Type} {U : ι → Z.Opens} {G : ι → Fn Z} {W : Z.Opens} {S S' : Set (Fn Z)} (H : Pack U G W S) (hS : ∀ x ∈ S, x ∈ Subring.closure S') : Pack U G W S'

theorem **cst_mem** {B V : Scheme.{0}} [IsAffine B] (a : Γ(B, ⊤)) (b : V ⟶ B) [IsOpenImmersion b] (hb : b ''ᵁ ⊤ = B.basicOpen a) (q : Y ⟶ V) (h : Y ⟶ Z) (p : Z ⟶ B) (e : h ≫ p = q ≫ b) (c : Γ(V, ⊤)) : cst q c ∈ Subring.closure (pb h '' insert (inv (cst p a)) (Set.range (cst p)))

theorem **gm_const** (L : LB X) (j : Y ⟶ X) (N : ℕ) (y : Y) (i i' : L.ι) (hi : j y ∈ L.U i) (hi' : j y ∈ L.U i') : (gm (𝟙 Y) y _ ((LB.LinComb L (K

theorem **pack_of_relAmple** {S : Scheme.{0}} (A : Over S) (L : LB A.left) (U : S.Opens) (hU : IsAffineOpen U) (U₀ : S.Opens) (H : RelAmple (Over.mk (A.hom |_ U₀)) (L.pull (A.hom ⁻¹ᵁ U₀).ι)) (a : Γ(U.toScheme, ⊤)) (ha : U.toScheme.basicOpen a ≤ U.ι ⁻¹ᵁ U₀) (c : (A.hom ⁻¹ᵁ U).toScheme → L.ι) (hc : ∀ z, (A.hom ⁻¹ᵁ U).ι z ∈ L.U (c z)) : ∃ W, IsLoc W ⊤ (cst (A.hom |_ U) a) ∧ Pack (fun i => (A.hom ⁻¹ᵁ U).ι ⁻¹ᵁ L.U i) (fun i z => (gm (A.hom ⁻¹ᵁ U).ι z _ (L.g i (c z))).val) W (insert (inv (cst (A.hom |_ U) a)) (Set.range (cst (A.hom |_ U))))

theorem **relAmple_of_locally** : ∀ (S : Scheme.{0}) (A : Over S) (L : LB A.left), L.IsLB → QuasiCompact A.hom → QuasiSeparated A.hom → (∀ s : S, ∃ (U : S.Opens) (_ : s ∈ U), RelAmple (Over.mk (A.hom |_ U)) (L.pull (A.hom ⁻¹ᵁ U).ι)) → RelAmple A L

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.RelAmpleOfLocallyOnBaseStatement

theorem **_root_.GlobalTheory.T.relAmple_of_locally_on_base** : ∀ (S : Scheme.{0}) (A : Over S) (L : LB A.left), L.IsLB → QuasiCompact A.hom → QuasiSeparated A.hom → (∀ s : S, ∃ (U : S.Opens) (_ : s ∈ U), RelAmple (Over.mk (A.hom |_ U)) (L.pull (A.hom ⁻¹ᵁ U).ι)) → RelAmple A L

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.RingKrullDimLeOfQuasiFinite

theorem **exists_quotient** (s t : M.Sec) : ∃ r : Γ(Y, nonzeroLocus t), ∀ (y : Y) (k : M.ι), y ∈ nonzeroLocus t → y ∈ M.U k → gs (𝟙 Y) y _ (s.f k) = gs (𝟙 Y) y _ r ∗ gs (𝟙 Y) y _ (t.f k)

theorem **gs_sum_mul** {I : Type} [Fintype I] (c : I → Γ(Y, ⊤)) {U : Y.Opens} (b : I → Γ(Y, U)) (y : Y) (hy : y ∈ U) : gs (𝟙 Y) y U (∑ i, resG Y le_top (c i) ∗ b i) = ∑ i, gs (𝟙 Y) y ⊤ (c i) ∗ gs (𝟙 Y) y U (b i)

theorem **exists_sec_sum** {I : Type} [Fintype I] (c : I → Γ(Y, ⊤)) (s : I → M.Sec) : ∃ t : M.Sec, ∀ k, t.f k = ∑ i, resG Y le_top (c i) ∗ (s i).f k

theorem **isAffineOpen_nonzeroLocus_sum** (hM : M.IsLB) {I : Type} [Fintype I] (c : I → Γ(Y, ⊤)) (s : I → M.Sec) (hs : ∀ i, IsAffineOpen (nonzeroLocus (s i))) (t : M.Sec) (ht : ∀ k, t.f k = ∑ i, resG Y le_top (c i) ∗ (s i).f k) : IsAffineOpen (nonzeroLocus t)

theorem **exists_isUnit_sum** {K : Type} [Field K] [Infinite K] {I J : Type} [Fintype I] [Finite J] (R : J → Type) [∀ j, CommRing (R j)] [∀ j, IsLocalRing (R j)] (ψ : ∀ j, K →+* R j) (a : ∀ j, I → R j) (h : ∀ j, ∃ i, IsUnit (a j i)) : ∃ c : I → K, ∀ j, IsUnit (∑ i, ψ j (c i) ∗ a j i)

theorem **exists_isAffineOpen_of_finite** {K : Type} [Field K] [Infinite K] (φ : K →+* Γ(Y, ⊤)) (hM : M.IsLB) {I : Type} [Fintype I] (s : I → M.Sec) (hc : ⊔ i, nonzeroLocus (s i) = ⊤) (hs : ∀ i, IsAffineOpen (nonzeroLocus (s i))) (F : Set Y) (hF : F.Finite) : ∃ D : Y.Opens, IsAffineOpen D ∧ ∀ y ∈ F, y ∈ D

theorem **exists_affine_le_coheight** {X : Scheme.{0}} [NoetherianSpace X] (h : ∀ F : Set X, F.Finite → ∃ D : X.Opens, IsAffineOpen D ∧ ∀ y ∈ F, y ∈ D) (n : ℕ) : ∃ D : Fin n → X.Opens, (∀ i, IsAffineOpen (D i)) ∧ ∀ x : X, x ∉ ⊔ i, D i → (n : ℕ∞) ≤ Order.coheight x

theorem **ringKrullDim_le_of_quasiFinite** (R S : Type) [CommRing R] [CommRing S] [Algebra R S] [Algebra.QuasiFinite R S] : ringKrullDim S ≤ ringKrullDim R

theorem **coheight_le_of_smooth** {K : Type} [Field K] {X : Scheme.{0}} (f : X ⟶ Spec (CommRingCat.of K)) {g : ℕ} (hf : SmoothOfRelativeDimension g f) (x : X) : Order.coheight x ≤ g

theorem **exists_cover_of_infinite** (K : Type) [Field K] [Infinite K] (P : PolAV (Spec (CommRingCat.of K))) : ∃ D : Fin (P.g + 1) → P.A.left.Opens, (∀ i, IsAffineOpen (D i)) ∧ ⊔ i, D i = ⊤

theorem **exists_cover** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → (∀ (K : Type) [Field K] (X : Scheme.{0}) [IsIntegral X] (f : X ⟶ Spec (CommRingCat.of K)) (g : ℕ), SmoothOfRelativeDimension g f → ∀ (B : Type) [CommRing B] [Nontrivial B] [Algebra K B] (j : Spec (CommRingCat.of B) ⟶ X), IsOpenImmersion j → j ≫ f = Spec.map (CommRingCat.ofHom (algebraMap K B)) → Algebra.trdeg K B = g) → ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), ∃ D : Fin (P.g + 1) → P.A.left.Opens, (∀ i, IsAffineOpen (D i)) ∧ ⊔ i, D i = ⊤

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SecBaseChangeOfPerf

theorem **gs_sum** (r : Z ⟶ X) (z : Z) (U : X.Opens) (h : r z ∈ U) {I : Type} [Fintype I] (a : I → Γ(X, U)) : gs r z U (∑ i, a i) = ∑ i, gs r z U (a i)

theorem **sec_ext** {M : LB X} {s s' : LB.Sec M} (h : ∀ j, s.f j = s'.f j) : s = s'

theorem **gs_rp** (h : X' ⟶ X) {U : X.Opens} {V : X'.Opens} (le : V ≤ h ⁻¹ᵁ U) (r : Z ⟶ X') (z : Z) (hz : r z ∈ V) (a : Γ(X, U)) : gs r z V ((X'.presheaf.map (homOfLE le).op).hom ((h.app U).hom a)) = gs (r ≫ h) z U a

theorem **gs_secChart** {A' : LB X'} {A : LB X} {h : X' ⟶ X} (a : ChartPull A' A h) (s : LB.Sec A) (r : Z ⟶ X') (z : Z) (i : A'.ι) (hz : r z ∈ A'.U i) : gs r z (A'.U i) ((secChart a s).f i) = gs (r ≫ h) z (A.U (a.idx i)) (s.f (a.idx i))

theorem **mem_wchart_iff** (j : K → L.ι) (y : X') : y ∈ LB.wchart L ψ j ↔ π y ∈ LB.wchart L φ j

theorem **Rep.sec_eq** {c : R' →+* Γ(X', ⊤)} {v : Fin F.a → R'} {s s' : LB.Sec (LB.LinComb L ψ ε)} (h : Rep π ψ F c v s.f) (h' : Rep π ψ F c v s'.f) : s = s'

theorem **Rep.sum** {c : R' →+* Γ(X', ⊤)} {I : Type} [Fintype I] {w : I → Fin F.a → R'} {b : I → ∀ j, Γ(X', LB.wchart L ψ j)} (h : ∀ l, Rep π ψ F c (w l) (b l)) (a : I → R') : Rep π ψ F c (fun k => ∑ l, w l k ∗ a l) (fun j => ∑ l, rs X' le_top (c (a l)) ∗ b l j)

theorem **Rep.pull** {X'' : Scheme.{0}} {R'' : Type} [CommRing R''] {c : R' →+* Γ(X', ⊤)} {c' : R'' →+* Γ(X'', ⊤)} {v : Fin F.a → R'} {v' : Fin F.a → R''} {s : ∀ j, Γ(X', LB.wchart L ψ j)} (h : Rep π ψ F c v s) (ρ : X'' ⟶ X') {π' : X'' ⟶ X} (hπ : π' = ρ ≫ π) {ψ' : K → (X'' ⟶ Y)} (e' : ∀ k, ψ' k = ρ ≫ ψ k) (hv : ∀ k, c' (v' k) = ρ.appTop.hom (c (v k))) {s' : ∀ j, Γ(X'', LB.wchart L ψ' j)} (le : ∀ j, LB.wchart L ψ' j ≤ ρ ⁻¹ᵁ LB.wchart L ψ j) (hs : ∀ j, s' j = (X''.presheaf.map (homOfLE (le j)).op).hom ((ρ.app _).hom (s j))) : Rep π' ψ' F c' v' s'

theorem **rep_main** (g : R →+* R') (f' : X' ⟶ Spec (CommRingCat.of R')) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom g))) (c : R' →+* Γ(X', ⊤)) (hc : ∀ x, c x = f'.appTop.hom ((Scheme.ΓSpecIso (CommRingCat.of R')).inv.hom x)) : (∀ v, (F.D₀.map g).mulVec v = 0 → ∃ s : LB.Sec (LB.LinComb L ψ ε), Rep π ψ F c v s.f) ∧ (∀ s : LB.Sec (LB.LinComb L ψ ε), ∃ v, (F.D₀.map g).mulVec v = 0 ∧ Rep π ψ F c v s.f) ∧ (∀ v v' (s : LB.Sec (LB.LinComb L ψ ε)), (F.D₀.map g).mulVec v = 0 → (F.D₀.map g).mulVec v' = 0 → Rep π ψ F c v s.f → Rep π ψ F c v' s.f → v = v')

theorem **snd_eq** (h : T' ⟶ T) : (snd T' A).left = (h ▷ A).left ≫ (snd T A).left

theorem **fst_eq** (h : T' ⟶ T) : (h ▷ A).left ≫ (fst T A).left = (fst T' A).left ≫ h.left

theorem **isPullback_snd** (T : Over S) : IsPullback (snd T A).left (fst T A).left A.hom T.hom

theorem **isPullback_whisker** (h : T' ⟶ T) : IsPullback (h ▷ A).left (fst T' A).left (fst T A).left h.left

theorem **isPullback_affine** [IsAffine T.left] (h : T ⟶ Over.mk t₀) : IsPullback (h ▷ A).left ((fst T A).left ≫ T.left.toSpecΓ) (fst (Over.mk t₀) A).left (Spec.map (CommRingCat.ofHom (cfun h)))

theorem **exists_pullSec** (h : T' ⟶ T) (s : LB.Sec (LPow A L n T)) : ∃ s' : LB.Sec (LPow A L n T'), IsPullSec A L n (h ▷ A) (whiskerRight_snd h A) s s'

theorem **RepAt.pull** {h : T ⟶ Over.mk t₀} {v : Fin F.a → R} {s : LB.Sec (LPow A L n T)} (H : RepAt F h (fun k => cfun h (v k)) s) (h' : T' ⟶ T) {s' : LB.Sec (LPow A L n T')} (hs : IsPullSec A L n (h' ▷ A) (whiskerRight_snd h' A) s s') : RepAt F (h' ≫ h) (fun k => cfun (h' ≫ h) (v k)) s'

theorem **repAt_main** [IsAffine T.left] (h : T ⟶ Over.mk t₀) : (∀ v, (F.D₀.map (cfun h)).mulVec v = 0 → ∃ s, RepAt F h v s) ∧ (∀ s, ∃ v, (F.D₀.map (cfun h)).mulVec v = 0 ∧ RepAt F h v s) ∧ (∀ v v' s, (F.D₀.map (cfun h)).mulVec v = 0 → (F.D₀.map (cfun h)).mulVec v' = 0 → RepAt F h v s → RepAt F h v' s → v = v')

theorem **basis_of_rep** [IsAffine T.left] (h : T ⟶ Over.mk t₀) {m : ℕ} (B : Matrix (Fin F.a) (Fin m) R) (hB : F.D₀ * B = 0) (hu : ∀ v : Fin F.a → Γ(T.left, ⊤), (F.D₀.map (cfun h)).mulVec v = 0 → ∃! z : Fin m → Γ(T.left, ⊤), (B.map (cfun h)).mulVec z = v) (b : Fin m → LB.Sec (LPow A L n T)) (hb : ∀ i, RepAt F h (fun k => cfun h (B k i)) (b i)) : LinearIndependent Γ(T.left, ⊤) (fun i => (SecM.ofSec A (b i) : SecM A L n T)) ∧ Submodule.span Γ(T.left, ⊤) (Set.range fun i => (SecM.ofSec A (b i) : SecM A L n T)) = ⊤

theorem **exists_h1Zero** (hV : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ), 1 ≤ n → FirstCohomologyVanishes (pw P.L n)) (hI : ∀ (X X' : Scheme.{0}) (e : X' ⟶ X) (M : LB X) (M' : LB X'), IsIso e → M.IsLB → M'.IsLB → (∃ w : LB.Iso M' (M.pull e), w.IsGen) → FirstCohomologyVanishes M → FirstCohomologyVanishes M') (hP : PolBaseChangeSt) {S : Scheme.{0}} (P : PolAV S) (n : ℕ) (hn : 1 ≤ n) (T : Over S) (K : Type) [Field K] [IsAlgClosed K] (q : Spec (CommRingCat.of K) ⟶ T.left) : ∃ (X' : Scheme.{0}) (π : X' ⟶ (T ⊗ P.A).left) (f' : X' ⟶ Spec (CommRingCat.of K)), IsPullback π f' (fst T P.A).left q ∧ FirstCohomologyVanishes ((LPow P.A P.L n T).pull π)

theorem **exists_geomPoint** {R : Type} [CommRing R] (q : PrimeSpectrum R) : ∃ (K : Type) (_ : Field K) (_ : IsAlgClosed K) (g : R →+* K), ∀ r ∈ q.asIdeal, g r = 0

theorem **secBaseChange_of_perf** : (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)) → (∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) → (∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (n : ℕ), 1 ≤ n → FirstCohomologyVanishes (pw P.L n)) → (∀ (X X' : Scheme.{0}) (e : X' ⟶ X) (M : LB X) (M' : LB X'), IsIso e → M.IsLB → M'.IsLB → (∃ w : LB.Iso M' (M.pull e), w.IsGen) → FirstCohomologyVanishes M → FirstCohomologyVanishes M') → (∀ (R K : Type) [CommRing R] [Field K] (g₀ : R →+* K) (a b c : ℕ) (D₀

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SecDivOfThickenings

theorem **gs_zero** {X Z : Scheme.{0}} (q : Z ⟶ X) (z : Z) (U : X.Opens) : gs q z U 0 = 0

theorem **cst_comp_eq_zero** {R R' : Type} [CommRing R] [CommRing R'] (φ : R →+* R') {r : R} (hr : φ r = 0) {Z : Scheme.{0}} (q : Z ⟶ Spec (CommRingCat.of R')) (z : Z) : cst (q ≫ Spec.map (CommRingCat.ofHom φ)) z r = 0

theorem **app_comp_eq_zero** {R : Type} [CommRing R] {X Y Z : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of R)) (a : Y ⟶ X) (b : Z ⟶ Y) (r : R) (hr : ∀ z, cst (b ≫ a ≫ p) z r = 0) (U : X.Opens) (f : Γ(X, U)) (t : Γ(Y, a ⁻¹ᵁ U)) (h : (a.app U).hom f = resTop _ (scal (a ≫ p) r) * t) : ((b ≫ a).app U).hom f = 0

theorem **constFn_dvd_of_mem_ker** {R : Type} [CommRing R] {X Z : Scheme.{0}} (p : X ⟶ Spec (CommRingCat.of R)) (r : R) (i : Z ⟶ X) (l : pullback p (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {r})))) ⟶ Z) (hl : l ≫ i = pullback.fst _ _) {W : X.Opens} (hW : IsAffineOpen W) {a : Γ(X, W)} (ha : a ∈ i.ker.ideal (W, hW)) : constFn p W r ∣ a

theorem **map_le_jacobson** {R R' : Type} [CommRing R] [CommRing R'] (φ : R →+* R') (hφ : Function.Surjective φ) {I : Ideal R} (hI : I ≤ Ideal.jacobson ⊥) : I.map φ ≤ Ideal.jacobson ⊥

theorem **sec_div_of_thickenings** : ∀ (R : Type) [CommRing R] [IsDomain R] [IsNoetherianRing R] (I : Ideal R), I ≤ Ideal.jacobson ⊥ → ∀ (H : Over (Spec (CommRingCat.of R))) [GrpObj H] [Smooth H.hom] [GeometricallyConnected H.hom] (N : LB H.left), N.IsLB → ∀ (s : LB.Sec N) (r : R), r ≠ 0 → (∀ n : ℕ, ∃ t : ∀ j, Γ((thickObj R I n H).left, (fstG I n H).left ⁻¹ᵁ N.U j), ∀ j, ((fstG I n H).left.app _).hom (s.f j) = resTop (Y

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SecEqZeroAndHomEqOfThickenings

theorem **exists_mem_chart** (hN : N.IsLB) (x : X) : ∃ i, x ∈ N.U i

theorem **associated_germ_sec** (s : N.Sec) (i j : N.ι) (x : X) (hi : x ∈ N.U i) (hj : x ∈ N.U j) : Associated ((X.presheaf.germ _ x hi).hom (s.f i)) ((X.presheaf.germ _ x hj).hom (s.f j))

theorem **secIdeal_le_ker** (hN : N.IsLB) (s : N.Sec) (φ : Y ⟶ X) [QuasiCompact φ] (h : ∀ j, (φ.app _).hom (s.f j) = 0) : secIdeal hN s ≤ φ.ker

theorem **eq_zero_of_secIdeal_eq_bot** (hN : N.IsLB) (s : N.Sec) (h : secIdeal hN s = ⊥) (j : N.ι) : s.f j = 0

theorem **sec_eq_zero_and_hom_eq_of_thickenings** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (I : Ideal R), I ≤ Ideal.jacobson ⊥ → ∀ (H : Over (Spec (CommRingCat.of R))) [GrpObj H] [Smooth H.hom] [GeometricallyConnected H.hom], (∀ (N : LB H.left), N.IsLB → ∀ s : LB.Sec N, (∀ (n : ℕ) j, ((fstG I n H).left.app _).hom (s.f j) = 0) → ∀ j, s.f j = 0) ∧ (∀ (Y : Over (Spec (CommRingCat.of R))) [IsSeparated Y.hom] (f g : H ⟶ Y), (∀ n : ℕ, (fstG I n H).left ≫ f.left = (fstG I n H).left ≫ g.left) → f = g)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SecEqZeroOfDvdPowUniformiser

theorem **sec_eq_zero_of_chart** {X : Scheme.{0}} [IsIntegral X] (N : LB X) (s : LB.Sec N) (j : N.ι) (x : X) (hx : x ∈ N.U j) (hj : s.f j = 0) (i : N.ι) : s.f i = 0

theorem **eq_zero_of_mem_pow_span** {X : Scheme.{0}} [IsIntegral X] [IsLocallyNoetherian X] (U : X.Opens) (x : X) (hx : x ∈ U) (a b : Γ(X, U)) (ha : x ∉ X.basicOpen a) (hb : ∀ e : ℕ, b ∈ Ideal.span {a} ^ e) : b = 0

theorem **sec_eq_zero_of_dvd_pow_uniformiser** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → Semi.lemmaBSt.{0}

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SecMInvariantEqBotOrUnivAtGeometricPoint

theorem **exists_sec_transfer** {M : LB Y} {N : LB Y'} (hM : M.IsLB) (hN : N.IsLB) (k : Y ⟶ Y') (c : ∀ (Z : Scheme.{0}) (_ : Z ⟶ Y) (z : Z), M.ι → N.ι → (Z.presheaf.stalk z)ˣ) (hc : ∀ p q, ∃ u : Γ(Y, M.U p ⊓ k ⁻¹ᵁ N.U q)ˣ, ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z), r z ∈ M.U p → k (r z) ∈ N.U q → gm r z _ u = c Z r z p q) (htr : ∀ (y : Y) p p' q q', y ∈ M.U p → y ∈ M.U p' → k y ∈ N.U q → k y ∈ N.U q' → c Y (𝟙 Y) y p q = gm (𝟙 Y) y (M.U p ⊓ M.U p') (M.g p p') * (gm (𝟙 Y ≫ k) y (N.U q ⊓ N.U q') (N.g q q'))⁻¹ * c Y (𝟙 Y) y p' q') (s : LB.Sec N) : ∃ s' : LB.Sec M, ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) p q, r z ∈ M.U p → k (r z) ∈ N.U q → gs r z _ (s'.f p) = (c Z r z p q).val * gs (r ≫ k) z _ (s.f q)

theorem **pw_comp** {Z : Scheme.{0}} (w : LB.Iso L' (L.pull F)) (φ' : K → (Y' ⟶ X')) (ε : K → ℤ) (k : Y ⟶ Y') (r : Z ⟶ Y) (z : Z) (p : K → L.ι) (q : K → L'.ι) : pw w (fun i => k ≫ φ' i) ε r z p q = pw w φ' ε (r ≫ k) z p q

theorem **trF_comp** {Z : Scheme.{0}} (φ' : K → (Y' ⟶ X')) (ε : K → ℤ) (k : Y ⟶ Y') (r : Z ⟶ Y) (z : Z) (q q' : K → L'.ι) : trF (fun i => k ≫ φ' i) ε r z q q' = trF φ' ε (r ≫ k) z q q'

theorem **mem_wchart_comp** (φ' : K → (Y' ⟶ X')) (k : Y ⟶ Y') (y : Y) (q : K → L'.ι) : y ∈ LB.wchart L' (fun i => k ≫ φ' i) q ↔ k y ∈ LB.wchart L' φ' q

theorem **exists_sec_up** {w : LB.Iso L' (L.pull F)} (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) {φ' : K → (Y' ⟶ X')} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (k : Y ⟶ Y') (hk : ∀ i, φ i = (k ≫ φ' i) ≫ F) (s' : LB.Sec (LB.LinComb L' φ' ε)) : ∃ s : LB.Sec (LB.LinComb L φ ε), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) p q, r z ∈ LB.wchart L φ p → k (r z) ∈ LB.wchart L' φ' q → gs r z (LB.wchart L φ p) (s.f p) = (pw w φ' ε (r ≫ k) z p q).val * gs (r ≫ k) z (LB.wchart L' φ' q) (s'.f q)

theorem **exists_sec_down** {w : LB.Iso L' (L.pull F)} (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) {φ' : K → (Y' ⟶ X')} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (h : Y' ⟶ Y) (hh : ∀ i, φ' i ≫ F = h ≫ φ i) (s : LB.Sec (LB.LinComb L φ ε)) : ∃ s' : LB.Sec (LB.LinComb L' φ' ε), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) p q, h (r z) ∈ LB.wchart L φ p → r z ∈ LB.wchart L' φ' q → gs (r ≫ h) z (LB.wchart L φ p) (s.f p) = (pw w φ' ε r z p q).val * gs r z (LB.wchart L' φ' q) (s'.f q)

theorem **exists_gen_up** {w : LB.Iso L' (L.pull F)} (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) {φ' : K → (Y' ⟶ X')} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (k : Y ⟶ Y') (hk : ∀ i, φ i = k ≫ φ' i ≫ F) (a : LB.Gen (LB.LinComb L' φ' ε)) (ha : a.IsGen) : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) p q, r z ∈ LB.wchart L φ p → k (r z) ∈ LB.wchart L' φ' q → gm r z (LB.wchart L φ p) (t.s p) = pw w φ' ε (r ≫ k) z p q * gm (r ≫ k) z (LB.wchart L' φ' q) (a.s q)

theorem **PullRel.unique_left** (hL' : L'.IsLB) {k : Y ⟶ Y'} (hk : k ≫ h = 𝟙 Y) {f₁ f₂ f'} (h₁ : PullRel w φ' φ ε h f₁ f') (h₂ : PullRel w φ' φ ε h f₂ f') : f₁ = f₂

theorem **PullRel.unique_right** (hL : L.IsLB) {f f₁ f₂} (h₁ : PullRel w φ' φ ε h f f₁) (h₂ : PullRel w φ' φ ε h f f₂) : f₁ = f₂

theorem **PullRel.add** {f₁ f₂ f₁' f₂'} (h₁ : PullRel w φ' φ ε h f₁ f₁') (h₂ : PullRel w φ' φ ε h f₂ f₂') : PullRel w φ' φ ε h (fun p => f₁ p + f₂ p) (fun q => f₁' q + f₂' q)

theorem **PullRel.smul** {f f'} (H : PullRel w φ' φ ε h f f') (c : Γ(Y, ⊤)) (c' : Γ(Y', ⊤)) (hc : h.appTop.hom c = c') : PullRel w φ' φ ε h (fun p => (Y.presheaf.map (homOfLE le_top).op).hom c * f p) (fun q => (Y'.presheaf.map (homOfLE le_top).op).hom c' * f' q)

theorem **exists_pullRel_up** (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) {k : Y ⟶ Y'} (hk : k ≫ h = 𝟙 Y) (hk' : h ≫ k = 𝟙 Y') (hh : ∀ i, φ' i ≫ F = h ≫ φ i) (s' : LB.Sec (LB.LinComb L' φ' ε)) : ∃ s : LB.Sec (LB.LinComb L φ ε), PullRel w φ' φ ε h s.f s'.f

theorem **isActOf_iff** (θ : ThetaGrp A L n T) (s s' : LB.Sec (LPow A L n T)) : (IsActOf A L n θ s s' → ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (j : Fin 2 → L.ι), r z ∈ LB.wchart L ![(snd T A).left, (translPt A θ.pt).left] j → gs r z (LB.wchart L (fun _ : Fin 1 => (snd T A).left) (fun _ => j 0)) (s'.f (fun _ => j 0)) = (gm r z (LB.wchart L ![(snd T A).left, (translPt A θ.pt).left] j) (θ.φ.s j)).val * gs (r ≫ (translT A θ.pt).left) z (LB.wchart L (fun _ : Fin 1 => (snd T A).left) (fun _ => j 1)) (s.f (fun _ => j 1))) ∧ ((∀ (z : (T ⊗ A).left) (j : Fin 2 → L.ι), z ∈ LB.wchart L ![(snd T A).left, (translPt A θ.pt).left] j → gs (𝟙 _) z (LB.wchart L (fun _ : Fin 1 => (snd T A).left) (fun _ => j 0)) (s'.f (fun _ => j 0)) = (gm (𝟙 _) z (LB.wchart L ![(snd T A).left, (translPt A θ.pt).left] j) (θ.φ.s j)).val * gs (𝟙 _ ≫ (translT A θ.pt).left) z (LB.wchart L (fun _ : Fin 1 => (snd T A).left)

theorem **exists_secEquiv** (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) (hk : k ≫ h = 𝟙 _) (hk' : h ≫ k = 𝟙 _) (hs : (snd T' A').left ≫ g = h ≫ (snd T A).left) (σ : Γ(T'.left, ⊤) → Γ(T.left, ⊤)) (hσ : ∀ c, h.appTop.hom ((fst T A).left.appTop.hom (σ c)) = (fst T' A').left.appTop.hom c) (n : ℤ) : ∃ E : SecM A' L' n T' ≃+ SecM A L n T, (∀ c x, E (c • x) = σ c • E x) ∧ ∀ x, PullRel w (fun _ : Fin 1 => (snd T' A').left) (fun _ => (snd T A).left) (fun _ => n) h (E x).1 x.1

theorem **exists_theta_up** (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) (hk : k ≫ h = 𝟙 _) (hs : (snd T' A').left ≫ g = h ≫ (snd T A).left) {n : ℤ} (θ' : ThetaGrp A' L' n T') (x : T ⟶ A) (hx : (translPt A' θ'.pt).left ≫ g = h ≫ (translPt A x).left) (hx' : (translT A' θ'.pt).left ≫ h = h ≫ (translT A x).left) : ∃ θ : ThetaGrp A L n T, ∀ (s₁ s₂ : LB.Sec (LPow A L n T)) (s₁' s₂' : LB.Sec (LPow A' L' n T')), PullRel w (fun _ : Fin 1 => (snd T' A').left) (fun _ => (snd T A).left) (fun _ => n) h s₁.f s₁'.f → PullRel w (fun _ : Fin 1 => (snd T' A').left) (fun _ => (snd T A).left) (fun _ => n) h s₂.f s₂'.f → IsActOf A' L' n θ' s₁' s₂' → IsActOf A L n θ s₁ s₂

theorem **exists_totalIso** (hg : IsPullback g A'.hom A.hom f) (hmul : ∀ (T : Over S') (x y : T ⟶ A'), toBase hg.w (x * y) = toBase hg.w x * toBase hg.w y) : ∃ (h : (baseOver S' ⊗ A').left ⟶ (Over.mk f ⊗ A).left) (k : (Over.mk f ⊗ A).left ⟶ (baseOver S' ⊗ A').left), k ≫ h = 𝟙 _ ∧ h ≫ k = 𝟙 _ ∧ (snd (baseOver S') A').left ≫ g = h ≫ (snd (Over.mk f) A).left ∧ h ≫ (fst (Over.mk f) A).left = (fst (baseOver S') A').left ∧ ∀ x' : baseOver S' ⟶ A', ∃ x : Over.mk f ⟶ A, (translPt A' x').left ≫ g = h ≫ (translPt A x).left ∧ (translT A' x').left ≫ h = h ≫ (translT A x).left

theorem **exists_baseChange_secEquiv** (hBC : PolBaseChangeSt) {S S' : Scheme.{0}} (P : PolAV S) (f : S' ⟶ S) : ∃ P' : PolAV S', P'.g = P.g ∧ ∀ n : ℤ, ∃ (E : SecM P'.A P'.L n (baseOver S') ≃+ SecM P.A P.L n (Over.mk f)) (σ : Γ((baseOver S').left, ⊤) → Γ((Over.mk f).left, ⊤)), Function.Bijective σ ∧ (∀ c x, E (c • x) = σ c • E x) ∧ ∀ θ' : ThetaGrp P'.A P'.L n (baseOver S'), ∃ θ : ThetaGrp P.A P.L n (Over.mk f), ∀ s₁' s₂' : LB.Sec (LPow P'.A P'.L n (baseOver S')), IsActOf P'.A P'.L n θ' s₁' s₂' → IsActOf P.A P.L n θ (SecM.toSec P.A (E (SecM.ofSec P'.A s₁'))) (SecM.toSec P.A (E (SecM.ofSec P'.A s₂')))

theorem **secM_invariant_eq_bot_or_univ_at_geometric_point** : PolBaseChangeSt → (GlobalSectionsRankAlgClosed → (∀ (S : Scheme.{0}) (P : PolAV S) (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S) (n : ℕ), 1 ≤ n → Module.finrank Γ((Over.mk t).left, ⊤) (SecM P.A P.L n (Over.mk t)) = n ^ P.g)) ∧ (IrredSt → (∀ (S : Scheme.{0}) (P : PolAV S) (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S) (k : ℕ), (k = 1 ∨ 4 ≤ k) → (k : K) ≠ 0 → ∀ W : Submodule Γ((Over.mk t).left, ⊤) (SecM P.A P.L k (Over.mk t)), (∀ (θ : ThetaGrp P.A P.L k (Over.mk t)) (s s' : LB.Sec (LPow P.A P.L k (Over.mk t))), IsActOf P.A P.L k θ s s' → SecM.ofSec P.A s ∈ W → SecM.ofSec P.A s' ∈ W) → W = ⊥ ∨ W = ⊤))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SerreVanishingVeryAmple

theorem **serre_h1Zero_and_sections_polynomial** : (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB X) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∀ M : LB X, M.IsLB → ∃ t₀ : ℤ, ∀ t, t₀ ≤ t → FirstCohomologyVanishes (tw M H t)) ∧ (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB X) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∃ t₀ : ℕ, ∀ t : ℕ, t₀ ≤ t → ∀ σ : LB.Sec (pw H t), ∃ F : MvPolynomial (Fin (m + 1)) R, F.IsHomogeneous t ∧ ∀ k : H.ι, σ.f k = MvPolynomial.eval₂Hom (cst f (H.U k)) (fun j => (s j).f k) F)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SerreVanishingVeryAmpleStatement

theorem **_root_.GlobalTheory.T.serre_vanishing_very_ample** : (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB X) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∀ M : LB X, M.IsLB → ∃ t₀ : ℤ, ∀ t, t₀ ≤ t → FirstCohomologyVanishes (tw M H t)) ∧ (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R))

[IsNoetherianRing R] (H : LB X) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∃ t₀ : ℕ, ∀ t : ℕ, t₀ ≤ t → ∀ σ : LB.Sec (pw H t), ∃ F : MvPolynomial (Fin (m + 1)) R, F.IsHomogeneous t ∧ ∀ k : H.ι, σ.f k = MvPolynomial.eval₂Hom (cst f (H.U k)) (fun j => (s j).f k) F)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SmoothDense

theorem **smooth_smoothLocus_ι_comp** {X Y : Scheme.{0}} (f : X ⟶ Y) [LocallyOfFinitePresentation f] : Smooth (f.smoothLocus.ι ≫ f)

theorem **smoothDense** : SmoothDenseSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SpanEqTopOfFunctionals

theorem **wchart_le_of_eq** {φ : K → (Y ⟶ X)} {φ' : K → (Y' ⟶ X)} (h : Y' ⟶ Y) (e : ∀ k, h ≫ φ k = φ' k) (j : K → L.ι) : LB.wchart L φ' j ≤ h ⁻¹ᵁ LB.wchart L φ j

theorem **sval_pullTo** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K → (Y' ⟶ X)} (s : LB.Sec (LB.LinComb L φ ε)) (h : Y' ⟶ Y) (e : ∀ k, h ≫ φ k = φ' k) (r : Z ⟶ Y') (z : Z) {J : (Z ⟶ X) → L.ι} (hJ : Good L z J) : sval (pullTo s h e) r z J = sval s (r ≫ h) z J

theorem **exists_gen_pullTo** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K → (Y' ⟶ X)} (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) (h : Y' ⟶ Y) (e : ∀ k, h ≫ φ k = φ' k) : ∃ t' : LB.Gen (LB.LinComb L φ' ε), t'.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) (J), Good L z J → val t' r z J = val t (r ≫ h) z J

theorem **whisker_snd_left** {T T' : Over S} (h : T' ⟶ T) : (h ▷ A).left ≫ (snd T A).left = (snd T' A).left

theorem **isPullSec_pullS** (a : ℤ) {T T' : Over S} (h : T' ⟶ T) (s : LB.Sec (LPow A L a T)) : IsPullSec A L a (h ▷ A) (whiskerRight_snd h A) s (pullS A a h s)

theorem **sv_pullS** (a : ℤ) {T T' : Over S} (h : T' ⟶ T) (s : LB.Sec (LPow A L a T)) {Z : Scheme.{0}} (r : Z ⟶ (T' ⊗ A).left) (z : Z) {J : (Z ⟶ A.left) → L.ι} (hJ : Good L z J) : sv A (pullS A a h s) r z J = sv A s (r ≫ (h ▷ A).left) z J

theorem **ofSec_ext** (hL : L.IsLB) {a : ℤ} {T : Over S} {s s' : LB.Sec (LPow A L a T)} (h : ∀ y J, Good L y J → sv A s (𝟙 _) y J = sv A s' (𝟙 _) y J) : (SecM.ofSec A s : SecM A L a T) = SecM.ofSec A s'

theorem **pullM_ofSec** (a : ℤ) {T T' : Over S} (h : T' ⟶ T) (s : LB.Sec (LPow A L a T)) : pullM A L a h (SecM.ofSec A s) = SecM.ofSec A (pullS A a h s)

theorem **pullM_comp** (hL : L.IsLB) (a : ℤ) {T T' T'' : Over S} (h' : T'' ⟶ T') (h : T' ⟶ T) (f : SecM A L a T) : pullM A L a h' (pullM A L a h f) = pullM A L a (h' ≫ h) f

theorem **pullM_id** (hL : L.IsLB) (a : ℤ) {T : Over S} (f : SecM A L a T) : pullM A L a (𝟙 T) f = f

theorem **sv_of_isTwistPull** {s : LB.Sec (LPow A L b T)} {s' : LB.Sec (LPow A L a T)} (H : IsTwistPull A L ψ m hm a b α s s') {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J : (Z ⟶ A.left) → L.ι} (hJ : Good L z J) : sv A s' r z J = (val α r z J).val ∗ sv A s (r ≫ ψ.left) z J

theorem **isTwistPull_of_sv** (hL : L.IsLB) (hα : α.IsGen) {s : LB.Sec (LPow A L b T)} {s' : LB.Sec (LPow A L a T)} (H : ∀ y J, Good L y J → sv A s' (𝟙 _) y J = (val α (𝟙 _) y J).val ∗ sv A s (𝟙 _ ≫ ψ.left) y J) : IsTwistPull A L ψ m hm a b α s s'

theorem **exists_isTwistPull** (hL : L.IsLB) (hα : α.IsGen) (s : LB.Sec (LPow A L b T)) : ∃ s' : LB.Sec (LPow A L a T), IsTwistPull A L ψ m hm a b α s s'

theorem **exists_twist_pull** (hL : L.IsLB) (hα : α.IsGen) {T' : Over S} (h : T' ⟶ T) {ψ' : T' ⊗ A ⟶ T' ⊗ A} {m' : T' ⊗ A ⟶ A} (hm' : ψ' ≫ snd T' A = m') (e : ψ' ≫ h ▷ A = h ▷ A ≫ ψ) : ∃ α' : LB.Gen (LB.LinComb L (K

theorem **exists_theta_pull** (hL : L.IsLB) {k : ℤ} {T T' : Over S} (h : T' ⟶ T) (θ : ThetaGrp A L k T) : ∃ θ' : ThetaGrp A L k T', θ'.pt = h ≫ θ.pt ∧ ∀ s s', IsActOf A L k θ s s' → IsActOf A L k θ' (pullS A k h s) (pullS A k h s')

theorem **mulT_comm** {T T' : Over S} (h : T' ⟶ T) (n : ℕ) : mulT A T' n ≫ h ▷ A = h ▷ A ≫ mulT A T n

theorem **isVVGen_pull** (P : PolAV S) (N n : ℕ) {T T' : Over S} (h : T' ⟶ T) {t : LB.Sec (LPow P.A P.L N T)} (ht : IsVVGen P N n t) : IsVVGen P N n (pullS P.A N h t)

theorem **inVV_of_isVVGen** (P : PolAV S) (N n : ℕ) {T : Over S} {t : LB.Sec (LPow P.A P.L N T)} (ht : IsVVGen P N n t) : InVV P N n t

theorem **span_eq_top_of_functionals** {R M I κ : Type} [CommRing R] [AddCommGroup M] [Module R M] [Finite κ] (b : Module.Basis κ R M) (t : I → M) (H : ∀ m : Ideal R, m.IsMaximal → ∀ l : M →ₗ[R] R, (∀ c, l (t c) ∈ m) → ∀ x, l x ∈ m) : Submodule.span R (Set.range t) = ⊤

theorem **functional_mem_of_span** {R R' M M' I κ : Type} [CommRing R] [CommRing R'] [AddCommGroup M] [Module R M] [AddCommGroup M'] [Module R' M'] [Finite κ] (σ : R →+* R') (φ : M →ₛₗ[σ] M') (b : Module.Basis κ R M) (b' : Module.Basis κ R' M') (hb : ∀ i, b' i = φ (b i)) (t : I → M) (ht : Submodule.span R' (Set.range fun c => φ (t c)) = ⊤) (l : M →ₗ[R] R) (hl : ∀ c, σ (l (t c)) = 0) (x : M) : σ (l x) = 0

theorem **exists_basis** (P : PolAV S) (n : ℕ) {V : S.Opens} {κ : Type} [Fintype κ] (b : κ → LB.Sec (LPow P.A P.L n (overRes (T

theorem **surjective_mulA** (P : PolAV S) (n : ℕ) (hn : n ≠ 0) (α : LB.Gen (LB.LinComb P.L (K

theorem **surjective_mulT** (n : ℕ) (h : Surjective (mulA A n).left) (T : Over S) : Surjective (mulT A T n).left

theorem **ofSec_eq_zero_of_twist** (hL : L.IsLB) {T : Over S} [IsReduced (T ⊗ A).left] {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} {hm : ψ ≫ snd T A = m} {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **comp_linComb** {A : Over S} [GrpObj A] {T T' : Over S} (h : T' ⟶ T) {n : ℕ} (y : Fin n → (T ⟶ A)) (a : Fin n → ℕ) : h ≫ linComb y a = linComb (fun i => h ≫ y i) a

theorem **linComb_pow** {A : Over S} [GrpObj A] [IsCommMonObj A] {T : Over S} {n : ℕ} (y : Fin n → (T ⟶ A)) (a : Fin n → ℕ) (N : ℕ) (hy : ∀ i, y i ^ N = 1) : linComb y a ^ N = 1

theorem **span_translates_eq_top** {k : ℤ} {T : Over S} {I : Type} [Nonempty I] (hirr : ∀ W : Submodule Γ(T.left, ⊤) (SecM A L k T), (∀ (θ : ThetaGrp A L k T) (s s' : LB.Sec (LPow A L k T)), IsActOf A L k θ s s' → SecM.ofSec A s ∈ W → SecM.ofSec A s' ∈ W) → W = ⊥ ∨ W = ⊤) (ρ : ThetaGrp A L k T → (SecM A L k T ≃ₗ[Γ(T.left, ⊤)] SecM A L k T)) (h₁ : ∀ θ s s', IsActOf A L k θ s s' ↔ ρ θ (SecM.ofSec A s) = SecM.ofSec A s') (h₃ : ∀ θ θ', θ.pt = θ'.pt → ∃ c : Γ(T.left, ⊤)ˣ, ∀ v, ρ θ' v = (c : Γ(T.left, ⊤)) • ρ θ v) (h₄ : ∀ θx θy, ∃ θ, θ.pt = θx.pt ∗ θy.pt ∧ ∀ v, ρ θ v = ρ θx (ρ θy v)) (U : I → ThetaGrp A L k T) (hU : ∀ θ : ThetaGrp A L k T, ∃ c, (U c).pt = θ.pt) (v : SecM A L k T) (hv : v ≠ 0) : Submodule.span Γ(T.left, ⊤) (Set.range fun c => ρ (U c) v) = ⊤

theorem **exists_geomPoint** (X : Scheme.{0}) [IsAffine X] (m : Ideal Γ(X, ⊤)) (hm : m.IsMaximal) : ∃ (K : Type) (_ : Field K) (_ : IsAlgClosed K) (u : Spec (CommRingCat.of K) ⟶ X), ∀ r, u.appTop.hom r = 0 ↔ r ∈ m

theorem **natCast_ne_zero** {K : Type} [Field K] (t : Spec (CommRingCat.of K) ⟶ S) {m : ℕ} (hm : IsUnit (m : Γ(S, ⊤))) : (m : K) ≠ 0

theorem **vvRank_of_theta** : SecBaseChangeAnySt → (∀ (S : Scheme.{0}) (P : PolAV S) (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S) (k : ℕ), (k = 1 ∨ 4 ≤ k) → (k : K) ≠ 0 → ∀ W : Submodule Γ((Over.mk t).left, ⊤) (SecM P.A P.L k (Over.mk t)),

(∀ (θ : ThetaGrp P.A P.L k (Over.mk t)) (s s' : LB.Sec (LPow P.A P.L k (Over.mk t))), IsActOf P.A P.L k θ s s' → SecM.ofSec P.A s ∈ W → SecM.ofSec P.A s' ∈ W) → W = ⊥ ∨ W = ⊤) → (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

theorem **vvRank** : SecBaseChangeAnySt → (∀ (S : Scheme.{0}) (P : PolAV S) (K : Type) [Field K] [IsAlgClosed K] (t : Spec (CommRingCat.of K) ⟶ S) (k : ℕ), (k = 1 ∨ 4 ≤ k) → (k : K) ≠ 0 → ∀ W : Submodule Γ((Over.mk t).left, ⊤) (SecM P.A P.L k (Over.mk t)), (∀ (θ : ThetaGrp P.A P.L k (Over.mk t)) (s s' : LB.Sec (LPow P.A P.L k (Over.mk t))), IsActOf P.A P.L k θ s s' → SecM.ofSec P.A s ∈ W → SecM.ofSec P.A s' ∈ W) → W = ⊥ ∨ W = ⊤) → (∀ (S : Scheme.{0}) (P : PolAV S) (n : ℕ), ∃ α : LB.Gen (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SpecHomCompCoaction

lemma **specHom_left** (φ : S →ₐ[R] T) : (specHom φ).left = Spec.map (CommRingCat.ofHom φ.toRingHom)

lemma **specHom_comp** (φ : S →ₐ[R] T) (ψ : T →ₐ[R] T') : specHom (ψ.comp φ) = specHom ψ ≫ specHom φ

lemma **specHom_injective** : Function.Injective (specHom : (S →ₐ[R] T) → _)

lemma **specHom_lift_comul** {T : Type} [CommRing T] [Algebra R T] (φ ψ : Λ →ₐ[R] T) : specHom ((Algebra.TensorProduct.lift φ ψ fun _ _ ↦ .all _ _).comp (Bialgebra.comulAlgHom R Λ)) = specHom φ ∗ specHom ψ

lemma **specHom_counit** {T : Type} [CommRing T] [Algebra R T] : specHom ((Algebra.ofId R T).comp (Bialgebra.counitAlgHom R Λ)) = toUnit _ ≫ η[specOver R Λ]

lemma **specHom_comp_coaction** {T : Type} [CommRing T] [Algebra R T] (φ : B ⊗[R] Λ →ₐ[R] T) : specHom (φ.comp ρ) ≫ v = (specHom (φ.comp includeRight) ≫ i) ∗ (specHom (φ.comp includeLeft) ≫ v)

lemma **coaction_counit** [IsMonHom i] [Mono v] (b : B) : TensorProduct.rid R B ((Coalgebra.counit (R

lemma **coaction_assoc** [IsCommMonObj A] [IsMonHom i] [Mono v] (b : B) : TensorProduct.assoc R B Λ Λ (ρ.toLinearMap.rTensor Λ (ρ b)) = (Coalgebra.comul (R

lemma **coaction_free** [Mono i] [Module.Finite R Λ] : Function.Surjective (productMap (includeLeft : B →ₐ[R] B ⊗[R] Λ) ρ)

lemma **isPushout_equalizer** {T : Type} [CommRing T] [Algebra R T] (α β : B →ₐ[R] T) (h₁ : Function.Surjective (productMap β α)) (h₂ : RingHom.ker (productMap β α) = Ideal.span {z | ∃ b ∈ AlgHom.equalizer α β, z = b ⊗ₜ[R] (1 : B) - (1 : B) ⊗ₜ[R] b}) : IsPushout (CommRingCat.ofHom (α.toRingHom.eqLocus β.toRingHom).subtype) (CommRingCat.ofHom (α.toRingHom.eqLocus β.toRingHom).subtype) (CommRingCat.ofHom α.toRingHom) (CommRingCat.ofHom β.toRingHom)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SplitModelToPlaceModel

theorem **powLB_eq_pw** (L : LB X) (n : ℕ) : IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.powLB L n = AbLaw.pw L n

theorem **isLB_powLB** (hL : L.IsLB) (n : ℕ) : (IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.powLB L n).IsLB

theorem **lbIso_powLB** (hL : L.IsLB) (n : ℕ) : LBIso (IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.powLB L n) (lbPow L n)

theorem **isAmpleLB_iff** (hL : L.IsLB) : GlobalTheory.IsAmpleLB L ↔ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsAmpleLB L

theorem **rigidified_iff** (L : LB A.left) : Out.Rigidified A L ↔ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsRigidified A L

theorem **sym_iff_locZero** (hL : L.IsLB) : Out.Sym A L ↔ ∀ s : S, ∃ U : S.Opens, s ∈ U ∧ pull (A.hom ⁻¹ᵁ U).ι (pull (GrpObj.inv : A ⟶ A).left (cls L hL) - cls L hL) = 0

theorem **isSymmetric_iff_cls** (hL : L.IsLB) : IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsSymmetric A L ↔ pull (GrpObj.inv : A ⟶ A).left (cls L hL) - cls L hL = 0

theorem **sym_of_isSymmetric** (hL : L.IsLB) (h : IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsSymmetric A L) : Out.Sym A L

theorem **sym_iff** (hA : IsAbelianScheme A) (hL : L.IsLB) : Out.Sym A L ↔ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsSymmetric A L

theorem **relAmple_iff_isProper_and** (hL : L.IsLB) : Out.RelAmple A L ↔ IsProper A.hom ∧ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsRelAmple A L

theorem **relAmple_iff** (hA : IsProper A.hom) (hL : L.IsLB) : Out.RelAmple A L ↔ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.IsRelAmple A L

theorem **memK_iff_cls** (hL : L.IsLB) {T : Over S} (x : T ⟶ A) : IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.MemK A L x ↔ lam (cls L hL) (fst T A ≫ x) (snd T A) = 0

theorem **inK_of_memK** (hL : L.IsLB) {T : Over S} {x : T ⟶ A} (h : IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.MemK A L x) : UU.InK A L x

theorem **exists_memK_of_inK** (hL : L.IsLB) {T : Over S} {x : T ⟶ A} (h : UU.InK A L x) (p : T.left) : ∃ V : T.left.Opens, p ∈ V ∧ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.MemK A L ((Over.homMk V.ι rfl : Over.mk (V.ι ≫ T.hom) ⟶ T) ≫ x)

theorem **kTriv_iff** (hL : L.IsLB) : Out.KTriv A L ↔ IndefiniteRowSideFullHecke.PolarizedAbelianVarieties.KTrivial A L

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SplitTorusHomEqOneOfNilpotent

theorem **spm_left** (φ : C →ₐ[A] D) : (spm φ).left = Spec.map (CommRingCat.ofHom φ.toRingHom)

theorem **spm_comp** (φ : C →ₐ[A] D) (ψ : D →ₐ[A] E) : spm (ψ.comp φ) = spm ψ ≫ spm φ

theorem **spm_injective** {φ ψ : C →ₐ[A] D} (h : spm φ = spm ψ) : φ = ψ

theorem **spm_surjective** (g : sp A D ⟶ sp A C) : ∃ φ : C →ₐ[A] D, spm φ = g

theorem **spm_left_surjective** (π : C →ₐ[A] D) (hs : Function.Surjective π) (hn : ∀ x, π x = 0 → IsNilpotent x) : Function.Surjective (spm π).left

theorem **pt_injective** {C : Type} [CommRing C] [Algebra A C] {φ ψ : R →ₐ[A] C} (h : spm φ ≫ j = spm ψ ≫ j) : φ = ψ

theorem **pt_one** (C : Type) [CommRing C] [Algebra A C] : spm ((Algebra.ofId A C).comp ε) ≫ j = 1

theorem **exists_pt** {C D : Type} [CommRing C] [Algebra A C] [CommRing D] [Algebra A D] (π : C →ₐ[A] D) (hs : Function.Surjective π) (hn : ∀ x, π x = 0 → IsNilpotent x) (g : sp A C ⟶ B) (hg : spm π ≫ g = 1) : ∃ φ : R →ₐ[A] C, spm φ ≫ j = g

theorem **mul_pt** {C : Type} [CommRing C] [Algebra A C] (K : Ideal C) (hK : ∀ x ∈ K, IsNilpotent x) (φ₁ φ₂ φ : R →ₐ[A] C) (h₁ : ∀ a, φ₁ a - algebraMap A C (ε a) ∈ K) (h₂ : ∀ a, φ₂ a - algebraMap A C (ε a) ∈ K) (h : (spm φ₁ ≫ j) ∗ (spm φ₂ ≫ j) = spm φ ≫ j) (a : R) : φ a - (φ₁ a + φ₂ a - algebraMap A C (ε a)) ∈ K ^ 2

theorem **pullback_map_fst** {S S' : Scheme.{0}} {p : S' ⟶ S} {Z : Over S'} {V W : Over S} (u : Z ⟶ (Over.pullback p).obj V) (f : V ⟶ W) : (u ≫ (Over.pullback p).map f).left ≫ pullback.fst W.hom p = u.left ≫ pullback.fst V.hom p ≫ f.left

theorem **comp_eq_of_pullback_map_eq** {S S' : Scheme.{0}} {p : S' ⟶ S} {Z V W : Over S} {f g : V ⟶ W} (h : (Over.pullback p).map f = (Over.pullback p).map g) (t : Z ⟶ V) (s : Z.left ⟶ S') (hs : s ≫ p = Z.hom) : t ≫ f = t ≫ g

theorem **torusPt_mul** {C : Type} [CommRing C] [Algebra A C] (χ χ' : Multiplicative X →* C) : torusPt A X (χ * χ') = torusPt A X χ * torusPt A X χ'

theorem **torusPt_fst** {C : Type} [CommRing C] [Algebra A C] (χ : Multiplicative X →* C) : ∃ φ : AddMonoidAlgebra (ULift.{0} ℤ) X →+* C, (∀ x, φ (AddMonoidAlgebra.single x 1) = χ (Multiplicative.ofAdd x)) ∧ (torusPt A X χ).left ≫ pullback.fst (splitTorusAff ZU X).hom (toZ _) = Spec.map (CommRingCat.ofHom φ)

theorem **ringHom_uliftInt_ext** {C : Type} [Semiring C] (f g : ULift.{0} ℤ →+* C) : f = g

theorem **ringHom_ext_int_torus** {C : Type} [Semiring C] {f g : AddMonoidAlgebra (ULift.{0} ℤ) X →+* C} (h : ∀ y, f (AddMonoidAlgebra.single y 1) = g (AddMonoidAlgebra.single y 1)) : f = g

theorem **spm_torusPt** {C D : Type} [CommRing C] [Algebra A C] [CommRing D] [Algebra A D] (ψ : C →ₐ[A] D) (χ : Multiplicative X →* C) : spm ψ ≫ torusPt A X χ = torusPt A X ((ψ : C →* D).comp χ)

theorem **isIso_torusPt_of** : IsIso (torusPt A X (AddMonoidAlgebra.of A X)).left

theorem **torus_diag** : (spm (dm A X (AddMonoidHom.inl X X)) ≫ torusPt A X (AddMonoidAlgebra.of A X)) * (spm (dm A X (AddMonoidHom.inr X X)) ≫ torusPt A X (AddMonoidAlgebra.of A X)) = spm (dm A X ((AddMonoidHom.id X).prod (AddMonoidHom.id X))) ≫ torusPt A X (AddMonoidAlgebra.of A X)

theorem **exists_retraction** : ∃ r : AddMonoidAlgebra A (X × X) →ₗ[A] AddMonoidAlgebra A X, ∀ d, r (dm A X ((AddMonoidHom.id X).prod (AddMonoidHom.id X)) d - (dm A X (AddMonoidHom.inl X X) d + dm A X (AddMonoidHom.inr X X) d)) = d

theorem **exists_chart** [IsLocalRing A] (B : Over (Spec (CommRingCat.of A))) [GrpObj B] : ∃ (R : Type) (_ : CommRing R) (_ : Algebra A R) (j : sp A R ⟶ B) (_ : IsOpenImmersion j.left) (ε : R →ₐ[A] A), spm ε ≫ j = 1

theorem **isNilpotent_of_mem_map** {C : Type} [CommRing C] [Algebra A C] {J : Ideal A} (hJ : IsNilpotent J) (k : ℕ) (x : C) (hx : x ∈ (J ^ (k + 1)).map (algebraMap A C)) : IsNilpotent x

theorem **map_mem_map** {C D : Type} [CommRing C] [Algebra A C] [CommRing D] [Algebra A D] (I : Ideal A) (r : C →ₗ[A] D) {x : C} (hx : x ∈ I.map (algebraMap A C)) : r x ∈ I.map (algebraMap A D)

theorem **splitTorus_hom_eq_one_of_nilpotent** (A : Type) [CommRing A] [IsLocalRing A] (J : Ideal A) (hJ : IsNilpotent J) (X : Type) [AddCommGroup X] (B : Over (Spec (CommRingCat.of A))) [GrpObj B] (f : splitTorus (Spec (CommRingCat.of A)) X ⟶ B) (hf : IsMonHom f) (h : (Over.pullback (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk J)))).map f = (Over.pullback (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk J)))).map (toUnit _ ≫ η[B])) : f = toUnit _ ≫ η[B]

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SplitTorusHomProperTrivialField

lemma **spm_comp** {R R' R'' : Type} [CommRing R] [Algebra K R] [CommRing R'] [Algebra K R'] [CommRing R''] [Algebra K R''] (g : R →ₐ[K] R') (h : R' →ₐ[K] R'') : spm h ≫ spm g = spm (h.comp g)

lemma **spm_id** {R : Type} [CommRing R] [Algebra K R] : spm (AlgHom.id K R) = 𝟙 _

lemma **point_ext** {V F : Type} [CommRing V] [IsDomain V] [Algebra K V] [CommRing F] [Algebra K F] (j : V →ₐ[K] F) (hj : Function.Injective j) {B : Over (Spec (CommRingCat.of K))} [IsSeparated B.hom] {b b' : sp K V ⟶ B} (h : spm j ≫ b = spm j ≫ b') : b = b'

lemma **point_extend** {V F : Type} [CommRing V] [IsDomain V] [ValuationRing V] [Algebra K V] [Field F] [Algebra K F] [Algebra V F] [IsScalarTower K V F] [IsFractionRing V F] {B : Over (Spec (CommRingCat.of K))} [IsProper B.hom] (p : sp K F ⟶ B) : ∃ b : sp K V ⟶ B, spm (IsScalarTower.toAlgHom K V F) ≫ b = p

lemma **units_family_trivial** {B : Over (Spec (CommRingCat.of K))} [GrpObj B] [IsProper B.hom] (Φ : ∀ (R : Type) [CommRing R] [Algebra K R], Rˣ →* (sp K R ⟶ B)) (hΦ : ∀ (R R' : Type) [CommRing R] [Algebra K R] [CommRing R'] [Algebra K R'] (g : R →ₐ[K] R') (x : Rˣ), Φ R' (Units.map (g : R →* R') x) = spm g ≫ Φ R x) (L : Type) [Field L] [Algebra K L] (x : Lˣ) : Φ L x = 1

lemma **mul_left_fst** {S S' A : Scheme.{0}} (f : S' ⟶ S) (G : Over S) [GrpObj G] {a : A ⟶ S'} {c : A ⟶ S} (x y : Over.mk a ⟶ (Over.pullback f).obj G) (x₀ y₀ : Over.mk c ⟶ G) (hx : x₀.left = x.left ≫ pullback.fst G.hom f) (hy : y₀.left = y.left ≫ pullback.fst G.hom f) : (x * y).left ≫ pullback.fst G.hom f = (x₀ * y₀).left

lemma **charRing_single** (R : Type) [CommRing R] (φ : Multiplicative X →* Rˣ) (a : X) : charRing X R φ (AddMonoidAlgebra.single a 1) = φ (Multiplicative.ofAdd a)

lemma **ptFun_fst** (R : Type) [CommRing R] [Algebra K R] (φ : Multiplicative X →* Rˣ) : (ptFun K X R φ).left ≫ Rosenlicht.torusFst _ X = Spec.map (CommRingCat.ofHom (charRing X R φ))

lemma **torus_ext** {Z : Over (Spec (CommRingCat.of K))} {t t' : Z ⟶ splitTorus (Spec (CommRingCat.of K)) X} (h : t.left ≫ Rosenlicht.torusFst _ X = t'.left ≫ Rosenlicht.torusFst _ X) : t = t'

lemma **ptFun_mul** (R : Type) [CommRing R] [Algebra K R] (φ ψ : Multiplicative X →* Rˣ) : ptFun K X R (φ * ψ) = ptFun K X R φ * ptFun K X R ψ

lemma **pt_nat** {R R' : Type} [CommRing R] [Algebra K R] [CommRing R'] [Algebra K R'] (g : R →ₐ[K] R') (φ : Multiplicative X →* Rˣ) : pt K X R' ((Units.map (g : R →* R')).comp φ) = spm g ≫ pt K X R φ

lemma **pt_univ** : (pt K X _ (univ K X)).left = (Rosenlicht.torusIso K X).hom

lemma **char_eq_prod** {I : Type} [Fintype I] (e : Module.Basis I ℤ X) (G : Type) [CommGroup G] (φ : Multiplicative X →* G) : φ = ∏ i, coch (e.coord i).toAddMonoidHom G (φ (Multiplicative.ofAdd (e i)))

theorem **splitTorus_hom_proper_trivial_field** (K : Type) [Field K] (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (B : Over (Spec (CommRingCat.of K))) [GrpObj B] [IsProper B.hom] (f : splitTorus (Spec (CommRingCat.of K)) X ⟶ B) (hf : IsMonHom f) : f = toUnit _ ≫ η[B]

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SplitTorusHomProperTrivialLocal

theorem **pullback_map_eq_one** {S S' : Scheme.{0}} (p : S' ⟶ S) (X : Type) [AddCommGroup X] (B : Over S) [GrpObj B] (f : splitTorus S X ⟶ B) (hf : IsMonHom f) (h : ∀ g : splitTorus S' X ⟶ (Over.pullback p).obj B, IsMonHom g → g = toUnit _ ≫ η[(Over.pullback p).obj B]) : (Over.pullback p).map f = (Over.pullback p).map (toUnit _ ≫ η[B])

theorem **eq_one_of_isNilpotent_maximalIdeal** (A : Type) [CommRing A] [IsLocalRing A] (hA : IsNilpotent (IsLocalRing.maximalIdeal A)) (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (B : Over (Spec (CommRingCat.of A))) [GrpObj B] [IsProper B.hom] (f : splitTorus (Spec (CommRingCat.of A)) X ⟶ B) (hf : IsMonHom f) : f = toUnit _ ≫ η[B]

theorem **eq_zero_of_forall_mem_map_pow** {A C : Type} [CommRing A] [IsLocalRing A] [IsNoetherianRing A] [CommRing C] [Algebra A C] [Module.Free A C] (x : C) (hx : ∀ n : ℕ, x ∈ (IsLocalRing.maximalIdeal A ^ (n + 1)).map (algebraMap A C)) : x = 0

theorem **pt_ext_of_quotients** {A C : Type} [CommRing A] [CommRing C] [Algebra A C] {N : Type} (I : N → Ideal C) (hI : ∀ x, (∀ i, x ∈ I i) → x = 0) {B : Over (Spec (CommRingCat.of A))} [IsSeparated B.hom] {a b : sp A C ⟶ B} (h : ∀ i, spm (Ideal.Quotient.mkₐ A (I i)) ≫ a = spm (Ideal.Quotient.mkₐ A (I i)) ≫ b) : a = b

theorem **splitTorus_hom_proper_trivial_local** (A : Type) [CommRing A] [IsLocalRing A] [IsNoetherianRing A] (X : Type) [AddCommGroup X] [Module.Free ℤ X] [Module.Finite ℤ X] (B : Over (Spec (CommRingCat.of A))) [GrpObj B] [IsProper B.hom] (f : splitTorus (Spec (CommRingCat.of A)) X ⟶ B) (hf : IsMonHom f) : f = toUnit _ ≫ η[B]

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SurjectiveAppOfIsAbelianScheme

theorem **eq_comp_section_of_local_rigidity** {X T Y S : Scheme.{0}} (p : X ⟶ T) (ε : T ⟶ X) (g : X ⟶ Y) (π : T ⟶ S) (q : Y ⟶ S) [IsSeparated q] (he : ε ≫ p = 𝟙 T) (hg : g ≫ q = p ≫ π) (hp : IsOpenMap p) (hloc : ∀ (t : T) (y : Y), (∀ x : X, p x = t → g x = y) → ∃ V : T.Opens, t ∈ V ∧ (p ⁻¹ᵁ V).ι ≫ g = (p ⁻¹ᵁ V).ι ≫ p ≫ ε ≫ g) (hconn : ∀ s : S, _root_.IsPreconnected (π ⁻¹' {s})) (hmeet : ∀ s : S, ∃ (t : T) (y : Y), π t = s ∧ ∀ x : X, p x = t → g x = y) : g = p ≫ ε ≫ g

theorem **surjective_app_of_isAbelianScheme** (hG : GlobalFnsSt) {S : Scheme.{0}} (A : Over S) [GrpObj A] (hA : IsAbelianScheme A) (V : S.Opens) : Function.Surjective (A.hom.app V).hom

theorem **rigidity_of_local_rigidity** : GlobalFnsSt → (∀ (X T Y : Scheme.{0}) (p : X ⟶ T) (ε : T ⟶ X) (g : X ⟶ Y), ε ≫ p = 𝟙 T → UniversallyClosed p → (∀ V : T.Opens, Function.Surjective (p.app V).hom) → ∀ (t : T) (y : Y), (∀ x : X, p x = t → g x = y) → ∃ V : T.Opens, t ∈ V ∧ (p ⁻¹ᵁ V).ι ≫ g = (p ⁻¹ᵁ V).ι ≫ p ≫ ε ≫ g) → RigiditySt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SymmetricLiftOddOrderExistsUnique

theorem **defect_pow** {G U : Type} [CommGroup G] [CommGroup U] (f n : G → U) (X : G) (k : ℕ) (hX : X ^ k = 1) (α β : U) (hα : ∀ m, f (X⁻¹ * m) * n m * (f m⁻¹ * (n (X⁻¹ * m))⁻¹) = α) (hβ : ∏ j ∈ Finset.range k, f (X ^ j) = β) : α ^ k = β ^ 2

theorem **isGen_of_gm** {t : LB.Gen (LB.LinComb L φ ε)} (h : ∀ (y : Y) (j j' : K → L.ι), y ∈ LB.wchart L φ j → y ∈ LB.wchart L φ j' → gm (𝟙 Y) y (LB.wchart L φ j) (t.s j) = trF φ ε (𝟙 Y) y j j' * gm (𝟙 Y) y (LB.wchart L φ j') (t.s j')) : t.IsGen

theorem **exists_const** (hL : L.IsLB) {S₀ : Scheme.{0}} (p : Y ⟶ S₀) (e : S₀ ⟶ Y) (he : e ≫ p = 𝟙 _) (hp : Function.Surjective p.appTop.hom) (hF : ∀ (U : Type) [CommGroup U] (h : (Y ⟶ X) → U), ∏ k, h (φ k) ^ ε k = 1) {u : LB.Gen (LB.LinComb L φ ε)} (hu : u.IsGen) : ∃ a : Γ(S₀, ⊤)ˣ, ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J), Good L z J → val u r z J = gm (r ≫ p) z ⊤ a

theorem **surj_fst** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (T : Over S) : Function.Surjective (fst T A).left.appTop.hom

theorem **exists_constT** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} {K : Type} [Fintype K] {φ : K → ((T ⊗ A).left ⟶ A.left)} {ε : K → ℤ} (hF : ∀ (U : Type) [CommGroup U] (h : ((T ⊗ A).left ⟶ A.left) → U), ∏ k, h (φ k) ^ ε k = 1) {u : LB.Gen (LB.LinComb L φ ε)} (hu : u.IsGen) : ∃ a : Γ(T.left, ⊤)ˣ, ∀ (m : T ⊗ A ⟶ A) (y : (T ⊗ A).left) (J), Good L y J → val u (lift (fst T A) m).left y J = gm (fst T A).left y ⊤ a

theorem **lift_translT** {T : Over S} (x : T ⟶ A) (m : T ⊗ A ⟶ A) : lift (fst T A) m ≫ translT A x = lift (fst T A) ((fst T A ≫ x) * m)

theorem **lift_negT** {T : Over S} (m : T ⊗ A ⟶ A) : lift (fst T A) m ≫ negT A T = lift (fst T A) m⁻¹

theorem **lift_translPt** {T : Over S} (x : T ⟶ A) (m : T ⊗ A ⟶ A) : lift (fst T A) m ≫ translPt A x = (fst T A ≫ x) * m

theorem **translT_eq** {T : Over S} (x : T ⟶ A) : translT A x = lift (fst T A) ((fst T A ≫ x) * snd T A)

theorem **negT_eq** (T : Over S) : negT A T = lift (fst T A) (snd T A)⁻¹

theorem **comp_pow'** {T T' : Over S} (ρ : T' ⟶ T) (x : T ⟶ A) (n : ℕ) : ρ ≫ x ^ n = (ρ ≫ x) ^ n

theorem **comp_inv'** {T T' : Over S} (ρ : T' ⟶ T) (x : T ⟶ A) : ρ ≫ x⁻¹ = (ρ ≫ x)⁻¹

theorem **val_symGen** (θ : ThetaGrp A L k T) (nu : LB.Gen (negLB A L k T)) {x : T ⟶ A} (hx : θ.pt = x) {y : (T ⊗ A).left} {J} (hJ : Good L y J) (m : T ⊗ A ⟶ A) : val (symGen A L k θ nu) (lift (fst T A) m).left y J = thetaVal θ y J ((fst T A ≫ x)⁻¹ * m) * negVal nu y J m * (thetaVal θ y J m⁻¹ * (negVal nu y J ((fst T A ≫ x)⁻¹ * m))⁻¹)

theorem **gm_powGen** (θ : ThetaGrp A L k T) (n : ℕ) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) (j : Fin n × Fin 2 → L.ι) (hj : r z ∈ LB.wchart L (powφ A θ.pt n) j) : gm r z (LB.wchart L (powφ A θ.pt n) j) ((powGen A L k θ n).s j) = ∏ j₀ : Fin n, gm r z (LB.wchart L (fun b => (translT A (θ.pt ^ (j₀ : ℕ))).left ≫ (![(snd T A).left, (translPt A θ.pt).left] b)) (fun b => j (j₀, b))) ((LB.pullGen L ![(snd T A).left, (translPt A θ.pt).left] ![k, -k] θ.φ (translT A (θ.pt ^ (j₀ : ℕ))).left).s (fun b => j (j₀, b)))

theorem **powGen_isGen** (θ : ThetaGrp A L k T) (n : ℕ) : (powGen A L k θ n).IsGen

theorem **val_powGen** (θ : ThetaGrp A L k T) (n : ℕ) {x : T ⟶ A} (hx : θ.pt = x) {y : (T ⊗ A).left} {J} (hJ : Good L y J) (m : T ⊗ A ⟶ A) : val (powGen A L k θ n) (lift (fst T A) m).left y J = ∏ j ∈ Finset.range n, thetaVal θ y J ((fst T A ≫ x) ^ j * m)

theorem **form_pow** {n : ℕ} {x : T ⟶ A} (hx : x ^ n = 1) (U : Type) [CommGroup U] (h : ((T ⊗ A).left ⟶ A.left) → U) : ∏ p, h (powφ A x n p) ^ powε k n p = 1

theorem **symGen_isGen** (θ : ThetaGrp A L k T) {nu : LB.Gen (negLB A L k T)} (hnu : nu.IsGen) : (symGen A L k θ nu).IsGen

theorem **exists_symConst** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) (θ : ThetaGrp A L k T) {nu : LB.Gen (negLB A L k T)} (hnu : nu.IsGen) : ∃ a : Γ(T.left, ⊤)ˣ, ∀ (m : T ⊗ A ⟶ A) (y : (T ⊗ A).left) (J), Good L y J → val (symGen A L k θ nu) (lift (fst T A) m).left y J = gm (fst T A).left y ⊤ a

theorem **exists_scale** (hL : L.IsLB) (θ : ThetaGrp A L k T) (c : Γ(T.left, ⊤)ˣ) : ∃ θ' : ThetaGrp A L k T, θ'.pt = θ.pt ∧ ∀ y J, Good L y J → ∀ m, thetaVal θ' y J m = thetaVal θ y J m * gm (fst T A).left y ⊤ c

theorem **exists_ratio** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {K : Type} [Fintype K] {φ : K → ((T ⊗ A).left ⟶ A.left)} {ε : K → ℤ} {u u' : LB.Gen (LB.LinComb L φ ε)} (hu : u.IsGen) (hu' : u'.IsGen) : ∃ c : Γ(T.left, ⊤)ˣ, ∀ (m : T ⊗ A ⟶ A) (y : (T ⊗ A).left) (J), Good L y J → val u (lift (fst T A) m).left y J = val u' (lift (fst T A) m).left y J * gm (fst T A).left y ⊤ c

theorem **id_eq_lift** : 𝟙 (T ⊗ A).left = (lift (fst T A) (snd T A)).left

theorem **exists_symLift** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {n : ℕ} (hn : Odd n) {x : T ⟶ A} (hx : x ^ n = 1) (θ : ThetaGrp A L k T) (hθ : θ.pt = x) {nu : LB.Gen (negLB A L k T)} (hnu : nu.IsGen) : ∃ θ' : ThetaGrp A L k T, SymLift A L k n x θ'

theorem **symLift_unique** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {n : ℕ} (hn : Odd n) {x : T ⟶ A} (θ θ' : ThetaGrp A L k T) (h : SymLift A L k n x θ) (h' : SymLift A L k n x θ') : LB.Gen.Corr L _ _ _ _ θ.φ θ'.φ

theorem **symLift_odd** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResι V ≫ x) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ nu : LB.Gen (negLB P.A P.L k T), nu.IsGen) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ), Odd k → ∀ (T : Over S) (x : T ⟶

P.A), x ^ k = 1 → (∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L k (overRes V), SymLift P.A P.L k k (overResι V ≫ x) θ) ∧ ∀ θ θ' : ThetaGrp P.A P.L k T, SymLift P.A P.L k k x θ → SymLift P.A P.L k k x θ' → LB.Gen.Corr P.L _ _ _ _ θ.φ θ'.φ) ∧ SymLiftSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.SymmetricLiftOddOrderExistsUniqueStatement

theorem **_root_.GlobalTheory.T.symmetric_lift_odd_order_exists_unique** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResι V ≫ x) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ nu : LB.Gen (negLB P.A P.L k T), nu.IsGen) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ), Odd k → ∀ (T : Over S) (x : T ⟶ P.A), x ^ k = 1 → (∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L k (overRes V), SymLift P.A P.L k k (overResι V ≫ x) θ) ∧ ∀ θ θ' : ThetaGrp P.A P.L k T, SymLift P.A P.L k k x θ → SymLift P.A P.L k k x θ' → LB.Gen.Corr P.L _ _ _ _ θ.φ θ'.φ) ∧ SymLiftSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.ThetaGrpActsLinearly

theorem **sv_ext** (hL : L.IsLB) {v w : SecM A L k T} (h : ∀ y J, Good L y J → sv v (𝟙 _) y J = sv w (𝟙 _) y J) : v = w

theorem **sv_add** (v w : SecM A L k T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : sv (v + w) r z J = sv v r z J + sv w r z J

theorem **sv_smul** (c : Γ(T.left, ⊤)) (v : SecM A L k T) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (z : Z) {J} (hJ : Good L z J) : sv (c • v) r z J = gs (r ≫ (fst T A).left) z ⊤ c * sv v r z J

theorem **translT_snd** (x : T ⟶ A) : (translT A x).left ≫ (snd T A).left = (translPt A x).left

theorem **translT_translPt** (x y : T ⟶ A) : translT A x ≫ translPt A y = translPt A (y * x)

theorem **translT_translT** (x y : T ⟶ A) : (translT A x).left ≫ (translT A y).left = (translT A (y * x)).left

theorem **translPt_one** : translPt A (1 : T ⟶ A) = snd T A

theorem **translT_one** : (translT A (1 : T ⟶ A)).left = 𝟙 _

theorem **isActOf_iff** (hL : L.IsLB) (θ : ThetaGrp A L k T) (s s' : LB.Sec (LPow A L k T)) : IsActOf A L k θ s s' ↔ ∀ y J, Good L y J → sv (SecM.ofSec A s') (𝟙 _) y J = (tval θ (𝟙 _) y J).val * sv (SecM.ofSec A s) (𝟙 _ ≫ (translT A θ.pt).left) y J

theorem **exists_act** (hL : L.IsLB) (θ : ThetaGrp A L k T) (v : SecM A L k T) : ∃ w : SecM A L k T, ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → sv w r z J = (tval θ r z J).val * sv v (r ≫ (translT A θ.pt).left) z J

theorem **exists_lin** (hL : L.IsLB) (θ : ThetaGrp A L k T) : ∃ f : SecM A L k T →ₗ[Γ(T.left, ⊤)] SecM A L k T, ∀ v (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → sv (f v) r z J = (tval θ r z J).val * sv v (r ≫ (translT A θ.pt).left) z J

theorem **exists_inv** (hL : L.IsLB) (θ : ThetaGrp A L k T) : ∃ θ' : ThetaGrp A L k T, θ'.pt = θ.pt⁻¹ ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → tval θ' r z J = (tval θ (r ≫ (translT A θ.pt⁻¹).left) z J)⁻¹

theorem **exists_mul** (hL : L.IsLB) (θx θy : ThetaGrp A L k T) : ∃ θ : ThetaGrp A L k T, θ.pt = θy.pt * θx.pt ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → tval θ r z J = tval θx r z J * tval θy (r ≫ (translT A θx.pt).left) z J

theorem **exists_equiv** (hL : L.IsLB) (θ : ThetaGrp A L k T) : ∃ f : SecM A L k T ≃ₗ[Γ(T.left, ⊤)] SecM A L k T, ∀ v (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → sv (f v) r z J = (tval θ r z J).val * sv v (r ≫ (translT A θ.pt).left) z J

theorem **exists_ratio** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) (θ θ' : ThetaGrp A L k T) (h : θ.pt = θ'.pt) : ∃ c : Γ(T.left, ⊤)ˣ, ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → (tval θ' r z J).val = gs (r ≫ (fst T A).left) z ⊤ c.val * (tval θ r z J).val

theorem **thetaGrp_acts_linearly** : GlobalFnsSt → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℤ) (T : Over S), ∃ ρ : ThetaGrp P.A P.L k T → (SecM P.A P.L k T ≃ₗ[Γ(T.left, ⊤)] SecM P.A P.L k T), (∀ θ s s', IsActOf P.A P.L k θ s s' ↔ ρ θ (SecM.ofSec P.A s) = SecM.ofSec P.A s') ∧ (∀ θx θy c, CommIsOf P.A P.L k θx θy c → ∀ v, ρ θx (ρ θy v) = (c : Γ(T.left, ⊤)) • ρ θy (ρ θx v)) ∧ (∀ θ θ', θ.pt = θ'.pt → ∃ c : Γ(T.left, ⊤)ˣ, ∀ v, ρ θ' v = (c : Γ(T.left, ⊤)) • ρ θ v) ∧ ∀ θx θy, ∃ θ, θ.pt = θx.pt * θy.pt ∧ ∀ v, ρ θ v = ρ θx (ρ θy v)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.TrdegEqOfLocallyIsStandardSmoothOfRelativeDimens

theorem **trdeg_eq_of_locally_isStandardSmoothOfRelativeDimension** {K B : Type} [Field K] [CommRing B] [IsDomain B] [Algebra K B] {g : ℕ} (h : RingHom.Locally (RingHom.IsStandardSmoothOfRelativeDimension g) (algebraMap K B)) : Algebra.trdeg K B = g

theorem **trdeg_eq_of_smoothOfRelativeDimension** : ∀ (K : Type) [Field K] (X : Scheme.{0}) [IsIntegral X] (f : X ⟶ Spec (CommRingCat.of K)) (g : ℕ), SmoothOfRelativeDimension g f → ∀ (B : Type) [CommRing B] [Nontrivial B] [Algebra K B] (j : Spec (CommRingCat.of B) ⟶ X), IsOpenImmersion j → j ≫ f = Spec.map (CommRingCat.ofHom (algebraMap K B)) → Algebra.trdeg K B = g

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.TrivBasicOpenOfTrivFibre

theorem **triv_comp** {X Y Z : Scheme.{0}} {M : LB X} {f : Y ⟶ X} (h : Z ⟶ Y) {k : Z ⟶ X} (hk : h ≫ f = k) (ht : Triv (M.pull f)) : Triv (M.pull k)

theorem **triv_pull_pull** {X Y Z : Scheme.{0}} {M : LB X} (f : Y ⟶ X) (h : Z ⟶ Y) (ht : Triv (M.pull (h ≫ f))) : Triv ((M.pull f).pull h)

theorem **eq_bot_of_small** {S : Type} [CommRing S] [IsLocalRing S] (J : Ideal S) (hJ : J.FG) (H : ∀ (I' : Ideal S) (e : S), e ∈ J → IsLocalRing.maximalIdeal S * J ≤ I' → I' ≤ J → J ≤ I' ⊔ Ideal.span {e} → J ≤ I') : J = ⊥

theorem **exists_ann** {R : Type} [CommRing R] (p : Ideal R) [p.IsPrime] (I : Ideal R) (hI : I.FG) (hIp : I ≤ p) (H : ∀ (T : Type) [CommRing T] [IsLocalRing T] (g : R →+* T) (e : T), (∀ r ∉ p, IsUnit (g r)) → (∀ a ∈ IsLocalRing.maximalIdeal T, a * e = 0) → e ∈ IsLocalRing.maximalIdeal T → I ≤ RingHom.ker ((Ideal.Quotient.mk (Ideal.span {e})).comp g) → I ≤ RingHom.ker g) : ∃ c ∉ p, ∀ x ∈ I, c * x = 0

theorem **isPullback_iso_comp** {W P X Y Z : Scheme.{0}} {a : P ⟶ X} {b : P ⟶ Y} {f : X ⟶ Z} {g : Y ⟶ Z} (h : IsPullback a b f g) (e : W ⟶ P) [IsIso e] : IsPullback (e ≫ a) (e ≫ b) f g

theorem **isPullback_sqMap** : IsPullback (sqMap φ A) ((Over.pullback φ).obj A ⊗ (Over.pullback φ).obj A).hom (A ⊗ A).hom φ

theorem **lift_sqMap** (a b : A ⟶ A) : (lift ((Over.pullback φ).map a) ((Over.pullback φ).map b)).left ≫ sqMap φ A = pullback.fst A.hom φ ≫ (lift a b).left

theorem **triv_lift** (hobs : ∀ (R : Type) [CommRing R] [IsLocalRing R] (ε : R), (∀ a ∈ IsLocalRing.maximalIdeal R, a * ε = 0) → ∀ (A : Over (Spec (CommRingCat.of R))) [GrpObj A], IsAbelianScheme A → ∀ M : LB (A ⊗ A).left, M.IsLB → (∃ t : LB.Gen (M.pull (lift (𝟙 A) 1).left), t.IsGen) → (∃ t : LB.Gen (M.pull (lift 1 (𝟙 A)).left), t.IsGen) → ∀ (X' : Scheme.{0}) (π : X' ⟶ (A ⊗ A).left) (f' : X' ⟶ Spec (CommRingCat.of (R ⧸ Ideal.span {ε}))), IsPullback π f' (A ⊗ A).hom (Spec.map (CommRingCat.ofHom (Ideal.Quotient.mk (Ideal.span {ε})))) → ε ∈ IsLocalRing.maximalIdeal R → (∃ t : LB.Gen (M.pull π), t.IsGen) → ∃ t : LB.Gen M, t.IsGen) {S : Scheme.{0}} (A : Over S)

[GrpObj A] (hA : IsAbelianScheme A) (M : LB (A ⊗ A).left) (hM : M.IsLB) (h₁ : Triv (M.pull (lift (𝟙 A) 1).left)) (h₂ : Triv (M.pull (lift 1 (𝟙 A)).left)) (T : Type) [CommRing T] [IsLocalRing T] (φ : Spec (CommRingCat.of T) ⟶ S) (e : T) (he

theorem **bijective_appTop** (hG : GlobalFnsSt) {S S' X : Scheme.{0}} (B : Over S) [GrpObj B] (hB : IsAbelianScheme B) {φ : S' ⟶ S} {π : X ⟶ B.left} {f : X ⟶ S'} (h : IsPullback π f B.hom φ) : Function.Bijective f.appTop.hom

theorem **triv_basicOpen_of_triv_fibre** : GlobalFnsSt → (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) (M : LB X), M.IsLB → Nonempty (Perf f M) → UniversallyClosed f → Surjective f → ∀ (K : Type) [Field K] (g₀ : R →+* K) (X₀ : Scheme.{0}) (π₀ : X₀ ⟶ X) (f₀ : X₀ ⟶ Spec (CommRingCat.of K)), IsPullback π₀ f₀ f (Spec.map (CommRingCat.ofHom g₀)) → Function.Bijective f₀.appTop.hom → Triv (M.pull π₀) → ∃ (r : R) (I : Ideal R), g₀ r ≠ 0 ∧ I.FG ∧ ∀ (R' : Type) [CommRing R'] (g : R →+* R') (X' : Scheme.{0}) (π : X' ⟶ X) (f' : X' ⟶ Spec (CommRingCat.of R')), IsPullback π f' f (Spec.map (CommRingCat.ofHom g)) → IsUnit (g r) → (Triv (M.pull π) ↔ I ≤ RingHom.ker g)) → (∀ (R : Type) [CommRing R] [IsLocalRing R] (ε : R), (∀ a ∈ IsLocalRing.maximalIdeal R, a * ε = 0) → ∀ (A : Over (Spec (CommRingCat.of R))) [GrpObj A], IsAbelianScheme A → ∀ M : LB (A ⊗ A).left, M.IsLB → (∃ t :

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.TrivPullCompIff

theorem **TrivAt.comp** {L : LB X} {φ : K → (Y ⟶ X)} {ε : K → ℤ} {r : Z ⟶ Y} (H : TrivAt L φ ε r) (h : Z' ⟶ Z) : TrivAt L φ ε (h ≫ r)

theorem **triv_pull_iff** (L : LB X) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (r : Z ⟶ Y) : Triv ((LB.LinComb L φ ε).pull r) ↔ TrivAt L φ ε r

theorem **trivAt_of_formal** {L : LB X} (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} (r : Z ⟶ Y) (h : ∀ (U : Type) [CommGroup U] (k : (Z ⟶ X) → U), ∏ i, k (r ≫ φ i) ^ ε i = 1) : TrivAt L φ ε r

theorem **triv_of_local** {L : LB X} (hL : L.IsLB) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (p : Y ⟶ T) (e : T ⟶ Y) (he : e ≫ p = 𝟙 T) (hF : ∀ (Z : Scheme.{0}) (b : Z ⟶ T) (U : Type) [CommGroup U] (k : (Z ⟶ X) → U), ∏ i, k ((b ≫ e) ≫ φ i) ^ ε i = 1) (hs : ∀ V : T.Opens, ∃ (Z : Scheme.{0}) (m : Z ⟶ Y) (q : Z ⟶ V.toScheme), IsPullback m q p V.ι ∧ Function.Surjective q.appTop.hom) (h : ∀ s : T, ∃ U : T.Opens, s ∈ U ∧ TrivAt L φ ε (p ⁻¹ᵁ U).ι) : Triv (LB.LinComb L φ ε)

theorem **triv_pull_comp_iff** (q : W ⟶ Y) {Z : Scheme.{0}} (π : Z ⟶ W) : Triv ((LB.LinComb L (fun k => q ≫ φ k) ε).pull π) ↔ TrivAt L φ ε (π ≫ q)

theorem **isOpen_locTriv** (p : Y ⟶ T) : IsOpen (locTriv L φ ε p)

theorem **locTriv_subset** (p : Y ⟶ T) : locTriv L φ ε p ⊆ fibTriv L φ ε p

theorem **mem_range_spec** {R F : Type} [CommRing R] [Field F] (g : R →+* F) (x : PrimeSpectrum R) : x ∈ Set.range (Spec.map (CommRingCat.ofHom g)) ↔ RingHom.ker g = x.asIdeal

theorem **mem_fibTriv_iff** {p : Y ⟶ T} {R : Type} [CommRing R] {j : Spec (CommRingCat.of R) ⟶ T} [IsOpenImmersion j] {q : W ⟶ Y} {f : W ⟶ Spec (CommRingCat.of R)} (H : IsPullback q f p j) (x : PrimeSpectrum R) : j x ∈ fibTriv L φ ε p ↔ ∃ (F : Type) (_ : Field F) (g : R →+* F) (Z : Scheme.{0}) (π : Z ⟶ W) (f' : Z ⟶ Spec (CommRingCat.of F)), RingHom.ker g = x.asIdeal ∧ IsPullback π f' f (Spec.map (CommRingCat.ofHom g)) ∧ Triv ((LB.LinComb L (fun k => q ≫ φ k) ε).pull π)

theorem **fibTriv_clopen** (p : Y ⟶ T) (h : ∀ (R : Type) [CommRing R] (j : Spec (CommRingCat.of R) ⟶ T), ∃ (W : Scheme.{0}) (q : W ⟶ Y) (f : W ⟶ Spec (CommRingCat.of R)), IsPullback q f p j ∧ IsClosed {x : PrimeSpectrum R | ∃ (F : Type) (_ : Field F) (g : R →+* F) (Z : Scheme.{0}) (π : Z ⟶ W) (f' : Z ⟶ Spec (CommRingCat.of F)), RingHom.ker g = x.asIdeal ∧ IsPullback π f' f (Spec.map (CommRingCat.ofHom g)) ∧ Triv ((LB.LinComb L (fun k => q ≫ φ k) ε).pull π)} ∧ ∀ (F : Type) [Field F] (g : R →+* F) (Z : Scheme.{0}) (π : Z ⟶ W) (f' : Z ⟶ Spec (CommRingCat.of F)), IsPullback π f' f (Spec.map (CommRingCat.ofHom g)) → Triv ((LB.LinComb L (fun k => q ≫ φ k) ε).pull π) → ∃ r : R, g r ≠ 0 ∧ Triv ((LB.LinComb L (fun k => q ≫ φ k) ε).pull (f ⁻¹ᵁ (PrimeSpectrum.basicOpen r)).ι)) : fibTriv L φ ε p ⊆ locTriv L φ ε p ∧ IsClosed (fibTriv L φ ε p)

theorem **exists_cube_chart** {T : Scheme.{0}} (j : T ⟶ A.left) : ∃ q : ((Over.pullback (j ≫ A.hom)).obj A ⊗ (Over.pullback (j ≫ A.hom)).obj A).left ⟶ (G3 A).left, IsPullback q ((Over.pullback (j ≫ A.hom)).obj A ⊗ (Over.pullback (j ≫ A.hom)).obj A).hom (x3 A 2).left j ∧ q ≫ (x3 A 0).left = (fst _ _ : _ ⟶ (Over.pullback (j ≫ A.hom)).obj A).left ≫ pullback.fst A.hom (j ≫ A.hom) ∧ q ≫ (x3 A 1).left = (snd _ _ : _ ⟶ (Over.pullback (j ≫ A.hom)).obj A).left ≫ pullback.fst A.hom (j ≫ A.hom)

theorem **prod_face** [IsCommMonObj A] {Z : Scheme.{0}} (r : Z ⟶ (G3 A).left) (i : Fin 3) (b : Z ⟶ S) (h : r ≫ (x3 A i).left = b ≫ η[A].left) (U : Type) [CommGroup U] (k : (Z ⟶ A.left) → U) : ∏ I, k (r ≫ thφ A (x3 A) I) ^ thε I = 1

theorem **one_left_fst** {S' : Scheme.{0}} (f : S' ⟶ S) (T : Over S') : (1 : T ⟶ (Over.pullback f).obj A).left ≫ pullback.fst A.hom f = (T.hom ≫ f) ≫ η[A].left

theorem **isAb_baseChange** (hA : IsAbelianScheme A) {S' : Scheme.{0}} (f : S' ⟶ S) : IsAbelianScheme ((Over.pullback f).obj A)

theorem **exists_gen_theta** : (∀ (X S : Scheme.{0}) (f : X ⟶ S), Smooth f → GeometricallyConnected f → GeometricallyIntegral f) → GlobalFnsSt → (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)), Smooth f → IsProjBy f → ∀ M : LB X, M.IsLB → Nonempty (Perf f M)) → (∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s) → (∀ (R : Type) [CommRing R] (X Y : Over (Spec (CommRingCat.of R))), IsProjBy X.hom → IsProjBy Y.hom → IsProjBy (X ⊗ Y).hom) → (∀ (R : Type) [CommRing R] (X : Scheme.{0}) (f : X ⟶ Spec (CommRingCat.of R)) (M : LB X), M.IsLB → Nonempty (Perf f M) → Nonempty (Perf f (pw M (-1))) → IsP

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.TwistedPowTranslate

theorem **exists_gen_of_sec** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} (s : LB.Sec (LB.LinComb L φ ε)) (hs : ∀ (y : Y) J, Good L y J → IsUnit (sval s (𝟙 Y) y J)) : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen

theorem **gen_congr** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (hf : FormEq φ ε φ' ε') (h : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen) : ∃ t : LB.Gen (LB.LinComb L φ' ε'), t.IsGen

theorem **gen_mul** {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (h : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen) (h' : ∃ t : LB.Gen (LB.LinComb L φ' ε'), t.IsGen) : ∃ t : LB.Gen (LB.LinComb L (Sum.elim φ φ') (Sum.elim ε ε')), t.IsGen

theorem **gen_of_pull_top** (hL : L.IsLB) {φ : K → (Y ⟶ X)} {ε : K → ℤ} (W : Y.Opens) (hW : ∀ y, y ∈ W) (c : LB.Gen ((LB.LinComb L φ ε).pull W.ι)) (hc : c.IsGen) : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen

theorem **gen_square** (hsq : ∀ (x y : Over.mk t ⟶ P.A) (p : (Over.mk t).left), ∃ V : (Over.mk t).left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb P.L (K

theorem **gen_iter** (hsq : ∀ (x y : Over.mk t ⟶ P.A) (p : (Over.mk t).left), ∃ V : (Over.mk t).left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb P.L (K

theorem **exists_root** [IsAlgClosed K] (r : ℕ) [IsFinite (mulA P.A r).left] [Surjective (mulA P.A r).left] (c : Over.mk t ⟶ P.A) : ∃ d : Over.mk t ⟶ P.A, d ^ r = c

theorem **twisted_pow_translate** : (∀ (S : Scheme.{0}) (P : PolAV S) (T : Over S) (x y : T ⟶ P.A) (p : T.left), ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.UniformisationLawsOfBaseChangeLaws

theorem **uniformisationLaws_of_baseChange_laws** : PolBaseChangeSt → GlobalFnsSt → SecBaseChangeSt → DVR.TwistBaseChangeStatement → Laws4

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.UniqueFactorizationMonoidOfIsRegularRingOfIsLoca

theorem **isRegularRing_of_isLocalization** {R : Type*} [CommRing R] [IsRegularRing R] (M : Submonoid R) (S : Type*) [CommRing S] [Algebra R S] [IsLocalization M S] : IsRegularRing S

private lemma **prime_of_mem_maximalIdeal_of_not_mem_sq** {A : Type} [CommRing A] [IsRegularLocalRing A] {x : A} (hx : x ∈ maximalIdeal A) (hx2 : x ∉ maximalIdeal A ^ 2) : Prime x

private lemma **isPrincipal_of_isPrincipal_map_away_prime** {A : Type} [CommRing A] [IsDomain A] [IsNoetherianRing A] {x : A} (hx : Prime x) {p : Ideal A} [hp : p.IsPrime] (hxp : x ∉ p) (h : (p.map (algebraMap A (Localization.Away x))).IsPrincipal) : p.IsPrincipal

private lemma **uniqueFactorizationMonoid_localization_of_height_le** (hPD : ∀ (R : Type) [CommRing R] [IsRegularLocalRing R] (M : Type) [AddCommGroup M] [Module R M] [Module.Finite R M], ∃ n : ℕ, HasProjectiveDimensionLE (ModuleCat.of R M) n) (hPrin : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] [IsLocalRing R] [IsDomain R] (x : R) (R' : Type) [CommRing R'] [Algebra R R'] [IsLocalization.Away x R'] (I : Ideal R) (n : ℕ), HasProjectiveDimensionLE (ModuleCat.of R I) n → Module.Projective R' (I.map (algebraMap R R')) → (I.map (algebraMap R R')).IsPrincipal) {R : Type} [CommRing R] [IsRegularRing R] (n : ℕ) : ∀ (Q : Ideal R) [Q.IsPrime], Q.height ≤ n → UniqueFactorizationMonoid (Localization.AtPrime Q)

theorem **_root_.GlobalTheory.uniqueFactorizationMonoid_of_isRegularRing_of_isLocalRing** : (∀ (R : Type) [CommRing R] [IsRegularLocalRing R] (M : Type) [AddCommGroup M] [Module R M] [Module.Finite R M], ∃ n : ℕ, HasProjectiveDimensionLE (ModuleCat.of R M) n) → (∀ (R : Type) [CommRing R] [IsNoetherianRing R] [IsLocalRing R] [IsDomain R] (x : R) (R' : Type) [CommRing R'] [Algebra R R'] [IsLocalization.Away x R'] (I : Ideal R) (n : ℕ), HasProjectiveDimensionLE (ModuleCat.of R I) n → Module.Projective R' (I.map (algebraMap R R')) → (I.map (algebraMap R R')).IsPrincipal) → ∀ (R : Type) [CommRing R] [IsRegularRing R] [IsLocalRing R] [IsDomain R], UniqueFactorizationMonoid R

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.UnramExtLaw

theorem **unramExt_law** : UnramExtSt

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.VeryAmpleByOfRelAmple

theorem **veryAmpleBy_of_relAmple** {R : Type} [CommRing R] {A : Over (Spec (CommRingCat.of R))} {L : LB A.left} (hL : L.IsLB) (hA : RelAmple A L) {X : Scheme.{0}} (e : X ⟶ A.left) [IsIso e] {f : X ⟶ Spec (CommRingCat.of R)} (hf : e ≫ A.hom = f) {H : LB X} (hH : H.IsLB) (hw : LBIso H (L.pull e)) : ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw H N)), (pw H N).IsLB ∧ VeryAmpleBy f (pw H N) s

theorem **polAV_veryAmpleBy** : PolBaseChangeSt → ∀ (S : Scheme.{0}) (P : PolAV S) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ S) (X : Scheme.{0}) (π : X ⟶ P.A.left) (f : X ⟶ Spec (CommRingCat.of R)), IsPullback π f P.A.hom t → ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → LB.Sec (pw (P.L.pull π) N)), (pw (P.L.pull π) N).IsLB ∧ VeryAmpleBy f (pw (P.L.pull π) N) s

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.WeilIsAlternatingSkewNatural

theorem **val_div_eq** {X Y S₀ : Scheme.{0}} {M : LB X} (hM : M.IsLB) {K : Type} [Fintype K] {φ : K → (Y ⟶ X)} {ε : K → ℤ} (p : Y ⟶ S₀) (hp : Function.Surjective p.appTop.hom) {u u' : LB.Gen (LB.LinComb M φ ε)} (hu : u.IsGen) (hu' : u'.IsGen) {Z : Scheme.{0}} (r r' : Z ⟶ Y) (hr : r ≫ p = r' ≫ p) (z : Z) (J) (hJ : Good M z J) : val u r z J * (val u' r z J)⁻¹ = val u r' z J * (val u' r' z J)⁻¹

theorem **surj_fst** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (T : Over S) : Function.Surjective (fst T A).left.appTop.hom

theorem **commIsOf_iff** (hL : L.IsLB) {T : Over S} (θx θy : ThetaGrp A L k T) (c : Γ(T.left, ⊤)ˣ) : CommIsOf A L k θx θy c ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T ⊗ A).left) (z : Z) (J), Good L z J → tval θx r z J * tval θy (r ≫ (translT A θx.pt).left) z J = tval θy r z J * tval θx (r ≫ (translT A θy.pt).left) z J * gm (r ≫ (fst T A).left) z ⊤ c

theorem **translT_fst** {T : Over S} (x : T ⟶ A) : (translT A x).left ≫ (fst T A).left = (fst T A).left

theorem **tval_div** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} (θ θ' : ThetaGrp A L k T) (h : θ.pt = θ'.pt) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) (x : T ⟶ A) (z : Z) (J) (hJ : Good L z J) : tval θ r z J * (tval θ' r z J)⁻¹ = tval θ (r ≫ (translT A x).left) z J * (tval θ' (r ≫ (translT A x).left) z J)⁻¹

theorem **commIsOf_change** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} {θx θy θx' θy' : ThetaGrp A L k T} {c : Γ(T.left, ⊤)ˣ} (hx : θx'.pt = θx.pt) (hy : θy'.pt = θy.pt) (H : CommIsOf A L k θx θy c) : CommIsOf A L k θx' θy' c

theorem **commIsOf_self** (hL : L.IsLB) {T : Over S} (θ : ThetaGrp A L k T) : CommIsOf A L k θ θ 1

theorem **commIsOf_symm** (hL : L.IsLB) {T : Over S} {θx θy : ThetaGrp A L k T} {c : Γ(T.left, ⊤)ˣ} (H : CommIsOf A L k θx θy c) : CommIsOf A L k θy θx c⁻¹

theorem **whisker_translPt** {T T' : Over S} (h : T' ⟶ T) (x : T ⟶ A) : h ▷ A ≫ translPt A x = translPt A (h ≫ x)

theorem **whisker_translT** {T T' : Over S} (h : T' ⟶ T) (x : T ⟶ A) : translT A (h ≫ x) ≫ h ▷ A = h ▷ A ≫ translT A x

theorem **exists_pull** (hL : L.IsLB) {T T' : Over S} (h : T' ⟶ T) (θ : ThetaGrp A L k T) : ∃ θ' : ThetaGrp A L k T', θ'.pt = h ≫ θ.pt ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T' ⊗ A).left) (z : Z) (J), Good L z J → tval θ' r z J = tval θ (r ≫ (h ▷ A).left) z J

theorem **exists_commIsOf_pull** (hL : L.IsLB) {T T' : Over S} (h : T' ⟶ T) {θx θy : ThetaGrp A L k T} {c : Γ(T.left, ⊤)ˣ} (H : CommIsOf A L k θx θy c) : ∃ θx' θy' : ThetaGrp A L k T', θx'.pt = h ≫ θx.pt ∧ θy'.pt = h ≫ θy.pt ∧ CommIsOf A L k θx' θy' (Units.map h.left.appTop.hom.toMonoidHom c)

theorem **weilIs_symm** (hL : L.IsLB) {T : Over S} {x y : T ⟶ A} {c : Γ(T.left, ⊤)ˣ} (H : WeilIs A L k x y c) : WeilIs A L k y x c⁻¹

theorem **weilIs_pull** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T T' : Over S} (h : T' ⟶ T) {x y : T ⟶ A} {c : Γ(T.left, ⊤)ˣ} (H : WeilIs A L k x y c) : WeilIs A L k (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom c)

theorem **weilIs_alternating_skew_natural** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResₗ V ≫ x) → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ x : T ⟶ P.A, x ^ k = 1 → WeilIs P.A P.L k x x 1) ∧ (∀ (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k y x c⁻¹) ∧ ∀ (T' : Over S) (h : T' ⟶ T) (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom c)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.WeilIsIffCommIsOfAndLiftSubsingleton

theorem **weilIs_of_commIsOf** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} {θx θy : ThetaGrp A L k T} {c : Γ(T.left, ⊤)ˣ} (H : CommIsOf A L k θx θy c) : WeilIs A L k θx.pt θy.pt c

theorem **weilIs_iff_commIsOf** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {T : Over S} (θx θy : ThetaGrp A L k T) (c : Γ(T.left, ⊤)ˣ) : WeilIs A L k θx.pt θy.pt c ↔ CommIsOf A L k θx θy c

theorem **val_powGen** {T : Over S} (θ : ThetaGrp A L k T) (n : ℕ) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) {z : Z} {J} (hJ : Good L z J) : val (powGen A L k θ n) r z J = ∏ i : Fin n, tval θ (r ≫ (translT A (θ.pt ^ (i : ℕ)))).left) z J

theorem **val_symGen** {T : Over S} (θ : ThetaGrp A L k T) (ν : LB.Gen (negLB A L k T)) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) {z : Z} {J} (hJ : Good L z J) : val (symGen A L k θ ν) r z J = (tval θ (r ≫ (translT A θ.pt⁻¹).left) z J * nval ν r z J) * (tval θ (r ≫ (negT A T).left) z J * (nval ν (r ≫ (translT A θ.pt⁻¹).left) z J)⁻¹)

theorem **exists_neg_pull** (hL : L.IsLB) {T T' : Over S} (h : T' ⟶ T) (ν : LB.Gen (negLB A L k T)) (hν : ν.IsGen) : ∃ ν' : LB.Gen (negLB A L k T'), ν'.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ (T' ⊗ A).left) (z : Z) (J), Good L z J → nval ν' r z J = nval ν (r ≫ (h ▷ A).left) z J

theorem **whisker_negT** {T T' : Over S} (h : T' ⟶ T) : negT A T' ≫ h ▷ A = h ▷ A ≫ negT A T

theorem **exists_symLift_pull** (hL : L.IsLB) {n : ℕ} {T T' : Over S} (h : T' ⟶ T) {x : T ⟶ A} {θ : ThetaGrp A L k T} (H : SymLift A L k n x θ) : ∃ θ' : ThetaGrp A L k T', SymLift A L k n (h ≫ x) θ'

theorem **exists_symLift_empty** (hL : L.IsLB) (n : ℕ) {T : Over S} [IsEmpty T.left] (x : T ⟶ A) : ∃ θ : ThetaGrp A L k T, SymLift A L k n x θ

theorem **exists_section** {T : Over S} [Subsingleton T.left] {V : T.left.Opens} {p : T.left} (hp : p ∈ V) : ∃ h : T ⟶ overRes V, h ≫ overResι V = 𝟙 T

theorem **weilIs_iff_commIsOf_and_lift_subsingleton** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResι V ≫ x) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ), Odd k → ∀ (T : Over S) (x : T ⟶ P.A), x ^ k = 1 → (∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L k (overRes V), SymLift P.A P.L k k (overResι V ≫ x) θ) ∧ ∀ θ θ' : ThetaGrp P.A P.L k T, SymLift P.A P.L k k x θ → SymLift P.A P.L k k x θ' → LB.Gen.Corr P.L _ _ _ _ θ.φ θ'.φ) → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ (θx θy : ThetaGrp P.A P.L k T) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k θx.pt θy.pt c ↔ CommIsOf P.A P.L k θx θy c) ∧ (∀ x : T ⟶ P.A, x ^ k = 1 → Subsingleton T.left → ∃ θ : ThetaGrp P.A P.L k T, θ.pt = x ∧ (Odd k → SymLift P.A P.L k k x θ))

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.WeilIsValueEq

theorem **comm_eq** {B U : Type} [CommGroup B] [CommGroup U] (v : B → B → B → U) (φ φ' ψ χ : B → U) (k : ℤ) (l : ℕ) (x y : B) (hv : ∀ a, v a x 1 * v a y x = v a y 1 * v a x y) (hA : ∀ w, φ (w * x) * φ 1 * v w y x ^ k = φ w * φ x * v w y 1 ^ k) (hB : χ y * ψ 1 * (∏ i ∈ Finset.range l, v (x ^ i) x 1) ^ k = χ 1 * ψ y * (∏ i ∈ Finset.range l, v (x ^ i) x y) ^ k) (hC : φ' x * φ 1 ^ l = φ' 1 * φ x ^ l) : χ 1 * φ' x * (φ' 1 * χ y)⁻¹ = ψ 1 * φ (x ^ l) * (φ 1 * ψ y)⁻¹

theorem **oneGen_isGen** : (LB.oneGen L (Y

theorem **Rep.one** : Rep L (Y

theorem **Rep.mul** (h : Rep L D f) (h' : Rep L D' f') : Rep L (D + D') (fun Z r z J => f Z r z J * f' Z r z J)

theorem **Rep.inv** (h : Rep L D f) : Rep L (-D) (fun Z r z J => (f Z r z J)⁻¹)

theorem **Rep.congr** (h : Rep L D f) (hD : D = D') (hf : ∀ Z r z J, f Z r z J = f' Z r z J) : Rep L D' f'

theorem **Rep.pow** (h : Rep L D f) (n : ℕ) : Rep L (n • D) (fun Z r z J => f Z r z J ^ n)

theorem **Rep.zpow** (h : Rep L D f) (n : ℤ) : Rep L (n • D) (fun Z r z J => f Z r z J ^ n)

theorem **Rep.theta** {k : ℤ} {T : Over S} (θ : ThetaGrp A L k T) (s : Y ⟶ T) (a : Y ⟶ A) : Rep L (k • pt a - k • pt ((s ≫ θ.pt) * a)) (fun Z r z J => thetaVal θ (r ≫ s) (r ≫ a) z J)

theorem **Rep.cube** [IsCommMonObj A] {t : LB.Gen (Theta A L)} (ht : t.IsGen) (a b c : Y ⟶ A) : Rep L (pt (a * b * c) - pt (a * b) - pt (a * c) - pt (b * c) + pt a + pt b + pt c - pt (1 : Y ⟶ A)) (fun Z r z J => cubeVal t z J (r ≫ a) (r ≫ b) (r ≫ c))

theorem **Rep.cubeProd** [IsCommMonObj A] {t : LB.Gen (Theta A L)} (ht : t.IsGen) (x c : Y ⟶ A) (n : ℕ) : Rep L (pt (x ^ n * c) - pt c - pt (x ^ n) + pt (1 : Y ⟶ A) + n • (pt x + pt c - pt (x * c) - pt (1 : Y ⟶ A))) (fun Z r z J => ∏ i ∈ Finset.range n, cubeVal t z J ((r ≫ x) ^ i) (r ≫ x) (r ≫ c))

theorem **Rep.rigid** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {Z : Over S} {D : FreeAbelianGroup ((Z ⊗ A).left ⟶ A.left)} {f : ValFam L (Z ⊗ A)} (h : Rep L D f) (hD : D = 0) (a : Z ⟶ A) {z : Z.left} {J} (hJ : Good L z J) : f Z (lift (𝟙 Z) a) z J = f Z (lift (𝟙 Z) 1) z J

theorem **commVal_eq** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {t : LB.Gen (Theta A L)} (ht : t.IsGen) {k : ℤ} {l : ℕ} {T T' Z : Over S} (χ φ' : ThetaGrp A L (k * l) T) (ψ φ : ThetaGrp A L k T') (s : Z ⟶ T) (s' : Z ⟶ T') (hx : s' ≫ ψ.pt = (s ≫ χ.pt) ^ l) (hy : s' ≫ φ.pt = s ≫ φ'.pt) {z : Z.left} {J} (hJ : Good L z J) : commVal χ φ' s 1 z J = commVal ψ φ s' 1 z J

theorem **weilIs_value_eq** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {t : LB.Gen (Theta A L)} (ht : t.IsGen) {k : ℤ} {l : ℕ} {T : Over S} {x y : T ⟶ A} {c c' : Γ(T.left, ⊤)ˣ} (h : WeilIs A L (k * l) x y c) (h' : WeilIs A L k (x ^ l) y c') : c = c'

theorem **weilIs_mul_iff** : GlobalFnsSt → (∀ (S : Scheme.{0}) (P : PolAV S) (M : LB P.A.left), M.IsLB → ∃ t : LB.Gen (Theta P.A M), t.IsGen) → (∀ (S : Scheme.{0}) (P : PolAV S) (N : ℕ) (T : Over S) (x : T ⟶ P.A), x ^ N = 1 → ∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ θ : ThetaGrp P.A P.L N (overRes V), θ.pt = overResι V ≫ x) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (x y : T ⟶ P.A), x ^ k = 1 → y ^ k = 1 → ∃! c : Γ(T.left, ⊤)ˣ, WeilIs P.A P.L k x y c) → ∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (l : ℕ) (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), IsUnit ((k * l : ℕ) : Γ(T.left, ⊤)) → x ^ (k * l) = 1 → y ^ k = 1 → (WeilIs P.A P.L (k * l) x y c ↔ WeilIs P.A P.L k (x ^ l) y c)

Development.AbelianVarietyTheory.Proofs.AbelianSchemeLawProofs.WeilStOfParts

theorem **weilSt_of_parts** : (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (x y : T ⟶ P.A), x ^ k = 1 → y ^ k = 1 → ∃! c : Γ(T.left, ⊤)ˣ, WeilIs P.A P.L k x y c) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (x x' y : T ⟶ P.A) (c c' : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k x' y c' → WeilIs P.A P.L k (x * x') y (c * c')) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S), (∀ x : T ⟶ P.A, x ^ k = 1 → WeilIs P.A P.L k x x 1) ∧ (∀ (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k y x c⁻¹) ∧ ∀ (T' : Over S) (h : T' ⟶ T) (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), WeilIs P.A P.L k x y c → WeilIs P.A P.L k (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom c)) → (∀ (S : Scheme.{0}) (P : PolAV S) (k : ℕ) (T : Over S) (l : ℕ) (x y : T ⟶ P.A) (c : Γ(T.left, ⊤)ˣ), IsUnit ((k * l : ℕ) : Γ(T.left, ⊤)) → x ^ (k * l) = 1 → y ^ k = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AddCharTrivialIntegralFiniteAdeleMul

theorem **mem_of_isClosed** (F : Type) [Field F] [NumberField F] {C : Set (FiniteAdeleRing (𝒪 F) F)} (hC : IsClosed C) (h : ∀ a : 𝒪 F, algebraMap F (FiniteAdeleRing (𝒪 F) F) (a : F) ∈ C) (α : FiniteAdeleRing (𝒪 F) F) (hα : ∀ v, α v ∈ v.adicCompletionIntegers F) : α ∈ C

theorem **addChar_eq_one_of_integral** : ∀ (F : Type) [Field F] [NumberField F] (ψ : AddChar (AF F) ℂ), Continuous ψ → ∀ x : F, (∀ a : 𝒪 F, ψ (pairA F 0 (algebraMap F (Af F) ((a : F) * x))) = 1) → ∀ α ∈ intAf F, ψ (pairA F 0 (α * algebraMap F (Af F) x)) = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AddCharTrivialIntegralFiniteAdeleMulStatement

theorem **_root_.GlobalTheory.T.addChar_trivial_integralFiniteAdele_mul** : ∀ (F : Type) [Field F] [NumberField F] (ψ : AddChar (AF F) ℂ), Continuous ψ → ∀ x : F, (∀ a : 𝒪 F, ψ (pairA F 0 (algebraMap F (Af F) ((a : F) * x))) = 1) → ∀ α ∈ intAf F, ψ (pairA F 0 (α * algebraMap F (Af F) x)) = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AddNsmulResidueAssoc

theorem **exists_finite_mod** {W : Type} [AddCommGroup W] {G : Set W} (hG : G.Finite) {k : ℕ} (hk : 0 < k) : ∃ R : Set W, R.Finite ∧ ∀ x ∈ Submodule.span ℤ G, ∃ r ∈ R, ∃ u ∈ Submodule.span ℤ G, x = r + k • u

theorem **exists_int_gen** {E M : Type} [Field E] [NumberField E] [AddCommGroup M] [Module (𝒪 E) M] (N : Submodule (𝒪 E) M) (hN : N.FG) : ∃ A : Set M, A.Finite ∧ A ⊆ N ∧ ∀ x ∈ N, x ∈ Submodule.span ℤ A

theorem **triadd_mem_span** {X Y Z W : Type} [AddCommGroup X] [AddCommGroup Y] [AddCommGroup Z] [AddCommGroup W] (f : X → Y → Z → W) (h1 : ∀ a a' s t, f (a + a') s t = f a s t + f a' s t) (h2 : ∀ a s s' t, f a (s + s') t = f a s t + f a s' t) (h3 : ∀ a s t t', f a s (t + t') = f a s t + f a s t') {A : Set X} {B : Set Y} {C : Set Z} {a s t} (ha : a ∈ Submodule.span ℤ A) (hs : s ∈ Submodule.span ℤ B) (ht : t ∈ Submodule.span ℤ C) : f a s t ∈ Submodule.span ℤ ((fun p : X × Y × Z => f p.1 p.2.1 p.2.2) '' (A ×ˢ B ×ˢ C))

theorem **star_sum_add** {E : Type} [CommRing E] [StarRing E] {n : ℕ} (c : E) (s s' g : Fin n → E) : c * ∑ k, star ((s + s') k) * g k = c * ∑ k, star (s k) * g k + c * ∑ k, star (s' k) * g k

theorem **pair_tmul_split** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (ε β : Af F) (x : F) (k : ℕ) (a t g : E) : pairA F 0 (ε * (algebraMap F _ x + k • β)) ⊗ₜ[F] (a * t * g) = pairA F 0 ε ⊗ₜ[F] (algebraMap F E x * a * t * g) + k • (pairA F 0 (ε * β) ⊗ₜ[F] (a * t * g))

theorem **add_nsmul_residue_add** {W : Type} [AddCommGroup W] (k : ℕ) (y u y' u' : W) : y + k • u + (y' + k • u') = y + y' + k • (u + u')

theorem **neg_nsmul_residue** {W : Type} [AddCommGroup W] (k : ℕ) (y u : W) : -(y + k • u) = -y + k • -u

theorem **add_nsmul_residue_assoc** {W : Type} [AddCommGroup W] (k : ℕ) (r u' u : W) : r + k • u' + k • u = r + k • (u' + u)

theorem **genMap_mem_span** (S T : HSp D) {A : Set D.host.E} {B : Set (Fin S.rank → D.host.E)} {C : Set (Fin T.rank → D.host.E)} {a s t} (ha : a ∈ Submodule.span ℤ A) (hs : s ∈ Submodule.span ℤ B) (ht : t ∈ Submodule.span ℤ C) : genMap S T a s t ∈ Submodule.span ℤ ((fun p : _ × _ × _ => genMap S T p.1 p.2.1 p.2.2) '' (A ×ˢ B ×ˢ C))

theorem **evalV_idemAt** (F : Type) [Field F] [NumberField F] (P : HeightOneSpectrum (𝒪 F) → Prop) (v : HeightOneSpectrum (𝒪 F)) : evalV F v (idemAt F P) = if P v then 1 else 0

theorem **latO_finite_mod** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), ΛS.FG → ΛT.FG → ∀ (P : HeightOneSpectrum (𝒪 D.host.F) → Prop) (k : ℕ), 0 < k → ∃ R : Set (WO S T), R.Finite ∧ ∀ t ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) P, ∃ r ∈ R, ∃ u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) P, t = r + k • u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AdeleIdemPrimeCriterion

private lemma **one_lt_valued_inv_iff_mem** {E : Type} [Field E] [NumberField E] (Q : HeightOneSpectrum (𝒪 E)) (x : 𝒪 E) (hx : x ≠ 0) : 1 < Valued.v ((algebraMap E (FiniteAdeleRing (𝒪 E) E) (x : E)⁻¹) Q) ↔ x ∈ Q.asIdeal

theorem **_root_.GlobalTheory.adele_idem_prime_criterion** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (Φ : _root_.HeckeCharacterZeroLocus.ThmBIn.AE F E ≃+* (mixedSpace E × FiniteAdeleRing (𝒪 E) E)), (∀ x : E, Φ (_root_.HeckeCharacterZeroLocus.ThmBIn.incA F E x) = (mixedEmbedding E x, algebraMap E _ x)) → (∀ y, _root_.HeckeCharacterZeroLocus.ThmBIn.finPart F E y ∈ _root_.HeckeCharacterZeroLocus.ThmBIn.intAE F E ↔ ∀ P, Valued.v ((Φ y).2 P) ≤ 1) → (∀ v ε, _root_.HeckeCharacterZeroLocus.ThmBIn.IsPlaceIdem F v ε → (Φ (algebraMap (_root_.HeckeCharacterZeroLocus.ThmBIn.AF F) (_root_.HeckeCharacterZeroLocus.ThmBIn.AE F E) ε)).1 = 0 ∧ ∀ P : HeightOneSpectrum (𝒪 E), (Φ (algebraMap (_root_.HeckeCharacterZeroLocus.ThmBIn.AF F) (_root_.HeckeCharacterZeroLocus.ThmBIn.AE F E) ε)).2 P = if P.under (𝒪 F) = v then 1 else 0) → ∀ (e : _root_.HeckeCharacterZeroLocus.ThmBIn

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AdelicPairReducedBasis

theorem **intClosure_le** (O : Subring A) {a : A} (ha : a ∈ intClosure O) : a ∈ O

theorem **mem_ratSpan** {Z : Subring A} {a : A} (ha : a ∈ Z) : a ∈ ratSpan Z

theorem **den_mul** (φ : ℚ →+* A) (q : ℚ) : (q.den : A) * φ q = (q.num : A)

theorem **rat_mem** (φ : ℚ →+* A) (Z : Subring A) (q : ℚ) : φ q ∈ ratSpan Z

theorem **inv_mul** (φ : ℚ →+* A) {N : ℕ} (hN : 0 < N) : φ (N : ℚ)⁻¹ * (N : A) = 1

theorem **cancel** (φ : ℚ →+* A) {N : ℕ} (hN : 0 < N) {a b : A} (h : (N : A) * a = N * b) : a = b

theorem **mem_of_mul** (φ : ℚ →+* A) (O : Subring A) (hint : ∀ (z : ℤ) (N : ℕ), 0 < N → ∀ e ∈ O, (z : A) = N * e → (N : ℤ) ∣ z) {N : ℕ} (hN : 0 < N) {c : A} (hc : c ∈ O) (h : (N : A) * c ∈ intClosure O) : c ∈ intClosure O

theorem **exists_basis** (φ : ℚ →+* A) (O : Subring A) [IsReduced A] {β : Type} [Fintype β] (w : β → A) (hw : ∀ i, w i ∈ O) (hint : ∀ (z : ℤ) (N : ℕ), 0 < N → ∀ e ∈ O, (z : A) = N * e → (N : ℤ) ∣ z) (h0 : ∀ a : A, (∀ M : ℕ, 0 < M → ∃ c, c ∈ O ∧ a = (M : A) * c) → a = 0) (hden : ∀ a : A, ∃ M : ℕ, 0 < M ∧ (M : A) * a ∈ O) (hspan : ∀ a ∈ O, ∃ r : β → A, (∀ i, r i ∈ intClosure O) ∧ a = ∑ i, r i * w i) (hdiv : ∀ (r : β → A) (N : ℕ), 0 < N → (∀ i, r i ∈ intClosure O) → (∃ e, e ∈ O ∧ ∑ i, r i * w i = (N : A) * e) → ∀ i, ∃ s, s ∈ intClosure O ∧ r i = (N : A) * s) : ∃ (R : Type) (_ : CommRing R) (_ : Algebra ℚ R) (R₀ : Subring R) (_ : Algebra R A) (b : Module.Basis β R A), AdelicPair R R₀ ∧ IsReduced R ∧ (∀ i, b i = w i) ∧ ∀ a : A, a ∈ O ↔ ∀ i, b.repr a i ∈ R₀

theorem **finiteAdele_free** : ∀ (F : Type) [Field F] [NumberField F] (β : Type) [Fintype β] (ω : Module.Basis β ℤ (𝒪 F)), ∃ (R : Type) (_ : CommRing R) (_ : Algebra ℚ R) (R₀ : Subring R) (_ : Algebra R (FiniteAdeleRing (𝒪 F) F)) (b : Module.Basis β R (FiniteAdeleRing (𝒪 F) F)), AdelicPair R R₀ ∧ IsReduced R ∧ (∀ i, b i = algebraMap F (FiniteAdeleRing (𝒪 F) F) (ω i : 𝒪 F)) ∧ ∀ a : FiniteAdeleRing (𝒪 F) F, (∀ v, (a : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ↔ ∀ i, b.repr a i ∈ R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AdelicPairReducedBasisStatement

theorem **_root_.GlobalTheory.T.adelic_pair_reduced_basis** : ∀ (F : Type) [Field F] [NumberField F] (β : Type) [Fintype β] (ω : Module.Basis β ℤ (𝒪 F)), ∃ (R : Type) (_ : CommRing R) (_ : Algebra ℚ R) (R₀ : Subring R) (_ : Algebra R (FiniteAdeleRing (𝒪 F) F)) (b : Module.Basis β R (FiniteAdeleRing (𝒪 F) F)), AdelicPair R R₀ ∧ IsReduced R ∧ (∀ i, b i = algebraMap F (FiniteAdeleRing (𝒪 F) F) (ω i : 𝒪 F)) ∧ ∀ a : FiniteAdeleRing (𝒪 F) F, (∀ v, (a : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ↔ ∀ i, b.repr a i ∈ R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AdelicUnitaryScalingUnit

theorem **finite_det_one_add_smul_eq_zero** {E K : Type} [Field E] [Field K] (φ : E →+* K) {n : ℕ} (G : Matrix (Fin n) (Fin n) K) : {l : E | (1 + φ l • G).det = 0}.Finite

theorem **finite_not_isUnit_det** {E R : Type} [Field E] [CommRing R] (φ : E →+* R) {n : ℕ} (g : Matrix (Fin n) (Fin n) R) (ε : R) (hε : ε * ε = ε) (hfin : {M : Ideal R | M.IsMaximal ∧ 1 - ε ∈ M}.Finite) : {l : E | ¬ IsUnit (1 + φ l • (ε • g + (1 - ε) • (1 : Matrix (Fin n) (Fin n) R))).det}.Finite

theorem **adEval_surjective** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : Function.Surjective (adEval F v)

theorem **finite_primes_one_sub** (F : Type) [Field F] [NumberField F] (e : FiniteAdeleRing (𝒪 F) F) (he : {v | e v ≠ 0}.Finite) : {P : Ideal (FiniteAdeleRing (𝒪 F) F) | P.IsPrime ∧ 1 - e ∈ P}.Finite

theorem **infinite_norm_one** {F E : Type} [Field F] [Field E] [CharZero E] [Algebra F E] (c : E ≃ₐ[F] E) (hc : ∀ x, c (c x) = x) (hc1 : c ≠ 1) : {l : E | l * c l = 1}.Infinite

theorem **exists_norm_one_isUnit_det** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E), Module.finrank F E = 2 → (∀ x, c (c x) = x) → c ≠ 1 → ∀ (n : ℕ) (g : Matrix (Fin n) (Fin n) (RA F E)) (e : FiniteAdeleRing (𝒪 F) F), e * e = e → {v | e v ≠ 0}.Finite → ∃ lam : E, lam * c lam = 1 ∧ IsUnit (1 + ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] lam) • ((e ⊗ₜ[F] (1 : E)) • g + ((1 - e) ⊗ₜ[F] (1 : E)) • (1 : Matrix (Fin n) (Fin n) (RA F E)))).det

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AdelicUnitaryScalingUnitStatement

theorem **_root_.GlobalTheory.T.adelic_unitary_scaling_unit** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E), Module.finrank F E = 2 → (∀ x, c (c x) = x) → c ≠ 1 → ∀ (n : ℕ) (g : Matrix (Fin n) (Fin n) (RA F E)) (e : FiniteAdeleRing (𝒪 F) F), e * e = e → {v | e v ≠ 0}.Finite → ∃ lam : E, lam * c lam = 1 ∧ IsUnit (1 + ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] lam) • ((e ⊗ₜ[F] (1 : E)) • g + ((1 - e) ⊗ₜ[F] (1 : E)) • (1 : Matrix (Fin n) (Fin n) (RA F E)))).det

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AevalSumElimEqZeroOfMemPts

theorem **tensor_eq_zero_of_forall_contract** {V W : Type} [AddCommGroup V] [Module ℚ V] [AddCommGroup W] [Module ℚ W] (t : V ⊗[ℚ] W) (h : ∀ m : W →ₗ[ℚ] ℚ, TensorProduct.rid ℚ V (LinearMap.lTensor V m t) = 0) : t = 0

theorem **aeval_sum_elim_eq_zero_of_tensor** {ι : Type} [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (hΓ : ∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) (t : MvPolynomial (ι × ι) ℚ ⊗[ℚ] MvPolynomial (ι × ι) ℚ) (ht : ∀ g ∈ Γ, ∀ h ∈ Γ, Algebra.TensorProduct.productMap (MvPolynomial.aeval (fun ij : ι × ι => g ij.1 ij.2)) (MvPolynomial.aeval (fun ij : ι × ι => h ij.1 ij.2)) t = (0 : ℂ)) (R : Type) [CommRing R] [Algebra ℚ R] (x : Matrix ι ι R) (hx : x ∈ pts Γ R) (y : Matrix ι ι R) (hy : y ∈ pts Γ R) : Algebra.TensorProduct.productMap (MvPolynomial.aeval (fun ij : ι × ι => x ij.1 ij.2)) (MvPolynomial.aeval (fun ij : ι × ι => y ij.1 ij.2)) t = (0 : R)

theorem **aeval_sum_elim_eq_zero_of_mem_pts** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ F : MvPolynomial ((ι × ι) ⊕ (ι × ι)) ℚ, (∀ g ∈ Γ, ∀ h ∈ Γ, MvPolynomial.aeval (Sum.elim (fun ij : ι × ι => g ij.1 ij.2) (fun ij : ι × ι => h ij.1 ij.2)) F = (0 : ℂ)) → ∀ (R : Type) [CommRing R] [Algebra ℚ R], ∀ x ∈ pts Γ R, ∀ y ∈ pts Γ R, MvPolynomial.aeval (Sum.elim (fun ij : ι × ι => x ij.1 ij.2) (fun ij : ι × ι => y ij.1 ij.2)) F = (0 : R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AevalSumElimEqZeroOfMemPtsStatement

theorem **_root_.GlobalTheory.T.aeval_sum_elim_eq_zero_of_mem_pts'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ F : MvPolynomial ((ι × ι) ⊕ (ι × ι)) ℚ, (∀ g ∈ Γ, ∀ h ∈ Γ, MvPolynomial.aeval (Sum.elim (fun ij : ι × ι => g ij.1 ij.2) (fun ij : ι × ι => h ij.1 ij.2)) F = (0 : ℂ)) → ∀ (R : Type) [CommRing R] [Algebra ℚ R], ∀ x ∈ pts Γ R, ∀ y ∈ pts Γ R, MvPolynomial.aeval (Sum.elim (fun ij : ι × ι => x ij.1 ij.2) (fun ij : ι × ι => y ij.1 ij.2)) F = (0 : R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AlgebraMapNormTensorEqProdConjugates

theorem **_root_.GlobalTheory.CNorm.algebraMap_norm_tensor_eq_prod_conjugates** (F E K : Type) [Field F] [Field E] [Algebra F E] [FiniteDimensional F E] [IsGalois F E] [CommRing K] [Algebra F K] (y : K ⊗[F] E) : algebraMap K (K ⊗[F] E) (Algebra.norm K y) = ∏ g : E ≃ₐ[F] E, Algebra.TensorProduct.map (AlgHom.id K K) (g : E →ₐ[F] E) y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.AlgebraMapNormTensorEqProdConjugatesStatement

theorem **_root_.GlobalTheory.T.algebraMap_norm_tensor_eq_prod_galois_conjugates** : ∀ (F E K : Type) [Field F] [Field E] [Algebra F E] [FiniteDimensional F E] [IsGalois F E] [CommRing K] [Algebra F K] (y : K ⊗[F] E), algebraMap K (K ⊗[F] E) (Algebra.norm K y) = ∏ g : E ≃ₐ[F] E, Algebra.TensorProduct.map (AlgHom.id K K) (g : E →ₐ[F] E) y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ApplyAddInfRat

theorem **apply_real_add** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {u : WO S T} (hu : u.map (infPart D) = 0) {f : WO S T → ℂ} {ε : ℂ} (hf : heisO ψ S T u f = ε • f) (z : RW T S) : f (iotaW σ T S z + u) = ε * f (iotaW σ T S z)

theorem **exists_decomp** (h : Chart θ σ) {Λ : AddSubgroup (WO S T)} (h2 : ∀ γ : WO S T, γ.map (infPart D) = 0 → ∃ ρ ∈ ratL T S, ∃ u ∈ Λ, (γ - ρ - u).map (finPart D) = 0) (w : WO S T) : ∃ γ z, ∃ u ∈ Λ, w = ratW T S γ + (iotaW σ T S z + u)

theorem **apply_add_infRat** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (v : Matrix (Fin T.rank) (Fin S.rank) D.host.E) {ε : ℂ} (he : heisO ψ S T (finRat S T (-v)) f = ε • f) (z : RW T S) : f (iotaW σ T S (z + infRat S T v)) = (ψ (-symplOH S T (ratW T S v) (finRat S T (-v))) * ε) * Complex.exp (-(Real.pi * Complex.I) * (-omO S T (infRat S T v) z : ℝ)) * f (iotaW σ T S z)

theorem **infRat_injective** (S T : HSp D) : Function.Injective (infRat S T)

theorem **vacUniqueO** : ∀ D : D7, VacUniqueO D

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.BinaryFormUniversal

theorem **binary_form_universal** {k : Type u_1} [Field k] [Finite k] {a n : k} (hD : a ^ (2 : ℕ) - (4 : k) * n ≠ (0 : k)) (u : k) : ∃ (r : k) (s : k), r ^ (2 : ℕ) + a * r * s + n * s ^ (2 : ℕ) = u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.BlockDiagPartOfRatIdempotentOneAndMulOfTriangula

theorem **peirce_block_decomposition** {A : Type*} [Ring A] (E F : A) (hE : E * E = E) (hF : F * F = F) (hEF : E * F = 0) (hFE : F * E = 0) (h1 : E + F = 1) : E * 1 * E + F * 1 * F = 1 ∧ (∀ x y : A, F * x * E = 0 → F * y * E = 0 → E * (x * y) * E + F * (x * y) * F = (E * x * E + F * x * F) * (E * y * E + F * y * F) ∧ F * (x * y) * E = 0) ∧ (∀ x : A, F * x * E = 0 → x = (E * x * E + F * x * F) + E * x * F) ∧ (∀ x : A, E * (E * x * E + F * x * F) * E + F * (E * x * E + F * x * F) * F = E * x * E + F * x * F)

theorem **blk_mul_of_triangular** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (e : Matrix ι ι ℚ) (R : Type) [CommRing R] [Algebra ℚ R], e * e = e → blk e (1 : Matrix ι ι R) = 1 ∧ (∀ x y : Matrix ι ι R, (1 - e.map (algebraMap ℚ R)) * x * e.map (algebraMap ℚ R) = 0 → (1 - e.map (algebraMap ℚ R)) * y * e.map (algebraMap ℚ R) = 0 → blk e (x * y) = blk e x * blk e y ∧ (1 - e.map (algebraMap ℚ R)) * (x * y) * e.map (algebraMap ℚ R) = 0) ∧ (∀ x : Matrix ι ι R, (1 - e.map (algebraMap ℚ R)) * x * e.map (algebraMap ℚ R) = 0 → x = blk e x + e.map (algebraMap ℚ R) * x * (1 - e.map (algebraMap ℚ R))) ∧ (∀ x : Matrix ι ι R, blk e (blk e x) = blk e x) ∧ ∀ γ : Matrix ι ι ℚ, blk e (γ.map (algebraMap ℚ R)) = (blk e γ).map (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.BlockDiagPartOfRatIdempotentOneAndMulOfTriangulaStatement

theorem **_root_.GlobalTheory.T.blockDiagPartOfRatIdempotent_one_and_mul_of_triangular** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (e : Matrix ι ι ℚ) (R : Type) [CommRing R] [Algebra ℚ R], e * e = e → blk e (1 : Matrix ι ι R) = 1 ∧ (∀ x y : Matrix ι ι R, (1 - e.map (algebraMap ℚ R)) * x * e.map (algebraMap ℚ R) = 0 → (1 - e.map (algebraMap ℚ R)) * y * e.map (algebraMap ℚ R) = 0 → blk e (x * y) = blk e x * blk e y ∧ (1 - e.map (algebraMap ℚ R)) * (x * y) * e.map (algebraMap ℚ R) = 0) ∧ (∀ x : Matrix ι ι R, (1 - e.map (algebraMap ℚ R)) * x * e.map (algebraMap ℚ R) = 0 → x = blk e x + e.map (algebraMap ℚ R) * x * (1 - e.map (algebraMap ℚ R))) ∧ (∀ x : Matrix ι ι R, blk e (blk e x) = blk e x) ∧ ∀ γ : Matrix ι ι ℚ, blk e (γ.map (algebraMap ℚ R)) = (blk e γ).map (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.BorelHarishChandra

theorem **Bdd.cont** {F : T → Matrix ι ι ℂ} (hF : Bdd F) {f : Matrix ι ι ℂ → ℝ} (hf : Continuous f) : ∃ C : ℝ, ∀ t, f (F t) ≤ C

theorem **Bdd.comp** {F : T → Matrix ι ι ℂ} (hF : Bdd F) {f : Matrix ι ι ℂ → Matrix ι ι ℂ} (hf : Continuous f) : Bdd fun t => f (F t)

theorem **Bdd.mul** {F G : T → Matrix ι ι ℂ} (hF : Bdd F) (hG : Bdd G) : Bdd fun t => F t * G t

theorem **Bdd.adj** {F : T → Matrix ι ι ℂ} (hF : Bdd F) : Bdd fun t => (F t).adjugate

theorem **Bdd.ct** {F : T → Matrix ι ι ℂ} (hF : Bdd F) : Bdd fun t => (F t)ᴴ

theorem **bdd_unitary** {F : T → Matrix ι ι ℂ} (hF : ∀ t, (F t)ᴴ * F t = 1) : Bdd F

theorem **Bdd.nsq_act** {m : ℕ} (d : Fin m → ℕ) (v : TIdx ι d → ℂ) {F : T → Matrix ι ι ℂ} (hF : Bdd F) : ∃ C : ℝ, ∀ t, nsq (act d v (F t)) ≤ C

theorem **sq_le_nsq** {κ : Type} [Fintype κ] (v : κ → ℂ) (k : κ) : ‖v k‖ ^ 2 ≤ nsq v

theorem **core** {m : ℕ} (d : Fin m → ℕ) (v : TIdx ι d → ℂ) (h1 : ∀ C : ℝ, ∃ C' : ℝ, ∀ y : Matrix ι ι ℂ, y.det = 1 → nsq (act d v y) ≤ C → ∃ h : Matrix ι ι ℂ, h.det = 1 ∧ act d v h = v ∧ ∀ i j, ‖(h * y) i j‖ ≤ C') (h2 : ∀ g : Matrix ι ι ℂ, g.det = 1 → act d v g = v → act d v gᴴ = v) (c : ℝ) (hc : 0 < c) (K N D : T → Matrix ι ι ℂ) (e : T → ι → ℂ) (hK : ∀ t, (K t)ᴴ * K t = 1) (hKd : ∀ t, (K t).det = 1) (hNd : ∀ t, (N t).det = 1) (hDd : ∀ t, (D t).det = 1) (hDh : ∀ t, (D t)ᴴ = D t) (hDe : ∀ t, D t * diagonal (e t) = 1) (hN : Bdd N) (hN1 : Bdd fun t => D t * N t * diagonal (e t)) (hN2 : Bdd fun t => D t * (D t * N t * diagonal (e t)) * diagonal (e t)) (hw : ∀ t j, act d v (K t * D t * N t) j = 0 ∨ c ≤ ‖act d v (K t * D t * N t) j‖) : ∃ C' : ℝ, ∀ t, nsq (act d v (K t * D t * N t)) ≤ C'

theorem **borel_harishChandra** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] [LinearOrder ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ), (∀ C : ℝ, ∃ C' : ℝ, ∀ y : Matrix ι ι ℂ, y.det = 1 → nsq (act d v y) ≤ C → ∃ h : Matrix ι ι ℂ, h.det = 1 ∧ act d v h = v ∧ ∀ i j, ‖(h * y) i j‖ ≤ C') → (∀ g : Matrix ι ι ℂ, g.det = 1 → act d v g = v → act d v gᴴ = v) → ∀ C c : ℝ, 0 < c → ∃ C' : ℝ, ∀ (k : Matrix ι ι ℝ) (a : ι → ℝ) (n : Matrix ι ι ℝ), IsSiegel C k a n → (∀ j, act d v ((k * diagonal a * n).map (fun r : ℝ => (r : ℂ))) j = 0 ∨ c ≤ ‖act d v ((k * diagonal a * n).map (fun r : ℝ => (r : ℂ))) j‖) → nsq (act d v ((k * diagonal a * n).map (fun r : ℝ => (r : ℂ)))) ≤ C'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.BorelHarishChandraStatement

theorem **_root_.GlobalTheory.T.nsq_act_le_of_isSiegel_of_entries_zero_or_ge** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] [LinearOrder ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ), (∀ C : ℝ, ∃ C' : ℝ, ∀ y : Matrix ι ι ℂ, y.det = 1 → nsq (act d v y) ≤ C → ∃ h : Matrix ι ι ℂ, h.det = 1 ∧ act d v h = v ∧ ∀ i j, ‖(h * y) i j‖ ≤ C') → (∀ g : Matrix ι ι ℂ, g.det = 1 → act d v g = v → act d v gᴴ = v) → ∀ C c : ℝ, 0 < c → ∃ C' : ℝ, ∀ (k : Matrix ι ι ℝ) (a : ι → ℝ) (n : Matrix ι ι ℝ), IsSiegel C k a n → (∀ j, act d v ((k * diagonal a * n).map (fun r : ℝ => (r : ℂ))) j = 0 ∨ c ≤ ‖act d v ((k * diagonal a * n).map (fun r : ℝ => (r : ℂ))) j‖) → nsq (act d v ((k * diagonal a * n).map (fun r : ℝ => (r : ℂ)))) ≤ C'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.BruhatFunctionExistsClosedSubgroup

private lemma **exists_nonneg_continuous_ne_zero_on_cosets** {G : Type*} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [T2Space G] (H : Subgroup G) (hH : IsClosed (H : Set G)) : ∃ f : G → ℝ, Continuous f ∧ (∀ x, 0 ≤ f x) ∧ (∀ K : Set G, IsCompact K → IsCompact (tsupport f ∩ (K * (H : Set G)))) ∧ ∀ x, ∃ h : H, f (x * h) ≠ 0

private lemma **exists_bruhat_function_of_positive_on_cosets** {G : Type*} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (H : Subgroup G) (hH : IsClosed (H : Set G)) (μ : Measure H) [μ.IsHaarMeasure] (f : G → ℝ) (hf : Continuous f) (hf0 : ∀ x, 0 ≤ f x) (hfK : ∀ K : Set G, IsCompact K → IsCompact (tsupport f ∩ (K * (H : Set G)))) (hpos : ∀ x, ∃ h : H, f (x * h) ≠ 0) : ∃ β : G → ℝ, Continuous β ∧ (∀ x, 0 ≤ β x) ∧ (∀ K : Set G, IsCompact K → IsCompact (tsupport β ∩ (K * (H : Set G)))) ∧ ∀ x, ∫ h : H, β (x * h) ∂μ = 1

theorem **_root_.GlobalTheory.bruhatFunctionExists_closedSubgroup** : _root_.HeckeCharacterZeroLocus.BruhatFunctionExistsVariant.{u}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CardQuotSpanAlgebraMap

theorem **card_quot_span_algebraMap** : ∀ (R O : Type) [CommRing R] [Nontrivial R] [CommRing O] [Algebra R O] [Module.Free R O] [Module.Finite R O] (t : R), Nat.card (O ⧸ Ideal.span {algebraMap R O t}) = Nat.card (R ⧸ Ideal.span {t}) ^ Module.finrank R O

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CardRingHomRealEqNrRealPlaces

theorem **card_ringHom_real_eq_nrRealPlaces** : ∀ (F : Type) [Field F] [NumberField F], Nat.card (F →+* ℝ) = InfinitePlace.nrRealPlaces F

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CardRingHomRealEqNrRealPlacesStatement

theorem **_root_.GlobalTheory.T.card_ringHom_real_eq_nrRealPlaces'** : ∀ (F : Type) [Field F] [NumberField F], Nat.card (F →+* ℝ) = InfinitePlace.nrRealPlaces F

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CayleyTransformUnitarySkewInvolutive

theorem **map_mul_map_inv** (σ : R →+* R) (M : Matrix (Fin n) (Fin n) R) (h : IsUnit M.det) : M.map σ * (M⁻¹).map σ = 1

theorem **map_one_add** (σ : R →+* R) (M : Matrix (Fin n) (Fin n) R) : (1 + M).map σ = 1 + M.map σ

theorem **map_one_sub** (σ : R →+* R) (M : Matrix (Fin n) (Fin n) R) : (1 - M).map σ = 1 - M.map σ

theorem **one_add_cayley** (g : Matrix (Fin n) (Fin n) R) (h : IsUnit (1 + g).det) : 1 + (1 - g) * (1 + g)⁻¹ = (2 : R) • (1 + g)⁻¹

theorem **one_sub_cayley** (g : Matrix (Fin n) (Fin n) R) (h : IsUnit (1 + g).det) : 1 - (1 - g) * (1 + g)⁻¹ = (2 : R) • (g * (1 + g)⁻¹)

theorem **one_add_cayley_mul** (g : Matrix (Fin n) (Fin n) R) (h : IsUnit (1 + g).det) (c : R) (hc : 2 * c = 1) : (1 + (1 - g) * (1 + g)⁻¹) * (c • (1 + g)) = 1

private theorem **eq_zero_of_mul_mul_eq_zero_of_isUnit_det** (A B X : Matrix (Fin n) (Fin n) R) (hA : IsUnit A.det) (hB : IsUnit B.det) (h : A * X * B = 0) : X = 0

private theorem **isUnit_det_map_of_isUnit_det** (σ : R →+* R) (M : Matrix (Fin n) (Fin n) R) (h : IsUnit M.det) : IsUnit (M.map σ).det

theorem **cayley_transform** : ∀ (R : Type) [CommRing R] (σ : R →+* R) (n : ℕ) (G : Matrix (Fin n) (Fin n) R), (∀ x, σ (σ x) = x) → (G.map σ)ᵀ = G → IsUnit G.det → IsUnit (2 : R) → (∀ g : Matrix (Fin n) (Fin n) R, gᵀ * G * g.map σ = G → IsUnit (1 + g).det → ((1 - g) * (1 + g)⁻¹)ᵀ * G + G * ((1 - g) * (1 + g)⁻¹).map σ = 0 ∧ IsUnit (1 + (1 - g) * (1 + g)⁻¹ : Matrix (Fin n) (Fin n) R).det) ∧ (∀ g : Matrix (Fin n) (Fin n) R, IsUnit (1 + g).det → (1 - (1 - g) * (1 + g)⁻¹) * (1 + (1 - g) * (1 + g)⁻¹)⁻¹ = g) ∧ (∀ S : Matrix (Fin n) (Fin n) R, Sᵀ * G + G * S.map σ = 0 → IsUnit (1 + S).det → ((1 - S) * (1 + S)⁻¹)ᵀ * G * ((1 - S) * (1 + S)⁻¹).map σ = G) ∧ ∀ A : Matrix (Fin n) (Fin n) R, (A - (G * A.map σ * G⁻¹)ᵀ)ᵀ * G + G * (A - (G * A.map σ * G⁻¹)ᵀ).map σ = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CayleyTransformUnitarySkewInvolutiveStatement

theorem **_root_.GlobalTheory.T.cayleyTransform_unitary_skew_involutive** : ∀ (R : Type) [CommRing R] (σ : R →+* R) (n : ℕ) (G : Matrix (Fin n) (Fin n) R), (∀ x, σ (σ x) = x) → (G.map σ)ᵀ = G → IsUnit G.det → IsUnit (2 : R) → (∀ g : Matrix (Fin n) (Fin n) R, gᵀ * G * g.map σ = G → IsUnit (1 + g).det → ((1 - g) * (1 + g)⁻¹)ᵀ * G + G * ((1 - g) * (1 + g)⁻¹).map σ = 0 ∧ IsUnit (1 + (1 - g) * (1 + g)⁻¹ : Matrix (Fin n) (Fin n) R).det) ∧ (∀ g : Matrix (Fin n) (Fin n) R, IsUnit (1 + g).det → (1 - (1 - g) * (1 + g)⁻¹) * (1 + (1 - g) * (1 + g)⁻¹)⁻¹ = g) ∧ (∀ S : Matrix (Fin n) (Fin n) R, Sᵀ * G + G * S.map σ = 0 → IsUnit (1 + S).det → ((1 - S) * (1 + S)⁻¹)ᵀ * G * ((1 - S) * (1 + S)⁻¹).map σ = G) ∧ ∀ A : Matrix (Fin n) (Fin n) R, (A - (G * A.map σ * G⁻¹)ᵀ)ᵀ * G + G * (A - (G * A.map σ * G⁻¹)ᵀ).map σ = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CharSeriesExistsNeZero

theorem **charSeries_kill_finset** {A I : Type} [AddCommGroup A] (e : I → A → ℂ) (c : I → ℂ) (i₀ : I) (hmul : ∀ i a b, e i (a + b) = e i a * e i b) (hle : ∀ i a, ‖e i a‖ ≤ 1) (h0 : ∀ a, e i₀ a = 1) (hneg : ∀ i, i ≠ i₀ → ∃ a, e i a = -1) (hs : Summable (fun i => ‖c i‖)) (hF : ∀ a, ∑' i, c i * e i a = 0) (s : Finset I) : ∃ c' : I → ℂ, c' i₀ = c i₀ ∧ (∀ i ∈ s, i ≠ i₀ → c' i = 0) ∧ (∀ i, ‖c' i‖ ≤ ‖c i‖) ∧ ∀ a, ∑' i, c' i * e i a = 0

theorem **charSeries_exists_ne_zero** : ∀ (A I : Type) [AddCommGroup A] (e : I → A → ℂ) (c : I → ℂ) (i₀ : I), (∀ i a b, e i (a + b) = e i a * e i b) → (∀ i a, ‖e i a‖ ≤ 1) → (∀ a, e i₀ a = 1) → (∀ i, i ≠ i₀ → ∃ a, e i a = -1) → Summable (fun i => ‖c i‖) → c i₀ ≠ 0 → ∃ a, ∑' i, c i * e i a ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CircleHomEqZpow

theorem **circle_hom_eq_zpow** : ∀ f : Circle →* Circle, Continuous f → ∃ n : ℤ, ∀ z, f z = z ^ n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ClassFinitenessCompletelyReducible

theorem **adjugate_entry_mem_subring** {ι : Type} [Fintype ι] [DecidableEq ι] {R : Type} [CommRing R] (S : Subring R) (k : Matrix ι ι R) (hk : ∀ i j, k i j ∈ S) : ∀ i j, k.adjugate i j ∈ S

theorem **mul_entry_mem_subring** {ι : Type} [Fintype ι] {R : Type} [CommRing R] (S : Subring R) (k l : Matrix ι ι R) (hk : ∀ i j, k i j ∈ S) (hl : ∀ i j, l i j ∈ S) : ∀ i j, (k * l) i j ∈ S

theorem **cf_of_ss** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), IsQGroup Γ → SS Γ → AdelicPair R R₀ → CF Γ R R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ClassFinitenessCompletelyReducibleStatement

theorem **_root_.GlobalTheory.T.classFiniteness_completelyReducible** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), IsQGroup Γ → SS Γ → AdelicPair R R₀ → CF Γ R R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ClassGroupCoprimeRepresentative

theorem **exists_integral_ideal_in_class_coprime** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)), m ≠ ⊥ → ∀ c : ClassGroup (𝒪 K), ∃ b : (Ideal (𝒪 K))⁰, ClassGroup.mk0 b = c ∧ (b : Ideal (𝒪 K)) ⊔ m = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ClassNumberFiniteHolds

theorem **classNumberFinite_holds** : HeckeCharacterZeroLocus.ClassNumberFinite

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ClassNumberFiniteIntegralLevelClosedSubgroupGL

theorem **classFinite_integralLevel** : ∀ (F : Type) [Field F] [NumberField F] (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)), (∀ (B : Type) [CommRing B] [Algebra F B], (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((1 : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) ∧ (∀ g h : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g * h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0)) ∧ ∀ g : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ClassNumberFiniteIntegralLevelClosedSubgroupGLStatement

theorem **_root_.GlobalTheory.T.classNumber_finite_integralLevel_closedSubgroupGL** : ∀ (F : Type) [Field F] [NumberField F] (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)), (∀ (B : Type) [CommRing B] [Algebra F B], (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((1 : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) ∧ (∀ g h : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g * h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0)) ∧ ∀ g : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CmCharacterConjugateInverse

theorem **ideleChar_conj_eq_inv_of_trivial_on_base** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ χ : (AE F E)ˣ →* Circle, (∀ a : (AF F)ˣ, χ (idF F E a) = 1) → ∀ x y : (AE F E)ˣ, (y : AE F E) = cA σ x → χ y = (χ x)⁻¹

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CmCharacterConjugationSymmetry

theorem **_root_.GlobalTheory.cA_incA_apply** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) (x : E) : _root_.HeckeCharacterZeroLocus.ThmBIn.cA σ (_root_.HeckeCharacterZeroLocus.ThmBIn.incA F E x) = _root_.HeckeCharacterZeroLocus.ThmBIn.incA F E (σ x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CmConjugationOnPrimesProperties

theorem **conjIdem_sigR_tmul** (R : Type) [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (a : R) (x : E) : sigR R σ (a ⊗ₜ[F] x) = a ⊗ₜ[F] σ x

theorem **conjIdem_sigR_sigR** (R : Type) [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (h : ∀ x, σ (σ x) = x) (y : R ⊗[F] E) : sigR R σ (sigR R σ y) = y

theorem **conjIdem_sigR_algebraMap** (R : Type) [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (a : R) : sigR R σ (algebraMap R (R ⊗[F] E) a) = algebraMap R (R ⊗[F] E) a

theorem **conjIdem_sigR_inclR** (R : Type) [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (x : E) : sigR R σ (inclR (F

theorem **conjIdem_finPart_cA** (σ : E →ₐ[F] E) (y : AE F E) : finPart F E (cA σ y) = sigR (Af F) σ (finPart F E y)

theorem **conjIdem_sigR_mem_intAE** (σ : E →ₐ[F] E) (h : ∀ x : 𝒪 E, ∃ y : 𝒪 E, (y : E) = σ x) (y : Af F ⊗[F] E) (hy : y ∈ intAE F E) : sigR (Af F) σ y ∈ intAE F E

theorem **cmConjugation_on_primes_adeles_idempotents** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∃ c : HeightOneSpectrum (𝒪 E) ≃ HeightOneSpectrum (𝒪 E), (∀ P, c (c P) = P) ∧ (∀ P, normP (c P) = normP P) ∧ (∀ P (x y : 𝒪 E), (y : E) = σ x → (y ∈ (c P).asIdeal ↔ x ∈ P.asIdeal) ∧ (y ∈ (c P).asIdeal ^ 2 ↔ x ∈ P.asIdeal ^ 2)) ∧ (∀ P Q : HeightOneSpectrum (𝒪 E), P.under (𝒪 F) = Q.under (𝒪 F) ↔ Q = P ∨ Q = c P) ∧ (∀ y, cA σ (cA σ y) = y) ∧ (∀ y : AE F E, finPart F E (cA σ y) ∈ intAE F E ↔ finPart F E y ∈ intAE F E) ∧ ∀ P e f, IsPrimeIdem σ P e f → IsPrimeIdem σ (c P) (cA σ e) f

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CmExtensionAdelesStandardModelDictionary

theorem **cmAdele_dictionary** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∃ Φ : AE F E ≃+* (mixedSpace E × FiniteAdeleRing (𝒪 E) E), (∀ x : E, Φ (incA F E x) = (mixedEmbedding E x, algebraMap E _ x)) ∧ (∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v ((Φ y).2 P) ≤ 1) ∧ (∀ v ε, IsPlaceIdem F v ε → (Φ (algebraMap (AF F) (AE F E) ε)).1 = 0 ∧ ∀ P : HeightOneSpectrum (𝒪 E), (Φ (algebraMap (AF F) (AE F E) ε)).2 P = if P.under (𝒪 F) = v then 1 else 0) ∧ Continuous Φ.symm ∧ ∀ x : (mixedSpace E × FiniteAdeleRing (𝒪 E) E)ˣ, (x : (mixedSpace E × FiniteAdeleRing (𝒪 E) E)).2 = 1 → (∀ w, ((x : (mixedSpace E × FiniteAdeleRing (𝒪 E) E)).1.2 w).im = 0) → ∃ a : (AF F)ˣ, Φ (idF F E a : (AE F E)ˣ) = x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CmExtensionAdelesStandardModelDictionaryStatement

theorem **_root_.GlobalTheory.T.cmExtension_adeles_standardModel_dictionary** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∃ Φ : AE F E ≃+* (mixedSpace E × FiniteAdeleRing (𝒪 E) E), (∀ x : E, Φ (incA F E x) = (mixedEmbedding E x, algebraMap E _ x)) ∧ (∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v ((Φ y).2 P) ≤ 1) ∧ (∀ v ε, IsPlaceIdem F v ε → (Φ (algebraMap (AF F) (AE F E) ε)).1 = 0 ∧ ∀ P : HeightOneSpectrum (𝒪 E), (Φ (algebraMap (AF F) (AE F E) ε)).2 P = if P.under (𝒪 F) = v then 1 else 0) ∧ Continuous Φ.symm ∧ ∀ x : (mixedSpace E × FiniteAdeleRing (𝒪 E) E)ˣ, (x : (mixedSpace E × FiniteAdeleRing (𝒪 E) E)).2 = 1 → (∀ w, ((x : (mixedSpace E × FiniteAdeleRing (𝒪 E) E)).1.2 w).im = 0) → ∃ a : (AF F)ˣ, Φ (idF F E a : (AE F E)ˣ) = x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CmLocalCharacterValueBounded

theorem **idem_glue_mul** {R : Type*} [CommRing R] {e : R} (he : e * e = e) (a b : R) : (e * a + (1 - e)) * (e * b + (1 - e)) = e * (a * b) + (1 - e)

theorem **intValuation_of_mem_notMem_sq** {R : Type*} [CommRing R] [IsDedekindDomain R] (P : HeightOneSpectrum R) {a : R} (h1 : a ∈ P.asIdeal) (h2 : a ∉ P.asIdeal ^ 2) : P.intValuation a = WithZero.exp (-1 : ℤ)

theorem **unifIdele_exists_chiAt** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (χ : (AE F E)ˣ →* Circle) (P : HeightOneSpectrum (𝒪 E)), (∃ x, IsUnifIdele σ P x) ∧ (IsUnramAt σ χ P → ∀ x, IsUnifIdele σ P x → chiAt σ χ P = (χ x : ℂ)) ∧ ‖chiAt σ χ P‖ ≤ 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CompactMeetsFinitelyManyLeftCosetsOpen

theorem **compact_subset_finite_leftCosets** (G : Type) [Group G] [TopologicalSpace G] [IsTopologicalGroup G] (U : Subgroup G) (K : Set G) (hU : IsOpen (U : Set G)) (hK : IsCompact K) : ∃ T : Finset G, ∀ k ∈ K, ∃ t ∈ T, ∃ u ∈ U, k = t * u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CompactMeetsFinitelyManyLeftCosetsOpenStatement

theorem **_root_.GlobalTheory.T.compact_meets_finitely_many_leftCosets_open** : ∀ (G : Type) [Group G] [TopologicalSpace G] [IsTopologicalGroup G] (U : Subgroup G) (K : Set G), IsOpen (U : Set G) → IsCompact K → ∃ T : Finset G, ∀ k ∈ K, ∃ t ∈ T, ∃ u ∈ U, k = t * u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CompletelyReducibleKroneckerPowerTranspose

theorem **exists_semisimple_rep_tpow** {ι : Type} [Fintype ι] [DecidableEq ι] {G : Type} [Group G] (ρ : G →* Matrix ι ι ℂ) (h : SS (Set.range ρ)) (n : ℕ) : ∃ (V : Type) (_ : AddCommGroup V) (_ : Module ℂ V) (r : Representation ℂ G V) (b : Module.Basis (Fin n → ι) ℂ

V), r.IsSemisimpleRepresentation ∧ ∀ g, LinearMap.toMatrix b b (r g) = tpow n (ρ g)

theorem **ss_tpow_transpose** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∀ n : ℕ, SS ((fun g => (tpow n g)ᵀ) '' Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CompletelyReducibleKroneckerPowerTransposeStatement

theorem **_root_.GlobalTheory.T.completelyReducible_kroneckerPower_transpose** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∀ n : ℕ, SS ((fun g => (tpow n g)ᵀ) '' Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CompletelyReducibleToMatrixIffIsSemisimpleRepres

private theorem **subrepresentation_isCompl_iff** {G V : Type} [Group G] [AddCommGroup V] [Module ℂ V] {ρ : Representation ℂ G V} (p q : Subrepresentation ρ) : IsCompl p q ↔ IsCompl p.toSubmodule q.toSubmodule

theorem **_root_.GlobalTheory.ClassFin.ss_range_toMatrix_iff_isSemisimpleRepresentation** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (b : Module.Basis ι ℂ V), _root_.GlobalTheory.ClassFin.SS (Set.range fun g => LinearMap.toMatrix b b (ρ g)) ↔ ρ.IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CompletelyReducibleToMatrixIffIsSemisimpleRepresStatement

theorem **_root_.GlobalTheory.T.completelyReducible_toMatrix_iff_isSemisimpleRepresentation** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (b : Module.Basis ι ℂ V), SS (Set.range fun g => LinearMap.toMatrix b b (ρ g)) ↔ ρ.IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ComplexEmbeddingHermitianQuadrupleSum

theorem **symplOH_add_left** (x x' y : WO S T) : symplOH S T (x + x') y = symplOH S T x y + symplOH S T x' y

theorem **symplOH_add_right** (x y y' : WO S T) : symplOH S T x (y + y') = symplOH S T x y + symplOH S T x y'

theorem **gramInv_star** (V : HSp D) (i j) : star (V.gram⁻¹ i j) = V.gram⁻¹ j i

theorem **herm_sum** {A : Type} [CommRing A] {m n : ℕ} (c : A →+* A) (hc : ∀ a, c (c a) = a) (P : Fin n → Fin n → A) (Q : Fin m → Fin m → A) (hP : ∀ i k, c (P i k) = P k i) (hQ : ∀ i k, c (Q i k) = Q k i) (x y : Matrix (Fin m) (Fin n) A) : c (∑ a, ∑ b, ∑ e, ∑ d, P a b * c (y e b) * Q e d * x d a) = ∑ a, ∑ b, ∑ e, ∑ d, P a b * c (x e b) * Q e d * y d a

theorem **hO_herm** (x y : WO S T) : cA D (hO S T x y) = hO S T y x

theorem **symplO_skew** (x y : WO S T) : symplO S T y x = -symplO S T x y

theorem **map_hO_zero** {A B : Type} [CommRing A] [CommRing B] {m n : ℕ} (q : A →+* B) (c : A →+* A) (hc : ∀ z, q z = 0 → q (c z) = 0) (P : Fin n → Fin n → A) (Q : Fin m → Fin m → A) (x y : Matrix (Fin m) (Fin n) A) (h : x.map q = 0 ∨ y.map q = 0) : q (∑ a, ∑ b, ∑ e, ∑ d, P a b * c (y e b) * Q e d * x d a) = 0

theorem **map_trace_hO_zero** {F E A B : Type} [Field F] [Field E] [Algebra F E] [FiniteDimensional F E] [CommRing A] [CommRing B] [Algebra F A] [Algebra F B] (p : A →ₐ[F] B) (s : E →ₐ[F] E) {m n : ℕ} (c : A ⊗[F] E) (P : Fin n → Fin n → A ⊗[F] E) (Q : Fin m → Fin m → A ⊗[F] E) (x y : Matrix (Fin m) (Fin n) (A ⊗[F] E)) (h : x.map (Algebra.TensorProduct.map p (AlgHom.id F E)).toRingHom = 0 ∨ y.map (Algebra.TensorProduct.map p (AlgHom.id F E)).toRingHom = 0) : p (Algebra.trace A (A ⊗[F] E) (c * ∑ a, ∑ b, ∑ e, ∑ d, P a b * sigR A s (y e b) * Q e d * x d a)) = 0

theorem **map_symplO_eq_zero** {B : Type} [CommRing B] [Algebra D.host.F B] (p : AF D.host.F →ₐ[D.host.F] B) (x y : WO S T) (h : x.map (Algebra.TensorProduct.map p (AlgHom.id D.host.F D.host.E)).toRingHom = 0 ∨ y.map (Algebra.TensorProduct.map p (AlgHom.id D.host.F D.host.E)).toRingHom = 0) : p (symplO S T x y) = 0

theorem **snd_symplO** {x y : WO S T} (h : x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) : sndA D.host.F (symplO S T x y) = 0

theorem **fst_symplO** {x y : WO S T} (h : x.map (infPart D) = 0 ∨ y.map (infPart D) = 0) : fstA D.host.F (symplO S T x y) = 0

theorem **symplOH_fin_real** {u x : WO S T} (hu : u.map (infPart D) = 0) (hx : x.map (finPart D) = 0) : symplOH S T u x = 0 ∧ symplOH S T x u = 0

theorem **continuous_symplOH** {X : Type} [TopologicalSpace X] {x y : X → WO S T} (hx : Continuous x) (hy : Continuous y) : Continuous fun t => symplOH S T (x t) (y t)

theorem **formAt_smul_left** (τ : D.what₂.Tau) (c : ℝ) (x y : Fin T.rank → Fin S.rank → ℂ) : formAt S T τ (c • x) y = c * formAt S T τ x y

theorem **formAt_smul_right** (τ : D.what₂.Tau) (c : ℝ) (x y : Fin T.rank → Fin S.rank → ℂ) : formAt S T τ x (c • y) = c * formAt S T τ x y

theorem **complexEmbedding_hermitianQuadrupleSum** {A : Type} [CommRing A] (φ : A →+* ℂ) (c : A →+* A) (hc : ∀ a, φ (c a) = starRingEnd ℂ (φ a)) {m n : ℕ} (x y : Fin m → Fin n → A) (P : Fin n → Fin n → A) (Q : Fin m → Fin m → A) : φ (∑ a, ∑ b, ∑ e, ∑ d, P a b * c (y e b) * Q e d * x d a) = ∑ a, ∑ b, ∑ e, ∑ d, φ (P a b) * starRingEnd ℂ (φ (y e b)) * φ (Q e d) * φ (x d a)

theorem **valTau_symplO** (hθ : IsInfFam θ) (τ : D.what₂.Tau) (x y : WO S T) : valTau θ τ (symplO S T x y) = formAt S T τ (piW θ T S x τ) (piW θ T S y τ)

theorem **psi_symplOH** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) (w : WO S T) (x : RW T S) : ψ (symplOH S T w (iotaW σ T S x)) = Complex.exp (-(Real.pi * Complex.I) * (omO S T (piW θ T S w) x : ℝ))

theorem **symplOH_self** (x : WO S T) : symplOH S T x x = 0

theorem **omO_self** (h : Chart θ σ) (x : RW T S) : omO S T x x = 0

theorem **theta_zero** : IsThetaFunO ψ S T 0 where

theorem **theta_add** {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : IsThetaFunO ψ S T (f + g) where

theorem **theta_smul** (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (c • f) where

theorem **theta_sub** {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : IsThetaFunO ψ S T (f - g)

theorem **heisO_heisO** (w w' : WO S T) (f : WO S T → ℂ) : heisO ψ S T w (heisO ψ S T w' f) = ψ (symplOH S T w w') • heisO ψ S T (w + w') f

theorem **heisO_quasi** (w₀ : WO S T) (f : WO S T → ℂ) (hf : ∀ γ w, f (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * f w) (γ w) : heisO ψ S T w₀ f (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * heisO ψ S T w₀ f w

theorem **heisO_comm_fin_real** {u x : WO S T} (hu : u.map (infPart D) = 0) (hx : x.map (finPart D) = 0) {f : WO S T → ℂ} (hf : heisO ψ S T u f = f) : heisO ψ S T u (heisO ψ S T x f) = heisO ψ S T x f

theorem **continuous_of_level** (h : Chart θ σ) (hψ : Continuous ψ) (f : WO S T → ℂ) (hU : ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u f = f) (hs : Sm (iotaW σ T S) f) : Continuous f

theorem **theta_sm** (h : Chart θ σ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Sm (iotaW σ T S) f

theorem **heisO_iota** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) (x : RW T S) (f : WO S T → ℂ) (w : WO S T) : heisO ψ S T (iotaW σ T S x) f w = Complex.exp (-(Real.pi * Complex.I) * (omO S T (piW θ T S w) x : ℝ)) * f (w + iotaW σ T S x)

theorem **theta_heisO_real** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {x : WO S T} (hx : x.map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (heisO ψ S T x f)

theorem **dRho_eq_dLine** (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (f : WO S T → ℂ) : Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f = dLine ψ S T (lineAt θ τ z) f

theorem **dLine_eq_hD** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) (f : WO S T → ℂ) : dLine ψ S T L f = hD (iotaW σ T S) (piW θ T S) (omO S T) (piW θ T S (L 1)) f

theorem **theta_dLine** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (dLine ψ S T L f)

theorem **heisO_line_add** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) (f : WO S T → ℂ) (s t : ℝ) : heisO ψ S T (L (s + t)) f = heisO ψ S T (L s) (heisO ψ S T (L t) f)

theorem **hasDerivAt_heisO** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S T) (s : ℝ) : HasDerivAt (fun t : ℝ => heisO ψ S T (L t) f w) (heisO ψ S T (L s) (dLine ψ S T L f) w) s

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousBicharacterWeilSetup

theorem **exists_coercive** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) (hpos : ∀ v, v ≠ 0 → 0 < om v (J v)) : ∃ c : ℝ, 0 < c ∧ ∀ v, c * ‖v‖ ^ 2 ≤ om v (J v)

theorem **exists_exponent** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) : ∃ M : V →L[ℝ] V →L[ℝ] ℂ, ∀ y v, M y v = -(Real.pi : ℂ) * (Complex.I * (om y v : ℝ) + (om y (J v) : ℝ))

theorem **contDiff_bilin** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) : ContDiff ℝ ∞ fun v : V => om v (J v)

theorem **swap_mul** {G : Type} [AddCommGroup G] (β : G → G → ℂ) (hl : ∀ x y z, β (x + y) z = β x z * β y z) (hr : ∀ x y z, β x (y + z) = β x y * β x z) (hd : ∀ x, β x x = 1) (x y : G) : β x y * β y x = 1

theorem **summable_coeff** {G V : Type} [AddCommGroup G] [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (p : G →+ V) (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) (Γ U₀ : AddSubgroup G) (χ : G → ℂ) (hpos : ∀ v, v ≠ 0 → 0 < om v (J v)) (hinj : ∀ g ∈ Γ, p g = 0 → g = 0) (hfin : ∀ R : ℝ, {g : G | g ∈ Γ ∧ g ∈ U₀ ∧ ‖p g‖ ≤ R}.Finite) (hχ0 : ∀ x, x ∉ U₀ → χ x = 0) (hχ1 : ∀ x, ‖χ x‖ ≤ 1) (b : Γ → ℂ) (hb : ∀ g, ‖b g‖ = 1) (w : G) (y : V) (M : V →L[ℝ] V →L[ℝ] ℂ) (R : ℝ) : Summable fun g : Γ => ‖b g * χ (g + w) * Complex.exp (-(Real.pi / 2 : ℂ) * (om (p g + y) (J (p g + y)) : ℝ))‖ * Real.exp (R * ‖M (p g + y)‖)

theorem **series_quasi** (hl : ∀ x y z, β (x + y) z = β x z * β y z) (hr : ∀ x y z, β x (y + z) = β x y * β x z) (hΓ : ∀ g ∈ Γ, ∀ g' ∈ Γ, β g g' = 1) (a : V) (g₀ : G) (hg₀ : g₀ ∈ Γ) (w : G) : β g₀ w * series p ι β om J Γ χ a (g₀ + w) = series p ι β om J Γ χ a w

theorem **series_inv** (hl : ∀ x y z, β (x + y) z = β x z * β y z) (hr : ∀ x y z, β x (y + z) = β x y * β x z) (hd : ∀ x, β x x = 1) (hβ : ∀ x v, β x (ι v) = Complex.exp (-(Real.pi * Complex.I) * (om (p x) v : ℝ))) (a : V) (u : G) (hu : p u = 0) (hχ : ∀ x, β x u * χ (x + u) = χ x) (w : G) : β w u * series p ι β om J Γ χ a (w + u) = series p ι β om J Γ χ a w

theorem **series_expand** (hp : ∀ v, p (ι v) = v) (hl : ∀ x y z, β (x + y) z = β x z * β y z) (hr : ∀ x y z, β x (y + z) = β x y * β x z) (hβ : ∀ x v, β x (ι v) = Complex.exp (-(Real.pi * Complex.I) * (om (p x) v : ℝ))) (hsym : ∀ x y, om x (J y) = om y (J x)) (hχ : ∀ x v, χ (x + ι v) = χ x) (M : V → V → ℂ) (hM : ∀ y v, M y v = -(Real.pi : ℂ) * (Complex.I * (om y v : ℝ) + (om y (J v) : ℝ))) (a : V) (w : G) (v : V) : β w (ι v) * series p ι β om J Γ χ a (w + ι v) = Complex.exp (-(Real.pi / 2 : ℂ) * (om v (J v) : ℝ)) * ∑' g : Γ, (β g w * β (ι a) (g + w) * χ (g + w) * Complex.exp (-(Real.pi / 2 : ℂ) * (om (p g + (a + p w)) (J (p g + (a + p w))) : ℝ))) * Complex.exp (M (p g + (a + p w)) v)

theorem **series_neg** (hp : ∀ v, p (ι v) = v) (hl : ∀ x y z, β (x + y) z = β x z * β y z) (hr : ∀ x y z, β x (y + z) = β x y * β x z) (hd : ∀ x, β x x = 1) (hβ : ∀ x v, β x (ι v) = Complex.exp (-(Real.pi * Complex.I) * (om (p x) v : ℝ))) (halt : ∀ v, om v v = 0) (hχ : ∀ x v, χ (x + ι v) = χ x) (a : V) : series p ι β om J Γ χ a (ι (-a)) = ∑' g : Γ, (χ g * Complex.exp (-(Real.pi / 2 : ℂ) * (om (p g) (J (p g)) : ℝ))) * Complex.exp (2 * Real.pi * Complex.I * (om (p g) a : ℝ))

theorem **exists_theta** : ∀ (G V : Type) [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (p : G →+ V) (ι : V →+ G) (β : G → G → ℂ) (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) (Γ U₀ U : AddSubgroup G) (χ : G → ℂ), Continuous p → Continuous ι → (∀ v, p (ι v) = v) → (∀ x y z, β (x + y) z = β x z * β y z) → (∀ x y z, β x (y + z) = β x y * β x z) → (∀ x, β x x = 1) → (∀ x y, ‖β x y‖ = 1) → Continuous (fun q : G × G => β q.1 q.2) → (∀ x v, β x (ι v) = Complex.exp (-(Real.pi * Complex.I) * (om (p x) v : ℝ))) → (∀ v, om v v = 0) → (∀ v, J (J v) = -v) → (∀ x y, om x (J y) = om y (J x)) → (∀ v, v ≠ 0 → 0 < om v (J v)) → (∀ g ∈ Γ, ∀ g' ∈ Γ, β g g' = 1) → (∀ g ∈ Γ, p g = 0 → g = 0) → (∀ v, ι v ∈ U₀) → (∀ R : ℝ, {g : G | g ∈ Γ ∧ g ∈ U₀ ∧ ‖p g‖ ≤ R}.Finite) → (∀ x, x ∉ U₀ → χ x = 0) → (∀ x, ‖χ x‖ ≤ 1) → χ 0 = 1 → (∀ x v, χ (x +

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousBicharacterWeilSetupStatement

theorem **_root_.GlobalTheory.T.continuous_bicharacter_weil_setup** : ∀ (G V : Type) [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (p : G →+ V) (ι : V →+ G) (β : G → G → ℂ) (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) (Γ U₀ U : AddSubgroup G) (χ : G → ℂ), Continuous p → Continuous ι → (∀ v, p (ι v) = v) → (∀ x y z, β (x + y) z = β x z * β y z) → (∀ x y z, β x (y + z) = β x y * β x z) → (∀ x, β x x = 1) → (∀ x y, ‖β x y‖ = 1) → Continuous (fun q : G × G => β q.1 q.2) → (∀ x v, β x (ι v) = Complex.exp (-(Real.pi * Complex.I) * (om (p x) v : ℝ))) → (∀ v, om v v = 0) → (∀ v, J (J v) = -v) → (∀ x y, om x (J y) = om y (J x)) → (∀ v, v ≠ 0 → 0 < om v (J v)) → (∀ g ∈ Γ, ∀ g' ∈ Γ, β g g' = 1) → (∀ g ∈ Γ, p g = 0 → g = 0) → (∀ v, ι v ∈ U₀) → (∀ R : ℝ, {g : G | g ∈ Γ ∧ g ∈ U₀ ∧ ‖p g‖ ≤ R}.Finite) → (∀ x, x ∉ U₀ → χ x = 0) →

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousCharacterFiniteOrderIntegralUnits

theorem **finite_order_of_small** {R A B : Type} [CommRing R] [CommRing A] [CommRing B] [TopologicalSpace R] [TopologicalSpace A] [TopologicalSpace B] [ContinuousAdd B] (p : R →+* A) (q : R →+* B) (s : A × B → R) (hs : Continuous s) (hsz : ∀ z, s (p z, q z) = z) (S : Subring B) (hsmall : ∀ W ∈ 𝒩 (0 : B), ∃ N : ℕ, 0 < N ∧ ∀ y ∈ S, N • y ∈ W) (hfin : ∀ N : ℕ, 0 < N → Finite (S ⧸ Ideal.span {(N : S)})) (χ : Rˣ →* Circle) (hχ : Continuous χ) : ∃ N : ℕ, 0 < N ∧ (∀ x : Rˣ, (p x = 1 ∧ q x ∈ S ∧ q (x⁻¹ : Rˣ) ∈ S) → (∃ y ∈ S, q x - 1 = N • y) → χ x = 1) ∧ ∃ m : ℕ, 0 < m ∧ ∀ x : Rˣ, (p x = 1 ∧ q x ∈ S ∧ q (x⁻¹ : Rˣ) ∈ S) → χ x ^ m = 1

theorem **mem_hatRing** (z : RA F E) : z ∈ hatRing F E ↔ z ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))

theorem **exists_int_add_nsmul** (N : ℕ) (hN : 0 < N) (z : RA F E) (hz : z ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))) : ∃ (x : 𝒪 E), ∃ w ∈ hat1 F (Set.range (algebraMap (𝒪 E) E)), z = (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (x : E) + N • w

theorem **finite_quotient_hatRing** (N : ℕ) (hN : 0 < N) : Finite (hatRing F E ⧸ Ideal.span {(N : hatRing F E)})

theorem **integralFiniteAdeles_eq_adelicCompletion_integers** (D : D7) : intAE D = hat1 D.host.F (Set.range (algebraMap (𝒪 D.host.E) D.host.E))

theorem **char_finUnit_finite_order** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle), Continuous χ → ∃ N : ℕ, 0 < N ∧ (∀ x, FinUnit x → (∃ y ∈ intAE D, finPart D (x : AE D) - 1 = N • y) → χ x = 1) ∧ ∃ m : ℕ, 0 < m ∧ ∀ x, FinUnit x → χ x ^ m = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousCharacterFiniteOrderIntegralUnitsStatement

theorem **_root_.GlobalTheory.T.continuous_character_finite_order_integral_units** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle), Continuous χ → ∃ N : ℕ, 0 < N ∧ (∀ x, FinUnit x → (∃ y ∈ intAE D, finPart D (x : AE D) - 1 = N • y) → χ x = 1) ∧ ∃ m : ℕ, 0 < m ∧ ∀ x, FinUnit x → χ x ^ m = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousHomDifferentiable

theorem **exists_integral_ne_zero** (χ : ℝ → ℂ) (hc : Continuous χ) (h0 : χ 0 = 1) : ∃ δ : ℝ, (∫ x in (0:ℝ)..δ, χ x) ≠ 0

theorem **continuous_hom_differentiable** : ∀ (χ : ℝ → ℂ), Continuous χ → χ (0 : ℝ) = (1 : ℂ) → (∀ (s t : ℝ), χ (s + t) = χ s * χ t) → Differentiable ℝ χ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousInfiniteAdeleMap

theorem **toAdele_tmul** (a : AF F) (x : E) : toAdele F E (a ⊗ₜ[F] x) = adeleMap F E a * algebraMap E _ x

theorem **toAdele_incA** (x : E) : toAdele F E (incA F E x) = algebraMap E _ x

theorem **toAdele_algebraMap** (a : AF F) : toAdele F E (algebraMap (AF F) (AE F E) a) = adeleMap F E a

theorem **continuous_toAdele** : Continuous (toAdele F E)

theorem **continuous_finiteAdeleMap** : Continuous (finiteAdeleMap F E)

theorem **continuous_infiniteAdeleMap** : Continuous (infiniteAdeleMap F E)

theorem **exists_finite** : ∃ ι : Af F ⊗[F] E →+* FiniteAdeleRing (𝒪 E) E, Function.Bijective ι ∧ (∀ z, (toAdele F E z).2 = ι (finPart F E z)) ∧ ∀ y, y ∈ intAE F E ↔ ∀ Q, ι y Q ∈ Q.adicCompletionIntegers E

theorem **finPart_mem_intAE_iff** (z : AE F E) : finPart F E z ∈ intAE F E ↔ ∀ Q, (toAdele F E z).2 Q ∈ Q.adicCompletionIntegers E

theorem **bijective_toAdele** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) : Function.Bijective (toAdele F E)

theorem **continuous_adeleEquiv_symm** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) : Continuous (adeleEquiv h).symm

theorem **isCMExtension_of** (h : IsCMExt F E σ) : IsCMExtension F E

theorem **isCMExt_iff** (σ : E →ₐ[F] E) : IsCMExt F E σ ↔ ∃ _ : IsCMExtension F E, σ = (conj F E : E →ₐ[F] E)

theorem **sig_sig** (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

theorem **exists_of_fixed** (h : IsCMExt F E σ) {x : E} (hx : σ x = x) : ∃ t, algebraMap F E t = x

theorem **adeleConj_algebraMap** [IsCMExtension F E] (x : E) : adeleConj F E (algebraMap E _ x) = algebraMap E _ (conj F E x)

theorem **adeleConj_adeleMap** [IsCMExtension F E] (a : AF F) : adeleConj F E (adeleMap F E a) = adeleMap F E a

theorem **toAdele_cA** [IsCMExtension F E] (z : AE F E) : toAdele F E (cA (conj F E : E →ₐ[F] E) z) = adeleConj F E (toAdele F E z)

theorem **cO_cO** (h : IsCMExt F E σ) (x : 𝒪 E) : cO σ (cO σ x) = x

theorem **eq_of_le** {P Q : HeightOneSpectrum (𝒪 E)} (h : P.asIdeal ≤ Q.asIdeal) : P = Q

theorem **under_cP** (h : IsCMExt F E σ) (P : HeightOneSpectrum (𝒪 E)) : (cP h P).under (𝒪 F) = P.under (𝒪 F)

theorem **eq_or_eq_cP** (h : IsCMExt F E σ) {P Q : HeightOneSpectrum (𝒪 E)} (hQ : Q.under (𝒪 F) = P.under (𝒪 F)) : Q = P ∨ Q = cP h P

theorem **cA_cA** (h : IsCMExt F E σ) (z : AE F E) : cA σ (cA σ z) = z

theorem **finPart_cA_mem** {z : AE F E} (hz : finPart F E z ∈ intAE F E) : finPart F E (cA σ z) ∈ intAE F E

theorem **fst_cA** (h : IsCMExt F E σ) {z : AE F E} (hz : (toAdele F E z).1 = 0) : (toAdele F E (cA σ z)).1 = 0

theorem **IsIndicator.unique** {S : Set (HeightOneSpectrum (𝒪 E))} {a b : AdeleRing (𝒪 E) E} (ha : IsIndicator S a) (hb : IsIndicator S b) : a = b

theorem **fst_mul** (a b : AdeleRing (𝒪 E) E) : (a * b).1 = a.1 * b.1

theorem **fst_add** (a b : AdeleRing (𝒪 E) E) : (a + b).1 = a.1 + b.1

theorem **fst_one_sub** (a : AdeleRing (𝒪 E) E) : (1 - a).1 = 1 - a.1

theorem **ev_mul** (a b : AdeleRing (𝒪 E) E) (Q) : (a * b).2 Q = a.2 Q * b.2 Q

theorem **ev_add** (a b : AdeleRing (𝒪 E) E) (Q) : (a + b).2 Q = a.2 Q + b.2 Q

theorem **ev_one_sub** (a : AdeleRing (𝒪 E) E) (Q) : (1 - a).2 Q = 1 - a.2 Q

theorem **IsIndicator.idem** {S : Set (HeightOneSpectrum (𝒪 E))} {a : AdeleRing (𝒪 E) E} (ha : IsIndicator S a) : a * a = a

theorem **exists_isIndicator** (P : HeightOneSpectrum (𝒪 E)) : ∃ a, IsIndicator {P} a

theorem **isPlaceIdem_iff** (v : HeightOneSpectrum (𝒪 F)) (ε : AF F) : IsPlaceIdem F v ε ↔ IsIndicator {v} ε

theorem **place_comp** {v : HeightOneSpectrum (𝒪 F)} {ε : AF F} (h : IsPlaceIdem F v ε) : IsIndicator {Q | Q.under (𝒪 F) = v} (toAdele F E (algebraMap _ _ ε))

theorem **Comparison₂.conj** (h : IsCMExt F E σ) {P : HeightOneSpectrum (𝒪 E)} {e : AE F E} (ht : Comparison₂ F P e) : Comparison₂ F (cP h P) (cA σ e)

theorem **not_int_iff** {e : AE F E} (he : e * e = e) {x : 𝒪 E} (hx : x ≠ 0) : finPart F E (e * incA F E ((x : E)⁻¹)) ∉ intAE F E ↔ ∃ Q, (toAdele F E e).2 Q ≠ 0 ∧ x ∈ Q.asIdeal

theorem **tie_iff** {e : AE F E} (he : e * e = e) (h0 : (toAdele F E e).1 = 0) (P : HeightOneSpectrum (𝒪 E)) : Comparison₂ F P e ↔ IsIndicator {P} (toAdele F E e)

theorem **idem_of** {S : Set (HeightOneSpectrum (𝒪 E))} {e : AE F E} (he : IsIndicator S (toAdele F E e)) : e * e = e

theorem **delta_conj** {P : HeightOneSpectrum (𝒪 E)} {e : AE F E} (hd : IsIndicator {P} (toAdele F E e)) : IsIndicator {cP h P} (toAdele F E (cA σ e))

theorem **of_isPrimeIdem** {P : HeightOneSpectrum (𝒪 E)} {e f : AE F E} (hp : IsPrimeIdem σ P e f) : IsIndicator {P} (toAdele F E e) ∧ IsIndicator {Q | Q.under (𝒪 F) = P.under (𝒪 F)} (toAdele F E f)

theorem **exists_isPrimeIdem** (P : HeightOneSpectrum (𝒪 E)) : ∃ e f, IsPrimeIdem σ P e f

theorem **coe_toIdele** (x : (AE F E)ˣ) : (toIdele F E x : AdeleRing (𝒪 E) E) = toAdele F E x

theorem **toIdele_principal** (x : Eˣ) : toIdele F E (Units.map (incA F E).toMonoidHom x) = principalIdele E x

theorem **ideleEquiv_apply** (h : IsCMExt F E σ) (x : (AE F E)ˣ) : ideleEquiv h x = toIdele F E x

theorem **isHeckeChar_pull** (χ : UnitaryHeckeChar E) : IsHeckeChar F E (pull F χ)

theorem **isHeckeChar_iff** (h : IsCMExt F E σ) (χ : (AE F E)ˣ →* Circle) : IsHeckeChar F E χ ↔ ∃! χ' : UnitaryHeckeChar E, pull F χ' = χ

theorem **pull_eq_one_iff** (h : IsCMExt F E σ) (χ : UnitaryHeckeChar E) : pull F χ = 1 ↔ χ = UnitaryHeckeChar.trivial E

theorem **localIdele_same** (P : HeightOneSpectrum (𝒪 E)) (u : (P.adicCompletion E)ˣ) : (localIdele P u).val.2 P = u

theorem **localIdele_ne** {P Q : HeightOneSpectrum (𝒪 E)} (hQ : Q ≠ P) (u : (P.adicCompletion E)ˣ) : (localIdele P u).val.2 Q = 1

theorem **mix_eq_localIdele** {P : HeightOneSpectrum (𝒪 E)} {a : AdeleRing (𝒪 E) E} (ha : IsIndicator {P} a) (y : AdeleRing (𝒪 E) E) (u : (P.adicCompletion E)ˣ) (hu : y.2 P = u) : a * y + (1 - a) = (localIdele P u).val

theorem **toAdele_mix** (e y : AE F E) : toAdele F E (e * y + (1 - e)) = toAdele F E e * toAdele F E y + (1 - toAdele F E e)

theorem **valued_eq_iff** (P : HeightOneSpectrum (𝒪 E)) (r : 𝒪 E) : Valued.v (algebraMap E (P.adicCompletion E) r) = WithZero.exp (-1 : ℤ) ↔ r ∈ P.asIdeal ∧ r ∉ P.asIdeal ^ 2

theorem **isUnifIdele_iff** (P : HeightOneSpectrum (𝒪 E)) (x : (AE F E)ˣ) : IsUnifIdele σ P x ↔ ∃ ϖ : (P.adicCompletion E)ˣ, IsUniformiser P ϖ ∧ (∃ r : 𝒪 E, algebraMap E (P.adicCompletion E) r = ϖ) ∧ toIdele F E x = localIdele P ϖ

theorem **exists_isUnifIdele** (P : HeightOneSpectrum (𝒪 E)) : ∃ x, IsUnifIdele σ P x

theorem **isUnramAt_iff** (χ : UnitaryHeckeChar E) (P : HeightOneSpectrum (𝒪 E)) : IsUnramAt σ (pull F χ) P ↔ χ.IsUnramifiedAt P

theorem **chiAt_eq** (χ : UnitaryHeckeChar E) (P : HeightOneSpectrum (𝒪 E)) : chiAt σ (pull F χ) P = χ.valueAt P

theorem **lfac_eq** (χ : UnitaryHeckeChar E) (P : HeightOneSpectrum (𝒪 E)) (s : ℂ) : lfac σ (pull F χ) P s = χ.eulerFactor P s

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ContinuousSymmFinite

theorem **continuous_symm_finite** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ι : FiniteAdeleRing (𝒪 F) F ⊗[F] E ≃+* FiniteAdeleRing (𝒪 E) E), (∀ y, y ∈ AddSubgroup.closure {y : FiniteAdeleRing (𝒪 F) F ⊗[F] E | ∃ a ∈ {a : FiniteAdeleRing (𝒪 F) F | ∀ v : HeightOneSpectrum (𝒪 F), a v ∈ v.adicCompletionIntegers F}, ∃ x : 𝒪 E, y = a ⊗ₜ[F] (x : E)} ↔ ∀ P, Valued.v (ι y P) ≤ 1) → letI : TopologicalSpace (FiniteAdeleRing (𝒪 F) F ⊗[F] E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CoordDivisible

theorem **algebraMap_mem** (a : 𝒪 F) : algebraMap F (FiniteAdeleRing (𝒪 F) F) (a : F) ∈ intAdeles F

theorem **exists_of_mem** (k : F) (h : algebraMap F (FiniteAdeleRing (𝒪 F) F) k ∈ intAdeles F) : ∃ a : 𝒪 F, (a : F) = k

theorem **dvd_of_eq** (x : 𝒪 F) (N : ℕ) (hN : 0 < N) (e : FiniteAdeleRing (𝒪 F) F) (he : e ∈ intAdeles F) (h : algebraMap F (FiniteAdeleRing (𝒪 F) F) (x : F) = (N : FiniteAdeleRing (𝒪 F) F) * e) : ∃ a : 𝒪 F, x = (N : 𝒪 F) * a

theorem **int_dvd** (z : ℤ) (N : ℕ) (a : 𝒪 F) (h : (z : 𝒪 F) = (N : 𝒪 F) * a) : (N : ℤ) ∣ z

theorem **coord_divisible** : ∀ (F : Type) [Field F] [NumberField F] (β : Type) [Fintype β] (ω : Module.Basis β ℤ (𝒪 F)) (r : β → FiniteAdeleRing (𝒪 F) F) (N : ℕ), 0 < N → (∀ i, (∀ M : ℕ, 0 < M → ∃ (z : ℤ) (c : FiniteAdeleRing (𝒪 F) F), (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ r i = (z : FiniteAdeleRing (𝒪 F) F) + (M : FiniteAdeleRing (𝒪 F) F) * c)) → (∃ e : FiniteAdeleRing (𝒪 F) F, (∀ v, (e : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ ∑ i, r i * algebraMap F (FiniteAdeleRing (𝒪 F) F) (ω i : 𝒪 F) = (N : FiniteAdeleRing (𝒪 F) F) * e) → ∀ i, ∃ s : FiniteAdeleRing (𝒪 F) F, (∀ M : ℕ, 0 < M → ∃ (z : ℤ) (c : FiniteAdeleRing (𝒪 F) F), (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ s = (z : FiniteAdeleRing (𝒪 F) F) + (M : FiniteAdeleRing (𝒪 F) F) * c) ∧ r i = (N : FiniteAdeleRing

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CoordsLeftMulMatrixConstBaseChange

theorem **repr_algebraMap_of_basis_image** (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] (ω : Module.Basis β ℚ F) (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A) (hb : ∀ i, b i = algebraMap F A (ω i)) (x : F) (k : β) : b.repr (algebraMap F A x) k = algebraMap ℚ R (ω.repr x k)

theorem **coords_const_baseChange** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ x : F, (∀ k, b.repr (algebraMap F A x) k = algebraMap ℚ R (ω.repr x k)) ∧ Algebra.leftMulMatrix b (algebraMap F A x) = (Algebra.leftMulMatrix ω x).map (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CoordsLeftMulMatrixConstBaseChangeStatement

theorem **_root_.GlobalTheory.T.coords_leftMulMatrix_const_baseChange** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ x : F, (∀ k, b.repr (algebraMap F A x) k = algebraMap ℚ R (ω.repr x k)) ∧ Algebra.leftMulMatrix b (algebraMap F A x) = (Algebra.leftMulMatrix ω x).map (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CountableIdealSummableAbsNormRpow

theorem **_root_.GlobalTheory.countable_ideal_and_summable_absNorm_rpow** : ∀ (K : Type) [Field K] [NumberField K], Countable (Ideal (𝒪 K)) ∧ ∀ σ : ℝ, 1 < σ → (Summable fun I : Ideal (𝒪 K) => (Ideal.absNorm I : ℝ) ^ (-σ)) ∧ Summable fun P : HeightOneSpectrum (𝒪 K) => (Ideal.absNorm P.asIdeal : ℝ) ^ (-σ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CyclicKummerNormFixedHilbert90

theorem **_root_.GlobalTheory.CNorm.kummer_basic** : ∀ (p : ℕ) (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), _root_.GlobalTheory.CNorm.IsKum p ζ δ θ σ → FiniteDimensional F E ∧ Module.finrank F E = p ∧ (∀ y : E, σ^[p] y = y) ∧ (∀ y : E, σ y = y → ∃ a : F, y = algebraMap F E a) ∧ (∀ y : E, ∏ i ∈ Finset.range p, σ^[i] y = 1 → ∃ z : E, z ≠ 0 ∧ y = σ z / z) ∧ (∀ (u : F) (y : E), y ^ p = algebraMap F E u → ∃ (k : ℕ) (r : F), u = δ ^ k * r ^ p) ∧ ∀ x : Fˣ, x ∈ _root_.GlobalTheory.CNorm.kNorm p (RingHom.id F) δ ↔ ∃ y : E, ∏ i ∈ Finset.range p, σ^[i] y = algebraMap F E (x : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.CyclicKummerNormFixedHilbert90Statement

theorem **_root_.GlobalTheory.T.cyclic_hilbert90_kummer_and_mem_kNorm_iff** : ∀ (p : ℕ) (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → FiniteDimensional F E ∧ Module.finrank F E = p ∧ (∀ y : E, σ^[p] y = y) ∧ (∀ y : E, σ y = y → ∃ a : F, y = algebraMap F E a) ∧ (∀ y : E, ∏ i ∈ Finset.range p, σ^[i] y = 1 → ∃ z : E, z ≠ 0 ∧ y = σ z / z) ∧ (∀ (u : F) (y : E), y ^ p = algebraMap F E u → ∃ (k : ℕ) (r : F), u = δ ^ k * r ^ p) ∧ ∀ x : Fˣ, x ∈ kNorm p (RingHom.id F) δ ↔ ∃ y : E, ∏ i ∈ Finset.range p, σ^[i] y = algebraMap F E (x : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DedekindZetaEulerProductMulSubOneEntire

theorem **_root_.GlobalTheory.dedekindZeta_eulerProduct_mul_sub_one_entire** (K : Type) [Field K] [NumberField K] : ∃ Z : ℂ → ℂ, Differentiable ℂ Z ∧ ∀ s : ℂ, 1 < s.re → Z s = (s - 1) * Complex.exp (∑' P : HeightOneSpectrum (𝓞 K), -Complex.log (1 - 1 * ((Ideal.absNorm P.asIdeal : ℕ) : ℂ) ^ (-s)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DescendBlockDiagPartRatPolyOfFibreConst

theorem **exists_component_subst** {σ : Type} (s : σ → MvPolynomial σ ℚ) (l : ℂ →ₗ[ℚ] ℚ) (b : MvPolynomial σ ℂ) : ∃ b' : MvPolynomial σ ℚ, ∀ m, MvPolynomial.coeff m (MvPolynomial.aeval s b') = l (MvPolynomial.coeff m (MvPolynomial.aeval (fun i => MvPolynomial.map (algebraMap ℚ ℂ) (s i)) b))

theorem **exists_poly_blk** {ι : Type} [Fintype ι] [DecidableEq ι] (e : Matrix ι ι ℚ) : ∃ s : ι × ι → MvPolynomial (ι × ι) ℚ, ∀ (x : Matrix ι ι ℂ) (ij : ι × ι), ev (s ij) x = blk e x ij.1 ij.2

theorem **descend_blk_rat** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∀ f : MvPolynomial (ι × ι) ℚ, (∀ x ∈ Γ, ∀ x' ∈ Γ, blk e x = blk e x' → ev f x = ev f x') → ∃ b : MvPolynomial (ι × ι) ℚ, ∀ x ∈ Γ, ev f x = ev b (blk e x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DescendBlockDiagPartRatPolyOfFibreConstStatement

theorem **_root_.GlobalTheory.T.descend_blockDiagPart_ratPoly_of_fibre_const** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∀ f : MvPolynomial (ι × ι) ℚ, (∀ x ∈ Γ, ∀ x' ∈ Γ, blk e x = blk e x' → ev f x = ev f x') → ∃ b : MvPolynomial (ι × ι) ℚ, ∀ x ∈ Γ, ev f x = ev b (blk e x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DetLeadingMinorGramIwasawa

theorem **det_leading_gram_iwasawa** : ∀ (N : ℕ) (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n : Matrix (Fin N) (Fin N) ℝ), kᵀ * k = 1 → (∀ i, n i i = 1) → (∀ i j, j < i → n i j = 0) → ∀ (j : ℕ) (h : j ≤ N), (((k * diagonal a * n)ᵀ * (k * diagonal a * n)).submatrix (Fin.castLE h) (Fin.castLE h)).det = ∏ i : Fin j, a (Fin.castLE h i) ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DetLeadingMinorGramIwasawaStatement

theorem **_root_.GlobalTheory.T.det_leading_minor_gram_iwasawa** : ∀ (N : ℕ) (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n : Matrix (Fin N) (Fin N) ℝ), kᵀ * k = 1 → (∀ i, n i i = 1) → (∀ i j, j < i → n i j = 0) → ∀ (j : ℕ) (h : j ≤ N), (((k * diagonal a * n)ᵀ * (k * diagonal a * n)).submatrix (Fin.castLE h) (Fin.castLE h)).det = ∏ i : Fin j, a (Fin.castLE h i) ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DifferentiableExpMulOfAnnihilated

theorem **_root_.GlobalTheory.differentiable_exp_mul_of_annihilated** : ∀ (g : ℕ) (Hm : Matrix (Fin g) (Fin g) ℂ), Hm.IsHermitian → ∀ f : (Fin g → ℂ) → ℂ, Differentiable ℝ f → (∀ w z : Fin g → ℂ, (fderiv ℝ f w z + Real.pi * Complex.I * ((w ·ᵥ Hm.mulVec (star z)).im : ℂ) * f w) + Complex.I * (fderiv ℝ f w (Complex.I • z) + Real.pi * Complex.I * ((w ·ᵥ Hm.mulVec (star (Complex.I • z))).im : ℂ) * f w) = 0) → Differentiable ℂ fun w : Fin g → ℂ => Complex.exp (Real.pi / 2 * (w ·ᵥ Hm.mulVec (star w))) * f w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DifferentiableExpMulOfAnnihilatedStatement

theorem **_root_.GlobalTheory.T.differentiable_exp_mul_of_annihilated'** : ∀ (g : ℕ) (Hm : Matrix (Fin g) (Fin g) ℂ), Hm.IsHermitian → ∀ f : (Fin g → ℂ) → ℂ, Differentiable ℝ f → (∀ w z : Fin g → ℂ, (fderiv ℝ f w z + Real.pi * Complex.I * ((w ·ᵥ Hm.mulVec (star z)).im : ℂ) * f w) + Complex.I * (fderiv ℝ f w (Complex.I • z) + Real.pi * Complex.I * ((w ·ᵥ Hm.mulVec (star (Complex.I • z))).im : ℂ) * f w) = 0) → Differentiable ℂ fun w : Fin g → ℂ => Complex.exp (Real.pi / 2 * (w ·ᵥ Hm.mulVec (star w))) * f w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.DifferentiableOfChar

theorem **differentiable_of_char** (χ : ℝ → ℂ) (hc : Continuous χ) (h0 : χ 0 = 1) (hm : ∀ s t, χ (s + t) = χ s * χ t) : Differentiable ℝ χ

theorem **differentiable_heisO** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {S T : HSp D} {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (L : ℝ →+ WO S T) (hL : Continuous L) (hfin : ∀ t, (L t).map (finPart D) = 0) (w : WO S T) : Differentiable ℝ fun t : ℝ => heisO ψ S T (L t) f w

theorem **dRho_sum** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {θ : InfFam D} (hθ : IsInfFam θ) {S T : HSp D} {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {ι : Type} (s : Finset ι) (a : ι → WO S T) (c : ι → ℂ) (ha : ∀ i ∈ s, (a i).map (infPart D) = 0) : dRho (heisO ψ S T) θ τ z (∑ i ∈ s, c i • heisO ψ S T (a i) f) = ∑ i ∈ s, c i • heisO ψ S T (a i) (dRho (heisO ψ S T) θ τ z f)

theorem **isFockVacAt_sum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D), IsCharAF D.host.F ψ → IsInfFam θ → ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → ∀ (τ : D.what₂.Tau) (J : Matrix (Fin T.rank) (Fin S.rank) ℂ → Matrix (Fin T.rank) (Fin S.rank) ℂ), IsFockVacAt (heisO ψ S T) θ τ J f → ∀ (ι : Type) (s : Finset ι) (a : ι → WO S T) (c : ι → ℂ), (∀ i ∈ s, (a i).map (infPart D) = 0) → IsFockVacAt (heisO ψ S T) θ τ J (∑ i ∈ s, c i • heisO ψ S T (a i) f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EntriesBoundedOfConjSpanningBounded

theorem **entries_bounded_of_conj_spanning_bounded** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (g : σ → Matrix ι ι ℂ), Submodule.span ℂ (Set.range g) = ⊤ → ∀ C : ℝ, ∃ C' : ℝ, ∀ P : Matrix ι ι ℂ, P.det = 1 → (∀ s i j, ‖(P * g s * P⁻¹) i j‖ ≤ C) → ∀ i j, ‖P i j‖ ≤ C'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EntriesBoundedOfConjSpanningBoundedStatement

theorem **_root_.GlobalTheory.T.entries_bounded_of_conj_spanning_bounded'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (g : σ → Matrix ι ι ℂ), Submodule.span ℂ (Set.range g) = ⊤ → ∀ C : ℝ, ∃ C' : ℝ, ∀ P : Matrix ι ι ℂ, P.det = 1 → (∀ s i j, ‖(P * g s * P⁻¹) i j‖ ≤ C) → ∀ i j, ‖P i j‖ ≤ C'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EqZeroOfAeFundamentalDomain

theorem **ae_zero_of_lintegral_eq** {X : Type} [MeasurableSpace X] (ν : Measure X) [IsFiniteMeasure ν] (a b : ℕ → X → ℂ) (c : X → ℂ) (C : ℝ) (ha : ∀ n, Measurable (a n)) (hb : ∀ n, Measurable (b n)) (hc : Measurable c) (hC : ∀ n x, ‖a n x‖ ≤ C) (ha0 : ∀ x, Tendsto (fun n => a n x) atTop (𝓝 0)) (hbc : ∀ x, Tendsto (fun n => b n x) atTop (𝓝 (c x))) (he : ∀ n, ∫⁻ x, ‖b n x‖ₑ ^ 2 ∂ν = ∫⁻ x, ‖a n x‖ₑ ^ 2 ∂ν) : ∀ᵐ x ∂ν, c x = 0

theorem **eq_zero_of_ae_fundamentalDomain** {G : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [SecondCountableTopology G] [MeasurableSpace G] [BorelSpace G] (Γ : AddSubgroup G) [DiscreteTopology Γ] (μ : Measure G) [μ.IsAddHaarMeasure] (𝓕 : Set G) (h𝓕 : IsAddFundamentalDomain Γ 𝓕 μ) (c : G → ℂ) (hc : Continuous c) (hp : ∀ γ ∈ Γ, ∀ w, ‖c (γ + w)‖ = ‖c w‖) (h0 : ∀ᵐ x ∂μ.restrict 𝓕, c x = 0) : c = 0

theorem **tendsto_diffQuot** {F : ℝ → ℂ} {d : ℂ} (h : HasDerivAt F d 0) : Tendsto (fun n : ℕ => ((1 / ((n : ℝ) + 1) : ℝ) : ℂ)⁻¹ * (F (1 / ((n : ℝ) + 1)) - F 0)) atTop (𝓝 d)

theorem **norm_diffQuot_le** {F F' : ℝ → ℂ} {C : ℝ} (h : ∀ s, HasDerivAt F (F' s) s) (hC : ∀ s, ‖F' s‖ ≤ C) (t : ℝ) (ht : t ≠ 0) : ‖(t : ℂ)⁻¹ * (F t - F 0)‖ ≤ C

theorem **norm_periodic** (hψ : IsStdChar D ψ (-1)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : ∀ γ ∈ ratL T S, ∀ w, ‖f (γ + w)‖ = ‖f w‖

theorem **exists_bound** (hψ : IsStdChar D ψ (-1)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : ∃ C : ℝ, ∀ w, ‖f w‖ ≤ C

theorem **om_dLine** (hW : IsWeilDatumO ψ S T χS χT om) (g : GA D S) (h : GA D T) {L L' : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) (hL' : Continuous L') (hL0' : ∀ t, (L' t).map (finPart D) = 0) (e : ∀ t, L' t = actO S T g h (L t)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (dLine ψ S T L f) = dLine ψ S T L' (om (g, h) f)

theorem **compAt_mul** (θ : InfFam D) (τ : D.what₂.Tau) {k m n : ℕ} (x : Mx D k m) (y : Mx D m n) : compAt θ τ (HMul.hMul x y : Mx D k n) = compAt θ τ x * compAt θ τ y

theorem **compAt_one** (θ : InfFam D) (τ : D.what₂.Tau) {k : ℕ} : compAt θ τ (1 : Mx D k k) = 1

theorem **mt_mul** (V : HSp D) (g g' : GA D V) : mt V (g * g') = mt V g * mt V g'

theorem **mt_one** (V : HSp D) : mt V 1 = 1

theorem **compAt_mt_mul_inv** (θ : InfFam D) (τ : D.what₂.Tau) (V : HSp D) (g : GA D V) : compAt θ τ (mt V g) * compAt θ τ (mt V g⁻¹) = 1

theorem **compAt_mt_inv_mul** (θ : InfFam D) (τ : D.what₂.Tau) (V : HSp D) (g : GA D V) : compAt θ τ (mt V g⁻¹) * compAt θ τ (mt V g) = 1

theorem **compAt_actO** (θ : InfFam D) (τ : D.what₂.Tau) (g : GA D S) (h : GA D T) (x : WO S T) : compAt θ τ (actO S T g h x) = compAt θ τ (mt T h) * compAt θ τ x * compAt θ τ (mt S g⁻¹)

theorem **actO_add** (g : GA D S) (h : GA D T) (x y : WO S T) : actO S T g h (x + y) = actO S T g h x + actO S T g h y

theorem **lineAt_actO** (hθ : IsInfFam θ) (τ : D.what₂.Tau) (g : GA D S) (h : GA D T) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (t : ℝ) : lineAt θ τ (compAt θ τ (mt T h) * z * compAt θ τ (mt S g⁻¹)) t = actO S T g h (lineAt θ τ z t)

theorem **om_dRho** (hW : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f) = Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ (compAt θ τ (mt T h) * z * compAt θ τ (mt S g⁻¹)) (om (g, h) f)

theorem **cplxO_move** (θ : InfFam D) (τ : D.what₂.Tau) (g : GA D S) (h : GA D T) {sS sS' : Matrix (Fin S.rank) (Fin S.rank) ℂ} {sT sT' : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hS : sS' * compAt θ τ (mt S g) = compAt θ τ (mt S g) * sS) (hT : sT' * compAt θ τ (mt T h) = compAt θ τ (mt T h) * sT) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) : cplxO sS' sT' (compAt θ τ (mt T h) * z * compAt θ τ (mt S g⁻¹)) = compAt θ τ (mt T h) * cplxO sS sT z * compAt θ τ (mt S g⁻¹)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EqZeroOfContractions

theorem **coeff_eq_contraction** {V W : Type} [AddCommGroup V] [Module ℚ V] [AddCommGroup W] [Module ℚ W] {κ : Type} [DecidableEq κ] (b : Module.Basis κ ℚ W) (i : κ) (t : TensorProduct ℚ V W) : TensorProduct.equivFinsuppOfBasisRight b t i = TensorProduct.rid ℚ V (LinearMap.lTensor V (b.coord i) t)

theorem **eq_zero_of_contractions** {V W : Type} [AddCommGroup V] [Module ℚ V] [AddCommGroup W] [Module ℚ W] (t : TensorProduct ℚ V W) (h : ∀ m : W →ₗ[ℚ] ℚ, TensorProduct.rid ℚ V (LinearMap.lTensor V m t) = 0) : t = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EqZeroOfLintegralEqZero

theorem **taylor_bound** {φ φ' φ'' : ℝ → ℂ} (h1 : ∀ s, HasDerivAt φ (φ' s) s) (h2 : ∀ s, HasDerivAt φ' (φ'' s) s) {C : ℝ} (hC : ∀ s, ‖φ'' s‖ ≤ C) {t : ℝ} (ht : 0 < t) : ‖(t : ℂ)⁻¹ * (φ t - φ 0) - φ' 0‖ ≤ C * t

theorem **eq_zero_of_lintegral_eq_zero** {G : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [MeasurableSpace G] [BorelSpace G] (Γ : AddSubgroup G) [Countable Γ] (μ : Measure G) [μ.IsAddHaarMeasure] (𝓕 : Set G) (h𝓕 : IsAddFundamentalDomain Γ 𝓕 μ) (f : G → ℂ) (hf : Continuous f) (hp : ∀ γ ∈ Γ, ∀ w, ‖f (γ + w)‖ = ‖f w‖) (h0 : ∫⁻ w in 𝓕, ‖f w‖ₑ ^ 2 ∂μ = 0) : f = 0

theorem **lintegral_eq_zero_of_approx** {G : Type} [MeasurableSpace G] (ν : Measure G) (hν : ν Set.univ = 1) (δ : G → ℂ) (C C' : ℝ) (h : ∀ t : ℝ, 0 < t → ∃ a b r : G → ℂ, δ = a - b ∧ ∫⁻ w, ‖a w‖ₑ ^ 2 ∂ν = ∫⁻ w, ‖r w‖ₑ ^ 2 ∂ν ∧ (∀ w, ‖r w‖ ≤ C * t) ∧ ∀ w, ‖b w‖ ≤ C' * t) : ∫⁻ w, ‖δ w‖ₑ ^ 2 ∂ν = 0

theorem **theta_norm_periodic** (hψ : IsStdChar D ψ (-1)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : ∀ γ ∈ ratL T S, ∀ w, ‖f (γ + w)‖ = ‖f w‖

theorem **theta_bounded** (hψ : IsStdChar D ψ (-1)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : ∃ C : ℝ, ∀ w, ‖f w‖ ≤ C

theorem **heisO_zero** (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **quotient_bound** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : ∃ C : ℝ, ∀ t : ℝ, 0 < t → ∀ w, ‖((t : ℂ)⁻¹ • (heisO ψ S T (L t) f - f) - dLine ψ S T L f) w‖ ≤ C * t

theorem **om_dLine** (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) {L L' : ℝ →+ WO S T} (hL : IsRealDir T S L) (hL' : IsRealDir T S L') (hLL : ∀ t, L' t = actO S T g h (L t)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (dLine ψ S T L f) = dLine ψ S T L' (om (g, h) f)

theorem **isLineAt_lineAt** (hθ : IsInfFam θ) {m n : ℕ} (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) : IsLineAt θ τ z (lineAt θ τ z)

theorem **isRealDir_lineAt** (hθ : IsInfFam θ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) : IsRealDir T S (lineAt θ τ z)

theorem **compAt_mul** (τ : D.what₂.Tau) {k m n : ℕ} (x : Mx D k m) (y : Mx D m n) : compAt θ τ (HMul.hMul x y : Mx D k n) = compAt θ τ x * compAt θ τ y

theorem **compAt_mt_mul_inv** (τ : D.what₂.Tau) (g : GA D S) : compAt θ τ (mt S g) * compAt θ τ (mt S g⁻¹) = 1

theorem **lineAt_move** (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (t : ℝ) : lineAt θ τ (compAt θ τ (mt T h) * z * compAt θ τ (mt S g⁻¹)) t = actO S T g h (lineAt θ τ z t)

theorem **theta_dRho** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f)

theorem **om_dRho** (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f) = Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ (compAt θ τ (mt T h) * z * compAt θ τ (mt S g⁻¹)) (om (g, h) f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EulerProductNeZeroOneOfEntire

private lemma **neg_one_le_re_pow_of_norm_le_one** {c : ℂ} (hc : ‖c‖ ≤ 1) (k : ℕ) : -1 ≤ (c ^ k).re

theorem **_root_.GlobalTheory.not_summable_two_add_two_re_pow_div** (c : ℂ) (hc : ‖c‖ ≤ 1) : ¬ Summable fun k : ℕ => (2 + 2 * (c ^ (k + 1)).re) / ((k : ℝ) + 1)

private lemma **norm_mul_natCast_cpow_neg_le** {n : ℕ} (hn : 2 ≤ n) {c : ℂ} (hc : ‖c‖ ≤ 1) (s : ℂ) : ‖c * (n : ℂ) ^ (-s)‖ ≤ (n : ℝ) ^ (-s.re)

private lemma **natCast_rpow_neg_le_half** {n : ℕ} (hn : 2 ≤ n) {σ : ℝ} (hσ : 1 < σ) : (n : ℝ) ^ (-σ) ≤ 1 / 2

theorem **_root_.GlobalTheory.conj_neg_log_euler_factor** (n : ℕ) (hn : 2 ≤ n) (c : ℂ) (hc : ‖c‖ ≤ 1) (s : ℂ) (hs : 1 < s.re) : starRingEnd ℂ (-Complex.log (1 - c * (n : ℂ) ^ (-(starRingEnd ℂ s)))) = -Complex.log (1 - starRingEnd ℂ c * (n : ℂ) ^ (-s))

theorem **_root_.GlobalTheory.eulerProduct_ne_zero_one_of_entire** : ∀ (ι : Type) (N : ι → ℕ) (c : ι → ℂ), (∀ i, 2 ≤ N i) → (∀ i, ‖c i‖ ≤ 1) → (∀ σ : ℝ, 1 < σ → Summable fun i => (N i : ℝ) ^ (-σ)) → Nonempty ι → ∀ Z L : ℂ → ℂ, Differentiable ℂ Z → Differentiable ℂ L → (∀ s : ℂ, 1 < s.re → Z s = (s - 1) * Complex.exp (∑' i, -Complex.log (1 - 1 * ((N i : ℕ) : ℂ) ^ (-s)))) → (∀ s : ℂ, 1 < s.re → L s = Complex.exp (∑' i, -Complex.log (1 - c i * ((N i : ℕ) : ℂ) ^ (-s)))) → L 1 ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EvalComplexPolyEqZeroOnRatZariskiClosure

private lemma **ringEquiv_eval_eq_eval_map** {σ : Type} (e : ℂ ≃+* ℂ) (x : σ → ℂ) (p : MvPolynomial σ ℂ) : e (MvPolynomial.eval x p) = MvPolynomial.eval (fun i => e (x i)) (MvPolynomial.map (e : ℂ →+* ℂ) p)

private lemma **ev_eq_evC_map_algebraMap** {ι : Type} (p : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι ℂ) : _root_.GlobalTheory.ClassFin.ev p g = _root_.GlobalTheory.ClassFin.evC (MvPolynomial.map (algebraMap ℚ ℂ) p) g

private lemma **mem_span_rational_of_vanishing** {ι : Type} (Γ : Set (Matrix ι ι ℂ)) (hΓ : ∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) (f : MvPolynomial (ι × ι) ℂ) (hf : ∀ g ∈ Γ, _root_.GlobalTheory.ClassFin.evC f g = 0) : f ∈ Submodule.span ℂ {w | ∃ p ∈ _root_.GlobalTheory.ClassFin.vanI Γ, MvPolynomial.map (algebraMap ℚ ℂ) p = w}

theorem **_root_.GlobalTheory.ClassFin.evC_eq_zero_of_mem_pts_of_stable** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ g ∈ Γ, _root_.GlobalTheory.ClassFin.evC f g = 0) → ∀ y ∈ _root_.GlobalTheory.ClassFin.pts Γ ℂ, _root_.GlobalTheory.ClassFin.evC f y = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EvalComplexPolyEqZeroOnRatZariskiClosureStatement

theorem **_root_.GlobalTheory.T.evalComplexPoly_eq_zero_on_ratZariskiClosure** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ g ∈ Γ, evC f g = 0) → ∀ y ∈ pts Γ ℂ, evC f y = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EvenNormsOfUnitNonsquare

theorem **evenNorms_of_unit_nonsquare** {K : Type} [Field K] (w : Valuation K ℤᵐ⁰) (hsq : ∀ x : K, w (x - 1) < 1 → IsSquare x) (δ : K) (hδ : w δ = 1) (hn : ¬ IsSquare δ) : EvenNorms w δ

theorem **evenNorms_of_odd_place** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), Valued.v (2 : v.adicCompletion F) = 1 → ∀ δ : v.adicCompletion F, Valued.v δ = 1 → ¬ IsSquare δ → EvenNorms Valued.v δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EvenOrderNormsOneAddFourMulOfNoRoot

private lemma **exists_exp_two_mul_of_eq_sq** {K : Type} [Field K] (w : Valuation K ℤᵐ⁰) {u y : K} (hy : y ≠ 0) (h : w u = w (y ^ 2)) : ∃ n : ℤ, w u = WithZero.exp (2 * n)

theorem **_root_.GlobalTheory.QNorm.evenNorms_one_add_four_mul** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (ρ : K), (2 : K) ≠ 0 → w ρ ≤ 1 → (∀ x : K, w x ≤ 1 → w (x ^ 2 + x - ρ) = 1) → _root_.GlobalTheory.QNorm.EvenNorms w (1 + 4 * ρ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EvenOrderNormsOneAddFourMulOfNoRootStatement

theorem **_root_.GlobalTheory.T.evenOrderNorms_one_add_four_mul_of_no_root** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (ρ : K), (2 : K) ≠ 0 → w ρ ≤ 1 → (∀ x : K, w x ≤ 1 → w (x ^ 2 + x - ρ) = 1) → EvenNorms w (1 + 4 * ρ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.EventuallyFinEqOne

theorem **eq_one_of_pow_near** {z : ℂ} (hz : ‖z‖ = 1) (h : ∀ k : ℕ, ‖z ^ k - 1‖ < 1) : z = 1

theorem **eventually_fin_eq_one** {F : Type} [Field F] [NumberField F] {ψ : AddChar (AF F) ℂ} (hψ : IsCharAF F ψ) : ∀ᶠ c : Af F in 𝒩 0, ψ (pairA F 0 c) = 1

theorem **isOpen_annSub** (hψ : IsCharAF D.host.F ψ) {a : WO S T} (ha : a.map (infPart D) = 0) : IsOpen (annSub ψ S T a : Set (WO S T))

theorem **isTheta_heisO_fin** (hψ : IsCharAF D.host.F ψ) {a : WO S T} (ha : a.map (infPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (heisO ψ S T a f) where

theorem **isTheta_sum_heisO_fin** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D), IsCharAF D.host.F ψ → ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → ∀ (ι : Type) (s : Finset ι) (a : ι → WO S T) (c : ι → ℂ), (∀ i ∈ s, (a i).map (infPart D) = 0) → IsThetaFunO ψ S T (∑ i ∈ s, c i • heisO ψ S T (a i) f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAdelicUnitaryTranslateStandardLattice

theorem **tv_nsmul_left** (d : ℕ) (a : FiniteAdeleRing (𝒪 F) F) (x : Fin n → E) : tv (d • a) x = d • tv a x

theorem **tv_nsmul_right** (d : ℕ) (a : FiniteAdeleRing (𝒪 F) F) (x : Fin n → E) : tv a (d • x) = d • tv a x

theorem **tv_mul_mem** {L : Set (Fin n → E)} {a b : FiniteAdeleRing (𝒪 F) F} (ha : ∀ v, a v ∈ v.adicCompletionIntegers F) (hb : ∀ v, b v ∈ v.adicCompletionIntegers F) {x : Fin n → E} (hx : x ∈ L) : tv (a * b) x ∈ hat F L

theorem **glue_mulVec_tv** (e a : FiniteAdeleRing (𝒪 F) F) (γ : Matrix (Fin n) (Fin n) E) (x : Fin n → E) : glue e γ *ᵥ tv a x = tv (e * a) (γ *ᵥ x) + tv ((1 − e) * a) x

theorem **fm_mul** (γ δ : Matrix (Fin n) (Fin n) E) : fm F γ * fm F δ = fm F (γ * δ)

theorem **fm_one** : fm F (1 : Matrix (Fin n) (Fin n) E) = 1

theorem **glue_mul** {e : FiniteAdeleRing (𝒪 F) F} (he : e * e = e) {γ δ : Matrix (Fin n) (Fin n) E} (h : γ * δ = 1) : glue e γ * glue e δ = 1

theorem **glue_map_sg** (c : E ≃ₐ[F] E) (e : FiniteAdeleRing (𝒪 F) F) (γ : Matrix (Fin n) (Fin n) E) : ((glue e γ).map (sg (FiniteAdeleRing (𝒪 F) F) c))ᵀ = glue e (γ.map c)ᵀ

theorem **isUnA_of** (c : E ≃ₐ[F] E) {g : Matrix (Fin n) (Fin n) (RA F E)} (h : (g.map (sg (FiniteAdeleRing (𝒪 F) F) c))ᵀ * g = 1) : IsUnA c (1 : Matrix (Fin n) (Fin n) E) g

theorem **exists_div** (d : ℕ) (e : FiniteAdeleRing (𝒪 F) F) (he : ∀ v, e v ∈ v.adicCompletionIntegers F) (hd : ∀ v : HeightOneSpectrum (𝒪 F), (d : 𝒪 F) ∈ v.asIdeal → e v = 0) : ∃ a : FiniteAdeleRing (𝒪 F) F, (∀ v, a v ∈ v.adicCompletionIntegers F) ∧ d • a = e

theorem **idem_apply** (k : ℕ) (v : HeightOneSpectrum (𝒪 F)) : idem F k v = if (k : 𝒪 F) ∈ v.asIdeal then 1 else 0

theorem **idem_mul** (k : ℕ) : idem F k * idem F k = idem F k

theorem **idem_int** (k : ℕ) (v : HeightOneSpectrum (𝒪 F)) : idem F k v ∈ v.adicCompletionIntegers F

theorem **exists_div_idem** {k d : ℕ} (h : d.Coprime k) : ∃ a : FiniteAdeleRing (𝒪 F) F, (∀ v, a v ∈ v.adicCompletionIntegers F) ∧ d • a = idem F k

theorem **exists_div_one_sub_idem** (k : ℕ) : ∃ a : FiniteAdeleRing (𝒪 F) F, (∀ v, a v ∈ v.adicCompletionIntegers F) ∧ k • a = 1 − idem F k

theorem **glue_mem_hat** {L₁ L₂ : Set (Fin n → E)} {k k' : ℕ} (hk : k'.Coprime k) {γ : Matrix (Fin n) (Fin n) E} (h₁ : ∀ x ∈ L₁, k' • γ *ᵥ x ∈ L₂) (h₂ : ∀ x ∈ L₁, k • x ∈ L₂) : ∀ z ∈ hat F L₁, glue (idem F k) γ *ᵥ z ∈ hat F L₂

theorem **exists_factor** {k M : ℕ} (h : k.Coprime M) (D : ℕ) (hD : 0 < D) : ∃ D₁ D₂ : ℕ, D = D₁ * D₂ ∧ D₁.Coprime M ∧ D₂.Coprime k

theorem **congAtA_glue** (L : Submodule (𝒪 E) (Fin n → E)) {k k' M : ℕ} (hk : k'.Coprime k) (hM : k.Coprime M) {γ : Matrix (Fin n) (Fin n) E} (h : ∀ x ∈ L, (k * k') • γ *ᵥ x ∈ L) (D : ℕ) (hD : 0 < D) : CongAtA F L M D (glue (idem F k) γ)

theorem **span_eq_top** (Λ : GenLat E n) (hU : Λ.Unimod) : Submodule.span E (Λ.La : Set (Fin n → E)) = ⊤

theorem **span_smul_le_iff** (k : ℕ) (L₁ L₂ : Submodule (𝒪 E) (Fin n → E)) : Ideal.span {(k : 𝒪 E)} • L₁ ≤ L₂ ↔ ∀ x ∈ L₁, k • x ∈ L₂

theorem **exists_pos** (Λ : GenLat E n) (hU : Λ.Unimod) {M : ℕ} (hT : Λ.TrivAt M) : ∃ k : ℕ, 0 < k ∧ k.Coprime M ∧ (∀ x ∈ Λ.La, k • x ∈ stdLat E n) ∧ ∀ x ∈ stdLat E n, k • x ∈ Λ.La

theorem **exists_adelic_translate** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (n M : ℕ) (Λ : GenLat E n), 0 < M → Λ.Unimod → Λ.TrivAt M → ∃ g : GL (Fin n) (RA (maximalRealSubfield E) E), IsUnA (IsCMField.complexConj E) (1 : Matrix (Fin n) (Fin n) E) g.1 ∧ (Λ.La : Set (Fin n → E)) = latOf (maximalRealSubfield E) g.1 (stdLat E n : Set (Fin n → E)) ∧ ∀ D : ℕ, 0 < D → CongAtA (maximalRealSubfield E) (stdLat E n) M D g.1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAdelicUnitaryTranslateStandardLatticeStatement

theorem **_root_.GlobalTheory.T.exists_adelic_unitary_translate_standardLattice** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (n M : ℕ) (Λ : GenLat E n), 0 < M → Λ.Unimod → Λ.TrivAt M → ∃ g : GL (Fin n) (RA (maximalRealSubfield E) E), IsUnA (IsCMField.complexConj E) (1 : Matrix (Fin n) (Fin n) E) g.1 ∧ (Λ.La : Set (Fin n → E)) = latOf (maximalRealSubfield E) g.1 (stdLat E n : Set (Fin n → E)) ∧ ∀ D : ℕ, 0 < D → CongAtA (maximalRealSubfield E) (stdLat E n) M D g.1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAdmissiblePlacesContainingPrimeSupersetMon

theorem **exists_admissiblePlaces_superset_and_mono** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F], 0 < p → (∀ S₀ : Finset (HeightOneSpectrum (𝒪 F)), ∃ S, S₀ ⊆ S ∧ AdmP p F S) ∧ ∀ S T : Finset (HeightOneSpectrum (𝒪 F)), S ⊆ T → AdmP p F S → AdmP p F T

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAdmissiblePlacesContainingPrimeSupersetMonStatement

theorem **_root_.GlobalTheory.T.exists_admissiblePlacesContainingPrime_superset_mono** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F], 0 < p → (∀ S₀ : Finset (HeightOneSpectrum (𝒪 F)), ∃ S, S₀ ⊆ S ∧ AdmP p F S) ∧ ∀ S T : Finset (HeightOneSpectrum (𝒪 F)), S ⊆ T → AdmP p F S → AdmP p F T

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAdmissibleSupersetFinitePlaces

theorem **intValuation_of_span_eq_mul** {R : Type} [CommRing R] [IsDedekindDomain R] (v : HeightOneSpectrum R) (I : Ideal R) (hI : I ≠ 0) (hv : ¬ v.asIdeal ∣ I) (x : R) (hx0 : x ≠ 0) (hx : v.asIdeal * I = Ideal.span {x}) : v.intValuation x = WithZero.exp (−1)

theorem **exists_admissible_superset_finite_places** : ∀ (F : Type) [Field F] [NumberField F] (S₀ : Finset (HeightOneSpectrum (𝒪 F))), ∃ S, S₀ ⊆ S ∧ admitted F S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAnisotropicOrthogonalBasisHermitian

theorem **exists_anisotropic_orthogonal_basis** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∃ x : Fin n → Fin n → E, (∀ i, hermE c G (x i) (x i) ≠ 0) ∧ (∀ i j, i ≠ j → hermE c G (x i) (x j) = 0) ∧ Submodule.span E (Set.range x) = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAnisotropicOrthogonalBasisHermitianStatement

theorem **_root_.GlobalTheory.T.exists_anisotropic_orthogonal_basis_hermitian** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∃ x : Fin n → Fin n → E, (∀ i, hermE c G (x i) (x i) ≠ 0) ∧ (∀ i j, i ≠ j → hermE c G (x i) (x j) = 0) ∧ Submodule.span E (Set.range x) = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsApproxModLocalSquares

theorem **_root_.GlobalTheory.QNorm.exists_approx_mod_squares** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (y : ∀ v : HeightOneSpectrum (𝒪 F), v.adicCompletion F) (z : (F →+* ℝ) → ℝ), (∀ v, y v ≠ 0) → (∀ σ, z σ ≠ 0) → ∃ x : F, x ≠ 0 ∧ (∀ v ∈ S, IsSquare (algebraMap F (v.adicCompletion F) x * y v)) ∧ ∀ σ : F →+* ℝ, 0 < σ x * z σ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsApproxModLocalSquaresStatement

theorem **_root_.GlobalTheory.T.exists_approx_mod_local_squares** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (y : ∀ v : HeightOneSpectrum (𝒪 F), v.adicCompletion F) (z : (F →+* ℝ) → ℝ), (∀ v, y v ≠ 0) → (∀ σ, z σ ≠ 0) → ∃ x : F, x ≠ 0 ∧ (∀ v ∈ S, IsSquare (algebraMap F (v.adicCompletion F) x * y v)) ∧ ∀ σ : F →+* ℝ, 0 < σ x * z σ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsApproxModPowers

theorem **mul_pow_mem_nhds** {K : Type*} [NontriviallyNormedField K] [CompleteSpace K] [CharZero K] {p : ℕ} (hp : 0 < p) {y : K} (hy : y ≠ 0) : {a : K | a ≠ 0 ∧ ∃ r : K, a = y * r ^ p} ∈ nhds y

theorem **exists_approx_mod_powers** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (y : ∀ v : HeightOneSpectrum (𝒪 F), v.adicCompletion F), 0 < p → (∀ v, y v ≠ 0) → ∃ x : F, x ≠ 0 ∧ ∀ v ∈ S, ∃ r : v.adicCompletion F, algebraMap F (v.adicCompletion F) x = y v * r ^ p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsApproxModPowersStatement

theorem **_root_.GlobalTheory.T.exists_approx_mod_powers'** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (y : ∀ v : HeightOneSpectrum (𝒪 F), v.adicCompletion F), 0 < p → (∀ v, y v ≠ 0) → ∃ x : F, x ≠ 0 ∧ ∀ v ∈ S, ∃ r : v.adicCompletion F, algebraMap F (v.adicCompletion F) x = y v * r ^ p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsAuxiliaryPlacesSquarePatternsSUnits

theorem **mem_sqs_sUnits_iff_isSquare** (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (x : Fˣ) (hx : x ∈ sUnits F S) : x ∈ sqs (sUnits F S) ↔ IsSquare (x : F)

private theorem **relIndex_adjoinRoot_ne_zero_and_le_two** {G : Type} [CommGroup G] (Q : Subgroup G) (d : G) (hd : d * d ∈ Q) : Q.relIndex (_root_.GlobalTheory.QNorm.adjoinRoot Q d hd) ≠ 0 ∧ Q.relIndex (_root_.GlobalTheory.QNorm.adjoinRoot Q d hd) ≤ 2

private theorem **valued_algebraMap_eq_one_of_mem_sUnits** {F : Type} [Field F] [NumberField F] {S : Finset (HeightOneSpectrum (𝒪 F))} {v : HeightOneSpectrum (𝒪 F)} (hv : v ∉ S) {x : Fˣ} (hx : x ∈ _root_.GlobalTheory.QNorm.sUnits F S) : Valued.v (algebraMap F (v.adicCompletion F) (x : F)) = 1

private theorem **isSquare_mul_iff_of_mem_sUnits** {F : Type} [Field F] [NumberField F] {S : Finset (HeightOneSpectrum (𝒪 F))} (hS : _root_.GlobalTheory.QNorm.admitted F S) {v : HeightOneSpectrum (𝒪 F)} (hv : v ∉ S) {x y : Fˣ} (hx : x ∈ _root_.GlobalTheory.QNorm.sUnits F S) (hy : y ∈ _root_.GlobalTheory.QNorm.sUnits F S) : IsSquare (algebraMap F (v.adicCompletion F) ((x * y : Fˣ) : F)) ↔ (IsSquare (algebraMap F (v.adicCompletion F) (x : F)) ↔ IsSquare (algebraMap F (v.adicCompletion F) (y : F)))

private theorem **relIndex_sqSub_le_two_of_not_mem** {F : Type} [Field F] [NumberField F] {S : Finset (HeightOneSpectrum (𝒪 F))} (hS : _root_.GlobalTheory.QNorm.admitted F S) {v : HeightOneSpectrum (𝒪 F)} (hv : v ∉ S) : (_root_.GlobalTheory.QNorm.sqSub (algebraMap F (v.adicCompletion F))).relIndex (_root_.GlobalTheory.QNorm.sUnits F S) ≤ 2

private theorem **relIndex_iInf_sqSub_le_two_pow_card** {F : Type} [Field F] [NumberField F] {S : Finset (HeightOneSpectrum (𝒪 F))} (hS : _root_.GlobalTheory.QNorm.admitted F S) (T : Finset (HeightOneSpectrum (𝒪 F))) (hT : ∀ v ∈ T, v ∉ S) : (⨅ v ∈ T, _root_.GlobalTheory.QNorm.sqSub (algebraMap F (v.adicCompletion F))).relIndex (_root_.GlobalTheory.QNorm.sUnits F S) ≤ 2 ^ T.card

theorem **_root_.GlobalTheory.QNorm.exists_auxiliary_places** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), _root_.GlobalTheory.QNorm.admitted F S → δ ∈ _root_.GlobalTheory.QNorm.sUnits F S → ¬ IsSquare (δ : F) → ∃ T : Finset (HeightOneSpectrum (𝒪 F)), Disjoint S T ∧ T.card + 1 = S.card + Fintype.card (InfinitePlace F) ∧ (∀ v ∈ T, IsSquare (algebraMap F (v.adicCompletion F) δ)) ∧ ∀ A : Finset (HeightOneSpectrum (𝒪 F)), ∃ u ∈ _root_.GlobalTheory.QNorm.sUnits F S, ∀ v ∈ T, (IsSquare (algebraMap F (v.adicCompletion F) u) ↔ v ∉ A)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsCmAdeleTensorEquivMixedFinite

theorem **exists_adele_tensor_equiv** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsTotallyReal F → IsTotallyComplex E → Module.finrank F E = 2 → (∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) → letI : TopologicalSpace (AdeleRing (𝒪 F) F) := TopologicalSpace (AdeleRing (𝒪 F) F ⊗[F] E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsCmAdeleTensorEquivMixedFiniteStatement

theorem **_root_.GlobalTheory.T.exists_cm_adele_tensor_equiv_mixed_finite** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsTotallyReal F → IsTotallyComplex E → Module.finrank F E = 2 → (∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) → letI : TopologicalSpace (AdeleRing (𝒪 F) F) := TopologicalSpace (AdeleRing (𝒪 F) F ⊗[F] E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsCommonDenominatorNilpotentCoboundary

theorem **exists_denominator_of_isField_localization** (B A : Type) [CommRing B] [CommRing A] [Algebra ℚ B] [Algebra B A] [Algebra.FiniteType B A] [IsReduced A] (M : Submonoid B) (hF : IsField (Localization M)) (f : A) (hf : IsNilpotent (f ⊗ₜ[B] (1 : A) - (1 : A) ⊗ₜ[B] f)) : ∃ m ∈ M, ∃ b' : B, algebraMap B A m * f = algebraMap B A b'

theorem **exists_common_denominator_of_isNilpotent** : ∀ (B A : Type) [CommRing B] [CommRing A] [Algebra ℚ B] [Algebra B A] [Algebra.FiniteType B A] [IsReduced A] [Nontrivial B], Function.Injective (algebraMap B A) → ∀ s : Finset A, (∀ f ∈ s, IsNilpotent (f ⊗ₜ[B] (1 : A) - (1 : A) ⊗ₜ[B] f)) → ∃ b : B, b ≠ 0 ∧ ∀ f ∈ s, ∃ b' : B, algebraMap B A b * f = algebraMap B A b'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsCommonDenominatorNilpotentCoboundaryStatement

theorem **_root_.GlobalTheory.T.exists_common_denominator_nilpotent_coboundary** : ∀ (B A : Type) [CommRing B] [CommRing A] [Algebra ℚ B] [Algebra B A] [Algebra.FiniteType B A] [IsReduced A] [Nontrivial B], Function.Injective (algebraMap B A) → ∀ s : Finset A, (∀ f ∈ s, IsNilpotent (f ⊗ₜ[B] (1 : A) - (1 : A) ⊗ₜ[B] f)) → ∃ b : B, b ≠ 0 ∧ ∀ f ∈ s, ∃ b' : B, algebraMap B A b * f = algebraMap B A b'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsComplexPolyBlockDiagPartOfFibreConst

lemma **evC_add** (p q : MvPolynomial (ι × ι) ℂ) (y : Matrix ι ι ℂ) : evC (p + q) y = evC p y + evC q y

lemma **evC_mul** (p q : MvPolynomial (ι × ι) ℂ) (y : Matrix ι ι ℂ) : evC (p * q) y = evC p y * evC q y

lemma **evC_smul** (c : ℂ) (p : MvPolynomial (ι × ι) ℂ) (y : Matrix ι ι ℂ) : evC (c • p) y = c * evC p y

lemma **evC_rtr** (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : evC (rtr g f) y = evC f (y * g)

lemma **evC_ltr** (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : evC (ltr g f) y = evC f (g * y)

lemma **evC_map** (p : MvPolynomial (ι × ι) ℚ) (y : Matrix ι ι ℂ) : evC (MvPolynomial.map (algebraMap ℚ ℂ) p) y = ev p y

lemma **exists_fg_rtr** (f : MvPolynomial (ι × ι) ℂ) : ∃ V : Submodule ℂ (MvPolynomial (ι × ι) ℂ), V.FG ∧ ∀ g, rtr g f ∈ V

lemma **exists_finset_rtr** (Γ : Set (Matrix ι ι ℂ)) (f : MvPolynomial (ι × ι) ℂ) : ∃ s : Finset (MvPolynomial (ι × ι) ℂ), (∀ p ∈ s, ∃ g ∈ Γ, p = rtr g f) ∧ ∀ g ∈ Γ, rtr g f ∈ Submodule.span ℂ (s : Set (MvPolynomial (ι × ι) ℂ))

lemma **exists_common_zero** (D : Ideal (MvPolynomial (ι × ι) ℂ)) (hD : D ≠ ⊤) : ∃ z : Matrix ι ι ℂ, ∀ b ∈ D, evC b z = 0

theorem **GlobalTheory.ClassFin.exists_evC_blk_of_fibre_const** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ x ∈ Γ, ∀ x' ∈ Γ, blk e x = blk e x' → evC f x = evC f x') → ∃ b : MvPolynomial (ι × ι) ℂ, ∀ x ∈ Γ, evC f x = evC b (blk e x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsComplexPolyBlockDiagPartOfFibreConstStatement

theorem **_root_.GlobalTheory.T.exists_complexPoly_blockDiagPart_of_fibre_const** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ x ∈ Γ, ∀ x' ∈ Γ, blk e x = blk e x' → evC f x = evC f x') → ∃ b : MvPolynomial (ι × ι) ℂ, ∀ x ∈ Γ, evC f x = evC b (blk e x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsCongrIffTraceFormsIsometric

theorem **finiteDimensional_of_involution** (h2 : (2 : E) ≠ 0) (hc : ∀ t, c (c t) = t) (hF : ∀ t : E, c t = t ↔ ∃ s : F, algebraMap F E s = t) : FiniteDimensional F E

theorem **exists_traceForm** (hF : ∀ t : E, c t = t ↔ ∃ s : F, algebraMap F E s = t) {V : Type*} [AddCommGroup V] [Module E V] [Module F V] [IsScalarTower F E V] {B : V →ₗ[E] V →ₛₗ[c] E} (hB : ∀ x y, c (B x y) = B y x) : ∃ b : V →ₗ[F] V →ₗ[F] F, ∀ x y, algebraMap F E (b x y) = B x y + B y x

theorem **trace_eq_of_apply_self_eq** {V W : Type*} [AddCommGroup V] [Module E V] [AddCommGroup W] [Module E W] {B : V →ₗ[E] V →ₛₗ[c] E} {B' : W →ₗ[E] W →ₛₗ[c] E} {φ : V → W} (hadd : ∀ x y, φ (x + y) = φ x + φ y) (hφ : ∀ x, B' (φ x) (φ x) = B x x) (x y : V) : B' (φ x) (φ y) + B' (φ y) (φ x) = B x y + B y x

theorem **exists_map_smul_eq** (h2 : (2 : E) ≠ 0) (hc : ∀ t, c (c t) = t) (hF : ∀ t : E, c t = t ↔ ∃ s : F, algebraMap F E s = t) {V W : Type*} [AddCommGroup V] [Module E V] [Module F V] [IsScalarTower F E V] [AddCommGroup W] [Module E W] [Module F W] [IsScalarTower F E W] {B : V →ₗ[E] V →ₛₗ[c] E} {B' : W →ₗ[E] W →ₛₗ[c] E} (hB' : ∀ x y, c (B' x y) = B' y x) (φ : V →ₗ[F] W) (hφ : ∀ x, B' (φ x) (φ x) = B x x) {e : V} (he : B e e ≠ 0) : ∃ φ' : V →ₗ[F] W, (∀ x, B' (φ' x) (φ' x) = B x x) ∧ ∀ t : E, φ' (t • e) = t • φ e

theorem **exists_linearMap_isometry** (h2 : (2 : E) ≠ 0) (hc : ∀ t, c (c t) = t) (hF : ∀ t : E, c t = t ↔ ∃ s : F, algebraMap F E s = t) (d : ℕ) : ∀ {V : Type u} {W : Type v} [AddCommGroup V] [Module E V] [Module F V] [IsScalarTower F E V] [FiniteDimensional E V] [AddCommGroup W] [Module E W] [Module F W] [IsScalarTower F E W] (B : V →ₗ[E] V →ₛₗ[c] E) (B' : W →ₗ[E] W →ₛₗ[c] E), (∀ x y, c (B x y) = B y x) → (∀ x y, c (B' x y) = B' y x) → finrank E V = d → ∀ φ : V →ₗ[F] W, (∀ x, B' (φ x) (φ x) = B x x) → ∃ ψ : V →ₗ[E] W, ∀ x y, B' (ψ x) (ψ y) = B x y

theorem **GlobalTheory.LocalQuad.exists_congr_iff_trace_forms_isometric** {F E : Type*} [Field F] [Field E] [Algebra F E] (h2 : (2 : E) ≠ 0) (c : E →+* E) (hc : ∀ t, c (c t) = t) (hF : ∀ t : E, c t = t ↔ ∃ s : F, algebraMap F E s = t) {n : Type*} [Fintype n] [DecidableEq n] {G G' : Matrix n n E} (hG : ∀ a b, c (G a b) = G b a) (hG' : ∀ a b, c (G' a b) = G' b a) (hd : G.det ≠ 0) (hd' : G'.det ≠ 0) : (∃ P : Matrix n n E, P.det ≠ 0 ∧ G' = Pᵀ * G * P.map c) ↔ ∃ φ : (n → E) ≃ₗ[F] (n → E), ∀ x, x ᵥ· G' *ᵥ (c ∘ x) = φ x ᵥ· G *ᵥ (c ∘ φ x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsCongrIffTraceFormsIsometricCm

theorem **isotropic_ternary_of_units** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (h2 : Valued.v (2 : v.adicCompletion F) = 1) (a : Fin 3 → v.adicCompletion F) (ha : ∀ i, Valued.v (a i) = 1) : ∃ x : Fin 3 → v.adicCompletion F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **isotropic_adicCompletion_of_five_le** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) {ι : Type*} [Fintype ι] (hι : 5 ≤ Fintype.card ι) (a : ι → v.adicCompletion F) : ∃ x : ι → v.adicCompletion F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **exists_norm_over_extension** (H : GlobalTheory.Published.HasseNormSt) (F : Type) [Field F] [NumberField F] (L : Type) [Field L] [NumberField L] [Algebra F L] {α β γ : F} (hα : α ≠ 0) (hβ : β ≠ 0) (hγ : γ ≠ 0) {s : L} (hs : s ^ 2 = algebraMap F L (α * β * γ)) (hr : ∀ σ : F →+* ℝ, ∃ x y z t : ℝ, (x ≠ 0 ∨ y ≠ 0 ∨ z ≠ 0 ∨ t ≠ 0) ∧ x ^ 2 - σ α * y ^ 2 - σ β * z ^ 2 + σ γ * t ^ 2 = 0) (hf : ∀ v : HeightOneSpectrum (𝒪 F), ∃ x y z t : v.adicCompletion F, (x ≠ 0 ∨ y ≠ 0 ∨ z ≠ 0 ∨ t ≠ 0) ∧ x ^ 2 - algebraMap F _ α * y ^ 2 - algebraMap F _ β * z ^ 2 + algebraMap F _ γ * t ^ 2 = 0) : ∃ X Y : L, algebraMap F L β = X ^ 2 - algebraMap F L α * Y ^ 2

theorem **isotropic_four** (H : GlobalTheory.Published.HasseNormSt) (F : Type) [Field F] [NumberField F] (a : Fin 4 → F) (ha : ∀ i, a i ≠ 0) (hr : ∀ σ : F →+* ℝ, ∃ x : Fin 4 → ℝ, x ≠ 0 ∧ ∑ i, σ (a i) * x i ^ 2 = 0) (hf : ∀ v : HeightOneSpectrum (𝒪 F), ∃ x : Fin 4 → v.adicCompletion F, x ≠ 0 ∧ ∑ i, algebraMap F (v.adicCompletion F) (a i) * x i ^ 2 = 0) : ∃ x : Fin 4 → F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **exists_value_isSquare_div** (F : Type) [Field F] [NumberField F] {a a' : F} (ha : a ≠ 0) (T : Finset (HeightOneSpectrum (𝒪 F))) (p q : (v : T) → v.1.adicCompletion F) (hpq : ∀ v, algebraMap F _ a * p v ^ 2 + algebraMap F _ a' * q v ^ 2 ≠ 0) (p' q' : (F →+* ℝ) → ℝ) (hpq' : ∀ σ, σ a * p' σ ^ 2 + σ a' * q' σ ^ 2 ≠ 0) : ∃ x y : F, a * x ^ 2 + a' * y ^ 2 ≠ 0 ∧ (∀ v : T, IsSquare (algebraMap F (v.1.adicCompletion F) (a * x ^ 2 + a' * y ^ 2) / (algebraMap F _ a * p v ^ 2 + algebraMap F _ a' * q v ^ 2))) ∧ ∀ σ : F →+* ℝ, IsSquare (σ (a * x ^ 2 + a' * y ^ 2) / (σ a * p' σ ^ 2 + σ a' * q' σ ^ 2))

theorem **finite_setOf_valued_ne_one** (F : Type) [Field F] [NumberField F] {c : F} (hc : c ≠ 0) : {v : HeightOneSpectrum (𝒪 F) | Valued.v (algebraMap F (v.adicCompletion F) c) ≠ 1}.Finite

theorem **isotropic_add_four** (H : GlobalTheory.Published.HasseNormSt) (F : Type) [Field F] [NumberField F] (n : ℕ) (a : Fin (n + 4) → F) (ha : ∀ i, a i ≠ 0) (hr : ∀ σ : F →+* ℝ, ∃ x : Fin (n + 4) → ℝ, x ≠ 0 ∧ ∑ i, σ (a i) * x i ^ 2 = 0) (hf : ∀ v : HeightOneSpectrum (𝒪 F), ∃ x : Fin (n + 4) → v.adicCompletion F, x ≠ 0 ∧ ∑ i, algebraMap F (v.adicCompletion F) (a i) * x i ^ 2 = 0) : ∃ x : Fin (n + 4) → F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **isotropic_of_locally_isotropic** (H : GlobalTheory.Published.HasseNormSt) (F : Type) [Field F] [NumberField F] {ι : Type*} [Fintype ι] (hι : 4 ≤ Fintype.card ι) (a : ι → F) (ha : ∀ i, a i ≠ 0) (hr : ∀ σ : F →+* ℝ, ∃ x : ι → ℝ, x ≠ 0 ∧ ∑ i, σ (a i) * x i ^ 2 = 0) (hf : ∀ v : HeightOneSpectrum (𝒪 F), ∃ x : ι → v.adicCompletion F, x ≠ 0 ∧ ∑ i, algebraMap F (v.adicCompletion F) (a i) * x i ^ 2 = 0) : ∃ x : ι → F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **isotropic_of_indefinite** (H : GlobalTheory.Published.HasseNormSt) (F : Type) [Field F] [NumberField F] {ι : Type*} [Fintype ι] (a : ι → F) (hι : 5 ≤ Fintype.card ι) (ha : ∀ i, a i ≠ 0) (hr : ∀ σ : F →+* ℝ, ∃ i j, σ (a i) < 0 ∧ 0 < σ (a j)) : ∃ x : ι → F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **landherr_of_hasseNormSt** (H : GlobalTheory.Published.HasseNormSt) : GlobalTheory.Published.Landherr

theorem **trace_forms_and_landherr** : (∀ {F : Type u} {E : Type v} [Field F] [Field E] [Algebra F E], (2 : E) ≠ 0 → ∀ c : E →+* E, (∀ t, c (c t) = t) → (∀ t : E, c t = t ↔ ∃ s : F, algebraMap F E s = t) → ∀ {n : Type w} [Fintype n] [DecidableEq n] {G G' : Matrix n n E}, (∀ a b, c (G a b) = G b a) → (∀ a b, c (G' a b) = G' b a) → G.det ≠ 0 → G'.det ≠ 0 → ((∃ P : Matrix n n E, P.det ≠ 0 ∧ G' = Pᵀ * G * P.map c) ↔ ∃ φ : (n → E) ≃ₗ[F] (n → E), ∀ x, x ·ᵥ G' *ᵥ (c ∘ x) = φ x ·ᵥ G *ᵥ (c ∘ φ x))) ∧ (GlobalTheory.Published.HasseNormSt → ∀ (F : Type) [Field F] [NumberField F] {ι : Type w} [Fintype ι], 4 ≤ Fintype.card ι → ∀ a : ι → F, (∀ i, a i ≠ 0) → (∀ σ : F →+* ℝ, ∃ x : ι → ℝ, x ≠ 0 ∧ ∑ i, σ (a i) * x i ^ 2 = 0) → (∀ v : HeightOneSpectrum (𝒪 F), ∃ x : ι → v.adicCompletion F, x ≠ 0 ∧ ∑ i, algebraMap F (v.adicCompletion F) (a i) * x i ^ 2 = 0) → ∃ x : ι → F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0) ∧ (Global

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsConjHarmOfSpan

private lemma **unitary_conj_pow** {ι : Type} [Fintype ι] [DecidableEq ι] (U D : Matrix ι ι ℂ) (hU : U * Uᴴ = 1) (n : ℕ) : (U * D * Uᴴ) ^ n = U * D ^ n * Uᴴ

private lemma **aeval_unitary_conj_eq_conj_aeval** {ι : Type} [Fintype ι] [DecidableEq ι] (U D : Matrix ι ι ℂ) (hU : U * Uᴴ = 1) (p : ℂ[X]) : aeval (U * D * Uᴴ) p = U * aeval D p * Uᴴ

private lemma **aeval_diagonal_adelic_lattice** {ι : Type} [Fintype ι] [DecidableEq ι] (f : ι → ℂ) (p : ℂ[X]) : aeval (diagonal f) p = diagonal fun i => p.eval (f i)

private lemma **exists_interp_aeval_eq_of_isLog** {ι : Type} [Fintype ι] [DecidableEq ι] {H A : Matrix ι ι ℂ} (h : _root_.GlobalTheory.ClassFin.IsLog H A) : ∃ S : Finset ℝ, ∀ p : ℂ[X], (∀ x ∈ S, p.eval ((Real.exp x : ℝ) : ℂ) = x) → aeval A p = H

private lemma **exists_poly_eval_exp_eq** (S : Finset ℝ) : ∃ p : ℂ[X], ∀ x ∈ S, p.eval ((Real.exp x : ℝ) : ℂ) = x

private lemma **aeval_conj_eq_conj_aeval** {ι : Type} [Fintype ι] [DecidableEq ι] (k D : Matrix ι ι ℂ) (hk : IsUnit k.det) (p : ℂ[X]) : aeval (k⁻¹ * D * k) p = k⁻¹ * aeval D p * k

private lemma **isLog_eq_conj_of_conj** {ι : Type} [Fintype ι] [DecidableEq ι] {H A H' B k : Matrix ι ι ℂ} (hH : _root_.GlobalTheory.ClassFin.IsLog H A) (hH' : _root_.GlobalTheory.ClassFin.IsLog H' B) (hk : IsUnit k.det) (hB : B = k⁻¹ * A * k) : H' = k⁻¹ * H * k

private lemma **isLog_neg_inv** {ι : Type} [Fintype ι] [DecidableEq ι] {H A : Matrix ι ι ℂ} (hH : _root_.GlobalTheory.ClassFin.IsLog H A) : _root_.GlobalTheory.ClassFin.IsLog (-H) A⁻¹

private lemma **isLog_isHermitian** {ι : Type} [Fintype ι] [DecidableEq ι] {H A : Matrix ι ι ℂ} (hH : _root_.GlobalTheory.ClassFin.IsLog H A) : Hᴴ = H

private lemma **eq_zero_of_trace_zero_of_re_trace_squareZero** {ι : Type} [Fintype ι] [DecidableEq ι] (M : Matrix ι ι ℂ) (hM : Mᴴ = M) (htr : M.trace = 0) (hV : ∀ N : Matrix ι ι ℂ, N * N = 0 → Complex.re (Matrix.trace (M * (N + Nᴴ))) = 0) : M = 0

private lemma **norm_add_add_add_le** (a b c d : ℂ) : ‖a + b + c + d‖ ≤ ‖a‖ + ‖b‖ + ‖c‖ + ‖d‖

private lemma **re_trace_isLog_sq_le_second_order** {ι : Type} [Fintype ι] [DecidableEq ι] (H A Y1 Y2 Y3 Y4 : Matrix ι ι ℂ) (hH : _root_.GlobalTheory.ClassFin.IsLog H A) : ∃ K ε : ℝ, 0 < ε ∧ ∀ t : ℝ, |t| < ε → ∀ H' : Matrix ι ι ℂ, _root_.GlobalTheory.ClassFin.IsLog H' (A + ((t : ℂ) • Y1 + (t : ℂ) ^ 2 • Y2 + (t : ℂ) ^ 3 • Y3 + (t : ℂ) ^ 4 • Y4)) → Complex.re (Matrix.trace (H' * H')) ≤ Complex.re (Matrix.trace (H * H)) + t * (2 * Complex.re (Matrix.trace (H * A⁻¹ * Y1))) + t ^ 2 * K

private lemma **gram_expansion_quartic_in_t** {ι : Type} [Fintype ι] [DecidableEq ι] (h N : Matrix ι ι ℂ) (hNN : N * N = 0) (t : ℝ) : ((1 + (t : ℂ) • N) * h * (1 - (t : ℂ) • N))ᴴ * ((1 + (t : ℂ) • N) * h * (1 - (t : ℂ) • N)) - hᴴ * h = (t : ℂ) • (hᴴ * (N + Nᴴ) * h - Nᴴ * (hᴴ * h) - hᴴ * h * N) + (t : ℂ) ^ 2 • (hᴴ * Nᴴ * N * h - Nᴴ * (hᴴ * (N + Nᴴ) * h) - hᴴ * (N + Nᴴ) * h * N + Nᴴ * (hᴴ * h) * N) + (t : ℂ) ^ 3 • (-(Nᴴ * (hᴴ * Nᴴ * N * h)) - hᴴ * Nᴴ * N * h * N + Nᴴ * (hᴴ * (N + Nᴴ) * h) * N) + (t : ℂ) ^ 4 • (Nᴴ * (hᴴ * Nᴴ * N * h) * N)

private lemma **eq_zero_of_nonneg_linear_add_quadratic** {a K ε : ℝ} (hε : 0 < ε) (h : ∀ t : ℝ, |t| < ε → 0 ≤ t * a + t ^ 2 * K) : a = 0

private lemma **isLog_commute** {ι : Type} [Fintype ι] [DecidableEq ι] {H A : Matrix ι ι ℂ} (hH : _root_.GlobalTheory.ClassFin.IsLog H A) : H * A = A * H

theorem **_root_.GlobalTheory.ClassFin.exists_conj_harm_of_span** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (τ : σ ≃ σ) (g : σ → Matrix ι ι ℂ), (∀ s, g (τ s) * g s = 1) → Submodule.span ℂ (Set.range g) = ⊤ → ∃ P : Matrix ι ι ℂ, IsUnit P.det ∧ _root_.GlobalTheory.ClassFin.Harm (fun s => P * g s * P⁻¹)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsConjHarmOfSpanStatement

theorem **_root_.GlobalTheory.T.exists_conj_harm_of_span'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (τ : σ ≃ σ) (g : σ → Matrix ι ι ℂ), (∀ s, g (τ s) * g s = 1) → Submodule.span ℂ (Set.range g) = ⊤ → ∃ P : Matrix ι ι ℂ, IsUnit P.det ∧ Harm (fun s => P * g s * P⁻¹)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsContinuousCircleCharExtension

theorem **circle_hom_extend_of_injective** {A B : Type} [CommGroup A] [CommGroup B] (i : A →* B) (hi : Function.Injective i) (ψ : A →* Circle) : ∃ χ : B →* Circle, ∀ a, χ (i a) = ψ a

theorem **exists_continuous_circle_char_extension** : ∀ (G : Type) [CommGroup G] [TopologicalSpace G] [IsTopologicalGroup G] (H N : Subgroup G) (ψ : H →* Circle), IsOpen (H : Set G) → Continuous ψ → (∀ x : H, (x : G) ∈ N → ψ x = 1) → ∃ χ : G →* Circle, Continuous χ ∧ (∀ x ∈ N, χ x = 1) ∧ ∀ x : H, χ x = ψ x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsContinuousCircleCharExtensionStatement

theorem **_root_.GlobalTheory.T.exists_continuous_circle_char_extension'** : ∀ (G : Type) [CommGroup G] [TopologicalSpace G] [IsTopologicalGroup G] (H N : Subgroup G) (ψ : H →* Circle), IsOpen (H : Set G) → Continuous ψ → (∀ x : H, (x : G) ∈ N → ψ x = 1) → ∃ χ : G →* Circle, Continuous χ ∧ (∀ x ∈ N, χ x = 1) ∧ ∀ x : H, χ x = ψ x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsDenominatorPrimesPrescribed

theorem **exists_pow_eq_span_singleton** (E : Type) [Field E] [NumberField E] (a : Ideal (𝒪 E)) (ha : a ≠ ⊥) : ∃ (h : ℕ) (s : 𝒪 E), h ≠ 0 ∧ a ^ h = Ideal.span {s}

theorem **exists_denominator_primes_prescribed** : ∀ (E : Type) [Field E] [NumberField E] (Q : Ideal (𝒪 E) → Prop) (ι : Type) [Finite ι] (x : ι → E), (∀ i, ∀ P : HeightOneSpectrum (𝒪 E), ¬ Q P.asIdeal → P.valuation E (x i) ≤ 1) → ∃ s : 𝒪 E, s ≠ 0 ∧ (∀ P : Ideal (𝒪 E), P.IsPrime → s ∈ P → Q P) ∧ ∀ i, ∃ b : 𝒪 E, (s : E) * x i = (b : E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsDualFamilySelfDualHermitianSpace

theorem **exists_extension** {O E V : Type} [CommRing O] [IsDomain O] [Field E] [Algebra O E] [IsFractionRing O E] [AddCommGroup V] [Module E V] [Module O V] [IsScalarTower O E V] (Λ : Submodule O V) (hspan : Submodule.span E (Λ : Set V) = ⊤) (φ : Λ →ₗ[O] O) : ∃ ψ : V →ₗ[E] E, ∀ x : Λ, ψ x = algebraMap O E (φ x)

theorem **exists_dual_family** {E : Type} [Field E] [NumberField E] [StarRing E] {r : ℕ} (G : Matrix (Fin r) (Fin r) E) (hG : ∀ i j, star (G i j) = G j i) (hdet : IsUnit G.det) (Λ : Submodule (𝒪 E) (Fin r → E)) (hfg : Λ.FG) (hspan : Submodule.span E (Λ : Set (Fin r → E)) = ⊤) (hsd : ∀ y : Fin r → E, (∀ x ∈ Λ, ∃ b : 𝒪 E, (b : E) = ∑ i, ∑ j, star (x i) * G i j * y j) → y ∈ Λ) : ∃ (n : ℕ) (s s' : Fin n → Fin r → E), (∀ i, s i ∈ Λ ∧ s' i ∈ Λ) ∧ ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * G k j = if a = j then 1 else 0

theorem **dualFamily** : ∀ (D : D7) (S : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)), IsSelfDualC S Λ → ∃ (n : ℕ) (s s' : Fin n → Fin S.rank → D.host.E), (∀ i, s i ∈ Λ ∧ s' i ∈ Λ) ∧ ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * S.gram k j = if a = j then 1 else 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsDualFamilySelfDualHermitianSpaceStatement

theorem **_root_.GlobalTheory.T.exists_dual_family_selfDual_hermitianSpace** : ∀ (D : D7) (S : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)), IsSelfDualC S Λ → ∃ (n : ℕ) (s s' : Fin n → Fin S.rank → D.host.E), (∀ i, s i ∈ Λ ∧ s' i ∈ Λ) ∧ ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * S.gram k j = if a = j then 1 else 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsEnergyMin

theorem **energy_nonneg** {H A : Matrix ι ι ℂ} (h : IsLog H A) : 0 ≤ (Matrix.trace (H * H)).re

theorem **isCompact_box** (C : ℝ) : IsCompact {x : Matrix ι ι ℂ | ∀ i j, ‖x i j‖ ≤ C}

theorem **exists_det_one** (P : Matrix ι ι ℂ) (hP : IsUnit P.det) : ∃ Q : Matrix ι ι ℂ, Q.det = 1 ∧ ∀ x : Matrix ι ι ℂ, Q * x * Q⁻¹ = P * x * P⁻¹

theorem **inv_eq_adj** {P : Matrix ι ι ℂ} (hP : P.det = 1) : P⁻¹ = P.adjugate

theorem **GlobalTheory.ClassFin.exists_energy_min** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (g : σ → Matrix ι ι ℂ), (∀ s, IsUnit (g s).det) → Submodule.span ℂ (Set.range g) = ⊤ → ∃ (P : Matrix ι ι ℂ) (H : σ → Matrix ι ι ℂ), IsUnit P.det ∧ (∀ s, IsLog (H s) ((P * g s * P⁻¹)ᴴ * (P * g s * P⁻¹))) ∧ ∀ (P' : Matrix ι ι ℂ) (H' : σ → Matrix ι ι ℂ), IsUnit P'.det → (∀ s, IsLog (H' s) ((P' * g s * P'⁻¹)ᴴ * (P' * g s * P'⁻¹))) → ∑ s, (Matrix.trace (H s * H s)).re ≤ ∑ s, (Matrix.trace (H' s * H' s)).re

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsEntireMellinSmoothedDirichletSum

lemma **weight_bound** {wt : ℝ → ℂ} (hc : Continuous wt) {a b : ℝ} (hs : ∀ y, wt y ≠ 0 → a ≤ y ∧ y ≤ b) : ∃ M : ℝ, 0 ≤ M ∧ ∀ y : ℝ, 0 < y → ‖wt y‖ ≤ M * y ^ (-2 : ℝ)

lemma **weight_mellinConvergent** {wt : ℝ → ℂ} (hc : Continuous wt) {a b : ℝ} (ha : 0 < a) (hs : ∀ y, wt y ≠ 0 → a ≤ y ∧ y ≤ b) (s : ℂ) : MellinConvergent wt s

lemma **continuousOn_sum** {ι : Type} {a : ι → ℂ} {N : ι → ℝ} {wt : ℝ → ℂ} (hc : Continuous wt) {α β : ℝ} (hs : ∀ y, wt y ≠ 0 → α ≤ y ∧ y ≤ β) (hN : ∀ i, a i ≠ 0 → 1 ≤ N i) (hsum : Summable fun i => ‖a i‖ * N i ^ (-2 : ℝ)) : ContinuousOn (fun t : ℝ => ∑' i, a i * wt (t * N i)) (Ioi 0)

lemma **integral_norm_comp** (wt : ℝ → ℂ) (s : ℂ) {n : ℝ} (hn : 0 < n) : ∫ t in Ioi (0 : ℝ), ‖(t : ℂ) ^ (s - 1) • wt (t * n)‖ = n ^ (-s.re) * ∫ t in Ioi (0 : ℝ), ‖(t : ℂ) ^ (s - 1) • wt t‖

lemma **mellin_sum** {ι : Type} [Countable ι] {a : ι → ℂ} {N : ι → ℝ} {wt : ℝ → ℂ} {s : ℂ} (hw : MellinConvergent wt s) (hN : ∀ i, a i ≠ 0 → 1 ≤ N i) (hsum : Summable fun i => ‖a i‖ * N i ^ (-s.re)) : mellin (fun t : ℝ => ∑' i, a i * wt (t * N i)) s = mellin wt s * ∑' i, a i * (N i : ℂ) ^ (-s)

theorem **exists_entire_mellin** : ∀ (ι : Type) [Countable ι] (a : ι → ℂ) (N : ι → ℝ) (R : ℂ) (wt : ℝ → ℂ), ContDiff ℝ ∞ wt → (∃ a b : ℝ, 0 < a ∧ ∀ y, wt y ≠ 0 → a ≤ y ∧ y ≤ b) → (∀ i, a i ≠ 0 → 1 ≤ N i) → (∀ σ : ℝ, 1 < σ → Summable fun i => ‖a i‖ * N i ^ (-σ)) → (∀ A : ℕ, ∃ C : ℝ, ∀ t : ℝ, 0 < t → t ≤ 1 → ‖∑' i, a i * wt (t * N i) - R / (t : ℂ)‖ ≤ C * t ^ A) → ∃ Z : ℂ → ℂ, Differentiable ℂ Z ∧ Z 1 = R ∧ ∀ s : ℂ, 1 < s.re → Z s = mellin wt s + ((s - 1) * ∑' i, a i * (N i : ℂ) ^ (-s))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsEqPowMulPthPowAdicCompletionUnit

theorem **exists_eq_pow_mul_pow_of_isCyclic** {G : Type*} [CommGroup G] [IsCyclic G] {p : ℕ} (hp : p.Prime) {x : G} (hx : ∀ r : G, r ^ p ≠ x) (y : G) : ∃ (k : ℕ) (r : G), y = x ^ k * r ^ p

theorem **exists_pow_eq_of_residue_pow** {R : Type*} [CommRing R] [IsLocalRing R] [HenselianRing R (maximalIdeal R)] {n : ℕ} (hn : IsUnit (n : R)) {u : R} (hu : IsUnit u) (c : ResidueField R) (hc : c ^ n = residue R u) : ∃ v : R, v ^ n = u

theorem **exists_eq_pow_mul_pow_of_henselian** {R : Type*} [CommRing R] [IsLocalRing R] [HenselianRing R (maximalIdeal R)] [Finite (ResidueField R)] {p : ℕ} (hp : p.Prime) (hn : IsUnit (p : R)) (x y : Rˣ) (hx : ∀ r : R, r ^ p ≠ x) : ∃ (k : ℕ) (r : R), (y : R) = x ^ k * r ^ p

theorem **exists_eq_pow_mul_pow_of_unit** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (x y : v.adicCompletion F), p.Prime → v.valuation F (p : F) = 1 → Valued.v x = 1 → Valued.v y = 1 → (∀ r : v.adicCompletion F, r ^ p ≠ x) → ∃ (k : ℕ) (r : v.adicCompletion F), y = x ^ k * r ^ p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsEqPowMulPthPowAdicCompletionUnitStatement

theorem **_root_.GlobalTheory.T.exists_eq_pow_mul_pth_pow_adicCompletion_unit** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (x y : v.adicCompletion F), p.Prime → v.valuation F (p : F) = 1 → Valued.v x = 1 → Valued.v y = 1 → (∀ r : v.adicCompletion F, r ^ p ≠ x) → ∃ (k : ℕ) (r : v.adicCompletion F), y = x ^ k * r ^ p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsEqTransposeMulMapOfIdempotent

theorem **exists_shear** (G : Matrix ι ι R) (p : ι) (τ : ι → R) (hτ : τ p = 0) : ∃ P : Matrix ι ι R, IsUnit P.det ∧ ∀ x y, (Pᵀ * G * P.map c) x y = G x y + τ x * G p y + c (τ y) * G x p + τ x * c (τ y) * G p p

theorem **hermitian_congr** (hc : ∀ x, c (c x) = x) {G : Matrix ι ι R} (hG : ∀ a b, c (G a b) = G b a) (P : Matrix ι ι R) (a b : ι) : c ((Pᵀ * G * P.map c) a b) = (Pᵀ * G * P.map c) b a

theorem **det_congr** (G P : Matrix ι ι R) : (Pᵀ * G * P.map c).det = P.det * c P.det * G.det

theorem **congr_mul** (G P Q : Matrix ι ι R) : (P * Q)ᵀ * G * (P * Q).map c = Qᵀ * (Pᵀ * G * P.map c) * Q.map c

theorem **exists_congr_symm** {G H P : Matrix ι ι R} (hP : IsUnit P.det) (h : Pᵀ * G * P.map c = H) : ∃ Q : Matrix ι ι R, IsUnit Q.det ∧ G = Qᵀ * H * Q.map c

theorem **mem_of_fixed_of_not_isUnit** (hc : ∀ x, c (c x) = x) (hloc : ∀ x, c x = x → IsUnit x ∨ IsUnit (1 - x)) {x : R} (hx : c x = x) (hu : ¬ IsUnit x) (n : Ideal R) [hn : n.IsMaximal] : x ∈ n

theorem **mem_of_forall_trace_mem** (hc : ∀ x, c (c x) = x) (hw : ∃ w, w + c w = 1) (n : Ideal R) [hn : n.IsMaximal] {g : R} (hg : ∀ t, t * g + c (t * g) ∈ n) : g ∈ n

theorem **exists_congr_insert** (hc : ∀ x, c (c x) = x) (hloc : ∀ x, c x = x → IsUnit x ∨ IsUnit (1 - x)) (hw : ∃ w, w + c w = 1) {G : Matrix ι ι R} (hG : ∀ a b, c (G a b) = G b a) (hdet : IsUnit G.det) (s : Finset ι) (hs : ∀ a ∈ s, ∀ b, b ≠ a → G a b = 0) (hne : ∃ a, a ∉ s) : ∃ p ∉ s, ∃ P : Matrix ι ι R, IsUnit P.det ∧ ∀ a ∈ insert p s, ∀ b, b ≠ a → (Pᵀ * G * P.map c) a b = 0

theorem **exists_eq_diagonal_of_isUnit_or** (hc : ∀ x, c (c x) = x) (hloc : ∀ x, c x = x → IsUnit x ∨ IsUnit (1 - x)) (hw : ∃ w, w + c w = 1) {G : Matrix ι ι R} (hG : ∀ a b, c (G a b) = G b a) (hdet : IsUnit G.det) : ∃ (P : Matrix ι ι R) (d : ι → R), IsUnit P.det ∧ (∀ a, IsUnit (d a)) ∧ (∀ a, c (d a) = d a) ∧ G = Pᵀ * diagonal d * P.map c

theorem **exists_eq_transpose_mul_map_of_isUnit_or** (hc : ∀ x, c (c x) = x) (hloc : ∀ x, c x = x → IsUnit x ∨ IsUnit (1 - x)) (hw : ∃ w, w + c w = 1) (hnorm : ∀ u, c u = u → IsUnit u → ∃ x, x * c x = u) {G : Matrix ι ι R} (hG : ∀ a b, c (G a b) = G b a) (hdet : IsUnit G.det) : ∃ P : Matrix ι ι R, IsUnit P.det ∧ G = Pᵀ * P.map c

theorem **isUnit_or_of_fixed** [IsLocalRing R₀] (hfix : ∀ x, c x = x → x ∈ (algebraMap R₀ R).range) (x : R) (hx : c x = x) : IsUnit x ∨ IsUnit (1 - x)

theorem **exists_eq_transpose_mul_map** [IsLocalRing R₀] (hc : ∀ x, c (c x) = x) (hfix : ∀ x, c x = x → x ∈ (algebraMap R₀ R).range) (hw : ∃ w, w + c w = 1) (hnorm : ∀ u, c u = u → IsUnit u → ∃ x, x * c x = u) {G : Matrix ι ι R} (hG : ∀ a b, c (G a b) = G b a) (hdet : IsUnit G.det) : ∃ P : Matrix ι ι R, IsUnit P.det ∧ G = Pᵀ * P.map c

theorem **exists_binary_eq_of_finite** {k : Type*} [Field k] [Finite k] {a n : k} (hD : a ^ 2 - 4 * n ≠ 0) (u : k) : ∃ r s, r ^ 2 + a * r * s + n * s ^ 2 = u

theorem **exists_add_mul_sq_eq** {R₀ : Type*} [CommRing R₀] (I : Ideal R₀) [HenselianRing R₀ I] (n : R₀) {e : R₀} (he : e ∈ I) : ∃ t, t + n * t ^ 2 = e

theorem **exists_norm_eq_of_henselian** (c : R →+* R) (hc : ∀ x, c (c x) = x) (hbase : ∀ r, c (algebraMap R₀ R r) = algebraMap R₀ R r) (hfix : ∀ x, c x = x → x ∈ (algebraMap R₀ R).range) (hgal : ∃ ω, IsUnit (ω - c ω)) {u : R} (hcu : c u = u) (hu : IsUnit u) : ∃ x, x * c x = u

theorem **exists_eq_transpose_mul_map_of_henselian** (c : R →+* R) (hc : ∀ x, c (c x) = x) (hbase : ∀ r, c (algebraMap R₀ R r) = algebraMap R₀ R r) (hfix : ∀ x, c x = x → x ∈ (algebraMap R₀ R).range) (hgal : ∃ ω, IsUnit (ω - c ω)) {ι : Type*} [Fintype ι] [DecidableEq ι] {G : Matrix ι ι R} (hG : ∀ a b, c (G a b) = G b a) (hdet : IsUnit G.det) : ∃ P : Matrix ι ι R, IsUnit P.det ∧ G = Pᵀ * P.map c

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsForallAbvSubLt

theorem **_root_.GlobalTheory.LocalQuad.Approx.exists_forall_abv_sub_lt** {F : Type*} [Field F] {ι : Type*} [Finite ι] {v : ι → AbsoluteValue F ℝ} (h : ∀ i, (v i).IsNontrivial) (hv : Pairwise fun i j ↦ ¬(v i).IsEquiv (v j)) (z : ι → F) {ε : ℝ} (hε : 0 < ε) : ∃ x : F, ∀ i, v i (z i - x) < ε

theorem **_root_.GlobalTheory.LocalQuad.Approx.abv_inl** (T : Finset (HeightOneSpectrum (𝒪 F))) (v : T) (x : F) : _root_.GlobalTheory.LocalQuad.Approx.abv T (Sum.inl v) x = ‖algebraMap F (v.1.adicCompletion F) x‖

theorem **_root_.GlobalTheory.LocalQuad.Approx.abv_isNontrivial** (T : Finset (HeightOneSpectrum (𝒪 F))) (i : T ⊕ (F →+* ℝ)) : (_root_.GlobalTheory.LocalQuad.Approx.abv T i).IsNontrivial

private theorem **not_isEquiv_adicAbv_real_place** (v : HeightOneSpectrum (𝒪 F)) (σ : F →+* ℝ) : ¬(NumberField.HeightOneSpectrum.adicAbv F v).IsEquiv (InfinitePlace.mk (Complex.ofRealHom.comp σ)).1

theorem **_root_.GlobalTheory.LocalQuad.Approx.abv_pairwise** (T : Finset (HeightOneSpectrum (𝒪 F))) : Pairwise fun i j ↦ ¬(_root_.GlobalTheory.LocalQuad.Approx.abv T i).IsEquiv (_root_.GlobalTheory.LocalQuad.Approx.abv T j)

theorem **_root_.GlobalTheory.LocalQuad.denseRange_finite_real** (F : Type) [Field F] [NumberField F] (T : Finset (HeightOneSpectrum (𝒪 F))) : DenseRange (fun x : F => ((fun v : T => algebraMap F (v.1.adicCompletion F) x, fun σ : F →+* ℝ => σ x) : ((v : T) → v.1.adicCompletion F) × ((F →+* ℝ) → ℝ)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsGaloisConjugatesHeckeCharTrivialOnRealIdel

theorem **continuous_unitsDivConjCircleChar** : Continuous arg2

theorem **exists_transport** {G K : Type} [CommGroup G] [TopologicalSpace G] [IsTopologicalGroup G] [Field K] (σ₀ σ : K →+* ℂ) (f : G →* Circle) (hy : ∀ x, ∃ y, ((f x : Circle) : ℂ) = σ₀ y) (m : ℕ) (hm : 0 < m) (hf : ∀ x, f x ^ m = 1) (U : Set G) (hU : U ∈ 𝒩 (1 : G)) (h1 : ∀ x ∈ U, f x = 1) : ∃ ψ : G →* Circle, Continuous ψ ∧ ∀ x y, ((f x : Circle) : ℂ) = σ₀ y → ((ψ x : Circle) : ℂ) = σ y

theorem **one_mem_intAE** (D : D7) : (1 : Af D.host.F ⊗[D.host.F] D.host.E) ∈ intAE D

theorem **mul_mem_intAE** (D : D7) {x y : Af D.host.F ⊗[D.host.F] D.host.E} (hx : x ∈ intAE D) (hy : y ∈ intAE D) : x * y ∈ intAE D

theorem **angle_coe** (D : D7) (θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ) (x : (AE D)ˣ) : ((angle D θ x : Circle) : ℂ) = θ (infPart D (x : AE D)) / starRingEnd ℂ (θ (infPart D (x : AE D)))

theorem **continuous_angle** (D : D7) {θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ} (h : Continuous θ) : Continuous (angle D θ)

theorem **coe_prod_zpow** {ι : Type} [Fintype ι] (u : ι → Circle) (k : ι → ℤ) : ((∏ i, u i ^ k i : Circle) : ℂ) = ∏ i, (u i : ℂ) ^ k i

theorem **exists_type** (D : D7) (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (hθ : IsInfFam θ) (χ : (AE D)ˣ →* Circle) (hc : Continuous χ) (hF : ∀ a, χ (idF D a) = 1) : ∃ k : D.what₂.Tau → ℤ, ∀ x : (AE D)ˣ, finPart D (x : AE D) = 1 → χ x = ∏ τ, angle D (θ τ) x ^ k τ

theorem **angle_idF** (D : D7) {θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ} (h : ∀ z, θ (sigR (Ainf D.host.F) D.sp.cm.sig z) = starRingEnd ℂ (θ z)) (a : (AF D.host.F)ˣ) : angle D θ (idF D a) = 1

theorem **angle_inc** (D : D7) {τ : D.host.E →+* ℂ} {θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ} (h : Formalism.MainNotions.StandardLattice.IsInfEmb τ θ) (e : D.host.Eˣ) : ((angle D θ (Units.map (incA D).toMonoidHom e) : Circle) : ℂ) = τ ((e : D.host.E) / IsCMField.complexConj D.host.E (e : D.host.E))

theorem **exists_conjugates** : ∀ (D : D7) (Q : Par D) (χ₀ : (AE D)ˣ →* Circle), IsHeckeChar D χ₀ → (∀ a, χ₀ (idF D a) = 1) → (∀ x, FinUnit x → ∃ y : Q.c.F, ((χ₀ x : Circle) : ℂ) = Q.c.F.val.toRingHom y) → ∃ χ : (Q.c.F →+* ℂ) → (AE D)ˣ →* Circle, χ Q.c.F.val.toRingHom = χ₀ ∧ ∀ σ, IsHeckeChar D (χ σ) ∧ (∀ a, χ σ (idF D a) = 1) ∧ MuConj Q.c.F.val.toRingHom σ χ₀ (χ σ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsGaloisConjugatesHeckeCharTrivialOnRealIdelStatement

theorem **_root_.GlobalTheory.T.exists_galoisConjugates_heckeChar_trivialOnRealIdeles** : ∀ (D : D7) (Q : Par D) (χ₀ : (AE D)ˣ →* Circle), IsHeckeChar D χ₀ → (∀ a, χ₀ (idF D a) = 1) → (∀ x, FinUnit x → ∃ y : Q.c.F, ((χ₀ x : Circle) : ℂ) = Q.c.F.val.toRingHom y) → ∃ χ : (Q.c.F →+* ℂ) → (AE D)ˣ →* Circle, χ Q.c.F.val.toRingHom = χ₀ ∧ ∀ σ, IsHeckeChar D (χ σ) ∧ (∀ a, χ σ (idF D a) = 1) ∧ MuConj Q.c.F.val.toRingHom σ χ₀ (χ σ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsGlobalUniformizerResidueRepresentatives

theorem **exists_uniformizer_and_residue_representatives** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), (∃ π : F, v.valuation F π = WithZero.exp (-1) ∧ Valued.v (algebraMap F (v.adicCompletion F) π) = WithZero.exp (-1)) ∧ (∀ x : F, Valued.v (algebraMap F (v.adicCompletion F) x) = v.valuation F x) ∧ (∀ x : v.adicCompletion F, Valued.v x ≤ 1 → ∃ a : 𝒪 F, Valued.v (x - algebraMap F (v.adicCompletion F) (a : F)) < 1) ∧ ∃ R : Finset (v.adicCompletion F), R.card ≤ Ideal.absNorm v.asIdeal ∧ (∀ r ∈ R, Valued.v r ≤ 1) ∧ ∀ x : v.adicCompletion F, Valued.v x ≤ 1 → ∃ r ∈ R, Valued.v (x - r) < 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsGlobalUniformizerResidueRepresentativesStatement

theorem **_root_.GlobalTheory.T.exists_global_uniformizer_residue_representatives** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), (∃ π : F, v.valuation F π = WithZero.exp (-1) ∧ Valued.v (algebraMap F (v.adicCompletion F) π) = WithZero.exp (-1)) ∧ (∀ x : F, Valued.v (algebraMap F (v.adicCompletion F) x) = v.valuation F x) ∧ (∀ x : v.adicCompletion F, Valued.v x ≤ 1 → ∃ a : 𝒪 F, Valued.v (x - algebraMap F (v.adicCompletion F) (a : F)) < 1) ∧ ∃ R : Finset (v.adicCompletion F), R.card ≤ Ideal.absNorm v.asIdeal ∧ (∀ r ∈ R, Valued.v r ≤ 1) ∧ ∀ x : v.adicCompletion F, Valued.v x ≤ 1 → ∃ r ∈ R, Valued.v (x - r) < 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsGroupHomRangeEqOfAutStable

theorem **IsPre.exists_group_range** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ∃ (G : Type) (_ : Group G) (ρ : G →* Matrix ι ι ℂ), Set.range ρ = Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsGroupHomRangeEqOfAutStableStatement

theorem **_root_.GlobalTheory.T.exists_group_hom_range_eq_of_isPre** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ∃ (G : Type) (_ : Group G) (ρ : G →* Matrix ι ι ℂ), Set.range ρ = Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsHermiteReducedLatticeBasis

theorem **exists_lt_two_mul** {S : Type} [Nonempty S] (f : S → ℝ) (c : ℝ) (hc : 0 < c) (hf : ∀ s, c ≤ f s) : ∃ s, ∀ t, f s < 2 * f t

theorem **prod_castLE_succ** {N : ℕ} (f : Fin N → ℝ) (p q : Fin N) (hq : (q : ℕ) = p + 1) (h : (q : ℕ) ≤ N) : ∏ i : Fin q, f (Fin.castLE h i) = (∏ l ∈ Finset.univ.filter (· < p), f l) * f p

theorem **mul_upper** {N : ℕ} (n m : Matrix (Fin N) (Fin N) ℝ) (hn1 : ∀ i, n i i = 1) (hn0 : ∀ i j, j < i → n i j = 0) (hm1 : ∀ i, m i i = 1) (hm0 : ∀ i j, j < i → m i j = 0) : (∀ i, (n * m) i i = 1) ∧ ∀ i j, j < i → (n * m) i j = 0

theorem **le_of_swap** {N : ℕ} (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n : Matrix (Fin N) (Fin N) ℝ) (hk : kᵀ * k = 1) (ha : ∀ i, 0 < a i) (hn1 : ∀ i, n i i = 1) (hn0 : ∀ i j, j < i → n i j = 0) (hν : ∀ i j, i < j → |n i j| ≤ 1 / 2) (p q : Fin N) (hq : (q : ℕ) = p + 1) (hlt : ∏ j, gramMinor (k * diagonal a * n) j < 2 * ∏ j, gramMinor ((k * diagonal a * n).submatrix id (Equiv.swap p q)) j) : a p ≤ 2 * a q

theorem **exists_reduced** : ∀ (N : ℕ) (g : Matrix (Fin N) (Fin N) ℝ), g.det ≠ 0 → ∃ (γ : Matrix (Fin N) (Fin N) ℤ) (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n : Matrix (Fin N) (Fin N) ℝ), (γ.det = 1 ∨ γ.det = -1) ∧ kᵀ * k = 1 ∧ (∀ i, 0 < a i) ∧ (∀ i, n i i = 1) ∧ (∀ i j, j < i → n i j = 0) ∧ (∀ i j, i < j → |n i j| ≤ 1 / 2) ∧ (∀ p q : Fin N, (q : ℕ) = p + 1 → a p ≤ 2 * a q) ∧ g * γ.map (Int.cast : ℤ → ℝ) = k * diagonal a * n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIndependentPrimeRelIndexSubgroups

theorem **relIndex_eq_prime_of_generated** {G : Type} [CommGroup G] {p : ℕ} (hp : p.Prime) (E K : Subgroup G) (x : G) (hxE : x ∈ E) (hxK : x ∉ K) (hxp : x ^ p ∈ K) (hgen : ∀ y ∈ E, ∃ k : ℕ, y / x ^ k ∈ K) : K.relIndex E = p

theorem **exists_independent_prime_relIndex_subgroups** (p : ℕ) (G ι : Type) [CommGroup G] [DecidableEq ι] (U H : Subgroup G) (K : ι → Subgroup G) (good : ι → Prop) (m : ℕ) (hp : p.Prime) (hH : H.relIndex U = p ^ m) (h1 : ∀ i, good i → U ⊓ H ≤ K i) (h2 : ∀ i, good i → ∀ x ∈ U, x ^ p ∈ K i) (h2' : ∀ i, good i → ∀ x ∈ U, x ∉ K i → ∀ y ∈ U, ∃ k : ℕ, y / x ^ k ∈ K i) (h3 : ∀ x ∈ U, x ∉ H → ∃ i, good i ∧ x ∉ K i) (k : ℕ) (hk : k ≤ m) : ∃ T : Finset ι, T.card = k ∧ (∀ i ∈ T, good i) ∧ (∀ i ∈ T, ∃ u ∈ U, u ∉ K i ∧ ∀ j ∈ T, j ≠ i → u ∈ K j) ∧ ∃ D : Subgroup G, (∀ x, x ∈ D ↔ ∀ i ∈ T, x ∈ K i) ∧ D.relIndex U = p ^ k

theorem **exists_indep_prime_index** : ∀ (p : ℕ) (G ι : Type) [CommGroup G] [DecidableEq ι] (U H : Subgroup G) (K : ι → Subgroup G) (good : ι → Prop) (m : ℕ), p.Prime → H.relIndex U = p ^ m → (∀ i, good i → U ⊓ H ≤ K i) → (∀ i, good i → ∀ x ∈ U, x ^ p ∈ K i) → (∀ i, good i → ∀ x ∈ U, x ∉ K i → ∀ y ∈ U, ∃ k : ℕ, y / x ^ k ∈ K i) → (∀ x ∈ U, x ∉ H → ∃ i, good i ∧ x ∉ K i) → ∃ T : Finset ι, T.card = m ∧ (∀ i ∈ T, good i) ∧ ∀ i ∈ T, ∃ u ∈ U, u ∉ K i ∧ ∀ j ∈ T, j ≠ i → u ∈ K j

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsInfiniteAdeleTensorEquivMixedSpace

theorem **mixed_ext** [IsTotallyComplex E] {x y : mixedSpace E} (h : ∀ w, x.2 w = y.2 w) : x = y

theorem **exists_place_over** [IsTotallyComplex E] [IsTotallyReal F] (v : InfinitePlace F) : ∃ w : {w : InfinitePlace E // w.IsComplex}, InfinitePlace.mk (w.1.embedding.comp (algebraMap F E)) = v

theorem **exists_equiv_mixedSpace** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsTotallyReal F → IsTotallyComplex E → Module.finrank F E = 2 → (∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) → letI : TopologicalSpace (InfiniteAdeleRing F ⊗[F] E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsInfiniteAdeleTensorEquivMixedSpaceStatement

theorem **_root_.GlobalTheory.T.exists_infiniteAdele_tensor_equiv_mixedSpace** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsTotallyReal F → IsTotallyComplex E → Module.finrank F E = 2 → (∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) → letI : TopologicalSpace (InfiniteAdeleRing F ⊗[F] E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIntegralSplitIdempotentOverTwo

theorem **idem_alg** {T : Type} [CommRing T] (p U R w m : T) (hU : p * U = U) (hR : R * R - R + m * p = 0) (hδ : U - 2 * (U * R) = p) (hw : w * w - w + m = 0) : (U * w - U * R) * (U * w - U * R) = U * w - U * R ∧ (U * w - U * R) * (U * (1 - w) - U * R) = 0 ∧ (U * w - U * R) + (U * (1 - w) - U * R) = p

theorem **idem_tensor** {F A E K : Type} [Field F] [CommRing A] [Algebra F A] [CommRing E] [Algebra F E] [CommRing K] (φ : K →ₙ+* A) (c : A ⊗[F] E →+* A ⊗[F] E) (ω : E) (m : ℕ) (r u : K) (hc1 : ∀ a : A, c (a ⊗ₜ 1) = a ⊗ₜ 1) (hcω : ∀ a : A, c (a ⊗ₜ ω) = a ⊗ₜ (1 − ω)) (hω : ω * ω − ω + m = 0) (hr : r * r − r + m = 0) (hu : u * (1 − 2 * r) = 1) (e : A ⊗[F] E) (he : e = φ u ⊗ₜ ω + φ (−(u * r)) ⊗ₜ 1) : e * e = e ∧ e * c e = 0 ∧ e + c e = algebraMap A (A ⊗[F] E) (φ 1)

theorem **evalV_singleV_same** (y : Fv F v) : evalV F v (singleV F v y) = y

theorem **evalV_singleV_ne** (y : Fv F v) {w : HeightOneSpectrum (𝒪 F)} (hw : w ≠ v) : evalV F w (singleV F v y) = 0

theorem **af_ext** {a b : Af F} (h : ∀ w, evalV F w a = evalV F w b) : a = b

theorem **singleV_mem_intAf** (y : v.adicCompletionIntegers F) : singleV F v (y : Fv F v) ∈ intAf F

theorem **exists_root** (h2 : (2 : 𝒪 F) ∈ v.asIdeal) (k : ℕ) : ∃ r u : v.adicCompletionIntegers F, r * r − r + ((2 * k : ℕ) : v.adicCompletionIntegers F) = 0 ∧ u * (1 − 2 * r) = 1

theorem **exists_omega** (D : D7) : ∃ (ω : 𝒪 D.host.E) (k : ℕ), (ω : D.host.E) * ω − ω + ((2 * k : ℕ) : D.host.E) = 0 ∧ D.host.conj ω = 1 − ω

theorem **star_eq_conj** (D : D7) (x : D.host.E) : star x = D.host.conj x

theorem **exists_split_idempotent** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)), (2 : 𝒪 D.host.F) ∈ v.asIdeal → ∃ e : Formalism.L2.Theta.AE D, e * e = e ∧ e * cA D e = 0 ∧ e + cA D e = algebraMap (AF D.host.F) (Formalism.L2.Theta.AE D) (pairA D.host.F 0 (singleV D.host.F v 1)) ∧ ∃ α ∈ intAf D.host.F, ∃ β ∈ intAf D.host.F, ∃ ω : 𝒪 D.host.E, e = pairA D.host.F 0 α ⊗ₜ[D.host.F] (ω : D.host.E) + pairA D.host.F 0 β ⊗ₜ[D.host.F] (1 : D.host.E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIntegralSplitIdempotentOverTwoStatement

theorem **_root_.GlobalTheory.T.exists_integral_split_idempotent_over_two** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)), (2 : 𝒪 D.host.F) ∈ v.asIdeal → ∃ e : Formalism.L2.Theta.AE D, e * e = e ∧ e * cA D e = 0 ∧ e + cA D e = algebraMap (AF D.host.F) (Formalism.L2.Theta.AE D) (pairA D.host.F 0 (singleV D.host.F v 1)) ∧ ∃ α ∈ intAf D.host.F, ∃ β ∈ intAf D.host.F, ∃ ω : 𝒪 D.host.E, e = pairA D.host.F 0 α ⊗ₜ[D.host.F] (ω : D.host.E) + pairA D.host.F 0 β ⊗ₜ[D.host.F] (1 : D.host.E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsInverseChart

theorem **exists_inverse_chart** {D : D7} {θ : InfFam D} (hθ : IsInfFam θ) : ∃ σ : (D.what₂.Tau → ℂ) →+ AE D, GlobalTheory.FockCompare.Chart θ σ

theorem **continuous_ray** {ι : Type} (τ : ι) (c : ℂ) : Continuous (ray τ c)

theorem **lineAt_exists** (D : D7) : LineAtExists D

theorem **lineAt_unique** (D : D7) : LineAtUnique D

theorem **lineAt_exists_unique** : ∀ D : D7, LineAtExists D ∧ LineAtUnique D

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsInverseOneAddPreservingAdelicCompletion

theorem **_root_.GlobalTheory.UnRep.Nakayama.mem_jacobson** (a : _root_.GlobalTheory.UnRep.Nakayama.intAdeles F) (ha : ∀ v, Valued.v (a.1 v) < 1) : a ∈ Ideal.jacobson (⊥ : Ideal (_root_.GlobalTheory.UnRep.Nakayama.intAdeles F))

theorem **_root_.GlobalTheory.UnRep.Nakayama.valuation_natCast_lt_one** (N : ℕ) (v : HeightOneSpectrum (𝒪 F)) (hv : (N : 𝒪 F) ∈ v.asIdeal) : Valued.v ((N : ℕ) : v.adicCompletion F) < 1

private lemma **integral_tmul_smul_mem_hat** {S : Set (Fin n → E)} {b : FiniteAdeleRing (𝒪 F) F} (hb : ∀ v, b v ∈ v.adicCompletionIntegers F) {z : Fin n → _root_.GlobalTheory.AdLat.RA F E} (hz : z ∈ _root_.GlobalTheory.AdLat.hat F S) : (b ⊗ₜ[F] (1 : E) : _root_.GlobalTheory.AdLat.RA F E) • z ∈ _root_.GlobalTheory.AdLat.hat F S

private lemma **one_tmul_mem_hat** {S : Set (Fin n → E)} {x : Fin n → E} (hx : x ∈ S) : (fun i ↦ (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] x i) ∈ _root_.GlobalTheory.AdLat.hat F S

theorem **_root_.GlobalTheory.UnRep.exists_inverse_one_add_preserving_hat** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (N : ℕ) (e : FiniteAdeleRing (𝒪 F) F), (∀ v, e v ∈ v.adicCompletionIntegers F) → (∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∉ v.asIdeal → e v = 0) → ∀ X : Matrix (Fin n) (Fin n) (_root_.GlobalTheory.AdLat.RA F E), (∀ z ∈ _root_.GlobalTheory.AdLat.hat F (L : Set (Fin n → E)), X *ᵥ z ∈ _root_.GlobalTheory.AdLat.hat F (L : Set (Fin n → E))) → ∃ W : Matrix (Fin n) (Fin n) (_root_.GlobalTheory.AdLat.RA F E), (1 + (e ⊗ₜ[F] (1 : E) : _root_.GlobalTheory.AdLat.RA F E) • (N • X)) * W = 1 ∧ ∀ z ∈ _root_.GlobalTheory.AdLat.hat F (L : Set (Fin n → E)), W *ᵥ z ∈ _root_.GlobalTheory.AdLat.hat F (L : Set (Fin n → E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsInverseOneAddPreservingAdelicCompletionStatement

theorem **_root_.GlobalTheory.T.exists_inverse_one_add_preserving_adelicCompletion** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (N : ℕ) (e : FiniteAdeleRing (𝒪 F) F), (∀ v, e v ∈ v.adicCompletionIntegers F) → (∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∉ v.asIdeal → e v = 0) → ∀ X : Matrix (Fin n) (Fin n) (RA F E), (∀ z ∈ hat F (L : Set (Fin n → E)), X *ᵥ z ∈ hat F (L : Set (Fin n → E))) → ∃ W : Matrix (Fin n) (Fin n) (RA F E), (1 + (e ⊗ₜ[F] (1 : E) : RA F E) • (N • X)) * W = 1 ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), W *ᵥ z ∈ hat F (L : Set (Fin n → E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsHermitianLogOfPosDefGram

theorem **exists_isLog_gram** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (x : Matrix ι ι ℂ), IsUnit x.det → ∃ H : Matrix ι ι ℂ, IsLog H (xᴴ * x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsHermitianLogOfPosDefGramStatement

theorem **_root_.GlobalTheory.T.exists_isHermitianLogOfPosDef_gram** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (x : Matrix ι ι ℂ), IsUnit x.det → ∃ H : Matrix ι ι ℂ, IsLog H (xᴴ * x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsKum

theorem **exists_isKum** : ∀ (p : ℕ) (F : Type) [Field F] [CharZero F] (ζ δ : F), p.Prime → IsPrimitiveRoot ζ p → (∀ r : F, r ^ p ≠ δ) → ∃ (E : Type) (_ : Field E) (_ : Algebra F E) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsOpenIsCompact

theorem **isOpen_glSub** (S : Subring A) (hS : IsOpen (S : Set A)) : IsOpen (glSub (ι

theorem **isCompact_glSub** [T2Space A] (S : Subring A) (hS : IsCompact (S : Set A)) : IsCompact (glSub (ι

theorem **exists_isOpen_isCompact** : ∀ (F : Type) [Field F] [NumberField F] (n : ℕ), ∃ U : Subgroup (GL (Fin n) (FiniteAdeleRing (𝒪 F) F)), IsOpen (U : Set (GL (Fin n) (FiniteAdeleRing (𝒪 F) F))) ∧ IsCompact (U : Set (GL (Fin n) (FiniteAdeleRing (𝒪 F) F)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsPrimeIdemOfComp

theorem **conj_conj** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) (x : E) : σ (σ x) = x

theorem **exists_of_conj_eq** {σ : E →ₐ[F] E} (h : IsCMExt F E σ) {x : E} (hx : σ x = x) : ∃ t : F, algebraMap F E t = x

theorem **mem_intAE_iff** (z : Af F ⊗[F] E) : z ∈ intAE F E ↔ swapAf F E z ∈ Formalism.L2.DIV.RInt F E

theorem **exists_comp** : ∃ Φ : AE F E →+* FiniteAdeleRing (𝒪 E) E, Comp Φ

theorem **cO_coe** (σ : E →ₐ[F] E) (x : 𝒪 E) : (cO σ x : E) = σ (x : E)

theorem **eq_of_le** {P Q : HeightOneSpectrum (𝒪 E)} (h : P.asIdeal ≤ Q.asIdeal) : P = Q

theorem **cO_cO** (h : IsCMExt F E σ) (x : 𝒪 E) : cO σ (cO σ x) = x

theorem **mem_cP** (h : IsCMExt F E σ) (P : HeightOneSpectrum (𝒪 E)) (x : 𝒪 E) : x ∈ (cP h P).asIdeal ↔ cO σ x ∈ P.asIdeal

theorem **cA_incA** (x : E) : cA σ (incA F E x) = incA F E (σ x)

theorem **cA_cA** (h : IsCMExt F E σ) (z : AE F E) : cA σ (cA σ z) = z

theorem **parts_cA** (z : AE F E) : infPart F E (cA σ z) = sigR (Ainf F) σ (infPart F E z) ∧ finPart F E (cA σ z) = sigR (Af F) σ (finPart F E z)

theorem **finPart_cA_mem** {z : AE F E} (hz : finPart F E z ∈ intAE F E) : finPart F E (cA σ z) ∈ intAE F E

theorem **Comparison₂.conj** (hσ : IsCMExt F E σ) {P : HeightOneSpectrum (𝒪 E)} {e : AE F E} (h : Comparison₂ P e) : Comparison₂ (cP hσ P) (cA σ e)

theorem **inv_mem_iff** (Q : HeightOneSpectrum (𝒪 E)) {x : 𝒪 E} (hx : x ≠ 0) : algebraMap E (Q.adicCompletion E) ((x : E)⁻¹) ∈ Q.adicCompletionIntegers E ↔ x ∉ Q.asIdeal

theorem **under_cP** (hσ : IsCMExt F E σ) (P : HeightOneSpectrum (𝒪 E)) : (cP hσ P).under (𝒪 F) = P.under (𝒪 F)

theorem **eq_or_eq_cP** (hσ : IsCMExt F E σ) {P Q : HeightOneSpectrum (𝒪 E)} (h : Q.under (𝒪 F) = P.under (𝒪 F)) : Q = P ∨ Q = cP hσ P

theorem **exists_single** (K : Type) [Field K] [NumberField K] (v : HeightOneSpectrum (𝒪 K)) : ∃ α : FiniteAdeleRing (𝒪 K) K, α v = 1 ∧ ∀ w, w ≠ v → α w = 0

theorem **exists_placeIdem** (v : HeightOneSpectrum (𝒪 F)) : ∃ ε, IsPlaceIdem F v ε

theorem **ev_mul** (a b : AE F E) (Q : HeightOneSpectrum (𝒪 E)) : Φ (a * b) Q = Φ a Q * Φ b Q

theorem **ev_add** (a b : AE F E) (Q : HeightOneSpectrum (𝒪 E)) : Φ (a + b) Q = Φ a Q + Φ b Q

theorem **ev_zero** (Q : HeightOneSpectrum (𝒪 E)) : Φ 0 Q = 0

theorem **Comp.ext'** {a b : AE F E} (h1 : infPart F E a = infPart F E b) (h2 : ∀ Q, Φ a Q = Φ b Q) : a = b

theorem **exists_delta** (P : HeightOneSpectrum (𝒪 E)) : ∃ e, IsDelta Φ P e

theorem **delta_unique** {P : HeightOneSpectrum (𝒪 E)} {e e' : AE F E} (h : IsDelta Φ P e) (h' : IsDelta Φ P e') : e = e'

theorem **delta_idem** {P : HeightOneSpectrum (𝒪 E)} {e : AE F E} (h : IsDelta Φ P e) : e * e = e

theorem **not_int_iff** {e : AE F E} (he : e * e = e) {x : 𝒪 E} (hx : x ≠ 0) : finPart F E (e * incA F E ((x : E)⁻¹)) ∉ intAE F E ↔ ∃ Q, Φ e Q ≠ 0 ∧ x ∈ Q.asIdeal

theorem **tie_iff** {e : AE F E} (he : e * e = e) (h0 : infPart F E e = 0) (P : HeightOneSpectrum (𝒪 E)) : Comparison₂ P e ↔ IsDelta Φ P e

theorem **delta_conj** (hσ : IsCMExt F E σ) {P : HeightOneSpectrum (𝒪 E)} {e : AE F E} (h : IsDelta Φ P e) : IsDelta Φ (cP hσ P) (cA σ e)

theorem **place_comp** {v : HeightOneSpectrum (𝒪 F)} {ε : AF F} (h : IsPlaceIdem F v ε) : infPart F E (algebraMap _ (AE F E) ε) = 0 ∧ (∀ Q : HeightOneSpectrum (𝒪 E), Q.under (𝒪 F) = v → Φ (algebraMap _ _ ε) Q = 1) ∧ (∀ Q : HeightOneSpectrum (𝒪 E), Q.under (𝒪 F) ≠ v → Φ (algebraMap _ _ ε) Q = 0)

theorem **exists_isPrimeIdem_of_comp** (hσ : IsCMExt F E σ) (P : HeightOneSpectrum (𝒪 E)) : ∃ e f, e * e = e ∧ IsPrimeIdem σ P e f

theorem **exists_isPrimeIdem** (hσ : IsCMExt F E σ) (P : HeightOneSpectrum (𝒪 E)) : ∃ e f, e * e = e ∧ IsPrimeIdem σ P e f

theorem **exists_isUnifIdele** (hσ : IsCMExt F E σ) (P : HeightOneSpectrum (𝒪 E)) : ∃ x, IsUnifIdele σ P x

theorem **chiAt_one** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ P : HeightOneSpectrum (𝒪 E), chiAt σ (1 : (AE F E)ˣ →* Circle) P = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsQGroupWeilRestriction

theorem **ev_map_complexAut** {ι : Type} (f : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι ℂ) (σ : ℂ ≃+* ℂ) : ev f (g.map σ) = σ (ev f g)

theorem **exists_isQGroup_weilRestriction** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)), (∀ (B : Type) [CommRing B] [Algebra F B], (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((1 : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) ∧ (∀ g h : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g * h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0)) ∧ ∀ g : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsQGroupWeilRestrictionStatement

theorem **_root_.GlobalTheory.T.exists_isQGroup_weilRestriction_alt** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)), (∀ (B : Type) [CommRing B] [Algebra F B], (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((1 : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) ∧ (∀ g h : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0) → (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g * h : GL (Fin n) (B)) : Matrix (Fin n) (Fin n) (B)) ij.1 ij.2) P = 0)) ∧ ∀ g : GL (Fin n) B, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (B)) : Matrix

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsQuadraticSqrtExtensionOfNotSquare

theorem **exists_isQuad** : ∀ (F : Type) [Field F] [NumberField F] (δ : F), ¬ IsSquare δ → ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsQuadraticSqrtExtensionOfNotSquareStatement

theorem **_root_.GlobalTheory.T.exists_isQuadraticSqrtExtension_of_not_square** : ∀ (F : Type) [Field F] [NumberField F] (δ : F), ¬ IsSquare δ → ∃ (E : Type) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsRealEmbeddingEqHom

theorem **exists_isReal_embedding_eq_hom** {F : Type} [Field F] (σ : F →+* ℝ) : ∃ (v : InfinitePlace F) (hv : v.IsReal), InfinitePlace.embedding_of_isReal hv = σ

theorem **hasseNorm** : GlobalTheory.Published.HasseNorm

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsUnitFormSelfOfNotMem

theorem **isIntegral_of_involution** (hσ : ∀ x, σ (σ x) = x) (hfix : ∀ x, σ x = x → x ∈ (algebraMap R S).range) : Algebra.IsIntegral R S

theorem **algebraMap_mem_of_isMaximal** [Algebra.IsIntegral R S] (n : Ideal S) [n.IsMaximal] {r : R} (hr : r ∈ maximalIdeal R) : algebraMap R S r ∈ n

theorem **isUnit_or_mem_of_fixed** (hfix : ∀ x, σ x = x → x ∈ (algebraMap R S).range) {x : S} (hx : σ x = x) : IsUnit x ∨ x ∈ (maximalIdeal R).map (algebraMap R S)

theorem **exists_isUnit_add_mul** (hσ : ∀ x, σ (σ x) = x) (hfix : ∀ x, σ x = x → x ∈ (algebraMap R S).range) (hgal : ∃ ω, IsUnit (ω − σ ω)) {u : S} (hu : u * σ u = 1) : ∃ a, IsUnit (a + u * σ a)

theorem **exists_add_eq_one** (hσ : ∀ x, σ (σ x) = x) (hgal : ∃ ω, IsUnit (ω − σ ω)) : ∃ t, t + σ t = 1

theorem **exists_basis_one** [IsLocalRing R] [Module.Free R S] (hrank : finrank R S = 2) : ∃ B : Basis (Fin 2) R S, B 0 = 1

theorem **eq_coord** (B : Basis (Fin 2) R S) (hB : B 0 = 1) (x : S) : x = algebraMap R S (B.repr x 0) + algebraMap R S (B.repr x 1) * B 1

theorem **repr_coord_zero** (B : Basis (Fin 2) R S) (hB : B 0 = 1) (r s : R) : B.repr (algebraMap R S r + algebraMap R S s * B 1) 0 = r

theorem **repr_coord_one** (B : Basis (Fin 2) R S) (hB : B 0 = 1) (r s : R) : B.repr (algebraMap R S r + algebraMap R S s * B 1) 1 = s

theorem **coord_inj** (B : Basis (Fin 2) R S) (hB : B 0 = 1) {r s r' s' : R} (h : algebraMap R S r + algebraMap R S s * B 1 = algebraMap R S r' + algebraMap R S s' * B 1) : r = r' ∧ s = s'

theorem **trace_coord** (B : Basis (Fin 2) R S) (hB : B 0 = 1) {a b : R} (hab : B 1 * B 1 = algebraMap R S b + algebraMap R S a * B 1) (r s : R) : Algebra.trace R S (algebraMap R S r + algebraMap R S s * B 1) = 2 * r + s * a

theorem **norm_coord** (B : Basis (Fin 2) R S) (hB : B 0 = 1) {a b : R} (hab : B 1 * B 1 = algebraMap R S b + algebraMap R S a * B 1) (r s : R) : Algebra.norm R (algebraMap R S r + algebraMap R S s * B 1) = r * (r + s * a) − s * b * s

theorem **isUnit_different** [Algebra.FormallyUnramified R S] (B : Basis (Fin 2) R S) (hB : B 0 = 1) {a b : R} (hab : B 1 * B 1 = algebraMap R S b + algebraMap R S a * B 1) : IsUnit (2 * B 1 − algebraMap R S a)

theorem **apply_eq_of_ne_one** [IsLocalRing R] [Algebra.FormallyUnramified R S] (B : Basis (Fin 2) R S) (hB : B 0 = 1) {a b : R} (hab : B 1 * B 1 = algebraMap R S b + algebraMap R S a * B 1) (σ : S ≃ₐ[R] S) (hσ : σ ≠ 1) : σ (B 1) = algebraMap R S a − B 1

theorem **add_apply_eq_trace** (hrank : finrank R S = 2) (σ : S ≃ₐ[R] S) (hσ : σ ≠ 1) (x : S) : x + σ x = algebraMap R S (Algebra.trace R S x)

theorem **apply_apply** (hrank : finrank R S = 2) (σ : S ≃ₐ[R] S) (hσ : σ ≠ 1) (x : S) : σ (σ x) = x

theorem **exists_isUnit_sub** (hrank : finrank R S = 2) (σ : S ≃ₐ[R] S) (hσ : σ ≠ 1) : ∃ ω, IsUnit (ω − σ ω)

theorem **mem_range_of_fixed** (hrank : finrank R S = 2) (σ : S ≃ₐ[R] S) (hσ : σ ≠ 1) (x : S) (hx : σ x = x) : x ∈ (algebraMap R S).range

theorem **existsUnique_ne_one** (hrank : finrank R S = 2) : ∃! σ : S ≃ₐ[R] S, σ ≠ 1

theorem **trace_surjective** (hrank : finrank R S = 2) : Function.Surjective (Algebra.trace R S)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsometryMatrixRange

theorem **exists_isometry_matrix_range** {ι : Type} [Fintype ι] [DecidableEq ι] (W : Submodule ℂ (ι → ℂ)) : ∃ (n : ℕ) (Q : Matrix ι (Fin n) ℂ), Qᴴ * Q = 1 ∧ ∀ v, v ∈ W ↔ ∃ x, v = Q *ᵥ x

theorem **ss_of_harm** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (τ : σ ≃ σ) (g : σ → Matrix ι ι ℂ), (∀ s, g (τ s) * g s = 1) → Harm g → SS (Set.range g)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsotropicBasisNonorthogonalTwoVectors

theorem **hfL_apply** (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (y x : Fin n → E) : hfL σ G y x = hf σ G x y

theorem **hf_symm** {σ : E →+* E} {G : Matrix (Fin n) (Fin n) E} (hG : IsHerm σ G) (x y : Fin n → E) : σ (hf σ G x y) = hf σ G y x

theorem **hfL_ne** {σ : E →+* E} {G : Matrix (Fin n) (Fin n) E} (hG : IsHerm σ G) {y : Fin n → E} (hy : y ≠ 0) : ∃ x, hfL σ G y x ≠ 0

theorem **hf_comb** {σ : E →+* E} (G : Matrix (Fin n) (Fin n) E) (a b : E) (u w : Fin n → E) : hf σ G (a • u + b • w) (a • u + b • w) = a * σ a * hf σ G u u + a * σ b * hf σ G u w + b * σ a * hf σ G w u + b * σ b * hf σ G w w

theorem **exists_isotropic_avoid** [CharZero E] {σ : E →+* E} {G : Matrix (Fin n) (Fin n) E} (hG : IsHerm σ G) (hz : ∃ z : Fin n → E, z ≠ 0 ∧ hf σ G z z = 0) {ι : Type} [Finite ι] (f : ι → (Fin n → E) →ₗ[E] E) (hf0 : ∀ i, ∃ x, f i x ≠ 0) : ∃ z, hf σ G z z = 0 ∧ ∀ i, f i z ≠ 0

theorem **exists_isotropic_basis** (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (hG : IsHerm σ G) (hz : ∃ z : Fin n → E, z ≠ 0 ∧ hf σ G z z = 0) (x y : Fin n → E) (hx : x ≠ 0) (hy : y ≠ 0) : ∃ z : Fin n → Fin n → E, LinearIndependent E z ∧ ∀ i, hf σ G (z i) (z i) = 0 ∧ hf σ G x (z i) ≠ 0 ∧ hf σ G y (z i) ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsotropicBasisNonorthogonalTwoVectorsStatement

theorem **_root_.GlobalTheory.T.exists_isotropic_basis_nonorthogonal_two_vectors** : ∀ (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), IsHerm σ G → (∃ z : Fin n → E, z ≠ 0 ∧ hf σ G z z = 0) → ∀ x y : Fin n → E, x ≠ 0 → y ≠ 0 → ∃ z : Fin n → Fin n → E, LinearIndependent E z ∧ ∀ i, hf σ G (z i) (z i) = 0 ∧ hf σ G x (z i) ≠ 0 ∧ hf σ G y (z i) ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsIsotropicOfIndefinite

theorem **_root_.GlobalTheory.UTrans.herm_diagonal_self** {r : ℕ} (g u : Fin r → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm (Matrix.diagonal g) u u = ∑ i, u i * g i * _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (u i)

theorem **_root_.GlobalTheory.UTrans.exists_isotropic_diagonal** {n : ℕ} (hn : 3 ≤ n) {g : Fin n → E} (hg : ∀ i, _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (g i) = g i) (hg0 : ∀ i, g i ≠ 0) (hs : ∀ τ : E →+* ℂ, (∃ i, 0 < (τ (g i)).re) ∧ ∃ j, (τ (g j)).re < 0) : ∃ z : Fin n → E, z ≠ 0 ∧ _root_.ReciprocityTheory.ChartRestriction.G.FD.herm (Matrix.diagonal g) z z = 0

private lemma **herm_eq_dotProduct_mulVec_cc** {r : ℕ} (H : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm H x y = x ⬝ᵥ (H *ᵥ fun b => _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (y b))

private lemma **herm_mulVec_mulVec'** {r : ℕ} (G P : Matrix (Fin r) (Fin r) E) (u v : Fin r → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm G (P *ᵥ u) (P *ᵥ v) = _root_.ReciprocityTheory.ChartRestriction.G.FD.herm (Pᵀ * G * P.map _root_.ReciprocityTheory.ChartRestriction.G.FD.cc) u v

private lemma **re_mul_cc_eq_normSq_mul_re** (τ : E →+* ℂ) (y d : E) : (τ (y * d * _root_.ReciprocityTheory.ChartRestriction.G.FD.cc y)).re = Complex.normSq (τ y) * (τ d).re

private lemma **exists_re_pos_of_herm_diagonal_re_pos** {r : ℕ} (τ : E →+* ℂ) (d y : Fin r → E) (hy : 0 < (τ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm (Matrix.diagonal d) y y)).re) : ∃ i, 0 < (τ (d i)).re

private lemma **exists_re_neg_of_herm_diagonal_re_neg** {r : ℕ} (τ : E →+* ℂ) (d y : Fin r → E) (hy : (τ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm (Matrix.diagonal d) y y)).re < 0) : ∃ i, (τ (d i)).re < 0

theorem **_root_.GlobalTheory.UTrans.exists_isotropic_of_indefinite** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), (∀ a, σ a = IsCMField.complexConj E a) → _root_.GlobalTheory.UTrans.IsHerm σ G → 3 ≤ n → (∀ τ : E →+* ℂ, (∃ x, 0 < (τ (_root_.GlobalTheory.UTrans.hf σ G x x)).re) ∧ ∃ y, (τ (_root_.GlobalTheory.UTrans.hf σ G y y)).re < 0) → ∃ z : Fin n → E, z ≠ 0 ∧ _root_.GlobalTheory.UTrans.hf σ G z z = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsLeftInvariantPolySeparatingCoset

lemma **mem_vanC** {S : Set (Matrix ι ι ℂ)} {f : MvPolynomial (ι × ι) ℂ} : f ∈ vanC S ↔ ∀ g ∈ S, evC f g = 0

lemma **evC_ltr** (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : evC (ltr g f) y = evC f (g * y)

lemma **exists_rtr** (h : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : ∃ f' : MvPolynomial (ι × ι) ℂ, ∀ z, evC f' z = evC f (z * h)

lemma **evC_map** (f : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι ℂ) : evC (MvPolynomial.map (algebraMap ℚ ℂ) f) g = ev f g

lemma **mem_of_forall_vanC** {Γ : Set (Matrix ι ι ℂ)} (hΓ : IsQGroup Γ) {z : Matrix ι ι ℂ} (hz : ∀ f ∈ vanC Γ, evC f z = 0) : z ∈ Γ

lemma **vanC_sup_eq_top** {Γ : Set (Matrix ι ι ℂ)} (hΓ : IsQGroup Γ) {y : Matrix ι ι ℂ} (hy : y.det = 1) (hyΓ : y ∉ Γ) : vanC Γ ⊔ vanC ((fun g => g * y) '' Γ) = ⊤

theorem **exists_isInv_evC_one_ne** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ), IsQGroup Γ → IsReynolds Γ Rey → ∀ y : Matrix ι ι ℂ, y.det = 1 → y ∉ Γ → ∃ f, IsInv Γ f ∧ evC f 1 ≠ evC f y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsLeftInvariantPolySeparatingCosetStatement

theorem **_root_.GlobalTheory.T.exists_leftInvariantPoly_separating_coset** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ), IsQGroup Γ → IsReynolds Γ Rey → ∀ y : Matrix ι ι ℂ, y.det = 1 → y ∉ Γ → ∃ f, IsInv Γ f ∧ evC f 1 ≠ evC f y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsLeftTranslationReynoldsOperator

theorem **evC_ltr** (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : evC (ltr g f) y = evC f (g * y)

theorem **isInv_iff** (Γ : Set (Matrix ι ι ℂ)) (f : MvPolynomial (ι × ι) ℂ) : IsInv Γ f ↔ ∀ g ∈ Γ, ltr g f = f

theorem **exists_isReynolds** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∃ Rey, IsReynolds Γ Rey

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsLeftTranslationReynoldsOperatorStatement

theorem **_root_.GlobalTheory.T.exists_leftTranslation_reynoldsOperator** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∃ Rey, IsReynolds Γ Rey

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsLocalNonnormOfNonsquareAdicCompletion

private theorem **exists_even_valuation_of_mul_sq_eq_norm** {K : Type} [Field K] (w : Valuation K (WithZero (Multiplicative ℤ))) (Δ y s p q : K) (hΔ : _root_.GlobalTheory.QNorm.EvenNorms w Δ) (hs : s ≠ 0) (hy : y ≠ 0) (h : y * s ^ 2 = p ^ 2 - Δ * q ^ 2) : ∃ n : ℤ, w y = WithZero.exp (2 * n)

private theorem **not_norm_of_odd_valuation_of_norm_mul** {K : Type} [Field K] (w : Valuation K (WithZero (Multiplicative ℤ))) (δ Δ y c d : K) (hδ : δ ≠ 0) (hΔ : _root_.GlobalTheory.QNorm.EvenNorms w Δ) (hy : y ≠ 0) (hodd : ∀ n : ℤ, w y ≠ WithZero.exp (2 * n)) (hc : y = c ^ 2 - δ * Δ * d ^ 2) : ¬ ∃ a b : K, y = a ^ 2 - δ * b ^ 2

theorem **_root_.GlobalTheory.QNorm.exists_local_nonnorm** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (δ : v.adicCompletion F), δ ≠ 0 → ¬ IsSquare δ → ∃ x : v.adicCompletion F, x ≠ 0 ∧ ¬ ∃ a b : v.adicCompletion F, x = a ^ 2 - δ * b ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsLocalNonnormOfNonsquareAdicCompletionStatement

theorem **_root_.GlobalTheory.T.exists_local_nonnorm_of_nonsquare_adicCompletion** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (δ : v.adicCompletion F), δ ≠ 0 → ¬ IsSquare δ → ∃ x : v.adicCompletion F, x ≠ 0 ∧ ¬ ∃ a b : v.adicCompletion F, x = a ^ 2 - δ * b ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsMinNormTensorOfIsClosedTensorSLOrbit

theorem **tpow_mul'** {ι : Type} [Fintype ι] (d : ℕ) (x y : Matrix ι ι ℂ) : tpow d (x * y) = tpow d x * tpow d y

theorem **tpow_one'** {ι : Type} [Fintype ι] [DecidableEq ι] (d : ℕ) : tpow d (1 : Matrix ι ι ℂ) = 1

theorem **act_mul'** {ι : Type} [Fintype ι] [DecidableEq ι] {m : ℕ} (d : Fin m → ℕ) (v : TIdx ι d → ℂ) (x y : Matrix ι ι ℂ) : act d (act d v x) y = act d v (x * y)

theorem **act_one'** {ι : Type} [Fintype ι] [DecidableEq ι] {m : ℕ} (d : Fin m → ℕ) (v : TIdx ι d → ℂ) : act d v 1 = v

theorem **norm_sq_le_nsq** {κ : Type} [Fintype κ] (v : κ → ℂ) (k : κ) : ‖v k‖ ^ 2 ≤ nsq v

theorem **continuous_nsq** {κ : Type} [Fintype κ] : Continuous (nsq : (κ → ℂ) → ℝ)

theorem **exists_isMinVec_of_isClosed** (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ) (hc : IsClosed (orb d v)) : ∃ w ∈ orb d v, IsMinVec d w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsMinNormTensorOfIsClosedTensorSLOrbitStatement

theorem **_root_.GlobalTheory.T.exists_minNormTensor_of_isClosed_tensorSLOrbit** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ), IsClosed (orb d v) → ∃ w ∈ orb d v, IsMinVec d w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsMultiplierDescendBlockDiagPartOfRatIdempot

theorem **exists_multiplier_descend_of_closed** {σ X : Type} [Fintype σ] [Nonempty X] (p : X → σ → ℂ) (hcl : ∀ y : σ → ℂ, (∀ f : MvPolynomial σ ℂ, (∀ x, MvPolynomial.aeval (p x) f = 0) → MvPolynomial.aeval y f = 0) → ∃ x, p x = y) (q : σ → MvPolynomial σ ℂ) (s : Finset (MvPolynomial σ ℂ)) (hs : ∀ f ∈ s, ∀ x x', (∀ i, MvPolynomial.aeval (p x) (q i) = MvPolynomial.aeval (p x') (q i)) → MvPolynomial.aeval (p x) f = MvPolynomial.aeval (p x') f) : ∃ b : MvPolynomial σ ℂ, (∃ x, MvPolynomial.aeval (p x) (MvPolynomial.aeval q b) ≠ 0) ∧ ∀ f ∈ s, ∃ b' : MvPolynomial σ ℂ, ∀ x, MvPolynomial.aeval (p x) (MvPolynomial.aeval q b) * MvPolynomial.aeval (p x) f = MvPolynomial.aeval (p x) (MvPolynomial.aeval q b')

theorem **exists_poly_blk_entries** {ι : Type} [Fintype ι] [DecidableEq ι] (e : Matrix ι ι ℚ) : ∃ q : ι × ι → MvPolynomial (ι × ι) ℂ, ∀ (x : Matrix ι ι ℂ) (ij : ι × ι), MvPolynomial.aeval (fun kl : ι × ι => x kl.1 kl.2) (q ij) = blk e x ij.1 ij.2

theorem **exists_multiplier_descend_blk** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), pts Γ ℂ ⊆ Γ → Γ.Nonempty → ∀ s : Finset (MvPolynomial (ι × ι) ℂ), (∀ f ∈ s, ∀ x ∈ Γ, ∀ x' ∈ Γ, blk e x = blk e x' → evC f x = evC f x') → ∃ b : MvPolynomial (ι × ι) ℂ, (∃ x ∈ Γ, evC b (blk e x) ≠ 0) ∧ ∀ f ∈ s, ∃ b' : MvPolynomial (ι × ι) ℂ, ∀ x ∈ Γ, evC b (blk e x) * evC f x = evC b' (blk e x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsMultiplierDescendBlockDiagPartOfRatIdempotStatement

theorem **_root_.GlobalTheory.T.exists_multiplier_descend_blockDiagPartOfRatIdempotent** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), pts Γ ℂ ⊆ Γ → Γ.Nonempty → ∀ s : Finset (MvPolynomial (ι × ι) ℂ), (∀ f ∈ s, ∀ x ∈ Γ, ∀ x' ∈ Γ, blk e x = blk e x' → evC f x = evC f x') → ∃ b : MvPolynomial (ι × ι) ℂ, (∃ x ∈ Γ, evC b (blk e x) ≠ 0) ∧ ∀ f ∈ s, ∃ b' : MvPolynomial (ι × ι) ℂ, ∀ x ∈ Γ, evC b (blk e x) * evC f x = evC b' (blk e x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNatSmulMem

theorem **exists_nat_mul_isIntegral** {E : Type} [Field E] [NumberField E] (a : E) : ∃ d : ℕ, 0 < d ∧ IsIntegral ℤ ((d : E) * a)

theorem **nat_smul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (RingOfIntegers E) (Fin n → E)) (d : ℕ) {x : Fin n → E} (hx : x ∈ L) : (d : E) • x ∈ L

theorem **exists_nat_smul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (RingOfIntegers E) (Fin n → E)) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (y : Fin n → E) : ∃ d : ℕ, 0 < d ∧ (d : E) • y ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNontrivialHeckeChar

theorem **exists_arch** : ∃ (θ : AE F E →+* ℂ) (τ : E →+* ℂ), Continuous θ ∧ (∀ x, θ (incA F E x) = τ x) ∧ ∀ α, θ (algebraMap (AF F) (AE F E) (pairA F 0 α)) = 0

theorem **exists_fin** : ∃ Φ : AE F E →+* FiniteAdeleRing (𝒪 E) E, Continuous Φ ∧ (∀ x, Φ (incA F E x) = algebraMap E _ x) ∧ ∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v (Φ y P) ≤ 1

theorem **cA_base** (σ : E →ₐ[F] E) (a : AF F) : cA σ (algebraMap (AF F) (AE F E) a) = algebraMap (AF F) (AE F E) a

theorem **cA_inc** (σ : E →ₐ[F] E) (x : E) : cA σ (incA F E x) = incA F E (σ x)

theorem **cA_cont** (σ : E →ₐ[F] E) : Continuous (cA σ)

theorem **cA_int** (σ : E →ₐ[F] E) (y : AE F E) (h : finPart F E y ∈ intAE F E) : finPart F E (cA σ y) ∈ intAE F E

theorem **idem_arch** {σ : E →ₐ[F] E} {θ : AE F E →+* ℂ} (hθ : ∀ α, θ (algebraMap (AF F) (AE F E) (pairA F 0 α)) = 0) {P : HeightOneSpectrum (𝒪 E)} {e f : AE F E} (h : IsPrimeIdem σ P e f) : θ e = 0 ∧ θ (cA σ e) = 0

theorem **idem_fin** {σ : E →ₐ[F] E} {Φ : AE F E →+* FiniteAdeleRing (𝒪 E) E} (hι : ∀ x, Φ (incA F E x) = algebraMap E _ x) (hint : ∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v (Φ y P) ≤ 1) {P : HeightOneSpectrum (𝒪 E)} {e f : AE F E} (h : IsPrimeIdem σ P e f) : Φ e P = 1 ∧ ∀ Q, Q ≠ P → Φ e Q = 0

theorem **val_alg** (P : HeightOneSpectrum (𝒪 E)) (x : E) : Valued.v (algebraMap E (FiniteAdeleRing (𝒪 E) E) x P) = P.valuation E x

theorem **unif_val** {σ : E →ₐ[F] E} {Φ : AE F E →+* FiniteAdeleRing (𝒪 E) E} (hι : ∀ x, Φ (incA F E x) = algebraMap E _ x) (hint : ∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v (Φ y P) ≤ 1) {P : HeightOneSpectrum (𝒪 E)} {π : 𝒪 E} (hπ : P.asIdeal = Ideal.span {π}) {x : (AE F E)ˣ} (h : IsUnifIdele σ P x) (Q : HeightOneSpectrum (𝒪 E)) : Valued.v (Φ (x : AE F E) Q) = Q.valuation E (π : E)

theorem **exists_char** (σ : E →ₐ[F] E) {k : ℕ} (hk : ∀ u : (𝒪 E)ˣ, u ^ k = 1) {θ : AE F E →+* ℂ} {τ : E →+* ℂ} (hθc : Continuous θ) (hθ : ∀ x, θ (incA F E x) = τ x) {Φ : AE F E →+* FiniteAdeleRing (𝒪 E) E} (hΦc : Continuous Φ) (hι : ∀ x, Φ (incA F E x) = algebraMap E _ x) (hint : ∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v (Φ y P) ≤ 1) : ∃ χ : (AE F E)ˣ →* Circle, IsHeckeChar F E χ ∧ (∀ a : (AF F)ˣ, χ (idF F E a) = 1) ∧ ∀ x : (AE F E)ˣ, finPart F E (x : AE F E) ∈ intAE F E → finPart F E ((x⁻¹ : (AE F E)ˣ) : AE F E) ∈ intAE F E → (χ x : ℂ) = (θ x / θ (cA σ x) / star (θ x / θ (cA σ x))) ^ k

theorem **exists_of_principal** (σ : E →ₐ[F] E) (hσ : IsCMExt F E σ) [IsPrincipalIdealRing (𝒪 E)] {k : ℕ} (hk : ∀ u : (𝒪 E)ˣ, u ^ k = 1) (a : 𝒪 E) (ha : σ (a : E) ^ (2 * k) ≠ (a : E) ^ (2 * k)) : ∃ χ : (AE F E)ˣ →* Circle, IsHeckeChar F E χ ∧ (∀ a : (AF F)ˣ, χ (idF F E a) = 1) ∧ χ ≠ 1 ∧ ∃ P : HeightOneSpectrum (𝒪 E), chiAt σ χ P ≠ 0 ∧ chiAt σ χ P ≠ 1

theorem **exists_nontrivial_heckeChar** : ∃ (F E : Type) (_ : Field F) (_ : NumberField F) (_ : Field E) (_ : NumberField E) (_ : Algebra F E) (σ : E →ₐ[F] E), IsCMExt F E σ ∧ ∃ χ : (AE F E)ˣ →* Circle, IsHeckeChar F E χ ∧ (∀ a : (AF F)ˣ, χ (idF F E a) = 1) ∧ χ ≠ 1 ∧ ∃ P : HeightOneSpectrum (𝒪 E), chiAt σ χ P ≠ 0 ∧ chiAt σ χ P ≠ 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNonzeroIsotropic

theorem **quadForm_totalDegree_le** {k : Type u_1} [Field k] (a : Fin 3 → Fin 3 → k) : (∑ i : Fin 3, ∑ j : Fin 3, C (a i j) * X i * X j : MvPolynomial (Fin 3) k).totalDegree ≤ 2

theorem **exists_nonzero_isotropic** : ∀ {k : Type u_1} [inst : Field k] [Finite k] (a : Fin (3 : ℕ) → Fin (3 : ℕ) → k), ∃ (x : Fin (3 : ℕ) → k), x ≠ (0 : Fin (3 : ℕ) → k) ∧ ∑ i : Fin (3 : ℕ), ∑ j : Fin (3 : ℕ), a i j * x i * x j = (0 : k)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNormOneReflectionCorrectionAtLevel

theorem **exists_nat_mul_integral** {E : Type} [Field E] [NumberField E] (a : E) : ∃ m : ℕ, 0 < m ∧ ∃ b : 𝒪 E, (b : E) = (m : E) * a

theorem **exists_nat_smul_mem_of_span** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (y : Fin n → E) : ∃ d : ℕ, 0 < d ∧ (d : E) • y ∈ L

theorem **exists_nat_smul_map_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hfg : L.FG) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (φ : (Fin n → E) →ₗ[E] (Fin n → E)) : ∃ d : ℕ, 0 < d ∧ ∀ v ∈ L, (d : E) • φ v ∈ L

theorem **exists_refl_correction** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (x : Fin n → E) (u : E) (N D : ℕ), hermE c G x x ≠ 0 → u * c u = 1 → 0 < N → 0 < D → ∃ u' : E, u' * c u' = 1 ∧ CongAt L N D (refl c G x (u' / u)) ∧ ∃ s : 𝒪 E, SplitOr c N s ∧ ∀ v ∈ L, s • (refl c G x u' *ᵥ v) ∈ L ∧ s • (refl c G x u'⁻¹ *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNormOneReflectionCorrectionAtLevelStatement

theorem **_root_.GlobalTheory.T.exists_norm_one_reflection_correction_at_level** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (x : Fin n → E) (u : E) (N D : ℕ), hermE c G x x ≠ 0 → u * c u = 1 → 0 < N → 0 < D → ∃ u' : E, u' * c u' = 1 ∧ CongAt L N D (refl c G x (u' / u)) ∧ ∃ s : 𝒪 E, SplitOr c N s ∧ ∀ v ∈ L, s • (refl c G x u' *ᵥ v) ∈ L ∧ s • (refl c G x u'⁻¹ *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNormalFormNonsquareAdicCompletion

theorem **exists_exp** {x : K} (hx : x ≠ 0) : ∃ n : ℤ, w x = WithZero.exp n

theorem **map_two_le** : w 2 ≤ 1

theorem **map_four_le** : w 4 ≤ 1

theorem **evenNorms_of_le_four** (h2 : (2 : K) ≠ 0) (hsq : ∀ x : K, w (x - 1) < w 4 → IsSquare x) (δ : K) (hδ : w δ = 1) (hns : ¬ IsSquare δ) (h : w (δ - 1) ≤ w 4) : EvenNorms w δ

theorem **not_isSquare_mul_sq** {δ c : K} (hc : c ≠ 0) (hns : ¬ IsSquare δ) : ¬ IsSquare (δ * c ^ 2)

theorem **unit_case** (h2 : (2 : K) ≠ 0) (π : K) (hπ : w π = WithZero.exp (-1)) (hsq : ∀ x : K, w (x - 1) < w 4 → IsSquare x) (hres : w 2 < 1 → ∀ x : K, w x ≤ 1 → ∃ y : K, w y ≤ 1 ∧ w (x - y ^ 2) < 1) (m : ℤ) (hm : w 4 = WithZero.exp m) (N : ℕ) : ∀ δ : K, w δ = 1 → ¬ IsSquare δ → w (δ - 1) ≤ WithZero.exp (m + N) → ∃ c : K, c ≠ 0 ∧ ((w (δ * c ^ 2) = 1 ∧ w 4 < w (δ * c ^ 2 - 1) ∧ ∃ n : ℤ, w (δ * c ^ 2 - 1) = WithZero.exp (2 * n + 1)) ∨ EvenNorms w (δ * c ^ 2))

theorem **general** (h2 : (2 : K) ≠ 0) (π : K) (hπ : w π = WithZero.exp (-1)) (hsq : ∀ x : K, w (x - 1) < w 4 → IsSquare x) (hres : w 2 < 1 → ∀ x : K, w x ≤ 1 → ∃ y : K, w y ≤ 1 ∧ w (x - y ^ 2) < 1) (δ : K) (hδ : δ ≠ 0) (hns : ¬ IsSquare δ) : ∃ c : K, c ≠ 0 ∧ ((∃ n : ℤ, w (δ * c ^ 2) = WithZero.exp (2 * n + 1)) ∨ (w (δ * c ^ 2) = 1 ∧ w 4 < w (δ * c ^ 2 - 1) ∧ ∃ n : ℤ, w (δ * c ^ 2 - 1) = WithZero.exp (2 * n + 1)) ∨ EvenNorms w (δ * c ^ 2))

private theorem **evenNorms_of_valuation_sub_one_le_four** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : v.adicCompletion F) (he : Valued.v e = 1) (hns : ¬ IsSquare e) (hle : Valued.v (e - 1) ≤ Valued.v (4 : v.adicCompletion F)) : _root_.GlobalTheory.QNorm.EvenNorms Valued.v e

private theorem **exists_normalForm_nonsquare_unit_lemma** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (π : v.adicCompletion F) (hπ : Valued.v π = WithZero.exp (-1)) (hsq : Valued.v (2 : v.adicCompletion F) < 1 → ∀ x : v.adicCompletion F, Valued.v x ≤ 1 → ∃ y : v.adicCompletion F, Valued.v y ≤ 1 ∧ Valued.v (x - y ^ 2) < 1) (a : ℤ) (ha : Valued.v (4 : v.adicCompletion F) = WithZero.exp a) (n : ℕ) : ∀ e : v.adicCompletion F, Valued.v e = 1 → ¬ IsSquare e → ∀ j : ℤ, Valued.v (e - 1) = WithZero.exp j → j ≤ a + n → ∃ c : v.adicCompletion F, c ≠ 0 ∧ ((Valued.v (e * c ^ 2) = 1 ∧ Valued.v (4 : v.adicCompletion F) < Valued.v (e * c ^ 2 - 1) ∧ ∃ m : ℤ, Valued.v (e * c ^ 2 - 1) = WithZero.exp (2 * m + 1)) ∨ _root_.GlobalTheory.QNorm.EvenNorms Valued.v (e * c ^ 2))

private theorem **exists_sq_near_of_valuation_two_lt_one** {K : Type} [Field K] (w : Valuation K ℤᵐ⁰) (R : Finset K) (hR : ∀ r ∈ R, w r ≤ 1) (hcov : ∀ x : K, w x ≤ 1 → ∃ r ∈ R, w (x - r) < 1) (h2 : w 2 < 1) (x : K) (hx : w x ≤ 1) : ∃ y : K, w y ≤ 1 ∧ w (x - y ^ 2) < 1

theorem **_root_.GlobalTheory.QNorm.exists_normal_form_of_not_isSquare** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (δ : v.adicCompletion F), δ ≠ 0 → ¬ IsSquare δ → ∃ c : v.adicCompletion F, c ≠ 0 ∧ ((∃ n : ℤ, Valued.v (δ * c ^ 2) = WithZero.exp (2 * n + 1)) ∨ (Valued.v (δ * c ^ 2) = 1 ∧ Valued.v (4 : v.adicCompletion F) < Valued.v (δ * c ^ 2 - 1) ∧ ∃ n : ℤ, Valued.v (δ * c ^ 2 - 1) = WithZero.exp (2 * n + 1)) ∨ _root_.GlobalTheory.QNorm.EvenNorms Valued.v (δ * c ^ 2))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNormaliseMeanSquareOfCocompact

theorem **exists_normalise_meanSquare_of_cocompact** : ∀ (G : Type) [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (Γ : AddSubgroup G) [DiscreteTopology Γ] (K : Set G), IsCompact K → (∀ x : G, ∃ g ∈ Γ, g + x ∈ K) → ∀ f : G → ℂ, Continuous f → f ≠ 0 → (∀ g ∈ Γ, ∀ w, ‖f (g + w)‖ = ‖f w‖) → (∃ (μ : MeasureTheory.Measure G) (ℱ : Set G), μ.IsAddHaarMeasure ∧ MeasureTheory.IsAddFundamentalDomain Γ ℱ μ ∧ μ ℱ = 1) ∧ ∃ c : ℂ, ∀ (μ : MeasureTheory.Measure G) (ℱ : Set G), μ.IsAddHaarMeasure → MeasureTheory.IsAddFundamentalDomain Γ ℱ μ → μ ℱ = 1 → ∫ w in ℱ, ‖c * f w‖ ^ 2 ∂μ = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNormaliseMeanSquareOfCocompactStatement

theorem **_root_.GlobalTheory.T.exists_normalise_meanSquare_of_cocompact'** : ∀ (G : Type) [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (Γ : AddSubgroup G) [DiscreteTopology Γ] (K : Set G), IsCompact K → (∀ x : G, ∃ g ∈ Γ, g + x ∈ K) → ∀ f : G → ℂ, Continuous f → f ≠ 0 → (∀ g ∈ Γ, ∀ w, ‖f (g + w)‖ = ‖f w‖) → (∃ (μ : MeasureTheory.Measure G) (ℱ : Set G), μ.IsAddHaarMeasure ∧ MeasureTheory.IsAddFundamentalDomain Γ ℱ μ ∧ μ ℱ = 1) ∧ ∃ c : ℂ, ∀ (μ : MeasureTheory.Measure G) (ℱ : Set G), μ.IsAddHaarMeasure → MeasureTheory.IsAddFundamentalDomain Γ ℱ μ → μ ℱ = 1 → ∫ w in ℱ, ‖c * f w‖ ^ 2 ∂μ = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNsmulMem

theorem **exists_nat_mul_integral** {E : Type} [Field E] [NumberField E] (a : E) : ∃ (e : ℕ) (b : RingOfIntegers E), 0 < e ∧ (e : E) * a = algebraMap (RingOfIntegers E) E b

theorem **exists_nsmul_mem** {E : Type} [Field E] {n : ℕ} [NumberField E] (L : Submodule (RingOfIntegers E) (Fin n → E)) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (w : Fin n → E) : ∃ d : ℕ, 0 < d ∧ d • w ∈ L

theorem **exists_nsmul_mulVec_mem** {E : Type} [Field E] {n : ℕ} [NumberField E] (L : Submodule (RingOfIntegers E) (Fin n → E)) (hfg : L.FG) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (M : Matrix (Fin n) (Fin n) E) : ∃ d : ℕ, 0 < d ∧ ∀ v ∈ L, d • M.mulVec v ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsNsmulMulVecMem

theorem **exists_nat_mul_integral** {E : Type} [Field E] [NumberField E] (c : E) : ∃ K : ℕ, 0 < K ∧ IsIntegral ℤ ((K : E) * c)

theorem **exists_nsmul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (RingOfIntegers E) (Fin n → E)) (hL : Submodule.span E (↑L : Set (Fin n → E)) = ⊤) (w : Fin n → E) : ∃ N : ℕ, 0 < N ∧ N • w ∈ L

theorem **exists_nsmul_mulVec_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (RingOfIntegers E) (Fin n → E)) (hfg : L.FG) (hL : Submodule.span E (↑L : Set (Fin n → E)) = ⊤) (M : Matrix (Fin n) (Fin n) E) : ∃ N : ℕ, 0 < N ∧ ∀ v ∈ L, N • Matrix.mulVec M v ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlaceIsSquareAndNotIsSquare

theorem **exists_place_isSquare_and_not_isSquare** : ∀ (F : Type) [Field F] [NumberField F] (δ u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), δ ≠ 0 → u ≠ 0 → ¬ IsSquare δ → ¬ IsSquare u → ¬ IsSquare (u * δ) → ∃ v, v ∉ S₀ ∧ IsSquare (algebraMap F (v.adicCompletion F) δ) ∧ ¬ IsSquare (algebraMap F (v.adicCompletion F) u)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlaceNotIsSquareAdicCompletion

theorem **normSub_eq_top_of_isSquare_image** {F L : Type} [Field F] [Field L] [CharZero L] (φ : F →+* L) (δ : F) (hδ : δ ≠ 0) (h : IsSquare (φ δ)) (x : Fˣ) : x ∈ normSub φ δ

theorem **exists_not_mem_not_isSquare_completion** : ∀ (F : Type) [Field F] [NumberField F] (u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), u ≠ 0 → ¬ IsSquare u → ∃ v, v ∉ S₀ ∧ ¬ IsSquare (algebraMap F (v.adicCompletion F) u)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlaceNotIsSquareAdicCompletionStatement

theorem **_root_.GlobalTheory.T.exists_place_not_isSquare_adicCompletion** : ∀ (F : Type) [Field F] [NumberField F] (u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), u ≠ 0 → ¬ IsSquare u → ∃ v, v ∉ S₀ ∧ ¬ IsSquare (algebraMap F (v.adicCompletion F) u)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlaceNotPowAdicCompletion

theorem **exists_not_mem_not_pow_completion** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → u ≠ 0 → (∀ r : F, r ^ p ≠ u) → ∃ v, v ∉ S₀ ∧ ∀ r : v.adicCompletion F, r ^ p ≠ algebraMap F (v.adicCompletion F) u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlaceNotPowAdicCompletionStatement

theorem **_root_.GlobalTheory.T.exists_place_not_pow_adicCompletion** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → u ≠ 0 → (∀ r : F, r ^ p ≠ u) → ∃ v, v ∉ S₀ ∧ ∀ r : v.adicCompletion F, r ^ p ≠ algebraMap F (v.adicCompletion F) u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlacePowAndNotPow

theorem **exists_place_pow_and_not_pow** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ δ u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → δ ≠ 0 → u ≠ 0 → (∀ r : F, r ^ p ≠ δ) → (∀ (k : ℕ) (r : F), u ≠ δ ^ k * r ^ p) → ∃ v, v ∉ S₀ ∧ (∃ r : v.adicCompletion F, r ^ p = algebraMap F (v.adicCompletion F) δ) ∧ ∀ r : v.adicCompletion F, r ^ p ≠ algebraMap F (v.adicCompletion F) u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlacePowAndNotPowStatement

theorem **_root_.GlobalTheory.T.exists_place_pow_and_not_pow'** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ δ u : F) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → δ ≠ 0 → u ≠ 0 → (∀ r : F, r ^ p ≠ δ) → (∀ (k : ℕ) (r : F), u ≠ δ ^ k * r ^ p) → ∃ v, v ∉ S₀ ∧ (∃ r : v.adicCompletion F, r ^ p = algebraMap F (v.adicCompletion F) δ) ∧ ∀ r : v.adicCompletion F, r ^ p ≠ algebraMap F (v.adicCompletion F) u

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPlaceValuationAut

theorem **exists_place_valuation_aut** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E ≃ₐ[F] E) (w : HeightOneSpectrum (𝒪 E)) : ∃ w' : HeightOneSpectrum (𝒪 E), w'.under (𝒪 F) = w.under (𝒪 F) ∧ ∀ y : E, w'.valuation E y = w.valuation E (σ y)

theorem **sUnits_over_and_aut** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E] (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → (∀ x : Fˣ, Units.map (algebraMap F E : F →* E) x ∈ sUnits E SE ↔ x ∈ sUnits F S) ∧ ∀ (τ : E ≃ₐ[F] E) (y : Eˣ), y ∈ sUnits E SE → Units.map (τ : E →* E) y ∈ sUnits E SE

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPolynomialSectionBlockDiagonalMap

lemma **corner_mul** (hE : E * E = E) {x y : A} (hx : E * x * (1 - E) = x) (hy : E * y * (1 - E) = y) : x * y = 0

lemma **diag_comm** (hE : E * E = E) (g : A) : E * (E * g * E + (1 - E) * g * (1 - E)) = (E * g * E + (1 - E) * g * (1 - E)) * E

lemma **corner_mul_comm** {x D : A} (hD : E * D = D * E) (hx : E * x * (1 - E) = x) : E * (x * D) * (1 - E) = x * D

lemma **corner_idem** (hE : E * E = E) (g : A) : E * (E * g * (1 - E)) * (1 - E) = E * g * (1 - E)

lemma **corner_diag** (hE : E * E = E) {x : A} (hx : E * x * (1 - E) = x) : E * x * E + (1 - E) * x * (1 - E) = 0

lemma **inv_comm** {D D' : A} (hD : E * D = D * E) (h1 : D * D' = 1) (h2 : D' * D = 1) : E * D' = D' * E

lemma **map_rat** (e : Matrix ι ι ℚ) : φ.mapMatrix (e.map (algebraMap ℚ R)) = e.map (algebraMap ℚ S)

lemma **map_blk** (e : Matrix ι ι ℚ) (g : Matrix ι ι R) : φ.mapMatrix (blk e g) = blk e (φ.mapMatrix g)

lemma **map_linExt** (c : ι × ι → ι × ι → ℚ) (g : Matrix ι ι R) : φ.mapMatrix (linExt c g) = linExt c (φ.mapMatrix g)

lemma **map_sec** (e : Matrix ι ι ℚ) (c : ι × ι → ι × ι → ℚ) (g : Matrix ι ι R) : φ.mapMatrix (sec e c g) = sec e c (φ.mapMatrix g)

lemma **exists_poly** (e : Matrix ι ι ℚ) (c : ι × ι → ι × ι → ℚ) : ∃ f : ι → ι → MvPolynomial (ι × ι) ℚ, ∀ (g : Matrix ι ι ℂ) i j, ev (f i j) g = sec e c g i j

lemma **linExt_sub** (c : ι × ι → ι × ι → ℚ) (x y : Matrix ι ι ℂ) : linExt c (x - y) = linExt c x - linExt c y

lemma **linExt_zero** (c : ι × ι → ι × ι → ℚ) : linExt c (0 : Matrix ι ι ℂ) = 0

theorem **GlobalTheory.ClassFin.exists_blk_section** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∃ (c : ι × ι → ι × ι → ℚ) (b : ι → ι → MvPolynomial (ι × ι) ℚ), (∀ X : Matrix ι ι ℂ, 1 + linExt c X ∈ Γ ∧ e.map (algebraMap ℚ ℂ) * linExt c X * (1 - e.map (algebraMap ℚ ℂ)) = linExt c X) ∧ (Matrix.of fun i j => ev (b i j) (1 : Matrix ι ι ℂ)) = 1 ∧ ∀ g ∈ Γ, (Matrix.of fun i j => ev (b i j) (blk e g)) ∈ Γ ∧ blk e (Matrix.of fun i j => ev (b i j) (blk e g)) = blk e g ∧ g = (1 + linExt c (e.map (algebraMap ℚ ℂ) * g * (1 - e.map (algebraMap ℚ ℂ)) * (blk e g).adjugate)) * (Matrix.of fun i j => ev (b i j) (blk e g))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPolynomialSectionBlockDiagonalMapStatement

theorem **_root_.GlobalTheory.T.exists_polynomial_section_blockDiagonal_map** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∃ (c : ι × ι → ι × ι → ℚ) (b : ι → ι → MvPolynomial (ι × ι) ℚ), (∀ X : Matrix ι ι ℂ, 1 + linExt c X ∈ Γ ∧ e.map (algebraMap ℚ ℂ) * linExt c X * (1 - e.map (algebraMap ℚ ℂ)) = linExt c X) ∧ (Matrix.of fun i j => ev (b i j) (1 : Matrix ι ι ℂ)) = 1 ∧ ∀ g ∈ Γ, (Matrix.of fun i j => ev (b i j) (blk e g)) ∈ Γ ∧ blk e (Matrix.of fun i j => ev (b i j) (blk e g)) = blk e g ∧ g = (1 + linExt c (e.map (algebraMap ℚ ℂ) * g * (1 - e.map (algebraMap ℚ ℂ)) * (blk e g).adjugate)) * (Matrix.of fun i j => ev (b i j) (blk e g))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPolynomialSectionBlockDiagonalRational

theorem **evHom_generic** {R : Type} [CommRing R] [Algebra ℚ R] (y : Matrix ι ι R) : (evHom y).mapMatrix (Matrix.mvPolynomialX ι ι ℚ) = y

theorem **map_ev** {R S : Type} [CommRing R] [Algebra ℚ R] [CommRing S] [Algebra ℚ S] (φ : R →ₐ[ℚ] S) (f : MvPolynomial (ι × ι) ℚ) (x : Matrix ι ι R) : φ (ev f x) = ev f (φ.mapMatrix x)

theorem **map_rat** {R S : Type} [CommRing R] [Algebra ℚ R] [CommRing S] [Algebra ℚ S] (φ : R →ₐ[ℚ] S) (e : Matrix ι ι ℚ) : φ.mapMatrix (e.map (algebraMap ℚ R)) = e.map (algebraMap ℚ S)

theorem **map_blk** {R S : Type} [CommRing R] [Algebra ℚ R] [CommRing S] [Algebra ℚ S] (φ : R →ₐ[ℚ] S) (e : Matrix ι ι ℚ) (x : Matrix ι ι R) : φ.mapMatrix (blk e x) = blk e (φ.mapMatrix x)

theorem **map_linExt** {R S : Type} [CommRing R] [Algebra ℚ R] [CommRing S] [Algebra ℚ S] (φ : R →ₐ[ℚ] S) (c : ι × ι → ι × ι → ℚ) (x : Matrix ι ι R) : φ.mapMatrix (linExt c x) = linExt c (φ.mapMatrix x)

theorem **map_poly** {R S : Type} [CommRing R] [Algebra ℚ R] [CommRing S] [Algebra ℚ S] (φ : R →ₐ[ℚ] S) (b : ι → ι → MvPolynomial (ι × ι) ℚ) (x : Matrix ι ι R) : φ.mapMatrix (Matrix.of fun i j => ev (b i j) x) = Matrix.of fun i j => ev (b i j) (φ.mapMatrix x)

theorem **transfer_mem** {F : ∀ (R : Type) [CommRing R] [Algebra ℚ R], Matrix ι ι R → Matrix ι ι R} (hF : IsNatural F) {Γ Γ' : Set (Matrix ι ι ℂ)} (h : ∀ g ∈ Γ, F ℂ g ∈ Γ') {R : Type} [CommRing R] [Algebra ℚ R] {y : Matrix ι ι R} (hy : y ∈ pts Γ R) : F R y ∈ pts Γ' R

theorem **transfer_eq** {F G : ∀ (R : Type) [CommRing R] [Algebra ℚ R], Matrix ι ι R → Matrix ι ι R} (hF : IsNatural F) (hG : IsNatural G) {Γ : Set (Matrix ι ι ℂ)} (h : ∀ g ∈ Γ, F ℂ g = G ℂ g) {R : Type} [CommRing R] [Algebra ℚ R] {y : Matrix ι ι R} (hy : y ∈ pts Γ R) : F R y = G R y

theorem **mem_pts_univ** {R : Type} [CommRing R] [Algebra ℚ R] (y : Matrix ι ι R) : y ∈ pts (Set.univ : Set (Matrix ι ι ℂ)) R

theorem **exists_section** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → ∃ (c : ι × ι → ι × ι → ℚ) (b : ι → ι → MvPolynomial (ι × ι) ℚ), ∀ (R : Type) [CommRing R] [Algebra ℚ R], (∀ X : Matrix ι ι R, 1 + linExt c X ∈ pts Γ R) ∧ (∀ X Y : Matrix ι ι R, linExt c X * linExt c Y = 0) ∧ (∀ y ∈ pts Γ R, blk e y = 1 → ∃ X : Matrix ι ι R, y = 1 + linExt c X) ∧ (∀ x ∈ pts Γ R, (1 - e.map (algebraMap ℚ R)) * x * e.map (algebraMap ℚ R) = 0 ∧ blk e x ∈ pts (blk e '' Γ) R) ∧ ∀ y ∈ pts (blk e '' Γ) R, (Matrix.of fun i j => ev (b i j) (y)) ∈ pts Γ R ∧ blk e (Matrix.of fun i j => ev (b i j) (y)) = y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPosMulNormSqLe

theorem **exists_pos_mul_norm_sq_le** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (om : V →ₗ[ℝ] V →ₗ[ℝ] ℝ) (J : V →ₗ[ℝ] V) (h : ∀ v : V, v ≠ 0 → 0 < om v (J v)) : ∃ c : ℝ, 0 < c ∧ ∀ v : V, c * ‖v‖ ^ 2 ≤ om v (J v)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPosNatMulMemRingOfIntegers

theorem **exists_pos_nat_mul_mem_ringOfIntegers** {E : Type} [Field E] [NumberField E] (a : E) : ∃ m : ℕ, 0 < m ∧ ∃ b : NumberField.RingOfIntegers E, (b : E) = (m : E) * a

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPrimeIdempotentCMField

theorem **exists_indicator** (K : Type) [Field K] [NumberField K] (v : HeightOneSpectrum (𝒪 K)) : ∃ a : FiniteAdeleRing (𝒪 K) K, a v = 1 ∧ ∀ w, w ≠ v → a w = 0

theorem **cA_algebraMap** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) (ε : AF F) : cA σ (algebraMap (AF F) (AE F E) ε) = algebraMap (AF F) (AE F E) ε

theorem **cA_incA** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) (x : E) : cA σ (incA F E x) = incA F E (σ x)

theorem **prod_ext** {K : Type} [Field K] [NumberField K] {M : Type} {a b : M × FiniteAdeleRing (𝒪 K) K} (h1 : a.1 = b.1) (h2 : ∀ P, a.2 P = b.2 P) : a = b

theorem **exists_isPrimeIdem** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ P : HeightOneSpectrum (𝒪 E), ∃ e f, IsPrimeIdem σ P e f

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPrimeIdempotentCMFieldStatement

theorem **_root_.GlobalTheory.T.exists_primeIdempotent_CMField** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ P : HeightOneSpectrum (𝒪 E), ∃ e f, IsPrimeIdem σ P e f

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPrimeInvolutionOfRingInvolution

theorem **exists_primeInvolution_of_ringInvolution** {R : Type} [CommRing R] [IsDedekindDomain R] [Module.Free ℤ R] [Module.Finite ℤ R] (s : R ≃+* R) (hs : ∀ x, s (s x) = x) : ∃ c : HeightOneSpectrum R ≃ HeightOneSpectrum R, (∀ P, c (c P) = P) ∧ (∀ P, Ideal.absNorm (c P).asIdeal = Ideal.absNorm P.asIdeal) ∧ (∀ P x, s x ∈ (c P).asIdeal ↔ x ∈ P.asIdeal) ∧ (∀ P x, s x ∈ (c P).asIdeal ^ 2 ↔ x ∈ P.asIdeal ^ 2)

theorem **cmConjugation_on_primes** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsTotallyReal F → IsTotallyComplex E → Module.finrank F E = 2 → (∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) → (∀ x, σ (σ x) = x) ∧ ∃ c : HeightOneSpectrum (𝒪 E) ≃ HeightOneSpectrum (𝒪 E), (∀ P, c (c P) = P) ∧ (∀ P, Ideal.absNorm (c P).asIdeal = Ideal.absNorm

P.asIdeal) ∧ (∀ x : 𝒪 E, ∃ y : 𝒪 E, (y : E) = σ x) ∧ (∀ P (x y : 𝒪 E), (y : E) = σ x → (y ∈ (c P).asIdeal ↔ x ∈ P.asIdeal) ∧ (y ∈ (c P).asIdeal ^ 2 ↔ x ∈ P.asIdeal ^ 2)) ∧ ∀ P Q : HeightOneSpectrum (𝒪 E), P.under (𝒪 F) = Q.under (𝒪 F) ↔ Q = P ∨ Q = c P

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsPushoutPrimitiveRoot

theorem **isField_tensor_of_coprime** (F A B : Type) [Field F] [Field A] [Field B] [Algebra F A] [Algebra F B] (h : Nat.Coprime (Module.finrank F A) (Module.finrank F B)) : IsField (A ⊗[F] B)

theorem **coprime_finrank_cyclotomic** (p : ℕ) (hp : p.Prime) (F A : Type) [Field F] [Field A] [Algebra F A] [IsCyclotomicExtension {p} F A] : Nat.Coprime (Module.finrank F A) p

theorem **exists_pushout_primitiveRoot** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E], p.Prime → p ≠ 2 → Module.finrank F E = p → ∃ (F' E' : Type) (_ : Field F') (_ : NumberField F') (_ : Field E') (_ : NumberField E') (_ : Algebra F F') (_ : Algebra F' E') (_ : Algebra E E') (_ : Algebra F E') (_ : IsScalarTower F F' E') (_ : IsScalarTower F E E') (ζ : F'), Algebra.IsPushout F F' E E' ∧ IsTotallyComplex F' ∧ IsGalois F' E' ∧ IsPrimitiveRoot ζ p ∧ Module.finrank F' E' = p ∧ Nat.Coprime (Module.finrank F F') p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsQuasiReflectionProdMulVecEq

theorem **herm_sub_smul_left** (z x w : Fin n → E) (e : E) : hermE c G (z - e • x) w = hermE c G z w - e * hermE c G x w

theorem **herm_sub_smul_right** (z x w : Fin n → E) (e : E) : hermE c G w (z - e • x) = hermE c G w z - c e * hermE c G w x

theorem **herm_smul_left** (x w : Fin n → E) (e : E) : hermE c G (e • x) w = e * hermE c G x w

theorem **herm_conj** (h : IsHermForm c G) (x y : Fin n → E) : c (hermE c G x y) = hermE c G y x

theorem **refl_mulVec** (x v : Fin n → E) (u : E) : refl c G x u *ᵥ v = v + ((u - 1) * hermE c G v x / hermE c G x x) • x

theorem **exists_refl** (h : IsHermForm c G) (x z : Fin n → E) (hz : hermE c G z z = hermE c G x x) (e : E) (he : c e = e) (he2 : e * e = 1) (hy : hermE c G (z - e • x) (z - e • x) ≠ 0) : ∃ u : E, u * c u = 1 ∧ refl c G (z - e • x) u *ᵥ z = e • x

theorem **GlobalTheory.UnRep.exists_refl_prod_mulVec_eq** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ x z : Fin n → E, hermE c G x x ≠ 0 → hermE c G z z = hermE c G x x → ∃ l : List ((Fin n → E) × E), (∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1 ∧ p.1 ∈ Submodule.span E {x, z}) ∧ (l.map fun p => refl c G p.1 p.2).prod *ᵥ z = x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsQuasiReflectionProdMulVecEqStatement

theorem **_root_.GlobalTheory.T.exists_quasiReflection_prod_mulVec_eq** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ x z : Fin n → E, hermE c G x x ≠ 0 → hermE c G z z = hermE c G x x → ∃ l : List ((Fin n → E) × E), (∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1 ∧ p.1 ∈ Submodule.span E {x, z}) ∧ (l.map fun p => refl c G p.1 p.2).prod *ᵥ z = x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsQuaternaryOfCoordinates

theorem **exists_norm_of_quaternary** [CharZero k] {α β γ s : k} (hα : α ≠ 0) (hβ : β ≠ 0) (hγ : γ ≠ 0) (hs : s ^ 2 = α * β * γ) {x y z t : k} (h0 : x ≠ 0 ∨ y ≠ 0 ∨ z ≠ 0 ∨ t ≠ 0) (h : x ^ 2 - α * y ^ 2 - β * z ^ 2 + γ * t ^ 2 = 0) : ∃ X Y : k, β = X ^ 2 - α * Y ^ 2

theorem **exists_quaternary_of_coordinates** {α β γ : k} (hα : α ≠ 0) (hβ : β ≠ 0) {X₀ X₁ Y₀ Y₁ : k} (h1 : β = X₀ ^ 2 + α * β * γ * X₁ ^ 2 - α * (Y₀ ^ 2 + α * β * γ * Y₁ ^ 2)) (h2 : X₀ * X₁ = α * (Y₀ * Y₁)) : ∃ x y z t : k, (x ≠ 0 ∨ y ≠ 0 ∨ z ≠ 0 ∨ t ≠ 0) ∧ x ^ 2 - α * y ^ 2 - β * z ^ 2 + γ * t ^ 2 = 0

theorem **isotropic_four_iff** {a : Fin 4 → k} (ha : a 0 ≠ 0) : (∃ x : Fin 4 → k, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0) ↔ ∃ x y z t : k, (x ≠ 0 ∨ y ≠ 0 ∨ z ≠ 0 ∨ t ≠ 0) ∧ x ^ 2 - (-(a 0 * a 1)) * y ^ 2 - (-(a 0 * a 2)) * z ^ 2 + (a 0 * a 3) * t ^ 2 = 0

theorem **isotropic_succ_iff** {n : ℕ} {a : Fin (n + 1) → k} : (∃ x : Fin (n + 1) → k, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0) ↔ ∃ (t : k) (z : Fin n → k), (t ≠ 0 ∨ z ≠ 0) ∧ a 0 * t ^ 2 + ∑ i, a i.succ * z i ^ 2 = 0

theorem **exists_value_isotropic** [CharZero k] {n : ℕ} {a a' : k} (ha : a ≠ 0) (ha' : a' ≠ 0) {b : Fin (n + 1) → k} (hb : b 0 ≠ 0) {x y : k} {z : Fin (n + 1) → k} (h0 : x ≠ 0 ∨ y ≠ 0 ∨ z ≠ 0) (h : a * x ^ 2 + (a' * y ^ 2 + ∑ i, b i * z i ^ 2) = 0) : ∃ x y : k, a * x ^ 2 + a' * y ^ 2 ≠ 0 ∧ ∀ μ : k, IsSquare (μ / (a * x ^ 2 + a' * y ^ 2)) → ∃ (t : k) (z : Fin (n + 1) → k), (t ≠ 0 ∨ z ≠ 0) ∧ μ * t ^ 2 + ∑ i, b i * z i ^ 2 = 0

theorem **isotropic_of_injective** {ι κ : Type*} [Fintype ι] [Fintype κ] (e : κ → ι) (he : Function.Injective e) {a : ι → k} (h : ∃ x : κ → k, x ≠ 0 ∧ ∑ j, a (e j) * x j ^ 2 = 0) : ∃ x : ι → k, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

theorem **isotropic_real** {ι : Type*} [Fintype ι] {a : ι → ℝ} {i j : ι} (hi : a i < 0) (hj : 0 < a j) : ∃ x : ι → ℝ, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatBilinForm

theorem **map_rat_smul_real** {U : Type} [AddCommGroup U] [Module ℝ U] {V : Type} [AddCommGroup V] [Module ℚ V] (p : V →+ U) (c : ℚ) (x : V) : p (c • x) = (c : ℝ) • p x

theorem **exists_rat_value** {U : Type} [AddCommGroup U] [Module ℝ U] (om : U →ₗ[ℝ] U →ₗ[ℝ] ℝ) {V : Type} [AddCommGroup V] (p : V →+ U) (L : AddSubgroup V) (hL : ∀ x : V, ∃ N : ℕ, 0 < N ∧ N • x ∈ L) (hint : ∀ x ∈ L, ∀ y ∈ L, ∃ k : ℤ, om (p x) (p y) = (k : ℝ)) (u v : V) : ∃ q : ℚ, (q : ℝ) = -(om (p u) (p v))

theorem **exists_rat_bilinForm** {U : Type} [AddCommGroup U] [Module ℝ U] (om : U →ₗ[ℝ] U →ₗ[ℝ] ℝ) {V : Type} [AddCommGroup V] [Module ℚ V] (p : V →+ U) (L : AddSubgroup V) (hL : ∀ x : V, ∃ N : ℕ, 0 < N ∧ N • x ∈ L) (hint : ∀ x ∈ L, ∀ y ∈ L, ∃ k : ℤ, om (p x) (p y) = (k : ℝ)) : ∃ B : LinearMap.BilinForm ℚ V, ∀ u v : V, ((B u v : ℚ) : ℝ) = -(om (p u) (p v))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatIdempotentBlockDiagPartKillsSpan

theorem **exists_rat_idempotent_blk_eq_zero** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ¬ SS Γ → ∃ e : Matrix ι ι ℚ, e * e = e ∧ (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) ∧ ∃ a ∈ Submodule.span ℂ Γ, a ≠ 0 ∧ blk e a = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatIdempotentBlockDiagPartKillsSpanStatement

theorem **_root_.GlobalTheory.T.exists_rat_idempotent_blockDiagPart_kills_span** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ¬ SS Γ → ∃ e : Matrix ι ι ℚ, e * e = e ∧ (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) ∧ ∃ a ∈ Submodule.span ℂ Γ, a ≠ 0 ∧ blk e a = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatIdempotentOfAutStableSubspace

theorem **exists_rat_idempotent_of_stable** (ι : Type) [Fintype ι] [DecidableEq ι] (W : Submodule ℂ (ι → ℂ)) (hW : ∀ σ : ℂ ≃+* ℂ, ∀ v ∈ W, (fun i => σ (v i)) ∈ W) : ∃ e : Matrix ι ι ℚ, e * e = e ∧ ∀ v, v ∈ W ↔ e.map (algebraMap ℚ ℂ) *ᵥ v = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatIdempotentOfAutStableSubspaceStatement

theorem **_root_.GlobalTheory.T.exists_rat_idempotent_of_autStable_subspace** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (W : Submodule ℂ (ι → ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ v ∈ W, (fun i => σ (v i)) ∈ W) → ∃ e : Matrix ι ι ℚ, e * e = e ∧ ∀ v, v ∈ W ↔ e.map (algebraMap ℚ ℂ) *ᵥ v = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatLinearParamBlockDiagKernel

theorem **corner_mul_corner** {A : Type} [Ring A] (E x y : A) (hE : E * E = E) (hx : E * x * (1 - E) = x) (hy : E * y * (1 - E) = y) : x * y = 0

theorem **exists_linExt_corner_proj** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → ∃ c : ι × ι → ι × ι → ℚ, (∀ X : Matrix ι ι ℂ, 1 + linExt c X ∈ Γ ∧ e.map (algebraMap ℚ ℂ) * linExt c X * (1 - e.map (algebraMap ℚ ℂ)) = linExt c X) ∧ ∀ ε : Matrix ι ι ℂ, 1 + ε ∈ Γ → e.map (algebraMap ℚ ℂ) * ε * (1 - e.map (algebraMap ℚ ℂ)) = ε → linExt c ε = ε

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatLinearParamBlockDiagKernelStatement

theorem **_root_.GlobalTheory.T.exists_rat_linear_param_blockDiag_kernel** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsQGroup Γ → e * e = e → ∃ c : ι × ι → ι × ι → ℚ, (∀ X : Matrix ι ι ℂ, 1 + linExt c X ∈ Γ ∧ e.map (algebraMap ℚ ℂ) * linExt c X * (1 - e.map (algebraMap ℚ ℂ)) = linExt c X) ∧ ∀ ε : Matrix ι ι ℂ, 1 + ε ∈ Γ → e.map (algebraMap ℚ ℂ) * ε * (1 - e.map (algebraMap ℚ ℂ)) = ε → linExt c ε = ε

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatOfForallComplexRingEquivFixed

theorem **_root_.GlobalTheory.ClassFin.exists_rat_of_forall_ringEquiv_fixed** : ∀ z : ℂ, (∀ σ : ℂ ≃+* ℂ, σ z = z) → ∃ q : ℚ, z = q

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatOfForallComplexRingEquivFixedStatement

theorem **_root_.GlobalTheory.T.exists_rat_of_forall_complex_ringEquiv_fixed** : ∀ z : ℂ, (∀ σ : ℂ ≃+* ℂ, σ z = z) → ∃ q : ℚ, z = q

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatPolyEvalEqTensorPowerSumRightAction

theorem **exists_mvPolynomial_ev_eq_act** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), ∃ Φ : TIdx ι d → MvPolynomial (ι × ι) ℚ, ∀ (R : Type) [CommRing R] [Algebra ℚ R] (y : Matrix ι ι R) (k : TIdx ι d), ev (Φ k) y = act d (fun k => algebraMap ℚ R (t k)) y k

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatPolyEvalEqTensorPowerSumRightActionStatement

theorem **_root_.GlobalTheory.T.exists_ratPoly_eval_eq_tensorPowerSumRightAction** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), ∃ Φ : TIdx ι d → MvPolynomial (ι × ι) ℚ, ∀ (R : Type) [CommRing R] [Algebra ℚ R] (y : Matrix ι ι R) (k : TIdx ι d), ev (Φ k) y = act d (fun k => algebraMap ℚ R (t k)) y k

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatProjectorOfAutStableMatrices

theorem **repr_entry** (w : Matrix ι ι ℂ) (k : ι × ι) : (Module.Basis.ofEquivFun (entryEquiv ι)).repr w k = w k.1 k.2

theorem **linExt_eq_sum** (p : Matrix ι ι ℚ →ₗ[ℚ] Matrix ι ι ℚ) (X : Matrix ι ι ℂ) : linExt (fun ij kl => p (single kl.1 kl.2 1) ij.1 ij.2) X = ∑ kl : ι × ι, X kl.1 kl.2 • castLin ι (p (single kl.1 kl.2 1))

theorem **linExt_cast** (p : Matrix ι ι ℚ →ₗ[ℚ] Matrix ι ι ℚ) (Y : Matrix ι ι ℚ) : linExt (fun ij kl => p (single kl.1 kl.2 1) ij.1 ij.2) (castLin ι Y) = castLin ι (p Y)

theorem **exists_rat_projector** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (E : Submodule ℂ (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ x ∈ E, x.map σ ∈ E) → ∃ c : ι × ι → ι × ι → ℚ, (∀ X : Matrix ι ι ℂ, linExt c X ∈ E) ∧ ∀ x ∈ E, linExt c x = x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRatProjectorOfAutStableMatricesStatement

theorem **_root_.GlobalTheory.T.exists_rat_projector_of_autStable_matrices** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (E : Submodule ℂ (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ x ∈ E, x.map σ ∈ E) → ∃ c : ι × ι → ι × ι → ℚ, (∀ X : Matrix ι ι ℂ, linExt c X ∈ E) ∧ ∀ x ∈ E, linExt c x = x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalCoordPolynomialBaseChange

theorem **repr_algebraMap_of_basis_eq** {F β : Type} [Field F] [Algebra ℚ F] [Fintype β] (ω : Module.Basis β ℚ F) {R A : Type} [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A) (hb : ∀ i, b i = algebraMap F A (ω i)) (a : F) (k : β) : b.repr (algebraMap F A a) k = algebraMap ℚ R (ω.repr a k)

private theorem **basis_repr_mul_of_basis_eq_algebraMap** {F β : Type} [Field F] [Algebra ℚ F] [Fintype β] (ω : Module.Basis β ℚ F) {R A : Type} [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A) (hb : ∀ i, b i = algebraMap F A (ω i)) (u v : A) (k : β) : b.repr (u * v) k = ∑ i, ∑ j, b.repr u i * b.repr v j * algebraMap ℚ R (ω.repr (ω i * ω j) k)

theorem **exists_coord_polynomial** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (τ : Type) (P : MvPolynomial τ F), ∃ Q : β → MvPolynomial (τ × β) ℚ, ∀ (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ (x : τ → A) (k : β), b.repr (MvPolynomial.aeval x P) k = MvPolynomial.aeval (fun p : τ × β => b.repr (x p.1) p.2) (Q k)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalCoordPolynomialBaseChangeStatement

theorem **_root_.GlobalTheory.T.exists_rational_coord_polynomial_baseChange** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (τ : Type) (P : MvPolynomial τ F), ∃ Q : β → MvPolynomial (τ × β) ℚ, ∀ (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ (x : τ → A) (k : β), b.repr (MvPolynomial.aeval x P) k = MvPolynomial.aeval (fun p : τ × β => b.repr (x p.1) p.2) (Q k)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalEquationsWeilRestriction

theorem **aeval_coordOf** (c : β → ℚ) (y : Matrix ((κ × β) ⊕ Unit) ((κ × β) ⊕ Unit) R) (i j : κ) (k : β) : MvPolynomial.aeval (fun ij : ((κ × β) ⊕ Unit) × ((κ × β) ⊕ Unit) => y ij.1 ij.2) (coordOf c (gen ((κ × β) ⊕ Unit)) i j k) = coordOf c y i j k

theorem **aeval_det** (y : Matrix ((κ × β) ⊕ Unit) ((κ × β) ⊕ Unit) R) : MvPolynomial.aeval (fun ij : ((κ × β) ⊕ Unit) × ((κ × β) ⊕ Unit) => y ij.1 ij.2) ((gen ((κ × β) ⊕ Unit)).toBlocks₁₁.det) = y.toBlocks₁₁.det

theorem **aeval_comp_coord** (c : β → ℚ) (y : Matrix ((κ × β) ⊕ Unit) ((κ × β) ⊕ Unit) R) (Q : MvPolynomial ((κ × κ) × β) ℚ) : MvPolynomial.aeval (fun ij : ((κ × β) ⊕ Unit) × ((κ × β) ⊕ Unit) => y ij.1 ij.2) (MvPolynomial.aeval (fun p : (κ × κ) × β => coordOf c (gen ((κ × β) ⊕ Unit)) p.1.1 p.1.2 p.2) Q) = MvPolynomial.aeval (fun p : (κ × κ) × β => coordOf c y p.1.1 p.1.2 p.2) Q

theorem **sum_leftMul_one** (b : Module.Basis β R A) (x : A) (k : β) : ∑ l, Algebra.leftMulMatrix b x k l * b.repr 1 l = b.repr x k

theorem **leftMul_expand** (b : Module.Basis β R A) (x : A) (k l : β) : Algebra.leftMulMatrix b x k l = ∑ m, b.repr x m * Algebra.leftMulMatrix b (b m) k l

theorem **exists_equations** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)), ∃ S' : Set (MvPolynomial (((Fin n × β) ⊕ Unit) × ((Fin n × β) ⊕ Unit)) ℚ), ∀ (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ y : Matrix ((Fin n × β) ⊕ Unit) ((Fin n × β) ⊕ Unit) R, (∀ f ∈ S', ev f y = 0) ↔ ∃ g : GL (Fin n) A, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => (g : Matrix (Fin n) (Fin n) A) ij.1 ij.2) P = 0) ∧ y = emb b g

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalEquationsWeilRestrictionStatement

theorem **_root_.GlobalTheory.T.exists_rational_equations_weil_restriction** : ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)), ∃ S' : Set (MvPolynomial (((Fin n × β) ⊕ Unit) × ((Fin n × β) ⊕ Unit)) ℚ), ∀ (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ y : Matrix ((Fin n × β) ⊕ Unit) ((Fin n × β) ⊕ Unit) R, (∀ f ∈ S', ev f y = 0) ↔ ∃ g : GL (Fin n) A, (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => (g : Matrix (Fin n) (Fin n) A) ij.1 ij.2) P = 0) ∧ y = emb b g

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalInvariantTensor

theorem **coordAlg_mono** {S T : Set (Σ n : ℕ, ((Fin n → ι) → ℚ))} (h : S ⊆ T) : coordAlg S ≤ coordAlg T

theorem **good_zero** (Γ : Set (Matrix ι ι ℂ)) : Good Γ 0

theorem **Good.add** {Γ : Set (Matrix ι ι ℂ)} {φ ψ : Matrix ι ι ℂ → ℂ} (h1 : Good Γ φ) (h2 : Good Γ ψ) : Good Γ (φ + ψ)

theorem **Good.smul** {Γ : Set (Matrix ι ι ℂ)} {φ : Matrix ι ι ℂ → ℂ} (c : ℂ) (h1 : Good Γ φ) : Good Γ (c • φ)

theorem **good_sum** {Γ : Set (Matrix ι ι ℂ)} {β : Type} (s : Finset β) (φ : β → Matrix ι ι ℂ → ℂ) (h : ∀ b, Good Γ (φ b)) : Good Γ (∑ b ∈ s, φ b)

theorem **good_coord** {Γ : Set (Matrix ι ι ℂ)} (hΓ : IsPre Γ) (n : ℕ) (l : (Fin n → ι) → ℂ) (hl : ∀ g ∈ Γ, l ᵥ* tpow n g = l) (b : Fin n → ι) : Good Γ (fun y => (l ᵥ* tpow n y) b)

theorem **good_homog** {Γ : Set (Matrix ι ι ℂ)} (hΓ : IsPre Γ) (hS : SS Γ) (n : ℕ) (F : MvPolynomial (ι × ι) ℂ) (hF : IsInv Γ F) (hn : F.IsHomogeneous n) : Good Γ (evC F)

theorem **exists_finite_family** {Γ : Set (Matrix ι ι ℂ)} (hΓ : IsPre Γ) (hS : SS Γ) : ∃ S : Set (Σ n : ℕ, ((Fin n → ι) → ℚ)), S.Finite ∧ AllFixed Γ S ∧ ∀ f, IsInv Γ f → evC f ∈ coordAlg S

theorem **exists_rational_invariant_tensor** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∃ (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), (∀ g ∈ Γ, act d (fun k => ((t k : ℚ) : ℂ)) g = (fun k => ((t k : ℚ) : ℂ))) ∧ ∀ f, IsInv Γ f → ∃ P : MvPolynomial (TIdx ι d) ℂ, ∀ y : Matrix ι ι ℂ, evC f y = MvPolynomial.eval (act d (fun k => ((t k : ℚ) : ℂ)) y) P

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalInvariantTensorStatement

theorem **_root_.GlobalTheory.T.exists_rational_invariant_tensor'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∃ (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), (∀ g ∈ Γ, act d (fun k => ((t k : ℚ) : ℂ)) g = (fun k => ((t k : ℚ) : ℂ))) ∧ ∀ f, IsInv Γ f → ∃ P : MvPolynomial (TIdx ι d) ℂ, ∀ y : Matrix ι ι ℂ, evC f y = MvPolynomial.eval (act d (fun k => ((t k : ℚ) : ℂ)) y) P

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalMatrixApproxAdeleLinearMap

theorem **GlobalTheory.UnRep.exists_rational_approx_after_adele_linear** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (φ : Matrix (Fin n) (Fin n) (RA F E) →+ Matrix (Fin n) (Fin n) (RA F E)), (∀ (a : FiniteAdeleRing (𝒪 F) F) (M : Matrix (Fin n) (Fin n) (RA F E)), φ ((a ⊗ₜ[F] (1 : E) : RA F E) • M) = (a ⊗ₜ[F] (1 : E) : RA F E) • φ M) → ∀ (M : Matrix (Fin n) (Fin n) (RA F E)) (D : ℕ), 0 < D → ∃ M₀ : Matrix (Fin n) (Fin n) E, ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), φ (M − fm F M₀) *ᵥ z = D • w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalMatrixApproxAdeleLinearMapStatement

theorem **_root_.GlobalTheory.T.exists_rational_matrix_approx_adele_linear_map** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (φ : Matrix (Fin n) (Fin n) (RA F E) →+ Matrix (Fin n) (Fin n) (RA F E)), (∀ (a : FiniteAdeleRing (𝒪 F) F) (M : Matrix (Fin n) (Fin n) (RA F E)), φ ((a ⊗ₜ[F] (1 : E) : RA F E) • M) = (a ⊗ₜ[F] (1 : E) : RA F E) • φ M) → ∀ (M : Matrix (Fin n) (Fin n) (RA F E)) (D : ℕ), 0 < D → ∃ M₀ : Matrix (Fin n) (Fin n) E, ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), φ (M − fm F M₀) *ᵥ z = D • w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalUnitarySplitDenominator

theorem **exists_dvd_coprime_mul_pow** (N : ℕ) (m : ℕ) (hm : 0 < m) : ∃ j u : ℕ, u.Coprime N ∧ m ∣ u * N ^ j

theorem **inv_integral** {V : Type} [AddCommGroup V] (Λ : AddSubgroup V) (k k' : V →+ V) (hk : ∀ z, k' (k z) = z) (N t m : ℕ) (ht : t.Coprime N) (hm : 0 < m) (h1 : ∀ z ∈ Λ, ∃ w ∈ Λ, t • (k z − z) = N • w) (h2 : ∀ z ∈ Λ, m • k' z ∈ Λ) : ∃ t' : ℕ, t'.Coprime N ∧ ∀ z ∈ Λ, t' • k' z ∈ Λ

theorem **exists_mem_primes** {E : Type} [Field E] [NumberField E] (I : Ideal (𝒪 E)) (hI : I ≠ ⊥) : ∃ s ∈ I, s ≠ 0 ∧ ∀ P : Ideal (𝒪 E), P.IsPrime → s ∈ P → I ≤ P

theorem **fm_mul** (A B : Matrix (Fin n) (Fin n) E) : fm F (A * B) = fm F A * fm F B

theorem **fm_one** : fm F (1 : Matrix (Fin n) (Fin n) E) = 1

theorem **fm_mulVec_pure** (M : Matrix (Fin n) (Fin n) E) (a : FiniteAdeleRing (𝒪 F) F) (x : Fin n → E) : fm F M *ᵥ (fun i => a ⊗ₜ[F] x i) = fun i => a ⊗ₜ[F] (M *ᵥ x) i

theorem **congAtA_of_congAt** (L : Submodule (𝒪 E) (Fin n → E)) (N D : ℕ) (δ : Matrix (Fin n) (Fin n) E) (h : CongAt L N D δ) : CongAtA F L N D (fm F δ)

theorem **smul_fm_mem** (L : Submodule (𝒪 E) (Fin n → E)) (s : 𝒪 E) (M : Matrix (Fin n) (Fin n) E) (h : ∀ v ∈ L, s • (M *ᵥ v) ∈ L) : ∀ z ∈ hat F (L : Set (Fin n → E)), ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • (fm F M *ᵥ z) ∈ hat F (L : Set (Fin n → E))

theorem **smul_mem_hat** (L : Submodule (𝒪 E) (Fin n → E)) (s : 𝒪 E) : ∀ z ∈ hat F (L : Set (Fin n → E)), ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • z ∈ hat F (L : Set (Fin n → E))

theorem **congAtA_mul** (L : Submodule (𝒪 E) (Fin n → E)) (N D : ℕ) (a b : Matrix (Fin n) (Fin n) (RA F E)) (ha : CongAtA F L N D a) (hb : CongAtA F L N D b) : CongAtA F L N D (a * b)

theorem **one_tmul_natCast_smul** (m : ℕ) (z : Fin n → RA F E) : ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (((m : 𝒪 E)) : E)) • z = m • z

theorem **one_tmul_natCast_mul_smul** (m : ℕ) (s : 𝒪 E) (z : Fin n → RA F E) : ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] ((((m : 𝒪 E) * s : 𝒪 E)) : E)) • z = m • ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • z

theorem **exists_rep_split_denominator** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (RA F E)) (N : ℕ), IsUnA c G g.1 → 0 < N → ∃ γ : GL (Fin n) E, IsUnE c G γ.1 ∧ CongAtA F L N N (fm F γ.1 * g.1) ∧ ∃ s : 𝒪 E, SplitOff c N s ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • ((fm F γ.1 * g.1) *ᵥ z) ∈ hat F (L : Set (Fin n → E)) ∧ ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • (((g⁻¹).1 * fm F (γ⁻¹).1) *ᵥ z) ∈ hat F (L : Set (Fin n → E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRationalUnitarySplitDenominatorStatement

theorem **_root_.GlobalTheory.T.exists_rational_unitary_split_denominator** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (RA F E)) (N : ℕ), IsUnA c G g.1 → 0 < N → ∃ γ : GL (Fin n) E, IsUnE c G γ.1 ∧ CongAtA F L N N (fm F γ.1 * g.1) ∧ ∃ s : 𝒪 E, SplitOff c N s ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • ((fm F γ.1 * g.1) *ᵥ z) ∈ hat F (L : Set (Fin n → E)) ∧ ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] (s : E)) • (((g⁻¹).1 * fm F (γ⁻¹).1) *ᵥ z) ∈ hat F (L : Set (Fin n → E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRealPlaceSymplecticFormPositiveComplexStru

theorem **map_inv_of_det_ne** {E : Type} [Field E] {n : ℕ} (φ : E →+* ℂ) (G : Matrix (Fin n) (Fin n) E) (h : G.det ≠ 0) : (G.map φ)⁻¹ = G⁻¹.map φ

theorem **trace_eq_sum** {m n : ℕ} (P : Matrix (Fin n) (Fin n) ℂ) (Q : Matrix (Fin m) (Fin m) ℂ) (x y : Matrix (Fin m) (Fin n) ℂ) : (P * yᴴ * Q * x).trace = ∑ a, ∑ b, ∑ e, ∑ d, P a b * starRingEnd ℂ (y e b) * Q e d * x d a

theorem **formAt_eq** (S T : HSp D) (τ : D.what₂.Tau) (x y : Matrix (Fin T.rank) (Fin S.rank) ℂ) : formAt S T τ x y = 2 * (tauE D τ D.host.xi0 * ((gramAt D S τ)⁻¹ * yᴴ * gramAt D T τ * x).trace).re

theorem **realPlace_imaginaryGenerator_re_eq_zero** (τ : D.what₂.Tau) : (tauE D τ D.host.xi0).re = 0

theorem **cplxO_add** {m n : ℕ} (s : Matrix (Fin n) (Fin n) ℂ) (t : Matrix (Fin m) (Fin m) ℂ) (x y : Matrix (Fin m) (Fin n) ℂ) : cplxO s t (x + y) = cplxO s t x + cplxO s t y

theorem **cplxO_smul** {m n : ℕ} (s : Matrix (Fin n) (Fin n) ℂ) (t : Matrix (Fin m) (Fin m) ℂ) (c : ℝ) (x : Matrix (Fin m) (Fin n) ℂ) : cplxO s t (c • x) = c • cplxO s t x

theorem **exists_real_form** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), IsStdChar D ψ (-1) → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∃ (om : GlobalTheory.FockCompare.RW T S →ₗ[ℝ] GlobalTheory.FockCompare.RW T S →ₗ[ℝ] ℝ) (J : GlobalTheory.FockCompare.RW T S →ₗ[ℝ] GlobalTheory.FockCompare.RW T S), (∀ x y : WO S T, y.map (finPart D) = 0 → ψ (symplOH S T x y) = Complex.exp (-(Real.pi * Complex.I) * (om (GlobalTheory.FockCompare.piW θ T S x) (GlobalTheory.FockCompare.piW θ T S y) : ℝ))) ∧ (∀ z τ, J z τ = cplxO (sS τ) (sT τ) (z τ)) ∧ (∀ v, om v v = 0) ∧ (∀ v, J (J v) = -v) ∧ (∀ x y, om x (J y) = om y (J x)) ∧ ∀ v, v ≠ 0 → 0 < om v (J v)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRealPlaceSymplecticFormPositiveComplexStruStatement

theorem **_root_.GlobalTheory.T.exists_realPlace_sympleсticForm_positiveComplexStructure** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), IsStdChar D ψ (-1) → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∃ (om : GlobalTheory.FockCompare.RW T S →ₗ[ℝ] GlobalTheory.FockCompare.RW T S →ₗ[ℝ] ℝ) (J : GlobalTheory.FockCompare.RW T S →ₗ[ℝ] GlobalTheory.FockCompare.RW T S), (∀ x y : WO S T, y.map (finPart D) = 0 → ψ (symplOH S T x y) = Complex.exp (-(Real.pi * Complex.I) * (om (GlobalTheory.FockCompare.piW θ T S x) (GlobalTheory.FockCompare.piW θ T S y) : ℝ))) ∧ (∀ z τ, J z τ = cplxO (sS τ) (sT τ) (z τ)) ∧ (∀ v, om v v = 0) ∧ (∀ v, J (J v) = -v) ∧ (∀ x y, om x (J y) = om y (J x)) ∧ ∀ v, v ≠ 0 → 0 < om v (J v)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRealTranslateIsMinNormTensorInOrbit

private lemma **posDef_det_one_eq_unitary_conj_exp_diagonal** {ι : Type} [Fintype ι] [DecidableEq ι] (P : Matrix ι ι ℂ) (hP : P.PosDef) (hdet : P.det = 1) : ∃ (U : Matrix ι ι ℂ) (δ : ι → ℝ), U * Uᴴ = 1 ∧ ∑ i, δ i = 0 ∧ P = (U * diagonal fun i => ((Real.exp (2 * δ i) : ℝ) : ℂ)) * Uᴴ

private lemma **unitary_conj_exp_diagonal_conjTranspose** {ι : Type} [Fintype ι] [DecidableEq ι] (U : Matrix ι ι ℂ) (δ : ι → ℝ) (s : ℝ) : ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ)ᴴ = (U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ

private lemma **unitary_conj_exp_diagonal_mul** {ι : Type} [Fintype ι] [DecidableEq ι] {U : Matrix ι ι ℂ} (hU : U * Uᴴ = 1) (δ : ι → ℝ) (s t : ℝ) : ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ) * ((U * diagonal fun i => ((Real.exp (t * δ i) : ℝ) : ℂ)) * Uᴴ) = (U * diagonal fun i => ((Real.exp ((s + t) * δ i) : ℝ) : ℂ)) * Uᴴ

private lemma **exists_unitary_exp_diagonal_fixing_of_nsq_eq** {ι : Type} [Fintype ι] [DecidableEq ι] {m : ℕ} (d : Fin m → ℕ) (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (hv : _root_.GlobalTheory.ClassFin.IsMinVec d v) (g : Matrix ι ι ℂ) (hg : g.det = 1) (hn : _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v g) = _root_.GlobalTheory.ClassFin.nsq v) : ∃ (U : Matrix ι ι ℂ) (δ : ι → ℝ), U * Uᴴ = 1 ∧ ∑ i, δ i = 0 ∧ g * gᴴ = (U * diagonal fun i => ((Real.exp (2 * δ i) : ℝ) : ℂ)) * Uᴴ ∧ ∀ s : ℝ, _root_.GlobalTheory.ClassFin.act d v ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ) = v

private lemma **map_conj_mul_conjTranspose_eq_one** {ι : Type} [Fintype ι] [DecidableEq ι] {U : Matrix ι ι ℂ} (hU : U * Uᴴ = 1) : U.map (starRingEnd ℂ) * (U.map (starRingEnd ℂ))ᴴ = 1

private lemma **unitary_conj_exp_diagonal_map_conj** {ι : Type} [Fintype ι] [DecidableEq ι] (U : Matrix ι ι ℂ) (δ : ι → ℝ) (s : ℝ) : ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ).map (starRingEnd ℂ) = (U.map (starRingEnd ℂ) * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * (U.map (starRingEnd ℂ))ᴴ

private lemma **unitary_conj_exp_zero_diagonal_eq_one** {ι : Type} [Fintype ι] [DecidableEq ι] {U : Matrix ι ι ℂ} (hU : U * Uᴴ = 1) (δ : ι → ℝ) : (U * diagonal fun i => ((Real.exp (0 * δ i) : ℝ) : ℂ)) * Uᴴ = 1

private lemma **posSemidef_eq_of_mul_self_eq** {ι : Type} [Fintype ι] [DecidableEq ι] (X Y : Matrix ι ι ℂ) (hX : X.PosSemidef) (hY : Y.PosSemidef) (h : X * X = Y * Y) : X = Y

private lemma **nsq_act_eq_of_gram_eq** {ι : Type} [Fintype ι] [DecidableEq ι] {m : ℕ} (d : Fin m → ℕ) (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (a b : Matrix ι ι ℂ) (ha : IsUnit a.det) (hab : b * bᴴ = a * aᴴ) : _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v b) = _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v a)

theorem **_root_.GlobalTheory.ClassFin.exists_real_translate_isMinVec** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (t : _root_.GlobalTheory.ClassFin.TIdx ι d → ℝ), (∃ w ∈ _root_.GlobalTheory.ClassFin.orb d (fun k => (t k : ℂ)), _root_.GlobalTheory.ClassFin.IsMinVec d w) → ∃ g : Matrix ι ι ℝ, g.det = 1 ∧ _root_.GlobalTheory.ClassFin.IsMinVec d (_root_.GlobalTheory.ClassFin.act d (fun k => (t k : ℂ)) (g.map (fun r : ℝ => (r : ℂ))))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRealTranslateIsMinNormTensorInOrbitStatement

theorem **_root_.GlobalTheory.T.exists_real_translate_isMinNormTensorInOrbit** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℝ), (∃ w ∈ orb d (fun k => (t k : ℂ)), IsMinVec d w) → ∃ g : Matrix ι ι ℝ, g.det = 1 ∧ IsMinVec d (act d (fun k => (t k : ℂ)) (g.map (fun r : ℝ => (r : ℂ))))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRepSuppSplit

theorem **finrank_two** (D : D7) : Module.finrank D.host.F D.host.E = 2

theorem **isUnA_finHom** {D : D7} (V : HSp D) (g : GA D V) : GlobalTheory.UnRep.IsUnA (cF D) V.gramᵀ (finHom V g).1

theorem **herm_neg_eq** {D : D7} (V : HSp D) (u v : Fin V.rank → D.host.E) : herm (-V.gramᵀ) u v = -GlobalTheory.UnRep.hermE (cF D) V.gramᵀ u v

theorem **exists_rep_suppSplit** : ∀ (D : D7) (V : HSp D) (N : ℕ), 0 < N → ∀ g : GA D V, ∃ γ ∈ GQ V, ∃ a ∈ GinfC V, ∃ h ∈ GfC V, ∃ k ∈ GfC V, g = γ * a * h * k ∧ FinCong V N k ∧ GlobalTheory.UnRep.SuppSplit (cF D) N (finHom V h).1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsReynoldsOfCompletelyReducible

private lemma **projection_comm_of_mapsTo** {p q : Submodule ℂ V} (h : IsCompl p q) (T : V →ₗ[ℂ] V) (hp : ∀ v ∈ p, T v ∈ p) (hq : ∀ v ∈ q, T v ∈ q) (v : V) : p.projection q h (T v) = T (p.projection q h v)

private lemma **sub_projection_mem'** {p q : Submodule ℂ V} (h : IsCompl p q) (v : V) : v - p.projection q h v ∈ q

theorem **_root_.GlobalTheory.ClassFin.exists_reynolds_of_completelyReducible** (V : Type) [AddCommGroup V] [Module ℂ V] (S : Set (V →ₗ[ℂ] V)) : (∀ U : Submodule ℂ V, (∀ s ∈ S, ∀ v ∈ U, (s : V → V) v ∈ U) → ∃ U' : Submodule ℂ V, (∀ s ∈ S, ∀ v ∈ U', (s : V → V) v ∈ U') ∧ IsCompl U U') → ∃ Rey : V →ₗ[ℂ] V, (∀ v : V, ∀ s ∈ S, (s : V → V) ((Rey : V → V) v) = (Rey : V → V) v) ∧ (∀ v : V, (∀ s ∈ S, (s : V → V) v = v) → (Rey : V → V) v = v) ∧ (∀ v : V, ∀ s ∈ S, (Rey : V → V) ((s : V → V) v) = (Rey : V → V) v) ∧ (∀ v : V, v - (Rey : V → V) v ∈ Submodule.span ℂ {x : V | ∃ s ∈ S, ∃ w : V, x = (s : V → V) w - w}) ∧ (∀ T : V →ₗ[ℂ] V, (∀ s ∈ S, ∀ v : V, (T : V → V) ((s : V → V) v) = (s : V → V) ((T : V → V) v)) → ∀ v : V, (Rey : V → V) ((T : V → V) v) = (T : V → V) ((Rey : V → V) v)) ∧ ∀ U : Submodule ℂ V, (∀ s ∈ S, ∀ v ∈ U, (s : V → V) v ∈ U) → ∀ v ∈ U, (Rey : V → V) v ∈ U

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsReynoldsOfCompletelyReducibleStatement

theorem **_root_.GlobalTheory.T.exists_reynolds_of_completelyReducible'** : ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (S : Set (V →ₗ[ℂ] V)), (∀ U : Submodule ℂ V, (∀ s ∈ S, ∀ v ∈ U, s v ∈ U) → ∃ U' : Submodule ℂ V, (∀ s ∈ S, ∀ v ∈ U', s v ∈ U') ∧ IsCompl U U') → ∃ Rey : V →ₗ[ℂ] V, (∀ v, ∀ s ∈ S, s (Rey v) = Rey v) ∧ (∀ v, (∀ s ∈ S, s v = v) → Rey v = v) ∧ (∀ v, ∀ s ∈ S, Rey (s v) = Rey v) ∧ (∀ v, v - Rey v ∈ Submodule.span ℂ {x | ∃ s ∈ S, ∃ w, x = s w - w}) ∧ (∀ T : V →ₗ[ℂ] V, (∀ s ∈ S, ∀ v, T (s v) = s (T v)) → ∀ v, Rey (T v) = T (Rey v)) ∧ ∀ U : Submodule ℂ V, (∀ s ∈ S, ∀ v ∈ U, s v ∈ U) → ∀ v ∈ U, Rey v ∈ U

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRingEquivApplyNeOfIsAlgebraic

theorem **exists_ringEquiv_apply_ne_of_isAlgebraic** : ∀ z : ℂ, IsAlgebraic ℚ z → (∀ q : ℚ, z ≠ q) → ∃ σ : ℂ ≃+* ℂ, σ z ≠ z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRingEquivNeOfTranscendental

theorem **exists_ringEquiv_ne_of_transcendental** : ∀ z : ℂ, Transcendental ℚ z → ∃ σ : ℂ ≃+* ℂ, σ z ≠ z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRingHomAdicCompletion

theorem **GlobalTheory.LocalQuad.exists_ringHom_adicCompletion** (K L : Type) [Field K] [NumberField K] [Field L] [NumberField L] [Algebra K L] (w : HeightOneSpectrum (𝒪 L)) : ∃ (v : HeightOneSpectrum (𝒪 K)) (φ : v.adicCompletion K →+* w.adicCompletion L), ∀ x : K, φ (algebraMap K (v.adicCompletion K) x) = algebraMap L (w.adicCompletion L) (algebraMap K L x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsRingHomAdicCompletionUnder

theorem **GlobalTheory.QNorm.exists_ringHom_adicCompletion_under** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (w : HeightOneSpectrum (𝒪 E)), ∃ φ : (w.under (𝒪 F)).adicCompletion F →+* w.adicCompletion E, ∀ x : F, φ (algebraMap F ((w.under (𝒪 F)).adicCompletion F) x) = algebraMap E (w.adicCompletion E) (algebraMap F E x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSemicharacterCompletedSelfDualLattice

theorem **exists_abstract** {G R : Type} [AddCommGroup G] [AddCommGroup R] (ψ : AddChar R ℂ) (hψ : ∀ r, ‖ψ r‖ ≤ 1) (B : G → G → R) (hl : ∀ x y z, B (x + y) z = B x z + B y z) (hr : ∀ x y z, B x (y + z) = B x y + B x z) (ha : ∀ x, B x x = 0) (L : AddSubgroup G) (K : G → Prop) (K0 : K 0) (Ksub : ∀ x y, K x → K y → K (x - y)) (hKL : ∀ l ∈ L, K l → l = 0) (hB : ∀ x, K x → ∀ l ∈ L, B x l = 0) (h2 : ∀ a ∈ L, ∀ u ∈ L, ψ (B a u + B a u) = 1) (p q : G → G) (padd : ∀ x y, p (x + y) = p x + p y) (pq : ∀ x, p x + q x = x) (adj : ∀ x y, B (p x) y = B x (q y)) (pL : ∀ l ∈ L, p l ∈ L) : ∃ χ : G → ℂ, (∀ x, (¬ ∃ l ∈ L, K (x - l)) → χ x = 0) ∧ (∀ x, ‖χ x‖ ≤ 1) ∧ χ 0 = 1 ∧ (∀ x y, K y → χ (x + y) = χ x) ∧ (∀ u ∈ L, ∀ x, ψ (B x (2 • u)) * χ (x + 2 • u) = χ x) ∧ ∀ u ∈ L, B (p u) u = 0 → ∀ x, ψ (B x u) * χ (x + u) = χ x

theorem **latO_inf** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : HeightOneSpectrum (𝒪 D.host.F) → Prop) : ∀ l ∈ latO S T LS LT P, l.map (infPart D) = 0

theorem **map_mul_add** {m n : ℕ} (a : AE D) (x y : Mx D m n) : (x + y).map (fun z => a * z) = x.map (fun z => a * z) + y.map (fun z => a * z)

theorem **latO_mul_int** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : HeightOneSpectrum (𝒪 D.host.F) → Prop) (b : 𝒪 D.host.E) : ∀ l ∈ latO S T LS LT P, l.map (fun z => incA D (b : D.host.E) * z) ∈ latO S T LS LT P

theorem **exists_int_add_star** (D : D7) : ∃ b : 𝒪 D.host.E, (b : D.host.E) + star (b : D.host.E) = 1

theorem **exists_semichar** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → (∀ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), l.map (infPart D) = 0) ∧ ∃ χ : WO S T → ℂ, (∀ x, (¬ ∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), (x - l).map (finPart D) = 0) → χ x = 0) ∧ (∀ x, ‖χ x‖ ≤ 1) ∧ χ 0 = 1 ∧ (∀ x y, y.map (finPart D) = 0 → χ (x + y) = χ x) ∧ (∀ u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), ∀ x, ψ (symplOH S T x (2 • u)) * χ (x + 2 • u) = χ x) ∧ ∀ e, IsSplitIdem D e → ∀ u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSemicharacterCompletedSelfDualLatticeStatement

theorem **_root_.GlobalTheory.T.exists_semicharacter_completedSelfDualLattice** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → (∀ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), l.map (infPart D) = 0) ∧ ∃ χ : WO S T → ℂ, (∀ x, (¬ ∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), (x - l).map (finPart D) = 0) → χ x = 0) ∧ (∀ x, ‖χ x‖ ≤ 1) ∧ χ 0 = 1 ∧ (∀ x y, y.map (finPart D) = 0 → χ (x + y) = χ x) ∧ (∀ u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), ∀ x, ψ (symplOH S T x (2 • u)) * χ (x + 2 • u) = χ x) ∧ ∀ e, IsSplitIdem D e → ∀ u ∈ (latO S T (ΛS : Set (Fin S.rank

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSmoothCompactSupportHasSumLatticeTranslate

theorem **exists_radius_lattice_translate_norm_le** (V : Type) [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L] : ∃ R : ℝ, 0 < R ∧ ∀ y : V, ∃ l : L, ‖y + l‖ ≤ R

theorem **exists_smooth_compactSupport_hasSum_lattice_translates** : ∀ (V : Type) [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L], ∃ η : V → ℝ, ContDiff ℝ ∞ η ∧ HasCompactSupport η ∧ ∀ y, HasSum (fun l : L => η (y + l)) 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSmoothHomogeneousHasSumUnitSmul

theorem **hasSum_quot** {G : Type} [Group G] (T : Subgroup G) [Finite T] (f : G ⧸ T → ℝ) (s : ℝ) (hf : HasSum f s) : HasSum (fun g : G => f (g : G ⧸ T) / Nat.card T) s

theorem **contDiffAt_normAtPlace** (w : InfinitePlace K) (x : mixedSpace K) (hx : normAtPlace w x ≠ 0) : ContDiffAt ℝ ∞ (fun y : mixedSpace K => normAtPlace w y) x

theorem **contDiffAt_logMap** (x : mixedSpace K) (hx : mixedEmbedding.norm x ≠ 0) : ContDiffAt ℝ ∞ (logMap (K

private lemma **le_of_mul_le_of_one_le'** {m t B : ℝ} (hm : 1 ≤ m) (hB : 0 ≤ B) (h : m * t ≤ B) : t ≤ B

theorem **log_le** (x : mixedSpace K) (hx : mixedEmbedding.norm x ≠ 0) (w : InfinitePlace K) : Real.log (normAtPlace w x) ≤ Fintype.card (InfinitePlace K) * ‖logMap x‖ + Real.log (mixedEmbedding.norm x) * (Module.finrank ℚ K : ℝ)⁻¹

theorem **exists_smooth_homogeneous_hasSum_unit_smul** : ∀ (K : Type) [Field K] [NumberField K], ∃ h : mixedSpace K → ℝ, ContDiffOn ℝ ∞ h {x | mixedEmbedding.norm x ≠ 0} ∧ (∀ (t : ℝ) x, 0 < t → h (t • x) = h x) ∧ (∀ x, mixedEmbedding.norm x ≠ 0 → HasSum (fun u : (𝒪 K)ˣ => h (u • x)) 1) ∧ (∀ a b : ℝ, 0 < a → ∃ C : Set (mixedSpace K), IsCompact C ∧ (∀ x ∈ C, mixedEmbedding.norm x ≠ 0) ∧ ∀ x, h x ≠ 0 → a ≤ mixedEmbedding.norm x → mixedEmbedding.norm x ≤ b → x ∈ C)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSubSqLtOneOfTwoLtOne

theorem **exists_sub_sq_mem_of_two_mem** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (h2 : (2 : 𝒪 F) ∈ v.asIdeal) (a : 𝒪 F) : ∃ b : 𝒪 F, a - b ^ 2 ∈ v.asIdeal

theorem **exists_sub_sq_lt_one_of_two_lt_one** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), Valued.v (2 : v.adicCompletion F) < 1 → ∀ x : v.adicCompletion F, Valued.v x ≤ 1 → ∃ y : v.adicCompletion F, Valued.v y ≤ 1 ∧ Valued.v (x - y ^ 2) < 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSumIntegralBasis

theorem **isClosed_int_add_mul** {A : Type} [CommRing A] [TopologicalSpace A] [IsTopologicalRing A] (O : Subring A) (M : ℕ) (hM : 0 < M) (hcl : IsClosed {a : A | ∃ c ∈ O, a = (M : A) * c}) : IsClosed {x : A | ∃ (z : ℤ) (c : A), c ∈ O ∧ x = (z : A) + (M : A) * c}

theorem **exists_sum_of_approx** {A : Type} [CommRing A] [TopologicalSpace A] [IsTopologicalRing A] (O : Subring A) (hO : IsCompact (O : Set A)) (hcl : ∀ M : ℕ, 0 < M → IsClosed {a : A | ∃ c ∈ O, a = (M : A) * c}) (hz : ∀ a : A, (∀ M : ℕ, 0 < M → ∃ c ∈ O, a = (M : A) * c) → a = 0) {β : Type} [Fintype β] (w : β → A) (a : A) (ha : ∀ M : ℕ, 0 < M → ∃ (z : β → ℤ) (c : A), c ∈ O ∧ a = ∑ i, (z i : A) * w i + (M : A) * c) : ∃ r : β → A, (∀ i, ∀ M : ℕ, 0 < M → ∃ (z : ℤ) (c : A), c ∈ O ∧ r i = (z : A) + (M : A) * c) ∧ a = ∑ i, r i * w i

theorem **exists_sum_integralBasis** : ∀ (F : Type) [Field F] [NumberField F] (β : Type) [Fintype β] (ω : Module.Basis β ℤ (𝒪 F)) (a : FiniteAdeleRing (𝒪 F) F), (∀ v, (a : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) → ∃ r : β → FiniteAdeleRing (𝒪 F) F, (∀ i, (∀ M : ℕ, 0 < M → ∃ (z : ℤ) (c : FiniteAdeleRing (𝒪 F) F), (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ r i = (z : FiniteAdeleRing (𝒪 F) F) + (M : FiniteAdeleRing (𝒪 F) F) * c)) ∧ a = ∑ i, r i * algebraMap F (FiniteAdeleRing (𝒪 F) F) (ω i : 𝒪 F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsSupersetMemSUnits

theorem **finite_valuation_ne_one** (F : Type) [Field F] [NumberField F] (x : Fˣ) : {v : HeightOneSpectrum (𝒪 F) | v.valuation F (x : F) ≠ 1}.Finite

theorem **exists_superset_mem_sUnits** : ∀ (F : Type) [Field F] [NumberField F] (X : Finset Fˣ) (S₀ : Finset (HeightOneSpectrum (𝒪 F))), ∃ S, S₀ ⊆ S ∧ ∀ S', S ⊆ S' → ∀ x ∈ X, x ∈ sUnits F S'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTensorFiniteAdeleEquiv

theorem **exists_tensor_finiteAdele_equiv** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], ∃ ι : FiniteAdeleRing (𝒪 F) F ⊗[F] E ≃+* FiniteAdeleRing (𝒪 E) E, (∀ x : E, ι (1 ⊗ₜ[F] x) = algebraMap E _ x) ∧ (∀ y, y ∈ AddSubgroup.closure {y : FiniteAdeleRing (𝒪 F) F ⊗[F] E | ∃ a ∈ {a : FiniteAdeleRing (𝒪 F) F | ∀ v : HeightOneSpectrum (𝒪 F), a v ∈ v.adicCompletionIntegers F}, ∃ x : 𝒪 E, y = a ⊗ₜ[F] (x : E)} ↔ ∀ P, Valued.v (ι y P) ≤ 1) ∧ ∀ (v : HeightOneSpectrum (𝒪 F)) (α : FiniteAdeleRing (𝒪 F) F), α v = 1 → (∀ v', v' ≠ v → α v' = 0) → ∀ P : HeightOneSpectrum (𝒪 E), ι (α ⊗ₜ[F] 1) P = if P.under (𝒪 F) = v then 1 else 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTensorStabiliserClosedSLOrbit

theorem **exists_tensor_stabiliser_closed_orbit** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsQGroup Γ → SS Γ → ∃ (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), (∀ g ∈ Γ, act d (fun k => ((t k : ℚ) : ℂ)) g = (fun k => ((t k : ℚ) : ℂ))) ∧ (∀ y : Matrix ι ι ℂ, y.det = 1 → act d (fun k => ((t k : ℚ) : ℂ)) y = (fun k => ((t k : ℚ) : ℂ)) → y ∈ Γ) ∧ IsClosed (orb d (fun k => ((t k : ℚ) : ℂ)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTensorStabiliserClosedSLOrbitStatement

theorem **_root_.GlobalTheory.T.exists_tensor_stabiliser_closed_SL_orbit** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsQGroup Γ → SS Γ → ∃ (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), (∀ g ∈ Γ, act d (fun k => ((t k : ℚ) : ℂ)) g = (fun k => ((t k : ℚ) : ℂ))) ∧ (∀ y : Matrix ι ι ℂ, y.det = 1 → act d (fun k => ((t k : ℚ) : ℂ)) y = (fun k => ((t k : ℚ) : ℂ)) → y ∈ Γ) ∧ IsClosed (orb d (fun k => ((t k : ℚ) : ℂ)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTraceZeroAdjustmentOfHermitianLattice

theorem **exists_split** (b : ℕ) (N : ℕ) (hN : 0 < N) : ∃ A B e : ℕ, N = A * B ∧ A ∣ b ^ e ∧ B.Coprime b

theorem **exists_nsmul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (y : Fin n → E) : ∃ N : ℕ, 0 < N ∧ N • y ∈ L

theorem **exists_nsmul_mulVec_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hfg : L.FG) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (M : Matrix (Fin n) (Fin n) E) : ∃ N : ℕ, 0 < N ∧ ∀ v ∈ L, N • (M *ᵥ v) ∈ L

theorem **tv_mul_mulVec** {R : Type} [CommRing R] {n : ℕ} (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (x v : Fin n → R) (t : R) (q : ℤ) : tv σ G x (q * t) *ᵥ v = v + q • ((tv σ G x t - 1) *ᵥ v)

theorem **exists_tv_adjust** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), Module.finrank F E = 2 → IsHerm (c : E →+* E) G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (x : Fin n → E) (t : E) (b : ℕ), 0 < b → t + c t = 0 → ∃ (t' : E) (k e : ℕ), t' + c t' = 0 ∧ k.Coprime b ∧ ∀ v ∈ L, (∃ w ∈ L, k • (tv (c : E →+* E) G x t' *ᵥ v - tv (c : E →+* E) G x t *ᵥ v) = b • w) ∧ b ^ e • (tv (c : E →+* E) G x t' *ᵥ v) ∈ L ∧ b ^ e • (tv (c : E →+* E) G x (-t') *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTraceZeroAdjustmentOfHermitianLatticeStatement

theorem **_root_.GlobalTheory.T.exists_traceZero_adjustment_of_hermitian_lattice** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), Module.finrank F E = 2 → IsHerm (c : E →+* E) G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (x : Fin n → E) (t : E) (b : ℕ), 0 < b → t + c t = 0 → ∃ (t' : E) (k e : ℕ), t' + c t' = 0 ∧ k.Coprime b ∧ ∀ v ∈ L, (∃ w ∈ L, k • (tv (c : E →+* E) G x t' *ᵥ v - tv (c : E →+* E) G x t *ᵥ v) = b • w) ∧ b ^ e • (tv (c : E →+* E) G x t' *ᵥ v) ∈ L ∧ b ^ e • (tv (c : E →+* E) G x (-t') *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTwoPlacesOverOfIsSquareLocal

theorem **val_integer_le_one** {E Γ : Type} [Field E] [LinearOrderedCommGroupWithZero Γ] (u : Valuation E Γ) (x : 𝒪 E) : u (algebraMap (𝒪 E) E x) ≤ 1

theorem **exists_centre_over** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) (u : Valuation E ℤᵐ⁰) (hu : ∀ a : F, u (algebraMap F E a) = v.valuation F a) : ∃ w : HeightOneSpectrum (𝒪 E), w.under (𝒪 F) = v ∧ ∀ x : 𝒪 E, x ∈ w.asIdeal ↔ u (algebraMap (𝒪 E) E x) < 1

theorem **IsQuad.exists_embedding_root** {F E L : Type} [Field F] [Field E] [Algebra F E] [Field L] {δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsQuad δ θ σ) (φ : F →+* L) (r : L) (hr : φ δ = r * r) : ∃ ψ : E →+* L, (∀ a : F, ψ (algebraMap F E a) = φ a) ∧ ψ θ = r

theorem **exists_val_sub_mul_lt_one** {F L Γ : Type} [Field L] [LinearOrderedCommGroupWithZero Γ] [Valued L Γ] (φ : F → L) (hφ : DenseRange φ) (r c : L) : ∃ ρ : F, Valued.v ((r - φ ρ) * c) < (1 : Γ)

theorem **two_places_of_isSquare** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ v : HeightOneSpectrum (𝒪 F), IsSquare (algebraMap F (v.adicCompletion F) δ) → ∃ w w' : HeightOneSpectrum (𝒪 E), w ≠ w' ∧ w.under (𝒪 F) = v ∧ w'.under (𝒪 F) = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsTwoPlacesOverOfIsSquareLocalStatement

theorem **_root_.GlobalTheory.T.exists_two_places_over_of_isSquare_local** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ v : HeightOneSpectrum (𝒪 F), IsSquare (algebraMap F (v.adicCompletion F) δ) → ∃ w w' : HeightOneSpectrum (𝒪 E), w ≠ w' ∧ w.under (𝒪 F) = v ∧ w'.under (𝒪 F) = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUniqueIdeleOfLinearEquivQuotientLocal

theorem **valuation_eq_exp_neg_count** (v : HeightOneSpectrum R) {x : K} (hx : x ≠ 0) : v.valuation K x = exp (-FractionalIdeal.count K v (FractionalIdeal.spanSingleton R⁰ x))

theorem **le_of_count_le** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) (h : ∀ v : HeightOneSpectrum R, FractionalIdeal.count K v J ≤ FractionalIdeal.count K v I) : I ≤ J

theorem **eq_of_count_eq** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) (h : ∀ v : HeightOneSpectrum R, FractionalIdeal.count K v I = FractionalIdeal.count K v J) : I = J

theorem **mem_iff_valuation_le** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (x : K) : x ∈ I ↔ ∀ v : HeightOneSpectrum R, v.valuation K x ≤ exp (-FractionalIdeal.count K v I)

theorem **exists_mem_valuation_eq** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (v : HeightOneSpectrum R) : ∃ y ∈ I, v.valuation K y = exp (-FractionalIdeal.count K v I)

theorem **valued_algebraMap** (v : HeightOneSpectrum R) (x : K) : Valued.v (algebraMap K (v.adicCompletion K) x) = v.valuation K x

theorem **mem_localIdeal_iff** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (v : HeightOneSpectrum R) (x : v.adicCompletion K) : x ∈ localIdeal v I ↔ Valued.v x ≤ exp (-FractionalIdeal.count K v I)

theorem **localIdeal_zero** (v : HeightOneSpectrum R) : localIdeal v (0 : FractionalIdeal R⁰ K) = ⊥

theorem **mem_localIdeal_one_iff** (v : HeightOneSpectrum R) (x : v.adicCompletion K) : x ∈ localIdeal v (1 : FractionalIdeal R⁰ K) ↔ x ∈ v.adicCompletionIntegers K

@[simp] theorem **adele_mul_apply** (a b : 𝔸ᶠ[R, K]) (v : HeightOneSpectrum R) : (a * b) v = a v * b v

@[simp] theorem **adele_add_apply** (a b : 𝔸ᶠ[R, K]) (v : HeightOneSpectrum R) : (a + b) v = a v + b v

@[simp] theorem **adele_sub_apply** (a b : 𝔸ᶠ[R, K]) (v : HeightOneSpectrum R) : (a - b) v = a v - b v

@[simp] theorem **adele_one_apply** (v : HeightOneSpectrum R) : (1 : 𝔸ᶠ[R, K]) v = 1

@[simp] theorem **adele_zero_apply** (v : HeightOneSpectrum R) : (0 : 𝔸ᶠ[R, K]) v = 0

theorem **adele_algebraMap_apply** (x : K) (v : HeightOneSpectrum R) : algebraMap K 𝔸ᶠ[R, K] x v = algebraMap K (v.adicCompletion K) x

theorem **adele_smul_apply** (r : R) (a : 𝔸ᶠ[R, K]) (v : HeightOneSpectrum R) : (r • a) v = algebraMap K (v.adicCompletion K) (algebraMap R K r) * a v

theorem **mem_adelicIdeal** {I : FractionalIdeal R⁰ K} {a : 𝔸ᶠ[R, K]} : a ∈ adelicIdeal I ↔ ∀ v, a v ∈ localIdeal v I

theorem **mem_adelicIdeal_iff** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) {a : 𝔸ᶠ[R, K]} : a ∈ adelicIdeal I ↔ ∀ v, Valued.v (a v) ≤ exp (-FractionalIdeal.count K v I)

theorem **algebraMap_mem_adelicIdeal** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (x : K) : algebraMap K 𝔸ᶠ[R, K] x ∈ adelicIdeal I ↔ x ∈ I

theorem **idele_apply_mul_inv** (t : 𝔸ᶠ[R, K]ˣ) (v : HeightOneSpectrum R) : (t : 𝔸ᶠ[R, K]) v * ((t⁻¹ : 𝔸ᶠ[R, K]ˣ) : 𝔸ᶠ[R, K]) v = 1

theorem **idele_apply_ne_zero** (t : 𝔸ᶠ[R, K]ˣ) (v : HeightOneSpectrum R) : (t : 𝔸ᶠ[R, K]) v ≠ 0

theorem **valued_idele** (t : 𝔸ᶠ[R, K]ˣ) (v : HeightOneSpectrum R) : Valued.v ((t : 𝔸ᶠ[R, K]) v) = exp (-ord_vw t v)

theorem **ord_vw_mul** (t s : 𝔸ᶠ[R, K]ˣ) (v : HeightOneSpectrum R) : ord_vw (t * s) v = ord_vw t v + ord_vw s v

theorem **ord_vw_one** (v : HeightOneSpectrum R) : ord_vw (1 : 𝔸ᶠ[R, K]ˣ) v = 0

theorem **ord_vw_inv** (t : 𝔸ᶠ[R, K]ˣ) (v : HeightOneSpectrum R) : ord_vw t⁻¹ v = -ord_vw t v

theorem **ord_vw_eventually_zero** (t : 𝔸ᶠ[R, K]ˣ) : ∀ᶠ v in Filter.cofinite, ord_vw t v = 0

theorem **idealOf_ne_zero** (t : 𝔸ᶠ[R, K]ˣ) : idealOf t ≠ 0

theorem **count_idealOf** (t : 𝔸ᶠ[R, K]ˣ) (v : HeightOneSpectrum R) : FractionalIdeal.count K v (idealOf t) = ord_vw t v

theorem **idealOf_one** : idealOf (1 : 𝔸ᶠ[R, K]ˣ) = 1

theorem **idealOf_mul** (t s : 𝔸ᶠ[R, K]ˣ) : idealOf (t * s) = idealOf t * idealOf s

theorem **idealOf_inv** (t : 𝔸ᶠ[R, K]ˣ) : idealOf t⁻¹ = (idealOf t)⁻¹

theorem **smul_def** (t : 𝔸ᶠ[R, K]ˣ) (I : FractionalIdeal R⁰ K) : t • I = idealOf t * I

theorem **smul_one_eq** (t : 𝔸ᶠ[R, K]ˣ) : t • (1 : FractionalIdeal R⁰ K) = idealOf t

theorem **smul_mul** (t : 𝔸ᶠ[R, K]ˣ) (I J : FractionalIdeal R⁰ K) : t • (I * J) = (t • I) * J

theorem **smul_ne_zero** (t : 𝔸ᶠ[R, K]ˣ) {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) : t • I ≠ 0

theorem **smul_le_smul** (t : 𝔸ᶠ[R, K]ˣ) {I J : FractionalIdeal R⁰ K} (h : I ≤ J) : t • I ≤ t • J

theorem **count_smul** (t : 𝔸ᶠ[R, K]ˣ) {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (v : HeightOneSpectrum R) : FractionalIdeal.count K v (t • I) = ord_vw t v + FractionalIdeal.count K v I

theorem **mul_mem_localIdeal_smul** (t : 𝔸ᶠ[R, K]ˣ) (I : FractionalIdeal R⁰ K) (v : HeightOneSpectrum R) (x : v.adicCompletion K) : (t : 𝔸ᶠ[R, K]) v * x ∈ localIdeal v (t • I) ↔ x ∈ localIdeal v I

theorem **mul_mem_adelicIdeal_smul** (t : 𝔸ᶠ[R, K]ˣ) (I : FractionalIdeal R⁰ K) (a : 𝔸ᶠ[R, K]) : (t : 𝔸ᶠ[R, K]) * a ∈ adelicIdeal (t • I) ↔ a ∈ adelicIdeal I

theorem **mem_smul_iff** (t : 𝔸ᶠ[R, K]ˣ) {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (x : K) : x ∈ t • I ↔ ∃ a ∈ adelicIdeal I, (t : 𝔸ᶠ[R, K]) * a = algebraMap K 𝔸ᶠ[R, K] x

theorem **idealOf_unitEmbedding** (x : Kˣ) : idealOf (FiniteAdeleRing.unitEmbedding R K x) = FractionalIdeal.spanSingleton R⁰ (x : K)

theorem **smul_eq_self_iff** (t : 𝔸ᶠ[R, K]ˣ) {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) : t • I = I ↔ ∀ v, Valued.v ((t : 𝔸ᶠ[R, K]) v) = 1

theorem **exists_idealOf_eq** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) : ∃ t : 𝔸ᶠ[R, K]ˣ, idealOf t = I

theorem **exists_smul_eq** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) : ∃ t : 𝔸ᶠ[R, K]ˣ, J = t • I

theorem **setOf_valued_le_eq** (v : HeightOneSpectrum R) (n : ℤ) : ∃ c : v.adicCompletion K, Valued.v.restrict c ≠ 0 ∧ {y : v.adicCompletion K | Valued.v y ≤ exp n} = {y | Valued.v.restrict y ≤ Valued.v.restrict c}

theorem **isOpen_valued_le** (v : HeightOneSpectrum R) (n : ℤ) : IsOpen {y : v.adicCompletion K | Valued.v y ≤ exp n}

theorem **isClosed_valued_le** (v : HeightOneSpectrum R) (n : ℤ) : IsClosed {y : v.adicCompletion K | Valued.v y ≤ exp n}

theorem **exists_exp_neg_lt** {γ : ℤᵐ⁰} (hγ : γ ≠ 0) : ∃ n : ℕ, exp (-(n : ℤ)) < γ

theorem **exists_limit** (v : HeightOneSpectrum R) (u : ℕ → v.adicCompletion K) (hu : ∀ m n : ℕ, n ≤ m → Valued.v (u m - u n) ≤ exp (-(n : ℤ))) : ∃ t : v.adicCompletion K, ∀ n : ℕ, Valued.v (t - u n) ≤ exp (-(n : ℤ))

theorem **exists_valued_sub_le** (v : HeightOneSpectrum R) {a : v.adicCompletion K} (ha : Valued.v a ≤ 1) (n : ℕ) : ∃ r : R, Valued.v (a - algebraMap K (v.adicCompletion K) (algebraMap R K r)) ≤ exp (-(n : ℤ))

theorem **exists_sub_mem_adelicIdeal_of_integral** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) {a : 𝔸ᶠ[R, K]} (ha : ∀ v, Valued.v (a v) ≤ 1) : ∃ r : R, a - algebraMap K 𝔸ᶠ[R, K] (algebraMap R K r) ∈ adelicIdeal I

theorem **exists_mul_integral** (a : 𝔸ᶠ[R, K]) : ∃ b : R, b ≠ 0 ∧ ∀ v, Valued.v ((algebraMap K 𝔸ᶠ[R, K] (algebraMap R K b) * a) v) ≤ 1

theorem **exists_sub_mem_adelicIdeal** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (a : 𝔸ᶠ[R, K]) : ∃ x : K, a - algebraMap K 𝔸ᶠ[R, K] x ∈ adelicIdeal I

@[simp] theorem **quotEquivAdelic_mk** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (x : K) : quotEquivAdelic hI (Submodule.Quotient.mk x) = Submodule.Quotient.mk (algebraMap K 𝔸ᶠ[R, K] x)

@[simp] theorem **adelicMulEquiv_mk** (t : 𝔸ᶠ[R, K]ˣ) {I J : FractionalIdeal R⁰ K} (h : J = t • I) (a : 𝔸ᶠ[R, K]) : adelicMulEquiv t h (Submodule.Quotient.mk a) = Submodule.Quotient.mk ((t : 𝔸ᶠ[R, K]) * a)

theorem **quotEquivAdelic_mulEquiv** (t : 𝔸ᶠ[R, K]ˣ) {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) (h : J = t • I) (q : K ⧸ (I : Submodule R K)) : quotEquivAdelic hJ (mulEquiv t hI h q) = adelicMulEquiv t h (quotEquivAdelic hI q)

theorem **mulEquiv_mk_eq_mk_iff_adelic** (t : 𝔸ᶠ[R, K]ˣ) {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (h : J = t • I) (x y : K) : mulEquiv t hI h (Submodule.Quotient.mk x) = Submodule.Quotient.mk y ↔ (t : 𝔸ᶠ[R, K]) * algebraMap K _ x - algebraMap K _ y ∈ adelicIdeal J

theorem **mulEquiv_mk_eq_mk_iff** (t : 𝔸ᶠ[R, K]ˣ) {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (h : J = t • I) (x y : K) : mulEquiv t hI h (Submodule.Quotient.mk x) = Submodule.Quotient.mk y ↔ ∀ v : HeightOneSpectrum R, (t : 𝔸ᶠ[R, K]) v * algebraMap K (v.adicCompletion K) x - algebraMap K (v.adicCompletion K) y ∈ localIdeal v J

theorem **mulEquiv_one** {I : FractionalIdeal R⁰ K} (hI : I ≠ 0) (h : I = (1 : 𝔸ᶠ[R, K]ˣ) • I) : mulEquiv 1 hI h = LinearEquiv.refl R _

theorem **mulEquiv_mul** (t s : 𝔸ᶠ[R, K]ˣ) {I J L : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) (h : J = s • I) (h' : L = t • J) (h'' : L = (t * s) • I) : mulEquiv (t * s) hI h'' = (mulEquiv s hI h).trans (mulEquiv t hJ h')

theorem **mulEquiv_unitEmbedding_mk** (c : Kˣ) {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (h : J = FiniteAdeleRing.unitEmbedding R K c • I) (x : K) : mulEquiv (FiniteAdeleRing.unitEmbedding R K c) hI h (Submodule.Quotient.mk x) = Submodule.Quotient.mk ((c : K) * x)

theorem **mulEquiv_mk_mul** (t : 𝔸ᶠ[R, K]ˣ) {I J I' J' : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hI' : I' ≠ 0) (h : J = t • I) (h' : J' = t • I') {c : K} (hc : ∀ z ∈ I, c * z ∈ I') {x y : K} (hxy : mulEquiv t hI h (Submodule.Quotient.mk x) = Submodule.Quotient.mk y) : mulEquiv t hI' h' (Submodule.Quotient.mk (c * x)) = Submodule.Quotient.mk (c * y)

theorem **mulEquiv_mem_iff** (t : 𝔸ᶠ[R, K]ˣ) {I J I' : FractionalIdeal R⁰ K} (hI : I ≠ 0) (h : J = t • I) (hle : I ≤ I') {x y : K} (hxy : mulEquiv t hI h (Submodule.Quotient.mk x) = Submodule.Quotient.mk y) : y ∈ t • I' ↔ x ∈ I'

theorem **mem_adelicIdeal_one_iff** {a : 𝔸ᶠ[R, K]} : a ∈ adelicIdeal (1 : FractionalIdeal R⁰ K) ↔ ∀ v, Valued.v (a v) ≤ 1

theorem **exists_adele_of_end_one** (ψ : (𝔸ᶠ[R, K] ⧸ adelicIdeal (1 : FractionalIdeal R⁰ K)) →ₗ[R] (𝔸ᶠ[R, K] ⧸ adelicIdeal (1 : FractionalIdeal R⁰ K))) : ∃ t : 𝔸ᶠ[R, K], ∀ a, ψ (Submodule.Quotient.mk a) = Submodule.Quotient.mk (t * a)

theorem **exists_adele_of_linearMap** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) (ψ : (𝔸ᶠ[R, K] ⧸ adelicIdeal I) →ₗ[R] (𝔸ᶠ[R, K] ⧸ adelicIdeal J)) : ∃ t : 𝔸ᶠ[R, K], ∀ a, ψ (Submodule.Quotient.mk a) = Submodule.Quotient.mk (t * a)

theorem **eq_zero_of_forall_mul_mem** {J : FractionalIdeal R⁰ K} (hJ : J ≠ 0) {t : 𝔸ᶠ[R, K]} (h : ∀ x : K, t * algebraMap K 𝔸ᶠ[R, K] x ∈ adelicIdeal J) : t = 0

theorem **existsUnique_adele_of_linearMap** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (hJ : J ≠ 0) (φ : (K ⧸ (I : Submodule R K)) →ₗ[R] (K ⧸ (J : Submodule R K))) : ∃! t : 𝔸ᶠ[R, K], ∀ x : K, quotEquivAdelic hJ (φ (Submodule.Quotient.mk x)) = Submodule.Quotient.mk (t * algebraMap K 𝔸ᶠ[R, K] x)

theorem **mulEquiv_injective** {t s : 𝔸ᶠ[R, K]ˣ} {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (h : J = t • I) (h' : J = s • I) (e : mulEquiv t hI h = mulEquiv s hI h') : t = s

theorem **ne_zero_of_linearEquiv** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (φ : (K ⧸ (I : Submodule R K)) ≃ₗ[R] (K ⧸ (J : Submodule R K))) : J ≠ 0

theorem **existsUnique_mulEquiv** {I J : FractionalIdeal R⁰ K} (hI : I ≠ 0) (φ : (K ⧸ (I : Submodule R K)) ≃ₗ[R] (K ⧸ (J : Submodule R K))) : ∃! t : 𝔸ᶠ[R, K]ˣ, ∃ h : J = t • I, φ = mulEquiv t hI h

theorem **existsUnique_idele_of_linearEquiv_quotient** {K : Type*} [Field K] [NumberField K] (𝔞 : FractionalIdeal (𝒪 K)⁰ K) (φ : (K ⧸ ((1 : FractionalIdeal (𝒪 K)⁰ K) : Submodule (𝒪 K) K)) →ₗ[𝒪 K] (K ⧸ (𝔞 : Submodule (𝒪 K) K))) (hφ : Function.Bijective φ) : ∃! t : (FiniteAdeleRing (𝒪 K) K)ˣ, ∃ h : 𝔞 = t⁻¹ • (1 : FractionalIdeal (𝒪 K)⁰ K), ∀ q, φ q = mulEquiv t⁻¹ one_ne_zero h q

theorem **existsUnique_idele_of_linearEquiv_quotient_local** {K : Type*} [Field K] [NumberField K] (𝔞 : FractionalIdeal (𝒪 K)⁰ K) (φ : (K ⧸ ((1 : FractionalIdeal (𝒪 K)⁰ K) : Submodule (𝒪 K) K)) →ₗ[𝒪 K] (K ⧸ (𝔞 : Submodule (𝒪 K) K))) (hφ : Function.Bijective φ) : ∃! t : (FiniteAdeleRing (𝒪 K) K)ˣ, (∀ x : K, x ∈ 𝔞 ↔ ∀ v, (t : FiniteAdeleRing (𝒪 K) K) v * algebraMap K (v.adicCompletion K) x ∈ v.adicCompletionIntegers K) ∧ ∀ x y : K, φ (Submodule.Quotient.mk x) = Submodule.Quotient.mk y ↔ ∀ v, ((t⁻¹ : (FiniteAdeleRing (𝒪 K) K)ˣ) : FiniteAdeleRing (𝒪 K) K) v * algebraMap K (v.adicCompletion K) x - algebraMap K (v.adicCompletion K) y ∈ localIdeal v 𝔞

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitEvenOrderQuadraticNormValuesOneModFour

theorem **exists_not_mem_sq_add_sub** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : ∃ ρ : 𝒪 F, ∀ a : 𝒪 F, a ^ 2 + a - ρ ∉ v.asIdeal

theorem **exists_evenNorms_unit** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), ∃ Δ : v.adicCompletion F, EvenNorms Valued.v Δ ∧ Valued.v Δ = 1 ∧ Valued.v (Δ - 1) ≤ Valued.v (4 : v.adicCompletion F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitMulSqQuadraticDefectStep

theorem **exists_unit_mul_sq_sub_one_lt** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (δ π : K) (k : ℕ), w π = WithZero.exp (-1) → (∀ x : K, w x ≤ 1 → ∃ y : K, w y ≤ 1 ∧ w (x - y ^ 2) < 1) → w δ = 1 → w 4 < w (δ - 1) → w (δ - 1) = WithZero.exp (-(2 * (k : ℤ))) → ∃ c : K, w c = 1 ∧ w (δ * c ^ 2 - 1) < w (δ - 1)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitMulSqQuadraticDefectStepStatement

theorem **_root_.GlobalTheory.T.exists_unit_mul_sq_quadratic_defect_step** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (δ π : K) (k : ℕ), w π = WithZero.exp (-1) → (∀ x : K, w x ≤ 1 → ∃ y : K, w y ≤ 1 ∧ w (x - y ^ 2) < 1) → w δ = 1 → w 4 < w (δ - 1) → w (δ - 1) = WithZero.exp (-(2 * (k : ℤ))) → ∃ c : K, w c = 1 ∧ w (δ * c ^ 2 - 1) < w (δ - 1)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitNormEqOfAlgHom

theorem **exists_unit_norm_eq_of_algHom** : ∀ (K A : Type) [Field K] [CommRing A] [Algebra K A] [Module.Finite K A] [IsReduced A] (ψ : A →ₐ[K] K) (c : K), c ≠ 0 → ∃ y : Aˣ, Algebra.norm K (y : A) = c

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitNotLocalKummerNormOfNotPow

theorem **nonnorm_pIdx_map** (p : ℕ) {G H : Type} [CommGroup G] [CommGroup H] (f : G →* H) (hf : Function.Injective f) (U : Subgroup G) : pIdx p (U.map f) = pIdx p U

theorem **nonnorm_pTor_map** (p : ℕ) {G H : Type} [CommGroup G] [CommGroup H] (f : G →* H) (hf : Function.Injective f) (U : Subgroup G) : pTor p (U.map f) = pTor p U

theorem **exists_local_nonnorm** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (ζ δ : F) (v : HeightOneSpectrum (𝒪 F)), p.Prime → IsPrimitiveRoot ζ p → (∀ r : v.adicCompletion F, r ^ p ≠ algebraMap F (v.adicCompletion F) δ) → ∃ c : (v.adicCompletion F)ˣ, c ∉ kNorm p (RingHom.id (v.adicCompletion F)) (algebraMap F (v.adicCompletion F) δ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitNotLocalKummerNormOfNotPowStatement

theorem **_root_.GlobalTheory.T.exists_unit_not_localKummerNorm_of_not_pow** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (ζ δ : F) (v : HeightOneSpectrum (𝒪 F)), p.Prime → IsPrimitiveRoot ζ p → (∀ r : v.adicCompletion F, r ^ p ≠ algebraMap F (v.adicCompletion F) δ) → ∃ c : (v.adicCompletion F)ˣ, c ∉ kNorm p (RingHom.id (v.adicCompletion F)) (algebraMap F (v.adicCompletion F) δ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitValuationEqOfAdm

theorem **exists_unit_valuation_eq_of_adm** : ∀ (F : Type) [Field F] [NumberField F] (S T : Finset (HeightOneSpectrum (𝒪 F))) (n : HeightOneSpectrum (𝒪 F) → ℤ), admitted F S → (∀ v, v ∉ T → n v = 0) → ∃ a : Fˣ, ∀ v, v ∉ S → v.valuation F (a : F) = WithZero.exp (n v)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitValuedTernaryLtOne

theorem **exists_ne_zero_sum_eq_zero** {k : Type*} [Field k] [Finite k] (a : Fin 3 → Fin 3 → k) : ∃ x : Fin 3 → k, x ≠ 0 ∧ ∑ i, ∑ j, a i j * x i * x j = 0

theorem **exists_unit_valued_ternary_lt_one** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (c : Fin 3 → Fin 3 → v.adicCompletion F) (hc : ∀ i j, Valued.v (c i j) ≤ 1) : ∃ x : Fin 3 → v.adicCompletion F, (∀ i, Valued.v (x i) ≤ 1) ∧ (∃ i, Valued.v (x i) = 1) ∧ Valued.v (∑ i, ∑ j, c i j * x i * x j) < 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitaryAdelicDecompositionSplitSupport

theorem **mix_val** {R : Type} [CommRing R] {n : ℕ} (ε : R) (hε : ε * ε = ε) (κ : GL (Fin n) R) : (mix ε hε κ).1 = ε • κ.1 + (1 − ε) • (1 : Matrix (Fin n) (Fin n) R)

theorem **mix_inv_val** {R : Type} [CommRing R] {n : ℕ} (ε : R) (hε : ε * ε = ε) (κ : GL (Fin n) R) : ((mix ε hε κ)⁻¹).1 = ε • (κ⁻¹).1 + (1 − ε) • (1 : Matrix (Fin n) (Fin n) R)

theorem **_root_.GlobalTheory.UnRep.SplitRep.mix_mul_mix** {R : Type} [CommRing R] {n : ℕ} (ε : R) (hε : ε * ε = ε) (hε' : (1 − ε) * (1 − ε) = 1 − ε) (κ : GL (Fin n) R) : (_root_.GlobalTheory.UnRep.SplitRep.mix ε hε κ).1 * (_root_.GlobalTheory.UnRep.SplitRep.mix (1 − ε) hε' κ).1 = κ.1

theorem **_root_.GlobalTheory.UnRep.SplitRep.isUnA_mix** (c : E ≃ₐ[F] E) (G : Matrix (Fin n) (Fin n) E) (ε : _root_.GlobalTheory.AdLat.RA F E) (hε : ε * ε = ε) (hs : _root_.GlobalTheory.AdLat.sg (FiniteAdeleRing (𝒪 F) F) c ε = ε) (κ : GL (Fin n) (_root_.GlobalTheory.AdLat.RA F E)) (hκ : _root_.GlobalTheory.UnRep.IsUnA c G κ.1) : _root_.GlobalTheory.UnRep.IsUnA c G (_root_.GlobalTheory.UnRep.SplitRep.mix ε hε κ).1

theorem **hermA_eq** (c : E ≃ₐ[F] E) (G : Matrix (Fin n) (Fin n) E) (x y : Fin n → RA F E) : hermA c G x y = x ⬝ᵥ (fm F G *ᵥ fun b => sg (FiniteAdeleRing (𝒪 F) F) c (y b))

theorem **_root_.GlobalTheory.UnRep.SplitRep.exists_multiple** (s : 𝒪 E) (hs : s ≠ 0) : ∃ m : 𝒪 F, m ≠ 0 ∧ s ∣ algebraMap (𝒪 F) (𝒪 E) m ∧ ∀ v : HeightOneSpectrum (𝒪 F), m ∈ v.asIdeal → ∃ P : Ideal (𝒪 E), P.IsPrime ∧ s ∈ P ∧ P.comap (algebraMap (𝒪 F) (𝒪 E)) = v.asIdeal

theorem **_root_.GlobalTheory.UnRep.SplitRep.exists_div** (m : 𝒪 F) (hm : m ≠ 0) (e : FiniteAdeleRing (𝒪 F) F) (h1 : ∀ v : HeightOneSpectrum (𝒪 F), m ∈ v.asIdeal → e v = 1) (h0 : ∀ v : HeightOneSpectrum (𝒪 F), m ∉ v.asIdeal → e v = 0) : ∃ a : FiniteAdeleRing (𝒪 F) F, (∀ v, a v ∈ v.adicCompletionIntegers F) ∧ (m : F) • a = 1 − e

private theorem **galoisAut_one_tmul_eq_one_tmul** (c : E ≃ₐ[F] E) (z : E) : _root_.GlobalTheory.AdLat.sg (FiniteAdeleRing (𝒪 F) F) c ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] z) = 1 ⊗ₜ[F] c z

private theorem **isUnA_fm_of_isUnE** (c : E ≃ₐ[F] E) (G γ : Matrix (Fin n) (Fin n) E) (hγ : _root_.GlobalTheory.UnRep.IsUnE c G γ) : _root_.GlobalTheory.UnRep.IsUnA c G (_root_.GlobalTheory.UnRep.fm F γ)

theorem **_root_.GlobalTheory.UnRep.exists_rep_suppSplit** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), _root_.GlobalTheory.UnRep.IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (_root_.GlobalTheory.AdLat.RA F E)) (N : ℕ), _root_.GlobalTheory.UnRep.IsUnA c G g.1 → 0 < N → ∃ (γ : GL (Fin n) E) (h k : GL (Fin n) (_root_.GlobalTheory.AdLat.RA F E)), _root_.GlobalTheory.UnRep.IsUnE c G γ.1 ∧ _root_.GlobalTheory.UnRep.IsUnA c G h.1 ∧ _root_.GlobalTheory.UnRep.IsUnA c G k.1 ∧ g.1 = _root_.GlobalTheory.UnRep.fm F γ.1 * h.1 * k.1 ∧ _root_.GlobalTheory.UnRep.SuppSplit c N h.1 ∧ (∀ z ∈ _root_.GlobalTheory.AdLat.hat F (L : Set (Fin n → E)), ∃ w ∈ _root_.GlobalTheory.AdLat.hat F (L : Set (Fin n → E)), k.1 *ᵥ z − z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitaryAdelicDecompositionSplitSupportStatement

theorem **_root_.GlobalTheory.T.exists_unitary_adelic_decomposition_split_support** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (RA F E)) (N : ℕ), IsUnA c G g.1 → 0 < N → ∃ (γ : GL (Fin n) E) (h k : GL (Fin n) (RA F E)), IsUnE c G γ.1 ∧ IsUnA c G h.1 ∧ IsUnA c G k.1 ∧ g.1 = fm F γ.1 * h.1 * k.1 ∧ SuppSplit c N h.1 ∧ (∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), k.1 *ᵥ z − z = N • w) ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), (k⁻¹).1 *ᵥ z ∈ hat F (L : Set (Fin n → E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitaryCongruentLattice

theorem **fv_mulVec** (γ : Matrix (Fin n) (Fin n) E) (v : Fin n → E) : fv F (γ *ᵥ v) = fm F γ *ᵥ fv F v

theorem **fv_smul** (s : E) (v : Fin n → E) : fv F (s • v) = ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] s : RA F E) • fv F v

theorem **fv_sub** (x y : Fin n → E) : fv F (x − y) = fv F x − fv F y

theorem **nsmul_cancel** {M : ℕ} (hM : 0 < M) {x y : Fin n → RA F E} (h : M • x = M • y) : x = y

theorem **span_smul_le** (s : 𝒪 E) (A B : Submodule (𝒪 E) (Fin n → E)) : Ideal.span {s} • A ≤ B ↔ ∀ v ∈ A, s • v ∈ B

theorem **exists_nat_coprime** (s : 𝒪 E) (N : ℕ) (h : ∀ P : Ideal (𝒪 E), P.IsPrime → s ∈ P → (N : 𝒪 E) ∉ P) : ∃ k : ℕ, k.Coprime N ∧ s ∣ (k : 𝒪 E)

theorem **smul_mem_of_dvd** {s t : 𝒪 E} (h : s ∣ t) {B : Submodule (𝒪 E) (Fin n → E)} {v : Fin n → E} (hv : s • v ∈ B) : t • v ∈ B

theorem **hermE_one** (x y : Fin n → E) : hermE (IsCMField.complexConj E) 1 x y = ∑ j, IsCMField.complexConj E (y j) * x j

theorem **unimod_move** (γ : GL (Fin n) E) (hγ : IsUnE (IsCMField.complexConj E) 1 γ.1) (Λ Λ' : GenLat E n) (h : ∀ v, v ∈ Λ.La ↔ γ.1 *ᵥ v ∈ Λ'.La) (hU : Λ.Unimod) : Λ'.Unimod

theorem **exists_move** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (n M bad : ℕ) (Λ : GenLat E n), 0 < M → 0 < bad → Λ.Unimod → Λ.TrivAt M → ∃ (γ : GL (Fin n) E) (Λ' : GenLat E n), (∀ i j, ∑ b, (IsCMField.complexConj E) (γ.1 b i) * γ.1 b j = if i = j then 1 else 0) ∧ CongAt (stdLat E n) M M γ.1 ∧ (∀ v, v ∈ Λ.La ↔ γ.1 *ᵥ v ∈ Λ'.La) ∧ Λ'.Unimod ∧ Λ'.TrivAt bad ∧ Λ'.TrivAt M ∧ GenLat.SplitSupp H Λ' ∧ (Odd M → 2 ∣ bad → ∃ N, Λ'.LevOK N M)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitaryCorrectionCongruentOneAtLevel

theorem **exists_nat_smul_mem** [NumberField E] (L : Submodule (𝒪 E) (Fin n → E)) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (x : Fin n → E) : ∃ d : ℕ, 0 < d ∧ d • x ∈ L

theorem **exists_nat_mulVec_mem** [NumberField E] (L : Submodule (𝒪 E) (Fin n → E)) (hfg : L.FG) (hL : Submodule.span E (L : Set (Fin n → E)) = ⊤) (M : Matrix (Fin n) (Fin n) E) : ∃ d : ℕ, 0 < d ∧ ∀ v ∈ L, d • (M *ᵥ v) ∈ L

theorem **congAt_mul** {L : Submodule (𝒪 E) (Fin n → E)} {N D : ℕ} {A B : Matrix (Fin n) (Fin n) E} (hA : CongAt L N D A) (hB : CongAt L N D B) : CongAt L N D (A * B)

theorem **congAt_conj** [CharZero E] {L : Submodule (𝒪 E) (Fin n → E)} {N D d e : ℕ} (he : 0 < e) {A R R' : Matrix (Fin n) (Fin n) E} (hRR : R * R' = 1) (hR : ∀ v ∈ L, d • (R *ᵥ v) ∈ L) (hR' : ∀ v ∈ L, e • (R' *ᵥ v) ∈ L) (hA : CongAt L N (D * d * e) A) : CongAt L N D (R * A * R')

theorem **splitOr_mul** {F : Type} [Field F] [Algebra F E] {c : E ≃ₐ[F] E} {N : ℕ} {s t : 𝒪 E} (hs : SplitOr c N s) (ht : SplitOr c N t) : SplitOr c N (s * t)

theorem **smul_mul_mem** {L : Submodule (𝒪 E) (Fin n → E)} {s t : 𝒪 E} {A B : Matrix (Fin n) (Fin n) E} (hA : ∀ v ∈ L, s • (A *ᵥ v) ∈ L) (hB : ∀ v ∈ L, t • (B *ᵥ v) ∈ L) (v : Fin n → E) (hv : v ∈ L) : (s * t) • ((A * B) *ᵥ v) ∈ L

theorem **exists_unitary_correction** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (γ₀ : GL (Fin n) E) (N D : ℕ), IsUnE c G γ₀.1 → 0 < N → 0 < D → ∃ γ : GL (Fin n) E, IsUnE c G γ.1 ∧ CongAt L N D (γ * γ₀⁻¹).1 ∧ ∃ s : 𝒪 E, SplitOr c N s ∧ ∀ v ∈ L, s • (γ.1 *ᵥ v) ∈ L ∧ s • ((γ⁻¹).1 *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsUnitaryCorrectionCongruentOneAtLevelStatement

theorem **_root_.GlobalTheory.T.exists_unitary_correction_congruent_one_at_level** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (γ₀ : GL (Fin n) E) (N D : ℕ), IsUnE c G γ₀.1 → 0 < N → 0 < D → ∃ γ : GL (Fin n) E, IsUnE c G γ.1 ∧ CongAt L N D (γ * γ₀⁻¹).1 ∧ ∃ s : 𝒪 E, SplitOr c N s ∧ ∀ v ∈ L, s • (γ.1 *ᵥ v) ∈ L ∧ s • ((γ⁻¹).1 *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsZeroCoordNeZero

theorem **Congr.refl** (G : Matrix n n E) : Congr c G G

theorem **Congr.trans** {G G' G'' : Matrix n n E} (h : Congr c G G') (h' : Congr c G' G'') : Congr c G G''

theorem **Congr.symm** {G G' : Matrix n n E} (h : Congr c G G') : Congr c G' G

theorem **Congr.det** {G G' : Matrix n n E} (h : Congr c G G') : ∃ x : E, x ≠ 0 ∧ G'.det = x * c x * G.det

theorem **Congr.submatrix** {G G' : Matrix n n E} (h : Congr c G G') (e : m ≃ n) : Congr c (G.submatrix e e) (G'.submatrix e e)

theorem **Congr.fromBlocks** {G G' : Matrix n n E} (h : Congr c G G') (D : Matrix m m E) : Congr c (Matrix.fromBlocks G 0 0 D) (Matrix.fromBlocks G' 0 0 D)

theorem **congr_diagonal_of_tail** {r : ℕ} {g g' : Fin (r + 1) → E} (h0 : g 0 = g' 0) (h : Congr c (Matrix.diagonal fun i : Fin r => g i.succ) (Matrix.diagonal fun i : Fin r => g' i.succ)) : Congr c (Matrix.diagonal g) (Matrix.diagonal g')

theorem **exists_zero_coord_ne_zero** {F : Type*} [Field F] [CharZero F] {ι : Type*} [Fintype ι] {a : ι → F} (ha : ∀ i, a i ≠ 0) {z : ι → F} (hz : z ≠ 0) (h : ∑ i, a i * z i ^ 2 = 0) (i : ι) : ∃ w : ι → F, w i ≠ 0 ∧ ∑ k, a k * w k ^ 2 = 0

theorem **mem_real** {x : E} (hx : cc x = x) : x ∈ maximalRealSubfield E

theorem **re_ne_zero** (τ : E →+* ℂ) {x : E} (hx : cc x = x) (h0 : x ≠ 0) : (τ x).re ≠ 0

theorem **exists_represent** (H : Isotropy) {r : ℕ} (hr : 2 ≤ r) {g : Fin r → E} (hg : ∀ i, cc (g i) = g i) (hg0 : ∀ i, g i ≠ 0) {c : E} (hc : cc c = c) (hc0 : c ≠ 0) (hs : ∀ τ : E →+* ℂ, ∃ i, (0 < (τ (g i)).re ↔ 0 < (τ c).re)) : ∃ z : Fin r → E, herm (Matrix.diagonal g) z z = c

theorem **posCount_succ** (τ : E →+* ℂ) {r : ℕ} (g : Fin (r + 1) → E) : posCount τ g = (if 0 < (τ (g 0)).re then 1 else 0) + posCount τ fun i : Fin r => g i.succ

theorem **posIndex_diagonal_map** (τ : E →+* ℂ) {r : ℕ} (g : Fin r → E) : GlobalTheory.Inertia.posIndex ((Matrix.diagonal g).map τ) = posCount τ g

theorem **posIndex_of_congr** (τ : E →+* ℂ) {r : ℕ} {G G' : Matrix (Fin r) (Fin r) E} (h : Congr (IsCMField.complexConj E : E →+* E) G G') : GlobalTheory.Inertia.posIndex (G'.map τ) = GlobalTheory.Inertia.posIndex (G.map τ)

theorem **exists_sign_iff** (τ : E →+* ℂ) {r : ℕ} {g g' : Fin r → E} (h : posCount τ g = posCount τ g') (j : Fin r) : ∃ i, (0 < (τ (g i)).re ↔ 0 < (τ (g' j)).re)

theorem **congr_diagonal** (H : Isotropy) (r : ℕ) : ∀ g g' : Fin r → E, (∀ i, cc (g i) = g i) → (∀ i, cc (g' i) = g' i) → (∀ i, g i ≠ 0) → (∀ i, g' i ≠ 0) → (∀ τ : E →+* ℂ, posCount τ g = posCount τ g') → (∃ x : E, ∏ i, g i = x * cc x * ∏ i, g' i) → Congr (IsCMField.complexConj E : E →+* E) (Matrix.diagonal g) (Matrix.diagonal g')

theorem **landherr_of_isotropy** (H : ∀ (F : Type) [Field F] [NumberField F] (ι : Type) [Fintype ι] (a : ι → F), 5 ≤ Fintype.card ι → (∀ i, a i ≠ 0) → (∀ σ : F →+* ℝ, ∃ i j, σ (a i) < 0 ∧ 0 < σ (a j)) → ∃ x : ι → F, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0) : GlobalTheory.Published.Landherr

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExistsZeroFiveOfValuation

lemma **qf_add** (a x y : n → K) : qf a (x + y) = qf a x + bf a x y + qf a y

lemma **qf_smul** (a x : n → K) (s : K) : qf a (s • x) = s ^ 2 * qf a x

lemma **bf_smul_left** (a x y : n → K) (s : K) : bf a (s • x) y = s * bf a x y

lemma **bf_smul_right** (a x y : n → K) (s : K) : bf a x (s • y) = s * bf a x y

lemma **bf_add_left** (a x y z : n → K) : bf a (x + y) z = bf a x z + bf a y z

lemma **bf_self** (a x : n → K) : bf a x x = 2 * qf a x

lemma **qf_three** (a p q r : n → K) (x : Fin 3 → K) (w : K) : qf a (x 0 • p + x 1 • q + x 2 • r) / w = ∑ i, ∑ j, (![![qf a p / w, bf a p q / w, bf a p r / w], ![0, qf a q / w, bf a q r / w], ![0, 0, qf a r / w]] : Fin 3 → Fin 3 → K) i j * x i * x j

lemma **val_bf_sq_le** (h2 : (2 : K) ≠ 0) (v : Valuation K ℤᵐ⁰) (hsq : ∀ x : K, v (x − 1) < v 4 → IsSquare x) (a : n → K) (han : ∀ x, qf a x = 0 → x = 0) (x y : n → K) : v (bf a x y) ^ 2 ≤ v (qf a x) * v (qf a y)

lemma **val_det_le** (h2 : (2 : K) ≠ 0) (v : Valuation K ℤᵐ⁰) (hsq : ∀ x : K, v (x - 1) < v 4 → IsSquare x) (a : n → K) (han : ∀ x, qf a x = 0 → x = 0) (P : Matrix n n K) : v (∏ i, 2 * a i) * v P.det ^ 2 ≤ ∏ i, v (qf a (P i))

lemma **exists_basis** (h2 : (2 : K) ≠ 0) (v : Valuation K ℤᵐ⁰) (hsq : ∀ x : K, v (x - 1) < v 4 → IsSquare x) (a : n → K) (han : ∀ x, qf a x = 0 → x = 0) : ∃ P : Matrix n n K, (∀ j, qf a (P j) ≠ 0) ∧ ∀ (c : n → K) (j : n), v (c j) ^ 2 * v (qf a (P j)) ≤ v (qf a (∑ k, c k • P k))

lemma **exists_three** (f : Fin 5 → Bool) : ∃ i j k : Fin 5, i ≠ j ∧ i ≠ k ∧ j ≠ k ∧ f j = f i ∧ f k = f i

lemma **exists_scale** (v : Valuation K ℤᵐ⁰) (hv : Function.Surjective v) {x y : K} (hx : x ≠ 0) (hy : y ≠ 0) (h : decide (Even (WithZero.log (v y))) = decide (Even (WithZero.log (v x)))) : ∃ u : K, v u ^ 2 * v y = v x

theorem **exists_zero_five_of_valuation** {K : Type*} [Field K] (h2 : (2 : K) ≠ 0) (v : Valuation K ℤᵐ⁰) (hv : Function.Surjective v) (hsq : ∀ x : K, v (x - 1) < v 4 → IsSquare x) (hcw : ∀ c : Fin 3 → Fin 3 → K, (∀ i j, v (c i j) ≤ 1) → ∃ x : Fin 3 → K, (∀ i, v (x i) ≤ 1) ∧ (∃ i, v (x i) = 1) ∧ v (∑ i, ∑ j, c i j * x i * x j) < 1) (a : Fin 5 → K) : ∃ x : Fin 5 → K, x ≠ 0 ∧ ∑ i, a i * x i ^ 2 = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ExpSeriesContDiffAndHasDerivAt

private lemma **norm_smulRight_le_opNorm_mul_norm** {V E : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [NormedAddCommGroup E] [NormedSpace ℂ E] (L : V →L[ℝ] ℂ) (w : E) : ‖L.smulRight w‖ ≤ ‖L‖ * ‖w‖

private lemma **mul_exp_mul_le_exp_add_one_mul** (r R : ℝ) : r * Real.exp (R * r) ≤ Real.exp ((R + 1) * r)

private lemma **summable_norm_smulRight_mul_exp** {V E I : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [NormedAddCommGroup E] [NormedSpace ℂ E] (m : I → V →L[ℝ] ℂ) (w : I → E) (hw : ∀ R : ℝ, Summable fun i => ‖w i‖ * Real.exp (R * ‖m i‖)) (R : ℝ) : Summable fun i => ‖(m i).smulRight (w i)‖ * Real.exp (R * ‖m i‖)

private lemma **norm_cexp_le_exp_mul_opNorm_of_norm_le** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] (L : V →L[ℝ] ℂ) {y : V} {r : ℝ} (hy : ‖y‖ ≤ r) : ‖Complex.exp (L y)‖ ≤ Real.exp (r * ‖L‖)

private lemma **hasFDerivAt_tsum_cexp_smul_of_summable_exp** {V E I : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [NormedAddCommGroup E] [NormedSpace ℂ E] [CompleteSpace E] (m : I → V →L[ℝ] ℂ) (w : I → E) (hw : ∀ R : ℝ, Summable fun i => ‖w i‖ * Real.exp (R * ‖m i‖)) (x : V) : HasFDerivAt (fun v => ∑' i, Complex.exp (m i v) • w i) (∑' i, Complex.exp (m i x) • (m i).smulRight (w i)) x

private lemma **contDiff_tsum_cexp_smul_of_summable_exp** {V I : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] (m : I → V →L[ℝ] ℂ) (n : ℕ) : ∀ (E : Type) [NormedAddCommGroup E] [NormedSpace ℂ E] [CompleteSpace E] (w : I → E), (∀ R : ℝ, Summable fun i => ‖w i‖ * Real.exp (R * ‖m i‖)) → ContDiff ℝ n (fun v => ∑' i, Complex.exp (m i v) • w i)

theorem **_root_.GlobalTheory.expSeries_contDiff_and_hasDerivAt** : ∀ (V : Type) [NormedAddCommGroup V] [NormedSpace ℝ V] (I : Type) (c : I → ℂ) (m : I → V →L[ℝ] ℂ), (∀ R : ℝ, Summable fun i => ‖c i‖ * Real.exp (R * ‖m i‖)) → ContDiff ℝ ∞ (fun v : V => ∑' i, c i * Complex.exp (m i v)) ∧ ∀ z : V, Summable (fun i => c i * m i z) ∧ HasDerivAt (fun t : ℝ => ∑' i, c i * Complex.exp (m i (t • z))) (∑' i, c i * m i z) 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FinTensMemLatO

theorem **finTens_add** {m n : ℕ} (α : Af D.host.F) (M N : Matrix (Fin m) (Fin n) D.host.E) : finTens α (M + N) = finTens α M + finTens α N

theorem **finTens_zero** {m n : ℕ} (α : Af D.host.F) : finTens α (0 : Matrix (Fin m) (Fin n) D.host.E) = 0

theorem **finTens_neg** {m n : ℕ} (α : Af D.host.F) (M : Matrix (Fin m) (Fin n) D.host.E) : finTens α (-M) = -finTens α M

theorem **smul_mem_latRat** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} (c : 𝒪 D.host.E) {M : Matrix (Fin T.rank) (Fin S.rank) D.host.E} (hM : M ∈ latRat S T LS LT) : (c : D.host.E) • M ∈ latRat S T LS LT

theorem **finTens_mem_latO** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {α : Af D.host.F} (hα : α ∈ intAf D.host.F) {M : Matrix (Fin T.rank) (Fin S.rank) D.host.E} (hM : M ∈ latRat S T LS LT) : finTens α M ∈ latO S T LS LT fun _ => True

theorem **latO_induction** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {p : WO S T → Prop} (mem : ∀ α ∈ intAf D.host.F, ∀ M ∈ latRat S T LS LT, p (finTens α M)) (zero : p 0) (add : ∀ x y, p x → p y → p (x + y)) (neg : ∀ x, p x → p (-x)) {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) : p u

theorem **gram_star** (S : HSp D) (i j : Fin S.rank) : star (S.gram i j) = S.gram j i

theorem **gram_det_isUnit** (S : HSp D) : IsUnit S.gram.det

theorem **hRat_genMap** (S T : HSp D) (x : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (a : D.host.E) (s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) : hRat S T x (genMap S T a s t) = star a * ∑ c, ∑ d, star (t c) * T.gram c d * (x *ᵥ s) d

theorem **hO_finTens** (S T : HSp D) (α β : Af D.host.F) (M N : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hO S T (finTens α M) (finTens β N) = (pairA D.host.F 0 (α * β)) ⊗ₜ[D.host.F] hRat S T M N

theorem **trace_tmul** {F E : Type} [Field F] [Field E] [Algebra F E] [FiniteDimensional F E] (R : Type) [CommRing R] [Algebra F R] (a : R) (e : E) : Algebra.trace R (R ⊗[F] E) (a ⊗ₜ[F] e) = a * algebraMap F R (Algebra.trace F E e)

theorem **symplO_finTens** (S T : HSp D) (α β : Af D.host.F) (M N : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : symplO S T (finTens α M) (finTens β N) = pairA D.host.F 0 (α * β * algebraMap D.host.F (Af D.host.F) (Algebra.trace D.host.F D.host.E (D.host.xi0 * hRat S T M N)))

theorem **hO_add_left** (S T : HSp D) (x x' y : WO S T) : hO S T (x + x') y = hO S T x y + hO S T x' y

theorem **hO_add_right** (S T : HSp D) (x y y' : WO S T) : hO S T x (y + y') = hO S T x y + hO S T x y'

theorem **symplO_add_left** (S T : HSp D) (x x' y : WO S T) : symplO S T (x + x') y = symplO S T x y + symplO S T x' y

theorem **symplO_add_right** (S T : HSp D) (x y y' : WO S T) : symplO S T x (y + y') = symplO S T x y + symplO S T x y'

theorem **symplHom_apply** (S T : HSp D) (x y : WO S T) : symplHom S T x y = symplO S T x y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteAdelicDoubleCosetsOfRationalSLSubgroup

theorem **cf_congr_of_vanI_eq** {ι : Type} [Fintype ι] [DecidableEq ι] {Γ Δ : Set (Matrix ι ι ℂ)} (h : vanI Γ = vanI Δ) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R) : CF Γ R R₀ ↔ CF Δ R R₀

theorem **cf_of_isQGroup** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), IsQGroup Γ → AdelicPair R R₀ → CF Γ R R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteAdelicDoubleCosetsOfRationalSLSubgroupStatement

theorem **_root_.GlobalTheory.T.finiteAdelicDoubleCosets_of_rational_SL_subgroup** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), IsQGroup Γ → AdelicPair R R₀ → CF Γ R R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteIntegralClassesClosedOrbit

theorem **finite_bdd_denominator_vectors** {κ : Type} [Fintype κ] (D : ℕ) (hD : 0 < D) (B : ℝ) : {w : κ → ℚ | (∀ k, ∃ z : ℤ, (D : ℚ) * w k = z) ∧ nsq (fun k => ((w k : ℚ) : ℂ)) ≤ B}.Finite

theorem **finite_integral_classes_closed_orbit** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), IsClosed (orb d (fun k => ((t k : ℚ) : ℂ))) → ∀ D : ℕ, 0 < D → ∃ Z : Finset (TIdx ι d → ℚ), ∀ h : Matrix ι ι ℚ, h.det = 1 → (∀ k, ∃ z : ℤ, (D : ℚ) * act d t h k = z) → ∃ γ : Matrix ι ι ℤ, γ.det = 1 ∧ act d t (h * γ.map (Int.cast : ℤ → ℚ)) ∈ Z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteIntegralClassesClosedOrbitStatement

theorem **_root_.GlobalTheory.T.finite_integral_classes_closedOrbit** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → ℚ), IsClosed (orb d (fun k => ((t k : ℚ) : ℂ))) → ∀ D : ℕ, 0 < D → ∃ Z : Finset (TIdx ι d → ℚ), ∀ h : Matrix ι ι ℚ, h.det = 1 → (∀ k, ∃ z : ℤ, (D : ℚ) * act d t h k = z) → ∃ γ : Matrix ι ι ℤ, γ.det = 1 ∧ act d t (h * γ.map (Int.cast : ℤ → ℚ)) ∈ Z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteIntegralCosetsOfBoundedDenominator

theorem **adjugate_entry_mem_subring** {ι : Type} [Fintype ι] [DecidableEq ι] {R : Type} [CommRing R] (R₀ : Subring R) (A : Matrix ι ι R) (hA : ∀ i j, A i j ∈ R₀) (i j : ι) : A.adjugate i j ∈ R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteRationalPointsCompletedLattice

theorem **exists_denom** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hL : L.FG) : ∃ d : E, d ≠ 0 ∧ ∀ x ∈ L, ∀ i, IsIntegral ℤ (d * x i)

theorem **finite_of_denom** (d : E) (hd : d ≠ 0) (R : ℝ) : {x : E | IsIntegral ℤ (d * x) ∧ ∀ φ : E →+* ℂ, ‖φ x‖ ≤ R}.Finite

theorem **mem_spanSingleton_inv** (d : E) (hd : d ≠ 0) (x : E) : x ∈ FractionalIdeal.spanSingleton (𝒪 E)⁰ d⁻¹ ↔ IsIntegral ℤ (d * x)

theorem **isIntegral_star** (D : D7) {x : D.host.E} (hx : IsIntegral ℤ x) : IsIntegral ℤ (star x)

theorem **norm_emb_le** (D : D7) {x : D.host.E} {R : ℝ} (h : ∀ τ, ‖tauE D τ x‖ ≤ R) (φ : D.host.E →+* ℂ) : ‖φ x‖ ≤ R

theorem **finite_rational_points** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ R : ℝ, {g : Matrix (Fin T.rank) (Fin S.rank) D.host.E | (∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), ((ratW T S g : WO S T) - l).map (finPart D) = 0) ∧ ∀ τ i j, ‖tauE D τ (g i j)‖ ≤ R}.Finite

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteRationalPointsCompletedLatticeStatement

theorem **_root_.GlobalTheory.T.finite_rational_points_completed_lattice** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ R : ℝ, {g : Matrix (Fin T.rank) (Fin S.rank) D.host.E | (∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), ((ratW T S g : WO S T) - l).map (finPart D) = 0) ∧ ∀ τ i j, ‖tauE D τ (g i j)‖ ≤ R}.Finite

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FiniteResidueFieldAdicCompletionIntegers

theorem **isAdicComplete_adicCompletionIntegers** : IsAdicComplete (maximalIdeal (v.adicCompletionIntegers K)) (v.adicCompletionIntegers K)

theorem **isUnit_algebraMap_iff** (s : 𝒪 K) : IsUnit (algebraMap (𝒪 K) (v.adicCompletionIntegers K) s) ↔ s ∉ v.asIdeal

theorem **finite_residueField_adicCompletionIntegers** : Finite (ResidueField (v.adicCompletionIntegers K))

theorem **locEmb_apply** (x : E) : locEmb v x = 1 ⊗ₜ x

theorem **locConj_tmul** (a : v.adicCompletion (maximalRealSubfield E)) (x : E) : locConj v (a ⊗ₜ x) = a ⊗ₜ cc x

theorem **locConj_locEmb** (x : E) : locConj v (locEmb v x) = locEmb v (cc x)

theorem **locConj_algebraMap** (a : v.adicCompletion (maximalRealSubfield E)) : locConj v (algebraMap _ (LocAlg v) a) = algebraMap _ (LocAlg v) a

theorem **locConj_locConj** (z : LocAlg v) : locConj v (locConj v z) = z

theorem **locEmb_coe** (a : (maximalRealSubfield E)) : locEmb v (a : E) = algebraMap (v.adicCompletion (maximalRealSubfield E)) (LocAlg v) (algebraMap (maximalRealSubfield E) _ a)

theorem **locEmb_algebraMap** (s : 𝒪 (maximalRealSubfield E)) : locEmb v (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) = algebraMap (v.adicCompletionIntegers (maximalRealSubfield E)) (LocAlg v) (algebraMap (𝒪 (maximalRealSubfield E)) _ s)

theorem **algebraMap_locAlg_injective** : Function.Injective (algebraMap (v.adicCompletion (maximalRealSubfield E)) (LocAlg v))

theorem **exists_coord** {θ : E} (hθ : θ ≠ 0) (hθ : cc θ = -θ) (z : LocAlg v) : ∃ p q : v.adicCompletion (maximalRealSubfield E), z = algebraMap _ (LocAlg v) p + algebraMap _ (LocAlg v) q * locEmb v θ

theorem **exists_of_locConj_eq** {z : LocAlg v} (hz : locConj v z = z) : ∃ a : v.adicCompletion (maximalRealSubfield E), algebraMap _ (LocAlg v) a = z

theorem **locEmb_mem_locOrder** (a : 𝒪 E) : locEmb v a ∈ locOrder v

theorem **locConj_algebraMap'** (a : v.adicCompletionIntegers (maximalRealSubfield E)) : locConj v (algebraMap _ (LocAlg v) a) = algebraMap _ (LocAlg v) a

theorem **locConj_mem** {z : LocAlg v} (hz : z ∈ locOrder v) : locConj v z ∈ locOrder v

theorem **isIntegral_of_mem_locOrder** {z : LocAlg v} (hz : z ∈ locOrder v) : IsIntegral (v.adicCompletionIntegers (maximalRealSubfield E)) z

theorem **coe_locOrderConj** (z : locOrder v) : (locOrderConj v z : LocAlg v) = locConj v z

theorem **exists_of_mem_locOrder_of_locConj_eq** {z : LocAlg v} (hz : z ∈ locOrder v) (hc : locConj v z = z) : ∃ a : v.adicCompletionIntegers (maximalRealSubfield E), algebraMap _ (LocAlg v) a = z

theorem **exists_sub_conj_notMem** [Algebra.Unramified (𝒪 (maximalRealSubfield E)) (𝒪 E)] (w : Ideal (𝒪 E)) (hw : w ≠ ⊤) : ∃ x : 𝒪 E, x - IsCMField.ringOfIntegersComplexConj E x ∉ w

theorem **exists_isUnit_sub_locOrderConj** [Algebra.Unramified (𝒪 (maximalRealSubfield E)) (𝒪 E)] : ∃ ω : locOrder v, IsUnit (ω - locOrderConj v ω)

theorem **algebraMapSubmonoid_le** : Algebra.algebraMapSubmonoid (𝒪 E) v.asIdeal.primeCompl ≤ (𝒪 E)⁰

theorem **mem_span_semiLoc_iff** {V : Type*} [AddCommGroup V] [Module E V] [Module (𝒪 E) V] [IsScalarTower (𝒪 E) E V] (N : Submodule (𝒪 E) V) (x : V) : x ∈ Submodule.span (semiLoc v) (N : Set V) ↔ ∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ N

theorem **mem_semiLoc_iff** (z : E) : z ∈ semiLoc v ↔ ∃ s ∉ v.asIdeal, IsIntegral ℤ ((algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) * z)

theorem **cc_algebraMap** (s : 𝒪 (maximalRealSubfield E)) : cc (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) = algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s

theorem **cc_mem_semiLoc** {z : E} (hz : z ∈ semiLoc v) : cc z ∈ semiLoc v

theorem **locEmb_mem_locOrder_iff** (z : E) : locEmb v z ∈ locOrder v ↔ z ∈ semiLoc v

theorem **exists_frame** {L : Submodule (𝒪 E) (Fin r → E)} (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) : ∃ g : GL (Fin r) E, ∀ x, (∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L) ↔ ∀ i, (((g⁻¹ : GL (Fin r) E) : Matrix (Fin r) (Fin r) E) *ᵥ x) i ∈ semiLoc v

theorem **exists_generator** {𝔟 : FractionalIdeal (𝒪 E)⁰ E} (h𝔟 : 𝔟 ≠ 0) : ∃ b : E, b ≠ 0 ∧ ∀ z, (∃ s ∉ v.asIdeal, (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) * z ∈ 𝔟) ↔ b⁻¹ * z ∈ semiLoc v

theorem **herm_eq_dotProduct** (G : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : herm G x y = x ᵥ* G ⬝ᵥ fun i => cc (y i)

theorem **herm_mulVec** (G P : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : herm G (P *ᵥ x) (P *ᵥ y) = herm (Pᵀ * G * P.map cc) x y

theorem **one_notMem** : (1 : 𝒪 (maximalRealSubfield E)) ∉ v.asIdeal

theorem **exists_smul_col_mem** (hg : ∀ x, (∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L) ↔ ∀ i, (((g⁻¹ : GL (Fin r) E) : Matrix (Fin r) (Fin r) E) *ᵥ x) i ∈ semiLoc v) (j : Fin r) : ∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • (fun k => (g : Matrix (Fin r) (Fin r) E) k j) ∈ L

theorem **gram_mem_semiLoc** (hmod : ∀ x, x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ 𝔟) (hg : ∀ x, (∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L) ↔ ∀ i, (((g⁻¹ : GL (Fin r) E) : Matrix (Fin r) (Fin r) E) *ᵥ x) i ∈ semiLoc v) (hb : ∀ z, (∃ s ∉ v.asIdeal, (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) * z ∈ 𝔟) ↔ b⁻¹ * z ∈ semiLoc v) (i j : Fin r) : b⁻¹ * ((g : Matrix (Fin r) (Fin r) E)ᵀ * G * (g : Matrix (Fin r) (Fin r) E).map cc) i j ∈ semiLoc v

theorem **gram_inv_mem_semiLoc** (hd : G.det ≠ 0) (hL : L.FG) (hmod : ∀ x, x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ 𝔟) (hg : ∀ x, (∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L) ↔ ∀ i, (((g⁻¹ : GL (Fin r) E) : Matrix (Fin r) (Fin r) E) *ᵥ x) i ∈ semiLoc v) (hb0 : b ≠ 0) (hb : ∀ z, (∃ s ∉ v.asIdeal, (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) * z ∈ 𝔟) ↔ b⁻¹ * z ∈ semiLoc v) (i j : Fin r) : b * ((g : Matrix (Fin r) (Fin r) E)ᵀ * G * (g : Matrix (Fin r) (Fin r) E).map cc)⁻¹ i j ∈ semiLoc v

theorem **mem_locLattice_of_smul_mem** {s : 𝒪 (maximalRealSubfield E)} (hs : s ∉ v.asIdeal) {x : Fin r → E} (hx : algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L) : (fun i => locEmb v (x i)) ∈ locLattice v L

theorem **smul_mem_locLattice** {z : LocAlg v} (hz : z ∈ locOrder v) {x : Fin r → LocAlg v} (hx : x ∈ locLattice v L) : z • x ∈ locLattice v L

theorem **mem_locLattice_iff_of_frame** {g : GL (Fin r) E} (hg : ∀ x, (∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L) ↔ ∀ i, (((g⁻¹ : GL (Fin r) E) : Matrix (Fin r) (Fin r) E) *ᵥ x) i ∈ semiLoc v) (x : Fin r → LocAlg v) : x ∈ locLattice v L ↔ ∀ i, (((g⁻¹ : GL (Fin r) E) : Matrix (Fin r) (Fin r) E).map (locEmb v) *ᵥ x) i ∈ locOrder v

theorem **locEmb_mem_locLattice_iff** (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (x : Fin r → E) : (fun i => locEmb v (x i)) ∈ locLattice v L ↔ ∃ s ∉ v.asIdeal, algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s • x ∈ L

theorem **mem_iff_forall_mem_locLattice** {L : Submodule (𝒪 E) (Fin r → E)} (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (x : Fin r → E) : x ∈ L ↔ ∀ v : HeightOneSpectrum (𝒪 (maximalRealSubfield E)), (fun i => locEmb v (x i)) ∈ locLattice v L

theorem **finite_setOf_locLattice_ne** {L L' : Submodule (𝒪 E) (Fin r → E)} (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hL' : ReciprocityTheory.MultiplicativeHeights.IsLattice L') : {v : HeightOneSpectrum (𝒪 (maximalRealSubfield E)) | locLattice v L ≠ locLattice v L'}.Finite

theorem **locOrderConj_locOrderConj** (z : locOrder v) : locOrderConj v (locOrderConj v z) = z

theorem **mem_range_of_locOrderConj_eq** (z : locOrder v) (hz : locOrderConj v z = z) : z ∈ (algebraMap (v.adicCompletionIntegers (maximalRealSubfield E)) (locOrder v)).range

theorem **exists_eq_transpose_mul_map_locConj** {M N : Matrix (Fin r) (Fin r) (LocAlg v)} (hM : ∀ i j, M i j ∈ locOrder v) (hN : ∀ i j, N i j ∈ locOrder v) (hMN : M * N = 1) (hh : ∀ i j, locConj v (M i j) = M j i) : ∃ P : GL (Fin r) (LocAlg v), (∀ i j, (P : Matrix (Fin r) (Fin r) (LocAlg v)) i j ∈ locOrder v) ∧ (∀ i j, ((P⁻¹ : GL (Fin r) (LocAlg v)) : Matrix (Fin r) (Fin r) (LocAlg v)) i j ∈ locOrder v) ∧ M = (P : Matrix (Fin r) (Fin r) (LocAlg v))ᵀ * (P : Matrix (Fin r) (Fin r) (LocAlg v)).map (locConj v)

theorem **exists_unit_locConj_mul_eq** {𝔟 : FractionalIdeal (𝒪 E)⁰ E} (hc : ∀ z ∈ 𝔟, cc z ∈ 𝔟) {b : E} (hb0 : b ≠ 0) (hb : ∀ z, (∃ s ∉ v.asIdeal, (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) * z ∈ 𝔟) ↔ b⁻¹ * z ∈ semiLoc v) : ∃ ε : (locOrder v)ˣ, locConj v (((ε : locOrder v) : LocAlg v) * locEmb v b) = ((ε : locOrder v) : LocAlg v) * locEmb v b

theorem **mulVec_mem_locOrder** {P : Matrix (Fin r) (Fin r) (LocAlg v)} (hP : ∀ i j, P i j ∈ locOrder v) {x : Fin r → LocAlg v} (hx : ∀ i, x i ∈ locOrder v) (i : Fin r) : (P *ᵥ x) i ∈ locOrder v

theorem **exists_orthogonal_frame** {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, cc (G a b) = G b a) (hd : G.det ≠ 0) {L : Submodule (𝒪 E) (Fin r → E)} (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) {𝔟 : FractionalIdeal (𝒪 E)⁰ E} (hmod : ∀ x, x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ 𝔟) {b : E} (hb0 : b ≠ 0) (hb : ∀ z, (∃ s ∉ v.asIdeal, (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E) s : E) * z ∈ 𝔟) ↔ b⁻¹ * z ∈ semiLoc v) (ε : (locOrder v)ˣ) (hε : locConj v (((ε : locOrder v) : LocAlg v) * locEmb v b) = ((ε : locOrder v) : LocAlg v) * locEmb v b) : ∃ h : GL (Fin r) (LocAlg v), (h : Matrix (Fin r) (Fin r) (LocAlg v))ᵀ * G.map (locEmb v) * (h : Matrix (Fin r) (Fin r) (LocAlg v)).map (locConj v) = (((ε : locOrder v) : LocAlg v) * locEmb v b) • 1 ∧ ∀ x, x ∈ locLattice v L ↔ ∀ i, (((h⁻¹ : GL (Fin r) (LocAlg v)) : Matrix (Fin r) (Fin r) (LocAlg v)) *ᵥ x) i ∈ locOrder v

theorem **isLocalIsometry_one_iff** (v : HeightOneSpectrum (𝒪 (maximalRealSubfield E))) : IsLocalIsometry v G L G L' 1 ↔ locLattice v L' = locLattice v L

theorem **IsLocalIsometry.det_div_eq_mul_locConj** {v : HeightOneSpectrum (𝒪 (maximalRealSubfield E))} {γ : GL (Fin r) (LocAlg v)} (h : IsLocalIsometry v G L G' L' γ) (hd' : G'.det ≠ 0) : ∃ z : LocAlg v, locEmb v (G.det / G'.det) = z * locConj v z

theorem **congr_of_local_norm** (H : Published.Landherr) (H' : Published.HasseNormSt) (hG : ∀ a b, cc (G a b) = G b a) (hG' : ∀ a b, cc (G' a b) = G' b a) (hd : G.det ≠ 0) (hd' : G'.det ≠ 0) (hs : ∀ τ : E →+* ℂ, ∃ Q : Matrix (Fin r) (Fin r) ℂ, Q.det ≠ 0 ∧ G'.map τ = Qᵀ * G.map τ * Q.map conj) (hf : ∀ v : HeightOneSpectrum (𝒪 (maximalRealSubfield E)), ∃ z : LocAlg v, locEmb v (G.det / G'.det) = z * locConj v z) : ∃ P : Matrix (Fin r) (Fin r) E, P.det ≠ 0 ∧ G' = Pᵀ * G * P.map cc

theorem **exists_isLocalIsometry_of_modular** (hb : b ≠ 0) (hc : ∀ z ∈ b, cc z ∈ b) (hG : ∀ a b, cc (G a b) = G b a) (hG' : ∀ a b, cc (G' a b) = G' b a) (hd : G.det ≠ 0) (hd' : G'.det ≠ 0) (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hL' : ReciprocityTheory.MultiplicativeHeights.IsLattice L') (hmod : ∀ x, x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ b) (hmod' : ∀ x, x ∈ L' ↔ ∀ y ∈ L', herm G' x y ∈ b) (v : HeightOneSpectrum (𝒪 (maximalRealSubfield E))) : ∃ γ, IsLocalIsometry v G L G' L' γ

theorem **one_genus_of_modular** (hb : b ≠ 0) (hc : ∀ z ∈ b, cc z ∈ b) (hG : ∀ a b, cc (G a b) = G b a) (hd : G.det ≠ 0) (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hL' : ReciprocityTheory.MultiplicativeHeights.IsLattice L') (hmod : ∀ x, x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ b) (hmod' : ∀ x, x ∈ L' ↔ ∀ y ∈ L', herm G x y ∈ b) : ∃ γ : ∀ v : HeightOneSpectrum (𝒪 (maximalRealSubfield E)), GL (Fin r) (LocAlg v), (∀ v, IsLocalIsometry v G L G L' (γ v)) ∧ {v | γ v ≠ 1}.Finite

theorem **congr_of_modular** (H : Published.Landherr) (H' : Published.HasseNormSt) (hb : b ≠ 0) (hc : ∀ z ∈ b, cc z ∈ b) (hG : ∀ a b, cc (G a b) = G b a) (hG' : ∀ a b, cc (G' a b) = G' b a) (hd : G.det ≠ 0) (hd' : G'.det ≠ 0) (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hL' : ReciprocityTheory.MultiplicativeHeights.IsLattice L') (hmod : ∀ x, x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ b) (hmod' : ∀ x, x ∈ L' ↔ ∀ y ∈ L', herm G' x y ∈ b) (hs : ∀ τ : E →+* ℂ, ∃ Q : Matrix (Fin r) (Fin r) ℂ, Q.det ≠ 0 ∧ G'.map τ = Qᵀ * G.map τ * Q.map conj) : ∃ P : Matrix (Fin r) (Fin r) E, P.det ≠ 0 ∧ G' = Pᵀ * G * P.map cc

theorem **exists_orthonormal_frame** (hG : ∀ a b, cc (G a b) = G b a) (hd : G.det ≠ 0) (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hsd : ∀ x, x ∈ L ↔ ∀ y ∈ L, IsIntegral ℤ (herm G x y)) (v : HeightOneSpectrum (𝒪 (maximalRealSubfield E))) : ∃ h : GL (Fin r) (LocAlg v), (h : Matrix (Fin r) (Fin r) (LocAlg v))ᵀ * G.map (locEmb v) * (h : Matrix (Fin r) (Fin r) (LocAlg v)).map (locConj v) = 1 ∧ ∀ x, x ∈ locLattice v L ↔ ∀ i, (((h⁻¹ : GL (Fin r) (LocAlg v)) : Matrix (Fin r) (Fin r) (LocAlg v)) *ᵥ x) i ∈ locOrder v

theorem **cc_mem_inv_mul_different** {ξ : E} (hξ : cc ξ = -ξ) (z : E) (hz : z ∈ ((FractionalIdeal.spanSingleton (𝒪 E)⁰ ξ * (differentIdeal ℤ (𝒪 E) : Ideal (𝒪 E)))⁻¹ : FractionalIdeal (𝒪 E)⁰ E)) : cc z ∈ ((FractionalIdeal.spanSingleton (𝒪 E)⁰ ξ * (differentIdeal ℤ (𝒪 E) : Ideal (𝒪 E)))⁻¹ : FractionalIdeal (𝒪 E)⁰ E)

theorem **modular_of_perfect** {ξ : E} (hξ0 : ξ ≠ 0) (hd : G.det ≠ 0) (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (h : (∀ x ∈ L, ∀ y ∈ L, ∃ k : ℤ, EFf ξ G x y = k) ∧ ∀ φ : L →+ ℤ, ∃ x ∈ L, ∀ y : L, (φ y : ℚ) = EFf ξ G x y) (x : Fin r → E) : x ∈ L ↔ ∀ y ∈ L, herm G x y ∈ ((FractionalIdeal.spanSingleton (𝒪 E)⁰ ξ * (differentIdeal ℤ (𝒪 E) : Ideal (𝒪 E)))⁻¹ : FractionalIdeal (𝒪 E)⁰ E)

theorem **signMatrix_herm** (p : ℕ) (a b : Fin r) : cc (signMatrix E r p a b) = signMatrix E r p b a

theorem **det_signMatrix_ne_zero** (p : ℕ) : (signMatrix E r p).det ≠ 0

theorem **gram_signMatrix** (τ : E →+* ℂ) (p : ℕ) : gram τ (signMatrix E r p) 1 = Matrix.diagonal fun a : Fin r => ((sgn p (a : ℕ) : ℝ) : ℂ)

theorem **mem_frameLattice_one_iff** (x : Fin r → E) : x ∈ frameLattice (𝒪 E) (1 : GL (Fin r) E) ↔ ∀ i, IsIntegral ℤ (x i)

theorem **selfDual_signMatrix** (p : ℕ) (x : Fin r → E) : x ∈ frameLattice (𝒪 E) (1 : GL (Fin r) E) ↔ ∀ y ∈ frameLattice (𝒪 E) (1 : GL (Fin r) E), IsIntegral ℤ (herm (signMatrix E r p) x y)

theorem **congr_signMatrix** [Algebra.Unramified (𝒪 (maximalRealSubfield E)) (𝒪 E)] (H : Published.Landherr) (H' : Published.HasseNormSt) (hG : ∀ a b, cc (G a b) = G b a) (hd : G.det ≠ 0) (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hsd : ∀ x, x ∈ L ↔ ∀ y ∈ L, IsIntegral ℤ (herm G x y)) (Ψ : Set (E →+* ℂ)) (hΨ : ∀ τ : E →+* ℂ, τ ∈ Ψ ∨ ComplexEmbedding.conjugate τ ∈ Ψ) (p : ℕ) (hs : ∀ τ ∈ Ψ, ∃ Q : Matrix (Fin r) (Fin r) ℂ, Q.det ≠ 0 ∧ gram τ G Q = Matrix.diagonal fun a : Fin r => ((sgn p (a : ℕ) : ℝ) : ℂ)) : ∃ P : Matrix (Fin r) (Fin r) E, P.det ≠ 0 ∧ G = Pᵀ * signMatrix E r p * P.map cc

theorem **selfDual_lattices_one_genus** {E : Type} [Field E] [NumberField E] [IsCMField E] [Algebra.Unramified (𝒪 (maximalRealSubfield E)) (𝒪 E)] {r : ℕ} {G G' : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, cc (G a b) = G b a) (hG' : ∀ a b, cc (G' a b) = G' b a) (hd : G.det ≠ 0) (hd' : G'.det ≠ 0) {L L' : Submodule (𝒪 E) (Fin r → E)} (hL : ReciprocityTheory.MultiplicativeHeights.IsLattice L) (hL' : ReciprocityTheory.MultiplicativeHeights.IsLattice L') (hdual : ((∀ x, x ∈ L ↔ ∀ y ∈ L, IsIntegral ℤ (herm G x y)) ∧ ∀ x, x ∈ L' ↔ ∀ y ∈ L', IsIntegral ℤ (herm G' x y)) ∨ ∃ ξ : E, ξ ≠ 0 ∧ cc ξ = -ξ ∧ ((∀ x ∈ L, ∀ y ∈ L, ∃ k : ℤ, EFf ξ G x y = k) ∧ ∀ φ : L →+ ℤ, ∃ x ∈ L, ∀ y : L, (φ y : ℚ) = EFf ξ G x y) ∧ ((∀ x ∈ L', ∀ y ∈ L', ∃ k : ℤ, EFf ξ G' x y = k) ∧ ∀ φ : L' →+ ℤ, ∃ x ∈ L', ∀ y : L', (φ y : ℚ) = EFf ξ G' x y)) : (∀ v : HeightOneSpectrum (𝒪 (maximalRealSubfield E)), ∃ γ : GL (Fin r) (LocAlg v), IsLo

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FinrankSUnitsNumberFieldAddOne

theorem **exists_mem_only** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : ∃ a : 𝒪 F, a ≠ 0 ∧ a ∈ v.asIdeal ∧ ∀ w : HeightOneSpectrum (𝒪 F), w ≠ v → a ∉ w.asIdeal

theorem **sUnits_rank** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), Module.finrank ℤ (Additive (sUnits F S)) + 1 = S.card + Fintype.card (InfinitePlace F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FinrankSUnitsNumberFieldAddOneStatement

theorem **_root_.GlobalTheory.T.finrank_sUnitsNumberField_add_one** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), Module.finrank ℤ (Additive (sUnits F S)) + 1 = S.card + Fintype.card (InfinitePlace F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FinrankSpanBlockDiagPartImageLt

theorem **finrank_span_blk_image_lt** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ) (a : Matrix ι ι ℂ), a ∈ Submodule.span ℂ Γ → a ≠ 0 → blk e a = 0 → Module.finrank ℂ (Submodule.span ℂ (blk e '' Γ)) < Module.finrank ℂ (Submodule.span ℂ Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FinrankSpanBlockDiagPartImageLtStatement

theorem **_root_.GlobalTheory.T.finrank_span_blockDiagPart_image_lt** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ) (a : Matrix ι ι ℂ), a ∈ Submodule.span ℂ Γ → a ≠ 0 → blk e a = 0 → Module.finrank ℂ (Submodule.span ℂ (blk e '' Γ)) < Module.finrank ℂ (Submodule.span ℂ Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FirstInequalitySUnits

private theorem **index_sqs_top_eq** {G : Type} [CommGroup G] (U : Subgroup G) : (sqs (⊤ : Subgroup U)).index = (sqs U).relIndex U

private theorem **relIndex_sqs_range_eq** {U A : Type} [CommGroup U] [CommGroup A] (j : U →* A) (hj : Function.Injective j) : (sqs j.range).relIndex j.range = (sqs (⊤ : Subgroup U)).index

private theorem **arith_first_ineq** (m n e s t : ℕ) (hd : m ∣ n) (h : 2 * 2 ^ e * n = (2 ^ s) ^ 2) (hs : 2 * s ≤ e + t) : m ≠ 0 ∧ 2 * m ≤ 2 ^ t

theorem **first_inequality_sUnits** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → δ ∈ sUnits F S → ¬ IsSquare (δ : F) → (normSub (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≠ 0 ∧ 2 * (normSub (RingHom.id F) (δ :

F)).relIndex (sUnits F S) ≤ 2 ^ (tFin F δ S + tReal F δ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FirstInequalitySUnitsPrimeDegree

private theorem **pIdx_range_eq** (p : ℕ) {U A : Type} [CommGroup U] [CommGroup A] (j : U →* A) (hj : Function.Injective j) : pIdx p j.range = (pows p (⊤ : Subgroup U)).index

private theorem **index_pows_top_eq** (p : ℕ) {G : Type} [CommGroup G] (U : Subgroup G) : (pows p (⊤ : Subgroup U)).index = pIdx p U

private theorem **pIdx_top_eq** (p : ℕ) {G : Type} [CommGroup G] (U : Subgroup G) : pIdx p (⊤ : Subgroup U) = pIdx p U

private theorem **pTor_range_eq** (p : ℕ) {U A : Type} [CommGroup U] [CommGroup A] (j : U →* A) (hj : Function.Injective j) : pTor p j.range = pTor p (⊤ : Subgroup U)

private theorem **pTor_top_eq** (p : ℕ) {G : Type} [CommGroup G] (U : Subgroup G) : pTor p (⊤ : Subgroup U) = pTor p U

private theorem **arith_first_ineq** (p m n c e r t : ℕ) (hp : p.Prime) (hd : m ∣ n) (hn : n ≠ 0) (h : n ^ (p - 1) * p ^ (e + p * r) * p ^ p = 1 ^ (p - 1) * p * (p ^ (c + r)) ^ p) (hs : p * c ≤ e + (p - 1) * t) : m ≠ 0 ∧ p * m ≤ p ^ t

theorem **first_inequality_sUnits_prime** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → (kNorm p (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≠ 0 ∧ p * (kNorm p (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≤ p ^ tNon p F δ S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FirstInequalitySUnitsPrimeDegreeStatement

theorem **_root_.GlobalTheory.T.first_inequality_sUnits_prime_degree** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → (kNorm p (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≠ 0 ∧ p * (kNorm p (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≤ p ^ tNon p F δ S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FirstInequalitySUnitsStatement

theorem **_root_.GlobalTheory.T.two_mul_relIndex_normSub_sUnits_le** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → δ ∈ sUnits F S → ¬ IsSquare (δ : F) → (normSub (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≠ 0 ∧ 2 * (normSub (RingHom.id F) (δ : F)).relIndex (sUnits F S) ≤ 2 ^ (tFin F δ S + tReal F δ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FockVacuumAnnihilationEquationRealCoordinates

theorem **deriv_twisted_line** {R : Type} [NormedAddCommGroup R] [NormedSpace ℝ R] (F : R → ℂ) (hF : Differentiable ℝ F) (om : R →ₗ[ℝ] R →ₗ[ℝ] ℝ) (w u : R) : deriv (fun t : ℝ => Complex.exp (-(Real.pi * Complex.I) * (om w (t • u) : ℂ)) * F (w + t • u)) 0 = fderiv ℝ F w u - Real.pi * Complex.I * (om w u : ℂ) * F w

theorem **lineAt_eq** (h : Chart θ σ) (S T : HSp D) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (t : ℝ) : lineAt θ τ z t = iotaW σ T S (t • (Pi.single τ z : RW T S))

theorem **dRho_eq** (h : Chart θ σ) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (f : WO S T → ℂ) (hF : Differentiable ℝ fun z : RW T S => f (iotaW σ T S z)) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (w : RW T S) : Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f (iotaW σ T S w) = fderiv ℝ (fun z : RW T S => f (iotaW σ T S z)) w (Pi.single τ z) - Real.pi * Complex.I * (omO S T w (Pi.single τ z) : ℂ) * f (iotaW σ T S w)

theorem **jO_single** (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) : jO S T sS sT (Pi.single τ z) = Pi.single τ (cplxO (sS τ) (sT τ) z)

theorem **vacuum_annihilation** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) f) → Differentiable ℝ (fun z : RW T S => f (iotaW σ T S z)) ∧ ∀ w z : RW T S, (fderiv ℝ (fun z : RW T S => f (iotaW σ T S z)) w z - Real.pi * Complex.I * (omO S T w z : ℂ) * f (iotaW σ T S w)) + Complex.I * (fderiv ℝ (fun z : RW T S => f (iotaW σ T S z)) w (jO S T sS sT z) - Real.pi * Complex.I * (omO S T w (jO S T sS sT z) : ℂ) * f (iotaW σ T S w)) = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.FockVacuumAnnihilationEquationRealCoordinatesStatement

theorem **_root_.GlobalTheory.T.fockVacuum_annihilation_equation_real_coordinates** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) f) → Differentiable ℝ (fun z : RW T S => f (iotaW σ T S z)) ∧ ∀ w z : RW T S, (fderiv ℝ (fun z : RW T S => f (iotaW σ T S z)) w z - Real.pi * Complex.I * (omO S T w z : ℂ) * f (iotaW σ T S w)) + Complex.I * (fderiv ℝ (fun z : RW T S => f (iotaW σ T S z)) w (jO S T sS sT z) - Real.pi * Complex.I * (omO S T w (jO S T sS sT z) : ℂ) * f (iotaW σ T S w)) = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GaloisPlacesTransitive

theorem **galois_places_transitive** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E], (∀ (τ : E ≃ₐ[F] E) (w : HeightOneSpectrum (𝒪 E)), ∃ w' : HeightOneSpectrum (𝒪 E), w'.under (𝒪 F) = w.under (𝒪 F) ∧ ∀ y : E, w'.valuation E y = w.valuation E (τ y)) ∧ ∀ w w' : HeightOneSpectrum (𝒪 E), w'.under (𝒪 F) = w.under (𝒪 F) → ∃ τ : E ≃ₐ[F] E, ∀ y : E, w'.valuation E y = w.valuation E (τ y)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GaloisStableSubmoduleLeSpanRational

private theorem **exists_rational_support_subset_of_galois_stable** {V κ : Type} [AddCommGroup V] [Module ℂ V] (b : Module.Basis κ ℂ V) (W : Submodule ℂ V) (hW : ∀ σ : ℂ ≃+* ℂ, ∀ w ∈ W, ∃ w' ∈ W, ∀ k, b.repr w' k = σ (b.repr w k)) : ∀ n : ℕ, ∀ w ∈ W, w ≠ 0 → (b.repr w).support.card = n → ∃ u ∈ W, u ≠ 0 ∧ (∀ k, ∃ q : ℚ, b.repr u k = q) ∧ (b.repr u).support ⊆ (b.repr w).support

theorem **exists_rational_support_subset** {V κ : Type} [AddCommGroup V] [Module ℂ V] (b : Module.Basis κ ℂ V) (W : Submodule ℂ V) (hW : ∀ σ : ℂ ≃+* ℂ, ∀ w ∈ W, ∃ w' ∈ W, ∀ k, b.repr w' k = σ (b.repr w k)) (n : ℕ) : ∀ w ∈ W, w ≠ 0 → (b.repr w).support.card = n → ∃ u ∈ W, u ≠ 0 ∧ (∀ k, ∃ q : ℚ, b.repr u k = q) ∧ (b.repr u).support ⊆ (b.repr w).support

private theorem **mem_span_rational_of_galois_stable_of_support_card** {V κ : Type} [AddCommGroup V] [Module ℂ V] (b : Module.Basis κ ℂ V) (W : Submodule ℂ V) (hW : ∀ σ : ℂ ≃+* ℂ, ∀ w ∈ W, ∃ w' ∈ W, ∀ k, b.repr w' k = σ (b.repr w k)) : ∀ n : ℕ, ∀ w ∈ W, (b.repr w).support.card = n → w ∈ Submodule.span ℂ {w | w ∈ W ∧ ∀ k, ∃ q : ℚ, b.repr w k = q}

theorem **le_span_rational** : ∀ (V κ : Type) [AddCommGroup V] [Module ℂ V] (b : Module.Basis κ ℂ V) (W : Submodule ℂ V), (∀ σ : ℂ ≃+* ℂ, ∀ w ∈ W, ∃ w' ∈ W, ∀ k, b.repr w' k = σ (b.repr w k)) → W ≤ Submodule.span ℂ {w | w ∈ W ∧ ∀ k, ∃ q : ℚ, b.repr w k = q}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GaloisStableSubmoduleLeSpanRationalStatement

theorem **_root_.GlobalTheory.T.galoisStable_submodule_le_span_rational** : ∀ (V κ : Type) [AddCommGroup V] [Module ℂ V] (b : Module.Basis κ ℂ V) (W : Submodule ℂ V), (∀ σ : ℂ ≃+* ℂ, ∀ w ∈ W, ∃ w' ∈ W, ∀ k, b.repr w' k = σ (b.repr w k)) → W ≤ Submodule.span ℂ {w | w ∈ W ∧ ∀ k, ∃ q : ℚ, b.repr w k = q}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GlFreeAlgebraEmbIntegralAndRational

theorem **res_mul'** {R A β κ : Type} [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A) (g h : Matrix κ κ A) : res b (g * h) = res b g * res b h

theorem **res_one'** {R A β κ : Type} [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A) : res b (1 : Matrix κ κ A) = 1

theorem **repr_mem** {R A β : Type} [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] (b : Module.Basis β R A) (R₀ : Subring R) (h1 : ∀ i, b.repr 1 i ∈ R₀) (x : A) (hx : ∀ p q, Algebra.leftMulMatrix b x p q ∈ R₀) (k : β) : b.repr x k ∈ R₀

theorem **inv_mem** {R ι : Type} [CommRing R] [Fintype ι] [DecidableEq ι] (R₀ : Subring R) (M N : Matrix ι ι R) (hNM : N * M = 1) (hM : ∀ p q, M p q ∈ R₀) (hd : N.det ∈ R₀) (p q : ι) : N p q ∈ R₀

theorem **repr_map** {F β : Type} [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) {R A : Type} [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A) (hb : ∀ i, b i = algebraMap F A (ω i)) (x : F) (k : β) : b.repr (algebraMap F A x) k = algebraMap ℚ R (ω.repr x k)

theorem **res_map** {F β : Type} [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) {κ R A : Type} [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A) (hb : ∀ i, b i = algebraMap F A (ω i)) (g : Matrix κ κ F) : res b (g.map (algebraMap F A)) = (res ω g).map (algebraMap ℚ R)

theorem **emb_integral_and_rational** : (∀ (R A β κ : Type) [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A) (R₀ : Subring R) (u : GL κ A), (∀ i, b.repr 1 i ∈ R₀) → (∀ p q, emb b u p q ∈ R₀) → ∀ i j k, b.repr ((u : Matrix κ κ A) i j) k ∈ R₀ ∧ b.repr (((u⁻¹ : GL κ A) : Matrix κ κ A) i j) k ∈ R₀) ∧ ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (κ : Type) [Fintype κ] [DecidableEq κ] (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ γ : GL κ F, emb b (Units.map (RingHom.mapMatrix (algebraMap F A)).toMonoidHom γ) = (emb ω γ).map (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GlFreeAlgebraEmbIntegralAndRationalStatement

theorem **_root_.GlobalTheory.T.glFreeAlgebraEmb_integral_and_rational** : (∀ (R A β κ : Type) [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A) (R₀ : Subring R) (u : GL κ A), (∀ i, b.repr 1 i ∈ R₀) → (∀ p q, emb b u p q ∈ R₀) → ∀ i j k, b.repr ((u : Matrix κ κ A) i j) k ∈ R₀ ∧ b.repr (((u⁻¹ : GL κ A) : Matrix κ κ A) i j) k ∈ R₀) ∧ ∀ (F β : Type) [Field F] [Algebra ℚ F] [Fintype β] [DecidableEq β] (ω : Module.Basis β ℚ F) (κ : Type) [Fintype κ] [DecidableEq κ] (R A : Type) [CommRing R] [CommRing A] [Algebra ℚ R] [Algebra R A] [Algebra F A] (b : Module.Basis β R A), (∀ i, b i = algebraMap F A (ω i)) → ∀ γ : GL κ F, emb b (Units.map (RingHom.mapMatrix (algebraMap F A)).toMonoidHom γ) = (emb ω γ).map (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GlFreeAlgebraToSpecialLinearEmbInjectiveHom

theorem **res_one'** : res b (1 : Matrix κ κ A) = 1

theorem **res_mul'** (g h : Matrix κ κ A) : res b (g * h) = res b g * res b h

theorem **res_inj'** (g h : Matrix κ κ A) (e : res b g = res b h) : g = h

theorem **emb_one'** : emb b (1 : GL κ A) = 1

theorem **emb_mul'** (g h : GL κ A) : emb b (g * h) = emb b g * emb b h

theorem **emb_det'** (g : GL κ A) : (emb b g).det = 1

theorem **emb_adjugate'** (g : GL κ A) : (emb b g).adjugate = emb b g⁻¹

theorem **emb_injective'** : Function.Injective (fun g : GL κ A => emb b g)

theorem **emb_hom** : ∀ (R A β κ : Type) [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A), emb b (1 : GL κ A) = 1 ∧ (∀ g h : GL κ A, emb b (g * h) = emb b g * emb b h) ∧ (∀ g : GL κ A, (emb b g).det = 1 ∧ (emb b g).adjugate = emb b g⁻¹) ∧ Function.Injective (fun g : GL κ A => emb b g)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GlFreeAlgebraToSpecialLinearEmbInjectiveHomStatement

theorem **_root_.GlobalTheory.T.glFreeAlgebraToSpecialLinearEmb_injective_hom** : ∀ (R A β κ : Type) [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A), emb b (1 : GL κ A) = 1 ∧ (∀ g h : GL κ A, emb b (g * h) = emb b g * emb b h) ∧ (∀ g : GL κ A, (emb b g).det = 1 ∧ (emb b g).adjugate = emb b g⁻¹) ∧ Function.Injective (fun g : GL κ A => emb b g)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GramLeadingMinorLowerBoundIntegral

theorem **gramLB_shift** {n : Type} [Fintype n] [DecidableEq n] {A : Matrix n n ℝ} (hA : A.IsHermitian) (t : ℝ) : ∃ U : Matrix n n ℝ, U * Uᴴ = 1 ∧ A + t • (1 : Matrix n n ℝ) = U * diagonal (fun i => hA.eigenvalues i + t) * Uᴴ

theorem **gramLB_one_le_det** {n : Type} [Fintype n] [DecidableEq n] {R : Matrix n n ℝ} (hR : R.PosSemidef) : 1 ≤ (R + 1).det

theorem **gramLB_exists_shift** {n : Type} [Fintype n] [DecidableEq n] {A : Matrix n n ℝ} (hA : A.PosDef) : ∃ k : ℝ, 0 < k ∧ k ≤ 1 ∧ (A − k • (1 : Matrix n n ℝ)).PosSemidef

theorem **gramLB_det_mono** {n : Type} [Fintype n] [DecidableEq n] {P Q : Matrix n n ℝ} (hP : P.PosDef) (hQ : Q.PosSemidef) : P.det ≤ (P + Q).det

theorem **gramLB_posDef** {n : Type} [Fintype n] [DecidableEq n] {g : Matrix n n ℝ} (hg : g.det ≠ 0) : (gᵀ * g).PosDef

theorem **gram_leading_minor_lower_bound** : ∀ (N : ℕ) (g : Matrix (Fin N) (Fin N) ℝ), g.det ≠ 0 → ∃ c : ℝ, 0 < c ∧ ∀ γ : Matrix (Fin N) (Fin N) ℤ, γ.det ≠ 0 → ∀ (j : ℕ) (h : j ≤ N), c ≤ (((g * γ.map (Int.cast : ℤ → ℝ))ᵀ * (g * γ.map (Int.cast : ℤ → ℝ))).submatrix (Fin.castLE h) (Fin.castLE h)).det

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GramLeadingMinorLowerBoundIntegralStatement

theorem **_root_.GlobalTheory.T.gram_leading_minor_lower_bound_integral** : ∀ (N : ℕ) (g : Matrix (Fin N) (Fin N) ℝ), g.det ≠ 0 → ∃ c : ℝ, 0 < c ∧ ∀ γ : Matrix (Fin N) (Fin N) ℤ, γ.det ≠ 0 → ∀ (j : ℕ) (h : j ≤ N), c ≤ (((g * γ.map (Int.cast : ℤ → ℝ))ᵀ * (g * γ.map (Int.cast : ℤ → ℝ))).submatrix (Fin.castLE h) (Fin.castLE h)).det

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GramMinorSwapAdjacentColumns

private theorem **exch_sum_split** {N : ℕ} (p q : Fin N) (hq : (q : ℕ) = p + 1) (f : Fin N → ℝ) (hf : ∀ r, q < r → f r = 0) : ∑ r, f r = ∑ r ∈ Finset.univ.filter (· < p), f r + (f p + f q)

private theorem **det_updateRow_last_single_eq_mul_det_submatrix** {m : ℕ} (X : Matrix (Fin (m + 1)) (Fin (m + 1)) ℝ) (x : ℝ) : (X.updateRow (Fin.last m) (Pi.single (Fin.last m) x)).det = x * (X.submatrix Fin.castSucc Fin.castSucc).det

theorem **gram_minor_swap_columns** : ∀ (N : ℕ) (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n g g' : Matrix (Fin N) (Fin N) ℝ) (p q : Fin N), kᵀ * k = 1 → (∀ i, n i i = 1) → (∀ i j, j < i → n i j = 0) → (q : ℕ) = p + 1 → g = k * diagonal a * n → g' = g.submatrix id (Equiv.swap p q) → (∀ (j : ℕ) (h : j ≤ N), j ≠ q → ((g'ᵀ * g').submatrix (Fin.castLE h) (Fin.castLE h)).det = ((gᵀ * g).submatrix (Fin.castLE h) (Fin.castLE h)).det) ∧ ∀ h : (q : ℕ) ≤ N, ((g'ᵀ * g').submatrix (Fin.castLE h) (Fin.castLE h)).det = (∏ l ∈ Finset.univ.filter (· < p), a l ^ 2) * (a q ^ 2 + n p q ^ 2 * a p ^ 2)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.GramMinorSwapAdjacentColumnsStatement

theorem **_root_.GlobalTheory.T.gram_minor_swap_adjacent_columns** : ∀ (N : ℕ) (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n g g' : Matrix (Fin N) (Fin N) ℝ) (p q : Fin N), kᵀ * k = 1 → (∀ i, n i i = 1) → (∀ i j, j < i → n i j = 0) → (q : ℕ) = p + 1 → g = k * diagonal a * n → g' = g.submatrix id (Equiv.swap p q) → (∀ (j : ℕ) (h : j ≤ N), j ≠ q → ((g'ᵀ * g').submatrix (Fin.castLE h) (Fin.castLE h)).det = ((gᵀ * g).submatrix (Fin.castLE h) (Fin.castLE h)).det) ∧ ∀ h : (q : ℕ) ≤ N, ((g'ᵀ * g').submatrix (Fin.castLE h) (Fin.castLE h)).det = (∏ l ∈ Finset.univ.filter (· < p), a l ^ 2) * (a q ^ 2 + n p q ^ 2 * a p ^ 2)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasDerivAtZeroAndIsTheta

theorem **hasDerivAt_exp_mul** (ι : R →+ G) (c : ℂ) (f : G → ℂ) (hf : Sm ι f) (u : R) (w : G) : HasDerivAt (fun t : ℝ => Complex.exp (t * c) * f (w + ι (t • u))) (c * f w + fderiv ℝ (fun x : R => f (w + ι x)) 0 u) 0

theorem **contDiff_additive** [FiniteDimensional ℝ R] (φ : R →+ ℂ) (h : Continuous φ) : ContDiff ℝ ∞ φ

theorem **sm_deriv** (ι : R →+ G) (φ : G →+ ℂ) (hφ : ContDiff ℝ ∞ fun x => φ (ι x)) (f : G → ℂ) (hf : Sm ι f) (u : R) : Sm ι fun w => φ w * f w + fderiv ℝ (fun x : R => f (w + ι x)) 0 u

theorem **sm_translate** (ι : R →+ G) (φ : G →+ ℂ) (hφ : ContDiff ℝ ∞ fun x => φ (ι x)) (f : G → ℂ) (hf : Sm ι f) (v : G) : Sm ι fun w => Complex.exp (φ w) * f (w + v)

theorem **symplOH_add_left** (S T : HSp D) (x x' y : WO S T) : symplOH S T (x + x') y = symplOH S T x y + symplOH S T x' y

theorem **symplOH_add_right** (S T : HSp D) (x y y' : WO S T) : symplOH S T x (y + y') = symplOH S T x y + symplOH S T x y'

theorem **symplOH_zero_left** (S T : HSp D) (y : WO S T) : symplOH S T 0 y = 0

theorem **symplOH_zero_right** (S T : HSp D) (y : WO S T) : symplOH S T y 0 = 0

theorem **gram_inv_star** (V : HSp D) (i j) : star (V.gram⁻¹ i j) = V.gram⁻¹ j i

theorem **herm_sum** {A : Type} [CommRing A] {m n : ℕ} (σ : A →+* A) (hσ : ∀ a, σ (σ a) = a) (P : Fin m → Fin m → A) (Q : Fin n → Fin n → A) (hP : ∀ i k, σ (P i k) = P k i) (hQ : ∀ i k, σ (Q i k) = Q k i) (x y : Matrix (Fin n) (Fin m) A) : σ (∑ a, ∑ b, ∑ c, ∑ d, P a b * σ (y c b) * Q c d * x d a) = ∑ a, ∑ b, ∑ c, ∑ d, P a b * σ (x c b) * Q c d * y d a

theorem **hO_herm** (S T : HSp D) (x y : WO S T) : cA D (hO S T x y) = hO S T y x

theorem **symplO_skew** (S T : HSp D) (x y : WO S T) : symplO S T y x = -symplO S T x y

theorem **symplOH_skew** (S T : HSp D) (x y : WO S T) : symplOH S T y x = -symplOH S T x y

theorem **continuous_symplOH** (S T : HSp D) {X : Type} [TopologicalSpace X] {x y : X → WO S T} (hx : Continuous x) (hy : Continuous y) : Continuous fun t => symplOH S T (x t) (y t)

theorem **sum_reorder** {A : Type} [CommRing A] {m n : ℕ} (σ : A → A) (P : Fin m → Fin m → A) (Q : Fin n → Fin n → A) (x y : Matrix (Fin n) (Fin m) A) : ∑ a, ∑ b, ∑ c, ∑ d, P a b * σ (y c b) * Q c d * x d a = ∑ i, ∑ j, ∑ k, ∑ l, σ (yᵀ i j) * P k i * Q j l * xᵀ k l

theorem **map_symplOH_eq_zero** {B : Type} [CommRing B] [Algebra D.host.F B] (p : AF D.host.F →ₐ[D.host.F] B) (S T : HSp D) (x y : WO S T) (h : x.map (Algebra.TensorProduct.map p (AlgHom.id D.host.F D.host.E)).toRingHom = 0 ∨ y.map (Algebra.TensorProduct.map p (AlgHom.id D.host.F D.host.E)).toRingHom = 0) : p (symplOH S T x y) = 0

theorem **snd_symplOH** (S T : HSp D) (x y : WO S T) (h : x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) : sndA D.host.F (symplOH S T x y) = 0

theorem **fst_symplOH** (S T : HSp D) (x y : WO S T) (h : x.map (infPart D) = 0 ∨ y.map (infPart D) = 0) : fstA D.host.F (symplOH S T x y) = 0

theorem **symplOH_fin_real** (S T : HSp D) (u v : WO S T) (hu : u.map (infPart D) = 0) (hv : v.map (finPart D) = 0) : symplOH S T u v = 0 ∧ symplOH S T v u = 0

theorem **exists_log** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ lam : AF D.host.F →+ ℂ, Continuous lam ∧ ∀ a, sndA D.host.F a = 0 → ψ a = Complex.exp (lam a)

theorem **heisO_heisO** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w w' : WO S T) (f : WO S T → ℂ) : heisO ψ S T w (heisO ψ S T w' f) = ψ (symplOH S T w w') • heisO ψ S T (w + w') f

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **heisO_quasi** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w₀ : WO S T) (f : WO S T → ℂ) (hf : ∀ γ w, f (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * f w) (γ w) : heisO ψ S T w₀ f (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * heisO ψ S T w₀ f w

theorem **heisO_fix_of_real** (u v : WO S T) (hu : u.map (infPart D) = 0) (hv : v.map (finPart D) = 0) (f : WO S T → ℂ) (hf : heisO ψ S T u f = f) : heisO ψ S T u (heisO ψ S T v f) = heisO ψ S T v f

theorem **isTheta_zero** : IsThetaFunO ψ S T 0 where

theorem **isTheta_add** {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : IsThetaFunO ψ S T (f + g) where

theorem **isTheta_smul** (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (c • f) where

theorem **isTheta_sub** {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : IsThetaFunO ψ S T (f - g)

theorem **continuous_of_level** (h : Chart θ σ) (hψ : Continuous ψ) (f : WO S T → ℂ) (hU : ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u f = f) (hs : Sm (iotaW σ T S) f) : Continuous f

theorem **contDiff_phase** (h : Chart θ σ) {lam : AF D.host.F →+ ℂ} (hl : Continuous lam) (v : WO S T) : ContDiff ℝ ∞ fun x => phase lam S T v (iotaW σ T S x)

theorem **isTheta_heisO_real** (hψ : IsStdChar D ψ (-1)) {v : WO S T} (hv : v.map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (heisO ψ S T v f)

theorem **additive_real** (φ : ℝ →+ ℂ) (h : Continuous φ) (t : ℝ) : φ t = t * φ 1

theorem **psi_sympl_line** {lam : AF D.host.F →+ ℂ} (hl : Continuous lam) (hlam : ∀ a, sndA D.host.F a = 0 → ψ a = Complex.exp (lam a)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) (w : WO S T) (t : ℝ) : ψ (symplOH S T w (L t)) = Complex.exp (t * phase lam S T (L 1) w)

theorem **hasDerivAt_zero_and_isTheta** (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : (∀ w, HasDerivAt (fun t : ℝ => heisO ψ S T (L t) f w) (dLine ψ S T L f w) 0) ∧ IsThetaFunO ψ S T (dLine ψ S T L f)

theorem **heisO_line_add** (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) (s t : ℝ) (f : WO S T → ℂ) : heisO ψ S T (L s) (heisO ψ S T (L t) f) = heisO ψ S T (L (s + t)) f

theorem **hasDerivAt_line** (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S T) (s : ℝ) : HasDerivAt (fun t : ℝ => heisO ψ S T (L t) f w) (heisO ψ S T (L s) (dLine ψ S T L f) w) s

theorem **isLineAt_lineAt** (hθ : IsInfFam θ) {m n : ℕ} (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) : IsLineAt θ τ z (lineAt θ τ z)

theorem **isTheta_dRho** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasFiniteAdelicDoubleCosetsOfBlockDiagPartImage

private lemma **exists_nat_mul_ev_mem** {R : Type} [CommRing R] [Algebra ℚ R] (R₀ : Subring R) (f : MvPolynomial (ι × ι) ℚ) : ∃ N : ℕ, 0 < N ∧ ∀ u : Matrix ι ι R, (∀ i j, u i j ∈ R₀) → (N : R) * _root_.GlobalTheory.ClassFin.ev f u ∈ R₀

private lemma **exists_nat_mul_ev_section_mem** {R : Type} [CommRing R] [Algebra ℚ R] (R₀ : Subring R) (b : ι → ι → MvPolynomial (ι × ι) ℚ) : ∃ N : ℕ, 0 < N ∧ ∀ u : Matrix ι ι R, (∀ i j, u i j ∈ R₀) → ∀ i j, (N : R) * _root_.GlobalTheory.ClassFin.ev (b i j) u ∈ R₀

private lemma **exists_nat_mul_rat_eq_int** {κ : Type} [Fintype κ] (q : κ → ℚ) : ∃ M : ℕ, 0 < M ∧ ∀ k, ∃ z : ℤ, (M : ℚ) * q k = z

private lemma **exists_nat_mul_linExt_mem** {R : Type} [CommRing R] [Algebra ℚ R] (R₀ : Subring R) (c : ι × ι → ι × ι → ℚ) : ∃ K : ℕ, 0 < K ∧ ∀ A : Matrix ι ι R, (∀ i j, A i j ∈ R₀) → ∀ i j, (K : R) * _root_.GlobalTheory.ClassFin.linExt c A i j ∈ R₀

private lemma **matrix_mul_mem_subring_of_mem** {R : Type} [CommRing R] (R₀ : Subring R) {A B : Matrix ι ι R} (hA : ∀ i j, A i j ∈ R₀) (hB : ∀ i j, B i j ∈ R₀) : ∀ i j, (A * B) i j ∈ R₀

private lemma **exists_finset_integral_classes** {R : Type} [CommRing R] [Algebra ℚ R] (R₀ : Subring R) (S : Set (Matrix ι ι R)) (hone : 1 ∈ S) (hmul : ∀ x ∈ S, ∀ y ∈ S, x * y ∈ S) (hadj : ∀ x ∈ S, x.adjugate ∈ S) (hdet : ∀ x ∈ S, x.det = 1) (hi : ∀ M : ℕ, 0 < M → ∀ r ∈ R₀, ∃ z : ℤ, ∃ a ∈ R₀, r = (z : R) + (M : R) * a) (hunit : ∀ N : ℕ, 0 < N → IsUnit (N : R)) (N : ℕ) (hN : 0 < N) : ∃ F : Finset (Matrix ι ι R), (∀ t ∈ F, t ∈ S) ∧ ∀ w ∈ S, (∀ i j, (N : R) * w i j ∈ R₀) → ∃ t ∈ F, ∃ p ∈ S, (∀ i j, p i j ∈ R₀) ∧ w = t * p

private lemma **idempotent_block_mul_of_upper** {A : Type*} [Ring A] (E x y : A) (hE : E * E = E) (hx : (1 - E) * x * E = 0) (hy : (1 - E) * y * E = 0) : E * (x * y) * E + (1 - E) * (x * y) * (1 - E) = (E * x * E + (1 - E) * x * (1 - E)) * (E * y * E + (1 - E) * y * (1 - E))

private lemma **blk_mul_of_upper** {R : Type} [CommRing R] [Algebra ℚ R] (e : Matrix ι ι ℚ) (he : e * e = e) (x y : Matrix ι ι R) (hx : (1 - e.map (algebraMap ℚ R)) * x * e.map (algebraMap ℚ R) = 0) (hy : (1 - e.map (algebraMap ℚ R)) * y * e.map (algebraMap ℚ R) = 0) : _root_.GlobalTheory.ClassFin.blk e (x * y) = _root_.GlobalTheory.ClassFin.blk e x * _root_.GlobalTheory.ClassFin.blk e y

private lemma **idempotent_block_one** {A : Type*} [Ring A] (E : A) (hE : E * E = E) : E * 1 * E + (1 - E) * 1 * (1 - E) = 1

private lemma **blk_one_of_idempotent** {R : Type} [CommRing R] [Algebra ℚ R] (e : Matrix ι ι ℚ) (he : e * e = e) : _root_.GlobalTheory.ClassFin.blk e (1 : Matrix ι ι R) = 1

private lemma **blk_map_algebraMap_eq_map_blk** {R : Type} [CommRing R] [Algebra ℚ R] (e g : Matrix ι ι ℚ) : _root_.GlobalTheory.ClassFin.blk e (g.map (algebraMap ℚ R)) = (_root_.GlobalTheory.ClassFin.blk e g).map (algebraMap ℚ R)

private lemma **linExt_add** {R : Type} [CommRing R] [Algebra ℚ R] (c : ι × ι → ι × ι → ℚ) (X Y : Matrix ι ι R) : _root_.GlobalTheory.ClassFin.linExt c (X + Y) = _root_.GlobalTheory.ClassFin.linExt c X + _root_.GlobalTheory.ClassFin.linExt c Y

private lemma **linExt_smul** {R : Type} [CommRing R] [Algebra ℚ R] (c : ι × ι → ι × ι → ℚ) (k : R) (X : Matrix ι ι R) : _root_.GlobalTheory.ClassFin.linExt c (k • X) = k • _root_.GlobalTheory.ClassFin.linExt c X

private lemma **linExt_map_algebraMap** {R : Type} [CommRing R] [Algebra ℚ R] (c : ι × ι → ι × ι → ℚ) (X : Matrix ι ι ℚ) : (_root_.GlobalTheory.ClassFin.linExt c X).map (algebraMap ℚ R) = _root_.GlobalTheory.ClassFin.linExt c (X.map (algebraMap ℚ R))

theorem **_root_.GlobalTheory.ClassFin.cf_of_cf_blk_image** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), _root_.GlobalTheory.ClassFin.IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → _root_.GlobalTheory.ClassFin.AdelicPair R R₀ → _root_.GlobalTheory.ClassFin.CF (_root_.GlobalTheory.ClassFin.blk e '' Γ) R R₀ → _root_.GlobalTheory.ClassFin.CF Γ R R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasFiniteAdelicDoubleCosetsOfBlockDiagPartImageStatement

theorem **_root_.GlobalTheory.T.hasFiniteAdelicDoubleCosets_of_blockDiagPart_image** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ) (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), IsQGroup Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → AdelicPair R R₀ → CF (blk e '' Γ) R R₀ → CF Γ R R₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasSumLogZetaSqMul

theorem **hasSum_neg_log_succ** {z : ℂ} (hz : ‖z‖ < 1) : HasSum (fun k : ℕ => z ^ (k + 1) / ((k : ℂ) + 1)) (-Complex.log (1 - z))

theorem **hasSum_prod_neg_log** {ι : Type} (z : ι → ℂ) (hz : ∀ i, ‖z i‖ ≤ 1 / 2) (hs : Summable fun i => ‖z i‖) : HasSum (fun p : ι × ℕ => z p.1 ^ (p.2 + 1) / ((p.2 : ℂ) + 1)) (∑' i, -Complex.log (1 - z i))

theorem **hasSum_one** {ι : Type} (N : ι → ℕ) (c : ι → ℂ) (hN : ∀ i, 2 ≤ N i) (hc : ∀ i, ‖c i‖ ≤ 1) (s : ℂ) (hs : 1 < s.re) (hsum : Summable fun i => (N i : ℝ) ^ (-s.re)) : HasSum (fun p : ι × ℕ => c p.1 ^ (p.2 + 1) / ((p.2 : ℂ) + 1) * Complex.exp (-((((p.2 : ℝ) + 1) * Real.log (N p.1) : ℝ) : ℂ) * s)) (∑' i, -Complex.log (1 - c i * ((N i : ℕ) : ℂ) ^ (-s)))

theorem **hasSum_log_zeta_sq_mul** : ∀ (ι : Type) (N : ι → ℕ) (c : ι → ℂ), (∀ i, 2 ≤ N i) → (∀ i, ‖c i‖ ≤ 1) → (∀ σ : ℝ, 1 < σ → Summable fun i => (N i : ℝ) ^ (-σ)) → ∀ s : ℂ, 1 < s.re → HasSum (fun p : ι × ℕ => (((2 + 2 * (c p.1 ^ (p.2 + 1)).re) / ((p.2 : ℝ) + 1) : ℝ) : ℂ) * Complex.exp (-((((p.2 : ℝ) + 1) * Real.log (N p.1) : ℝ) : ℂ) * s)) (2 * (∑' i, -Complex.log (1 - 1 * ((N i : ℕ) : ℂ) ^ (-s))) + (∑' i, -Complex.log (1 - c i * ((N i : ℕ) : ℂ) ^ (-s))) + (∑' i, -Complex.log (1 - starRingEnd ℂ (c i) * ((N i : ℕ) : ℂ) ^ (-s))))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasseNormPrincipleOfCyclicTower

private theorem **prod_univ_eq_prod_range_orderOf_of_generator** {G β : Type} [Group G] [Fintype G] [CommMonoid β] {g : G} (hg : ∀ x, x ∈ Subgroup.zpowers g) (f : G → β) : ∏ σ, f σ = ∏ i ∈ Finset.range (orderOf g), f (g ^ i)

private theorem **prod_range_pow_pow_prod_range_eq_prod_range_mul** {F E : Type} [Field F] [Field E] [Algebra F E] (τ : E ≃ₐ[F] E) (y : E) (d n : ℕ) : ∏ j ∈ Finset.range n, ((τ ^ d) ^ j) (∏ i ∈ Finset.range d, (τ ^ i) y) = ∏ m ∈ Finset.range (d * n), (τ ^ m) y

private theorem **exists_eq_pow_finrank_mul_norm_of_norm_eq_cyclic** (F M E : Type) [Field F] [Field M] [Field E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [FiniteDimensional F E] [IsGalois F E] (hcyc : IsCyclic (E ≃ₐ[F] E)) (x : F) (hx : x ≠ 0) (e₀ : E) (he₀ : Algebra.norm M e₀ = algebraMap F M x) : ∃ b : F, ∃ c : E, c ≠ 0 ∧ x = b ^ Module.finrank M E * Algebra.norm F c

theorem **_root_.GlobalTheory.CNorm.np_of_cyclic_tower** : ∀ (F M E : Type) [Field F] [NumberField F] [Field M] [NumberField M] [Field E] [NumberField E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [IsGalois F E], IsCyclic (E ≃ₐ[F] E) → _root_.GlobalTheory.CNorm.NP M E → _root_.GlobalTheory.CNorm.NP F M → _root_.GlobalTheory.CNorm.NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasseNormPrincipleOfCyclicTowerStatement

theorem **_root_.GlobalTheory.T.hasseNormPrinciple_of_cyclic_tower** : ∀ (F M E : Type) [Field F] [NumberField F] [Field M] [NumberField M] [Field E] [NumberField E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [IsGalois F E], IsCyclic (E ≃ₐ[F] E) → NP M E → NP F M → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasseNormPrincipleOfIsCyclic

theorem **np_of_finrank_one** (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] (h : Module.finrank F E = 1) : NP F E

theorem **np_of_finrank_prime** (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E] (hp : p.Prime) (h : Module.finrank F E = p) : NP F E

theorem **hasseNormPrinciple_cyclic_of_finrank** (n : ℕ) : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], Module.finrank F E = n → IsGalois F E → IsCyclic (E ≃ₐ[F] E) → NP F E

theorem **np_cyclic** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], IsGalois F E → IsCyclic (E ≃ₐ[F] E) → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasseNormPrincipleOfIsCyclicStatement

theorem **_root_.GlobalTheory.T.hasseNormPrinciple_of_isCyclic** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], IsGalois F E → IsCyclic (E ≃ₐ[F] E) → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasseNormTheoremQuadraticExtension

theorem **hasseNormSt** : GlobalTheory.Published.HasseNormSt

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HasseNormTheoremQuadraticExtensionStatement

theorem **_root_.GlobalTheory.T.hasse_norm_theorem_quadratic_extension** : GlobalTheory.Published.HasseNormSt

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HeightOneSpectrumPlacesOverFiniteSet

theorem **_root_.GlobalTheory.QNorm.finite_places_over** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) : {w : HeightOneSpectrum (𝒪 E) | w.under (𝒪 F) = v}.Finite

theorem **_root_.GlobalTheory.QNorm.places_over** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], (∀ S : Finset (HeightOneSpectrum (𝒪 F)), ∃ SE : Finset (HeightOneSpectrum (𝒪 E)), ∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) ∧ (∀ v : HeightOneSpectrum (𝒪 F), ∃ w : HeightOneSpectrum (𝒪 E), w.under (𝒪 F) = v) ∧ ∀ (w : HeightOneSpectrum (𝒪 E)), ∃ e : ℕ, 0 < e ∧ ∀ x : F, w.valuation E (algebraMap F E x) = (w.under (𝒪 F)).valuation F x ^ e

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HeightOneSpectrumPlacesOverFiniteSetStatement

theorem **_root_.GlobalTheory.T.heightOneSpectrum_places_over_finite_set** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], (∀ S : Finset (HeightOneSpectrum (𝒪 F)), ∃ SE : Finset (HeightOneSpectrum (𝒪 E)), ∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) ∧ (∀ v : HeightOneSpectrum (𝒪 F), ∃ w : HeightOneSpectrum (𝒪 E), w.under (𝒪 F) = v) ∧ ∀ (w : HeightOneSpectrum (𝒪 E)), ∃ e : ℕ, 0 < e ∧ ∀ x : F, w.valuation E (algebraMap F E x) = (w.under (𝒪 F)).valuation F x ^ e

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HerbrandPrimeOrder

theorem **herbPhi_arith** (q h0 h1 a t bI bT cI cT i c : ℕ) (hi : i ≠ 0) (hcT : cT ≠ 0) (e1 : a * bT * cT = bI * cI * t) (e2 : cI * c ^ q = i ^ q * cT) (e3 : h0 * i * bT = bI * h1 * c) : h0 ^ q * a * bT ^ (q + 1) = h1 ^ q * t * bI ^ (q + 1)

theorem **herbPhi_nrm_dif** {A : Type} [CommGroup A] (σ : A →* A) (n : ℕ) (a : A) : nrm σ n (dif σ a) = σ^[n] a / a

theorem **herbPhi_stable** (p : ℕ) (A : Type) [CommGroup A] (σ : A →* A) (hp : p.Prime) (C : Subgroup A) (hst : ∀ x ∈ C, σ x ∈ C) (hn : ∀ x ∈ C, nrm σ p x = 1) (hI : pIdx p C ≠ 0) (hT : pTor p C ≠ 0) : (C.map (dif σ)).relIndex C ≠ 0 ∧ Nat.card ↥(C ⊓ fix σ) ≠ 0 ∧ pIdx p C * Nat.card ↥(C ⊓ fix σ) ^ (p - 1) = (C.map (dif σ)).relIndex C ^ (p - 1) * pTor p C

theorem **herbrand_prime_order** : ∀ (p : ℕ) (A : Type) [CommGroup A] (σ : A →* A), p.Prime → (∀ a, σ^[p] a = a) → pIdx p (⊤ : Subgroup A) ≠ 0 → pTor p (⊤ : Subgroup A) ≠ 0 → (nrm σ p).range.relIndex (fix σ) ≠ 0 ∧ (dif σ).range.relIndex (nrm σ p).ker ≠ 0 ∧ pIdx p (fix σ) ≠ 0 ∧ pTor p (fix σ) ≠ 0 ∧ (nrm σ p).range.relIndex (fix σ) ^ (p - 1) * pIdx p (⊤ : Subgroup A) * pTor p (fix σ) ^ p = (dif σ).range.relIndex (nrm σ p).ker ^ (p - 1) * pTor p (⊤ : Subgroup A) * pIdx p (fix σ) ^ p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HerbrandPrimeOrderStatement

theorem **_root_.GlobalTheory.T.herbrand_prime_order'** : ∀ (p : ℕ) (A : Type) [CommGroup A] (σ : A →* A), p.Prime → (∀ a, σ^[p] a = a) → pIdx p (⊤ : Subgroup A) ≠ 0 → pTor p (⊤ : Subgroup A) ≠ 0 → (nrm σ p).range.relIndex (fix σ) ≠ 0 ∧ (dif σ).range.relIndex (nrm σ p).ker ≠ 0 ∧ pIdx p (fix σ) ≠ 0 ∧ pTor p (fix σ) ≠ 0 ∧ (nrm σ p).range.relIndex (fix σ) ^ (p - 1) * pIdx p (⊤ : Subgroup A) * pTor p (fix σ) ^ p = (dif σ).range.relIndex (nrm σ p).ker ^ (p - 1) * pTor p (⊤ : Subgroup A) * pIdx p (fix σ) ^ p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HermitianLogTraceSqFirstVariationBound

private lemma **sum_normSq_apply_right_eq_one_of_mul_conjTranspose** (W : Matrix ι ι ℂ) (h : W * Wᴴ = 1) (i : ι) : ∑ j, Complex.normSq (W i j) = 1

private lemma **sum_normSq_apply_left_eq_one_of_conjTranspose_mul** (W : Matrix ι ι ℂ) (h : Wᴴ * W = 1) (j : ι) : ∑ i, Complex.normSq (W i j) = 1

private lemma **norm_apply_le_one_of_conjTranspose_mul_eq_one** (V : Matrix ι ι ℂ) (hV : Vᴴ * V = 1) (k j : ι) : ‖V k j‖ ≤ 1

private lemma **norm_conj_diag_apply_le** (V Y : Matrix ι ι ℂ) (hV : Vᴴ * V = 1) (δ : ℝ) (hY : ∀ i j, ‖Y i j‖ ≤ δ) (j : ι) : ‖(Vᴴ * Y * V) j j‖ ≤ (Fintype.card ι : ℝ) * Fintype.card ι * δ

private lemma **le_eigenvalue_of_add** (U V Y : Matrix ι ι ℂ) (x y : ι → ℝ) (m δ : ℝ) (hU : U * Uᴴ = 1) (hV : V * Vᴴ = 1) (hx : ∀ i, m ≤ x i) (hY : ∀ i j, ‖Y i j‖ ≤ δ) (hB : U * diagonal (fun i => (x i : ℂ)) * Uᴴ + Y = V * diagonal (fun j => (y j : ℂ)) * Vᴴ) (j : ι) : m - (Fintype.card ι : ℝ) * Fintype.card ι * δ ≤ y j

private lemma **conj_real_diagonal_mul_conj_real_diagonal** (U : Matrix ι ι ℂ) (hU : Uᴴ * U = 1) (a b : ι → ℝ) : U * diagonal (fun i => (a i : ℂ)) * Uᴴ * (U * diagonal (fun i => (b i : ℂ)) * Uᴴ) = U * diagonal (fun i => ((a i * b i : ℝ) : ℂ)) * Uᴴ

private lemma **re_trace_conj_diagonal_mul** (U V : Matrix ι ι ℂ) (a b : ι → ℝ) : Complex.re (Matrix.trace (U * diagonal (fun i => (a i : ℂ)) * Uᴴ * (V * diagonal (fun j => (b j : ℂ)) * Vᴴ))) = ∑ i, ∑ j, a i * b j * Complex.normSq ((Uᴴ * V) i j)

private lemma **sum_norm_sq_eq_re_trace_mul_conjTranspose** (Y : Matrix ι ι ℂ) : ∑ i, ∑ j, ‖Y i j‖ ^ (2 : ℕ) = Complex.re (Matrix.trace (Y * Yᴴ))

private lemma **conjTranspose_unitary_conj_real_diagonal** (U : Matrix ι ι ℂ) (a : ι → ℝ) : (U * diagonal (fun i => (a i : ℂ)) * Uᴴ)ᴴ = U * diagonal (fun i => (a i : ℂ)) * Uᴴ

theorem **_root_.GlobalTheory.ClassFin.LogSqVar.first_variation** (ι : Type) [Fintype ι] [DecidableEq ι] (H A : Matrix ι ι ℂ) : _root_.GlobalTheory.ClassFin.IsLog H A → ∃ (C : ℝ) (δ : ℝ), (0 : ℝ) < δ ∧ ∀ (Y H' : Matrix ι ι ℂ), (∀ (i j : ι), ‖Y i j‖ ≤ δ) → _root_.GlobalTheory.ClassFin.IsLog H' (A + Y) → Complex.re (Matrix.trace (H' * H')) ≤ Complex.re (Matrix.trace (H * H)) + (2 : ℝ) * Complex.re (Matrix.trace (H * A⁻¹ * Y)) + C * ∑ i : ι, ∑ j : ι, ‖Y i j‖ ^ (2 : ℕ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HermitianLogTraceSqFirstVariationBoundStatement

theorem **_root_.GlobalTheory.T.hermitian_log_trace_sq_first_variation_bound** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (H A : Matrix ι ι ℂ), IsLog H A → ∃ C δ : ℝ, 0 < δ ∧ ∀ Y H' : Matrix ι ι ℂ, (∀ i j, ‖Y i j‖ ≤ δ) → IsLog H' (A + Y) → (Matrix.trace (H' * H')).re ≤ (Matrix.trace (H * H)).re + 2 * (Matrix.trace (H * A⁻¹ * Y)).re + C * ∑ i, ∑ j, ‖Y i j‖ ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HigherUnitsLeSquaresHigherUnits

private lemma **exp_neg_two_mul_le_mul_of_lt** {e : ℕ} {a b : ℤᵐ⁰} (ha : WithZero.exp (-((e : ℤ) + 1)) < a) (hb : WithZero.exp (-((e : ℤ) + 1)) < b) : WithZero.exp (-(2 * (e : ℤ))) ≤ a * b

theorem **_root_.GlobalTheory.QNorm.hiUnits_le_sqs_hiUnits** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : ℕ), Valued.v (2 : v.adicCompletion F) = WithZero.exp (-(e : ℤ)) → _root_.GlobalTheory.QNorm.hiUnits (Valued.v : Valuation (v.adicCompletion F) ℤᵐ⁰) (2 * e + 1) ≤ _root_.GlobalTheory.QNorm.sqs (_root_.GlobalTheory.QNorm.hiUnits (Valued.v : Valuation (v.adicCompletion F) ℤᵐ⁰) (e + 1))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HigherUnitsLeSquaresHigherUnitsStatement

theorem **_root_.GlobalTheory.T.higherUnits_le_squares_higherUnits** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : ℕ), Valued.v (2 : v.adicCompletion F) = WithZero.exp (-(e : ℤ)) → hiUnits (Valued.v : Valuation (v.adicCompletion F) ℤᵐ⁰) (2 * e + 1) ≤ sqs (hiUnits (Valued.v : Valuation (v.adicCompletion F) ℤᵐ⁰) (e + 1))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Hilbert90BaseChange

theorem **exists_isUnit_of_forall_maximal** {K A : Type} [Field K] [Infinite K] [CommRing A] [Algebra K A] [IsArtinianRing A] (θ : A →ₗ[K] A) (h : ∀ m : Ideal A, m.IsMaximal → ∃ c, θ c ∉ m) : ∃ c, IsUnit (θ c)

theorem **hilbert90_baseChange** : ∀ (F E K : Type) [Field F] [CharZero F] [Field E] [Algebra F E] [FiniteDimensional F E] [Field K] [Algebra F K] (τ : E ≃ₐ[F] E) (m : ℕ), orderOf τ = m → ∀ x : K ⊗[F] E, ∏ i ∈ Finset.range m, (Algebra.TensorProduct.map (AlgHom.id K K) (τ : E →ₐ[F] E))^[i] x = 1 → ∃ z : (K ⊗[F] E)ˣ, x * Algebra.TensorProduct.map (AlgHom.id K K) (τ : E →ₐ[F] E) z = z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Hilbert90BaseChangeStatement

theorem **_root_.GlobalTheory.T.hilbert90_baseChange'** : ∀ (F E K : Type) [Field F] [CharZero F] [Field E] [Algebra F E] [FiniteDimensional F E] [Field K] [Algebra F K] (τ : E ≃ₐ[F] E) (m : ℕ), orderOf τ = m → ∀ x : K ⊗[F] E, ∏ i ∈ Finset.range m, (Algebra.TensorProduct.map (AlgHom.id K K) (τ : E →ₐ[F] E))^[i] x = 1 → ∃ z : (K ⊗[F] E)ˣ, x * Algebra.TensorProduct.map (AlgHom.id K K) (τ : E →ₐ[F] E) z = z

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Hilbert90SUnitsKummer

theorem **hilbert90_sUnits_kummer** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → AdmP p F S → (∀ v, v ∉ S → v.valuation F δ = 1) → ∀ u : Eˣ, u ∈ sUnits E SE → ∏ i ∈ Finset.range p, σ^[i] (u : E) = 1 → ∃ y : Eˣ, y ∈ sUnits E SE ∧ (u : E) = σ y / y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Hilbert90SUnitsKummerStatement

theorem **_root_.GlobalTheory.T.hilbert90_sUnits_kummer'** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → AdmP p F S → (∀ v, v ∉ S → v.valuation F δ = 1) → ∀ u : Eˣ, u ∈ sUnits E SE → ∏ i ∈ Finset.range p, σ^[i] (u : E) = 1 → ∃ y : Eˣ, y ∈ sUnits E SE ∧ (u : E) = σ y / y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Hilbert90SUnitsQuadraticExtension

private lemma **eq_one_of_sq_eq_one_withZero_int** {r : WithZero (Multiplicative ℤ)} (h : r ^ 2 = 1) : r = 1

private lemma **valuation_algebraMap_eq_of_unit_at_under** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {δ : F} {θ : E} {σ : E ≃ₐ[F] E} (hq : _root_.GlobalTheory.QNorm.IsQuad δ θ σ) (w : HeightOneSpectrum (𝒪 E)) (h2 : (w.under (𝒪

F)).valuation F 2 = 1) (hδ : (w.under (𝒪 F)).valuation F δ = 1) (c : F) : w.valuation E (algebraMap F E c) = (w.under (𝒪 F)).valuation F c

theorem **_root_.GlobalTheory.QNorm.hilbert90_sUnits_quadratic_extension_qnorm** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), _root_.GlobalTheory.QNorm.IsQuad δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → _root_.GlobalTheory.QNorm.admitted F S → (∀ v, v ∉ S → v.valuation F δ = 1) → ∀ u : Eˣ, u ∈ _root_.GlobalTheory.QNorm.sUnits E SE → (u : E) * σ u = 1 → ∃ y : Eˣ, y ∈ _root_.GlobalTheory.QNorm.sUnits E SE ∧ (u : E) = y / σ y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Hilbert90SUnitsQuadraticExtensionStatement

theorem **_root_.GlobalTheory.T.hilbert90_sUnits_quadratic_extension_of_adm** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → admitted F S → (∀ v, v ∉ S → v.valuation F δ = 1) → ∀ u : Eˣ, u ∈ sUnits E SE → (u : E) * σ u = 1 → ∃ y : Eˣ, y ∈ sUnits E SE ∧ (u : E) = y / σ y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HilbertFinitenessLeftInvariantMatrixPoly

theorem **evC_ltr** (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : evC (ltr g f) y = evC f (g * y)

theorem **isInv_iff** (Γ : Set (Matrix ι ι ℂ)) (f : MvPolynomial (ι × ι) ℂ) : IsInv Γ f ↔ ∀ g ∈ Γ, ltr g f = f

theorem **ltr_homog** (g : Matrix ι ι ℂ) {f : MvPolynomial (ι × ι) ℂ} {n : ℕ} (hf : f.IsHomogeneous n) : (ltr g f).IsHomogeneous n

theorem **ltr_comp** (g : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) (n : ℕ) : ltr g (homogeneousComponent n f) = homogeneousComponent n (ltr g f)

theorem **isInv_comp** {Γ : Set (Matrix ι ι ℂ)} {f : MvPolynomial (ι × ι) ℂ} (hf : IsInv Γ f) (n : ℕ) : IsInv Γ (homogeneousComponent n f)

theorem **hilbert_finiteness** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ), IsReynolds Γ Rey → ∃ s : Finset (MvPolynomial (ι × ι) ℂ), (∀ F ∈ s, IsInv Γ F ∧ ∃ n, F.IsHomogeneous n) ∧ ∀ f, IsInv Γ f → f ∈ Algebra.adjoin ℂ (s : Set (MvPolynomial (ι × ι) ℂ))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HilbertFinitenessLeftInvariantMatrixPolyStatement

theorem **_root_.GlobalTheory.T.hilbert_finiteness_leftInvariantMatrixPoly** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ), IsReynolds Γ Rey → ∃ s : Finset (MvPolynomial (ι × ι) ℂ), (∀ F ∈ s, IsInv Γ F ∧ ∃ n, F.IsHomogeneous n) ∧ ∀ f, IsInv Γ f → f ∈ Algebra.adjoin ℂ (s : Set (MvPolynomial (ι × ι) ℂ))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HomSymplecticHalfFormAlgebraicLawsContinuous

theorem **sum_mul_left** {A : Type} [CommRing A] {m n : ℕ} (c : A →+* A) (P : Fin n → Fin n → A) (Q : Fin m → Fin m → A) (t : A) (x y : Matrix (Fin m) (Fin n) A) : (∑ a, ∑ b, ∑ e, ∑ d, P a b * c (y e b) * Q e d * (x.map fun z => t * z) d a) = t * ∑ a, ∑ b, ∑ e, ∑ d, P a b * c (y e b) * Q e d * x d a

theorem **sum_mul_right** {A : Type} [CommRing A] {m n : ℕ} (c : A →+* A) (P : Fin n → Fin n → A) (Q : Fin m → Fin m → A) (t : A) (x y : Matrix (Fin m) (Fin n) A) : (∑ a, ∑ b, ∑ e, ∑ d, P a b * c ((y.map fun z => t * z) e b) * Q e d * x d a) = c t * ∑ a, ∑ b, ∑ e, ∑ d, P a b * c (y e b) * Q e d * x d a

theorem **hO_mul_left** (t : Formalism.L2.Theta.AE D) (x y : WO S T) : hO S T (x.map fun z => t * z) y = t * hO S T x y

theorem **hO_mul_right** (t : Formalism.L2.Theta.AE D) (x y : WO S T) : hO S T x (y.map fun z => t * z) = cA D t * hO S T x y

theorem **hO_rat** (M N : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hO S T (M.map (incA D)) (N.map (incA D)) = incA D (GlobalTheory.LatVac.hRat S T M N)

theorem **symplOH_rat** (M N : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : symplOH S T (M.map (incA D)) (N.map (incA D)) = algebraMap D.host.F (AF D.host.F) (1 / 2 * Algebra.trace D.host.F D.host.E (D.host.xi0 * GlobalTheory.LatVac.hRat S T M N))

theorem **hO_half** (e : Formalism.L2.Theta.AE D) (u : WO S T) (he : e * cA D e = 0) (hu : IsHalf e u) : hO S T u u = 0

theorem **symplOH_laws** : ∀ (D : D7) (S T : HSp D), (∀ x y z : WO S T, symplOH S T (x + y) z = symplOH S T x z + symplOH S T y z) ∧ (∀ x y z : WO S T, symplOH S T x (y + z) = symplOH S T x y + symplOH S T x z) ∧ (∀ x : WO S T, symplOH S T x x = 0) ∧ (∀ x y : WO S T, symplOH S T x y + symplOH S T x y = symplO S T x y) ∧ Continuous (fun q : WO S T × WO S T => symplOH S T q.1 q.2) ∧ (∀ g ∈ ratL T S, ∀ g' ∈ ratL T S, ∃ a : D.host.F, symplOH S T g g' = algebraMap D.host.F (AF D.host.F) a) ∧ (∀ (a : AE D) (x y : WO S T), symplOH S T (x.map fun z => a * z) y = symplOH S T x (y.map fun z => cA D a * z)) ∧ (∀ (e a : AE D) (u : WO S T), e * cA D e = 0 → IsHalf e u → symplOH S T (u.map fun z => a * z) u = 0) ∧ ∀ x y : WO S T, x.map (finPart D) = 0 → y.map (infPart D) = 0 → symplOH S T x y = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.HomSymplecticHalfFormAlgebraicLawsContinuousStatement

theorem **_root_.GlobalTheory.T.homSymplecticHalfForm_algebraic_laws_continuous** : ∀ (D : D7) (S T : HSp D), (∀ x y z : WO S T, symplOH S T (x + y) z = symplOH S T x z + symplOH S T y z) ∧ (∀ x y z : WO S T, symplOH S T x (y + z) = symplOH S T x y + symplOH S T x z) ∧ (∀ x : WO S T, symplOH S T x x = 0) ∧ (∀ x y : WO S T, symplOH S T x y + symplOH S T x y = symplO S T x y) ∧ Continuous (fun q : WO S T × WO S T => symplOH S T q.1 q.2) ∧ (∀ g ∈ ratL T S, ∀ g' ∈ ratL T S, ∃ a : D.host.F, symplOH S T g g' = algebraMap D.host.F (AF D.host.F) a) ∧ (∀ (a : AE D) (x y : WO S T), symplOH S T (x.map fun z => a * z) y = symplOH S T x (y.map fun z => cA D a * z)) ∧ (∀ (e a : AE D) (u : WO S T), e * cA D e = 0 → IsHalf e u → symplOH S T (u.map fun z => a * z) u = 0) ∧ ∀ x y : WO S T, x.map (finPart D) = 0 → y.map (infPart D) = 0 → symplOH S T x y = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IdealCharacterLContinuationNeZeroOne

private theorem **two_le_absNorm_heightOneSpectrum** {K : Type} [Field K] [NumberField K] (P : HeightOneSpectrum (𝒪 K)) : 2 ≤ Ideal.absNorm P.asIdeal

private theorem **one_sub_mul_natCast_cpow_neg_ne_zero** {c : ℂ} (hc : ‖c‖ ≤ 1) {n : ℕ} (hn : 2 ≤ n) {s : ℂ} (hs : 0 < s.re) : 1 - c * (n : ℂ) ^ (-s) ≠ 0

theorem **_root_.GlobalTheory.idealCharacterL_continuation_ne_zero_one** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (X : Ideal (𝒪 K) → ℂ) (φ : mixedSpace K → ℂ), m ≠ ⊥ → X ⊥ = 0 → (∀ I J, X (I * J) = X I * X J) → (∀ I, I ≠ ⊥ → I ⊔ m = ⊤ → ‖X I‖ = 1) → (∀ I, I ⊔ m ≠ ⊤ → X I = 0) → (∀ x y, φ (x * y) = φ x * φ y) → (∀ (t : ℝ) x, 0 < t → φ (t • x) = φ x) → ContDiffOn ℝ ∞ φ {x | mixedEmbedding.norm x ≠ 0} → (∀ x, mixedEmbedding.norm x ≠ 0 → ‖φ x‖ = 1) → (∀ β : 𝒪 K, β ≠ 0 → β - 1 ∈ m → X (Ideal.span {β}) = φ (mixedEmbedding K β)) → (∃ J, J ≠ ⊥ ∧ J ⊔ m = ⊤ ∧ X J ≠ 1) → ∀ c : HeightOneSpectrum (𝒪 K) → ℂ, (∀ P, ‖c P‖ ≤ 1) → (∀ P, ¬ m ≤ P.asIdeal → c P = X P.asIdeal) → ∃ L : ℂ → ℂ, DifferentiableOn ℂ L {s | 0 < s.re} ∧ (∀ s : ℂ, 1 < s.re → L s = Complex.exp (∑' P : HeightOneSpectrum (𝒪 K), -Complex.log (1 - c P * ((Ideal.absNorm P.asIdeal : ℕ) : ℂ) ^ (-s)))) ∧ L 1 ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IdealCharacterLSeriesContinuation

private theorem **eq_of_norm_sub_div_le** {f : ℝ → ℂ} {R R' : ℂ} {C C' : ℝ} (h : ∀ t : ℝ, 0 < t → t ≤ 1 → ‖f t - R / t‖ ≤ C) (h' : ∀ t : ℝ, 0 < t → t ≤ 1 → ‖f t - R' / t‖ ≤ C') : R = R'

private theorem **exists_differentiable_eq_div_of_family** {ι : Type} (M G : ι → ℂ → ℂ) (F : ℂ → ℂ) (hM : ∀ i, Differentiable ℂ (M i)) (hG : ∀ i, Differentiable ℂ (G i)) (hGF : ∀ i s, 1 < s.re → G i s = M i s * F s) (hne : ∀ s, ∃ i, M i s ≠ 0) : ∃ Z : ℂ → ℂ, Differentiable ℂ Z ∧ (∀ s, 1 < s.re → Z s = F s) ∧ ∀ i s, M i s ≠ 0 → Z s = G i s / M i s

private theorem **differentiable_mellin_of_support_in_Icc_pos** {w : ℝ → ℂ} (hw : Continuous w) {a b : ℝ} (ha : 0 < a) (hab : ∀ y, w y ≠ 0 → a ≤ y ∧ y ≤ b) : Differentiable ℂ (mellin w)

private theorem **exists_weight_mellin_ne_zero** (s₀ : ℂ) : ∃ w : ℝ → ℂ, ContDiff ℝ ∞ w ∧ (∃ a b : ℝ, 0 < a ∧ ∀ y, w y ≠ 0 → a ≤ y ∧ y ≤ b) ∧ mellin w s₀ ≠ 0

theorem **_root_.GlobalTheory.idealCharacter_LSeries_continuation** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (X : Ideal (𝒪 K) → ℂ) (φ : mixedSpace K → ℂ), m ≠ ⊥ → X ⊥ = 0 → (∀ I J, X (I * J) = X I * X J) → (∀ I, I ≠ ⊥ → I ⊔ m = ⊤ → ‖X I‖ = 1) → (∀ I, I ⊔ m ≠ ⊤ → X I = 0) → (∀ x y, φ (x * y) = φ x * φ y) → (∀ (t : ℝ) x, 0 < t → φ (t • x) = φ x) → ContDiffOn ℝ ∞ φ {x | mixedEmbedding.norm x ≠ 0} → (∀ x, mixedEmbedding.norm x ≠ 0 → ‖φ x‖ = 1) → (∀ β : 𝒪 K, β ≠ 0 → β - 1 ∈ m → X (Ideal.span {β}) = φ (mixedEmbedding K β)) → ∃ Z : ℂ → ℂ, Differentiable ℂ Z ∧ (∀ s : ℂ, 1 < s.re → Z s = (s - 1) * ∑' I, X I * ((Ideal.absNorm I : ℕ) : ℂ) ^ (-s)) ∧ ((∃ J, J ≠ ⊥ ∧ J ⊔ m = ⊤ ∧ X J ≠ 1) → Z 1 = 0)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IdealSumPeriodicDilate

theorem **tsum_ideal_eq_sum_classes** {R : Type} [CommRing R] (m b : Ideal R) [Fintype (R ⧸ m)] (hbm : b ⊔ m = ⊤) (s : R ⧸ m → R) (hsb : ∀ r, s r ∈ b) (hs : ∀ r, Ideal.Quotient.mk m (s r) = r) (ψ g : R → ℂ) (hψ : ∀ a b, a - b ∈ m → ψ a = ψ b) (hg : Summable fun α : b => ψ α * g α) : ∑' α : b, ψ α * g α = ∑ r : R ⧸ m, ψ (s r) * ∑' β : ↥(b * m), g ((β : R) + s r)

theorem **finite_support_dilate** (K : Type) [Field K] [NumberField K] (f : mixedSpace K → ℂ) (hf : HasCompactSupport f) (t : ℝ) (ht : 0 < t) : {α : 𝒪 K | f (t • mixedEmbedding K (α : K)) ≠ 0}.Finite

theorem **tsum_idealLattice** (K : Type) [Field K] [NumberField K] (J : (Ideal (𝒪 K))⁰) (F : mixedSpace K → ℂ) : ∑' l : idealLattice K (FractionalIdeal.mk0 K J), F l = ∑' β : ↥(J.1), F (mixedEmbedding K ((β : 𝒪 K) : K))

theorem **ideal_sum_periodic_dilate** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (ψ : 𝒪 K → ℂ) (f : mixedSpace K → ℂ), m ≠ ⊥ → (∀ a b, a - b ∈ m → ψ a = ψ b) → ContDiff ℝ ∞ f → HasCompactSupport f → ∃ κ : ℂ, ∀ b : Ideal (𝒪 K), b ≠ ⊥ → b ⊔ m = ⊤ → ∀ A : ℕ, ∃ C : ℝ, ∀ t : ℝ, 0 < t → t ≤ 1 → ‖∑' α : b, ψ α * f (t • mixedEmbedding K ((α : 𝒪 K) : K)) - κ / (((Ideal.absNorm b : ℝ) * t ^ Module.finrank ℚ K : ℝ) : ℂ)‖ ≤ C * t ^ A

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IdeleCongruenceUnitCriterion

theorem **exists_level** (U : Set (FiniteAdeleRing (𝒪 K) K)) (hU : U ∈ 𝒩 (0 : FiniteAdeleRing (𝒪 K) K)) : ∃ M : ℕ, M ≠ 0 ∧ ∀ a : FiniteAdeleRing (𝒪 K) K, (∀ P : HeightOneSpectrum (𝒪 K), Valued.v (a P) ≤ P.intValuation (M : 𝒪 K)) → a ∈ U

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IdempotentAdelesAndSmallAdelicMatrices

theorem **exists_dvd_pow_mul_coprime** (N m : ℕ) (hm : m ≠ 0) : ∃ t k : ℕ, t.Coprime N ∧ m ∣ N ^ k * t

theorem **exists_indicator** (S : Finset (HeightOneSpectrum (𝒪 F))) : ∃ e : FiniteAdeleRing (𝒪 F) F, (∀ v ∈ S, e v = 1) ∧ ∀ v ∉ S, e v = 0

theorem **exists_nsmul_integral** (b : FiniteAdeleRing (𝒪 F) F) : ∃ m : ℕ, m ≠ 0 ∧ ∀ v, (m • b) v ∈ v.adicCompletionIntegers F

theorem **exists_coprime_nsmul_integral** (N : ℕ) (b : FiniteAdeleRing (𝒪 F) F) (hb : ∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → b v = 0) : ∃ t : ℕ, t.Coprime N ∧ ∀ v, (t • b) v ∈ v.adicCompletionIntegers F

theorem **fm_add** (A B : Matrix (Fin n) (Fin n) E) : fm F (A + B) = fm F A + fm F B

theorem **fm_zero** : fm F (0 : Matrix (Fin n) (Fin n) E) = 0

theorem **nsmul_tmul_one** (d : ℕ) (a : FiniteAdeleRing (𝒪 F) F) : ((d • a) ⊗ₜ[F] (1 : E) : RA F E) = d • (a ⊗ₜ[F] (1 : E) : RA F E)

theorem **exists_approx** (S : Set (Fin n → E)) (hS : ∀ M : Matrix (Fin n) (Fin n) (RA F E), ∃ d : ℕ, 0 < d ∧ ∀ z ∈ hat F S, d • (M *ᵥ z) ∈ hat F S) (M : Matrix (Fin n) (Fin n) (RA F E)) (D : ℕ) (hD : 0 < D) : ∃ M₀ : Matrix (Fin n) (Fin n) E, ∀ z ∈ hat F S, ∃ w ∈ hat F S, (M - fm F M₀) *ᵥ z = D • w

theorem **exists_coprime_small** (S : Set (Fin n → E)) (N D : ℕ) (e : FiniteAdeleRing (𝒪 F) F) (M : Matrix (Fin n) (Fin n) (RA F E)) (hD : 0 < D) (hM : ∃ d : ℕ, 0 < d ∧ ∀ z ∈ hat F S, d • (M *ᵥ z) ∈ hat F S) (he : ∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → e v = 0) : ∃ t : ℕ, t.Coprime N ∧ ∀ z ∈ hat F S, ∃ w ∈ hat F S, t • (((e ⊗ₜ[F] (1 : E) : RA F E) • M) *ᵥ z) = D • w

theorem **GlobalTheory.UnRep.idempotent_adeles_small_matrices** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ), ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → (∀ S : Finset (HeightOneSpectrum (𝒪 F)), ∃ e : FiniteAdeleRing (𝒪 F) F, (∀ v ∈ S, e v = 1) ∧ ∀ v ∉ S, e v = 0) ∧ (∀ M : Matrix (Fin n) (Fin n) (RA F E), ∃ d : ℕ, 0 < d ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), d • (M *ᵥ z) ∈ hat F (L : Set (Fin n → E))) ∧ (∀ (M : Matrix (Fin n) (Fin n) (RA F E)) (D : ℕ), 0 < D → ∃ M₀ : Matrix (Fin n) (Fin n) E, ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), (M - fm F M₀) *ᵥ z = D • w) ∧ (∀ (N D : ℕ) (e : FiniteAdeleRing (𝒪 F) F) (M : Matrix (Fin n) (Fin n) (RA F E)), 0 < D → (∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → e v = 0) → ∃ t : ℕ, t.Coprime N ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IdempotentAdelesAndSmallAdelicMatricesStatement

theorem **_root_.GlobalTheory.T.idempotent_adeles_and_small_adelic_matrices** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ), ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → (∀ S : Finset (HeightOneSpectrum (𝒪 F)), ∃ e : FiniteAdeleRing (𝒪 F) F, (∀ v ∈ S, e v = 1) ∧ ∀ v ∉ S, e v = 0) ∧ (∀ M : Matrix (Fin n) (Fin n) (RA F E), ∃ d : ℕ, 0 < d ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), d • (M *ᵥ z) ∈ hat F (L : Set (Fin n → E))) ∧ (∀ (M : Matrix (Fin n) (Fin n) (RA F E)) (D : ℕ), 0 < D → ∃ M₀ : Matrix (Fin n) (Fin n) E, ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), (M - fm F M₀) *ᵥ z = D • w) ∧ (∀ (N D : ℕ) (e : FiniteAdeleRing (𝒪 F) F) (M : Matrix (Fin n) (Fin n) (RA F E)), 0 < D → (∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → e v = 0) → ∃ t : ℕ, t.Coprime N ∧ ∀ z ∈ hat F (L : Set (Fin

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.Imports

lemma **exists_nat_mul_mem** {E : Type} [Field E] [NumberField E] (c : E) : ∃ m : ℕ, 0 < m ∧ IsIntegral ℤ ((m : E) * c)

theorem **main** {E : Type} [Field E] {n : ℕ} [NumberField E] (L : Submodule (RingOfIntegers E) (Fin n → E)) : Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ x : Fin n → E, ∃ d : ℕ, 0 < d ∧ d • x ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ImportsBis

theorem **exists_nat_mul_integral** {E : Type} [Field E] [NumberField E] (a : E) : ∃ M : ℕ, 0 < M ∧ IsIntegral ℤ ((M : E) * a)

theorem **exists_nsmul_mem** {E : Type} [inst : Field E] [NumberField E] {n : ℕ} (Λ : Submodule (NumberField.RingOfIntegers E) (Fin n → E)) (h : Submodule.IsLattice E Λ) (v : Fin n → E) : ∃ (N : ℕ), (0 : ℕ) < N ∧ N • v ∈ Λ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ImportsTer

theorem **exists_nat_mul_integral** {E : Type} [Field E] [NumberField E] (c : E) : ∃ N : ℕ, 0 < N ∧ ∃ a : 𝒪 E, (a : E) = (N : E) * c

theorem **exists_nat_smul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hL : Submodule.span E (↑L : Set (Fin n → E)) = ⊤) (y : Fin n → E) : ∃ N : ℕ, 0 < N ∧ N • y ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexFormulaNormsInvolution

theorem **two_mul_index_sqs_mul_relIndex_norm_involution** : ∀ (A : Type) [CommGroup A] (σ : A →* A) (ε : A), (∀ a, σ (σ a) = a) → (∀ a, a * σ a = 1 → ∃ b, a = b / σ b) → ε ≠ 1 → ε ^ 2 = 1 → σ ε = ε → (∀ a : A, a ^ 2 = 1 → a = 1 ∨ a = ε) → 2 * (sqs (⊤ : Subgroup A)).index * (MonoidHom.id A * σ).range.relIndex (σ.eqLocus (MonoidHom.id A)) = (sqs (σ.eqLocus (MonoidHom.id A))).relIndex (σ.eqLocus (MonoidHom.id A)) ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexFormulaNormsInvolutionStatement

theorem **_root_.GlobalTheory.T.index_formula_norms_involution** : ∀ (A : Type) [CommGroup A] (σ : A →* A) (ε : A), (∀ a, σ (σ a) = a) → (∀ a, a * σ a = 1 → ∃ b, a = b / σ b) → ε ≠ 1 → ε ^ 2 = 1 → σ ε = ε → (∀ a : A, a ^ 2 = 1 → a = 1 ∨ a = ε) → 2 * (sqs (⊤ : Subgroup A)).index * (MonoidHom.id A * σ).range.relIndex (σ.eqLocus (MonoidHom.id A)) = (sqs (σ.eqLocus (MonoidHom.id A))).relIndex (σ.eqLocus (MonoidHom.id A)) ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexRangeKerIterateEndomorphism

theorem **card_ker_comp_eq** {M : Type} [CommGroup M] (f g : M →* M) : Nat.card (f.comp g).ker = Nat.card g.ker * Nat.card (g.range.subgroupOf f.ker)

theorem **index_range_comp_mul** {M : Type} [CommGroup M] (f g : M →* M) : (f.comp g).range.index * Nat.card f.ker * Nat.card g.ker = f.range.index * g.range.index * Nat.card (f.comp g).ker

theorem **index_range_ker_pow_succ_endomorphism** {M : Type} [CommGroup M] (f : Monoid.End M) (hK : Nat.card (MonoidHom.ker f) ≠ 0) (k : ℕ) : Nat.card (MonoidHom.ker (f ^ (k + 1) : Monoid.End M)) ≠ 0 ∧ (MonoidHom.range (f ^ (k + 1) : Monoid.End M)).index * Nat.card (MonoidHom.ker f) ^ (k + 1) = (MonoidHom.range f).index ^ (k + 1) * Nat.card (MonoidHom.ker (f ^ (k + 1) : Monoid.End M))

theorem **index_ker_iterate** : ∀ (M : Type) [CommGroup M] (f : Monoid.End M) (k : ℕ), (MonoidHom.range (f ^ (k + 1) : Monoid.End M)).index ≠ 0 → Nat.card (MonoidHom.ker (f ^ (k + 1) : Monoid.End M)) ≠ 0 → (MonoidHom.range f).index ≠ 0 ∧ Nat.card (MonoidHom.ker f) ≠ 0 ∧ (MonoidHom.range (f ^ (k + 1) : Monoid.End M)).index * Nat.card (MonoidHom.ker f) ^ (k + 1) = (MonoidHom.range f).index ^ (k + 1) * Nat.card (MonoidHom.ker (f ^ (k + 1) : Monoid.End M))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexRangeKerIterateEndomorphismStatement

theorem **_root_.GlobalTheory.T.index_range_ker_iterate_endomorphism** : ∀ (M : Type) [CommGroup M] (f : Monoid.End M) (k : ℕ), (MonoidHom.range (f ^ (k + 1) : Monoid.End M)).index ≠ 0 → Nat.card (MonoidHom.ker (f ^ (k + 1) : Monoid.End M)) ≠ 0 → (MonoidHom.range f).index ≠ 0 ∧ Nat.card (MonoidHom.ker f) ≠ 0 ∧ (MonoidHom.range (f ^ (k + 1) : Monoid.End M)).index * Nat.card (MonoidHom.ker f) ^ (k + 1) = (MonoidHom.range f).index ^ (k + 1) * Nat.card (MonoidHom.ker (f ^ (k + 1) : Monoid.End M))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexSqsTopOfFg

theorem **index_double_prod** (L T : Type) [AddCommGroup L] [AddCommGroup T] [Module.Free ℤ L] [Module.Finite ℤ L] [Finite T] : (nsmulAddMonoidHom (α

theorem **index_double_of_fg** (M : Type) [AddCommGroup M] [AddGroup.FG M] : (nsmulAddMonoidHom (α

theorem **index_sqs_top_of_fg** : ∀ (G : Type) [CommGroup G] [Group.FG G], (sqs (⊤ : Subgroup G)).index = 2 ^ Module.finrank ℤ (Additive G) * Nat.card {a : G // a ^ 2 = 1}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexSqsTopOfFgStatement

theorem **_root_.GlobalTheory.T.index_sqs_of_fg** : ∀ (G : Type) [CommGroup G] [Group.FG G], (sqs (⊤ : Subgroup G)).index = 2 ^ Module.finrank ℤ (Additive G) * Nat.card {a : G // a ^ 2 = 1}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexSquaresLeTwoMulUnitsSquares

theorem **mem_hiUnits_sup_sqs_of_even** {K : Type} [Field K] (w : Valuation K ℤᵐ⁰) (p : Kˣ) (hp : w (p : K) = WithZero.exp (-1)) (x : Kˣ) (m : ℤ) (hx : w (x : K) = WithZero.exp (2 * m)) : x ∈ hiUnits w 0 ⊔ sqs (⊤ : Subgroup Kˣ)

theorem **index_sqs_top_le** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (π : K), w π = WithZero.exp (-1) → (sqs (hiUnits w 0)).relIndex (hiUnits w 0) ≠ 0 → (sqs (⊤ : Subgroup Kˣ)).index ≠ 0 ∧ (sqs (⊤ : Subgroup Kˣ)).index ≤ 2 * (sqs (hiUnits w 0)).relIndex (hiUnits w 0)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexSquaresLeTwoMulUnitsSquaresStatement

theorem **_root_.GlobalTheory.T.index_squares_le_two_mul_units_squares** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (π : K), w π = WithZero.exp (-1) → (sqs (hiUnits w 0)).relIndex (hiUnits w 0) ≠ 0 → (sqs (⊤ : Subgroup Kˣ)).index ≠ 0 ∧ (sqs (⊤ : Subgroup Kˣ)).index ≤ 2 * (sqs (hiUnits w 0)).relIndex (hiUnits w 0)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexUnitsLocalSquaresFiniteRealPlacesLe

private theorem **isSquare_real_iff_nonneg** (r : ℝ) : IsSquare r ↔ 0 ≤ r

theorem **index_sqSub_real_dvd_two** {F : Type} [Field F] (σ : F →+* ℝ) : (sqSub σ).index ∣ 2

theorem **exists_valuation_two_eq_exp** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : ∃ e : ℕ, v.valuation F 2 = WithZero.exp (-(e : ℤ))

private theorem **index_biInf_ne_zero_and_le_prod** {G ι : Type} [Group G] [DecidableEq ι] (S : Finset ι) (f : ι → Subgroup G) : (∀ i ∈ S, (f i).index ≠ 0) → (⊓ i ∈ S, f i).index ≠ 0 ∧ (⊓ i ∈ S, f i).index ≤ ∏ i ∈ S, (f i).index

theorem **index_locSq_le** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), (locSq F S).index ≠ 0 ∧ (locSq F S).index ≤ 4 ^ (S.card + Fintype.card (InfinitePlace F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexUnitsLocalSquaresFiniteRealPlacesLeStatement

theorem **_root_.GlobalTheory.T.index_unitsLocalSquaresFiniteRealPlaces_le** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), (locSq F S).index ≠ 0 ∧ (locSq F S).index ≤ 4 ^ (S.card + Fintype.card (InfinitePlace F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexUnitsSquareUnderRingHomAdicCompletionLe

theorem **sqIdx_hiUnits_anti** {K : Type} [Field K] (w : Valuation K ℤᵐ⁰) {m n : ℕ} (h : m ≤ n) : hiUnits w n ≤ hiUnits w m

theorem **sqIdx_relIndex_hiUnits_add** {K : Type} [Field K] (w : Valuation K ℤᵐ⁰) (π : K) (R : Finset K) (hπ : w π = WithZero.exp (-1)) (hR : ∀ x : K, w x ≤ 1 → ∃ r ∈ R, w (x - r) < 1) (n k : ℕ) : (hiUnits w (n + k)).relIndex (hiUnits w n) ≠ 0 ∧ (hiUnits w (n + k)).relIndex (hiUnits w n) ≤ R.card ^ k

theorem **sqIdx_sqSub_eq_comap** {F L : Type} [Field F] [Field L] (φ : F →+* L) : sqSub φ = (sqs (⊤ : Subgroup Lˣ)).comap (Units.map (φ : F →* L))

theorem **index_sqSub_adicCompletion_le** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : ℕ), v.valuation F 2 = WithZero.exp (-(e : ℤ)) → (sqSub (algebraMap F (v.adicCompletion F))).index ≠ 0 ∧ (sqSub (algebraMap F (v.adicCompletion F))).index ≤ 4 * Ideal.absNorm v.asIdeal ^ e

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IndexUnitsSquareUnderRingHomAdicCompletionLeStatement

theorem **_root_.GlobalTheory.T.index_unitsSquareUnderRingHom_adicCompletion_le** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : ℕ), v.valuation F 2 = WithZero.exp (-(e : ℤ)) → (sqSub (algebraMap F (v.adicCompletion F))).index ≠ 0 ∧ (sqSub (algebraMap F (v.adicCompletion F))).index ≤ 4 * Ideal.absNorm v.asIdeal ^ e

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IntegralFiniteAdeleCompactClosedMultiples

theorem **t2Space_finiteAdeleRing'** : T2Space (FiniteAdeleRing (𝒪 F) F)

theorem **natCast_apply** (M : ℕ) (v : HeightOneSpectrum (𝒪 F)) : (M : FiniteAdeleRing (𝒪 F) F) v = ((algebraMap (𝒪 F) F (M : 𝒪 F) : F) : v.adicCompletion F)

theorem **mul_apply'** (a b : FiniteAdeleRing (𝒪 F) F) (v : HeightOneSpectrum (𝒪 F)) : (a * b) v = a v * b v

theorem **pow_apply'** (a : FiniteAdeleRing (𝒪 F) F) (n : ℕ) (v : HeightOneSpectrum (𝒪 F)) : (a ^ n) v = a v ^ n

theorem **eq_zero_of_forall_mul** (a : FiniteAdeleRing (𝒪 F) F) (h : ∀ M : ℕ, 0 < M → ∃ c : FiniteAdeleRing (𝒪 F) F, (∀ v, c v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ a = (M : FiniteAdeleRing (𝒪 F) F) * c) : a = 0

theorem **exists_nat_mul_integral** (a : FiniteAdeleRing (𝒪 F) F) : ∃ M : ℕ, 0 < M ∧ ∀ v, ((M : FiniteAdeleRing (𝒪 F) F) * a) v ∈ HeightOneSpectrum.adicCompletionIntegers F v

theorem **integral_finiteAdele_properties** : ∀ (F : Type) [Field F] [NumberField F], IsCompact {a : FiniteAdeleRing (𝒪 F) F | (∀ v, (a : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v)} ∧ (∀ M : ℕ, 0 < M → IsClosed {a : FiniteAdeleRing (𝒪 F) F | ∃ c : FiniteAdeleRing (𝒪 F) F, (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ a = (M : FiniteAdeleRing (𝒪 F) F) * c}) ∧ (∀ a : FiniteAdeleRing (𝒪 F) F, (∀ M : ℕ, 0 < M → ∃ c : FiniteAdeleRing (𝒪 F) F, (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ a = (M : FiniteAdeleRing (𝒪 F) F) * c) → a = 0) ∧ (∀ x : F, (∀ v, (algebraMap F (FiniteAdeleRing (𝒪 F) F) x : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) → ∃ y : 𝒪 F, x = y) ∧ (∀ a : FiniteAdeleRing (𝒪 F) F, ∃ M : ℕ, 0 < M ∧ (∀ v, (((M : FiniteAdeleRing (𝒪 F) F) * a) :

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IntegralFiniteAdeleCompactClosedMultiplesStatement

theorem **_root_.GlobalTheory.T.integralFiniteAdele_compact_closed_multiples** : ∀ (F : Type) [Field F] [NumberField F], IsCompact {a : FiniteAdeleRing (𝒪 F) F | (∀ v, (a : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v)} ∧ (∀ M : ℕ, 0 < M → IsClosed {a : FiniteAdeleRing (𝒪 F) F | ∃ c : FiniteAdeleRing (𝒪 F) F, (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ a = (M : FiniteAdeleRing (𝒪 F) F) * c}) ∧ (∀ a : FiniteAdeleRing (𝒪 F) F, (∀ M : ℕ, 0 < M → ∃ c : FiniteAdeleRing (𝒪 F) F, (∀ v, (c : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ a = (M : FiniteAdeleRing (𝒪 F) F) * c) → a = 0) ∧ (∀ x : F, (∀ v, (algebraMap F (FiniteAdeleRing (𝒪 F) F) x : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) → ∃ y : 𝒪 F, x = y) ∧ (∀ a : FiniteAdeleRing (𝒪 F) F, ∃ M : ℕ, 0 < M ∧ (∀ v, (((

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.InvariantFormEqSumInvariantTensors

theorem **invariant_form_eq_sum_invariant_tensors** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∀ (n : ℕ) (F : MvPolynomial (ι × ι) ℂ), IsInv Γ F → F.IsHomogeneous n → ∃ l : (Fin n → ι) → (Fin n → ι) → ℂ, (∀ b, ∀ g ∈ Γ, l b ᵥ* tpow n g = l b) ∧ ∀ y : Matrix ι ι ℂ, evC F y = ∑ b, (l b ᵥ* tpow n y) b

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.InvariantFormEqSumInvariantTensorsStatement

theorem **_root_.GlobalTheory.T.invariant_form_eq_sum_invariant_tensors'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∀ (n : ℕ) (F : MvPolynomial (ι × ι) ℂ), IsInv Γ F → F.IsHomogeneous n → ∃ l : (Fin n → ι) → (Fin n → ι) → ℂ, (∀ b, ∀ g ∈ Γ, l b ᵥ* tpow n g = l b) ∧ ∀ y : Matrix ι ι ℂ, evC F y = ∑ b, (l b ᵥ* tpow n y) b

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.InvariantSubspaceNotSemisimple

theorem **exists_mem_mul_eq_zero_of_isNilpotent** {R : Type*} [Ring R] (J : Ideal R) [J.IsTwoSided] (hJ : IsNilpotent J) (h0 : J ≠ ⊥) : ∃ a ∈ J, a ≠ 0 ∧ ∀ b ∈ J, b * a = 0

theorem **map_mem_jacobson_of_surjective** {R S : Type*} [Ring R] [Ring S] (f : R →+* S) (hf : Function.Surjective f) {b : R} (hb : b ∈ Ring.jacobson R) : f b ∈ Ring.jacobson S

theorem **map_mem_spanAlg** {ι : Type} [Fintype ι] [DecidableEq ι] {Γ : Set (Matrix ι ι ℂ)} (h : IsPre Γ) (σ : ℂ ≃+* ℂ) {x : Matrix ι ι ℂ} (hx : x ∈ spanAlg h) : x.map σ ∈ spanAlg h

theorem **spanAlgMap_surjective** {ι : Type} [Fintype ι] [DecidableEq ι] {Γ : Set (Matrix ι ι ℂ)} (h : IsPre Γ) (σ : ℂ ≃+* ℂ) : Function.Surjective (spanAlgMap h σ)

theorem **ss_of_isSemisimpleRing** {ι : Type} [Fintype ι] [DecidableEq ι] {Γ : Set (Matrix ι ι ℂ)} (h : IsPre Γ) [IsSemisimpleRing (spanAlg h)] : SS Γ

theorem **exists_socle_radical_of_not_ss** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ¬ SS Γ → ∃ W : Submodule ℂ (ι → ℂ), (∀ g ∈ Γ, ∀ v ∈ W, g *ᵥ v ∈ W) ∧ (∀ σ : ℂ ≃+* ℂ, ∀ v ∈ W, (fun i => σ (v i)) ∈ W) ∧ ∃ a ∈ Submodule.span ℂ Γ, a ≠ 0 ∧ (∀ v, a *ᵥ v ∈ W) ∧ ∀ v ∈ W, a *ᵥ v = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.InvariantSubspaceNotSemisimpleStatement

theorem **_root_.GlobalTheory.T.invariant_subspace_not_semisimple** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ¬ SS Γ → ∃ W : Submodule ℂ (ι → ℂ), (∀ g ∈ Γ, ∀ v ∈ W, g *ᵥ v ∈ W) ∧ (∀ σ : ℂ ≃+* ℂ, ∀ v ∈ W, (fun i => σ (v i)) ∈ W) ∧ ∃ a ∈ Submodule.span ℂ Γ, a ≠ 0 ∧ (∀ v, a *ᵥ v ∈ W) ∧ ∀ v ∈ W, a *ᵥ v = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsAdicCompleteAndCardQuotientAdicCompletionInteg

theorem **mem_span_singleton_adicCompletionIntegers** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) {t : v.adicCompletionIntegers F} (ht : Valued.v (t : v.adicCompletion F) ≠ 0) (x : v.adicCompletionIntegers F) : x ∈ Ideal.span {t} ↔ Valued.v (x : v.adicCompletion F) ≤ Valued.v (t : v.adicCompletion F)

theorem **isAdicComplete_and_card_quotient_adicCompletionIntegers** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), (∀ t : v.adicCompletionIntegers F, Valued.v (t : v.adicCompletion F) < 1 → IsAdicComplete (Ideal.span {t}) (v.adicCompletionIntegers F)) ∧ ∀ (t : v.adicCompletionIntegers F) (e : ℕ), Valued.v (t : v.adicCompletion F) = WithZero.exp (-(e : ℤ)) → Nat.card (v.adicCompletionIntegers F ⧸ Ideal.span {t}) = Ideal.absNorm v.asIdeal ^ e

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsAdicCompleteSpanAlgebraMapOfFreeFinite

theorem **mem_smul_top_iff_repr** {R M ι : Type} [CommRing R] [AddCommGroup M] [Module R M] [Fintype ι] (b : Module.Basis ι R M) (J : Ideal R) (x : M) : x ∈ (J • ⊤ : Submodule R M) ↔ ∀ i, b.repr x i ∈ J

theorem **isAdicComplete_of_free_finite** {R M : Type} [CommRing R] [AddCommGroup M] [Module R M] [Module.Free R M] [Module.Finite R M] (I : Ideal R) (h : IsAdicComplete I R) : IsAdicComplete I M

theorem **isAdicComplete_span_algebraMap_of_free_finite** : ∀ (R O : Type) [CommRing R] [CommRing O] [Algebra R O] [Module.Free R O] [Module.Finite R O] (t : R), IsAdicComplete (Ideal.span {t}) R → IsAdicComplete (Ideal.span {algebraMap R O t}) O

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsAutStableSLMatrixGroupImageBlockDiagPart

private lemma **idempotent_corner_part_mul** {A : Type} [Ring A] (E g h : A) (hE : E * E = E) (hg : (1 - E) * g * E = 0) (hh : (1 - E) * h * E = 0) : (E * g * E + (1 - E) * g * (1 - E)) * (E * h * E + (1 - E) * h * (1 - E)) = E * (g * h) * E + (1 - E) * (g * h) * (1 - E)

private lemma **idempotent_corner_sum_one** {A : Type} [Ring A] (E : A) (hE : E * E = E) : E * 1 * E + (1 - E) * 1 * (1 - E) = 1

private lemma **idempotent_corner_part_inv** {A : Type} [Ring A] (E g h : A) (hE : E * E = E) (hg : (1 - E) * g * E = 0) (hh : (1 - E) * h * E = 0) (hgh : g * h = 1) : g * (E * h * E + (1 - E) * h * (1 - E)) = 1 + E * (-(g * E * h) * (1 - E))

theorem **isPre_blk_image** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsPre Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → IsPre (blk e '' Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsAutStableSLMatrixGroupImageBlockDiagPartStatement

theorem **_root_.GlobalTheory.T.isAutStableSLMatrixGroup_image_blockDiagPart** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (e : Matrix ι ι ℚ), IsPre Γ → e * e = e → (∀ g ∈ Γ, (1 - e.map (algebraMap ℚ ℂ)) * g * e.map (algebraMap ℚ ℂ) = 0) → IsPre (blk e '' Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsClosedImageSLUnderGeneratingInvariants

theorem **evC_aeval_invariants** {κ : Type} (Φ : κ → MvPolynomial (ι × ι) ℂ) (P : MvPolynomial κ ℂ) (y : Matrix ι ι ℂ) : evC (aeval Φ P) y = eval (fun k => evC (Φ k) y) P

theorem **evC_det_generic** [Fintype ι] [DecidableEq ι] (y : Matrix ι ι ℂ) : evC (Matrix.of fun i j : ι => (X (i, j) : MvPolynomial (ι × ι) ℂ)).det y = y.det

theorem **exists_evC_eq_of_relations** [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ) (κ : Type) [Fintype κ] (Φ : κ → MvPolynomial (ι × ι) ℂ) (hR : IsReynolds Γ Rey) (hΦ : ∀ k, IsInv Γ (Φ k)) (hgen : ∀ f, IsInv Γ f → f ∈ Algebra.adjoin ℂ (Set.range Φ)) (z : κ → ℂ) (hz : ∀ P : MvPolynomial κ ℂ, aeval Φ P = 0 → eval z P = 0) : ∃ y : Matrix ι ι ℂ, z = fun k => evC (Φ k) y

theorem **isClosed_image_invariants** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ) (κ : Type) [Fintype κ] (Φ : κ → MvPolynomial (ι × ι) ℂ), IsReynolds Γ Rey → (∀ g ∈ Γ, g.det = 1) → (∀ k, IsInv Γ (Φ k)) → (∀ f, IsInv Γ f → f ∈ Algebra.adjoin ℂ (Set.range Φ)) → IsClosed {z : κ → ℂ | ∃ y : Matrix ι ι ℂ, y.det = 1 ∧ z = fun k => evC (Φ k) y}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsClosedImageSLUnderGeneratingInvariantsStatement

theorem **_root_.GlobalTheory.T.isClosed_image_SL_under_generating_invariants** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (Rey : MvPolynomial (ι × ι) ℂ →ₗ[ℂ] MvPolynomial (ι × ι) ℂ) (κ : Type) [Fintype κ] (Φ : κ → MvPolynomial (ι × ι) ℂ), IsReynolds Γ Rey → (∀ g ∈ Γ, g.det = 1) → (∀ k, IsInv Γ (Φ k)) → (∀ f, IsInv Γ f → f ∈ Algebra.adjoin ℂ (Set.range Φ)) → IsClosed {z : κ → ℂ | ∃ y : Matrix ι ι ℂ, y.det = 1 ∧ z = fun k => evC (Φ k) y}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsClosedIsHermitianLogOfPosDef

theorem **isClosed_isLog** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι], IsClosed {p : Matrix ι ι ℂ × Matrix ι ι ℂ | IsLog p.1 p.2}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsClosedIsHermitianLogOfPosDefStatement

theorem **_root_.GlobalTheory.T.isClosed_isHermitianLogOfPosDef** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι], IsClosed {p : Matrix ι ι ℂ × Matrix ι ι ℂ | IsLog p.1 p.2}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsClosedZeroLocusGL

theorem **continuous_aeval_comp** {F A X σ : Type} [CommRing F] [CommRing A] [Algebra F A] [TopologicalSpace A] [IsTopologicalRing A] [TopologicalSpace X] (f : X → σ → A) (hf : ∀ i, Continuous fun x => f x i) (P : MvPolynomial σ F) : Continuous fun x => MvPolynomial.aeval (f x) P

theorem **isClosed_zeroLocusGL** {F : Type} [Field F] {n : ℕ} (S : Set (MvPolynomial (Fin n × Fin n) F)) (A : Type) [CommRing A] [Algebra F A] [TopologicalSpace A] [IsTopologicalRing A] [T2Space A] : IsClosed (HeckeCharacterZeroLocus.zeroLocusGL S A)

theorem **exists_isOpen_isCompact** : ∀ (F : Type) [Field F] [NumberField F] (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)) (hS : HeckeCharacterZeroLocus.IsGroupLocus S), ∃ U : Subgroup (HeckeCharacterZeroLocus.locusSubgroup S hS (FiniteAdeleRing (𝒪 F) F)), IsOpen

(U : Set (HeckeCharacterZeroLocus.locusSubgroup S hS (FiniteAdeleRing (𝒪 F) F))) ∧ IsCompact (U : Set (HeckeCharacterZeroLocus.locusSubgroup S hS (FiniteAdeleRing (𝒪 F) F)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsCompactIntegralPointsFiniteAdelicAlgebraicGrou

theorem **continuous_aeval_family** {F A X σ : Type} [CommRing F] [CommRing A] [Algebra F A] [TopologicalSpace A] [IsTopologicalRing A] [TopologicalSpace X] (f : X → σ → A) (hf : ∀ s, Continuous fun x => f x s) (P : MvPolynomial σ F) : Continuous fun x => MvPolynomial.aeval (f x) P

theorem **isClosed_of_equations** {F A ι : Type} [CommRing F] [CommRing A] [Algebra F A] [TopologicalSpace A] [IsTopologicalRing A] [T2Space A] [Fintype ι] [DecidableEq ι] (S : Set (MvPolynomial (ι × ι) F)) (H : Subgroup (GL ι A)) (hH : ∀ g, g ∈ H ↔ ∀ P ∈ S, MvPolynomial.aeval (fun ij : ι × ι => ((g : GL ι A) : Matrix ι ι A) ij.1 ij.2) P = 0) : IsClosed (H : Set (GL ι A))

theorem **isCompact_integral_points** : ∀ (F : Type) [Field F] [NumberField F] (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)) (H : Subgroup (GL (Fin n) (FiniteAdeleRing (𝒪 F) F))), (∀ g, g ∈ H ↔ (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (FiniteAdeleRing (𝒪 F) F)) : Matrix (Fin n) (Fin n) (FiniteAdeleRing (𝒪 F) F)) ij.1 ij.2) P = 0)) → IsCompact {u : H | (∀ i j, (∀ v, (((u : GL (Fin n) (FiniteAdeleRing (𝒪 F) F)) : Matrix (Fin n) (Fin n) (FiniteAdeleRing (𝒪 F) F)) i j : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ (∀ v, ((((u : GL (Fin n) (FiniteAdeleRing (𝒪 F) F))⁻¹ : GL (Fin n) (FiniteAdeleRing (𝒪 F) F)) : Matrix (Fin n) (Fin n) (FiniteAdeleRing (𝒪 F) F)) i j : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v))}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsCompactIntegralPointsFiniteAdelicAlgebraicGrouStatement

theorem **_root_.GlobalTheory.T.isCompact_integralPoints_finiteAdelic_algebraicGroup** : ∀ (F : Type) [Field F] [NumberField F] (n : ℕ) (S : Set (MvPolynomial (Fin n × Fin n) F)) (H : Subgroup (GL (Fin n) (FiniteAdeleRing (𝒪 F) F))), (∀ g, g ∈ H ↔ (∀ P ∈ S, MvPolynomial.aeval (fun ij : Fin n × Fin n => ((g : GL (Fin n) (FiniteAdeleRing (𝒪 F) F)) : Matrix (Fin n) (Fin n) (FiniteAdeleRing (𝒪 F) F)) ij.1 ij.2) P = 0)) → IsCompact {u : H | (∀ i j, (∀ v, (((u : GL (Fin n) (FiniteAdeleRing (𝒪 F) F)) : Matrix (Fin n) (Fin n) (FiniteAdeleRing (𝒪 F) F)) i j : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v) ∧ (∀ v, ((((u : GL (Fin n) (FiniteAdeleRing (𝒪 F) F))⁻¹ : GL (Fin n) (FiniteAdeleRing (𝒪 F) F)) : Matrix (Fin n) (Fin n) (FiniteAdeleRing (𝒪 F) F)) i j : FiniteAdeleRing (𝒪 F) F) v ∈ HeightOneSpectrum.adicCompletionIntegers F v))}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsCompletelyReducibleKroneckerOfSpanEqTop

theorem **exists_finset_span_image_eq** {G V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (f : G → V) : ∃ T : Finset G, ∀ S : Set G, (T : Set G) ⊆ S → Submodule.span ℂ (f '' S) = Submodule.span ℂ (Set.range f)

theorem **ss_kronecker_of_span_top** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (κ : Type) [Fintype κ] [DecidableEq κ] (G : Type) [Group G] (ρ : G →* Matrix ι ι ℂ) (ρ' : G →* Matrix κ κ ℂ), Submodule.span ℂ (Set.range ρ) = ⊤ → Submodule.span ℂ (Set.range ρ') = ⊤ → SS (Set.range fun g => ρ g ⊗ₖ ρ' g)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsCompletelyReducibleKroneckerOfSpanEqTopStatement

theorem **_root_.GlobalTheory.T.isCompletelyReducible_kronecker_of_span_eq_top** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (κ : Type) [Fintype κ] [DecidableEq κ] (G : Type) [Group G] (ρ : G →* Matrix ι ι ℂ) (ρ' : G →* Matrix κ κ ℂ), Submodule.span ℂ (Set.range ρ) = ⊤ → Submodule.span ℂ (Set.range ρ') = ⊤ → SS (Set.range fun g => ρ g ⊗ₖ ρ' g)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsCompletelyReducibleMatrixSetTransposeImage

private theorem **mem_dotPerp** (U : Submodule ℂ (ι → ℂ)) (v : ι → ℂ) : v ∈ dotPerp U ↔ ∀ w ∈ U, v ⬝ᵥ w = 0

private theorem **dotPerp_isCompl** {U U' : Submodule ℂ (ι → ℂ)} (h : IsCompl U U') : IsCompl (dotPerp U) (dotPerp U')

private theorem **dotPerp_dotPerp** (U : Submodule ℂ (ι → ℂ)) : dotPerp (dotPerp U) = U

private theorem **dotPerp_invariant** (U : Submodule ℂ (ι → ℂ)) (g : Matrix ι ι ℂ) (h : ∀ v ∈ U, g *ᵥ v ∈ U) : ∀ v ∈ dotPerp U, gᵀ *ᵥ v ∈ dotPerp U

theorem **ss_transpose** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), SS Γ → SS (Matrix.transpose '' Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsCompletelyReducibleMatrixSetTransposeImageStatement

theorem **_root_.GlobalTheory.T.isCompletelyReducibleMatrixSet_transpose_image** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), SS Γ → SS (Matrix.transpose '' Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsFiniteAdeleLikePairElementary

theorem **AdelicPair.exists_mul_mem** {R : Type} [CommRing R] [Algebra ℚ R] {R₀ : Subring R} (h : AdelicPair R R₀) (r : R) : ∃ M : ℕ, 0 < M ∧ (M : R) * r ∈ R₀

theorem **AdelicPair.exists_mul_mem_family** {R : Type} [CommRing R] [Algebra ℚ R] {R₀ : Subring R} (h : AdelicPair R R₀) (κ : Type) [Fintype κ] (r : κ → R) : ∃ M : ℕ, 0 < M ∧ ∀ k, (M : R) * r k ∈ R₀

private lemma **exists_pos_nat_clearing_rat_denominators** {κ : Type} [Fintype κ] (q : κ → ℚ) : ∃ M : ℕ, 0 < M ∧ ∀ k, ∃ z : ℤ, (M : ℚ) * q k = z

theorem **_root_.GlobalTheory.ClassFin.adelicPair_elementary** : ∀ (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), _root_.GlobalTheory.ClassFin.AdelicPair R R₀ → (∀ M : ℕ, 0 < M → ∀ r ∈ R₀, ∃ z : ℤ, ∃ a ∈ R₀, r = (z : R) + (M : R) * a) ∧ (∀ (κ : Type) [Fintype κ] (r : κ → R), ∃ M : ℕ, 0 < M ∧ ∀ k, (M : R) * r k ∈ R₀) ∧ Function.Injective (algebraMap ℚ R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsLogKronecker

theorem **isLog_kronecker** {ι : Type} [Fintype ι] [DecidableEq ι] {κ : Type} [Fintype κ] [DecidableEq κ] {H A : Matrix ι ι ℂ} {H' A' : Matrix κ κ ℂ} (h : IsLog H A) (h' : IsLog H' A') : IsLog (H ⊗ₖ (1 : Matrix κ κ ℂ) + (1 : Matrix ι ι ℂ) ⊗ₖ H') (A ⊗ₖ A')

theorem **harm_kronecker** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (κ : Type) [Fintype κ] [DecidableEq κ] (σ : Type) [Fintype σ] (g : σ → Matrix ι ι ℂ) (g' : σ → Matrix κ κ ℂ), Harm g → Harm g' → Harm (fun s => g s ⊗ₖ g' s)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsNilpotentTmulSubTmulOfConstOnFibres

theorem **isNilpotent_of_forall_algHom_eq_zero** (T : Type) [CommRing T] [Algebra ℂ T] [Algebra.FiniteType ℂ T] (x : T) (h : ∀ ω : T →ₐ[ℂ] ℂ, ω x = 0) : IsNilpotent x

theorem **isNilpotent_tmul_sub_tmul_of_const_on_fibres** : ∀ (B A : Type) [CommRing B] [CommRing A] [Algebra ℂ B] [Algebra ℂ A] [Algebra B A] [IsScalarTower ℂ B A] [Algebra.FiniteType ℂ A] (f : A), (∀ χ χ' : A →ₐ[ℂ] ℂ, (∀ b : B, χ (algebraMap B A b) = χ' (algebraMap B A b)) → χ f = χ' f) → IsNilpotent (f ⊗ₜ[B] (1 : A) - (1 : A) ⊗ₜ[B] f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsNilpotentTmulSubTmulOfConstOnFibresStatement

theorem **_root_.GlobalTheory.T.isNilpotent_tmul_sub_tmul_of_const_on_fibres'** : ∀ (B A : Type) [CommRing B] [CommRing A] [Algebra ℂ B] [Algebra ℂ A] [Algebra B A] [IsScalarTower ℂ B A] [Algebra.FiniteType ℂ A] (f : A), (∀ χ χ' : A →ₐ[ℂ] ℂ, (∀ b : B, χ (algebraMap B A b) = χ' (algebraMap B A b)) → χ f = χ' f) → IsNilpotent (f ⊗ₜ[B] (1 : A) - (1 : A) ⊗ₜ[B] f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsNormFromBaseChangeBaseChangeProperties

theorem **tensor_repr_map** {ι : Type} (b : Module.Basis ι F E) {K K' : Type} [CommRing K] [Algebra F K] [CommRing K'] [Algebra F K'] (φ : K →ₐ[F] K') (y : K ⊗[F] E) (i : ι) : (Algebra.TensorProduct.basis K' b).repr (Algebra.TensorProduct.map φ (AlgHom.id F E) y) i = φ ((Algebra.TensorProduct.basis K b).repr y i)

theorem **tensor_norm_map** [FiniteDimensional F E] {K K' : Type} [CommRing K] [Algebra F K] [CommRing K'] [Algebra F K'] (φ : K →ₐ[F] K') (y : K ⊗[F] E) : Algebra.norm K' (Algebra.TensorProduct.map φ (AlgHom.id F E) y) = φ (Algebra.norm K y)

theorem **tensor_norm_one_tmul** [FiniteDimensional F E] (K : Type) [CommRing K] [Algebra F K] (e : E) : Algebra.norm K ((1 : K) ⊗ₜ[F] e) = algebraMap F K (Algebra.norm F e)

theorem **tnorm_baseChange** : ∀ (F E : Type) [Field F] [Field E] [Algebra F E] [FiniteDimensional F E], (∀ (K K' : Type) [Field K] [Algebra F K] [Field K'] [Algebra F K'] (_ : K →ₐ[F] K') (x : F), TNorm F E K x → TNorm F E K' x) ∧ (∀ (K : Type) [Field K] [Algebra F K] (e : E), TNorm F E K (Algebra.norm F e)) ∧ ∀ (K : Type) [Field K] [Algebra F K] (x y : F), y ≠ 0 → TNorm F E K (x * y) → TNorm F E K y → TNorm F E K x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsNormFromBaseChangeBaseChangePropertiesStatement

theorem **_root_.GlobalTheory.T.isNormFromBaseChange_base_change_properties** : ∀ (F E : Type) [Field F] [Field E] [Algebra F E] [FiniteDimensional F E], (∀ (K K' : Type) [Field K] [Algebra F K] [Field K'] [Algebra F K'] (_ : K →ₐ[F] K') (x : F), TNorm F E K x → TNorm F E K' x) ∧ (∀ (K : Type) [Field K] [Algebra F K] (e : E), TNorm F E K (Algebra.norm F e)) ∧ ∀ (K : Type) [Field K] [Algebra F K] (x y : F), y ≠ 0 → TNorm F E K (x * y) → TNorm F E K y → TNorm F E K x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsNormFromBaseChangeOfPowFinrankCyclic

theorem **iter_prod** {A ι : Type} [CommMonoid A] (T : A →* A) (n : ℕ) (s : Finset ι) (f : ι → A) : T^[n] (∏ i ∈ s, f i) = ∏ i ∈ s, T^[n] (f i)

theorem **prod_range_mul_iter** {A : Type} [CommMonoid A] (T : A →* A) (d m : ℕ) (x : A) : ∏ i ∈ Finset.range (d * m), T^[i] x = ∏ k ∈ Finset.range m, (T^[d])^[k] (∏ j ∈ Finset.range d, T^[j] x)

theorem **prod_cyc** {G A : Type} [Group G] [Fintype G] [CommMonoid A] (σ : G) (h : ∀ g, g ∈ Subgroup.zpowers σ) (f : G → A) : ∏ g, f g = ∏ i ∈ Finset.range (Nat.card G), f (σ ^ i)

theorem **telescope** {A : Type} [CommMonoid A] (T : A →* A) (d : ℕ) (z : A) : (∏ j ∈ Finset.range d, T^[j] (T z)) * z = (∏ j ∈ Finset.range d, T^[j] z) * T^[d] z

theorem **core** {A : Type} [CommMonoid A] (T : A →* A) (d m : ℕ) (x a b : A) (hab : a * b = 1) (hb : T b = b) (ha : T a = a) (hx : ∏ i ∈ Finset.range (d * m), T^[i] x = a ^ m) (h90 : ∀ u : A, ∏ k ∈ Finset.range m, (T^[d])^[k] u = 1 → ∃ z : Aˣ, u * T^[d] z = z) : ∃ x' : A, T^[d] x' = x' ∧ ∏ j ∈ Finset.range d, T^[j] x' = a

theorem **map_pow_iter** {F E K : Type} [Field F] [Field E] [Algebra F E] [CommRing K] [Algebra F K] (σ : E ≃ₐ[F] E) (i : ℕ) (y : K ⊗[F] E) : Algebra.TensorProduct.map (AlgHom.id K K) ((σ ^ i : E ≃ₐ[F] E) : E →ₐ[F] E) y = (Algebra.TensorProduct.map (AlgHom.id K K) (σ : E →ₐ[F] E))^[i] y

theorem **exists_pow_of_pow_eq_one** {G : Type} [Group G] [Finite G] (σ : G) (d m : ℕ) (hm : m ≠ 0) (h : ∀ g, g ∈ Subgroup.zpowers σ) (ho : orderOf σ = d * m) (g : G) (hg : g ^ m = 1) : ∃ k : ℕ, g = σ ^ (d * k)

theorem **tnorm_of_tnorm_pow** : ∀ (F M E K : Type) [Field F] [CharZero F] [Field M] [Field E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [FiniteDimensional F E] [IsGalois F E] [Field K] [Algebra F K], IsCyclic (E ≃ₐ[F] E) → ∀ a : F, a ≠ 0 → TNorm F E K (a ^ Module.finrank M E) → TNorm F M K a

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsOpenCongruenceIdelesAndRealFiniteParts

theorem **isOpen_nsmul_adelicCompletion_integers** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (N : ℕ) (hN : 0 < N) : letI : TopologicalSpace (FiniteAdeleRing (𝒪 F) F) := ⊗[F] E

theorem **integralFiniteAdeles_eq_adelicCompletion_integers** (D : D7) : intAE D = GlobalTheory.AdLat.hat1 D.host.F (Set.range (algebraMap (𝒪 D.host.E) D.host.E))

theorem **exists_units_parts** {R A B : Type} [Ring R] [Ring A] [Ring B] [TopologicalSpace R] [TopologicalSpace A] [TopologicalSpace B] (e : R ≃+* A × B) (hec : Continuous e) (hes : Continuous e.symm) (f : R → A) (g : R → B) (he : ∀ z, (e z).1 = f z ∧ (e z).2 = g z) : ∃ a b : Rˣ →* Rˣ, Continuous a ∧ Continuous b ∧ (∀ x, x = a x * b x) ∧ (∀ x, f (a x : R) = f (x : R) ∧ g (a x : R) = 1) ∧ ∀ x, f (b x : R) = 1 ∧ g (b x : R) = g (x : R)

theorem **congr_open_and_parts** : ∀ (D : D7), (∀ N : ℕ, 0 < N → IsOpen {x : (AE D)ˣ | finPart D (x : AE D) ∈ intAE D ∧ finPart D ((x⁻¹ : (AE D)ˣ) : AE D) ∈ intAE D ∧ ∃ y ∈ intAE D, finPart D (x : AE D) - 1 = N • y}) ∧ ∃ a b : (AE D)ˣ →* (AE D)ˣ, Continuous a ∧ Continuous b ∧ (∀ x, x = a x * b x) ∧ (∀ x, infPart D (a x : AE D) = infPart D (x : AE D) ∧ finPart D (a x : AE D) = 1) ∧ ∀ x, infPart D (b x : AE D) = 1 ∧ finPart D (b x : AE D) = finPart D (x : AE D)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsOpenCongruenceIdelesAndRealFinitePartsStatement

theorem **_root_.GlobalTheory.T.isOpen_congruence_ideles_and_real_finite_parts** : ∀ (D : D7), (∀ N : ℕ, 0 < N → IsOpen {x : (AE D)ˣ | finPart D (x : AE D) ∈ intAE D ∧ finPart D ((x⁻¹ : (AE D)ˣ) : AE D) ∈ intAE D ∧ ∃ y ∈ intAE D, finPart D (x : AE D) - 1 = N • y}) ∧ ∃ a b : (AE D)ˣ →* (AE D)ˣ, Continuous a ∧ Continuous b ∧ (∀ x, x = a x * b x) ∧ (∀ x, infPart D (a x : AE D) = infPart D (x : AE D) ∧ finPart D (a x : AE D) = 1) ∧ ∀ x, infPart D (b x : AE D) = 1 ∧ finPart D (b x : AE D) = finPart D (x : AE D)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsOpenNsmulCompletedHomLatticeFiniteAdelic

theorem **isOpen_of_nhds** {G G' : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [AddCommGroup G'] (φ : G →+ G') (L : AddSubgroup G) (N : ℕ) (U : Set G) (hU : U ∈ 𝒩 0) (h : ∀ x ∈ U, ∃ l ∈ L, φ (x - N • l) = 0) : IsOpen {x : G | ∃ l ∈ L, φ (x - N • l)

= 0}

theorem **isOpen_intAf** (F : Type) [Field F] [NumberField F] : IsOpen (intAf F)

theorem **zero_mem_intAf** (F : Type) [Field F] [NumberField F] : (0 : Af F) ∈ intAf F

theorem **finPart_tmul** (a : AF D.host.F) (x : D.host.E) : finPart D (a ⊗ₜ[D.host.F] x) = sndA D.host.F a ⊗ₜ[D.host.F] x

theorem **finPart_eq_sum** {ι : Type} [Fintype ι] (b : Module.Basis ι D.host.F D.host.E) (z : AE D) : finPart D z = ∑ k, sndA D.host.F ((Algebra.TensorProduct.basis (AF D.host.F) b).coord k z) ⊗ₜ[D.host.F] b k

theorem **nsmul_tmul_cancel** {F A E : Type} [Field F] [CharZero F] [CommRing A] [Algebra F A] [AddCommGroup E] [Module F E] {N M : ℕ} (hN : 0 < N) (hM : 0 < M) (a : A) (e : E) : N • ((algebraMap F A ((N * M : ℕ) : F)⁻¹ * a) ⊗ₜ[F] (M • e)) = a ⊗ₜ[F] e

theorem **isOpen_nsmul_latO** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ N : ℕ, 0 < N → IsOpen {x : WO S T | ∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), (x - N • l).map (finPart D) = 0}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsOpenNsmulCompletedHomLatticeFiniteAdelicStatement

theorem **_root_.GlobalTheory.T.isOpen_nsmul_completedHomLattice_finiteAdelic** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ N : ℕ, 0 < N → IsOpen {x : WO S T | ∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), (x - N • l).map (finPart D) = 0}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsQuadraticSqrtExtensionBasicFacts

theorem **IsQuad.conj_coord** (h : IsQuad δ θ σ) (a b : F) : σ (algebraMap F E a + algebraMap F E b * θ) = algebraMap F E a - algebraMap F E b * θ

theorem **_root_.GlobalTheory.QNorm.IsQuad.coord_indep** (h : _root_.GlobalTheory.QNorm.IsQuad δ θ σ) (a b : F) (hab : algebraMap F E a + algebraMap F E b * θ = 0) : a = 0 ∧ b = 0

theorem **_root_.GlobalTheory.QNorm.IsQuad.root_ne_zero** (h : _root_.GlobalTheory.QNorm.IsQuad δ θ σ) : θ ≠ 0

theorem **IsQuad.conj_conj** (h : IsQuad δ θ σ) (x : E) : σ (σ x) = x

theorem **IsQuad.finrank_eq_two** (h : IsQuad δ θ σ) : Module.finrank F E = 2

theorem **_root_.GlobalTheory.QNorm.isQuad_basic** : ∀ (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), _root_.GlobalTheory.QNorm.IsQuad δ θ σ → (∀ x, σ (σ x) = x) ∧ (∀ x, σ x = x → ∃ a : F, x = algebraMap F E a) ∧ (∀ a b : F, algebraMap F E a + algebraMap F E b * θ = 0 → a = 0 ∧ b = 0) ∧ (∀ x : E, ∃ a b : F, x * σ x = algebraMap F E (a ^ 2 - δ * b ^ 2)) ∧ (∀ x : E, x * σ x = 1 → ∃ y : E, y ≠ 0 ∧ x = y / σ y) ∧ (∀ u : F, IsSquare (algebraMap F E u) → IsSquare u ∨ IsSquare (u * δ)) ∧ Module.finrank F E = 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsQuadraticSqrtExtensionBasicFactsStatement

theorem **_root_.GlobalTheory.T.isQuadraticSqrtExtension_basic_facts** : ∀ (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → (∀ x, σ (σ x) = x) ∧ (∀ x, σ x = x → ∃ a : F, x = algebraMap F E a) ∧ (∀ a b : F, algebraMap F E a + algebraMap F E b * θ = 0 → a = 0 ∧ b = 0) ∧ (∀ x : E, ∃ a b : F, x * σ x = algebraMap F E (a ^ 2 - δ * b ^ 2)) ∧ (∀ x : E, x * σ x = 1 → ∃ y : E, y ≠ 0 ∧ x = y / σ y) ∧ (∀ u : F, IsSquare (algebraMap F E u) → IsSquare u ∨ IsSquare (u * δ)) ∧ Module.finrank F E = 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsQuadraticSqrtExtensionIsGaloisAndEmbedding

private theorem **quad_unique** {F E : Type} [Field F] [Field E] [Algebra F E] {δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsQuad δ θ σ) {a b c d : F} (e : algebraMap F E a + algebraMap F E b * θ = algebraMap F E c + algebraMap F E d * θ) : a = c ∧ b = d

theorem **isQuad_galois_and_embedding** : ∀ (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → IsGalois F E ∧ (∀ τ : E ≃ₐ[F] E, τ = 1 ∨ τ = σ) ∧ ∀ (L : Type) [Field L] (φ : F →+* L) (r : L), r ^ 2 = φ δ → ∃ ψ : E →+* L, (∀ a : F, ψ (algebraMap F E a) = φ a) ∧ ψ θ = r

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsQuadraticSqrtExtensionIsGaloisAndEmbeddingStatement

theorem **_root_.GlobalTheory.T.isQuadraticSqrtExtension_isGalois_and_embedding** : ∀ (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → IsGalois F E ∧ (∀ τ : E ≃ₐ[F] E, τ = 1 ∨ τ = σ) ∧ ∀ (L : Type) [Field L] (φ : F →+* L) (r : L), r ^ 2 = φ δ → ∃ ψ : E →+* L, (∀ a : F, ψ (algebraMap F E a) = φ a) ∧ ψ θ = r

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSelfDualCLatFrame

theorem **hO_smul_left** (a : AE D) (x y : WO S T) : hO S T (a • x) y = a * hO S T x y

theorem **hO_smul_right** (a : AE D) (x y : WO S T) : hO S T x (a • y) = cA D a * hO S T x y

theorem **gram_inv_star** (a b : Fin S.rank) : star (S.gram⁻¹ a b) = S.gram⁻¹ b a

theorem **hO_herm** (x y : WO S T) : cA D (hO S T x y) = hO S T y x

theorem **symplO_skew** (x y : WO S T) : symplO S T y x = -symplO S T x y

theorem **symplO_add_left** (x x' y : WO S T) : symplO S T (x + x') y = symplO S T x y + symplO S T x' y

theorem **symplO_add_right** (x y y' : WO S T) : symplO S T x (y + y') = symplO S T x y + symplO S T x y'

theorem **symplO_smul** (a : AE D) (x y : WO S T) : symplO S T (a • x) y = symplO S T x (cA D a • y)

theorem **symplO_zero_right** (x : WO S T) : symplO S T x 0 = 0

theorem **symplO_zero_left** (x : WO S T) : symplO S T 0 x = 0

theorem **isBichar_symplO** (ψ : AddChar (AF D.host.F) ℂ) : IsBichar fun x y : WO S T => ψ (symplO S T x y) where

theorem **heisL_heisL** (ψ : AddChar (AF D.host.F) ℂ) (w w' : WO S T) (f : WO S T → ℂ) : heisL S T ψ w (heisL S T ψ w' f) = ψ (symplOH S T w w') • heisL S T ψ (w + w') f

theorem **heisL_comm** (ψ : AddChar (AF D.host.F) ℂ) (w w' : WO S T) (f : WO S T → ℂ) : heisL S T ψ w (heisL S T ψ w' f) = ψ (symplO S T w w') • heisL S T ψ w' (heisL S T ψ w f)

theorem **mem_half** {e : AE D} {u : WO S T} : u ∈ half S T e ↔ e • u = u

theorem **isHalf_iff** {e : AE D} {u : WO S T} : IsHalf e u ↔ u ∈ half S T e

theorem **heisL_half** (ψ : AddChar (AF D.host.F) ℂ) {e : AE D} (he : cA D e ∗ e = 0) (x : WO S T) (hx : x ∈ half S T e) (x' : WO S T) (hx' : x' ∈ half S T e) (f : WO S T → ℂ) : heisL S T ψ x (heisL S T ψ x' f) = heisL S T ψ (x + x') f

theorem **inv_smul_nsmul** {k : ℕ} (hk : 0 < k) (t : WO S T) : incA D (k : D.host.E)⁻¹ • (k • t) = t

theorem **half_div** (e : AE D) (k : ℕ) (hk : 0 < k) (x : WO S T) (hx : x ∈ half S T e) : ∃ x' ∈ half S T e, k • x' = x

theorem **mem_latFrame** (y : Fin S.rank → D.host.E) : y ∈ latFrame S ↔ ∑ i, y i • S.frame i ∈ S.L0

theorem **coe_latFrame** : (latFrame S : Set (Fin S.rank → D.host.E)) = L0c D S

theorem **isSelfDualC_latFrame** : IsSelfDualC S (latFrame S)

theorem **latO_le_top** (P : HeightOneSpectrum (𝒪 D.host.F) → Prop) : latO S T LS LT P ≤ latO S T LS LT fun _ => True

theorem **placeIdem_smul_mem** (hε : IsPlaceIdem D v ε) {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) : algebraMap (AF D.host.F) (AE D) ε • u ∈ latO S T LS LT fun w => w = v

theorem **idem_smul_mem** (hε : IsPlaceIdem D v ε) {e : AE D} (he : e ∗ e = e) (heε : algebraMap (AF D.host.F) (AE D) ε ∗ e = e) {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) : e • u ∈ (latO S T LS LT fun w => w = v) ⊓ half S T e

theorem **infPart_half** (hε : IsPlaceIdem D v ε) {e : AE D} (heε : algebraMap (AF D.host.F) (AE D) ε ∗ e = e) {u : WO S T} (hu : u ∈ half S T e) : u.map (infPart D) = 0

theorem **halfInd_eq_ind** (e : AE D) (Λ : AddSubgroup (WO S T)) : halfInd e Λ = ind (Λ ⊓ half S T e)

theorem **exists_placeModel** (ψ : AddChar (AF D.host.F) ℂ) {e : AE D} (hec : e ∗ cA D e = 0) (Λ : AddSubgroup (WO S T)) (hLL' : ∀ u ∈ Λ ⊓ half S T e, ∀ y ∈ Λ ⊓ half S T (cA D e), ψ (symplO S T u y) = 1) (hX : ∀ x ∈ half S T e, (∀ y ∈ Λ ⊓ half S T (cA D e), ψ (symplO S T x y) = 1) → x ∈ Λ ⊓ half S T e) (hY : ∀ y ∈ half S T (cA D e), (∀ u ∈ Λ ⊓ half S T e, ψ (symplO S T u y) = 1) → y ∈ Λ ⊓ half S T (cA D e)) (hfin : ∀ k : ℕ, 0 < k → ∃ R : Set (WO S T), R.Finite ∧ ∀ t ∈ Λ ⊓ half S T e, ∃ r ∈ R, ∃ u ∈ Λ ⊓ half S T e, t = r + k • u) {c : ℂ} (hc : c ≠ 0) (f₀ : WO S T → ℂ) (hf : ∀ u ∈ Λ ⊓ half S T e, heisO ψ S T u f₀ = f₀) (hf' : ∀ u ∈ Λ ⊓ half S T (cA D e), heisO ψ S T u f₀ = f₀) : ∃ Tm, IsPlaceModel ψ (symplO S T) (heisO ψ S T) e Λ c f₀ Tm

theorem **mem_of_dual** {ψ : AddChar (AF D.host.F) ℂ} (hε : IsPlaceIdem D v ε) (hdual : ∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ (latO S T LS LT fun _ => True) ↔ ∀ u' ∈ (latO S T LS LT fun _ => True), ψ (symplO S T u u') = 1)) {a : AE D} (ha : a ∗ a = a) (h1 : algebraMap (AF D.host.F) (AE D) ε ∗ a = a) (h2 : algebraMap (AF D.host.F) (AE D) ε ∗ cA D a = cA D a) (x : WO S T) (hx : x ∈ half S T a) (h : ∀ y ∈ (latO S T LS LT fun w => w = v) ⊓ half S T (cA D a), ψ (symplO S T x y) = 1) : x ∈ (latO S T LS LT fun w => w = v) ⊓ half S T a

theorem **placeModelExistsO** : ∀ D : D7, PlaceModelExistsO D

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSemisimpleRepresentationOfISupRangeEquivariant

theorem **_root_.GlobalTheory.ClassFin.isSemisimpleRepresentation_of_iSup_range** : ∀ (G : Type) [Group G] (W : Type) [AddCommGroup W] [Module ℂ W] (ρ : Representation ℂ G W) (J : Type) (V : J → Type) [∀ j, AddCommGroup (V j)] [∀ j, Module ℂ (V j)] (r : ∀ j, Representation ℂ G (V j)) (p : ∀ j, V j →ₗ[ℂ] W), (∀ j, (r j).IsSemisimpleRepresentation) → (∀ j g v, p j (r j g v) = ρ g (p j v)) → (⨆ j, LinearMap.range (p j)) = ⊤ → ρ.IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSemisimpleRepresentationOfISupRangeEquivariantStatement

theorem **_root_.GlobalTheory.T.isSemisimpleRepresentation_of_iSup_range_equivariant** : ∀ (G : Type) [Group G] (W : Type) [AddCommGroup W] [Module ℂ W] (ρ : Representation ℂ G W) (J : Type) (V : J → Type) [∀ j, AddCommGroup (V j)] [∀ j, Module ℂ (V j)] (r : ∀ j, Representation ℂ G (V j)) (p : ∀ j, V j →ₗ[ℂ] W), (∀ j, (r j).IsSemisimpleRepresentation) → (∀ j g v, p j (r j g v) = ρ g (p j v)) → (⨆ j, LinearMap.range (p j)) = ⊤ → ρ.IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSemisimpleRepresentationTprodChevalley

theorem **_root_.GlobalTheory.ClassFin.isIrreducible_toRepresentation_of_isAtom** (σ : Subrepresentation ρ) (h : IsAtom σ) : σ.toRepresentation.IsIrreducible

theorem **iSup_atoms_eq_top** [FiniteDimensional ℂ V] (h : ρ.IsSemisimpleRepresentation) : (⨆ σ : {σ : Subrepresentation ρ // IsAtom σ}, σ.1.toSubmodule) = ⊤

private theorem **iSup_atoms_toSubmodule_eq_whole_of_semisimple** {G : Type} [Group G] {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (ρ : Representation ℂ G V) (hρ : ρ.IsSemisimpleRepresentation) : ⨆ σ : {σ : Subrepresentation ρ // IsAtom σ}, σ.1.toSubmodule = ⊤

theorem **_root_.GlobalTheory.ClassFin.isSemisimpleRepresentation_tprod** : ∀ (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (W : Type) [AddCommGroup W] [Module ℂ W] (ρ' : Representation ℂ G W) [FiniteDimensional ℂ V] [FiniteDimensional ℂ W], ρ.IsSemisimpleRepresentation → ρ'.IsSemisimpleRepresentation → (ρ.tprod ρ').IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSemisimpleRepresentationTprodChevalleyStatement

theorem **_root_.GlobalTheory.T.isSemisimpleRepresentation_tprod_chevalley** : ∀ (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (W : Type) [AddCommGroup W] [Module ℂ W] (ρ' : Representation ℂ G W) [FiniteDimensional ℂ V] [FiniteDimensional ℂ W], ρ.IsSemisimpleRepresentation → ρ'.IsSemisimpleRepresentation → (ρ.tprod ρ').IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSemisimpleRepresentationTprodOfIsIrreducibleAl

theorem **span_range_toMatrix_eq_top_of_isIrreducible** {ι : Type} [Fintype ι] [DecidableEq ι] {G : Type} [Group G] {V : Type} [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (b : Module.Basis ι ℂ V) (h : ρ.IsIrreducible) : Submodule.span ℂ (Set.range ((LinearMap.toMatrixAlgEquiv b : Module.End ℂ V →* Matrix ι ι ℂ).comp ρ)) = ⊤

theorem **isSemisimpleRepresentation_tprod_of_isIrreducible** : ∀ (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (W : Type) [AddCommGroup W] [Module ℂ W] (ρ' : Representation ℂ G W) [FiniteDimensional ℂ V] [FiniteDimensional ℂ W], ρ.IsIrreducible → ρ'.IsIrreducible → (ρ.tprod ρ').IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSemisimpleRepresentationTprodOfIsIrreducibleAlStatement

theorem **_root_.GlobalTheory.T.isSemisimpleRepresentation_tprod_of_isIrreducible'** : ∀ (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) (W : Type) [AddCommGroup W] [Module ℂ W] (ρ' : Representation ℂ G W) [FiniteDimensional ℂ V] [FiniteDimensional ℂ W], ρ.IsIrreducible → ρ'.IsIrreducible → (ρ.tprod ρ').IsSemisimpleRepresentation

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSplitPlaceOfSplitPl

theorem **evalAt_apply** (v : HeightOneSpectrum (𝒪 F)) (a : FiniteAdeleRing (𝒪 F) F) : evalAt v a = a v

theorem **loc_tmul** (v : HeightOneSpectrum (𝒪 F)) (a : FiniteAdeleRing (𝒪 F) F) (x : E) : loc E v (a ⊗ₜ[F] x) = a v ⊗ₜ[F] x

theorem **eq_zero_of_loc** {x : RA F E} (h : ∀ v, loc E v x = 0) : x = 0

theorem **eq_of_loc** {x y : RA F E} (h : ∀ v, loc E v x = loc E v y) : x = y

theorem **ind_pos** {S : Set (HeightOneSpectrum (𝒪 F))} {w} (h : w ∈ S) : ind S w = 1

theorem **ind_neg** {S : Set (HeightOneSpectrum (𝒪 F))} {w} (h : w ∉ S) : ind S w = 0

theorem **ind_int** (S : Set (HeightOneSpectrum (𝒪 F))) (w) : ind S w ∈ w.adicCompletionIntegers F

theorem **loc_ind_pos** {S : Set (HeightOneSpectrum (𝒪 F))} {w} (h : w ∈ S) : loc E w (ind S ⊗ₜ[F] (1 : E)) = 1

theorem **loc_ind_neg** {S : Set (HeightOneSpectrum (𝒪 F))} {w} (h : w ∉ S) : loc E w (ind S ⊗ₜ[F] (1 : E)) = 0

theorem **exists_inv** (m : ℕ) (hm : m ≠ 0) (S : Set (HeightOneSpectrum (𝒪 F))) (hS : ∀ v ∈ S, (m : 𝒪 F) ∉ v.asIdeal) : ∃ a : FiniteAdeleRing (𝒪 F) F, (∀ v, a v ∈ v.adicCompletionIntegers F) ∧ (m : RA F E) * (a ⊗ₜ[F] (1 : E)) = ind S ⊗ₜ[F] (1 : E)

theorem **isUnA_iff** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (g : Matrix (Fin n) (Fin n) (RA F E)) : IsUnA c G g ↔ IsUnitaryA c G g

theorem **fm_eq_ratM** {n : ℕ} (γ : Matrix (Fin n) (Fin n) E) : fm F γ = ratM F γ

theorem **isSplitPlace_of_splitPl** (c : E ≃ₐ[F] E) {v : HeightOneSpectrum (𝒪 F)} (h : SplitPl c v) : IsSplitPlace E v

theorem **aut_eq** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) (σ : E ≃ₐ[F] E) : σ = 1 ∨ σ = c

theorem **conj_conj** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) (x : E) : c (c x) = x

theorem **moved_of_not_le** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) {P Q : Ideal (𝒪 E)} (hQ : Q.IsPrime) (h : P.comap (algebraMap (𝒪 F) (𝒪 E)) = Q.comap (algebraMap (𝒪 F) (𝒪 E))) (hPQ : ¬ P ≤ Q) : Moved c Q

theorem **splitPl_iff** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) (v : HeightOneSpectrum (𝒪 F)) : SplitPl c v ↔ IsSplitPlace E v

theorem **support_iff** (p : HeightOneSpectrum (𝒪 F) → Prop) {n : ℕ} (h : Matrix (Fin n) (Fin n) (RA F E)) : (∃ e : FiniteAdeleRing (𝒪 F) F, e * e = e ∧ {v | e v ≠ 0}.Finite ∧ (∀ v, e v ≠ 0 → p v) ∧ (e ⊗ₜ[F] (1 : E)) • (h - 1) = h - 1) ↔ ∃ T : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∈ T, p v) ∧ ∀ v ∉ T, h.map (loc E v) = 1

theorem **suppSplit_iff** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) {n : ℕ} (N : ℕ) (h : Matrix (Fin n) (Fin n) (RA F E)) : SuppSplit c N h ↔ SupportedAtSplit N h

theorem **notMem_of_coprime** {t N : ℕ} (ht : t.Coprime N) {v : HeightOneSpectrum (𝒪 F)} (hv : (N : 𝒪 F) ∈ v.asIdeal) : (t : 𝒪 F) ∉ v.asIdeal

theorem **congOneAt_of_congAtA** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (N D : ℕ) (k : Matrix (Fin n) (Fin n) (RA F E)) (h : CongAtA F L N D k) : CongOneAt N D (L : Set (Fin n → E)) k

theorem **exists_coprime_dvd** (N m : ℕ) (hm : m ≠ 0) : ∃ t j : ℕ, t.Coprime N ∧ m ∣ t * N ^ j

theorem **exists_glue** {n : ℕ} (L : Set (Fin n → E)) {N : ℕ} (hN : N ≠ 0) (D : ℕ) (u : Fin n → RA F E) (h : ∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → ∃ y ∈ hat F L, ∀ i, loc E v (u i) = loc E v ((D • y) i)) : ∃ y ∈ hat F L, ∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → ∀ i, loc E v (u i) = loc E v ((D • y) i)

theorem **congAtA_of_congOneAt** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) {N D d : ℕ} (hN : N ≠ 0) (hD : D ≠ 0) (hd : d ≠ 0) (k : Matrix (Fin n) (Fin n) (RA F E)) (hk : ∀ z ∈ hat F (L : Set (Fin n → E)), d • k *ᵥ z ∈ hat F (L : Set (Fin n → E))) (h : CongOneAt N D (L : Set (Fin n → E)) k) {t j : ℕ} (hdvd : D * d ∣ t * N ^ j) : ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), t • (k *ᵥ z - z) = D • w

theorem **congAtA_iff_of_bounded** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) {N D d : ℕ} (hN : N ≠ 0) (hD : D ≠ 0) (hd : d ≠ 0) (k : Matrix (Fin n) (Fin n) (RA F E)) (hk : ∀ z ∈ hat F (L : Set (Fin n → E)), d • k *ᵥ z ∈ hat F (L : Set (Fin n → E))) : CongAtA F L N D k ↔ CongOneAt N D (L : Set (Fin n → E)) k

theorem **congAtA_iff** [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hfg : L.FG) (hsp : Submodule.span E (L : Set (Fin n → E)) = ⊤) {N D : ℕ} (hN : N ≠ 0) (hD : D ≠ 0) (k : Matrix (Fin n) (Fin n) (RA F E)) : CongAtA F L N D k ↔ CongOneAt N D (L : Set (Fin n → E)) k

theorem **inc_injective** : Function.Injective (Algebra.TensorProduct.includeRight (R

theorem **unitary_iff_matrix** {R : Type} [CommRing R] {n : ℕ} (s : R →+* R) (G g : Matrix (Fin n) (Fin n) R) : (∀ x y : Fin n → R, ∑ a, ∑ b, (g *ᵥ x) a * G a b * s ((g *ᵥ y) b) = ∑ a, ∑ b, x a * G a b * s (y b)) ↔ gᵀ * G * g.map s = G

theorem **isUnA_iff_matrix** (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (g : Matrix (Fin n) (Fin n) (RA F E)) : IsUnA c G g ↔ gᵀ * fm F G * g.map (sg (FiniteAdeleRing (𝒪 F) F) c).toRingHom = fm F G

theorem **fm_eq_map** {n : ℕ} (γ : Matrix (Fin n) (Fin n) E) : fm F γ = γ.map (Algebra.TensorProduct.includeRight (R

theorem **isUnE_iff** (c : E ≃ₐ[F] E) {n : ℕ} (G γ : Matrix (Fin n) (Fin n) E) : IsUnE c G γ ↔ IsUnitaryA c G (ratM F γ)

theorem **isUnit_of_isUnA** (c : E ≃ₐ[F] E) {n : ℕ} {G : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) {g : Matrix (Fin n) (Fin n) (RA F E)} (hg : IsUnA c G g) : IsUnit g

theorem **isUnit_of_isUnE** (c : E ≃ₐ[F] E) {n : ℕ} {G : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) {γ : Matrix (Fin n) (Fin n) E} (hγ : IsUnE c G γ) : IsUnit γ

theorem **isHermForm_iff** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) : IsHermForm c G ↔ c ≠ 1 ∧ (∀ a b, G b a = c (G a b)) ∧ G.det ≠ 0

theorem **hasRep_of** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) {n : ℕ} {G : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) (L : Submodule (𝒪 E) (Fin n → E)) {N : ℕ} (hN : N ≠ 0) (H : HasSplitRepresentatives c G (L : Set (Fin n → E)) N N) : HasRep c G L N

theorem **of_hasRep** (c : E ≃ₐ[F] E) (h2 : Module.finrank F E = 2) (hc : c ≠ 1) {n : ℕ} {G : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) (L : Submodule (𝒪 E) (Fin n → E)) (N D : ℕ) (H : HasRep c G L (N * D)) : HasSplitRepresentatives c G (L : Set (Fin n → E)) N D

theorem **RepStatement.mono** {a b : ℕ} (hab : a ≤ b) (H : RepStatement a) : RepStatement b

theorem **repStatement_iff** : RepStatement 2 ↔ SplitRepresentatives

theorem **typings_agree** : (∀ (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E) (g : Matrix (Fin n) (Fin n) (RA F E)), IsUnA c G g ↔ IsUnitaryA c G g) ∧ (∀ (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] (c : E ≃ₐ[F] E), Module.finrank F E = 2 → c ≠ 1 → (∀ v : HeightOneSpectrum (𝒪 F), SplitPl c v ↔ IsSplitPlace E v) ∧ ∀ (n N : ℕ) (h : Matrix (Fin n) (Fin n) (RA F E)), SuppSplit c N h ↔ SupportedAtSplit N h) ∧ (∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ →

∀ N D : ℕ, N ≠ 0 → D ≠ 0 → ∀ k : Matrix (Fin n) (Fin n) (RA F E), CongAtA F L N D k ↔ CongOneAt N D (L : Set (Fin n → E)) k) ∧ (RepStatement 2 ↔ SplitRepresentatives)

theorem **repStatement_zero_iff** : RepStatement 0 ↔ (∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (RA F E)) (N : ℕ), IsUnA c G g.1 → 0 < N → ∃ (γ : GL (Fin n) E) (h k : GL (Fin n) (RA F E)), IsUnE c G γ.1 ∧ IsUnA c G h.1 ∧ IsUnA c G k.1 ∧ g.1 = fm F γ.1 * h.1 * k.1 ∧ SuppSplit c N h.1 ∧ (∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), k.1 *ᵥ z - z = N • w) ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), (k⁻¹).1 *ᵥ z ∈ hat F (L : Set (Fin n → E)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSquareMulOfNonsquareUnits

theorem **_root_.GlobalTheory.QNorm.isSquare_of_isSquare_residue_odd** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (h2 : Valued.v (2 : v.adicCompletion F) = 1) (z : v.adicCompletionIntegers F) (hz : Valued.v (z : v.adicCompletion F) = 1) (h : IsSquare (residue (v.adicCompletionIntegers F) z)) : IsSquare (z : v.adicCompletion F)

theorem **_root_.GlobalTheory.QNorm.isSquare_mul_of_nonsquare_units** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)), Valued.v (2 : v.adicCompletion F) = 1 → ∀ x y : v.adicCompletion F, Valued.v x = 1 → Valued.v y = 1 → ¬ IsSquare x → ¬ IsSquare y → IsSquare (x * y)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSquareOfSUnitLocalSquareFiniteRealPlaces

theorem **mem_normSub_of_isSquare** {F L : Type} [Field F] [CharZero F] [Field L] (φ : F →+* L) (δ : Fˣ) (h : IsSquare (φ (δ : F))) (x : Fˣ) : x ∈ normSub φ (δ : F)

theorem **isSquare_of_mem_locSq** : ∀ (F : Type) [Field F] [NumberField F] [DecidableEq (HeightOneSpectrum (𝒪 F))] (S T : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → sUnits F (S ∪ T) ⊔ finSq F T = ⊤ → ∀ y ∈ sUnits F (S ∪ T), y ∈ locSq F S → IsSquare (y : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IsSquareOfValuedSubOneLt

theorem **_root_.GlobalTheory.LocalQuad.SquareTheorem.isSquare_of_norm_sub_one_lt** {K : Type*} [NormedField K] [CompleteSpace K] [IsUltrametricDist K] (x : K) (h : ‖x - 1‖ < ‖(4 : K)‖) : IsSquare x

theorem **_root_.GlobalTheory.LocalQuad.isSquare_of_valued_sub_one_lt** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (x : v.adicCompletion F) (h : Valued.v (x - 1) < Valued.v (4 : v.adicCompletion F)) : IsSquare x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IwasawaDecompositionRealMatrix

theorem **exists_orthogonal_mul_upper** (N : ℕ) (g : Matrix (Fin N) (Fin N) ℝ) : ∃ (k r : Matrix (Fin N) (Fin N) ℝ), kᵀ * k = 1 ∧ (∀ i j, j < i → r i j = 0) ∧ g = k * r

theorem **iwasawa_real** : ∀ (N : ℕ) (g : Matrix (Fin N) (Fin N) ℝ), g.det ≠ 0 → ∃ (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n : Matrix (Fin N) (Fin N) ℝ), kᵀ * k = 1 ∧ (∀ i, 0 < a i) ∧ (∀ i, n i i = 1) ∧ (∀ i j, j < i → n i j = 0) ∧ g = k * diagonal a * n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.IwasawaDecompositionRealMatrixStatement

theorem **_root_.GlobalTheory.T.iwasawa_decomposition_real_matrix** : ∀ (N : ℕ) (g : Matrix (Fin N) (Fin N) ℝ), g.det ≠ 0 → ∃ (k : Matrix (Fin N) (Fin N) ℝ) (a : Fin N → ℝ) (n : Matrix (Fin N) (Fin N) ℝ), kᵀ * k = 1 ∧ (∀ i, 0 < a i) ∧ (∀ i, n i i = 1) ∧ (∀ i j, j < i → n i j = 0) ∧ g = k * diagonal a * n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.KummerGalois

theorem **IsKum.minpoly_eq** {p : ℕ} {F E : Type} [Field F] [Field E] [Algebra F E] {ζ δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsKum p ζ δ θ σ) : minpoly F θ = X ^ p - C δ

theorem **IsKum.pow_apply** {p : ℕ} {F E : Type} [Field F] [Field E] [Algebra F E] {ζ δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsKum p ζ δ θ σ) (k : ℕ) : (σ ^ k) θ = algebraMap F E (ζ ^ k) * θ

theorem **kummer_galois** : ∀ (p : ℕ) (F E : Type) [Field F] [CharZero F] [Field E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → FiniteDimensional F E ∧ Module.finrank F E = p ∧ IsGalois F E ∧ orderOf σ = p ∧ (∀ τ : E ≃ₐ[F] E, τ ∈ Subgroup.zpowers σ) ∧ Nonempty (E ≃ₐ[F] AdjoinRoot (X ^ p - C δ : F[X])) ∧ ∀ (L : Type) [Field L] [Algebra F L] (r : L), r ^ p = algebraMap F L δ → ∃ ψ : E →ₐ[F] L, ψ θ = r

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LandauSummableOfExpDirichletSeriesEntire

private theorem **hasSum_prod_fin_pow** {ι : Type} (z : ι → ℂ) (g : ℂ) (hz : Summable fun i => ‖z i‖) (h : HasSum z g) (n : ℕ) : HasSum (fun f : Fin n → ι => ∏ k, z (f k)) (g ^ n) ∧ Summable fun f : Fin n → ι => ‖∏ k, z (f k)‖

theorem **_root_.GlobalTheory.hasSum_tuple_prod_exp** {ι : Type} (z : ι → ℂ) (g : ℂ) (h : HasSum z g) : HasSum (fun p : (Σ n : ℕ, Fin n → ι) => (∏ k, z (p.2 k)) / (p.1.factorial : ℂ)) (Complex.exp g)

theorem **_root_.GlobalTheory.landau_exp_summable** : ∀ (ι : Type) (a l : ι → ℝ) (F : ℂ → ℂ), (∀ i, 0 ≤ a i) → (∀ i, 0 ≤ l i) → Differentiable ℂ F → (∀ s : ℂ, 1 < s.re → ∃ g : ℂ, HasSum (fun i => (a i : ℂ) * Complex.exp (-(l i : ℂ) * s)) g ∧ F s = Complex.exp g) → Summable a

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LandherrAndLandherrSt

theorem **_root_.GlobalTheory.QNorm.landherr_and_landherrSt** : _root_.GlobalTheory.Published.Landherr ∧ _root_.GlobalTheory.Published.LandherrSt

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LatticeSumDilateHaarAsymptotic

theorem **integral_div_covolume_eq_of_haar** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] (μ ν : Measure V) [μ.IsAddHaarMeasure] [ν.IsAddHaarMeasure] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L] (f : V → ℂ) (s : ℝ) : (∫ x, f x ∂μ) / ((ZLattice.covolume L μ * s : ℝ) : ℂ) = (∫ x, f x ∂ν) / ((ZLattice.covolume L ν * s : ℝ) : ℂ)

theorem **lattice_sum_dilate_haar** : ∀ (V : Type) [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] (μ : Measure V) [μ.IsAddHaarMeasure] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L] (f : V → ℂ), ContDiff ℝ ∞ f → HasCompactSupport f → ∀ A : ℕ, ∃ C : ℝ, ∀ (x₀ : V) (t : ℝ), 0 < t → t ≤ 1 → ‖∑' l : L, f (t • ((l : V) + x₀)) - (∫ x, f x ∂μ) / ((ZLattice.covolume L μ * t ^ Module.finrank ℝ V : ℝ) : ℂ)‖ ≤ C * t ^ A

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LatticeSumDilateHaarAsymptoticStatement

theorem **_root_.GlobalTheory.T.lattice_sum_dilate_haar_asymptotic** : ∀ (V : Type) [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] (μ : Measure V) [μ.IsAddHaarMeasure] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L] (f : V → ℂ), ContDiff ℝ ∞ f → HasCompactSupport f → ∀ A : ℕ, ∃ C : ℝ, ∀ (x₀ : V) (t : ℝ), 0 < t → t ≤ 1 → ‖∑' l : L, f (t • ((l : V) + x₀)) - (∫ x, f x ∂μ) / ((ZLattice.covolume L μ * t ^ Module.finrank ℝ V : ℝ) : ℂ)‖ ≤ C * t ^ A

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LeftTranslateMatrixPolyCompletelyReducible

theorem **ltr_mul_left** {ι : Type} [Fintype ι] [DecidableEq ι] (g h : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : ltr (g * h) f = ltr h (ltr g f)

theorem **ltr_one_left** {ι : Type} [Fintype ι] [DecidableEq ι] (f : MvPolynomial (ι × ι) ℂ) : ltr (1 : Matrix ι ι ℂ) f = f

theorem **ltr_completely_reducible** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∀ U : Submodule ℂ (MvPolynomial (ι × ι) ℂ), (∀ g ∈ Γ, ∀ f ∈ U, ltr g f ∈ U) → ∃ U' : Submodule ℂ (MvPolynomial (ι × ι) ℂ), (∀ g ∈ Γ, ∀ f ∈ U', ltr g f ∈ U') ∧ IsCompl U U'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LeftTranslateMatrixPolyCompletelyReducibleStatement

theorem **_root_.GlobalTheory.T.leftTranslateMatrixPoly_completelyReducible** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → SS Γ → ∀ U : Submodule ℂ (MvPolynomial (ι × ι) ℂ), (∀ g ∈ Γ, ∀ f ∈ U, ltr g f ∈ U) → ∃ U' : Submodule ℂ (MvPolynomial (ι × ι) ℂ), (∀ g ∈ Γ, ∀ f ∈ U', ltr g f ∈ U') ∧ IsCompl U U'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LeftTranslateMatrixPolyTensorFormProperties

theorem **exists_monomial_eq_prod_X** {σ : Type} (n : ℕ) : ∀ d : σ →₀ ℕ, d.degree = n → ∃ a : Fin n → σ, (MvPolynomial.monomial d (1 : ℂ)) = ∏ k, MvPolynomial.X (a k)

theorem **evC_ltr** (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ) : evC (ltr g f) y = evC f (g * y)

theorem **evC_ext** {f f' : MvPolynomial (ι × ι) ℂ} (h : ∀ y, evC f y = evC f' y) : f = f'

theorem **tpow_mul_complex** (n : ℕ) (y z : Matrix ι ι ℂ) : tpow n (y * z) = tpow n y * tpow n z

theorem **tensorForm_apply** (n : ℕ) (b : Fin n → ι) (l : (Fin n → ι) → ℂ) : tensorForm n b l = ∑ a : Fin n → ι, l a • ∏ k, MvPolynomial.X (a k, b k)

theorem **evC_tensorForm** (n : ℕ) (b : Fin n → ι) (l : (Fin n → ι) → ℂ) (y : Matrix ι ι ℂ) : evC (tensorForm n b l) y = (l ∘* tpow n y) b

theorem **ltr_tensorForms** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι], (∀ (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ), evC (ltr g f) y = evC f (g * y)) ∧ (∀ (Γ : Set (Matrix ι ι ℂ)) (f : MvPolynomial (ι × ι) ℂ), IsInv Γ f ↔ ∀ g ∈ Γ, ltr g f = f) ∧ (∀ (g : Matrix ι ι ℂ) (n : ℕ) (f : MvPolynomial (ι × ι) ℂ), f.IsHomogeneous n → (ltr g f).IsHomogeneous n) ∧ ∀ n : ℕ, ∃ p : (Fin n → ι) → ((Fin n → ι) → ℂ) →₁[ℂ] MvPolynomial (ι × ι) ℂ, (∀ b l (y : Matrix ι ι ℂ), evC (p b l) y = (l ∘* tpow n y) b) ∧ (∀ b l (g : Matrix ι ι ℂ), ltr g (p b l) = p b (l ∘* tpow n g)) ∧ ∀ f : MvPolynomial (ι × ι) ℂ, f.IsHomogeneous n → ∃ l : (Fin n → ι) → (Fin n → ι) → ℂ, f = ∑ b, p b (l b)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LeftTranslateMatrixPolyTensorFormPropertiesStatement

theorem **_root_.GlobalTheory.T.leftTranslateMatrixPoly_tensorForm_properties** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι], (∀ (g y : Matrix ι ι ℂ) (f : MvPolynomial (ι × ι) ℂ), evC (ltr g f) y = evC f (g * y)) ∧ (∀ (Γ : Set (Matrix ι ι ℂ)) (f : MvPolynomial (ι × ι) ℂ), IsInv Γ f ↔ ∀ g ∈ Γ, ltr g f = f) ∧ (∀ (g : Matrix ι ι ℂ) (n : ℕ) (f : MvPolynomial (ι × ι) ℂ), f.IsHomogeneous n → (ltr g f).IsHomogeneous n) ∧ ∀ n : ℕ, ∃ p : (Fin n → ι) → ((Fin n → ι) → ℂ) →₁[ℂ] MvPolynomial (ι × ι) ℂ, (∀ b l (y : Matrix ι ι ℂ), evC (p b l) y = (l ∘* tpow n y) b) ∧ (∀ b l (g : Matrix ι ι ℂ), ltr g (p b l) = p b (l ∘* tpow n g)) ∧ ∀ f : MvPolynomial (ι × ι) ℂ, f.IsHomogeneous n → ∃ l : (Fin n → ι) → (Fin n → ι) → ℂ, f = ∑ b, p b (l b)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LinearIndependentWave

theorem **isBichar_one** : IsBichar (fun _ _ : G => (1 : ℂ))

theorem **isBichar_neg_one_zpow** : IsBichar (fun a b : ℤ => (-1 : ℂ) ^ (a * b))

theorem **ind_add** (L : AddSubgroup G) (t : G) {u : G} (hu : u ∈ L) : ind L (t + u) = ind L t

theorem **IsBichar.neg_mul** (h : IsBichar χ) (x y : G) : χ (-x) y * χ x y = 1

theorem **IsBichar.nsmul** (h : IsBichar χ) (k : ℕ) (x y : G) : χ (k • x) y = χ x (k • y)

theorem **wave_trans** (x x' y : G) : (fun t => wave χ L x y (t + x')) = wave χ L (x' + x) y

theorem **wave_modul** (h : IsBichar χ) (x y y' : G) : (fun t => χ t y' * wave χ L x y t) = χ (-x) y' • wave χ L x (y' + y)

theorem **wave_zero** (h : IsBichar χ) : wave χ L 0 0 = ind L

theorem **wave_add** (h : IsBichar χ) (hLL' : ∀ u ∈ L, ∀ y ∈ L', χ u y = 1) (x y : G) {l l' : G} (hl : l ∈ L) (hl' : l' ∈ L') : wave χ L (x + l) (y + l') = χ l y • wave χ L x y

theorem **linearIndependent_wave** (h : IsBichar χ) (hY : ∀ y ∈ Y, (∀ u ∈ L, χ u y = 1) → y ∈ L') : LinearIndependent ℂ fun i : (X ⧸ L.addSubgroupOf X) × (Y ⧸ L'.addSubgroupOf Y) => wave χ L (i.1.out : X) (i.2.out : Y)

theorem **exists_map_wave** {V : Type} [AddCommGroup V] [Module ℂ V] (h : IsBichar χ) (hL : L ≤ X) (hL' : L' ≤ Y) (hLL' : ∀ u ∈ L, ∀ y ∈ L', χ u y = 1) (hY : ∀ y ∈ Y, (∀ u ∈ L, χ u y = 1) → y ∈ L') (ρ : G → V →₁[ℂ] V) (ρX : ∀ x ∈ X, ∀ x' ∈ X, ∀ f, ρ x (ρ x' f) = ρ (x + x') f) (ρY : ∀ y ∈ Y, ∀ y' ∈ Y, ∀ f, ρ y (ρ y' f) = ρ (y + y') f) (ρXY : ∀ x ∈ X, ∀ y ∈ Y, ∀ f, ρ x (ρ y f) = χ x y • ρ y (ρ x f)) (f₀ : V) (hf : ∀ u ∈ L, ρ u f₀ = f₀) (hf' : ∀ u ∈ L', ρ u f₀ = f₀) : ∃ T : (G → ℂ) →₁[ℂ] V, ∀ x ∈ X, ∀ y ∈ Y, T (wave χ L x y) = ρ x (ρ y f₀)

theorem **trans_mem** {Φ : G → ℂ} (hΦ : Φ ∈ waveSpan χ X Y L) {x : G} (hx : x ∈ X) : (fun t => Φ (t + x)) ∈ waveSpan χ X Y L

theorem **modul_mem** (h : IsBichar χ) {Φ : G → ℂ} (hΦ : Φ ∈ waveSpan χ X Y L) {y : G} (hy : y ∈ Y) : (fun t => χ t y * Φ t) ∈ waveSpan χ X Y L

theorem **ind_nsmul_mem** (h : IsBichar χ) (hL : L ≤ X) (hL' : L' ≤ Y) (hLL' : ∀ u ∈ L, ∀ y ∈ L', χ u y = 1) (hX : ∀ x ∈ X, (∀ y ∈ L', χ x y = 1) → x ∈ L) {k : ℕ} (dX : ∀ x ∈ X, ∃ x' ∈ X, k • x' = x) (dY : ∀ y ∈ Y, ∃ y' ∈ Y, k • y' = y) (R : Set G) (hR : R.Finite) (hfin : ∀ t ∈ L, ∃ r ∈ R, ∃ u ∈ L, t = r + k • u) : ind (L.map (nsmulAddMonoidHom k)) ∈ waveSpan χ X Y L

theorem **mem_of_finite_cosets** {K : AddSubgroup G} (hK : ind K ∈ waveSpan χ X Y L) (P : Finset G) : ∀ Φ : G → ℂ, (∀ t, Φ t ≠ 0 → t ∈ X) → (∀ u ∈ K, ∀ t, Φ (t + u) = Φ t) → (∀ t, Φ t ≠ 0 → ∃ s ∈ P, ∃ u ∈ K, t = s + u) → Φ ∈ waveSpan χ X Y L

theorem **exists_intertwiner** {V : Type} [AddCommGroup V] [Module ℂ V] (h : IsBichar χ) (hL : L ≤ X) (hL' : L' ≤ Y) (hLL' : ∀ u ∈ L, ∀ y ∈ L', χ u y = 1) (hX : ∀ x ∈ X, (∀ y ∈ L', χ x y = 1) → x ∈ L) (hY : ∀ y ∈ Y, (∀ u ∈ L, χ u y = 1) → y ∈ L') (dX : ∀ k : ℕ, 0 < k → ∀ x ∈ X, ∃ x' ∈ X, k • x' = x) (dY : ∀ k : ℕ, 0 < k → ∀ y ∈ Y, ∃ y' ∈ Y, k • y' = y) (tf : ∀ k : ℕ, 0 < k → Function.Injective fun t : G => k

• t) (hfin : ∀ k : ℕ, 0 < k → ∃ R : Set G, R.Finite ∧ ∀ t ∈ L, ∃ r ∈ R, ∃ u ∈ L, t = r + k • u) (ρ : G → V →ₗ[ℂ] V) (ρX : ∀ x ∈ X, ∀ x' ∈ X, ∀ f, ρ x (ρ x' f) = ρ (x + x') f) (ρY : ∀ y ∈ Y, ∀ y' ∈ Y, ∀ f, ρ y (ρ y' f) = ρ (y + y') f) (ρXY : ∀ x ∈ X, ∀ y ∈ Y, ∀ f, ρ x (ρ y f) = χ x y • ρ y (ρ x f)) (f₀ : V) (hf : ∀ u ∈ L, ρ u f₀ = f₀) (hf' : ∀ u ∈ L', ρ u f₀ = f₀) : ∃ T : (G → ℂ) →ₗ[ℂ] V, T (ind L) = f₀ ∧ ∀ Φ : G → ℂ, (∀ t, Φ t ≠ 0 → t ∈ X) → ∀ k : ℕ, 0 < k → (∀ u ∈ L, ∀ t, Φ (t + k • u)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.LogSqLeTangentAddSq

theorem **log_sq_le_tangent_add_sq** : ∀ a : ℝ, 0 < a → ∃ C : ℝ, 0 ≤ C ∧ ∀ x y : ℝ, a ≤ x → a ≤ y → (Real.log y) ^ 2 ≤ (Real.log x) ^ 2 + 2 * Real.log x / x * (y − x) + C * (y − x) ^ 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MapFinNeg

theorem **map_fin_add** {m n : ℕ} {x y : Mx D m n} (hx : x.map (finPart D) = 0) (hy : y.map (finPart D) = 0) : (x + y).map (finPart D) = 0

theorem **map_fin_neg** {m n : ℕ} {x : Mx D m n} (hx : x.map (finPart D) = 0) : (−x).map (finPart D) = 0

theorem **piW_eq_zero** (θ : InfFam D) (S T : HSp D) {u : WO S T} (hu : u.map (infPart D) = 0) : piW θ T S u = 0

theorem **lineAt_eq** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (hσ : Chart θ σ) (S T : HSp D) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (v : RW T S) (h1 : v τ = z) (h2 : ∀ τ', τ' ≠ τ → v τ' = 0) (t : ℝ) : lineAt θ τ z t = iotaW σ T S (t • v)

theorem **exists_ne_zero** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (hψ : IsStdChar D ψ (−1)) (hθ : IsInfFam θ) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (hS : ∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) (hT : ∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (dS : IsSelfDualC S ΛS) (dT : IsSelfDualC T ΛT) : ∃ f : WO S T → ℂ, f ≠ 0 ∧ ∀ c : ℂ, IsThetaFunO ψ S T (fun w => c * f w) ∧ IsLatVac (heisO ψ S T) (latO S T ΛS ΛT fun _ => True) (fun w => c * f w) ∧ ∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) (fun w => c * f w)

theorem **vacExists** : ∀ D : D7, VacExistsO D

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MatrixEntriesBoundedOfHermitianLogTraceSqLe

private lemma **energy_entry_sq_le** {ι : Type} [Fintype ι] (x : Matrix ι ι ℂ) (i j : ι) : ‖x i j‖ ^ 2 ≤ (Matrix.trace (xᴴ * x)).re

private lemma **energy_trace_conj** {ι : Type} [Fintype ι] [DecidableEq ι] (U : Matrix ι ι ℂ) (hU : U * Uᴴ = 1) (d : ι → ℝ) : (Matrix.trace (U * diagonal (fun i => (d i : ℂ)) * Uᴴ)).re = ∑ i, d i

theorem **entries_le_of_energy_le** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (E : ℝ), ∃ C : ℝ, ∀ x H : Matrix ι ι ℂ, IsLog H (xᴴ * x) → (Matrix.trace (H * H)).re ≤ E → ∀ i j, ‖x i j‖ ≤ C ∧ ‖H i j‖ ≤ C

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MatrixEntriesBoundedOfHermitianLogTraceSqLeStatement

theorem **_root_.GlobalTheory.T.matrix_entries_bounded_of_hermitian_log_trace_sq_le** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (E : ℝ), ∃ C : ℝ, ∀ x H : Matrix ι ι ℂ, IsLog H (xᴴ * x) → (Matrix.trace (H * H)).re ≤ E → ∀ i j, ‖x i j‖ ≤ C ∧ ‖H i j‖ ≤ C

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MatrixKroneckerPowerTensorActionAlgebra

theorem **tpow_diagonal_alg** (n : ℕ) (a : ι → R) : tpow n (diagonal a) = diagonal (fun b : Fin n → ι => ∏ j, a (b j))

theorem **tpow_one_alg** (n : ℕ) : tpow n (1 : Matrix ι ι R) = 1

theorem **tpow_mul_alg** (n : ℕ) (y z : Matrix ι ι R) : tpow n (y * z) = tpow n y * tpow n z

theorem **act_apply_alg** {m : ℕ} (d : Fin m → ℕ) (t : TIdx ι d → R) (y : Matrix ι ι R) (k : TIdx ι d) : act d t y k = ∑ a : Fin (d k.1) → ι, t (k.1, a) * ∏ j, y (a j) (k.2 j)

theorem **tpow_act_algebra** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (R : Type) [CommRing R], (∀ n : ℕ, tpow n (1 : Matrix ι ι R) = 1 ∧ (∀ y z : Matrix ι ι R, tpow n (y * z) = tpow n y * tpow n z) ∧ ∀ y : Matrix ι ι R, (tpow n y)ᵀ = tpow n yᵀ) ∧ ∀ (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → R), act d t 1 = t ∧ (∀ y z : Matrix ι ι R, act d t (y * z) = act d (act d t y) z) ∧ (∀ (a : ι → R) (k : TIdx ι d), act d t (diagonal a) k = t k * ∏ j, a (k.2 j)) ∧ (∀ (S : Subring R) (y : Matrix ι ι R), (∀ k, t k ∈ S) → (∀ i j, y i j ∈ S) → ∀ k, act d t y k ∈ S) ∧ (∀ (c : R) (y : Matrix ι ι R), act d (c • t) y = c • act d t y) ∧ ∀ (R' : Type) [CommRing R'] (φ : R →+* R') (y : Matrix ι ι R), act d (fun k => φ (t k)) (y.map φ) = fun k => φ (act d t y k)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MatrixKroneckerPowerTensorActionAlgebraStatement

theorem **_root_.GlobalTheory.T.matrixKroneckerPower_tensorAction_algebra** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (R : Type) [CommRing R], (∀ n : ℕ, tpow n (1 : Matrix ι ι R) = 1 ∧ (∀ y z : Matrix ι ι R, tpow n (y * z) = tpow n y * tpow n z) ∧ ∀ y : Matrix ι ι R, (tpow n y)ᵀ = tpow n yᵀ) ∧ ∀ (m : ℕ) (d : Fin m → ℕ) (t : TIdx ι d → R), act d t 1 = t ∧ (∀ y z : Matrix ι ι R, act d t (y * z) = act d (act d t y) z) ∧ (∀ (a : ι → R) (k : TIdx ι d), act d t (diagonal a) k = t k * ∏ j, a (k.2 j)) ∧ (∀ (S : Subring R) (y : Matrix ι ι R), (∀ k, t k ∈ S) → (∀ i j, y i j ∈ S) → ∀ k, act d t y k ∈ S) ∧ (∀ (c : R) (y : Matrix ι ι R), act d (c • t) y = c • act d t y) ∧ ∀ (R' : Type) [CommRing R'] (φ : R →+* R') (y : Matrix ι ι R), act d (fun k => φ (t k)) (y.map φ) = fun k => φ (act d t y k)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemIffForallMemLocalizeRingOfIntegers

theorem **exists_smul_mem** {N : Submodule A V} (hN : Submodule.span K (N : Set V) = ⊤) (x : V) : ∃ d ∈ O⁰, d • x ∈ N

theorem **exists_smul_mem_of_finite** {N : Submodule A V} (hN : Submodule.span K (N : Set V) = ⊤) {s : Set V} (hs : s.Finite) : ∃ d ∈ O⁰, ∀ x ∈ s, d • x ∈ N

theorem **exists_smul_le** {M N : Submodule A V} (hM : M.FG) (hN : Submodule.span K (N : Set V) = ⊤) : ∃ d ∈ O⁰, ∀ x ∈ M, d • x ∈ N

theorem **span_eq_top_of_smul_mem** {s : Set V} {N : Submodule A V} (hs : Submodule.span K s = ⊤) {d : O} (hd : d ∈ O⁰) (h : ∀ x ∈ s, d • x ∈ N) : Submodule.span K (N : Set V) = ⊤

theorem **fg_of_smul_mem** [IsNoetherianRing A] {M N : Submodule A V} (hN : N.FG) {d : O} (hd : d ∈ O⁰) (h : ∀ x ∈ M, d • x ∈ N) : M.FG

theorem **localize_def** (v : HeightOneSpectrum O) (L : Submodule O V) : localize K v L = Submodule.span (v.valuationSubringAtPrime K) (L : Set V)

theorem **mem_localRing_iff_valuation** (v : HeightOneSpectrum O) (x : K) : x ∈ v.valuationSubringAtPrime K ↔ v.valuation K x ≤ 1

theorem **mem_localize_iff** (v : HeightOneSpectrum O) (L : Submodule O V) (x : V) : x ∈ localize K v L ↔ ∃ s ∉ v.asIdeal, s • x ∈ L

theorem **mem_localize_of_mem** (v : HeightOneSpectrum O) {L : Submodule O V} {x : V} (hx : x ∈ L) : x ∈ localize K v L

theorem **localize_restrictScalars** (v : HeightOneSpectrum O) (M : Submodule (v.valuationSubringAtPrime K) V) : localize K v (M.restrictScalars O) = M

theorem **mem_of_smul_mem** (v : HeightOneSpectrum O) {M : Submodule (v.valuationSubringAtPrime K) V} {s : O} (hs : s ∉ v.asIdeal) {x : V} (hx : s • x ∈ M) : x ∈ M

theorem **mem_localRing_iff** (v : HeightOneSpectrum O) (x : K) : x ∈ v.valuationSubringAtPrime K ↔ ∃ s ∉ v.asIdeal, s • x ∈ (1 : Submodule O K)

theorem **mem_of_forall_mem_localize** {L : Submodule O V} {x : V} (hx : x ∈ Submodule.span K (L : Set V)) (h : ∀ v : HeightOneSpectrum O, x ∈ localize K v L) : x ∈ L

theorem **mem_of_forall_mem_localize_of_not_isField** (hO : ¬ IsField O) {L : Submodule O V} {x : V} (h : ∀ v : HeightOneSpectrum O, x ∈ localize K v L) : x ∈ L

theorem **mem_iff_forall_mem_localize** {L : Submodule O V} (hL : Submodule.span K (L : Set V) = ⊤) (x : V) : x ∈ L ↔ ∀ v : HeightOneSpectrum O, x ∈ localize K v L

theorem **iInf_localize** {L : Submodule O V} (hL : Submodule.span K (L : Set V) = ⊤) : ⨅ v : HeightOneSpectrum O, (localize K v L).restrictScalars O = L

theorem **eventually_notMem** {d : O} (hd : d ∈ O⁰) : ∀ᶠ v : HeightOneSpectrum O in Filter.cofinite, d ∉ v.asIdeal

theorem **eventually_localize_le** {L L' : Submodule O V} (hL : L.FG) (hL' : Submodule.span K (L' : Set V) = ⊤) : ∀ᶠ v : HeightOneSpectrum O in Filter.cofinite, localize K v L ≤ localize K v L'

theorem **eventually_localize_eq** (L L' : Submodule O V) [h : Submodule.IsLattice K L] [h' : Submodule.IsLattice K L'] : ∀ᶠ v : HeightOneSpectrum O in Filter.cofinite, localize K v L = localize K v L'

theorem **exists_mem_submonoid_forall** {ι : Type*} {T : Set ι} (hT : T.Finite) (S : Submonoid O) (P : ι → O → Prop) (hP : ∀ i s t, P i s → P i (t * s)) (h : ∀ i ∈ T, ∃ s ∈ S, P i s) : ∃ s ∈ S, ∀ i ∈ T, P i s

theorem **exists_notMem_smul_mem** {v w : HeightOneSpectrum O} (hvw : w ≠ v) {M : Submodule (w.valuationSubringAtPrime K) V} {d : O} (hd : d ∈ O⁰) {x : V} (hx : d • x ∈ M) : ∃ s ∉ v.asIdeal, s • x ∈ M

theorem **mem_glue** {M : ∀ v : HeightOneSpectrum O, Submodule (v.valuationSubringAtPrime K) V} {x : V} : x ∈ glue M ↔ ∀ v, x ∈ M v

theorem **glue_localize** {L : Submodule O V} (hL : Submodule.span K (L : Set V) = ⊤) : glue (fun v => localize K v L) = L

theorem **localize_glue** {M : ∀ v : HeightOneSpectrum O, Submodule (v.valuationSubringAtPrime K) V} (hM : Submodule.span K (glue M : Set V) = ⊤) (v : HeightOneSpectrum O) : localize K v (glue M) = M v

theorem **isLattice_glue** (L : Submodule O V) [hL : Submodule.IsLattice K L] (M : ∀ v : HeightOneSpectrum O, Submodule (v.valuationSubringAtPrime K) V) (hM : ∀ v, Submodule.IsLattice K (M v)) (hML : ∀ᶠ v in Filter.cofinite, M v = localize K v L) : Submodule.IsLattice K (glue M)

theorem **mem_frameLattice_iff** (g : GL ι K) (x : ι → K) : x ∈ frameLattice A g ↔ ∃ a : ι → A, (g : Matrix ι ι K) *ᵥ (fun i => algebraMap A K (a i)) = x

theorem **mem_frameLattice_iff_inv** (g : GL ι K) (x : ι → K) : x ∈ frameLattice A g ↔ ∀ i, (((g⁻¹ : GL ι K) : Matrix ι ι K) *ᵥ x) i ∈ (1 : Submodule A K)

theorem **exists_eq_frameLattice** [IsDomain A] [IsPrincipalIdealRing A] [IsFractionRing A K] (M : Submodule A (ι → K)) [Submodule.IsLattice K M] : ∃ g : GL ι K, M = frameLattice A g

theorem **mem_one_localRing** (v : HeightOneSpectrum O) (x : K) : x ∈ (1 : Submodule (v.valuationSubringAtPrime K) K) ↔ x ∈ v.valuationSubringAtPrime K

@[simp] theorem **toCompletion_apply** (x : ι → K) (i : ι) : toCompletion v x i = algebraMap K (v.adicCompletion K) (x i)

theorem **algebraMap_mem_adicCompletionIntegers_iff** (x : K) : algebraMap K (v.adicCompletion K) x ∈ v.adicCompletionIntegers K ↔ x ∈ v.valuationSubringAtPrime K

theorem **exists_sub_mem_adicCompletionIntegers** (y : v.adicCompletion K) : ∃ x : K, y - algebraMap K (v.adicCompletion K) x ∈ v.adicCompletionIntegers K

theorem **mem_rationalPart** {N : Submodule (v.adicCompletionIntegers K) (ι → v.adicCompletion K)} {x : ι → K} : x ∈ rationalPart v N ↔ toCompletion v x ∈ N

theorem **complete_le_iff** {M : Submodule (v.valuationSubringAtPrime K) (ι → K)} {N : Submodule (v.adicCompletionIntegers K) (ι → v.adicCompletion K)} : complete v M ≤ N ↔ M ≤ rationalPart v N

theorem **complete_span** (s : Set (ι → K)) : complete v (Submodule.span (v.valuationSubringAtPrime K) s) = Submodule.span _ (toCompletion v '' s)

theorem **toCompletion_mulVec** (g : Matrix ι ι K) (x : ι → K) : toCompletion v (g *ᵥ x) = g.map (algebraMap K (v.adicCompletion K)) *ᵥ toCompletion v x

theorem **complete_frameLattice** (g : GL ι K) : complete v (frameLattice (v.valuationSubringAtPrime K) g) = frameLattice (v.adicCompletionIntegers K) (Matrix.GeneralLinearGroup.map (algebraMap K (v.adicCompletion K)) g)

theorem **rationalPart_frameLattice** (g : GL ι K) : rationalPart v (frameLattice (v.adicCompletionIntegers K) (Matrix.GeneralLinearGroup.map (algebraMap K (v.adicCompletion K)) g)) = frameLattice (v.valuationSubringAtPrime K) g

theorem **rationalPart_complete** (M : Submodule (v.valuationSubringAtPrime K) (ι → K)) [Submodule.IsLattice K M] : rationalPart v (complete v M) = M

theorem **exists_smul_mem_completion** {N : Submodule (v.adicCompletionIntegers K) (ι → v.adicCompletion K)} (hN : Submodule.span (v.adicCompletion K) (N : Set (ι → v.adicCompletion K)) = ⊤) (y : ι → v.adicCompletion K) : ∃ d ∈ O⁰, d • y ∈ N

theorem **exists_smul_le_completion** {N N' : Submodule (v.adicCompletionIntegers K) (ι → v.adicCompletion K)} (hN : N.FG) (hN' : Submodule.span (v.adicCompletion K) (N' : Set (ι → v.adicCompletion K)) = ⊤) : ∃ d ∈ O⁰, ∀ y ∈ N, d • y ∈ N'

theorem **complete_rationalPart_of_span_eq_top** {N : Submodule (v.adicCompletionIntegers K) (ι → v.adicCompletion K)} (hN : Submodule.span (v.adicCompletion K) (N : Set (ι → v.adicCompletion K)) = ⊤) : complete v (rationalPart v N) = N

theorem **complete_rationalPart** (N : Submodule (v.adicCompletionIntegers K) (ι → v.adicCompletion K)) [hN : Submodule.IsLattice (v.adicCompletion K) N] : complete v (rationalPart v N) = N

theorem **isLattice_iff** {K : Type} [Field K] [NumberField K] {r : ℕ} (L : Submodule (𝓞 K) (Fin r → K)) : ReciprocityTheory.MultiplicativeHeights.IsLattice L ↔ Submodule.IsLattice K L

theorem **mem_one_ringOfIntegers_iff** {K : Type*} [Field K] [NumberField K] (x : K) : x ∈ (1 : Submodule (𝓞 K) K) ↔ IsIntegral ℤ x

theorem **mem_iff_forall_mem_localize_ringOfIntegers** {K V : Type*} [Field K] [NumberField K] [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] (L : Submodule (𝒪 K) V) (x : V) : x ∈ L ↔ ∀ v : HeightOneSpectrum (𝒪 K), x ∈ localize K v L

@[simp] theorem **hermPairing_apply** (G : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : hermPairing G x y = herm G x y

theorem **hermPairing_separating** {G : Matrix (Fin r) (Fin r) E} (hG : G.det ≠ 0) (x : Fin r → E) (hx : ∀ y, hermPairing G x y = 0) : x = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemLatOIffExistsActO

theorem **finEmb_tmul** (e : D.host.E) (α : Af D.host.F) : finEmb D (e ⊗ₜ[D.host.F] α) = (pairA D.host.F 0 α) ⊗ₜ[D.host.F] e

theorem **toEAf_tmul** (x : AF D.host.F) (e : D.host.E) : toEAf D (x ⊗ₜ[D.host.F] e) = e ⊗ₜ[D.host.F] (sndA D.host.F x)

theorem **finEmb_mul** (a : AE D) (x : EAf D) : finEmb D (toEAf D a * x) = a * finEmb D x

theorem **cF_tmul** (e : D.host.E) (α : Af D.host.F) : cF D (e ⊗ₜ[D.host.F] α) = star e ⊗ₜ[D.host.F] α

theorem **toEAf_cA** (a : AE D) : toEAf D (cA D a) = cF D (toEAf D a)

theorem **toEAf_incA** (e : D.host.E) : toEAf D (incA D e) = eToEAf D e

theorem **rankOne_add_left** (S T : HSp D) (a : D.host.E) (s s' : Fin S.rank → EAf D) (t : Fin T.rank → EAf D) : rankOne S T a (s + s') t = rankOne S T a s t + rankOne S T a s' t

theorem **rankOne_add_right** (S T : HSp D) (a : D.host.E) (s : Fin S.rank → EAf D) (t t' : Fin T.rank → EAf D) : rankOne S T a s (t + t') = rankOne S T a s t + rankOne S T a s t'

theorem **smul_mem_closure** {n : ℕ} (Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) (r : EAf D) (hr : r ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) : ∀ y ∈ AddSubgroup.closure (pureSet Λ), r • y ∈ AddSubgroup.closure (pureSet Λ)

theorem **mem_closure_of_mem_hatL** {n : ℕ} (Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) (y : Fin n → EAf D) (hy : y ∈ Formalism.L2.Herm.hatL (F

theorem **pure_mem_hatL** {n : ℕ} (Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) (α : Af D.host.F) (hα : α ∈ Formalism.UG.intAd D.host.F) (x : Fin n → D.host.E) (hx : x ∈ Λ) : (fun i => x i ⊗ₜ[D.host.F] α) ∈ Formalism.L2.Herm.hatL (F

theorem **rankOne_pure** (S T : HSp D) (a : D.host.E) (s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) (α β : Af D.host.F) : rankOne S T a (fun i => s i ⊗ₜ[D.host.F] α) (fun i => t i ⊗ₜ[D.host.F] β) = Matrix.of fun i j => (pairA D.host.F 0 (β * α)) ⊗ₜ[D.host.F] (a * t i * ∑ k, star (s k) * S.gram k j)

theorem **rankOne_mem_latO** (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (a : D.host.E) (ha : a ∈ D.host.AA) (s : Fin S.rank → EAf D) (hs : s ∈ Formalism.L2.Herm.hatL (F

theorem **mt_mul** (V : HSp D) (g g' : GA D V) : mt V (g * g') = mt V g * mt V g'

theorem **mt_one** (V : HSp D) : mt V 1 = 1

theorem **gram_mul_inv** (V : HSp D) (g : GA D V) : V.gram.map (incA D) * mt V g⁻¹ = ((mt V g).map (cA D))ᵀ * V.gram.map (incA D)

theorem **actO_rankOne** (S T : HSp D) (g : GA D S) (h : GA D T) (a : D.host.E) (s : Fin S.rank → EAf D) (t : Fin T.rank → EAf D) : actO S T g h (rankOne S T a s t) = rankOne S T a ((mt S g).map (toEAf D) *ᵥ s) ((mt T h).map (toEAf D) *ᵥ t)

theorem **mulVec_mem_hatL** {n : ℕ} (M : Matrix (Fin n) (Fin n) (EAf D)) (Λ Λ' : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) (β β' : GL (Fin n) (EAf D)) (hβ : Formalism.L2.DIV.IsFrame β Λ) (hβ' : Formalism.L2.DIV.IsFrame β' Λ') (hM : β'.val = M * β.val) (x : Fin n → EAf D) (hx : x ∈ Formalism.L2.Herm.hatL (F

theorem **actO_mem_latO** (S T : HSp D) (g : GA D S) (h : GA D T) (ΛS ΛS' : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT ΛT' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS βS' : GL (Fin S.rank) (EAf D)) (βT βT' : GL (Fin T.rank) (EAf D)) (hS : Formalism.L2.DIV.IsFrame βS ΛS) (hS' : Formalism.L2.DIV.IsFrame βS' ΛS') (hT : Formalism.L2.DIV.IsFrame βT ΛT) (hT' : Formalism.L2.DIV.IsFrame βT' ΛT') (hg : βS'.val = (mt S g).map (toEAf D) * βS.val) (hh : βT'.val = (mt T h).map (toEAf D) * βT.val) (u : WO S T) (hu : u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) : actO S T g h u ∈ latO S T (ΛS' : Set (Fin S.rank → D.host.E)) (ΛT' : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **frame_inv** (V : HSp D) (g : GA D V) (β β' : GL (Fin V.rank) (EAf D)) (hg : β'.val = (mt V g).map (toEAf D) * β.val) : β.val = (mt V g⁻¹).map (toEAf D) * β'.val

theorem **actO_actO_inv** (S T : HSp D) (g : GA D S) (h : GA D T) (u : WO S T) : actO S T g h (actO S T g⁻¹ h⁻¹ u) = u

theorem **mem_latO_iff_exists_actO** : ∀ (D : D7) (S T : HSp D) (g : GA D S) (h : GA D T) (ΛS ΛS' : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT ΛT' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS βS' : GL (Fin S.rank) (EAf D)) (βT βT' : GL (Fin T.rank) (EAf D)), Formalism.L2.DIV.IsFrame βS ΛS → Formalism.L2.DIV.IsFrame βS' ΛS' → Formalism.L2.DIV.IsFrame βT ΛT → Formalism.L2.DIV.IsFrame βT' ΛT' → Submodule.IsLattice D.host.E ΛS → Submodule.IsLattice D.host.E ΛS' → Submodule.IsLattice D.host.E ΛT → Submodule.IsLattice D.host.E ΛT' → βS'.val = (mt S g).map (toEAf D) * βS.val → βT'.val = (mt T h).map (toEAf D) * βT.val → ∀ u : WO S T, u ∈ (latO S T (ΛS' : Set (Fin S.rank → D.host.E)) (ΛT' : Set (Fin T.rank → D.host.E)) fun _ => True) ↔ ∃ u₀ ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), u = actO S T g h u₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemPowsSUnitsChooseT

theorem **relIndex_sup_zpowers_chooseT** (p : ℕ) {G : Type} [CommGroup G] (Q : Subgroup G) (d : G) (hp : p.Prime) (hd : d ^ p ∈ Q) (hn : d ∉ Q) : Q.relIndex (Q ⊔ Subgroup.zpowers d) = p

theorem **mem_pows_sUnits_chooseT** (p : ℕ) (hp : p ≠ 0) (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (x : Fˣ) (hx : x ∈ sUnits F S) (r : F) (hr : r ^ p = (x : F)) : x ∈ pows p (sUnits F S)

theorem **exists_auxiliary_places_prime** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → ∃ T : Finset (HeightOneSpectrum (𝒪 F)), Disjoint S T ∧ T.card + 1 = S.card + Fintype.card (InfinitePlace F) ∧ (∀ v ∈ T, ∃ r : v.adicCompletion F, r ^ p = algebraMap F (v.adicCompletion F) (δ : F)) ∧ ∀ v ∈ T, ∃ u ∈ sUnits F S, u ∉ powSub p (algebraMap F (v.adicCompletion F)) ∧ ∀ v' ∈ T, v' ≠ v → u ∈ powSub p (algebraMap F (v'.adicCompletion F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemPowsSUnitsChooseTStatement

theorem **_root_.GlobalTheory.T.exists_places_with_pth_roots_and_separating_units** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → ∃ T : Finset (HeightOneSpectrum (𝒪 F)), Disjoint S T ∧ T.card + 1 = S.card + Fintype.card (InfinitePlace F) ∧ (∀ v ∈ T, ∃ r : v.adicCompletion F, r ^ p = algebraMap F (v.adicCompletion F) (δ : F)) ∧ ∀ v ∈ T, ∃ u ∈ sUnits F S, u ∉ powSub p (algebraMap F (v.adicCompletion F)) ∧ ∀ v' ∈ T, v' ≠ v → u ∈ powSub p (algebraMap F (v'.adicCompletion F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemRangeAlgebraMapOfConjFixed

theorem **mem_range_algebraMap_of_conj_fixed** (F E : Type) [Field F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) (hE : IsTotallyComplex E) (h2 : Module.finrank F E = 2) (hσ : ∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) (e : E) (he : σ e = e) : ∃ c : F, algebraMap F E c = e

theorem **exists_mul_conj_eq_algebraMap** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) (hE : IsTotallyComplex E) (h2 : Module.finrank F E = 2) (hσ : ∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) (R : Type) [CommRing R] [Algebra F R] (y : R ⊗[F] E) : ∃ n : R, y * Algebra.TensorProduct.map (AlgHom.id F R) σ y = algebraMap R (R ⊗[F] E) n

theorem **exists_unit_mul_conj_eq_algebraMap** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsTotallyReal F → IsTotallyComplex E → Module.finrank F E = 2 → (∀ (τ : E →+* ℂ) (x : E), τ (σ x) = star (τ x)) → ∀ (R : Type) [CommRing R] [Algebra F R] (x : (R ⊗[F] E)ˣ), ∃ n : Rˣ, (x : R ⊗[F] E) * Algebra.TensorProduct.map (AlgHom.id F R) σ (x : R ⊗[F] E) = algebraMap R (R ⊗[F] E) n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemRangeAlgebraMapOfIsNilpotentTmulSub

theorem **exists_algHom_algebraicClosure_ker_le** (K A : Type) [Field K] [CommRing A] [Algebra K A] [Algebra.FiniteType K A] (m : Ideal A) [m.IsMaximal] : ∃ φ : A →ₐ[K] AlgebraicClosure K, ∀ x, φ x = 0 → x ∈ m

theorem **mem_range_algebraMap_of_isNilpotent_tmul_sub** (K A : Type) [Field K] [CharZero K] [CommRing A] [Algebra K A] [Algebra.FiniteType K A] [IsReduced A] (f : A) (hf : IsNilpotent (f ⊗ₜ[K] (1 : A) - (1 : A) ⊗ₜ[K] f)) : f ∈ (algebraMap K A).range

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemRangeOfOneTmulMem

theorem **exists_nat_mul_integral** (y : E) : ∃ d : ℕ, 0 < d ∧ (d : E) * y ∈ Set.range (algebraMap (𝒪 E) E)

theorem **mul_mem_hat** {x y : RA F E} (hx : x ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))) (hy : y ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))) : x * y ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))

theorem **exists_eq_one_tmul_add** (z : RA F E) : ∃ (x : E), ∃ h ∈ hat1 F (Set.range (algebraMap (𝒪 E) E)), z = (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] x + h

theorem **mem_range_of_one_tmul_mem** {x : E} (hx : (1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] x ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))) : x ∈ Set.range (algebraMap (𝒪 E) E)

theorem **mem_hat_of_idempotent** (z : RA F E) (hz : z * z = z) : z ∈ hat1 F (Set.range (algebraMap (𝒪 E) E))

theorem **idempotent_mem_intAE** : ∀ (D : D7) (z : Af D.host.F ⊗[D.host.F] D.host.E), z * z = z → z ∈ intAE D

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemRatZariskiClosurePointsIffOfIsReduced

theorem **mem_pts_zeroSet_of_isReduced** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (S' : Set (MvPolynomial (ι × ι) ℚ)) (R : Type) [CommRing R] [Algebra ℚ R] (y : Matrix ι ι R), (y ∈ pts {g : Matrix ι ι ℂ | ∀ f ∈ S', ev f g = 0} R → ∀ f ∈ S', ev f y = 0) ∧ (IsReduced R → (∀ f ∈ S', ev f y = 0) → y ∈ pts {g : Matrix ι ι ℂ | ∀ f ∈ S', ev f g = 0} R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemRatZariskiClosurePointsIffOfIsReducedStatement

theorem **_root_.GlobalTheory.T.mem_ratZariskiClosurePoints_iff_of_isReduced** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (S' : Set (MvPolynomial (ι × ι) ℚ)) (R : Type) [CommRing R] [Algebra ℚ R] (y : Matrix ι ι R), (y ∈ pts {g : Matrix ι ι ℂ | ∀ f ∈ S', ev f g = 0} R → ∀ f ∈ S', ev f y = 0) ∧ (IsReduced R → (∀ f ∈ S', ev f y = 0) → y ∈ pts {g : Matrix ι ι ℂ | ∀ f ∈ S', ev f g = 0} R)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemSpanRatPolysVanishingOfAutStable

theorem **evC_map_ringEquiv** {ι : Type} (σ : ℂ ≃+* ℂ) (f : MvPolynomial (ι × ι) ℂ) (g : Matrix ι ι ℂ) : evC (MvPolynomial.map (σ : ℂ →+* ℂ) f) g = σ (evC f (g.map σ.symm))

theorem **evC_map_rat** {ι : Type} (f : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι ℂ) : evC (MvPolynomial.map (algebraMap ℚ ℂ) f) g = ev f g

theorem **mem_span_vanI_of_vanish** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ g ∈ Γ, evC f g = 0) → f ∈ Submodule.span ℂ ((MvPolynomial.map (algebraMap ℚ ℂ)) '' vanI Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemSpanRatPolysVanishingOfAutStableStatement

theorem **_root_.GlobalTheory.T.mem_span_ratPolysVanishing_of_autStable** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ g ∈ Γ, evC f g = 0) → f ∈ Submodule.span ℂ ((MvPolynomial.map (algebraMap ℚ ℂ)) '' vanI Γ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemSpanRationalKroneckerInvariants

theorem **mem_span_rational** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ (n : ℕ) (l : (Fin n → ι) → ℂ), (∀ g ∈ Γ, l ᵥ* tpow n g = l) → l ∈ Submodule.span ℂ {μ : (Fin n → ι) → ℂ | (∀ g ∈ Γ, μ ᵥ* tpow n g = μ) ∧ ∀ a, ∃ q : ℚ, μ a = q}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemSpanRationalKroneckerInvariantsStatement

theorem **_root_.GlobalTheory.T.mem_span_rational_kroneckerInvariants** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ (n : ℕ) (l : (Fin n → ι) → ℂ), (∀ g ∈ Γ, l ᵥ* tpow n g = l) → l ∈ Submodule.span ℂ {μ : (Fin n → ι) → ℂ | (∀ g ∈ Γ, μ ᵥ* tpow n g = μ) ∧ ∀ a, ∃ q : ℚ, μ a = q}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemUnitsLocalKummerNormsOfBaseChangeNorm

theorem **isUnit_of_norm_ne_zero** {L A : Type} [Field L] [CommRing A] [Algebra L A] [Module.Finite L A] (y : A) (h : Algebra.norm L y ≠ 0) : IsUnit y

theorem **IsKum.nonempty_tensor_equiv** {p : ℕ} {F E : Type} [Field F] [Field E] [Algebra F E] {ζ δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsKum p ζ δ θ σ) (L : Type) [Field L] [Algebra F L] : Nonempty ((L ⊗[F] E) ≃ₐ[L] AdjoinRoot (X ^ p - C (algebraMap F L δ)) : L[X]))

theorem **mem_kNorm_of_tNorm** : ∀ (p : ℕ) (F E L : Type) [Field F] [CharZero F] [Field E] [Algebra F E] [Field L] [Algebra F L] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ x : Fˣ, TNorm F E L (x : F) → x ∈ kNorm p (algebraMap F L) δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MemUnitsLocalKummerNormsOfBaseChangeNormStatement

theorem **_root_.GlobalTheory.T.mem_unitsLocalKummerNorms_of_baseChange_norm** : ∀ (p : ℕ) (F E L : Type) [Field F] [CharZero F] [Field E] [Algebra F E] [Field L] [Algebra F L] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ x : Fˣ, TNorm F E L (x : F) → x ∈ kNorm p

(algebraMap F L) δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MinNormTensorStabilizerConjTransposeInvariant

private lemma **posDef_det_one_exists_unitary_exp_diagonal_sum_zero** {ι : Type} [Fintype ι] [DecidableEq ι] (P : Matrix ι ι ℂ) (hP : P.PosDef) (hdet : P.det = 1) : ∃ (U : Matrix ι ι ℂ) (δ : ι → ℝ), U * Uᴴ = 1 ∧ ∑ i, δ i = 0 ∧ P = (U * diagonal fun i => ((Real.exp (2 * δ i) : ℝ) : ℂ)) * Uᴴ

private lemma **unitary_conj_exp_diagonal_conjTranspose** {ι : Type} [Fintype ι] [DecidableEq ι] (U : Matrix ι ι ℂ) (δ : ι → ℝ) (s : ℝ) : ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ)ᴴ = (U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ

private lemma **unitary_conj_exp_diagonal_mul_add_exponents** {ι : Type} [Fintype ι] [DecidableEq ι] {U : Matrix ι ι ℂ} (hU : U * Uᴴ = 1) (δ : ι → ℝ) (s t : ℝ) : ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ) * ((U * diagonal fun i => ((Real.exp (t * δ i) : ℝ) : ℂ)) * Uᴴ) = (U * diagonal fun i => ((Real.exp ((s + t) * δ i) : ℝ) : ℂ)) * Uᴴ

private lemma **exists_unitary_exp_diagonal_fixing_of_nsq_eq** {ι : Type} [Fintype ι] [DecidableEq ι] {m : ℕ} (d : Fin m → ℕ) (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (hv : _root_.GlobalTheory.ClassFin.IsMinVec d v) (g : Matrix ι ι ℂ) (hg : g.det = 1) (hn : _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v g) = _root_.GlobalTheory.ClassFin.nsq v) : ∃ (U : Matrix ι ι ℂ) (δ : ι → ℝ), U * Uᴴ = 1 ∧ ∑ i, δ i = 0 ∧ g * gᴴ = (U * diagonal fun i => ((Real.exp (2 * δ i) : ℝ) : ℂ)) * Uᴴ ∧ ∀ s : ℝ, _root_.GlobalTheory.ClassFin.act d v ((U * diagonal fun i => ((Real.exp (s * δ i) : ℝ) : ℂ)) * Uᴴ) = v

theorem **_root_.GlobalTheory.ClassFin.act_conjTranspose_of_isMinVec** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ), _root_.GlobalTheory.ClassFin.IsMinVec d v → ∀ g : Matrix ι ι ℂ, g.det = 1 → _root_.GlobalTheory.ClassFin.act d v g = v → _root_.GlobalTheory.ClassFin.act d v gᴴ = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MinNormTensorStabilizerConjTransposeInvariantStatement

theorem **_root_.GlobalTheory.T.minNormTensor_stabilizer_conjTranspose_invariant** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ), IsMinVec d v → ∀ g : Matrix ι ι ℂ, g.det = 1 → act d v g = v → act d v gᴴ = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MinkowskiLinearForms

theorem **minkowski_linear_forms** {ι : Type} [Fintype ι] [DecidableEq ι] (A : Matrix ι ι ℝ) (hA : Matrix.det A ≠ (0 : ℝ)) {t : ℝ} (ht : (0 : ℝ) < t) (hlt : |Matrix.det A| < t ^ Fintype.card ι) : ∃ (z : ι → ℤ), z ≠ (0 : ι → ℤ) ∧ ∀ (j : ι), |∑ i : ι, (↑(z i) : ℝ) * A i j| < t

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MovedOfSplitPl

theorem **exists_subfamily** {α : Type} (r : α → α → Prop) (hr : Equivalence r) {m₀ : ℕ} (f : Fin m₀ → α) : ∃ (m : ℕ) (ι : Fin m → Fin m₀), ∀ g i, r g (f i) → ∃! j, r g (f (ι j))

theorem **fm_mul** (A B : Matrix (Fin n) (Fin n) E) : fm F (A * B) = fm F A * fm F B

theorem **fm_one** : fm F (1 : Matrix (Fin n) (Fin n) E) = 1

theorem **Bd.mul** {L : Submodule (𝒪 E) (Fin n → E)} {a : 𝒪 E} {M : Matrix (Fin n) (Fin n) (RA F E)} (h : Bd L a M) (b : 𝒪 E) : Bd L (b * a) M

theorem **Bd.of_sub_one** {L : Submodule (𝒪 E) (Fin n → E)} {a : 𝒪 E} {M : Matrix (Fin n) (Fin n) (RA F E)} (h : Bd L a (M - 1)) : Bd L a M

theorem **cut_inv** (h : GL (Fin n) (RA F E)) (e : RA F E) (he : e • (h.1 - 1) = h.1 - 1) : e • ((h⁻¹).1 - 1) = (h⁻¹).1 - 1

theorem **exists_nat_bd** (L : Submodule (𝒪 E) (Fin n → E)) (hL : L.FG) (hsp : Submodule.span E (L : Set (Fin n → E)) = ⊤) (N : ℕ) (e : FiniteAdeleRing (𝒪 F) F) (he : ∀ v : HeightOneSpectrum (𝒪 F), (N : 𝒪 F) ∈ v.asIdeal → e v = 0) (M : Matrix (Fin n) (Fin n) (RA F E)) (hM : (e ⊗ₜ[F] (1 : E) : RA F E) • (M - 1) = M - 1) : ∃ t : ℕ, t.Coprime N ∧ Bd L (t : 𝒪 E) M

theorem **exists_supp_bd** (L : Submodule (𝒪 E) (Fin n → E)) (hL : L.FG) (hsp : Submodule.span E (L : Set (Fin n → E)) = ⊤) (e : FiniteAdeleRing (𝒪 F) F) (M : Matrix (Fin n) (Fin n) (RA F E)) (hM : (e ⊗ₜ[F] (1 : E) : RA F E) • (M - 1) = M - 1) : ∃ s : 𝒪 F, s ≠ 0 ∧ (∀ v : HeightOneSpectrum (𝒪 F), s ∈ v.asIdeal → e v ≠ 0) ∧ Bd L (algebraMap (𝒪 F) (𝒪 E) s) M

theorem **smul_mem_of_bd** {L Λ : Submodule (𝒪 E) (Fin n → E)} (hL : L.FG) (hsp : Submodule.span E (L : Set (Fin n → E)) = ⊤) {h : GL (Fin n) (RA F E)} (hΛ : (Λ : Set (Fin n → E)) = latOf F h.1 L) {a : 𝒪 E} (h1 : Bd L a h.1) (h2 : Bd L a (h⁻¹).1) : (∀ v ∈ Λ, a • v ∈ L) ∧ ∀ v ∈ L, a • v ∈ Λ

theorem **mem_conjLat** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Λ : Submodule (𝒪 E) (Fin n → E)) (v : Fin n → E) : v ∈ (conjLat H Λ).La ↔ (fun j => IsCMField.complexConj E (v j)) ∈ Λ

theorem **conj_mem_range** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) {x : E} (hx : x ∈ (1 : Submodule (𝒪 E) E)) : IsCMField.complexConj E x ∈ (1 : Submodule (𝒪 E) E)

theorem **conj_mem_std** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) {v : Fin n → E} (hv : v ∈ stdLat E n) : (fun j => IsCMField.complexConj E (v j)) ∈ stdLat E n

theorem **conjLat_le** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) {Λ : Submodule (𝒪 E) (Fin n → E)} {a : 𝒪 E} (ha : IsCMField.complexConj E (a : E) = a) (h : (∀ v ∈ Λ, a • v ∈ stdLat E n) ∧ ∀ v ∈ stdLat E n, a • v ∈ Λ) : Ideal.span {a} • (conjLat H Λ).La ≤ stdLat E n ∧ Ideal.span {a} • stdLat E n ≤ (conjLat H Λ).La

theorem **sum_one** (x y : Fin n → E) : ∑ a, ∑ b, x a * (1 : Matrix (Fin n) (Fin n) E) a b * IsCMField.complexConj E (y b) = ∑ a, x a * IsCMField.complexConj E (y a)

theorem **conjLat_unimod** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) {Λ : Submodule (𝒪 E) (Fin n → E)} {h : GL (Fin n) (RA (maximalRealSubfield E) E)} (hh : IsUnA (IsCMField.complexConj E) 1 h.1) (hΛ : (Λ : Set (Fin n → E)) = latOf (maximalRealSubfield E) h.1 (stdLat E n)) : (conjLat H Λ).Unimod

theorem **moved_of_splitPl** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (v : HeightOneSpectrum (𝒪 (maximalRealSubfield E))) (hv : SplitPl (IsCMField.complexConj E) v) (P : Ideal (𝒪 E)) (hP : P.IsPrime) (hPv : P.comap (algebraMap (𝒪 (maximalRealSubfield E)) (𝒪 E)) = v.asIdeal) : Moved (IsCMField.complexConj E) P

theorem **exists_lat** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (l : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H) (bad : ℕ) (h2 : 2 ∣ bad) (h : GL (Fin n) (RA (maximalRealSubfield E) E)) (hh : IsUnA (IsCMField.complexConj E) 1 h.1) (hs : SuppSplit (IsCMField.complexConj E) (bad * l.M) h.1) : ∃ (Λ : Submodule (𝒪 E) (Fin n → E)) (N : ℕ), (Λ : Set (Fin n → E)) = latOf (maximalRealSubfield E) h.1 (stdLat E n) ∧ (conjLat H Λ).Unimod ∧ (conjLat H Λ).TrivAt bad ∧ (conjLat H Λ).TrivAt l.M ∧ (conjLat H Λ).LevOK N l.M ∧ GenLat.SplitSupp H (conjLat H Λ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MulLogEqNegLogMul

theorem **mul_diagonal_eq_neg_of_exp** {ι : Type} [Fintype ι] [DecidableEq ι] (y : Matrix ι ι ℂ) (d d' : ι → ℝ) (h : diagonal (fun i => (Real.exp (d' i) : ℂ)) * y * diagonal (fun i => (Real.exp (d i) : ℂ)) = y) : y * diagonal (fun i => (d i : ℂ)) = -(diagonal (fun i => (d' i : ℂ)) * y)

theorem **mul_log_eq_neg_log_mul** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (x x' H H' : Matrix ι ι ℂ), IsLog H (xᴴ * x) → IsLog H' (x'ᴴ * x') → x' * x = 1 → x * H = -(H' * x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.MulLogEqNegLogMulStatement

theorem **_root_.GlobalTheory.T.mul_log_eq_neg_log_mul'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (x x' H H' : Matrix ι ι ℂ), IsLog H (xᴴ * x) → IsLog H' (x'ᴴ * x') → x' * x = 1 → x * H = -(H' * x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NcardPlacesOverEqOfPrimeDegree

theorem **ncard_places_over_eq_of_prime_degree** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E], p.Prime → Module.finrank F E = p → ∀ (v : HeightOneSpectrum (𝒪 F)) (w w' : HeightOneSpectrum (𝒪 E)), w ≠ w' → w.under (𝒪 F) = v → w'.under (𝒪 F) = v → Set.ncard {u : HeightOneSpectrum (𝒪 E) | u.under (𝒪 F) = v} = p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NormCoeffMinpolyLeOneOfNormLeOne

theorem **norm_coeff_minpoly_le_one_of_norm_le_one** (K : Type) [NontriviallyNormedField K] [IsUltrametricDist K] [CompleteSpace K] (L : Type) [Field L] [Algebra K L] [FiniteDimensional K L] (x : L) (h : ‖Algebra.norm K x‖ ≤ 1) (n : ℕ) : ‖(minpoly K x).coeff n‖ ≤ 1

theorem **isIntegral_iff_valued_norm_le_one** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (L : Type) [Field L] [Algebra (v.adicCompletion F) L] [FiniteDimensional (v.adicCompletion F) L] [Algebra (v.adicCompletionIntegers F) L] [IsScalarTower (v.adicCompletionIntegers F) (v.adicCompletion F) L] (x : L) : IsIntegral (v.adicCompletionIntegers F) x ↔ Valued.v (Algebra.norm (v.adicCompletion F) x) ≤ 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NormDifferenceIndexIdentity

theorem **herb_relIndex_map_range** {A : Type} [CommGroup A] (f : A →* A) (H : Subgroup A) : (H.map f).relIndex f.range = (H ⊔ f.ker).index

theorem **herb_lattice** {A : Type} [CommGroup A] (B C K : Subgroup A) (hCK : C ≤ K) : (C ⊔ B).index * Nat.card ↥(B ⊓ K) = (B ⊔ K).index * C.relIndex K * Nat.card ↥(C ⊓ B)

theorem **herb_nrm_apply_sigma** {A : Type} [CommGroup A] (σ : A →* A) (p : ℕ) (h : ∀ a, σ^[p] a = a) (a : A) : nrm σ p (σ a) = nrm σ p a

theorem **herb_nrm_of_fix** {A : Type} [CommGroup A] (σ : A →* A) (p : ℕ) (b : A) (hb : b ∈ fix σ) : nrm σ p b = b ^ p

theorem **herb_index_identity** : ∀ (p : ℕ) (A : Type) [CommGroup A] (σ : A →* A), (∀ a, σ^[p] a = a) → (nrm σ p).range.relIndex (fix σ) * ((dif σ).range.map (dif σ)).relIndex (dif σ).range * pTor p (fix σ) = pIdx p (fix σ) * (dif σ).range.relIndex (nrm σ p).ker * Nat.card ↥((dif σ).range ⊓ fix σ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NormDifferenceIndexIdentityStatement

theorem **_root_.GlobalTheory.T.norm_difference_index_identity** : ∀ (p : ℕ) (A : Type) [CommGroup A] (σ : A →* A), (∀ a, σ^[p] a = a) → (nrm σ p).range.relIndex (fix σ) * ((dif σ).range.map (dif σ)).relIndex (dif σ).range * pTor p (fix σ) = pIdx p (fix σ) * (dif σ).range.relIndex (nrm σ p).ker * Nat.card ↥((dif σ).range ⊓ fix σ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NormOneApproximationAtLevelPrimes

lemma **exists_split** (N : ℕ) : ∀ M : ℕ, 0 < M → ∃ D₁ D₂ k : ℕ, M = D₁ * D₂ ∧ D₁ ∣ N ^ k ∧ D₂.Coprime N

lemma **exists_nat_multiple** (α : 𝒪 E) (h : α ≠ 0) : ∃ n : ℕ, 0 < n ∧ ∃ γ : 𝒪 E, (n : 𝒪 E) = α * γ

lemma **one_add_ne_zero** (N : ℕ) (hN : N ≠ 1) (z : 𝒪 E) : 1 + (N : 𝒪 E) * z ≠ 0

lemma **exists_coprime_multiple** (N : ℕ) (hN : 0 < N) (z : 𝒪 E) (hw : 1 + (N : 𝒪 E) * z ≠ 0) : ∃ t : ℕ, t.Coprime N ∧ ∃ γ : 𝒪 E, (t : 𝒪 E) = (1 + (N : 𝒪 E) * z) * γ

lemma **val_eq_one** (P : HeightOneSpectrum (𝒪 E)) (d : 𝒪 E) (hd : d ∉ P.asIdeal) : P.valuation E (d : E) = 1

lemma **val_le_of_mul** (P : HeightOneSpectrum (𝒪 E)) (x y : E) (d : 𝒪 E) (hd : d ∉ P.asIdeal) (h : x * d = y) (hy : P.valuation E y ≤ 1) : P.valuation E x ≤ 1

lemma **val_le_of_unit** (P : HeightOneSpectrum (𝒪 E)) (y : E) (D : ℕ) (hD : (D : 𝒪 E) ∉ P.asIdeal) (hy : P.valuation E y = 1) : P.valuation E ((y - 1) / D) ≤ 1

lemma **val_le_of_cong** (P : HeightOneSpectrum (𝒪 E)) (N D D₁ D₂ k n : ℕ) (hD0 : 0 < D) (hM : D * n = D₁ * D₂) (h1 : D₁ ∣ N ^ k) (hN : (N : 𝒪 E) ∉ P.asIdeal) (hD : (D : 𝒪 E) ∈ P.asIdeal) (α β p b : 𝒪 E) (hab : p * D₂ + b = 1) : P.valuation E ((((p * D₂ * α + b : 𝒪 E) : E) / ((p * D₂ * β + b : 𝒪 E) : E) - 1) / D) ≤ 1

lemma **exists_conj** (c : E ≃ₐ[F] E) (x : 𝒪 E) : ∃ y : 𝒪 E, (y : E) = c x

lemma **hilbert90_matrix_subalgebra_norm_one** (c : E ≃ₐ[F] E) (hc : ∀ x, c (c x) = x) (hc1 : c ≠ 1) (u : E) (hu : u * c u = 1) : ∃ α : 𝒪 E, α ≠ 0 ∧ u * c α = α

theorem **GlobalTheory.UnRep.normOne_approx** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E), (∀ x, c (c x) = x) → c ≠ 1 → ∀ (u : E) (N D : ℕ), u * c u = 1 → 0 < N → 0 < D → ∃ u' : E, u' * c u' = 1 ∧ (∃ t : ℕ, t.Coprime N ∧ ∃ b : 𝒪 E, (t : E) * (u' / u - 1) = (D : E) * (b : E)) ∧ ∃ s : 𝒪 E, SplitOr c N s ∧ (∃ b : 𝒪 E, (s : E) * (u' - 1) = (D : E) * (b : E)) ∧ ∃ b : 𝒪 E, (s : E) * (u'⁻¹ - 1) = (D : E) * (b : E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NormOneApproximationAtLevelPrimesStatement

theorem **_root_.GlobalTheory.T.normOne_approximation_at_level_primes** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E), (∀ x, c (c x) = x) → c ≠ 1 → ∀ (u : E) (N D : ℕ), u * c u = 1 → 0 < N → 0 < D → ∃ u' : E, u' * c u' = 1 ∧ (∃ t : ℕ, t.Coprime N ∧ ∃ b : 𝒪 E, (t : E) * (u' / u - 1) = (D : E) * (b : E)) ∧ ∃ s : 𝒪 E, SplitOr c N s ∧ (∃ b : 𝒪 E, (s : E) * (u' - 1) = (D : E) * (b : E)) ∧ ∃ b : 𝒪 E, (s : E) * (u'⁻¹ - 1) = (D : E) * (b : E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NormOneInfinite

theorem **normOne_infinite** {F E : Type} [Field F] [Field E] [CharZero E] [Algebra F E] (c : Gal(E/F)) (hc : ∀ x : E, (c : E → E) ((c : E → E) x) = x) (h1 : c ≠ (1 : Gal(E/F))) : Set.Infinite {l : E | l * (c : E → E) l = (1 : E)}

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NotLinearIndependent

theorem **exists_coord** {U : Type} [AddCommGroup U] [Module ℝ U] [FiniteDimensional ℝ U] (J : U →ₗ[ℝ] U) (hJ : ∀ x, J (J x) = -x) : ∃ (g : ℕ) (e : U ≃ₗ[ℝ] (Fin g → ℂ)), ∀ v, e (J v) = Complex.I • e v

theorem **exists_ratForm** {U : Type} [AddCommGroup U] [Module ℝ U] (om : U →ₗ[ℝ] U →ₗ[ℝ] ℝ) {V : Type} [AddCommGroup V] [Module ℚ V] (p : V →+ U) (L : AddSubgroup V) (hN : ∀ x : V, ∃ N : ℕ, 0 < N ∧ N • x ∈ L) (hint : ∀ x ∈ L, ∀ y ∈ L, ∃ k : ℤ, om (p x) (p y) = k) : ∃ B : LinearMap.BilinForm ℚ V, ∀ u v, (B u v : ℝ) = -om (p u) (p v)

theorem **span_image** {U : Type} [AddCommGroup U] [Module ℝ U] {V : Type} [AddCommGroup V] (p : V →+ U) (hsp : Submodule.span ℝ (Set.range p) = ⊤) (L : AddSubgroup V) (hN : ∀ x : V, ∃ N : ℕ, 0 < N ∧ N • x ∈ L) : Submodule.span ℝ (p '' (L : Set V)) = ⊤

theorem **fg_of_integral** {U : Type} [AddCommGroup U] [Module ℝ U] [FiniteDimensional ℝ U] (om : U →ₗ[ℝ] U →ₗ[ℝ] ℝ) (hom : ∀ x, om x = 0 → x = 0) {V : Type} [AddCommGroup V] (p : V →+ U) (hp : Function.Injective p) (L : AddSubgroup V) (hsp : Submodule.span ℝ (p '' (L : Set V)) = ⊤) (hint : ∀ x ∈ L, ∀ y ∈ L, ∃ k : ℤ, om (p x) (p y) = k) : (AddSubgroup.toIntSubmodule L).FG

theorem **not_linearIndependent** : ∀ (U : Type) [NormedAddCommGroup U] [NormedSpace ℝ U] [FiniteDimensional ℝ U] (J : U →ₗ[ℝ] U) (om : U →ₗ[ℝ] U →ₗ[ℝ] ℝ), (∀ x, J (J x) = -x) → (∀ x y, om x y = -om y x) → (∀ x y, om (J x) (J y) = om x y) → (∀ x, x ≠ 0 → om (J x) x < 0) → ∀ (V : Type) [AddCommGroup V] [Module ℚ V] [FiniteDimensional ℚ V] (p : V →+ U), Function.Injective p → Submodule.span ℝ (Set.range p) = ⊤ → ∀ L : AddSubgroup V, (∀ x : V, ∃ N : ℕ, 0 < N ∧ N • x ∈ L) → (∀ x : V, x ∈ L ↔ ∀ y ∈ L, ∃ k : ℤ, om (p x) (p y) = k) → ∀ c : V → ℂ, (∀ v, ‖c v‖ = 1) → ∀ f : Fin 2 → U → ℂ, (∀ k, Differentiable ℝ (f k)) → (∀ k w z, (fderiv ℝ (f k) w z - Real.pi * Complex.I * (om w z : ℂ) * f k w) + Complex.I * (fderiv ℝ (f k) w (J z) - Real.pi * Complex.I * (om w (J z) : ℂ) * f k w) = 0) → (∀ k, ∀ v ∈ L, ∀ z, f k (z + p v) = c v * Complex.exp (-(Real.pi * Complex.I) * (om z (p v) : ℂ)) * f k z) → ¬ Line

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NotLinearIndependentStatement

theorem **_root_.GlobalTheory.T.not_linearIndependent_alt** : ∀ (U : Type) [NormedAddCommGroup U] [NormedSpace ℝ U] [FiniteDimensional ℝ U] (J : U →ₗ[ℝ] U) (om : U →ₗ[ℝ] U →ₗ[ℝ] ℝ), (∀ x, J (J x) = -x) → (∀ x y, om x y = -om y x) → (∀ x y, om (J x) (J y) = om x y) → (∀ x, x ≠ 0 → om (J x) x < 0) → ∀ (V : Type) [AddCommGroup V] [Module ℚ V] [FiniteDimensional ℚ V] (p : V →+ U), Function.Injective p → Submodule.span ℝ (Set.range p) = ⊤ → ∀ L : AddSubgroup V, (∀ x : V, ∃ N : ℕ, 0 < N ∧ N • x ∈ L) → (∀ x : V, x ∈ L ↔ ∀ y ∈ L, ∃ k : ℤ, om (p x) (p y) = k) → ∀ c : V → ℂ, (∀ v, ‖c v‖ = 1) → ∀ f : Fin 2 → U → ℂ, (∀ k, Differentiable ℝ (f k)) → (∀ k w z, (fderiv ℝ (f k) w z - Real.pi * Complex.I * (om w z : ℂ) * f k w) + Complex.I * (fderiv ℝ (f k) w (J z) - Real.pi * Complex.I * (om w (J z) : ℂ) * f k w) = 0) → (∀ k, ∀ v ∈ L, ∀ z, f k (z + p v) = c v * Complex.exp (-(Real.pi * Complex.I) * (om z (p

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NotMemSpanOfCommute

theorem **notMem_span_of_commute** {V : Type} [AddCommGroup V] [Module ℂ V] (G : Set (Module.End ℂ V)) (hc : ∀ a ∈ G, ∀ b ∈ G, Commute a b) (s : Set V) (hs : ∀ x ∈ s, ∃ g ∈ G, g x = 0) (f : V) (hf : ∀ B ∈ Submonoid.closure G, B f ≠ 0) : f ∉ Submodule.span ℂ s

theorem **isHalf_smul** {m n : ℕ} {e : AE D} (he : e * e = e) (u : Mx D m n) : IsHalf e (e • u)

theorem **isHalf_add** {m n : ℕ} {e : AE D} {u v : Mx D m n} (hu : IsHalf e u) (hv : IsHalf e v) : IsHalf e (u + v)

theorem **isHalf_sub** {m n : ℕ} {e : AE D} {u v : Mx D m n} (hu : IsHalf e u) (hv : IsHalf e v) : IsHalf e (u - v)

theorem **isHalf_nsmul** {m n : ℕ} {e : AE D} {u : Mx D m n} (hu : IsHalf e u) (k : ℕ) : IsHalf e (k • u)

theorem **isHalf_zero** {m n : ℕ} (e : AE D) : IsHalf e (0 : Mx D m n)

theorem **half_orth** {e : AE D} (he : IsSplitIdem D e) {x y : WO S T} (hx : IsHalf e x) (hy : IsHalf e y) : symplO S T x y = 0

theorem **heisL_add_half** {e : AE D} (he : IsSplitIdem D e) {x y : WO S T} (hx : IsHalf e x) (hy : IsHalf e y) : heisL ψ S T (x + y) = heisL ψ S T x * heisL ψ S T y

theorem **heisL_commute_half** {e : AE D} (he : IsSplitIdem D e) {x y : WO S T} (hx : IsHalf e x) (hy : IsHalf e y) : Commute (heisL ψ S T x) (heisL ψ S T y)

theorem **psi_skew_ne** {x y : WO S T} (h : ψ (symplO S T x y) ≠ 1) : ψ (symplO S T y x) ≠ 1

theorem **exists_dual_half** {Λ : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) (hd : ∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ Λ ↔ ∀ u' ∈ Λ, ψ (symplO S T u u') = 1)) (hst : ∀ u ∈ Λ, cA D e • u ∈ Λ) {y : WO S T} (hy : IsHalf e y) (hyΛ : y ∉ Λ) : ∃ l ∈ Λ, IsHalf (cA D e) l ∧ ψ (symplO S T l y) ≠ 1

theorem **exists_count_form** {Λ : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) (hd : ∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ Λ ↔ ∀ u' ∈ Λ, ψ (symplO S T u u') = 1)) (hst : ∀ u ∈ Λ, cA D e • u ∈ Λ) {f : WO S T → ℂ} (hf0 : f ≠ 0) (hf : IsLatVac (heisO ψ S T) Λ f) : ∃ κ : (WO S T → ℂ) →ₗ[ℂ] ℂ, κ f = 1 ∧ ∀ y, IsHalf e y → 0 ≤ κ (heisO ψ S T y f)

theorem **exists_invariant_form** {Λ : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) (hd : ∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ Λ ↔ ∀ u' ∈ Λ, ψ (symplO S T u u') = 1)) (hst : ∀ u ∈ Λ, cA D e • u ∈ Λ) {f : WO S T → ℂ} (hf0 : f ≠ 0) (hf : IsLatVac (heisO ψ S T) Λ f) : ∃ ℓ : (WO S T → ℂ) →ₗ[ℂ] ℂ, ℓ f ≠ 0 ∧ ∀ y, IsHalf e y → ∀ φ, IsThetaFunO ψ S T φ → ℓ (heisO ψ S T y φ) = ℓ φ

theorem **sum_reps_invariant** {Λ Λ' : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) {R : Finset (WO S T)} (hR : ∀ u ∈ R, u ∈ Λ' ∧ IsHalf e u) (hR' : ∀ u ∈ Λ', IsHalf e u → ∃! v, v ∈ R ∧ u - v ∈ Λ) {G : WO S T → ℂ} (hG : ∀ d ∈ Λ, d ∈ Λ' → IsHalf e d → heisO ψ S T d G = G) {a : WO S T} (ha : a ∈ Λ') (hae : IsHalf e a) : heisO ψ S T a (∑ u ∈ R, heisO ψ S T u G) = ∑ u ∈ R, heisO ψ S T u G

theorem **isLatVac_sum** {Λ Λ' : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) (hd : ∀ u ∈ Λ', ∀ u' ∈ Λ', ψ (symplO S T u u') = 1) (hst : ∀ u ∈ Λ', e • u ∈ Λ') (hst' : ∀ u ∈ Λ', cA D e • u ∈ Λ') (hoff : ∀ u ∈ Λ', (1 - (e + cA D e)) • u ∈ Λ) {R R' : Finset (WO S T)} (hR : ∀ u ∈ R, u ∈ Λ' ∧ IsHalf e u) (hRu : ∀ u ∈ Λ', IsHalf e u → ∃! v, v ∈ R ∧ u - v ∈ Λ) (hR' : ∀ u ∈ R', u ∈ Λ' ∧ IsHalf (cA D e) u) (hRu' : ∀ u ∈ Λ', IsHalf (cA D e) u → ∃! v, v ∈ R' ∧ u - v ∈ Λ) {f : WO S T → ℂ} (hf : IsLatVac (heisO ψ S T) Λ f) : IsLatVac (heisO ψ S T) Λ' (∑ u ∈ R, ∑ u' ∈ R', heisO ψ S T u (heisO ψ S T u' f))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NpOfFinrankPrimeOfIsPrimitiveRoot

theorem **exists_isKum_of_finrank_prime** (p : ℕ) (F E : Type) [Field F] [Field E] [Algebra F E] [IsGalois F E] (ζ : F) (hp : p.Prime) (hζ : IsPrimitiveRoot ζ p) (hr : Module.finrank F E = p) : ∃ (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ

theorem **np_of_finrank_prime_of_isPrimitiveRoot** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsTotallyComplex F] [IsGalois F E] (ζ : F), p.Prime → IsPrimitiveRoot ζ p → Module.finrank F E = p → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NpOfFinrankTwo

theorem **exists_basis_one_sqrt_of_finrank_two** {F E : Type} [Field F] [CharZero F] [Field E] [Algebra F E] (h : Module.finrank F E = 2) : ∃ (e : Module.Basis (Fin 2) F E) (δ : F), δ ≠ 0 ∧ e 0 = 1 ∧ e 1 * e 1 = algebraMap F E δ

theorem **TNorm.exists_sq_sub** {F E : Type} [Field F] [Field E] [Algebra F E] (e : Module.Basis (Fin 2) F E) (h0 : e 0 = 1) {δ : F} (h1 : e 1 * e 1 = algebraMap F E δ) (L : Type) [Field L] [Algebra F L] {x : F} (h : TNorm F E L x) : ∃ a b : L, algebraMap F L x = a ^ 2 -

algebraMap F L δ * b ^ 2

theorem **np_of_finrank_two** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], Module.finrank F E = 2 → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NpOfFinrankTwoStatement

theorem **_root_.GlobalTheory.T.np_of_finrank_two'** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E], Module.finrank F E = 2 → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NpOfOddPrimeDegree

theorem **tNorm_baseChange_of_ringHom** (F F' E E' K K' : Type) [Field F] [Field F'] [Field E] [Field E'] [Algebra F F'] [Algebra F E] [Algebra F' E'] [Algebra E E'] [Algebra F E'] [IsScalarTower F F' E'] [IsScalarTower F E E'] [Algebra.IsPushout F F' E E'] [FiniteDimensional F E] [Field K] [Algebra F K] [Field K'] [Algebra F' K'] (φ : K →+* K') (hφ : ∀ x : F, φ (algebraMap F K x) = algebraMap F' K' (algebraMap F F' x)) (x : F) (h : TNorm F E K x) : TNorm F' E' K' (algebraMap F F' x)

theorem **exists_norm_eq_of_coprime_pow** {F E : Type} [Field F] [Field E] [Algebra F E] [FiniteDimensional F E] {m n : ℕ} (hc : Nat.Coprime m n) {x : F} (hx : x ≠ 0) {a b : E} (ha : Algebra.norm F a = x ^ m) (hb : Algebra.norm F b = x ^ n) : ∃ y : E, Algebra.norm F y = x

theorem **np_of_odd_prime_degree** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E], p.Prime → p ≠ 2 → Module.finrank F E = p → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.NpOfOddPrimeDegreeStatement

theorem **_root_.GlobalTheory.T.np_of_odd_prime_degree'** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [IsGalois F E], p.Prime → p ≠ 2 → Module.finrank F E = p → NP F E

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OfIntermediateFieldTower

theorem **bijective_mul_out_inv** {H G : Type} [Group H] [Group G] (φ : H →* G) (hφ : Function.Injective φ) : Function.Bijective (fun p : H × (G ⧸ φ.range) => φ p.1 * (p.2.out)⁻¹)

theorem **TNorm.of_tower** (F M E K : Type) [Field F] [Field M] [Field E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [FiniteDimensional F E] [IsGalois F E] [Field K] [Algebra F K] [Algebra M K] [IsScalarTower F M K] (x : F) (hx : TNorm F E K x) : TNorm M E K (algebraMap F M x)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OmAtSmulRight

theorem **omAt_add_left** (S T : HSp D) (τ : D.what₂.Tau) (x x' y : Matrix (Fin T.rank) (Fin S.rank) ℂ) : omAt S T τ (x + x') y = omAt S T τ x y + omAt S T τ x' y

theorem **omAt_add_right** (S T : HSp D) (τ : D.what₂.Tau) (x y y' : Matrix (Fin T.rank) (Fin S.rank) ℂ) : omAt S T τ x (y + y') = omAt S T τ x y + omAt S T τ x y'

theorem **omAt_smul_left** (S T : HSp D) (τ : D.what₂.Tau) (c : ℝ) (x y : Matrix (Fin T.rank) (Fin S.rank) ℂ) : omAt S T τ (c • x) y = c * omAt S T τ x y

theorem **omAt_smul_right** (S T : HSp D) (τ : D.what₂.Tau) (c : ℝ) (x y : Matrix (Fin T.rank) (Fin S.rank) ℂ) : omAt S T τ x (c • y) = c * omAt S T τ x y

theorem **omO_apply** (S T : HSp D) (x y : RW T S) : omO S T x y = ∑ τ, omAt S T τ (x τ) (y τ)

theorem **cplxO_add** {m n : ℕ} (s : Matrix (Fin n) (Fin n) ℂ) (t : Matrix (Fin m) (Fin m) ℂ) (x y : Matrix (Fin m) (Fin n) ℂ) : cplxO s t (x + y) = cplxO s t x + cplxO s t y

theorem **cplxO_smul** {m n : ℕ} (s : Matrix (Fin n) (Fin n) ℂ) (t : Matrix (Fin m) (Fin m) ℂ) (c : ℝ) (x : Matrix (Fin m) (Fin n) ℂ) : cplxO s t (c • x) = c • cplxO s t x

theorem **jO_apply** (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (x : RW T S) (τ : D.what₂.Tau) : jO S T sS sT x τ = cplxO (sS τ) (sT τ) (x τ)

theorem **finRat_inf** (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : (finRat S T γ).map (infPart D) = 0

theorem **finRat_fin** (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : (finRat S T γ).map (finPart D) = (ratW T S γ).map (finPart D)

theorem **piW_of_inf** (θ : InfFam D) {S T : HSp D} {u : WO S T} (hu : u.map (infPart D) = 0) : piW θ T S u = 0

theorem **piW_ratW** {θ : InfFam D} (hθ : IsInfFam θ) (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : piW θ T S (ratW T S γ) = infRat S T γ

theorem **ratW_eq** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : Chart θ σ) (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : ratW T S γ = iotaW σ T S (infRat S T γ) + finRat S T γ

theorem **latO_inf** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : HeightOneSpectrum (𝒪 D.host.F) → Prop) {u : WO S T} (hu : u ∈ latO S T LS LT P) : u.map (infPart D) = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OmVacuumEqSplitFacSmulSum

theorem **idem_smul_mem** {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {e : AE D} (he : e * e = e) : ∀ u ∈ latO S T LS LT fun _ => True, e • u ∈ latO S T LS LT fun _ => True

theorem **off_smul_mt** {e : AE D} {V : HSp D} {g : GA D V} (hg : SuppAt e V g) : (1 − (e + cA D e)) • mt V g = (1 − (e + cA D e)) • (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D))

theorem **off_smul_mt_inv** {e : AE D} {V : HSp D} {g : GA D V} (hg : SuppAt e V g) : (1 − (e + cA D e)) • mt V g⁻¹ = (1 − (e + cA D e)) • (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D))

theorem **off_smul_actO** {e : AE D} {g : GA D S} {h : GA D T} (hg : SuppAt e S g) (hh : SuppAt e T h) (u : WO S T) : (1 − (e + cA D e)) • actO S T g h u = (1 − (e + cA D e)) • u

theorem **compAt_mt** {e : AE D} (he : IsSplitIdem D e) {V : HSp D} {g : GA D V} (hg : SuppAt e V g) (θ : InfFam D) (τ : D.what₂.Tau) : compAt θ τ (mt V g) = 1

theorem **pairNorm_smul** {m n : ℕ} (μ : Measure (Mx D m n)) (ℱ : Set (Mx D m n)) (c : ℝ) (f : Mx D m n → ℂ) : nrm2 μ ℱ ((c : ℂ) • f) = c ^ 2 * nrm2 μ ℱ f

theorem **reps_eq** {Λ Λ' : AddSubgroup (WO S T)} {e : AE D} {R : Finset (WO S T)} (hR : ∀ u ∈ R, u ∈ Λ' ∧ IsHalf e u) (hRu : ∀ u ∈ Λ', IsHalf e u → ∃! v, v ∈ R ∧ u − v ∈ Λ) {u v : WO S T} (hu : u ∈ R) (hv : v ∈ R) (h : u − v ∈ Λ) : u = v

theorem **sum_sub_notMem** {Λ Λ' : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) (hst : ∀ u ∈ Λ, e • u ∈ Λ) (hstc : ∀ u ∈ Λ, cA D e • u ∈ Λ) {R R' : Finset (WO S T)} (hR : ∀ u ∈ R, u ∈ Λ' ∧ IsHalf e u) (hRu : ∀ u ∈ Λ', IsHalf e u → ∃! v, v ∈ R ∧ u − v ∈ Λ) (hR' : ∀ u ∈ R', u ∈ Λ' ∧ IsHalf (cA D e) u) (hRu' : ∀ u ∈ Λ', IsHalf (cA D e) u → ∃! v, v ∈ R' ∧ u − v ∈ Λ) {u v u' v' : WO S T} (hu : u ∈ R) (hv : v ∈ R) (hu' : u' ∈ R') (hv' : v' ∈ R') (hne : (u, u') ≠ (v, v')) : u + u' − (v + v') ∉ Λ

theorem **form_sum** {Λ Λ' : AddSubgroup (WO S T)} {e : AE D} (he : IsSplitIdem D e) (hd : ∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ Λ ↔ ∀ u' ∈ Λ, ψ (symplO S T u u') = 1)) (hdd : ∀ u ∈ Λ', ∀ u' ∈ Λ', ψ (symplO S T u u') = 1) (hstc : ∀ u ∈ Λ, cA D e • u ∈ Λ) {R R' : Finset (WO S T)} (hR : ∀ u ∈ R, u ∈ Λ' ∧ IsHalf e u) (hRu : ∀ u ∈ Λ', IsHalf e u → ∃! v, v ∈ R ∧ u − v ∈ Λ) (hR' : ∀ u ∈ R', u ∈ Λ' ∧ IsHalf (cA D e) u) {f : WO S T → ℂ} (hf : IsLatVac (heisO ψ S T) Λ f) (hth : ∀ u : WO S T, IsHalf e u → IsThetaFunO ψ S T (heisO ψ S T u f)) {ℓ : (WO S T → ℂ) →ₗ[ℂ] ℂ} (hℓ : ∀ y, IsHalf (cA D e) y → ∀ φ, IsThetaFunO ψ S T φ → ℓ (heisO ψ S T y φ) = ℓ φ) : ℓ (∑ u ∈ R, ∑ u' ∈ R', heisO ψ S T u (heisO ψ S T u' f)) = R'.card * ℓ f

theorem **om_vacuum_eq_splitFac_smul_sum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → S.SigOK → T.SigOK → ∀ (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ (ΛS ΛS' : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT ΛT' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC S ΛS' → IsSelfDualC T ΛT → IsSelfDualC T ΛT' → ∀ (e : AE D), IsSplitIdem D e → ∀ (g : GA D S) (h : GA D T), SuppAt e S g → SuppAt e T h → ∀ x y, IsDetAt e S g x → IsDetAt e T h y → (∀ u : WO S T, u ∈ (latO S T (ΛS' : Set (Fin S.rank → D.host.E)) (ΛT' : Set (Fin T.rank → D.host.E)) fun _ => Tru

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OneAddSmulMemOfSqZero

theorem **exists_poly_ev_line** {ι : Type} [Fintype ι] [DecidableEq ι] (f : MvPolynomial (ι × ι) ℚ) (A B : Matrix ι ι ℂ) : ∃ p : Polynomial ℂ, ∀ t : ℂ, p.eval t = ev f (A + t • B)

theorem **one_add_smul_mem_of_sq_zero** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsQGroup Γ → ∀ ε : Matrix ι ι ℂ, ε * ε = 0 → 1 + ε ∈ Γ → ∀ t : ℂ, 1 + t • ε ∈ Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OneAddSmulMemOfSqZeroStatement

theorem **_root_.GlobalTheory.T.one_add_smul_mem_of_sq_zero'** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsQGroup Γ → ∀ ε : Matrix ι ι ℂ, ε * ε = 0 → 1 + ε ∈ Γ → ∀ t : ℂ, 1 + t • ε ∈ Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OneEqSumPlaceIdempotentsOverTwoAddTwoMul

theorem **evalV_singleV_same** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : evalV F v (singleV F v 1) = 1

theorem **evalV_singleV_ne** (F : Type) [Field F] [NumberField F] (v v' : HeightOneSpectrum (𝒪 F)) (h : v' ≠ v) : evalV F v' (singleV F v 1) = 0

theorem **half_mem_adicCompletionIntegers** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (h : (2 : 𝒪 F) ∉ v.asIdeal) : algebraMap F (Fv F v) (2 : F)⁻¹ ∈ v.adicCompletionIntegers F

theorem **one_eq_sum_singleV_add_two_mul** : ∀ (F : Type) [Field F] [NumberField F], ∃ P : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∈ P, (2 : 𝒪 F) ∈ v.asIdeal) ∧ (∀ v : HeightOneSpectrum (𝒪 F), singleV F v 1 ∈ intAf F ∧ evalV F v (singleV F v 1) = 1 ∧ ∀ v', v' ≠ v → evalV F v' (singleV F v 1) = 0) ∧ ∃ β ∈ intAf F, (1 : Af F) = ∑ v ∈ P, singleV F v 1 + 2 * β

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OneEqSumPlaceIdempotentsOverTwoAddTwoMulStatement

theorem **_root_.GlobalTheory.T.one_eq_sum_placeIdempotents_over_two_add_two_mul** : ∀ (F : Type) [Field F] [NumberField F], ∃ P : Finset (HeightOneSpectrum (𝒪 F)), (∀ v ∈ P, (2 : 𝒪 F) ∈ v.asIdeal) ∧ (∀ v : HeightOneSpectrum (𝒪 F), singleV F v 1 ∈ intAf F ∧ evalV F v (singleV F v 1) = 1 ∧ ∀ v', v' ≠ v → evalV F v' (singleV F v 1) = 0) ∧ ∃ β ∈ intAf F, (1 : Af F) = ∑ v ∈ P, singleV F v 1 + 2 * β

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.OneSubRePosOfNormLeHalf

theorem **eulerTerm_norm_le_rpow** (n : ℕ) (hn : 2 ≤ n) (c s : ℂ) (hc : ‖c‖ ≤ 1) : ‖c * (n : ℂ) ^ (-s)‖ ≤ (n : ℝ) ^ (-s.re)

theorem **eulerTerm_norm_le_half** (n : ℕ) (hn : 2 ≤ n) (c s : ℂ) (hc : ‖c‖ ≤ 1) (hs : 1 < s.re) : ‖c * (n : ℂ) ^ (-s)‖ ≤ 1 / 2

theorem **one_sub_re_pos_of_norm_le_half** (z : ℂ) (hz : ‖z‖ ≤ 1 / 2) : 0 < (1 − z).re

theorem **eulerProduct_eq_exp_and_pos** : ∀ (ι : Type) (N : ι → ℕ) (c : ι → ℂ), (∀ i, 2 ≤ N i) → (∀ i, ‖c i‖ ≤ 1) → (∀ σ : ℝ, 1 < σ → Summable fun i => (N i : ℝ) ^ (-σ)) → (∀ s : ℂ, 1 < s.re → Summable (fun i => -Complex.log (1 − c i * ((N i : ℕ) : ℂ) ^ (-s))) ∧ HasProd (fun i => (1 − c i * ((N i : ℕ) : ℂ) ^ (-s))⁻¹) (Complex.exp (∑' i, -Complex.log (1 − c i * ((N i : ℕ) : ℂ) ^ (-s))))) ∧ ∀ e : ι ≃ ι, (∀ i, N (e i) = N i) → (∀ i, c (e i) = starRingEnd ℂ (c i)) → ∀ s : ℂ, 1 < s.re → s.im = 0 → (Complex.exp (∑' i, -Complex.log (1 − c i * ((N i : ℕ) : ℂ) ^ (-s)))).im = 0 ∧ 0 < (Complex.exp (∑' i, -Complex.log (1 − c i * ((N i : ℕ) : ℂ) ^ (-s)))).re

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PIdxUnitsFiniteExtensionAdicCompletion

theorem **pIdx_range_of_injective** (p : ℕ) {A B : Type} [CommGroup A] [CommGroup B] (j : A →* B) (hj : Function.Injective j) : pIdx p j.range = pIdx p (⊤ : Subgroup A)

theorem **pTor_range_of_injective** (p : ℕ) {A B : Type} [CommGroup A] [CommGroup B] (j : A →* B) (hj : Function.Injective j) : pTor p j.range = pTor p (⊤ : Subgroup A)

theorem **pTor_pIdx_subgroup_int** (p : ℕ) (hp : 0 < p) (H : Subgroup (Multiplicative ℤ)) (hH : H ≠ ⊥) : pTor p H = 1 ∧ pIdx p H = p

theorem **pIdx_of_ker_eq_range** (p : ℕ) (hp : 0 < p) {A B : Type} [CommGroup A] [CommGroup B] (j : A →* B) (hj : Function.Injective j) (g : B →* Multiplicative ℤ) (hk : g.ker = j.range) (hg : g.range ≠ ⊥) (N : ℕ) (hN : N ≠ 0) (h1 : pTor p (⊤ : Subgroup A) ≠ 0) (h2 : pIdx p (⊤ : Subgroup A) = N * pTor p (⊤ : Subgroup A)) : pTor p (⊤ : Subgroup B) ≠ 0 ∧ pIdx p (⊤ : Subgroup B) = p * N * pTor p (⊤ : Subgroup B)

theorem **pIdx_units_finite_extension_adicCompletion** (p : ℕ) (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : ℕ) (L : Type) [Field L] [Algebra (v.adicCompletion F) L] [FiniteDimensional (v.adicCompletion F) L] (hp : 0 < p) (he : v.valuation F (p : F) = WithZero.exp (-(e : ℤ))) : pTor p (⊤ : Subgroup Lˣ) ≠ 0 ∧ pIdx p (⊤ : Subgroup Lˣ) = p * (Ideal.absNorm v.asIdeal ^ e) ^ Module.finrank (v.adicCompletion F) L * pTor p (⊤ : Subgroup Lˣ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PIdxUnitsFiniteExtensionAdicCompletionStatement

theorem **_root_.GlobalTheory.T.pIdx_units_finite_extension_adicCompletion'** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (e : ℕ) (L : Type) [Field L] [Algebra (v.adicCompletion F) L] [FiniteDimensional (v.adicCompletion F) L], 0 < p → v.valuation F (p : F) = WithZero.exp (-(e : ℤ)) → pTor p (⊤ : Subgroup Lˣ) ≠ 0 ∧ pIdx p (⊤ : Subgroup Lˣ) = p * (Ideal.absNorm v.asIdeal ^ e) ^ Module.finrank (v.adicCompletion F) L * pTor p (⊤ : Subgroup Lˣ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowCardLeIndexIInfOfPrime

theorem **pow_card_le_index_iInf_of_prime** : ∀ (p : ℕ) (G ι : Type) [CommGroup G] [Fintype ι] (K : ι → Subgroup G), p.Prime → (∀ (i : ι) (x : G), x ^ p ∈ K i) → (∀ i, ∃ x, x ∉ K i ∧ ∀ j, j ≠ i → x ∈ K j) → (⊓ i, K i).index ≠ 0 → p ^ Fintype.card ι ≤ (⊓ i, K i).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowCardLeIndexIInfOfPrimeStatement

theorem **_root_.GlobalTheory.T.pow_card_le_index_iInf_of_prime'** : ∀ (p : ℕ) (G ι : Type) [CommGroup G] [Fintype ι] (K : ι → Subgroup G), p.Prime → (∀ (i : ι) (x : G), x ^ p ∈ K i) → (∀ i, ∃ x, x ∉ K i ∧ ∀ j, j ≠ i → x ∈ K j) → (⊓ i, K i).index ≠ 0 → p ^ Fintype.card ι ≤ (⊓ i, K i).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowCountPlacesNotLocalPowerLeIndex

theorem **mem_kNorm_iff_map_mem_id** (p : ℕ) {F L : Type} [Field F] [Field L] (φ : F →+* L) (δ : F) (x : Fˣ) : x ∈ kNorm p φ δ ↔ Units.map (φ : F →* L) x ∈ kNorm p (RingHom.id L) (φ δ)

theorem **pow_tNon_le_index_finNorm** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ δ : F) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → (finNorm p F δ S).index ≠ 0 ∧ p ^ tNon p F δ S ≤ (finNorm p F δ S).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowCountPlacesNotLocalPowerLeIndexStatement

theorem **_root_.GlobalTheory.T.pow_countPlacesNotLocalPower_le_index** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ δ : F) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → (finNorm p F δ S).index ≠ 0 ∧ p ^ tNon p F δ S ≤ (finNorm p F δ S).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowIndexTorsionFiniteKerRange

theorem **pIdx_pTor_ker_range_ne_zero** : ∀ (p : ℕ) (A A' : Type) [CommGroup A] [CommGroup A'] (g : A →* A'), pIdx p (⊤ : Subgroup A) ≠ 0 → pTor p (⊤ : Subgroup A) ≠ 0 → pTor p g.range ≠ 0 → pIdx p g.ker ≠ 0 ∧ pTor p g.ker ≠ 0 ∧ pIdx p g.range ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowIndexTorsionFiniteKerRangeStatement

theorem **_root_.GlobalTheory.T.powIndex_torsion_finite_ker_range** : ∀ (p : ℕ) (A A' : Type) [CommGroup A] [CommGroup A'] (g : A →* A'), pIdx p (⊤ : Subgroup A) ≠ 0 → pTor p (⊤ : Subgroup A) ≠ 0 → pTor p g.range ≠ 0 → pIdx p g.ker ≠ 0 ∧ pTor p g.ker ≠ 0 ∧ pIdx p g.range ≠ 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowIndexTorsionKerRangeMul

theorem **pTor_eq_card_inf_ker** (p : ℕ) {G : Type} [CommGroup G] (U : Subgroup G) : pTor p U = Nat.card (U ⊓ (powMonoidHom p : G →* G).ker : Subgroup G)

theorem **pIdx_pTor_ker_range** : ∀ (p : ℕ) (A A' : Type) [CommGroup A] [CommGroup A'] (g : A →* A'), pIdx p g.ker ≠ 0 → pTor p g.ker ≠ 0 → pIdx p g.range ≠ 0 → pTor p g.range ≠ 0 → pIdx p (⊤ : Subgroup A) ≠ 0 ∧ pTor p (⊤ : Subgroup A) ≠ 0 ∧ pIdx p (⊤ : Subgroup A) * pTor p g.ker * pTor p g.range = pIdx p g.ker * pIdx p g.range * pTor p (⊤ : Subgroup A)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowIndexTorsionKerRangeMulStatement

theorem **_root_.GlobalTheory.T.powIndex_torsion_ker_range_mul** : ∀ (p : ℕ) (A A' : Type) [CommGroup A] [CommGroup A'] (g : A →* A'), pIdx p g.ker ≠ 0 → pTor p g.ker ≠ 0 → pIdx p g.range ≠ 0 → pTor p g.range ≠ 0 → pIdx p (⊤ : Subgroup A) ≠ 0 ∧ pTor p (⊤ : Subgroup A) ≠ 0 ∧ pIdx p (⊤ : Subgroup A) * pTor p g.ker * pTor p g.range = pIdx p g.ker * pIdx p g.range * pTor p (⊤ : Subgroup A)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowOfSUnitLocalPowerAtPlaces

theorem **pow_of_mem_sUnits_union_of_mem_finPow** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] [DecidableEq (HeightOneSpectrum (𝒪 F))] (ζ : F) (S T : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → sUnits F (S ∪ T) ⊔ finPow p F T = ⊤ → ∀ y ∈ sUnits F (S ∪ T), y ∈ finPow p F S → ∃ r : F, r ^ p = (y : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowOfSUnitLocalPowerAtPlacesStatement

theorem **_root_.GlobalTheory.T.pow_of_sUnit_localPower_at_places** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] [DecidableEq (HeightOneSpectrum (𝒪 F))] (ζ : F) (S T : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → sUnits F (S ∪ T) ⊔ finPow p F T = ⊤ → ∀ y ∈ sUnits F (S ∪ T), y ∈ finPow p F S → ∃ r : F, r ^ p = (y : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowPrimeRangeEqIteratedDifference

private lemma **pow_apply_ofMul_eq_iterate** {M : Type} [CommGroup M] (ψ : AddMonoid.End (Additive M)) (φ : M →* M) (hψ : ∀ a, ψ (Additive.ofMul a) = Additive.ofMul (φ a)) (n : ℕ) (a : M) : (ψ ^ n) (Additive.ofMul a) = Additive.ofMul ((⇑φ)^[n] a)

private lemma **cyclotomic_comp_X_add_one_eq** (p : ℕ) [Fact p.Prime] : ∃ H : ℤ[X], (cyclotomic p ℤ).comp (X + 1) = X ^ (p - 1) + (p : ℤ[X]) * (1 + X * H)

private lemma **exists_pow_factor_of_aeval_cyclotomic** {S : Type*} [Ring S] (p : ℕ) [Fact p.Prime] (y : S) (hy : aeval y ((cyclotomic p ℤ).comp (X + 1)) = 0) : ∃ W V : S, y ^ (p - 1) = (p : S) * W ∧ y ^ (p - 1) = W * p ∧ (p : S) = y ^ (p - 1) * V ∧ (p : S) = V * y ^ (p - 1)

theorem **_root_.GlobalTheory.CNorm.pows_eq_range_dif_pow** : ∀ (p : ℕ) (M : Type) [CommGroup M] (τ : M →* M) (lam : Monoid.End M), p.Prime → (∀ a, lam a = τ a / a) → (∀ a, _root_.GlobalTheory.CNorm.nrm τ p a = 1) → _root_.GlobalTheory.CNorm.pows p (⊤ : Subgroup M) = MonoidHom.range (lam ^ (p - 1) : Monoid.End M) ∧ ∀ a : M, a ^ p = 1 ↔ (lam ^ (p - 1) : Monoid.End M) a = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowPrimeRangeEqIteratedDifferenceStatement

theorem **_root_.GlobalTheory.T.pow_prime_range_eq_iterated_difference** : ∀ (p : ℕ) (M : Type) [CommGroup M] (τ : M →* M) (lam : Monoid.End M), p.Prime → (∀ a, lam a = τ a / a) → (∀ a, nrm τ p a = 1) → pows p (⊤ : Subgroup M) = MonoidHom.range (lam ^ (p - 1) : Monoid.End M) ∧ ∀ a : M, a ^ p = 1 ↔ (lam ^ (p - 1) : Monoid.End M) a = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PowsSubgroupOfSelfSPow

theorem **pows_subgroupOf_self_sPow** (p : ℕ) {G : Type} [CommGroup G] (U : Subgroup G) : (pows p U).subgroupOf U = pows p (⊤ : Subgroup U)

theorem **pTor_sUnits** (p : ℕ) (F : Type) [Field F] [NumberField F] (ζ : F) (S : Finset (HeightOneSpectrum (𝒪 F))) (hp : p.Prime) (hζ : IsPrimitiveRoot ζ p) : pTor p (sUnits F S) = p

theorem **_root_.GlobalTheory.CNorm.pTor_pIdx_sUnits** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] (ζ : F) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → _root_.GlobalTheory.CNorm.pTor p (_root_.GlobalTheory.QNorm.sUnits F S) = p ∧ _root_.GlobalTheory.CNorm.pIdx p (_root_.GlobalTheory.QNorm.sUnits F S) = p ^ (S.card + Fintype.card (InfinitePlace F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PrimeIdempotentEqIndicatorPrime

theorem **_root_.GlobalTheory.isPrimeIdem_eq_indicator** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), ∀ Φ : _root_.HeckeCharacterZeroLocus.ThmBIn.AE F E ≃+* (mixedSpace E × FiniteAdeleRing (𝒪 E) E), (∀ x : E, Φ (_root_.HeckeCharacterZeroLocus.ThmBIn.incA F E x) = (mixedEmbedding E x, algebraMap E _ x)) → (∀ y, _root_.HeckeCharacterZeroLocus.ThmBIn.finPart F E y ∈ _root_.HeckeCharacterZeroLocus.ThmBIn.intAE F E ↔ ∀ P, Valued.v ((Φ y).2 P) ≤ 1) → (∀ v ε, _root_.HeckeCharacterZeroLocus.ThmBIn.IsPlaceIdem F v ε → (Φ (algebraMap (_root_.HeckeCharacterZeroLocus.ThmBIn.AF F) (_root_.HeckeCharacterZeroLocus.ThmBIn.AE F E) ε)).1 = 0 ∧ ∀ P : HeightOneSpectrum (𝒪 E), (Φ (algebraMap (_root_.HeckeCharacterZeroLocus.ThmBIn.AF F) (_root_.HeckeCharacterZeroLocus.ThmBIn.AE F E) ε)).2 P = if P.under (𝒪 F) = v then 1 else 0) → ∀ P e f, _root_.HeckeCharacte

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PrimeIdempotentEqIndicatorPrimeStatement

theorem **_root_.GlobalTheory.T.primeIdempotent_eq_indicator_prime** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), ∀ Φ : AE F E ≃+* (mixedSpace E × FiniteAdeleRing (𝒪 E) E), (∀ x : E, Φ (incA F E x) = (mixedEmbedding E x, algebraMap E _ x)) → (∀ y, finPart F E y ∈ intAE F E ↔ ∀ P, Valued.v ((Φ y).2 P) ≤ 1) → (∀ v ε, IsPlaceIdem F v ε → (Φ (algebraMap (AF F) (AE F E) ε)).1 = 0 ∧ ∀ P : HeightOneSpectrum (𝒪 E), (Φ (algebraMap (AF F) (AE F E) ε)).2 P = if P.under (𝒪 F) = v then 1 else 0) → ∀ P e f, IsPrimeIdem σ P e f → ((Φ e).1 = 0 ∧ (Φ e).2 P = 1 ∧ ∀ Q, Q ≠ P → (Φ e).2 Q = 0)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PrimeMulCardPlacesLeCardOverKummer

theorem **prime_mul_card_le_card_over** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → p * S.card ≤ SE.card + (p - 1) * tNon p F δ S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PrimeMulCardPlacesLeCardOverKummerStatement

theorem **_root_.GlobalTheory.T.prime_mul_card_places_le_card_over_kummer** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → p * S.card ≤ SE.card + (p - 1) * tNon p F δ S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ProdAbsNormPowValuationTwoLe

theorem **prod_absNorm_pow_le_two_pow** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (e : HeightOneSpectrum (𝒪 F) → ℕ), (∀ v ∈ S, v.valuation F 2 = WithZero.exp (-(e v : ℤ))) → ∏ v ∈ S, Ideal.absNorm v.asIdeal ^ e v ≤ 2 ^ Module.finrank ℚ F

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ProdAbsNormPowValuationTwoLeStatement

theorem **_root_.GlobalTheory.T.prod_absNorm_pow_valuation_two_le** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (e : HeightOneSpectrum (𝒪 F) → ℕ), (∀ v ∈ S, v.valuation F 2 = WithZero.exp (-(e v : ℤ))) → ∏ v ∈ S, Ideal.absNorm v.asIdeal ^ e v ≤ 2 ^ Module.finrank ℚ F

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ProdLeftEqMaps

theorem **prodLeft_eq_maps** {F A B E : Type} [Field F] [CommRing A] [CommRing B] [CommRing E] [Algebra F A] [Algebra F B] [Algebra F E] (z : TensorProduct F (A × B) E) : TensorProduct.prodLeft F F A B E z = ((Algebra.TensorProduct.map (AlgHom.fst F A B) (AlgHom.id F E) : TensorProduct F (A × B) E → TensorProduct F A E) z, (Algebra.TensorProduct.map (AlgHom.snd F A B) (AlgHom.id F E) : TensorProduct F (A × B) E → TensorProduct F B E) z)

theorem **main** : ∀ {F A B E : Type} [inst : Field F] [inst_1 : CommRing A] [inst_2 : CommRing B] [inst_3 : CommRing E] [inst_4 : Algebra F A] [inst_5 : Algebra F B] [inst_6 : Algebra F E] (z : TensorProduct F (A × B) E), (Algebra.TensorProduct.map (AlgHom.fst F A B) (AlgHom.id F E) : TensorProduct F (A × B) E → TensorProduct F A E) z = (0 : TensorProduct F A E) → (Algebra.TensorProduct.map (AlgHom.snd F A B) (AlgHom.id F E) : TensorProduct F (A × B) E → TensorProduct F B E) z = (0 : TensorProduct F B E) → z = (0 : TensorProduct F (A × B) E)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.PureTensorLIntertwine

theorem **pureTensorL_intertwine** {ι X : Type} (p : (X → ℂ) → Prop) (p0 : p 0) (padd : ∀ a b, p a → p b → p (a + b)) (psmul : ∀ (c : ℂ) a, p a → p (c • a)) (A A' : ι → (X → ℂ) → (X → ℂ)) (Φ : Module.End ℂ (X → ℂ)) (hA : ∀ i f, p f → p (A i f) ∧ Φ (A i f) = A' i (Φ f)) (f : X → ℂ) (hf : p f) (P : MvPolynomial ι ℂ) : p (pureTensorL A f P) ∧ Φ (pureTensorL A f P) = pureTensorL A' (Φ f) P

theorem **cplxO_move** {m n : ℕ} {sS sS' G G' : Matrix (Fin n) (Fin n) ℂ} {sT sT' H : Matrix (Fin m) (Fin m) ℂ} (hG : G * G' = 1) (hS : sS' * G = G * sS) (hT : sT' * H = H * sT) (z : Matrix (Fin m) (Fin n) ℂ) : cplxO sS' sT' (H * z * G') = H * cplxO sS sT z * G'

theorem **unitMap_move** {m n : ℕ} {B G G' : Matrix (Fin n) (Fin n) ℂ} {B' H : Matrix (Fin m) (Fin m) ℂ} (hG : G * G' = 1) (i : Fin m) (j : Fin n) : unitMap (G * B) (H * B') i j = H * unitMap B B' i j * G'

theorem **om_pureTensorO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → ∀ (sS sS' BS BS' : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT sT' BT BT' : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (g : GA D S) (h : GA D T), (∀ τ, sS' τ * compAt θ τ (mt S g) = compAt θ τ (mt S g) * sS τ) → (∀ τ, sT' τ * compAt θ τ (mt T h) = compAt θ τ (mt T h) * sT τ) → (∀ τ, BS' τ = compAt θ τ (mt S g) * BS τ) → (∀ τ, BT' τ = compAt θ τ (mt T h) * BT τ) → ∀ f, IsThetaFunO ψ S T f → ∀ P, IsThetaFunO ψ S T (pureTensorO ψ θ S T sS sT BS BT f P) ∧ om (g, h) (pureTensorO ψ θ S T sS sT BS BT f P) = pureTensorO ψ θ S T sS' sT' BS' BT' (om (g, h) f) P

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.QuasiReflectionEqMulTransvectionsOfHyperbolicPla

theorem **hf_add_left** (x y v : Fin n → R) : hf σ G (x + y) v = hf σ G x v + hf σ G y v

theorem **hf_add_right** (x y v : Fin n → R) : hf σ G v (x + y) = hf σ G v x + hf σ G v y

theorem **hf_smul_left** (a : R) (x v : Fin n → R) : hf σ G (a • x) v = a * hf σ G x v

theorem **hf_smul_right** (a : R) (x v : Fin n → R) : hf σ G v (a • x) = σ a * hf σ G v x

theorem **tv_zero** (x : Fin n → R) : tv σ G x 0 = 1

private lemma **transvection_zero'** {E : Type} [Field E] {n : ℕ} (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (w : Fin n → E) : _root_.GlobalTheory.UTrans.tv σ G w 0 = 1

private lemma **quasiReflection_eq_transvectionGen_mul_shifted** {E : Type} [Field E] [CharZero E] {n : ℕ} (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (hinv : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) (x z : Fin n → E) (a u : E) (hzz : _root_.GlobalTheory.UTrans.hf σ G z z = 0) (hxz : _root_.GlobalTheory.UTrans.hf σ G x z ≠ 0) (hxx : _root_.GlobalTheory.UTrans.hf σ G x x ≠ 0) (hyy : _root_.GlobalTheory.UTrans.hf σ G (x + a • z) (x + a • z) ≠ 0) (hu : u * σ u = 1) (hu1 : u ≠ 1) (hD : u * σ a * _root_.GlobalTheory.UTrans.hf σ G x z + a * σ (_root_.GlobalTheory.UTrans.hf σ G x z) ≠ 0) : ∃ m ∈ _root_.GlobalTheory.UTrans.TvGen σ G, _root_.GlobalTheory.UTrans.tv σ G x ((u - 1) / _root_.GlobalTheory.UTrans.hf σ G x x) = m * _root_.GlobalTheory.UTrans.tv σ G (x + a • z) ((u - 1) / _root_.GlobalTheory.UTrans.hf σ G (x + a • z) (x + a • z))

private lemma **transvection_isometry** {E : Type} [Field E] {n : ℕ} (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (hinv : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) (w : Fin n → E) (t : E) (ht : t + σ t + t * σ t * _root_.GlobalTheory.UTrans.hf σ G w w = 0) (v v' : Fin n → E) : _root_.GlobalTheory.UTrans.hf σ G ((_root_.GlobalTheory.UTrans.tv σ G w t).mulVec v) ((_root_.GlobalTheory.UTrans.tv σ G w t).mulVec v') = _root_.GlobalTheory.UTrans.hf σ G v v'

private lemma **isometry_mul_transvection** {E : Type} [Field E] {n : ℕ} (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (g : Matrix (Fin n) (Fin n) E) (hg : ∀ v w, _root_.GlobalTheory.UTrans.hf σ G (g.mulVec v) (g.mulVec w) = _root_.GlobalTheory.UTrans.hf σ G v w) (w : Fin n → E) (t : E) : g * _root_.GlobalTheory.UTrans.tv σ G w t = _root_.GlobalTheory.UTrans.tv σ G (g.mulVec w) t * g

private lemma **transvection_mul_isotropic** {E : Type} [Field E] {n : ℕ} (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (w : Fin n → E) (hw : _root_.GlobalTheory.UTrans.hf σ G w w = 0) (s t : E) : _root_.GlobalTheory.UTrans.tv σ G w s * _root_.GlobalTheory.UTrans.tv σ G w t = _root_.GlobalTheory.UTrans.tv σ G w (s + t)

theorem **_root_.GlobalTheory.UTrans.PlaneRefl.quasiRefl_eq_mul_of_plane** : ∀ (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), _root_.GlobalTheory.UTrans.IsHerm σ G → ∀ (x z : Fin n → E) (a u : E), _root_.GlobalTheory.UTrans.hf σ G z z = 0 → _root_.GlobalTheory.UTrans.hf σ G x z ≠ 0 → _root_.GlobalTheory.UTrans.hf σ G x x ≠ 0 → _root_.GlobalTheory.UTrans.hf σ G (x + a • z) (x + a • z) ≠ 0 → u * σ u = 1 → ∃ m ∈ _root_.GlobalTheory.UTrans.TvGen σ G, _root_.GlobalTheory.UTrans.tv σ G x ((u - 1) / _root_.GlobalTheory.UTrans.hf σ G x x) = m * _root_.GlobalTheory.UTrans.tv σ G (x + a • z) ((u - 1) / _root_.GlobalTheory.UTrans.hf σ G (x + a • z) (x + a • z))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.QuasiReflectionEqMulTransvectionsOfHyperbolicPlaStatement

theorem **_root_.GlobalTheory.T.quasiReflection_eq_mul_transvections_of_hyperbolicPlane** : ∀ (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), IsHerm σ G → ∀ (x z : Fin n → E) (a u : E), hf σ G z z = 0 → hf σ G x z ≠ 0 → hf σ G x x ≠ 0 → hf σ G (x + a • z) (x + a • z) ≠ 0 → u * σ u = 1 → ∃ m ∈ TvGen σ G, tv σ G x ((u - 1) / hf σ G x x) = m * tv σ G (x + a • z) ((u - 1) / hf σ G (x + a • z) (x + a • z))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.QuasiReflectionEqUnitaryTransvectionMul

theorem **eventually_quad_ne** {E : Type} [Field E] [CharZero E] (a b c : E) (ha : a ≠ 0) : ∀ᶠ t : ℚ in Filter.cofinite, a + (t : E) * b + (t : E) ^ 2 * c ≠ 0

theorem **hf_line_left** (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (x w z : Fin n → E) (t : E) : hf σ G (x + t • w) z = hf σ G x z + t * hf σ G w z

theorem **hf_line_right** (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (x w z : Fin n → E) (t : E) : hf σ G z (x + t • w) = hf σ G z x + σ t * hf σ G z w

theorem **eventually_aniso** (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (x w : Fin n → E) (hx : hf σ G x x ≠ 0) : ∀ᶠ t : ℚ in Filter.cofinite, hf σ G (x + (t : E) • w) (x + (t : E) • w) ≠ 0

theorem **eventually_pair** (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (x w z : Fin n → E) (hx : hf σ G x z ≠ 0) : ∀ᶠ t : ℚ in Filter.cofinite, hf σ G (x + (t : E) • w) z ≠ 0

theorem **eventually_chain** (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E) (hG : IsHerm σ G) {ι : Type} (z : ι → Fin n → E) (l : ι → E) (x : Fin n → E) (u : E) (hx : hf σ G x x ≠ 0) (hu : u * σ u = 1) (hz : ∀ i, hf σ G (z i) (z i) = 0) (hxz : ∀ i, hf σ G x (z i) ≠ 0) (S : Finset ι) : ∀ᶠ t : ℚ in Filter.cofinite, ∃ m ∈ TvGen σ G, tv σ G x ((u - 1) / hf σ G x x) = m * tv σ G (x + (t : E) • ∑ i ∈ S, l i • z i) ((u - 1) / hf σ G (x + (t : E) • ∑ i ∈ S, l i • z i) (x + (t : E) • ∑ i ∈ S, l i • z i))

theorem **GlobalTheory.UTrans.quasiReflection_eq_tvGen_mul** : ∀ (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), IsHerm σ G → (∃ z : Fin n → E, z ≠ 0 ∧ hf σ G z z = 0) → ∀ (x y : Fin n → E) (u : E), hf σ G x x ≠ 0 → hf σ G y y ≠ 0 → u * σ u = 1 → ∃ m ∈ TvGen σ G, tv σ G x ((u - 1) / hf σ G x x) = m * tv σ G y ((u - 1) / hf σ G y y)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatComponentMemRatPolysVanishingOnMatrixSet

theorem **exists_component_mem_vanI_of_mem_span** {ι : Type} [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)) (l : ℂ →ₗ[ℚ] ℚ) (f : MvPolynomial (ι × ι) ℂ) (hf : f ∈ Submodule.span ℂ ((MvPolynomial.map (algebraMap ℚ ℂ)) '' vanI Γ)) : ∀ c : ℂ, ∃ f' ∈ vanI Γ, ∀ m, MvPolynomial.coeff m f' = l (c * MvPolynomial.coeff m f)

theorem **component_mem_vanI** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ g ∈ Γ, evC f g = 0) → ∀ (l : ℂ →ₗ[ℚ] ℚ) (f' : MvPolynomial (ι × ι) ℚ), (∀ m, MvPolynomial.coeff m f' = l (MvPolynomial.coeff m f)) → f' ∈ vanI Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatComponentMemRatPolysVanishingOnMatrixSetStatement

theorem **_root_.GlobalTheory.T.rat_component_mem_ratPolysVanishingOnMatrixSet** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ σ : ℂ ≃+* ℂ, ∀ g ∈ Γ, g.map σ ∈ Γ) → ∀ f : MvPolynomial (ι × ι) ℂ, (∀ g ∈ Γ, evC f g = 0) → ∀ (l : ℂ →ₗ[ℚ] ℚ) (f' : MvPolynomial (ι × ι) ℚ), (∀ m, MvPolynomial.coeff m f' = l (MvPolynomial.coeff m f)) → f' ∈ vanI Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatMemRatZariskiClosurePoints

lemma **_root_.GlobalTheory.ClassFin.RatPts.ev_map_algebraMap** {ι R : Type} [CommRing R] [Algebra ℚ R] (f : MvPolynomial (ι × ι) ℚ) (γ : Matrix ι ι ℚ) : _root_.GlobalTheory.ClassFin.ev f (γ.map (algebraMap ℚ R)) = algebraMap ℚ R (_root_.GlobalTheory.ClassFin.ev f γ)

theorem **_root_.GlobalTheory.ClassFin.RatPts.rat_mem_pts** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ γ : Matrix ι ι ℚ, γ.map (algebraMap ℚ ℂ) ∈ Γ → γ ∈ _root_.GlobalTheory.ClassFin.pts Γ ℚ) ∧ ∀ (R : Type) [CommRing R] [Algebra ℚ R], ∀ γ ∈ _root_.GlobalTheory.ClassFin.pts Γ ℚ, γ.map (algebraMap ℚ R) ∈ _root_.GlobalTheory.ClassFin.pts Γ R

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatMemRatZariskiClosurePointsStatement

theorem **_root_.GlobalTheory.T.rat_mem_ratZariskiClosurePoints** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), (∀ γ : Matrix ι ι ℚ, γ.map (algebraMap ℚ ℂ) ∈ Γ → γ ∈ pts Γ ℚ) ∧ ∀ (R : Type) [CommRing R] [Algebra ℚ R], ∀ γ ∈ pts Γ ℚ, γ.map (algebraMap ℚ R) ∈ pts Γ R

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatZariskiClosureOfAutStableSLGroup

theorem **subset_pts** (Γ : Set (Matrix ι ι ℂ)) : Γ ⊆ pts Γ ℂ

theorem **vanI_pts** (Γ : Set (Matrix ι ι ℂ)) : vanI (pts Γ ℂ) = vanI Γ

theorem **ev_map_aut** (σ : ℂ ≃+* ℂ) (f : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι ℂ) : ev f (g.map σ) = σ (ev f g)

theorem **exists_poly_of_dual** (φ : Module.Dual ℂ (Matrix ι ι ℂ)) : ∃ f : MvPolynomial (ι × ι) ℂ, ∀ g, evC f g = φ g

theorem **ratClosure** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → IsQGroup (pts Γ ℂ) ∧ vanI (pts Γ ℂ) = vanI Γ ∧ Submodule.span ℂ (pts Γ ℂ) = Submodule.span ℂ Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatZariskiClosureOfAutStableSLGroupStatement

theorem **_root_.GlobalTheory.T.ratZariskiClosure_of_autStable_SL_group** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → IsQGroup (pts Γ ℂ) ∧ vanI (pts Γ ℂ) = vanI Γ ∧ Submodule.span ℂ (pts Γ ℂ) = Submodule.span ℂ Γ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatZariskiClosurePointsIsGroup

theorem **ptsGroup_map_ev** {A B : Type} [CommRing A] [Algebra ℚ A] [CommRing B] [Algebra ℚ B] (φ : A →ₐ[ℚ] B) (f : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι A) : φ (ev f g) = ev f (g.map φ)

theorem **ptsGroup_generic_map** {R : Type} [CommRing R] [Algebra ℚ R] (g : Matrix ι ι R) : (Matrix.mvPolynomialX ι ι ℚ).map (MvPolynomial.aeval (R

theorem **ptsGroup_ev_adjugate** {R : Type} [CommRing R] [Algebra ℚ R] (f : MvPolynomial (ι × ι) ℚ) (g : Matrix ι ι R) : ev (ev f (Matrix.mvPolynomialX ι ι ℚ).adjugate) g = ev f g.adjugate

theorem **ptsGroup_ev_det** {R : Type} [CommRing R] [Algebra ℚ R] (g : Matrix ι ι R) : ev ((Matrix.mvPolynomialX ι ι ℚ).det - 1) g = g.det - 1

theorem **ptsGroup_ev_one** (R : Type) [CommRing R] [Algebra ℚ R] (f : MvPolynomial (ι × ι) ℚ) : ev f (1 : Matrix ι ι R) = algebraMap ℚ R (ev f (1 : Matrix ι ι ℚ))

theorem **ptsGroup_ev_mul** {R : Type} [CommRing R] [Algebra ℚ R] (f : MvPolynomial (ι × ι) ℚ) (x y : Matrix ι ι R) : MvPolynomial.aeval (Sum.elim (fun ij : ι × ι => x ij.1 ij.2) (fun ij : ι × ι => y ij.1 ij.2)) (ev f ((Matrix.of fun i j => (MvPolynomial.X (Sum.inl (i, j)) : MvPolynomial ((ι × ι) ⊕ (ι × ι)) ℚ)) * (Matrix.of fun i j => (MvPolynomial.X (Sum.inr (i, j)) : MvPolynomial ((ι × ι) ⊕ (ι × ι)) ℚ)))) = ev f (x * y)

theorem **pts_group** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ∀ (R : Type) [CommRing R] [Algebra ℚ R], (1 : Matrix ι ι R) ∈ pts Γ R ∧ (∀ x ∈ pts Γ R, ∀ y ∈ pts Γ R, x * y ∈ pts Γ R) ∧ (∀ x ∈ pts Γ R, x.adjugate ∈ pts Γ R) ∧ ∀ x ∈ pts Γ R, x.det = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RatZariskiClosurePointsIsGroupStatement

theorem **_root_.GlobalTheory.T.ratZariskiClosurePoints_isGroup** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (Γ : Set (Matrix ι ι ℂ)), IsPre Γ → ∀ (R : Type) [CommRing R] [Algebra ℚ R], (1 : Matrix ι ι R) ∈ pts Γ R ∧ (∀ x ∈ pts Γ R, ∀ y ∈ pts Γ R, x * y ∈ pts Γ R) ∧ (∀ x ∈ pts Γ R, x.adjugate ∈ pts Γ R) ∧ ∀ x ∈ pts Γ R, x.det = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RationalHermitianRankOneLatticeSelfDual

theorem **mem_mul_of_trace** (D : D7) (z : D.host.E) (h : ∀ a : 𝒪 D.host.E, ∃ n : ℤ, Algebra.trace ℚ D.host.E (D.host.xi0 * (a * z)) = n) : z ∈ D.host.AA * D.host.AAbar

theorem **mem_AA_of_trace** (D : D7) (z : D.host.E) (h : ∀ a ∈ D.host.AA, ∃ n : ℤ, Algebra.trace ℚ D.host.E (D.host.xi0 * (star a * z)) = n) : z ∈ D.host.AA

theorem **form_smul** (T : HSp D) (e : D.host.E) (t y : Fin T.rank → D.host.E) : ∑ c, ∑ d, star (t c) * T.gram c d * (e • y) d = e * ∑ c, ∑ d, star (t c) * T.gram c d * y d

theorem **genMap_mulVec** (S T : HSp D) (a : D.host.E) (s' s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) : genMap S T a s' t *ᵥ s = (a * ∑ c, ∑ d, star (s' c) * S.gram c d * s d) • t

theorem **hRat_add_left** (S T : HSp D) (x x' y : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hRat S T (x + x') y = hRat S T x y + hRat S T x' y

theorem **hRat_add_right** (S T : HSp D) (x y y' : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hRat S T x (y + y') = hRat S T x y + hRat S T x y'

theorem **hRat_neg_left** (S T : HSp D) (x y : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hRat S T (-x) y = -hRat S T x y

theorem **hRat_neg_right** (S T : HSp D) (x y : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hRat S T x (-y) = -hRat S T x y

theorem **hRat_mem** (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) {ρ ℓ : Matrix (Fin T.rank) (Fin S.rank) D.host.E} (hρ : ρ ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E))) (hℓ : ℓ ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E))) : hRat S T ρ ℓ ∈ NAA D

theorem **latRat_selfDual** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (∃ (n : ℕ) (s s' : Fin n → Fin S.rank → D.host.E), (∀ i, s i ∈ ΛS ∧ s' i ∈ ΛS) ∧ ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * S.gram k j = if a = j then 1 else 0) → ∀ ρ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ρ ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) ↔ ∀ ℓ ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)), ∃ n : ℤ, Algebra.trace ℚ D.host.E (D.host.xi0 * hRat S T ρ ℓ) = n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RationalHermitianRankOneLatticeSelfDualStatement

theorem **_root_.GlobalTheory.T.rationalHermitianRankOneLattice_selfDual** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (∃ (n : ℕ) (s s' : Fin n → Fin S.rank → D.host.E), (∀ i, s i ∈ ΛS ∧ s' i ∈ ΛS) ∧ ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * S.gram k j = if a = j then 1 else 0) → ∀ ρ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ρ ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) ↔ ∀ ℓ ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)), ∃ n : ℤ, Algebra.trace ℚ D.host.E (D.host.xi0 * hRat S T ρ ℓ) = n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RationalPointsFinitePartSelfDualLatticeIff

theorem **symplO_nsmul** (S T : HSp D) (N : ℕ) (x y : WO S T) : symplO S T x (N • y) = symplO S T (N • x) y

theorem **symplO_eq_add** (S T : HSp D) (x y : WO S T) : symplO S T x y = symplOH S T x y + symplOH S T x y

theorem **psi_real** {θ : InfFam D} {ψ : AddChar (AF D.host.F) ℂ} (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (x y : WO S T) (h : x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) : ψ (symplO S T x y) = Complex.exp (-(2 * Real.pi * Complex.I) * (omO S T (piW θ T S x) (piW θ T S y) : ℂ))

theorem **psi_fin_mul** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (x y : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : ψ (symplO S T (finRat S T x) (finRat S T y)) * Complex.exp (-(2 * Real.pi * Complex.I) * (omO S T (infRat S T x) (infRat S T y) : ℂ)) = 1

theorem **psi_fin_iff** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (x y : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : ψ (symplO S T (finRat S T x) (finRat S T y)) = 1 ↔ ∃ k : ℤ, omO S T (infRat S T x) (infRat S T y) = k

theorem **eq_zero_of_parts** {θ : InfFam D} (hθ : IsInfFam θ) {S T : HSp D} {z : WO S T} (hf : z.map (finPart D) = 0) (hi : z.map (infPart D) = 0) : z = 0

theorem **exists_div** {S T : HSp D} {u : WO S T} (hu : u.map (infPart D) = 0) {N : ℕ} (hN : 0 < N) : ∃ γ : WO S T, γ.map (infPart D) = 0 ∧ N • γ = u

theorem **finRat_mem_iff** : ∀ (D : D7) (S T : HSp D) (ψ : AddChar (AF D.host.F) ℂ) (Λ : AddSubgroup (WO S T)), IsStdChar D ψ (-1) → (∀ u ∈ Λ, u.map (infPart D) = 0) → (∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ Λ ↔ ∀ u' ∈ Λ, ψ (symplO S T u u') = 1)) → (∀ γ : WO S T, γ.map (infPart D) = 0 → ∃ ρ ∈ ratL T S, ∃ u ∈ Λ, (γ - ρ - u).map (finPart D) = 0) → (∀ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ N : ℕ, 0 < N ∧ N • finRat S T γ ∈ Λ) → ∀ x : Matrix (Fin T.rank) (Fin S.rank) D.host.E, finRat S T x ∈ Λ ↔ ∀ y : Matrix (Fin T.rank) (Fin S.rank) D.host.E, finRat S T y ∈ Λ → ∃ k : ℤ, omO S T (infRat S T x) (infRat S T y) = k

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RationalPointsFinitePartSelfDualLatticeIffStatement

theorem **_root_.GlobalTheory.T.rationalPoints_finitePart_selfDual_lattice_iff** : ∀ (D : D7) (S T : HSp D) (ψ : AddChar (AF D.host.F) ℂ) (Λ : AddSubgroup (WO S T)), IsStdChar D ψ (-1) → (∀ u ∈ Λ, u.map (infPart D) = 0) → (∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ Λ ↔ ∀ u' ∈ Λ, ψ (symplO S T u u') = 1)) → (∀ γ : WO S T, γ.map (infPart D) = 0 → ∃ ρ ∈ ratL T S, ∃ u ∈ Λ, (γ - ρ - u).map (finPart D) = 0) → (∀ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ N : ℕ, 0 < N ∧ N • finRat S T γ ∈ Λ) → ∀ x : Matrix (Fin T.rank) (Fin S.rank) D.host.E, finRat S T x ∈ Λ ↔ ∀ y : Matrix (Fin T.rank) (Fin S.rank) D.host.E, finRat S T y ∈ Λ → ∃ k : ℤ, omO S T (infRat S T x) (infRat S T y) = k

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RayCharacterExtendsToIdeals

theorem **exists_multiplicative_ideal_extension** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (c : HeightOneSpectrum (𝒪 K) → ℂ), (∀ P, ¬ m ≤ P.asIdeal → ‖c P‖ = 1) → ∃ X : Ideal (𝒪 K) → ℂ, X ⊥ = 0 ∧ X ⊤ = 1 ∧ (∀ I J, X (I * J) = X I * X J) ∧ (∀ I, I ≠ ⊥ → I ⊔ m = ⊤ → ‖X I‖ = 1) ∧ (∀ I, I ⊔ m ≠ ⊤ → X I = 0) ∧ ∀ P, ¬ m ≤ P.asIdeal → X P.asIdeal = c P

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RayInvariantContinuousEqOne

theorem **exists_near_one_mod_ideal** (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (hm : m ≠ ⊥) : ∃ C : ℝ, ∀ z : mixedSpace K, ∃ β : 𝒪 K, β - 1 ∈ m ∧ ‖z - mixedEmbedding K β‖ ≤ C

theorem **eq_one_of_homogeneous_of_eq_one_on_class** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (φ : mixedSpace K → ℂ), m ≠ ⊥ → ContinuousOn φ {x | mixedEmbedding.norm x ≠ 0} → (∀ (t : ℝ) x, 0 < t → φ (t • x) = φ x) → (∀ β : 𝒪 K, β ≠ 0 → β - 1 ∈ m → φ (mixedEmbedding K β) = 1) → ∀ x, mixedEmbedding.norm x ≠ 0 → φ x = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RealIdemMulOfReal

theorem **gramInv_star** (S : HSp D) (i j : Fin S.rank) : star (S.gram⁻¹ i j) = S.gram⁻¹ j i

theorem **hO_herm** (S T : HSp D) (x y : WO S T) : cA D (hO S T x y) = hO S T y x

theorem **symplO_skew** (S T : HSp D) (x y : WO S T) : symplO S T y x = -symplO S T x y

theorem **hO_smul_left** (S T : HSp D) (a : AE D) (x y : WO S T) : hO S T (a • x) y = a * hO S T x y

theorem **hO_smul_right** (S T : HSp D) (a : AE D) (x y : WO S T) : hO S T x (a • y) = cA D a * hO S T x y

theorem **symplO_smul** (S T : HSp D) (a : AE D) (x y : WO S T) : symplO S T (a • x) y = symplO S T x (cA D a • y)

theorem **symplO_zero_left** (S T : HSp D) (y : WO S T) : symplO S T 0 y = 0

theorem **symplO_zero_right** (S T : HSp D) (x : WO S T) : symplO S T x 0 = 0

theorem **symplO_sub_left** (S T : HSp D) (x x' y : WO S T) : symplO S T (x - x') y = symplO S T x y - symplO S T x' y

theorem **isHalf_iff** {m n : ℕ} (e : AE D) (u : Mx D m n) : IsHalf e u ↔ e • u = u

theorem **symplO_eq_zero** (S T : HSp D) {a b : AE D} (hab : cA D a * b = 0) {x y : WO S T} (hx : IsHalf a x) (hy : IsHalf b y) : symplO S T x y = 0

theorem **symplOH_add_left** (S T : HSp D) (x x' y : WO S T) : symplOH S T (x + x') y = symplOH S T x y + symplOH S T x' y

theorem **symplOH_add_right** (S T : HSp D) (x y y' : WO S T) : symplOH S T x (y + y') = symplOH S T x y + symplOH S T x y'

theorem **symplOH_add_self** (S T : HSp D) (x y : WO S T) : symplOH S T x y + symplOH S T x y = symplO S T x y

theorem **symplOH_skew** (S T : HSp D) (x y : WO S T) : symplOH S T y x = -symplOH S T x y

theorem **heisO_heisO** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w w' : WO S T) (f : WO S T → ℂ) : heisO ψ S T w (heisO ψ S T w' f) = ψ (symplOH S T w w') • heisO ψ S T (w + w') f

theorem **heisO_add_of_orth** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) {w w' : WO S T} (h : symplO S T w w' = 0) (f : WO S T → ℂ) : heisO ψ S T (w + w') f = heisO ψ S T w (heisO ψ S T w' f)

theorem **heisO_comm** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w w' : WO S T) (f : WO S T → ℂ) : heisO ψ S T w (heisO ψ S T w' f) = ψ (symplO S T w w') • heisO ψ S T w' (heisO ψ S T w f)

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **heisL_apply** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w : WO S T) (f : WO S T → ℂ) : heisL ψ S T w f = heisO ψ S T w f

theorem **heisO_smul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w : WO S T) (c : ℂ) (f : WO S T → ℂ) : heisO ψ S T w (c • f) = c • heisO ψ S T w f

theorem **heisO_sum** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w : WO S T) {ι : Type} (s : Finset ι) (f : ι → WO S T → ℂ) : heisO ψ S T w (∑ i ∈ s, f i) = ∑ i ∈ s, heisO ψ S T w (f i)

theorem **cA_realIdem** : cA D (realIdem D) = realIdem D

theorem **infPart_realIdem** : infPart D (realIdem D) = 1

theorem **finPart_realIdem** : finPart D (realIdem D) = 0

theorem **adele_ext** {a b : AE D} (h1 : infPart D a = infPart D b) (h2 : finPart D a = finPart D b) : a = b

theorem **realIdem_mul_of_fin** {a : AE D} (h : infPart D a = 0) : realIdem D * a = 0

theorem **realIdem_mul_of_real** {a : AE D} (h : finPart D a = 0) : realIdem D * a = a

theorem **symplO_real_fin** (S T : HSp D) {x y : WO S T} (hx : x.map (finPart D) = 0) (hy : y.map (infPart D) = 0) : symplO S T x y = 0

theorem **latO_fin** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) : u.map (infPart D) = 0

theorem **isSplitIdem_conj** {e : AE D} (he : IsSplitIdem D e) : IsSplitIdem D (cA D e)

theorem **infPart_splitIdem** {e : AE D} (he : IsSplitIdem D e) : infPart D e = 0

theorem **half_fin** {m n : ℕ} {e : AE D} (he : IsSplitIdem D e) {u : Mx D m n} (hu : IsHalf e u) : u.map (infPart D) = 0

theorem **continuous_symplO** (S T : HSp D) {X : Type} [TopologicalSpace X] {x y : X → WO S T} (hx : Continuous x) (hy : Continuous y) : Continuous fun t => symplO S T (x t) (y t)

theorem **exists_nsmul_mem** {m n : ℕ} (U : AddSubgroup (Mx D m n)) (hU : IsOpen (U : Set (Mx D m n))) {y : Mx D m n} (hy : y.map (infPart D) = 0) : ∃ N : ℕ, 0 < N ∧ N • y ∈ U

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RealPosAtOneOfRight

theorem **real_pos_at_one_of_right** (L : ℂ → ℂ) (hc : ContinuousAt L 1) (h1 : L 1 ≠ 0) (h : ∀ s : ℂ, 1 < s.re → s.im = 0 → (L s).im = 0 ∧ 0 < (L s).re) : (L 1).im = 0 ∧ 0 < (L 1).re

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RealSymplecticFormAlternatingInvariantDefinite

theorem **cartan_swap** {G s : Matrix (Fin k) (Fin k) ℂ} (h : IsCartanInv G s) : sᴴ * G = G * s

theorem **cartan_herm** {G s : Matrix (Fin k) (Fin k) ℂ} (h : IsCartanInv G s) : Gᴴ = G

theorem **cartanInvolution_inv_eq_self** {G s : Matrix (Fin k) (Fin k) ℂ} (h : IsCartanInv G s) : s⁻¹ = s

theorem **cartan_inv_mul** {G s : Matrix (Fin k) (Fin k) ℂ} (h : IsCartanInv G s) : (G * s)⁻¹ = s * G⁻¹

theorem **cartan_inv_conj** {G s : Matrix (Fin k) (Fin k) ℂ} (h : IsCartanInv G s) : s * G⁻¹ * sᴴ = G⁻¹

theorem **trace_pos** {P : Matrix (Fin n) (Fin n) ℂ} {Q : Matrix (Fin m) (Fin m) ℂ} (hP : P.PosDef) (hQ : Q.PosDef) {x : Matrix (Fin m) (Fin n) ℂ} (hx : x ≠ 0) : 0 < (P * xᴴ * Q * x).trace.re

theorem **cplxO_cplxO** {s : Matrix (Fin n) (Fin n) ℂ} {t : Matrix (Fin m) (Fin m) ℂ} (hs : s * s = 1) (ht : t * t = 1) (x : Matrix (Fin m) (Fin n) ℂ) : cplxO s t (cplxO s t x) = -x

theorem **tr_conj** {G : Matrix (Fin n) (Fin n) ℂ} {H : Matrix (Fin m) (Fin m) ℂ} (hG : Gᴴ = G) (hH : Hᴴ = H) (x y : Matrix (Fin m) (Fin n) ℂ) : (G⁻¹ * xᴴ * H * y).trace = star (G⁻¹ * yᴴ * H * x).trace

theorem **tr_J_left** {G s : Matrix (Fin n) (Fin n) ℂ} {H t : Matrix (Fin m) (Fin m) ℂ} (hG : IsCartanInv G s) (x y : Matrix (Fin m) (Fin n) ℂ) : (G⁻¹ * yᴴ * H * cplxO s t x).trace = Complex.I * ((G * s)⁻¹ * yᴴ * (H * t) * x).trace

theorem **tr_J_J** {G s : Matrix (Fin n) (Fin n) ℂ} {H t : Matrix (Fin m) (Fin m) ℂ} (hG : IsCartanInv G s) (hH : IsCartanInv H t) (x y : Matrix (Fin m) (Fin n) ℂ) : (G⁻¹ * (cplxO s t y)ᴴ * H * cplxO s t x).trace = (G⁻¹ * yᴴ * H * x).trace

theorem **realPlace_imaginaryGenerator_re_eq_zero** (D : D7) (τ : D.what₂.Tau) : (tauE D τ D.host.xi0).re = 0

theorem **omAt_zero_right** {D : D7} (S T : HSp D) (τ : D.what₂.Tau) (x : Matrix (Fin T.rank) (Fin S.rank) ℂ) {y : Matrix (Fin T.rank) (Fin S.rank) ℂ} (h : y = 0) : omAt S T τ x y = 0

theorem **omAt_J_neg** {D : D7} (S T : HSp D) (τ : D.what₂.Tau) {s : Matrix (Fin S.rank) (Fin S.rank) ℂ} {t : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hs : IsCartanInv (gramAt D S τ) s) (ht : IsCartanInv (gramAt D T τ) t) {x : Matrix (Fin T.rank) (Fin S.rank) ℂ} (hx : x ≠ 0) : omAt S T τ (cplxO s t x) x < 0

theorem **form_alt_inv_def** : ∀ (D : D7) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → (∀ x, jO S T sS sT (jO S T sS sT x) = -x) ∧ (∀ x y, omO S T x y = -omO S T y x) ∧ (∀ x y, omO S T (jO S T sS sT x) (jO S T sS sT y) = omO S T x y) ∧ ∀ x, x ≠ 0 → omO S T (jO S T sS sT x) x < 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RealSymplecticFormAlternatingInvariantDefiniteStatement

theorem **_root_.GlobalTheory.T.realSymplecticForm_alternating_invariant_definite** : ∀ (D : D7) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → (∀ x, jO S T sS sT (jO S T sS sT x) = -x) ∧ (∀ x y, omO S T x y = -omO S T y x) ∧ (∀ x y, omO S T (jO S T sS sT x) (jO S T sS sT y) = omO S T x y) ∧ ∀ x, x ≠ 0 → omO S T (jO S T sS sT x) x < 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexHigherUnitsSuccLeCard

private theorem **le_exp_sub_one_of_lt_exp** {a : ℤᵐ⁰} {m : ℤ} (h : a < WithZero.exp m) : a ≤ WithZero.exp (m - 1)

theorem **relIndex_hiUnits_succ_le** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (π : K) (R : Finset K) (n : ℕ), w π = WithZero.exp (-1) → (∀ x : K, w x ≤ 1 → ∃ r ∈ R, w (x - r) < 1) → (hiUnits w (n + 1)).relIndex (hiUnits w n) ≠ 0 ∧ (hiUnits w (n + 1)).relIndex (hiUnits

w n) ≤ R.card

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexHigherUnitsSuccLeCardStatement

theorem **_root_.GlobalTheory.T.relIndex_higherUnits_succ_le_card** : ∀ (K : Type) [Field K] (w : Valuation K ℤᵐ⁰) (π : K) (R : Finset K) (n : ℕ), w π = WithZero.exp (-1) → (∀ x : K, w x ≤ 1 → ∃ r ∈ R, w (x - r) < 1) → (hiUnits w (n + 1)).relIndex (hiUnits w n) ≠ 0 ∧ (hiUnits w (n + 1)).relIndex (hiUnits w n) ≤ R.card

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexNormMulRelIndexSqsKer

theorem **relIndex_norm_mul_relIndex_sqs_ker** : ∀ (A : Type) [CommGroup A] (σ : A →* A), (∀ a, σ (σ a) = a) → (∀ a, a * σ a = 1 → ∃ b, a = b / σ b) → (MonoidHom.id A * σ).range.relIndex (σ.eqLocus (MonoidHom.id A)) * (sqs (MonoidHom.id A * σ).ker).relIndex (MonoidHom.id A * σ).ker = (sqs (σ.eqLocus (MonoidHom.id A))).relIndex (σ.eqLocus (MonoidHom.id A))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexNormMulRelIndexSqsKerStatement

theorem **_root_.GlobalTheory.T.relIndex_norm_mul_relIndex_sqs_ker'** : ∀ (A : Type) [CommGroup A] (σ : A →* A), (∀ a, σ (σ a) = a) → (∀ a, a * σ a = 1 → ∃ b, a = b / σ b) → (MonoidHom.id A * σ).range.relIndex (σ.eqLocus (MonoidHom.id A)) * (sqs (MonoidHom.id A * σ).ker).relIndex (MonoidHom.id A * σ).ker = (sqs (σ.eqLocus (MonoidHom.id A))).relIndex (σ.eqLocus (MonoidHom.id A))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexOneUnitsMulEqCardQuotient

theorem **mem_oneUnits_iff_exists_eq_one_add** {O : Type} [CommRing O] (t : O) (u : Oˣ) : u ∈ oneUnits t ↔ ∃ x : O, (u : O) = 1 + t * x

theorem **relIndex_oneUnits_mul_eq_card_quotient** : ∀ (O : Type) [CommRing O] [IsDomain O] (t s : O), t ≠ 0 → s ∣ t → Ideal.span {t} ≤ Ideal.jacobson (⊥ : Ideal O) → (oneUnits (s * t)).relIndex (oneUnits t) = Nat.card (O ⧸ Ideal.span {s})

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexOneUnitsMulEqCardQuotientStatement

theorem **_root_.GlobalTheory.T.relIndex_oneUnits_mul_eq_card_quotient'** : ∀ (O : Type) [CommRing O] [IsDomain O] (t s : O), t ≠ 0 → s ∣ t → Ideal.span {t} ≤ Ideal.jacobson (⊥ : Ideal O) → (oneUnits (s * t)).relIndex (oneUnits t) = Nat.card (O ⧸ Ideal.span {s})

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexSquaresLeTwoMulOfSubgroup

theorem **relIndex_le_two_of_coset** {G : Type} [CommGroup G] (H K : Subgroup G) (ε : G) (h : ∀ x ∈ K, x ∈ H ∨ x * ε⁻¹ ∈ H) : H.relIndex K ≤ 2

theorem **relIndex_sqs_le_two_mul** : ∀ (G : Type) [CommGroup G] (U H : Subgroup G) (ε : G), H ≤ U → H.relIndex U ≠ 0 → (∀ a : G, a ^ 2 = 1 → a = 1 ∨ a = ε) → (sqs H).relIndex H ≠ 0 → (sqs U).relIndex U ≠ 0 ∧ (sqs U).relIndex U ≤ 2 * (sqs H).relIndex H

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexSquaresLeTwoMulOfSubgroupStatement

theorem **_root_.GlobalTheory.T.relIndex_squares_le_two_mul_of_subgroup** : ∀ (G : Type) [CommGroup G] (U H : Subgroup G) (ε : G), H ≤ U → H.relIndex U ≠ 0 → (∀ a : G, a ^ 2 = 1 → a = 1 ∨ a = ε) → (sqs H).relIndex H ≠ 0 → (sqs U).relIndex U ≠ 0 ∧ (sqs U).relIndex U ≤ 2 * (sqs H).relIndex H

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexSquaresSUnitsNumberField

theorem **sqs_subgroupOf_self** {G : Type} [CommGroup G] (U : Subgroup G) : (sqs U).subgroupOf U = sqs (⊤ : Subgroup U)

theorem **card_sq_eq_one_sUnits** (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) : Nat.card {a : sUnits F S // a ^ 2 = 1} = 2

theorem **relIndex_sqs_sUnits** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), (sqs (sUnits F S)).relIndex (sUnits F S) = 2 ^ (S.card + Fintype.card (InfinitePlace F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RelIndexSquaresSUnitsNumberFieldStatement

theorem **_root_.GlobalTheory.T.relIndex_squares_sUnitsNumberField** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), (sqs (sUnits F S)).relIndex (sUnits F S) = 2 ^ (S.card + Fintype.card (InfinitePlace F))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RestrictScalarsMatrixOfBasisHomInjectiveIsUnit

theorem **res_mulVec_repr** {R A β κ : Type} [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] (b : Module.Basis β R A) (g : Matrix κ κ A) (v : κ → A) : res b g *ᵥ (fun p : κ × β => b.repr (v p.1) p.2) = fun p : κ × β => b.repr ((g *ᵥ v) p.1) p.2

theorem **res_one_mul_injective_isUnit** : ∀ (R A β κ : Type) [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A), res b (1 : Matrix κ κ A) = 1 ∧ (∀ g h : Matrix κ κ A, res b (g * h) = res b g * res b h) ∧ Function.Injective (fun g : Matrix κ κ A => res b g) ∧ ∀ g : Matrix κ κ A, IsUnit (res b g).det → IsUnit g

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.RestrictScalarsMatrixOfBasisHomInjectiveIsUnitStatement

theorem **_root_.GlobalTheory.T.restrictScalarsMatrixOfBasis_hom_injective_isUnit** : ∀ (R A β κ : Type) [CommRing R] [CommRing A] [Algebra R A] [Fintype β] [DecidableEq β] [Fintype κ] [DecidableEq κ] (b : Module.Basis β R A), res b (1 : Matrix κ κ A) = 1 ∧ (∀ g h : Matrix κ κ A, res b (g * h) = res b g * res b h) ∧ Function.Injective (fun g : Matrix κ κ A => res b g) ∧ ∀ g : Matrix κ κ A, IsUnit (res b g).det → IsUnit g

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitMemGlobalNormOfMemLocalNorms

theorem **_root_.GlobalTheory.QNorm.index_eq_relIndex_mul_index_sup** {G : Type} [CommGroup G] (U W : Subgroup G) : W.index = W.relIndex U * (U ⊔ W).index

theorem **_root_.GlobalTheory.QNorm.sqs_le_locSq** (F : Type) [Field F] [NumberField F] (U : Subgroup Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))) : _root_.GlobalTheory.QNorm.sqs U ≤ _root_.GlobalTheory.QNorm.locSq F S

private lemma **eq_one_of_mul_self_eq_one_withZero** (a : WithZero (Multiplicative ℤ)) (h : a * a = 1) : a = 1

theorem **_root_.GlobalTheory.QNorm.mem_sqs_sUnits_of_isSquare** (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (y : Fˣ) (hy : y ∈ _root_.GlobalTheory.QNorm.sUnits F S) (h : IsSquare (y : F)) : y ∈ _root_.GlobalTheory.QNorm.sqs (_root_.GlobalTheory.QNorm.sUnits F S)

private lemma **sqs_le_self** {G : Type} [CommGroup G] (U : Subgroup G) : _root_.GlobalTheory.QNorm.sqs U ≤ U

theorem **_root_.GlobalTheory.QNorm.sUnit_mem_normSub_of_mem_locNorm** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), _root_.GlobalTheory.QNorm.admitted F S → δ ∈ _root_.GlobalTheory.QNorm.sUnits F S → ¬ IsSquare (δ : F) → ∀ x ∈ _root_.GlobalTheory.QNorm.sUnits F S, x ∈ _root_.GlobalTheory.QNorm.locNorm F δ S → x ∈ _root_.GlobalTheory.QNorm.normSub (RingHom.id F) (δ : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitMemGlobalNormOfMemLocalNormsStatement

theorem **_root_.GlobalTheory.T.sUnit_mem_globalNorm_of_mem_localNorms** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → δ ∈ sUnits F S → ¬ IsSquare (δ : F) → ∀ x ∈ sUnits F S, x ∈ locNorm F δ S → x ∈ normSub (RingHom.id F) (δ : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitMemKummerNormsOfLocalNorms

theorem **index_eq_relIndex_mul_index_sup_normThm** {G : Type} [CommGroup G] (U W : Subgroup G) : W.index = W.relIndex U * (U ⊔ W).index

theorem **inf_le_of_index_count_normThm** {G : Type} [CommGroup G] (p : ℕ) (hp : 0 < p) (U W D N : Subgroup G) (s k t : ℕ) (hWD : W ≤ D) (hND : N ≤ D) (hk : k + 1 = 2 * s) (hW0 : W.index ≠ 0) (hW : W.index ≤ p ^ (2 * s)) (hWU : W.relIndex U = p ^ k) (hD : p ^ t ≤ D.index) (hN0 : N.relIndex U ≠ 0) (hN : p * N.relIndex U ≤ p ^ t) : D ⊓ U ≤ N

theorem **pows_le_finPow_normThm** (p : ℕ) (F : Type) [Field F] [NumberField F] (U : Subgroup Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))) : pows p U ≤ finPow p F S

theorem **mem_pows_sUnits_normThm** (p : ℕ) (hp : 0 < p) (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))) (y : Fˣ) (hy : y ∈ sUnits F S) (h : ∃ r : F, r ^ p = (y : F)) : y ∈ pows p (sUnits F S)

theorem **sUnit_mem_kNorm_of_mem_finNorm** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → ∀ x ∈ sUnits F S, x ∈ finNorm p F δ S → x ∈ kNorm p (RingHom.id F) (δ : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitMemKummerNormsOfLocalNormsStatement

theorem **_root_.GlobalTheory.T.sUnit_mem_kummerNorms_of_local_norms** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → ∀ x ∈ sUnits F S, x ∈ finNorm p F δ S → x ∈ kNorm p (RingHom.id F) (δ : F)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsNumberFieldFg

private lemma **valuation_algebraMap_unit_eq_one** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (u : (𝒪 F)ˣ) : v.valuation F (algebraMap (𝒪 F) F u) = 1

private lemma **exists_unit_of_valuation_eq_one** {F : Type} [Field F] [NumberField F] (y : Fˣ) (hy : ∀ v : HeightOneSpectrum (𝒪 F), v.valuation F y = 1) : ∃ u : (𝒪 F)ˣ, algebraMap (𝒪 F) F u = y

theorem **_root_.GlobalTheory.QNorm.sUnits_fg** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), Group.FG (_root_.GlobalTheory.QNorm.sUnits F S)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsNumberFieldFgStatement

theorem **_root_.GlobalTheory.T.sUnitsNumberField_fg** : ∀ (F : Type) [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝒪 F))), Group.FG (sUnits F S)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsPlacesOverAndQuadraticConj

theorem **sUnits_over_and_conj** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → (∀ x : Fˣ, Units.map (algebraMap F E : F →* E) x ∈ sUnits E SE ↔ x ∈ sUnits F S) ∧ ∀ y : Eˣ, y ∈ sUnits E SE → Units.map (σ : E →* E) y ∈ sUnits E SE

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsPlacesOverAndQuadraticConjStatement

theorem **_root_.GlobalTheory.T.sUnits_places_over_and_quadratic_conj** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → (∀ x : Fˣ, Units.map (algebraMap F E : F →* E) x ∈ sUnits E SE ↔ x ∈ sUnits F S) ∧ ∀ y : Eˣ, y ∈ sUnits E SE → Units.map (σ : E →* E) y ∈ sUnits E SE

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsSupLocalKummerNormsNeTop

theorem **sUnits_sup_finNorm_ne_top** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → sUnits F S ⊔ finNorm p F δ S ≠ ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsSupLocalKummerNormsNeTopStatement

theorem **_root_.GlobalTheory.T.sUnits_sup_localKummerNorms_ne_top** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (ζ : F) (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → IsPrimitiveRoot ζ p → AdmP p F S → δ ∈ sUnits F S → (∀ r : F, r ^ p ≠ (δ : F)) → sUnits F S ⊔ finNorm p F δ S ≠ ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsSupUnitsLocalNormsSqrtNeTop

theorem **normSub_id_le_normSub** {F L : Type} [Field F] [Field L] (φ : F →+* L) (δ : F) : normSub (RingHom.id F) δ ≤ normSub φ δ

theorem **normSub_id_le_locNorm** (F : Type) [Field F] [NumberField F] (δ : F) (S : Finset (HeightOneSpectrum (𝒪 F))) : normSub (RingHom.id F) δ ≤ locNorm F δ S

theorem **sUnits_sup_locNorm_ne_top** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → δ ∈ sUnits F S → ¬ IsSquare (δ : F) → sUnits F S ⊔ locNorm F δ S ≠ ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsSupUnitsLocalNormsSqrtNeTopStatement

theorem **_root_.GlobalTheory.T.sUnits_sup_unitsLocalNormsSqrt_ne_top** : ∀ (F : Type) [Field F] [NumberField F] (δ : Fˣ) (S : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → δ ∈ sUnits F S → ¬ IsSquare (δ : F) → sUnits F S ⊔ locNorm F δ S ≠ ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsUnionSupLocalSquaresEqTop

theorem **sUnits_union_sup_finSq_eq_top** : ∀ (F : Type) [Field F] [NumberField F] [DecidableEq (HeightOneSpectrum (𝒪 F))] (S T : Finset (HeightOneSpectrum (𝒪 F))), admitted F S → Disjoint S T → (∀ A : Finset (HeightOneSpectrum (𝒪 F)), ∃ u ∈ sUnits F S, ∀ v ∈ T, (IsSquare (algebraMap F (v.adicCompletion F) (u : F)) ↔ v ∉ A)) → sUnits F (S ∪ T) ⊔ finSq F T = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsUnionSupPowersEqTopAtPlaces

theorem **sUnits_union_sup_finPow_eq_top** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [DecidableEq (HeightOneSpectrum (𝒪 F))] (S T : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → AdmP p F S → Disjoint S T → (∀ v ∈ T, ∃ u ∈ sUnits F S, u ∉ powSub p (algebraMap F (v.adicCompletion F)) ∧ ∀ v' ∈ T, v' ≠ v → u ∈ powSub p (algebraMap F (v'.adicCompletion F))) → sUnits F (S ∪ T) ⊔ finPow p F T = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SUnitsUnionSupPowersEqTopAtPlacesStatement

theorem **_root_.GlobalTheory.T.sUnits_union_sup_powers_eq_top_at_places** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [DecidableEq (HeightOneSpectrum (𝒪 F))] (S T : Finset (HeightOneSpectrum (𝒪 F))), p.Prime → AdmP p F S → Disjoint S T → (∀ v ∈ T, ∃ u ∈ sUnits F S, u ∉ powSub p (algebraMap F (v.adicCompletion F)) ∧ ∀ v' ∈ T, v' ≠ v → u ∈ powSub p (algebraMap F (v'.adicCompletion F))) → sUnits F (S ∪ T) ⊔ finPow p F T = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ScaledFiniteMatrixMemLattice

theorem **exists_nsmul_mem_lattice** {E : Type} [Field E] [NumberField E] {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (h : Submodule.IsLattice E Λ) (v : Fin n → E) : ∃ N : ℕ, 0 < N ∧ N • v ∈ Λ

theorem **AA_ne_zero** (D : D7) : D.host.AA ≠ 0

theorem **exists_nsmul_finRat_mem** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), Submodule.IsLattice D.host.E ΛS → Submodule.IsLattice D.host.E ΛT → ∀ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ N : ℕ, 0 < N ∧ N • finRat S T γ ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ScaledFiniteMatrixMemLatticeStatement

theorem **_root_.GlobalTheory.T.scaled_finite_matrix_mem_lattice** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), Submodule.IsLattice D.host.E ΛS → Submodule.IsLattice D.host.E ΛT → ∀ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ N : ℕ, 0 < N ∧ N • finRat S T γ ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ScaledMatrixMemRationalLattice

theorem **exists_nat_integral** {E : Type} [Field E] [NumberField E] (c : E) : ∃ n : ℕ, 0 < n ∧ ∃ d : 𝒪 E, (d : E) = n * c

theorem **AA_ne_zero** (D : D7) : D.host.AA ≠ 0

theorem **genMap_one** (ha0 : a ≠ 0) : genMap S T 1 s t = a⁻¹ • genMap S T a s t

theorem **genMap_zero_right** : genMap S T 1 s 0 = 0

theorem **genMap_add_right** : genMap S T 1 s (u + v) = genMap S T 1 s u + genMap S T 1 s v

theorem **genMap_smul_right** : genMap S T 1 s (c • u) = c • genMap S T 1 s u

theorem **genMap_zero_left** : genMap S T 1 0 t = 0

theorem **genMap_add_left** : genMap S T 1 (x + y) t = genMap S T 1 x t + genMap S T 1 y t

theorem **genMap_smul_left** : genMap S T 1 (c • x) t = star c • genMap S T 1 x t

theorem **genMap_mem_sat** (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (hS : Submodule.IsLattice D.host.E ΛS) (hT : Submodule.IsLattice D.host.E ΛT) (s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) : genMap S T 1 s t ∈ sat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E))

theorem **latRat_full** (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (hS : Submodule.IsLattice D.host.E ΛS) (hT : Submodule.IsLattice D.host.E ΛT) (M : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : ∃ N : ℕ, 0 < N ∧ N • M ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ScaledMatrixMemRationalLatticeStatement

theorem **_root_.GlobalTheory.T.scaled_matrix_mem_rational_lattice** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), Submodule.IsLattice D.host.E ΛS → Submodule.IsLattice D.host.E ΛT → ∀ M : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ N : ℕ, 0 < N ∧ N • M ∈ latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SelfDualLatticePairingDual

theorem **tensor_prod_eq_zero** {F A B E : Type} [Field F] [CommRing A] [CommRing B] [CommRing E] [Algebra F A] [Algebra F B] [Algebra F E] (z : (A × B) ⊗[F] E) (h1 : Algebra.TensorProduct.map (AlgHom.fst F A B) (AlgHom.id F E) z = 0) (h2 : Algebra.TensorProduct.map (AlgHom.snd F A B) (AlgHom.id F E) z = 0) : z = 0

theorem **approx_tmul** {F A E : Type} [Field F] [CommRing A] [CommRing E] [Algebra F A] [Algebra F E] (a : F) (N : ℕ) (β : A) (x : E) : (algebraMap F A a + N • β) ⊗ₜ[F] x = (1 : A) ⊗ₜ[F] (a • x) + β ⊗ₜ[F] (N • x)

theorem **eq_zero_of_parts** (z : Formalism.L2.Theta.AE D) (h1 : infPart D z = 0) (h2 : finPart D z = 0) : z = 0

theorem **infPart_pair** (α : Af D.host.F) (x : D.host.E) : infPart D (pairA D.host.F 0 α ⊗ₜ[D.host.F] x) = 0

theorem **finPart_tmul** (a : AF D.host.F) (x : D.host.E) : finPart D (a ⊗ₜ[D.host.F] x) = sndA D.host.F a ⊗ₜ[D.host.F] x

theorem **finPart_pair** (α : Af D.host.F) (x : D.host.E) : finPart D (pairA D.host.F 0 α ⊗ₜ[D.host.F] x) = α ⊗ₜ[D.host.F] x

theorem **finPart_incA** (x : D.host.E) : finPart D (incA D x) = (1 : Af D.host.F) ⊗ₜ[D.host.F] x

theorem **one_mem_intAf** (F : Type) [Field F] [NumberField F] : (1 : Af F) ∈ intAf F

theorem **mul_mem_intAf** {F : Type} [Field F] [NumberField F] {a b : Af F} (ha : a ∈ intAf F) (hb : b ∈ intAf F) : a * b ∈ intAf F

theorem **infPart_of_mem** {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) (i : Fin T.rank) (j : Fin S.rank) : infPart D (u i j) = 0

theorem **hRat_smul** (c : D.host.E) (M N : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : hRat S T (c • M) N = c * hRat S T M N

theorem **char_rat** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (z : D.host.E) : ψ (pairA D.host.F 0 (algebraMap D.host.F (Af D.host.F) (Algebra.trace D.host.F D.host.E z))) = 1 ↔ ∃ n : ℤ, Algebra.trace ℚ D.host.E z = n

theorem **char_sympl_eq_one** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) {u u' : WO S T} (hu : u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) (hu' : u' ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) : ψ (symplO S T u u') = 1

theorem **approx** (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) (γ : WO S T) : ∃ M : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True, ∀ i j, finPart D (γ i j) = (1 : Af D.host.F) ⊗ₜ[D.host.F] M i j + finPart D (u i j)

theorem **mem_of_char** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) {u : WO S T} (hu : u.map (infPart D) = 0) (h : ∀ u' ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True, ψ (symplO S T u u') = 1) : u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **pair_mul** {F : Type} [Field F] [NumberField F] (α β : Af F) : pairA F 0 α * pairA F 0 β = pairA F 0 (α * β)

theorem **pair_add** {F : Type} [Field F] [NumberField F] (α β : Af F) : pairA F 0 (α + β) = pairA F 0 α + pairA F 0 β

theorem **finTens_add_left** {m n : ℕ} (α β : Af D.host.F) (M : Matrix (Fin m) (Fin n) D.host.E) : finTens (α + β) M = finTens α M + finTens β M

theorem **finTens_zero_left** {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) : finTens (0 : Af D.host.F) M = 0

theorem **tmul_smul_finTens** {m n : ℕ} (α β : Af D.host.F) (x : D.host.E) (M : Matrix (Fin m) (Fin n) D.host.E) : (pairA D.host.F 0 α ⊗ₜ[D.host.F] x : Formalism.L2.Theta.AE D) • finTens β M = finTens (α * β) (x • M)

theorem **cA_cA** (z : Formalism.L2.Theta.AE D) : cA D (cA D z) = z

theorem **le_of_two_of_half** {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} (G : AddSubgroup (WO S T)) (h2 : ∀ u ∈ latO S T LS LT fun _ => True, 2 • u ∈ G) (hh : ∀ e, IsSplitIdem D e → ∀ u ∈ latO S T LS LT fun _ => True, IsHalf e u → u ∈ G) {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) : u ∈ G

theorem **heisO_heisO** (ψ : AddChar (AF D.host.F) ℂ) (u u' : WO S T) (f : WO S T → ℂ) : heisO ψ S T u (heisO ψ S T u' f) = ψ (symplOH S T u u') • heisO ψ S T (u + u') f

theorem **heisO_smul** (ψ : AddChar (AF D.host.F) ℂ) (u : WO S T) (c : ℂ) (f : WO S T → ℂ) : heisO ψ S T u (c • f) = c • heisO ψ S T u f

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **char_half** {ψ : AddChar (AF D.host.F) ℂ} {u u' : WO S T} (h : ψ (symplO S T u u') = 1) : ψ (symplOH S T u u') * ψ (symplOH S T u u') = 1

theorem **latO_selfDual_approx_sign** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → (∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) ↔ ∀ u' ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), ψ (symplO S T u u') = 1)) ∧ (∀ γ : WO S T, γ.map (infPart D) = 0 → ∃ ρ ∈ ratL T S, ∃ u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), (γ - ρ - u).map (finPart D) = 0) ∧ ∀ f : WO S T → ℂ, IsLatVac (heisO ψ S T) (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) f → ∀ u ∈ (latO S T (ΛS : Set (Fin S.rank

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SelfDualLatticePairingDualStatement

theorem **_root_.GlobalTheory.T.self_dual_lattice_pairing_dual** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → (∀ u : WO S T, u.map (infPart D) = 0 → (u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) ↔ ∀ u' ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), ψ (symplO S T u u') = 1)) ∧ (∀ γ : WO S T, γ.map (infPart D) = 0 → ∃ ρ ∈ ratL T S, ∃ u ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), (γ - ρ - u).map (finPart D) = 0) ∧ ∀ f : WO S T → ℂ, IsLatVac (heisO ψ S T) (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) f → ∀ u ∈ (lat

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SiegelCover

private lemma **le_two_pow_mul_of_consecutive_le** {N : ℕ} (a : Fin N → ℝ) (ha : ∀ i, 0 < a i) (h : ∀ p q : Fin N, (q : ℕ) = p + 1 → a p ≤ 2 * a q) (i j : Fin N) (hij : i < j) : a i ≤ 2 ^ N * a j

private lemma **exists_reduced_iwasawa_decomposition** {ι : Type} [Fintype ι] [DecidableEq ι] [LinearOrder ι] (g : Matrix ι ι ℝ) (hg : g.det ≠ 0) : ∃ (γ : Matrix ι ι ℤ) (k : Matrix ι ι ℝ) (a : ι → ℝ) (n : Matrix ι ι ℝ), (γ.det = 1 ∨ γ.det = -1) ∧ kᵀ * k = 1 ∧ (∀ i, 0 < a i) ∧ (∀ i, n i i = 1) ∧ (∀ i j, j < i → n i j = 0) ∧ (∀ i j, |n i j| ≤ 1) ∧ (∀ i j, i < j → a i ≤ 2 ^ Fintype.card ι * a j) ∧ g * γ.map (Int.cast : ℤ → ℝ) = k * diagonal a * n

private lemma **isSiegel_of_reduced** {ι : Type} [Fintype ι] [DecidableEq ι] [LinearOrder ι] {C : ℝ} {g : Matrix ι ι ℝ} {γ : Matrix ι ι ℤ} {k : Matrix ι ι ℝ} {a : ι → ℝ} {n : Matrix ι ι ℝ} (hC : 1 ≤ C) (hg : g.det = 1) (hγ : γ.det = 1) (hk : kᵀ * k = 1) (ha : ∀ i, 0 < a i) (hn1 : ∀ i, n i i = 1) (hn0 : ∀ i j, j < i → n i j = 0) (hnb : ∀ i j, |n i j| ≤ 1) (hlt : ∀ i j, i < j → a i ≤ C * a j) (hdec : g * γ.map (Int.cast : ℤ → ℝ) = k * diagonal a * n) : _root_.GlobalTheory.ClassFin.IsSiegel C k a n

private lemma **exists_det_one_of_det_neg_one** {ι : Type} [Fintype ι] [DecidableEq ι] [LinearOrder ι] {g : Matrix ι ι ℝ} {γ : Matrix ι ι ℤ} {k : Matrix ι ι ℝ} {a : ι → ℝ} {n : Matrix ι ι ℝ} (hγ : γ.det = -1) (hk : kᵀ * k = 1) (hn1 : ∀ i, n i i = 1) (hn0 : ∀ i j, j < i → n i j = 0) (hnb : ∀ i j, |n i j| ≤ 1) (hdec : g * γ.map (Int.cast : ℤ → ℝ) = k * diagonal a * n) : ∃ (γ' : Matrix ι ι ℤ) (k' n' : Matrix ι ι ℝ), γ'.det = 1 ∧ k'ᵀ * k' = 1 ∧ (∀ i, n' i i = 1) ∧ (∀ i j, j < i → n' i j = 0) ∧ (∀ i j, |n' i j| ≤ 1) ∧ g * γ'.map (Int.cast : ℤ → ℝ) = k' * diagonal a * n'

theorem **_root_.GlobalTheory.ClassFin.siegel_cover** (ι : Type) [Fintype ι] [DecidableEq ι] [LinearOrder ι] : ∃ C : ℝ, ∀ g : Matrix ι ι ℝ, g.det = 1 → ∃ γ : Matrix ι ι ℤ, γ.det = 1 ∧ ∃ k a n, _root_.GlobalTheory.ClassFin.IsSiegel C k a n ∧ g * γ.map (Int.cast : ℤ → ℝ) = k * diagonal a * n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SiegelCoverStatement

theorem **_root_.GlobalTheory.T.exists_siegel_decomposition_mul_int_matrix_of_det_eq_one** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] [LinearOrder ι], ∃ C : ℝ, ∀ g : Matrix ι ι ℝ, g.det = 1 → ∃ γ : Matrix ι ι ℤ, γ.det = 1 ∧ ∃ k a n, IsSiegel C k a n ∧ g * γ.map (Int.cast : ℤ → ℝ) = k * diagonal a * n

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SmoothedIdealCharacterSumMainTermIdealClassRepre

private theorem **exists_periodic_mod_ideal_eq_character_div_infinity_type** {K : Type} [Field K] [NumberField K] (m : Ideal (𝒪 K)) (X : Ideal (𝒪 K) → ℂ) (φ : mixedSpace K → ℂ) (hXmul : ∀ I J, X (I * J) = X I * X J) (hXcop : ∀ I, I ⊔ m ≠ ⊤ → X I = 0) (hφmul : ∀ x y, φ (x * y) = φ x * φ y) (hφnorm : ∀ x, mixedEmbedding.norm x ≠ 0 → ‖φ x‖ = 1) (hXφ : ∀ β : 𝒪 K, β ≠ 0 → β - 1 ∈ m → X (Ideal.span {β}) = φ (mixedEmbedding K β)) : ∃ ψ : 𝒪 K → ℂ, (∀ a b, a - b ∈ m → ψ a = ψ b) ∧ ∀ β : 𝒪 K, β ≠ 0 → X (Ideal.span {β}) = ψ β * φ (mixedEmbedding K β)

private theorem **contDiffAt_mixedEmbedding_norm_of_ne_zero** {K : Type} [Field K] [NumberField K] {x : mixedSpace K} (hx : mixedEmbedding.norm x ≠ 0) : ContDiffAt ℝ ∞ (↑‖(mixedEmbedding.norm (K

private theorem **contDiff_and_hasCompactSupport_mul_weight_mixedNorm** {K : Type} [Field K] [NumberField K] (h : mixedSpace K → ℝ) (hh : ContDiffOn ℝ ∞ h {x | mixedEmbedding.norm x ≠ 0}) (hC : ∀ a b : ℝ, 0 < a → ∃ C, IsCompact C ∧ (∀ x ∈ C, mixedEmbedding.norm x ≠ 0) ∧ ∀ x : mixedSpace K, h x ≠ 0 → a ≤ mixedEmbedding.norm x → mixedEmbedding.norm x ≤ b → x ∈ C) (φ : mixedSpace K → ℂ) (hφ : ContDiffOn ℝ ∞ φ {x | mixedEmbedding.norm x ≠ 0}) (wt : ℝ → ℂ) (hwt : ContDiff ℝ ∞ wt) (a b : ℝ) (ha : 0 < a) (hab : ∀ y, wt y ≠ 0 → a ≤ y ∧ y ≤ b) : ContDiff ℝ ∞ (fun x => (h x : ℂ) * φ x * wt (mixedEmbedding.norm x)) ∧ HasCompactSupport (fun x => (h x : ℂ) * φ x * wt (mixedEmbedding.norm x))

private theorem **finite_setOf_ringOfIntegers_mixedEmbedding_mem_compact** {K : Type} [Field K] [NumberField K] (C : Set (mixedSpace K)) (hC : IsCompact C) : {β : 𝒪 K | mixedEmbedding K (β : K) ∈ C}.Finite

private theorem **mixedEmbedding_norm_eq_absNorm_span_singleton** {K : Type} [Field K] [NumberField K] (β : 𝒪 K) : mixedEmbedding.norm (mixedEmbedding K (β : K)) = (Ideal.absNorm (Ideal.span {β}) : ℝ)

private theorem **tsum_principal_le_eq_tsum_mul_cutoff_span** {K : Type} [Field K] [NumberField K] (B : Ideal (𝒪 K)) (G : Ideal (𝒪 K) → ℂ) (hG0 : G ⊥ = 0) (h : mixedSpace K → ℝ) (hsum : ∀ x : mixedSpace K, mixedEmbedding.norm x ≠ 0 → HasSum (fun u : (𝒪 K)ˣ => h (u • x)) 1) (hfin : (Function.support fun β : B => (h (mixedEmbedding K ((β : 𝒪 K) : K)) : ℂ) * G (Ideal.span {(β : 𝒪 K)})).Finite) : ∑' J : {J : Ideal (𝒪 K) // J ≠ ⊥ ∧ J.IsPrincipal ∧ J ≤ B}, G J = ∑' β : B, (h (mixedEmbedding K ((β : 𝒪 K) : K)) : ℂ) * G (Ideal.span {(β : 𝒪 K)})

private theorem **finite_support_mul_weight_absNorm_of_bounded_support** {K : Type} [Field K] [NumberField K] (X : Ideal (𝒪 K) → ℂ) (wt : ℝ → ℂ) (a b : ℝ) (hw : ∀ y, wt y ≠ 0 → a ≤ y ∧ y ≤ b) (t : ℝ) (ht : 0 < t) : (Function.support fun I : Ideal (𝒪 K) => X I * wt (t * (Ideal.absNorm I : ℝ))).Finite

private theorem **tsum_ideal_eq_sum_classGroup_tsum_mul_isPrincipal** {K : Type} [Field K] [NumberField K] (b : ClassGroup (𝒪 K) → (Ideal (𝒪 K))⁰) (hb : Function.Bijective (fun c => ClassGroup.mk0 (b c))) (F : Ideal (𝒪 K) → ℂ) (hF0 : F ⊥ = 0) (hfin : (Function.support F).Finite) : ∑' I, F I = ∑ c, ∑' I : {I : Ideal (𝒪 K) // I ≠ ⊥ ∧ (I * (b c : Ideal (𝒪 K))).IsPrincipal}, F I

theorem **_root_.GlobalTheory.smoothedSum_main_term_representatives** : ∀ (K : Type) [Field K] [NumberField K] (m : Ideal (𝒪 K)) (X : Ideal (𝒪 K) → ℂ) (φ : mixedSpace K → ℂ), m ≠ ⊥ → X ⊥ = 0 → (∀ I J, X (I * J) = X I * X J) → (∀ I, I ≠ ⊥ → I ⊔ m = ⊤ → ‖X I‖ = 1) → (∀ I, I ⊔ m ≠ ⊤ → X I = 0) → (∀ x y, φ (x * y) = φ x * φ y) → (∀ (t : ℝ) x, 0 < t → φ (t • x) = φ x) → ContDiffOn ℝ ∞ φ {x | mixedEmbedding.norm x ≠ 0} → (∀ x, mixedEmbedding.norm x ≠ 0 → ‖φ x‖ = 1) → (∀ β : 𝒪 K, β ≠ 0 → β - 1 ∈ m → X (Ideal.span {β}) = φ (mixedEmbedding K β)) → ∀ (wt : ℝ → ℂ), ContDiff ℝ ∞ wt → (∃ a b : ℝ, 0 < a ∧ ∀ y, wt y ≠ 0 → a ≤ y ∧ y ≤ b) → ∃ κ : ℂ, ∀ b : ClassGroup (𝒪 K) → (Ideal (𝒪 K))⁰, Function.Bijective (fun c => ClassGroup.mk0 (b c)) → (∀ c, (b c : Ideal (𝒪 K)) ⊔ m = ⊤) → ∀ A : ℕ, ∃ C : ℝ, ∀ t : ℝ, 0 < t → t ≤ 1 → ‖∑' I, X I * wt (t * (Ideal.absNorm I : ℝ)) - (κ * ∑ c, (X (b c))⁻¹) / (t : ℂ)‖ ≤ C * t ^

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SmulEqZeroOfInf

theorem **smul_eq_zero_of_inf** {a : AE D} (h : infPart D a = 0) : eInf D • a = 0

theorem **smul_eq_self_of_fin** {a : AE D} (h : finPart D a = 0) : eInf D • a = a

theorem **cA_smul** (c : AF D.host.F) (a : AE D) : cA D (c • a) = c • cA D a

theorem **symplOH_add_left** (x x' y : WO S T) : symplOH S T (x + x') y = symplOH S T x y + symplOH S T x' y

theorem **symplOH_add_right** (x y y' : WO S T) : symplOH S T x (y + y') = symplOH S T x y + symplOH S T x y'

theorem **symplOH_smul_left** (c : AF D.host.F) (x y : WO S T) : symplOH S T (c • x) y = c * symplOH S T x y

theorem **symplOH_smul_right** (c : AF D.host.F) (x y : WO S T) : symplOH S T x (c • y) = c * symplOH S T x y

theorem **symplOH_zero_left** (y : WO S T) : symplOH S T 0 y = 0

theorem **symplOH_zero_right** (y : WO S T) : symplOH S T y 0 = 0

theorem **symplOH_neg_right** (x y : WO S T) : symplOH S T x (-y) = -symplOH S T x y

theorem **hO_herm** (x y : WO S T) : cA D (hO S T x y) = hO S T y x

theorem **symplOH_skew** (x y : WO S T) : symplOH S T y x = -symplOH S T x y

theorem **symplOH_self** (x : WO S T) : symplOH S T x x = 0

theorem **smul_fin** {u : WO S T} (hu : u.map (infPart D) = 0) : eInf D • u = 0

theorem **smul_real** {v : WO S T} (hv : v.map (finPart D) = 0) : eInf D • v = v

theorem **symplOH_fin_real** {u v : WO S T} (hu : u.map (infPart D) = 0) (hv : v.map (finPart D) = 0) : symplOH S T u v = 0

theorem **symplOH_real_fin** {u v : WO S T} (hu : u.map (infPart D) = 0) (hv : v.map (finPart D) = 0) : symplOH S T v u = 0

theorem **continuous_symplOH** (u : WO S T) : Continuous fun w : WO S T => symplOH S T w u

theorem **ker_mem_nhds** {F : Type} [Field F] [NumberField F] {ψ : AddChar (AF F) ℂ} (hc : Continuous ψ) (hn : ∀ x, ‖ψ x‖ = 1) : {a : Af F | ψ (pairA F 0 a) = 1} ∈ 𝒩 (0 : Af F)

theorem **heisO_heisO** (w w' : WO S T) (f : WO S T → ℂ) : heisO ψ S T w (heisO ψ S T w' f) = ψ (symplOH S T w w') • heisO ψ S T (w + w') f

theorem **isThetaFunO_heisO** (hc : Continuous ψ) (hn : ∀ x, ‖ψ x‖ = 1) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {u : WO S T} (hu : u.map (infPart D) = 0) : IsThetaFunO ψ S T (heisO ψ S T u f)

theorem **dRho_heisO** {θ : InfFam D} (hθ : IsInfFam θ) {u : WO S T} (hu : u.map (infPart D) = 0) (τ : D.whatₔ.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (g : WO S T → ℂ) : Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z (heisO ψ S T u g) = heisO ψ S T u (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z g)

theorem **pureTensorL_comm** {ι X : Type} (A : ι → (X → ℂ) → (X → ℂ)) (H : (X → ℂ) →ₗ[ℂ] (X → ℂ)) (h : ∀ i g, A i (H g) = H (A i g)) (f : X → ℂ) (P : MvPolynomial ι ℂ) : pureTensorL A (H f) P = H (pureTensorL A f P)

theorem **finite_translate** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D), IsStdChar D ψ (-1) → IsInfFam θ → ∀ f, IsThetaFunO ψ S T f → (∀ (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (u v : WO S T), u.map (infPart D) = 0 → v.map (finPart D) = 0 → ratW T S γ = u + v → heisO ψ S T u f 0 = heisO ψ S T (-v) f 0) ∧ ∀ (u : WO S T), u.map (infPart D) = 0 → IsThetaFunO ψ S T (heisO ψ

S T u f) ∧ ∀ (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (P : MvPolynomial (FVar D T.rank S.rank) ℂ), pureTensorO ψ θ S T sS sT BS BT (heisO ψ S T u f) P = heisO ψ S T u (pureTensorO ψ θ S T sS sT BS BT f P)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SmulMemLatO

theorem **liftFin_tmul** (D : D7) (α : Af D.host.F) (m : D.host.E) : liftFin D (α ⊗ₜ[D.host.F] m) = pairA D.host.F 0 α ⊗ₜ[D.host.F] m

theorem **mul_pair_tmul** (D : D7) (w : AE D) (α : Af D.host.F) (m : D.host.E) : w ∗ (pairA D.host.F 0 α ⊗ₜ[D.host.F] m) = liftFin D (finPart D w ∗ (α ⊗ₜ[D.host.F] m))

theorem **smul_mem_latO** (D : D7) (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : HeightOneSpectrum (𝒪 D.host.F) → Prop) (w : AE D) (hw : finPart D w ∈ intAE D) : ∀ u ∈ latO S T LS LT P, w • u ∈ latO S T LS LT P

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SpanRangeEqTopOfIsIrreducible

private theorem **subalgebra_end_eq_whole_of_isSimpleModule** {V : Type} [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (A : Subalgebra ℂ (Module.End ℂ V)) [IsSimpleModule A V] : A = ⊤

theorem **_root_.GlobalTheory.ClassFin.span_range_eq_top_of_isIrreducible** : ∀ (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) [FiniteDimensional ℂ V], ρ.IsIrreducible → Submodule.span ℂ (Set.range ρ) = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SpecialLinearClassNumberOneAdelic

theorem **exists_matrix_span_of_smul_mem** {ι : Type} [Fintype ι] [DecidableEq ι] (L : Submodule ℤ (ι → ℤ)) (N : ℤ) (hN : N ≠ 0) (hL : ∀ v : ι → ℤ, N • v ∈ L) : ∃ H : Matrix ι ι ℤ, (∀ j, (fun i => H i j) ∈ L) ∧ ∀ v ∈ L, ∃ c : ι → ℤ, v = H *ᵥ c

theorem **exists_matrix_basis_of_lattice** {ι : Type} [Fintype ι] [DecidableEq ι] (L : Submodule ℤ (ι → ℚ)) (M : ℕ) (hM : 0 < M) (h1 : ∀ v ∈ L, ∀ i, ∃ z : ℤ, (M : ℚ) * v i = z) (h2 : ∀ z : ι → ℤ, (fun i => (M : ℚ) * z i) ∈ L) : ∃ h g : Matrix ι ι ℚ, h * g = 1 ∧ (∀ j, (fun i => h i j) ∈ L) ∧ ∀ v ∈ L, ∀ i, ∃ z : ℤ, (g *ᵥ v) i = z

theorem **det_mem_subring_of_entries** {ι : Type} [Fintype ι] [DecidableEq ι] {R : Type} [CommRing R] (R₀ : Subring R) (k : Matrix ι ι R) (hk : ∀ i j, k i j ∈ R₀) : k.det ∈ R₀

theorem **sl_class_number_one** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), AdelicPair R R₀ → ∀ x : Matrix ι ι R, x.det = 1 → ∃ (h : Matrix ι ι ℚ) (k : Matrix ι ι R), h.det = 1 ∧ k.det = 1 ∧ (∀ i j, k i j ∈ R₀) ∧ x = h.map (algebraMap ℚ R) ∗ k

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SpecialLinearClassNumberOneAdelicStatement

theorem **_root_.GlobalTheory.T.specialLinear_classNumber_one_adelic** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (R : Type) [CommRing R] [Algebra ℚ R] (R₀ : Subring R), AdelicPair R R₀ → ∀ x : Matrix ι ι R, x.det = 1 → ∃ (h : Matrix ι ι ℚ) (k : Matrix ι ι R), h.det = 1 ∧ k.det = 1 ∧ (∀ i j, k i j ∈ R₀) ∧ x = h.map (algebraMap ℚ R) ∗ k

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SqNormIntegralFundamentalDomainEqL2NormSq

theorem **actO_two_nsmul** (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g h (2 • x) = 2 • actO S T g h x

theorem **actO_inv_actO** (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g⁻¹ h⁻¹ (actO S T g h x) = x

theorem **isHalf_actO** (e : AE D) (g : GA D S) (h : GA D T) {x : WO S T} (hx : IsHalf e x) : IsHalf e (actO S T g h x)

theorem **sqNormIntegralFundamentalDomain_eq_l2Norm_sq** (μ : Measure (WO S T)) (ℱ : Set (WO S T)) {f : WO S T → ℂ} (hf : Continuous f) : nrm2 μ ℱ f = (∫⁻ w in ℱ, ‖f w‖ₑ ^ 2 ∂μ).toReal

theorem **isVacO_om** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →∗ Circle) (om : GA D S × GA D T →∗ Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → ∀ (sS sS' : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT sT' : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (LS LS' : Set (Fin S.rank → D.host.E)) (LT LT' : Set (Fin T.rank → D.host.E)) (g : GA D S) (h : GA D T), (∀ τ, sS' τ ∗ compAt θ τ (mt S g) = compAt θ τ (mt S g) ∗ sS τ) → (∀ τ, sT' τ ∗ compAt θ τ (mt T h) = compAt θ τ (mt T h) ∗ sT τ) → (∀ u : WO S T, u ∈ (latO S T LS' LT' fun _ => True) ↔ ∃ u₀ ∈ (latO S T LS LT fun _ => True), u = actO S T g h u₀) → ∀ f, IsVacO ψ θ S T sS sT LS LT f → IsVacO ψ θ S T sS' sT' LS' LT' (om (g, h) f)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SsTransfer

theorem **ss_transfer** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (S Γ : Set (Matrix ι ι ℂ)), (S ⊆ Γ → Γ ⊆ (Submodule.span ℂ S : Set (Matrix ι ι ℂ)) → SS S → SS Γ) ∧ ∀ P : Matrix ι ι ℂ, IsUnit P.det → SS ((fun x => P ∗ x ∗ P⁻¹) '' S) → SS S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StandardHermitianFormHostFieldInhabitant

theorem **bar_bar** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

theorem **bar_rt** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : H.bar H.rt = -H.rt

theorem **bar_w** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : H.bar H.w0 = 1 - H.w0

theorem **w_norm** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ∃ m : ℕ, 2 ≤ m ∧ H.w0 ∗ H.bar H.w0 = (m : 𝒪 E)

theorem **w_not_unit** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ¬ IsUnit H.w0

theorem **w_ne_zero** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : H.w0 ≠ 0

theorem **moved_iff** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (P : Ideal (𝒪 E)) : Moved (IsCMField.complexConj E) P ↔ P.map (H.bar : 𝒪 E →+* 𝒪 E) ≠ P

theorem **splitOff_w** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : SplitOff (IsCMField.complexConj E) 1 H.w0

theorem **single_mem** (n : ℕ) (i : Fin n) : (Pi.single i (1 : E) : Fin n → E) ∈ stdLat E n

theorem **stdLat_eq_span** (n : ℕ) : stdLat E n = Submodule.span (𝒪 E) (Set.range fun i : Fin n => (Pi.single i (1 : E) : Fin n → E))

theorem **hermE_one** {F : Type} [Field F] [Algebra F E] (c : E ≃ₐ[F] E) {n : ℕ} (x y : Fin n → E) : hermE c 1 x y = ∑ a, x a ∗ c (y a)

theorem **isUnE_one_iff** {F : Type} [Field F] [Algebra F E] (c : E ≃ₐ[F] E) {n : ℕ} (γ : Matrix (Fin n) (Fin n) E) : IsUnE c 1 γ ↔ ∀ i j, ∑ b, c (γ b i) ∗ γ b j = if i = j then 1 else 0

theorem **main** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (n : ℕ), IsHermForm (IsCMField.complexConj E) (1 : Matrix (Fin n) (Fin n) E) ∧ Module.finrank (maximalRealSubfield E) E = 2 ∧ (stdLat E n).FG ∧ Submodule.span E (stdLat E n : Set (Fin n → E)) = ⊤ ∧ SplitOff (IsCMField.complexConj E) 1 H.w0 ∧ ¬ IsUnit H.w0 ∧ (∀ P : Ideal (𝒪 E),

P.IsPrime → (Moved (IsCMField.complexConj E) P ↔ P.map (H.bar : 𝒪 E →+* 𝒪 E) ≠ P)) ∧ ∀ γ : Matrix (Fin n) (Fin n) E, IsUnE (IsCMField.complexConj E) 1 γ ↔ ∀ i j, ∑ b, (IsCMField.complexConj E) (γ b i) * γ b j = if i = j then 1 else 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StandardHermitianFormHostFieldInhabitantStatement

theorem **_root_.GlobalTheory.T.standardHermitianForm_hostField_inhabitant** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (n : ℕ), IsHermForm (IsCMField.complexConj E) (1 : Matrix (Fin n) (Fin n) E) ∧ Module.finrank (maximalRealSubfield E) E = 2 ∧ (stdLat E n).FG ∧ Submodule.span E (stdLat E n : Set (Fin n → E)) = ⊤ ∧ SplitOff (IsCMField.complexConj E) 1 H.w0 ∧ ¬ IsUnit H.w0 ∧ (∀ P : Ideal (𝒪 E), P.IsPrime → (Moved (IsCMField.complexConj E) P ↔ P.map (H.bar : 𝒪 E →+* 𝒪 E) ≠ P)) ∧ ∀ γ : Matrix (Fin n) (Fin n) E, IsUnE (IsCMField.complexConj E) 1 γ ↔ ∀ i j, ∑ b, (IsCMField.complexConj E) (γ b i) * γ b j = if i = j then 1 else 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StarTraceHerm

theorem **star_trace_herm** {m n : ℕ} {A : Matrix (Fin n) (Fin n) ℂ} {B : Matrix (Fin m) (Fin m) ℂ} (hA : Aᴴ = A) (hB : Bᴴ = B) (x y : Matrix (Fin m) (Fin n) ℂ) : star (A * yᴴ * B * x).trace = (A * xᴴ * B * y).trace

theorem **trace_pos** {m n : ℕ} {A : Matrix (Fin n) (Fin n) ℂ} {B : Matrix (Fin m) (Fin m) ℂ} (hA : A.PosDef) (hB : B.PosDef) (x : Matrix (Fin m) (Fin n) ℂ) (hx : x ≠ 0) : 0 < (A * xᴴ * B * x).trace

theorem **adj_mul** (h : IsCartanInv G s) : sᴴ * G = G * s

theorem **adj_sq** (h : IsCartanInv G s) : sᴴ * sᴴ = 1

theorem **herm** (h : IsCartanInv G s) : Gᴴ = G

theorem **inv_mul_adj** (h : IsCartanInv G s) : G⁻¹ * sᴴ = (G * s)⁻¹

theorem **form_cplxO** {m n : ℕ} {GS sS : Matrix (Fin n) (Fin n) ℂ} {GT sT : Matrix (Fin m) (Fin m) ℂ} {ξ : ℂ} (hξ : ξ.re = 0) (hS : IsCartanInv GS sS) (hT : IsCartanInv GT sT) (x y : Matrix (Fin m) (Fin n) ℂ) : (ξ * (GS⁻¹ * (cplxO sS sT y)ᴴ * GT * x).trace).re = ξ.im * ((GS * sS)⁻¹ * yᴴ * (GT * sT) * x).trace.re

theorem **cplxO_pos** : ∀ (m n : ℕ) (GS sS : Matrix (Fin n) (Fin n) ℂ) (GT sT : Matrix (Fin m) (Fin m) ℂ) (ξ : ℂ), ξ.re = 0 → 0 < ξ.im → IsCartanInv GS sS → IsCartanInv GT sT → (∀ x : Matrix (Fin m) (Fin n) ℂ, (ξ * (GS⁻¹ * xᴴ * GT * x).trace).re = 0) ∧ (∀ x : Matrix (Fin m) (Fin n) ℂ, cplxO sS sT (cplxO sS sT x) = -x) ∧ (∀ x y : Matrix (Fin m) (Fin n) ℂ, (ξ * (GS⁻¹ * (cplxO sS sT y)ᴴ * GT * x).trace).re = (ξ * (GS⁻¹ * (cplxO sS sT x)ᴴ * GT * y).trace).re) ∧ ∀ x : Matrix (Fin m) (Fin n) ℂ, x ≠ 0 → 0 < (ξ * (GS⁻¹ * (cplxO sS sT x)ᴴ * GT * x).trace).re

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StdCharFinitePartRationalEqExpTrace

theorem **sum_cmType_eq_trace** {F E ι : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [Fintype ι] (h2 : Module.finrank F E = 2) (τ : ι → (E →+* ℂ)) (hi : Function.Injective τ) (hcm : ∀ σ, (∃ t, τ t = σ) ↔ ¬ ∃ t, τ t = ComplexEmbedding.conjugate σ) (x : F) (hx : ∀ σ : E →+* ℂ, ComplexEmbedding.conjugate σ (algebraMap F E x) = σ (algebraMap F E x)) : ∑ t, τ t (algebraMap F E x) = ((Algebra.trace ℚ F x : ℚ) : ℂ)

theorem **tauE_injective** (D : D7) : Function.Injective (tauE D)

theorem **tauE_cm** (D : D7) (σ : D.host.E →+* ℂ) : (∃ τ, tauE D τ = σ) ↔ ¬ ∃ τ, tauE D τ = ComplexEmbedding.conjugate σ

theorem **sum_tauE_eq_trace** (D : D7) (x : D.host.F) : ∑ t, tauE D t (algebraMap D.host.F D.host.E x) = ((Algebra.trace ℚ D.host.F x : ℚ) : ℂ)

theorem **char_fin_rat** {F : Type} [Field F] [NumberField F] (ψ : AddChar (AF F) ℂ) (h : IsCharAF F ψ) (x : F) : ψ (pairA F 0 (algebraMap F (Af F) x)) = psiInf F ψ (algebraMap F (Ainf F) (-x))

theorem **tmul_one_eq** {F E A : Type} [Field F] [Field E] [CommRing A] [Algebra F E] [Algebra F A] (x : F) : (algebraMap F A x) ⊗ₜ[F] (1 : E) = (1 : A) ⊗ₜ[F] (algebraMap F E x)

theorem **stdChar_fin_rat** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ x : D.host.F, ψ (pairA D.host.F 0 (algebraMap D.host.F (Af D.host.F) x)) = Complex.exp (2 * Real.pi * Complex.I * ((Algebra.trace ℚ D.host.F x : ℚ) : ℂ))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StdCharFinitePartRationalEqExpTraceStatement

theorem **_root_.GlobalTheory.T.stdChar_finitePart_rational_eq_exp_trace** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ x : D.host.F, ψ (pairA D.host.F 0 (algebraMap D.host.F (Af D.host.F) x)) = Complex.exp (2 * Real.pi * Complex.I * ((Algebra.trace ℚ D.host.F x : ℚ) : ℂ))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StdCharacterHalfSymplecticFormRealPoint

theorem **map_inv_field** {E : Type} [Field E] (τ : E →+* ℂ) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (h : G.det ≠ 0) : (G.map τ)⁻¹ = G⁻¹.map τ

theorem **ringHom_map_quadrupleSum_products** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) {m n : ℕ} (P : Fin n → Fin n → A) (y : Fin m → Fin n → A) (Q : Fin m → Fin m → A) (x : Fin m → Fin n → A) : φ (∑ a, ∑ b, ∑ c, ∑ d, P a b * y c b * Q c d * x d a) = ∑ a, ∑ b, ∑ c, ∑ d, φ (P a b) * φ (y c b) * φ (Q c d) * φ (x d a)

theorem **trace_four** {m n : ℕ} (P : Matrix (Fin n) (Fin n) ℂ) (y : Matrix (Fin m) (Fin n) ℂ) (Q : Matrix (Fin m) (Fin m) ℂ) (x : Matrix (Fin m) (Fin n) ℂ) : (P * yᴴ * Q * x).trace = ∑ a, ∑ b, ∑ c, ∑ d, P a b * starRingEnd ℂ (y c b) * Q c d * x d a

theorem **comp_hO** {θ : InfFam D} (hθ : IsInfFam θ) (S T : HSp D) (τ : D.what₂.Tau) (x y : WO S T) : θ τ (infPart D (hO S T x y)) = hAt S T τ (piW θ T S x τ) (piW θ T S y τ)

theorem **valTau_symplO** {θ : InfFam D} (hθ : IsInfFam θ) (S T : HSp D) (τ : D.what₂.Tau) (x y : WO S T) : valTau θ τ (symplO S T x y) = omAt S T τ (piW θ T S x τ) (piW θ T S y τ)

theorem **finPart_cA_eq_zero** {z : AE D} (hz : finPart D z = 0) : finPart D (cA D z) = 0

theorem **finPart_hO** (S T : HSp D) {x y : WO S T} (h : x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) : finPart D (hO S T x y) = 0

theorem **sndA_symplO** (S T : HSp D) {x y : WO S T} (h : x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) : sndA D.host.F (symplO S T x y) = 0

theorem **char_symplOH** : ∀ (D : D7) (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ), IsInfFam θ → IsStdChar D ψ (-1) → ∀ (S T : HSp D) (x y : WO S T), (x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) → ψ (symplOH S T x y) = Complex.exp (-(Real.pi * Complex.I) * (omO S T (piW θ T S x) (piW θ T S y) : ℂ))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StdCharacterHalfSymplecticFormRealPointStatement

theorem **_root_.GlobalTheory.T.stdCharacter_halfSymplecticForm_realPoint** : ∀ (D : D7) (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ), IsInfFam θ → IsStdChar D ψ (-1) → ∀ (S T : HSp D) (x y : WO S T), (x.map (finPart D) = 0 ∨ y.map (finPart D) = 0) → ψ (symplOH S T x y) = Complex.exp (-(Real.pi * Complex.I) * (omO S T (piW θ T S x) (piW θ T S y) : ℂ))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.StrongApproximation

theorem **mem_stab** {H : AddSubgroup (Fin n → R)} {A : Matrix (Fin n) (Fin n) R} : A ∈ stab H ↔ ∀ z ∈ H, A *ᵥ z ∈ H

theorem **exists_of_cong** {H : AddSubgroup (Fin n → R)} {D : ℕ} (hD : IsUnit (D : R)) {A : Matrix (Fin n) (Fin n) R} (h : ∀ z ∈ H, ∃ w ∈ H, A *ᵥ z = D • w) : ∃ B ∈ stab H, A = (D : R) • B

theorem **natCast_smul_mem** {H : AddSubgroup (Fin n → R)} (k : ℕ) {A : Matrix (Fin n) (Fin n) R} (h : A ∈ stab H) : (k : R) • A ∈ stab H

theorem **mem_of_coprime** {H : AddSubgroup (Fin n → R)} {k l : ℕ} (hkl : k.Coprime l) {A : Matrix (Fin n) (Fin n) R} (hk : (k : R) • A ∈ stab H) (hl : (l : R) • A ∈ stab H) : A ∈ stab H

theorem **mem_scal** {H : AddSubgroup (Fin n → R)} {a : R} : a ∈ scal H ↔ a • (1 : Matrix (Fin n) (Fin n) R) ∈ stab H

theorem **exists_entry_bound** {H : AddSubgroup (Fin n → R)} (hb : ∀ A : Matrix (Fin n) (Fin n) R, ∃ d : ℕ, 0 < d ∧ (d : R) • A ∈ stab H) : ∃ d : ℕ, 0 < d ∧ ∀ B ∈ stab H, ∀ i j, (d : R) * B i j ∈ scal H

theorem **det_cong** (S : Subring R) (t D : ℕ) (B : Matrix (Fin n) (Fin n) R) (hB : ∀ i j, B i j ∈ S) : ∃ s ∈ S, ((t : R) • (1 : Matrix (Fin n) (Fin n) R) + (D : R) • B).det = (t : R) ^ n + D * s

theorem **glue** {H : AddSubgroup (Fin n → R)} {M d t k e : ℕ} (hM : IsUnit (M : R)) (ht : t.Coprime (M * d)) (hk : k.Coprime (M * d)) {g m m' C W : Matrix (Fin n) (Fin n) R} (hg : (d : R) • g ∈ stab H) (hC : C ∈ stab H) (hW : W ∈ stab H) (hm' : (((M * d) ^ e : ℕ) : R) • m' ∈ stab H) (h1 : (t : R) • (m * g - 1) = (M : R) • C) (h2 : (k : R) • (m' - m) = ((M * d : ℕ) : R) • W) : ∃ Y ∈ stab H, m' * g = 1 + (M : R) • Y

theorem **mul_cong** {H : AddSubgroup (Fin n → R)} {M t t' : ℕ} {A A' C C' : Matrix (Fin n) (Fin n) R} (hC : C ∈ stab H) (hC' : C' ∈ stab H) (h : (t : R) • (A - 1) = (M : R) • C) (h' : (t' : R) • (A' - 1) = (M : R) • C') : ∃ B ∈ stab H, ((t * t' : ℕ) : R) • (A * A' - 1) = (M : R) • B

theorem **emb_mulVec** (γ : Matrix (Fin n) (Fin n) E) (v : Fin n → E) : (emb F).mapMatrix γ *ᵥ fv F v = fv F (γ *ᵥ v)

theorem **isUnit_nat** {D : ℕ} (hD : 0 < D) : IsUnit (D : RA F E)

theorem **emb_smul** (a : E) (γ : Matrix (Fin n) (Fin n) E) : (emb F).mapMatrix (a • γ) = emb F a • (emb F).mapMatrix γ

theorem **emb_mem_stab** {L : Submodule (𝒪 E) (Fin n → E)} {γ : Matrix (Fin n) (Fin n) E} (h : ∀ v ∈ L, γ *ᵥ v ∈ L) : (emb F).mapMatrix γ ∈ stab (hat F (L : Set (Fin n → E)))

theorem **conj_mem_scal** (c : E ≃ₐ[F] E) {L : Submodule (𝒪 E) (Fin n → E)} (hL : L.FG) (hsp : Submodule.span E (L : Set (Fin n → E)) = ⊤) (i : Fin n) {s : E} (hs : emb F s ∈ scal (hat F (L : Set (Fin n → E)))) : emb F (c s) ∈ scal (hat F (L : Set (Fin n → E)))

theorem **exists_bound_stab** {L : Submodule (𝒪 E) (Fin n → E)} (hL : L.FG) (hsp : Submodule.span E (L : Set (Fin n → E)) = ⊤) (A : Matrix (Fin n) (Fin n) (RA F E)) : ∃ d : ℕ, 0 < d ∧ (d : RA F E) • A ∈ stab (hat F (L : Set (Fin n → E)))

theorem **strong_approximation** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), Module.finrank F E = 2 → IsHerm (c : E →+* E) G → (∃ z : Fin n → E, z ≠ 0 ∧ hf (c : E →+* E) G z z = 0) → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ g : GL (Fin n) (RA F E), (∀ x y, hermA c G ((g : Matrix (Fin n) (Fin n) (RA F E)) *ᵥ x) ((g : Matrix (Fin n) (Fin n) (RA F E)) *ᵥ y) = hermA c G x y) → (g : Matrix (Fin n) (Fin n) (RA F E)).det = 1 → ∀ M : ℕ, 0 < M → ∃ γ : Matrix (Fin n) (Fin n) E, IsU (c : E →+* E) G γ ∧ γ.det = 1 ∧ ∀ z ∈ hat F (L : Set (Fin n → E)), ∃ w ∈ hat F (L : Set (Fin n → E)), ((γ.map (fun a => ((1 : FiniteAdeleRing (𝒪 F) F) ⊗ₜ[F] a : RA F E))) * (g : Matrix (Fin n) (Fin n) (RA F E))) *ᵥ z = z + M • w

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SubgroupPowImageUnitsCongruentOneModOfIsAdicComp

theorem **deep_smod_iff** {O : Type} [CommRing O] (π a b : O) (n : ℕ) : a ≡ b [SMOD (Ideal.span {π} ^ n • ⊤ : Submodule O O)] ↔ π ^ n ∣ a - b

theorem **deep_exists_solution** {O : Type} [CommRing O] (π : O) [IsAdicComplete (Ideal.span {π}) O] (h : O[X]) (y : O) : ∃ x : O, x + π * h.eval x = y

theorem **deep_binomial** {O : Type} [CommRing O] (t : O) (n : ℕ) : ∃ g : O[X], ∀ x : O, (1 + t * x) ^ n = 1 + n * t * x + t ^ 2 * x ^ 2 * g.eval x

theorem **deep_mem_oneUnits** {O : Type} [CommRing O] (t : O) (u : Oˣ) : u ∈ oneUnits t ↔ ∃ x : O, (u : O) = 1 + t * x

theorem **deep_isUnit** {O : Type} [CommRing O] (π : O) [IsAdicComplete (Ideal.span {π}) O] (x : O) : IsUnit (1 + π * x)

theorem **deep_units_pows** : ∀ (p : ℕ) (O : Type) [CommRing O] [IsDomain O] [CharZero O] (π : O), 0 < p → π ≠ 0 → IsAdicComplete (Ideal.span {π}) O → pows p (oneUnits ((p : O) * π)) = oneUnits ((p : O) ^ 2 * π) ∧ pTor p (oneUnits ((p : O) * π)) = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SubgroupPowImageUnitsCongruentOneModOfIsAdicCompStatement

theorem **_root_.GlobalTheory.T.subgroupPowImage_unitsCongruentOneMod_of_isAdicComplete** : ∀ (p : ℕ) (O : Type) [CommRing O] [IsDomain O] [CharZero O] (π : O), 0 < p → π ≠ 0 → IsAdicComplete (Ideal.span {π}) O → pows p (oneUnits ((p : O) * π)) = oneUnits ((p : O) ^ 2 * π) ∧ pTor p (oneUnits ((p : O) * π)) = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SubgroupPowRelIndexTopOfGroupFg

private theorem **card_quot_range_nsmul_eq_pow_finrank_mul_card_ker** (M : Type) [AddCommGroup M] [AddGroup.FG M] (p : ℕ) (hp : 0 < p) : Nat.card (nsmulAddMonoidHom (α

theorem **_root_.GlobalTheory.CNorm.pIdx_top_of_fg** : ∀ (p : ℕ) (G : Type) [CommGroup G] [Group.FG G], 0 < p → _root_.GlobalTheory.CNorm.pTor p (⊤ : Subgroup G) ≠ 0 ∧ _root_.GlobalTheory.CNorm.pIdx p (⊤ : Subgroup G) = p ^ Module.finrank ℤ (Additive G) * _root_.GlobalTheory.CNorm.pTor p (⊤ : Subgroup G)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SubgroupPowRelIndexUnitsOfIsAdicComplete

theorem **pIdx_eq_pTor_of_finite_group** (p : ℕ) {G : Type} [CommGroup G] [Finite G] (U : Subgroup G) : pTor p U ≠ 0 ∧ pIdx p U = pTor p U

theorem **pIdx_units_of_isAdicComplete** : ∀ (p : ℕ) (O : Type) [CommRing O] [IsDomain O] [CharZero O] (π : O), 0 < p → π ≠ 0 → IsAdicComplete (Ideal.span {π}) O → Nat.card (O ⧸ Ideal.span {(p : O) * π}) ≠ 0 → pTor p (⊤ : Subgroup Oˣ) ≠ 0 ∧ pIdx p (⊤ : Subgroup Oˣ) = Nat.card (O ⧸ Ideal.span {(p : O)}) * pTor p (⊤ : Subgroup Oˣ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SubgroupPowRelIndexUnitsOfIsAdicCompleteStatement

theorem **_root_.GlobalTheory.T.subgroupPowRelIndex_units_of_isAdicComplete** : ∀ (p : ℕ) (O : Type) [CommRing O] [IsDomain O] [CharZero O] (π : O), 0 < p → π ≠ 0 → IsAdicComplete (Ideal.span {π}) O → Nat.card (O ⧸ Ideal.span {(p : O) * π}) ≠ 0 → pTor p (⊤ : Subgroup Oˣ) ≠ 0 ∧ pIdx p (⊤ : Subgroup Oˣ) = Nat.card (O ⧸ Ideal.span {(p : O)}) * pTor p (⊤ : Subgroup Oˣ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SumHermitianLogGramEqZeroOfEnergyMinimum

lemma **log_herm** {H A : Matrix ι ι ℂ} (h : IsLog H A) : Hᴴ = H

lemma **log_comm** {H A : Matrix ι ι ℂ} (h : IsLog H A) : A * H = H * A

lemma **exists_path** (X : Matrix ι ι ℂ) (hX : Xᴴ = X) : ∃ P Q : ℝ → Matrix ι ι ℂ, P 0 = 1 ∧ Q 0 = 1 ∧ (∀ t, P t * Q t = 1) ∧ (∀ t, (P t)ᴴ = P t) ∧ (∀ t, (Q t)ᴴ = Q t) ∧ HasDerivAt P X 0 ∧ HasDerivAt Q (-X) 0

lemma **deriv_entry** {Y : ℝ → Matrix ι ι ℂ} {D : Matrix ι ι ℂ} (h : HasDerivAt Y D 0) (i j : ι) : HasDerivAt (fun t => Y t i j) (D i j) 0

lemma **deriv_phi** {Y : ℝ → Matrix ι ι ℂ} {D : Matrix ι ι ℂ} (h : HasDerivAt Y D 0) (h0 : Y 0 = 0) (M : Matrix ι ι ℂ) (C : ℝ) : HasDerivAt (fun t => 2 * (Matrix.trace (M * Y t)).re + C * ∑ i, ∑ j, ‖Y t i j‖ ^ 2) (2 * (Matrix.trace (M * D)).re) 0

lemma **trace_id** {g g' H H' X : Matrix ι ι ℂ} (hg : g' * g = 1) (hc : (gᴴ * g) * H = H * (gᴴ * g)) (hi : g * H = -(H' * g)) : Matrix.trace (H * (gᴴ * g)⁻¹ * (-X * (gᴴ * g) + gᴴ * X * g + gᴴ * X * g - (gᴴ * g) * X)) = -2 * Matrix.trace (H * X) - 2 * Matrix.trace (H' * X)

theorem **sum_log_eq_zero** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (σ : Type) [Fintype σ] (τ : σ ≃ σ) (g H : σ → Matrix ι ι ℂ), (∀ s, g (τ s) * g s = 1) → (∀ s, IsLog (H s) ((g s)ᴴ * g s)) → (∀ (P' : Matrix ι ι ℂ) (H' : σ → Matrix ι ι ℂ), IsUnit P'.det → (∀ s, IsLog (H' s) ((P' * g s * P'⁻¹)ᴴ * (P' * g s * P'⁻¹))) → ∑ s, (Matrix.trace (H s * H s)).re ≤ ∑ s, (Matrix.trace (H' s * H' s)).re) → ∑ s, H s = 0

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SummableGaussianCoset

theorem **exp_le_inv_pow** {c : ℝ} (hc : 0 < c) (R : ℝ) (n : ℕ) {t : ℝ} (ht : 0 < t) : Real.exp (-c * t ^ 2 + R * t) ≤ ((n.factorial : ℝ) * Real.exp ((R + 1) ^ 2 / (4 * c))) * t⁻¹ ^ n

theorem **summable_lattice** {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (L : Submodule ℤ V) [DiscreteTopology L] {c : ℝ} (hc : 0 < c) (R : ℝ) (x : V) : Summable fun z : L => Real.exp (-c * ‖(z : V) - x‖ ^ 2 + R * ‖(z : V) - x‖)

theorem **discrete_of_finite** {V : Type} [NormedAddCommGroup V] (L : AddSubgroup V) (h : {v : V | v ∈ L ∧ ‖v‖ ≤ 1}.Finite) : DiscreteTopology L

theorem **summable_gaussian_coset** : ∀ (G V : Type) [AddCommGroup G] [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (p : G →+ V) (Γ U : AddSubgroup G), (∀ g ∈ Γ, p g = 0 → g = 0) → (∀ R : ℝ, {g : G | g ∈ Γ ∧ g ∈ U ∧ ‖p g‖ ≤ R}.Finite) → ∀ (c : ℝ), 0 < c → ∀ (R : ℝ) (w : G) (y : V), Summable fun g : Γ => Set.indicator (U : Set G) (fun _ => Real.exp (-c * ‖p g + y‖ ^ 2 + R * ‖p g + y‖)) ((g : G) + w)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SymplecticAdelicFormBiadditiveAlternatingRationa

theorem **trace_one_tmul** {F E : Type} [Field F] [Field E] [Algebra F E] [Module.Finite F E] (R : Type) [CommRing R] [Algebra F R] (e : E) : Algebra.trace R (R ⊗[F] E) ((1 : R) ⊗ₜ[F] e) = algebraMap F R (Algebra.trace F E e)

theorem **eq_zero_of_eq_neg** {F R : Type} [Field F] [CharZero F] [CommRing R] [Algebra F R] {a : R} (h : a = -a) : a = 0

theorem **map_sesq** {E A : Type} [CommRing E] [CommRing A] (ι : E →+* A) (c : A →+* A) (s : E → E) (hc : ∀ e, c (ι e) = ι (s e)) {m n : ℕ} (ξ : E) (P : Fin m → Fin m → E) (Q : Fin n → Fin n → E) (x y : Fin n → Fin m → E) : ι ξ * ∑ a, ∑ b, ∑ c', ∑ d, ι (P a b) * c (ι (y c' b)) * ι (Q c' d) * ι (x d a) = ι (ξ * ∑ a, ∑ b, ∑ c', ∑ d, P a b * s (y c' b) * Q c' d * x d a)

theorem **symplO_rat** (D : D7) (S T : HSp D) (γ γ' : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : ∃ a : D.host.F, symplO S T (ratW T S γ) (ratW T S γ') = algebraMap D.host.F (AF D.host.F) a

theorem **symplO_alt** : ∀ (D : D7) (S T : HSp D), (∀ x x' y : WO S T, symplO S T (x + x') y = symplO S T x y + symplO S T x' y) ∧ (∀ x y y' : WO S T, symplO S T x (y + y') = symplO S T x y + symplO S T x y') ∧ (∀ x y : WO S T, symplO S T x y = -symplO S T y x) ∧ (∀ x : WO S T, symplO S T x x = 0) ∧ ∀ γ γ' : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ a : D.host.F, symplO S T (ratW T S γ) (ratW T S γ') = algebraMap D.host.F (AF D.host.F) a

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.SymplecticAdelicFormBiadditiveAlternatingRationaStatement

theorem **_root_.GlobalTheory.T.sympLecticAdelicForm_biadditive_alternating_rational** : ∀ (D : D7) (S T : HSp D), (∀ x x' y : WO S T, symplO S T (x + x') y = symplO S T x y + symplO S T x' y) ∧ (∀ x y y' : WO S T, symplO S T x (y + y') = symplO S T x y + symplO S T x y') ∧ (∀ x y : WO S T, symplO S T x y = -symplO S T y x) ∧ (∀ x : WO S T, symplO S T x x = 0) ∧ ∀ γ γ' : Matrix (Fin T.rank) (Fin S.rank) D.host.E, ∃ a : D.host.F, symplO S T (ratW T S γ) (ratW T S γ') = algebraMap D.host.F (AF D.host.F) a

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TensorActionHermitianExpFixedOfMinNorm

theorem **_root_.GlobalTheory.ClassFin.act_exp_eq_of_isMinVec** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (U : Matrix ι ι ℂ) (δ : ι → ℝ), _root_.GlobalTheory.ClassFin.IsMinVec d v → U * Uᴴ = 1 → ∑ i, δ i = 0 → _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v (U * diagonal (fun i => (Real.exp (δ i) : ℂ)) * Uᴴ)) = _root_.GlobalTheory.ClassFin.nsq v → ∀ s : ℝ, _root_.GlobalTheory.ClassFin.act d v (U * diagonal (fun i => (Real.exp (s * δ i) : ℂ)) * Uᴴ) = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TensorExistsOfGaloisFixed

theorem **tensor_exists_of_galois_fixed** : ∀ (F M E K : Type) [Field F] [CharZero F] [Field M] [Field E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [FiniteDimensional M E] [IsGalois M E] [CommRing K] [Algebra F K] (y : K ⊗[F] E), (∀ h : E ≃ₐ[M] E, Algebra.TensorProduct.map (AlgHom.id K K) ((h.restrictScalars F : E ≃ₐ[F] E) : E →ₐ[F] E) y = y) → ∃ z : K ⊗[F] M, Algebra.TensorProduct.map (AlgHom.id K K) (IsScalarTower.toAlgHom F M E) z = y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TensorExistsOfGaloisFixedStatement

theorem **_root_.GlobalTheory.T.exists_tensor_eq_of_fixed_by_algEquiv** : ∀ (F M E K : Type) [Field F] [CharZero F] [Field M] [Field E] [Algebra F M] [Algebra M E] [Algebra F E] [IsScalarTower F M E] [FiniteDimensional M E] [IsGalois M E] [CommRing K] [Algebra F K] (y : K ⊗[F] E), (∀ h : E ≃ₐ[M] E, Algebra.TensorProduct.map (AlgHom.id K K) ((h.restrictScalars F : E ≃ₐ[F] E) : E →ₐ[F] E) y = y) → ∃ z : K ⊗[F] M, Algebra.TensorProduct.map (AlgHom.id K K) (IsScalarTower.toAlgHom F M E) z = y

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TensorPowerSumRightActionNormAndContinuity

private lemma **tpow_apply_eq_prod** {ι : Type} (n : ℕ) (y : Matrix ι ι ℂ) (a b : Fin n → ι) : _root_.GlobalTheory.ClassFin.tpow n y a b = ∏ k, y (a k) (b k)

private lemma **tpow_conjTranspose** {ι : Type} (n : ℕ) (y : Matrix ι ι ℂ) : (_root_.GlobalTheory.ClassFin.tpow n y)ᴴ = _root_.GlobalTheory.ClassFin.tpow n yᴴ

private lemma **nsq_eq_sum_sum_norm_sq_components** {ι : Type} [Fintype ι] {m : ℕ} (d : Fin m → ℕ) (x : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) : _root_.GlobalTheory.ClassFin.nsq x = ∑ i, ∑ a : Fin (d i) → ι, ‖x (i, a)‖ ^ 2

private lemma **act_apply_eq_vecMul_tpow** {ι : Type} [Fintype ι] {m : ℕ} (d : Fin m → ℕ) (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (y : Matrix ι ι ℂ) (i : Fin m) (a : Fin (d i) → ι) : _root_.GlobalTheory.ClassFin.act d v y (i, a) = ((fun b => v (i, b)) ᵥ* _root_.GlobalTheory.ClassFin.tpow (d i) y) a

private lemma **sum_norm_sq_vecMul_eq_of_mul_conjTranspose_eq_one** {κ : Type} [Fintype κ] [DecidableEq κ] (w : κ → ℂ) (M : Matrix κ κ ℂ) (hM : M * Mᴴ = 1) : ∑ c, ‖(w ᵥ* M) c‖ ^ 2 = ∑ a, ‖w a‖ ^ 2

private lemma **sum_norm_sq_vecMul_le_card_sq_mul_bound_sq** {κ : Type} [Fintype κ] (w : κ → ℂ) (M : Matrix κ κ ℂ) (K : ℝ) (hK : ∀ a c, ‖M a c‖ ≤ K) : ∑ c, ‖(w ᵥ* M) c‖ ^ 2 ≤ (Fintype.card κ : ℝ) ^ 2 * K ^ 2 * ∑ a, ‖w a‖ ^ 2

private lemma **norm_tpow_apply_le_pow_of_forall_norm_le** {ι : Type} (n : ℕ) (y : Matrix ι ι ℂ) (B : ℝ) (hy : ∀ i j, ‖y i j‖ ≤ B) (a b : Fin n → ι) : ‖_root_.GlobalTheory.ClassFin.tpow n y a b‖ ≤ B ^ n

theorem **_root_.GlobalTheory.ClassFin.ActNorm.act_norm_continuity** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ), (∀ (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (U : Matrix ι ι ℂ), U * Uᴴ = 1 → _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v U) = _root_.GlobalTheory.ClassFin.nsq v) ∧ (∀ C : ℝ, ∃ C' : ℝ, 0 ≤ C' ∧ ∀ (v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ) (y : Matrix ι ι ℂ), (∀ i j, ‖y i j‖ ≤ C) → _root_.GlobalTheory.ClassFin.nsq (_root_.GlobalTheory.ClassFin.act d v y) ≤ C' * _root_.GlobalTheory.ClassFin.nsq v) ∧ ∀ v : _root_.GlobalTheory.ClassFin.TIdx ι d → ℂ, Continuous (_root_.GlobalTheory.ClassFin.act d v)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TensorSLOrbitMapProperOfIsClosed

lemma **tpow_mul'** (n : ℕ) (y z : Matrix ι ι ℂ) : tpow n (y * z) = tpow n y * tpow n z

lemma **tpow_one'** (n : ℕ) : tpow n (1 : Matrix ι ι ℂ) = 1

lemma **act_mul'** (t : TIdx ι d → ℂ) (y z : Matrix ι ι ℂ) : act d t (y * z) = act d (act d t y) z

lemma **act_one'** (t : TIdx ι d → ℂ) : act d t 1 = t

lemma **continuous_act_right** (t : TIdx ι d → ℂ) : Continuous (act d t)

lemma **continuous_act_left** (y : Matrix ι ι ℂ) : Continuous (fun t : TIdx ι d → ℂ => act d t y)

lemma **isCompact_box** (N : ℝ) : IsCompact (box ι N)

lemma **box_mono** {N M : ℝ} (h : N ≤ M) : box ι N ⊆ box ι M

lemma **box_mul** (N : ℝ) (g : Matrix ι ι ℂ) (hg : g.det = 1) : ∃ M : ℝ, ∀ a ∈ box ι N, a * g ∈ box ι M

lemma **det_adj** {y : Matrix ι ι ℂ} (hy : y.det = 1) : y.adjugate.det = 1

lemma **adj_mul** {y : Matrix ι ι ℂ} (hy : y.det = 1) : y.adjugate * y = 1

lemma **mul_adj** {y : Matrix ι ι ℂ} (hy : y.det = 1) : y * y.adjugate = 1

lemma **exists_interior** (v : TIdx ι d → ℂ) (hX : IsClosed (orb d v)) : ∃ (n : ℝ) (O : Set (TIdx ι d → ℂ)) (y₀ : Matrix ι ι ℂ), IsOpen O ∧ y₀.det = 1 ∧ act d v y₀ ∈ O ∧ ∀ w ∈ O, w ∈ orb d v → w ∈ act d v '' box ι n

lemma **locally_box** (v : TIdx ι d → ℂ) (hX : IsClosed (orb d v)) (w : TIdx ι d → ℂ) (hw : w ∈ orb d v) : ∃ O : Set (TIdx ι d → ℂ), IsOpen O ∧ w ∈ O ∧ ∃ N : ℝ, O ∩ orb d v ⊆ act d v '' box ι N

lemma **isCompact_nsq** {κ : Type} [Fintype κ] (C : ℝ) : IsCompact {w : κ → ℂ | nsq w ≤ C}

theorem **GlobalTheory.ClassFin.orbit_proper_of_isClosed** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ), IsClosed (orb d v) → ∀ C : ℝ, ∃ C' : ℝ, ∀ y : Matrix ι ι ℂ, y.det = 1 → nsq (act d v y) ≤ C → ∃ h : Matrix ι ι ℂ, h.det = 1 ∧ act d v h = v ∧ ∀ i j, ‖(h * y) i j‖ ≤ C'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TensorSLOrbitMapProperOfIsClosedStatement

theorem **_root_.GlobalTheory.T.tensorSLOrbit_map_proper_of_isClosed** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (m : ℕ) (d : Fin m → ℕ) (v : TIdx ι d → ℂ), IsClosed (orb d v) → ∀ C : ℝ, ∃ C' : ℝ, ∀ y : Matrix ι ι ℂ, y.det = 1 → nsq (act d v y) ≤ C → ∃ h : Matrix ι ι ℂ, h.det = 1 ∧ act d v h = v ∧ ∀ i j, ‖(h * y) i j‖ ≤ C'

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ThetaAtEqSumHeisOfFourierSum

theorem **isSBHalf_halfInd** {D : D7} {m n : ℕ} (e : AE D) (Λ : AddSubgroup (Mx D m n)) : IsSBHalf e Λ (halfInd e Λ)

theorem **thetaAt_eq_sum_heis_of_fourier_sum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (e : AE D) (Λ : AddSubgroup (WO S T)) (f₀ Φ f : WO S T → ℂ) (Y : Finset (WO S T)) (a : WO S T → ℂ), IsThetaAt ψ (symplO S T) (heisO ψ S T) e Λ 1 f₀ Φ f → (∀ u ∈ Λ, u.map (fun z => e * z) ∈ Λ) → (∀ y ∈ Y, IsHalf (cA D e) y) → (∀ t, Φ t = halfInd e Λ t * ∑ y ∈ Y, a y * ψ (symplO S T t y)) → f = ∑ y ∈ Y, a y • heisO ψ S T y f₀

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ThetaNormSqSumHeisenbergTranslatesDistinctChars

theorem **setIntegral_add_right** {G E : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [MeasurableSpace G] [BorelSpace G] [NormedAddCommGroup E] [NormedSpace ℝ E] (Γ : AddSubgroup G) [Countable Γ] (μ : Measure G) [μ.IsAddHaarMeasure] (𝓕 : Set G) (h𝓕 : IsAddFundamentalDomain Γ 𝓕 μ) (h : G → E) (hp : ∀ γ ∈ Γ, ∀ w, h (γ + w) = h w) (u : G) : ∫ w in 𝓕, h (w + u) ∂μ = ∫ w in 𝓕, h w ∂μ

theorem **integral_norm_sq_sum** {X ι : Type} [MeasurableSpace X] (ν : Measure X) [IsFiniteMeasure ν] (s : Finset ι) (g : ι → X → ℂ) (c : ι → ℂ) (C : ℝ) (hm : ∀ i, Measurable (g i)) (hC : ∀ i x, ‖g i x‖ ≤ C) (hc : ∀ i ∈ s, ‖c i‖ = 1) (h1 : ∀ i ∈ s, ∫ x, ‖g i x‖ ^ 2 ∂ν = 1) (h0 : ∀ i ∈ s, ∀ j ∈ s, i ≠ j → ∫ x, g i x * (starRingEnd ℂ) (g j x) ∂ν = 0) : ∫ x, ‖(∑ i ∈ s, c i • g i) x‖ ^ 2 ∂ν = (s.card : ℝ)

theorem **heisO_quasi** {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (u : WO S T) (γ : WO S T) (hγ : γ ∈ ratL T S) (w : WO S T) : heisO ψ S T u f (γ + w) = ψ (-(symplOH S T γ w)) * heisO ψ S T u f w

theorem **continuous_heisO** (hψ : IsCharAF D.host.F ψ) {f : WO S T → ℂ} (hf : Continuous f) (u : WO S T) : Continuous (heisO ψ S T u f)

theorem **norm_heisO** (hψ : IsCharAF D.host.F ψ) (f : WO S T → ℂ) (u w : WO S T) : ‖heisO ψ S T u f w‖ = ‖f (w + u)‖

theorem **mul_conj_of_norm_one** {z : ℂ} (h : ‖z‖ = 1) : z * (starRingEnd ℂ) z = 1

theorem **thetaNormSq_sum_heisenbergTranslates_distinctChars** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (hψ : IsCharAF D.host.F ψ) (μ : Measure (WO S T)) (ℱ : Set (WO S T)) (hμ : IsCovolOne (ratL T S) μ ℱ) (f : WO S T → ℂ) (hf : IsThetaFunO ψ S T f) (hn : nrm2 μ ℱ f = 1) (ι : Type) (s : Finset ι) (a : ι → WO S T) (c : ι → ℂ) (hc : ∀ i ∈ s, ‖c i‖ = 1) (hl : ∀ i ∈ s, ∀ j ∈ s, i ≠ j → ∃ (l : WO S T) (ε : ℂ), ‖ε‖ = 1 ∧ heisO ψ S T l f = ε • f ∧ ψ (symplO S T (a i - a j) l) ≠ 1) : nrm2 μ ℱ (∑ i ∈ s, c i • heisO ψ S T (a i) f) = (s.card : ℝ)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TraceLogLeLogDet

theorem **jensen_exp** {ι : Type} [Fintype ι] (w d : ι → ℝ) (hw : ∀ j, 0 ≤ w j) (h1 : ∑ j, w j = 1) : 0 < ∑ j, w j * Real.exp (d j) ∧ ∑ j, w j * d j ≤ Real.log (∑ j, w j * Real.exp (d j)) ∧ (∑ j, w j * d j = Real.log (∑ j, w j * Real.exp (d j)) → ∀ j, w j ≠ 0 → Real.exp (d j) = ∑ j, w j * Real.exp (d j))

theorem **compress_diag** {ι κ : Type} [Fintype ι] [DecidableEq ι] (C : Matrix ι κ ℂ) (f : ι → ℝ) (k : κ) : (Cᴴ * diagonal (fun i => (f i : ℂ)) * C) k k = ((∑ j, Complex.normSq (C j k) * f j : ℝ) : ℂ)

theorem **exists_diag** {κ : Type} [Fintype κ] [DecidableEq κ] (B : Matrix κ κ ℂ) (hB : B.IsHermitian) : ∃ (V : Matrix κ κ ℂ) (m : κ → ℂ), V * Vᴴ = 1 ∧ Vᴴ * V = 1 ∧ Vᴴ * B * V = diagonal m

theorem **trace_log_le_log_det** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (κ : Type) [Fintype κ] [DecidableEq κ] (H A : Matrix ι ι ℂ) (Q : Matrix ι κ ℂ), IsLog H A → Qᴴ * Q = 1 → (Matrix.trace (Qᴴ * H * Q)).re ≤ Real.log (Qᴴ * A * Q).det.re ∧ ((Matrix.trace (Qᴴ * H * Q)).re = Real.log (Qᴴ * A * Q).det.re → ∃ N : Matrix κ κ ℂ, A * Q = Q * N)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TransvectionIdentitiesUnitary

theorem **hf_add_left** (x x' y : Fin n → R) : hf σ G (x + x') y = hf σ G x y + hf σ G x' y

theorem **hf_add_right** (x y y' : Fin n → R) : hf σ G x (y + y') = hf σ G x y + hf σ G x y'

theorem **hf_smul_left** (c : R) (x y : Fin n → R) : hf σ G (c • x) y = c * hf σ G x y

theorem **hf_smul_right** (c : R) (x y : Fin n → R) : hf σ G x (c • y) = σ c * hf σ G x y

theorem **hf_conj** (hσ : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) (x y : Fin n → R) : σ (hf σ G x y) = hf σ G y x

theorem **tv_mulVec** (x : Fin n → R) (t : R) (v : Fin n → R) : tv σ G x t *ᵥ v = v + (t * hf σ G v x) • x

theorem **ext_mulVec** {A B : Matrix (Fin n) (Fin n) R} (h : ∀ v, A *ᵥ v = B *ᵥ v) : A = B

theorem **tv_mul** (x : Fin n → R) (t t' : R) : tv σ G x t * tv σ G x t' = tv σ G x (t + t' + t * t' * hf σ G x x)

theorem **tv_det** (x : Fin n → R) (t : R) : (tv σ G x t).det = 1 + t * hf σ G x x

theorem **isU_tv** (hσ : ∀ a, σ (σ a) = a) (hG : ∀ a b, σ (G a b) = G b a) (x : Fin n → R) (t : R) (h : t + σ t + t * σ t * hf σ G x x = 0) : IsU σ G (tv σ G x t)

theorem **conj_tv** (γ : Matrix (Fin n) (Fin n) R) (x : Fin n → R) (t : R) (h : IsU σ G γ) : γ * tv σ G x t = tv σ G (γ *ᵥ x) t * γ

theorem **tv_smul** (a : R) (x : Fin n → R) (t : R) : tv σ G (a • x) t = tv σ G x (t * a * σ a)

theorem **isU_mul** (γ γ' : Matrix (Fin n) (Fin n) R) (h : IsU σ G γ) (h' : IsU σ G γ') : IsU σ G (γ * γ')

theorem **calculus** : ∀ (R : Type) [CommRing R] (n : ℕ) (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R), (∀ a, σ (σ a) = a) → (∀ a b, σ (G a b) = G b a) → (∀ x y, σ (hf σ G x y) = hf σ G y x) ∧ (∀ x t v, tv σ G x t *ᵥ v = v + (t * hf σ G v x) • x) ∧ (∀ x t t', tv σ G x t * tv σ G x t' = tv σ G x (t + t' + t * t' * hf σ G x x)) ∧ (∀ x t, (tv σ G x t).det = 1 + t * hf σ G x x) ∧ (∀ x t, t + σ t + t * σ t * hf σ G x x = 0 → IsU σ G (tv σ G x t)) ∧ (∀ γ x t, IsU σ G γ → γ * tv σ G x t = tv σ G (γ *ᵥ x) t * γ) ∧ (∀ (a : R) x t, tv σ G (a • x) t = tv σ G x (t * a * σ a)) ∧ (∀ γ γ', IsU σ G γ → IsU σ G γ' → IsU σ G (γ * γ'))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TransvectionIdentitiesUnitaryStatement

theorem **_root_.GlobalTheory.T.transvection_identities_unitary** : ∀ (R : Type) [CommRing R] (n : ℕ) (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R), (∀ a, σ (σ a) = a) → (∀ a b, σ (G a b) = G b a) → (∀ x y, σ (hf σ G x y) = hf σ G y x) ∧ (∀ x t v, tv σ G x t *ᵥ v = v + (t * hf σ G v x) • x) ∧ (∀ x t t', tv σ G x t * tv σ G x t' = tv σ G x (t + t' + t * t' * hf σ G x x)) ∧ (∀ x t, (tv σ G x t).det = 1 + t * hf σ G x x) ∧ (∀ x t, t + σ t + t * σ t * hf σ G x x = 0 → IsU σ G (tv σ G x t)) ∧ (∀ γ x t, IsU σ G γ → γ * tv σ G x t = tv σ G (γ *ᵥ x) t * γ) ∧ (∀ (a : R) x t, tv σ G (a • x) t = tv σ G x (t * a * σ a)) ∧ (∀ γ γ', IsU σ G γ → IsU σ G γ' → IsU σ G (γ * γ'))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TsumIdealEqSumClassGroup

theorem **_root_.GlobalTheory.existsUnique_isPrincipal_mul_classRep** {K : Type} [Field K] [NumberField K] (b : ClassGroup (𝒪 K) → (Ideal (𝒪 K))⁰) (hb : Function.Bijective (fun c => ClassGroup.mk0 (b c))) (I : Ideal (𝒪 K)) (hI : I ≠ ⊥) : ∃! c, (I * (b c : Ideal (𝒪 K))).IsPrincipal

theorem **_root_.GlobalTheory.tsum_isPrincipal_mul_eq** {K : Type} [Field K] [NumberField K] (B : Ideal (𝒪 K)) (hB : B ≠ ⊥) (F : Ideal (𝒪 K) → ℂ) : ∑' I : {I : Ideal (𝒪 K) // I ≠ ⊥ ∧ (I * B).IsPrincipal}, F (I * B) = ∑' J : {J : Ideal (𝒪 K) // J ≠ ⊥ ∧ J.IsPrincipal ∧ J ≤ B}, F J

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TsumIdealMulAbsNormCpowEqExpTsum

theorem **tsum_ideal_mul_absNorm_cpow_eq_exp_tsum** : ∀ (K : Type) [Field K] [NumberField K] (X : Ideal (𝒪 K) → ℂ), X ⊥ = 0 → X ⊤ = 1 → (∀ I J, X (I * J) = X I * X J) → (∀ I, ‖X I‖ ≤ 1) → ∀ s : ℂ, 1 < s.re → ∑' I, X I * ((Ideal.absNorm I : ℕ) : ℂ) ^ (-s) = Complex.exp (∑' P : HeightOneSpectrum (𝒪 K), -Complex.log (1 - X P.asIdeal * ((Ideal.absNorm P.asIdeal : ℕ) : ℂ) ^ (-s)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoMulCardInfinitePlaceLeQuadratic

theorem **card_real_eq_pos_add_tReal** (F : Type) [Field F] [NumberField F] (δ : F) (hδ : δ ≠ 0) : Nat.card (F →+* ℝ) = Nat.card {φ : F →+* ℝ // 0 < φ δ} + tReal F δ

theorem **two_mul_card_pos_le** (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E) (h : IsQuad δ θ σ) : 2 * Nat.card {φ : F →+* ℝ // 0 < φ δ} ≤ Nat.card (E →+* ℝ)

theorem **two_mul_card_infinitePlace_le** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → 2 * Fintype.card (InfinitePlace F) ≤ Fintype.card (InfinitePlace E) + tReal F δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoMulCardInfinitePlaceLeQuadraticStatement

theorem **_root_.GlobalTheory.T.two_mul_card_infinitePlace_le_quadratic** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → 2 * Fintype.card (InfinitePlace F) ≤ Fintype.card (InfinitePlace E) + tReal F δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoMulCardPlacesLeCardOverQuadratic

theorem **two_mul_card_le_card_over** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → 2 * S.card ≤ SE.card + tFin F δ S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoMulCardPlacesLeCardOverQuadraticStatement

theorem **_root_.GlobalTheory.T.two_mul_card_places_le_card_over_quadratic** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ (S : Finset (HeightOneSpectrum (𝒪 F))) (SE : Finset (HeightOneSpectrum (𝒪 E))), (∀ w : HeightOneSpectrum (𝒪 E), w ∈ SE ↔ w.under (𝒪 F) ∈ S) → 2 * S.card ≤ SE.card + tFin F δ S

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoMulIndexSqsInvolution

private theorem **mem_sqs_iff_exists_sq** {G : Type} [CommGroup G] (U : Subgroup G) (x : G) : x ∈ sqs U ↔ ∃ y ∈ U, y ^ 2 = x

theorem **two_mul_index_sqs_involution** : ∀ (A : Type) [CommGroup A] (σ : A →* A) (ε : A), (∀ a, σ (σ a) = a) → (∀ a, a * σ a = 1 → ∃ b, a = b / σ b) → ε ≠ 1 → ε ^ 2 = 1 → σ ε = ε → (∀ a : A, a ^ 2 = 1 → a = 1 ∨ a = ε) → 2 * (sqs (⊤ : Subgroup A)).index = (sqs (σ.eqLocus (MonoidHom.id A))).relIndex (σ.eqLocus (MonoidHom.id A)) * (sqs (MonoidHom.id A * σ).ker).relIndex (MonoidHom.id A * σ).ker

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoMulIndexSqsInvolutionStatement

theorem **_root_.GlobalTheory.T.two_mul_index_sqs_eq_relIndex_fixed_mul_relIndex_ker** : ∀ (A : Type) [CommGroup A] (σ : A →* A) (ε : A), (∀ a, σ (σ a) = a) → (∀ a, a * σ a = 1 → ∃ b, a = b / σ b) → ε ≠ 1 → ε ^ 2 = 1 → σ ε = ε → (∀ a : A, a ^ 2 = 1 → a = 1 ∨ a = ε) → 2 * (sqs (⊤ : Subgroup A)).index = (sqs (σ.eqLocus (MonoidHom.id A))).relIndex (σ.eqLocus (MonoidHom.id A)) * (sqs (MonoidHom.id A * σ).ker).relIndex (MonoidHom.id A * σ).ker

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoPlacesOfEmbeddings

theorem **two_places_of_embeddings** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) (ψ ψ' : E →ₐ[F] v.adicCompletion F), ψ ≠ ψ' → ∃ w w' : HeightOneSpectrum (𝒪 E), w ≠ w' ∧ w.under (𝒪 F) = v ∧ w'.under (𝒪 F) = v

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoPowCardLeIndexIInf

theorem **two_pow_card_le_index_iInf** : ∀ (G ι : Type) [CommGroup G] [Fintype ι] (K : ι → Subgroup G), (∀ i, ∃ x, x ∉ K i ∧ ∀ j, j ≠ i → x ∈ K j) → (⊓ i, K i).index ≠ 0 → 2 ^ Fintype.card ι ≤ (⊓ i, K i).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoPowCardLeIndexIInfStatement

theorem **_root_.GlobalTheory.T.two_pow_card_le_index_iInf_of_independent** : ∀ (G ι : Type) [CommGroup G] [Fintype ι] (K : ι → Subgroup G), (∀ i, ∃ x, x ∉ K i ∧ ∀ j, j ≠ i → x ∈ K j) → (⊓ i, K i).index ≠ 0 → 2 ^ Fintype.card ι ≤ (⊓ i, K i).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoPowLeIndexLocalNormsSqrtExtension

theorem **norm_of_isSquare_mul** {L : Type} [Field L] {x y δ : L} (hx : x ≠ 0) (h : IsSquare (x * y)) (hn : ∃ a b : L, x = a ^ 2 - δ * b ^ 2) : ∃ a b : L, y = a ^ 2 - δ * b ^ 2

theorem **mem_normSub_real_of_pos** {F : Type} [Field F] (σ : F →+* ℝ) (δ : F) (x : Fˣ) (h : 0 < σ (x : F)) : x ∈ normSub σ δ

theorem **locSq_le_locNorm_of_delta** (F : Type) [Field F] [NumberField F] (δ : F) (S : Finset (HeightOneSpectrum (𝒪 F))) : locSq F S ≤ locNorm F δ S

theorem **two_pow_le_index_locNorm** : ∀ (F : Type) [Field F] [NumberField F] (δ : F) (S : Finset (HeightOneSpectrum (𝒪 F))), δ ≠ 0 → (locNorm F δ S).index ≠ 0 ∧ 2 ^ (tFin F δ S + tReal F δ) ≤ (locNorm F δ S).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.TwoPowLeIndexLocalNormsSqrtExtensionStatement

theorem **_root_.GlobalTheory.T.two_pow_le_index_localNormsSqrtExtension** : ∀ (F : Type) [Field F] [NumberField F] (δ : F) (S : Finset (HeightOneSpectrum (𝒪 F))), δ ≠ 0 → (locNorm F δ S).index ≠ 0 ∧ 2 ^ (tFin F δ S + tReal F δ) ≤ (locNorm F δ S).index

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnipotentUpperSizeReduction

theorem **unipotent_size_reduction_column** {N : ℕ} (n : Matrix (Fin N) (Fin N) ℝ) (h1 : ∀ i, n i i = 1) (h0 : ∀ i j, j < i → n i j = 0) (j : Fin N) (k : ℕ) : ∃ z : Fin N → ℤ, z j = 1 ∧ (∀ l, j < l → z l = 0) ∧ ∀ i, i < j → (j : ℕ) ≤ i + k → |∑ l, n i l * (z l : ℝ)| ≤ 1 / 2

theorem **unipotent_size_reduction** : ∀ (N : ℕ) (n : Matrix (Fin N) (Fin N) ℝ), (∀ i, n i i = 1) → (∀ i j, j < i → n i j = 0) → ∃ ν : Matrix (Fin N) (Fin N) ℤ, (∀ i, ν i i = 1) ∧ (∀ i j, j < i → ν i j = 0) ∧ ∀ i j, i < j → |(n * ν.map (Int.cast : ℤ → ℝ)) i j| ≤ 1 / 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnipotentUpperSizeReductionStatement

theorem **_root_.GlobalTheory.T.unipotent_upper_size_reduction** : ∀ (N : ℕ) (n : Matrix (Fin N) (Fin N) ℝ), (∀ i, n i i = 1) → (∀ i j, j < i → n i j = 0) → ∃ ν : Matrix (Fin N) (Fin N) ℤ, (∀ i, ν i i = 1) ∧ (∀ i j, j < i → ν i j = 0) ∧ ∀ i j, i < j → |(n * ν.map (Int.cast : ℤ → ℝ)) i j| ≤ 1 / 2

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryCharPiComplexUnitsEqProdZpow

private lemma **continuous_circle_toUnits'** : Continuous Circle.toUnits

private lemma **div_norm_sq_eq_div_conj_of_ne_zero** (x : ℂ) (hx : x ≠ 0) : (x / ‖x‖) ^ 2 = x / starRingEnd ℂ x

theorem **_root_.GlobalTheory.unitaryChar_pi_complexUnits_eq_prod_zpow** : ∀ (ι : Type) [Fintype ι] (f : (ι → ℂˣ) →* Circle), Continuous f → (∀ z : ι → ℂˣ, (∀ i, ((z i : ℂ)).im = 0) → f z = 1) → ∃ k : ι → ℤ, ∀ z, ((f z : Circle) : ℂ) = ∏ i, ((z i : ℂ) / starRingEnd ℂ (z i : ℂ)) ^ k i

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryCharPiComplexUnitsEqProdZpowStatement

theorem **_root_.GlobalTheory.T.exists_zpow_prod_div_conj_of_continuous_trivial_on_real** : ∀ (ι : Type) [Fintype ι] (f : (ι → ℂˣ) →* Circle), Continuous f → (∀ z : ι → ℂˣ, (∀ i, ((z i : ℂ)).im = 0) → f z = 1) → ∃ k : ι → ℤ, ∀ z, ((f z : Circle) : ℂ) = ∏ i, ((z i : ℂ) / starRingEnd ℂ (z i : ℂ)) ^ k i

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryClassNumberDecomposition

theorem **exists_reps** {X ι : Type} (P : X → Prop) (c : X → ι → Prop) (R : X → X → Prop) (I : Set ι) (hI : I.Finite) (h1 : ∀ x, P x → ∃ i ∈ I, c x i) (h2 : ∀ x y i, P x → P y → c x i → c y i → R x y) : ∃ T : Set X, T.Finite ∧ (∀ y ∈ T, P y) ∧ ∀ x, P x → ∃ y ∈ T, R x y

theorem **fm_mul** (γ δ : Matrix (Fin n) (Fin n) E) : fm F (γ * δ) = fm F γ * fm F δ

theorem **fm_one** : fm F (1 : Matrix (Fin n) (Fin n) E) = 1

theorem **mem_one_iff** {E : Type} [Field E] [NumberField E] (x : E) : x ∈ (1 : FractionalIdeal (nonZeroDivisors (𝓞 E)) E) ↔ IsIntegral ℤ x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryEqQuasiReflectionMulTransvections

theorem **tv_zero** {R : Type} [CommRing R] {n : ℕ} (σ : R →+* R) (G : Matrix (Fin n) (Fin n) R) (x : Fin n → R) : tv σ G x 0 = 1

private theorem **refl_eq_tv_sub_one_div_hf** {F E : Type} [Field F] [Field E] [Algebra F E] (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (x : Fin n → E) (u : E) : _root_.GlobalTheory.UnRep.refl c G x u = _root_.GlobalTheory.UTrans.tv (c : E →+* E) G x ((u - 1) / _root_.GlobalTheory.UTrans.hf (c : E →+* E) G x x)

private theorem **det_refl_eq_of_hermE_ne_zero** {F E : Type} [Field F] [Field E] [Algebra F E] (c : E ≃ₐ[F] E) {n : ℕ} (G : Matrix (Fin n) (Fin n) E) (x : Fin n → E) (u : E) (hx : _root_.GlobalTheory.UnRep.hermE c G x x ≠ 0) : (_root_.GlobalTheory.UnRep.refl c G x u).det = u

theorem **_root_.GlobalTheory.UTrans.isU_eq_tv_mul_tvGen** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), _root_.GlobalTheory.UTrans.IsHerm (c : E →+* E) G → (∃ z : Fin n → E, z ≠ 0 ∧ _root_.GlobalTheory.UTrans.hf (c : E →+* E) G z z = 0) → ∀ γ, _root_.GlobalTheory.UTrans.IsU (c : E →+* E) G γ → (∀ x, _root_.GlobalTheory.UTrans.hf (c : E →+* E) G x x ≠ 0 → ∃ m ∈ _root_.GlobalTheory.UTrans.TvGen (c : E →+* E) G, γ = _root_.GlobalTheory.UTrans.tv (c : E →+* E) G x ((γ.det - 1) / _root_.GlobalTheory.UTrans.hf (c : E →+* E) G x x) * m) ∧ (γ.det = 1 → γ ∈ _root_.GlobalTheory.UTrans.TvGen (c : E →+* E) G)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryEqQuasiReflectionMulTransvectionsStatement

theorem **_root_.GlobalTheory.T.unitary_eq_quasiReflection_mul_transvections** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHerm (c : E →+* E) G → (∃ z : Fin n → E, z ≠ 0 ∧ hf (c : E →+* E) G z z = 0) → ∀ γ, IsU (c : E →+* E) G γ → (∀ x, hf (c : E →+* E) G x x ≠ 0 → ∃ m ∈ TvGen (c : E →+* E) G, γ = tv (c : E →+* E) G x ((γ.det - 1) / hf (c : E →+* E) G x x) * m) ∧ (γ.det = 1 → γ ∈ TvGen (c : E →+* E) G)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryGroupWeakApproximationAtPrimes

theorem **preserves_iff** {R : Type} [CommRing R] (σ : R →+* R) {n : ℕ} (G g : Matrix (Fin n) (Fin n) R) : (∀ x y : Fin n → R, ∑ a, ∑ b, (g *ᵥ x) a * G a b * σ ((g *ᵥ y) b) = ∑ a, ∑ b, x a * G a b * σ (y b)) ↔ gᵀ * G * g.map σ = G

theorem **map_inv'** {K R : Type} [CommRing K] [CommRing R] (ι : K →+* R) {n : ℕ} (A : Matrix (Fin n) (Fin n) K) (hA : IsUnit A.det) : (A.map ι)⁻¹ = A⁻¹.map ι

theorem **map_smul''** {K R : Type} [CommRing K] [CommRing R] (ι : K →+* R) {n : ℕ} (r : K) (A : Matrix (Fin n) (Fin n) K) : (r • A).map ι = ι r • A.map ι

theorem **idem_factor** {R M : Type} [CommRing R] [Ring M] [Algebra R M] (e : R) (he : e * e = e) (S T U : M) (hU : U * (1 + S) = 1) : e • (1 + T) = e • ((1 + e • (-((S - T) * U))) * (1 + S))

theorem **idem_diff** {R M : Type} [CommRing R] [Ring M] [Algebra R M] (e : R) (he : e * e = e) (S T U V W : M) (hU : U * (1 + S) = 1) (hU' : (1 + S) * U = 1) (hV : V * (1 + T) = 1) (hV' : (1 + T) * V = 1) (hW : W * (1 + e • (-((S - T) * U))) = 1) : e • (V - U) = e • (U * W * (S - T) * U)

theorem **preserves_form_smul_idempotent_glue** {R : Type} [CommRing R] (σ : R →+* R) {n : ℕ} (G h : Matrix (Fin n) (Fin n) R) (e l : R) (he : e * e = e) (hσ : σ e = e) (hl : l * σ l = 1) (hh : hᵀ * G * h.map σ = G) : (l • (e • h + (1 - e) • (1 : Matrix (Fin n) (Fin n) R)))ᵀ * G * (l • (e • h + (1 - e) • (1 : Matrix (Fin n) (Fin n) R))).map σ = G

theorem **core** {K R : Type} [Field K] [CommRing R] (ι : K →+* R) (c : K →+* K) (σ : R →+* R) (hσι : ∀ x, σ (ι x) = ι (c x)) (hc : ∀ x, c (c x) = x) (hσ : ∀ x, σ (σ x) = x) {n : ℕ} (G : Matrix (Fin n) (Fin n) K) (hG : (G.map c)ᵀ = G) (hdet : G.det ≠ 0) (h2 : (2 : K) ≠ 0) (O : Subring (Matrix (Fin n) (Fin n) R)) (hb : ∀ M, ∃ d : ℕ, 0 < d ∧ d • M ∈ O) (e : R) (he : e * e = e) (hσe : σ e = e) (he0 : e ≠ 0) (heO : ∀ X ∈ O, e • X ∈ O) (N : ℕ) (hN : 0 < N) (hnak : ∀ X ∈ O, ∃ W ∈ O, (1 + e • (N • X)) * W = 1) (hapx : ∀ (M : Matrix (Fin n) (Fin n) R) (D : ℕ), 0 < D → ∃ M₀ : Matrix (Fin n) (Fin n) K, ∃ X ∈ O, (M - (G.map ι * M.map σ * (G.map ι)⁻¹)ᵀ) - (M₀.map ι - (G.map ι * (M₀.map ι).map σ * (G.map ι)⁻¹)ᵀ) = D • X) (g h : Matrix (Fin n) (Fin n) R) (hg : h * g = 1) (hh : hᵀ * G.map ι * h.map σ = G.map ι) (lam : K) (hlam : lam * c lam = 1) (hu : IsUnit (1 + ι lam • (e • h + (1 - e) • (1 : Matrix (Fin

theorem **exists_stab** {R : Type} [CommRing R] {n : ℕ} (Λ : AddSubgroup (Fin n → R)) (M : Matrix (Fin n) (Fin n) R) (D : ℕ) (hD : IsUnit (D : R)) (h : ∀ z ∈ Λ, ∃ w ∈ Λ, M *ᵥ z = D • w) : ∃ X ∈ stab Λ, M = D • X

theorem **skewPart_smul** {R : Type} [CommRing R] (σ : R →+* R) {n : ℕ} (G : Matrix (Fin n) (Fin n) R) (a : R) (ha : σ a = a) (A : Matrix (Fin n) (Fin n) R) : skewPart σ G (a • A) = a • skewPart σ G A

theorem **unitary_inv** {R : Type} [CommRing R] (σ : R →+* R) {n : ℕ} (G g h : Matrix (Fin n) (Fin n) R) (hgh : g * h = 1) (hg : gᵀ * G * g.map σ = G) : hᵀ * G * h.map σ = G

theorem **weak_approximation** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝓞 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (RA F E)) (N D : ℕ), IsUnA c G g.1 → 0 < N → 0 < D → ∃ γ : GL (Fin n) E, IsUnE c G γ.1 ∧ CongAtA F L N D (fm F γ.1 * g.1)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryGroupWeakApproximationAtPrimesStatement

theorem **_root_.GlobalTheory.T.unitaryGroup_weak_approximation_at_primes** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → Module.finrank F E = 2 → ∀ L : Submodule (𝓞 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ (g : GL (Fin n) (RA F E)) (N D : ℕ), IsUnA c G g.1 → 0 < N → 0 < D → ∃ γ : GL (Fin n) E, IsUnE c G γ.1 ∧ CongAtA F L N D (fm F γ.1 * g.1)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryMatrixEqProdQuasiReflections

theorem **isUnE_mul** {a b : Matrix (Fin n) (Fin n) E} (ha : IsUnE c G a) (hb : IsUnE c G b) : IsUnE c G (a * b)

theorem **isUnE_prod** (hG : IsHermForm c G) (l : List ((Fin n → E) × E)) (hl : ∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1) : IsUnE c G (l.map fun p => refl c G p.1 p.2).prod

theorem **exists_inv** (hG : IsHermForm c G) (l : List ((Fin n → E) × E)) (hl : ∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1) : ∃ l' : List ((Fin n → E) × E), (∀ p ∈ l', hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1) ∧ (l'.map fun p => refl c G p.1 p.2).prod * (l.map fun p => refl c G p.1 p.2).prod = 1

theorem **prod_fix** (hG : IsHermForm c G) (v : Fin n → E) (l : List ((Fin n → E) × E)) (hl : ∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ hermE c G v p.1 = 0) : (l.map fun p => refl c G p.1 p.2).prod *ᵥ v = v

theorem **herm_span** {v x z y : Fin n → E} (hx : hermE c G v x = 0) (hz : hermE c G v z = 0) (hy : y ∈ Submodule.span E {x, z}) : hermE c G v y = 0

theorem **exists_fix** (hG : IsHermForm c G) {γ : Matrix (Fin n) (Fin n) E} (hγ : IsUnE c G γ) (x : Fin n → Fin n → E) (h1 : ∀ i, hermE c G (x i) (x i) ≠ 0) (h2 : ∀ i j, i ≠ j → hermE c G (x i) (x j) = 0) (k : ℕ) : ∃ l : List ((Fin n → E) × E), (∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1) ∧ ∀ i : Fin n, i.val < k → ((l.map fun p => refl c G p.1 p.2).prod * γ) *ᵥ x i = x i

theorem **GlobalTheory.UnRep.isUnE_eq_prod_refl** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ γ, IsUnE c G γ → ∃ l : List ((Fin n → E) × E), (∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1) ∧ γ = (l.map fun p => refl c G p.1 p.2).prod

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryMatrixEqProdQuasiReflectionsStatement

theorem **_root_.GlobalTheory.T.unitaryMatrix_eq_prod_quasiReflections** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ γ, IsUnE c G γ → ∃ l : List ((Fin n → E) × E), (∀ p ∈ l, hermE c G p.1 p.1 ≠ 0 ∧ p.2 * c p.2 = 1) ∧ γ = (l.map fun p => refl c G p.1 p.2).prod

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryQuasiReflectionLaws

theorem **_root_.GlobalTheory.UnRep.ReflLaws.hermE_eq_dot** (v x : Fin n → E) : _root_.GlobalTheory.UnRep.hermE c G v x = (G *ᵥ fun b => c (x b)) ⬝ᵥ v

theorem **_root_.GlobalTheory.UnRep.ReflLaws.refl_mulVec** (x : Fin n → E) (u : E) (v : Fin n → E) : _root_.GlobalTheory.UnRep.refl c G x u *ᵥ v = v + ((u - 1) * _root_.GlobalTheory.UnRep.hermE c G v x / _root_.GlobalTheory.UnRep.hermE c G x x) • x

private theorem **hermE_add_smul_left'** (v x y : Fin n → E) (t : E) : _root_.GlobalTheory.UnRep.hermE c G (v + t • x) y = _root_.GlobalTheory.UnRep.hermE c G v y + t * _root_.GlobalTheory.UnRep.hermE c G x y

private theorem **hermE_conj_eq_swap_of_involutive_of_conj_symm** (hinv : ∀ z : E, c (c z) = z) (hG : ∀ a b, c (G a b) = G b a) (a b : Fin n → E) : c (_root_.GlobalTheory.UnRep.hermE c G a b) = _root_.GlobalTheory.UnRep.hermE c G b a

private theorem **hermE_add_smul_right'** (v x y : Fin n → E) (t : E) : _root_.GlobalTheory.UnRep.hermE c G v (y + t • x) = _root_.GlobalTheory.UnRep.hermE c G v y + c t * _root_.GlobalTheory.UnRep.hermE c G v x

theorem **_root_.GlobalTheory.UnRep.ReflLaws.refl_laws** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), _root_.GlobalTheory.UnRep.IsHermForm c G → ∀ x : Fin n → E, _root_.GlobalTheory.UnRep.hermE c G x x ≠ 0 → (∀ u v, _root_.GlobalTheory.UnRep.refl c G x u *ᵥ v = v + ((u - 1) * _root_.GlobalTheory.UnRep.hermE c G v x / _root_.GlobalTheory.UnRep.hermE c G x x) • x) ∧ _root_.GlobalTheory.UnRep.refl c G x 1 = 1 ∧ (∀ u u', _root_.GlobalTheory.UnRep.refl c G x u * _root_.GlobalTheory.UnRep.refl c G x u' = _root_.GlobalTheory.UnRep.refl c G x (u * u')) ∧ ∀ u, u * c u = 1 → _root_.GlobalTheory.UnRep.IsUnE c G (_root_.GlobalTheory.UnRep.refl c G x u)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryQuasiReflectionLawsStatement

theorem **_root_.GlobalTheory.T.unitaryQuasiReflection_laws** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), IsHermForm c G → ∀ x : Fin n → E, hermE c G x x ≠ 0 → (∀ u v, refl c G x u *ᵥ v = v + ((u - 1) * hermE c G v x / hermE c G x x) • x) ∧ refl c G x 1 = 1 ∧ (∀ u u', refl c G x u * refl c G x u' = refl c G x (u * u')) ∧ ∀ u, u * c u = 1 → IsUnE c G (refl c G x u)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryTransvectionSubmonoidAdjustIntegralAway

theorem **tvGen_adjust** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), Module.finrank F E = 2 → IsHerm (c : E →+* E) G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ m ∈ TvGen (c : E →+* E) G, ∀ b : ℕ, 0 < b → ∀ N : ℕ, ∃ m' ∈ TvGen (c : E →+* E) G, ∃ k e : ℕ, k.Coprime b ∧ ∀ v ∈ L, (∃ w ∈ L, k • (m' *ᵥ v - m *ᵥ v) = b ^ N • w) ∧ b ^ e • (m' *ᵥ v) ∈ L ∧ b ^ e • (m'⁻¹ *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryTransvectionSubmonoidAdjustIntegralAwayStatement

theorem **_root_.GlobalTheory.T.unitaryTransvectionSubmonoid_adjust_integral_away** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E) (n : ℕ) (G : Matrix (Fin n) (Fin n) E), Module.finrank F E = 2 → IsHerm (c : E →+* E) G → ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n → E)) = ⊤ → ∀ m ∈ TvGen (c : E →+* E) G, ∀ b : ℕ, 0 < b → ∀ N : ℕ, ∃ m' ∈ TvGen (c : E →+* E) G, ∃ k e : ℕ, k.Coprime b ∧ ∀ v ∈ L, (∃ w ∈ L, k • (m' *ᵥ v - m *ᵥ v) = b ^ N • w) ∧ b ^ e • (m' *ᵥ v) ∈ L ∧ b ^ e • (m'⁻¹ *ᵥ v) ∈ L

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryTransvectionSubmonoidNormalDetOne

theorem **IsU.transpose_mul_mul_map** {R : Type} [CommRing R] {n : ℕ} {σ : R →+* R} {G γ : Matrix (Fin n) (Fin n) R} (h : IsU σ G γ) : γᵀ * G * γ.map σ = G

theorem **IsU.det_mul_conj_det** {E : Type} [Field E] {n : ℕ} {σ : E →+* E} {G γ : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) (h : IsU σ G γ) : γ.det * σ γ.det = 1

theorem **IsU.nonsing_inv** {E : Type} [Field E] {n : ℕ} {σ : E →+* E} {G γ : Matrix (Fin n) (Fin n) E} (hG : G.det ≠ 0) (h : IsU σ G γ) : IsU σ G γ⁻¹

theorem **tvGen_normal** : ∀ (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), IsHerm σ G → (∀ γ, IsU σ G γ → γ.det * σ γ.det = 1 ∧ IsU σ G γ⁻¹) ∧ (∀ m ∈ TvGen σ G, IsU σ G m ∧ m.det = 1 ∧ m⁻¹ ∈ TvGen σ G ∧ ∀ γ, IsU σ G γ → ∃ m' ∈ TvGen σ G, γ * m = m' * γ ∧ ∃ m'' ∈ TvGen σ G, m * γ = γ * m'')

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitaryTransvectionSubmonoidNormalDetOneStatement

theorem **_root_.GlobalTheory.T.unitaryTransvectionSubmonoid_normal_det_one** : ∀ (E : Type) [Field E] [CharZero E] (n : ℕ) (σ : E →+* E) (G : Matrix (Fin n) (Fin n) E), IsHerm σ G → (∀ γ, IsU σ G γ → γ.det * σ γ.det = 1 ∧ IsU σ G γ⁻¹) ∧ (∀ m ∈ TvGen σ G, IsU σ G m ∧ m.det = 1 ∧ m⁻¹ ∈ TvGen σ G ∧ ∀ γ, IsU σ G γ → ∃ m' ∈ TvGen σ G, γ * m = m' * γ ∧ ∃ m'' ∈ TvGen σ G, m * γ = γ * m'')

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitsLocalKummerNormsEqTopOfPow

theorem **kNorm_eq_top_of_pow** : ∀ (p : ℕ) (F L : Type) [Field F] [Field L] [CharZero L] (φ : F →+* L) (δ : F) (r : L), 0 < p → δ ≠ 0 → r ^ p = φ δ → kNorm p φ δ = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitsLocalKummerNormsEqTopOfPowStatement

theorem **_root_.GlobalTheory.T.unitsLocalKummerNorms_eq_top_of_pow** : ∀ (p : ℕ) (F L : Type) [Field F] [Field L] [CharZero L] (φ : F →+* L) (δ : F) (r : L), 0 < p → δ ≠ 0 → r ^ p = φ δ → kNorm p φ δ = ⊤

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitsLocalPowersAtPlacesIndexLe

private theorem **exists_valuation_natCast_eq_exp_neg_and_dvd** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝓞 F)) {p : ℕ} (hp : 0 < p) : ∃ e : ℕ, HeightOneSpectrum.valuation F v (p : F) = WithZero.exp (-(e : ℤ)) ∧ v.asIdeal ^ e ∣ Ideal.span {(p : 𝓞 F)}

private theorem **powSub_eq_comap_pows_units_map** {F L : Type} [Field F] [Field L] {p : ℕ} (hp : 0 < p) (φ : F →+* L) : _root_.GlobalTheory.CNorm.powSub p φ = (_root_.GlobalTheory.CNorm.pows p (⊤ : Subgroup Lˣ)).comap (Units.map φ.toMonoidHom)

private theorem **powSub_index_ne_zero_and_le_pIdx** {F L : Type} [Field F] [Field L] {p : ℕ} (hp : 0 < p) (φ : F →+* L) (h : _root_.GlobalTheory.CNorm.pIdx p (⊤ : Subgroup Lˣ) ≠ 0) : (_root_.GlobalTheory.CNorm.powSub p φ).index ≠ 0 ∧ (_root_.GlobalTheory.CNorm.powSub p φ).index ≤ _root_.GlobalTheory.CNorm.pIdx p (⊤ : Subgroup Lˣ)

private theorem **pTor_units_le** {L : Type} [Field L] {p : ℕ} (hp : 0 < p) : _root_.GlobalTheory.CNorm.pTor p (⊤ : Subgroup Lˣ) ≤ p

private theorem **prod_absNorm_pow_le_pow_finrank** {F : Type} [Field F] [NumberField F] (S : Finset (HeightOneSpectrum (𝓞 F))) {p : ℕ} (hp : 0 < p) (e : HeightOneSpectrum (𝓞 F) → ℕ) (he : ∀ v ∈ S, v.asIdeal ^ e v ∣ Ideal.span {(p : 𝓞 F)}) : ∏ v ∈ S, Ideal.absNorm v.asIdeal ^ e v ≤ p ^ Module.finrank ℚ F

theorem **_root_.GlobalTheory.CNorm.index_finPow_le** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (S : Finset (HeightOneSpectrum (𝓞 F))), 0 < p → (_root_.GlobalTheory.CNorm.finPow p F S).index ≠ 0 ∧ (_root_.GlobalTheory.CNorm.finPow p F S).index ≤ p ^ (2 * (S.card + Fintype.card (InfinitePlace F)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitsLocalPowersAtPlacesIndexLeStatement

theorem **_root_.GlobalTheory.T.unitsLocalPowersAtPlaces_index_le** : ∀ (p : ℕ) (F : Type) [Field F] [NumberField F] [IsTotallyComplex F] (S : Finset (HeightOneSpectrum (𝓞 F))), 0 < p → (finPow p F S).index ≠ 0 ∧ (finPow p F S).index ≤ p ^ (2 * (S.card + Fintype.card (InfinitePlace F)))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitsMappingToPowersLeUnitsLocalKummerNorms

theorem **norm_map_of_basis** {F L A B ι : Type} [CommRing F] [CommRing L] [CommRing A] [CommRing B] [Algebra F A] [Algebra L B] [Fintype ι] [DecidableEq ι] (φ : F →+* L) (ψ : A →+* B) (hψ : ∀ c, ψ (algebraMap F A c) = algebraMap L B (φ c)) (b : Module.Basis ι F A) (b' : Module.Basis ι L B) (hb : ∀ i, ψ (b i) = b' i) (y : A) : Algebra.norm L (ψ y) = φ (Algebra.norm F y)

theorem **exists_basis_pow_root** {L : Type} [Field L] {p : ℕ} (hp : 0 < p) (a : L) : ∃ b : Module.Basis (Fin p) L (AdjoinRoot (X ^ p - C a : L[X])), ∀ i, b i = AdjoinRoot.root (X ^ p - C a : L[X]) ^ (i : ℕ)

theorem **powSub_le_kNorm_and_kNorm_map** : ∀ (p : ℕ), 0 < p → (∀ (F L : Type) [Field F] [Field L] (φ : F →+* L) (δ : F), powSub p φ ≤ kNorm p φ δ) ∧ ∀ (F L : Type) [Field F] [Field L] (φ : F →+* L) (δ : F), kNorm p (RingHom.id F) δ ≤ kNorm p φ δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.UnitsMappingToPowersLeUnitsLocalKummerNormsStatement

theorem **_root_.GlobalTheory.T.unitsMappingToPowers_le_unitsLocalKummerNorms** : ∀ (p : ℕ), 0 < p → (∀ (F L : Type) [Field F] [Field L] (φ : F →+* L) (δ : F), powSub p φ ≤ kNorm p φ δ) ∧ ∀ (F L : Type) [Field F] [Field L] (φ : F →+* L) (δ : F), kNorm p (RingHom.id F) δ ≤ kNorm p φ δ

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ValuationAlgebraMapEqCount

theorem **valuation_algebraMap_eq_count** {r : R} (hr : r ≠ 0) : v.valuation K (algebraMap R K r) = WithZero.exp (-FractionalIdeal.count K v (spanSingleton R⁰ (algebraMap R K r)))

theorem **valuation_eq_exp_neg_count** {x : K} (hx : x ≠ 0) : v.valuation K x = WithZero.exp (-FractionalIdeal.count K v (spanSingleton R⁰ x))

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ValuationEqOneOfNormOneOfNotMoved

theorem **intValuation_map_le_of_map_le** {R : Type} [CommRing R] [IsDedekindDomain R] (σ : R →+* R) (P : HeightOneSpectrum R) (h : P.asIdeal.map σ ≤ P.asIdeal) (a : R) : P.intValuation (σ a) ≤ P.intValuation a

theorem **valuation_eq_one_of_norm_one_of_not_moved** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (c : E ≃ₐ[F] E), (∀ x, c (c x) = x) → ∀ (u : E) (P : HeightOneSpectrum (𝓞 E)), u * c u = 1 → ¬ Moved c P.asIdeal → P.valuation E u = 1

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ValuationUnramifiedKummerOfCoprimeUnit

theorem **IsKum.sum_conj_eq** {p : ℕ} {F E : Type} [Field F] [Field E] [Algebra F E] {ζ δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsKum p ζ δ θ σ) (a : ℕ → F) (k : ℕ) (hk : k < p) : ∑ j ∈ Finset.range p, algebraMap F E (((ζ ^ k)⁻¹) ^ j) * (σ ^ j) (∑ i ∈ Finset.range p, a i • θ ^ i) = (p : E) * (a k • θ ^ k)

theorem **exists_uniformizer_lt_one** (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝓞 E)) (T : Finset (HeightOneSpectrum (𝓞 E))) (hw : w ∈ T) : ∃ y : E, w.valuation E y = WithZero.exp (-1) ∧ ∀ w' ∈ T, w'.valuation E y < 1

theorem **kummer_valuation_unramified** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ w : HeightOneSpectrum (𝓞 E), (w.under (𝓞 F)).valuation F (p : F) = 1 → (w.under (𝓞 F)).valuation F δ = 1 → ∀ x : F, w.valuation E (algebraMap F E x) = (w.under (𝓞 F)).valuation F x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ValuationUnramifiedKummerOfCoprimeUnitStatement

theorem **_root_.GlobalTheory.T.valuation_unramified_kummer_of_coprime_unit** : ∀ (p : ℕ) (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (ζ δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsKum p ζ δ θ σ → ∀ w : HeightOneSpectrum (𝒪 E), (w.under (𝒪 F)).valuation F (p : F) = 1 → (w.under (𝒪 F)).valuation F δ = 1 → ∀ x : F, w.valuation E (algebraMap F E x) = (w.under (𝒪 F)).valuation F x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ValuationUnramifiedOfUnit

theorem **IsQuad.exists_base_value** {F E : Type} [Field F] [Field E] [Algebra F E] {δ : F} {θ : E} {σ : E ≃ₐ[F] E} (h : IsQuad δ θ σ) (w : Valuation E ℤᵐ⁰) (h2 : w 2 = 1) (hδ : w (algebraMap F E δ) = 1) (x : E) : ∃ u : F, w x = w (algebraMap F E u)

theorem **valuation_unramified_of_unit** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), IsQuad δ θ σ → ∀ w : HeightOneSpectrum (𝒪 E), (w.under (𝒪 F)).valuation F 2 = 1 → (w.under (𝒪 F)).valuation F δ = 1 → ∀ x : F, w.valuation E (algebraMap F E x) = (w.under (𝒪 F)).valuation F x

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.WithZeroEqOfLeExpIff

theorem **withZero_eq_of_le_exp_iff** {a b : ℤᵐ⁰} (ha : a ≤ 1) (hb : b ≤ 1) (h : ∀ n : ℕ, a ≤ WithZero.exp (-(n : ℤ)) ↔ b ≤ WithZero.exp (-(n : ℤ))) : a = b

theorem **_root_.GlobalTheory.QNorm.intValuation_of_eq_comap** {R : Type} [CommRing R] [IsDedekindDomain R] (e : R ≃+* R) (w w' : HeightOneSpectrum R) (h : w'.asIdeal = w.asIdeal.comap e) (x : R) : w'.intValuation x = w.intValuation (e x)

theorem **_root_.GlobalTheory.QNorm.valuation_of_eq_comap** {E : Type} [Field E] [NumberField E] (σ : E ≃+* E) (w w' : HeightOneSpectrum (𝒪 E)) (h : w'.asIdeal = w.asIdeal.comap (RingOfIntegers.mapRingEquiv σ)) (y : E) : w'.valuation E y = w.valuation E (σ y)

private lemma **ringOfIntegers_mapRingEquiv_algebraMap** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E ≃ₐ[F] E) (n : 𝒪 F) : RingOfIntegers.mapRingEquiv (σ : E ≃+* E) (algebraMap (𝒪 F) (𝒪 E) n) = algebraMap (𝒪 F) (𝒪 E) n

private lemma **exists_algebraMap_eq_mul_conj_of_isQuad** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {δ : F} {θ : E} {σ : E ≃ₐ[F] E} (hq : _root_.GlobalTheory.QNorm.IsQuad δ θ σ) (x : 𝒪 E) : ∃ n : 𝒪 F, algebraMap (𝒪 F) (𝒪 E) n = x * RingOfIntegers.mapRingEquiv (σ : E ≃+* E) x

theorem **_root_.GlobalTheory.QNorm.conj_places_transitive** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (δ : F) (θ : E) (σ : E ≃ₐ[F] E), _root_.GlobalTheory.QNorm.IsQuad δ θ σ → (∀ w : HeightOneSpectrum (𝒪 E), ∃ w' : HeightOneSpectrum (𝒪 E), w'.under (𝒪 F) = w.under (𝒪 F) ∧ ∀ y : E, w'.valuation E y = w.valuation E (σ y)) ∧ ∀ w w' : HeightOneSpectrum (𝒪 E), w'.under (𝒪 F) = w.under (𝒪 F) → w' = w ∨ ∀ y : E, w'.valuation E y = w.valuation E (σ y)

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ZlatticeTsumDilateSubIntegralLeRapid

theorem **fourier_comp_smul_of_pos** {V : Type} [NormedAddCommGroup V] [InnerProductSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] (f : V → ℂ) {t : ℝ} (ht : 0 < t) (w : V) : (𝓕 (fun x => f (t • x)) : V → ℂ) w = (((t ^ Module.finrank ℝ V)⁻¹ : ℝ) : ℂ) * (𝓕 f : V → ℂ) (t⁻¹ • w)

theorem **latticeSum_dilate_sub_integral_le** (V : Type) [NormedAddCommGroup V] [InnerProductSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L] (f : V → ℂ) (hf : ContDiff ℝ ∞ f) (hc : HasCompactSupport f) (A : ℕ) : ∃ C : ℝ, ∀ (x₀ : V) (t : ℝ), 0 < t → t ≤ 1 → ‖∑' l : L, f (t • ((l : V) + x₀)) - (∫ x, f x) / ((ZLattice.covolume L * t ^ Module.finrank ℝ V : ℝ) : ℂ)‖ ≤ C * t ^ A

Development.AbelianVarietyTheory.Proofs.AdelicLatticeProofs.ZlatticeTsumDilateSubIntegralLeRapidStatement

theorem **_root_.GlobalTheory.T.zlattice_tsum_dilate_sub_integral_le_rapid** : ∀ (V : Type) [NormedAddCommGroup V] [InnerProductSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] (L : Submodule ℤ V) [DiscreteTopology L] [IsZLattice ℝ L] (f : V → ℂ), ContDiff ℝ ∞ f → HasCompactSupport f → ∀ A : ℕ, ∃ C : ℝ, ∀ (x₀ : V) (t : ℝ), 0 < t → t ≤ 1 → ‖∑' l : L, f (t • ((l : V) + x₀)) - (∫ x, f x) / ((ZLattice.covolume L * t ^ Module.finrank ℝ V : ℝ) : ℂ)‖ ≤ C * t ^ A

Development.AbelianVarietyTheory.Proofs.AlgebraicCurves.AlgebraicIndependentFreeVar

theorem **_root_.GlobalTheory.existsUnique_ringHom_OmS** : ∀ (K Ω L : Type) [Field K] [Field Ω] [Algebra K Ω] [Field L] [Algebra K L] (D N : ℕ) (j₀ : Fin (N + 1)) (x : Fin N → Ω) (τ : _root_.ReciprocityTheory.ChartRestriction.G.T0.TVar D N → L) (φ : Ω →ₐ[K] L), Algebra.trdeg K Ω = D → AlgebraicIndependent K τ → (∀ i : Fin D, τ (i.succ, j₀) + ∑ k, τ (i.succ, j₀.succAbove k) * φ (x k) = 0) → ∃! P : _root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N →+* L, (∀ a : Ω, P (algebraMap Ω (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N) a) = φ a) ∧ ∀ p, P (_root_.ReciprocityTheory.ChartRestriction.G.T0.sImg j₀ x p) = τ p

Development.AbelianVarietyTheory.Proofs.AlgebraicCurves.ExistsCompleteDiscreteValuationRingOfIsNormDiscr

theorem **normDiscrete_mem_vRing_iff** (x : Ω) : x ∈ vRing v ↔ 0 ≤ v x

theorem **normDiscrete_map_inv** {x : Ω} {n : ℤ} (hx : v x = (n : WithTop ℤ)) : v x⁻¹ = ((-n : ℤ) : WithTop ℤ)

theorem **normDiscrete_isUnit_iff** (x : vRing v) : IsUnit x ↔ v (x : Ω) = 0

theorem **normDiscrete_associated_pow** {π : vRing v} (hπ : v (π : Ω) = 1) (x : vRing v) {n : ℕ} (hx : v (x : Ω) = ((n : ℤ) : WithTop ℤ)) : Associated (π ^ n) x

theorem **normDiscrete_exists_nat** (hv : IsNormDiscrete v) (x : vRing v) (hx : x ≠ 0) : ∃ n : ℕ, v (x : Ω) = ((n : ℤ) : WithTop ℤ)

theorem **normDiscrete_irreducible** {π : vRing v} (hπ : v (π : Ω) = 1) : Irreducible π

theorem **normDiscrete_isDiscreteValuationRing** (hv : IsNormDiscrete v) : IsDiscreteValuationRing (vRing v)

theorem **exists_complete_discreteValuationRing_of_isNormDiscrete** : ∀ (Ω : Type) [Field Ω] (v : AddValuation Ω (WithTop ℤ)), IsNormDiscrete v → ∃ (V K' : Type) (_ : CommRing V) (_ : IsDomain V) (_ : IsDiscreteValuationRing V) (_ : Field K') (_ : Algebra V K') (_ : IsFractionRing V K') (f : Ω →+* K'), IsAdicComplete (IsLocalRing.maximalIdeal V) V ∧ (∀ a : Ω, (∃ x : V, algebraMap V K' x = f a) ↔ 0 ≤ v a) ∧ (∀ (n : ℕ) (a : Ω) (x : V), algebraMap V K' x = f a → (IsDiscreteValuationRing.addVal V x = n ↔ v a = ((n : ℤ) : WithTop ℤ))) ∧ ∀ x : V, ∃ (a : Ω) (x' : V), algebraMap V K' x' = f a ∧ x - x' ∈ IsLocalRing.maximalIdeal V

Development.AbelianVarietyTheory.Proofs.AlgebraicCurves.ExistsMulEqMapOfIsAlgebraic

private lemma **mem_span_of_lift_aeval_eq_zero_at_chart_point** (K Ω : Type) [Field K] [Field Ω] [Algebra K Ω] (D N : ℕ) (j₀ : Fin (N + 1)) (x : Fin N → Ω) (y : Fin (N + 1) → Ω) (hy₀ : y j₀ = 1) (hy : ∀ k, y (j₀.succAbove k) = x k) (w : _root_.ReciprocityTheory.ChartRestriction.G.T0.KTp K D N ⊗[K] Ω) (hw : Algebra.TensorProduct.lift (MvPolynomial.aeval (R

private lemma **genericSection_baseChange_surjective_ker_lemma** (K Ω : Type) [Field K] [Field Ω] [Algebra K Ω] (D N : ℕ) (j₀ : Fin (N + 1)) (x : Fin N → Ω) [Algebra (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N)] [FiniteDimensional (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N)] [IsScalarTower K (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N)] (y : Fin (N + 1) → Ω) (hcomp : ∀ p : _root_.ReciprocityTheory.ChartRestriction.G.T0.KTp K D N, algebraMap (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N) (algebraMap (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTp K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.KT

theorem **_root_.GlobalTheory.genericSection_quotient** : ∀ (K Ω K' : Type) [Field K] [Field Ω] [Algebra K Ω] (D N : ℕ) (j₀ : Fin (N + 1)) (x : Fin N → Ω) [Algebra (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N)] [FiniteDimensional (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N)] [Field K'] [Algebra K K'] [Algebra (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) K'] [IsScalarTower K (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) K'] (T : _root_.ReciprocityTheory.ChartRestriction.G.T0.TVar D N → K') (y : Fin (N + 1) → Ω), (∀ p : _root_.ReciprocityTheory.ChartRestriction.G.T0.KTp K D N, algebraMap (_root_.ReciprocityTheory.ChartRestriction.G.T0.KTf K D N) (_root_.ReciprocityTheory.ChartRestriction.G.T0.OmS Ω D N) (algebraMap (_root_.R

Development.AbelianVarietyTheory.Proofs.AlgebraicCurves.PlacesRestrictInjective

theorem **val_aeval_eq_coeff** {Ω U σ : Type} [Field Ω] [Field U] [Algebra Ω U] (w : AddValuation U (WithTop ℤ)) (t : σ → U) (ht : ∀ s, t s ∈ vRing w) (hind : letI

theorem **val_aeval_congr** {Ω U σ : Type} [Field Ω] [Field U] [Algebra Ω U] (w w' : AddValuation U (WithTop ℤ)) (t : σ → U) (ht : ∀ s, t s ∈ vRing w) (hind : letI

theorem **exists_orders** {Ω X : Type} [Field Ω] (S : Set X) (hS : S.Finite) (v : X → AddValuation Ω (WithTop ℤ)) (hinj : Set.InjOn v S) (hn : ∀ w ∈ S, IsNormDiscrete (v w)) (n : X → ℤ) : ∃ g : Ω, g ≠ 0 ∧ ∀ w ∈ S, v w g = n w

theorem **exists_integral_lift** {Ω X : Type} [Field Ω] (S : Set X) (hS : S.Finite) (v : X → AddValuation Ω (WithTop ℤ)) (hinj : Set.InjOn v S) (hn : ∀ w ∈ S, IsNormDiscrete (v w)) (w : X) (hw : w ∈ S) (a : Ω) (ha : 0 ≤ v w a) : ∃ l : Ω, (∀ w' ∈ S, 0 ≤ v w' l) ∧ 0 < v w (a - l)

theorem **exists_gauss_variables** (d : Data K ι Ω WT A) (hd : d.AxH 𝒜) (la : IsDedekindDomain.HeightOneSpectrum (𝒪 K)) (c : ι) (j0 : Fin (d.N + 1)) (hw0 : d.w j0 c ≠ 0) : ∃ t : SVar d.D d.N → OmS (Ω c) d.D d.N, (∀ P : MvPolynomial (SVar d.D d.N) (Ω c), ∃ Q : MvPolynomial (SVar d.D d.N) (Ω c), MvPolynomial.aeval t Q = algebraMap _ (OmS (Ω c) d.D d.N) P) ∧ ∀ w ∈ d.PlOver la c, (∀ j, 0 ≤ d.Wres c w (ratio (Ω

theorem **wres_injOn** (d : Data K ι Ω WT A) (hd : d.AxH 𝒜) (h : d.S1) (la : IsDedekindDomain.HeightOneSpectrum (𝒪 K)) (c : ι) : Set.InjOn (d.Wres c) (d.PlOver la c)

theorem **places_restrict_injective** (K : Type) [Field K] [NumberField K] (ι : Type) [Fintype ι] (Ω : ι → Type) [∀ c, Field (Ω c)] [∀ c, Algebra K (Ω c)] (WT : Type) [AddCommGroup WT] (A : ι → Type) [∀ c, CommRing (A c)] [∀ c, Algebra (Ω c) (A c)] [∀ c, Algebra K (A c)] [∀ c, Algebra (𝒪 K) (A c)] (𝒜 : (c : ι) → WT → Submodule (Ω c) (A c)) (d : Data K ι Ω WT A) (hd : d.AxH 𝒜) (h : d.S1) (la : IsDedekindDomain.HeightOneSpectrum (𝒪 K)) (c : ι) : (d.PlOver la c).Finite ∧ Set.InjOn (d.Wres c) (d.PlOver la c) ∧ (∀ n : AddValuation (OmS (Ω c) d.D d.N) (WithTop ℤ) → ℤ, ∃ g : Ω c, g ≠ 0 ∧ ∀ w ∈ d.PlOver la c, d.Wres c w g = n w) ∧ (∀ G : WT, ∃ e ∈ 𝒜 c G, e ≠ 0 ∧ ∀ w ∈ d.PlOver la c, d.ord c (d.Wres c w) G e = 0) ∧ ∀ w ∈ d.PlOver la c, ∀ a : Ω c, 0 ≤ d.Wres c w a → ∃ l : Ω c, (∀ w' ∈ d.PlOver la c, 0 ≤ d.Wres c w' l) ∧ 0 < d.Wres c w (a - l)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.AdditiveEqPolynomialVonMangoldtConvolutionOne

theorem **mangoldtConv_eq_of_dvd_pow** {f g : F[X]} (hf : f.Monic) (hf' : Irreducible f) (hg : g.Monic) (hg' : Irreducible g) {n : ℕ} (h : g ∣ f ^ n) : g = f

theorem **mangoldt_pow** (s : F[X] → ℂ) (f : F[X]) (m : ℕ) (hf : f.Monic) (hf' : Irreducible f) (hm : 0 < m) : mangoldt s (f ^ m) = s f

theorem **mangoldt_eq_zero** (s : F[X] → ℂ) (P : F[X]) (h : ¬ ∃ (f : F[X]) (m : ℕ), f.Monic ∧ Irreducible f ∧ 0 < m ∧ P = f ^ m) : mangoldt s P = 0

theorem **conv_const_one_eq_divisorSum** (a : F[X] → ℂ) (P : F[X]) (hP : P.Monic) : conv a (fun _ => 1) P = ∑ᶠ D ∈ {D : F[X] | D.Monic ∧ D ∣ P}, a D

theorem **mangoldtConv_finite_monicDivisors** (P : F[X]) (hP : P.Monic) : {D : F[X] | D.Monic ∧ D ∣ P}.Finite

theorem **mangoldt_divisorSum_mul** (s : F[X] → ℂ) (f Q : F[X]) (hf : f.Monic) (hf' : Irreducible f) (hQ : Q.Monic) : ∑ᶠ D ∈ {D : F[X] | D.Monic ∧ D ∣ f * Q}, mangoldt s D = s f + ∑ᶠ D ∈ {D : F[X] | D.Monic ∧ D ∣ Q}, mangoldt s D

theorem **IsAdditive.eq_divisorSum_mangoldt** {s : F[X] → ℂ} (hs : IsAdditive s) (P : F[X]) (hP : P.Monic) : s P = ∑ᶠ D ∈ {D : F[X] | D.Monic ∧ D ∣ P}, mangoldt s D

theorem **mangoldt_conv_spec** : ∀ (F : Type) [Field F] (s : F[X] → ℂ), IsAdditive s → (∀ P : F[X], P.Monic → s P = conv (mangoldt s) (fun _ => 1) P) ∧ (∀ (f : F[X]) (m : ℕ), f.Monic → Irreducible f → 0 < m → mangoldt s (f ^ m) = s f) ∧ ∀ P : F[X], (¬ ∃ (f : F[X]) (m : ℕ), f.Monic ∧ Irreducible f ∧ 0 < m ∧ P = f ^ m) → mangoldt s P = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.AdditiveEqPolynomialVonMangoldtConvolutionOneStatement

theorem **_root_.GlobalTheory.T.additive_eq_polynomialVonMangoldt_convolution_one** : ∀ (F : Type) [Field F] (s : F[X] → ℂ), IsAdditive s → (∀ P : F[X], P.Monic → s P = conv (mangoldt s) (fun _ => 1) P) ∧ (∀ (f : F[X]) (m : ℕ), f.Monic → Irreducible f → 0 < m → mangoldt s (f ^ m) = s f) ∧ ∀ P : F[X], (¬ ∃ (f : F[X]) (m : ℕ), f.Monic ∧ Irreducible f ∧ 0 < m ∧ P = f ^ m) → mangoldt s P = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.AdditiveMulEqVonMangoldtConvolution

theorem **conv_congr_monic** {a a' b b' : F[X] → ℂ} (ha : ∀ P : F[X], P.Monic → a P = a' P) (hb : ∀ P : F[X], P.Monic → b P = b' P) (P : F[X]) : conv a b P = conv a' b' P

theorem **conv_eq_sum** (a b : F[X] → ℂ) {P : F[X]} (hP : P.Monic) : conv a b P = ∑ x ∈ (GlobalTheory.T.monicFactorizationConvolution_finite_comm_assoc_add F P hP).1.toFinset, a x.1 * b x.2

theorem **pow_mul_conv** (q : ℂ) (a b : F[X] → ℂ) {P : F[X]} (hP : P.Monic) : q ^ P.natDegree * conv a b P = conv (fun D => q ^ D.natDegree * a D) (fun D => q ^ D.natDegree * b D) P

theorem **additive_mul_conv** {s : F[X] → ℂ} (hs : IsAdditive s) (a b : F[X] → ℂ) {P : F[X]} (hP : P.Monic) : s P * conv a b P = conv (fun D => s D * a D) b P + conv a (fun D => s D * b D) P

theorem **pow_natDegree_ne_one** {q : ℕ} (hq : 2 ≤ q) {n : ℕ} (hn : n ≠ 0) : (q : ℂ) ^ n - 1 ≠ 0

theorem **eq_zero_of_rec** {q : ℕ} (hq : 2 ≤ q) {E : F[X] → ℂ} (h1 : E 1 = 0) (hE : ∀ P : F[X], P.Monic → (q : ℂ) ^ P.natDegree * E P = conv (fun _ => 1) E P) : ∀ P : F[X], P.Monic → E P = 0

theorem **mangoldt_one** (s : F[X] → ℂ) : mangoldt s 1 = 0

theorem **conv_at_one** (a b : F[X] → ℂ) : conv a b 1 = a 1 * b 1

theorem **additive_mul_eq_conv** (F : Type) [Field F] (q : ℕ) (hq : 2 ≤ q) (s m : F[X] → ℂ) (hs : IsAdditive s) (_hm1 : m 1 = 1) (hm : ∀ P : F[X], P.Monic → (q : ℂ) ^ P.natDegree * m P = conv (fun _ => 1) m P) (P : F[X]) (hP : P.Monic) : s P * m P = conv (fun D => mangoldt s D / ((q : ℂ) ^ D.natDegree - 1)) m P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.AdditiveMulEqVonMangoldtConvolutionStatement

theorem **_root_.GlobalTheory.T.additive_mul_eq_vonMangoldt_convolution** : ∀ (F : Type) [Field F] (q : ℕ), 2 ≤ q → ∀ (s m : F[X] → ℂ), IsAdditive s → m 1 = 1 → (∀ P : F[X], P.Monic → (q : ℂ) ^ P.natDegree * m P = conv (fun _ => 1) m P) → ∀ P : F[X], P.Monic → s P * m P = conv (fun D => mangoldt s D / ((q : ℂ) ^ D.natDegree - 1)) m P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.AsAlgebraHomSurjectiveOfFinrankIntertwiningMapEq

theorem **asAlgebraHom_surjective_of_finrank_intertwiningMap_eq_one** : ∀ (K : Type) [Field K] [CharZero K] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (τ : Representation K G V), Module.finrank K (Representation.IntertwiningMap τ τ) = 1 → Nontrivial V ∧ Function.Surjective τ.asAlgebraHom

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardFixedByPowPrimePowModEq

theorem **card_fixedBy_pow_prime_pow_modEq** (G : Type) [Group G] (X : Type) [Finite X] [MulAction G X] (x : G) (p k : ℕ) (hp : p.Prime) : Nat.card (MulAction.fixedBy X (x ^ p ^ k)) ≡ Nat.card (MulAction.fixedBy X x) [MOD p]

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardFixedByPowPrimePowModEqStatement

theorem **_root_.GlobalTheory.T.card_fixedBy_pow_prime_pow_modEq_card_fixedBy** : ∀ (G : Type) [Group G] (X : Type) [Finite X] [MulAction G X] (x : G) (p k : ℕ), p.Prime → Nat.card (MulAction.fixedBy X (x ^ p ^ k)) ≡ Nat.card (MulAction.fixedBy X x) [MOD p]

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardFixedByQuotientEqRelIndexOfPRegular

theorem **conj_mem_zpowers_of_mem_of_pRegular** {G : Type} [Group G] (c : G) (p : ℕ) (H : Subgroup G) (hc : Nat.Coprime (orderOf c) p) (hp : ∀ h ∈ H, ∃ k : ℕ, h ^ p ^ k ∈ Subgroup.zpowers c) (g : G) (hg : g⁻¹ * c * g ∈ H) : g⁻¹ * c * g ∈ Subgroup.zpowers c

theorem **card_fixedBy_quotient_eq_relIndex_of_pRegular** : ∀ (G : Type) [Group G] [Finite G] (c : G) (p : ℕ) (H N : Subgroup G), Nat.Coprime (orderOf c) p → Subgroup.zpowers c ≤ H → (∀ h ∈ H, ∃ k : ℕ, h ^ p ^ k ∈ Subgroup.zpowers c) → (∀ g : G, g ∈ N ↔ ∀ d ∈ Subgroup.zpowers c, g * d * g⁻¹ ∈ Subgroup.zpowers c) → Nat.card (MulAction.fixedBy (G ⧸ H) c) = H.relIndex N

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardFixedByQuotientEqRelIndexOfPRegularStatement

theorem **_root_.GlobalTheory.T.card_fixedBy_quotient_eq_relIndex_normalizer_of_coprime** : ∀ (G : Type) [Group G] [Finite G] (c : G) (p : ℕ) (H N : Subgroup G), Nat.Coprime (orderOf c) p → Subgroup.zpowers c ≤ H → (∀ h ∈ H, ∃ k : ℕ, h ^ p ^ k ∈ Subgroup.zpowers c) → (∀ g : G, g ∈ N ↔ ∀ d ∈ Subgroup.zpowers c, g * d * g⁻¹ ∈ Subgroup.zpowers c) → Nat.card (MulAction.fixedBy (G ⧸ H) c) = H.relIndex N

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardNonregularResiduesModPowSubOneLe

theorem **card_nonregular_residues_le** (q n : ℕ) (hq : 2 ≤ q) (hn : 0 < n) : ((Finset.Ioo 0 (q ^ n - 1)).filter (fun j => ∃ e ∈ Finset.Ioo 0 n, j * q ^ e ≡ j [MOD q ^ n - 1])).card ≤ ∑ p ∈ n.primeFactors, (q ^ (n / p) - 2)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardNonregularResiduesModPowSubOneLeStatement

theorem **_root_.GlobalTheory.T.card_nonregular_residues_mod_pow_sub_one_le** : ∀ q n : ℕ, 2 ≤ q → 0 < n → ((Finset.Ioo 0 (q ^ n - 1)).filter (fun j => ∃ e ∈ Finset.Ioo 0 n, j * q ^ e ≡ j [MOD q ^ n - 1])).card ≤ ∑ p ∈ n.primeFactors, (q ^ (n / p) - 2)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardSubspacesMulGeneralLinearGroupCard

theorem **card_GL_eq_prod_range** (F : Type) [Field F] [Finite F] (n : ℕ) : Nat.card (GL (Fin n) F) = ∏ i ∈ Finset.range n, (Nat.card F ^ n - Nat.card F ^ i)

theorem **card_subspaces_mul_glCard** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (a : ℕ), a ≤ Module.finrank F V → (Nat.card {W : Submodule F V // Module.finrank F W = a} : ℂ) * (Nat.card F : ℂ) ^ (a * (Module.finrank F V - a)) * glCard F a * glCard F (Module.finrank F V - a) = glCard F (Module.finrank F V)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardSubspacesMulGeneralLinearGroupCardStatement

theorem **_root_.GlobalTheory.T.card_subspaces_mul_generalLinearGroupCard** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (a : ℕ), a ≤ Module.finrank F V → (Nat.card {W : Submodule F V // Module.finrank F W = a} : ℂ) * (Nat.card F : ℂ) ^ (a * (Module.finrank F V - a)) * glCard F a * glCard F (Module.finrank F V - a) = glCard F (Module.finrank F V)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardSubspacesOfFinrankMulCardBases

theorem **card_fibre** {F V : Type} [Field F] [AddCommGroup V] [Module F V] [FiniteDimensional F V] (a : ℕ) (W : Submodule F V) (hW : Module.finrank F W = a) : Nat.card {s : {s : Fin a → V // LinearIndependent F s} // Submodule.span F (Set.range s.1) = W} = Nat.card {t : Fin a → W // LinearIndependent F t}

theorem **card_subspaces_mul** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (a : ℕ), a ≤ Module.finrank F V → Nat.card {W : Submodule F V // Module.finrank F W = a} * ∏ i ∈ Finset.range a, (Nat.card F ^ a - Nat.card F ^ i) = ∏ i ∈ Finset.range a, (Nat.card F ^ Module.finrank F V - Nat.card F ^ i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CardSubspacesOfFinrankMulCardBasesStatement

theorem **_root_.GlobalTheory.T.card_subspaces_of_finrank_mul_card_bases** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (a : ℕ), a ≤ Module.finrank F V → Nat.card {W : Submodule F V // Module.finrank F W = a} * ∏ i ∈ Finset.range a, (Nat.card F ^ a - Nat.card F ^ i) = ∏ i ∈ Finset.range a, (Nat.card F ^ Module.finrank F V - Nat.card F ^ i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharacterCosetSumEqZeroOfSubgroupSum

theorem **_root_.GlobalTheory.BI.char_coset_sum_eq_zero** : ∀ (G : Type) [Group G] [Finite G] (χ : G → ℂ), _root_.GlobalTheory.BI.IsChar ℂ ℂ χ → ∀ H : Subgroup G, ∑ᶠ u : H, χ u = 0 → ∀ g : G, ∑ᶠ u : H, χ (g * u) = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharacterCosetSumEqZeroOfSubgroupSumStatement

theorem **_root_.GlobalTheory.T.character_coset_sum_eq_zero_of_subgroup_sum** : ∀ (G : Type) [Group G] [Finite G] (χ : G → ℂ), IsChar ℂ ℂ χ → ∀ H : Subgroup G, ∑ᶠ u : H, χ u = 0 → ∀ g : G, ∑ᶠ u : H, χ (g * u) = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharacterEqInducedOfImprimitivitySystem

theorem **character_eq_ind_of_imprimitive** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] (ρ : Representation L G V) (T : Subgroup G) (e : Module.End L V), ρ.IsIrreducible → e ≠ 0 → e * e = e → (∀ t ∈ T, ρ t * e = e * ρ t) → (∀ g ∉ T, e * ρ g * e = 0) → ∃ ψ : T → L, IsChar L L ψ ∧ ∀ g, ρ.character g = ind T ψ g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharacterEqInducedOfImprimitivitySystemStatement

theorem **_root_.GlobalTheory.T.character_eq_induced_of_imprimitivity_system** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] (ρ : Representation L G V) (T : Subgroup G) (e : Module.End L V), ρ.IsIrreducible → e ≠ 0 → e * e = e → (∀ t ∈ T, ρ t * e = e * ρ t) → (∀ g ∉ T, e * ρ g * e = 0) → ∃ ψ : T → L, IsChar L L ψ ∧ ∀ g, ρ.character g = ind T ψ g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharacterExtendScalarsRep

theorem **character_extendScalarsRep** (K L : Type) [Field K] [Field L] [Algebra K L] {G : Type} [Group G] {n : ℕ} (ρ : Representation K G (Fin n → K)) (g : G) : (extendScalarsRep K L ρ).character g = algebraMap K L (ρ.character g)

theorem **isChar_baseChange** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (χ : G → L), IsChar K L χ → IsChar L L χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharpolyBlockAndInvariantUnderStabilizerNilradic

theorem **charpoly_eq_restrict_mul_mapQ** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A) : LinearMap.charpoly g = LinearMap.charpoly (g.restrict hg) * LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx))

theorem **charpoly_block** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A), LinearMap.charpoly g = LinearMap.charpoly (g.restrict hg) * LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx)) ∧ ∀ h ∈ nilRad A, LinearMap.charpoly (g + h) = LinearMap.charpoly g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharpolyBlockAndInvariantUnderStabilizerNilradicStatement

theorem **_root_.GlobalTheory.T.charpoly_block_and_invariant_under_stabilizerNilradical** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A), LinearMap.charpoly g = LinearMap.charpoly (g.restrict hg) * LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx)) ∧ ∀ h ∈ nilRad A, LinearMap.charpoly (g + h) = LinearMap.charpoly g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharpolyEndomorphismDensityGLRecursion

theorem **cyclic_conj** {F : Type} [Field F] {V V' : Type} [AddCommGroup V] [Module F V] [AddCommGroup V'] [Module F V'] (e : V ≃ₗ[F] V') (g : Module.End F V) (v : V) : (∀ U : Submodule F V, (∀ x ∈ U, g x ∈ U) → v ∈ U → U = ⊤) ↔ (∀ U : Submodule F V', (∀ x ∈ U, (e.conj g) x ∈ U) → e v ∈ U → U = ⊤)

theorem **glCard_ne_zero_of_finite** (F : Type) [Field F] [Finite F] (n : ℕ) : glCard F n ≠ 0

theorem **mass_const** {F : Type} [Field F] [Finite F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (c : ℂ) (P : F[X]) : mass (fun _ : Module.End F V => c) P = c * mass (fun _ : Module.End F V => (1 : ℂ)) P

theorem **mu_one** (F : Type) [Field F] : mu (1 : F[X]) = 1

theorem **mu_rec** : ∀ (F : Type) [Field F] [Finite F] (P : F[X]), P.Monic → (Nat.card F : ℂ) ^ P.natDegree * mu P = conv (fun _ => 1) mu P ∧ mu (1 : F[X]) = 1

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharpolyEndomorphismDensityGLRecursionStatement

theorem **_root_.GlobalTheory.T.charpolyEndomorphismDensityGL_recursion** : ∀ (F : Type) [Field F] [Finite F] (P : F[X]), P.Monic → (Nat.card F : ℂ) ^ P.natDegree * mu P = conv (fun _ => 1) mu P ∧ mu (1 : F[X]) = 1

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharpolyEqFinsumPsi

theorem **charpoly_eq_finsum_psi** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (g : Module.End F V) : s (LinearMap.charpoly g) = ∑ᶠ W : {W : Submodule F V // ∀ x ∈ W, g x ∈ W}, psi s W.1 (g.restrict W.2)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CharpolyEqFinsumPsiStatement

theorem **_root_.GlobalTheory.T.apply_charpoly_eq_finsum_psi_invariant_submodules** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (g : Module.End F V), s (LinearMap.charpoly g) = ∑ᶠ W : {W : Submodule F V // ∀ x ∈ W, g x ∈ W}, psi s W.1 (g.restrict W.2)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CommuteImageAbelianByPrimePowerOfNormalCentral

theorem **_root_.GlobalTheory.BI.commute_all** : ∀ (G : Type) [Group G] [Finite G] (M : Type) [Monoid M] (φ : G →* M), _root_.GlobalTheory.BI.AbByP G → (∀ B : Subgroup G, B.Normal → (∀ a ∈ B, ∀ b ∈ B, φ a * φ b = φ b * φ a) → ∀ b ∈ B, ∀ g : G, φ b * φ g = φ g * φ b) → ∀ a b : G, φ a * φ b = φ b * φ a

Development.AbelianVarietyTheory.Proofs.CharacterTheory.CommuteImageAbelianByPrimePowerOfNormalCentralStatement

theorem **_root_.GlobalTheory.T.commute_image_abelianByPrimePower_of_normal_central** : ∀ (G : Type) [Group G] [Finite G] (M : Type) [Monoid M] (φ : G →* M), AbByP G → (∀ B : Subgroup G, B.Normal → (∀ a ∈ B, ∀ b ∈ B, φ a * φ b = φ b * φ a) → ∀ b ∈ B, ∀ g : G, φ b * φ g = φ g * φ b) → ∀ a b : G, φ a * φ b = φ b * φ a

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ConjOfPrimaryFinrankKerPowEq

theorem **exists_conj_of_aeval'_equiv** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] (g g' : Module.End F V) (φ : Module.AEval' g ≃ₗ[F[X]] Module.AEval' g') : ∃ e : V ≃ₗ[F] V, e.conj g = g'

theorem **primary_conj_of_finrank_ker_eq** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g g' : Module.End F V) (f : F[X]) (m : ℕ), Irreducible f → aeval g (f ^ m) = 0 → aeval g' (f ^ m) = 0 → (∀ j : ℕ, Module.finrank F (LinearMap.ker (aeval g (f ^ j))) = Module.finrank F (LinearMap.ker (aeval g' (f ^ j)))) → ∃ e : V ≃ₗ[F] V, e.conj g = g'

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ConjOfPrimaryFinrankKerPowEqStatement

theorem **_root_.GlobalTheory.T.conj_of_primary_finrank_ker_pow_eq** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g g' : Module.End F V) (f : F[X]) (m : ℕ), Irreducible f → aeval g (f ^ m) = 0 → aeval g' (f ^ m) = 0 → (∀ j : ℕ, Module.finrank F (LinearMap.ker (aeval g (f ^ j))) = Module.finrank F (LinearMap.ker (aeval g' (f ^ j)))) → ∃ e : V ≃ₗ[F] V, e.conj g = g'

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassEqOfLinearEquivInvariant

theorem **mass_eq_mass_pi_of_invariant** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (Φ : ∀ (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V], Module.End F V → ℂ), (∀ (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), Φ V' (e.conj g) = Φ V g) → ∀ P : F[X], P.natDegree = Module.finrank F V → mass (Φ V) P = mass (Φ (Fin P.natDegree → F)) P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassEqOfLinearEquivInvariantStatement

theorem **_root_.GlobalTheory.T.endomorphismMass_eq_of_linearEquiv_invariant** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (Φ : ∀ (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V], Module.End F V → ℂ), (∀ (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), Φ V' (e.conj g) = Φ V g) → ∀ P : F[X], P.natDegree = Module.finrank F V → mass (Φ V) P = mass (Φ (Fin P.natDegree → F)) P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassMobiusInverseMonicEq

theorem **mass_comp_charpoly_eq_mul** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (P : F[X]) : mass (fun g : Module.End F V => s (LinearMap.charpoly g)) P = s P * mass (fun _ : Module.End F V => (1 : ℂ)) P

theorem **mass_psi_eq_glCard_mul_mangoldt_div** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → ∀ P : F[X], P.Monic → P.natDegree = Module.finrank F V → mass (psi s V) P = glCard F P.natDegree * mangoldt s P / ((Nat.card F : ℂ) ^ P.natDegree - 1)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassMobiusInverseMonicEqStatement

theorem **_root_.GlobalTheory.T.endomorphismMass_mobiusInverse_monic_eq** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → ∀ P : F[X], P.Monic → P.natDegree = Module.finrank F V → mass (psi s V) P = glCard F P.natDegree * mangoldt s P / ((Nat.card F : ℂ) ^ P.natDegree - 1)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassMulParabolicInductionTrivial

theorem **finsum_subtype_dite** {α : Type} [Fintype α] (p : α → Prop) (f : {x // p x} → ℂ) : ∑ᶠ x, f x = ∑ x, if h : p x then f ⟨x, h⟩ else 0

theorem **mass_eq** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ) (P : F[X]) : mass φ P = ∑ᶠ g, if LinearMap.charpoly g = P then φ g else 0

theorem **finite_pairs** {F : Type} [Field F] (P : F[X]) (hP : P.Monic) : {x : F[X] × F[X] | x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P}.Finite

theorem **finite_end** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] : Finite (Module.End F V)

theorem **mass_eq_zero** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ) (P : F[X]) (h : Module.finrank F V ≠ P.natDegree) : mass φ P = 0

theorem **indicator_split** {F : Type} [Field F] (P : F[X]) (T : Finset (F[X] × F[X])) (hT : ∀ x, x ∈ T ↔ x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P) {a b c : F[X]} (ha : a.Monic) (hb : b.Monic) (hc : c = a * b) (z : ℂ) : (if c = P then z else 0) = ∑ x ∈ T, (if a = x.1 then z else 0) * (if b = x.2 then (1 : ℂ) else 0)

theorem **stab_sum** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (W : Submodule F V) (φ : Module.End F W → ℂ) (P : F[X]) (T : Finset (F[X] × F[X])) (hT : ∀ x, x ∈ T ↔ x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P) : ∑ᶠ g : {g : Module.End F V // ∀ x ∈ W, g x ∈ W}, (if LinearMap.charpoly g.1 = P then φ (g.1.restrict g.2) else 0) = ∑ x ∈ T, (Nat.card F : ℂ) ^ (Module.finrank F W * Module.finrank F (V ⧸ W)) * (mass φ x.1 * mass (fun _ : Module.End F (V ⧸ W) => (1 : ℂ)) x.2)

theorem **glCard_ne_zero** (F : Type) [Field F] [Finite F] (n : ℕ) : glCard F n ≠ 0

theorem **mass_mult** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (Φ : ∀ (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V], Module.End F V → ℂ), (∀ (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), Φ V' (e.conj g) = Φ V g) → ∀ P : F[X], P.Monic → P.natDegree = Module.finrank F V → mass (fun g : Module.End F V => ∑ᶠ W : {W : Submodule F V // ∀ x ∈ W, g x ∈ W}, Φ W.1 (g.restrict W.2)) P / glCard F P.natDegree = conv (fun D => mass (Φ (Fin D.natDegree → F)) D / glCard F D.natDegree) mu P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassMulParabolicInductionTrivialStatement

theorem **_root_.GlobalTheory.T.endomorphismMass_mul_parabolic_induction_trivial** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (Φ : ∀ (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V], Module.End F V → ℂ), (∀ (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), Φ V' (e.conj g) = Φ V g) → ∀ P : F[X], P.Monic → P.natDegree = Module.finrank F V → mass (fun g : Module.End F V => ∑ᶠ W : {W : Submodule F V // ∀ x ∈ W, g x ∈ W}, Φ W.1 (g.restrict W.2)) P / glCard F P.natDegree = conv (fun D => mass (Φ (Fin D.natDegree → F)) D / glCard F D.natDegree) mu P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassOfCharpolyCardCyclicVectors

theorem **charpoly_mulLeft_root** (D : F[X]) (hD : D.Monic) [FiniteDimensional F (AdjoinRoot D)] : LinearMap.charpoly (LinearMap.mulLeft F (AdjoinRoot.root D)) = D

theorem **map_mk** {V : Type} [AddCommGroup V] [Module F V] (D : F[X]) (f : AdjoinRoot D →ₗ[F] V) (g : Module.End F V) (h : ∀ a, f (AdjoinRoot.root D * a) = g (f a)) (p : F[X]) : f (AdjoinRoot.mk D p) = aeval g p (f 1)

theorem **cyclic_one** (D : F[X]) (U : Submodule F (AdjoinRoot D)) (hU : ∀ a ∈ U, AdjoinRoot.root D * a ∈ U) (h1 : (1 : AdjoinRoot D) ∈ U) : U = ⊤

theorem **card_pairs** {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (D : F[X]) (hD : D.Monic) (hV : Module.finrank F V = D.natDegree) : Nat.card (AdjoinRoot D ≃ₗ[F] V) = Nat.card (Σ g : {g : Module.End F V // LinearMap.charpoly g = D}, {v : V // ∀ U : Submodule F V, (∀ x ∈ U, g.1 x ∈ U) → v ∈ U → U = ⊤})

theorem **mass_card_cyclic** (F : Type) [Field F] [Finite F] (D : F[X]) (hD : D.Monic) : mass (fun g : Module.End F (Fin D.natDegree → F) => (Nat.card {v : Fin D.natDegree → F // ∀ U : Submodule F (Fin D.natDegree → F), (∀ x ∈ U, g x ∈ U) → v ∈ U → U = ⊤} : ℂ)) D = glCard F D.natDegree

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EndomorphismMassOfCharpolyCardCyclicVectorsStatement

theorem **_root_.GlobalTheory.T.endomorphismMassOfCharpoly_card_cyclic_vectors** : ∀ (F : Type) [Field F] [Finite F] (D : F[X]), D.Monic → mass (fun g : Module.End F (Fin D.natDegree → F) => (Nat.card {v : Fin D.natDegree → F // ∀ U : Submodule F (Fin D.natDegree → F), (∀ x ∈ U, g x ∈ U) → v ∈ U → U = ⊤} : ℂ)) D = glCard F D.natDegree

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EqOfIsPrimitiveOfEqOnEllipticFiniteGL

theorem **_root_.GlobalTheory.CuspGL.isCharacter_of_isIrreducibleCharacter_complex** {G : Type} [Group G] {χ : G → ℂ} (h : _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ) : _root_.GlobalTheory.BI.IsChar ℂ ℂ χ

theorem **_root_.GlobalTheory.CuspGL.generalLinearGroupCardComplex_finiteField_ne_zero** (F : Type) [Field F] [Finite F] (d : ℕ) : _root_.GlobalTheory.CuspGL.glCard F d ≠ 0

theorem **_root_.GlobalTheory.CuspGL.sum_coef_mul_eq_pair** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (d : ℕ) (hd : 0 < d) (χ χ' : GL (Fin d) F → ℂ) (h : _root_.GlobalTheory.BI.IsChar ℂ ℂ χ) (h' : _root_.GlobalTheory.BI.IsChar ℂ ℂ χ') (hp : _root_.GlobalTheory.CuspGL.IsPrimitive (_root_.GlobalTheory.CuspGL.extZero χ)) (hp' : _root_.GlobalTheory.CuspGL.IsPrimitive (_root_.GlobalTheory.CuspGL.extZero χ')) (a b : ℕ → ℤ) (ha : ∀ k : ℕ, _root_.GlobalTheory.CuspGL.ipGL (_root_.GlobalTheory.CuspGL.extZero χ) (fun g : Module.End F (Fin d → F) => _root_.GlobalTheory.CuspGL.rootSum θ k (LinearMap.charpoly g)) = a k * _root_.GlobalTheory.CuspGL.glCard F d) (hb : ∀ k : ℕ, _root_.GlobalTheory.CuspGL.ipGL (_root_.GlobalTheory.CuspGL.extZero χ') (fun g : Module.End F (Fin d → F) => _root_.GlobalTheory.CuspGL.rootSum θ

private theorem **charpoly_toLin''** {F : Type} [Field F] {d : ℕ} (M : Matrix (Fin d) (Fin d) F) : (Matrix.toLin' M).charpoly = M.charpoly

private theorem **sum_range_apply_mul_mod_eq_of_coprime** {m u : ℕ} (hu : Nat.Coprime u m) (c : ℕ → ℂ) : ∑ k ∈ Finset.range m, c (k * u % m) = ∑ k ∈ Finset.range m, c k

private theorem **character_elliptic_eq_coeff_sum_lemma** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (d : ℕ) (hd : 0 < d) (χ : GL (Fin d) F → ℂ) (hχ : _root_.GlobalTheory.BI.IsChar ℂ ℂ χ) (hp : _root_.GlobalTheory.CuspGL.IsPrimitive (_root_.GlobalTheory.CuspGL.extZero χ)) (a : ℕ → ℤ) (ha : ∀ k : ℕ, _root_.GlobalTheory.CuspGL.ipGL (_root_.GlobalTheory.CuspGL.extZero χ) (fun g : Module.End F (Fin d → F) => _root_.GlobalTheory.CuspGL.rootSum θ k (LinearMap.charpoly g)) = a k * _root_.GlobalTheory.CuspGL.glCard F d) {ζ : AlgebraicClosure F} (hω : IsPrimitiveRoot (θ ζ) (Nat.card F ^ d - 1)) (j : ℕ) (x : GL (Fin d) F) (hx : (x : Matrix (Fin d) (Fin d) F).charpoly = minpoly F (ζ ^ j)) (hdeg : (minpoly F (ζ ^ j)).natDegree = d) : χ x = ∑ k ∈ Finset.range (Nat.card F ^ d - 1), (a k : ℂ) * θ ζ ^ (j * k)

theorem **_root_.GlobalTheory.CuspGL.eq_of_isPrimitive_of_eq_on_elliptic** (F : Type) [Field F] [Finite F] (d : ℕ) (χ χ' : GL (Fin d) F → ℂ) (hd : 0 < d) (hi : _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ) (hi' : _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ') (hp : _root_.GlobalTheory.CuspGL.IsPrimitive (_root_.GlobalTheory.CuspGL.extZero χ)) (hp' : _root_.GlobalTheory.CuspGL.IsPrimitive (_root_.GlobalTheory.CuspGL.extZero χ')) (he : ∀ x : GL (Fin d) F, Irreducible (Matrix.charpoly (x : Matrix (Fin d) (Fin d) F)) → χ x = χ' x) : χ = χ'

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EqOfIsPrimitiveOfEqOnEllipticFiniteGLStatement

theorem **_root_.GlobalTheory.T.eq_of_isPrimitive_of_eq_on_elliptic_finiteGL** : ∀ (F : Type) [Field F] [Finite F] (d : ℕ) (χ χ' : GL (Fin d) F → ℂ), 0 < d → IsIrrChar ℂ ℂ χ → IsIrrChar ℂ ℂ χ' → IsPrimitive (extZero χ) → IsPrimitive (extZero χ') → (∀ x : GL (Fin d) F, Irreducible (Matrix.charpoly (x : Matrix (Fin d) (Fin d) F)) → χ x = χ' x) → χ = χ'

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EqZeroOfFinsumMulStarPowEqZero

theorem **eq_zero_of_finsum_mul_star_pow_eq_zero** {K : Type} [Field K] (θ : K →* ℂ) (hθ : ∀ x y : K, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (N : ℕ) (hN : 0 < N) (c : K → ℂ) (h : ∀ l : ℕ, ∑ᶠ α ∈ {α : K | α ^ N = 1}, c α * star (θ α ^ l) = 0) : ∀ α ∈ {α : K | α ^ N = 1}, c α = 0

theorem **torus_fourier_injective** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ n : ℕ, 0 < n → ∀ c : AlgebraicClosure F → ℂ, (∀ l : ℕ, ∑ᶠ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n - 1) = 1}, c α * star (θ α ^ l) = 0) → ∀ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n - 1) = 1}, c α = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EqZeroOfNotPrimary

theorem **irreducible_dvd_of_dvd_charpoly_of_aeval_eq_zero** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (T : Module.End F V) (p q : F[X]) (hp : Irreducible p) (hd : p ∣ LinearMap.charpoly T) (hq : aeval T q = 0) : p ∣ q

theorem **eq_zero_of_not_primary** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → 0 < Module.finrank F V → ∀ g : Module.End F V, (¬ ∃ (f : F[X]) (m : ℕ), f.Monic ∧ Irreducible f ∧ 0 < m ∧ LinearMap.charpoly g = f ^ m) → φ g = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EqZeroOfNotPrimaryStatement

theorem **_root_.GlobalTheory.T.classFn_eq_zero_of_charpoly_not_irreducible_pow** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → 0 < Module.finrank F V → ∀ g : Module.End F V, (¬ ∃ (f : F[X]) (m : ℕ), f.Monic ∧ Irreducible f ∧ 0 < m ∧ LinearMap.charpoly g = f ^ m) → φ g = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.EssFiniteTypeOfFiniteDimensionalSubfieldClosure

theorem **essFiniteType_of_finiteDimensional_subfield_closure** {o Om : Type*} [CommRing o] [IsDomain o] [Field Om] [Algebra o Om] [FaithfulSMul o Om] {T : Set Om} (hT : T.Finite) (h : FiniteDimensional (Subfield.closure (Set.range (algebraMap o Om) ∪ T)) Om) : Algebra.EssFiniteType o Om

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsAbelianByPrimePowerNotDvdCardFixedBy

theorem **exists_abelianByPrimePower_not_dvd_card_fixedBy₂** : ∀ (G : Type) [Group G] [Finite G] (x : G) (p : ℕ), p.Prime → ∃ H : Subgroup G, AbByP H ∧ ¬ p ∣ Nat.card (MulAction.fixedBy (G ⧸ H) x)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsAbelianByPrimePowerNotDvdCardFixedByStatement

theorem **_root_.GlobalTheory.T.exists_abelianByPrimePower_not_dvd_card_fixedBy** : ∀ (G : Type) [Group G] [Finite G] (x : G) (p : ℕ), p.Prime → ∃ H : Subgroup G, AbByP H ∧ ¬ p ∣ Nat.card (MulAction.fixedBy (G ⧸ H) x)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsCharacterEigenvectorOfCommutingRepresentat

theorem **exists_common_eigenvector_of_commute** {L : Type} [Field L] [IsAlgClosed L] {ι : Type} {V : Type} [AddCommGroup V] [Module L V] [FiniteDimensional L V] [Nontrivial V] (f : ι → Module.End L V) (hf : ∀ a b, f a ∗ f b = f b ∗ f a) : ∃ v : V, v ≠ 0 ∧ ∀ b, ∃ μ : L, f b v = μ • v

theorem **exists_character_eigenvector_of_commute** : ∀ (L : Type) [Field L] [IsAlgClosed L] (B : Type) [Group B] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] [Nontrivial V] (ρ : Representation L B V), (∀ a b, ρ a ∗ ρ b = ρ b ∗ ρ a) → ∃ (lam : B →* Lˣ) (v : V), v ≠ 0 ∧ ∀ b, ρ b v = (lam b : L) • v

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsCharacterEigenvectorOfCommutingRepresentatStatement

theorem **_root_.GlobalTheory.T.exists_character_eigenvector_of_commuting_representation** : ∀ (L : Type) [Field L] [IsAlgClosed L] (B : Type) [Group B] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] [Nontrivial V] (ρ : Representation L B V), (∀ a b, ρ a ∗ ρ b = ρ b ∗ ρ a) → ∃ (lam : B →* Lˣ) (v : V), v ≠ 0 ∧ ∀ b, ρ b v = (lam b : L) • v

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsCharacterQuotientOfInjectiveIntertwining

theorem **exists_isChar_eq_add_of_injective** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (ρ : Representation K G V) (σ : Representation K G W) (f : Representation.IntertwiningMap σ ρ), Function.Injective f → ∃ c : G → L, IsChar K L c ∧ ∀ g, algebraMap K L (ρ.character g) = algebraMap K L (σ.character g) + c g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsCharacterQuotientOfInjectiveIntertwiningStatement

theorem **_root_.GlobalTheory.T.exists_character_quotient_of_injective_intertwining** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (ρ : Representation K G V) (σ : Representation K G W) (f : Representation.IntertwiningMap σ ρ), Function.Injective f → ∃ c : G → L, IsChar K L c ∧ ∀ g, algebraMap K L (ρ.character g) = algebraMap K L (σ.character g) + c g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsConjEqAddOfIsCoprime

theorem **intertwine_aeval_apply** {F : Type} [Field F] {V W : Type} [AddCommGroup V] [Module F V] [AddCommGroup W] [Module F W] (f : V →ₗ[F] W) (a : Module.End F V) (b : Module.End F W) (h : ∀ x, f (a x) = b (f x)) (p : F[X]) (x : V) : f (aeval a p x) = aeval b p (f x)

theorem **nilRad_eq_zero_of_commute** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A) (hc : IsCoprime (LinearMap.charpoly (g.restrict hg)) (LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx)))) (u : Module.End F V) (hu : u ∈ nilRad A) (hug : ∀ x, u (g x) = g (u x)) : u = 0

theorem **exists_conj_eq_add_of_isCoprime** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A), IsCoprime (LinearMap.charpoly (g.restrict hg)) (LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx))) → ∀ h ∈ nilRad A, ∃ e : V ≃ₗ[F] V, e.conj g = g + h

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsCoprimeOrderOfPowPrimePow

theorem **exists_coprime_orderOf_pow_prime_pow** (G : Type) [Group G] [Finite G] (x : G) (p : ℕ) (hp : p.Prime) : ∃ k : ℕ, Nat.Coprime (orderOf (x ^ p ^ k)) p

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsCoprimeOrderOfPowPrimePowStatement

theorem **_root_.GlobalTheory.T.exists_coprime_orderOf_pow_prime_pow'** : ∀ (G : Type) [Group G] [Finite G] (x : G) (p : ℕ), p.Prime → ∃ k : ℕ, Nat.Coprime (orderOf (x ^ p ^ k)) p

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsHolomorphicFamily

theorem **exists_basis_away** {R S : Type} [CommRing R] [CommRing S] [Algebra R S] [Algebra.Smooth R S] (p : Ideal S) [p.IsPrime] : ∃ (g : S) (m : ℕ) (s : Fin m → S) (b : Module.Basis (Fin m) (Localization.Away g) Ω[Localization.Away g⁄R]), g ∉ p ∧ ∀ i, b i = D R (Localization.Away g) (algebraMap S _ (s i))

theorem **exists_etale_away** {R S : Type} [CommRing R] [CommRing S] [Algebra R S] [Algebra.Smooth R S] (p : Ideal S) [p.IsPrime] : ∃ (g : S) (m : ℕ) (s : Fin m → S), g ∉ p ∧ (MvPolynomial.eval₂Hom (algebraMap R (Localization.Away g)) fun i => algebraMap S (Localization.Away g) (s i)).Etale

theorem **exists_finset_of_mem_nhds** {A : Type} {x : A → ℂ} {U : Set (A → ℂ)} (hU : U ∈ 𝓝 x) : ∃ (S : Finset A) (ε : ℝ), 0 < ε ∧ ∀ y : A → ℂ, (∀ a ∈ S, ‖y a − x a‖ < ε) → y ∈ U

theorem **exists_holomorphic_family** : ∀ (K A : Type) [Field K] [CharZero K] [CommRing A] [Algebra K A] [Algebra.Smooth K A] (σ : K →+* ℂ) (φ₀ : A →+* ℂ), φ₀.comp (algebraMap K A) = σ → ∃ (m : ℕ) (s : Fin m → A) (B : Set (Fin m → ℂ)) (φ : (Fin m → ℂ) → (A →+* ℂ)) (S : Finset A) (ε : ℝ), IsOpen B ∧ 0 < ε ∧ (∀ b ∈ B, (φ b).comp (algebraMap K A) = σ ∧ ∀ i, φ b (s i) = b i) ∧ (∀ a : A, DifferentiableOn ℂ (fun b => φ b a) B) ∧ ∀ ψ : A →+* ℂ, ψ.comp (algebraMap K A) = σ → (∀ a ∈ S, ‖ψ a − φ₀ a‖ < ε) → (fun i => ψ (s i)) ∈ B ∧ φ (fun i => ψ (s i)) = ψ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsHolomorphicFamilyStatement

theorem **_root_.GlobalTheory.T.exists_holomorphic_family_of_ringHom_near** : ∀ (K A : Type) [Field K] [CharZero K] [CommRing A] [Algebra K A] [Algebra.Smooth K A] (σ : K →+* ℂ) (φ₀ : A →+* ℂ), φ₀.comp (algebraMap K A) = σ → ∃ (m : ℕ) (s : Fin m → A) (B : Set (Fin m → ℂ)) (φ : (Fin m → ℂ) → (A →+* ℂ)) (S : Finset A) (ε : ℝ), IsOpen B ∧ 0 < ε ∧ (∀ b ∈ B, (φ b).comp (algebraMap K A) = σ ∧ ∀ i, φ b (s i) = b i) ∧ (∀ a : A, DifferentiableOn ℂ (fun b => φ b a) B) ∧ ∀ ψ : A →+* ℂ, ψ.comp (algebraMap K A) = σ → (∀ a ∈ S, ‖ψ a − φ₀ a‖ < ε) → (fun i => ψ (s i)) ∈ B ∧ φ (fun i => ψ (s i)) = ψ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsInducedIntPrimeTo

private theorem **exists_commute_coprime_mul_prime_power_order** {G : Type} [Group G] [Finite G] (x : G) {p : ℕ} (hp : p.Prime) : ∃ c q : G, x = c * q ∧ Commute c q ∧ Nat.Coprime p (orderOf c) ∧ ∃ a : ℕ, q ^ p ^ a = 1

private theorem **zpow_mul_zpow_eq_of_commute_of_zpow_eq** {G : Type} [Group G] {c w : G} (hcw : Commute c w) {M : ℤ} (hc : c ^ M = c) (hw : w ^ M = 1) (i : ℤ) : (c ^ i * w) ^ M = c ^ i

private theorem **inv_mul_mul_zpow** {G : Type} [Group G] (s a : G) (n : ℤ) : (s⁻¹ * a * s) ^ n = s⁻¹ * a ^ n * s

theorem **_root_.GlobalTheory.exists_induced_int_prime_to** : ∀ (G : Type) [Group G] [Finite G] (x : G) (p : ℕ), p.Prime → ∃ (E : Subgroup G) (f : G → ℤ) (ι : Type) (_ : Fintype ι) (r : ι → ℂ) (lam : ι → E →* ℂˣ), (∃ l : ℕ, l.Prime ∧ ∀ y : E, ∃ j : ℕ, y ^ l ^ j ∈ Subgroup.center E) ∧ ¬ (p : ℤ) ∣ f x ∧ (∀ i, IsIntegral ℤ (r i)) ∧ ∀ g, (f g : ℂ) = ∑ i, r i * _root_.GlobalTheory.BI.ind E (fun e => (lam i e : ℂ)) g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsInducedIntPrimeToStatement

theorem **_root_.GlobalTheory.T.exists_integral_comb_induced_linear_not_dvd** : ∀ (G : Type) [Group G] [Finite G] (x : G) (p : ℕ), p.Prime → ∃ (E : Subgroup G) (f : G → ℤ) (ι : Type) (_ : Fintype ι) (r : ι → ℂ) (lam : ι → E →* ℂˣ), (∃ l : ℕ, l.Prime ∧ ∀ y : E, ∃ j : ℕ, y ^ l ^ j ∈ Subgroup.center E) ∧ ¬ (p : ℤ) ∣ f x ∧ (∀ i, IsIntegral ℤ (r i)) ∧ ∀ g, (f g : ℂ) = ∑ i, r i * ind E (fun e => (lam i e : ℂ)) g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIntCoeffPrimitiveGLCharacter

theorem **exists_int_coeff_of_isPrimitive** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (d : ℕ) (χ : GL (Fin d) F → ℂ) (hd : 0 < d) (hχ : IsChar ℂ ℂ χ) (hp : IsPrimitive (extZero χ)) : ∃ a : ℕ → ℤ, (∀ k : ℕ, ipGL (extZero χ) (fun g : Module.End F (Fin d → F) => rootSum θ k (LinearMap.charpoly g)) = a k * glCard F d) ∧ (d : ℂ) * χ 1 = (∑ k ∈ Finset.range (Nat.card F ^ d − 1), (a k : ℂ)) * ∏ i ∈ Finset.Ico 1 d, (1 − (Nat.card F : ℂ) ^ i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIntCoeffPrimitiveGLCharacterStatement

theorem **_root_.GlobalTheory.T.exists_int_coeff_primitive_GL_character** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (d : ℕ) (χ : GL (Fin d) F → ℂ), 0 < d → IsChar ℂ ℂ χ → IsPrimitive (extZero χ) → ∃ a : ℕ → ℤ, (∀ k : ℕ, ipGL (extZero χ) (fun g : Module.End F (Fin d → F) => rootSum θ k (LinearMap.charpoly g)) = a k * glCard F d) ∧ (d : ℂ) * χ 1 = (∑ k ∈ Finset.range (Nat.card F ^ d − 1), (a k : ℂ)) * ∏ i ∈ Finset.Ico 1 d, (1 − (Nat.card F : ℂ) ^ i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIntRelationOfIsIntegralRelation

theorem **exists_int_eq_of_isIntegral_of_mul_eq_int** {x : ℂ} (hx : IsIntegral ℤ x) {n k : ℤ} (hn : n ≠ 0) (h : (n : ℂ) * x = k) : ∃ j : ℤ, (j : ℂ) = x

theorem **exists_int_retraction_of_isIntegral** : ∀ (ι : Type) [Fintype ι] (r : ι → ℂ), (∀ i, IsIntegral ℤ (r i)) → ∃ c : ι → ℤ, ∀ (m : ι → ℤ) (d : ℤ), ∑ i, r i * (m i : ℂ) = d → ∑ i, c i * m i = d

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIntRelationOfIsIntegralRelationStatement

theorem **_root_.GlobalTheory.T.exists_int_relation_of_isIntegral_relation** : ∀ (ι : Type) [Fintype ι] (r : ι → ℂ), (∀ i, IsIntegral ℤ (r i)) → ∃ c : ι → ℤ, ∀ (m : ι → ℤ) (d : ℤ), ∑ i, r i * (m i : ℂ) = d → ∑ i, c i * m i = d

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIrrCharAdd

theorem **exists_irreducible_subrepresentation** {K G V : Type} [Field K] [Group G] [AddCommGroup V] [Module K V] [FiniteDimensional K V] [Nontrivial V] (ρ : Representation K G V) : ∃ S : Subrepresentation ρ, S.toRepresentation.IsIrreducible

theorem **exists_irrChar_add** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (a : G → L), IsChar K L a → a ≠ 0 → ∃ φ c : G → L, IsIrrChar K L φ ∧ IsChar K L c ∧ a = φ + c

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIrrCharAddStatement

theorem **_root_.GlobalTheory.T.exists_isIrrChar_add_isChar_of_ne_zero** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (a : G → L), IsChar K L a → a ≠ 0 → ∃ φ c : G → L, IsIrrChar K L φ ∧ IsChar K L c ∧ a = φ + c

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIsCharEqAddOfPairNeZero

theorem **exists_isChar_eq_add_of_pair_ne_zero** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (φ a : G → L), IsIrrChar K L φ → IsChar K L a → pair a φ ≠ 0 → ∃ c : G → L, IsChar K L c ∧ a = φ + c

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsIsCharEqAddOfPairNeZeroStatement

theorem **_root_.GlobalTheory.T.exists_isChar_eq_add_of_pair_ne_zero'** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (φ a : G → L), IsIrrChar K L φ → IsChar K L a → pair a φ ≠ 0 → ∃ c : G → L, IsChar K L c ∧ a = φ + c

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsMonomialBasisAbelianByPrimePower

theorem **exists_monomial_basis_of_isCompl** {k : Type} [Field k] {G : Type} [Group G] {V : Type} [AddCommGroup V] [Module k V] (ρ : Representation k G V) (U U' : Submodule k V) (h : IsCompl U U') (hU : ∃ (n : ℕ) (b : Module.Basis (Fin n) k U), ∀ (g : G) (i : Fin n), ∃ (j : Fin n) (c : k), ρ g (b i : V) = c • (b j : V)) (hU' : ∃ (n : ℕ) (b : Module.Basis (Fin n) k U'), ∀ (g : G) (i : Fin n), ∃ (j : Fin n) (c : k), ρ g (b i : V) = c • (b j : V)) : ∃ (n : ℕ) (b : Module.Basis (Fin n) k V), ∀ (g : G) (i : Fin n), ∃ (j : Fin n) (c : k), ρ g (b i) = c • b j

theorem **exists_monomial_basis_of_abByP** : ∀ (k : Type) [Field k] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module k V] [FiniteDimensional k V] (ρ : Representation k G V) [IsAlgClosed k], AbByP G → (Nat.card G : k) ≠ 0 → ∃ (n : ℕ) (b : Module.Basis

(Fin n) k V), ∀ (g : G) (i : Fin n), ∃ (j : Fin n) (c : k), ρ g (b i) = c • b j

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsMonomialBasisAbelianByPrimePowerStatement

theorem **_root_.GlobalTheory.T.exists_monomial_basis_abelianByPrimePower** : ∀ (k : Type) [Field k] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module k V] [FiniteDimensional k V] (ρ : Representation k G V) [IsAlgClosed k], AbByP G → (Nat.card G : k) ≠ 0 → ∃ (n : ℕ) (b : Module.Basis (Fin n) k V), ∀ (g : G) (i : Fin n), ∃ (j : Fin n) (c : k), ρ g (b i) = c • b j

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsMonomialBasisOfImprimitive

theorem **exists_monomial_basis_of_imprimitive** (k : Type) [Field k] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module k V] [FiniteDimensional k V] (ρ : Representation k G V) (T : Subgroup G) (W : Submodule k V) (hind : iSupIndep (fun q : G ⧸ T => W.map (ρ q.out))) (hsup : ⨆ q : G ⧸ T, W.map (ρ q.out) = ⊤) (hW : ∃ (n : ℕ) (b : Module.Basis (Fin n) k W), ∀ t ∈ T, ∀ i : Fin n, ∃ (j : Fin n) (c : k), ρ t (b i : V) = c • (b j : V)) : ∃ (n : ℕ) (b : Module.Basis (Fin n) k V), ∀ (g : G) (i : Fin n), ∃ (j : Fin n) (c : k), ρ g (b i) = c • b j

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsPermOfSumMinEqPositiveFamily

theorem **sum_min_succ_eq_add_card** {d : ℕ} (e : Fin d → ℕ) (j : ℕ) : ∑ i, min (j + 1) (e i) = ∑ i, min j (e i) + (Finset.univ.filter (fun i => j < e i)).card

theorem **card_gt_eq_card_eq_add_card_gt** {d : ℕ} (e : Fin d → ℕ) (j : ℕ) : (Finset.univ.filter (fun i => j < e i)).card = (Finset.univ.filter (fun i => e i = j + 1)).card + (Finset.univ.filter (fun i => j + 1 < e i)).card

theorem **exists_equiv_of_sum_min_eq** : ∀ (d d' : ℕ) (e : Fin d → ℕ) (e' : Fin d' → ℕ), (∀ i, 0 < e i) → (∀ i, 0 < e' i) → (∀ j : ℕ, ∑ i, min j (e i) = ∑ i, min j (e' i)) → ∃ σ : Fin d ≃ Fin d', ∀ i, e' (σ i) = e i

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsPermOfSumMinEqPositiveFamilyStatement

theorem **_root_.GlobalTheory.T.exists_perm_of_sum_min_eq_positive_family** : ∀ (d d' : ℕ) (e : Fin d → ℕ) (e' : Fin d' → ℕ), (∀ i, 0 < e i) → (∀ i, 0 < e' i) → (∀ j : ℕ, ∑ i, min j (e i) = ∑ i, min j (e' i)) → ∃ σ : Fin d ≃ Fin d', ∀ i, e' (σ i) = e i

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsSmulEqTransposeMulDiagonalMul

theorem **exists_rat_eq_transpose_mul_diagonal_mul** {n : ℕ} (A : Matrix (Fin n) (Fin n) ℚ) (hs : A.IsSymm) (hp : ∀ x : Fin n → ℚ, x ≠ 0 → 0 < x ⬝ᵥ A *ᵥ x) : ∃ (d : Fin n → ℚ) (Q : Matrix (Fin n) (Fin n) ℚ), (∀ i, 0 < d i) ∧ Q.det ≠ 0 ∧ A = Qᵀ * Matrix.diagonal d * Q

theorem **exists_pos_nat_mul_eq_intCast** {ι : Type} [Finite ι] (f : ι → ℚ) : ∃ (b : ℕ) (g : ι → ℤ), 0 < b ∧ ∀ i, (g i : ℚ) = b * f i

theorem **exists_smul_eq_transpose_mul_diagonal_mul** : (∀ (n : ℕ) (A : Matrix (Fin n) (Fin n) ℤ), (A.map (fun x : ℤ => (x : ℝ))).PosDef → ∃ (N : ℕ) (c : Fin n → ℕ) (l : Matrix (Fin n) (Fin n) ℤ), 0 < N ∧ (∀ i, 0 < c i) ∧ l.det ≠ 0 ∧ (N : ℤ) • A = lᵀ * Matrix.diagonal (fun i => (c i : ℤ)) * l)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsStandardUnipotentRadicalBijOnNilradical

private lemma **card_fin_le_val_eq_sub** (d a : ℕ) : Fintype.card {i : Fin d // a ≤ i.val} = d − a

private lemma **finrank_lowSub_of_le** {d a : ℕ} (had : a ≤ d) : Module.finrank F (_root_.GlobalTheory.CuspGL.UnipNilRad.lowSub F d a) = a

private lemma **mem_lowSub_iff_apply_eq_zero** {d a : ℕ} (v : Fin d → F) : v ∈ _root_.GlobalTheory.CuspGL.UnipNilRad.lowSub F d a ↔ ∀ i : Fin d, a ≤ i.val → v i = 0

private lemma **toLin'_mem_nilRad_lowSub_iff** {d a : ℕ} (N : Matrix (Fin d) (Fin d) F) : Matrix.toLin' N ∈ _root_.GlobalTheory.CuspGL.nilRad (_root_.GlobalTheory.CuspGL.UnipNilRad.lowSub F d a) ↔ ∀ i j : Fin d, N i j = 0 ∨ (i.val < a ∧ a ≤ j.val)

theorem **_root_.GlobalTheory.CuspGL.UnipNilRad.exists_unip_bijOn_nilRad** : ∀ (F : Type) [Field F] [Finite F] (d a : ℕ), a ≤ d → ∃ (A : Submodule F (Fin d → F)) (H : Subgroup (GL (Fin d) F)), Module.finrank F A = a ∧ (H : Set (GL (Fin d) F)) = {u : GL (Fin d) F | ∀ i j : Fin d, (u : Matrix (Fin d) (Fin d) F) i j = (if i = j then 1 else 0) ∨ (i.val < a ∧ a ≤ j.val)} ∧ Set.BijOn (fun u : GL (Fin d) F => Matrix.toLin' (u : Matrix (Fin d) (Fin d) F) − 1) (H : Set (GL (Fin d) F)) (_root_.GlobalTheory.CuspGL.nilRad A)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsStandardUnipotentRadicalBijOnNilradicalStatement

theorem **_root_.GlobalTheory.T.exists_standardUnipotentRadical_bijOn_nilradical** : ∀ (F : Type) [Field F] [Finite F] (d a : ℕ), a ≤ d → ∃ (A : Submodule F (Fin d → F)) (H : Subgroup (GL (Fin d) F)), Module.finrank F A = a ∧ (H : Set (GL (Fin d) F)) = {u : GL (Fin d) F | ∀ i j : Fin d, (u : Matrix (Fin d) (Fin d) F) i j = (if i = j then 1 else 0) ∨ (i.val < a ∧ a ≤ j.val)} ∧ Set.BijOn (fun u : GL (Fin d) F => Matrix.toLin' (u : Matrix (Fin d) (Fin d) F) − 1) (H : Set (GL (Fin d) F)) (nilRad A)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsSylowOverSubgroup

theorem **exists_sylow_over_subgroup** (G : Type) [Group G] [Finite G] (C : Subgroup G) (p : ℕ) (hp : p.Prime) : ∃ H : Subgroup G, C ≤ H ∧ (∀ h ∈ H, ∀ c ∈ C, h * c * h⁻¹ ∈ C) ∧ (∀ h ∈ H, ∃ k : ℕ, h ^ p ^ k ∈ C) ∧ ∀ N : Subgroup G, (∀ g : G, g ∈ N ↔ ∀ c ∈ C, g * c * g⁻¹ ∈ C) → ¬ p ∣ H.relIndex N

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsSylowOverSubgroupStatement

theorem **_root_.GlobalTheory.T.exists_subgroup_ge_normalizing_not_dvd_relIndex** : ∀ (G : Type) [Group G] [Finite G] (C : Subgroup G) (p : ℕ), p.Prime → ∃ H : Subgroup G, C ≤ H ∧ (∀ h ∈ H, ∀ c ∈ C, h * c * h⁻¹ ∈ C) ∧ (∀ h ∈ H, ∃ k : ℕ, h ^ p ^ k ∈ C) ∧ ∀ N : Subgroup G, (∀ g : G, g ∈ N ↔ ∀ c ∈ C, g * c * g⁻¹ ∈ C) → ¬ p ∣ H.relIndex N

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsTeichmullerMapAlgebraicClosureFiniteField

private theorem **exists_pow_eq_one_algebraicClosure_of_finite** (F : Type) [Field F] [Finite F] (x : AlgebraicClosure F) (hx : x ≠ 0) : ∃ n : ℕ, 0 < n ∧ x ^ n = 1

theorem **_root_.GlobalTheory.CuspGL.TeichEmb.exists_teichmuller** : ∀ (F : Type) [Field F] [Finite F], ∃ θ : AlgebraicClosure F →* ℂ, ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsTeichmullerMapAlgebraicClosureFiniteFieldStatement

theorem **_root_.GlobalTheory.T.exists_teichmuller_map_algebraicClosure_finiteField** : ∀ (F : Type) [Field F] [Finite F], ∃ θ : AlgebraicClosure F →* ℂ, ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsWeightIdempotent

theorem **exists_weight_idempotent** (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module L V] (ρ : Representation L G V) (B T : Subgroup G) (lam : B →* Lˣ) (v : V) (hB : B.Normal) (hv : v ≠ 0) (hlam : ∀ b : B, ρ b v = (lam b : L) • v) (hT : ∀ g : G, g ∈ T ↔ ∀ b b' : B, (b' : G) = g⁻¹ * b * g → lam b' = lam b) : ∃ e : Module.End L V, e ≠ 0 ∧ e * e = e ∧ (∀ b : B, ρ b * e = (lam b : L) • e) ∧ (∀ t ∈ T, ρ t * e = e * ρ t) ∧ ∀ g ∉ T, e * ρ g * e = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExistsWeightIdempotentStatement

theorem **_root_.GlobalTheory.T.exists_weight_idempotent_of_eigenvector** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module L V] (ρ : Representation L G V) (B T : Subgroup G) (lam : B →* Lˣ) (v : V), B.Normal → v ≠ 0 → (∀ b : B, ρ b v = (lam b : L) • v) → (∀ g : G, g ∈ T ↔ ∀ b b' : B, (b' : G) = g⁻¹ * b * g → lam b' = lam b) → ∃ e : Module.End L V, e ≠ 0 ∧ e * e = e ∧ (∀ b : B, ρ b * e = (lam b : L) • e) ∧ (∀ t ∈ T, ρ t * e = e * ρ t) ∧ ∀ g ∉ T, e * ρ g * e = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExponentRegularOfNatDegreeMinpolyGeneratorPow

theorem **exponent_regular_of_natDegree_minpoly_pow** : ∀ (F : Type) [Field F] [Finite F] (d k : ℕ) (ζ : AlgebraicClosure F), 0 < d → orderOf ζ = Nat.card F ^ d − 1 → (minpoly F (ζ ^ k)).natDegree = d → ∀ i : ℕ, 0 < i → i < d → ¬ k ∗ Nat.card F ^ i ≡ k [MOD Nat.card F ^ d − 1]

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ExponentRegularOfNatDegreeMinpolyGeneratorPowStatement

theorem **_root_.GlobalTheory.T.exponent_regular_of_natDegree_minpoly_generator_pow** : ∀ (F : Type) [Field F] [Finite F] (d k : ℕ) (ζ : AlgebraicClosure F), 0 < d → orderOf ζ = Nat.card F ^ d − 1 → (minpoly F (ζ ^ k)).natDegree = d → ∀ i : ℕ, 0 < i → i < d → ¬ k ∗ Nat.card F ^ i ≡ k [MOD Nat.card F ^ d − 1]

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingAddNatPosSymm

private theorem **character_pairing_eq_finrank_hom** (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] (G : Type) [Group G] [Finite G] {n m : ℕ} (ρ : Representation K G (Fin n → K)) (σ : Representation K G (Fin m → K)) : pair (fun g => algebraMap K L (σ.character g)) (fun g => algebraMap K L (ρ.character g)) = (Module.finrank K (Representation.IntertwiningMap ρ σ) : L)

theorem **pair_add_nat_pos_symm** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G], (∀ a b c : G → L, pair (a + b) c = pair a c + pair b c) ∧ (∀ a b : G → L, IsChar K L a → IsChar K L b → ∃ n : ℕ, pair a b = (n : L)) ∧ (∀ φ : G → L, IsIrrChar K L φ → ∃ n : ℕ, 0 < n ∧ pair φ φ = (n : L)) ∧ ∀ a b : G → L, pair a b = pair b a

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingAddNatPosSymmStatement

theorem **_root_.GlobalTheory.T.finiteGroupFunctionPairing_add_nat_pos_symm** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G], (∀ a b c : G → L, pair (a + b) c = pair a c + pair b c) ∧ (∀ a b : G → L, IsChar K L a → IsChar K L b → ∃ n : ℕ, pair a b = (n : L)) ∧ (∀ φ : G → L, IsIrrChar K L φ → ∃ n : ℕ, 0 < n ∧ pair φ φ = (n : L)) ∧ ∀ a b : G → L, pair a b = pair b a

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingCharacterEqFinrankInte

theorem **_root_.GlobalTheory.BI.pair_character_eq_finrank** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (ρ : Representation K G V) (σ : Representation K G W), _root_.GlobalTheory.BI.pair (fun g => algebraMap K L (σ.character g)) (fun g => algebraMap K L (ρ.character g)) = (Module.finrank K (Representation.IntertwiningMap ρ σ) : L)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingCharacterEqFinrankInteStatement

theorem **_root_.GlobalTheory.T.finiteGroupFunctionPairing_character_eq_finrank_intertwining** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (ρ : Representation K G V) (σ : Representation K G W), pair (fun g => algebraMap K L (σ.character g)) (fun g => algebraMap K L (ρ.character g)) = (Module.finrank K (Representation.IntertwiningMap ρ σ) : L)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingEqZeroOfDistinctIrredu

theorem **pair_eq_zero_of_isIrrChar_ne** : ∀ (G : Type) [Group G] [Finite G] (a b : G → ℂ), IsIrrChar ℂ ℂ a → IsIrrChar ℂ ℂ b → a ≠ b → pair a b = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingEqZeroOfDistinctIrreduStatement

theorem **_root_.GlobalTheory.T.finiteGroupFunctionPairing_eq_zero_of_distinct_irreducible** : ∀ (G : Type) [Group G] [Finite G] (a b : G → ℂ), IsIrrChar ℂ ℂ a → IsIrrChar ℂ ℂ b → a ≠ b → pair a b = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingSelfIrreducibleAlgClos

theorem **pair_self_of_isIrrChar_algClosed** (L : Type) [Field L] [CharZero L] [IsAlgClosed L] (G : Type) [Group G] [Finite G] (χ : G → L) (h : IsIrrChar L L χ) : pair χ χ = 1

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteGroupFunctionPairingSelfIrreducibleAlgClosStatement

theorem **_root_.GlobalTheory.T.finiteGroupFunctionPairing_self_irreducible_algClosed** : ∀ (L : Type) [Field L] [CharZero L] [IsAlgClosed L] (G : Type) [Group G] [Finite G] (χ : G → L), IsIrrChar L L χ → pair χ χ = 1

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FiniteMonicNatDegree

private theorem **torus_iff** {F : Type} [Field F] [Finite F] (n : ℕ) (α : AlgebraicClosure F) (h0 : α ≠ 0) : α ^ (Nat.card F ^ n − 1) = 1 ↔ (minpoly F α).natDegree ∣ n

private theorem **finite_monic_natDegree** {F : Type} [Field F] [Finite F] (n : ℕ) : {P : F[X] | P.Monic ∧ P.natDegree = n ∧ P.coeff 0 ≠ 0}.Finite

private theorem **finite_torus** {F : Type} [Field F] [Finite F] (n : ℕ) (hn : 0 < n) : {α : AlgebraicClosure F | α ^ (Nat.card F ^ n − 1) = 1}.Finite

theorem **sum_poly_eq_sum_torus** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (n : ℕ), 0 < n → ∀ (c : F[X] → ℂ) (l : ℕ), (∀ P : F[X], P.Monic → P.natDegree = n → (¬ ∃ (f : F[X]) (m : ℕ), f.Monic ∧ Irreducible f ∧ 0 < m ∧ P = f ^ m) → c P = 0) → ∑ᶠ P ∈

{P : F[X] | P.Monic ∧ P.natDegree = n ∧ P.coeff 0 ≠ 0}, c P * star (rootSum θ l P) = ∏ᶠ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n - 1) = 1}, ((n / (minpoly F α).natDegree : ℕ) : ℂ) * c (minpoly F α ^ (n / (minpoly F α).natDegree)) * star (θ α ^ l)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinrankEigenspaceMonomialMatrixLift

lemma **orbProd_succ** (m : ℕ) (j : Fin n) : orbProd σ a (m + 1) j = orbProd σ a m j * a ((σ ^ m) j)

lemma **orbProd_add** (d m : ℕ) (j : Fin n) : orbProd σ a (d + m) j = orbProd σ a d j * orbProd σ a m ((σ ^ d) j)

lemma **orbProd_eq** {j : Fin n} (hj : ∀ m : ℕ, (σ ^ m) j = j → orbProd σ a m j = 1) (m m' : ℕ) (h : (σ ^ m) j = (σ ^ m') j) : orbProd σ a m j = orbProd σ a m' j

lemma **apply_pow** {v : Fin n → K} (hv : ∀ j, v (σ j) = a j * v j) (m : ℕ) (j : Fin n) : v ((σ ^ m) j) = orbProd σ a m j * v j

lemma **exists_back** (m : ℕ) (j : Fin n) : ∃ m' : ℕ, (σ ^ m') ((σ ^ m) j) = j

lemma **exists_rep** (i : Fin n) : ∃ j : Fin n, (∀ m : ℕ, j ≤ (σ ^ m) j) ∧ ∃ m : ℕ, (σ ^ m) j = i

lemma **exists_vec** {s : Fin n} (hs : GoodRep σ a s) : ∃ v : Fin n → K, (∀ j, v (σ j) = a j * v j) ∧ v s = 1 ∧ ∀ t, GoodRep σ a t → t ≠ s → v t = 0

theorem **finrank_eq_card** (E : Submodule K (Fin n → K)) (hE : ∀ v, v ∈ E ↔ ∀ j, v (σ j) = a j * v j) : Module.finrank K E = Nat.card {j // GoodRep σ a j}

lemma **mem_eigenspace_iff** (c : Fin n → K) (μ : K) (hμ : μ ≠ 0) (v : Fin n → K) : v ∈ Module.End.eigenspace (Matrix.toLin' (Matrix.of fun i j => if i = σ j then c j else 0)) μ ↔ ∀ j, v (σ j) = (c j * μ⁻¹) * v j

theorem **finrank_eigenspace_monomial_lift** : ∀ (k : Type) [Field k] (θ : k →* ℂ), (∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (n : ℕ) (σ : Equiv.Perm (Fin n)) (c : Fin n → k), (∀ j, c j ≠ 0) → ∀ μ : k, μ ≠ 0 → Module.finrank k (Module.End.eigenspace (Matrix.toLin' (Matrix.of fun i j => if i = σ j then c j else 0)) μ) = Module.finrank ℂ (Module.End.eigenspace (Matrix.toLin' (Matrix.of fun i j => if i = σ j then θ (c j) else 0)) (θ μ))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinrankEigenspaceMonomialMatrixLiftStatement

theorem **_root_.GlobalTheory.T.finrank_eigenspace_monomial_matrix_lift** : ∀ (k : Type) [Field k] (θ : k →* ℂ), (∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (n : ℕ) (σ : Equiv.Perm (Fin n)) (c : Fin n → k), (∀ j, c j ≠ 0) → ∀ μ : k, μ ≠ 0 → Module.finrank k (Module.End.eigenspace (Matrix.toLin' (Matrix.of fun i j => if i = σ j then c j else 0)) μ) = Module.finrank ℂ (Module.End.eigenspace (Matrix.toLin' (Matrix.of fun i j => if i = σ j then θ (c j) else 0)) (θ μ))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumCardCyclicVectorsInvariantSubspaces

theorem **finsum_card_cyclic_eq** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) : ∏ᶠ W : {W : Submodule F V // ∀ x ∈ W, g x ∈ W}, (Nat.card {v : W.1 // ∀ U : Submodule F (W.1), (∀ x ∈ U, (g.restrict W.2) x ∈ U) → v ∈ U → U = ⊤} : ℂ) = (Nat.card F : ℂ) ^ Module.finrank F V

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumCardCyclicVectorsInvariantSubspacesStatement

theorem **_root_.GlobalTheory.T.finsum_card_cyclic_vectors_invariant_subspaces** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V), ∏ᶠ W : {W : Submodule F V // ∀ x ∈ W, g x ∈ W}, (Nat.card {v : W.1 // ∀ U : Submodule F (W.1), (∀ x ∈ U, (g.restrict W.2) x ∈ U) → v ∈ U → U = ⊤} : ℂ) = (Nat.card F : ℂ) ^ Module.finrank F V

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumMobiusInverseCharpolyInvariantLe

theorem **finsum_psi_le** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A) : ∏ᶠ W : {W : Submodule F V // W ≤ A ∧ ∀ x ∈ W, g x ∈ W}, psi s W.1 (g.restrict W.2.2) = s (LinearMap.charpoly (g.restrict hg))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumMobiusInverseCharpolyInvariantLeStatement

theorem **_root_.GlobalTheory.T.finsum_mobiusInverseCharpoly_invariant_le** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A), ∏ᶠ W : {W : Submodule F V // W ≤ A ∧ ∀ x ∈ W, g x ∈ W}, psi s W.1 (g.restrict W.2.2) = s (LinearMap.charpoly (g.restrict hg))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumMobiusInverseCharpolyTransversalSubspaces

private lemma **card_nilRad_eq_card_linearMap_quot** (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] (A : Submodule F V) : Nat.card (_root_.GlobalTheory.CuspGL.nilRad A) = Nat.card ((V ⧸ A) →ₗ[F] A)

private lemma **card_invariant_complements_over_eq_card_nilRad** (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A) (U : Submodule F (V ⧸ A)) (hU : ∀ y ∈ U, A.mapQ A g hg y ∈ U) : Nat.card {x : Module.End F V × Submodule F V // x.1 ∈ _root_.GlobalTheory.CuspGL.nilRad A ∧ x.2 ⊓ A = ⊥ ∧ (∀ v ∈ x.2, (g + x.1) v ∈ x.2) ∧ x.2.map A.mkQ = U} = Nat.card (_root_.GlobalTheory.CuspGL.nilRad A)

private lemma **psi_restrict_complement_eq_psi_quotient** (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A) (n : Module.End F V) (hn : n ∈ _root_.GlobalTheory.CuspGL.nilRad A) (W : Submodule F V) (hWA : W ⊓ A = ⊥) (hW : ∀ v ∈ W, (g + n) v ∈ W) (hU : ∀ y ∈ W.map A.mkQ, A.mapQ A g hg y ∈ W.map A.mkQ) : _root_.GlobalTheory.CuspGL.psi s W ((g + n).restrict hW) = _root_.GlobalTheory.CuspGL.psi s (W.map A.mkQ) ((A.mapQ A g hg).restrict hU)

theorem **_root_.GlobalTheory.CuspGL.finsum_psi_transversal** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A), ∏ᶠ x : {x : Module.End F V × Submodule F V // x.1 ∈ _root_.GlobalTheory.CuspGL.nilRad A ∧ x.2 ⊓ A = ⊥ ∧ ∀ v ∈ x.2, (g + x.1) v ∈ x.2}, _root_.GlobalTheory.CuspGL.psi s x.1.2 ((g + x.1.1).restrict x.2.2.2) = (Nat.card (_root_.GlobalTheory.CuspGL.nilRad A) : ℂ) * s (LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx)))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumMobiusInverseCharpolyTransversalSubspacesStatement

theorem **_root_.GlobalTheory.T.finsum_mobiusInverseCharpoly_transversal_subspaces** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (A : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A), ∏ᶠ x : {x : Module.End F V × Submodule F V // x.1 ∈ nilRad A ∧ x.2 ⊓ A = ⊥ ∧ ∀ v ∈ x.2, (g + x.1) v ∈ x.2}, psi s x.1.2 ((g + x.1.1).restrict x.2.2.2) = (Nat.card (nilRad A) : ℂ) * s (LinearMap.charpoly (A.mapQ A g (fun x hx => hg x hx)))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumSubspaceStabilizerMulBlockFunctions

theorem **finsum_stabiliser_mul_blocks** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (W : Submodule F V) (Φ : Module.End F W → ℂ) (Φ' : Module.End F (V ⧸ W) → ℂ) : ∑ᶠ g : {g : Module.End F V // ∀ x ∈ W, g x ∈ W}, Φ (g.1.restrict g.2) * Φ' (W.mapQ W g.1 (fun x hx => g.2 x hx)) = (Nat.card F : ℂ) ^ (Module.finrank F W * Module.finrank F (V ⧸ W)) * ((∑ᶠ a, Φ a) * ∑ᶠ b, Φ' b)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumSubspaceStabilizerMulBlockFunctionsStatement

theorem **_root_.GlobalTheory.T.finsum_subspaceStabilizer_mul_block_functions** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (W : Submodule F V) (Φ : Module.End F W → ℂ) (Φ' : Module.End F (V ⧸ W) → ℂ), ∑ᶠ g : {g : Module.End F V // ∀ x ∈ W, g x ∈ W}, Φ (g.1.restrict g.2) * Φ' (W.mapQ W g.1 (fun x hx => g.2 x hx)) = (Nat.card F : ℂ) ^ (Module.finrank F W * Module.finrank F (V ⧸ W)) * ((∑ᶠ a, Φ a) * ∑ᶠ b, Φ' b)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumUnipotentRadicalEqZero

theorem **IsPrimitive.finsum_unipotentRadical_eq_zero** (F : Type) [Field F] [Finite F] (d : ℕ) (φ : Module.End F (Fin d → F) → ℂ) (hφ : IsPrimitive φ) (a : ℕ) (ha : 0 < a) (had : a < d) : ∑ᶠ u ∈ {u : GL (Fin d) F | ∀ i j : Fin d, (u : Matrix (Fin d) (Fin d) F) i j = (if i = j then 1 else 0) ∨ (i.val < a ∧ a ≤ j.val)}, φ (Matrix.toLin' (u : Matrix (Fin d) (Fin d) F)) = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FinsumUnipotentRadicalEqZeroStatement

theorem **_root_.GlobalTheory.T.finsum_unipotent_radical_eq_zero_of_isPrimitive** : ∀ (F : Type) [Field F] [Finite F] (d : ℕ) (φ : Module.End F (Fin d → F) → ℂ), IsPrimitive φ → ∀ a : ℕ, 0 < a → a < d → ∑ᶠ u ∈ {u : GL (Fin d) F | ∀ i j : Fin d, (u : Matrix (Fin d) (Fin d) F) i j = (if i = j then 1 else 0) ∨ (i.val < a ∧ a ≤ j.val)}, φ (Matrix.toLin' (u : Matrix (Fin d) (Fin d) F)) = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FrobeniusTorusCharacterOrthogonality

theorem **finsum_rootsOfUnity_pow_mul_star** {K : Type} [Field K] (θ : K →* ℂ) {N : ℕ} (hN : 0 < N) {ζ : K} (hζ : IsPrimitiveRoot ζ N) (hω : IsPrimitiveRoot (θ ζ) N) (k l : ℕ) : ∑ᶠ α ∈ {α : K | α ^ N = 1}, θ α ^ k * star (θ α ^ l) = if k ≡ l [MOD N] then ((N : ℕ) : ℂ) else 0

theorem **torus_orthogonality** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ n : ℕ, 0 < n → (∃ ζ : AlgebraicClosure F, IsPrimitiveRoot ζ (Nat.card F ^ n − 1) ∧ IsPrimitiveRoot (θ ζ) (Nat.card F ^ n − 1)) ∧ ∀ k l : ℕ, ∑ᶠ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n − 1) = 1}, θ α ^ k * star (θ α ^ l) = if k ≡ l [MOD Nat.card F ^ n − 1] then ((Nat.card F ^ n − 1 : ℕ) : ℂ) else 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FrobeniusTorusCharacterOrthogonalityStatement

theorem **_root_.GlobalTheory.T.frobeniusTorus_character_orthogonality** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ n : ℕ, 0 < n → (∃ ζ : AlgebraicClosure F, IsPrimitiveRoot ζ (Nat.card F ^ n − 1) ∧ IsPrimitiveRoot (θ ζ) (Nat.card F ^ n − 1)) ∧ ∀ k l : ℕ, ∑ᶠ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n − 1) = 1}, θ α ^ k * star (θ α ^ l) = if k ≡ l [MOD Nat.card F ^ n − 1] then ((Nat.card F ^ n − 1 : ℕ) : ℂ) else 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FrobeniusTorusMinpolyDegreeRootPowerSum

theorem **_root_.GlobalTheory.CuspGL.minpoly_natDegree_dvd_iff_pow** (α : AlgebraicClosure F) (m : ℕ) : (minpoly F α).natDegree ∣ m ↔ α ^ Nat.card F ^ m = α

theorem **exists_algHom_pow_card_pow** (i : ℕ) : ∃ f : AlgebraicClosure F →ₐ[F] AlgebraicClosure F, ∀ x, f x = x ^ Nat.card F ^ i

theorem **_root_.GlobalTheory.CuspGL.roots_minpoly_eq_map_range** (α : AlgebraicClosure F) : ((minpoly F α).map (algebraMap F (AlgebraicClosure F))).roots = (Multiset.range (minpoly F α).natDegree).map fun i => α ^ Nat.card F ^ i

private theorem **sum_range_mul_eq_nsmul_of_periodic** {M : Type} [AddCommMonoid M] (f : ℕ → M) (d : ℕ) (hf : ∀ i j, f (d * j + i) = f i) (e : ℕ) : ∑ i ∈ Finset.range (d * e), f i = e • ∑ i ∈ Finset.range d, f i

theorem **_root_.GlobalTheory.CuspGL.torus_minpoly** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (n : ℕ), 0 < n → ∀ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n − 1) = 1}, (minpoly F α).natDegree ∣ n ∧ (minpoly F α).coeff 0 ≠ 0 ∧ ∀ k : ℕ, _root_.GlobalTheory.CuspGL.rootSum θ k (minpoly F α ^ (n / (minpoly F α).natDegree)) = ∑ i ∈ Finset.range n, θ (α ^ Nat.card F ^ i) ^ k

Development.AbelianVarietyTheory.Proofs.CharacterTheory.FrobeniusTorusMinpolyDegreeRootPowerSumStatement

theorem **_root_.GlobalTheory.T.frobeniusTorus_minpoly_degree_rootPowerSum** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (n : ℕ), 0 < n → ∀ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n − 1) = 1}, (minpoly F α).natDegree ∣ n ∧ (minpoly F α).coeff 0 ≠ 0 ∧ ∀ k : ℕ, rootSum θ k (minpoly F α ^ (n / (minpoly F α).natDegree)) = ∑ i ∈ Finset.range n, θ (α ^ Nat.card F ^ i) ^ k

Development.AbelianVarietyTheory.Proofs.CharacterTheory.GeneralLinearFunctionExtendZeroEndProperties

theorem **extZero_toLin_injective** : Function.Injective fun x : GL (Fin d) F => Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)

theorem **extZero_isUnit_iff** (g : Module.End F (Fin d → F)) : IsUnit g ↔ ∃ x : GL (Fin d) F, Matrix.toLin' (x : Matrix (Fin d) (Fin d) F) = g

theorem **extZero_toLin** (χ : GL (Fin d) F → ℂ) (x : GL (Fin d) F) : extZero χ (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)) = χ x

theorem **extZero_of_not_isUnit** (χ : GL (Fin d) F → ℂ) (g : Module.End F (Fin d → F)) (hg : ¬ IsUnit g) : extZero χ g = 0

theorem **ipGL_eq_finsum_GL** (φ φ' : Module.End F (Fin d → F) → ℂ) : ipGL φ φ' = ∑ᶠ x : GL (Fin d) F, φ (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)) * star (φ' (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)))

theorem **isClassFn_extZero** (χ : GL (Fin d) F → ℂ) (hχ : ∀ g x : GL (Fin d) F, χ (x⁻¹ * g * x) = χ g) : IsClassFn (extZero χ)

theorem **extZero_spec** : ∀ (F : Type) [Field F] [Finite F] (d : ℕ) (χ : GL (Fin d) F → ℂ), (∀ x : GL (Fin d) F, extZero χ (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)) = χ x) ∧ (∀ g, ¬ IsUnit g → extZero χ g = 0) ∧ (∀ φ φ' : Module.End F (Fin d → F) → ℂ, ipGL φ φ' = ∑ᶠ x : GL (Fin d) F, φ (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)) * star (φ' (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)))) ∧ ((∀ g x : GL (Fin d) F, χ (x⁻¹ * g * x) = χ g) → IsClassFn (extZero χ))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.GeneralLinearFunctionExtendZeroEndPropertiesStatement

theorem **_root_.GlobalTheory.T.generalLinearFunctionExtendZeroEnd_properties** : ∀ (F : Type) [Field F] [Finite F] (d : ℕ) (χ : GL (Fin d) F → ℂ), (∀ x : GL (Fin d) F, extZero χ (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)) = χ x) ∧ (∀ g, ¬ IsUnit g → extZero χ g = 0) ∧ (∀

φ φ' : Module.End F (Fin d → F) → ℂ, ipGL φ φ' = ∑ᶠ x : GL (Fin d) F, φ (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)) * star (φ' (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F)))) ∧ ((∀ g x : GL (Fin d) F, χ (x⁻¹ * g * x) = χ g) → IsClassFn (extZero χ))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.Imports

theorem **main** {L : Type} [Field L] [IsAlgClosed L] {ι V : Type} [AddCommGroup V] [Module L V] [FiniteDimensional L V] [Nontrivial V] (f : ι → Module.End L V) (hc : ∀ (a b : ι), f a * f b = f b * f a) : ∃ (v : V), v ≠ (0 : V) ∧ ∀ (b : ι), ∃ (μ : L), (f b : V → V) v = μ • v

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ImportsAlt

theorem **main** : ∀ {G : Type u_1} [inst : Group G] {V : Type u_2} [inst_1 : AddCommGroup V] [inst_2 : Module ℂ V] (ρ : Representation ℂ G V), Representation.IsIrreducible ρ ↔ Nontrivial V ∧ ∀ (U : Submodule ℂ V), (∀ (g : G), ∀ v ∈ U, ((ρ : G → V →ₗ[ℂ] V) g : V → V) v ∈ U) → U = ⊥ ∨ U = ⊤

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IndMulClassFunEqIndMulRestrict

theorem **ind_mul_classFun_eq_ind_mul_restrict** : ∀ (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → ℂ) (φ : G → ℂ), (∀ g x : G, φ (x⁻¹ * g * x) = φ g) → ind H ψ * φ = ind H (fun h => ψ h * φ h)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IndRestrictEqCardFixedByMul

theorem **out_conj_mem_iff_smul_eq** {G : Type} [Group G] (H : Subgroup G) (g : G) (q : G ⧸ H) : q.out⁻¹ * g * q.out ∈ H ↔ g • q = q

theorem **ind_restrict_eq_card_fixedBy_mul** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (χ : G → L), (∀ g x : G, χ (x⁻¹ * g * x) = χ g) → ∀ g : G, ind H (fun h : H => χ h) g = (Nat.card (MulAction.fixedBy (G ⧸ H) g) : L) * χ g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IndRestrictEqCardFixedByMulStatement

theorem **_root_.GlobalTheory.T.ind_restrict_eq_card_fixedBy_mul_of_classFn** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (χ : G → L), (∀ g x : G, χ (x⁻¹ * g * x) = χ g) → ∀ g : G, ind H (fun h : H => χ h) g = (Nat.card (MulAction.fixedBy (G ⧸ H) g) : L) * χ g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InducedClassFunctionFromSubgroupEqFinsumCosets

theorem **subgroupFunctionExtendByZero_conj_of_mem** {L G : Type} [Field L] [Group G] (H : Subgroup G) (ψ : H → L) (hψ : ∀ h x : H, ψ (x⁻¹ * h * x) = ψ h) (a : G) (y : H) : ext0 H ψ ((y : G)⁻¹ * a * y) = ext0 H ψ a

theorem **out_mul_bijective** {G : Type} [Group G] (H : Subgroup G) : Function.Bijective (fun p : (G ⧸ H) × H => p.1.out * (p.2 : G))

theorem **ind_eq_finsum_cosets** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → L), (∀ h x : H, ψ (x⁻¹ * h * x) = ψ h) → ∀ g : G, ind H ψ g = ∑ᶠ q : G ⧸ H, ext0 H ψ (q.out⁻¹ * g * q.out)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InducedClassFunctionFromSubgroupEqFinsumCosetsStatement

theorem **_root_.GlobalTheory.T.inducedClassFunctionFromSubgroup_eq_finsum_cosets** : ∀ (L : Type) [Field L] [CharZero L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → L), (∀ h x : H, ψ (x⁻¹ * h * x) = ψ h) → ∀ g : G, ind H ψ g = ∑ᶠ q : G ⧸ H, ext0 H ψ (q.out⁻¹ * g * q.out)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InnerProductRootPowerSumEqOfFrobeniusModEq

theorem **exists_algEquiv_pow_card_pow** (F : Type) [Field F] [Finite F] (i : ℕ) : ∃ σ : AlgebraicClosure F ≃ₐ[F] AlgebraicClosure F, ∀ α, σ α = α ^ (Nat.card F ^ i)

theorem **ipGL_rootSum_eq_of_modEq** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (φ : Module.End F V → ℂ) (hc : IsClassFn φ) (hp : IsPrimitive φ) (hn : 0 < Module.finrank F V) (k l i : ℕ) (h : k * Nat.card F ^ i ≡ l [MOD (Nat.card F ^ Module.finrank F V - 1)]) : ipGL φ (fun g : Module.End F V => rootSum θ k (LinearMap.charpoly g)) = ipGL φ (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InnerProductRootPowerSumEqOfFrobeniusModEqStatement

theorem **_root_.GlobalTheory.T.innerProduct_rootPowerSum_eq_of_frobenius_modEq** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → 0 < Module.finrank F V → ∀ k l i : ℕ, k * Nat.card F ^ i ≡ l [MOD (Nat.card F ^ Module.finrank F V - 1)] → ipGL φ (fun g : Module.End F V => rootSum θ k (LinearMap.charpoly g)) = ipGL φ (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantEqTopOfIrreducible

theorem **invariant_eq_top_of_irreducible** {k : Type} [Field k] {G : Type} [Group G] {V : Type} [AddCommGroup V] [Module k V] (ρ : Representation k G V) (h : ρ.IsIrreducible) (U : Submodule k V) (hU : ∀ g, ∀ v ∈ U, ρ g v ∈ U) (v : V) (hv : v ∈ U) (hv0 : v ≠ 0) : U = ⊤

theorem **finrank_eq_one_or_imprimitive** : ∀ (k : Type) [Field k] (G : Type) [Group G] [Finite G] (V : Type) [AddCommGroup V] [Module k V] [FiniteDimensional k V] (ρ : Representation k G V) [IsAlgClosed k], AbByP G → ρ.IsIrreducible → Module.finrank k V = 1 ∨ ∃ (T : Subgroup G) (W : Submodule k V), T ≠ ⊤ ∧ (∀ t ∈ T, ∀ w ∈ W, ρ t w ∈ W) ∧ iSupIndep (fun q : G ⧸ T => W.map (ρ q.out)) ∧ ⊔ q : G ⧸ T, W.map (ρ q.out) = ⊤

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantSubspaceMobiusInverseCharpolyLinearEqui

theorem **_root_.GlobalTheory.CuspGL.invariantSubspaceMobiusInverseRec_linearEquiv_conj** {F : Type} [Field F] (s : F[X] → ℂ) (n : ℕ) : ∀ (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), _root_.GlobalTheory.CuspGL.invariantSubspaceMobiusInverseRec s n V' (e.conj g) = _root_.GlobalTheory.CuspGL.invariantSubspaceMobiusInverseRec s n V g

theorem **_root_.GlobalTheory.CuspGL.psi_conj** (F : Type) [Field F] (s : F[X] → ℂ) (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V) : _root_.GlobalTheory.CuspGL.psi s V' (e.conj g) = _root_.GlobalTheory.CuspGL.psi s V g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantSubspaceMobiusInverseCharpolyLinearEquiStatement

theorem **_root_.GlobalTheory.T.invariantSubspaceMobiusInverseCharpoly_linearEquiv_conj** : ∀ (F : Type) [Field F] (s : F[X] → ℂ) (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), psi s V' (e.conj g) = psi s V g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantSubspaceMobiusInverseCharpolyOfIrreduci

theorem **psi_of_finrank_eq_zero** {F : Type} [Field F] (s : F[X] → ℂ) (hs : IsAdditive s) (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) (h : Module.finrank F V = 0) : psi s V g = 0

theorem **coe_aeval_restrict_apply** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] (g : Module.End F V) (W : Submodule F V) (hW : ∀ x ∈ W, g x ∈ W) (p : F[X]) (x : W) : ((aeval (g.restrict hW : Module.End F W) p x : W) : V) = aeval g p x

theorem **finrank_eq_zero_of_irreducible_charpoly** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) (hg : Irreducible (LinearMap.charpoly g)) (W : Submodule F V) (hne : W ≠ ⊤) (hW : ∀ x ∈ W, g x ∈ W) : Module.finrank F W = 0

theorem **psi_of_irreducible_charpoly** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → ∀ g : Module.End F V, Irreducible (LinearMap.charpoly g) → psi s V g = s (LinearMap.charpoly g)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantSubspaceMobiusInverseCharpolyOfIrreduciStatement

theorem **_root_.GlobalTheory.T.invariantSubspaceMobiusInverseCharpoly_of_irreducible** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → ∀ g : Module.End F V, Irreducible (LinearMap.charpoly g) → psi s V g = s (LinearMap.charpoly g)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantSubspaceMobiusInverseOneProdIco

theorem **sum_oneValue_finrank** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [Fintype (Submodule F V)] : ∑ W : Submodule F V, oneValue (Nat.card F) (Module.finrank F W) = (Module.finrank F V : ℂ)

theorem **psi_one_eq_oneValue** (F : Type) [Field F] [Finite F] (s : F[X] → ℂ) (hs : ∀ j : ℕ, s ((X − C 1) ^ j) = j) (n : ℕ) : ∀ (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V], Module.finrank F V = n → psi s V 1 = oneValue (Nat.card F) n

theorem **psi_one** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (hs : ∀ j : ℕ, s ((X − C 1) ^ j) = j) (h : 0 < Module.finrank F V) : psi s V 1 = ∏ i ∈ Finset.Ico 1 (Module.finrank F V), (1 − (Nat.card F : ℂ) ^ i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvariantSubspaceMobiusInverseOneProdIcoStatement

theorem **_root_.GlobalTheory.T.invariantSubspaceMobiusInverse_one_prod_Ico** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), (∀ j : ℕ, s ((X − C 1) ^ j) = j) → 0 < Module.finrank F V → psi s V 1 = ∏ i ∈ Finset.Ico 1 (Module.finrank F V), (1 − (Nat.card F : ℂ) ^ i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvertibleEndInnerProductRootPowerSumEqTorusSum

theorem **ipGL_rootSum_eq_sum_torus** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → 0 < Module.finrank F V → ∀ l : ℕ, ipGL φ (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g)) = ∑ᶠ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ Module.finrank F V − 1) = 1}, ((Module.finrank F V / (minpoly F α).natDegree : ℕ) : ℂ) * mass φ (minpoly F α ^ (Module.finrank F V / (minpoly F α).natDegree)) * star (θ α ^ l)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvertibleEndInnerProductRootPowerSumEqTorusSumStatement

theorem **_root_.GlobalTheory.T.invertibleEndInnerProduct_rootPowerSum_eq_torusSum** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → 0 < Module.finrank F V → ∀ l : ℕ, ipGL φ (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g)) = ∑ᶠ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ Module.finrank F V − 1) = 1}, ((Module.finrank F V / (minpoly F α).natDegree : ℕ) : ℂ) * mass φ (minpoly F α ^ (Module.finrank F V / (minpoly F α).natDegree)) * star (θ α ^ l)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvertibleEndInnerProductRootPowerSumMobius

theorem **IsAdditive.map_pow_monic** {F : Type} [Field F] {s : F[X] → ℂ} (hs : IsAdditive s) {f : F[X]} (hf : f.Monic) (m : ℕ) : s (f ^ m) = m * s f

theorem **ipGL_psi_rootSum** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (hn : 0 < Module.finrank F V) (k l : ℕ) : ipGL (psi (rootSum θ k) V) (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g)) = glCard F (Module.finrank F V) * (((Finset.range (Module.finrank F V)).filter (fun i => k * Nat.card F ^ i ≡ l [MOD (Nat.card F ^ Module.finrank F V − 1)])).card : ℂ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvertibleEndInnerProductRootPowerSumMobiusStatement

theorem **_root_.GlobalTheory.T.invertibleEndInnerProduct_rootPowerSum_mobius** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → 0 < Module.finrank F V → ∀ k l : ℕ, ipGL (psi (rootSum θ k) V) (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g)) = glCard F (Module.finrank F V) * (((Finset.range (Module.finrank F V)).filter (fun i => k * Nat.card F ^ i ≡ l [MOD (Nat.card F ^ Module.finrank F V − 1)])).card : ℂ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvertibleEndomorphismInnerProductPrimitiveMobiu

theorem **neg_mem_nilRad** {A : Submodule F V} {h : Module.End F V} (hh : h ∈ nilRad A) : -h ∈ nilRad A

theorem **add_preserves** {A : Submodule F V} {g h : Module.End F V} (hh : h ∈ nilRad A) (hg : ∀ x ∈ A, g x ∈ A) : ∀ x ∈ A, (g + h) x ∈ A

theorem **add_preserves_iff** {A : Submodule F V} {g h : Module.End F V} (hh : h ∈ nilRad A) : (∀ x ∈ A, (g + h) x ∈ A) ↔ ∀ x ∈ A, g x ∈ A

theorem **isUnit_add** [FiniteDimensional F V] {A : Submodule F V} {g h : Module.End F V} (hh : h ∈ nilRad A) (hg : ∀ x ∈ A, g x ∈ A) (hu : IsUnit g) : IsUnit (g + h)

theorem **isUnit_add_iff** [FiniteDimensional F V] {A : Submodule F V} {g h : Module.End F V} (hh : h ∈ nilRad A) (hg : ∀ x ∈ A, g x ∈ A) : IsUnit (g + h) ↔ IsUnit g

theorem **sum_mul_eq_zero** [Fintype (Module.End F V)] {φ : Module.End F V → ℂ} (hφ : IsPrimitive φ) {A : Submodule F V} (h0 : A ≠ ⊥) (h1 : A ≠ ⊤) (f : Module.End F V → ℂ) (hf : ∀ g, ∀ h ∈ nilRad A, f (g + h) = f g) (hf0 : ∀ g, ¬ (∀ x ∈ A, g x ∈ A) → f g = 0) : ∑ g, φ g * f g = 0

theorem **psi_bot** {s : F[X] → ℂ} (hs : IsAdditive s) (U : Type) [AddCommGroup U] [Module F U] [FiniteDimensional F U] [Subsingleton U] (g : Module.End F U) : psi s U g = 0

theorem **ipGL_psi_eq** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ) (hs : IsAdditive s) (φ : Module.End F V → ℂ) (hφ : IsPrimitive φ) : ipGL φ (psi s V) = ipGL φ (fun g => s (LinearMap.charpoly g))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.InvertibleEndomorphismInnerProductPrimitiveMobiuStatement

theorem **_root_.GlobalTheory.T.invertibleEndomorphismInnerProduct_primitive_mobiusInverse** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → ∀ φ : Module.End F V → ℂ, IsPrimitive φ → ipGL φ (psi s V) = ipGL φ (fun g => s (LinearMap.charpoly g))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrrCharEqOneOfSubsingleton

theorem **irrChar_eq_one_of_subsingleton** {G : Type} [Group G] [Fintype G] [Subsingleton G] (χ : G → ℂ) (h : CuspElliptic.IsIrrChar χ) (x : G) : χ x = 1

theorem **isIrrChar_of_norm_one** {G : Type} [Group G] [Fintype G] (χ : G → ℂ) (h : CuspElliptic.IsIrrChar χ) : GlobalTheory.BI.IsIrrChar ℂ ℂ χ

theorem **isPrimitive_of_isCuspChar** {F : Type} [Field F] [Fintype F] [DecidableEq F] {d : ℕ} (χ : GL (Fin d) F → ℂ) (h : CuspElliptic.IsCuspChar χ) : IsPrimitive (extZero χ)

theorem **cuspElliptic_statement** : CuspElliptic.Statement

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrreducibleCharacterLinearOrInducedAbelianByPrim

theorem **nontrivial_of_isIrreducible** {L G V : Type} [Field L] [Group G] [AddCommGroup V] [Module L V] (ρ : Representation L G V) (h : ρ.IsIrreducible) : Nontrivial V

theorem **irrChar_linear_or_induced** : ∀ (L : Type) [Field L] [CharZero L] [IsAlgClosed L] (G : Type) [Group G] [Finite G] (χ : G → L), AbByP G → IsIrrChar L L χ → (∃ lam : G →* Lˣ, ∀ g, χ g = (lam g : L)) ∨ ∃ T : Subgroup G, T ≠ ⊤ ∧ ∃ ψ : T → L, IsChar L L ψ ∧ χ = ind T ψ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrreducibleCharacterLinearOrInducedAbelianByPrimStatement

theorem **_root_.GlobalTheory.T.irreducibleCharacter_linear_or_induced_abelianByPrimePower** : ∀ (L : Type) [Field L] [CharZero L] [IsAlgClosed L] (G : Type) [Group G] [Finite G] (χ : G → L), AbByP G → IsIrrChar L L χ → (∃ lam : G →* Lˣ, ∀ g, χ g = (lam g : L)) ∨ ∃ T : Subgroup G, T ≠ ⊤ ∧ ∃ ψ : T → L, IsChar L L ψ ∧ χ = ind T ψ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrreducibleIdempotentEqOneAndCharacterEqWeightAl

theorem **irreducible_idempotent_eq_one_and_character_eq_weight** (L : Type) [Field L] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] (ρ : Representation L G V) (hρ : ρ.IsIrreducible) : (∀ e : Module.End L V, e ≠ 0 → e * e = e → (∀ g, ρ g * e = e * ρ g) → e = 1) ∧ (∀ (lam : G →* Lˣ) (v : V), v ≠ 0 → (∀ g, ρ g v = (lam g : L) • v) → ∀ g, ρ.character g = (lam g : L))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrreducibleIdempotentEqOneAndCharacterEqWeightAlStatement

theorem **_root_.GlobalTheory.T.idempotent_eq_one_and_character_eq_of_isIrreducible** : ∀ (L : Type) [Field L] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] (ρ : Representation L G V), ρ.IsIrreducible → (∀ e : Module.End L V, e ≠ 0 → e * e = e → (∀ g, ρ g * e = e * ρ g) → e = 1) ∧ (∀ (lam : G →* Lˣ) (v : V), v ≠ 0 → (∀ g, ρ g v = (lam g : L) • v) → ∀ g, ρ.character g = (lam g : L))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrreducibleOfVirtualCharacterNormOne

theorem **finsupp_eq_single_of_sum_sq_eq_one** {α : Type} (c : α →₀ ℤ) (h : (c.sum fun _ m => m ^ 2) = 1) : ∃ a, c.support = {a} ∧ (c a = 1 ∨ c a = -1)

private theorem **exists_pos_nat_apply_one_of_isIrrChar** {G : Type} [Group G] (χ : G → ℂ) (h : IsIrrChar ℂ ℂ χ) : ∃ n : ℕ, 0 < n ∧ χ 1 = n

theorem **irr_of_norm_one** : ∀ (G : Type) [Group G] [Finite G] (φ : G → ℂ), φ ∈ Virt ℂ ℂ G → ∑ᶠ g, φ g * star (φ g) = (Nat.card G : ℂ) → (IsIrrChar ℂ ℂ φ ∨ IsIrrChar ℂ ℂ (-φ)) ∧ (IsChar ℂ ℂ φ → IsIrrChar ℂ ℂ φ ∧ ∃ n : ℕ, 0 < n ∧ φ 1 = n)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IrreducibleOfVirtualCharacterNormOneStatement

theorem **_root_.GlobalTheory.T.irreducible_of_virtualCharacter_norm_one** : ∀ (G : Type) [Group G] [Finite G] (φ : G → ℂ), φ ∈ Virt ℂ ℂ G → ∑ᶠ g, φ g * star (φ g) = (Nat.card G : ℂ) → (IsIrrChar ℂ ℂ φ ∨ IsIrrChar ℂ ℂ (-φ)) ∧ (IsChar ℂ ℂ φ → IsIrrChar ℂ ℂ φ ∧ ∃ n : ℕ, 0 < n ∧ φ 1 = n)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsAbelianByPrimePowerGroupSubgroup

theorem **_root_.GlobalTheory.BI.AbByP.subgroup** : ∀ (G : Type) [Group G] (T : Subgroup G), _root_.GlobalTheory.BI.AbByP G → _root_.GlobalTheory.BI.AbByP T

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsAbelianByPrimePowerGroupSubgroupStatement

theorem **_root_.GlobalTheory.T.isAbelianByPrimePowerGroup_subgroup** : ∀ (G : Type) [Group G] (T : Subgroup G), AbByP G → AbByP T

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharOfIsPrimitiveRoot

theorem **isChar_of_isPrimitiveRoot** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] [IsAlgClosed L] (m : ℕ) (ζ : K), m ≠ 0 → IsPrimitiveRoot ζ m → ∀ (G : Type) [Group G] [Finite G], (∀ g : G, g ^ m = 1) → ∀ χ : G → L, IsChar L L χ → IsChar K L χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharTraceMulIdempotent

theorem **_root_.GlobalTheory.BI.isChar_trace_mul_idempotent** : ∀ (L : Type) [Field L] (T : Type) [Group T] (V : Type) [AddCommGroup V] [Module L V] [FiniteDimensional L V] (ρ : Representation L T V) (e : Module.End L V), e * e = e → (∀ t, ρ t * e = e * ρ t) → _root_.GlobalTheory.BI.IsChar L L (fun t => LinearMap.trace L V (ρ t * e))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterBrauerLiftAbelianByPrimePower

theorem **basis_smul_eq_smul_cancel** {k V ι : Type} [Field k] [AddCommGroup V] [Module k V] (b : Module.Basis ι k V) {x y : k} {i j : ι} (hx : x ≠ 0) (h : x • b i = y • b j) : i = j ∧ x = y

theorem **isChar_brauerLift** : ∀ (k : Type) [Field k] [IsAlgClosed k] (θ : k →* ℂ), (∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (G : Type) [Group G] [Finite G], AbByP G → (Nat.card G : k) ≠ 0 → ∀ (n : ℕ) (ρ : G →* GL (Fin n) k), IsChar ℂ ℂ (fun g => ((Matrix.charpoly (ρ g : Matrix (Fin n) (Fin n) k)).roots.map θ).sum)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterCuspidalRootSumRegular

private lemma **finsum_proper_invariant_eq_sum_range_finrank** {F : Type} [Field F] [Finite F] (d : ℕ) (g : Module.End F (Fin d → F)) (Ψ : ∀ W : Submodule F (Fin d → F), (∀ v ∈ W, g v ∈ W) → ℂ) : (∑ᶠ (W : { W : Submodule F (Fin d → F) // W ≠ ⊤ ∧ ∀ x ∈ W, g x ∈ W }), Ψ W.1 W.2.2) = ∑ a ∈ Finset.range d, ∑ᶠ (W : { W : Submodule F (Fin d → F) // Module.finrank F W = a ∧ ∀ v ∈ W, g v ∈ W }), Ψ W.1 W.2.2

private lemma **psi_rootSum_charpoly_mem_virt** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (k : ℕ) (d : ℕ) : (fun x : GL (Fin d) F => _root_.GlobalTheory.CuspGL.psi (_root_.GlobalTheory.CuspGL.rootSum θ k) (Fin d → F) (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F))) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ (GL (Fin d) F)

theorem **_root_.GlobalTheory.CuspGL.isChar_psi_rootSum_of_regular** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (d k : ℕ) (hd : 0 < d) (hk : ∀ i : ℕ, 0 < i → i < d → ¬ k ∗ Nat.card F ^ i ≡ k [MOD Nat.card F ^ d - 1]) : _root_.GlobalTheory.BI.IsChar ℂ ℂ (fun x : GL (Fin d) F => (-1) ^ (d - 1) ∗ _root_.GlobalTheory.CuspGL.psi (_root_.GlobalTheory.CuspGL.rootSum θ k) (Fin d → F) (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F))) ∧ _root_.GlobalTheory.CuspGL.ipGL (_root_.GlobalTheory.CuspGL.psi (_root_.GlobalTheory.CuspGL.rootSum θ k) (Fin d → F)) (_root_.GlobalTheory.CuspGL.psi (_root_.GlobalTheory.CuspGL.rootSum θ k) (Fin d → F)) = _root_.GlobalTheory.CuspGL.glCard F d

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterCuspidalRootSumRegularStatement

theorem **_root_.GlobalTheory.T.isCharacter_cuspidal_rootSum_regular** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ d k : ℕ, 0 < d → (∀ i : ℕ, 0 < i → i < d → ¬ k ∗ Nat.card F ^ i ≡ k [MOD Nat.card F ^ d - 1]) → IsChar ℂ ℂ (fun x : GL (Fin d) F => (-1) ^ (d - 1) ∗ psi (rootSum θ k) (Fin d → F) (Matrix.toLin' (x : Matrix (Fin d) (Fin d) F))) ∧ ipGL (psi (rootSum θ k) (Fin d → F)) (psi (rootSum θ k) (Fin d → F)) = glCard F d

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterEsymmRootsCharpoly

private lemma **det_submatrix_subsetEnum_eq_det_submatrix_val** {n j : ℕ} (A : Matrix (Fin n) (Fin n) ℂ) (S : Set.powersetCard (Fin n) j) : (A.submatrix (_root_.GlobalTheory.subsetEnum S) (_root_.GlobalTheory.subsetEnum S)).det = (A.submatrix (Subtype.val : {x // x ∈ S.1} → Fin n) Subtype.val).det

private lemma **trace_compoundMatrix_eq_sum_principal_minors** {n : ℕ} (j : ℕ) (A : Matrix (Fin n) (Fin n) ℂ) : (_root_.GlobalTheory.compoundMatrix j A).trace = ∑ s ∈ Finset.powersetCard j Finset.univ, (A.submatrix (Subtype.val : s → Fin n) Subtype.val).det

private lemma **esymm_roots_charpoly_eq_sum_principal_minors** {n : ℕ} (j : ℕ) (M : Matrix (Fin n) (Fin n) ℂ) : M.charpoly.roots.esymm j = ∑ s ∈ Finset.powersetCard j Finset.univ, (M.submatrix (Subtype.val : s → Fin n) Subtype.val).det

theorem **_root_.GlobalTheory.BI.isChar_esymm_roots_charpoly** : ∀ (G : Type) [Group G] (n j : ℕ) (ρ : G →* GL (Fin n) ℂ), _root_.GlobalTheory.BI.IsChar ℂ ℂ (fun g => (Matrix.charpoly (ρ g : Matrix (Fin n) (Fin n) ℂ)).roots.esymm j)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterFinsumConjSubgroupFunctionExtendByZer

private theorem **trace_block_permutation** (K : Type) [Field K] (X : Type) [Fintype X] [DecidableEq X] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (π : X → X) (A : X → W →ₗ[K] W) : LinearMap.trace K (X → W) (LinearMap.pi fun q => A q ∘ₗ LinearMap.proj (π q)) = ∑ q, if π q = q then LinearMap.trace K W (A q) else 0

theorem **cosetCocycle_mem** (g : G) (q : G ⧸ H) : q.out⁻¹ ∗ g ∗ (g⁻¹ • q).out ∈ H

theorem **cosetCocycle_one** (q : G ⧸ H) : cosetCocycle H 1 q = 1

theorem **cosetCocycle_mul** (a b : G) (q : G ⧸ H) : cosetCocycle H (a ∗ b) q = cosetCocycle H a q ∗ cosetCocycle H b (a⁻¹ • q)

theorem **inv_smul_eq_iff_conj_mem** (g : G) (q : G ⧸ H) : g⁻¹ • q = q ↔ q.out⁻¹ ∗ g ∗ q.out ∈ H

theorem **isChar_finsum_conj_ext0_of_isChar** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → L), IsChar K L ψ → IsChar K L (fun g : G => ∑ᶠ q : G ⧸ H, ext0 H ψ (q.out⁻¹ ∗ g ∗ q.out))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterFinsumConjSubgroupFunctionExtendByZerStatement

theorem **_root_.GlobalTheory.T.isCharacter_finsum_conj_subgroupFunctionExtendByZero** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → L), IsChar K L ψ → IsChar K L (fun g : G => ∑ᶠ q : G ⧸ H, ext0 H ψ (q.out⁻¹ ∗ g ∗ q.out))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldAbelianByPrimePower

theorem **isChar_of_abByP** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] [IsAlgClosed L] (m : ℕ) (ζ : K), m ≠ 0 → IsPrimitiveRoot ζ m → ∀ (G : Type) [Group G] [Finite G], (∀ g : G, g ^ m = 1) → AbByP G → ∀ χ : G → L, IsChar L L χ → IsChar K L χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldAbelianByPrimePowerStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_abelianByPrimePower** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] [IsAlgClosed L] (m : ℕ) (ζ : K), m ≠ 0 → IsPrimitiveRoot ζ m → ∀ (G : Type) [Group G] [Finite G], (∀ g : G, g ^ m = 1) → AbByP G → ∀ χ : G → L, IsChar L L χ → IsChar K L χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldInducedClassFunction

theorem **isChar_ind** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → L), IsChar K L ψ → IsChar K L (ind H ψ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldInducedClassFunctionStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_inducedClassFunction** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (H : Subgroup G) (ψ : H → L), IsChar K L ψ → IsChar K L (ind H ψ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldInductionIrreducible

theorem **_root_.GlobalTheory.BI.isChar_induction** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (Q : (G → L) → Prop), Q 0 → (∀ φ c : G → L, _root_.GlobalTheory.BI.IsIrrChar K L φ → _root_.GlobalTheory.BI.IsChar K L c → Q c → Q (φ + c)) → ∀ a : G → L, _root_.GlobalTheory.BI.IsChar K L a → Q a

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldInductionIrreducibleStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_induction_irreducible** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (Q : (G → L) → Prop), Q 0 → (∀ φ c : G → L, IsIrrChar K L φ → IsChar K L c → Q c → Q (φ + c)) → ∀ a : G → L, IsChar K L a → Q a

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldLinearOfPrimitiveRoot

theorem **isChar_linear** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (m : ℕ) (ζ : K) (lam : G →* Lˣ), m ≠ 0 → IsPrimitiveRoot ζ m → (∀ g : G, g ^ m = 1) → IsChar K L (fun g => (lam g : L))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldLinearOfPrimitiveRootStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_linear_of_primitiveRoot** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (m : ℕ) (ζ : K) (lam : G →* Lˣ), m ≠ 0 → IsPrimitiveRoot ζ m → (∀ g : G, g ^ m = 1) → IsChar K L (fun g => (lam g : L))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldOfRepresentation

theorem **transportRep_apply** (ρ : Representation K G V) (e : V ≃ₗ[K] W) (g : G) (w : W) : transportRep ρ e g w = e (ρ g (e.symm w))

theorem **character_transportRep** [FiniteDimensional K V] [FiniteDimensional K W] (ρ : Representation K G V) (e : V ≃ₗ[K] W) (g : G) : (transportRep ρ e).character g = ρ.character g

theorem **isIrreducible_transportRep** (ρ : Representation K G V) (e : V ≃ₗ[K] W) (h : ρ.IsIrreducible) : (transportRep ρ e).IsIrreducible

theorem **_root_.GlobalTheory.BI.isChar_and_isIrrChar_of_rep** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (ρ : Representation K G V), _root_.GlobalTheory.BI.IsChar K L (fun g => algebraMap K L (ρ.character g)) ∧ (ρ.IsIrreducible → _root_.GlobalTheory.BI.IsIrrChar K L (fun g => algebraMap K L (ρ.character g)))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldOfRepresentationStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_of_representation** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (ρ : Representation K G V), IsChar K L (fun g => algebraMap K L (ρ.character g)) ∧ (ρ.IsIrreducible → IsIrrChar K L (fun g => algebraMap K L (ρ.character g)))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldOfVirtualAndCharacter

private theorem **isChar_of_isIrrChar** (K L : Type) [Field K] [Field L] [Algebra K L] {G : Type} [Group G] (φ : G → L) (hφ : _root_.GlobalTheory.BI.IsIrrChar K L φ) : _root_.GlobalTheory.BI.IsChar K L φ

theorem **_root_.GlobalTheory.BI.isChar_of_isChar_of_mem_virt** (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (χ : G → L) (hχ : _root_.GlobalTheory.BI.IsChar L L χ) (hv : χ ∈ _root_.GlobalTheory.BI.Virt K L G) : _root_.GlobalTheory.BI.IsChar K L χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldOfVirtualAndCharacterStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_of_virtual_and_character** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (χ : G → L), IsChar L L χ → χ ∈ Virt K L G → IsChar K L χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldRestrictClassFunctionOne

theorem **isChar_restrict_conj_one** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (χ : G → L), IsChar K L χ → (∀ (H : Type) [Group H] (f : H →* G), IsChar K L (fun h => χ (f h))) ∧ (∀ g x : G, χ (x⁻¹ * g * x) = χ g) ∧ ∃ n : ℕ, χ 1 = (n : L) ∧ (n = 0 → χ = 0)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldRestrictClassFunctionOneStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_restrict_classFunction_one** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (χ : G → L), IsChar K L χ → (∀ (H : Type) [Group H] (f : H →* G), IsChar K L (fun h => χ (f h))) ∧ (∀ g x : G, χ (x⁻¹ * g * x) = χ g) ∧ ∃ n : ℕ, χ 1 = (n : L) ∧ (n = 0 → χ = 0)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldZeroAndAdd

private theorem **exists_representation_character_eq_add** (K : Type) [Field K] (G : Type) [Group G] (n m : ℕ) (ρ : Representation K G (Fin n → K)) (σ : Representation K G (Fin m → K)) : ∃ τ : Representation K G (Fin (n + m) → K), ∀ g, τ.character g = ρ.character g + σ.character g

theorem **_root_.GlobalTheory.BI.isChar_zero_and_add** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G], _root_.GlobalTheory.BI.IsChar K L (0 : G → L) ∧ ∀ a b : G → L, _root_.GlobalTheory.BI.IsChar K L a → _root_.GlobalTheory.BI.IsChar K L b → _root_.GlobalTheory.BI.IsChar K L (a + b)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsCharacterOverSubfieldZeroAndAddStatement

theorem **_root_.GlobalTheory.T.isCharacterOverSubfield_zero_and_add** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G], IsChar K L (0 : G → L) ∧ ∀ a b : G → L, IsChar K L a → IsChar K L b → IsChar K L (a + b)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsPrimitiveEndClassFunctionInvariantSubspaceMobi

theorem **finsum_sub** {α : Type} [Fintype α] (p : α → Prop) (f : {a // p a} → ℂ) : ∑ᶠ x, f x = ∑ a, if h : p a then f ⟨a, h⟩ else 0

theorem **psi_trivial** {F : Type} [Field F] (s : F[X] → ℂ) (hs : IsAdditive s) (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [Subsingleton V] (g : Module.End F V) : psi s V g = 0

theorem **res_top** {F : Type} [Field F] (s : F[X] → ℂ) {V : Type} [AddCommGroup V] [Module F V] [FiniteDimensional F V] (k : Module.End F V) : res s ⊤ k = psi s V k

theorem **step** {F : Type} [Field F] [Finite F] (s : F[X] → ℂ) (hs : IsAdditive s) (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (ih : ∀ W : Submodule F V, W ≠ ⊤ → IsPrimitive (psi s W)) : IsPrimitive (psi s V)

theorem **isPrimitive_psi** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → IsPrimitive (psi s V)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsPrimitiveEndClassFunctionInvariantSubspaceMobiStatement

theorem **_root_.GlobalTheory.T.isPrimitiveEndClassFunction_invariantSubspaceMobiusInverse** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (s : F[X] → ℂ), IsAdditive s → IsPrimitive (psi s V)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsPrimitiveExtendZeroOfSumUnipotentRadical

theorem **exists_linearEquiv_mem_iff_of_finrank_eq** {F V : Type} [Field F] [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A B : Submodule F V) (h : Module.finrank F A = Module.finrank F B) : ∃ e : V ≃ₗ[F] V, ∀ x, e x ∈ B ↔ x ∈ A

theorem **finsum_nilRad_extZero_eq_zero** (F : Type) [Field F] [Finite F] (d : ℕ) (χ : GL (Fin d) F → ℂ) (hχ : IsChar ℂ ℂ χ) (A : Submodule F (Fin d → F)) (H : Subgroup (GL (Fin d) F)) (hb : Set.BijOn (fun u : GL (Fin d) F => Matrix.toLin' (u : Matrix (Fin d) (Fin d) F) - 1) (H : Set (GL (Fin d) F)) (nilRad A)) (h0 : ∑ᶠ u ∈ (H : Set (GL (Fin d) F)), χ u = 0) (g : Module.End F (Fin d → F)) (hg : ∀ x ∈ A, g x ∈ A) : ∑ᶠ h ∈ nilRad A, extZero χ (g + h) = 0

theorem **isPrimitive_extZero_of_sum_unipotent** : ∀ (F : Type) [Field F] [Finite F] (d : ℕ) (χ : GL (Fin d) F → ℂ), IsChar ℂ ℂ χ → (∀ a : ℕ, 0 < a → a < d → ∑ᶠ u ∈ {u : GL (Fin d) F | ∀ i j : Fin d, (u : Matrix (Fin d) (Fin d) F) i j = (if i = j then 1 else 0) ∨ (i.val < a ∧ a ≤ j.val)}, χ u = 0) → IsPrimitive (extZero χ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsPrimitiveExtendZeroOfSumUnipotentRadicalStatement

theorem **_root_.GlobalTheory.T.isPrimitive_extendZero_of_sum_unipotentRadical** : ∀ (F : Type) [Field F] [Finite F] (d : ℕ) (χ : GL (Fin d) F → ℂ), IsChar ℂ ℂ χ → (∀ a : ℕ, 0 < a → a < d → ∑ᶠ u ∈ {u : GL (Fin d) F | ∀ i j : Fin d, (u : Matrix (Fin d) (Fin d) F) i j = (if i = j then 1 else 0) ∨ (i.val < a ∧ a ≤ j.val)}, χ u = 0) → IsPrimitive (extZero χ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsSimpleModuleOfFinrankEnd

theorem **isSimpleModule_of_finrank_end** {R M : Type} [Ring R] [Algebra ℂ R] [AddCommGroup M] [Module R M] [Module ℂ M] [IsScalarTower ℂ R M] [SMulCommClass R ℂ M] [IsSemisimpleModule R M] (h : Module.finrank ℂ (Module.End R M) = 1) : IsSimpleModule R M

Development.AbelianVarietyTheory.Proofs.CharacterTheory.IsUnitIffCharpolyCoeffZeroNe

theorem **_root_.GlobalTheory.CuspGL.ipGL_charpoly_eq_finsum_mass** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ) (s : F[X] → ℂ) : _root_.GlobalTheory.CuspGL.ipGL φ (fun g => s (LinearMap.charpoly g)) = ∑ᶠ P ∈ {P : F[X] | P.Monic ∧ P.natDegree = Module.finrank F V ∧ P.coeff 0 ≠ 0}, _root_.GlobalTheory.CuspGL.mass φ P * star (s P)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.KerAevalAddStabilizerNilradicalSocle

theorem **pow_add_nilRad** {A : Submodule F V} {g h : Module.End F V} (hg : ∀ x ∈ A, g x ∈ A) (hh : h ∈ nilRad A) (n : ℕ) : (∀ x ∈ A, ((g + h) ^ n) x = (g ^ n) x ∧ (g ^ n) x ∈ A) ∧ ∀ x, ((g + h) ^ n) x - (g ^ n) x ∈ A

theorem **aeval_add_nilRad** {A : Submodule F V} {g h : Module.End F V} (hg : ∀ x ∈ A, g x ∈ A) (hh : h ∈ nilRad A) (p : F[X]) : (∀ x ∈ A, aeval (g + h) p x = aeval g p x) ∧ ∀ x, aeval (g + h) p x - aeval g p x ∈ A

theorem **ker_aeval_add_nilRad** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g h : Module.End F V) (f : F[X]), h ∈ nilRad (LinearMap.ker (aeval g f)) → LinearMap.ker (aeval g f) ≤ LinearMap.ker (aeval (g + h) f) ∧ (LinearMap.ker (aeval (g + h) f) = LinearMap.ker (aeval g f) → ∀ j : ℕ, Module.finrank F (LinearMap.ker (aeval g (f ^ j))) = Module.finrank F (LinearMap.ker (aeval (g + h) (f ^ j))))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.KerAevalAddStabilizerNilradicalSocleStatement

theorem **_root_.GlobalTheory.T.ker_aeval_add_stabilizerNilradical_socle** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g h : Module.End F V) (f : F[X]), h ∈ nilRad (LinearMap.ker (aeval g f)) → LinearMap.ker (aeval g f) ≤ LinearMap.ker (aeval (g + h) f) ∧ (LinearMap.ker (aeval (g + h) f) = LinearMap.ker (aeval g f) → ∀ j : ℕ, Module.finrank F (LinearMap.ker (aeval g (f ^ j))) = Module.finrank F (LinearMap.ker (aeval (g + h) (f ^ j))))

Development.AbelianVarietyTheory.Proofs.CharacterTheory.LinearMapShortExactCommute

theorem **trace_eq_add_of_shortExact** (K : Type) [Field K] (U : Type) [AddCommGroup U] [Module K U] [FiniteDimensional K U] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (i : U →ₗ[K] V) (p : V →ₗ[K] W) (a : U →ₗ[K] U) (b : V →ₗ[K] V) (c : W →ₗ[K] W) (hi : Function.Injective i) (hp : Function.Surjective p) (hex : LinearMap.range i = LinearMap.ker p) (hb : b ∘ₗ i = i ∘ₗ a) (hc : p ∘ₗ b = c ∘ₗ p) : LinearMap.trace K V b = LinearMap.trace K U a + LinearMap.trace K W c

Development.AbelianVarietyTheory.Proofs.CharacterTheory.LinearMapShortExactCommuteStatement

theorem **_root_.GlobalTheory.T.linear_map_short_exact_commute** : ∀ (K : Type) [Field K] (U : Type) [AddCommGroup U] [Module K U] [FiniteDimensional K U] (V : Type) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (W : Type) [AddCommGroup W] [Module K W] [FiniteDimensional K W] (i : U →ₗ[K] V) (p : V →ₗ[K] W) (a : U →ₗ[K] U) (b : V →ₗ[K] V) (c : W →ₗ[K] W), Function.Injective i → Function.Surjective p → LinearMap.range i = LinearMap.ker p → b ∘ₗ i = i ∘ₗ a → p ∘ₗ b = c ∘ₗ p → LinearMap.trace K V b = LinearMap.trace K U a + LinearMap.trace K W c

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MatrixTraceIffIsCharacterOverSubfieldComplex

theorem **matrixChar_iff_isChar** : ∀ (G : Type) [Group G] (χ : G → ℂ), (∃ (n : ℕ) (ρ : G →* GL (Fin n) ℂ), ∀ x, χ x = Matrix.trace (ρ x : Matrix (Fin n) (Fin n) ℂ)) ↔ IsChar ℂ ℂ χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MatrixTraceIffIsCharacterOverSubfieldComplexStatement

theorem **_root_.GlobalTheory.T.matrixTrace_iff_isCharacterOverSubfield_complex** : ∀ (G : Type) [Group G] (χ : G → ℂ), (∃ (n : ℕ) (ρ : G →* GL (Fin n) ℂ), ∀ x, χ x = Matrix.trace (ρ x : Matrix (Fin n) (Fin n) ℂ)) ↔ IsChar ℂ ℂ χ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MemVirtualCharactersIffSubCharacters

theorem **mem_virt_iff_sub** (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (f : G → L) : f ∈ Virt K L G ↔ ∃ a b : G → L, IsChar K L a ∧ IsChar K L b ∧ f = a − b

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MemVirtualCharactersIffSubCharactersStatement

theorem **_root_.GlobalTheory.T.mem_virtualCharacters_iff_sub_characters** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] (G : Type) [Group G] (f : G → L), f ∈ Virt K L G ↔ ∃ a b : G → L, IsChar K L a ∧ IsChar K L b ∧ f = a − b

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MemVirtualCharactersOfAbelianByPrimePower

theorem **mem_virt_of_forall_abByP** (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G] (hH : ∀ H : Subgroup G, AbByP H → ∀ ψ : H → L, IsChar L L ψ → IsChar K L ψ) : ∀ χ : G → L, IsChar L L χ → χ ∈ Virt K L G

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MemVirtualCharactersOfAbelianByPrimePowerStatement

theorem **_root_.GlobalTheory.T.mem_virtualCharacters_of_abelianByPrimePower** : ∀ (K L : Type) [Field K] [Field L] [Algebra K L] [CharZero K] [CharZero L] (G : Type) [Group G] [Finite G], (∀ H : Subgroup G, AbByP H → ∀ ψ : H → L, IsChar L L ψ → IsChar K L ψ) → ∀ χ : G → L, IsChar L L χ → χ ∈ Virt K L G

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MonicFactorizationConvolutionCancelOfOne

theorem **finite_pairs** {F : Type} [Field F] (P : F[X]) (hP : P ≠ 0) : {x : F[X] × F[X] | x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P}.Finite

theorem **conv_cancel** : ∀ (F : Type) [Field F] (a b m : F[X] → ℂ), m 1 = 1 → (∀ P : F[X], P.Monic → conv a m P = conv b m P) → ∀ P : F[X], P.Monic → a P = b P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MonicFactorizationConvolutionCancelOfOneStatement

theorem **_root_.GlobalTheory.T.monicFactorizationConvolution_cancel_of_one** : ∀ (F : Type) [Field F] (a b m : F[X] → ℂ), m 1 = 1 → (∀ P : F[X], P.Monic → conv a m P = conv b m P) → ∀ P : F[X], P.Monic → a P = b P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MonicFactorizationConvolutionFiniteCommAssocAdd

theorem **finite_monic_factorisations** (P : F[X]) : {x : F[X] × F[X] | x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P}.Finite

theorem **conv_eq_sum** (a b : F[X] → ℂ) (P : F[X]) : conv a b P = ∑ x ∈ (finite_monic_factorisations P).toFinset, a x.1 * b x.2

theorem **conv_comm** (a b : F[X] → ℂ) (P : F[X]) : conv a b P = conv b a P

theorem **conv_assoc** (a b c : F[X] → ℂ) (P : F[X]) : conv (conv a b) c P = conv a (conv b c) P

theorem **conv_add_left** (a b c : F[X] → ℂ) (P : F[X]) : conv (a + b) c P = conv a c P + conv b c P

theorem **conv_sub_left** (a b c : F[X] → ℂ) (P : F[X]) : conv (a − b) c P = conv a c P − conv b c P

theorem **conv_basic** : ∀ (F : Type) [Field F] (P : F[X]), P.Monic → {x : F[X] × F[X] | x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P}.Finite ∧ ∀ a b c : F[X] → ℂ, conv a b P = conv b a P ∧ conv (conv a b) c P = conv a (conv b c) P ∧ conv (a + b) c P = conv a c P + conv b c P ∧ conv (a − b) c P = conv a c P − conv b c P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MonicFactorizationConvolutionFiniteCommAssocAddStatement

theorem **_root_.GlobalTheory.T.monicFactorizationConvolution_finite_comm_assoc_add** : ∀ (F : Type) [Field F] (P : F[X]), P.Monic → {x : F[X] × F[X] | x.1.Monic ∧ x.2.Monic ∧ x.1 * x.2 = P}.Finite ∧ ∀ a b c : F[X] → ℂ, conv a b P = conv b a P ∧ conv (conv a b) c P = conv a (conv b c) P ∧ conv (a + b) c P = conv a c P + conv b c P ∧ conv (a − b) c P = conv a c P − conv b c P

Development.AbelianVarietyTheory.Proofs.CharacterTheory.MultisetPowerSumNewtonIdentity

private theorem **sum_antidiagonal_filter_ioo_eq_sum_ioo** (k : ℕ) (g : ℕ → ℕ → ℂ) : ∑ a ∈ Finset.HasAntidiagonal.antidiagonal k with a.1 ∈ Set.Ioo 0 k, g a.1 a.2 = ∑ i ∈ Finset.Ioo 0 k, g i (k − i)

private theorem **aeval_psum_eq_sum_map_pow** (s : Multiset ℂ) (n : ℕ) : MvPolynomial.aeval (fun x : s => (x : ℂ)) (MvPolynomial.psum s ℂ n) = (s.map (· ^ n)).sum

theorem **multiset_newton_psum** (s : Multiset ℂ) (r : ℕ) (hr : 0 < r) : (s.map (· ^ r)).sum = (−1) ^ (r + 1) * (r : ℂ) * s.esymm r − ∑ i ∈ Finset.Ioo 0 r, (−1) ^ i * s.esymm i * (s.map (· ^ (r − i))).sum

Development.AbelianVarietyTheory.Proofs.CharacterTheory.NoNormEightOfFourierVanishingOrderFifteen

theorem **no_norm_eight_of_fourier_vanishing_order_fifteen** : ∀ (ω : ℂ) (c : ℕ → ℤ), IsPrimitiveRoot ω 15 → ∑ k ∈ Finset.range 15, c k ^ 2 = 8 → (∀ j : ℕ, j < 15 → ¬ 5 ∣ j → ∑ k ∈ Finset.range 15, (c k : ℂ) * ω ^ (j * k) = 0) → False

Development.AbelianVarietyTheory.Proofs.CharacterTheory.NoNormEightOfFourierVanishingOrderFifteenStatement

theorem **_root_.GlobalTheory.T.false_of_sum_sq_eq_eight_of_fourier_vanish_order_15** : ∀ (ω : ℂ) (c : ℕ → ℤ), IsPrimitiveRoot ω 15 → ∑ k ∈ Finset.range 15, c k ^ 2 = 8 → (∀ j : ℕ, j < 15 → ¬ 5 ∣ j → ∑ k ∈ Finset.range 15, (c k : ℂ) * ω ^ (j * k) = 0) → False

Development.AbelianVarietyTheory.Proofs.CharacterTheory.NoNormTwelveOrderSixtythree

private theorem **sum_range_63_primitiveRoot_pow_eq_ite** (ω : ℂ) (hω : IsPrimitiveRoot ω 63) (n : ℕ) : ∑ j ∈ Finset.range 63, ω ^ (j * n) = if 63 ∣ n then (63 : ℂ) else 0

theorem **fourier_inv_sixtythree** (ω : ℂ) (c : ℕ → ℤ) (hω : IsPrimitiveRoot ω 63) (m : ℕ) (hm : m < 63) : ∑ j ∈ Finset.range 63, (∑ k ∈ Finset.range 63, (c k : ℂ) * ω ^ (j * k)) * ω ^ (j * (62 * m)) = 63 * (c m : ℂ)

theorem **mixed_diff_sixtythree** (ω : ℂ) (c : ℕ → ℤ) (hω : IsPrimitiveRoot ω 63) (h : ∀ j : ℕ, j < 63 → ¬ 21 ∣ j → ¬ 9 ∣ j → ∑ k ∈ Finset.range 63, (c k : ℂ) * ω ^ (j * k) = 0) (m : ℕ) (hm : m < 53) : c (m + 10) − c (m + 3) = c (m + 7) − c m

private theorem **apply_add_21_eq_of_fourier_vanish_order_63** (ω : ℂ) (c : ℕ → ℤ) (hω : IsPrimitiveRoot ω 63) (h : ∀ j : ℕ, j < 63 → ¬ 21 ∣ j → ¬ 9 ∣ j → ∑ k ∈ Finset.range 63, (c k : ℂ) * ω ^ (j * k) = 0) (n : ℕ) (hn : n < 42) : c (n + 21) = c n

theorem **no_norm_twelve_order_sixtythree** : ∀ (ω : ℂ) (c : ℕ → ℤ), IsPrimitiveRoot ω 63 → ∑ k ∈ Finset.range 63, c k ^ 2 = 12 → (∀ j : ℕ, j < 63 → ¬ 21 ∣ j → ¬ 9 ∣ j → ∑ k ∈ Finset.range 63, (c k : ℂ) * ω ^ (j * k) = 0) → False

Development.AbelianVarietyTheory.Proofs.CharacterTheory.NoNormTwelveOrderSixtythreeStatement

theorem **_root_.GlobalTheory.T.false_of_sum_sq_eq_12_of_fourier_vanish_order_63** : ∀ (ω : ℂ) (c : ℕ → ℤ), IsPrimitiveRoot ω 63 → ∑ k ∈ Finset.range 63, c k ^ 2 = 12 → (∀ j : ℕ, j < 63 → ¬ 21 ∣ j → ¬ 9 ∣ j → ∑ k ∈ Finset.range 63, (c k : ℂ) * ω ^ (j * k) = 0) → False

Development.AbelianVarietyTheory.Proofs.CharacterTheory.OneMemClosureInducedCentralByPrimePower

theorem **one_mem_closure_ind_centralByPrime** : ∀ (G : Type) [Group G] [Finite G], (1 : G → ℂ) ∈ AddSubgroup.closure {f : G → ℂ | ∃ (E : Subgroup G) (ψ : E → ℂ), (∃ l : ℕ, l.Prime ∧ ∀ y : E, ∃ j : ℕ, y ^ l ^ j ∈ Subgroup.center E) ∧ ψ ∈ Virt ℂ ℂ E ∧ f = ind E ψ}

Development.AbelianVarietyTheory.Proofs.CharacterTheory.OneMemOfMulMemOfForallPrimeNotDvd

theorem **one_mem_of_mul_mem_of_forall_prime_not_dvd** (X : Type) [Finite X] (S : AddSubgroup (X → ℤ)) (hmul : ∀ f ∈ S, ∀ g ∈ S, f * g ∈ S) (hp : ∀ (x : X) (p : ℕ), p.Prime → ∃ f ∈ S, ¬ (p : ℤ) ∣ f x) : (1 : X → ℤ) ∈ S

Development.AbelianVarietyTheory.Proofs.CharacterTheory.OneMemOfMulMemOfForallPrimeNotDvdStatement

theorem **_root_.GlobalTheory.T.one_mem_of_mul_mem_of_forall_prime_exists_not_dvd** : ∀ (X : Type) [Finite X] (S : AddSubgroup (X → ℤ)), (∀ f ∈ S, ∀ g ∈ S, f * g ∈ S) → (∀ (x : X) (p : ℕ), p.Prime → ∃ f ∈ S, ¬ (p : ℤ) ∣ f x) → (1 : X → ℤ) ∈ S

Development.AbelianVarietyTheory.Proofs.CharacterTheory.OneMemVirtualCharactersOfIntegralCombination

theorem **one_mem_of_integral_combination** : ∀ (G : Type) [Group G] [Finite G] (S : AddSubgroup (G → ℂ)), S ≤ Virt ℂ ℂ G → ∀ (ι : Type) [Fintype ι] (r : ι → ℂ) (s : ι → G → ℂ), (∀ i, IsIntegral ℤ (r i)) → (∀ i, s i ∈ S) → (∀ g, ∑ i, r i * s i g = 1) → (1 : G → ℂ) ∈ S

Development.AbelianVarietyTheory.Proofs.CharacterTheory.OneMemVirtualCharactersOfIntegralCombinationStatement

theorem **_root_.GlobalTheory.T.one_mem_virtualCharacters_of_integral_combination** : ∀ (G : Type) [Group G] [Finite G] (S : AddSubgroup (G → ℂ)), S ≤ Virt ℂ ℂ G → ∀ (ι : Type) [Fintype ι] (r : ι → ℂ) (s : ι → G → ℂ), (∀ i, IsIntegral ℤ (r i)) → (∀ i, s i ∈ S) → (∀ g, ∑ i, r i * s i g = 1) → (1 : G → ℂ) ∈ S

Development.AbelianVarietyTheory.Proofs.CharacterTheory.OrthogonalIntVectorsNotAgreeOnRegular

private lemma **abs_sum_le_sum_sq_int** (s : Finset ℕ) (a : ℕ → ℤ) : |∑ k ∈ s, a k| ≤ ∑ k ∈ s, a k ^ 2

theorem **orthogonal_int_vectors_not_agree_on_regular** : ∀ (q n : ℕ) (ω : ℂ) (a b : ℕ → ℤ), 2 ≤ q → 0 < n → IsPrimitiveRoot ω (q ^ n - 1) → ∑ k ∈ Finset.range (q ^ n - 1), a k ^ 2 = n → ∑ k ∈ Finset.range (q ^ n - 1), b k ^ 2 = n → ∑ k ∈ Finset.range (q ^ n - 1), a k * b k = 0 → 0 < (∑ k ∈ Finset.range (q ^ n - 1), a k) * (∑ k ∈ Finset.range (q ^ n - 1), b k) → (∀ j : ℕ, (∀ e : ℕ, 0 < e → e < n → ¬ j * q ^ e ≡ j [MOD q ^ n - 1]) → ∑ k ∈ Finset.range (q ^ n - 1), (a k : ℂ) * ω ^ (j * k) = ∑ k ∈ Finset.range (q ^ n - 1), (b k : ℂ) * ω ^ (j * k)) → False

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ParabolicInductionTrivialPreservesVirtualCharact

theorem **_root_.GlobalTheory.ParabVirt.sum_fibres** {G X : Type} [Group G] [Fintype G] [Fintype X] (H : Subgroup G) (π : G → X) (hs : Function.Surjective π) (hπ : ∀ s t, π s = π t ↔ s⁻¹ * t ∈ H) (f : X → ℂ) : ∑ t, f (π t) = (Nat.card H : ℂ) * ∑ x, f x

theorem **lin_apply** (x : GL (Fin n) F) (v : Fin n → F) : lin x v = Matrix.toLin' (x : Matrix (Fin n) (Fin n) F) v

theorem **_root_.GlobalTheory.ParabVirt.lin_mul** (x y : GL (Fin n) F) (v : Fin n → F) : _root_.GlobalTheory.ParabVirt.lin (x * y) v = _root_.GlobalTheory.ParabVirt.lin x (_root_.GlobalTheory.ParabVirt.lin y v)

theorem **_root_.GlobalTheory.ParabVirt.lin_one** (v : Fin n → F) : _root_.GlobalTheory.ParabVirt.lin (1 : GL (Fin n) F) v = v

theorem **lin_inv** (x : GL (Fin n) F) (v : Fin n → F) : lin x⁻¹ v = (lin x).symm v

theorem **_root_.GlobalTheory.ParabVirt.lin_surjective** (e : (Fin n → F) ≃ₗ[F] (Fin n → F)) : ∃ x : GL (Fin n) F, _root_.GlobalTheory.ParabVirt.lin x = e

theorem **_root_.GlobalTheory.ParabVirt.map_eq_of_stable** (x : GL (Fin n) F) (W : Submodule F (Fin n → F)) (h : ∀ v ∈ W, _root_.GlobalTheory.ParabVirt.lin x v ∈ W) : W.map (_root_.GlobalTheory.ParabVirt.lin x).toLinearMap = W

theorem **_root_.GlobalTheory.ParabVirt.mem_stab** (W : Submodule F (Fin n → F)) (x : GL (Fin n) F) : x ∈ _root_.GlobalTheory.ParabVirt.stab W ↔ ∀ v ∈ W, Matrix.toLin' (x : Matrix (Fin n) (Fin n) F) v ∈ W

private theorem **lin_mul_toLinearMap** (x y : GL (Fin n) F) : (_root_.GlobalTheory.ParabVirt.lin (x * y)).toLinearMap = (_root_.GlobalTheory.ParabVirt.lin x).toLinearMap ∘ₗ (_root_.GlobalTheory.ParabVirt.lin y).toLinearMap

theorem **_root_.GlobalTheory.ParabVirt.map_eq_map_iff** (W : Submodule F (Fin n → F)) (s t : GL (Fin n) F) : W.map (_root_.GlobalTheory.ParabVirt.lin s).toLinearMap = W.map (_root_.GlobalTheory.ParabVirt.lin t).toLinearMap ↔ s⁻¹ * t ∈ _root_.GlobalTheory.ParabVirt.stab W

theorem **_root_.GlobalTheory.ParabVirt.conj_mem_stab_iff** (W : Submodule F (Fin n → F)) (x t : GL (Fin n) F) : t⁻¹ * x * t ∈ _root_.GlobalTheory.ParabVirt.stab W ↔ x ∈ _root_.GlobalTheory.ParabVirt.stab (W.map (_root_.GlobalTheory.ParabVirt.lin t).toLinearMap)

private theorem **exists_gl_map_lin_eq_of_finrank_eq** (W₀ W : Submodule F (Fin n → F)) (h : Module.finrank F W₀ = Module.finrank F W) : ∃ t : GL (Fin n) F, W₀.map (_root_.GlobalTheory.ParabVirt.lin t).toLinearMap = W

theorem **_root_.GlobalTheory.ParabVirt.parab_virt** : ∀ (F : Type) [Field F] [Finite F] (Φ : ∀ (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V], Module.End F V → ℂ), (∀ (V V' : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] [AddCommGroup V'] [Module F V'] [FiniteDimensional F V'] (e : V ≃ₗ[F] V') (g : Module.End F V), Φ V' (e.conj g) = Φ V g) → ∀ a n : ℕ, (fun x : GL (Fin a) F => Φ (Fin a → F) (Matrix.toLin' (x : Matrix (Fin a) (Fin a) F))) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ (GL (Fin a) F) → (fun x : GL (Fin n) F => ∑ᶠ W : {W : Submodule F (Fin n → F) // Module.finrank F W = a ∧ ∀ v ∈ W, Matrix.toLin' (x : Matrix (Fin n) (Fin n) F) v ∈ W}, Φ W.1 ((Matrix.toLin' (x : Matrix (Fin n) (Fin n) F)).restrict W.2.2)) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ (GL (Fin n) F)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ParsevalBoundIntegerVectorSmallFourierSupport

theorem **parsevalSep_orth** (N : ℕ) (ω : ℂ) (hN : 0 < N) (hω : IsPrimitiveRoot ω N) (k l : ℕ) (hk : k < N) (hl : l < N) : ∑ j ∈ Finset.range N, ω ^ (j * k) * (starRingEnd ℂ ω) ^ (j * l) = if k = l then (N : ℂ) else 0

theorem **parsevalSep_identity** (N : ℕ) (ω : ℂ) (hN : 0 < N) (hω : IsPrimitiveRoot ω N) (c : ℕ → ℤ) : ∑ j ∈ Finset.range N, ‖∑ k ∈ Finset.range N, (c k : ℂ) * ω ^ (j * k)‖ ^ 2 = (N : ℝ) * ((∑ k ∈ Finset.range N, c k ^ 2 : ℤ) : ℝ)

theorem **parsevalSep_bound** : ∀ (n N : ℕ) (ω : ℂ) (c : ℕ → ℤ) (S : Finset ℕ), 0 < n → 0 < N → IsPrimitiveRoot ω N → ∑ k ∈ Finset.range N, c k ^ 2 = 2 * n → |∑ k ∈ Finset.range N, c k| ≤ (n : ℤ) - 1 → (∀ j : ℕ, 0 < j → j < N → j ∉ S → ∑ k ∈ Finset.range N, (c k : ℂ) * ω ^ (j * k) = 0) → 2 * n * N ≤ (n - 1) ^ 2 + 4 * n ^ 2 * S.card

Development.AbelianVarietyTheory.Proofs.CharacterTheory.ParsevalBoundIntegerVectorSmallFourierSupportStatement

theorem **_root_.GlobalTheory.T.parseval_bound_integerVector_smallFourierSupport** : ∀ (n N : ℕ) (ω : ℂ) (c : ℕ → ℤ) (S : Finset ℕ), 0 < n → 0 < N → IsPrimitiveRoot ω N → ∑ k ∈ Finset.range N, c k ^ 2 = 2 * n → |∑ k ∈ Finset.range N, c k| ≤ (n : ℤ) - 1 → (∀ j : ℕ, 0 < j → j < N → j ∉ S → ∑ k ∈ Finset.range N, (c k : ℂ) * ω ^ (j * k) = 0) → 2 * n * N ≤ (n - 1) ^ 2 + 4 * n ^ 2 * S.card

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimaryEndomorphismCyclicDecompositionFinrankKer

theorem **finrank_ker_lsmul_pow_quotient** {F : Type} [Field F] (f : F[X]) (hf : f ≠ 0) (e j : ℕ) : Module.finrank F (LinearMap.ker ((LinearMap.lsmul F[X] (F[X] ⧸ Ideal.span {f ^ e}) (f ^ j)).restrictScalars F)) = f.natDegree * min j e

theorem **primary_structure** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) (f : F[X]) (m : ℕ), Irreducible f → aeval g (f ^ m) = 0 → ∃ (d : ℕ) (e : Fin d → ℕ), (∀ i, 0 < e i) ∧ Nonempty (Module.AEval' g ≃ₗ[F[X]] ((i : Fin d) → F[X] ⧸ Ideal.span {f ^ e i})) ∧ ∀ j : ℕ, Module.finrank F (LinearMap.ker (aeval g (f ^ j))) = f.natDegree * ∑ i, min j (e i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimaryEndomorphismCyclicDecompositionFinrankKerStatement

theorem **_root_.GlobalTheory.T.primary_endomorphism_cyclic_decomposition_finrank_ker** : ∀ (F : Type) [Field F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (g : Module.End F V) (f : F[X]) (m : ℕ), Irreducible f → aeval g (f ^ m) = 0 → ∃ (d : ℕ) (e : Fin d → ℕ), (∀ i, 0 < e i) ∧ Nonempty (Module.AEval' g ≃ₗ[F[X]] ((i : Fin d) → F[X] ⧸ Ideal.span {f ^ e i})) ∧ ∀ j : ℕ, Module.finrank F (LinearMap.ker (aeval g (f ^ j))) = f.natDegree * ∑ i, min j (e i)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimitiveClassFunctionExpansionRootSums

theorem **finsum_mem_mul_sub_sum** {G : Type} [Finite G] (S : Set G) (c : ℂ) (φ : G → ℂ) (T : Finset ℕ) (a : ℕ → ℂ) (ψ : ℕ → G → ℂ) : ∑ᶠ g ∈ S, (c * φ g - ∑ k ∈ T, a k * ψ k g) = c * (∑ᶠ g ∈ S, φ g) - ∑ k ∈ T, a k * ∑ᶠ g ∈ S, ψ k g

theorem **card_filter_mul_pow_modEq** (q n i l : ℕ) (hq : 1 < q) (hn : 0 < n) (hi : i ≤ n) : ((Finset.range (q ^ n - 1)).filter (fun k => k * q ^ i ≡ l [MOD (q ^ n - 1)])).card = 1

theorem **sum_mul_card_filter_modEq** (q n l : ℕ) (hq : 1 < q) (hn : 0 < n) (A : ℕ → ℂ) (hA : ∀ k i : ℕ, k * q ^ i ≡ l [MOD (q ^ n - 1)] → A k = A l) : ∑ k ∈ Finset.range (q ^ n - 1), A k * (((Finset.range n).filter (fun i => k * q ^ i ≡ l [MOD (q ^ n - 1)])).card : ℂ) = n * A l

theorem **exists_minpoly_eq_of_irreducible** (F : Type) [Field F] [Finite F] (f : F[X]) (hm : f.Monic) (hf : Irreducible f) (h0 : f.coeff 0 ≠ 0) (m : ℕ) : ∃ α : AlgebraicClosure F, α ^ (Nat.card F ^ (f.natDegree * m) - 1) = 1 ∧ minpoly F α = f

theorem **eq_zero_of_ipGL_rootSum_eq_zero** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (φ : Module.End F V → ℂ) (hc : IsClassFn φ) (hp : IsPrimitive φ) (hn : 0 < Module.finrank F V) (h : ∀ l, ipGL φ (fun g : Module.End F V => rootSum θ l (LinearMap.charpoly g)) = 0) (g : Module.End F V) (hg : IsUnit g) : φ g = 0

theorem **expand_primitive** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (φ : Module.End F V → ℂ) (hc : IsClassFn φ) (hp : IsPrimitive φ) (hn : 0 < Module.finrank F V) (g : Module.End F V) (hg : IsUnit g) : glCard F (Module.finrank F V) * (Module.finrank F V : ℂ) * φ g = ∑ k ∈ Finset.range (Nat.card F ^ Module.finrank F V - 1), ipGL φ (fun g : Module.End F V => rootSum θ k (LinearMap.charpoly g)) * psi (rootSum θ k) V g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimitiveClassFunctionExpansionRootSumsStatement

theorem **_root_.GlobalTheory.T.primitiveClassFunction_expansion_rootSums** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → 0 < Module.finrank F V → ∀ g : Module.End F V, IsUnit g → glCard F (Module.finrank F V) * (Module.finrank F V : ℂ) * φ g = ∑ k ∈ Finset.range (Nat.card F ^ Module.finrank F V - 1), ipGL φ (fun g : Module.End F V => rootSum θ k (LinearMap.charpoly g)) * psi (rootSum θ k) V g

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimitiveEndClassFunctionEqZeroOfPrimary

theorem **primitive_eq_zero_of_primary** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → ∀ (f : F[X]) (m : ℕ), Irreducible f → (∀ g : Module.End F V, LinearMap.charpoly g = f ^ m → aeval g f = 0 → φ g = 0) → ∀ g : Module.End F V, LinearMap.charpoly g = f ^ m → φ g = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimitiveEndClassFunctionEqZeroOfPrimaryStatement

theorem **_root_.GlobalTheory.T.primitiveEndClassFunction_eq_zero_of_primary** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → ∀ (f : F[X]) (m : ℕ), Irreducible f → (∀ g : Module.End F V, LinearMap.charpoly g = f ^ m → aeval g f = 0 → φ g = 0) → ∀ g : Module.End F V, LinearMap.charpoly g = f ^ m → φ g = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimitiveEndClassFunctionZeroOfCharpolyMassZero

theorem **finsum_mem_lin_comb_of_finite** {α : Type} [Finite α] (S : Set α) (a b : ℂ) (u v : α → ℂ) : ∑ᶠ x ∈ S, (a * u x - b * v x) = a * (∑ᶠ x ∈ S, u x) - b * ∑ᶠ x ∈ S, v x

theorem **eq_zero_of_mass_charpoly_eq_zero** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ) (hc : IsClassFn φ) (hp : IsPrimitive φ) (g : Module.End F V) (hm : mass φ (LinearMap.charpoly g) = 0) : φ g = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.PrimitiveEndClassFunctionZeroOfCharpolyMassZeroStatement

theorem **_root_.GlobalTheory.T.primitiveEndClassFunction_zero_of_charpoly_mass_zero** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (φ : Module.End F V → ℂ), IsClassFn φ → IsPrimitive φ → ∀ g : Module.End F V, mass φ (LinearMap.charpoly g) = 0 → φ g = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RegularIffNatDegreeAndExistsEllipticGL

theorem **pow_card_pow_eq_iff_natDegree_minpoly_dvd** (F : Type) [Field F] [Finite F] (α : AlgebraicClosure F) (e : ℕ) : α ^ Nat.card F ^ e = α ↔ (minpoly F α).natDegree ∣ e

theorem **exists_GL_charpoly_eq_minpoly** (F K : Type) [Field F] [Field K] [Algebra F K] (α : K) (hx : IsIntegral F α) (h0 : α ≠ 0) (n : ℕ) (hn : (minpoly F α).natDegree = n) : ∃ x : GL (Fin n) F, Matrix.charpoly (x : Matrix (Fin n) (Fin n) F) = minpoly F α

theorem **regular_iff_natDegree_and_exists_elliptic** : ∀ (F : Type) [Field F] [Finite F] (n : ℕ), 0 < n → ∀ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n - 1) = 1}, ((∀ e : ℕ, 0 < e → e < n → α ^ Nat.card F ^ e ≠ α) ↔ (minpoly F α).natDegree = n) ∧ ((minpoly F α).natDegree = n → ∃ x : GL (Fin n) F, Matrix.charpoly (x : Matrix (Fin n) (Fin n) F) = minpoly F α)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RegularIffNatDegreeAndExistsEllipticGLStatement

theorem **_root_.GlobalTheory.T.regular_iff_natDegree_and_exists_elliptic_GL** : ∀ (F : Type) [Field F] [Finite F] (n : ℕ), 0 < n → ∀ α ∈ {α : AlgebraicClosure F | α ^ (Nat.card F ^ n - 1) = 1}, ((∀ e : ℕ, 0 < e → e < n → α ^ Nat.card F ^ e ≠ α) ↔ (minpoly F α).natDegree = n) ∧ ((minpoly F α).natDegree = n → ∃ x : GL (Fin n) F, Matrix.charpoly (x : Matrix (Fin n) (Fin n) F) = minpoly F α)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RootPowerSumCharacterAdditiveAndXSubOnePow

theorem **rootSum_isAdditive_and_pow_X_sub_one** : ∀ (F : Type) [Field F] (θ : AlgebraicClosure F →* ℂ) (k : ℕ), IsAdditive (rootSum θ k) ∧ ∀ j : ℕ, rootSum θ k ((X - C 1 : F[X]) ^ j) = j

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RootPowerSumCharacterAdditiveAndXSubOnePowStatement

theorem **_root_.GlobalTheory.T.rootPowerSumCharacter_additive_and_X_sub_one_pow** : ∀ (F : Type) [Field F] (θ : AlgebraicClosure F →* ℂ) (k : ℕ), IsAdditive (rootSum θ k) ∧ ∀ j : ℕ, rootSum θ k ((X - C 1 : F[X]) ^ j) = j

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RootPowerSumCharpolyMemVirtualCharactersGL

private theorem **sylow_normal_of_prime_pow_mem_center** (E : Type) [Group E] [Finite E] (p : ℕ) [hp : Fact p.Prime] (P : Sylow p E) (hE : ∀ (y : E), ∃ j, y ^ p ^ j ∈ Subgroup.center E) : (P : Subgroup E).Normal

private theorem **exists_p_regular_split_of_normal_sylow_commute** (E : Type) [Group E] [Finite E] (p : ℕ) [Fact p.Prime] (P : Sylow p E) [(P : Subgroup E).Normal] (hcomm : ∀ x y : E, x ∈ (P : Subgroup E) → ¬ p ∣ orderOf y → Commute y x) : ∃ H : Subgroup E, ∃ π : E →* H, ¬ p ∣ Nat.card H ∧ ∀ g, ∃ s, ∃ m : ℕ, g = (π g : E) * s ∧ Commute (π g : E) s ∧ s ^ p ^ m = 1

private theorem **exists_p_regular_split_of_pow_mem_center** (E : Type) [Group E] [Finite E] (p : ℕ) [hp : Fact p.Prime] (hE : ∃ l, Nat.Prime l ∧ ∀ (y : E), ∃ j, y ^ l ^ j ∈ Subgroup.center E) : ∃ H : Subgroup E, ∃ π : E →* H, ¬ p ∣ Nat.card H ∧ ∀ g, ∃ s, ∃ m : ℕ, g = (π g : E) * s ∧ Commute (π g : E) s ∧ s ^ p ^ m = 1

private theorem **powerSum_roots_charpoly_complex_mem_virt** (G : Type) [Group G] [Finite G] (n : ℕ) (ρ : G →* GL (Fin n) ℂ) (r : ℕ) : (fun g => (Multiset.map (fun x => x ^ r) ((ρ g : Matrix (Fin n) (Fin n) ℂ).charpoly.roots)).sum) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G

private theorem **powerSum_roots_charpoly_mem_virt_of_abByP** (k : Type) [Field k] [IsAlgClosed k] (θ : k →* ℂ) (hθ : ∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (H : Type) [Group H] [Finite H] (hab : _root_.GlobalTheory.BI.AbByP H) (hc : (Nat.card H : k) ≠ 0) (n : ℕ) (ρ : H →* GL (Fin n) k) (r : ℕ) : (fun g => (Multiset.map (fun α => θ α ^ r) ((ρ g : Matrix (Fin n) (Fin n) k).charpoly.roots)).sum) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ H

private theorem **powerSum_roots_charpoly_mem_virt_of_split** (k : Type) [Field k] [IsAlgClosed k] (p : ℕ) [Fact p.Prime] [CharP k p] (θ : k →* ℂ) (hθ : ∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (E : Type) [Group E] [Finite E] (hab : _root_.GlobalTheory.BI.AbByP E) (H : Subgroup E) (π : E →* H) (hc : (Nat.card H : k) ≠ 0) (hdec : ∀ g, ∃ s, ∃ m : ℕ, g = (π g : E) * s ∧ Commute (π g : E) s ∧ s ^ p ^ m = 1) (n : ℕ) (ρ : E →* GL (Fin n) k) (r : ℕ) : (fun g => (Multiset.map (fun α => θ α ^ r) ((ρ g : Matrix (Fin n) (Fin n) k).charpoly.roots)).sum) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ E

private theorem **rootSum_charpoly_mem_virt_of_pow_mem_center** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (n k : ℕ) (E : Subgroup (GL (Fin n) F)) (hE : ∃ l, Nat.Prime l ∧ ∀ (y : E), ∃ j, y ^ l ^ j ∈ Subgroup.center E) : (fun x : E => _root_.GlobalTheory.CuspGL.rootSum θ k (Matrix.charpoly ((x : GL (Fin n) F) : Matrix (Fin n) (Fin n) F))) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ E

theorem **_root_.GlobalTheory.CuspGL.rootSum_charpoly_mem_virt** (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ) (hθ : ∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (n k : ℕ) : (fun x : GL (Fin n) F => _root_.GlobalTheory.CuspGL.rootSum θ k (Matrix.charpoly (x : Matrix (Fin n) (Fin n) F))) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ (GL (Fin n) F)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RootPowerSumCharpolyMemVirtualCharactersGLStatement

theorem **_root_.GlobalTheory.T.rootPowerSum_charpoly_mem_virtualCharacters_GL** : ∀ (F : Type) [Field F] [Finite F] (θ : AlgebraicClosure F →* ℂ), (∀ x y : AlgebraicClosure F, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ n k : ℕ, (fun x : GL (Fin n) F => rootSum θ k (Matrix.charpoly (x : Matrix (Fin n) (Fin n) F))) ∈ Virt ℂ ℂ (GL (Fin n) F)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RootsCharpolyPowEqMapPow

lemma **card_roots_charpoly** (M : Matrix (Fin n) (Fin n) k) : Multiset.card M.charpoly.roots = n

lemma **det_scalar_sub** (M : Matrix (Fin n) (Fin n) k) (t : k) : (Matrix.scalar (Fin n) t - M).det = (M.charpoly.roots.map (fun z => t - z)).prod

lemma **det_aeval_prod** (M : Matrix (Fin n) (Fin n) k) (s : Multiset k) : (aeval M (s.map (fun a => C a - X)).prod).det = (M.charpoly.roots.map (fun z => eval z (s.map (fun a => C a - X)).prod)).prod

lemma **det_aeval** (M : Matrix (Fin n) (Fin n) k) (q : k[X]) : (aeval M q).det = (M.charpoly.roots.map (fun z => eval z q)).prod

theorem **roots_charpoly_pow** (k : Type) [Field k] [IsAlgClosed k] (n r : ℕ) (M : Matrix (Fin n) (Fin n) k) : (Matrix.charpoly (M ^ r)).roots = (Matrix.charpoly M).roots.map (· ^ r)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.RootsCharpolyPowEqMapPowStatement

theorem **_root_.GlobalTheory.T.roots_charpoly_pow_eq_map_pow** : ∀ (k : Type) [Field k] [IsAlgClosed k] (n r : ℕ) (M : Matrix (Fin n) (Fin n) k), (Matrix.charpoly (M ^ r)).roots = (Matrix.charpoly M).roots.map (· ^ r)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SeparationNumericalIneqCuspidalGL

theorem **sepIneq_four_mul_add_five_le** (k : ℕ) (hk : 5 ≤ k) : 4 * k + 5 ≤ 2 ^ k

theorem **sepIneq_sum_lt** (q n : ℕ) (hq : 2 ≤ q) : ∑ p ∈ n.primeFactors, (q ^ (n / p) - 2) < q ^ (n / 2 + 1)

theorem **sepIneq_large** (q n : ℕ) (hq : 2 ≤ q) (hn : 11 ≤ n) : (n - 1) ^ 2 + 4 * n ^ 2 * ∑ p ∈ n.primeFactors, (q ^ (n / p) - 2) < 2 * n * (q ^ n - 1)

theorem **sepIneq_sub** (t k c : ℕ) (hc : c ≤ 2) (hk : 0 < k) : (((2 + t) ^ k - c : ℕ) : ℤ) = (2 + (t : ℤ)) ^ k - c

theorem **sep_numerical_ineq** : ∀ q n : ℕ, 2 ≤ q → 0 < n → ¬ (q = 2 ∧ (n = 4 ∨ n = 6)) → (n − 1) ^ 2 + 4 * n ^ 2 * ∑ p ∈ n.primeFactors, (q ^ (n / p) − 2) < 2 * n * (q ^ n − 1)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SeparationNumericalIneqCuspidalGLStatement

theorem **_root_.GlobalTheory.T.separation_numerical_ineq_cuspidalGL** : ∀ q n : ℕ, 2 ≤ q → 0 < n → ¬ (q = 2 ∧ (n = 4 ∨ n = 6)) → (n − 1) ^ 2 + 4 * n ^ 2 * ∑ p ∈ n.primeFactors, (q ^ (n / p) − 2) < 2 * n * (q ^ n − 1)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.StarVirtualCharacterEqApplyInv

theorem **_root_.GlobalTheory.CuspGL.star_virt_eq_inv** : ∀ (G : Type) [Group G] [Finite G] (χ : G → ℂ), χ ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G → ∀ g, star (χ g) = χ g⁻¹

Development.AbelianVarietyTheory.Proofs.CharacterTheory.StarVirtualCharacterEqApplyInvStatement

theorem **_root_.GlobalTheory.T.star_virtualCharacter_eq_apply_inv** : ∀ (G : Type) [Group G] [Finite G] (χ : G → ℂ), χ ∈ Virt ℂ ℂ G → ∀ g, star (χ g) = χ g⁻¹

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SumCharpolyRootsEqFinsumEigenspaceFinrank

theorem **sum_roots_charpoly_eq_finsum_eigenspace** (k : Type) [Field k] [IsAlgClosed k] (V : Type) [AddCommGroup V] [Module k V] [FiniteDimensional k V] (g : Module.End k V) (N : ℕ) (hN : (N : k) ≠ 0) (hg : g ^ N = 1) (h : k → ℂ) : ((LinearMap.charpoly g).roots.map h).sum = ∑ᶠ μ ∈ {μ : k | μ ^ N = 1}, (Module.finrank k (Module.End.eigenspace g μ) : ℂ) * h μ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SumCharpolyRootsEqFinsumEigenspaceFinrankStatement

theorem **_root_.GlobalTheory.T.sum_charpoly_roots_eq_finsum_eigenspace_finrank** : ∀ (k : Type) [Field k] [IsAlgClosed k] (V : Type) [AddCommGroup V] [Module k V] [FiniteDimensional k V] (g : Module.End k V) (N : ℕ), (N : k) ≠ 0 → g ^ N = 1 → ∀ h : k → ℂ, ((LinearMap.charpoly g).roots.map h).sum = ∑ᶠ μ ∈ {μ : k | μ ^ N = 1}, (Module.finrank k (Module.End.eigenspace g μ) : ℂ) * h μ

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SumConjIdempotentEqCardSmul

theorem **sum_conj_idempotent_eq_card_smul** (L : Type) [Field L] (G : Type) [Group G] [Fintype G] (V : Type) [AddCommGroup V] [Module L V] (ρ : Representation L G V) (T : Subgroup G) (e : Module.End L V) (hirr : ρ.IsIrreducible) (he0 : e ≠ 0) (hee : e * e = e) (hT : ∀ t ∈ T, ρ t * e = e * ρ t) (hoff : ∀ g ∉ T, e * ρ g * e = 0) : ∑ x : G, ρ x * e * ρ x⁻¹ = (Nat.card T : L) • (1 : Module.End L V)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SumGaussianMulProd

lemma **prod_succ_succ** (x : ℂ) (n a : ℕ) : ∏ i ∈ Finset.range (a + 1), (x ^ (n + 1) − x ^ i) = (x ^ (n + 1) − 1) * x ^ a * ∏ i ∈ Finset.range a, (x ^ n − x ^ i)

lemma **den_ne** (x : ℂ) (hx : x ≠ 0) (h1 : ∀ k : ℕ, x ^ (k + 1) − 1 ≠ 0) (a : ℕ) : ∏ i ∈ Finset.range a, (x ^ a − x ^ i) ≠ 0

lemma **sum_eq** (x : ℂ) (hx : x ≠ 0) (h1 : ∀ k : ℕ, x ^ (k + 1) − 1 ≠ 0) (n : ℕ) : ∑ a ∈ Finset.range n, (∏ i ∈ Finset.range (a + 1), (x ^ n − x ^ i)) / (∏ i ∈ Finset.range (a + 1), (x ^ (a + 1) − x ^ i)) * ∏ i ∈ Finset.range a, (1 − x ^ (i + 1)) = n

theorem **sum_gaussian_mul_prod** : ∀ (q n : ℕ) (N : ℕ → ℕ), 2 ≤ q → (∀ a, a ≤ n → N a * ∏ i ∈ Finset.range a, (q ^ a − q ^ i) = ∏ i ∈ Finset.range a, (q ^ n − q ^ i)) → ∑ a ∈ Finset.Icc 1 n, (N a : ℂ) * ∏ i ∈ Finset.Ico 1 a, (1 − (q : ℂ) ^ i) = n

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SumPrimitiveEndClassFunctionRestrictMixedSubspac

theorem **exists_comp_eq_of_ker_le** {F : Type} [Field F] {U X Y : Type} [AddCommGroup U] [Module F U] [AddCommGroup X] [Module F X] [AddCommGroup Y] [Module F Y] (p : U →ₗ[F] X) (f : U →ₗ[F] Y) (h : LinearMap.ker p ≤ LinearMap.ker f) : ∃ k : X →ₗ[F] Y, ∀ u, k (p u) = f u

theorem **exists_nilRad_extend** {F : Type} [Field F] {V : Type} [AddCommGroup V] [Module F V] (A W : Submodule F V) (h' : Module.End F W) (hh : h' ∈ nilRad (A.comap W.subtype)) : ∃ L : Module.End F V, L ∈ nilRad A ∧ ∀ x : W, L x = h' x

theorem **sum_restrict_eq_zero_of_mixed** (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A W : Submodule F V) (g : Module.End F V) (hg : ∀ x ∈ A, g x ∈ A) (h1 : W ⊓ A ≠ ⊥) (h2 : ¬ W ≤ A) (φ : Module.End F W → ℂ) (hφ : IsPrimitive φ) : ∑ᶠ h : {h : Module.End F V // h ∈ nilRad A ∧ ∀ x ∈ W, (g + h) x ∈ W}, φ ((g + h.1).restrict h.2.2) = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.SumPrimitiveEndClassFunctionRestrictMixedSubspacStatement

theorem **_root_.GlobalTheory.T.sum_primitiveEndClassFunction_restrict_mixed_subspace** : ∀ (F : Type) [Field F] [Finite F] (V : Type) [AddCommGroup V] [Module F V] [FiniteDimensional F V] (A W : Submodule F V) (g : Module.End F V), (∀ x ∈ A, g x ∈ A) → W ⊓ A ≠ ⊥ → ¬ W ≤ A → ∀ φ : Module.End F W → ℂ, IsPrimitive φ → ∑ᶠ h : {h : Module.End F V // h ∈ nilRad A ∧ ∀ x ∈ W, (g + h) x ∈ W}, φ ((g + h.1).restrict h.2.2) = 0

Development.AbelianVarietyTheory.Proofs.CharacterTheory.TraceLiftMonomialMatrixEqSumEigenvalues

theorem **ncard_pow_eq_one** (k : Type) [Field k] [IsAlgClosed k] (N : ℕ) (hN : (N : k) ≠ 0) : {μ : k | μ ^ N = 1}.Finite ∧ {μ : k | μ ^ N = 1}.ncard = N

theorem **bijOn_pow_eq_one** (k : Type) [Field k] [IsAlgClosed k] (θ : k →* ℂ) (hθ : ∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) (N : ℕ) (hN : (N : k) ≠ 0) : Set.BijOn θ {μ : k | μ ^ N = 1} {μ : ℂ | μ ^ N = 1}

theorem **trace_lift_monomial** : ∀ (k : Type) [Field k] [IsAlgClosed k] (θ : k →* ℂ), (∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (n N : ℕ) (σ : Equiv.Perm (Fin n)) (c : Fin n → k), (∀ j, c j ≠ 0) → (N : k) ≠ 0 → (Matrix.of fun i j => if i = σ j then c j else 0 : Matrix (Fin n) (Fin n) k) ^ N = 1 → (Matrix.of fun i j => if i = σ j then θ (c j) else 0 : Matrix (Fin n) (Fin n) ℂ) ^ N = 1 → Matrix.trace (Matrix.of fun i j => if i = σ j then θ (c j) else 0 : Matrix (Fin n) (Fin n) ℂ) = ((Matrix.charpoly (Matrix.of fun i j => if i = σ j then c j else 0 : Matrix (Fin n) (Fin n) k)).roots.map θ).sum

Development.AbelianVarietyTheory.Proofs.CharacterTheory.TraceLiftMonomialMatrixEqSumEigenvaluesStatement

theorem **_root_.GlobalTheory.T.trace_lift_monomialMatrix_eq_sum_eigenvalues** : ∀ (k : Type) [Field k] [IsAlgClosed k] (θ : k →* ℂ), (∀ x y : k, x ≠ 0 → y ≠ 0 → θ x = θ y → x = y) → ∀ (n N : ℕ) (σ : Equiv.Perm (Fin n)) (c : Fin n → k), (∀ j, c j ≠ 0) → (N : k) ≠ 0 → (Matrix.of fun i j => if i = σ j then c j else 0 : Matrix (Fin n) (Fin n) k) ^ N = 1 → (Matrix.of fun i j => if i = σ j then θ (c j) else 0 : Matrix (Fin n) (Fin n) ℂ) ^ N = 1 → Matrix.trace (Matrix.of fun i j => if i = σ j then θ (c j) else 0 : Matrix (Fin n) (Fin n) ℂ) = ((Matrix.charpoly (Matrix.of fun i j => if i = σ j then c j else 0 : Matrix (Fin n) (Fin n) k)).roots.map θ).sum

Development.AbelianVarietyTheory.Proofs.CharacterTheory.VirtualCharacterInnerEqIntMulCard

private lemma **exists_nat_sum_mul_apply_inv_eq_mul_card** (G : Type) [Group G] [Fintype G] (x y : G → ℂ) (hx : _root_.GlobalTheory.BI.IsChar ℂ ℂ x) (hy : _root_.GlobalTheory.BI.IsChar ℂ ℂ y) : ∃ n : ℕ, ∑ g, x g * y g⁻¹ = n * (Nat.card G : ℂ)

theorem **_root_.GlobalTheory.BI.finsum_mul_star_virt_int** : ∀ (G : Type) [Group G] [Finite G] (a b : G → ℂ), a ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G → b ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G → ∃ m : ℤ, ∑ᶠ g, a g * star (b g) = m * (Nat.card G : ℂ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.VirtualCharacterInnerEqIntMulCardStatement

theorem **_root_.GlobalTheory.T.virtualCharacter_inner_eq_int_mul_card** : ∀ (G : Type) [Group G] [Finite G] (a b : G → ℂ), a ∈ Virt ℂ ℂ G → b ∈ Virt ℂ ℂ G → ∃ m : ℤ, ∑ᶠ g, a g * star (b g) = m * (Nat.card G : ℂ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.VirtualCharacterIrreducibleExpansionNormSq

private theorem **exists_irreducible_expansion_of_isCharacter_lemma_3** {G : Type} [Group G] [Finite G] (a : G → ℂ) (ha : _root_.GlobalTheory.BI.IsChar ℂ ℂ a) : ∃ c : (G → ℂ) →₀ ℤ, (∀ χ ∈ c.support, _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ) ∧ a = c.sum (fun χ m => m • χ)

private theorem **exists_irreducible_expansion_of_mem_lemma** {G : Type} [Group G] [Finite G] (φ : G → ℂ) (hφ : φ ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G) : ∃ c : (G → ℂ) →₀ ℤ, (∀ χ ∈ c.support, _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ) ∧ φ = c.sum (fun χ m => m • χ)

private theorem **sum_mul_inv_irreducible_lemma_2** {G : Type} [Group G] [Fintype G] (χ ψ : G → ℂ) (hχ : _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ) (hψ : _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ ψ) : ∑ g, χ g * ψ g⁻¹ = if χ = ψ then (Nat.card G : ℂ) else 0

theorem **_root_.GlobalTheory.BI.virt_expand** : ∀ (G : Type) [Group G] [Finite G] (φ : G → ℂ), φ ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G → ∃ c : (G → ℂ) →₀ ℤ, (∀ χ ∈ c.support, _root_.GlobalTheory.BI.IsIrrChar ℂ ℂ χ) ∧ φ = c.sum (fun χ m => m • χ) ∧ ∑ᶠ g, φ g * star (φ g) = (c.sum fun _ m => (m : ℂ) ^ 2) * (Nat.card G : ℂ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.VirtualCharacterIrreducibleExpansionNormSqStatement

theorem **_root_.GlobalTheory.T.virtualCharacter_irreducible_expansion_normSq** : ∀ (G : Type) [Group G] [Finite G] (φ : G → ℂ), φ ∈ Virt ℂ ℂ G → ∃ c : (G → ℂ) →₀ ℤ, (∀ χ ∈ c.support, IsIrrChar ℂ ℂ χ) ∧ φ = c.sum (fun χ m => m • χ) ∧ ∑ᶠ g, φ g * star (φ g) = (c.sum fun _ m => (m : ℂ) ^ 2) * (Nat.card G : ℂ)

Development.AbelianVarietyTheory.Proofs.CharacterTheory.VirtualCharactersMulComapConjIndLinear

theorem **_root_.GlobalTheory.BI.isChar_mem_virt_complex** {G : Type} [Group G] {χ : G → ℂ} (h : _root_.GlobalTheory.BI.IsChar ℂ ℂ χ) : χ ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G

theorem **_root_.GlobalTheory.BI.isChar_mul_complex** {G : Type} [Group G] {a b : G → ℂ} (ha : _root_.GlobalTheory.BI.IsChar ℂ ℂ a) (hb : _root_.GlobalTheory.BI.IsChar ℂ ℂ b) : _root_.GlobalTheory.BI.IsChar ℂ ℂ (a * b)

private lemma **trace_mulLeft_complex_lemma** (z : ℂ) : LinearMap.trace ℂ ℂ (Algebra.lmul ℂ ℂ z) = z

theorem **_root_.GlobalTheory.BI.isChar_monoidHom_complex** {G : Type} [Group G] (lam : G →* ℂˣ) : _root_.GlobalTheory.BI.IsChar ℂ ℂ (fun g => (lam g : ℂ))

theorem **_root_.GlobalTheory.BI.subgroupFunctionExtendByZero_add** {G : Type} [Group G] (H : Subgroup G) (a b : H → ℂ) (g : G) : _root_.GlobalTheory.BI.ext0 H (a + b) g = _root_.GlobalTheory.BI.ext0 H a g + _root_.GlobalTheory.BI.ext0 H b g

theorem **_root_.GlobalTheory.BI.virt_mul_comap_conj_ind_linear** : ∀ (G : Type) [Group G] [Finite G], (∀ a ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G, ∀ b ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G, a * b ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G) ∧ (∀ a ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G, ∀ (H : Type) [Group H] (f : H →* G), (fun h => a (f h)) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ H) ∧ (∀ a ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G, ∀ g x : G, a (x⁻¹ * g * x) = a g) ∧ (∀ (H : Subgroup G) (ψ : H → ℂ), ψ ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ H → _root_.GlobalTheory.BI.ind H ψ ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G) ∧ ∀ lam : G →* ℂˣ, (fun g => (lam g : ℂ)) ∈ _root_.GlobalTheory.BI.Virt ℂ ℂ G

Development.AbelianVarietyTheory.Proofs.CharacterTheory.VirtualCharactersMulComapConjIndLinearStatement

theorem **_root_.GlobalTheory.T.virtualCharacters_mul_comap_conj_ind_linear** : ∀ (G : Type) [Group G] [Finite G], (∀ a ∈ Virt ℂ ℂ G, ∀ b ∈ Virt ℂ ℂ G, a * b ∈ Virt ℂ ℂ G) ∧ (∀ a ∈ Virt ℂ ℂ G, ∀ (H : Type) [Group H] (f : H →* G), (fun h => a (f h)) ∈ Virt ℂ ℂ H) ∧ (∀ a ∈ Virt ℂ ℂ G, ∀ g x : G, a (x⁻¹ * g * x) = a g) ∧ (∀ (H : Subgroup G) (ψ : H → ℂ), ψ ∈ Virt ℂ ℂ H → ind H ψ ∈ Virt ℂ ℂ G) ∧ ∀ lam : G →* ℂˣ, (fun g => (lam g : ℂ)) ∈ Virt ℂ ℂ G

Development.AbelianVarietyTheory.Proofs.CharacterTheory.W8AddTransposeEqE8

theorem **w8_add_transpose_eq_e8** : W8 + W8ᵀ = E8

theorem **e8_transpose** : E8ᵀ = E8

theorem **e8_det_eq_one** : E8.det = 1

theorem **e8inv_mul_e8** : E8inv * E8 = 1

theorem **e8_mul_e8inv** : E8 * E8inv = 1

theorem **E8roots_transpose_mul_self** : E8rootsᵀ * E8roots = (4 : ℤ) • E8

theorem **e8_posDef** : (E8.map (fun x : ℤ => (x : ℝ))).PosDef

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.AbelianProjectiveBaseChange

theorem **isProjBy_of_isPullback** {S : Scheme.{0}} (P : PolAV S) {R : Type} [CommRing R] {t : Spec (CommRingCat.of R) ⟶ S} {X : Scheme.{0}} {π : X ⟶ P.A.left} {f : X ⟶ Spec (CommRingCat.of R)} (h : IsPullback π f P.A.hom t) : IsProjBy f

theorem **abelian_projective_baseChange** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), (IsProjBy P.A.hom ∧ IsSeparated P.A.hom) ∧ ∀ (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of R)), IsPullback π f (fst P.A P.A).left t → IsProjBy f ∧ IsSeparated f ∧ Smooth f ∧ UniversallyClosed f ∧ Surjective f ∧ IsPullback (π ≫ (snd P.A P.A).left) f P.A.hom (t ≫ P.A.hom)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.AbelianVarietyTorsionCardEqPowProof

theorem **_root_.GlobalTheory.abMulN_holds** : _root_.GlobalTheory.Published.AbMulNSt

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.AbelianVarietyTorsionCardEqPowProofStatement

theorem **_root_.GlobalTheory.T.abelianVarietyTorsionCardEqPow** : GlobalTheory.Published.AbMulNSt

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.AcyclicSliceSecCubeOfIsAffineOpen

theorem **restrict_torsion** (hM : M.IsLB) {V W W' : Y.Opens} (hW : IsAffineOpen W) (hW' : IsAffineOpen W') (h : W ≤ V) (e : W' ≤ W) (a : Γ(Y, V)) (ha : W ⊓ Y.basicOpen a ≤ W') : (∀ y : SecOn M W, SecOn.restrict M e y = 0 → ∃ n : ℕ, rs Y h (a ^ n) • y = 0) ∧ ∀ z : SecOn M W', ∃ (n : ℕ) (y : SecOn M W), SecOn.restrict M e y = rs Y h (a.trans h) (a ^ n) • z

theorem **secCubeLE_exactAt** (hM : M.IsLB) {κ : Type} [LinearOrder κ] [Finite κ] (V : Finset κ → Y.Opens) (hV : Antitone V) (ha : ∀ s, IsAffineOpen (V s)) (U : κ → Y.Opens) (hU : V ∅ ≤ ⊔ k, U k) (hVU : ∀ k s, V s ⊓ U k ≤ V (insert k s)) (p : ℕ) : (cubeComplex (secCubeLE M V hV)).ExactAt p

theorem **acyclic_slice_secCube_of_isAffineOpen** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (μ κ : Type) [LinearOrder μ] [LinearOrder κ] [Fintype κ] (D : μ → Y.Opens) (e : κ ↪o μ) (s₀ : Finset μ), M.IsLB → (⊓ i ∈ s₀, D i) ≤ ⊔ k, D (e k) → (∀ t : Finset κ, IsAffineOpen (⊓ i ∈ s₀ ∪ t.map e.toEmbedding, D i)) → (cubeComplex (Cube.slice e s₀ (secCube f M D))).Acyclic

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.AlternatingMapSumSmulEqDetSmul

theorem **jetWedge_apply** (ω : M [⋀^Fin D]→ₗ[A] N) (v : Fin (D + 1) → A × M) : jetWedge ω v = ∑ i : Fin (D + 1), ((-1 : A) ^ (i : ℕ) * (v i).1) • ω (fun k => (v (i.succAbove k)).2)

theorem **jetWedge_shear** (ω : M [⋀^Fin D]→ₗ[A] N) (h : A) (u : M) (v : Fin (D + 1) → A × M) : jetWedge ω (fun i => (h * (v i).1, h • (v i).2 + (v i).1 • u)) = h ^ (D + 1) • jetWedge ω v

theorem **alternating_map_sum_smul_eq_det_smul** {n : ℕ} (ω : M [⋀^Fin n]→ₗ[A] N) (a : Matrix (Fin n) (Fin n) A) (w : Fin n → M) : ω (fun i => ∑ k, a i k • w k) = a.det • ω w

theorem **alternating_sub_mem_smul_top** (ω : M [⋀^Fin D]→ₗ[A] N) (I : Ideal A) (v w : Fin D → M) (hvw : ∀ k, v k - w k ∈ I • (⊤ : Submodule A M)) : ω v - ω w ∈ I • (⊤ : Submodule A N)

theorem **jacForm_eq_jetWedge** (f : Fin (D + 1) → A) : jacForm d ω f = jetWedge ω (fun i => (f i, d (f i)))

theorem **jacForm_mul** (h : A) (f : Fin (D + 1) → A) : jacForm d ω (fun i => h * f i) = h ^ (D + 1) • jacForm d ω f

theorem **form_d_mul_sub_mem** (I : Ideal A) (h : A) (g : Fin D → A) (hg : ∀ k, g k ∈ I) : ω (fun k => d (h * g k)) - h ^ D • ω (fun k => d (g k)) ∈ I • (⊤ : Submodule A N)

theorem **jacFormAlt_apply** (f : Fin (D + 1) → A) : jacobianMultilinearTo d ω f = jacForm d ω f

@[simp] theorem **altRestrictScalars_apply** (ω : M [⋀^Fin D]→ₗ[B] N) (v : Fin D → M) : altRestrictScalars A ω v = ω v

@[simp] theorem **exteriorPowerMapOver_ιMulti** (φ : L →ₗ[A] M) (v : Fin D → L) : exteriorPowerMapOver B D φ (exteriorPower.ιMulti A D v) = exteriorPower.ιMulti B D (fun k => φ (v k))

@[simp] theorem **wedgeContract_ιMulti** (φ : Fin D → L →ₗ[A] B) (v : Fin D → L) : wedgeContract φ (exteriorPower.ιMulti A D v) = (Matrix.of fun j k => φ k (v j)).det

theorem **jac_mul** (h : A) (f : Fin (D + 1) → A) : jac R (fun i => h * f i) = h ^ (D + 1) • jac R f

theorem **dWedge_mul_sub_mem** (I : Ideal A) (h : A) (g : Fin D → A) (hg : ∀ k, g k ∈ I) : dWedge R (fun k => h * g k) - h ^ D • dWedge R g ∈ I • (⊤ : Submodule A (⋀[A]^D Ω[A⁄R]))

@[simp] theorem **jacAlt_apply** (f : Fin (D + 1) → A) : jacobianMultilinear R A D f = jac R f

theorem **map_jac** (S B : Type*) [CommRing S] [CommRing B] [Algebra R S] [Algebra S B] [Algebra R B] [Algebra A B] [IsScalarTower R A B] [IsScalarTower R S B] [SMulCommClass S A B] (f : Fin (D + 1) → A) : exteriorPowerMapOver B D (KaehlerDifferential.map R S A B) (jac R f) = jac S (fun i => algebraMap A B (f i))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.AwayEquivAwayFracEqIff

theorem **ringPiece_mk_ext** {m' : ℤ} {σ : LB.Sec (lbZPow L m)} {τ : LB.Sec (lbZPow L m')} (h : m = m') (e : ∀ i, zfun σ i = zfun τ i) : GradedMonoid.mk (A

theorem **ringPiece_mk_assoc** {p q r : ℤ} (a : LB.Sec (lbZPow L p)) (b : LB.Sec (lbZPow L q)) (c : LB.Sec (lbZPow L r)) : GradedMonoid.mk (A

theorem **SecRing.of_mul_of** {m' : ℤ} (σ : LB.Sec (lbZPow L m)) (τ : LB.Sec (lbZPow L m')) : SecRing.of m σ * SecRing.of m' τ = SecRing.of (m + m') (zmul σ τ)

theorem **SecRing.one_def** : (1 : SecRing L) = SecRing.of 0 (zone L)

theorem **SecRing.of_pow** (σ : LB.Sec (lbZPow L m)) (k : ℕ) : SecRing.of m σ ^ k = SecRing.of (k * m) (zpowSec σ k)

theorem **modulePiece_mk_ext** {n' : ℤ} {a : Γ(twistZ F L n, V)} {b : Γ(twistZ F L n', V)} (h : n = n') (e : ∀ i, zcoord a i = zcoord b i) : GradedMonoid.mk (A

theorem **modulePiece_mk_one** (a : Γ(twistZ F L n, V)) : GradedMonoid.mk (A

theorem **modulePiece_mk_mul** {p q : ℤ} (σ : LB.Sec (lbZPow L p)) (τ : LB.Sec (lbZPow L q)) (a : Γ(twistZ F L n, V)) : GradedMonoid.mk (A

theorem **of_injective** (n : ℤ) : Function.Injective (of (F

theorem **of_smul_of** (σ : LB.Sec (lbZPow L m)) (a : Γ(twistZ F L n, V)) {d : ℤ} (h : n + m = d) : SecRing.of m σ • of n a = of d ((mulSecOfEq F σ h).app V a)

theorem **of_pow_smul_of** (σ : LB.Sec (lbZPow L m)) (k : ℕ) (a : Γ(twistZ F L n, V)) {d : ℤ} (h : n + k * m = d) : SecRing.of m σ ^ k • of n a = of d ((mulSecOfEq F (zpowSec σ k) h).app V a)

theorem **induction_on** {P : SecModule F L V → Prop} (y : SecModule F L V) (zero : P 0) (hof : ∀ n a, P (of n a)) (add : ∀ y z, P y → P z → P (y + z)) : P y

theorem **resAdd_of** (h : W ≤ V) (a : Γ(twistZ F L n, V)) : resAdd F L h (of n a) = of n (resM _ h a)

@[simp] theorem **res_of** (h : W ≤ V) (a : Γ(twistZ F L n, V)) : res F L h (of n a) = of n (resM _ h a)

theorem **res_apply** (h : W ≤ V) (y : SecModule F L V) (n : ℤ) : (res F L h y n : Γ(twistZ F L n, W)) = resM (twistZ F L n) h (y n)

theorem **smul_bijective** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (hV : V ≤ Out.nonzeroLocus σ) : Function.Bijective fun y : SecModule F L V => SecRing.of m σ • y

theorem **isLocalizedModule_res** [F.IsQuasicoherent] (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (hW : IsCompact (W : Set X)) (hW' : IsQuasiSeparated (W : Set X)) (h : V ≤ W) (hV : V = W ⊓ Out.nonzeroLocus σ) : IsLocalizedModule.Away (SecRing.of m σ) (res F L h) where

theorem **pow_smul_awayEquiv** (k : ℕ) (y : SecModule F L W) : SecRing.of m σ ^ k • awayEquiv hL σ hW hW' h hV (awayFrac _ _ k y) = res F L h y

theorem **awayEquiv_awayFrac_eq_iff** (k : ℕ) {d : ℤ} (hd : n + k * m = d) (a : Γ(twistZ F L d, W)) (b : Γ(twistZ F L n, V)) : awayEquiv hL σ hW hW' h hV (awayFrac _ _ k (of d a)) = of n b ↔ resM _ h a = (mulSecOfEq F (zpowSec σ k) hd).app V b

theorem **mem_awayPiece_iff** (z : LocalizedModule.Away (SecRing.of m σ) (SecModule F L W)) : z ∈ awayPiece F σ W n ↔ ∃ b : Γ(twistZ F L n, V), awayEquiv hL σ hW hW' h hV z = of n b

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.BijOnKerAndExactOfQuasiIsoBaseChange

theorem **bijOn_ker_and_exact_of_quasiIso** {R : Type} [CommRing R] {K C : CochainComplex (ModuleCat.{0} R) ℕ} (φ : K ⟶ C) (h : QuasiIso φ) : Set.BijOn (φ.f 0).hom (LinearMap.ker (K.d 0 1).hom) (LinearMap.ker (C.d 0 1).hom) ∧ (Function.Exact (C.d 0 1).hom (C.d 1 2).hom → Function.Exact (K.d 0 1).hom (K.d 1 2).hom)

theorem **bijOn_ker_and_exact_of_quasiIso_baseChange** (R R' : Type) [CommRing R] [CommRing R'] [inst : Algebra R R'] (K C : CochainComplex (ModuleCat.{0} R) ℕ) (φ : K ⟶ C) (h : QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)) : Set.BijOn ((φ.f 0).hom.baseChange R') (LinearMap.ker ((K.d 0 1).hom.baseChange R')) (LinearMap.ker ((C.d 0 1).hom.baseChange R')) ∧ (Function.Exact ((C.d 0 1).hom.baseChange R') ((C.d 1 2).hom.baseChange R') → Function.Exact ((K.d 0 1).hom.baseChange R') ((K.d 1 2).hom.baseChange R'))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.BijectiveToPushforwardComp

theorem **modRes_modRes** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (s : Γ(F, U)) : F.presheaf.map (homOfLE h₂).op (F.presheaf.map (homOfLE h₁).op s) = F.presheaf.map (homOfLE (h₂.trans h₁)).op s

theorem **modRes_self** {U : X.Opens} (h : U ≤ U) (s : Γ(F, U)) : F.presheaf.map (homOfLE h).op s = s

theorem **modRes_smul** {U V : X.Opens} (h : V ≤ U) (a : Γ(X, U)) (s : Γ(F, U)) : F.presheaf.map (homOfLE h).op (a • s) = resG X h a • F.presheaf.map (homOfLE h).op s

theorem **app_modRes** (φ : F ⟶ G) {U V : X.Opens} (h : V ≤ U) (s : Γ(F, U)) : φ.app V (F.presheaf.map (homOfLE h).op s) = G.presheaf.map (homOfLE h).op (φ.app U s)

theorem **modSec_ext_of_cover** {ι : Type} {V : X.Opens} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) {a b : Γ(F, V)} (h : ∀ k, F.presheaf.map (homOfLE (hW k)).op a = F.presheaf.map (homOfLE (hW k)).op b) : a = b

theorem **exists_modSec_of_cover** {ι : Type} {V : X.Opens} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) (a : ∀ k, Γ(F, W k)) (h : ∀ k l, F.presheaf.map (homOfLE (inf_le_left : W k ⊓ W l ≤ W k)).op (a k) = F.presheaf.map (homOfLE (inf_le_right : W k ⊓ W l ≤ W l)).op (a l)) : ∃ b : Γ(F, V), ∀ k, F.presheaf.map (homOfLE (hW k)).op b = a k

theorem **IsFrameFamily.rel** {F : X.Modules} {σ : ∀ i, Γ(F, L.U i)} (hσ : IsFrameFamily L σ) (i j : L.ι) {W : X.Opens} (hi : W ≤ L.U i) (hj : W ≤ L.U j) (h : W ≤ L.U j ⊓ L.U i) : F.presheaf.map (homOfLE hi).op (σ i) = resG X h (L.g j i : Γ(X, L.U j ⊓ L.U i)) • F.presheaf.map (homOfLE hj).op (σ j)

theorem **isFrameFamily_chartSec** (hL : L.IsLB) : IsFrameFamily L (chartSec hL)

theorem **IsFrameFamily.map** {F G : X.Modules} {σ : ∀ i, Γ(F, L.U i)} (hσ : IsFrameFamily L σ) (φ : F ⟶ G) : IsFrameFamily L fun i => φ.app (L.U i) (σ i)

theorem **app_eq_smul** (hL : L.IsLB) {F : X.Modules} (φ : toMod L ⟶ F) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (s : Γ(toMod L, V)) : φ.app V s = chartEquiv hL i h s • F.presheaf.map (homOfLE h).op (φ.app (L.U i) (chartSec hL i))

theorem **hom_ext_chartSec** (hL : L.IsLB) {F : X.Modules} {φ ψ : toMod L ⟶ F} (h : ∀ i, φ.app (L.U i) (chartSec hL i) = ψ.app (L.U i) (chartSec hL i)) : φ = ψ

theorem **exists_frame_glue** (hL : L.IsLB) {F : X.Modules} {σ : ∀ i, Γ(F, L.U i)} (hσ : IsFrameFamily L σ) {V : X.Opens} (s : Γ(toMod L, V)) : ∃ t : Γ(F, V), ∀ i, F.presheaf.map (homOfLE (inf_le_left : V ⊓ L.U i ≤ V)).op t = coord s i • F.presheaf.map (homOfLE (inf_le_right : V ⊓ L.U i ≤ L.U i)).op (σ i)

theorem **modRes_frameFun** (hL : L.IsLB) {F : X.Modules} {σ : ∀ i, Γ(F, L.U i)} (hσ : IsFrameFamily L σ) {V : X.Opens} (s : Γ(toMod L, V)) (i : L.ι) {W : X.Opens} (hV : W ≤ V) (hi : W ≤ L.U i) : F.presheaf.map (homOfLE hV).op (frameFun hL hσ s) = resG X (le_inf hV hi) (coord s i) • F.presheaf.map (homOfLE hi).op (σ i)

theorem **homOfFrames_app_chartSec** (hL : L.IsLB) {F : X.Modules} {σ : ∀ i, Γ(F, L.U i)} (hσ : IsFrameFamily L σ) (i : L.ι) : (homOfFrames hL hσ).app (L.U i) (chartSec hL i) = σ i

theorem **bijective_toPushforward_comp** (hL : L.IsLB) (f : Y ⟶ X) (G : Y.Modules) : Function.Bijective fun ψ : toMod (L.pull f) ⟶ G => toPushforward L f ≫ (Scheme.Modules.pushforward f).map ψ

theorem **isIso_pullbackHom** {X Y : Scheme.{0}} {L : LB X} (hL : L.IsLB) (f : Y ⟶ X) : IsIso (GlobalTheory.LB.pullbackHom L f)

@[simp] theorem **pullbackIso_hom** (hL : L.IsLB) (f : Y ⟶ X) : (pullbackIso hL f).hom = pullbackHom L f

theorem **unit_comp_pullbackHom** (L : LB X) (f : Y ⟶ X) : (Scheme.Modules.pullbackPushforwardAdjunction f).unit.app (toMod L) ≫ (Scheme.Modules.pushforward f).map (pullbackHom L f) = toPushforward L f

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechComplexAcyclicAdditiveLineBundleNoSections

theorem **isIso_shear** {C : Type*} [Category C] [CartesianMonoidalCategory C] (A : C) [GrpObj A] : IsIso (lift μ[A] (snd A A))

theorem **isPullback_mul_snd** {S : Scheme.{0}} (A : Over S) [GrpObj A] : IsPullback μ[A].left (snd A A).left A.hom A.hom

theorem **exists_fin_affine_cover** (Z : Scheme.{0}) [CompactSpace Z] : ∃ (n : ℕ) (E : Fin n → Z.Opens), (∀ i, IsAffineOpen (E i)) ∧ ⨆ i, E i = ⊤

theorem **prod_res_pullU_one** {X X' : Scheme.{0}} {J : Type} [Fintype J] (φ : J → (X' ⟶ X)) (ε : J → ℤ) (W : X'.Opens) (V : J → X.Opens) (h : ∀ k, W ≤ φ k ⁻¹ᵁ V k) : ∏ k, res (h k) (pullU (φ k) (V k) 1) ^ ε k = 1

theorem **linComb_g_eq_one** {X X' : Scheme.{0}} (L : LB X) (hL : ∀ i j, L.g i j = 1) {J : Type} [Fintype J] (φ : J → (X' ⟶ X)) (ε : J → ℤ) (i j : J → L.ι) : (L.LinComb φ ε).g i j = 1

theorem **tw_pull_triv** {X X' : Scheme.{0}} (N : LB X') (π : X' ⟶ X) : tw N ((Inh.trivLB X).pull π) 0 = tw N ((Inh.trivLB X).pull π) 1

theorem **nontrivial_secOnR_triv** {R : Type} [CommRing R] {X : Scheme.{0}} [Nonempty X] (f : X ⟶ Spec (CommRingCat.of R)) : Nontrivial (SecOnR f (Inh.trivLB X) ⊤)

theorem **cechLB_acyclic_of_additive** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → (∃ w : LB.Iso (M.pull μ[P.A].left) (tw (M.pull (fst P.A P.A).left) (M.pull (snd P.A P.A).left) 1), w.IsGen) → (∀ (σ : LB.Sec M) i, σ.f i = 0) → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → P.A.left.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (cechLB P.A.hom M D).Acyclic

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechComplexAcyclicAdditiveLineBundleNoSectionsStatement

theorem **_root_.GlobalTheory.T.cechComplex_acyclic_additiveLineBundle_noSections** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (M : LB P.A.left), M.IsLB → (∃ w : LB.Iso (M.pull μ[P.A].left) (tw (M.pull (fst P.A P.A).left) (M.pull (snd P.A P.A).left) 1), w.IsGen) → (∀ (σ : LB.Sec M) i, σ.f i = 0) → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → P.A.left.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (cechLB P.A.hom M D).Acyclic

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechLBHomologyEquivOfAffineCovers

theorem **acyclic_slice_secCube_of_isSeparated** [Fintype κ] (hM : M.IsLB) [IsSeparated f] (E : ν → Y.Opens) (hE : ∀ i, IsAffineOpen (E i)) (e : κ ↪o ν) (hc : ⊔ k, E (e k) = ⊤) {s₀ : Finset ν} (h₀ : s₀.Nonempty) : (cubeComplex (Cube.slice e s₀ (secCube f M E))).Acyclic

theorem **cechLB_homology_equiv_of_affine_covers** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (κ' : Type) [LinearOrder κ'] [Fintype κ'] (D' : κ' → Y.Opens), M.IsLB → IsSeparated f → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (∀ i, IsAffineOpen (D' i)) → ⊔ i, D' i = ⊤ → ∀ n, Nonempty ((cechLB f M D).homology n ≃ₗ[R] (cechLB f M D').homology n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechLBHomologyEquivOfAffineCoversStatement

theorem **_root_.GlobalTheory.T.nonempty_cechLB_homology_equiv_of_affine_covers** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (κ' : Type) [LinearOrder κ'] [Fintype κ'] (D' : κ' → Y.Opens), M.IsLB → IsSeparated f → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (∀ i, IsAffineOpen (D' i)) → ⊔ i, D' i = ⊤ → ∀ n, Nonempty ((cechLB f M D).homology n ≃ₗ[R] (cechLB f M D').homology n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechLBHomologyFiniteOfAffineCover

theorem **cechLB_homology_finite_of_affine_cover** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), M.IsLB → IsSeparated f → IsProjBy f → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → ∀ n, Module.Finite R ((cechLB f M D).homology n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechLBHomologyFiniteOfAffineCoverStatement

theorem **_root_.GlobalTheory.T.finite_cechLB_homology_of_affine_cover** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), M.IsLB → IsSeparated f → IsProjBy f → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → ∀ n, Module.Finite R ((cechLB f M D).homology n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechLBHomologyZeroEquivSections

theorem **cechLB_homology_zero_equiv_sections** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), ⊔ i, D i = ⊤ → Nonempty ((cechLB f M D).homology 0 ≃ₗ[R] SecOnR f M ⊤)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechLBPullIsoOfIsOpenImmersion

theorem **bijective_appLE_of_le_opensRange** {Y Y' : Scheme.{0}} (e : Y' ⟶ Y) [IsOpenImmersion e] {V : Y.Opens} (hV : V ≤ e.opensRange) {V' : Y'.Opens} (h : V' ≤ e ⁻¹ᵁ V) (h' : e ⁻¹ᵁ V ≤ V') : Function.Bijective (e.appLE V V' h).hom

theorem **SecOn.pullLE_bijective_of_le_opensRange** {Y Y' : Scheme.{0}} (M : LB Y) (e : Y' ⟶ Y) [IsOpenImmersion e] {V : Y.Opens} (hV : V ≤ e.opensRange) {V' : Y'.Opens} (h : V' ≤ e ⁻¹ᵁ V) (h' : e ⁻¹ᵁ V ≤ V') : Function.Bijective (SecOn.pullLE M e V V' h)

theorem **cechLB_pull_iso_of_isOpenImmersion** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (Y' : Scheme.{0}) (e : Y' ⟶ Y) [IsOpenImmersion e] (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), M.IsLB → (∀ i, D i ≤ e.opensRange) → Nonempty (cechLB (e ≫ f) (M.pull e) (fun i => e ⁻¹ᵁ D i) ≅ cechLB f M D)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechModulesFunctorShortExactOfAcyclic

theorem **exists_addEquiv_of_surjective_of_ker** {A B C : Type*} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] (f : A →+ B) (g : A →+ C) (hf : Function.Surjective f) (hg : Function.Surjective g) (h : ∀ a, f a = 0 ↔ g a = 0) : ∃ e : B ≃+ C, ∀ a, e (f a) = g a

theorem **δ_cechClass** {S : ShortComplex (Cube.{u} ι R)} (hS : (S.map (cechFunctor ι R)).ShortExact) (p : ℕ) (x₃ : LinearMap.ker (Cube.d S.X₃ (p + 1))) (x₂ : Cube.Term S.X₂ (p + 1)) (x₁ : LinearMap.ker (Cube.d S.X₁ (p + 1 + 1))) (h₂ : Cube.termMap S.g (p + 1) x₂ = x₃.1) (h₁ : Cube.termMap S.f (p + 1 + 1) x₁.1 = Cube.d S.X₂ (p + 1) x₂) : (hS.δ p (p + 1) rfl).hom (cechClass S.X₃ p x₃) = cechClass S.X₁ (p + 1) x₁

theorem **exists_lift_cechCocycle** {S : ShortComplex (Cube.{u} ι R)} (hS : (S.map (cechFunctor ι R)).ShortExact) (p : ℕ) (x₃ : LinearMap.ker (Cube.d S.X₃ (p + 1))) : ∃ (x₂ : Cube.Term S.X₂ (p + 1)) (x₁ : LinearMap.ker (Cube.d S.X₁ (p + 1 + 1))), Cube.termMap S.g (p + 1) x₂ = x₃.1 ∧ Cube.termMap S.f (p + 1 + 1) x₁.1 = Cube.d S.X₂ (p + 1) x₂

theorem **Cube.SemiHom.homology_δ** {R' : Type*} [Ring R'] {σ : R →+* R'} {S : ShortComplex (Cube.{u} ι R)} {S' : ShortComplex (Cube.{u} ι R')} (hS : (S.map (cechFunctor ι R)).ShortExact) (hS' : (S'.map (cechFunctor ι R')).ShortExact) (ψ₁ : Cube.SemiHom σ S.X₁ S'.X₁) (ψ₂ : Cube.SemiHom σ S.X₂ S'.X₂) (ψ₃ : Cube.SemiHom σ S.X₃ S'.X₃) (hf : ∀ s x, ψ₂.app s ((S.f.app s).hom x) = (S'.f.app s).hom (ψ₁.app s x)) (hg : ∀ s x, ψ₃.app s ((S.g.app s).hom x) = (S'.g.app s).hom (ψ₂.app s x)) (p : ℕ) (x : (cechComplex S.X₃).homology p) : ψ₁.homology (p + 1) ((hS.δ p (p + 1) rfl).hom x) = (hS'.δ p (p + 1) rfl).hom (ψ₃.homology p x)

theorem **cechAcyclic_of_injective** (I : Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u}) [Injective I] : CechAcyclic V U I

theorem **CechAcyclic.of_shortExact₃** {S : ShortComplex (Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u})} (hS : S.ShortExact) (h₁ : CechAcyclic V U S.X₁) (h₂ : CechAcyclic V U S.X₂) : CechAcyclic V U S.X₃

theorem **CechAcyclic.of_shortExact₂** {S : ShortComplex (Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u})} (hS : S.ShortExact) (h₁ : CechAcyclic V U S.X₁) (h₃ : CechAcyclic V U S.X₃) : CechAcyclic V U S.X₂

theorem **sections_shortExact_of_subsingleton** {S : ShortComplex (Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u})} (hS : S.ShortExact) (W : Opens T) (h : Subsingleton (Sheaf.H'.{u} S.X₁ 1 W)) : Function.Injective (S.f.hom.app (op W)) ∧ Function.Exact (S.f.hom.app (op W)) (S.g.hom.app (op W)) ∧ Function.Surjective (S.g.hom.app (op W))

theorem **cechAbSheafFunctor_shortExact** : (S.map (cechAbSheafFunctor V U)).ShortExact

theorem **cechAbδ_exact₃** (n : ℕ) : Function.Exact (cechAbHMap V U S.g.hom n).hom (cechAbδ hS h₁ n)

theorem **cechAbδ_exact₁** (n : ℕ) : Function.Exact (cechAbδ hS h₁ n) (cechAbHMap V U S.f.hom (n + 1)).hom

theorem **cechAbδ_naturality** (hS' : S'.ShortExact) (h₁' : CechAcyclic V U S'.X₁) (φ : S ⟶ S') (n : ℕ) (x : cechAbH S.X₃.obj V U n) : (cechAbHMap V U φ.τ₁.hom (n + 1)).hom (cechAbδ hS h₁ n x) = cechAbδ hS' h₁' n ((cechAbHMap V U φ.τ₃.hom n).hom x)

theorem **cechAbδ_surjective** [Finite ι] [Injective S.X₂] (n : ℕ) : Function.Surjective (cechAbδ hS h₁ n)

theorem **CechComparison.exists_succ** [Finite ι] {n : ℕ} (e : CechComparison V U n) : ∃ e' : CechComparison V U (n + 1), ∀ (S : ShortComplex (Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u})) (hS : S.ShortExact) (h₁ : CechAcyclic V U S.X₁) (h₃ : CechAcyclic V U S.X₃) x, e'.1 S.X₁ h₁ (cechAbδ hS h₁ n x) = SheafLES.δ hS n V (e.1 S.X₃ h₃ x)

theorem **cechAbHEquivH'_toCechAbH** (hF : CechAcyclic V U F) (x : F.obj.obj (op V)) : cechAbHEquivH' hc hF 0 (toCechAbH F.obj V U x) = (SheafLES.equivZero F V).symm x

theorem **cechAbHEquivH'_map** (hF : CechAcyclic V U F) (hG : CechAcyclic V U G) (φ : F ⟶ G) (n : ℕ) (x : cechAbH F.obj V U n) : cechAbHEquivH' hc hG n ((cechAbHMap V U φ.hom n).hom x) = SheafLES.map φ n V (cechAbHEquivH' hc hF n x)

theorem **cechAbHEquivH'_δ** {S : ShortComplex (Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u})} (hS : S.ShortExact) (h₁ : CechAcyclic V U S.X₁) (h₃ : CechAcyclic V U S.X₃) (n : ℕ) (x : cechAbH S.X₃.obj V U n) : cechAbHEquivH' hc h₁ (n + 1) (cechAbδ hS h₁ n x) = SheafLES.δ hS n V (cechAbHEquivH' hc h₃ n x)

theorem **cechAcyclicMod_iff** : CechAcyclicMod V U F ↔ CechAcyclic (T

theorem **cechAcyclicMod_iff_of_le** (h : ∀ i, U i ≤ V) : CechAcyclicMod V U F ↔ ∀ s : Finset ι, s.Nonempty → ∀ q, 0 < q → Subsingleton (cohOn F (⊓ i ∈ s, U i) q)

theorem **CechAcyclicMod.of_shortExact₃** {S : ShortComplex X.Modules} (hS : S.ShortExact) (h₁ : CechAcyclicMod V U S.X₁) (h₂ : CechAcyclicMod V U S.X₂) : CechAcyclicMod V U S.X₃

theorem **CechAcyclicMod.of_shortExact₂** {S : ShortComplex X.Modules} (hS : S.ShortExact) (h₁ : CechAcyclicMod V U S.X₁) (h₃ : CechAcyclicMod V U S.X₃) : CechAcyclicMod V U S.X₂

theorem **cechModulesFunctor_shortExact_of_acyclic** : (S.map (cechModulesFunctor R V U)).ShortExact

theorem **cechδ_def** (h : (S.map (cechModulesFunctor R V U)).ShortExact) (n : ℕ) : cechδ R hS h₁ n = (h.δ n (n + 1) rfl).hom

theorem **cechAbHEquiv_δ** (n : ℕ) (x : cechAbH S.X₃.presheaf V U n) : cechAbHEquiv R S.X₁ V U (n + 1) (cechAbδ (T

theorem **toH'_cechHAddEquivCohOn** (hF : CechAcyclicMod V U F) (n : ℕ) (x : cechAbH F.presheaf V U n) : cohOn.toH' F V n (cechHAddEquivCohOn R hc hF n (cechAbHEquiv R F V U n x)) = cechAbHEquivH' (T

theorem **cechHAddEquivCohOn_map** (hF : CechAcyclicMod V U F) (hG : CechAcyclicMod V U G) (φ : F ⟶ G) (n : ℕ) (x : cechH R F V U n) : cechHAddEquivCohOn R hc hG n ((cechHMap R V U φ n).hom x) = cohOn.map R φ V n (cechHAddEquivCohOn R hc hF n x)

theorem **toH'_cechHEquivCohOn** (hF : CechAcyclicMod V U F) (n : ℕ) (x : cechAbH F.presheaf V U n) : cohOn.toH' F V n (cechHEquivCohOn R hc hF n (cechAbHEquiv R F V U n x)) = cechAbHEquivH' (T

theorem **cechHEquivCohOn_map** (hF : CechAcyclicMod V U F) (hG : CechAcyclicMod V U G) (φ : F ⟶ G) (n : ℕ) (x : cechH R F V U n) : cechHEquivCohOn R hc hG n ((cechHMap R V U φ n).hom x) = cohOn.map R φ V n (cechHEquivCohOn R hc hF n x)

theorem **cechHEquivCohOn_δ** {S : ShortComplex X.Modules} (hS : S.ShortExact) (h₁ : CechAcyclicMod V U S.X₁) (h₃ : CechAcyclicMod V U S.X₃) (n : ℕ) (x : cechH R S.X₃ V U n) : cechHEquivCohOn R hc h₁ (n + 1) (cechδ R hS h₁ n x) = cohOn.δ R hS V n (cechHEquivCohOn R hc h₃ n x)

theorem **cechHEquivCohOn_zero** (hF : CechAcyclicMod V U F) : cechHEquivCohOn R hc hF 0 = cechHZeroEquivCohOn R F hc

theorem **subsingleton_cohOn_of_cechAcyclic** [Fintype ι] (hF : CechAcyclicMod V U F) {n : ℕ} (hn : Fintype.card ι ≤ n) : Subsingleton (cohOn F V n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CechModulesFunctorShortExactOfIsAffineOpen

theorem **cechAcyclicMod_of_isAffineOpen** (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (cechOpen V U s)) (F : X.Modules) [F.IsQuasicoherent] : CechAcyclicMod V U F

theorem **isAffineOpen_cechOpen_of_le** (h : ∀ i, U i ≤ V) (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (⊓ i ∈ s, U i)) (s : Finset ι) (hs : s.Nonempty) : IsAffineOpen (cechOpen V U s)

theorem **isAffineOpen_finset_biInf** [IsAffineHom (pullback.diagonal (terminal.from X))] (hU : ∀ i, IsAffineOpen (U i)) (s : Finset ι) (hs : s.Nonempty) : IsAffineOpen (⊓ i ∈ s, U i)

theorem **isAffineHom_diagonal_of_isSeparated** {S : Scheme.{u}} [IsAffine S] (f : X ⟶ S) [IsSeparated f] : IsAffineHom (pullback.diagonal (terminal.from X))

theorem **exists_fin_affine_cover** (hV : IsCompact (V : Set X)) : ∃ (m : ℕ) (U : Fin m → X.Opens), (∀ i, IsAffineOpen (U i)) ∧ ⊔ i, U i = V

theorem **cechAcyclicMod_top_of_isAffineOpen** (F : X.Modules) [F.IsQuasicoherent] : CechAcyclicMod ⊤ U F

theorem **cechHEquivCoh_def** (F : X.Modules) [F.IsQuasicoherent] (n : ℕ) : cechHEquivCoh R hc hU F n = cechHEquivCohOn R hc.ge (cechAcyclicMod_top_of_isAffineOpen hU F) n

theorem **cechHEquivCoh_map** {F G : X.Modules} [F.IsQuasicoherent] [G.IsQuasicoherent] (φ : F ⟶ G) (n : ℕ) (x : cechH R F ⊤ U n) : cechHEquivCoh R hc hU G n ((cechHMap R ⊤ U φ n).hom x) = cohOn.map R φ ⊤ n (cechHEquivCoh R hc hU F n x)

theorem **cechHEquivCoh_toCechH** (F : X.Modules) [F.IsQuasicoherent] (x : sectionsOn F ⊤) : cechHEquivCoh R hc hU F 0 (toCechH R F ⊤ U x) = (cohOn.zeroEquiv R F ⊤).symm x

theorem **cechModulesFunctor_shortExact_of_isAffineOpen** {S : ShortComplex X.Modules} (hS : S.ShortExact) [S.X₁.IsQuasicoherent] : (S.map (cechModulesFunctor R ⊤ U)).ShortExact

theorem **cechHEquivCoh_δ** {S : ShortComplex X.Modules} (hS : S.ShortExact) [S.X₁.IsQuasicoherent] [S.X₃.IsQuasicoherent] (h : (S.map (cechModulesFunctor R ⊤ U)).ShortExact) (n : ℕ) (x : cechH R S.X₃ ⊤ U n) : cechHEquivCoh R hc hU S.X₁ (n + 1) ((h.δ n (n + 1) rfl).hom x) = cohOn.δ R hS ⊤ n (cechHEquivCoh R hc hU S.X₃ n x)

theorem **cohOn.subsingleton_of_card_le** [Fintype ι] (hc : V ≤ ⊔ i, U i) (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (cechOpen V U s)) (F : X.Modules) [F.IsQuasicoherent] {n : ℕ} (hn : Fintype.card ι ≤ n) : Subsingleton (cohOn F V n)

theorem **coh.subsingleton_of_card_le** [Fintype ι] (hc : ⊔ i, U i = ⊤) (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (⊓ i ∈ s, U i)) (F : X.Modules) [F.IsQuasicoherent] {n : ℕ} (hn : Fintype.card ι ≤ n) : Subsingleton (coh F n)

theorem **sectionsOn.exact_of_isAffineOpen** {S : ShortComplex X.Modules} (hS : S.Exact) [S.X₁.IsQuasicoherent] [S.X₂.IsQuasicoherent] [S.X₃.IsQuasicoherent] {U : X.Opens} (hU : IsAffineOpen U) : Function.Exact (sectionsOn.map R S.f U) (sectionsOn.map R S.g U)

theorem **sectionsOn.shortExact_of_isAffineOpen** {S : ShortComplex X.Modules} (hS : S.ShortExact) [S.X₁.IsQuasicoherent] {U : X.Opens} (hU : IsAffineOpen U) : Function.Injective (sectionsOn.map R S.f U) ∧ Function.Exact (sectionsOn.map R S.f U) (sectionsOn.map R S.g U) ∧ Function.Surjective (sectionsOn.map R S.g U)

theorem **inj_cofanProj** (k : κ) : c.inj k ≫ cofanProj hc k = 𝟙 _

theorem **inj_cofanProj_of_ne** {j k : κ} (h : j ≠ k) : c.inj j ≫ cofanProj hc k = 0

theorem **inj_comp_sum_cofanProj** {T : Finset κ} {j : κ} (hj : j ∈ T) : c.inj j ≫ ∑ k ∈ T, cofanProj hc k ≫ c.inj k = c.inj j

theorem **exists_finset_sum_cofanProj_app** [Small.{u} κ] [∀ k, (F k).IsQuasicoherent] {W : X.Opens} (hW : IsAffineOpen W) (s : Γ(c.pt, W)) : ∃ T : Finset κ, ∀ T', T ⊆ T' → (∑ k ∈ T', cofanProj hc k ≫ c.inj k).app W s = s

theorem **cohOn.directSumMap_lof** (k : κ) (x : cohOn (F k) V n) : cohOn.directSumMap c R V n (DirectSum.lof R κ (fun k => cohOn (F k) V n) k x) = cohOn.map R (c.inj k) V n x

theorem **cohOn.directSumMap_injective** : Function.Injective (cohOn.directSumMap c R V n)

theorem **cohOn.directSumMap_bijective** {ι : Type} [Finite ι] {U : ι → X.Opens} (hV : V ≤ ⨆ i, U i) (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (cechOpen V U s)) : Function.Bijective (cohOn.directSumMap c R V n)

theorem **coh.sigmaEquiv_lof** {ι : Type} [Finite ι] {U : ι → X.Opens} (hcov : ⨆ i, U i = ⊤) (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (⨅ i ∈ s, U i)) (k : κ) (x : coh (F k) n) : coh.sigmaEquiv F R n hcov hU (DirectSum.lof R κ (fun k => coh (F k) n) k x) = cohOn.map R (Sigma.ι F k) ⊤ n x

theorem **SheafLES.pi_res_bijective_of_disjoint** {T : TopCat.{u}} {κ : Type w} {V : Opens T} {U : κ → Opens T} (h : ∀ i, U i ≤ V) (hc : V ≤ ⨆ i, U i) (hd : Pairwise fun i j => Disjoint (U i) (U j)) (F : Sheaf (Opens.grothendieckTopology T) AddCommGrpCat.{u}) (n : ℕ) : Function.Bijective fun (x : Sheaf.H'.{u} F n V) (i : κ) => res F n (homOfLE (h i)) x

theorem **cohOn.pi_res_bijective_of_disjoint** (h : ∀ i, U i ≤ V) (hc : V ≤ ⨆ i, U i) (hd : Pairwise fun i j => Disjoint (U i) (U j)) (F : X.Modules) (n : ℕ) : Function.Bijective (LinearMap.pi fun i => cohOn.res R F (h i) n)

theorem **coh.disjointEquiv_apply** (hc : ⨆ i, U i = ⊤) (hd : Pairwise fun i j => Disjoint (U i) (U j)) (F : X.Modules) (n : ℕ) (x : coh F n) (i : κ) : coh.disjointEquiv R hc hd F n x i = cohOn.res R F le_top n x

theorem **coh.disjointRestrictEquiv_apply** (F : X.Modules) (n : ℕ) (x : coh F n) (i : κ) : coh.disjointRestrictEquiv R j hj hc hd F n x i = cohOn.pull (σ

theorem **cech_computes_coh_of_affine_cover** (X : Scheme.{u}) (R : Type v) [CommRing R] [Algebra R Γ(X, ⊤)] : ((X.IsSeparated ∨ IsAffineHom (pullback.diagonal (terminal.from X))) → ∀ (ι : Type) (U : ι → X.Opens), (∀ i, IsAffineOpen (U i)) → ∀ s : Finset ι, s.Nonempty → IsAffineOpen (⨅ i ∈ s, U i)) ∧ (∀ (ι : Type) [LinearOrder ι] [Fintype ι] (U : ι → X.Opens), ⨆ i, U i = ⊤ → (∀ s : Finset ι, s.Nonempty → IsAffineOpen (⨅ i ∈ s, U i)) → (∃ e : ∀ (F : X.Modules) [F.IsQuasicoherent] (n : ℕ), cechH R F ⊤ U n ≃ₗ[R] coh F n, (∀ (F : X.Modules) [F.IsQuasicoherent] (x : sectionsOn F ⊤), e F 0 (toCechH R F ⊤ U x) = (cohOn.zeroEquiv R F ⊤).symm x) ∧ (∀ (F G : X.Modules) [F.IsQuasicoherent] [G.IsQuasicoherent] (φ : F ⟶ G) (n : ℕ) (x : cechH R F ⊤ U n), e G n ((cechHMap R ⊤ U φ n).hom x) = cohOn.map R φ ⊤ n (e F n x)) ∧ ∀ (S : ShortComplex X.Modules) (hS : S.ShortExact) [S.X₁.IsQuasicoherent] [S.X₃.IsQua

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ClosedBallValuationSubringToSubring

theorem **mem_closedBallSubring** {x : K} : x ∈ closedBallSubring K ↔ ‖x‖ ≤ 1

theorem **mem_openBallIdeal** {x : closedBallSubring K} : x ∈ openBallIdeal K ↔ ‖(x : K)‖ < 1

theorem **isUnit_iff_norm_eq_one** {x : closedBallSubring K} : IsUnit x ↔ ‖(x : K)‖ = 1

theorem **mem_openBallIdeal_iff_not_isUnit** {x : closedBallSubring K} : x ∈ openBallIdeal K ↔ ¬ IsUnit x

theorem **openBallIdeal_eq_maximalIdeal** : openBallIdeal K = IsLocalRing.maximalIdeal (closedBallSubring K)

theorem **openBallIdeal_isMaximal** : (openBallIdeal K).IsMaximal

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CoeConjEquivSymm

@[simp] theorem **coe_conjEquiv** (N : Subgroup G) (b : G) (hb : ∀ u : G, u ∈ N ↔ b * u * b⁻¹ ∈ N) (u : N) : ((conjEquiv N b hb u : N) : G) = b * (u : G) * b⁻¹

@[simp] theorem **coe_conjEquiv_symm** (N : Subgroup G) (b : G) (hb : ∀ u : G, u ∈ N ↔ b * u * b⁻¹ ∈ N) (u : N) : (((conjEquiv N b hb).symm u : N) : G) = b⁻¹ * (u : G) * b

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CohomologyPresheafFunctorMapAppApply

lemma **freeSheafHomEquiv_apply** (U : C) (F : Sheaf J AddCommGrpCat.{v}) (f : freeSheaf J U ⟶ F) : freeSheafHomEquiv U F f = f.hom.app (op U) (freeSheafGen J U)

lemma **freeSheafHomEquiv_comp** {U : C} {F G : Sheaf J AddCommGrpCat.{v}} (f : freeSheaf J U ⟶ F) (g : F ⟶ G) : freeSheafHomEquiv U G (f ≫ g) = g.hom.app (op U) (freeSheafHomEquiv U F f)

lemma **freeSheafMap_gen** {V U : C} (i : V ⟶ U) : (freeSheafMap J i).hom.app (op V) (freeSheafGen J V) = (freeSheaf J U).obj.map i.op (freeSheafGen J U)

lemma **freeSheafHomEquiv_freeSheafMap_comp** {V U : C} (i : V ⟶ U) {F : Sheaf J AddCommGrpCat.{v}} (f : freeSheaf J U ⟶ F) : freeSheafHomEquiv V F (freeSheafMap J i ≫ f) = F.obj.map i.op (freeSheafHomEquiv U F f)

lemma **sectionsEquiv_mk₀** {F : Sheaf J AddCommGrpCat.{v}} {U : C} (f : freeSheaf J U ⟶ F) : sectionsEquiv F U (Ext.mk₀ f) = freeSheafHomEquiv U F f

lemma **sectionsEquiv_map** {F G : Sheaf J AddCommGrpCat.{v}} (f : F ⟶ G) (U : C) (x : F.H' 0 U) : sectionsEquiv G U (((Sheaf.cohomologyPresheafFunctor J 0).map f).app (op U) x) = f.hom.app (op U) (sectionsEquiv F U x)

lemma **sectionsEquiv_restrict** (F : Sheaf J AddCommGrpCat.{v}) {V U : C} (i : V ⟶ U) (x : F.H' 0 U) : sectionsEquiv F V ((F.cohomologyPresheaf 0).map i.op x) = F.obj.map i.op (sectionsEquiv F U x)

lemma **_root_.GlobalTheory.Flasque.subsingleton_H'_of_injective** (I : Sheaf J AddCommGrpCat.{v}) [Injective I] (n : ℕ) (U : C) : Subsingleton (I.H' (n + 1) U)

lemma **exact_map_connecting** (n : ℕ) (U : C) : Function.Exact (((Sheaf.cohomologyPresheafFunctor J n).map S.g).app (op U)) (connecting hS n U)

lemma **exact_connecting_map** (n : ℕ) (U : C) : Function.Exact (connecting hS n U) (((Sheaf.cohomologyPresheafFunctor J (n + 1)).map S.f).app (op U))

lemma **connecting_naturality** {T : ShortComplex (Sheaf J AddCommGrpCat.{v})} (hT : T.ShortExact) (φ : S ⟶ T) (n : ℕ) (U : C) (x : S.X₃.H' n U) : connecting hT n U (((Sheaf.cohomologyPresheafFunctor J n).map φ.τ₃).app (op U) x) = ((Sheaf.cohomologyPresheafFunctor J (n + 1)).map φ.τ₁).app (op U) (connecting hS n U x)

lemma **connecting_restrict** (n : ℕ) {V U : C} (i : V ⟶ U) (x : S.X₃.H' n U) : connecting hS n V ((S.X₃.cohomologyPresheaf n).map i.op x) = (S.X₁.cohomologyPresheaf (n + 1)).map i.op (connecting hS n U x)

lemma **connecting_injective** (n : ℕ) (U : C) (h : Subsingleton (S.X₂.H' n U)) : Function.Injective (connecting hS n U)

lemma **connecting_surjective** (n : ℕ) (U : C) (h : Subsingleton (S.X₂.H' (n + 1) U)) : Function.Surjective (connecting hS n U)

lemma **shiftEquiv_apply** (n : ℕ) (U : C) (h : Subsingleton (S.X₂.H' n U)) (h' : Subsingleton (S.X₂.H' (n + 1) U)) (x : S.X₃.H' n U) : shiftEquiv hS n U h h' x = connecting hS n U x

lemma **connecting₀_apply** (U : C) (s : S.X₃.obj.obj (op U)) : connecting₀ hS U s = connecting hS 0 U ((sectionsEquiv S.X₃ U).symm s)

lemma **exact_app_connecting₀** (U : C) : Function.Exact (S.g.hom.app (op U)) (connecting₀ hS U)

lemma **exact_connecting₀_map** (U : C) : Function.Exact (connecting₀ hS U) (((Sheaf.cohomologyPresheafFunctor J 1).map S.f).app (op U))

lemma **connecting₀_surjective** (U : C) (h : Subsingleton (S.X₂.H' 1 U)) : Function.Surjective (connecting₀ hS U)

lemma **ker_connecting₀** (U : C) : (connecting₀ hS U).ker = (S.g.hom.app (op U)).hom.range

lemma **connecting₀_naturality** {T : ShortComplex (Sheaf J AddCommGrpCat.{v})} (hT : T.ShortExact) (φ : S ⟶ T) (U : C) (s : S.X₃.obj.obj (op U)) : connecting₀ hT U (φ.τ₃.hom.app (op U) s) = ((Sheaf.cohomologyPresheafFunctor J 1).map φ.τ₁).app (op U) (connecting₀ hS U s)

lemma **connecting₀_restrict** {V U : C} (i : V ⟶ U) (s : S.X₃.obj.obj (op U)) : connecting₀ hS V (S.X₃.obj.map i.op s) = (S.X₁.cohomologyPresheaf 1).map i.op (connecting₀ hS U s)

lemma **cokerEquiv_mk** (U : C) (h : Subsingleton (S.X₂.H' 1 U)) (s : S.X₃.obj.obj (op U)) : cokerEquiv hS U h (QuotientAddGroup.mk s) = connecting₀ hS U s

lemma **_root_.GlobalTheory.Flasque.injectiveShortComplex_shortExact** [EnoughInjectives A] (F : A) : (_root_.GlobalTheory.Flasque.injectiveShortComplex F).ShortExact where

lemma **_root_.GlobalTheory.Flasque.exists_hom_τ₁_eq** {S T : ShortComplex A} (hS : S.ShortExact) [Injective T.X₂] (f : S.X₁ ⟶ T.X₁) : ∃ φ : S ⟶ T, φ.τ₁ = f

private lemma **freeSheafHomEquiv_eq_yonedaEquiv** (U : C) (F : Sheaf J AddCommGrpCat.{v}) (h : _root_.GlobalTheory.Flasque.freeSheaf J U ⟶ F) : _root_.GlobalTheory.Flasque.freeSheafHomEquiv U F h = yonedaEquiv (((AddCommGrpCat.adj.whiskerRight Cᵒᵖ).homEquiv _ _) ((sheafificationAdjunction J _).homEquiv _ _ h))

private lemma **freeSheafHomEquiv_freeSheafMap_comp'** {V U : C} (i : V ⟶ U) (F : Sheaf J AddCommGrpCat.{v}) (h : _root_.GlobalTheory.Flasque.freeSheaf J U ⟶ F) : _root_.GlobalTheory.Flasque.freeSheafHomEquiv V F (_root_.GlobalTheory.Flasque.freeSheafMap J i ≫ h) = F.obj.map i.op (_root_.GlobalTheory.Flasque.freeSheafHomEquiv U F h)

theorem **_root_.GlobalTheory.Flasque.isFlasque_of_injective** (I : TopCat.Sheaf AddCommGrpCat.{u} X) [hI : Injective I] : I.IsFlasque

lemma **isFlasque_injectiveShortComplex_X₃** (F : TopCat.Sheaf AddCommGrpCat.{u} X) [F.IsFlasque] : (injectiveShortComplex F).X₃.IsFlasque

private lemma **freeSheafHomEquiv_comp'** (U : C) {F G : Sheaf J AddCommGrpCat.{v}} (h : _root_.GlobalTheory.Flasque.freeSheaf J U ⟶ F) (f : F ⟶ G) : _root_.GlobalTheory.Flasque.freeSheafHomEquiv U G (h ≫ f) = f.hom.app (op U) (_root_.GlobalTheory.Flasque.freeSheafHomEquiv U F h)

theorem **_root_.GlobalTheory.Flasque.subsingleton_H'_of_isFlasque** (F : TopCat.Sheaf AddCommGrpCat.{u} X) [hF : F.IsFlasque] (n : ℕ) (U : Opens X) : Subsingleton (F.H' (n + 1) U)

theorem **_root_.GlobalTheory.Flasque.subsingleton_H'_of_isFlasque_of_pos** (F : TopCat.Sheaf AddCommGrpCat.{u} X) [F.IsFlasque] {n : ℕ} (hn : 0 < n) (U : Opens X) : Subsingleton (F.H' n U)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CokerDualMapEquivResidueFieldOfOrigin

theorem **vec_apply** {P : Type} [AddCommGroup P] [Module R P] {a : ℕ} (b : Module.Basis (Fin a) R P) (v : Fin a → R') : vec b v = ∑ k, v k ⊗ₜ[R] b k

theorem **vec_mulVec** {P Q : Type} [AddCommGroup P] [Module R P] [AddCommGroup Q] [Module R Q] {a c : ℕ} (b : Module.Basis (Fin a) R P) (b' : Module.Basis (Fin c) R Q) (d : P →ₗ[R] Q) (v : Fin a → R') : vec b' (((LinearMap.toMatrix b b' d).map (algebraMap R R')).mulVec v) = d.baseChange R' (vec b v)

theorem **coker_dualMap_equiv** {P Q : Type} [AddCommGroup P] [Module R P] [AddCommGroup Q] [Module R Q] {a c : ℕ} (b : Module.Basis (Fin a) R P) (b' : Module.Basis (Fin c) R Q) (d : P →ₗ[R] Q) : Nonempty ((Module.Dual R P ⧸ LinearMap.range d.dualMap) ≃ₗ[R] ((Fin a → R) ⧸ LinearMap.range (LinearMap.toMatrix b b' d).transpose.mulVecLin))

theorem **cech_d** (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (D : κ → Y.Opens) (p : ℕ) : ((cechLB f M D).d p (p + 1)).hom = Cube.d (secCube f M D) (p + 1)

theorem **exists_equiv_sec** (hM : M.IsLB) (hc : ⊔ i, D i = ⊤) (hD : ∀ s : Finset κ, s.Nonempty → IsAffineOpen (⊓ i ∈ s, D i)) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hφ : QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)) : ∃ E : LinearMap.ker ((C.d 0 1).hom.baseChange R') ≃ LB.Sec (M.pull π), ∀ x s, SecOnR.of f' _ _ (GlobalTheory.LB.ofSec (E x) (⊓ i ∈ s.1, π ⁻¹ᵁ D i)) = secTermBaseChange M D H.w 1 ((φ.f 0).hom.baseChange R' x.1) s

theorem **exact_of_h1Zero** (hM : M.IsLB) (hc : ⊔ i, D i = ⊤) (hD : ∀ s : Finset κ, s.Nonempty → IsAffineOpen (⊓ i ∈ s, D i)) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hφ : QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)) (h : FirstCohomologyVanishes (M.pull π)) : Function.Exact ((C.d 0 1).hom.baseChange R') ((C.d 1 2).hom.baseChange R')

theorem **coord_secTermBaseChange** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) {a : ℕ} (v : Fin a → R') (x : Fin a → C.X 0) (p : κ × M.ι) (h : π ⁻¹ᵁ ((⊓ l ∈ ({p.1} : Finset κ), D l) ⊓ M.U p.2) ≤ (⊓ l ∈ ({p.1} : Finset κ), π ⁻¹ᵁ D l) ⊓ (M.pull π).U fun _ => p.2) : rs Y' h (SecOnR.coord (f

theorem **triv_of_origin** (hL : L.IsLB) {X : Scheme.{0}} (q : X ⟶ (A ⊗ A).left) (s : X ⟶ S) (h : q ≫ (fst A A).left = s ≫ η[A].left) : Triv ((translDiff A L (𝟙 A)).pull q)

theorem **isPullback_whiskerRight** {T : Over S} (x : T ⟶ A) : IsPullback (x ▷ A).left (fst T A).left (fst A A).left x.left

theorem **eq_origin_of_triv** (hK : KTriv A L) {R : Type} [CommRing R] {t : Spec (CommRingCat.of R) ⟶ S} (x : Over.mk t ⟶ A) (h : Triv ((translDiff A L (𝟙 A)).pull (x ▷ A).left)) : x.left = t ≫ η[A].left

theorem **coker_dualMap_equiv_residueField_of_origin** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t₀ : Spec (CommRingCat.of R) ⟶ P.A.left) (χ₀ : R →+* K) (O : Type) [CommRing O] [IsLocalRing O] [Algebra R O] [(RingHom.ker χ₀).IsPrime] [IsLocalization.AtPrime O (RingHom.ker χ₀)] (t : Spec (CommRingCat.of O) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of O)), IsPullback π f (fst P.A P.A).left t → IsOpenImmersion t₀ → Spec.map (CommRingCat.ofHom χ₀) ≫ t₀ = η[P.A].left → t = Spec.map (CommRingCat.ofHom (algebraMap R O)) ≫ t₀ → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (C : CochainComplex (ModuleCat.{0} O) ℕ) (φ : C ⟶ cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (∀ p, Module.Free O (C.X p) ∧ Module.Finite O (C.X p)) → (∀ (R' : Type)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ContractMultilinearApply

private theorem **apply_update** (φ : ∀ i, X i →ₗ[R] R) [inst : DecidableEq ι] (x : ∀ i, X i) (k : ι) (y : X k) (i : ι) : φ i (Function.update x k y i) = Function.update (fun i => φ i (x i)) k (φ k y) i

@[simp] theorem **contractMultilinear_apply** (φ : ∀ i, X i →ₗ[R] R) (j : ι) (x : ∀ i, X i) : contractMultilinear φ j x = (∏ i ∈ Finset.univ.erase j, φ i (x i)) • x j

theorem **contract_def** (φ : ∀ i, X i →ₗ[R] R) (j : ι) : contract φ j = PiTensorProduct.lift (contractMultilinear φ j)

@[simp] theorem **contract_tprod** (φ : ∀ i, X i →ₗ[R] R) (j : ι) (x : ∀ i, X i) : contract φ j (⊗ₜ[R] i, x i) = (∏ i ∈ Finset.univ.erase j, φ i (x i)) • x j

@[simp] theorem **prodMultilinear_apply** (φ : ∀ i, X i →ₗ[R] R) (x : ∀ i, X i) : prodMultilinear φ x = ∏ i, φ i (x i)

theorem **prodForm_def** (φ : ∀ i, X i →ₗ[R] R) : prodForm φ = PiTensorProduct.lift (prodMultilinear φ)

@[simp] theorem **prodForm_tprod** (φ : ∀ i, X i →ₗ[R] R) (x : ∀ i, X i) : prodForm φ (⊗ₜ[R] i, x i) = ∏ i, φ i (x i)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CstSmoothAndFlat

theorem **cst_smooth_and_flat** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (V : Y.Opens), IsAffineOpen V → (Smooth f → (cst f V).Smooth) ∧ (Flat f → (cst f V).Flat)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexAcyclicIffQuasiIsoCechAug

theorem **lowerFace_obj** (A : Cube.{v} ι R) (s : Finset κ) : (lowerFace e A).obj s = A.obj (s.map e.toEmbedding)

theorem **upperFace_obj** (A : Cube.{v} ι R) (s : Finset κ) : (upperFace e i A).obj s = A.obj (insert i (s.map e.toEmbedding))

theorem **res_lowerFace** (A : Cube.{v} ι R) {s t : Finset κ} (h : s ⊆ t) : res (lowerFace e A) h = res A (Finset.map_subset_map.2 h)

theorem **res_upperFace** (A : Cube.{v} ι R) {s t : Finset κ} (h : s ⊆ t) : res (upperFace e i A) h = res A (Finset.insert_subset_insert i (Finset.map_subset_map.2 h))

theorem **faceMap_app** (A : Cube.{v} ι R) (s : Finset κ) : ((faceMap e i A).app s).hom = res A (Finset.subset_insert i (s.map e.toEmbedding))

theorem **lowerFaceMap_app** {A B : Cube.{v} ι R} (φ : A ⟶ B) (s : Finset κ) : (lowerFaceMap e φ).app s = φ.app (s.map e.toEmbedding)

theorem **upperFaceMap_app** {A B : Cube.{v} ι R} (φ : A ⟶ B) (s : Finset κ) : (upperFaceMap e i φ).app s = φ.app (insert i (s.map e.toEmbedding))

theorem **faceMap_naturality** {A B : Cube.{v} ι R} (φ : A ⟶ B) : lowerFaceMap e φ ≫ faceMap e i B = faceMap e i A ≫ upperFaceMap e i φ

theorem **cubeSign_map** (s : Finset κ) (k : κ) : cubeSign (s.map e.toEmbedding) (e k) = cubeSign s k

theorem **upperSign_mul_self** (s : Finset κ) : upperSign e i s * upperSign e i s = 1

theorem **upperSign_smul_smul** {M : Type*} [AddCommGroup M] (s : Finset κ) (x : M) : upperSign e i s • upperSign e i s • x = x

theorem **upperSign_eq_one** (h : ∀ k, e k < i) (s : Finset κ) : upperSign e i s = 1

theorem **upperSign_eq_pow** (h : ∀ k, i ≤ e k) (s : Finset κ) : upperSign e i s = (-1) ^ s.card

theorem **totalD_map** (A : Cube.{v} ι R) (x : ∀ t, A.obj t) (s : Finset κ) : totalD A x (s.map e.toEmbedding) = totalD (lowerFace e A) (fun s => x (s.map e.toEmbedding)) s

theorem **notMem_map_of_range_eq** (he : Set.range e = {i}ᶜ) (s : Finset κ) : i ∉ s.map e.toEmbedding

theorem **map_preimage_of_notMem** (he : Set.range e = {i}ᶜ) {t : Finset ι} (ht : i ∉ t) : (t.preimage e e.injective.injOn).map e.toEmbedding = t

theorem **preimage_insert_map** (he : Set.range e = {i}ᶜ) (s : Finset κ) : (insert i (s.map e.toEmbedding)).preimage e e.injective.injOn = s

theorem **preimage_map_orderEmbedding** (s : Finset κ) : (s.map e.toEmbedding).preimage e e.injective.injOn = s

theorem **exists_eq_map_or_eq_insert** (he : Set.range e = {i}ᶜ) (t : Finset ι) : (∃ s : Finset κ, t = s.map e.toEmbedding) ∨ ∃ s : Finset κ, t = insert i (s.map e.toEmbedding)

theorem **card_insert_map** (he : Set.range e = {i}ᶜ) (s : Finset κ) : (insert i (s.map e.toEmbedding)).card = s.card + 1

theorem **totalD_insert_map** (he : Set.range e = {i}ᶜ) (A : Cube.{v} ι R) (x : ∀ t, A.obj t) (s : Finset κ) : upperSign e i s • totalD A x (insert i (s.map e.toEmbedding)) = totalD (upperFace e i A) (fun s => upperSign e i s • x (insert i (s.map e.toEmbedding))) s + (-1 : ℤ) ^ s.card • ((faceMap e i A).app s).hom (x (s.map e.toEmbedding))

theorem **termLower_apply** (A : Cube.{v} ι R) (p : ℕ) (x : Term A p) (s : {s : Finset κ // s.card = p}) : termLower e A p x s = x ⟨s.1.map e.toEmbedding, (Finset.card_map _).trans s.2⟩

theorem **termUpper_apply** (he : Set.range e = {i}ᶜ) (A : Cube.{v} ι R) (p : ℕ) (x : Term A (p + 1)) (s : {s : Finset κ // s.card = p}) : termUpper he A p x s = upperSign e i s.1 • x ⟨insert i (s.1.map e.toEmbedding), by rw [card_insert_map he, s.2]⟩

theorem **termIncl_apply_of_notMem** {p : ℕ} (y : Term (upperFace e i A) p) (t : {t : Finset ι // t.card = p + 1}) (ht : i ∉ t.1) : termIncl e i A p y t = 0

theorem **termIncl_apply_insert** {p : ℕ} (y : Term (upperFace e i A) p) (s : {s : Finset κ // s.card = p}) : termIncl e i A p y ⟨insert i (s.1.map e.toEmbedding), by rw [card_insert_map he, s.2]⟩ = upperSign e i s.1 • y s

theorem **termSec_apply_of_mem** {p : ℕ} (y : Term (lowerFace e A) p) (t : {t : Finset ι // t.card = p}) (ht : i ∈ t.1) : termSec e i A p y t = 0

theorem **termSec_apply_map** {p : ℕ} (y : Term (lowerFace e A) p) (s : {s : Finset κ // s.card = p}) : termSec e i A p y (s.1.map e.toEmbedding, (Finset.card_map _).trans s.2) = y s

theorem **termUpper_termIncl** {p : ℕ} (y : Term (upperFace e i A) p) : termUpper he A p (termIncl e i A p y) = y

theorem **termLower_termIncl** {p : ℕ} (y : Term (upperFace e i A) p) : termLower e A (p + 1) (termIncl e i A p y) = 0

theorem **termLower_termSec** {p : ℕ} (y : Term (lowerFace e A) p) : termLower e A p (termSec e i A p y) = y

theorem **termUpper_termSec** {p : ℕ} (y : Term (lowerFace e A) (p + 1)) : termUpper he A p (termSec e i A (p + 1) y) = 0

theorem **term_ext_faces** {p : ℕ} {x y : Term A (p + 1)} (h₁ : termUpper he A p x = termUpper he A p y) (h₂ : termLower e A (p + 1) x = termLower e A (p + 1) y) : x = y

theorem **termLower_zero_injective** : Function.Injective (termLower e A 0)

theorem **termIncl_add_termSec** {p : ℕ} (x : Term A (p + 1)) : termIncl e i A p (termUpper he A p x) + termSec e i A (p + 1) (termLower e A (p + 1) x) = x

theorem **termSec_termLower_zero** (x : Term A 0) : termSec e i A 0 (termLower e A 0 x) = x

theorem **termLower_d** {p : ℕ} (x : Term A p) : termLower e A (p + 1) (d A p x) = d (lowerFace e A) p (termLower e A p x)

theorem **termUpper_d** {p : ℕ} (x : Term A (p + 1)) : termUpper he A (p + 1) (d A (p + 1) x) = d (upperFace e i A) p (termUpper he A p x) + (-1 : ℤ) ^ (p + 1) • termMap (faceMap e i A) (p + 1) (termLower e A (p + 1) x)

theorem **termUpper_d_zero** (x : Term A 0) : termUpper he A 0 (d A 0 x) = termMap (faceMap e i A) 0 (termLower e A 0 x)

theorem **d_termIncl** {p : ℕ} (y : Term (upperFace e i A) p) : d A (p + 1) (termIncl e i A p y) = termIncl e i A (p + 1) (d (upperFace e i A) p y)

theorem **d_termSec** {p : ℕ} (y : Term (lowerFace e A) p) : d A p (termSec e i A p y) = termSec e i A (p + 1) (d (lowerFace e A) p y) + termIncl e i A p ((-1 : ℤ) ^ p • termMap (faceMap e i A) p y)

theorem **termLower_termMap** {p : ℕ} (x : Term A p) : termLower e B p (termMap φ p x) = termMap (lowerFaceMap e φ) p (termLower e A p x)

theorem **termUpper_termMap** {p : ℕ} (x : Term A (p + 1)) : termUpper he B p (termMap φ (p + 1) x) = termMap (upperFaceMap e i φ) p (termUpper he A p x)

theorem **termMap_termIncl** {p : ℕ} (y : Term (upperFace e i A) p) : termMap φ (p + 1) (termIncl e i A p y) = termIncl e i B p (termMap (upperFaceMap e i φ) p y)

theorem **succEmbedding_f** (p : ℕ) : succEmbedding.f p = p + 1

theorem **succEmbedding_f_ne_zero** (p : ℕ) : succEmbedding.f p ≠ 0

theorem **isZero_shiftSucc_X_zero** (K : CochainComplex (ModuleCat.{v} R) ℕ) : IsZero ((shiftSucc K).X 0)

theorem **isZero_shiftSucc_homology_zero** (K : CochainComplex (ModuleCat.{v} R) ℕ) : IsZero ((shiftSucc K).homology 0)

theorem **shiftSucc_d** (K : CochainComplex (ModuleCat.{v} R) ℕ) (p : ℕ) : (shiftSucc K).d (p + 1) (p + 1 + 1) = (shiftSuccXIso K p).hom ≫ K.d p (p + 1) ≫ (shiftSuccXIso K (p + 1)).inv

theorem **cubeLowerProj_f** (A : Cube.{v} ι R) (p : ℕ) : (cubeLowerProj e A).f p = ModuleCat.ofHom (Cube.termLower e A p)

theorem **cubeLowerProj_naturality** {A B : Cube.{v} ι R} (φ : A ⟶ B) : cubeComplexMap φ ≫ cubeLowerProj e B = cubeLowerProj e A ≫ cubeComplexMap (Cube.lowerFaceMap e φ)

theorem **isIso_cubeLowerProj** (he : Function.Surjective e) (A : Cube.{v} ι R) : IsIso (cubeLowerProj e A)

theorem **cubeUpperIncl_f_succ** (A : Cube.{v} ι R) (p : ℕ) : (cubeUpperIncl he A).f (p + 1) = (shiftSuccXIso _ p).hom ≫ cubeInclX e i A p

theorem **cubeUpperIncl_naturality** {A B : Cube.{v} ι R} (φ : A ⟶ B) : extendMap (cubeComplexMap (Cube.upperFaceMap e i φ)) succEmbedding ≫ cubeUpperIncl he B = cubeUpperIncl he A ≫ cubeComplexMap φ

theorem **cubeFaceSequence_shortExact** (A : Cube.{v} ι R) : (cubeFaceSequence he A).ShortExact

theorem **cubeFaceSequence_δ** (A : Cube.{v} ι R) (p : ℕ) : (cubeFaceSequence_shortExact he A).δ p (p + 1) rfl ≫ (shiftSuccHomologyIso (cubeComplex (Cube.upperFace e i A)) p).hom = (-1 : ℤ) ^ p • homologyMap (cubeComplexMap (Cube.faceMap e i A)) p

theorem **mono_homologyMap_τ₂** (j : ι) (h₁ : Mono (homologyMap φ.τ₁ j)) (h₂ : Mono (homologyMap φ.τ₃ j)) (h₃ : ∀ i, c.Rel i j → Epi (homologyMap φ.τ₃ i)) : Mono (homologyMap φ.τ₂ j)

theorem **epi_homologyMap_τ₂** (i : ι) (h₁ : Epi (homologyMap φ.τ₁ i)) (h₂ : Epi (homologyMap φ.τ₃ i)) (h₃ : ∀ j, c.Rel i j → Mono (homologyMap φ.τ₁ j)) : Epi (homologyMap φ.τ₂ i)

theorem **quasiIso_τ₂** (h₁ : QuasiIso φ.τ₁) (h₃ : QuasiIso φ.τ₃) : QuasiIso φ.τ₂

theorem **quasiIso_cubeComplexMap_of_faces** {A B : Cube.{v} ι R} (φ : A ⟶ B) (h₀ : QuasiIso (cubeComplexMap (Cube.lowerFaceMap e φ))) (h₁ : QuasiIso (cubeComplexMap (Cube.upperFaceMap e i φ))) : QuasiIso (cubeComplexMap φ)

theorem **cubeFaceSequence_δ_eq** (A : Cube.{v} ι R) (p : ℕ) : (cubeFaceSequence_shortExact he A).δ p (p + 1) rfl = homologyMap (cubeComplexMap (Cube.faceMap e i A)) p ≫ (Int.negOnePow p • shiftSuccHomologyIso (cubeComplex (Cube.upperFace e i A)) p).inv

theorem **mono_cubeFaceSequence_δ_iff** (A : Cube.{v} ι R) (p : ℕ) : Mono ((cubeFaceSequence_shortExact he A).δ p (p + 1) rfl) ↔ Mono (homologyMap (cubeComplexMap (Cube.faceMap e i A)) p)

theorem **epi_cubeFaceSequence_δ_iff** (A : Cube.{v} ι R) (p : ℕ) : Epi ((cubeFaceSequence_shortExact he A).δ p (p + 1) rfl) ↔ Epi (homologyMap (cubeComplexMap (Cube.faceMap e i A)) p)

theorem **cubeComplex_exactAt_zero_iff_faces** (A : Cube.{v} ι R) : (cubeComplex A).ExactAt 0 ↔ Mono (homologyMap (cubeComplexMap (Cube.faceMap e i A)) 0)

theorem **cubeComplex_exactAt_succ_iff_faces** (A : Cube.{v} ι R) (p : ℕ) : (cubeComplex A).ExactAt (p + 1) ↔ Epi (homologyMap (cubeComplexMap (Cube.faceMap e i A)) p) ∧ Mono (homologyMap (cubeComplexMap (Cube.faceMap e i A)) (p + 1))

theorem **cubeComplex_acyclic_iff_quasiIso_faceMap** (A : Cube.{v} ι R) : (cubeComplex A).Acyclic ↔ QuasiIso (cubeComplexMap (Cube.faceMap e i A))

theorem **cubeComplex_exactAt_succ_of_faces** (A : Cube.{v} ι R) (p : ℕ) (h₀ : (cubeComplex (Cube.lowerFace e A)).ExactAt (p + 1)) (h₁ : (cubeComplex (Cube.upperFace e i A)).ExactAt p) : (cubeComplex A).ExactAt (p + 1)

theorem **cubeComplex_acyclic_of_faces** (A : Cube.{v} ι R) (h₀ : (cubeComplex (Cube.lowerFace e A)).Acyclic) (h₁ : (cubeComplex (Cube.upperFace e i A)).Acyclic) : (cubeComplex A).Acyclic

theorem **quasiIsoAt_zero_cubeComplexMap_iff** {A B : Cube.{v} ι R} (φ : A ⟶ B) : QuasiIsoAt (cubeComplexMap φ) 0 ↔ Set.BijOn (φ.app ⌀).hom (LinearMap.ker (Cube.aug A)) (LinearMap.ker (Cube.aug B))

theorem **cubeComplex_acyclic_of_faces_of_bijOn** {κ : Type} [LinearOrder κ] {e : κ ↪o ι} {i : ι} (he : Set.range e = {i}ᶜ) (A : Cube.{v} ι R) (h₀ : ∀ p, (cubeComplex (Cube.lowerFace e A)).ExactAt (p + 1)) (h₁ : ∀ p, (cubeComplex (Cube.upperFace e i A)).ExactAt (p + 1)) (h : Set.BijOn ((Cube.faceMap e i A).app ∅).hom (LinearMap.ker (Cube.aug (Cube.lowerFace e A))) (LinearMap.ker (Cube.aug (Cube.upperFace e i A)))) : (cubeComplex A).Acyclic

theorem **cubeComplex_acyclic_of_bijective** (A : Cube.{v} ι R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → Function.Bijective (Cube.res A (Finset.subset_insert i s))) : (cubeComplex A).Acyclic

theorem **cechAugHomology_comp** (A : Cube.{v} ι R) : cechAugHomology A ≫ (cechHomologyZeroIso A).hom ≫ ModuleCat.ofHom (LinearMap.ker (Cube.d A 1)).subtype = ModuleCat.ofHom (Cube.aug A)

theorem **cechAugHomology_naturality** {A B : Cube.{v} ι R} (φ : A ⟶ B) : φ.app ∅ ≫ cechAugHomology B = cechAugHomology A ≫ homologyMap (cechComplexMap φ) 0

theorem **isIso_cechAugHomology_iff** (A : Cube.{v} ι R) : IsIso (cechAugHomology A) ↔ (cubeComplex A).ExactAt 0 ∧ (cubeComplex A).ExactAt 1

theorem **cubeComplex_acyclic_iff_cech** (A : Cube.{v} ι R) : (cubeComplex A).Acyclic ↔ IsIso (cechAugHomology A) ∧ ∀ p, IsZero ((cechComplex A).homology (p + 1))

theorem **cubeComplex_acyclic_iff_exact** (A : Cube.{v} ι R) : (cubeComplex A).Acyclic ↔ Function.Injective (Cube.aug A) ∧ Function.Exact (Cube.aug A) (Cube.d A 1) ∧ ∀ p, Function.Exact (Cube.d A (p + 1)) (Cube.d A (p + 2))

theorem **quasiIsoAt_zero_cechAug_iff** (A : Cube.{v} ι R) : QuasiIsoAt (cechAug A) 0 ↔ (cubeComplex A).ExactAt 0 ∧ (cubeComplex A).ExactAt 1

theorem **cubeComplex_acyclic_iff_quasiIso_cechAug** (A : Cube.{v} ι R) : (cubeComplex A).Acyclic ↔ QuasiIso (cechAug A)

theorem **range_otherDirections** (i : ι) : Set.range (otherDirections i) = {i}ᶜ

theorem **cube_cone_of_faces** {κ : Type} [LinearOrder κ] {e : κ ↪o ι} {i : ι} (he : Set.range e = {i}ᶜ) (A : Cube.{v} ι R) : (∃ hS : (ShortComplex.mk (cubeUpperIncl he A) (cubeLowerProj e A) (cubeFaceSequence he A).zero).ShortExact, (∀ p, Nonempty ((cubeFaceSequence he A).map (eval _ _ p)).Splitting) ∧ ∀ p, hS.δ p (p + 1) rfl ≫ (shiftSuccHomologyIso (cubeComplex (Cube.upperFace e i A)) p).hom = (-1 : ℤ) ^ p • homologyMap (cubeComplexMap (Cube.faceMap e i A)) p) ∧ ((∀ s : Finset ι, i ∉ s → Function.Bijective (Cube.res A (Finset.subset_insert i s))) → (cubeComplex A).Acyclic) ∧ (∀ (B : Cube.{v} ι R) (φ : A ⟶ B), QuasiIso (cubeComplexMap (Cube.lowerFaceMap e φ)) → QuasiIso (cubeComplexMap (Cube.upperFaceMap e i φ)) → QuasiIso (cubeComplexMap φ)) ∧ ((cubeComplex A).Acyclic ↔ IsIso (cechAugHomology A) ∧ ∀ p, IsZero ((cechComplex A).homology (p + 1)))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexExactAtOfIsLocalizedModuleRes

theorem **exists_smul_eq_zero_of_forall** [Finite ι] (A : Cube.{v} ι R) (S : Submonoid R) {p : ℕ} (x : Term A p) (h : ∀ s, ∃ c ∈ S, c • x s = 0) : ∃ c ∈ S, c • x = 0

theorem **exists_smul_eq_zero_of_d_eq_zero** [Finite ι] (A : Cube.{v} ι R) (S : Submonoid R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → ∀ y, res A (Finset.subset_insert i s) y = 0 → ∃ c ∈ S, c • y = 0) {p : ℕ} (x : Term A p) (hx : d A p x = 0) (h0 : ∀ s : {s : Finset ι // s.card = p}, i ∈ s.1 → x s = 0) : ∃ c ∈ S, c • x = 0

theorem **exists_smul_eq_d** [Finite ι] (A : Cube.{v} ι R) (S : Submonoid R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → ∀ y, res A (Finset.subset_insert i s) y = 0 → ∃ c ∈ S, c • y = 0) (hA' : ∀ s : Finset ι, i ∉ s → ∀ z, ∃ c ∈ S, ∃ y, res A (Finset.subset_insert i s) y = c • z) {p : ℕ} (x : Term A (p + 1)) (hx : d A (p + 1) x = 0) : ∃ c ∈ S, ∃ y, d A p y = c • x

theorem **smul_homologyπ_eq_zero** (K : CochainComplex (ModuleCat.{v} R) ℕ) (q p : ℕ) (z : K.cycles p) (c : R) (y : K.X q) (h : (K.d q p).hom y = c • (K.iCycles p).hom z) : c • (K.homologyπ p).hom z = 0

theorem **cubeComplex_exists_smul_homology_eq_zero** [Finite ι] (A : Cube.{v} ι R) (S : Submonoid R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → ∀ y, Cube.res A (Finset.subset_insert i s) y = 0 → ∃ c ∈ S, c • y = 0) (hA' : ∀ s : Finset ι, i ∉ s → ∀ z, ∃ c ∈ S, ∃ y, Cube.res A (Finset.subset_insert i s) y = c • z) (p : ℕ) (x : (cubeComplex A).homology p) : ∃ c ∈ S, c • x = 0

theorem **cubeComplex_exists_smul_homology_eq_zero_of_isLocalizedModule** [Finite ι] (A : Cube.{v} ι R) (S : Submonoid R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → IsLocalizedModule S (Cube.res A (Finset.subset_insert i s))) (p : ℕ) (x : (cubeComplex A).homology p) : ∃ c ∈ S, c • x = 0

theorem **cubeComplex_exists_pow_smul_homology_eq_zero** [Finite ι] (A : Cube.{v} ι R) (a : R) (i : ι) (hA : ∀ s : Finset ι, i ∉ s → IsLocalizedModule.Away a (Cube.res A (Finset.subset_insert i s))) (p : ℕ) (x : (cubeComplex A).homology p) : ∃ n : ℕ, a ^ n • x = 0

theorem **cubeComplex_exactAt_of_isLocalizedModule_res** [Finite ι] (A : Cube.{v} ι R) (f : ι → R) (hA : ∀ (i : ι) (s : Finset ι), i ∉ s → IsLocalizedModule.Away (f i) (Cube.res A (Finset.subset_insert i s))) (hf : Ideal.span (Set.range f) = ⊤) (p : ℕ) : (cubeComplex A).ExactAt p

theorem **isUnit_algebraMap_end_of_dvd** {a b : R} (h : a ∣ b) (hb : IsUnit (algebraMap R (Module.End R N) b)) : IsUnit (algebraMap R (Module.End R N) a)

theorem **isUnit_powers_of_dvd** {a b : R} (h : a ∣ b) (g : M →ₗ[R] N) [IsLocalizedModule.Away b g] (x : Submonoid.powers a) : IsUnit (algebraMap R (Module.End R N) x)

theorem **isLocalizedModule_away_of_comp** {a b : R} (g : M →ₗ[R] N) (h : N →ₗ[R] P) [IsLocalizedModule.Away b g] [IsLocalizedModule.Away (a * b) (h ∘ₗ g)] : IsLocalizedModule.Away a h

theorem **Cube.isLocalizedModule_res_insert** [DecidableEq ι] (A : Cube.{v} ι R) (f : ι → R) {M : Type*} [AddCommGroup M] [Module R M] (g : ∀ s, M →ₗ[R] A.obj s) (hg : ∀ s, IsLocalizedModule.Away (∏ i ∈ s, f i) (g s)) (hres : ∀ {s t : Finset ι} (h : s ⊆ t) (m : M), Cube.res A h (g s m) = g t m) (i : ι) (s : Finset ι) (hi : i ∉ s) : IsLocalizedModule.Away (f i) (Cube.res A (Finset.subset_insert i s))

theorem **cubeComplex_exactAt_of_isLocalizedModule** [LinearOrder ι] [Finite ι] (A : Cube.{v} ι R) (f : ι → R) {M : Type*} [AddCommGroup M] [Module R M] (g : ∀ s, M →ₗ[R] A.obj s) (hg : ∀ s, IsLocalizedModule.Away (∏ i ∈ s, f i) (g s)) (hres : ∀ {s t : Finset ι} (h : s ⊆ t) (m : M), Cube.res A h (g s m) = g t m) (hf : Ideal.span (Set.range f) = ⊤) (p : ℕ) : (cubeComplex A).ExactAt p

theorem **locRes_mk_one** {s t : Finset ι} (h : s ⊆ t) (m : M) : locRes f M h (LocalizedModule.mk m 1) = LocalizedModule.mk m 1

theorem **locRes_comp_mkLinearMap** {s t : Finset ι} (h : s ⊆ t) : locRes f M h ∘ₗ LocalizedModule.mkLinearMap _ M = LocalizedModule.mkLinearMap _ M

theorem **locRes_mk** {s t : Finset ι} (h : s ⊆ t) (m : M) (a : Submonoid.powers (∏ i ∈ s, f i)) (b : Submonoid.powers (∏ i ∈ t, f i)) (c : R) (hb : (b : R) = c * a) : locRes f M h (LocalizedModule.mk m a) = LocalizedModule.mk (c • m) b

theorem **locRes_self** (s : Finset ι) : locRes f M (subset_refl s) = LinearMap.id

theorem **locRes_comp** {s t w : Finset ι} (h : s ⊆ t) (h' : t ⊆ w) : locRes f M h' ∘ₗ locRes f M h = locRes f M (h.trans h')

theorem **res_toLocCube** {s t : Finset ι} (h : s ⊆ t) (m : M) : Cube.res (locCube f M) h (toLocCube f M s m) = toLocCube f M t m

theorem **isLocalizedModule_res_locCube** [DecidableEq ι] (i : ι) (s : Finset ι) (hi : i ∉ s) : IsLocalizedModule.Away (f i) (Cube.res (locCube f M) (Finset.subset_insert i s))

theorem **toLocCube_empty_bijective** : Function.Bijective (toLocCube f M ∅)

theorem **locCubeEmptyEquiv_apply** (m : M) : locCubeEmptyEquiv f M m = toLocCube f M ∅ m

theorem **isLocalizedModule_toLocCube_singleton** (i : ι) : IsLocalizedModule.Away (f i) (toLocCube f M {i})

theorem **locCubeMap_app_toLocCube** {N : Type v} [AddCommGroup N] [Module R N] (φ : M →ₗ[R] N) (s : Finset ι) (m : M) : ((locCubeMap f φ).app s).hom (toLocCube f M s m) = toLocCube f N s (φ m)

theorem **locCubeMap_id** : locCubeMap f (LinearMap.id : M →ₗ[R] M) = 𝟙 _

theorem **locCube_exactAt** (hf : Ideal.span (Set.range f) = ⊤) (p : ℕ) : (cubeComplex (locCube f M)).ExactAt p

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexExactAtOfMonomialBasis

theorem **cubeComplex_above_exactAt** [LinearOrder ι] [Fintype ι] (w : Finset ι) {p : ℕ} (hp : p ≠ Fintype.card ι) : (cubeComplex (Cube.above R w)).ExactAt p

theorem **monomialCoord_basis** {κ : Type} [DecidableEq κ] (T : κ → Finset ι) {s : Finset ι} {M : Type} [AddCommGroup M] [Module R M] (b : Module.Basis {a : κ // T a ⊆ s} R M) (a : {a : κ // T a ⊆ s}) : monomialCoord T b (b a) = DirectSum.lof R κ (fun a => PLift (T a ⊆ s) →₀ R) a.1 (Finsupp.single ⟨a.2⟩ 1)

theorem **cubeComplex_exactAt_of_monomialBasis** (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (κ : Type) (T : κ → Finset ι) (A : Cube.{0} ι R) (b : ∀ s : Finset ι, Module.Basis {a : κ // T a ⊆ s} R (A.obj s)) (hb : ∀ (s t : Finset ι) (h : s ⊆ t) (a : {a : κ // T a ⊆ s}), Cube.res A h (b s a) = b t ⟨a.1, a.2.trans h⟩) (p : ℕ) (hp : p ≠ Fintype.card ι) : (cubeComplex A).ExactAt p

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexGrFreeFiniteAndExactAt

theorem **cubeComplex_grFree_finite_and_exactAt** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (r : ℕ) (d : Fin r → ℤ), (∀ n p, Module.Finite R ((cubeComplex (grCube ι (grFree ι R d) n)).homology p)) ∧ ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∀ p, (cubeComplex (grCube ι (grFree ι R d) n)).ExactAt p

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexGradedModuleExactAtOfLargeDegree

theorem **exactAt_quot_of_shortExact** {R : Type} [CommRing R] {S : ShortComplex (CochainComplex (ModuleCat.{0} R) ℕ)} (hS : S.ShortExact) (p : ℕ) (h₂ : S.X₂.ExactAt p) (h₁ : S.X₁.ExactAt (p + 1)) : S.X₃.ExactAt p

theorem **cubeComplex_grCube_exactAt_of_card_lt** (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (q : ℕ) : ∀ (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] [Decomposition 𝒜], Module.Finite (MvPolynomial ι R) M → ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∀ p, Fintype.card ι < p + q → (cubeComplex (grCube ι 𝒜 n)).ExactAt p

theorem **cubeComplex_grCube_exactAt_of_finite** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] [Decomposition 𝒜], Module.Finite (MvPolynomial ι R) M → ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∀ p, (cubeComplex (grCube ι 𝒜 n)).ExactAt p

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexGradedModuleExactAtOfLargeDegreeStatement

theorem **_root_.GlobalTheory.T.cubeComplex_gradedModule_exactAt_of_large_degree** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] [Decomposition 𝒜], Module.Finite (MvPolynomial ι R) M → ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∀ p, (cubeComplex (grCube ι 𝒜 n)).ExactAt p

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexPolyGradingFiniteExact

theorem **finite_neg_sum_eq** (ι : Type) [Fintype ι] (n : ℤ) : {a : {a : ι → ℤ // ∑ i, a i = n} | Finset.univ.filter (fun i => a.1 i < 0) = Finset.univ}.Finite

theorem **cubeComplex_polyGrading_finite_exact** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (n : ℤ), (∀ p, Module.Finite R ((cubeComplex (grCube ι (polyGrading ι R) n)).homology p)) ∧ (1 ≤ n → ∀ p, (cubeComplex (grCube ι (polyGrading ι R) n)).ExactAt p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.CubeComplexPolyGradingFiniteExactStatement

theorem **_root_.GlobalTheory.T.finite_homology_and_exact_cubeComplex_polyGrading** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (n : ℤ), (∀ p, Module.Finite R ((cubeComplex (grCube ι (polyGrading ι R) n)).homology p)) ∧ (1 ≤ n → ∀ p, (cubeComplex (grCube ι (polyGrading ι R) n)).ExactAt p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.DehomogenisingCompOfSaturating

theorem **dehomogenising_comp_of_saturating** {ι : Type} [LinearOrder ι] [Fintype ι] {R : Type} [CommRing R] {N : Type} [AddCommGroup N] [Module R N] [Module (MvPolynomial ι R) N] [IsScalarTower R (MvPolynomial ι R) N] (𝒩 : ℤ → Submodule R N) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩] {G : Type} [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) {ψ : N →ₗ[MvPolynomial ι R] G} (hψ : IsGradedMap 𝒩 𝒢 ψ) (i : ι) (h₁ : ∀ y, ψ y = 0 → ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • y = 0) (h₂ : ∀ z : G, ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • z ∈ LinearMap.range ψ) {B : Type} [CommRing B] (π : MvPolynomial ι R →+* B) {P : Type} [AddCommGroup P] [Module B P] (ε : G →ₛₗ[π] P) (hπ : π (MvPolynomial.X i) = 1) (hε : Function.Surjective ε) (hk : ∀ n, ∀ y ∈ 𝒢 n, ε y = 0 → ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • y = 0) : Function.Surjective (ε.co

theorem **exists_noetherian_model_grCube** (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (G : Type) [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) [PolyGradedSMul ι 𝒢] [Decomposition 𝒢] (H : ∀ i : ι, ∃ (B : Type) (_ : CommRing B) (_ : Algebra R B) (π : MvPolynomial ι R →ₐ[R] B) (P : Type) (_ : AddCommGroup P) (_ : Module B P) (ε : G →ₛₗ[(π : MvPolynomial ι R →+* B)] P), Algebra.Smooth R B ∧ Module.FinitePresentation B P ∧ Module.Projective B P ∧ Function.Surjective π ∧ π (MvPolynomial.X i) = 1 ∧ Function.Surjective ε ∧ ∀ n, ∀ y ∈ 𝒢 n, ε y = 0 → ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • y = 0) : ∃ (R₀ : Subring R) (_ : IsNoetherianRing R₀) (C₀ : Cube.{0} ι R₀) (θ : Cube.baseChange R C₀ ⟶ grCube ι 𝒢 0), (∀ I : Finset ι, I.Nonempty → Module.Flat R₀ (C₀.obj I)) ∧ (∀ p, Module.Finite R₀ ((cechC

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.DehomogenisingEqZeroIff

theorem **dehomogenising_eq_zero_iff** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩] (i : ι) (B : Type) [CommRing B] (π : MvPolynomial ι R →+* B) (P : Type) [AddCommGroup P] [Module B P] (ε : M →ₛₗ[π] P), π (X i) = 1 → (∀ n, ∀ y ∈ 𝒩 n, ε y = 0 → ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y = 0) → ∀ y, ε y = 0 ↔ ∃ w : M, y = (X i - 1 : MvPolynomial ι R) • w

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.DoubleCosetSetoidRelIffMem

@[simp] theorem **mem_repStabilizer** (ρ : Representation k G V) (v : V) (g : G) : g ∈ repStabilizer ρ v ↔ ρ g v = v

@[simp] theorem **mem_clmStabilizer** (π : G →* (E →L[𝕜] E)) (v : E) (g : G) : g ∈ clmStabilizer π v ↔ π g v = v

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.EventuallyExistsFiniteSubsetNonzeroLocus

theorem **topologicalKrullDim_add_one_le** {T : Type*} [TopologicalSpace T] [QuasiSober T] {s t : Set T} (hs : IsClosed s) (ht : IsClosed t) (h : ∀ x ∈ s, ∃ y ∈ t, y ∉ s ∧ y ⤳ x) : topologicalKrullDim s + 1 ≤ topologicalKrullDim t

theorem **isMulBy_secHom** (F : X.Modules) (hL : L.IsLB) (s : L.Sec) : IsMulBy (secHom F s) s

theorem **mem_range_app_iff** (H : IsMulBy φ s) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (t : Γ(F, V)) : t ∈ Set.range (φ.app V) ↔ ∃ u, resG X h (s.f i) • u = t

theorem **mem_range_stalkLinear_iff** (H : IsMulBy φ s) (i : L.ι) {x : X} (hx : x ∈ L.U i) (t : (F.presheaf.stalk x :)) : t ∈ Set.range (stalkLinear φ x) ↔ ∃ u, X.presheaf.germ _ x hx (s.f i) • u = t

theorem **injective_app_iff** (H : IsMulBy φ s) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) : Function.Injective (φ.app V) ↔ IsSMulRegular Γ(F, V) (resG X h (s.f i))

theorem **mono_iff** [IsLocallyNoetherian X] [F.IsQuasicoherent] (H : IsMulBy φ s) (hL : L.IsLB) : Mono φ ↔ assPoints F ⊆ nonzeroLocus s

theorem **stalkLinear_π_eq_zero_iff** (H : IsMulBy φ s) (i : L.ι) {x : X} (hx : x ∈ L.U i) (t : (F.presheaf.stalk x :)) : stalkLinear (cokernel.π φ) x t = 0 ↔ ∃ u, X.presheaf.germ _ x hx (s.f i) • u = t

theorem **support_cokernel_subset** (H : IsMulBy φ s) : support (cokernel φ) ⊆ support F \ nonzeroLocus s

theorem **support_cokernel** [IsFiniteQC F] (H : IsMulBy φ s) (hL : L.IsLB) : support (cokernel φ) = support F \ nonzeroLocus s

theorem **suppDim_cokernel_add_one_le** [IsLocallyNoetherian X] [IsFiniteQC F] (H : IsMulBy φ s) (hL : L.IsLB) [Mono φ] : suppDim (cokernel φ) + 1 ≤ suppDim F

theorem **ideal_secIdeal** (hL : L.IsLB) (s : L.Sec) (W : X.affineOpens) (i : L.ι) (h : W.1 ≤ L.U i) : (secIdeal hL s).ideal W = Ideal.span {resG X h (s.f i)}

theorem **map_germ_ideal_secIdeal** (hL : L.IsLB) (s : L.Sec) (W : X.affineOpens) {x : X} (hx : x ∈ W.1) (i : L.ι) (hi : x ∈ L.U i) : ((secIdeal hL s).ideal W).map (X.presheaf.germ W.1 x hx).hom = Ideal.span {X.presheaf.germ _ x hi (s.f i)}

theorem **coe_support_secIdeal** (hL : L.IsLB) (s : L.Sec) : ((secIdeal hL s).support : Set X) = (nonzeroLocus s : Set X)ᶜ

theorem **IsMulBy.ker_π_app** [F.IsQuasicoherent] {φ : G ⟶ F} {s : L.Sec} (H : IsMulBy φ s) (hL : L.IsLB) (W : X.affineOpens) : LinearMap.ker (appLinear (cokernel.π φ) W.1) = (secIdeal hL s).ideal W • ⊤

theorem **surjective_and_ker_of_isBaseChange** {A B M N : Type*} [CommRing A] [CommRing B] [Algebra A B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module A N] [Module B N] [IsScalarTower A B N] {l : M →ₗ[A] N} (hl : IsBaseChange B l) (h : Function.Surjective (algebraMap A B)) : Function.Surjective l ∧ LinearMap.ker l = RingHom.ker (algebraMap A B) • ⊤

theorem **surjective_and_ker_unit_subschemeι** (I : X.IdealSheafData) (F : X.Modules) [F.IsQuasicoherent] (W : X.affineOpens) : Function.Surjective (((Scheme.Modules.pullbackPushforwardAdjunction I.subschemeι).unit.app F).app W.1) ∧ LinearMap.ker (appLinear ((Scheme.Modules.pullbackPushforwardAdjunction I.subschemeι).unit.app F) W.1) = I.ideal W • ⊤

theorem **surjective_and_ker_unitToPush_subschemeι** (I : X.IdealSheafData) (W : X.affineOpens) : Function.Surjective ((unitToPush I.subschemeι).app W.1) ∧ LinearMap.ker (appLinear (unitToPush I.subschemeι) W.1) = I.ideal W • ⊤

theorem **IsMulBy.exists_cokernel_iso** [F.IsQuasicoherent] [G.IsQuasicoherent] {φ : G ⟶ F} {s : L.Sec} (H : IsMulBy φ s) (hL : L.IsLB) {E : X.Modules} [E.IsQuasicoherent] (η : F ⟶ E) (h : ∀ W : X.affineOpens, Function.Surjective (η.app W.1) ∧ LinearMap.ker (appLinear η W.1) = (secIdeal hL s).ideal W • ⊤) : ∃ e : cokernel φ ≅ E, cokernel.π φ ≫ e.hom = η

theorem **cutHom_app** (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (t : Γ(twistZ F L (-1), V)) : (cutHom F hL s).app V t = resG X h (s.f i) • zchartEquiv hL (-1) i h t

theorem **isMulBy_cutHom** : IsMulBy (cutHom F hL s) s

theorem **shortExact_cutComplex** [h : Mono (cutHom F hL s)] : (cutComplex F hL s).ShortExact where

theorem **mono_cutHom_iff** [IsLocallyNoetherian X] [F.IsQuasicoherent] : Mono (cutHom F hL s) ↔ assPoints F ⊆ nonzeroLocus s

theorem **mono_secHom_iff** [IsLocallyNoetherian X] [F.IsQuasicoherent] : Mono (secHom F s) ↔ assPoints F ⊆ nonzeroLocus s

theorem **support_cut** [IsFiniteQC F] : support (cut F hL s) = support F \ nonzeroLocus s

theorem **support_cut_eq_inter** [IsFiniteQC F] : support (cut F hL s) = support F ∩ (secIdeal hL s).support

theorem **surjective_cutπ_app** [F.IsQuasicoherent] (W : X.affineOpens) : Function.Surjective ((cutπ F hL s).app W.1)

theorem **ker_cutπ_app** [F.IsQuasicoherent] (W : X.affineOpens) : LinearMap.ker (appLinear (cutπ F hL s) W.1) = (secIdeal hL s).ideal W • ⊤

theorem **exists_cut_iso_pushforward_pullback** [F.IsQuasicoherent] : ∃ e : cut F hL s ≅ (Scheme.Modules.pushforward (secIdeal hL s).subschemeι).obj ((Scheme.Modules.pullback (secIdeal hL s).subschemeι).obj F), cutπ F hL s ≫ e.hom = (Scheme.Modules.pullbackPushforwardAdjunction (secIdeal hL s).subschemeι).unit.app F

theorem **exists_cut_structureModule_iso** : ∃ e : cut (structureModule X) hL s ≅ (Scheme.Modules.pushforward (secIdeal hL s).subschemeι).obj (structureModule (secIdeal hL s).subscheme), cutπ _ hL s ≫ e.hom = unitToPush (secIdeal hL s).subschemeι

theorem **suppDim_cut_add_one_le** : suppDim (cut F hL s) + 1 ≤ suppDim F

theorem **suppDim_cut_le** {d : ℕ} (h : suppDim F ≤ (d + 1 : ℕ)) : suppDim (cut F hL s) ≤ d

theorem **isZero_cut** (h : suppDim F ≤ 0) : IsZero (cut F hL s)

theorem **FiniteCohTwists.cut** (h : FiniteCohTwists k F) : FiniteCohTwists k (cut F hL s)

theorem **interNum_cons_eq_interNum_cut** (hP : ∀ i, (P i).IsLB) (h : FiniteCohTwists k F) : interNum k F (Fin.cons L P : Fin (d + 1) → LB X) = interNum k (cut F hL s) P

theorem **interNum_cons_eq_interNumSub** [Mono (cutHom (structureModule X) hL s)] (hP : ∀ i, (P i).IsLB) (h : FiniteCohTwists k (structureModule X)) : interNum k (structureModule X) (Fin.cons L P : Fin (d + 1) → LB X) = interNumSub k (secIdeal hL s) P

theorem **exists_finite_subset_nonzeroLocus** {W : Type*} [AddCommGroup W] [Module k W] (φ : W →ₛₗ[algebraMap k Γ(X, ⊤)] L.Sec) {S : Set X} (hS : S.Finite) (h : ∀ x ∈ S, ∃ w, x ∈ nonzeroLocus (φ w)) : ∃ w, S ⊆ nonzeroLocus (φ w)

theorem **exists_mem_finite_subset_nonzeroLocus** [Module k L.Sec] [IsScalarTower k Γ(X, ⊤) L.Sec] (W : Submodule k L.Sec) {S : Set X} (hS : S.Finite) (h : ∀ x ∈ S, ∃ s ∈ W, x ∈ nonzeroLocus s) : ∃ s ∈ W, S ⊆ nonzeroLocus s

theorem **exists_sum_finite_subset_nonzeroLocus** {ι : Type*} [Fintype ι] (s : ι → L.Sec) {S : Set X} (hS : S.Finite) (h : ∀ x ∈ S, ∃ i, x ∈ nonzeroLocus (s i)) : ∃ c : ι → k, S ⊆ nonzeroLocus (∑ i, algebraMap k Γ(X, ⊤) (c i) • s i)

theorem **exists_mem_mono_cutHom** (hL : L.IsLB) [Module k L.Sec] [IsScalarTower k Γ(X, ⊤) L.Sec] (W : Submodule k L.Sec) (h : ∀ x : X, ∃ s ∈ W, x ∈ nonzeroLocus s) : ∃ s ∈ W, Mono (cutHom F hL s)

theorem **exists_sum_mono_cutHom** (hL : L.IsLB) {ι : Type*} [Fintype ι] (s : ι → L.Sec) (h : ⊔ i, nonzeroLocus (s i) = ⊤) : ∃ c : ι → k, Mono (cutHom F hL (∑ i, algebraMap k Γ(X, ⊤) (c i) • s i))

theorem **IsGenerated.exists_mono_cutHom** (h : IsGenerated L) : ∃ s, Mono (cutHom F h.isLB s)

theorem **IsAmpleLB.exists_mono_cutHom** [IsLocallyNoetherian X] (h : IsAmpleLB L) : ∃ n, 1 ≤ n ∧ ∃ s : (lbPow L n).Sec, Mono (cutHom F (isLB_lbPow h.isLB n) s) ∧ IsAffineOpen (nonzeroLocus s)

theorem **IsAmpleLB.eventually_exists_mem_nonzeroLocus_le** (h : IsAmpleLB L) (hM : M.IsLB) {x : X} {U : X.Opens} (hx : x ∈ U) : ∀ᶠ n : ℤ in atTop, ∀ (N : LB X) (hN : N.IsLB), cls N hN = cls M hM + n • cls L h.isLB → ∃ t : N.Sec, x ∈ nonzeroLocus t ∧ nonzeroLocus t ≤ U

theorem **IsAmpleLB.eventually_exists_finite_subset_nonzeroLocus** (h : IsAmpleLB L) (hM : M.IsLB) {S : Set X} (hS : S.Finite) : ∀ᶠ n : ℤ in atTop, ∀ (N : LB X) (hN : N.IsLB), cls N hN = cls M hM + n • cls L h.isLB → ∃ t : N.Sec, S ⊆ nonzeroLocus t

theorem **IsAmpleLB.eventually_exists_mono_cutHom** (h : IsAmpleLB L) (hM : M.IsLB) (F : X.Modules) [IsFiniteQC F] : ∀ᶠ n : ℤ in atTop, ∀ (N : LB X) (hN : N.IsLB), cls N hN = cls M hM + n • cls L h.isLB → ∃ t : N.Sec, Mono (cutHom F hN t)

theorem **IsAmpleLB.eventually_exists_mono_cutHom_pow** (h : IsAmpleLB L) (F : X.Modules) [IsFiniteQC F] : ∀ᶠ n : ℕ in atTop, ∃ t : (lbPow L n).Sec, Mono (cutHom F (isLB_lbPow h.isLB n) t)

theorem **IsAmpleLB.exists_sub_of_mono_cutHom** (h : IsAmpleLB L) (F : X.Modules) [IsFiniteQC F] (c : Pic X) : ∃ (A B : LB X) (hA : A.IsLB) (hB : B.IsLB) (s : A.Sec) (t : B.Sec), Mono (cutHom F hA s) ∧ Mono (cutHom F hB t) ∧ c = cls A hA − cls B hB

theorem **cut_by_regular_section** {X : Scheme.{0}} [AlgebraicGeometry.IsNoetherian X] (F : X.Modules) [IsFiniteQC F] {L : LB X} (hL : L.IsLB) : (∀ s : L.Sec, (∀ (i : L.ι) (V : X.Opens) (h : V ≤ L.U i) (t : Γ(twistZ F L (-1), V)), (cutHom F hL s).app V t = resG X h (s.f i) • zchartEquiv hL (-1) i h t) ∧ (Mono (cutHom F hL s) ↔ ∀ x ∈ assPoints F, x ∈ nonzeroLocus s) ∧ IsFiniteQC (cokernel (cutHom F hL s)) ∧ support (cokernel (cutHom F hL s)) = support F \ nonzeroLocus s ∧ ((secIdeal hL s).support : Set X) = (nonzeroLocus s : Set X)ᶜ ∧ (∀ (W : X.affineOpens) (i : L.ι) (h : W.1 ≤ L.U i), (secIdeal hL s).ideal W = Ideal.span {resG X h (s.f i)}) ∧ (∀ W : X.affineOpens, Function.Surjective ((cokernel.π (cutHom F hL s)).app W.1) ∧ LinearMap.ker (appLinear (cokernel.π (cutHom F hL s)) W.1) = (secIdeal hL s).ideal W • ⊤) ∧ Nonempty (cokernel (cutHom F hL s) ≅ (Scheme.Modules.pushforward (secIdeal hL

theorem **mono_cutHom_structureModule_iff** {X : Scheme.{0}} [IsIntegral X] [IsLocallyNoetherian X] {L : LB X} (hL : L.IsLB) (s : L.Sec) : Mono (cutHom (structureModule X) hL s) ↔ genericPoint X ∈ nonzeroLocus s

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.EventuallyExistsResMEqMulSecOfEqZpowSec

@[simp] theorem **zfun_zpowSec** (σ : LB.Sec (lbZPow L m)) (k : ℕ) (i : L.ι) : zfun (zpowSec σ k) i = zfun σ i ^ k

theorem **basicOpen_resG_zfun** (σ : LB.Sec (lbZPow L m)) (i : L.ι) {W : X.Opens} (h : W ≤ L.U i) : X.basicOpen (resG X h (zfun σ i)) = W ⊓ Out.nonzeroLocus σ

theorem **basicOpen_zfun** (σ : LB.Sec (lbZPow L m)) (i : L.ι) : X.basicOpen (zfun σ i) = L.U i ⊓ Out.nonzeroLocus σ

theorem **nonzeroLocus_eq_of_zfun** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) {V : X.Opens} (h : ∀ i, X.basicOpen (zfun σ i) = L.U i ⊓ V) : Out.nonzeroLocus σ = V

theorem **nonzeroLocus_zofSec** (hL : L.IsLB) (s : LB.Sec L) : Out.nonzeroLocus (zofSec s) = Out.nonzeroLocus s

theorem **nonzeroLocus_le_zpowSec** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) (k : ℕ) : Out.nonzeroLocus σ ≤ Out.nonzeroLocus (zpowSec σ k)

theorem **exists_finite_affine_chart_cover** (hL : L.IsLB) {W : X.Opens} (hW : IsCompact (W : Set X)) : ∃ (κ : Type) (_ : Finite κ) (T : κ → X.Opens) (c : κ → L.ι), (∀ a, IsAffineOpen (T a)) ∧ (∀ a, T a ≤ W) ∧ (∀ a, T a ≤ L.U (c a)) ∧ W ≤ iSup T

theorem **mulSecOfEq_zpowSec_app** (σ : LB.Sec (lbZPow L m)) {a b c : ℕ} (hc : c = a + b) {d e : ℤ} (h₁ : n + a * m = d) (h₂ : d + b * m = e) (h₃ : n + c * m = e) {V : X.Opens} (t : Γ(twistZ F L n, V)) : (mulSecOfEq F (zpowSec σ b) h₂).app V ((mulSecOfEq F (zpowSec σ a) h₁).app V t) = (mulSecOfEq F (zpowSec σ c) h₃).app V t

theorem **eventually_mulSecOfEq_zpowSec_eq_zero** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) {W V : X.Opens} (hW : IsCompact (W : Set X)) (h : V ≤ W) (hV : W ⊓ Out.nonzeroLocus σ ≤ V) (a : Γ(twistZ F L n, W)) (ha : resM _ h a = 0) : ∀ᶠ k : ℕ in atTop, ∀ (d : ℤ) (hd : n + k * m = d), (mulSecOfEq F (zpowSec σ k) hd).app W a = 0

theorem **eventually_exists_resM_eq_mulSecOfEq_zpowSec** (hL : L.IsLB) (σ : LB.Sec (lbZPow L m)) {W V : X.Opens} (hW : IsCompact (W : Set X)) (hW' : IsQuasiSeparated (W : Set X)) (h : V ≤ W) (hV : V = W ⊓ Out.nonzeroLocus σ) (b : Γ(twistZ F L n, V)) : ∀ᶠ k : ℕ in atTop, ∀ (d : ℤ) (hd : n + k * m = d), ∃ a : Γ(twistZ F L d, W), resM _ h a = (mulSecOfEq F (zpowSec σ k) hd).app V b

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.EventuallyExistsResMEqSecHomSecToPow

theorem **eval₂Hom_unit_mul** {R B : Type*} [CommRing R] [CommRing B] (c : R →+* B) (u : Bˣ) (b : ι → B) {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) : eval₂Hom c (fun j => u * b j) p = ((u ^ d : Bˣ) : B) * eval₂Hom c b p

@[simp] theorem **chartEval_C** (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (r : R) : chartEval R x i h (C r) = resG X le_top (algebraMap R Γ(X, ⊤) r)

@[simp] theorem **chartEval_X** (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (j : ι) : chartEval R x i h (MvPolynomial.X j) = resG X h ((x j).f i)

@[simp] theorem **resG_chartEval** (i : L.ι) {V W : X.Opens} (h : V ≤ L.U i) (h' : W ≤ V) (p : MvPolynomial ι R) : resG X h' (chartEval R x i h p) = chartEval R x i (h'.trans h) p

@[simp] theorem **zfun_formSec** {p : MvPolynomial ι R} {d : ℤ} (hp : p ∈ polyGrading ι R d) (i : L.ι) : zfun (formSec x hp) i = chartEval R x i le_rfl p

theorem **chartEval_xProd** (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (I : Finset ι) : chartEval R x i h (xProd R I) = ∏ j ∈ I, resG X h ((x j).f i)

@[simp] theorem **zfun_prodSec** (I : Finset ι) (i : L.ι) : zfun (prodSec x I) i = ∏ j ∈ I, (x j).f i

theorem **nonzeroLocus_prodSec** (hL : L.IsLB) (I : Finset ι) : Out.nonzeroLocus (prodSec x I) = ⊓ j ∈ I, Out.nonzeroLocus (x j)

theorem **gammaOp_commute** : OperatorsCommute (gammaOp R x F V)

@[simp] theorem **gammaRes_gammaOf** (h : W ≤ V) (a : sectionsOn (twistZ F L n) V) : gammaRes R x F h (gammaOf R x F V n a) = gammaOf R x F W n (sectionsOn.res R _ h a)

theorem **isGradedMap_gammaRes** (h : W ≤ V) : IsGradedMap (gammaGrading R x F V) (gammaGrading R x F W) (gammaRes R x F h)

theorem **gammaRes_gammaRes** (h : V ≤ U) (h' : W ≤ V) (y : GammaStar R x F U) : gammaRes R x F h' (gammaRes R x F h y) = gammaRes R x F (h'.trans h) y

@[simp] theorem **gammaMap_gammaOf** (φ : F ⟶ G) (a : sectionsOn (twistZ F L n) V) : gammaMap R x φ V (gammaOf R x F V n a) = gammaOf R x G V n (sectionsOn.map R (map (lbZPow L n) φ) V a)

theorem **gammaCoord_gammaOf** (i : L.ι) (a : sectionsOn (twistZ F L n) V) : sectionsOn.toΓ F _ (gammaCoord R x F V i (gammaOf R x F V n a)) = zcoord (sectionsOn.toΓ _ V a) i

theorem **gammaCoord_smul** (i : L.ι) (p : MvPolynomial ι R) (y : GammaStar R x F V) : sectionsOn.toΓ F _ (gammaCoord R x F V i (p • y)) = chartEval R x i (inf_le_right : V ⊓ L.U i ≤ L.U i) p • sectionsOn.toΓ F _ (gammaCoord R x F V i y)

theorem **smul_gammaOf** {p : MvPolynomial ι R} {e e' d : ℤ} (hp : p ∈ polyGrading ι R e) (σ : LB.Sec (lbZPow L e')) (hσ : ∀ i, zfun σ i = chartEval R x i le_rfl p) (he : e = e') (h : n + e' = d) (a : sectionsOn (twistZ F L n) V) : p • gammaOf R x F V n a = gammaOf R x F V d (sectionsOn.map R (mulSecOfEq F σ h) V a)

theorem **xProd_pow_smul_gammaOf** (I : Finset ι) (k : ℕ) {d : ℤ} (h : n + k ∗ I.card = d) (a : sectionsOn (twistZ F L n) V) : xProd R I ^ k • gammaOf R x F V n a = gammaOf R x F V d (sectionsOn.map R (mulSecOfEq F (zpowSec (prodSec x I) k) h) V a)

theorem **gammaOp_bijective** (hL : L.IsLB) {j : ι} (hV : V ≤ Out.nonzeroLocus (x j)) (n : ℤ) : Function.Bijective (gammaOp R x F V j n)

theorem **xProd_smul_gammaStar_bijective** (hL : L.IsLB) {I : Finset ι} (hV : V ≤ ⊓ j ∈ I, Out.nonzeroLocus (x j)) : Function.Bijective fun y : GammaStar R x F V => xProd R I • y

theorem **eventually_xProd_pow_smul_eq_zero** {I : Finset ι} (h : V ≤ W) (hV : W ⊓ ⊓ j ∈ I, Out.nonzeroLocus (x j) ≤ V) (y : GammaStar R x F W) (hy : gammaRes R x F h y = 0) : ∀ᶠ k : ℕ in atTop, xProd R I ^ k • y = 0

theorem **eventually_exists_gammaRes_eq** {I : Finset ι} (h : V ≤ W) (hV : V = W ⊓ ⊓ j ∈ I, Out.nonzeroLocus (x j)) (z : GammaStar R x F V) : ∀ᶠ k : ℕ in atTop, ∃ y, gammaRes R x F h y = xProd R I ^ k • z

theorem **xProd_pow_smul_gammaLocEquiv** {I : Finset ι} (h : V ≤ W) (hV : V = W ⊓ ⊓ j ∈ I, Out.nonzeroLocus (x j)) (z : homLoc (gammaGrading R x F W) I n) (k : ℕ) (y : GammaStar R x F W) (e : awayFrac _ (xProd R I) k y = z) : xProd R I ^ k • gammaOf R x F V n (gammaLocEquiv R x F hL hW hW' h hV n z) = gammaRes R x F h y

theorem **gammaLocEquiv_eq_iff** {I : Finset ι} (h : V ≤ W) (hV : V = W ⊓ ⊓ j ∈ I, Out.nonzeroLocus (x j)) (z : homLoc (gammaGrading R x F W) I n) (k : ℕ) (y : GammaStar R x F W) (e : awayFrac _ (xProd R I) k y = z) (b : sectionsOn (twistZ F L n) V) : gammaLocEquiv R x F hL hW hW' h hV n z = b ↔ xProd R I ^ k • gammaOf R x F V n b = gammaRes R x F h y

theorem **gammaLocEquiv_homLocRes** {I J : Finset ι} (hIJ : I ⊆ J) (h : V ≤ W) (hV : V = W ⊓ ⊓ j ∈ I, Out.nonzeroLocus (x j)) (h' : U ≤ V) (hU : U = W ⊓ ⊓ j ∈ J, Out.nonzeroLocus (x j)) (z : homLoc (gammaGrading R x F W) I n) : gammaLocEquiv R x F hL hW hW' (h'.trans h) hU n (homLocRes _ hIJ n z) = sectionsOn.res R _ h' (gammaLocEquiv R x F hL hW hW' h hV n z)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.EventuallySubsingletonCohTwistZOfIsProjOver

theorem **mem_submodulePowSat** {g : A} {K : Submodule A M} {m : M} : m ∈ submodulePowSat g K ↔ ∃ k : ℕ, g ^ k • m ∈ K

theorem **le_submodulePowSat** (g : A) (K : Submodule A M) : K ≤ submodulePowSat g K

theorem **mem_submodulePowSat_of_pow_smul_mem** {g : A} {K : Submodule A M} {m : M} {l : ℕ} (h : g ^ l • m ∈ submodulePowSat g K) : m ∈ submodulePowSat g K

theorem **exists_fg_submodulePowSat_eq_top** {g : A} (f : M →ₗ[A] N) (h₁ : ∀ m, f m = 0 → ∃ k : ℕ, g ^ k • m = 0) (h₂ : ∀ z, ∃ (k : ℕ) (m : M), f m = g ^ k • z) {K : Submodule A N} (hK : K.FG) (h : submodulePowSat g K = ⊤) : ∃ K' : Submodule A M, K'.FG ∧ submodulePowSat g K' = ⊤

theorem **exists_fg_isHomogeneous_ge** {ι R : Type*} [CommRing R] {M : Type*} [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] [PolyGradedSMul ι 𝒜] {K : Submodule (MvPolynomial ι R) M} (hK : K.FG) : ∃ K' : Submodule (MvPolynomial ι R) M, K'.FG ∧ K'.IsHomogeneous 𝒜 ∧ K ≤ K'

@[simp] theorem **ratioEval_C** (i : ι) (r : R) : ratioEval R x i (C r) = resG X le_top (algebraMap R Γ(X, ⊤) r)

@[simp] theorem **ratioEval_X** (i j : ι) : ratioEval R x i (MvPolynomial.X j) = secRatio (x i) (x j)

theorem **finite_ratioEval_iff** (i : ι) : (ratioEval R x i).Finite ↔ Module.Finite (ratioEval R x i).range Γ(X, nonzeroLocus (x i))

theorem **X_smul_gammaOf_secRatio_smul** (i j : ι) (n : ℤ) (b : Γ(twistZ F L n, nonzeroLocus (x i))) : (MvPolynomial.X i : MvPolynomial ι R) • gammaOf R x F _ n ((sectionsOn.toΓ _ _).symm (secRatio (x i) (x j) • b)) = (MvPolynomial.X j : MvPolynomial ι R) • gammaOf R x F _ n ((sectionsOn.toΓ _ _).symm b)

theorem **exists_fg_submodulePowSat_gammaStar_locus** [QCoh.IsFiniteQC F] (hL : L.IsLB) {i : ι} (ha : IsAffineOpen (nonzeroLocus (x i))) (hf : (ratioEval R x i).Finite) : ∃ K : Submodule (MvPolynomial ι R) (GammaStar R x F (nonzeroLocus (x i))), K.FG ∧ submodulePowSat (MvPolynomial.X i) K = ⊤

theorem **exists_fg_isHomogeneous_submodulePowSat** : ∃ M : Submodule (MvPolynomial ι R) (GammaStar R x F ⊤), M.FG ∧ M.IsHomogeneous (gammaGrading R x F ⊤) ∧ ∀ i, submodulePowSat (MvPolynomial.X i) M = ⊤

theorem **exists_fg_cech_iso_grCech** : ∃ M : Submodule (MvPolynomial ι R) (GammaStar R x F ⊤), M.FG ∧ M.IsHomogeneous (gammaGrading R x F ⊤) ∧ (∀ s : Finset ι, s.Nonempty → ∀ n : ℤ, Function.Bijective (homLocMap (isGradedMap_subtype (gammaGrading R x F ⊤) M) s n)) ∧ ∀ [LinearOrder ι] (n : ℤ), Nonempty (cech R (twistZ F L n) ⊤ (fun j => nonzeroLocus (x j)) ≅ grCech ι (grSub (gammaGrading R x F ⊤) M) n)

theorem **exists_fg_coh_equiv_grH** [LinearOrder ι] : ∃ M : Submodule (MvPolynomial ι R) (GammaStar R x F ⊤), M.FG ∧ M.IsHomogeneous (gammaGrading R x F ⊤) ∧ ∀ (n : ℤ) (p : ℕ), Nonempty (grH ι (grSub (gammaGrading R x F ⊤) M) n p ≃ₗ[R] coh (twistZ F L n) p)

theorem **finite_coh_twistZ** (n : ℤ) (p : ℕ) : Module.Finite R (coh (twistZ F L n) p)

theorem **eventually_subsingleton_coh_twistZ** : ∀ᶠ n : ℤ in atTop, ∀ p, 0 < p → Subsingleton (coh (twistZ F L n) p)

theorem **finite_coh** (p : ℕ) : Module.Finite R (coh F p)

theorem **isFinite_iff_pullRatio** (φ : X ⟶ PSpace R N) {U : Fin (N + 1) → X.Opens} (hU : ∀ i, φ ⁻¹ᵁ chart R N i = U i) : IsFinite φ ↔ ∀ i, IsAffineOpen (U i) ∧ RingHom.Finite (eval₂Hom (baseRingHom (φ ≫ PSpace.str R N) (U i)).hom (pullRatio φ (U i) i (hU i).ge))

theorem **isAffineHom_of_isAffineOpen_preimage_chart** (φ : X ⟶ PSpace R N) (h : ∀ i, IsAffineOpen (φ ⁻¹ᵁ chart R N i)) : IsAffineHom φ

theorem **isFinite_toProj_of_isProper** [IsProper f] (ha : ∀ i, IsAffineOpen (nonzeroLocus (s i))) : IsFinite (toProj f s hs)

theorem **isFinite_toProj_iff** : IsFinite (toProj f s hs) ↔ ∀ i, IsAffineOpen (nonzeroLocus (s i)) ∧ (Twist.ratioEval R s i).Finite

theorem **ProjSpace.secPull_coord_of_isFinite** {N : ℕ} (φ : X ⟶ PSpace R N) [IsFinite φ] (hφ : φ ≫ PSpace.str R N = f) : ⊔ i, nonzeroLocus (secPull (coord R N i) φ) = ⊤ ∧ ∀ i, IsAffineOpen (nonzeroLocus (secPull (coord R N i) φ)) ∧ (ratioEval R (fun j => secPull (coord R N j) φ) i).Finite

theorem **Twist.finite_ratioEval_of_isProper** [IsProper f] {L : LB X} {N : ℕ} (x : Fin (N + 1) → L.Sec) (hc : ⊔ i, nonzeroLocus (x i) = ⊤) (ha : ∀ i, IsAffineOpen (nonzeroLocus (x i))) (i : Fin (N + 1)) : (ratioEval R x i).Finite

theorem **serre_of_isFinite** {N : ℕ} (φ : X ⟶ PSpace R N) [IsFinite φ] (hφ : φ ≫ PSpace.str R N = f) : (∀ (n : ℤ) (p : ℕ), Module.Finite R (coh (twistZ F ((O1 R N).pull φ) n) p)) ∧ ∀ᶠ n : ℤ in atTop, ∀ p, 0 < p → Subsingleton (coh (twistZ F ((O1 R N).pull φ) n) p)

theorem **IsAmpleLB.finite_coh** [IsProper f] {L : LB X} (h : IsAmpleLB L) (p : ℕ) : Module.Finite R (coh F p)

theorem **IsAmpleLB.eventually_subsingleton_coh_twistZ** [IsProper f] {L : LB X} (h : IsAmpleLB L) : ∀ᶠ n : ℤ in atTop, ∀ p, 0 < p → Subsingleton (coh (twistZ F L n) p)

theorem **IsAmpleLB.eventually_subsingleton_coh_twist** [IsProper f] {L : LB X} (h : IsAmpleLB L) : ∀ᶠ n : ℕ in atTop, ∀ p, 0 < p → Subsingleton (coh (twist F (lbPow L n)) p)

theorem **isProper_and_exists_isAmpleLB_of_isProjOver** (h : IsProjOver f) : IsProper f ∧ ∃ L : LB X, IsAmpleLB L

theorem **finite_coh_of_isProjOver** (h : IsProjOver f) (p : ℕ) : Module.Finite R (coh F p)

theorem **finite_coh_twist_of_isProjOver** (h : IsProjOver f) {M : LB X} (hM : M.IsLB) (p : ℕ) : Module.Finite R (coh (twist F M) p)

theorem **eventually_subsingleton_coh_twistZ_of_isProjOver** (h : IsProjOver f) {L : LB X} (hL : IsAmpleLB L) : ∀ᶠ n : ℤ in atTop, ∀ p, 0 < p → Subsingleton (coh (twistZ F L n) p)

theorem **IsAmpleLB.finiteCoh** [IsProper f] {L : LB X} (h : IsAmpleLB L) : FiniteCoh k F

theorem **finiteCoh_of_isProjOver** (h : IsProjOver f) : FiniteCoh k F

theorem **finiteCohTwists_of_isProjOver** (h : IsProjOver f) : FiniteCohTwists k F

theorem **serre_finiteness_vanishing** (R : Type) [CommRing R] [IsNoetherianRing R] (X : Scheme.{0}) [Algebra R Γ(X, ⊤)] (F : X.Modules) [IsFiniteQC F] : (∀ (L : LB X) (ι : Type) [Finite ι] (x : ι → L.Sec), L.IsLB → ⊔ i, nonzeroLocus (x i) = ⊤ → (∀ i, IsAffineOpen (nonzeroLocus (x i))) → (∀ i, Module.Finite (ratioEval R x i).range Γ(X, nonzeroLocus (x i))) → (∃ M : Submodule (MvPolynomial ι R) (GammaStar R x F ⊤), M.FG ∧ M.IsHomogeneous (gammaGrading R x F ⊤) ∧ (∀ s : Finset ι, s.Nonempty → ∀ n : ℤ, Function.Bijective (homLocMap (isGradedMap_subtype (gammaGrading R x F ⊤) M) s n)) ∧ ∀ [LinearOrder ι] (n : ℤ), Nonempty (cech R (twistZ F L n) ⊤ (fun j => nonzeroLocus (x j)) ≅ grCech ι (grSub (gammaGrading R x F ⊤) M) n)) ∧ (∀ (n : ℤ) (p : ℕ), Module.Finite R (coh (twistZ F L n) p)) ∧ ∀ᶠ n : ℤ in atTop, ∀ p, 0 < p → Subsingleton (coh (twistZ F L n) p)) ∧ ∀ f : X ⟶ Sp R, (∀ r, algebraMap R Γ(X,

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExactAndKerBaseChangeOfForallResidueField

theorem **exact_lTensor_iff_of_faithfullyFlat** (k L : Type*) [CommRing k] [CommRing L] [Algebra R k] [Algebra R L] [Algebra k L] [IsScalarTower R k L] [Module.FaithfullyFlat k L] (f : A →ₗ[R] B) (g : B →ₗ[R] C) : Exact (f.lTensor L) (g.lTensor L) ↔ Exact (f.lTensor k) (g.lTensor k)

theorem **exact_lTensor_iff_residueField** (p : Ideal R) [p.IsPrime] (L : Type*) [Field L] [Algebra R L] (hL : RingHom.ker (algebraMap R L) = p) (f : A →ₗ[R] B) (g : B →ₗ[R] C) : Exact (f.lTensor L) (g.lTensor L) ↔ Exact (f.lTensor p.ResidueField) (g.lTensor p.ResidueField)

theorem **split_injective_of_forall_lTensor_residueField_injective** [Module.Finite R B] [Module.Finite R C] [Module.Projective R C] (l : B →ₗ[R] C) (h : ∀ (m : Ideal R) [m.IsMaximal], Injective (l.lTensor m.ResidueField)) : ∃ r : C →ₗ[R] B, r ∘ₗ l = LinearMap.id

theorem **lTensor_liftQ_injective_of_exact** (N : Type*) [AddCommGroup N] [Module R N] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (hfg : LinearMap.range f ≤ LinearMap.ker g) (h : Exact (f.lTensor N) (g.lTensor N)) : Injective (((LinearMap.range f).liftQ g hfg).lTensor N)

theorem **exists_homotopy_of_forall_exact_residueField** [Module.Finite R B] [Module.Projective R B] [Module.Finite R C] [Module.Projective R C] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (hfg : g ∘ₗ f = 0) (h : ∀ (m : Ideal R) [m.IsMaximal], Exact (f.lTensor m.ResidueField) (g.lTensor m.ResidueField)) : ∃ (s : B →ₗ[R] A) (t : C →ₗ[R] B), f ∘ₗ s + t ∘ₗ g = LinearMap.id

theorem **forall_exact_residueField_of_isLocalRing** [IsLocalRing R] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Exact (f.lTensor (IsLocalRing.ResidueField R)) (g.lTensor (IsLocalRing.ResidueField R))) (m : Ideal R) [m.IsMaximal] : Exact (f.lTensor m.ResidueField) (g.lTensor m.ResidueField)

theorem **exact_of_homotopy** {f : A →ₗ[R] B} {g : B →ₗ[R] C} (hfg : g ∘ₗ f = 0) {s : B →ₗ[R] A} {t : C →ₗ[R] B} (h : f ∘ₗ s + t ∘ₗ g = LinearMap.id) : Exact f g

theorem **shortCyclesComparison_iCycles** (S : ShortComplex C) [S.HasHomology] [(S.map F).HasHomology] : shortCyclesComparison F S ≫ (S.map F).iCycles = F.map S.iCycles

theorem **map_toCycles_shortCyclesComparison** (S : ShortComplex C) [S.HasHomology] [(S.map F).HasHomology] : F.map S.toCycles ≫ shortCyclesComparison F S = (S.map F).toCycles

theorem **map_homologyπ_shortHomologyComparison** (S : ShortComplex C) [S.HasHomology] [(S.map F).HasHomology] : F.map S.homologyπ ≫ shortHomologyComparison F S = shortCyclesComparison F S ≫ (S.map F).homologyπ

theorem **epi_map_homologyπ** (S : ShortComplex C) [S.HasHomology] : Epi (F.map S.homologyπ)

theorem **cyclesComparison_iCycles** (K : HomologicalComplex C c) (p : ι) : cyclesComparison F K p ≫ ((F.mapHomologicalComplex c).obj K).iCycles p = F.map (K.iCycles p)

theorem **map_homologyπ_homologyComparison** (K : HomologicalComplex C c) (p : ι) : F.map (K.homologyπ p) ≫ homologyComparison F K p = cyclesComparison F K p ≫ ((F.mapHomologicalComplex c).obj K).homologyπ p

theorem **isZero_homology_extendScalars_iff** (S : Type w) [CommRing S] [inst : Algebra R S] (K : HomologicalComplex (ModuleCat.{u} R) c) {p q r : ι} (hp : c.prev q = p) (hr : c.next q = r) : IsZero ((((ModuleCat.extendScalars.{u, w, u} (algebraMap R S)).mapHomologicalComplex c).obj K).homology q) ↔ Exact ((K.d p q).hom.lTensor S) ((K.d q r).hom.lTensor S)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExactAndSurjectiveAppOfExactOfEpi

theorem **appLinear_apply** (φ : M ⟶ N) (U : X.Opens) (s : Γ(M, U)) : appLinear φ U s = φ.app U s

theorem **coe_appLinear** (φ : M ⟶ N) (U : X.Opens) : ⇑(appLinear φ U) = φ.app U

theorem **appLinear_comp** (φ : M ⟶ N) (ψ : N ⟶ P) (U : X.Opens) : appLinear (φ ≫ ψ) U = appLinear ψ U ∘ₗ appLinear φ U

theorem **appLinear_id** (M : X.Modules) (U : X.Opens) : appLinear (𝟙 M) U = LinearMap.id

theorem **appLinear_zero** (M N : X.Modules) (U : X.Opens) : appLinear (0 : M ⟶ N) U = 0

theorem **appLinear_add** (φ ψ : M ⟶ N) (U : X.Opens) : appLinear (φ + ψ) U = appLinear φ U + appLinear ψ U

theorem **app_app** (φ : M ⟶ N) (ψ : N ⟶ P) (U : X.Opens) (s : Γ(M, U)) : ψ.app U (φ.app U s) = (φ ≫ ψ).app U s

theorem **app_res** (φ : M ⟶ N) (h : V ≤ U) (s : Γ(M, U)) : φ.app V (M.presheaf.map (homOfLE h).op s) = N.presheaf.map (homOfLE h).op (φ.app U s)

theorem **sections_obj** (U : X.Opens) (M : X.Modules) : (sections U).obj M = ModuleCat.of Γ(X, U) Γ(M, U)

theorem **sections_map** (U : X.Opens) (φ : M ⟶ N) : (sections U).map φ = ModuleCat.ofHom (appLinear φ U)

theorem **injective_app_of_mono** (φ : M ⟶ N) [Mono φ] (U : X.Opens) : Function.Injective (φ.app U)

theorem **exact_app_of_exact_of_mono** {S : ShortComplex X.Modules} (hS : S.Exact) [Mono S.f] (U : X.Opens) : Function.Exact (S.f.app U) (S.g.app U)

theorem **isQuasicoherent_of_iso** (e : M ≅ N) [M.IsQuasicoherent] : N.IsQuasicoherent

theorem **isQuasicoherent_of_isColimit_Spec** {R : CommRingCat.{u}} [HasColimitsOfShape J (ModuleCat.{u} R)] {D : J ⥤ (Spec R).Modules} [∀ j, (D.obj j).IsQuasicoherent] {c : Cocone D} (hc : IsColimit c) : c.pt.IsQuasicoherent

theorem **preservesColimit_moduleSpecΓFunctor** {R : CommRingCat.{u}} [HasColimitsOfShape J (ModuleCat.{u} R)] (D : J ⥤ (Spec R).Modules) [∀ j, (D.obj j).IsQuasicoherent] : PreservesColimit D (moduleSpecΓFunctor (R

theorem **isQuasicoherent_of_isColimit** {D : J ⥤ X.Modules} [∀ j, (D.obj j).IsQuasicoherent] {c : Cocone D} (hc : IsColimit c) : c.pt.IsQuasicoherent

theorem **preservesColimit_sections** (hU : IsAffineOpen U) (D : J ⥤ X.Modules) [∀ j, (D.obj j).IsQuasicoherent] : PreservesColimit D (sections U)

theorem **isQuasicoherent_of_isZero** (h : IsZero M) : M.IsQuasicoherent

theorem **isQuasicoherent_of_exact_app** {K : X.Modules} (ι : K ⟶ M) (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hι : ∀ U, IsAffineOpen U → Function.Injective (ι.app U)) (hex : ∀ U, IsAffineOpen U → Function.Exact (ι.app U) (φ.app U)) : K.IsQuasicoherent

theorem **isQuasicoherent_X₁_of_exact** {S : ShortComplex X.Modules} (hS : S.Exact) [Mono S.f] [S.X₂.IsQuasicoherent] [S.X₃.IsQuasicoherent] : S.X₁.IsQuasicoherent

theorem **isQuasicoherent_X₃_of_exact** {S : ShortComplex X.Modules} (hS : S.Exact) [Epi S.g] [S.X₁.IsQuasicoherent] [S.X₂.IsQuasicoherent] : S.X₃.IsQuasicoherent

theorem **injective_kernel_ι_app** (φ : M ⟶ N) (U : X.Opens) : Function.Injective ((kernel.ι φ).app U)

theorem **exact_kernel_ι_app** (φ : M ⟶ N) (U : X.Opens) : Function.Exact ((kernel.ι φ).app U) (φ.app U)

theorem **isQuasicoherent_of_epi_of_mono** {I : X.Modules} (e : M ⟶ I) (m : I ⟶ N) [Epi e] [Mono m] [M.IsQuasicoherent] [N.IsQuasicoherent] : I.IsQuasicoherent

theorem **exact_and_surjective_app_of_exact_of_epi** {S : ShortComplex X.Modules} (hS : S.Exact) [Epi S.g] [S.X₁.IsQuasicoherent] [S.X₂.IsQuasicoherent] (hU : IsAffineOpen U) : Function.Exact (S.f.app U) (S.g.app U) ∧ Function.Surjective (S.g.app U)

theorem **exact_cokernel_π_app** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) : Function.Exact (φ.app U) ((cokernel.π φ).app U)

theorem **surjective_cokernel_π_app** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) : Function.Surjective ((cokernel.π φ).app U)

theorem **subsingleton_sections_of_isZero** (h : IsZero M) (U : X.Opens) : Subsingleton Γ(M, U)

theorem **surjective_app_of_epi** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] [Epi φ] (hU : IsAffineOpen U) : Function.Surjective (φ.app U)

theorem **exact_app_iff_surjective_lift** (S : ShortComplex X.Modules) (U : X.Opens) : Function.Exact (S.f.app U) (S.g.app U) ↔ Function.Surjective ((kernel.lift S.g S.f S.zero).app U)

theorem **exact_app_of_exact** {S : ShortComplex X.Modules} (hS : S.Exact) [S.X₁.IsQuasicoherent] [S.X₂.IsQuasicoherent] [S.X₃.IsQuasicoherent] (hU : IsAffineOpen U) : Function.Exact (S.f.app U) (S.g.app U)

theorem **injective_image_ι_app** (φ : M ⟶ N) (U : X.Opens) : Function.Injective ((image.ι φ).app U)

theorem **surjective_factorThruImage_app** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) : Function.Surjective ((factorThruImage φ).app U)

theorem **range_image_ι_app** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) : LinearMap.range (appLinear (image.ι φ) U) = LinearMap.range (appLinear φ U)

theorem **isZero_iff_affineCover** [M.IsQuasicoherent] : IsZero M ↔ ∀ i, Subsingleton Γ(M, 𝒰 i)

theorem **mono_iff_affineCover** (φ : M ⟶ N) : Mono φ ↔ ∀ i, Function.Injective (φ.app (𝒰 i))

theorem **epi_iff_affineCover** (φ : M ⟶ N) : Epi φ ↔ ∀ i, Function.Surjective (φ.app (𝒰 i))

theorem **isIso_iff_affineCover** (φ : M ⟶ N) : IsIso φ ↔ ∀ i, Function.Bijective (φ.app (𝒰 i))

theorem **exact_iff_affineCover** : S.Exact ↔ ∀ i, Function.Exact (S.f.app (𝒰 i)) (S.g.app (𝒰 i))

theorem **shortExact_iff_affineCover** : S.ShortExact ↔ ∀ i, Function.Injective (S.f.app (𝒰 i)) ∧ Function.Exact (S.f.app (𝒰 i)) (S.g.app (𝒰 i)) ∧ Function.Surjective (S.g.app (𝒰 i))

theorem **isQuasicoherent_of_isLimit** {J : Type} [SmallCategory J] [FinCategory J] {D : J ⥤ X.Modules} [h : ∀ j, (D.obj j).IsQuasicoherent] {c : Cone D} (hc : IsLimit c) : c.pt.IsQuasicoherent

theorem **range_kernel_ι_app** (φ : M ⟶ N) (U : X.Opens) : LinearMap.range (appLinear (kernel.ι φ) U) = LinearMap.ker (appLinear φ U)

theorem **ker_cokernel_π_app** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) : LinearMap.ker (appLinear (cokernel.π φ) U) = LinearMap.range (appLinear φ U)

theorem **kernelSectionsEquiv_apply_coe** (φ : M ⟶ N) (U : X.Opens) (s : Γ(kernel φ, U)) : (kernelSectionsEquiv φ U s : Γ(M, U)) = (kernel.ι φ).app U s

theorem **cokernelSectionsEquiv_mk** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) (s : Γ(N, U)) : cokernelSectionsEquiv φ hU (Submodule.Quotient.mk s) = (cokernel.π φ).app U s

theorem **imageSectionsEquiv_apply_coe** (φ : M ⟶ N) [M.IsQuasicoherent] [N.IsQuasicoherent] (hU : IsAffineOpen U) (s : Γ(image φ, U)) : (imageSectionsEquiv φ hU s : Γ(N, U)) = (image.ι φ).app U s

theorem **sigmaSectionsEquiv_lof** {ι : Type u} [DecidableEq ι] (F : ι → X.Modules) [∀ i, (F i).IsQuasicoherent] (hU : IsAffineOpen U) (i : ι) (s : Γ(F i, U)) : sigmaSectionsEquiv F hU (DirectSum.lof Γ(X, U) ι (fun i ↦ Γ(F i, U)) i s) = (Sigma.ι F i).app U s

theorem **piSectionsEquiv_apply** {ι : Type u} (F : ι → X.Modules) (U : X.Opens) (s : Γ(∏ᶜ F, U)) (i : ι) : piSectionsEquiv F U s i = (Pi.π F i).app U s

theorem **app_sum** {ι : Type w} (t : Finset ι) (φ : ι → (M ⟶ N)) (U : X.Opens) (s : Γ(M, U)) : (∑ i ∈ t, φ i).app U s = ∑ i ∈ t, (φ i).app U s

theorem **biproductSectionsEquiv_apply** {ι : Type} [Fintype ι] (F : ι → X.Modules) [HasBiproduct F] (U : X.Opens) (s : Γ(⊕ F, U)) (i : ι) : biproductSectionsEquiv F U s i = (biproduct.π F i).app U s

theorem **biproductSectionsEquiv_symm_apply** {ι : Type} [Fintype ι] (F : ι → X.Modules) [HasBiproduct F] (U : X.Opens) (s : ∀ i, Γ(F i, U)) : (biproductSectionsEquiv F U).symm s = ∑ i, (biproduct.ι F i).app U (s i)

theorem **biprodSectionsEquiv_apply** (M N : X.Modules) (U : X.Opens) (s : Γ(M ⊞ N, U)) : biprodSectionsEquiv M N U s = ((biprod.fst : M ⊞ N ⟶ M).app U s, (biprod.snd : M ⊞ N ⟶ N).app U s)

theorem **isZero_iff_forall_affine** : IsZero M ↔ ∀ U, IsAffineOpen U → Subsingleton Γ(M, U)

theorem **epi_iff_forall_affine** (φ : M ⟶ N) : Epi φ ↔ ∀ U, IsAffineOpen U → Function.Surjective (φ.app U)

theorem **isIso_iff_forall_affine** (φ : M ⟶ N) : IsIso φ ↔ ∀ U, IsAffineOpen U → Function.Bijective (φ.app U)

theorem **exact_iff_forall_affine** : S.Exact ↔ ∀ U, IsAffineOpen U → Function.Exact (S.f.app U) (S.g.app U)

theorem **hom_eq_zero_iff_affineCover** {ι : Type*} (𝒰 : ι → X.Opens) (hcov : IsOpenCover 𝒰) (haff : ∀ i, IsAffineOpen (𝒰 i)) [M.IsQuasicoherent] [N.IsQuasicoherent] (φ : M ⟶ N) : φ = 0 ↔ ∀ i s, φ.app (𝒰 i) s = 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExactCechOneOfH1Zero

theorem **exact_cech_one_of_h1Zero** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), M.IsLB → ⊔ i, D i = ⊤ → FirstCohomologyVanishes M → Function.Exact (Cube.d (secCube f M D) 1) (Cube.d (secCube f M D) 2)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsAddEquivOfSurjectiveOfKerIff

theorem **exists_addEquiv_of_surjective_of_ker_iff** {A : Type u_1} {B : Type u_2} {C : Type u_3} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] (f : A →+ B) (g : A →+ C) (hf : Function.Surjective f) (hg : Function.Surjective g) (h : ∀ a : A, f a = 0 ↔ g a = 0) : ∃ e : B ≃+ C, ∀ a : A, e (f a) = g a

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsBasicOpenEqZeroOfMemIInfPowSmul

theorem **localizedModule_mk_one_eq_zero_iff** (p : Ideal A) [p.IsPrime] (x : M) : (LocalizedModule.mk x 1 : LocalizedModule p.primeCompl M) = 0 ↔ ∃ s ∉ p, s • x = 0

theorem **exists_smul_eq_zero_of_mem_iInf_pow_smul** (I : Ideal A) {x : M} (hx : x ∈ (⨅ n : ℕ, I ^ n • ⊤ : Submodule A M)) : ∃ f : A, f − 1 ∈ I ∧ f • x = 0

theorem **eq_zero_of_forall_associatedPrimes_exists_smul_eq_zero** {x : M} (h : ∀ p ∈ associatedPrimes A M, ∃ s ∉ p, s • x = 0) : x = 0

theorem **under_mem_associatedPrimes** [Module B M] [IsScalarTower A B M] {q : Ideal B} (hq : q ∈ associatedPrimes B M) : q.under A ∈ associatedPrimes A M

theorem **associatedPrimes_subset_of_flat** [IsNoetherianRing A] [Module.Flat A M] : associatedPrimes A M ⊆ associatedPrimes A A

theorem **under_eq_of_mem_associatedPrimes_tensor_quotient** (p : Ideal A) [hp : p.IsPrime] {q : Ideal B} (hq : q ∈ associatedPrimes B (M ⊗[A] (A ⧸ p))) : q.under A = p

theorem **mem_associatedPrimes_tensor_quotient_under** [IsNoetherianRing A] {N : Type*} [AddCommGroup N] [Module A N] [hN : Module.Finite A N] {q : Ideal B} (hq : q ∈ associatedPrimes B (M ⊗[A] N)) : q ∈ associatedPrimes B (M ⊗[A] (A ⧸ q.under A))

theorem **injective_lTensor_residueField** (p : Ideal A) [p.IsPrime] : Function.Injective (AlgebraTensorModule.lTensor B M (IsScalarTower.toAlgHom A (A ⧸ p) p.ResidueField).toLinearMap)

theorem **associatedPrimes_tensor_residueField** (p : Ideal A) [hp : p.IsPrime] : associatedPrimes B (M ⊗[A] p.ResidueField) = associatedPrimes B (M ⊗[A] (A ⧸ p))

theorem **mem_associatedPrimes_iff_of_flat_module** [IsNoetherianRing A] (q : Ideal B) [q.IsPrime] : q ∈ associatedPrimes B M ↔ q.under A ∈ associatedPrimes A A ∧ q ∈ associatedPrimes B (M ⊗[A] (q.under A).ResidueField)

theorem **mem_associatedPrimes_tensor_residueField_iff** (p : Ideal A) [p.IsPrime] (q : Ideal B) : q ∈ associatedPrimes B (B ⊗[A] p.ResidueField) ↔ ∃ Q ∈ associatedPrimes (p.Fiber B) (p.Fiber B), Q.comap (Algebra.TensorProduct.includeRight : B →ₐ[A] p.Fiber B) = q

theorem **mem_associatedPrimes_iff_of_flat** [IsNoetherianRing A] (q : Ideal B) [q.IsPrime] : q ∈ associatedPrimes B B ↔ q.under A ∈ associatedPrimes A A ∧ ∃ Q ∈ associatedPrimes ((q.under A).Fiber B) ((q.under A).Fiber B), Q.comap (Algebra.TensorProduct.includeRight : B →ₐ[A] (q.under A).Fiber B) = q

theorem **ker_includeRight_fiber** (p : Ideal A) [p.IsPrime] : RingHom.ker (Algebra.TensorProduct.includeRight : B →ₐ[A] p.Fiber B) = p.map (algebraMap A B)

theorem **mem_associatedPrimes_iff_eq_map_of_flat** [IsNoetherianRing A] (q : Ideal B) [hq : q.IsPrime] [NoZeroDivisors ((q.under A).Fiber B)] : q ∈ associatedPrimes B B ↔ q.under A ∈ associatedPrimes A A ∧ q = (q.under A).map (algebraMap A B)

theorem **mem_associatedPrimes_iff_generic_of_flat** [IsNoetherianRing A] (q : Ideal B) [hq : q.IsPrime] [NoZeroDivisors ((q.under A).Fiber B)] : q ∈ associatedPrimes B B ↔ q.under A ∈ associatedPrimes A A ∧ ∀ q' : Ideal B, q'.IsPrime → q'.under A = q.under A → q ≤ q'

theorem **eq_zero_of_flat_of_forall_fiber** [IsNoetherianRing A] [IsNoetherianRing B] {L : Type*} [AddCommGroup L] [Module B L] [Module.Flat B L] (hfib : ∀ p : PrimeSpectrum A, p.asIdeal ∈ associatedPrimes A A → NoZeroDivisors (p.asIdeal.Fiber B)) {x : L} (hx : ∀ q : PrimeSpectrum B, q.asIdeal.under A ∈ associatedPrimes A A → ∃ q' : PrimeSpectrum B, q'.asIdeal.under A = q.asIdeal.under A ∧ (LocalizedModule.mk x 1 : LocalizedModule q'.asIdeal.primeCompl L) = 0) : x = 0

theorem **exists_sup_maximalIdeal_pow_stable** [IsNoetherianRing R] [IsLocalRing R] (J : ℕ → Ideal R) (hJ : Antitone J) (k : ℕ) : ∃ m, ∀ m' ≥ m, J m' ⊔ maximalIdeal R ^ k = J m ⊔ maximalIdeal R ^ k

theorem **exists_le_maximalIdeal_pow_of_iInf_eq_bot** [IsNoetherianRing R] [IsLocalRing R] [IsPrecomplete (maximalIdeal R) R] (J : ℕ → Ideal R) (hJ : Antitone J) (h0 : ⊓ m, J m = ⊥) (k : ℕ) : ∃ m, J m ≤ maximalIdeal R ^ k

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsBasisKaehlerOfAlgebraicIndependent

theorem **alternating_smul_add_mem** (ω : M [⋀^ι]→ₗ[R] N) (P : Submodule R N) (θ : M) (a : ι → R) (w : ι → M) (h₀ : ω w ∈ P) (h₁ : ∀ i, ω (Function.update w i θ) ∈ P) : ω (fun i => a i • θ + w i) ∈ P

theorem **eq_lift_det_smul_ιMulti** {n : ℕ} (b : Basis (Fin n) R M) (η : ⋀[R]^n M) : η = exteriorPower.alternatingMapLinearEquiv b.det η • exteriorPower.ιMulti R n b

theorem **dWedge_mem_of_two_mem** (t : A) (x : Fin p → A) {i j : Fin p} (hij : i ≠ j) (hi : t ∣ x i) (hj : t ∣ x j) : ∃ η : ⋀[A]^p Ω[A⁄R], dWedge R x = t • η

@[simp] theorem **formMap_ιMulti** (v : Fin p → Ω[V⁄K]) : formMap K V F p (exteriorPower.ιMulti V p v) = exteriorPower.ιMulti F p (fun k => KaehlerDifferential.map K K V F (v k))

theorem **formMap_dWedge** (x : Fin p → V) : formMap K V F p (dWedge K x) = dWedge K (fun k => algebraMap V F (x k))

theorem **ιMulti_mem_regForms** (w : Fin p → Ω[F⁄K]) (hw : ∀ k, w k ∈ LinearMap.range (KaehlerDifferential.map K K V F)) : exteriorPower.ιMulti F p w ∈ regForms K V F p

theorem **dWedge_mem_regForms** (x : Fin p → V) : dWedge K (fun k => algebraMap V F (x k)) ∈ regForms K V F p

theorem **exists_dWedge_eq_smul_regForms** (t : V) (x : Fin p → V) {i j : Fin p} (hij : i ≠ j) (hi : t ∣ x i) (hj : t ∣ x j) : ∃ η ∈ regForms K V F p, dWedge K (fun k => algebraMap V F (x k)) = t • η

theorem **dlogWedge_eq_smul** (S : Finset (Fin p)) (x : Fin p → F) : dlogWedge K S x = (∏ k ∈ S, x k)⁻¹ • dWedge K x

@[simp] theorem **dlogWedge_empty** (x : Fin p → F) : dlogWedge K ∅ x = dWedge K x

theorem **dlog_mul** {x y : F} (hx : x ≠ 0) (hy : y ≠ 0) : (x * y)⁻¹ • D K F (x * y) = x⁻¹ • D K F x + y⁻¹ • D K F y

theorem **dlog_zpow** {x : F} (hx : x ≠ 0) (m : ℤ) : (x ^ m)⁻¹ • D K F (x ^ m) = m • x⁻¹ • D K F x

theorem **dlog_unit_mem_range** (u : V×) : (algebraMap V F u)⁻¹ • D K F (algebraMap V F u) ∈ LinearMap.range (KaehlerDifferential.map K K V F)

theorem **dlogWedge_mem_regForms** (S : Finset (Fin p)) (x : Fin p → V) (hS : ∀ k ∈ S, IsUnit (x k)) : dlogWedge K S (fun k => algebraMap V F (x k)) ∈ regForms K V F p

theorem **smul_dlogWedge_mem_regForms** (t : V) (ht : algebraMap V F t ≠ 0) (S : Finset (Fin p)) (x : Fin p → F) (hS : ∀ k ∈ S, ∃ (m : ℤ) (u : V×), x k = algebraMap V F u * algebraMap V F t ^ m) (hS' : ∀ k ∉ S, x k ∈ Set.range (algebraMap V F)) : t • dlogWedge K S x ∈ regForms K V F p

@[simp] theorem **ordTop_vw_zero** : ordTop_vw V (0 : F) = ⊤

theorem **ordTop_vw_of_ne_zero** {f : F} (hf : f ≠ 0) : ordTop_vw V f = (WithZero.log (Ring.ordFrac V f) : ℤ)

theorem **ordTop_vw_eq_top_iff** {f : F} : ordTop_vw V f = ⊤ ↔ f = 0

theorem **ordTop_vw_mul** (f g : F) : ordTop_vw V (f * g) = ordTop_vw V f + ordTop_vw V g

theorem **ordTop_vw_unit_mul_zpow** {t : V} (ht : Irreducible t) (u : V×) (m : ℤ) : ordTop_vw V (algebraMap V F u * algebraMap V F t ^ m) = m

theorem **exists_eq_unit_mul_zpow** {t : V} (ht : Irreducible t) {f : F} (hf : f ≠ 0) : ∃ (m : ℤ) (u : V×), f = algebraMap V F u * algebraMap V F t ^ m

theorem **le_ordTop_vw_iff** {t : V} (ht : Irreducible t) (m : ℤ) (f : F) : (m : WithTop ℤ) ≤ ordTop_vw V f ↔ ∃ a : V, f = algebraMap V F t ^ m * algebraMap V F a

@[simp] theorem **fracKaehlerBasis_apply** (b : Basis (Fin n) V Ω[V⁄K]) (k : Fin n) : fracKaehlerBasis F b k = KaehlerDifferential.map K K V F (b k)

theorem **topFormGen_eq** : topFormGen F b = exteriorPower.ιMulti F n (fracKaehlerBasis F b)

@[simp] theorem **topFormCoeff_topFormGen** : topFormCoeff F b (topFormGen F b) = 1

theorem **topFormCoeff_smul_topFormGen** (ω : ⋀[F]^n Ω[F⁄K]) : topFormCoeff F b ω • topFormGen F b = ω

theorem **topFormGen_ne_zero** : topFormGen F b ≠ 0

theorem **mem_regForms_iff** (ω : ⋀[F]^n Ω[F⁄K]) : ω ∈ regForms K V F n ↔ ∃ a : V, ω = a • topFormGen F b

theorem **topFormOrd_vw_smul_topFormGen** (f : F) : topFormOrd_vw F b (f • topFormGen F b) = ordTop_vw V f

theorem **topFormOrd_vw_smul** (f : F) (ω : ⋀[F]^n Ω[F⁄K]) : topFormOrd_vw F b (f • ω) = ordTop_vw V f + topFormOrd_vw F b ω

theorem **topFormOrd_vw_eq_top_iff** (ω : ⋀[F]^n Ω[F⁄K]) : topFormOrd_vw F b ω = ⊤ ↔ ω = 0

theorem **le_topFormOrd_vw_iff** {t : V} (ht : Irreducible t) (m : ℤ) (ω : ⋀[F]^n Ω[F⁄K]) : (m : WithTop ℤ) ≤ topFormOrd_vw F b ω ↔ ∃ η ∈ regForms K V F n, ω = algebraMap V F t ^ m • η

theorem **topFormOrd_vw_nonneg_iff** (ω : ⋀[F]^n Ω[F⁄K]) : 0 ≤ topFormOrd_vw F b ω ↔ ω ∈ regForms K V F n

theorem **topFormOrd_vw_eq** (b' : Basis (Fin n) V Ω[V⁄K]) (ω : ⋀[F]^n Ω[F⁄K]) : topFormOrd_vw F b ω = topFormOrd_vw F b' ω

theorem **topFormOrd_vw_topFormGen** (b' : Basis (Fin n) V Ω[V⁄K]) : topFormOrd_vw F b (topFormGen F b') = 0

theorem **topFormOrd_vw_dWedge_nonneg** (x : Fin n → V) : 0 ≤ topFormOrd_vw F b (dWedge K fun k => algebraMap V F (x k))

theorem **one_le_topFormOrd_vw_dWedge** (x : Fin n → V) {i j : Fin n} (hij : i ≠ j) (hi : x i ∈ maximalIdeal V) (hj : x j ∈ maximalIdeal V) : 1 ≤ topFormOrd_vw F b (dWedge K fun k => algebraMap V F (x k))

theorem **neg_one_le_topFormOrd_vw_dlogWedge** (S : Finset (Fin n)) (x : Fin n → F) (hS : ∀ k ∈ S, x k ≠ 0) (hS' : ∀ k ∉ S, x k ∈ Set.range (algebraMap V F)) : ((-1 : ℤ) : WithTop ℤ) ≤ topFormOrd_vw F b (dlogWedge K S x)

theorem **topFormOrd_vw_dlogWedge_nonneg** (S : Finset (Fin n)) (x : Fin n → V) (hS : ∀ k ∈ S, IsUnit (x k)) : 0 ≤ topFormOrd_vw F b (dlogWedge K S fun k => algebraMap V F (x k))

theorem **topFormOrd_vw_dWedge_basis** (z : Fin n → V) (c : Basis (Fin n) V Ω[V⁄K]) (hc : ∀ k, c k = D K V (z k)) : topFormOrd_vw F b (dWedge K fun k => algebraMap V F (z k)) = 0

theorem **topFormOrd_vw_dlogWedge_basis** (z : Fin n → V) (c : Basis (Fin n) V Ω[V⁄K]) (hc : ∀ k, c k = D K V (z k)) {i : Fin n} (hi : Irreducible (z i)) : topFormOrd_vw F b (dlogWedge K {i} fun k => algebraMap V F (z k)) = ((-1 : ℤ) : WithTop ℤ)

theorem **exists_basis_kaehler_of_algebraicIndependent** {K F ι : Type*} [Field K] [Field F] [Algebra K F] {v : ι → F} (hv : AlgebraicIndependent K v) [Algebra.IsSeparable (IntermediateField.adjoin K (Set.range v)) F] : ∃ b : Basis ι F Ω[F⁄K], ∀ i, b i = D K F (v i)

theorem **rank_kaehler_eq_trdeg** (K F : Type*) [Field K] [Field F] [Algebra K F] [PerfectField K] [Algebra.EssFiniteType K F] : Module.rank F Ω[F⁄K] = Algebra.trdeg K F

theorem **one_tmul_D_ne_zero** {t : V} (ht : Irreducible t) : (1 : ResidueField V) ⊗ₜ[V] D K V t ≠ 0

theorem **exists_basis_kaehler_uniformiser** {t : V} (ht : Irreducible t) : ∃ (m : ℕ) (z : Fin (m + 1) → V) (b : Basis (Fin (m + 1)) V Ω[V⁄K]), z 0 = t ∧ ∀ k, b k = D K V (z k)

theorem **trdeg_eq_of_basis_kaehler** {n : ℕ} (b : Basis (Fin n) V Ω[V⁄K]) : Algebra.trdeg K F = n

theorem **exists_basis_kaehler_of_trdeg** {n : ℕ} (hn : Algebra.trdeg K F = n) {t : V} (ht : Irreducible t) : ∃ (h : 0 < n) (z : Fin n → V) (b : Basis (Fin n) V Ω[V⁄K]), z ⟨0, h⟩ = t ∧ ∀ k, b k = D K V (z k)

theorem **trdeg_residueField_of_trdeg** {n : ℕ} (hn : Algebra.trdeg K F = n) : Algebra.trdeg K (ResidueField V) = (n - 1 : ℕ)

theorem **topFormOrd_vw_properties** (K V F : Type*) [Field K] [CharZero K] [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] [Algebra K V] [Algebra.EssFiniteType K V] [Algebra.FormallySmooth K V] [Field F] [Algebra V F] [IsFractionRing V F] [Algebra K F] [IsScalarTower K V F] {n : ℕ} (hn : Algebra.trdeg K F = n) : Algebra.trdeg K (ResidueField V) = (n - 1 : ℕ) ∧ (∀ t : V, Irreducible t → ∃ (h : 0 < n) (z : Fin n → V) (b : Basis (Fin n) V Ω[V⁄K]), z ⟨0, h⟩ = t ∧ ∀ k, b k = D K V (z k)) ∧ Nonempty (Basis (Fin n) V Ω[V⁄K]) ∧ ∀ b : Basis (Fin n) V Ω[V⁄K], (∀ (b' : Basis (Fin n) V Ω[V⁄K]) (ω : ⋀[F]^n Ω[F⁄K]), topFormOrd_vw F b ω = topFormOrd_vw F b' ω) ∧ (∀ ω : ⋀[F]^n Ω[F⁄K], ∃! f : F, ω = f • topFormGen F b) ∧ (∀ f : F, topFormOrd_vw F b (f • topFormGen F b) = ordTop_vw V f) ∧ (∀ x : Fin n → V, 0 ≤ topFormOrd_vw F b (dWedge K fun i => algebraMap V F (x i))) ∧ (∀ (x : Fin n → V) (i j : Fin n

theorem **exists_smooth_discreteValuationRing** : ∃ (V : Subalgebra K (RatFunc K)) (_ : IsDiscreteValuationRing V), Algebra.EssFiniteType K V ∧ Algebra.FormallySmooth K V ∧ IsFractionRing V (RatFunc K) ∧ Algebra.trdeg K (RatFunc K) = (1 : ℕ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsBasisMapOfLiftBaseChangeSurjective

theorem **_root_.GlobalTheory.Avoid.exists_basis_map_one_tmul_of_surjective** {K L V W ι : Type*} [CommRing K] [Field L] [Algebra K L] [AddCommGroup V] [Module K V] [AddCommGroup W] [Module L W] [Module.Finite L W] {d : ℕ} (hd : finrank L W = d) (G : L ⊗[K] V →ₗ[L] W) (hG : Surjective G) (v : ι → V) (hv : Submodule.span K (Set.range v) = ⊤) : ∃ (σ : Fin d → ι) (b : Basis (Fin d) L W), Injective σ ∧ ∀ k, b k = G (1 ⊗ₜ v (σ k))

theorem **pairingDual_ιMulti_comp** {R M N : Type*} [CommRing R] [AddCommGroup M] [Module R M] [AddCommGroup N] [Module R N] {r : ℕ} (φ : M →ₗ[R] N) (f : Fin r → Dual R N) (x : ⋀[R]^r M) : pairingDual R M r (ιMulti R r fun k => f k ∘ₗ φ) x = pairingDual R N r (ιMulti R r f) (exteriorPower.map r φ x)

theorem **exists_minor_ne_zero_of_separating** {K V ι : Type*} [Field K] [AddCommGroup V] [Module K V] [Finite ι] [LinearOrder ι] (c : ι → Dual K V) (hc : ∀ v : V, (∀ j, c j v = 0) → v = 0) {r : ℕ} (x : ⋀[K]^r V) (hx : x ≠ 0) : ∃ J : Set.powersetCard ι r, pairingDual K V r (ιMulti_family K r c J) x ≠ 0

theorem **exists_minor_ne_zero_of_span** {K V ι : Type*} [Field K] [AddCommGroup V] [Module K V] [Finite ι] [LinearOrder ι] (c : ι → Dual K V) (hc : Submodule.span K (Set.range c) = ⊤) {r : ℕ} (x : ⋀[K]^r V) (hx : x ≠ 0) : ∃ J : Set.powersetCard ι r, pairingDual K V r (ιMulti_family K r c J) x ≠ 0

theorem **exists_minor_ne_zero** {K V ι : Type*} [Field K] [AddCommGroup V] [Module K V] [Finite ι] [LinearOrder ι] (c : Basis ι K (Dual K V)) {r : ℕ} (x : ⋀[K]^r V) (hx : x ≠ 0) : ∃ J : Set.powersetCard ι r, pairingDual K V r (ιMulti_family K r c J) x ≠ 0

theorem **_root_.GlobalTheory.Avoid.exists_orderEmbedding_minor_ne_zero** {K V ι : Type*} [Field K] [AddCommGroup V] [Module K V] [Finite ι] [LinearOrder ι] (c : ι → Dual K V) (hc : Submodule.span K (Set.range c) = ⊤) {r : ℕ} (x : ⋀[K]^r V) (hx : x ≠ 0) : ∃ J : Fin r ↪o ι, pairingDual K V r (ιMulti K r fun k => c (J k)) x ≠ 0

theorem **minor_ιMulti** {R V ι : Type*} [CommRing R] [AddCommGroup V] [Module R V] [LinearOrder ι] (c : ι → Dual R V) {r : ℕ} (J : Set.powersetCard ι r) (v : Fin r → V) : pairingDual R V r (ιMulti_family R r c J) (ιMulti R r v) = (Matrix.of fun i j => c (Set.powersetCard.ofFinEmbEquiv.symm J j) (v i)).det

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsDehomogenisationTwistedChart

theorem **val_secUnit** (σ : LB.Sec H) (k : H.ι) {O : Y.Opens} (hk : O ≤ H.U k) (h : O ≤ nonzeroLocus σ) : (secUnit σ k hk h).val = rs Y hk (σ.f k)

theorem **rs_secUnit_zpow** (σ : LB.Sec H) (k : H.ι) {O O' : Y.Opens} (hk : O ≤ H.U k) (h : O ≤ nonzeroLocus σ) (e : O' ≤ O) (t : ℤ) : rs Y e (secUnit σ k hk h ^ t).val = (secUnit σ k (e.trans hk) (e.trans h) ^ t).val

theorem **secUnit_compat** (σ : LB.Sec H) (k l : H.ι) {O : Y.Opens} (hH : O ≤ H.U k ⊓ H.U l) (h : O ≤ nonzeroLocus σ) : secUnit σ k (hH.trans inf_le_left) h = res hH (H.g k l) * secUnit σ l (hH.trans inf_le_right) h

theorem **twCoord_divPow** (σ : LB.Sec H) (h : V ≤ nonzeroLocus σ) (t : ℤ) (x : SecOn (tw M H t) V) (p : M.ι) (k : H.ι) : twCoord (divPow M σ h t x) p k = (secUnit σ k (inf_le_right.trans inf_le_right) (inf_le_left.trans h) ^ (-t)).val * twCoord x p k

theorem **divPow_zero** (σ : LB.Sec H) (h : V ≤ nonzeroLocus σ) (x : SecOn (tw M H 0) V) : divPow M σ h 0 x = x

theorem **divPow_injective** (σ : LB.Sec H) (h : V ≤ nonzeroLocus σ) (t : ℤ) : Function.Injective (divPow M σ h t)

theorem **divPow_mulSec** (σ : LB.Sec H) (h : V ≤ nonzeroLocus σ) (t : ℤ) (x : SecOn (tw M H t) V) : divPow M σ h (t + 1) (mulSec M t σ V x) = divPow M σ h t x

theorem **dehom_of** (t : ℤ) (a : SecOnR f (tw M H t) V) : dehom f s hx i h r hr (GrSum.of hx t a) = SecOnR.lin f (divPow M (s i) h t) a

theorem **ratio_self** : r i = 1

theorem **exists_dehom_top** (hx : ∀ V : Y.Opens, OperatorsCommute (ι

theorem **exists_dehom_chart** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsDehomogenisationTwistedChartStatement

theorem **_root_.GlobalTheory.T.exists_dehomogenisation_twisted_chart** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsDualNeZeroCompEqZero

theorem **exists_dual_ne_zero_comp_eq_zero** {R : Type u_1} [CommRing R] [IsDomain R] {M : Type u_2} [AddCommGroup M] [Module R M] [Module.Free R M] [Module.Finite R M] {f : M →ₗ[R] M} (h : LinearMap.det f = 0) : ∃ φ : Module.Dual R M, φ ≠ 0 ∧ φ ∘ₗ f = 0

theorem **length_self_eq_top** (R : Type u_1) [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] : Module.length R R = ⊤

theorem **main** : ∀ {R : Type u_1} [inst : CommRing R] [inst_1 : IsDomain R] [IsDiscreteValuationRing R] {M : Type u_2} [inst_3 : AddCommGroup M] [inst_4 : Module R M] [Module.Free R M] [Module.Finite R M] {f : M →ₗ[R] M}, (LinearMap.det : (M →ₗ[R] M) → R) f = (0 : R) → Module.length R (M ⧸ LinearMap.range f) = ⊤

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsFiniteProjectiveCechComplexLineBundle

theorem **exists_finite_projective_complex_cechLB** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → Smooth f → ∃ (K : CochainComplex (ModuleCat.{0} R) ℕ) (φ : K ⟶ cechLB f M (fun i => nonzeroLocus (s i))), (∀ p, Module.Finite R (K.X p) ∧ Module.Projective R (K.X p)) ∧ ∀ (R' : Type) [CommRing R'] [Algebra R R'], QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsFiniteProjectiveCechComplexLineBundleStatement

theorem **_root_.GlobalTheory.T.exists_finite_projective_cechComplex_lineBundle** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → Smooth f → ∃ (K : CochainComplex (ModuleCat.{0} R) ℕ) (φ : K ⟶ cechLB f M (fun i => nonzeroLocus (s i))), (∀ p, Module.Finite R (K.X p) ∧ Module.Projective R (K.X p)) ∧ ∀ (R' : Type) [CommRing R'] [Algebra R R'], QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsFiniteProjectiveComplexOfNoetherianModel

theorem **Cube.nonempty_extendScalars_iso_baseChange** {ι : Type} {R₀ R : Type} [CommRing R₀] [CommRing R] [inst : Algebra R₀ R] (C : Cube.{0} ι R₀) : Nonempty (C ⋙ ModuleCat.extendScalars.{0, 0, 0} (algebraMap R₀ R) ≅ Cube.baseChange R C)

theorem **finite_projective_extendScalars_obj** {R₀ R : Type} [CommRing R₀] [CommRing R] (f : R₀ →+* R) (M : ModuleCat.{0} R₀) [Module.Finite R₀ M] [Module.Projective R₀ M] : Module.Finite R ((ModuleCat.extendScalars.{0, 0, 0} f).obj M) ∧ Module.Projective R ((ModuleCat.extendScalars.{0, 0, 0} f).obj M)

theorem **quasiIso_extendScalars_extendScalars** {R₀ R R' : Type} [CommRing R₀] [CommRing R] [CommRing R'] (f : R₀ →+* R) (g : R →+* R') {K L : CochainComplex (ModuleCat.{0} R₀) ℕ} (φ : K ⟶ L) (h : QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (g.comp f)).mapHomologicalComplex _).map φ)) : QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} g).mapHomologicalComplex _).map (((ModuleCat.extendScalars.{0, 0, 0} f).mapHomologicalComplex _).map φ))

theorem **exists_finite_projective_complex_of_noetherian_model** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R₀ R : Type) [CommRing R₀] [IsNoetherianRing R₀] [CommRing R] [Algebra R₀ R] (C₀ : Cube.{0} ι R₀) (A : Cube.{0} ι R) (θ : Cube.baseChange R C₀ ⟶ A), (∀ s : Finset ι, s.Nonempty → Module.Flat R₀ (C₀.obj s)) → (∀ p, Module.Finite R₀ ((cechComplex C₀).homology p)) → (∀ s : Finset ι, s.Nonempty → Function.Bijective (θ.app s).hom) → ∃ (K : CochainComplex (ModuleCat.{0} R) ℕ) (φ : K ⟶ cechComplex A), (∀ p, Fintype.card ι ≤ p + 1 → IsZero (K.X (p + 1))) ∧ (∀ p, Module.Finite R (K.X p) ∧ Module.Projective R (K.X p)) ∧ ∀ (R' : Type) [CommRing R'] [Algebra R R'], QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsFiniteProjectiveComplexQuasiIso

theorem **flat_of_shortExact_of_flat_of_flat** (f : A →ₗ[R] B) (g : B →ₗ[R] C) (hf : Function.Injective f) (hg : Function.Surjective g) (h : Function.Exact f g) [Module.Flat R B] [Module.Flat R C] : Module.Flat R A

theorem **flat_prod** [Module.Flat R A] [Module.Flat R B] : Module.Flat R (A × B)

theorem **exact_range_subtype_rangeRestrict** {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Function.Exact f g) : Function.Exact (LinearMap.range f).subtype g.rangeRestrict

theorem **flat_range_of_exact** {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Function.Exact f g) [Module.Flat R B] [Module.Flat R (LinearMap.range g)] : Module.Flat R (LinearMap.range f)

theorem **lTensor_exact_of_flat_range** {f : A →ₗ[R] B} {g : B →ₗ[R] C} {k : C →ₗ[R] D} (h : Function.Exact f g) (h' : Function.Exact g k) [Module.Flat R (LinearMap.range k)] (M : Type*) [AddCommGroup M] [Module R M] : Function.Exact (f.lTensor M) (g.lTensor M)

theorem **flat_range_of_exact_of_bounded** {E : ℕ → Type*} [∀ p, AddCommGroup (E p)] [∀ p, Module R (E p)] [∀ p, Module.Flat R (E p)] (D : ∀ p, E p →ₗ[R] E (p + 1)) (hD : ∀ p, Function.Exact (D p) (D (p + 1))) (n : ℕ) (hn : ∀ p, n < p → Subsingleton (E p)) (p : ℕ) : Module.Flat R (LinearMap.range (D p))

theorem **coneExact_id** {K : ℕ → Type v} [∀ p, AddCommGroup (K p)] (d : ∀ p, K p →+ K (p + 1)) : ConeExact (fun p => d p) (fun p => d p) (fun _ x => x)

theorem **coneD_apply** (dK : ∀ p, K p →ₗ[R] K (p + 1)) (dC : ∀ p, C p →ₗ[R] C (p + 1)) (φ : ∀ p, K p →ₗ[R] C p) (p : ℕ) (x : C p × K (p + 1)) : coneD dK dC φ p x = (dC p x.1 - φ (p + 1) x.2, - dK (p + 1) x.2)

theorem **coneExact_iff** (hK : ∀ p x, dK (p + 1) (dK p x) = 0) (hC : ∀ p x, dC (p + 1) (dC p x) = 0) (hφ : ∀ p x, dC p (φ p x) = φ (p + 1) (dK p x)) : ConeExact (fun p => dK p) (fun p => dC p) (fun p => φ p) ↔ Function.Injective ((φ 0).prod (dK 0)) ∧ Function.Exact ((φ 0).prod (dK 0)) (coneD dK dC φ 0) ∧ ∀ p, Function.Exact (coneD dK dC φ p) (coneD dK dC φ (p + 1))

theorem **flat_range_coneD** [∀ p, Module.Flat R (K (p + 1))] [∀ p, Module.Flat R (C p)] (h : ∀ p, Function.Exact (coneD dK dC φ p) (coneD dK dC φ (p + 1))) (n : ℕ) (hnK : ∀ p, n < p → Subsingleton (K p)) (hnC : ∀ p, n < p → Subsingleton (C p)) (p : ℕ) : Module.Flat R (LinearMap.range (coneD dK dC φ p))

theorem **ConeExact.flat_zero** [∀ p, Module.Flat R (K (p + 1))] [∀ p, Module.Flat R (C p)] (h : ConeExact (fun p => dK p) (fun p => dC p) (fun p => φ p)) (hK : ∀ p x, dK (p + 1) (dK p x) = 0) (hC : ∀ p x, dC (p + 1) (dC p x) = 0) (hφ : ∀ p x, dC p (φ p x) = φ (p + 1) (dK p x)) (n : ℕ) (hnK : ∀ p, n < p → Subsingleton (K p)) (hnC : ∀ p, n < p → Subsingleton (C p)) : Module.Flat R (K 0)

theorem **ConeExact.lTensor** [∀ p, Module.Flat R (K (p + 1))] [∀ p, Module.Flat R (C p)] (h : ConeExact (fun p => dK p) (fun p => dC p) (fun p => φ p)) (hK : ∀ p x, dK (p + 1) (dK p x) = 0) (hC : ∀ p x, dC (p + 1) (dC p x) = 0) (hφ : ∀ p x, dC p (φ p x) = φ (p + 1) (dK p x)) (n : ℕ) (hnK : ∀ p, n < p → Subsingleton (K p)) (hnC : ∀ p, n < p → Subsingleton (C p)) (M : Type*) [AddCommGroup M] [Module R M] : ConeExact (fun p => (dK p).lTensor M) (fun p => (dC p).lTensor M) (fun p => (φ p).lTensor M)

theorem **exists_free_step** {C₀ C₁ K₁ K₂ : Type*} [AddCommGroup C₀] [AddCommGroup C₁] [AddCommGroup K₁] [AddCommGroup K₂] [Module R C₀] [Module R C₁] [Module R K₁] [Module R K₂] [Module.Finite R K₁] (d : C₀ →ₗ[R] C₁) (φ : K₁ →ₗ[R] C₁) (δ : K₁ →ₗ[R] K₂) (B : Submodule R C₀)

[Module.Finite R (LinearMap.ker d ⧸ B.comap (LinearMap.ker d).subtype)] : ∃ (m : ℕ) (φ₀ : (Fin m → R) →ₗ[R] C₀) (δ₀ : (Fin m → R) →ₗ[R] K₁), (∀ x, d (φ₀ x) = φ (δ₀ x)) ∧ (∀ x, δ (δ₀ x) = 0) ∧ ∀ c k, d c = φ k → δ k = 0 → ∃ x, c - φ₀ x ∈ B ∧ δ₀ x = k

theorem **exists_free_complex** (n : ℕ) : ∀ (C : ℕ → ModuleCat.{v} R) (d : ∀ p, C p →ₗ[R] C (p + 1)) (B : Submodule R (C 0)) (_ : ∀ p, n ≤ p → Subsingleton (C p)) (_ : Module.Finite R (LinearMap.ker (d 0) ⧸ B.comap (LinearMap.ker (d 0)).subtype)) (_ : ∀ p, Module.Finite R (LinearMap.ker (d (p + 1)) ⧸ (LinearMap.range (d p)).comap (LinearMap.ker (d (p + 1))).subtype)), ∃ (m : ℕ → ℕ) (δ : ∀ p, (Fin (m p) → R) →ₗ[R] (Fin (m (p + 1)) → R)) (φ : ∀ p, (Fin (m p) → R) →ₗ[R] C p), (∀ p, n ≤ p → m p = 0) ∧ (∀ p x, δ (p + 1) (δ p x) = 0) ∧ (∀ p x, d p (φ p x) = φ (p + 1) (δ p x)) ∧ (∀ c k, d 0 c = φ 1 k → δ 1 k = 0 → ∃ x, c - φ 0 x ∈ B ∧ δ 0 x = k) ∧ ∀ p c k, d (p + 1) c = φ (p + 2) k → δ (p + 2) k = 0 → ∃ c' k', d p c' + φ (p + 1) k' = c ∧ δ (p + 1) k' = k

theorem **shortComplex_quasiIso_iff** {S T : ShortComplex (ModuleCat.{v} R)} (φ : S ⟶ T) : ShortComplex.QuasiIso φ ↔ (∀ y : T.X₂, T.g y = 0 → ∃ (x : S.X₂) (z : T.X₁), S.g x = 0 ∧ T.f z + φ.τ₂ x = y) ∧ (∀ (x : S.X₂) (z : T.X₁), S.g x = 0 → T.f z = φ.τ₂ x → ∃ w : S.X₁, S.f w = x)

theorem **cochain_d_d_apply** (K : CochainComplex (ModuleCat.{v} R) ℕ) (i j k : ℕ) (x : K.X i) : (K.d j k).hom ((K.d i j).hom x) = 0

theorem **cochain_comm_apply** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) (i j : ℕ) (x : K.X i) : (C.d i j).hom ((φ.f i).hom x) = (φ.f j).hom ((K.d i j).hom x)

theorem **quasiIsoAt_zero_iff** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) : QuasiIsoAt φ 0 ↔ (∀ y : C.X 0, (C.d 0 1).hom y = 0 → ∃ x : K.X 0, (K.d 0 1).hom x = 0 ∧ (φ.f 0).hom x = y) ∧ (∀ x : K.X 0, (K.d 0 1).hom x = 0 → (φ.f 0).hom x = 0 → x = 0)

theorem **quasiIsoAt_succ_iff** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) (p : ℕ) : QuasiIsoAt φ (p + 1) ↔ (∀ y : C.X (p + 1), (C.d (p + 1) (p + 2)).hom y = 0 → ∃ (x : K.X (p + 1)) (z : C.X p), (K.d (p + 1) (p + 2)).hom x = 0 ∧ (C.d p (p + 1)).hom z + (φ.f (p + 1)).hom x = y) ∧ (∀ (x : K.X (p + 1)) (z : C.X p), (K.d (p + 1) (p + 2)).hom x = 0 → (C.d p (p + 1)).hom z = (φ.f (p + 1)).hom x → ∃ w : K.X p, (K.d p (p + 1)).hom w = x)

theorem **quasiIso_iff_coneExact** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) : QuasiIso φ ↔ ConeExact (K

theorem **coneExact_lTensor_of_quasiIso** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) [hφ : QuasiIso φ] (hK : ∀ p, Module.Flat R (K.X p)) (hC : ∀ p, Module.Flat R (C.X p)) (n : ℕ) (hnK : ∀ p, n < p → IsZero (K.X p)) (hnC : ∀ p, n < p → IsZero (C.X p)) (M : Type*) [AddCommGroup M] [Module R M] : ConeExact (K

theorem **quasiIso_tensorLeft_of_flat** {K C : CochainComplex (ModuleCat.{u} R) ℕ} (φ : K ⟶ C) [QuasiIso φ] (hK : ∀ p, Module.Flat R (K.X p)) (hC : ∀ p, Module.Flat R (C.X p)) (n : ℕ) (hnK : ∀ p, n < p → IsZero (K.X p)) (hnC : ∀ p, n < p → IsZero (C.X p)) (M : ModuleCat.{u} R) : QuasiIso (((MonoidalCategory.tensorLeft M).mapHomologicalComplex _).map φ)

theorem **quasiIso_extendScalars_of_flat** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) [QuasiIso φ] (hK : ∀ p, Module.Flat R (K.X p)) (hC : ∀ p, Module.Flat R (C.X p)) (n : ℕ) (hnK : ∀ p, n < p → IsZero (K.X p)) (hnC : ∀ p, n < p → IsZero (C.X p)) {S : Type w} [CommRing S] (f : R →+* S) : QuasiIso (((ModuleCat.extendScalars.{u, w, v} f).mapHomologicalComplex _).map φ)

theorem **finite_ker_quotient_of_finite_homology** (C : CochainComplex (ModuleCat.{v} R) ℕ) (i j k : ℕ) (hi : (ComplexShape.up ℕ).prev j = i) (hk : (ComplexShape.up ℕ).next j = k) [Module.Finite R (C.homology j)] : Module.Finite R (LinearMap.ker (C.d j k).hom ⧸ (LinearMap.range (C.d i j).hom).comap (LinearMap.ker (C.d j k).hom).subtype)

theorem **exists_finite_complex_quasiIso** [IsNoetherianRing R] (C : CochainComplex (ModuleCat.{u} R) ℕ) (n : ℕ) (hn : ∀ p, n < p → IsZero (C.X p)) (hfin : ∀ p, Module.Finite R (C.homology p)) : ∃ (K : CochainComplex (ModuleCat.{u} R) ℕ) (φ : K ⟶ C), (∀ p, n < p → IsZero (K.X p)) ∧ (∀ p, Module.Finite R (K.X p)) ∧ (∀ p, 1 ≤ p → Module.Free R (K.X p)) ∧ QuasiIso φ

theorem **flat_zero_of_quasiIso** {K C : CochainComplex (ModuleCat.{v} R) ℕ} (φ : K ⟶ C) [hφ : QuasiIso φ] (hK : ∀ p, 1 ≤ p → Module.Flat R (K.X p)) (hC : ∀ p, Module.Flat R (C.X p)) (n : ℕ) (hnK : ∀ p, n < p → IsZero (K.X p)) (hnC : ∀ p, n < p → IsZero (C.X p)) : Module.Flat R (K.X 0)

theorem **exists_finite_projective_complex_quasiIso** [IsNoetherianRing R] (C : CochainComplex (ModuleCat.{u} R) ℕ) (n : ℕ) (hn : ∀ p, n < p → IsZero (C.X p)) (hflat : ∀ p, Module.Flat R (C.X p)) (hfin : ∀ p, Module.Finite R (C.homology p)) : ∃ (K : CochainComplex (ModuleCat.{u} R) ℕ) (φ : K ⟶ C), (∀ p, n < p → IsZero (K.X p)) ∧ (∀ p, Module.Finite R (K.X p)) ∧ (∀ p, 1 ≤ p → Module.Free R (K.X p)) ∧ (∀ p, Module.Projective R (K.X p)) ∧ QuasiIso φ ∧ (∀ M : ModuleCat.{u} R, QuasiIso (((MonoidalCategory.tensorLeft M).mapHomologicalComplex _).map φ)) ∧ (∀ (S : Type w) [CommRing S] [Algebra R S], QuasiIso (((ModuleCat.extendScalars.{u, w, u} (algebraMap R S)).mapHomologicalComplex _).map φ))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsGenThetaOfIsAbelianScheme

theorem **exists_gen_theta_of_isAbelianScheme** : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Spec (CommRingCat.of K))) [GrpObj B], IsAbelianScheme B → ∀ M : LB B.left, M.IsLB → ∃ t : LB.Gen (Theta B M), t.IsGen

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsGradedPresentationSaturating

theorem **exists_graded_presentation_saturating** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (G : Type) [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) [PolyGradedSMul ι 𝒢] [Decomposition 𝒢], (∀ i : ι, Module.FinitePresentation (MvPolynomial ι R) (QuotSMulTop (X i - 1 : MvPolynomial ι R) G)) → ∃ (a b : ℕ) (d : Fin a → ℤ) (e : Fin b → ℤ) (Φ : Matrix (Fin a) (Fin b) (MvPolynomial ι R)) (ψ : ((Fin a → MvPolynomial ι R) ⧸ LinearMap.range Φ.mulVecLin) →ₗ[MvPolynomial ι R] G), (∀ l j, Φ l j ∈ polyGrading ι R (e j - d l)) ∧ IsGradedMap (grQuot (grFree ι R d) (LinearMap.range Φ.mulVecLin)) 𝒢 ψ ∧ ∀ i : ι, (∀ y, ψ y = 0 → ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y = 0) ∧ (∀ z : G, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • z ∈ LinearMap.range ψ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsHighTwistSectionNeZero

theorem **nontrivial_of_invertible** (A L : Type) [CommRing A] [Nontrivial A] [AddCommGroup L] [Module A L] [Module.Invertible A L] : Nontrivial L

theorem **sec_eq_zero** (σ : LB.Sec M) (h : ∀ i, σ.f i = 0) : σ = 0

theorem **exists_restrict_ne_zero** (s : LB.Sec H) {t₀ : ℤ} (a : SecOn (tw M H t₀) ⊤) (ha : SecOn.restrict _ (le_top : nonzeroLocus s ≤ ⊤) a ≠ 0) (t : ℤ) (ht : t₀ ≤ t) : ∃ b : SecOn (tw M H t) ⊤, SecOn.restrict _ (le_top : nonzeroLocus s ≤ ⊤) b ≠ 0

theorem **exists_sec_tw_ne_zero** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [Nonempty Y] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∃ t₀ : ℤ, ∀ t, t₀ ≤ t → ∃ (σ : LB.Sec (tw M H t)) (i : (tw M H t).ι), σ.f i ≠ 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsHighTwistSectionNeZeroStatement

theorem **_root_.GlobalTheory.T.exists_high_twist_section_ne_zero** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [Nonempty Y] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∃ t₀

: ℤ, ∀ t, t₀ ≤ t → ∃ (σ : LB.Sec (tw M H t)) (i : (tw M H t).ι), σ.f i ≠ 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsHomLocBijectiveGrSum

theorem **GrSum.X_smul_bijective** {A : ℤ → Type} [∀ n, AddCommGroup (A n)] [∀ n, Module R (A n)] {x : ι → ∀ n, A n →ₗ[R] A (n + 1)} (hx : OperatorsCommute x) {i : ι} (h : ∀ t, Function.Bijective (x i t)) : Function.Bijective fun m : GrSum hx => (X i : MvPolynomial ι R) • m

theorem **xProd_smul_bijective_of** {N : Type} [AddCommGroup N] [Module (MvPolynomial ι R) N] (s : Finset ι) (h : ∀ i ∈ s, Function.Bijective fun m : N => (X i : MvPolynomial ι R) • m) : Function.Bijective fun m : N => xProd R s • m

theorem **exists_homLoc_bijective_of** {M : Type} [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] {N : Type} [AddCommGroup N] [Module R N] [Module (MvPolynomial ι R) N] [IsScalarTower R (MvPolynomial ι R) N] {𝒜 : ℤ → Submodule R M} [PolyGradedSMul ι 𝒜] [Decomposition 𝒜] {𝒩 : ℤ → Submodule R N} [PolyGradedSMul ι 𝒩] [Decomposition 𝒩] {f : M →ₗ[MvPolynomial ι R] N} (hf : IsGradedMap 𝒜 𝒩 f) (s : Finset ι) (hb : Function.Bijective fun m : N => xProd R s • m) (h₁ : ∀ n, ∀ y ∈ 𝒜 n, f y = 0 → ∃ k : ℕ, xProd R s ^ k • y = 0) (h₂ : ∀ n, ∀ a ∈ 𝒩 n, ∃ (k : ℕ) (y : M), f y = xProd R s ^ k • a) (n : ℤ) : ∃ φ : homLoc 𝒜 s n →ₗ[R] 𝒩 n, Function.Bijective φ ∧ ∀ (z : homLoc 𝒜 s n) (k : ℕ) (m : M), awayFrac M (xProd R s) k m = z → xProd R s ^ k • (φ z : N) = f m

theorem **exists_homLoc_bijective_grSum** {M : Type} [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] {𝒜 : ℤ → Submodule R M} [PolyGradedSMul ι 𝒜] [Decomposition 𝒜] {A : ℤ → Type} [∀ n, AddCommGroup (A n)] [∀ n, Module R (A n)] {x : ι → ∀ n, A n →ₗ[R] A (n + 1)} (hx : OperatorsCommute x) {f : M →ₗ[MvPolynomial ι R] GrSum hx} (hf : IsGradedMap 𝒜 (GrSum.grading hx) f) (s : Finset ι) (hb : ∀ i ∈ s, ∀ t, Function.Bijective (x i t)) (h₁ : ∀ n, ∀ y ∈ 𝒜 n, f y = 0 → ∃ k : ℕ, xProd R s ^ k • y = 0) (h₂ : ∀ n (a : A n), ∃ (k : ℕ) (y : M), f y = xProd R s ^ k • GrSum.of hx n a) (n : ℤ) : ∃ φ : homLoc 𝒜 s n →ₗ[R] A n, Function.Bijective φ ∧ ∀ (z : homLoc 𝒜 s n) (k : ℕ) (m : M), awayFrac M (xProd R s) k m = z → xProd R s ^ k • GrSum.of hx n (φ z) = f m

theorem **homLoc_grSum_res** {M : Type} [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] {𝒜 : ℤ → Submodule R M} [PolyGradedSMul ι 𝒜] {A : ℤ → Type} [∀ n, AddCommGroup (A n)] [∀ n, Module R (A n)] {x : ι → ∀ n, A n →ₗ[R] A (n + 1)} (hx : OperatorsCommute x) {B : ℤ → Type} [∀ n, AddCommGroup (B n)] [∀ n, Module R (B n)] {y : ι → ∀ n, B n →ₗ[R] B (n + 1)} (hy : OperatorsCommute y) (r : ∀ n, A n →ₗ[R] B n) (hr : ∀ i n a, r (n + 1) (x i n a) = y i n (r n a)) {f : M →ₗ[MvPolynomial ι R] GrSum hx} (hf : IsGradedMap 𝒜 (GrSum.grading hx) f) {g : M →ₗ[MvPolynomial ι R] GrSum hy} (hfg : ∀ n m a, f m = GrSum.of hx n a → g m = GrSum.of hy n (r n a)) {s t : Finset ι} (h : s ⊆ t) (hb : ∀ i ∈ t, ∀ n, Function.Bijective (y i n)) {n : ℤ} (z : homLoc 𝒜 s n) (a : A n) (b : B n) (ha : ∀ (k : ℕ) (m : M), awayFrac M (xProd R s) k m = z → xProd R s ^ k • GrSum.of h

theorem **exists_grCube_hom_bijective** (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (G : Type) [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) [PolyGradedSMul ι 𝒢] [Decomposition 𝒢] (A : ℤ → Cube.{0} ι R) (x : ι → ∀ t, A t ⟶ A (t + 1)) (hx : ∀ I : Finset ι, OperatorsCommute (A

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsHomogeneousFamilySaturating

theorem **_root_.GlobalTheory.exists_homogeneous_family_saturating** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [_root_.GlobalTheory.PolyGradedSMul ι 𝒜] [Decomposition 𝒜], (∀ i : ι, ∃ T : Finset M, ∀ y : M, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ Submodule.span (MvPolynomial ι R) (T : Set M)) → ∃ (r : ℕ) (d : Fin r → ℤ) (g : Fin r → M), (∀ l, g l ∈ 𝒜 (d l)) ∧ ∀ (i : ι) (y : M), ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ Submodule.span (MvPolynomial ι R) (Set.range g)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsHomogeneousFormEqHighPowerSection

theorem **exists_X_pow_smul_mem_span_of_dehomogenisation** {ι : Type} {R : Type} [CommRing R] {G : Type} [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) [PolyGradedSMul ι 𝒢] [Decomposition 𝒢] (i : ι) {B N : Type} [CommRing B] [AddCommGroup N] [Module B N] {π : MvPolynomial ι R →+* B} (hπ : Function.Surjective π) (hi : π (MvPolynomial.X i) = 1) (ε : G →ₛₗ[π] N) (hker : ∀ n, ∀ y ∈ 𝒢 n, ε y = 0 → ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • y = 0) {g : G} (hg : g ∈ 𝒢 0) (hgen : ∀ z : N, ∃ b : B, z = b • ε g) (K : Submodule (MvPolynomial ι R) G) (hK : g ∈ K) (y : G) : ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • y ∈ K

theorem **operatorsCommute_mulSec** (V : Y.Opens) : OperatorsCommute (ι

theorem **restrict_eq_zero_of_X_pow_smul** (i : Fin (m + 1)) (k : ℕ) : ∀ (n : ℤ) (a : SecOnR f (tw M H n) ⊤), (MvPolynomial.X i : MvPolynomial (Fin (m + 1)) R) ^ k • GrSum.of (operatorsCommute_mulSec f s M ⊤) n a = 0 → SecOnR.restrict f _ (le_top : nonzeroLocus (s i) ≤ ⊤) a = 0

theorem **grSum_eq_zero_of_X_pow_smul** (hs : ⊔ i, nonzeroLocus (s i) = ⊤) (y : GrSum (operatorsCommute_mulSec f s M ⊤)) (h : ∀ i, ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial (Fin (m + 1)) R) ^ k • y = 0) : y = 0

theorem **twCoord_twZeroTrivEquiv_one** (hH : H.IsLB) (V : Y.Opens) (p : (Inh.trivLB Y).ι) (k : H.ι) : SecOn.twCoord (twZeroTrivEquiv hH V 1) p k = 1

theorem **exists_eq_smul_restrict_one** (hH : H.IsLB) (V : Y.Opens) (z : SecOn (tw (Inh.trivLB Y) H 0) V) : ∃ b : Γ(Y, V), z = b • SecOn.restrict _ (le_top : V ≤ ⊤) (twZeroTrivEquiv hH ⊤ 1)

theorem **exists_form_eq_sec_pw** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∃ t₀ : ℕ, ∀ t : ℕ, t₀ ≤ t → ∀ σ : LB.Sec (pw H t), ∃ F : MvPolynomial (Fin (m + 1)) R, F.IsHomogeneous t ∧ ∀ k : H.ι, σ.f k = MvPolynomial.eval₂Hom (cst f (H.U k)) (fun j => (s j).f k) F

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsHomogeneousFormEqHighPowerSectionStatement

theorem **_root_.GlobalTheory.T.exists_homogeneousForm_eq_highPowerSection** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → ∃ t₀ : ℕ, ∀ t : ℕ, t₀ ≤ t → ∀ σ : LB.Sec (pw H t), ∃ F : MvPolynomial (Fin (m + 1)) R, F.IsHomogeneous t ∧ ∀ k : H.ι, σ.f k = MvPolynomial.eval₂Hom (cst f (H.U k)) (fun j => (s j).f k) F

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsIsGenOfUniqueFactorizationMonoid

theorem **invertible_of_isLocalizedModule_span** {A M : Type*} [CommRing A] [AddCommGroup M] [Module A M] (s : Set A) (hs : Ideal.span s = ⊤) (B : s → Type*) [∀ r, CommRing (B r)] [∀ r, Algebra A (B r)] [∀ r : s, IsLocalization.Away r.1 (B r)] (N : s → Type*) [∀ r, AddCommGroup (N r)] [∀ r, Module A (N r)] [∀ r, Module (B r) (N r)] [∀ r : s, IsScalarTower A (B r) (N r)] (j : ∀ r, M →ₗ[A] N r) [∀ r : s, IsLocalizedModule.Away r.1 (j r)] (h : ∀ r, Module.Invertible (B r) (N r)) : Module.Invertible A M

theorem **invertible_of_ringEquiv** {A B M : Type*} [CommRing A] [CommRing B] [AddCommGroup M] [Module A M] [Module B M] (e : A ≃+* B) (h : ∀ (a : A) (m : M), a • m = e a • m) [Module.Invertible A M] : Module.Invertible B M

theorem **bijective_smul_of_isBaseChange** {A B M N : Type*} [CommRing A] [CommRing B] [Algebra A B] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module A N] [Module B N] [IsScalarTower A B N] {j : M →ₗ[A] N} (hj : IsBaseChange B j) {e : M} (he : Function.Bijective fun a : A => a • e) : Function.Bijective fun b : B => b • j e

theorem **free_iff_exists_bijective_smul** {A M : Type*} [CommRing A] [AddCommGroup M] [Module A M] [Module.Invertible A M] : Module.Free A M ↔ ∃ e : M, Function.Bijective fun a : A => a • e

theorem **isFrame_chartSec** {L : LB X} (hL : L.IsLB) (i : L.ι) : IsFrame (chartSec hL i)

theorem **isFrame_genSec** {L : LB X} (hL : L.IsLB) {t : LB.Gen L} (ht : t.IsGen) (U : X.Opens) : IsFrame (ofSec (DVR.genSec t ht) U)

theorem **map_smul_resG** (h : V ≤ U) (a : Γ(X, U)) (s : Γ(F, U)) : F.presheaf.map (homOfLE h).op (a • s) = resG X h a • F.presheaf.map (homOfLE h).op s

theorem **app_map** (φ : F ⟶ G) (h : V ≤ U) (s : Γ(F, U)) : φ.app V (F.presheaf.map (homOfLE h).op s) = G.presheaf.map (homOfLE h).op (φ.app U s)

@[simp] theorem **appLinear_apply** (φ : F ⟶ G) (V : X.Opens) (s : Γ(F, V)) : appLinear φ V s = φ.app V s

@[simp] theorem **appLinearEquiv_apply** (φ : F ≅ G) (V : X.Opens) (s : Γ(F, V)) : appLinearEquiv φ V s = φ.hom.app V s

theorem **mod_ext_of_cover** {ι : Type*} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) {s t : Γ(F, V)} (h : ∀ k, F.presheaf.map (homOfLE (hW k)).op s = F.presheaf.map (homOfLE (hW k)).op t) : s = t

theorem **exists_mod_of_cover** {ι : Type*} (W : ι → X.Opens) (hW : ∀ k, W k ≤ V) (hc : V ≤ iSup W) (s : ∀ k, Γ(F, W k)) (h : ∀ k l, F.presheaf.map (homOfLE (inf_le_left : W k ⊓ W l ≤ W k)).op (s k) = F.presheaf.map (homOfLE (inf_le_right : W k ⊓ W l ≤ W l)).op (s l)) : ∃ t : Γ(F, V), ∀ k, F.presheaf.map (homOfLE (hW k)).op t = s k

theorem **map** (he : IsFrame e) (h : V ≤ U) : IsFrame (F.presheaf.map (homOfLE h).op e)

theorem **app** (he : IsFrame e) (φ : F ≅ G) : IsFrame (φ.hom.app U e)

theorem **repr_symm_apply** (he : IsFrame e) (h : V ≤ U) (a : Γ(X, V)) : (he.repr h).symm a = a • F.presheaf.map (homOfLE h).op e

theorem **repr_smul** (he : IsFrame e) (h : V ≤ U) (s : Γ(F, V)) : he.repr h s • F.presheaf.map (homOfLE h).op e = s

theorem **repr_eq** (he : IsFrame e) (h : V ≤ U) {s : Γ(F, V)} {a : Γ(X, V)} (ha : a • F.presheaf.map (homOfLE h).op e = s) : he.repr h s = a

theorem **resG_repr** (he : IsFrame e) (h : V ≤ U) (h' : W ≤ V) (s : Γ(F, V)) : resG X h' (he.repr h s) = he.repr (h'.trans h) (F.presheaf.map (homOfLE h').op s)

theorem **repr_change** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') (s : Γ(F, V)) : he.repr h s = he.repr h (F.presheaf.map (homOfLE h').op e') ∗ he'.repr h' s

theorem **repr_self** (he : IsFrame e) (h : V ≤ U) : he.repr h (F.presheaf.map (homOfLE h).op e) = 1

theorem **val_unit** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') : (he.unit he' h h' : Γ(X, V)) = he.repr h (F.presheaf.map (homOfLE h').op e')

theorem **resG_unit** (he : IsFrame e) (he' : IsFrame e') (h : V ≤ U) (h' : V ≤ U') (h'' : W ≤ V) : resG X h'' (he.unit he' h h' : Γ(X, V)) = he.repr (h''.trans h) (F.presheaf.map (homOfLE (h''.trans h')).op e')

theorem **exists_isFrame_of_iso** (φ : unitMod U.toScheme ≅ F.restrict U.ι) : ∃ e : Γ(F, U), IsFrame e

theorem **resG_ofFrames_g** (he : ∀ i, IsFrame (e i)) (i j : ι) {W : X.Opens} (h : W ≤ U i ⊓ U j) : resG X h ((ofFrames he).g i j : Γ(X, U i ⊓ U j)) = (he i).repr (h.trans inf_le_left) (F.presheaf.map (homOfLE (h.trans inf_le_right)).op (e j))

theorem **ofFrames_isLB** (he : ∀ i, IsFrame (e i)) (hU : ⊔ i, U i = ⊤) : (ofFrames he).IsLB where

theorem **coord_toFramesLinear** (he : ∀ i, IsFrame (e i)) (V : X.Opens) (s : Γ(F, V)) (i : ι) : coord (toFramesLinear he V s) i = (he i).repr (inf_le_right : V ⊓ U i ≤ U i) (F.presheaf.map (homOfLE inf_le_left).op s)

theorem **bijective_toFramesLinear** (he : ∀ i, IsFrame (e i)) (hU : ⊔ i, U i = ⊤) (V : X.Opens) : Function.Bijective (toFramesLinear he V)

theorem **coord_ofFramesIso_hom_app** (he : ∀ i, IsFrame (e i)) (hU : ⊔ i, U i = ⊤) (V : X.Opens) (s : Γ(F, V)) (i : ι) : coord ((ofFramesIso he hU).hom.app V s) i = (he i).repr (inf_le_right : V ⊓ U i ≤ U i) (F.presheaf.map (homOfLE inf_le_left).op s)

theorem **ofFramesIso_hom_app_frame** (he : ∀ i, IsFrame (e i)) (hU : ⊔ i, U i = ⊤) (i : ι) : (ofFramesIso he hU).hom.app (U i) (e i) = chartSec (ofFrames_isLB he hU) i

theorem **exists_toMod_iso** (F : X.Modules) (h : ∀ x : X, ∃ (U : X.Opens) (_ : x ∈ U) (e : Γ(F, U)), IsFrame e) : ∃ L : LB X, L.IsLB ∧ Nonempty (toMod L ≅ F)

theorem **exists_toMod_iso_iff** (F : X.Modules) : (∃ L : LB X, L.IsLB ∧ Nonempty (toMod L ≅ F)) ↔ ∀ x : X, ∃ (U : X.Opens) (_ : x ∈ U) (e : Γ(F, U)), IsFrame e

theorem **exists_toMod_iso_of_restrict_iso** (F : X.Modules) (h : ∀ x : X, ∃ (U : X.Opens) (_ : x ∈ U), Nonempty (unitMod U.toScheme ≅ F.restrict U.ι)) : ∃ L : LB X, L.IsLB ∧ Nonempty (toMod L ≅ F)

theorem **repr_chartSec** (hL : L.IsLB) (i : L.ι) {V : X.Opens} (h : V ≤ L.U i) (s : Γ(toMod L, V)) : (isFrame_chartSec hL i).repr h s = chartEquiv hL i h s

theorem **map_chartSec_change** (hL : L.IsLB) (i j : L.ι) {V : X.Opens} (hi : V ≤ L.U i) (hj : V ≤ L.U j) : (toMod L).presheaf.map (homOfLE hj).op (chartSec hL j) = resG X (le_inf hi hj) (L.g i j : Γ(X, L.U i ⊓ L.U j)) • (toMod L).presheaf.map (homOfLE hi).op (chartSec hL i)

theorem **resG_homFun** (hL : L.IsLB) (hM : M.IsLB) (φ : toMod L ⟶ toMod M) (p : M.ι) (q : L.ι) {W : X.Opens} (h : W ≤ M.U p ⊓ L.U q) : resG X h (homFun hL hM φ p q) = chartEquiv hM p (h.trans inf_le_left) (appLinear φ W ((toMod L).presheaf.map (homOfLE (h.trans inf_le_right)).op (chartSec hL q)))

theorem **homCoord_secOfHom** (hL : L.IsLB) (hM : M.IsLB) (φ : toMod L ⟶ toMod M) (p : M.ι) (q : L.ι) : homCoord (secOfHom hL hM φ) p q = homFun hL hM φ p q

theorem **homOfSec_secOfHom** (hL : L.IsLB) (hM : M.IsLB) (φ : toMod L ⟶ toMod M) : homOfSec hL hM (secOfHom hL hM φ) = φ

theorem **secOfHom_homOfSec** (hL : L.IsLB) (hM : M.IsLB) (w : LB.Sec (LB.div M L)) : secOfHom hL hM (homOfSec hL hM w) = w

theorem **genSec_injective** {t t' : LB.Gen L} (ht : t.IsGen) (ht' : t'.IsGen) (h : DVR.genSec t ht = DVR.genSec t' ht') : t = t'

theorem **exists_genSec_eq** (σ : LB.Sec L) (h : ∀ i, IsUnit (σ.f i)) : ∃ (t : LB.Gen L) (ht : t.IsGen), DVR.genSec t ht = σ

theorem **isUnit_homFun** (hL : L.IsLB) (hM : M.IsLB) (φ : toMod L ≅ toMod M) (p : M.ι) (q : L.ι) : IsUnit (homFun hL hM φ.hom p q)

theorem **existsUnique_gen_of_iso** (hL : L.IsLB) (hM : M.IsLB) (φ : toMod L ≅ toMod M) : ∃! w : LB.Gen (LB.div M L), ∃ hw : w.IsGen, isoOfGen hL hM w hw = φ

theorem **lbIso_iff_nonempty_iso** (hL : L.IsLB) (hM : M.IsLB) : LBIso L M ↔ Nonempty (toMod L ≅ toMod M)

theorem **isFrame_ofSec_iff** (hL : L.IsLB) (σ : LB.Sec L) : IsFrame (ofSec σ ⊤) ↔ ∃ (t : LB.Gen L) (ht : t.IsGen), DVR.genSec t ht = σ

theorem **exists_isGen_iff_isFrame** (hL : L.IsLB) : (∃ t : LB.Gen L, t.IsGen) ↔ ∃ e : Γ(toMod L, ⊤), IsFrame e

theorem **exists_isGen_iff_nonempty_iso** (hL : L.IsLB) : (∃ t : LB.Gen L, t.IsGen) ↔ Nonempty (unitMod X ≅ toMod L)

theorem **isFrame_of_bijective** [F.IsQuasicoherent] (hU : IsAffineOpen U) {e : Γ(F, U)} (he : Function.Bijective fun a : Γ(X, U) => a • e) : IsFrame e

theorem **isFrame_iff_bijective** [F.IsQuasicoherent] (hU : IsAffineOpen U) (e : Γ(F, U)) : IsFrame e ↔ Function.Bijective fun a : Γ(X, U) => a • e

theorem **exists_isFrame_of_invertible** [F.IsQuasicoherent] (hU : IsAffineOpen U) [Module.Invertible Γ(X, U) Γ(F, U)] {x : X} (hx : x ∈ U) : ∃ (V : X.Opens) (_ : x ∈ V) (e : Γ(F, V)), IsFrame e

theorem **invertible_sections** (hL : L.IsLB) (hU : IsAffineOpen U) : Module.Invertible Γ(X, U) Γ(toMod L, U)

theorem **exists_isFrame_iff_free** (hL : L.IsLB) (hU : IsAffineOpen U) : (∃ e : Γ(toMod L, U), IsFrame e) ↔ Module.Free Γ(X, U) Γ(toMod L, U)

theorem **invertible_sec_top** [IsAffine X] (hL : L.IsLB) : Module.Invertible Γ(X, ⊤) (LB.Sec L)

theorem **exists_genSec_eq_iff_bijective** [IsAffine X] (hL : L.IsLB) (σ : LB.Sec L) : (∃ (t : LB.Gen L) (ht : t.IsGen), DVR.genSec t ht = σ) ↔ Function.Bijective fun a : Γ(X, ⊤) => a • σ

theorem **exists_isGen_iff_free_top** [IsAffine X] (hL : L.IsLB) : (∃ t : LB.Gen L, t.IsGen) ↔ Module.Free Γ(X, ⊤) (LB.Sec L)

theorem **exists_isFrame_of_subsingleton_pic** (hL : L.IsLB) (hU : IsAffineOpen U) [Subsingleton (CommRing.Pic Γ(X, U))] : ∃ e : Γ(toMod L, U), IsFrame e

theorem **exists_isGen_of_subsingleton_pic_top** [IsAffine X] [Subsingleton (CommRing.Pic Γ(X, ⊤))] (hL : L.IsLB) : ∃ t : LB.Gen L, t.IsGen

theorem **exists_toMod_iso_of_invertible** [IsAffine X] (F : X.Modules) [F.IsQuasicoherent] [Module.Invertible Γ(X, ⊤) Γ(F, ⊤)] : ∃ L : LB X, L.IsLB ∧ Nonempty (toMod L ≅ F)

theorem **globalRingEquiv_apply** (r : R) : globalRingEquiv R r = (Scheme.ΓSpecIso (CommRingCat.of R)).inv r

theorem **sp_smul_def** (r : R) (σ : LB.Sec L) : r • σ = (show Γ(Sp R, ⊤) from (Scheme.ΓSpecIso (CommRingCat.of R)).inv r) • σ

theorem **sp_smul_f** (r : R) (σ : LB.Sec L) (i : L.ι) : (r • σ).f i = resG (Sp R) le_top ((Scheme.ΓSpecIso (CommRingCat.of R)).inv r) * σ.f i

theorem **spec_smul_top** {S : CommRingCat.{0}} (F : (Spec S).Modules) (r : S) (s : Γ(F, ⊤)) : r • s = (show Γ(Spec S, ⊤) from (Scheme.ΓSpecIso S).inv r) • s

theorem **invertible_sec** (hL : L.IsLB) : Module.Invertible R (LB.Sec L)

theorem **exists_genSec_eq_iff_bijective_sp** (hL : L.IsLB) (σ : LB.Sec L) : (∃ (t : LB.Gen L) (ht : t.IsGen), DVR.genSec t ht = σ) ↔ Function.Bijective fun r : R => r • σ

theorem **exists_isGen_iff_free** (hL : L.IsLB) : (∃ t : LB.Gen L, t.IsGen) ↔ Module.Free R (LB.Sec L)

theorem **lbIso_iff_nonempty_linearEquiv** (hL : L.IsLB) (hM : M.IsLB) : LBIso L M ↔ Nonempty (LB.Sec L ≃ₗ[R] LB.Sec M)

theorem **exists_toMod_iso_tilde** {S : CommRingCat.{0}} (N : ModuleCat.{0} S) [Module.Invertible S N] : ∃ L : LB (Spec S), L.IsLB ∧ Nonempty (toMod L ≅ tilde N)

theorem **exists_sec_linearEquiv** (N : Type*) [AddCommGroup N] [Module R N] [Module.Invertible R N] : ∃ L : LB (Sp R), L.IsLB ∧ Nonempty (LB.Sec L ≃ₗ[R] N)

theorem **exists_isGen_of_subsingleton_pic** [Subsingleton (CommRing.Pic R)] (hL : L.IsLB) : ∃ t : LB.Gen L, t.IsGen

theorem **subsingleton_pic_iff** : Subsingleton (CommRing.Pic R) ↔ ∀ L : LB (Sp R), L.IsLB → ∃ t : LB.Gen L, t.IsGen

theorem **exists_isGen_of_isLocalRing** [IsLocalRing R] (hL : L.IsLB) : ∃ t : LB.Gen L, t.IsGen

theorem **exists_isGen_of_finite_maximalSpectrum** [Finite (MaximalSpectrum R)] (hL : L.IsLB) : ∃ t : LB.Gen L, t.IsGen

theorem **exists_isGen_of_uniqueFactorizationMonoid** [IsDomain R] [UniqueFactorizationMonoid R] (hL : L.IsLB) : ∃ t : LB.Gen L, t.IsGen

theorem **exists_isGen_of_isPrincipalIdealRing** [IsDomain R] [IsPrincipalIdealRing R] (hL : L.IsLB) : ∃ t : LB.Gen L, t.IsGen

theorem **invertible_sheaves_and_modules.{v}** : (∀ (X : Scheme.{0}) (F : X.Modules), (∀ x : X, ∃ (U : X.Opens) (_ : x ∈ U) (e : Γ(F, U)), ∀ (V : X.Opens) (h : V ≤ U), Function.Bijective fun a : Γ(X, V) => a • F.presheaf.map (homOfLE h).op e) → ∃ L : LB X, L.IsLB ∧ Nonempty (toMod L ≅ F)) ∧ (∀ (R : Type) [CommRing R], (∀ L : LB (Sp R), L.IsLB → Module.Invertible R (LB.Sec L)) ∧ (∀ (N : Type v) [AddCommGroup N] [Module R N], Module.Invertible R N → ∃ L : LB (Sp R), L.IsLB ∧ Nonempty (LB.Sec L ≃ₗ[R] N)) ∧ (∀ L : LB (Sp R), L.IsLB → ((∃ t : LB.Gen L, t.IsGen) ↔ Module.Free R (LB.Sec L))) ∧ (∀ L M : LB (Sp R), L.IsLB → M.IsLB → (LBIso L M ↔ Nonempty (LB.Sec L ≃ₗ[R] LB.Sec M))) ∧ ((IsLocalRing R ∨ Finite (MaximalSpectrum R) ∨ (∃ _ : IsDomain R, IsPrincipalIdealRing R ∨ UniqueFactorizationMonoid R)) → ∀ L : LB (Sp R), L.IsLB → ∃ t : LB.Gen L, t.IsGen)) ∧ (∀ (X : Scheme.{0}) (L M : LB X) (hL : L.Is

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsIsRegularQuotSMulTop

theorem **maximalIdeal_smul_top_lt_top** {R : Type u_1} [CommRing R] [IsLocalRing R] (M : Type u_2) [AddCommGroup M] [Module R M] [Module.Finite R M] [Nontrivial M] : IsLocalRing.maximalIdeal R • (⊤ : Submodule R M) < ⊤

theorem **exists_isRegular_quotSMulTop** {R : Type u_1} [CommRing R] [IsNoetherianRing R] [IsLocalRing R] [Small.{u_2, u_1} R] {M : Type u_2} [AddCommGroup M] [Module R M] [Module.Finite R M] {xs : List R} (hxs : IsRegular M xs) {n : ℕ} (hn : n + 1 ≤ xs.length) {y : R} (hy : y ∈ IsLocalRing.maximalIdeal R) (hreg : IsSMulRegular M y) : ∃ zs : List R, zs.length = n ∧ IsRegular (QuotSMulTop y M) zs

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsLinearEquivMapLatticeEq

theorem **span_range_basis_coe** {R : Type*} [CommRing R] {V : Type*} [AddCommGroup V] [Module R V] {ι : Type*} (A : Submodule R V) (b : Basis ι R A) : Submodule.span R (Set.range fun i => (b i : V)) = A

theorem **exists_linearEquiv_map_lattice_eq** : ∀ {R : Type u_1} [inst : CommRing R] [inst_1 : IsDomain R] [IsDiscreteValuationRing R] (K : Type u_2) [inst_3 : Field K] [inst_4 : Algebra R K] [IsFractionRing R K] {V : Type u_3} [inst_6 : AddCommGroup V] [inst_7 : Module K V] [inst_8 : Module R V] [inst_9 : IsScalarTower R K V] (A B : Submodule R V) [Submodule.IsLattice K A] [Submodule.IsLattice K B], Exists fun (g : LinearEquiv (RingHom.id K) V V) ↦ Eq (Submodule.map (LinearMap.restrictScalars R (LinearEquiv.toLinearMap g)) A) B

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsLinearMapMultilinearMapBaseChange

theorem **multilinearMap_ext_one_tmul** {ι : Type*} [Finite ι] {Φ Ψ : MultilinearMap K (fun _ : ι ↦ K ⊗[R] G) N} (h : ∀ x : ι → G, Φ (fun i ↦ 1 ⊗ₜ x i) = Ψ fun i ↦ 1 ⊗ₜ x i) : Φ = Ψ

theorem **exists_linearMap_multilinearMap_baseChange** (d : ℕ) : ∃ E : MultilinearMap R (fun _ : Fin d ↦ G) N →ₗ[R] MultilinearMap K (fun _ : Fin d ↦ K ⊗[R] G) N, ∀ φ x, E φ (fun i ↦ 1 ⊗ₜ x i) = φ x

theorem **existsUnique_multilinearMap_baseChange** {d : ℕ} (φ : MultilinearMap R (fun _ : Fin d ↦ G) N) : ∃! Φ : MultilinearMap K (fun _ : Fin d ↦ K ⊗[R] G) N, ∀ x, Φ (fun i ↦ 1 ⊗ₜ x i) = φ x

theorem **DiffVanish.multiDiff_eq_of_le_card** (h : DiffVanish (d + 1) φ) (hd : d ≤ Fintype.card ι) (g : ι → G) (x y : G) : multiDiff g φ x = multiDiff g φ y

theorem **DiffVanish.multiDiff_update_add_of_le_card** [DecidableEq ι] (h : DiffVanish (d + 1) φ) (hd : d ≤ Fintype.card ι) (g : ι → G) (i : ι) (a b : G) : multiDiff (Function.update g i (a + b)) φ = multiDiff (Function.update g i a) φ + multiDiff (Function.update g i b) φ

theorem **bwdDiff_chiPic_eq_chiPic_cut** {L : LB X} (hL : L.IsLB) (s : L.Sec) [Mono (cutHom F hL s)] (h : FiniteCohTwists k F) : bwdDiff (cls L hL) (chiPic k F) = chiPic k (cut F hL s)

theorem **IsAmpleLB.diffVanish_chiPic_of_suppDim_lt** [IsLocallyNoetherian X] {A : LB X} (hA : IsAmpleLB A) [IsFiniteQC F] (h : FiniteCohTwists k F) {n : ℕ} (hn : suppDim F < n) : DiffVanish n (chiPic k F)

theorem **IsAmpleLB.diffVanish_chiPic** [IsLocallyNoetherian X] {A : LB X} (hA : IsAmpleLB A) [IsFiniteQC F] (h : FiniteCohTwists k F) {d : ℕ} (hd : suppDim F ≤ d) : DiffVanish (d + 1) (chiPic k F)

theorem **IsAmpleLB.finiteCohTwists** [IsProper f] {A : LB X} (hA : IsAmpleLB A) : FiniteCohTwists k F

theorem **IsAmpleLB.diffVanish_chiPic_of_isProper** [IsProper f] {A : LB X} (hA : IsAmpleLB A) {d : ℕ} (hd : suppDim F ≤ d) : DiffVanish (d + 1) (chiPic k F)

theorem **diffVanish_chiPic_of_isProjOver_of_lt** (h : IsProjOver f) {n : ℕ} (hn : suppDim F < n) : DiffVanish n (chiPic k F)

theorem **diffVanish_chiPic_of_isProjOver** (h : IsProjOver f) {d : ℕ} (hd : suppDim F ≤ d) : DiffVanish (d + 1) (chiPic k F)

theorem **interNum_eq_zero_of_isProjOver** (h : IsProjOver f) {ι : Type} [Fintype ι] (hn : suppDim F < Fintype.card ι) {L : ι → LB X} (hL : ∀ i, (L i).IsLB) : interNum k F L = 0

theorem **isLB_update** [DecidableEq ι] (hL : ∀ i, (L i).IsLB) (i : ι) (hM : M.IsLB) : ∀ j, (Function.update L i M j).IsLB

theorem **cls_update** [DecidableEq ι] (hL : ∀ i, (L i).IsLB) (i : ι) (hM : M.IsLB) : (fun j ↦ cls _ (isLB_update hL i hM j)) = Function.update (fun j ↦ cls (L j) (hL j)) i (cls M hM)

theorem **interNum_eq_zero** {n : ℕ} (h : DiffVanish n (chiPic k F)) (hn : n ≤ Fintype.card ι) (hL : ∀ i, (L i).IsLB) : interNum k F L = 0

theorem **chiPic_twist** (hM : M.IsLB) : DiffVanish (d + 1) (chiPic k (twist F M))

theorem **chiPic_pushforward_of_isAffineHom** {Y : Scheme.{0}} [Algebra k Γ(Y, ⊤)] {f : X ⟶ Y} (hf : ∀ r : k, f.appTop (algebraMap k Γ(Y, ⊤) r) = algebraMap k Γ(X, ⊤) r) [IsAffineHom f] [F.IsQuasicoherent] : DiffVanish (d + 1) (chiPic k ((Scheme.Modules.pushforward f).obj F))

theorem **form_cls** {P : Fin d → LB X} (hP : ∀ i, (P i).IsLB) : h.form (fun i ↦ cls (P i) (hP i)) = interNum k F P

theorem **multiDiff_chiPic** (hd : d ≤ Fintype.card ι) (hL : ∀ i, (L i).IsLB) (c : Pic X) : multiDiff (fun i ↦ cls (L i) (hL i)) (chiPic k F) c = interNum k F L

theorem **interNum_twist** (hd : d ≤ Fintype.card ι) (hL : ∀ i, (L i).IsLB) (hM : M.IsLB) : interNum k (twist F M) L = interNum k F L

theorem **interNum_update_of_cls_eq_add** (hM : M.IsLB) (hA : A.IsLB) (hB : B.IsLB) (e : cls M hM = cls A hA + cls B hB) : interNum k F (Function.update L i M) = interNum k F (Function.update L i A) + interNum k F (Function.update L i B)

theorem **interNum_update_tens** (hA : A.IsLB) (hB : B.IsLB) : interNum k F (Function.update L i (tens A B)) = interNum k F (Function.update L i A) + interNum k F (Function.update L i B)

theorem **interNum_update_of_cls_eq_sum** {κ : Type*} (s : Finset κ) {P : κ → LB X} (hP : ∀ j, (P j).IsLB) (n : κ → ℤ) (hM : M.IsLB) (e : cls M hM = ∑ j ∈ s, n j • cls (P j) (hP j)) : interNum k F (Function.update L i M) = ∑ j ∈ s, n j * interNum k F (Function.update L i (P j))

theorem **interNum_update_of_cls_eq_zsmul** (hM : M.IsLB) (hA : A.IsLB) (n : ℤ) (e : cls M hM = n • cls A hA) : interNum k F (Function.update L i M) = n * interNum k F (Function.update L i A)

theorem **interNum_cons_of_cls_eq_add** {e : ℕ} (he : d ≤ e + 1) {P : Fin e → LB X} (hP : ∀ i, (P i).IsLB) (hM : M.IsLB) (hA : A.IsLB) (hB : B.IsLB) (hc : cls M hM = cls A hA + cls B hB) : interNum k F (Fin.cons M P : Fin (e + 1) → LB X) = interNum k F (Fin.cons A P : Fin (e + 1) → LB X) + interNum k F (Fin.cons B P : Fin (e + 1) → LB X)

theorem **interNum_of_cls_eq_zsmul** {P Q : Fin d → LB X} (hP : ∀ i, (P i).IsLB) (hQ : ∀ i, (Q i).IsLB) (n : Fin d → ℤ) (e : ∀ i, cls (Q i) (hQ i) = n i • cls (P i) (hP i)) : interNum k F Q = (∏ i, n i) * interNum k F P

theorem **interNum_of_cls_eq_sum** [DecidableEq ι] (hL : ∀ j, (L j).IsLB) {Q : Fin d → LB X} (hQ : ∀ i, (Q i).IsLB) (n : Fin d → ι → ℤ) (e : ∀ i, cls (Q i) (hQ i) = ∑ j, n i j • cls (L j) (hL j)) : interNum k F Q = ∑ a : Fin d → ι, (∏ i, n i (a i)) * interNum k F (L ∘ a)

theorem **exists_polynomial_chi** (hA : A.IsLB) (hB : B.IsLB) : ∃ p : Polynomial K, p.natDegree ≤ d ∧ (∀ (n : ℤ) (N : LB X) (hN : N.IsLB), cls N hN = cls A hA + n • cls B hB → p.eval (n : K) = chi k (twist F N)) ∧ p.coeff d = (d.factorial : K)⁻¹ * interNum k F fun _ : Fin d ↦ B

theorem **exists_polynomial_chi_twistZ** (hM : M.IsLB) : ∃ p : Polynomial K, p.natDegree ≤ d ∧ (∀ n : ℤ, p.eval (n : K) = chi k (twistZ F M n)) ∧ p.coeff d = (d.factorial : K)⁻¹ * interNum k F fun _ : Fin d ↦ M

theorem **existsUnique_interForm** : ∃! Φ : MultilinearMap K (fun _ : Fin d ↦ K ⊗[ℤ] Pic X) K, ∀ (P : Fin d → LB X) (hP : ∀ i, (P i).IsLB), Φ (fun i ↦ 1 ⊗ₜ cls (P i) (hP i)) = interNum k F P

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsMatricesOfProjectiveComplex

theorem **projMatrix_abstract** {S : Type} [CommRing S] {P₀ P₁ P₂ F₀ F₁ G₁ G₂ : Type} [AddCommGroup P₀] [Module S P₀] [AddCommGroup P₁] [Module S P₁] [AddCommGroup P₂] [Module S P₂] [AddCommGroup F₀] [Module S F₀] [AddCommGroup F₁] [Module S F₁] [AddCommGroup G₁] [Module S G₁] [AddCommGroup G₂] [Module S G₂] (d₀ : P₀ →ₗ[S] P₁) (d₁ : P₁ →ₗ[S] P₂) (p₀ : F₀ →ₗ[S] P₀) (s₀ : P₀ →ₗ[S] F₀) (p₁ : F₁ →ₗ[S] P₁) (s₁ : P₁ →ₗ[S] F₁) (D₀ : F₀ →ₗ[S] G₁) (D₁ : G₁ →ₗ[S] G₂) (q₁ : G₁ →ₗ[S] F₁) (q₀ : G₁ →ₗ[S] F₀) (i₁ : F₁ →ₗ[S] G₁) (i₀ : F₀ →ₗ[S] G₁) (r₂ : G₂ →ₗ[S] P₂) (r₁ : G₂ →ₗ[S] F₁) (r₀ : G₂ →ₗ[S] P₀) (h₀ : p₀ ∘ₗ s₀ = LinearMap.id) (h₁ : p₁ ∘ₗ s₁ = LinearMap.id) (hq₁ : q₁ ∘ₗ D₀ = s₁ ∘ₗ d₀ ∘ₗ p₀) (hq₀ : q₀ ∘ₗ D₀ = LinearMap.id − s₀ ∘ₗ p₀) (hi : i₁ ∘ₗ q₁ + i₀ ∘ₗ q₀ = LinearMap.id) (hr₂ : r₂ ∘ₗ D₁ = d₁ ∘ₗ p₁ ∘ₗ q₁) (hr₁ : r₁ ∘ₗ D₁ = q₁ − s₁ ∘ₗ p₁ ∘ₗ q₁) (hr₀ : r₀ ∘ₗ D₁ = p₀ ∘ₗ q₀) : Set.BijOn p₀ {v | D₀ v =

theorem **projMatrix_mulVec** {R : Type} [CommRing R] (R' : Type) [CommRing R'] [Algebra R R'] {a b : ℕ} (f : (Fin a → R) →ₗ[R] (Fin b → R)) (x : R' ⊗[R] (Fin a → R)) : ((LinearMap.toMatrix' f).map (algebraMap R R')).mulVec (TensorProduct.piScalarRight R R' R' (Fin a) x) = TensorProduct.piScalarRight R R' R' (Fin b) (f.baseChange R' x)

theorem **projMatrix_sum** {R : Type} [CommRing R] (R' : Type) [CommRing R'] [Algebra R R'] {a : ℕ} {P : Type} [AddCommGroup P] [Module R P] (p : (Fin a → R) →ₗ[R] P) (x : R' ⊗[R] (Fin a → R)) : ∑ k, TensorProduct.piScalarRight R R' R' (Fin a) x k ⊗ₜ[R] p (Pi.single k 1) = p.baseChange R' x

theorem **exists_matrices_of_projective_complex** : ∀ (R : Type) [CommRing R] (P₀ P₁ P₂ : Type) [AddCommGroup P₀] [Module R P₀] [AddCommGroup P₁] [Module R P₁] [AddCommGroup P₂] [Module R P₂] [Module.Finite R P₀] [Module.Projective R P₀] [Module.Finite R P₁] [Module.Projective R P₁] [Module.Finite R P₂] [Module.Projective R P₂] (d₀ : P₀ →ₗ[R] P₁) (d₁ : P₁ →ₗ[R] P₂), d₁ ∘ₗ d₀ = 0 → ∃ (a b c : ℕ) (D₀ : Matrix (Fin b) (Fin a) R) (D₁ : Matrix (Fin c) (Fin b) R) (u : Fin a → P₀), D₁ * D₀ = 0 ∧ ∀ (R' : Type) [CommRing R'] [Algebra R R'], Set.BijOn (fun v : Fin a → R' => ∑ k, v k ⊗ₜ[R] u k) {v | (D₀.map (algebraMap R R')).mulVec v = 0} (LinearMap.ker (d₀.baseChange R')) ∧ (Function.Exact (d₀.baseChange R') (d₁.baseChange R') → ∀ w : Fin b → R', (D₁.map (algebraMap R R')).mulVec w = 0 → ∃ v, (D₀.map (algebraMap R R')).mulVec v = w)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsMonomialBasisCechCubePolynomialGrading

theorem **exists_xProd_pow_smul_monomial** {ι : Type} [DecidableEq ι] (R : Type) [CommRing R] (s : Finset ι) (j : ℕ) (m : ι →₀ ℕ) : ∃ m' : ι →₀ ℕ, (∀ i, m' i = m i + if i ∈ s then j else 0) ∧ xProd R s ^ j • (monomial m 1 : MvPolynomial ι R) = monomial m' 1

theorem **awayFrac_monomial_congr** {ι : Type} [DecidableEq ι] (R : Type) [CommRing R] (s : Finset ι) (a : ι → ℤ) {k j : ℕ} {m m' : ι →₀ ℕ} (hm : ∀ i, (m i : ℤ) = a i + if i ∈ s then (k : ℤ) else 0) (hm' : ∀ i, (m' i : ℤ) = a i + if i ∈ s then (j : ℤ) else 0) : awayFrac (MvPolynomial ι R) (xProd R s) k (monomial m 1) = awayFrac (MvPolynomial ι R) (xProd R s) j (monomial m' 1)

theorem **exists_laurent_numerator** {ι : Type} [DecidableEq ι] [Fintype ι] (s : Finset ι) (a : ι → ℤ) (ha : ∀ i, i ∉ s → 0 ≤ a i) : ∃ (k : ℕ) (m : ι →₀ ℕ), ∀ i, (m i : ℤ) = a i + if i ∈ s then (k : ℤ) else 0

theorem **exists_basis_grCube_polyGrading** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (n : ℤ), ∃ b : ∀ s : Finset ι, Module.Basis {a : {a : ι → ℤ // ∑ i, a i = n} // Finset.univ.filter (fun i => a.1 i < 0) ⊆ s} R ((grCube ι (polyGrading ι R) n).obj s), ∀ (s t : Finset ι) (h : s ⊆ t) (a : {a : {a : ι → ℤ // ∑ i, a i = n} // Finset.univ.filter (fun i => a.1 i < 0) ⊆ s}), Cube.res (grCube ι (polyGrading ι R) n) h (b s a) = b t (a.1, a.2.trans h)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsMonomialBasisCechCubePolynomialGradingStatement

theorem **_root_.GlobalTheory.T.exists_monomial_basis_cechCube_polynomialGrading** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (n : ℤ), ∃ b : ∀ s : Finset ι, Module.Basis {a : {a : ι → ℤ // ∑ i, a i = n} // Finset.univ.filter (fun i => a.1 i < 0) ⊆ s} R ((grCube ι (polyGrading ι R) n).obj s), ∀ (s t : Finset ι) (h : s ⊆ t) (a : {a : {a : ι → ℤ // ∑ i, a i = n} // Finset.univ.filter (fun i => a.1 i < 0) ⊆ s}), Cube.res (grCube ι (polyGrading ι R) n) h (b s a) = b t (a.1, a.2.trans h)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsMvPolynomialOfIteratedBackwardDifferenceVa

theorem **bwdDiff_apply** (g : G) (φ : G → M) (x : G) : bwdDiff g φ x = φ x − φ (x − g)

theorem **multiDiff_apply** {ι : Type*} [Fintype ι] (g : ι → G) (φ : G → M) (x : G) : multiDiff g φ x = ∑ S : Finset ι, (-1 : ℤ) ^ S.card • φ (x − ∑ i ∈ S, g i)

theorem **shiftAct_of** (g : G) (φ : G → M) (x : G) : shiftAct G M (AddMonoidAlgebra.of' ℤ G g) φ x = φ (x − g)

theorem **shiftAct_diffElt** (g : G) (φ : G → M) : shiftAct G M (diffElt g) φ = bwdDiff g φ

theorem **of'_add** (a b : G) : AddMonoidAlgebra.of' ℤ G (a + b) = AddMonoidAlgebra.of' ℤ G a * AddMonoidAlgebra.of' ℤ G b

theorem **of'_sum** {ι : Type*} (s : Finset ι) (g : ι → G) : AddMonoidAlgebra.of' ℤ G (∑ i ∈ s, g i) = ∏ i ∈ s, AddMonoidAlgebra.of' ℤ G (g i)

theorem **diffElt_zero** : diffElt (0 : G) = 0

theorem **diffElt_add** (a b : G) : diffElt (a + b) = diffElt a + diffElt b − diffElt a * diffElt b

theorem **diffElt_neg** (a : G) : diffElt (-a) = -(AddMonoidAlgebra.of' ℤ G (-a)) * diffElt a

theorem **multiDiff_eq_shiftAct** {ι : Type*} [Fintype ι] (g : ι → G) (φ : G → M) : multiDiff g φ = shiftAct G M (∏ i, diffElt (g i)) φ

@[simp] theorem **multiDiff_of_isEmpty** {ι : Type*} [Fintype ι] [IsEmpty ι] (g : ι → G) (φ : G → M) : multiDiff g φ = φ

theorem **multiDiff_cons** {d : ℕ} (a : G) (g : Fin d → G) (φ : G → M) : multiDiff (Fin.cons a g : Fin (d + 1) → G) φ = bwdDiff a (multiDiff g φ)

theorem **multiDiff_cons'** {d : ℕ} (a : G) (g : Fin d → G) (φ : G → M) : multiDiff (Fin.cons a g : Fin (d + 1) → G) φ = multiDiff g (bwdDiff a φ)

theorem **multiDiff_const** {d : ℕ} (a : G) (φ : G → M) : multiDiff (fun _ : Fin d ↦ a) φ = (bwdDiff a)^[d] φ

theorem **multiDiff_comp_equiv** {ι κ : Type*} [Fintype ι] [Fintype κ] (e : κ ≃ ι) (g : ι → G) (φ : G → M) : multiDiff (g ∘ e) φ = multiDiff g φ

theorem **multiDiff_add** {ι : Type*} [Fintype ι] (g : ι → G) (φ ψ : G → M) : multiDiff g (φ + ψ) = multiDiff g φ + multiDiff g ψ

theorem **multiDiff_sub** {ι : Type*} [Fintype ι] (g : ι → G) (φ ψ : G → M) : multiDiff g (φ − ψ) = multiDiff g φ − multiDiff g ψ

theorem **multiDiff_map** {ι N : Type*} [Fintype ι] [AddCommGroup N] (f : M →+ N) (g : ι → G) (φ : G → M) : multiDiff g (f ∘ φ) = f ∘ multiDiff g φ

theorem **multiDiff_comap** {ι H : Type*} [Fintype ι] [AddCommGroup H] (f : H →+ G) (g : ι → H) (φ : G → M) : multiDiff g (φ ∘ f) = multiDiff (f ∘ g) φ ∘ f

theorem **multiDiff_translate** {ι : Type*} [Fintype ι] (g : ι → G) (φ : G → M) (a x : G) : multiDiff g (fun y ↦ φ (a + y)) x = multiDiff g φ (a + x)

theorem **diffElt_mem** (g : G) : diffElt g ∈ diffIdeal G

theorem **span_diffElt_image** {S : Set G} (hS : AddSubgroup.closure S = ⊤) : Ideal.span (diffElt '' S) = diffIdeal G

theorem **prod_diffElt_mem** {ι : Type*} (s : Finset ι) (g : ι → G) : ∏ i ∈ s, diffElt (g i) ∈ diffIdeal G ^ s.card

theorem **shiftAct_eq_zero_of_generators** {S : Set G} (hS : AddSubgroup.closure S = ⊤) {n : ℕ} {φ : G → M} (h : ∀ g : Fin n → G, (∀ i, g i ∈ S) → multiDiff g φ = 0) {r : AddMonoidAlgebra ℤ G} (hr : r ∈ diffIdeal G ^ n) : shiftAct G M r φ = 0

theorem **diffVanish_of_generators** {S : Set G} (hS : AddSubgroup.closure S = ⊤) {n : ℕ} {φ : G → M} (h : ∀ g : Fin n → G, (∀ i, g i ∈ S) → multiDiff g φ = 0) : DiffVanish n φ

theorem **DiffVanish.shiftAct_eq_zero** {n : ℕ} {φ : G → M} (h : DiffVanish n φ) {r : AddMonoidAlgebra ℤ G} (hr : r ∈ diffIdeal G ^ n) : shiftAct G M r φ = 0

theorem **DiffVanish.multiDiff_eq_zero** {n : ℕ} {φ : G → M} (h : DiffVanish n φ) {ι : Type*} [Fintype ι] (hn : n ≤ Fintype.card ι) (g : ι → G) : multiDiff g φ = 0

theorem **DiffVanish.mono** {m n : ℕ} {φ : G → M} (h : DiffVanish m φ) (hmn : m ≤ n) : DiffVanish n φ

theorem **diffVanish_zero_iff** {φ : G → M} : DiffVanish 0 φ ↔ φ = 0

theorem **diffVanish_succ_iff** {n : ℕ} {φ : G → M} : DiffVanish (n + 1) φ ↔ ∀ a, DiffVanish n (bwdDiff a φ)

theorem **DiffVanish.map** {N : Type*} [AddCommGroup N] {n : ℕ} {φ : G → M} (h : DiffVanish n φ) (f : M →+ N) : DiffVanish n (f ∘ φ)

theorem **DiffVanish.comap** {H : Type*} [AddCommGroup H] {n : ℕ} {φ : G → M} (h : DiffVanish n φ) (f : H →+ G) : DiffVanish n (φ ∘ f)

theorem **DiffVanish.translate** {n : ℕ} {φ : G → M} (h : DiffVanish n φ) (a : G) : DiffVanish n (fun y ↦ φ (a + y))

theorem **DiffVanish.multiDiff_eq** {d : ℕ} {φ : G → M} (h : DiffVanish (d + 1) φ) (g : Fin d → G) (x y : G) : multiDiff g φ x = multiDiff g φ y

theorem **DiffVanish.multiDiff_update_add** {d : ℕ} {φ : G → M} (h : DiffVanish (d + 1) φ) (g : Fin d → G) (i : Fin d) (a b : G) : multiDiff (Function.update g i (a + b)) φ = multiDiff (Function.update g i a) φ + multiDiff (Function.update g i b) φ

theorem **DiffVanish.form_apply** {d : ℕ} {φ : G → M} (h : DiffVanish (d + 1) φ) (g : Fin d → G) (x : G) : h.form g = multiDiff g φ x

theorem **DiffVanish.form_comp_perm** {d : ℕ} {φ : G → M} (h : DiffVanish (d + 1) φ) (g : Fin d → G) (σ : Equiv.Perm (Fin d)) : h.form (g ∘ σ) = h.form g

theorem **multiDiff_polynomial_eval** {R : Type*} [CommRing R] {d : ℕ} (z : Fin d → R) (p : Polynomial R) (hp : p.natDegree ≤ d) (y : R) : multiDiff z (fun t ↦ p.eval t) y = d.factorial * p.coeff d * ∏ i, z i

theorem **diffVanish_polynomial_eval** {R : Type*} [CommRing R] {d : ℕ} (p : Polynomial R) (hp : p.natDegree ≤ d) : DiffVanish (d + 1) (fun t ↦ p.eval t)

theorem **multiDiff_pow** {R : Type*} [CommRing R] {d : ℕ} (z : Fin d → R) (y : R) : multiDiff z (fun t ↦ t ^ d) y = d.factorial * ∏ i, z i

theorem **pow_sub_pow_mem_pow_succ** {R : Type*} [CommRing R] {I : Ideal R} {a b : R} (ha : a ∈ I) (hb : b ∈ I) (hab : a - b ∈ I ^ 2) (d : ℕ) : a ^ d - b ^ d ∈ I ^ (d + 1)

theorem **diffElt_sub_sum_mem** {ι : Type*} (s : Finset ι) (g : ι → G) (x : G) : diffElt (x - ∑ i ∈ s, g i) - (diffElt x - ∑ i ∈ s, diffElt (g i)) ∈ diffIdeal G ^ 2

theorem **DiffVanish.multiDiff_form_diag** {d : ℕ} {φ : G → M} (h : DiffVanish (d + 1) φ) (g : Fin d → G) (x : G) : multiDiff g (fun y ↦ h.form fun _ ↦ y) x = d.factorial • h.form g

theorem **DiffVanish.sub_form_diag** {d : ℕ} {φ : G → K} (h : DiffVanish (d + 1) φ) : DiffVanish d (φ - fun y ↦ (d.factorial : K)⁻¹ * h.form fun _ ↦ y)

theorem **isHomogeneous_diffTopPoly** (d : ℕ) {ι : Type*} [Fintype ι] [DecidableEq ι] (g : ι → G) (φ : G → K) (x : G) : (diffTopPoly d g φ x).IsHomogeneous d

theorem **DiffVanish.eval_diffTopPoly** {d : ℕ} {φ : G → K} (h : DiffVanish (d + 1) φ) {ι : Type*} [Fintype ι] [DecidableEq ι] (g : ι → G) (x : G) (n : ι → ℤ) : MvPolynomial.eval (fun i ↦ (n i : K)) (diffTopPoly d g φ x) = (d.factorial : K)⁻¹ * h.form fun _ ↦ ∑ i, n i • g i

theorem **exists_mvPolynomial_of_iteratedBackwardDifferenceVanish** {ι : Type*} [Fintype ι] [DecidableEq ι] (g : ι → G) (d : ℕ) : ∀ φ : G → K, DiffVanish (d + 1) φ → ∃ P : MvPolynomial ι K, P.totalDegree ≤ d ∧ (∀ n : ι → ℤ, P.eval (fun i ↦ (n i : K)) = φ (∑ i, n i • g i)) ∧ MvPolynomial.homogeneousComponent d P = diffTopPoly d g φ 0

theorem **DiffVanish.exists_mvPolynomial** {d : ℕ} {φ : G → K} (h : DiffVanish (d + 1) φ) {ι : Type*} [Fintype ι] [DecidableEq ι] (g : ι → G) (x : G) : ∃ P : MvPolynomial ι K, P.totalDegree ≤ d ∧ (∀ n : ι → ℤ, P.eval (fun i ↦ (n i : K)) = φ (x + ∑ i, n i • g i)) ∧ MvPolynomial.homogeneousComponent d P = diffTopPoly d g φ x ∧ ∀ n : ι → ℤ, (MvPolynomial.homogeneousComponent d P).eval (fun i ↦ (n i : K)) = (d.factorial : K)⁻¹ * h.form fun _ ↦ ∑ i, n i • g i

theorem **DiffVanish.exists_polynomial** {d : ℕ} {φ : G → K} (h : DiffVanish (d + 1) φ) (a x : G) : ∃ p : Polynomial K, p.natDegree ≤ d ∧ (∀ n : ℤ, p.eval (n : K) = φ (x + n • a)) ∧ p.coeff d = (d.factorial : K)⁻¹ * multiDiff (fun _ : Fin d ↦ a) φ x

theorem **DiffVanish.exists_polynomial_int** {d : ℕ} {φ : G → ℤ} (h : DiffVanish (d + 1) φ) (K : Type*) [Field K] [CharZero K] (a x : G) : ∃ p : Polynomial K, p.natDegree ≤ d ∧ (∀ n : ℤ, p.eval (n : K) = φ (x + n • a)) ∧ p.coeff d = (d.factorial : K)⁻¹ * multiDiff (fun _ : Fin d ↦ a) φ x

theorem **diffVanish_pow_int** (d : ℕ) : DiffVanish (d + 1) (fun t : ℤ ↦ t ^ d) ∧ ¬ DiffVanish d (fun t : ℤ ↦ t ^ d)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsNhdsPowMulTwCoordEqZero

theorem **exists_nhds_pow_mul_twCoord_eq_zero** {Y : Scheme.{0}} {H M : LB Y} {d : ℤ} (σ : LB.Sec (pw H d)) {t : ℤ} (hH : H.IsLB) (hM : M.IsLB) (a : SecOn (tw M H t) ⊤) (ha : SecOn.restrict _ (le_top : nonzeroLocus σ ≤ ⊤) a = 0) (x : Y) : ∃ (W : Y.Opens) (n : ℕ), x ∈ W ∧ ∀ (p : M.ι) (l : H.ι) (O : Y.Opens) (_ : O ≤ W) (h : O ≤ ⊤ ⊓ (M.U p ⊓ H.U l)), rs Y (h.trans (inf_le_right.trans inf_le_right) : O ≤ H.U l) (σ.f l) ^ n * rs Y h (SecOn.twCoord a p l) = 0

theorem **exists_pow_mul_twCoord_eq_zero** : ∀ (Y : Scheme.{0}) (H M : LB Y) (d : ℤ) (σ : LB.Sec (pw H d)) (V : Y.Opens) (t : ℤ), H.IsLB → M.IsLB → V = nonzeroLocus σ → CompactSpace Y → ∀ a : SecOn (tw M H t) ⊤, SecOn.restrict _ (le_top : V ≤ ⊤) a = 0 → ∃ k : ℕ, ∀ (p : M.ι) (l : H.ι), rs Y (inf_le_right.trans inf_le_right : ⊤ ⊓ (M.U p ⊓ H.U l) ≤ H.U l) (σ.f l) ^ k * SecOn.twCoord a p l = 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsNoetherianSubringHomogeneousMatrixFlatChar

private theorem **exists_generalized_inverse_mod_ideal** {R S B P : Type} [CommRing R] [CommRing S] [Algebra R S] [CommRing B] [Algebra R B] [AddCommGroup P] [Module B P] {a b r : ℕ} (Φ : Matrix (Fin a) (Fin b) S) (u : S) (g : Fin r → S) (π : S →ₐ[R] B) (hker : RingHom.ker (π : S →+* B) = Ideal.span (Set.range g)) (hπ : Function.Surjective π) (ε : (Fin a → S) →ₗ[(π : S →+* B)] P) (hproj : Module.Projective B P) (hε : Function.Surjective ε) (hkerε : ∀ y, ε y = 0 ↔ ∃ w w', y = Φ.mulVec w + u • w') (hu : π u = 0) : ∃ (Ψ : Matrix (Fin b) (Fin a) S), Φ * Ψ * Φ = Φ + ∑ k, g k • E k

private theorem **chart_witnesses_lemma** {ι : Type} [Fintype ι] {R : Type} [CommRing R] {a b : ℕ} (Φ : Matrix (Fin a) (Fin b) (MvPolynomial ι R)) (i : ι) (H : ∃ (B : Type) (_ : CommRing B) (_ : Algebra R B) (π : MvPolynomial ι R →ₐ[R] B) (P : Type) (_ : AddCommGroup P) (_ : Module B P) (ε : (Fin a → MvPolynomial ι R) →ₗ[(π : MvPolynomial ι R →+* B)] P), Algebra.Smooth R B ∧ Module.Projective B P ∧ Function.Surjective π ∧ π (X i) = 1 ∧ Function.Surjective ε ∧ ∀ y, ε y = 0 ↔ ∃ w w', y = Φ.mulVec w + (X i - 1 : MvPolynomial ι R) • w') : ∃ (r : ℕ) (g : Fin r → MvPolynomial ι R), Algebra.FormallySmooth R (MvPolynomial ι R ⧸ Ideal.span (Set.range

g)) ∧ (∃ c : Fin r → MvPolynomial ι R, (X i - 1 : MvPolynomial ι R) = ∑ k, c k * g k) ∧ (∀ k, ∃ (C : Matrix (Fin b) (Fin a) (MvPolynomial ι R)) (C' : Matrix (Fin a) (Fin a) (MvPolynomial ι R)), g k • (1 : Matrix (Fin a) (Fin a) (MvPolynomial ι R)) = Φ *

private theorem **formallySmooth_quotient_of_aeval_shift_mem_sq** {ι R : Type} [CommRing R] {r : ℕ} (g : Fin r → MvPolynomial ι R) (a : ι → Fin r → MvPolynomial ι R) (b : Fin r → Fin r → Fin r → MvPolynomial ι R) (h : ∀ j, aeval (fun i => X i + ∑ k, a i k * g k) (g j) = ∑ k, ∑ l, b j k l * (g k * g l)) : Algebra.FormallySmooth R (MvPolynomial ι R ⧸ Ideal.span (Set.range g))

theorem **_root_.GlobalTheory.exists_noetherian_subring_homogeneous_matrix_flat_charts** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (a b : ℕ) (d : Fin a → ℤ) (e : Fin b → ℤ) (Φ : Matrix (Fin a) (Fin b) (MvPolynomial ι R)), (∀ l j, Φ l j ∈ _root_.GlobalTheory.polyGrading ι R (e j - d l)) → (∀ i : ι, ∃ (B : Type) (_ : CommRing B) (_ : Algebra R B) (π : MvPolynomial ι R →ₐ[R] B) (P : Type) (_ : AddCommGroup P) (_ : Module B P) (ε : (Fin a → MvPolynomial ι R) →ₛₗ[(π : MvPolynomial ι R →+* B)] P), Algebra.Smooth R B ∧ Module.Projective B P ∧ Function.Surjective π ∧ π (X i) = 1 ∧ Function.Surjective ε ∧ ∀ y, ε y = 0 ↔ ∃ w w', y = Φ.mulVec w + (X i - 1 : MvPolynomial ι R) • w') → ∃ (R₀ : Subring R) (Φ₀ : Matrix (Fin a) (Fin b) (MvPolynomial ι R₀)), IsNoetherianRing R₀ ∧ Φ₀.map (MvPolynomial.map R₀.subtype) = Φ ∧ (∀ l j, Φ₀ l j ∈ _root_.GlobalTheory.polyGrading ι R₀ (e j - d

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsNonzeroLocusEqAffineOpenLocallyFactorial

theorem **effCartier_of_stalkId_of_isLocallyNoetherian** {W : Scheme.{0}} [IsLocallyNoetherian W] (G : W.IdealSheafData) (hG : ∀ x : W, ∃ t ∈ nonZeroDivisors (W.presheaf.stalk x), FinalTheorems.stalkId G x = Ideal.span {t}) : FinalTheorems.EffCartier G

theorem **exists_radical_span_singleton_eq** {A : Type} [CommRing A] [IsDomain A] [UniqueFactorizationMonoid A] {a : A} (ha : a ≠ 0) : ∃ t ∈ nonZeroDivisors A, (Ideal.span {a}).radical = Ideal.span {t}

theorem **stalkId_vanishingIdeal_compl** {X : Scheme.{0}} (U : X.Opens) (x : X) (a : X.presheaf.stalk x) (h : ∀ q : PrimeSpectrum (X.presheaf.stalk x), X.fromSpecStalk x q ∈ U ↔ a ∉ q.asIdeal) : FinalTheorems.stalkId (Scheme.IdealSheafData.vanishingIdeal U.compl) x = (Ideal.span {a}).radical

theorem **exists_nonzeroLocus_eq** : ∀ (X : Scheme.{0}) [IsIntegral X] [IsLocallyNoetherian X] [X.IsSeparated] (U : X.Opens), IsAffineOpen U → (U : Set X).Nonempty → (∀ x : X, UniqueFactorizationMonoid (X.presheaf.stalk x)) → ∃ (L : LB X) (σ : LB.Sec L), L.IsLB ∧ nonzeroLocus σ = U

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsNonzeroLocusEqAffineOpenLocallyFactorialStatement

theorem **_root_.GlobalTheory.T.exists_nonzeroLocus_eq_affineOpen_locallyFactorial** : ∀ (X : Scheme.{0}) [IsIntegral X] [IsLocallyNoetherian X] [X.IsSeparated] (U : X.Opens), IsAffineOpen U → (U : Set X).Nonempty → (∀ x : X, UniqueFactorizationMonoid (X.presheaf.stalk x)) → ∃ (L : LB X) (σ : LB.Sec L), L.IsLB ∧ nonzeroLocus σ = U

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsPowMulMemRangeOfLocalDenominators

theorem **exists_dvd_pow_of_forall_prime_dvd** {A : Type} [CommRing A] [IsDomain A] [UniqueFactorizationMonoid A] (a y : A) : y ≠ 0 → (∀ q, Prime q → q ∣ y → q ∣ a) → ∃ n : ℕ, y ∣ a ^ n

theorem **exists_pow_mul_mem_range_of_local_denominators** {A B : Type} [CommRing A] [IsDomain A] [UniqueFactorizationMonoid A] [CommRing B] [IsDomain B] (φ : A →+* B) (hφ : Function.Injective φ) (W : Set (PrimeSpectrum A)) (hW : W.Nonempty) (a : A) (ha : ∀ (q : A) (p : PrimeSpectrum A), Prime q → p.asIdeal = Ideal.span {q} → p ∉ W → q ∣ a) (v : B) (hv : ∀ p ∈ W, ∃ s, s ∉ p.asIdeal ∧ ∃ x, φ x = φ s * v) : ∃ (n : ℕ) (x : A), φ x = φ a ^ n * v

theorem **exists_eq_basicOpen_of_isAffineOpen_of_uniqueFactorization** : ∀ (A : Type) [CommRing A] [IsDomain A] [IsNoetherianRing A] [UniqueFactorizationMonoid A] (U : (Spec (CommRingCat.of A)).Opens), IsAffineOpen U → (U : Set (Spec (CommRingCat.of A))).Nonempty → ∃ a : A, U = PrimeSpectrum.basicOpen a

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsPowSmulEqAddSubOneSmul

theorem **exists_pow_smul_eq_add_sub_one_smul** {S M : Type} [CommRing S] [AddCommGroup M] [Module S M] (x : S) (k : ℕ) (y : M) : ∃ w : M, x ^ k • y = y + (x - 1) • w

private lemma **comap_ker_inf_smul_whole_eq_smul_whole_of_regular** {S F G : Type} [CommRing S] [AddCommGroup F] [Module S F] [AddCommGroup G] [Module S G] (u : S) (φ : F →ₗ[S] G) (hreg : ∀ z : G, u • z = 0 → z = 0) : Submodule.comap (LinearMap.ker φ).subtype (LinearMap.ker φ ⊓ u • (⊤ : Submodule S F)) = u • (⊤ : Submodule S (LinearMap.ker φ))

private lemma **ker_sup_smul_whole_eq_ker_mkQ_comp** {S F G : Type} [CommRing S] [AddCommGroup F] [Module S F] [AddCommGroup G] [Module S G] (u : S) (φ : F →ₗ[S] G) (hker : ∀ (y : F) (w : G), φ y = u • w → ∃ (k v : F), φ k = 0 ∧ y = k + u • v) : LinearMap.ker φ ⊔ u • (⊤ : Submodule S F) = LinearMap.ker ((u • (⊤ : Submodule S G)).mkQ ∘ₗ φ)

theorem **finite_quotSMulTop_ker_of_regular** {S F G : Type} [CommRing S] [AddCommGroup F] [Module S F] [AddCommGroup G] [Module S G] (u : S) (φ : F →ₗ[S] G) (hreg : ∀ z : G, u • z = 0 → z = 0) [Module.Finite S F] [Module.FinitePresentation S (QuotSMulTop u G)] (hsurj : ∀ z : G, ∃ (y : F) (w : G), φ y = z + u • w) (hker : ∀ (y : F) (w : G), φ y = u • w → ∃ (k v : F), φ k = 0 ∧ y = k + u • v) : Module.Finite S (QuotSMulTop u (LinearMap.ker φ))

theorem **finite_quotSMulTop_ker_of_graded** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (F : Type) [AddCommGroup F] [Module R F] [Module (MvPolynomial ι R) F] [IsScalarTower R (MvPolynomial ι R) F] (𝒜 : ℤ → Submodule R F) [PolyGradedSMul ι 𝒜] [Decomposition 𝒜] (G : Type) [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (ℬ : ℤ → Submodule R G) [PolyGradedSMul ι ℬ] [Decomposition ℬ] (φ : F →ₗ[MvPolynomial ι R] G) (i : ι), IsGradedMap 𝒜 ℬ φ → Module.Finite (MvPolynomial ι R) F → (∀ z : G, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • z ∈ LinearMap.range φ) → Module.FinitePresentation (MvPolynomial ι R) (QuotSMulTop (X i - 1 : MvPolynomial ι R) G) → Module.Finite (MvPolynomial ι R) (QuotSMulTop (X i - 1 : MvPolynomial ι R) (LinearMap.ker φ))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSecOnRTensorEquiv

theorem **exists_tensor_addEquiv** {R A B C P Q N N' : Type*} [CommRing R] [CommRing A] [CommRing B] [CommRing C] [Algebra R A] [Algebra R B] [Algebra R C] [Algebra A C] [Algebra B C] [IsScalarTower R A C] [IsScalarTower R B C] [Algebra.IsPushout R B A C] [AddCommGroup P] [Module R P] [Module A P] [IsScalarTower R A P] [AddCommGroup Q] [Module R Q] [Module B Q] [IsScalarTower R B Q] [AddCommGroup N] [Module A N] [Module C N] [IsScalarTower A C N] [AddCommGroup N'] [Module B N'] [Module C N'] [IsScalarTower B C N'] {j : P →ₗ[A] N} {j' : Q →ₗ[B] N'} (hj : IsBaseChange C j) (hj' : IsBaseChange C j') : ∃ E : P ⊗[R] Q ≃+ N ⊗[C] N', (∀ x y, E (x ⊗ₜ y) = j x ⊗ₜ j' y) ∧ ∀ (r : R) z, E (r • z) = algebraMap R C r • E z

@[simp] theorem **twCoord_mul** (s : SecOn M V) (t : SecOn H V) (p : M.ι) (k : H.ι) : twCoord (mul s t) p k = rs X (inf_le_inf_left V inf_le_left) (coord s p) * rs X (inf_le_inf_left V inf_le_right) (coord t k)

theorem **restrict_mul** (h : W ≤ V) (s : SecOn M V) (t : SecOn H V) : restrict _ h (mul s t) = mul (restrict M h s) (restrict H h t)

@[simp] theorem **secMulHom_tmul** (s : SecOn M V) (t : SecOn H V) : secMulHom M H V (s ⊗ₜ t) = mul s t

theorem **exists_mul_eq** (hM : M.IsLB) (hH : H.IsLB) (p : M.ι) (k : H.ι) (hp : V ≤ M.U p) (hk : V ≤ H.U k) (x : SecOn (tw M H 1) V) : ∃ s t, mul s t = x

theorem **secMulHom_bijective** (hM : M.IsLB) (hH : H.IsLB) (hV : IsAffineOpen V) : Function.Bijective (secMulHom M H V)

theorem **isAffineOpen_inf** (H : IsPullback p₁ p₂ f f') {U : Y.Opens} {W : Y'.Opens} (hU : IsAffineOpen U) (hW : IsAffineOpen W) : IsAffineOpen (p₁ ⁻¹ᵁ U ⊓ p₂ ⁻¹ᵁ W)

theorem **isPushout_prod** (H : IsPullback p₁ p₂ f f') {U : Y.Opens} {W : Y'.Opens} {O : Z.Opens} (e₁ : O ≤ p₁ ⁻¹ᵁ U) (e₂ : O ≤ p₂ ⁻¹ᵁ W) (hU : IsAffineOpen U) (hW : IsAffineOpen W) (e : O = p₁ ⁻¹ᵁ U ⊓ p₂ ⁻¹ᵁ W) : IsPushout (CommRingCat.ofHom (cst f' W)) (CommRingCat.ofHom (cst f U)) (p₂.appLE W O e₂) (p₁.appLE U O e₁)

theorem **exists_secOnR_tensor_equiv** : ∀ (R : Type) [CommRing R] (Y Y' Z : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (f' : Y' ⟶ Spec (CommRingCat.of R)) (p₁ : Z ⟶ Y) (p₂ : Z ⟶ Y') (M : LB Y) (M' : LB Y'), IsPullback p₁ p₂ f f' → M.IsLB → M'.IsLB → ∀ (U : Y.Opens) (W : Y'.Opens) (O : Z.Opens) (e₁ : O ≤ p₁ ⁻¹ᵁ U) (e₂ : O ≤ p₂ ⁻¹ᵁ W), IsAffineOpen U → IsAffineOpen W → O = p₁ ⁻¹ᵁ U ⊓ p₂ ⁻¹ᵁ W → ∃ E : SecOnR f M U ⊗[R] SecOnR f' M' W ≃ₗ[R] SecOnR (p₁ ≫ f) (tw (M.pull p₁) (M'.pull p₂) 1) O, ∀ (a : SecOn M U) (b : SecOn M' W) (p : (M.pull p₁).ι) (l : (M'.pull p₂).ι), SecOn.twCoord ((SecOnR.of _ _ O).symm (E (SecOnR.of f M U a ⊗ₜ SecOnR.of f' M' W b))) p l = rs Z (inf_le_inf_left O inf_le_left) (GlobalTheory.LB.coord (SecOn.pullLE M p₁ U O e₁ a) p) ∗ rs Z (inf_le_inf_left O inf_le_right) (GlobalTheory.LB.coord (SecOn.pullLE M' p₂ W O e₂ b) l)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSectionLiftFibreNoFirstCohomology

theorem **id_appLE** {Y : Scheme.{0}} {U V : Y.Opens} (e : V ≤ (𝟙 Y : Y ⟶ Y) ⁻¹ᵁ U) (a : Γ(Y, U)) : ((𝟙 Y : Y ⟶ Y).appLE U V e).hom a = rs Y e a

theorem **exists_sec_of_ker** {R : Type} [CommRing R] {Y : Scheme.{0}} {f : Y ⟶ Spec (CommRingCat.of R)} {M : LB Y} (P : Perf f M) (v : Fin P.a → R) (hv : P.D₀.mulVec v = 0) : ∃ s : LB.Sec M, ∀ i, rs Y (P.le i) (s.f (P.ch i)) = Seesaw.loc₀ P v i

theorem **rs_topEquiv_smul** {R : Type} [CommRing R] {Y : Scheme.{0}} (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (c : R) (x : SecOnR f M ⊤) (j : M.ι) {V : Y.Opens} (h : V ≤ M.U j) : rs Y h ((SecOnR.topEquiv f M (c • x)).f j) = cst f V c ∗ rs Y h ((SecOnR.topEquiv f M x).f j)

theorem **exists_common_denominator** {R K : Type} [CommRing R] [Field K] (g : R →+* K) (hg : ∀ y : K, ∃ a b : R, g b ≠ 0 ∧ y ∗ g b = g a) {n : ℕ} (z : Fin n → K) : ∃ (c : R) (w : Fin n → R), g c ≠ 0 ∧ g ∘ w = g c • z

theorem **exists_pull_eq_smul** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y), Flat f → IsProjBy f → M.IsLB → ∀ (K : Type) [Field K] (g : R →+* K), (∀ y : K, ∃ a b : R, g b ≠ 0 ∧ y ∗ g b = g a) → ∀ (Y₀ : Scheme.{0}) (π : Y₀ ⟶ Y) (f₀ : Y₀ ⟶ Spec (CommRingCat.of K)) (h : IsPullback π f₀ f (Spec.map (CommRingCat.ofHom g))), FirstCohomologyVanishes (M.pull π) → ∀ x₀ : SecOnR f₀ (M.pull π) ⊤, ∃ (c : R) (x : SecOnR f M ⊤), g c ≠ 0 ∧ SecOnR.pull M h.w ⊤ ⊤ le_top x = g c • x₀

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSectionLiftFibreNoFirstCohomologyStatement

theorem **_root_.GlobalTheory.T.exists_section_lift_fibre_noFirstCohomology** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y), Flat f → IsProjBy f → M.IsLB → ∀ (K : Type) [Field K] (g : R →+* K), (∀ y : K, ∃ a b : R, g b ≠ 0 ∧ y ∗ g b = g a) → ∀ (Y₀ : Scheme.{0}) (π : Y₀ ⟶ Y) (f₀ : Y₀ ⟶ Spec (CommRingCat.of K)) (h : IsPullback π f₀ f (Spec.map (CommRingCat.ofHom g))), FirstCohomologyVanishes (M.pull π) → ∀ x₀ : SecOnR f₀ (M.pull π) ⊤, ∃ (c : R) (x : SecOnR f M ⊤), g c ≠ 0 ∧ SecOnR.pull M h.w ⊤ ⊤ le_top x = g c • x₀

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSmulEqMapOfIsCompactOfIsQuasiSeparated

theorem **exists_smul_eq_zero_of_isCompact** (hφ : ∀ {V W} (h : V ≤ W) (a : A W), φ V (rA h a) = rB h (φ W a)) (hbot : ∀ V : Set.Iic U, V.1 = ⊥ → Subsingleton (A V)) (sepA : ∀ {V₁ V₂ W : Set.Iic U} (h₁ : V₁ ≤ W) (h₂ : V₂ ≤ W), W.1 ≤ V₁.1 ⊔ V₂.1 → ∀ a : A W, rA h₁ a = 0 → rA h₂ a = 0 → a = 0) (hker : ∀ V : Set.Iic U, IsAffineOpen V.1 → ∀ a : A V, φ V a = 0 → ∃ s ∈ S, s • a = 0) (W : Set.Iic U) (hW : IsCompact (W.1 : Set X)) (a : A W) (ha : φ W a = 0) : ∃ s ∈ S, s • a = 0

theorem **exists_smul_eq_map_of_isCompact_of_isQuasiSeparated** (hφ : ∀ {V W} (h : V ≤ W) (a : A W), φ V (rA h a) = rB h (φ W a)) (hrB : ∀ {V₁ V₂ V₃} (h : V₁ ≤ V₂) (h' : V₂ ≤ V₃) (b : B V₃), rB h (rB h' b) = rB (h.trans h') b) (hbotA : ∀ V : Set.Iic U, V.1 = ⊥ → Subsingleton (A V)) (hbotB : ∀ V : Set.Iic U, V.1 = ⊥ → Subsingleton (B V)) (sepA : ∀ {V₁ V₂ W : Set.Iic U} (h₁ : V₁ ≤ W) (h₂ : V₂ ≤ W), W.1 ≤ V₁.1 ⊔ V₂.1 → ∀ a : A W, rA h₁ a = 0 → rA h₂ a = 0 → a = 0) (sepB : ∀ {V₁ V₂ W : Set.Iic U} (h₁ : V₁ ≤ W) (h₂ : V₂ ≤ W), W.1 ≤ V₁.1 ⊔ V₂.1 → ∀ b : B W, rB h₁ b = 0 → rB h₂ b = 0 → b = 0) (glueA : ∀ {V₁ V₂ W : Set.Iic U} (h₁ : V₁ ≤ W) (h₂ : V₂ ≤ W), W.1 ≤ V₁.1 ⊔ V₂.1 → ∀ (a₁ : A V₁) (a₂ : A V₂), rA (inf_le_left : V₁ ⊓ V₂ ≤ V₁) a₁ = rA inf_le_right a₂ → ∃ a : A W, rA h₁ a = a₁ ∧ rA h₂ a = a₂) (hker : ∀ V : Set.Iic U, IsAffineOpen V.1 → ∀ a : A V, φ V a = 0 → ∃ s ∈ S, s • a = 0) (hsurj : ∀ V : Se

theorem **subsingleton_sections_of_eq_bot** (h : W = ⊥) : Subsingleton Γ(F, W)

theorem **section_eq_zero_of_res₂** (h₁ : V₁ ≤ W) (h₂ : V₂ ≤ W) (h : W ≤ V₁ ⊔ V₂) (s : Γ(F, W)) (e₁ : F.presheaf.map (homOfLE h₁).op s = 0) (e₂ : F.presheaf.map (homOfLE h₂).op s = 0) : s = 0

theorem **exists_section_glue₂** (h₁ : V₁ ≤ W) (h₂ : V₂ ≤ W) (h : W ≤ V₁ ⊔ V₂) (s₁ : Γ(F, V₁)) (s₂ : Γ(F, V₂)) (e : F.presheaf.map (homOfLE (inf_le_left : V₁ ⊓ V₂ ≤ V₁)).op s₁ = F.presheaf.map (homOfLE inf_le_right).op s₂) : ∃ s : Γ(F, W), F.presheaf.map (homOfLE h₁).op s = s₁ ∧ F.presheaf.map (homOfLE h₂).op s = s₂

theorem **exists_pow_smul_eq_zero_of_isCompact** (hU : IsCompact (U : Set X)) (f : Γ(X, U)) (h : V ≤ U) (hV : V = X.basicOpen f) (t : Γ(F, U)) (ht : F.presheaf.map (homOfLE h).op t = 0) : ∃ n : ℕ, f ^ n • t = 0

theorem **exists_pow_smul_eq_res_of_qcqs** (hU : IsCompact (U : Set X)) (hU' : IsQuasiSeparated (U : Set X)) (f : Γ(X, U)) (h : V ≤ U) (hV : V = X.basicOpen f) (s : Γ(F, V)) : ∃ (n : ℕ) (t : Γ(F, U)), X.presheaf.map (homOfLE h).op f ^ n • s = F.presheaf.map (homOfLE h).op t

theorem **isLocalizedModule_resBasicOpen_of_qcqs** (hU : IsCompact (U : Set X)) (hU' : IsQuasiSeparated (U : Set X)) (f : Γ(X, U)) : IsLocalizedModule.Away f (resBasicOpen F f)

theorem **isLocalizedModule_of_ringHom_of_qcqs** (hU : IsCompact (U : Set X)) (hU' : IsQuasiSeparated (U : Set X)) {A : Type*} [CommRing A] (φ : A →+* Γ(X, U)) (a : A) (h : V ≤ U) (hV : V = X.basicOpen (φ a)) [Module A Γ(F, U)] [Module A Γ(F, V)] (hsU : ∀ (r : A) (s : Γ(F, U)), r • s = φ r • s) (hsV : ∀ (r : A) (s : Γ(F, V)), r • s = X.presheaf.map (homOfLE h).op (φ r) • s) (l : Γ(F, U) →ₗ[A] Γ(F, V)) (hl : ∀ s, l s = F.presheaf.map (homOfLE h).op s) : IsLocalizedModule.Away a l

theorem **isLocalizedModule_resBasicOpen_top** [CompactSpace X] [QuasiSeparatedSpace X] (f : Γ(X, ⊤)) : IsLocalizedModule.Away f (resBasicOpen F f)

theorem **app_res** (h : V ≤ W) (s : Γ(F, W)) : φ.app V (F.presheaf.map (homOfLE h).op s) = G.presheaf.map (homOfLE h).op (φ.app W s)

theorem **exists_smul_eq_zero_of_app_eq_zero** (hU : IsCompact (U : Set X)) (S : Submonoid Γ(X, U)) (hker : ∀ (V : X.Opens) (h : V ≤ U), IsAffineOpen V → ∀ a : Γ(F, V), φ.app V a = 0 → ∃ s ∈ S, X.presheaf.map (homOfLE h).op s • a = 0) (a : Γ(F, U)) (ha : φ.app U a = 0) : ∃ s ∈ S, s • a = 0

theorem **exists_smul_eq_app** (hU : IsCompact (U : Set X)) (hU' : IsQuasiSeparated (U : Set X)) (S : Submonoid Γ(X, U)) (hker : ∀ (V : X.Opens) (h : V ≤ U), IsAffineOpen V → ∀ a : Γ(F, V), φ.app V a = 0 → ∃ s ∈ S, X.presheaf.map (homOfLE h).op s • a = 0) (hsurj : ∀ (V : X.Opens) (h : V ≤ U), IsAffineOpen V → ∀ b : Γ(G, V), ∃ s ∈ S, ∃ a : Γ(F, V), X.presheaf.map (homOfLE h).op s • b = φ.app V a) (b : Γ(G, U)) : ∃ s ∈ S, ∃ a : Γ(F, U), s • b = φ.app U a

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSmulHomologyEqZeroOfAcyclicFibre

theorem **exists_smul_mem_range_of_exact_fibre** {R : Type*} [CommRing R] {A B C : Type*} [AddCommGroup A] [Module R A] [AddCommGroup B] [Module R B] [AddCommGroup C] [Module R C] [Module.Finite R B] [Module.Projective R B] [Module.Finite R C] [Module.Projective R C] (K : Type*) [Field K] [Algebra R K] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (hfg : g ∘ₗ f = 0) (h : Exact (f.lTensor K) (g.lTensor K)) (b : B) (hb : g b = 0) : ∃ r : R, algebraMap R K r ≠ 0 ∧ r • b ∈ LinearMap.range f

theorem **exists_smul_homology_eq_zero_of_acyclic_fibre** (R : Type) [CommRing R] [IsNoetherianRing R] (K : Type) [Field K] (χ : R →+* K) (C : CochainComplex (ModuleCat.{0} R) ℕ) (n : ℕ) (hn : ∀ p, n < p → IsZero (C.X p)) (hflat : ∀ p, Module.Flat R (C.X p)) (hfin : ∀ p, Module.Finite R (C.homology p)) (hac : (((ModuleCat.extendScalars.{0, 0, 0} χ).mapHomologicalComplex _).obj C).Acyclic) (p : ℕ) : ∃ r : R, χ r ≠ 0 ∧ ∀ z : C.homology p, r • z = 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSmulOfTwCoord

theorem **GrSum.apply_eq_eval₂Hom_mul** {ι R B : Type*} [CommRing R] [CommRing B] {A : ℤ → Type*} [∀ n, AddCommGroup (A n)] [∀ n, Module R (A n)] {x : ι → ∀ n, A n →ₗ[R] A (n + 1)} (hx : OperatorsCommute x) (c : R →+* B) (b : ι → B) (φ : ∀ n, A n → B) (hadd : ∀ n a a', φ n (a + a') = φ n a + φ n a') (hsmul : ∀ n (r : R) a, φ n (r • a) = c r * φ n a) (hop : ∀ i n a, φ (n + 1) (x i n a) = b i * φ n a) (F : MvPolynomial ι R) {t t' : ℤ} {a : A t} {a' : A t'} (h : F • GrSum.of hx t a = GrSum.of hx t' a') : φ t' a' = MvPolynomial.eval₂Hom c b F * φ t a

theorem **AbLaw.exists_smul_of_twCoord** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y) (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsSymmetricVeryAmple

theorem **exists_iso_pull_of_triv** {X : Scheme.{0}} {H : LB X} (hH : H.IsLB) (f : X ⟶ X) (h : Triv (LB.LinComb H (K

theorem **exists_symmetric_veryAmple** : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Spec (CommRingCat.of K))) [GrpObj B], IsAbelianScheme B → ∃ (H : LB B.left) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB ∧ VeryAmpleBy B.hom H s ∧ ∃ w : LB.Iso H (H.pull (GrpObj.inv : B ⟶ B).left), w.IsGen

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsTwExtendPowMul

theorem **basicOpen_rs_eq** {L : LB Y} (s : LB.Sec L) (k : L.ι) {W : Y.Opens} (h : W ≤ L.U k) : Y.basicOpen (rs Y h (s.f k)) = nonzeroLocus s ⊓ W

theorem **fun_extend** {W O : Y.Opens} (hW : IsAffineOpen W) (f : Γ(Y, W)) (hO : Y.basicOpen f = O) (h : O ≤ W) (z : Γ(Y, O)) : ∃ (n : ℕ) (x : Γ(Y, W)), rs Y h x = rs Y h f ^ n * z

theorem **fun_kill** {W O : Y.Opens} (hW : IsCompact (W : Set Y)) (f x : Γ(Y, W)) (hO : Y.basicOpen f = O) (h : O ≤ W) (hx : rs Y h x = 0) : ∃ n : ℕ, f ^ n * x = 0

theorem **exists_fin_cover** [CompactSpace Y] {M H : LB Y} (hM : M.IsLB) (hH : H.IsLB) : ∃ (κ : Type) (_ : Fintype κ) (W : κ → Y.Opens) (p : κ → M.ι) (l : κ → H.ι), (∀ a, IsAffineOpen (W a)) ∧ (∀ a, W a ≤ M.U (p a) ⊓ H.U (l a)) ∧ ⊤ ≤ ⨆ a, W a

theorem **unit_shift** {R : Type} [CommRing R] (A B : Rˣ) (t d : ℤ) (k : ℕ) {u v : R} (h : u = (B ^ d).val * v) : (A * B ^ (t + k * d)).val * v ^ k = u ^ k * (A * B ^ t).val

theorem **pw_rel** {H : LB Y} {d : ℤ} (σ : LB.Sec (pw H d)) (u : ∀ l, Γ(Y, H.U l)) (hu : ∀ l, u l = σ.f l) (l m : H.ι) {O : Y.Opens} (hl : O ≤ H.U l) (hm : O ≤ H.U m) (h : O ≤ H.U l ⊓ H.U m) : rs Y hl (u l) = (res h (H.g l m) ^ d).val * rs Y hm (u m)

theorem **exists_tw_extend_pow_mul** : ∀ (Y : Scheme.{0}) (H M : LB Y) (d : ℤ) (σ : LB.Sec (pw H d)) (V : Y.Opens) (t : ℤ), H.IsLB → M.IsLB → V = nonzeroLocus σ → CompactSpace Y → QuasiSeparatedSpace Y → ∀ y : SecOn (tw M H t) V, ∃ (k : ℕ) (a : SecOn (tw M H (t + k * d)) ⊤), ∀ (p : M.ι) (l : H.ι), rs Y (inf_le_inf_right _ le_top : V ⊓ (M.U p ⊓ H.U l) ≤ ⊤ ⊓ (M.U p ⊓ H.U l)) (SecOn.twCoord a p l) = rs Y (inf_le_right.trans inf_le_right : V ⊓ (M.U p ⊓ H.U l) ≤ H.U l) (σ.f l) ^ k * SecOn.twCoord y p l

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsTwistedSectionsCubeBijective

theorem **twistedSumRestrict_of** {V W : Y.Opens} (h : W ≤ V) (t : ℤ) (a : SecOnR f (tw M H t) V) : twistedSumRestrict hx h (GrSum.of (hx V) t a) = GrSum.of (hx W) t (SecOnR.restrict f (tw M H t) h a)

theorem **twistedSumRestrict_comp** {U V W : Y.Opens} (h : V ≤ U) (h' : W ≤ V) (y : GrSum (hx U)) : twistedSumRestrict hx h' (twistedSumRestrict hx h y) = twistedSumRestrict hx (h'.trans h) y

theorem **exists_grCube_twistedSections_hom_secCube_bijective** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsTwistedSectionsCubeBijectiveStatement

theorem **_root_.GlobalTheory.T.exists_twistedSections_cube_bijective** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsUniqueHomExtensionSemiAbelian

theorem **existsUnique_hom_extension_semiAbelian** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsUniqueLbIsoPullEqOfCoheightLeOne

theorem **ringKrullDim_stalk_le_one_iff** (x : X) : ringKrullDim (X.presheaf.stalk x) ≤ 1 ↔ Order.coheight x ≤ 1

theorem **coheight_apply_of_isOpenImmersion** (f : Y ⟶ X) [IsOpenImmersion f] (y : Y) : Order.coheight (f y) = Order.coheight y

theorem **genericPoint_mem_opens** [IsIntegral X] (W : X.Opens) [h : Nonempty W] : genericPoint X ∈ W

theorem **coheight_genericPoint** [IsIntegral X] : Order.coheight (genericPoint X) = 0

theorem **germToFunctionField_mem_range_stalk** [IsIntegral X] (V : X.Opens) [Nonempty V] (s : Γ(X, V)) {x : X} (hx : x ∈ V) : X.germToFunctionField V s ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range

theorem **germToFunctionField_res** [IsIntegral X] {V W : X.Opens} [Nonempty V] [Nonempty W] (h : W ≤ V) (s : Γ(X, V)) : X.germToFunctionField W (X.presheaf.map (homOfLE h).op s) = X.germToFunctionField V s

theorem **exists_germToFunctionField_eq_of_coheight_eq_one** [IsIntegral X] [IsLocallyNoetherian X] [IsNormalScheme X] (V : X.Opens) [Nonempty V] (t : X.functionField) (h : ∀ x ∈ V, Order.coheight x = 1 → t ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range) : ∃ s : Γ(X, V), X.germToFunctionField V s = t

theorem **exists_germToFunctionField_eq_iff_coheight_eq_one** [IsIntegral X] [IsLocallyNoetherian X] [IsNormalScheme X] (V : X.Opens) [Nonempty V] (t : X.functionField) : (∃ s : Γ(X, V), X.germToFunctionField V s = t) ↔ ∀ x ∈ V, Order.coheight x = 1 → t ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range

theorem **bijective_res_of_eq_bot** {V W : X.Opens} (h : W ≤ V) (hV : V = ⊥) : Function.Bijective (X.presheaf.map (homOfLE h).op).hom

theorem **bijective_res_of_coheight_eq_one** [IsIntegral X] [IsLocallyNoetherian X] [IsNormalScheme X] {V W : X.Opens} (h : W ≤ V) [Nonempty W] (hW : ∀ x ∈ V, Order.coheight x = 1 → x ∈ W) : Function.Bijective (X.presheaf.map (homOfLE h).op).hom

theorem **bijective_app_iff_of_isOpenImmersion** (f : Y ⟶ X) [IsOpenImmersion f] (V : X.Opens) : Function.Bijective (f.app V).hom ↔ Function.Bijective (X.presheaf.map (homOfLE (inf_le_left : V ⊓ f.opensRange ≤ V)).op).hom

theorem **bijective_app_of_coheight_eq_one** [IsIntegral X] [IsLocallyNoetherian X] [IsNormalScheme X] (f : Y ⟶ X) [IsOpenImmersion f] [Nonempty Y] (hf : ∀ x : X, Order.coheight x = 1 → x ∈ Set.range f) (V : X.Opens) : Function.Bijective (f.app V).hom

theorem **bijective_app_of_coheight_le_one** [IsLocallyNoetherian X] [IsNormalScheme X] (f : Y ⟶ X) [IsOpenImmersion f] (hf : ∀ x : X, Order.coheight x ≤ 1 → x ∈ Set.range f) (V : X.Opens) : Function.Bijective (f.app V).hom

theorem **bijective_res_inf_of_coheight_le_one** [IsLocallyNoetherian X] [IsNormalScheme X] {U : X.Opens} (hU : ∀ x : X, Order.coheight x ≤ 1 → x ∈ U) (V : X.Opens) : Function.Bijective (X.presheaf.map (homOfLE (inf_le_left : V ⊓ U ≤ V)).op).hom

theorem **res_appLE_apply** (f : Y ⟶ X) {V : X.Opens} {W W' : Y.Opens} (e : W ≤ f ⁻¹ᵁ V) (h : W' ≤ W) (s : Γ(X, V)) : (Y.presheaf.map (homOfLE h).op).hom ((f.appLE V W e).hom s) = (f.appLE V W' (h.trans e)).hom s

theorem **appLE_res_apply** (f : Y ⟶ X) {V V' : X.Opens} {W : Y.Opens} (h : V' ≤ V) (e : W ≤ f ⁻¹ᵁ V') (s : Γ(X, V)) : (f.appLE V' W e).hom ((X.presheaf.map (homOfLE h).op).hom s) = (f.appLE V W (e.trans (f.preimage_mono h))).hom s

theorem **bijective_appLE_of_eq** (f : Y ⟶ X) {V : X.Opens} {W : Y.Opens} (h : W = f ⁻¹ᵁ V) (hf : Function.Bijective (f.app V).hom) : Function.Bijective (f.appLE V W h.le).hom

theorem **injective_appLE_of_eq** (f : Y ⟶ X) {V : X.Opens} {W : Y.Opens} (h : W = f ⁻¹ᵁ V) (hf : Function.Injective (f.app V).hom) : Function.Injective (f.appLE V W h.le).hom

theorem **existsUnique_sec_of_charts** (f : Y ⟶ X) (hf : ∀ V, Function.Bijective (f.app V).hom) {L : LB X} {L' : LB Y} (e : L'.ι ≃ L.ι) (hU : ∀ j, L'.U j = f ⁻¹ᵁ L.U (e j)) (hg : ∀ i j, (L'.g i j).val = (f.appLE (L.U (e i) ⊓ L.U (e j)) (L'.U i ⊓ L'.U j) (inf_le_inf (hU i).le (hU j).le)).hom (L.g (e i) (e j)).val) (s : L'.Sec) : ∃! t : L.Sec, ∀ j, (f.appLE (L.U (e j)) (L'.U j) (hU j).le).hom (t.f (e j)) = s.f j

theorem **existsUnique_gen_of_charts** (f : Y ⟶ X) (hf : ∀ V, Function.Bijective (f.app V).hom) {L : LB X} {L' : LB Y} (e : L'.ι ≃ L.ι) (hU : ∀ j, L'.U j = f ⁻¹ᵁ L.U (e j)) (hg : ∀ i j, (L'.g i j).val = (f.appLE (L.U (e i) ⊓ L.U (e j)) (L'.U i ⊓ L'.U j) (inf_le_inf (hU i).le (hU j).le)).hom (L.g (e i) (e j)).val) (s : L'.Gen) (hs : s.IsGen) : ∃! t : L.Gen, t.IsGen ∧ ∀ j, (f.appLE (L.U (e j)) (L'.U j) (hU j).le).hom (t.s (e j)).val = (s.s j).val

theorem **lb_pull_U** (L : LB X) (f : Y ⟶ X) (j : PUnit.{u + 1} → L.ι) : (L.pull f).U j = f ⁻¹ᵁ L.U (j PUnit.unit)

theorem **lb_pull_g** (L : LB X) (f : Y ⟶ X) (i j : PUnit.{u + 1} → L.ι) : ((L.pull f).g i j).val = (f.appLE (L.U (i PUnit.unit) ⊓ L.U (j PUnit.unit)) ((L.pull f).U i ⊓ (L.pull f).U j) (inf_le_inf (lb_pull_U L f i).le (lb_pull_U L f j).le)).hom (L.g (i PUnit.unit) (j PUnit.unit)).val

theorem **lbSecPull_f** (L : LB X) (s : L.Sec) (f : Y ⟶ X) (j : PUnit.{u + 1} → L.ι) : (lbSecPull L s f).f j = (f.appLE (L.U (j PUnit.unit)) ((L.pull f).U j) (lb_pull_U L f j).le).hom (s.f (j PUnit.unit))

theorem **lbSec_ext** {L : LB X} {s t : L.Sec} (h : ∀ i, s.f i = t.f i) : s = t

theorem **lbGen_ext** {L : LB X} {s t : L.Gen} (h : ∀ i, (s.s i).val = (t.s i).val) : s = t

theorem **genPull_s_val** (L : LB X) (c : L.Gen) (f : Y ⟶ X) (j : PUnit.{u + 1} → L.ι) : ((L.genPull c f).s j).val = (f.appLE (L.U (j PUnit.unit)) ((L.pull f).U j) (lb_pull_U L f j).le).hom (c.s (j PUnit.unit)).val

theorem **bijective_lbSecPull** (f : Y ⟶ X) (hf : ∀ V, Function.Bijective (f.app V).hom) (L : LB X) : Function.Bijective fun s : L.Sec ↦ lbSecPull L s f

theorem **existsUnique_genPull_eq** (f : Y ⟶ X) (hf : ∀ V, Function.Bijective (f.app V).hom) (L : LB X) (r : (L.pull f).Gen) (hr : r.IsGen) : ∃! t : L.Gen, t.IsGen ∧ L.genPull t f = r

theorem **lb_div_pull_U** (L M : LB X) (f : Y ⟶ X) (p : (PUnit.{u + 1} → M.ι) × (PUnit.{u + 1} → L.ι)) : ((M.pull f).div (L.pull f)).U p = f ⁻¹ᵁ (M.div L).U (p.1 PUnit.unit, p.2 PUnit.unit)

theorem **lb_div_pull_g** (L M : LB X) (f : Y ⟶ X) (p q : (PUnit.{u + 1} → M.ι) × (PUnit.{u + 1} → L.ι)) : ((M.pull f).div (L.pull f)).g p q = res (inf_le_inf (lb_div_pull_U L M f p).le (lb_div_pull_U L M f q).le) (pullU f ((M.div L).U (p.1 PUnit.unit, p.2 PUnit.unit) ⊓ (M.div L).U (q.1 PUnit.unit, q.2 PUnit.unit)) ((M.div L).g (p.1 PUnit.unit, p.2 PUnit.unit) (q.1 PUnit.unit, q.2 PUnit.unit)))

theorem **lbIsoPull_s_val** {L M : LB X} (w : LB.Iso L M) (f : Y ⟶ X) (p : (PUnit.{u + 1} → M.ι) × (PUnit.{u + 1} → L.ι)) : ((lbIsoPull w f).s p).val = (f.appLE ((M.div L).U (p.1 PUnit.unit, p.2 PUnit.unit)) (((M.pull f).div (L.pull f)).U p) (lb_div_pull_U L M f p).le).hom (w.s (p.1 PUnit.unit, p.2 PUnit.unit)).val

theorem **existsUnique_lbIsoPull_eq** (f : Y ⟶ X) (hf : ∀ V, Function.Bijective (f.app V).hom) (L M : LB X) (w : LB.Iso (L.pull f) (M.pull f)) (hw : w.IsGen) : ∃! v : LB.Iso L M, v.IsGen ∧ lbIsoPull v f = w

theorem **resG_resG** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (s : Γ(X, U)) : resG X h₂ (resG X h₁ s) = resG X (h₂.trans h₁) s

theorem **resG_refl** {U : X.Opens} (s : Γ(X, U)) : resG X le_rfl s = s

theorem **resG_eq_mul_of_injective** {A B G T A' B' T' : X.Opens} (hA : T ≤ A) (hB : T ≤ B) (hG : T ≤ G) (hA' : A' ≤ A) (hB' : B' ≤ B) (hT : T' ≤ T) (kA : T' ≤ A') (kB : T' ≤ B') (hi : Function.Injective (resG X hT)) {a : Γ(X, A)} {b : Γ(X, B)} {g : Γ(X, G)} (h : resG X kA (resG X hA' a) = resG X (hT.trans hG) g * resG X kB (resG X hB' b)) : resG X hA a = resG X hG g * resG X hB b

theorem **res_mem_secOn** {L : LB X} {V W : X.Opens} (h : W ≤ V) {s : ∀ i, Γ(X, V ⊓ L.U i)} (hs : s ∈ secOn L V) : (fun i ↦ resG X (inf_le_inf_right _ h) (s i)) ∈ secOn L W

theorem **res_sec_mem_secOn** {L : LB X} (W : X.Opens) (s : L.Sec) : (fun i ↦ resG X inf_le_right (s.f i)) ∈ secOn L W

theorem **bijective_resG_of_coheight_le_one** [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] {V W : X.Opens} (h : W ≤ V) (hW : ∀ x ∈ V, Order.coheight x ≤ 1 → x ∈ W) : Function.Bijective (resG X h)

theorem **existsUnique_secOn_of_coheight_le_one** [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] (L : LB X) {V W : X.Opens} (h : W ≤ V) (hW : ∀ x ∈ V, Order.coheight x ≤ 1 → x ∈ W) {s : ∀ i, Γ(X, W ⊓ L.U i)} (hs : s ∈ secOn L W) : ∃! t : ∀ i, Γ(X, V ⊓ L.U i), t ∈ secOn L V ∧ ∀ i, resG X (inf_le_inf_right _ h) (t i) = s i

theorem **existsUnique_sec_res_eq_of_coheight_le_one** [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] (L : LB X) {U : X.Opens} (hU : ∀ x : X, Order.coheight x ≤ 1 → x ∈ U) {s : ∀ i, Γ(X, U ⊓ L.U i)} (hs : s ∈ secOn L U) : ∃! t : L.Sec, ∀ i, resG X inf_le_right (t.f i) = s i

theorem **nonempty_inf_of_isIntegral** (V W : X.Opens) [Nonempty V] [Nonempty W] : Nonempty (V ⊓ W : X.Opens)

theorem **injective_resG_of_inf_le** (W : X.Opens) [Nonempty W] {T T' : X.Opens} (h : T' ≤ T) (hW : W ⊓ T ≤ T') : Function.Injective (resG X h)

theorem **germToFunctionField_mem_range_stalk_of_secOn** {L : LB X} {W : X.Opens} {s : ∀ i, Γ(X, W ⊓ L.U i)} (hs : s ∈ secOn L W) {x : X} {i j : L.ι} (hi : x ∈ L.U i) (hj : x ∈ L.U j) [Nonempty (W ⊓ L.U i : X.Opens)] [Nonempty (W ⊓ L.U j : X.Opens)] (h : X.germToFunctionField _ (s j) ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range) : X.germToFunctionField _ (s i) ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range

theorem **exists_resG_eq_of_secOn** {L : LB X} {W : X.Opens} [Nonempty W] {s : ∀ i, Γ(X, W ⊓ L.U i)} (hs : s ∈ secOn L W) (i : L.ι) {O : X.Opens} (h : W ⊓ L.U i ≤ O) (hO : O ≤ L.U i) (H : ∀ x ∈ O, Order.coheight x = 1 → x ∉ W → ∃ (j : L.ι) (_ : Nonempty (W ⊓ L.U j : X.Opens)), x ∈ L.U j ∧ X.germToFunctionField (W ⊓ L.U j) (s j) ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range) : ∃ t : Γ(X, O), resG X h t = s i

theorem **exists_secOn_res_eq_iff** {L : LB X} (hL : L.IsLB) {V W : X.Opens} (h : W ≤ V) [Nonempty W] {s : ∀ i, Γ(X, W ⊓ L.U i)} (hs : s ∈ secOn L W) : (∃ t ∈ secOn L V, ∀ i, resG X (inf_le_inf_right _ h) (t i) = s i) ↔ ∀ x ∈ V, Order.coheight x = 1 → x ∉ W → ∃ (i : L.ι) (_ : Nonempty (W ⊓ L.U i : X.Opens)), x ∈ L.U i ∧ X.germToFunctionField (W ⊓ L.U i) (s i) ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range

theorem **exists_sec_res_eq_iff** {L : LB X} (hL : L.IsLB) {W : X.Opens} [Nonempty W] {s : ∀ i, Γ(X, W ⊓ L.U i)} (hs : s ∈ secOn L W) : (∃ t : L.Sec, ∀ i, resG X inf_le_right (t.f i) = s i) ↔ ∀ x : X, Order.coheight x = 1 → x ∉ W → ∃ (i : L.ι) (_ : Nonempty (W ⊓ L.U i : X.Opens)), x ∈ L.U i ∧ X.germToFunctionField (W ⊓ L.U i) (s i) ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range

theorem **lbSec_eq_of_res_eq** {L : LB X} (W : X.Opens) [Nonempty W] {t t' : L.Sec} (h : ∀ i, resG X (inf_le_right : W ⊓ L.U i ≤ L.U i) (t.f i) = resG X inf_le_right (t'.f i)) : t = t'

theorem **lb_pull_ι_U** (L : LB X) (W : X.Opens) (j : PUnit.{1} → L.ι) : (L.pull W.ι).U j = W.ι ⁻¹ᵁ (W ⊓ L.U (j PUnit.unit))

theorem **bijective_ι_appLE** (W : X.Opens) {V : X.Opens} (hV : V ≤ W) {T : W.toScheme.Opens} (h : T = W.ι ⁻¹ᵁ V) : Function.Bijective (W.ι.appLE V T h.le).hom

theorem **existsUnique_secOn_of_pull** (L : LB X) (W : X.Opens) (r : (L.pull W.ι).Sec) : ∃! s : ∀ i, Γ(X, W ⊓ L.U i), s ∈ secOn L W ∧ ∀ j : PUnit.{1} → L.ι, (W.ι.appLE (W ⊓ L.U (j PUnit.unit)) _ (lb_pull_ι_U L W j).le).hom (s (j PUnit.unit)) = r.f j

theorem **exists_lbSecPull_eq_iff** [IsIntegral X] [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] {L : LB X} (hL : L.IsLB) {W : X.Opens} [Nonempty W] (r : (L.pull W.ι).Sec) : (∃ t : L.Sec, lbSecPull L t W.ι = r) ↔ ∀ x : X, Order.coheight x = 1 → x ∉ W → ∃ (i : L.ι) (_ : Nonempty (W ⊓ L.U i : X.Opens)) (a : Γ(X, W ⊓ L.U i)), x ∈ L.U i ∧ (W.ι.appLE (W ⊓ L.U i) _ (lb_pull_ι_U L W fun _ ↦ i).le).hom a = r.f (fun _ ↦ i) ∧ X.germToFunctionField (W ⊓ L.U i) a ∈ (algebraMap (X.presheaf.stalk x) X.functionField).range

theorem **bijective_lbSecPull_of_coheight_le_one** {X Y : Scheme.{u}} [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] (f : Y ⟶ X) [IsOpenImmersion f] (hf : ∀ x : X, Order.coheight x ≤ 1 → x ∈ Set.range f) (L : LB X) : Function.Bijective fun s : L.Sec ↦ lbSecPull L s f

theorem **existsUnique_genPull_eq_of_coheight_le_one** {X Y : Scheme.{u}} [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] (f : Y ⟶ X) [IsOpenImmersion f] (hf : ∀ x : X, Order.coheight x ≤ 1 → x ∈ Set.range f) (L : LB X) (r : (L.pull f).Gen) (hr : r.IsGen) : ∃! t : L.Gen, t.IsGen ∧ L.genPull t f = r

theorem **existsUnique_lbIsoPull_eq_of_coheight_le_one** {X Y : Scheme.{u}} [IsLocallyNoetherian X] [GlobalTheory.IsNormalScheme X] (f : Y ⟶ X) [IsOpenImmersion f] (hf : ∀ x : X, Order.coheight x ≤ 1 → x ∈ Set.range f) (L M : LB X) (w : LB.Iso (L.pull f) (M.pull f)) (hw : w.IsGen) : ∃! v : LB.Iso L M, v.IsGen ∧ lbIsoPull v f = w

theorem **lbIso_pull_iff** {X Y : Scheme.{0}} (f : Y ⟶ X) (hf : ∀ V, Function.Bijective (f.app V).hom) (L M : LB X) : LBIso (L.pull f) (M.pull f) ↔ LBIso L M

theorem **extension_over_codim_two** (X : Scheme.{0}) [IsLocallyNoetherian X] (hX : Geo.Tor.IsNormalScheme X) : (∀ U : X.Opens, (∀ x : X, ringKrullDim (X.presheaf.stalk x) ≤ 1 → x ∈ U) → (∀ V : X.Opens, Function.Bijective (resG X (inf_le_left : V ⊓ U ≤ V))) ∧ (∀ (L : LB X) (r : (L.pull U.ι).Sec), ∃! t : L.Sec, lbSecPull L t U.ι = r) ∧ (∀ (L : LB X) (V : X.Opens), ∀ s ∈ secOn L (V ⊓ U), ∃! t : ∀ i, Γ(X, V ⊓ L.U i), t ∈ secOn L V ∧ ∀ i, resG X (inf_le_inf_right _ inf_le_left) (t i) = s i) ∧ (∀ (L : LB X) (r : (L.pull U.ι).Gen), r.IsGen → ∃! t : L.Gen, t.IsGen ∧ L.genPull t U.ι = r) ∧ (∀ (L M : LB X) (w : LB.Iso (L.pull U.ι) (M.pull U.ι)), w.IsGen → ∃! v : LB.Iso L M, v.IsGen ∧ LB.linCombIso v (K

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsUnitSmulEqOfBijectiveToSpanSingleton

theorem **mem_fixedSubring_iff** {a : A} : a ∈ FixedPoints.subring A G ↔ ∀ g : G, g • a = a

theorem **mem_fixedSubalgebra_iff** (R : Type*) [CommRing R] [Algebra R A] [SMulCommClass G R A] {a : A} : a ∈ FixedPoints.subalgebra R A G ↔ ∀ g : G, g • a = a

theorem **isIntegral_fixedSubring** [Finite G] : Algebra.IsIntegral (FixedPoints.subring A G) A

theorem **isIntegral_fixedSubalgebra** [Finite G] (R : Type*) [CommRing R] [Algebra R A] [SMulCommClass G R A] : Algebra.IsIntegral (FixedPoints.subalgebra R A G) A

theorem **isIntegrallyClosed_of_isInvariant** (B : Type*) [CommRing B] [Algebra B A] [FaithfulSMul B A] [SMulCommClass G B A] [Algebra.IsInvariant B A G] [IsDomain A] [IsIntegrallyClosed A] : IsDomain B ∧ IsIntegrallyClosed B

theorem **isIntegrallyClosed_fixedSubring** [IsDomain A] [IsIntegrallyClosed A] : IsIntegrallyClosed (FixedPoints.subring A G)

theorem **isIntegrallyClosed_fixedSubalgebra** (R : Type*) [CommRing R] [Algebra R A] [SMulCommClass G R A] [IsDomain A] [IsIntegrallyClosed A] : IsIntegrallyClosed (FixedPoints.subalgebra R A G)

theorem **finite_of_isInvariant** [Finite G] (R B : Type*) [CommRing R] [CommRing B] [Algebra R A] [Algebra R B] [Algebra B A] [IsScalarTower R B A] [Algebra.IsInvariant B A G] [Algebra.FiniteType R A] : Module.Finite B A

theorem **finiteType_of_isInvariant** [Finite G] (R B : Type*) [CommRing R] [CommRing B] [Algebra R A] [Algebra R B] [Algebra B A] [IsScalarTower R B A] [FaithfulSMul B A] [Algebra.IsInvariant B A G] [IsNoetherianRing R] [Algebra.FiniteType R A] : Algebra.FiniteType R B

theorem **finite_fixedSubalgebra** [Finite G] (R : Type*) [CommRing R] [Algebra R A] [SMulCommClass G R A] [Algebra.FiniteType R A] : Module.Finite (FixedPoints.subalgebra R A G) A

theorem **finiteType_fixedSubalgebra** [Finite G] (R : Type*) [CommRing R] [Algebra R A] [SMulCommClass G R A] [IsNoetherianRing R] [Algebra.FiniteType R A] : Algebra.FiniteType R (FixedPoints.subalgebra R A G)

theorem **isNoetherianRing_fixedSubalgebra** [Finite G] (R : Type*) [CommRing R] [Algebra R A] [SMulCommClass G R A] [IsNoetherianRing R] [Algebra.FiniteType R A] : IsNoetherianRing (FixedPoints.subalgebra R A G)

theorem **exists_orbit_eq_primesOver_of_isInvariant** [Finite G] (B : Type*) [CommRing B] [Algebra B A] [FaithfulSMul B A] [SMulCommClass G B A] [Algebra.IsInvariant B A G] (P : Ideal B) [P.IsPrime] : ∃ Q : Ideal A, Q.IsPrime ∧ Q.LiesOver P ∧ MulAction.orbit G Q = P.primesOver A

theorem **exists_orbit_eq_primesOver** [Finite G] (P : Ideal (FixedPoints.subring A G)) [P.IsPrime] : ∃ Q : Ideal A, Q.IsPrime ∧ Q.LiesOver P ∧ MulAction.orbit G Q = P.primesOver A

theorem **under_eq_iff_exists_smul** [Finite G] (B : Type*) [CommRing B] [Algebra B A] [SMulCommClass G B A] [Algebra.IsInvariant B A G] (Q Q' : Ideal A) [Q.IsPrime] [Q'.IsPrime] : Q.under B = Q'.under B ↔ ∃ g : G, Q' = g • Q

theorem **isBaseChange_of_forall_eq** [Module.Flat R S] {ι : Type*} [Finite ι] {P P' : ι → Type*} [∀ i, AddCommGroup (P i)] [∀ i, Module R (P i)] [∀ i, AddCommGroup (P' i)] [∀ i, Module R (P' i)] [∀ i, Module S (P' i)] [∀ i, IsScalarTower R S (P' i)] {N N' : Type*} [AddCommGroup N] [Module R N] [AddCommGroup N'] [Module R N'] [Module S N'] [IsScalarTower R S N'] (φ ψ : ∀ i, M →ₗ[R] P i) (φ' ψ' : ∀ i, M' →ₗ[S] P' i) (f : M →ₗ[R] M') (hf : IsBaseChange S f) (h : ∀ i, P i →ₗ[R] P' i) (hh : ∀ i, IsBaseChange S (h i)) (hφ : ∀ i x, h i (φ i x) = φ' i (f x)) (hψ : ∀ i x, h i (ψ i x) = ψ' i (f x)) (j : N →ₗ[R] M) (hj : Function.Injective j) (hj' : ∀ x, x ∈ Set.range j ↔ ∀ i, φ i x = ψ i x) (j' : N' →ₗ[S] M') (hj₁ : Function.Injective j') (hj₁' : ∀ x, x ∈ Set.range j' ↔ ∀ i, φ' i x = ψ' i x) (k : N →ₗ[R] N') (hk : ∀ x, j' (k x) = f (j x)) : IsBaseChange S k

@[simp] theorem **coe_invariantsMap** (ρ : Representation R G M) (ρ' : Representation S G M') (f : M →ₗ[R] M') (hf : ∀ g x, f (ρ g x) = ρ' g (f x)) (x : ρ.invariants) : (invariantsMap ρ ρ' f hf x : M') = f x

theorem **isBaseChange_invariantsMap** [Finite G] [Module.Flat R S] (ρ : Representation R G M) (ρ' : Representation S G M') (f : M →ₗ[R] M') (hb : IsBaseChange S f) (hf : ∀ g x, f (ρ g x) = ρ' g (f x)) : IsBaseChange S (invariantsMap ρ ρ' f hf)

theorem **span_image_invariants** [Finite G] [Module.Flat R S] (ρ : Representation R G M) (ρ' : Representation S G M') (f : M →ₗ[R] M') (hb : IsBaseChange S f) (hf : ∀ g x, f (ρ g x) = ρ' g (f x)) : Submodule.span S (f '' ρ.invariants) = ρ'.invariants

theorem **mem_baseChange_invariants_iff** [Finite G] [Module.Flat R S] (ρ : Representation R G M) (x : S ⊗[R] M) : x ∈ ρ.invariants.baseChange S ↔ ∀ g, (ρ g).baseChange S x = x

theorem **injective_baseChange_invariants_subtype** [Module.Flat R S] (ρ : Representation R G M) : Function.Injective (ρ.invariants.subtype.baseChange S)

theorem **isLocalizedModule_invariantsMap** [Finite G] (T : Submonoid R) [IsLocalization T S] (ρ : Representation R G M) (ρ' : Representation S G M') (f : M →ₗ[R] M') [IsLocalizedModule T f] (hf : ∀ g x, f (ρ g x) = ρ' g (f x)) : IsLocalizedModule T (invariantsMap ρ ρ' f hf)

theorem **span_inf_restrictScalars_of_isLocalization** (T : Submonoid R) [IsLocalization T S] (L : Submodule R M') (W : Submodule S M') : Submodule.span S ((L ⊓ W.restrictScalars R : Submodule R M') : Set M') = Submodule.span S (L : Set M') ⊓ W

@[simp] theorem **coe_fixedSubalgebraMap** (f : A →ₐ[R] A') (hf : ∀ (g : G) a, f (g • a) = g • f a) (x : FixedPoints.subalgebra R A G) : (fixedSubalgebraMap G S f hf x : A') = f x

theorem **isBaseChange_fixedSubalgebraMap** [Finite G] [Module.Flat R S] (f : A →ₐ[R] A') (hb : IsBaseChange S f.toLinearMap) (hf : ∀ (g : G) a, f (g • a) = g • f a) : IsBaseChange S (fixedSubalgebraMap G S f hf).toLinearMap

theorem **isInvariant_of_isBaseChange** [Finite G] [Module.Flat R S] [Algebra.IsInvariant R A G] (f : A →ₐ[R] A') (hb : IsBaseChange S f.toLinearMap) (hf : ∀ (g : G) a, f (g • a) = g • f a) : Algebra.IsInvariant S A' G

theorem **faithfulSMul_of_isBaseChange** [Module.Flat R S] [FaithfulSMul R A] (f : A →ₐ[R] A') (hb : IsBaseChange S f.toLinearMap) : FaithfulSMul S A'

theorem **tensorMulSemiringAction_smul_tmul** (g : G) (s : S) (a : A) : letI

theorem **tensorMulSemiringAction_includeRight** (g : G) (a : A) : letI

theorem **tensorMulSemiringAction_smulCommClass** : letI

theorem **isBaseChange_fixedSubalgebra_tensor** [Finite G] [Module.Flat R S] : letI

theorem **isInvariant_tensor** [Finite G] [Module.Flat R S] [Algebra.IsInvariant R A G] : letI

theorem **localizationMulSemiringAction_smul_algebraMap** (U : Submonoid A) (hU : ∀ (g : G), ∀ u ∈ U, g • u ∈ U) (A' : Type*) [CommRing A'] [Algebra A A'] [IsLocalization U A'] (g : G) (a : A) : letI

theorem **localizationMulSemiringAction_smulDistribClass** (U : Submonoid A) (hU : ∀ (g : G), ∀ u ∈ U, g • u ∈ U) (A' : Type*) [CommRing A'] [Algebra A A'] [IsLocalization U A'] : letI

theorem **smul_mem_algebraMapSubmonoid** (T : Submonoid B) (g : G) : ∀ u ∈ Algebra.algebraMapSubmonoid A T, g • u ∈ Algebra.algebraMapSubmonoid A T

theorem **isInvariant_of_isLocalization** [Finite G] [Algebra.IsInvariant B A G] (T : Submonoid B) (B' A' : Type*) [CommRing B'] [CommRing A'] [Algebra B B'] [Algebra A A'] [Algebra B' A'] [Algebra B A'] [IsScalarTower B B' A'] [IsScalarTower B A A'] [IsLocalization T B'] [IsLocalization (Algebra.algebraMapSubmonoid A T) A'] [MulSemiringAction G A'] [SMulDistribClass G A A'] : SMulCommClass G B' A' ∧ Algebra.IsInvariant B' A' G ∧ (FaithfulSMul B A → FaithfulSMul B' A')

theorem **invertible_of_bijective_toSpanSingleton** {M : Type*} [AddCommGroup M] [Module A M] {e : M} (he : Function.Bijective (LinearMap.toSpanSingleton A M e)) : Module.Invertible A M

theorem **exists_unit_smul_eq_of_bijective_toSpanSingleton** {M : Type*} [AddCommGroup M] [Module A M] {e e' : M} (he : Function.Bijective (LinearMap.toSpanSingleton A M e)) (he' : Function.Bijective (LinearMap.toSpanSingleton A M e')) : ∃ u : Aˣ, e' = (u : A) • e

theorem **bijective_toSpanSingleton_tprod** {ι : Type*} [Finite ι] {M : ι → Type*} [∀ i, AddCommGroup (M i)] [∀ i, Module A (M i)] (e : ∀ i, M i) (he : ∀ i, Function.Bijective (LinearMap.toSpanSingleton A (M i) (e i))) : Function.Bijective (LinearMap.toSpanSingleton A (⨂[A] i, M i) (PiTensorProduct.tprod A e))

theorem **dual_apply_smul_comm** [Module.Invertible A M] (f : Module.Dual A M) (x y : M) : f x • y = f y • x

theorem **exists_sum_dual_apply_eq_one** (A M : Type*) [CommRing A] [AddCommGroup M] [Module A M] [Module.Invertible A M] : ∃ s : Finset (Module.Dual A M × M), ∑ p ∈ s, p.1 p.2 = 1

theorem **tprod_comp_perm** [Module.Invertible A M] {ι : Type*} [Finite ι] (x : ι → M) (σ : Equiv.Perm ι) : PiTensorProduct.tprod A (s

theorem **free_of_invertible_of_finite_maximalSpectrum** (A M : Type*) [CommRing A] [AddCommGroup M] [Module A M] [Finite (MaximalSpectrum A)] [Module.Invertible A M] : Module.Free A M

theorem **exists_bijective_toSpanSingleton_of_finite_maximalSpectrum** (A M : Type*) [CommRing A] [AddCommGroup M] [Module A M] [Finite (MaximalSpectrum A)] [Module.Invertible A M] : ∃ e : M, Function.Bijective (LinearMap.toSpanSingleton A M e)

theorem **exists_dual_apply_notMem_of_finite** (M : Type*) [AddCommGroup M] [Module A M] [Module.Invertible A M] (s : Set (Ideal A)) (hs : s.Finite) (hp : ∀ P ∈ s, P.IsPrime) : ∃ (e : M) (f : Module.Dual A M), ∀ P ∈ s, f e ∉ P

theorem **bijective_toSpanSingleton_of_isBaseChange** [Module.Invertible A M] (f : Module.Dual A M) (e : M) {A' M' : Type*} [CommRing A'] [Algebra A A'] [AddCommGroup M'] [Module A M'] [Module A' M'] [IsScalarTower A A' M'] (l : M →ₗ[A] M') (hl : IsBaseChange A' l) (hu : IsUnit (algebraMap A A' (f e))) : Function.Bijective (LinearMap.toSpanSingleton A' M' (l e))

theorem **exists_invariant_notMem_orbit_generator** (G : Type*) [Group G] [Finite G] [MulSemiringAction G A] (M : Type*) [AddCommGroup M] [Module A M] [Module.Invertible A M] (Q : Ideal A) [Q.IsPrime] : ∃ (s : A) (e : M), (∀ g : G, g • s = s) ∧ (∀ g : G, s ∉ g • Q) ∧ ∃ f : Module.Dual A M, f e = s

@[simp] theorem **piTensorMap$_{sl}$_tprod** {A A' : Type*} [CommRing A] [CommRing A'] (σ : A →+* A') {ι : Type*} {M N : ι → Type*} [∀ i, AddCommGroup (M i)] [∀ i, Module A (M i)] [∀ i, AddCommGroup (N i)] [∀ i, Module A' (N i)] (f : ∀ i, M i →$_{sl}$[σ] N i) (x : ∀ i, M i) : piTensorMap$_{sl}$ σ f (PiTensorProduct.tprod A x) = PiTensorProduct.tprod A' fun i ↦ f i (x i)

@[simp] theorem **smulSemilinearMap_apply** (g : G) (x : M) : smulSemilinearMap A M g x = g • x

theorem **smulCommClass_of_smulDistribClass** (B : Type*) [CommRing B] [Algebra B A] [SMulCommClass G B A] [Module B M] [IsScalarTower B A M] : SMulCommClass G B M

theorem **bijective_toSpanSingleton_smul** {e : M} (he : Function.Bijective (LinearMap.toSpanSingleton A M e)) (g : G) : Function.Bijective (LinearMap.toSpanSingleton A M (g • e))

theorem **forall_smul_eq_iff_of_invariant_generator** {n : M} (hn : Function.Bijective (LinearMap.toSpanSingleton A M n)) (hn' : ∀ g : G, g • n = n) (y : M) : (∀ g : G, g • y = y) ↔ ∃ a ∈ FixedPoints.subring A G, y = a • n

theorem **bijective_toSpanSingleton_invariants** {n : M} (hn : Function.Bijective (LinearMap.toSpanSingleton A M n)) (hn' : ∀ g : G, g • n = n) : Function.Bijective (LinearMap.toSpanSingleton B (Representation.ofDistribMulAction B G M).invariants ⟨n, hn'⟩)

theorem **isBaseChange_invariants_subtype** {n : M} (hn : Function.Bijective (LinearMap.toSpanSingleton A M n)) (hn' : ∀ g : G, g • n = n) : IsBaseChange A (Representation.ofDistribMulAction B G M).invariants.subtype

theorem **piTensorDistribMulAction_smul_tprod** (g : G) (x : ∀ i, M i) : letI

theorem **piTensorDistribMulAction_smulDistribClass** : letI

theorem **tensorNorm_eq_of_equiv** [Finite G] [Module.Invertible A M] (κ κ' : ι ≃ G) (x : M) : tensorNorm A κ x = tensorNorm A κ' x

theorem **tensorNorm_smul_group** [Finite G] [Module.Invertible A M] (κ : ι ≃ G) (g : G) (x : M) : tensorNorm A κ (g • x) = tensorNorm A κ x

theorem **tensorNorm_smul** [Fintype G] (κ : ι ≃ G) (a : A) (x : M) : tensorNorm A κ (a • x) = (∏ g : G, g • a) • tensorNorm A κ x

theorem **smul_tensorNorm** [Finite G] [Module.Invertible A M] (κ : ι ≃ G) (x : M) (g : G) : letI

theorem **bijective_toSpanSingleton_tensorNorm** [Finite G] (κ : ι ≃ G) {e : M} (he : Function.Bijective (LinearMap.toSpanSingleton A M e)) : Function.Bijective (LinearMap.toSpanSingleton A _ (tensorNorm A κ e))

theorem **exists_unit_tensorNorm_eq** [Finite G] (κ : ι ≃ G) {e e' : M} (he : Function.Bijective (LinearMap.toSpanSingleton A M e)) (he' : Function.Bijective (LinearMap.toSpanSingleton A M e')) : ∃ u : (FixedPoints.subring A G)ˣ, tensorNorm A κ e' = ((u : FixedPoints.subring A G) : A) • tensorNorm A κ e

theorem **forall_smul_eq_iff_tensorNorm** [Finite G] (κ : ι ≃ G) {e : M} (he : Function.Bijective (LinearMap.toSpanSingleton A M e)) (y : ⨂[A] _ : ι, M) : letI

theorem **exists_invariant_notMem_orbit_bijective** (G : Type*) [Group G] [Finite G] {A : Type*} [CommRing A] [MulSemiringAction G A] (M : Type*) [AddCommGroup M] [Module A M] [Module.Invertible A M] (Q : Ideal A) [Q.IsPrime] : ∃ (s : A) (e : M), (∀ g : G, g • s = s) ∧ (∀ g : G, s ∉ g • Q) ∧ Function.Bijective (LinearMap.toSpanSingleton (Localization.Away s) (LocalizedModule (.powers s) M) (LocalizedModule.mkLinearMap (.powers s) M e))

theorem **finite_group_invariants_and_norms** (G : Type v) [Group G] [Finite G] (A : Type u) [CommRing A] [MulSemiringAction G A] : Algebra.IsIntegral (FixedPoints.subring A G) A ∧ (IsDomain A → IsIntegrallyClosed A → IsDomain (FixedPoints.subring A G) ∧ IsIntegrallyClosed (FixedPoints.subring A G)) ∧ (∀ (R : Type u) [CommRing R] [Algebra R A] [SMulCommClass G R A] [IsNoetherianRing R] [Algebra.FiniteType R A], Algebra.FiniteType R (FixedPoints.subalgebra R A G) ∧ Module.Finite (FixedPoints.subalgebra R A G) A) ∧ (∀ (P : Ideal (FixedPoints.subring A G)) [P.IsPrime], ∃ Q : Ideal A, Q.IsPrime ∧ Q.LiesOver P ∧ MulAction.orbit G Q = P.primesOver A) ∧ (∀ (R S M : Type u) [CommRing R] [CommRing S] [Algebra R S] [Module.Flat R S] [AddCommGroup M] [Module R M] (ρ : Representation R G M), Function.Injective (ρ.invariants.subtype.baseChange S) ∧ ∀ x : S ⊗[R] M, x ∈ ρ.invariants.baseChange S ↔ ∀ g, (ρ

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsWitnessesOfFormallySmoothQuotient

theorem **exists_sum_mul_of_mem_span_range_sq** {S : Type} [CommRing S] {r : ℕ} (g : Fin r → S) (x : S) (hx : x ∈ Ideal.span (Set.range g) * Ideal.span (Set.range g)) : ∃ b : Fin r → Fin r → S, x = ∑ k, ∑ l, b k l * (g k * g l)

theorem **exists_witnesses_of_formallySmooth_quotient** : ∀ (ι : Type) [Fintype ι] (R : Type) [CommRing R] (r : ℕ) (g : Fin r → MvPolynomial ι R), Algebra.FormallySmooth R (MvPolynomial ι R ⧸ Ideal.span (Set.range g)) → ∃ (a : ι → Fin r → MvPolynomial ι R) (b : Fin r → Fin r → Fin r → MvPolynomial ι R), ∀ j, aeval (fun i => X i + ∑ k, a i k * g k) (g j) = ∑ k, ∑ l, b j k l * (g k * g l)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsXPowSmulEqZeroOfEqXSubOneSmul

private lemma **decompose_X_smul_add_one** (y : M) (m : ℤ) : (decompose 𝒩 ((X i : MvPolynomial ι R) • y) (m + 1) : M) = (X i : MvPolynomial ι R) • (decompose 𝒩 y m : M)

private lemma **decompose_X_sub_one_smul_add_one** (w : M) (m : ℤ) : (decompose 𝒩 ((X i - 1 : MvPolynomial ι R) • w) (m + 1) : M) = (X i : MvPolynomial ι R) • (decompose 𝒩 w m : M) - (decompose 𝒩 w (m + 1) : M)

private lemma **exists_decompose_eq_zero_of_lt** (y : M) : ∃ N : ℤ, ∀ j < N, (decompose 𝒩 y j : M) = 0

private lemma **exists_decompose_eq_zero_of_gt** (y : M) : ∃ N : ℤ, ∀ j > N, (decompose 𝒩 y j : M) = 0

private lemma **eq_zero_of_X_smul_recurrence_of_vanish_below** {g : ℤ → M} {N c : ℤ} (hN : ∀ j < N, g j = 0) (hrec : ∀ m < c, g (m + 1) = (X i : MvPolynomial ι R) • g m) : g c = 0

private lemma **exists_X_pow_smul_eq_zero_of_vanish_above** {g : ℤ → M} {N c : ℤ} (hN : ∀ j > N, g j = 0) (hrec : ∀ m ≥ c, g (m + 1) = (X i : MvPolynomial ι R) • g m) : ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • g c = 0

theorem **_root_.GlobalTheory.exists_X_pow_smul_eq_zero_of_eq_X_sub_one_smul** {n : ℤ} {z w : M} (hz : z ∈ 𝒩 n) (h : z = (X i - 1 : MvPolynomial ι R) • w) : ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • z = 0

private lemma **X_pow_smul_mem_add_of_mem** {j : ℤ} {v : M} (hv : v ∈ 𝒩 j) (k : ℕ) : (X i : MvPolynomial ι R) ^ k • v ∈ 𝒩 (j + k)

private lemma **exists_mem_add_X_sub_one_smul_of_mem_of_le** {j n : ℤ} {v : M} (hv : v ∈ 𝒩 j) (hn : j ≤ n) : ∃ y' ∈ 𝒩 n, ∃ w : M, v = y' + (X i - 1 : MvPolynomial ι R) • w

theorem **_root_.GlobalTheory.exists_homogeneous_add_X_sub_one_smul** (y : M) : ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∃ y' ∈ 𝒩 n, ∃ w : M, y = y' + (X i - 1 : MvPolynomial ι R) • w

private lemma **eq_zero_of_X_sub_one_smul_eq_zero** {y : M} (hy : (X i - 1 : MvPolynomial ι R) • y = 0) : y = 0

theorem **_root_.GlobalTheory.X_sub_one_smul_graded** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [_root_.GlobalTheory.PolyGradedSMul ι 𝒩] [Decomposition 𝒩] (i : ι), (∀ y : M, (X i - 1 : MvPolynomial ι R) • y = 0 → y = 0) ∧ (∀ (n : ℤ) (z w : M), z ∈ 𝒩 n → z = (X i - 1 : MvPolynomial ι R) • w → ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • z = 0) ∧ (∀ y : M, ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∃ y' ∈ 𝒩 n, ∃ w : M, y = y' + (X i : MvPolynomial ι R) • w)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsXPowSmulMemSpanOfFiniteQuot

theorem **_root_.GlobalTheory.exists_X_pow_smul_mem_span_of_finite_quot** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [_root_.GlobalTheory.PolyGradedSMul ι 𝒩] [Decomposition 𝒩] (i : ι), Module.Finite (MvPolynomial ι R) (QuotSMulTop (X i - 1 : MvPolynomial ι R) M) → ∃ T : Finset M, ∀ y : M, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ Submodule.span (MvPolynomial ι R) (T : Set M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ExistsXPowSmulMemSpanOfFiniteQuotStatement

theorem **_root_.GlobalTheory.T.exists_X_pow_smul_mem_span_of_finite_quotSMulTop** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩] (i : ι), Module.Finite (MvPolynomial ι R) (QuotSMulTop (X i - 1 : MvPolynomial ι R) M) → ∃ T : Finset M, ∀ y : M, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ Submodule.span (MvPolynomial ι R) (T : Set M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteCechHomologyOfFiniteCubeHomology

theorem **finite_cycles_of_finite_homology** {R : Type} [Ring R] {α : Type} {c : ComplexShape α} (K : HomologicalComplex (ModuleCat.{0} R) c) (i j : α) (hij : c.prev j = i) [Module.Finite R (K.X i)] [Module.Finite R (K.homology j)] : Module.Finite R (K.cycles j)

theorem **finite_cechHomology_of_finite_cubeHomology** (ι : Type) [LinearOrder ι] (R : Type) [CommRing R] [IsNoetherianRing R] (A : Cube.{0} ι R) (h₀ : Module.Finite R (A.obj ∅)) (h : ∀ p, Module.Finite R ((cubeComplex A).homology p)) (p : ℕ) : Module.Finite R ((cechComplex A).homology p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteCechLineBundleHomology

theorem **finite_cechLB_homology** (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y) (hH : H.IsLB) (hM : M.IsLB) (hv : VeryAmpleBy f H s) (p : ℕ) : Module.Finite R ((cechLB f M (fun i => nonzeroLocus (s i))).homology p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteCechLineBundleHomologyStatement

theorem **_root_.GlobalTheory.T.finite_cechLineBundle_homology** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ p, Module.Finite R ((cechLB f M (fun i => nonzeroLocus (s i))).homology p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteFibrePoints

theorem **finite_fibre_points** {K : Type} [Field K] [IsAlgClosed K] {X Y : Over (Spec (CommRingCat.of K))} [LocallyOfFiniteType X.hom] (f : X ⟶ Y) [IsFinite f.left] (y : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ Y) : Finite {x : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ X // x ≫ f = y}

theorem **finite_torsion** : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Spec (CommRingCat.of K))) [GrpObj B], IsAbelianScheme B → ∀ n : ℕ, 1 ≤ n → Finite {x : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ B // x ^ n = 1}

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteGrCechHomology

theorem **finite_grCech_homology** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩], Module.Finite (MvPolynomial ι R) M → ∀ n p, Module.Finite R ((cechComplex (grCube ι 𝒩 n)).homology p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteHomologyCubeComplexGradedModule

private theorem **finite_of_exact_of_finite** {R A B C : Type*} [CommRing R] [IsNoetherianRing R] [AddCommGroup A] [Module R A] [AddCommGroup B] [Module R B] [AddCommGroup C] [Module R C] {f : A →ₗ[R] B} {g : B →ₗ[R] C} (h : Function.Exact f g) (hA : Module.Finite R A) (hC : Module.Finite R C) : Module.Finite R B

theorem **finite_cubeComplex_grCube_homology** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩], Module.Finite (MvPolynomial ι R) M → ∀ n p, Module.Finite R ((cubeComplex (grCube ι 𝒩 n)).homology p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteHomologyCubeComplexGradedModuleStatement

theorem **_root_.GlobalTheory.T.finite_homology_cubeComplex_gradedModule** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩], Module.Finite (MvPolynomial ι R) M → ∀ n p, Module.Finite R ((cubeComplex (grCube ι 𝒩 n)).homology p)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinitePieceOfPolyGradedSMul

theorem **finite_piece_of_polyGradedSMul** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩], Module.Finite (MvPolynomial ι R) M → ∀ n, Module.Finite R (𝒩 n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FiniteQuotSMulTopXSubOneOfChart

theorem **_root_.GlobalTheory.finite_quotSMulTop_X_sub_one_of_chart** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (G : Type) [AddCommGroup G] [Module R G] [Module (MvPolynomial ι R) G] [IsScalarTower R (MvPolynomial ι R) G] (𝒢 : ℤ → Submodule R G) [_root_.GlobalTheory.PolyGradedSMul ι 𝒢] [Decomposition 𝒢] (i : ι) (B : Type) (_ : CommRing B) (_ : Algebra R B) (π : MvPolynomial ι R →ₐ[R] B) (P : Type) (_ : AddCommGroup P) (_ : Module B P) (ε : G →ₛₗ[(π : MvPolynomial ι R →+* B)] P), (Function.Surjective π ∧ π (MvPolynomial.X i) = 1 ∧ Function.Surjective ε ∧ ∀ n, ∀ y ∈ 𝒢 n, ε y = 0 → ∃ k : ℕ, (MvPolynomial.X i : MvPolynomial ι R) ^ k • y = 0) → (Module.Finite B P → Module.Finite (MvPolynomial ι R) (QuotSMulTop ((MvPolynomial.X i : MvPolynomial ι R) − 1) G)) ∧ (Algebra.FinitePresentation R B → Module.FinitePresentation B P → Module.FinitePresentation (MvPolynomial ι R) (Qu

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinrankCechLBHomologyExtProd

theorem **isAffineOpen_preimage_inf_of_isSeparated** {Y Z : Scheme.{0}} [Y.IsSeparated] (p : Z ⟶ Y) {U : Y.Opens} (hU : IsAffineOpen U) {V : Z.Opens} (hV : IsAffineOpen V) : IsAffineOpen (p ⁻¹ᵁ U ⊓ V)

theorem **rs_coord_pullLE_restrict** {Y Z : Scheme.{0}} {M : LB Y} (p : Z ⟶ Y) {V V' : Y.Opens} {O O' B B' : Z.Opens} (h : V' ≤ V) (hO : O' ≤ O) (e : O ≤ p ⁻¹ᵁ V) (e' : O' ≤ p ⁻¹ᵁ V') (a : SecOn M V) (j : (M.pull p).ι) (h₁ : B' ≤ O' ⊓ (M.pull p).U j) (h₂ : B ≤ O ⊓ (M.pull p).U j) (h₃ : B' ≤ B) : rs Z h₁ (GlobalTheory.LB.coord (SecOn.pullLE M p V' O' e' (SecOn.restrict M h a)) j) = rs Z h₃ (rs Z h₂ (GlobalTheory.LB.coord (SecOn.pullLE M p V O e a) j))

theorem **finrank_cechLB_homology_extProd** : ∀ (K : Type) [Field K] (Y Y' Z : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of K)) (f' : Y' ⟶ Spec (CommRingCat.of K)) (p₁ : Z ⟶ Y) (p₂ : Z ⟶ Y') (M : LB Y) (M' : LB Y'), IsPullback p₁ p₂ f f' → M.IsLB → M'.IsLB → IsSeparated f → IsSeparated f' → ∀ (κ₁ : Type) [LinearOrder κ₁] [Fintype κ₁] (U : κ₁ → Y.Opens) (κ₂ : Type) [LinearOrder κ₂] [Fintype κ₂] (W : κ₂ → Y'.Opens) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Z.Opens), (∀ i, IsAffineOpen (U i)) → ⊔ i, U i = ⊤ → (∀ i, IsAffineOpen (W i)) → ⊔ i, W i = ⊤ → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (∀ n, Module.Finite K ((cechLB f M U).homology n)) → (∀ n, Module.Finite K ((cechLB f' M' W).homology n)) → ∀ n, Module.finrank K ((cechLB (p₁ ≫ f) (tw (M.pull p₁) (M'.pull p₂) 1) D).homology n) = ∑ x ∈ Finset.HasAntidiagonal.antidiagonal n, Module.finrank K ((cechLB f M U).homology x.1) * Module.finra

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinrankCechLBHomologyExtProdStatement

theorem **_root_.GlobalTheory.T.finrank_cechLB_homology_extProd_alt** : ∀ (K : Type) [Field K] (Y Y' Z : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of K)) (f' : Y' ⟶ Spec (CommRingCat.of K)) (p₁ : Z ⟶ Y) (p₂ : Z ⟶ Y') (M : LB Y) (M' : LB Y'), IsPullback p₁ p₂ f f' → M.IsLB → M'.IsLB → IsSeparated f → IsSeparated f' → ∀ (κ₁ : Type) [LinearOrder κ₁] [Fintype κ₁] (U : κ₁ → Y.Opens) (κ₂ : Type) [LinearOrder κ₂] [Fintype κ₂] (W : κ₂ → Y'.Opens) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Z.Opens), (∀ i, IsAffineOpen (U i)) → ⊔ i, U i = ⊤ → (∀ i, IsAffineOpen (W i)) → ⊔ i, W i = ⊤ → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (∀ n, Module.Finite K ((cechLB f M U).homology n)) → (∀ n, Module.Finite K ((cechLB f' M' W).homology n)) → ∀ n, Module.finrank K ((cechLB (p₁ ≫ f) (tw (M.pull p₁) (M'.pull p₂) 1) D).homology n) = ∑ x ∈ Finset.HasAntidiagonal.antidiagonal n, Module.finrank K ((cechLB f M U).ho

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinrankMumfordBundleCechHomologyOrigin

theorem **nonempty_extendScalars_equiv** {R : Type} (O : Type) [CommRing R] [CommRing O] [inst : Algebra R O] (M : ModuleCat.{0} R) : Nonempty ((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R O)).obj M ≃ₗ[O] O ⊗[R] M)

theorem **nonempty_homology_extendScalars_equiv** {R : Type} (O : Type) [CommRing R] [CommRing O] [Algebra R O] [Module.Flat R O] (C : CochainComplex (ModuleCat.{0} R) ℕ) (n : ℕ) : Nonempty ((((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R O)).mapHomologicalComplex _).obj C).homology n ≃ₗ[O] O ⊗[R] C.homology n)

theorem **finrank_homology** (K : Type) [Field K] [IsAlgClosed K] (R : Type) [CommRing R] [Algebra K R] [Algebra.FiniteType K R] (χ₀ : R →ₐ[K] K) (O : Type) [CommRing O] [Algebra R O] [(RingHom.ker χ₀).IsPrime] [IsLocalization.AtPrime O (RingHom.ker χ₀)] [IsRegularLocalRing O] (g : ℕ) (hg : ringKrullDim O = g) (C : CochainComplex (ModuleCat.{0} R) ℕ) (hz : ∀ p, g < p → IsZero (C.X p)) (hflat : ∀ p, Module.Flat R (C.X p)) (hfin : ∀ p, Module.Finite R (C.homology p)) (hsupp : ∀ χ : R →ₐ[K] K, χ ≠ χ₀ → ∀ p, ∃ r : R, χ r ≠ 0 ∧ ∀ z : C.homology p, r • z = 0) (C' : CochainComplex (ModuleCat.{0} O) ℕ) (e : ((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R O)).mapHomologicalComplex _).obj C ≅ C') (hq : ∀ (E : CochainComplex (ModuleCat.{0} O) ℕ) (φ : E ⟶ C'), (∀ p, Module.Free O (E.X p) ∧ Module.Finite O (E.X p)) → (∀ (R' : Type) [CommRing R'] [Algebra O R'], QuasiIso (((ModuleCat.extendScalars.{0, 0

theorem **nonempty_homology_restrictScalars_equiv** {K R : Type} [CommRing K] [CommRing R] (a : K →+* R) {Y : Scheme.{0}} (f : Y ⟶ Spec (CommRingCat.of R)) (f₀ : Y ⟶ Spec (CommRingCat.of K)) (w : f₀ = f ≫ Spec.map (CommRingCat.ofHom a)) (M : LB Y) {κ : Type} [LinearOrder κ] (D : κ → Y.Opens) (n : ℕ) : Nonempty ((cechLB f₀ M D).homology n ≃ₗ[K] (ModuleCat.restrictScalars a).obj ((cechLB f M D).homology n))

theorem **lbIso_pull_pull_comp** {X Y Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) (φ : Y ⟶ X) (ψ : Z ⟶ Y) : LBIso ((L.pull φ).pull ψ) (L.pull (ψ ≫ φ))

theorem **finrank_cechLB_homology** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of R)), IsPullback π f (fst P.A P.A).left t → IsOpenImmersion t → (∃ p q, t p = η[P.A].left q) → ∀ D : Fin (P.g + 1) → Y.Opens, (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → ∀ n, Module.Finite K ((cechLB (f ≫ t ≫ P.A.hom) ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology n) ∧ Module.finrank K ((cechLB (f ≫ t ≫ P.A.hom) ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology n) = if n = P.g then 1 else 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinrankMumfordBundleCechHomologyOriginStatement

theorem **_root_.GlobalTheory.T.finrank_mumfordBundle_cech_homology_origin** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of R)), IsPullback π f (fst P.A P.A).left t → IsOpenImmersion t → (∃ p q, t p = η[P.A].left q) → ∀ D : Fin (P.g + 1) → Y.Opens, (∀ i, IsAffineOpen (D i)) → ⨆ i, D i = ⊤ → ∀ n, Module.Finite K ((cechLB (f ≫ t ≫ P.A.hom) ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology n) ∧ Module.finrank K ((cechLB (f ≫ t ≫ P.A.hom) ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology n) = if n = P.g then 1 else 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinrankPrincipalPolarisationSectionModuleLeOne

theorem **le_one_of_convolution** (a b : ℕ → ℕ) (g : ℕ) (h : ∀ n, ∑ x ∈ Finset.HasAntidiagonal.antidiagonal n, a x.1 * b x.2 = if n = g then 1 else 0) : a 0 ≤ 1

theorem **exists_fin_affine_cover** (X : Scheme.{0}) [CompactSpace X] : ∃ (n : ℕ) (D : Fin n → X.Opens), (∀ i, IsAffineOpen (D i)) ∧ ⨆ i, D i = ⊤

theorem **isPullback_mul_snd** {S : Scheme.{0}} (A : Over S) [GrpObj A] : IsPullback (lift (fst A A ≫ 𝟙 A) (snd A A) ≫ μ[A]).left (snd A A).left A.hom A.hom

theorem **finrank_secOnR_le_one** (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) : Module.finrank K (SecOnR P.A.hom P.L ⊤) ≤ 1

theorem **finrank_secM_le_one** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K)))) ≤ 1

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FinrankPrincipalPolarisationSectionModuleLeOneStatement

theorem **_root_.GlobalTheory.T.finrank_principalPolarisation_sectionModule_le_one** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K)))) ≤ 1

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FirstCohomologyVanishesOfExactSecCube

theorem **firstCohomologyVanishes_of_exact_secCube** (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (hM : M.IsLB) (hc : ⨆ i, D i = ⊤) (hA : ∀ i, IsAffineOpen (D i)) (h : Function.Exact (Cube.d (secCube f M D) 1) (Cube.d (secCube f M D) 2)) : FirstCohomologyVanishes M

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FlatHomLocOfFlatHomLocSingleton

private theorem **flat_loc_over_algebra** {A S M N : Type} [CommRing A] [CommRing S] [Algebra A S] [AddCommGroup M] [Module A M] [Module S M] [IsScalarTower A S M] [AddCommGroup N] [Module A N] [Module S N] [IsScalarTower A S N] (T : Submonoid S) (f : M →ₗ[S] N) [IsLocalizedModule T f] (h : Module.Flat A M) : Module.Flat A N

private theorem **flat_iff_forall_piece** {R M : Type} [CommRing R] [AddCommGroup M] [Module R M] (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] : Module.Flat R M ↔ ∀ n, Module.Flat R (𝒜 n)

theorem **flat_homLoc_of_flat_homLoc_singleton** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] [Decomposition 𝒜] (i : ι) (s : Finset ι) (n : ℤ), i ∈ s → Module.Flat R (homLoc 𝒜 {i} 0) → Module.Flat R (homLoc 𝒜 s n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FlatQuotSMulTopCokerOfMatrixWitnesses

theorem **flat_quotSMulTop_coker_of_matrix_witnesses** : ∀ (R S : Type) [CommRing R] [CommRing S] [Algebra R S] (r a b : ℕ) (g : Fin r → S) (u : S) (Φ : Matrix (Fin a) (Fin b) S), Module.Flat R (S ⧸ Ideal.span (Set.range g)) → (∃ c : Fin r → S, u = ∑ k, c k * g k) → (∀ k, ∃ (C : Matrix (Fin b) (Fin a) S) (C' : Matrix (Fin a) (Fin a) S), g k • (1 : Matrix (Fin a) (Fin a) S) = Φ * C + u • C') → (∃ (Ψ : Matrix (Fin b) (Fin a) S) (E : Fin r → Matrix (Fin a) (Fin b) S), Φ * Ψ * Φ = Φ + ∑ k, g k • E k) → Module.Flat R (QuotSMulTop u ((Fin a → S) ⧸ LinearMap.range Φ.mulVecLin))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FreeYonedaHomEquivFreeToPullbackComp

lemma **extMap_apply** {Z F : A} {Z' G : B} (a : Z' ⟶ Φ.obj Z) (b : Φ.obj F ⟶ G) (n : ℕ) (x : Ext.{w} Z F n) : extMap.{w, w'} Φ a b n x = (Ext.mk₀ a).comp ((x.mapExactFunctor Φ).comp (Ext.mk₀ b) (add_zero n)) (zero_add n)

lemma **extMap_mk₀** {Z F : A} {Z' G : B} (a : Z' ⟶ Φ.obj Z) (b : Φ.obj F ⟶ G) (g : Z ⟶ F) : extMap.{w, w'} Φ a b 0 (Ext.mk₀ g) = Ext.mk₀ (a ≫ Φ.map g ≫ b)

lemma **extMap_comp_right** {Z F F' : A} {Z' G : B} (a : Z' ⟶ Φ.obj Z) (b : Φ.obj F' ⟶ G) (s : F ⟶ F') (n : ℕ) (x : Ext.{w} Z F n) : extMap.{w, w'} Φ a b n (x.comp (Ext.mk₀ s) (add_zero n)) = extMap.{w, w'} Φ a (Φ.map s ≫ b) n x

lemma **comp_extMap_right** {Z F : A} {Z' G G' : B} (a : Z' ⟶ Φ.obj Z) (b : Φ.obj F ⟶ G) (t : G ⟶ G') (n : ℕ) (x : Ext.{w} Z F n) : (extMap.{w, w'} Φ a b n x).comp (Ext.mk₀ t) (add_zero n) = extMap.{w, w'} Φ a (b ≫ t) n x

lemma **extMap_comp_left** {Z Z₁ F : A} {Z' G : B} (a : Z' ⟶ Φ.obj Z₁) (b : Φ.obj F ⟶ G) (s : Z₁ ⟶ Z) (n : ℕ) (x : Ext.{w} Z F n) : extMap.{w, w'} Φ a b n ((Ext.mk₀ s).comp x (zero_add n)) = extMap.{w, w'} Φ (a ≫ Φ.map s) b n x

lemma **comp_extMap_left** {Z F : A} {Z' Z₁ G : B} (a : Z' ⟶ Φ.obj Z) (b : Φ.obj F ⟶ G) (t : Z₁ ⟶ Z') (n : ℕ) (x : Ext.{w} Z F n) : (Ext.mk₀ t).comp (extMap.{w, w'} Φ a b n x) (zero_add n) = extMap.{w, w'} Φ (t ≫ a) b n x

private lemma **comp_comp_mk0_eq_comp_comp_of_square** {X₀ X₁ X₂ X₃ Y₂ Y₃ : B} {n : ℕ} (a : Ext.{w'} X₀ X₁ 0) (y : Ext.{w'} X₁ X₂ n) (e : Ext.{w'} X₂ X₃ 1) (t₃ : X₃ ⟶ Y₃) (t₂ : X₂ ⟶ Y₂) (f : Ext.{w'} Y₂ Y₃ 1) (h : e.comp (Ext.mk₀ t₃) (add_zero 1) = (Ext.mk₀ t₂).comp f (zero_add 1)) : a.comp ((y.comp e rfl).comp (Ext.mk₀ t₃) (add_zero _)) (zero_add _) = (a.comp (y.comp (Ext.mk₀ t₂) (add_zero n)) (zero_add n)).comp f rfl

lemma **extMap_extClass** {Z : A} {Z' : B} (a : Z' ⟶ Φ.obj Z) {S : ShortComplex A} {T : ShortComplex B} (hS : S.ShortExact) (hT : T.ShortExact) (φ : S.map Φ ⟶ T) (n : ℕ) (x : Ext.{w} Z S.X₃ n) : extMap.{w, w'} Φ a (F

lemma **freeYonedaHomEquiv_freeToPullback_comp** {V : C} {W : D} (i : W ⟶ p.obj V) {G : Sheaf K AddCommGrpCat.{v}} (b : (p.sheafPullback AddCommGrpCat.{v} J K).obj ((freeYoneda J).obj V) ⟶ G) : freeYonedaHomEquiv W G (freeToPullback J K p i ≫ b) = G.obj.map i.op (freeYonedaHomEquiv V _ ((p.sheafAdjunctionContinuous AddCommGrpCat.{v} J K).homEquiv _ _ b))

lemma **map_comp_freeToPullback** {V : C} {W W' : D} (i : W ⟶ p.obj V) (k : W' ⟶ W) : (freeYoneda K).map k ≫ freeToPullback J K p i = freeToPullback J K p (k ≫ i)

lemma **freeToPullback_comp_map** {V V' : C} {W : D} (i : W ⟶ p.obj V') (k : V' ⟶ V) : freeToPullback J K p i ≫ (p.sheafPullback AddCommGrpCat.{v} J K).map ((freeYoneda J).map k) = freeToPullback J K p (i ≫ p.map k)

lemma **equivZero_sitePull** (φ : F ⟶ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).obj G) {V : C} {W : D} (i : W ⟶ p.obj V) (x : Sheaf.H'.{w} F 0 V) : equivZero G W (sitePull.{w, w'} p φ 0 i x) = G.obj.map i.op (φ.hom.app (op V) (equivZero F V x))

lemma **sitePull_map** (φ : F' ⟶ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).obj G) (s : F ⟶ F') (n : ℕ) {V : C} {W : D} (i : W ⟶ p.obj V) (x : Sheaf.H'.{w} F n V) : sitePull.{w, w'} p φ n i (map s n V x) = sitePull.{w, w'} p (s ≫ φ) n i x

lemma **map_sitePull** (φ : F ⟶ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).obj G) (t : G ⟶ G') (n : ℕ) {V : C} {W : D} (i : W ⟶ p.obj V) (x : Sheaf.H'.{w} F n V) : map t n W (sitePull.{w, w'} p φ n i x) = sitePull.{w, w'} p (φ ≫ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).map t) n i x

lemma **sitePull_res** (φ : F ⟶ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).obj G) (n : ℕ) {V V' : C} {W : D} (i : W ⟶ p.obj V') (k : V' ⟶ V) (x : Sheaf.H'.{w} F n V) : sitePull.{w, w'} p φ n i (res F n k x) = sitePull.{w, w'} p φ n (i ≫ p.map k) x

lemma **res_sitePull** (φ : F ⟶ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).obj G) (n : ℕ) {V : C} {W W' : D} (i : W ⟶ p.obj V) (k : W' ⟶ W) (x : Sheaf.H'.{w} F n V) : res G n k (sitePull.{w, w'} p φ n i x) = sitePull.{w, w'} p φ n (k ≫ i) x

lemma **sitePull_δ** {S : ShortComplex (Sheaf J AddCommGrpCat.{v})} {T : ShortComplex (Sheaf K AddCommGrpCat.{v})} (hS : S.ShortExact) (hT : T.ShortExact) (φ : S ⟶ T.map (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K)) (n : ℕ) {V : C} {W : D} (i : W ⟶ p.obj V) (x : Sheaf.H'.{w} S.X₃ n V) : sitePull.{w, w'} p (G

lemma **sitePull_bijective_succ** {S : ShortComplex (Sheaf J AddCommGrpCat.{v})} {T : ShortComplex (Sheaf K AddCommGrpCat.{v})} (hS : S.ShortExact) (hT : T.ShortExact) (φ : S ⟶ T.map (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K)) (n : ℕ) {V : C} {W : D} (i : W ⟶ p.obj V) (h₂ : Function.Surjective (sitePull.{w, w'} p (G

lemma **sitePull_bijective_zero** (φ : F ⟶ (p.sheafPushforwardContinuous AddCommGrpCat.{v} J K).obj G) {V : C} {W : D} (i : W ⟶ p.obj V) (h : Function.Bijective fun s : F.obj.obj (op V) ↦ G.obj.map i.op (φ.hom.app (op V) s)) : Function.Bijective (sitePull.{w, w'} p φ 0 i)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.FstHomCompSqzOfLeibniz

theorem **sqzPoint_apply** (φ : O →+* C) (n : ι → C) (hn : ∀ a b, n a * n b = 0) (x : TrivSqZeroExt O (ι → O)) : sqzPoint φ n hn x = φ x.fst + ∑ i, φ (x.snd i) * n i

@[simp] theorem **sqzPoint_inl** (φ : O →+* C) (n : ι → C) (hn : ∀ a b, n a * n b = 0) (a : O) : sqzPoint φ n hn (inl a) = φ a

@[simp] theorem **sqzPoint_inr** (φ : O →+* C) (n : ι → C) (hn : ∀ a b, n a * n b = 0) (v : ι → O) : sqzPoint φ n hn (inr v) = ∑ i, φ (v i) * n i

theorem **leibniz_map_one** (ε : A →+* O) (d : A →+ M) (hd : ∀ f f', d (f * f') = ε f • d f' + ε f' • d f) : d 1 = 0

@[simp] theorem **sqzOfLeibniz_fst** (ε : A →+* O) (d : A →+ M) (hd : ∀ f f', d (f * f') = ε f • d f' + ε f' • d f) (f : A) : (sqzOfLeibniz ε d hd f).fst = ε f

@[simp] theorem **sqzOfLeibniz_snd** (ε : A →+* O) (d : A →+ M) (hd : ∀ f f', d (f * f') = ε f • d f' + ε f' • d f) (f : A) : (sqzOfLeibniz ε d hd f).snd = d f

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.HigherDirectImagesVanishOfIsAffineHom

theorem **SheafPull.pull_map** (s : F' ⟶ F) (i : W ⟶ (Opens.map f).obj V) (x : Sheaf.H'.{u} F' n V) : pull.{u, u} f φ n i (map s n V x) = pull.{u, u} f (s ≫ φ) n i x

theorem **SheafPull.map_pull** (s : G ⟶ G') (i : W ⟶ (Opens.map f).obj V) (x : Sheaf.H'.{u} F n V) : map s n W (pull.{u, u} f φ n i x) = pull.{u, u} f (φ ≫ (push f).map s) n i x

theorem **SheafPull.pull_res** (k : V' ⟶ V) (i : W ⟶ (Opens.map f).obj V') (x : Sheaf.H'.{u} F n V) : pull.{u, u} f φ n i (res F n k x) = pull.{u, u} f φ n (i ≫ (Opens.map f).map k) x

theorem **SheafPull.res_pull** (k : W' ⟶ W) (i : W ⟶ (Opens.map f).obj V) (x : Sheaf.H'.{u} F n V) : res G n k (pull.{u, u} f φ n i x) = pull.{u, u} f φ n (k ≫ i) x

theorem **SheafPull.equivZero_pull** (i : W ⟶ (Opens.map f).obj V) (x : Sheaf.H'.{u} F 0 V) : equivZero G W (pull.{u, u} f φ 0 i x) = G.obj.map i.op (φ.hom.app (op V) (equivZero F V x))

theorem **SheafPull.pull_δ** {A : ShortComplex (AbSheaf X)} {B : ShortComplex (AbSheaf Y)} (hA : A.ShortExact) (hB : B.ShortExact) (ψ : A ⟶ B.map (push f)) (i : W ⟶ (Opens.map f).obj V) (x : Sheaf.H'.{u} A.X₃ n V) : pull.{u, u} f (G

theorem **smulHom_pushforward** (g : Y ⟶ X) (G : Y.Modules) (r : Γ(X, ⊤)) : smulHom ((Scheme.Modules.pushforward g).obj G) r = (Scheme.Modules.pushforward g).map (smulHom G (g.appTop r))

theorem **toAbSheafPush_comp** (s : F' ⟶ F) (φ : F ⟶ (Scheme.Modules.pushforward g).obj G) : toAbSheafPush (s ≫ φ) = (toAbSheaf X).map s ≫ toAbSheafPush φ

theorem **toAbSheafPush_comp_map** (φ : F ⟶ (Scheme.Modules.pushforward g).obj G) (s : G ⟶ G') : toAbSheafPush (φ ≫ (Scheme.Modules.pushforward g).map s) = toAbSheafPush φ ≫ (SheafPull.push g.base).map ((toAbSheaf Y).map s)

theorem **cohOn.toH'_pull** (h : W ≤ g ⁻¹ᵁ V) (x : cohOn F V n) : cohOn.toH' G W n (cohOn.pull hσ φ h n x) = SheafPull.pull.{u, u} g.base (toAbSheafPush φ) n (homOfLE h) (cohOn.toH' F V n x)

theorem **cohOn.pull_map** (s : F' ⟶ F) (h : W ≤ g ⁻¹ᵁ V) (x : cohOn F' V n) : cohOn.pull hσ φ h n (cohOn.map R s V n x) = cohOn.pull hσ (s ≫ φ) h n x

theorem **cohOn.map_pull** (s : G ⟶ G') (h : W ≤ g ⁻¹ᵁ V) (x : cohOn F V n) : cohOn.map S s W n (cohOn.pull hσ φ h n x) = cohOn.pull hσ (φ ≫ (Scheme.Modules.pushforward g).map s) h n x

theorem **cohOn.pull_res** (k : V' ≤ V) (h : W ≤ g ⁻¹ᵁ V') (x : cohOn F V n) : cohOn.pull hσ φ h n (cohOn.res R F k n x) = cohOn.pull hσ φ (h.trans (g.preimage_mono k)) n x

theorem **cohOn.res_pull** (k : W' ≤ W) (h : W ≤ g ⁻¹ᵁ V) (x : cohOn F V n) : cohOn.res S G k n (cohOn.pull hσ φ h n x) = cohOn.pull hσ φ (k.trans h) n x

theorem **cohOn.zeroEquiv_pull** (h : W ≤ g ⁻¹ᵁ V) (x : cohOn F V 0) : cohOn.zeroEquiv S G W (cohOn.pull hσ φ h 0 x) = sectionsOn.res S G h (sectionsOn.mapOver hσ φ V (cohOn.zeroEquiv R F V x))

theorem **cohOn.pull_δ** {A : ShortComplex X.Modules} {B : ShortComplex Y.Modules} (hA : A.ShortExact) (hB : B.ShortExact) (ψ : A ⟶ B.map (Scheme.Modules.pushforward g)) (h : W ≤ g ⁻¹ᵁ V) (x : cohOn A.X₃ V n) : cohOn.pull hσ (G

theorem **cohOn.pull_bijective_of_toAbSheaf** (h : W ≤ g ⁻¹ᵁ V) (H : Function.Bijective (SheafPull.pull.{u, u} g.base (toAbSheafPush φ) n (homOfLE h))) : Function.Bijective (cohOn.pull hσ φ h n)

theorem **higherDirectImagesVanish_of_isAffineHom** (g : Y ⟶ X) [IsAffineHom g] (G : Y.Modules) [G.IsQuasicoherent] : SheafPull.HigherDirectImagesVanish.{u} g.base ((toAbSheaf Y).obj G)

theorem **cohOn.pull_id_bijective** [IsAffineHom g] [G.IsQuasicoherent] : Function.Bijective (cohOn.pull hσ (𝟙 ((Scheme.Modules.pushforward g).obj G)) (le_refl (g ⁻¹ᵁ V)) n)

theorem **cohOn.pull_restrict_bijective** [IsOpenImmersion g] (h : W ≤ g ⁻¹ᵁ V) (h' : V ≤ g ''ᵁ W) : Function.Bijective (cohOn.pull hσ (G

theorem **cohOn.pushforwardEquiv_apply** (V : X.Opens) (n : ℕ) (x : cohOn ((Scheme.Modules.pushforward g).obj G) V n) : cohOn.pushforwardEquiv R hg G V n x = cohOn.pull (σ

theorem **cohOn.pushforwardEquiv_map** {G' : Y.Modules} [G'.IsQuasicoherent] (s : G ⟶ G') (V : X.Opens) (n : ℕ) (x : cohOn ((Scheme.Modules.pushforward g).obj G) V n) : cohOn.pushforwardEquiv R hg G' V n (cohOn.map R ((Scheme.Modules.pushforward g).map s) V n x) = cohOn.map R s (g ⁻¹ᵁ V) n (cohOn.pushforwardEquiv R hg G V n x)

theorem **cohOn.zeroEquiv_pushforwardEquiv** (V : X.Opens) (x : cohOn ((Scheme.Modules.pushforward g).obj G) V ∅) : sectionsOn.toΓ G (g ⁻¹ᵁ V) (cohOn.zeroEquiv R G _ (cohOn.pushforwardEquiv R hg G V ∅ x)) = sectionsOn.toΓ _ V (cohOn.zeroEquiv R _ V x)

theorem **coh.pushforwardEquiv_def** (n : ℕ) : coh.pushforwardEquiv R hg G n = cohOn.pushforwardEquiv R hg G ⊤ n

theorem **coh.pushforwardEquiv_map** {G' : Y.Modules} [G'.IsQuasicoherent] (s : G ⟶ G') (n : ℕ) (x : coh ((Scheme.Modules.pushforward g).obj G) n) : coh.pushforwardEquiv R hg G' n (cohOn.map R ((Scheme.Modules.pushforward g).map s) ⊤ n x) = cohOn.map R s ⊤ n (coh.pushforwardEquiv R hg G n x)

theorem **hdim_pushforward** (k : Type v) [Field k] [Algebra k Γ(X, ⊤)] [Algebra k Γ(Y, ⊤)] (hg : ∀ r, g.appTop (algebraMap k Γ(X, ⊤) r) = algebraMap k Γ(Y, ⊤) r) [IsAffineHom g] (G : Y.Modules) [G.IsQuasicoherent] (n : ℕ) : hdim k ((Scheme.Modules.pushforward g).obj G) n = hdim k G n

theorem **appTop_algebraΓOfHom** {R : Type u} [CommRing R] (g : Y ⟶ X) (p : X ⟶ Spec (.of R)) {q : Y ⟶ Spec (.of R)} (w : g ≫ p = q) (r : R) : letI

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.Imports

theorem **addVal_prod** : ∀ {R : Type u_1} [inst : CommRing R] [inst_1 : IsDomain R] [inst_2 : IsDiscreteValuationRing R] {ι : Type u_2} (s : Finset ι) (a : ι → R), (IsDiscreteValuationRing.addVal R : R → ℕ∞) (∏ i ∈ s, a i) = ∑ i ∈ s, (IsDiscreteValuationRing.addVal R : R → ℕ∞) (a i)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IntegralAgainstLinearApply

theorem **integrable_smul** (μ : Measure K) [IsFiniteMeasure μ] (p : C(K, B)) (f : C(K, ℂ)) : Integrable (fun k => f k • p k) μ

@[simp] theorem **integralAgainstLinear_apply** (μ : Measure K) [IsFiniteMeasure μ] (p : C(K, B)) (f : C(K, ℂ)) : integralAgainstLinear μ p f = ∫ k, f k • p k ∂μ

theorem **norm_integral_smul_le** (μ : Measure K) [IsFiniteMeasure μ] (p : C(K, B)) (f : C(K, ℂ)) : ‖∫ k, f k • p k ∂μ‖ ≤ (μ.real Set.univ * ‖p‖) * ‖f‖

@[simp] theorem **integralAgainst_apply** (μ : Measure K) [IsFiniteMeasure μ] (p : C(K, B)) (f : C(K, ℂ)) : integralAgainst μ p f = ∫ k, f k • p k ∂μ

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsAffineHomOfIsPullbackSpecMap

theorem **baseRingHom_res** (p : X ⟶ Spec R) {U V : X.Opens} (h : V ≤ U) (r : R) : X.presheaf.map (homOfLE h).op (baseRingHom p U r) = baseRingHom p V r

theorem **baseRingHom_top** (p : X ⟶ Spec R) : baseRingHom p ⊤ = (Scheme.ΓSpecIso R).inv ≫ p.appTop

theorem **baseRingHom_comp_app** (g : Y ⟶ X) (p : X ⟶ Spec R) (U : X.Opens) : baseRingHom p U ≫ g.app U = baseRingHom (g ≫ p) (g ⁻¹ᵁ U)

theorem **app_baseRingHom** (g : Y ⟶ X) (p : X ⟶ Spec R) (U : X.Opens) (r : R) : g.app U (baseRingHom p U r) = baseRingHom (g ≫ p) (g ⁻¹ᵁ U) r

theorem **baseRingHom_comp_SpecMap** {L : CommRingCat.{u}} (q : Y ⟶ Spec L) (ι : R ⟶ L) (V : Y.Opens) : baseRingHom (q ≫ Spec.map ι) V = ι ≫ baseRingHom q V

theorem **appLinearBase_apply** (p : X ⟶ Spec R) (g : Y ⟶ X) {F : X.Modules} {G : Y.Modules} (φ : F ⟶ (Scheme.Modules.pushforward g).obj G) (U : X.Opens) (s : Γ(F, U)) : appLinearBase p g φ U s = φ.app U s

theorem **baseRingHom_algebraMap** (V : Y.Opens) (r : R) : baseRingHom q V (algebraMap R L r) = baseRingHom (g ≫ p) V r

theorem **isAffineHom_of_isPullback_SpecMap** : IsAffineHom g

theorem **isLocalization_app** {V : X.Opens} (hV : IsAffineOpen V) : letI

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsAffineOpenBiInfOfLe

theorem **isAffineOpen_biInf_of_le** (E : ν → Y.Opens) (hE : ∀ i V, IsAffineOpen V → IsAffineOpen (E i ⊓ V)) (u : Finset ν) {V : Y.Opens} (hV : IsAffineOpen V) (h : ⊓ i ∈ u, E i ≤ V) : IsAffineOpen (⊓ i ∈ u, E i)

theorem **acyclic_comap_secCube** [LinearOrder ν] [LinearOrder κ] [Fintype κ] (hM : M.IsLB) (E : ν → Y.Opens) (hE : ∀ i V, IsAffineOpen V → IsAffineOpen (E i ⊓ V)) (e : κ ↪o ν) (hc : ⊔ k, E (e k) = ⊤) (s₀ : Finset ν) {V : Y.Opens} (hV : IsAffineOpen V) (h : ⊓ i ∈ s₀, E i ≤ V) (g : Finset κ → Finset ν) (hg : Monotone g) (hgs : ∀ t, g t = s₀ ∪ t.map e.toEmbedding) : (cubeComplex (Cube.comap g hg (secCube f M E))).Acyclic

theorem **biInf_sumVertex** [LinearOrder ν] [LinearOrder κ] (E : ν ⊕ₗ κ → Y.Opens) (s : Finset ν) (t : Finset κ) : ⊓ i ∈ Cube.sumVertex s t, E i = (⊓ i ∈ s, E (toLex (Sum.inl i))) ⊓ ⊓ k ∈ t, E (toLex (Sum.inr k))

theorem **cechLB_homology_equiv_trunc₂** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (κ₁ : Type) [LinearOrder κ₁] [Fintype κ₁] (U : κ₁ → Y.Opens) (κ₂ : Type) [LinearOrder κ₂] [Fintype κ₂] (W : κ₂ → Y.Opens), M.IsLB → IsSeparated f → (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → ⊔ i, U i = ⊤ → ⊔ j, W j = ⊤ → (∀ i V, IsAffineOpen V → IsAffineOpen (U i ⊓ V)) → (∀ j V, IsAffineOpen V → IsAffineOpen (W j ⊓ V)) → (∀ i j, IsAffineOpen (U i ⊓ W j)) → ∀ n, Nonempty ((cechLB f M D).homology n ≃ₗ[R] (cubeComplex (Cube.trunc₂ (secCube f M fun u : κ₁ ⊕ₗ κ₂ => Sum.elim U W (ofLex u)))).homology (n + 2))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsBaseChangeAwayOfIntertwine

theorem **exists_awayFrac_common** {f : A₀} (z w : LocalizedModule.Away f N₀) : ∃ k m m', awayFrac N₀ f k m = z ∧ awayFrac N₀ f k m' = w

theorem **exists_away_map_of_intertwine** (ψ : N₀ →ₗ[R₀] N) (x₀ : A₀) (x : A) (h : ∀ (k : ℕ) y, ψ (x₀ ^ k • y) = x ^ k • ψ y) : ∃ θ : LocalizedModule.Away x₀ N₀ →ₗ[R₀] LocalizedModule.Away x N, ∀ (k : ℕ) y, θ (awayFrac N₀ x₀ k y) = awayFrac N x k (ψ y)

theorem **isBaseChange_away_of_intertwine** (ψ : N₀ →ₗ[R₀] N) (hψ : IsBaseChange R ψ) (x₀ : A₀) (x : A) (h : ∀ (k : ℕ) y, ψ (x₀ ^ k • y) = x ^ k • ψ y) (θ : LocalizedModule.Away x₀ N₀ →ₗ[R₀] LocalizedModule.Away x N) (hθ : ∀ (k : ℕ) y, θ (awayFrac N₀ x₀ k y) = awayFrac N x k (ψ y)) : IsBaseChange R θ

theorem **exists_isBaseChange_away_xProd** : ∀ (ι : Type) (R₀ R : Type) [CommRing R₀] [CommRing R] [Algebra R₀ R] (N₀ N : Type) [AddCommGroup N₀] [Module R₀ N₀] [Module (MvPolynomial ι R₀) N₀] [IsScalarTower R₀ (MvPolynomial ι R₀) N₀] [AddCommGroup N] [Module R N] [Module (MvPolynomial ι R) N] [IsScalarTower R (MvPolynomial ι R) N] [Module R₀ N] [IsScalarTower R₀ R N] [IsScalarTower R₀ (MvPolynomial ι R) N] (ψ : N₀ →ₗ[R₀] N) (s : Finset ι), IsBaseChange R ψ → (∀ (i : ι) y, ψ ((X i : MvPolynomial ι R₀) • y) = (X i : MvPolynomial ι R) • ψ y) → ∃ θ : LocalizedModule.Away (xProd R₀ s) N₀ →ₗ[R₀] LocalizedModule.Away (xProd R s) N, IsBaseChange R θ ∧ ∀ (k : ℕ) y, θ (awayFrac N₀ (xProd R₀ s) k y) = awayFrac N (xProd R s) k (ψ y)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsBaseChangeCokerMatrixMvPolynomial

theorem **exists_isBaseChange_coker** {R₀ R A₀ A M₁ M₂ N₁ N₂ : Type*} [CommRing R₀] [CommRing R] [Algebra R₀ R] [Ring A₀] [Ring A] [Algebra R₀ A₀] [Algebra R A] [SMul R₀ A] [AddCommGroup M₁] [AddCommGroup M₂] [AddCommGroup N₁] [AddCommGroup N₂] [Module R₀ M₁] [Module R₀ M₂] [Module A₀ M₁] [Module A₀ M₂] [IsScalarTower R₀ A₀ M₁] [IsScalarTower R₀ A₀ M₂] [Module R₀ N₁] [Module R₀ N₂] [Module R N₁] [Module R N₂] [Module A N₁] [Module A N₂] [IsScalarTower R₀ R N₁] [IsScalarTower R₀ R N₂] [IsScalarTower R A N₁] [IsScalarTower R A N₂] [IsScalarTower R₀ A N₂] (f₀ : M₁ →ₗ[A₀] M₂) (f : N₁ →ₗ[A] N₂) (h₁ : M₁ →ₗ[R₀] N₁) (h₂ : M₂ →ₗ[R₀] N₂) (b₁ : IsBaseChange R h₁) (b₂ : IsBaseChange R h₂) (hcomm : ∀ x, h₂ (f₀ x) = f (h₁ x)) : ∃ ψ : (M₂ ⧸ LinearMap.range f₀) →ₗ[R₀] (N₂ ⧸ LinearMap.range f), IsBaseChange R ψ ∧ ∀ y, ψ (Submodule.Quotient.mk y) = Submodule.Quotient.mk (h₂ y)

theorem **isBaseChange_coker_matrix_mvPolynomial** : ∀ (ι : Type) (R₀ R : Type) [CommRing R₀] [CommRing R] [Algebra R₀ R] (a b : ℕ) (Φ₀ : Matrix (Fin a) (Fin b) (MvPolynomial ι R₀)) (Φ : Matrix (Fin a) (Fin b) (MvPolynomial ι R)), Φ₀.map (MvPolynomial.map (algebraMap R₀ R)) = Φ → ∃ ψ : ((Fin a → MvPolynomial ι R₀) ⧸ LinearMap.range Φ₀.mulVecLin) →ₗ[R₀] ((Fin a → MvPolynomial ι R) ⧸ LinearMap.range Φ.mulVecLin), IsBaseChange R ψ ∧ ∀ y : Fin a → MvPolynomial ι R₀, ψ (Submodule.Quotient.mk y) = Submodule.Quotient.mk fun l => MvPolynomial.map (algebraMap R₀ R) (y l)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsBaseChangeGradedPiece

theorem **isBaseChange_gradedPiece** : ∀ (R₀ R : Type) [CommRing R₀] [CommRing R] [Algebra R₀ R] (N₀ N : Type) [AddCommGroup N₀] [Module R₀ N₀] [AddCommGroup N] [Module R₀ N] [Module R N] [IsScalarTower R₀ R N] (𝒩₀ : ℤ → Submodule R₀ N₀) [Decomposition 𝒩₀] (𝒩 : ℤ → Submodule R N) [Decomposition 𝒩] (ψ : N₀ →ₗ[R₀] N) (n : ℤ) (θ : 𝒩₀ n →ₗ[R₀] 𝒩 n), IsBaseChange R ψ → (∀ j, ∀ y ∈ 𝒩₀ j, ψ y ∈ 𝒩 j) → (∀ y, (θ y : N) = ψ y) → IsBaseChange R θ

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsCounitDerivationCounitComp

theorem **isCounitDerivation_iff** (φ : H →ₗ[k] k) : IsCounitDerivation φ ↔ ∀ a b : H, φ (a ∗ b) = Coalgebra.counit (R

@[simp] theorem **contrRight_tmul** (φ : H →ₗ[k] k) (a b : H) : contrRight φ (a ⊗ₜ[k] b) = φ b • a

@[simp] theorem **contrLeft_tmul** (φ : H →ₗ[k] k) (a b : H) : contrLeft φ (a ⊗ₜ[k] b) = φ a • b

theorem **contrRight_counit_comul** (a : H) : contrRight (Coalgebra.counit (R

theorem **contrLeft_counit_comul** (a : H) : contrLeft (Coalgebra.counit (R

theorem **contrRight_mul** {φ : H →ₗ[k] k} (hφ : IsCounitDerivation φ) (s t : H ⊗[k] H) : contrRight φ (s ∗ t) = contrRight (Coalgebra.counit (R

theorem **contrLeft_mul** {φ : H →ₗ[k] k} (hφ : IsCounitDerivation φ) (s t : H ⊗[k] H) : contrLeft φ (s ∗ t) = contrLeft (Coalgebra.counit (R

theorem **contrRight_one** (φ : H →ₗ[k] k) : contrRight φ 1 = φ 1 • (1 : H)

theorem **contrLeft_one** (φ : H →ₗ[k] k) : contrLeft φ 1 = φ 1 • (1 : H)

theorem **counit_contrRight** (φ : H →ₗ[k] k) (s : H ⊗[k] H) : Coalgebra.counit (R

theorem **counit_contrLeft** (φ : H →ₗ[k] k) (s : H ⊗[k] H) : Coalgebra.counit (R

theorem **IsCounitDerivation.map_one** {φ : H →ₗ[k] k} (hφ : IsCounitDerivation φ) : φ 1 = 0

theorem **derivRight_apply** (φ : H →ₗ[k] k) (hφ : IsCounitDerivation φ) (y : H) : derivRight φ hφ y = TensorProduct.rid k H (LinearMap.lTensor H φ (Coalgebra.comul (R

theorem **derivLeft_apply** (φ : H →ₗ[k] k) (hφ : IsCounitDerivation φ) (y : H) : derivLeft φ hφ y = TensorProduct.lid k H (LinearMap.rTensor H φ (Coalgebra.comul (R

theorem **counit_derivRight** (φ : H →ₗ[k] k) (hφ : IsCounitDerivation φ) (y : H) : Coalgebra.counit (R

theorem **counit_derivLeft** (φ : H →ₗ[k] k) (hφ : IsCounitDerivation φ) (y : H) : Coalgebra.counit (R

theorem **isCounitDerivation_counit_comp** (D : Derivation k H H) : IsCounitDerivation (Coalgebra.counit (R

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsDiscreteValuationRingLocalizationOfIsDedekindD

lemma **length_quot_span_singleton** (a : R) : Module.length R (R ⧸ Ideal.span {a}) = addVal R a

lemma **addVal_prod** {ι : Type*} (s : Finset ι) (a : ι → R) : addVal R (∏ i ∈ s, a i) = ∑ i ∈ s, addVal R (a i)

lemma **addVal_det_of_injective** (f : M →ₗ[R] M) (hf : Function.Injective f) : addVal R (LinearMap.det f) = Module.length R (M ⧸ LinearMap.range f)

lemma **length_quot_range_of_not_injective** (f : M →ₗ[R] M) (hf : ¬ Function.Injective f) : Module.length R (M ⧸ LinearMap.range f) = ⊤

lemma **addVal_det** (f : M →ₗ[R] M) : addVal R (LinearMap.det f) = Module.length R (M ⧸ LinearMap.range f)

lemma **liesOver_maximalIdeal** (n : Ideal S) [n.IsMaximal] : n.LiesOver (maximalIdeal R)

lemma **length_quot_maximal** (n : Ideal S) [n.IsMaximal] : Module.length R (S ⧸ n) = n.inertiaDeg R

lemma **isLocalHom_algebraMap** (n : Ideal S) [n.IsMaximal] (B : Type*) [CommRing B] [IsLocalRing B] [Algebra S B] [IsLocalization.AtPrime B n] [Algebra R B] [IsScalarTower R S B] : IsLocalHom (algebraMap R B)

lemma **inertiaDeg_eq_finrank_residueField** (n : Ideal S) [n.IsMaximal] (B : Type*) [CommRing B] [IsLocalRing B] [Algebra S B] [IsLocalization.AtPrime B n] [Algebra R B] [IsScalarTower R S B] [IsLocalHom (algebraMap R B)] : n.inertiaDeg R = Module.finrank (ResidueField R) (ResidueField B)

lemma **finite_residueField** (n : Ideal S) [n.IsMaximal] (B : Type*) [CommRing B] [IsLocalRing B] [Algebra S B] [IsLocalization.AtPrime B n] [Algebra R B] [IsScalarTower R S B] [IsLocalHom (algebraMap R B)] : Module.Finite (ResidueField R) (ResidueField B)

lemma **finite_maximalSpectrum** : Finite (MaximalSpectrum S)

lemma **length_localization_quot_maximal** (m n : MaximalSpectrum S) [Decidable (n = m)] (B : Type*) [CommRing B] [Algebra S B] [IsLocalization.AtPrime B n.asIdeal] : Module.length B (B ⧸ m.asIdeal.map (algebraMap S B)) = if n = m then 1 else 0

lemma **length_eq_sum_tensor** {X : Type*} [AddCommGroup X] [Module S X] (hX : IsFiniteLength S X) : ∀ [Module R X] [IsScalarTower R S X], Module.length R X = ∑ n : MaximalSpectrum S, Module.length (A n) (A n ⊗[S] X) * n.asIdeal.inertiaDeg R

lemma **length_quot_eq_sum** (I : Ideal S) (hI : Module.length R (S ⧸ I) ≠ ⊤) : Module.length R (S ⧸ I) = ∑ n : MaximalSpectrum S, Module.length (A n) (A n ⧸ I.map (algebraMap S (A n))) * n.asIdeal.inertiaDeg R

theorem **addVal_norm_eq_length** (x : S) : addVal R (Algebra.norm R x) = Module.length R (S ⧸ Ideal.span {x})

theorem **addVal_norm_ne_top_iff** (x : S) : addVal R (Algebra.norm R x) ≠ ⊤ ↔ x ∈ nonZeroDivisors S

theorem **length_quot_span_ne_top_iff** (x : S) : Module.length R (S ⧸ Ideal.span {x}) ≠ ⊤ ↔ x ∈ nonZeroDivisors S

theorem **addVal_norm_eq_sum_length** {x : S} (hx : x ∈ nonZeroDivisors S) : addVal R (Algebra.norm R x) = ∑ n : MaximalSpectrum S, Module.length (A n) (A n ⧸ Ideal.span {algebraMap S (A n) x}) * n.asIdeal.inertiaDeg R

theorem **addVal_norm_eq_sum** (x : S) : addVal R (Algebra.norm R x) = ∑ n : MaximalSpectrum S, addVal (A n) (algebraMap S (A n) x) * n.asIdeal.inertiaDeg R

theorem **addVal_localization_ne_top** {x : S} (hx : x ∈ nonZeroDivisors S) (n : MaximalSpectrum S) : addVal (A n) (algebraMap S (A n) x) ≠ ⊤

theorem **isDiscreteValuationRing_localization_of_isDedekindDomain** [IsDedekindDomain S] (n : MaximalSpectrum S) : IsDiscreteValuationRing (Localization.AtPrime n.asIdeal)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsFiniteLengthOfIsArtinianRingAnnihilator

theorem **exists_finite_specializes_of_isClosed** {Z : Set T} (hZ : IsClosed Z) : ∃ S : Set T, S.Finite ∧ S ⊆ Z ∧ ∀ x ∈ Z, ∃ y ∈ S, y ⤳ x

theorem **exists_cover_of_topologicalKrullDim_lt** (P : Opens T → Prop) (hP : ∀ S : Set T, S.Finite → ∃ A, P A ∧ S ⊆ A) (n : ℕ) {Z : Set T} (hZ : IsClosed Z) (hn : topologicalKrullDim Z < n) : ∃ U : Fin n → Opens T, (∀ i, P (U i)) ∧ ∀ x ∈ Z, ∃ i, x ∈ U i

theorem **QCoh.subsingleton_sections_of_notMem_support** {W : X.Opens} (h : ∀ x ∈ W, x ∉ support F) : Subsingleton Γ(F, W)

theorem **cohOn.subsingleton_of_notMem_support** {W : X.Opens} (h : ∀ x ∈ W, x ∉ support F) (n : ℕ) : Subsingleton (cohOn F W n)

theorem **coh.subsingleton_of_support_subset** {ι : Type} [Fintype ι] {U : ι → X.Opens} (hU : ∀ s : Finset ι, s.Nonempty → IsAffineOpen (⊓ i ∈ s, U i)) [F.IsQuasicoherent] {Z : Set X} (hZ : IsClosed Z) (hF : support F ⊆ Z) (hc : ∀ x ∈ Z, ∃ i, x ∈ U i) {n : ℕ} (hn : Fintype.card ι ≤ n) : Subsingleton (coh F n)

theorem **IsAmpleLB.exists_cover_of_dim_lt** [NoetherianSpace X] (h : IsAmpleLB L) {Z : Set X} (hZ : IsClosed Z) {n : ℕ} (hn : topologicalKrullDim Z < n) : ∃ N, 1 ≤ N ∧ ∃ s : Fin n → (lbPow L N).Sec, (∀ i, IsAffineOpen (nonzeroLocus (s i))) ∧ ∀ x ∈ Z, ∃ i, x ∈ nonzeroLocus (s i)

theorem **isAffineOpen_finset_biInf_nonzeroLocus** {ι : Type*} (hL : L.IsLB) {s : ι → L.Sec} (ha : ∀ i, IsAffineOpen (nonzeroLocus (s i))) (t : Finset ι) (ht : t.Nonempty) : IsAffineOpen (⊓ i ∈ t, nonzeroLocus (s i))

theorem **IsAmpleLB.subsingleton_coh_of_support_subset** [NoetherianSpace X] (h : IsAmpleLB L) (F : X.Modules) [F.IsQuasicoherent] {Z : Set X} (hZ : IsClosed Z) (hF : support F ⊆ Z) {n : ℕ} (hn : topologicalKrullDim Z < n) : Subsingleton (coh F n)

theorem **IsAmpleLB.noetherianSpace** [IsLocallyNoetherian X] (h : IsAmpleLB L) : NoetherianSpace X

theorem **IsAmpleLB.subsingleton_coh_of_dim_lt** [IsLocallyNoetherian X] (h : IsAmpleLB L) (F : X.Modules) [F.IsQuasicoherent] {n : ℕ} (hn : topologicalKrullDim X < n) : Subsingleton (coh F n)

theorem **IsAmpleLB.subsingleton_coh_of_suppDim_lt** [IsLocallyNoetherian X] (h : IsAmpleLB L) (F : X.Modules) [IsFiniteQC F] {n : ℕ} (hn : suppDim F < n) : Subsingleton (coh F n)

theorem **isNoetherian_of_isProjOver** {R : Type} [CommRing R] [IsNoetherianRing R] {f : X ⟶ Sp R} (hf : IsProjOver f) : AlgebraicGeometry.IsNoetherian X

theorem **subsingleton_coh_of_isProjOver** {R : Type} [CommRing R] [IsNoetherianRing R] {f : X ⟶ Sp R} (hf : IsProjOver f) (F : X.Modules) [IsFiniteQC F] {n : ℕ} (hn : suppDim F < n) : Subsingleton (coh F n)

theorem **map_annihilator_of_isLocalizedModule** {A B M N : Type*} [CommRing A] [CommRing B] [Algebra A B] (T : Submonoid A) [IsLocalization T B] [AddCommGroup M] [Module A M] [Module.Finite A M] [AddCommGroup N] [Module A N] [Module B N] [IsScalarTower A B N] (f : M →ₗ[A] N) [IsLocalizedModule T f] : (Module.annihilator A M).map (algebraMap A B) = Module.annihilator B N

theorem **isFiniteLength_of_isArtinianRing_annihilator** {A M : Type*} [CommRing A] [IsNoetherianRing A] [AddCommGroup M] [Module A M] [Module.Finite A M] [IsArtinianRing (A ⧸ Module.annihilator A M)] : IsFiniteLength A M

theorem **length_eq_mul_length_residueField** (R : Type*) {S M : Type*} [CommRing R] [CommRing S] [Algebra R S] [IsLocalRing S] [AddCommGroup M] [Module S M] (h : IsFiniteLength S M) [iR : Module R M] [iT : IsScalarTower R S M] : Module.length R M = Module.length S M * Module.length R (IsLocalRing.ResidueField S)

@[simp] theorem **QCoh.annihilatorSheaf_ideal** (U : X.affineOpens) : (annihilatorSheaf F).ideal U = Module.annihilator Γ(X, U) Γ(F, U)

theorem **QCoh.coe_support_annihilatorSheaf** : ((annihilatorSheaf F).support : Set X) = support F

theorem **QCoh.range_annihilatorSheaf_subschemeι** : Set.range (annihilatorSheaf F).subschemeι = support F

theorem **QCoh.topologicalKrullDim_annihilatorSheaf_subscheme** : topologicalKrullDim (annihilatorSheaf F).subscheme = suppDim F

theorem **QCoh.germ_support_bijective** (hf : (support F).Finite) (hc : ∀ x ∈ support F, IsClosed ({x} : Set X)) : Function.Bijective fun (s : Γ(F, ⊤)) (x : support F) => (F.presheaf.germ ⊤ x.1 (Opens.mem_top _) s : (F.presheaf.stalk x.1 :))

theorem **QCoh.isLocallyArtinian_annihilatorSheaf** (h : suppDim F ≤ 0) : IsLocallyArtinian (annihilatorSheaf F).subscheme

theorem **QCoh.isClosed_singleton_of_suppDim_le_zero** (h : suppDim F ≤ 0) {x : X} (hx : x ∈ support F) : IsClosed ({x} : Set X)

theorem **QCoh.isArtinianScheme_annihilatorSheaf** [CompactSpace X] (h : suppDim F ≤ 0) : IsArtinianScheme (annihilatorSheaf F).subscheme

theorem **QCoh.finite_support_of_suppDim_le_zero** [CompactSpace X] (h : suppDim F ≤ 0) : (support F).Finite

theorem **QCoh.isFiniteLength_sections** (h : suppDim F ≤ 0) {U : X.Opens} (hU : IsAffineOpen U) : IsFiniteLength Γ(X, U) Γ(F, U)

theorem **QCoh.isFiniteLength_stalk** (h : suppDim F ≤ 0) (x : X) : IsFiniteLength (X.presheaf.stalk x) (F.presheaf.stalk x :)

theorem **length_coh_zero_eq_sum** [CompactSpace X] (h : suppDim F ≤ 0) : Module.length R (coh F 0) = ∑ x ∈ (finite_support_of_suppDim_le_zero h).toFinset, Module.length (X.presheaf.stalk x) (F.presheaf.stalk x :) * (letI

theorem **subsingleton_coh_of_suppDim_le_zero** (h : suppDim F ≤ 0) {n : ℕ} (hn : 0 < n) : Subsingleton (coh F n)

theorem **exists_isGen_of_finite** {Z : Scheme.{0}} [Finite Z] [IsAffine Z] {M : LB Z} (hM : M.IsLB) : ∃ t : M.Gen, t.IsGen

theorem **nonempty_twist_iso_of_suppDim_le_zero** (h : suppDim F ≤ 0) {M : LB X} (hM : M.IsLB) : Nonempty (twist F M ≅ F)

theorem **residueDegree_eq_finrank** (x : X) : f.residueDegree x = letI

theorem **isNoetherian_of_locallyOfFiniteType_field** : AlgebraicGeometry.IsNoetherian X

theorem **isFinite_annihilatorSheaf** (h : suppDim F ≤ 0) : IsFinite ((annihilatorSheaf F).subschemeι ≫ f)

theorem **finiteCohTwists_of_suppDim_le_zero** (h : suppDim F ≤ 0) : letI

theorem **finite_residueField_of_suppDim_le_zero** (h : suppDim F ≤ 0) {x : X} (hx : x ∈ support F) : letI

theorem **hdim_zero_eq_sum** (h : suppDim F ≤ 0) {S : Finset X} (hS : support F ⊆ S) : letI

theorem **hdim_zero_eq_sum_toNat** (h : suppDim F ≤ 0) {S : Finset X} (hS : support F ⊆ S) : letI

theorem **hdim_twist_of_suppDim_le_zero** {M : LB X} (hM : M.IsLB) (n : ℕ) : hdim k (twist F M) n = hdim k F n

theorem **chi_twist_of_suppDim_le_zero** {M : LB X} (hM : M.IsLB) : chi k (twist F M) = chi k F

theorem **chi_eq_hdim_zero_of_suppDim_le_zero** : chi k F = hdim k F 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsFiniteMulByNatAbelianVarietyAlgClosed

theorem **pw_one** {X : Scheme.{0}} (H : LB X) : pw H ((1 : ℕ) : ℤ) = H

theorem **relAmple_of_veryAmpleBy'** {R : Type} [CommRing R] {A : Over (Spec (CommRingCat.of R))} {L : LB A.left} (hL : L.IsLB) {M : ℕ} (s : Fin (M + 1) → LB.Sec L) (hA : VeryAmpleBy A.hom L s) : RelAmple A L

theorem **mul_eq_one_of_covolume_identity** {G : Type} [CommGroup G] (a b c : G) : a * b ^ (-1 : ℤ) * c⁻¹ = (b * a⁻¹ * c)⁻¹

theorem **exists_gen** {X : Scheme.{0}} (L : LB X) (m : X ⟶ X) (e : ℤ) (w : LB.Iso (pw L e) (L.pull m)) (hw : w.IsGen) : ∃ α : LB.Gen (LB.LinComb L (K

theorem **isFinite_mulA** : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Spec (CommRingCat.of K))) [GrpObj B], IsAbelianScheme B → ∀ n : ℕ, 1 ≤ n → IsFinite (mulA B n).left

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsFiniteMulByNatAbelianVarietyAlgClosedStatement

theorem **_root_.GlobalTheory.T.isFinite_mulByNat_abelianVariety_algClosed** : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Spec (CommRingCat.of K))) [GrpObj B], IsAbelianScheme B → ∀ n : ℕ, 1 ≤ n → IsFinite (mulA B n).left

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsGloballyGeneratedIffGeneratingSections

theorem **exists_finite_subset_le_of_basis** {T : Type*} [TopologicalSpace T] [T0Space T] (P : Opens T → Prop) (h0 : P ⊥) (hb : ∀ (x : T) (U : Opens T), x ∈ U → ∃ A, P A ∧ x ∈ A ∧ A ≤ U) (hadd : ∀ A B, P A → P B → ∃ C, P C ∧ C ≤ A ⊔ B ∧ (∀ x ∈ A, x ∈ C ∨ x ∈ B) ∧ ∀ x ∈ B, x ∈ C ∨ x ∈ A) {S : Set T} (hS : S.Finite) {U : Opens T} (hU : S ⊆ U) : ∃ A, P A ∧ S ⊆ A ∧ A ≤ U

theorem **nonzeroLocus_add_le** (s t : M.Sec) : nonzeroLocus (s + t) ≤ nonzeroLocus s ⊔ nonzeroLocus t

theorem **nonzeroLocus_neg** (s : M.Sec) : nonzeroLocus (-s) = nonzeroLocus s

theorem **nonzeroLocus_le_add_sup** (s t : M.Sec) : nonzeroLocus s ≤ nonzeroLocus (s + t) ⊔ nonzeroLocus t

theorem **isAffineOpen_nonzeroLocus_add** {s t : M.Sec} (hs : IsAffineOpen (nonzeroLocus s)) (ht : IsAffineOpen (nonzeroLocus t)) : IsAffineOpen (nonzeroLocus (s + t))

theorem **IsAmpleLB.exists_affine_finite_subset** (h : IsAmpleLB L) {S : Set X} (hS : S.Finite) {U : X.Opens} (hU : S ⊆ U) : ∃ n, 1 ≤ n ∧ ∃ s : (lbPow L n).Sec, S ⊆ nonzeroLocus s ∧ nonzeroLocus s ≤ U ∧ IsAffineOpen (nonzeroLocus s)

theorem **exists_sec_finite_subset_of_isQuasiAffine** [X.IsQuasiAffine] (hM : M.IsLB) {S : Set X} (hS : S.Finite) {U : X.Opens} (hU : S ⊆ U) : ∃ s : M.Sec, S ⊆ nonzeroLocus s ∧ nonzeroLocus s ≤ U ∧ IsAffineOpen (nonzeroLocus s)

theorem **exists_isGen_of_nonzeroLocus_eq_top** (s : M.Sec) (h : nonzeroLocus s = ⊤) : ∃ t : M.Gen, t.IsGen

theorem **exists_affine_finite_subset_trivial** (hM : M.IsLB) {S : Set X} (hS : S.Finite) {W : X.Opens} (hW : IsAffineOpen W) (hSW : S ⊆ W) : ∃ V : X.Opens, IsAffineOpen V ∧ S ⊆ V ∧ V ≤ W ∧ ∃ t : (M.pull V.ι).Gen, t.IsGen

theorem **IsAmpleLB.exists_affine_finite_subset_trivial** (h : IsAmpleLB L) (hM : M.IsLB) {S : Set X} (hS : S.Finite) {U : X.Opens} (hU : S ⊆ U) : ∃ V : X.Opens, IsAffineOpen V ∧ S ⊆ V ∧ V ≤ U ∧ ∃ t : (M.pull V.ι).Gen, t.IsGen

theorem **sectionHom_app** (s : Γ(F, ⊤)) (V : X.Opens) (a : Γ(structureModule X, V)) : (sectionHom s).app V a = structureModuleSectionsEquiv V a • resM F le_top s

theorem **app_eq_smul** (φ : structureModule X ⟶ F) (V : X.Opens) (a : Γ(structureModule X, V)) : φ.app V a = structureModuleSectionsEquiv V a • resM F le_top (φ.app ⊤ ((structureModuleSectionsEquiv ⊤).symm 1))

@[simp] theorem **sectionsHom_app_freeBasis** (s : ι → Γ(F, ⊤)) (i : ι) : (sectionsHom s).app ⊤ (freeBasis X i) = s i

theorem **exists_sum_smul_freeBasis** [Fintype ι] (V : X.Opens) (t : Γ(freeModule X ι, V)) : ∃ a : ι → Γ(X, V), ∑ i, a i • resM _ le_top (freeBasis X i) = t

theorem **range_appLinear_free** [Finite ι] (φ : freeModule X ι ⟶ F) (V : X.Opens) : LinearMap.range (appLinear φ V) = Submodule.span Γ(X, V) (Set.range fun i => resM F le_top (φ.app ⊤ (freeBasis X i)))

theorem **germ_mem_span** (s : ι → Γ(F, ⊤)) {V : X.Opens} {x : X} (hx : x ∈ V) {t : Γ(F, V)} (ht : t ∈ Submodule.span Γ(X, V) (Set.range fun i => resM F le_top (s i))) : F.presheaf.germ V x hx t ∈ Submodule.span (X.presheaf.stalk x) (Set.range fun i => (F.presheaf.germ ⊤ x trivial (s i) : (F.presheaf.stalk x :)))

theorem **epi_iff_spanStalks** [Finite ι] (φ : freeModule X ι ⟶ F) : Epi φ ↔ SpanStalks fun i => φ.app ⊤ (freeBasis X i)

theorem **epi_sectionsHom_iff** [Finite ι] (s : ι → Γ(F, ⊤)) : Epi (sectionsHom s) ↔ SpanStalks s

theorem **SpanStalks.isGloballyGenerated** [Finite ι] {s : ι → Γ(F, ⊤)} (h : SpanStalks s) : IsGloballyGenerated F

theorem **isGloballyGenerated_iff_spanStalks** : IsGloballyGenerated F ↔ ∃ (n : ℕ) (s : Fin n → Γ(F, ⊤)), SpanStalks s

theorem **IsGloballyGenerated.of_epi** (h : IsGloballyGenerated F) (φ : F ⟶ G) [Epi φ] : IsGloballyGenerated G

theorem **isGloballyGenerated_iff_of_iso** (e : F ≅ G) : IsGloballyGenerated F ↔ IsGloballyGenerated G

theorem **IsGloballyGenerated.isFiniteQC** (h : IsGloballyGenerated F) [F.IsQuasicoherent] : IsFiniteQC F

theorem **spanStalks_of_forall_exists_affine** [F.IsQuasicoherent] (s : ι → Γ(F, ⊤)) (h : ∀ x : X, ∃ V : X.Opens, IsAffineOpen V ∧ x ∈ V ∧ Submodule.span Γ(X, V) (Set.range fun i => resM F (le_top : V ≤ ⊤) (s i)) = ⊤) : SpanStalks s

theorem **QCoh.IsGloballyGenerated.twist** (h : IsGloballyGenerated F) (hM : IsGenerated M) : IsGloballyGenerated (twist F M)

theorem **Twist.eventually_exists_span_resM_eq_top** [CompactSpace X] [QuasiSeparatedSpace X] [IsFiniteQC F] (hL : L.IsLB) {d : ℤ} (σ : LB.Sec (lbZPow L d)) (hσ : IsAffineOpen (nonzeroLocus σ)) (r : ℤ) : ∀ᶠ k : ℕ in atTop, ∀ n : ℤ, r + k * d = n → ∃ (m : ℕ) (a : Fin m → Γ(twistZ F L n, ⊤)), Submodule.span Γ(X, nonzeroLocus σ) (Set.range fun j => resM _ (le_top : nonzeroLocus σ ≤ ⊤) (a j)) = ⊤

theorem **IsAmpleLB.eventually_isGloballyGenerated_twistZ** (h : IsAmpleLB L) (F : X.Modules) [IsFiniteQC F] : ∀ᶠ n : ℤ in atTop, IsGloballyGenerated (twistZ F L n)

theorem **IsAmpleLB.eventually_isGloballyGenerated_twist** (h : IsAmpleLB L) (F : X.Modules) [IsFiniteQC F] : ∀ᶠ n : ℕ in atTop, IsGloballyGenerated (twist F (lbPow L n))

theorem **QCoh.IsGloballyGenerated.exists_epi_sigma** (hL : L.IsLB) {n : ℤ} (h : IsGloballyGenerated (twistZ F L n)) : ∃ (k : ℕ) (φ : (⨿ fun _ : Fin k => toMod (lbZPow L (-n))) ⟶ F), Epi φ

theorem **IsAmpleLB.eventually_exists_epi_sigma** (h : IsAmpleLB L) (F : X.Modules) [IsFiniteQC F] : ∀ᶠ n : ℤ in atTop, ∃ (k : ℕ) (φ : (⨿ fun _ : Fin k => toMod (lbZPow L (-n))) ⟶ F), Epi φ

theorem **Twist.secOfTwist_f** (j : J ⟶ structureModule X) (a : Γ(twist J M, ⊤)) (i : M.ι) : (secOfTwist j a).f i = resG X (le_inf le_top le_rfl) (structureModuleSectionsEquiv _ (j.app _ (coord a i)))

theorem **Twist.exists_mem_nonzeroLocus_secOfTwist** (hM : M.IsLB) (j : J ⟶ structureModule X) {ι : Type} {a : ι → Γ(twist J M, ⊤)} (ha : SpanStalks a) {U : X.Opens} {x : X} (hx : x ∈ U) (t : Γ(J, U)) (ht : structureModuleSectionsEquiv U (j.app U t) = 1) : ∃ i, x ∈ nonzeroLocus (secOfTwist j (a i))

theorem **isGloballyGenerated_toMod_iff** (hM : M.IsLB) : IsGloballyGenerated (toMod M) ↔ IsGenerated M

theorem **QCoh.exists_idealModuleι_app_eq_iff** (I : X.IdealSheafData) (V : X.Opens) (f : Γ(X, V)) : (∃ t, structureModuleSectionsEquiv V ((idealModuleι I).app V t) = f) ↔ I.subschemeι.app V f = 0

theorem **QCoh.exists_idealModuleι_app_eq_iff_mem** (I : X.IdealSheafData) (V : X.affineOpens) (f : Γ(X, V)) : (∃ t, structureModuleSectionsEquiv V.1 ((idealModuleι I).app V t) = f) ↔ f ∈ I.ideal V

theorem **QCoh.notMem_support_of_mem_basicOpen** (I : X.IdealSheafData) {V : X.Opens} (t : Γ(idealModule I, V)) {x : X} (hx : x ∈ X.basicOpen (structureModuleSectionsEquiv V ((idealModuleι I).app V t))) : x ∉ I.support

theorem **QCoh.exists_idealModuleι_app_eq_one** (I : X.IdealSheafData) {V : X.Opens} (hV : ∀ x ∈ V, x ∉ I.support) : ∃ t, structureModuleSectionsEquiv V ((idealModuleι I).app V t) = 1

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsHomogeneousRangeMulVecLin

theorem **isGradedMap_mulVecLin** {ι : Type} {R : Type*} [CommRing R] {κ μ : Type*} [Fintype μ] {d : κ → ℤ} {e : μ → ℤ} {Φ : Matrix κ μ (MvPolynomial ι R)} (h : ∀ l j, Φ l j ∈ polyGrading ι R (e j - d l)) : IsGradedMap (grFree ι R e) (grFree ι R d) Φ.mulVecLin

theorem **isHomogeneous_range_mulVecLin** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (a b : ℕ) (d : Fin a → ℤ) (e : Fin b → ℤ) (Φ : Matrix (Fin a) (Fin b) (MvPolynomial ι R)), (∀ l j, Φ l j ∈ polyGrading ι R (e j - d l)) → (LinearMap.range Φ.mulVecLin).IsHomogeneous (grFree ι R d)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsIsoGrCechMapPolyToGamma

theorem **isGradedMap_polyToGamma** : IsGradedMap (polyGrading ι R) (gammaGrading R x _ ⊤) (polyToGamma R x hL)

theorem **gammaOf_formSection** (p : polyGrading ι R n) : gammaOf R x _ ⊤ n (formSection R x hL n p) = polyToGamma R x hL p

theorem **gammaCoord_polyToGamma** (i : L.ι) {V : X.Opens} (p : MvPolynomial ι R) : (sectionsOn.toΓ _ _ (gammaCoord R x _ V i (gammaRes R x _ le_top (polyToGamma R x hL p))) : Γ(X, V ⊓ L.U i)) = chartEval R x i inf_le_right p

theorem **zcoord_formSection** (p : polyGrading ι R n) (i : L.ι) : (zcoord (sectionsOn.toΓ _ ⊤ (formSection R x hL n p)) i : Γ(X, ⊤ ⊓ L.U i)) = chartEval R x i inf_le_right p.1

theorem **globalEquiv_formSection** (p : polyGrading ι R n) : LB.globalEquiv _ ((unitTwistIso (lbZPow L n)).hom.app ⊤ (sectionsOn.toΓ _ ⊤ (formSection R x hL n p))) = Twist.formSec x p.2

theorem **eq_zero_of_gammaCoord** {F : X.Modules} {V : X.Opens} {i : L.ι} (h : V ≤ L.U i) {y : GammaStar R x F V} (hy : y ∈ gammaGrading R x F V n) (h0 : sectionsOn.toΓ _ _ (gammaCoord R x F V i y) = 0) : y = 0

theorem **gammaRes_smul_sub_eq_zero** {V : X.Opens} {i : L.ι} (h : V ≤ L.U i) {p q : MvPolynomial ι R} {y : GammaStar R x (structureModule X) ⊤} (h₁ : q • y ∈ gammaGrading R x _ ⊤ n) (h₂ : p ∈ polyGrading ι R n) (u : Γ(X, V ⊓ L.U i)) (hu : u = sectionsOn.toΓ _ _ (gammaCoord R x _ V i (gammaRes R x _ le_top y))) (e : chartEval R x i inf_le_right q * u = chartEval R x i inf_le_right p) : gammaRes R x _ (le_top : V ≤ ⊤) (q • y - polyToGamma R x hL p) = 0

theorem **formSec_one** (h : (1 : MvPolynomial ι R) ∈ polyGrading ι R 0) : Twist.formSec x h = zone L

theorem **formSec_mul** {p q : MvPolynomial ι R} (hp : p ∈ polyGrading ι R n) (hq : q ∈ polyGrading ι R m) (h : p * q ∈ polyGrading ι R (n + m)) : Twist.formSec x h = zmul (Twist.formSec x hp) (Twist.formSec x hq)

theorem **formSec_smul** (r : R) {p : MvPolynomial ι R} (hp : p ∈ polyGrading ι R n) (h : r • p ∈ polyGrading ι R n) : Twist.formSec x h = algebraMap R Γ(X, ⊤) r • Twist.formSec x hp

theorem **polyToSecRing_of_mem** {p : MvPolynomial ι R} (hp : p ∈ polyGrading ι R n) : polyToSecRing R x p = SecRing.of n (Twist.formSec x hp)

theorem **polyToSecRing_bijective** (h : ∀ n, Function.Bijective (formSecAdd R x n)) : Function.Bijective (polyToSecRing R x)

theorem **isCompact_top** : IsCompact ((⊤ : (PSpace R N).Opens) : Set (PSpace R N))

theorem **isQuasiSeparated_top** : IsQuasiSeparated ((⊤ : (PSpace R N).Opens) : Set (PSpace R N))

theorem **isAffineOpen_biInf_coord** (s : Finset (Fin (N + 1))) (hs : s.Nonempty) : IsAffineOpen (⊓ i ∈ s, Out.nonzeroLocus (ProjSpace.coord R N i))

theorem **chartEval_coord** (i : Fin (N + 1)) {U : (PSpace R N).Opens} (h : U ≤ chart R N i) (p : MvPolynomial (Fin (N + 1)) R) : chartEval (L

theorem **formSection_injective** (n : ℤ) : Function.Injective (formSection R (ProjSpace.coord R N) (O1_isLB R N) n)

theorem **polyToGamma_injective** : Function.Injective (polyToGamma R (ProjSpace.coord R N) (O1_isLB R N))

theorem **exists_pow_smul_mem_range** (i : Fin (N + 1)) (y : GammaStar R (ProjSpace.coord R N) (structureModule _) ⊤) : ∃ k : ℕ, (X i : MvPolynomial (Fin (N + 1)) R) ^ k • y ∈ LinearMap.range (polyToGamma R (ProjSpace.coord R N) (O1_isLB R N))

theorem **formSection_bijective** (hN : 1 ≤ N) (n : ℤ) : Function.Bijective (formSection R (ProjSpace.coord R N) (O1_isLB R N) n)

theorem **formSecAdd_bijective** (hN : 1 ≤ N) (n : ℤ) : Function.Bijective (formSecAdd R (ProjSpace.coord R N) n)

theorem **formSec_eq** {n : ℕ} {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : ProjSpace.formSec n f hf = Twist.formSec (ProjSpace.coord R N) (mem_polyGrading_natCast.2 hf)

theorem **subsingleton_coh** (n : ℤ) {p : ℕ} (h0 : p ≠ 0) (hp : p ≠ N) : Subsingleton (coh (sheafO R N n) p)

theorem **nonempty_basis_coh_top** (hN : 1 ≤ N) (n : ℤ) : Nonempty (Module.Basis (NegExp N n) R (coh (sheafO R N n) N))

theorem **subsingleton_coh_top** (hN : 1 ≤ N) {n : ℤ} (hn : -(N : ℤ) ≤ n) : Subsingleton (coh (sheafO R N n) N)

theorem **subsingleton_coh_of_neg_le** {n : ℤ} (hn : -(N : ℤ) ≤ n) {p : ℕ} (hp : 0 < p) : Subsingleton (coh (sheafO R N n) p)

theorem **subsingleton_coh_zero** (hN : 1 ≤ N) {n : ℤ} (hn : n < 0) : Subsingleton (coh (sheafO R N n) 0)

theorem **nonempty_coh_zero_equiv** (n : ℤ) : Nonempty (coh (sheafO R 0 n) 0 ≃ₗ[R] R)

theorem **finite_free_coh** (n : ℤ) (p : ℕ) : Module.Finite R (coh (sheafO R N n) p) ∧ Module.Free R (coh (sheafO R N n) p)

theorem **finrank_coh_zero** [Nontrivial R] (hN : 1 ≤ N) (n : ℤ) : Module.finrank R (coh (sheafO R N n) 0) = (n + N).toNat.choose N

theorem **finrank_coh_top** [Nontrivial R] (hN : 1 ≤ N) (n : ℤ) : Module.finrank R (coh (sheafO R N n) N) = (-n - 1).toNat.choose N

theorem **chi_sheafO** (n : ℤ) : chi k (sheafO k N n) = Ring.choose (n + N) N

theorem **chi_sheafO_eq_eval** (n : ℤ) : (chi k (sheafO k N n) : ℚ) = (Polynomial.preHilbertPoly ℚ N 0).eval (n : ℚ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsIsoToInterAppPushforwardInclusion

@[simp] lemma **interFunctor_obj** (W B : Opens X) : (interFunctor W).obj B = B ⊓ W

@[simp] lemma **interSheaf_obj_obj_obj** (W : Opens X) (F : X.Sheaf A) (B : Opens X) : ((interSheaf A W).obj F).obj.obj (op B) = F.obj.obj (op (B ⊓ W))

@[simp] lemma **interSheaf_obj_obj_map** (W : Opens X) (F : X.Sheaf A) {B B' : Opens X} (i : B' ⟶ B) : ((interSheaf A W).obj F).obj.map i.op = F.obj.map (homOfLE (inf_le_inf_right W i.le)).op

@[simp] lemma **interSheaf_map_hom_app** (W : Opens X) {F G : X.Sheaf A} (f : F ⟶ G) (B : Opens X) : ((interSheaf A W).map f).hom.app (op B) = f.hom.app (op (B ⊓ W))

@[simp] lemma **toInter_app_hom_app** (W : Opens X) (F : X.Sheaf A) (B : Opens X) : ((toInter A W).app F).hom.app (op B) = F.obj.map (homOfLE inf_le_left).op

lemma **isIso_toInter_app_of_isBasis** {𝔅 : Set (Opens X)} (h𝔅 : Opens.IsBasis 𝔅) (W : Opens X) (G : X.Sheaf A) (h : ∀ B ∈ 𝔅, IsIso (G.obj.map (homOfLE (inf_le_left : B ⊓ W ≤ B)).op)) : IsIso ((toInter A W).app G)

lemma **isIso_toInter_app_pushforward** {Y : TopCat.{u}} (j : Y ⟶ X) (W : Opens X) (hj : ∀ y, j y ∈ W) (G : Y.Sheaf A) : IsIso ((toInter A W).app ((TopCat.Sheaf.pushforward A j).obj G))

lemma **toInter_app_comp_map_comp_inv** (W : Opens X) {F G : X.Sheaf A} (f : F ⟶ G) [IsIso ((toInter A W).app G)] : (toInter A W).app F ≫ (interSheaf A W).map f ≫ inv ((toInter A W).app G) = f

lemma **exact_map_interSheaf** (hS : S.ShortExact) (W : Opens X) : (S.map (interSheaf AddCommGrpCat.{u} W)).Exact ∧ Mono (S.map (interSheaf AddCommGrpCat.{u} W)).f

lemma **shortExact_map_interSheaf** (hS : S.ShortExact) {𝔅 : Set (Opens X)} (h𝔅 : Opens.IsBasis 𝔅) (hinf : ∀ B ∈ 𝔅, ∀ B' ∈ 𝔅, B ⊓ B' ∈ 𝔅) (hsurj : ∀ B ∈ 𝔅, Function.Surjective (S.g.hom.app (op B))) {W : Opens X} (hW : W ∈ 𝔅) : (S.map (interSheaf AddCommGrpCat.{u} W)).ShortExact

lemma **exists_isBasis_lift** (hS : S.ShortExact) {𝔅 : Set (Opens X)} (h𝔅 : Opens.IsBasis 𝔅) {V : Opens X} (s : S.X₃.obj.obj (op V)) {x : X} (hx : x ∈ V) : ∃ W ∈ 𝔅, x ∈ W ∧ W ≤ V ∧ ∃ t : S.X₂.obj.obj (op (V ⊓ W)), S.g.hom.app (op (V ⊓ W)) t = S.X₃.obj.map (homOfLE inf_le_left).op s

lemma **map_toInter_δ_eq_zero** (hS : S.ShortExact) {V W : Opens X} (s : S.X₃.obj.obj (op V)) (t : S.X₂.obj.obj (op (V ⊓ W))) (ht : S.g.hom.app (op (V ⊓ W)) t = S.X₃.obj.map (homOfLE inf_le_left).op s) : map.{w} ((toInter AddCommGrpCat.{u} W).app S.X₁) 1 V (δ hS 0 V ((equivZero S.X₃ V).symm s)) = 0

theorem **exists_mem_map_toInter_eq_zero** {𝔅 : Set (Opens X)} (h𝔅 : Opens.IsBasis 𝔅) (hinf : ∀ B ∈ 𝔅, ∀ B' ∈ 𝔅, B ⊓ B' ∈ 𝔅) (V : Opens X) (n : ℕ) (F : X.Sheaf AddCommGrpCat.{u}) (hF : ∀ q, 0 < q → q < n + 1 → ∀ B ∈ 𝔅, Subsingleton (Sheaf.H'.{w} F q B)) (α : Sheaf.H'.{w} F (n + 1) V) {x : X} (hx : x ∈ V) : ∃ W ∈ 𝔅, x ∈ W ∧ W ≤ V ∧ map.{w} ((toInter AddCommGrpCat.{u} W).app F) (n + 1) V α = 0

lemma **map_eq_zero_of_map_toInter_eq_zero** {W : Opens X} {F G : X.Sheaf AddCommGrpCat.{u}} (f : F ⟶ G) [IsIso ((toInter AddCommGrpCat.{u} W).app G)] {n : ℕ} {V : Opens X} {α : Sheaf.H'.{w} F n V} (h : map.{w} ((toInter AddCommGrpCat.{u} W).app F) n V α = 0) : map.{w} f n V α = 0

theorem **exists_cover_map_toInter_eq_zero** {𝔅 : Set (Opens X)} (h𝔅 : Opens.IsBasis 𝔅) (hinf : ∀ B ∈ 𝔅, ∀ B' ∈ 𝔅, B ⊓ B' ∈ 𝔅) (F : X.Sheaf AddCommGrpCat.{u}) (V : Opens X) {n : ℕ} (hn : 1 ≤ n) (hF : ∀ q, 0 < q → q < n → ∀ B ∈ 𝔅, Subsingleton (Sheaf.H'.{w} F q B)) (α : Sheaf.H'.{w} F n V) : ∃ 𝔘 ⊆ 𝔅, sSup 𝔘 = V ∧ ∀ W ∈ 𝔘, map.{w} ((toInter AddCommGrpCat.{u} W).app F) n V α = 0

theorem **exists_finite_cover_map_toInter_eq_zero** {𝔅 : Set (Opens X)} (h𝔅 : Opens.IsBasis 𝔅) (hinf : ∀ B ∈ 𝔅, ∀ B' ∈ 𝔅, B ⊓ B' ∈ 𝔅) (F : X.Sheaf AddCommGrpCat.{u}) (V : Opens X) (hV : IsCompact (V : Set X)) {n : ℕ} (hn : 1 ≤ n) (hF : ∀ q, 0 < q → q < n → ∀ B ∈ 𝔅, Subsingleton (Sheaf.H'.{w} F q B)) (α : Sheaf.H'.{w} F n V) : ∃ 𝔘 ⊆ 𝔅, 𝔘.Finite ∧ sSup 𝔘 = V ∧ ∀ W ∈ 𝔘, map.{w} ((toInter AddCommGrpCat.{u} W).app F) n V α = 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsKStableOnKFiniteVectors

theorem **mem_kFiniteVectors** (ε : Fin n → Bool) (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)) (v : H) : v ∈ kFiniteVectors ε σ ↔ IsKFin ε σ v

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsQuasicoherentOfForallExistsAffine

theorem **resModule_smul** (h : V ≤ U) (r : Γ(X, U)) (s : Γ(F, V)) : letI

theorem **res_isScalarTower** (h : V ≤ U) : letI

theorem **resLinear_apply** (h : V ≤ U) (s : Γ(F, U)) : resLinear F h s = F.presheaf.map (homOfLE h).op s

theorem **resBasicOpen_apply** (f : Γ(X, U)) (s : Γ(F, U)) : resBasicOpen F f s = F.presheaf.map (homOfLE (X.basicOpen_le f)).op s

theorem **res_res** (h : V ≤ U) (h' : W ≤ V) (s : Γ(F, U)) : F.presheaf.map (homOfLE h').op (F.presheaf.map (homOfLE h).op s) = F.presheaf.map (homOfLE (h'.trans h)).op s

theorem **res_self** (h : U ≤ U) (s : Γ(F, U)) : F.presheaf.map (homOfLE h).op s = s

theorem **res_res_ring** (h : V ≤ U) (h' : W ≤ V) (s : Γ(X, U)) : X.presheaf.map (homOfLE h').op (X.presheaf.map (homOfLE h).op s) = X.presheaf.map (homOfLE (h'.trans h)).op s

theorem **res_self_ring** (h : U ≤ U) (s : Γ(X, U)) : X.presheaf.map (homOfLE h).op s = s

theorem **isUnit_res_of_le_basicOpen** (f : Γ(X, U)) (h : V ≤ U) (hV : V ≤ X.basicOpen f) : IsUnit (X.presheaf.map (homOfLE h).op f)

theorem **isLocalizedModule_resBasicOpen_iff** (f : Γ(X, U)) : IsLocalizedModule.Away f (resBasicOpen F f) ↔ (∀ s : Γ(F, X.basicOpen f), ∃ (n : ℕ) (t : Γ(F, U)), X.presheaf.map (homOfLE (X.basicOpen_le f)).op f ^ n • s = F.presheaf.map (homOfLE (X.basicOpen_le f)).op t) ∧ (∀ t : Γ(F, U), F.presheaf.map (homOfLE (X.basicOpen_le f)).op t = 0 → ∃ n : ℕ, f ^ n • t = 0)

theorem **fromSpec_image_le** (W : (Spec Γ(X, U)).Opens) : hU.fromSpec ''ᵁ W ≤ U

theorem **fromSpec_appIso_inv_res** (W : (Spec Γ(X, U)).Opens) (r : Γ(X, U)) : (hU.fromSpec.appIso W).inv ((Spec Γ(X, U)).presheaf.map W.leTop.op ((Scheme.ΓSpecIso Γ(X, U)).inv r)) = X.presheaf.map (homOfLE (fromSpec_image_le hU W)).op r

theorem **restrict_fromSpec_smul** (W : (Spec Γ(X, U)).Opens) (r : Γ(X, U)) (s : Γ(F.restrict hU.fromSpec, W)) : (F.restrictAppIso hU.fromSpec W).hom (r • s) = X.presheaf.map (homOfLE (fromSpec_image_le hU W)).op r • (F.restrictAppIso hU.fromSpec W).hom s

theorem **isLocalizing_restrict_fromSpec_iff** : IsLocalizing (modulesSpecToSheaf.obj (F.restrict hU.fromSpec)) ↔ ∀ f : Γ(X, U), IsLocalizedModule.Away f (resBasicOpen F f)

theorem **isQuasicoherent_restrict_fromSpec_iff** (hU : IsAffineOpen U) : (F.restrict hU.fromSpec).IsQuasicoherent ↔ ∀ f : Γ(X, U), IsLocalizedModule.Away f (resBasicOpen F f)

theorem **isLocalizedModule_resBasicOpen** [F.IsQuasicoherent] (hU : IsAffineOpen U) (f : Γ(X, U)) : IsLocalizedModule.Away f (resBasicOpen F f)

theorem **nonempty_presentation_Spec** {R : CommRingCat.{u}} (M : (Spec R).Modules) [M.IsQuasicoherent] : Nonempty M.Presentation

theorem **nonempty_presentation_over_of_restrict** (P : (F.restrict U.ι).Presentation) : Nonempty (F.over U).Presentation

theorem **nonempty_presentation_restrict_ι** (hU : IsAffineOpen U) [(F.restrict hU.fromSpec).IsQuasicoherent] : Nonempty (F.restrict U.ι).Presentation

theorem **isQuasicoherent_of_affineCover** {ι : Type*} (𝒰 : ι → X.Opens) (hcov : IsOpenCover 𝒰) (haff : ∀ i, IsAffineOpen (𝒰 i)) (h : ∀ i (f : Γ(X, 𝒰 i)), IsLocalizedModule.Away f (resBasicOpen F f)) : F.IsQuasicoherent

theorem **isQuasicoherent_of_forall_exists_affine** (h : ∀ x : X, ∃ U : X.Opens, IsAffineOpen U ∧ x ∈ U ∧ ∀ f : Γ(X, U), IsLocalizedModule.Away f (resBasicOpen F f)) : F.IsQuasicoherent

theorem **isQuasicoherent_iff_forall_affine** : F.IsQuasicoherent ↔ ∀ U : X.Opens, IsAffineOpen U → ∀ f : Γ(X, U), IsLocalizedModule.Away f (resBasicOpen F f)

theorem **isQuasicoherent_iff_affineCover** {ι : Type*} (𝒰 : ι → X.Opens) (hcov : IsOpenCover 𝒰) (haff : ∀ i, IsAffineOpen (𝒰 i)) : F.IsQuasicoherent ↔ ∀ i (f : Γ(X, 𝒰 i)), IsLocalizedModule.Away f (resBasicOpen F f)

theorem **isQuasicoherent_of_restrict** {ι : Type*} {Y : ι → Scheme.{u}} (g : ∀ i, Y i ⟶ X) [∀ i, IsOpenImmersion (g i)] (hcov : ∀ x : X, ∃ i, x ∈ (g i).opensRange) (h : ∀ i, (F.restrict (g i)).IsQuasicoherent) : F.IsQuasicoherent

theorem **exists_pow_smul_eq_res** (hU : IsAffineOpen U) (f : Γ(X, U)) (h : V ≤ U) (hV : V = X.basicOpen f) (s : Γ(F, V)) : ∃ (n : ℕ) (t : Γ(F, U)), X.presheaf.map (homOfLE h).op f ^ n • s = F.presheaf.map (homOfLE h).op t

theorem **exists_pow_smul_eq_zero** (hU : IsAffineOpen U) (f : Γ(X, U)) (h : V ≤ U) (hV : V = X.basicOpen f) (t : Γ(F, U)) (ht : F.presheaf.map (homOfLE h).op t = 0) : ∃ n : ℕ, f ^ n • t = 0

theorem **isLocalizedModule_resLinear** (hU : IsAffineOpen U) (f : Γ(X, U)) (h : V ≤ U) (hV : V = X.basicOpen f) : letI

theorem **isLocalizedModule_of_ringHom** (hU : IsAffineOpen U) {A : Type*} [CommRing A] (φ : A →+* Γ(X, U)) (a : A) (h : V ≤ U) (hV : V = X.basicOpen (φ a)) [Module A Γ(F, U)] [Module A Γ(F, V)] (hsU : ∀ (r : A) (s : Γ(F, U)), r • s = φ r • s) (hsV : ∀ (r : A) (s : Γ(F, V)), r • s = X.presheaf.map (homOfLE h).op (φ r) • s) (l : Γ(F, U) →ₗ[A] Γ(F, V)) (hl : ∀ s, l s = F.presheaf.map (homOfLE h).op s) : IsLocalizedModule.Away a l

theorem **bijective_of_isLocalizedModule_comp** {B M N : Type*} [CommRing B] [AddCommGroup M] [Module B M] [AddCommGroup N] [Module B N] (φ : M →ₗ[B] N) (s : Set B) (hs : Ideal.span s = ⊤) (P : s → Type*) [∀ b, AddCommGroup (P b)] [∀ b, Module B (P b)] (g : ∀ b, N →ₗ[B] P b) [∀ b : s, IsLocalizedModule.Away b.1 (g b)] [∀ b : s, IsLocalizedModule.Away b.1 (g b ∘ₗ φ)] : Function.Bijective φ

theorem **isLocalizedModule_comp_liftBaseChange** {A B M N P : Type*} (C : Type*) [CommRing A] [CommRing B] [CommRing C] [Algebra A B] [Algebra A C] [Algebra B C] [IsScalarTower A B C] [AddCommGroup M] [Module A M] [AddCommGroup N] [Module A N] [Module B N] [IsScalarTower A B N] [AddCommGroup P] [Module A P] [Module B P] [Module C P] [IsScalarTower A B P] [IsScalarTower A C P] [IsScalarTower B C P] (l : M →ₗ[A] N) (g : N →ₗ[B] P) (S : Submonoid B) [IsLocalization S C] (h : IsBaseChange C (g.restrictScalars A ∘ₗ l)) : IsLocalizedModule S (g ∘ₗ l.liftBaseChange B)

theorem **isBaseChange_resLinear** [F.IsQuasicoherent] (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) : letI

theorem **isBaseChange_of_eq_res** [F.IsQuasicoherent] (hU : IsAffineOpen U) (hV : IsAffineOpen V) (h : V ≤ U) [alg : Algebra Γ(X, U) Γ(X, V)] [mod : Module Γ(X, U) Γ(F, V)] [IsScalarTower Γ(X, U) Γ(X, V) Γ(F, V)] (halg : ∀ r, algebraMap Γ(X, U) Γ(X, V) r = X.presheaf.map (homOfLE h).op r) (l : Γ(F, U) →ₗ[Γ(X, U)] Γ(F, V)) (hl : ∀ s, l s = F.presheaf.map (homOfLE h).op s) : IsBaseChange Γ(X, V) l

theorem **isQuasicoherent_iff_isBaseChange** : F.IsQuasicoherent ↔ ∀ (U V : X.Opens), IsAffineOpen U → IsAffineOpen V → ∀ h : V ≤ U, letI

theorem **germ_smul** (x : X) (hx : x ∈ U) (r : Γ(X, U)) (s : Γ(F, U)) : F.presheaf.germ U x hx (r • s) = X.presheaf.germ U x hx r • F.presheaf.germ U x hx s

theorem **stalkModuleSections_smul** (x : U) (r : Γ(X, U)) (t : (F.presheaf.stalk (x : X) :)) : r • t = X.presheaf.germ U x x.2 r • t

theorem **germLinear_apply** (x : U) (s : Γ(F, U)) : germLinear F x s = F.presheaf.germ U x x.2 s

theorem **mem_primeCompl_primeIdealOf_iff** (hU : IsAffineOpen U) (x : U) (f : Γ(X, U)) : f ∈ (hU.primeIdealOf x).asIdeal.primeCompl ↔ (x : X) ∈ X.basicOpen f

theorem **isLocalizedModule_germLinear** [F.IsQuasicoherent] (hU : IsAffineOpen U) (x : U) : IsLocalizedModule (hU.primeIdealOf x).asIdeal.primeCompl (germLinear F x) where

theorem **isBaseChange_germLinear** [F.IsQuasicoherent] (hU : IsAffineOpen U) (x : U) : IsBaseChange (X.presheaf.stalk (x : X)) (germLinear F x)

theorem **isQuasicoherent_tfae** : List.TFAE [F.IsQuasicoherent, ∀ U : X.Opens, IsAffineOpen U → ∀ f : Γ(X, U), IsLocalizedModule.Away f (resBasicOpen F f), ∃ (ι : Type u) (𝒰 : ι → X.Opens), IsOpenCover 𝒰 ∧ (∀ i, IsAffineOpen (𝒰 i)) ∧ ∀ i (f : Γ(X, 𝒰 i)), IsLocalizedModule.Away f (resBasicOpen F f)]

theorem **quasicoherent_local** : (F.IsQuasicoherent ↔ ∀ U : X.Opens, IsAffineOpen U → ∀ f : Γ(X, U), IsLocalizedModule.Away f (resBasicOpen F f)) ∧ (∀ (ι : Type u) (𝒰 : ι → X.Opens), IsOpenCover 𝒰 → (∀ i, IsAffineOpen (𝒰 i)) → (F.IsQuasicoherent ↔ ∀ i (f : Γ(X, 𝒰 i)), IsLocalizedModule.Away f (resBasicOpen F f))) ∧ (F.IsQuasicoherent → ∀ (U V : X.Opens), IsAffineOpen U → IsAffineOpen V → ∀ h : V ≤ U, letI

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsRegularLocalRingAndRingKrullDimEqOfPolAV

theorem **isRegularLocalRing_and_ringKrullDim_eq_of_polAV** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t₀ : Spec (CommRingCat.of R) ⟶ P.A.left) (m : Ideal R) [m.IsMaximal] (O : Type) [CommRing O] [Algebra R O] [IsLocalization.AtPrime O m], IsOpenImmersion t₀ → IsRegularLocalRing O ∧ ringKrullDim O = P.g

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsRegularOfSupportDimQuotientAddLengthLe

theorem **IsQuasiRegularSeq.mem_pow_smul_of_map_mem** {ι : Type*} [Finite ι] {f : ι → A} (h : IsQuasiRegularSeq M f) {F : Type*} [AddCommGroup F] [Module A F] (g : F →ₗ[A] M) (hg : ∀ z, g z ∈ (Ideal.span (Set.range f) • ⊤ : Submodule A M) → z ∈ (Ideal.span (Set.range f) • ⊤ : Submodule A F)) (n : ℕ) (z : F) (hz : g z ∈ (Ideal.span (Set.range f) ^ n • ⊤ : Submodule A M)) : z ∈ (Ideal.span (Set.range f) ^ n • ⊤ : Submodule A F)

theorem **sup_pow_smul_top_eq_top** {I : Ideal A} {N : Submodule A M} (h : N ⊔ I • ⊤ = ⊤) (n : ℕ) : N ⊔ I ^ n • ⊤ = ⊤

theorem **flat_of_basis_mod_pow** [IsNoetherianRing A] (B : Type*) [CommRing B] [IsNoetherianRing B] [Algebra A B] [Module B M] [IsScalarTower A B M] [Module.Finite B M] {I : Ideal A} (hI : I.map (algebraMap A B) ≤ (⊥ : Ideal B).jacobson) {S : Type*} (m : S → M) (hsurj : Submodule.span A (Set.range m) ⊔ I • ⊤ = ⊤) (hinj : ∀ (n : ℕ) (z : S →₀ A), Finsupp.linearCombination A m z ∈ (I ^ n • ⊤ : Submodule A M) → ∀ s, z s ∈ I ^ n) : Module.Flat A M

theorem **IsQuasiRegularSeq.flat_of_isMaximal** [IsNoetherianRing A] (B : Type*) [CommRing B] [IsNoetherianRing B] [Algebra A B] [Module B M] [IsScalarTower A B M] [Module.Finite B M] {ι : Type*} [Finite ι] {f : ι → A} (h : IsQuasiRegularSeq M f) (hmax : (Ideal.span (Set.range f)).IsMaximal) (hI : (Ideal.span (Set.range f)).map (algebraMap A B) ≤ (⊥ : Ideal B).jacobson) : Module.Flat A M

theorem **flat_of_isWeaklyRegular_of_le_jacobson** [IsNoetherianRing A] [IsLocalRing A] (B : Type*) [CommRing B] [IsNoetherianRing B] [Algebra A B] [Module B M] [IsScalarTower A B M] [Module.Finite B M] (hB : (maximalIdeal A).map (algebraMap A B) ≤ (⊥ : Ideal B).jacobson) {xs : List A} (hxs : Ideal.ofList xs = maximalIdeal A) (h : IsWeaklyRegular M xs) : Module.Flat A M

theorem **flat_of_isWeaklyRegular_of_isLocalHom** [IsNoetherianRing A] [IsLocalRing A] (B : Type*) [CommRing B] [IsNoetherianRing B] [IsLocalRing B] [Algebra A B] [IsLocalHom (algebraMap A B)] [Module B M] [IsScalarTower A B M] [Module.Finite B M] {xs : List A} (hxs : Ideal.ofList xs = maximalIdeal A) (h : IsWeaklyRegular M xs) : Module.Flat A M

theorem **isLocalRing_quotient_span_singleton** [IsLocalRing R] {x : R} (hx : x ∈ maximalIdeal R) : IsLocalRing (R ⧸ Ideal.span {x})

theorem **ofList_map_mk_eq_maximalIdeal** [IsLocalRing R] {x : R} {xs : List R} (h : Ideal.ofList (x :: xs) = maximalIdeal R) [IsLocalRing (R ⧸ Ideal.span {x})] : Ideal.ofList (xs.map (Ideal.Quotient.mk (Ideal.span {x}))) = maximalIdeal _

theorem **exists_ofList_cons_eq_maximalIdeal** [IsLocalRing R] {xs : List R} (h : Ideal.ofList xs = maximalIdeal R) {x : R} (hx : x ∈ maximalIdeal R) (hx2 : x ∉ maximalIdeal R ^ 2) : ∃ ys : List R, ys.length + 1 = xs.length ∧ Ideal.ofList (x :: ys) = maximalIdeal R

private theorem **is_domain_of_regular_local** (n : ℕ) : ∀ (R : Type u) [CommRing R] [IsNoetherianRing R] [IsLocalRing R] (xs : List R), xs.length = n → Ideal.ofList xs = maximalIdeal R → (n : WithBot ℕ∞) ≤ ringKrullDim R → IsDomain R

theorem **exists_ofList_eq_maximalIdeal_length_eq** (R : Type*) [CommRing R] [IsRegularLocalRing R] : ∃ xs : List R, Ideal.ofList xs = maximalIdeal R ∧ (xs.length : WithBot ℕ∞) = ringKrullDim R

theorem **isDomain_of_isRegularLocalRing** (R : Type*) [CommRing R] [IsRegularLocalRing R] : IsDomain R

private theorem **is_weakly_regular_of_system_parameters** (n : ℕ) : ∀ (R : Type u) [CommRing R] [IsNoetherianRing R] [IsLocalRing R] (xs : List R), xs.length = n → Ideal.ofList xs = maximalIdeal R → (n : WithBot ℕ∞) ≤ ringKrullDim R → IsWeaklyRegular R xs

theorem **isRegular_of_ofList_eq_maximalIdeal** [IsNoetherianRing R] [IsLocalRing R] {xs : List R} (h : Ideal.ofList xs = maximalIdeal R) (hd : (xs.length : WithBot ℕ∞) ≤ ringKrullDim R) : RingTheory.Sequence.IsRegular R xs

theorem **isRegular_of_length_eq_spanFinrank** [IsRegularLocalRing R] {xs : List R} (h : Ideal.ofList xs = maximalIdeal R) (hd : xs.length = (maximalIdeal R).spanFinrank) : RingTheory.Sequence.IsRegular R xs

theorem **length_le_ringKrullDim_quotient_of_isAssociatedPrime** {M : Type*} [AddCommGroup M] [Module R M] [Module.Finite R M] {xs : List R} (hm : ∀ x ∈ xs, x ∈ maximalIdeal R) (h : IsWeaklyRegular M xs) {p : Ideal R} (hp : IsAssociatedPrime p M) : (xs.length : WithBot ℕ∞) ≤ ringKrullDim (R ⧸ p)

theorem **exists_isRegular_quotSMulTop** [Small.{v} R] {M : Type v} [AddCommGroup M] [Module R M] [Module.Finite R M] {xs : List R} (h : RingTheory.Sequence.IsRegular M xs) {n : ℕ} (hn : n + 1 ≤ xs.length) {y : R} (hy : y ∈ maximalIdeal R) (hreg : IsSMulRegular M y) : ∃ zs : List R, zs.length = n ∧ RingTheory.Sequence.IsRegular (QuotSMulTop y M) zs

theorem **supportDim_le_supportDim_quotient_add_length** {M : Type*} [AddCommGroup M] [Module R M] [Module.Finite R M] {ys : List R} (hm : ∀ y ∈ ys, y ∈ maximalIdeal R) : Module.supportDim R M ≤ Module.supportDim R (M ⧸ Ideal.ofList ys • (⊤ : Submodule R M)) + ys.length

theorem **isRegular_of_supportDim_quotient_add_length_le** [Small.{v} R] {M : Type v} [AddCommGroup M] [Module R M] [Module.Finite R M] {xs : List R} (h : RingTheory.Sequence.IsRegular M xs) {ys : List R} (hm : ∀ y ∈ ys, y ∈ maximalIdeal R) (hd : Module.supportDim R (M ⧸ Ideal.ofList ys • (⊤ : Submodule R M)) + ys.length ≤ xs.length) : RingTheory.Sequence.IsRegular M ys

theorem **isCM_of_isRegularLocalRing** (R : Type*) [CommRing R] [IsRegularLocalRing R] : IsCM R

theorem **isRegular_of_isCM_of_ringKrullDim_quotient_add_length_le** [IsLocalRing R] (hR : IsCM R) {ys : List R} (hm : ∀ y ∈ ys, y ∈ maximalIdeal R) (hd : ringKrullDim (R ⧸ Ideal.ofList ys) + ys.length ≤ ringKrullDim R) : RingTheory.Sequence.IsRegular R ys

theorem **ringKrullDim_quotient_le_zero_of_forall_eq** [IsLocalRing R] {I : Ideal R} (h : ∀ p : Ideal R, p.IsPrime → I ≤ p → p = maximalIdeal R) : ringKrullDim (R ⧸ I) ≤ 0

theorem **isRegular_of_isCM_of_radical_eq** [IsLocalRing R] (hR : IsCM R) {ys : List R} (hr : (Ideal.ofList ys).radical = maximalIdeal R) (hd : (ys.length : WithBot ℕ∞) ≤ ringKrullDim R) : RingTheory.Sequence.IsRegular R ys

theorem **isRegular_of_radical_eq_maximalIdeal** [IsRegularLocalRing R] {ys : List R} (hr : (Ideal.ofList ys).radical = maximalIdeal R) (hd : (ys.length : WithBot ℕ∞) ≤ ringKrullDim R) : RingTheory.Sequence.IsRegular R ys

theorem **flat_of_isCM_of_ringKrullDim_add_le** [IsRegularLocalRing A] [IsLocalRing B] [IsLocalHom (algebraMap A B)] (hB : IsCM B) (hd : ringKrullDim A + ringKrullDim (B ⧸ (maximalIdeal A).map (algebraMap A B)) ≤ ringKrullDim B) : Module.Flat A B

theorem **ringKrullDim_quotient_map_maximalIdeal_le_zero** [IsLocalRing A] [IsLocalRing B] [IsLocalHom (algebraMap A B)] [Algebra.QuasiFinite A B] : ringKrullDim (B ⧸ (maximalIdeal A).map (algebraMap A B)) ≤ 0

theorem **flat_of_isCM_of_quasiFinite** [IsRegularLocalRing A] [IsLocalRing B] [IsLocalHom (algebraMap A B)] [Algebra.QuasiFinite A B] (hB : IsCM B) (hd : ringKrullDim A ≤ ringKrullDim B) : Module.Flat A B

theorem **flat_of_isRegularLocalRing_of_quasiFinite** [IsRegularLocalRing A] [IsRegularLocalRing B] [IsLocalHom (algebraMap A B)] [Algebra.QuasiFinite A B] (hd : ringKrullDim A = ringKrullDim B) : Module.Flat A B

theorem **free_of_isRegularLocalRing_of_finite** [IsRegularLocalRing A] [IsRegularLocalRing B] [IsLocalHom (algebraMap A B)] [Module.Finite A B] (hd : ringKrullDim A = ringKrullDim B) : Module.Free A B

theorem **flat_localization_atPrime_of_isCM** (p : Ideal A) [p.IsPrime] (q : Ideal B) [q.IsPrime] [q.LiesOver p] [IsRegularLocalRing (Localization.AtPrime p)] [Algebra.QuasiFiniteAt A q] (hB : IsCM (Localization.AtPrime q)) (hd : p.height ≤ q.height) : Module.Flat A (Localization.AtPrime q)

theorem **flat_of_isRegularRing_of_isCM** [IsRegularRing A] [Algebra.QuasiFinite A B] (hB : ∀ (q : Ideal B) [q.IsMaximal], IsCM (Localization.AtPrime q)) (hd : ∀ q : Ideal B, q.IsMaximal → (q.under A).height ≤ q.height) : Module.Flat A B

theorem **flat_of_isRegularRing_of_isRegularRing** [IsRegularRing A] [IsRegularRing B] [Algebra.QuasiFinite A B] (hd : ∀ q : Ideal B, q.IsMaximal → (q.under A).height ≤ q.height) : Module.Flat A B

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsSMulRegularQuotIffMemNonZeroDivisors

theorem **IsQuasiRegularSeq.isSMulRegular_powGr** (h : IsQuasiRegularSeq M f) {x : R} (hx : IsSMulRegular (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) x) (a : ℕ) : IsSMulRegular (powGr (Ideal.span (Set.range f)) M a) x

theorem **IsQuasiRegularSeq.mem_pow_smul_top_of_smul_mem** (h : IsQuasiRegularSeq M f) {x : R} (hx : IsSMulRegular (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) x) (a : ℕ) {m : M} (hm : x • m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)) : m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)

theorem **IsQuasiRegularSeq.isSMulRegular_quot_pow** (h : IsQuasiRegularSeq M f) {x : R} (hx : IsSMulRegular (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) x) (a : ℕ) : IsSMulRegular (M ⧸ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)) x

theorem **IsQuasiRegularSeq.map_mem_pow_smul_top_iff** (h : IsQuasiRegularSeq M f) (T : Submonoid R) (hT : ∀ t ∈ T, IsSMulRegular (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) t) {N : Type*} [AddCommGroup N] [Module R N] (g : M →ₗ[R] N) [IsLocalizedModule T g] (a : ℕ) (m : M) : g m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R N) ↔ m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)

theorem **isSMulRegular_quot_smul_top_iff** (J : Ideal R) (x : R) : IsSMulRegular (R ⧸ (J • ⊤ : Submodule R R)) x ↔ IsSMulRegular (R ⧸ J) x

theorem **isSMulRegular_quot_iff_mem_nonZeroDivisors** (J : Ideal R) (x : R) : IsSMulRegular (R ⧸ J) x ↔ Ideal.Quotient.mk J x ∈ nonZeroDivisors (R ⧸ J)

theorem **IsQuasiRegularSeq.isSMulRegular_powGr_self** (h : IsQuasiRegularSeq R f) {x : R} (hx : IsSMulRegular (R ⧸ Ideal.span (Set.range f)) x) (a : ℕ) : IsSMulRegular (powGr (Ideal.span (Set.range f)) R a) x

theorem **IsQuasiRegularSeq.isSMulRegular_quot_pow_self** (h : IsQuasiRegularSeq R f) {x : R} (hx : IsSMulRegular (R ⧸ Ideal.span (Set.range f)) x) (a : ℕ) : IsSMulRegular (R ⧸ Ideal.span (Set.range f) ^ a) x

theorem **IsQuasiRegularSeq.mem_pow_of_mul_mem** (h : IsQuasiRegularSeq R f) {x : R} (hx : IsSMulRegular (R ⧸ Ideal.span (Set.range f)) x) (a : ℕ) {y : R} (hy : x * y ∈ Ideal.span (Set.range f) ^ a) : y ∈ Ideal.span (Set.range f) ^ a

theorem **IsQuasiRegularSeq.comap_map_pow** (h : IsQuasiRegularSeq R f) (T : Submonoid R) (hT : ∀ t ∈ T, IsSMulRegular (R ⧸ Ideal.span (Set.range f)) t) (S : Type*) [CommRing S] [Algebra R S] [IsLocalization T S] (a : ℕ) : ((Ideal.span (Set.range f) ^ a).map (algebraMap R S)).comap (algebraMap R S) = Ideal.span (Set.range f) ^ a

theorem **isSMulRegular_quot_of_notMem** {P : Ideal R} [P.IsPrime] {x : R} (hx : x ∉ P) : IsSMulRegular (R ⧸ P) x

theorem **IsQuasiRegularSeq.mem_pow_of_mul_mem_of_notMem** (h : IsQuasiRegularSeq R f) [(Ideal.span (Set.range f)).IsPrime] {x y : R} (hx : x ∉ Ideal.span (Set.range f)) (a : ℕ) (hy : x * y ∈ Ideal.span (Set.range f) ^ a) : y ∈ Ideal.span (Set.range f) ^ a

theorem **radical_pow_of_isPrime** (P : Ideal R) [hP : P.IsPrime] {a : ℕ} (ha : a ≠ 0) : (P ^ a).radical = P

theorem **IsQuasiRegularSeq.isPrimary_pow** (h : IsQuasiRegularSeq R f) [hI : (Ideal.span (Set.range f)).IsPrime] {a : ℕ} (ha : a ≠ 0) : (Ideal.span (Set.range f) ^ a).IsPrimary

theorem **IsQuasiRegularSeq.comap_map_pow_atPrime** (h : IsQuasiRegularSeq R f) [(Ideal.span (Set.range f)).IsPrime] [IsLocalization.AtPrime S (Ideal.span (Set.range f))] (a : ℕ) : ((Ideal.span (Set.range f) ^ a).map (algebraMap R S)).comap (algebraMap R S) = Ideal.span (Set.range f) ^ a

theorem **IsQuasiRegularSeq.comap_maximalIdeal_pow** (h : IsQuasiRegularSeq R f) [(Ideal.span (Set.range f)).IsPrime] [IsLocalRing S] [IsLocalization.AtPrime S (Ideal.span (Set.range f))] (a : ℕ) : (IsLocalRing.maximalIdeal S ^ a).comap (algebraMap R S) = Ideal.span (Set.range f) ^ a

theorem **IsQuasiRegularSeq.algebraMap_mem_maximalIdeal_pow_iff** (h : IsQuasiRegularSeq R f) [(Ideal.span (Set.range f)).IsPrime] [IsLocalRing S] [IsLocalization.AtPrime S (Ideal.span (Set.range f))] (a : ℕ) (y : R) : algebraMap R S y ∈ IsLocalRing.maximalIdeal S ^ a ↔ y ∈ Ideal.span (Set.range f) ^ a

theorem **MonoDegLE.card** (ι : Type*) [Fintype ι] (k : ℕ) : Fintype.card (MonoDegLE ι k) = (k + Fintype.card ι).choose (Fintype.card ι)

theorem **MonoDegLE.sum_eq_sum_range** {ι N : Type*} [Finite ι] [AddCommMonoid N] (k : ℕ) (g : (ι →₀ ℕ) → N) : ∑ α : MonoDegLE ι k, g α.1 = ∑ a ∈ Finset.range (k + 1), ∑ α : MonoDeg ι a, g α.1

theorem **IsQuasiRegularSeq.eq_zero_of_sum_range_mem** {f : ι → R} (h : IsQuasiRegularSeq M f) (c : (ι →₀ ℕ) → M) (hc : ∀ α, c α ∈ (Ideal.span (Set.range f) • ⊤ : Submodule R M) → c α = 0) (n : ℕ) (hn : ∑ a ∈ Finset.range n, ∑ α : MonoDeg ι a, monoVal f α.1 • c α.1 ∈ (Ideal.span (Set.range f) ^ n • ⊤ : Submodule R M)) : ∀ α, α.degree < n → c α = 0

theorem **exists_sub_sum_range_mem** (f : ι → R) {N : Type*} (s : N → M) (hs : ∀ m, ∃ y, m - s y ∈ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) (n : ℕ) (m : M) : ∃ c : (ι →₀ ℕ) → N, m - ∑ a ∈ Finset.range n, ∑ α : MonoDeg ι a, monoVal f α.1 • s (c α.1) ∈ (Ideal.span (Set.range f) ^ n • ⊤ : Submodule R M)

theorem **monoLEToQuotPow_apply** (f : ι → R) (s : N →ₗ[A] M) (k : ℕ) (c : MonoDegLE ι k → N) : monoLEToQuotPow f s k c = Submodule.Quotient.mk (∑ α, monoVal f α.1 • s (c α))

theorem **monoLEToQuotPow_single** [DecidableEq ι] (f : ι → R) (s : N →ₗ[A] M) (k : ℕ) (α : MonoDegLE ι k) (y : N) : monoLEToQuotPow f s k (Pi.single α y) = Submodule.Quotient.mk (monoVal f α.1 • s y)

theorem **monoLEToQuotPow_surjective** (f : ι → R) (s : N →ₗ[A] M) (hs : Function.Surjective fun y => (Submodule.Quotient.mk (s y) : M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M))) (k : ℕ) : Function.Surjective (monoLEToQuotPow f s k)

theorem **IsQuasiRegularSeq.monoLEToQuotPow_injective** {f : ι → R} (h : IsQuasiRegularSeq M f) (s : N →ₗ[A] M) (hs : Function.Injective fun y => (Submodule.Quotient.mk (s y) : M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M))) (k : ℕ) : Function.Injective (monoLEToQuotPow f s k)

private theorem **mk_linearCombination** (P : Submodule R M) {κ : Type*} (m : κ → M) : (fun y : κ →₀ A => (Submodule.Quotient.mk (Finsupp.linearCombination A m y) : M ⧸ P)) = Finsupp.linearCombination A fun j => (Submodule.Quotient.mk (m j) : M ⧸ P)

private theorem **monoLEToQuotPow_comp_basis** (f : ι → R) {κ : Type*} (m : κ → M) (k : ℕ) : monoLEToQuotPow f (Finsupp.linearCombination A m) k ∘ (Pi.basis fun _ : MonoDegLE ι k => Finsupp.basisSingleOne (R

theorem **IsQuasiRegularSeq.linearIndependent_monoVal_smul** {f : ι → R} (h : IsQuasiRegularSeq M f) {κ : Type*} {m : κ → M} (hm : LinearIndependent A fun j => (Submodule.Quotient.mk (m j) : M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M))) (k : ℕ) : LinearIndependent A fun p : MonoDegLE ι k × κ => (Submodule.Quotient.mk (monoVal f p.1.1 • m p.2) : M ⧸ (Ideal.span (Set.range f) ^ (k + 1) • ⊤ : Submodule R M))

theorem **span_monoVal_smul_eq_top** (f : ι → R) {κ : Type*} {m : κ → M} (hm : Submodule.span A (Set.range fun j => (Submodule.Quotient.mk (m j) : M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M))) = ⊤) (k : ℕ) : Submodule.span A (Set.range fun p : MonoDegLE ι k × κ => (Submodule.Quotient.mk (monoVal f p.1.1 • m p.2) : M ⧸ (Ideal.span (Set.range f) ^ (k + 1) • ⊤ : Submodule R M))) = ⊤

@[simp] theorem **IsQuasiRegularSeq.quotPowBasis_apply** {f : ι → R} (h : IsQuasiRegularSeq M f) {κ : Type*} (b : Module.Basis κ A (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M))) (m : κ → M) (hm : ∀ j, Submodule.Quotient.mk (m j) = b j) (k : ℕ) (p : MonoDegLE ι k × κ) : h.quotPowBasis b m hm k p = Submodule.Quotient.mk (monoVal f p.1.1 • m p.2)

@[simp] theorem **quotSMulTopEquivQuotOver_mk** (J : Ideal R) (x : R) : quotSMulTopEquivQuotOver A J (Submodule.Quotient.mk x) = Ideal.Quotient.mk J x

@[simp] theorem **IsQuasiRegularSeq.quotPowBasisSelf_apply** (h : IsQuasiRegularSeq R f) {κ : Type*} (b : Module.Basis κ A (R ⧸ Ideal.span (Set.range f))) (r : κ → R) (hr : ∀ j, Ideal.Quotient.mk _ (r j) = b j) (k : ℕ) (p : MonoDegLE ι k × κ) : h.quotPowBasisSelf b r hr k p = Ideal.Quotient.mk _ (monoVal f p.1.1 * r p.2)

theorem **IsQuasiRegularSeq.nonempty_basis_quot_pow_self** {ι : Type*} [Fintype ι] {f : ι → R} (h : IsQuasiRegularSeq R f) {d : ℕ} (b : Module.Basis (Fin d) A (R ⧸ Ideal.span (Set.range f))) (k : ℕ) : Nonempty (Module.Basis (Fin (d * (k + Fintype.card ι).choose (Fintype.card ι))) A (R ⧸ Ideal.span (Set.range f) ^ (k + 1)))

theorem **IsQuasiRegularSeq.powers** {R : Type u} [CommRing R] {n : ℕ} {f : Fin n → R} (h : IsQuasiRegularSeq R f) : (∀ x : R, IsSMulRegular (R ⧸ Ideal.span (Set.range f)) x → ∀ a : ℕ, IsSMulRegular (powGr (Ideal.span (Set.range f)) R a) x ∧ IsSMulRegular (R ⧸ Ideal.span (Set.range f) ^ a) x) ∧ (∀ [(Ideal.span (Set.range f)).IsPrime] (a : ℕ), (a ≠ 0 → (Ideal.span (Set.range f) ^ a).IsPrimary ∧ (Ideal.span (Set.range f) ^ a).radical = Ideal.span (Set.range f)) ∧ ((Ideal.span (Set.range f) ^ a).map (algebraMap R (Localization.AtPrime (Ideal.span (Set.range f))))).comap (algebraMap R (Localization.AtPrime (Ideal.span (Set.range f)))) = Ideal.span (Set.range f) ^ a ∧ ∀ y : R, algebraMap R (Localization.AtPrime (Ideal.span (Set.range f))) y ∈ IsLocalRing.maximalIdeal (Localization.AtPrime (Ideal.span (Set.range f))) ^ a ↔ y ∈ Ideal.span (Set.range f) ^ a) ∧ (∀ (A : Type v) [CommRing A] [Algebra A

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsSaturatedSubmoduleIffExistsIsComplOfValuationR

theorem **exists_isCompl_of_projective_quotient** (N : Submodule R M) [Module.Projective R (M ⧸ N)] : ∃ N' : Submodule R M, IsCompl N N'

theorem **finite_of_isCompl** [Module.Finite R M] {N N' : Submodule R M} (h : IsCompl N N') : Module.Finite R N

theorem **exists_disjoint_sup_eq_of_isCompl_submoduleOf** {P N : Submodule R M} (hle : N ≤ P) {Q : Submodule R P} (h : IsCompl (N.submoduleOf P) Q) : ∃ N' : Submodule R M, Disjoint N N' ∧ N ⊔ N' = P

theorem **projective_of_finite_of_isTorsionFree** [Module.Finite R M] [Module.IsTorsionFree R M] : Module.Projective R M

theorem **IsSaturatedSubmodule.projective_quotient** [Module.Finite R M] {N : Submodule R M} (h : IsSaturatedSubmodule N) : Module.Projective R (M ⧸ N)

theorem **IsSaturatedSubmodule.exists_isCompl** [Module.Finite R M] {N : Submodule R M} (h : IsSaturatedSubmodule N) : ∃ N' : Submodule R M, IsCompl N N'

theorem **IsSaturatedIn.exists_disjoint_sup_eq** {P N : Submodule R M} [Module.Finite R P] (h : IsSaturatedIn P N) : ∃ N' : Submodule R M, Disjoint N N' ∧ N ⊔ N' = P

theorem **free_of_finite_of_isTorsionFree_of_valuationRing** [Module.Finite R M] [Module.IsTorsionFree R M] : Module.Free R M

theorem **IsSaturatedSubmodule.free_quotient_of_valuationRing** [Module.Finite R M] {N : Submodule R M} (h : IsSaturatedSubmodule N) : Module.Free R (M ⧸ N)

theorem **IsSaturatedSubmodule.exists_isCompl_of_valuationRing** [Module.Finite R M] {N : Submodule R M} (h : IsSaturatedSubmodule N) : ∃ N' : Submodule R M, IsCompl N N'

theorem **IsSaturatedSubmodule.finite_of_valuationRing** [Module.Finite R M] {N : Submodule R M} (h : IsSaturatedSubmodule N) : Module.Finite R N

theorem **IsSaturatedSubmodule.free_of_valuationRing** [Module.Finite R M] [Module.IsTorsionFree R M] {N : Submodule R M} (h : IsSaturatedSubmodule N) : Module.Free R N

theorem **IsSaturatedIn.finite_of_valuationRing** {P N : Submodule R M} [Module.Finite R P] (h : IsSaturatedIn P N) : Module.Finite R N

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsWeaklyRegularIffIsQuasiRegularSeqOfIsLocalRing

theorem **MonoDeg.mulVar_injective** (i : ι) (a : ℕ) : Function.Injective (MonoDeg.mulVar i : MonoDeg ι a → MonoDeg ι (a + 1))

theorem **monoVal_mulVar** (f : ι → R) (i : ι) {a : ℕ} (γ : MonoDeg ι a) : monoVal f (MonoDeg.mulVar i γ).1 = f i * monoVal f γ.1

theorem **monoVal_toOption** (f : Option ι → R) {a : ℕ} (β : MonoDeg ι a) : monoVal f β.toOption.1 = monoVal (f ∘ some) β.1

theorem **sum_monoVal_smul_option** (f : Option ι → R) (a : ℕ) (c : MonoDeg (Option ι) (a + 1) → M) : ∑ α, monoVal f α.1 • c α = ∑ β : MonoDeg ι (a + 1), monoVal (f ∘ some) β.1 • c β.toOption + f none • ∑ γ : MonoDeg (Option ι) a, monoVal f γ.1 • c (MonoDeg.mulVar none γ)

theorem **mem_pow_smul_iff** (f : ι → R) (a : ℕ) (N : Submodule R M) (x : M) : x ∈ Ideal.span (Set.range f) ^ a • N ↔ ∃ c : MonoDeg ι a → M, (∀ α, c α ∈ N) ∧ ∑ α, monoVal f α.1 • c α = x

theorem **sum_monoVal_smul_mem_pow_succ** (f : ι → R) {a : ℕ} {c : MonoDeg ι a → M} (hc : ∀ α, c α ∈ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) : ∑ α, monoVal f α.1 • c α ∈ (Ideal.span (Set.range f) ^ (a + 1) • ⊤ : Submodule R M)

theorem **isQuasiRegularSeq_iff_eq_zero** (f : ι → R) : IsQuasiRegularSeq M f ↔ ∀ (a : ℕ) (c : MonoDeg ι a → M), ∑ α, monoVal f α.1 • c α = 0 → ∀ α, c α ∈ (Ideal.span (Set.range f) • ⊤ : Submodule R M)

private theorem **mem_pow_of_regular** {f : ι → R} (h : IsQuasiRegularSeq M f) {x : R} (hx : IsSMulRegular (M ⧸ (Ideal.span (Set.range f) • ⊤ : Submodule R M)) x) (a : ℕ) (m : M) (hm : x • m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)) : m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)

theorem **IsQuasiRegularSeq.option** {f : Option ι → R} (h : IsQuasiRegularSeq M (f ∘ some)) (hx : IsSMulRegular (M ⧸ (Ideal.span (Set.range (f ∘ some)) • ⊤ : Submodule R M)) (f none)) : IsQuasiRegularSeq M f

theorem **IsQuasiRegularSeq.mem_pow_succ_smul_of_smul_mem** {f : ι → R} (h : IsQuasiRegularSeq M f) (i : ι) {a : ℕ} {m : M} (hm : m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)) (hfm : f i • m ∈ (Ideal.span (Set.range f) ^ (a + 2) • ⊤ : Submodule R M)) : m ∈ (Ideal.span (Set.range f) ^ (a + 1) • ⊤ : Submodule R M)

theorem **IsQuasiRegularSeq.mem_pow_smul_of_smul_mem** {f : ι → R} (h : IsQuasiRegularSeq M f) (i : ι) {a : ℕ} {m : M} (hfm : f i • m ∈ (Ideal.span (Set.range f) ^ (a + 1) • ⊤ : Submodule R M)) : m ∈ (Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M)

theorem **IsQuasiRegularSeq.isSMulRegular_of_iInf_eq_bot** {f : ι → R} (h : IsQuasiRegularSeq M f) (hsep : (⨅ a : ℕ, Ideal.span (Set.range f) ^ a • ⊤ : Submodule R M) = ⊥) (i : ι) : IsSMulRegular M (f i)

theorem **IsQuasiRegularSeq.quotSMulTop** {f : Option ι → R} (h : IsQuasiRegularSeq M f) : IsQuasiRegularSeq (QuotSMulTop (f none) M) (f ∘ some)

theorem **ofList_ofFn_eq_span_range** {n : ℕ} (f : Fin n → R) : Ideal.ofList (List.ofFn f) = Ideal.span (Set.range f)

theorem **isQuasiRegularSeq_of_isWeaklyRegular** {n : ℕ} {f : Fin n → R} (h : IsWeaklyRegular M (List.ofFn f)) : IsQuasiRegularSeq M f

theorem **isQuasiRegularSeq_get_of_isWeaklyRegular** {rs : List R} (h : IsWeaklyRegular M rs) : IsQuasiRegularSeq M rs.get

theorem **isQuasiRegularSeq_of_isRegular** {n : ℕ} {f : Fin n → R} (h : RingTheory.Sequence.IsRegular M (List.ofFn f)) : IsQuasiRegularSeq M f

theorem **IsQuasiRegularSeq.quotSMulTop_tail** {n : ℕ} {f : Fin (n + 1) → R} (h : IsQuasiRegularSeq M f) : IsQuasiRegularSeq (QuotSMulTop (f 0) M) fun i : Fin n => f i.succ

theorem **IsQuasiRegularSeq.isWeaklyRegular** [IsNoetherianRing R] [Module.Finite R M] {n : ℕ} {f : Fin n → R} (h : IsQuasiRegularSeq M f) (hf : ∀ i, f i ∈ (⊥ : Ideal R).jacobson) : IsWeaklyRegular M (List.ofFn f)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.IsWeaklyRegularMapCAppendMapLinForm

theorem **killVars_rename** (S : Set σ) (p : MvPolynomial {x : σ // x ∉ S} R) : killVars R S (rename Subtype.val p) = p

theorem **killVars_X_of_notMem** {S : Set σ} {x : σ} (hx : x ∉ S) : killVars R S (X x) = X (x, hx)

theorem **killVars_X_of_mem** {S : Set σ} {x : σ} (hx : x ∈ S) : killVars R S (X x) = 0

theorem **killVars_C** (S : Set σ) (r : R) : killVars R S (C r) = C r

theorem **killVars_surjective** (S : Set σ) : Function.Surjective (killVars R S)

theorem **sub_rename_killVars_mem** (S : Set σ) (p : MvPolynomial σ R) : p - rename Subtype.val (killVars R S p) ∈ Ideal.span (X '' S : Set (MvPolynomial σ R))

theorem **comap_killVars_map_C** (S : Set σ) (I : Ideal R) : (I.map (C : R →+* MvPolynomial {x : σ // x ∉ S} R)).comap (killVars R S) = I.map (C : R →+* MvPolynomial σ R) ⊔ Ideal.span (X '' S)

theorem **ker_killVars** (S : Set σ) : RingHom.ker (killVars R S) = Ideal.span (X '' S)

theorem **ker_map_comp_killVars** (S : Set σ) (I : Ideal R) : RingHom.ker ((MvPolynomial.map (Ideal.Quotient.mk I)).comp (killVars R S).toRingHom) = I.map (C : R →+* MvPolynomial σ R) ⊔ Ideal.span (X '' S)

@[simp] theorem **quotientSpanXEquiv_mk** (S : Set σ) (p : MvPolynomial σ R) : quotientSpanXEquiv R S (Ideal.Quotient.mk _ p) = killVars R S p

theorem **mem_map_C_of_X_mul_mem** (x : σ) (I : Ideal R) {p : MvPolynomial σ R} (hp : X x * p ∈ I.map (C : R →+* MvPolynomial σ R)) : p ∈ I.map (C : R →+* MvPolynomial σ R)

theorem **mem_of_X_mul_mem** {S : Set σ} {x : σ} (hx : x ∉ S) (I : Ideal R) {p : MvPolynomial σ R} (hp : X x * p ∈ I.map (C : R →+* MvPolynomial σ R) ⊔ Ideal.span (X '' S)) : p ∈ I.map (C : R →+* MvPolynomial σ R) ⊔ Ideal.span (X '' S)

theorem **ofList_map_C_append_map_X** (rs : List R) (l : List σ) : Ideal.ofList (rs.map (C : R →+* MvPolynomial σ R) ++ l.map X) = (Ideal.ofList rs).map C ⊔ Ideal.span (X '' {x | x ∈ l})

theorem **isWeaklyRegular_map_C_append_map_X** {rs : List R} (hrs : IsWeaklyRegular R rs) {l : List σ} (hl : l.Nodup) : IsWeaklyRegular (MvPolynomial σ R) (rs.map (C : R →+* MvPolynomial σ R) ++ l.map X)

theorem **isRegular_map_C_append_map_X** {rs : List R} (hrs : RingTheory.Sequence.IsRegular R rs) {l : List σ} (hl : l.Nodup) : RingTheory.Sequence.IsRegular (MvPolynomial σ R) (rs.map (C : R →+* MvPolynomial σ R) ++ l.map X)

theorem **isWeaklyRegular_map_X** {l : List σ} (hl : l.Nodup) : IsWeaklyRegular (MvPolynomial σ R) (l.map X : List (MvPolynomial σ R))

theorem **isRegular_map_X** [Nontrivial R] {l : List σ} (hl : l.Nodup) : RingTheory.Sequence.IsRegular (MvPolynomial σ R) (l.map X : List (MvPolynomial σ R))

theorem **isWeaklyRegular_ofFn_X** {n : ℕ} {e : Fin n → σ} (he : Function.Injective e) : IsWeaklyRegular (MvPolynomial σ R) (List.ofFn fun k => (X (e k) : MvPolynomial σ R))

theorem **isRegular_ofFn_X** [Nontrivial R] {n : ℕ} {e : Fin n → σ} (he : Function.Injective e) : RingTheory.Sequence.IsRegular (MvPolynomial σ R) (List.ofFn fun k => (X (e k) : MvPolynomial σ R))

theorem **isPrime_map_C_sup_span_X** (S : Set σ) (I : Ideal R) [I.IsPrime] : (I.map (C : R →+* MvPolynomial σ R) ⊔ Ideal.span (X '' S)).IsPrime

@[simp] theorem **mem_elimIndices** {S : Set ι} {j₀ : J} {x : ι × J} : x ∈ elimIndices S j₀ ↔ x.1 ∈ S ∧ x.2 = j₀

theorem **linForm_eq** [DecidableEq J] (u : J → R) (i : ι) (j₀ : J) : linForm u i = C (u j₀) * X (i, j₀) + ∑ j ∈ Finset.univ.erase j₀, C (u j) * X (i, j)

theorem **linFormHom_X_of_mem** (u : J → R) {S : Set ι} (j₀ : J) {i : ι} (hi : i ∈ S) : linFormHom u S j₀ (X (i, j₀)) = linForm u i

theorem **linFormHom_X_of_notMem** (u : J → R) {S : Set ι} {j₀ : J} {x : ι × J} (hx : x ∉ elimIndices S j₀) : linFormHom u S j₀ (X x) = X x

theorem **linFormHom_linForm** [DecidableEq J] (u v : J → R) {S : Set ι} (j₀ : J) {i : ι} (hi : i ∈ S) : linFormHom u S j₀ (linForm v i) = linForm (fun j => if j = j₀ then v j₀ * u j₀ else v j₀ * u j + v j) i

theorem **linFormHom_comp_eq_id** {u v : J → R} (S : Set ι) {j₀ : J} (h₀ : v j₀ * u j₀ = 1) (h : ∀ j, j ≠ j₀ → v j₀ * u j + v j = 0) : (linFormHom u S j₀).comp (linFormHom v S j₀) = AlgHom.id R _

theorem **linFormEquiv_X_of_mem** {S : Set ι} {i : ι} (hi : i ∈ S) : linFormEquiv u S j₀ hu (X (i, j₀)) = linForm u i

theorem **linFormEquiv_X_of_notMem** {S : Set ι} {x : ι × J} (hx : x ∉ elimIndices S j₀) : linFormEquiv u S j₀ hu (X x) = X x

theorem **linFormEquiv_C** (S : Set ι) (r : R) : linFormEquiv u S j₀ hu (C r) = C r

theorem **linFormEquiv_symm_linForm** {S : Set ι} {i : ι} (hi : i ∈ S) : (linFormEquiv u S j₀ hu).symm (linForm u i) = X (i, j₀)

theorem **linFormEquiv_symm_X_of_notMem** {S : Set ι} {x : ι × J} (hx : x ∉ elimIndices S j₀) : (linFormEquiv u S j₀ hu).symm (X x) = X x

theorem **linFormEquiv_symm_X_of_mem** {S : Set ι} {i : ι} (hi : i ∈ S) : (linFormEquiv u S j₀ hu).symm (X (i, j₀)) = linForm (linFormInvCoeff u j₀ hu) i

theorem **image_linFormEquiv_X** (S : Set ι) : linFormEquiv u S j₀ hu '' (X '' elimIndices S j₀) = linForm u '' S

theorem **linFormElim_apply** (S : Set ι) (p : MvPolynomial (ι × J) R) : linFormElim u hu S p = killVars R (elimIndices S j₀) ((linFormEquiv u S j₀ hu).symm p)

theorem **linFormElim_X_of_notMem** {S : Set ι} {x : ι × J} (hx : x ∉ elimIndices S j₀) : linFormElim u hu S (X x) = X ⟨x, hx⟩

theorem **linFormElim_X_of_mem** {S : Set ι} {i : ι} (hi : i ∈ S) [DecidableEq J] : linFormElim u hu S (X (i, j₀)) = -∑ j : {j : J // j ≠ j₀}, C ((↑hu.unit⁻¹ : R) * u j) * X ((i, j.1), fun h => j.2 h.2)

theorem **linFormElim_linForm** {S : Set ι} {i : ι} (hi : i ∈ S) : linFormElim u hu S (linForm u i) = 0

theorem **linFormElim_rename** (S : Set ι) (p : MvPolynomial {x : ι × J // x ∉ elimIndices S j₀} R) : linFormElim u hu S (rename Subtype.val p) = p

theorem **linFormElim_surjective** (S : Set ι) : Function.Surjective (linFormElim u hu S)

theorem **map_linFormEquiv_map_C_sup_span_X** (S : Set ι) (I : Ideal R) : (I.map (C : R →+* MvPolynomial (ι × J) R) ⊔ Ideal.span (X '' elimIndices S j₀)).map (linFormEquiv u S j₀ hu) = I.map C ⊔ Ideal.span (linForm u '' S)

theorem **comap_linFormElim_map_C** (S : Set ι) (I : Ideal R) : (I.map (C : R →+* MvPolynomial {x : ι × J // x ∉ elimIndices S j₀} R)).comap (linFormElim u hu S) = I.map C ⊔ Ideal.span (linForm u '' S)

theorem **ker_linFormElim** (S : Set ι) : RingHom.ker (linFormElim u hu S) = Ideal.span (linForm u '' S)

theorem **ker_map_comp_linFormElim** (S : Set ι) (I : Ideal R) : RingHom.ker ((MvPolynomial.map (Ideal.Quotient.mk I)).comp (linFormElim u hu S).toRingHom) = I.map C ⊔ Ideal.span (linForm u '' S)

@[simp] theorem **quotientSpanLinFormEquiv_mk** (S : Set ι) (p : MvPolynomial (ι × J) R) : quotientSpanLinFormEquiv u hu S (Ideal.Quotient.mk _ p) = linFormElim u hu S p

theorem **quotientSpanLinFormEquiv_symm_apply** (S : Set ι) (p : MvPolynomial {x : ι × J // x ∉ elimIndices S j₀} R) : (quotientSpanLinFormEquiv u hu S).symm p = Ideal.Quotient.mk _ (rename Subtype.val p)

@[simp] theorem **quotientMapCSupSpanLinFormEquiv_mk** (S : Set ι) (I : Ideal R) (p : MvPolynomial (ι × J) R) : quotientMapCSupSpanLinFormEquiv u hu S I (Ideal.Quotient.mk _ p) = MvPolynomial.map (Ideal.Quotient.mk I) (linFormElim u hu S p)

theorem **range_linForm_comp** {κ : Type*} (e : κ → ι) : Set.range (fun k => linForm u (e k)) = linForm u '' Set.range e

theorem **isPrime_map_C_sup_span_linForm** (S : Set ι) (I : Ideal R) [I.IsPrime] : (I.map (C : R →+* MvPolynomial (ι × J) R) ⊔ Ideal.span (linForm u '' S)).IsPrime

theorem **isPrime_span_linForm** [IsDomain R] (S : Set ι) : (Ideal.span (linForm u '' S)).IsPrime

private theorem **forall₂_linFormEquiv** (rs : List R) (l : List ι) : List.Forall₂ (fun (r s : MvPolynomial (ι × J) R) => ∀ x : MvPolynomial (ι × J) R, (linFormEquiv u Set.univ j₀ hu).toAddEquiv (r • x) = s • (linFormEquiv u Set.univ j₀ hu).toAddEquiv x) (rs.map C ++ (l.map fun i => (i, j₀)).map X) (rs.map C ++ l.map (linForm u))

theorem **isWeaklyRegular_map_C_append_map_linForm** {rs : List R} (hrs : IsWeaklyRegular R rs) {l : List ι} (hl : l.Nodup) : IsWeaklyRegular (MvPolynomial (ι × J) R) (rs.map (C : R →+* MvPolynomial (ι × J) R) ++ l.map (linForm u))

theorem **isRegular_map_C_append_map_linForm** {rs : List R} (hrs : RingTheory.Sequence.IsRegular R rs) {l : List ι} (hl : l.Nodup) : RingTheory.Sequence.IsRegular (MvPolynomial (ι × J) R) (rs.map (C : R →+* MvPolynomial (ι × J) R) ++ l.map (linForm u))

theorem **isWeaklyRegular_map_linForm** {l : List ι} (hl : l.Nodup) : IsWeaklyRegular (MvPolynomial (ι × J) R) (l.map (linForm u))

theorem **isRegular_map_linForm** [Nontrivial R] {l : List ι} (hl : l.Nodup) : RingTheory.Sequence.IsRegular (MvPolynomial (ι × J) R) (l.map (linForm u))

theorem **isWeaklyRegular_C_cons_map_linForm** {r : R} (hr : r ∈ nonZeroDivisors R) {l : List ι} (hl : l.Nodup) : IsWeaklyRegular (MvPolynomial (ι × J) R) (C r :: l.map (linForm u))

theorem **isRegular_C_cons_map_linForm** {r : R} (hr : r ∈ nonZeroDivisors R) (hr' : ¬ IsUnit r) {l : List ι} (hl : l.Nodup) : RingTheory.Sequence.IsRegular (MvPolynomial (ι × J) R) (C r :: l.map (linForm u))

theorem **isWeaklyRegular_ofFn_linForm** {n : ℕ} {e : Fin n → ι} (he : Function.Injective e) : IsWeaklyRegular (MvPolynomial (ι × J) R) (List.ofFn fun k => linForm u (e k))

theorem **isWeaklyRegular_C_cons_ofFn_linForm** {r : R} (hr : r ∈ nonZeroDivisors R) {n : ℕ} {e : Fin n → ι} (he : Function.Injective e) : IsWeaklyRegular (MvPolynomial (ι × J) R) (C r :: List.ofFn fun k => linForm u (e k))

theorem **linForm_elimination** (S : Set ι) (σ : Type*) : (∃ φ : MvPolynomial (ι × J) R ≃ₐ[R] MvPolynomial (ι × J) R, (∀ i ∈ S, φ (X (i, j₀)) = ∑ j, C (u j) * X (i, j)) ∧ ∀ x : ι × J, ¬ (x.1 ∈ S ∧ x.2 = j₀) → φ (X x) = X x) ∧ (∃ ψ : (MvPolynomial (ι × J) R ⧸ Ideal.span (linForm u '' S)) ≃ₐ[R] MvPolynomial {x : ι × J // x ∉ elimIndices S j₀} R, ∀ x : {x : ι × J // x ∉ elimIndices S j₀}, ψ (Ideal.Quotient.mk _ (X x.1)) = X x) ∧ (∀ l : List ι, l.Nodup → IsWeaklyRegular (MvPolynomial (ι × J) R) (l.map (linForm u)) ∧ (Nontrivial R → RingTheory.Sequence.IsRegular (MvPolynomial (ι × J) R) (l.map (linForm u))) ∧ ∀ r ∈ nonZeroDivisors R, IsWeaklyRegular (MvPolynomial (ι × J) R) (C r :: l.map (linForm u)) ∧ (¬ IsUnit r → RingTheory.Sequence.IsRegular (MvPolynomial (ι × J) R) (C r :: l.map (linForm u)))) ∧ (∀ l : List σ, l.Nodup → IsWeaklyRegular (MvPolynomial σ R) (l.map X : List (MvPolynomial σ R)) ∧

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthBallQuotientEqSumMulFdeg

theorem **_root_.GlobalTheory.PointSupport.exists_algHom_ker_eq** (K : Type) [Field K] [IsAlgClosed K] (R : Type) [CommRing R] [Algebra K R] [Algebra.FiniteType K R] (M : Ideal R) [M.IsMaximal] : ∃ χ : R →ₐ[K] K, RingHom.ker χ = M

theorem **_root_.GlobalTheory.PointSupport.finiteLength_and_finrank_eq_length** : ∀ (K : Type) [Field K] [IsAlgClosed K] (R : Type) [CommRing R] [Algebra K R] [Algebra.FiniteType K R] (χ₀ : R →ₐ[K] K) (O : Type) [CommRing O] [Algebra R O] [(RingHom.ker χ₀).IsPrime] [IsLocalization.AtPrime O (RingHom.ker χ₀)] (H : Type) [AddCommGroup H] [Module R H] [Module K H] [IsScalarTower K R H] [Module.Finite R H], (∀ χ : R →ₐ[K] K, χ ≠ χ₀ → ∃ r : R, χ r ≠ 0 ∧ ∀ z : H, r • z = 0) → IsFiniteLength O (O ⊗[R] H) ∧ Module.Finite K H ∧ (Module.finrank K H : ℕ∞) = Module.length O (O ⊗[R] H)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthBaseChangeQuotientOfMapMaximalIdeal

theorem **nonempty_baseChange_subquotient_equiv** [Module.Flat R S] {X : Type*} [AddCommGroup X] [Module R X] (A B : Submodule R X) : Nonempty ((S ⊗[R] (A ⧸ B.submoduleOf A)) ≃ₗ[S] (A.baseChange S ⧸ (B.baseChange S).submoduleOf (A.baseChange S)))

theorem **length_baseChange_flat** : Module.length S (S ⊗[R] X) = Module.length R X * Module.length S (S ⧸ (maximalIdeal R).map (algebraMap R S))

theorem **length_baseChange_subquotient_flat** (A B : Submodule R X) : Module.length S (A.baseChange S ⧸ (B.baseChange S).submoduleOf (A.baseChange S)) = Module.length R (A ⧸ B.submoduleOf A) * Module.length S (S ⧸ (maximalIdeal R).map (algebraMap R S))

theorem **length_quotient_span_singleton_eq_addVal** {S : Type*} [CommRing S] [IsDomain S] [IsDiscreteValuationRing S] (x : S) : Module.length S (S ⧸ Ideal.span {x}) = addVal S x

theorem **map_maximalIdeal_eq_pow_ramificationIdx** : (maximalIdeal R).map (algebraMap R S) = maximalIdeal S ^ ((maximalIdeal S).ramificationIdx R)

theorem **length_fibre_eq_ramificationIdx** : Module.length S (S ⧸ (maximalIdeal R).map (algebraMap R S)) = ((maximalIdeal S).ramificationIdx R : ℕ)

theorem **ramificationIdx_maximalIdeal_pos** : 0 < (maximalIdeal S).ramificationIdx R

theorem **addVal_algebraMap_uniformizer** {ϖ : R} (hϖ : Irreducible ϖ) : addVal S (algebraMap R S ϖ) = ((maximalIdeal S).ramificationIdx R : ℕ)

theorem **addVal_algebraMap** (x : R) : addVal S (algebraMap R S x) = addVal R x * ((maximalIdeal S).ramificationIdx R : ℕ)

theorem **length_baseChange_dvr** : Module.length S (S ⊗[R] X) = Module.length R X * ((maximalIdeal S).ramificationIdx R : ℕ)

theorem **length_baseChange_subquotient_dvr** (A B : Submodule R X) : Module.length S (A.baseChange S ⧸ (B.baseChange S).submoduleOf (A.baseChange S)) = Module.length R (A ⧸ B.submoduleOf A) * ((maximalIdeal S).ramificationIdx R : ℕ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthCokerOfNotInjective

theorem **addVal_finset_prod** {ι : Type*} (s : Finset ι) (a : ι → R) : IsDiscreteValuationRing.addVal R (∏ i ∈ s, a i) = ∑ i ∈ s, IsDiscreteValuationRing.addVal R (a i)

theorem **length_self_eq_top** : Module.length R R = ⊤

theorem **length_coker_of_injective** (f : M →ₗ[R] M) (hf : Function.Injective f) : Module.length R (M ⧸ LinearMap.range f) = IsDiscreteValuationRing.addVal R (LinearMap.det f)

theorem **length_coker_of_not_injective** (f : M →ₗ[R] M) (hf : LinearMap.ker f ≠ ⊥) : Module.length R (M ⧸ LinearMap.range f) = ⊤

theorem **length_coker** (f : M →ₗ[R] M) : Module.length R (M ⧸ LinearMap.range f) = IsDiscreteValuationRing.addVal R (LinearMap.det f)

theorem **relLength_map_eq_addVal_det** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {V : Type u_2} [AddCommGroup V] [Module R V] {A : Submodule R V} [Module.Free R A] [Module.Finite R A] (g : V →ₗ[R] V) (hg : ∀ x ∈ A, g x ∈ A) : GlobalTheory.relLength A (Submodule.map g A) = IsDiscreteValuationRing.addVal R (LinearMap.det (LinearMap.restrict g hg))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthCommensurableIffOfLeOfLe

theorem **relLength_eq_add_of_le** {A B C : Submodule R V} (hCB : C ≤ B) (hBA : B ≤ A) : relLength A C = relLength A B + relLength B C

theorem **relLength_inf_eq_add** (A B C : Submodule R V) : relLength A (B ⊓ C) = relLength A B + relLength (A ⊓ B) C

theorem **relLength_sup_eq_add** (A B C : Submodule R V) : relLength (A ⊔ B) C = relLength A (B ⊔ C) + relLength B C

theorem **relLength_eq_add_sup_of_le** {A B : Submodule R V} (hBA : B ≤ A) (C : Submodule R V) : relLength A C = relLength A (B ⊔ C) + relLength B C

theorem **relLength_eq_add_inf_of_le** (A : Submodule R V) {B C : Submodule R V} (hCB : C ≤ B) : relLength A C = relLength A B + relLength (A ⊓ B) C

theorem **relLength_le_add** (A B C : Submodule R V) : relLength A C ≤ relLength A B + relLength B C

theorem **relLength_inf_le_add** (A B C : Submodule R V) : relLength A (B ⊓ C) ≤ relLength A B + relLength A C

theorem **relLength_sup_le_add** (A B C : Submodule R V) : relLength (A ⊔ B) C ≤ relLength A C + relLength B C

theorem **relLength_ne_top_trans** {A B C : Submodule R V} (hAB : relLength A B ≠ ⊤) (hBC : relLength B C ≠ ⊤) : relLength A C ≠ ⊤

theorem **relLength_inf_ne_top_iff** {A B C : Submodule R V} : relLength A (B ⊓ C) ≠ ⊤ ↔ relLength A B ≠ ⊤ ∧ relLength A C ≠ ⊤

theorem **relLength_sup_ne_top_iff** {A B C : Submodule R V} : relLength (A ⊔ B) C ≠ ⊤ ↔ relLength A C ≠ ⊤ ∧ relLength B C ≠ ⊤

@[trans] theorem **LengthCommensurable.trans** (hAB : LengthCommensurable A B) (hBC : LengthCommensurable B C) : LengthCommensurable A C

theorem **lengthCommensurable_equivalence** : Equivalence (LengthCommensurable : Submodule R V → Submodule R V → Prop)

theorem **lengthCommensurable_iff_of_le_of_le** (hCB : C ≤ B) (hBA : B ≤ A) : LengthCommensurable A C ↔ LengthCommensurable A B ∧ LengthCommensurable B C

theorem **of_between_left** (h : LengthCommensurable A C) (hCB : C ≤ B) (hBA : B ≤ A) : LengthCommensurable A B

theorem **of_between_right** (h : LengthCommensurable A C) (hCB : C ≤ B) (hBA : B ≤ A) : LengthCommensurable B C

theorem **inf_right** (hAB : LengthCommensurable A B) (hAC : LengthCommensurable A C) : LengthCommensurable A (B ⊓ C)

theorem **sup_right** (hAB : LengthCommensurable A B) (hAC : LengthCommensurable A C) : LengthCommensurable A (B ⊔ C)

theorem **inf_left** (hAC : LengthCommensurable A C) (hBC : LengthCommensurable B C) : LengthCommensurable (A ⊓ B) C

theorem **sup_left** (hAC : LengthCommensurable A C) (hBC : LengthCommensurable B C) : LengthCommensurable (A ⊔ B) C

theorem **sup_inf** (h : LengthCommensurable A B) : LengthCommensurable (A ⊔ B) (A ⊓ B)

theorem **inf** (hA : LengthCommensurable A A') (hB : LengthCommensurable B B') : LengthCommensurable (A ⊓ B) (A' ⊓ B')

theorem **sup** (hA : LengthCommensurable A A') (hB : LengthCommensurable B B') : LengthCommensurable (A ⊔ B) (A' ⊔ B')

theorem **toNat_relLength_eq_add_of_le** (h : LengthCommensurable A C) (hCB : C ≤ B) (hBA : B ≤ A) : (relLength A C).toNat = (relLength A B).toNat + (relLength B C).toNat

theorem **signedLength_eq_toNat_sub_of_le** {D : Submodule R V} (h : LengthCommensurable A B) (hAD : LengthCommensurable A D) (hDA : D ≤ A) (hDB : D ≤ B) : signedLength A B = ((relLength A D).toNat : ℤ) − ((relLength B D).toNat : ℤ)

theorem **signedLength_eq_add** (hAB : LengthCommensurable A B) (hBC : LengthCommensurable B C) : signedLength A C = signedLength A B + signedLength B C

theorem **signedLength_add** (hAB : LengthCommensurable A B) (hBC : LengthCommensurable B C) : signedLength A B + signedLength B C = signedLength A C

theorem **signedLength_eq_sub_right** (hAC : LengthCommensurable A C) (hBC : LengthCommensurable B C) : signedLength A B = signedLength A C - signedLength B C

theorem **signedLength_eq_sub_left** (hCA : LengthCommensurable C A) (hCB : LengthCommensurable C B) : signedLength A B = signedLength C B - signedLength C A

theorem **signedLength_eq_toNat_sub_of_ge** {D : Submodule R V} (h : LengthCommensurable A B) (hDA : LengthCommensurable D A) (hAD : A ≤ D) (hBD : B ≤ D) : signedLength A B = ((relLength D B).toNat : ℤ) - ((relLength D A).toNat : ℤ)

theorem **lengthCommensurable_finset_inf'** {ι : Type*} {s : Finset ι} (hs : s.Nonempty) {f : ι → Submodule R V} (h : ∀ i ∈ s, LengthCommensurable A (f i)) : LengthCommensurable A (s.inf' hs f)

theorem **lengthCommensurable_finset_sup'** {ι : Type*} {s : Finset ι} (hs : s.Nonempty) {f : ι → Submodule R V} (h : ∀ i ∈ s, LengthCommensurable A (f i)) : LengthCommensurable A (s.sup' hs f)

theorem **relLength_chain** : (∀ A B C : Submodule R V, C ≤ B → B ≤ A → relLength A C = relLength A B + relLength B C) ∧ (∀ A B C : Submodule R V, relLength A C ≤ relLength A B + relLength B C) ∧ Equivalence (LengthCommensurable : Submodule R V → Submodule R V → Prop) ∧ (∀ A B C : Submodule R V, LengthCommensurable A B → LengthCommensurable A C → LengthCommensurable A (B ⊓ C) ∧ LengthCommensurable A (B ⊔ C)) ∧ (∀ A A' B B' : Submodule R V, LengthCommensurable A A' → LengthCommensurable B B' → LengthCommensurable (A ⊓ B) (A' ⊓ B') ∧ LengthCommensurable (A ⊔ B) (A' ⊔ B')) ∧ (∀ A B C : Submodule R V, LengthCommensurable A B → LengthCommensurable B C → signedLength A C = signedLength A B + signedLength B C)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthCommensurableInfRestrictScalarsOfLeSpan

theorem **exists_ne_zero_smul_mem_of_mem_span_fractionField** {M : Submodule R V} {x : V} (hx : x ∈ Submodule.span K (M : Set V)) : ∃ a : R, a ≠ 0 ∧ a • x ∈ M

theorem **exists_ne_zero_smul_le_of_le_span_fractionField** {M N : Submodule R V} (hN : N.FG) (h : (N : Set V) ⊆ Submodule.span K (M : Set V)) : ∃ a : R, a ≠ 0 ∧ a • N ≤ M

theorem **span_inf_restrictScalars_eq** {M : Submodule R V} {W : Submodule K V} (h : W ≤ Submodule.span K (M : Set V)) : Submodule.span K ((M ⊓ W.restrictScalars R : Submodule R V) : Set V) = W

theorem **fractionField_smul_inf_restrictScalars** {s : K} (hs : s ≠ 0) (0 : Submodule R V) (W : Submodule K V) : s • 0 ⊓ W.restrictScalars R = s • (0 ⊓ W.restrictScalars R)

theorem **isLattice_comap_subtype** {M : Submodule R V} (hM : M.FG) {W : Submodule K V} (h : W ≤ Submodule.span K (M : Set V)) : Submodule.IsLattice K (M.comap (W.subtype.restrictScalars R)) where

theorem **finrank_inf_restrictScalars_of_le_span** {M : Submodule R V} (hM : M.FG) {W : Submodule K V} (h : W ≤ Submodule.span K (M : Set V)) : Module.finrank R ↥(M ⊓ W.restrictScalars R) = Module.finrank K W

theorem **log_ordFrac_units_smul_zpow** {ϖ : R} (hϖ : Irreducible ϖ) (u : Rˣ) (n : ℤ) : WithZero.log (Ring.ordFrac R (u • algebraMap R K ϖ ^ n)) = n

theorem **log_ordFrac_algebraMap** {a : R} (ha : a ≠ 0) : WithZero.log (Ring.ordFrac R (algebraMap R K a)) = (addVal R a).toNat

theorem **relLength_ne_top_of_subset_span** {A B : Submodule R V} (hA : A.FG) (h : (A : Set V) ⊆ Submodule.span K (B : Set V)) : relLength A B ≠ ⊤

theorem **subset_span_of_relLength_ne_top** {A B : Submodule R V} (h : relLength A B ≠ ⊤) : (A : Set V) ⊆ Submodule.span K (B : Set V)

theorem **lengthCommensurable_iff_span_eq** {A B : Submodule R V} (hA : A.FG) (hB : B.FG) : LengthCommensurable A B ↔ Submodule.span K (A : Set V) = Submodule.span K (B : Set V)

theorem **lengthCommensurable_inf_restrictScalars_of_le_span** {M 0 : Submodule R V} (hM : M.FG) (h0 : 0.FG) {W : Submodule K V} (hW : W ≤ Submodule.span K (M : Set V)) (hW' : W ≤ Submodule.span K (0 : Set V)) : LengthCommensurable (M ⊓ W.restrictScalars R) (0 ⊓ W.restrictScalars R)

theorem **signedLength_smul_self_fractionField** {B : Submodule R V} (hB : B.FG) {s : K} (hs : s ≠ 0) : LengthCommensurable B (s • B) ∧ signedLength B (s • B) = WithZero.log (Ring.ordFrac R s) * Module.finrank R B

theorem **LengthCommensurable.signedLength_smul_right** {A B : Submodule R V} (h : LengthCommensurable A B) (hB : B.FG) {s : K} (hs : s ≠ 0) : LengthCommensurable A (s • B) ∧ signedLength A (s • B) = signedLength A B + WithZero.log (Ring.ordFrac R s) * Module.finrank R B

theorem **signedLength_inf_smul_inf_of_le_span** {M 0 : Submodule R V} (hM : M.FG) (h0 : 0.FG) {W : Submodule K V} (hW : W ≤ Submodule.span K (M : Set V)) (hW' : W ≤ Submodule.span K (0 : Set V)) {s : K} (hs : s ≠ 0) : LengthCommensurable (M ⊓ W.restrictScalars R) (0 ⊓ W.restrictScalars R) ∧ LengthCommensurable (M ⊓ W.restrictScalars R) (s • 0 ⊓ W.restrictScalars R) ∧ signedLength (M ⊓ W.restrictScalars R) (s • 0 ⊓ W.restrictScalars R) = signedLength (M ⊓ W.restrictScalars R) (0 ⊓ W.restrictScalars R) + WithZero.log (Ring.ordFrac R s) * Module.finrank K W

theorem **signedLength_inf_smul_inf** (M 0 : Submodule R V) [Submodule.IsLattice K M] [Submodule.IsLattice K 0] (W : Submodule K V) {s : K} (hs : s ≠ 0) : LengthCommensurable (M ⊓ W.restrictScalars R) (0 ⊓ W.restrictScalars R) ∧ LengthCommensurable (M ⊓ W.restrictScalars R) (s • 0 ⊓ W.restrictScalars R) ∧ signedLength (M ⊓ W.restrictScalars R) (s • 0 ⊓ W.restrictScalars R) = signedLength (M ⊓ W.restrictScalars R) (0 ⊓ W.restrictScalars R) + WithZero.log (Ring.ordFrac R s) * Module.finrank K W

theorem **signedLength_inf_smul_inf_zpow** {ϖ : R} (hϖ : Irreducible ϖ) (M 0 : Submodule R V) [Submodule.IsLattice K M] [Submodule.IsLattice K 0] (W : Submodule K V) (u : Rˣ) (n : ℤ) : signedLength (M ⊓ W.restrictScalars R) ((u • algebraMap R K ϖ ^ n) • 0 ⊓ W.restrictScalars R) = signedLength (M ⊓ W.restrictScalars R) (0 ⊓ W.restrictScalars R) + n * Module.finrank K W

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthCommensurableSpanSpanOfDetNeZero

theorem **length_self_eq_top_of_dvr** : Module.length R R = ⊤

theorem **addVal_finset_prod** {ι : Type*} (s : Finset ι) (a : ι → R) : addVal R (∏ i ∈ s, a i) = ∑ i ∈ s, addVal R (a i)

theorem **length_quotient_range_eq_top_of_det_eq_zero** [Module.Free R M] [Module.Finite R M] {f : M →ₗ[R] M} (hf : LinearMap.det f = 0) : Module.length R (M ⧸ LinearMap.range f) = ⊤

theorem **length_quotient_eq_addVal_det_equiv** [Module.Free R M] [Module.Finite R M] (N : Submodule R M) (e : M ≃ₗ[R] N) : Module.length R (M ⧸ N) = addVal R (LinearMap.det (N.subtype ∘ₗ (e : M →ₗ[R] N)))

theorem **length_quotient_range_eq_addVal_det** [Module.Free R M] [Module.Finite R M] (f : M →ₗ[R] M) : Module.length R (M ⧸ LinearMap.range f) = addVal R (LinearMap.det f)

theorem **length_quotient_range_ne_top_iff** [Module.Free R M] [Module.Finite R M] {f : M →ₗ[R] M} : Module.length R (M ⧸ LinearMap.range f) ≠ ⊤ ↔ LinearMap.det f ≠ 0

theorem **length_quotient_range_toLin'_eq_addVal_det** (P : Matrix ι ι R) : Module.length R ((ι → R) ⧸ LinearMap.range (Matrix.toLin' P)) = addVal R P.det

theorem **length_quotient_range_mulVecLin_eq_addVal_det** (P : Matrix ι ι R) : Module.length R ((ι → R) ⧸ LinearMap.range P.mulVecLin) = addVal R P.det

theorem **length_quotient_span_cols_eq_addVal_det** (P : Matrix ι ι R) : Module.length R ((ι → R) ⧸ Submodule.span R (Set.range P.col)) = addVal R P.det

theorem **length_quotient_span_rows_eq_addVal_det** (P : Matrix ι ι R) : Module.length R ((ι → R) ⧸ Submodule.span R (Set.range P.row)) = addVal R P.det

theorem **length_quotient_range_eq_addVal_det_toMatrix** {M' : Type*} [AddCommGroup M'] [Module R M'] (b' : Basis ι R M') (b : Basis ι R M) (f : M' →ₗ[R] M) : Module.length R (M ⧸ LinearMap.range f) = addVal R (LinearMap.toMatrix b' b f).det

theorem **length_quotient_span_eq_addVal_det** (b : Basis ι R M) (v : ι → M) : Module.length R (M ⧸ Submodule.span R (Set.range v)) = addVal R (b.det v)

theorem **length_quotient_span_ne_top_iff** (b : Basis ι R M) {v : ι → M} : Module.length R (M ⧸ Submodule.span R (Set.range v)) ≠ ⊤ ↔ b.det v ≠ 0

theorem **length_quotient_span_eq_top_of_det_eq_zero** (b : Basis ι R M) {v : ι → M} (hv : b.det v = 0) : Module.length R (M ⧸ Submodule.span R (Set.range v)) = ⊤

theorem **length_quotient_eq_addVal_det_basis** (b : Basis ι R M) (N : Submodule R M) (c : Basis ι R N) : Module.length R (M ⧸ N) = addVal R (b.det fun i => (c i : M))

theorem **relLength_span_eq_addVal_det** {A : Submodule R V} (b : Basis ι R A) (v : ι → A) : relLength A (Submodule.span R (Set.range fun i => (v i : V))) = addVal R (b.det v)

theorem **relLength_span_span_eq_addVal_det** {a v : ι → V} (ha : LinearIndependent R a) (P : Matrix ι ι R) (hv : ∀ j, v j = ∑ i, P i j • a i) : relLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range v)) = addVal R P.det

theorem **relLength_map_eq_addVal_det_restrict** {A : Submodule R V} [Module.Free R A] [Module.Finite R A] (g : V →ₗ[R] V) (hg : ∀ x ∈ A, g x ∈ A) : relLength A (A.map g) = addVal R (LinearMap.det (g.restrict hg))

private theorem **relLength_chain** {S V : Type*} [Ring S] [AddCommGroup V] [Module S V] {A B C : Submodule S V} (hAB : B ≤ A) (hBC : C ≤ B) : relLength A C = relLength A B + relLength B C

private theorem **signedLength_of_common** {S V : Type*} [Ring S] [AddCommGroup V] [Module S V] {A B C : Submodule S V} (hA : C ≤ A) (hB : C ≤ B) {m n : ℕ} (hm : relLength A C = m) (hn : relLength B C = n) : LengthCommensurable A B ∧ signedLength A B = (m : ℤ) - n

theorem **ordFrac_algebraMap_eq_exp** {x : R} {n : ℕ} (h : addVal R x = n) : Ring.ordFrac R (algebraMap R K x) = WithZero.exp (n : ℤ)

theorem **ordFrac_unit_mul_zpow** {ϖ : R} (hϖ : Irreducible ϖ) (u : Rˣ) (n : ℤ) : Ring.ordFrac R (algebraMap R K u * algebraMap R K ϖ ^ n) = WithZero.exp n

private theorem **lattice_length_ord_frac_det** {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, b j = ∑ i, g i j • a i) : LengthCommensurable (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) ∧ Ring.ordFrac R g.det = WithZero.exp (signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)))

theorem **lengthCommensurable_span_span_of_det_ne_zero** {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, b j = ∑ i, g i j • a i) : LengthCommensurable (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b))

theorem **ordFrac_det_eq_exp_signedLength_span** {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, b j = ∑ i, g i j • a i) : Ring.ordFrac R g.det = WithZero.exp (signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)))

theorem **signedLength_span_eq_log_ordFrac_det** {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, b j = ∑ i, g i j • a i) : signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) = WithZero.log (Ring.ordFrac R g.det)

theorem **signedLength_span_eq_log_ordFrac_det_generalLinear** {a b : ι → V} (ha : LinearIndependent K a) (g : GL ι K) (hb : ∀ j, b j = ∑ i, g i j • a i) : LengthCommensurable (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) ∧ signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) = WithZero.log (Ring.ordFrac R (g : Matrix ι ι K).det)

theorem **signedLength_span_eq_of_det_eq_unit_mul_zpow** {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hb : ∀ j, b j = ∑ i, g i j • a i) {ϖ : R} (hϖ : Irreducible ϖ) (u : Rˣ) (n : ℤ) (hg : g.det = algebraMap R K u * algebraMap R K ϖ ^ n) : signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) = n

theorem **lengthCommensurable_span_basis** (a b : Basis ι K V) : LengthCommensurable (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b))

theorem **ordFrac_basis_det_eq_exp_signedLength** (a b : Basis ι K V) : Ring.ordFrac R (a.det b) = WithZero.exp (signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)))

theorem **signedLength_span_basis_eq_log_ordFrac_det** (a b : Basis ι K V) : signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) = WithZero.log (Ring.ordFrac R (a.det b))

private theorem **span_range_coe_basis** {A : Submodule R V} (a : Basis ι R A) : Submodule.span R (Set.range fun i => (a i : V)) = A

private theorem **span_range_extendOfIsLattice** {A : Submodule R V} [A.IsLattice K] (a : Basis ι R A) : Submodule.span R (Set.range (a.extendOfIsLattice K)) = A

theorem **signedLength_eq_log_ordFrac_det_of_bases** {A B : Submodule R V} (a : Basis ι R A) (b : Basis ι R B) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, (b j : V) = ∑ i, g i j • (a i : V)) : LengthCommensurable A B ∧ signedLength A B = WithZero.log (Ring.ordFrac R g.det)

theorem **signedLength_map_eq_log_ordFrac_det** (A : Submodule R V) [A.IsLattice K] (g : V →ₗ[K] V) (hg : LinearMap.det g ≠ 0) : LengthCommensurable A (A.map (g.restrictScalars R)) ∧ signedLength A (A.map (g.restrictScalars R)) = WithZero.log (Ring.ordFrac R (LinearMap.det g))

theorem **signedLength_eq_log_ordFrac_det_of_map_eq** {A B : Submodule R V} [A.IsLattice K] (g : V ≃ₗ[K] V) (h : A.map ((g : V →ₗ[K] V).restrictScalars R) = B) : LengthCommensurable A B ∧ signedLength A B = WithZero.log (Ring.ordFrac R (LinearMap.det (g : V →ₗ[K] V)))

theorem **exists_linearEquiv_map_eq_of_isLattice** (A B : Submodule R V) [A.IsLattice K] [B.IsLattice K] : ∃ g : V ≃ₗ[K] V, A.map ((g : V →ₗ[K] V).restrictScalars R) = B

theorem **lengthCommensurable_of_isLattice** (A B : Submodule R V) [A.IsLattice K] [B.IsLattice K] : LengthCommensurable A B

theorem **valuation_det_eq_exp_neg_signedLength** {A B : Submodule R V} [A.IsLattice K] (g : V ≃ₗ[K] V) (h : A.map ((g : V →ₗ[K] V).restrictScalars R) = B) : (IsDiscreteValuationRing.maximalIdeal R).valuation K (LinearMap.det (g : V →ₗ[K] V)) = WithZero.exp (-signedLength A B)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthEqTopOfFinrankNeZero

theorem **length_self_eq_top** : Module.length R R = ⊤

theorem **addVal_prod** {ι : Type*} (s : Finset ι) (a : ι → R) : addVal R (∏ i ∈ s, a i) = ∑ i ∈ s, addVal R (a i)

theorem **length_eq_top_of_finrank_ne_zero** [Module.Finite R M] (h : finrank R M ≠ 0) : Module.length R M = ⊤

theorem **length_coker_eq_addVal_det** [Module.Free R M] [Module.Finite R M] (f : M →ₗ[R] M) : Module.length R (M ⧸ LinearMap.range f) = addVal R (LinearMap.det f)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthLocalizedModuleEqZeroOfNotAnnihilatorLe

theorem **isFiniteLength_of_isTorsion** [IsNoetherianRing R] [Ring.KrullDimLE 1 R] {X : Type*} [AddCommGroup X] [Module R X] [Module.Finite R X] (hX : Module.IsTorsion R X) : IsFiniteLength R X

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthQuotRangeEqAddValDet

theorem **length_quot_range_eq_addVal_det** {M : Type*} [AddCommGroup M] [Module R M] [Module.Free R M] [Module.Finite R M] (f : M →ₗ[R] M) : Module.length R (M ⧸ LinearMap.range f) = addVal R (LinearMap.det f)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthQuotientMaximalIdealPowSmulTop

theorem **nonempty_quotient_smul_top_equiv_pi** {ι : Type*} [Finite ι] (b : Module.Basis ι R M) (a : R) : Nonempty ((M ⧸ (a • ⊤ : Submodule R M)) ≃ₗ[R] (ι → R ⧸ Ideal.span {a}))

theorem **length_quotient_smul_top_of_free** [Nontrivial R] [Module.Free R M] [Module.Finite R M] (a : R) : Module.length R (M ⧸ (a • ⊤ : Submodule R M)) = Module.length R (R ⧸ Ideal.span {a}) * (Module.finrank R M : ℕ∞)

theorem **submoduleOf_smul_self** (A : Submodule R M) (a : R) : (a • A).submoduleOf A = (a • ⊤ : Submodule R A)

theorem **relLength_smul_self_eq_length** (A : Submodule R M) (a : R) : relLength A (a • A) = Module.length R (A ⧸ (a • ⊤ : Submodule R A))

theorem **length_quotient_span_singleton_dvr** (a : R) : Module.length R (R ⧸ Ideal.span {a}) = addVal R a

theorem **addVal_irreducible_pow** {ϖ : R} (hϖ : Irreducible ϖ) (e : ℕ) : addVal R (ϖ ^ e) = e

theorem **length_quotient_smul_top_dvr** (a : R) : Module.length R (M ⧸ (a • ⊤ : Submodule R M)) = addVal R a * (Module.finrank R M : ℕ∞)

theorem **relLength_smul_self** (hA : A.FG) (a : R) : relLength A (a • A) = addVal R a * (Module.finrank R A : ℕ∞)

theorem **relLength_smul_self_of_addVal_eq** (hA : A.FG) {a : R} {n : ℕ} (hn : addVal R a = n) : relLength A (a • A) = ((n * Module.finrank R A : ℕ) : ℕ∞)

theorem **relLength_irreducible_pow_smul_self** (hA : A.FG) {ϖ : R} (hϖ : Irreducible ϖ) (e : ℕ) : relLength A (ϖ ^ e • A) = ((e * Module.finrank R A : ℕ) : ℕ∞)

theorem **relLength_smul_self_ne_top** (hA : A.FG) {a : R} (ha : a ≠ 0) : relLength A (a • A) ≠ ⊤

theorem **lengthCommensurable_smul_self** (hA : A.FG) {a : R} (ha : a ≠ 0) : LengthCommensurable A (a • A)

theorem **signedLength_smul_self_of_addVal_eq** (hA : A.FG) {a : R} {n : ℕ} (hn : addVal R a = n) : signedLength A (a • A) = (n : ℤ) * (Module.finrank R A : ℤ)

theorem **signedLength_smul_self** (hA : A.FG) {a : R} (ha : a ≠ 0) : signedLength A (a • A) = ((addVal R a).toNat : ℤ) * (Module.finrank R A : ℤ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthQuotientSpanSingleton

theorem **addVal_prod** {ι : Type*} (s : Finset ι) (f : ι → R) : addVal R (∏ i ∈ s, f i) = ∑ i ∈ s, addVal R (f i)

theorem **length_quotient_span_singleton** {x : R} (hx : x ≠ 0) : Module.length R (R ⧸ Ideal.span ({x} : Set R)) = addVal R x

theorem **length_quotient_eq_addVal_det** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {M : Type u_2} [AddCommGroup M] [Module R M] [Module.Free R M] [Module.Finite R M] (N : Submodule R M) (e : M ≃ₗ[R] N) : Module.length R (M ⧸ N) = (addVal R : R → ℕ∞) (LinearMap.det (N.subtype ∘ₗ (e : M →ₗ[R] N)))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthRestrictScalarsEqSumInertiaDegOfSubset

theorem **length_restrictScalars_of_isLocalHom** [Module.Finite (ResidueField R) (ResidueField S)] : Module.length R X = Module.length S X * (Module.finrank (ResidueField R) (ResidueField S) : ℕ∞)

theorem **isFiniteLength_restrictScalars_iff_of_isLocalHom** [Module.Finite (ResidueField R) (ResidueField S)] : IsFiniteLength R X ↔ IsFiniteLength S X

theorem **length_eq_length_localizedModule** (T : Submonoid S) (A : Type*) [CommRing A] [Algebra S A] [IsLocalization T A] {X X' : Type*} [AddCommGroup X] [Module S X] [AddCommGroup X'] [Module S X'] [Module A X'] [IsScalarTower S A X'] (f : X →ₗ[S] X') [IsLocalizedModule T f] : Module.length A X' = Module.length (Localization T) (LocalizedModule T X)

theorem **length_localizedModule_eq_add_of_exact** (T : Submonoid S) {N M P : Type*} [AddCommGroup N] [AddCommGroup M] [AddCommGroup P] [Module S N] [Module S M] [Module S P] (f : N →ₗ[S] M) (g : M →ₗ[S] P) (hf : Function.Injective f) (hg : Function.Surjective g) (H : Function.Exact f g) : Module.length (Localization T) (LocalizedModule T M) = Module.length (Localization T) (LocalizedModule T N) + Module.length (Localization T) (LocalizedModule T P)

theorem **length_localizedModule_congr** (T : Submonoid S) {M N : Type*} [AddCommGroup M] [AddCommGroup N] [Module S M] [Module S N] (e : M ≃ₗ[S] N) : Module.length (Localization T) (LocalizedModule T M) = Module.length (Localization T) (LocalizedModule T N)

theorem **length_localizedModule_quotient_self** (m : Ideal S) [hm : m.IsMaximal] : Module.length (Localization.AtPrime m) (LocalizedModule m.primeCompl (S ⧸ m)) = 1

theorem **length_localizedModule_quotient_of_ne** (m n : Ideal S) [hm : m.IsMaximal] [hn : n.IsPrime] (h : n ≠ m) : Module.length (Localization.AtPrime n) (LocalizedModule n.primeCompl (S ⧸ m)) = 0

theorem **length_localizedModule_atPrime_le** (n : Ideal S) [n.IsPrime] {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) : Module.length (Localization.AtPrime n) (LocalizedModule n.primeCompl X) ≤ Module.length S X

theorem **finite_support_length_localizedModule** {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) : {n : MaximalSpectrum S | Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) ≠ 0}.Finite

theorem **length_restrictScalars_eq_sum_of_subset** {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) [iR : Module R X] [iT : IsScalarTower R S X] (T : Finset (MaximalSpectrum S)) (hT : ∀ n ∉ T, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) = 0) : Module.length R X = ∑ n ∈ T, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) * Module.length R (S ⧸ n.asIdeal)

theorem **length_restrictScalars_eq_finsum** {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) [Module R X] [IsScalarTower R S X] : Module.length R X = ∑ᶠ n : MaximalSpectrum S, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) * Module.length R (S ⧸ n.asIdeal)

theorem **length_eq_sum_length_localizedModule** {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) (T : Finset (MaximalSpectrum S)) (hT : ∀ n ∉ T, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) = 0) : Module.length S X = ∑ n ∈ T, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X)

theorem **length_quotient_eq_finrank** (p : Ideal R) (n : Ideal S) [p.IsMaximal] [n.LiesOver p] [Module.Finite (R ⧸ p) (S ⧸ n)] : Module.length R (S ⧸ n) = (Module.finrank (R ⧸ p) (S ⧸ n) : ℕ∞)

theorem **length_quotient_eq_inertiaDeg** (p : Ideal R) (n : Ideal S) [p.IsMaximal] [n.IsMaximal] [n.LiesOver p] [Module.Finite (R ⧸ p) (S ⧸ n)] : Module.length R (S ⧸ n) = (n.inertiaDeg R : ℕ∞)

theorem **length_restrictScalars_eq_sum_inertiaDeg_of_subset** [IsLocalRing R] {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) [Module R X] [IsScalarTower R S X] (T : Finset (MaximalSpectrum S)) (hT : ∀ n ∉ T, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) = 0) (hover : ∀ n ∈ T, n.asIdeal.LiesOver (maximalIdeal R)) (hfin : ∀ n ∈ T, Module.Finite R (S ⧸ n.asIdeal)) : Module.length R X = ∑ n ∈ T, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) * (n.asIdeal.inertiaDeg R : ℕ∞)

theorem **liesOver_maximalIdeal_of_isIntegral** [Algebra.IsIntegral R S] (n : Ideal S) [n.IsMaximal] : n.LiesOver (maximalIdeal R)

theorem **finite_maximalSpectrum_of_finite** [Module.Finite R S] : Finite (MaximalSpectrum S)

theorem **inertiaDeg_eq_finrank_quotient_maximalIdeal** [Module.Finite R S] (n : Ideal S) [n.IsMaximal] : haveI

theorem **length_restrictScalars_eq_sum_inertiaDeg** [Module.Finite R S] [Fintype (MaximalSpectrum S)] {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) [Module R X] [IsScalarTower R S X] : Module.length R X = ∑ n : MaximalSpectrum S, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) * (n.asIdeal.inertiaDeg R : ℕ∞)

theorem **length_restrictScalars_eq_sum_inertiaDeg_of_isLocalizedModule** [Module.Finite R S] [Fintype (MaximalSpectrum S)] {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) [Module R X] [IsScalarTower R S X] (A : MaximalSpectrum S → Type*) [∀ n, CommRing (A n)] [∀ n, Algebra S (A n)] [∀ n, IsLocalization.AtPrime (A n) n.asIdeal] (Y : MaximalSpectrum S → Type*) [∀ n, AddCommGroup (Y n)] [∀ n, Module S (Y n)] [∀ n, Module (A n) (Y n)] [∀ n, IsScalarTower S (A n) (Y n)] (f : ∀ n, X →ₗ[S] Y n) [∀ n, IsLocalizedModule n.asIdeal.primeCompl (f n)] : Module.length R X = ∑ n : MaximalSpectrum S, Module.length (A n) (Y n) * (n.asIdeal.inertiaDeg R : ℕ∞)

theorem **isFiniteLength_restrictScalars_of_finite** [Module.Finite R S] {X : Type*} [AddCommGroup X] [Module S X] (h : IsFiniteLength S X) [Module R X] [IsScalarTower R S X] : IsFiniteLength R X

theorem **length_restrictScalars_local_and_semilocal** (R S X : Type*) [CommRing R] [CommRing S] [Algebra R S] [IsLocalRing R] [AddCommGroup X] [Module R X] [Module S X] [IsScalarTower R S X] (h : IsFiniteLength S X) : (∀ [IsLocalRing S] [IsLocalHom (algebraMap R S)] [Module.Finite (ResidueField R) (ResidueField S)], Module.length R X = Module.length S X * (Module.finrank (ResidueField R) (ResidueField S) : ℕ∞)) ∧ (∀ [Module.Finite R S], Finite (MaximalSpectrum S) ∧ ∀ [Fintype (MaximalSpectrum S)], Module.length R X = ∑ n : MaximalSpectrum S, Module.length (Localization.AtPrime n.asIdeal) (LocalizedModule n.asIdeal.primeCompl X) * (n.asIdeal.inertiaDeg R : ℕ∞))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LengthSelfEqTop

lemma **length_quot_span** (x : R) : Module.length R (R ⧸ Ideal.span {x}) = IsDiscreteValuationRing.addVal R x

lemma **length_self_eq_top** : Module.length R R = ⊤

lemma **addVal_prod** {ι : Type*} (s : Finset ι) (a : ι → R) : IsDiscreteValuationRing.addVal R (∏ i ∈ s, a i) = ∑ i ∈ s, IsDiscreteValuationRing.addVal R (a i)

theorem **main** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {M : Type u_2} [AddCommGroup M] [Module R M] {ι : Type u_3} [Fintype ι] [DecidableEq ι] {M' : Type u_4} [AddCommGroup M'] [Module R M'] (b' : Module.Basis ι R M') (b : Module.Basis ι R M) (f : M' →ₗ[R] M) : Module.length R (M ⧸ LinearMap.range f) = (IsDiscreteValuationRing.addVal R : R → ℕ∞) (Matrix.det ((LinearMap.toMatrix b' b : (M' →ₗ[R] M) → Matrix ι ι R) f))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.LinearIndependentLaurentMonomial

theorem **linearIndependent_laurentMonomial** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (s : Finset ι) (n : ℤ) (v : {a : ι → ℤ // ∑ i, a i = n ∧ ∀ i, i ∉ s → 0 ≤ a i} → LocalizedModule.Away (xProd R s) (MvPolynomial ι R)), (∀ a (k : ℕ) (m : ι →₀ ℕ), (∀ i, (m i : ℤ) = a.1 i + if i ∈ s then (k : ℤ) else 0) → v a = awayFrac (MvPolynomial ι R) (xProd R s) k (monomial m 1)) → LinearIndependent R v

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MapFOneInjectiveOfSectionsSurjective

lemma **freeYonedaHomEquiv_comp** (U : C) {F G : Sheaf J AddCommGrpCat.{v}} (φ : (freeYoneda J).obj U ⟶ F) (f : F ⟶ G) : freeYonedaHomEquiv U G (φ ≫ f) = f.hom.app (op U) (freeYonedaHomEquiv U F φ)

lemma **freeYonedaHomEquiv_apply** (U : C) {F : Sheaf J AddCommGrpCat.{v}} (φ : (freeYoneda J).obj U ⟶ F) : freeYonedaHomEquiv U F φ = φ.hom.app (op U) (tautSection U)

lemma **freeYonedaHomEquiv_map_comp** {U V : C} (i : V ⟶ U) {F : Sheaf J AddCommGrpCat.{v}} (φ : (freeYoneda J).obj U ⟶ F) : freeYonedaHomEquiv V F ((freeYoneda J).map i ≫ φ) = F.obj.map i.op (freeYonedaHomEquiv U F φ)

lemma **map_apply** (f : F ⟶ G) (n : ℕ) (U : C) (x : Ext.{w'} ((freeYoneda J).obj U) F n) : map f n U x = x.comp (Ext.mk₀ f) (add_zero n)

lemma **res_apply** (n : ℕ) {U V : C} (i : V ⟶ U) (x : Ext.{w'} ((freeYoneda J).obj U) F n) : res F n i x = (Ext.mk₀ ((freeYoneda J).map i)).comp x (zero_add n)

@[simp] lemma **map_id** (n : ℕ) (U : C) (x : Sheaf.H'.{w'} F n U) : map (𝟙 F) n U x = x

lemma **map_comp** (f : F ⟶ G) (g : G ⟶ K) (n : ℕ) (U : C) (x : Sheaf.H'.{w'} F n U) : map (f ≫ g) n U x = map g n U (map f n U x)

@[simp] lemma **map_zero_hom** (n : ℕ) (U : C) (x : Sheaf.H'.{w'} F n U) : map (0 : F ⟶ G) n U x = 0

lemma **map_add_hom** (f g : F ⟶ G) (n : ℕ) (U : C) (x : Sheaf.H'.{w'} F n U) : map (f + g) n U x = map f n U x + map g n U x

@[simp] lemma **res_id** (n : ℕ) (U : C) (x : Sheaf.H'.{w'} F n U) : res F n (𝟙 U) x = x

lemma **res_comp** (n : ℕ) {U V W : C} (i : V ⟶ U) (j : W ⟶ V) (x : Sheaf.H'.{w'} F n U) : res F n (j ≫ i) x = res F n j (res F n i x)

lemma **res_map** (f : F ⟶ G) (n : ℕ) {U V : C} (i : V ⟶ U) (x : Sheaf.H'.{w'} F n U) : res G n i (map f n U x) = map f n V (res F n i x)

lemma **equivZero_mk₀** (U : C) (φ : (freeYoneda J).obj U ⟶ F) : equivZero.{w'} F U (Ext.mk₀ φ) = freeYonedaHomEquiv U F φ

lemma **equivZero_map** (f : F ⟶ G) (U : C) (x : Sheaf.H'.{w'} F 0 U) : equivZero G U (map f 0 U x) = f.hom.app (op U) (equivZero F U x)

lemma **equivZero_res** {U V : C} (i : V ⟶ U) (x : Sheaf.H'.{w'} F 0 U) : equivZero F V (res F 0 i x) = F.obj.map i.op (equivZero F U x)

lemma **map_equivZero_symm** (f : F ⟶ G) (U : C) (s : F.obj.obj (op U)) : map f 0 U ((equivZero.{w'} F U).symm s) = (equivZero G U).symm (f.hom.app (op U) s)

lemma **δ_apply** (n : ℕ) (U : C) (x : Ext.{w'} ((freeYoneda J).obj U) S.X₃ n) : δ hS n U x = x.comp hS.extClass rfl

theorem **exact₂** (n : ℕ) (U : C) : Function.Exact (map.{w'} S.f n U) (map.{w'} S.g n U)

theorem **exact₃** (n : ℕ) (U : C) : Function.Exact (map.{w'} S.g n U) (δ.{w'} hS n U)

theorem **exact₁** (n : ℕ) (U : C) : Function.Exact (δ.{w'} hS n U) (map.{w'} S.f (n + 1) U)

theorem **map_zero_injective_of_mono** {F G : Sheaf J AddCommGrpCat.{v}} (f : F ⟶ G) [Mono f] (U : C) : Function.Injective (map.{w'} f 0 U)

theorem **injective_zero** (U : C) : Function.Injective (map.{w'} S.f 0 U)

@[simp] lemma **δ_map** (n : ℕ) (U : C) (x : Sheaf.H'.{w'} S.X₂ n U) : δ hS n U (map S.g n U x) = 0

@[simp] lemma **map_δ** (n : ℕ) (U : C) (x : Sheaf.H'.{w'} S.X₃ n U) : map S.f (n + 1) U (δ hS n U x) = 0

theorem **δ_naturality** {S₁ S₂ : ShortComplex (Sheaf J AddCommGrpCat.{v})} (h₁ : S₁.ShortExact) (h₂ : S₂.ShortExact) (φ : S₁ ⟶ S₂) (n : ℕ) (U : C) (x : Sheaf.H'.{w'} S₁.X₃ n U) : map φ.τ₁ (n + 1) U (δ h₁ n U x) = δ h₂ n U (map φ.τ₃ n U x)

theorem **δ_res** (n : ℕ) {U V : C} (i : V ⟶ U) (x : Sheaf.H'.{w'} S.X₃ n U) : res S.X₁ (n + 1) i (δ hS n U x) = δ hS n V (res S.X₃ n i x)

theorem **δ_equivZero_symm_eq_zero_iff** (U : C) (s : S.X₃.obj.obj (op U)) : δ.{w'} hS 0 U ((equivZero S.X₃ U).symm s) = 0 ↔ ∃ t, S.g.hom.app (op U) t = s

theorem **sections_surjective_of_subsingleton** (U : C) (h : Subsingleton (Sheaf.H'.{w'} S.X₁ 1 U)) : Function.Surjective (S.g.hom.app (op U))

theorem **δ_surjective_of_subsingleton** (n : ℕ) (U : C) (h : Subsingleton (Sheaf.H'.{w'} S.X₂ (n + 1) U)) : Function.Surjective (δ.{w'} hS n U)

theorem **δ_injective_of_subsingleton** (n : ℕ) (U : C) (h : Subsingleton (Sheaf.H'.{w'} S.X₂ n U)) : Function.Injective (δ.{w'} hS n U)

theorem **map_f_injective_of_surjective** (n : ℕ) (U : C) (h : Function.Surjective (map.{w'} S.g n U)) : Function.Injective (map.{w'} S.f (n + 1) U)

theorem **map_f_one_injective_of_sections_surjective** (U : C) (h : Function.Surjective (S.g.hom.app (op U))) : Function.Injective (map.{w'} S.f 1 U)

theorem **subsingleton₂** (n : ℕ) (U : C) (h₁ : Subsingleton (Sheaf.H'.{w'} S.X₁ n U)) (h₃ : Subsingleton (Sheaf.H'.{w'} S.X₃ n U)) : Subsingleton (Sheaf.H'.{w'} S.X₂ n U)

theorem **subsingleton₃** (n : ℕ) (U : C) (h₂ : Subsingleton (Sheaf.H'.{w'} S.X₂ n U)) (h₁ : Subsingleton (Sheaf.H'.{w'} S.X₁ (n + 1) U)) : Subsingleton (Sheaf.H'.{w'} S.X₃ n U)

theorem **subsingleton₁** (n : ℕ) (U : C) (h₃ : Subsingleton (Sheaf.H'.{w'} S.X₃ n U)) (h₂ : Subsingleton (Sheaf.H'.{w'} S.X₂ (n + 1) U)) : Subsingleton (Sheaf.H'.{w'} S.X₁ (n + 1) U)

theorem **subsingleton₁_zero** (U : C) (h₂ : Subsingleton (Sheaf.H'.{w'} S.X₂ 0 U)) : Subsingleton (Sheaf.H'.{w'} S.X₁ 0 U)

theorem **subsingleton₁_one_of_sections_surjective** (U : C) (h : Function.Surjective (S.g.hom.app (op U))) (h₂ : Subsingleton (Sheaf.H'.{w'} S.X₂ 1 U)) : Subsingleton (Sheaf.H'.{w'} S.X₁ 1 U)

theorem **toSheaf_preservesColimitsOfShape** (K : Type*) [Category K] [HasColimitsOfShape K AddCommGrpCat.{v}] : PreservesColimitsOfShape K (SheafOfModules.toSheaf.{v} R) where

theorem **shortExact_map_toSheaf** {S : ShortComplex (SheafOfModules.{v} R)} (hS : S.ShortExact) : (S.map (SheafOfModules.toSheaf.{v} R)).ShortExact

@[simp] lemma **toAbSheaf_obj_obj** (M : X.Modules) : ((toAbSheaf X).obj M).obj = M.presheaf

@[simp] lemma **toAbSheaf_map_hom** {M N : X.Modules} (φ : M ⟶ N) : ((toAbSheaf X).map φ).hom = φ.mapPresheaf

theorem **shortExact_map_toAbSheaf** {S : ShortComplex X.Modules} (hS : S.ShortExact) : (S.map (toAbSheaf X)).ShortExact

theorem **exact_map_toAbSheaf_iff** (S : ShortComplex X.Modules) : (S.map (toAbSheaf X)).Exact ↔ S.Exact

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MapSpanBaseChangeEquivSubmoduleOf

theorem **baseChange_map** (f : V →ₗ[R] V') (A : Submodule R V) : (A.map f).baseChange S = (A.baseChange S).map (f.baseChange S)

theorem **baseChange_range** (f : V →ₗ[R] V') : (LinearMap.range f).baseChange S = LinearMap.range (f.baseChange S)

theorem **ker_baseChange_mkQ** (B : Submodule R V) : LinearMap.ker (B.mkQ.baseChange S) = B.baseChange S

theorem **baseChangeQuotEquiv_tmul** (B : Submodule R V) (s : S) (x : V) : baseChangeQuotEquiv S B (s ⊗ₜ Submodule.Quotient.mk x) = Submodule.Quotient.mk (s ⊗ₜ x)

theorem **baseChange_ker** (f : V →ₗ[R] V') : (LinearMap.ker f).baseChange S = LinearMap.ker (f.baseChange S)

theorem **baseChange_comap** (f : V →ₗ[R] V') (B : Submodule R V') : (B.comap f).baseChange S = (B.baseChange S).comap (f.baseChange S)

theorem **baseChange_inf** (A B : Submodule R V) : (A ⊓ B).baseChange S = A.baseChange S ⊓ B.baseChange S

theorem **baseChange_iInf** {ι : Type*} [Finite ι] (A : ι → Submodule R V) : (⨅ i, A i).baseChange S = ⨅ i, (A i).baseChange S

theorem **baseChange_injective** {f : V →ₗ[R] V'} (hf : Function.Injective f) : Function.Injective (f.baseChange S)

theorem **coe_baseChangeEquiv_apply** (A : Submodule R V) (x : S ⊗[R] A) : (baseChangeEquiv S A x : S ⊗[R] V) = A.subtype.baseChange S x

theorem **map_baseChangeEquiv_submoduleOf** (A B : Submodule R V) : ((B.submoduleOf A).baseChange S).map (baseChangeEquiv S A : S ⊗[R] A →ₗ[S] A.baseChange S) = (B.baseChange S).submoduleOf (A.baseChange S)

theorem **baseChangeSubquotEquiv_tmul** (A B : Submodule R V) (s : S) (x : A) : baseChangeSubquotEquiv S A B (s ⊗ₜ Submodule.Quotient.mk x) = Submodule.Quotient.mk (s ⊗ₜ (x : V), Submodule.tmul_mem_baseChange_of_mem s x.2)

theorem **lift_rank_baseChange.{u, v, w}** {R : Type u} (S : Type v) [CommRing R] [CommRing S] [Algebra R S] [Module.Flat R S] [NoZeroDivisors S] [FaithfulSMul R S] {V : Type w} [AddCommGroup V] [Module R V] (A : Submodule R V) : Cardinal.lift.{w} (Module.rank S (A.baseChange S)) = Cardinal.lift.{max v w} (Module.rank R A)

theorem **finrank_baseChange** [NoZeroDivisors S] [FaithfulSMul R S] (A : Submodule R V) : Module.finrank S (A.baseChange S) = Module.finrank R A

theorem **baseChangeToField_tmul** (s : S) (v : V) : baseChangeToField R K S L V (s ⊗ₜ v) = algebraMap S L s ⊗ₜ[K] v

theorem **baseChangeToField_naturality** (f : V →ₗ[K] V') (x : S ⊗[R] V) : f.baseChange L (baseChangeToField R K S L V x) = baseChangeToField R K S L V' ((f.restrictScalars R).baseChange S x)

theorem **spanBaseChange_eq_map** (M : Submodule R V) : spanBaseChange K S L M = (M.baseChange S).map (baseChangeToField R K S L V)

theorem **spanBaseChange_map** [LinearMap.CompatibleSMul V V' R K] (f : V →ₗ[K] V') (M : Submodule R V) : spanBaseChange K S L (M.map (f.restrictScalars R)) = (spanBaseChange K S L M).map ((f.baseChange L).restrictScalars S)

theorem **baseChangeToField_injective** : Function.Injective (baseChangeToField R K S L V)

theorem **comap_baseChangeToField** (W : Submodule K V) : ((W.baseChange L).restrictScalars S).comap (baseChangeToField R K S L V) = (W.restrictScalars R).baseChange S

theorem **spanBaseChange_inf_subspace** (M : Submodule R V) (W : Submodule K V) : spanBaseChange K S L (M ⊓ W.restrictScalars R) = spanBaseChange K S L M ⊓ (W.baseChange L).restrictScalars S

theorem **coe_spanBaseChangeEquiv_apply** (M : Submodule R V) (x : S ⊗[R] M) : (spanBaseChangeEquiv K S L M x : L ⊗[K] V) = baseChangeToField R K S L V (M.subtype.baseChange S x)

theorem **map_spanBaseChangeEquiv_submoduleOf** (M N : Submodule R V) : ((N.submoduleOf M).baseChange S).map (spanBaseChangeEquiv K S L M : S ⊗[R] M →ₗ[S] spanBaseChange K S L M) = (spanBaseChange K S L N).submoduleOf (spanBaseChange K S L M)

theorem **finrank_spanBaseChange** [NoZeroDivisors S] [FaithfulSMul R S] (M : Submodule R V) : Module.finrank S (spanBaseChange K S L M) = Module.finrank R M

theorem **lift_rank_spanBaseChange.{u, v}** {R : Type*} (K S : Type*) (L : Type u) [CommRing R] [Field K] [CommRing S] [CommRing L] [Algebra R K] [Algebra K L] [Algebra S L] {V : Type v} [AddCommGroup V] [Module K V] [Module R V] [IsScalarTower R K V] [Algebra R S] [Algebra R L] [IsScalarTower R K L] [IsScalarTower R S L] [IsFractionRing R K] [FaithfulSMul S L] [Module.Flat R S] [NoZeroDivisors S] [FaithfulSMul R S] (M : Submodule R V) : Cardinal.lift.{v} (Module.rank S (spanBaseChange K S L M)) = Cardinal.lift.{max u v} (Module.rank R M)

theorem **flat_spanBaseChange_inf_subspace.{u, v}** {R : Type*} (K S : Type*) (L : Type u) [CommRing R] [IsDomain R] [CommRing S] [IsDomain S] [Field K] [Field L] [Algebra R S] [FaithfulSMul R S] [Module.Flat R S] [Algebra R K] [IsFractionRing R K] [Algebra S L] [IsFractionRing S L] [Algebra K L] [Algebra R L] [IsScalarTower R K L] [IsScalarTower R S L] {V : Type v} [AddCommGroup V] [Module K V] [Module R V] [IsScalarTower R K V] (M : Submodule R V) (W : Submodule K V) : spanBaseChange K S L (M ⊓ W.restrictScalars R) = spanBaseChange K S L M ⊓ (W.baseChange L).restrictScalars S ∧ Cardinal.lift.{v} (Module.rank S (spanBaseChange K S L M)) = Cardinal.lift.{max u v} (Module.rank R M) ∧ Cardinal.lift.{v} (Module.rank L (W.baseChange L)) = Cardinal.lift.{max u v} (Module.rank K W) ∧ Module.finrank S (spanBaseChange K S L M) = Module.finrank R M ∧ Module.finrank L (W.baseChange L) = Module.finrank

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MemNonzeroLocusIffMemBasicOpenChart

theorem **mem_nonzeroLocus_iff_mem_basicOpen_chart** {Y : Scheme.{0}} {H : LB Y} (s : LB.Sec H) (k : H.ι) {x : Y} (hk : x ∈ H.U k) : x ∈ nonzeroLocus s ↔ x ∈ Y.basicOpen (s.f k)

theorem **mem_basicOpen_finset_prod_iff** {Y : Scheme.{0}} {U : Y.Opens} {ι : Type} (f : ι → Γ(Y, U)) (I : Finset ι) {x : Y} (hx : x ∈ U) : x ∈ Y.basicOpen (∏ i ∈ I, f i) ↔ ∀ i ∈ I, x ∈ Y.basicOpen (f i)

theorem **mem_finset_biInf_opens_iff** {Y : Scheme.{0}} {ι : Type} (D : ι → Y.Opens) (I : Finset ι) {x : Y} : x ∈ (⊓ i ∈ I, D i) ↔ ∀ i ∈ I, x ∈ D i

theorem **exists_prod_sec_nonzeroLocus** : ∀ (Y : Scheme.{0}) (H : LB Y) (ι : Type) (s : ι → LB.Sec H) (I : Finset ι), H.IsLB → ∃ σ : LB.Sec (pw H (I.card : ℤ)), (∀ l : H.ι, σ.f l = ∏ i ∈ I, (s i).f l) ∧ nonzeroLocus σ = ⊓ i ∈ I, nonzeroLocus (s i)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MemRangeOfForallHeightEqOneOfIsReflexive

theorem **mem_minimalPrimes_span_singleton_of_height_eq_one** {p : Ideal A} [p.IsPrime] (hp : p.height = 1) {a : A} (ha : a ≠ 0) (hap : a ∈ p) : p ∈ (Ideal.span {a}).minimalPrimes

theorem **finite_height_eq_one_mem** {a : A} (ha : a ≠ 0) : {p : PrimeSpectrum A | p.asIdeal.height = 1 ∧ a ∈ p.asIdeal}.Finite

theorem **krullDimLE_one_of_height_eq_one** (p : Ideal A) [p.IsPrime] (hp : p.height = 1) (B : Type*) [CommRing B] [Algebra A B] [IsLocalization.AtPrime B p] : Ring.KrullDimLE 1 B

theorem **isDiscreteValuationRing_of_height_eq_one** (p : Ideal A) [p.IsPrime] (hp : p.height = 1) (B : Type*) [CommRing B] [IsDomain B] [Algebra A B] [IsLocalization.AtPrime B p] : IsDiscreteValuationRing B

theorem **height_eq_one_of_forall_mem_iff_dvd** {p : Ideal A} [hp : p.IsPrime] {a b : A} (ha : a ≠ 0) (h : ∀ r, r ∈ p ↔ a ∣ r * b) : p.height = 1

theorem **height_eq_one_of_mem_associatedPrimes** {a : A} (ha : a ≠ 0) {p : Ideal A} (hp : p ∈ associatedPrimes A (A ⧸ Ideal.span {a})) : p.height = 1

theorem **mem_associatedPrimes_iff_height_eq_one** {a : A} (ha : a ≠ 0) {p : Ideal A} : p ∈ associatedPrimes A (A ⧸ Ideal.span {a}) ↔ p.IsPrime ∧ p.height = 1 ∧ a ∈ p

theorem **dvd_of_forall_height_eq_one** {a b : A} (ha : a ≠ 0) (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → a ∈ p → ∃ s, s ∉ p ∧ a ∣ s * b) : a ∣ b

theorem **exists_smul_eq_of_forall_dual_dvd_of_projective** [Module.Projective A M] {a : A} {m : M} (h : ∀ φ : Module.Dual A M, a ∣ φ m) : ∃ m', a • m' = m

theorem **exists_smul_eq_of_forall_dual_dvd_of_isReflexive** [IsDomain A] [Module.IsReflexive A M] {a : A} (ha : a ≠ 0) {m : M} (h : ∀ φ : Module.Dual A M, a ∣ φ m) : ∃ m', a • m' = m

theorem **mem_range_of_forall_height_eq_one_of_dual** (hM : ∀ {a : A}, a ≠ 0 → ∀ {m : M}, (∀ φ : Module.Dual A M, a ∣ φ m) → ∃ m', a • m' = m) (f : M →ₗ[A] V) (hf : Function.Injective f) [Module.IsTorsionFree A V] {v : V} {a : A} (ha : a ≠ 0) (hav : a • v ∈ LinearMap.range f) (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → a ∈ p → ∃ s, s ∉ p ∧ s • v ∈ LinearMap.range f) : v ∈ LinearMap.range f

theorem **mem_range_of_forall_height_eq_one_of_projective** [Module.Projective A M] (f : M →ₗ[A] V) (hf : Function.Injective f) [Module.IsTorsionFree A V] {v : V} {a : A} (ha : a ≠ 0) (hav : a • v ∈ LinearMap.range f) (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → a ∈ p → ∃ s, s ∉ p ∧ s • v ∈ LinearMap.range f) : v ∈ LinearMap.range f

theorem **mem_range_of_forall_height_eq_one_of_isReflexive** [Module.IsReflexive A M] (f : M →ₗ[A] V) (hf : Function.Injective f) [Module.IsTorsionFree A V] {v : V} {a : A} (ha : a ≠ 0) (hav : a • v ∈ LinearMap.range f) (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → a ∈ p → ∃ s, s ∉ p ∧ s • v ∈ LinearMap.range f) : v ∈ LinearMap.range f

theorem **isTorsionFree_of_isLocalizedModule** (f : M →ₗ[A] V) [IsLocalizedModule A⁰ f] : Module.IsTorsionFree A V

theorem **injective_of_isLocalizedModule_nonZeroDivisors** [Module.IsTorsionFree A M] (f : M →ₗ[A] V) [IsLocalizedModule A⁰ f] : Function.Injective f

theorem **mem_range_iff_forall_height_eq_one_of_projective** [Module.Projective A M] (f : M →ₗ[A] V) [IsLocalizedModule A⁰ f] (v : V) : v ∈ LinearMap.range f ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → ∃ s, s ∉ p ∧ s • v ∈ LinearMap.range f

theorem **mem_range_iff_forall_height_eq_one_of_isReflexive** [Module.IsReflexive A M] (f : M →ₗ[A] V) [IsLocalizedModule A⁰ f] (v : V) : v ∈ LinearMap.range f ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → ∃ s, s ∉ p ∧ s • v ∈ LinearMap.range f

theorem **le_of_forall_height_eq_one_of_projective** [Module.IsTorsionFree A V] {N N' : Submodule A V} [Module.Projective A N'] (h₀ : ∀ v ∈ N, ∃ a : A, a ≠ 0 ∧ a • v ∈ N') (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → ∀ v ∈ N, ∃ s, s ∉ p ∧ s • v ∈ N') : N ≤ N'

theorem **le_of_forall_height_eq_one_of_isReflexive** [Module.IsTorsionFree A V] {N N' : Submodule A V} [Module.IsReflexive A N'] (h₀ : ∀ v ∈ N, ∃ a : A, a ≠ 0 ∧ a • v ∈ N') (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → ∀ v ∈ N, ∃ s, s ∉ p ∧ s • v ∈ N') : N ≤ N'

theorem **height_eq_one_of_mem_associatedPrimes_coker_of_dual** (hM : ∀ {a : A}, a ≠ 0 → ∀ {m : M}, (∀ φ : Module.Dual A M, a ∣ φ m) → ∃ m', a • m' = m) (f : M →ₗ[A] V) (hf : Function.Injective f) [Module.IsTorsionFree A V] (ht : ∀ v : V, ∃ a : A, a ≠ 0 ∧ a • v ∈ LinearMap.range f) {p : Ideal A} (hp : p ∈ associatedPrimes A (V ⧸ LinearMap.range f)) : p.height = 1

theorem **height_eq_one_of_mem_associatedPrimes_coker_of_projective** [Module.Projective A M] (f : M →ₗ[A] V) (hf : Function.Injective f) [Module.IsTorsionFree A V] (ht : ∀ v : V, ∃ a : A, a ≠ 0 ∧ a • v ∈ LinearMap.range f) {p : Ideal A} (hp : p ∈ associatedPrimes A (V ⧸ LinearMap.range f)) : p.height = 1

theorem **height_eq_one_of_mem_associatedPrimes_coker_of_isReflexive** [Module.IsReflexive A M] (f : M →ₗ[A] V) (hf : Function.Injective f) [Module.IsTorsionFree A V] (ht : ∀ v : V, ∃ a : A, a ≠ 0 ∧ a • v ∈ LinearMap.range f) {p : Ideal A} (hp : p ∈ associatedPrimes A (V ⧸ LinearMap.range f)) : p.height = 1

theorem **mem_range_rTensor_iff** {M : Type*} [AddCommGroup M] [Module A M] (v : K ⊗[A] M) : v ∈ LinearMap.range ((IsScalarTower.toAlgHom A B K).toLinearMap.rTensor M) ↔ ∃ s, s ∉ p ∧ s • v ∈ LinearMap.range (TensorProduct.mk A K M 1)

theorem **mem_range_algebraMap_atPrime_iff** (x : K) : x ∈ (algebraMap B K).range ↔ ∃ s, s ∉ p ∧ s • x ∈ (algebraMap A K).range

theorem **mem_primeLocalSubalgebra_iff** (p : Ideal A) [p.IsPrime] (x : K) : x ∈ primeLocalSubalgebra K p ↔ ∃ s, s ∉ p ∧ s • x ∈ (algebraMap A K).range

theorem **mem_range_algebraMap_iff_mem_primeLocalSubalgebra** (p : Ideal A) [p.IsPrime] (B : Type*) [CommRing B] [Algebra A B] [IsLocalization.AtPrime B p] [Algebra B K] [IsScalarTower A B K] (x : K) : x ∈ (algebraMap B K).range ↔ x ∈ primeLocalSubalgebra K p

theorem **ord_map_ringEquiv** {B : Type*} [CommRing B] (e : A ≃+* B) (a : A) : Ring.ord B (e a) = Ring.ord A a

theorem **ordFrac_eq_heightOneOrdHom** (p : Ideal A) [p.IsPrime] (hp : p.height = 1) (B : Type*) [CommRing B] [Nontrivial B] [Algebra A B] [IsLocalization.AtPrime B p] [IsNoetherianRing B] [Ring.KrullDimLE 1 B] [Algebra B K] [IsScalarTower A B K] [IsFractionRing B K] : Ring.ordFrac B = heightOneOrdHom K p hp

theorem **heightOneOrdHom_eq_exp** {x : K} (hx : x ≠ 0) : heightOneOrdHom K p hp x = WithZero.exp (heightOneOrd_vw p hp x)

theorem **heightOneOrd_vw_mul** {x y : K} (hx : x ≠ 0) (hy : y ≠ 0) : heightOneOrd_vw p hp (x * y) = heightOneOrd_vw p hp x + heightOneOrd_vw p hp y

theorem **heightOneOrd_vw_inv** (x : K) : heightOneOrd_vw p hp x⁻¹ = -heightOneOrd_vw p hp x

theorem **heightOneOrd_vw_div** {x y : K} (hx : x ≠ 0) (hy : y ≠ 0) : heightOneOrd_vw p hp (x / y) = heightOneOrd_vw p hp x - heightOneOrd_vw p hp y

theorem **heightOneOrd_vw_zpow** (x : K) (n : ℤ) : heightOneOrd_vw p hp (x ^ n) = n * heightOneOrd_vw p hp x

theorem **heightOneOrd_vw_algebraMap_of_notMem** {a : A} (ha : a ∉ p) : heightOneOrd_vw p hp (algebraMap A K a) = 0

theorem **heightOneOrd_vw_nonneg_of_mem** {x : K} (hx : x ≠ 0) (h : x ∈ primeLocalSubalgebra K p) : 0 ≤ heightOneOrd_vw p hp x

theorem **finite_heightOneOrd_vw_ne_zero** {x : K} (hx : x ≠ 0) : {p : PrimeSpectrum A | ∃ hp : p.asIdeal.height = 1, heightOneOrd_vw p.asIdeal hp x ≠ 0}.Finite

theorem **mem_range_algebraMap_of_forall_height_eq_one_mem** {x : K} {a : A} (ha : a ≠ 0) (hax : a • x ∈ (algebraMap A K).range) (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → a ∈ p → x ∈ primeLocalSubalgebra K p) : x ∈ (algebraMap A K).range

theorem **mem_range_algebraMap_of_pow_smul_mem** {x : K} {a : A} (ha : a ≠ 0) {n : ℕ} (hax : a ^ n • x ∈ (algebraMap A K).range) (h : ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → a ∈ p → x ∈ primeLocalSubalgebra K p) : x ∈ (algebraMap A K).range

theorem **mem_range_algebraMap_iff_forall_height_eq_one** (x : K) : x ∈ (algebraMap A K).range ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → x ∈ primeLocalSubalgebra K p

theorem **iInf_primeLocalSubalgebra_height_eq_one** : ⨅ p : {p : PrimeSpectrum A // p.asIdeal.height = 1}, primeLocalSubalgebra K p.1.asIdeal = ⊥

theorem **mem_range_tensor_iff_forall_height_eq_one** {M : Type*} [AddCommGroup M] [Module A M] [Module.Projective A M] (v : K ⊗[A] M) : v ∈ LinearMap.range (TensorProduct.mk A K M 1) ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → v ∈ LinearMap.range ((primeLocalSubalgebra K p).val.toLinearMap.rTensor M)

theorem **mem_range_tensor_iff_forall_height_eq_one_of_isReflexive** {M : Type*} [AddCommGroup M] [Module A M] [Module.IsReflexive A M] (v : K ⊗[A] M) : v ∈ LinearMap.range (TensorProduct.mk A K M 1) ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → v ∈ LinearMap.range ((primeLocalSubalgebra K p).val.toLinearMap.rTensor M)

theorem **heightOneOrdHom_eq_valuation_inv** (B : Type*) [CommRing B] [IsDomain B] [IsDiscreteValuationRing B] [Algebra A B] [IsLocalization.AtPrime B p] [Algebra B K] [IsScalarTower A B K] [IsFractionRing B K] (x : K) : heightOneOrdHom K p hp x = ((IsDiscreteValuationRing.maximalIdeal B).valuation K x)⁻¹

theorem **heightOneOrd_vw_of_irreducible** {ϖ : primeLocalSubalgebra K p} (hϖ : Irreducible ϖ) : heightOneOrd_vw p hp (ϖ : K) = 1

theorem **exists_heightOneOrd_vw_eq** (n : ℤ) : ∃ x : K, x ≠ 0 ∧ heightOneOrd_vw p hp x = n

theorem **mem_primeLocalSubalgebra_iff_heightOneOrd_vw_nonneg** {x : K} (hx : x ≠ 0) : x ∈ primeLocalSubalgebra K p ↔ 0 ≤ heightOneOrd_vw p hp x

theorem **heightOneOrd_vw_algebraMap_pos_iff** {a : A} (ha : a ≠ 0) : 0 < heightOneOrd_vw p hp (algebraMap A K a) ↔ a ∈ p

theorem **min_le_heightOneOrd_vw_add** {x y : K} (hx : x ≠ 0) (hy : y ≠ 0) (hxy : x + y ≠ 0) : min (heightOneOrd_vw p hp x) (heightOneOrd_vw p hp y) ≤ heightOneOrd_vw p hp (x + y)

theorem **mem_range_algebraMap_iff_heightOneOrd_vw_nonneg** {x : K} (hx : x ≠ 0) : x ∈ (algebraMap A K).range ↔ ∀ (p : Ideal A) [p.IsPrime] (hp : p.height = 1), 0 ≤ heightOneOrd_vw p hp x

theorem **exists_smul_eq_iff_heightOneOrd_vw_le** {x y : K} (hx : x ≠ 0) (hy : y ≠ 0) : (∃ a : A, a • y = x) ↔ ∀ (p : Ideal A) [p.IsPrime] (hp : p.height = 1), heightOneOrd_vw p hp y ≤ heightOneOrd_vw p hp x

theorem **exists_unit_smul_eq_iff_heightOneOrd_vw_eq** {x y : K} (hx : x ≠ 0) (hy : y ≠ 0) : (∃ u : Aˣ, u • y = x) ↔ ∀ (p : Ideal A) [p.IsPrime] (hp : p.height = 1), heightOneOrd_vw p hp x = heightOneOrd_vw p hp y

theorem **normal_domain_height_one.{u, v, w}** (A : Type u) [CommRing A] [IsDomain A] [IsNoetherianRing A] [IsIntegrallyClosed A] (K : Type v) [Field K] [Algebra A K] [IsFractionRing A K] : (∀ (p : Ideal A) [p.IsPrime] (hp : p.height = 1), IsDiscreteValuationRing (Localization.AtPrime p) ∧ ∃ _ : IsDiscreteValuationRing (primeLocalSubalgebra K p), ∀ x : K, x ≠ 0 → WithZero.exp (-heightOneOrd_vw p hp x) = (IsDiscreteValuationRing.maximalIdeal (primeLocalSubalgebra K p)).valuation K x) ∧ (∀ x : K, x ∈ (algebraMap A K).range ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → x ∈ primeLocalSubalgebra K p) ∧ (∀ (M : Type w) [AddCommGroup M] [Module A M] [Module.Projective A M], Function.Injective (TensorProduct.mk A K M 1) ∧ ∀ v : K ⊗[A] M, v ∈ LinearMap.range (TensorProduct.mk A K M 1) ↔ ∀ (p : Ideal A) [p.IsPrime], p.height = 1 → v ∈ LinearMap.range ((primeLocalSubalgebra K p).val.toLinearMap.rTensor

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MemSaturationOfPowSmulMem

theorem **le_saturation** (g : A) (K : Submodule A M) : K ≤ saturation g K

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MemSpanOfSaturatingOfLe

theorem **mem_span_of_saturating_of_le** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] [IsNoetherianRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒩 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩] (r : ℕ) (d : Fin r → ℤ) (g : Fin r → M), (∀ l, g l ∈ 𝒩 (d l)) → (∀ (i : ι) (y : M), ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y ∈ Submodule.span (MvPolynomial ι R) (Set.range g)) → (∀ y : M, (∀ i : ι, ∃ k : ℕ, (X i : MvPolynomial ι R) ^ k • y = 0) → y = 0) → ∃ n₀ : ℤ, ∀ n, n₀ ≤ n → ∀ y ∈ 𝒩 n, y ∈ Submodule.span (MvPolynomial ι R) (Set.range g)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MonomialCountAndBinomialEstimates

theorem **setOf_degree_eq_coe_finsuppAntidiag** (σ : Type*) [Fintype σ] [DecidableEq σ] (a : ℕ) : {α : σ →₀ ℕ | α.degree = a} = ↑(finsuppAntidiag (univ : Finset σ) a)

theorem **ncard_degree_eq** (σ : Type*) [Fintype σ] (a : ℕ) : {α : σ →₀ ℕ | α.degree = a}.ncard = (a + Fintype.card σ - 1).choose a

theorem **ncard_degree_le** (σ : Type*) [Fintype σ] (k : ℕ) : {α : σ →₀ ℕ | α.degree ≤ k}.ncard = (k + Fintype.card σ).choose (Fintype.card σ)

theorem **natCard_degree_eq** (σ : Type*) [Fintype σ] (a : ℕ) : Nat.card {α : σ →₀ ℕ // α.degree = a} = (a + Fintype.card σ - 1).choose a

theorem **natCard_degree_le** (σ : Type*) [Fintype σ] (k : ℕ) : Nat.card {α : σ →₀ ℕ // α.degree ≤ k} = (k + Fintype.card σ).choose (Fintype.card σ)

theorem **fintypeCard_degree_le** (σ : Type*) [Fintype σ] (k : ℕ) [Fintype {α : σ →₀ ℕ // α.degree ≤ k}] : Fintype.card {α : σ →₀ ℕ // α.degree ≤ k} = (k + Fintype.card σ).choose (Fintype.card σ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MulPullOfSymm

theorem **iso_of_hasGen** {X : Scheme.{0}} {L : LB X} {F : X ⟶ X} {m : ℤ} (h : HasGen L (K

theorem **mulPull_of_symm** : ∀ (K : Type) [Field K] [IsAlgClosed K] (B : Over (Spec (CommRingCat.of K))) [GrpObj B], IsAbelianScheme B → ∀ H : LB B.left, H.IsLB → (∃ w : LB.Iso H (H.pull (GrpObj.inv : B ⟶ B).left), w.IsGen) → ∀ n : ℕ, ∃ w : LB.Iso (pw H ((n ^ 2 : ℕ) : ℤ)) (H.pull (mulA B n).left), w.IsGen

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MumfordBundleCechCohomologyFinrank

theorem **exists_cover_point** (X : Scheme.{0}) [CompactSpace X] (o : X) (ho : IsClosed {o}) : ∃ (r : ℕ) (V : Fin (r + 1) → X.Opens), (∀ j, IsAffineOpen (V j)) ∧ ⊔ j, V j = ⊤ ∧ o ∈ V 0 ∧ ∀ j, j ≠ 0 → o ∉ V j

theorem **isAffineOpen_preimage_inf** {X Y : Scheme.{0}} [X.IsSeparated] (g : Y ⟶ X) {D : X.Opens} (hD : IsAffineOpen D) {V : Y.Opens} (hV : IsAffineOpen V) : IsAffineOpen (g ⁻¹ᵁ D ⊓ V)

theorem **acyclic_cechLB_of_acyclic** {R R' : Type} [CommRing R] [CommRing R'] {Y : Scheme.{0}} (f : Y ⟶ Spec (CommRingCat.of R)) (f' : Y ⟶ Spec (CommRingCat.of R')) (M : LB Y) {κ : Type} [LinearOrder κ] (D : κ → Y.Opens) (h : (cechLB f M D).Acyclic) : (cechLB f' M D).Acyclic

theorem **nonempty_cech_sliceFst_iso** {R : Type} [CommRing R] {Y : Scheme.{0}} (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) {κ₁ κ₂ : Type} [LinearOrder κ₁] [LinearOrder κ₂] (U : κ₁ → Y.Opens) (W : κ₂ → Y.Opens) (T : Finset κ₂) (O : Y.Opens) (hO : O = ⊓ j ∈ T, W j) : Nonempty (cechLB f M (fun i => U i ⊓ O) ≅ cechComplex (Cube.sliceFst (secCube f M fun u : κ₁ ⊕ₗ κ₂ => Sum.elim U W (ofLex u)) T))

theorem **over_affineOpen** {V : P.A.left.Opens} (hV : IsAffineOpen V) {κ : Type} [LinearOrder κ] [Fintype κ] (D : κ → P.A.left.Opens) (hD : ∀ i, IsAffineOpen (D i)) (hc : ⊔ i, D i = ⊤) (M : LB (P.A ⊗ P.A).left) (hM : M.IsLB) : (∀ i, IsAffineOpen ((snd P.A P.A).left ⁻¹ᵁ D i ⊓ (fst P.A P.A).left ⁻¹ᵁ V)) ∧ ∃ (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec Γ(P.A.left, V)) (E : κ → Y.Opens), IsPullback π f (fst P.A P.A).left hV.fromSpec ∧ (∀ i, IsAffineOpen (E i)) ∧ ⊔ i, E i = ⊤ ∧ Nonempty (cechLB (f ≫ hV.fromSpec ≫ P.A.hom) (M.pull π) E ≅ cechLB (P.A ⊗ P.A).hom M fun i => (snd P.A P.A).left ⁻¹ᵁ D i ⊓ (fst P.A P.A).left ⁻¹ᵁ V)

theorem **finrank_cechLB_translDiff** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → (P.A ⊗ P.A).left.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → ∀ n, Module.finrank K ((cechLB (P.A ⊗ P.A).hom (translDiff P.A P.L (𝟙 P.A)) D).homology n) = if n = P.g then 1 else 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MumfordBundleCechCohomologyFinrankStatement

theorem **_root_.GlobalTheory.T.mumford_bundle_cech_cohomology_finrank** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → (P.A ⊗ P.A).left.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → ∀ n, Module.finrank K ((cechLB (P.A ⊗ P.A).hom (translDiff P.A P.L (𝟙 P.A)) D).homology n) = if n = P.g then 1 else 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MumfordBundleCechFiniteVanishesOffOrigin

theorem **exists_iso_pull_pull** {X Y Z : Scheme.{0}} {L : LB X} (hL : L.IsLB) (a : Y ⟶ X) (b : Z ⟶ Y) : ∃ w : LB.Iso ((L.pull a).pull b) (L.pull (b ≫ a)), w.IsGen

theorem **isAffineOpen_biInf_of_isSeparated** {R : Type} [CommRing R] {Y : Scheme.{0}} (f : Y ⟶ Spec (CommRingCat.of R)) [IsSeparated f] {κ : Type} {D : κ → Y.Opens} (hD : ∀ i, IsAffineOpen (D i)) (s : Finset κ) (hs : s.Nonempty) : IsAffineOpen (⊓ i ∈ s, D i)

theorem **cohomology_finite_and_vanishes_off_origin** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of R)), IsPullback π f (fst P.A P.A).left t → IsOpenImmersion t → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → IsNoetherianRing R ∧ (∀ p, Module.Flat R ((cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).X p)) ∧ (∀ p, Module.Finite R ((cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology p)) ∧ ∀ χ : R →+* K, (Spec.map (CommRingCat.ofHom χ) ≫ t) ≫ P.A.hom = 𝟙 _ → Spec.map (CommRingCat.ofHom χ) ≫ t ≠ η[P.A].left → ∀ p, ∃ r : R, χ r ≠ 0 ∧ ∀ z : (cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology p, r • z = 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MumfordBundleCechFiniteVanishesOffOriginStatement

theorem **_root_.GlobalTheory.T.mumfordBundle_cech_finite_vanishes_off_origin** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of R)), IsPullback π f (fst P.A P.A).left t → IsOpenImmersion t → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → IsNoetherianRing R ∧ (∀ p, Module.Flat R ((cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).X p)) ∧ (∀ p, Module.Finite R ((cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology p)) ∧ ∀ χ : R →+* K, (Spec.map (CommRingCat.ofHom χ) ≫ t) ≫ P.A.hom = 𝟙 _ → Spec.map (CommRingCat.ofHom χ) ≫ t ≠ η[P.A].left → ∀ p, ∃ r : R, χ r ≠ 0 ∧ ∀ z : (cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).homology p, r • z = 0

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MumfordBundleFibreCechAcyclicOffOrigin

theorem **lbIso_pull_translDiff** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : LB A.left} (hL : L.IsLB) {T : Over S} (p : T ⟶ A ⊗ A) (x : 𝟙_ (Over S) ⟶ A) (h : p ≫ fst A A = toUnit T ≫ x) : Geo.Tor.LBIso ((LB.LinComb L (K

theorem **fibre_acyclic** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (t : Spec (CommRingCat.of K) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of K)), IsPullback π f (fst P.A P.A).left t → t ≫ P.A.hom = 𝟙 _ → t ≠ η[P.A].left → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).Acyclic

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.MumfordBundleFibreCechAcyclicOffOriginStatement

theorem **_root_.GlobalTheory.T.mumfordBundle_fibre_cech_acyclic_off_origin** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (t : Spec (CommRingCat.of K) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of K)), IsPullback π f (fst P.A P.A).left t → t ≫ P.A.hom = 𝟙 _ → t ≠ η[P.A].left → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).Acyclic

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.NatCardAlgHomEqFinrankOfIsReducedOfIsFractionRin

theorem **discr_eq_det_toMatrix_sq_mul** (e : Basis ι R S) (v : ι → S) : Algebra.discr R v = (e.toMatrix v).det ^ 2 * Algebra.discr R e

theorem **discr_eq_basis_det_sq_mul** (e : Basis ι R S) (v : ι → S) : Algebra.discr R v = e.det v ^ 2 * Algebra.discr R e

theorem **basis_det_mul_left** (e : Basis ι R S) (u : S) (v : ι → S) : e.det (fun i => u * v i) = Algebra.norm R u * e.det v

theorem **discr_mul_left_of_basis** (e : Basis ι R S) (u : S) (v : ι → S) : Algebra.discr R (fun i => u * v i) = Algebra.norm R u ^ 2 * Algebra.discr R v

theorem **addVal_discr_eq_two_mul_length_add** (e : Basis ι R S) (v : ι → S) : addVal R (Algebra.discr R v) = 2 * Module.length R (S ⧸ Submodule.span R (Set.range v)) + addVal R (Algebra.discr R e)

theorem **two_mul_length_le_addVal_discr** (e : Basis ι R S) (v : ι → S) : 2 * Module.length R (S ⧸ Submodule.span R (Set.range v)) ≤ addVal R (Algebra.discr R v)

theorem **two_mul_length_le_addVal_discr_of_card** [Module.Free R S] [Module.Finite R S] (h : Fintype.card ι = finrank R S) (v : ι → S) : 2 * Module.length R (S ⧸ Submodule.span R (Set.range v)) ≤ addVal R (Algebra.discr R v)

theorem **length_quotient_span_ne_top_of_discr_ne_zero** (e : Basis ι R S) {v : ι → S} (hv : Algebra.discr R v ≠ 0) : Module.length R (S ⧸ Submodule.span R (Set.range v)) ≠ ⊤

theorem **addVal_discr_eq_two_mul_length_add_of_equiv** {M : Type*} [AddCommGroup M] [Module R M] (φ : M ≃ₗ[R] S) (e : Basis ι R S) (v : ι → M) : addVal R (Algebra.discr R fun i => φ (v i)) = 2 * Module.length R (M ⧸ Submodule.span R (Set.range v)) + addVal R (Algebra.discr R e)

theorem **two_mul_length_le_addVal_discr_of_equiv** {M : Type*} [AddCommGroup M] [Module R M] (φ : M ≃ₗ[R] S) (e : Basis ι R S) (v : ι → M) : 2 * Module.length R (M ⧸ Submodule.span R (Set.range v)) ≤ addVal R (Algebra.discr R fun i => φ (v i))

theorem **det_algHom_basis_ne_zero** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) : (Matrix.of fun i j => P j (e i)).det ≠ 0

theorem **algHom_matrix_mul_leftMulMatrix** (e : Basis ι R S) (P : ι → (S →ₐ[R] L)) (x : S) : (Matrix.of fun j i => P j (e i)) * (Algebra.leftMulMatrix e x).map (algebraMap R L) = Matrix.diagonal (fun j => P j x) * Matrix.of fun j i => P j (e i)

theorem **map_charpoly_leftMulMatrix_eq_prod_algHom** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (x : S) : (Algebra.leftMulMatrix e x).charpoly.map (algebraMap R L) = ∏ j, (X - C (P j x))

theorem **algebraMap_norm_eq_prod_algHom_of_basis** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (x : S) : algebraMap R L (Algebra.norm R x) = ∏ j, P j x

theorem **algebraMap_trace_eq_sum_algHom_of_basis** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (x : S) : algebraMap R L (Algebra.trace R S x) = ∑ j, P j x

theorem **map_traceMatrix_eq_mul_transpose_of_basis** {κ : Type*} (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (v : κ → S) : (Algebra.traceMatrix R v).map (algebraMap R L) = (Matrix.of fun i j => P j (v i)) * (Matrix.of fun i j => P j (v i)).transpose

theorem **algebraMap_discr_eq_det_sq_of_basis** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (v : ι → S) : algebraMap R L (Algebra.discr R v) = (Matrix.of fun i j => P j (v i)).det ^ 2

theorem **algebraMap_discr_basis_ne_zero** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) : algebraMap R L (Algebra.discr R e) ≠ 0

theorem **algebraMap_norm_eq_prod_algHom_of_card** (h : Fintype.card ι = finrank R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (x : S) : algebraMap R L (Algebra.norm R x) = ∏ j, P j x

theorem **algebraMap_trace_eq_sum_algHom_of_card** (h : Fintype.card ι = finrank R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (x : S) : algebraMap R L (Algebra.trace R S x) = ∑ j, P j x

theorem **algebraMap_discr_eq_det_sq_of_card** (h : Fintype.card ι = finrank R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (v : ι → S) : algebraMap R L (Algebra.discr R v) = (Matrix.of fun i j => P j (v i)).det ^ 2

theorem **etale_of_isReduced_of_perfectField** (K S : Type*) [Field K] [PerfectField K] [CommRing S] [Algebra K S] [Module.Finite K S] [IsReduced S] : Algebra.Etale K S

theorem **natCard_algHom_eq_finrank_of_etale_baseChange** [Module.Free R S] [Module.Finite R S] [IsSepClosed L] [Algebra.FormallyEtale L (L ⊗[R] S)] : Nat.card (S →ₐ[R] L) = finrank R S

theorem **natCard_algHom_eq_finrank_of_isReduced_baseChange** [Module.Free R S] [Module.Finite R S] [IsAlgClosed L] [IsReduced (L ⊗[R] S)] : Nat.card (S →ₐ[R] L) = finrank R S

theorem **natCard_algHom_eq_finrank_of_etale_tower** [Module.Free R S] [Module.Finite R S] (K : Type*) [Field K] [Algebra R K] [Algebra K L] [IsScalarTower R K L] [IsSepClosed L] [Algebra.FormallyEtale K (K ⊗[R] S)] : Nat.card (S →ₐ[R] L) = finrank R S

theorem **natCard_algHom_eq_finrank_of_isReduced_of_isFractionRing** [Module.Free R S] [Module.Finite R S] [IsReduced S] (K : Type*) [Field K] [PerfectField K] [Algebra R K] [IsFractionRing R K] [Algebra K L] [IsScalarTower R K L] [IsSepClosed L] : Nat.card (S →ₐ[R] L) = finrank R S

theorem **natCard_algHom_eq_finrank_of_etale** : Nat.card (S →ₐ[K] L) = finrank K S

theorem **nonempty_equiv_algHom_of_etale** {ι : Type*} [Fintype ι] (h : Fintype.card ι = finrank K S) : Nonempty (ι ≃ (S →ₐ[K] L))

theorem **algebraMap_norm_eq_prod_algHom** [Fintype (S →ₐ[K] L)] (x : S) : algebraMap K L (Algebra.norm K x) = ∏ P : S →ₐ[K] L, P x

theorem **algebraMap_trace_eq_sum_algHom** [Fintype (S →ₐ[K] L)] (x : S) : algebraMap K L (Algebra.trace K S x) = ∑ P : S →ₐ[K] L, P x

theorem **card_eq_finrank_of_equiv_algHom** (P : ι ≃ (S →ₐ[K] L)) : Fintype.card ι = finrank K S

theorem **algebraMap_norm_eq_prod_of_equiv_algHom** (P : ι ≃ (S →ₐ[K] L)) (x : S) : algebraMap K L (Algebra.norm K x) = ∏ j, P j x

theorem **algebraMap_trace_eq_sum_of_equiv_algHom** (P : ι ≃ (S →ₐ[K] L)) (x : S) : algebraMap K L (Algebra.trace K S x) = ∑ j, P j x

theorem **algebraMap_discr_eq_det_sq_of_equiv_algHom** [DecidableEq ι] (P : ι ≃ (S →ₐ[K] L)) (v : ι → S) : algebraMap K L (Algebra.discr K v) = (Matrix.of fun i j => P j (v i)).det ^ 2

theorem **discr_basis_ne_zero_of_etale** [DecidableEq ι] (e : Basis ι K S) : Algebra.discr K e ≠ 0

theorem **discr_norm_trace_of_etale** [DecidableEq ι] (P : ι ≃ (S →ₐ[K] L)) : Fintype.card ι = finrank K S ∧ (∀ v : ι → S, algebraMap K L (Algebra.discr K v) = (Matrix.of fun i j => P j (v i)).det ^ 2) ∧ (∀ x : S, algebraMap K L (Algebra.norm K x) = ∏ j, P j x) ∧ (∀ x : S, algebraMap K L (Algebra.trace K S x) = ∑ j, P j x)

theorem **discr_norm_trace_of_isReduced** (P : ι ≃ (S →ₐ[K] L)) : Fintype.card ι = finrank K S ∧ (∀ v : ι → S, algebraMap K L (Algebra.discr K v) = (Matrix.of fun i j => P j (v i)).det ^ 2) ∧ (∀ x : S, algebraMap K L (Algebra.norm K x) = ∏ j, P j x) ∧ (∀ x : S, algebraMap K L (Algebra.trace K S x) = ∑ j, P j x)

theorem **det_algHom_sq_eq_discr_and_addVal** (e : Basis ι R S) {P : ι → (S →ₐ[R] L)} (hP : Function.Injective P) (v : ι → S) : (Matrix.of fun i j => P j (v i)).det ^ 2 = algebraMap R L (Algebra.discr R v) ∧ Algebra.discr R v = (e.toMatrix v).det ^ 2 * Algebra.discr R e ∧ Algebra.discr R e ≠ 0 ∧ addVal R (Algebra.discr R v) = 2 * Module.length R (S ⧸ Submodule.span R (Set.range v)) + addVal R (Algebra.discr R e) ∧ 2 * Module.length R (S ⧸ Submodule.span R (Set.range v)) ≤ addVal R (Algebra.discr R v)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.NonemptyBaseChangeGrCubeIso

theorem **map_xProd_pow_smul_of_X** {ι R₀ R : Type} [CommRing R₀] [CommRing R] {N₀ N : Type} [AddCommGroup N₀] [Module (MvPolynomial ι R₀) N₀] [AddCommGroup N] [Module (MvPolynomial ι R) N] (ψ : N₀ → N) (hX : ∀ (i : ι) y, ψ ((X i : MvPolynomial ι R₀) • y) = (X i : MvPolynomial ι R) • ψ y) (s : Finset ι) (k : ℕ) (y : N₀) : ψ (xProd R₀ s ^ k • y) = xProd R s ^ k • ψ y

theorem **nonempty_baseChange_grCube_iso** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R₀ R : Type) [CommRing R₀] [CommRing R] [Algebra R₀ R] (N₀ : Type) [AddCommGroup N₀] [Module R₀ N₀] [Module (MvPolynomial ι R₀) N₀] [IsScalarTower R₀ (MvPolynomial ι R₀) N₀] (𝒩₀ : ℤ → Submodule R₀ N₀) [PolyGradedSMul ι 𝒩₀] [Decomposition 𝒩₀] (N : Type) [AddCommGroup N] [Module R N] [Module (MvPolynomial ι R) N] [IsScalarTower R (MvPolynomial ι R) N] (𝒩 : ℤ → Submodule R N) [PolyGradedSMul ι 𝒩] [Decomposition 𝒩] [Module R₀ N] [IsScalarTower R₀ R N] [IsScalarTower R₀ (MvPolynomial ι R) N] (ψ : N₀ →ₗ[R₀] N) (n : ℤ), IsBaseChange R ψ → (∀ (i : ι) y, ψ ((X i : MvPolynomial ι R₀) • y) = (X i : MvPolynomial ι R) • ψ y) → (∀ j, ∀ y ∈ 𝒩₀ j, ψ y ∈ 𝒩 j) → Nonempty (Cube.baseChange R (grCube ι 𝒩₀ n) ≅ grCube ι 𝒩 n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.NonemptyBasisPolyGrading

theorem **topCech_surjective** (A : Cube.{0} ι R) {N : ℕ} (hN : Fintype.card ι = N + 1) : Function.Surjective (topCech A hN)

theorem **topCech_zero_bijective** (A : Cube.{0} ι R) (hN : Fintype.card ι = 0 + 1) : Function.Bijective (topCech A hN)

theorem **topCech_eq_zero_iff** (A : Cube.{0} ι R) {N : ℕ} (hN : Fintype.card ι = N + 1 + 1) (x : A.obj Finset.univ) : topCech A hN x = 0 ↔ ∃ y, Cube.d A (N + 1) y = CubeTop.topTerm A hN x

theorem **exists_basis_top** {κ : Type} (T : κ → Finset ι) (A : Cube.{0} ι R) (b : ∀ s : Finset ι, Module.Basis {a : κ // T a ⊆ s} R (A.obj s)) (hb : ∀ (s t : Finset ι) (h : s ⊆ t) (a : {a : κ // T a ⊆ s}), Cube.res A h (b s a) = b t (a.1, a.2.trans h)) {N : ℕ} (hN : Fintype.card ι = N + 1 + 1) : ∃ c : Module.Basis {a : κ // T a = Finset.univ} R ((cechComplex A).homology (N + 1)), ∀ a, c a = topCech A hN (b Finset.univ (a.1, Finset.subset_univ _))

theorem **card_natVec** (N : ℕ) (hN : 1 ≤ N) (m : ℤ) : Nat.card {a : Fin (N + 1) → ℕ // ∑ i, (a i : ℤ) = m} = (m + N).toNat.choose N

theorem **card_negExp** (N : ℕ) (hN : 1 ≤ N) (n : ℤ) : Nat.card (NegExp N n) = (-n - 1).toNat.choose N

theorem **isEmpty_negExp** (N : ℕ) {n : ℤ} (hn : -(N : ℤ) ≤ n) : IsEmpty (NegExp N n)

theorem **choose_eq_add** (N : ℕ) (hN : 1 ≤ N) (n : ℤ) : Ring.choose (n + N) N = ((n + N).toNat.choose N : ℤ) + (-1) ^ N * (-n - 1).toNat.choose N

theorem **nonempty_basis_polyGrading** : Nonempty (Module.Basis {a : Fin (N + 1) → ℕ // ∑ i, (a i : ℤ) = n} R (polyGrading (Fin (N + 1)) R n))

theorem **cube_exactAt** {p : ℕ} (hp : p ≠ N + 1) : (cubeComplex (grCube (Fin (N + 1)) (polyGrading (Fin (N + 1)) R) n)).ExactAt p

theorem **grAug_bijective** (hN : 1 ≤ N) : Function.Bijective (grAug (Fin (N + 1)) (polyGrading (Fin (N + 1)) R) n)

theorem **isZero_grH_poly** {p : ℕ} (h0 : p ≠ 0) (hp : p ≠ N) : IsZero (grH (Fin (N + 1)) (polyGrading (Fin (N + 1)) R) n p)

theorem **nonempty_basis_grH_top** (hN : 1 ≤ N) : Nonempty (Module.Basis (NegExp N n) R (grH (Fin (N + 1)) (polyGrading (Fin (N + 1)) R) n N))

theorem **nonempty_grH_zero_equiv** : Nonempty (grH (Fin (0 + 1)) (polyGrading (Fin (0 + 1)) R) n 0 ≃ₗ[R] R)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.NonemptyHomLocSingletonEquivQuotSMulTop

theorem **nonempty_homLoc_singleton_equiv_quotSMulTop** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (M : Type) [AddCommGroup M] [Module R M] [Module (MvPolynomial ι R) M] [IsScalarTower R (MvPolynomial ι R) M] (𝒜 : ℤ → Submodule R M) [PolyGradedSMul ι 𝒜] [Decomposition 𝒜] (i : ι), Nonempty (homLoc 𝒜 {i} 0 ≃ₗ[R] QuotSMulTop (X i - 1 : MvPolynomial ι R) M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.NonemptyPerfectSectionsComplexOfNoetherian

theorem **nonempty_perf_of_noetherian** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R], Flat f → IsProjBy f → ∀ M : LB Y, M.IsLB → Nonempty (Perf f M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.NonemptyPerfectSectionsComplexOfNoetherianStatement

theorem **_root_.GlobalTheory.T.nonempty_perfectSectionsComplex_of_noetherian** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R], Flat f → IsProjBy f → ∀ M : LB Y, M.IsLB → Nonempty (Perf f M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.OneLeFinrankSecM

theorem **wchart_le_tw** (N t : ℤ) (j : Fin 1 → L.ι) : LB.wchart L (fun _ : Fin 1 => π) j ≤ (tw (L.pull π) (pw (L.pull π) N) t).U ((fun _ => j 0), (fun _ => j 0))

theorem **lcg_tw** (N t : ℤ) (j j' : Fin 1 → L.ι) : LB.lcg L (K

theorem **exists_sec_ne_zero** [IsIntegral Y] (hL : L.IsLB) (N t : ℤ) (σ : LB.Sec (tw (L.pull π) (pw (L.pull π) N) t)) (i : (tw (L.pull π) (pw (L.pull π) N) t).ι) (hi : σ.f i ≠ 0) : ∃ τ : LB.Sec (LB.LinComb L (K

theorem **exists_sq_eq** (K : Type) [Field K] (N : ℕ) (hN : 1 ≤ N) (t₀ : ℤ) : ∃ (n : ℕ) (t : ℤ), (n : K) ≠ 0 ∧ t₀ ≤ t ∧ ((n ^ 2 * 1 : ℕ) : ℤ) = 1 + N * t

theorem **one_le_finrank_secM** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), 1 ≤ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K))))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.OneLeFinrankSecMStatement

theorem **_root_.GlobalTheory.T.one_le_finrank_secM_one_of_isAlgClosed** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))), 1 ≤ Module.finrank Γ((baseOver (Spec (CommRingCat.of K))).left, ⊤) (SecM P.A P.L 1 (baseOver (Spec (CommRingCat.of K))))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.OneSubSmulPeirceCut

@[simp] theorem **peirceProjection_apply** (e : R) (he : IsIdempotentElem e) (x : A) : peirceCut A e he x = e • x + (1 - e) • (1 : A)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.OrdFracEqValuationInvOfIsLocalization

theorem **faithfulSMul_of_tower** : FaithfulSMul R A

theorem **ker_liftBaseChange_eq_of_finrank_le** [Fintype ι] [DecidableEq ι] (c : V →ₗ[K] (ι → L)) {W₀ : Submodule K V} (hW : W₀ ≤ LinearMap.ker c) [FiniteDimensional K (V ⧸ W₀)] (hr : finrank K (V ⧸ W₀) ≤ Fintype.card ι) {θ : ι → V} (hdet : (Matrix.of fun i j => c (θ j) i).det ≠ 0) : LinearMap.ker (c.liftBaseChange L) = W₀.baseChange L

private theorem **valuation_algebraMap_le_one** (a : A) : P.valuation L (algebraMap A L a) ≤ 1

private theorem **valuation_algebraMap_unit** (u : Aˣ) : P.valuation L (algebraMap A L u) = 1

private theorem **valuation_algebraMap_irreducible** {ϖ : A} (hϖ : Irreducible ϖ) : P.valuation L (algebraMap A L ϖ) = WithZero.exp (-1)

theorem **ordFrac_eq_valuation_inv_of_isLocalization** (x : L) : Ring.ordFrac A x = (P.valuation L x)⁻¹

theorem **log_ordFrac_eq_neg_log_valuation** (x : L) : WithZero.log (Ring.ordFrac A x) = -WithZero.log (P.valuation L x)

private theorem **exists_algebra_localization** (P : HeightOneSpectrum S) : ∃ _ : Algebra (Localization.AtPrime P.asIdeal) L, IsScalarTower S (Localization.AtPrime P.asIdeal) L ∧ IsScalarTower R (Localization.AtPrime P.asIdeal) L ∧ IsFractionRing (Localization.AtPrime P.asIdeal) L

private theorem **dvr_localization** (P : HeightOneSpectrum S) : IsDiscreteValuationRing (Localization.AtPrime P.asIdeal)

theorem **not_isField_of_finite** : ¬ IsField S

theorem **finite_heightOneSpectrum** : Finite (HeightOneSpectrum S)

theorem **finrank_field_pos** : 0 < finrank K L

private theorem **log_ordFrac_norm_algebraMap** {x : S} (hx : x ≠ 0) : WithZero.log (Ring.ordFrac R (Algebra.norm K (algebraMap S L x))) = ∑ P : HeightOneSpectrum S, (P.asIdeal.inertiaDeg R : ℤ) * -WithZero.log (P.valuation L (algebraMap S L x))

theorem **log_ordFrac_norm_eq_sum** {x : L} (hx : x ≠ 0) : WithZero.log (Ring.ordFrac R (Algebra.norm K x)) = ∑ P : HeightOneSpectrum S, (P.asIdeal.inertiaDeg R : ℤ) * -WithZero.log (P.valuation L x)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.PredMulSumPadicValNatFactorialAddCardLe

@[simp] theorem **opWord_nil** (D : Λ → B) : opWord D [] = 1

@[simp] theorem **opWord_cons** (D : Λ → B) (l : Λ) (w : List Λ) : opWord D (l :: w) = D l * opWord D w

theorem **opWord_append** (D : Λ → B) (w w' : List Λ) : opWord D (w ++ w') = opWord D w * opWord D w'

theorem **opWord_perm** {D : Λ → B} (hD : ∀ l l', Commute (D l) (D l')) {w w' : List Λ} (h : w.Perm w') : opWord D w = opWord D w'

theorem **isOrderingRule_toList** : IsOrderingRule (fun α : Λ →₀ ℕ => (Finsupp.toMultiset α).toList)

theorem **IsOrderingRule.length_eq** {ord : (Λ →₀ ℕ) → List Λ} (h : IsOrderingRule ord) (α : Λ →₀ ℕ) : (ord α).length = degree α

theorem **opPoly_monomial** (D : Λ → B) (ord : (Λ →₀ ℕ) → List Λ) (α : Λ →₀ ℕ) (c : R) : opPoly D ord (MvPolynomial.monomial α c) = c • opWord D (ord α)

theorem **opPoly_mul_of_commute** {D : Λ → B} (hD : ∀ l l', Commute (D l) (D l')) {ord : (Λ →₀ ℕ) → List Λ} (h : IsOrderingRule ord) (P Q : MvPolynomial Λ R) : opPoly D ord (P * Q) = opPoly D ord P * opPoly D ord Q

theorem **IsOrderingRule.ord_zero** {ord : (Λ →₀ ℕ) → List Λ} (h : IsOrderingRule ord) : ord 0 = []

theorem **IsOrderingRule.ord_single** {ord : (Λ →₀ ℕ) → List Λ} (h : IsOrderingRule ord) (l : Λ) : ord (single l 1) = [l]

theorem **opPoly_C** (D : Λ → B) {ord : (Λ →₀ ℕ) → List Λ} (h : IsOrderingRule ord) (c : R) : opPoly D ord (MvPolynomial.C c) = algebraMap R B c

theorem **opPoly_X** (D : Λ → B) {ord : (Λ →₀ ℕ) → List Λ} (h : IsOrderingRule ord) (l : Λ) : opPoly D ord (MvPolynomial.X l : MvPolynomial Λ R) = D l

@[simp] theorem **commEval_X** (D : Λ → B) (hD : ∀ l l', Commute (D l) (D l')) (l : Λ) : commEval D hD (MvPolynomial.X l : MvPolynomial Λ R) = D l

@[simp] theorem **constDeriv_X** (v : σ → R) (i : σ) : constDeriv v (X i) = C (v i)

theorem **constDeriv_single** [DecidableEq σ] (i : σ) : constDeriv (Pi.single i (1 : R)) = pderiv i

theorem **constDeriv_commute** (v v' : σ → R) : Commute ((constDeriv v).toLinearMap : Module.End R (MvPolynomial σ R)) (constDeriv v').toLinearMap

theorem **pderiv_commute** (i j : σ) : Commute ((pderiv i).toLinearMap : Module.End R (MvPolynomial σ R)) (pderiv j).toLinearMap

@[simp] theorem **pderivEval_X** (i : σ) : pderivEval (X i : MvPolynomial σ R) = (pderiv i).toLinearMap

@[simp] theorem **constDerivEval_X** (v : Λ → σ → R) (l : Λ) : constDerivEval v (X l) = (constDeriv (v l)).toLinearMap

theorem **X_mem_germIdeal** (i : σ) : (X i : MvPowerSeries σ R) ∈ germIdeal σ R

theorem **exists_eq_sum_X_mul** [Fintype σ] {n : ℕ} {g : MvPowerSeries σ R} (hg : ∀ d : σ →₀ ℕ, degree d < n + 1 → coeff d g = 0) : ∃ h : σ → MvPowerSeries σ R, (∀ i (d : σ →₀ ℕ), degree d < n → coeff d (h i) = 0) ∧ g = ∑ i, X i * h i

theorem **germIdeal_eq_span** [Finite σ] : germIdeal σ R = Ideal.span (Set.range (X : σ → MvPowerSeries σ R))

theorem **mem_germIdeal_pow_iff** [Finite σ] {n : ℕ} {g : MvPowerSeries σ R} : g ∈ germIdeal σ R ^ n ↔ ∀ d : σ →₀ ℕ, degree d < n → coeff d g = 0

theorem **mem_germIdeal_pow_iff_le_order** [Finite σ] {n : ℕ} {g : MvPowerSeries σ R} : g ∈ germIdeal σ R ^ n ↔ (n : ℕ∞) ≤ g.order

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.PresheafCubeExactAtSuccOfSurjective

theorem **cechAbAug_apply** (P : (Opens T)ᵒᵖ ⥤ AddCommGrpCat.{v}) (V : Opens T) (U : ι → Opens T) (x : P.obj (op V)) (s : {s : Finset ι // s.card = 1}) : cechAbAug P V U x s = P.map (homOfLE (cechOpen_le V U s.1)).op x

private theorem **cubeComplex_exactAt_one_of_hom_surj_empty_inj_nonempty** {A B : _root_.GlobalTheory.Cube ι R} (φ : A ⟶ B) (hsurj : ∀ s : Finset ι, s = ∅ → Function.Surjective (φ.app s)) (hinj : ∀ s : Finset ι, s.Nonempty → Function.Injective (φ.app s)) (hB : (_root_.GlobalTheory.cubeComplex B).ExactAt 1) : (_root_.GlobalTheory.cubeComplex A).ExactAt 1

private theorem **cubeComplex_presheafCube_exactAt_one** (V : Opens T) (U : ι → Opens T) : (_root_.GlobalTheory.cubeComplex (_root_.GlobalTheory.presheafCube P V U)).ExactAt 1

private theorem **cubeComplex_presheafCube_exactAt_add_one_of_card** (n : ℕ) : ∀ {α : Type} [LinearOrder α] [Fintype α], Fintype.card α = n → ∀ (V : Opens T) (U : α → Opens T) (p : ℕ), (_root_.GlobalTheory.cubeComplex (_root_.GlobalTheory.presheafCube P V U)).ExactAt (p + 1)

theorem **_root_.GlobalTheory.subsingleton_cechAbH_succ_of_isFlasque** (V : Opens T) (U : ι → Opens T) (p : ℕ) : Subsingleton (_root_.GlobalTheory.cechAbH F.obj V U (p + 1))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ProjectiveOfIsLocallyConstantFinrankFiber

theorem **baseChange_surjective_iff** (S : Type*) [CommRing S] [Algebra R S] (φ : M →ₗ[R] N) : Surjective (φ.baseChange S) ↔ Subsingleton (S ⧸ LinearMap.range φ)

theorem **isOpen_setOf_baseChange_surjective** [Module.Finite R N] (φ : M →ₗ[R] N) : IsOpen {p : PrimeSpectrum R | Surjective (φ.baseChange p.asIdeal.ResidueField)}

theorem **exists_baseChange_surjective** [Module.Finite R M] (k : Type*) [Field k] [Algebra R k] : ∃ φ : (Fin (finrank k (k ⊗[R] M)) → R) →ₗ[R] M, Surjective (φ.baseChange k)

theorem **finrank_baseChange_fin** (k : Type*) [Field k] [Algebra R k] (n : ℕ) : finrank k (k ⊗[R] (Fin n → R)) = n

theorem **eventually_finrank_fiber_le** [Module.Finite R M] (p : PrimeSpectrum R) : ∀ᶠ q : PrimeSpectrum R in 𝒩 p, finrank q.asIdeal.ResidueField (q.asIdeal.ResidueField ⊗[R] M) ≤ finrank p.asIdeal.ResidueField (p.asIdeal.ResidueField ⊗[R] M)

theorem **upperSemicontinuous_finrank_fiber** [Module.Finite R M] : UpperSemicontinuous fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField (p.asIdeal.ResidueField ⊗[R] M)

theorem **isLocallyConstant_finrank_fiber** [Module.Finite R M] [Module.Projective R M] : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField (p.asIdeal.ResidueField ⊗[R] M)

theorem **mem_freeLocus_of_eventually_finrank_eq** [IsReduced R] [Module.Finite R M] (p : PrimeSpectrum R) (h : ∀ᶠ q : PrimeSpectrum R in 𝒩 p, finrank q.asIdeal.ResidueField (q.asIdeal.ResidueField ⊗[R] M) = finrank p.asIdeal.ResidueField (p.asIdeal.ResidueField ⊗[R] M)) : p ∈ Module.freeLocus R M

theorem **projective_of_isLocallyConstant_finrank_fiber** [IsReduced R] [Module.FinitePresentation R M] (h : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField (p.asIdeal.ResidueField ⊗[R] M)) : Module.Projective R M

theorem **exists_comp_comp_eq_of_projective** (g : M₁ →ₗ[R] M₂) [Module.Projective R M₂] [Module.Projective R (M₂ ⧸ LinearMap.range g)] : ∃ s : M₂ →ₗ[R] M₁, g ∘ₗ s ∘ₗ g = g

theorem **exists_retraction_ker_of_projective** (g : M₁ →ₗ[R] M₂) [Module.Projective R (LinearMap.range g)] : ∃ r : M₁ →ₗ[R] LinearMap.ker g, r ∘ₗ (LinearMap.ker g).subtype = LinearMap.id

theorem **tensorKer_injective_of_retraction** (S N : Type*) [CommRing S] [Algebra R S] [AddCommGroup N] [Module R N] [Module S N] [IsScalarTower R S N] (g : M₁ →ₗ[R] M₂) (r : M₁ →ₗ[R] LinearMap.ker g) (hr : r ∘ₗ (LinearMap.ker g).subtype = LinearMap.id) : Injective

(LinearMap.tensorKer S N g)

theorem **tensorKer_bijective_of_comp_comp_eq** (S N : Type*) [CommRing S] [Algebra R S] [AddCommGroup N] [Module R N] [Module S N] [IsScalarTower R S N] (g : M₁ →ₗ[R] M₂) (s : M₂ →ₗ[R] M₁) (hs : g ∘ₗ s ∘ₗ g = g) : Bijective (LinearMap.tensorKer S N g)

theorem **exists_baseChange_homology_equiv** (S : Type*) [CommRing S] [Algebra R S] (f : M₀ →ₗ[R] M₁) (g : M₁ →ₗ[R] M₂) (hfg : g ∘ₗ f = 0) (s : M₂ →ₗ[R] M₁) (hs : g ∘ₗ s ∘ₗ g = g) : ∃ e : S ⊗[R] (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) ≃ₗ[S] LinearMap.ker (g.baseChange S) ⧸ (LinearMap.range (f.baseChange S)).submoduleOf (LinearMap.ker (g.baseChange S)), ∀ (a : S) (x : LinearMap.ker g), e (a ⊗ₜ Submodule.Quotient.mk x) = Submodule.Quotient.mk (a ⊗ₜ (x : M₁)), by simp⟩

theorem **exists_homology_retract** (f : M₀ →ₗ[R] M₁) (g : M₁ →ₗ[R] M₂) (hfg : g ∘ₗ f = 0) (u : M₁ →ₗ[R] M₀) (hu : f ∘ₗ u ∘ₗ f = f) (s : M₂ →ₗ[R] M₁) (hs : g ∘ₗ s ∘ₗ g = g) : ∃ (ι : (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) →ₗ[R] M₁) (ρ : M₁ →ₗ[R] LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)), ρ ∘ₗ ι = LinearMap.id

theorem **finrank_homology_add_finrank_range** [Module.Finite k V₁] (f : V₀ →ₗ[k] V₁) (g : V₁ →ₗ[k] V₂) (hfg : g ∘ₗ f = 0) : finrank k (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) + finrank k (LinearMap.range f) + finrank k (LinearMap.range g) = finrank k V₁

theorem **finrank_homology_add_finrank** [Module.Finite k V₁] [Module.Finite k V₂] (f : V₀ →ₗ[k] V₁) (g : V₁ →ₗ[k] V₂) (hfg : g ∘ₗ f = 0) : finrank k (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) + finrank k V₂ = finrank k (V₁ ⧸ LinearMap.range f) + finrank k (V₂ ⧸ LinearMap.range g)

theorem **finrank_baseChange_homology_add** (k : Type*) [Field k] [Algebra R k] [Module.Finite R M₁] [Module.Finite R M₂] (f : M₀ →ₗ[R] M₁) (g : M₁ →ₗ[R] M₂) (hfg : g ∘ₗ f = 0) : finrank k (LinearMap.ker (g.baseChange k) ⧸ (LinearMap.range (f.baseChange k)).submoduleOf (LinearMap.ker (g.baseChange k))) + finrank k (k ⊗[R] M₂) = finrank k (k ⊗[R] (M₁ ⧸ LinearMap.range f)) + finrank k (k ⊗[R] (M₂ ⧸ LinearMap.range g))

theorem **upperSemicontinuous_finrank_linearMap_homology** [Module.Finite R M₁] [Module.Finite R M₂] [Module.Projective R M₂] (f : M₀ →ₗ[R] M₁) (g : M₁ →ₗ[R] M₂) (hfg : g ∘ₗ f = 0) : UpperSemicontinuous fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField (LinearMap.ker (g.baseChange p.asIdeal.ResidueField) ⧸ (LinearMap.range (f.baseChange p.asIdeal.ResidueField)).submoduleOf (LinearMap.ker (g.baseChange p.asIdeal.ResidueField)))

theorem **projective_coker_of_isLocallyConstant** [IsReduced R] [Module.Finite R M₀] [Module.Finite R M₁] [Module.Projective R M₁] [Module.Finite R M₂] [Module.Projective R M₂] (f : M₀ →ₗ[R] M₁) (g : M₁ →ₗ[R] M₂) (hfg : g ∘ₗ f = 0) (h : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField (LinearMap.ker (g.baseChange p.asIdeal.ResidueField) ⧸ (LinearMap.range (f.baseChange p.asIdeal.ResidueField)).submoduleOf (LinearMap.ker (g.baseChange p.asIdeal.ResidueField)))) : Module.Projective R (M₁ ⧸ LinearMap.range f) ∧ Module.Projective R (M₂ ⧸ LinearMap.range g)

theorem **grauert_linearMap_homology** [IsReduced R] [Module.Finite R M₀] [Module.Finite R M₁] [Module.Projective R M₁] [Module.Finite R M₂] [Module.Projective R M₂] (f : M₀ →ₗ[R] M₁) (g : M₁ →ₗ[R] M₂) (hfg : g ∘ₗ f = 0) (h : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField (LinearMap.ker (g.baseChange p.asIdeal.ResidueField) ⧸ (LinearMap.range (f.baseChange p.asIdeal.ResidueField)).submoduleOf (LinearMap.ker (g.baseChange p.asIdeal.ResidueField)))) : Module.Finite R (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) ∧ Module.Projective R (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) ∧ ∀ (S : Type*) [CommRing S] [Algebra R S], ∃ e : S ⊗[R] (LinearMap.ker g ⧸ (LinearMap.range f).submoduleOf (LinearMap.ker g)) ≃ₗ[S] LinearMap.ker (g.baseChange S) ⧸ (LinearMap.range (f.baseChange S)).submoduleOf (LinearMap.ker (g.baseChange S

theorem **free_of_finite_of_isTorsionFree** (M : Type*) [AddCommGroup M] [Module R M] [Module.Finite R M] [Module.IsTorsionFree R M] : Module.Free R M

theorem **ker_of_valuationRing** [Module.Finite R M₀] [Module.IsTorsionFree R M₀] [Module.IsTorsionFree R M₁] (d : M₀ →ₗ[R] M₁) : Module.Free R (LinearMap.range d) ∧ Module.Finite R (LinearMap.ker d) ∧ Module.Free R (LinearMap.ker d) ∧ (∃ N : Submodule R M₀, IsCompl (LinearMap.ker d) N) ∧ ∀ (S N : Type*) [CommRing S] [Algebra R S] [AddCommGroup N] [Module R N] [Module S N] [IsScalarTower R S N], Injective (LinearMap.tensorKer S N d) ∧ (Module.Flat R N → Bijective (LinearMap.tensorKer S N d))

theorem **preservesKernel_of_comp_comp_eq** {X Y : C} (g : X ⟶ Y) [HasKernel g] (s : Y ⟶ X) (hs : g ≫ s ≫ g = g) : PreservesLimit (parallelPair g 0) F

theorem **preservesCokernel_of_comp_comp_eq** {X Y : C} (f : X ⟶ Y) [HasCokernel f] (u : Y ⟶ X) (hu : f ≫ u ≫ f = f) : PreservesColimit (parallelPair f 0) F

theorem **preservesLeftHomologyOf_of_comp_comp_eq** (S : ShortComplex C) (u : S.X₂ ⟶ S.X₁) (hu : S.f ≫ u ≫ S.f = S.f) (s : S.X₃ ⟶ S.X₂) (hs : S.g ≫ s ≫ S.g = S.g) : F.PreservesLeftHomologyOf S

theorem **preservesRightHomologyOf_of_comp_comp_eq** (S : ShortComplex C) (u : S.X₂ ⟶ S.X₁) (hu : S.f ≫ u ≫ S.f = S.f) (s : S.X₃ ⟶ S.X₂) (hs : S.g ≫ s ≫ S.g = S.g) : F.PreservesRightHomologyOf S

theorem **nonempty_homology_equiv** {R : Type u₁} [Ring R] (S : ShortComplex (ModuleCat.{v} R)) : Nonempty (S.homology ≃ₗ[R] (LinearMap.ker S.g.hom ⧸ (LinearMap.range S.f.hom).submoduleOf (LinearMap.ker S.g.hom)))

theorem **nonempty_homology_map_extendScalars_equiv** {R : Type u₁} (A : Type u₂) [CommRing R] [CommRing A] [inst : Algebra R A] (S : ShortComplex (ModuleCat.{v} R)) : Nonempty ((S.map (ModuleCat.extendScalars (algebraMap R A))).homology ≃ₗ[A] (LinearMap.ker (S.g.hom.baseChange A) ⧸ (LinearMap.range (S.f.hom.baseChange A)).submoduleOf (LinearMap.ker (S.g.hom.baseChange A))))

theorem **nonempty_extendScalars_obj_equiv** {R : Type u₁} (A : Type u₂) [CommRing R] [CommRing A] [inst : Algebra R A] (M : ModuleCat.{v} R) : Nonempty ((ModuleCat.extendScalars (algebraMap R A)).obj M ≃ₗ[A] A ⊗[R] M)

theorem **sum_finrank_homology_eq** {k : Type u₁} [Field k] {ι : Type*} {c : ComplexShape ι} [c.EulerCharSigns] (K : HomologicalComplex (ModuleCat.{v} k) c) [∀ i, Module.Finite k (K.X i)] (s : Finset ι) (hs : ∀ i ∉ s, IsZero (K.X i)) : ∑ i ∈ s, (c.χ i : ℤ) * finrank k (K.homology i) = ∑ i ∈ s, (c.χ i : ℤ) * finrank k (K.X i)

theorem **finite_homology_extendScalars** (K : HomologicalComplex (ModuleCat.{v} R) c) (i : ι) [Module.Finite R (K.X i)] (A : Type w) [CommRing A] [IsNoetherianRing A] [Algebra R A] : Module.Finite A ((((ModuleCat.extendScalars (algebraMap R A)).mapHomologicalComplex c).obj K).homology i)

theorem **upperSemicontinuous_finrank_homology_fiber** (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (i : ι) : UpperSemicontinuous fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homology i)

theorem **sum_finrank_homology_extendScalars** [c.EulerCharSigns] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] (s : Finset ι) (hs : ∀ i ∉ s, IsZero (K.X i)) (k : Type w) [Field k] [Algebra R k] : ∑ i ∈ s, (c.χ i : ℤ) * finrank k ((((ModuleCat.extendScalars (algebraMap R k)).mapHomologicalComplex c).obj K).homology i) = ∑ i ∈ s, (c.χ i : ℤ) * finrank k (k ⊗[R] K.X i)

theorem **sum_finrank_homology_fiber_eq_sum_rankAtStalk** [c.EulerCharSigns] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (s : Finset ι) (hs : ∀ i ∉ s, IsZero (K.X i)) (p : PrimeSpectrum R) : ∑ i ∈ s, (c.χ i : ℤ) * finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homology i) = ∑ i ∈ s, (c.χ i : ℤ) * Module.rankAtStalk (R

theorem **isLocallyConstant_sum_finrank_homology_fiber** [c.EulerCharSigns] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (s : Finset ι) (hs : ∀ i ∉ s, IsZero (K.X i)) : IsLocallyConstant fun p : PrimeSpectrum R => ∑ i ∈ s, (c.χ i : ℤ) * finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homology i)

theorem **isLocallyConstant_homologyEulerChar_fiber** [c.EulerCharSigns] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (s : Finset ι) (hs : ∀ i ∉ s, IsZero (K.X i)) : IsLocallyConstant fun p : PrimeSpectrum R => (((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homologyEulerChar

theorem **isLocallyConstant_sum_range_finrank_homology_fiber** (K : CochainComplex (ModuleCat.{v} R) ℕ) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (n : ℕ) (hn : ∀ i, n < i → IsZero (K.X i)) : IsLocallyConstant fun p : PrimeSpectrum R => ∑ i ∈ Finset.range (n + 1), (-1 : ℤ) ^ i * finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex _).obj K).homology i)

theorem **map_iCycles_d** {C D : Type*} [Category* C] [Category* D] [Abelian C] [Preadditive D] (F : C ⥤ D) [F.Additive] (K : HomologicalComplex C c) (i j : ι) : F.map (K.iCycles i) ≫ ((F.mapHomologicalComplex c).obj K).d i j = 0

theorem **exists_map_homology_iso** {C D : Type*} [Category* C] [Category* D] [Abelian C] [Abelian D] (F : C ⥤ D) [F.Additive] (K : HomologicalComplex C c) (i : ι) [F.PreservesLeftHomologyOf (K.sc i)] : ∃ e : F.obj (K.homology i) ≅ ((F.mapHomologicalComplex c).obj K).homology i, F.map (K.homologyπ i) ≫ e.hom = ((F.mapHomologicalComplex c).obj K).liftCycles (F.map (K.iCycles i)) (c.next i) rfl (map_iCycles_d F K i _) ≫ ((F.mapHomologicalComplex c).obj K).homologyπ i

theorem **exists_generalizedInverse_of_isLocallyConstant** [IsReduced R] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (i : ι) (h : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homology i)) : (∃ u : K.X i ⟶ K.X (c.prev i), K.d (c.prev i) i ≫ u ≫ K.d (c.prev i) i = K.d (c.prev i) i) ∧ ∃ s : K.X (c.next i) ⟶ K.X i, K.d i (c.next i) ≫ s ≫ K.d i (c.next i) = K.d i (c.next i)

theorem **preservesHomologyOf_of_isLocallyConstant** [IsReduced R] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (i : ι) (h : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homology i)) {D : Type*} [Category* D] [Preadditive D] (F : ModuleCat.{v} R ⥤ D) [F.Additive] : F.PreservesLeftHomologyOf (K.sc i) ∧ F.PreservesRightHomologyOf (K.sc i)

theorem **grauert_homology** [IsReduced R] (K : HomologicalComplex (ModuleCat.{v} R) c) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (i : ι) (h : IsLocallyConstant fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex c).obj K).homology i)) : Module.Finite R (K.homology i) ∧ Module.Projective R (K.homology i) ∧ ∀ (S : Type w) [CommRing S] (φ : R →+* S), ∃ e : (ModuleCat.extendScalars φ).obj (K.homology i) ≅ (((ModuleCat.extendScalars φ).mapHomologicalComplex c).obj K).homology i, (ModuleCat.extendScalars φ).map (K.homologyπ i) ≫ e.hom = (((ModuleCat.extendScalars φ).mapHomologicalComplex c).obj K).liftCycles ((ModuleCat.extendScalars φ).map (K.iCycles i)) (c.next i) rfl (map_iCycles_d _ K i _) ≫ (((ModuleCat.extendScalars φ).mapHomologicalComplex c).obj K).homologyπ i

theorem **ker_free_of_valuationRing** {O : Type*} [CommRing O] [IsDomain O] [ValuationRing O] (F : Type*) [Field F] [Algebra O F] [IsFractionRing O F] {M₀ M₁ : Type*} [AddCommGroup M₀] [Module O M₀] [AddCommGroup M₁] [Module O M₁] [Module.Finite O M₀] [Module.Projective O M₀] [Module.Projective O M₁] (d : M₀ →ₗ[O] M₁) : Module.Finite O (LinearMap.ker d) ∧ Module.Free O (LinearMap.ker d) ∧ (∃ N : Submodule O M₀, IsCompl (LinearMap.ker d) N) ∧ Bijective (LinearMap.tensorKer F F d) ∧ Injective (LinearMap.tensorKer (IsLocalRing.ResidueField O) (IsLocalRing.ResidueField O) d)

theorem **semicontinuity_finite_projective_complex** (K : CochainComplex (ModuleCat.{v} R) ℕ) [∀ i, Module.Finite R (K.X i)] [∀ i, Module.Projective R (K.X i)] (n : ℕ) (hn : ∀ i, n < i → IsZero (K.X i)) : (∀ (i : ℕ) (p : PrimeSpectrum R), Module.Finite p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex _).obj K).homology i)) ∧ (∀ i : ℕ, UpperSemicontinuous fun p : PrimeSpectrum R => finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex _).obj K).homology i)) ∧ (IsLocallyConstant fun p : PrimeSpectrum R => ∑ i ∈ Finset.range (n + 1), (-1 : ℤ) ^ i * finrank p.asIdeal.ResidueField ((((ModuleCat.extendScalars (algebraMap R p.asIdeal.ResidueField)).mapHomologicalComplex _).obj K).homology i)) ∧ (∀ i : ℕ, IsReduced R → (IsLocallyConstant fun p : PrimeSpectrum R => finrank

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.PushComparisonBijectiveOfIsClosedEmbedding

lemma **restr_eq** {j : Y ⟶ X} (hj : Topology.IsOpenEmbedding j) : restr hj = hj.sheafPullback AddCommGrpCat.{u}

theorem **shortExact_map_restr** {j : Y ⟶ X} (hj : Topology.IsOpenEmbedding j) {S : ShortComplex (AbSheaf X)} (hS : S.ShortExact) : (S.map (restr hj)).ShortExact

theorem **isFlasque_restr** {j : Y ⟶ X} (hj : Topology.IsOpenEmbedding j) (F : AbSheaf X) (hF : TopCat.Sheaf.IsFlasque F) : TopCat.Sheaf.IsFlasque ((restr hj).obj F) where

theorem **isFlasque_push** (f : Y ⟶ X) (G : AbSheaf Y) (hG : TopCat.Sheaf.IsFlasque G) : TopCat.Sheaf.IsFlasque ((push f).obj G) where

lemma **isFlasque_of_injective'** (I : AbSheaf X) [hI : Injective I] : TopCat.Sheaf.IsFlasque I

lemma **epi_push_map** (f : Y ⟶ X) {G G' : AbSheaf Y} (g : G ⟶ G') (h : ∀ (V : Opens X) (t : G'.obj.obj (op ((Opens.map f).obj V))), ∀ x ∈ V, ∃ (W : Opens X) (i : W ⟶ V), x ∈ W ∧ ∃ s, g.hom.app (op ((Opens.map f).obj W)) s = G'.obj.map ((Opens.map f).map i).op t) : Epi ((push f).map g)

lemma **shortExact_map_push** (f : Y ⟶ X) {S : ShortComplex (AbSheaf Y)} (hS : S.ShortExact) (h : Epi ((push f).map S.g)) : (S.map (push f)).ShortExact

theorem **preservesEpimorphisms_push** {f : Y ⟶ X} (hf : Topology.IsClosedEmbedding f) : (push f).PreservesEpimorphisms where

theorem **shortExact_map_push_of_isClosedEmbedding** {f : Y ⟶ X} (hf : Topology.IsClosedEmbedding f) {S : ShortComplex (AbSheaf Y)} (hS : S.ShortExact) : (S.map (push f)).ShortExact

theorem **pull_restrUnit_bijective** (F : AbSheaf X) (n : ℕ) {V : Opens X} {W : Opens Y} (i : W ⟶ (Opens.map j).obj V) (h : V ≤ hj.functor.obj W) : Function.Bijective (pull.{w, w'} j ((restrUnit hj).app F) n i)

theorem **restrEquiv_δ** {S : ShortComplex (AbSheaf X)} (hS : S.ShortExact) (n : ℕ) (V : Opens Y) (x : Sheaf.H'.{w} S.X₃ n (hj.functor.obj V)) : restrEquiv.{w, w'} hj S.X₁ (n + 1) V (δ hS n _ x) = δ (shortExact_map_restr hj hS) n V (restrEquiv.{w, w'} hj S.X₃ n V x)

theorem **pushComparison_map** {G G' : AbSheaf Y} (t : G ⟶ G') (n : ℕ) (V : Opens X) (x : Sheaf.H'.{w} ((push f).obj G) n V) : pushComparison.{w, w'} f G' n V (map ((push f).map t) n V x) = map t n _ (pushComparison.{w, w'} f G n V x)

theorem **pushComparison_δ** {S : ShortComplex (AbSheaf Y)} (hS : S.ShortExact) (hS' : (S.map (push f)).ShortExact) (n : ℕ) (V : Opens X) (x : Sheaf.H'.{w} ((push f).obj S.X₃) n V) : pushComparison.{w, w'} f S.X₁ (n + 1) V (δ hS' n V x) = δ hS n _ (pushComparison.{w, w'} f S.X₃ n V x)

theorem **pull_eq** {F : AbSheaf X} {G : AbSheaf Y} (φ : F ⟶ (push f).obj G) (n : ℕ) {V : Opens X} {W : Opens Y} (i : W ⟶ (Opens.map f).obj V) (x : Sheaf.H'.{w} F n V) : pull.{w, w'} f φ n i x = res G n i (pushComparison.{w, w'} f G n V (map φ n V x))

theorem **pushComparison_bijective_zero** (G : AbSheaf Y) (V : Opens X) : Function.Bijective (pushComparison.{w, w'} f G 0 V)

theorem **pushComparison_bijective_of_class** (P : AbSheaf Y → Prop) (hP : ∀ S : ShortComplex (AbSheaf Y), S.ShortExact → Injective S.X₂ → P S.X₁ → Epi ((push f).map S.g) ∧ P S.X₃) (G : AbSheaf Y) (hG : P G) (n : ℕ) (V : Opens X) : Function.Bijective (pushComparison.{w, w'} f G n V)

theorem **higherDirectImagesVanish_of_basis** (G : AbSheaf Y) (h : ∀ (V : Opens X), ∀ x ∈ V, ∃ W ≤ V, x ∈ W ∧ ∀ n, 0 < n → Subsingleton (Sheaf.H'.{w'} G n ((Opens.map f).obj W))) : HigherDirectImagesVanish.{w'} f G

theorem **HigherDirectImagesVanish.epi** {S : ShortComplex (AbSheaf Y)} (hS : S.ShortExact) (h : HigherDirectImagesVanish.{w'} f S.X₁) : Epi ((push f).map S.g)

theorem **HigherDirectImagesVanish.shortExact** {S : ShortComplex (AbSheaf Y)} (hS : S.ShortExact) (h : HigherDirectImagesVanish.{w'} f S.X₁) : (S.map (push f)).ShortExact

theorem **HigherDirectImagesVanish.quotient** {S : ShortComplex (AbSheaf Y)} (hS : S.ShortExact) [Injective S.X₂] (h : HigherDirectImagesVanish.{w'} f S.X₁) : HigherDirectImagesVanish.{w'} f S.X₃

theorem **pushComparison_bijective** {G : AbSheaf Y} (h : HigherDirectImagesVanish.{w'} f G) (n : ℕ) (V : Opens X) : Function.Bijective (pushComparison.{w, w'} f G n V)

theorem **pushComparison_bijective_of_isClosedEmbedding** (hf : Topology.IsClosedEmbedding f) (G : AbSheaf Y) (n : ℕ) (V : Opens X) : Function.Bijective (pushComparison.{w, w'} f G n V)

lemma **pushEquiv_apply** {G : AbSheaf Y} (h : HigherDirectImagesVanish.{w'} f G) (n : ℕ) (V : Opens X) (x : Sheaf.H'.{w} ((push f).obj G) n V) : pushEquiv.{w, w'} h n V x = pushComparison.{w, w'} f G n V x

lemma **closedEquiv_apply** (hf : Topology.IsClosedEmbedding f) (G : AbSheaf Y) (n : ℕ) (V : Opens X) (x : Sheaf.H'.{w} ((push f).obj G) n V) : closedEquiv.{w, w'} hf G n V x = pushComparison.{w, w'} f G n V x

theorem **injective_push** (f : Y ⟶ X) (I : AbSheaf Y) [Injective I] : Injective ((push f).obj I)

lemma **exists_hom_injectiveShortComplex** (f : Y ⟶ X) {F : AbSheaf X} {G : AbSheaf Y} (φ : F ⟶ (push f).obj G) : ∃ ψ : injectiveShortComplex F ⟶ (injectiveShortComplex G).map (push f), ψ.τ₁ = φ

theorem **pull_pull** (f : Y ⟶ X) (g : Z ⟶ Y) {F : AbSheaf X} {G : AbSheaf Y} {L : AbSheaf Z} (φ : F ⟶ (push f).obj G) (ψ : G ⟶ (push g).obj L) (n : ℕ) {V : Opens X} {W : Opens Y} {T : Opens Z} (i : W ⟶ (Opens.map f).obj V) (k : T ⟶ (Opens.map g).obj W) (x : Sheaf.H'.{w} F n V) : pull.{w', w''} g ψ n k (pull.{w, w'} f φ n i x) = pull.{w, w''} (g ≫ f) (G

theorem **pull_id** {F G : AbSheaf X} (φ : F ⟶ G) (n : ℕ) {V W : Opens X} (i : W ⟶ V) (x : Sheaf.H'.{w} F n V) : pull.{w, w} (𝟙 X) (G

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RelAmpleNearFibreFlatProjective

theorem **isProper_lift** (ψ : Y' ⟶ P) (f' : Y' ⟶ T) [IsProper f'] (q : P ⟶ S) [IsSeparated q] (t : T ⟶ S) (h : ψ ≫ q = f' ≫ t) : IsProper (pullback.lift ψ f' h)

theorem **isFinite_lift** (ψ : Y' ⟶ P) (f' : Y' ⟶ T) [IsProper f'] (q : P ⟶ S) [IsSeparated q] (t : T ⟶ S) [IsAffineHom t] (h : ψ ≫ q = f' ≫ t) {ι : Type} (D : ι → P.Opens) (hD : ⊔ i, D i = ⊤) (hD' : ∀ i, IsAffineOpen (D i)) (hψ : ∀ i, IsAffineOpen (ψ ⁻¹ᵁ D i)) : IsFinite (pullback.lift ψ f' h)

theorem **isAffineHom_of_locallyQuasiFinite** (ψ : Y' ⟶ P) [LocallyQuasiFinite ψ] (f' : Y' ⟶ T) [IsProper f'] (q : P ⟶ S) [IsSeparated q] (t : T ⟶ S) [IsAffineHom t] (h : ψ ≫ q = f' ≫ t) : IsAffineHom ψ

theorem **exists_nhds_quasiFinite** (f : Y ⟶ S) [IsProper f] (j : W ⟶ Y) [IsOpenImmersion j] (q : P ⟶ S) [IsSeparated q] (φ : W ⟶ P) [LocallyOfFiniteType φ] (hφ : φ ≫ q = j ≫ f) (t : T ⟶ S) [IsAffineHom t] (ρ : Y' ⟶ W) (f' : Y' ⟶ T) (h : IsPullback (ρ ≫ j) f' f t) {ι : Type} (D : ι → P.Opens) (hD : ⊔ i, D i = ⊤) (hD' : ∀ i, IsAffineOpen (D i)) (hψ : ∀ i, IsAffineOpen ((ρ ≫ φ) ⁻¹ᵁ D i)) (s : T) : ∃ U : S.Opens, t s ∈ U ∧ f ⁻¹ᵁ U ≤ j ''ᵁ φ.quasiFiniteLocus

theorem **exists_isAffineHom** (f : Y ⟶ S) [IsProper f] (j : W ⟶ Y) [IsOpenImmersion j] (q : P ⟶ S) [IsSeparated q] (φ : W ⟶ P) [LocallyOfFiniteType φ] (hφ : φ ≫ q = j ≫ f) (t : T ⟶ S) [IsAffineHom t] [IsOpenImmersion t] (π : Y' ⟶ Y) (f' : Y' ⟶ T) (h : IsPullback π f' f t) (hπ : ∀ y, π y ∈ j ''ᵁ φ.quasiFiniteLocus) : ∃ ρ : Y' ⟶ W, ρ ≫ j = π ∧ IsAffineHom (ρ ≫ φ)

theorem **res_res** {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (u : Γ(X, U)ˣ) : res h₂ (res h₁ u) = res (h₂.trans h₁) u

theorem **pw_pw** (H : LB X) (a b : ℤ) : AbLaw.pw (AbLaw.pw H a) b = AbLaw.pw H (a * b)

theorem **pw_pull** (H : LB X) (f : Y ⟶ X) (a : ℤ) : (AbLaw.pw H a).pull f = AbLaw.pw (H.pull f) a

theorem **cocycle_on** {H : LB X} (hH : H.IsLB) (i j k : H.ι) {W : X.Opens} (hi : W ≤ H.U i) (hj : W ≤ H.U j) (hk : W ≤ H.U k) : res (le_inf hi hj) (H.g i j) * res (le_inf hj hk) (H.g j k) = res (le_inf hi hk) (H.g i k)

theorem **rel_on** {H : LB X} (hH : H.IsLB) (i k i' k' : H.ι) {W : X.Opens} (hi : W ≤ H.U i) (hk : W ≤ H.U k) (hi' : W ≤ H.U i') (hk' : W ≤ H.U k') : res (le_inf hi hk) (H.g i k) = res (le_inf hi hi') (H.g i i') * (res (le_inf hk hk') (H.g k k'))⁻¹ * res (le_inf hi' hk') (H.g i' k')

theorem **lbIso_tw** {H : LB X} (hH : H.IsLB) (t : ℤ) : LBIso (AbLaw.pw H (t + 1)) (tw H H t)

theorem **exists_pw_h1Zero** (R : Type) [CommRing R] [IsNoetherianRing R] (B : Over (Spec (CommRingCat.of R))) (L : LB B.left) (hL : L.IsLB) (h : RelAmple B L) : ∃ d : ℕ, 1 ≤ d ∧ FirstCohomologyVanishes (AbLaw.pw L d) ∧ ∃ (n : ℕ) (s : Fin (n + 1) → LB.Sec (AbLaw.pw L d)), VeryAmpleBy B.hom (AbLaw.pw L d) s

theorem **preimage_nonzeroLocus_of_pull_eq** {f : Y ⟶ Spec (CommRingCat.of R)} {f₀ : Y₀ ⟶ Spec (CommRingCat.of K)} {π : Y₀ ⟶ Y} {g : R →+* K} (w : π ≫ f = f₀ ≫ Spec.map (CommRingCat.ofHom g)) (M : LB Y) (x : SecOnR f M ⊤) (x₀ : SecOnR f₀ (M.pull π) ⊤) {c : K} (hc : c ≠ 0) (h : SecOnR.pull M w ⊤ ⊤ le_top x = c • x₀) : π ⁻¹ᵁ nonzeroLocus (SecOnR.topEquiv f M x) = nonzeroLocus (SecOnR.topEquiv f₀ (M.pull π) x₀)

theorem **exists_hom_preimage_eq** (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (hH : H.IsLB) {n : ℕ} (s : Fin (n + 1) → LB.Sec H) (hs : ⊔ i, nonzeroLocus (s i) = ⊤) : ∃ (P : Scheme.{0}) (q : P ⟶ Spec (CommRingCat.of R)) (φ : Y ⟶ P) (D : Fin (n + 1) → P.Opens), IsProper q ∧ φ ≫ q = f ∧ ⊔ i, D i = ⊤ ∧ (∀ i, IsAffineOpen (D i)) ∧ ∀ i, φ ⁻¹ᵁ D i = nonzeroLocus (s i)

theorem **relAmple_of_affine_loci** {S : Scheme.{0}} [IsAffine S] (A : Over S) [IsProper A.hom] (L : LB A.left) (hL : L.IsLB) {d : ℕ} (hd : 1 ≤ d) {n : ℕ} (s : Fin (n + 1) → LB.Sec (pw L d)) (hs : ⊔ i, nonzeroLocus (s i) = ⊤) (hs' : ∀ i, IsAffineOpen (nonzeroLocus (s i))) : RelAmple A L

theorem **exists_affine_loci_near** (f : Y ⟶ Spec (CommRingCat.of R)) [IsProper f] (H : LB Y) (hH : H.IsLB) {n : ℕ} (s : Fin (n + 1) → LB.Sec H) (g : R →+* K) (π : Y₀ ⟶ Y) (f₀ : Y₀ ⟶ Spec (CommRingCat.of K)) (h : IsPullback π f₀ f (Spec.map (CommRingCat.ofHom g))) (hs : ⊔ i, π ⁻¹ᵁ nonzeroLocus (s i) = ⊤) (hs' : ∀ i, IsAffineOpen (π ⁻¹ᵁ nonzeroLocus (s i))) : ∃ r : R, g r ≠ 0 ∧ ∀ (Y' : Scheme.{0}) (π' : Y' ⟶ Y) f', IsPullback π' f' f (Scheme.Opens.ι (X

theorem **relAmple_near_fibre** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (B : Over (Spec (CommRingCat.of R))) (L : LB B.left), Flat B.hom → IsProjBy B.hom → L.IsLB → ∀ (K : Type) [Field K] (g : R →+* K), (∀ y : K, ∃ a b : R, g b ≠ 0 ∧ y * g b = g a) → RelAmple ((Over.pullback (Spec.map (CommRingCat.ofHom g))).obj B) (L.pull (pullback.fst B.hom (Spec.map (CommRingCat.ofHom g)))) → ∃ r : R, g r ≠ 0 ∧ RelAmple ((Over.pullback (Scheme.Opens.ι (X

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RelAmpleNearFibreFlatProjectiveStatement

theorem **_root_.GlobalTheory.T.relAmple_near_fibre_flat_projective** : ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (B : Over (Spec (CommRingCat.of R))) (L : LB B.left), Flat B.hom → IsProjBy B.hom → L.IsLB → ∀ (K : Type) [Field K] (g : R →+* K), (∀ y : K, ∃ a b : R, g b ≠ 0 ∧ y * g b = g a) → RelAmple ((Over.pullback (Spec.map (CommRingCat.ofHom g))).obj B) (L.pull (pullback.fst B.hom (Spec.map (CommRingCat.ofHom g)))) → ∃ r : R, g r ≠ 0 ∧ RelAmple ((Over.pullback (Scheme.Opens.ι (X

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RelLengthSpanEqAddValDet

theorem **addVal_prod** (s : Finset ι) (c : ι → R) : addVal R (∏ i ∈ s, c i) = ∑ i ∈ s, addVal R (c i)

theorem **length_self_eq_top** : Module.length R R = ⊤

theorem **length_coker_of_det_eq_zero** (P : Matrix ι ι R) (h : P.det = 0) : Module.length R ((ι → R) ⧸ LinearMap.range P.mulVecLin) = ⊤

theorem **length_coker_of_injective** {M : Type*} [AddCommGroup M] [Module R M] (b : Module.Basis ι R M) (f : M →ₗ[R] M) (hf : Function.Injective f) : Module.length R (M ⧸ LinearMap.range f) = addVal R (LinearMap.det f)

theorem **length_coker** (P : Matrix ι ι R) : Module.length R ((ι → R) ⧸ LinearMap.range P.mulVecLin) = addVal R P.det

theorem **relLength_span_eq_addVal_det** {V : Type*} [AddCommGroup V] [Module R V] {a v : ι → V} (ha : LinearIndependent R a) (P : Matrix ι ι R) (hv : ∀ j, v j = ∑ i, P i j • a i) : relLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range v)) = addVal R P.det

theorem **relLength_span_eq** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {V : Type u_2} [AddCommGroup V] [Module R V] {ι : Type u_3} [Fintype ι] [DecidableEq ι] {a v : ι → V} (ha : LinearIndependent R a) (P : Matrix ι ι R) (hv : ∀ j, v j = ∑ i, P i j • a i) : relLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range v)) = (addVal R : R → ℕ∞) P.det

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RelLengthSpanMatrix

theorem **relLength_tower** {R V : Type*} [Ring R] [AddCommGroup V] [Module R V] {A B C : Submodule R V} (hBA : B ≤ A) (hCB : C ≤ B) : relLength A C = relLength B C + relLength A B

theorem **ord_prod** {R ι : Type*} [CommRing R] [IsDomain R] (s : Finset ι) (d : ι → R) (hd : ∀ i, d i ≠ 0) : Ring.ord R (∏ i ∈ s, d i) = ∑ i ∈ s, Ring.ord R (d i)

theorem **length_coker** {R ι : Type*} [CommRing R] [IsDomain R] [IsPrincipalIdealRing R] [Fintype ι] (f : (ι → R) →ₗ[R] (ι → R)) (hf : Function.Injective f) : Module.length R ((ι → R) ⧸ LinearMap.range f) = Ring.ord R (LinearMap.det f)

theorem **relLength_span_matrix** {R V ι : Type*} [CommRing R] [IsDomain R] [IsPrincipalIdealRing R] [AddCommGroup V] [Module R V] [Fintype ι] [DecidableEq ι] {a c : ι → V} (ha : LinearIndependent R a) (M : Matrix ι ι R) (hM : M.det ≠ 0) (hc : ∀ j, c j = ∑ i, M i j • a i) : relLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range c)) = Ring.ord R M.det

theorem **signedLength_span_eq** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {K : Type u_2} [Field K] [Algebra R K] [IsFractionRing R K] {V : Type u_3} [AddCommGroup V] [Module K V] [Module R V] [IsScalarTower R K V] {ι : Type u_4} [Fintype ι] [DecidableEq ι] {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hb : ∀ j, b j = ∑ i, g i j • a i) {ϖ : R} (hϖ : Irreducible ϖ) (u : Rˣ) (n : ℤ) (hdet : g.det = algebraMap R K (u : R) * algebraMap R K ϖ ^ n) : signedLength (Submodule.span R (Set.range a)) (Submodule.span R (Set.range b)) = n

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RestrictPullbackIsoHomAppPullbackSections

theorem **section_eq_of_cover** {ι : Type*} (𝒰 : ι → X.Opens) (h : ∀ i, 𝒰 i ≤ W) (hcov : W ≤ iSup 𝒰) (s t : Γ(F, W)) (H : ∀ i, F.presheaf.map (homOfLE (h i)).op s = F.presheaf.map (homOfLE (h i)).op t) : s = t

theorem **exists_finite_affine_cover_of_isCompact** (hW : IsCompact (W : Set X)) : ∃ (ι : Type u) (_ : Fintype ι) (𝒰 : ι → X.Opens), (∀ i, IsAffineOpen (𝒰 i)) ∧ W = iSup 𝒰

theorem **exists_pow_res_smul_eq_zero_of_isCompact** (hW : IsCompact (W : Set X)) (h : W ≤ U) (f : Γ(X, U)) (t : Γ(F, W)) (ht : F.presheaf.map (homOfLE (inf_le_left : W ⊓ X.basicOpen f ≤ W)).op t = 0) : ∃ n : ℕ, X.presheaf.map (homOfLE h).op f ^ n • t = 0

theorem **exists_pow_res_smul_eq_res_of_qcqs** (hU : IsCompact (U : Set X)) (hU' : IsQuasiSeparated (U : Set X)) (f : Γ(X, U)) (s : Γ(F, X.basicOpen f)) : ∃ (n : ℕ) (t : Γ(F, U)), X.presheaf.map (homOfLE (X.basicOpen_le f)).op f ^ n • s = F.presheaf.map (homOfLE (X.basicOpen_le f)).op t

theorem **pushforward_smul** (V : Y.Opens) (r : Γ(Y, V)) (s : Γ((Scheme.Modules.pushforward f).obj F, V)) : r • s = (f.app V r • (show Γ(F, f ⁻¹ᵁ V) from s) : Γ(F, f ⁻¹ᵁ V))

theorem **finite_sections_pushforward** [IsFinite f] {V : Y.Opens} (hV : IsAffineOpen V) [hF : Module.Finite Γ(X, f ⁻¹ᵁ V) Γ(F, f ⁻¹ᵁ V)] : Module.Finite Γ(Y, V) Γ((Scheme.Modules.pushforward f).obj F, V)

theorem **hom_app_res** {F F' : X.Modules} (φ : F ⟶ F') {U V : X.Opens} (h : V ≤ U) (s : Γ(F, U)) : φ.app V (F.presheaf.map (homOfLE h).op s) = F'.presheaf.map (homOfLE h).op (φ.app U s)

theorem **homAppLinear_apply** {F F' : X.Modules} (φ : F ⟶ F') (U : X.Opens) (s : Γ(F, U)) : homAppLinear φ U s = φ.app U s

theorem **appLEModule_smul** (e : U ≤ f ⁻¹ᵁ V) (r : Γ(Y, V)) (s : Γ(F, U)) : letI

theorem **appLE_isScalarTower** (e : U ≤ f ⁻¹ᵁ V) : letI

theorem **pullbackSections_apply** (e : U ≤ f ⁻¹ᵁ V) (s : Γ(G, V)) : pullbackSections f G e s = ((Scheme.Modules.pullback f).obj G).presheaf.map (homOfLE e).op (((Scheme.Modules.pullbackPushforwardAdjunction f).unit.app G).app V s)

theorem **pullbackSections_smul** (e : U ≤ f ⁻¹ᵁ V) (r : Γ(Y, V)) (s : Γ(G, V)) : pullbackSections f G e (r • s) = f.appLE V U e r • pullbackSections f G e s

theorem **res_pullbackSections** (e : U ≤ f ⁻¹ᵁ V) (h : U' ≤ U) (s : Γ(G, V)) : ((Scheme.Modules.pullback f).obj G).presheaf.map (homOfLE h).op (pullbackSections f G e s) = pullbackSections f G (h.trans e) s

theorem **pullbackSections_res** (e : U ≤ f ⁻¹ᵁ V') (h : V' ≤ V) (s : Γ(G, V)) : pullbackSections f G e (G.presheaf.map (homOfLE h).op s) = pullbackSections f G (e.trans (f.preimage_mono h)) s

theorem **pullback_map_app_pullbackSections** {G G' : Y.Modules} (φ : G ⟶ G') (e : U ≤ f ⁻¹ᵁ V) (s : Γ(G, V)) : ((Scheme.Modules.pullback f).map φ).app U (pullbackSections f G e s) = pullbackSections f G' e (φ.app V s)

theorem **res_bijective_of_eq** (h : U' ≤ U) (e : U' = U) : Function.Bijective (F.presheaf.map (homOfLE h).op)

theorem **fromSpec_image_top** (hU : IsAffineOpen U) : hU.fromSpec ''ᵁ ⊤ = U

theorem **sectionsEquivFromSpec_apply** (hU : IsAffineOpen U) (s : Γ(F, U)) : sectionsEquivFromSpec F hU s = F.presheaf.map (homOfLE (fromSpec_image_le hU ⊤)).op s

theorem **tildePullbackIso_hom_app_toOpen** (M : ModuleCat.{u} R) (m : M) : ((tildePullbackIso φ).hom.app M).app ⊤ (tilde.toOpen ((ModuleCat.extendScalars φ.hom).obj M) ⊤ ((ModuleCat.extendRestrictScalarsAdj φ.hom).unit.app M m)) = ((Scheme.Modules.pullbackPushforwardAdjunction (Spec.map φ)).unit.app (tilde M)).app ⊤ (tilde.toOpen M ⊤ m)

theorem **isQuasicoherent_pullback_Spec** (M : (Spec R).Modules) [M.IsQuasicoherent] : ((Scheme.Modules.pullback (Spec.map φ)).obj M).IsQuasicoherent

theorem **exists_equiv_pullback_Spec** (M : (Spec R).Modules) [M.IsQuasicoherent] : letI

theorem **restrictPullbackIso_hom_app_unit** (G : Y.Modules) (V : Y.Opens) (s : Γ(G, V)) : ((restrictPullbackIso w).hom.app G).app (j ⁻¹ᵁ f ⁻¹ᵁ V) (((Scheme.Modules.pullback p).obj (G.restrict i)).presheaf.map (eqToHom (show j ⁻¹ᵁ f ⁻¹ᵁ V = p ⁻¹ᵁ i ⁻¹ᵁ V by rw [← Scheme.Hom.comp_preimage, ← w, Scheme.Hom.comp_preimage])).op (((Scheme.Modules.pullbackPushforwardAdjunction p).unit.app (G.restrict i)).app (i ⁻¹ᵁ V) (G.presheaf.map (homOfLE (i.image_preimage_le V)).op s))) = ((Scheme.Modules.pullback f).obj G).presheaf.map (homOfLE (j.image_preimage_le (f ⁻¹ᵁ V))).op (((Scheme.Modules.pullbackPushforwardAdjunction f).unit.app G).app V s)

theorem **restrictPullbackIso_hom_app_pullbackSections** (G : Y.Modules) {V : Y.Opens} {V' : Y'.Opens} {U' : X'.Opens} (hV : i ''ᵁ V' ≤ V) (e' : U' ≤ p ⁻¹ᵁ V') (e : j ''ᵁ U' ≤ f ⁻¹ᵁ V) (s : Γ(G, V)) : ((restrictPullbackIso w).hom.app G).app U' (pullbackSections p (G.restrict i) e' (G.presheaf.map (homOfLE hV).op s)) = pullbackSections f G e s

theorem **exists_affine_pair** (x : X) {W : X.Opens} (hx : x ∈ W) : ∃ (U : X.Opens) (V : Y.Opens), IsAffineOpen U ∧ IsAffineOpen V ∧ x ∈ U ∧ U ≤ W ∧ U ≤ f ⁻¹ᵁ V

theorem **isBaseChange_pullbackSections** (hU : IsAffineOpen U) (hV : IsAffineOpen V) (e : U ≤ f ⁻¹ᵁ V) : letI

theorem **isBaseChange_of_eq_pullbackSections** (hU : IsAffineOpen U) (hV : IsAffineOpen V) (e : U ≤ f ⁻¹ᵁ V) [alg : Algebra Γ(Y, V) Γ(X, U)] [mod : Module Γ(Y, V) Γ((Scheme.Modules.pullback f).obj G, U)] [IsScalarTower Γ(Y, V) Γ(X, U) Γ((Scheme.Modules.pullback f).obj G, U)] (halg : ∀ r, algebraMap Γ(Y, V) Γ(X, U) r = f.appLE V U e r) (l : Γ(G, V) →ₗ[Γ(Y, V)] Γ((Scheme.Modules.pullback f).obj G, U)) (hl : ∀ s, l s = pullbackSections f G e s) : IsBaseChange Γ(X, U) l

theorem **isBaseChange_pullbackSections_of_isPushout** (hU : IsAffineOpen U) (hV : IsAffineOpen V) (e : U ≤ f ⁻¹ᵁ V) {A B : Type*} [CommRing A] [CommRing B] [Algebra A B] [Algebra A Γ(Y, V)] [Algebra B Γ(X, U)] [Algebra A Γ(X, U)] [Algebra Γ(Y, V) Γ(X, U)] [IsScalarTower A B Γ(X, U)] [IsScalarTower A Γ(Y, V) Γ(X, U)] [Algebra.IsPushout A B Γ(Y, V) Γ(X, U)] (halg : ∀ r, algebraMap Γ(Y, V) Γ(X, U) r = f.appLE V U e r) [Module A Γ(G, V)] [IsScalarTower A Γ(Y, V) Γ(G, V)] [Module A Γ((Scheme.Modules.pullback f).obj G, U)] [Module B Γ((Scheme.Modules.pullback f).obj G, U)] [IsScalarTower A B Γ((Scheme.Modules.pullback f).obj G, U)] [IsScalarTower B Γ(X, U) Γ((Scheme.Modules.pullback f).obj G, U)] (l : Γ(G, V) →ₗ[A] Γ((Scheme.Modules.pullback f).obj G, U)) (hl : ∀ s, l s = pullbackSections f G e s) : IsBaseChange B l

theorem **finite_sections_pullback** (hU : IsAffineOpen U) (hV : IsAffineOpen V) (e : U ≤ f ⁻¹ᵁ V) [Module.Finite Γ(Y, V) Γ(G, V)] : Module.Finite Γ(X, U) Γ((Scheme.Modules.pullback f).obj G, U)

theorem **app_injective_of_mono** (φ : F ⟶ F') [Mono φ] (U : X.Opens) : Function.Injective (φ.app U)

theorem **mono_of_locally_app_injective** (φ : F ⟶ F') (h : ∀ (W : X.Opens) (x : X), x ∈ W → ∃ U : X.Opens, x ∈ U ∧ U ≤ W ∧ Function.Injective (φ.app U)) : Mono φ

theorem **epi_of_locally_app_surjective** (φ : F ⟶ F') (h : ∀ (W : X.Opens) (x : X), x ∈ W → ∃ U : X.Opens, x ∈ U ∧ U ≤ W ∧ Function.Surjective (φ.app U)) : Epi φ

theorem **app_surjective_of_epi_of_isAffineOpen** (φ : F ⟶ F') [Epi φ] [F.IsQuasicoherent] [F'.IsQuasicoherent] {U : X.Opens} (hU : IsAffineOpen U) : Function.Surjective (φ.app U)

theorem **epi_pushforward_map** [IsAffineHom f] (φ : F ⟶ F') [Epi φ] [F.IsQuasicoherent] [F'.IsQuasicoherent] : Epi ((Scheme.Modules.pushforward f).map φ)

theorem **shortExact_map_pushforward** [IsAffineHom f] {S : ShortComplex X.Modules} (hS : S.ShortExact) [S.X₂.IsQuasicoherent] [S.X₃.IsQuasicoherent] : (S.map (Scheme.Modules.pushforward f)).ShortExact

theorem **exact_map_pullback** {S : ShortComplex Y.Modules} (hS : S.Exact) [Epi S.g] : (S.map (Scheme.Modules.pullback f)).Exact ∧ Epi (S.map (Scheme.Modules.pullback f)).g

theorem **injective_of_isBaseChange_of_flat** {A B M M' N N' : Type*} [CommRing A] [CommRing B] [Algebra A B] [Module.Flat A B] [AddCommGroup M] [Module A M] [AddCommGroup M'] [Module A M'] [AddCommGroup N] [Module A N] [Module B N] [IsScalarTower A B N] [AddCommGroup N'] [Module A N'] [Module B N'] [IsScalarTower A B N'] {l : M →ₗ[A] N} {l' : M' →ₗ[A] N'} (hl : IsBaseChange B l) (hl' : IsBaseChange B l') (a : M →ₗ[A] M') (b : N →ₗ[B] N') (hab : ∀ m, b (l m) = l' (a m)) (ha : Function.Injective a) : Function.Injective b

theorem **mono_pullback_map** [Flat f] {G G' : Y.Modules} (φ : G ⟶ G') [Mono φ] [G.IsQuasicoherent] [G'.IsQuasicoherent] : Mono ((Scheme.Modules.pullback f).map φ)

theorem **shortExact_map_pullback** [Flat f] {S : ShortComplex Y.Modules} (hS : S.ShortExact) [S.X₁.IsQuasicoherent] [S.X₂.IsQuasicoherent] : (S.map (Scheme.Modules.pullback f)).ShortExact

theorem **quasicoherent_push_pull** {X Y : Scheme.{u}} (f : X ⟶ Y) : (QuasiCompact f → QuasiSeparated f → ∀ F : X.Modules, F.IsQuasicoherent → ((Scheme.Modules.pushforward f).obj F).IsQuasicoherent) ∧ (IsFinite f ∨ IsClosedImmersion f → ∀ F : X.Modules, F.IsQuasicoherent → (∀ U : X.Opens, IsAffineOpen U → Module.Finite Γ(X, U) Γ(F, U)) → ((Scheme.Modules.pushforward f).obj F).IsQuasicoherent ∧ ∀ V : Y.Opens, IsAffineOpen V → Module.Finite Γ(Y, V) Γ((Scheme.Modules.pushforward f).obj F, V)) ∧ (IsAffineHom f → ∀ S : ShortComplex X.Modules, S.ShortExact → S.X₂.IsQuasicoherent → S.X₃.IsQuasicoherent → (S.map (Scheme.Modules.pushforward f)).ShortExact) ∧ (∀ G : Y.Modules, G.IsQuasicoherent → ((Scheme.Modules.pullback f).obj G).IsQuasicoherent ∧ ∀ (U : X.Opens) (V : Y.Opens), IsAffineOpen U → IsAffineOpen V → ∀ e : U ≤ f ⁻¹ᵁ V, letI

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RingKrullDimQuotientEqOfMemMinimalPrimes

theorem **ringKrullDim_eq_of_isIntegral** (R S : Type*) [CommRing R] [CommRing S] [Algebra R S] [Algebra.IsIntegral R S] [FaithfulSMul R S] : ringKrullDim S = ringKrullDim R

theorem **trdeg_eq_of_isAlgebraic_top** (R S A : Type u) [CommRing R] [Nontrivial R] [CommRing S] [CommRing A] [NoZeroDivisors A] [Algebra R S] [Algebra S A] [Algebra R A] [IsScalarTower R S A] [FaithfulSMul R S] [FaithfulSMul S A] [Algebra.IsAlgebraic S A] : Algebra.trdeg R A = Algebra.trdeg R S

theorem **trdeg_eq_of_isAlgebraic_bot** (R S A : Type u) [CommRing R] [Nontrivial R] [CommRing S] [CommRing A] [NoZeroDivisors A] [Algebra R S] [Algebra S A] [Algebra R A] [IsScalarTower R S A] [FaithfulSMul R S] [FaithfulSMul S A] [Algebra.IsAlgebraic R S] : Algebra.trdeg R A = Algebra.trdeg S A

theorem **exists_ringKrullDim_eq_and_trdeg_eq** (k B : Type u) [Field k] [CommRing B] [IsDomain B] [Algebra k B] [Algebra.FiniteType k B] : ∃ d : ℕ, ringKrullDim B = d ∧ Algebra.trdeg k B = d

theorem **ringKrullDim_eq_of_trdeg_eq** (k B : Type u) [Field k] [CommRing B] [IsDomain B] [Algebra k B] [Algebra.FiniteType k B] {d : ℕ} (h : Algebra.trdeg k B = d) : ringKrullDim B = d

theorem **quasiFiniteAt_of_isolated** (R S : Type*) [CommRing R] [CommRing S] [Algebra R S] [Algebra.FiniteType R S] (q : Ideal S) [q.IsPrime] (hmin : ∀ q' : Ideal S, q'.IsPrime → q'.under R = q.under R → q' ≤ q → q' = q) (hmax : ∀ q' : Ideal S, q'.IsPrime → q'.under R = q.under R → q ≤ q' → q' = q) : Algebra.QuasiFiniteAt R q

theorem **isAlgebraic_of_quasiFiniteAt** (R S : Type*) [CommRing R] [IsDomain R] [CommRing S] [IsDomain S] [Algebra R S] [FaithfulSMul R S] (q : Ideal S) [q.IsPrime] [Algebra.QuasiFiniteAt R q] : Algebra.IsAlgebraic R S

theorem **liesOver_maximalIdeal_of_mem_minimalPrimes** {q : Ideal A} (hq : q ∈ ((maximalIdeal O).map (algebraMap O A)).minimalPrimes) : q.LiesOver (maximalIdeal O)

theorem **map_maximalIdeal_eq_span_of_irreducible** {ϖ : O} (hϖ : Irreducible ϖ) : (maximalIdeal O).map (algebraMap O A) = Ideal.span {algebraMap O A ϖ}

theorem **height_eq_one_of_mem_minimalPrimes_map_maximalIdeal** {q : Ideal A} (hq : q ∈ ((maximalIdeal O).map (algebraMap O A)).minimalPrimes) : q.height = 1

theorem **trdeg_quotient_eq_of_mem_minimalPrimes** (k : Type u) [Field k] [Algebra O k] (hk : Function.Surjective (algebraMap O k)) {q : Ideal A} (hq : q ∈ ((maximalIdeal O).map (algebraMap O A)).minimalPrimes) [Algebra k (A ⧸ q)] [IsScalarTower O k (A ⧸ q)] : Algebra.trdeg k (A ⧸ q) = Algebra.trdeg O A

theorem **exists_trdeg_eq_natCast** : ∃ d : ℕ, Algebra.trdeg O A = d

theorem **ringKrullDim_quotient_eq_of_mem_minimalPrimes** {q : Ideal A} (hq : q ∈ ((maximalIdeal O).map (algebraMap O A)).minimalPrimes) {d : ℕ} (hd : Algebra.trdeg O A = d) : ringKrullDim (A ⧸ q) = d

theorem **trdeg_fractionRing_eq** (K F : Type u) [Field K] [Algebra O K] [IsFractionRing O K] [Field F] [Algebra A F] [IsFractionRing A F] [Algebra O F] [Algebra K F] [IsScalarTower O A F] [IsScalarTower O K F] : Algebra.trdeg K F = Algebra.trdeg O A

theorem **ringKrullDim_genericFibre_eq** (K : Type u) [Field K] [Algebra O K] [IsFractionRing O K] {d : ℕ} (hd : Algebra.trdeg O A = d) : ringKrullDim (K ⊗[O] A) = d

theorem **trdeg_residueField_eq_of_mem_minimalPrimes** (k : Type u) [Field k] [Algebra O k] (hk : Function.Surjective (algebraMap O k)) (q : Ideal A) [q.IsPrime] (hq : q ∈ ((maximalIdeal O).map (algebraMap O A)).minimalPrimes) [Algebra k q.ResidueField] [IsScalarTower O k q.ResidueField] : Algebra.trdeg k q.ResidueField = Algebra.trdeg O A

theorem **height_eq_one_and_ringKrullDim_eq_of_flat** (O K k : Type u) [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] [Field K] [Algebra O K] [IsFractionRing O K] [Field k] [Algebra O k] (hk : Function.Surjective (algebraMap O k)) (A F : Type u) [CommRing A] [IsDomain A] [Algebra O A] [Algebra.FiniteType O A] [Module.Flat O A] [Field F] [Algebra A F] [IsFractionRing A F] [Algebra O F] [Algebra K F] [IsScalarTower O A F] [IsScalarTower O K F] {ϖ : O} (hϖ : Irreducible ϖ) (q : Ideal A) [q.IsPrime] (hq : q ∈ (Ideal.span {algebraMap O A ϖ}).minimalPrimes) [Algebra k q.ResidueField] [IsScalarTower O k q.ResidueField] : q.height = 1 ∧ ∃ d : ℕ, ringKrullDim (A ⧸ q) = d ∧ Algebra.trdeg k q.ResidueField = d ∧ Algebra.trdeg K F = d ∧ ringKrullDim (K ⊗[O] A) = d

theorem **isLocalHom_algebraMap_localization_of_liesOver** (O : Type*) [CommRing O] [IsLocalRing O] {A : Type*} [CommRing A] [Algebra O A] (q : Ideal A) [q.IsPrime] [q.LiesOver (maximalIdeal O)] : IsLocalHom (algebraMap O (Localization.AtPrime q))

theorem **height_eq_one_and_ringKrullDim_eq_of_flat_canonical** (O : Type u) [CommRing O] [IsDomain O] [IsDiscreteValuationRing O] (A : Type u) [CommRing A] [IsDomain A] [Algebra O A] [Algebra.FiniteType O A] [Module.Flat O A] {ϖ : O} (hϖ : Irreducible ϖ) (q : Ideal A) [q.IsPrime] (hq : q ∈ (Ideal.span {algebraMap O A ϖ}).minimalPrimes) : haveI

theorem **topologicalKrullDim_eq_of_isEmbedding** {W X : Type*} [TopologicalSpace W] [TopologicalSpace X] {e : W → X} (he : IsEmbedding e) {Z : Set X} (hZ : Set.range e = Z) : topologicalKrullDim Z = topologicalKrullDim W

theorem **topologicalKrullDim_le_of_forall_exists_isOpen** {X : Type*} [TopologicalSpace X] {d : WithBot ℕ∞} (h : ∀ x : X, ∃ U : Set X, IsOpen U ∧ x ∈ U ∧ topologicalKrullDim U ≤ d) : topologicalKrullDim X ≤ d

theorem **topologicalKrullDim_eq_of_forall_exists_isOpen** {X : Type*} [TopologicalSpace X] {Z : Set X} (hZ : Z.Nonempty) {d : WithBot ℕ∞} (h : ∀ x ∈ Z, ∃ U : Set X, IsOpen U ∧ x ∈ U ∧ topologicalKrullDim (U ∩ Z : Set X) = d) : topologicalKrullDim Z = d

theorem **exists_affine_chart_trdeg_eq** (x : Y) : letI

theorem **topologicalKrullDim_preimage_genericPoint_eq** {d : ℕ} (hd : letI

theorem **mem_minimalPrimes_of_mem_irreducibleComponents** {S : Type*} [TopologicalSpace S] {i : S → Y} (hi : IsEmbedding i) (hr : Set.range i = f ⁻¹' {closedPoint O}) {Z : Set S} (hZ : Z ∈ irreducibleComponents S) {A : Type u} [CommRing A] [Algebra O A] {j : Spec (.of A) ⟶ Y} [IsOpenImmersion j] (hj : j ≫ f = Spec.map (CommRingCat.ofHom (algebraMap O A))) {q : PrimeSpectrum A} (hq : IsGenericPoint (j q) (i '' Z)) : q.asIdeal ∈ ((maximalIdeal O).map (algebraMap O A)).minimalPrimes

theorem **isClosed_image_of_mem_irreducibleComponents** {S : Type*} [TopologicalSpace S] {i : S → Y} (hi : IsEmbedding i) (hr : Set.range i = f ⁻¹' {closedPoint O}) {Z : Set S} (hZ : Z ∈ irreducibleComponents S) : IsClosed (i '' Z)

theorem **topologicalKrullDim_eq_of_mem_irreducibleComponents_of_trdeg_eq** {S : Type*} [TopologicalSpace S] {i : S → Y} (hi : IsEmbedding i) (hr : Set.range i = f ⁻¹' {closedPoint O}) {Z : Set S} (hZ : Z ∈ irreducibleComponents S) {d : ℕ} (hd : letI

theorem **ringKrullDim_stalk_eq_one_of_mem_irreducibleComponents** {S : Type*} [TopologicalSpace S] {i : S → Y} (hi : IsEmbedding i) (hr : Set.range i = f ⁻¹' {closedPoint O}) {Z : Set S} (hZ : Z ∈ irreducibleComponents S) {z : S} (hz : IsGenericPoint z Z) : ringKrullDim (Y.presheaf.stalk (i z)) = 1

theorem **exists_topologicalKrullDim_fibres_eq_of_isEmbedding** {S G : Type*} [TopologicalSpace S] [TopologicalSpace G] {i : S → Y} (hi : IsEmbedding i) (hr : Set.range i = f ⁻¹' {closedPoint O}) {i' : G → Y} (hi' : IsEmbedding i') (hr' : Set.range i' = f ⁻¹' {genericPoint

(Spec (.of O))}) : ∃ d : ℕ, topologicalKrullDim G = d ∧ ∀ Z ∈ irreducibleComponents S, topologicalKrullDim Z = d

theorem **topologicalKrullDim_component_fiber_eq** {Z : Set (f.fiber (closedPoint O))} (hZ : Z ∈ irreducibleComponents (f.fiber (closedPoint O))) : topologicalKrullDim Z = topologicalKrullDim (f.fiber (genericPoint (Spec (.of O))))

theorem **isEmbedding_and_range_eq_of_isPullback** {T P : Scheme.{u}} {g : T ⟶ Spec (.of O)} [IsPreimmersion g] {i : P ⟶ Y} {p : P ⟶ T} (h : IsPullback i p f g) : IsEmbedding i ∧ Set.range i = f ⁻¹' Set.range g

theorem **exists_topologicalKrullDim_fibres_eq_of_isPullback** (K k : Type u) [Field K] [Algebra O K] [IsFractionRing O K] [Field k] [Algebra O k] (hk : Function.Surjective (algebraMap O k)) {Y₀ Y₁ : Scheme.{u}} {i₀ : Y₀ ⟶ Y} {p₀ : Y₀ ⟶ Spec (.of k)} {i₁ : Y₁ ⟶ Y} {p₁ : Y₁ ⟶ Spec (.of K)} (h₀ : IsPullback i₀ p₀ f (Spec.map (CommRingCat.ofHom (algebraMap O k)))) (h₁ : IsPullback i₁ p₁ f (Spec.map (CommRingCat.ofHom (algebraMap O K)))) : ∃ d : ℕ, topologicalKrullDim Y₁ = d ∧ ∀ Z ∈ irreducibleComponents Y₀, topologicalKrullDim Z = d ∧ ∀ z, IsGenericPoint z Z → ringKrullDim (Y.presheaf.stalk (i₀ z)) = 1

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.RootCharOfAddNatCast

theorem **pow_val_add** {x : B} (hx : x ^ n = 1) (a b : ZMod n) : x ^ (a + b).val = x ^ a.val * x ^ b.val

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SecCubeExactAtZeroAndOne

theorem **secCube_exactAt_zero_and_one** (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (hD : ⊔ i, D i = ⊤) : (cubeComplex (secCube f M D)).ExactAt 0 ∧ (cubeComplex (secCube f M D)).ExactAt 1

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SecOnBasicOpenLocalization

theorem **secOn_basicOpen_localization** : ∀ (Y : Scheme.{0}) (M : LB Y) (V W : Y.Opens) (a : Γ(Y, V)) (h : W ≤ V), M.IsLB → IsCompact (V : Set Y) → IsQuasiSeparated (V : Set Y) → W = Y.basicOpen a → (∀ x : SecOn M V, SecOn.restrict M h x = 0 → ∃ k : ℕ, a ^ k • x = 0) ∧ ∀ y : SecOn M W, ∃ (k : ℕ) (x : SecOn M V), SecOn.restrict M h x = rs Y h a ^ k • y

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SecOnFinitePresentationProjective

theorem **SecOn.isLocalizedModule_chartBasicOpen** (hM : M.IsLB) (hV : IsAffineOpen V) (i : M.ι) (a : Γ(Y, V)) (h : Y.basicOpen a ≤ M.U i) : IsLocalizedModule (Submonoid.powers a) (SecOn.chartBasicOpen hM i a h)

theorem **secOn_finitePresentation_projective** : ∀ (Y : Scheme.{0}) (M : LB Y) (V : Y.Opens), M.IsLB → IsAffineOpen V → Module.FinitePresentation Γ(Y, V) (SecOn M V) ∧ Module.Projective Γ(Y, V) (SecOn M V)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SectionsTranslationDifferenceEqZero

theorem **hasGen_of_iso_tw** (M : LB X) (p q m : Y ⟶ X) (h : ∃ w : LB.Iso (M.pull m) (tw (M.pull p) (M.pull q) 1), w.IsGen) : HasGen M (K

theorem **gs_sec** {M : LB X} (σ : LB.Sec M) (r : Z ⟶ X) (z : Z) {i j : M.ι} (hi : r z ∈ M.U i) (hj : r z ∈ M.U j) : gs r z (M.U i) (σ.f i) = (gm r z (M.U i ⊓ M.U j) (M.g i j)).val * gs r z (M.U j) (σ.f j)

theorem **isUnit_gs** (r : Z ⟶ X) (z : Z) {U : X.Opens} (a : Γ(X, U)) (h : r z ∈ X.basicOpen a) : IsUnit (gs r z U a)

theorem **exists_global** {M : LB X} (hM : M.IsLB) (ν : X ⟶ X) (H : HasGen M (K

theorem **gen_of_sec_ne_zero** [IsIntegral X] (hX : IsField Γ(X, ⊤)) {M : LB X} (hM : M.IsLB) (ν : X ⟶ X) (hν : Function.Surjective ν) (H : HasGen M (K

theorem **sec_translDiff_eq_zero** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (x : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ P.A), x ≠ η[P.A] → ∀ (σ : LB.Sec (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SectionsTranslationDifferenceEqZeroStatement

theorem **_root_.GlobalTheory.T.sections_translationDifference_eq_zero** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (x : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ P.A), x ≠ η[P.A] → ∀ (σ : LB.Sec (LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SerreVanishingH1Twist

theorem **exists_h1Zero_tw_of_veryAmpleBy** (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y) (hH : H.IsLB) (hM : M.IsLB) (hs : VeryAmpleBy f H s) : ∃ t₀ : ℤ, ∀ t, t₀ ≤ t → FirstCohomologyVanishes (tw M H t)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SerreVanishingH1TwistStatement

theorem **_root_.GlobalTheory.T.serre_vanishing_h1_twist** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) [IsNoetherianRing R] (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∃ t₀ : ℤ, ∀ t, t₀ ≤ t → FirstCohomologyVanishes (tw M H t)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SigmaSplitSymmInl

@[simp] theorem **innerAut_apply** (u : Aˣ) (x : A) : innerAut R u x = (u : A) * x * ((u⁻¹ : Aˣ) : A)

theorem **EntriesBounded.mono** {M : Matrix n n K} {c d : ℝ} (h : EntriesBounded M c) (hcd : c ≤ d) : EntriesBounded M d

theorem **EntriesBounded.mul** [IsUltrametricDist K] [Fintype n] {M N : Matrix n n K} {c d : ℝ} (hM : EntriesBounded M c) (hN : EntriesBounded N d) : EntriesBounded (M * N) (c * d)

@[simp] theorem **dropLast_apply** (x : Fin (d + 1) → K) (i : Fin d) : dropLast K d x i = x i.castSucc

theorem **sigmaSplit_self** (j : ι) (x : F j) : sigmaSplit F j ⟨j, x⟩ = Sum.inl x

theorem **sigmaSplit_of_ne** {i j : ι} (h : i ≠ j) (x : F i) : sigmaSplit F j ⟨i, x⟩ = Sum.inr ⟨⟨i, h⟩, x⟩

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SignedLengthEqLogOrdFracDet

theorem **ord_finprod** {R : Type*} [CommRing R] [IsDomain R] {ι : Type*} (s : Finset ι) (c : ι → R) (hc : ∀ i, c i ≠ 0) : Ring.ord R (∏ i ∈ s, c i) = ∑ i ∈ s, Ring.ord R (c i)

theorem **length_coker** {R : Type*} [CommRing R] [IsDomain R] [IsPrincipalIdealRing R] {M : Type*} [AddCommGroup M] [Module R M] {ι : Type*} [Fintype ι] (b : Basis ι R M) (φ : M →ₗ[R] M) (hφ : Function.Injective φ) : Module.length R (M ⧸ LinearMap.range φ) = Ring.ord R (LinearMap.det φ)

theorem **relLength_chain** {R V : Type*} [Ring R] [AddCommGroup V] [Module R V] {A D C : Submodule R V} (hCD : C ≤ D) (hDA : D ≤ A) : relLength A C = relLength A D + relLength D C

theorem **relLength_span_columns** {R : Type*} [CommRing R] [IsDomain R] [IsPrincipalIdealRing R] {V : Type*} [AddCommGroup V] [Module R V] {ι : Type*} [Fintype ι] [DecidableEq ι] {A : Submodule R V} (a : Basis ι R A) (h : Matrix ι ι R) (hh : h.det ≠ 0) (v : ι → V) (hv : ∀ j, v j = ∑ i, h i j • (a i : V)) : Submodule.span R (Set.range v) ≤ A ∧ relLength A (Submodule.span R (Set.range v)) = Ring.ord R h.det

theorem **ordFrac_algebraMap_eq_exp** {R : Type*} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {K : Type*} [Field K] [Algebra R K] [IsFractionRing R K] {x : R} (hx : x ≠ 0) {n : ℕ} (hn : Ring.ord R x = n) : Ring.ordFrac R (algebraMap R K x) = WithZero.exp (n : ℤ)

theorem **signedLength_eq_log_ordFrac_det** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {K : Type u_2} [Field K] [Algebra R K] [IsFractionRing R K] {V : Type u_3} [AddCommGroup V] [Module K V] [Module R V] [IsScalarTower R K V] {ι : Type u_4} [Fintype ι] [DecidableEq ι] {A B : Submodule R V} (a : Module.Basis ι R A) (b : Module.Basis ι R B) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, (b j : V) = ∑ i, g i j • (a i : V)) : LengthCommensurable A B ∧ signedLength A B = WithZero.log (Ring.ordFrac R g.det)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SignedLengthMapEqLogOrdFracDet

theorem **ord_prod** {ι : Type*} (s : Finset ι) (a : ι → R) (ha : ∀ i, a i ≠ 0) : Ring.ord R (∏ i ∈ s, a i) = ∑ i ∈ s, Ring.ord R (a i)

theorem **length_coker_eq_ord_det** [IsPrincipalIdealRing R] {M : Type*} [AddCommGroup M] [Module R M] [Module.Free R M] [Module.Finite R M] (f : M →ₗ[R] M) (hf : Function.Injective f) : Module.length R (M ⧸ LinearMap.range f) = Ring.ord R (LinearMap.det f)

theorem **relLength_chain** {A B C : Submodule R V} (hBA : B ≤ A) (hCB : C ≤ B) : relLength A C = relLength A B + relLength B C

theorem **signedLength_of_common** {A B C : Submodule R V} (hA : C ≤ A) (hB : C ≤ B) {m n : ℕ} (hm : relLength A C = m) (hn : relLength B C = n) : LengthCommensurable A B ∧ signedLength A B = (m : ℤ) − n

theorem **exists_relLength_map** (A : Submodule R V) [Submodule.IsLattice K A] (φ : V →ₗ[K] V) (hφ : LinearMap.det φ ≠ 0) (hA : ∀ x ∈ A, φ x ∈ A) : ∃ n : ℕ, relLength A (A.map (φ.restrictScalars R)) = n ∧ Ring.ordFrac R (LinearMap.det φ) = WithZero.exp (n : ℤ)

theorem **exists_smul_preserves** (A : Submodule R V) [Submodule.IsLattice K A] (g : V →ₗ[K] V) : ∃ r : R, r ≠ 0 ∧ ∀ x ∈ A, r • g x ∈ A

theorem **signedLength_map_eq_log_ordFrac_det** {R : Type u_1} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {K : Type u_2} [Field K] [Algebra R K] [IsFractionRing R K] {V : Type u_3} [AddCommGroup V] [Module K V] [Module R V] [IsScalarTower R K V] (A : Submodule R V) [Submodule.IsLattice K A] (g : V →ₗ[K] V) (hg : LinearMap.det g ≠ 0) : LengthCommensurable A (A.map (g.restrictScalars R)) ∧ signedLength A (A.map (g.restrictScalars R)) = WithZero.log (Ring.ordFrac R (LinearMap.det g))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SignedLengthOfCommonLower

theorem **relLength_bot_add** {A D : Submodule S V} (h : D ≤ A) : relLength A ⊥ = relLength A D + relLength D ⊥

theorem **relLength_add** {A C D : Submodule S V} (hCD : C ≤ D) (hDA : D ≤ A) : relLength A C = relLength A D + relLength D C

theorem **signedLength_of_common_lower** : ∀ {S : Type u_1} {V : Type u_2} [inst : Ring S] [inst_1 : AddCommGroup V] [inst_2 : Module S V] {A B C : Submodule S V}, C ≤ A → C ≤ B → ∀ {m n : ℕ}, GlobalTheory.relLength A C = (↑m : ℕ∞) → GlobalTheory.relLength B C = (↑n : ℕ∞) → GlobalTheory.LengthCommensurable A B ∧ GlobalTheory.signedLength A B = (↑m : ℤ) − (↑n : ℤ)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SignedLengthSpanEqOrdFracDet

theorem **ord_prod** {R : Type*} [CommRing R] [IsDomain R] {ι : Type*} (s : Finset ι) (c : ι → R) (hc : ∀ i, c i ≠ 0) : Ring.ord R (∏ i ∈ s, c i) = ∑ i ∈ s, Ring.ord R (c i)

theorem **length_coker** {R : Type*} [CommRing R] [IsDomain R] [IsPrincipalIdealRing R] {M : Type*} [AddCommGroup M] [Module R M] {ι : Type*} [Fintype ι] [DecidableEq ι] (e₀ : Basis ι R M) (f : M →ₗ[R] M) (hf : Function.Injective f) : Module.length R (M ⧸ LinearMap.range f) = Ring.ord R (LinearMap.det f)

theorem **relLength_chain** {R V : Type*} [Ring R] [AddCommGroup V] [Module R V] {A D C : Submodule R V} (hD : D ≤ A) (hC : C ≤ D) : relLength A C = relLength A D + relLength D C

theorem **signedLength_of_common** {R V : Type*} [Ring R] [AddCommGroup V] [Module R V] {A B C : Submodule R V} (hA : C ≤ A) (hB : C ≤ B) {m n : ℕ} (hm : relLength A C = m) (hn : relLength B C = n) : LengthCommensurable A B ∧ signedLength A B = (m : ℤ) − n

theorem **relLength_span** {R : Type*} [CommRing R] [IsDomain R] [IsPrincipalIdealRing R] {V : Type*} [AddCommGroup V] [Module R V] {ι : Type*} [Fintype ι] [DecidableEq ι] {a c : ι → V} (ha : LinearIndependent R a) (P : Matrix ι ι R) (hP : P.det ≠ 0) (hc : ∀ j, c j = ∑ i, P i j • a i) : span R (Set.range c) ≤ span R (Set.range a) ∧ relLength (span R (Set.range a)) (span R (Set.range c)) = Ring.ord R P.det

theorem **signedLength_span_eq_ordFrac_det** {R : Type*} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] {K : Type*} [Field K] [Algebra R K] [IsFractionRing R K] {V : Type*} [AddCommGroup V] [Module K V] [Module R V] [IsScalarTower R K V] {ι : Type*} [Fintype ι] [DecidableEq ι] {a b : ι → V} (ha : LinearIndependent K a) (g : Matrix ι ι K) (hg : g.det ≠ 0) (hb : ∀ j, b j = ∑ i, g i j • a i) : LengthCommensurable (span R (Set.range a)) (span R (Set.range b)) ∧ Ring.ordFrac R g.det = WithZero.exp (signedLength (span R (Set.range a)) (span R (Set.range b)))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SpanLaurentMonomialEqHomLoc

theorem **span_laurentMonomial_eq_homLoc** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (s : Finset ι) (n : ℤ) (v : {a : ι → ℤ // ∑ i, a i = n ∧ ∀ i, i ∉ s → 0 ≤ a i} → LocalizedModule.Away (xProd R s) (MvPolynomial ι R)), (∀ a (k : ℕ) (m : ι →₀ ℕ), (∀ i, (m i : ℤ) = a.1 i + if i ∈ s then (k : ℤ) else 0) → v a = awayFrac (MvPolynomial ι R) (xProd R s) k (monomial m 1)) → Submodule.span R (Set.range v) = homLoc (polyGrading ι R) s n

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SpanSingletonEqTopIffIsUnitOfIsBaseChange

theorem **finite** [Module.Finite A' (A' ⊗[A] M)] : Module.Finite A M

theorem **flat** [Module.Flat A' (A' ⊗[A] M)] : Module.Flat A M

theorem **fg_of_baseChange** {N : Submodule A M} (h : (N.baseChange A').FG) : N.FG

theorem **finitePresentation** [Module.FinitePresentation A' (A' ⊗[A] M)] : Module.FinitePresentation A M

theorem **projective** [Module.Finite A' (A' ⊗[A] M)] [Module.Projective A' (A' ⊗[A] M)] : Module.Projective A M

theorem **rankAtStalk_eq** [Module.Finite A' (A' ⊗[A] M)] [Module.Flat A' (A' ⊗[A] M)] {r : ℕ} (h : ∀ q : PrimeSpectrum A', Module.rankAtStalk (A' ⊗[A] M) q = r) (p : PrimeSpectrum A) : Module.rankAtStalk M p = r

theorem **one_tmul_injective** : Function.Injective fun m : M ↦ (1 : A') ⊗ₜ[A] m

theorem **linearIndependent_iff** {ι : Type*} (v : ι → M) : LinearIndependent A v ↔ LinearIndependent A' fun i ↦ (1 : A') ⊗ₜ[A] v i

theorem **algebraMap_dvd_iff** (a b : A) : algebraMap A A' a ∣ algebraMap A A' b ↔ a ∣ b

theorem **finite_of_isBaseChange** [Module.Finite A' M'] : Module.Finite A M

theorem **projective_of_isBaseChange** [Module.Finite A' M'] [Module.Projective A' M'] : Module.Projective A M

theorem **germ_smul** (U : X.Opens) (x : X) (hx : x ∈ U) (r : Γ(X, U)) (m : Γ(F, U)) : F.presheaf.germ U x hx (r • m) = X.presheaf.germ U x hx r • F.presheaf.germ U x hx m

theorem **injective_smul_of_stalk** {V : X.Opens} (hV : V ≤ U) (h : ∀ (x : X) (hx : x ∈ V), Function.Injective fun a : X.presheaf.stalk x ↦ a • F.presheaf.germ U x (hV hx) s) : Function.Injective fun a : Γ(X, V) ↦ a • F.presheaf.map (homOfLE hV).op s

theorem **bijective_smul_of_stalk** (h : ∀ (x : X) (hx : x ∈ U), Function.Bijective fun a : X.presheaf.stalk x ↦ a • F.presheaf.germ U x hx s) {V : X.Opens} (hV : V ≤ U) : Function.Bijective fun a : Γ(X, V) ↦ a • F.presheaf.map (homOfLE hV).op s

theorem **bijective_smul_stalk** (h : ∀ (V : X.Opens) (hV : V ≤ U), Function.Bijective fun a : Γ(X, V) ↦ a • F.presheaf.map (homOfLE hV).op s) (x : X) (hx : x ∈ U) : Function.Bijective fun a : X.presheaf.stalk x ↦ a • F.presheaf.germ U x hx s

theorem **sectionHom_app** (W : (U : Scheme.{u}).Opens) (a : Γ(X, U.ι ''ᵁ W)) : ((sectionHom s).app W a : Γ(F, U.ι ''ᵁ W)) = a • F.presheaf.map (homOfLE (U.ι_image_le W)).op s

theorem **isIso_sectionHom_iff** : IsIso (sectionHom s) ↔ ∀ (x : X) (hx : x ∈ U), Function.Bijective fun a : X.presheaf.stalk x ↦ a • F.presheaf.germ U x hx s

theorem **freeIso_hom** (h : ∀ (x : X) (hx : x ∈ U), Function.Bijective fun a : X.presheaf.stalk x ↦ a • F.presheaf.germ U x hx s) : (freeIso h).hom = sectionHom s

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.StructureHomApply

theorem **map_structureHom** (s : X ⟶ Spec (CommRingCat.of R)) {U V : X.Opens} (h : V ≤ U) (r : R) : (X.presheaf.map (homOfLE h).op).hom (structureHom s U r) = structureHom s V r

theorem **equiv_apply** (U : X.Opens) (a : Sections s U) : equiv s U a = a

theorem **restrict_restrict** {U V W : X.Opens} (h : V ≤ U) (k : W ≤ V) (a : Sections s U) : restrict s k (restrict s h a) = restrict s (k.trans h) a

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SubalgebraEqBotOfIsIntegralOfIsIntegrallyClosed

theorem **_root_.GlobalTheory.integralClosure_restrictScalars_of_isIntegral** (A A' B : Type*) [CommRing A] [CommRing A'] [CommRing B] [Algebra A A'] [Algebra A' B] [Algebra A B] [IsScalarTower A A' B] [Algebra.IsIntegral A A'] : (integralClosure A' B).restrictScalars A = integralClosure A B

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SubsingletonOfPoints

theorem **subsingleton_of_points** (K : Type) [Field K] [IsAlgClosed K] (R : Type) [CommRing R] (s : Spec (CommRingCat.of R) ⟶ Spec (CommRingCat.of K)) [LocallyOfFiniteType s] (H : Type) [AddCommGroup H] [Module R H] (h : ∀ χ : R →+* K, Spec.map (CommRingCat.ofHom χ) ≫ s = 𝟙 _ → ∃ r : R, χ r ≠ 0 ∧ ∀ z : H, r • z = 0) : Subsingleton H

theorem **acyclic_away** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (R : Type) [CommRing R] (t : Spec (CommRingCat.of R) ⟶ P.A.left) (Y : Scheme.{0}) (π : Y ⟶ (P.A ⊗ P.A).left) (f : Y ⟶ Spec (CommRingCat.of R)), IsPullback π f (fst P.A P.A).left t → IsOpenImmersion t → (∀ p q, t p ≠ η[P.A].left q) → ∀ (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens), (∀ i, IsAffineOpen (D i)) → ⊔ i, D i = ⊤ → (cechLB f ((translDiff P.A P.L (𝟙 P.A)).pull π) D).Acyclic

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SubsingletonOfSheafPullback

lemma **shortExact_map_sheafPullback** {S : ShortComplex (X.Sheaf AddCommGrpCat.{u})} (hS : S.ShortExact) [(hj.sheafPullback AddCommGrpCat.{u}).PreservesZeroMorphisms] : (S.map (hj.sheafPullback AddCommGrpCat.{u})).ShortExact

lemma **subsingleton_of_sheafPullback** (V : Opens Y) (n : ℕ) (F : X.Sheaf AddCommGrpCat.{u}) (h : Subsingleton (Sheaf.H'.{u} ((hj.sheafPullback AddCommGrpCat.{u}).obj F) (n + 1) V)) : Subsingleton (Sheaf.H'.{u} F (n + 1) (hj.functor.obj V))

lemma **subsingleton_of_restrict** (j : Y ⟶ X) [IsOpenImmersion j] (F : X.Modules) (V : Y.Opens) (n : ℕ) (h : Subsingleton (cohOn (F.restrict j) V (n + 1))) : Subsingleton (cohOn F (j ''ᵁ V) (n + 1))

lemma **map_toInterModule_eq_zero** (F : X.Modules) (W V : X.Opens) (n : ℕ) (α : cohOn F V n) (h : SheafLES.map.{u} ((Kempf.toInter AddCommGrpCat.{u} W).app ((toAbSheaf X).obj F)) n V (cohOn.toH' F V n α) = 0) : cohOn.map ℤ (toInterModule F W) V n α = 0

lemma **mono_lift_toInterModule** (F : X.Modules) {ι : Type u} [Finite ι] (W : ι → X.Opens) (hW : ⊤ ≤ iSup W) : Mono (Pi.lift fun i ↦ toInterModule F (W i))

lemma **subsingleton_top** [IsAffine X] (n : ℕ) (ih : ∀ q, 0 < q → q < n + 1 → ∀ (G : X.Modules) [G.IsQuasicoherent] (U : X.Opens), IsAffineOpen U → Subsingleton (cohOn G U q)) (F : X.Modules) [F.IsQuasicoherent] : Subsingleton (cohOn F ⊤ (n + 1))

theorem **cohOn.subsingleton_of_isAffineOpen** {X : Scheme.{u}} (F : X.Modules) [F.IsQuasicoherent] {U : X.Opens} (hU : IsAffineOpen U) {n : ℕ} (hn : 0 < n) : Subsingleton (cohOn F U n)

theorem **coh.subsingleton_of_isAffine** {X : Scheme.{u}} [IsAffine X] (F : X.Modules) [F.IsQuasicoherent] {n : ℕ} (hn : 0 < n) : Subsingleton (coh F n)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SumNegOnePowFinrankEqZero

theorem **finite_of_isNoetherian** (h : Function.Exact f g) [Module.Finite R A] [IsNoetherian R C] : Module.Finite R B

theorem **finite** [IsNoetherianRing R] (h : Function.Exact f g) [Module.Finite R A] [Module.Finite R C] : Module.Finite R B

theorem **finrank_eq_range_add_range** {A B C : Type*} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] [Module k A] [Module k B] [Module k C] {f : A →ₗ[k] B} {g : B →ₗ[k] C} (h : Function.Exact f g) [Module.Finite k B] : finrank k B = finrank k (LinearMap.range f) + finrank k (LinearMap.range g)

theorem **finrank_le** {A B C : Type*} [AddCommGroup A] [AddCommGroup B] [AddCommGroup C] [Module k A] [Module k B] [Module k C] {f : A →ₗ[k] B} {g : B →ₗ[k] C} (h : Function.Exact f g) [Module.Finite k A] [Module.Finite k C] : finrank k B ≤ finrank k A + finrank k C

theorem **sum_neg_one_pow_finrank** {V : ℕ → Type*} [∀ i, AddCommGroup (V i)] [∀ i, Module k (V i)] (f : ∀ i, V i →ₗ[k] V (i + 1)) (N : ℕ) (hV : ∀ i < N, Module.Finite k (V i)) (h : ∀ i < N, Function.Exact (f i) (f (i + 1))) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (V i) = finrank k (LinearMap.ker (f 0)) - (-1) ^ N * finrank k (LinearMap.ker (f N))

theorem **sum_neg_one_pow_finrank_eq_zero** {V : ℕ → Type*} [∀ i, AddCommGroup (V i)] [∀ i, Module k (V i)] (f : ∀ i, V i →ₗ[k] V (i + 1)) (N : ℕ) (hV : ∀ i < N, Module.Finite k (V i)) (h₀ : Function.Injective (f 0)) (h : ∀ i < N, Function.Exact (f i) (f (i + 1))) (hN : Subsingleton (V N)) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (V i) = 0

theorem **sum_neg_one_pow_finrank₃** (N : ℕ) (hA : ∀ i < N, Module.Finite k (A i)) (hB : ∀ i < N, Module.Finite k (B i)) (hC : ∀ i < N, Module.Finite k (C i)) (h₂ : ∀ i < N, Function.Exact (f i) (g i)) (h₃ : ∀ i < N, Function.Exact (g i) (δ i)) (h₁ : ∀ i < N, Function.Exact (δ i) (f (i + 1))) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * (finrank k (A i) - finrank k (B i) + finrank k (C i)) = finrank k (LinearMap.ker (f 0)) - (-1) ^ N * finrank k (LinearMap.ker (f N))

theorem **sum_neg_one_pow_finrank_eq_add** (N : ℕ) (hA : ∀ i < N, Module.Finite k (A i)) (hB : ∀ i < N, Module.Finite k (B i)) (hC : ∀ i < N, Module.Finite k (C i)) (h₀ : Function.Injective (f 0)) (h₂ : ∀ i < N, Function.Exact (f i) (g i)) (h₃ : ∀ i < N, Function.Exact (g i) (δ i)) (h₁ : ∀ i < N, Function.Exact (δ i) (f (i + 1))) (hN : Function.Injective (f N)) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (B i) = ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (A i) + ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (C i)

theorem **shortExact_toAbSheaf** {S : ShortComplex X.Modules} (hS : S.ShortExact) : (S.map (toAbSheaf X)).ShortExact

lemma **toH'_δ** (x : cohOn S.X₃ U n) : toH' S.X₁ U (n + 1) (δ R hS U n x) = SheafLES.δ.{u} (shortExact_toAbSheaf hS) n U (toH' S.X₃ U n x)

lemma **δ_eq_δ** (R' : Type t) [CommRing R'] [Algebra R' Γ(X, ⊤)] (x : cohOn S.X₃ U n) : δ R hS U n x = δ R' hS U n x

theorem **exact₂** : Function.Exact (map R S.f U n) (map R S.g U n)

theorem **exact₃** : Function.Exact (map R S.g U n) (δ R hS U n)

theorem **exact₁** : Function.Exact (δ R hS U n) (map R S.f U (n + 1))

theorem **map_zero_injective_of_mono** {F G : X.Modules} (φ : F ⟶ G) [Mono φ] : Function.Injective (map R φ U 0)

theorem **injective_zero** : Function.Injective (map R S.f U 0)

@[simp] lemma **δ_map** (x : cohOn S.X₂ U n) : δ R hS U n (map R S.g U n x) = 0

@[simp] lemma **map_δ** (x : cohOn S.X₃ U n) : map R S.f U (n + 1) (δ R hS U n x) = 0

theorem **δ_naturality** {S' : ShortComplex X.Modules} (hS' : S'.ShortExact) (φ : S ⟶ S') (x : cohOn S.X₃ U n) : map R φ.τ₁ U (n + 1) (δ R hS U n x) = δ R hS' U n (map R φ.τ₃ U n x)

theorem **res_δ** {V : X.Opens} (h : V ≤ U) (x : cohOn S.X₃ U n) : res R S.X₁ h (n + 1) (δ R hS U n x) = δ R hS V n (res R S.X₃ h n x)

theorem **map_g_surjective** (h : Subsingleton (cohOn S.X₁ U (n + 1))) : Function.Surjective (map R S.g U n)

theorem **map_f_injective** (h : Subsingleton (cohOn S.X₃ U n)) : Function.Injective (map R S.f U (n + 1))

theorem **app_surjective** (h : Subsingleton (cohOn S.X₁ U 1)) : Function.Surjective (S.g.app U)

theorem **subsingleton₂** (h₁ : Subsingleton (cohOn S.X₁ U n)) (h₃ : Subsingleton (cohOn S.X₃ U n)) : Subsingleton (cohOn S.X₂ U n)

theorem **subsingleton₃** (h₂ : Subsingleton (cohOn S.X₂ U n)) (h₁ : Subsingleton (cohOn S.X₁ U (n + 1))) : Subsingleton (cohOn S.X₃ U n)

theorem **subsingleton₁** (h₃ : Subsingleton (cohOn S.X₃ U n)) (h₂ : Subsingleton (cohOn S.X₂ U (n + 1))) : Subsingleton (cohOn S.X₁ U (n + 1))

theorem **subsingleton₁_zero** (h₂ : Subsingleton (cohOn S.X₂ U 0)) : Subsingleton (cohOn S.X₁ U 0)

theorem **finite₂** (h₁ : Module.Finite R (cohOn S.X₁ U n)) (h₃ : Module.Finite R (cohOn S.X₃ U n)) : Module.Finite R (cohOn S.X₂ U n)

theorem **finite₃** (h₂ : Module.Finite R (cohOn S.X₂ U n)) (h₁ : Module.Finite R (cohOn S.X₁ U (n + 1))) : Module.Finite R (cohOn S.X₃ U n)

theorem **finite₁** (h₃ : Module.Finite R (cohOn S.X₃ U n)) (h₂ : Module.Finite R (cohOn S.X₂ U (n + 1))) : Module.Finite R (cohOn S.X₁ U (n + 1))

theorem **finite₁_zero** (h₂ : Module.Finite R (cohOn S.X₂ U 0)) : Module.Finite R (cohOn S.X₁ U 0)

theorem **forall_finite₁** (h₂ : ∀ n, Module.Finite R (cohOn S.X₂ U n)) (h₃ : ∀ n, Module.Finite R (cohOn S.X₃ U n)) : ∀ n, Module.Finite R (cohOn S.X₁ U n) | 0 => finite₁_zero R hS U (h₂ 0) | n + 1 => finite₁ R hS U n (h₃ n) (h₂ (n + 1)) theorem forall_finite₂ (h₁ : ∀ n, Module.Finite R (cohOn S.X₁ U n)) (h₃ : ∀ n, Module.Finite R (cohOn S.X₃ U n)) : ∀ n, Module.Finite R (cohOn S.X₂ U n)

theorem **forall_finite₃** (h₁ : ∀ n, Module.Finite R (cohOn S.X₁ U n)) (h₂ : ∀ n, Module.Finite R (cohOn S.X₂ U n)) : ∀ n, Module.Finite R (cohOn S.X₃ U n)

theorem **finrank₂_le** (h₁ : Module.Finite k (cohOn S.X₁ U n)) (h₃ : Module.Finite k (cohOn S.X₃ U n)) : finrank k (cohOn S.X₂ U n) ≤ finrank k (cohOn S.X₁ U n) + finrank k (cohOn S.X₃ U n)

theorem **finrank₃_le** (h₂ : Module.Finite k (cohOn S.X₂ U n)) (h₁ : Module.Finite k (cohOn S.X₁ U (n + 1))) : finrank k (cohOn S.X₃ U n) ≤ finrank k (cohOn S.X₂ U n) + finrank k (cohOn S.X₁ U (n + 1))

theorem **finrank₁_le** (h₃ : Module.Finite k (cohOn S.X₃ U n)) (h₂ : Module.Finite k (cohOn S.X₂ U (n + 1))) : finrank k (cohOn S.X₁ U (n + 1)) ≤ finrank k (cohOn S.X₃ U n) + finrank k (cohOn S.X₂ U (n + 1))

theorem **finrank₁_zero_le** (h₂ : Module.Finite k (cohOn S.X₂ U 0)) : finrank k (cohOn S.X₁ U 0) ≤ finrank k (cohOn S.X₂ U 0)

theorem **sum_neg_one_pow_finrank** (N : ℕ) (h₁ : ∀ i < N, Module.Finite k (cohOn S.X₁ U i)) (h₂ : ∀ i < N, Module.Finite k (cohOn S.X₂ U i)) (h₃ : ∀ i < N, Module.Finite k (cohOn S.X₃ U i)) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * (finrank k (cohOn S.X₁ U i) - finrank k (cohOn S.X₂ U i) + finrank k (cohOn S.X₃ U i)) = -(-1) ^ N * finrank k (LinearMap.ker (map k S.f U N))

theorem **sum_neg_one_pow_finrank_eq_add** (N : ℕ) (h₁ : ∀ i < N, Module.Finite k (cohOn S.X₁ U i)) (h₃ : ∀ i < N, Module.Finite k (cohOn S.X₃ U i)) (hN : Function.Injective (map k S.f U N)) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (cohOn S.X₂ U i) = ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (cohOn S.X₁ U i) + ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * finrank k (cohOn S.X₃ U i)

theorem **hdim₂_le** (n : ℕ) (h₁ : Module.Finite k (coh S.X₁ n)) (h₃ : Module.Finite k (coh S.X₃ n)) : hdim k S.X₂ n ≤ hdim k S.X₁ n + hdim k S.X₃ n

theorem **hdim₃_le** (n : ℕ) (h₂ : Module.Finite k (coh S.X₂ n)) (h₁ : Module.Finite k (coh S.X₁ (n + 1))) : hdim k S.X₃ n ≤ hdim k S.X₂ n + hdim k S.X₁ (n + 1)

theorem **hdim₁_le** (n : ℕ) (h₃ : Module.Finite k (coh S.X₃ n)) (h₂ : Module.Finite k (coh S.X₂ (n + 1))) : hdim k S.X₁ (n + 1) ≤ hdim k S.X₃ n + hdim k S.X₂ (n + 1)

theorem **hdim₁_zero_le** (h₂ : Module.Finite k (coh S.X₂ 0)) : hdim k S.X₁ 0 ≤ hdim k S.X₂ 0

theorem **sum_neg_one_pow_hdim_eq_add** (N : ℕ) (h₁ : ∀ i < N, Module.Finite k (coh S.X₁ i)) (h₃ : ∀ i < N, Module.Finite k (coh S.X₃ i)) (hN : Subsingleton (coh S.X₁ N)) : ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * hdim k S.X₂ i = ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * hdim k S.X₁ i + ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * hdim k S.X₃ i

theorem **finsum_neg_one_pow_hdim_eq_sum** (F : X.Modules) (N : ℕ) (hN : ∀ i, N ≤ i → Subsingleton (coh F i)) : ∑ᶠ i, (-1 : ℤ) ^ i * hdim k F i = ∑ i ∈ Finset.range N, (-1 : ℤ) ^ i * hdim k F i

theorem **finsum_neg_one_pow_hdim_eq_add** (h₁ : ∀ i, Module.Finite k (coh S.X₁ i)) (h₃ : ∀ i, Module.Finite k (coh S.X₃ i)) (hv₁ : ∃ N, ∀ i, N ≤ i → Subsingleton (coh S.X₁ i)) (hv₃ : ∃ N, ∀ i, N ≤ i → Subsingleton (coh S.X₃ i)) : ∑ᶠ i, (-1 : ℤ) ^ i * hdim k S.X₂ i = ∑ᶠ i, (-1 : ℤ) ^ i * hdim k S.X₁ i + ∑ᶠ i, (-1 : ℤ) ^ i * hdim k S.X₃ i

theorem **linear_longExact** {S : ShortComplex X.Modules} (hS : S.ShortExact) : (∀ (R : Type v) [CommRing R] [Algebra R Γ(X, ⊤)] (U : X.Opens), (∃ d : ∀ n, cohOn S.X₃ U n →ₗ[R] cohOn S.X₁ U (n + 1), (∀ n x, cohOn.toH' S.X₁ U (n + 1) (d n x) = SheafLES.δ.{u} (shortExact_toAbSheaf hS) n U (cohOn.toH' S.X₃ U n x)) ∧ Function.Injective (cohOn.map R S.f U 0) ∧ ∀ n, Function.Exact (cohOn.map R S.f U n) (cohOn.map R S.g U n) ∧ Function.Exact (cohOn.map R S.g U n) (d n) ∧ Function.Exact (d n) (cohOn.map R S.f U (n + 1))) ∧ (IsNoetherianRing R → (Module.Finite R (cohOn S.X₂ U 0) → Module.Finite R (cohOn S.X₁ U 0)) ∧ ∀ n, (Module.Finite R (cohOn S.X₁ U n) → Module.Finite R (cohOn S.X₃ U n) → Module.Finite R (cohOn S.X₂ U n)) ∧ (Module.Finite R (cohOn S.X₂ U n) → Module.Finite R (cohOn S.X₁ U (n + 1)) → Module.Finite R (cohOn S.X₃ U n)) ∧ (Module.Finite R (cohOn S.X₃ U n) → Module.Finite R (cohOn S.X₂

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.SummableNormBoundedSeries

theorem **summable_norm_of_bounded** {a : ℕ → K} {B : ℝ} (ha : ∀ n, ‖a n‖ ≤ B) {t : K} (ht : ‖t‖ < 1) : Summable fun n => ‖a n * t ^ n‖

theorem **tsum_coeff_mul** [CompleteSpace K] (f g : K⟦X⟧) (t : K) (hf : Summable fun n => ‖coeff n f * t ^ n‖) (hg : Summable fun n => ‖coeff n g * t ^ n‖) : ∑' n, coeff n (f * g) * t ^ n = (∑' n, coeff n f * t ^ n) * ∑' n, coeff n g * t ^ n

theorem **tsum_coeff_C** (c t : K) : ∑' n, coeff n (C c : K⟦X⟧) * t ^ n = c

theorem **tsum_coeff_X** (t : K) : ∑' n, coeff n (X : K⟦X⟧) * t ^ n = t

theorem **evalDisc_apply** (f : O⟦X⟧) : evalDisc i hi t ht f = ∑' n, i (coeff n f) * t ^ n

@[simp] theorem **evalDisc_C** (a : O) : evalDisc i hi t ht (C a) = i a

@[simp] theorem **evalDisc_X** : evalDisc i hi t ht X = t

theorem **mem_boundedSeries** {f : F⟦X⟧} : f ∈ boundedSeries F ↔ ∃ B : ℝ, ∀ n, ‖coeff n f‖ ≤ B

theorem **C_mem_boundedSeries** (c : F) : (C c : F⟦X⟧) ∈ boundedSeries F

theorem **X_mem_boundedSeries** : (X : F⟦X⟧) ∈ boundedSeries F

theorem **summable_norm_boundedSeries** (f : boundedSeries F) {π : F} (hπ : ‖π‖ < 1) : Summable fun n => ‖coeff n (f : F⟦X⟧) * π ^ n‖

theorem **evalBounded_apply** (f : boundedSeries F) : evalBounded π hπ f = ∑' n, coeff n (f : F⟦X⟧) * π ^ n

@[simp] theorem **evalBounded_C** (c : F) : evalBounded π hπ ⟨C c, C_mem_boundedSeries c⟩ = c

@[simp] theorem **evalBounded_X** : evalBounded π hπ ⟨X, X_mem_boundedSeries⟩ = π

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ThreeTermComplexOfProjectiveCechModel

theorem **d_termMap** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) (p : ℕ) (x : R' ⊗[R] (cechLB f M D).X p) : ((cechLB f' (M.pull π) fun i => π ⁻¹ᵁ D i).d p (p + 1)).hom (termMap M D w p x) = termMap M D w (p + 1) (((cechLB f M D).d p (p + 1)).hom.baseChange R' x)

theorem **termMap_bijective** [Finite κ] (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hD : ∀ I : Finset κ, I.Nonempty → IsAffineOpen (⊓ i ∈ I, D i)) (p : ℕ) : Function.Bijective (termMap M D H.w p)

theorem **termMap_bijOn** [Finite κ] (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hD : ∀ I : Finset κ, I.Nonempty → IsAffineOpen (⊓ i ∈ I, D i)) : Set.BijOn (termMap M D H.w 0) (LinearMap.ker (((cechLB f M D).d 0 1).hom.baseChange R')) (LinearMap.ker ((cechLB f' (M.pull π) fun i => π ⁻¹ᵁ D i).d 0 1).hom)

theorem **exact_baseChange** [Finite κ] (hM : M.IsLB) (H : IsPullback π f' f (Spec.map (CommRingCat.ofHom (algebraMap R R')))) (hD : ∀ I : Finset κ, I.Nonempty → IsAffineOpen (⊓ i ∈ I, D i)) (h : Function.Exact ((cechLB f' (M.pull π) fun i => π ⁻¹ᵁ D i).d 0 1).hom ((cechLB f' (M.pull π) fun i => π ⁻¹ᵁ D i).d 1 2).hom) : Function.Exact (((cechLB f M D).d 0 1).hom.baseChange R') (((cechLB f M D).d 1 2).hom.baseChange R')

theorem **iSup_preimage** (π : Y' ⟶ Y) (hc : ⨆ i, D i = ⊤) : ⨆ i, π ⁻¹ᵁ D i = ⊤

theorem **coord_termMap_sum** (w : π ≫ f = f' ≫ Spec.map (CommRingCat.ofHom (algebraMap R R'))) {a : ℕ} (v : Fin a → R') (x : Fin a → (cechLB f M D).X 0) (s : {s : Finset κ // s.card = 0 + 1}) (j : M.ι) (h : π ⁻¹ᵁ ((⊓ i ∈ s.1, D i) ⊓ M.U j) ≤ (⊓ i ∈ s.1, π ⁻¹ᵁ D i) ⊓ (M.pull π).U fun _ => j) : rs Y' h (SecOnR.coord (f

theorem **perf_of_projective_model** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (K : CochainComplex (ModuleCat.{0} R) ℕ) (φ : K ⟶ cechLB f M D), M.IsLB → ⨆ i, D i = ⊤ → (∀ I : Finset κ, I.Nonempty → IsAffineOpen (⊓ i ∈ I, D i)) → ((∀ p, Module.Finite R (K.X p) ∧ Module.Projective R (K.X p)) ∧ ∀ (R' : Type) [CommRing R'] [Algebra R R'], QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)) → Nonempty (Perf f M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.ThreeTermComplexOfProjectiveCechModelStatement

theorem **_root_.GlobalTheory.T.threeTermComplex_of_projective_cech_model** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (M : LB Y) (κ : Type) [LinearOrder κ] [Fintype κ] (D : κ → Y.Opens) (K : CochainComplex (ModuleCat.{0} R) ℕ) (φ : K ⟶ cechLB f M D), M.IsLB → ⨆ i, D i = ⊤ → (∀ I : Finset κ, I.Nonempty → IsAffineOpen (⊓ i ∈ I, D i)) → ((∀ p, Module.Finite R (K.X p) ∧ Module.Projective R (K.X p)) ∧ ∀ (R' : Type) [CommRing R'] [Algebra R R'], QuasiIso (((ModuleCat.extendScalars.{0, 0, 0} (algebraMap R R')).mapHomologicalComplex _).map φ)) → Nonempty (Perf f M)

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.TopClassSurjective

theorem **univ_subset** {n : ℕ} (hn : Fintype.card ι = n) (s : {s : Finset ι // s.card = n}) : (Finset.univ : Finset ι) ⊆ s.1

theorem **topTerm_surjective** (A : Cube.{0} ι R) {n : ℕ} (hn : Fintype.card ι = n) : Function.Surjective (topTerm A hn)

theorem **topTerm_res_mem** (A : Cube.{0} ι R) {p : ℕ} (hn : Fintype.card ι = p + 1) (i : ι) (x : A.obj (Finset.univ.erase i)) : ∃ y, Cube.d A p y = topTerm A hn (Cube.res A (Finset.erase_subset i _) x)

theorem **d_top** (A : Cube.{0} ι R) : (cubeComplex A).d (Fintype.card ι) (Fintype.card ι + 1) = 0

theorem **topClass_surjective** (A : Cube.{0} ι R) : Function.Surjective (topClass A)

theorem **topClass_res** (A : Cube.{0} ι R) (i : ι) (x : A.obj (Finset.univ.erase i)) : topClass A (Cube.res A (Finset.erase_subset i _) x) = 0

theorem **span_topClass** {κ : Type} (T : κ → Finset ι) (A : Cube.{0} ι R) (b : ∀ s : Finset ι, Module.Basis {a : κ // T a ⊆ s} R (A.obj s)) (hb : ∀ (s t : Finset ι) (h : s ⊆ t) (a : {a : κ // T a ⊆ s}), Cube.res A h (b s a) = b t ⟨a.1, a.2.trans h⟩) : Submodule.span R

((fun a : κ => topClass A (b Finset.univ (a, Finset.subset_univ _))) '' {a | T a = Finset.univ}) = ⊤

theorem **main** : ∀ (ι : Type) [LinearOrder ι] [Fintype ι] (R : Type) [CommRing R] (κ : Type) (T : κ → Finset ι) (A : Cube.{0} ι R) (b : ∀ s : Finset ι, Module.Basis {a : κ // T a ⊆ s} R (A.obj s)), (∀ (s t : Finset ι) (h : s ⊆ t) (a : {a : κ // T a ⊆ s}), Cube.res A h (b s a) = b t ⟨a.1, a.2.trans h⟩) → ((∀ a, T a ≠ Finset.univ) → (cubeComplex A).ExactAt (Fintype.card ι)) ∧ ({a | T a = Finset.univ}.Finite → Module.Finite R ((cubeComplex A).homology (Fintype.card ι)))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.TranslateDivAdditive

theorem **exists_gen_additive** {S : Scheme.{0}} {G : Over S} [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) {T : Over S} (a b c : T ⟶ G) (i : Fin 2 → L.ι) (hi : ∀ k (y : T.left), (![a.left, (1 : T ⟶ G).left] k) y ∈ L.U (i k)) : ∃ u : LB.Gen (LB.LinComb L (K

theorem **exists_iso_of_cube** {S : Scheme.{0}} [Unique S] {G : Over S} [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (t : LB.Gen (Theta G L)) (ht : t.IsGen) (x : 𝟙_ (Over S) ⟶ G) : ∃ w : LB.Iso ((LB.LinComb L (K

theorem **translate_div_additive** : ∀ (K : Type) [Field K] [IsAlgClosed K] (P : PolAV (Spec (CommRingCat.of K))) (x : 𝟙_ (Over (Spec (CommRingCat.of K))) ⟶ P.A), ∃ w : LB.Iso ((LB.LinComb P.L (K

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.TwistedSectionsLocaliseToCharts

theorem **eval₂Hom_xProd_pow** {ι : Type} {R B : Type*} [CommRing R] [CommRing B] (c : R →+* B) (b : ι → B) (I : Finset ι) (k : ℕ) : MvPolynomial.eval₂Hom c b (xProd R I ^ k) = (∏ i ∈ I, b i) ^ k

theorem **twisted_sections_localise** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.TwistedSectionsLocaliseToChartsStatement

theorem **_root_.GlobalTheory.T.twistedSections_localise_to_charts** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H) (M : LB Y), H.IsLB → M.IsLB → VeryAmpleBy f H s → ∀ (hx : ∀ V : Y.Opens, OperatorsCommute (ι

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.VeryAmpleCompactQuasiSeparatedAffineCharts

theorem **isRetrocompact_nonzeroLocus** (hH : H.IsLB) (s : LB.Sec H) : IsRetrocompact (X

theorem **isAffineOpen_biInf_nonzeroLocus** {κ : Type} (hH : H.IsLB) (s : κ → LB.Sec H) (ha : ∀ i, IsAffineOpen (nonzeroLocus (s i))) (I : Finset κ) (hI : I.Nonempty) : IsAffineOpen (⨅ i ∈ I, nonzeroLocus (s i))

theorem **quasiSeparatedSpace_of_cover** {κ : Type} (hH : H.IsLB) (s : κ → LB.Sec H) (hc : ⨆ i, nonzeroLocus (s i) = ⊤) (ha : ∀ i, IsAffineOpen (nonzeroLocus (s i))) : QuasiSeparatedSpace Y

theorem **compactSpace_of_finite_cover** {κ : Type} [Finite κ] (D : κ → Y.Opens) (hc : ⨆ i, D i = ⊤) (ha : ∀ i, IsAffineOpen (D i)) : CompactSpace Y

theorem **compact_quasiSeparated_isAffineOpen** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → CompactSpace Y ∧ QuasiSeparatedSpace Y ∧ ∀ I : Finset (Fin (m + 1)), I.Nonempty → IsAffineOpen (⨅ i ∈ I, nonzeroLocus (s i))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.VeryAmpleCompactQuasiSeparatedAffineChartsStatement

theorem **_root_.GlobalTheory.T.veryAmple_compact_quasiSeparated_affine_charts** : ∀ (R : Type) [CommRing R] (Y : Scheme.{0}) (f : Y ⟶ Spec (CommRingCat.of R)) (H : LB Y) (m : ℕ) (s : Fin (m + 1) → LB.Sec H), H.IsLB → VeryAmpleBy f H s → CompactSpace Y ∧ QuasiSeparatedSpace Y ∧ ∀ I : Finset (Fin (m + 1)), I.Nonempty → IsAffineOpen (⨅ i ∈ I, nonzeroLocus (s i))

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.WeightAddLeOfMemSupportDividedContraction

@[simp] theorem **commEval_X** (a : σ → A) (ha : ∀ i j, Commute (a i) (a j)) (i : σ) : commEval k a ha (X i) = a i

@[simp] theorem **commEval_C** (a : σ → A) (ha : ∀ i j, Commute (a i) (a j)) (c : k) : commEval k a ha (C c) = algebraMap k A c

theorem **commute_commEval** {a : σ → A} (ha : ∀ i j, Commute (a i) (a j)) {b : A} (hb : ∀ i, Commute b (a i)) (P : MvPolynomial σ k) : Commute b (commEval k a ha P)

theorem **commute_mulLeft_mulRight** {a : ι → A} {a' : κ → A} (ha : ∀ i j, Commute (a i) (a j)) (ha' : ∀ i j, Commute (a' i) (a' j)) (v w : ι ⊕ κ) : Commute (Sum.elim (fun i => LinearMap.mulLeft k (a i)) (fun j => LinearMap.mulRight k (a' j)) v : Module.End k A) (Sum.elim (fun i => LinearMap.mulLeft k (a i)) (fun j => LinearMap.mulRight k (a' j)) w)

@[simp] theorem **orderedEval_C** (c : k) : orderedEval k a a' ha ha' (C c) = algebraMap k A c

@[simp] theorem **orderedEval_one** : orderedEval k a a' ha ha' 1 = 1

@[simp] theorem **multiChoose_zero** (n : σ →₀ ℕ) : multiChoose n 0 = 1

theorem **multiChoose_eq_mul_erase** (n β : σ →₀ ℕ) (i : σ) : multiChoose n β = (n i).choose (β i) * multiChoose n (β.erase i)

theorem **multiChoose_add_single_left** {β : σ →₀ ℕ} {i : σ} (h : β i = 0) (n : σ →₀ ℕ) (e : ℕ) : multiChoose (n + Finsupp.single i e) β = multiChoose n β

theorem **le_of_multiChoose_ne_zero** {n β : σ →₀ ℕ} (h : multiChoose n β ≠ 0) : β ≤ n

theorem **erase_add_single_self** (β : σ →₀ ℕ) (i : σ) (e : ℕ) : (β + Finsupp.single i e).erase i = β.erase i

theorem **multiChoose_pascal** (n β : σ →₀ ℕ) (i : σ) : multiChoose (n + Finsupp.single i 1) (β + Finsupp.single i 1) = multiChoose n (β + Finsupp.single i 1) + multiChoose n β

theorem **multiChoose_mul_sub** (n β : σ →₀ ℕ) (i : σ) : multiChoose n β * (n i - β i) = (β i + 1) * multiChoose n (β + Finsupp.single i 1)

theorem **dividedDeriv_monomial** (β n : σ →₀ ℕ) (c : R) : dividedDeriv R β (monomial n c) = multiChoose n β • monomial (n - β) c

theorem **coeff_dividedDeriv** (β m : σ →₀ ℕ) (P : MvPolynomial σ R) : coeff m (dividedDeriv R β P) = multiChoose (m + β) β • coeff (m + β) P

@[simp] theorem **dividedDeriv_zero** : dividedDeriv R (0 : σ →₀ ℕ) = 1

theorem **pderiv_dividedDeriv** (β : σ →₀ ℕ) (i : σ) (P : MvPolynomial σ R) : pderiv i (dividedDeriv R β P) = (β i + 1) • dividedDeriv R (β + Finsupp.single i 1) P

theorem **dividedDeriv_single_one** (i : σ) (P : MvPolynomial σ R) : dividedDeriv R (Finsupp.single i 1) P = pderiv i P

theorem **pderiv_pderiv_comm** (i j : σ) (P : MvPolynomial σ R) : pderiv i (pderiv j P) = pderiv j (pderiv i P)

theorem **commute_pderiv** (i j : σ) : Commute ((pderiv i).toLinearMap : Module.End R (MvPolynomial σ R)) (pderiv j).toLinearMap

theorem **finsupp_induction_single** {p : (σ →₀ ℕ) → Prop} (h0 : p 0) (h : ∀ β i, p β → p (β + Finsupp.single i 1)) (β : σ →₀ ℕ) : p β

theorem **dividedDeriv_add_single_X_mul** (β : σ →₀ ℕ) (i : σ) (P : MvPolynomial σ R) : dividedDeriv R (β + Finsupp.single i 1) (X i * P) = X i * dividedDeriv R (β + Finsupp.single i 1) P + dividedDeriv R β P

theorem **dividedDeriv_X_mul_of_eq_zero** {β : σ →₀ ℕ} {i : σ} (h : β i = 0) (P : MvPolynomial σ R) : dividedDeriv R β (X i * P) = X i * dividedDeriv R β P

theorem **monomial_induction_inl** {p : MvPolynomial (ι ⊕ κ) R → Prop} (c : R) (h0 : ∀ t : κ →₀ ℕ, p (rename Sum.inr (monomial t c))) (h : ∀ (s : ι ⊕ κ →₀ ℕ) (i : ι), p (monomial s c) → p (X (Sum.inl i) * monomial s c)) (s : ι ⊕ κ →₀ ℕ) : p (monomial s c)

theorem **induction_inl** {p : MvPolynomial (ι ⊕ κ) R → Prop} (h0 : ∀ q : MvPolynomial κ R, p (rename Sum.inr q)) (hadd : ∀ P Q, p P → p Q → p (P + Q)) (h : ∀ P (i : ι), p P → p (X (Sum.inl i) * P)) (P : MvPolynomial (ι ⊕ κ) R) : p P

theorem **rowContraction_apply** (γ : ι → κ → R) (i : ι) (P : MvPolynomial (ι ⊕ κ) R) : rowContraction γ i P = ∑ j, γ i j • pderiv (Sum.inr j) P

theorem **commute_pderiv_rowContraction** (γ : ι → κ → R) (v : ι ⊕ κ) (i : ι) : Commute ((pderiv v).toLinearMap : Module.End R (MvPolynomial (ι ⊕ κ) R)) (rowContraction γ i)

theorem **commute_rowContraction** (γ : ι → κ → R) (i i' : ι) : Commute (rowContraction γ i) (rowContraction γ i')

theorem **rowContraction_X_inl_mul** (γ : ι → κ → R) (i i' : ι) (P : MvPolynomial (ι ⊕ κ) R) : rowContraction γ i (X (Sum.inl i') * P) = X (Sum.inl i') * rowContraction γ i P

@[simp] theorem **rowContraction_C** (γ : ι → κ → R) (i : ι) (c : R) : rowContraction γ i (C c) = 0

theorem **contraction_apply** (γ : ι → κ → R) (P : MvPolynomial (ι ⊕ κ) R) : contraction γ P = ∑ i, ∑ j, γ i j • pderiv (Sum.inl i) (pderiv (Sum.inr j) P)

theorem **contraction_eq_sum** (γ : ι → κ → R) : contraction γ = ∑ i, (pderiv (Sum.inl i)).toLinearMap * rowContraction γ i

theorem **commute_contraction_rowContraction** (γ : ι → κ → R) (i : ι) : Commute (contraction γ) (rowContraction γ i)

theorem **contraction_rename_inr** (γ : ι → κ → R) (q : MvPolynomial κ R) : contraction γ (rename Sum.inr q) = 0

theorem **contraction_X_inl_mul** (γ : ι → κ → R) (i : ι) (P : MvPolynomial (ι ⊕ κ) R) : contraction γ (X (Sum.inl i) * P) = X (Sum.inl i) * contraction γ P + rowContraction γ i P

theorem **contraction_pow_X_inl_mul** (γ : ι → κ → R) (n : ℕ) (i : ι) (P : MvPolynomial (ι ⊕ κ) R) : (contraction γ ^ (n + 1)) (X (Sum.inl i) * P) = X (Sum.inl i) * (contraction γ ^ (n + 1)) P + (n + 1) • rowContraction γ i ((contraction γ ^ n) P)

@[simp] theorem **dividedContraction_zero** (γ : ι → κ → R) : dividedContraction γ 0 = 1

theorem **dividedContraction_succ_rename_inr** (γ : ι → κ → R) (n : ℕ) (q : MvPolynomial κ R) : dividedContraction γ (n + 1) (rename Sum.inr q) = 0

theorem **dividedContraction_succ_X_inl_mul** (γ : ι → κ → R) (n : ℕ) (i : ι) (P : MvPolynomial (ι ⊕ κ) R) : dividedContraction γ (n + 1) (X (Sum.inl i) * P) = X (Sum.inl i) * dividedContraction γ (n + 1) P + rowContraction γ i (dividedContraction γ n P)

theorem **factorial_smul_dividedContraction** (γ : ι → κ → R) (n : ℕ) : n.factorial • dividedContraction γ n = contraction γ ^ n

@[simp] theorem **dividedContraction_one** (γ : ι → κ → R) : dividedContraction γ 1 = contraction γ

Development.AbelianVarietyTheory.Proofs.CoherentCohomologyProofs.XSmulMemWeightedHomogeneousSubmodule

private theorem **coeff_comp_X_add_C_of_natDegree_le_add_one** (P : K[X]) (r : K) (e : ℕ) (hP : P.natDegree ≤ e + 1) : (P.comp (X + C r)).coeff e = P.coeff e + r * (e + 1) * P.coeff (e + 1)

theorem **coeff_comp_X_add_C_sub** (P : K[X]) (r : K) (e : ℕ) (hP : P.degree ≤ (e + 1 : ℕ)) : (P.comp (X + C r) - P).coeff e = r * (e + 1) * P.coeff (e + 1)

private theorem **coeff_comp_X_add_C_of_natDegree_le** (P : K[X]) (r : K) (e : ℕ) (hP : P.natDegree ≤ e) : (P.comp (X + C r)).coeff e = P.coeff e

theorem **degree_comp_X_add_C_sub_lt** (P : K[X]) (r : K) (e : ℕ) (hP : P.degree ≤ (e : ℕ)) : (P.comp (X + C r) - P).degree < (e : ℕ)

theorem **exists_comp_X_add_one_sub_eq** (d : ℕ) (Q : K[X]) (hQ : Q.degree < (d : ℕ)) : ∃ P : K[X], P.degree ≤ (d : ℕ) ∧ P.comp (X + C 1) - P = Q

theorem **polynomial_eq_of_eventually_eval_int** {P Q : K[X]} (h : ∀ᶠ n : ℤ in atTop, P.eval (n : K) = Q.eval (n : K)) : P = Q

private theorem **eval_comp_X_add_one_sub_eq_eval_succ_sub** (P : K[X]) (n : ℤ) : (P.comp (X + C 1) - P).eval (n : K) = P.eval ((n + 1 : ℤ) : K) - P.eval (n : K)

theorem **exists_polynomial_of_diff_from** {f : ℤ → K} {Q : K[X]} {d : ℕ} (hQ : Q.degree < (d : ℕ)) (N : ℤ) (h : ∀ n, N ≤ n → f (n + 1) - f n = Q.eval (n : K)) : ∃ P : K[X], P.degree ≤ (d : ℕ) ∧ P.comp (X + C 1) - P = Q ∧ ∀ n, N ≤ n → f n = P.eval (n : K)

theorem **exists_polynomial_of_diff_eventually** {f : ℤ → K} {Q : K[X]} {d : ℕ} (hQ : Q.degree < (d : ℕ)) (h : ∀ᶠ n : ℤ in atTop, f (n + 1) - f n = Q.eval (n : K)) : ∃ P : K[X], P.degree ≤ (d : ℕ) ∧ P.comp (X + C 1) - P = Q ∧ (∀ e, d = e + 1 → Q.coeff e = d * P.coeff d) ∧ ∀ᶠ n : ℤ in atTop, f n = P.eval (n : K)

theorem **proj_apply** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] (n : ℤ) (m : M) : proj 𝒜 n m = (decompose 𝒜 m n : M)

theorem **proj_mem** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] (n : ℤ) (m : M) : proj 𝒜 n m ∈ 𝒜 n

theorem **proj_of_mem** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] {n : ℤ} {m : M} (h : m ∈ 𝒜 n) : proj 𝒜 n m = m

theorem **proj_of_mem_ne** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] {i n : ℤ} {m : M} (h : m ∈ 𝒜 i) (hi : i ≠ n) : proj 𝒜 n m = 0

theorem **proj_map** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] (𝒩 : ℤ → Submodule R N) [Decomposition 𝒩] (f : M →ₗ[R] N) (d : ℤ) (hf : ∀ n, ∀ m ∈ 𝒜 n, f m ∈ 𝒩 (n + d)) (n : ℤ) (m : M) : proj 𝒩 (n + d) (f m) = f (proj 𝒜 n m)

theorem **exists_bound_proj** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] (G : Finset M) : ∃ b : ℤ, ∀ g ∈ G, ∀ n : ℤ, b < |n| → proj 𝒜 n g = 0

theorem **inf_range_eq_map** (𝒜 : ℤ → Submodule R M) [Decomposition 𝒜] (𝒩 : ℤ → Submodule R N) [Decomposition 𝒩] (f : M →ₗ[R] N) (d : ℤ) (hf : ∀ n, ∀ m ∈ 𝒜 n, f m ∈ 𝒩 (n + d)) (n : ℤ) : 𝒩 (n + d) ⊓ LinearMap.range f = (𝒜 n).map f

theorem **smul_mem_of_X_smul_mem** (P : Submodule R M) (hP : ∀ i, ∀ m ∈ P, (MvPolynomial.X i : MvPolynomial ι R) • m ∈ P) (p : MvPolynomial ι R) : ∀ m ∈ P, p • m ∈ P

theorem **mem_of_mem_span** (P : Submodule R M) (hP : ∀ i, ∀ m ∈ P, (MvPolynomial.X i : MvPolynomial ι R) • m ∈ P) {G : Set M} (hG : ∀ g ∈ G, g ∈ P) {m : M} (hm : m ∈ Submodule.span (MvPolynomial ι R) G) : m ∈ P

private theorem **X_pow_smul_mem_add_nat_cast** (i : ι) (k : ℕ) (n : ℤ) (m : M) (hm : m ∈ 𝒜 n) : ((MvPolynomial.X i : MvPolynomial ι R) ^ k) • m ∈ 𝒜 (n + k)

private theorem **monomial_smul_mem_add_degree** (s : ι →₀ ℕ) (a : R) : ∀ n : ℤ, ∀ m ∈ 𝒜 n, (MvPolynomial.monomial s a) • m ∈ 𝒜 (n + s.degree)

theorem **smul_mem_of_isHomogeneous** {p : MvPolynomial ι R} {d : ℕ} (hp : p.IsHomogeneous d) (n : ℤ) (m : M) (hm : m ∈ 𝒜 n) : p • m ∈ 𝒜 (n + d)

theorem **proj_X_smul** (i : ι) (n : ℤ) (m : M) : proj 𝒜 n ((MvPolynomial.X i : MvPolynomial ι R) • m) = (MvPolynomial.X i : MvPolynomial ι R) • proj 𝒜 (n − 1) m

theorem **finite_piece** [Finite ι] [Module.Finite (MvPolynomial ι R) M] (n : ℤ) : Module.Finite R (𝒜 n)

theorem **finrank_map_add_finrank_inf_ker** (f : M →ₗ[k] N) (B : Submodule k M) [Module.Finite k B] : finrank k (B.map f) + finrank k (B ⊓ LinearMap.ker f : Submodule k M) = finrank k B

theorem **finrank_sup_map_add_finrank_inf_comap** (f : M →ₗ[k] N) (B : Submodule k M) (A : Submodule k N) [Module.Finite k B] [Module.Finite k A] : finrank k (A ⊔ B.map f : Submodule k N) + finrank k (B ⊓ A.comap f : Submodule k M) = finrank k A + finrank k B

private theorem **eventually_mem_of_homogeneous_mem_of_X_smul_mem** [Finite ι] [Module.Finite (MvPolynomial ι k) M] (A B : Submodule (MvPolynomial ι k) M) (hA : ∀ n, ∀ m ∈ A, proj 𝒜 n m ∈ A) (hs : ∀ i, ∀ m ∈ B, (MvPolynomial.X i : MvPolynomial ι k) • m ∈ A) : ∀ᶠ n : ℤ in atTop, ∀ m ∈ B, m ∈ 𝒜 n → m ∈ A

private theorem **finrank_graded_inf_add_eq_of_X_smul_comap** [Finite ι] [Module.Finite (MvPolynomial ι k) M] (i : ι) (A B : Submodule (MvPolynomial ι k) M) (hA : ∀ n, ∀ m ∈ A, proj 𝒜 n m ∈ A) (hB : ∀ n, ∀ m ∈ B, proj 𝒜 n m ∈ B) (n : ℤ) : finrank k ((A ⊔ B.map (LinearMap.lsmul (MvPolynomial ι k) M (MvPolynomial.X i))).restrictScalars k ⊓ 𝒜 (n + 1) : Submodule k M) + finrank k ((B ⊓ A.comap (LinearMap.lsmul (MvPolynomial ι k) M (MvPolynomial.X i))).restrictScalars k ⊓ 𝒜 n : Submodule k M) = finrank k (A.restrictScalars k ⊓ 𝒜 (n + 1) : Submodule k M) + finrank k (B.restrictScalars k ⊓ 𝒜 n : Submodule k M)

theorem **exists_polynomial_subquotient** [Finite ι] [Module.Finite (MvPolynomial ι k) M] (s : Finset ι) (A B : Submodule (MvPolynomial ι k) M) (hAB : A ≤ B) (hA : ∀ n, ∀ m ∈ A, proj 𝒜 n m ∈ A) (hB : ∀ n, ∀ m ∈ B, proj 𝒜 n m ∈ B) (hs : ∀ i ∉ s, ∀ m ∈ B, (MvPolynomial.X i : MvPolynomial ι k) • m ∈ A) : ∃ P : ℚ[X], P.degree < (s.card : ℕ) ∧ ∀ᶠ n : ℤ in atTop, (finrank k (B.restrictScalars k ⊓ 𝒜 n : Submodule k M) : ℚ) − finrank k (A.restrictScalars k ⊓ 𝒜 n : Submodule k M) = P.eval (n : ℚ)

theorem **IsHilbertPoly.unique** {𝒜 : ℤ → Submodule k M} {P Q : ℚ[X]} (hP : IsHilbertPoly 𝒜 P) (hQ : IsHilbertPoly 𝒜 Q) : P = Q

theorem **IsHilbertPoly.shift** {𝒜 : ℤ → Submodule k M} {P : ℚ[X]} (hP : IsHilbertPoly 𝒜 P) (d : ℤ) : IsHilbertPoly (fun n => 𝒜 (n + d)) (P.comp (X + C (d : ℚ))) where

theorem **finrank_eq_add_of_exact** (𝒜' : ℤ → Submodule k M') [Decomposition 𝒜'] (𝒜 : ℤ → Submodule k M) [Decomposition 𝒜] (f : M' →ₗ[k] M) (g : M →ₗ[k] M'') (d : ℤ) (hf : ∀ n, ∀ m ∈ 𝒜' n, f m ∈ 𝒜 (n + d)) (hinj : Function.Injective f) (hfg : Function.Exact f g) (n : ℤ) [Module.Finite k (𝒜 (n + d))] : finrank k (𝒜 (n + d)) = finrank k (𝒜' n) + finrank k ((𝒜 (n + d)).map g)

theorem **IsHilbertPoly.coker** {𝒜' : ℤ → Submodule k M'} [Decomposition 𝒜'] {𝒜 : ℤ → Submodule k M} [Decomposition 𝒜] {P' P : ℚ[X]} (hP' : IsHilbertPoly 𝒜' P') (hP : IsHilbertPoly 𝒜 P) (f : M' →ₗ[k] M) (g : M →ₗ[k] M'') (d : ℤ) (hf : ∀ n, ∀ m ∈ 𝒜' n, f m ∈ 𝒜 (n + d)) (hinj : Function.Injective f) (hfg : Function.Exact f g) : IsHilbertPoly (fun n => (𝒜 n).map g) (P − P'.comp (X − C (d : ℚ))) where

theorem **exists_isHilbertPoly** [Fintype ι] [Module.Finite (MvPolynomial ι k) M] (𝒜 : ℤ → Submodule k M) [Decomposition 𝒜] (hX : ∀ (i : ι) (n : ℤ), ∀ m ∈ 𝒜 n, (MvPolynomial.X i : MvPolynomial ι k) • m ∈ 𝒜 (n + 1)) : ∃ P : ℚ[X], IsHilbertPoly 𝒜 P ∧ P.degree < (Fintype.card ι : ℕ)

theorem **coeff_sub_comp_X_sub_C** {K : Type*} [Field K] (P : K[X]) (r : K) (e : ℕ) (hP : P.degree ≤ (e + 1 : ℕ)) : (P − P.comp (X − C r)).degree < (e + 1 : ℕ) ∧ (P − P.comp (X − C r)).coeff e = r * (e + 1) * P.coeff (e + 1)

theorem **X_smul_mem_weightedHomogeneousSubmodule** (i : ι) (n : ℤ) (m : MvPolynomial ι k) (hm : m ∈ weightedHomogeneousSubmodule k (fun _ : ι => (1 : ℤ)) n) : (MvPolynomial.X i : MvPolynomial ι k) • m ∈ weightedHomogeneousSubmodule k (fun _ : ι => (1 : ℤ)) (n + 1)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.AbsTwoMulInnerLeOfNormSubEqInfDist

theorem **exists_int_comb_norm_sub_sq_le** (k : ℕ) (v : Fin k → E) {x : E} (hx : x ∈ span ℝ (Set.range v)) : ∃ w ∈ span ℤ (Set.range v), ‖x − w‖ ^ 2 ≤ (∑ i, ‖v i‖ ^ 2) / 4

theorem **exists_mem_span_int_norm_sub_sq_le** {ι : Type*} [Fintype ι] (v : ι → E) {x : E} (hx : x ∈ span ℝ (Set.range v)) : ∃ w ∈ span ℤ (Set.range v), ‖x − w‖ ^ 2 ≤ (∑ i, ‖v i‖ ^ 2) / 4

theorem **exists_mem_span_int_norm_sub_le** {ι : Type*} [Fintype ι] (v : ι → E) {r : ℝ} (hr : ∀ i, ‖v i‖ ≤ r) {x : E} (hx : x ∈ span ℝ (Set.range v)) : ∃ w ∈ span ℤ (Set.range v), ‖x − w‖ ≤ √(Fintype.card ι) / 2 * r

theorem **exists_mem_norm_sub_sq_le_of_mem_span** {ι : Type*} [Fintype ι] {v : ι → E} (hL : ∀ i, v i ∈ L) {x : E} (hx : x ∈ span ℝ (Set.range v)) : ∃ w ∈ L, w ∈ span ℝ (Set.range v) ∧ ‖x − w‖ ^ 2 ≤ (∑ i, ‖v i‖ ^ 2) / 4

theorem **exists_mem_norm_sub_le_of_mem_span** {ι : Type*} [Fintype ι] {v : ι → E} {r : ℝ} (hL : ∀ i, v i ∈ L) (hr : ∀ i, ‖v i‖ ≤ r) {x : E} (hx : x ∈ span ℝ (Set.range v)) : ∃ w ∈ L, w ∈ span ℝ (Set.range v) ∧ ‖x − w‖ ≤ √(Fintype.card ι) / 2 * r

theorem **coveringRadius_le_sqrt_sum** {ι : Type*} [Fintype ι] {v : ι → E} (hv : span ℝ (Set.range v) = ⊤) (hL : ∀ i, v i ∈ L) : coveringRadius L ≤ √(∑ i, ‖v i‖ ^ 2) / 2

theorem **coveringRadius_le_of_span_eq_top** {ι : Type*} [Fintype ι] {v : ι → E} {r : ℝ} (hv : span ℝ (Set.range v) = ⊤) (hL : ∀ i, v i ∈ L) (hr : ∀ i, ‖v i‖ ≤ r) : coveringRadius L ≤ √(Fintype.card ι) / 2 * r

theorem **exists_mem_norm_sub_le_of_linearIndependent** [FiniteDimensional ℝ E] {ι : Type*} [Fintype ι] {v : ι → E} {r : ℝ} (hv : LinearIndependent ℝ v) (hn : Fintype.card ι = finrank ℝ E) (hL : ∀ i, v i ∈ L) (hr : ∀ i, ‖v i‖ ≤ r) (x : E) : ∃ w ∈ L, ‖x − w‖ ≤ √(finrank ℝ E) / 2 * r

theorem **coveringRadius_le_of_linearIndependent** [FiniteDimensional ℝ E] {ι : Type*} [Fintype ι] {v : ι → E} {r : ℝ} (hv : LinearIndependent ℝ v) (hn : Fintype.card ι = finrank ℝ E) (hL : ∀ i, v i ∈ L) (hr : ∀ i, ‖v i‖ ≤ r) : coveringRadius L ≤ √(finrank ℝ E) / 2 * r

theorem **exists_mem_norm_sub_le_lastMin_of_mem_span** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] {x : E} (hx : x ∈ span ℝ (L : Set E)) : ∃ w ∈ L, ‖x − w‖ ≤ √(finrank ℤ L) / 2 * lastMin L

theorem **exists_mem_norm_sub_le_lastMin** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] (x : E) : ∃ w ∈ L, ‖x − w‖ ≤ √(finrank ℝ E) / 2 * lastMin L

theorem **coveringRadius_le_lastMin** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] : coveringRadius L ≤ √(finrank ℝ E) / 2 * lastMin L

theorem **coveringRadius_inSpan_le_lastMin** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] : coveringRadius (inSpan L) ≤ √(finrank ℤ L) / 2 * lastMin L

theorem **exists_closest_point** {F : Type*} [NormedAddCommGroup F] [NormedSpace ℝ F] [FiniteDimensional ℝ F] (L : Submodule ℤ F) [DiscreteTopology L] (x : F) : ∃ w ∈ L, ∀ z ∈ L, ‖x - w‖ ≤ ‖x - z‖

theorem **abs_two_mul_inner_le_of_norm_le** {a y : E} (h₁ : ‖a‖ ≤ ‖a + y‖) (h₂ : ‖a‖ ≤ ‖a - y‖) : |2 * ⟪a, y⟫_ℝ| ≤ ‖y‖ ^ 2

theorem **abs_two_mul_inner_le_of_closest** {p w : E} (hw : w ∈ L) (h : ∀ z ∈ L, ‖p - w‖ ≤ ‖p - z‖) {y : E} (hy : y ∈ L) : |2 * ⟪p - w, y⟫_ℝ| ≤ ‖y‖ ^ 2

theorem **closest_iff_abs_two_mul_inner_le** {p w : E} (hw : w ∈ L) : (∀ z ∈ L, ‖p - w‖ ≤ ‖p - z‖) ↔ ∀ y ∈ L, |2 * ⟪p - w, y⟫_ℝ| ≤ ‖y‖ ^ 2

theorem **abs_two_mul_inner_le_of_norm_sub_eq_infDist** {p w : E} (hw : w ∈ L) (h : ‖p - w‖ = Metric.infDist p (L : Set E)) {y : E} (hy : y ∈ L) : |2 * ⟪p - w, y⟫_ℝ| ≤ ‖y‖ ^ 2

theorem **exists_closest_point_inner** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] (p : E) : ∃ w ∈ L, (∀ z ∈ L, ‖p - w‖ ≤ ‖p - z‖) ∧ ‖p - w‖ ≤ ‖p‖ ∧ ∀ y ∈ L, |2 * ⟪p - w, y⟫_ℝ| ≤ ‖y‖ ^ 2

theorem **coveringRadius_and_closest_point** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] : (∀ (v : Fin (finrank ℝ E) → E) (r : ℝ), LinearIndependent ℝ v → (∀ i, v i ∈ L) → (∀ i, ‖v i‖ ≤ r) → ∀ x : E, ∃ w ∈ L, ‖x - w‖ ≤ √(finrank ℝ E) / 2 * r) ∧ coveringRadius L ≤ √(finrank ℝ E) / 2 * lastMin L ∧ (∀ x : E, ∃ w ∈ L, ∀ z ∈ L, ‖x - w‖ ≤ ‖x - z‖) ∧ (∀ p : E, ∀ w ∈ L, (∀ z ∈ L, ‖p - w‖ ≤ ‖p - z‖) → ∀ y ∈ L, |2 * ⟪p - w, y⟫_ℝ| ≤ ‖y‖ ^ 2)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ActualClassesCompactSupport

theorem **actualClasses_compactSupport** : Formalism.FourierDecayAlt.P5.ThmPActualCpt

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.BlockEmbConjTranspose

theorem **blockEmb_conjTranspose** (k i r' : ℕ) (g : Matrix (Fin (2 * r' + k + 1)) (Fin (2 * r' + k + 1)) ℂ) : (ιLev k i r' g)ᴴ = ιLev k i r' gᴴ

theorem **blockEmb_mem_Kset** : ∀ k r' : ℕ, ∀ y ∈ Kset k r', ιLev k 1 r' y ∈ Kset k (1 + r')

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.CoeffSubstInvMemOfTwoLeOrder

theorem **mem_weightGe** {w : σ → ℕ} {n : ℕ} {f : MvPowerSeries σ R} : f ∈ weightGe w n ↔ ∀ d, weight w d < n → coeff d f = 0

theorem **mem_weightGe_iff_le_weightedOrder** {w : σ → ℕ} {n : ℕ} {f : MvPowerSeries σ R} : f ∈ weightGe w n ↔ (n : ℕ∞) ≤ f.weightedOrder w

theorem **weightGe_mono** {w : σ → ℕ} {m n : ℕ} (h : m ≤ n) {f : MvPowerSeries σ R} (hf : f ∈ weightGe w n) : f ∈ weightGe w m

theorem **mem_weightGe_zero** {w : σ → ℕ} (f : MvPowerSeries σ R) : f ∈ weightGe w 0

theorem **mul_mem_weightGe** {w : σ → ℕ} {m n : ℕ} {f g : MvPowerSeries σ R} (hf : f ∈ weightGe w m) (hg : g ∈ weightGe w n) : f * g ∈ weightGe w (m + n)

theorem **smul_mem_weightGe** {w : σ → ℕ} {n : ℕ} (r : R) {f : MvPowerSeries σ R} (hf : f ∈ weightGe w n) : r • f ∈ weightGe w n

theorem **prod_pow_mem_weightGe** {w : τ → ℕ} {w' : σ → ℕ} {a : σ → MvPowerSeries τ R} (ha : ∀ s, a s ∈ weightGe w (w' s)) (d : σ →₀ ℕ) : (d.prod fun s e => a s ^ e) ∈ weightGe w (weight w' d)

theorem **mem_degGe** {n : ℕ} {f : MvPowerSeries σ R} : f ∈ degGe σ R n ↔ ∀ d : σ →₀ ℕ, d.degree < n → coeff d f = 0

theorem **mem_degGe_iff_le_order** {n : ℕ} {f : MvPowerSeries σ R} : f ∈ degGe σ R n ↔ (n : ℕ∞) ≤ f.order

theorem **mem_degGe_one** {f : MvPowerSeries σ R} : f ∈ degGe σ R 1 ↔ constantCoeff f = 0

theorem **eq_of_forall_sub_mem_degGe** {f g : MvPowerSeries σ R} (h : ∀ n, f - g ∈ degGe σ R n) : f = g

theorem **prod_pow_sub_mem_degGe** {a b : σ → MvPowerSeries τ R} {k : ℕ} (ha : ∀ s, a s ∈ degGe τ R 1) (hb : ∀ s, b s ∈ degGe τ R 1) (hab : ∀ s, a s - b s ∈ degGe τ R (k + 1)) (d : σ →₀ ℕ) : (d.prod fun s e => a s ^ e) - (d.prod fun s e => b s ^ e) ∈ degGe τ R (k + d.degree)

theorem **subst_sub_subst_mem_degGe** [Finite σ] {a b : σ → MvPowerSeries τ R} {k m : ℕ} (ha : ∀ s, a s ∈ degGe τ R 1) (hb : ∀ s, b s ∈ degGe τ R 1) (hab : ∀ s, a s - b s ∈ degGe τ R (k + 1)) {f : MvPowerSeries σ R} (hf : f ∈ degGe σ R m) : f.subst a - f.subst b ∈ degGe τ R (k + m)

theorem **subst_mem_weightGe** {w : τ → ℕ} {w' : σ → ℕ} {a : σ → MvPowerSeries τ R} (ha' : HasSubst a) (ha : ∀ s, a s ∈ weightGe w (w' s)) {m : ℕ} {f : MvPowerSeries σ R} (hf : f ∈ weightGe w' m) : f.subst a ∈ weightGe w m

theorem **mem_coeffsIn** {κ : Type*} {S : Subring R} {w : κ → σ → ℕ} {g : κ → ℕ} {f : MvPowerSeries σ R} : f ∈ coeffsIn S w g ↔ ∀ d, (∀ k, weight (w k) d < g k) → coeff d f ∈ S

theorem **smul_mem_coeffsIn** {κ : Type*} {S : Subring R} {w : κ → σ → ℕ} {g : κ → ℕ} {r : R} (hr : r ∈ S) {f : MvPowerSeries σ R} (hf : f ∈ coeffsIn S w g) : r • f ∈ coeffsIn S w g

theorem **X_mem_coeffsIn** {κ : Type*} {S : Subring R} {w : κ → σ → ℕ} {g : κ → ℕ} (s : σ) : (X s : MvPowerSeries σ R) ∈ coeffsIn S w g

theorem **subst_mem_coeffsIn** {κ : Type*} {S : Subring R} {w : κ → τ → ℕ} {w' : κ → σ → ℕ} {g : κ → ℕ} {a : σ → MvPowerSeries τ R} (ha' : HasSubst a) (ha : ∀ k s, a s ∈ weightGe (w k) (w' k s)) (haS : ∀ s, a s ∈ coeffsIn S w g) {f : MvPowerSeries σ R} (hf : f ∈ coeffsIn S w' g) : f.subst a ∈ coeffsIn S w g

theorem **mem_coeffsInDeg** {S : Subring R} {g : ℕ} {f : MvPowerSeries σ R} : f ∈ coeffsInDeg σ S g ↔ ∀ d : σ →₀ ℕ, d.degree < g → coeff d f ∈ S

theorem **subst_mem_coeffsInDeg** [Finite σ] {S : Subring R} {g : ℕ} {a : σ → MvPowerSeries τ R} (ha : ∀ s, constantCoeff (a s) = 0) (haS : ∀ s, a s ∈ coeffsInDeg τ S g) {f : MvPowerSeries σ R} (hf : f ∈ coeffsInDeg σ S g) : f.subst a ∈ coeffsInDeg τ S g

@[simp] theorem **linearPart_apply** (u : ι → MvPowerSeries σ R) (i : ι) (j : σ) : linearPart u i j = coeff (single j 1) (u i)

theorem **linearPart_X** [DecidableEq σ] : linearPart (X : σ → MvPowerSeries σ R) = 1

theorem **linearPart_map** {S : Type*} [CommRing S] (φ : R →+* S) (u : ι → MvPowerSeries σ R) : linearPart (fun i => MvPowerSeries.map φ (u i)) = (linearPart u).map φ

theorem **mem_degGe_two** {f : MvPowerSeries σ R} : f ∈ degGe σ R 2 ↔ constantCoeff f = 0 ∧ ∀ j, coeff (single j 1) f = 0

theorem **higherPart_mem_degGe** [Fintype σ] {u : ι → MvPowerSeries σ R} {i : ι} (hu : constantCoeff (u i) = 0) : higherPart u i ∈ degGe σ R 2

theorem **subst_eq_linearPart_add** [Fintype σ] (u : ι → MvPowerSeries σ R) (i : ι) {v : σ → MvPowerSeries τ R} (hv : HasSubst v) : (u i).subst v = ∑ j, linearPart u i j • v j + (higherPart u i).subst v

theorem **linearPart_subst** [Fintype σ] {u : ι → MvPowerSeries σ R} {v : σ → MvPowerSeries τ R} (hu : ∀ i, constantCoeff (u i) = 0) (hv : ∀ j, constantCoeff (v j) = 0) : linearPart (fun i => (u i).subst v) = linearPart u ∗ linearPart v

theorem **coeff_approxLimit** (j : τ) (d : σ →₀ ℕ) : coeff d (approxLimit u B j) = coeff d (((approxStep u B)^[d.degree + 1] 0) j)

theorem **approxStep_mem_degGe** (hu : ∀ i, constantCoeff (u i) = 0) {v : τ → MvPowerSeries σ R} (hv : ∀ j, v j ∈ degGe σ R 1) (j : τ) : approxStep u B v j ∈ degGe σ R 1

theorem **approxStep_sub_mem_degGe** (hu : ∀ i, constantCoeff (u i) = 0) {v v' : τ → MvPowerSeries σ R} (hv : ∀ j, v j ∈ degGe σ R 1) (hv' : ∀ j, v' j ∈ degGe σ R 1) {k : ℕ} (h : ∀ j, v j − v' j ∈ degGe σ R (k + 1)) (j : τ) : approxStep u B v j − approxStep u B v' j ∈ degGe σ R (k + 2)

theorem **iterate_mem_degGe** (hu : ∀ i, constantCoeff (u i) = 0) (n : ℕ) (j : τ) : ((approxStep u B)^[n] 0) j ∈ degGe σ R 1

theorem **iterate_sub_mem_degGe** (hu : ∀ i, constantCoeff (u i) = 0) (n m : ℕ) (j : τ) : ((approxStep u B)^[n + m] 0) j − ((approxStep u B)^[n] 0) j ∈ degGe σ R (n + 1)

theorem **approxLimit_sub_mem_degGe** (hu : ∀ i, constantCoeff (u i) = 0) (n : ℕ) (j : τ) : approxLimit u B j − ((approxStep u B)^[n] 0) j ∈ degGe σ R (n + 1)

theorem **approxLimit_mem_degGe** (hu : ∀ i, constantCoeff (u i) = 0) (j : τ) : approxLimit u B j ∈ degGe σ R 1

theorem **approxStep_approxLimit** (hu : ∀ i, constantCoeff (u i) = 0) (j : τ) : approxStep u B (approxLimit u B) j = approxLimit u B j

theorem **subst_approxLimit** [DecidableEq σ] (hu : ∀ i, constantCoeff (u i) = 0) (hB : linearPart u ∗ B = 1) (i : σ) : (u i).subst (approxLimit u B) = X i

theorem **approxLimit_mem_coeffsInDeg** (hu : ∀ i, constantCoeff (u i) = 0) {S : Subring R} {g : ℕ} (hB : ∀ j i, B j i ∈ S) (huS : ∀ i, higherPart u i ∈ coeffsInDeg τ S g) (j : τ) : approxLimit u B j ∈ coeffsInDeg σ S g

theorem **higherPart_mem_coeffsInDeg** [Fintype σ] {u : ι → MvPowerSeries σ R} {i : ι} (hu : constantCoeff (u i) = 0) {S : Subring R} {g : ℕ} (huS : u i ∈ coeffsInDeg σ S g) : higherPart u i ∈ coeffsInDeg σ S g

theorem **eq_of_subst_eq_X** [Finite σ] [Finite τ] {u w : σ → MvPowerSeries τ R} {v : τ → MvPowerSeries σ R} (hv : ∀ j, constantCoeff (v j) = 0) (hw : ∀ i, constantCoeff (w i) = 0) (huv : ∀ i, (u i).subst v = X i) (hvw : ∀ j, (v j).subst w = X j) : u = w

theorem **constantCoeff_substInv** (hu : ∀ i, constantCoeff (u i) = 0) (i : σ) : constantCoeff (substInv u i) = 0

theorem **hasSubst_substInv** (hu : ∀ i, constantCoeff (u i) = 0) : HasSubst (substInv u)

theorem **subst_substInv_right** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) (i : σ) : (u i).subst (substInv u) = X i

theorem **linearPart_mul_eq_one** (hu : ∀ i, constantCoeff (u i) = 0) (hv : ∀ i, constantCoeff (v i) = 0) (h : ∀ i, (u i).subst v = X i) : linearPart u ∗ linearPart v = 1

theorem **linearPart_substInv** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) : linearPart (substInv u) = (linearPart u)⁻¹

theorem **isUnit_linearPart_substInv** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) : IsUnit (linearPart (substInv u))

theorem **substInv_substInv** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) : substInv (substInv u) = u

theorem **subst_substInv_left** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) (i : σ) : (substInv u i).subst u = X i

theorem **eq_substInv_of_right** (hu : ∀ i, constantCoeff (u i) = 0) (hv : ∀ i, constantCoeff (v i) = 0) (h : ∀ i, (u i).subst v = X i) : v = substInv u

theorem **eq_substInv_of_left** (hu : ∀ i, constantCoeff (u i) = 0) (hv : ∀ i, constantCoeff (v i) = 0) (h : ∀ i, (v i).subst u = X i) : v = substInv u

theorem **exists_substInv_iff** (hu : ∀ i, constantCoeff (u i) = 0) : (∃ v : σ → MvPowerSeries σ R, (∀ i, constantCoeff (v i) = 0) ∧ (∀ i, (u i).subst v = X i) ∧ ∀ i, (v i).subst u = X i) ↔ IsUnit (linearPart u)

theorem **subst_substInv_subst** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) (f : MvPowerSeries σ R) : (f.subst u).subst (substInv u) = f

theorem **subst_subst_substInv** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) (f : MvPowerSeries σ R) : (f.subst (substInv u)).subst u = f

@[simp] theorem **substAlgEquiv_apply** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) (f : MvPowerSeries σ R) : substAlgEquiv hu hA f = f.subst u

@[simp] theorem **substAlgEquiv_symm_apply** (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) (f : MvPowerSeries σ R) : (substAlgEquiv hu hA).symm f = f.subst (substInv u)

theorem **map_substInv** {S : Type∗} [CommRing S] (φ : R →+∗ S) (hu : ∀ i, constantCoeff (u i) = 0) (hA : IsUnit (linearPart u)) : (fun i => MvPowerSeries.map φ (substInv u i)) = substInv fun i => MvPowerSeries.map φ (u i)

theorem **coeff_substInv_mem** (hu : ∀ i, constantCoeff (u i) = 0) {S : Subring R} {g : ℕ} (hB : ∀ i j, (linearPart u)⁻¹ i j ∈ S) (huS : ∀ i (d : σ →₀ ℕ), d.degree < g → coeff d (u i) ∈ S) (i : σ) (d : σ →₀ ℕ) (hd : d.degree < g) : coeff d (substInv u i) ∈ S

theorem **two_le_order_sub_X_iff** : (∀ i, (2 : ℕ∞) ≤ (u i − X i).order) ↔ (∀ i, constantCoeff (u i) = 0) ∧ linearPart u = 1

theorem **two_le_order_substInv_sub_X** (hu : ∀ i, (2 : ℕ∞) ≤ (u i − X i).order) (i : σ) : (2 : ℕ∞) ≤ (substInv u i − X i).order

theorem **coeff_substInv_mem_of_two_le_order** (hu : ∀ i, (2 : ℕ∞) ≤ (u i − X i).order) {S : Subring R} {g : ℕ} (huS : ∀ i (d : σ →₀ ℕ), d.degree < g → coeff d (u i) ∈ S) (i : σ) (d : σ →₀ ℕ) (hd : d.degree < g) : coeff d (substInv u i) ∈ S

@[simp] theorem **sumFamily_inl** [Finite τ] [Finite τ'] (a : σ → MvPowerSeries τ R) (a' : σ' → MvPowerSeries τ' R) (i : σ) : sumFamily a a' (Sum.inl i) = rename Sum.inl (a i)

@[simp] theorem **sumFamily_inr** [Finite τ] [Finite τ'] (a : σ → MvPowerSeries τ R) (a' : σ' → MvPowerSeries τ' R) (j : σ') : sumFamily a a' (Sum.inr j) = rename Sum.inr (a' j)

theorem **constantCoeff_sumFamily** [Finite τ] [Finite τ'] {a : σ → MvPowerSeries τ R} {a' : σ' → MvPowerSeries τ' R} (ha : ∀ i, constantCoeff (a i) = 0) (ha' : ∀ j, constantCoeff (a' j) = 0) (s : σ ⊕ σ') : constantCoeff (sumFamily a a' s) = 0

theorem **mem_coeffsIn_groupWeight** [Finite σ] [Finite σ'] {S : Subring R} {g g' : ℕ} {f : MvPowerSeries (σ ⊕ σ') R} : f ∈ coeffsIn S (groupWeight σ σ') (fun b => bif b then g else g') ↔ ∀ (α : σ →₀ ℕ) (β : σ' →₀ ℕ), α.degree < g → β.degree < g' → coeff (α.sumElim β) f ∈ S

theorem **X_mem_weightGe** (w : σ → ℕ) (s : σ) : (X s : MvPowerSeries σ R) ∈ weightGe w (w s)

theorem **rename_mem_weightGe** [Finite σ] (e : σ → τ) {w : τ → ℕ} {m : ℕ} {f : MvPowerSeries σ R} (hf : f ∈ weightGe (w ∘ e) m) : rename e f ∈ weightGe w m

theorem **rename_mem_coeffsIn** {κ : Type*} [Finite σ] (e : σ → τ) {S : Subring R} {w : κ → τ → ℕ} {g : κ → ℕ} {f : MvPowerSeries σ R} (hf : f ∈ coeffsIn S (fun k => w k ∘ e) g) : rename e f ∈ coeffsIn S w g

theorem **coeff_subst_sumFamily_mem** [Finite σ] [Finite σ'] [Finite τ] [Finite τ'] {S : Subring R} {g g' : ℕ} {a : σ → MvPowerSeries τ R} {a' : σ' → MvPowerSeries τ' R} (ha : ∀ i, constantCoeff (a i) = 0) (ha' : ∀ j, constantCoeff (a' j) = 0) (haS : ∀ i (d : τ →₀ ℕ), d.degree < g → coeff d (a i) ∈ S) (haS' : ∀ j (d : τ' →₀ ℕ), d.degree < g' → coeff d (a' j) ∈ S) {f : MvPowerSeries (σ ⊕ σ') R} (hf : ∀ (α : σ →₀ ℕ) (β : σ' →₀ ℕ), α.degree < g → β.degree < g' → coeff (α.sumElim β) f ∈ S) (α : τ →₀ ℕ) (β : τ' →₀ ℕ) (hα : α.degree < g) (hβ : β.degree < g') : coeff (α.sumElim β) (f.subst (sumFamily a a')) ∈ S

theorem **subst_sumFamily_rename_inl** [Finite σ] [Finite σ'] [Finite τ] [Finite τ'] {a : σ → MvPowerSeries τ R} {a' : σ' → MvPowerSeries τ' R} (ha : ∀ i, constantCoeff (a i) = 0) (ha' : ∀ j, constantCoeff (a' j) = 0) (f : MvPowerSeries σ R) : (rename Sum.inl f).subst (sumFamily a a') = rename Sum.inl (f.subst a)

theorem **subst_sumFamily_rename_inr** [Finite σ] [Finite σ'] [Finite τ] [Finite τ'] {a : σ → MvPowerSeries τ R} {a' : σ' → MvPowerSeries τ' R} (ha : ∀ i, constantCoeff (a i) = 0) (ha' : ∀ j, constantCoeff (a' j) = 0) (f : MvPowerSeries σ' R) : (rename Sum.inr f).subst (sumFamily a a') = rename Sum.inr (f.subst a')

theorem **subst_sumFamily_sumFamily** {υ' : Type*} [Finite σ] [Finite σ'] [Finite τ] [Finite τ'] (b : υ → MvPowerSeries σ R) (b' : υ' → MvPowerSeries σ' R) {a : σ → MvPowerSeries τ R} {a' : σ' → MvPowerSeries τ' R} (ha : ∀ i, constantCoeff (a i) = 0) (ha' : ∀ j, constantCoeff (a' j) = 0) (s : υ ⊕ υ') : (sumFamily b b' s).subst (sumFamily a a') = sumFamily (fun i => (b i).subst a) (fun j => (b' j).subst a') s

theorem **sumFamily_X** [Finite σ] [Finite σ'] (s : σ ⊕ σ') : sumFamily (fun i : σ => (X i : MvPowerSeries σ R)) (fun j : σ' => X j) s = X s

theorem **substInv_sumFamily** [Fintype σ] [Fintype σ'] [DecidableEq σ] [DecidableEq σ'] {a : σ → MvPowerSeries σ R} {a' : σ' → MvPowerSeries σ' R} (ha : ∀ i, constantCoeff (a i) = 0) (ha' : ∀ j, constantCoeff (a' j) = 0) (hA : IsUnit (linearPart a)) (hA' : IsUnit (linearPart a')) : IsUnit (linearPart (sumFamily a a')) ∧ substInv (sumFamily a a') = sumFamily (substInv a) (substInv a')

theorem **formal_inverse_function_theorem** : (∀ (R : Type) [CommRing R] (n : ℕ) (u : Fin n → MvPowerSeries (Fin n) R), (∀ i, constantCoeff (u i) = 0) → IsUnit (Matrix.of fun i j => coeff (single j 1) (u i) : Matrix (Fin n) (Fin n) R) → ∃! v : Fin n → MvPowerSeries (Fin n) R, (∀ i, constantCoeff (v i) = 0) ∧ (∀ i, (u i).subst v = X i) ∧ ∀ i, (v i).subst u = X i) ∧ (∀ (C : Type) [CommRing C] (S : Subring C) (g n m : ℕ) (l : Fin n → MvPowerSeries (Fin n) C) (l' : Fin m → MvPowerSeries (Fin m) C), (∀ i, (2 : ℕ∞) ≤ (l i - X i).order) → (∀ j, (2 : ℕ∞) ≤ (l' j - X j).order) → (∀ i (d : Fin n →₀ ℕ), d.degree < g → coeff d (l i) ∈ S) → (∀ j (d : Fin m →₀ ℕ), d.degree < g → coeff d (l' j) ∈ S) → ∃ (e : Fin n → MvPowerSeries (Fin n) C) (e' : Fin m → MvPowerSeries (Fin m) C), ((∀ i, (2 : ℕ∞) ≤ (e i - X i).order) ∧ (∀ i, (l i).subst e = X i) ∧ (∀ i, (e i).subst l = X i) ∧ ∀ i (d : Fin n →₀ ℕ), d.degree

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ConjTransposeBoostMulSignatureMulBoost

lemma **norm_toEuclideanLin_le** (M : Matrix m n ℂ) (x : EuclideanSpace ℂ n) : ‖toEuclideanLin M x‖ ≤ ‖M‖ * ‖x‖

lemma **dotProduct_mulVec_eq_inner** (M : Matrix n n ℂ) (v : n → ℂ) : star v ⬝ᵥ M *ᵥ v = ⟪(toLp 2 v : EuclideanSpace ℂ n), toEuclideanLin M (toLp 2 v)⟫_ℂ

lemma **inner_conjTranspose_mul** (M N : Matrix m n ℂ) (x y : EuclideanSpace ℂ n) : ⟪x, toEuclideanLin (Mᴴ * N) y⟫_ℂ = ⟪toEuclideanLin M x, toEuclideanLin N y⟫_ℂ

lemma **inner_one_sub** (M : Matrix m n ℂ) (x : EuclideanSpace ℂ n) : ⟪x, toEuclideanLin (1 - Mᴴ * M) x⟫_ℂ = ((‖x‖ ^ 2 - ‖toEuclideanLin M x‖ ^ 2 : ℝ) : ℂ)

lemma **isHermitian_one_sub** [DecidableEq n] (M : Matrix m n ℂ) : (1 - Mᴴ * M).IsHermitian

lemma **posDef_one_sub_conjTranspose_mul** {Z : Matrix m n ℂ} (hZ : ‖Z‖ < 1) : (1 - Zᴴ * Z).PosDef

theorem **norm_lt_one_of_posDef** {Z : Matrix m n ℂ} (hP : (1 - Zᴴ * Z).PosDef) : ‖Z‖ < 1

theorem **posDef_iff_norm_lt_one** (Z : Matrix m n ℂ) : (1 - Zᴴ * Z).PosDef ↔ ‖Z‖ < 1

theorem **l2_opNorm_le_hilbertSchmidt** (Z : Matrix m n ℂ) : ‖Z‖ ≤ Real.sqrt (∑ i, ∑ j, ‖Z i j‖ ^ 2)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ContinuousNormMulOneAddAbsLogPow

theorem **continuous_norm_mul_one_add_abs_log_pow** (N : ℕ) : Continuous fun t : ℂ => ‖t‖ * (1 + |Real.log ‖t‖|) ^ (N + 1)

theorem **logWeight_le_prod** {κ : Type*} (S : Finset κ) (z : κ → ℂ) : lwP₂ S z ≤ ∏ j ∈ S, (1 + |Real.log ‖z j‖|)

theorem **HasLogGrowth.tendsto_prod_mul** {κ X : Type*} [TopologicalSpace X] [LocallyCompactSpace X] {S : Finset κ} {w : X → κ → ℂ} {U : Set X} (hU : IsOpen U) {f : X → ℂ} (hf : HasLogGrowth S w U f) {p : X} (hp : p ∈ U) (hB : p ∈ coordDivisor S w) (hw : ∀ j ∈ S, ContinuousAt (fun x => w x j) p) : Tendsto (fun x => (∏ j ∈ S, w x j) * f x) (𝓝 p) (𝓝 0)

private theorem **tendsto_ite_eq_zero_nhds_zero** {α : Type*} (p : α → ℂ) : Tendsto (fun t : ℂ => fun j => if p j = 0 then t else p j) (𝓝 0) (𝓝 p)

private theorem **ite_eq_zero_notMem_coordDivisor** {α : Type*} (S : Finset α) (p : α → ℂ) {t : ℂ} (ht : t ≠ 0) : (fun j => if p j = 0 then t else p j) ∉ coordDivisor S id

theorem **HasLogGrowth.eq_zero_of_eq_coord_mul** (hU : IsOpen U) {f h : (ι → ℂ) → ℂ} (hf : HasLogGrowth S id U f) (hh : ContinuousOn h U) {k : ι} (he : ∀ z ∈ U, z ∉ coordDivisor S id → h z = z k * f z) {p : ι → ℂ} (hp : p ∈ U) (hk : p k = 0) : h p = 0

theorem **exists_eq_coord_mul_of_isOpen** (hU : IsOpen U) {f : (ι → ℂ) → ℂ} (hf : DifferentiableOn ℂ f U) (i : ι) (h0 : ∀ z ∈ U, z i = 0 → f z = 0) : ∃ g : (ι → ℂ) → ℂ, DifferentiableOn ℂ g U ∧ ∀ z ∈ U, f z = z i * g z

theorem **HasLogGrowth.exists_extension_of_prod_mul** (hU : IsOpen U) {f : (ι → ℂ) → ℂ} (hf : HasLogGrowth S id U f) (l : List ι) (hl : ∀ j ∈ l, j ∈ S) : ∀ h : (ι → ℂ) → ℂ, DifferentiableOn ℂ h U → (∀ z ∈ U, z ∉ coordDivisor S id → h z = (l.map z).prod * f z) → ∃ f' : (ι → ℂ) → ℂ, DifferentiableOn ℂ f' U ∧ Set.EqOn f' f (U \ coordDivisor S id)

theorem **HasLogGrowth.exists_differentiableOn_extension** {ι : Type*} [Fintype ι] {S : Finset ι} {U : Set (ι → ℂ)} (hU : IsOpen U) {f : (ι → ℂ) → ℂ} (hf : DifferentiableOn ℂ f (U \ coordDivisor S id)) (hg : HasLogGrowth S id U f) : ∃ f' : (ι → ℂ) → ℂ, DifferentiableOn ℂ f' U ∧ Set.EqOn f' f (U \ coordDivisor S id)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.DifferentiableOnOfTendstoLocallyUniformlyOn

theorem **norm_fderiv_le_of_ball** {f : E → V} {c : E} {r M : ℝ} (hr : 0 < r) (hf : DifferentiableOn ℂ f (ball c r)) (hM : ∀ z ∈ ball c r, ‖f z‖ ≤ M) : ‖fderiv ℂ f c‖ ≤ 2 * M / r

theorem **norm_sub_sub_fderiv_le_of_ball** {f : E → V} {c z : E} {r M : ℝ} (hf : DifferentiableOn ℂ f (ball c r)) (hM : ∀ z ∈ ball c r, ‖f z‖ ≤ M) (hz : z ∈ ball c r) : ‖f z - f c - fderiv ℂ f c (z - c)‖ ≤ 4 * M * (‖z - c‖ / r) ^ 2

theorem **tendstoUniformlyOn_fderiv_ball** [CompleteSpace V] {ι : Type*} {l : Filter ι} [l.NeBot] {F : ι → E → V} {f : E → V} {c : E} {r : ℝ} (hlim : TendstoUniformlyOn F f l (ball c (2 * r))) (hF : ∀ᶠ n in l, DifferentiableOn ℂ (F n) (ball c (2 * r))) : DifferentiableOn ℂ f (ball c r) ∧ TendstoUniformlyOn (fun n => fderiv ℂ (F n)) (fderiv ℂ f) l (ball c r)

theorem **exists_ball_bound** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) {c : E} (hc : c ∈ U) : ∃ r > 0, ∃ M : ℝ, ball c r ⊆ U ∧ ∀ z ∈ ball c r, ‖f z‖ ≤ M

theorem **differentiableOn_fderiv_apply** [CompleteSpace V] {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) (v : E) : DifferentiableOn ℂ (fun x => fderiv ℂ f x v) U

theorem **contDiffOn_nat** {U : Set E} (hU : IsOpen U) (n : ℕ) : ∀ {f : E → V}, DifferentiableOn ℂ f U → ContDiffOn ℂ n f U

theorem **differentiableOn_iteratedFDeriv** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) (n : ℕ) : DifferentiableOn ℂ (iteratedFDeriv ℂ n f) U

theorem **norm_iteratedFDeriv_le_pow** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) {δ M : ℝ} (hδ : 0 < δ) (n : ℕ) : ∀ x : E, closedBall x (n * δ) ⊆ U → (∀ z ∈ closedBall x (n * δ), ‖f z‖ ≤ M) → ‖iteratedFDeriv ℂ n f x‖ ≤ M * (2 / δ) ^ n

theorem **norm_iteratedFDeriv_le_factorial** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) {c : E} {R M : ℝ} (hR : 0 < R) (hc : closedBall c R ⊆ U) (hM : ∀ z ∈ closedBall c R, ‖f z‖ ≤ M) (n : ℕ) : ‖iteratedFDeriv ℂ n f c‖ ≤ n ! * (M * (2 * Real.exp 1 / R) ^ n)

theorem **iteratedDeriv_line** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) {c : E} (hc : c ∈ U) (y : E) (n : ℕ) : iteratedDeriv n (fun t : ℂ => f (c + t • y)) 0 = iteratedFDeriv ℂ n f c (fun _ => y)

theorem **hasSum_taylorSeries_ball** {f : E → V} {c : E} {R : ℝ} (hf : DifferentiableOn ℂ f (ball c R)) {y : E} (hy : ‖y‖ < R) : HasSum (fun n : ℕ => (n ! : ℂ)⁻¹ • iteratedFDeriv ℂ n f c (fun _ => y)) (f (c + y))

theorem **hasFPowerSeriesAt_taylor** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) {c : E} (hc : c ∈ U) : HasFPowerSeriesAt f (fun n => (n ! : ℂ)⁻¹ • iteratedFDeriv ℂ n f c) c

theorem **analyticOnNhd_of_differentiableOn** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) : AnalyticOnNhd ℂ f U

theorem **analyticOnNhd_iff_differentiableOn** {f : E → V} {U : Set E} (hU : IsOpen U) : AnalyticOnNhd ℂ f U ↔ DifferentiableOn ℂ f U

theorem **analyticAt_of_differentiableOn** {f : E → V} {s : Set E} {c : E} (hf : DifferentiableOn ℂ f s) (hs : s ∈ 𝓝 c) : AnalyticAt ℂ f c

theorem **analyticAt_iff_eventually_differentiableAt** {f : E → V} {c : E} : AnalyticAt ℂ f c ↔ ∀ᶠ z in 𝓝 c, DifferentiableAt ℂ f z

theorem **analyticOnNhd_of_differentiable** {f : E → V} (hf : Differentiable ℂ f) (s : Set E) : AnalyticOnNhd ℂ f s

theorem **contDiffOn_of_differentiableOn** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) (n : WithTop ℕ∞) : ContDiffOn ℂ n f U

theorem **contDiffAt_of_differentiableOn** {f : E → V} {s : Set E} {c : E} (hf : DifferentiableOn ℂ f s) (hs : s ∈ 𝓝 c) (n : WithTop ℕ∞) : ContDiffAt ℂ n f c

theorem **contDiff_of_differentiable** {f : E → V} (hf : Differentiable ℂ f) (n : WithTop ℕ∞) : ContDiff ℂ n f

theorem **differentiableOn_fderiv** {f : E → V} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) : DifferentiableOn ℂ (fderiv ℂ f) U

theorem **eqOn_of_eventuallyEq** {f g : E → V} {U : Set E} (hU : IsOpen U) (hU' : IsPreconnected U) (hf : DifferentiableOn ℂ f U) (hg : DifferentiableOn ℂ g U) {c : E} (hc : c ∈ U) (hfg : f =ᶠ[𝓝 c] g) : EqOn f g U

theorem **eqOn_of_eqOn_open** {f g : E → V} {U W : Set E} (hU : IsOpen U) (hU' : IsPreconnected U) (hf : DifferentiableOn ℂ f U) (hg : DifferentiableOn ℂ g U) (hW : IsOpen W) (hne : W.Nonempty) (hWU : W ⊆ U) (hfg : EqOn f g W) : EqOn f g U

private theorem **exists_ball_limit** [l.NeBot] (hU : IsOpen U) (hlim : TendstoLocallyUniformlyOn F f l U) (hF : ∀ᶠ n in l, DifferentiableOn ℂ (F n) U) {c : E} (hc : c ∈ U) : ∃ r > 0, DifferentiableOn ℂ f (ball c r) ∧ TendstoUniformlyOn (fun n => fderiv ℂ (F n)) (fderiv ℂ f) l (ball c r)

theorem **differentiableOn_of_tendstoLocallyUniformlyOn** [l.NeBot] (hU : IsOpen U) (hlim : TendstoLocallyUniformlyOn F f l U) (hF : ∀ᶠ n in l, DifferentiableOn ℂ (F n) U) : DifferentiableOn ℂ f U

theorem **tendstoLocallyUniformlyOn_fderiv** (hU : IsOpen U) (hlim : TendstoLocallyUniformlyOn F f l U) (hF : ∀ᶠ n in l, DifferentiableOn ℂ (F n) U) : TendstoLocallyUniformlyOn (fun n => fderiv ℂ (F n)) (fderiv ℂ f) l U

theorem **tendstoLocallyUniformlyOn_iteratedFDeriv** (hU : IsOpen U) (hlim : TendstoLocallyUniformlyOn F f l U) (hF : ∀ᶠ n in l, DifferentiableOn ℂ (F n) U) (k : ℕ) : TendstoLocallyUniformlyOn (fun n => iteratedFDeriv ℂ k (F n)) (iteratedFDeriv ℂ k f) l U

theorem **differentiableOn_of_hasSumLocallyUniformlyOn** (hU : IsOpen U) (hsum : HasSumLocallyUniformlyOn F f U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) : DifferentiableOn ℂ f U

theorem **hasSumLocallyUniformlyOn_fderiv** (hU : IsOpen U) (hsum : HasSumLocallyUniformlyOn F f U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) : HasSumLocallyUniformlyOn (fun i => fderiv ℂ (F i)) (fderiv ℂ f) U

theorem **hasSumLocallyUniformlyOn_iteratedFDeriv** (hU : IsOpen U) (hsum : HasSumLocallyUniformlyOn F f U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) (k : ℕ) : HasSumLocallyUniformlyOn (fun i => iteratedFDeriv ℂ k (F i)) (iteratedFDeriv ℂ k f) U

theorem **hasSumLocallyUniformlyOn_of_locally_normal** (hle : ∀ x ∈ U, ∃ t ∈ 𝓝 x, ∃ u : ι → ℝ, Summable u ∧ ∀ i, ∀ z ∈ t, ‖F i z‖ ≤ u i) : HasSumLocallyUniformlyOn F (fun z => ∑' i, F i z) U

theorem **differentiableOn_tsum_of_locally_normal** (hU : IsOpen U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) (hle : ∀ x ∈ U, ∃ t ∈ 𝓝 x, ∃ u : ι → ℝ, Summable u ∧ ∀ i, ∀ z ∈ t, ‖F i z‖ ≤ u i) : DifferentiableOn ℂ (fun z => ∑' i, F i z) U

theorem **hasSum_fderiv_of_locally_normal** (hU : IsOpen U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) (hle : ∀ x ∈ U, ∃ t ∈ 𝓝 x, ∃ u : ι → ℝ, Summable u ∧ ∀ i, ∀ z ∈ t, ‖F i z‖ ≤ u i) {z : E} (hz : z ∈ U) : HasSum (fun i => fderiv ℂ (F i) z) (fderiv ℂ (fun w => ∑' i, F i w) z)

theorem **hasSum_iteratedFDeriv_of_locally_normal** (hU : IsOpen U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) (hle : ∀ x ∈ U, ∃ t ∈ 𝓝 x, ∃ u : ι → ℝ, Summable u ∧ ∀ i, ∀ z ∈ t, ‖F i z‖ ≤ u i) (k : ℕ) {z : E} (hz : z ∈ U) : HasSum (fun i => iteratedFDeriv ℂ k (F i) z) (iteratedFDeriv ℂ k (fun w => ∑' i, F i w) z)

theorem **differentiableOn_tsum_of_normal_on_compacts** (hU : IsOpen U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) (hle : ∀ K ⊆ U, IsCompact K → ∃ u : ι → ℝ, Summable u ∧ ∀ i, ∀ z ∈ K, ‖F i z‖ ≤ u i) : DifferentiableOn ℂ (fun z => ∑' i, F i z) U ∧ ∀ k : ℕ, HasSumLocallyUniformlyOn (fun i => iteratedFDeriv ℂ k (F i)) (iteratedFDeriv ℂ k (fun w => ∑' i, F i w)) U

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.DiscreteTopologySpanOfLinearIndependent

theorem **det_gram_real_sum_smul** (e : ι → E) (m : Matrix ι ι ℝ) : (gram ℝ (fun j => ∑ i, m i j • e i)).det = m.det ^ 2 * (gram ℝ e).det

theorem **free_of_fg** {V : Type*} [AddCommGroup V] [Module ℝ V] {L : Submodule ℤ V} (h : L.FG) : Module.Free ℤ L

theorem **sqrt_det_gram_eq_relIndex_mul** {L' L : Submodule ℤ E} (h : L' ≤ L) (b' : Basis ι ℤ L') (b : Basis ι ℤ L) : √(gram ℝ fun i => (b' i : E)).det = (L'.toAddSubgroup.relIndex L.toAddSubgroup : ℝ) * √(gram ℝ fun i => (b i : E)).det

theorem **spanCovol_eq_sqrt_det_gram** {L : Submodule ℤ E} (b : Basis ι ℤ L) : spanCovol L = √(gram ℝ fun i => (b i : E)).det

theorem **spanCovol_span** {v : ι → E} (hv : LinearIndependent ℤ v) : spanCovol (span ℤ (Set.range v)) = √(gram ℝ v).det

theorem **spanCovol_bot** : spanCovol (⊥ : Submodule ℤ E) = 1

theorem **spanCovol_map_linearIsometry** {L : Submodule ℤ E} (hL : L.FG) (T : E →ₗᵢ[ℝ] F) : spanCovol (L.map (T.toLinearMap.restrictScalars ℤ)) = spanCovol L

theorem **discreteTopology_of_le** {N L : Submodule ℤ V} [DiscreteTopology L] (h : N ≤ L) : DiscreteTopology N

theorem **fg_of_discrete** (L : Submodule ℤ V) [DiscreteTopology L] : L.FG

theorem **exists_basis_range_subset** {ι : Type*} {v : ι → V} (hv : LinearIndependent ℝ v) : ∃ β : Basis (Fin (finrank ℝ V)) ℝ V, Set.range v ⊆ Set.range β

theorem **discreteTopology_span_of_linearIndependent** {ι : Type*} {v : ι → V} (hv : LinearIndependent ℝ v) : DiscreteTopology (span ℤ (Set.range v))

theorem **span_range_coe_basis** {ι : Type*} {L : Submodule ℤ V} (b : Basis ι ℤ L) : span ℤ (Set.range fun i => (b i : V)) = L

theorem **linearIndependent_real_coe_basis** {ι : Type*} [Fintype ι] {L : Submodule ℤ V} [DiscreteTopology L] (b : Basis ι ℤ L) : LinearIndependent ℝ fun i => (b i : V)

theorem **spanCovol_eq_covolume** [MeasurableSpace E] [BorelSpace E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] : spanCovol L = ZLattice.covolume L

theorem **spanCovol_comap_subtype** (L : Submodule ℤ E) [DiscreteTopology L] (S : Submodule ℝ E) (h : (L : Set E) ⊆ S) : spanCovol (ZLattice.comap ℝ L S.subtype) = spanCovol L

theorem **spanCovol_eq_covolume_comap_span** [MeasurableSpace E] [BorelSpace E] (L : Submodule ℤ E) [DiscreteTopology L] : spanCovol L = ZLattice.covolume (ZLattice.comap ℝ L (span ℝ (L : Set E)).subtype)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.DualRestrictSurjectiveOfIsTorsionFreeQuotient

theorem **standardAltForm_apply** [Fintype ι] [DecidableEq ι] (d : ι → R) (x y : ι ⊕ ι → R) : standardAltForm d x y = ∑ i, d i * (x (Sum.inl i) * y (Sum.inr i) - x (Sum.inr i) * y (Sum.inl i))

theorem **isAlt_standardAltForm** [Fintype ι] [DecidableEq ι] (d : ι → R) : (standardAltForm d).IsAlt

theorem **isSymplecticBasisOfType_standardAltForm** [Fintype ι] [DecidableEq ι] (d : ι → R) : IsSymplecticBasisOfType (standardAltForm d) (Pi.basisFun R (ι ⊕ ι)) d

theorem **left_right** [DecidableEq ι] (h : IsSymplecticFamily B e f d) (i j : ι) : B (e i) (f j) = if i = j then d i else 0

theorem **right_left** [DecidableEq ι] (h : IsSymplecticFamily B e f d) (hB : B.IsAlt) (i j : ι) : B (f i) (e j) = if i = j then -d i else 0

theorem **swap** (h : IsSymplecticFamily B e f d) (hB : B.IsAlt) : IsSymplecticFamily (-B) f e d where

theorem **apply_left_sum** [Fintype ι] (h : IsSymplecticFamily B e f d) (c : ι ⊕ ι → R) (k : ι) : B (e k) (∑ i, c i • Sum.elim e f i) = c (Sum.inr k) * d k

theorem **apply_sum_right** [Fintype ι] (h : IsSymplecticFamily B e f d) (c : ι ⊕ ι → R) (k : ι) : B (∑ i, c i • Sum.elim e f i) (f k) = c (Sum.inl k) * d k

theorem **linearIndependent** [Finite ι] [IsDomain R] (h : IsSymplecticFamily B e f d) (hd : ∀ i, d i ≠ 0) : LinearIndependent R (Sum.elim e f)

theorem **sum** {κ : Type*} {e' f' : κ → M} {d' : κ → R} (hB : B.IsAlt) (h : IsSymplecticFamily B e f d) (h' : IsSymplecticFamily B e' f' d') (h₁ : ∀ i j, B (e i) (e' j) = 0) (h₂ : ∀ i j, B (e i) (f' j) = 0) (h₃ : ∀ i j, B (f i) (e' j) = 0) (h₄ : ∀ i j, B (f i) (f' j) = 0) : IsSymplecticFamily B (Sum.elim e e') (Sum.elim f f') (Sum.elim d d') where

theorem **exists_pair_dvd_of_bilinForm** (B : LinearMap.BilinForm R M) (N : Submodule R M) : ∃ e ∈ N, ∃ f ∈ N, (∀ y ∈ N, B e f ∣ B e y) ∧ (∀ x ∈ N, B e f ∣ B x f) ∧ (∀ x ∈ N, ∀ y ∈ N, B e y = 0 → B x f = 0 → B e f ∣ B x y)

theorem **exists_isSymplecticFamily_span_eq** [Module.Free R M] [Module.Finite R M] {B : LinearMap.BilinForm R M} (hB : B.IsAlt) (N : Submodule R M) (hN : ∀ x ∈ N, (∀ y ∈ N, B x y = 0) → x = 0) : ∃ (g : ℕ) (e f : Fin g → M) (d : Fin g → R), IsSymplecticFamily B e f d ∧ (∀ i, d i ≠ 0) ∧ (∀ i j, i ≤ j → d i ∣ d j) ∧ Submodule.span R (Set.range (Sum.elim e f)) = N

theorem **elim_eq** (b : Basis (ι ⊕ ι) R M) : Sum.elim (fun i => b (Sum.inl i)) (fun i => b (Sum.inr i)) = b

theorem **apply_left** [Finite ι] (h : IsSymplecticBasisOfType B b d) (x : M) (k : ι) : B (b (Sum.inl k)) x = b.repr x (Sum.inr k) * d k

theorem **apply_right** [Finite ι] (h : IsSymplecticBasisOfType B b d) (x : M) (k : ι) : B x (b (Sum.inr k)) = b.repr x (Sum.inl k) * d k

theorem **apply_eq_sum** [Fintype ι] (h : IsSymplecticBasisOfType B b d) (hB : B.IsAlt) (x y : M) : B x y = ∑ i, d i * (b.repr x (Sum.inl i) * b.repr y (Sum.inr i) - b.repr x (Sum.inr i) * b.repr y (Sum.inl i))

theorem **toMatrix_eq** [Fintype ι] [DecidableEq ι] (h : IsSymplecticBasisOfType B b d) (hB : B.IsAlt) : LinearMap.BilinForm.toMatrix b B = Matrix.fromBlocks 0 (Matrix.diagonal d) (-Matrix.diagonal d) 0

theorem **swap** (h : IsSymplecticBasisOfType B b d) (hB : B.IsAlt) : IsSymplecticBasisOfType (-B) (b.reindex (Equiv.sumComm ι ι)) d

theorem **reindex** {κ : Type*} (h : IsSymplecticBasisOfType B b d) (σ : ι ≃ κ) : IsSymplecticBasisOfType B (b.reindex (σ.sumCongr σ)) fun k => d (σ.symm k)

theorem **isUnit** [Finite ι] (h : IsSymplecticBasisOfType B b d) (hp : Surjective B) (k : ι) : IsUnit (d k)

theorem **finrank_eq** [Fintype ι] [StrongRankCondition R] (b : Basis (ι ⊕ ι) R M) : finrank R M = 2 * Fintype.card ι

theorem **left_left** (h : IsSymplecticBasis B b) (i j : ι) : B (b (Sum.inl i)) (b (Sum.inl j)) = 0

theorem **right_right** (h : IsSymplecticBasis B b) (i j : ι) : B (b (Sum.inr i)) (b (Sum.inr j)) = 0

theorem **left_right** [DecidableEq ι] (h : IsSymplecticBasis B b) (i j : ι) : B (b (Sum.inl i)) (b (Sum.inr j)) = if i = j then 1 else 0

theorem **right_left** [DecidableEq ι] (h : IsSymplecticBasis B b) (hB : B.IsAlt) (i j : ι) : B (b (Sum.inr i)) (b (Sum.inl j)) = if i = j then -1 else 0

theorem **apply_left** [Finite ι] (h : IsSymplecticBasis B b) (x : M) (k : ι) : B (b (Sum.inl k)) x = b.repr x (Sum.inr k)

theorem **apply_right** [Finite ι] (h : IsSymplecticBasis B b) (x : M) (k : ι) : B x (b (Sum.inr k)) = b.repr x (Sum.inl k)

theorem **apply_eq_sum** [Fintype ι] (h : IsSymplecticBasis B b) (hB : B.IsAlt) (x y : M) : B x y = ∑ i, (b.repr x (Sum.inl i) * b.repr y (Sum.inr i) - b.repr x (Sum.inr i) * b.repr y (Sum.inl i))

theorem **toMatrix_eq** [Fintype ι] [DecidableEq ι] (h : IsSymplecticBasis B b) (hB : B.IsAlt) : LinearMap.BilinForm.toMatrix b B = -Matrix.J ι R

theorem **swap** (h : IsSymplecticBasis B b) (hB : B.IsAlt) : IsSymplecticBasis (-B) (b.reindex (Equiv.sumComm ι ι))

theorem **reindex** {κ : Type*} (h : IsSymplecticBasis B b) (σ : ι ≃ κ) : IsSymplecticBasis B (b.reindex (σ.sumCongr σ))

theorem **bijective** [Finite ι] (h : IsSymplecticBasis B b) (hB : B.IsAlt) : Bijective B

theorem **bijective_standardAltForm_one** [Fintype ι] [DecidableEq ι] : Bijective (standardAltForm fun _ : ι => (1 : R))

theorem **bijective_neg_bilinForm** (h : Bijective B) : Bijective (-B)

theorem **IsSymplecticFamily.exists_isSymplecticBasis** [IsDomain R] [Finite ι] {e f : ι → M} {d : ι → R} (h : IsSymplecticFamily B e f d) (hd : ∀ i, d i ≠ 0) (hspan : Submodule.span R (Set.range (Sum.elim e f)) = ⊤) (hp : Surjective B) : ∃ b : Basis (ι ⊕ ι) R M, IsSymplecticBasis B b ∧ ∀ i, b (Sum.inl i) = e i

theorem **exists_isSymplecticBasisOfType** (hB : B.IsAlt) (hnd : Injective B) : ∃ (g : ℕ) (b : Basis (Fin g ⊕ Fin g) R M) (d : Fin g → R), IsSymplecticBasisOfType B b d ∧ (∀ i, d i ≠ 0) ∧ ∀ i j, i ≤ j → d i ∣ d j

theorem **exists_isSymplecticBasisOfType_pos** {M : Type*} [AddCommGroup M] [Module.Free ℤ M] [Module.Finite ℤ M] {B : LinearMap.BilinForm ℤ M} (hB : B.IsAlt) (hnd : Injective B) : ∃ (g : ℕ) (b : Basis (Fin g ⊕ Fin g) ℤ M) (d : Fin g → ℤ), IsSymplecticBasisOfType B b d ∧ (∀ i, 0 < d i) ∧ ∀ i j, i ≤ j → d i ∣ d j

theorem **exists_isSymplecticBasis** (hB : B.IsAlt) (hp : Bijective B) : ∃ (g : ℕ) (b : Basis (Fin g ⊕ Fin g) R M), IsSymplecticBasis B b

theorem **exists_isSymplecticBasis_of_finrank** (hB : B.IsAlt) (hp : Bijective B) {g : ℕ} (hg : finrank R M = 2 * g) : ∃ b : Basis (Fin g ⊕ Fin g) R M, IsSymplecticBasis B b

theorem **quotientOrthogonalToDual_mk** (B : LinearMap.BilinForm R M) (hB : B.IsAlt) (I : Submodule R M) (y : M) (x : I) : quotientOrthogonalToDual B hB I (Submodule.Quotient.mk y) x = B y x

theorem **orthogonalQuotientForm_mk** (B : LinearMap.BilinForm R M) (hB : B.IsAlt) (I : Submodule R M) (x y : B.orthogonal I) : orthogonalQuotientForm B hB I (Submodule.Quotient.mk x) (Submodule.Quotient.mk y) = B x y

theorem **isTorsionFree_quotient_iff** [IsDomain R] (I : Submodule R M) : IsTorsionFree R (M ⧸ I) ↔ ∀ (r : R) (x : M), r ≠ 0 → r • x ∈ I → x ∈ I

theorem **isTorsionFree_quotient_orthogonal** [IsDomain R] (B : LinearMap.BilinForm R M) (I : Submodule R M) : IsTorsionFree R (M ⧸ B.orthogonal I)

theorem **exists_projection_of_isTorsionFree_quotient** (hI : IsTorsionFree R (M ⧸ I)) : ∃ p : M →ₗ[R] I, ∀ x : I, p x = x

theorem **dualRestrict_surjective_of_isTorsionFree_quotient** (hI : IsTorsionFree R (M ⧸ I)) : Surjective I.dualRestrict

theorem **orthogonal_orthogonal_of_isTorsionFree_quotient** (hB : B.IsAlt) (hp : Surjective B) (hI : IsTorsionFree R (M ⧸ I)) : B.orthogonal (B.orthogonal I) = I

theorem **exists_isSymplecticFamily_of_isotropic** {ι : Type*} [Finite ι] (hB : B.IsAlt) (hp : Surjective B) (hI : IsTorsionFree R (M ⧸ I)) (hiso : I ≤ B.orthogonal I) (v : Basis ι R I) : ∃ f : ι → M, IsSymplecticFamily B (fun i => (v i : M)) f fun _ => 1

theorem **exists_isSymplecticBasis_extend** [Module.Free R M] {ι : Type*} [Finite ι] (hB : B.IsAlt) (hp : Bijective B) (hI : IsTorsionFree R (M ⧸ I)) (hiso : I ≤ B.orthogonal I) (v : Basis ι R I) : ∃ (g : ℕ) (b : Basis ((ι ⊕ Fin g) ⊕ (ι ⊕ Fin g)) R M), IsSymplecticBasis B b ∧ ∀ i, b (Sum.inl (Sum.inl i)) = v i

theorem **IsSymplecticBasis.mem_orthogonal_iff** {ι κ : Type*} [Finite ι] [Finite κ] {b : Basis ((ι ⊕ κ) ⊕ (ι ⊕ κ)) R M} (hb : IsSymplecticBasis B b) {v : Basis ι R I} (hv : ∀ i, b (Sum.inl (Sum.inl i)) = v i) (x : M) : x ∈ B.orthogonal I ↔ ∀ i, b.repr x (Sum.inr (Sum.inl i)) = 0

theorem **IsSymplecticBasis.orthogonal_eq_span** {ι κ : Type*} [Finite ι] [Finite κ] {b : Basis ((ι ⊕ κ) ⊕ (ι ⊕ κ)) R M} (hb : IsSymplecticBasis B b) {v : Basis ι R I} (hv : ∀ i, b (Sum.inl (Sum.inl i)) = v i) : B.orthogonal I = Submodule.span R (Set.range (fun i => b (Sum.inl i)) ∪ Set.range fun k => b (Sum.inr (Sum.inr k)))

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EncardSmulMemLeEncardSmulInterNonempty

theorem **encard_smul_mem_le_encard_smul_inter_nonempty** (B : Set X) (x : X) : {g : G | g • x ∈ B}.encard ≤ {g : G | (g • B ∩ B).Nonempty}.encard

theorem **tsum_indicator_smul_eq_encard** (B : Set X) (x : X) : ∑' g : G, ((fun y => g • y) ⁻¹' B).indicator (fun _ => (1 : ℝ≥0∞)) x = ({g : G | g • x ∈ B}.encard : ℝ≥0∞)

theorem **sum_restrict_preimage_smul_le** (hB : NullMeasurableSet B μ) {m : ℝ≥0∞} (hm : ∀ᵐ x ∂μ.restrict D, ({g : G | g • x ∈ B}.encard : ℝ≥0∞) ≤ m) : (Measure.sum fun g : G => (μ.restrict D).restrict ((fun y => g • y) ⁻¹' B)) ≤ m • μ.restrict D

theorem **setLIntegral_le_mul_of_ae_covers** (hD : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D) (hB : NullMeasurableSet B μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) {m : ℝ≥0∞} (hm : ∀ᵐ x ∂μ.restrict D, ({g : G | g • x ∈ B}.encard : ℝ≥0∞) ≤ m) : ∫⁻ x in B, u x ∂μ ≤ m * ∫⁻ x in D, u x ∂μ

theorem **setIntegral_le_mul_of_ae_covers** (hD : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D) (hB : NullMeasurableSet B μ) {u : X → ℝ} (hu : ∀ (g : G) (x : X), u (g • x) = u x) (hu0 : ∀ x, 0 ≤ u x) (huB : AEStronglyMeasurable u (μ.restrict B)) (huD : IntegrableOn u D μ) {m : ℕ} (hm : ∀ᵐ x ∂μ.restrict D, {g : G | g • x ∈ B}.encard ≤ m) : IntegrableOn u B μ ∧ ∫ x in B, u x ∂μ ≤ m * ∫ x in D, u x ∂μ

theorem **setLIntegral_le_mul_of_isFundamentalDomain** (hD : IsFundamentalDomain G D μ) (hB : NullMeasurableSet B μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) {m : ℝ≥0∞} (hm : ∀ x, ({g : G | g • x ∈ B}.encard : ℝ≥0∞) ≤ m) : ∫⁻ x in B, u x ∂μ ≤ m * ∫⁻ x in D, u x ∂μ

theorem **setLIntegral_le_encard_mul_of_ae_covers** (hD : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D) (hB : NullMeasurableSet B μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ∫⁻ x in B, u x ∂μ ≤ ({g : G | (g • B ∩ B).Nonempty}.encard : ℝ≥0∞) * ∫⁻ x in D, u x ∂μ

theorem **ae_encard_smul_mem_le_one** (hD : IsFundamentalDomain G D μ) : ∀ᵐ x ∂μ, {g : G | g • x ∈ D}.encard ≤ 1

theorem **setLIntegral_le_of_isFundamentalDomain_of_ae_covers** {D' : Set X} (hD : IsFundamentalDomain G D μ) (hD' : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D') {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ∫⁻ x in D, u x ∂μ ≤ ∫⁻ x in D', u x ∂μ

theorem **measure_le_of_isFundamentalDomain_of_ae_covers** {D' : Set X} (hD : IsFundamentalDomain G D μ) (hD' : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D') : μ D ≤ μ D'

theorem **pairwise_aedisjoint_out_inv_smul** (hD : IsFundamentalDomain G D μ) (K : Subgroup G) : Pairwise (Function.onFun (AEDisjoint μ) fun q : G ⧸ K => (q.out)⁻¹ • D)

theorem **isFundamentalDomain_iUnion_out_inv_smul** (hD : IsFundamentalDomain G D μ) (K : Subgroup G) : IsFundamentalDomain K (⋃ q : G ⧸ K, (q.out)⁻¹ • D) μ where

theorem **setLIntegral_subgroup_eq_tsum** (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ∫⁻ x in D', u x ∂μ = ∑' _ : G ⧸ K, ∫⁻ x in D, u x ∂μ

theorem **setLIntegral_subgroup_eq_index_mul** [K.FiniteIndex] (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ∫⁻ x in D', u x ∂μ = (K.index : ℝ≥0∞) * ∫⁻ x in D, u x ∂μ

theorem **measure_subgroup_eq_index_mul** [K.FiniteIndex] (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) : μ D' = (K.index : ℝ≥0∞) * μ D

theorem **setLAverage_subgroup_eq** [K.FiniteIndex] (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ⨍⁻ x in D', u x ∂μ = ⨍⁻ x in D, u x ∂μ

theorem **measure_le_measure_subgroup** (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) : μ D ≤ μ D'

theorem **integrableOn_subgroup_iff** [K.FiniteIndex] (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) {f : X → E} (hf : ∀ (g : G) (x : X), f (g • x) = f x) : IntegrableOn f D' μ ↔ IntegrableOn f D μ

theorem **setIntegral_subgroup_eq_index_smul** [K.FiniteIndex] (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) {f : X → E} (hf : ∀ (g : G) (x : X), f (g • x) = f x) : ∫ x in D', f x ∂μ = (K.index : ℝ) • ∫ x in D, f x ∂μ

theorem **setAverage_subgroup_eq** [K.FiniteIndex] (hD : IsFundamentalDomain G D μ) (hD' : IsFundamentalDomain K D' μ) {f : X → E} (hf : ∀ (g : G) (x : X), f (g • x) = f x) : ⨍ x in D', f x ∂μ = ⨍ x in D, f x ∂μ

theorem **measurableConstSMul_of_surjective** [MeasurableConstSMul G X] (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) : MeasurableConstSMul H X where

theorem **smulInvariantMeasure_of_surjective** [SMulInvariantMeasure G X μ] (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) : SMulInvariantMeasure H X μ where

theorem **ae_exists_smul_mem_of_surjective** (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D

theorem **setLIntegral_le_mul_of_surjective** (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hB : NullMeasurableSet B μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) {m : ℝ≥0∞} (hm : ∀ x, ({g : G | g • x ∈ B}.encard : ℝ≥0∞) ≤ m) : ∫⁻ x in B, u x ∂μ ≤ m * ∫⁻ x in D, u x ∂μ

theorem **isFundamentalDomain_map_of_surjective** (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hφ' : Function.Surjective φ') (hφs' : ∀ (k : K) (x : X), φ' k • x = k • x) (hD' : IsFundamentalDomain H' D' μ) : IsFundamentalDomain (K.map φ) D' μ where

theorem **finiteIndex_map_of_surjective** [K.FiniteIndex] (hφ : Function.Surjective φ) : (K.map φ).FiniteIndex

theorem **setLIntegral_eq_index_map_mul** [K.FiniteIndex] (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hφ' : Function.Surjective φ') (hφs' : ∀ (k : K) (x : X), φ' k • x = k • x) (hD' : IsFundamentalDomain H' D' μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ∫⁻ x in D', u x ∂μ = ((K.map φ).index : ℝ≥0∞) * ∫⁻ x in D, u x ∂μ

theorem **measure_eq_index_map_mul** [K.FiniteIndex] (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hφ' : Function.Surjective φ') (hφs' : ∀ (k : K) (x : X), φ' k • x = k • x) (hD' : IsFundamentalDomain H' D' μ) : μ D' = ((K.map φ).index : ℝ≥0∞) * μ D

theorem **setLAverage_eq_of_surjective** [K.FiniteIndex] (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hφ' : Function.Surjective φ') (hφs' : ∀ (k : K) (x : X), φ' k • x = k • x) (hD' : IsFundamentalDomain H' D' μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ⨍⁻ x in D', u x ∂μ = ⨍⁻ x in D, u x ∂μ

theorem **measure_le_measure_of_surjective** (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hφ' : Function.Surjective φ') (hφs' : ∀ (k : K) (x : X), φ' k • x = k • x) (hD' : IsFundamentalDomain H' D' μ) : μ D ≤ μ D'

theorem **setAverage_eq_of_surjective** [K.FiniteIndex] (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) (hφ' : Function.Surjective φ') (hφs' : ∀ (k : K) (x : X), φ' k • x = k • x) (hD' : IsFundamentalDomain H' D' μ) {E : Type*} [NormedAddCommGroup E] [NormedSpace ℝ E] {f : X → E} (hf : ∀ (g : G) (x : X), f (g • x) = f x) : (IntegrableOn f D' μ ↔ IntegrableOn f D μ) ∧ ⨍ x in D', f x ∂μ = ((K.map φ).index : ℝ) • ⨍ x in D, f x ∂μ ∧ ⨍⁻ x in D', f x ∂μ = ⨍⁻ x in D, f x ∂μ

theorem **unfolding_isFundamentalDomain** (hφ : Function.Surjective φ) (hφs : ∀ (g : G) (x : X), φ g • x = g • x) (hD : IsFundamentalDomain H D μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : (∀ B : Set X, NullMeasurableSet B μ → ∀ m : ℝ≥0∞, (∀ x, ({g : G | g • x ∈ B}.encard : ℝ≥0∞) ≤ m) → ∫⁻ x in B, u x ∂μ ≤ m * ∫⁻ x in D, u x ∂μ) ∧ (∀ B : Set X, NullMeasurableSet B μ → ∫⁻ x in B, u x ∂μ ≤ (⨆ g : G | (g • B ∩ B).Nonempty}.encard : ℝ≥0∞) * ∫⁻ x in D, u x ∂μ) ∧ (Function.Surjective φ' → (∀ (k : K) (x : X), φ' k • x = k • x) → IsFundamentalDomain H' D' μ → μ D ≤ μ D' ∧ (K.FiniteIndex → (∃ n : ℕ, 0 < n ∧ n ∣ K.index ∧ μ D' = n * μ D ∧ ∫⁻ x in D', u x ∂μ = n * ∫⁻ x in D, u x ∂μ) ∧ ⨍⁻ x in D', u x ∂μ = ⨍⁻ x in D, u x ∂μ))

theorem **setLIntegral_le_mul_of_range_toPermHom** (hD : IsFundamentalDomain (MulAction.toPermHom G X).range D μ) (hB : NullMeasurableSet B μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) {m : ℝ≥0∞} (hm : ∀ x, ({g : G | g • x ∈ B}.encard : ℝ≥0∞) ≤ m) : ∫⁻ x in B, u x ∂μ ≤ m * ∫⁻ x in D, u x ∂μ

theorem **measure_le_measure_of_range_toPermHom** (hD : IsFundamentalDomain (MulAction.toPermHom G X).range D μ) (hD' : IsFundamentalDomain (MulAction.toPermHom K X).range D' μ) : μ D ≤ μ D'

theorem **setLAverage_eq_of_range_toPermHom** [K.FiniteIndex] (hD : IsFundamentalDomain (MulAction.toPermHom G X).range D μ) (hD' : IsFundamentalDomain (MulAction.toPermHom K X).range D' μ) {u : X → ℝ≥0∞} (hu : ∀ (g : G) (x : X), u (g • x) = u x) : ⨍⁻ x in D', u x ∂μ = ⨍⁻ x in D, u x ∂μ

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EnormPowLeInvMeasureMulLintegralBall

theorem **enorm_pow_mul_le_lintegral_circle** {f : ℂ → H} (hf : DiffContOnCl ℂ f (ball 0 1)) (n : ℕ) : ‖f 0‖ₑ ^ n * ENNReal.ofReal (2 * π) ≤ ∫⁻ t in Ioc 0 (2 * π), ‖f (circleMap 0 1 t)‖ₑ ^ n

theorem **measurePreserving_smul_of_norm_eq_one** (μ : Measure E) [μ.IsAddHaarMeasure] {c : ℂ} (hc : ‖c‖ = 1) : MeasurePreserving (fun z : E => c • z) μ μ

theorem **enorm_pow_mul_lintegral_le_of_circular** (μ : Measure E) [μ.IsAddHaarMeasure] {S : Set E} (hS : MeasurableSet S) (hSc : ∀ c : ℂ, ‖c‖ = 1 → ∀ z ∈ S, c • z ∈ S) {φ : E → H} (hφc : ContinuousOn φ S) (hφ : ∀ z ∈ S, DiffContOnCl ℂ (fun c : ℂ => φ (c • z)) (ball 0 1)) {w : E → ℝ≥0∞} (hw : Measurable w) (hwc : ∀ c : ℂ, ‖c‖ = 1 → ∀ z ∈ S, w (c • z) = w z) (n : ℕ) : ‖φ 0‖ₑ ^ n * ∫⁻ z in S, w z ∂μ ≤ ∫⁻ z in S, ‖φ z‖ₑ ^ n * w z ∂μ

theorem **enorm_pow_mul_lintegral_le_of_balanced** (μ : Measure E) [μ.IsAddHaarMeasure] {S : Set E} (hS : MeasurableSet S) (hSc : Balanced ℂ S) {φ : E → H} (hφ : ∀ z ∈ S, DifferentiableAt ℂ φ z) {w : E → ℝ≥0∞} (hw : Measurable w) (hwc : ∀ c : ℂ, ‖c‖ = 1 → ∀ z ∈ S, w (c • z) = w z) (n : ℕ) : ‖φ 0‖ₑ ^ n * ∫⁻ z in S, w z ∂μ ≤ ∫⁻ z in S, ‖φ z‖ₑ ^ n * w z ∂μ

theorem **enorm_pow_mul_lintegral_ball_le** (μ : Measure E) [μ.IsAddHaarMeasure] {r : ℝ} {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball 0 r)) {w : ℝ → ℝ≥0∞} (hw : Measurable w) (n : ℕ) : ‖φ 0‖ₑ ^ n * ∫⁻ z in ball 0 r, w ‖z‖ ∂μ ≤ ∫⁻ z in ball 0 r, ‖φ z‖ₑ ^ n * w ‖z‖ ∂μ

theorem **enorm_pow_mul_lintegral_closedBall_le** (μ : Measure E) [μ.IsAddHaarMeasure] {r : ℝ} {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball 0 r)) {w : ℝ → ℝ≥0∞} (hw : Measurable w) (n : ℕ) : ‖φ 0‖ₑ ^ n * ∫⁻ z in closedBall 0 r, w ‖z‖ ∂μ ≤ ∫⁻ z in closedBall 0 r, ‖φ z‖ₑ ^ n * w ‖z‖ ∂μ

theorem **enorm_pow_mul_lintegral_ball_le_of_centre** (μ : Measure E) [μ.IsAddHaarMeasure] {a : E} {r : ℝ} {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball a r)) {w : ℝ → ℝ≥0∞} (hw : Measurable w) (n : ℕ) : ‖φ a‖ₑ ^ n * ∫⁻ z in ball a r, w ‖z - a‖ ∂μ ≤ ∫⁻ z in ball a r, ‖φ z‖ₑ ^ n * w ‖z - a‖ ∂μ

theorem **enorm_pow_le_inv_measure_mul_lintegral_ball** (μ : Measure E) [μ.IsAddHaarMeasure] {a : E} {r : ℝ} (hr : 0 < r) {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball a r)) (n : ℕ) : ‖φ a‖ₑ ^ n ≤ (μ (ball a r))⁻¹ * ∫⁻ z in ball a r, ‖φ z‖ₑ ^ n ∂μ

theorem **norm_pow_mul_integral_ball_le** (μ : Measure E) [μ.IsAddHaarMeasure] {a : E} {r : ℝ} {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball a r)) {w : ℝ → ℝ} (hw : Measurable w) (hw0 : ∀ t, 0 ≤ t → 0 ≤ w t) (n : ℕ) (hi : IntegrableOn (fun z => w ‖z - a‖) (ball a r) μ) (hi' : IntegrableOn (fun z => ‖φ z‖ ^ n * w ‖z - a‖) (ball a r) μ) : ‖φ a‖ ^ n * ∫ z in ball a r, w ‖z - a‖ ∂μ ≤ ∫ z in ball a r, ‖φ z‖ ^ n * w ‖z - a‖ ∂μ

theorem **norm_pow_le_inv_measure_mul_integral_ball** (μ : Measure E) [μ.IsAddHaarMeasure] {a : E} {r : ℝ} (hr : 0 < r) {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball a r)) (n : ℕ) (hi : IntegrableOn (fun z => ‖φ z‖ ^ n) (ball a r) μ) : ‖φ a‖ ^ n ≤ (μ.real (ball a r))⁻¹ * ∫ z in ball a r, ‖φ z‖ ^ n ∂μ

theorem **integrableOn_norm_pow_ball** {H : Type*} [NormedAddCommGroup H] (μ : Measure E) [μ.IsAddHaarMeasure] {a : E} {r : ℝ} {φ : E → H} (hφ : ContinuousOn φ (closedBall a r)) (n : ℕ) : IntegrableOn (fun z => ‖φ z‖ ^ n) (ball a r) μ

theorem **volume_ball_of_finrank_eq_two_mul** {D : ℕ} (hD : Module.finrank ℝ E = 2 * D) (a : E) {r : ℝ} (hr : 0 < r) : volume (ball a r) = ENNReal.ofReal (π ^ D * r ^ (2 * D) / D.factorial)

theorem **volume_closedBall_of_finrank_eq_two_mul** {D : ℕ} (hD : Module.finrank ℝ E = 2 * D) (a : E) {r : ℝ} (hr : 0 < r) : volume (closedBall a r) = ENNReal.ofReal (π ^ D * r ^ (2 * D) / D.factorial)

theorem **volume_ball_complex** (a : E) {r : ℝ} (hr : 0 < r) : volume (ball a r) = ENNReal.ofReal (π ^ Module.finrank ℂ E * r ^ (2 * Module.finrank ℂ E) / (Module.finrank ℂ E).factorial)

theorem **norm_pow_le_factorial_div_mul_integral_ball** {a : E} {r : ℝ} (hr : 0 < r) {φ : E → H} (hφ : DifferentiableOn ℂ φ (ball a r)) (n : ℕ) (hi : IntegrableOn (fun z => ‖φ z‖ ^ n) (ball a r)) : ‖φ a‖ ^ n ≤ (Module.finrank ℂ E).factorial / (π ^ Module.finrank ℂ E * r ^ (2 * Module.finrank ℂ E)) * ∫ z in ball a r, ‖φ z‖ ^ n

theorem **volume_ball_euclideanSpace_complex** (a : EuclideanSpace ℂ ι) {r : ℝ} (hr : 0 < r) : volume (ball a r) = ENNReal.ofReal (π ^ Fintype.card ι * r ^ (2 * Fintype.card ι) / (Fintype.card ι).factorial)

theorem **volume_closedBall_euclideanSpace_complex** (a : EuclideanSpace ℂ ι) {r : ℝ} (hr : 0 < r) : volume (closedBall a r) = ENNReal.ofReal (π ^ Fintype.card ι * r ^ (2 * Fintype.card ι) / (Fintype.card ι).factorial)

theorem **measurePreserving_toLp_complex** : MeasurePreserving (WithLp.toLp 2 : (ι → ℂ) → EuclideanSpace ℂ ι)

theorem **preimage_toLp_ball_complex** {r : ℝ} (hr : 0 < r) : (WithLp.toLp 2 : (ι → ℂ) → EuclideanSpace ℂ ι) ⁻¹' ball 0 r = {x : ι → ℂ | ∑ i, ‖x i‖ ^ 2 < r ^ 2}

theorem **volume_sum_norm_sq_lt** {r : ℝ} (hr : 0 < r) : volume {x : ι → ℂ | ∑ i, ‖x i‖ ^ 2 < r ^ 2} = ENNReal.ofReal (π ^ Fintype.card ι * r ^ (2 * Fintype.card ι) / (Fintype.card ι).factorial)

theorem **enorm_pow_mul_lintegral_sum_norm_sq_lt_le** {H : Type*} [NormedAddCommGroup H] [NormedSpace ℂ H] [CompleteSpace H] {R : ℝ} {φ : (ι → ℂ) → H} (hφ : DifferentiableOn ℂ φ {x | ∑ i, ‖x i‖ ^ 2 < R}) {w : ℝ → ℝ≥0∞} (hw : Measurable w) (n : ℕ) : ‖φ 0‖ₑ ^ n * ∫⁻ z in {x : ι → ℂ | ∑ i, ‖x i‖ ^ 2 < R}, w (∑ i, ‖z i‖ ^ 2) ≤ ∫⁻ z in {x : ι → ℂ | ∑ i, ‖x i‖ ^ 2 < R}, ‖φ z‖ₑ ^ n * w (∑ i, ‖z i‖ ^ 2)

theorem **submean_sq_closedBall** {D : ℕ} {H : Type*} [NormedAddCommGroup H] [NormedSpace ℂ H] [CompleteSpace H] {r : ℝ} (hr : 0 < r) {φ : EuclideanSpace ℂ (Fin D) → H} (hφ : DifferentiableOn ℂ φ (ball 0 r)) : (∀ w : ℝ → ℝ≥0∞, Measurable w → ‖φ 0‖ₑ ^ 2 ∫⁻ z in closedBall (0 : EuclideanSpace ℂ (Fin D)) r, w ‖z‖ ≤ ∫⁻ z in closedBall (0 : EuclideanSpace ℂ (Fin D)) r, ‖φ z‖ₑ ^ 2 * w ‖z‖) ∧ ‖φ 0‖ₑ ^ 2 ≤ (volume (closedBall (0 : EuclideanSpace ℂ (Fin D)) r))⁻¹ * ∫⁻ z in closedBall (0 : EuclideanSpace ℂ (Fin D)) r, ‖φ z‖ₑ ^ 2 ∧ volume (closedBall (0 : EuclideanSpace ℂ (Fin D)) r) = ENNReal.ofReal (π ^ D * r ^ (2 * D) / D.factorial)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EntireApplyEqOfPeriodicOfSpanReal

theorem **entire_apply_eq_of_values_on_compact** {F : E → V} (hF : Differentiable ℂ F) {K : Set E} (hK : IsCompact K) (hcov : ∀ x, ∃ k ∈ K, F x = F k) (z w : E) : F z = F w

theorem **entire_apply_eq_of_periodic_of_compact** {F : E → V} (hF : Differentiable ℂ F) {G K : Set E} (hK : IsCompact K) (hcov : ∀ x, ∃ k ∈ K, ∃ g ∈ G, x = k + g) (hper : ∀ x, ∀ g ∈ G, F (x + g) = F x) (z w : E) : F z = F w

theorem **entire_apply_eq_of_periodic_of_bounded** [FiniteDimensional ℂ E] {F : E → V} (hF : Differentiable ℂ F) {G K : Set E} (hK : Bornology.IsBounded K) (hcov : ∀ x, ∃ k ∈ K, ∃ g ∈ G, x = k + g) (hper : ∀ x, ∀ g ∈ G, F (x + g) = F x) (z w : E) : F z = F w

theorem **entire_const_of_addSubgroup_periodic** [FiniteDimensional ℂ E] {F : E → V} (hF : Differentiable ℂ F) (G : AddSubgroup E) {K : Set E} (hK : Bornology.IsBounded K) (hcov : ∀ x, ∃ g ∈ G, ∃ k ∈ K, x = g + k) (hper : ∀ x, ∀ g ∈ G, F (x + g) = F x) : ∃ c, ∀ z, F z = c

theorem **entire_apply_eq_of_isZLattice_periodic** [NormedSpace ℝ E] [FiniteDimensional ℝ E] {F : E → V} (hF : Differentiable ℂ F) (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] (hper : ∀ x, ∀ g ∈ L, F (x + g) = F x) (z w : E) : F z = F w

theorem **periodic_of_mem_span_int** {F : E → V} {S : Set E} (hper : ∀ x, ∀ g ∈ S, F (x + g) = F x) : ∀ x, ∀ g ∈ Submodule.span ℤ S, F (x + g) = F x

theorem **entire_apply_eq_of_periodic_of_span_real** [NormedSpace ℝ E] [FiniteDimensional ℝ E] {F : E → V} (hF : Differentiable ℂ F) {S : Set E} (hS : Submodule.span ℝ S = ⊤) (hper : ∀ x, ∀ g ∈ S, F (x + g) = F x) (z w : E) : F z = F w

theorem **iteratedDeriv_zero_eq_zero_of_poly_bound** {F : ℂ → V} (hF : Differentiable ℂ F) {C : ℝ} {m : ℕ} (hb : ∀ z, ‖F z‖ ≤ C * (1 + ‖z‖) ^ m) {n : ℕ} (hn : m < n) : iteratedDeriv n F 0 = 0

theorem **entire_eq_taylor_sum_of_poly_bound** {F : ℂ → V} (hF : Differentiable ℂ F) {C : ℝ} {m : ℕ} (hb : ∀ z, ‖F z‖ ≤ C * (1 + ‖z‖) ^ m) (z : ℂ) : F z = ∑ k ∈ Finset.range (m + 1), z ^ k • ((k ! : ℂ)⁻¹ • iteratedDeriv k F 0)

theorem **entire_exists_coeff_of_poly_bound** {F : ℂ → V} (hF : Differentiable ℂ F) {C : ℝ} {m : ℕ} (hb : ∀ z, ‖F z‖ ≤ C * (1 + ‖z‖) ^ m) : ∃ a : ℕ → V, ∀ z, F z = ∑ k ∈ Finset.range (m + 1), z ^ k • a k

theorem **entire_exists_polynomial_of_poly_bound** {F : ℂ → ℂ} (hF : Differentiable ℂ F) {C : ℝ} {m : ℕ} (hb : ∀ z, ‖F z‖ ≤ C * (1 + ‖z‖) ^ m) : ∃ p : Polynomial ℂ, p.natDegree ≤ m ∧ ∀ z, F z = p.eval z

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EqZeroOfEvalGridEqZeroOfTotalDegreeLe

theorem **eq_zero_of_eval_box_eq_zero** (d : σ → ℕ) (P : MvPolynomial σ R) (hdeg : ∀ i, P.degreeOf i ≤ d i) (hP : ∀ x : σ → ℕ, (∀ i, x i ≤ d i) → eval (fun i => (x i : R)) P = 0) : P = 0

theorem **eq_zero_of_eval_grid_eq_zero** (d : ℕ) (P : MvPolynomial σ R) (hdeg : ∀ i, P.degreeOf i ≤ d) (hP : ∀ x : σ → ℕ, (∀ i, x i ≤ d) → eval (fun i => (x i : R)) P = 0) : P = 0

theorem **mem_of_grid_mem** (d : ℕ) (Φ : (σ → A) → V) (hΦ : ∀ l : Module.Dual K V, ∃ P : MvPolynomial σ K, (∀ i, P.degreeOf i ≤ d) ∧ ∀ x, l (Φ x) = eval (fun i => algebraMap A K (x i)) P) (U : Submodule K V) (hU : ∀ x : σ → ℕ, (∀ i, x i ≤ d) → Φ (fun i => (x i : A)) ∈ U) (y : σ → A) : Φ y ∈ U

theorem **span_grid_eq_span_range** (d : ℕ) (Φ : (σ → A) → V) (hΦ : ∀ l : Module.Dual K V, ∃ P : MvPolynomial σ K, (∀ i, P.degreeOf i ≤ d) ∧ ∀ x, l (Φ x) = eval (fun i => algebraMap A K (x i)) P) : Submodule.span K ((fun x : σ → ℕ => Φ fun i => (x i : A)) '' {x | ∀ i, x i ≤ d}) = Submodule.span K (Set.range Φ)

theorem **exists_poly_dual_sum_monomial_smul** (d : ℕ) (s : Finset (σ →₀ ℕ)) (v : (σ →₀ ℕ) → V) (hs : ∀ a ∈ s, ∀ i, a i ≤ d) (l : Module.Dual K V) : ∃ P : MvPolynomial σ K, (∀ i, P.degreeOf i ≤ d) ∧ ∀ x : σ → A, l (∑ a ∈ s, (a.prod fun i n => algebraMap A K (x i) ^ n) • v a) = eval (fun i => algebraMap A K (x i)) P

theorem **span_grid_eq_span_range_of_eq_sum** (d : ℕ) (Φ : (σ → A) → V) (s : Finset (σ →₀ ℕ)) (v : (σ →₀ ℕ) → V) (hs : ∀ a ∈ s, ∀ i, a i ≤ d) (hΦ : ∀ x, Φ x = ∑ a ∈ s, (a.prod fun i n => algebraMap A K (x i) ^ n) • v a) : Submodule.span K ((fun x : σ → ℕ => Φ fun i => (x i : A)) '' {x | ∀ i, x i ≤ d}) = Submodule.span K (Set.range Φ)

theorem **span_grid_eval_eq_span_range** {B : Type*} [CommRing B] [Algebra K B] (d : ℕ) (F : MvPolynomial σ B) (hF : ∀ i, F.degreeOf i ≤ d) : Submodule.span K ((fun x : σ → ℕ => eval (fun i => (x i : B)) F) '' {x | ∀ i, x i ≤ d}) = Submodule.span K (Set.range fun x : σ → A => eval (fun i => algebraMap K B (algebraMap A K (x i))) F)

theorem **exists_nsmul_add_apply_ne_zero** [∀ j, IsAddTorsionFree (W j)] (l : ∀ j, V →+ W j) (v w : V) : ∃ c : ℕ, c ≤ Fintype.card ι ∧ ∀ j, l j v ≠ 0 ∨ l j w ≠ 0 → l j (c • w + v) ≠ 0

theorem **exists_sum_nsmul_apply_ne_zero_finset** [∀ j, IsAddTorsionFree (W j)] (l : ∀ j, V →+ W j) (s : κ → V) (t : Finset κ) : ∃ c : κ → ℕ, (∀ k, c k ≤ Fintype.card ι) ∧ ∀ j, (∃ k ∈ t, l j (s k) ≠ 0) → l j (∑ k ∈ t, c k • s k) ≠ 0

theorem **exists_sum_nsmul_apply_ne_zero** [∀ j, IsAddTorsionFree (W j)] [Fintype κ] (l : ∀ j, V →+ W j) (s : κ → V) (h : ∀ j, ∃ k, l j (s k) ≠ 0) : ∃ c : κ → ℕ, (∀ k, c k ≤ Fintype.card ι) ∧ ∀ j, l j (∑ k, c k • s k) ≠ 0

theorem **exists_sum_natCast_smul_apply_ne_zero** [Fintype κ] (l : ∀ j, V →ₗ[K] W j) (s : κ → V) (h : ∀ j, ∃ k, l j (s k) ≠ 0) : ∃ c : κ → ℕ, (∀ k, c k ≤ Fintype.card ι) ∧ ∀ j, l j (∑ k, (c k : K) • s k) ≠ 0

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EqZeroOfSemicharacterMultiplier

theorem **mem_thetaLevel** : f ∈ thetaLevel N τ ↔ Differentiable ℂ f ∧ ∀ (n n' : ι → ℤ) (z : ι → ℂ), f (z + τ *ᵥ ofIntVec n + ofIntVec n') = eC (-(N : ℂ) * (ofIntVec n ⬝ᵥ (τ *ᵥ ofIntVec n) / 2 + ofIntVec n ⬝ᵥ z)) * f z

theorem **mem_thetaLevelChar** : f ∈ thetaLevelChar a b N τ ↔ Differentiable ℂ f ∧ ∀ (n n' : ι → ℤ) (z : ι → ℂ), f (z + τ *ᵥ ofIntVec n + ofIntVec n') = eC ((N : ℂ) * (ofRealVec a ⬝ᵥ ofIntVec n' - ofRealVec b ⬝ᵥ ofIntVec n)) * eC (-(N : ℂ) * (ofIntVec n ⬝ᵥ (τ *ᵥ ofIntVec n) / 2 + ofIntVec n ⬝ᵥ z)) * f z

theorem **eC_ne_zero** (x : ℂ) : eC x ≠ 0

theorem **eC_nat_mul** (N : ℕ) (x : ℂ) : eC (N * x) = eC x ^ N

theorem **ofRealVec_zero** : ofRealVec (0 : ι → ℝ) = 0

theorem **mem_thetaLevelChar_iff_autFactor** : f ∈ thetaLevelChar a b N τ ↔ Differentiable ℂ f ∧ ∀ (n n' : ι → ℤ) (z : ι → ℂ), f (z + τ *ᵥ ofIntVec n + ofIntVec n') = autFactor a b τ n n' z ^ N * f z

@[simp] theorem **thetaLevelChar_zero** (N : ℕ) (τ : Matrix ι ι ℂ) : thetaLevelChar 0 0 N τ = thetaLevel N τ

theorem **one_mem_thetaLevelChar** : (1 : (ι → ℂ) → ℂ) ∈ thetaLevelChar a b 0 τ

theorem **mul_mem_thetaLevelChar** (hf : f ∈ thetaLevelChar a b N τ) (hg : g ∈ thetaLevelChar a b M τ) : f * g ∈ thetaLevelChar a b (N + M) τ

theorem **prod_mem_thetaLevelChar** {κ : Type*} (s : Finset κ) {d : κ → ℕ} {F : κ → (ι → ℂ) → ℂ} (hF : ∀ i ∈ s, F i ∈ thetaLevelChar a b (d i) τ) : ∏ i ∈ s, F i ∈ thetaLevelChar a b (∑ i ∈ s, d i) τ

theorem **pow_mem_thetaLevelChar** (hf : f ∈ thetaLevelChar a b N τ) (k : ℕ) : f ^ k ∈ thetaLevelChar a b (k * N) τ

theorem **prod_translate_mem_thetaLevelChar** {κ : Type*} (s : Finset κ) {d : κ → ℕ} {F : κ → (ι → ℂ) → ℂ} (hF : ∀ i ∈ s, F i ∈ thetaLevelChar a b (d i) τ) (u : κ → ι → ℂ) (hu : ∑ i ∈ s, (d i : ℂ) • u i = 0) : (fun z => ∏ i ∈ s, F i (z + u i)) ∈ thetaLevelChar a b (∑ i ∈ s, d i) τ

private theorem **coeffMap_injective** (hN : 0 < N) (τ : Matrix ι ι ℂ) : Function.Injective (coeffMap N τ)

theorem **finiteDimensional_thetaLevel** (hN : 0 < N) (τ : Matrix ι ι ℂ) : FiniteDimensional ℂ (thetaLevel N τ)

theorem **finrank_thetaLevel_le** (hN : 0 < N) (τ : Matrix ι ι ℂ) : Module.finrank ℂ (thetaLevel N τ) ≤ N ^ Fintype.card ι

theorem **thetaShift_apply** (f : (ι → ℂ) → ℂ) (z : ι → ℂ) : thetaShift a b N τ f z = eC (N * (ofRealVec a ⬝ᵥ (τ *ᵥ ofRealVec a) / 2 + ofRealVec a ⬝ᵥ (z + ofRealVec b))) * f (z + τ *ᵥ ofRealVec a + ofRealVec b)

theorem **thetaShift_symm_apply** (g : (ι → ℂ) → ℂ) (w : ι → ℂ) : (thetaShift a b N τ).symm g w = eC (-(N * (ofRealVec a ⬝ᵥ (τ *ᵥ ofRealVec a) / 2 + ofRealVec a ⬝ᵥ (w - τ *ᵥ ofRealVec a)))) * g (w - τ *ᵥ ofRealVec a - ofRealVec b)

theorem **differentiable_thetaShift_iff** : Differentiable ℂ (thetaShift a b N τ f) ↔ Differentiable ℂ f

theorem **thetaShift_mem_iff** (hτ : τ.IsSymm) (a' b' : ι → ℝ) : thetaShift a b N τ f ∈ thetaLevelChar (a' + a) (b' + b) N τ ↔ f ∈ thetaLevelChar a' b' N τ

theorem **map_thetaShift_thetaLevel** (a b : ι → ℝ) (N : ℕ) (hτ : τ.IsSymm) : (thetaLevel N τ).map (thetaShift a b N τ).toLinearMap = thetaLevelChar a b N τ

@[simp] theorem **thetaLevelEquivChar_apply** (hτ : τ.IsSymm) (f : thetaLevel N τ) : (thetaLevelEquivChar a b N hτ f : (ι → ℂ) → ℂ) = thetaShift a b N τ f

theorem **levelTheta_zero_zero** (c : ι → ℤ) (z : ι → ℂ) : levelTheta 0 0 N τ c z = theta (fun i => (c i : ℝ) / N) 0 ((N : ℂ) • z) ((N : ℂ) • τ)

theorem **ofRealVec_natCast_smul** (N : ℕ) (b : ι → ℝ) : ofRealVec ((N : ℝ) • b) = (N : ℂ) • ofRealVec b

theorem **ofIntVec_natCast_smul** (N : ℕ) (n : ι → ℤ) : ofIntVec ((N : ℤ) • n) = (N : ℂ) • ofIntVec n

theorem **thetaShift_levelTheta** (hτ : τ.IsSymm) (c : ι → ℤ) : thetaShift a b N τ (levelTheta 0 0 N τ c) = levelTheta a b N τ c

theorem **smul_mem_siegelH** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) : (N : ℂ) • τ ∈ SiegelH ι

theorem **differentiable_theta** (hτ : τ ∈ SiegelH ι) (a b : ι → ℝ) : Differentiable ℂ fun z => theta a b z τ

theorem **theta_ne_zero** (hτ : τ ∈ SiegelH ι) (a b : ι → ℝ) : (fun z => theta a b z τ) ≠ 0

theorem **levelTheta_ne_zero** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) (c : ι → ℤ) : levelTheta a b N τ c ≠ 0

theorem **levelTheta_mem** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) (c : ι → ℤ) : levelTheta a b N τ c ∈ thetaLevelChar a b N τ

theorem **levelTheta_zero_zero_mem** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) (c : ι → ℤ) : levelTheta 0 0 N τ c ∈ thetaLevel N τ

theorem **levelTheta_add_div** (hτ : τ.IsSymm) (hN : 0 < N) (c m : ι → ℤ) (z : ι → ℂ) : levelTheta 0 0 N τ c (z + (N : ℂ)⁻¹ • ofIntVec m) = eC (((c ⬝ᵥ m : ℤ) : ℂ) / N) * levelTheta 0 0 N τ c z

theorem **intCast_zmod_eq_of_eC_eq** (hN : 0 < N) {p q : ℤ} (h : eC (p / N) = eC (q / N)) : (p : ZMod N) = q

theorem **linearIndependent_levelTheta** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) {κ : Type*} {r : κ → ι → ℤ} (hr : Function.Injective fun k i => (r k i : ZMod N)) : LinearIndependent ℂ fun k => levelTheta a b N τ (r k)

theorem **finiteDimensional_thetaLevelChar** (hτ : τ.IsSymm) (hN : 0 < N) : FiniteDimensional ℂ (thetaLevelChar a b N τ)

theorem **finrank_thetaLevelChar** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) : Module.finrank ℂ (thetaLevelChar a b N τ) = N ^ Fintype.card ι

theorem **finrank_thetaLevel** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) : Module.finrank ℂ (thetaLevel N τ) = N ^ Fintype.card ι

theorem **span_levelTheta** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) {κ : Type*} {r : κ → ι → ℤ} (hr : Function.Bijective fun k i => (r k i : ZMod N)) : Submodule.span ℂ (Set.range fun k => levelTheta a b N τ (r k)) = thetaLevelChar a b N τ

@[simp] theorem **thetaLevelCharBasis_apply** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) {κ : Type*} {r : κ → ι → ℤ} (hr : Function.Bijective fun k i => (r k i : ZMod N)) (k : κ) : (thetaLevelCharBasis a b hτ hN hr k : (ι → ℂ) → ℂ) = levelTheta a b N τ (r k)

@[simp] theorem **thetaLevelBasis_apply** (hτ : τ ∈ SiegelH ι) (hN : 0 < N) {κ : Type*} {r : κ → ι → ℤ} (hr : Function.Bijective fun k i => (r k i : ZMod N)) (k : κ) : (thetaLevelBasis hτ hN hr k : (ι → ℂ) → ℂ) = fun z => theta (fun i => (r k i : ℝ) / N) 0 ((N : ℂ) • z) ((N : ℂ) • τ)

theorem **bijective_fin_zmod** (hN : 0 < N) : Function.Bijective fun (c : ι → Fin N) i => ((((c i : ℕ) : ℤ)) : ZMod N)

theorem **levelTheta_one** : levelTheta a b 1 τ 0 = fun z => theta a b z τ

theorem **theta_mem_thetaLevelChar_one** (hτ : τ ∈ SiegelH ι) : (fun z => theta a b z τ) ∈ thetaLevelChar a b 1 τ

theorem **thetaLevelChar_one** (hτ : τ ∈ SiegelH ι) : thetaLevelChar a b 1 τ = Submodule.span ℂ {fun z => theta a b z τ}

theorem **multiplier_add_of_ne_zero** (hτ : τ.IsSymm) {χ : (ι → ℤ) → (ι → ℤ) → ℂ} (hf : f ≠ 0) (h : ∀ (n n' : ι → ℤ) (z : ι → ℂ), f (z + τ *ᵥ ofIntVec n + ofIntVec n') = χ n n' * eC (-(N : ℂ) * (ofIntVec n ⬝ᵥ (τ *ᵥ ofIntVec n) / 2 + ofIntVec n ⬝ᵥ z)) * f z) (n n' m m' : ι → ℤ) : χ (n + m) (n' + m') = χ n n' * χ m m'

theorem **exists_half_char_of_sign** {α : (ι → ℤ) → (ι → ℤ) → ℂ} (hα : ∀ n n' m m', α (n + m) (n' + m') = α n n' * α m m') (hs : ∀ n n', α n n' = 1 ∨ α n n' = -1) : ∃ p q : ι → ℤ, ∀ n n', α n n' = eC (ofRealVec (fun i => (p i : ℝ) / 2) ⬝ᵥ ofIntVec n' - ofRealVec (fun i => (q i : ℝ) / 2) ⬝ᵥ ofIntVec n)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EventuallyExistsMulIsLowerTriangular

theorem **pderiv_algHom_of_linear** {R : Type*} [CommRing R] (G : MvPolynomial τ R →ₐ[R] MvPolynomial σ R) (M : σ → τ → R) (hG : ∀ j, G (X j) = ∑ i, M i j • X i) (i : σ) (Q : MvPolynomial τ R) : pderiv i (G Q) = ∑ j, M i j • G (pderiv j Q)

theorem **constantCoeff_algHom_of_linear** {R : Type*} [CommRing R] (G : MvPolynomial τ R →ₐ[R] MvPolynomial σ R) (M : σ → τ → R) (hG : ∀ j, G (X j) = ∑ i, M i j • X i) (Q : MvPolynomial τ R) : constantCoeff (G Q) = constantCoeff Q

theorem **fockInner_algHom_of_linear** (F : MvPolynomial σ ℂ →ₐ[ℂ] MvPolynomial τ ℂ) (G : MvPolynomial τ ℂ →ₐ[ℂ] MvPolynomial σ ℂ) (M : σ → τ → ℂ) (hF : ∀ i, F (X i) = ∑ j, M i j • X j) (hG : ∀ j, G (X j) = ∑ i, star (M i j) • X i) (P : MvPolynomial σ ℂ) (Q : MvPolynomial τ ℂ) : fockInner (F P) Q = fockInner P (G Q)

theorem **fockInner_colSubst** (A B : T → Matrix κ κ ℂ) (P Q : Poly ℂ T p q κ) : fockInner (colSubst A B P) Q = fockInner P (colSubst (fun t => (A t)ᴴ) (fun t => (B t)ᴴ) Q)

theorem **fockInner_colAct** [DecidableEq κ] (g : T → GL κ ℂ) (P Q : Poly ℂ T p q κ) : fockInner (colAct ℂ T p q κ g P) Q = fockInner P (colAct ℂ T p q κ (star g) Q)

theorem **exists_fockInner_sub_eq_zero** {σ : Type*} (W : Submodule ℂ (MvPolynomial σ ℂ)) [FiniteDimensional ℂ W] (P : MvPolynomial σ ℂ) : ∃ Q ∈ W, ∀ w ∈ W, fockInner (P - Q) w = 0

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.EventuallyIteratedFDerivEqZeroOfRealSubspace

theorem **fderiv_apply_eq_zero_of_frequently** {f : E → V} {x s : E} (hf : DifferentiableAt ℂ f x) (h : ∃ᶠ t in 𝓝[≠] (0 : ℂ), f (x + t • s) = f x) : fderiv ℂ f x s = 0

theorem **frequently_nhdsNE_of_eventually_ofReal** {p : ℂ → Prop} (h : ∀ᶠ t : ℝ in 𝓝 0, p t) : ∃ᶠ t in 𝓝[≠] (0 : ℂ), p t

theorem **fderiv_eq_zero_of_real_lines** {f : E → V} {x : E} {S : Set E} (hS : Submodule.span ℂ S = ⊤) (hf : DifferentiableAt ℂ f x) (h : ∀ s ∈ S, ∀ᶠ t : ℝ in 𝓝 0, f (x + (t : ℂ) • s) = f x) : fderiv ℂ f x = 0

theorem **eq_of_eventually_real_lines** {f : E → V} {U S : Set E} (hS : Submodule.span ℂ S = ⊤) (hU : IsOpen U) (hU' : IsPreconnected U) (hf : DifferentiableOn ℂ f U) (h : ∀ z ∈ U, ∀ s ∈ S, ∀ᶠ t : ℝ in 𝓝 0, f (z + (t : ℂ) • s) = f z) {z w : E} (hz : z ∈ U) (hw : w ∈ U) : f z = f w

theorem **eq_of_invariant_real_lines** {f : E → V} {U S : Set E} (hS : Submodule.span ℂ S = ⊤) (hU : IsOpen U) (hU' : IsPreconnected U) (hf : DifferentiableOn ℂ f U) (h : ∀ z ∈ U, ∀ s ∈ S, ∀ t : ℝ, z + (t : ℂ) • s ∈ U → f (z + (t : ℂ) • s) = f z) {z w : E} (hz : z ∈ U) (hw : w ∈ U) : f z = f w

theorem **eq_of_invariant_realSubspace** {f : E → V} {U : Set E} {S : Submodule ℝ E} (hS : Submodule.span ℂ (S : Set E) = ⊤) (hU : IsOpen U) (hU' : IsPreconnected U) (hf : DifferentiableOn ℂ f U) (h : ∀ z ∈ U, ∀ s ∈ S, z + s ∈ U → f (z + s) = f z) {z w : E} (hz : z ∈ U) (hw : w ∈ U) : f z = f w

theorem **span_scaledRealPoints** {a : ℂ} (ha : a ≠ 0) : Submodule.span ℂ (scaledRealPoints ι a : Set (ι → ℂ)) = ⊤

theorem **eq_of_invariant_real** {f : (ι → ℂ) → V} {U : Set (ι → ℂ)} (hU : IsOpen U) (hU' : IsPreconnected U) (hf : DifferentiableOn ℂ f U) (h : ∀ z ∈ U, ∀ x : ι → ℝ, (z + fun i => (x i : ℂ)) ∈ U → f (z + fun i => (x i : ℂ)) = f z) {z w : ι → ℂ} (hz : z ∈ U) (hw : w ∈ U) : f z = f w

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsCompactHaarMeasure

theorem **exists_isKMeas** : ∀ k r : ℕ, ∃ μK : Measure (Mt k r), IsKMeas k r μK

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsConeMeasure

theorem **continuous_sqLen** (n : ℕ) : Continuous (sqLen : (Fin n → ℂ) → ℝ)

theorem **sqLen_nonneg** {n : ℕ} (w : Fin n → ℂ) : 0 ≤ sqLen w

theorem **continuous_param** (n : ℕ) : Continuous (param n)

theorem **normSq_param_last** (n : ℕ) (p : (Fin n → ℂ) × ℝ) : Complex.normSq (param n p (Fin.last n)) = sqLen p.1

theorem **sqLen_param** (n : ℕ) (p : (Fin n → ℂ) × ℝ) : sqLen (param n p) = 2 * sqLen p.1

theorem **herm_param** (n : ℕ) (p : (Fin n → ℂ) × ℝ) : herm (Matrix.diagonal (fun i : Fin (n + 1) => if i = Fin.last n then (-1 : ℂ) else 1)) (param n p) (param n p) = 0

theorem **lintegral_cone** (n : ℕ) (F : ℝ → ℝ≥0∞) (hF : Measurable F) : ∫⁻ v, F (sqLen v) ∂coneMeasure n = (∫⁻ w : Fin n → ℂ, F (2 * sqLen w)) * ENNReal.ofReal (2 * Real.pi)

theorem **lintegral_scale** (n : ℕ) (s : ℝ) (hs : 0 < s) (F : ℝ → ℝ≥0∞) (hF : Measurable F) : ∫⁻ w : Fin n → ℂ, F (s * sqLen w) = ENNReal.ofReal (s ^ (-(n : ℝ))) * ∫⁻ w : Fin n → ℂ, F (sqLen w)

theorem **gauss_ne_top** (n : ℕ) : ∫⁻ w : Fin n → ℂ, ENNReal.ofReal (Real.exp (-(2 * sqLen w))) ≠ ∞

theorem **gauss_ne_zero** (n : ℕ) : ∫⁻ w : Fin n → ℂ, ENNReal.ofReal (Real.exp (-(2 * sqLen w))) ≠ 0

theorem **ae_cone** (n : ℕ) (hn : 1 ≤ n) : ∀ᵐ v ∂coneMeasure n, v ≠ 0 ∧ herm (Matrix.diagonal (fun i : Fin (n + 1) => if i = Fin.last n then (-1 : ℂ) else 1)) v v = 0

theorem **Hpk_zero** (k : ℕ) : Hpk k 0 = Matrix.diagonal (fun i : Fin (2 * 0 + k + 1) => if i = Fin.last (2 * 0 + k) then (-1 : ℂ) else 1)

theorem **exists_coneMeasure** : ∀ k : ℕ, 1 ≤ k → ∃ m : Measure (Fin (2 * 0 + k + 1) → ℂ), SFinite m ∧ (∀ᵐ v ∂m, v ≠ 0 ∧ herm (Hpk k 0) v v = 0) ∧ (∀ s : ℝ, 0 < s → ∀ f : ℝ → ℝ≥0∞, Measurable f → (∫⁻ v, f (s * sqLen v) ∂m) = ENNReal.ofReal (s ^ (-((2 * 0 + k : ℕ) : ℝ))) * ∫⁻ v, f (sqLen v) ∂m) ∧ (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) ≠ 0 ∧ (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) ≠ ∞ ∧ ∀ g ∈ GV k 0, (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen (g *ᵥ v))) ∂m) = ∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsConeMeasurePosRank

lemma **radMeas_map** (N : ℕ) (t : ℝ) (ht : 0 < t) : (radMeas N).map (fun c => (t : ℂ) * c) = ENNReal.ofReal ((t ^ (2 * N + 2))⁻¹) • radMeas N

lemma **par_meas** (k r' : ℕ) : Measurable (par k r')

lemma **cone_def** (k r' : ℕ) : coneMeas k r' = ((radMeas (2 * r' + k + 1)).prod (volume : Measure (HPar k r'))).map (par k r')

lemma **cone_sfinite** (k r' : ℕ) : SFinite (coneMeas k r')

lemma **cone_map** (k r' : ℕ) (t : ℝ) (ht : 0 < t) : (coneMeas k r').map (fun v => (t : ℂ) • v) = ENNReal.ofReal ((t ^ (2 * (2 * r' + k + 1) + 2))⁻¹) • coneMeas k r'

lemma **sqLen_cont** {n : ℕ} : Continuous (sqLen : (Fin n → ℂ) → ℝ)

lemma **sqLen_nonneg** {n : ℕ} (v : Fin n → ℂ) : 0 ≤ sqLen v

lemma **sqLen_smul** {n : ℕ} (t : ℝ) (v : Fin n → ℂ) : sqLen ((t : ℂ) • v) = t ^ 2 * sqLen v

lemma **cone_homog** (k r' : ℕ) (s : ℝ) (hs : 0 < s) (f : ℝ → ℝ≥0∞) (hf : Measurable f) : (∫⁻ v, f (s * sqLen v) ∂(coneMeas k r')) = ENNReal.ofReal (s ^ (-((2 * (1 + r') + k : ℕ) : ℝ))) * ∫⁻ v, f (sqLen v) ∂(coneMeas k r')

lemma **par_eq** (k r' : ℕ) (p : ℂ × HPar k r') : par k r' p = p.1 • mk3 k r' (hb k r' p.2.1 p.2.2) p.2.1 1

lemma **cone_ae** (k r' : ℕ) : ∀ᵐ v ∂(coneMeas k r'), v ≠ 0 ∧ herm (Hpk k (1 + r')) v v = 0

lemma **cone_ne_zero** (k r' : ℕ) : coneMeas k r' ≠ 0

lemma **ball_compact** {n : ℕ} : IsCompact {v : Fin n → ℂ | sqLen v ≤ 1}

lemma **gauss_finite** {X : Type*} [MeasurableSpace X] (m : Measure X) (q : X → ℝ) (hq : Measurable q) (h0 : ∀ v, 0 ≤ q v) (d : ℕ) (hh : ∀ s : ℝ, 0 < s → ∀ f : ℝ → ℝ≥0∞, Measurable f → (∫⁻ v, f (s * q v) ∂m) = ENNReal.ofReal (s ^ (-(d : ℝ))) * ∫⁻ v, f (q v) ∂m) (hfin : m {v | q v ≤ 1} < ∞) : (∫⁻ v, ENNReal.ofReal (Real.exp (-q v)) ∂m) ≠ ∞

theorem **exists_cone_measure_pos_rank** : ∀ k r' : ℕ, ∃ m : Measure (Fin (2 * (1 + r') + k + 1) → ℂ), SFinite m ∧ (∀ᵐ v ∂m, v ≠ 0 ∧ herm (Hpk k (1 + r')) v v = 0) ∧ (∀ s : ℝ, 0 < s → ∀ f : ℝ → ℝ≥0∞, Measurable f → (∫⁻ v, f (s * sqLen v) ∂m) = ENNReal.ofReal (s ^ (-((2 * (1 + r') + k : ℕ) : ℝ))) * ∫⁻ v, f (sqLen v) ∂m) ∧ (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) ≠ 0 ∧ (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) ≠ ∞ ∧ ∀ g ∈ GV k (1 + r'), (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen (g *ᵥ v))) ∂m) = ∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsIdealInClassOfNormLeSqrtDiscr

theorem **sum_zeta_pow_succ_le** (k : ℕ) {X : ℝ} (hX : 1 ≤ X) (N : ℕ) (hN : (N : ℝ) ≤ X) : ((∑ n ∈ Ioc 0 N, (ζ ^ (k + 1) : ArithmeticFunction ℕ) n : ℕ) : ℝ) ≤ N * (1 + Real.log X) ^ k

theorem **sum_zeta_pow_le** {k : ℕ} (hk : 1 ≤ k) {X : ℝ} (hX : 1 ≤ X) : ((∑ n ∈ Ioc 0 ⌊X⌋₊, (ζ ^ k : ArithmeticFunction ℕ) n : ℕ) : ℝ) ≤ X * (1 + Real.log X) ^ (k - 1)

theorem **card_finMulAntidiag_eq_zeta_pow** (k n : ℕ) : (Nat.finMulAntidiag k n).card = (ζ ^ k : ArithmeticFunction ℕ) n

theorem **exists_colouring_maximal_ideals** : ∃ c : Ideal (𝒪 K) → Fin (finrank ℚ K), ∀ p : ℕ, p.Prime → ∀ P Q : Ideal (𝒪 K), P.IsMaximal → Q.IsMaximal → (p : 𝒪 K) ∈ P → (p : 𝒪 K) ∈ Q → c P = c Q → P = Q

theorem **ncard_ideal_absNorm_eq_le** {n : ℕ} (hn : n ≠ 0) : {I : Ideal (𝒪 K) | Ideal.absNorm I = n}.ncard ≤ (ζ ^ finrank ℚ K : ArithmeticFunction ℕ) n

theorem **two_pow_mul_factorial_succ_le** (n : ℕ) : (2 : ℝ) ^ n * ((n + 1).factorial : ℝ) ≤ ((n : ℝ) + 1) ^ (n + 1)

theorem **minkowski_const_le_one** : (4 / π) ^ InfinitePlace.nrComplexPlaces K * (((finrank ℚ K).factorial : ℝ) / (finrank ℚ K : ℝ) ^ finrank ℚ K) ≤ 1

theorem **exists_ideal_in_class_of_norm_le_sqrt_discr** (C : ClassGroup (𝒪 K)) : ∃ I : (Ideal (𝒪 K))⁰, ClassGroup.mk0 I = C ∧ (Ideal.absNorm (I : Ideal (𝒪 K)) : ℝ) ≤ √|(discr K : ℝ)|

theorem **one_le_sqrt_abs_discr** : 1 ≤ √|(discr K : ℝ)|

theorem **classNumber_le_sqrt_discr_mul_pow_pred** : (classNumber K : ℝ) ≤ √|(discr K : ℝ)| * (1 + Real.log √|(discr K : ℝ)|) ^ (finrank ℚ K - 1)

theorem **classNumber_le_sqrt_discr_mul_pow** : (classNumber K : ℝ) ≤ √|(discr K : ℝ)| * (1 + Real.log √|(discr K : ℝ)|) ^ finrank ℚ K

theorem **classNumber_le_of_finrank_eq_two_mul** {f : ℕ} (hf : finrank ℚ K = 2 * f) : (classNumber K : ℝ) ≤ √|(discr K : ℝ)| * (1 + Real.log √|(discr K : ℝ)|) ^ (2 * f)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsIsClosedEmbeddingOfAffine

theorem **PtsOver.t2Space** [T1Space C] (p : X ⟶ S) [IsSeparated p] (s : Sp C ⟶ S) : T2Space (PtsOver p s)

@[simp] theorem **PtsOver.baseChangeHomeomorph_val** {P : Scheme.{0}} {g : P ⟶ X} {q : P ⟶ Sp C} {p : X ⟶ S} {s : Sp C ⟶ S} (hP : IsPullback g q p s) (z : Pts q) : (PtsOver.baseChangeHomeomorph hP z).1 = z.1 ≫ g

theorem **exists_isClosedEmbedding_of_affine** [T1Space C] {A : CommRingCat.{0}} (q : Spec A ⟶ Sp C) [LocallyOfFiniteType q] : ∃ (n : ℕ) (c : Pts q → (Fin n → C)), IsClosedEmbedding c

theorem **mem_range_map_fromSpec** (p : X ⟶ S) (s : Sp C ⟶ S) {U : X.Opens} (hU : IsAffineOpen U) (x : PtsOver p s) (hx : x.1 (IsLocalRing.closedPoint C) ∈ U) : x ∈ Set.range (PtsOver.map (p

theorem **Pts.locallyCompactSpace** [T1Space C] [LocallyCompactSpace C] (p : X ⟶ Sp C) [LocallyOfFiniteType p] : LocallyCompactSpace (Pts p)

theorem **PtsOver.locallyCompactSpace** [T1Space C] [LocallyCompactSpace C] (p : X ⟶ S) [LocallyOfFiniteType p] (s : Sp C ⟶ S) : LocallyCompactSpace (PtsOver p s)

theorem **Pts.secondCountableTopology_of_cover** [T1Space C] [SecondCountableTopology C] (p : X ⟶ Sp C) [LocallyOfFiniteType p] {ι : Type*} [Countable ι] {U : ι → X.Opens} (hU : ∀ i, IsAffineOpen (U i)) (hc : iSup U = ⊤) : SecondCountableTopology (Pts p)

theorem **PtsOver.secondCountableTopology** [T1Space C] [SecondCountableTopology C] (p : X ⟶ S) [LocallyOfFiniteType p] [QuasiCompact p] (s : Sp C ⟶ S) : SecondCountableTopology (PtsOver p s)

theorem **continuousMul_over** (A : Over (Sp C)) [MonObj A] : ContinuousMul (𝟙_ (Over (Sp C)) ⟶ A) where

theorem **isTopologicalGroup_over** (A : Over (Sp C)) [GrpObj A] : IsTopologicalGroup (𝟙_ (Over (Sp C)) ⟶ A) where

theorem **le_pt_preimage_chart_iff** (c : R →+* C) (v : Fin (N + 1) → C) (hv : Ideal.span (Set.range v) = ⊤) (i : Fin (N + 1)) : ⊤ ≤ pt c v hv ⁻¹ᵘ chart R N i ↔ IsUnit (v i)

theorem **valR_pt_ratio** (c : R →+* C) (v : Fin (N + 1) → C) (hv : Ideal.span (Set.range v) = ⊤) {i : Fin (N + 1)} (hi : IsUnit (v i)) (j : Fin (N + 1)) : valR (pt c v hv) (chart R N i) (ratio R N i j) = v j * Ring.inverse (v i)

theorem **projPt_surjective** (c : R →+* C) : Function.Surjective (projPt (N

theorem **projPt_eq_iff** (c : R →+* C) (v w : Unimodular C N) : projPt c v = projPt c w ↔ ∃ u : Cˣ, w.1 = u • v.1

theorem **continuous_projPt** (c : R →+* C) : Continuous (projPt (N

theorem **continuous_supSpherePt** (c : R →+* F) : Continuous (supSpherePt c N)

theorem **supSpherePt_surjective** (c : R →+* F) : Function.Surjective (supSpherePt c N)

theorem **compactSpace_pSpace** [ProperSpace F] (c : R →+* F) (N : ℕ) : CompactSpace (PtsOver (PSpace.str R N) (spMap c))

theorem **PtsOver.compactSpace** [ProperSpace F] {p : X ⟶ Sp R} (hp : IsProjOver p) (s : Sp F ⟶ Sp R) : CompactSpace (PtsOver p s)

theorem **span_range_sphere** (v : Metric.sphere (0 : EuclideanSpace 𝕜 (Fin (N + 1))) 1) : Ideal.span (Set.range v.1.ofLp) = ⊤

theorem **continuous_spherePt** (c : R →+* 𝕜) : Continuous (spherePt c N)

theorem **spherePt_surjective** (c : R →+* 𝕜) : Function.Surjective (spherePt c N)

theorem **complex_points_topology** : (∀ (X : Scheme.{0}) (p : X ⟶ Sp ℂ), (IsSeparated p → T2Space (Pts p)) ∧ (LocallyOfFiniteType p → LocallyCompactSpace (Pts p)) ∧ (LocallyOfFiniteType p → QuasiCompact p → SecondCountableTopology (Pts p)) ∧ (IsProjOver p → CompactSpace (Pts p))) ∧ (∀ N : ℕ, ∃ π : Metric.sphere (0 : EuclideanSpace ℂ (Fin (N + 1))) 1 → Pts (PSpace.str ℂ N), (∀ v, (π v).1 = pt (RingHom.id ℂ) v.1.ofLp (span_range_sphere v)) ∧ Continuous π ∧ Function.Surjective π) ∧ ∀ (A : Over (Sp ℂ)) [GrpObj A], IsTopologicalGroup (𝟙_ (Over (Sp ℂ)) ⟶ A)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsIsFundamentalDomainAndQuotMeasureEq

theorem **isAEOrbitSection_sign_Ici** : IsAEOrbitSection ℤˣ (Ici (0 : ℝ)) volume where

theorem **isAEAddOrbitSection_zmultiples_Ico** : IsAEAddOrbitSection (AddSubgroup.zmultiples (1 : ℝ)) (Ico (0 : ℝ) 1) volume where

theorem **exists_measurableSet_orbitSection_of_discreteOrbits** [Countable G] [TopologicalSpace X] [SecondCountableTopology X] [MeasurableSpace X] [OpensMeasurableSpace X] [MeasurableConstSMul G X] (hfix : ∀ g : G, MeasurableSet {x : X | g • x = x}) (hdisc : ∀ x : X, ∃ U ∈ 𝒩 x, ∀ g : G, g • x ∈ U → g • x = x) : ∃ D : Set X, MeasurableSet D ∧ (∀ x, ∃ g : G, g • x ∈ D) ∧ ∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x

theorem **measure_le_of_ae_unique_of_ae_covers** [Countable G] [MeasurableConstSMul G X] [SMulInvariantMeasure G X μ] {S B : Set X} (hS : NullMeasurableSet S μ) (hB : NullMeasurableSet B μ) (hu : ∀ᵐ x ∂μ, x ∈ S → ∀ g : G, g • x ∈ S → g • x = x) (hc : ∀ᵐ x ∂μ, x ∈ S → ∃ g : G, g • x ∈ B) : μ S ≤ μ B

theorem **map_restrict_le_quotMeasure** {D : Set X} (hD : NullMeasurableSet D μ) (hu : ∀ᵐ x ∂μ, x ∈ D → ∀ g : G, g • x ∈ D → g • x = x) : (μ.restrict D).map (Quotient.mk (MulAction.orbitRel G X)) ≤ quotMeasure G μ

theorem **isAEOrbitSection_of_forall** {D : Set X} (hD : MeasurableSet D) (hc : ∀ x, ∃ g : G, g • x ∈ D) (hu : ∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x) (μ : Measure X) : IsAEOrbitSection G D μ

theorem **IsAEOrbitSection.of_sameOrbits** {H : Type*} [SMul H X] {D : Set X} (hD : IsAEOrbitSection H D μ) (h₁ : ∀ (h : H) (x : X), ∃ g : G, g • x = h • x) (h₂ : ∀ (g : G) (x : X), ∃ h : H, h • x = g • x) : IsAEOrbitSection G D μ where

theorem **isAEOrbitSection_of_isFundamentalDomain** [Countable G] {D : Set X} (hD : IsFundamentalDomain G D μ) : IsAEOrbitSection G D μ where

theorem **isAEOrbitSection_of_isFundamentalDomain_of_surjective** [Countable G] {Q : Type*} [Group Q] [MulAction Q X] (φ : G →* Q) (hφ : Function.Surjective φ) (hact : ∀ g (x : X), φ g • x = g • x) {D : Set X} (hD : IsFundamentalDomain Q D μ) : IsAEOrbitSection G D μ

theorem **isFundamentalDomain_of_orbitSection** {D : Set X} (hD : NullMeasurableSet D μ) (hc : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ D) (hu : ∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x) (hfree : ∀ g : G, g ≠ 1 → μ {x | g • x = x} = 0) : IsFundamentalDomain G D μ where

theorem **IsAEOrbitSection.isFundamentalDomain** {D : Set X} (hD : IsAEOrbitSection G D μ) (hfree : ∀ g : G, g ≠ 1 → μ {x | g • x = x} = 0) : IsFundamentalDomain G D μ

theorem **measure_fixed_eq_zero_of_isFundamentalDomain** {D : Set X} (hD : IsFundamentalDomain G D μ) {g : G} (hg : g ≠ 1) : μ {x | g • x = x} = 0

theorem **map_restrict_le_map_restrict_of_ae_unique_of_ae_covers** {S B : Set X} (hS : NullMeasurableSet S μ) (hB : NullMeasurableSet B μ) (hu : ∀ᵐ x ∂μ, x ∈ S → ∀ g : G, g • x ∈ S → g • x = x) (hc : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ B) : (μ.restrict S).map (Quotient.mk (MulAction.orbitRel G X)) ≤ (μ.restrict B).map (Quotient.mk (MulAction.orbitRel G X))

theorem **quotMeasure_le_map_restrict** {B : Set X} (hB : NullMeasurableSet B μ) (hc : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ B) : quotMeasure G μ ≤ (μ.restrict B).map (Quotient.mk (MulAction.orbitRel G X))

theorem **quotMeasure_univ_le** {B : Set X} (hB : NullMeasurableSet B μ) (hc : ∀ᵐ x ∂μ, ∃ g : G, g • x ∈ B) : quotMeasure G μ univ ≤ μ B

theorem **quotMeasure_eq** (hD : IsAEOrbitSection G D μ) : quotMeasure G μ = (μ.restrict D).map (Quotient.mk (MulAction.orbitRel G X))

theorem **map_restrict_eq** (hD : IsAEOrbitSection G D μ) (hD' : IsAEOrbitSection G D' μ) : (μ.restrict D).map (Quotient.mk (MulAction.orbitRel G X)) = (μ.restrict D').map (Quotient.mk (MulAction.orbitRel G X))

theorem **measurePreserving_mk** (hD : IsAEOrbitSection G D μ) : MeasurePreserving (Quotient.mk (MulAction.orbitRel G X)) (μ.restrict D) (quotMeasure G μ)

theorem **quotMeasure_apply** (hD : IsAEOrbitSection G D μ) {U : Set (Quotient (MulAction.orbitRel G X))} (hU : MeasurableSet U) : quotMeasure G μ U = μ (Quotient.mk (MulAction.orbitRel G X) ⁻¹' U ∩ D)

theorem **quotMeasure_univ** (hD : IsAEOrbitSection G D μ) : quotMeasure G μ univ = μ D

theorem **measure_eq** (hD : IsAEOrbitSection G D μ) (hD' : IsAEOrbitSection G D' μ) : μ D = μ D'

theorem **lintegral_quotMeasure** (hD : IsAEOrbitSection G D μ) {f : Quotient (MulAction.orbitRel G X) → ℝ≥0∞} (hf : AEMeasurable f (quotMeasure G μ)) : ∫⁻ q, f q ∂quotMeasure G μ = ∫⁻ x in D, f (Quotient.mk _ x) ∂μ

theorem **integral_quotMeasure** (hD : IsAEOrbitSection G D μ) {E : Type*} [NormedAddCommGroup E] [NormedSpace ℝ E] {f : Quotient (MulAction.orbitRel G X) → E} (hf : AEStronglyMeasurable f (quotMeasure G μ)) : ∫ q, f q ∂quotMeasure G μ = ∫ x in D, f (Quotient.mk _ x) ∂μ

theorem **integrable_quotMeasure_iff** (hD : IsAEOrbitSection G D μ) {E : Type*} [NormedAddCommGroup E] {f : Quotient (MulAction.orbitRel G X) → E} (hf : AEStronglyMeasurable f (quotMeasure G μ)) : Integrable f (quotMeasure G μ) ↔ IntegrableOn (fun x => f (Quotient.mk _ x)) D μ

theorem **quotMeasure_apply_eq_zero_iff** (hD : IsAEOrbitSection G D μ) {U : Set (Quotient (MulAction.orbitRel G X))} (hU : MeasurableSet U) : quotMeasure G μ U = 0 ↔ μ (Quotient.mk (MulAction.orbitRel G X) ⁻¹' U) = 0

theorem **quotMeasure_eq_zero_iff** (hD : IsAEOrbitSection G D μ) : quotMeasure G μ = 0 ↔ μ = 0

theorem **quotMeasure_image_le** (hD : IsAEOrbitSection G D μ) {A : Set X} (hA : MeasurableSet A) : MeasurableSet (Quotient.mk (MulAction.orbitRel G X) '' A) ∧ quotMeasure G μ (Quotient.mk (MulAction.orbitRel G X) '' A) ≤ μ A

theorem **sigmaFinite_quotMeasure** (hD : IsAEOrbitSection G D μ) [SigmaFinite μ] : SigmaFinite (quotMeasure G μ)

theorem **quotMeasure_eq_of_isFundamentalDomain** {D : Set X} (hD : IsFundamentalDomain G D μ) : quotMeasure G μ = (μ.restrict D).map (Quotient.mk (MulAction.orbitRel G X))

theorem **quotMeasure_univ_eq_covolume** [h : HasFundamentalDomain G X μ] : quotMeasure G μ univ = covolume G X μ

theorem **lintegral_eq_lintegral_quotMeasure_tsum** [h : HasFundamentalDomain G X μ] {f : X → ℝ≥0∞} (hf : Measurable f) {F : Quotient (MulAction.orbitRel G X) → ℝ≥0∞} (hF : ∀ x, F (Quotient.mk _ x) = ∑' g : G, f (g • x)) : ∫⁻ x, f x ∂μ = ∫⁻ q, F q ∂quotMeasure G μ

theorem **exists_measurableSet_orbitSection** [ProperlyDiscontinuousSMul G X] : ∃ D : Set X, MeasurableSet D ∧ (∀ x, ∃ g : G, g • x ∈ D) ∧ ∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x

theorem **exists_isFundamentalDomain_of_free** [ProperlyDiscontinuousSMul G X] (hfree : ∀ (g : G) (x : X), g • x = x → g = 1) : ∃ D : Set X, MeasurableSet D ∧ (∀ x, ∃! g : G, g • x ∈ D) ∧ ∀ μ : Measure X, IsFundamentalDomain G D μ

theorem **exists_isAEOrbitSection** [ProperlyDiscontinuousSMul G X] (μ : Measure X) : ∃ D : Set X, MeasurableSet D ∧ IsAEOrbitSection G D μ

theorem **exists_isFundamentalDomain_of_ae_free** [ProperlyDiscontinuousSMul G X] (μ : Measure X) (hfree : μ {x | MulAction.stabilizer G x ≠ ⊥} = 0) : ∃ D : Set X, MeasurableSet D ∧ (∀ x, ∃ g : G, g • x ∈ D) ∧ (∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x) ∧ IsFundamentalDomain G D μ

theorem **exists_isFundamentalDomain_of_surjective** [ProperlyDiscontinuousSMul G X] {Q : Type*} [Group Q] [MulAction Q X] (φ : G →* Q) (hφ : Function.Surjective φ) (hact : ∀ g (x : X), φ g • x = g • x) (μ : Measure X) (hfree : ∀ g : G, φ g ≠ 1 → μ {x | g • x = x} = 0) : ∃ D : Set X, MeasurableSet D ∧ (∀ x, ∃ g : G, g • x ∈ D) ∧ (∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x) ∧ IsFundamentalDomain Q D μ

theorem **exists_isFundamentalDomain_range_toPermHom** [ProperlyDiscontinuousSMul G X] (μ : Measure X) (hfree : ∀ g : G, (∃ x : X, g • x ≠ x) → μ {x | g • x = x} = 0) : ∃ D : Set X, MeasurableSet D ∧ (∀ x, ∃ g : G, g • x ∈ D) ∧ (∀ x ∈ D, ∀ g : G, g • x ∈ D → g • x = x) ∧ IsFundamentalDomain (MulAction.toPermHom G X).range D μ

theorem **exists_isFundamentalDomain_and_quotMeasure_eq** [ProperlyDiscontinuousSMul G X] (μ : Measure X) [SMulInvariantMeasure G X μ] (hfree : μ {x | MulAction.stabilizer G x ≠ ⊥} = 0) : (∃ D : Set X, MeasurableSet D ∧ IsFundamentalDomain G D μ) ∧ ∀ D : Set X, IsFundamentalDomain G D μ → (μ.restrict D).map (Quotient.mk (MulAction.orbitRel G X)) = quotMeasure G μ

theorem **quotMeasure_eq_zero_iff** [ProperlyDiscontinuousSMul G X] (μ : Measure X) [SMulInvariantMeasure G X μ] : quotMeasure G μ = 0 ↔ μ = 0

theorem **countable_of_discrete_subgroup** : Countable Γ

theorem **exists_nhds_one_inter_subgroup** : ∃ V ∈ 𝒩 (1 : G), ∀ g ∈ Γ, g ∈ V → g = 1

theorem **exists_isFundamentalDomain_subgroup** : ∃ D : Set G, MeasurableSet D ∧ (∀ x, ∃! g : Γ, g • x ∈ D) ∧ ∀ μ : Measure G, IsFundamentalDomain Γ D μ

theorem **exists_isFundamentalDomain_subgroup_op** : ∃ D : Set G, MeasurableSet D ∧ (∀ x, ∃! g : Γ.op, g • x ∈ D) ∧ ∀ μ : Measure G, IsFundamentalDomain Γ.op D μ

theorem **hasFundamentalDomain_subgroup** (μ : Measure G) : HasFundamentalDomain Γ G μ

theorem **hasFundamentalDomain_subgroup_op** (μ : Measure G) : HasFundamentalDomain Γ.op G μ

theorem **exists_isFundamentalDomain_and_quotMeasure_eq_subgroup** (μ : Measure G) [μ.IsMulLeftInvariant] : (∃ D : Set G, MeasurableSet D ∧ IsFundamentalDomain Γ D μ) ∧ ∀ D : Set G, IsFundamentalDomain Γ D μ → (μ.restrict D).map (Quotient.mk (MulAction.orbitRel Γ G)) = quotMeasure Γ μ

theorem **exists_isFundamentalDomain_and_quotMeasure_eq_subgroup_op** (μ : Measure G) [μ.IsMulRightInvariant] : (∃ D : Set G, MeasurableSet D ∧ IsFundamentalDomain Γ.op D μ) ∧ ∀ D : Set G, IsFundamentalDomain Γ.op D μ → (μ.restrict D).map (Quotient.mk (MulAction.orbitRel Γ.op G)) = quotMeasure Γ.op μ

theorem **properlyDiscontinuousSMul_of_discrete_of_properSMul** {G X : Type*} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [T2Space G] [MulAction G X] [TopologicalSpace X] [T2Space X] [LocallyCompactSpace X] [ProperSMul G X] (Γ : Subgroup G) [DiscreteTopology Γ] : ProperlyDiscontinuousSMul Γ X

theorem **addQuotMeasure_zmultiples_one_univ** : addQuotMeasure (AddSubgroup.zmultiples (1 : ℝ)) (volume : Measure ℝ) univ = 1

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsIsUnitEvalOfMapResidueNeZero

theorem **IsPolynomialFunction.linear** (φ : Module.Dual K M) : IsPolynomialFunction K φ

theorem **IsPolynomialFunction.const** (c : K) : IsPolynomialFunction K (fun _ : M => c)

theorem **IsPolynomialFunction.add** {P Q : M → K} (hP : IsPolynomialFunction K P) (hQ : IsPolynomialFunction K Q) : IsPolynomialFunction K (fun x => P x + Q x)

theorem **IsPolynomialFunction.mul** {P Q : M → K} (hP : IsPolynomialFunction K P) (hQ : IsPolynomialFunction K Q) : IsPolynomialFunction K (fun x => P x * Q x)

theorem **IsPolynomialFunction.eval** {σ : Type*} (L : σ → Module.Dual K M) (p : MvPolynomial σ K) : IsPolynomialFunction K (fun x => MvPolynomial.eval (fun i => L i x) p)

theorem **IsPolynomialFunction.comp** {N : Type*} [AddCommMonoid N] [Module K N] {P : M → K} (hP : IsPolynomialFunction K P) (L : N →ₗ[K] M) : IsPolynomialFunction K (fun y => P (L y))

theorem **IsPolynomialFunction.continuous** {𝕜 E : Type*} [NontriviallyNormedField 𝕜] [CompleteSpace 𝕜] [AddCommGroup E] [Module 𝕜 E] [TopologicalSpace E] [IsTopologicalAddGroup E] [ContinuousSMul 𝕜 E] [T2Space E] [FiniteDimensional 𝕜 E] {P : E → 𝕜} (hP : IsPolynomialFunction 𝕜 P) : Continuous P

theorem **frameMul_frameMul** (x : ∀ s, l s → V s) (b : ∀ s, Matrix (l s) (m s) K) (c : ∀ s, Matrix (m s) (n s) K) : frameMul (frameMul x b) c = frameMul x (fun s => b s * c s)

theorem **frameMul_diagonal** [∀ s, DecidableEq (m s)] (x : ∀ s, m s → V s) (d : ∀ s, m s → K) : frameMul x (fun s => Matrix.diagonal (d s)) = fun s j => d s j • x s j

@[simp] theorem **entries_apply** {α : Type*} (b : ∀ s, Matrix (m s) (n s) α) (s : ι) (i : m s) (j : n s) : entries b ⟨s, i, j⟩ = b s i j

@[simp] theorem **ofEntries_apply** {α : Type*} (ξ : EntryIndex m n → α) (s : ι) (i : m s) (j : n s) : ofEntries ξ s i j = ξ ⟨s, i, j⟩

@[simp] theorem **entries_ofEntries** {α : Type*} (ξ : EntryIndex m n → α) : entries (ofEntries ξ) = ξ

@[simp] theorem **ofEntries_entries** {α : Type*} (b : ∀ s, Matrix (m s) (n s) α) : ofEntries (entries b) = b

theorem **HasTriangularWeight.smul** {a : ∀ s, Fin (r s) → ℕ} {P : (∀ s, Fin (r s) → V s) → W} (hP : HasTriangularWeight K a P) (x : ∀ s, Fin (r s) → V s) (d : ∀ s, Fin (r s) → K) (hd : ∀ s i, d s i ≠ 0) : P (fun s i => d s i • x s i) = (∏ s, ∏ i, d s i ^ a s i) • P x

theorem **HasTriangularWeight.of_continuous** [∀ s, AddCommGroup (V s)] [∀ s, Module 𝕜 (V s)] {a : ∀ s, Fin (r s) → ℕ} {P : (∀ s, Fin (r s) → V s) → W} (hP : HasTriangularWeight 𝕜 a P) (x : ∀ s, Fin (r s) → V s) (hc : Continuous fun b : ∀ s, Matrix (Fin (r s)) (Fin (r s)) 𝕜 => P (frameMul x b)) (b : ∀ s, Matrix (Fin (r s)) (Fin (r s)) 𝕜) (hb : ∀ s, (b s).IsUpperTriangular) : P (frameMul x b) = (∏ s, ∏ i, b s i i ^ a s i) • P x

theorem **exists_orthonormal_frameMul_eq** (hV : ∀ s, finrank 𝕜 (V s) = r s) (x : ∀ s, Fin (r s) → V s) : ∃ (y : ∀ s, Fin (r s) → V s) (b : ∀ s, Matrix (Fin (r s)) (Fin (r s)) 𝕜), (∀ s, Orthonormal 𝕜 (y s)) ∧ (∀ s, (b s).IsUpperTriangular) ∧ (∀ s i, ‖b s i i‖ ≤ ‖x s i‖) ∧ frameMul y b = x

theorem **HasTriangularWeight.norm_le_prod** (hV : ∀ s, finrank 𝕜 (V s) = r s) {a : ∀ s, Fin (r s) → ℕ} {P : (∀ s, Fin (r s) → V s) → W} (hP : HasTriangularWeight 𝕜 a P) (hc : Continuous P) {C : ℝ} (hC : ∀ y, (∀ s, Orthonormal 𝕜 (y s)) → ‖P y‖ ≤ C) (x : ∀ s, Fin (r s) → V s) : ‖P x‖ ≤ C * ∏ s, ∏ i, ‖x s i‖ ^ a s i

theorem **HasTriangularWeight.norm_le_prod_pow** (hV : ∀ s, finrank 𝕜 (V s) = r s) {a : ∀ s, Fin (r s) → ℕ} {P : (∀ s, Fin (r s) → V s) → W} (hP : HasTriangularWeight 𝕜 a P) (hc : Continuous P) {C : ℝ} (hC : ∀ y, (∀ s, Orthonormal 𝕜 (y s)) → ‖P y‖ ≤ C) {R : ι → ℝ} {x : ∀ s, Fin (r s) → V s} (hx : ∀ s i, ‖x s i‖ ≤ R s) : ‖P x‖ ≤ C * ∏ s, R s ^ ∑ i, a s i

theorem **HasTriangularWeight.poly** {V : ι → Type*} [∀ s, AddCommGroup (V s)] [∀ s, Module K (V s)] {a : ∀ s, Fin (r s) → ℕ} {P : (∀ s, Fin (r s) → V s) → K} (hP : HasTriangularWeight K a P) (w : ∀ s, Fin (r s) → V s) {p : MvPolynomial (EntryIndex (fun s => Fin (r s)) (fun s => Fin (r s))) K} (hp : ∀ ξ : ∀ s, Matrix (Fin (r s)) (Fin (r s)) K, P (frameMul w ξ) = MvPolynomial.eval (entries ξ) p) : HasTriangularWeightPoly a p

theorem **HasTriangularWeightPoly.hasTriangularWeight** {V : ι → Type*} [∀ s, AddCommGroup (V s)] [∀ s, Module K (V s)] {a : ∀ s, Fin (r s) → ℕ} {p : MvPolynomial (EntryIndex (fun s => Fin (r s)) (fun s => Fin (r s))) K} (hp : HasTriangularWeightPoly a p) (L : ∀ s, Fin (r s) → Module.Dual K (V s)) : HasTriangularWeight K a fun y : ∀ s, Fin (r s) → V s => MvPolynomial.eval (entries fun s => Matrix.of fun i j => L s i (y s j)) p

theorem **eval₂Hom_detProdPoly** {R S : Type*} [CommRing R] [CommRing S] {n : ι → Type*} [∀ s, Fintype (n s)] [∀ s, DecidableEq (n s)] (f : R →+* S) (ξ : EntryIndex n n → S) : MvPolynomial.eval₂Hom f ξ (detProdPoly R n) = ∏ s, (ofEntries ξ s).det

theorem **eval_detProdPoly** {R : Type*} [CommRing R] {n : ι → Type*} [∀ s, Fintype (n s)] [∀ s, DecidableEq (n s)] (ξ : EntryIndex n n → R) : MvPolynomial.eval ξ (detProdPoly R n) = ∏ s, (ofEntries ξ s).det

theorem **HasTriangularWeightPoly.norm_eval_le** {a : ∀ s, Fin (r s) → ℕ} {p : MvPolynomial (EntryIndex (fun s => Fin (r s)) (fun s => Fin (r s))) 𝕜} (hp : HasTriangularWeightPoly a p) {C : ℝ} (hC : ∀ u : ∀ s, Matrix (Fin (r s)) (Fin (r s)) 𝕜, (∀ s, u s ∈ Matrix.unitaryGroup

(Fin (r s)) 𝕜) → ‖MvPolynomial.eval (entries u) p‖ ≤ C) {R : ι → ℝ} (hR : ∀ s, 0 ≤ R s) {x : ∀ s, Matrix (Fin (r s)) (Fin (r s)) 𝕜} (hx : ∀ s j, ∑ i, ‖x s i j‖ ^ 2 ≤ R s ^ 2) : ‖MvPolynomial.eval (entries x) p‖ ≤ C * ∏ s, R s ^ ∑ i, a s i

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsLinearIndependentSqrtEqSuccMinQuad

theorem **mem_span_of_lt_succMinOf** {v : Fin n → V} (hv : LinearIndependent K v) (hS : ∀ j, v j ∈ S) (hN : ∀ j : Fin n, N (v j) ≤ succMinOf K N S (j + 1)) {i : ℕ} (hi : i ≤ n) {x : V} (hx : x ∈ S) (hx0 : 0 ≤ N x) (h : N x < succMinOf K N S (i + 1)) : x ∈ span K (v '' {j | (j : ℕ) < i})

theorem **exists_linearIndependent_eq_succMinOf** (hfin : ∀ R, {x | x ∈ S ∧ N x ≤ R}.Finite) (hN : ∀ x ∈ S, 0 ≤ N x) (hn : (n : Cardinal) ≤ Module.rank K (span K S)) : ∃ v : Fin n → V, LinearIndependent K v ∧ (∀ j, v j ∈ S) ∧ ∀ j : Fin n, N (v j) = succMinOf K N S (j + 1)

theorem **mem_span_of_norm_lt_succMin** {v : Fin n → E} (hv : LinearIndependent ℝ v) (hL : ∀ j, v j ∈ L) (hN : ∀ j : Fin n, ‖v j‖ ≤ succMin L (j + 1)) {i : ℕ} (hi : i ≤ n) {x : E} (hx : x ∈ L) (h : ‖x‖ < succMin L (i + 1)) : x ∈ span ℝ (v '' {j | (j : ℕ) < i})

theorem **succMin_le_norm_of_notMem_span** {v : Fin n → E} (hv : LinearIndependent ℝ v) (hL : ∀ j, v j ∈ L) (hN : ∀ j : Fin n, ‖v j‖ ≤ succMin L (j + 1)) {i : ℕ} (hi : i ≤ n) {x : E} (hx : x ∈ L) (h : x ∉ span ℝ (v '' {j | (j : ℕ) < i})) : succMin L (i + 1) ≤ ‖x‖

theorem **exists_linearIndependent_norm_eq_succMin** (hn : n ≤ finrank ℝ (span ℝ (L : Set E))) : ∃ v : Fin n → E, LinearIndependent ℝ v ∧ (∀ j, v j ∈ L) ∧ ∀ j : Fin n, ‖v j‖ = succMin L (j + 1)

theorem **exists_basis_norm_eq_succMin** : ∃ b : Basis (Fin (finrank ℝ E)) ℝ E, (∀ j, b j ∈ L) ∧ ∀ j : Fin (finrank ℝ E), ‖b j‖ = succMin L (j + 1)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsMinorProdEqSmulColActBasePoly

theorem **exists_linearEquiv_apply_eq** [FiniteDimensional K V] [FiniteDimensional K V'] (hd : finrank K V = finrank K V') {v : ι → V} {w : ι → V'} (hv : LinearIndependent K v) (hw : LinearIndependent K w) : ∃ e : V ≃ₗ[K] V', ∀ i, e (v i) = w i

theorem **exists_linearEquiv_comp_eq_and_apply_eq** [FiniteDimensional K V] {E E' : V →ₗ[K] W} (hE : Function.Surjective E) (hE' : Function.Surjective E') {v v' : ι → V} (hv : LinearIndependent K v) (hv' : LinearIndependent K v') (h : ∀ i, E (v i) = 0) (h' : ∀ i, E' (v' i) = 0) : ∃ g : V ≃ₗ[K] V, (∀ x, E' (g x) = E x) ∧ ∀ i, g (v i) = v' i

theorem **exists_linearIndependent_flag** {m : ℕ} (hm : m ≤ finrank K V) (v : Fin m → V) : ∃ w : Fin m → V, LinearIndependent K w ∧ ∀ b, v b ∈ Submodule.span K (w '' Set.Iic b)

theorem **exists_linearIndependent_flag_mem** {m : ℕ} (U : Submodule K V) (hm : m ≤ finrank K U) (v : Fin m → V) (hv : ∀ b, v b ∈ U) : ∃ w : Fin m → V, LinearIndependent K w ∧ (∀ a, w a ∈ U) ∧ ∀ b, v b ∈ Submodule.span K (w '' Set.Iic b)

theorem **card_le_card_add_finrank_ker** [Fintype ι] (ε : ι → κ → K) : Fintype.card κ ≤ Fintype.card ι + finrank K (LinearMap.ker (Matrix.of ε).mulVecLin)

theorem **mem_ker_mulVecLin_of** (ε : ι → κ → K) (x : κ → K) : x ∈ LinearMap.ker (Matrix.of ε).mulVecLin ↔ ∀ a, ε a ⬝ᵥ x = 0

theorem **surjective_mulVecLin_of** [Finite ι] {ε : ι → κ → K} (hε : LinearIndependent K ε) : Function.Surjective (Matrix.of ε).mulVecLin

theorem **exists_covecAct_eq_and_vecAct_eq** {ε ε' : ι → κ → K} {φ φ' : ι' → κ → K} (hε : LinearIndependent K ε) (hε' : LinearIndependent K ε') (hφ : LinearIndependent K φ) (hφ' : LinearIndependent K φ') (h : ∀ a b, ε a ⬝ᵥ φ b = 0) (h' : ∀ a b, ε' a ⬝ᵥ φ' b = 0) : ∃ g : GL κ K, (∀ a, covecAct g (ε a) = ε' a) ∧ ∀ b, vecAct g (φ b) = φ' b

theorem **exists_linearIndependent_perp_flag** {p q : ℕ} (hpq : p + q ≤ Fintype.card κ) (ε : Fin p → κ → K) (φ : Fin q → κ → K) (h : ∀ a b, ε a ⬝ᵥ φ b = 0) : ∃ (ε' : Fin p → κ → K) (φ' : Fin q → κ → K), LinearIndependent K ε' ∧ LinearIndependent K φ' ∧ (∀ a b, ε' a ⬝ᵥ φ' b = 0) ∧ (∀ a, ε a ∈ Submodule.span K (ε' '' Set.Iic a)) ∧ ∀ b, φ b ∈ Submodule.span K (φ' '' Set.Iic b)

theorem **exists_det_eq_smul_of_mem_span** {A M ι : Type*} [CommRing A] [Algebra R A] [AddCommGroup M] [Module R M] [Fintype ι] [DecidableEq ι] (L : ι → M →ₗ[R] A) (v w : ι → M) (h : ∀ b, v b ∈ Submodule.span R (Set.range w)) : ∃ r : R, (Matrix.of fun a b => L a (v b)).det = r • (Matrix.of fun a b => L a (w b)).det

theorem **mem_span_range_lead** {M : Type*} [AddCommGroup M] [Module R M] {m : ℕ} {v w : Fin m → M} (h : ∀ b, v b ∈ Submodule.span R (w '' Set.Iic b)) (i : Fin m) (b : Fin (i.val + 1)) : v (lead i b) ∈ Submodule.span R (Set.range fun a => w (lead i a))

theorem **exists_xMinor_eq_smul** (t : T) (i : Fin p) {ε ε' : Fin p → κ → R} (h : ∀ a, ε a ∈ Submodule.span R (ε' '' Set.Iic a)) : ∃ r : R, (xMinor t i ε : Poly R T p q κ) = r • xMinor t i ε'

theorem **exists_yMinor_eq_smul** (t : T) (i : Fin q) {φ φ' : Fin q → κ → R} (h : ∀ b, φ b ∈ Submodule.span R (φ' '' Set.Iic b)) : ∃ r : R, (yMinor t i φ : Poly R T p q κ) = r • yMinor t i φ'

theorem **exists_minorProd_eq_smul** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {ε ε' : T → Fin p → κ → R} {φ φ' : T → Fin q → κ → R} (hε : ∀ t a, ε t a ∈ Submodule.span R (ε' t '' Set.Iic a)) (hφ : ∀ t b, φ t b ∈ Submodule.span R (φ' t '' Set.Iic b)) : ∃ r : R, minorProd c c' ε φ = r • minorProd c c' ε' φ'

theorem **exists_act_eq_of_isPerp** [Fintype κ] [DecidableEq κ] {ε ε' : T → Fin p → κ → K} {φ φ' : T → Fin q → κ → K} (hε : ∀ t, LinearIndependent K (ε t)) (hε' : ∀ t, LinearIndependent K (ε' t)) (hφ : ∀ t, LinearIndependent K (φ t)) (hφ' : ∀ t, LinearIndependent K (φ' t)) (h : IsPerp ε φ) (h' : IsPerp ε' φ') : ∃ g : T → GL κ K, (∀ t a, covecAct (g t) (ε t a) = ε' t a) ∧ ∀ t b, vecAct (g t) (φ t b) = φ' t b

theorem **exists_colAct_minorProd_eq** [Fintype κ] [DecidableEq κ] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {ε ε' : T → Fin p → κ → K} {φ φ' : T → Fin q → κ → K} (hε : ∀ t, LinearIndependent K (ε t)) (hε' : ∀ t, LinearIndependent K (ε' t)) (hφ : ∀ t, LinearIndependent K (φ t)) (hφ' : ∀ t, LinearIndependent K (φ' t)) (h : IsPerp ε φ) (h' : IsPerp ε' φ') : ∃ g : T → GL κ K, colAct K T p q κ g (minorProd c c' ε φ) = minorProd c c' ε' φ'

theorem **IsPerp.exists_minorProd_eq_smul** [Fintype κ] (hpq : p + q ≤ Fintype.card κ) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {ε : T → Fin p → κ → K} {φ : T → Fin q → κ → K} (h : IsPerp ε φ) : ∃ (r : K) (ε' : T → Fin p → κ → K) (φ' : T → Fin q → κ → K), (∀ t, LinearIndependent K (ε' t)) ∧ (∀ t, LinearIndependent K (φ' t)) ∧ IsPerp ε' φ' ∧ minorProd c c' ε φ = r • minorProd c c' ε' φ'

theorem **exists_colAct_basePoly_eq** {ε : T → Fin p → Fin n → K} {φ : T → Fin q → Fin n → K} (hε : ∀ t, LinearIndependent K (ε t)) (hφ : ∀ t, LinearIndependent K (φ t)) (hp : IsPerp ε φ) : ∃ g : T → GL (Fin n) K, colAct K T p q (Fin n) g (basePoly K h c c') = minorProd c c' ε φ

theorem **exists_minorProd_eq_smul_colAct_basePoly** {ε : T → Fin p → Fin n → K} {φ : T → Fin q → Fin n → K} (hp : IsPerp ε φ) : ∃ (r : K) (g : T → GL (Fin n) K), minorProd c c' ε φ = r • colAct K T p q (Fin n) g (basePoly K h c c')

theorem **minorProd_mem_orbitSpan** {ε : T → Fin p → Fin n → K} {φ : T → Fin q → Fin n → K} (hp : IsPerp ε φ) : minorProd c c' ε φ ∈ orbitSpan K h c c'

theorem **linearIndependent_covecAct** {R ι : Type*} [CommRing R] [Fintype κ] [DecidableEq κ] (g : GL κ R) {ε : ι → κ → R} (hε : LinearIndependent R ε) : LinearIndependent R fun a => covecAct g (ε a)

theorem **linearIndependent_vecAct** {R ι : Type*} [CommRing R] [Fintype κ] [DecidableEq κ] (g : GL κ R) {φ : ι → κ → R} (hφ : LinearIndependent R φ) : LinearIndependent R fun b => vecAct g (φ b)

theorem **range_colAct_basePoly** : (Set.range fun g : T → GL (Fin n) K => colAct K T p q (Fin n) g (basePoly K h c c')) = {P | ∃ (ε : T → Fin p → Fin n → K) (φ : T → Fin q → Fin n → K), (∀ t, LinearIndependent K (ε t)) ∧ (∀ t, LinearIndependent K (φ t)) ∧ IsPerp ε φ ∧ P = minorProd c c' ε φ}

theorem **orbitSpan_eq_span_linearIndependent** : orbitSpan K h c c' = Submodule.span K {P | ∃ (ε : T → Fin p → Fin n → K) (φ : T → Fin q → Fin n → K), (∀ t, LinearIndependent K (ε t)) ∧ (∀ t, LinearIndependent K (φ t)) ∧ IsPerp ε φ ∧ P = minorProd c c' ε φ}

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsMulMulConjTransposeEqOne

theorem **card_eq_card_selfAdjoint_sq** (hs : ∃ a : K, star a ≠ a) : Nat.card K = Nat.card (selfAdjoint K) ^ 2

theorem **exists_mul_star_eq_of_norm_eq_one** (hs : ∃ a : K, star a ≠ a) {u : K} (hu : u * star u = 1) : ∃ z : K, z ≠ 0 ∧ u * star z = z

private theorem **mem_ker_normHom** (z : Kˣ) : z ∈ (normHom K).ker ↔ (z : K) * star (z : K) = 1

private theorem **mem_ker_quotHom** (z : Kˣ) : z ∈ (quotHom K).ker ↔ star (z : K) = z

private theorem **range_quotHom** (hs : ∃ a : K, star a ≠ a) : (quotHom K).range = (normHom K).ker

private theorem **card_ker_mul_card_range** {G H : Type*} [Group G] [Group H] (f : G →* H) : Nat.card f.ker * Nat.card f.range = Nat.card G

private theorem **range_normHom** [Finite K] (hs : ∃ a : K, star a ≠ a) : (normHom K).range = (quotHom K).ker

theorem **exists_mul_star_eq** [Finite K] (hs : ∃ a : K, star a ≠ a) {c : K} (hc : star c = c) : ∃ t : K, t * star t = c

theorem **card_selfAdjoint** {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) : Nat.card (selfAdjoint K) = q

theorem **two_le_of_card_eq_sq** {K : Type*} [Field K] [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) : 2 ≤ q

theorem **card_norm_eq_one** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) : Nat.card {u : K // u * star u = 1} = q + 1

theorem **sphereCard_succ** [Fintype K] (n : ℕ) (c : K) : sphereCard K (n + 1) c = ∑ t : K, sphereCard K n (c - t * star t)

theorem **sphereCard_eq_one** [Finite K] (hs : ∃ a : K, star a ≠ a) (n : ℕ) {c : K} (hc : star c = c) (h0 : c ≠ 0) : sphereCard K n c = sphereCard K n 1

theorem **sphereCard_formula** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) (n : ℕ) : (q : ℤ) * sphereCard K n 1 = q ^ n * (q ^ n - (-1) ^ n) ∧ (q : ℤ) * sphereCard K n 0 = q ^ (2 * n) + (-1) ^ n * (q - 1) * q ^ n

theorem **sphereCard_succ_one** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) (n : ℕ) : (sphereCard K (n + 1) 1 : ℤ) = q ^ n * (q ^ (n + 1) - (-1) ^ (n + 1))

theorem **IsOrthonormal.linearIndependent** {ι : Type*} [DecidableEq ι] {s : ι → V} (hs : IsOrthonormal h s) : LinearIndependent K s

theorem **IsOrthonormal.apply_eq_sum** {ι : Type*} [Fintype ι] [DecidableEq ι] {b : Basis ι K V} (hb : IsOrthonormal h b) (x y : V) : h x y = ∑ i, b.repr x i * star (b.repr y i)

theorem **exists_apply_self_eq_one** [Finite K] (hs : ∃ a : K, star a ≠ a) (hh : ∀ x y, h y x = star (h x y)) {x y : V} (hxy : h x y ≠ 0) : ∃ v, h v v = 1

@[simp] private theorem **res_apply** (v : V) (x y : perp h v) : res h v x y = h x y

private theorem **finrank_perp** [FiniteDimensional K V] {v : V} (hv : h v v = 1) {n : ℕ} (hn : finrank K V = n + 1) : finrank K (perp h v) = n

private theorem **res_nondeg** (hh : ∀ x y, h y x = star (h x y)) (hnd : ∀ x, (∀ y, h x y = 0) → x = 0) {v : V} (hv : h v v = 1) (x : perp h v) (hx : ∀ y, res h v x y = 0) : x = 0

private theorem **isOrthonormal_cons** (hh : ∀ x y, h y x = star (h x y)) {v : V} (hv : h v v = 1) {n : ℕ} {s : Fin n → perp h v} (hs : IsOrthonormal (res h v) s) : IsOrthonormal h (Fin.cons v fun i => (s i : V) : Fin (n + 1) → V)

private theorem **card_frames_succ** [Finite V] (hh : ∀ x y, h y x = star (h x y)) (n : ℕ) : Nat.card {s : Fin (n + 1) → V // IsOrthonormal h s} = Nat.card (Σ v : {v : V // h v v = 1}, {s : Fin n → perp h v // IsOrthonormal (res h v) s})

private theorem **exists_frame** [Finite K] (hs : ∃ a : K, star a ≠ a) (n : ℕ) : ∀ (V : Type u) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (h : V →ₗ[K] V →ₗ⋆[K] K), (∀ x y, h y x = star (h x y)) → (∀ x, (∀ y, h x y = 0) → x = 0) → finrank K V = n → ∃ s : Fin n → V, IsOrthonormal h s

theorem **exists_orthonormal_basis** [Finite K] (hs : ∃ a : K, star a ≠ a) {V : Type*} [AddCommGroup V] [Module K V] [FiniteDimensional K V] {h : V →ₗ[K] V →ₗ⋆[K] K} (hh : ∀ x y, h y x = star (h x y)) (hnd : ∀ x, (∀ y, h x y = 0) → x = 0) {ι : Type*} [Fintype ι] [DecidableEq ι] (hι : Fintype.card ι = finrank K V) : ∃ b : Basis ι K V, IsOrthonormal h b

theorem **card_apply_self_eq_one** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) {V : Type*} [AddCommGroup V] [Module K V] [FiniteDimensional K V] {h : V →ₗ[K] V →ₗ⋆[K] K} (hh : ∀ x y, h y x = star (h x y)) (hnd : ∀ x, (∀ y, h x y = 0) → x = 0) {n : ℕ} (hn : finrank K V = n + 1) : (Nat.card {v : V // h v v = 1} : ℤ) = q ^ n * (q ^ (n + 1) - (-1) ^ (n + 1))

private theorem **card_frames** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) (n : ℕ) : ∀ (V : Type u) [AddCommGroup V] [Module K V] [FiniteDimensional K V] (h : V →ₗ[K] V →ₗ⋆[K] K), (∀ x y, h y x = star (h x y)) → (∀ x, (∀ y, h x y = 0) → x = 0) → finrank K V = n → (Nat.card {s : Fin n → V // IsOrthonormal h s} : ℤ) = ∏ i ∈ Finset.range n, (q : ℤ) ^ i * (q ^ (i + 1) - (-1) ^ (i + 1))

private theorem **gramForm_apply** {ι : Type*} [Fintype ι] (H : Matrix ι ι K) (x y : ι → K) : gramForm H x y = ∑ i, ∑ j, x i * H i j * star (y j)

private theorem **gramForm_herm** {ι : Type*} [Fintype ι] {H : Matrix ι ι K} (hH : H.IsHermitian) (x y : ι → K) : gramForm H y x = star (gramForm H x y)

private theorem **gramForm_nondeg** {ι : Type*} [Fintype ι] [DecidableEq ι] {H : Matrix ι ι K} (hH : H.det ≠ 0) (x : ι → K) (hx : ∀ y, gramForm H x y = 0) : x = 0

private theorem **isOrthonormal_gramForm_iff** {ι κ : Type*} [Fintype ι] [DecidableEq κ] (H : Matrix ι ι K) (P : Matrix κ ι K) : IsOrthonormal (gramForm H) (fun i => P i) ↔ P * H * P.conjTranspose = 1

theorem **card_unitaryGroup_int** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) (ι : Type*) [Fintype ι] [DecidableEq ι] : (Nat.card (Matrix.unitaryGroup ι K) : ℤ) = ∏ i ∈ Finset.range (Fintype.card ι), (q : ℤ) ^ i * (q ^ (i + 1) - (-1) ^ (i + 1))

theorem **card_specialUnitaryGroup_mul** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) (ι : Type*) [Fintype ι] [DecidableEq ι] [Nonempty ι] : Nat.card (Matrix.specialUnitaryGroup ι K) * (q + 1) = Nat.card (Matrix.unitaryGroup ι K)

private theorem **prod_eq** (q ε : ℚ) (hq : q ≠ 0) (n : ℕ) : ∏ i ∈ Finset.range (n + 1), q ^ i * (q ^ (i + 1) - ε ^ (i + 1)) = (q ^ ((n + 1) ^ 2 - 1) * ∏ i ∈ Finset.Icc 2 (n + 1), (1 - ε ^ i * (q ^ i)⁻¹)) * (q - ε)

theorem **card_specialUnitaryGroup** [Finite K] {q : ℕ} (hq : Nat.card K = q ^ 2) (hs : ∃ a : K, star a ≠ a) (ι : Type*) [Fintype ι] [DecidableEq ι] {r : ℕ} (hr : Fintype.card ι = r) (h0 : 1 ≤ r) : (Nat.card (Matrix.specialUnitaryGroup ι K) : ℚ) = (q : ℚ) ^ (r ^ 2 - 1) * ∏ i ∈ Finset.Icc 2 r, (1 - (-1) ^ i * ((q : ℚ) ^ i)⁻¹)

theorem **card_specialLinearGroup_mul** (ι : Type*) [Fintype ι] [DecidableEq ι] [Nonempty ι] : Nat.card (Matrix.SpecialLinearGroup ι F) * (Nat.card F - 1) = Nat.card (GL ι F)

private theorem **prod_GL** (q : ℚ) (n : ℕ) : ∏ i ∈ Finset.range n, (q ^ n - q ^ i) = ∏ i ∈ Finset.range n, q ^ i * (q ^ (i + 1) - 1 ^ (i + 1))

theorem **card_specialLinearGroup** {q : ℕ} (hq : Nat.card F = q) (ι : Type*) [Fintype ι] [DecidableEq ι] {r : ℕ} (hr : Fintype.card ι = r) (h0 : 1 ≤ r) : (Nat.card (Matrix.SpecialLinearGroup ι F) : ℚ) = (q : ℚ) ^ (r ^ 2 - 1) * ∏ i ∈ Finset.Icc 2 r, (1 - ((q : ℚ) ^ i)⁻¹)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsNeZeroForallNormLeOfCovolumeLe

theorem **covolume_eq_relIndex_mul_of_haar** (L' L : Submodule ℤ V) [DiscreteTopology L'] [IsZLattice ℝ L'] [DiscreteTopology L] [IsZLattice ℝ L] (h : L' ≤ L) : ZLattice.covolume L' μ = (L'.toAddSubgroup.relIndex L.toAddSubgroup : ℝ) * ZLattice.covolume L μ

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsNeZeroMemFractionalIdealNormSqLe

theorem **exists_ne_zero_mem_fractionalIdeal_place_le** (I : (FractionalIdeal (𝒪 K)⁰ K)ˣ) (g : InfinitePlace K → ℝ≥0) (h : minkowskiBound K I ≤ volume (convexBodyLT K g)) : ∃ a ∈ (I : FractionalIdeal (𝒪 K)⁰ K), a ≠ 0 ∧ ∀ w : InfinitePlace K, w a ≤ g w

theorem **exists_ne_zero_mem_fractionalIdeal_norm_sq_le** (hf : finrank ℚ E = 2 * f) (J : FractionalIdeal (𝒪 E)⁰ E) (hJ : J ≠ 0) : ∃ a ∈ J, a ≠ 0 ∧ ∀ τ : E →+* ℂ, ‖τ a‖ ^ 2 ≤ 2 / Real.pi * ((FractionalIdeal.absNorm J : ℝ) * √|(discr E : ℝ)|) ^ ((f : ℝ)⁻¹)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsNeZeroNormSqLeMulCovolumeRpow

theorem **measure_le_two_pow_mul_covolume** (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] (μ : Measure E) [μ.IsAddHaarMeasure] {s : Set E} (hsymm : ∀ x ∈ s, -x ∈ s) (hconv : Convex ℝ s) (hL : ∀ x ∈ L, x ≠ 0 → x ∉ s) : μ s ≤ ENNReal.ofReal (2 ^ finrank ℝ E * ZLattice.covolume L μ)

theorem **exists_linearMap_det_eq_prod_inv_succMin** (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] : ∃ T : E →ₗ[ℝ] E, LinearMap.det T = (∏ i : Fin (finrank ℝ E), succMin L (i + 1))⁻¹ ∧ ∀ x ∈ L, x ≠ 0 → 1 ≤ ‖T x‖

theorem **volume_real_unitBall_pos** : 0 < volume.real (Metric.ball (0 : E) 1)

theorem **two_div_sqrt_pow_le_volume_unitBall** : (2 / √(finrank ℝ E : ℝ)) ^ finrank ℝ E ≤ volume.real (Metric.ball (0 : E) 1)

theorem **two_pow_div_volume_unitBall_le** : 2 ^ finrank ℝ E / volume.real (Metric.ball (0 : E) 1) ≤ (finrank ℝ E : ℝ) ^ ((finrank ℝ E : ℝ) / 2)

theorem **prod_succMin_mul_volume_unitBall_le** : (∏ i : Fin (finrank ℝ E), succMin L (i + 1)) * volume.real (Metric.ball (0 : E) 1) ≤ 2 ^ finrank ℝ E * ZLattice.covolume L

theorem **prod_succMin_le_two_pow_div_mul_covolume** : ∏ i : Fin (finrank ℝ E), succMin L (i + 1) ≤ 2 ^ finrank ℝ E / volume.real (Metric.ball (0 : E) 1) * ZLattice.covolume L

theorem **prod_succMin_le_rpow_mul_covolume** : ∏ i : Fin (finrank ℝ E), succMin L (i + 1) ≤ (finrank ℝ E : ℝ) ^ ((finrank ℝ E : ℝ) / 2) * ZLattice.covolume L

theorem **succMin_mul_minNorm_pow_le** {k : ℕ} (h1 : 1 ≤ k) (hk : k ≤ finrank ℝ E) : succMin L k * minNorm L ^ (finrank ℝ E - 1) ≤ (finrank ℝ E : ℝ) ^ ((finrank ℝ E : ℝ) / 2) * ZLattice.covolume L

theorem **prod_succMin_le_rpow_mul_spanCovol** : ∏ i : Fin (finrank ℤ L), succMin L (i + 1) ≤ (finrank ℤ L : ℝ) ^ ((finrank ℤ L : ℝ) / 2) * spanCovol L

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsOrthonormalOfLinearIndependent

theorem **norm_orthonormalBasis_det_le** (b : OrthonormalBasis ι 𝕜 E) (x : ι → E) : ‖b.toBasis.det x‖ ≤ ∏ k, ‖x k‖

theorem **norm_det_le_prod_col** (A : Matrix ι ι 𝕜) : ‖A.det‖ ≤ ∏ j, √(∑ i, ‖A i j‖ ^ 2)

theorem **norm_det_le_prod_row** (A : Matrix ι ι 𝕜) : ‖A.det‖ ≤ ∏ i, √(∑ j, ‖A i j‖ ^ 2)

theorem **norm_det_sq_le_prod_col** (A : Matrix ι ι 𝕜) : ‖A.det‖ ^ 2 ≤ ∏ j, ∑ i, ‖A i j‖ ^ 2

theorem **norm_det_sq_le_prod_row** (A : Matrix ι ι 𝕜) : ‖A.det‖ ^ 2 ≤ ∏ i, ∑ j, ‖A i j‖ ^ 2

theorem **det_gram_nonneg** (x : ι → E) : 0 ≤ (gram 𝕜 x).det

theorem **re_det_gram_nonneg** (x : ι → E) : 0 ≤ RCLike.re (gram 𝕜 x).det

theorem **gram_sum_smul** [Fintype κ] (e : ι → E) (m : Matrix ι κ 𝕜) : gram 𝕜 (fun j => ∑ i, m i j • e i) = mᴴ * gram 𝕜 e * m

theorem **re_det_gram_sum_smul** (e : ι → E) (m : Matrix ι ι 𝕜) : RCLike.re (gram 𝕜 (fun j => ∑ i, m i j • e i)).det = ‖m.det‖ ^ 2 * RCLike.re (gram 𝕜 e).det

theorem **exists_orthonormal_of_linearIndependent** {x : ι → E} (hx : LinearIndependent 𝕜 x) : ∃ (e : ι → E) (m : Matrix ι ι 𝕜), Orthonormal 𝕜 e ∧ ∀ j, x j = ∑ i, m i j • e i

theorem **norm_sum_smul_sq** {e : ι → E} (he : Orthonormal 𝕜 e) (c : ι → 𝕜) : ‖∑ i, c i • e i‖ ^ 2 = ∑ i, ‖c i‖ ^ 2

theorem **re_det_gram_le_prod** (x : ι → E) : RCLike.re (gram 𝕜 x).det ≤ ∏ k, ‖x k‖ ^ 2

theorem **det_gram_comp_equiv** [DecidableEq ι] (x : ι → E) (σ : κ ≃ ι) : (gram 𝕜 (x ∘ σ)).det = (gram 𝕜 x).det

theorem **sum_norm_apply_sq_le** {e : ι → E} (he : Orthonormal 𝕜 e) (φ : E →L[𝕜] 𝕜) : ∑ i, ‖φ (e i)‖ ^ 2 ≤ ‖φ‖ ^ 2

theorem **norm_det_apply_sq_le** (φ : ι → E →L[𝕜] 𝕜) (x : ι → E) : ‖(of fun j k => φ j (x k)).det‖ ^ 2 ≤ RCLike.re (gram 𝕜 x).det * ∏ j, ‖φ j‖ ^ 2

theorem **norm_det_apply_le** (φ : ι → E →L[𝕜] 𝕜) (x : ι → E) : ‖(of fun j k => φ j (x k)).det‖ ≤ (∏ k, ‖x k‖) * ∏ j, ‖φ j‖

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsQr

theorem **_root_.GlobalTheory.QRDecomp.exists_qr** (n m : ℕ) (A : Matrix (Fin n) (Fin m) ℂ) (hA : IsUnit (Aᴴ * A).det) : ∃ (Q : Matrix (Fin n) (Fin m) ℂ) (u : Matrix (Fin m) (Fin m) ℂ), Qᴴ * Q = 1 ∧ u.BlockTriangular id ∧ IsUnit u.det ∧ A = Q * u

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsShortBasisAdaptedOfSubsetSpan

theorem **exists_add_orthogonal_norm_sq_le** (W : Submodule ℝ E) (u : E) : ∃ p ∈ W, ∃ q ∈ Wᗮ, u = p + q ∧ ‖q‖ ^ 2 ≤ ‖u‖ ^ 2

theorem **exists_mem_span_int_norm_sub_sq_le_sum** {ι : Type*} (v : ι → E) (s : Finset ι) {x : E} (hx : x ∈ span ℝ (v '' s)) : ∃ w ∈ span ℤ (v '' s), ‖x − w‖ ^ 2 ≤ (1 / 4) * ∑ i ∈ s, ‖v i‖ ^ 2

theorem **exists_short_generator_inf_span_insert** (L : Submodule ℤ E) [DiscreteTopology L] {ι : Type*} (v : ι → E) (s : Finset ι) (hv : ∀ i ∈ s, v i ∈ L) {u : E} (huL : u ∈ L) (hu : u ∉ span ℝ (v '' s)) : ∃ b : E, L ⊓ (span ℝ (insert u (v '' s))).restrictScalars ℤ = span ℤ {b} ⊔ L ⊓ (span ℝ (v '' s)).restrictScalars ℤ ∧ ‖b‖ ^ 2 ≤ ‖u‖ ^ 2 + (1 / 4) * ∑ i ∈ s, ‖v i‖ ^ 2

theorem **exists_short_family_adapted** (L : Submodule ℤ E) [DiscreteTopology L] {n : ℕ} {v : Fin n → E} (hv : LinearIndependent ℝ v) (hL : ∀ i, v i ∈ L) : ∃ b : Fin n → E, LinearIndependent ℝ b ∧ (∀ s : ℕ, span ℤ (b '' {i : Fin n | (i : ℕ) < s}) = L ⊓ (span ℝ (v '' {i : Fin n | (i : ℕ) < s})).restrictScalars ℤ) ∧ ∀ i, ‖b i‖ ^ 2 ≤ ‖v i‖ ^ 2 + (1 / 4) * ∑ j ∈ Finset.Iio i, ‖v j‖ ^ 2

theorem **exists_short_basis_adapted_of_subset_span** (L : Submodule ℤ E) [DiscreteTopology L] {n : ℕ} {v : Fin n → E} (hv : LinearIndependent ℝ v) (hL : ∀ i, v i ∈ L) (hspan : (L : Set E) ⊆ span ℝ (Set.range v)) : ∃ b : Basis (Fin n) ℤ L, (∀ s : ℕ, span ℤ ((fun i => (b i : E)) '' {i : Fin n | (i : ℕ) < s}) = L ⊓ (span ℝ (v '' {i : Fin n | (i : ℕ) < s})).restrictScalars ℤ) ∧ ∀ i, ‖(b i : E)‖ ^ 2 ≤ ‖v i‖ ^ 2 + (1 / 4) * ∑ j ∈ Finset.Iio i, ‖v j‖ ^ 2

theorem **norm_sq_add_sum_Iio_le** {n : ℕ} {v : Fin n → E} {i : Fin n} {r : ℝ} (h : ∀ j ≤ i, ‖v j‖ ≤ r) : ‖v i‖ ^ 2 + (1 / 4) * ∑ j ∈ Finset.Iio i, ‖v j‖ ^ 2 ≤ (((i : ℕ) + 1 + 3 : ℝ) / 4) * r ^ 2

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsUnitaryCommMulEqOfGram

private lemma **exists_unitary_fix_proj_of_gram_eq_of_mul_eq_zero** {n m : ℕ} (Q : Matrix (Fin n) (Fin n) ℂ) (hQ : Q^H = Q) (X Y : Matrix (Fin n) (Fin m) ℂ) (h : X^H * X = Y^H * Y) (hX : Q * X = 0) (hY : Q * Y = 0) : ∃ W : Matrix (Fin n) (Fin n) ℂ, W^H * W = 1 ∧ W * Q = Q ∧ W * X = Y

theorem **_root_.GlobalTheory.HCXi.exists_unitary_comm_mul_eq_of_gram** {n m : ℕ} (H : Matrix (Fin n) (Fin n) ℂ) (hH : H^H = H) (hHH : H * H = 1) (A B : Matrix (Fin n) (Fin m) ℂ) (h1 : A^H * A = B^H * B) (h2 : A^H * H * A = B^H * H * B) : ∃ U : Matrix (Fin n) (Fin n) ℂ, U^H * U = 1 ∧ U * H = H * U ∧ U * A = B

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ExistsUnitaryOfGramEq

theorem **exists_isometry_comp** {W V : Type} [AddCommGroup W] [Module ℂ W] [NormedAddCommGroup V] [InnerProductSpace ℂ V] [FiniteDimensional ℂ V] (a b : W →ₗ[ℂ] V) (h : ∀ x, ‖b x‖ = ‖a x‖) : ∃ u : V →ₗᵢ[ℂ] V, ∀ x, u (a x) = b x

theorem **exists_unitary_mul_eq** {n : ℕ} {ι : Type} [Fintype ι] [DecidableEq ι] (A B : Matrix (Fin n) ι ℂ) (h : A^H * A = B^H * B) : ∃ U : Matrix (Fin n) (Fin n) ℂ, U^H * U = 1 ∧ U * A = B

theorem **exists_unitary_of_gram_eq** : ∀ (n m : ℕ) (E : Matrix (Fin n) (Fin n) ℂ) (A B : Matrix (Fin n) (Fin m) ℂ), E^H = E → E * E = E → E * A = A → E * B = B → A^H * A = B^H * B → ∃ U : Matrix (Fin n) (Fin n) ℂ, U^H * U = 1 ∧ U * E = E * U ∧ U * (1 − E) = 1 − E ∧ U * A = B

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FderivIntegralApplyOfHolomorphicOfDominated

theorem **aestronglyMeasurable_fderiv_apply_of_param** {F : T → E → V} {x₀ : E} (hmeas : ∀ᶠ y in 𝓝 x₀, AEStronglyMeasurable (fun t => F t y) μ) (hdiff : ∀ᵐ t ∂μ, DifferentiableAt ℂ (F t) x₀) (v : E) : AEStronglyMeasurable (fun t => fderiv ℂ (F t) x₀ v) μ

theorem **aestronglyMeasurable_fderiv_of_param** {F : T → E → V} {x₀ : E} (hmeas : ∀ᶠ y in 𝓝 x₀, AEStronglyMeasurable (fun t => F t y) μ) (hdiff : ∀ᵐ t ∂μ, DifferentiableAt ℂ (F t) x₀) : AEStronglyMeasurable (fun t => fderiv ℂ (F t) x₀) μ

theorem **hasFDerivAt_integral_of_holomorphic_of_dominated** {F : T → E → V} {x₀ : E} {s : Set E} {bound : T → ℝ} (hs : s ∈ 𝓝 x₀) (hmeas : ∀ y ∈ s, AEStronglyMeasurable (fun t => F t y) μ) (hdiff : ∀ᵐ t ∂μ, DifferentiableOn ℂ (F t) s) (hbound : ∀ᵐ t ∂μ, ∀ y ∈ s, ‖F t y‖ ≤ bound t) (hint : Integrable bound μ) : Integrable (fun t => F t x₀) μ ∧ Integrable (fun t => fderiv ℂ (F t) x₀) μ ∧ HasFDerivAt (fun y => ∫ t, F t y ∂μ) (∫ t, fderiv ℂ (F t) x₀ ∂μ) x₀

theorem **differentiableOn_integral_of_holomorphic_of_locally_dominated** {F : T → E → V} {U : Set E} (hU : IsOpen U) (hmeas : ∀ y ∈ U, AEStronglyMeasurable (fun t => F t y) μ) (hdiff : ∀ᵐ t ∂μ, DifferentiableOn ℂ (F t) U) (hdom : ∀ x ∈ U, ∃ s ∈ 𝓝 x, ∃ bound : T → ℝ, Integrable bound μ ∧ ∀ᵐ t ∂μ, ∀ y ∈ s, ‖F t y‖ ≤ bound t) : DifferentiableOn ℂ (fun y => ∫ t, F t y ∂μ) U ∧ ∀ x ∈ U, Integrable (fun t => F t x) μ ∧ Integrable (fun t => fderiv ℂ (F t) x) μ ∧ HasFDerivAt (fun y => ∫ t, F t y ∂μ) (∫ t, fderiv ℂ (F t) x ∂μ) x

theorem **differentiableOn_setIntegral_of_holomorphic_of_continuousOn** [TopologicalSpace T] [OpensMeasurableSpace T] {F : T → E → V} {U : Set E} (hU : IsOpen U) {S K : Set T} (hK : IsCompact K) (hSK : S ⊆ K) (hS : MeasurableSet S) (hfin : μ S ≠ ⊤) (hcont : ContinuousOn (Function.uncurry F) (K ×ˢ U)) (hdiff : ∀ t ∈ S, DifferentiableOn ℂ (F t) U) : DifferentiableOn ℂ (fun y => ∫ t in S, F t y ∂μ) U ∧ ∀ x ∈ U, IntegrableOn (fun t => F t x) S μ ∧ IntegrableOn (fun t => fderiv ℂ (F t) x) S μ ∧ HasFDerivAt (fun y => ∫ t in S, F t y ∂μ) (∫ t in S, fderiv ℂ (F t) x ∂μ) x

theorem **differentiableOn_integral_of_holomorphic_of_continuousOn** [TopologicalSpace T] [OpensMeasurableSpace T] [CompactSpace T] [IsFiniteMeasure μ] {F : T → E → V} {U : Set E} (hU : IsOpen U) (hcont : ContinuousOn (Function.uncurry F) (univ ×ˢ U)) (hdiff : ∀ t, DifferentiableOn ℂ (F t) U) : DifferentiableOn ℂ (fun y => ∫ t, F t y ∂μ) U ∧ ∀ x ∈ U, Integrable (fun t => F t x) μ ∧ Integrable (fun t => fderiv ℂ (F t) x) μ ∧ HasFDerivAt (fun y => ∫ t, F t y ∂μ) (∫ t, fderiv ℂ (F t) x ∂μ) x

theorem **differentiableOn_intervalIntegral_of_holomorphic_of_continuousOn** {F : ℝ → E → V} {U : Set E} (hU : IsOpen U) {a b : ℝ} (hcont : ContinuousOn (Function.uncurry F) ([[a, b]] ×ˢ U)) (hdiff : ∀ t ∈ [[a, b]], DifferentiableOn ℂ (F t) U) : DifferentiableOn ℂ (fun y => ∫ t in a..b, F t y) U ∧ ∀ x ∈ U, IntervalIntegrable (fun t => F t x) volume a b ∧ IntervalIntegrable (fun t => fderiv ℂ (F t) x) volume a b ∧ HasFDerivAt (fun y => ∫ t in a..b, F t y) (∫ t in a..b, fderiv ℂ (F t) x) x

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FiniteAndFinrankLeOfPointwise

theorem **norm_sq_le_sum_mul** {E : Type*} [SeminormedAddCommGroup E] [NormedSpace ℂ E] (b : OrthonormalBasis ι ℂ V) (l : V →ₗ[ℂ] E) (f : V) : ‖l f‖ ^ 2 ≤ (∑ i, ‖l (b i)‖ ^ 2) * ‖f‖ ^ 2

theorem **sum_norm_sq_eq** (b : OrthonormalBasis ι ℂ V) (c : OrthonormalBasis κ ℂ V) (l : V →ₗ[ℂ] E) : ∑ i, ‖l (b i)‖ ^ 2 = ∑ j, ‖l (c j)‖ ^ 2

theorem **sum_norm_sq_eq_adjoint** [FiniteDimensional ℂ V] [FiniteDimensional ℂ E] (b : OrthonormalBasis ι ℂ V) (e : OrthonormalBasis κ ℂ E) (l : V →ₗ[ℂ] E) : ∑ i, ‖l (b i)‖ ^ 2 = ∑ k, ‖LinearMap.adjoint l (e k)‖ ^ 2

theorem **adjoint_sq_sq_le** [FiniteDimensional ℂ V] [FiniteDimensional ℂ E] (l : V →ₗ[ℂ] E) {e : E} (he : ‖e‖ = 1) : ‖LinearMap.adjoint l e‖ ^ 2 ≤ ‖l (LinearMap.adjoint l e)‖

theorem **sum_norm_sq_le** [FiniteDimensional ℂ E] (b : OrthonormalBasis ι ℂ V) (l : V →ₗ[ℂ] E) {C : ℝ} (hC : 0 ≤ C) (h : ∀ f, ‖l f‖ ^ 2 ≤ C * ‖f‖ ^ 2) : ∑ i, ‖l (b i)‖ ^ 2 ≤ finrank ℂ E * C

theorem **integral_sum_eq_finrank'** {ι : Type*} [Fintype ι] (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hnorm : ∀ f, ‖f‖ ^ 2 = ∫ x, ‖l x f‖ ^ 2 ∂μ) (b : OrthonormalBasis ι ℂ V) : ∫ x, ∑ i, ‖l x (b i)‖ ^ 2 ∂μ = finrank ℂ V

theorem **finrank_le_of_finite'** [IsFiniteMeasure μ] [FiniteDimensional ℂ V] [∀ x, FiniteDimensional ℂ (E x)] {r : ℕ} (hr : ∀ x, finrank ℂ (E x) ≤ r) (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hnorm : ∀ f, ‖f‖ ^ 2 = ∫ x, ‖l x f‖ ^ 2 ∂μ) {C : ℝ} (hC : 0 ≤ C) (h : ∀ f x, ‖l x f‖ ^ 2 ≤ C * ‖f‖ ^ 2) : (finrank ℂ V : ℝ) ≤ r * C * μ.real Set.univ

theorem **finite_and_finrank_le'** [IsFiniteMeasure μ] [∀ x, FiniteDimensional ℂ (E x)] {r : ℕ} (hr : ∀ x, finrank ℂ (E x) ≤ r) (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hnorm : ∀ f, ‖f‖ ^ 2 = ∫ x, ‖l x f‖ ^ 2 ∂μ) {C : ℝ} (hC : 0 ≤ C) (h : ∀ f x, ‖l x f‖ ^ 2 ≤ C * ‖f‖ ^ 2) : FiniteDimensional ℂ V ∧ (finrank ℂ V : ℝ) ≤ r * C * μ.real Set.univ

theorem **sum_eq_finrank_of_homogeneous'** [IsProbabilityMeasure μ] {ι : Type*} [Fintype ι] (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hnorm : ∀ f, ‖f‖ ^ 2 = ∫ x, ‖l x f‖ ^ 2 ∂μ) (hhom : ∀ x y, ∃ u : V →ₗ[ℂ] V, (∀ f, ‖u f‖ = ‖f‖) ∧ ∀ f, ‖l y (u f)‖ = ‖l x f‖) (b : OrthonormalBasis ι ℂ V) (x : X) : ∑ i, ‖l x (b i)‖ ^ 2 = finrank ℂ V

theorem **integrable_inner** (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (f g : V) : Integrable (fun x => ⟪l x f, l x g⟫_ℂ) μ

theorem **integral_inner_self** (l : ∀ x, V →ₗ[ℂ] E x) (f : V) : ∫ x, ⟪l x f, l x f⟫_ℂ ∂μ = ((∫ x, ‖l x f‖ ^ 2 ∂μ : ℝ) : ℂ)

theorem **l2_inner_product_norm_sq_eq_integral** (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hpos : ∀ f, ∫ x, ‖l x f‖ ^ 2 ∂μ = 0 → f = 0) (f : V) : letI

theorem **exists_orthonormal_basis** [FiniteDimensional ℂ V] (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hpos : ∀ f, ∫ x, ‖l x f‖ ^ 2 ∂μ = 0 → f = 0) : ∃ b : Basis (Fin (finrank ℂ V)) ℂ V, ∀ i j, ∫ x, ⟪l x (b i), l x (b j)⟫_ℂ ∂μ = if i = j then 1 else 0

theorem **norm_sq_le_bergman_mul** (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hpos : ∀ f, ∫ x, ‖l x f‖ ^ 2 ∂μ = 0 → f = 0) (b : Basis ι ℂ V) (hb : ∀ i j, ∫ x, ⟪l x (b i), l x (b j)⟫_ℂ ∂μ = if i = j then 1 else 0) (f : V) (x : X) : ‖l x f‖ ^ 2 ≤ (∑ i, ‖l x (b i)‖ ^ 2) * ∫ y, ‖l y f‖ ^ 2 ∂μ

theorem **finite_and_finrank_le_mul** [IsFiniteMeasure μ] [∀ x, FiniteDimensional ℂ (E x)] {r : ℕ} (hr : ∀ x, finrank ℂ (E x) ≤ r) (l : ∀ x, V →ₗ[ℂ] E x) (hmeas : ∀ f, AEStronglyMeasurable (fun x => ‖l x f‖) μ) (hsep : ∀ f, (∀ x, l x f = 0) → f = 0) {C : ℝ} (hC : 0 ≤ C) (h : ∀ f x, ‖l x f‖ ^ 2 ≤ C * ∫ y, ‖l y f‖ ^ 2 ∂μ) : FiniteDimensional ℂ V ∧ (finrank ℂ V : ℝ) ≤ r * C * μ.real Set.univ

theorem **finite_and_finrank_le** [IsProbabilityMeasure μ] [∀ x, FiniteDimensional ℂ (E x)] (hE : ∀ x, finrank ℂ (E x) ≤ 1) (l : ∀ x, V →ₗ[ℂ] E x) (hmeas : ∀ f, AEStronglyMeasurable (fun x => ‖l x f‖) μ) (hsep : ∀ f, (∀ x, l x f = 0) → f = 0) {C : ℝ} (hC : 0 ≤ C) (h : ∀ f x, ‖l x f‖ ^ 2 ≤ C * ∫ y, ‖l y f‖ ^ 2 ∂μ) : FiniteDimensional ℂ V ∧ (finrank ℂ V : ℝ) ≤ C

theorem **bergman_eq_finrank_of_transitive** (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hpos : ∀ f, ∫ x, ‖l x f‖ ^ 2 ∂μ = 0 → f = 0) (ρ : G → V →ₗ[ℂ] V) (hρ : ∀ g x f, ‖l (g • x) (ρ g f)‖ = ‖l x f‖) (b : Basis ι ℂ V) (hb : ∀ i j, ∫ x, ⟪l x (b i), l x (b j)⟫_ℂ ∂μ = if i = j then 1 else 0) (x : X) : ∑ i, ‖l x (b i)‖ ^ 2 = finrank ℂ V

theorem **norm_sq_le_finrank_mul** [FiniteDimensional ℂ V] (l : ∀ x, V →ₗ[ℂ] E x) (hint : ∀ f, Integrable (fun x => ‖l x f‖ ^ 2) μ) (hpos : ∀ f, ∫ x, ‖l x f‖ ^ 2 ∂μ = 0 → f = 0) (ρ : G → V →ₗ[ℂ] V) (hρ : ∀ g x f, ‖l (g • x) (ρ g f)‖ = ‖l x f‖) (f : V) (x : X) : ‖l x f‖ ^ 2 ≤ finrank ℂ V * ∫ y, ‖l y f‖ ^ 2 ∂μ

theorem **finite_and_finrank_le_of_pointwise** (p : X → V → ℝ) (hp : ∀ x, ∃ l : V →ₗ[ℂ] ℂ, ∀ f, p x f = ‖l f‖) (hmeas : ∀ f, AEStronglyMeasurable (fun x => p x f) μ) (hsep : ∀ f, (∀ x, p x f = 0) → f = 0) {C : ℝ} (hC : 0 ≤ C) (h : ∀ f x, p x f ^ 2 ≤ C * ∫ y, p y f ^ 2 ∂μ) : FiniteDimensional ℂ V ∧ (finrank ℂ V : ℝ) ≤ C

theorem **norm_sq_le_finrank_mul_fun** {G : Type*} [Group G] [MulAction G X] [MulAction.IsPretransitive G X] [MeasurableConstSMul G X] [SMulInvariantMeasure G X μ] (V : Submodule ℂ (X → ℂ)) [FiniteDimensional ℂ V] (hint : ∀ f ∈ V, Integrable (fun x => ‖f x‖ ^ 2) μ) (hpos : ∀ f ∈ V, ∫ x, ‖f x‖ ^ 2 ∂μ = 0 → f = 0) (hG : ∀ g : G, ∀ f ∈ V, (fun x => f (g • x)) ∈ V) {f : X → ℂ} (hf : f ∈ V) (x : X) : ‖f x‖ ^ 2 ≤ finrank ℂ V * ∫ y, ‖f y‖ ^ 2 ∂μ

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FiniteAndNcardSetOfSubmoduleLeRelIndexEqLe

theorem **exists_ker_eq_of_index_prime** {p : ℕ} (hp : p.Prime) (H : AddSubgroup M) (hH : H.index = p) : ∃ f : M →ₗ[ℤ] ZMod p, f.toAddMonoidHom.ker = H

theorem **encard_setOf_index_prime_le** [Module.Free ℤ M] [Module.Finite ℤ M] {p : ℕ} (hp : p.Prime) : {H : AddSubgroup M | H.index = p}.encard ≤ ((p ^ finrank ℤ M : ℕ) : ℕ∞)

theorem **exists_le_relIndex_eq_prime** {p : ℕ} (hp : p.Prime) (H : AddSubgroup M) (h0 : H.index ≠ 0) (hd : p ∣ H.index) : ∃ K : AddSubgroup M, H ≤ K ∧ H.relIndex K = p

theorem **encard_biUnion_le_mul** {α β : Type*} (s : Set α) (t : α → Set β) (a b : ℕ) (hs : s.encard ≤ a) (ht : ∀ i ∈ s, (t i).encard ≤ b) : (⋃ i ∈ s, t i).encard ≤ ((a * b : ℕ) : ℕ∞)

theorem **setOf_le_relIndex_eq_image** (K : AddSubgroup M) (m : ℕ) : {H : AddSubgroup M | H ≤ K ∧ H.relIndex K = m} = (fun H' : AddSubgroup K => H'.map K.subtype) '' {H' | H'.index = m}

theorem **encard_setOf_index_eq_le** [Module.Free ℤ M] [Module.Finite ℤ M] {m : ℕ} (hm : m ≠ 0) : {H : AddSubgroup M | H.index = m}.encard ≤ ((m ^ finrank ℤ M : ℕ) : ℕ∞)

theorem **finite_setOf_index_eq** [Module.Free ℤ M] [Module.Finite ℤ M] {m : ℕ} (hm : m ≠ 0) : {H : AddSubgroup M | H.index = m}.Finite

theorem **ncard_setOf_index_eq_le** [Module.Free ℤ M] [Module.Finite ℤ M] {m : ℕ} (hm : m ≠ 0) : {H : AddSubgroup M | H.index = m}.ncard ≤ m ^ finrank ℤ M

theorem **finite_and_ncard_setOf_index_pi_le** (ρ : ℕ) {m : ℕ} (hm : m ≠ 0) : {H : AddSubgroup (Fin ρ → ℤ) | H.index = m}.Finite ∧ {H : AddSubgroup (Fin ρ → ℤ) | H.index = m}.ncard ≤ m ^ ρ

theorem **encard_setOf_le_relIndex_eq_le** (K : AddSubgroup M) [Module.Free ℤ K] [Module.Finite ℤ K] {m : ℕ} (hm : m ≠ 0) : {H : AddSubgroup M | H ≤ K ∧ H.relIndex K = m}.encard ≤ ((m ^ finrank ℤ K : ℕ) : ℕ∞)

theorem **encard_setOf_submodule_le_relIndex_eq_le** {R : Type*} [Ring R] [Module R M] (L : Submodule R M) [Module.Free ℤ L] [Module.Finite ℤ L] {m : ℕ} (hm : m ≠ 0) : {H : Submodule R M | H ≤ L ∧ H.toAddSubgroup.relIndex L.toAddSubgroup = m}.encard ≤ ((m ^ finrank ℤ L : ℕ) : ℕ∞)

theorem **finite_and_ncard_setOf_submodule_le_relIndex_eq_le** {R : Type*} [Ring R] [Module R M] (L : Submodule R M) [Module.Free ℤ L] [Module.Finite ℤ L] {m : ℕ} (hm : m ≠ 0) : {H : Submodule R M | H ≤ L ∧ H.toAddSubgroup.relIndex L.toAddSubgroup = m}.Finite ∧ {H : Submodule R M | H ≤ L ∧ H.toAddSubgroup.relIndex L.toAddSubgroup = m}.ncard ≤ m ^ finrank ℤ L

theorem **card_le_of_family_separated** {V ι : Type*} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] (s : Finset ι) (f : ι → V) {d R : ℝ} (hd : 0 < d) (hR : 0 ≤ R) (hs : ∀ i ∈ s, ‖f i‖ ≤ R) (h : ∀ i ∈ s, ∀ j ∈ s, i ≠ j → d ≤ ‖f i - f j‖) : (s.card : ℝ) ≤

(1 + 2 * R / d) ^ finrank ℝ V

theorem **finite_and_ncard_le_of_forall_finset** {α : Type*} (S : Set α) (B : ℝ) (h : ∀ s : Finset α, (s : Set α) ⊆ S → (s.card : ℝ) ≤ B) : S.Finite ∧ (S.ncard : ℝ) ≤ B

theorem **exists_zsmul_mem_span_of_linearIndependent** {A : Type*} [AddCommGroup A] [Module.Finite ℤ A] {n : ℕ} (hn : finrank ℤ A = n) {v : Fin n → A} (hv : LinearIndependent ℤ v) (x : A) : ∃ k : ℤ, k ≠ 0 ∧ k • x ∈ Submodule.span ℤ (Set.range v)

theorem **linearMap_ext_of_linearIndependent** {A B : Type*} [AddCommGroup A] [AddCommGroup B] [Module.Finite ℤ A] [Module.IsTorsionFree ℤ B] {n : ℕ} (hn : finrank ℤ A = n) {v : Fin n → A} (hv : LinearIndependent ℤ v) {f g : A →ₗ[ℤ] B} (h : ∀ i, f (v i) = g (v i)) : f = g

theorem **finite_and_ncard_setOf_norm_le_of_rank** (L : Submodule ℤ E) [Module.Finite ℤ L] {n : ℕ} (hn : finrank ℤ L = n) {v : Fin n → L} (hv : LinearIndependent ℤ v) {d R : ℝ} (hd : 0 < d) (hR : 0 ≤ R) (hL : ∀ x ∈ L, x ≠ 0 → d ≤ ‖x‖) : {x : L | ‖(x : E)‖ ≤ R}.Finite ∧ (({x : L | ‖(x : E)‖ ≤ R}.ncard : ℕ) : ℝ) ≤ (1 + 2 * R / d) ^ n

theorem **finite_and_ncard_le_of_norm_apply_le** (L : Submodule ℤ E) [Module.Finite ℤ L] {n : ℕ} (hn : finrank ℤ L = n) {v : Fin n → L} (hv : LinearIndependent ℤ v) {d R : ℝ} (hd : 0 < d) (hR : 0 ≤ R) (hL : ∀ x ∈ L, x ≠ 0 → d ≤ ‖x‖) {Γ : Type*} (G : Set Γ) (φ : Γ → L →ₗ[ℤ] L) (hφ : Set.InjOn φ G) (hG : ∀ g ∈ G, ∀ i, ‖(φ g (v i) : E)‖ ≤ R) : G.Finite ∧ (G.ncard : ℝ) ≤ (1 + 2 * R / d) ^ (n ^ 2)

theorem **finite_and_ncard_le_of_norm_map_le** (L : Submodule ℤ E) [Module.Finite ℤ L] {n : ℕ} (hn : finrank ℤ L = n) {v : Fin n → L} (hv : LinearIndependent ℤ v) {lam1 lamN c : ℝ} (h1 : 0 < lam1) (hc : 0 ≤ c) (hL : ∀ x ∈ L, x ≠ 0 → lam1 ≤ ‖x‖) (hvN : ∀ i, ‖(v i : E)‖ ≤ lamN) {Γ : Type*} (G : Set Γ) (φ : Γ → L →ₗ[ℤ] L) (hφ : Set.InjOn φ G) (hG : ∀ g ∈ G, ∀ x : L, ‖(φ g x : E)‖ ≤ c * ‖(x : E)‖) : G.Finite ∧ (G.ncard : ℝ) ≤ (1 + 2 * c * lamN / lam1) ^ (n ^ 2)

theorem **finite_and_ncard_linearEquiv_le** (L : Submodule ℤ E) [Module.Finite ℤ L] {n : ℕ} (hn : finrank ℤ L = n) {v : Fin n → L} (hv : LinearIndependent ℤ v) {lam1 lamN c : ℝ} (h1 : 0 < lam1) (hc : 0 ≤ c) (hL : ∀ x ∈ L, x ≠ 0 → lam1 ≤ ‖x‖) (hvN : ∀ i, ‖(v i : E)‖ ≤ lamN) (G : Set (L ≃ₗ[ℤ] L)) (hG : ∀ g ∈ G, ∀ x : L, ‖(g x : E)‖ ≤ c * ‖(x : E)‖) : G.Finite ∧ (G.ncard : ℝ) ≤ (1 + 2 * c * lamN / lam1) ^ (n ^ 2)

theorem **linearIndependent_int_of_real** (L : Submodule ℤ E) {ι : Type*} {v : ι → L} (hv : LinearIndependent ℝ fun i => (v i : E)) : LinearIndependent ℤ v

theorem **ZLattice_finite_and_ncard_le_of_norm_map_le** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] {n : ℕ} (hn : finrank ℝ E = n) {v : Fin n → L} (hv : LinearIndependent ℝ fun i => (v i : E)) {lam1 lamN c : ℝ} (h1 : 0 < lam1) (hc : 0 ≤ c) (hL : ∀ x ∈ L, x ≠ 0 → lam1 ≤ ‖x‖) (hvN : ∀ i, ‖(v i : E)‖ ≤ lamN) {Γ : Type*} (G : Set Γ) (φ : Γ → L →ₗ[ℤ] L) (hφ : Set.InjOn φ G) (hG : ∀ g ∈ G, ∀ x : L, ‖(φ g x : E)‖ ≤ c * ‖(x : E)‖) : G.Finite ∧ (G.ncard : ℝ) ≤ (1 + 2 * c * lamN / lam1) ^ (n ^ 2)

theorem **ZLattice_finite_and_ncard_le_of_mapsTo** [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] {n : ℕ} (hn : finrank ℝ E = n) {v : Fin n → E} (hvL : ∀ i, v i ∈ L) (hv : LinearIndependent ℝ v) {lam1 lamN c : ℝ} (h1 : 0 < lam1) (hc : 0 ≤ c) (hL : ∀ x ∈ L, x ≠ 0 → lam1 ≤ ‖x‖) (hvN : ∀ i, ‖v i‖ ≤ lamN) {Γ : Type*} (G : Set Γ) (φ : Γ → E →ₗ[ℝ] E) (hφ : Set.InjOn φ G) (hφL : ∀ g ∈ G, ∀ x ∈ L, φ g x ∈ L) (hG : ∀ g ∈ G, ∀ x ∈ L, ‖φ g x‖ ≤ c * ‖x‖) : G.Finite ∧ (G.ncard : ℝ) ≤ (1 + 2 * c * lamN / lam1) ^ (n ^ 2)

theorem **card_subgroups_le_and_card_automorphisms_le** : (∀ ρ m : ℕ, 1 ≤ m → {H : AddSubgroup (Fin ρ → ℤ) | H.index = m}.Finite ∧ {H : AddSubgroup (Fin ρ → ℤ) | H.index = m}.ncard ≤ m ^ ρ) ∧ (∀ (E : Type) [NormedAddCommGroup E] [InnerProductSpace ℝ E] [FiniteDimensional ℝ E] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] (n : ℕ), finrank ℤ L = n → ∀ (lam1 lamN c : ℝ), 0 < lam1 → 1 ≤ c → (∀ x ∈ L, x ≠ 0 → lam1 ≤ ‖x‖) → ∀ v : Fin n → L, LinearIndependent ℤ v → (∀ i, ‖(v i : E)‖ ≤ lamN) → ∀ G : Set (L ≃ₗ[ℤ] L), (∀ g ∈ G, ∀ x : L, ‖(g x : E)‖ ≤ c * ‖(x : E)‖) → G.Finite ∧ (G.ncard : ℝ) ≤ (1 + 2 * c * lamN / lam1) ^ (n ^ 2))

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FiniteInterClosedBallOfSeparated

theorem **finset_card_le_of_separated** (s : Finset E) {d R : ℝ} (hd : 0 < d) (hR : 0 ≤ R) (x : E) (hs : ∀ a ∈ s, dist a x ≤ R) (hsep : (s : Set E).Pairwise fun a b => d ≤ dist a b) : (s.card : ℝ) ≤ (1 + 2 * R / d) ^ finrank ℝ E

theorem **one_add_two_mul_div_pow_le** {d R : ℝ} (hd : 0 < d) (hR : d ≤ R) (n : ℕ) : (1 + 2 * R / d) ^ n ≤ (3 * R / d) ^ n

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FinrankMulDiscrRpowLeSumSqOfMemTraceDual

theorem **card_mul_rpow_prod_le_sum** {ι : Type*} [Fintype ι] [Nonempty ι] (z : ι → ℝ) (hz : ∀ i, 0 ≤ z i) : (Fintype.card ι : ℝ) * (∏ i, z i) ^ ((Fintype.card ι : ℝ)⁻¹) ≤ ∑ i, z i

theorem **norm_eq_prod_complexEmbeddings** (x : K) : ((Algebra.norm ℚ x : ℚ) : ℂ) = ∏ σ : K →+* ℂ, σ x

theorem **trace_eq_sum_complexEmbeddings** (x : K) : ((Algebra.trace ℚ K x : ℚ) : ℂ) = ∑ σ : K →+* ℂ, σ x

theorem **abs_norm_eq_prod_complexEmbeddings** (x : K) : |((Algebra.norm ℚ x : ℚ) : ℝ)| = ∏ σ : K →+* ℂ, ‖σ x‖

theorem **finrank_mul_abs_norm_rpow_le_sum** (x : K) : (finrank ℚ K : ℝ) * |((Algebra.norm ℚ x : ℚ) : ℝ)| ^ ((finrank ℚ K : ℝ)⁻¹) ≤ ∑ σ : K →+* ℂ, ‖σ x‖

theorem **finrank_mul_abs_norm_rpow_le_sum_sq** (x : K) : (finrank ℚ K : ℝ) * |((Algebra.norm ℚ x : ℚ) : ℝ)| ^ (2 / (finrank ℚ K : ℝ)) ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2

theorem **one_le_abs_norm_of_isIntegral** {x : K} (hx : IsIntegral ℤ x) (h0 : x ≠ 0) : 1 ≤ |Algebra.norm ℚ x|

theorem **finrank_le_sum_norm_of_isIntegral** {x : K} (hx : IsIntegral ℤ x) (h0 : x ≠ 0) : (finrank ℚ K : ℝ) ≤ ∑ σ : K →+* ℂ, ‖σ x‖

theorem **eq_zero_of_isIntegral_of_sum_norm_lt** {x : K} (hx : IsIntegral ℤ x) (h : ∑ σ : K →+* ℂ, ‖σ x‖ < finrank ℚ K) : x = 0

theorem **absNorm_le_abs_norm_of_mem** {J : FractionalIdeal (𝒪 K)⁰ K} {x : K} (hx : x ∈ J) (h0 : x ≠ 0) : FractionalIdeal.absNorm J ≤ |Algebra.norm ℚ x|

theorem **finrank_mul_absNorm_rpow_le_sum_sq** {J : FractionalIdeal (𝒪 K)⁰ K} {x : K} (hx : x ∈ J) (h0 : x ≠ 0) : (finrank ℚ K : ℝ) * ((FractionalIdeal.absNorm J : ℚ) : ℝ) ^ (2 / (finrank ℚ K : ℝ)) ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2

theorem **dual_one_eq_inv_differentIdeal** : FractionalIdeal.dual ℤ ℚ (1 : FractionalIdeal (𝒪 K)⁰ K) = (differentIdeal ℤ (𝒪 K) : FractionalIdeal (𝒪 K)⁰ K)⁻¹

theorem **absNorm_inv_differentIdeal** : FractionalIdeal.absNorm (differentIdeal ℤ (𝒪 K) : FractionalIdeal (𝒪 K)⁰ K)⁻¹ = |(discr K : ℚ)|⁻¹

theorem **finrank_mul_discr_rpow_le_sum_sq** {x : K} (hx : x ∈ (differentIdeal ℤ (𝒪 K) : FractionalIdeal (𝒪 K)⁰ K)⁻¹) (h0 : x ≠ 0) : (finrank ℚ K : ℝ) * |(discr K : ℝ)| ^ (-2 / (finrank ℚ K : ℝ)) ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2

theorem **finrank_div_rootDiscr_sq_le_sum_sq** {x : K} (hx : x ∈ (differentIdeal ℤ (𝒪 K) : FractionalIdeal (𝒪 K)⁰ K)⁻¹) (h0 : x ≠ 0) : (finrank ℚ K : ℝ) / rootDiscr K ^ 2 ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2

theorem **finrank_mul_discr_rpow_le_sum_sq_of_mem_traceDual** {x : K} (hx : x ∈ Submodule.traceDual ℤ ℚ (1 : Submodule (𝒪 K) K)) (h0 : x ≠ 0) : (finrank ℚ K : ℝ) * |(discr K : ℝ)| ^ (-2 / (finrank ℚ K : ℝ)) ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2

theorem **eq_zero_of_embedding_eq_zero** {A : Type*} [Ring A] [Nontrivial A] (σ : K →+* A) {x : K} (h : σ x = 0) : x = 0

theorem **exists_int_eq_mul_trace_of_mem_traceDual** {q : ℚ} {x : K} (hx : x ∈ Submodule.traceDual ℤ ℚ (Submodule.span (𝒪 K) {(q : K)})) : ∃ m : ℤ, q * Algebra.trace ℚ K x = m

theorem **trace_mem_of_mem_traceDual** {q : ℚ} (hq : q ≠ 0) {x : K} (hx : x ∈ Submodule.traceDual ℤ ℚ (Submodule.span (𝒪 K) {(q : K)})) : ∃ m : ℤ, Algebra.trace ℚ K x = q⁻¹ * m

theorem **trace_mem_of_mem_dual_spanSingleton** {q : ℚ} (hq : q ≠ 0) {x : K} (hx : x ∈ FractionalIdeal.dual ℤ ℚ (FractionalIdeal.spanSingleton (𝒪 K)⁰ (q : K))) : ∃ m : ℤ, Algebra.trace ℚ K x = q⁻¹ * m

theorem **bijective_ofRealHom_comp** : Function.Bijective (fun σ : F →+* ℝ => Complex.ofRealHom.comp σ)

theorem **card_realEmbeddings** : Fintype.card (F →+* ℝ) = finrank ℚ F

theorem **norm_eq_prod_realEmbeddings** (x : F) : ((Algebra.norm ℚ x : ℚ) : ℝ) = ∏ σ : F →+* ℝ, σ x

theorem **trace_eq_sum_realEmbeddings** (x : F) : ((Algebra.trace ℚ F x : ℚ) : ℝ) = ∑ σ : F →+* ℝ, σ x

theorem **sum_abs_realEmbeddings** (x : F) : ∑ σ : F →+* ℝ, |σ x| = ∑ σ : F →+* ℂ, ‖σ x‖

theorem **norm_pos_of_totallyPos** {a : F} (ha : ∀ σ : F →+* ℝ, 0 < σ a) : 0 < Algebra.norm ℚ a

theorem **ne_zero_of_totallyPos** {a : F} (ha : ∀ σ : F →+* ℝ, 0 < σ a) : a ≠ 0

theorem **finrank_mul_norm_rpow_le_trace** {a : F} (ha : ∀ σ : F →+* ℝ, 0 < σ a) : (finrank ℚ F : ℝ) * ((Algebra.norm ℚ a : ℚ) : ℝ) ^ (1 / (finrank ℚ F : ℝ)) ≤ ((Algebra.trace ℚ F a : ℚ) : ℝ)

theorem **finrank_le_trace_of_isIntegral** {a : F} (hi : IsIntegral ℤ a) (ha : ∀ σ : F →+* ℝ, 0 < σ a) : (finrank ℚ F : ℚ) ≤ Algebra.trace ℚ F a

theorem **eq_zero_of_isIntegral_of_sum_abs_lt** {x : F} (hx : IsIntegral ℤ x) (h : ∑ σ : F →+* ℝ, |σ x| < finrank ℚ F) : x = 0

theorem **eq_zero_or_totallyPos** {a : F} (ha : ∀ σ : F →+* ℝ, 0 ≤ σ a) : a = 0 ∨ ∀ σ : F →+* ℝ, 0 < σ a

theorem **trace_lower_bounds** (F K : Type*) [Field F] [NumberField F] [IsTotallyReal F] [Field K] [NumberField K] : (∀ a : F, (∀ σ : F →+* ℝ, 0 < σ a) → (finrank ℚ F : ℝ) * ((Algebra.norm ℚ a : ℚ) : ℝ) ^ (1 / (finrank ℚ F : ℝ)) ≤ ((Algebra.trace ℚ F a : ℚ) : ℝ)) ∧ (∀ a : F, IsIntegral ℤ a → (∀ σ : F →+* ℝ, 0 < σ a) → (finrank ℚ F : ℚ) ≤ Algebra.trace ℚ F a) ∧ (∀ a : K, IsIntegral ℤ a → ∑ σ : K →+* ℂ, ‖σ a‖ < finrank ℚ K → a = 0) ∧ (∀ x ∈ (differentIdeal ℤ (𝒪 K) : FractionalIdeal (𝒪 K)⁰ K)⁻¹, x ≠ 0 → (finrank ℚ K : ℝ) * |(discr K : ℝ)| ^ (-2 / (finrank ℚ K : ℝ)) ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2) ∧ (∀ (J : FractionalIdeal (𝒪 K)⁰ K), ∀ x ∈ J, x ≠ 0 → (finrank ℚ K : ℝ) * ((FractionalIdeal.absNorm J : ℚ) : ℝ) ^ (2 / (finrank ℚ K : ℝ)) ≤ ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2) ∧ (∀ (a : F) (σ : F →+* ℝ), σ a = 0 → a = 0) ∧ (∀ q : ℚ, q ≠ 0 → ∀ x ∈ Submodule.traceDual ℤ ℚ (Submodule.span (𝒪 F) {(q : F)}), ∃ m : ℤ, Alg

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FormalExpMulFormalExpNeg

private theorem **hasSubst_sub_one** {g : MvPowerSeries σ A} (hg : constantCoeff g = 1) : PowerSeries.HasSubst (g - 1)

theorem **constantCoeff_formalExp** {f : MvPowerSeries σ A} (hf : constantCoeff f = 0) : constantCoeff (formalExp f) = 1

theorem **constantCoeff_formalLog** {g : MvPowerSeries σ A} (hg : constantCoeff g = 1) : constantCoeff (formalLog g) = 0

theorem **isUnit_formalExp** {f : MvPowerSeries σ A} (hf : constantCoeff f = 0) : IsUnit (formalExp f)

theorem **pderiv_formalExp** {f : MvPowerSeries σ A} (hf : constantCoeff f = 0) (i : σ) : pderiv A i (formalExp f) = formalExp f * pderiv A i f

private theorem **one_add_X_mul_deriv_log** : (1 + PowerSeries.X) * PowerSeries.derivative A (PowerSeries.log A) = 1

theorem **mul_pderiv_formalLog** {g : MvPowerSeries σ A} (hg : constantCoeff g = 1) (i : σ) : g * pderiv A i (formalLog g) = pderiv A i g

theorem **eq_formalExp_of_pderiv** {f u : MvPowerSeries σ A} (hf : constantCoeff f = 0) (hu : constantCoeff u = 1) (h : ∀ i, pderiv A i u = u * pderiv A i f) : u = formalExp f

theorem **eq_formalLog_of_pderiv** {g u : MvPowerSeries σ A} (hg : constantCoeff g = 1) (hu : constantCoeff u = 0) (h : ∀ i, g * pderiv A i u = pderiv A i g) : u = formalLog g

@[simp] theorem **formalExp_zero** : formalExp (0 : MvPowerSeries σ A) = 1

@[simp] theorem **formalLog_one** : formalLog (1 : MvPowerSeries σ A) = 0

theorem **formalExp_add** {f g : MvPowerSeries σ A} (hf : constantCoeff f = 0) (hg : constantCoeff g = 0) : formalExp (f + g) = formalExp f * formalExp g

theorem **formalExp_mul_formalExp_neg** {f : MvPowerSeries σ A} (hf : constantCoeff f = 0) : formalExp f * formalExp (-f) = 1

theorem **formalLog_mul** {f g : MvPowerSeries σ A} (hf : constantCoeff f = 1) (hg : constantCoeff g = 1) : formalLog (f * g) = formalLog f + formalLog g

theorem **formalLog_prod** {κ : Type*} (s : Finset κ) {g : κ → MvPowerSeries σ A} (hg : ∀ k ∈ s, constantCoeff (g k) = 1) : formalLog (∏ k ∈ s, g k) = ∑ k ∈ s, formalLog (g k)

theorem **formalLog_pow** (n : ℕ) {g : MvPowerSeries σ A} (hg : constantCoeff g = 1) : formalLog (g ^ n) = n • formalLog g

theorem **subst_formalExp** {a : σ → MvPowerSeries τ A} (ha : HasSubst a) {f : MvPowerSeries σ A} (hf : constantCoeff f = 0) : subst a (formalExp f) = formalExp (subst a f)

theorem **subst_formalLog** {a : σ → MvPowerSeries τ A} (ha : HasSubst a) {g : MvPowerSeries σ A} (hg : constantCoeff g = 1) : subst a (formalLog g) = formalLog (subst a g)

theorem **map_formalLog** (φ : A →+* B) {g : MvPowerSeries σ A} (hg : constantCoeff g = 1) : map φ (formalLog g) = formalLog (map φ g)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FormalLogEqSumOfMultiplicativeParam

theorem **subst_subst** [Finite ι] {a : ι → MvPowerSeries κ A} {b : κ → MvPowerSeries μ A} (hb : HasSubst b) (ha : ∀ i, constantCoeff (a i) = 0) (f : MvPowerSeries ι A) : subst b (subst a f) = subst (fun i => subst b (a i)) f

theorem **subst_sumElim** (b : κ → MvPowerSeries μ A) (u : σ → MvPowerSeries κ A) (t : τ → MvPowerSeries κ A) : (fun i => subst b (Sum.elim u t i)) = Sum.elim (fun i => subst b (u i)) fun j => subst b (t j)

theorem **subst_elim_X** (f : MvPowerSeries (σ ⊕ τ) A) : subst (Sum.elim (fun i => X (inl i)) fun j => X (inr j) : σ ⊕ τ → MvPowerSeries (σ ⊕ τ) A) f = f

theorem **subst_elim_elim_X** (f : MvPowerSeries (σ ⊕ τ ⊕ ι) A) : subst (Sum.elim (fun i => X (inl i)) (Sum.elim (fun j => X (inr (inl j))) fun j => X (inr (inr j))) : σ ⊕ τ ⊕ ι → MvPowerSeries (σ ⊕ τ ⊕ ι) A) f = f

theorem **pderiv_subst'** [Fintype ι] {a : ι → MvPowerSeries κ A} (ha : ∀ i, constantCoeff (a i) = 0) (f : MvPowerSeries ι A) (j : κ) : pderiv A j (subst a f) = ∑ i, subst a (pderiv A i f) * pderiv A j (a i)

theorem **pderiv_X_ite** [DecidableEq κ] (i j : κ) : pderiv A i (X j : MvPowerSeries κ A) = if j = i then 1 else 0

theorem **coeff_subst_kill** (e : ι → κ) {b : κ → MvPowerSeries κ A} (hb0 : ∀ i, b (e i) = 0) (hb1 : ∀ k, k ∉ Set.range e → b k = X k) (f : MvPowerSeries κ A) (α : κ →₀ ℕ) : coeff α (subst b f) = if ∀ i, α (e i) = 0 then coeff α f else 0

theorem **ext_of_pderiv_of_subst** [IsAddTorsionFree A] (e : ι → κ) {b : κ → MvPowerSeries κ A} (hb0 : ∀ i, b (e i) = 0) (hb1 : ∀ k, k ∉ Set.range e → b k = X k) {f g : MvPowerSeries κ A} (hD : ∀ i, pderiv A (e i) f = pderiv A (e i) g) (h0 : subst b f = subst b g) : f = g

theorem **subst_elim_zero_eq_zero_iff** [Finite σ] [Finite τ] (f : MvPowerSeries (σ ⊕ τ) A) : subst (Sum.elim (fun _ => 0) X : σ ⊕ τ → MvPowerSeries τ A) f = 0 ↔ ∀ α : σ ⊕ τ →₀ ℕ, (∀ i, α (inl i) = 0) → coeff α f = 0

theorem **existsUnique_primitive_sum** [Finite σ] [Finite τ] [Algebra ℚ A] (a : σ → MvPowerSeries (σ ⊕ τ) A) (ha : ∀ i j, pderiv A (inl i) (a j) = pderiv A (inl j) (a i)) : ∃! f : MvPowerSeries (σ ⊕ τ) A, subst (Sum.elim (fun _ => 0) X : σ ⊕ τ → MvPowerSeries τ A) f = 0 ∧ ∀ i, pderiv A (inl i) f = a i

theorem **eq_sum_of_additive_param** [Fintype σ] [Finite τ] {f : MvPowerSeries (σ ⊕ τ) A} (h : subst (Sum.elim (fun i => X (inl i) + X (inr (inl i))) fun j => X (inr (inr j)) : σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ τ) A) f = subst (Sum.elim (fun i => X (inl i)) fun j => X (inr (inr j)) : σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ τ) A) f + subst (Sum.elim (fun i => X (inr (inl i))) fun j => X (inr (inr j)) : σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ τ) A) f) : f = ∑ i, X (inl i) * subst (fun j => X (inr j) : τ → MvPowerSeries (σ ⊕ τ) A) (subst (Sum.elim (fun _ => 0) X : σ ⊕ τ → MvPowerSeries τ A) (pderiv A (inl i) f))

theorem **existsUnique_coboundary_param** [Finite σ] [Finite τ] {B : MvPowerSeries (σ ⊕ σ ⊕ τ) A} (h0 : subst (Sum.elim (fun _ => 0) (Sum.elim (fun _ => 0) X) : σ ⊕ σ ⊕ τ → MvPowerSeries τ A) B = 0) (hsymm : subst (Sum.elim (fun i => X (inr (inl i))) (Sum.elim (fun i => X (inl i)) fun j => X (inr (inr j))) : σ ⊕ σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ τ) A) B = B) (hcoc : subst (Sum.elim (fun i => X (inl i)) (Sum.elim (fun i => X (inr (inl i))) fun j => X (inr (inr (inr j)))) : σ ⊕ σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ σ ⊕ τ) A) B + subst (Sum.elim (fun i => X (inl i) + X (inr (inl i))) (Sum.elim (fun i => X (inr (inr (inl i)))) fun j => X (inr (inr (inr j)))) : σ ⊕ σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ σ ⊕ τ) A) B = subst (Sum.elim (fun i => X (inr (inl i))) (Sum.elim (fun i => X (inr (inr (inl i)))) fun j => X (inr (inr (inr j)))) : σ ⊕ σ ⊕ τ → MvPowerSeries (σ ⊕ σ ⊕ σ ⊕ τ) A) B + subst (Sum.elim (fun i => X (in

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FrameIsotropicOrthonormal

theorem **entry** (k r : ℕ) (M : Matrix (Fin (2 * r + k + 1)) (Fin (2 * r + k + 1)) ℂ) (a b : Fin (r + 1)) : ((frame k r)ᴴ * M * frame k r) a b = star (fvec k r (isoVec k r) a) ⬝ᵥ (M *ᵥ fvec k r (isoVec k r) b)

theorem **bvec_pair** {n : ℕ} (M : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : star (bvec i) ⬝ᵥ (M *ᵥ bvec j) = M i j

theorem **fvec_castSucc** (k r : ℕ) (z : Fin (2 * r + k + 1) → ℂ) (a : Fin r) : fvec k r z a.castSucc = bvec (eI k r a)

theorem **fvec_last** (k r : ℕ) (z : Fin (2 * r + k + 1) → ℂ) : fvec k r z (Fin.last r) = z

theorem **half** : (((Real.sqrt 2)⁻¹ : ℝ) : ℂ) * ((((Real.sqrt 2)⁻¹ : ℝ) : ℂ) * 2) = 1

theorem **star_isoVec** (k r : ℕ) : star (isoVec k r) = (((Real.sqrt 2)⁻¹ : ℝ) : ℂ) • (star (bvec (wI k r 0)) + star (bvec (wI k r (Fin.last k))))

theorem **frame_orthonormal_isotropic** : ∀ k r : ℕ, 1 ≤ k → (frame k r)ᴴ * frame k r = 1 ∧ (frame k r)ᴴ * Hpk k r * frame k r = 0

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.FrameIsotropicOrthonormalStatement

theorem **_root_.GlobalTheory.T.frame_isotropic_orthonormal** : ∀ k r : ℕ, 1 ≤ k → (frame k r)ᴴ * frame k r = 1 ∧ (frame k r)ᴴ * Hpk k r * frame k r = 0

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.GaussianSumOfOneSeparatedOfLe

theorem **finset_sum_exp_neg_le_of_card_le** {ι : Type*} (s : Finset ι) (f : ι → ℝ) {t C : ℝ} (hf : ∀ i ∈ s, t ≤ f i) (hc : ∀ u, t ≤ u → ((s.filter fun i => f i ≤ u).card : ℝ) ≤ C * exp ((u - t) / 2)) : ∑ i ∈ s, exp (-f i) ≤ 2 * C * exp (-t)

theorem **summable_exp_neg_of_card_le** {ι : Type*} (f : ι → ℝ) {t C : ℝ} (hf : ∀ i, t ≤ f i) (hc : ∀ (s : Finset ι) u, t ≤ u → (∀ i ∈ s, f i ≤ u) → (s.card : ℝ) ≤ C * exp ((u - t) / 2)) : Summable (fun i => exp (-f i)) ∧ ∑' i, exp (-f i) ≤ 2 * C * exp (-t)

theorem **finset_card_le_of_separated_family** {ι : Type*} (u : ι → E) {r R : ℝ} (hr : 0 < r) (hR : 0 ≤ R) (hu : Pairwise fun i j => r ≤ dist (u i) (u j)) (w : E) (s : Finset ι) (hs : ∀ i ∈ s, dist (u i) w ≤ R) : (s.card : ℝ) ≤ (1 + 2 * R / r) ^ finrank ℝ E

theorem **gaussian_tail_of_separated** {ι : Type*} (u : ι → E) (w : E) {r q : ℝ} (hr : 0 < r) (hq : 0 < q) (hd : (finrank ℝ E : ℝ) ≤ q * r ^ 2) (hu : Pairwise fun i j => r ≤ dist (u i) (u j)) (hw : ∀ i, r ≤ ‖u i - w‖) : Summable (fun i => exp (-(q * ‖u i - w‖ ^ 2))) ∧ ∑' i, exp (-(q * ‖u i - w‖ ^ 2)) ≤ 2 * 3 ^ finrank ℝ E * exp (-(q * r ^ 2))

theorem **gaussian_tail_addMonoidHom** {A : Type*} [AddGroup A] (u : A →+ E) {b q : ℝ} (hb : 0 < b) (hd : (finrank ℝ E : ℝ) + 1 ≤ q * b) (hu : ∀ y, y ≠ 0 → b ≤ ‖u y‖ ^ 2) : Summable (fun y : {y : A // y ≠ 0} => exp (-(q * ‖u y‖ ^ 2))) ∧ ∑' y : {y : A // y ≠ 0}, exp (-(q * ‖u y‖ ^ 2)) ≤ 2 * 3 ^ finrank ℝ E * exp (-(q * b))

theorem **gaussian_tail_sum_le_of_min_norm_sq** {A : Type*} [AddGroup A] (u : A →+ E) {b q : ℝ} (hb : 0 < b) (hd : (finrank ℝ E : ℝ) + 1 ≤ q * b) (hu : ∀ y, y ≠ 0 → b ≤ ‖u y‖ ^ 2) : Summable (fun y : {y : A // y ≠ 0} => exp (-(q * ‖u y‖ ^ 2))) ∧ ∑' y : {y : A // y ≠ 0}, exp (-(q * ‖u y‖ ^ 2)) ≤ 2 * 4.3 ^ finrank ℝ E * exp (-(q * b))

theorem **gaussian_tail_addMonoidHom_sixteen** {A : Type*} [AddGroup A] (u : A →+ E) {g : ℕ} (hg : finrank ℝ E = 2 * g) {R : ℝ} (hR0 : 0 ≤ R) (hR : 16 * (2 * (g : ℝ) + 1) ≤ R ^ 2) (hu : ∀ y, y ≠ 0 → R ≤ ‖u y‖) : Summable (fun y : {y : A // y ≠ 0} => exp (-‖u y‖ ^ 2 / 16)) ∧ ∑' y : {y : A // y ≠ 0}, exp (-‖u y‖ ^ 2 / 16) ≤ 2 * 9 ^ g * exp (-R ^ 2 / 16)

theorem **gaussian_sum_of_one_separated** {ι : Type*} (u : ι → E) (w : E) (hd : 1 ≤ finrank ℝ E) (hu : Pairwise fun i j => 1 ≤ dist (u i) (u j)) : Summable (fun i => exp (-(π / 2 * ‖w - u i‖ ^ 2))) ∧ ∑' i, exp (-(π / 2 * ‖w - u i‖ ^ 2)) ≤ 2 * (3 * (finrank ℝ E : ℝ)) ^ finrank ℝ E

theorem **gaussian_sum_of_one_separated_of_le** {ι : Type*} (u : ι → E) (w : E) (hd : 1 ≤ finrank ℝ E) (hu : Pairwise fun i j => 1 ≤ dist (u i) (u j)) {ρ : ℝ} (hρ : 0 ≤ ρ) (hw : ∀ i, ρ ≤ ‖w - u i‖) : Summable (fun i => exp (-(π * ‖w - u i‖ ^ 2))) ∧ ∑' i, exp (-(π * ‖w - u i‖ ^ 2)) ≤ 2 * (3 * (finrank ℝ E : ℝ)) ^ finrank ℝ E * exp (-(π / 2 * ρ ^ 2))

theorem **subgroup_gaussian_tail_sixteen** (G : AddSubgroup E) {g : ℕ} (hg : finrank ℝ E = 2 * g) {R : ℝ} (hR0 : 0 ≤ R) (hR : 16 * (2 * (g : ℝ) + 1) ≤ R ^ 2) (hG : ∀ x ∈ G, x ≠ 0 → R ≤ ‖x‖) : Summable (fun x : {x : G // x ≠ 0} => exp (-‖(x.1 : E)‖ ^ 2 / 16)) ∧ ∑' x : {x : G // x ≠ 0}, exp (-‖(x.1 : E)‖ ^ 2 / 16) ≤ 2 * 9 ^ g * exp (-R ^ 2 / 16)

theorem **subgroup_gaussian_tail** (G : AddSubgroup E) {b q : ℝ} (hb : 0 < b) (hd : (finrank ℝ E : ℝ) + 1 ≤ q * b) (hG : ∀ y ∈ G, y ≠ 0 → b ≤ ‖y‖ ^ 2) : Summable (fun y : {y : G // y ≠ 0} => exp (-(q * ‖(y.1 : E)‖ ^ 2))) ∧ ∑' y : {y : G // y ≠ 0}, exp (-(q * ‖(y.1 : E)‖ ^

2)) ≤ 2 * 3 ^ finrank ℝ E * exp (-(q * b))

theorem **subgroup_gaussian_tail_dyadic** (G : AddSubgroup E) {b q : ℝ} (hb : 0 < b) (hd : (finrank ℝ E : ℝ) + 1 ≤ q * b) (hG : ∀ y ∈ G, y ≠ 0 → b ≤ ‖y‖ ^ 2) : Summable (fun y : {y : G // y ≠ 0} => exp (-(q * ‖(y.1 : E)‖ ^ 2))) ∧ ∑' y : {y : G // y ≠ 0}, exp (-(q * ‖(y.1 : E)‖ ^ 2)) ≤ 2 * 4.3 ^ finrank ℝ E * exp (-(q * b))

theorem **set_gaussian_sum_of_one_separated** {S : Set E} (w : E) (hd : 1 ≤ finrank ℝ E) (hS : S.Pairwise fun a b => 1 ≤ dist a b) : (Summable (fun y : S => exp (-(π / 2 * ‖w - y‖ ^ 2))) ∧ ∑' y : S, exp (-(π / 2 * ‖w - y‖ ^ 2)) ≤ 2 * (3 * (finrank ℝ E : ℝ)) ^ finrank ℝ E) ∧ (Summable (fun y : S => exp (-(π * ‖w - y‖ ^ 2))) ∧ ∑' y : S, exp (-(π * ‖w - y‖ ^ 2)) ≤ 2 * (3 * (finrank ℝ E : ℝ)) ^ finrank ℝ E * exp (-(π / 2 * infDist w S ^ 2)))

theorem **gaussian_tail_sums** : (∀ (G : AddSubgroup E) (g : ℕ) (R : ℝ), finrank ℝ E = 2 * g → 0 ≤ R → 16 * (2 * (g : ℝ) + 1) ≤ R ^ 2 → (∀ x ∈ G, x ≠ 0 → R ≤ ‖x‖) → Summable (fun x : {x : G // x ≠ 0} => exp (-‖(x.1 : E)‖ ^ 2 / 16)) ∧ ∑' x : {x : G // x ≠ 0}, exp (-‖(x.1 : E)‖ ^ 2 / 16) ≤ 2 * 9 ^ g * exp (-R ^ 2 / 16)) ∧ (∀ (S : Set E) (w : E), 1 ≤ finrank ℝ E → (S.Pairwise fun a b => 1 ≤ dist a b) → (Summable (fun y : S => exp (-(π / 2 * ‖w - y‖ ^ 2))) ∧ ∑' y : S, exp (-(π / 2 * ‖w - y‖ ^ 2)) ≤ 2 * (3 * (finrank ℝ E : ℝ)) ^ finrank ℝ E) ∧ (Summable (fun y : S => exp (-(π * ‖w - y‖ ^ 2))) ∧ ∑' y : S, exp (-(π * ‖w - y‖ ^ 2)) ≤ 2 * (3 * (finrank ℝ E : ℝ)) ^ finrank ℝ E * exp (-(π / 2 * infDist w S ^ 2)))) ∧ (∀ (G : AddSubgroup E) (b q : ℝ), 0 < b → (finrank ℝ E : ℝ) + 1 ≤ q * b → (∀ y ∈ G, y ≠ 0 → b ≤ ‖y‖ ^ 2) → Summable (fun y : {y : G // y ≠ 0} => exp (-(q * ‖(y.1 : E)‖ ^ 2))) ∧ ∑' y : {y :

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.GramMinorDetTriangularMul

theorem **submatrix_mul_upperTriangular_castLE** {n m : ℕ} (A : Matrix (Fin n) (Fin m) ℂ) (u : Matrix (Fin m) (Fin m) ℂ) {j : ℕ} (hj : j ≤ m) (hu : u.BlockTriangular id) : (A * u).submatrix id (Fin.castLE hj) = A.submatrix id (Fin.castLE hj) * u.submatrix (Fin.castLE hj) (Fin.castLE hj)

theorem **det_leading_gram_mul_upperTriangular** : ∀ (n m : ℕ) (A : Matrix (Fin n) (Fin m) ℂ) (u : Matrix (Fin m) (Fin m) ℂ) (j : ℕ) (hj : j ≤ m), u.BlockTriangular id → (((A * u)ᴴ * (A * u)).submatrix (Fin.castLE hj) (Fin.castLE hj)).det = ((∏ b : Fin j, Complex.normSq (u (Fin.castLE hj b) (Fin.castLE hj b)) : ℝ) : ℂ) * ((Aᴴ * A).submatrix (Fin.castLE hj) (Fin.castLE hj)).det

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.GramMinorDetTriangularMulStatement

theorem **_root_.GlobalTheory.T.gram_minor_det_triangular_mul** : ∀ (n m : ℕ) (A : Matrix (Fin n) (Fin m) ℂ) (u : Matrix (Fin m) (Fin m) ℂ) (j : ℕ) (hj : j ≤ m), u.BlockTriangular id → (((A * u)ᴴ * (A * u)).submatrix (Fin.castLE hj) (Fin.castLE hj)).det = ((∏ b : Fin j, Complex.normSq (u (Fin.castLE hj b) (Fin.castLE hj b)) : ℝ) : ℂ) * ((Aᴴ * A).submatrix (Fin.castLE hj) (Fin.castLE hj)).det

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.GramPositiveVolumeContinuous

theorem **sq_half** : (((Real.sqrt 2)⁻¹ : ℝ) : ℂ) * (((Real.sqrt 2)⁻¹ : ℝ) : ℂ) = 1 / 2

theorem **bvec_pair** {n : ℕ} (M : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : star (bvec i) ⬝ᵥ M *ᵥ bvec j = M i j

theorem **frame_pair** (k r : ℕ) (M : Mat k r) (a b : Fin (r + 1)) : ((frame k r)ᴴ * M * frame k r) a b = star (fvec k r (isoVec k r) a) ⬝ᵥ M *ᵥ fvec k r (isoVec k r) b

theorem **pair_iso_left** (k r : ℕ) (M : Mat k r) (j) : star (isoVec k r) ⬝ᵥ M *ᵥ bvec j = (((Real.sqrt 2)⁻¹ : ℝ) : ℂ) * (M (wI k r 0) j + M (wI k r (Fin.last k)) j)

theorem **pair_iso_right** (k r : ℕ) (M : Mat k r) (v) : star v ⬝ᵥ M *ᵥ isoVec k r = (((Real.sqrt 2)⁻¹ : ℝ) : ℂ) * (star v ⬝ᵥ M *ᵥ bvec (wI k r 0) + star v ⬝ᵥ M *ᵥ bvec (wI k r (Fin.last k)))

theorem **frame_orthonormal** (k r : ℕ) (hk : 1 ≤ k) : (frame k r)ᴴ * frame k r = 1

lemma **mulVec_inj** (k r : ℕ) (hk : 1 ≤ k) (g : Mat k r) (hg : g ∈ GV k r) : Function.Injective (g * frame k r).mulVec

lemma **gram_pos** (k r : ℕ) (hk : 1 ≤ k) (g : Mat k r) (hg : g ∈ GV k r) (a : Fin (r + 1)) : 0 < gram k r g a

lemma **gram_continuous** (k r : ℕ) (a : Fin (r + 1)) : Continuous fun g : Mat k r => gram k r g a

lemma **gram_measurable** (k r : ℕ) (a : Fin (r + 1)) : Measurable fun x : Mt k r => gram k r (Matrix.of x) a

theorem **gram_vHalf_pos_measurable_continuousOn** : ∀ k r : ℕ, 1 ≤ k → (∀ g ∈ GV k r, ∀ a : Fin (r + 1), 0 < gram k r g a) ∧ (∀ g ∈ GV k r, 0 < vHalf k r g) ∧ (Measurable fun x : Mt k r => vHalf k r (Matrix.of x)) ∧ ContinuousOn (vHalf k r) (GV k r)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.GramPositiveVolumeContinuousStatement

theorem **_root_.GlobalTheory.T.gram_positive_volume_continuous** : ∀ k r : ℕ, 1 ≤ k → (∀ g ∈ GV k r, ∀ a : Fin (r + 1), 0 < gram k r g a) ∧ (∀ g ∈ GV k r, 0 < vHalf k r g) ∧ (Measurable fun x : Mt k r => vHalf k r (Matrix.of x)) ∧ ContinuousOn (vHalf k r) (GV k r)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.HalfVolumeEquivariant

theorem **gram_nonneg_vmul** (k r : ℕ) (g : Mat k r) (a : Fin (r + 1)) : 0 ≤ gram k r g a

theorem **gram_unitary_mul** (k r : ℕ) (κ g : Mat k r) (hκ : κᴴ * κ = 1) (a : Fin (r + 1)) : gram k r (κ * g) a = gram k r g a

theorem **gram_mul_flag** (k r : ℕ) (hk : 1 ≤ k) (g p : Mat k r) (hp : p ∈ Pset k r) (a : Fin (r + 1)) : gram k r (g * p) a = gram k r g a * gram k r p a

theorem **vHalf_left_inv_and_mul** : ∀ k r : ℕ, 1 ≤ k → (∀ κ ∈ Kset k r, ∀ g : Mat k r, vHalf k r (κ * g) = vHalf k r g) ∧ ∀ g ∈ GV k r, ∀ p ∈ Pset k r, vHalf k r (g * p) = vHalf k r g * vHalf k r p

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.HalfVolumeEquivariantStatement

theorem **_root_.GlobalTheory.T.half_volume_equivariant** : ∀ k r : ℕ, 1 ≤ k → (∀ κ ∈ Kset k r, ∀ g : Mat k r, vHalf k r (κ * g) = vHalf k r g) ∧ ∀ g ∈ GV k r, ∀ p ∈ Pset k r, vHalf k r (g * p) = vHalf k r g * vHalf k r p

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.HalfVolumeLeviRecursion

theorem **det_gram_cons** {n m : ℕ} (e : Fin n → ℂ) (he : star e ⬝ᵥ e = 1) (w : Fin m → Fin n → ℂ) (hw : ∀ b, star e ⬝ᵥ w b = 0) (a : ℂ) (c : Fin m → ℂ) (v : Fin (m + 1) → Fin n → ℂ) (h0 : v 0 = a • e) (hs : ∀ b, v b.succ = w b + c b • e) : (Matrix.of fun i j => star (v i) ⬝ᵥ v j).det = (Complex.normSq a : ℂ) * (Matrix.of fun i j => star (w i) ⬝ᵥ w j).det

theorem **gram_eq** (k r : ℕ) (g : Mat k r) (j : Fin (r + 1)) : gram k r g j = (Matrix.of fun i i' : Fin (j.val + 1) => star (g *ᵥ fvec k r (isoVec k r) (Fin.castLE (Nat.succ_le_of_lt j.isLt) i)) ⬝ᵥ (g *ᵥ fvec k r (isoVec k r) (Fin.castLE (Nat.succ_le_of_lt j.isLt) i'))).det.re

theorem **gram_nonneg** (k r : ℕ) (g : Mat k r) (j : Fin (r + 1)) : 0 ≤ gram k r g j

theorem **gram_K** (k r : ℕ) (κ g : Mat k r) (hκ : κᴴ * κ = 1) (j : Fin (r + 1)) : gram k r (κ * g) j = gram k r g j

theorem **fvec_zero** (k r' : ℕ) (j : Fin (1 + r' + 1)) (hj : j.val = 0) : fvec k (1 + r') (isoVec k (1 + r')) j = e0 k r'

theorem **liftVec_add'** (k i r' : ℕ) (v w : Fin (2 * r' + k + 1) → ℂ) : liftVec k i r' (v + w) = liftVec k i r' v + liftVec k i r' w

theorem **liftVec_smul'** (k i r' : ℕ) (c : ℂ) (v : Fin (2 * r' + k + 1) → ℂ) : liftVec k i r' (c • v) = c • liftVec k i r' v

theorem **fvec_succ** (k r' : ℕ) (j : Fin (1 + r' + 1)) (b : Fin (r' + 1)) (hj : j.val = b.val + 1) : fvec k (1 + r') (isoVec k (1 + r')) j = liftVec k 1 r' (fvec k r' (isoVec k r') b)

theorem **star_e0_dotProduct** (k r' : ℕ) (v : Fin (2 * (1 + r') + k + 1) → ℂ) : star (e0 k r') ⬝ᵥ v = v (eI k (1 + r') (a0 r'))

theorem **star_e0_dotProduct_self** (k r' : ℕ) : star (e0 k r') ⬝ᵥ e0 k r' = 1

theorem **star_e0_dotProduct_lift** (k r' : ℕ) (w : Fin (2 * r' + k + 1) → ℂ) : star (e0 k r') ⬝ᵥ liftVec k 1 r' w = 0

theorem **gram_zero** (j : Fin (1 + r' + 1)) (hj : j.val = 0) : gram k (1 + r') M j = Complex.normSq a

theorem **gram_succ** (j : Fin (1 + r' + 1)) (b : Fin (r' + 1)) (hj : j.val = b.val + 1) : gram k (1 + r') M j = Complex.normSq a * gram k r' h b

theorem **vHalf_block** (ha : a ≠ 0) : vHalf k (1 + r') M = Complex.normSq a ^ (-(((2 * (1 + r') + k : ℕ) : ℝ) / 2)) * vHalf k r' h

theorem **sqLen_eq** {n : ℕ} (v : Fin n → ℂ) : sqLen v = (star v ⬝ᵥ v).re

theorem **levi_embedding_mulVec_e0** (k r' : ℕ) (y : Mat k r') : ιLev k 1 r' y *ᵥ e0 k r' = e0 k r'

theorem **parabolic_mulVec_e0_smul** {k r' : ℕ} {p : Mat k (1 + r')} (hp : p ∈ Pset k (1 + r')) : ∃ a : ℂ, p *ᵥ e0 k r' = a • e0 k r'

theorem **vHalf_level** : ∀ k r' : ℕ, 1 ≤ k → ∀ x ∈ GV k (1 + r'), ∃ g₁ ∈ GV k r', ∀ y ∈ GV k r', vHalf k (1 + r') (x * ιLev k 1 r' y) = sqLen (x *ᵥ e0 k r') ^ (-(((2 * (1 + r') + k : ℕ) : ℝ) / 2)) * vHalf k r' (g₁ * y)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.HalfVolumeLeviRecursionStatement

theorem **_root_.GlobalTheory.T.half_volume_levi_recursion** : ∀ k r' : ℕ, 1 ≤ k → ∀ x ∈ GV k (1 + r'), ∃ g₁ ∈ GV k r', ∀ y ∈ GV k r', vHalf k (1 + r') (x * ιLev k 1 r' y) = sqLen (x *ᵥ e0 k r') ^ (-(((2 * (1 + r') + k : ℕ) : ℝ) / 2)) * vHalf k r' (g₁ * y)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.HasSumLocallyUniformlyOnIteratedFDerivRiemannThe

@[simp] theorem **riemannThetaJointForm_apply** (a : ι → ℝ) (n : ι → ℤ) (p : (ι → ℂ) × (ι → ι → ℂ)) : riemannThetaJointForm a n p = 2 * π * I * (((fun i => (n i : ℂ)) + a i) ⬝ᵥ (of p.2 *ᵥ fun i => (n i : ℂ) + a i)) / 2 + (fun i => (n i : ℂ) + a i) ⬝ᵥ p.1)

theorem **riemannThetaTerm_eq_cexp_jointForm** (a b : ι → ℝ) (n : ι → ℤ) (p : (ι → ℂ) × (ι → ι → ℂ)) : riemannThetaTerm a b (of p.2) p.1 n = cexp (riemannThetaJointForm a n p + riemannThetaForm a n fun i => (b i : ℂ))

theorem **differentiable_riemannThetaTerm** (a b : ι → ℝ) (n : ι → ℤ) : Differentiable ℂ fun p : (ι → ℂ) × (ι → ι → ℂ) => riemannThetaTerm a b (of p.2) p.1 n

theorem **IsSiegel.exists_isOpen_norm_riemannThetaTerm_le** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : ∃ U : Set ((ι → ℂ) × (ι → ι → ℂ)), IsOpen U ∧ (v, of.symm Ω) ∈ U ∧ ∃ u : (ι → ℤ) → ℝ, Summable u ∧ ∀ n, ∀ p ∈ U, ‖riemannThetaTerm a b (of p.2) p.1 n‖ ≤ u n

theorem **IsSiegel.analyticAt_riemannTheta_entries** {Ω : Matrix ι ι ℂ} (h : IsSiegel Ω) (a b : ι → ℝ) (v : ι → ℂ) : AnalyticAt ℂ (fun p : (ι → ℂ) × (ι → ι → ℂ) => riemannTheta a b (of p.2) p.1) (v, of.symm Ω)

theorem **differentiableOn_riemannTheta_comp** (a b : ι → ℝ) {Ω : E → Matrix ι ι ℂ} {v : E → ι → ℂ} {U : Set E} (h : ∀ x ∈ U, IsSiegel (Ω x)) (hΩ : ∀ i j, DifferentiableOn ℂ (fun y => Ω y i j) U) (hv : DifferentiableOn ℂ v U) : DifferentiableOn ℂ (fun y => riemannTheta a b (Ω y) (v y)) U

@[simp] theorem **mem_siegelUpperCoords** {x : {p : ι × ι // p.1 ≤ p.2} → ℂ} : x ∈ siegelUpperCoords ι ↔ IsSiegel (symmOfUpper x)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IntegrableMonomialMulConjComplexGaussian

theorem **integrable_exp_neg_mul_norm_sq** {b : ℝ} (hb : 0 < b) : Integrable fun z : E => exp (-b * ‖z‖ ^ 2)

theorem **integral_exp_neg_mul_norm_sq** {g : ℕ} (hE : Module.finrank ℝ E = 2 * g) {b : ℝ} (hb : 0 < b) : ∫ z : E, exp (-b * ‖z‖ ^ 2) = (π / b) ^ g

theorem **isProbabilityMeasure_complexGaussian** {g : ℕ} (hE : Module.finrank ℝ E = 2 * g) : IsProbabilityMeasure (complexGaussian E g)

theorem **integrable_norm_pow_mul_exp_neg_mul_norm_sq** (n : ℕ) {b : ℝ} (hb : 0 < b) : Integrable fun z : E => ‖z‖ ^ n * exp (-b * ‖z‖ ^ 2)

theorem **finrank_real_euclideanSpace_complex** (ι : Type*) [Fintype ι] : Module.finrank ℝ (EuclideanSpace ℂ ι) = 2 * Fintype.card ι

theorem **volume_preserving_ofLp_complex** (ι : Type*) [Fintype ι] : MeasurePreserving (@WithLp.ofLp 2 (ι → ℂ))

theorem **finrank_real_euclideanSpace_complex_fin** (g : ℕ) : Module.finrank ℝ (EuclideanSpace ℂ (Fin g)) = 2 * g

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IntegrableOnOfNormLeSumAbsLogAddLogLog

theorem **log_norm_le_circleAverage_log_norm** {g : ℂ → ℂ} (hg : AnalyticOnNhd ℂ g (closedBall 0 1)) (h0 : g 0 ≠ 0) : Real.log ‖g 0‖ ≤ Real.circleAverage (fun z => Real.log ‖g z‖) 0 1

private theorem **measurable_logGap** (M : ℝ) : Measurable (logGap M)

private theorem **lintegral_logGap_circle_le** {g : ℂ → ℂ} (hg : AnalyticOnNhd ℂ g (closedBall 0 1)) (h0 : g 0 ≠ 0) {M : ℝ} (hM : ∀ z ∈ closedBall (0 : ℂ) 1, ‖g z‖ ≤ M) : ∫⁻ t in Ioc 0 (2 * π), logGap M (g (circleMap 0 1 t)) ≤ ENNReal.ofReal (2 * π * (Real.log M - Real.log ‖g 0‖))

theorem **integrableOn_log_norm_ball_zero** (μ : Measure E) [μ.IsAddHaarMeasure] {f : E → ℂ} {r M : ℝ} (hf : DifferentiableOn ℂ f (ball 0 r)) (hM : ∀ z ∈ ball (0 : E) r, ‖f z‖ ≤ M) (h0 : f 0 ≠ 0) : (∀ᵐ z ∂(μ.restrict (ball 0 r)), f z ≠ 0) ∧ IntegrableOn (fun z => Real.log ‖f z‖) (ball 0 r) μ

theorem **integrableOn_log_norm_ball** (μ : Measure E) [μ.IsAddHaarMeasure] {f : E → ℂ} {a : E} {r M : ℝ} (hf : DifferentiableOn ℂ f (ball a r)) (hM : ∀ z ∈ ball a r, ‖f z‖ ≤ M) (h0 : f a ≠ 0) : (∀ᵐ z ∂(μ.restrict (ball a r)), f z ≠ 0) ∧ IntegrableOn (fun z => Real.log ‖f z‖) (ball a r) μ

theorem **exists_integrableOn_log_norm_of_frequently_ne_zero** (μ : Measure E) [μ.IsAddHaarMeasure] {f : E → ℂ} {x : E} (hf : ∀ᶠ z in 𝓝 x, DifferentiableAt ℂ f z) (hx : ∃ᶠ z in 𝓝 x, f z ≠ 0) : ∃ r > 0, (∀ᵐ z ∂(μ.restrict (ball x r)), f z ≠ 0) ∧ IntegrableOn (fun z => Real.log ‖f z‖) (ball x r) μ

theorem **locallyIntegrableOn_log_norm_of_frequently_ne_zero** (μ : Measure E) [μ.IsAddHaarMeasure] {f : E → ℂ} {U : Set E} (hU : IsOpen U) (hf : DifferentiableOn ℂ f U) (hx : ∀ x ∈ U, ∃ᶠ z in 𝓝 x, f z ≠ 0) : (∀ᵐ z ∂(μ.restrict U), f z ≠ 0) ∧ LocallyIntegrableOn (fun z =>

Real.log ‖f z‖) U μ

theorem **exists_integrableOn_log_norm_div** (μ : Measure E) [μ.IsAddHaarMeasure] {g h : E → ℂ} {x : E} (hg : ∀ᶠ z in 𝒩 x, DifferentiableAt ℂ g z) (hh : ∀ᶠ z in 𝒩 x, DifferentiableAt ℂ h z) (hg0 : ∃ᶠ z in 𝒩 x, g z ≠ 0) (hh0 : ∃ᶠ z in 𝒩 x, h z ≠ 0) : ∃ r > 0, (∀ᵐ z ∂(μ.restrict (ball x r)), g z ≠ 0 ∧ h z ≠ 0) ∧ IntegrableOn (fun z => Real.log ‖g z / h z‖) (ball x r) μ

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IntegralConjInnerMulInnerOfIsEmptyEquiv

theorem **eq_haarProb** (μ : Measure K) [μ.IsHaarMeasure] [IsProbabilityMeasure μ] : μ = haarProb K

theorem **eq_smul_haarProb** (μ : Measure K) [μ.IsHaarMeasure] : μ = μ Set.univ • haarProb K

theorem **isMulRightInvariant_of_compactSpace** (μ : Measure K) [μ.IsHaarMeasure] : μ.IsMulRightInvariant

theorem **isInvInvariant_of_compactSpace** (μ : Measure K) [μ.IsHaarMeasure] : μ.IsInvInvariant

theorem **isCompact_unitaryGroup** (n 𝕜 : Type*) [Fintype n] [DecidableEq n] [RCLike 𝕜] : IsCompact (Matrix.unitaryGroup n 𝕜 : Set (Matrix n n 𝕜))

theorem **integrable_of_continuous_of_compactSpace** [IsFiniteMeasure μ] {E : Type*} [NormedAddCommGroup E] {f : K → E} (hf : Continuous f) : Integrable f μ

theorem **inner_map_map_of_norm_map** {G : Type*} [Monoid G] {𝕜 E : Type*} [RCLike 𝕜] [NormedAddCommGroup E] [InnerProductSpace 𝕜 E] (ρ : Representation 𝕜 G E) (h : ∀ k v, ‖ρ k v‖ = ‖v‖) (k : G) (u v : E) : ⟪ρ k u, ρ k v⟫_𝕜 = ⟪u, v⟫_𝕜

theorem **inner_rankOneAverage** (hρc : ∀ v, Continuous fun k => ρ k v) (hρu : ∀ k v, ‖ρ k v‖ = ‖v‖) (hσc : ∀ w, Continuous fun k => σ k w) (u v : V) (u' w : W) : ⟪u', rankOneAverage μ ρ σ hρc hρu hσc v w u⟫_ℂ = ∫ k, conj ⟪u, ρ k v⟫_ℂ * ⟪u', σ k w⟫_ℂ ∂μ

theorem **finrank_pos_of_isIrreducible** [h : ρ.IsIrreducible] : 0 < Module.finrank ℂ V

theorem **integral_conj_inner_mul_inner** [ρ.IsIrreducible] (hρc : ∀ v, Continuous fun k => ρ k v) (hρu : ∀ k v, ‖ρ k v‖ = ‖v‖) (u v u' v' : V) : ∫ k, conj ⟪u, ρ k v⟫_ℂ * ⟪u', ρ k v'⟫_ℂ ∂μ = ⟪u', u⟫_ℂ * ⟪v, v'⟫_ℂ / (Module.finrank ℂ V : ℂ)

theorem **integral_conj_entry_mul_entry** [ρ.IsIrreducible] (hρc : ∀ v, Continuous fun k => ρ k v) (hρu : ∀ k v, ‖ρ k v‖ = ‖v‖) {ι : Type*} [Fintype ι] [DecidableEq ι] (b : OrthonormalBasis ι ℂ V) (i j i' j' : ι) : ∫ k, conj ⟪b i, ρ k (b j)⟫_ℂ * ⟪b i', ρ k (b j')⟫_ℂ ∂μ = if i = i' ∧ j = j' then (Module.finrank ℂ V : ℂ)⁻¹ else 0

theorem **integral_conj_character_mul_character_self** [ρ.IsIrreducible] (hρc : ∀ v, Continuous fun k => ρ k v) (hρu : ∀ k v, ‖ρ k v‖ = ‖v‖) : ∫ k, conj (ρ.character k) * ρ.character k ∂μ = 1

theorem **isSemisimpleRepresentation_of_norm_map** {G V : Type*} [Group G] [NormedAddCommGroup V] [InnerProductSpace 𝕜 V] [FiniteDimensional 𝕜 V] (ρ : Representation 𝕜 G V) (hρu : ∀ k v, ‖ρ k v‖ = ‖v‖) : ρ.IsSemisimpleRepresentation

theorem **isSemisimpleRepresentation_of_invariant_core** {G V : Type*} [Group G] [AddCommGroup V] [Module 𝕜 V] [FiniteDimensional 𝕜 V] (ρ : Representation 𝕜 G V) (c : InnerProductSpace.Core 𝕜 V) (hc : ∀ k u v, c.inner (ρ k u) (ρ k v) = c.inner u v) : ρ.IsSemisimpleRepresentation

theorem **isIrreducible_toRepresentation_of_isAtom** {F G V : Type*} [Field F] [Monoid G] [AddCommGroup V] [Module F V] {ρ : Representation F G V} {U : Subrepresentation ρ} (hU : IsAtom U) : U.toRepresentation.IsIrreducible

theorem **averageCore_inner_map_map** [MeasurableSpace K] [BorelSpace K] {V E : Type*} [AddCommGroup V] [Module 𝕜 V] [NormedAddCommGroup E] [InnerProductSpace 𝕜 E] (ρ : Representation 𝕜 K V) (e : V ↪ₗ[𝕜] E) (he : Function.Injective e) (hc : ∀ v, Continuous fun k => e (ρ k v)) (g : K) (u v : V) : (averageCore ρ e he hc).inner (ρ g u) (ρ g v) = (averageCore ρ e he hc).inner u v

theorem **exists_invariant_core_of_continuous_coeff** {V : Type*} [AddCommGroup V] [Module 𝕜 V] [FiniteDimensional 𝕜 V] (ρ : Representation 𝕜 K V) (hρc : ∀ (l : Module.Dual 𝕜 V) (v : V), Continuous fun k => l (ρ k v)) : ∃ c : InnerProductSpace.Core 𝕜 V, ∀ k u v, c.inner (ρ k u) (ρ k v) = c.inner u v

theorem **isSemisimpleRepresentation_of_continuous_coeff** {V : Type*} [AddCommGroup V] [Module 𝕜 V] [FiniteDimensional 𝕜 V] (ρ : Representation 𝕜 K V) (hρc : ∀ (l : Module.Dual 𝕜 V) (v : V), Continuous fun k => l (ρ k v)) : ρ.IsSemisimpleRepresentation

theorem **exists_invariant_core** (ρ : Representation 𝕜 K V) (hρc : ∀ v, Continuous fun k => ρ k v) : ∃ c : InnerProductSpace.Core 𝕜 V, ∀ k u v, c.inner (ρ k u) (ρ k v) = c.inner u v

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IntegralLogNormLeHalfLogIntegralSq

theorem **coe_sphereRotation** (θ : ℝ) (x : sphere (0 : E) 1) : (sphereRotation θ x : E) = circleMap 0 1 θ • (x : E)

theorem **continuous_sphereRotation** : Continuous (Function.uncurry (sphereRotation (E

theorem **measurePreserving_sphereRotation** [FiniteDimensional ℂ E] (μ : Measure E) [μ.IsAddHaarMeasure] (θ : ℝ) : MeasurePreserving (sphereRotation θ) μ.toSphere μ.toSphere

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsClosedImmersionIffPullRatio

theorem **appLE_of_eq** {φ ψ : Z ⟶ Y} (h : φ = ψ) (U : Y.Opens) (V : Z.Opens) (h₁ : V ≤ φ ⁻¹ᵁ U) (h₂ : V ≤ ψ ⁻¹ᵁ U) : φ.appLE U V h₁ = ψ.appLE U V h₂

theorem **resG_appLE** (g : Z ⟶ Y) {U : Y.Opens} {V W : Z.Opens} (h : V ≤ g ⁻¹ᵁ U) (h' : W ≤ V) (a : Γ(Y, U)) : resG Z h' ((g.appLE U V h).hom a) = (g.appLE U W (h'.trans h)).hom a

theorem **appLE_resG** (g : Z ⟶ Y) {U U' : Y.Opens} {V : Z.Opens} (h' : U' ≤ U) (h : V ≤ g ⁻¹ᵁ U') (a : Γ(Y, U)) : (g.appLE U' V h).hom (resG Y h' a) = (g.appLE U V (h.trans (g.preimage_mono h'))).hom a

theorem **appLE_appLE** {X : Scheme.{0}} (g : Z ⟶ Y) (φ : Y ⟶ X) {U : X.Opens} {V : Y.Opens} {W : Z.Opens} (h : V ≤ φ ⁻¹ᵁ U) (h' : W ≤ g ⁻¹ᵁ V) (a : Γ(X, U)) : (g.appLE V W h').hom ((φ.appLE U V h).hom a) = ((g ≫ φ).appLE U W (h'.trans (g.preimage_mono h))).hom a

theorem **baseRingHom_appLE** {B : CommRingCat.{0}} (p : Y ⟶ Spec B) (g : Z ⟶ Y) (U : Y.Opens) (V : Z.Opens) (h : V ≤ g ⁻¹ᵁ U) : baseRingHom p U ≫ g.appLE U V h = baseRingHom (g ≫ p) V

theorem **eq_of_baseRingHom_top** {B : CommRingCat.{0}} {p q : Y ⟶ Spec B} (h : baseRingHom p ⊤ = baseRingHom q ⊤) : p = q

theorem **resG_pullRatio** (h : V ≤ W) : resG Y h (pullRatio φ W i hi j) = pullRatio φ V i (h.trans hi) j

@[simp] theorem **pullRatio_self** : pullRatio φ W i hi i = 1

theorem **appLE_ratioOn** {U : (PSpace R N).Opens} (hU : U ≤ chart R N i) (hW : W ≤ φ ⁻¹ᵁ U) : (φ.appLE U W hW).hom (ratioOn U i hU j) = pullRatio φ W i (hW.trans (φ.preimage_mono hU)) j

theorem **pullRatio_mul_pullRatio** : pullRatio φ W i hi j * pullRatio φ W j hj k = pullRatio φ W i hi k

theorem **basicOpen_pullRatio** : Y.basicOpen (pullRatio φ W i hi j) = W ⊓ φ ⁻¹ᵁ chart R N j

theorem **appLE_pullRatio** (g : Z ⟶ Y) {T : Z.Opens} (h : T ≤ g ⁻¹ᵁ W) : (g.appLE W T h).hom (pullRatio φ W i hi j) = pullRatio (g ≫ φ) T i (h.trans (g.preimage_mono hi)) j

theorem **pullRatio_of_eq** {ψ : Y ⟶ PSpace R N} (h : φ = ψ) (hi : W ≤ φ ⁻¹ᵁ chart R N i) (hi' : W ≤ ψ ⁻¹ᵁ chart R N i) : pullRatio φ W i hi j = pullRatio ψ W i hi' j

theorem **appLE_comp_aeval_ratio** : (φ.appLE (chart R N i) W hi).hom.comp (RingHomClass.toRingHom (aeval (R

theorem **ι_comp_eq** (φ : Y ⟶ PSpace R N) (W : Y.Opens) (i : Fin (N + 1)) (hi : W ≤ φ ⁻¹ᵁ chart R N i) : W.ι ≫ φ = W.toSpecΓ ≫ Spec.map (φ.appLE (chart R N i) W hi) ≫ (isAffineOpen_chart R N i).fromSpec

theorem **hom_ext** {φ ψ : Y ⟶ PSpace R N} (h : φ ≫ PSpace.str R N = ψ ≫ PSpace.str R N) (H : ∀ y : Y, ∃ (W : Y.Opens) (_ : y ∈ W) (i : Fin (N + 1)) (h₁ : W ≤ φ ⁻¹ᵁ chart R N i) (h₂ : W ≤ ψ ⁻¹ᵁ chart R N i), ∀ j, pullRatio φ W i h₁ j = pullRatio ψ W i h₂ j) : φ = ψ

theorem **ofChartHom_preimage** (i : Fin (N + 1)) (ρ : Γ(PSpace R N, chart R N i) ⟶ Γ(Y, ⊤)) : ⊤ ≤ ofChartHom i ρ ⁻¹ᵁ chart R N i

theorem **ofChartHom_appLE** (i : Fin (N + 1)) (ρ : Γ(PSpace R N, chart R N i) ⟶ Γ(Y, ⊤)) : (ofChartHom i ρ).appLE (chart R N i) ⊤ (ofChartHom_preimage i ρ) = ρ

theorem **chartLift_algebraMap** {A : Type} [CommRing A] (i : Fin (N + 1)) (c : R →+* A) (r : Fin (N + 1) → A) (a : R) : chartLift i c r (algebraMap R _ a) = c a

theorem **chartLift_ratio** {A : Type} [CommRing A] (i : Fin (N + 1)) (c : R →+* A) {r : Fin (N + 1) → A} (hr : r i = 1) (j : Fin (N + 1)) : chartLift i c r (ratio R N i j) = r j

theorem **ofCoords_preimage_top** (f : Y ⟶ Sp R) (U : Y.Opens) (i : Fin (N + 1)) (r : Fin (N + 1) → Γ(Y, U)) : ⊤ ≤ ofCoords f U i r ⁻¹ᵁ chart R N i

theorem **ofCoords_appLE** (f : Y ⟶ Sp R) (U : Y.Opens) (i : Fin (N + 1)) (r : Fin (N + 1) → Γ(Y, U)) : (ofCoords f U i r).appLE (chart R N i) ⊤ (ofCoords_preimage_top f U i r) = CommRingCat.ofHom (R

theorem **ofCoords_preimage** (f : Y ⟶ Sp R) (U : Y.Opens) (i : Fin (N + 1)) (r : Fin (N + 1) → Γ(Y, U)) (g : Z ⟶ U.toScheme) : ⊤ ≤ (g ≫ ofCoords f U i r) ⁻¹ᵁ chart R N i

theorem **ofCoords_str** (f : Y ⟶ Sp R) (U : Y.Opens) (i : Fin (N + 1)) (r : Fin (N + 1) → Γ(Y, U)) : ofCoords f U i r ≫ PSpace.str R N = U.ι ≫ f

theorem **pullRatio_comp_ofCoords** (f : Y ⟶ Sp R) (U : Y.Opens) (i : Fin (N + 1)) {r : Fin (N + 1) → Γ(Y, U)} (hr : r i = 1) (g : Z ⟶ U.toScheme) (T : Z.Opens) (h : T ≤ (g ≫ ofCoords f U i r) ⁻¹ᵁ chart R N i) (h' : T ≤ (g ≫ U.ι) ⁻¹ᵁ U) (j : Fin (N + 1)) : pullRatio (g ≫ ofCoords f U i r) T i h j = ((g ≫ U.ι).appLE U T h').hom (r j)

theorem **isCoordFamily_pullRatio** (φ : Y ⟶ PSpace R N) : IsCoordFamily (fun i => φ ⁻¹ᵁ chart R N i) (fun i => pullRatio φ _ i le_rfl) where

theorem **appLE_mul** (g : Z ⟶ Y) {T : Z.Opens} {i j : Fin (N + 1)} (hi : T ≤ g ⁻¹ᵁ U i) (hj : T ≤ g ⁻¹ᵁ U j) (k : Fin (N + 1)) : (g.appLE _ T hi).hom (r i j) * (g.appLE _ T hj).hom (r j k) = (g.appLE _ T hi).hom (r i k)

theorem **isUnit_appLE** (g : Z ⟶ Y) {T : Z.Opens} {i j : Fin (N + 1)} (hi : T ≤ g ⁻¹ᵁ U i) (hj : T ≤ g ⁻¹ᵁ U j) : IsUnit ((g.appLE _ T hi).hom (r i j))

theorem **basicOpen_eq** (i j : Fin (N + 1)) : Y.basicOpen (r i j) = U i ⊓ U j

theorem **comp_ofCoords_eq** (f : Y ⟶ Sp R) (i j : Fin (N + 1)) (a : Z ⟶ (U i).toScheme) (b : Z ⟶ (U j).toScheme) (e : a ≫ (U i).ι = b ≫ (U j).ι) : a ≫ ofCoords f (U i) i (r i) = b ≫ ofCoords f (U j) j (r j)

theorem **ι_toHom** (f : Y ⟶ Sp R) (i : Fin (N + 1)) : (U i).ι ≫ H.toHom f = ofCoords f (U i) i (r i)

theorem **exists_mem** (y : Y) : ∃ i, y ∈ U i

theorem **toHom_str** (f : Y ⟶ Sp R) : H.toHom f ≫ PSpace.str R N = f

theorem **le_toHom_preimage** (f : Y ⟶ Sp R) (i : Fin (N + 1)) : U i ≤ H.toHom f ⁻¹ᵁ chart R N i

theorem **pullRatio_toHom** (f : Y ⟶ Sp R) (i j : Fin (N + 1)) : pullRatio (H.toHom f) (U i) i (H.le_toHom_preimage f i) j = r i j

theorem **toHom_preimage** (f : Y ⟶ Sp R) (i : Fin (N + 1)) : H.toHom f ⁻¹ᵁ chart R N i = U i

theorem **surjective_appLE_chart_iff** {φ : Y ⟶ PSpace R N} {W : Y.Opens} {i : Fin (N + 1)} (hi : W ≤ φ ⁻¹ᵁ chart R N i) : Function.Surjective (φ.appLE (chart R N i) W hi).hom ↔ Function.Surjective (eval₂Hom (baseRingHom (φ ≫ PSpace.str R N) W).hom (pullRatio φ W i hi))

theorem **isClosedImmersion_iff_pullRatio** (φ : Y ⟶ PSpace R N) {U : Fin (N + 1) → Y.Opens} (hU : ∀ i, φ ⁻¹ᵁ chart R N i = U i) : IsClosedImmersion φ ↔ ∀ i, IsAffineOpen (U i) ∧ Function.Surjective (eval₂Hom (baseRingHom (φ ≫ PSpace.str R N) (U i)).hom (pullRatio φ (U i) i (hU i).ge))

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsClosedImmersionSigmaDesc

private lemma **isClosedEmbedding_sigma_desc_of_pairwise_disjoint_range** {ι : Type} [Finite ι] {Z : ι → Type*} [∀ i, TopologicalSpace (Z i)] {W : Type*} [TopologicalSpace W] (g : ∀ i, Z i → W) (hg : ∀ i, Topology.IsClosedEmbedding (g i)) (h : Pairwise fun i j => Disjoint (Set.range (g i)) (Set.range (g j))) : Topology.IsClosedEmbedding (fun p : Σ i, Z i => g p.1 p.2)

theorem **_root_.GlobalTheory.ProjSpace.isClosedImmersion_sigmaDesc** {ι : Type} [Finite ι] {X : ι → Scheme.{0}} {Y : Scheme.{0}} (f : ∀ i, X i ⟶ Y) (hf : ∀ i, IsClosedImmersion (f i)) (h : Pairwise fun i j => Disjoint (Set.range (f i)) (Set.range (f j))) : IsClosedImmersion (Sigma.desc f)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsHarmonicProdOfDisjointColumns

theorem **laplacian_mul** (t : T) (a : Fin p) (b : Fin q) (P Q : Poly R T p q κ) : laplacian t a b (P * Q) = laplacian t a b P * Q + P * laplacian t a b Q + cross t a b P Q + cross t a b Q P

theorem **cross_add_left** (t : T) (a : Fin p) (b : Fin q) (P P' Q : Poly R T p q κ) : cross t a b (P + P') Q = cross t a b P Q + cross t a b P' Q

theorem **cross_add_right** (t : T) (a : Fin p) (b : Fin q) (P Q Q' : Poly R T p q κ) : cross t a b P (Q + Q') = cross t a b P Q + cross t a b P Q'

theorem **cross_mul_left** (t : T) (a : Fin p) (b : Fin q) (P P' Q : Poly R T p q κ) : cross t a b (P * P') Q = cross t a b P Q * P' + P * cross t a b P' Q

theorem **cross_mul_right** (t : T) (a : Fin p) (b : Fin q) (P Q Q' : Poly R T p q κ) : cross t a b P (Q * Q') = cross t a b P Q * Q' + Q * cross t a b P Q'

@[simp] theorem **cross_C_left** (t : T) (a : Fin p) (b : Fin q) (r : R) (Q : Poly R T p q κ) : cross t a b (C r) Q = 0

@[simp] theorem **cross_C_right** (t : T) (a : Fin p) (b : Fin q) (r : R) (P : Poly R T p q κ) : cross t a b P (C r) = 0

theorem **IsHarmonic.mul_of_cross** {P Q : Poly R T p q κ} (hP : IsHarmonic P) (hQ : IsHarmonic Q) (h : ∀ t a b, cross t a b P Q = 0) (h' : ∀ t a b, cross t a b Q P = 0) : IsHarmonic (P * Q)

theorem **cross_eq_zero_of_mem_adjoin** (t : T) (a : Fin p) (b : Fin q) {s s' : Set (Poly R T p q κ)} (h : ∀ u ∈ s, ∀ v ∈ s', cross t a b u v = 0) {P Q : Poly R T p q κ} (hP : P ∈ Algebra.adjoin R s) (hQ : Q ∈ Algebra.adjoin R s') : cross t a b P Q = 0

theorem **isHarmonic_of_mem_adjoin** {s : Set (Poly R T p q κ)} (hs : ∀ u ∈ s, IsHarmonic u) (h : ∀ t a b, ∀ u ∈ s, ∀ v ∈ s, cross t a b u v = 0) {P : Poly R T p q κ} (hP : P ∈ Algebra.adjoin R s) : IsHarmonic P

theorem **pderiv_rowForm** [DecidableEq T] (t t' : T) (r r' : Fin p ⊕ Fin q) (k : κ) (v : κ → R) : pderiv (t, r, k) (rowForm t' r' v : Poly R T p q κ) = if t' = t ∧ r' = r then C (v k) else 0

theorem **isHarmonic_rowForm** (t : T) (r : Fin p ⊕ Fin q) (v : κ → R) : IsHarmonic (rowForm t r v : Poly R T p q κ)

theorem **cross_rowForm** [DecidableEq T] (t t₁ t₂ : T) (a : Fin p) (b : Fin q) (r₁ r₂ : Fin p ⊕ Fin q) (v w : κ → R) : cross t a b (rowForm t₁ r₁ v) (rowForm t₂ r₂ w : Poly R T p q κ) = if (t₁ = t ∧ r₁ = Sum.inl a) ∧ t₂ = t ∧ r₂ = Sum.inr b then C (v ⬝ᵥ w) else 0

theorem **isHarmonic_of_mem_adjoin_formSet** {ε : T → ι → κ → R} {φ : T → ι' → κ → R} (h : ∀ t c d, ε t c ⬝ᵥ φ t d = 0) {P : Poly R T p q κ} (hP : P ∈ Algebra.adjoin R (formSet ε φ)) : IsHarmonic P

theorem **det_mem_of_forall_mem** {A σ m : Type*} [CommRing A] [SetLike σ A] [SubringClass σ A] [Fintype m] [DecidableEq m] (s : σ) (M : Matrix m m A) (h : ∀ i j, M i j ∈ s) : M.det ∈ s

theorem **minorProd_mem_closure** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → R) (φ : T → Fin q → κ → R) : minorProd c c' ε φ ∈ Subring.closure (formSet ε φ)

theorem **isHarmonic_minorProd** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {ε : T → Fin p → κ → R} {φ : T → Fin q → κ → R} (h : IsPerp ε φ) : IsHarmonic (minorProd c c' ε φ)

theorem **isHarmonic_of_mem_supported** {s : Set (Var T p q κ)} (h : ∀ t a b k, xVar t a k ∈ s → yVar t b k ∉ s) {P : Poly R T p q κ} (hP : P ∈ supported R s) : IsHarmonic P

theorem **isHarmonic_blockPoly** [Fintype κ] [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (col : T → (i : Fin p) → Fin (i.val + 1) → κ) (col' : T → (i : Fin q) → Fin (i.val + 1) → κ) (h : ∀ t i j a b, c t i ≠ 0 → c' t j ≠ 0 → col t i a ≠ col' t j b) : IsHarmonic (blockPoly R c c' col col')

theorem **blockPoly_isWeightedHomogeneous** [Fintype T] {M : Type*} [AddCommMonoid M] (w : T → Fin p ⊕ Fin q → M) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (col : T → (i : Fin p) → Fin (i.val + 1) → κ) (col' : T → (i : Fin q) → Fin (i.val + 1) → κ) : IsWeightedHomogeneous (rowWeight w) (blockPoly R c c' col col') (∑ t, ((∑ a, partOf (c t) a • w t (Sum.inl a)) + ∑ b, partOf (c' t) b • w t (Sum.inr b)))

theorem **exists_eval_blockPoly_eq_one** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (col : T → (i : Fin p) → Fin (i.val + 1) → κ) (col' : T → (i : Fin q) → Fin (i.val + 1) → κ) (h : ∀ t i j a b, c t i ≠ 0 → c t j ≠ 0 → col t i a = col t j b → a.val = b.val) (h' : ∀ t i j a b, c' t i ≠ 0 → c' t j ≠ 0 → col' t i a = col' t j b → a.val = b.val) : ∃ z : Var T p q κ → R, (∀ v, z v = 0 ∨ z v = 1) ∧ eval z (blockPoly R c c' col col') = 1

theorem **blockPoly_ne_zero** [Fintype T] [Nontrivial R] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (col : T → (i : Fin p) → Fin (i.val + 1) → κ) (col' : T → (i : Fin q) → Fin (i.val + 1) → κ) (h : ∀ t i j a b, c t i ≠ 0 → c t j ≠ 0 → col t i a = col t j b → a.val = b.val) (h' : ∀ t i j a b, c' t i ≠ 0 → c' t j ≠ 0 → col' t i a = col' t j b → a.val = b.val) : blockPoly R c c' col col' ≠ 0

theorem **pderiv_xVar_colSubst** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (a : Fin p) (l : κ') (P : Poly R T p q κ) : pderiv (xVar t a l) (colSubst A B P) = ∑ k : κ, A t k l • colSubst A B (pderiv (xVar t a k) P)

theorem **pderiv_yVar_colSubst** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (b : Fin q) (l : κ') (P : Poly R T p q κ) : pderiv (yVar t b l) (colSubst A B P) = ∑ k : κ, B t l k • colSubst A B (pderiv (yVar t b k) P)

theorem **laplacian_colSubst** (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) (t : T) (a : Fin p) (b : Fin q) (P : Poly R T p q κ) : laplacian t a b (colSubst A B P) = ∑ k : κ, ∑ k' : κ, (A t * B t) k k' • colSubst A B (pderiv (xVar t a k) (pderiv (yVar t b k') P))

theorem **laplacian_colSubst_of_mul_eq_one** [DecidableEq κ] (A : T → Matrix κ κ' R) (B : T → Matrix κ' κ R) {t : T} (h : A t * B t = 1) (a : Fin p) (b : Fin q) (P : Poly R T p q κ) : laplacian t a b (colSubst A B P) = colSubst A B (laplacian t a b P)

theorem **IsHarmonic.colSubst** [DecidableEq κ] {A : T → Matrix κ κ' R} {B : T → Matrix κ' κ R} (h : ∀ t, A t * B t = 1) {P : Poly R T p q κ} (hP : IsHarmonic P) : IsHarmonic (colSubst A B P)

theorem **laplacian_colAct** (g : T → GL κ R) (t : T) (a : Fin p) (b : Fin q) (P : Poly R T p q κ) : laplacian t a b (colAct R T p q κ g P) = colAct R T p q κ g (laplacian t a b P)

theorem **IsHarmonic.colAct** (g : T → GL κ R) {P : Poly R T p q κ} (hP : IsHarmonic P) : IsHarmonic (colAct R T p q κ g P)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsProjOverCompOfIsClosedImmersion

theorem **isUnit_of_basicOpen_eq** {U : Y.Opens} (s : Γ(Y, U)) (h : Y.basicOpen s = U) : IsUnit s

theorem **fromSpec_appLE_self** {U : Y.Opens} (hU : IsAffineOpen U) : hU.fromSpec.appLE U ⊤ hU.fromSpec_preimage_self.ge = (Scheme.ΓSpecIso _).inv

theorem **mem_spOpen** {a : A} {x : Sp A} : x ∈ spOpen a ↔ a ∉ x.asIdeal

theorem **iSup_spOpen_eq_top_iff** {ι : Type} {v : ι → A} : ⨆ i, spOpen (v i) = ⊤ ↔ Ideal.span (Set.range v) = ⊤

theorem **resG_fnOn** {W V : (Sp A).Opens} (h : V ≤ W) (a : A) : resG _ h (fnOn W a) = fnOn V a

theorem **basicOpen_fnOn** (W : (Sp A).Opens) (a : A) : (Sp A).basicOpen (fnOn W a) = W ⊓ spOpen a

theorem **fnOn_top_bijective** : Function.Bijective (fnOn (A

theorem **fromSpec_eq_spMap_fnOn** {W : (Sp A).Opens} (hW : IsAffineOpen W) : hW.fromSpec = spMap (R'

theorem **map_irrelevant_eval₂Hom** (c : R →+* A) {v : Fin (N + 1) → A} (hv : Ideal.span (Set.range v) = ⊤) : (HomogeneousIdeal.irrelevant (grading R N)).toIdeal.map (eval₂Hom c v) = ⊤

@[reassoc] theorem **pt_str** : pt c v hv ≫ PSpace.str R N = spMap c

theorem **pt_preimage_basicOpen** {n : ℕ} (hn : 0 < n) {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : pt c v hv ⁻¹ᵁ Proj.basicOpen (grading R N) f = spOpen (eval₂ c v f)

theorem **pt_preimage_chart** (i : Fin (N + 1)) : pt c v hv ⁻¹ᵁ chart R N i = spOpen (v i)

theorem **span_range_comp_eq_top** (ψ : A →+* B) : Ideal.span (Set.range (ψ ∘ v)) = ⊤

@[reassoc] theorem **spMap_pt** (ψ : A →+* B) : spMap ψ ≫ pt c v hv = pt (ψ.comp c) (ψ ∘ v) (span_range_comp_eq_top v hv ψ)

@[reassoc] theorem **pt_baseChange** {R₀ : Type} [CommRing R₀] (φ : R₀ →+* R) : pt c v hv ≫ baseChange N φ = pt (c.comp φ) v hv

theorem **fromSpec_chart** (i : Fin (N + 1)) : (isAffineOpen_chart R N i).fromSpec = Spec.map (Proj.awayToSection (grading R N) (X i)) ≫ Proj.awayι (grading R N) (X i) (X_mem R N i) one_pos

theorem **awayToSection_ratioAway** (i j : Fin (N + 1)) : (Proj.awayToSection (grading R N) (X i)).hom (ratioAway R N i j) = ratio R N i j

theorem **pullRatio_spMap_awayι** (i j : Fin (N + 1)) (ρ : HomogeneousLocalization.Away (grading R N) (X i) →+* A) (h : ⊤ ≤ (spMap ρ ≫ Proj.awayι (grading R N) (X i) (X_mem R N i) one_pos) ⁻¹ᵁ chart R N i) : pullRatio (spMap ρ ≫ Proj.awayι (grading R N) (X i) (X_mem R N i) one_pos) ⊤ i h j = fnOn ⊤ (ρ (ratioAway R N i j))

theorem **pullRatio_pt_top_mul** (i j : Fin (N + 1)) (hu : IsUnit (v i)) (h : ⊤ ≤ pt c v hv ⁻¹ᵁ chart R N i) : pullRatio (pt c v hv) ⊤ i h j * fnOn ⊤ (v i) = fnOn ⊤ (v j)

theorem **pullRatio_pt_mul** {W : (Sp A).Opens} (i j : Fin (N + 1)) (h : W ≤ pt c v hv ⁻¹ᵁ chart R N i) : pullRatio (pt c v hv) W i h j * fnOn W (v i) = fnOn W (v j)

theorem **eq_zero_of_fnOn_eq_zero** {ι : Type} {v : ι → A} (hv : Ideal.span (Set.range v) = ⊤) {a : A} (h : ∀ i, fnOn (spOpen (v i)) a = 0) : a = 0

theorem **exists_unit_smul_of_mul_eq** {ι : Type} [Finite ι] {v w : ι → A} (hv : Ideal.span (Set.range v) = ⊤) (hw : Ideal.span (Set.range w) = ⊤) (h : ∀ i j, v i * w j = v j * w i) : ∃ u : Aˣ, w = u • v

theorem **span_range_units_smul** {ι : Type} {v : ι → A} (hv : Ideal.span (Set.range v) = ⊤) (u : Aˣ) : Ideal.span (Set.range (u • v)) = ⊤

theorem **pt_units_smul** (c : R →+* A) {v : Fin (N + 1) → A} (hv : Ideal.span (Set.range v) = ⊤) (u : Aˣ) : pt c (u • v) (span_range_units_smul hv u) = pt c v hv

theorem **pt_eq_pt_iff** (c : R →+* A) {v w : Fin (N + 1) → A} (hv : Ideal.span (Set.range v) = ⊤) (hw : Ideal.span (Set.range w) = ⊤) : pt c v hv = pt c w hw ↔ ∃ u : Aˣ, w = u • v

theorem **eq_pt** {c : R →+* A} {v : Fin (N + 1) → A} (hv : Ideal.span (Set.range v) = ⊤) {φ : Sp A ⟶ PSpace R N} (hφ : φ ≫ PSpace.str R N = spMap c) (h : ∀ i, spOpen (v i) ≤ φ ⁻¹ᵁ chart R N i) (H : ∀ i j, pullRatio φ _ i (h i) j * fnOn _ (v i) = fnOn _ (v j)) : φ = pt c v hv

theorem **exists_eq_pt_of_isGen** (c : R →+* A) (φ : Sp A ⟶ PSpace R N) (hφ : φ ≫ PSpace.str R N = spMap c) (t : ((O1 R N).pull φ).Gen) (ht : t.IsGen) : ∃ (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤), φ = pt c v hv

theorem **exists_eq_pt** [Subsingleton (CommRing.Pic A)] (c : R →+* A) (φ : Sp A ⟶ PSpace R N) (hφ : φ ≫ PSpace.str R N = spMap c) : ∃ (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤), φ = pt c v hv

@[simp] theorem **ptEquiv_mk** [Subsingleton (CommRing.Pic A)] (c : R →+* A) (v : Unimodular A N) : (ptEquiv c N (Quotient.mk _ v)).1 = pt c v.1 v.2

theorem **span_range_eq_top_iff_of_isLocalRing** [IsLocalRing A] {ι : Type} {v : ι → A} : Ideal.span (Set.range v) = ⊤ ↔ ∃ i, IsUnit (v i)

theorem **span_range_eq_top_iff_ne_zero** {K : Type} [Field K] {ι : Type} {v : ι → K} : Ideal.span (Set.range v) = ⊤ ↔ v ≠ 0

theorem **pt_eq_pt_of_injective** (c : R →+* A) {ψ : A →+* B} (hψ : Function.Injective ψ) {v w : Fin (N + 1) → A} (hv : Ideal.span (Set.range v) = ⊤) (hw : Ideal.span (Set.range w) = ⊤) (h : spMap ψ ≫ pt c v hv = spMap ψ ≫ pt c w hw) : pt c v hv = pt c w hw

theorem **exists_smul_eq_primitive** {ι : Type} [Finite ι] {v : ι → K} (hv : v ≠ 0) : ∃ (u : Kˣ) (w : ι → V), (∃ i, w i = 1) ∧ algebraMap V K ∘ w = u • v

theorem **exists_primitive_eq** (c : R →+* V) (φ : Sp K ⟶ PSpace R N) (hφ : φ ≫ PSpace.str R N = spMap ((algebraMap V K).comp c)) : ∃ (w : Fin (N + 1) → V) (hw : Ideal.span (Set.range w) = ⊤), (∃ i, w i = 1) ∧ spMap (algebraMap V K) ≫ pt c w hw = φ

theorem **existsUnique_extend** (c : R →+* V) (φ : Sp K ⟶ PSpace R N) (hφ : φ ≫ PSpace.str R N = spMap ((algebraMap V K).comp c)) : ∃! ψ : Sp V ⟶ PSpace R N, ψ ≫ PSpace.str R N = spMap c ∧ spMap (algebraMap V K) ≫ ψ = φ

theorem **existsUnique_extend_of_isProjOver** {f : Y ⟶ Sp R} (hf : IsProjOver f) (c : R →+* V) (φ : Sp K ⟶ Y) (hφ : φ ≫ f = spMap ((algebraMap V K).comp c)) : ∃! ψ : Sp V ⟶ Y, ψ ≫ f = spMap c ∧ spMap (algebraMap V K) ≫ ψ = φ

theorem **exists_mem_of_surjective** (hg : Function.Surjective g) {n : ℕ} {b : T} (hb : b ∈ 𝓑 n) : ∃ a ∈ 𝓐 n, g a = b

theorem **irrelevant_le_map_of_surjective** (hg : Function.Surjective g) : HomogeneousIdeal.irrelevant 𝓑 ≤ (HomogeneousIdeal.irrelevant 𝓐).map g

theorem **awayMap_surjective** (hg : Function.Surjective g) {d : ℕ} {s : S} (hs : s ∈ 𝓐 d) : Function.Surjective (HomogeneousLocalization.Away.map g s)

theorem **isClosedImmersion_projMap** (hg : Function.Surjective g) (h : HomogeneousIdeal.irrelevant 𝓑 ≤ (HomogeneousIdeal.irrelevant 𝓐).map g) : IsClosedImmersion (Proj.map g h)

theorem **exists_lift_of_cover** {X : Scheme.{0}} (ι : Z ⟶ X) [IsClosedImmersion ι] (φ : Y ⟶ X) {κ : Type} (U : κ → X.Opens) (hU : ∀ i, IsAffineOpen (U i)) (hc : ⊔ i, U i = ⊤) (H : ∀ i s, (ι.app (U i)).hom s = 0 → (φ.app (U i)).hom s = 0) : ∃ ψ : Y ⟶ Z, ψ ≫ ι = φ

theorem **quotHom_eq_zero_iff** {f : MvPolynomial (Fin (N + 1)) R} : quotHom I f = 0 ↔ f ∈ I

theorem **isProjOver_vPlus** : IsProjOver (vPlusι I ≫ PSpace.str R N)

theorem **vPlusι_app_formOn_eq_zero_iff** (i : Fin (N + 1)) {n : ℕ} {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : ((vPlusι I).app (chart R N i)).hom (formOn (chart R N i) i le_rfl n f hf) = 0 ↔ ∃ m, X i ^ m * f ∈ I

theorem **exists_comp_vPlusι_iff** (φ : Y ⟶ PSpace R N) : (∃ ψ : Y ⟶ VPlus I, ψ ≫ vPlusι I = φ) ↔ ∀ (i : Fin (N + 1)) (n : ℕ) (f : MvPolynomial (Fin (N + 1)) R) (hf : f.IsHomogeneous n), f ∈ I → (φ.app (chart R N i)).hom (formOn (chart R N i) i le_rfl n f hf) = 0

theorem **appLE_pt_formOn_mul** (c : R →+* A) (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤) {W : (Sp A).Opens} (i : Fin (N + 1)) (h : W ≤ pt c v hv ⁻¹ᵁ chart R N i) {n : ℕ} {f : MvPolynomial (Fin (N + 1)) R} (hf : f.IsHomogeneous n) : ((pt c v hv).appLE (chart R N i) W h).hom (formOn (chart R N i) i le_rfl n f hf) * fnOn W (v i) ^ n = fnOn W (eval₂ c v f)

theorem **exists_comp_vPlusι_eq_pt_iff** (c : R →+* A) (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤) : (∃ ψ : Sp A ⟶ VPlus I, ψ ≫ vPlusι I = pt c v hv) ↔ ∀ f ∈ I, eval₂ c v f = 0

theorem **mem_range_vPlusι_iff** (x : PSpace R N) : x ∈ Set.range (vPlusι I) ↔ I ≤ x.asHomogeneousIdeal

theorem **range_vPlusι** : Set.range (vPlusι I) = ProjectiveSpectrum.zeroLocus (grading R N) I

theorem **exists_eq_pt_vPlus** [Subsingleton (CommRing.Pic A)] (c : R →+* A) (ψ : Sp A ⟶ VPlus I) (hψ : ψ ≫ vPlusι I ≫ PSpace.str R N = spMap c) : ∃ (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤), (∀ f ∈ I, eval₂ c v f = 0) ∧ ψ ≫ vPlusι I = pt c v hv

theorem **isProjOver_comp_of_isClosedImmersion** (g : Z ⟶ Y) [IsClosedImmersion g] {f : Y ⟶ Sp R} (h : IsProjOver f) : IsProjOver (g ≫ f)

theorem **isProjOver_of_isPullback_spMap** {R' : Type} [CommRing R'] (c : R →+* R') {g : Z ⟶ Y} {f' : Z ⟶ Sp R'} {f : Y ⟶ Sp R} (H : IsPullback g f' f (spMap c)) (h : IsProjOver f) : IsProjOver f'

theorem **isProjOver_pullback_snd** {R' : Type} [CommRing R'] (c : R →+* R') {f : Y ⟶ Sp R} (h : IsProjOver f) : IsProjOver (pullback.snd f (spMap c))

theorem **isProjOver_pullback** {f : Y ⟶ Sp R} {g : Z ⟶ Sp R} (hf : IsProjOver f) (hg : IsProjOver g) : IsProjOver (pullback.fst f g ≫ f)

theorem **killHom_surjective** : Function.Surjective (killHom R he)

theorem **linearι_preimage_chart** (i : Fin (N + 1)) : linearι R he ⁻¹ᵁ chart R M (e i) = chart R N i

theorem **linearι_preimage_chart_of_notMem** {k : Fin (M + 1)} (hk : k ∉ Set.range e) : linearι R he ⁻¹ᵁ chart R M k = ⊥

@[reassoc] theorem **linearι_str** : linearι R he ≫ PSpace.str R M = PSpace.str R N

theorem **isProjOver_sigmaDesc** {ι : Type} [Finite ι] {T : ι → Scheme.{0}} (f : ∀ i, T i ⟶ Sp R) (h : ∀ i, IsProjOver (f i)) : IsProjOver (Sigma.desc f)

theorem **isProjOver_coprodDesc** {f : Y ⟶ Sp R} {g : Z ⟶ Sp R} (hf : IsProjOver f) (hg : IsProjOver g) : IsProjOver (coprod.desc f g)

theorem **isProjOver_of_disjoint_cover** {ι : Type} [Finite ι] (f : Y ⟶ Sp R) (U : ι → Y.Opens) (hU : ⨆ i, U i = ⊤) (hd : Pairwise fun i j => Disjoint (U i) (U j)) (h : ∀ i, IsProjOver ((U i).ι ≫ f)) : IsProjOver f

theorem **pt_linearι** {M : ℕ} {e : Fin (N + 1) → Fin (M + 1)} (he : Function.Injective e) (c : R →+* A) (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤) (hw : Ideal.span (Set.range (Function.extend e v 0)) = ⊤) : pt c v hv ≫ linearι R he = pt c (Function.extend e v 0) hw

theorem **points_and_stability** : (∀ (A : Type) [CommRing A] [Algebra R A], (∀ (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤), pt (algebraMap R A) v hv ≫ PSpace.str R N = spMap (algebraMap R A) ∧ (∀ i, pt (algebraMap R A) v hv ⁻¹ᵁ chart R N i = PrimeSpectrum.basicOpen (v i)) ∧ (∀ (B : Type) [CommRing B] [Algebra R B] (ψ : A →ₐ[R] B), ∃ h : Ideal.span (Set.range (ψ ∘ v)) = ⊤, spMap (ψ : A →+* B) ≫ pt (algebraMap R A) v hv = pt (algebraMap R B) (ψ ∘ v) h) ∧ ∀ (w : Fin (N + 1) → A) (hw : Ideal.span (Set.range w) = ⊤), pt (algebraMap R A) v hv = pt (algebraMap R A) w hw ↔ ∃ u : Aˣ, w = u • v) ∧ ((IsLocalRing A ∨ Subsingleton (CommRing.Pic A)) → ∀ φ : Sp A ⟶ PSpace R N, φ ≫ PSpace.str R N = spMap (algebraMap R A) → ∃ (v : Fin (N + 1) → A) (hv : Ideal.span (Set.range v) = ⊤), φ = pt (algebraMap R A) v hv)) ∧ (∀ (V K : Type) [CommRing V] [IsDomain V] [ValuationRing V] [Field K] [Algebra

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsProjOverIffIsProperAndExistsIsAmpleLB

theorem **IsAmpleLB.pull_of_isImmersion_of_compactSpace** (h : IsAmpleLB L) (f : Y ⟶ X) [IsImmersion f] [CompactSpace Y] : IsAmpleLB (L.pull f)

theorem **IsAmpleLB.pull_of_isImmersion** (h : IsAmpleLB L) (f : Y ⟶ X) [IsImmersion f] [QuasiCompact f] : IsAmpleLB (L.pull f)

theorem **IsAmpleLB.pull_of_isFinite** (h : IsAmpleLB L) (f : Y ⟶ X) [IsFinite f] : IsAmpleLB (L.pull f)

theorem **IsAmpleLB.pull_of_isClosedImmersion** (h : IsAmpleLB L) (f : Y ⟶ X) [IsClosedImmersion f] : IsAmpleLB (L.pull f)

theorem **IsAmpleLB.pull_of_isPullback** (h : IsAmpleLB L) {S' : Scheme.{0}} {π : Y ⟶ X} {f' : Y ⟶ S'} {f : X ⟶ S} {g : S' ⟶ S} (H : IsPullback π f' f g) [IsAffineHom g] : IsAmpleLB (L.pull π)

theorem **isAmpleLB_congr** {M : LB X} (hL : L.IsLB) (hM : M.IsLB) (H : LBIso L M) : IsAmpleLB L ↔ IsAmpleLB M

theorem **exists_sec_div_mul** (hA : A.IsLB) (hB : B.IsLB) (t : (LB.div B A).Sec) (s : A.Sec) : ∃ u : B.Sec, ∀ (y : X) k l, y ∈ B.U k → y ∈ A.U l → gs (𝟙 X) y _ (u.f k) = gs (𝟙 X) y (B.U k ⊓ A.U l) (t.f (k, l)) * gs (𝟙 X) y _ (s.f l)

theorem **exists_sec_nonzeroLocus_eq_inf_div** (hA : A.IsLB) (hB : B.IsLB) (t : (LB.div B A).Sec) (s : A.Sec) : ∃ u : B.Sec, nonzeroLocus u = nonzeroLocus t ⊓ nonzeroLocus s

theorem **exists_secMaps** (hA : A.IsLB) (hB : B.IsLB) (w : LB.Iso A B) (hw : w.IsGen) (s : A.Sec) : ∃ u : B.Sec, SecMaps w s u

theorem **exists_sec_pow_nonzeroLocus_eq_inf** (hL : L.IsLB) (hM : M.IsLB) (hN : N.IsLB) (H : LBIso M (LB.div N L)) {n : ℕ} (s : (lbPow L n).Sec) (t : (lbPow M n).Sec) : ∃ u : (lbPow N n).Sec, nonzeroLocus u = nonzeroLocus t ⊓ nonzeroLocus s

theorem **IsAmpleLB.tensor_isGenerated** (hL : IsAmpleLB L) (hM : IsGenerated M) (hN : N.IsLB) (H : ∃ t : (LB.div (LB.div N L) M).Gen, t.IsGen) : IsAmpleLB N

theorem **IsAmpleLB.tensor** (hL : IsAmpleLB L) (hM : IsAmpleLB M) (hN : N.IsLB) (H : ∃ t : (LB.div (LB.div N L) M).Gen, t.IsGen) : IsAmpleLB N

theorem **IsAmpleLB.box** {L : LB X} {M : LB Y} {N : LB Z} (hL : IsAmpleLB L) (hM : IsAmpleLB M) (hN : N.IsLB) {p : Z ⟶ X} {q : Z ⟶ Y} {f : X ⟶ S} {g : Y ⟶ S} (P : IsPullback p q f g) [S.IsSeparated] (H : ∃ t : (LB.div (LB.div N (L.pull p)) (M.pull q)).Gen, t.IsGen) : IsAmpleLB N

theorem **IsAmpleLB.linComb_of_isWideProduct** [IsAffine S] {L : LB X} (hL : IsAmpleLB L) {j : ℕ} {π : X ⟶ S} {π' : Y ⟶ S} {ρ : Fin j → (Y ⟶ X)} (hw : Segre.IsWideProduct π' (fun _ => π) ρ) : IsAmpleLB (LB.LinComb L (K

theorem **isWideProduct_of_isLimit** {G' G : Over S} {j : ℕ} {pr : Fin j → (G' ⟶ G)} (lim : IsLimit (Fan.mk G' pr)) : Segre.IsWideProduct G'.hom (fun _ => G.hom) (fun i => (pr i).left)

theorem **IsAmpleLB.boxLB** [IsAffine S] {G' G : Over S} {j : ℕ} {pr : Fin j → (G' ⟶ G)} (lim : IsLimit (Fan.mk G' pr)) {L : LB G.left} (hL : IsAmpleLB L) : IsAmpleLB (boxLB L pr)

theorem **isClosedImmersion_morphismRestrict_of_surjective** (φ : X ⟶ Y) {U : Y.Opens} {W : X.Opens} (hU : IsAffineOpen U) (hW : IsAffineOpen W) (e : φ ⁻¹ᵁ U = W) (h : Function.Surjective (φ.appLE U W e.ge).hom) : IsClosedImmersion (φ ∣_ U)

theorem **exists_generators** (f : X ⟶ Sp R) [LocallyOfFiniteType f] {W : X.Opens} (hW : IsAffineOpen W) : ∃ (m : ℕ) (g : Fin m → Γ(X, W)), Function.Surjective (MvPolynomial.eval₂Hom (baseRingHom f W).hom g)

theorem **exists_sec_eq_fn_mul_pow** [CompactSpace X] [QuasiSeparatedSpace X] (hL : L.IsLB) {n : ℕ} (s : (lbPow L n).Sec) (g : Γ(X, nonzeroLocus s)) : ∃ N : ℕ, ∀ k, N ≤ k → ∃ u : (lbPow L (n * k)).Sec, ∀ J, resG X (inf_le_right : nonzeroLocus s ⊓ (lbPow L (n * k)).U J ≤ _) (u.f J) = resG X inf_le_left g * resG X inf_le_right ((secPow s k).f J)

theorem **isImmersion_toProj** (f : X ⟶ Sp R) {M : ℕ} (t : Fin (M + 1) → L.Sec) (ht : ⨆ i, nonzeroLocus (t i) = ⊤) (H : ∀ x : X, ∃ (i : Fin (M + 1)) (W : X.Opens) (e : nonzeroLocus (t i) = W), x ∈ W ∧ IsAffineOpen W ∧ ∃ (ι : Type) (a : ι → Fin (M + 1)), Function.Surjective (MvPolynomial.eval₂Hom (baseRingHom f W).hom fun j => resG X e.ge (secRatio (t i) (t (a j))))) : IsImmersion (toProj f t ht)

theorem **IsAmpleLB.exists_isImmersion_mul** (h : IsAmpleLB L) (f : X ⟶ Sp R) [LocallyOfFiniteType f] : ∃ n, 1 ≤ n ∧ ∃ N, ∀ k, N ≤ k → ∃ (M : ℕ) (t : Fin (M + 1) → (lbPow L (n * k)).Sec) (ht : ⨆ i, nonzeroLocus (t i) = ⊤), IsImmersion (toProj f t ht)

theorem **IsAmpleLB.exists_isImmersion** (h : IsAmpleLB L) (f : X ⟶ Sp R) [LocallyOfFiniteType f] : ∃ n, 1 ≤ n ∧ ∃ (M : ℕ) (t : Fin (M + 1) → (lbPow L n).Sec) (ht : ⊔ i, nonzeroLocus (t i) = ⊤), IsImmersion (toProj f t ht)

theorem **IsAmpleLB.isQProjOver** (h : IsAmpleLB L) (f : X ⟶ Sp R) [LocallyOfFiniteType f] : IsQProjOver f

theorem **isClosedImmersion_of_isProper** {M : ℕ} (φ : X ⟶ PSpace R M) [IsImmersion φ] [IsProper (φ ≫ PSpace.str R M)] : IsClosedImmersion φ

theorem **IsAmpleLB.exists_veryAmpleBy_mul** (h : IsAmpleLB L) (f : X ⟶ Sp R) [IsProper f] : ∃ n, 1 ≤ n ∧ ∃ N, ∀ k, N ≤ k → ∃ (M : ℕ) (t : Fin (M + 1) → (lbPow L (n * k)).Sec), VeryAmpleBy f (lbPow L (n * k)) t

theorem **IsAmpleLB.exists_veryAmpleBy** (h : IsAmpleLB L) (f : X ⟶ Sp R) [IsProper f] : ∃ n, 1 ≤ n ∧ ∃ (M : ℕ) (t : Fin (M + 1) → (lbPow L n).Sec), VeryAmpleBy f (lbPow L n) t

theorem **IsAmpleLB.isProjOver** (h : IsAmpleLB L) (f : X ⟶ Sp R) [IsProper f] : IsProjOver f

theorem **exists_isAmpleLB_of_isProjOver** {f : X ⟶ Sp R} (h : IsProjOver f) : ∃ L : LB X, IsAmpleLB L

theorem **isProjOver_iff_isProper_and_exists_isAmpleLB** (f : X ⟶ Sp R) : IsProjOver f ↔ IsProper f ∧ ∃ L : LB X, IsAmpleLB L

theorem **isProjOver_comp_of_isFinite** (g : Y ⟶ X) [IsFinite g] {f : X ⟶ Sp R} (h : IsProjOver f) : IsProjOver (g ≫ f)

theorem **exists_veryAmpleBy_of_lbIso** {f : X ⟶ Sp R} {B : LB X} (hL : L.IsLB) (hB : B.IsLB) (H : LBIso L B) {M : ℕ} {s : Fin (M + 1) → L.Sec} (h : VeryAmpleBy f L s) : ∃ t : Fin (M + 1) → B.Sec, VeryAmpleBy f B t

theorem **isAmpleLB_of_veryAmpleBy** {f : X ⟶ Sp R} {M : ℕ} {s : Fin (M + 1) → L.Sec} (h : VeryAmpleBy f L s) : IsAmpleLB L

theorem **relAmple_of_isAmpleLB** (A : Over S) [IsProper A.hom] {L : LB A.left} (hL : L.IsLB) (h : ∀ U : S.Opens, IsAffineOpen U → IsAmpleLB (L.pull (A.hom ⁻¹ᵁ U).ι)) : RelAmple A L

theorem **relAmple_iff_isAmpleLB** (A : Over S) [IsProper A.hom] {L : LB A.left} (hL : L.IsLB) : RelAmple A L ↔ ∀ U : S.Opens, IsAffineOpen U → IsAmpleLB (L.pull (A.hom ⁻¹ᵁ U).ι)

theorem **relAmple_iff_isAmpleLB_of_isAffine** [IsAffine S] (A : Over S) [IsProper A.hom] {L : LB A.left} (hL : L.IsLB) : RelAmple A L ↔ IsAmpleLB L

theorem **exists_gen_div_div_tw** (hL : L.IsLB) (hM : M.IsLB) : ∃ t : (LB.div (LB.div (tw L M 1) L) M).Gen, t.IsGen

theorem **IsAmpleLB.tw_one** (hL : IsAmpleLB L) (hM : IsGenerated M) : IsAmpleLB (tw L M 1)

theorem **isProper_of_veryAmpleBy** {R : Type} [CommRing R] {f : X ⟶ Sp R} {L : LB X} {M : ℕ} {s : Fin (M + 1) → L.Sec} (h : VeryAmpleBy f L s) : IsProper f

theorem **relAmple_of_isAbelianScheme** (A : Over S) [GrpObj A] (hA : IsAbelianScheme A) {L : LB A.left} (hL : L.IsLB) (h : ∀ U : S.Opens, IsAffineOpen U → IsAmpleLB (L.pull (A.hom ⁻¹ᵁ U).ι)) : RelAmple A L

theorem **isAmpleLB_basic** : (∀ {X : Scheme.{0}} {L M : LB X}, L.IsLB → M.IsLB → LBIso L M → (IsAmpleLB L ↔ IsAmpleLB M)) ∧ (∀ {X : Scheme.{0}} {L : LB X} {n : ℕ}, L.IsLB → 1 ≤ n → (IsAmpleLB (lbPow L n) ↔ IsAmpleLB L)) ∧ (∀ {X Y : Scheme.{0}} {L : LB X} (f : Y ⟶ X), IsAffineHom f → IsAmpleLB L → IsAmpleLB (L.pull f)) ∧ (∀ {X Y S S' : Scheme.{0}} {L : LB X} {π : Y ⟶ X} {f' : Y ⟶ S'} {f : X ⟶ S} {g : S' ⟶ S}, IsPullback π f' f g → IsAffineHom g → IsAmpleLB L → IsAmpleLB (L.pull π)) ∧ (∀ {X Y : Scheme.{0}} {L : LB X} (f : Y ⟶ X), IsImmersion f → QuasiCompact f → IsAmpleLB L → IsAmpleLB (L.pull f)) ∧ (∀ {X : Scheme.{0}} {L M N : LB X}, IsAmpleLB L → IsGenerated M → N.IsLB → (∃ t : (LB.div (LB.div N L) M).Gen, t.IsGen) → IsAmpleLB N) ∧ (∀ (R : Type) [CommRing R] (N : ℕ), IsAmpleLB (O1 R N)) ∧ (∀ {R : Type} [CommRing R] {X : Scheme.{0}} {L : LB X} (f : X ⟶ Sp R), LocallyOfFiniteType f → IsAmpleL

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsProjOverMorphismRestrictOfRelAmple

theorem **isUnit_resG_sec** (a : L.Sec) (k : L.ι) {W : Y.Opens} (ha : W ≤ nonzeroLocus a) (hk : W ≤ L.U k) : IsUnit (resG Y hk (a.f k))

theorem **le_nonzeroLocus_of_isUnit** (a : L.Sec) (k : L.ι) {W : Y.Opens} (hk : W ≤ L.U k) (h : IsUnit (resG Y hk (a.f k))) : W ≤ nonzeroLocus a

theorem **ext_charts** (a : L.Sec) {W : Y.Opens} (ha : W ≤ nonzeroLocus a) {r r' : Γ(Y, W)} (h : ∀ k, resG Y (inf_le_left : W ⊓ L.U k ≤ W) r = resG Y inf_le_left r') : r = r'

theorem **exists_secRatio** (a b : L.Sec) : ∃ r : Γ(Y, nonzeroLocus a), ∀ k, resG Y (inf_le_right : nonzeroLocus a ⊓ L.U k ≤ L.U k) (b.f k) = resG Y inf_le_left r * resG Y inf_le_right (a.f k)

theorem **secRatio_spec** (a b : L.Sec) (k : L.ι) : resG Y (inf_le_right : nonzeroLocus a ⊓ L.U k ≤ L.U k) (b.f k) = resG Y inf_le_left (secRatio a b) * resG Y inf_le_right (a.f k)

theorem **secRatio_on** (a b : L.Sec) (k : L.ι) {W : Y.Opens} (ha : W ≤ nonzeroLocus a) (hk : W ≤ L.U k) : resG Y hk (b.f k) = resG Y ha (secRatio a b) * resG Y hk (a.f k)

theorem **eq_resG_secRatio** (a b : L.Sec) {W : Y.Opens} (ha : W ≤ nonzeroLocus a) {r : Γ(Y, W)} (h : ∀ k, resG Y (inf_le_right : W ⊓ L.U k ≤ L.U k) (b.f k) = resG Y inf_le_left r * resG Y inf_le_right (a.f k)) : r = resG Y ha (secRatio a b)

theorem **eq_secRatio** (a b : L.Sec) {r : Γ(Y, nonzeroLocus a)} (h : ∀ k, resG Y (inf_le_right : nonzeroLocus a ⊓ L.U k ≤ L.U k) (b.f k) = resG Y inf_le_left r * resG Y inf_le_right (a.f k)) : r = secRatio a b

@[simp] theorem **secRatio_self** (a : L.Sec) : secRatio a a = 1

theorem **secRatio_mul_secRatio** (a b c : L.Sec) {W : Y.Opens} (ha : W ≤ nonzeroLocus a) (hb : W ≤ nonzeroLocus b) : resG Y ha (secRatio a b) * resG Y hb (secRatio b c) = resG Y ha (secRatio a c)

theorem **basicOpen_secRatio** (a b : L.Sec) : Y.basicOpen (secRatio a b) = nonzeroLocus a ⊓ nonzeroLocus b

theorem **isCoordFamily_secRatio** {s : Fin (M + 1) → L.Sec} (hs : ⊔ i, nonzeroLocus (s i) = ⊤) : IsCoordFamily (fun i => nonzeroLocus (s i)) (fun i j => secRatio (s i) (s j)) where

@[reassoc] theorem **toProj_str** : toProj f s hs ≫ PSpace.str R M = f

theorem **toProj_preimage_chart** (i : Fin (M + 1)) : toProj f s hs ⁻¹ᵁ chart R M i = nonzeroLocus (s i)

theorem **pullRatio_toProj** (i j : Fin (M + 1)) : pullRatio (toProj f s hs) (nonzeroLocus (s i)) i (toProj_preimage_chart f s hs i).ge j = secRatio (s i) (s j)

theorem **pullRatio_toProj_on** (i j : Fin (M + 1)) {W : Y.Opens} (h : W ≤ toProj f s hs ⁻¹ᵁ chart R M i) : pullRatio (toProj f s hs) W i h j = resG Y (h.trans (toProj_preimage_chart f s hs i).le) (secRatio (s i) (s j))

theorem **sec_eq_pullRatio_toProj_mul** (i j : Fin (M + 1)) (k : L.ι) {W : Y.Opens} (h : W ≤ toProj f s hs ⁻¹ᵁ chart R M i) (hk : W ≤ L.U k) : resG Y hk ((s j).f k) = pullRatio (toProj f s hs) W i h j * resG Y hk ((s i).f k)

theorem **eq_toProj_of_locally** {φ : Y ⟶ PSpace R M} (hφ : φ ≫ PSpace.str R M = f) (H : ∀ y : Y, ∃ (W : Y.Opens) (_ : y ∈ W) (k : L.ι) (hk : W ≤ L.U k) (i : Fin (M + 1)) (hi : W ≤ φ ⁻¹ᵁ chart R M i) (u : Γ(Y, W)), ∀ j, resG Y hk ((s j).f k) = u ∗ pullRatio φ W i hi j) : φ = toProj f s hs

theorem **eq_toProj_of_pullRatio** {φ : Y ⟶ PSpace R M} (hφ : φ ≫ PSpace.str R M = f) (h : ∀ i, nonzeroLocus (s i) ≤ φ ⁻¹ᵁ chart R M i) (H : ∀ i j, pullRatio φ (nonzeroLocus (s i)) i (h i) j = secRatio (s i) (s j)) : φ = toProj f s hs

theorem **resG_secPull_coord** (φ : Y ⟶ PSpace R M) (j : Fin (M + 1)) (i : PUnit → Fin (M + 1)) {W : Y.Opens} (h : W ≤ ((O1 R M).pull φ).U i) : resG Y h ((secPull (coord R M j) φ).f i) = pullRatio φ W (i ⟨⟩) (h.trans (pull_U (O1 R M) φ i).le) j

theorem **restrict_pullback_O1_transition** (φ : Y ⟶ PSpace R M) (i j : PUnit → Fin (M + 1)) {W : Y.Opens} (hi : W ≤ ((O1 R M).pull φ).U i) (hj : W ≤ ((O1 R M).pull φ).U j) : resG Y (le_inf hi hj) (((O1 R M).pull φ).g i j).val = pullRatio φ W (i ⟨⟩) (hi.trans (pull_U (O1 R M) φ i).le) (j ⟨⟩)

theorem **iSup_nonzeroLocus_secPull** {s : Fin (M + 1) → L.Sec} (hs : ⨆ i, nonzeroLocus (s i) = ⊤) (g : Z ⟶ Y) : ⨆ i, nonzeroLocus (secPull (s i) g) = ⊤

theorem **pullback_O1_toProj_chart** (i : PUnit → Fin (M + 1)) : ((O1 R M).pull (toProj f s hs)).U i = nonzeroLocus (s (i ⟨⟩))

theorem **val_toProjIso** (k : L.ι) (i : PUnit → Fin (M + 1)) : (@id (Γ(Y, L.U k ⊓ ((O1 R M).pull (toProj f s hs)).U i)ˣ) ((toProjIso f s hs).s (k, i))).val = resG Y inf_le_left ((s (i ⟨⟩))).f k)

theorem **secMaps_toProjIso** (j : Fin (M + 1)) : SecMaps (toProjIso f s hs) (secPull (coord R M j) (toProj f s hs)) (s j)

theorem **toProjIso_isGen** : (toProjIso f s hs).IsGen

theorem **lbIso_pullback_O1_toProj** : LBIso ((O1 R M).pull (toProj f s hs)) L

theorem **eq_toProjIso** {w : LB.Iso ((O1 R M).pull (toProj f s hs)) L} (hw : ∀ j, SecMaps w (secPull (coord R M j) (toProj f s hs)) (s j)) : w = toProjIso f s hs

theorem **exists_chart_of_iSup_nonzeroLocus** (y : Y) : ∃ k, y ∈ L.U k

theorem **eq_toProj_of_secMaps** {φ : Y ⟶ PSpace R M} (hφ : φ ≫ PSpace.str R M = f) {w : LB.Iso ((O1 R M).pull φ) L} (hw : ∀ j, SecMaps w (secPull (coord R M j) φ) (s j)) : φ = toProj f s hs

theorem **toProj_eq_comp_of_locally** (g : Z ⟶ Y) {B : LB Z} {t : Fin (M + 1) → B.Sec} (ht : ⨆ i, nonzeroLocus (t i) = ⊤) (H : ∀ z : Z, ∃ (W : Z.Opens) (_ : z ∈ W) (k : L.ι) (hk : W ≤ g ⁻¹ᵁ L.U k) (l : B.ι) (hl : W ≤ B.U l) (u : Γ(Z, W)), ∀ j, resG Z hl ((t j).f l) = u ∗ (g.appLE _ W hk).hom ((s j).f k)) : toProj (g ≫ f) t ht = g ≫ toProj f s hs

theorem **toProj_eq_of_locally** {t : Fin (M + 1) → L'.Sec} (ht : ⨆ i, nonzeroLocus (t i) = ⊤) (H : ∀ y : Y, ∃ (W : Y.Opens) (_ : y ∈ W) (k : L.ι) (hk : W ≤ L.U k) (l : L'.ι) (hl : W ≤ L'.U l) (u : Γ(Y, W)), ∀ j, resG Y hl ((t j).f l) = u ∗ resG Y hk ((s j).f k)) : toProj f t ht = toProj f s hs

theorem **iSup_nonzeroLocus_secScale** (u : Γ(Y, ⊤)ˣ) : ⨆ i, nonzeroLocus (secScale u.val (s i)) = ⊤

theorem **toProj_secScale** (u : Γ(Y, ⊤)ˣ) : toProj f (fun i => secScale u.val (s i)) (iSup_nonzeroLocus_secScale hs u) = toProj f s hs

theorem **toProj_eq_of_secMaps** {t : Fin (M + 1) → L'.Sec} (ht : ⨆ i, nonzeroLocus (t i) = ⊤) {w : LB.Iso L L'} (hw : ∀ j, SecMaps w (s j) (t j)) : toProj f t ht = toProj f s hs

theorem **iSup_nonzeroLocus_of_secMaps** (hL : L.IsLB) (hL' : L'.IsLB) {t : Fin (M + 1) → L'.Sec} {w : LB.Iso L L'} (hw : ∀ j, SecMaps w (s j) (t j)) : ⨆ i, nonzeroLocus (t i) = ⊤

theorem **toProj_secPull** (g : Z ⟶ Y) : toProj (g ≫ f) (fun i => secPull (s i) g) (iSup_nonzeroLocus_secPull hs g) = g ≫ toProj f s hs

theorem **toProj_eq_comp_of_secMaps** (g : Z ⟶ Y) {B : LB Z} {t : Fin (M + 1) → B.Sec} (ht : ⨆ i, nonzeroLocus (t i) = ⊤) {w : LB.Iso (L.pull g) B} (hw : ∀ j, SecMaps w (secPull (s j) g) (t j)) : toProj (g ≫ f) t ht = g ≫ toProj f s hs

theorem **toProj_coord** : toProj (PSpace.str R M) (coord R M) (iSup_nonzeroLocus_coord R M) = 𝟙 (PSpace R M)

theorem **toProj_secPull_coord** (φ : Y ⟶ PSpace R M) : toProj (φ ≫ PSpace.str R M) (fun i => secPull (coord R M i) φ) (iSup_nonzeroLocus_secPull (iSup_nonzeroLocus_coord R M) φ) = φ

theorem **pullRatio_comp_baseChange** {R' : Type} [CommRing R'] (c : R →+* R') (φ : Y ⟶ PSpace R' M) {W : Y.Opens} {i : Fin (M + 1)} (h : W ≤ (φ ≫ baseChange M c) ⁻¹ᵁ chart R M i) (h' : W ≤ φ ⁻¹ᵁ chart R' M i) (j : Fin (M + 1)) : pullRatio (φ ≫ baseChange M c) W i h j = pullRatio φ W i h' j

theorem **toProj_baseChange** {R' : Type} [CommRing R'] (c : R →+* R') (f : Y ⟶ Sp R') (s : Fin (M + 1) → L.Sec) (hs : ⨆ i, nonzeroLocus (s i) = ⊤) : toProj f s hs ≫ baseChange M c = toProj (f ≫ spMap c) s hs

theorem **isClosedImmersion_toProj_iff** (f : Y ⟶ Sp R) (s : Fin (M + 1) → L.Sec) (hs : ⨆ i, nonzeroLocus (s i) = ⊤) : IsClosedImmersion (toProj f s hs) ↔ ∀ i, IsAffineOpen (nonzeroLocus (s i)) ∧ Function.Surjective (eval₂Hom (baseRingHom f (nonzeroLocus (s i))).hom fun j => secRatio (s i) (s j))

theorem **veryAmpleBy_iff** (f : Y ⟶ Sp R) (s : Fin (M + 1) → L.Sec) : VeryAmpleBy f L s ↔ ∃ hs : ⨆ i, nonzeroLocus (s i) = ⊤, IsClosedImmersion (toProj f s hs)

theorem **isProjOver_of_veryAmpleBy** {f : Y ⟶ Sp R} {s : Fin (M + 1) → L.Sec} (h : VeryAmpleBy f L s) : IsProjOver f

theorem **veryAmpleBy_secPull_coord** (φ : Y ⟶ PSpace R M) [IsClosedImmersion φ] : VeryAmpleBy (φ ≫ PSpace.str R M) ((O1 R M).pull φ) fun i => secPull (coord R M i) φ

theorem **isProjBy_iff_isProjOver** (f : Y ⟶ Sp R) : IsProjBy f ↔ IsProjOver f

theorem **veryAmpleBy_comp_iff** {T : Scheme.{0}} (q : Y ⟶ T) (e : T ⟶ Sp R) [IsIso e] (s : Fin (M + 1) → L.Sec) : VeryAmpleBy (q ≫ e) L s ↔ (⨆ i, nonzeroLocus (s i) = ⊤) ∧ ∀ i, IsAffineOpen (nonzeroLocus (s i)) ∧ ∃ r : Fin (M + 1) → Γ(Y, nonzeroLocus (s i)), (∀ j k, resG _ (inf_le_right : nonzeroLocus (s i) ⊓ L.U k ≤ L.U k) ((s j).f k) = resG _ inf_le_left (r j) ∗ resG _ inf_le_right ((s i).f k)) ∧ Function.Surjective (eval₂Hom (q.appLE ⊤ (nonzeroLocus (s i)) le_top).hom r)

theorem **isProjOver_of_isIso_comp** (g : Z ⟶ Y) [IsIso g] {f : Y ⟶ Sp R} (h : IsProjOver (g ≫ f)) : IsProjOver f

theorem **isProjOver_morphismRestrict_of_relAmple** {S : Scheme.{0}} {A : Over S} [GrpObj A] {B : LB A.left} (h : RelAmple A B) {U : S.Opens} (hU : IsAffineOpen U) : IsProjOver (R

theorem **veryAmpleAt_iff** {S : Scheme.{0}} (A : Over S) [GrpObj A] (B : LB A.left) (N : ℕ) : VeryAmpleAt A B N ↔ ∀ (U : S.Opens) (hU : IsAffineOpen U), ∃ (M : ℕ) (s : Fin (M + 1) → (powOver A B U N).Sec) (hs : ⨆ i, nonzeroLocus (s i) = ⊤), IsClosedImmersion (toProj (R

theorem **relAmple_iff** {S : Scheme.{0}} (A : Over S) [GrpObj A] (B : LB A.left) : RelAmple A B ↔ ∀ (U : S.Opens) (hU : IsAffineOpen U), ∃ N : ℕ, 1 ≤ N ∧ ∃ (M : ℕ) (s : Fin (M + 1) → (powOver A B U N).Sec) (hs : ⨆ i, nonzeroLocus (s i) = ⊤), IsClosedImmersion (toProj (R

theorem **isProjOver_of_relAmple** {A : Over (Sp R)} [GrpObj A] {B : LB A.left} (h : RelAmple A B) : IsProjOver A.hom

theorem **isProjOver_polAV** (P : PolAV (Sp R)) : IsProjOver P.A.hom

theorem **toProj_spec** (f : Y ⟶ Sp R) (s : Fin (M + 1) → L.Sec) (hs : ⊔ i, nonzeroLocus (s i) = ⊤) : toProj f s hs ≫ PSpace.str R M = f ∧ (∃ h : ∀ i, toProj f s hs ⁻¹ᵁ Proj.basicOpen (grading R M) (X i) = nonzeroLocus (s i), ∀ i j k, resG Y (inf_le_right : nonzeroLocus (s i) ⊓ L.U k ≤ L.U k) ((s j).f k) = resG Y inf_le_left (((toProj f s hs).appLE (chart R M i) _ (h i).ge).hom (ratio R M i j)) * resG Y inf_le_right ((s i).f k)) ∧ (∃! w : LB.Iso ((O1 R M).pull (toProj f s hs)) L, w.IsGen ∧ ∀ j, SecMaps w (secPull (coord R M j) (toProj f s hs)) (s j)) ∧ (∀ (φ : Y ⟶ PSpace R M) (w : LB.Iso ((O1 R M).pull φ) L), φ ≫ PSpace.str R M = f → (∀ j, SecMaps w (secPull (coord R M j) φ) (s j)) → φ = toProj f s hs) ∧ (∀ u : Γ(Y, ⊤)ˣ, ∃ hu : ⊔ i, nonzeroLocus (secScale u.val (s i)) = ⊤, toProj f (fun i => secScale u.val (s i)) hu = toProj f s hs) ∧ (∀ (B : LB Y) (t : Fin (M + 1) → B.Sec) (w : LB.Iso L B),

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.IsProjOverOfIsPullback

theorem **isProjOver_pspace_str** (R : Type) [CommRing R] (N : ℕ) : IsProjOver (PSpace.str R N)

theorem **isProjOver_of_isPullback** {R : Type} [CommRing R] {X Y Z : Scheme.{0}} {p : Z ⟶ X} {q : Z ⟶ Y} {f : X ⟶ Sp R} {g : Y ⟶ Sp R} (sq : IsPullback p q f g) (hf : IsProjOver f) (hg : IsProjOver g) : IsProjOver (p ≫ f)

theorem **isProjOver_pullback_str** (R : Type) [CommRing R] (N M : ℕ) : IsProjOver (pullback.fst (PSpace.str R N) (PSpace.str R M) ≫ PSpace.str R N)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.KsetTransitive

theorem **isUnit_det_gram_of_leftInverse** {n m : ℕ} (A : Matrix (Fin n) (Fin m) ℂ) (L : Matrix (Fin m) (Fin n) ℂ) (h : L * A = 1) : IsUnit (Aᴴ * A).det

theorem **Iwasawa.exists_unitary_comm** {n m : ℕ} (H : Matrix (Fin n) (Fin n) ℂ) (hH : Hᴴ = H) (hHH : H * H = 1) (A B : Matrix (Fin n) (Fin m) ℂ) (h1 : Aᴴ * A = Bᴴ * B) (h2 : Aᴴ * H * A = Bᴴ * H * B) : ∃ U : Matrix (Fin n) (Fin n) ℂ, Uᴴ * U = 1 ∧ U * H = H * U ∧ U * A = B

theorem **Iwasawa.kset_transitive** : ∀ (k r m : ℕ) (A B : Matrix (Fin (2 * r + k + 1)) (Fin m) ℂ), Aᴴ * A = Bᴴ * B → Aᴴ * Hpk k r * A = Bᴴ * Hpk k r * B → ∃ κ ∈ Kset k r, κ * A = B

theorem **iwasawa** : ∀ k r : ℕ, 1 ≤ k → ∀ g ∈ GV k r, ∃ κ ∈ Kset k r, ∃ p ∈ Pset k r, g = κ * p

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.KsetTransitiveOfGram

theorem **unitary_comm_of_gram** {n m : ℕ} (H : Matrix (Fin n) (Fin n) ℂ) (hH : Hᴴ = H) (hHH : H * H = 1) (A B : Matrix (Fin n) (Fin m) ℂ) (h1 : Aᴴ * A = Bᴴ * B) (h2 : Aᴴ * H * A = Bᴴ * H * B) : ∃ U : Matrix (Fin n) (Fin n) ℂ, Uᴴ * U = 1 ∧ U * H = H * U ∧ U * A = B

theorem **kset_transitive_of_gram** : ∀ (k r m : ℕ) (A B : Matrix (Fin (2 * r + k + 1)) (Fin m) ℂ), Aᴴ * A = Bᴴ * B → Aᴴ * Hpk k r * A = Bᴴ * Hpk k r * B → ∃ κ ∈ Kset k r, κ * A = B

lemma **sqLen_nonneg'** {n : ℕ} (v : Fin n → ℂ) : 0 ≤ sqLen v

lemma **sqLen_pos'** {n : ℕ} {v : Fin n → ℂ} (hv : v ≠ 0) : 0 < sqLen v

lemma **sqLen_smul'** {n : ℕ} (c : ℂ) (v : Fin n → ℂ) : sqLen (c • v) = Complex.normSq c * sqLen v

lemma **star_dot_self** {n : ℕ} (v : Fin n → ℂ) : star v ⬝ᵥ v = (sqLen v : ℂ)

lemma **continuous_sqLen'** {n : ℕ} : Continuous (sqLen : (Fin n → ℂ) → ℝ)

lemma **mul_mem_GV** {k r : ℕ} {a b : Mat k r} (ha : a ∈ GV k r) (hb : b ∈ GV k r) : a * b ∈ GV k r

lemma **mulVec_ne_zero** {k r : ℕ} {a : Mat k r} (ha : a ∈ GV k r) {y : Fin (2 * r + k + 1) → ℂ} (hy : y ≠ 0) : a *ᵥ y ≠ 0

lemma **exists_translate** {k r : ℕ} {x v : Fin (2 * r + k + 1) → ℂ} (hx0 : herm (Hpk k r) x x = 0) (hx1 : sqLen x = 1) (hv0 : herm (Hpk k r) v v = 0) : ∃ κ ∈ Kset k r, ∃ c : ℂ, Complex.normSq c = sqLen v ∧ v = c • (κ *ᵥ x)

lemma **core** (k r : ℕ) (μK : Measure (Mt k r)) (x : Fin (2 * r + k + 1) → ℂ) (hK : IsKMeas k r μK) (hx0 : herm (Hpk k r) x x = 0) (hx1 : sqLen x = 1) (g : Mat k r) (hg : g ∈ GV k r) (m : Measure (Fin (2 * r + k + 1) → ℂ)) [SFinite m] (d : ℝ) (hae : ∀ᵐ v ∂m, v ≠ 0 ∧ herm (Hpk k r) v v = 0) (hhom : ∀ s : ℝ, 0 < s → ∀ f : ℝ → ℝ≥0∞, Measurable f → (∫⁻ v, f (s * sqLen v) ∂m) = ENNReal.ofReal (s ^ d) * ∫⁻ v, f (sqLen v) ∂m) (h0 : (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) ≠ 0) (htop : (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) ≠ ∞) (hinv : ∀ g ∈ GV k r, (∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen (g *ᵥ v))) ∂m) = ∫⁻ v, ENNReal.ofReal (Real.exp (-sqLen v)) ∂m) : (∫⁻ κ : Mt k r, ENNReal.ofReal (sqLen ((g * Matrix.of κ) *ᵥ x) ^ d) ∂μK) = 1

theorem **poisson_integral** : ∀ (k r : ℕ) (μK : Measure (Mt k r)) (x : Fin (2 * r + k + 1) → ℂ), 1 ≤ k → IsKMeas k r μK → herm (Hpk k r) x x = 0 → sqLen x = 1 → ∀ g ∈ GV k r, (∫⁻ κ : Mt k r, ENNReal.ofReal (sqLen ((g * Matrix.of κ) *ᵥ x) ^ (-((2 * r + k : ℕ) : ℝ))) ∂μK) = 1

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.KsetTransitiveStatement

theorem **_root_.GlobalTheory.T.exists_mem_Kset_mul_mem_Pset_eq** : ∀ k r : ℕ, 1 ≤ k → ∀ g ∈ GV k r, ∃ κ ∈ Kset k r, ∃ p ∈ Pset k r, g = κ * p

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.L2OpNormConjTransposeExpMulSignMulExpSubLe

theorem **abs_log_norm_sub_re_sub_one_le** {μ : ℂ} {t : ℝ} (hμ : ‖μ - 1‖ ≤ t) (ht : t < 1) : μ ≠ 0 ∧ |Real.log ‖μ‖ - (μ - 1).re| ≤ t ^ 2 / (2 * (1 - t))

theorem **abs_log_norm_multisetProd_sub_le** (s : Multiset ℂ) {t : ℝ} (ht : t < 1) (hs : ∀ μ ∈ s, ‖μ - 1‖ ≤ t) : s.prod ≠ 0 ∧ |Real.log ‖s.prod‖ - (s.sum - s.card).re| ≤ s.card * (t ^ 2 / (2 * (1 - t)))

theorem **l2_opNorm_diagonal_sign_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) : ‖(diagonal j : Matrix n n ℂ)‖ ≤ 1

theorem **l2_opNorm_one_le** : ‖(1 : Matrix n n ℂ)‖ ≤ 1

theorem **l2_opNorm_map_star** (A : Matrix m n ℂ) : ‖A.map star‖ = ‖A‖

theorem **l2_opNorm_transpose** (A : Matrix m n ℂ) : ‖Aᵀ‖ = ‖A‖

theorem **l2_opNorm_submatrix_le** (A : Matrix n n ℂ) {e : m → n} (he : Function.Injective e) : ‖A.submatrix e e‖ ≤ ‖A‖

theorem **norm_dotProduct_mulVec_star_le** (A : Matrix n n ℂ) (x : n → ℂ) : ‖x ⬝ᵥ A *ᵥ star x‖ ≤ ‖A‖ * ∑ k, ‖x k‖ ^ 2

theorem **norm_diag_le_l2_opNorm** (A : Matrix n n ℂ) (k : n) : ‖A k k‖ ≤ ‖A‖

theorem **norm_trace_le_card_mul_l2_opNorm** (A : Matrix n n ℂ) : ‖A.trace‖ ≤ Fintype.card n * ‖A‖

theorem **abs_log_norm_det_one_add_sub_re_trace_le** {M : Matrix n n ℂ} {t : ℝ} (hM : ‖M‖ ≤ t) (ht : t < 1) : (1 + M).det ≠ 0 ∧ |Real.log ‖(1 + M).det‖ - M.trace.re| ≤ Fintype.card n * t ^ 2 / (2 * (1 - t))

theorem **abs_log_norm_det_sign_add_sub_re_trace_le_of_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {E : Matrix n n ℂ} {t : ℝ} (hE : ‖E‖ ≤ t) (ht : t < 1) : (diagonal j + E).det ≠ 0 ∧ |Real.log ‖(diagonal j + E).det‖ - (diagonal j * E).trace.re| ≤ Fintype.card n * t ^ 2 /

(2 * (1 - t))

theorem **abs_log_norm_det_sign_add_sub_re_trace_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {E : Matrix n n ℂ} (hE : ‖E‖ < 1) : (diagonal j + E).det ≠ 0 ∧ |Real.log ‖(diagonal j + E).det‖ - (diagonal j * E).trace.re| ≤ Fintype.card n * ‖E‖ ^ 2 / (2 * (1 - ‖E‖))

theorem **dotProduct_diagonal_mulVec_star** (d x : n → ℂ) : x ⬝ᵥ diagonal d *ᵥ star x = ∑ k, d k * ((‖x k‖ ^ 2 : ℝ) : ℂ)

theorem **le_re_dotProduct_diagonal_add_of_support** {j : n → ℂ} (E : Matrix n n ℂ) {x : n → ℂ} (hx : ∀ k, x k ≠ 0 → j k = 1) : (1 - ‖E‖) * ∑ k, ‖x k‖ ^ 2 ≤ (x ⬝ᵥ (diagonal j + E) *ᵥ star x).re

theorem **sum_norm_sq_pos** {x : n → ℂ} (hx : x ≠ 0) : 0 < ∑ k, ‖x k‖ ^ 2

theorem **re_dotProduct_diagonal_add_pos_of_support** {j : n → ℂ} {E : Matrix n n ℂ} (hE : ‖E‖ < 1) {x : n → ℂ} (hx0 : x ≠ 0) (hx : ∀ k, x k ≠ 0 → j k = 1) : 0 < (x ⬝ᵥ (diagonal j + E) *ᵥ star x).re

theorem **re_dotProduct_diagonal_add_neg_of_support** {j : n → ℂ} {E : Matrix n n ℂ} (hE : ‖E‖ < 1) {x : n → ℂ} (hx0 : x ≠ 0) (hx : ∀ k, x k ≠ 0 → j k = -1) : (x ⬝ᵥ (diagonal j + E) *ᵥ star x).re < 0

theorem **posDef_one_add** {E : Matrix n n ℂ} (hE : E.IsHermitian) (hE1 : ‖E‖ < 1) : (1 + E).PosDef

theorem **posDef_submatrix_diagonal_add** {j : n → ℂ} {E : Matrix n n ℂ} (hE : E.IsHermitian) (hE1 : ‖E‖ < 1) {e : m → n} (he : Function.Injective e) (hj : ∀ a, j (e a) = 1) : ((diagonal j + E).submatrix e e).PosDef

theorem **posDef_neg_submatrix_diagonal_add** {j : n → ℂ} {E : Matrix n n ℂ} (hE : E.IsHermitian) (hE1 : ‖E‖ < 1) {e : m → n} (he : Function.Injective e) (hj : ∀ a, j (e a) = -1) : (-(diagonal j + E).submatrix e e).PosDef

theorem **finrank_le_card_of_eq_zero** (S : Finset n) (W : Submodule ℂ (n → ℂ)) (h : ∀ x ∈ W, (∀ k ∈ S, x k = 0) → x = 0) : Module.finrank ℂ W ≤ S.card

theorem **isGreatest_finrank_pos_sign_add** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {E : Matrix n n ℂ} (hE : ‖E‖ < 1) : IsGreatest {d : ℕ | ∃ W : Submodule ℂ (n → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → 0 < (x ⬝ᵥ (diagonal j + E) *ᵥ star x).re} (Finset.univ.filter fun k => j k = 1).card

theorem **isGreatest_finrank_neg_sign_add** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {E : Matrix n n ℂ} (hE : ‖E‖ < 1) : IsGreatest {d : ℕ | ∃ W : Submodule ℂ (n → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → (x ⬝ᵥ (diagonal j + E) *ᵥ star x).re < 0} (Finset.univ.filter fun k => j k = -1).card

theorem **card_le_card_pos_of_posDef** (d : n → ℝ) (Q : Matrix n m ℂ) (h : (Qᴴ * diagonal (fun k => (d k : ℂ)) * Q).PosDef) : Fintype.card m ≤ (Finset.univ.filter fun k => 0 < d k).card

theorem **submatrix_eq_eigenvalues** {A : Matrix n n ℂ} (hA : A.IsHermitian) (e : m → n) : ∃ Q : Matrix n m ℂ, A.submatrix e e = Qᴴ * diagonal (fun k => (hA.eigenvalues k : ℂ)) * Q

theorem **card_eigenvalues_sign_add** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {E : Matrix n n ℂ} (hE : ‖E‖ < 1) (hH : (diagonal j + E).IsHermitian) : (Finset.univ.filter fun k => 0 < hH.eigenvalues k).card = (Finset.univ.filter fun k => j k = 1).card ∧ (Finset.univ.filter fun k => hH.eigenvalues k < 0).card = (Finset.univ.filter fun k => j k = -1).card

theorem **norm_exp_sub_one_sub_le** {A : Type*} [NormedRing A] [NormedAlgebra ℂ A] [CompleteSpace A] (x : A) : ‖exp x - 1 - x‖ ≤ Real.exp ‖x‖ - 1 - ‖x‖

theorem **norm_exp_sub_one_sub_le_of_le** {A : Type*} [NormedRing A] [NormedAlgebra ℂ A] [CompleteSpace A] {x : A} {ρ : ℝ} (hx : ‖x‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) : ‖exp x - 1 - x‖ ≤ 0.53 * ρ ^ 2

theorem **transpose_mul_diagonal_mul_map_star_apply** (j : n → ℂ) (g : Matrix n m ℂ) (k l : m) : (gᵀ * diagonal j * g.map star) k l = ∑ a, j a * (g a k * star (g a l))

theorem **conjTranspose_mul_diagonal_mul_eq_transpose** (j : n → ℂ) (g : Matrix n m ℂ) : gᴴ * diagonal j * g = (gᵀ * diagonal j * g.map star)ᵀ

theorem **isHermitian_transpose_mul_sign_mul_map_star** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) (g : Matrix n m ℂ) : (gᵀ * diagonal j * g.map star).IsHermitian

theorem **l2_opNorm_gram_exp_sub_sub_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix n n ℂ} {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) : ‖(exp u)ᵀ * diagonal j * (exp u).map star - diagonal j - (uᵀ * diagonal j + diagonal j * u.map star)‖ ≤ 2.2 * ρ ^ 2

theorem **l2_opNorm_gram_exp_sub_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix n n ℂ} {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) : ‖(exp u)ᵀ * diagonal j * (exp u).map star - diagonal j‖ ≤ 2.3 * ρ

theorem **l2_opNorm_conjTranspose_exp_mul_sign_mul_exp_sub_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix n n ℂ} {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) : ‖(exp u)ᴴ * diagonal j * exp u - diagonal j‖ ≤ 2.3 * ρ

theorem **submatrix_eq_diagonal_add** (j : n → ℂ) (G : Matrix n n ℂ) {e : m → n} (he : Function.Injective e) : G.submatrix e e = diagonal (j ∘ e) + (G - diagonal j).submatrix e e

theorem **abs_log_norm_det_gram_exp_submatrix_le** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix n n ℂ} (hu0 : ∀ k, u k k = 0) {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) {e : m → n} (he : Function.Injective e) : (((exp u)ᵀ * diagonal j * (exp u).map star).submatrix e e).det ≠ 0 ∧ |Real.log ‖(((exp u)ᵀ * diagonal j * (exp u).map star).submatrix e e).det‖| ≤ 5.7 * Fintype.card m * ρ ^ 2

theorem **signature_submatrix_of_norm_sub_sign_lt_one** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {G : Matrix n n ℂ} (hG : G.IsHermitian) (h : ‖G - diagonal j‖ < 1) {e : m → n} (he : Function.Injective e) : IsGreatest {d : ℕ | ∃ W : Submodule ℂ (m → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → 0 < (x ⬝ᵥ G.submatrix e e *ᵥ star x).re} (Finset.univ.filter fun a => j (e a) = 1).card ∧ IsGreatest {d : ℕ | ∃ W : Submodule ℂ (m → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → (x ⬝ᵥ G.submatrix e e *ᵥ star x).re < 0} (Finset.univ.filter fun a => j (e a) = -1).card ∧ (Finset.univ.filter fun a => 0 < (hG.submatrix e).eigenvalues a).card = (Finset.univ.filter fun a => j (e a) = 1).card ∧ (Finset.univ.filter fun a => (hG.submatrix e).eigenvalues a < 0).card = (Finset.univ.filter fun a => j (e a) = -1).card ∧ ((∀ a, j (e a) = 1) → (G.submatrix e e).PosDef) ∧ ((∀ a, j (e a) = -1) → (-G.submatrix e e).Po

theorem **gram_exp_submatrix_logDet_and_signature** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix n n ℂ} (hu0 : ∀ k, u k k = 0) {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) {e : m → n} (he : Function.Injective e) {B : Matrix m m ℂ} (hB : B = ((exp u)ᵀ * diagonal j * (exp u).map star).submatrix e e) : B.det ≠ 0 ∧ |Real.log ‖B.det‖| ≤ 11 * Fintype.card n * ρ ^ 2 ∧ IsGreatest {d : ℕ | ∃ W : Submodule ℂ (m → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → 0 < (x ⬝ᵥ B *ᵥ star x).re} (Finset.univ.filter fun a => j (e a) = 1).card ∧ IsGreatest {d : ℕ | ∃ W : Submodule ℂ (m → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → (x ⬝ᵥ B *ᵥ star x).re < 0} (Finset.univ.filter fun a => j (e a) = -1).card ∧ (∃ hH : B.IsHermitian, (Finset.univ.filter fun a => 0 < hH.eigenvalues a).card = (Finset.univ.filter fun a => j (e a) = 1).card ∧ (Finset.univ.filter fun a => hH.eigenvalues a < 0).card = (Finset.univ.fi

theorem **conjTranspose_exp_mul_sign_mul_exp_submatrix_logDet_and_signature** {j : n → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix n n ℂ} (hu0 : ∀ k, u k k = 0) {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) {e : m → n} (he : Function.Injective e) {B : Matrix m m ℂ} (hB : B = ((exp u)ᴴ * diagonal j * exp u).submatrix e e) : B.det ≠ 0 ∧ |Real.log ‖B.det‖| ≤ 11 * Fintype.card n * ρ ^ 2 ∧ (∃ hH : B.IsHermitian,

(Finset.univ.filter fun a => 0 < hH.eigenvalues a).card = (Finset.univ.filter fun a => j (e a) = 1).card ∧ (Finset.univ.filter fun a => hH.eigenvalues a < 0).card = (Finset.univ.filter fun a => j (e a) = -1).card) ∧ ((∀ a, j (e a) = 1) → B.PosDef) ∧ ((∀ a, j (e a) = -1) → (-B).PosDef)

theorem **gram_exp_leadingBlock_logDet_and_signature** {r i : ℕ} (hi : i ≤ r) {j : Fin r → ℂ} (hj : ∀ k, j k = 1 ∨ j k = -1) {u : Matrix (Fin r) (Fin r) ℂ} (hu0 : (∀ k l, k ≤ l → u k l = 0) ∨ ∀ k l, l ≤ k → u k l = 0) {ρ : ℝ} (hu : ‖u‖ ≤ ρ) (hρ : ρ ≤ 1 / 10) {B : Matrix (Fin i) (Fin i) ℂ} (hB : B = ((exp u)ᵀ * diagonal j * (exp u).map star).submatrix (Fin.castLE hi) (Fin.castLE hi)) : B.det ≠ 0 ∧ |Real.log ‖B.det‖| ≤ 11 * r * ρ ^ 2 ∧ IsGreatest {d : ℕ | ∃ W : Submodule ℂ (Fin i → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → 0 < (x ⬝ᵥ B *ᵥ star x).re} (Finset.univ.filter fun a => j (Fin.castLE hi a) = 1).card ∧ IsGreatest {d : ℕ | ∃ W : Submodule ℂ (Fin i → ℂ), Module.finrank ℂ W = d ∧ ∀ x ∈ W, x ≠ 0 → (x ⬝ᵥ B *ᵥ star x).re < 0} (Finset.univ.filter fun a => j (Fin.castLE hi a) = -1).card ∧ (∃ hH : B.IsHermitian, (Finset.univ.filter fun a => 0 < hH.eigenvalues a).card = (Finset.univ.filter fu

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LatticeFourierOfContDiffFiniteDimensional

theorem **isAddFundamentalDomain_latticeCell** (μ : Measure E) : IsAddFundamentalDomain L (latticeCell L) μ

theorem **isBounded_latticeCell** : Bornology.IsBounded (latticeCell L)

theorem **measurableSet_latticeCell** : MeasurableSet (latticeCell L)

theorem **volume_latticeCell_lt_top** : volume (latticeCell L) < ⊤

theorem **covolume_eq_volume_latticeCell** : ZLattice.covolume L = volume.real (latticeCell L)

theorem **integrableOn_latticeCell_of_continuous** {W : Type*} [NormedAddCommGroup W] {f : E → W} (hf : Continuous f) : IntegrableOn f (latticeCell L)

theorem **mem_innerDual_iff** {ν : E} : ν ∈ L* ↔ ∀ x ∈ L, ∃ n : ℤ, (n : ℝ) = ⟪ν, x⟫

theorem **cexp_two_pi_I_add** (s t : ℝ) : e[s + t] = e[s] * e[t]

theorem **cexp_two_pi_I_intCast** (n : ℤ) : e[(n : ℝ)] = 1

theorem **cexp_two_pi_I_half** : e[1 / 2] = -1

theorem **norm_cexp_two_pi_I** (t : ℝ) : ‖e[t]‖ = 1

theorem **cexp_inner_add_lattice** {ν : E} (hν : ν ∈ L*) {x : E} (hx : x ∈ L) (b : E) : e[⟪ν, b + x⟫] = e[⟪ν, b⟫]

theorem **cexp_neg_inner_add_lattice** {ν : E} (hν : ν ∈ L*) {x : E} (hx : x ∈ L) (b : E) : e[-⟪ν, b + x⟫] = e[-⟪ν, b⟫]

theorem **continuous_cexp_neg_inner_smul** {A : E → V} (hA : Continuous A) (ν : E) : Continuous fun b : E ↦ e[-⟪ν, b⟫] • A b

theorem **latticeFourierCoeff_eq_setIntegral** {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) {ν : E} (hν : ν ∈ L*) {F : Set E} (hF : IsAddFundamentalDomain L F volume) : latticeFourierCoeff L A ν = (ZLattice.covolume L)⁻¹ • ∫ b in F, e[-⟪ν, b⟫] • A b

theorem **norm_latticeFourierCoeff_le** {A : E → V} {S : ℝ} (hS : ∀ b, ‖A b‖ ≤ S) (ν : E) : ‖latticeFourierCoeff L A ν‖ ≤ S

theorem **latticeFourierCoeff_comp_add** {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) {ν : E} (hν : ν ∈ L*) (c : E) : latticeFourierCoeff L (fun b ↦ A (b + c)) ν = e[⟪ν, c⟫] • latticeFourierCoeff L A ν

theorem **latticeFourierCoeff_add** {A B : E → V} (hA : Continuous A) (hB : Continuous B) (ν : E) : latticeFourierCoeff L (fun b ↦ A b + B b) ν = latticeFourierCoeff L A ν + latticeFourierCoeff L B ν

theorem **latticeFourierCoeff_sub** {A B : E → V} (hA : Continuous A) (hB : Continuous B) (ν : E) : latticeFourierCoeff L (fun b ↦ A b - B b) ν = latticeFourierCoeff L A ν - latticeFourierCoeff L B ν

theorem **latticeFourierCoeff_smul** (a : ℂ) (A : E → V) (ν : E) : latticeFourierCoeff L (fun b ↦ a • A b) ν = a • latticeFourierCoeff L A ν

theorem **latticeFourierCoeff_clm** [CompleteSpace V] {W : Type*} [NormedAddCommGroup W] [NormedSpace ℂ W] [CompleteSpace W] (T : V →L[ℂ] W) {A : E → V} (hA : Continuous A) (ν : E) : latticeFourierCoeff L (fun b ↦ T (A b)) ν = T (latticeFourierCoeff L A ν)

theorem **latticeFourierCoeff_fwdDiff_iter** {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) (hc : Continuous A) {ν : E} (hν : ν ∈ L*) (c : E) (k : ℕ) : latticeFourierCoeff L ((fwdDiff c)^[k] A) ν = (e[⟪ν, c⟫] - 1) ^ k • latticeFourierCoeff L A ν

theorem **norm_fwdDiff_iter_le** (c : E) (k : ℕ) {A : E → V} (hA : ContDiff ℝ k A) {M : ℝ} (hM : ∀ x, ‖iteratedFDeriv ℝ k A x‖ ≤ M) (x : E) : ‖(fwdDiff c)^[k] A x‖ ≤ ‖c‖ ^ k * M

theorem **exists_bound_of_lattice_periodic** {W : Type*} [NormedAddCommGroup W] {f : E → W} (hf : ∀ x ∈ L, Function.Periodic f x) (hc : Continuous f) : ∃ M, ∀ b, ‖f b‖ ≤ M

theorem **pow_mul_norm_latticeFourierCoeff_le** {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) {k : ℕ} (hk : ContDiff ℝ k A) {M : ℝ} (hM : ∀ x, ‖iteratedFDeriv ℝ k A x‖ ≤ M) {ν : E} (hν : ν ∈ L*) : (4 * ‖ν‖) ^ k * ‖latticeFourierCoeff L A ν‖ ≤ M

theorem **latticeFourierCoeff_cexp_smul** [CompleteSpace V] [DecidableEq E] {μ ν : E} (hμ : μ ∈ L*) (hν : ν ∈ L*) (v : V) : latticeFourierCoeff L (fun b ↦ e[⟪μ, b⟫] • v) ν = if μ = ν then v else 0

theorem **hasSum_latticeFourierCoeff_tsum** [CompleteSpace V] [DecidableEq E] {ι : Type*} [Countable ι] {m : ι → E} (hm : ∀ j, m j ∈ L*) {a : ι → V} (ha : Summable fun j ↦ ‖a j‖) {ν : E} (hν : ν ∈ L*) : HasSum (fun j ↦ if m j = ν then a j else 0) (latticeFourierCoeff L (fun b ↦ ∑' j, e[⟪m j, b⟫] • a j) ν)

theorem **average_fundamentalDomain_eq_cube** [NormedSpace ℝ E] [FiniteDimensional ℝ E] {ι : Type*} [Fintype ι] (b : Basis ι ℝ E) (μ : Measure E) [μ.IsAddHaarMeasure] (g : E → V) : (μ.real (ZSpan.fundamentalDomain b))⁻¹ • ∫ y in ZSpan.fundamentalDomain b, g y ∂μ = ∫ x in {x : ι → ℝ | ∀ i, x i ∈ Ioc (0 : ℝ) 1}, g (b.equivFun.symm x)

theorem **latticeFourierCoeff_eq_setIntegral_haar** {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) {ν : E} (hν : ν ∈ L*) (μ : Measure E) [μ.IsAddHaarMeasure] {F : Set E} (hF : IsAddFundamentalDomain L F μ) : latticeFourierCoeff L A ν = (ZLattice.covolume L μ)⁻¹ • ∫ b in F, e[-⟪ν, b⟫] • A b ∂μ

theorem **exists_equiv_innerDual** {ι : Type*} [Fintype ι] (b : Basis ι ℤ L) : ∃ b' : Basis ι ℝ E, L* = Submodule.span ℤ (range b') ∧ ∃ φ : (ι → ℤ) ≃ L*, ∀ n x, ⟪(φ n : E), (b.ofZLatticeBasis ℝ L).equivFun.symm x⟫ = ∑ i, n i * x i

theorem **summable_norm_inv_pow_innerDual** {k : ℕ} (hk : finrank ℝ E < k) : Summable fun ν : L* ↦ ‖(ν : E)‖⁻¹ ^ k

theorem **summable_norm_latticeFourierCoeff** {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) {k : ℕ} (hk : finrank ℝ E < k) (hAk : ContDiff ℝ k A) : Summable fun ν : L* ↦ ‖latticeFourierCoeff L A ν‖

private theorem **hasSum_latticeFourier_complex** {A : E → ℂ} (hA : ∀ x ∈ L, Function.Periodic A x) (hc : Continuous A) (hs : Summable fun ν : L* ↦ latticeFourierCoeff L A ν) (y : E) : HasSum (fun ν : L* ↦ e[⟪(ν : E), y⟫] • latticeFourierCoeff L A ν) (A y)

theorem **hasSum_latticeFourier_of_summable_norm** [CompleteSpace V] {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) (hc : Continuous A) (hs : Summable fun ν : L* ↦ ‖latticeFourierCoeff L A ν‖) (y : E) : HasSum (fun ν : L* ↦ e[⟪(ν : E), y⟫] • latticeFourierCoeff L A ν) (A

y)

theorem **hasSum_latticeFourier_of_summable** [FiniteDimensional ℂ V] {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) (hc : Continuous A) (hs : Summable fun ν : L∗ ↦ latticeFourierCoeff L A ν) (y : E) : HasSum (fun ν : L∗ ↦ e[⟪(ν : E), y⟫] • latticeFourierCoeff L A ν) (A y)

theorem **latticeFourier_of_contDiff** [CompleteSpace V] {A : E → V} (hsm : ContDiff ℝ ∞ A) (hA : ∀ x ∈ L, Function.Periodic A x) : (BddAbove (range fun b ↦ ‖A b‖) ∧ ∀ ν : L∗, ‖latticeFourierCoeff L A ν‖ ≤ ⨆ b, ‖A b‖) ∧ (Summable fun ν : L∗ ↦ ‖latticeFourierCoeff L A ν‖) ∧ (∀ b, HasSum (fun ν : L∗ ↦ e[⟪(ν : E), b⟫] • latticeFourierCoeff L A ν) (A b)) ∧ ∀ (c : E) (ν : L∗), latticeFourierCoeff L (fun b ↦ A (b + c)) ν = e[⟪(ν : E), c⟫] • latticeFourierCoeff L A ν

theorem **latticeFourier_of_contDiff_finiteDimensional** [FiniteDimensional ℂ V] {A : E → V} (hsm : ContDiff ℝ ∞ A) (hA : ∀ x ∈ L, Function.Periodic A x) : (BddAbove (range fun b ↦ ‖A b‖) ∧ ∀ ν : L∗, ‖latticeFourierCoeff L A ν‖ ≤ ⨆ b, ‖A b‖) ∧ (Summable fun ν : L∗ ↦ ‖latticeFourierCoeff L A ν‖) ∧ (∀ b, HasSum (fun ν : L∗ ↦ e[⟪(ν : E), b⟫] • latticeFourierCoeff L A ν) (A b)) ∧ ∀ (c : E) (ν : L∗), latticeFourierCoeff L (fun b ↦ A (b + c)) ν = e[⟪(ν : E), c⟫] • latticeFourierCoeff L A ν

theorem **eq_zero_of_latticeFourierCoeff_eq_zero** [CompleteSpace V] {A : E → V} (hA : ∀ x ∈ L, Function.Periodic A x) (hc : Continuous A) (h : ∀ ν : L∗, latticeFourierCoeff L A ν = 0) (y : E) : A y = 0

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LineDerivListEqIteratedFDeriv

@[simp] theorem **lineDerivList_nil** (F : E → V) : lineDerivList 𝕜 [] F = F

theorem **lineDerivList_cons** (v : E) (l : List E) (F : E → V) : lineDerivList 𝕜 (v :: l) F = fun z => lineDeriv 𝕜 (lineDerivList 𝕜 l F) z v

theorem **lineDerivList_append** (l₁ l₂ : List E) (F : E → V) : lineDerivList 𝕜 (l₁ ++ l₂) F = lineDerivList 𝕜 l₁ (lineDerivList 𝕜 l₂ F)

theorem **lineDerivList_const_smul** (l : List E) (c : 𝕜) (F : E → V) : lineDerivList 𝕜 l (fun z => c • F z) = fun z => c • lineDerivList 𝕜 l F z

@[simp] theorem **lineDerivList_zero** (l : List E) : lineDerivList 𝕜 l (fun _ => (0 : V)) = fun _ => 0

theorem **lineDerivList_comp_linearMap** {E' : Type∗} [AddCommGroup E'] [Module 𝕜 E'] (L : E' →ₗ[𝕜] E) (l : List E') (F : E → V) : lineDerivList 𝕜 l (fun z => F (L z)) = fun z => lineDerivList 𝕜 (l.map L) F (L z)

theorem **lineDerivList_comp_add_const** (c : E) (l : List E) (F : E → V) : lineDerivList 𝕜 l (fun z => F (z + c)) = fun z => lineDerivList 𝕜 l F (z + c)

theorem **lineDerivList_map_smul** {κ : Type∗} (c : κ → 𝕜) (e : κ → E) (l : List κ) (F : E → V) : lineDerivList 𝕜 (l.map fun k => c k • e k) F = fun z => (l.map c).prod • lineDerivList 𝕜 (l.map e) F z

theorem **lineDerivList_replicate** (n : ℕ) (v : E) (F : E → V) (z : E) (t : 𝕜) : lineDerivList 𝕜 (List.replicate n v) F (z + t • v) = iteratedDeriv n (fun s : 𝕜 => F (z + s • v)) t

theorem **lineDerivList_congr** {F G : E → V} {a : E} (h : F =ᶠ[𝓝 a] G) (l : List E) : lineDerivList 𝕜 l F =ᶠ[𝓝 a] lineDerivList 𝕜 l G

theorem **analyticAt_lineDeriv** {F : E → V} {a : E} (h : AnalyticAt 𝕜 F a) (v : E) : AnalyticAt 𝕜 (fun z => lineDeriv 𝕜 F z v) a

theorem **analyticAt_lineDerivList** {F : E → V} {a : E} (h : AnalyticAt 𝕜 F a) (l : List E) : AnalyticAt 𝕜 (lineDerivList 𝕜 l F) a

theorem **lineDerivList_ofFn** {F : E → V} {a : E} (h : AnalyticAt 𝕜 F a) {n : ℕ} (m : Fin n → E) : lineDerivList 𝕜 (List.ofFn m) F a = iteratedFDeriv 𝕜 n F a m

theorem **lineDerivList_eq_iteratedFDeriv** {F : E → V} {a : E} (h : AnalyticAt 𝕜 F a) (l : List E) : lineDerivList 𝕜 l F a = iteratedFDeriv 𝕜 l.length F a l.get

theorem **lineDerivList_swap** {F : E → V} {a : E} (h : AnalyticAt 𝕜 F a) (v w : E) : lineDerivList 𝕜 [v, w] F a = lineDerivList 𝕜 [w, v] F a

theorem **lineDerivList_perm** {F : E → V} {a : E} (h : AnalyticAt 𝕜 F a) {l₁ l₂ : List E} (hl : l₁.Perm l₂) : lineDerivList 𝕜 l₁ F a = lineDerivList 𝕜 l₂ F a

theorem **lineDerivList_add** {F G : E → V} {a : E} (hF : AnalyticAt 𝕜 F a) (hG : AnalyticAt 𝕜 G a) (l : List E) : lineDerivList 𝕜 l (fun z => F z + G z) a = lineDerivList 𝕜 l F a + lineDerivList 𝕜 l G a

theorem **lineDerivList_sum** {κ : Type∗} (s : Finset κ) {F : κ → E → V} {a : E} (hF : ∀ k ∈ s, AnalyticAt 𝕜 (F k) a) (l : List E) : lineDerivList 𝕜 l (fun z => ∑ k ∈ s, F k z) a = ∑ k ∈ s, lineDerivList 𝕜 l (F k) a

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LinearIndependentRealOfIntOfDiscrete

theorem **det_gram_eq_normSq_det** {V : Type∗} [NormedAddCommGroup V] [InnerProductSpace 𝕜 V] (b : OrthonormalBasis ι 𝕜 V) (v : ι → V) : (gram 𝕜 v).det = ((‖b.toBasis.det v‖ ^ 2 : ℝ) : 𝕜)

theorem **det_gram_le_of_finrank_eq** {V : Type∗} [NormedAddCommGroup V] [InnerProductSpace 𝕜 V] [FiniteDimensional 𝕜 V] (h : finrank 𝕜 V = Fintype.card ι) (v : ι → V) : (gram 𝕜 v).det ≤ ((∏ i, ‖v i‖ ^ 2 : ℝ) : 𝕜)

theorem **norm_det_sq_le_prod_col** (A : Matrix ι ι 𝕜) : ‖A.det‖ ^ 2 ≤ ∏ j, ∑ i, ‖A i j‖ ^ 2

theorem **posSemidef_det_le_prod_diag** {A : Matrix ι ι 𝕜} (hA : A.PosSemidef) : A.det ≤ ∏ i, A i i

theorem **posSemidef_re_det_le_prod_diag** {A : Matrix ι ι 𝕜} (hA : A.PosSemidef) : RCLike.re A.det ≤ ∏ i, RCLike.re (A i i)

theorem **linearIndependent_real_of_int_of_discrete** {F : Type∗} [NormedAddCommGroup F] [NormedSpace ℝ F] [FiniteDimensional ℝ F] (L : Submodule ℤ F) [DiscreteTopology L] {v : ι → F} (hv : ∀ i, v i ∈ L) (hli : LinearIndependent ℤ v) : LinearIndependent ℝ v

theorem **covolume_span_basis_eq_sqrt_det_gram** [DecidableEq ι] (b : Basis ι ℝ E) : ZLattice.covolume (span ℤ (Set.range b)) = √(gram ℝ b).det

theorem **covolume_eq_sqrt_det_gram** [DecidableEq ι] (L : Submodule ℤ E) [DiscreteTopology L] [IsZLattice ℝ L] (b : Basis ι ℤ L) : ZLattice.covolume L = √(gram ℝ fun i ↦ (b i : E)).det

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LintegralGaussianHyperboloidFiber

theorem **continuous_sqLen** {n : ℕ} : Continuous (sqLen : (Fin n → ℂ) → ℝ)

theorem **form_eq** {k : ℕ} (v : Fin (k + 1) → ℂ) : star v ⬝ᵥ (sgnMat k *ᵥ v) = ((sqLen (Fin.init v : Fin k → ℂ) − Complex.normSq (v (Fin.last k)) : ℝ) : ℂ)

theorem **form_inv** {k : ℕ} {g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ} (hg : gᴴ * sgnMat k * g = sgnMat k) (v : Fin (k + 1) → ℂ) : Complex.normSq ((g *ᵥ v) (Fin.last k)) − sqLen (Fin.init (g *ᵥ v) : Fin k → ℂ) = Complex.normSq (v (Fin.last k)) − sqLen (Fin.init v : Fin k → ℂ)

theorem **normSq_det** {k : ℕ} {g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ} (hg : gᴴ * sgnMat k * g = sgnMat k) : Complex.normSq g.det = 1

theorem **continuous_mulVec** {n : ℕ} (g : Matrix (Fin n) (Fin n) ℂ) : Continuous (fun v : Fin n → ℂ => g *ᵥ v)

theorem **measurePreserving_mulVec** {n : ℕ} {g : Matrix (Fin n) (Fin n) ℂ} (hg : Complex.normSq g.det = 1) : MeasurePreserving (fun v : Fin n → ℂ => g *ᵥ v) volume volume

theorem **exists_lower** {n : ℕ} {g : Matrix (Fin n) (Fin n) ℂ} (hg : g.det ≠ 0) : ∃ c : ℝ, 0 < c ∧ ∀ v, c * sqLen v ≤ sqLen (g *ᵥ v)

theorem **integrable_gauss** {k : ℕ} {c : ℝ} (hc : 0 < c) : Integrable (fun w : Fin k → ℂ => Real.exp (-(c * sqLen w)))

theorem **continuous_pt** {k : ℕ} : Continuous (fun q : ℝ × ((Fin k → ℂ) × ℝ) => pt q.1 q.2)

theorem **sqLen_le_pt** {k : ℕ} (τ : ℝ) (p : (Fin k → ℂ) × ℝ) : sqLen p.1 ≤ sqLen (pt τ p)

theorem **continuous_slice** {k : ℕ} {g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ} (hg : g.det ≠ 0) : Continuous (slice g)

theorem **lintegral_slice** {k : ℕ} {g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ} (hg : gᴴ * sgnMat k * g = sgnMat k) (F : ℝ → ℝ≥0∞) (hF : Measurable F) : ∫⁻ τ in Set.Ioi 0, F τ * slice g τ = ∫⁻ τ in Set.Ioi 0, F τ * slice (k

theorem **slice_eq** {k : ℕ} {g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ} (hg : gᴴ * sgnMat k * g = sgnMat k) {τ : ℝ} (hτ : 0 ≤ τ) : slice g τ = slice (k

theorem **gaussian_slice_invariant** : ∀ (k : ℕ) (g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ), gᴴ * Matrix.diagonal (fun i : Fin (k + 1) => if i = Fin.last k then (-1 : ℂ) else 1) * g = Matrix.diagonal (fun i : Fin (k + 1) => if i = Fin.last k then (-1 : ℂ) else 1) → ∀ τ : ℝ, 0 ≤ τ → (∫⁻ p : (Fin k → ℂ) × ℝ in Set.univ ×ˢ Set.Ioc 0 (2 * Real.pi), ENNReal.ofReal (Real.exp (-sqLen (g *ᵥ (Fin.snoc (α

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LintegralGaussianHyperboloidFiberStatement

theorem **_root_.GlobalTheory.T.lintegral_gaussian_hyperboloid_fiber** : ∀ (k : ℕ) (g : Matrix (Fin (k + 1)) (Fin (k + 1)) ℂ), gᴴ * Matrix.diagonal (fun i : Fin (k + 1) => if i = Fin.last k then (-1 : ℂ) else 1) * g = Matrix.diagonal (fun i : Fin (k + 1) => if i = Fin.last k then (-1 : ℂ) else 1) → ∀ τ : ℝ, 0 ≤ τ → (∫⁻ p : (Fin k → ℂ) × ℝ in Set.univ ×ˢ Set.Ioc 0 (2 * Real.pi), ENNReal.ofReal (Real.exp (-sqLen (g *ᵥ (Fin.snoc (α

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LintegralHyperboloidFibration

theorem **angle_shift** (h : ℝ → ℝ≥0∞) (hp : Function.Periodic h (2 * Real.pi)) : ∫⁻ θ in Set.Ioo (-Real.pi) Real.pi, h θ = ∫⁻ θ in Set.Ioc 0 (2 * Real.pi), h θ

theorem **radial** (a : ℝ) (ha : 0 ≤ a) (F H : ℝ → ℝ≥0∞) : ∫⁻ ρ in Set.Ioi (Real.sqrt a), ENNReal.ofReal ρ * (F (ρ ^ 2 - a) * H ρ) = (1 / 2 : ℝ≥0∞) * ∫⁻ τ in Set.Ioi (0 : ℝ), F τ * H (Real.sqrt (a + τ))

theorem **plane** (a : ℝ) (ha : 0 ≤ a) (F : ℝ → ℝ≥0∞) (g : ℂ → ℝ≥0∞) (hF : Measurable F) (hg : Measurable g) : ∫⁻ z : ℂ, Set.indicator {z : ℂ | a < Complex.normSq z} (fun z => F (Complex.normSq z - a) * g z) z = (1 / 2 : ℝ≥0∞) * ∫⁻ τ in Set.Ioi (0 : ℝ), F τ * ∫⁻ θ in Set.Ioc 0 (2 * Real.pi), g (((Real.sqrt (a + τ) : ℝ) : ℂ) * Complex.exp (Complex.I * (θ : ℝ)))

theorem **measurable_snoc** (k : ℕ) : Measurable fun q : (Fin k → ℂ) × ℂ => (Fin.snoc (α

theorem **measurable_sqLen** (k : ℕ) : Measurable (sqLen : (Fin k → ℂ) → ℝ)

theorem **lintegral_snoc** (k : ℕ) (f : (Fin (k + 1) → ℂ) → ℝ≥0∞) (hf : Measurable f) : ∫⁻ v, f v = ∫⁻ w : Fin k → ℂ, ∫⁻ z : ℂ, f (Fin.snoc (α

theorem **lintegral_cone_polar** : ∀ (k : ℕ) (ψ : (Fin (k + 1) → ℂ) → ℝ≥0∞) (F : ℝ → ℝ≥0∞), Measurable ψ → Measurable F → (∫⁻ v : Fin (k + 1) → ℂ in {v | sqLen (Fin.init v : Fin k → ℂ) < Complex.normSq (v (Fin.last k))}, F (Complex.normSq (v (Fin.last k)) - sqLen (Fin.init v : Fin k → ℂ)) * ψ v) = (1 / 2 : ℝ≥0∞) * ∫⁻ τ : ℝ in Set.Ioi 0, F τ * ∫⁻ p : (Fin k → ℂ) × ℝ in Set.univ ×ˢ Set.Ioc 0 (2 * Real.pi), ψ (Fin.snoc (α

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LintegralHyperboloidFibrationStatement

theorem **_root_.GlobalTheory.T.lintegral_hyperboloid_fibration** : ∀ (k : ℕ) (ψ : (Fin (k + 1) → ℂ) → ℝ≥0∞) (F : ℝ → ℝ≥0∞), Measurable ψ → Measurable F → (∫⁻ v : Fin (k + 1) → ℂ in {v | sqLen (Fin.init v : Fin k → ℂ) < Complex.normSq (v (Fin.last k))}, F (Complex.normSq (v (Fin.last k)) - sqLen (Fin.init v : Fin k → ℂ)) * ψ v) = (1 / 2 : ℝ≥0∞) * ∫⁻ τ : ℝ in Set.Ioi 0, F τ * ∫⁻ p : (Fin k → ℂ) × ℝ in Set.univ ×ˢ Set.Ioc 0 (2 * Real.pi), ψ (Fin.snoc (α

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.LintegralProdPartialSums

theorem **lintegral_radial** {A : ℝ} (hA : 0 < A) (m : ℕ) : ∫⁻ r in Ioi (0 : ℝ), ENNReal.ofReal (r * ((A + r ^ 2) ^ (m + 2))⁻¹) = ENNReal.ofReal ((2 * (m + 1) * A ^ (m + 1))⁻¹)

theorem **lintegral_complex_inv_pow** {A : ℝ} (hA : 0 < A) (m : ℕ) : ∫⁻ z : ℂ, ENNReal.ofReal (((A + ‖z‖ ^ 2) ^ (m + 2))⁻¹) = ENNReal.ofReal (π / (m + 1) * (A ^ (m + 1))⁻¹)

theorem **lintegral_fin_succ** {n : ℕ} {f : (Fin (n + 1) → ℂ) → ℝ≥0∞} (hf : Measurable f) : ∫⁻ y, f y = ∫⁻ z : ℂ, ∫⁻ w : Fin n → ℂ, f (Fin.cons z w)

theorem **lintegral_pi_inv_pow** (n : ℕ) : ∀ (m : ℕ) {A : ℝ}, 0 < A → ∫⁻ z : Fin n → ℂ, ENNReal.ofReal (((A + ∑ k, ‖z k‖ ^ 2) ^ (m + n + 1))⁻¹) = ENNReal.ofReal (π ^ n * m ! / (m + n)! * (A ^ (m + 1))⁻¹)

theorem **lintegral_prod_partial_sums** (n : ℕ) : ∀ {A : ℝ}, 0 < A → ∫⁻ z : Fin n → ℂ, ENNReal.ofReal (∏ i : Fin n, ((A + ∑ k, if k ≤ i then ‖z k‖ ^ 2 else 0) ^ 2)⁻¹) = ENNReal.ofReal (π ^ n / n ! * (A ^ n)⁻¹)

theorem **lintegral_comp_vecMul** {n : Type*} [Fintype n] [DecidableEq n] (R : Matrix n n ℂ) (hR : R.det ≠ 0) {f : (n → ℂ) → ℝ≥0∞} (hf : Measurable f) : ∫⁻ b, f b = ENNReal.ofReal (‖R.det‖ ^ 2) * ∫⁻ y, f (Matrix.vecMul y R)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.MaximalCompactSubgroupProperties

theorem **closedSubgroup** (k r : ℕ) : IsClosedSubgroup (Kset k r)

theorem **compact** (k r : ℕ) : IsCompact (Kset k r)

theorem **hpk_mem** (k r : ℕ) : Hpk k r ∈ Kset k r

theorem **star_mem** {k r : ℕ} {κ : Mat k r} (h : κ ∈ Kset k r) : κᴴ ∈ Kset k r

theorem **comm_hpk** {k r : ℕ} {κ : Mat k r} (h : κ ∈ Kset k r) : κ * Hpk k r = Hpk k r * κ

theorem **sum_normSq_eq_trace** {n : ℕ} (g : Matrix (Fin n) (Fin n) ℂ) : ∑ i, ∑ j, Complex.normSq (g i j) = (trace (gᴴ * g)).re

theorem **gN_biinv** {k r : ℕ} {κ κ' : Mat k r} (h : κ ∈ Kset k r) (h' : κ' ∈ Kset k r) (g : Mat k r) : gN (κ * g * κ') = gN g

theorem **kset_facts** : ∀ k r : ℕ, IsClosedSubgroup (Kset k r) ∧ IsCompact (Kset k r) ∧ Hpk k r ∈ Kset k r ∧ (∀ κ ∈ Kset k r, κᴴ ∈ Kset k r ∧ κ * Hpk k r = Hpk k r * κ) ∧ ∀ κ ∈ Kset k r, ∀ κ' ∈ Kset k r, ∀ g : Mat k r, gN (κ * g * κ') = gN g

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.MaximalCompactSubgroupPropertiesStatement

theorem **_root_.GlobalTheory.T.maximal_compact_subgroup_properties** : ∀ k r : ℕ, IsClosedSubgroup (Kset k r) ∧ IsCompact (Kset k r) ∧ Hpk k r ∈ Kset k r ∧ (∀ κ ∈ Kset k r, κᴴ ∈ Kset k r ∧ κ * Hpk k r = Hpk k r * κ) ∧ ∀ κ ∈ Kset k r, ∀ κ' ∈ Kset k r, ∀ g : Mat k r, gN (κ * g * κ') = gN g

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.MulConjTransposeEqOneOfOrthonormal

theorem **range_subsetEnum** {n : Type*} [LinearOrder n] {i : ℕ} (S : Set.powersetCard n i) : Set.range (subsetEnum S) = (S.1 : Set n)

theorem **subsetEnum_injective** {n : Type*} [LinearOrder n] {i : ℕ} (S : Set.powersetCard n i) : Function.Injective (subsetEnum S)

theorem **alternatingMap_apply_eq_sum_minors** {n : Type*} [Fintype n] [LinearOrder n] {i : ℕ} {M : Type*} [AddCommGroup M] [Module R M] (f : (n → R) [⋀^Fin i]→ₗ[R] M) (A : Fin i → n → R) : f A = ∑ S : Set.powersetCard n i, ((Matrix.of A).submatrix id (subsetEnum S)).det • f (fun a => Pi.single (subsetEnum S a) 1)

theorem **det_mul_eq_sum_minors** {n : Type*} [Fintype n] [LinearOrder n] {i : ℕ} (A : Matrix (Fin i) n R) (B : Matrix n (Fin i) R) : (A * B).det = ∑ S : Set.powersetCard n i, (A.submatrix id (subsetEnum S)).det * (B.submatrix (subsetEnum S) id).det

theorem **det_mul_det_eq_of_range_eq** {m k n : Type*} [Fintype m] [DecidableEq m] [Fintype k] [DecidableEq k] (A : Matrix m n R) (B : Matrix n m R) (φ : k ≃ m) (e : m → n) (g : k → n) (hg : Function.Injective g) (h : Set.range e = Set.range g) : (A.submatrix id e).det * (B.submatrix e id).det = (A.submatrix φ g).det * (B.submatrix g φ).det

theorem **det_mul_eq_sum_minors_of_enum** {m n : Type*} [Fintype m] [DecidableEq m] [Fintype n] (e : Set.powersetCard n (Fintype.card m) → m → n) (he : ∀ S, Set.range (e S) = (S.1 : Set n)) (A : Matrix m n R) (B : Matrix n m R) : (A * B).det = ∑ S : Set.powersetCard n (Fintype.card m), (A.submatrix id (e S)).det * (B.submatrix (e S) id).det

theorem **det_submatrix_mul_eq_sum_minors** {l n o : Type*} [Fintype n] [LinearOrder n] {i : ℕ} (A : Matrix l n R) (B : Matrix n o R) (r : Fin i → l) (c : Fin i → o) : ((A * B).submatrix r c).det = ∑ S : Set.powersetCard n i, (A.submatrix r (subsetEnum S)).det * (B.submatrix (subsetEnum S) c).det

theorem **det_mul_diagonal_mul_eq_sum_minors** {n : Type*} [Fintype n] [LinearOrder n] {i : ℕ} (A : Matrix (Fin i) n R) (d : n → R) (B : Matrix n (Fin i) R) : (A * diagonal d * B).det = ∑ S : Set.powersetCard n i, (∏ s ∈ S.1, d s) * ((A.submatrix id (subsetEnum S)).det * (B.submatrix (subsetEnum S) id).det)

theorem **det_mul_conjTranspose_eq_sum_norm_sq** (A : Matrix (Fin i) n 𝕜) : (A * Aᴴ).det = ((∑ S : Set.powersetCard n i, ‖(A.submatrix id (subsetEnum S)).det‖ ^ 2 : ℝ) : 𝕜)

theorem **sum_norm_sq_minors_eq_one** (A : Matrix (Fin i) n 𝕜) (hA : A * Aᴴ = 1) : ∑ S : Set.powersetCard n i, ‖(A.submatrix id (subsetEnum S)).det‖ ^ 2 = 1

theorem **two_rpow_neg_half_le_choose_rpow** {i N r : ℕ} (hi : i ≤ N) (hr : N ≤ r) : (2 : ℝ) ^ (-(r : ℝ) / 2) ≤ (N.choose i : ℝ) ^ (-(1 : ℝ) / 2)

theorem **exists_choose_rpow_le_norm_minor** (A : Matrix (Fin i) n 𝕜) (hA : A * Aᴴ = 1) : i ≤ Fintype.card n ∧ ∃ S : Set.powersetCard n i, ((Fintype.card n).choose i : ℝ) ^ (-(1 : ℝ) / 2) ≤ ‖(A.submatrix id (subsetEnum S)).det‖

theorem **exists_two_rpow_le_norm_minor** (A : Matrix (Fin i) n 𝕜) (hA : A * Aᴴ = 1) {r : ℕ} (hr : Fintype.card n ≤ r) : ∃ S : Set.powersetCard n i, (2 : ℝ) ^ (-(r : ℝ) / 2) ≤ ‖(A.submatrix id (subsetEnum S)).det‖

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.NatCastMulCoeffOfPderivEq

theorem **subst_zero'** {a : σ → MvPowerSeries τ S} (ha : HasSubst a) : subst a (0 : MvPowerSeries σ R) = 0

theorem **subst_one'** {a : σ → MvPowerSeries τ S} (ha : HasSubst a) : subst a (1 : MvPowerSeries σ R) = 1

theorem **subst_neg'** {a : σ → MvPowerSeries τ S} (ha : HasSubst a) (f : MvPowerSeries σ R) : subst a (-f) = -subst a f

theorem **subst_sum'** {κ : Type*} {a : σ → MvPowerSeries τ S} (ha : HasSubst a) (s : Finset κ) (f : κ → MvPowerSeries σ R) : subst a (∑ k ∈ s, f k) = ∑ k ∈ s, subst a (f k)

theorem **subst_C'** {a : σ → MvPowerSeries τ S} (ha : HasSubst a) (r : R) : subst a (C r : MvPowerSeries σ R) = C (algebraMap R S r)

theorem **continuous_pderiv** [TopologicalSpace R] [DiscreteTopology R] (i : σ) : Continuous (pderiv R i : MvPowerSeries σ R → MvPowerSeries σ R)

theorem **pderiv_comm** (i j : σ) (f : MvPowerSeries σ R) : pderiv R i (pderiv R j f) = pderiv R j (pderiv R i f)

theorem **pderiv_subst** [Fintype σ] {a : σ → MvPowerSeries τ S} (ha : HasSubst a) (f : MvPowerSeries σ R) (j : τ) : pderiv S j (subst a f) = ∑ i, subst a (pderiv R i f) * pderiv S j (a i)

theorem **pderiv_powerSeries_subst** {a : MvPowerSeries τ S} (ha : PowerSeries.HasSubst a) (f : PowerSeries R) (j : τ) : pderiv S j (PowerSeries.subst a f) = PowerSeries.subst a (PowerSeries.derivative R f) * pderiv S j a

theorem **constantCoeff_subst_of_zero** {a : σ → MvPowerSeries τ S} (ha : HasSubst a) (ha0 : ∀ i, constantCoeff (a i) = 0) (f : MvPowerSeries σ R) : constantCoeff (subst a f) = algebraMap R S (constantCoeff f)

theorem **pderiv_map** {T : Type*} [CommRing T] (φ : R →+* T) (i : σ) (f : MvPowerSeries σ R) : pderiv T i (map φ f) = map φ (pderiv R i f)

theorem **natCast_mul_coeff_of_pderiv_eq** {f a : MvPowerSeries σ A} {s : σ} (h : pderiv A s f = a) {α : σ →₀ ℕ} (hα : α s ≠ 0) : (α s : A) * coeff α f = coeff (α - single s 1) a

theorem **ext_of_pderiv_of_coeff** [IsAddTorsionFree A] (e : ι → σ) {f g : MvPowerSeries σ A} (hD : ∀ i, pderiv A (e i) f = pderiv A (e i) g) (h0 : ∀ α : σ →₀ ℕ, (∀ i, α (e i) = 0) → coeff α f = coeff α g) : f = g

theorem **existsUnique_primitive_param** {e : ι → σ} (he : Function.Injective e) (a : ι → MvPowerSeries σ A) (ha : ∀ i j, pderiv A (e i) (a j) = pderiv A (e j) (a i)) : ∃! f : MvPowerSeries σ A, (∀ α : σ →₀ ℕ, (∀ i, α (e i) = 0) → coeff α f = 0) ∧ ∀ i, pderiv A (e i) f = a i

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.NeZeroOnHyperplaneOfEventuallyNeZero

theorem **differentiableOn_intervalIntegral_comp_clm** {E : Type*} [NormedAddCommGroup E] [NormedSpace ℂ E] [FiniteDimensional ℂ E] {U : Set E} (hU : IsOpen U) {h : E → ℂ} (hh : DifferentiableOn ℂ h U) {L : ℝ → E →L[ℂ] E} (hL : Continuous L) (hLU : ∀ s ∈ Icc (0 : ℝ) 1, MapsTo (L s) U U) : DifferentiableOn ℂ (fun z => ∫ s in (0 : ℝ)..1, h (L s z)) U

theorem **update_mem_polydisc** {z : ι → ℂ} (hz : z ∈ univ.pi fun j => ball (0 : ℂ) (r j)) {i : ι} {t : ℂ} (ht : t ∈ ball (0 : ℂ) (r i)) : update z i t ∈ univ.pi fun j => ball (0 : ℂ) (r j)

theorem **exists_eq_coord_mul_of_vanishing** {f : (ι → ℂ) → ℂ} (hf : DifferentiableOn ℂ f (univ.pi fun j => ball 0 (r j))) (i : ι) (h0 : ∀ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), f (update z i 0) = 0) : ∃ g : (ι → ℂ) → ℂ, DifferentiableOn ℂ g (univ.pi fun j => ball 0 (r j)) ∧ ∀ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), f z = z i * g z

theorem **differentiableOn_comp_update_polydisc** {f : (ι → ℂ) → ℂ} (hf : DifferentiableOn ℂ f (univ.pi fun j => ball 0 (r j))) (i : ι) {z : ι → ℂ} (hz : z ∈ univ.pi fun j => ball (0 : ℂ) (r j)) : DifferentiableOn ℂ (fun t => f (update z i t)) (ball 0 (r i))

theorem **exists_eq_coord_pow_mul** {f : (ι → ℂ) → ℂ} (hf : DifferentiableOn ℂ f (univ.pi fun j => ball 0 (r j))) (i : ι) {a : ι → ℂ} (ha : a ∈ univ.pi fun j => ball (0 : ℂ) (r j)) (h0 : ∀ t ∈ ball (0 : ℂ) (r i), t ≠ 0 → f (update a i t) ≠ 0) : ∃ (m : ℕ) (g : (ι → ℂ) → ℂ), DifferentiableOn ℂ g (univ.pi fun j => ball 0 (r j)) ∧ (∀ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), f z = z i ^ m * g z) ∧ ∃ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), g (update z i 0) ≠ 0

theorem **ne_zero_on_hyperplane_of_eventually_ne_zero** {g : (ι → ℂ) → ℂ} (hg : DifferentiableOn ℂ g (univ.pi fun j => ball 0 (r j))) (i : ι) (h1 : ∃ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), g (update z i 0) ≠ 0) {a : ι → ℂ} (ha : a ∈ univ.pi fun j => ball (0 : ℂ) (r j)) (h0 : ∀ᶠ z in 𝓝 a, ∀ t ∈ ball (0 : ℂ) (r i), t ≠ 0 → g (update z i t) ≠ 0) : g (update a i 0) ≠ 0

theorem **exists_eq_unit_mul_monomial** (S : Finset ι) {f : (ι → ℂ) → ℂ} (hf : DifferentiableOn ℂ f (univ.pi fun j => ball 0 (r j))) (h0 : ∀ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), (∀ j ∈ S, z j ≠ 0) → f z ≠ 0) : ∃ (u : (ι → ℂ) → ℂ) (m : ι → ℕ), DifferentiableOn ℂ u (univ.pi fun j => ball 0 (r j)) ∧ ∀ z ∈ univ.pi fun j => ball (0 : ℂ) (r j), u z ≠ 0 ∧ f z = u z * ∏ i ∈ S, z i ^ m i

theorem **isUnitMulMonomialOn_of_ne_zero** {ι κ : Type*} [Fintype ι] {S' : Finset ι} {r : ι → ℝ} {S : Finset κ} {v : (ι → ℂ) → κ → ℂ} (hv : ∀ j ∈ S, DifferentiableOn ℂ (fun z => v z j) (univ.pi fun i => Metric.ball 0 (r i))) (h0 : ∀ z ∈ univ.pi fun i => Metric.ball (0 : ℂ) (r i), z ∉ coordDivisor S' id → z ∉ coordDivisor S v) : IsUnitMulMonomialOn S v S' id (univ.pi fun i => Metric.ball 0 (r i))

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.NormDetMulProdSqLeOfEntryLe

theorem **leftMulMatrix_pi_basisFun** [DecidableEq ι] (x : ι → E) : leftMulMatrix (Pi.basisFun E ι) x = Matrix.diagonal x

theorem **algebraTrace_pi_apply** (x : ι → E) : Algebra.trace E (ι → E) x = ∑ i, x i

theorem **algebraTrace_one_tmul** (x : S) : Algebra.trace E (E ⊗[R] S) (1 ⊗ₜ x) = algebraMap R E (Algebra.trace R S x)

theorem **exists_algEquiv_fin_points** : ∃ (n : ℕ) (P : Fin n ≃ (S →ₐ[R] E)) (e : E ⊗[R] S ≃ₐ[E] (Fin n → E)), ∀ x j, e (1 ⊗ₜ x) j = P j x

theorem **finite_algHom_of_isFiniteSplit** : Finite (S →ₐ[R] E)

theorem **exists_algEquiv_points** {ι : Type*} (P : ι ≃ (S →ₐ[R] E)) : ∃ e : E ⊗[R] S ≃ₐ[E] (ι → E), ∀ x j, e (1 ⊗ₜ x) j = P j x

theorem **natCard_algHom_eq_finrank** [Nontrivial R] : Nat.card (S →ₐ[R] E) = finrank R S

theorem **card_eq_finrank_of_points** (P : ι ≃ (S →ₐ[R] E)) : Fintype.card ι = finrank R S

theorem **algebraMap_trace_eq_sum_points** (P : ι ≃ (S →ₐ[R] E)) (x : S) : algebraMap R E (Algebra.trace R S x) = ∑ j, P j x

theorem **det_points_sq_eq_discr** (P : ι ≃ (S →ₐ[R] E)) (v : ι → S) : (Matrix.of fun i j ↦ P j (v i)).det ^ 2 = algebraMap R E (discr R v)

theorem **det_points_ne_zero_of_basis** (P : ι ≃ (S →ₐ[R] E)) (b : Basis ι R S) : (Matrix.of fun i j ↦ P j (b i)).det ≠ 0

theorem **det_points_ne_zero_iff** {E : Type*} [Field E] [Algebra K E] [Module.Finite K A] [IsFiniteSplit E (E ⊗[K] A)] (P : ι ≃ (A →ₐ[K] E)) (v : ι → A) : (Matrix.of fun i j ↦ P j (v i)).det ≠ 0 ↔ LinearIndependent K v

theorem **discr_ne_zero_iff_of_etale** [Algebra.Etale K A] (h : Fintype.card ι = finrank K A) (v : ι → A) : discr K v ≠ 0 ↔ LinearIndependent K v

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.NormIteratedFDerivApplyLeOfHomogeneousPart

@[simp] theorem **homogeneousPart_zero** (F : E → V) (a v : E) : homogeneousPart 0 F a v = F a

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.NormPderivIterLeOfAnalyticAt

@[simp] theorem **pderivList_nil** (F : (ι → 𝕜) → V) : pderivList [] F = F

theorem **pderivList_append** (l₁ l₂ : List ι) (F : (ι → 𝕜) → V) : pderivList (l₁ ++ l₂) F = pderivList l₁ (pderivList l₂ F)

@[simp] theorem **pderivIter_zero** (F : (ι → 𝕜) → V) : pderivIter 0 F = F

theorem **update_eq_add_smul_single** (z : ι → 𝕜) (k : ι) (t : 𝕜) : Function.update z k t = z + (t - z k) • (Pi.single k 1 : ι → 𝕜)

theorem **pderivList_replicate** (k : ι) (n : ℕ) (F : (ι → 𝕜) → V) (z : ι → 𝕜) : pderivList (List.replicate n k) F z = iteratedDeriv n (fun t => F (Function.update z k t)) (z k)

theorem **pderivIter_single** (k : ι) (n : ℕ) (F : (ι → 𝕜) → V) (z : ι → 𝕜) : pderivIter (Finsupp.single k n) F z = iteratedDeriv n (fun t => F (Function.update z k t)) (z k)

@[simp] theorem **pderivIter_zero_fun** (α : ι →₀ ℕ) : pderivIter α (fun _ : ι → 𝕜 => (0 : V)) = fun _ => 0

theorem **polyDeriv_eq_sum_of_subset** (P : MvPolynomial ι 𝕜) {s : Finset (ι →₀ ℕ)} (hs : P.support ⊆ s) (F : (ι → 𝕜) → V) (a : ι → 𝕜) : polyDeriv P F a = ∑ α ∈ s, P.coeff α • pderivIter α F a

@[simp] theorem **polyDerivLinear_apply** (F : (ι → 𝕜) → V) (a : ι → 𝕜) (P : MvPolynomial ι 𝕜) : polyDerivLinear F a P = polyDeriv P F a

@[simp] theorem **polyDeriv_zero** (F : (ι → 𝕜) → V) (a : ι → 𝕜) : polyDeriv (0 : MvPolynomial ι 𝕜) F a = 0

@[simp] theorem **polyDeriv_monomial** (α : ι →₀ ℕ) (c : 𝕜) (F : (ι → 𝕜) → V) (a : ι → 𝕜) : polyDeriv (MvPolynomial.monomial α c) F a = c • pderivIter α F a

@[simp] theorem **polyDeriv_C** (c : 𝕜) (F : (ι → 𝕜) → V) (a : ι → 𝕜) : polyDeriv (MvPolynomial.C c) F a = c • F a

@[simp] theorem **polyDeriv_one** (F : (ι → 𝕜) → V) (a : ι → 𝕜) : polyDeriv (1 : MvPolynomial ι 𝕜) F a = F a

@[simp] theorem **polyDeriv_zero_fun** (P : MvPolynomial ι 𝕜) (a : ι → 𝕜) : polyDeriv P (fun _ => (0 : V)) a = 0

theorem **analyticAt_pderivIter** {F : (ι → 𝕜) → V} {a : ι → 𝕜} (h : AnalyticAt 𝕜 F a) (α : ι →₀ ℕ) : AnalyticAt 𝕜 (pderivIter α F) a

theorem **pderivList_perm** {F : (ι → 𝕜) → V} {a : ι → 𝕜} (h : AnalyticAt 𝕜 F a) {l₁ l₂ : List ι} (hl : l₁.Perm l₂) : pderivList l₁ F a = pderivList l₂ F a

theorem **pderivIter_add_index** {F : (ι → 𝕜) → V} {a : ι → 𝕜} (h : AnalyticAt 𝕜 F a) (α β : ι →₀ ℕ) : pderivIter (α + β) F a = pderivIter α (pderivIter β F) a

theorem **hasDerivAt_eval_line** (Q : MvPolynomial ι 𝕜) (z : ι → 𝕜) (k : ι) : HasDerivAt (fun t : 𝕜 => MvPolynomial.eval (z + t • (Pi.single k 1 : ι → 𝕜)) Q) (MvPolynomial.eval z (MvPolynomial.pderiv k Q)) 0

private theorem **cauchy_estimate_iterated_pderiv** {F : (ι → ℂ) → V} {a : ι → ℂ} {r : ι → ℝ} {M : ℝ} (α : ι →₀ ℕ) (hr : ∀ k, 0 ≤ r k) (hα : ∀ k, α k ≠ 0 → 0 < r k) (hF : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ ≤ r k) → AnalyticAt ℂ F z) (hM : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ = r k) →

‖F z‖ ≤ M) (s : Finset ι) : ∀ z : ι → ℂ, (∀ k ∈ s, z k = a k) → (∀ k ∉ s, ‖z k - a k‖ = r k) → ‖pderivIter (∑ k ∈ s, Finsupp.single k (α k)) F z‖ ≤ (∏ k ∈ s, ((α k).factorial : ℝ)) * M / ∏ k ∈ s, r k ^ α k

theorem **norm_pderivIter_le_of_analyticAt** {F : (ι → ℂ) → V} {a : ι → ℂ} {r : ι → ℝ} {M : ℝ} (α : ι →₀ ℕ) (hr : ∀ k, 0 ≤ r k) (hα : ∀ k, α k ≠ 0 → 0 < r k) (hF : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ ≤ r k) → AnalyticAt ℂ F z) (hM : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ = r k) → ‖F z‖ ≤ M) : ‖pderivIter α F a‖ ≤ (∏ k, ((α k).factorial : ℝ)) * M / ∏ k, r k ^ α k

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.OneLeDetOneAdd

theorem **one_le_det_one_add** {𝕜 : Type*} {n : Type*} [RCLike 𝕜] [Fintype n] [DecidableEq n] {D : Matrix n n 𝕜} (hD : D.PosSemidef) : (1 : 𝕜) ≤ (1 + D).det

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ParabolicLineStabilizerLevi

theorem **e0_eq** : e0 k r' = Pi.single (fst k r') 1

theorem **gram_mulVec_e0** : Hpk k (1 + r') *ᵥ e0 k r' = Pi.single (lst k r') 1

theorem **star_e0_vecMul_hpk_eq_single_lst** : star (e0 k r') ᵥ* Hpk k (1 + r') = Pi.single (lst k r') 1

theorem **pair_e0_left** (v : Fin (2 * (1 + r') + k + 1) → ℂ) : star (e0 k r') ⬝ᵥ (Hpk k (1 + r') *ᵥ v) = v (lst k r')

theorem **lift_lst** (w : Fin (2 * r' + k + 1) → ℂ) : liftVec k 1 r' w (lst k r') = 0

theorem **e0_lst_eq_zero** : e0 k r' (lst k r') = 0

theorem **decomp** (u : Fin (2 * (1 + r') + k + 1) → ℂ) (hu : u (lst k r') = 0) : u = liftVec k 1 r' ((incl k 1 r')ᴴ *ᵥ u) + u (fst k r') • e0 k r'

theorem **pair_unitary** {n : ℕ} (H p : Matrix (Fin n) (Fin n) ℂ) (hp : pᴴ * H * p = H) (v w : Fin n → ℂ) : star (p *ᵥ w) ⬝ᵥ (H *ᵥ (p *ᵥ v)) = star w ⬝ᵥ (H *ᵥ v)

theorem **pair_lift** (v w : Fin (2 * r' + k + 1) → ℂ) : star (liftVec k 1 r' w) ⬝ᵥ (Hpk k (1 + r') *ᵥ liftVec k 1 r' v) = star w ⬝ᵥ (Hpk k r' *ᵥ v)

theorem **pair_shift** (v w : Fin (2 * r' + k + 1) → ℂ) (c d : ℂ) : star (liftVec k 1 r' w + d • e0 k r') ⬝ᵥ (Hpk k (1 + r') *ᵥ (liftVec k 1 r' v + c • e0 k r')) = star w ⬝ᵥ (Hpk k r' *ᵥ v)

theorem **acts_through_level_one** : ∀ (k r' : ℕ) (a : ℂ), ∀ p ∈ GV k (1 + r'), p *ᵥ e0 k r' = a • e0 k r' → a ≠ 0 ∧ ∃ g₁ ∈ GV k r', ∀ w : Fin (2 * r' + k + 1) → ℂ, ∃ c : ℂ, p *ᵥ liftVec k 1 r' w = liftVec k 1 r' (g₁ *ᵥ w) + c • e0 k r'

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ParabolicLineStabilizerLeviStatement

theorem **_root_.GlobalTheory.T.parabolic_line_stabilizer_levi** : ∀ (k r' : ℕ) (a : ℂ), ∀ p ∈ GV k (1 + r'), p *ᵥ e0 k r' = a • e0 k r' → a ≠ 0 ∧ ∃ g₁ ∈ GV k r', ∀ w : Fin (2 * r' + k + 1) → ℂ, ∃ c : ℂ, p *ᵥ liftVec k 1 r' w = liftVec k 1 r' (g₁ *ᵥ w) + c • e0 k r'

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.PderivIterEqPderivListOfDifferentiableOn

theorem **norm_pderivIter_le_of_nonneg** {F : (ι → ℂ) → V} {U : Set (ι → ℂ)} {a : ι → ℂ} {r : ι → ℝ} {M : ℝ} (α : ι →₀ ℕ) (hr : ∀ k, 0 ≤ r k) (hα : ∀ k, α k ≠ 0 → 0 < r k) (hU : IsOpen U) (hF : DifferentiableOn ℂ F U) (hsub : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ ≤ r k) → z ∈ U) (hM : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ = r k) → ‖F z‖ ≤ M) : ‖pderivIter α F a‖ ≤ (∏ k, ((α k).factorial : ℝ)) * M / ∏ k, r k ^ α k

theorem **norm_pderivIter_le_of_differentiable** {F : (ι → ℂ) → V} {a : ι → ℂ} {r : ι → ℝ} {M : ℝ} (α : ι →₀ ℕ) (hr : ∀ k, 0 ≤ r k) (hα : ∀ k, α k ≠ 0 → 0 < r k) (hF : Differentiable ℂ F) (hM : ∀ z : ι → ℂ, (∀ k, ‖z k - a k‖ = r k) → ‖F z‖ ≤ M) : ‖pderivIter α F a‖ ≤ (∏ k, ((α k).factorial : ℝ)) * M / ∏ k, r k ^ α k

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.ResidueNormValuationSubringEqZeroIff

theorem **mem_normValuationSubring_iff** (x : C) : x ∈ 𝒪 ↔ ‖x‖ ≤ 1

theorem **residue_normValuationSubring_eq_zero_iff** (x : 𝒪) : IsLocalRing.residue 𝒪 x = 0 ↔ ‖(x : C)‖ < 1

theorem **norm_finsuppProd_pow** (t : σ →₀ ℕ) (z : σ → C) : ‖t.prod (z · ^ ·)‖ = t.prod (fun i n => ‖z i‖ ^ n)

theorem **finsuppProd_pow_mono** (t : σ →₀ ℕ) {c d : σ → ℝ} (h0 : ∀ i, 0 ≤ c i) (h : ∀ i, c i ≤ d i) : t.prod (c · ^ ·) ≤ t.prod (d · ^ ·)

theorem **finsuppProd_const_pow** (t : σ →₀ ℕ) (r : ℝ) : t.prod (fun _ n => r ^ n) = r ^ t.degree

theorem **norm_lt_of_hasSum** {ι : Type*} {f : ι → C} {s : C} {r : ℝ} (hr : 0 < r) (hf : HasSum f s) (h : ∀ i, ‖f i‖ < r) : ‖s‖ < r

theorem **exists_norm_sum_monomial_eq_one** [Infinite (IsLocalRing.ResidueField 𝒪)] (S : Finset (σ →₀ ℕ)) (b : (σ →₀ ℕ) → C) (hb : ∀ t, ‖b t‖ ≤ 1) {t₀ : σ →₀ ℕ} (h₀ : t₀ ∈ S) (hb₀ : ‖b t₀‖ = 1) : ∃ z : σ → C, (∀ i, ‖z i‖ ≤ 1) ∧ ‖∑ t ∈ S, b t * t.prod (z · ^ ·)‖ = 1

theorem **exists_norm_eq_of_tendsto_zero** [Infinite (IsLocalRing.ResidueField 𝒪)] (a : (σ →₀ ℕ) → C) (ha : Tendsto a cofinite (𝓝 0)) {t₀ : σ →₀ ℕ} (h₀ : ∀ t, ‖a t‖ ≤ ‖a t₀‖) : ∃ z : σ → C, (∀ i, ‖z i‖ ≤ 1) ∧ ∀ s, HasSum (fun t => a t * t.prod (z · ^ ·)) s → ‖s‖ = ‖a t₀‖

theorem **infinite_residueField_of_isAlgClosed** [IsAlgClosed C] : Infinite (IsLocalRing.ResidueField 𝒪)

theorem **isRestricted_const_iff** (g : MvPowerSeries σ C) (r : ℝ) : IsRestricted (fun _ => r) g ↔ Tendsto (fun t : σ →₀ ℕ => ‖coeff t g‖ * r ^ t.degree) cofinite (𝓝 0)

theorem **norm_term_le** (g : MvPowerSeries σ C) {c : σ → ℝ} {z : σ → C} (hz : ∀ i, ‖z i‖ ≤ c i) (t : σ →₀ ℕ) : ‖coeff t g * t.prod (z · ^ ·)‖ ≤ ‖coeff t g‖ * t.prod (c · ^ ·)

theorem **hasSum_seriesValue** [CompleteSpace C] {g : MvPowerSeries σ C} {c : σ → ℝ} (hg : IsRestricted c g) {z : σ → C} (hz : ∀ i, ‖z i‖ ≤ c i) : HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) (seriesValue g z)

theorem **isRestricted_of_summable** {g : MvPowerSeries σ C} {w : σ → C} (h : Summable fun t : σ →₀ ℕ => coeff t g * t.prod (w · ^ ·)) : IsRestricted (fun i => ‖w i‖) g

theorem **exists_isGreatest_gaussNorm** {g : MvPowerSeries σ C} {c : σ → ℝ} (hc : ∀ i, 0 ≤ c i) (hg : IsRestricted c g) : IsGreatest (Set.range fun t : σ →₀ ℕ => ‖coeff t g‖ * t.prod (c · ^ ·)) (gaussNorm norm c g)

theorem **norm_le_gaussNorm_of_hasSum** {g : MvPowerSeries σ C} {c : σ → ℝ} (hg : IsRestricted c g) {z : σ → C} (hz : ∀ i, ‖z i‖ ≤ c i) {s : C} (hs : HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) s) : ‖s‖ ≤ gaussNorm norm c g

theorem **exists_norm_eq_gaussNorm_of_hasSum** [Infinite (IsLocalRing.ResidueField 𝒪)] {g : MvPowerSeries σ C} {w : σ → C} (hg : IsRestricted (fun i => ‖w i‖) g) : ∃ z : σ → C, (∀ i, ‖z i‖ ≤ ‖w i‖) ∧ ∀ s, HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) s → ‖s‖ = gaussNorm norm (fun i => ‖w i‖) g

theorem **isGreatest_norm_seriesValue** [CompleteSpace C] [Infinite (IsLocalRing.ResidueField 𝒪)] {g : MvPowerSeries σ C} {w : σ → C} (hg : IsRestricted (fun i => ‖w i‖) g) : IsGreatest ((fun z => ‖seriesValue g z‖) '' {z | ∀ i, ‖z i‖ ≤ ‖w i‖}) (gaussNorm norm (fun i => ‖w

i‖) g)

theorem **norm_coeff_le_of_hasSum** [Infinite (IsLocalRing.ResidueField 𝒪)] (hd : ∀ r : ℝ, r < 1 → ∃ x : C, r < ‖x‖ ∧ ‖x‖ < 1) {g : MvPowerSeries σ C} {M : ℝ} (h : ∀ x : C, ‖x‖ < 1 → ∀ z : σ → C, (∀ i, ‖z i‖ ≤ ‖x‖) → ∃ s, HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) s ∧ ‖s‖ ≤ M) (t : σ →₀ ℕ) : ‖coeff t g‖ ≤ M

theorem **norm_coeff_le_of_norm_seriesValue_le** [CompleteSpace C] [Infinite (IsLocalRing.ResidueField 𝒪)] (hd : ∀ r : ℝ, r < 1 → ∃ x : C, r < ‖x‖ ∧ ‖x‖ < 1) {g : MvPowerSeries σ C} (hg : ∀ r : ℝ, 0 ≤ r → r < 1 → IsRestricted (fun _ => r) g) {M : ℝ} (h : ∀ z : σ → C, (∀ i, ‖z i‖ < 1) → ‖seriesValue g z‖ ≤ M) (t : σ →₀ ℕ) : ‖coeff t g‖ ≤ M

theorem **exists_map_eq_iff_norm_coeff_le_one** (g : MvPowerSeries σ C) : (∃ g₀ : MvPowerSeries σ 𝒪, MvPowerSeries.map (𝒪).subtype g₀ = g) ↔ ∀ t, ‖coeff t g‖ ≤ 1

theorem **isRestricted_of_norm_coeff_le** [Finite σ] {g : MvPowerSeries σ C} {M : ℝ} (h : ∀ t, ‖coeff t g‖ ≤ M) {r : ℝ} (h0 : 0 ≤ r) (h1 : r < 1) : IsRestricted (fun _ => r) g

theorem **exists_radius_lt_one** [Finite σ] {z : σ → C} (hz : ∀ i, ‖z i‖ < 1) : ∃ r : ℝ, 0 ≤ r ∧ r < 1 ∧ ∀ i, ‖z i‖ ≤ r

theorem **hasSum_seriesValue_of_norm_lt_one** [CompleteSpace C] [Finite σ] {g : MvPowerSeries σ C} (hg : ∀ r : ℝ, 0 ≤ r → r < 1 → IsRestricted (fun _ => r) g) {z : σ → C} (hz : ∀ i, ‖z i‖ < 1) : HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) (seriesValue g z)

theorem **norm_le_of_hasSum_of_norm_coeff_le** {g : MvPowerSeries σ C} {M : ℝ} (h : ∀ t, ‖coeff t g‖ ≤ M) {z : σ → C} (hz : ∀ i, ‖z i‖ ≤ 1) {s : C} (hs : HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) s) : ‖s‖ ≤ M

theorem **bounded_series_integral_of_dense** [CompleteSpace C] [Infinite (IsLocalRing.ResidueField 𝒪)] (hd : ∀ r : ℝ, r < 1 → ∃ x : C, r < ‖x‖ ∧ ‖x‖ < 1) [Finite σ] (g : MvPowerSeries σ C) (hg : ∀ r : ℝ, 0 ≤ r → r < 1 → Tendsto (fun t : σ →₀ ℕ => ‖coeff t g‖ * r ^ t.degree) cofinite (𝓝 0)) : (∀ z : σ → C, (∀ i, ‖z i‖ < 1) → HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) (seriesValue g z)) ∧ (∀ x : C, ‖x‖ < 1 → ∃ m : ℝ, IsGreatest ((fun z => ‖seriesValue g z‖) '' {z : σ → C | ∀ i, ‖z i‖ ≤ ‖x‖}) m ∧ IsGreatest (Set.range fun t : σ →₀ ℕ => ‖coeff t g‖ * ‖x‖ ^ t.degree) m) ∧ (∀ M : ℝ, (∀ z : σ → C, (∀ i, ‖z i‖ < 1) → ‖seriesValue g z‖ ≤ M) ↔ ∀ t, ‖coeff t g‖ ≤ M) ∧ ((∀ z : σ → C, (∀ i, ‖z i‖ < 1) → ‖seriesValue g z‖ ≤ 1) ↔ ∃ g₀ : MvPowerSeries σ 𝒪, MvPowerSeries.map (𝒪).subtype g₀ = g)

theorem **exists_lt_norm_lt_one_of_densely** {C : Type*} [DenselyNormedField C] (r : ℝ) (hr : r < 1) : ∃ x : C, r < ‖x‖ ∧ ‖x‖ < 1

theorem **exists_lt_norm_lt_one_of_isAlgClosed** {C : Type*} [NontriviallyNormedField C] [IsAlgClosed C] (r : ℝ) (hr : r < 1) : ∃ x : C, r < ‖x‖ ∧ ‖x‖ < 1

theorem **bounded_series_integral** {C : Type*} [NontriviallyNormedField C] [IsUltrametricDist C] [CompleteSpace C] [IsAlgClosed C] {σ : Type*} [Finite σ] (g : MvPowerSeries σ C) (hg : ∀ r : ℝ, 0 ≤ r → r < 1 → Tendsto (fun t : σ →₀ ℕ => ‖coeff t g‖ * r ^ t.degree) cofinite (𝓝 0)) : (∀ z : σ → C, (∀ i, ‖z i‖ < 1) → HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) (seriesValue g z)) ∧ (∀ x : C, ‖x‖ < 1 → ∃ m : ℝ, IsGreatest ((fun z => ‖seriesValue g z‖) '' {z : σ → C | ∀ i, ‖z i‖ ≤ ‖x‖}) m ∧ IsGreatest (Set.range fun t : σ →₀ ℕ => ‖coeff t g‖ * ‖x‖ ^ t.degree) m) ∧ (∀ M : ℝ, (∀ z : σ → C, (∀ i, ‖z i‖ < 1) → ‖seriesValue g z‖ ≤ M) ↔ ∀ t, ‖coeff t g‖ ≤ M) ∧ ((∀ z : σ → C, (∀ i, ‖z i‖ < 1) → ‖seriesValue g z‖ ≤ 1) ↔ ∃ g₀ : MvPowerSeries σ (NormedField.valuation (K

theorem **norm_coeff_le_of_hasSum_of_isAlgClosed** {C : Type*} [NontriviallyNormedField C] [IsUltrametricDist C] [IsAlgClosed C] {σ : Type*} {g : MvPowerSeries σ C} {f : (σ → C) → C} (hf : ∀ z : σ → C, (∀ i, ‖z i‖ < 1) → HasSum (fun t : σ →₀ ℕ => coeff t g * t.prod (z · ^ ·)) (f z)) {M : ℝ} (h : ∀ z : σ → C, (∀ i, ‖z i‖ < 1) → ‖f z‖ ≤ M) (t : σ →₀ ℕ) : ‖coeff t g‖ ≤ M

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.SetIntegralPowEqZeroOfPosFreqEuclidean

theorem **summable_prod_tuples** {u : ∀ i, J i → ℝ} (h0 : ∀ i j, 0 ≤ u i j) (hu : ∀ i, Summable (u i)) : Summable fun t : ∀ i, J i => ∏ i, u i (t i)

theorem **summable_norm_multilinear_tuples** (M : ContinuousMultilinearMap 𝕜 V W) {x : ∀ i, J i → V i} (hx : ∀ i, Summable fun j => ‖x i j‖) : Summable fun t : ∀ i, J i => ‖M fun i => x i (t i)‖

theorem **hasSum_multilinear_tuples** [CompleteSpace W] (M : ContinuousMultilinearMap 𝕜 V W) {x : ∀ i, J i → V i} {A : ∀ i, V i} (hx : ∀ i, Summable fun j => ‖x i j‖) (hA : ∀ i, HasSum (x i) (A i)) : HasSum (fun t : ∀ i, J i => M fun i => x i (t i)) (M A)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.SphericalFunctionFunctionalEq

lemma **gv_mul** {k r : ℕ} {a b : Mat k r} (ha : a ∈ GV k r) (hb : b ∈ GV k r) : a * b ∈ GV k r

lemma **meas_v** {k r : ℕ} (hk : 1 ≤ k) {α : Type} [TopologicalSpace α] [MeasurableSpace α] [OpensMeasurableSpace α] {f : α → Mat k r} (hf : Continuous f) : Measurable fun t => ENNReal.ofReal (vHalf k r (f t))

lemma **cont_of** {k r : ℕ} : Continuous fun κ : Mt k r => (Matrix.of κ : Mat k r)

lemma **inner** {k r : ℕ} (μK : Measure (Mt k r)) (hk : 1 ≤ k) (hμ : IsKMeas k r μK) {x : Mat k r} (hx : x ∈ GV k r) {g : Mat k r} (hg : g ∈ GV k r) : (∫⁻ κ : Mt k r, ENNReal.ofReal (vHalf k r (x * Matrix.of κ * g)) ∂μK) = XiE k r μK x * ENNReal.ofReal (vHalf k r g)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.SphericalFunctionLeOne

lemma **gv_mul** {k r : ℕ} {a b : Mat k r} (ha : a ∈ GV k r) (hb : b ∈ GV k r) : a * b ∈ GV k r

lemma **meas_v** {k r : ℕ} (hk : 1 ≤ k) {α : Type} [TopologicalSpace α] [MeasurableSpace α] [OpensMeasurableSpace α] {f : α → Mat k r} (hf : Continuous f) : Measurable fun t => ENNReal.ofReal (vHalf k r (f t)) ^ 2

lemma **cont_of** {k r : ℕ} : Continuous fun κ : Mt k r => (Matrix.of κ : Mat k r)

lemma **cont_lev** (k i r' : ℕ) : Continuous (ιLev k i r')

lemma **sqLen_nonneg** {n : ℕ} (v : Fin n → ℂ) : 0 ≤ sqLen v

lemma **diag_re** {n m : ℕ} (A : Matrix (Fin n) (Fin m) ℂ) (a : Fin m) : ((Aᴴ * A) a a).re = sqLen fun i => A i a

lemma **col_facts** (k r : ℕ) (hk : 1 ≤ k) (a : Fin (r + 1)) : herm (Hpk k r) (fvec k r (isoVec k r) a) (fvec k r (isoVec k r) a) = 0 ∧ sqLen (fvec k r (isoVec k r) a) = 1

lemma **sq_rpow** {s : ℝ} (hs : 0 ≤ s) (m : ℕ) : ENNReal.ofReal (s ^ (-((m : ℝ) / 2))) ^ 2 = ENNReal.ofReal (s ^ (-(m : ℝ)))

lemma **vHalf_zero** (k : ℕ) (x : Mat k 0) : vHalf k 0 x = sqLen (x *ᵥ isoVec k 0) ^ (-((k : ℝ) / 2))

lemma **base** (k : ℕ) (μK : Measure (Mt k 0)) (hk : 1 ≤ k) (hμ : IsKMeas k 0 μK) (g : Mat k 0) (hg : g ∈ GV k 0) : (∫⁻ κ : Mt k 0, ENNReal.ofReal (vHalf k 0 (g * Matrix.of κ)) ^ 2 ∂μK) = 1

lemma **step** (k r' : ℕ) (hk : 1 ≤ k) (ih : ∀ μ : Measure (Mt k r'), IsKMeas k r' μ → ∀ g ∈ GV k r', (∫⁻ κ : Mt k r', ENNReal.ofReal (vHalf k r' (g * Matrix.of κ)) ^ 2 ∂μ) = 1) (μK : Measure (Mt k (1 + r'))) (hμ : IsKMeas k (1 + r') μK) (g : Mat k (1 + r')) (hg : g ∈ GV k (1 + r')) : (∫⁻ κ : Mt k (1 + r'), ENNReal.ofReal (vHalf k (1 + r') (g * Matrix.of κ)) ^ 2 ∂μK) = 1

theorem **lintegral_vHalf_sq** : ∀ (k r : ℕ) (μK : Measure (Mt k r)), 1 ≤ k → IsKMeas k r μK → ∀ g ∈ GV k r, (∫⁻ κ : Mt k r, ENNReal.ofReal (vHalf k r (g ∗ Matrix.of κ)) ^ 2 ∂μK) = 1

theorem **two_mul_le_one_add_sq_ennreal** (a : ℝ≥0∞) : 2 ∗ a ≤ 1 + a ^ 2

theorem **lintegral_le_one_of_lintegral_sq_eq_one** {X : Type∗} [MeasurableSpace X] (μ : Measure X) (hμ : μ Set.univ = 1) (f : X → ℝ≥0∞) (hf : ∫⁻ x, f x ^ 2 ∂μ = 1) : ∫⁻ x, f x ∂μ ≤ 1

theorem **XiE_le_one** : ∀ (k r : ℕ) (μK : Measure (Mt k r)), 1 ≤ k → IsKMeas k r μK → ∀ g ∈ GV k r, XiE k r μK g ≤ 1

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.StarInvMulVecDotProductMulVecInvMulVec

theorem **trace_mul_nonneg_of_posSemidef** {A B : Matrix n n 𝕜} (hA : A.PosSemidef) (hB : B.PosSemidef) : 0 ≤ (A ∗ B).trace

theorem **re_trace_mul_nonneg_of_posSemidef** {A B : Matrix n n 𝕜} (hA : A.PosSemidef) (hB : B.PosSemidef) : 0 ≤ RCLike.re (A ∗ B).trace

theorem **posSemidef_of_le** {A B : Matrix n n 𝕜} (hA : A.PosSemidef) (h : A ≤ B) : B.PosSemidef

theorem **posDef_of_le** {A B : Matrix n n 𝕜} (hA : A.PosDef) (h : A ≤ B) : B.PosDef

theorem **conjTranspose_mul_mul_le_of_le** {m : Type∗} [Finite m] {A B : Matrix n n 𝕜} (h : A ≤ B) (C : Matrix n m 𝕜) : Cᴴ ∗ A ∗ C ≤ Cᴴ ∗ B ∗ C

theorem **transpose_le_transpose_of_le** {A B : Matrix n n 𝕜} (h : A ≤ B) : Aᵀ ≤ Bᵀ

theorem **real_smul_le_real_smul_of_le** {A B : Matrix n n 𝕜} (h : A ≤ B) {c : ℝ} (hc : 0 ≤ c) : c • A ≤ c • B

theorem **inv_le_inv_of_posDef_of_le** {A B : Matrix n n 𝕜} (hA : A.PosDef) (h : A ≤ B) : B⁻¹ ≤ A⁻¹

theorem **one_le_det_one_add_of_posSemidef** {D : Matrix n n 𝕜} (hD : D.PosSemidef) : 1 ≤ (1 + D).det

theorem **det_le_det_of_posSemidef_of_le** {A B : Matrix n n 𝕜} (hA : A.PosSemidef) (h : A ≤ B) : A.det ≤ B.det

theorem **re_det_le_re_det_of_posSemidef_of_le** {A B : Matrix n n 𝕜} (hA : A.PosSemidef) (h : A ≤ B) : RCLike.re A.det ≤ RCLike.re B.det

theorem **re_det_pos_of_posDef** {A : Matrix n n 𝕜} (hA : A.PosDef) : 0 < RCLike.re A.det

theorem **pow_mul_re_det_le_of_smul_le** {B B' : Matrix n n 𝕜} (hB : B.PosSemidef) {c : ℝ} (hc : 0 ≤ c) (h : c • B ≤ B') : c ^ Fintype.card n ∗ RCLike.re B.det ≤ RCLike.re B'.det

theorem **inv_real_smul_of_posDef** {B : Matrix n n 𝕜} (hB : B.PosDef) {c : ℝ} (hc : c ≠ 0) : (c • B)⁻¹ = c⁻¹ • B⁻¹

theorem **le_trace_smul_one_of_posSemidef** {A : Matrix n n 𝕜} (hA : A.PosSemidef) : A ≤ A.trace • (1 : Matrix n n 𝕜)

theorem **smul_eq_re_smul_of_nonneg** {c : 𝕜} (hc : 0 ≤ c) (A : Matrix n n 𝕜) : c • A = RCLike.re c • A

theorem **le_re_trace_smul_one_of_posSemidef** {A : Matrix n n 𝕜} (hA : A.PosSemidef) : A ≤ RCLike.re A.trace • (1 : Matrix n n 𝕜)

theorem **le_re_trace_mul_smul_inv** {l b : Matrix n n 𝕜} (hl : l.PosDef) (hb : b.PosSemidef) : b ≤ RCLike.re (l ∗ b).trace • l⁻¹

theorem **norm_sq_apply_le_of_posSemidef** {A : Matrix n n 𝕜} (hA : A.PosSemidef) (i j : n) : ‖A i j‖ ^ 2 ≤ RCLike.re (A i i) ∗ RCLike.re (A j j)

theorem **inv_smul_one_le_inv_of_le_smul_one** {A : Matrix n n 𝕜} (hA : A.PosDef) {c : ℝ} (hc : 0 < c) (h : A ≤ c • (1 : Matrix n n 𝕜)) : c⁻¹ • (1 : Matrix n n 𝕜) ≤ A⁻¹

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.SummableLatticeExpNegPiMulNormSqDiv

theorem **summable_lattice_exp_neg_mul_norm_sq** {a : ℝ} (ha : 0 < a) : Summable fun x : L ↦ Real.exp (-a ∗ ‖(x : E)‖ ^ 2)

theorem **exp_neg_mul_norm_add_sq_le** {a R : ℝ} (ha : 0 ≤ a) (x : E) {u : E} (hu : ‖u‖ ≤ R) : Real.exp (-a ∗ ‖x + u‖ ^ 2) ≤ Real.exp (a ∗ R ^ 2) ∗ Real.exp (-(a / 2) ∗ ‖x‖ ^ 2)

theorem **summable_lattice_exp_neg_mul_norm_add_sq** {a : ℝ} (ha : 0 < a) (u : E) : Summable fun x : L ↦ Real.exp (-a ∗ ‖(x : E) + u‖ ^ 2)

theorem **continuous_tsum_lattice_translate** [CompleteSpace V] {f : E → V} (hc : Continuous f) (hdom : ∀ R : ℝ, ∃ g : L → ℝ, Summable g ∧ ∀ (x : L) (u : E), ‖u‖ ≤ R → ‖f (x + u)‖ ≤ g x) : Continuous fun u : E ↦ ∑' x : L, f (x + u)

theorem **periodic_tsum_lattice_translate** (f : E → V) {z : E} (hz : z ∈ L) : Function.Periodic (fun u : E ↦ ∑' x : L, f (x + u)) z

theorem **integrable_of_lattice_translate_dominated** {f : E → V} (hc : Continuous f) (hdom : ∀ R : ℝ, ∃ g : L → ℝ, Summable g ∧ ∀ (x : L) (u : E), ‖u‖ ≤ R → ‖f (x + u)‖ ≤ g x) : Integrable f

theorem **latticeFourierCoeff_tsum_translate** [CompleteSpace V] {f : E → V} (hc : Continuous f) (hdom : ∀ R : ℝ, ∃ g : L → ℝ, Summable g ∧ ∀ (x : L) (u : E), ‖u‖ ≤ R → ‖f (x + u)‖ ≤ g x) {y : E} (hy : y ∈ dualLattice L) : latticeFourierCoeff L (fun u : E ↦ ∑' x : L, f (x + u)) y = (ZLattice.covolume L)⁻¹ • 𝓕 f y

theorem **hasSum_poisson_lattice** [CompleteSpace V] {f : E → V} (hc : Continuous f) (hdom : ∀ R : ℝ, ∃ g : L → ℝ, Summable g ∧ ∀ (x : L) (u : E), ‖u‖ ≤ R → ‖f (x + u)‖ ≤ g x) (hs : Summable fun y : dualLattice L ↦ ‖𝓕 f y‖) (u : E) : HasSum (fun y : dualLattice L ↦ (ZLattice.covolume L)⁻¹ • e[⟪(y : E), u⟫] • 𝓕 f y) (∑' x : L, f (x + u))

theorem **tsum_poisson_lattice** [CompleteSpace V] {f : E → V} (hc : Continuous f) (hdom : ∀ R : ℝ, ∃ g : L → ℝ, Summable g ∧ ∀ (x : L) (u : E), ‖u‖ ≤ R → ‖f (x + u)‖ ≤ g x) (hs : Summable fun y : dualLattice L ↦ ‖𝓕 f y‖) (u : E) : ∑' x : L, f (x + u) = (ZLattice.covolume L)⁻¹ • ∑' y : dualLattice L, e[⟪(y : E), u⟫] • 𝓕 f y

theorem **norm_cexp_neg_pi_mul_norm_sq_add** (b : ℂ) (w v : E) : ‖Complex.exp (-π ∗ b ∗ ‖v‖ ^ 2 + 2 ∗ π ∗ Complex.I ∗ ⟪w, v⟫)‖ = Real.exp (-(π ∗ b.re) ∗ ‖v‖ ^ 2)

theorem **norm_cexp_neg_pi_mul_norm_sq_div** (b : ℂ) (v : E) : ‖Complex.exp (-π ∗ ‖v‖ ^ 2 / b)‖ = Real.exp (-(π ∗ b⁻¹.re) ∗ ‖v‖ ^ 2)

theorem **fourier_cexp_neg_pi_mul_norm_sq_add** {b : ℂ} (hb : 0 < b.re) (w y : E) : 𝓕 (fun v : E ↦ Complex.exp (-π ∗ b ∗ ‖v‖ ^ 2 + 2 ∗ π ∗ Complex.I ∗ ⟪w, v⟫)) y = b ^ (-(finrank ℝ E / 2 : ℂ)) ∗ Complex.exp (-π ∗ ‖y - w‖ ^ 2 / b)

theorem **summable_norm_lattice_cexp_gaussian** {b : ℂ} (hb : 0 < b.re) (w u : E) : Summable fun x : L ↦ ‖Complex.exp (-π ∗ b ∗ ‖(x : E) + u‖ ^ 2 + 2 ∗ π ∗ Complex.I ∗ ⟪w, (x : E) + u⟫)‖

theorem **summable_norm_lattice_cexp_gaussian_div** {b : ℂ} (hb : 0 < b.re) (w : E) : Summable fun y : L ↦ ‖Complex.exp (-π ∗ ‖(y : E) - w‖ ^ 2 / b)‖

theorem **poisson_cexp_gaussian_lattice** {b : ℂ} (hb : 0 < b.re) (w u : E) : (Summable fun x : L ↦ ‖Complex.exp (-π ∗ b ∗ ‖(x : E) + u‖ ^ 2 + 2 ∗ π ∗ Complex.I ∗ ⟪w, (x : E) + u⟫)‖) ∧ (Summable fun y : dualLattice L ↦ ‖Complex.exp (-π ∗ ‖(y : E) - w‖ ^ 2 / b) ∗ Complex.exp (2 ∗ π ∗ Complex.I ∗ ⟪(y : E), u⟫)‖) ∧ ∑' x : L, Complex.exp (-π ∗ b ∗ ‖(x : E) + u‖ ^ 2 + 2 ∗ π ∗ Complex.I ∗ ⟪w, (x : E) + u⟫) = (ZLattice.covolume L : ℂ)⁻¹ ∗ b ^ (-(finrank ℝ E / 2 : ℂ)) ∗ ∑' y : dualLattice L, Complex.exp (-π ∗ ‖(y : E) - w‖ ^ 2 / b) ∗ Complex.exp (2 ∗ π ∗ Complex.I ∗ ⟪(y : E), u⟫)

theorem **poisson_gaussian_lattice** {t : ℝ} (ht : 0 < t) (u : E) : (Summable fun x : L ↦ Real.exp (-π * t * ‖(x : E) + u‖ ^ 2)) ∧ (Summable fun y : dualLattice L ↦ ‖(Real.exp (-π * ‖(y : E)‖ ^ 2 / t) : ℂ) * Complex.exp (2 * π * Complex.I * ⟪(y : E), u⟫)‖) ∧ ((∑' x : L, Real.exp (-π * t * ‖(x : E) + u‖ ^ 2) : ℝ) : ℂ) = (((ZLattice.covolume L)⁻¹ * t ^ (-(finrank ℝ E / 2 : ℝ)) : ℝ) : ℂ) * ∑' y : dualLattice L, (Real.exp (-π * ‖(y : E)‖ ^ 2 / t) : ℂ) * Complex.exp (2 * π * Complex.I * ⟪(y : E), u⟫)

theorem **summable_lattice_exp_neg_pi_mul_norm_sq_div** {t : ℝ} (ht : 0 < t) : Summable fun y : L ↦ Real.exp (-π * ‖(y : E)‖ ^ 2 / t)

theorem **poisson_gaussian_lattice_cos** {t : ℝ} (ht : 0 < t) (u : E) : (Summable fun y : dualLattice L ↦ Real.exp (-π * ‖(y : E)‖ ^ 2 / t) * Real.cos (2 * π * ⟪(y : E), u⟫)) ∧ ∑' x : L, Real.exp (-π * t * ‖(x : E) + u‖ ^ 2) = (ZLattice.covolume L)⁻¹ * t ^ (-(finrank ℝ E / 2 : ℝ)) * ∑' y : dualLattice L, Real.exp (-π * ‖(y : E)‖ ^ 2 / t) * Real.cos (2 * π * ⟪(y : E), u⟫)

theorem **tsum_gaussian_lattice_eq_dualLattice** {t : ℝ} (ht : 0 < t) : ∑' x : L, Real.exp (-π * t * ‖(x : E)‖ ^ 2) = (ZLattice.covolume L)⁻¹ * t ^ (-(finrank ℝ E / 2 : ℝ)) * ∑' y : dualLattice L, Real.exp (-π * ‖(y : E)‖ ^ 2 / t)

theorem **tsum_gaussian_lattice_add_le** {t : ℝ} (ht : 0 < t) (u : E) : ∑' x : L, Real.exp (-π * t * ‖(x : E) + u‖ ^ 2) ≤ ∑' x : L, Real.exp (-π * t * ‖(x : E)‖ ^ 2)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.SummablePiProd

theorem **summable_pi_prod** {ι : Type u_1} [inst : Fintype ι] {J : ι → Type u_2} {u : (i : ι) → J i → ℝ} (h0 : ∀ (i : ι) (j : J i), (0 : ℝ) ≤ u i j) (hs : ∀ (i : ι), Summable (u i)) : Summable fun (t : (i : ι) → J i) ↦ ∏ i : ι, u i (t i)

Development.AbelianVarietyTheory.Proofs.CovolumeBounds.XiExistsOfBesselXiIntegrable

theorem **xiExists_of_besselXiIntegrable** : BesselXiIntegrable → Formalism.FourierDecayAlt.B.XiExists

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AbelianSchemeHomExtendsDedekindOfIsGrpPullbackAl

theorem **abelianScheme_hom_extends_dedekind_of_isGrpPullback** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (G G' : Over (Sp R)) [GrpObj G] [GrpObj G'] (G₀ G₀' : Over (Sp K)) [GrpObj G₀] [GrpObj G₀'] (φ : G₀.left ⟶ G.left) (φ' : G₀'.left ⟶ G'.left), IsAbelianScheme G → IsAbelianScheme G' → IsGrpPullback G₀ G (spMap (algebraMap R K)) φ → IsGrpPullback G₀' G' (spMap (algebraMap R K)) φ' → ∀ f : G₀ ⟶ G₀', IsMonHom f → ∃ F : G ⟶ G', IsMonHom F ∧ f.left ≫ φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AbelianSchemeHomExtendsDedekindOfIsGrpPullbackAlStatement

theorem **_root_.GlobalTheory.T.abelianScheme_hom_extends_dedekind_of_isGrpPullback_alt** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (G G' : Over (Sp R)) [GrpObj G] [GrpObj G'] (G₀ G₀' : Over (Sp K)) [GrpObj G₀] [GrpObj G₀'] (φ : G₀.left ⟶ G.left) (φ' : G₀'.left ⟶ G'.left), IsAbelianScheme G → IsAbelianScheme G' → IsGrpPullback G₀ G (spMap (algebraMap R K)) φ → IsGrpPullback G₀' G' (spMap (algebraMap R K)) φ' → ∀ f : G₀ ⟶ G₀', IsMonHom f → ∃ F : G ⟶ G

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AbelianSchemeSurjectiveOfOntoComplexPoints

theorem **surjective_of_points** {R K : Type} [CommRing R] [Field K] [IsAlgClosed K] (σ : R →+* K) (hσ : Function.Injective σ) {P A : Scheme.{0}} (a : A ⟶ Sp R) [LocallyOfFiniteType a] (ha : IsOpenMap a) (ψ : P ⟶ A) (hψ : IsClosed (Set.range ψ)) (h : ∀ x : Sp K ⟶ A, x ≫ a = spMap σ → ∃ p : Sp K ⟶ P, x = p ≫ ψ) : Surjective ψ

theorem **surjective_of_onto_complex_points** : ∀ (R : Type) [CommRing R] [IsDomain R] (σ : R →+* ℂ), Function.Injective σ → ∀ (A B : Over (Sp R)) [GrpObj A] [GrpObj B] (Φ : B ⟶ A), IsAbelianScheme A → IsAbelianScheme B → (∀ y : Over.mk (spMap σ) ⟶ A, ∃ x : Over.mk (spMap σ) ⟶ B, y = x ≫ Φ) → Surjective Φ.left ∧ ∀ (k : Type) [Field k] [IsAlgClosed k] (t : Sp k ⟶ Sp R) (y : Over.mk t ⟶ A), ∃ x : Over.mk t ⟶ B, y = x ≫ Φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AbelianSchemeSurjectiveOfOntoComplexPointsStatement

theorem **_root_.GlobalTheory.T.abelianScheme_surjective_of_onto_complex_points** : ∀ (R : Type) [CommRing R] [IsDomain R] (σ : R →+* ℂ), Function.Injective σ → ∀ (A B : Over (Sp R)) [GrpObj A] [GrpObj B] (Φ : B ⟶ A), IsAbelianScheme A → IsAbelianScheme B → (∀ y : Over.mk (spMap σ) ⟶ A, ∃ x : Over.mk (spMap σ) ⟶ B, y = x ≫ Φ) → Surjective Φ.left ∧ ∀ (k : Type) [Field k] [IsAlgClosed k] (t : Sp k ⟶ Sp R) (y : Over.mk t ⟶ A), ∃ x : Over.mk t ⟶ B, y = x ≫ Φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AppellHumbertTrivialOfNowhereVanishing

theorem **_root_.GlobalTheory.AppellHumbertTrivial.eq_zero_of_gaussian_decay** (f : ℂ → ℂ) (hf : Differentiable ℂ f) (C a : ℝ) (ha : a < 0) (h : ∀ t, ‖f t‖ ≤ C * Real.exp (‖t‖ ^ 2 * a)) : f 0 = 0

theorem **_root_.GlobalTheory.AppellHumbertTrivial.exists_bound_of_periodic** {g : ℕ} (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ] (hΛ : Submodule.span ℝ (Λ : Set (Fin g → ℂ)) = ⊤) {F : Type} [NormedAddCommGroup F] (φ : (Fin g → ℂ) → F) (hφ : Continuous φ) (hp : ∀ (l : Λ) z, φ (z + l) = φ z) : Bornology.IsBounded (Set.range φ)

theorem **_root_.GlobalTheory.AppellHumbertTrivial.hermForm_add_left** (B : Matrix (Fin g) (Fin g) ℂ) (x y w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B (x + y) w = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B x w + _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B y w

theorem **_root_.GlobalTheory.AppellHumbertTrivial.hermForm_add_right** (B : Matrix (Fin g) (Fin g) ℂ) (x y w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B w (x + y) = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B w x + _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B w y

theorem **_root_.GlobalTheory.AppellHumbertTrivial.hermForm_smul_left** (B : Matrix (Fin g) (Fin g) ℂ) (t : ℂ) (x w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B (t • x) w = t * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B x w

theorem **_root_.GlobalTheory.AppellHumbertTrivial.hermForm_smul_right** (B : Matrix (Fin g) (Fin g) ℂ) (t : ℂ) (x w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B w (t • x) = star t * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B w x

theorem **_root_.GlobalTheory.AppellHumbertTrivial.hermForm_swap** {B : Matrix (Fin g) (Fin g) ℂ} (hB : B.IsHermitian) (x w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B w x = star (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm

B x w)

theorem **_root_.GlobalTheory.AppellHumbertTrivial.continuous_hermForm_self** (B : Matrix (Fin g) (Fin g) ℂ) : Continuous fun z => _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B z z

theorem **_root_.GlobalTheory.AppellHumbertTrivial.re_hermForm_add_self** {B : Matrix (Fin g) (Fin g) ℂ} (hB : B.IsHermitian) (x y : Fin g → ℂ) : (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B (x + y) (x + y)).re = (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B x x).re + 2 * (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B x y).re + (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B y y).re

theorem **_root_.GlobalTheory.appellHumbert_trivial_of_nowhere_vanishing** : ∀ (g : ℕ) (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ], Submodule.span ℝ (Λ : Set (Fin g → ℂ)) = ⊤ → ∀ (B : Matrix (Fin g) (Fin g) ℂ) (c : Λ → ℂ) (r : (Fin g → ℂ) → ℂ), B.IsHermitian → (∀ l, ‖c l‖ = 1) → Differentiable ℂ r → (∀ z, r z ≠ 0) → (∀ (l : Λ) z, r (z + l) = c l * Complex.exp (Real.pi * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B z l + Real.pi / 2 * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm B l l) * r z) → B = 0 ∧ (∀ l, c l = 1) ∧ ∀ z, r z = r 0

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AppellHumbertUnifTangentWeilPairing

theorem **eq_top_of_add_discrete** {g : ℕ} (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ] (W : Submodule ℂ (Fin g → ℂ)) (h : ∀ z : Fin g → ℂ, ∃ l : Λ, z - (l : Fin g → ℂ) ∈ W) : W = ⊤

theorem **dim_eq** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g g' : ℕ} (u : HolUnif A L g) (u' : HolUnif A L g') (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (hF : ∀ z, u'.ex (F z) = u.ex z) : g = g'

theorem **tangent_and_weil_of_isAH** : ThetaUnifAV → ∀ (P : Out.PolAV (Sp ℂ)) (g : ℕ) (u : HolUnif P.A P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ), u.IsAH Hm χ → ∃ Ef : LinearMap.BilinForm ℤ u.Λ, (∀ a b : u.Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) ∧ Function.Bijective Ef ∧ g = P.g ∧ (∀ v, IsTangent v → ∃! w, u.IsDiff w v) ∧ (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) ∧ ∀ (k : ℕ) (a b : u.Λ) (c : ℂˣ), 0 < k → (c : ℂ) = Complex.exp (-(2 * Real.pi * Complex.I) * (Ef a b : ℂ) / k) → Out.WeilIs P.A P.L k (u.ex ((k : ℂ)⁻¹ • (a : Fin g → ℂ))) (u.ex ((k : ℂ)⁻¹ • (b : Fin g → ℂ))) (unitOn c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.AppellHumbertUnifTangentWeilPairingStatement

theorem **_root_.GlobalTheory.T.appellHumbert_unif_tangent_weilPairing** : ThetaUnifAV → ∀ (P : Out.PolAV (Sp ℂ)) (g : ℕ) (u : HolUnif P.A P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ), u.IsAH Hm χ → ∃ Ef : LinearMap.BilinForm ℤ u.Λ, (∀ a b : u.Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) ∧ Function.Bijective Ef ∧ g = P.g ∧ (∀ v, IsTangent v → ∃! w, u.IsDiff w v) ∧ (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) ∧ ∀ (k : ℕ) (a b : u.Λ) (c : ℂˣ), 0 < k → (c : ℂ) = Complex.exp (-(2 * Real.pi * Complex.I) * (Ef a b : ℂ) / k) → Out.WeilIs P.A P.L k (u.ex ((k : ℂ)⁻¹ • (a : Fin g → ℂ))) (u.ex ((k : ℂ)⁻¹ • (b : Fin g → ℂ))) (unitOn c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ArchimedeanHypothesisLowerBoundCmPoint

theorem **step5Arch_cm** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] [DecidableEq (E →+* ℂ)] (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, ∀ G : MvPolynomial (FrVar Dw.mult) z.K, IsFrPoly F z.y G → ∀ σ : z.K →+* ℂ, Step5Arch z.e z.Tl.dual z.LamL z.ℓ z.U G (fun b i => a (b, i)) n La (supC Xo c U ξ (goodOf c (σ.comp (algebraMap c.F z.K))) a n F) σ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ArchimedeanHypothesisLowerBoundCmPointStatement

theorem **_root_.GlobalTheory.T.archimedeanHypothesis_lowerBound_cmPoint** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] [DecidableEq (E →+* ℂ)] (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, ∀ G : MvPolynomial (FrVar Dw.mult) z.K, IsFrPoly F z.y G → ∀ σ : z.K →+* ℂ, Step5Arch z.e z.Tl.dual z.LamL z.ℓ z.U G (fun b i => a (b, i)) n La (supC Xo c U ξ (goodOf c (σ.comp (algebraMap c.F z.K))) a n F) σ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.BarEqOfLamEqMultiplier

theorem **bar_eq_of_lamEq_multiplier** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K : Type) [Field K] [NumberField K] (P B : Out.PolAV (Sp (𝒪 K))) (act : OAct (𝒪 E) P.A) (f : P.A ⟶ B.A) (h : 𝒪 E), 1 ≤ P.g → (∀ a, LamEq P.L (act.act a) (𝟙 P.A) P.L (𝟙 P.A) (act.act (H.bar a))) → LamEq B.L f f P.L (act.act h) (𝟙 P.A) → H.bar h = h

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ChartCoverFOfSecBaseChange

theorem **univFrame_borelPairs** : Out.SecBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (p A : TRing c.F) (j : Sp p.R ⟶ Xo.XF c) (y₀ : FrPt Xo c p.R) (π : p.R →ₐ[c.F] A.R) (X : (∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A.R)) (s : (A.R)ˣ) (u : FrPt Xo c A.R), OpenChart j y₀ → UnivFrame y₀ π X s u → ∃ (B : TRing c.F) (uB uB' : FrPt Xo c B.R) (tB : Ix Dw → (B.R)ˣ) (q q' : A.R →ₐ[c.F] B.R), FrMap q u uB ∧ FrMap q' u uB' ∧ FrBor tB uB uB' ∧ ∀ (T : Type) [CommRing T] [Algebra c.F T] (a a' : A.R →ₐ[c.F] T) (z z' : FrPt Xo c T) (t : Ix Dw → Tˣ), FrMap a u z → FrMap a' u z' → FrBor t z z' → ∃ b : B.R →ₐ[c.F] T, b.comp q = a ∧ b.comp q' = a' ∧ ∀ i, b (tB i : B.R) = (t i : T)

theorem **ChartCover.exists_lift** {S : Type} [CommRing S] [Algebra c.F S] (w : FrPt Xo c S) : ∃ xF : Sp S ⟶ Xo.XF c, xF ≫ pullback.fst _ _ = w.x ∧ xF ≫ pullback.snd _ _ = spMap (algebraMap c.F S)

theorem **ChartCover.near** (hS : Out.SecBaseChangeSt) {R A : Type} [CommRing R] [Algebra c.F R] [CommRing A] [Algebra c.F A] {j : Sp R ⟶ Xo.XF c} {y₀ : FrPt Xo c R} {π : R →ₐ[c.F] A} {X : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A} {s : Aˣ} {u : FrPt Xo c A} (hj : OpenChart j y₀) (hu : UnivFrame y₀ π X s u) {S : Type} [CommRing S] [Algebra c.F S] (w : FrPt Xo c S) (xF : Sp S ⟶ Xo.XF c) (h1 : xF ≫ pullback.fst _ _ = w.x) (h2 : xF ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (p : Sp S) (hp : xF p ∈ Set.range j) : ∃ g : S, g ∉ p.asIdeal ∧ ∀ (T : Type) [CommRing T] [Algebra c.F T] (lam : S →ₐ[c.F] T), IsUnit (lam g) → ∃ (z : FrPt Xo c T) (a : A →ₐ[c.F] T), FrMap lam w z ∧ FrMap a u z

theorem **ChartCover.cls_loc** (hS : Out.SecBaseChangeSt) {R : Type} [CommRing R] [Algebra c.F R] {A : TRing c.F} {j : Sp R ⟶ Xo.XF c} {y₀ : FrPt Xo c R} {π : R →ₐ[c.F] A.R} {X : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A.R} {s : (A.R)ˣ} {u : FrPt Xo c A.R} (hj : OpenChart j y₀) (hu : UnivFrame y₀ π X s u) {S : Type} [CommRing S] [Algebra c.F S] (w : FrPt Xo c S) (xF : Sp S ⟶ Xo.XF c) (h1 : xF ≫ pullback.fst _ _ = w.x) (h2 : xF ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (m : Ideal S) [hm : m.IsPrime] (hp : xF ((m, hm) : PrimeSpectrum S) ∈ Set.range j) : ∃ cw : A.R →ₐ[c.F] Localization.AtPrime m, ChartCls (K

theorem **chartCoverF_of_secBaseChange** : Out.SecBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), ChartCoverF Xo c

theorem **evR_frMap_of_isReduced** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (f : Rul Xo c) (hf : IsNatural (K

theorem **evR_frBor** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S : Type) [CommRing S] [Algebra c.F S] [_root_.IsReduced S] (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (f : Rul Xo c) (hf : f ∈ sec Xo c a n) (t : Formalism.L2.Pj.Ix Dw → Sˣ) (y y' : FrPt Xo c S) (hb : FrBor t y y') : evR Xo c f y' = wchar (Formalism.L2.Pj.ex Dw a n) t * evR Xo c f y

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ClausesIffExistsUnique

theorem **reading_iff** {L : LB X} (hL : L.IsLB) {L' : LB X'} (g : X' ⟶ Y) (φ : Y ⟶ X) (wP : LB.Iso L' (L.pull (g ≫ φ))) (hw : wP.IsGen) (hL' : L'.IsLB) (th : LB.Sec L') (s : LB.Sec (LB.LinComb L (K

theorem **lettersEquiv_fields** (L : LamLetters o) : (lettersEquiv o L).m = L.m ∧ HEq (lettersEquiv o L).e L.e ∧ (∀ k, (lettersEquiv o L).src k = srcEquiv o (L.src k)) ∧ (lettersEquiv o L).U = L.U ∧ (lettersEquiv o L).ℓ = L.ℓ

theorem **isLetter_iff** (L : LamLetters o) (k : Fin o.xB.P.g) (w : epsR o.K₁ ⟶ o.xB.P.A) : L.IsLetter k w ↔ (lettersEquiv o L).IsLetter k w

theorem **isGood_iff** (L : LamLetters o) : L.IsGood ↔ (lettersEquiv o L).IsGood

theorem **gx_isPullback** {S : Type} [CommRing S] (x : Sp S ⟶ Xo.X) : IsPullback (gxFst x) (gxSnd x) Xo.univ.t.G.hom x

theorem **toFibre_ext** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {A : Over (Sp S)} {g g' : A.left ⟶ Gx x} (h : g ≫ gxFst x = g' ≫ gxFst x) (h₁ : g ≫ gxSnd x = A.hom) (h₂ : g' ≫ gxSnd x = A.hom) : g = g'

theorem **existsUnique_iso** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {A : Over (Sp S)} {ψ : A.left ⟶ Xo.univ.t.G.left} (hp : IsPullback ψ A.hom Xo.univ.t.G.hom x) : ∃! eY : A ≅ Over.mk (gxSnd x), eY.hom.left ≫ gxFst x = ψ

theorem **tie_iff** (o : IndefiniteRowSideFullHecke.LatticeLetterSources.CMObj Dw c) : letI

theorem **fdeg_host** (D : D7) : fdeg D.host.E = D.R.f

theorem **clauses_iff** (T : Core D) (j : ℕ) (La : ℝ) (k : (T.U j T.σ₀).Comp) (x₀ : ((T.U j T.σ₀).X k).dom) : letI

theorem **clauses_iff_existsUnique** (T : Core D) (j : ℕ) (La : ℝ) (k : (T.U j T.σ₀).Comp) (x₀ : ((T.U j T.σ₀).X k).dom) : letI

theorem **conclusion_iff** (T : Core D) (j : ℕ) (La : ℝ) (k : (T.U j T.σ₀).Comp) (x₀ : ((T.U j T.σ₀).X k).dom) : letI

theorem **lattice_iff** {T : Core D} (h : LawsCore T) {j : ℕ} (hj : InR D j) (k : (T.U j T.σ₀).Comp) (Xx : DataX D.host.E D.R.r) (hX : Xx.toData = (T.U j T.σ₀).X k) : (∀ a ∈ D.host.AA, ∀ v ∈ Xx.L', a • v ∈ Xx.Λ) ↔ Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u}

theorem **frame_eq** {r : ℕ} (Xx : DataX E r) {X : Data E r} (hX : Xx.toData = X) (x : Frame X.Ψ r) : (hX ▸ x : Frame Xx.Ψ r) = IndefiniteRowSideFullHecke.LatticeLetterSources.frameOf Xx hX x

theorem **lamPtSt_at_iff** (T : Core D) (j : ℕ) (La : ℝ) : LamPtSt T j La ↔ (LawsCore T → InR D j → ∀ (k : (T.U j T.σ₀).Comp) (Xx : DataX D.host.E D.R.r) (hX : Xx.toData = (T.U j T.σ₀).X k), Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u} → ∀ (x₀ : ((T.U j T.σ₀).X k).dom) (lam : Fin D.R.r → Fin D.R.r → D.host.E), LamBasis Xx La (IndefiniteRowSideFullHecke.LatticeLetterSources.frameOf Xx hX x₀.1) lam → letI

theorem **lamPtSt_iff** (T : Core D) (j : ℕ) : (∀ La, LamPtSt T j La) ↔ IndefiniteRowSideFullHecke.LatticeLetterSources.LamPtSt T j

theorem **comparison** : (∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' : ℕ) (o : LamObj c nΨ n') (L : LamLetters o), (∀ k w, L.IsLetter k w ↔ (lettersEquiv o L).IsLetter k w) ∧ (L.IsGood ↔ (lettersEquiv o L).IsGood)) ∧ (∀ (D : D7) (T : Core D) (j : ℕ) (La : ℝ) (k : (T.U j T.σ₀).Comp) (x₀ : ((T.U j T.σ₀).X k).dom), letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmActionPolarisedCurveAdjointConj

theorem **linear_eq_of_ex** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} (u : HolUnif A L g) (F G : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : ∀ z, u.ex (F z) = u.ex (G z)) : F = G

theorem **exists_scalar** {g : ℕ} (hg : g = 1) (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) : ∃ c : ℂ, ∀ z, F z = c • z

theorem **map_bar** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (j : OK H →+* ℂ) (x y : OK H) (h : (y : 𝒪 E) = H.bar x) : j y = starRingEnd ℂ (j x)

theorem **lamEq_act_bar** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsDedekindDomain R] (σ : R →+* ℂ) (P : Out.PolAV (Sp R)) (a : OAct (OK H) P.A), Function.Injective σ → P.g = 1 → ∀ x y : OK H, (y : 𝒪 E) = H.bar x → LamEq P.L (a.act x) (𝟙 P.A) P.L (𝟙 P.A) (a.act y)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmActionPolarisedCurveAdjointConjStatement

theorem **_root_.GlobalTheory.T.cmAction_polarisedCurve_adjoint_conj** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsDedekindDomain R] (σ : R →+* ℂ) (P : Out.PolAV (Sp R)) (a : OAct (OK H) P.A), Function.Injective σ → P.g = 1 → ∀ x y : OK H, (y : 𝒪 E) = H.bar x → LamEq P.L (a.act x) (𝟙 P.A) P.L (𝟙 P.A) (a.act y)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmApproximationTransference

theorem **exists_short_family** {V : Type*} [NormedAddCommGroup V] [InnerProductSpace ℝ V] [FiniteDimensional ℝ V] {ι : Type*} [Fintype ι] [DecidableEq ι] (b : Basis ι ℝ V) (bd : ι → V) (hd : ∀ i j, inner ℝ (b i) (bd j) = if i = j then 1 else 0) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ x ∈ span ℤ (Set.range bd), x ≠ 0 → lam ≤ ‖x‖) : ∃ y : Fin (finrank ℝ V) → V, (∀ i, y i ∈ span ℤ (Set.range b)) ∧ LinearIndependent ℝ y ∧ ∀ i, ‖y i‖ ≤ (finrank ℝ V : ℝ) ^ 2 / lam

theorem **sumElim_bijective** (hσ : Function.Injective σ) (hcm : ∀ τ : K →+* ℂ, (∃ i, σ i = τ) ↔ ¬ ∃ i, σ i = ComplexEmbedding.conjugate τ) : Function.Bijective (Sum.elim σ fun i => ComplexEmbedding.conjugate (σ i))

theorem **finrank_eq_two_mul** (hσ : Function.Injective σ) (hcm : ∀ τ : K →+* ℂ, (∃ i, σ i = τ) ↔ ¬ ∃ i, σ i = ComplexEmbedding.conjugate τ) : finrank ℚ K = 2 * Fintype.card ι

theorem **trace_eq_two_mul_sum_re** (hσ : Function.Injective σ) (hcm : ∀ τ : K →+* ℂ, (∃ i, σ i = τ) ↔ ¬ ∃ i, σ i = ComplexEmbedding.conjugate τ) (y : K) : (Algebra.trace ℚ K y : ℝ) = 2 * ∑ i, (σ i y).re

theorem **exists_ratBasis** {L : Submodule (𝒪 K) (Fin r → K)} (hL : IsLattice L) : ∃ (n : ℕ) (b : Basis (Fin n) ℚ (Fin r → K)), ∀ x, x ∈ L ↔ x ∈ span ℤ (Set.range b)

theorem **h_add_left** (S : HermForm K r) (x x' y : Fin r → K) : S.h (x + x') y = S.h x y + S.h x' y

theorem **h_add_right** (S : HermForm K r) (x y y' : Fin r → K) : S.h x (y + y') = S.h x y + S.h x y'

theorem **h_smul_left** (S : HermForm K r) (a : K) (x y : Fin r → K) : S.h (a • x) y = S.c a * S.h x y

theorem **h_smul_right** (S : HermForm K r) (a : K) (x y : Fin r → K) : S.h x (a • y) = a * S.h x y

theorem **traceForm_apply** (S : HermForm K r) (x y : Fin r → K) : traceForm S x y = Algebra.trace ℚ K (S.h x y)

theorem **discr_smul_mem** [IsCMField K] {S : HermForm K r} {L : Submodule (𝓞 K) (Fin r → K)} (hL : IsSelfDual S L) {z : Fin r → K} (hz : ∀ y ∈ L, ∃ k : ℤ, traceForm S z y = k) : (discr K : K) • z ∈ L

theorem **embVec_add** (τ : K →+* ℂ) (x y : Fin r → K) : embVec τ (x + y) = embVec τ x + embVec τ y

theorem **norm_emb_sub_sq** (x : Fin r → K) (w : ι → Fin r → ℂ) : ‖(WithLp.toLp 2 fun ik : ι × Fin r => w ik.1 ik.2 : EuclideanSpace ℂ (ι × Fin r)) - emb σ Q x‖ ^ 2 = ∑ i, normSq (Q i *ᵥ embVec (σ i) x - w i)

theorem **norm_emb_sq** (x : Fin r → K) : ‖emb σ Q x‖ ^ 2 = majSq σ Q x

theorem **norm_embDual_sq** (x : Fin r → K) : ‖embDual p σ Q x‖ ^ 2 = 4 * majSq σ Q x

theorem **majSq_intCast_smul** (d : ℤ) (x : Fin r → K) : majSq σ Q ((d : K) • x) = (d : ℝ) ^ 2 * majSq σ Q x

theorem **normSq_le_majSq** (i : ι) (x : Fin r → K) : normSq (Q i *ᵥ embVec (σ i) x) ≤ majSq σ Q x

theorem **inner_emb_embDual** {S : HermForm K r} (hσ : Function.Injective σ) (hcm : ∀ τ : K →+* ℂ, (∃ i, σ i = τ) ↔ ¬ ∃ i, σ i = ComplexEmbedding.conjugate τ) (hQ : ∀ i x y, σ i (S.h x y) = jForm p (Q i *ᵥ embVec (σ i) x) (Q i *ᵥ embVec (σ i) y)) (x y : Fin r → K) : inner ℝ (emb σ Q y) (embDual p σ Q x) = (traceForm S x y : ℝ)

theorem **cmApproximation_transference** : ReciprocityTheory.ChartRestriction.G.FD.CoveringRadiusStatement

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmCurveBaseChangeComp

theorem **lbIso_pull_comp** {X X' X'' : Scheme.{0}} {L : LB X} {L' : LB X'} {φ : X' ⟶ X} (h : LBIso (L.pull φ) L') (ψ : X'' ⟶ X') : LBIso (L.pull (ψ ≫ φ)) (L'.pull ψ)

theorem **dualMapR_comp** (f : R →+* R') (g : R' →+* R'') : spMap (Chart.dualMapR g) ≫ spMap (Chart.dualMapR f) = spMap (Chart.dualMapR (g.comp f))

theorem **isPolFibre_comp** {f : R →+* R'} {g : R' →+* R''} {P : Out.PolAV (Sp R)} {P' : Out.PolAV (Sp R')} {P'' : Out.PolAV (Sp R'')} {φ : P'.A.left ⟶ P.A.left} {ψ : P''.A.left ⟶ P'.A.left} (h : IsPolFibre f P P' φ) (h' : IsPolFibre g P' P'' ψ) : IsPolFibre (g.comp f) P P'' (ψ ≫ φ)

theorem **isVecOver_comp** {f : R →+* R'} {g : R' →+* R''} {A : Over (Sp R)} {A' : Over (Sp R')} {A'' : Over (Sp R'')} {φ : A'.left ⟶ A.left} {ψ : A''.left ⟶ A'.left} {v : epsR R ⟶ A} {v' : epsR R' ⟶ A'} {v'' : epsR R'' ⟶ A''} (h : IsVecOver f φ v v') (h' : IsVecOver g ψ v' v'') : IsVecOver (g.comp f) (ψ ≫ φ) v v''

theorem **exists_isPolFibre_lift** {f : R →+* R'} {g : R' →+* R''} {P : Out.PolAV (Sp R)} {P' : Out.PolAV (Sp R')} {P'' : Out.PolAV (Sp R'')} {φ : P'.A.left ⟶ P.A.left} {χ : P''.A.left ⟶ P.A.left} (h : IsPolFibre f P P' φ) (n : IsPolFibre (g.comp f) P P'' χ) : ∃ ψ : P''.A.left ⟶ P'.A.left, ψ ≫ φ = χ ∧ IsPolFibre g P' P'' ψ

theorem **isVecOver_cancel** {f : R →+* R'} {g : R' →+* R''} {A : Over (Sp R)} {A' : Over (Sp R')} {A'' : Over (Sp R'')} {φ : A'.left ⟶ A.left} {ψ : A''.left ⟶ A'.left} {v : epsR R ⟶ A} {v' : epsR R' ⟶ A'} {v'' : epsR R'' ⟶ A''} (hφ : IsPullback φ A'.hom A.hom (spMap f)) (hψ : IsPullback ψ A''.hom A'.hom (spMap g)) (h : IsVecOver f φ v v') (n : IsVecOver (g.comp f) (ψ ≫ φ) v v'') : IsVecOver g ψ v' v''

theorem **isLieNormSq_iff** {f : R →+* R'} {P : Out.PolAV (Sp R)} {P' : Out.PolAV (Sp R')} {φ : P'.A.left ⟶ P.A.left} {v : epsR R ⟶ P.A} {v' : epsR R' ⟶ P'.A} (h : IsPolFibre f P P' φ) (hv : IsVecOver f φ v v') (σ : R' →+* ℂ) (r : ℝ) : IsLieNormSq P' v' σ r ↔ IsLieNormSq P v (σ.comp f) r

theorem **ellMap_comp** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (R' : Type) [CommRing R'] (ω' : R') (C' : Ell H R' ω') (f : R →+* R') (φ : C'.P.A.left ⟶ C.P.A.left), EllMap C' C f φ → (∀ (R'' : Type) [CommRing R''] (ω'' : R'') (C'' : Ell H R'' ω'') (f' : R' →+* R'') (φ' : C''.P.A.left ⟶ C'.P.A.left), EllMap C'' C' f' φ' → EllMap C'' C (f'.comp f) (φ' ≫ φ)) ∧ (∀ σ : R' →+* ℂ, C.IsPrincipalAt (σ.comp f) → C'.IsPrincipalAt σ) ∧ ∀ (σ : R' →+* ℂ) (r : ℝ), Out.PolBaseChangeSt → (IsLieNormSq C'.P C'.v σ r ↔ IsLieNormSq C.P C.v (σ.comp f) r)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmCurveBaseChangeCompStatement

theorem **_root_.GlobalTheory.T.cmCurveBaseChange_comp** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (R' : Type) [CommRing R'] (ω' : R') (C' : Ell H R' ω') (f : R →+* R') (φ : C'.P.A.left ⟶ C.P.A.left), EllMap C' C f φ → (∀ (R'' : Type) [CommRing R''] (ω'' : R'') (C'' : Ell H R'' ω'') (f' : R' →+* R'') (φ' : C''.P.A.left ⟶ C'.P.A.left), EllMap C'' C' f' φ' → EllMap C'' C (f'.comp f) (φ' ≫ φ)) ∧ (∀ σ : R' →+* ℂ, C.IsPrincipalAt (σ.comp f) → C'.IsPrincipalAt σ) ∧ ∀ (σ : R' →+* ℂ) (r : ℝ), Out.PolBaseChangeSt → (IsLieNormSq C'.P C'.v σ r ↔ IsLieNormSq C.P C.v (σ.comp f) r)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmCurveLieCharacterOfAdjoinGenerator

theorem **isLieComb_one_iff** (c : R) (v w : epsR R ⟶ G) : IsLieComb ![c] ![v] w ↔ w.left = spMap (epsScaleR c) ≫ v.left

theorem **isLieComb_comp** {c : R} {v w : epsR R ⟶ G} (h : IsLieComb ![c] ![v] w) (f : G ⟶ G') : IsLieComb ![c] ![v ≫ f] (w ≫ f)

theorem **isTangent_comp** {v : epsR R ⟶ G} (h : IsTangent v) (f : G ⟶ G') [IsMonHom f] : IsTangent (v ≫ f)

theorem **spMap_epsScaleR_one** : spMap (epsScaleR (1 : R)) = 𝟙 _

theorem **isLieComb_mul** (x y : epsR R ⟶ G) : IsLieComb ![(1 : R), 1] ![x, y] (x * y)

theorem **exists_char** [IsCommMonObj G] {O : Type} [CommRing O] (act : OAct O G) {v : epsR R ⟶ G} (hv : IsLieBasis ![v]) : ∃ τ : O →+* R, (∀ c, IsLieComb ![τ c] ![v] (v ≫ act.act c)) ∧ ∀ c r, IsLieComb ![r] ![v] (v ≫ act.act c) → τ c = r

theorem **ell_char** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω), ∃ τ₀ : OK H →+* R, τ₀ (wK H) = ω ∧ ∀ c, IsLieComb ![τ₀ c] ![C.v] (C.v ≫ C.act.act c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmCurveLieCharacterOfAdjoinGeneratorStatement

theorem **_root_.GlobalTheory.T.cmCurve_lie_character_of_adjoinGenerator** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω), ∃ τ₀ : OK H →+* R, τ₀ (wK H) = ω ∧ ∀ c, IsLieComb ![τ₀ c] ![C.v] (C.v ≫ C.act.act c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmDatumFormSqrtNegDPurePeriod

theorem **cmDatum_form_sqrtNegD_pure_period** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (D : Datum H) {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) {ρ : ℂ} (hty : ∀ i, ty i (H.rt : E) = ρ) {x : D.La} (hx : x ≠ 0)

: ∃ r : ℝ, 0 < r ∧ ((D.EF (H.w0 • x) x : ℤ) : ℝ) = ρ.im * r

theorem **definite_datum_form_pos** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ), (D.q = 0 ∧ 0 < ((ε : ℤ) : ℝ) * H.rt0.im) ∨ (D.p = 0 ∧ ((ε : ℤ) : ℝ) * H.rt0.im < 0) → ∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmEigenLineDegreeFormula

theorem **exists_compositum** (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (M : IntermediateField ℚ ℂ) [NumberField M] : ∃ (k : IntermediateField ℚ ℂ) (_ : NumberField k) (j : K →+* k) (i : M →+* k), (∀ x, (j x : ℂ) = σ x) ∧ ∀ x, (i x : ℂ) = x

theorem **mapRingHom_injective** {K K' : Type} [Field K] [Field K'] (j : K →+* K') : Function.Injective (RingOfIntegers.mapRingHom j)

theorem **degSt** : CMCurve → ThetaUnifAV → Out.Laws → HomExtDvr → DegSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmEigenLineDegreeFormulaStatement

theorem **_root_.GlobalTheory.T.cmEigenLine_degreeFormula** : CMCurve → ThetaUnifAV → Out.Laws → HomExtDvr → DegSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmEllipticCurveExistsOfUniformisation

theorem **half_one_add_sqrt_quadratic** {K : Type} [Field K] [CharZero K] {d : ℕ} (h4 : d % 4 = 3) {r w : K} (hr : r ^ 2 = -(d : K)) (hw : 2 * w = 1 + r) : w * w + (((1 + d) / 4 : ℕ) : K) = w

theorem **cmCurve_of_uniformisation** : ThetaUnifAV → Out.PolBaseChangeSt → (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → CMCurve

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmEllipticCurveExistsOfUniformisationStatement

theorem **_root_.GlobalTheory.T.cmEllipticCurve_exists_of_uniformisation** : ThetaUnifAV → Out.PolBaseChangeSt → (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → CMCurve

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmExistsHolds

theorem **cmExists_holds** : GlobalTheory.CM.CMExistsSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmFamilyDescendsToNumberField

theorem **exists_field** (hT : ThetaUnifAV) (hB : Out.PolBaseChangeSt) (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (P Q : Out.PolAV (Sp k)) {G H : Over (Sp ℂ)} [GrpObj G] [GrpObj H] {φ : G.left ⟶ P.A.left} {m : H.left ⟶ Q.A.left} (hP : IsGrpPullback G P.A (spMap (algebraMap k ℂ)) φ) (hQ : IsGrpPullback H Q.A (spMap (algebraMap k ℂ)) m) (f : G ⟶ H) [IsMonHom f] : ∃ k₁ : IntermediateField ℚ ℂ, FiniteDimensional ℚ k₁ ∧ ∀ (k₂ : IntermediateField ℚ ℂ) (h₂ : k ≤ k₂), k₁ ≤ k₂ → ∀ (G₂ H₂ : Over (Sp k₂)) (φ₂ : G₂.left ⟶ P.A.left) (m₂ : H₂.left ⟶ Q.A.left) (φ₃ : G.left ⟶ G₂.left) (m₃ : H.left ⟶ H₂.left), φ₂ ≫ P.A.hom = G₂.hom ≫ spMap (IntermediateField.inclusion h₂).toRingHom → IsPullback m₂ H₂.hom Q.A.hom (spMap (IntermediateField.inclusion h₂).toRingHom) → φ₃ ≫ G₂.hom = G.hom ≫ spMap (algebraMap k₂ ℂ) → m₃ ≫ H₂.hom = H.hom ≫ spMap (algebraMap k₂ ℂ) → φ₃ ≫ φ₂ = φ → m₃ ≫ m₂ = m → ∃ f₂ : G₂ ⟶ H

theorem **exists_mul** {S S' : Scheme.{0}} {j : S' ⟶ S} {G H : Over S} [GrpObj H] {G' H' : Over S'} [GrpObj H'] {φ : G'.left ⟶ G.left} {m : H'.left ⟶ H.left} (w : φ ≫ G.hom = G'.hom ≫ j) (hm : IsGrpPullback H' H j m) {x' y' : G' ⟶ H'} {x y : G ⟶ H} (hx : x'.left ≫ m = φ ≫ x.left) (hy : y'.left ≫ m = φ ≫ y.left) : (x' * y').left ≫ m = φ ≫ (x * y).left

theorem **isFam_descends_to_numberField** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (ω : k) (C : Ell H k ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp k)) (L : Submodule (𝒪 E) D.La) (ω' : ℂ) (C' : Ell H ℂ ω') (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp ℂ)) (m : SABMap T'.t T.t (spMap (algebraMap k ℂ))) (ψ' : L → (C'.P.A ⟶ T'.t.G)), EllMap C' C (algebraMap k ℂ) φ → IsFam C' T' h L ψ' → ∃ k₁ : IntermediateField ℚ ℂ, FiniteDimensional ℚ k₁ ∧ k ≤ k₁ ∧ ∀ (k₂ : IntermediateField ℚ ℂ) (h₂ : k ≤ k₂), k₁ ≤ k₂ → ∀ (ω₂ : k₂) (C₂ : Ell H k₂ ω₂) (φ₂ : C₂.P.A.left ⟶ C.P.A.left) (φ₃ : C'.P.A.left ⟶ C₂.P.A.left) (T₂ : PAT H (Sp k₂)) (m₂ : SABMap T₂.t T.t (spMap (IntermediateField.inclusion h₂).toRingHom)) (m₃ : SABMap T'.t T₂.t (spMap (algebraMap k₂ ℂ))), EllMap C₂ C (IntermediateField.inclus

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmFamilyDescendsToNumberFieldStatement

theorem **_root_.GlobalTheory.T.cmFamily_descends_to_numberField** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (ω : k) (C : Ell H k ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp k)) (L : Submodule (𝒪 E) D.La) (ω' : ℂ) (C' : Ell H ℂ ω') (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp ℂ)) (m : SABMap T'.t T.t (spMap (algebraMap k ℂ))) (ψ' : L → (C'.P.A ⟶ T'.t.G)), EllMap C' C (algebraMap k ℂ) φ → IsFam C' T' h L ψ' → ∃ k₁ : IntermediateField ℚ ℂ, FiniteDimensional ℚ k₁ ∧ k ≤ k₁ ∧ ∀ (k₂ : IntermediateField ℚ ℂ) (h₂ : k ≤ k₂), k₁ ≤ k₂ → ∀ (ω₂ : k₂) (C₂ : Ell H k₂ ω₂) (φ₂ : C₂.P.A.left ⟶ C.P.A.left) (φ₃ : C'.P.A.left ⟶ C₂.P.A.left) (T₂ : PAT H (Sp k₂)) (m₂ : SABMap T₂.t T.t (spMap (IntermediateField.inclusion h₂).toRingHom)) (m₃ : SABMap T'.t T₂.t (spMap (algebraMap k₂ ℂ))), EllMap C₂ C

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmFamilyExtendsOverDedekindDomain

theorem **isFam_extends_dedekind** : Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (ω : R) (C : Ell H R ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (L : Submodule (𝒪 E) D.La) (ω' : K) (C' : Ell H K ω') (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp K)) (m : SABMap T'.t T.t (spMap (algebraMap R K))) (ψ' : L → (C'.P.A ⟶ T'.t.G)), EllMap C' C (algebraMap R K) φ → IsFam C' T' h L ψ' → ∃ ψ : L → (C.P.A ⟶ T.t.G), IsFam C T h L ψ ∧ ∀ a, (ψ' a).left ≫ m.φ = φ ≫ (ψ a).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmFamilyExtendsOverDedekindDomainStatement

theorem **_root_.GlobalTheory.T.cmFamily_extends_over_dedekindDomain** : Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (ω : R) (C : Ell H R ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (L : Submodule (𝒪 E) D.La) (ω' : K) (C' : Ell H K ω') (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp K)) (m : SABMap T'.t T.t (spMap (algebraMap R K))) (ψ' : L → (C'.P.A ⟶ T'.t.G)), EllMap C' C (algebraMap R K) φ → IsFam C' T' h L ψ' → ∃ ψ : L → (C.P.A ⟶ T.t.G), IsFam C T h L ψ ∧ ∀ a, (ψ' a).left ≫ m.φ = φ ≫ (ψ a).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmGeneratorSqQuadraticEq

theorem **cm_generator_sq_quadratic_eq** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : H.w0 * H.w0 = H.w0 - (((H.d + 1) / 4 : ℕ) : 𝒪 E)

theorem **main** : ∀ {E : Type} [inst : Field E] [inst_1 : NumberField E] [inst_2 : NumberField.IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) {a : NumberField.RingOfIntegers E}, a ∈ GlobalTheory.CM.OK H → ∃ (u : ℤ) (v : ℤ), a = (↑u : NumberField.RingOfIntegers E) + (↑v : NumberField.RingOfIntegers E) * ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host.w0 H

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmHostRootNegDiscriminantNeNegSelf

theorem **cmHost_rootNegDiscriminant_ne_zero** (H : Host E) : H.rt0 ≠ 0

theorem **cmHost_rootNegDiscriminant_ne_neg_self** (H : Host E) : H.rt0 ≠ -H.rt0

theorem **cmHost_conj_sqrtNegDisc** (H : Host E) : starRingEnd ℂ H.rt0 = -H.rt0

theorem **host_emb_rt_sq** (H : Host E) (τ : E →+* ℂ) : τ (H.rt : E) ^ 2 = H.rt0 ^ 2

theorem **host_mem_compl_psiSet** (H : Host E) (τ : E →+* ℂ) : τ ∈ (psiSet H)ᶜ ↔ τ (H.rt : E) = -H.rt0

theorem **card_psiSet_eq_card_compl** (H : Host E) : Nat.card (psiSet H) = Nat.card ((psiSet H)ᶜ : Set (E →+* ℂ))

theorem **card_psiSet** (H : Host E) : Nat.card (psiSet H) = Module.finrank ℚ E / 2

theorem **labK_eq_iff** (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (τ : E →+* ℂ) (a : E) (x : c.F) : labK c K τ a = algebraMap c.F K x ↔ τ a = (x : ℂ)

theorem **labK_injective** (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] : Function.Injective (labK c K)

theorem **labK_bijective** (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K] : Function.Bijective (labK c K)

theorem **labK_spec** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (l : Level H) (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K], Function.Bijective (labK c K) ∧ Splits E K ∧ (∀ τ, ∃ t : 𝒪 E →+* 𝒪 K, ∀ a : 𝒪 E, (t a : K) = labK c K τ (a : E)) ∧ ∃ ρ : 𝒪 K, ρ ^ 2 = -(H.d : 𝒪 K) ∧ (∀ τ, τ ∈ psiSet H ↔ labK c K τ (H.rt : E) = (ρ : K)) ∧ (∀ τ, τ ∈ (psiSet H)ᶜ ↔ labK c K τ (H.rt : E) = ((-ρ : 𝒪 K) : K)) ∧ Nat.card (psiSet H) = Module.finrank ℚ E / 2 ∧ Nat.card ((psiSet H)ᶜ : Set (E →+* ℂ)) = Module.finrank ℚ E / 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmLieLineDegreeOfEigenLineDegreeFormula

theorem **lamDegSt_of_degSt** : DegSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → LamDegSt H (logOm E χ).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmLieLineDegreeOfEigenLineDegreeFormulaStatement

theorem **_root_.GlobalTheory.T.cmLieLineDegree_of_eigenLineDegreeFormula** : DegSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → LamDegSt H (logOm E χ).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmLieStatementsIffLibrary

theorem **pnsmulPt_eq_frozen** {S : Scheme.{0}} {G : Over S} [GrpObj G] {T : Over S} (n : ℕ) (x : T ⟶ G) : pnsmulPt n x = Formalism.L2.Pj.abelian₂.pnsmulPt n x

theorem **LamObj.ofFrozen_toFrozen** (o : LamObj c nΨ n') : LamObj.ofFrozen o.toFrozen = o

theorem **LamObj.toFrozen_ofFrozen** (o : Formalism.L2.Pj.abelian₂.LamObj c nΨ n') : (LamObj.ofFrozen o).toFrozen = o

theorem **rosatiConj_iff_frozen** {K : Type} [Field K] [NumberField K] (A : PolAV (Sp (𝒪 K))) (oa : A.OAct (𝒪 E)) : RosatiConj A oa ↔ Formalism.L2.Pj.abelian₂.RosatiConj A oa

theorem **isIsogSum_toFrozen** (o : LamObj c nΨ n') : Formalism.L2.Pj.abelian₂.IsIsogSum o.toFrozen ↔ IsIsogSum o

theorem **polClause_toFrozen** (o : LamObj c nΨ n') : Formalism.L2.Pj.abelian₂.PolClause o.toFrozen ↔ PolClause o

theorem **fixed_of_lambdaBundle_objectCondition** (o : LamObj c nΨ n') (f : ℕ) (La : ℝ) : Formalism.L2.Pj.abelian₂.LamObjGood o.toFrozen f La ↔ LamObjGood o f La

theorem **lambdaBundle_objectCondition_of_fixed** (o : Formalism.L2.Pj.abelian₂.LamObj c nΨ n') (f : ℕ) (La : ℝ) : LamObjGood (LamObj.ofFrozen o) f La ↔ Formalism.L2.Pj.abelian₂.LamObjGood o f La

theorem **LamLetters.isLetter_toFrozen** {o : LamObj c nΨ n'} (L : LamLetters o) (k : Fin o.xB.P.g) (w : epsR o.K₁ ⟶ o.xB.P.A) : L.toFrozen.IsLetter k w ↔ L.IsLetter k w

theorem **LamLetters.isLetter_ofFrozen** {o : Formalism.L2.Pj.abelian₂.LamObj c nΨ n'} (L : Formalism.L2.Pj.abelian₂.LamLetters o) (k : Fin o.xB.P.g) (w : epsR o.K₁ ⟶ o.xB.P.A) : (LamLetters.ofFrozen L).IsLetter k w ↔ L.IsLetter k w

theorem **LamLetters.isGood_toFrozen** {o : LamObj c nΨ n'} (L : LamLetters o) : L.toFrozen.IsGood ↔ L.IsGood

theorem **LamLetters.isGood_ofFrozen** {o : Formalism.L2.Pj.abelian₂.LamObj c nΨ n'} (L : Formalism.L2.Pj.abelian₂.LamLetters o) : (LamLetters.ofFrozen L).IsGood ↔ L.IsGood

theorem **lambdaBundle_degree_iff** (H : Host E) (x : ℝ) : LamDegSt H x ↔ Formalism.L2.Pj.abelian₂.LamDegSt H x

theorem **lambdaBundle_index_iff** : LamIndSt ↔ Formalism.L2.Pj.abelian₂.LamIndSt

theorem **lambdaBundle_length_iff** : LamLenSt ↔ Formalism.L2.Pj.abelian₂.LamLenSt

theorem **lambdaBundle_conditions_iff** : (∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (x : ℝ), GlobalTheory.LamF.LamDegSt H x ↔ Formalism.L2.Pj.abelian₂.LamDegSt H x) ∧ (GlobalTheory.LamF.LamIndSt ↔ Formalism.L2.Pj.abelian₂.LamIndSt) ∧ (GlobalTheory.LamF.LamLenSt ↔ Formalism.L2.Pj.abelian₂.LamLenSt)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmLieStatementsIffLibraryStatement

theorem **_root_.GlobalTheory.T.cmLieStatements_iff_library** : (∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (x : ℝ), GlobalTheory.LamF.LamDegSt H x ↔ Formalism.L2.Pj.abelian₂.LamDegSt H x) ∧ (GlobalTheory.LamF.LamIndSt ↔ Formalism.L2.Pj.abelian₂.LamIndSt) ∧ (GlobalTheory.LamF.LamLenSt ↔ Formalism.L2.Pj.abelian₂.LamLenSt)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmPointLowerBoundFree

theorem **cmPoint_lowerBound_free** : ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (e : (Σ b, Fin (m b)) ≃ Fin x.P.g) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (G : MvPolynomial (FrVar m) K) (a : ∀ b, Fin (m b) → ℕ) (n r : ℕ) (La logOm : ℝ) (S : (K →+* ℂ) → ℝ), T.Good → abelian₂.Good → (∀ p,

KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → (∀ j, (ℓ j).ndeg = logOm / 2) → U.det ≠ 0 → G ≠ 0 → 1 ≤ La → 1 ≤ r → (∀ b, m b ≤ r) → (∀ σ, 0 < S σ) → (∀ p, Step5Loc e T.dual abelian₂ ℓ U G n p) → (∀ σ, Step5Arch e T.dual abelian₂ ℓ U G a n La (S σ) σ) → − ((pDeg logOm + (1 / 2) * Real.log La) * (((∑ b, ∑ i, a b i : ℕ) : ℝ) + (x.P.g : ℝ) * n)) − (((∑ b, ∑ i, a b i : ℕ) : ℝ) * Real.log ((r : ℝ) * La) ≤ (∑ σ : K →+* ℂ, Real.log (S σ)) / (Module.finrank ℚ K : ℝ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmPointLowerBoundOfCoefficientBound

theorem **cmLowerBound_of_coeffBound** : (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (H : (K →+* ℂ) → Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (La logOm : ℝ) (V : Type) [Fintype V] (c : V → Fin x.P.g) (G : MvPolynomial V K) (μ : V →₀ ℕ) (n : ℕ) (S : (K →+* ℂ) → ℝ) (Rr : ℝ), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → U.det ≠ 0 → 1 ≤ La → (∀ σ, (H σ).PosDef) → (∀ σ, abelian₂.n σ = Real.sqrt (H σ).det.re) → (∀ p : HeightOneSpectrum (𝒪 K), ∃ (cd : K) (cl : Fin x.P.g → K), IsLocGen p cd abelian₂.I ∧ (∀ j, IsLocGen p (cl j) (ℓ j).I) ∧ p.valuation K ((∏ j, cl j) * U.det) ≤ p.valuation K cd) → (∀ σ j, ReciprocityTheory.MultiplicativeHeights.hq (H σ) (ReciprocityTheory.MultiplicativeHeights.embVec σ (U j)) ≤ (La * (ℓ j).n σ) ^ 2) → (∀ j, (ℓ j)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmPointLowerBoundOfCoefficientBoundStatement

theorem **_root_.GlobalTheory.T.cmPoint_lower_bound_of_coefficient_bound** : (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (H : (K →+* ℂ) → Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (La logOm : ℝ) (V : Type) [Fintype V] (c : V → Fin x.P.g) (G : MvPolynomial V K) (μ : V →₀ ℕ) (n : ℕ) (S : (K →+* ℂ) → ℝ) (Rr : ℝ), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → U.det ≠ 0 → 1 ≤ La → (∀ σ, (H σ).PosDef) → (∀ σ, abelian₂.n σ = Real.sqrt (H σ).det.re) → (∀ p : HeightOneSpectrum (𝒪 K), ∃ (cd : K) (cl : Fin x.P.g → K), IsLocGen p cd abelian₂.I ∧ (∀ j, IsLocGen p (cl j) (ℓ j).I) ∧ p.valuation K ((∏ j, cl j) * U.det) ≤ p.valuation K cd) → (∀ σ j, ReciprocityTheory.MultiplicativeHeights.hq (H σ) (ReciprocityTheory.MultiplicativeHeights.embVec σ (U j)) ≤

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmPointObjectExistsOfLaws

theorem **additive_ext** {V L M : Type} [AddCommGroup V] [Module ℂ V] [Group M] {P : L → V} (hP : Submodule.span ℝ (Set.range P) = ⊤) {u v : V → M} (hu : ∀ z w, u (z + w) = u z * u w) (hv : ∀ z w, v (z + w) = v z * v w) (h : ∀ a (c : ℂ), u (c • P a) = v (c • P a)) (z : V) : u z = v z

theorem **exists_sum** {J V L : Type} [Fintype J] [AddCommGroup V] [Module ℂ V] {W : J → Type} [∀ j, AddCommGroup (W j)] [∀ j, Module ℂ (W j)] (F : ∀ j, W j →ₗ[ℂ] V) {P : L → V} (hP : Submodule.span ℝ (Set.range P) = ⊤) {n : ℕ} (hn : 0 < n) (h : ∀ a, ∃ x : ∀ j, W j, n • P a = ∑ j, F j (x j)) (z : V) : ∃ x : ∀ j, W j, z = ∑ j, F j (x j)

theorem **exists_ten_exp** (hT : ThetaUnifAV) (hL : Out.Laws) {R : Type} [CommRing R] [IsNoetherianRing R] {ω : R} (C : Ell H R ω) (σ : R →+* ℂ) {ω' : ℂ} (C' : Ell H ℂ ω') (π : C'.P.A.left ⟶ C.P.A.left) (u₀ : HolUnif C'.P.A C'.P.L C'.P.g) (c : ℂ) (D : Datum H) (ε : ℤˣ) (hm : EllMap C' C σ π) (hc : c ≠ 0) (hu : PureType.IsQuadLattice u₀.Λ H.d c) (hε : 0 < ((ε : ℤ) : ℝ) * ω'.im) (hD : (D.q = 0 ∧ 2 * ω' - 1 = H.rt0) ∨ (D.p = 0 ∧ 2 * ω' - 1 = -H.rt0)) : ∃ (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (t : IsTen C D h T) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (P : D.La →+ (Fin g → ℂ)) (e : (Fin g → ℂ) → (Over.mk (spMap σ) ⟶ T.t.G)), IsKForm D ε h ∧ IsTypeFn H D.p D.q ty ∧ IsExp T σ D ty P e ∧ (∀ a (z : ℂ), (e (z • P a)).left = (u₀.ex (fun _ => c * z)).left ≫ π ≫ (t.ten.tmul a).left) ∧ ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (f : D.La →+ V), (∀ a, f (H.w0 • a) = ω' • f a) → ∃ F : (Fin g → ℂ) →ₗ[ℂ] V, ∀

theorem **hasType_of_moduli** {D : Datum H} {l : Level H} (Xo : Moduli D l) (K₁ : IntermediateField ℚ ℂ) [NumberField K₁] (hk : KA l ≤ K₁) (ρ : 𝒪 K₁) (hρ : ((ρ : K₁) : ℂ) = H.rt0) (B : SAB H (Sp (𝒪 K₁))) (y : Sp K₁ ⟶ Xo.X) (t₀ : SAB H (Sp K₁)) (m : SABMap t₀ Xo.univ.t y) (m' : SABMap t₀ B (spMap (algebraMap (𝒪 K₁) K₁))) (hy : y ≫ Xo.s = spMap (IntermediateField.inclusion hk).toRingHom) : HasType H B.act ρ D.p D.q

theorem **cmExists** : CMCurve → CMPoint → ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → CMExistsSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmPointObjectExistsOfLawsStatement

theorem **_root_.GlobalTheory.T.cmPointObject_exists_of_laws** : CMCurve → CMPoint → ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → CMExistsSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmTensorSmoothOfRelativeDimension

theorem **finrank_fibre** (O M k : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] [Field k] [Algebra O k] : Module.finrank k (k ⊗[O] M) = Module.finrank O M

theorem **exists_isLieBasis** {k : Type} [Field k] (G : Over (Sp k)) [GrpObj G] [IsCommMonObj G] {n : ℕ} (h : SmoothOfRelativeDimension n G.hom) : ∃ v : Fin n → (epsR k ⟶ G), IsLieBasis v

theorem **dim_eq_field** {k O M : Type} [Field k] [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] {G A : Over (Sp k)} [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] {a : OAct O G} {b : OAct O A} (t : Serre.IsTensor A b G a M) (hG : SmoothOfRelativeDimension 1 G.hom) {n : ℕ} (hA : SmoothOfRelativeDimension n A.hom) : n = Module.finrank O M

theorem **smoothOfRelativeDimension** {S : Scheme.{0}} {O M : Type} [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] {G A : Over S} [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] {a : OAct O G} {b : OAct O A} (t : Serre.IsTensor A b G a M) (hab : IsAbelianScheme G) (hG : SmoothOfRelativeDimension 1 G.hom) : SmoothOfRelativeDimension (Module.finrank O M) A.hom

theorem **tensor_smoothOfRelativeDimension** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (A : Over (Sp R)) [GrpObj A] [IsCommMonObj A] (b : OAct (OK H) A), Nonempty (Serre.IsTensor A b C.P.A C.act D.La) → SmoothOfRelativeDimension (Module.finrank ℚ E / 2 * (D.p + D.q)) A.hom

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CmTensorSmoothOfRelativeDimensionStatement

theorem **_root_.GlobalTheory.T.cmTensor_smoothOfRelativeDimension** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (A : Over (Sp R)) [GrpObj A] [IsCommMonObj A] (b : OAct (OK H) A), Nonempty (Serre.IsTensor A b C.P.A C.act D.La) → SmoothOfRelativeDimension (Module.finrank ℚ E / 2 * (D.p + D.q)) A.hom

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CommutatorPairingsEqOfCubeLBPairingIso

theorem **exists_unitOn** {K : Type} [CommRing K] (c : Γ(Sp K, ⊤)ˣ) : ∃ z : Kˣ, unitOn z = c

theorem **comp_pow_eq_one** {S : Scheme.{0}} {T X A : Over S} [GrpObj X] [GrpObj A] (f : X ⟶ A) [IsMonHom f] {x : T ⟶ X} {k : ℕ} (hx : x ^ k = 1) : (x ≫ f) ^ k = 1

theorem **pull_pow_eq_one** {S : Scheme.{0}} {T T' X : Over S} [GrpObj X] (t : T' ⟶ T) {x : T ⟶ X} {k : ℕ} (hx : x ^ k = 1) : (t ≫ x) ^ k = 1

theorem **weilIs_iff_of_lamEq** : Out.Laws → ∀ (R : Type) [CommRing R] [IsDedekindDomain R] [CharZero R] (X Y A B : Out.PolAV (Sp R)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → LamEq A.L f₁ f₂ B.L g₁ g₂ → ∀ (T : Over (Sp R)) (k : ℕ) (x : T ⟶ X.A) (y : T ⟶ Y.A) (c : Γ(T.left, ⊤)ˣ), 0 < k → x ^ k = 1 → y ^ k = 1 → (Out.WeilIs A.A A.L k (x ≫ f₁) (y ≫ f₂) c ↔ Out.WeilIs B.A B.L k (x ≫ g₁) (y ≫ g₂) c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CommutatorPairingsEqOfCubeLBPairingIsoStatement

theorem **_root_.GlobalTheory.T.commutatorPairings_eq_of_cubeLBPairingIso** : Out.Laws → ∀ (R : Type) [CommRing R] [IsDedekindDomain R] [CharZero R] (X Y A B : Out.PolAV (Sp R)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → LamEq A.L f₁ f₂ B.L g₁ g₂ → ∀ (T : Over (Sp R)) (k : ℕ) (x : T ⟶ X.A) (y : T ⟶ Y.A) (c : Γ(T.left, ⊤)ˣ), 0 < k → x ^ k = 1 → y ^ k = 1 → (Out.WeilIs A.A A.L k (x ≫ f₁) (y ≫ f₂) c ↔ Out.WeilIs B.A B.L k (x ≫ g₁) (y ≫ g₂) c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CongruenceActionMemFrameOrbit

theorem **scal_mul** (x : Sp S ⟶ Xo.X) (a b : S) (i : (PPx x).ι) : scal x (a * b) i = scal x a i * scal x b i

theorem **scal_one** (x : Sp S ⟶ Xo.X) (i : (PPx x).ι) : scal x 1 i = 1

theorem **exists_unit_of_generators** {x : Sp S ⟶ Xo.X} (s s' : LB.Sec (PPx x)) (hs : ∀ s'' : LB.Sec (PPx x), ∃! a : S, ∀ i, s''.f i = scal x a i * s.f i) (hs' : ∀ s'' : LB.Sec (PPx x), ∃! a : S, ∀ i, s''.f i = scal x a i * s'.f i) : ∃ u : Sˣ, ∀ i, s'.f i = scal x (u : S) i * s.f i

theorem **frBor_of_eq** [Algebra c.F S] (y y' : FrPt Xo c S) (hx : y'.x = y.x) (hu : y'.u = y.u) : ∃ t : Ix Dw → Sˣ, FrBor t y y'

theorem **perFr_lact** {r : ℕ} (X : Data E r) (x : Frame X.Ψ r) (γ : GL (Fin r) E) (hγ : ∀ u v, herm X.Gm ((γ : Matrix (Fin r) (Fin r) E) *ᵥ u) ((γ : Matrix (Fin r) (Fin r) E) *ᵥ v) = herm X.Gm u v) (v : Fin r → E) (τ : E →+* ℂ) (i : ℕ) : Formalism.L2.PjFree.perFr X (lact γ x) ((γ : Matrix (Fin r) (Fin r) E) *ᵥ v) τ i = Formalism.L2.PjFree.perFr X x v τ i

theorem **congr_lact_mem_orbit** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.IsUnf U → l.A = 1 → ∀ k (γ : GL (Fin r) E), IsUStab (U.X k).Gm (U.X k).Λ γ → (∀ x ∈ (U.X k).Λ, ∃ y ∈ (U.X k).Λ, Matrix.mulVec (γ : Matrix (Fin r) (Fin r) E) x - x = (l.M : E) • y) → ∀ x ∈ (U.X k).dom, ∃ γ' ∈ (U.X k).Γ, ∃ b ∈ Bor (U.X k).Ψ r, lact γ x = rmul (lact γ' x) b

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CountablePointsOfOnto

theorem **eq_top_of_countable** {V L : Type} [AddCommGroup V] [Module ℂ V] [Countable L] (W : Submodule ℂ V) (P : L → V) (h : ∀ z, ∃ a, z - P a ∈ W) : W = ⊤

theorem **exists_basis** {V L J : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [AddCommGroup L] (b : Module.Basis J ℤ L) (P : L →+ V) (hi : Function.Injective P) (hd : DiscreteTopology P.range) (hs : Submodule.span ℝ (Set.range P) = ⊤) : ∃ B : Module.Basis J ℝ V, ∀ i, B i = P (b i)

theorem **exists_real_equiv** {V L : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [AddCommGroup L] (P P' : L →+ V) (hi : Function.Injective P) (hd : DiscreteTopology P.range) (hs : Submodule.span ℝ (Set.range P) = ⊤) (hi' : Function.Injective P') (hd' : DiscreteTopology P'.range) (hs' : Submodule.span ℝ (Set.range P') = ⊤) : ∃ h : V ≃ₗ[ℝ] V, ∀ a, P' a = h (P a)

theorem **exists_complex_equiv** {U V I : Type} [AddCommGroup U] [Module ℂ U] [AddCommGroup V] [Module ℂ V] [Module ℝ U] [Module ℝ V] [IsScalarTower ℝ ℂ U] [IsScalarTower ℝ ℂ V] (h : V ≃ₗ[ℝ] V) (F F' : I → (U →ₗ[ℂ] V)) (s : Set U) (hs : Submodule.span ℝ s = ⊤) (hF : ⨆ j, LinearMap.range (F j) = ⊤) (e : ∀ j, ∀ x ∈ s, F' j x = h (F j x)) : ∃ h' : V ≃ₗ[ℂ] V, ∀ v, h' v = h v

theorem **exists_compare** {S S' : Scheme.{0}} {G : Over S} {A B : Over S'} [GrpObj G] [GrpObj A] [GrpObj B] {f : S' ⟶ S} {α : A.left ⟶ G.left} {β : B.left ⟶ G.left} (hA : IsGrpPullback A G f α) (hB : IsGrpPullback B G f β) : ∃ κ : A ⟶ B, IsMonHom κ ∧ ∀ (T : Over S') (z : T ⟶ B), ∃ y : T ⟶ A, y ≫ κ = z

theorem **exists_linear_onto** {A B : Over (Sp ℂ)} [GrpObj A] [GrpObj B] {A₀ : Over (Sp ℂ)} [GrpObj A₀] (h₀ : IsAbelianScheme A₀) (hB : IsAbelianScheme B) {L₀ : LB A₀.left} {L : LB B.left} {g₀ g : ℕ} (u₀ : HolUnif A₀ L₀ g₀) (u : HolUnif B L g) (d₁ : ∀ v, IsTangent v → ∃! w, u.IsDiff w v) (d₂ : ∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) (κ : A ⟶ B) [IsMonHom κ] (hκ : ∀ (T : Over (Sp ℂ)) (z : T ⟶ B), ∃ y : T ⟶ A, y ≫ κ = z) {I : Type} (ψ : I → (A₀ ⟶ A)) (hψ : ∀ i, IsMonHom (ψ i)) (ho : OntoAt ψ (𝟙 (Sp ℂ))) : ∃ F : I → ((Fin g₀ → ℂ) →ₗ[ℂ] (Fin g → ℂ)), (∀ j, ∀ x ∈ u₀.Λ, F j x ∈ u.Λ) ∧ ∀ z, ∃ w ∈ ⨆ j, LinearMap.range (F j), z - w ∈ u.Λ

theorem **countable_points_of_onto** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (P₀ : Out.PolAV (Sp ℂ)), Set.Countable {y : Sp ℂ ⟶ Xo.X | y ≫ Xo.s = spMap (algebraMap (KA l) ℂ) ∧ ∃ (t : SAB H (Sp ℂ)) (_ : SABMap t Xo.univ.t y) (n : ℕ) (ψ : Fin n → (P₀.A ⟶ t.G)), (∀ i, IsMonHom (ψ i)) ∧ OntoAt ψ (𝟙 (Sp ℂ))}

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CubeLBPairingIsoIffWeilPairingCompatibleOverRing

theorem **exists_hom** {G H : Over S} {G' H' : Over S'} [GrpObj G] [GrpObj H] [GrpObj G'] [GrpObj H'] {φ : G'.left ⟶ G.left} {ψ : H'.left ⟶ H.left} (hG : IsGrpPullback G' G s φ) (hH : IsGrpPullback H' H s ψ) (w : G ⟶ H) [IsMonHom w] : ∃ w' : G' ⟶ H', IsMonHom w' ∧ w'.left ≫ ψ = φ ≫ w.left

theorem **cube_pull** {G : Over S} {G' : Over S'} [GrpObj G] [IsCommMonObj G] [GrpObj G'] [IsCommMonObj G'] {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G s φ) {L : LB G.left} {L' : LB G'.left} (hL : L.IsLB) (hL' : L'.IsLB) (w : LBIso L' (L.pull φ)) {T : Over S} {T' : Over S'} (U : T'.left ⟶ T.left) (hU : U ≫ T.hom = T'.hom ≫ s) {n : ℕ} (x' : Fin n → (T' ⟶ G')) (x : Fin n → (T ⟶ G)) (hx : ∀ i, (x' i).left ≫ φ = U ≫ (x i).left) : LBIso (cubeLB G' x' L') ((cubeLB G x L).pull U)

theorem **exists_gen_cube_one** {G : Over S} [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (T : Over S) : ∃ t : LB.Gen (cubeLB G (fun _ : Fin 2 => (1 : T ⟶ G)) L), t.IsGen

theorem **cube_zero** {X Y G : Over S} [GrpObj X] [GrpObj Y] [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (f₁ : X ⟶ G) (f₂ : Y ⟶ G) [IsMonHom f₁] [IsMonHom f₂] : LBIso (cubeLB G (fun _ : Fin 2 => (1 : 𝟙_ (Over S) ⟶ G)) L) ((cubeLB G ![fst X Y ≫ f₁, snd X Y ≫ f₂] L).pull (eSec (X ⊗ Y)))

theorem **weilEq_iff** {X Y A B : Over S} {X' Y' A' B' : Over S'} [GrpObj X] [GrpObj Y] [GrpObj A] [GrpObj B] [IsCommMonObj A] [IsCommMonObj B] [GrpObj X'] [GrpObj Y'] [GrpObj A'] [GrpObj B'] {x : X'.left ⟶ X.left} {y : Y'.left ⟶ Y.left} {a : A'.left ⟶ A.left} {b : B'.left ⟶ B.left} (hX : IsGrpPullback X' X s x) (hY : IsGrpPullback Y' Y s y) (hA : IsGrpPullback A' A s a) (hB : IsGrpPullback B' B s b) {L : LB A.left} {M : LB B.left} {L' : LB A'.left} {M' : LB B'.left} (hL : L.IsLB) (hM : M.IsLB) (hL' : L'.IsLB) (hM' : M'.IsLB)

(wL : LBIso L' (L.pull a)) (wM : LBIso M' (M.pull b)) {f₁ : X ⟶ A} {f₂ : Y ⟶ A} {g₁ : X ⟶ B} {g₂ : Y ⟶ B} {f₁' : X' ⟶ A'} {f₂' : Y' ⟶ A'} {g₁' : X' ⟶ B'} {g₂' : Y' ⟶ B'} (e₁ : f₁'.left ≫ a = x ≫ f₁.left) (e₂ : f₂'.left ≫ a = y ≫ f₂.left) (e₃ : g₁'.left ≫ b = x ≫ g₁.left) (e₄ : g₂'.left ≫ b = y ≫ g₂.left) {k : Type} [Field k] (t : Sp k ⟶ S') : WeilEq L' f₁' f₂' M' g₁' g₂

theorem **lamEq_weilEq_over_ring** : Out.Laws → ∀ (R : Type) [CommRing R] (X Y A B : Out.PolAV (Sp R)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → ∀ (k : Type) [Field k] [IsAlgClosed k] [CharZero k] (β : R →+* k), (LamEq A.L f₁ f₂ B.L g₁ g₂ → WeilEq A.L f₁ f₂ B.L g₁ g₂ (spMap β)) ∧ (IsDedekindDomain R → Function.Injective β → WeilEq A.L f₁ f₂ B.L g₁ g₂ (spMap β) → LamEq A.L f₁ f₂ B.L g₁ g₂)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CubeLBPairingIsoIffWeilPairingCompatibleOverRingStatement

theorem **_root_.GlobalTheory.T.cubeLBPairingIso_iff_weilPairingCompatible_over_ring** : Out.Laws → ∀ (R : Type) [CommRing R] (X Y A B : Out.PolAV (Sp R)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → ∀ (k : Type) [Field k] [IsAlgClosed k] [CharZero k] (β : R →+* k), (LamEq A.L f₁ f₂ B.L g₁ g₂ → WeilEq A.L f₁ f₂ B.L g₁ g₂ (spMap β)) ∧ (IsDedekindDomain R → Function.Injective β → WeilEq A.L f₁ f₂ B.L g₁ g₂ (spMap β) → LamEq A.L f₁ f₂ B.L g₁ g₂)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CurveHomExtOfDerivativeLieBasis

theorem **hom_ext_of_lieBasis** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (k : Type) [Field k] (s : k →+* ℂ) (P Q : Out.PolAV (Sp k)) (v : epsR k ⟶ P.A) (f f' : P.A ⟶ Q.A), (∀ (R' : Type) [CommRing R'] (f' : k →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) → IsMonHom f → IsMonHom f' → v ≫ f = v ≫ f' → f = f'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CurveHomExtOfDerivativeLieBasisStatement

theorem **_root_.GlobalTheory.T.curve_hom_ext_of_derivative_lieBasis** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (k : Type) [Field k] (s : k →+* ℂ) (P Q : Out.PolAV (Sp k)) (v : epsR k ⟶ P.A) (f f' : P.A ⟶ Q.A), (∀ (R' : Type) [CommRing R'] (f' : k →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) → IsMonHom f → IsMonHom f' → v ≫ f = v ≫ f' → f = f'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CuspFormBoundedBySupNormAtGoodEmbeddingAllEmbedd

theorem **cusp_bound_allEmb** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, ∀ (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) (N : ℝ), IsMetric y N (fun _ => 1) → cabs (evR Xo c F y) ≤ Real.sqrt N ^ n * supC Xo c U ξ (goodOf c σ) a n F

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CuspFormBoundedBySupNormAtGoodEmbeddingAllEmbeddStatement

theorem **_root_.GlobalTheory.T.cuspForm_bounded_by_supNorm_at_good_embedding_all_embeddings** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, ∀ (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) (N : ℝ), IsMetric y N (fun _ => 1) → cabs (evR Xo c F y) ≤ Real.sqrt N ^ n * supC Xo c U ξ (goodOf c σ) a n F

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CuspFormValueLeSupNormAtOrthonormalFrame

theorem **norm_diag_of_triangular** {m : ℕ} (Y : Matrix (Fin m) (Fin m) ℂ) (κ : ℝ) (hκ : 0 < κ) (hY : ∀ i j, j < i → Y i j = 0) (h : Y.conjTranspose * ((κ : ℂ) • (1 : Matrix (Fin m) (Fin m) ℂ)) * Y = 1) (i : Fin m) : ‖Y i i‖ = (Real.sqrt κ)⁻¹

theorem **cabs_evR_le_supC** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, ∀ σ, GoodEmb c σ → ∀ (y : FrPt Xo c (CAt σ)) (N : ℝ), IsMetric y N (fun _ => 1) → cabs (evR Xo c F y) ≤ Real.sqrt N ^ n * supC Xo c U ξ σ a n F

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.CuspFormValueLeSupNormAtOrthonormalFrameStatement

theorem **_root_.GlobalTheory.T.cuspForm_value_le_supNorm_at_orthonormal_frame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, ∀ σ, GoodEmb c σ → ∀ (y : FrPt Xo c (CAt σ)) (N : ℝ), IsMetric y N (fun _ => 1) → cabs (evR Xo c F y) ≤ Real.sqrt N ^ n * supC Xo c U ξ σ a n F

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DataXOfModelOfGamAt

theorem **dataXOfModel_of_gamAt** : ∀ (D : D7) (T : Core D), GamAt T → ∀ j, ∀ (_h : LawsCore T), InR D j → ∀ (σ₀ : T.Q.c.F →+* ℂ), GoodEmb T.Q.c σ₀ → ∀ (k : (T.U j σ₀).Comp), ∃ Xx : DataX D.host.E D.R.r, Xx.toData = (T.U j σ₀).X k ∧ Xx.g = T.Q.l.M ∧ Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u}

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegEigLatEqOfIso

theorem **diff_mem_eigLat** {K : Type} [Field K] [NumberField K] {P P' : Out.PolAV (Sp (𝒪 K))} {F : GenFib P} {F' : GenFib P'} {O : Type} [CommRing O] {a : OAct O P.A} {a' : OAct O P'.A} {f : P.A ⟶ P'.A} (hf : ∀ x, a.act x ≫ f = f ≫ a'.act x) {M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)} (hM : F.IsDiffOf F' f M) (τ : O →+* 𝒪 K) {c : Fin F.P₀.g → K} (hc : c ∈ F.eigLat a τ) : M c ∈ F'.eigLat a' τ

theorem **map_eigLat_of_iso** {K : Type} [Field K] [NumberField K] {P P' : Out.PolAV (Sp (𝒪 K))} {F : GenFib P} {F' : GenFib P'} {O : Type} [CommRing O] {a : OAct O P.A} {a' : OAct O P'.A} (e : P.A ≅ P'.A) (hf : ∀ x, a.act x ≫ e.hom = e.hom ≫ a'.act x) {M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)} {M' : (Fin F'.P₀.g → K) →ₗ[K] (Fin F.P₀.g → K)} (hM : F.IsDiffOf F' e.hom M) (hM' : F'.IsDiffOf F e.inv M') (τ : O →+* 𝒪 K) : (F.eigLat a τ).map (M.restrictScalars (𝒪 K)) = F'.eigLat a' τ

theorem **deg_eigLat_eq_of_iso** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P P' : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (F' : GenFib P') (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)) (O : Type) [CommRing O] (a : OAct O P.A) (a' : OAct O P'.A) (e : P.A ≅ P'.A), F.IsHerm h → F'.IsHerm h' → Serre.IsLinearHom a a' e.hom → LamEq P'.L e.hom e.hom P.L (𝟙 P.A) (𝟙 P.A) → ∀ τ : O →+* 𝒪 K, deg h (F.eigLat a τ) = deg h' (F'.eigLat a' τ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegEqNdegOfLine

theorem **deg_eq_ndeg_of_line** : ∀ (K : Type) [Field K] [NumberField K] (V : Type) [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] (h : HermFam K V) (Λ : Submodule (𝒪 K) V) (v : V) (ℓ : MLine K), v ≠ 0 → ℓ.Good → (∀ z, z ∈ Λ ↔ ∃ c ∈ ℓ.I, z = c • v) → (∀ s, ℓ.n s ^ 2 = (h.form s v v).re) → deg h Λ = ℓ.ndeg

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegLatticeNumberFieldExtensionEq

theorem **form_eq_sum** {L : Type} [Field L] {n : ℕ} (h : HermFam L (Fin n → L)) (s : L →+* ℂ) (x y : Fin n → L) : h.form s x y = ∑ a, ∑ b, starRingEnd ℂ (s (x a)) * s (y b) * h.form s (Pi.single a 1) (Pi.single b 1)

private lemma **eq_sum_apply_smul_single** {F : Type} [Field F] (z : Fin n → F) : z = ∑ a, z a • (Pi.single a 1 : Fin n → F)

private lemma **hermFam_form_map_eq_of_eq_on_basis** (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K (Fin n → K)) (h' : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K' (Fin n → K')) (hf : ∀ s a b, h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)) (s : K' →+* ℂ) (x y : Fin n → K) : h'.form s (fun i => j (x i)) (fun i => j (y i)) = h.form (s.comp j) x y

private lemma **hermFam_form_zero_right** {F W : Type} [Field F] [AddCommGroup W] [Module F W] (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam F W) (σ : F →+* ℂ) (x : W) : h.form σ x 0 = 0

private lemma **linearIndependent_map_of_linearIndependent** (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K (Fin n → K)) (h' : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K' (Fin n → K')) (hf : ∀ s a b, h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)) {r : ℕ} {v : Fin r → Fin n → K} (hv : LinearIndependent K v) : LinearIndependent K' (fun a i => j (v a i))

private lemma **map_mem_span_range_map_of_mem_span** {r : ℕ} {v : Fin r → Fin n → K} {x : Fin n → K} (hx : x ∈ Submodule.span K (Set.range v)) : (fun i => j (x i)) ∈ Submodule.span K' (Set.range fun a i => j (v a i))

private lemma **isFrame_span_image_map** (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K (Fin n → K)) (h' : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K' (Fin n → K')) (hf : ∀ s a b, h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)) {B : Submodule (𝒪 K) (Fin n → K)} {r : ℕ} {v : Fin r → Fin n → K} (hB : _root_.ReciprocityTheory.GeometryOfNumbersTheory.IsFrame K B v) : _root_.ReciprocityTheory.GeometryOfNumbersTheory.IsFrame K' (Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (B : Set (Fin n → K)))) (fun a i => j (v a i))

private lemma **span_image_map_span_eq_span_image** (S : Set (Fin n → K)) : Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (Submodule.span (𝒪 K) S : Set (Fin n → K))) = Submodule.span (𝒪 K') ((fun c i => j (c i)) '' S)

private lemma **fg_span_image_map_of_fg** {B : Submodule (𝒪 K) (Fin n → K)} (hB : B.FG) : (Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (B : Set (Fin n → K)))).FG

private lemma **le_span_range_of_isFrame_fin_zero** {B : Submodule (𝒪 L) V} {v : Fin 0 → V} (hB : _root_.ReciprocityTheory.GeometryOfNumbersTheory.IsFrame L B v) : B ≤ Submodule.span (𝒪 L) (Set.range v)

private lemma **exists_linearForm_vanish_init_eq_one_last** {r : ℕ} {v : Fin (r + 1) → V} (hv : LinearIndependent L v) : ∃ f : V →ₗ[L] L, (∀ i : Fin r, f (v i.castSucc) = 0) ∧ f (v (Fin.last r)) = 1

private lemma **map_span_range_eq_one_of_form_last** {r : ℕ} {v : Fin (r + 1) → V} {f : V →ₗ[L] L} (h0 : ∀ i : Fin r, f (v i.castSucc) = 0) (h1 : f (v (Fin.last r)) = 1) : (Submodule.span (𝒪 L) (Set.range v)).map (f.restrictScalars (𝒪 L)) = 1

private lemma **span_range_inf_ker_eq_span_range_castSucc** {R : Type} [CommRing R] [Algebra R L] [Module R V] [IsScalarTower R L V] [FaithfulSMul R L] {r : ℕ} {v : Fin (r + 1) → V} {f : V →ₗ[L] L} (h0 : ∀ i : Fin r, f (v i.castSucc) = 0) (h1 : f (v (Fin.last r)) = 1) : Submodule.span R (Set.range v) ⊓ (LinearMap.ker f).restrictScalars R = Submodule.span R (Set.range fun i : Fin r => v i.castSucc)

private lemma **relIndex_eq_relIndex_map_mul_relIndex_inf_ker** {A B : Submodule (𝒪 L) V} (hAB : A ≤ B) (f : V →ₗ[L] L) : A.toAddSubgroup.relIndex B.toAddSubgroup = (A.map (f.restrictScalars (𝒪 L))).toAddSubgroup.relIndex (B.map (f.restrictScalars (𝒪 L))).toAddSubgroup * (A ⊓ (LinearMap.ker f).restrictScalars (𝒪 L)).toAddSubgroup.relIndex (B ⊓ (LinearMap.ker f).restrictScalars (𝒪 L)).toAddSubgroup

private lemma **log_relIndex_one_eq_neg_log_absNorm_of_one_le** (𝔞 : FractionalIdeal (𝒪 L)⁰ L) (h1 : 1 ≤ (𝔞 : Submodule (𝒪 L) L)) : Real.log ((1 : Submodule (𝒪 L) L).toAddSubgroup.relIndex (𝔞 : Submodule (𝒪 L) L).toAddSubgroup) = -Real.log (FractionalIdeal.absNorm 𝔞 : ℝ)

private lemma **log_relIndex_map_lemma_4** (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K (Fin n → K)) (h' : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K' (Fin n → K')) (hf : ∀ s a b, h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)) (r : ℕ) : ∀ {B : Submodule (𝒪 K) (Fin n → K)} {v : Fin r → Fin n → K}, B.FG → _root_.ReciprocityTheory.GeometryOfNumbersTheory.IsFrame K B v → Real.log ((Submodule.span (𝒪 K') (Set.range fun a i => j (v a i))).toAddSubgroup.relIndex (Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (B : Set (Fin n → K)))).toAddSubgroup) / finrank ℚ K' = Real.log ((Submodule.span (𝒪 K) (Set.range v)).toAddSubgroup.relIndex B.toAddSubgroup) / finrank ℚ K

theorem **_root_.GlobalTheory.deg_span_image_eq** : ∀ (K K' : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] (j : K →+* K') (n : ℕ) (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K (Fin n → K)) (h' : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K' (Fin n → K')) (Λ : Submodule (𝒪 K) (Fin n → K)), Λ.FG → (∀ s a b, h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)) → _root_.ReciprocityTheory.GeometryOfNumbersTheory.deg h' (Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (Λ : Set (Fin n → K)))) = _root_.ReciprocityTheory.GeometryOfNumbersTheory.deg h Λ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegLatticeNumberFieldExtensionEqStatement

theorem **_root_.GlobalTheory.T.deg_lattice_numberField_extension_eq** : ∀ (K K' : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] (j : K →+* K') (n : ℕ) (h : HermFam K (Fin n → K)) (h' : HermFam K' (Fin n → K')) (Λ : Submodule (𝒪 K) (Fin n → K)), Λ.FG → (∀ s a b, h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)) → deg h' (Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (Λ : Set (Fin n → K)))) = deg h Λ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegOfLatRankEqZero

theorem **eigLat_le** {P : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) {O : Type} [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K) : F.eigLat act τ ≤ F.lieLat

theorem **eigLat_fg** {P : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) {O : Type} [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K) : (F.eigLat act τ).FG

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegreeLieLatticeHermitianInvariant

theorem **injective_of_isometry** (h : HermFam K V) (h' : HermFam K V') (M : V →ₗ[K] V') (hM : ∀ s c c', h'.form s (M c) (M c') = h.form s c c') : Function.Injective M

theorem **deg_map_of_isometry** (h : HermFam K V) (h' : HermFam K V') (M : V →ₗ[K] V') (hM : ∀ s c c', h'.form s (M c) (M c') = h.form s c c') {E : Submodule (𝒪 K) V} (hE : E.FG) : deg h E = deg h' (E.map (M.restrictScalars (𝒪 K)))

theorem **existsUnique_coord** (F : GenFib P) {u : epsR (𝒪 K) ⟶ P.A} (hu : IsTangent u) : ∃! c, F.IsCoord u c

theorem **image_coord** (F F' : GenFib P) {M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)} (hM : F.IsDiffOf F' (𝟙 P.A) M) (p : (epsR (𝒪 K) ⟶ P.A) → Prop) : Submodule.span (𝒪 K) {c | ∃ u, F'.IsCoord u c ∧ p u} = (Submodule.span (𝒪 K) {c | ∃ u, F.IsCoord u c ∧ p u}).map (M.restrictScalars (𝒪 K))

theorem **deg_lieLat_eq** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (F' : GenFib P) (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)), F.IsHerm h → F'.IsHerm h' → deg h F.lieLat = deg h' F'.lieLat ∧ ∀ (O : Type) [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K), deg h (F.eigLat act τ) = deg h' (F'.eigLat act τ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegreeLieLatticeHermitianInvariantStatement

theorem **_root_.GlobalTheory.T.degree_lie_lattice_hermitian_invariant** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (F' : GenFib P) (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)), F.IsHerm h → F'.IsHerm h' → deg h F.lieLat = deg h' F'.lieLat ∧ ∀ (O : Type) [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K), deg h (F.eigLat act τ) = deg h' (F'.eigLat act τ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegreeLieLatticesCmTripleBaseChange

theorem **vec_comp** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] {f : R →+* R'} {g : R' →+* R''} {A : Over (Sp R)} {A' : Over (Sp R')} {A'' : Over (Sp R'')} {φ : A'.left ⟶ A.left} {ψ : A''.left ⟶ A'.left} {v : epsR R ⟶ A} {v' : epsR R' ⟶ A'} {v'' : epsR R'' ⟶ A''} (h : IsVecOver f φ v v') (h' : IsVecOver g ψ v' v'') : IsVecOver (g.comp f) (ψ ≫ φ) v v''

theorem **algebraMap_comp_mapRingHom** (j : K →+* K') : (algebraMap (𝒪 K') K').comp (RingOfIntegers.mapRingHom j) = j.comp (algebraMap (𝒪 K) K)

theorem **isCoord_over** (hξ : IsGrpPullback P'.A P.A (spMap (RingOfIntegers.mapRingHom j)) ξ) (hψ : IsGrpPullback Q.A F.P₀.A (spMap j) ψ) (w : ψ ≫ F.φ = φ' ≫ ξ) (hv : ∀ i, IsVecOver j ψ (F.v i) (v' i)) {u : epsR (𝒪 K) ⟶ P.A} {c : Fin F.P₀.g → K} (hc : F.IsCoord u c) {u' : epsR (𝒪 K') ⟶ P'.A} (hu : IsVecOver (RingOfIntegers.mapRingHom j) ξ u u') : (F₁).IsCoord u' (fun i => j (c i))

theorem **image_subset_lieLat** (hξ : IsGrpPullback P'.A P.A (spMap (RingOfIntegers.mapRingHom j)) ξ) (hψ : IsGrpPullback Q.A F.P₀.A (spMap j) ψ) (w : ψ ≫ F.φ = φ' ≫ ξ) (hv : ∀ i, IsVecOver j ψ (F.v i) (v' i)) {c : Fin F.P₀.g → K} (hc : c ∈ F.lieLat) : (fun i => j (c i)) ∈ (F₁).lieLat

theorem **exists_pow_smul_mem** {O R k : Type} [CommRing O] [CommRing R] [Field k] [Algebra O R] [Algebra R k] [Algebra O k] [IsScalarTower O R k] (r : O) [IsLocalization.Away r R] {n : ℕ} (N : Submodule O (Fin n → k)) (M : Fin n → Fin n → k) (hM : ∀ i, ∃ l : ℕ, r ^ l • M i ∈ N) (a : Fin n → R) : ∃ l : ℕ, r ^ l • (∑ i, algebraMap R k (a i) • M i) ∈ N

theorem **lieLat_eq_span** (hB : Out.PolBaseChangeSt) (hξ : IsPolFibre (RingOfIntegers.mapRingHom j) P P' ξ) (hψ : IsGrpPullback Q.A F.P₀.A (spMap j) ψ) (w : ψ ≫ F.φ = φ' ≫ ξ) (hv : ∀ i, IsVecOver j ψ (F.v i) (v' i)) : (F₁).lieLat = Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (F.lieLat : Set (Fin F.P₀.g → K)))

theorem **eigLat_eq_span** (hB : Out.PolBaseChangeSt) (hξ : IsPolFibre (RingOfIntegers.mapRingHom j) P P' ξ) (hψ : IsGrpPullback Q.A F.P₀.A (spMap j) ψ) (w : ψ ≫ F.φ = φ' ≫ ξ) (hv : ∀ i, IsVecOver j ψ (F.v i) (v' i)) {O : Type} [CommRing O] (act : OAct O P.A) (act' : OAct O P'.A) (hact : ∀ a, (act'.act a).left ≫ ξ = ξ ≫ (act.act a).left) (τ : O →+* 𝒪 K) : (F₁).eigLat act' ((RingOfIntegers.mapRingHom j).comp τ) = Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (F.eigLat act τ : Set (Fin F.P₀.g → K)))

theorem **form_single_eq** (hU : ThetaUnifAV) (hψ : IsPolFibre j F.P₀ (withDim Q F.P₀.g hd) ψ) (hv : ∀ i, IsVecOver j ψ (F.v i) (v' i)) {h : HermFam K (Fin F.P₀.g → K)} {h' : HermFam K' (Fin F.P₀.g → K')} (H : F.IsHerm h) (H' : (F₁).IsHerm h') (s : K' →+* ℂ) (a b : Fin F.P₀.g) : h'.form s (Pi.single a 1) (Pi.single b 1) = h.form (s.comp j) (Pi.single a 1) (Pi.single b 1)

theorem **deg_eq_of_isPolFibre** (hU : ThetaUnifAV) (hL : Out.Laws) (hξ : IsPolFibre (RingOfIntegers.mapRingHom j) P P' ξ) (F : GenFib P) (F' : GenFib P') (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K' (Fin F'.P₀.g → K')) (H : F.IsHerm h) (H' : F'.IsHerm h') : deg h' F'.lieLat = deg h F.lieLat ∧ ∀ (O : Type) [CommRing O] (act : OAct O P.A) (act' : OAct O P'.A), (∀ a, (act'.act a).left ≫ ξ = ξ ≫ (act.act a).left) → ∀ τ : O →+* 𝒪 K, deg h' (F'.eigLat act' ((RingOfIntegers.mapRingHom j).comp τ)) = deg h (F.eigLat act τ)

theorem **deg_lieLat_eigLat_baseChange** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K K' : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] (j : K →+* K') (T : PAT H (Sp (𝒪 K))) (T' : PAT H (Sp (𝒪 K'))) (_ : SABMap T'.t T.t (spMap (RingOfIntegers.mapRingHom j))) (F : GenFib T.pol) (F' : GenFib T'.pol) (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K' (Fin F'.P₀.g → K')), F.IsHerm h → F'.IsHerm h' → deg h' F'.lieLat = deg h F.lieLat ∧ ∀ τ : 𝒪 E →+* 𝒪 K, deg h' (F'.eigLat T'.t.act ((RingOfIntegers.mapRingHom j).comp τ)) = deg h (F.eigLat T.t.act τ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DegreeLieLatticesCmTripleBaseChangeStatement

theorem **_root_.GlobalTheory.T.degree_lieLattices_cmTriple_baseChange** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K K' : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] (j : K →+* K') (T : PAT H (Sp (𝒪 K))) (T' : PAT H (Sp (𝒪 K'))) (_ : SABMap T'.t T.t (spMap (RingOfIntegers.mapRingHom j))) (F : GenFib T.pol) (F' : GenFib T'.pol) (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K' (Fin F'.P₀.g → K')), F.IsHerm h → F'.IsHerm h' → deg h' F'.lieLat = deg h F.lieLat ∧ ∀ τ : 𝒪 E →+* 𝒪 K, deg h' (F'.eigLat T'.t.act ((RingOfIntegers.mapRingHom j).comp τ)) = deg h (F.eigLat T.t.act τ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DenseClosedPoints

theorem **dense_closedPoints** {X : Scheme.{0}} (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] : Dense (closedPoints X)

theorem **hom_ext_of_points** {X Y : Scheme.{0}} (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] [Y.IsSeparated] (f g : X ⟶ Y) (H : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → x ≫ f = x ≫ g) : f = g

theorem **eq_zero_of_points** {X : Scheme.{0}} (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (W : X.Opens) (h : Γ(X, W)) (H : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ∀ e : ⊤ ≤ x ⁻¹ᵁ W, valAt x W e h = 0) : h = 0

theorem **lift_toSpecΓ** {X : Scheme.{0}} {R : CommRingCat.{0}} (W : X.Opens) (x : Spec R ⟶ X) (x' : Spec R ⟶ W) (hx : x' ≫ W.ι = x) (e : ⊤ ≤ x ⁻¹ᵁ W) : x' ≫ W.toSpecΓ = Spec.map (x.appLE W ⊤ e ≫ (Scheme.ΓSpecIso R).hom)

theorem **range_subset** {X T : Scheme.{0}} (W : X.Opens) (x : T ⟶ X) (e : ⊤ ≤ x ⁻¹ᵁ W) : Set.range x ⊆ Set.range W.ι

theorem **top_le_of_lift** {X T : Scheme.{0}} (W : X.Opens) (x : T ⟶ X) (x' : T ⟶ W) (hx : x' ≫ W.ι = x) : ⊤ ≤ x ⁻¹ᵁ W

theorem **valAt'_eq** {X : Scheme.{0}} (x : Sp ℂ ⟶ X) (W : X.Opens) (e : ⊤ ≤ x ⁻¹ᵁ W) (h : Γ(X, W)) : valAt' x W h = (ev x W e).hom h

theorem **exists_local** {X Y : Scheme.{0}} (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] {Z : Type} (ex : Z → (Sp ℂ ⟶ X)) (ey : Z → (Sp ℂ ⟶ Y)) (hex : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ∃ z, ex z = x) (W : X.Opens) (D : Y.Opens) (hD : IsAffineOpen D) (hWD : ∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → ⊤ ≤ ey w ⁻¹ᵁ D) (hf : ∀ f : Γ(Y, D), ∃ h : Γ(X, W), ∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → valAt' (ex w) W h = valAt' (ey w) D f) : ∃ g : W.toScheme ⟶ Y, ∀ w (x' : Sp ℂ ⟶ W), x' ≫ W.ι = ex w → x' ≫ g = ey w

theorem **exists_hom** {X Y : Scheme.{0}} (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] {Z : Type} (ex : Z → (Sp ℂ ⟶ X)) (ey : Z → (Sp ℂ ⟶ Y)) (hex : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ∃ z, ex z = x) {S : Scheme.{0}} (q : Y ⟶ S) [IsSeparated q] [IsAffine S] (H : ∀ z, ∃ (W : X.Opens) (D : Y.Opens), IsAffineOpen D ∧ ⊤ ≤ ex z ⁻¹ᵁ W ∧ (∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → ⊤ ≤ ey w ⁻¹ᵁ D) ∧ ∀ f : Γ(Y, D), ∃ h : Γ(X, W), ∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → valAt' (ex w) W h = valAt' (ey w) D f) : ∃ g : X ⟶ Y, ∀ z, ex z ≫ g = ey z

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DetCondTensorDefinite

theorem **rightAction_apply** (a : O') : rightAction k O O' M a = LinearMap.baseChange k (Algebra.lsmul O O M a : M →ₗ[O] M)

theorem **rightAction_tmul** (a : O') (c : k) (m : M) : rightAction k O O' M a (c ⊗ₜ m) = c ⊗ₜ (a • m)

theorem **nonempty_eigenSub_equiv** [Algebra O' k] [IsScalarTower O O' k] {I : Type} {σ : I → O' →+* k} (h : EigenSplit.IsSplitting σ) (i : I) (hi : σ i = algebraMap O' k) : Nonempty (EigenSplit.eigenSub (rightAction k O O' M) (σ i) ≃ₗ[k] k ⊗[O'] M)

theorem **injective_of_charZero** {E k : Type} [Field E] [NumberField E] [Field k] [CharZero k] (τ : 𝒪 E →+* k) : Function.Injective τ

theorem **finrank_eigenSub** {E k O M : Type} [Field E] [NumberField E] [Field k] [IsAlgClosed k] [CharZero k] [CommRing O] [Algebra O k] [Algebra O (𝒪 E)] [AddCommGroup M] [Module O M] [Module (𝒪 E) M] [IsScalarTower O (𝒪 E) M] (τ : 𝒪 E →+* k) (hτ : τ.comp (algebraMap O (𝒪 E)) = algebraMap O k) : Module.finrank k (EigenSplit.eigenSub (rightAction k O (𝒪 E) M) τ) = Module.finrank (𝒪 E) M

theorem **lieMult_of_param** {k O V : Type} [Field k] [CommRing O] [AddCommGroup V] [Module k V] [Module.Finite k V] {A : Over (Sp k)} [GrpObj A] (act : OAct O A) (ρ : O →+* Module.End k V) (Φ : V → (epsR k ⟶ A)) (inj : Function.Injective Φ) (tan : ∀ x, IsTangent x ↔ x ∈ Set.range Φ) (add : ∀ y z, Φ (y + z) = Φ y ∗ Φ z) (smul : ∀ (c : k) y, IsLieComb ![c] ![Φ y] (Φ (c • y))) (eqv : ∀ a y, Φ (ρ a y) = Φ y ≫ act.act a) (τ : O →+* k) {n : ℕ} (hn : Module.finrank k (EigenSplit.eigenSub ρ τ) = n) : LieMult act (𝟙 (Sp k)) τ n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DetDvdEquivariantFramePolynomial

theorem **exists_lower** {K : Type} [Field K] {n : ℕ} (x y : Matrix (Fin n) (Fin n) K) (l : Fin n) (hl : ∀ i, i ≤ l) (hx : x.det ≠ 0) (h : ∀ i, i ≠ l → y i = x i) : ∃ b : Matrix (Fin n) (Fin n) K, y = b ∗ x ∧ (∀ i j, i < j → b i j = 0) ∧ (∀ i, i ≠ l → b i i = 1) ∧ b.det = b l l

theorem **det_dvd_of_equivariant** : ∀ (K : Type) [Field K] [Infinite K] (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (P : MvPolynomial (FrVar m) K) (a : ∀ b, Fin (m b) → ℕ), (∀ (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) (b : B) (t : Matrix (Fin (m b)) (Fin (m b)) K), (∀ b', (X b').det ≠ 0) → t.det ≠ 0 → t.BlockTriangular id → evalFr P (Function.update X b (X b ∗ t)) = (∏ i, t i i ^ (a b i)) ∗ evalFr P X) → ∀ b : B, (∀ i, 1 ≤ a b i) → ∃ Q : MvPolynomial (FrVar m) K, P = (Matrix.of fun i k : Fin (m b) => (MvPolynomial.X (b, (k, i)) : MvPolynomial (FrVar m) K)).det ∗ Q

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.DetDvdEquivariantFramePolynomialStatement

theorem **_root_.GlobalTheory.T.det_dvd_equivariant_framePolynomial** : ∀ (K : Type) [Field K] [Infinite K] (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (P : MvPolynomial (FrVar m) K) (a : ∀ b, Fin (m b) → ℕ), (∀ (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) (b : B) (t : Matrix (Fin (m b)) (Fin (m b)) K), (∀ b', (X b').det ≠ 0) → t.det ≠ 0 → t.BlockTriangular id → evalFr P (Function.update X b (X b ∗ t)) = (∏ i, t i i ^ (a b i)) ∗ evalFr P X) → ∀ b : B, (∀ i, 1 ≤ a b i) → ∃ Q : MvPolynomial (FrVar m) K, P = (Matrix.of fun i k : Fin (m b) => (MvPolynomial.X (b, (k, i)) : MvPolynomial (FrVar m) K)).det ∗ Q

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EigenLatticeNumberFieldBaseChange

theorem **eigenBC_relation** {K K' : Type} [Field K] [NumberField K] [Field K'] [NumberField K'] [Algebra K K'] {m : ℕ} (s : Fin m → 𝒪 K') (x : Fin m → K) (φ : 𝒪 K' →ₗ[𝒪 K] 𝒪 K) (h : ∑ i, (s i : K') ∗ algebraMap K K' (x i) = 0) : ∑ i, ((φ (s i) : 𝒪 K) : K) ∗ x i = 0

theorem **eigenBC_frame** (K K' : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] [Algebra K K'] : ∃ (m : ℕ) (b : Fin m → 𝒪 K') (φ : Fin m → (𝒪 K' →ₗ[𝒪 K] 𝒪 K)), ∀ s, s = ∑ k, φ k s • b k

theorem **eigenLattice_baseChange** : ∀ (K K' O : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] [CommRing O] (j : K →+* K') (n : ℕ) (Λ : Submodule (𝒪 K) (Fin n → K)) (r : O → Matrix (Fin n) (Fin n) K) (τ : O → K) (c' : Fin n → K'), Λ.FG → (c' ∈ Submodule.span (𝒪 K') ((fun c i => j (c i)) '' {c | c ∈ Λ ∧ ∀ a, (r a).mulVec c = τ a • c}) ↔ c' ∈ Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (Λ : Set (Fin n → K))) ∧ ∀ a, ((r a).map j).mulVec c' = j (τ a) • c')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EigenLatticeNumberFieldBaseChangeStatement

theorem **_root_.GlobalTheory.T.eigenLattice_numberField_baseChange** : ∀ (K K' O : Type) [Field K] [NumberField K] [Field K'] [NumberField K'] [CommRing O] (j : K →+* K') (n : ℕ) (Λ : Submodule (𝒪 K) (Fin n → K)) (r : O → Matrix (Fin n) (Fin n) K) (τ : O → K) (c' : Fin n → K'), Λ.FG → (c' ∈ Submodule.span (𝒪 K') ((fun c i => j (c i)) '' {c | c ∈ Λ ∧ ∀ a, (r a).mulVec c = τ a • c}) ↔ c' ∈ Submodule.span (𝒪 K') ((fun c i => j (c i)) '' (Λ : Set (Fin n → K))) ∧ ∀ a, ((r a).map j).mulVec c' = j (τ a) • c')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EllMapCancelAndSabMapCancel

theorem **dualMapR_comp** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R →+* R') (f' : R' →+* R'') : Chart.dualMapR (f'.comp f) = (Chart.dualMapR f').comp (Chart.dualMapR f)

theorem **ellMap_cancel** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] {f : R →+* R'} {f' : R' →+* R''} {ω : R} {ω' : R'} {ω'' : R''} {C : Ell H R ω} {C' : Ell H R' ω'} {C'' : Ell H R'' ω''} {φ : C'.P.A.left ⟶ C.P.A.left} {ψ : C''.P.A.left ⟶ C.P.A.left} (h : EllMap C' C f φ) (k : EllMap C'' C (f'.comp f) ψ) : ∃ φ' : C''.P.A.left ⟶ C'.P.A.left, EllMap C'' C' f' φ' ∧ φ' ≫ φ = ψ

theorem **ellMap_cancel_and_sabMap_cancel** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R R' R'' : Type) [CommRing R] [CommRing R'] [CommRing R''] (f : R →+* R') (f' : R' →+* R''), (∀ (ω : R) (ω' : R') (ω'' : R'') (C : Ell H R ω) (C' : Ell H R' ω') (C'' : Ell H R'' ω'') (φ : C'.P.A.left ⟶ C.P.A.left) (ψ : C''.P.A.left ⟶ C.P.A.left), EllMap C' C f φ → EllMap C'' C (f'.comp f) ψ → ∃ φ' : C''.P.A.left ⟶ C'.P.A.left, EllMap C'' C' f' φ' ∧ φ' ≫ φ = ψ) ∧ ∀ (S : Scheme.{0}) (y : Sp R' ⟶ S) (y' : Sp R'' ⟶ S) (t : SAB H S) (t' : SAB H (Sp R')) (t'' : SAB H (Sp R'')) (m : SABMap t' t y) (n : SABMap t'' t y'), spMap f' ≫ y = y' → ∃ m' : SABMap t'' t' (spMap f'), m'.φ ≫ m.φ = n.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EllPullExistsOAct

theorem **ellPull_exists_oAct** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {f : S' ⟶ S} {g : G'.left ⟶ G.left} (hg : IsGrpPullback G' G f g) {O : Type} [CommRing O] (o : OAct O G) : ∃ o' : OAct O G', ∀ a, (o'.act a).left ≫ g = g ≫ (o.act a).left

theorem **ellPull_prim** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {f : S' ⟶ S} {g : G'.left ⟶ G.left} (hg : IsGrpPullback G' G f g) {L : LB G.left} {L' : LB G'.left} (hL : L.IsLB) (hL' : L'.IsLB) (i : LBIso (L.pull g) L') {w : G ⟶ G} {w' : G' ⟶ G'} (hw : w'.left ≫ g = g ≫ w.left) (h : LBIso L (cubeLB G ![𝟙 G, w] L)) : LBIso L' (cubeLB G' ![𝟙 G', w'] L')

theorem **exists_ellMap** : Out.PolBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (R' : Type) [CommRing R'] (f : R →+* R'), ∃ (C' : Ell H R' (f ω)) (φ : C'.P.A.left ⟶ C.P.A.left), EllMap C' C f φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EllipticCurveLatticeBaseChangeStable

theorem **curve_lattice_baseChange** : ∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (Lf : (R →+* ℂ) → PeriodPair) (R₁ : Type) [CommRing R₁] (f : R →+* R₁) (P₁ : Out.PolAV (Sp R₁)) (φ : P₁.A.left ⟶ P.A.left) (v₁ : epsR R₁ ⟶ P₁.A), (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : R →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ (Lf s').lattice ↔ z • w' ∈ u'.Λ) → IsPolFibre f P P₁ φ → IsVecOver f φ v v₁ → (P₁.g = 1 ∧ IsLieBasis ![v₁] ∧ (∀ (R' : Type) [CommRing R'] (f' : R₁ →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P₁.A.

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EllipticCurveLatticeBaseChangeStableStatement

theorem **_root_.GlobalTheory.T.ellipticCurve_lattice_baseChange_stable** : ∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (Lf : (R →+* ℂ) → PeriodPair) (R₁ : Type) [CommRing R₁] (f : R →+* R₁) (P₁ : Out.PolAV (Sp R₁)) (φ : P₁.A.left ⟶ P.A.left) (v₁ : epsR R₁ ⟶ P₁.A), (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : R →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ (Lf s').lattice ↔ z • w' ∈ u'.Λ) → IsPolFibre f P P₁ φ → IsVecOver f φ v v₁ → (P₁.g = 1 ∧ IsLieBasis ![v₁] ∧ (∀ (R' : Type) [CommRing R'] (f' : R₁ →+* R') (P' : Out.P

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqOfGeomPointsAlg

theorem **eq_of_geomPoints_alg** (R K A : Type) [CommRing R] [IsDomain R] [Field K] [CharZero K] [Algebra R K] [IsFractionRing R K] [CommRing A] [Algebra R A] [Module.Flat R A] (het : (Algebra.TensorProduct.includeLeftRingHom : K →+* K ⊗[R] A).Etale) (a b : A) (h : ∀ (k' : Type) [Field k'] [IsAlgClosed k'] [CharZero k'] (ψ : A →+* k'), ψ a = ψ b) : a = b

theorem **eq_of_geomPoints** : ∀ (R K : Type) [CommRing R] [IsDomain R] [Field K] [CharZero K] [Algebra R K] [IsFractionRing R K] (T : Over (Sp R)), IsFinite T.hom → Flat T.hom → Etale (pullback.snd T.hom (spMap (algebraMap R K))) → ∀ a b : Γ(T.left, ⊤), (∀ (k' : Type) [Field k'] [IsAlgClosed k'] [CharZero k'] (β : R →+* k') (t : Over.mk (spMap β) ⟶ T), t.left.appTop.hom a = t.left.appTop.hom b) → a = b

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqOfLamEq

theorem **eq_one_of_gen_cubeLB** {S : Scheme.{0}} {A : Over S} [GrpObj A] [IsCommMonObj A] {L : LB A.left} (hL : L.IsLB) (hK : Out.KTriv A L) {T : Over S} (x : T ⟶ A) (g₁ : ∃ t : LB.Gen (cubeLB A ![fst T A ≫ x, snd T A] L), t.IsGen) : x = 1

theorem **eq_of_lamEq** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) (X : Over S) (f g : X ⟶ P.A), LamEq P.L f (𝟙 P.A) P.L g (𝟙 P.A) → f = g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqOfLamEqStatement

theorem **_root_.GlobalTheory.T.eq_of_lamEq_polAV** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) (X : Over S) (f g : X ⟶ P.A), LamEq P.L f (𝟙 P.A) P.L g (𝟙 P.A) → f = g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqOfLieBasisComplex

theorem **exists_lift_isMonHom** {S S' : Scheme.{0}} {j : S' ⟶ S} {G H : Over S} {G' H' : Over S'} [GrpObj G] [GrpObj H] [GrpObj G'] [GrpObj H'] {φ : G'.left ⟶ G.left} {ψ : H'.left ⟶ H.left} (hG : IsGrpPullback G' G j φ) (hH : IsGrpPullback H' H j ψ) (f : G ⟶ H) [IsMonHom f] : ∃ f' : G' ⟶ H', IsMonHom f' ∧ f'.left ≫ ψ = φ ≫ f.left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqOrEqNegOfPurelyImaginaryOfNormEq

theorem **eq_or_eq_neg_of_purelyImaginary_of_norm_eq** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (ξ ξ' : E) (τ : E →+* ℂ), cc ξ = -ξ → cc ξ' = -ξ' → ‖τ ξ‖ = ‖τ ξ'‖ → ξ' = ξ ∨ ξ' = -ξ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqZeroOfEventuallyDotProductStarEqZero

private lemma **dotProduct_mulVec_star_add_smul_expand** (C : Matrix (Fin r) (Fin r) ℂ) (a b : ℂ) (x y : Fin r → ℂ) : (a • x + b • y) ⬝ᵥ C *ᵥ star (a • x + b • y) = a * star a * (x ⬝ᵥ C *ᵥ star x) + b * star b * (y ⬝ᵥ C *ᵥ star y) + a * star b * (x ⬝ᵥ C *ᵥ star y) + b * star a * (y ⬝ᵥ C *ᵥ star x)

theorem **eq_zero_of_forall_dotProduct_star_eq_zero** {C : Matrix (Fin r) (Fin r) ℂ} (h : ∀ u : Fin r → ℂ, u ⬝ᵥ C *ᵥ star u = 0) : C = 0

theorem **eq_zero_of_eventually_dotProduct_star_eq_zero** {C : Matrix (Fin r) (Fin r) ℂ} {v : Fin r → ℂ} (h : ∀ᶠ w in 𝓝 v, w ⬝ᵥ C *ᵥ star w = 0) : C = 0

theorem **dotProduct_star_eq_re** {A : Matrix (Fin r) (Fin r) ℂ} (hA : A.IsHermitian) (w : Fin r → ℂ) : w ⬝ᵥ A *ᵥ star w = ((w ⬝ᵥ A *ᵥ star w).re : ℂ)

private lemma **transpose_col_mul_mul_map_conj_apply** (B : Matrix (Fin r) (Fin r) ℂ) (x : Fin r → ℂ) (i j : Fin 1) : ((Matrix.of fun k (_ : Fin 1) => x k)ᵀ * B * (Matrix.of fun k (_ : Fin 1) => x k).map (starRingEnd ℂ)) i j = x ⬝ᵥ B *ᵥ star x

private lemma **of_col_mulVec_eq_smul** (x : Fin r → ℂ) (a : Fin 1 → ℂ) : (Matrix.of fun k (_ : Fin 1) => x k) *ᵥ a = a 0 • x

private lemma **mul_of_col_eq_of_mulVec** (g : Matrix (Fin r) (Fin r) ℂ) (x : Fin r → ℂ) : g * (Matrix.of fun k (_ : Fin 1) => x k) = Matrix.of fun k (_ : Fin 1) => (g *ᵥ x) k

theorem **exists_unitary_mulVec_eq** {B : Matrix (Fin r) (Fin r) ℂ} (hB : B.IsHermitian) (hdet : B.det ≠ 0) {v w : Fin r → ℂ} (hv : v ≠ 0) (hw : w ≠ 0) (h : w ⬝ᵥ B *ᵥ star w = v ⬝ᵥ B *ᵥ star v) : ∃ g : Matrix (Fin r) (Fin r) ℂ, gᵀ * B * g.map (starRingEnd ℂ) = B ∧ g *ᵥ v = w

private lemma **dotProduct_mulVec_star_real_smul** (C : Matrix (Fin r) (Fin r) ℂ) (a : ℝ) (x : Fin r → ℂ) : ((a : ℂ) • x) ⬝ᵥ C *ᵥ star ((a : ℂ) • x) = ((a ^ 2 : ℝ) : ℂ) * (x ⬝ᵥ C *ᵥ star x)

theorem **eq_smul_of_norm_eq_one_on_orbit** : ∀ (r : ℕ) (A B : Matrix (Fin r) (Fin r) ℂ) (v : Fin r → ℂ), A.IsHermitian → B.IsHermitian → B.det ≠ 0 → (∀ h : Matrix (Fin r) (Fin r) ℂ, hᵀ * B * h.map (starRingEnd ℂ) = B → ‖(h *ᵥ v) ⬝ᵥ (A *ᵥ star (h *ᵥ v))‖ = 1) → ∃ t : ℝ, t ≠ 0 ∧ A = (t : ℂ) • B

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqZeroOfPow

theorem **push_tanExp** (Φ : (epsR R ⟶ G') ≃* (Over.mk ((epsR R).hom ≫ f) ⟶ G)) (hΦ : ∀ x, (Φ x).left = x.left ≫ φ) {n : ℕ} (v : Fin n → (epsR R ⟶ G')) (c : Fin n → R) : Φ (tanExp v c) = ∏ i, scaleOver f (c i) ≫ Φ (v i)

theorem **prod_comp** {T G₁ : Over S} [GrpObj G₁] [IsCommMonObj G₁] (F : G ⟶ G₁) [IsMonHom F] {J : Type} (s : Finset J) (x : J → (T ⟶ G)) : (∏ i ∈ s, x i) ≫ F = ∏ i ∈ s, x i ≫ F

theorem **eq_zero_of_pow** {K : Type} [Field K] [CharZero K] {A : Over (Sp K)} [GrpObj A] [IsCommMonObj A] {n : ℕ} {v : Fin n → (epsR K ⟶ A)} (hv : IsLieBasis v) {c : Fin n → K} {N : ℕ} (hN : 0 < N) (h : tanExp v c ^ N = 1) : c = 0

theorem **lamInd** : LamIndSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqZeroOfSumEmbeddingMul

theorem **_root_.GlobalTheory.EigenNorm.hermForm_add_left** (z z' w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm (z + z') w = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z w + _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z' w

theorem **_root_.GlobalTheory.EigenNorm.hermForm_add_right** (z w w' : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z (w + w') = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z w + _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z w'

theorem **_root_.GlobalTheory.EigenNorm.hermForm_smul_left** (r : ℂ) (z w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm (r • z) w = r * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z w

theorem **_root_.GlobalTheory.EigenNorm.hermForm_smul_right** (r : ℂ) (z w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z (r • w) = starRingEnd ℂ r * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z w

theorem **hermForm_sum_left** {κ : Type*} (s : Finset κ) (z : κ → Fin g → ℂ) (w : Fin g → ℂ) : hermForm Hm (∑ i ∈ s, z i) w = ∑ i ∈ s, hermForm Hm (z i) w

theorem **hermForm_sum_right** {κ : Type*} (s : Finset κ) (z : Fin g → ℂ) (w : κ → Fin g → ℂ) : hermForm Hm z (∑ i ∈ s, w i) = ∑ i ∈ s, hermForm Hm z (w i)

theorem **_root_.GlobalTheory.EigenNorm.hermForm_conj** (hH : Hm.IsHermitian) (z w : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm w z = starRingEnd ℂ (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z w)

theorem **eq_zero_of_sum_embedding_mul** {E : Type} [Field E] [NumberField E] (f : (E →+* ℂ) → ℂ) (hf : ∀ s : 𝒪 E, ∑ τ : E →+* ℂ, τ (s : E) * f τ = 0) (τ : E →+* ℂ) : f τ = 0

theorem **conj_eq_neg_of_sq** {ρ : ℂ} {d : ℕ} (hd : d ≠ 0) (h : ρ ^ 2 = -(d : ℂ)) : starRingEnd ℂ ρ = -ρ

theorem **hermForm_eigen** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (g : ℕ) (Hm : Matrix (Fin g) (Fin g) ℂ), Hm.IsHermitian → ∀ (L : Type) [AddCommGroup L] [Module (𝒪 E) L] (h : L → L → E) (ρ : ℂ) (c : ℝ) (x : (E →+* ℂ) → L →+ (Fin g → ℂ)), ρ ^ 2 = -(H.d : ℂ) → (∀ a b e, h (a + b) e = h a e + h b e) → (∀ (s : 𝒪 E) a e, h (s • a) e = s * h a e) → (∀ a b, h b a = IsCMField.complexConj E (h a b)) → (∀ τ (s : 𝒪 E) a, x τ (s • a) = τ (s : E) • x τ a) → (∀ τ, τ (H.rt : E) ≠ ρ → x τ = 0) → (∀ a, hermForm Hm (∑ τ, x τ a) (∑ τ, x τ a) = c * ∑ τ ∈ Finset.univ.filter (fun τ : E →+* ℂ => τ (H.rt : E) = ρ), τ (h a a)) → ∀ τ τ' a b, τ (H.rt : E) = ρ → hermForm Hm (x τ a) (x τ' b) = open Classical in if τ' = τ then c * τ (h a b) else 0

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EqZeroOfSumEmbeddingSmul

theorem **primeSpectrum_dualNumber_eq** (x y : PrimeSpectrum (DualNumber ℂ)) : x = y

theorem **isDiff_of_isLieComb** {w : Fin g → ℂ} {v v' : epsR ℂ ⟶ A} (h : u.IsDiff w v) (c : ℂ) (hc : IsLieComb ![c] ![v] v') : u.IsDiff (c • w) v'

theorem **lift_unique** {g₀ : ℕ} (F F' : (Fin g₀ → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : ∀ z, u.ex (F z) = u.ex (F' z)) : F = F'

theorem **exists_eigen_components** {E : Type} [Field E] [NumberField E] {L V : Type} [AddCommGroup L] [Module (𝒪 E) L] [AddCommGroup V] [Module ℂ V] (X : L →+ V) : ∃ x : (E →+* ℂ) → L →+ V, ∀ (s : 𝒪 E) a, X (s • a) = ∑ τ : E →+* ℂ, τ (s : E) • x τ a

theorem **eq_zero_of_sum_embedding_smul** {E : Type} [Field E] [NumberField E] {g : ℕ} (f : (E →+* ℂ) → Fin g → ℂ) (hf : ∀ s : 𝒪 E, ∑ τ : E →+* ℂ, τ (s : E) • f τ = 0) (τ : E →+* ℂ) : f τ = 0

theorem **quadratic_eq_zero** {L : Type} [AddCommGroup L] (B : L → L → ℂ) (hl : ∀ a b c, B (a + b) c = B a c + B b c) (hr : ∀ a b c, B a (b + c) = B a b + B a c) {m : ℕ} (e : Fin m → L) (he : Submodule.span ℤ (Set.range e) = ⊤) (h1 : ∀ i, B (e i) (e i) = 0) (h2 : ∀ i j, B (e i + e j) (e i + e j) = 0) (a : L) : B a a = 0

theorem **exists_eigen_hermForm_eq** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (P₀ P : Out.PolAV (Sp ℂ)), P₀.g = 1 → ∀ (u₀ : HolUnif P₀.A P₀.L P₀.g) (u : HolUnif P.A P.L P.g) (H₀ : Matrix (Fin P₀.g) (Fin P₀.g) ℂ) (χ₀ : u₀.Λ → ℂ) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ), u₀.IsAH H₀ χ₀ → u.IsAH Hm χ → (∀ v, IsTangent v → ∃! w, u.IsDiff w v) → (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) → ∀ (act : OAct (𝒪 E) P.A) (L : Type) [AddCommGroup L] [Module (𝒪 E) L] (h : L → L → E) (ρ : ℂ) (φ : L → (P₀.A ⟶ P.A)) (v₀ : epsR ℂ ⟶ P₀.A) (w₀ : Fin P₀.g → ℂ) (m : ℕ) (e : Fin m → L) (nn : L → ℕ), ρ ^ 2 = -(H.d : ℂ) → (∀ a b c, h (a + b) c = h a c + h b c) → (∀ (s : 𝒪 E) a c, h (s • a) c = s * h a c) → (∀ a b, h b a = IsCMField.complexConj E (h a b)) → (∀ a, (nn a : ℂ) = ∑ τ ∈ Finset.univ.filter (fun τ : E →+* ℂ => τ (H.rt : E) = ρ), τ (h a a)) → (∀ a, IsMonHom (φ a)) → (∀ a b, φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.EvalFrEqProdDetPowMul

theorem **evalFr_map_blockDetPoly** {K S : Type} [CommRing K] [CommRing S] (φ : K →+* S) {B : Type} [Fintype B] (m : B → ℕ) (b : B) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) S) : evalFr (MvPolynomial.map φ (blockDetPoly K m b)) X = (X b).det

theorem **evalFr_blockDetPoly** {K : Type} [CommRing K] {B : Type} [Fintype B] (m : B → ℕ) (b : B) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) : evalFr (blockDetPoly K m b) X = (X b).det

theorem **exists_eq_prod_blockDetPoly_pow_mul** (K : Type) [Field K] [Infinite K] (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (a : ∀ b, Fin (m b) → ℕ) (s : ℕ) : ∀ (n : B → ℕ), ∑ b, n b = s → ∀ (P : MvPolynomial (FrVar m) K), (∀ (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) (b : B) (t : Matrix (Fin (m b)) (Fin (m b)) K), (∀ b', (X b').det ≠ 0) → t.det ≠ 0 → t.BlockTriangular id → evalFr P (Function.update X b (X b ∗ t)) = (∏ i, t i i ^ (a b i + n b)) ∗ evalFr P X) → ∃ G : MvPolynomial (FrVar m) K, P = (∏ b, blockDetPoly K m b ^ n b) ∗ G

theorem **evalFr_eq_prod_det_pow_mul** : ∀ (K : Type) [Field K] [Infinite K] (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (P : MvPolynomial (FrVar m) K) (a : ∀ b, Fin (m b) → ℕ) (N : ℕ), (∀ (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) (b : B) (t : Matrix (Fin (m b)) (Fin (m b)) K), (∀ b', (X b').det ≠ 0) → t.det ≠ 0 → t.BlockTriangular id → evalFr P (Function.update X b (X b ∗ t)) = (∏ i, t i i ^ (a b i + N)) ∗ evalFr P X) → ∃ G : MvPolynomial (FrVar m) K, ∀ (S : Type) [CommRing S] (φ : K →+* S) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) S), evalFr (MvPolynomial.map φ P) X = (∏ b, (X b).det) ^ N ∗ evalFr (MvPolynomial.map φ G) X

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsAbelianSchemeModelOfIsogenyDedekind

theorem **exists_kernel** {S : Scheme.{0}} {G H : Over S} [GrpObj G] [GrpObj H] [IsSeparated H.hom] (f : G ⟶ H) [IsMonHom f] : ∃ (Z : Over S) (_ : GrpObj Z) (i : Z ⟶ G), IsMonHom i ∧ IsClosedImmersion i.left ∧ IsPullback i (toUnit Z) f η[H]

theorem **isPullback_kernel_baseChange** {S S' : Scheme.{0}} (j : S' ⟶ S) {Z G H : Over S} [GrpObj H] {i : Z ⟶ G} {f : G ⟶ H} (h : IsPullback i (toUnit Z) f η[H]) : IsPullback ((Over.pullback j).map i) (toUnit _) ((Over.pullback j).map f) η[(Over.pullback j).obj H]

theorem **isFinite_of_pow_eq_one** {S : Scheme.{0}} {Z G : Over S} [GrpObj G] [IsSeparated G.hom] (i : Z ⟶ G) [IsClosedImmersion i.left] (n : ℕ) [hn : IsFinite (𝟙 G ^ n).left] (h : i ^ n = 1) : IsFinite Z.hom

theorem **exists_abelianScheme_model_of_isogeny_dedekind** : Out.Laws → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (P : Out.PolAV (Sp R)) (A₀ B₀ : Over (Sp K)) [GrpObj A₀] [GrpObj B₀] (φ : A₀.left ⟶ P.A.left) (ψ : A₀ ⟶ B₀) (n : ℕ), IsGrpPullback A₀ P.A (spMap (algebraMap R K)) φ → IsAbelianScheme B₀ → IsMonHom ψ → IsFinite ψ.left → Flat ψ.left → Surjective ψ.left → 1 ≤ n → (∀ (T : Over (Sp K)) (x : T ⟶ A₀), x ≫ ψ = 1 → x ^ n = 1) → ∃ (B : Over (Sp R)) (_ : GrpObj B) (φB : B₀.left ⟶ B.left) (Ψ : P.A ⟶ B), IsAbelianScheme B ∧ IsGrpPullback B₀ B (spMap (algebraMap R K)) φB ∧ IsMonHom Ψ ∧ IsFinite Ψ.left ∧ Flat Ψ.left ∧ Surjective Ψ.left ∧ ψ.left ≫ φB = φ ≫ Ψ.left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsAppellHumbertUniformisationOfKeyDatumIso

theorem **eq_comp_of_isLieComb** {A A' : Over (Sp ℂ)} [GrpObj A] [GrpObj A'] (ψ : A' ⟶ A) (hm : ∀ a b : epsR ℂ ⟶ A', (a ∗ b) ≫ ψ = (a ≫ ψ) ∗ (b ≫ ψ)) {n : ℕ} {c : Fin n → ℂ} {v' : Fin n → (epsR ℂ ⟶ A')} {v : Fin n → (epsR ℂ ⟶ A)} (hv : ∀ i, (v i).left = (v' i).left ≫ ψ.left) {w' : epsR ℂ ⟶ A'} {w : epsR ℂ ⟶ A} (hw' : IsLieComb c v' w') (hw : IsLieComb c v w) : w = w' ≫ ψ

theorem **eq_comp_of_isLieComb_cast** {A A' : Over (Sp ℂ)} [GrpObj A] [GrpObj A'] (ψ : A' ⟶ A) (hm : ∀ a b : epsR ℂ ⟶ A', (a ∗ b) ≫ ψ = (a ≫ ψ) ∗ (b ≫ ψ)) {n n' : ℕ} (h : n = n') {c : Fin n' → ℂ} {v' : Fin n' → (epsR ℂ ⟶ A')} {v : Fin n → (epsR ℂ ⟶ A)} (hv : ∀ i, (v i).left = (v' (i.cast h)).left ≫ ψ.left) {w' : epsR ℂ ⟶ A'} {w : epsR ℂ ⟶ A} (hw' : IsLieComb c v' w') (hw : IsLieComb (fun i => c (i.cast h)) v w) : w = w' ≫ ψ

theorem **exists_push_key** {x x' : KeyIn ℂ} (H : KeyPB x' x (RingHom.id ℂ)) {g : ℕ} (u' : HolUnif x'.P.A x'.P.L g) : ∃ u : HolUnif x.P.A x.P.L g, u.Λ = u'.Λ ∧ (∀ Hm χ', u'.IsAH Hm χ' → ∃ χ, u.IsAH Hm χ) ∧ (u'.IsRig x'.rig → u.IsRig x.rig) ∧ (∀ θ, u'.IsFn x'.th θ → u.IsFn x.th θ) ∧ ∀ (h : x.P.g = x'.P.g) (c : Fin x'.P.g → ℂ) (w' : epsR ℂ ⟶ x'.P.A) (w : epsR ℂ ⟶ x.P.A) (W : Fin g → ℂ), IsLieComb c x'.v w' → IsLieComb (fun i => c (i.cast h)) x.v w → u'.IsDiff W w' → u.IsDiff W w

theorem **uniformisation_transport** : ∀ (x x' : KeyIn ℂ), KeyPB x' x (RingHom.id ℂ) → ∀ (hg : x'.P.g = x.P.g) (g : ℕ) (u : HolUnif x.P.A x.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume → ∃ (u' : HolUnif x'.P.A x'.P.L g) (χ' : u'.Λ → ℂ) (θ' : (Fin g → ℂ) → ℂ) (Fd' : Set (Fin g → ℂ)), u'.IsAH (Hm.map (RingHom.id ℂ)) χ' ∧ u'.IsRig x'.rig ∧ u'.IsFn x'.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume ∧ (∫ z in Fd', ‖θ' z‖ ^ 2 ∗ Real.exp (-Real.pi ∗ (hermForm (Hm.map (RingHom.id ℂ)) z z).re)) / (volume Fd').toReal = (∫ z in Fd, ‖θ z‖ ^ 2 ∗ Real.exp (-Real.pi ∗ (hermForm Hm z z).re)) / (volume Fd).toReal ∧ ∀ (cf : Fin x.P.g → ℂ) (w : epsR ℂ ⟶ x.P.A) (w' : epsR ℂ ⟶ x'.P.A) (W : Fin g → ℂ), IsLieComb cf x.v w → IsLieComb (fun i => (RingHom.id ℂ) (cf (i.cast hg))) x'.v

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsAppellHumbertUniformisationOfKeyDatumIsoStatement

theorem **_root_.GlobalTheory.T.exists_appellHumbert_uniformisation_of_keyDatumIso** : ∀ (x x' : KeyIn ℂ), KeyPB x' x (RingHom.id ℂ) → ∀ (hg : x'.P.g = x.P.g) (g : ℕ) (u : HolUnif x.P.A x.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume → ∃ (u' : HolUnif x'.P.A x'.P.L g) (χ' : u'.Λ → ℂ) (θ' : (Fin g → ℂ) → ℂ) (Fd' : Set (Fin g → ℂ)), u'.IsAH (Hm.map (RingHom.id ℂ)) χ' ∧ u'.IsRig x'.rig ∧ u'.IsFn x'.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume ∧ (∫ z in Fd', ‖θ' z‖ ^ 2 ∗ Real.exp (-Real.pi ∗ (hermForm (Hm.map (RingHom.id ℂ)) z z).re)) / (volume Fd').toReal = (∫ z in Fd, ‖θ z‖ ^ 2 ∗ Real.exp (-Real.pi ∗ (hermForm Hm z z).re)) / (volume Fd).toReal ∧ ∀ (cf : Fin x.P.g → ℂ) (w : epsR ℂ ⟶ x.P.A) (w' : epsR ℂ ⟶ x'.P.A) (W : Fin g → ℂ), IsLieComb cf x.v w → IsLieComb

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCdvrCompletionIntegralIff

theorem **univFrame_borelPairs** : Out.SecBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (p A : TRing c.F) (j : Sp p.R ⟶ Xo.XF c) (y₀ : FrPt Xo c p.R) (π : p.R →ₐ[c.F] A.R) (X : (∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A.R)) (s : (A.R)ˣ) (u : FrPt Xo c A.R), OpenChart j y₀ → UnivFrame y₀ π X s u → ∃ (B : TRing c.F) (uB uB' : FrPt Xo c B.R) (tB : Ix Dw → (B.R)ˣ) (q q' : A.R →ₐ[c.F] B.R), FrMap q u uB ∧ FrMap q' u uB' ∧ FrBor tB uB uB' ∧ ∀ (T : Type) [CommRing T] [Algebra c.F T] (a a' : A.R →ₐ[c.F] T) (z z' : FrPt Xo c T) (t : Ix Dw → Tˣ), FrMap a u z → FrMap a' u z' → FrBor t z z' → ∃ b : B.R →ₐ[c.F] T, b.comp q = a ∧ b.comp q' = a' ∧ ∀ i, b (tB i : B.R) = (t i : T)

theorem **ChartCover.exists_lift** {S : Type} [CommRing S] [Algebra c.F S] (w : FrPt Xo c S) : ∃ xF : Sp S ⟶ Xo.XF c, xF ≫ pullback.fst _ _ = w.x ∧ xF ≫ pullback.snd _ _ = spMap (algebraMap c.F S)

theorem **ChartCover.near** (hS : Out.SecBaseChangeSt) {R A : Type} [CommRing R] [Algebra c.F R] [CommRing A] [Algebra c.F A] {j : Sp R ⟶ Xo.XF c} {y₀ : FrPt Xo c R} {π : R →ₐ[c.F] A} {X : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A} {s : Aˣ} {u : FrPt Xo c A} (hj : OpenChart j y₀) (hu : UnivFrame y₀ π X s u) {S : Type} [CommRing S] [Algebra c.F S] (w : FrPt Xo c S) (xF : Sp S ⟶ Xo.XF c) (h1 : xF ≫ pullback.fst _ _ = w.x) (h2 : xF ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (p : Sp S) (hp : xF p ∈ Set.range j) : ∃ g : S, g ∉ p.asIdeal ∧ ∀ (T : Type) [CommRing T] [Algebra c.F T] (lam : S →ₐ[c.F] T), IsUnit (lam g) → ∃ (z : FrPt Xo c T) (a : A →ₐ[c.F] T), FrMap lam w z ∧ FrMap a u z

theorem **ChartCover.cls_loc** (hS : Out.SecBaseChangeSt) {R : Type} [CommRing R] [Algebra c.F R] {A : TRing c.F} {j : Sp R ⟶ Xo.XF c} {y₀ : FrPt Xo c R} {π : R →ₐ[c.F] A.R} {X : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) A.R} {s : (A.R)ˣ} {u : FrPt Xo c A.R}

(hj : OpenChart j y₀) (hu : UnivFrame y₀ π X s u) {S : Type} [CommRing S] [Algebra c.F S] (w : FrPt Xo c S) (xF : Sp S ⟶ Xo.XF c) (h1 : xF ≫ pullback.fst _ _ = w.x) (h2 : xF ≫ pullback.snd _ _ = spMap (algebraMap c.F S)) (m : Ideal S) [hm : m.IsPrime] (hp : xF ⟨m, hm⟩ : PrimeSpectrum S) ∈ Set.range j) : ∃ cw : A.R →ₐ[c.F] Localization.AtPrime m, ChartCls (K

theorem **chartCoverF_of_secBaseChange** : Out.SecBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), ChartCoverF Xo c

theorem **GlobalTheory.LamLoc.evR_frBor** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S : Type) [CommRing S] [_root_.IsReduced S] [Algebra c.F S] (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) : ∀ f ∈ sec Xo c a n, ∀ (t : Ix Dw → S˟) (y y' : FrPt Xo c S), FrBor t y y' → evR Xo c f y' = wchar (ex Dw a n) t * evR Xo c f y

theorem **one_push₀** (h : IsGrpPullback G' G f φ) (T : Over S') : (1 : Over.mk (T.hom ≫ f) ⟶ G).left = (1 : T ⟶ G').left ≫ φ

theorem **one_push** (h : IsGrpPullback G' G f φ) (T' : Over S') (T : Over S₀) (g : T'.left ⟶ T.left) (hg : g ≫ T.hom = T'.hom ≫ f) : (1 : T' ⟶ G').left ≫ φ = g ≫ (1 : T ⟶ G).left

theorem **mul_push** (h : IsGrpPullback G' G f φ) (T' : Over S') (T : Over S₀) (g : T'.left ⟶ T.left) (hg : g ≫ T.hom = T'.hom ≫ f) (a b : T' ⟶ G') (a₀ b₀ : T ⟶ G) (ha : a.left ≫ φ = g ≫ a₀.left) (hb : b.left ≫ φ = g ≫ b₀.left) : (a * b).left ≫ φ = g ≫ (a₀ * b₀).left

theorem **prod_push** (h : IsGrpPullback G' G f φ) (T' : Over S') (T : Over S₀) (g : T'.left ⟶ T.left) (hg : g ≫ T.hom = T'.hom ≫ f) {m : ℕ} (a : Fin m → (T' ⟶ G')) (a₀ : Fin m → (T ⟶ G)) (ha : ∀ j, (a j).left ≫ φ = g ≫ (a₀ j).left) : ((List.ofFn a).prod).left ≫ φ = g ≫ ((List.ofFn a₀).prod).left

theorem **spMap_comp₂** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **spMap_frac_cancel** {V K A : Type} [CommRing V] [CommRing K] [Algebra V K] [IsFractionRing V K] [CommRing A] (g h : Sp A ⟶ Sp K) (e : g ≫ spMap (algebraMap V K) = h ≫ spMap (algebraMap V K)) : g = h

theorem **dualMap_alg** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) : (dualMap f).comp (algebraMap A (DualNumber A)) = (algebraMap B (DualNumber B)).comp f

theorem **dualMap_epsMul** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (a : A) : (dualMap f).comp (epsMul a) = (epsMul (f a)).comp (dualMap f)

theorem **comb_eq** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {m : ℕ} {v : Fin m → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)} (hv : ∀ j, IsTanAt x (v j)) {a : Fin m → S} {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (h : IsCombAt x v a w) : (combT x (fun j => ovT (t

theorem **lieInt_change** (T : CDVR) (io : c.F →+* T.Fr) {G : SAB H (Sp T.V)} {t' : SAB H (Sp T.Fr)} {x : Sp T.Fr ⟶ Xo.X} (m1 : SABMap t' Xo.univ.t x) (m2 : SABMap t' G (spMap (algebraMap T.V T.Fr))) (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber T.Fr) ⟶ Xo.univ.t.G.left)) (hu : ∀ τ j, IsTanAt x (u τ j)) (h : LieInt T io m1 m2 u) (k : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) T.V) (hk : ∀ τ, IsUnit (k τ).det) : LieInt T io m1 m2 (fun τ i => (combT x (fun j => ovT (t

theorem **sp_comp** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

theorem **dual_alg** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) : spMap (Formalism.L2.Pj.dualMap ψ) ≫ spMap (algebraMap S (DualNumber S)) = spMap (algebraMap T (DualNumber T)) ≫ spMap ψ

theorem **prod_pull** {S' S₀ : Scheme.{0}} {G' : Over S'} [GrpObj G'] {G : Over S₀} [GrpObj G] {f : S' ⟶ S₀} {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) (T' : Over S') (T : Over S₀) (g : T.left ⟶ T'.left) (hg : g ≫ T'.hom ≫ f = T.hom) {m : ℕ} (a : Fin m → (T' ⟶ G')) (a₀ : Fin m → (T ⟶ G)) (ha : ∀ j, (a₀ j).left = g ≫ (a j).left ≫ φ) : ((List.ofFn a₀).prod).left = g ≫ ((List.ofFn a).prod).left ≫ φ

theorem **comb_match** {V K F : Type} [CommRing V] [CommRing K] [CommRing F] {X : Scheme.{0}} {B : Over X} [GrpObj B] {G₁ : Over (Sp V)} [GrpObj G₁] {G₂ : Over (Sp K)} [GrpObj G₂] {f₁ : Sp V ⟶ X} {f₂ : Sp K ⟶ X} {φ₁ : G₁.left ⟶ B.left} {φ₂ : G₂.left ⟶ B.left} (h₁ : IsGrpPullback G₁ B f₁ φ₁) (h₂ : IsGrpPullback G₂ B f₂ φ₂) (α : V →+* F) (β : K →+* F) (hf : spMap α ≫ f₁ = spMap β ≫ f₂) {m : ℕ} {v₁ : Fin m → (epsR V ⟶ G₁)} {v₂ : Fin m → (epsR K ⟶ G₂)} (hv : ∀ i, spMap (dualMap α) ≫ (v₁ i).left ≫ φ₁ = spMap (dualMap β) ≫ (v₂ i).left ≫ φ₂) {c₁ : Fin m → V} {c₂ : Fin m → K} (hc : ∀ i, α (c₁ i) = β (c₂ i)) {w₁ : epsR V ⟶ G₁} {w₂ : epsR K ⟶ G₂} (hw₁ : IsLieComb c₁ v₁ w₁) (hw₂ : IsLieComb c₂ v₂ w₂) : spMap (dualMap α) ≫ w₁.left ≫ φ₁ = spMap (dualMap β) ≫ w₂.left ≫ φ₂

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmAdjoinActionOfEquation

private theorem **okAct_group_identity** {A : Type*} [CommGroup A] (i w : A) (k m n m' n' : ℤ) : i ^ (m * m' - n * n' * k) * w ^ (m * n' + n * m' + n * n') = (i ^ m * w ^ n) ^ m' * (w ^ m * (w * i ^ (-k)) ^ n) ^ n'

theorem **exists_okAct_of_equation** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (W : G ⟶ G), IsMonHom W → (W ≫ W) * (𝟙 G) ^ ((1 + H.d) / 4) = W → ∃ a : OAct (OK H) G, a.act (wK H) = W ∧ ∀ m n : ℤ, a.act ((m : OK H) + (n : OK H) * wK H) = (𝟙 G) ^ m * W ^ n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmAdjoinActionOfEquationStatement

theorem **_root_.GlobalTheory.T.exists_cmAdjoin_action_of_equation** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (W : G ⟶ G), IsMonHom W → (W ≫ W) * (𝟙 G) ^ ((1 + H.d) / 4) = W → ∃ a : OAct (OK H) G, a.act (wK H) = W ∧ ∀ m n : ℤ, a.act ((m : OK H) + (n : OK H) * wK H) = (𝟙 G) ^ m * W ^ n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmCurveHomFamilyToFibreAtPeriod

theorem **two_mul_sub_one_sq_of_ringHom** (τ : OK H →+* ℂ) : (2 * τ (wK H) - 1) ^ 2 = -(H.d : ℂ)

theorem **exists_hermitianPairing_eq_im_mul_pureType** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) {ω : ℂ} {x : D.La} (hx : x ∈ PureType.lat hP (2 * ω - 1)) (h0 : x ≠ 0) : ∃ r : ℝ, 0 < r ∧ ((D.EF (H.w0 • x) x : ℤ) : ℝ) = ω.im * r

theorem **exists_isFam_complex** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (ω : ℂ) (C : Ell H ℂ ω) (u₀ : HolUnif C.P.A C.P.L C.P.g) (c : ℂ) (T : PAT H (Sp ℂ)) (m : SABMap T.t Xo.univ.t (FU.u i Pe)) (ε : ℤˣ) (h : (FU.Di i).La → (FU.Di i).La → OK H), PureType.IsThetaUnif u₀ → c ≠ 0 → PureType.IsQuadLattice u₀.Λ H.d c → 0 < ((ε : ℤ) : ℝ) * ω.im → IsKForm (FU.Di i) ε h → ∃ ψ : PureType.lat hP (2 * ω - 1) → (C.P.A ⟶ T.t.G), IsFam C T h _ ψ ∧ ∀ a (z : ℂ), (u₀.ex (fun _ => c * z) ≫ ψ a).left ≫ m.φ = (FU.ex i Pe (z • Pe a)).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmCurveHomFamilyToFibreAtPeriodStatement

theorem **_root_.GlobalTheory.T.exists_cmCurve_homFamily_to_fibre_at_period** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (ω : ℂ) (C : Ell H ℂ ω) (u₀ : HolUnif C.P.A C.P.L C.P.g) (c : ℂ) (T : PAT H (Sp ℂ)) (m : SABMap T.t Xo.univ.t (FU.u i Pe)) (ε : ℤˣ) (h : (FU.Di i).La → (FU.Di i).La → OK H), PureType.IsThetaUnif u₀ → c ≠ 0 → PureType.IsQuadLattice u₀.Λ H.d c → 0 < ((ε : ℤ) : ℝ) ∗ ω.im → IsKForm (FU.Di i) ε h → ∃ ψ : PureType.lat hP (2 ∗ ω − 1) → (C.P.A ⟶ T.t.G), IsFam C T h _ ψ ∧ ∀ a (z : ℂ), (u₀.ex (fun _ => c ∗ z) ≫ ψ a).left ≫ m.φ = (FU.ex i Pe (z • Pe a)).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmCurvesOfWeierstrassCurves

theorem **exists_periodPair** (h2 : Module.finrank ℚ K = 2) (σ : K →+* ℂ) (hσ : ¬ ComplexEmbedding.IsReal σ) (I : (FractionalIdeal (𝒪 K)⁰ K)ˣ) : ∃ L : PeriodPair, ∀ z, z ∈ L.lattice ↔ ∃ x ∈ I.1, z = σ x

theorem **exists_mul_of_invariants** {L L' : PeriodPair} {c₄ Δ : ℂ} (h : c₄ ^ 3 ∗ (L.g₂ ^ 3 − 27 ∗ L.g₃ ^ 2) = 1728 ∗ L.g₂ ^ 3 ∗ Δ) (h₄ : c₄ = 12 ∗ L'.g₂) (hΔ : Δ = L'.g₂ ^ 3 − 27 ∗ L'.g₃ ^ 2) : ∃ c : ℂ, c ≠ 0 ∧ ∀ z, z ∈ L'.lattice ↔ ∃ w ∈ L.lattice, z = c ∗ w

theorem **exists_field** (h2 : Module.finrank ℚ K = 2) (σ : K →+* ℂ) (hσ : ¬ ComplexEmbedding.IsReal σ) : ∃ M : IntermediateField ℚ ℂ, FiniteDimensional ℚ M ∧ (∀ a, σ a ∈ M) ∧ ∀ I : (FractionalIdeal (𝒪 K)⁰ K)ˣ, ∃ W : WeierstrassCurve (𝒪 M), IsUnit W.Δ ∧ ∀ L' : PeriodPair, ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M)) W.c₄ = 12 ∗ L'.g₂ → ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M)) W.Δ = L'.g₂ ^ 3 − 27 ∗ L'.g₃ ^ 2 → ∃ c : ℂ, c ≠ 0 ∧ ∀ z, z ∈ L'.lattice ↔ ∃ x ∈ I.1, z = c ∗ σ x

theorem **mem_inv_iff** (𝔟 : (Ideal (𝒪 K))⁰) (x : K) : x ∈ ((FractionalIdeal.mk0 K 𝔟)⁻¹).1 ↔ ∀ b ∈ (𝔟 : Ideal (𝒪 K)), IsIntegral ℤ (x ∗ (b : K))

theorem **exists_cm_curves_of_weierstrass_curves** : (∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (Lf : (R →+* ℂ) → PeriodPair), (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : R →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ (Lf s').lattice ↔ z • w' ∈ u'.Λ)) ∧ ∀ s : R →+* ℂ, s W.c₄ = 12 ∗ (Lf s).g₂ ∧ s W.c₆ = 216 ∗ (Lf s).g₃ ∧ s W.Δ = (Lf s).g₂ ^ 3 − 27 ∗ (Lf s).g₃ ^ 2) → ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ ComplexEmbedding.IsReal σ₀ →

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmEllipticCurveIntegersWithPeriod

theorem **isQuadLattice_of_linear** {g g' : ℕ} (hg : g = 1) (hg' : g' = 1) {Λ : Submodule ℤ (Fin g → ℂ)} {Λ' : Submodule ℤ (Fin g' → ℂ)} (hd : DiscreteTopology Λ) (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (hF : ∀ z, z ∈ Λ ↔ F z ∈ Λ') {d : ℕ} {c : ℂ} (hc : c ≠ 0) (h : PureType.IsQuadLattice Λ' d c) : ∃ c' : ℂ, c' ≠ 0 ∧ PureType.IsQuadLattice Λ d c'

theorem **isQuadLattice_of_isGrpPullback** (hU : ThetaUnifAV) (hL : Out.Laws) {R : Type} [CommRing R] (P : Out.PolAV (Sp R)) (hg : P.g = 1) (σ : R →+* ℂ) {d : ℕ} (h : ∀ (Q : Out.PolAV (Sp ℂ)) (φ : Q.A.left ⟶ P.A.left), IsPolFibre σ P Q φ → ∀ u : HolUnif Q.A Q.L Q.g, ∃ c : ℂ, c ≠ 0 ∧ PureType.IsQuadLattice u.Λ d c) (Q : Out.PolAV (Sp ℂ)) (hQ : Q.g = 1) (φ : Q.A.left ⟶ P.A.left) (hφ : IsGrpPullback Q.A P.A (spMap σ) φ) (u : HolUnif Q.A Q.L Q.g) : ∃ c : ℂ, c ≠ 0 ∧ PureType.IsQuadLattice u.Λ d c

theorem **exists_ell_pos** (hC : CMCurve) (hU : ThetaUnifAV) (hL : Out.Laws) (hX : HomExtDvr) {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] {χ : DirichletCharacter ℂ H.d} (hχ : IsKroneckerChar χ) : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (ω : 𝒪 M) (C : Ell H (𝒪 M) ω) (x : ℝ), ((ω : M) : ℂ) = (1 + Complex.I ∗ √H.d) / 2 ∧ C.IsPrincipalAt ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M)) ∧ C.HasPeriod x ∧ (x : ℂ) = (Real.eulerMascheroniConstant + Real.log (2 ∗ Real.pi / √H.d) : ℝ) − deriv χ.LFunction 1 / χ.LFunction 1

theorem **exists_ell** : CMCurve → ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → ∀ ρ : ℂ, ρ ^ 2 = −(H.d : ℂ) → ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (ω : 𝒪 M) (C : Ell H (𝒪 M) ω) (x : ℝ), 2 ∗ ((ω : M) : ℂ) − 1 = ρ ∧ C.IsPrincipalAt ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M)) ∧ C.HasPeriod x ∧ (x : ℂ) = (Real.eulerMascheroniConstant + Real.log (2 ∗ Real.pi / √H.d) : ℝ) − deriv χ.LFunction 1 / χ.LFunction 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmEllipticCurveIntegersWithPeriodStatement

theorem **_root_.GlobalTheory.T.exists_cmEllipticCurve_integers_with_period** : CMCurve → ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → ∀ ρ : ℂ, ρ ^ 2 = −(H.d : ℂ) → ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (ω : 𝒪 M) (C : Ell H (𝒪 M) ω) (x : ℝ), 2 ∗ ((ω : M) : ℂ) − 1 = ρ ∧ C.IsPrincipalAt ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M)) ∧ C.HasPeriod x ∧ (x : ℂ) = (Real.eulerMascheroniConstant + Real.log (2 ∗ Real.pi / √H.d) : ℝ) − deriv χ.LFunction 1 / χ.LFunction 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmPointLatticeDataOfAdaptedBasis

theorem **exists_pair** (H : Host E) {r : ℕ} (X : FD.DataX E r) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (hr : 0 < r) (h𝔄 : 𝔄 ≠ 0) (hΨ : ∀ τ : E →+* ℂ, τ ∈ X.Ψ ↔ τ (H.rt : E) = H.rt0) (hΛ : X.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ X.L', v = a • u}) : ∃ (D₁ : Bool → Datum H) (e : ∀ b, (D₁ b).La ≃ₗ[𝒪 E] (𝔄 : Submodule (𝒪 E) E)) (s : Bool → ℤ), ((D₁ true).p = 1 ∧ (D₁ true).q = 0) ∧ ((D₁ false).p = 0 ∧ (D₁ false).q = 1) ∧ (∀ b, s b = 1 ∨ s b = −1) ∧ ∀ b x y, (((D₁ b).EF x y : ℤ) : ℚ) = Algebra.trace ℚ E (s b ∗ X.ξ0 ∗ (e b x : E) ∗ FD.cc (e b y : E))

theorem **exists_nsmul_mem** (𝔄 : Submodule (𝒪 E) E) (h : 𝔄 ≠ ⊥) {J : Type} [Fintype J] (c : J → E) : ∃ n : ℕ, 0 < n ∧ ∀ i, (n : E) ∗ c i ∈ 𝔄

theorem **exists_cmLat** (H : Host E) {r p q : ℕ} (Y : FD.Data E r) (x₀ : Y.dom) (X : FD.DataX E r) (hX : X.toData = Y) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (hr : 0 < r) (h𝔄 : 𝔄 ≠ 0) (hrk : r = p + q) (hp0 : Y.p0 = q) (hΨ : ∀ τ : E →+* ℂ, τ ∈ Y.Ψ ↔ τ (H.rt : E) = H.rt0) (hΛ : X.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ X.L', v = a • u}) (D : Datum H) {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) (φ : D.La ≃ₗ[𝒪 E] Y.Λ) (isom : ∀ a b, ((D.EF a b : ℤ) : ℚ) = Algebra.trace ℚ E (−Y.ξ0 ∗ FD.herm Y.Gm (φ a : Fin r → E) (φ b))) (ix : Fin g → ℕ) (hix : ∀ j, ty j (H.rt : E) ≠ H.rt0 → ix j < q) (pm : ∀ a j, Pe a j = perFr Y x₀.1 (φ a : Fin r → E) (ty j) (ix j)) (La : ℝ) (lam : Fin r → Fin r → E) (hl : FD.LamBasis X La (hX ▸ x₀.1) lam) : ∃ Z : CMLat hP p q, ∀ j (τ : E →+* ℂ), ‖τ (Z.hh j : E)‖ ≤ La ^ 2

theorem **exists_cmLat_unfAt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : Unf Xo c c.F.val.toRingHom r), IsUnf U → 0 < r → ∀ (k : U.Comp) (x₀ : (U.X k).dom) (W : PullbackSections.UnfAt U k x₀) (Xx : FD.DataX E r) (hX : Xx.toData = U.X k) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E), 𝔄 ≠ 0 →

Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ Xx.L', v = a • u} → ∀ (La : ℝ) (lam : Fin r → Fin r → E), FD.LamBasis Xx La (hX ▸ x₀.1) lam → ∃ Z : CMLat W.per Dw.p Dw.q, ∀ j (τ : E →+* ℂ), ‖τ (Z.hh j : E)‖ ≤ La ^ 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmPointLatticeDataOfAdaptedBasisStatement

theorem **_root_.GlobalTheory.T.exists_cmPoint_lattice_data_of_adapted_basis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : Unf Xo c c.F.val.toRingHom r), IsUnf U → 0 < r → ∀ (k : U.Comp) (x₀ : (U.X k).dom) (W : PullbackSections.UnfAt U k x₀) (Xx : FD.DataX E r) (hX : Xx.toData = U.X k) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E), 𝔄 ≠ 0 → Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ Xx.L', v = a • u} → ∀ (La : ℝ) (lam : Fin r → Fin r → E), FD.LamBasis Xx La (hX ▸ x₀.1) lam → ∃ Z : CMLat W.per Dw.p Dw.q, ∀ j (τ : E →+* ℂ), ‖τ (Z.hh j : E)‖ ≤ La ^ 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmPointLocalModelOverDvr

private theorem **spectrumMap_comp** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap (g.comp f) = spMap g ≫ spMap f

private theorem **sab_comp_exists** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {S S₁ S₂ : Scheme.{0}} {t : SAB H S} {t₁ : SAB H S₁} {t₂ : SAB H S₂} {f₁ : S₁ ⟶ S} {g : S₂ ⟶ S₁} (n : SABMap t₂ t₁ g) (m₁ : SABMap t₁ t f₁) (f₂ : S₂ ⟶ S) (h : f₂ = g ≫ f₁) : ∃ m : SABMap t₂ t f₂, m.φ = n.φ ≫ m₁.φ

private theorem **sab_cancel_exists** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {S S₁ S₂ : Scheme.{0}} {t : SAB H S} {t₁ : SAB H S₁} {t₂ : SAB H S₂} {f₁ : S₁ ⟶ S} {f₂ : S₂ ⟶ S} (m₁ : SABMap t₁ t f₁) (m₂ : SABMap t₂ t f₂) (g : S₂ ⟶ S₁) (h : f₂ = g ≫ f₁) : ∃ n : SABMap t₂ t₁ g, n.φ ≫ m₁.φ = m₂.φ

theorem **locMod_exists** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (T : CDVR) (iK : z.K →+* T.Fr), ((∀ a : 𝒪 z.K, iK (a : z.K) ∈ Set.range (algebraMap T.V T.Fr)) → Nonempty (LocMod z T iK)) ∧ ∀ (M : LocMod z T iK) (T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr), (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr) → ∃ (M' : LocMod z T' (i.comp iK)) (n : SABMap M'.G.t M.G.t (spMap ψ)) (n' : SABMap M'.t'.t M.t'.t (spMap i)), M'.m1.φ = n'.φ ≫ M.m1.φ ∧ M'.m2.φ ≫ n.φ = n'.φ ≫ M.m2.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmPointLocalModelOverDvrStatement

theorem **_root_.GlobalTheory.T.exists_cmPoint_localModel_over_dvr** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (T : CDVR) (iK : z.K →+* T.Fr), ((∀ a : 𝒪 z.K, iK (a : z.K) ∈ Set.range (algebraMap T.V T.Fr)) → Nonempty (LocMod z T iK)) ∧ ∀ (M : LocMod z T iK) (T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr), (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr) → ∃ (M' : LocMod z T' (i.comp iK)) (n : SABMap M'.G.t M.G.t (spMap ψ)) (n' : SABMap M'.t'.t M.t'.t (spMap i)), M'.m1.φ = n'.φ ≫ M.m1.φ ∧ M'.m2.φ ≫ n.φ = n'.φ ≫ M.m2.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmTensorLatticeCurvePrincipalTriple

theorem **exists_isTen** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsNoetherianRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → ∃ T : PAT H (Sp R), Nonempty (IsTen C D h T) ∧ T.g = Module.finrank ℚ E / 2 * (D.p + D.q)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCmTensorLatticeCurvePrincipalTripleStatement

theorem **_root_.GlobalTheory.T.exists_cm_tensor_lattice_curve_principal_triple** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsNoetherianRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → ∃ T : PAT H (Sp R), Nonempty (IsTen C D h T) ∧ T.g = Module.finrank ℚ E / 2 * (D.p + D.q)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCommGrpObjWeierstrassScheme

theorem **exists_grpObj_weierstrass** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ _ : GrpObj (schemeOver W), IsCommMonObj (schemeOver W) ∧ (η[schemeOver W].left = zero W ∧ (∀ (A : Type) [CommRing A] (ρ : R →+* A) (P Q T : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) (hQ : IsUnimodularSol (W.map ρ) Q) (hT : IsUnimodularSol (W.map ρ) T), (Proportional T ((W.map ρ).toProjective.addXYZ P Q) ∧ Proportional T (diagXYZ (W.map ρ).toProjective P Q)) → ∀ x y : Over.mk (spMap ρ) ⟶ schemeOver W, x.left = ptHom W ρ P hP → y.left = ptHom W ρ Q hQ → (lift x y ≫ μ[schemeOver W]).left = ptHom W ρ T hT) ∧ (∀ (A : Type) [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P), ∃ hN : IsUnimodularSol (W.map ρ) ((W.map ρ).toProjective.neg P), ptHom W ρ P hP ≫ (GrpObj.inv : schemeOver W ⟶ schemeOver W).left = ptHom W ρ _ hN))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCommGrpObjWeierstrassSchemeStatement

theorem **_root_.GlobalTheory.T.exists_commGrpObj_weierstrassScheme** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ _ : GrpObj (schemeOver W), IsCommMonObj (schemeOver W) ∧ (η[schemeOver W].left = zero W ∧ (∀ (A : Type) [CommRing A] (ρ : R →+* A) (P Q T : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) (hQ : IsUnimodularSol (W.map ρ) Q) (hT : IsUnimodularSol (W.map ρ) T), (Proportional T ((W.map ρ).toProjective.addXYZ P Q) ∧ Proportional T (diagXYZ (W.map ρ).toProjective P Q)) → ∀ x y : Over.mk (spMap ρ) ⟶ schemeOver W, x.left = ptHom W ρ P hP → y.left = ptHom W ρ Q hQ → (lift x y ≫ μ[schemeOver W]).left = ptHom W ρ T hT) ∧ (∀ (A : Type) [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P), ∃ hN : IsUnimodularSol (W.map ρ) ((W.map ρ).toProjective.neg P), ptHom W ρ P hP ≫ (GrpObj.inv : schemeOver W ⟶ schemeOver W).left = ptHom W ρ _ hN))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCommonDenominatorOfValuationLeOne

theorem **exists_common_denominator_of_valuation_le_one** {K : Type} [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)) {n : ℕ} (a : Fin n → K) (ha : ∀ m, p.valuation K (a m) ≤ 1) : ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ ∀ m, ∃ b : 𝒪 K, s • a m = algebraMap (𝒪 K) K b

theorem **IsModelAt.rows_span_lieLat** {K : Type} [Field K] [NumberField K] {A : Out.PolAV (Sp (𝒪 K))} {x : KeyIn K} {φ : x.P.A.left ⟶ A.A.left} (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) {p : HeightOneSpectrum (𝒪 K)} {M : Matrix (Fin x.P.g) (Fin x.P.g) K} (hM : IsModelAt A x φ p M) : (∀ i, ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • M i ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).lieLat) ∧ ∀ c ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).lieLat, ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • c ∈ Submodule.span (𝒪 K) (Set.range fun i => M i)

theorem **exists_keyLines** : Out.SecTwistSt → ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left), IsFibreOf A x (gmap K) φ → ∃ T abelian₂ : MLine K, T.Good ∧ abelian₂.Good ∧ (∀ p, KeyLocAt x T abelian₂ p) ∧ (∀ σ, KeyArchAt x T abelian₂ σ) ∧ ∀ p, ∃ M, IsModelAt A x φ p M ∧ IsLocGen p M.det abelian₂.I

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCompleteDVRCompletionIntegralIff

private lemma **mem_range_algebraMap_iff_valV_le_one** (T : CDVR) (x : T.Fr) : x ∈ Set.range (algebraMap T.V T.Fr) ↔ valV T.V x ≤ 1

private lemma **withZero_le_one_iff_log_nonpos** {x : WithZero (Multiplicative ℤ)} (hx : x ≠ 0) : x ≤ 1 ↔ WithZero.log x ≤ 0

theorem **exists_cdvr_completion_integral_iff** : ∀ (K : Type) [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)), ∃ (T : CDVR) (iK : K →+* T.Fr), T.FinRes ∧ ∀ y : K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation K y ≤ 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsCompleteDVRCompletionIntegralIffStatement

theorem **_root_.GlobalTheory.T.exists_completeDVR_completion_integral_iff** : ∀ (K : Type) [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)), ∃ (T : CDVR) (iK : K →+* T.Fr), T.FinRes ∧ ∀ y : K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation K y ≤ 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsConjAction

theorem **bar_mem** (H : Host E) (a : OK H) : H.bar (a : 𝒪 E) ∈ OK H

theorem **barK_wK_add** (H : Host E) : barK H (wK H) + wK H = 1

theorem **isLieComb_one_sub** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] [IsCommMonObj G] (v : epsR R ⟶ G) (hv : IsTangent v) (ω : R) (f g : G ⟶ G) (hfg : g ∗ f = 𝟙 G) (h : IsLieComb ![ω] ![v] (v ≫ f)) : IsLieComb ![1 - ω] ![v] (v ≫ g)

theorem **lbIso_of_adj** {S : Scheme.{0}} {B : Over S} [GrpObj B] [IsCommMonObj B] {M : LB B.left} (hM : M.IsLB) {f g : B ⟶ B} (h : LamEq M f (𝟙 B) M (𝟙 B) g) : LBIso (cubeLB B ![𝟙 B, g] M) (cubeLB B ![𝟙 B, f] M)

theorem **exists_conj_action** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) : ∃ a : OAct (OK H) C.P.A, (∀ x y : OK H, (y : 𝒪 E) = H.bar x → a.act x = C.act.act y) ∧ IsLieComb ![1 - ω] ![C.v] (C.v ≫ a.act (wK H)) ∧ (∀ x y : OK H, (y : 𝒪 E) = H.bar x → LamEq C.P.L (a.act x) (𝟙 C.P.A) C.P.L (𝟙 C.P.A) (a.act y)) ∧ LBIso C.P.L (cubeLB C.P.A ![𝟙 C.P.A, a.act (wK H)] C.P.L)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsConjDatum

theorem **bar_bar** (H : Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

theorem **emb_bar** (H : Host E) (τ : E →+* ℂ) (a : 𝒪 E) : τ ((H.bar a : 𝒪 E) : E) = starRingEnd ℂ (τ (a : E))

theorem **toLat_smul** {H : Host E} (D : Datum H) (s : 𝒪 E) (a : ConjLat D) : toLat D (s • a) = H.bar s • toLat D a

theorem **finite** {H : Host E} (D : Datum H) : Module.Finite (𝒪 E) (ConjLat D)

theorem **projective** {H : Host E} (D : Datum H) : Module.Projective (𝒪 E) (ConjLat D)

theorem **finrank_eq** {H : Host E} (D : Datum H) : Module.finrank (𝒪 E) (ConjLat D) = Module.finrank (𝒪 E) D.La

theorem **exists_conjDatum** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H), ∃ (D' : Datum H) (e : D'.La ≃+ D.La), D'.p = D.q ∧ D'.q = D.p ∧ (∀ (s : 𝒪 E) a, e (s • a) = H.bar s • e a) ∧ ∀ a b, D'.EF a b = D.EF (e a) (e b)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsConstOfSqMulTendstoZero

private lemma **tendsto_sq_mul_weierstrassP'** (L : PeriodPair) : Tendsto (fun z ↦ z ^ 2 * ℘[L] z) (𝓝[≠] 0) (𝓝 1)

private lemma **exists_tendsto_pow_mul_of_tendsto_pow_succ_mul_zero** {h : ℂ → ℂ} (k : ℕ) (hd : ∀ᶠ z in 𝓝[≠] (0:ℂ), DifferentiableAt ℂ h z) (h2 : Tendsto (fun z ↦ z ^ (k + 1) * h z) (𝓝[≠] 0) (𝓝 0)) : ∃ r, Tendsto (fun z ↦ z ^ k * h z) (𝓝[≠] 0) (𝓝 r)

private lemma **exists_const_of_periodic_of_mul_tendsto_zero** (L : PeriodPair) {F : ℂ → ℂ} (hd : ∀ z ∉ L.lattice, DifferentiableAt ℂ F z) (hp : ∀ z, ∀ l ∈ L.lattice, F (z + l) = F z) (h : Tendsto (fun z ↦ z * F z) (𝓝[≠] 0) (𝓝 0)) : ∃ k, ∀ z ∉ L.lattice, F z = k

private lemma **tendsto_sq_mul_weierstrassP_sub_one_div** (L : PeriodPair) : Tendsto (fun z ↦ (z ^ 2 * ℘[L] z - 1) / z) (𝓝[≠] 0) (𝓝 0)

private lemma **half_sub_half_notMem_lattice** (L : PeriodPair) : L.ω₁ / 2 - L.ω₂ / 2 ∉ L.lattice

private lemma **half_add_half_notMem_lattice** (L : PeriodPair) : L.ω₁ / 2 + L.ω₂ / 2 ∉ L.lattice

lemma **_root_.GlobalTheory.PeriodicWeierstrass.exists_eq_of_sq_mul_tendsto** (L : PeriodPair) {g : ℂ → ℂ} {a : ℂ} (hd : ∀ z ∉ L.lattice, DifferentiableAt ℂ g z) (hp : ∀ z, ∀ l ∈ L.lattice, g (z + l) = g z) (h : Tendsto (fun z ↦ z ^ 2 * g z) (𝓝[≠] 0) (𝓝 a)) : ∃ k, ∀ z ∉ L.lattice, g z = a * ℘[L] z + k

private lemma **tendsto_pow_three_mul_derivWeierstrassP** (L : PeriodPair) : Tendsto (fun z ↦ z ^ 3 * ℘'[L] z) (𝓝[≠] 0) (𝓝 (-2))

lemma **_root_.GlobalTheory.PeriodicWeierstrass.exists_eq_of_cube_mul_tendsto** (L : PeriodPair) {g : ℂ → ℂ} (hd : ∀ z ∉ L.lattice, DifferentiableAt ℂ g z) (hp : ∀ z, ∀ l ∈ L.lattice, g (z + l) = g z) (h : Tendsto (fun z ↦ z ^ 3 * g z) (𝓝[≠] 0) (𝓝 (-2))) : ∃ b k, ∀ z ∉ L.lattice, g z = ℘'[L] z + b * ℘[L] z + k

lemma **_root_.GlobalTheory.PeriodicWeierstrass.tendsto_leading** (W : WeierstrassCurve ℂ) {x y t : ℂ → ℂ} {c : ℂ} (hc : c ≠ 0) (hE : ∀ᶠ z in 𝓝[≠] 0, W.toAffine.Equation (x z) (y z)) (ht : AnalyticAt ℂ t 0) (ht0 : t 0 = 0) (ht1 : deriv t 0 = c) (hty : ∀ᶠ z in 𝓝[≠] 0, t z ∗ y z = -x z) (hy : Tendsto (fun z ↦ ‖y z‖) (𝓝[≠] 0) atTop) : Tendsto (fun z ↦ z ^ 2 * (c ^ 2 * x z)) (𝓝[≠] 0) (𝓝 1) ∧ Tendsto (fun z ↦ z ^ 3 * (c ^ 3 * y z)) (𝓝[≠] 0) (𝓝 (-1))

theorem **_root_.GlobalTheory.invariants_eq_of_periodic_solution** : ∀ (L : PeriodPair) (W : WeierstrassCurve ℂ) (x y t : ℂ → ℂ) (c : ℂ), c ≠ 0 → DifferentiableOn ℂ x (L.lattice : Set ℂ)ᶜ → DifferentiableOn ℂ y (L.lattice : Set ℂ)ᶜ → (∀ l ∈ L.lattice, ∀ z, x (z + l) = x z ∧ y (z + l) = y z) → (∀ z, z ∉ L.lattice → W.toAffine.Equation (x z) (y z)) → AnalyticAt ℂ t 0 → t 0 = 0 → deriv t 0 = c → (∀ᶠ z in nhdsWithin 0 {0}ᶜ, t z ∗ y z = -x z) → Filter.Tendsto (fun z => ‖y z‖) (nhdsWithin 0 {0}ᶜ) Filter.atTop → c ^ 4 * W.c₄ = 12 * L.g₂ ∧ c ^ 6 * W.c₆ = 216 * L.g₃ ∧ c ^ 12 * W.Δ = L.g₂ ^ 3 - 27 * L.g₃ ^ 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsConstantCmEvaluationFitsRoom

theorem **room_neg_log_epStar** (f d L M k0 S : ℝ) (hf : 1 ≤ f) (hd : 1 ≤ d) (hL : 0 ≤ L) (hM : 0 < M) (hk : 0 < k0) (hS : 1 ≤ S) (h : Real.log (M * k0) ≤ 19 * L + 3 + Real.log d) : 0 < epStar M f S k0 ∧ epStar M f S k0 ≤ 1 ∧ -Real.log (epStar M f S k0) ≤ 19 * L + 6 + Real.log d + f + Real.log S

theorem **room_log_radius** (f r : ℕ) (d e S : ℝ) (hf : 1 ≤ f) (hr : 1 ≤ r) (hd : 1 ≤ d) (he : 0 < e) (hS : 1 ≤ S) : 0 < 2 * ReciprocityTheory.MultiplicativeHeights.approxR r (d5mu f r d (Real.sqrt e / Real.sqrt f)) (3 * frameRadius S r / 5) ∧ Real.log (2 * ReciprocityTheory.MultiplicativeHeights.approxR r (d5mu f r d (Real.sqrt e / Real.sqrt f)) (3 * frameRadius S r / 5)) = Real.log (1600 / 3) + Real.log r + (1 / 2) * Real.log f + (1 / 2) * Real.log S + 3 * Real.log ((f : ℝ) * r) + Real.log d - (1 / 2) * Real.log e

theorem **room_of_constants** : ∃ C0 : ℝ, 0 < C0 ∧ ∀ (f r p0 : ℕ) (d L Om M k0 S : ℝ), 1 ≤ f → 1 ≤ r → p0 ≤ r → 1 ≤ d → 0 ≤ L → 0 < M → 0 < k0 → 1 ≤ S → Real.log (M * k0) ≤ 19 * L + 3 + Real.log d → 1 ≤ frameBasisRadius r p0 f d (Real.sqrt (epStar M f S k0)) (3 * frameRadius S r / 5) ∧ ∀ s t : ℝ, 0 ≤ s → 0 ≤ t → s + t ≤ S → ∀ C A : ℝ, C0 ≤ C → 2 ≤ A → (Formalism.L2.LamCM.pDeg Om + (1 / 2) * Real.log (frameBasisRadius r p0 f d (Real.sqrt (epStar M f S k0)) (3 * frameRadius S r / 5))) * (s + t) + s * Real.log ((r : ℝ) *

frameBasisRadius r p0 f d (Real.sqrt (epStar M f S k0)) (3 * frameRadius S r / 5)) ≤ C * ((f : ℝ) * r + L + Real.log d + |Om|) ^ A * (1 + Real.log S) * S

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsConstantCmEvaluationFitsRoomStatement

theorem **_root_.GlobalTheory.T.exists_constant_cmEvaluation_fits_room** : ∃ C0 : ℝ, 0 < C0 ∧ ∀ (f r p0 : ℕ) (d L Om M k0 S : ℝ), 1 ≤ f → 1 ≤ r → p0 ≤ r → 1 ≤ d → 0 ≤ L → 0 < M → 0 < k0 → 1 ≤ S → Real.log (M * k0) ≤ 19 * L + 3 + Real.log d → 1 ≤ frameBasisRadius r p0 f d (Real.sqrt (epStar M f S k0)) (3 * frameRadius S r / 5) ∧ ∀ s t : ℝ, 0 ≤ s → 0 ≤ t → s + t ≤ S → ∀ C A : ℝ, C0 ≤ C → 2 ≤ A → (Formalism.L2.LamCM.pDeg Om + (1 / 2) * Real.log (frameBasisRadius r p0 f d (Real.sqrt (epStar M f S k0)) (3 * frameRadius S r / 5))) * (s + t) + s * Real.log ((r : ℝ) * frameBasisRadius r p0 f d (Real.sqrt (epStar M f S k0)) (3 * frameRadius S r / 5)) ≤ C * ((f : ℝ) * r + L + Real.log d + |Om|) ^ A * (1 + Real.log S) * S

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsDetIdealLocalGenFullLattice

theorem **det_of_smul_rows** {n : ℕ} (s : Fin n → 𝒪 K) (x : Fin n → Fin n → K) : (Matrix.of fun i => s i • x i).det = (∏ i, s i) • (Matrix.of x).det

theorem **prod_notMem_asIdeal** {n : ℕ} (p : HeightOneSpectrum (𝒪 K)) (s : Fin n → 𝒪 K) (hs : ∀ i, s i ∉ p.asIdeal) : ∏ i, s i ∉ p.asIdeal

theorem **valuation_smul_of_notMem** (p : HeightOneSpectrum (𝒪 K)) {s : 𝒪 K} (hs : s ∉ p.asIdeal) (c : K) : p.valuation K (s • c) = p.valuation K c

theorem **valuation_smul_le** (p : HeightOneSpectrum (𝒪 K)) (s : 𝒪 K) (c : K) : p.valuation K (s • c) ≤ p.valuation K c

theorem **exists_detIdeal_isLocGen** : ∀ (K : Type) [Field K] [NumberField K] (n : ℕ) (Λ : Submodule (𝒪 K) (Fin n → K)), Λ.FG → Submodule.span K (Λ : Set (Fin n → K)) = ⊤ → ∃ I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K, I ≠ 0 ∧ ∀ (p : HeightOneSpectrum (𝒪 K)) (M : Matrix (Fin n) (Fin n) K), (∀ i, ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • M i ∈ Λ) → (∀ c ∈ Λ, ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • c ∈ Submodule.span (𝒪 K) (Set.range fun i => M i)) → IsLocGen p M.det I

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsDetIdealLocalGenFullLatticeStatement

theorem **_root_.GlobalTheory.T.exists_detIdeal_localGen_full_lattice** : ∀ (K : Type) [Field K] [NumberField K] (n : ℕ) (Λ : Submodule (𝒪 K) (Fin n → K)), Λ.FG → Submodule.span K (Λ : Set (Fin n → K)) = ⊤ → ∃ I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K, I ≠ 0 ∧ ∀ (p : HeightOneSpectrum (𝒪 K)) (M : Matrix (Fin n) (Fin n) K), (∀ i, ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • M i ∈ Λ) → (∀ c ∈ Λ, ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • c ∈ Submodule.span (𝒪 K) (Set.range fun i => M i)) → IsLocGen p M.det I

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsDiagonalCoordinates

theorem **mem_of_countable** {V : Type*} [AddCommGroup V] [Module ℂ V] {Λ : Set V} (hΛ : Λ.Countable) (W : Submodule ℂ V) {v : V} (h : ∀ z : ℂ, ∃ l ∈ Λ, z • v - l ∈ W) : v ∈ W

theorem **linear_eq_of_countable** {U V : Type*} [AddCommGroup U] [Module ℂ U] [AddCommGroup V] [Module ℂ V] {Λ : Set V} (hΛ : Λ.Countable) {F F' : U →ₗ[ℂ] V} (h : ∀ u, F u - F' u ∈ Λ) : F = F'

theorem **exists_lift** {V : Type} [AddCommGroup V] [Module ℂ V] (f : M →+ V) (hf : ∀ (o : O) a, f (o • a) = algebraMap O ℂ o • f a) : ∃ F : ℂ ⊗[O] M →ₗ[ℂ] V, ∀ z a, F (z ⊗ₜ a) = z • f a

theorem **tensor_unif** {X Y V : Type} [AddCommGroup X] [Module O X] [AddCommGroup Y] [Module O Y] [AddCommGroup V] [Module ℂ V] (eqv : M ⊗[O] X ≃ₗ[O] Y) (e₀ : ℂ →ₗ[O] X) (hs : Function.Surjective e₀) (hk : ∀ z, e₀ z = 0 ↔ z ∈ Set.range (algebraMap O ℂ)) (ex : V →+ Y) (hΛ : Set.Countable {v | ex v = 0}) (P : M →+ V) (hP : ∀ (o : O) a, P (o • a) = algebraMap O ℂ o • P a) (hi : Function.Injective P) (hex : ∀ (z : ℂ) a, ex (z • P a) = eqv (a ⊗ₜ e₀ z)) : Submodule.span ℂ (Set.range P) = ⊤ ∧ (∀ v, ex v = 0 ↔ v ∈ P.range) ∧ ∀ (V' : Type) [AddCommGroup V'] [Module ℂ V'] (f : M →+ V'), (∀ (o : O) a, f (o • a) = algebraMap O ℂ o • f a) → ∃ F : V →ₗ[ℂ] V', ∀ a, F (P a) = f a

theorem **exists_diagonal_coordinates** {V : Type} [AddCommGroup V] [Module ℂ V] [Module.Finite ℂ V] (r : 𝒪 E →+* Module.End ℂ V) : ∃ (g : ℕ) (B : V ≃ₗ[ℂ] (Fin g → ℂ)) (ty : Fin g → (E →+* ℂ)), ∀ (a : 𝒪 E) v j, B (r a v) j = ty j (a : E) * B v j

theorem **card_type** {M : Type} [AddCommGroup M] [Module (𝒪 E) M] [Module.Finite (𝒪 E) M] {g : ℕ} {ty : Fin g → (E →+* ℂ)} {P : M →+ (Fin g → ℂ)} (hP : ∀ (s : 𝒪 E) a i, P (s • a) i = ty i (s : E) * P a i) (hs : Submodule.span ℂ (Set.range P) = ⊤) {w : 𝒪 E} {ω : ℂ} (hw : ∀ a, P (w • a) = ω • P a) (hu : ∀ f : M →+ ℂ, (∀ a, f (w • a) = ω • f a) → ∃ F : (Fin g → ℂ) →ₗ[ℂ] ℂ, ∀ a, F (P a) = f a) (τ : E →+* ℂ) : (τ w = ω → Nat.card {i // ty i = τ} = Module.finrank (𝒪 E) M) ∧ (τ w ≠ ω → Nat.card {i // ty i = τ} = 0)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsDvrModelLieBasisAtPrime

theorem **exists_isModelAt** : Out.SecTwistSt → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left), IsFibreOf A x (gmap K) φ → ∀ p : HeightOneSpectrum (𝒪 K), ∃ M, IsModelAt A x φ p M

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsDvrModelLieBasisAtPrimeStatement

theorem **_root_.GlobalTheory.T.exists_dvrModel_lieBasis_at_prime** : Out.SecTwistSt → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left), IsFibreOf A x (gmap K) φ → ∀ p : HeightOneSpectrum (𝒪 K), ∃ M, IsModelAt A x φ p M

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsEllipticCurvePolarisedLatticeFamily

theorem **exists_polAV_latticeFamily** : ThetaUnifAV → Out.PolBaseChangeSt → (∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (Lf : (R →+* ℂ) → PeriodPair), (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : R →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ (Lf s').lattice ↔ z • w' ∈ u'.Λ)) ∧ ∀ s : R →+* ℂ, s W.c₄ = 12 * (Lf s).g₂ ∧ s W.c₆ = 216 * (Lf s).g₃ ∧ s W.Δ = (Lf s).g₂ ^ 3 - 27 * (Lf s).g₃ ^ 2)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsEllipticCurvePolarisedLatticeFamilyStatement

theorem **_root_.GlobalTheory.T.exists_ellipticCurve_polarised_lattice_family** : ThetaUnifAV → Out.PolBaseChangeSt → (∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (Lf : (R →+* ℂ) → PeriodPair), (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : R →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' →

u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ (Lf s').lattice ↔ z • w' ∈ u'.Λ)) ∧ ∀ s : R →+* ℂ, s W.c₄ = 12 * (Lf s).g₂ ∧ s W.c₆ = 216 * (Lf s).g₃ ∧ s W.Δ = (Lf s).g₂ ^ 3 - 27 * (Lf s).g₃ ^ 2)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsEqNsmul

theorem **section_ext** {K k : Type} [Field K] [Field k] (t : Sp k ⟶ Sp K) {G : Over (Sp K)} {s s' : 𝟙_ (Over (Sp K)) ⟶ G} (h : toUnit (Over.mk t) ≫ s = toUnit (Over.mk t) ≫ s') : s = s'

theorem **ev_add_smul** {D : Datum H} {k : Type} [Field k] {Q : Type} [Group Q] {act : 𝒪 E → Q → Q} {e : ℕ → Q → Q → kˣ → Prop} (I : TateIso D k Q act e) {n : ℕ} (hn : 0 < n) (a c : D.La) : I.ev n (a + n • c) = I.ev n a

theorem **exists_eq_nsmul** (D : Datum H) (l : Level H) {J J' : Ideal (𝒪 E)} (hJ : J * J' = Ideal.span {(l.M : 𝒪 E)}) {e : 𝒪 E} (he : e ∈ J) {a : D.La} (ha : a ∈ J' • (⊤ : Submodule (𝒪 E) D.La)) : ∃ c, e • a = l.M • c

theorem **exists_obj** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (B : Type) [CommRing B] (b : Base l B) (K : Type) [Field K] [NumberField K] (j : B →+* K) (σ : K →+* ℂ) (T : PAT H (Sp K)) (I : TateIso D ℂ (Over.mk (spMap σ) ⟶ T.t.G) (fun a x => x ≫ T.t.act.act a) (fun n x x' c => Out.WeilIs T.t.G T.t.PP.L n x x' (unitOn c))), HasType H T.t.act (j b.ρ) D.p D.q → (I.ζ l.A : ℂ) = σ (j b.ζ) → (∀ x : Over.mk (spMap σ) ⟶ T.t.G, x ^ l.M' = 1 → ∃ s : 𝟙_ (Over (Sp K)) ⟶ T.t.G, toUnit _ ≫ s = x) → ∃ o : Obj D l b (spMap j), o.t = T.t

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsEqProdBlockDetPolyPowMul

theorem **evalFr_map_blockDetPoly** {K S : Type} [CommRing K] [CommRing S] (φ : K →+* S) {B : Type} [Fintype B] (m : B → ℕ) (b : B) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) S) : evalFr (MvPolynomial.map φ (blockDetPoly K m b)) X = (X b).det

theorem **evalFr_blockDetPoly** {K : Type} [CommRing K] {B : Type} [Fintype B] (m : B → ℕ) (b : B) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) : evalFr (blockDetPoly K m b) X = (X b).det

theorem **exists_eq_prod_blockDetPoly_pow_mul** (K : Type) [Field K] [Infinite K] (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (a : ∀ b, Fin (m b) → ℕ) (s : ℕ) : ∀ (n : B → ℕ), ∑ b, n b = s → ∀ (P : MvPolynomial (FrVar m) K), (∀ (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) (b : B) (t : Matrix (Fin (m b)) (Fin (m b)) K), (∀ b', (X b').det ≠ 0) → t.det ≠ 0 → t.BlockTriangular id → evalFr P (Function.update X b (X b * t)) = (∏ i, t i i ^ (a b i + n b)) * evalFr P X) → ∃ G : MvPolynomial (FrVar m) K, P = (∏ b, blockDetPoly K m b ^ n b) * G

theorem **evalFr_eq_prod_det_pow_mul** : ∀ (K : Type) [Field K] [Infinite K] (B : Type) [Fintype B] [DecidableEq B] (m : B → ℕ) (P : MvPolynomial (FrVar m) K) (a : ∀ b, Fin (m b) → ℕ) (N : ℕ), (∀ (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) K) (b : B) (t : Matrix (Fin (m b)) (Fin (m b)) K), (∀ b', (X b').det ≠ 0) → t.det ≠ 0 → t.BlockTriangular id → evalFr P (Function.update X b (X b * t)) = (∏ i, t i i ^ (a b i + N)) * evalFr P X) → ∃ G : MvPolynomial (FrVar m) K, ∀ (S : Type) [CommRing S] (φ : K →+* S) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) S), evalFr (MvPolynomial.map φ P) X = (∏ b, (X b).det) ^ N * evalFr (MvPolynomial.map φ G) X

theorem **exists_genericRing** (F K : Type) [Field F] [Field K] [Algebra F K] (B : Type) [Fintype B] (m : B → ℕ) : ∃ (A : Type) (_ : CommRing A) (_ : Algebra F A) (π : K →ₐ[F] A) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) A), _root_.IsReduced A ∧ (∀ b, IsUnit (X b).det) ∧ ∀ t : A, ∃ (P : MvPolynomial (FrVar m) K) (N : ℕ), ∀ (S : Type) [CommRing S] [Algebra F S] (φ : K →ₐ[F] S) (Y : ∀ b, Matrix (Fin (m b)) (Fin (m b)) S), (∀ b, IsUnit (Y b).det) → ∃ k : A →ₐ[F] S, k.comp π = φ ∧ (∀ b, (X b).map k = Y b) ∧ (∏ b, (Y b).det) ^ N * k t = evalFr (MvPolynomial.map φ.toRingHom P) Y

theorem **evR_frChg_mul_triangular** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (F : Rul Xo c) (hF : F ∈ sec Xo c a n) (z z' : FrPt Xo c K) (T : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) K) (hT : ∀ b, (T b).BlockTriangular id) (hd : ∀ b, (T b).det ≠ 0) (h : FrChg (fun b => (T b).transpose) 1 z z') : evR Xo c F z' = (∏ b, ∏ i, T b i i ^ a (b, i)) * evR Xo c F z

theorem **exists_poly_div_det_field** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∃ (P : MvPolynomial (FrVar Dw.mult) K) (N : ℕ), ∀ (S : Type) [CommRing S] [Algebra c.F S] (φ : K →ₐ[c.F] S) (z z' : FrPt Xo c S) (Y : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) S)), (∀ b, IsUnit (Y b).det) → FrMap φ y z → FrChg (fun b => (Y b).transpose) 1 z z' → (∏ b, (Y b).det) ^ N * evR Xo c F z' = evalFr (MvPolynomial.map φ.toRingHom P) Y

theorem **exists_isFrPoly_field** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∃ G : MvPolynomial (FrVar Dw.mult) K, IsFrPoly F y G

theorem **coeff_aeval_scale** {K σ : Type} [CommRing K] (t : K) (G : MvPolynomial σ K) (m : σ →₀ ℕ) : MvPolynomial.coeff m (MvPolynomial.aeval (fun v => MvPolynomial.C t * MvPolynomial.X v : σ → MvPolynomial σ K) G) = t ^ m.degree * MvPolynomial.coeff m G

theorem **isHomogeneous_of_scaling** {K σ : Type} [Field K] [CharZero K] (G D : MvPolynomial σ K) (hD : D ≠ 0) (d : ℕ) (h : ∀ x : σ → K, MvPolynomial.eval x D ≠ 0 → MvPolynomial.eval (fun v => 2 * x v) G = 2 ^ d * MvPolynomial.eval x G) : G.IsHomogeneous d

theorem **exists_isFrPoly_isHomogeneous** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∃ G : MvPolynomial (FrVar Dw.mult) K, GlobalTheory.LamT.IsFrPoly F y G ∧ G.IsHomogeneous (∑ τ, ∑ i : Fin (Dw.mult τ), a (τ, (i : ℕ)))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsEqPtHomOfIsLocalRing

theorem **exists_eq_ptHom_of_isLocalRing** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R) (A : Type) [CommRing A] [IsLocalRing A] (ρ : R →+* A) (x : Sp A ⟶ scheme W), x ≫ toSpec W = spMap ρ → ∃ (P : Fin 3 → A) (h : IsUnimodularSol (W.map ρ) P), x = ptHom W ρ P h

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsEqSmulOfFinrankOne

theorem **exists_span_smul_mem** {r : ℕ} (hr : Module.finrank R M = r) : ∃ (v : Fin r → M) (c : R), c ≠ 0 ∧ ∀ m : M, ∃ s : Fin r → R, c • m = ∑ i, s i • v i

theorem **finrank_add_le_of_disjoint** {A B : Submodule R M} (h : Disjoint A B) : Module.finrank R A + Module.finrank R B ≤ Module.finrank R M

theorem **rt0_ne_neg** (H : Host E) : H.rt0 ≠ -H.rt0

theorem **exists_embedding** (H : Host E) {ρ : ℂ} (hρ : ρ ^ 2 = -(H.d : ℂ)) : ∃ τ : E →+* ℂ, τ (H.rt : E) = ρ

theorem **card_le_finrank_lat** {ρ ρ' : ℂ} (hne : ρ ≠ ρ') {n : ℕ} (hn : 0 < n) (hs : ∀ a : D.La, n • a ∈ PureType.lat hP ρ ⊔ PureType.lat hP ρ') {τ : E →+* ℂ} (hτ : τ (H.rt : E) = ρ) : Nat.card {i // ty i = τ} ≤ Module.finrank (𝒪 E) (PureType.lat hP ρ)

theorem **finrank_lat** (ht : IsTypeFn H D.p D.q ty) {n : ℕ} (hn : 0 < n) (hs : ∀ a : D.La, n • a ∈ PureType.lat hP H.rt0 ⊔ PureType.lat hP (-H.rt0)) : Module.finrank (𝒪 E) (PureType.lat hP H.rt0) = D.p ∧ Module.finrank (𝒪 E) (PureType.lat hP (-H.rt0)) = D.q

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFgForallOverLbIso

theorem **lbIso_pull_comp** {L : LB X} {M : LB Z} {a : Y ⟶ X} {b : Y ⟶ Z} (j : W ⟶ Y) {a' : W ⟶ X} {b' : W ⟶ Z} (ha : a' = j ≫ a) (hb : b' = j ≫ b) (h : LBIso (L.pull a) (M.pull b)) : LBIso (L.pull a') (M.pull b')

theorem **lbIso_pull_pull** {L : LB X} (hL : L.IsLB) {a : Y ⟶ X} (j : W ⟶ Y) {a' : W ⟶ X} (ha : a' = j ≫ a) : LBIso (L.pull a') ((L.pull a).pull j)

theorem **lbIso_pow_pull_comp** {L : LB X} {L' : LB X'} (hL : L.IsLB) (hL' : L'.IsLB) {F : X' ⟶ X} (h : LBIso L' (L.pull F)) (g : Y ⟶ X') {a : Y ⟶ X} (ha : a = g ≫ F) (N : ℕ) : LBIso (lbPow (L'.pull g) N) ((lbPow L N).pull a)

theorem **exists_fg_over_lbIso** {R₁ : Subalgebra k A} (h₁ : R₁.FG) (y : Sp R₁ ⟶ S) (Φ : pullback pB (spMap (algebraMap k A)) ⟶ Z) (hΦ : Φ ≫ q = pullback.snd _ _ ≫ spMap R₁.val.toRingHom ≫ y) {L : LB Z} {M : LB B} (hL : L.IsLB) (hM : M.IsLB) (H : LBIso (L.pull Φ) (M.pull (pullback.fst _ _))) : ∃ (R : Subalgebra k A) (h : R₁ ≤ R) (ψ : pullback pB (spMap (algebraMap k R)) ⟶ Z), R.FG ∧ ψ ≫ q = pullback.snd _ _ ≫ spMap (Subalgebra.inclusion h).toRingHom ≫ y ∧ bcMap pB R.val ≫ ψ = Φ ∧ LBIso (L.pull ψ) (M.pull (pullback.fst _ _))

theorem **exists_fg_forall_over_lbIso** {R₁ : Subalgebra k A} (h₁ : R₁.FG) (y : Sp R₁ ⟶ S) {J : Type} [Finite J] (Φ : J → (pullback pB (spMap (algebraMap k A)) ⟶ Z)) (hΦ : ∀ i, Φ i ≫ q = pullback.snd _ _ ≫ spMap R₁.val.toRingHom ≫ y) {L : J → LB Z} {M : J → LB B} (hL : ∀ i, (L i).IsLB) (hM : ∀ i, (M i).IsLB) (H : ∀ i, LBIso ((L i).pull (Φ i)) ((M i).pull (pullback.fst _ _))) : ∃ (R : Subalgebra k A) (h : R₁ ≤ R), R.FG ∧ ∀ i, ∃ ψ : pullback pB (spMap (algebraMap k R)) ⟶ Z, ψ ≫ q = pullback.snd _ _ ≫ spMap (Subalgebra.inclusion h).toRingHom ≫ y ∧ bcMap pB R.val ≫ ψ = Φ i ∧ LBIso ((L i).pull ψ) ((M i).pull (pullback.fst _ _))

theorem **dualMapR_comp** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R →+* R') (g : R' →+* R'') : (Chart.dualMapR g).comp (Chart.dualMapR f) = Chart.dualMapR (g.comp f)

theorem **IsHomFamily.descend** {D : Datum H} {R R' : Type} [CommRing R] [CommRing R'] {f : R →+* R'} (hf : Function.Injective f) {t : SAB H (Sp R)} {t' : SAB H (Sp R')} {n : t'.G.left ⟶ t.G.left} (hn : IsGrpPullback t'.G t.G (spMap f) n) (hact : ∀ a, (t'.act.act a).left ≫ n = n ≫ (t.act.act a).left) {B : Over (Sp R)} {B' : Over (Sp R')} [GrpObj B] [GrpObj B'] [Flat B.hom] {β : B'.left ⟶ B.left} (hβ : IsGrpPullback B' B (spMap f) β) {M : LB B.left} {M' : LB B'.left} {v : epsR R ⟶ B} {v' : epsR R' ⟶ B'} (hv : v'.left ≫ β = spMap (Chart.dualMapR f) ≫ v.left) {L : Submodule (𝒪 E) D.La} {ρ : R} {S S' T : Set L} {φ' : L → (B' ⟶ t'.G)} (h : IsHomFamily t' B' M' v' L (f ρ) S' φ') (hT : AddSubgroup.closure T = ⊤) (hS : S ⊆ T) (hd : ∀ a ∈ T, ∃ ψ : B ⟶ t.G, (φ' a).left ≫ n = β ≫ ψ.left ∧ ∃ N : ℕ, 1 ≤ N ∧ LBIso (lbPow (t.PP.L.pull ψ.left) N) (lbPow M (N * (D.EF (H.w0 • (a : D.La)) a).natAbs))) : ∃ φ :

theorem **cmPoint** (hT : ThetaUnifAV) (hW : Out.WeilSt) : CMPoint

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFieldCmCurvesUnitDiscriminant

private theorem **c4_pow_three_eq_j_mul_discr** {R : Type*} [CommRing R] (W : WeierstrassCurve R) [W.IsElliptic] : W.c₄ ^ 3 = W.j * W.Δ

theorem **exists_integral_curve** {j : ℂ} (hj : IsIntegral ℤ j) : ∃ E : WeierstrassCurve ℂ, IsIntegral ℤ E.a₁ ∧ IsIntegral ℤ E.a₂ ∧ IsIntegral ℤ E.a₃ ∧ IsIntegral ℤ E.a₄ ∧ IsIntegral ℤ E.a₆ ∧ E.Δ = 1 ∧ E.c₄ ^ 3 = j

theorem **latticeJ_eq_of_isHomothetic** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') : latticeJ L' = latticeJ L

private theorem **exists_ne_zero_upperHalfPlane_lattice_eq** (L : PeriodPair) : ∃ (a : ℂ) (τ : ℍ), a ≠ 0 ∧ ∀ x, x ∈ L.lattice ↔ ∃ m n : ℤ, a * (m * (τ : ℂ) + n) = x

theorem **isIntegral_latticeJ** (hK : Module.finrank ℚ K = 2) (σ : K →+* ℂ) (L : PeriodPair) (hL : ∀ (a : 𝒪 K), ∀ z ∈ L.lattice, σ (a : K) * z ∈ L.lattice) : IsIntegral ℤ (latticeJ L)

theorem **latticeJ_eq_of_class_eq** (σ : K →+* ℂ) {L L' : PeriodPair} {𝔞 𝔞' : (Ideal (𝒪 K))⁰} {c c' : ℂ} (hc : c ≠ 0) (hc' : c' ≠ 0) (h : ∀ z, z ∈ L.lattice ↔ ∃ a ∈ (𝔞 : Ideal (𝒪 K)), z = c * σ (a : K)) (h' : ∀ z, z ∈ L'.lattice ↔ ∃ a ∈ (𝔞' : Ideal (𝒪 K)), z = c' * σ (a : K)) (e : ClassGroup.mk0 𝔞 = ClassGroup.mk0 𝔞') : latticeJ L' = latticeJ L

theorem **finite_latticeJ** (σ : K →+* ℂ) (hσ : ¬ ComplexEmbedding.IsReal σ) : (latticeJ '' {L | ∀ (a : 𝒪 K), ∀ z ∈ L.lattice, σ (a : K) * z ∈ L.lattice}).Finite

theorem **exists_field_cm_curves_unit_discriminant** : ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ : K →+* ℂ), ¬ NumberField.ComplexEmbedding.IsReal σ → ∃ M : IntermediateField ℚ ℂ, FiniteDimensional ℚ M ∧ (∀ a, σ a ∈ M) ∧ ∀ L : PeriodPair, (∀ (a : 𝒪 K), ∀ z ∈ L.lattice, σ (a : K) * z ∈ L.lattice) → ∃ W : WeierstrassCurve (𝒪 M), IsUnit W.Δ ∧ (((W.c₄ : M) : ℂ)) ^ 3 * (L.g₂ ^ 3 - 27 * L.g₃ ^ 2) = 1728 * L.g₂ ^ 3 * ((W.Δ : M) : ℂ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFractionalIdealLocalGenOfValuation

theorem **exists_fractionalIdeal_isLocGen_of_valuation** : ∀ (K : Type) [Field K] [NumberField K] (oa : HeightOneSpectrum (𝒪 K) → ℤ), (Function.support oa).Finite → ∃ I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K, I ≠ 0 ∧ ∀ (p : HeightOneSpectrum (𝒪 K)) (a : K), p.valuation K a = WithZero.exp (-oa p) → IsLocGen p a I

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFractionalIdealLocalGenOfValuationStatement

theorem **_root_.GlobalTheory.T.exists_fractionalIdeal_localGen_of_valuation** : ∀ (K : Type) [Field K] [NumberField K] (oa : HeightOneSpectrum (𝒪 K) → ℤ), (Function.support oa).Finite → ∃ I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K, I ≠ 0 ∧ ∀ (p : HeightOneSpectrum (𝒪 K)) (a : K), p.valuation K a = WithZero.exp (-oa p) → IsLocGen p a I

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFractionalIdealTraceFormRankOne

theorem **trace_eq_sum** {Ω : Type} [Field Ω] [CharZero Ω] [Fintype (E →+* Ω)] (h : Nat.card (E →+* Ω) = Module.finrank ℚ E) (x : E) : ((Algebra.trace ℚ E x : ℚ) : Ω) = ∑ τ : E →+* Ω, τ x

theorem **re_pos_of_trace_pos** {β : E} (hβ : IsCMField.complexConj E β = β) (hβ0 : β ≠ 0) (h : ∀ x : E, x ≠ 0 → 0 < Algebra.trace ℚ E (β * x * IsCMField.complexConj E x)) (τ : E →+* ℂ) : 0 < (τ β).re

theorem **log_absNorm_of_selfDual** {𝔞 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (h𝔞 : 𝔞 ≠ 0) {c : E} (hc : c ≠ 0) (h : ∀ v, v ∈ 𝔞 ↔ ∀ y ∈ 𝔞, ∃ n : ℤ, Algebra.trace ℚ E (c * v * IsCMField.complexConj E y) = n) : 2 * Real.log (FractionalIdeal.absNorm 𝔞 : ℚ) + Real.log |(Algebra.norm ℚ c : ℚ)| = -(Module.finrank ℚ E * Real.log (rootDiscr E))

theorem **coe_rt_eq** (H : Host E) : (H.rt : E) = 2 * (H.w0 : E) - 1

theorem **conj_rt** (H : Host E) : IsCMField.complexConj E (H.rt : E) = -(H.rt : E)

theorem **coe_rt_ne_zero** (H : Host E) : (H.rt : E) ≠ 0

theorem **log_norm_rt** (H : Host E) : Real.log |(Algebra.norm ℚ (H.rt : E) : ℚ)| = Module.finrank ℚ E * (Real.log H.d / 2)

theorem **relTrace_eq** (H : Host E) {Ω : Type} [Field Ω] [CharZero Ω] (hΩ : Nat.card (E →+* Ω) = Module.finrank ℚ E) (τ₀ : E →+* Ω) (z : E) : ((Algebra.trace ℚ E (H.w0 * z / H.rt) : ℚ) : Ω) - (Algebra.trace ℚ E (z / H.rt) : ℚ) + (Algebra.trace ℚ E (z / H.rt) : ℚ) * τ₀ H.w0 = ∑ᶠ τ : {τ : E →+* Ω // τ (H.rt : E) = τ₀ (H.rt : E)}, τ.1 z

theorem **exists_ideal_traceForm_of_rank_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), D.p + D.q = 1 → IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → ∃ (𝔞 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (e : D.La ≃ₗ[𝒪 E] E) (β : E), 𝔞 ≠ 0 ∧ Function.Injective e ∧ LinearMap.range e = (𝔞 : Submodule (𝒪 E) E) ∧ IsCMField.complexConj E β = β ∧ (∀ τ : E →+* ℂ, 0 < (τ β).re) ∧ (∀ (Ω : Type) [Field Ω] [CharZero Ω], Nat.card (E →+* Ω) = Module.finrank ℚ E → ∀ (τ₀ : E →+* Ω) x y, τ₀ (((h x y : OK H) : 𝒪 E) : E) = ∑ᶠ τ : {τ : E →+* Ω // τ (H.rt : E) = τ₀ (H.rt : E)}, τ.1 (β * e x * IsCMField.complexConj E (e y))) ∧ 2 * Real.log (FractionalIdeal.absNorm 𝔞 : ℚ) + Real.log |(Algebra.norm ℚ β : ℚ)| = Module.finrank ℚ E * (Real.log H.d / 2 - Real.log (NumberField.rootDiscr E))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFractionalIdealTraceFormRankOneStatement

theorem **_root_.GlobalTheory.T.exists_fractionalIdeal_traceForm_rank_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), D.p + D.q = 1 → IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → ∃ (𝔞 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (e : D.La ≃ₗ[𝒪 E] E) (β : E), 𝔞 ≠ 0 ∧ Function.Injective e ∧ LinearMap.range e = (𝔞 : Submodule (𝒪 E) E) ∧ IsCMField.complexConj E β = β ∧ (∀ τ : E →+* ℂ, 0 < (τ β).re) ∧ (∀ (Ω : Type) [Field Ω] [CharZero Ω], Nat.card (E →+* Ω) = Module.finrank ℚ E → ∀ (τ₀ : E →+* Ω) x y, τ₀ (((h x y : OK H) : 𝒪 E) : E) = ∑ᶠ τ : {τ : E →+* Ω // τ (H.rt : E) = τ₀ (H.rt : E)}, τ.1 (β * e x * IsCMField.complexConj E (e y))) ∧ 2 * Real.log (FractionalIdeal.absNorm 𝔞 : ℚ) + Real.log |(Algebra.norm ℚ β : ℚ)| = Module.finrank ℚ E * (Real.log H.d / 2 - Real.log (NumberField.rootDiscr E))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFramedPointOfLieEigenbasis

theorem **exists_frPt_of_eigenbasis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (x : Sp K ⟶ Xo.X) (Φ : FibKey Xo K x) (e : (Σ b : E →+* ℂ, Fin (Dw.mult b)) ≃ Fin Φ.T.g) (U : Matrix (Fin Φ.T.g) (Fin Φ.T.g) K) (w : Fin Φ.T.g → (epsR K ⟶ Φ.T.t.G)), x ≫ Xo.s = sOf c K → U.det ≠ 0 → (∀ k, IsLieComb (U k) Φ.v (w k)) → (∀ τ i, IsEigAt c τ ((w (e ⟨τ, i⟩)).left ≫ Φ.m.φ)) → ∃ y : FrPt Xo c K, y.x = x ∧ SecTie Φ.m Φ.th y.s ∧ ∀ τ i, y.u τ i = (w (e ⟨τ, i⟩)).left ≫ Φ.m.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsFramedPointOfLieEigenbasisStatement

theorem **_root_.GlobalTheory.T.exists_framedPoint_of_lie_eigenbasis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (x : Sp K ⟶ Xo.X) (Φ : FibKey Xo K x) (e : (Σ b : E →+* ℂ, Fin (Dw.mult b)) ≃ Fin Φ.T.g) (U : Matrix (Fin Φ.T.g) (Fin Φ.T.g) K) (w : Fin Φ.T.g → (epsR K ⟶ Φ.T.t.G)), x ≫ Xo.s = sOf c K → U.det ≠ 0 → (∀ k, IsLieComb (U k) Φ.v (w k)) → (∀ τ i, IsEigAt c τ ((w (e ⟨τ, i⟩)).left ≫ Φ.m.φ)) → ∃ y : FrPt Xo c K, y.x = x ∧ SecTie Φ.m Φ.th y.s ∧ ∀ τ i, y.u τ i = (w (e ⟨τ, i⟩)).left ≫ Φ.m.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGenOfFieldExtension

theorem **exists_gen_of_isPullback** {k k' : Type} [Field k] [Field k'] (β : k →+* k') {X X' : Scheme.{0}} (f : X ⟶ Sp k) [IsProper f] [GeometricallyIntegral f] (π : X' ⟶ X) (p : X' ⟶ Sp k') (H : IsPullback π p f (spMap β)) {N : LB X} (hN : N.IsLB) (t : (N.pull π).Gen) (ht : t.IsGen) : ∃ u : N.Gen, u.IsGen

theorem **exists_gen_of_field_extension** : ∀ (k k' : Type) [Field k] [Field k'] (β : k →+* k') (A : Over (Sp k)) [GrpObj A], IsAbelianScheme A → ∀ (X' : Scheme.{0}) (π : X' ⟶ A.left) (p : X' ⟶ Sp k'), IsPullback π p A.hom (spMap β) → ∀ N : LB A.left, N.IsLB → (∃ t : LB.Gen (N.pull π), t.IsGen) → ∃ t : LB.Gen N, t.IsGen

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGenOfGenericFibre

theorem **cst_spMap** {A C : Type} [CommRing A] [CommRing C] (q : Z ⟶ Sp C) (φ : A →+* C) (z : Z) (a : A) : cst (q ≫ spMap φ) z a = cst q z (φ a)

theorem **isUnit_cst** {A : Type} [CommRing A] (q : Z ⟶ Sp A) (z : Z) {a : A} (ha : IsUnit a) : IsUnit (cst q z a)

theorem **cst_zero** {A : Type} [CommRing A] (q : Z ⟶ Sp A) (z : Z) : cst q z 0 = 0

theorem **cst_one** {A : Type} [CommRing A] (q : Z ⟶ Sp A) (z : Z) : cst q z 1 = 1

theorem **exists_ratio** {N : LB X} (hN : N.IsLB) (σ : N.Sec) (π : Y ⟶ X) (t : (N.pull π).Gen) (ht : t.IsGen) : ∃ c : Γ(Y, ⊤), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (i : N.ι), π (r z) ∈ N.U i → gs (r ≫ π) z (N.U i) (σ.f i) = gs r z ⊤ c * (gm r z ((N.pull π).U fun _ => i) (t.s fun _ => i)).val

theorem **exists_sec_isUnit** {R k : Type} [CommRing R] [Field k] (j : R →+* k) (hj : j.Flat) [CompactSpace X] [QuasiSeparatedSpace X] (f : X ⟶ Sp R) (π : Y ⟶ X) (f' : Y ⟶ Sp k) (H : IsPullback π f' f (spMap j)) [Nonempty Y] (hs : Function.Surjective f'.appTop.hom) {N : LB X} (hN : N.IsLB) (t : (N.pull π).Gen) (ht : t.IsGen) : ∃ σ : N.Sec, ∀ y i, π y ∈ N.U i → IsUnit (gs π y (N.U i) (σ.f i))

theorem **exists_unit_dvr** {R V : Type} [CommRing R] [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] (g : R →+* V) (hg : Function.Injective g) (f : X ⟶ Sp R) (π : Y ⟶ X) (p : Y ⟶ Sp V) [Smooth p] [GeometricallyIntegral p] (hw : π ≫ f = p ≫ spMap g) (e : Sp R ⟶ X) (e' : Sp V ⟶ Y) (he : e' ≫ p = 𝟙 _) (he' : e' ≫ π = spMap g ≫ e) {N : LB X} (hN : N.IsLB) (σ : N.Sec) {a : R} (ha : a ≠ 0) (ρ : (N.pull e).Gen) (hρ : ∀ (Z : Scheme.{0}) (r : Z ⟶ Sp R) (z : Z) (i : N.ι), e (r z) ∈ N.U i → gs (r ≫ e) z (N.U i) (σ.f i) = cst r z a * (gm r z ((N.pull e).U fun _ => i) (ρ.s fun _ => i)).val) (hσ : ∀ (Z : Scheme.{0}) (r : Z ⟶ X) (z : Z) (i : N.ι), (f (r z)).asIdeal = ⊥ → r z ∈ N.U i → IsUnit (gs r z (N.U i) (σ.f i))) (y : Y) (i : N.ι) (hy : π y ∈ N.U i) : ∃ u, IsUnit u ∧ gs π y (N.U i) (σ.f i) = cst (π ≫ f) y a * u

theorem **exists_gen_of_fibre** {R k : Type} [CommRing R] [IsDedekindDomain R] [Field k] (j : R →+* k) (hj : Function.Injective j) [CompactSpace X] [QuasiSeparatedSpace X] (f : X ⟶ Sp R) [Smooth f] [GeometricallyIntegral f] (e : Sp R ⟶ X) (he : e ≫ f = 𝟙 _) (π : Y ⟶ X) (f' : Y ⟶ Sp k) (H : IsPullback π f' f (spMap j)) (hs : Function.Surjective f'.appTop.hom) {N : LB X} (hN : N.IsLB) (ρ : (N.pull e).Gen) (hρ : ρ.IsGen) (t : (N.pull π).Gen) (ht : t.IsGen) : ∃ u : N.Gen, u.IsGen

theorem **exists_gen_of_generic_fibre** : Out.Laws → ∀ (R : Type) [CommRing R] [IsDedekindDomain R] (A : Over (Sp R)) [GrpObj A] (N : LB A.left) (k : Type) [Field k] (j : R →+* k), IsAbelianScheme A → N.IsLB → Function.Injective j → (∃ t : (N.pull (eSec A)).Gen, t.IsGen) → (∃ t : (N.pull (pullback.fst A.hom (spMap j))).Gen, t.IsGen) → ∃ t : N.Gen, t.IsGen

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodCmPointAtCmBasisFrame

theorem **exists_good_cmPt_at_frame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)], (∀ a : 𝒪 E, star a = H.bar a) → ∀ (r : ℕ) (U : Unf Xo c c.F.val.toRingHom r), IsUnf U → 0 < r → ∀ (k : U.Comp) (x₀ : (U.X k).dom) (Xx : FD.DataX E r) (hX : Xx.toData = U.X k) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E),

𝔄 ≠ 0 → Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ Xx.L', v = a • u} → ∀ (La : ℝ) (lam : Fin r → Fin r → E), FD.LamBasis Xx La (hX ▸ x₀.1) lam → ∃ (z : CMPt Xo c) (σ₁ : z.K →ₐ[c.F] CAt c.F.val.toRingHom), z.Good (Module.finrank ℚ E / 2) La ∧ spMap σ₁.toRingHom ≫ z.y.x = (U.un k x₀).x

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodCmPointAtCmBasisFrameStatement

theorem **_root_.GlobalTheory.T.exists_good_cmPoint_at_cmBasis_frame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)], (∀ a : 𝒪 E, star a = H.bar a) → ∀ (r : ℕ) (U : Unf Xo c c.F.val.toRingHom r), IsUnf U → 0 < r → ∀ (k : U.Comp) (x₀ : (U.X k).dom) (Xx : FD.DataX E r) (hX : Xx.toData = U.X k) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E), 𝔄 ≠ 0 → Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ Xx.L', v = a • u} → ∀ (La : ℝ) (lam : Fin r → Fin r → E), FD.LamBasis Xx La (hX ▸ x₀.1) lam → ∃ (z : CMPt Xo c) (σ₁ : z.K →ₐ[c.F] CAt c.F.val.toRingHom), z.Good (Module.finrank ℚ E / 2) La ∧ spMap σ₁.toRingHom ≫ z.y.x = (U.un k x₀).x

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodCmPointOfIntegralCmObject

theorem **exists_cmPt_good_of_cmObj** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)], (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] [Algebra c.F K] (ρ : 𝒪 K) (σ : K →+* ℂ) (X : CMObj H K ρ σ Dw.p Dw.q) (La : ℝ) (x : Sp K ⟶ Xo.X) (Φ : FibKey Xo K x) (_ : SABMap Φ.T.t X.B.t (spMap (algebraMap (𝒪 K) K))), (∀ τ, τ ∈ LamF.psiSet H ↔ LamF.labK c K τ (H.rt : E) = (ρ : K)) → (∀ j (τ : E →+* ℂ), ‖τ (X.hh j : E)‖ ≤ La ^ 2) → x ≫ Xo.s = sOf c K → ∀ (S : Type) [CommRing S] [Algebra c.F S] (σ₁ : K →ₐ[c.F] S), ∃ (z : CMPt Xo c) (σ' : z.K →ₐ[c.F] S), z.Good (Module.finrank ℚ E / 2) La ∧ spMap σ'.toRingHom ≫ z.y.x = spMap σ₁.toRingHom ≫ x

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodCmPointOfIntegralCmObjectStatement

theorem **_root_.GlobalTheory.T.exists_good_cm_point_of_integral_cm_object** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)], (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] [Algebra c.F K] (ρ : 𝒪 K) (σ : K →+* ℂ) (X : CMObj H K ρ σ Dw.p Dw.q) (La : ℝ) (x : Sp K ⟶ Xo.X) (Φ : FibKey Xo K x) (_ : SABMap Φ.T.t X.B.t (spMap (algebraMap (𝒪 K) K))), (∀ τ, τ ∈ LamF.psiSet H ↔ LamF.labK c K τ (H.rt : E) = (ρ : K)) → (∀ j (τ : E →+* ℂ), ‖τ (X.hh j : E)‖ ≤ La ^ 2) → x ≫ Xo.s = sOf c K → ∀ (S : Type) [CommRing S] [Algebra c.F S] (σ₁ : K →ₐ[c.F] S), ∃ (z : CMPt Xo c) (σ' : z.K →ₐ[c.F] S), z.Good (Module.finrank ℚ E / 2) La ∧ spMap σ'.toRingHom ≫ z.y.x = spMap σ₁.toRingHom ≫ x

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodModelOfIsogeny

private theorem **over_pullback_map_left_fst** {S S' : Scheme} (φ : S' ⟶ S) (Ga X : Over S) (io : Ga ⟶ X) : ((Over.pullback φ).map io).left ≫ pullback.fst X.hom φ = pullback.fst _ _ ≫ io.left

private theorem **over_pullback_map_left_snd** {S S' : Scheme} (φ : S' ⟶ S) (Ga X : Over S) (io : Ga ⟶ X) : ((Over.pullback φ).map io).left ≫ pullback.snd X.hom φ = pullback.snd _ _

theorem **_root_.GlobalTheory.IsogenyModel.exists_flat_closure** (T K : Type) [CommRing T] [IsDedekindDomain T] [Field K] [Algebra T K] [IsFractionRing T K] (X : Over (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp T)) (Z : Over (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp K)) (i : Z ⟶ (Over.pullback (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.spMap (algebraMap T K))).obj X) (hi : IsClosedImmersion i.left) : ∃ (Ga : Over (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp T)) (io : Ga ⟶ X) (e : (Over.pullback (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.spMap (algebraMap T K))).obj Ga ≅ Z), IsClosedImmersion io.left ∧ Flat Ga.hom ∧ e.hom ≫ i = (Over.pullback (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.spMap (algebraMap T K))).map io

theorem **_root_.GlobalTheory.IsogenyModel.exists_kernel** {G B : Over S} [GrpObj G] [GrpObj B] [IsSeparated B.hom] (f : G ⟶ B) [IsMonHom f] [IsFinite f.left] : ∃ (Z : Over S) (_ : GrpObj Z) (i : Z ⟶ G), IsMonHom i ∧ IsClosedImmersion i.left ∧ IsFinite Z.hom ∧ IsPullback i (toUnit Z) f η[B]

theorem **_root_.GlobalTheory.IsogenyModel.exists_iso_of_same_kernel** {G A B Z : Over S} [GrpObj G] [GrpObj A] [GrpObj B] (q : G ⟶ A) (f : G ⟶ B) (i : Z ⟶ G) [IsMonHom q] [IsMonHom f] [Flat q.left] [Surjective q.left] [QuasiCompact q.left] [Surjective f.left] [IsProper A.hom] [IsSeparated B.hom] [IsReduced B.left] (hq : IsPullback i (toUnit Z) q η[A]) (hf : IsPullback i (toUnit Z) f η[B]) : ∃ e : A ≅ B, IsMonHom e.hom ∧ q ≫ e.hom = f

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodModelOfPowQuotient

theorem **isMonHom_prod** {G H : Over S} [GrpObj G] [GrpObj H] [IsCommMonObj H] {n : ℕ} (a : Fin n → (G ⟶ H)) (ha : ∀ i, IsMonHom (a i)) : IsMonHom (∏ i, a i)

theorem **comp_prod** {T G H : Over S} [GrpObj H] [IsCommMonObj H] {n : ℕ} (x : T ⟶ G) (a : Fin n → (G ⟶ H)) : x ≫ ∏ i, a i = ∏ i, x ≫ a i

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodReductionModelCmPeriodFibre

theorem **ellMap_twist** {R R' : Type} [CommRing R] [CommRing R'] {ω : R} {ω' : R'} {C' : Ell H R' ω'} {C : Ell H R ω} {f : R →+* R'} {φ : C'.P.A.left ⟶ C.P.A.left} (h : EllMap C' C f φ) (b : Bool) : EllMap (twist C' b) (twist C b) f φ

theorem **exists_pat_univ** {D : Datum H} {l : Level H} (Xo : Moduli D l) {S : Scheme.{0}} (f : S ⟶ Xo.X) : ∃ (T : PAT H S) (_ : SABMap T.t Xo.univ.t f), T.g = Module.finrank ℚ E / 2 * (D.p + D.q)

theorem **exists_sign** {ω ρ : ℂ} (h : 2 * ω - 1 = ρ) (hρ : ρ ^ 2 = -(H.d : ℂ)) : ∃ ε : ℤˣ, 0 < ((ε : ℤ) : ℝ) * ω.im

theorem **exists_pure_span** {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} {hP : IsPeriod D ty Pe} {p q : ℕ} (Z : CMLat hP p q) {N : ℕ} (hN : N = Module.finrank ℚ E / 2 * (D.p + D.q)) : ∃ (b : Fin N → Bool) (a : ∀ j, PureType.lat hP (if b j then H.rt0 else -H.rt0)), ∀ z : Fin g → ℂ, ∃ c : Fin N → ℂ, z = ∑ j, c j • Pe (a j)

theorem **ontoAt_of_exp** {S S' : Scheme.{0}} {G' : Over S'} {G : Over S} [GrpObj G'] [GrpObj G] [IsCommMonObj G] {f : S' ⟶ S} {φ : G'.left ⟶ G.left} (hφ : IsGrpPullback G' G f φ) (t : Sp ℂ ⟶ S') {s : Sp ℂ ⟶ S} (hs : t ≫ f = s) {V : Type} [AddCommGroup V] [Module ℂ V] (e : V → (Over.mk s ⟶ G)) (hadd : ∀ z w, e (z + w) = e z ∗ e w) (honto : Function.Surjective e) {N : ℕ} (v : Fin N → V) (hv : ∀ z, ∃ c : Fin N → ℂ, z = ∑ j, c j • v j) {B : Over S'} (ψ : Fin N → (B ⟶ G')) (hψ : ∀ j (z : ℂ), ∃ x : Over.mk t ⟶ B, x.left ≫ (ψ j).left ≫ φ = (e (z • v j)).left) : OntoAt (B

theorem **exists_model** : CMCurve → CMPoint → ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (p q : ℕ) (_ : CMLat hP p q) (k : IntermediateField ℚ ℂ), FiniteDimensional ℚ k → ∃ (K₁ : IntermediateField ℚ ℂ) (_ : NumberField K₁) (hk : KA l ≤ K₁) (ω : Bool → 𝒪 K₁) (C : ∀ b, Ell H (𝒪 K₁) (ω b)) (ε : Bool → ℤˣ) (h : Bool → (FU.Di i).La → (FU.Di i).La → OK H) (B : PAT H (Sp

(𝒪 K₁))) (y : Sp K₁ ⟶ Xo.X) (t₀ : SAB H (Sp K₁)) (m : SABMap t₀ Xo.univ.t y) (m' : SABMap t₀ B.t (spMap (algebraMap (𝒪 K₁) K₁))) (ψ : ∀ b : Bool, PureType.lat hP (if b then H.rt0 else -H.rt0) → ((C b).P.A ⟶ B.t.G)) (ω' : Bool

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGoodReductionModelCmPeriodFibreStatement

theorem **_root_.GlobalTheory.T.exists_goodReduction_model_cmPeriod_fibre** : CMCurve → CMPoint → ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (p q : ℕ) (_ : CMLat hP p q) (k : IntermediateField ℚ ℂ), FiniteDimensional ℚ k → ∃ (K₁ : IntermediateField ℚ ℂ) (_ : NumberField K₁) (hk : KA l ≤ K₁) (ω : Bool → 𝒪 K₁) (C : ∀ b, Ell H (𝒪 K₁) (ω b)) (ε : Bool → ℤˣ) (h : Bool → (FU.Di i).La → (FU.Di i).La → OK H) (B : PAT H (Sp (𝒪 K₁))) (y : Sp K₁ ⟶ Xo.X) (t₀ : SAB H (Sp K₁)) (m : SABMap t₀ Xo.univ.t y) (m' : SABMap t₀ B.t (spMap (algebraMap (𝒪 K₁) K₁))) (ψ : ∀ b : Bool, PureType.lat hP (if b then H.

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGrpObjOverOfPoints

theorem **_root_.GlobalTheory.exists_grpObj_over_of_points** : ∀ (S : Scheme.{0}) (X : Over S) (e : 𝟙_ _ ⟶ X) (m : X ⊗ X ⟶ X) (n : X ⟶ X), (∀ (T : Over S) (x : T ⟶ X), lift (toUnit T ≫ e) x ≫ m = x) → (∀ (T : Over S) (x : T ⟶ X), lift (x ≫ n) x ≫ m = toUnit T ≫ e) → (∀ (T : Over S) (x y : T ⟶ X), lift x y ≫ m = lift y x ≫ m) → (∀ (T : Over S) (x y z : T ⟶ X), lift (lift x y ≫ m) z ≫ m = lift x (lift y z ≫ m) ≫ m) → ∃ G : GrpObj X, η[X] = e ∧ μ[X] = m ∧ (GrpObj.inv : X ⟶ X) = n ∧ IsCommMonObj X

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsGrpObjPolarised

theorem **polarised_of_grpObj** (R : Type) [CommRing R] (W : _root_.WeierstrassCurve R) (hΔ : IsUnit W.Δ) [GrpObj (schemeOver W)] (hc : IsCommMonObj (schemeOver W)) (hη : η[schemeOver W].left = zero W) : IsAbelianScheme (schemeOver W) ∧ SmoothOfRelativeDimension 1 (schemeOver W).hom ∧ (originLB W).IsLB ∧ Out.Rigidified (schemeOver W) (originLB W) ∧ Out.Sym (schemeOver W) (originLB W) ∧ Out.RelAmple (schemeOver W) (originLB W) ∧ Out.KTriv (schemeOver W) (originLB W)

theorem **exists_grpObj_polarised** (R : Type) [CommRing R] (W : _root_.WeierstrassCurve R) (hΔ : IsUnit W.Δ) : ∃ _ : GrpObj (schemeOver W), IsCommMonObj (schemeOver W) ∧ η[schemeOver W].left = zero W ∧ (∀ (A : Type) [CommRing A] (ρ : R →+* A) (P Q T : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) (hQ : IsUnimodularSol (W.map ρ) Q) (hT : IsUnimodularSol (W.map ρ) T), (Proportional T ((W.map ρ).toProjective.addXYZ P Q) ∧ Proportional T (diagXYZ (W.map ρ).toProjective P Q)) → ∀ x y : Over.mk (spMap ρ) ⟶ schemeOver W, x.left = ptHom W ρ P hP → y.left = ptHom W ρ Q hQ → (lift x y ≫ μ[schemeOver W]).left = ptHom W ρ T hT) ∧ (∀ (A : Type) [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P), ∃ hN : IsUnimodularSol (W.map ρ) ((W.map ρ).toProjective.neg P), ptHom W ρ P hP ≫ (GrpObj.inv : schemeOver W ⟶ schemeOver W).left = ptHom W ρ _ hN) ∧ (IsAbelianScheme (schemeOver W) ∧

theorem **exists_polAV** : ∀ (R : Type) [CommRing R] (W : _root_.WeierstrassCurve R), IsUnit W.Δ → ∃ (P : Out.PolAV (Sp R)) (ψ : P.A.left ⟶ scheme W) (v : epsR R ⟶ P.A), P.g = 1 ∧ IsIso ψ ∧ ψ ≫ toSpec W = P.A.hom ∧ eSec P.A ≫ ψ = zero W ∧ (∃ h : IsUnimodularSol (W.map (algebraMap R (DualNumber R))) ![DualNumber.eps, -1, 0], v.left ≫ ψ = ptHom W (algebraMap R (DualNumber R)) ![DualNumber.eps, -1, 0] h) ∧ IsLieBasis ![v]

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsHermFormCurve

theorem **countable_lattice** : (u.Λ : Set (Fin g → ℂ)).Countable

theorem **lift_unique** {U : Type*} [AddCommGroup U] [Module ℂ U] {F F' : U →ₗ[ℂ] (Fin g → ℂ)} (h : ∀ z, u.ex (F z) = u.ex (F' z)) : F = F'

theorem **ex_smul_of_isLieComb** (P : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g) {v : epsR ℂ ⟶ P.A} (hv : IsLieBasis ![v]) (f : P.A ⟶ P.A) [hf : IsMonHom f] {a : ℂ} (ha : IsLieComb ![a] ![v] (v ≫ f)) (z : Fin P.g → ℂ) : u.ex (a • z) = u.ex z ≫ f

theorem **exists_hermForm_curve** (hT : ThetaUnifAV) (P : Out.PolAV (Sp ℂ)) (hg : P.g = 1) (u₀ : HolUnif P.A P.L P.g) {d : ℕ} (hd : 0 < d) {c : ℂ} (hc : c ≠ 0) (hΛ : PureType.IsQuadLattice u₀.Λ d c) : ∃ (u : HolUnif P.A P.L P.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ) (G : (Fin P.g → ℂ) →ₗ[ℂ] (Fin P.g → ℂ)), u.IsAH Hm χ ∧ (∀ z, u.ex (G z) = u₀.ex z) ∧ ∀ z w : ℂ, hermForm Hm (G fun _ => c * z) (G fun _ => c * w) = 2 * z * starRingEnd ℂ w / Real.sqrt d

theorem **exists_isExp** (T : PAT H (Sp ℂ)) (D : Datum H) (u : HolUnif T.t.G T.t.PP.L T.g) {Hm : Matrix (Fin T.g) (Fin T.g) ℂ} (hH : Hm.IsHermitian) (hpos : ∀ z : Fin T.g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) (P : D.La →+ (Fin T.g → ℂ)) (hi : Function.Injective P) (hΛ : ∀ v, v ∈ u.Λ ↔ v ∈ P.range) (hE : ∀ a b, (hermForm Hm (P a) (P b)).im = D.EF a b) (hW : ∀ (k : ℕ) (a b : D.La) (c : ℂˣ), 0 < k → (c : ℂ) = Complex.exp (-(2 * Real.pi * Complex.I) * (D.EF a b : ℂ) / k) → Out.WeilIs T.t.G T.t.PP.L k (u.ex ((k : ℂ)⁻¹ • P a)) (u.ex ((k : ℂ)⁻¹ • P b)) (unitOn c)) (r : 𝒪 E → Module.End ℂ (Fin T.g → ℂ)) (hr : ∀ s z, u.ex (r s z) = u.ex z ≫ T.t.act.act s) (hrP : ∀ s a, r s (P a) = P (s • a)) {g : ℕ} (B : (Fin T.g → ℂ) ≃ₗ[ℂ] (Fin g → ℂ)) (ty : Fin g → (E →+* ℂ)) (hB : ∀ (s : 𝒪 E) v j, B (r s v) j = ty j (s : E) * B v j) : ∃ e : (Fin g → ℂ) → (Over.mk (spMap (RingHom.id ℂ)) ⟶ T.t.G), IsExp T (RingHom.

theorem **exists_isExp_of_isTen** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (ω : ℂ) (C : Ell H ℂ ω) (u₀ : HolUnif C.P.A C.P.L C.P.g) (c : ℂ) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (T : PAT H (Sp ℂ)) (t : IsTen C D h T), c ≠ 0 → PureType.IsQuadLattice u₀.Λ H.d c → 0 < ((ε : ℤ) : ℝ) * ω.im → IsKForm D ε h → (D.q = 0 ∧ 2 * ω - 1 = H.rt0) ∨ (D.p = 0 ∧ 2 * ω - 1 = -H.rt0) → ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (e : (Fin g → ℂ) → (Over.mk (spMap (RingHom.id ℂ)) ⟶ T.t.G)), IsTypeFn H D.p D.q ty ∧ IsExp T (RingHom.id ℂ) D ty Pe e ∧ (∀ a (z : ℂ), (e (z • Pe a)).left = (u₀.ex (fun _ => c * z) ≫ t.ten.tmul a).left) ∧ ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (f : D.La →+ V), (∀ a, f (H.w0 • a) = ω • f a) → ∃ F : (Fin g → ℂ) →ₗ[ℂ] V, ∀ a, F (Pe a) = f a

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsHomFamilyOfPureTypeComplexTriple

theorem **exists_datum_period** (hT : ThetaUnifAV) (hL : Out.Laws) (T : PAT H (Sp ℂ)) {p q g : ℕ} {ty : Fin g → (E →+* ℂ)} (hty : IsTypeFn H p q ty) (hdet : ∀ τ : 𝒪 E →+* ℂ, (τ H.rt = H.rt0 → LieMult T.t.act (𝟙 (Sp ℂ)) τ p) ∧ (τ H.rt ≠ H.rt0 → LieMult T.t.act (𝟙 (Sp ℂ)) τ q)) (u : HolUnif T.pol.A T.pol.L T.pol.g) {Hm : Matrix (Fin T.pol.g) (Fin T.pol.g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) : ∃ (D : Datum H) (Pe : D.La →+ (Fin g → ℂ)) (F : (Fin g → ℂ) ≃ₗ[ℂ] (Fin T.pol.g → ℂ)), D.p = p ∧ D.q = q ∧ IsPeriod D ty Pe ∧ (∀ z, F z ∈ u.Λ ↔ z ∈ Pe.range) ∧ (∀ a b, (hermForm Hm (F (Pe a)) (F (Pe b))).im = D.EF a b) ∧ ∀ (s : 𝒪 E) z, u.ex (F fun k => ty k (s : E) * z k) = u.ex (F z) ≫ T.t.act.act s

theorem **exists_isHomFamily** (T : PAT H (Sp ℂ)) {g : ℕ} {ty : Fin g → (E →+* ℂ)} (u : HolUnif T.pol.A T.pol.L T.pol.g) {Hm : Matrix (Fin T.pol.g) (Fin T.pol.g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) (hD : ∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) (hθ : ∀ (k : ℕ) s, ∃! θ, u.IsThetaFn Hm χ k θ ∧ u.IsFnPow k s θ) {D : Datum H} {Pe : D.La →+ (Fin g → ℂ)} (F : (Fin g → ℂ) ≃ₗ[ℂ] (Fin T.pol.g → ℂ)) (hP : IsPeriod D ty Pe) (hF : ∀ z, F z ∈ u.Λ ↔ z ∈ Pe.range) (hE : ∀ a b, (hermForm Hm (F (Pe a)) (F (Pe b))).im = D.EF a b) (hact : ∀ (s : 𝒪 E) z, u.ex (F fun k => ty k (s : E) * z k) = u.ex (F z) ≫ T.t.act.act s) {PC : Out.PolAV (Sp ℂ)} (u₀ : HolUnif PC.A PC.L PC.g) {c : ℂ} (hg : PC.g = 1) (hu : IsThetaUnif u₀) (hc : c ≠ 0) (hΛ : IsQuadLattice u₀.Λ H.d c) {ρ : ℂ} (hρ : ρ ^ 2 = -(H.d : ℂ)) (L : Submodule (𝒪 E) D.La) (hLρ : ∀ a : L, (a : D.La) ∈ lat hP ρ) : ∃ φ : L → (PC.A ⟶ T.t.G), (∀ v, IsTangent v → IsHo

theorem **ontoAt_of_span** {A B : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} (u : HolUnif A L g) {I : Type} (φ : I → (B ⟶ A)) (v : I → Fin g → ℂ) (hv : Submodule.span ℂ (Set.range v) = ⊤) (x : ℂ → (𝟙_ (Over (Sp ℂ)) ⟶ B)) (h : ∀ a z, x z ≫ φ a = u.ex (z • v a)) : OntoAt (B

theorem **exists_isFam_of_pureType** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (ω : ℂ) (C : Ell H ℂ ω) (T : PAT H (Sp ℂ)) (r : ℕ), C.IsPrincipalAt (RingHom.id ℂ) → HasType H T.t.act (2 * ω - 1) r 0 → ∃ (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (ψ : (⊤ : Submodule (𝒪 E) D.La) → (C.P.A ⟶ T.t.G)), D.p + D.q = r ∧ IsKForm D ε h ∧ (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) ∧ IsFam C T h ⊤ ψ ∧ OntoAt ψ (spMap (RingHom.id ℂ))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsHomFamilyOfPureTypeComplexTripleStatement

theorem **_root_.GlobalTheory.T.exists_homFamily_of_pureType_complex_triple** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (ω : ℂ) (C : Ell H ℂ ω) (T : PAT H (Sp ℂ)) (r : ℕ), C.IsPrincipalAt (RingHom.id ℂ) → HasType H T.t.act (2 * ω - 1) r 0 → ∃ (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (ψ : (⊤ : Submodule (𝒪 E) D.La) → (C.P.A ⟶ T.t.G)), D.p + D.q = r ∧ IsKForm D ε h ∧ (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) ∧ IsFam C T h ⊤ ψ ∧ OntoAt ψ (spMap (RingHom.id ℂ))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsHomOfLinearScalingHermForm

lemma **val_mul** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (f f' : Γ(X, U)) : valAt' x U (f * f') = valAt' x U f * valAt' x U f'

lemma **val_add** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (f f' : Γ(X, U)) : valAt' x U (f + f') = valAt' x U f + valAt' x U f'

lemma **val_one** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) : valAt' x U 1 = 1

lemma **val_zero** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) : valAt' x U 0 = 0

lemma **val_resG** {U V : X.Opens} (hVU : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵁ V) (f : Γ(X, U)) : valAt' x V (resG X hVU f) = valAt' x U f

lemma **val_appLE** {Y : Scheme.{0}} (φ : X ⟶ Y) (U : Y.Opens) (V : X.Opens) (h : V ≤ φ ⁻¹ᵁ U) (f : Γ(Y, U)) (hx : ⊤ ≤ x ⁻¹ᵁ V) : valAt' x V ((φ.appLE U V h).hom f) = valAt' (x ≫ φ) U f

lemma **le_preimage_mono** {U V : X.Opens} (hVU : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵁ V) : ⊤ ≤ x ⁻¹ᵁ U

lemma **val_sec** {L : LB X} (s : L.Sec) (i j : L.ι) (hi : ⊤ ≤ x ⁻¹ᵁ L.U i) (hj : ⊤ ≤ x ⁻¹ᵁ L.U j) : valAt' x (L.U i) (s.f i) = valAt' x (L.U i ⊓ L.U j) (L.g i j).val * valAt' x (L.U j) (s.f j)

lemma **le_preimage_nonzeroLocus** {L : LB X} (s : L.Sec) (j : L.ι) (hj : ⊤ ≤ x ⁻¹ᵁ L.U j) : ⊤ ≤ x ⁻¹ᵁ Out.nonzeroLocus s ↔ valAt' x (L.U j) (s.f j) ≠ 0

lemma **exists_hom_affine** {W : X.Opens} {D : Y.Opens} (hW : IsAffineOpen W) (hD : IsAffineOpen D) (hin : ∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → ⊤ ≤ ey w ⁻¹ᵁ D) (hfn : ∀ f : Γ(Y, D), ∃ h : Γ(X, W), ∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → valAt' (ex w) W h = valAt' (ey w) D f) : ∃ g : W.toScheme ⟶ Y, ∀ w (y : Sp ℂ ⟶ W.toScheme), y ≫ W.ι = ex w → y ≫ g = ey w

lemma **exists_hom_of_values** {S : Scheme.{0}} (q : Y ⟶ S) [IsSeparated q] [IsAffine S] (h : ∀ z, ∃ (W : X.Opens) (D : Y.Opens), IsAffineOpen D ∧ ⊤ ≤ ex z ⁻¹ᵁ W ∧ (∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → ⊤ ≤ ey w ⁻¹ᵁ D) ∧ ∀ f : Γ(Y, D), ∃ h : Γ(X, W), ∀ w, ⊤ ≤ ex w ⁻¹ᵁ W → valAt' (ex w) W h = valAt' (ey w) D f) : ∃ g : X ⟶ Y, ∀ z, ex z ≫ g = ey z

lemma **chart_le** (j : PUnit → L.ι) : L.wchart (fun _ : PUnit => 𝟙 X) j ≤ σ ⁻¹ᵁ L.wchart (fun _ : Fin N => κ) (fun _ => j PUnit.unit)

lemma **exists_sec_pow** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (s : (L.LinComb (K

lemma **isThetaFn_comp** {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} {Hm' : Matrix (Fin g') (Fin g') ℂ} {χ' : u'.Λ → ℂ} (hχ : ∀ l, χ l ^ 2 = 1) (hχ' : ∀ l, χ' l ^ 2 = 1) (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (n : ℕ) (hΛ : ∀ l ∈ u.Λ, F l ∈ u'.Λ) (hH : ∀ z w, hermForm Hm' (F z) (F w) = (n : ℂ) * hermForm Hm z w) {N : ℕ} {θ₁ θ₂ : (Fin g' → ℂ) → ℂ} (h₁ : u'.IsThetaFn Hm' χ' N θ₁) (h₂ : u'.IsThetaFn Hm' χ' N θ₂) : u.IsThetaFn Hm χ (2 * (n * N)) fun z => θ₁ (F z) * θ₂ (F z)

lemma **le_preimage_lbPow** {X : Scheme.{0}} (x : Sp ℂ ⟶ X) (L : LB X) (k : ℕ) (j : PUnit → L.ι) : ⊤ ≤ x ⁻¹ᵁ (lbPow L k).U j ↔ ⊤ ≤ x ⁻¹ᵁ L.U (j PUnit.unit)

lemma **exists_quot** [LocallyOfFiniteType A.hom] {k : ℕ} {s t : (lbPow L k).Sec} {θ θ : (Fin g → ℂ) → ℂ} (hs : u.IsFnPow k s θ) (ht : u.IsFnPow k t θ) (j : (lbPow L k).ι) : ∃ h : Γ(A.left, A.left.basicOpen (t.f j)), ∀ w, ⊤ ≤ (u.ex w).left ⁻¹ᵁ A.left.basicOpen (t.f j) → valAt' (u.ex w).left _ h * θ w = θ w

lemma **isMonHom_of_additive** [Smooth A.hom] [IsSeparated A'.hom] (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (f : A ⟶ A') (hf : ∀ z, u'.ex (F z) = u.ex z ≫ f) : IsMonHom f where

lemma **exists_hom** [Smooth A.hom] [Smooth A'.hom] [IsSeparated A'.hom] (hL : L.IsLB) (hL' : L'.IsLB) (ha : Out.RelAmple A' L') {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} {Hm' : Matrix (Fin g') (Fin g') ℂ} {χ' : u'.Λ → ℂ} (hχ : ∀ l, χ l ^ 2 = 1) (hχ' : ∀ l, χ' l ^ 2 = 1) (hs : ∀ (k : ℕ) θ, u.IsThetaFn Hm χ k θ → ∃! s, u.IsFnPow k s θ) (hθ : ∀ (k : ℕ) s, ∃! θ, u'.IsThetaFn Hm' χ' k θ ∧ u'.IsFnPow k s θ) (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (n : ℕ) (hΛ : ∀ l ∈ u.Λ, F l ∈ u'.Λ) (hH : ∀ z w, hermForm Hm' (F z) (F w) = (n : ℂ) * hermForm Hm z w) : ∃ f : A.left ⟶ A'.left, ∀ z, (u.ex z).left ≫ f = (u'.ex (F z)).left

theorem **exists_hom_of_linear_scaling_hermForm** : ∀ (P P' : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g) (u' : HolUnif P'.A P'.L P'.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ) (Hm' : Matrix (Fin P'.g) (Fin P'.g) ℂ) (χ' : u'.Λ → ℂ), u.IsAH Hm χ → u'.IsAH Hm' χ' → (∀ (k : ℕ) θ, u.IsThetaFn Hm χ k θ → ∃! s, u.IsFnPow k s θ) → (∀ (k : ℕ) s, ∃! θ, u'.IsThetaFn Hm' χ' k θ ∧ u'.IsFnPow k s θ) → ∀ (F : (Fin P.g → ℂ) →ₗ[ℂ] (Fin P'.g → ℂ)) (n : ℕ), (∀ l ∈ u.Λ, F l ∈ u'.Λ) → (∀ z w, hermForm Hm' (F z) (F w) = (n : ℂ) * hermForm Hm z w) → ∃ f : P.A ⟶ P'.A, IsMonHom f ∧ ∀ z, u'.ex (F z) = u.ex z ≫ f

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIntegralEigenbasisOfFrameAtLocalModel

theorem **comb_eq** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {m : ℕ} {v : Fin m → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)} (hv : ∀ j, IsTanAt x (v j)) {a : Fin m → S} {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (h : IsCombAt x v a w) : (combT x (fun j => ovT (t

theorem **lieInt_change** (T : CDVR) (io : c.F →+* T.Fr) {G : SAB H (Sp T.V)} {t' : SAB H (Sp T.Fr)} {x : Sp T.Fr ⟶ Xo.X} (m1 : SABMap t' Xo.univ.t x) (m2 : SABMap t' G (spMap (algebraMap T.V T.Fr))) (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber T.Fr) ⟶ Xo.univ.t.G.left)) (hu : ∀ τ j, IsTanAt x (u τ j)) (h : LieInt T io m1 m2 u) (k : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) T.V) (hk : ∀ τ, IsUnit (k τ).det) : LieInt T io m1 m2 (fun τ i => (combT x (fun j => ovT (t

theorem **GlobalTheory.LamT.lieInt_exists_and_change** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (T : CDVR) (io : c.F →+* T.Fr) (G : SAB H (Sp T.V)) (t' : SAB H (Sp T.Fr)) (x : Sp T.Fr ⟶ Xo.X) (m1 : SABMap t' Xo.univ.t x) (m2 : SABMap t' G (spMap (algebraMap T.V T.Fr))), letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIntegralEigenbasisOfFrameAtLocalModelStatement

theorem **_root_.GlobalTheory.T.exists_integralEigenbasis_of_frame_at_localModel** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (T : CDVR) (io : c.F →+* T.Fr) (G : SAB H (Sp T.V)) (t' :

SAB H (Sp T.Fr)) (x : Sp T.Fr ⟶ Xo.X) (m1 : SABMap t' Xo.univ.t x) (m2 : SABMap t' G (spMap (algebraMap T.V T.Fr))), letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsInverseIsogenyPolAV

theorem **exists_inverse_isogeny_polAV** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] (σ : K →+* ℂ) (P P' : Out.PolAV (Sp K)) (Φ : P.A ⟶ P'.A), P.g = P'.g → IsMonHom Φ → Surjective Φ.left → ∃ (N : ℕ) (ψ : P'.A ⟶ P.A), 0 < N ∧ IsMonHom ψ ∧ Φ ≫ ψ = 𝟙 P.A ^ N ∧ ψ ≫ Φ = 𝟙 P'.A ^ N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsInverseIsogenyPolAVStatement

theorem **_root_.GlobalTheory.T.exists_inverse_isogeny_of_surjective_polAV** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] (σ : K →+* ℂ) (P P' : Out.PolAV (Sp K)) (Φ : P.A ⟶ P'.A), P.g = P'.g → IsMonHom Φ → Surjective Φ.left → ∃ (N : ℕ) (ψ : P'.A ⟶ P.A), 0 < N ∧ IsMonHom ψ ∧ Φ ≫ ψ = 𝟙 P.A ^ N ∧ ψ ≫ Φ = 𝟙 P'.A ^ N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsInverseIsogenyRingOfIntegers

theorem **exists_isGrpPullback_polAV** (hB : Out.PolBaseChangeSt) {S S' : Scheme.{0}} (f : S' ⟶ S) (P : Out.PolAV S) : ∃ (P' : Out.PolAV S') (φ : P'.A.left ⟶ P.A.left), P'.g = P.g ∧ IsGrpPullback P'.A P.A f φ

theorem **IsGrpPullback.id_pow_left** {S S' : Scheme.{0}} {f : S' ⟶ S} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) (N : ℕ) : (𝟙 G' ^ N).left ≫ φ = φ ≫ (𝟙 G ^ N).left

theorem **exists_inverse_isogeny_ringOfIntegers** : ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (P P' : Out.PolAV (Sp (𝒪 K))) (Φ : P.A ⟶ P'.A), P.g = P'.g → IsMonHom Φ → (∀ y : Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ P'.A, ∃ x : Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ P.A, y = x ≫ Φ) → ∃ (N : ℕ) (ψ : P'.A ⟶ P.A), 0 < N ∧ IsMonHom ψ ∧ Φ ≫ ψ = 𝟙 P.A ^ N ∧ ψ ≫ Φ = 𝟙 P'.A ^ N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsInverseIsogenyRingOfIntegersStatement

theorem **_root_.GlobalTheory.T.exists_inverse_isogeny_of_surjective_points** : ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (P P' : Out.PolAV (Sp (𝒪 K))) (Φ : P.A ⟶ P'.A), P.g = P'.g → IsMonHom Φ → (∀ y : Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ P'.A, ∃ x : Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ P.A, y = x ≫ Φ) → ∃ (N : ℕ) (ψ : P'.A ⟶ P.A), 0 < N ∧ IsMonHom ψ ∧ Φ ≫ ψ = 𝟙 P.A ^ N ∧ ψ ≫ Φ = 𝟙 P'.A ^ N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsInverseIsogenySumOfCmHoms

theorem **isMonHom_of_points** {G H : Over S} [GrpObj G] [GrpObj H] (f : G ⟶ H) (h : ∀ (T : Over S) (a b : T ⟶ G), (a ∗ b) ≫ f = a ≫ f ∗ b ≫ f) : IsMonHom f where

theorem **comp_prod** {T G H : Over S} [GrpObj H] [IsCommMonObj H] {n : ℕ} (x : T ⟶ G) (a : Fin n → (G ⟶ H)) : x ≫ ∏ i, a i = ∏ i, x ≫ a i

theorem **exists_point** {Q G' G H' H T : Over S} [GrpObj G'] [GrpObj G] [GrpObj H'] [GrpObj H] {p q : ℕ} (pG : Out.IsPowGrp G' G p) (pH : Out.IsPowGrp H' H q) (u : Q ⟶ G') (v : Q ⟶ H') [IsIso (lift u v)] (a : Fin p → (T ⟶ G)) (b : Fin q → (T ⟶ H)) : ∃ x : T ⟶ Q, (∀ i, x ≫ u ≫ pG.pr i = a i) ∧ ∀ i, x ≫ v ≫ pH.pr i = b i

theorem **exists_inverse_isogeny_of_sum** : ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (A A' B : Out.PolAV (Sp (𝒪 K))) (p q : ℕ) (φ : Fin p → (A.A ⟶ B.A)) (φ' : Fin q → (A'.A ⟶ B.A)), (∀ i, IsMonHom (φ i)) → (∀ i, IsMonHom (φ' i)) → B.g = p ∗ A.g + q ∗ A'.g → (∀ y : Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ B.A, ∃ (x : Fin p → (Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ A.A)) (x' : Fin q → (Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ A'.A)), y = (List.ofFn fun i => x i ≫ φ i).prod ∗ (List.ofFn fun i => x' i ≫ φ' i).prod) → ∃ (N : ℕ) (ψ : Fin p → (B.A ⟶ A.A)) (ψ' : Fin q → (B.A ⟶ A'.A)), 0 < N ∧ (∀ (Tb : Over (Sp (𝒪 K))) (x : Tb ⟶ B.A), padd B.A (psumPt fun i => x ≫ ψ i ≫ φ i) (psumPt fun i => x ≫ ψ' i ≫ φ' i) = pnsmulPt N x) ∧ ∀ (Tb : Over (Sp (𝒪 K))) (a : Fin p → (Tb ⟶ A.A)) (a' : Fin q → (Tb ⟶ A'.A)), (∀ i, padd B.A (psumPt fun

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsInverseIsogenySumOfCmHomsStatement

theorem **_root_.GlobalTheory.T.exists_inverse_isogeny_sum_of_cm_homs** : ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (A A' B : Out.PolAV (Sp (𝒪 K))) (p q : ℕ) (φ : Fin p → (A.A ⟶ B.A)) (φ' : Fin q → (A'.A ⟶ B.A)), (∀ i, IsMonHom (φ i)) → (∀ i, IsMonHom (φ' i)) → B.g = p ∗ A.g + q ∗ A'.g → (∀ y : Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ B.A, ∃ (x : Fin p → (Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ A.A)) (x' : Fin q → (Over.mk (spMap (σ.comp (algebraMap (𝒪 K) K))) ⟶ A'.A)), y = (List.ofFn fun i => x i ≫ φ i).prod ∗ (List.ofFn fun i => x' i ≫ φ' i).prod) → ∃ (N : ℕ) (ψ : Fin p → (B.A ⟶ A.A)) (ψ' : Fin q → (B.A ⟶ A'.A)), 0 < N ∧ (∀ (Tb : Over (Sp (𝒪 K))) (x : Tb ⟶ B.A), padd B.A (psumPt fun i => x ≫ ψ i ≫ φ i) (psumPt fun i => x ≫ ψ' i ≫ φ' i) = pnsmulPt N x) ∧ ∀ (Tb : Over (Sp (𝒪 K))) (a : Fin p → (Tb ⟶ A.A)) (a' : Fin q → (Tb ⟶ A'.A))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIsExpPush

theorem **isHolTo_comp_hom** {V : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] {X Y : Scheme.{0}} {B : Set V} {y : V → (Sp ℂ ⟶ X)} (h : IsHolTo B y) (φ : X ⟶ Y) : IsHolTo B (fun b => y b ≫ φ)

theorem **exists_isExp_push** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (σ : R →+* ℂ) (T : PAT H (Sp R)) (T' : PAT H (Sp ℂ)) (m : SABMap T'.t T.t (spMap σ)) (D : Datum H) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (e' : (Fin g → ℂ) → (Over.mk (spMap (RingHom.id ℂ)) ⟶ T'.t.G)), IsExp T' (RingHom.id ℂ) D ty Pe e' → ∃ e : (Fin g → ℂ) → (Over.mk (spMap σ) ⟶ T.t.G), IsExp T σ D ty Pe e ∧ ∀ z, (e z).left = (e' z).left ≫ m.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIsFramePolynomialOfRuleNumberField

theorem **evR_frChg_mul_triangular** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (F : Rul Xo c) (hF : F ∈ sec Xo c a n) (z z' : FrPt Xo c K) (T : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) K) (hT : ∀ b, (T b).BlockTriangular id) (hd : ∀ b, (T b).det ≠ 0) (h : FrChg (fun b => (T b).transpose) 1 z z') : evR Xo c F z' = (∏ b, ∏ i, T b i i ^ a (b, i)) ∗ evR Xo c F z

theorem **exists_isFrPoly** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∃ G : MvPolynomial (FrVar Dw.mult) K, IsFrPoly F y G

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIsFramePolynomialOfRuleNumberFieldStatement

theorem **_root_.GlobalTheory.T.exists_isFramePolynomialOfRule_numberField** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∃ G : MvPolynomial (FrVar Dw.mult) K, IsFrPoly F y G

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIsHomFamilyBaseChange

theorem **_root_.GlobalTheory.PureType.exists_hom_over** (hβ : β ≫ B.hom = B'.hom ≫ f) (hπ : IsPullback π G'.hom G.hom f) (φ : B ⟶ G) : ∃ φ' : B' ⟶ G', φ'.left ≫ π = β ≫ φ.left

theorem **_root_.GlobalTheory.PureType.eq_of_comp_eq** (hπ : IsPullback π G'.hom G.hom f) {T : Over S'} {x y : T ⟶ G'} (h : x.left ≫ π = y.left ≫ π) : x = y

theorem **_root_.GlobalTheory.PureType.isMonHom_over** (hβ : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsGrpPullback B' B f β) (hπ : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsGrpPullback G' G f π) {φ : B ⟶ G} [IsMonHom φ] {φ' : B' ⟶ G'} (h : φ'.left ≫ π = β ≫ φ.left) : IsMonHom φ'

theorem **mul_over** (hβ : β ≫ B.hom = B'.hom ≫ f) (hπ : IsGrpPullback G' G f π) {φ₁ φ₂ : B ⟶ G} {φ₁' φ₂' : B' ⟶ G'} (h₁ : φ₁'.left ≫ π = β ≫ φ₁.left) (h₂ : φ₂'.left ≫ π = β ≫ φ₂.left) : (φ₁' ∗ φ₂').left ≫ π = β ≫ (φ₁ ∗ φ₂).left

theorem **lbIso_pow_over** {B B' G G' : Scheme.{0}} {β : B' ⟶ B} {π : G' ⟶ G} {φ : B ⟶ G} {φ' : B' ⟶ G'} (h : φ' ≫ π = β ≫ φ) {P : LB G} {P' : LB G'} {M : LB B} {M' : LB B'} (hP : P.IsLB) (hP' : P'.IsLB) (hM : M.IsLB) (hM' : M'.IsLB) (eP : LBIso P' (P.pull π)) (eM : LBIso M' (M.pull β)) {n n' : ℕ} (hI : LBIso (lbPow (P.pull φ) n) (lbPow M n')) : LBIso (lbPow (P'.pull φ') n) (lbPow M' n')

theorem **spMap_spMap** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R →+* R') (g : R' →+* R'') : spMap g ≫ spMap f = spMap (g.comp f)

theorem **IsHomFamily.baseChange** {D : Datum H} {R R' : Type} [CommRing R] [CommRing R'] {f : R →+* R'} {t : SAB H (Sp R)} {t' : SAB H (Sp R')} (m : SABMap t' t (spMap f)) {B : Over (Sp R)} {B' : Over (Sp R')} [GrpObj B] [GrpObj B'] {β : B'.left ⟶ B.left} (hβ : IsGrpPullback B' B (spMap f) β) {M : LB B.left} {M' : LB B'.left} (hM : M.IsLB) (hM' : M'.IsLB) (eM : LBIso M' (M.pull β)) {v : epsR R ⟶ B} {v' : epsR R' ⟶ B'} (hv : v'.left ≫ β = spMap (Chart.dualMapR f) ≫ v.left) {L : Submodule (𝒪 E) D.La} {ρ : R} {S : Set L} {φ : L → (B ⟶ t.G)} {φ' : L → (B' ⟶ t'.G)} (hφ : ∀ a, (φ' a).left ≫ m.φ = β ≫ (φ a).left) (h : IsHomFamily t B M v L ρ S φ) : IsHomFamily t' B' M' v' L (f ρ) S φ'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIsHomFamilyComplex

theorem **eq_const** {g : ℕ} (hg : g = 1) (z : Fin g → ℂ) (i : Fin g) : z = fun _ => z i

theorem **hermForm_of_isQuadLattice** {g : ℕ} (hg : g = 1) {Λ : Submodule ℤ (Fin g → ℂ)} {d : ℕ} (hd : 0 < d) {c : ℂ} (hc : c ≠ 0) (hΛ : IsQuadLattice Λ d c) {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (hpos : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) {Ef : LinearMap.BilinForm ℤ Λ} (hE : ∀ a b : Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) (hb : Function.Surjective Ef) (z w : ℂ) : hermForm Hm (fun _ => c ∗ z) (fun _ => c ∗ w) = 2 ∗ z ∗ starRingEnd ℂ w / Real.sqrt d

theorem **IsHomFamily.comp** {D : Datum H} {R : Type} [CommRing R] {t : SAB H (Sp R)} {B B' : Over (Sp R)} [GrpObj B] [GrpObj B'] {M : LB B.left} {M' : LB B'.left} (hM : M.IsLB) (hM' : M'.IsLB) (e : B' ⟶ B) [IsMonHom e] (he : LBIso M' (M.pull e.left)) {v : epsR R ⟶ B'} {L : Submodule (𝒪 E) D.La} {ρ : R} {S : Set L} {φ : L → (B ⟶ t.G)} (h : IsHomFamily t B M (v ≫ e) L ρ S φ) : IsHomFamily t B' M' v L ρ S (fun a => e ≫ φ a)

theorem **hermForm_mulVec** {g g' : ℕ} (M : Matrix (Fin g') (Fin g) ℂ) (Hm : Matrix (Fin g') (Fin g') ℂ) (z w : Fin g → ℂ) : hermForm (M.transpose ∗ Hm ∗ M.transpose.conjTranspose) z w = hermForm Hm (M.mulVec z) (M.mulVec w)

theorem **hermForm_line** {D : Datum H} {g g' : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) (F : (Fin g → ℂ) ≃ₗ[ℂ] (Fin g' → ℂ)) {Hm : Matrix (Fin g') (Fin g') ℂ} (hH : Hm.IsHermitian) (hpos : ∀ z : Fin g' → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) (hE : ∀ a b, (hermForm Hm (F (Pe a)) (F (Pe b))).im = D.EF a b) {ρ : ℂ} (hρ : ρ ^ 2 = -(H.d : ℂ)) (a : lat hP ρ) : (∀ m n : ℤ, ((m : ℂ) + n ∗ ((1 + Complex.I ∗ Real.sqrt H.d) / 2)) • Pe a ∈ Pe.range) ∧ ∀ z w : ℂ, hermForm Hm (F (z • Pe a)) (F (w • Pe a)) = ((D.EF (H.w0 • (a : D.La)) a).natAbs : ℂ) ∗ (2 ∗ z ∗ starRingEnd ℂ w / Real.sqrt H.d)

theorem **polarised_of_sabMap** {S : Scheme.{0}} {t : SAB H S} (hp : IsProper t.G.hom) (hA : Out.RelAmple t.G t.PP.L) (hK : Out.KTriv t.G t.PP.L) {R : Type} [CommRing R] [IsLocalRing R] {y : Sp R ⟶ S} {t' : SAB H (Sp R)} (m : SABMap t' t y) : IsAbelianScheme t'.G ∧ (∃ d, SmoothOfRelativeDimension d t'.G.hom) ∧ Out.Sym t'.G t'.PP.L ∧ Out.RelAmple t'.G t'.PP.L ∧ Out.KTriv t'.G t'.PP.L ∧ LBIso t'.PP.L (t.PP.L.pull m.φ)

theorem **isTangent_scaleVec** {R : Type} [CommRing R] {A : Over (Sp R)} [GrpObj A] (c : R) {v : epsR R ⟶ A} (hv : IsTangent v) : IsTangent (CMCurveTest.scaleVec c v)

theorem **exists_isHomFamily_complex** (hT : ThetaUnifAV) (hW : Out.WeilSt) {D : Datum H} {l : Level H} {Xo : Moduli D l} {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (FU : FamilyUnif Xo σ ty) (i : Fin FU.n) {Pe : (FU.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (FU.Di i) ty Pe) {t : SAB H (Sp ℂ)} (m : SABMap t Xo.univ.t (FU.u i Pe)) {PC : Out.PolAV (Sp ℂ)} (u₀ : HolUnif PC.A PC.L PC.g) {c : ℂ} (hg : PC.g = 1) (hu : IsThetaUnif u₀) (hc : c ≠ 0) (hΛ : IsQuadLattice u₀.Λ H.d c) {ρ : ℂ} (hρ : ρ ^ 2 = -(H.d : ℂ)) : ∃ φ : lat hP ρ → (PC.A ⟶ t.G), (∀ v, IsTangent v → IsHomFamily t PC.A PC.L v (lat hP ρ) ρ Set.univ φ) ∧ ∀ a z, (u₀.ex fun _ => c ∗ z).left ≫ (φ a).left ≫ m.φ = (FU.ex i Pe (z • Pe a)).left

theorem **cmPoint_complex** (hT : ThetaUnifAV) (hW : Out.WeilSt) (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (k : IntermediateField ℚ ℂ) (hk : KA l ≤ k) (P₀ : Out.PolAV (Sp k)) (v₀ : epsR k ⟶ P₀.A) (hv₀ : IsTangent v₀) (PC : Out.PolAV (Sp ℂ)) (ψ : PC.A.left ⟶ P₀.A.left) (u₀ : HolUnif PC.A PC.L PC.g) (c : ℂ) (hg : PC.g = 1) (hψ : IsGrpPullback PC.A P₀.A (spMap (algebraMap k ℂ)) ψ) (hL : LBIso PC.L (P₀.L.pull ψ)) (hu : IsThetaUnif u₀) (hc : c ≠ 0) (hΛ : IsQuadLattice u₀.Λ H.d c) : IsSpread FU i hP hk P₀ v₀ ψ u₀ c (fun _ => Set.univ) ℂ (AlgHom.id k ℂ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsIsPolFibreOfKeyPB

theorem **eq_dualMapR_of_algebraMap** {R R' : Type} [CommRing R] [CommRing R'] (f : R →+* R') (F : DualNumber R →+* DualNumber R') (h : ∀ a, F (algebraMap R _ a) = algebraMap R' _ (f a)) (he : F DualNumber.eps = DualNumber.eps) : F = Chart.dualMapR f

theorem **exists_isPolFibre_of_keyPB** : ∀ (R R' : Type) [CommRing R] [CommRing R'] (f : R →+* R') (x : KeyIn R) (x' : KeyIn R'), KeyPB x' x f → ∃ (φ : x'.P.A.left ⟶ x.P.A.left) (hg : x'.P.g = x.P.g), IsPolFibre f x.P x'.P φ ∧ ∀ i, IsVecOver f φ (x.v (i.cast hg)) (x'.v i)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsKroneckerCmLogPeriodEqConst

theorem **exists_kronecker_logOm_re** (E : Type) [Field E] [NumberField E] {d d' : ℕ} (h : d' = d) [Fact d.Prime] [NeZero d] (hd : d % 4 = 3) : ∃ (_ : NeZero d') (χ : DirichletCharacter ℂ d'), IsKroneckerChar χ ∧ (GlobalTheory.CM.logOm E χ).re = Real.eulerMascheroniConstant + Real.log (2 ∗ Real.pi / rootDiscr E) - (deriv (DirichletCharacter.LFunction ((quadraticChar (ZMod d)).ringHomComp (Int.castRingHom ℂ))) 1 / DirichletCharacter.LFunction ((quadraticChar (ZMod d)).ringHomComp (Int.castRingHom ℂ)) 1).re

theorem **exists_kronecker_logOm_eq** : ∀ (D : D7) (Q : Par D), Q.PinHost → ∀ large₂ : SizeData₂ D, ∃ (_ : NeZero Q.H.d) (χ : DirichletCharacter ℂ Q.H.d), IsKroneckerChar χ ∧ (GlobalTheory.CM.logOm D.host.E χ).re = large₂.logOm

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsKroneckerCmLogPeriodEqParametersConstantStatement

theorem **_root_.GlobalTheory.T.exists_kronecker_cmLogPeriod_eq** : ∀ (D : D7) (Q : Par D), Q.PinHost → ∀ large₂ : SizeData₂ D, ∃ (_ : NeZero Q.H.d) (χ : DirichletCharacter ℂ Q.H.d), IsKroneckerChar χ ∧ (GlobalTheory.CM.logOm D.host.E χ).re = large₂.logOm

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsKroneckerLogOmCore

theorem **exists_kronecker_logOm_core** : ∀ (D : D7) (T : Core D) (h : LawsCore T) (large₂ : SizeData₂ D), ∃ (_ : NeZero T.Q.H.d) (χ : DirichletCharacter ℂ T.Q.H.d), IsKroneckerChar χ ∧ (GlobalTheory.CM.logOm D.host.E χ).re = large₂.logOm

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLamLettersIsGood

theorem **exists_lieComb_image** {R R' : Type} [CommRing R] [CommRing R'] (A B : Out.PolAV (Sp R)) (x y : KeyIn R') (f : Sp R' ⟶ Sp R) (φ : x.P.A.left ⟶ A.A.left) (ψ : y.P.A.left ⟶ B.A.left) (hx : IsFibreOf A x f φ) (hy : IsFibreOf B y f ψ) (F : A.A ⟶ B.A) (hF : IsMonHom F) (i : Fin x.P.g) : ∃ (u : Fin y.P.g → R') (w : epsR R' ⟶ y.P.A), w.left ≫ ψ = (x.v i).left ≫ φ ≫ F.left ∧ IsLieComb u y.v w

theorem **exists_lamLetters_isGood** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' f : ℕ) (o : LamObj c nΨ n') (La : ℝ), LamObjGood o f La → ∃ L : LamLetters o, L.IsGood ∧ ∀ b : E →+* ℂ, (b (H.rt : E) = H.rt0 → L.m b = nΨ) ∧ (b (H.rt : E) ≠ H.rt0 → L.m b = n')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLamObjGoodOfCmObj

theorem **prod_ofFn_sum** {M : Type} [Monoid M] {p q : ℕ} (f : Fin p ⊕ Fin q → M) : (List.ofFn fun i : Fin (p + q) => f (finSumFinEquiv.symm i)).prod = (List.ofFn fun i => f (Sum.inl i)).prod * (List.ofFn fun i => f (Sum.inr i)).prod

theorem **lamEq_adj** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {S : Scheme.{0}} (t : SAB H S) (a : 𝒪 E) : LamEq t.PP.L (t.act.act a) (𝕚 t.G) t.PP.L (𝕚 t.G) (t.act.act (H.bar a))

theorem **exists_lamObjGood_of_cmObj** : Out.SecTwistSt → ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l), (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] [Algebra c.F K] (ρ : 𝒪 K) (σ₁ : K →+* ℂ) (p q : ℕ) (X : CMObj H K ρ σ₁ p q) (La : ℝ), (∀ τ, τ ∈ psiSet H ↔ labK c K τ (H.rt : E) = (ρ : K)) → (∀ j (τ : E →+* ℂ), ‖τ (X.hh j : E)‖ ≤ La ^ 2) → ∀ (xB : KeyIn K) (φB : xB.P.A.left ⟶ X.B.pol.A.left), IsFibreOf X.B.pol xB (gmap K) φB → ∃ (oa : X.B.pol.OAct (𝒪 E)) (oaΨ : (X.A true).pol.OAct (𝒪 E)) (oa' : (X.A false).pol.OAct (𝒪 E)) (xΨ x' : KeyIn K) (φΨ : xΨ.P.A.left ⟶ (X.A true).pol.A.left) (φ' : x'.P.A.left ⟶ (X.A false).pol.A.left) (τΨ : Fin xΨ.P.g ≃ {τ : E →+* ℂ // τ (H.rt : E) = H.rt0}) (τ' : Fin x'.P.g ≃ {τ : E →+* ℂ // τ (H.rt : E) ≠ H.rt0}) (ℓΨ : Fin xΨ.P.g → MLine K)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLbHermitian

theorem **lam_sum** {I : Type} (s : Finset I) (c : I → Pic A.left) (x y : T ⟶ A) : lam (∑ i ∈ s, c i) x y = ∑ i ∈ s, lam (c i) x y

theorem **lam_mul_right** (x y z : T ⟶ G) : lam (cls L hL) x (y * z) = lam (cls L hL) x y + lam (cls L hL) x z

theorem **lam_lam** (u v : A ⟶ G) [IsMonHom u] [IsMonHom v] (x y : T ⟶ A) : lam (lam (cls L hL) u v) x y = lam (cls L hL) (x ≫ u) (y ≫ v) + lam (cls L hL) (y ≫ u) (x ≫ v)

theorem **herm_expand** {O : Type} [CommRing O] (bar : O →+* O) {M : Type} [AddCommGroup M] [Module O M] (h : M → M → O) (hadd : ∀ m m' n, h (m + m') n = h m n + h m' n) (hsmul : ∀ (c : O) m n, h (c • m) n = c * h m n) (hsym : ∀ m n, h n m = bar (h m n)) (P : Summand O M) (m n : M) : h m n = ∑ i, ∑ j, P.incl m i * bar (P.incl n j) * h (P.proj (Pi.single i 1)) (P.proj (Pi.single j 1))

theorem **exists_lb_hermitian** : ∀ (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (L : LB G.left) (O : Type) [CommRing O] (bar : O →+* O) (M : Type) [AddCommGroup M] [Module O M] (h : M → M → O) [Module.Finite O M] [Module.Projective O M] (A : Over S) [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (t : IsTensor A b G a M) (w : O), L.IsLB → (∃ tθ : LB.Gen (Theta G L), tθ.IsGen) → (∀ c, LBIso (cubeLB G ![fst G G ≫ a.act c, snd G G] L) (cubeLB G ![fst G G, snd G G ≫ a.act (bar c)] L)) → ((∀ c, bar (bar c) = c) ∧ (∀ m m' n, h (m + m') n = h m n + h m' n) ∧ (∀ (c : O) m n, h (c • m) n = c * h m n) ∧ ∀ m n, h n m = bar (h m n)) → w + bar w = 1 → ∃ N : LB A.left, N.IsLB ∧ (∃ tθ : LB.Gen (Theta A N), tθ.IsGen) ∧ (∀ m n, LBIso (cubeLB A ![fst G G ≫ t.tmul m, snd G G ≫ t.tmul n] N) (cubeLB G ![fst G G ≫ a.act (h m n), snd G G] L))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLeftInverse

theorem **exists_leftInverse** {O : Type} [CommRing O] (bar : O →+* O) {M : Type} [AddCommGroup M] [Module O M] (h : M → M → O) (hadd : ∀ m m' n, h (m + m') n = h m n + h m' n) (hsmul : ∀ (c : O) m n, h (c • m) n = c * h m n) (hsym : ∀ m n, h n m = bar (h m n)) (hperf : ∀ l : M →ₗ[O] O, ∃! n, ∀ m, l m = h m n) {n k : ℕ} (l : Fin k → (M →ₗ[O] O)) (hl : ∀ m m', (n : O) * h m m' = ∑ i, l i m * bar (l i m')) : ∃ r : (Fin k → O) →ₗ[O] M, ∀ m, r (LinearMap.pi l m) = n • m

theorem **lam_self** (laws : Out.Laws) {S : Scheme.{0}} [IsLocallyNoetherian S] (P : Out.PolAV S) {T : Over S} (x : T ⟶ P.A) : lam (cls P.L P.isLB) x x = 2 • pull x.left (cls P.L P.isLB)

theorem **relAmple_tensor** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) [IsCommMonObj P.A] (O : Type) [CommRing O] (bar : O →+* O) (M : Type) [AddCommGroup M] [Module O M] (h : M → M → O) [Module.Finite O M] [Module.Projective O M] (A : Over S) [GrpObj A] [IsCommMonObj A] (a : OAct O P.A) (b : OAct O A) (t : IsTensor A b P.A a M) (w : O) (N N' : LB A.left), N.IsLB → N'.IsLB → (∃ tθ : LB.Gen (Theta A N), tθ.IsGen) → (∀ m n, LBIso (cubeLB A ![fst P.A P.A ≫ t.tmul m, snd P.A P.A ≫ t.tmul n] N) (cubeLB P.A ![fst P.A P.A ≫ a.act (h m n), snd P.A P.A] P.L)) → (∀ c, LBIso (cubeLB P.A ![fst P.A P.A ≫ a.act c, snd P.A P.A] P.L) (cubeLB P.A ![fst P.A P.A, snd P.A P.A ≫ a.act (bar c)] P.L)) → ((∀ c, bar (bar c) = c) ∧ (∀ m m' n, h (m + m') n = h m n + h m' n) ∧ (∀ (c : O) m n, h (c • m) n = c * h m n) ∧ ∀ m n, h n m = bar (h m n)) → (∀ l : M →ₗ[O] O, ∃! n, ∀ m, l m = h m n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLieBasisLocalModel

theorem **exists_restrict_ringHom** {R K V F : Type} [CommRing R] [CommRing K] [CommRing V] [CommRing F] (a : R →+* K) (b : V →+* F) (hb : Function.Injective b) (i : K →+* F) (h : ∀ r : R, i (a r) ∈ Set.range b) : ∃ ψ : R →+* V, b.comp ψ = i.comp a

theorem **exists_isLieBasis_locMod** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (p : HeightOneSpectrum (𝒪 z.K)) (T : CDVR) (iK : z.K →+* T.Fr) (M : LocMod z T iK), (∀ y : z.K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation z.K y ≤ 1) → ∀ A : Matrix (Fin z.Φ.T.g) (Fin z.Φ.T.g) z.K, LamF.IsModelAt z.B.pol z.Φ.key z.m'.φ p A → ∃ vV : Fin z.Φ.T.g → (epsR T.V ⟶ M.G.t.G), IsLieBasis vV ∧ ∀ k, ∃ wK : epsR z.K ⟶ z.Φ.T.t.G, IsLieComb (A k) z.Φ.v wK ∧ spMap (Formalism.L2.Pj.dualMap (algebraMap T.V T.Fr)) ≫ (vV k).left ≫ M.n.φ = spMap (Formalism.L2.Pj.dualMap iK) ≫ wK.left ≫ z.m'.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLieBasisLocalModelStatement

theorem **_root_.GlobalTheory.T.exists_lieBasis_local_model** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (p : HeightOneSpectrum (𝒪 z.K)) (T : CDVR) (iK

: z.K →+* T.Fr) (M : LocMod z T iK), (∀ y : z.K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation z.K y ≤ 1) → ∀ A : Matrix (Fin z.Φ.T.g) (Fin z.Φ.T.g) z.K, LamF.IsModelAt z.B.pol z.Φ.key z.m'.φ p A → ∃ vV : Fin z.Φ.T.g → (epsR T.V ⟶ M.G.t.G), IsLieBasis vV ∧ ∀ k, ∃ wK : epsR z.K ⟶ z.Φ.T.t.G, IsLieComb (A k) z.Φ.v wK ∧ spMap (Formalism.L2.Pj.dualMap (algebraMap T.V T.Fr)) ≫ (vV k).left ≫ M.n.φ = spMap (Formalism.L2.Pj.dualMap iK) ≫ wK.left ≫ z.m'.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLieEigenbasisGenericFibreCmTriple

theorem **exists_eigenbasis_fibre** : Out.SecTwistSt → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (K : Type) [Field K] [NumberField K] (T : PAT H (Sp (𝒪 K))) (oa : T.pol.OAct (𝒪 E)) (S : Set (E →+* ℂ)) (lab : (E →+* ℂ) → (E →+* K)) (ρ : 𝒪 K), oa.act = T.t.act.act → Function.Bijective lab → ρ ^ 2 = -(H.d : 𝒪 K) → (∀ t, t ∈ S ↔ lab t (H.rt : E) = (ρ : K)) → HasType H T.t.act ρ 1 0 → ∃ (x : KeyIn K) (φ : x.P.A.left ⟶ T.pol.A.left) (τ : Fin x.P.g ≃ S), IsFibreOf T.pol x (gmap K) φ ∧ x.P.g = Module.finrank ℚ E / 2 ∧ IsLieType T.pol oa x φ τ lab

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLieEigenbasisGenericFibreCmTripleStatement

theorem **_root_.GlobalTheory.T.exists_lie_eigenbasis_genericFibre_cmTriple** : Out.SecTwistSt → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (K : Type) [Field K] [NumberField K] (T : PAT H (Sp (𝒪 K))) (oa : T.pol.OAct (𝒪 E)) (S : Set (E →+* ℂ)) (lab : (E →+* ℂ) → (E →+* K)) (ρ : 𝒪 K), oa.act = T.t.act.act → Function.Bijective lab → ρ ^ 2 = -(H.d : 𝒪 K) → (∀ t, t ∈ S ↔ lab t (H.rt : E) = (ρ : K)) → HasType H T.t.act ρ 1 0 → ∃ (x : KeyIn K) (φ : x.P.A.left ⟶ T.pol.A.left) (τ : Fin x.P.g ≃ S), IsFibreOf T.pol x (gmap K) φ ∧ x.P.g = Module.finrank ℚ E / 2 ∧ IsLieType T.pol oa x φ τ lab

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLieRepresentationEigenLattices

theorem **_root_.GlobalTheory.CM.GenFib.exists_lieRep_eigLat** (K : Type) [Field K] [NumberField K] (P : _root_.ReciprocityTheory.ChartRestriction.G.Out.PolAV (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp (𝒪 K))) (F : _root_.GlobalTheory.CM.GenFib P) (O : Type) [CommRing O] (act : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.OAct O P.A) : ∃ r : O →+* Module.End K (Fin F.P₀.g → K), (∀ a, F.IsDiffOf F (act.act a) (r a)) ∧ ∀ (τ : O →+* 𝒪 K) c, (c ∈ F.eigLat act τ ↔ c ∈ F.lieLat ∧ ∀ a, r a c = τ a • c) ∧ (c ∈ Submodule.span K (F.eigLat act τ : Set (Fin F.P₀.g → K)) ↔ ∀ a, r a c = τ a • c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLieRepresentationEigenLatticesStatement

theorem **_root_.GlobalTheory.T.exists_lieRepresentation_eigenLattices** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (O : Type) [CommRing O] (act : OAct O P.A), ∃ r : O →+* Module.End K (Fin F.P₀.g → K), (∀ a, F.IsDiffOf F (act.act a) (r a)) ∧ ∀ (τ : O →+* 𝒪 K) c, (c ∈ F.eigLat act τ ↔ c ∈ F.lieLat ∧ ∀ a, r a c = τ a • c) ∧ (c ∈ Submodule.span K (F.eigLat act τ : Set (Fin F.P₀.g → K)) ↔ ∀ a, r a c = τ a • c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsLinearOfIsMonHom

theorem **valAt'_comp_left** (x : 𝟙_ (Over (Sp ℂ)) ⟶ A) (f : A ⟶ A') (U : A'.left.Opens) (s : Γ(A'.left, U)) : valAt' (x ≫ f).left U s = valAt' x.left (f.left ⁻¹ᵁ U) ((f.left.app U).hom s)

theorem **isHolTo_ex_comp** (u : HolUnif A L g) (f : A ⟶ A') : IsHolTo (Set.univ : Set (Fin g → ℂ)) (fun z => (u.ex z ≫ f).left)

theorem **isTangent_comp** {R : Type} [CommRing R] {G G' : Over (Sp R)} [GrpObj G] [GrpObj G'] (f : G ⟶ G') [IsMonHom f] {v : epsR R ⟶ G} (hv : IsTangent v) : IsTangent (v ≫ f)

theorem **top_le_ex_zero** (u : HolUnif A L g) {v : epsR ℂ ⟶ A} (hv : IsTangent v) (U : A.left.Opens) (h : ⊤ ≤ v.left ⁻¹ᵁ U) : ⊤ ≤ (u.ex 0).left ⁻¹ᵁ U

theorem **isDiff_comp** (u : HolUnif A L g) (u' : HolUnif A' L' g') (f : A ⟶ A') [IsMonHom f] (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (hF : ∀ z, u'.ex (F z) = u.ex z ≫ f) {w : Fin g → ℂ} {v : epsR ℂ ⟶ A} (hv : IsTangent v) (hd : u.IsDiff w v) : u'.IsDiff (F w) (v ≫ f)

theorem **exists_linear_of_isMonHom** : ∀ (A A' : Over (Sp ℂ)) [GrpObj A] [GrpObj A'], IsAbelianScheme A → IsAbelianScheme A' → ∀ (L : LB A.left) (L' : LB A'.left) (g g' : ℕ) (u : HolUnif A L g) (u' : HolUnif A' L' g'), (∀ v, IsTangent v → ∃! w, u'.IsDiff w v) → (∀ w, ∃! v, IsTangent v ∧ u'.IsDiff w v) → ∀ f : A ⟶ A', IsMonHom f → ∃ F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ), (∀ z, u'.ex (F z) = u.ex z ≫ f) ∧ (∀ l ∈ u.Λ, F l ∈ u'.Λ) ∧ (∀ w v, IsTangent v → u.IsDiff w v → IsTangent (v ≫ f) ∧ u'.IsDiff (F w) (v ≫ f)) ∧ ∀ f' : A ⟶ A', (∀ z, u'.ex (F z) = u.ex z ≫ f') → f' = f

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsMetricDataCmPointComplexEmbedding

theorem **dualMap_eq_dualMapR** {S T : Type} [CommRing S] [CommRing T] (φ : S →+* T) : Formalism.L2.Pj.dualMap φ = Chart.dualMapR φ

theorem **exists_fibre** {K : Type} [Field K] (σ : K →+* ℂ) {x x₀ : Sp K ⟶ Xo.X} (hx : x₀ = x) (Φ : FibKey Xo K x) (s : LB.Sec (PPx x₀)) (hs : SecTie Φ.m Φ.th s) (x' : Sp ℂ ⟶ Xo.X) (s' : LB.Sec (PPx x')) (hg : SecGen x' s') (hx' : x' = spMap σ ≫ x₀) (hb : SecBC σ 1 x₀ x' s s') (hpq : 1 ≤ Dw.p + Dw.q) : ∃ Φ' : FibKey Xo ℂ x', SecTie Φ'.m Φ'.th s' ∧ KeyPB Φ'.key Φ.key σ ∧ ∀ (e : Φ'.T.g = Φ.T.g) (cf : Fin Φ.T.g → K) (w : epsR K ⟶ Φ.T.t.G), IsLieComb cf Φ.v w → ∃ w' : epsR ℂ ⟶ Φ'.T.t.G, IsLieComb (fun i => σ (cf i)) (fun i => Φ'.v (i.cast e.symm)) w' ∧ w'.left ≫ Φ'.m.φ = spMap (Formalism.L2.Pj.dualMap σ) ≫ w.left ≫ Φ.m.φ

theorem **exists_unif_of_eq** (x : KeyIn ℂ) {g : ℕ} (hg : g = x.P.g) (u : HolUnif x.P.A x.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)) (N : ℝ) (h : u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume ∧ N = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal) : ∃ (u : HolUnif x.P.A x.P.L x.P.g) (Hm : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin x.P.g → ℂ) → ℂ) (Fd : Set (Fin x.P.g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume ∧ N = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal

theorem **exists_unif** {K : Type} [Field K] [NumberField K] {x : KeyIn K} {T abelian₂ : MLine K} {σ : K →+* ℂ} (h : KeyArchAt x T abelian₂ σ) (x' : KeyIn ℂ) (hx : KeyPB x' x σ) : ∃ (u : HolUnif x'.P.A x'.P.L x'.P.g) (Hm : Matrix (Fin x'.P.g) (Fin x'.P.g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin x'.P.g → ℂ) → ℂ) (Fd : Set (Fin x'.P.g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x'.rig ∧ u.IsFn x'.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume ∧ T.n σ ^ 2 = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal

theorem **form_eq_gram** {K : Type} [Field K] [NumberField K] {n : ℕ} (h : ReciprocityTheory.GeometryOfNumbersTheory.HermFam K (Fin n → K)) (s : K →+* ℂ) (a b : Fin n → K) : star (ReciprocityTheory.MultiplicativeHeights.embVec s a) ⬝ᵥ (h.gram s (fun i => Pi.single i 1)).mulVec (ReciprocityTheory.MultiplicativeHeights.embVec s b) = h.form s a b

theorem **exists_isMetric** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (σ : z.K →+* ℂ) (yσ : FrPt Xo c (CAt (σ.comp (algebraMap c.F z.K)))), FrMap (embAlg σ) z.y yσ → ∃ Hm : Matrix (Fin z.Φ.T.g) (Fin z.Φ.T.g) ℂ, Hm.PosDef ∧ z.LamL.n σ = Real.sqrt Hm.det.re ∧ (∀ j, ReciprocityTheory.MultiplicativeHeights.hq Hm (ReciprocityTheory.MultiplicativeHeights.embVec σ (z.U j)) ≤ (La * (z.ℓ j).n σ) ^ 2) ∧ IsMetric yσ (z.Tl.n σ ^ 2) (gramBlock z.e z.U Hm σ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsMetricDataCmPointComplexEmbeddingStatement

theorem **_root_.GlobalTheory.T.exists_metricData_cmPoint_complexEmbedding** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (σ : z.K →+* ℂ) (yσ : FrPt Xo c (CAt (σ.comp (algebraMap c.F z.K)))), FrMap (embAlg σ) z.y yσ → ∃ Hm : Matrix (Fin z.Φ.T.g) (Fin z.Φ.T.g) ℂ, Hm.PosDef ∧ z.LamL.n σ = Real.sqrt Hm.det.re ∧ (∀ j, ReciprocityTheory.MultiplicativeHeights.hq Hm (ReciprocityTheory.MultiplicativeHeights.embVec σ (z.U j)) ≤ (La * (z.ℓ j).n σ) ^ 2) ∧ IsMetric yσ (z.Tl.n σ ^ 2) (gramBlock z.e z.U Hm σ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsMetrisedLineOfLieVector

theorem **exists_fractionalIdeal_single_mem** {K : Type} [Field K] [NumberField K] {n : ℕ} (Λ : Submodule (𝒪 K) (Fin n → K)) (hfg : Λ.FG) (hsp : Submodule.span K (Λ : Set (Fin n → K)) = ⊤) (i : Fin n) : ∃ I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K, I ≠ 0 ∧ ∀ c : K, c ∈ I ↔ Pi.single i c ∈ Λ

theorem **exists_length_of_lieVec** (hU : ThetaUnifAV) {K : Type} [Field K] (x : KeyIn K) (i : Fin x.P.g) (σ : K →+* ℂ) : ∃ r : ℝ, 0 < r ∧ ∃ (xσ : KeyIn ℂ) (u : HolUnif xσ.P.A xσ.P.L xσ.P.g) (Hm : Matrix (Fin xσ.P.g) (Fin xσ.P.g) ℂ) (χ : u.Λ → ℂ) (W : Fin xσ.P.g → ℂ) (hg : xσ.P.g = x.P.g), KeyPB xσ x σ ∧ u.IsAH Hm χ ∧ u.IsDiff W (xσ.v (i.cast hg.symm)) ∧ r ^ 2 = Real.pi * (W ⬝ᵥ Hm.mulVec (star W)).re

theorem **exists_isVecLine** : Out.SecTwistSt → ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left), IsFibreOf A x (gmap K) φ → ∀ i : Fin x.P.g, ∃ ℓ : MLine K, IsVecLine A x φ i ℓ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsMetrisedLineOfLieVectorStatement

theorem **_root_.GlobalTheory.T.exists_metrised_line_of_lieVector** : Out.SecTwistSt → ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left), IsFibreOf A x (gmap K) φ → ∀ i : Fin x.P.g, ∃ ℓ : MLine K, IsVecLine A x φ i ℓ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsMonHomFactorOfFlatSurjective

theorem **_root_.GlobalTheory.exists_monHom_factor_of_flat_surjective** : ∀ (S : Scheme.{0}) (G G' G'' : Over S) [GrpObj G] [GrpObj G'] [GrpObj G''] (Φ : G ⟶ G') (f : G ⟶ G''), IsMonHom Φ → IsMonHom f → Flat Φ.left → Surjective Φ.left → QuasiCompact Φ.left → (∀ (T : Over S) (x : T ⟶ G), x ≫ Φ = 1 → x ≫ f = 1) → (∃ ψ : G' ⟶ G'', IsMonHom ψ ∧ Φ ≫ ψ = f) ∧ Epi Φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsMulEquivPointsOfIsGrpPullback

theorem **_root_.GlobalTheory.LamF.exists_mulEquiv_points_of_isGrpPullback** : ∀ (S S' : Scheme.{0}) (f : S' ⟶ S) (G : Over S) (G' : Over S') [GrpObj G] [GrpObj G'] (φ : G'.left ⟶ G.left), _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsGrpPullback G' G f φ → ∀ T : Over S', ∃ Φ : (T ⟶ G') ≃* (Over.mk (T.hom ≫ f) ⟶ G), ∀ x, (Φ x).left = x.left ≫ φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNormalisedPolarisation

theorem **exists_normalised_polarisation** : Out.Laws → ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (P : Out.PolAV (Sp R)) (w w' : P.A ⟶ P.A), IsMonHom w → IsMonHom w' → w ∗ w' = 𝟙 P.A → LamEq P.L w (𝟙 P.A) P.L (𝟙 P.A) w' → ∃ L' : LB P.A.left, L'.IsLB ∧ Out.Rigidified P.A L' ∧ Out.Sym P.A L' ∧ Out.RelAmple P.A L' ∧ Out.KTriv P.A L' ∧ LamEq L' (𝟙 P.A) (𝟙 P.A) P.L (𝟙 P.A) (𝟙 P.A) ∧ LBIso L' (cubeLB P.A ![𝟙 P.A, w] L')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNormalisedPolarisationStatement

theorem **_root_.GlobalTheory.T.exists_normalized_symmetric_ample_lb** : Out.Laws → ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (P : Out.PolAV (Sp R)) (w w' : P.A ⟶ P.A), IsMonHom w → IsMonHom w' → w ∗ w' = 𝟙 P.A → LamEq P.L w (𝟙 P.A) P.L (𝟙 P.A) w' → ∃ L' : LB P.A.left, L'.IsLB ∧ Out.Rigidified P.A L' ∧ Out.Sym P.A L' ∧ Out.RelAmple P.A L' ∧ Out.KTriv P.A L' ∧ LamEq L' (𝟙 P.A) (𝟙 P.A) P.L (𝟙 P.A) (𝟙 P.A) ∧ LBIso L' (cubeLB P.A ![𝟙 P.A, w] L')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNumberFieldDefiningCmCurveHoms

theorem **Curve.isTangent** {R : Type} [CommRing R] {P : Out.PolAV (Sp R)} {v : epsR R ⟶ P.A} {Lf : (R →+* ℂ) → PeriodPair} (h : Curve P v Lf) : IsTangent v

theorem **HomOf.up** (hB : Out.PolBaseChangeSt) {R R₀ R₁ : Type} [CommRing R] [CommRing R₀] [CommRing R₁] {ρ : R →+* R₀} {P : Out.PolAV (Sp R)} {v : epsR R ⟶ P.A} {Q : Out.PolAV (Sp R)} {t : epsR R ⟶ Q.A} {a : R₀} (h : HomOf ρ P v Q t a) (g : R₀ →+* R₁) : HomOf (g.comp ρ) P v Q t (g a)

theorem **HomOf.fibre** {R R₀ R₁ : Type} [CommRing R] [CommRing R₀] [CommRing R₁] {f : R →+* R₀} {g : R₀ →+* R₁} {P : Out.PolAV (Sp R)} {v : epsR R ⟶ P.A} {Q : Out.PolAV (Sp R)} {t : epsR R ⟶ Q.A} {a : R₁} (h : HomOf (g.comp f) P v Q t a) {P₀ Q₀ : Out.PolAV (Sp R₀)} {φ : P₀.A.left ⟶ P.A.left} {ψ : Q₀.A.left ⟶ Q.A.left} {v₀ : epsR R₀ ⟶ P₀.A} {t₀ : epsR R₀ ⟶ Q₀.A} (hφ : IsPolFibre f P P₀ φ) (hψ : IsPolFibre f Q Q₀ ψ) (hv : IsVecOver f φ v v₀) (ht : IsVecOver f ψ t t₀) : HomOf g P₀ v₀ Q₀ t₀ a

theorem **multipliers_fg** (L L' : PeriodPair) : (multipliers L L').FG

theorem **DefinedOver.mono** (hB : Out.PolBaseChangeSt) {R : Type} [CommRing R] {k : IntermediateField ℚ ℂ} {ρ : R →+* k} {P : Out.PolAV (Sp R)} {v : epsR R ⟶ P.A} {Q : Out.PolAV (Sp R)} {t : epsR R ⟶ Q.A} {N N' : IntermediateField ℚ ℂ} {α : ℂ} (h : DefinedOver ρ P v Q t N α) (hN : N ≤ N') : DefinedOver ρ P v Q t N' α

theorem **exists_field** (hU : ThetaUnifAV) (hB : Out.PolBaseChangeSt) {R : Type} [CommRing R] {k : IntermediateField ℚ ℂ} [FiniteDimensional ℚ k] (ρ : R →+* k) {P : Out.PolAV (Sp R)} {v : epsR R ⟶ P.A} {Q : Out.PolAV (Sp R)} {t : epsR R ⟶ Q.A} {L L' : (R →+* ℂ) → PeriodPair} (hP : Curve P v L) (hQ : Curve Q t L') : ∃ N : IntermediateField ℚ ℂ, FiniteDimensional ℚ N ∧ k ≤ N ∧ ∀ α : ℂ, (∀ z ∈ (L ((algebraMap k ℂ).comp ρ)).lattice, α * z ∈ (L' ((algebraMap k ℂ).comp ρ)).lattice) → DefinedOver ρ P v Q t N α

theorem **exists_common_field** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ ComplexEmbedding.IsReal σ₀ → ∀ (𝔟 : ClassGroup (𝒪 K) → nonZeroDivisors (Ideal (𝒪 K))), (∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (κ : K →+* M) (P : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A) (L₀ : (𝒪 M →+* ℂ) → PeriodPair) (Q : ClassGroup (𝒪 K) → Out.PolAV (Sp (𝒪 M))) (t : ∀ C, epsR (𝒪 M) ⟶ (Q C).A) (L : ClassGroup (𝒪 K) → (𝒪 M →+* ℂ) → PeriodPair) (δ : ClassGroup (𝒪 K) → (𝒪 M)ˣ), (∀ a, (κ a : ℂ) = σ₀ a) ∧ (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : (𝒪 M) →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : (𝒪 M) →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUn

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNumberFieldDefiningCmCurveHomsStatement

theorem **_root_.GlobalTheory.T.exists_numberField_defining_cm_curve_homs** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ ComplexEmbedding.IsReal σ₀ → ∀ (𝔟 : ClassGroup (𝒪 K) → nonZeroDivisors (Ideal (𝒪 K))), (∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (κ : K →+* M) (P : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A) (L₀ : (𝒪 M →+* ℂ) → PeriodPair) (Q : ClassGroup (𝒪 K) → Out.PolAV (Sp (𝒪 M))) (t : ∀ C, epsR (𝒪 M) ⟶ (Q C).A) (L : ClassGroup (𝒪 K) → (𝒪 M →+* ℂ) → PeriodPair) (δ : ClassGroup (𝒪 K) → (𝒪 M)ˣ), (∀ a, (κ a : ℂ) = σ₀ a) ∧ (P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : (𝒪 M) →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ ∀ s' : (𝒪 M) →+* ℂ, (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNumberFieldPointOfCmPeriodLattice

theorem **prod_left** (h : IsGrpPullback G' G f φ) {t' : X ⟶ S'} {t : X ⟶ S} (e : t' ≫ f = t) {n : ℕ} (x' : Fin n → (Over.mk t' ⟶ G')) (x : Fin n → (Over.mk t ⟶ G)) (hx : ∀ i, (x' i).left ≫ φ = (x i).left) : (List.ofFn x').prod.left ≫ φ = (List.ofFn x).prod.left

theorem **exists_lift_pt** (h : IsPullback φ G'.hom G.hom f) {t' : X ⟶ S'} (x : X ⟶ G.left) (e : x ≫ G.hom = t' ≫ f) : ∃ x' : Over.mk t' ⟶ G', x'.left ≫ φ = x

theorem **pt_ext** (h : IsPullback φ G'.hom G.hom f) {T : Over S'} {x y : T ⟶ G'} (e : x.left ≫ φ = y.left ≫ φ) : x = y

theorem **exists_lift_hom** {H : Over S} {H' : Over S'} [GrpObj H] [GrpObj H'] {ψ : H'.left ⟶ H.left} (hG : IsGrpPullback G' G f φ) (hH : IsGrpPullback H' H f ψ) (w : G ⟶ H) [IsMonHom w] : ∃ w' : G' ⟶ H', IsMonHom w' ∧ w'.left ≫ ψ = φ ≫ w.left

theorem **onto_lift** {B : Over S} {B' : Over S'} {β : B'.left ⟶ B.left} (hB : IsPullback β B'.hom B.hom f) (hG : IsGrpPullback G' G f φ) {n : ℕ} (w : Fin n → (B ⟶ G)) (w' : Fin n → (B' ⟶ G')) (hw : ∀ i, (w' i).left ≫ φ = β ≫ (w i).left) {t' : X ⟶ S'} {t : X ⟶ S} (e : t' ≫ f = t) (h : ∀ y : Over.mk t ⟶ G, ∃ x : Fin n → (Over.mk t ⟶ B), y = (List.ofFn fun i => x i ≫ w i).prod) (y : Over.mk t' ⟶ G') : ∃ x : Fin n → (Over.mk t' ⟶ B'), y = (List.ofFn fun i => x i ≫ w' i).prod

theorem **isAbelianScheme_pow** {S : Scheme.{0}} (G : Over S) [GrpObj G] (hG : IsAbelianScheme G) : ∀ n : ℕ, IsAbelianScheme (Out.powObj G n) | 0 => by have

theorem **onto_of_onto** {R : Type} [CommRing R] [IsDomain R] (σ : R →+* ℂ) (hσ : Function.Injective σ) {A B : Over (Sp R)} [GrpObj A] [GrpObj B] (hA : IsAbelianScheme A) (hB : IsAbelianScheme B) {n : ℕ} (w : Fin n → (B ⟶ A)) (h : ∀ y : Over.mk (spMap σ) ⟶ A, ∃ x : Fin n → (Over.mk (spMap σ) ⟶ B), y = (List.ofFn fun i => x i ≫ w i).prod) (t : Sp ℂ ⟶ Sp R) (y : Over.mk t ⟶ A) : ∃ x : Fin n → (Over.mk t ⟶ B), y = (List.ofFn fun i => x i ≫ w i).prod

theorem **map_sum_eq_prod** {V Q : Type} [AddCommGroup V] [Group Q] (e : V → Q) (he : ∀ z w, e (z + w) = e z * e w) {n : ℕ} (z : Fin n → V) : e (∑ i, z i) = (List.ofFn fun i => e (z i)).prod

theorem **exists_numberField_point_of_cmLat** : CMCurve → CMPoint → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (p q : ℕ), Nonempty (CMLat hP p q) → ∃ (K₁ : IntermediateField ℚ ℂ) (hk : KA l ≤ K₁) (y : Sp K₁ ⟶ Xo.X), FiniteDimensional ℚ K₁ ∧ y ≫ Xo.s = spMap (IntermediateField.inclusion hk).toRingHom ∧ spMap (algebraMap K₁ ℂ) ≫ y = FU.u i Pe

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNumberFieldPointOfCmPeriodLatticeStatement

theorem **_root_.GlobalTheory.T.exists_numberField_point_of_cmPeriodLattice** : CMCurve → CMPoint → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo (algebraMap (KA l) ℂ) ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)) (hP : IsPeriod (FU.Di i) ty Pe) (p q : ℕ), Nonempty (CMLat hP p q) → ∃ (K₁ : IntermediateField ℚ ℂ) (hk : KA l ≤ K₁) (y : Sp K₁ ⟶ Xo.X), FiniteDimensional ℚ K₁ ∧ y ≫ Xo.s = spMap (IntermediateField.inclusion hk).toRingHom ∧ spMap (algebraMap K₁ ℂ) ≫ y = FU.u i Pe

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNumberFieldPointOfCountable

theorem **exists_numberField_factor** (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] {k₀ A : Type} [Field k₀] [CommRing A] (j : k₀ →+* k) (α : k₀ →+* A) (hα : α.FiniteType) (ψ : A →+* ℂ) (hψ : ∀ a, ψ (α a) = (j a : ℂ)) (halg : ∀ c, IsAlgebraic k (ψ c)) : ∃ (K₁ : IntermediateField ℚ ℂ) (hk : k ≤ K₁) (ψ₁ : A →+* K₁), FiniteDimensional ℚ K₁ ∧ ψ₁.comp α = (IntermediateField.inclusion hk).toRingHom.comp j ∧ (algebraMap K₁ ℂ).comp ψ₁ = ψ

theorem **exists_fg_le_range_subset** {k : Type} [Field k] [Algebra k ℂ] (R : Subalgebra k ℂ) (hR : R.FG) {X : Scheme.{0}} (y : Sp R ⟶ X) (U : X.Opens) (hU : (spMap R.val.toRingHom ≫ y) (IsLocalRing.closedPoint ℂ) ∈ U) : ∃ (R' : Subalgebra k ℂ) (h : R ≤ R'), R'.FG ∧ Set.range (spMap (Subalgebra.inclusion h).toRingHom ≫ y) ⊆ (U : Set X)

theorem **exists_numberField_point_of_countable** : ∀ (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (R : Subalgebra k ℂ), R.FG → ∀ (k₀ : Type) [Field k₀] (j : k₀ →+* k) (X : Scheme.{0}) (s : X ⟶ Sp k₀) [LocallyOfFiniteType s] (y : Sp R ⟶ X), y ≫ s = spMap ((algebraMap k R).comp j) → Set.Countable (Set.range fun ρ : R →ₐ[k] ℂ => spMap ρ.toRingHom ≫ y) → ∃ (K₁ : IntermediateField ℚ ℂ) (hk : k ≤ K₁) (y₁ : Sp K₁ ⟶ X), FiniteDimensional ℚ K₁ ∧ y₁ ≫ s = spMap ((IntermediateField.inclusion hk).toRingHom.comp j) ∧ spMap (algebraMap K₁ ℂ) ≫ y₁ = spMap R.val.toRingHom ≫ y

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsNumberFieldTorsionPoints

theorem **exists_field_alg** (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (B : Type) [CommRing B] [Algebra k B] [Module.Finite k B] : ∃ k₁ : IntermediateField ℚ ℂ, FiniteDimensional ℚ k₁ ∧ k ≤ k₁ ∧ ∀ (φ : B →ₐ[k] ℂ) (b : B), φ b ∈ k₁

theorem **exists_field_finite** (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (Z : Over (Sp k)) [IsFinite Z.hom] : ∃ k₁ : IntermediateField ℚ ℂ, FiniteDimensional ℚ k₁ ∧ k ≤ k₁ ∧ ∀ (k₂ : IntermediateField ℚ ℂ) (h₂ : k ≤ k₂), k₁ ≤ k₂ → ∀ y : Over.mk (spMap (algebraMap k ℂ)) ⟶ Z, ∃ y₂ : Over.mk (spMap (IntermediateField.inclusion h₂).toRingHom) ⟶ Z, spMap (algebraMap k₂ ℂ) ≫ y₂.left = y.left

theorem **exists_numberField_torsion_points** : Out.Laws → ∀ (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (P : Out.PolAV (Sp k)) (n : ℕ), 0 < n → ∃ k₁ : IntermediateField ℚ ℂ, FiniteDimensional ℚ k₁ ∧ k ≤ k₁ ∧ ∀ (k₂ : IntermediateField ℚ ℂ) (h₂ : k ≤ k₂), k₁ ≤ k₂ → ∀ x : Over.mk (spMap (algebraMap k ℂ)) ⟶ P.A, x ^ n = 1 → ∃! x₂ : Over.mk (spMap (IntermediateField.inclusion h₂).toRingHom) ⟶ P.A, spMap (algebraMap k₂ ℂ) ≫ x₂.left = x.left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsOActExtendDedekind

theorem **exists_oAct_extend_dedekind** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (O

: Type) [CommRing O] (G : Over (Sp R)) [GrpObj G] (G₀ : Over (Sp K)) [GrpObj G₀] (φ : G₀.left ⟶ G.left), IsAbelianScheme G → IsGrpPullback G₀ G (spMap (algebraMap R K)) φ → ∀ a₀ : OAct O G₀, ∃ a : OAct O G, ∀ c, (a₀.act c).left ≫ φ = φ ≫ (a.act c).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsOfCmTriplePullback

theorem **PAT.exists_of_sabMap** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S S' : Scheme.{0}) (f : S' ⟶ S) (T : PAT H S) (t' : SAB H S') (m : SABMap t' T.t f), (∃ T' : PAT H S', T'.t = t' ∧ T'.g = T.g) ∧ LBIso (T.t.PP.L.pull m.φ) t'.PP.L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsOfCmTriplePullbackStatement

theorem **_root_.GlobalTheory.T.exists_of_cmTriple_pullback** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S S' : Scheme.{0}) (f : S' ⟶ S) (T : PAT H S) (t' : SAB H S') (m : SABMap t' T.t f), (∃ T' : PAT H S', T'.t = t' ∧ T'.g = T.g) ∧ LBIso (T.t.PP.L.pull m.φ) t'.PP.L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsOrthonormalFrameBorelChangeComplexFramedPo

theorem **lact_one_frame** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} {Ψ : Finset (E →+* ℂ)} (y : FD.Frame Ψ r) : FD.lact (1 : GL (Fin r) E) y = y

theorem **exists_ortho_frBor** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r), IsUnf U → ∀ y : FrPt Xo c (CAt σ), ∃ (k : U.Comp) (x : (U.X k).dom) (t : Ix Dw → (CAt σ)ˣ), (U.X k).IsOrtho x.1 ∧ FrBor t (U.un k x) y

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsOrthonormalFrameBorelChangeComplexFramedPoStatement

theorem **_root_.GlobalTheory.T.exists_orthonormalFrame_borelChange_complexFramedPoint** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r), IsUnf U → ∀ y : FrPt Xo c (CAt σ), ∃ (k : U.Comp) (x : (U.X k).dom) (t : Ix Dw → (CAt σ)ˣ), (U.X k).IsOrtho x.1 ∧ FrBor t (U.un k x) y

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPatOfObj

theorem **exists_pat_of_obj** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (B : Type) [CommRing B] (b : Base l B) (S : Scheme.{0}) (s : S ⟶ Sp B) (o : Obj D l b s), ∃ T : PAT H S, T.t = o.t ∧ T.g = Module.finrank ℚ E / 2 * (D.p + D.q)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPatOfPolAV

theorem **exists_pat_of_polAV** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsNoetherianRing R] (P : Out.PolAV (Sp R)) (act : OAct (𝒪 E) P.A), (∀ a, LamEq P.L (act.act a) (𝟙 P.A) P.L (𝟙 P.A) (act.act (H.bar a))) → ∃ (T : PAT H (Sp R)) (e : T.t.G ≅ P.A), T.g = P.g ∧ Serre.IsLinearHom T.t.act act e.hom ∧ LamEq P.L e.hom e.hom T.t.PP.L (𝟙 T.t.G) (𝟙 T.t.G)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPeriodPairOfLatticeInLine

theorem **exists_periodPair_of_lattice_in_line** : ∀ (g : ℕ) (Λ : Submodule ℤ (Fin g → ℂ)) (w : Fin g → ℂ), g = 1 → w ≠ 0 → DiscreteTopology Λ → Submodule.span ℝ (Λ : Set (Fin g → ℂ)) = ⊤ → ∃ L : PeriodPair, ∀ z : ℂ, z ∈ L.lattice ↔ z • w ∈ Λ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPicardGroup

theorem **cl_eq** {X : Scheme.{0}} {L : LB X} (h : L.IsLB) : cl L = Multiplicative.ofAdd (CechPic.cls L h)

theorem **exists_picardGroup** : ∃ (Pic : Scheme.{0} → Type 1) (_ : ∀ X, CommGroup (Pic X)) (cl : ∀ {X : Scheme.{0}}, LB X → Pic X) (pl : ∀ {X Y : Scheme.{0}}, (Y ⟶ X) → (Pic X →* Pic Y)), (∀ (X : Scheme.{0}) (L M : LB X), L.IsLB → M.IsLB → (cl L = cl M ↔ LBIso L M)) ∧ (∀ (X : Scheme.{0}) (L : LB X), L.IsLB → (cl L = 1 ↔ ∃ t : LB.Gen L, t.IsGen)) ∧ (∀ (X : Scheme.{0}) (c : Pic X), pl (𝟙 X) c = c) ∧ (∀ (X Y Z : Scheme.{0}) (f : Y ⟶ X) (g : Z ⟶ Y) (c : Pic X), pl (g ≫ f) c = pl g (pl f c)) ∧ (∀ (X Y : Scheme.{0}) (L : LB X), L.IsLB → ∀ (I : Type) [Fintype I] (φ : I → (Y ⟶ X)) (ε : I → ℤ), (LB.LinComb L φ ε).IsLB ∧ cl (LB.LinComb L φ ε) = ∏ k, pl (φ k) (cl L) ^ ε k) ∧ ∀ (X : Scheme.{0}) (L M : LB X), L.IsLB → M.IsLB → (LB.div M L).IsLB ∧ cl (LB.div M L) = cl M / cl L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPolAVPullbackOfIsLocalRing

theorem **connectedSpace_sp** (R : Type) [CommRing R] [IsLocalRing R] : ConnectedSpace (Sp R)

theorem **exists_polAV_of_sabMap** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {S : Scheme.{0}} {t : SAB H S} (hp : IsProper t.G.hom) (hA : Out.RelAmple t.G t.PP.L) (hK : Out.KTriv t.G t.PP.L) {R : Type} [CommRing R] [IsLocalRing R] {y : Sp R ⟶ S} {t' : SAB H (Sp R)} (m : SABMap t' t y) : ∃ (P : Out.PolAV (Sp R)) (φ : P.A.left ⟶ t.G.left) (act : OAct (𝒪 E) P.A), IsGrpPullback P.A t.G y φ ∧ LBIso P.L (t.PP.L.pull φ) ∧ ∀ a, (act.act a).left ≫ φ = φ ≫ (t.act.act a).left

theorem **exists_polAV_pullback_of_isLocalRing** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S : Scheme.{0}) (t : SAB H S), IsProper t.G.hom → Out.RelAmple t.G t.PP.L → Out.KTriv t.G t.PP.L → ∀ (R : Type) [CommRing R] [IsLocalRing R] (y : Sp R ⟶ S), ∃ (P : Out.PolAV (Sp R)) (φ : P.A.left ⟶ t.G.left) (act : OAct (𝒪 E) P.A), IsGrpPullback P.A t.G y φ ∧ LBIso P.L (t.PP.L.pull φ) ∧ ∀ a, (act.act a).left ≫ φ = φ ≫ (t.act.act a).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPolarisedAbelianSchemeLieBasisInvariants

theorem **fibre_over_scheme** {R R' : Type} [CommRing R] [CommRing R'] (W : WeierstrassCurve R) (f : R →+* R') {A : Over (Sp R)} {A' : Over (Sp R')} [GrpObj A] [GrpObj A'] {ψ : A.left ⟶ scheme W} [IsIso ψ] (hψ : ψ ≫ toSpec W = A.hom) (he : eSec A ≫ ψ = zero W) {v : epsR R ⟶ A} {h : IsUnimodularSol (W.map (algebraMap R (DualNumber R))) ![DualNumber.eps, -1, 0]} (hv : v.left ≫ ψ = ptHom W (algebraMap R (DualNumber R)) ![DualNumber.eps, -1, 0] h) {φ : A'.left ⟶ A.left} (hφ : IsGrpPullback A' A (spMap f) φ) {v' : epsR R' ⟶ A'} (hv' : IsVecOver f φ v v') : ∃ h' : IsUnimodularSol (W.map ((algebraMap R' (DualNumber R')).comp f)) ![DualNumber.eps, -1, 0], IsPullback (φ ≫ ψ) A'.hom (toSpec W) (spMap f) ∧ eSec A' ≫ φ ≫ ψ = spMap f ≫ zero W ∧ v'.left ≫ φ ≫ ψ = ptHom W ((algebraMap R' (DualNumber R')).comp f) ![DualNumber.eps, -1, 0] h'

theorem **exists_polAV_invariants** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A), P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ (∀ (s' : R →+* ℂ) (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∃ L : PeriodPair, (∀ z : ℂ, z ∈ L.lattice ↔ z • w' ∈ u'.Λ) ∧ s' W.c₄ = 12 * L.g₂ ∧ s' W.c₆ = 216 * L.g₃ ∧ s' W.Δ = L.g₂ ^ 3 - 27 * L.g₃ ^ 2)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPolarisedAbelianSchemeLieBasisInvariantsStatement

theorem **_root_.GlobalTheory.T.exists_polarisedAbelianScheme_lieBasis_invariants** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∃ (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A), P.g = 1 ∧ IsLieBasis ![v] ∧ (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) ∧ (∀ (s' : R →+* ℂ) (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s' P P' φ' → IsVecOver s' φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∃ L : PeriodPair, (∀ z : ℂ, z ∈ L.lattice ↔ z • w' ∈ u'.Λ) ∧ s' W.c₄ = 12 * L.g₂ ∧ s' W.c₆ = 216 * L.g₃ ∧ s' W.Δ = L.g₂ ^ 3 - 27 * L.g₃ ^ 2)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPolarisedAbelianSchemePower

theorem **exists_polAV_isPowGrp** : ∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (n : ℕ), ∃ (Q : Out.PolAV (Sp R)) (_ : Out.IsPowGrp Q.A P.A n), Q.g = n * P.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPolarisedAbelianSchemePowerStatement

theorem **_root_.GlobalTheory.T.exists_polarisedAbelianScheme_power** : ∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (n : ℕ), ∃ (Q : Out.PolAV (Sp R)) (_ : Out.IsPowGrp Q.A P.A n), Q.g = n * P.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPolyDivDet

theorem **exists_genericRing** (F K : Type) [Field F] [Field K] [Algebra F K] (B : Type) [Fintype B] (m : B → ℕ) : ∃ (A : Type) (_ : CommRing A) (_ : Algebra F A) (π : K →ₐ[F] A) (X : ∀ b, Matrix (Fin (m b)) (Fin (m b)) A), _root_.IsReduced A ∧ (∀ b, IsUnit (X b).det) ∧ ∀ t : A, ∃ (P : MvPolynomial (FrVar m) K) (N : ℕ), ∀ (S : Type) [CommRing S] [Algebra F S] (φ : K →ₐ[F] S) (Y : ∀ b, Matrix (Fin (m b)) (Fin (m b)) S), (∀ b, IsUnit (Y b).det) → ∃ k : A →ₐ[F] S, k.comp π = φ ∧ (∀ b, (X b).map k = Y b) ∧ (∏ b, (Y b).det) ^ N * k t = evalFr (MvPolynomial.map φ.toRingHom P) Y

theorem **exists_poly_div_det** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∃ (P : MvPolynomial (FrVar Dw.mult) K) (N : ℕ), ∀ (S : Type) [CommRing S] [Algebra c.F S] (φ : K →ₐ[c.F] S) (z z' : FrPt Xo c S) (Y : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) S)), (∀ b, IsUnit (Y b).det) → FrMap φ y z → FrChg (fun b => (Y b).transpose) 1 z z' → (∏ b, (Y b).det) ^ N * evR Xo c F z' = evalFr (MvPolynomial.map φ.toRingHom P) Y

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPosKeyArchAt

theorem **pos_of_isArchNorm** (hU : ThetaUnifAV) (x : KeyIn ℂ) {nth nη : ℝ} (h : KeyPow.IsArchNorm x nth nη) : 0 < nth ∧ 0 < nη

theorem **exists_pos_keyArchAt** : ThetaUnifAV → ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (σ : K →+* ℂ), ∃ t n : ℝ, 0 < t ∧ 0 < n ∧ ∀ T abelian₂ : MLine K, T.n σ = t → abelian₂.n σ = n → KeyArchAt x T abelian₂ σ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPosKeyArchAtStatement

theorem **_root_.GlobalTheory.T.exists_pos_keyArchAt'** : ThetaUnifAV → ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (σ : K →+* ℂ), ∃ t n : ℝ, 0 < t ∧ 0 < n ∧ ∀ T abelian₂ : MLine K, T.n σ = t → abelian₂.n σ = n → KeyArchAt x T abelian₂ σ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPosNatMemFractionalIdeal

theorem **exists_pos_nat_mem_fractionalIdeal** {E : Type} [Field E] [NumberField E] {a : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (ha : a ≠ 0) : ∃ n : ℕ, 0 < n ∧ (n : E) ∈ a

theorem **fractionalIdeal_smul_eq_span** {E : Type} [Field E] [NumberField E] {r : ℕ} (a : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (L : Submodule (𝒪 E) (Fin r → E)) : (a : Submodule (𝒪 E) E) • L = Submodule.span (𝒪 E) {v | ∃ α ∈ a, ∃ u ∈ L, v = α • u}

theorem **exists_dataX_of_perfect_traceForm** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (𝔞 𝔟 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (k : ℝ) (g : ℕ), (∀ a : E, a ∈ 𝔞b ↔ cc a ∈ 𝔞) → FractionalIdeal.spanSingleton (nonZeroDivisors (𝒪 E)) X.ξ0 * 𝔞 * 𝔟 * (differentIdeal ℤ (𝒪 E) : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) = 1 → (∀ τ : E →+* ℂ, ‖τ X.ξ0‖ = k) → k ≤ 1 → (∀ x ∈ X.Λ, ∀ y ∈ X.Λ, ∃ n : ℤ, EFf X.ξ0 X.Gm x y = n) → (∀ φ : X.Λ →+ ℤ, ∃ x ∈ X.Λ, ∀ y : X.Λ, (φ y : ℚ) = EFf X.ξ0 X.Gm x y) → 0 < g → (∀ γ : GL (Fin r) E, IsUStab X.Gm X.Λ γ → (∀ x ∈ X.Λ, ∃ y ∈ X.Λ, Matrix.mulVec (γ : Matrix (Fin r) (Fin r) E) x - x = (g : E) • y) → γ ∈ X.Γ) → ∃ Xx : DataX E r, Xx.toData = X ∧ Xx.g = g ∧ Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔞, ∃ u ∈ Xx.L', v = a • u}

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPrincipalTripleModelOfIsogenyDedekind

theorem **exists_pat_model_of_isogeny** : ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (σ : K →+* ℂ) (P : Out.PolAV (Sp R)) (P₀ : Out.PolAV (Sp K)) (φ : P₀.A.left ⟶ P.A.left) (T₀ : PAT H (Sp K)) (ψ : P₀.A ⟶ T₀.t.G), IsPolFibre (algebraMap R K) P P₀ φ → P.g = T₀.g → IsMonHom ψ → Surjective ψ.left → ∃ (T : PAT H (Sp R)) (_ : SABMap T₀.t T.t (spMap (algebraMap R K))), T.g = T₀.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPrincipalTripleModelOfIsogenyDedekindStatement

theorem **_root_.GlobalTheory.T.exists_principalTriple_model_of_isogeny_dedekind** : ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (σ : K →+* ℂ) (P : Out.PolAV (Sp R)) (P₀ : Out.PolAV (Sp K)) (φ : P₀.A.left ⟶ P.A.left) (T₀ : PAT H (Sp K)) (ψ : P₀.A ⟶ T₀.t.G), IsPolFibre (algebraMap R K) P P₀ φ → P.g = T₀.g → IsMonHom ψ → Surjective ψ.left → ∃ (T : PAT H (Sp R)) (_ : SABMap T₀.t T.t (spMap (algebraMap R K))), T.g = T₀.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsProdPolAV

theorem **locZero_add** {Y B : Scheme.{0}} {π : Y ⟶ B} {c d : Pic Y} (H : LocZero π c) (H' : LocZero π d) : LocZero π (c + d)

theorem **rigidified_iff** (A : Over S) [GrpObj A] {L : LB A.left} (hL : L.IsLB) : Out.Rigidified A L ↔ pull (eSec A) (cls L hL) = 0

theorem **sym_iff** (A : Over S) [GrpObj A] {L : LB A.left} (hL : L.IsLB) : Out.Sym A L ↔ LocZero A.hom (pull (GrpObj.inv : A ⟶ A).left (cls L hL) - cls L hL)

theorem **eSec_comp** {A G : Over S} [GrpObj A] [GrpObj G] (f : A ⟶ G) [IsMonHom f] : eSec A ≫ f.left = eSec G

theorem **inv_comp_left** {A G : Over S} [GrpObj A] [GrpObj G] (f : A ⟶ G) [IsMonHom f] : (GrpObj.inv : A ⟶ A).left ≫ f.left = f.left ≫ (GrpObj.inv : G ⟶ G).left

theorem **exists_prod_polAV** : Out.Laws → ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (P P' : Out.PolAV (Sp R)), ∃ (Q : Out.PolAV (Sp R)) (p : Q.A → P.A) (q : Q.A → P'.A), IsMonHom p ∧ IsMonHom q ∧ IsIso (lift p q) ∧ Q.g = P.g + P'.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsProdPolAVStatement

theorem **_root_.GlobalTheory.T.exists_prod_polAV_add_g** : Out.Laws → ∀ (R : Type) [CommRing R] [IsNoetherianRing R] (P P' : Out.PolAV (Sp R)), ∃ (Q : Out.PolAV (Sp R)) (p : Q.A → P.A) (q : Q.A → P'.A), IsMonHom p ∧ IsMonHom q ∧ IsIso (lift p q) ∧ Q.g = P.g + P'.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPullback

theorem **PAT.exists_pullback** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S S' : Scheme.{0}) (f : S' → S) (T : PAT H S), ∃ (T' : PAT H S') (_ : SABMap T'.t T.t f), T'.g = T.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPullbackStatement

theorem **_root_.GlobalTheory.T.exists_pat_pullback_g_eq** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S S' : Scheme.{0}) (f : S' → S) (T : PAT H S), ∃ (T' : PAT H S') (_ : SABMap T'.t T.t f), T'.g = T.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsPureVectorsPeriodsSpan

theorem **span_pure_periods_eq_top** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {D : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {Pe : D.La →+ (Fin g → ℂ)} (hP : IsPeriod D ty Pe) {p q : ℕ} (Z : CMLat hP p q) : Submodule.span ℂ {v : Fin g → ℂ | ∃ (ε : ℤˣ) (a : PureType.lat hP (((ε : ℤ) : ℂ) * H.rt0)), Pe a = v} = ⊤

theorem **exists_pure_vectors_periods_span** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (hP : IsPeriod D ty Pe) (p q : ℕ) (Z : CMLat hP p q), ∃ (ε : Fin (Module.finrank ℚ E / 2 * (D.p + D.q)) → ℤˣ) (a : ∀ i, PureType.lat hP (((ε i : ℤ) : ℂ) * H.rt0)), ∀ z : Fin g → ℂ, ∃ c : Fin (Module.finrank ℚ E / 2 * (D.p + D.q)) → ℂ, z = ∑ i, c i • Pe (a i)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsRankOneCuspLatticeDataOfPrincipal

theorem **cc_eq_conj** (D : D7) (x : D.host.E) : ReciprocityTheory.ChartRestriction.G.FD.cc x = D.host.conj x

theorem **flagDomainConj_hostXi_eq_neg** (D : D7) : ReciprocityTheory.ChartRestriction.G.FD.cc D.host.xi0 = -D.host.xi0

theorem **halfDifferent_ne_bot** (D : D7) : D.host.𝔟 ≠ ⊥

theorem **absNorm_halfDifferent_le** (D : D7) : (Ideal.absNorm D.host.𝔟 : ℝ) ≤ |(NumberField.discr D.host.E : ℝ)|

theorem **absNorm_mul_mem_one** (D : D7) {a : D.host.E} (ha : a ∈ D.host.AA) : ∃ z : 𝒪 D.host.E, algebraMap (𝒪 D.host.E) D.host.E z = (Ideal.absNorm D.host.𝔟 : D.host.E) * a

theorem **absNorm_ccK_mem** (D : D7) : (Ideal.absNorm D.host.ccK : D.host.E) ∈ D.host.AA

theorem **exists_dataX_of_hasPrinc** : ∀ (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ) (hj : InR D j), ∀ σ, GoodEmb T.Q.c σ → GlobalTheory.FrameNorm.HasPrinc (Formalism.L2.PjFree.Unf.ofPj (T.U j σ)) T.Q.l.M → ∀ k : (T.U j σ).Comp, ∃ Xx : ReciprocityTheory.ChartRestriction.G.FD.DataX D.host.E D.R.r, Xx.toData = (T.U j σ).X k ∧ Xx.g = T.Q.l.M ∧ Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u} ∧ Xx.k0 = 2 * xiR D ∧ (Xx.NA : ℝ) ≤ |(NumberField.discr D.host.E : ℝ)| ∧ Xx.NC ≤ Ideal.absNorm D.host.ccK

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsRankOneCuspLatticeDataOfPrincipalStatement

theorem **_root_.GlobalTheory.T.exists_rankOneCuspLatticeData_of_principal** : ∀ (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ) (hj : InR D j), ∀ σ, GoodEmb T.Q.c σ → GlobalTheory.FrameNorm.HasPrinc (Formalism.L2.PjFree.Unf.ofPj (T.U j σ)) T.Q.l.M → ∀ k : (T.U j σ).Comp, ∃ Xx : ReciprocityTheory.ChartRestriction.G.FD.DataX D.host.E D.R.r, Xx.toData = (T.U j σ).X k ∧ Xx.g = T.Q.l.M ∧ Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u} ∧ Xx.k0 = 2 * xiR D ∧ (Xx.NA : ℝ) ≤ |(NumberField.discr D.host.E : ℝ)| ∧ Xx.NC ≤ Ideal.absNorm D.host.ccK

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsRigLBExtensionDedekind

theorem **exists_rigLB_extension_dedekind** : ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (G : Over (Sp R)) [GrpObj G], IsAbelianScheme G → ∀ (X' : Scheme.{0}) (π : X' → G.left) (p : X' → Sp K), IsPullback π p G.hom (spMap (algebraMap R K)) → ∀ L : LB X', L.IsLB → ∃ L' : RigLB G, LBIso (L'.L.pull π) L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsRigidifiedExtensionOfGenericFibre

theorem **exists_specializes_of_mem_closure** {X : Type*} [TopologicalSpace X] [NoetherianSpace X] [QuasiSober X] {O F : Set X} (hO : IsOpen O) (hF : IsClosed F) {y : X} (hy : y ∈ closure (O ∩ F)) : ∃ z ∈ O ∩ F, z ⤳ y

theorem **exists_equations'** (L : LB Y) (hL : L.IsLB) : ∃ f : L.ι → Y.functionFieldˣ, ∀ i k, genericPoint Y ∈ L.U i → genericPoint Y ∈ L.U k → unitVal _ (L.g i k) = f i / f k

theorem **isGen_of_equations'** (M N : LB Y) (a : M.ι → Y.functionFieldˣ) (b : N.ι → Y.functionFieldˣ) (ha : ∀ i k, genericPoint Y ∈ M.U i → genericPoint Y ∈ M.U k → unitVal _ (M.g i k) = a i / a k) (hb : ∀ i k, genericPoint Y ∈ N.U i → genericPoint Y ∈ N.U k → unitVal _ (N.g i k) = b i / b k) (s : ∀ p : M.ι × N.ι, Γ(Y, M.U p.1 ⊓ N.U p.2)ˣ) (hs : ∀ p, genericPoint Y ∈ (M.U p.1 ⊓ N.U p.2) → unitVal _ (s p) = a p.1 / b p.2) : LB.Gen.IsGen ⟨s⟩ : LB.Iso N M)

theorem **fieldMap_germ** {U : X.Opens} (h : genericPoint X ∈ U) (h' : genericPoint Y ∈ j ⁻¹ᵁ U) (s : Γ(X, U)) : fieldMap j hη (X.presheaf.germ U _ h s) = Y.presheaf.germ (j ⁻¹ᵁ U) _ h' (j.app U s)

theorem **exists_fieldMap_eq** [IsPreimmersion j] {y : Y} {u : Y.functionField} (hu : u ∈ localSubring y) : ∃ t ∈ localSubring (j y), fieldMap j hη t = u

theorem **fieldMap_surjective** [IsPreimmersion j] : Function.Surjective (fieldMap j hη)

theorem **fieldMap_mem_iff** [IsPreimmersion j] {y : Y} {t : X.functionField} : fieldMap j hη t ∈ localSubring y ↔ t ∈ localSubring (j y)

theorem **unitMap_mem_iff** [IsPreimmersion j] {y : Y} {t : X.functionFieldˣ} : unitMap j hη t ∈ localUnits y ↔ t ∈ localUnits (j y)

theorem **unitMap_surjective** [IsPreimmersion j] : Function.Surjective (unitMap j hη)

theorem **unitVal_pullU** {U : X.Opens} (h' : genericPoint Y ∈ j ⁻¹ᵁ U) (a : Γ(X, U)ˣ) : unitVal (j ⁻¹ᵁ U) (pullU j U a) = unitMap j hη (unitVal U a)

theorem **unitVal_pull** (M : LB X) (p q : PUnit → M.ι) (h : genericPoint Y ∈ (LB.wchart M (fun _ => j) p ⊓ LB.wchart M (fun _ => j) q)) : unitVal _ ((M.pull j).g p q) = unitMap j hη (unitVal _ (M.g (p ⟨⟩) (q ⟨⟩))))

theorem **isOpen_localUnits** (t : X.functionField×) : IsOpen {x : X | t ∈ localUnits x}

theorem **exists_lineBundle_extension_of_generalizing** (X Y : Scheme.{0}) [IsIntegral X] [IsIntegral Y] [IsLocallyNoetherian X] [CompactSpace X] (j : Y ⟶ X) [IsPreimmersion j] (hj : GeneralizingMap j) (hX : ∀ x : X, UniqueFactorizationMonoid (X.presheaf.stalk x)) (L : LB Y) (hL : L.IsLB) : ∃ L' : LB X, L'.IsLB ∧ LBIso (L'.pull j) L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsRosatiActionWeilPairingBalancedTriple

theorem **exists_oAct_rosatiConj** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E), (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] (T : PAT H (Sp (𝒪 K))), ∃ oa : T.pol.OAct (𝒪 E), oa.act = T.t.act.act ∧ RosatiConj T.pol oa

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsRosatiActionWeilPairingBalancedTripleStatement

theorem **_root_.GlobalTheory.T.exists_rosatiAction_weilPairing_balanced_triple** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E), (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] (T : PAT H (Sp (𝒪 K))), ∃ oa : T.pol.OAct (𝒪 E), oa.act = T.t.act.act ∧ RosatiConj T.pol oa

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsSecGeneric

theorem **exists_isPullOf** {O R : Type} [CommRing O] [CommRing R] [IsLocalRing R] (f : O →+* R) (P : Out.PolAV (Sp O)) (hg : 1 ≤ P.g) {t : LB.Gen (P.L.pull (eSec P.A))} (ht : t.IsGen) (θ : P.L.Sec) (hθ : ∀ s : Sp R, ∃ z, P.A.hom z = spMap f s ∧ z ∈ Out.nonzeroLocus θ) : ∃ y : KeyIn R, IsPullOf f P t θ y

theorem **IsPullOf.exists_keyPB** {O R R' : Type} [CommRing O] [CommRing R] [IsLocalRing R] [CommRing R'] {f : O →+* R} {g : R →+* R'} {P : Out.PolAV (Sp O)} {t : LB.Gen (P.L.pull (eSec P.A))} {θ : P.L.Sec} {y : KeyIn R} {x : KeyIn R'} (hy : IsPullOf f P t θ y) (hx : IsPullOf (g.comp f) P t θ x) : ∃ v' hv', KeyPB (x.P, x.g_pos, x.rig, x.rig_gen, x.th, x.th_ne, v', hv') y g

theorem **exists_sec_generic** (hT : Out.SecTwistSt) {O K : Type} [CommRing O] [IsDomain O] [Field K] [Algebra O K] [IsFractionRing O K] (P : Out.PolAV (Sp O)) {X : Scheme.{0}} {φ : X ⟶ P.A.left} {q : X ⟶ Sp K} (hp : IsPullback φ q P.A.hom (spMap (algebraMap O K))) : ∃ (θ : P.L.Sec) (z : X), φ z ∈ Out.nonzeroLocus θ

theorem **exists_scalar** (hT : Out.SecTwistSt) (P : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ P.A.left) (w : LB.Iso x.P.L (P.L.pull φ)) (hg : x.P.g = P.g) (hφ : IsGrpPullback x.P.A P.A (spMap (algebraMap (𝒪 K) K)) φ) (hw : w.IsGen) : ∃ a : K, a ≠ 0 ∧ {p | ¬ ∃ M, KeyPow.IsLocalModel x p a M}.Finite

theorem **exists_isThetaDeg** : Out.SecTwistSt → ThetaUnifAV → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (hg rig hr th hth), ∃ δ : ℝ, IsThetaDeg (F.key hg rig hr th hth) δ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsSmulOfPreservesFrameOrbits

theorem **_root_.GlobalTheory.OrbitScalar.exists_mem_dom** (X : _root_.ReciprocityTheory.ChartRestriction.G.FD.Data E r) : X.dom.Nonempty

private lemma **isOpen_dom_cubical_line_bundles** (X : _root_.ReciprocityTheory.ChartRestriction.G.FD.Data E r) : IsOpen (X.dom : Set (X.Ψ → Fin r → Fin r → ℂ))

private lemma **exists_scalar_of_eigen_on_open** {n : ℕ} (A : Matrix (Fin n) (Fin n) ℂ) (U : Set (Fin n → ℂ)) (hU : IsOpen U) (hne : U.Nonempty) (h : ∀ v ∈ U, ∃ c : ℂ, A *ᵥ v = c • v) : ∃ c : ℂ, ∀ w, A *ᵥ w = c • w

private lemma **exists_eq_smul_of_parallel_on_open** {n : ℕ} (A B : Matrix (Fin n) (Fin n) ℂ) (hB : B.det ≠ 0) (U : Set (Fin n → ℂ)) (hU : IsOpen U) (hne : U.Nonempty) (h : ∀ v ∈ U, ∀ i j, (A *ᵥ v) i * (B *ᵥ v) j = (A *ᵥ v) j * (B *ᵥ v) i) : ∃ c : ℂ, A = c • B

theorem **_root_.GlobalTheory.OrbitScalar.exists_smul_of_orbits** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : _root_.ReciprocityTheory.ChartRestriction.G.FD.Data E r) (γ : GL (Fin r) E), (∀ x ∈ X.dom, ∃ γ' ∈ X.Γ, ∃ b ∈ _root_.ReciprocityTheory.ChartRestriction.G.FD.Bor X.Ψ r, _root_.ReciprocityTheory.ChartRestriction.G.FD.lact γ x = _root_.ReciprocityTheory.ChartRestriction.G.FD.rmul (_root_.ReciprocityTheory.ChartRestriction.G.FD.lact γ' x) b) → ∃ γ' ∈ X.Γ, ∃ u : E, (γ' : Matrix (Fin r) (Fin r) E) = u • (γ : Matrix (Fin r) (Fin r) E)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTateIsoZetaEq

private lemma **isCoprime_add_sub_mul_pow_factorial** {j N : ℤ} (hj : IsCoprime j N) {n : ℕ} (hn : 0 < n) : IsCoprime (j + (1 - j) * N ^ n.factorial) n

private lemma **zmod_pow_eq_pow_factorial_of_dvd** {n : ℕ} (hn : 0 < n) (x : ZMod n) {c : ℕ} (hc : n.factorial ∣ c) (hc0 : 0 < c) : x ^ c = x ^ n.factorial

theorem **_root_.GlobalTheory.TateRoot.exists_tateIso_zeta_eq** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (D : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum H) (l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H) (k : Type) [Field k] (Q : Type) [Group Q] (act : 𝒪 E → Q → Q) (e : ℕ → Q → Q → kˣ → Prop) (_ : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.TateIso D k Q act e) (N : ℕ) (z : k), 0 < N → IsPrimitiveRoot z N → ∃ I' : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.TateIso D k Q act e, (I'.ζ N : k) = z

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTensorBundle

theorem **gramForm_add_left** (x y z : W) : gramForm β p g (x + y) z = gramForm β p g x z + gramForm β p g y z

theorem **gramForm_add_right** (x y z : W) : gramForm β p g x (y + z) = gramForm β p g x y + gramForm β p g x z

theorem **gramForm_zero** (y : W) : gramForm β p g 0 y = 0

theorem **gramForm_adj** (ha : ∀ c x y, β (c • x) y = β x (br c • y)) (c : O) (x y : W) : gramForm β p g (c • x) y = gramForm β p g x (br c • y)

theorem **gramForm_comm** (hs : ∀ x y, β x y = β y x) (ha : ∀ c x y, β (c • x) y = β x (br c • y)) (hg : ∀ i j, g j i = br (g i j)) (x y : W) : gramForm β p g x y = gramForm β p g y x

theorem **gramForm_polar** (hs : ∀ x y, β x y = β y x) (ha : ∀ c x y, β (c • x) y = β x (br c • y)) (hg : ∀ i j, g j i = br (g i j)) {w : O} (hw : w + br w = 1) (x y : W) : gramForm β p g (x + y) (w • (x + y)) + -gramForm β p g x (w • x) + -gramForm β p g y (w • y) + gramForm β p g 0 (w • 0) = gramForm β p g x y

theorem **gramForm_pure** (ha : ∀ c x y, β (c • x) y = β x (br c • y)) (hb : ∀ c, br (br c) = c) {x y : W} {u v : V} {s s' : I → O} (hx : ∀ i, p i x = s i • u) (hy : ∀ i, p i y = s' i • v) : gramForm β p g x y = β ((∑ k : I × I, br (s' k.2) * (g k.1 k.2 * s k.1)) • u) v

theorem **hermitian_expand** {M : Type} [AddCommGroup M] [Module O M] (h : M →ₗ[O] M →+ O) (hh : ∀ x y, h y x = br (h x y)) (e : I → M) (s s' : I → O) : ∑ k : I × I, br (s' k.2) * (h (e k.1) (e k.2) * s k.1) = h (∑ i, s i • e i) (∑ i, s' i • e i)

theorem **prodCls_left** (f f' : I → (A ⟶ G)) {T : Over S} (x : T ⟶ A) : prodCls L f f' x.left = ∏ k, diff₂ (ptCls L T) (x ≫ f k) (x ≫ f' k)

theorem **exists_tensor_bundle** : ∀ (S : Scheme.{0}) (O : Type) [CommRing O] (G A : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (M : Type) [AddCommGroup M] [Module O M] (t : Serre.IsTensor A b G a M) (br : O ≃+* O) (w : O) (L : LB G.left) (h : M →ₗ[O] M →+ O), Module.Finite O M → Module.Projective O M → w + br w = 1 → (∀ c, br (br c) = c) → (∀ x y, h y x = br (h x y)) → L.IsLB → (∃ t : LB.Gen (Theta G L), t.IsGen) → (∀ c, LamEq L (a.act c) (𝟙 G) L (𝟙 G) (a.act (br c))) → ∃ N : LB A.left, N.IsLB ∧ Out.Rigidified A N ∧ (∃ t : LB.Gen (Theta A N), t.IsGen) ∧ ∀ m m', LamEq N (t.tmul m) (t.tmul m') L (a.act (h m m')) (𝟙 G)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTensorDecompositionPureTypeCmTriple

theorem **ontoAt_of_isGrpPullback** {S S' : Scheme.{0}} {f : S' ⟶ S} {I : Type} {B A : Over S} {B' A' : Over S'} [GrpObj A] [GrpObj A'] {φ : B'.left ⟶ B.left} {χ : A'.left ⟶ A.left} (hφ : φ ≫ B.hom = B'.hom ≫ f) (hχ : IsGrpPullback A' A f χ) {ψ : I → (B ⟶ A)} {ψ' : I → (B' ⟶ A')} (hψ : ∀ i, (ψ' i).left ≫ χ = φ ≫ (ψ i).left) {k : Type} [Field k] {t' : Sp k ⟶ S'} {t : Sp k ⟶ S} (ht : t = t' ≫ f) (h : OntoAt (B

private lemma **spMap_comp_moduli_spaces** {A B C' : Type} [CommRing A] [CommRing B] [CommRing C'] (f : A →+* B) (g : B →+* C') : spMap g ≫ spMap f = spMap (g.comp f)

theorem **exists_isTen_of_hasType** : ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (k : IntermediateField ℚ ℂ) [NumberField k] (ω : 𝒪 k) (C : Ell H (𝒪 k) ω) (T : PAT H (Sp (𝒪 k))) (r : ℕ), C.IsPrincipalAt ((algebraMap k ℂ).comp (algebraMap (𝒪 k) k)) → HasType H T.t.act (2 * ω - 1) r 0 → ∃ (k₁ : IntermediateField ℚ ℂ) (_ : NumberField k₁) (hk : k ≤ k₁) (ω₁ : 𝒪 k₁) (C₁ : Ell H (𝒪 k₁) ω₁) (φ : C₁.P.A.left ⟶ C.P.A.left) (T₁ : PAT H (Sp (𝒪 k₁))) (_ : SABMap T₁.t T.t (spMap (RingOfIntegers.mapRingHom (IntermediateField.inclusion hk).toRingHom))) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), EllMap C₁ C (RingOfIntegers.mapRingHom (IntermediateField.inclusion hk).toRingHom) φ ∧ D.p + D.q = r ∧ IsKForm D ε h ∧ (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) ∧ Nonempty (IsTen C₁ D h T₁)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTensorDecompositionPureTypeCmTripleStatement

theorem **_root_.GlobalTheory.T.exists_tensor_decomposition_pure_type_cm_triple** : ThetaUnifAV → Out.Laws → HomExtDvr → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (k : IntermediateField ℚ ℂ) [NumberField k] (ω : 𝒪 k) (C : Ell H (𝒪 k) ω) (T : PAT H (Sp (𝒪 k))) (r : ℕ), C.IsPrincipalAt ((algebraMap k ℂ).comp (algebraMap (𝒪 k) k)) → HasType H T.t.act (2 * ω - 1) r 0 → ∃ (k₁ : IntermediateField ℚ ℂ) (_ : NumberField k₁) (hk : k ≤ k₁) (ω₁ : 𝒪 k₁) (C₁ : Ell H (𝒪 k₁) ω₁) (φ : C₁.P.A.left ⟶ C.P.A.left) (T₁ : PAT H (Sp (𝒪 k₁))) (_ : SABMap T₁.t T.t (spMap (RingOfIntegers.mapRingHom (IntermediateField.inclusion hk).toRingHom))) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), EllMap C₁ C (RingOfIntegers.mapRingHom (IntermediateField.inclusion hk).toRingHom) φ ∧ D.p + D.q = r ∧ IsKForm D ε h ∧ (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) ∧ Nonempty (IsTen C₁ D h T₁)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTensorProductHomOfCmCurveHomFamily

theorem **IsKForm.map_eq** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {D₁ D : Datum H} {ε : ℤˣ} {h₁ : D₁.La → D₁.La → OK H} {h : D.La → D.La → OK H} (k₁ : IsKForm D₁ ε h₁) (k : IsKForm D ε h) {L : Submodule (𝒪 E) D.La} (f : D₁.La →ₗ[𝒪 E] L) (c : 𝒪 E) (hf : ∀ a b, D.EF (f a) (f b) = D₁.EF (c • a) b) (a b : D₁.La) : h (f a) (f b) = h₁ (c • a) b

theorem **exists_famLift** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (D₁ D : Datum H) (ε : ℤˣ) (h₁ : D₁.La → D₁.La → OK H) (h : D.La → D.La → OK H) (T₁ T : PAT H (Sp R)) (t : IsTen C D₁ h₁ T₁) (L : Submodule (𝒪 E) D.La) (ψ : L → (C.P.A ⟶ T.t.G)) (f : D₁.La →ₗ[𝒪 E] L) (c : 𝒪 E), IsKForm D₁ ε h₁ → IsKForm D ε h → (∀ a b, D.EF (f a) (f b) = D₁.EF (c • a) b) → IsFam C T h L ψ → ∃ Φ : T₁.t.G ⟶ T.t.G, Serre.IsLinearHom T₁.t.act T.t.act Φ ∧ (∀ a, t.ten.tmul a ≫ Φ = ψ (f a)) ∧ LamEq T.t.PP.L Φ Φ T₁.t.PP.L (T₁.t.act.act c) (𝟙 T₁.t.G)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTensorProductHomOfCmCurveHomFamilyStatement

theorem **_root_.GlobalTheory.T.exists_tensorProduct_hom_of_cmCurveHomFamily** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (D₁ D : Datum H) (ε : ℤˣ) (h₁ : D₁.La → D₁.La → OK H) (h : D.La → D.La → OK H) (T₁ T : PAT H (Sp R)) (t : IsTen C D₁ h₁ T₁) (L : Submodule (𝒪 E) D.La) (ψ : L → (C.P.A ⟶ T.t.G)) (f : D₁.La →ₗ[𝒪 E] L) (c : 𝒪 E), IsKForm D₁ ε h₁ → IsKForm D ε h → (∀ a b, D.EF (f a) (f b) = D₁.EF (c • a) b) → IsFam C T h L ψ → ∃ Φ : T₁.t.G ⟶ T.t.G, Serre.IsLinearHom T₁.t.act T.t.act Φ ∧ (∀ a, t.ten.tmul a ≫ Φ = ψ (f a)) ∧ LamEq T.t.PP.L Φ Φ T₁.t.PP.L (T₁.t.act.act c) (𝟙 T₁.t.G)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTorsionFieldExt

theorem **exists_torsion_fieldExt** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) (k k' : Type) [Field k] [IsAlgClosed k] [CharZero k] [Field k'] [IsAlgClosed k'] (j : k →+* k') (t : Sp k ⟶ S), ∃ r : (Over.mk t ⟶ P.A) →* (Over.mk (spMap j ≫ t) ⟶ P.A), (∀ x, (r x).left = spMap j ≫ x.left) ∧ Function.Injective r ∧ (∀ (n : ℕ) y, 0 < n → y ^ n = 1 → ∃ x, r x = y) ∧ ∀ (n : ℕ) x y (c : kˣ), 0 < n → x ^ n = 1 → y ^ n = 1 → (Out.WeilIs P.A P.L n x y (unitOn c) ↔ Out.WeilIs P.A P.L n (r x) (r y) (unitOn (Units.map j.toMonoidHom c)))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTorsionFieldExtStatement

theorem **_root_.GlobalTheory.T.exists_torsion_injective_weil_compatible** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) (k k' : Type) [Field k] [IsAlgClosed k] [CharZero k] [Field k'] [IsAlgClosed k'] (j : k →+* k') (t : Sp k ⟶ S), ∃ r : (Over.mk t ⟶ P.A) →* (Over.mk (spMap j ≫ t) ⟶ P.A), (∀ x, (r x).left = spMap j ≫ x.left) ∧ Function.Injective r ∧ (∀ (n : ℕ) y, 0 < n → y ^ n = 1 → ∃ x, r x = y) ∧ ∀ (n : ℕ) x y (c : kˣ), 0 < n → x ^ n = 1 → y ^ n = 1 → (Out.WeilIs P.A P.L n x y (unitOn c) ↔ Out.WeilIs P.A P.L n (r x) (r y) (unitOn (Units.map j.toMonoidHom c)))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTorsionGeneratorsRational

theorem **exists_fibre** (hL : Out.Laws) (P : Out.PolAV (Sp R)) [IsCommMonObj P.A] {k : Type} [CommRing k] (t : R →+* k) : ∃ (P' : Out.PolAV (Sp k)) (φ : P'.A.left ⟶ P.A.left) (β : (𝟙_ (Over (Sp k)) ⟶ P'.A) ≃* (Over.mk (spMap t) ⟶ P.A)), IsPolFibre t P P' φ ∧ (∀ x, (β x).left = x.left ≫ φ) ∧ ∀ (n : ℤ) x y (c : Γ(Sp k, ⊤)ˣ), Out.WeilIs P'.A P'.L n x y c → Out.WeilIs P.A P.L n (β x) (β y) c

theorem **card_torsion** (hL : Out.Laws) (P : Out.PolAV (Sp R)) [IsCommMonObj P.A] {k : Type} [Field k] [IsAlgClosed k] (t : R →+* k) {n : ℕ} (hn : (n : k) ≠ 0) : Nat.card {x : Over.mk (spMap t) ⟶ P.A // x ^ n = 1} = n ^ (2 * P.g)

theorem **exists_weilIs** (hL : Out.Laws) (P : Out.PolAV (Sp R)) [IsCommMonObj P.A] {k : Type} [Field k] (t : R →+* k) {n : ℕ} {x y : Over.mk (spMap t) ⟶ P.A} (hx : x ^ n = 1) (hy : y ^ n = 1) : ∃ u : kˣ, Out.WeilIs P.A P.L n x y (unitOn u)

theorem **unitOn_injective** {k : Type} [CommRing k] : Function.Injective (unitOn (R

theorem **extPt_injective** (e : β.comp t = t') {A : Over (Sp R)} {x y : Over.mk (spMap t) ⟶ A} (h : extPt e ≫ x = extPt e ≫ y) : x = y

theorem **weilIs_extPt** (hL : Out.Laws) (P : Out.PolAV (Sp R)) (e : β.comp t = t') {n : ℤ} {x y : Over.mk (spMap t) ⟶ P.A} {u : Kˣ} (H : Out.WeilIs P.A P.L n x y (unitOn u)) : Out.WeilIs P.A P.L n (extPt e ≫ x) (extPt e ≫ y) (unitOn (Units.map β.toMonoidHom u))

theorem **exists_torsion_extPt** (hL : Out.Laws) (P : Out.PolAV (Sp R)) [IsCommMonObj P.A] [IsAlgClosed K] [IsAlgClosed k] [CharZero K] (e : β.comp t = t') {n : ℕ} (hn : 0 < n) {x : Over.mk (spMap t') ⟶ P.A} (hx : x ^ n = 1) : ∃ y, extPt e ≫ y = x

theorem **IsGenSystem.pow_eq_one** {ε : ℕ → (Over.mk (spMap t) ⟶ P.A)} {ζ : ℕ → Kˣ} (h : IsGenSystem P a tr t ε ζ) {n : ℕ} (hn : 0 < n) : ε n ^ n = 1

theorem **extPt_eq_one_iff** (e : β.comp t = t') {x : Over.mk (spMap t) ⟶ P.A} : extPt e ≫ x = 1 ↔ x = 1

theorem **unitsMap_injective** (β : K →+* k) : Function.Injective (Units.map β.toMonoidHom)

theorem **IsGenSystem.up** (hL : Out.Laws) [IsCommMonObj P.A] [IsAlgClosed K] [IsAlgClosed k] [CharZero K] (e : β.comp t = t') {ε : ℕ → (Over.mk (spMap t) ⟶ P.A)} {ζ : ℕ → Kˣ} (h : IsGenSystem P a tr t ε ζ) : IsGenSystem P a tr t' (fun n => extPt e ≫ ε n) (fun n => Units.map β.toMonoidHom (ζ n))

theorem **exists_unit_of_pow** [IsAlgClosed K] [CharZero K] (β : K →+* k) {n : ℕ} (hn : 0 < n) {z : kˣ} (hz : z ^ n = 1) : ∃ u : Kˣ, Units.map β.toMonoidHom u = z

theorem **IsGenSystem.down** (hL : Out.Laws) [IsCommMonObj P.A] [IsAlgClosed K] [IsAlgClosed k] [CharZero K] (e : β.comp t = t') {ε : ℕ → (Over.mk (spMap t') ⟶ P.A)} {ζ : ℕ → kˣ} (h : IsGenSystem P a tr t' ε ζ) : ∃ ε' ζ', IsGenSystem P a tr t ε' ζ'

theorem **exists_finite_coordinates** {Ω : Type} [Field Ω] {X : Scheme.{0}} (p : X ⟶ Sp R) [LocallyOfFiniteType p] (t : R →+* Ω) (x : Sp Ω ⟶ X) (hx : x ≫ p = spMap t) : ∃ s : Set Ω, s.Finite ∧ ∀ (B : Type) [CommRing B] (io : B →+* Ω), Function.Injective io → (∀ c, t c ∈ io.range) → (∀ y ∈ s, y ∈ io.range) → ∃ b : Sp B ⟶ X, spMap io ≫ b = x

theorem **exists_extend** {K Q k : Type} [Field K] [Field Q] [Field k] [CharZero K] [CharZero Q] [Algebra.IsAlgebraic ℚ Q] [IsAlgClosed k] (io : K →+* Q) (β : K →+* k) : ∃ β' : Q →+* k, β'.comp io = β

theorem **spMap_dual_scale** {R' : Type} [CommRing R'] (s : R →+* R') (c : R) : spMap (Chart.dualMapR s) ≫ spMap (epsScaleR c) = spMap (epsScaleR (s c)) ≫ spMap (Chart.dualMapR s)

theorem **exists_exp** (hU : ThetaUnifAV) (hL : Out.Laws) (s : R →+* ℂ) (P : Out.PolAV (Sp R)) [IsCommMonObj P.A] (v : epsR R ⟶ P.A) (L : PeriodPair) (hg : P.g = 1) (hv : IsTangent v) (hΛ : ∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s P P' φ' → IsVecOver s φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ L.lattice ↔ z • w' ∈ u'.Λ) : ∃ E : ℂ → (Over.mk (spMap s) ⟶ P.A), (∀ z z', E (z + z') = E z ∗ E z') ∧ Function.Surjective E ∧ (∀ z, E z = 1 ↔ z ∈ L.lattice) ∧ (∀ (f : P.A ⟶ P.A) (c : R), IsMonHom f → IsLieComb ![c] ![v] (v ≫ f) → ∀ z, E z ≫ f = E (s c ∗ z)) ∧ ∀ (n : ℕ) (x y : ℂ) (m : ℤ) (c : ℂˣ), 0 < n → x ∈ L.lattice → y ∈ L.lattice → (m : ℝ) = (x ∗ starRingEnd ℂ y).im / |(L.ω₁ ∗ starRingEnd ℂ L.ω₂).im| → (c : ℂ) = Complex.exp (-(2 ∗ Real.pi ∗ Complex.I) ∗ m / n) → Out.WeilIs P.A P.L n (E ((n : ℂ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTripleOfAbelianModelDedekind

theorem **isPullback_tensor** {a : X'.left ⟶ X.left} {b : Y'.left ⟶ Y.left} (ha : IsPullback a X'.hom X.hom f) (hb : IsPullback b Y'.hom Y.hom f) {c : (X' ⊗ Y').left ⟶ (X ⊗ Y).left} (h₁ : c ≫ (fst X Y).left = (fst X' Y').left ≫ a) (h₂ : c ≫ (snd X Y).left = (snd X' Y').left ≫ b) : IsPullback c (X' ⊗ Y').hom (X ⊗ Y).hom f

theorem **lbIso_cube_pull** (h : IsGrpPullback G' G f φ) (hM : M.IsLB) (hM' : M'.IsLB) (e : LBIso (M.pull φ) M') {T' : Over S'} {T : Over S} {n : ℕ} (x' : Fin n → (T' ⟶ G')) (x : Fin n → (T ⟶ G)) (U : Over.mk (T'.hom ≫ f) ⟶ T) (hx : ∀ i, pushPt h.1.w (x' i) = U ≫ x i) : LBIso ((cubeLB G x M).pull U.left) (cubeLB G' x' M')

theorem **lbIso_cube_of_eq** (hM : M.IsLB) {T T' : Over S} (ρ : T' ⟶ T) {n : ℕ} (x y : Fin n → (T ⟶ G)) (h : ∀ i, ρ ≫ x i = ρ ≫ y i) : LBIso ((cubeLB G x M).pull ρ.left) ((cubeLB G y M).pull ρ.left)

theorem **exists_gen_theta_zero** (hM : M.IsLB) : ∃ t : LB.Gen ((Theta G M).pull (eSec (G ⊗ G ⊗ G))), t.IsGen

theorem **exists_gen_of_lbIso** {X : Scheme.{0}} {L N : LB X} (hL : L.IsLB) (hN : N.IsLB) (H : LBIso L N) (h : ∃ t : L.Gen, t.IsGen) : ∃ t : N.Gen, t.IsGen

theorem **exists_triple_of_abelian_model** : Out.Laws → (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (t₀ : SAB H (Sp K)) (B : Over (Sp R)) [GrpObj B] (φ : t₀.G.left ⟶ B.left), IsAbelianScheme B → IsGrpPullback t₀.G B (spMap (algebraMap R K)) φ → ∃ (t : SAB H (Sp R)) (m : SABMap t₀ t (spMap (algebraMap R K))) (e : t.G ≅ B), IsMonHom e.hom ∧ m.φ ≫ e.hom.left = φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsTripleOfAbelianModelDedekindStatement

theorem **_root_.GlobalTheory.T.exists_triple_of_abelian_model_dedekind** : Out.Laws → (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (t₀ : SAB H (Sp K)) (B : Over (Sp R)) [GrpObj B] (φ : t₀.G.left ⟶ B.left), IsAbelianScheme B → IsGrpPullback t₀.G B (spMap (algebraMap R K)) φ → ∃ (t : SAB H (Sp R)) (m : SABMap t₀ t (spMap (algebraMap R K))) (e : t.G ≅ B), IsMonHom e.hom ∧ m.φ ≫ e.hom.left = φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniformisationOfComplexConjugateKeyDatum

theorem **le_pull_id** (x : Sp ℂ ⟶ A.left) (j : (L.pull (𝟙 A.left)).ι) : ⊤ ≤ x ⁻¹ᵁ (L.pull (𝟙 A.left)).U j ↔ ⊤ ≤ x ⁻¹ᵁ L.U (j PUnit.unit)

theorem **cj_le** (x : 𝟙_ (Over (Sp ℂ)) ⟶ A) (j : (L.pull ψ).ι) : ⊤ ≤ (cjPt h x).left ⁻¹ᵁ (L.pull ψ).U j ↔ ⊤ ≤ x.left ⁻¹ᵁ L.U (j PUnit.unit)

theorem **conjPull_isAH** (u : HolUnif A L g) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hA : u.IsAH Hm χ) : (conjPull h u).IsAH (Hm.map star) fun l => star (χ (star l.1, l.2))

theorem **exists_inv** {X : Scheme.{0}} {M N : LB X} (w : LB.Iso M N) (hw : w.IsGen) : ∃ w' : LB.Iso N M, w'.IsGen ∧ ∀ (x : Sp ℂ ⟶ X) (j : N.ι) (i : M.ι), ⊤ ≤ x ⁻¹ᵁ (N.U j ⊓ M.U i) → valAt' x (M.U i ⊓ N.U j) (w'.s (i, j)).val = (valAt' x (N.U j ⊓ M.U i) (w.s (j, i)).val)⁻¹

theorem **exists_conj_unif** : ∀ (x x' : KeyIn ℂ), KeyPB x' x (starRingEnd ℂ) → ∀ (hg : x'.P.g = x.P.g) (g : ℕ) (u : HolUnif x.P.A x.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume → ∃ (u' : HolUnif x'.P.A x'.P.L g) (χ' : u'.Λ → ℂ) (θ' : (Fin g → ℂ) → ℂ) (Fd' : Set (Fin g → ℂ)), u'.IsAH (Hm.map (starRingEnd ℂ)) χ' ∧ u'.IsRig x'.rig ∧ u'.IsFn x'.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume ∧ (∫ z in Fd',

‖θ' z‖ ^ 2 * Real.exp (-Real.pi * (hermForm (Hm.map (starRingEnd ℂ)) z z).re)) / (volume Fd').toReal = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal ∧ ∀ (cf : Fin x.P.g → ℂ) (w : epsR ℂ ⟶ x.P.A) (w' : epsR ℂ ⟶ x'.P.A) (W : Fin g → ℂ), IsLieComb cf x.v w → IsLieComb (fun i => (starRingEnd ℂ) (cf (i.cast hg))) x'.v w'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniformisationOfComplexConjugateKeyDatumStatement

theorem **_root_.GlobalTheory.T.exists_uniformisation_of_complexConjugate_keyDatum** : ∀ (x x' : KeyIn ℂ), KeyPB x' x (starRingEnd ℂ) → ∀ (hg : x'.P.g = x.P.g) (g : ℕ) (u : HolUnif x.P.A x.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume → ∃ (u' : HolUnif x'.P.A x'.P.L g) (χ' : u'.Λ → ℂ) (θ' : (Fin g → ℂ) → ℂ) (Fd' : Set (Fin g → ℂ)), u'.IsAH (Hm.map (starRingEnd ℂ)) χ' ∧ u'.IsRig x'.rig ∧ u'.IsFn x'.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume ∧ (∫ z in Fd', ‖θ' z‖ ^ 2 * Real.exp (-Real.pi * (hermForm (Hm.map (starRingEnd ℂ)) z z).re)) / (volume Fd').toReal = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal ∧ ∀ (cf : Fin x.P.g → ℂ) (w : epsR ℂ ⟶ x.P.A) (w' : epsR ℂ ⟶ x'.P.A) (W : Fin g → ℂ), IsLieComb cf x.v w → IsLieC

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueCmHermitianLatticeForm

theorem **kFun_coe** {H : Host E} (D : Datum H) (ε : ℤˣ) (x y : D.La) : ((kFun D ε x y : OK H) : 𝒪 E) = (((ε : ℤ) * (D.EF (H.w0 • x) y - D.EF x y) : 𝒪 E) + (((ε : ℤ) * D.EF x y : ℤ) : 𝒪 E) * H.w0

theorem **kform_w0_sq** (H : Host E) : H.w0 * H.w0 = H.w0 - (((1 + H.d) / 4 : ℕ) : 𝒪 E)

theorem **kform_bar_w0** (H : Host E) : H.bar H.w0 = 1 - H.w0

theorem **kform_skew** {H : Host E} (D : Datum H) (x y : D.La) : D.EF y x = - D.EF x y

theorem **kFun_add_left** {H : Host E} (D : Datum H) (ε : ℤˣ) (x x' y : D.La) : kFun D ε (x + x') y = kFun D ε x y + kFun D ε x' y

theorem **kFun_add_right** {H : Host E} (D : Datum H) (ε : ℤˣ) (x y y' : D.La) : kFun D ε x (y + y') = kFun D ε x y + kFun D ε x y'

theorem **kFun_zsmul** {H : Host E} (D : Datum H) (ε : ℤˣ) (m : ℤ) (x y : D.La) : kFun D ε (m • x) y = (m : OK H) * kFun D ε x y

theorem **kFun_w0** {H : Host E} (D : Datum H) (ε : ℤˣ) (x y : D.La) : kFun D ε (H.w0 • x) y = wK H * kFun D ε x y

theorem **kFun_smul** {H : Host E} (D : Datum H) (ε : ℤˣ) (a : OK H) (x y : D.La) : kFun D ε (a • x) y = a * kFun D ε x y

theorem **kLin_apply** {H : Host E} (D : Datum H) (ε : ℤˣ) (x y : D.La) : kLin D ε x y = kFun D ε x y

theorem **kform_coeff_unique** (H : Host E) {a b c d : ℤ} (h : (a : OK H) + (b : OK H) * wK H = (c : OK H) + (d : OK H) * wK H) : a = c ∧ b = d

theorem **kLin_perfect** {H : Host E} (D : Datum H) (ε : ℤˣ) (f : D.La →ₗ[OK H] OK H) : ∃! y, ∀ x, f x = kLin D ε x y

theorem **exists_kForm** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ), ∃ h : D.La →ₗ[OK H] D.La →+ OK H, IsKForm D ε (fun x y => h x y) ∧ (∀ h' : D.La → D.La → OK H, IsKForm D ε h' → h' = fun x y => h x y) ∧ (∀ x y, ((h y x : OK H) : 𝒪 E) = H.bar (h x y : OK H)) ∧ (∀ (s : 𝒪 E) x y, h (s • x) y = h x (H.bar s • y)) ∧ (∀ x, ((h x x : OK H) : 𝒪 E) = (((ε : ℤ) * D.EF (H.w0 • x) x : ℤ) : 𝒪 E)) ∧ (∀ f : D.La →ₗ[OK H] OK H, ∃! y, ∀ x, f x = h x y) ∧ ∃ (n : ℕ) (x y : Fin n → D.La), ∀ a, a = ∑ i, h a (y i) • x i

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueCmHermitianLatticeFormStatement

theorem **_root_.GlobalTheory.T.exists_unique_cmHermitianLatticeForm** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ), ∃ h : D.La →ₗ[OK H] D.La →+ OK H, IsKForm D ε (fun x y => h x y) ∧ (∀ h' : D.La → D.La → OK H, IsKForm D ε h' → h' = fun x y => h x y) ∧ (∀ x y, ((h y x : OK H) : 𝒪 E) = H.bar (h x y : OK H)) ∧ (∀ (s : 𝒪 E) x y, h (s • x) y = h x (H.bar s • y)) ∧ (∀ x, ((h x x : OK H) : 𝒪 E) = (((ε : ℤ) * D.EF (H.w0 • x) x : ℤ) : 𝒪 E)) ∧ (∀ f : D.La →ₗ[OK H] OK H, ∃! y, ∀ x, f x = h x y) ∧ ∃ (n : ℕ) (x y : Fin n → D.La), ∀ a, a = ∑ i, h a (y i) • x i

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueDifferentialMonHomLieBasis

theorem **isTangent_comp_hom** (f : G ⟶ G') [IsMonHom f] {v : epsR R ⟶ G} (hv : IsTangent v) : IsTangent (v ≫ f)

theorem **isLieComb_comp_hom** (f : G ⟶ G') [IsMonHom f] {n : ℕ} {c : Fin n → R} {v : Fin n → (epsR R ⟶ G)} {w : epsR R ⟶ G} (h : IsLieComb c v w) : IsLieComb c (fun i => v i ≫ f) (w ≫ f)

theorem **exists_linearMap** [IsCommMonObj G'] (f : G ⟶ G') [IsMonHom f] {n m : ℕ} {v : Fin n → (epsR R ⟶ G)} {v' : Fin m → (epsR R ⟶ G')} (hv : IsLieBasis v) (hv' : IsLieBasis v') : ∃ M : (Fin n → R) →ₗ[R] (Fin m → R), ∀ c w, IsLieComb c v w → IsLieComb (M c) v' (w ≫ f)

theorem **existsUnique_isDiffOf** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (P' : Out.PolAV (Sp (𝒪 K))) (F' : GenFib P') (f : P.A ⟶ P'.A), IsMonHom f → (∃! M, F.IsDiffOf F' f M) ∧ ∃ f₀ : F.P₀.A ⟶ F'.P₀.A, IsMonHom f₀ ∧ f₀.left ≫ F'.φ = F.φ ≫ f.left ∧ ∀ M, F.IsDiffOf F' f M → ∀ c w, IsLieComb c F.v w → IsLieComb (M c) F'.v (w ≫ f₀)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueDifferentialMonHomLieBasisStatement

theorem **_root_.GlobalTheory.T.existsUnique_differential_monHom_lieBasis** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (P' : Out.PolAV (Sp (𝒪 K))) (F' : GenFib P') (f : P.A ⟶ P'.A), IsMonHom f → (∃! M, F.IsDiffOf F' f M) ∧ ∃ f₀ : F.P₀.A ⟶ F'.P₀.A, IsMonHom f₀ ∧ f₀.left ≫ F'.φ = F.φ ≫ f.left ∧ ∀ M, F.IsDiffOf F' f M → ∀ c w, IsLieComb c F.v w → IsLieComb (M c) F'.v (w ≫ f₀)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueExtendRingOfIntegersHom

theorem **ringOfIntegers_hom_injective** (τ : 𝒪 E →+* 𝒪 K) : Function.Injective τ

theorem **existsUnique_extend_ringOfIntegers_hom** (τ : 𝒪 E →+* 𝒪 K) : ∃! τ' : E →+* K, ∀ a : 𝒪 E, τ' a = τ a

theorem **mapRingHom_bijective** : Function.Bijective fun σ : E →+* K => (RingOfIntegers.mapRingHom σ : 𝒪 E →+* 𝒪 K)

theorem **splits_iff_card** : Splits E K ↔ Nat.card (E →+* K) = Module.finrank ℚ E

theorem **card_ringHom_le_finrank** : Nat.card (E →+* K) ≤ Module.finrank ℚ E

theorem **comp_bijective_of_splits** (j : K →+* K') (h : Splits E K) : Function.Bijective fun σ : E →+* K => j.comp σ

theorem **splits_basic** : ∀ (E : Type) [Field E] [NumberField E], (∃ k : IntermediateField ℚ ℂ, FiniteDimensional ℚ k ∧ ∀ (τ : E →+* ℂ) a, τ a ∈ k) ∧ (∀ (k : IntermediateField ℚ ℂ) [NumberField k], (∀ (τ : E →+* ℂ) a, τ a ∈ k) → Splits E k) ∧ ∀ (K : Type) [Field K] [NumberField K], (Splits E K ↔ Nat.card (E →+* K) = Module.finrank ℚ E) ∧ (∀ τ : 𝒪 E →+* 𝒪 K, ∃! τ' : E →+* K, ∀ a : 𝒪 E, τ' a = τ a) ∧ ∀ (K' : Type) [Field K'] [NumberField K'] (j : K →+* K'), Splits E K → Splits E K' ∧ Function.Bijective fun τ : 𝒪 E →+* 𝒪 K => (RingOfIntegers.mapRingHom j).comp τ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueHomOfIsGrpPullback

theorem **existsUnique_hom_of_isGrpPullback** : ∀ (S S' : Scheme.{0}) (f : S' ⟶ S) (G G₁ : Over S) (G' G₁' : Over S') [GrpObj G] [GrpObj G'] [GrpObj G₁] [GrpObj G₁'] (φ : G'.left ⟶ G.left) (φ₁ : G₁'.left ⟶ G₁.left), IsGrpPullback G' G f φ → IsGrpPullback G₁' G₁ f φ₁ → ∀ F : G ⟶ G₁, (∃! F' : G' ⟶ G₁', F'.left ≫ φ₁ = φ ≫ F.left) ∧ ∀ F' : G' ⟶ G₁', F'.left ≫ φ₁ = φ ≫ F.left → IsMonHom F → IsMonHom F'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueLinearLiftOfHolomorphicAbelianVariet

theorem **tendsto_map_nhds_of_separating** [CompleteSpace V] [ProperSpace W] {ex : W →+ A} (hex : Function.Surjective ex) {G : V →+ A} {S : Set A} (h0 : (0 : A) ∈ S) {f : ι → A → Y} (hsep : ∀ a ∈ S, ∀ b ∈ S, (∀ i, f i a = f i b) → a = b) (hSe : IsOpen (ex ⁻¹' S)) (hSG : IsOpen (G ⁻¹' S)) (hfe : ∀ i, ContinuousOn (fun w ↦ f i (ex w)) (ex ⁻¹' S)) (hfG : ∀ i, ContinuousOn (fun z ↦ f i (G z)) (G ⁻¹' S)) : ∀ N ∈ 𝓝 (0 : W), ∀ᶠ z in 𝓝 (0 : V), G z ∈ ex '' N

theorem **existsUnique_linear_lift_of_separating** [NormedSpace ℂ V] [NormedSpace ℂ W] [FiniteDimensional ℂ V] [FiniteDimensional ℂ W] {ex : W →+ A} (hex : Function.Surjective ex) (hker : IsDiscrete (ex.ker : Set W)) {G : V →+ A} {S : Set A} (h0 : (0 : A) ∈ S) {f : ι → A → ℂ} (hsep : ∀ a ∈ S, ∀ b ∈ S, (∀ i, f i a = f i b) → a = b) (hSe : IsOpen (ex ⁻¹' S)) (hSG : IsOpen (G ⁻¹' S)) (hfe : ∀ i, DifferentiableOn ℂ (fun w ↦ f i (ex w)) (ex ⁻¹' S)) (hfG : ∀ i, DifferentiableOn ℂ (fun z ↦ f i (G z)) (G ⁻¹' S)) (hD : ∀ w, (∀ i, fderiv ℂ (fun w ↦ f i (ex w)) 0 w = 0) → w = 0) : ∃! F : V →ₗ[ℂ] W, ∀ z, ex (F z) = G z

theorem **existsUnique_linear_lift_of_isHolTo** : ∀ (A : Over (Sp ℂ)) [GrpObj A], IsAbelianScheme A → ∀ (L : LB A.left) (g m : ℕ) (u : HolUnif A L g), (∀ v, IsTangent v → ∃! w, u.IsDiff w v) → (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) → ∀ G : (Fin m → ℂ) → (𝟙_ (Over (Sp ℂ)) ⟶ A), (∀ z w, G (z + w) = G z ∗ G w) → IsHolTo (Set.univ : Set (Fin m → ℂ)) (fun z => (G z).left) → ∃! F : (Fin m → ℂ) →ₗ[ℂ] (Fin g → ℂ), ∀ z, u.ex (F z) = G z

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniqueLinearLiftOfHolomorphicAbelianVarietStatement

theorem **_root_.GlobalTheory.T.existsUnique_linearLift_of_holomorphic_abelianVariety** : ∀ (A : Over (Sp ℂ)) [GrpObj A], IsAbelianScheme A → ∀ (L : LB A.left) (g m : ℕ) (u : HolUnif A L g), (∀ v, IsTangent v → ∃! w, u.IsDiff w v) → (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) → ∀ G : (Fin m → ℂ) → (𝟙_ (Over (Sp ℂ)) ⟶ A), (∀ z w, G (z + w) = G z ∗ G w) → IsHolTo (Set.univ : Set (Fin m → ℂ)) (fun z => (G z).left) → ∃! F : (Fin m → ℂ) →ₗ[ℂ] (Fin g → ℂ), ∀ z, u.ex (F z) = G z

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUnitIntegralGeneratorPrincipalLocalModel

theorem **exists_secTie** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {t : SAB H (Sp S)} (m : SABMap t Xo.univ.t x) (σ : LB.Sec t.PP.L) : ∃ s : LB.Sec (PPx x), SecTie m σ s

theorem **exists_unit_secInt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (T : CDVR) (io : c.F →+* T.Fr) (G : PAT H (Sp T.V)) (t' : SAB H (Sp T.Fr)) (x : Sp T.Fr ⟶ Xo.X) (m1 : SABMap t' Xo.univ.t x) (m2 : SABMap t' G.t (spMap (algebraMap T.V T.Fr))), letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUnitIntegralGeneratorPrincipalLocalModelStatement

theorem **_root_.GlobalTheory.T.exists_unit_integral_generator_principal_local_model** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (T : CDVR) (io : c.F →+* T.Fr) (G : PAT H (Sp T.V)) (t' : SAB H (Sp T.Fr)) (x : Sp T.Fr ⟶ Xo.X) (m1 : SABMap t' Xo.univ.t x) (m2 : SABMap t' G.t (spMap (algebraMap T.V T.Fr))), letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUnitMul

theorem **res_res** {X : Scheme.{0}} {U V W : X.Opens} (h₁ : V ≤ U) (h₂ : W ≤ V) (a : Γ(X, U)) : resG X h₂ (resG X h₁ a) = resG X (h₂.trans h₁) a

theorem **top_le_preimage** {Y : Scheme.{0}} {U V : Y.Opens} (hU : IsAffineOpen U) (h : U ≤ V) {A : CommRingCat.{0}} (φ : Γ(Y, U) ⟶ A) : ⊤ ≤ (Spec.map φ ≫ hU.fromSpec) ⁻¹ᵁ V

theorem **appLE_spec** {Y : Scheme.{0}} {U V : Y.Opens} (hU : IsAffineOpen U) (h : U ≤ V) {A : CommRingCat.{0}} (φ : Γ(Y, U) ⟶ A) : (Spec.map φ ≫ hU.fromSpec).appLE V ⊤ (top_le_preimage hU h φ) = Y.presheaf.map (homOfLE h).op ≫ φ ≫ (Scheme.ΓSpecIso A).inv

theorem **dvd_on_affine** {Y : Scheme.{0}} {U V : Y.Opens} (hU : IsAffineOpen U) (h : U ≤ V) (a b : Γ(Y, V)) (H : ∀ (T : Scheme.{0}) (t : T ⟶ Y) (h : ⊤ ≤ t ⁻¹ᵁ V), (t.appLE V ⊤ h).hom a = 0 → (t.appLE V ⊤ h).hom b = 0) : resG Y h a ∣ resG Y h b

theorem **exists_unit_mul** (Y : Scheme.{0}) (V : Y.Opens) (a b : Γ(Y, V)) (ha : ∀ (V' : Y.Opens) (h : V' ≤ V), resG Y h a ∈ nonZeroDivisors Γ(Y, V')) (H : ∀ (T : Scheme.{0}) (t : T ⟶ Y) (h : ⊤ ≤ t ⁻¹ᵁ V), (t.appLE V ⊤ h).hom a = 0 ↔ (t.appLE V ⊤ h).hom b = 0) : ∃ c : Γ(Y, V)ˣ, b = (c : Γ(Y, V)) ∗ a

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniversalTorsionPairPolAV

theorem **etale_pullback_torsion** {S S' : Scheme.{0}} (f : S' ⟶ S) (G : Over S) [GrpObj G] [IsCommMonObj G] [Smooth G.hom] {n : ℕ} (hn : IsUnit (n : Γ(S', ⊤))) : Etale ((Over.pullback f).obj (torsion G n)).hom

theorem **etale_pullback_tensor** {S S' : Scheme.{0}} (f : S' ⟶ S) (A B : Over S) (hA : Etale ((Over.pullback f).obj A).hom) (hB : Etale ((Over.pullback f).obj B).hom) : Etale ((Over.pullback f).obj (A ⊗ B)).hom

theorem **torsion_universal** {S : Scheme.{0}} (G : Over S) [GrpObj G] (n : ℕ) : torsion.ι G n ^ n = 1 ∧ ∀ (T : Over S) (x : T ⟶ G), x ^ n = 1 → ∃ t : T ⟶ torsion G n, t ≫ torsion.ι G n = x

theorem **exists_universal_torsion_pair** : Out.Laws → ∀ (R : Type) [CommRing R] [IsDedekindDomain R] [CharZero R] (X Y : Out.PolAV (Sp R)) (k : ℕ), 0 < k → ∃ (T₀ : Over (Sp R)) (x₀ : T₀ ⟶ X.A) (y₀ : T₀ ⟶ Y.A), x₀ ^ k = 1 ∧ y₀ ^ k = 1 ∧ (∀ (T : Over (Sp R)) (x : T ⟶ X.A) (y : T ⟶ Y.A), x ^ k = 1 → y ^ k = 1 → ∃ t : T ⟶ T₀, x = t ≫ x₀ ∧ y = t ≫ y₀) ∧ ∀ a b : Γ(T₀.left, ⊤), (∀ (k' : Type) [Field k'] [IsAlgClosed k'] [CharZero k'] (β : R →+* k') (t : Over.mk (spMap β) ⟶ T₀), t.left.appTop.hom a = t.left.appTop.hom b) → a = b

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsUniversalTorsionPairPolAVStatement

theorem **_root_.GlobalTheory.T.exists_universal_torsion_pair_polAV** : Out.Laws → ∀ (R : Type) [CommRing R] [IsDedekindDomain R] [CharZero R] (X Y : Out.PolAV (Sp R)) (k : ℕ), 0 < k → ∃ (T₀ : Over (Sp R)) (x₀ : T₀ ⟶ X.A) (y₀ : T₀ ⟶ Y.A), x₀ ^ k = 1 ∧ y₀ ^ k = 1 ∧ (∀ (T : Over (Sp R)) (x : T ⟶ X.A) (y : T ⟶ Y.A), x ^ k = 1 → y ^ k = 1 → ∃ t : T ⟶ T₀, x = t ≫ x₀ ∧ y = t ≫ y₀) ∧ ∀ a b : Γ(T₀.left, ⊤), (∀ (k' : Type) [Field k'] [IsAlgClosed k'] [CharZero k'] (β : R →+* k') (t : Over.mk (spMap β) ⟶ T₀), t.left.appTop.hom a = t.left.appTop.hom b) → a = b

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsWeierstrassCoordinatesOnLine

theorem **exists_xyt** {g : ℕ} (V : WeierstrassCurve ℂ) (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ] (P : (Fin g → ℂ) → Fin 3 → ℂ) (F : Fin 3 → Fin 3 → (Fin g → ℂ) → ℂ) (hE : ∀ z, V.toProjective.Equation (P z)) (ho : ∀ i, IsOpen {z | P z i ≠ 0}) (hd : ∀ i j, DifferentiableOn ℂ (F i j) {z | P z i ≠ 0}) (hF : ∀ i j z, P z i ≠ 0 → F i j z * P z i = P z j) (hΛ : ∀ z, P z 2 = 0 ↔ z ∈ Λ) (hper : ∀ i j z l, l ∈ Λ → F i j (z + l) = F i j z) (h0 : P 0 0 = 0) (h1 : P 0 1 ≠ 0) (w : Fin g → ℂ) (hw : fderiv ℂ (F 1 0) 0 w = -1) : w ≠ 0 ∧ ∃ x y t : ℂ → ℂ, DifferentiableOn ℂ x {z | z • w ∈ Λ}ᶜ ∧ DifferentiableOn ℂ y {z | z • w ∈ Λ}ᶜ ∧ (∀ l : ℂ, l • w ∈ Λ → ∀ z, x (z + l) = x z ∧ y (z + l) = y z) ∧ (∀ z : ℂ, z • w ∉ Λ → V.toAffine.Equation (x z) (y z)) ∧ AnalyticAt ℂ t 0 ∧ t 0 = 0 ∧ deriv t 0 = 1 ∧ (∀ᶠ z in nhdsWithin 0 {0}ᶜ, t z * y z = -x z) ∧ Filter.Tendsto (fun z => ‖y z‖) (nhdsWithin 0 {0}ᶜ) Filte

theorem **top_le_ptHom_preimage_iff** {R : Type} [CommRing R] (W : WeierstrassCurve R) (f : R →+* ℂ) (P : Fin 3 → ℂ) (hP : IsUnimodularSol (W.map f) P) (i : Fin 3) : ⊤ ≤ ptHom W f P hP ⁻¹ᵁ coordOpen W i ↔ P i ≠ 0

theorem **eq_zero_and_ne_zero** {V : WeierstrassCurve ℂ} {P : Fin 3 → ℂ} (hP : IsUnimodularSol V P) (h2 : P 2 = 0) : P 0 = 0 ∧ P 1 ≠ 0

theorem **ptHom_eq_zero_iff** {R : Type} [CommRing R] (W : WeierstrassCurve R) (f : R →+* ℂ) (P : Fin 3 → ℂ) (hP : IsUnimodularSol (W.map f) P) : P 2 = 0 ↔ ptHom W f P hP = spMap f ≫ zero W

theorem **exists_coord_on_line** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R) (f : R →+* ℂ) (A : Over (Sp ℂ)) [GrpObj A] (φ : A.left ⟶ scheme W) (v : epsR ℂ ⟶ A) (h : IsUnimodularSol (W.map ((algebraMap ℂ (DualNumber ℂ)).comp f)) ![DualNumber.eps, -1, 0]), IsPullback φ A.hom (toSpec W) (spMap f) → eSec A ≫ φ = spMap f ≫ zero W → v.left ≫ φ = ptHom W ((algebraMap ℂ (DualNumber ℂ)).comp f) ![DualNumber.eps, -1, 0] h → ∀ (L₁ : LB A.left) (g : ℕ) (u : HolUnif A L₁ g) (w : Fin g → ℂ), u.IsDiff w v → w ≠ 0 ∧ ∃ x y t : ℂ → ℂ, DifferentiableOn ℂ x {z | z • w ∈ u.Λ}ᶜ ∧ DifferentiableOn ℂ y {z | z • w ∈ u.Λ}ᶜ ∧ (∀ l : ℂ, l • w ∈ u.Λ → ∀ z, x (z + l) = x z ∧ y (z + l) = y z) ∧ (∀ z : ℂ, z • w ∉ u.Λ → (W.map f).toAffine.Equation (x z) (y z)) ∧ AnalyticAt ℂ t 0 ∧ t 0 = 0 ∧ deriv t 0 = 1 ∧ (∀ᶠ z in nhdsWithin 0 {0}ᶜ, t z * y z = -x z) ∧ Filter.Tendsto (fun z => ‖y z‖) (nhdsWithin 0 {0}ᶜ) Filter.at

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ExistsWeierstrassCoordinatesOnLineStatement

theorem **_root_.GlobalTheory.T.exists_weierstrass_coordinates_on_line** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R) (f : R →+* ℂ) (A : Over (Sp ℂ)) [GrpObj A] (φ : A.left ⟶ scheme W) (v : epsR ℂ ⟶ A) (h : IsUnimodularSol (W.map ((algebraMap ℂ (DualNumber ℂ)).comp f)) ![DualNumber.eps, -1, 0]), IsPullback φ A.hom (toSpec W) (spMap f) → eSec A ≫ φ = spMap f ≫ zero W → v.left ≫ φ = ptHom W ((algebraMap ℂ (DualNumber ℂ)).comp f) ![DualNumber.eps, -1, 0] h → ∀ (L₁ : LB A.left) (g : ℕ) (u : HolUnif A L₁ g) (w : Fin g → ℂ), u.IsDiff w v → w ≠ 0 ∧ ∃ x y t : ℂ → ℂ, DifferentiableOn ℂ x {z | z • w ∈ u.Λ}ᶜ ∧ DifferentiableOn ℂ y {z | z • w ∈ u.Λ}ᶜ ∧ (∀ l : ℂ, l • w ∈ u.Λ → ∀ z, x (z + l) = x z ∧ y (z + l) = y z) ∧ (∀ z : ℂ, z • w ∉ u.Λ → (W.map f).toAffine.Equation (x z) (y z)) ∧ AnalyticAt ℂ t 0 ∧ t 0 = 0 ∧ deriv t 0 = 1 ∧ (∀ᶠ z in nhdsWithin 0 {0}ᶜ, t z * y z = -x z) ∧ Filter.Tendsto (fun z

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FamilyIdentitiesDescendFieldToDedekind

theorem **lamEq_descend** {R k : Type} [CommRing R] [IsDedekindDomain R] [Field k] (f : R →+* k) (hf : Function.Injective f) {X Y A B : Over (Sp R)} {X' Y' A' B' : Over (Sp k)} [GrpObj X] [GrpObj Y] [GrpObj A] [IsCommMonObj A] [GrpObj B] [IsCommMonObj B] [GrpObj A'] [IsCommMonObj A'] [GrpObj B'] [IsCommMonObj B'] (hX : IsAbelianScheme X) (hY : IsAbelianScheme Y) {x : X'.left ⟶ X.left} {y : Y'.left ⟶ Y.left} {a : A'.left ⟶ A.left} {b : B'.left ⟶ B.left} (hx : IsPullback x X'.hom X.hom (spMap f)) (hy : IsPullback y Y'.hom Y.hom (spMap f)) (hA : IsGrpPullback A' A (spMap f) a) (hB : IsGrpPullback B' B (spMap f) b) {L : LB A.left} {M : LB B.left} {L' : LB A'.left} {M' : LB B'.left} (hL : L.IsLB) (hM : M.IsLB) (hL' : L'.IsLB) (hM' : M'.IsLB) (iL : LBIso (L.pull a) L') (iM : LBIso (M.pull b) M') {f₁ : X ⟶ A} {f₂ : Y ⟶ A} {g₁ : X ⟶ B} {g₂ : Y ⟶ B} [IsMonHom f₁] [IsMonHom f₂] [IsMonHom g₁] [IsMonHom

theorem **isFam_descend** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsDedekindDomain R] (ω : R) (C : Ell H R ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (L : Submodule (𝒪 E) D.La) (ψ : L → (C.P.A ⟶ T.t.G)) (k : Type) [Field k] (ω' : k) (C' : Ell H k ω') (f : R →+* k) (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp k)) (m : SABMap T'.t T.t (spMap f)) (ψ' : L → (C'.P.A ⟶ T'.t.G)), Function.Injective f → EllMap C' C f φ → (∀ a, (ψ' a).left ≫ m.φ = φ ≫ (ψ a).left) → IsFam C' T' h L ψ' → IsFam C T h L ψ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FamilyIdentitiesDescendFieldToDedekindStatement

theorem **_root_.GlobalTheory.T.family_identities_descend_field_to_dedekind** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsDedekindDomain R] (ω : R) (C : Ell H R ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (L : Submodule (𝒪 E) D.La) (ψ : L → (C.P.A ⟶ T.t.G)) (k : Type) [Field k] (ω' : k) (C' : Ell H k ω') (f : R →+* k) (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp k)) (m : SABMap T'.t T.t (spMap f)) (ψ' : L → (C'.P.A ⟶ T'.t.G)), Function.Injective f → EllMap C' C f φ → (∀ a, (ψ' a).left ≫ m.φ = φ ≫ (ψ a).left) → IsFam C' T' h L ψ' → IsFam C T h L ψ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FamilyUniformisationFibreTheta

theorem **isHolTo_of_isPullback** {V : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] {X Y Z : Scheme.{0}} {φ : Y ⟶ Z} {s : Y ⟶ Sp ℂ} {p : Z ⟶ X} {y : Sp ℂ ⟶ X} (H : IsPullback φ s p y) {e : V → (Sp ℂ ⟶ Y)} (hs : ∀ z, e z ≫ s = 𝟙 _) (hol : IsHolTo Set.univ (fun z => e z ≫ φ)) : IsHolTo Set.univ e

theorem **weil_eq_of_grpPullback** {S S' : Scheme.{0}} {G : Over S} [GrpObj G] {A : Over S'} [GrpObj A] [IsCommMonObj A] {f : S' ⟶ S} {φ : A.left ⟶ G.left} (hp : IsGrpPullback A G f φ) {L₀ : LB G.left} {L : LB A.left} (hL₀ : L₀.IsLB) (hL : L.IsLB) (w : LB.Iso L (L₀.pull φ)) (hw : w.IsGen) {Z : Scheme.{0}} [Subsingleton Z] {t : Z ⟶ S} {t' : Z ⟶ S'} (ht : t = t' ≫ f) (k : ℤ) (x y : Over.mk t' ⟶ A) (x₀ y₀ : Over.mk t ⟶ G) (ex : x₀.left = x.left ≫ φ) (ey : y₀.left = y.left ≫ φ) (c c₀ : Γ(Z, ⊤)ˣ) (h : Out.WeilIs A L k x y c) (h₀ : Out.WeilIs G L₀ k x₀ y₀ c₀) : c = c₀

theorem **eq_zero_of_smul_mem_countable** {V : Type} [AddCommGroup V] [Module ℂ V] {C : Set V} (hC : C.Countable) {z : V} (h : ∀ t : ℂ, t • z ∈ C) : z = 0

theorem **surjective_of_countable** {V W : Type} [AddCommGroup V] [Module ℂ V] [AddCommGroup W] [Module ℂ W] (F : V →ₗ[ℂ] W) {C : Set W} (hC : C.Countable) (h : ∀ w, ∃ z, w - F z ∈ C) : Function.Surjective F

theorem **int_eq_of_exp_eq** {m n : ℤ} (h : ∀ k : ℕ, 0 < k → Complex.exp (-(2 * Real.pi * Complex.I) * (m : ℂ) / k) = Complex.exp (-(2 * Real.pi * Complex.I) * (n : ℂ) / k)) : m = n

theorem **unitOn_injective** {R : Type} [CommRing R] : Function.Injective (unitOn (R

theorem **isHolTo_slice** {V W : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] [NormedAddCommGroup W] [NormedSpace ℂ W] {X : Scheme.{0}} {y : V × W → (Sp ℂ ⟶ X)} (h : IsHolTo (Set.univ ×ˢ Set.univ) y) (b : V) : IsHolTo Set.univ (fun z => y (b, z))

theorem **familyUnif_fibre_theta** : ThetaUnifAV → Out.WeilSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (σ : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo σ ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)), IsPeriod (FU.Di i) ty Pe → ∀ (P : Out.PolAV (Sp ℂ)) (φ : P.A.left ⟶ Xo.univ.t.G.left), IsGrpPullback P.A Xo.univ.t.G (FU.u i Pe) φ → LBIso P.L (Xo.univ.t.PP.L.pull φ) → ∃ (u : HolUnif P.A P.L P.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ

: u.Λ → ℂ) (F : (Fin g → ℂ) ≃ₗ[ℂ] (Fin P.g → ℂ)), u.IsAH Hm χ ∧ (∀ v, IsTangent v → ∃! w, u.IsDiff w v) ∧ (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) ∧ (∀ n : ℕ, (∀ s, ∃! θ, u.IsThetaFn Hm χ n θ ∧ u.IsFnPow n s θ) ∧ ∀ θ, u.IsThetaFn Hm χ n θ → ∃! s, u.IsFnPow n s θ) ∧ (∀ z, (u.ex (F z)).left ≫ φ = (FU.ex i Pe z).left) ∧ (∀ z, F z ∈ u.Λ ↔ z ∈ Pe.range) ∧ ∀ a b, (hermF

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FamilyUniformisationFibreThetaStatement

theorem **_root_.GlobalTheory.T.familyUniformisation_fibre_theta** : ThetaUnifAV → Out.WeilSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (σ : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (FU : FamilyUnif Xo σ ty) (i : Fin FU.n) (Pe : (FU.Di i).La →+ (Fin g → ℂ)), IsPeriod (FU.Di i) ty Pe → ∀ (P : Out.PolAV (Sp ℂ)) (φ : P.A.left ⟶ Xo.univ.t.G.left), IsGrpPullback P.A Xo.univ.t.G (FU.u i Pe) φ → LBIso P.L (Xo.univ.t.PP.L.pull φ) → ∃ (u : HolUnif P.A P.L P.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ) (F : (Fin g → ℂ) ≃ₗ[ℂ] (Fin P.g → ℂ)), u.IsAH Hm χ ∧ (∀ v, IsTangent v → ∃! w, u.IsDiff w v) ∧ (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) ∧ (∀ n : ℕ, (∀ s, ∃! θ, u.IsThetaFn Hm χ n θ ∧ u.IsFnPow n s θ) ∧ ∀ θ, u.IsThetaFn Hm χ n θ → ∃! s, u.IsFnPow n s θ) ∧ (∀ z, (u.ex (F z)).left ≫ φ = (FU.ex i Pe z).left) ∧ (∀ z, F z ∈ u.Λ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FinrankInt

theorem **exists_enum** (H : Host E) : ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)), IsTypeFn H 1 0 ty ∧ ∀ i, ty i (H.rt : E) = H.rt0

theorem **trace_eq** (H : Host E) {g : ℕ} {ty : Fin g → (E →+* ℂ)} (hty : IsTypeFn H 1 0 ty) (a : E) : ((Algebra.trace ℚ E a : ℚ) : ℝ) = 2 * (∑ i, ty i a).re

theorem **exists_form** (𝔄 : Submodule (𝒪 E) E) (ξ : E) (hξ0 : ξ ≠ 0) (hdual : ∀ x : E, (∀ y ∈ 𝔄, ∃ n : ℤ, Algebra.trace ℚ E (ξ * x * IsCMField.complexConj E y) = n) ↔ x ∈ 𝔄) : ∃ EF : LinearMap.BilinForm ℤ 𝔄, Function.Bijective EF ∧ ∀ x y : 𝔄, (EF x y : ℚ) = Algebra.trace ℚ E (ξ * (x : E) * IsCMField.complexConj E (y : E))

theorem **finrank_int** (𝔄 : Submodule (𝒪 E) E) (hfg : 𝔄.FG) (hne : 𝔄 ≠ ⊥) : Module.finrank ℤ 𝔄 = Module.finrank ℚ E

theorem **exists_datum** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (𝔄 : Submodule (𝒪 E) E) (ξ : E), 𝔄.FG → 𝔄 ≠ ⊥ → IsCMField.complexConj E ξ = -ξ → (∀ τ : E →+* ℂ, τ (H.rt : E) = H.rt0 → (τ ξ).im < 0) → (∀ x : E, (∀ y ∈ 𝔄, ∃ n : ℤ, Algebra.trace ℚ E (ξ * x * IsCMField.complexConj E y) = n) ↔ x ∈ 𝔄) → ∃ (D : Datum H) (e : D.La ≃ₗ[𝒪 E] 𝔄), D.p = 1 ∧ D.q = 0 ∧ ∀ x y, ((D.EF x y : ℤ) : ℚ) = Algebra.trace ℚ E (ξ * (e x : E) * IsCMField.complexConj E (e y : E))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FormEqPiMulHermForm

theorem **form_eq_sum_sum** {K : Type} [Field K] {g : ℕ} (h : HermFam K (Fin g → K)) (s : K →+* ℂ) (x y : Fin g → K) : h.form s x y = ∑ j, ∑ i, s (y j) * star (s (x i)) * h.form s (Pi.single i 1) (Pi.single j 1)

theorem **hermForm_sum_sum** {g n : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (a b : Fin n → ℂ) (X : Fin n → Fin g → ℂ) : hermForm Hm (∑ j, a j • X j) (∑ i, b i • X i) = ∑ j, ∑ i, a j * star (b i) * hermForm Hm (X j) (X i)

theorem **form_eq_pi_mul_hermForm** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ (s : K →+* ℂ) (Q : Out.PolAV (Sp ℂ)) (ψ : Q.A.left ⟶ F.P₀.A.left) (u : HolUnif Q.A Q.L Q.g) (Hm : Matrix (Fin Q.g) (Fin Q.g) ℂ) (χ : u.Λ → ℂ), IsPolFibre s F.P₀ Q ψ → u.IsAH Hm χ → ∀ (c c' : Fin F.P₀.g → K) (w w' : epsR K ⟶ F.P₀.A) (v v' : epsR ℂ ⟶ Q.A) (W W' : Fin Q.g → ℂ), IsLieComb c F.v w → IsLieComb c' F.v w' → IsVecOver s ψ w v → IsVecOver s ψ w' v' → u.IsDiff W v → u.IsDiff W' v' → h.form s c c' = Real.pi * hermForm Hm W' W

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FormLetterLe

theorem **form_letter_le** (hU : ThetaUnifAV) (hL : Out.Laws) {E : Type} [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] {K : Type} [Field K] [NumberField K] {S : Set (E →+* ℂ)} (A B : Out.PolAV (Sp (𝒪 K))) (oa : A.OAct (𝒪 E)) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (τ : Fin x.P.g ≃ S) (lab : (E →+* ℂ) → (E →+* K)) (hx : IsFibreOf A x (gmap K) φ) (hT : IsLieType A oa x φ τ lab) (F' : GenFib B) (h' : HermFam K (Fin F'.P₀.g → K)) (H' : F'.IsHerm h') (f : A.A ⟶ B.A) (hf : IsMonHom f) (a : 𝒪 E) (hW : ∀ (Tb : Over (Sp (𝒪 K))) (k : ℕ) (x y : Tb ⟶ A.A) (c : Γ(Tb.left, ⊤)ˣ), 0 < k → pnsmulPt k x = toUnit Tb ≫ η[A.A] → pnsmulPt k y = toUnit Tb ≫ η[A.A] → (Out.WeilIs B.A B.L k (x ≫ f) (y ≫ f) c ↔ Out.WeilIs A.A A.L k (x ≫ oa.act a) y c)) (La : ℝ) (hb : ∀ τ : E →+* ℂ, ‖τ (a : E)‖ ≤ La ^ 2) (i : Fin x.P.g) (ℓ : MLine K) (hℓ : IsVecLine A x φ i ℓ) (U : Fin F'.P₀.g → K) (w' : epsR K ⟶ F'.P₀.A)

theorem **lamLen** : ThetaUnifAV → Out.Laws → LamLenSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FramePolynomialEquivariantAndNeZero

theorem **evR_frChg_tri** {S : Type} [CommRing S] [Algebra c.F S] (hS : IsReduced S) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {F : Rul Xo c} (hF : F ∈ sec Xo c a n) (t : Ix Dw → Sˣ) (g : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) S) (h0 : ∀ τ i j, i < j → g τ i j = 0) (hd : ∀ τ i, g τ i i = (t (some (τ, i)) : S)) {y y' : FrPt Xo c S} (h : FrChg g (t none : S) y y') : evR Xo c F y' = wchar (ex Dw a n) t * evR Xo c F y

theorem **evalFr_mul_tri** {K : Type} [CommRing K] [Algebra c.F K] {y : FrPt Xo c K} {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {F : Rul Xo c} (hF : F ∈ sec Xo c a n) {G : MvPolynomial (FrVar Dw.mult) K} (hG : IsFrPoly F y G) {S : Type} [CommRing S] [Algebra c.F S] (hS : IsReduced S) (φ : K →ₐ[c.F] S) (X T : ∀ τ : E →+* ℂ, Matrix (Fin (Dw.mult τ)) (Fin (Dw.mult τ)) S) (hX : ∀ b, IsUnit (X b).det) (hT : ∀ b, IsUnit (T b).det) (h0 : ∀ τ i j, i < j → T τ i j = 0) (hd : ∀ τ i, IsUnit (T τ i i)) : evalFr (MvPolynomial.map φ.toRingHom G) (fun b => X b * T b) = (∏ b, ∏ i : Fin (Dw.mult b), T b i i ^ a (b, i)) * evalFr (MvPolynomial.map φ.toRingHom G) X

theorem **isFrPoly_equiv_and_ne_zero** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (y : FrPt Xo c K), ∀ F ∈ sec Xo c a n, ∀ G : MvPolynomial (FrVar Dw.mult) K, IsFrPoly F y G → (∀ [DecidableEq (E →+* ℂ)] (σ : K →+* ℂ), IsEquiv (fun b (i : Fin (Dw.mult b)) => a (b, i)) (MvPolynomial.map σ G)) ∧ ∀ (σ : c.F →+* ℂ) (σ₁ : K →ₐ[c.F] CAt σ) (w : FrPt Xo c (CAt σ)), spMap σ₁.toRingHom ≫ y.x = w.x → evR Xo c F w ≠ 0 → G ≠ 0

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.FramePolynomialEquivariantAndNeZeroStatement

theorem **_root_.GlobalTheory.T.framePolynomial_equivariant_and_ne_zero** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [Algebra c.F K] (y : FrPt Xo c K) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, ∀ G : MvPolynomial (FrVar Dw.mult) K, IsFrPoly F y G → (∀ [DecidableEq (E →+* ℂ)] (σ : K →+* ℂ), IsEquiv (fun b (i : Fin (Dw.mult b)) => a (b, i)) (MvPolynomial.map σ G)) ∧ ∀ (σ : c.F →+* ℂ) (σ₁ : K →ₐ[c.F] CAt σ) (w : FrPt Xo c (CAt σ)), spMap σ₁.toRingHom ≫ y.x = w.x → evR Xo c F w ≠ 0 → G ≠ 0

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.GramStdBasisPosDefAndHq

theorem **gram_stdBasis_posDef_and_hq** : ∀ (K : Type) [Field K] [NumberField K] (n : ℕ) (h : HermFam K (Fin n → K)) (s : K →+* ℂ), (h.gram s (fun i => Pi.single i 1) : Matrix (Fin n) (Fin n) ℂ).PosDef ∧ ∀ c, ReciprocityTheory.MultiplicativeHeights.hq (h.gram s (fun i =>

Pi.single i 1)) (ReciprocityTheory.MultiplicativeHeights.embVec s c) = (h.form s c c).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HasPeriodOfEllMap

theorem **hasPeriod_of_ellMap** : Out.PolBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (M K : Type) [Field M] [NumberField M] [Field K] [NumberField K] (j : M →+* K) (ω : 𝒪 M) (ω' : 𝒪 K) (C : Ell H (𝒪 M) ω) (C' : Ell H (𝒪 K) ω') (φ : C'.P.A.left ⟶ C.P.A.left) (x : ℝ), EllMap C' C (RingOfIntegers.mapRingHom j) φ → C.HasPeriod x → C'.HasPeriod x

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HasTypeBaseChange

theorem **hasType_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R R' : Type) [CommRing R] [CommRing R'] (f : R →+* R') (t : SAB H (Sp R)) (t' : SAB H (Sp R')) (_ : SABMap t' t (spMap f)) (ρ : R) (p q : ℕ), (Function.Injective f → HasType H t.act ρ p q → HasType H t'.act (f ρ) p q) ∧ ((∀ (k : Type) [Field k] (β : R →+* k), Function.Injective β → ∃ β' : R' →+* k, β'.comp f = β ∧ Function.Injective β') → HasType H t'.act (f ρ) p q → HasType H t.act ρ p q)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HasTypeOfIso

theorem **hasType_of_iso** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (G G' : Over (Sp R)) [GrpObj G] [GrpObj G'] (a : OAct (𝒪 E) G) (a' : OAct (𝒪 E) G') (e : G ≅ G') (ρ : R) (p q : ℕ), Serre.IsLinearHom a a' e.hom → HasType H a' ρ p q → HasType H a ρ p q

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermFamIsometryInjective

theorem **hermFam_isometry_injective** {K : Type*} [Field K] [NumberField K] {V V' : Type*} [AddCommGroup V] [Module K V] [AddCommGroup V'] [Module K V'] (h : HermFam K V) (h' : HermFam K V') (M : V →ₗ[K] V') (hM : ∀ s a b, h'.form s (M a) (M b) = h.form s a b) : Function.Injective M

theorem **deg_map_isometry** : ∀ (K : Type) [Field K] [NumberField K] (V V' : Type) [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] [AddCommGroup V'] [Module K V'] [Module (𝒪 K) V'] [IsScalarTower (𝒪 K) K V'] (h : HermFam K V) (h' : HermFam K V') (M : V →ₗ[K] V') (Λ : Submodule (𝒪 K) V), Λ.FG → (∀ s a b, h'.form s (M a) (M b) = h.form s a b) → ReciprocityTheory.GeometryOfNumbersTheory.deg h' (Λ.map (M.restrictScalars (𝒪 K))) = ReciprocityTheory.GeometryOfNumbersTheory.deg h Λ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermFormPrincipalDimOne

theorem **hermForm_principal_dim_one** : ∀ (g : ℕ) (Λ : Submodule ℤ (Fin g → ℂ)) (w : Fin g → ℂ) (L : PeriodPair) (Hm : Matrix (Fin g) (Fin g) ℂ) (Ef : LinearMap.BilinForm ℤ Λ), g = 1 → w ≠ 0 → (∀ z : ℂ, z ∈ L.lattice ↔ z • w ∈ Λ) → Hm.IsHermitian → 0 < (hermForm Hm w w).re → (∀ a b : Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) → Function.Bijective Ef → ∀ z z' : ℂ, hermForm Hm (z • w) (z' • w) = z * starRingEnd ℂ z' / (|(L.ω₁ * starRingEnd ℂ L.ω₂).im| : ℝ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermFormPullEqOfLbIsoPow

lemma **valAt'_eq_unitVal** {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)ˣ) : valAt' x U f.val = unitVal x h f

lemma **unitVal_res** {U V : X.Opens} (hVU : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵘ V) (h' : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)ˣ) : unitVal x h (res hVU f) = unitVal x h' f

lemma **unitVal_pullU** {Y : Scheme.{0}} (φ : X ⟶ Y) (U : Y.Opens) (h : ⊤ ≤ x ⁻¹ᵘ (φ ⁻¹ᵘ U)) (h' : ⊤ ≤ (x ≫ φ) ⁻¹ᵘ U) (f : Γ(Y, U)ˣ) : unitVal x h (pullU φ U f) = unitVal (x ≫ φ) h' f

lemma **unitVal_comp_id** {U : X.Opens} (h : ⊤ ≤ (x ≫ 𝟙 X) ⁻¹ᵘ U) (h' : ⊤ ≤ x ⁻¹ᵘ U) (f : Γ(X, U)ˣ) : unitVal (x ≫ 𝟙 X) h f = unitVal x h' f

lemma **inf_of** {U V : X.Opens} (h : ⊤ ≤ x ⁻¹ᵘ U) (h' : ⊤ ≤ x ⁻¹ᵘ V) : ⊤ ≤ x ⁻¹ᵘ (U ⊓ V)

lemma **wchart_of** {Y : Scheme.{0}} (L : LB Y) (φ : X ⟶ Y) (j : L.ι) (h : ⊤ ≤ (x ≫ φ) ⁻¹ᵘ L.U j) : ⊤ ≤ x ⁻¹ᵘ L.wchart (K

lemma **exists_chart** {L : LB X} (hL : L.IsLB) : ∃ i, ⊤ ≤ x ⁻¹ᵘ L.U i

lemma **unitVal_linComb** {Y : Scheme.{0}} (L : LB Y) (φ : X ⟶ Y) (ε : ℤ) (j j' : L.ι) (h : ⊤ ≤ x ⁻¹ᵘ ((L.LinComb (K

lemma **unitVal_iso** {M N : LB X} (w : LB.Iso N M) (hw : w.IsGen) (i i' : M.ι) (j j' : N.ι) (hi : ⊤ ≤ x ⁻¹ᵘ M.U i) (hi' : ⊤ ≤ x ⁻¹ᵘ M.U i') (hj : ⊤ ≤ x ⁻¹ᵘ N.U j) (hj' : ⊤ ≤ x ⁻¹ᵘ N.U j') : unitVal x (U

lemma **idx_chart** {Y : Scheme.{0}} {L : LB X} {L' : LB Y} (f : X ⟶ Y) (m n : ℕ) (i : L.ι) (j : L'.ι) (hi : ⊤ ≤ x ⁻¹ᵘ L.U i) (hj : ⊤ ≤ (x ≫ f) ⁻¹ᵘ L'.U j) : ⊤ ≤ x ⁻¹ᵘ (LB.div (lbPow L n) (lbPow (L'.pull f) m)).U (idx f m n i j)

lemma **val_iso_pow** {Y : Scheme.{0}} {L : LB X} {L' : LB Y} (f : X ⟶ Y) (m n : ℕ) (w : LB.Iso (lbPow (L'.pull f) m) (lbPow L n)) (hw : w.IsGen) (i i' : L.ι) (j j' : L'.ι) (hi : ⊤ ≤ x ⁻¹ᵘ L.U i) (hi' : ⊤ ≤ x ⁻¹ᵘ L.U i') (hj : ⊤ ≤ (x ≫ f) ⁻¹ᵘ L'.U j) (hj' : ⊤ ≤ (x ≫ f) ⁻¹ᵘ L'.U j') : valAt' x _ (w.s (idx f m n i j)).val * valAt' (x ≫ f) (L'.U j ⊓ L'.U j') (L'.g j j').val ^ m = valAt' x (L.U i ⊓ L.U i') (L.g i i').val ^ n * valAt' x _ (w.s (idx f m n i' j')).val

lemma **exists_glue** {E κ : Type} [NormedAddCommGroup E] [NormedSpace ℂ E] (V : κ → Set E) (hV : ∀ i, IsOpen (V i)) (hc : ∀ z, ∃ i, z ∈ V i) (φ : κ → E → ℂ) (hφ : ∀ i, DifferentiableOn ℂ (φ i) (V i)) (ha : ∀ i j z, z ∈ V i → z ∈ V j → φ i z = φ j z) : ∃ r : E → ℂ, Differentiable ℂ r ∧ ∀ i z, z ∈ V i → r z = φ i z

lemma **hermForm_pull** {g g' : ℕ} (H : Matrix (Fin g') (Fin g') ℂ) (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (z w : Fin g → ℂ) : hermForm ((LinearMap.toMatrix' F).transpose * H * (LinearMap.toMatrix' F).transpose.conjTranspose) z w = hermForm H (F z) (F w)

lemma **hermForm_sub_smul** {g : ℕ} (H H' : Matrix (Fin g) (Fin g) ℂ) (a b : ℂ) (z w : Fin g → ℂ) : hermForm (a • H − b • H') z w = a * hermForm H z w − b * hermForm H' z w

theorem **hermForm_pull_eq_of_lbIso_pow** : ∀ (A A' : Over (Sp ℂ)) [GrpObj A] [GrpObj A'] (L : LB A.left) (L' : LB A'.left), L.IsLB → L'.IsLB → ∀ (g g' : ℕ) (u : HolUnif A L g) (u' : HolUnif A' L' g') (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (Hm' : Matrix (Fin g') (Fin g') ℂ) (χ' : u'.Λ → ℂ), u.IsAH Hm χ → u'.IsAH Hm' χ' → ∀ (f : A ⟶ A') (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)), (∀ z, u'.ex (F z) = u.ex z ≫ f) → ∀ m n : ℕ, LBIso (lbPow (L'.pull f.left) m) (lbPow L n) → ∀ z w, (m : ℂ) * hermForm Hm' (F z) (F w) = (n : ℂ) * hermForm Hm z w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermFormSumSum

theorem **form_sum_right** {J : Type} (t : Finset J) (x : V) (c : J → K) (y : J → V) : h.form s x (∑ j ∈ t, c j • y j) = ∑ j ∈ t, s (c j) * h.form s x (y j)

theorem **form_sum_left** {J : Type} (t : Finset J) (c : J → K) (x : J → V) (y : V) : h.form s (∑ i ∈ t, c i • x i) y = ∑ i ∈ t, starRingEnd ℂ (s (c i)) * h.form s (x i) y

theorem **hermForm_sum_sum** {g n : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (a : Fin n → ℂ) (W : Fin n → Fin g → ℂ) : hermForm Hm (∑ i, a i • W i) (∑ i, a i • W i) = ∑ i, starRingEnd ℂ (a i) * ∑ j, a j * hermForm Hm (W j) (W i)

theorem **form_eq** {K : Type} [Field K] {g n : ℕ} (h : HermFam K (Fin n → K)) (s : K →+* ℂ) (Hm : Matrix (Fin g) (Fin g) ℂ) (W : Fin n → Fin g → ℂ) (r : ℂ) (e : ∀ i j, h.form s (Pi.single i 1) (Pi.single j 1) = r * hermForm Hm (W j) (W i)) (c : Fin n → K) : h.form s c c = r * hermForm Hm (∑ i, s (c i) • W i) (∑ i, s (c i) • W i)

theorem **isLieNormSq_form** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ (s : K →+* ℂ) u c, F.IsCoord u c → IsLieNormSq P u (s.comp (algebraMap (𝒪 K) K)) (h.form s c c).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermitianLengthImageEigenBasisVectorLe

theorem **re_form_smul_le** {K : Type} [Field K] {V : Type} [AddCommGroup V] [Module K V] (h : HermFam K V) (s : K →+* ℂ) (t : K) {x : V} (hx : x ≠ 0) : (h.form s x (t • x)).re ≤ ‖s t‖ * (h.form s x x).re

theorem **diff_single_eq** {K : Type} [Field K] [NumberField K] {P P' : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) (F' : GenFib P') {f : P.A ⟶ P'.A} {f₀ : F.P₀.A ⟶ F'.P₀.A} (e : f₀.left ≫ F'.φ = F.φ ≫ f.left) {M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)} (hM : ∀ c w, IsLieComb c F.v w → IsLieComb (M c) F'.v (w ≫ f₀)) (i : Fin F.P₀.g) {U : Fin F'.P₀.g → K} {w : epsR K ⟶ F'.P₀.A} (hU : IsLieComb U F'.v w) (hw : w.left ≫ F'.φ = (F.v i).left ≫ F.φ ≫ f.left) : M (Pi.single i 1) = U

theorem **form_image_le** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P P' : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (F' : GenFib P') (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)) (f : P.A ⟶ P'.A) (g : P.A ⟶ P.A), F.IsHerm h → F'.IsHerm h' → IsMonHom f → IsMonHom g → LamEq P'.L f f P.L g (𝟙 P.A) → ∀ (i : Fin F.P₀.g) (t : K) (w : epsR K ⟶ F.P₀.A) (U : Fin F'.P₀.g → K) (w' : epsR K ⟶ F'.P₀.A), IsLieComb (Pi.single i t) F.v w → (F.v i).left ≫ F.φ ≫ g.left = w.left ≫ F.φ → IsLieComb U F'.v w' → w'.left ≫ F'.φ = (F.v i).left ≫ F.φ ≫ f.left → ∀ s : K →+* ℂ, (h'.form s U U).re ≤ ‖s t‖ * (h.form s (Pi.single i 1) (Pi.single i 1)).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermitianLengthImageEigenBasisVectorLeStatement

theorem **_root_.GlobalTheory.T.hermitianLength_image_eigenBasisVector_le** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P P' : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (F' : GenFib P') (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)) (f : P.A ⟶ P'.A) (g : P.A ⟶ P.A), F.IsHerm h → F'.IsHerm h' → IsMonHom f → IsMonHom g → LamEq P'.L f f P.L g (𝟙 P.A) → ∀ (i : Fin F.P₀.g) (t : K) (w : epsR K ⟶ F.P₀.A) (U : Fin F'.P₀.g → K) (w' : epsR K ⟶ F'.P₀.A), IsLieComb (Pi.single i t) F.v w → (F.v i).left ≫ F.φ ≫ g.left = w.left ≫ F.φ → IsLieComb U F'.v w' → w'.left ≫ F'.φ = (F.v i).left ≫ F.φ ≫ f.left → ∀ s : K →+* ℂ, (h'.form s U U).re ≤ ‖s t‖ * (h.form s (Pi.single i 1) (Pi.single i 1)).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermitianMetricDifferentialEqOfPairingRelation

theorem **isTangent_of_isLieComb** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g n : ℕ} (u : HolUnif A L g) {W : Fin n → (Fin g → ℂ)} {v : Fin n → (epsR ℂ ⟶ A)} {c : Fin n → ℂ} {w : epsR ℂ ⟶ A} (hv : ∀ i, IsTangent (v i)) (h : ∀ i, u.IsDiff (W i) (v i)) (hw : IsLieComb c v w) : IsTangent w

theorem **map_comb** {K : Type} [Field K] (s : K →+* ℂ) {A A' : Over (Sp K)} [GrpObj A] [GrpObj A'] {B B' : Over (Sp ℂ)} [GrpObj B] [GrpObj B'] {ψ : B.left ⟶ A.left} {ψ' : B'.left ⟶ A'.left} (hψ : IsGrpPullback B A (spMap s) ψ) (hψ' : IsGrpPullback B' A' (spMap s) ψ') {L : LB B.left} {L' : LB B'.left} {g g' n n' : ℕ} (u : HolUnif B L g) (u' : HolUnif B' L' g') {v : Fin n → (epsR K ⟶ A)} {v' : Fin n' → (epsR K ⟶ A')} (hv : ∀ i, IsTangent (v i)) (hv' : ∀ i, IsTangent (v' i)) {t : Fin n → (epsR ℂ ⟶ B)} {t' : Fin n' → (epsR ℂ ⟶ B')} {W : Fin n → Fin g → ℂ} {W' : Fin n' → Fin g' → ℂ} (hW : ∀ i, IsVecOver s ψ (v i) (t i) ∧ u.IsDiff (W i) (t i)) (hW' : ∀ i, IsVecOver s ψ' (v' i) (t' i) ∧ u'.IsDiff (W' i) (t' i)) (hu : ∀ x, IsTangent x → ∃! w, u'.IsDiff w x) {f : A ⟶ A'} {f₁ : B ⟶ B'} (e : f₁.left ≫ ψ' = ψ ≫ f.left) {Φ : (Fin g → ℂ) → (Fin g' → ℂ)} (hΦ : ∀ w x, IsTangent x → u.IsDiff w x → IsTangent

private lemma **hermForm_sum_smul_left** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) {J : Type*} (S : Finset J) (a : J → ℂ) (z : J → Fin g → ℂ) (w : Fin g → ℂ) : hermForm Hm (∑ i ∈ S, a i • z i) w = ∑ i ∈ S, a i * hermForm Hm (z i) w

private lemma **hermForm_sum_smul_right** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z : Fin g → ℂ) {J : Type*} (S : Finset J) (a : J → ℂ) (w : J → Fin g → ℂ) : hermForm Hm z (∑ i ∈ S, a i • w i) = ∑ i ∈ S, starRingEnd ℂ (a i) * hermForm Hm z (w i)

private lemma **hermFam_form_apply_zero** {K V : Type*} [Field K] [AddCommGroup V] [Module K V] (h : HermFam K V) (σ : K →+* ℂ) (x : V) : h.form σ x 0 = 0

private lemma **form_sum_smul_right** {K V : Type*} [Field K] [AddCommGroup V] [Module K V] (h : HermFam K V) (σ : K →+* ℂ) (x : V) {J : Type*} (S : Finset J) (a : J → K) (y : J → V) : h.form σ x (∑ i ∈ S, a i • y i) = ∑ i ∈ S, σ (a i) * h.form σ x (y i)

private lemma **form_sum_smul_left** {K V : Type*} [Field K] [AddCommGroup V] [Module K V] (h : HermFam K V) (σ : K →+* ℂ) (y : V) {J : Type*} (S : Finset J) (a : J → K) (x : J → V) : h.form σ (∑ i ∈ S, a i • x i) y = ∑ i ∈ S, starRingEnd ℂ (σ (a i)) * h.form σ (x i) y

theorem **form_eq** {K : Type} [Field K] {n g : ℕ} (h : HermFam K (Fin n → K)) (s : K →+* ℂ) (Hm : Matrix (Fin g) (Fin g) ℂ) (W : Fin n → Fin g → ℂ) (r : ℂ) (hh : ∀ i j, h.form s (Pi.single i 1) (Pi.single j 1) = r * hermForm Hm (W j) (W i)) (c c' : Fin n → K) : h.form s c c' = r * hermForm Hm (∑ i, s (c' i) • W i) (∑ i, s (c i) • W i)

theorem **form_diff_eq** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (P' : Out.PolAV (Sp (𝒪 K))) (F' : GenFib P') (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)) (f : P.A ⟶ P'.A) (g : P.A ⟶ P.A) (M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)) (N : (Fin F.P₀.g → K) →ₗ[K] (Fin F.P₀.g → K)), F.IsHerm h → F'.IsHerm h' → IsMonHom f → IsMonHom g → LamEq P'.L f f P.L g (𝟙 P.A) → F.IsDiffOf F' f M → F.IsDiffOf F g N → ∀ s c c', h'.form s (M c) (M c') = h.form s c (N c')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HermitianMetricDifferentialEqOfPairingRelationStatement

theorem **_root_.GlobalTheory.T.hermitianMetric_differential_eq_of_pairing_relation** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (P' : Out.PolAV (Sp (𝒪 K))) (F' : GenFib P') (h : HermFam K (Fin F.P₀.g → K)) (h' : HermFam K (Fin F'.P₀.g → K)) (f : P.A ⟶ P'.A) (g : P.A ⟶ P.A) (M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)) (N : (Fin F.P₀.g → K) →ₗ[K] (Fin F.P₀.g → K)), F.IsHerm h → F'.IsHerm h' → IsMonHom f → IsMonHom g → LamEq P'.L f f P.L g (𝟙 P.A) → F.IsDiffOf F' f M → F.IsDiffOf F g N → ∀ s c c', h'.form s (M c) (M c') = h.form s c (N c')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HolomorphicUniformisationTangentUnimodularRiemanStatement

theorem **_root_.GlobalTheory.T.holomorphicUniformisation_tangent_unimodular_riemannForm** : ThetaUnifAV → ∀ (P : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g), (∀ v, IsTangent v → ∃! w, u.IsDiff w v) ∧ (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) ∧ ∀ (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ), u.IsAH Hm χ → ∃ Ef : LinearMap.BilinForm ℤ u.Λ, (∀ a b : u.Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) ∧ Function.Bijective Ef

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HolomorphicUniformizationTangentUnimodularRiemann

theorem **injective_of_ker_le_discrete** {g g' : ℕ} (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ] (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)) (h : ∀ z, F z = 0 → z ∈ Λ) : Function.Injective F

theorem **lbIso_pow_pull_id** {X : Scheme.{0}} {L : LB X} (hL : L.IsLB) (n : ℕ) : LBIso (lbPow (L.pull (𝟙 X)) n) (lbPow L n)

theorem **tangent_and_unimodular** : ThetaUnifAV → ∀ (P : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g), (∀ v, IsTangent v → ∃! w, u.IsDiff w v) ∧ (∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) ∧ ∀ (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ), u.IsAH Hm χ → ∃ Ef : LinearMap.BilinForm ℤ u.Λ, (∀ a b : u.Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) ∧ Function.Bijective Ef

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HomDerivBaseChange

theorem **hom_deriv_baseChange** : ∀ (R : Type) [CommRing R] (P Q : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (t : epsR R ⟶ Q.A) (R₀ : Type) [CommRing R₀] (ρ : R →+* R₀) (P₀ Q₀ : Out.PolAV (Sp R₀)) (φ : P₀.A.left ⟶ P.A.left) (ψ : Q₀.A.left ⟶ Q.A.left) (v₀ : epsR R₀ ⟶ P₀.A) (t₀ : epsR R₀ ⟶ Q₀.A) (f : P₀.A ⟶ Q₀.A) (a : R₀), IsPolFibre ρ P P₀ φ → IsPolFibre ρ Q Q₀ ψ → IsVecOver ρ φ v v₀ → IsVecOver ρ ψ t t₀ → IsMonHom f → IsLieComb ![a] ![t₀] (v₀ ≫ f) → ∀ (R₁ : Type) [CommRing R₁] (g : R₀ →+* R₁), (∀ (P₁ Q₁ : Out.PolAV (Sp R₁)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (v₁ : epsR R₁ ⟶ P₁.A) (t₁ : epsR R₁ ⟶ Q₁.A), IsPolFibre (g.comp ρ) P P₁ φ₁ → IsPolFibre (g.comp ρ) Q Q₁ ψ₁ → IsVecOver (g.comp ρ) φ₁ v v₁ → IsVecOver (g.comp ρ) ψ₁ t t₁ → ∃ f₁ : P₁.A ⟶ Q₁.A, IsMonHom f₁ ∧ IsLieComb ![g a] ![t₁] (v₁ ≫ f₁))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HomDerivDescendsToNumberField

theorem **isLieComb_one_iff** {G : Over (Sp R)} [GrpObj G] (a : R) (t w : epsR R ⟶ G) : IsLieComb ![a] ![t] w ↔ w.left = spMap (epsScaleR a) ≫ t.left

theorem **epsScaleR_comp_dualMapR** (g : R →+* R') (a : R) : (epsScaleR (g a)).comp (Chart.dualMapR g) = (Chart.dualMapR g).comp (epsScaleR a)

theorem **isLieComb_pull** {g : R →+* R'} {G H : Over (Sp R)} {G' H' : Over (Sp R')} [GrpObj H] [GrpObj H'] {φ : G'.left ⟶ G.left} {ψ : H'.left ⟶ H.left} (hψ : IsPullback ψ H'.hom H.hom (spMap g)) {v : epsR R ⟶ G} {t : epsR R ⟶ H} {v' : epsR R' ⟶ G'} {t' : epsR R' ⟶ H'} (hv : IsVecOver g φ v v') (ht : IsVecOver g ψ t t') {f : G ⟶ H} {f' : G' ⟶ H'} (hf : f'.left ≫ ψ = φ ≫ f.left) {a : R} (h : IsLieComb ![a] ![t] (v ≫ f)) : IsLieComb ![g a] ![t'] (v' ≫ f')

theorem **exists_descent** (hT : ThetaUnifAV) (hB : Out.PolBaseChangeSt) (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (P Q : Out.PolAV (Sp k)) {Pc Qc : Out.PolAV (Sp ℂ)} {φc : Pc.A.left ⟶ P.A.left} {ψc : Qc.A.left ⟶ Q.A.left} (hP : IsPolFibre (algebraMap k ℂ) P Pc φc) (hQ : IsPolFibre (algebraMap k ℂ) Q Qc ψc) (fc : Pc.A ⟶ Qc.A) [IsMonHom fc] : ∃ (K : IntermediateField k ℂ) (P₁ Q₁ : Out.PolAV (Sp K)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (m : Pc.A.left ⟶ P₁.A.left) (n : Qc.A.left ⟶ Q₁.A.left) (f₁ : P₁.A ⟶ Q₁.A), FiniteDimensional k K ∧ IsPolFibre (algebraMap k K) P P₁ φ₁ ∧ IsPolFibre (algebraMap k K) Q Q₁ ψ₁ ∧ IsPolFibre (algebraMap K ℂ) P₁ Pc m ∧ IsPolFibre (algebraMap K ℂ) Q₁ Qc n ∧ m ≫ φ₁ = φc ∧ n ≫ ψ₁ = ψc ∧ IsMonHom f₁ ∧ fc.left ≫ n = m ≫ f₁.left

theorem **hom_deriv_descends_to_numberField** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (R : Type) [CommRing R] (P Q : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (t : epsR R ⟶ Q.A) (k : IntermediateField ℚ ℂ) (ρ : R →+* k) (α : ℂ), FiniteDimensional ℚ k → IsTangent v → (∀ (R' : Type) [CommRing R'] (f' : R →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ Q.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' Q P' φ' → IsVecOver f' φ' t v' → IsLieBasis ![v']) → (∀ (P₁ Q₁ : Out.PolAV (Sp ℂ)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (v₁ : epsR ℂ ⟶ P₁.A) (t₁ : epsR ℂ ⟶ Q₁.A), IsPolFibre ((algebraMap k ℂ).comp ρ) P P₁ φ₁ → IsPolFibre ((algebraMap k ℂ).comp ρ) Q Q₁ ψ₁ → IsVecOver ((algebraMap k ℂ).comp ρ) φ₁ v v₁ → IsVecOver ((algebraMap k ℂ).comp ρ) ψ₁ t t₁ → ∃ f₁ : P₁.A ⟶ Q₁.A, IsMonHom f₁ ∧ IsLieComb ![α] ![t₁] (v₁ ≫ f₁)) → ∃ (K₁ : IntermediateField ℚ ℂ) (hk : k ≤ K₁) (a : K₁), FiniteDimensiona

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HomDerivativeClosure

theorem **isLieComb_inv** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] {v x : epsR R ⟶ G} {a : R} (hv : IsTangent v) (hx : IsLieComb ![a] ![v] x) : IsLieComb ![-a] ![v] x⁻¹

theorem **hom_derivative_closure** : Out.PolBaseChangeSt → ∀ (R : Type) [CommRing R] (P Q : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (t : epsR R ⟶ Q.A) (R₁ : Type) [CommRing R₁] (ρ : R →+* R₁) (S : Set R₁), IsTangent t → (∀ a ∈ S, (∀ (P₁ Q₁ : Out.PolAV (Sp R₁)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (v₁ : epsR R₁ ⟶ P₁.A) (t₁ : epsR R₁ ⟶ Q₁.A), IsPolFibre ρ P P₁ φ₁ → IsPolFibre ρ Q Q₁ ψ₁ → IsVecOver ρ φ₁ v v₁ → IsVecOver ρ ψ₁ t t₁ → ∃ f₁ : P₁.A ⟶ Q₁.A, IsMonHom f₁ ∧ IsLieComb ![a] ![t₁] (v₁ ≫ f₁))) → ∀ a ∈ AddSubgroup.closure S, (∀ (P₁ Q₁ : Out.PolAV (Sp R₁)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (v₁ : epsR R₁ ⟶ P₁.A) (t₁ : epsR R₁ ⟶ Q₁.A), IsPolFibre ρ P P₁ φ₁ → IsPolFibre ρ Q Q₁ ψ₁ → IsVecOver ρ φ₁ v v₁ → IsVecOver ρ ψ₁ t t₁ → ∃ f₁ : P₁.A ⟶ Q₁.A, IsMonHom f₁ ∧ IsLieComb ![a] ![t₁] (v₁ ≫ f₁))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HomExtDvrOfCardTorsion

theorem **homExtDvr_of_card_torsion** : (∀ (k : Type) [Field k] [IsAlgClosed k] (B : Over (Sp k)) [GrpObj B] (g n : ℕ), IsAbelianScheme B → SmoothOfRelativeDimension g B.hom → (n : k) ≠ 0 → Nat.card {x : (𝟙_ (Over (Sp k)) ⟶ B) // x ^ n = 1} = n ^ (2 * g)) → HomExtDvr

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.HomExtOfLocalRingPoints

theorem **hom_ext_of_localRing_points** : ∀ (X Y : Scheme.{0}) (f g : X ⟶ Y), (∀ (A : Type) [CommRing A] [IsLocalRing A] (t : Sp A ⟶ X), t ≫ f = t ≫ g) → f = g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ISupRangeDifferentialEqTopOfJointlyOnto

theorem **eq_top_of_countable** {V : Type} [AddCommGroup V] [Module ℂ V] (S : Submodule ℂ V) {Λ : Set V} (hΛ : Λ.Countable) (h : ∀ z, ∃ s ∈ S, z - s ∈ Λ) : S = ⊤

theorem **eq_top_of_span_image** {K L : Type} [Field K] [Field L] (σ : K →+* L) {g : ℕ} (N : Submodule K (Fin g → K)) (h : Submodule.span L ((fun m k => σ (m k)) '' (N : Set (Fin g → K))) = ⊤) : N = ⊤

theorem **ex_zero** : u.ex 0 = 1

theorem **ex_sum** : ∀ {n : ℕ} (z : Fin n → Fin g → ℂ), u.ex (∑ m, z m) = (List.ofFn fun m => u.ex (z m)).prod | 0, z => by simpa using ex_zero u | n + 1, z => by rw [Fin.sum_univ_succ, u.ex_add, ex_sum, List.ofFn_succ, List.prod_cons] theorem sub_mem_of_ex_eq {z w : Fin g → ℂ} (h : u.ex z = u.ex w) : z - w ∈ u.Λ

theorem **surjective_of_isLieBasis** (hd : ∀ v, IsTangent v → ∃! w, u.IsDiff w v) (he : ∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) {n : ℕ} {v : Fin n → (epsR ℂ ⟶ A)} (hv : IsLieBasis v) {W : Fin n → Fin g → ℂ} (hW : ∀ i, u.IsDiff (W i) (v i)) : Function.Surjective (Fintype.linearCombination ℂ W)

theorem **exists_fibre** (hU : ThetaUnifAV) (hB : Out.PolBaseChangeSt) {K : Type} [Field K] (σ : K →+* ℂ) (P : Out.PolAV (Sp K)) {v : Fin P.g → (epsR K ⟶ P.A)} (hv : IsLieBasis v) : ∃ (Q : Out.PolAV (Sp ℂ)) (ψ : Q.A.left ⟶ P.A.left) (v' : Fin P.g → (epsR ℂ ⟶ Q.A)) (u : HolUnif Q.A Q.L Q.g) (W : Fin P.g → Fin Q.g → ℂ), IsPolFibre σ P Q ψ ∧ (∀ i, IsTangent (v' i)) ∧ (∀ i, IsVecOver σ ψ (v i) (v' i)) ∧ (∀ x, IsTangent x → ∃! w, u.IsDiff w x) ∧ (∀ w, ∃! x, IsTangent x ∧ u.IsDiff w x) ∧ (∀ i, u.IsDiff (W i) (v' i)) ∧ Function.Bijective (Fintype.linearCombination ℂ W)

theorem **exists_prod_of_ontoAt** {R : Type} [CommRing R] {s : R →+* ℂ} {I : Type} {B : I → Over (Sp R)} {A : Over (Sp R)} [GrpObj A] {B' : I → Over (Sp ℂ)} {A' : Over (Sp ℂ)} [GrpObj A'] {f : ∀ i, B i ⟶ A} {f' : ∀ i, B' i ⟶ A'} {φ : A'.left ⟶ A.left} {ψ : ∀ i, (B' i).left ⟶ (B i).left} (hφ : IsGrpPullback A' A (spMap s) φ) (hψ : ∀ i, IsPullback (ψ i) (B' i).hom (B i).hom (spMap s)) (hf : ∀ i, (f' i).left ≫ φ = ψ i ≫ (f i).left) (h : OntoAt f (spMap s)) (y : 𝟙_ (Over (Sp ℂ)) ⟶ A') : ∃ (n : ℕ) (i : Fin n → I) (x : ∀ m, 𝟙_ (Over (Sp ℂ)) ⟶ B' (i m)), y = (List.ofFn fun m => x m ≫ f' (i m)).prod

theorem **iSup_range_diff_eq_top** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (σ : K →+* ℂ) (I : Type) (Q : I → Out.PolAV (Sp (𝒪 K))) (FQ : ∀ i, GenFib (Q i)) (f : ∀ i, (Q i).A ⟶ P.A) (M : ∀ i, (Fin (FQ i).P₀.g → K) →ₗ[K] (Fin F.P₀.g → K)), Finite I → (∀ i, IsMonHom (f i)) → (∀ i, (FQ i).IsDiffOf F (f i) (M i)) → OntoAt f (spMap (σ.comp (algebraMap (𝒪 K) K))) → ⨆ i, LinearMap.range (M i) = ⊤

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ISupRangeDifferentialEqTopOfJointlyOntoStatement

theorem **_root_.GlobalTheory.T.iSup_range_differential_eq_top_of_jointly_onto** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (σ : K →+* ℂ) (I : Type) (Q : I → Out.PolAV (Sp (𝒪 K))) (FQ : ∀ i, GenFib (Q i)) (f : ∀ i, (Q i).A ⟶ P.A) (M : ∀ i, (Fin (FQ i).P₀.g → K) →ₗ[K] (Fin F.P₀.g → K)), Finite I → (∀ i, IsMonHom (f i)) → (∀ i, (FQ i).IsDiffOf F (f i) (M i)) → OntoAt f (spMap (σ.comp (algebraMap (𝒪 K) K))) → ⨆ i, LinearMap.range (M i) = ⊤

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ImDiagEqTrace

theorem **im_diag_eq_trace** {E : Type} [Field E] [NumberField E] [IsCMField E] {r g : ℕ} (q : ℕ) (ξ : E) (Gm : Matrix (Fin r) (Fin r) E) (Ψ : Finset (E →+* ℂ)) (hΨ : ∀ τ : E →+* ℂ, τ ∈ Ψ ↔ ComplexEmbedding.conjugate τ ∉ Ψ) (idx : Fin g ≃ Ψ × Fin r) (hξ : FD.cc ξ = -ξ) (x : FD.Frame Ψ r) (hx : ∀ τ : Ψ, FD.gram τ.1 Gm (FD.mat x τ) = Matrix.diagonal fun a : Fin r => ((FD.sgn q (a : ℕ) : ℝ) : ℂ)) (v w : Fin r → E) (z z' : Fin g → ℂ) (hz : ∀ k, z k = if ((idx k).2 : ℕ) < q then FD.hermC (idx k).1.1 Gm (FD.col (FD.mat x (idx k).1) (idx k).2) (FD.Data.emb (idx k).1.1 v) else ((FD.mat x (idx k).1)⁻¹ *ᵥ FD.Data.emb (idx k).1.1 v) (idx k).2) (hz' : ∀ k, z' k = if ((idx k).2 : ℕ) < q then FD.hermC (idx k).1.1 Gm (FD.col (FD.mat x (idx k).1) (idx k).2) (FD.Data.emb (idx k).1.1 w) else ((FD.mat x (idx k).1)⁻¹ *ᵥ FD.Data.emb (idx k).1.1 w) (idx k).2) : (hermForm (Matrix.diagonal fun k => ((2 * ((idx k).1

theorem **hm_eq_smul_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (ξ : ℝ), GoodEmb c σ → IsUnf U → ∀ (k : U.Comp) (x : (U.X k).dom) (W : PullbackSections.UnfAt U k x) (Wu : PullbackSections.UnitAt W), (∀ τ : E →+* ℂ, ‖τ (U.X k).ξ0‖ = ξ) → (U.X k).IsOrtho x.1 → Wu.Hm = ((2 * ξ : ℝ) : ℂ) • (1 : Matrix (Fin W.g) (Fin W.g) ℂ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.InjectiveOfSumEq

theorem **finrank_le_sum** {N : J → Submodule K V} (h : ⊔ i, N i = ⊤) : Module.finrank K V ≤ ∑ i, Module.finrank K (N i)

theorem **iSupIndep_of_sum_le** {N : J → Submodule K V} (h : ⊔ i, N i = ⊤) (hd : ∑ i, Module.finrank K (N i) ≤ Module.finrank K V) : iSupIndep N

theorem **sum_finrank_eq** {N : J → Submodule K V} (hi : iSupIndep N) (h : ⊔ i, N i = ⊤) : ∑ i, Module.finrank K (N i) = Module.finrank K V

theorem **injective_of_sum_eq** {W : J → Type} [∀ i, AddCommGroup (W i)] [∀ i, Module K (W i)] [∀ i, FiniteDimensional K (W i)] {M : ∀ i, W i →ₗ[K] V} (h : ⊔ i, LinearMap.range (M i) = ⊤) (hd : ∑ i, Module.finrank K (W i) = Module.finrank K V) (j : J) : Function.Injective (M j)

theorem **map_eigLat_le** {K : Type} [Field K] [NumberField K] {P P' : Out.PolAV (Sp (𝒪 K))} {F : GenFib P} {F' : GenFib P'} {O : Type} [CommRing O] {a : OAct O P.A} {a' : OAct O P'.A} {f : P.A ⟶ P'.A} (hf : ∀ x, a.act x ≫ f = f ≫ a'.act x) {M : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)} (hM : F.IsDiffOf F' f M) (τ : O →+* 𝒪 K) : (F.eigLat a τ).map (M.restrictScalars (𝒪 K)) ≤ F'.eigLat a' τ

theorem **letters_independent** : ThetaUnifAV → Out.Laws → IndSt

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegersOfCmFieldConjStructure

theorem **int_dvd_of_mul_eq_ringOfIntegers** {N m : ℤ} {z : 𝒪 E} (h : (N : 𝒪 E) * z = m) : N ∣ m

theorem **int_dvd_of_sq_dvd_squarefree_mul_sq** {d N v : ℤ} (hd : Squarefree d) (hN : N ≠ 0) (h : N ^ 2 ∣ d * v ^ 2) : N ∣ v

theorem **cm_generator_sq_quadratic_eq** (H : Host E) : H.w0 ^ 2 + (((1 + H.d) / 4 : ℕ) : 𝒪 E) = H.w0

theorem **exists_coord** (H : Host E) {a : 𝒪 E} (ha : a ∈ OK H) : ∃ u v : ℤ, a = u + v * H.w0

theorem **coord_eq_zero** (H : Host E) {u v : ℤ} (h : (u : 𝒪 E) + v * H.w0 = 0) : u = 0 ∧ v = 0

theorem **coord_mem** (H : Host E) (u v : ℤ) : (u : 𝒪 E) + v * H.w0 ∈ OK H

theorem **coord_mul_bar** (H : Host E) (u v : ℤ) : ((u : 𝒪 E) + v * H.w0) * H.bar (u + v * H.w0) = ((u ^ 2 + u * v + ((1 + H.d) / 4 : ℕ) * v ^ 2 : ℤ) : 𝒪 E)

theorem **mem_OK_of_int_mul_mem** (H : Host E) {N : ℤ} (hN : N ≠ 0) {y : 𝒪 E} (hy : (N : 𝒪 E) * y ∈ OK H) : y ∈ OK H

theorem **finite_int_OK** (H : Host E) : Module.Finite ℤ (OK H)

theorem **isIntegrallyClosed_OK** (H : Host E) : IsIntegrallyClosed (OK H)

theorem **isDedekindDomain_OK** (H : Host E) : IsDedekindDomain (OK H)

theorem **existsUnique_coord** (H : Host E) (a : OK H) : ∃! m : ℤ × ℤ, a = (m.1 : OK H) + (m.2 : OK H) * wK H

theorem **finrank_int_OK** (H : Host E) : Module.finrank ℤ (OK H) = 2

theorem **two_mul_finrank_OK** (H : Host E) (M : Type) [AddCommGroup M] [Module (𝒪 E) M] [Module.Finite (𝒪 E) M] : 2 * Module.finrank (OK H) M = Module.finrank ℚ E * Module.finrank (𝒪 E) M

theorem **OK_isDedekindDomain_and_projective** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E), IsDedekindDomain (OK H) ∧ (∀ a : OK H, ∃! m : ℤ × ℤ, a = (m.1 : OK H) + (m.2 : OK H) * wK H) ∧ (∃ br : OK H ≃+* OK H, (∀ a, ((br a : OK H) : 𝒪 E) = H.bar a) ∧ wK H + br (wK H) = 1 ∧ ∀ a, br (br a) = a) ∧ wK H ^ 2 + (((1 + H.d) / 4 : ℕ) : OK H) = wK H ∧ 2 * Module.finrank (OK H) (𝒪 E) = Module.finrank ℚ E ∧ ∀ (M : Type) [AddCommGroup M] [Module (𝒪 E) M], Module.Finite (𝒪 E) M → Module.Projective (𝒪 E) M → Module.Finite (OK H) M ∧ Module.Projective (OK H) M ∧ Module.finrank (OK H) M = Module.finrank ℚ E / 2 * Module.finrank (𝒪 E) M

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegersOfCmFieldConjStructureStatement

theorem **_root_.GlobalTheory.T.integers_of_cm_field_conj_structure** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E), IsDedekindDomain (OK H) ∧ (∀ a : OK H, ∃! m : ℤ × ℤ, a = (m.1 : OK H) + (m.2 : OK H) * wK H) ∧ (∃ br : OK H ≃+* OK H, (∀ a, ((br a : OK H) : 𝒪 E) = H.bar a) ∧ wK H + br (wK H) = 1 ∧ ∀ a, br (br a) = a) ∧ wK H ^ 2 + (((1 + H.d) / 4 : ℕ) : OK H) = wK H ∧ 2 * Module.finrank (OK H) (𝒪 E) = Module.finrank ℚ E ∧ ∀ (M : Type) [AddCommGroup M] [Module (𝒪 E) M], Module.Finite (𝒪 E) M → Module.Projective (𝒪 E) M → Module.Finite (OK H) M ∧ Module.Projective (OK H) M ∧ Module.finrank (OK H) M = Module.finrank ℚ E / 2 * Module.finrank (𝒪 E) M

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralEigenbasisBaseChangeDvr

theorem **flat_of_inj** {A B : Type} [CommRing A] [IsDomain A] [IsBezout A] [CommRing B] [IsDomain B] [Algebra A B] (hi : Function.Injective (algebraMap A B)) : Module.Flat A B

theorem **eig_flat** {V V' : Type} [CommRing V] [IsNoetherianRing V] [CommRing V'] [Algebra V V'] [Module.Flat V V'] {n m : ℕ} {J : Type} (L : J → (Fin n → V) →ₗ[V] (Fin n → V)) (χ : J → V) (L' : J → (Fin n → V') →ₗ[V'] (Fin n → V')) (hL : ∀ a c, L' a (algebraMap V V' ∘ c) = algebraMap V V' ∘ L a c) (b : Fin m → Fin n → V) (hb : ∀ k a, L a (b k) = χ a • b k) (hbb : ∀ c, (∀ a, L a c = χ a • c) → ∃! r : Fin m → V, c = ∑ k, r k • b k) (w : Fin n → V') (hw : ∀ a, L' a w = algebraMap V V' (χ a) • w) : ∃! r : Fin m → V', w = ∑ k, r k • (algebraMap V V' ∘ b k)

theorem **univChart_iso** {S : Type} [CommRing S] {G G' : Over (Spec (CommRingCat.of S))} [GrpObj G] [GrpObj G'] (e : G' ≅ G) [IsMonHom e.hom] {n : ℕ} {x : ∀ m : ℕ, sp (R₀

theorem **exists_charts** {R S : Type} [CommRing R] [IsLocalRing R] [CommRing S] {G₀ : Over (Sp R)} [GrpObj G₀] [Smooth G₀.hom] {G : Over (Sp S)} [GrpObj G] (f : R →+* S) {φ : G.left ⟶ G₀.left} (h : IsGrpPullback G G₀ (spMap f) φ) : ∃ (n : ℕ) (x : ∀ m : ℕ, sp (R₀

theorem **sp_comp** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

theorem **dual_alg** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) : spMap (Formalism.L2.Pj.dualMap ψ) ≫ spMap (algebraMap S (DualNumber S)) = spMap (algebraMap T (DualNumber T)) ≫ spMap ψ

theorem **dual_comp** {R S T : Type} [CommRing R] [CommRing S] [CommRing T] (f : S →+* T) (g : R →+* S) : spMap (Formalism.L2.Pj.dualMap f) ≫ spMap (Formalism.L2.Pj.dualMap g) = spMap (Formalism.L2.Pj.dualMap (f.comp g))

theorem **lieInt_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), ∀ (T T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr), (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr) → ∀ (x : Sp T.Fr ⟶ Xo.X) (x' : Sp T'.Fr ⟶ Xo.X) (G : PAT H (Sp T.V)) (G' : PAT H (Sp T'.V)) (t : SAB H (Sp T.Fr)) (t' : SAB H (Sp T'.Fr)) (m1 : SABMap t Xo.univ.t x) (m2 : SABMap t G.t (spMap (algebraMap T.V T.Fr))) (m1' : SABMap t' Xo.univ.t x') (m2' : SABMap t' G'.t (spMap (algebraMap T'.V T'.Fr))) (n : SABMap G'.t G.t (spMap ψ)) (n' : SABMap t' t (spMap i)), x' = spMap i ≫ x → m1'.φ = n'.φ ≫ m1.φ → m2'.φ ≫ n.φ = n'.φ ≫ m2.φ → ∀ (io : c.F →+* T.Fr) (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber T.Fr) ⟶ Xo.univ.t.G.left)), LieInt T io m1 m2 u → LieInt T' (i.comp io) m1' m2' (fun τ k => spMap (Formalism.L2.Pj.

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralEigenbasisBaseChangeDvrStatement

theorem **_root_.GlobalTheory.T.integral_eigenbasis_baseChange_dvr** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), ∀ (T T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr), (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr) → ∀ (x : Sp T.Fr ⟶ Xo.X) (x' : Sp T'.Fr ⟶ Xo.X) (G : PAT H (Sp T.V)) (G' : PAT H (Sp T'.V)) (t : SAB H (Sp T.Fr)) (t' : SAB H (Sp T'.Fr)) (m1 : SABMap t Xo.univ.t x) (m2 : SABMap t G.t (spMap (algebraMap T.V T.Fr))) (m1' : SABMap t' Xo.univ.t x') (m2' : SABMap t' G'.t (spMap (algebraMap T'.V T'.Fr))) (n : SABMap G'.t G.t (spMap ψ)) (n' : SABMap t' t (spMap i)), x' = spMap i ≫ x → m1'.φ = n'.φ ≫ m1.φ → m2'.φ ≫ n.φ = n'.φ ≫ m2.φ → ∀ (io : c.F →+* T.Fr) (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber T.Fr) ⟶ Xo.univ.t.G.left)), LieInt T io m1 m2 u → LieInt T' (i.comp io) m1'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralFormsLogSupNormLowerBoundAbsolute

theorem **lamwAt_absolute** : ∃ CL : ℝ, 0 < CL ∧ ∀ (D : D7) (T : Top D) (h : LawsP T) (large₂ : SizeData₂ D) (CA : ℝ), ParA large₂ CA → ∀ (j : ℕ) (w : WeightFamilies.Wt D j), LAMwAt large₂ (recB h.toLawsCore) j w CL 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralFormsLogSupNormLowerBoundAbsoluteStatement

theorem **_root_.GlobalTheory.T.integralForms_logSupNorm_lower_bound_absolute** : ∃ CL : ℝ, 0 < CL ∧ ∀ (D : D7) (T : Top D) (h : LawsP T) (large₂ : SizeData₂ D) (CA : ℝ), ParA large₂ CA → ∀ (j : ℕ) (w : WeightFamilies.Wt D j), LAMwAt large₂ (recB h.toLawsCore) j w CL 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralGeneratorAssociatedLocalGeneratorCMLine

theorem **fibre_ne_of_generates** {R : Type} [CommRing R] [IsLocalRing R] (P : Out.PolAV (Sp R)) (θ : P.L.Sec) (h : ∀ σ : P.L.Sec, ∃ c : ∀ i, Γ(P.A.left, P.L.U i), ∀ i, σ.f i = c i * θ.f i) (s : Sp R) : ∃ q : P.A.left, P.A.hom q = s ∧ q ∈ Out.nonzeroLocus θ

theorem **isLieBasis_cast** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] {n k : ℕ} (e : n = k) {v : Fin k → (epsR R ⟶ G)} (h : IsLieBasis v) : IsLieBasis fun i : Fin n => v (i.cast e)

theorem **isLieComb_cast** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] {n k : ℕ} (e : n = k) (B : Matrix (Fin k) (Fin k) R) {v w : Fin k → (epsR R ⟶ G)} (h : ∀ i, IsLieComb (B i) v (w i)) (i : Fin n) : IsLieComb ((Matrix.of fun i j : Fin n => B (i.cast e) (j.cast e)) i) (fun i : Fin n => v (i.cast e)) (w (i.cast e))

theorem **det_cast** {R : Type} [CommRing R] {n k : ℕ} (e : n = k) (B : Matrix (Fin k) (Fin k) R) : (Matrix.of fun i j : Fin n => B (i.cast e) (j.cast e)).det = B.det

theorem **eps_sq** {R R' : Type} [CommRing R] [CommRing R'] (f : R →+* R') : spMap (Chart.dualMapR f) ≫ (epsR R).hom = (epsR R').hom ≫ spMap f

theorem **stalk_of_isoMaps** {X X' : Scheme.{0}} {L : LB X} {L' : LB X'} {F : X' ⟶ X} {w : LB.Iso L' (L.pull F)} {θ : L.Sec} {θ' : (L.pull F).Sec} {σ : L'.Sec} (h : IsoMaps w σ θ') (hθ : PullSec F θ θ') (Z : Scheme.{0}) (r : Z ⟶ X') (z : Z) (j : L.ι) (k : L'.ι) (hj : F (r z) ∈ L.U j) (hk : r z ∈ L'.U k) : gs (r ≫ F) z _ (θ.f j) = (gw w r z j k).val * gs r z _ (σ.f k)

theorem **keyPB_of_sabMap** {R R' : Type} [CommRing R] [CommRing R'] {f : R →+* R'} {T : PAT H (Sp R)} {T' : PAT H (Sp R')} (m : SABMap T'.t T.t (spMap f)) {w : LB.Iso T'.t.PP.L (T.t.PP.L.pull m.φ)} (hw : RigIso m w) (hg : T'.g = T.g) {g₀ th hth v hv g₀' th' hth' v' hv'} (hs : ∀ (Z : Scheme.{0}) (r : Z ⟶ T'.t.G.left) (z : Z) j k, m.φ (r z) ∈ T.t.PP.L.U j → r z ∈ T'.t.PP.L.U k → gs (r ≫ m.φ) z _ (LB.Sec.f th j) = (gw w r z j k).val * gs r z _ (LB.Sec.f th' k)) (hl : ∀ i, (v' i).left ≫ m.φ = spMap (Chart.dualMapR f) ≫ (v (i.cast hg)).left) : KeyPB (T'.pol, g₀', T'.t.PP.rig, T'.t.PP.rigGen, th', hth', v', hv') (T.pol, g₀, T.t.PP.rig, T.t.PP.rigGen, th, hth, v, hv) f

theorem **eq_mul_of_secTie** {S S' : Type} [CommRing S] [CommRing S'] (φ : S →+* S') {x x₀ : Sp S ⟶ Xo.X} {x₁ x₂ : Sp S' ⟶ Xo.X} {t : SAB H (Sp S)} {t' : SAB H (Sp S')} {m : SABMap t Xo.univ.t x} {m' : SABMap t' Xo.univ.t (spMap φ ≫ x)} {n : SABMap t' t (spMap φ)} (hφ : m'.φ = n.φ ≫ m.φ) {th : LB.Sec t.PP.L} {s₀ : LB.Sec (PPx x₀)} {s₁ : LB.Sec (PPx x₁)} {s₂ : LB.Sec (PPx x₂)} {a : S'} (e₀ : x₀ = x) (e₁ : x₁ = spMap φ ≫ x₀) (e₂ : x₂ = x₁) (h₀ : SecTie m th s₀) (b₁ : SecBC φ 1 x₀ x₁ s₀ s₁) (b₂ : SecBC (RingHom.id S') a x₁ x₂ s₁ s₂) {w : LB.Iso t'.PP.L (t.PP.L.pull n.φ)} (hw : RigIso n w) {th' σ : LB.Sec t'.PP.L} (hs : ∀ (Z : Scheme.{0}) (r : Z ⟶ t'.G.left) (z : Z) j k, n.φ (r z) ∈ t.PP.L.U j → r z ∈ t'.PP.L.U k → gs (r ≫ n.φ) z _ (th.f j) = (gw w r z j k).val * gs r z _ (th'.f k)) (h₂ : SecTie m' σ s₂) : ∀ i, σ.f i = constFn t'.G.hom _ a * th'.f i

theorem **integral_generator_unit** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (p : HeightOneSpectrum (𝒪 z.K)) (T : CDVR) (iK : z.K →+* T.Fr) (M : LocMod z T iK), (∀ y : z.K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation z.K y ≤ 1) → letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralGeneratorAssociatedLocalGeneratorCMLineStatement

theorem **_root_.GlobalTheory.T.integralGenerator_associated_localGenerator_CM_line** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (p : HeightOneSpectrum (𝒪 z.K)) (T : CDVR) (iK : z.K →+* T.Fr) (M : LocMod z T iK), (∀ y : z.K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation z.K y ≤ 1) → letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralGeneratorBaseChangeValuationRing

theorem **secInt_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l), ∀ (T T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr), (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr) → ∀ (x : Sp T.Fr ⟶ Xo.X) (x' : Sp T'.Fr ⟶ Xo.X) (G : PAT H (Sp T.V)) (G' : PAT H (Sp T'.V)) (t : SAB H (Sp T.Fr)) (t' : SAB H (Sp T'.Fr)) (m1 : SABMap t Xo.univ.t x) (m2 : SABMap t G.t (spMap (algebraMap T.V T.Fr))) (m1' : SABMap t' Xo.univ.t x') (m2' : SABMap t' G'.t (spMap (algebraMap T'.V T'.Fr))) (n : SABMap G'.t G.t (spMap ψ)) (n' : SABMap t' t (spMap i)), x' = spMap i ≫ x → m1'.φ = n'.φ ≫ m1.φ → m2'.φ ≫ n.φ = n'.φ ≫ m2.φ → ∀ (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')), SecBC i 1 x x' s s' → SecInt T m1 m2 s → SecInt T' m1' m2' s'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralGeneratorBaseChangeValuationRingStatement

theorem **_root_.GlobalTheory.T.integral_generator_baseChange_valuationRing** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l), ∀ (T T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr), (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr) → ∀ (x : Sp T.Fr ⟶ Xo.X) (x' : Sp T'.Fr ⟶ Xo.X) (G : PAT H (Sp T.V)) (G' : PAT H (Sp T'.V)) (t : SAB H (Sp T.Fr)) (t' : SAB H (Sp T'.Fr)) (m1 : SABMap t Xo.univ.t x) (m2 : SABMap t G.t (spMap (algebraMap T.V T.Fr))) (m1' : SABMap t' Xo.univ.t x') (m2' : SABMap t' G'.t (spMap (algebraMap T'.V T'.Fr))) (n : SABMap G'.t G.t (spMap ψ)) (n' : SABMap t' t (spMap i)), x' = spMap i ≫ x → m1'.φ = n'.φ ≫ m1.φ → m2'.φ ≫ n.φ = n'.φ ≫ m2.φ → ∀ (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')), SecBC i 1 x x' s s' → SecInt T m1 m2 s → SecInt T' m1' m2' s'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralTangentVectorCoordinates

theorem **sc_comm** (a b : R) : sc a ≫ sc b = sc b ≫ sc a

theorem **sc_one** : sc (1 : R) = 𝟙 _

theorem **isLieComb_iff** {n : ℕ} (c : Fin n → R) (v : Fin n → (epsR R ⟶ G)) (w : epsR R ⟶ G) : IsLieComb c v w ↔ w = (List.ofFn fun i => sc (c i) ≫ v i).prod

theorem **isLieComb_one_iff** (c : R) (v w : epsR R ⟶ G) : IsLieComb ![c] ![v] w ↔ w = sc c ≫ v

theorem **isLieComb_mul** (v w : epsR R ⟶ G) : IsLieComb ![1, 1] ![v, w] (v * w)

theorem **isLieComb_nil** : IsLieComb (![] : Fin 0 → R) (![] : Fin 0 → (epsR R ⟶ G)) 1

theorem **eq_of_isVecOver** {R' : Type} [CommRing R'] {f : R →+* R'} (hf : Function.Injective f) {A : Over (Sp R)} [IsSeparated A.hom] {A' : Over (Sp R')} {φ : A'.left ⟶ A.left} {u u' : epsR R ⟶ A} {w : epsR R' ⟶ A'} (h : IsVecOver f φ u w) (h' : IsVecOver f φ u' w) : u = u'

theorem **isCoord_comb** {n : ℕ} {u : Fin n → (epsR (𝒪 K) ⟶ P.A)} {c : Fin n → Fin F.P₀.g → K} (h : ∀ i, F.IsCoord (u i) (c i)) {a : Fin n → 𝒪 K} {u' : epsR (𝒪 K) ⟶ P.A} (ha : IsLieComb a u u') : F.IsCoord u' (∑ i, a i • c i)

theorem **isCoord_one** : F.IsCoord 1 0

theorem **isCoord_mul** {u u' : epsR (𝒪 K) ⟶ P.A} {c c' : Fin F.P₀.g → K} (h : F.IsCoord u c) (h' : F.IsCoord u' c') : F.IsCoord (u * u') (c + c')

theorem **isCoord_smul** {u u' : epsR (𝒪 K) ⟶ P.A} {c : Fin F.P₀.g → K} {r : 𝒪 K} (h : F.IsCoord u c) (h' : IsLieComb ![r] ![u] u') : F.IsCoord u' (r • c)

theorem **mem_span_iff** (p : (epsR (𝒪 K) ⟶ P.A) → Prop) (h1 : p 1) (hm : ∀ u u', p u → p u' → p (u * u')) (hs : ∀ r u, p u → p (sc r ≫ u)) (c : Fin F.P₀.g → K) : c ∈ Submodule.span (𝒪 K) {c | ∃ u, F.IsCoord u c ∧ p u} ↔ ∃ u, F.IsCoord u c ∧ p u

theorem **integral_tangent_coordinates** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P), (∀ u, IsTangent u → ∃! c, F.IsCoord u c) ∧ (∀ u u' c, F.IsCoord u c → F.IsCoord u' c → u = u') ∧ (∀ u u' c c', F.IsCoord u c → F.IsCoord u' c' → F.IsCoord (u * u') (c + c')) ∧ (∀ u u' c (r : 𝒪 K), F.IsCoord u c → IsLieComb ![r] ![u] u' → F.IsCoord u' (r • c)) ∧ (∀ c, c ∈ F.lieLat ↔ ∃ u, F.IsCoord u c) ∧ ∀ (O : Type) [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K) c, c ∈ F.eigLat act τ ↔ ∃ u, F.IsCoord u c ∧ ∀ a, IsLieComb ![τ a] ![u] (u ≫ act.act a)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IntegralTangentVectorCoordinatesStatement

theorem **_root_.GlobalTheory.T.integral_tangent_vector_coordinates** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P), (∀ u, IsTangent u → ∃! c, F.IsCoord u c) ∧ (∀ u u' c, F.IsCoord u c → F.IsCoord u' c → u = u') ∧ (∀ u u' c c', F.IsCoord u c → F.IsCoord u' c' → F.IsCoord (u * u') (c + c')) ∧ (∀ u u' c (r : 𝒪 K), F.IsCoord u c → IsLieComb ![r] ![u] u' → F.IsCoord u' (r • c)) ∧ (∀ c, c ∈ F.lieLat ↔ ∃ u, F.IsCoord u c) ∧ ∀ (O : Type) [CommRing O] (act : OAct O P.A) (τ : O →+* 𝒪 K) c, c ∈ F.eigLat act τ ↔ ∃ u, F.IsCoord u c ∧ ∀ a, IsLieComb ![τ a] ![u] (u ≫ act.act a)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.InvMulVecEqZero

theorem **emb_smul'** (τ : E →+* ℂ) (a : E) (v : Fin r → E) : FD.Data.emb τ (a • v) = τ a • FD.Data.emb τ v

theorem **inv_mulVec_eq_zero** {p : ℕ} (hp : p ≤ r) (M : Matrix (Fin r) (Fin r) ℂ) (hM : M.det ≠ 0) {u : Fin r → ℂ} (hu : u ∈ Submodule.span ℂ (Set.range fun a : Fin p => FD.col M (Fin.castLE hp a))) (k : Fin r) (hk : p ≤ (k : ℕ)) : (M⁻¹.mulVec u) k = 0

theorem **perFr_pure** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : FD.DataX E r) (La : ℝ) (x : FD.Frame X.Ψ r) (lam : Fin r → Fin r → E), x ∈ X.toData.dom → FD.LamBasis X La x lam → ∀ (a : E) (j : Fin r) (τ : E →+* ℂ) (i : ℕ), ((j : ℕ) < X.p0 → τ ∈ X.Ψ → perFr X.toData x (a • lam j) τ i = 0) ∧ (X.p0 ≤ (j : ℕ) → τ ∉ X.Ψ → i < X.p0 → perFr X.toData x (a • lam j) τ i = 0)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsAbelianSchemeAndExtendOfGenericHoms

theorem **isIso_toUnit_splitTorus_punit** (S : Scheme.{0}) : IsIso (toUnit (splitTorus S PUnit))

theorem **isAbelianScheme_pullback_obj** {S S' : Scheme.{0}} (f : S' ⟶ S) (A : Over S) [GrpObj A] (h : IsAbelianScheme A) : IsAbelianScheme ((Over.pullback f).obj A) where

theorem **isSemiAbelian_of_isAbelianScheme** {S : Scheme.{0}} (A : Over S) [GrpObj A] (h : IsAbelianScheme A) : IsSemiAbelian A where

theorem **isAbelianScheme_powObj_of_isAbelianScheme** {S : Scheme.{0}} (A : Over S) [GrpObj A] (h : IsAbelianScheme A) : ∀ n, IsAbelianScheme (Out.powObj A n) | 0 => Diag.isAbelianScheme_unit S | n + 1 => AbelianSchemeGerms.isAbelianScheme_powObj A h n theorem comp_list_prod_ofFn {S : Scheme.{0}} {T T' G : Over S} [MonObj G] (t : T' ⟶ T) : ∀ {n : ℕ} (u : Fin n → (T ⟶ G)), t ≫ (List.ofFn u).prod = (List.ofFn fun i => t ≫ u i).prod | 0, u => by simp | n + 1, u => by rw [List.ofFn_succ, List.ofFn_succ, List.prod_cons, List.prod_cons, MonObj.comp_mul, comp_list_prod_ofFn t] theorem isTangent_comp_of_isMonHom {R : Type} [CommRing R] {B G : Over (Sp R)} [GrpObj B] [GrpObj G] (Φ : B ⟶ G) [IsMonHom Φ] (v : epsR R ⟶ B) (hv : IsTangent v) : IsTangent (v ≫ Φ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsAbelianSchemeOfGenericallySurjective

theorem **surjective_of_generic_fibre_subset_range** {V K : Type} [CommRing V] [IsDomain V] [Field K] [Algebra V K] [IsFractionRing V K] {X Y : Scheme.{0}} (f : X ⟶ Y) (g : Y ⟶ Sp V) [UniversallyClosed f] [Flat g] (h : ∀ y : Y, g y ∈ Set.range (spMap (algebraMap V K)) → y ∈ Set.range f) : Surjective f

theorem **isAbelianScheme_of_generically_surjective** : ∀ (V K : Type) [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] (B G : Over (Sp V)) [GrpObj B] [GrpObj G], IsAbelianScheme B → IsSemiAbelian G → ∀ ψ : B ⟶ G, (∀ x : G.left, G.hom x ∈ Set.range (spMap (algebraMap V K)) → x ∈ Set.range ψ.left) → IsAbelianScheme G

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsAbelianSchemeSchemeOver

theorem **GlobalTheory.WeierstrassCurve.isAbelianScheme_schemeOver** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), IsUnit W.Δ → ∀ [GrpObj (schemeOver W)], IsCommMonObj (schemeOver W) → IsAbelianScheme (schemeOver W)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsCMPairingFamilyOfPolarisationDiagonalComplex

private theorem **strictAnti_lt_of_succ_lt** (D : Datum H) (u v : D.La) : D.EF u v = -D.EF v u

private theorem **ef_w0_smul_left_swap** (D : Datum H) (u y : D.La) : D.EF (H.w0 • y) u = D.EF (H.w0 • u) y - D.EF u y

private theorem **ef_rt_smul_right** (D : Datum H) (u v : D.La) : D.EF u (H.rt • v) = 2 * D.EF u (H.w0 • v) - D.EF u v

private theorem **four_mul_ef_w0_smul_w0_smul** (D : Datum H) (u v : D.La) : 4 * D.EF (H.w0 • u) (H.w0 • v) = (1 + H.d) * D.EF u v

theorem **polarise** (D : Datum H) (L : Submodule (𝒪 E) D.La) {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (ω c : ℂ) (hω : (2 * ω - 1) ^ 2 = -(H.d : ℂ)) (x : L → (Fin g → ℂ)) (hadd : ∀ a b, x (a + b) = x a + x b) (hrt : ∀ a, x (H.rt • a) = (2 * ω - 1) • x a) (hq : ∀ a : L, hermForm Hm (x a) (x a) = (D.EF (H.w0 • (a : D.La)) a : ℂ) * c) (a b : L) : hermForm Hm (x a) (x b) = (((D.EF (H.w0 • (a : D.La)) b : ℂ) - D.EF a b) + (D.EF a b : ℂ) * ω) * c

theorem **smul_of_rt** {M : Type} [AddCommGroup M] [Module (𝒪 E) M] {g : ℕ} (ω : ℂ) (τ : OK H →+* ℂ) (hτ : τ (wK H) = ω) (x : M → (Fin g → ℂ)) (hadd : ∀ a b, x (a + b) = x a + x b) (hrt : ∀ a, x (H.rt • a) = (2 * ω - 1) • x a) (c : OK H) (a : M) : x ((c : 𝒪 E) • a) = τ c • x a

theorem **char_sq** (ω : ℂ) (τ : OK H →+* ℂ) (hτ : τ (wK H) = ω) : (2 * ω - 1) ^ 2 = -(H.d : ℂ)

theorem **isFam_of_isHomFamily** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (ω : ℂ) (C : Ell H ℂ ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (T : PAT H (Sp ℂ)) (L : Submodule (𝒪 E) D.La) (ψ : L → (C.P.A ⟶ T.t.G)), IsKForm D ε h → (∀ x ∈ L, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → PureType.IsHomFamily T.t C.P.A C.P.L C.v L (2 * ω - 1) Set.univ ψ → IsFam C T h L ψ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsCMPairingFamilyOfPolarisationDiagonalComplexStatement

theorem **_root_.GlobalTheory.T.isCMPairingFamily_of_polarisationDiagonal_complex** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (ω : ℂ) (C : Ell H ℂ ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (T : PAT H (Sp ℂ)) (L : Submodule (𝒪 E) D.La) (ψ : L → (C.P.A ⟶ T.t.G)), IsKForm D ε h → (∀ x ∈ L, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → PureType.IsHomFamily T.t C.P.A C.P.L C.v L (2 * ω - 1) Set.univ ψ → IsFam C T h L ψ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsCubeLBPairingIsoBaseChange

theorem **lamBC_psum_hom** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {T : Over S} {T' : Over S'} (ρ : (T' ⟶ G') →* (T ⟶ G)) {n : ℕ} (x : Fin n → (T' ⟶ G')) (I : Finset (Fin n)) : ρ (psum G' x I) = psum G (fun i => ρ (x i)) I

theorem **lamBC_psum_push** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {f : S' ⟶ S} {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) {T' : Over S'} {T : Over S} {n : ℕ} (x' : Fin n → (T' ⟶ G')) (x : Fin n → (T ⟶ G)) (U : Over.mk (T'.hom ≫ f) ⟶ T) (hx : ∀ i, pushPt h.1.w (x' i) = U ≫ x i) (I : Finset (Fin n)) : (psum G' x' I).left ≫ φ = U.left ≫ (psum G x I).left

theorem **lamEq_baseChange** : ∀ (S S' : Scheme.{0}) (s : S' ⟶ S) (X Y A B : Over S) (X' Y' A' B' : Over S') [GrpObj A] [GrpObj B] [GrpObj A'] [GrpObj B'] (x : X'.left ⟶ X.left) (y : Y'.left ⟶ Y.left) (a : A'.left ⟶ A.left) (b : B'.left ⟶ B.left), x ≫ X.hom = X'.hom ≫ s → y ≫ Y.hom = Y'.hom ≫ s → IsGrpPullback A' A s a → IsGrpPullback B' B s b → ∀ (L : LB A.left) (M : LB B.left) (L' : LB A'.left) (M' : LB B'.left), L.IsLB → M.IsLB → L'.IsLB → M'.IsLB → LBIso (L.pull a) L' → LBIso (M.pull b) M' → ∀ (f₁ : X ⟶ A) (f₂ : Y ⟶ A) (g₁ : X ⟶ B) (g₂ : Y ⟶ B) (f₁' : X' ⟶ A') (f₂' : Y' ⟶ A') (g₁' : X' ⟶ B') (g₂' : Y' ⟶ B'), f₁'.left ≫ a = x ≫ f₁.left → f₂'.left ≫ a = y ≫ f₂.left → g₁'.left ≫ b = x ≫ g₁.left → g₂'.left ≫ b = y ≫ g₂.left → LamEq L f₁ f₂ M g₁ g₂ → LamEq L' f₁' f₂' M' g₁' g₂'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsCubeLBPairingIsoBaseChangeStatement

theorem **_root_.GlobalTheory.T.isCubeLBPairingIso_baseChange** : ∀ (S S' : Scheme.{0}) (s : S' ⟶ S) (X Y A B : Over S) (X' Y' A' B' : Over S') [GrpObj A] [GrpObj B] [GrpObj A'] [GrpObj B'] (x : X'.left ⟶ X.left) (y : Y'.left ⟶ Y.left) (a : A'.left ⟶ A.left) (b : B'.left ⟶ B.left), x ≫ X.hom = X'.hom ≫ s → y ≫ Y.hom = Y'.hom ≫ s → IsGrpPullback A' A s a → IsGrpPullback B' B s b → ∀ (L : LB A.left) (M : LB B.left) (L' : LB A'.left) (M' : LB B'.left), L.IsLB → M.IsLB → L'.IsLB → M'.IsLB → LBIso (L.pull a) L' → LBIso (M.pull b) M' → ∀ (f₁ : X ⟶ A) (f₂ : Y ⟶ A) (g₁ : X ⟶ B) (g₂ : Y ⟶ B) (f₁' : X' ⟶ A') (f₂' : Y' ⟶ A') (g₁' : X' ⟶ B') (g₂' : Y' ⟶ B'), f₁'.left ≫ a = x

» f₁.left ≫ f₂'.left » a = y » f₂.left ≫ g₁'.left » b = x » g₁.left ≫ g₂'.left » b = y » g₂.left ≫ LamEq L f₁ f₂ M g₁ g₂ ≫ LamEq L' f₁' f₂' M' g₁' g₂'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsEigenvectorAtOfModel

theorem **isEigAt_of_model** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K] (x : Sp K ⟶ Xo.X) (t₀ : SAB H (Sp K)) (B : SAB H (Sp (𝒪 K))) (m : SABMap t₀ Xo.univ.t x) (m' : SABMap t₀ B (spMap (algebraMap (𝒪 K) K))) (τ : E →+* ℂ) (w : epsR K ⟶ t₀.G), (∀ a : 𝒪 E, ∃ w' : epsR K ⟶ t₀.G, IsLieComb ![LamF.labK c K τ (a : E)] ![w] w' ∧ w.left ≫ m'.φ ≫ (B.act.act a).left = w'.left ≫ m'.φ) → IsEigAt c τ (w.left ≫ m.φ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsEigenvectorAtOfModelStatement

theorem **_root_.GlobalTheory.T.isEigenvectorAt_of_model** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (K : Type) [Field K] [NumberField K] [Algebra c.F K] (x : Sp K ⟶ Xo.X) (t₀ : SAB H (Sp K)) (B : SAB H (Sp (𝒪 K))) (m : SABMap t₀ Xo.univ.t x) (m' : SABMap t₀ B (spMap (algebraMap (𝒪 K) K))) (τ : E →+* ℂ) (w : epsR K ⟶ t₀.G), (∀ a : 𝒪 E, ∃ w' : epsR K ⟶ t₀.G, IsLieComb ![LamF.labK c K τ (a : E)] ![w] w' ∧ w.left ≫ m'.φ ≫ (B.act.act a).left = w'.left ≫ m'.φ) → IsEigAt c τ (w.left ≫ m.φ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsFibreOfIffIsPolFibre

theorem **isFibreOf_iff_isPolFibre** : ∀ (R R' : Type) [CommRing R] [CommRing R'] [Nontrivial R'] (f : R →+* R') (A : Out.PolAV (Sp R)) (x : KeyIn R') (φ : x.P.A.left ⟶ A.A.left), IsFibreOf A x (spMap f) φ ↔ IsPolFibre f A x.P φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsIntegralDerivativeCurveHomGoodReduction

theorem **isLieComb_single_iff** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] (c : R) (v w : epsR R ⟶ G) : IsLieComb ![c] ![v] w ↔ w.left = spMap (epsScaleR c) ≫ v.left

theorem **isLieComb_fibre** {R R' : Type} [CommRing R] [CommRing R'] {g : R →+* R'} {G H : Over (Sp R)} [GrpObj H] {G' H' : Over (Sp R')} [GrpObj H'] {φ : G'.left ⟶ G.left} {ψ : H'.left ⟶ H.left} (hψ : IsPullback ψ H'.hom H.hom (spMap g)) {F : G ⟶ H} {f : G' ⟶ H'} (hf : f.left ≫ ψ = φ ≫ F.left) {v : epsR R ⟶ G} {t : epsR R ⟶ H} {v' : epsR R' ⟶ G'} {t' : epsR R' ⟶ H'} (hv : IsVecOver g φ v v') (ht : IsVecOver g ψ t t') {b : R} (h : IsLieComb ![b] ![t] (v ≫ F)) : IsLieComb ![g b] ![t'] (v' ≫ f)

theorem **exists_extension** {S S' : Scheme.{0}} {j : S' ⟶ S} {G H : Over S} [GrpObj G] [GrpObj H] (h : ∀ f : (Over.pullback j).obj G ⟶ (Over.pullback j).obj H, IsMonHom f → ∃! F : G ⟶ H, IsMonHom F ∧ (Over.pullback j).map F = f) {G' H' : Over S'} [GrpObj G'] [GrpObj H'] {φ : G'.left ⟶ G.left} {ψ : H'.left ⟶ H.left} (hφ : IsGrpPullback G' G j φ) (hψ : IsGrpPullback H' H j ψ) (f : G' ⟶ H') [IsMonHom f] : ∃ F : G ⟶ H, IsMonHom F ∧ f.left ≫ ψ = φ ≫ F.left

theorem **isIntegral_of_hom_derivative** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → Out.PolBaseChangeSt → ∀ (M : Type) [Field M] [NumberField M] (P Q : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A) (t : epsR (𝒪 M) ⟶ Q.A) (a : M), IsTangent v → (∀ (R' : Type) [CommRing R'] (f' : (𝒪 M) →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ Q.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' Q P' φ' → IsVecOver f' φ' t v' → IsLieBasis ![v']) → (∀ (P₁ Q₁ : Out.PolAV (Sp M)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (v₁ : epsR M ⟶ P₁.A) (t₁ : e

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsIntegralDerivativeCurveHomGoodReductionStatement

theorem **_root_.GlobalTheory.T.isIntegral_derivative_curve_hom_good_reduction** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → Out.PolBaseChangeSt → ∀ (M : Type) [Field M] [NumberField M] (P Q : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A) (t : epsR (𝒪 M) ⟶ Q.A) (a : M), IsTangent v → (∀ (R' : Type) [CommRing R'] (f' : (𝒪 M) →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ Q.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' Q P' φ' → IsVecOver f' φ' t v' → IsLieBasis ![v']) → (∀ (P₁ Q₁ : Out.PolAV (Sp M)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsIntegralLevelMulKappa

theorem **herm_smul_mem** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : DataX E r) (S : Set E) (hΛ : X.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ S, ∃ u ∈ X.L', v = a • u}) {v : Fin r → E} (hv : v ∈ X.L') {x : Fin r → E} (hx : x ∈ X.Λ) : FD.herm X.Gm x v • v ∈ X.Λ

theorem **transv_mem_of_span** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : DataX E r) (S : Set E) (hΛ : X.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ S, ∃ u ∈ X.L', v = a • u}) (m : ℕ) (hmκ : IsIntegral ℤ ((m : E) * X.κ0)) (v : Fin r → E) (hv : v ∈ X.L') (hiso : FD.herm X.Gm v v = 0) (b : E) (hb : cc b = b) (hbi : IsIntegral ℤ b) : ∃ γ ∈ X.Γ, ∀ τ : X.Ψ, (γ : Matrix (Fin r) (Fin r) E).map τ.1 = X.flowMat v τ ((((X.g * m : ℕ) : ℝ) : ℂ) * τ.1 b)

theorem **xiR_eq** (D : D7) : xiR D = 1 / ((Ideal.absNorm D.host.ccK : ℝ) * Real.sqrt D.host.d)

theorem **isIntegral_level_mul_kappa** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : FD.Data E r) {N d : ℕ} (hN : 0 < N) (hd : 0 < d) (τ : E →+* ℂ) (hτ : ‖τ X.ξ0‖ = 1 / ((N : ℝ) * Real.sqrt d)) : IsIntegral ℤ (((N * d : ℕ) : E) * X.κ0)

theorem **transv_model** : ∀ (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ), InR D j → ∀ (k : (T.U j T.σ₀).Comp) (Xx : DataX D.host.E D.R.r), Xx.toData = ((T.U j T.σ₀).X k) → Xx.g = T.Q.l.M → Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u} → ((D.host.g 0 * D.host.g 1 * Ideal.absNorm D.host.ccK * D.host.d : ℕ) : ℝ) * Xx.k0 = 2 * (D.host.g 0 * D.host.g 1 : ℕ) * Real.sqrt D.host.d ∧ ∀ v ∈ Xx.L', FD.herm ((T.U j T.σ₀).X k).Gm v v = 0 → ∀ b : D.host.E, cc b = b → IsIntegral ℤ b → ∃ γ ∈ ((T.U j T.σ₀).X k).Γ, ∀ τ : ((T.U j T.σ₀).X k).Ψ, (γ : Matrix (Fin D.R.r) (Fin D.R.r) D.host.E).map τ.1 = ((T.U j T.σ₀).X k).flowMat v τ ((((D.host.g 0 * D.host.g 1 * Ideal.absNorm D.host.ccK * D.host.d : ℕ) : ℝ) : ℂ) * τ.1 b)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsIsoAppLEOfOpensRange

theorem **isIso_appLE_of_opensRange** {X Y : Scheme.{0}} (f : X ⟶ Y) [IsOpenImmersion f] (U : Y.Opens) (hU : f.opensRange = U) (h : ⊤ ≤ f ⁻¹ᵁ U) : IsIso (f.appLE U ⊤ h)

theorem **appLE_SpecMap_comp** {Y : Scheme.{0}} {A B : CommRingCat.{0}} (f : Spec B ⟶ Y) (g : B ⟶ A) (U : Y.Opens) (h : ⊤ ≤ f ⁻¹ᵁ U) (p : Spec A ⟶ Y) (hp : p = Spec.map g ≫ f) (h' : ⊤ ≤ p ⁻¹ᵁ U) : p.appLE U ⊤ h' ≫ (Scheme.ΓSpecIso A).hom = f.appLE U ⊤ h ≫ (Scheme.ΓSpecIso B).hom ≫ g

theorem **appLE_structure** {Y : Scheme.{0}} {A B : CommRingCat.{0}} (f : Spec B ⟶ Y) (g : A ⟶ B) (U : Y.Opens) (h : ⊤ ≤ f ⁻¹ᵁ U) (t : Y ⟶ Spec A) (ht : f ≫ t = Spec.map g) : (Scheme.ΓSpecIso A).inv ≫ t.appLE ⊤ U le_top ≫ f.appLE U ⊤ h ≫ (Scheme.ΓSpecIso B).hom = g

theorem **top_le_preimage** {R : Type} [CommRing R] (W : WeierstrassCurve R) (i : Fin 3) : ⊤ ≤ awayChart W i ⁻¹ᵁ coordOpen W i

theorem **surjective_eval₂Hom_awayCoord** {R : Type} [CommRing R] (W : WeierstrassCurve R) (i : Fin 3) : Function.Surjective (MvPolynomial.eval₂Hom (awayConst W i) (awayCoord W i))

theorem **exists_coordFn** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R), ∃ c : ∀ i j : Fin 3, Γ(scheme W, coordOpen W i), (∀ (i j : Fin 3) (A : Type) [CommRing A] (ρ : R →+* A) (P : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) (h : ⊤ ≤ ptHom W ρ P hP ⁻¹ᵁ coordOpen W i), (Scheme.ΓSpecIso (CommRingCat.of A)).hom.hom (((ptHom W ρ P hP).appLE (coordOpen W i) ⊤ h).hom (c i j)) * P i = P j) ∧ ∀ i, Function.Surjective (MvPolynomial.eval₂Hom (((toSpec W).appLE ⊤ (coordOpen W i) le_top).hom.comp (Scheme.ΓSpecIso (CommRingCat.of R)).inv.hom) (c i))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsIsoOfSurjectivePrincipalPairing

theorem **exists_gen_of_lbIso** {X : Scheme.{0}} {P Q : LB X} (hP : P.IsLB) (hQ : Q.IsLB) (h : LBIso P Q) (H : ∃ t : LB.Gen P, t.IsGen) : ∃ t : LB.Gen Q, t.IsGen

theorem **locGen_base** {X Y B : Scheme.{0}} {L : LB X} (hL : L.IsLB) (π : Y ⟶ B) (a : B ⟶ X) {b : Y ⟶ X} (e : b = π ≫ a) : LocGen L π (fun _ : PUnit => b) (fun _ => 1)

theorem **eq_one_of_comp_eq_one** {S : Scheme.{0}} {A B : Over S} [GrpObj A] [GrpObj B] [IsCommMonObj A] [IsCommMonObj B] {L : LB A.left} {M : LB B.left} {Φ : A ⟶ B} (hL : L.IsLB) (hM : M.IsLB) (hK : Out.KTriv A L) (h : LamEq M Φ Φ L (𝟙 A) (𝟙 A)) {T : Over S} (x : T ⟶ A) (hx : x ≫ Φ = 1) : x = 1

theorem **isIso_of_lamEq** : ∀ (S : Scheme.{0}) [IsReduced S] (A B : Over S) [GrpObj A] [GrpObj B] (L : LB A.left) (M : LB B.left) (Φ : A ⟶ B), IsAbelianScheme A → IsAbelianScheme B → L.IsLB → M.IsLB → IsMonHom Φ → Out.KTriv A L → LamEq M Φ Φ L (𝟙 A) (𝟙 A) → Surjective Φ.left → IsIso Φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsIsoOfSurjectivePrincipalPairingStatement

theorem **_root_.GlobalTheory.T.isIso_of_surjective_principalPairing** : ∀ (S : Scheme.{0}) [IsReduced S] (A B : Over S) [GrpObj A] [GrpObj B] (L : LB A.left) (M : LB B.left) (Φ : A ⟶ B), IsAbelianScheme A → IsAbelianScheme B → L.IsLB → M.IsLB → IsMonHom Φ → Out.KTriv A L → LamEq M Φ Φ L (𝟙 A) (𝟙 A) → Surjective Φ.left → IsIso Φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsLieCombOneIffComp

theorem **isLieComb_one_iff_comp** (a : R) (v x : epsR R ⟶ G) : IsLieComb ![a] ![v] x ↔ x = Chart.spMap (Chart.scaleD a) ≫ v

theorem **isLieComb_one_iff** (a : R) (v x : epsR R ⟶ G) : IsLieComb ![a] ![v] x ↔ x.left = spMap (epsScaleR a) ≫ v.left

theorem **scale_add** {v : epsR R ⟶ G} (hv : IsTangent v) (a b : R) : Chart.spMap (Chart.scaleD (a + b)) ≫ v = (Chart.spMap (Chart.scaleD a) ≫ v) * (Chart.spMap (Chart.scaleD b) ≫ v)

theorem **scaleD_comp** (a b : R) : (Chart.scaleD a).comp (Chart.scaleD b) = Chart.scaleD (a * b)

theorem **scaleD_one** : Chart.scaleD (1 : R) = AlgHom.id R _

theorem **lieMultiple_calculus** : ∀ (R : Type) [CommRing R] (G G' : Over (Sp R)) [GrpObj G] [GrpObj G'] (v x y : epsR R ⟶ G) (a b : R), (IsLieComb ![a] ![v] x ↔ x.left = spMap (epsScaleR a) ≫ v.left) ∧ (IsTangent v → IsLieComb ![a] ![v] x → IsLieComb ![b] ![v] y → IsLieComb ![a + b] ![v] (x * y)) ∧ (IsLieComb ![b] ![v] x → IsLieComb ![a] ![x] y → IsLieComb ![a * b] ![v] y) ∧ (∀ f : G ⟶ G', IsLieComb ![a] ![v] x → IsLieComb ![a] ![v ≫ f] (x ≫ f)) ∧ IsLieComb ![1] ![v] v ∧ (IsTangent v → IsLieComb ![0] ![v] 1)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsLieNormSqIffEqPiDivCovolume

theorem **isLieNormSq_iff_eq_pi_div_covolume** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (s : R →+* ℂ) (L : PeriodPair), P.g = 1 → IsTangent v → (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s P P' φ' → IsVecOver s φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ L.lattice ↔ z • w' ∈ u'.Λ) → 0 < Real.pi / |(L.ω₁ * starRingEnd ℂ L.ω₂).im| ∧ ∀ r, IsLieNormSq P v s r ↔ r = Real.pi / |(L.ω₁ * starRingEnd ℂ L.ω₂).im|

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsMetricConjugateFramedPoint

theorem **isLieComb_of_two_pullbacks** {R S : Type} [CommRing R] [CommRing S] (f : R →+* S) {X : Scheme.{0}} {x : Sp R ⟶ X} {x' : Sp S ⟶ X} (hx : x' = spMap f ≫ x) {G : Over X} [GrpObj G] {G₀ : Over (Sp R)} [GrpObj G₀] {G' : Over (Sp S)} [GrpObj G'] {φ : G₀.left ⟶ G.left} {φ' : G'.left ⟶ G.left} (h : IsGrpPullback G₀ G x φ) (h' : IsGrpPullback G' G x' φ') {n n' : ℕ} (hn : n' = n) {c : Fin n → R} {v : Fin n → (epsR R ⟶ G₀)} {w : epsR R ⟶ G₀} (hc : IsLieComb c v w) {v' : Fin n' → (epsR S ⟶ G')} {w' : epsR S ⟶ G'} (hv : ∀ i, (v' i).left ≫ φ' = spMap (Formalism.L2.Pj.dualMap f) ≫ (v (i.cast hn)).left ≫ φ) (hw : w'.left ≫ φ' = spMap (Formalism.L2.Pj.dualMap f) ≫ w.left ≫ φ) : IsLieComb (fun i => f (c (i.cast hn))) v' w'

theorem **hermForm_map_conj** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z w : Fin g → ℂ) : hermForm (Hm.map (starRingEnd ℂ)) (fun i => starRingEnd ℂ (z i)) (fun i => starRingEnd ℂ (w i)) = starRingEnd ℂ (hermForm Hm z w)

theorem **isMetricAt_of_dim** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} (x : Sp ℂ ⟶ Xo.X) (s : LB.Sec (PPx x)) (uu : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left)) (N : ℝ) (Gr : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ) (Φ : FibKey Xo ℂ x) (g : ℕ) (hg : Φ.key.P.g = g) (u : HolUnif Φ.key.P.A Φ.key.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)) (w : ∀ b : E →+* ℂ, Fin (Dw.mult b) → (epsR ℂ ⟶ Φ.key.P.A)) (W : ∀ b : E →+* ℂ, Fin (Dw.mult b) → Fin g → ℂ) (h : SecTie Φ.m Φ.th s ∧ u.IsAH Hm χ ∧ u.IsRig Φ.key.rig ∧ u.IsFn Φ.key.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume ∧ N = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal ∧ (∀ b i, (w b i).left ≫ Φ.m.φ = uu b i ∧ u.IsDiff (W b i) (w b i)) ∧ ∀ b

theorem **isMetricAt_conj** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} (hpq : 1 ≤ Dw.p + Dw.q) (x x' : Sp ℂ ⟶ Xo.X) (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')) (uu uu' : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left)) (N : ℝ) (Gr : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ) (hx : x' = spMap (starRingEnd ℂ) ≫ x) (hs : SecBC (starRingEnd ℂ) 1 x x' s s') (hgen : SecGen x' s') (ht : ∀ τ i, IsTanAt x (uu τ i)) (ht' : ∀ τ i, IsTanAt x' (uu' τ i)) (hu : ∀ τ i, uu' τ i = spMap (Formalism.L2.Pj.dualMap (starRingEnd ℂ)) ≫ uu τ i) (h : IsMetricAt x s uu N Gr) : IsMetricAt x' s' uu' N (fun b => (Gr b).map (starRingEnd ℂ))

theorem **scal_one_mul_sec** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {S : Type} [CommRing S] (x : Sp S ⟶ Xo.X) (s : LB.Sec (PPx x)) (i : (PPx x).ι) : scal x 1 i * s.f i = s.f i

theorem **isMetric_conj** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) (y' : FrPt Xo c (CAt ((starRingEnd ℂ).comp σ))) (N : ℝ) (Gr : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ)), 1 ≤ Dw.p + Dw.q → FrMap (FrameNorm.conjAlg σ) y y' → IsMetric y N Gr → IsMetric y' N (fun b => (Gr b).map (starRingEnd ℂ))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsMetricConjugateFramedPointStatement

theorem **_root_.GlobalTheory.T.isMetric_conjugate_framedPoint** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) (y' : FrPt Xo c (CAt ((starRingEnd ℂ).comp σ))) (N : ℝ) (Gr : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ)), 1 ≤ Dw.p + Dw.q → FrMap (FrameNorm.conjAlg σ) y y' → IsMetric y N Gr → IsMetric y' N (fun b => (Gr b).map (starRingEnd ℂ))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsModelAtIntegralIffMemLieLat

theorem **isModelAt_integral_iff_mem_lieLat** : ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (p : HeightOneSpectrum (𝒪 K)) (M : Matrix (Fin x.P.g) (Fin x.P.g) K), IsModelAt A x φ p M → ∀ c : Fin x.P.g → K, (∀ m, p.valuation K (Matrix.vecMul c M⁻¹ m) ≤ 1) ↔ ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • c ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).lieLat

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsPullCubicalOfIso

theorem **chart_of_val** (hM : M.IsLB) {u u' : LB.Gen (LB.LinComb M φ ε)} (hu : u.IsGen) (hu' : u'.IsGen) {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (α : (Z.presheaf.stalk z)ˣ) (h : ∀ J, Good M z J → val u r z J = α * val u' r z J) (q : K → M.ι) (hq : r z ∈ LB.wchart M φ q) : gm r z (LB.wchart M φ q) (u.s q) = α * gm r z (LB.wchart M φ q) (u'.s q)

theorem **exists_rigidified** (he : eSec G' ≫ F = f ≫ eSec G) (L : RigLB G) (L' : RigLB G') (h : LBIso L'.L (L.L.pull F)) : ∃ w : LB.Iso L'.L (L.L.pull F), PGen w ∧ PMaps w (K

theorem **val_origin** (gp : IsGrpPullback G' G f F) {L : RigLB G} {L' : RigLB G'} (c : CubicalStr G L) {w : LB.Iso L'.L (L.L.pull F)} {t : LB.Gen (LB.LinComb L'.L (thφ G' (x3 G')) thε)} (H : PMaps w (thφ G' (x3 G')) (thφ G (x3 G)) thε t c.t (cubeMapOf gp).F3) (T : Over S') (y : T.left) (J') (hJ' : Good L'.L y J') : val t (lift (1 : T ⟶ G') (lift 1 1)).left y J' = 1

theorem **pmaps_cube** (hG : Out.GlobalFnsSt) (hA : IsAbelianScheme G') (gp : IsGrpPullback G' G f F) {L : RigLB G} {L' : RigLB G'} (c : CubicalStr G L) (c' : CubicalStr G' L') {w : LB.Iso L'.L (L.L.pull F)} (hw : PGen w) : PMaps w (thφ G' (x3 G')) (thφ G (x3 G)) thε c'.t c.t (cubeMapOf gp).F3

theorem **isPullCubical_of_iso** (hG : Out.GlobalFnsSt) (hA : IsAbelianScheme G') (gp : IsGrpPullback G' G f F) {L : RigLB G} {L' : RigLB G'} (c : CubicalStr G L) (c' : CubicalStr G' L') (h : LBIso L'.L (L.L.pull F)) : (cubeMapOf gp).IsPullCubical c' c

theorem **exists_tripleMap** : Out.GlobalFnsSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S S' : Scheme.{0}) (f : S' ⟶ S) (t' : SAB H S') (t : SAB H S) (φ : t'.G.left ⟶ t.G.left), IsAbelianScheme t'.G → IsGrpPullback t'.G t.G f φ → (∀ a, (t'.act.act a).left ≫ φ = φ ≫ (t.act.act a).left) → LBIso t'.PP.L (t.PP.L.pull φ) → ∃ m : SABMap t' t f, m.φ = φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsTenBaseChange

theorem **isTen_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (t : IsTen C D h T) (R' : Type) [CommRing R'] (ω' : R') (C' : Ell H R' ω') (f : R →+* R') (φ : C'.P.A.left ⟶ C.P.A.left) (T' : PAT H (Sp R')) (m : SABMap T'.t T.t (spMap f)), EllMap C' C f φ → ∃ t' : IsTen C' D h T', ∀ a, (t'.ten.tmul a).left ≫ m.φ = φ ≫ (t.ten.tmul a).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsUnimodularSolZeroNeg

theorem **ptHom_origin_inj** {ρ ρ' : R →+* A} {z w z' w' : A} (h : IsUnimodularSol (W.map ρ) ![z, -1, w]) (h' : IsUnimodularSol (W.map ρ') ![z', -1, w']) (e : ptHom W ρ ![z, -1, w] h = ptHom W ρ' ![z', -1, w'] h') : z = z' ∧ w = w'

theorem **exists_eq_ptHom_origin** (ρ : R →+* A) (g : Spec (.of A) ⟶ scheme W) (hY : ∀ p, g p ∈ coordOpen W 1) (hg : g ≫ toSpec W = Spec.map (CommRingCat.ofHom ρ)) : ∃ (z w : A) (h : IsUnimodularSol (W.map ρ) ![z, -1, w]), g = ptHom W ρ ![z, -1, w] h

theorem **SpecMap_zero_origin** (ρ : R →+* A) (h : IsUnimodularSol (W.map ρ) ![0, -1, 0]) : Spec.map (CommRingCat.ofHom ρ) ≫ zero W = ptHom W ρ ![0, -1, 0] h

theorem **isUnimodularSol_zero_neg** (V : WeierstrassCurve A) : IsUnimodularSol V ![0, -1, 0]

theorem **exists_eq_ptHom_eps** (f : R →+* A) (g : Spec (.of (DualNumber A)) ⟶ scheme W) (hg : g ≫ toSpec W = Spec.map (CommRingCat.ofHom ((algebraMap A (DualNumber A)).comp f))) (h0 : Spec.map (CommRingCat.ofHom (TrivSqZeroExt.fstHom A A A).toRingHom) ≫ g = Spec.map (CommRingCat.ofHom f) ≫ zero W) : ∃ (c : A) (h : IsUnimodularSol (W.map ((algebraMap A (DualNumber A)).comp f)) ![TrivSqZeroExt.inr c, -1, 0]), g = ptHom W ((algebraMap A (DualNumber A)).comp f) ![TrivSqZeroExt.inr c, -1, 0] h

theorem **epsScale_ptHom** (f : R →+* A) (c : A) (h : IsUnimodularSol (W.map ((algebraMap A (DualNumber A)).comp f)) ![DualNumber.eps, -1, 0]) : ∃ h', spMap (epsScaleR c) ≫ ptHom W ((algebraMap A (DualNumber A)).comp f) ![DualNumber.eps, -1, 0] h = ptHom W ((algebraMap A (DualNumber A)).comp f) ![TrivSqZeroExt.inr c, -1, 0] h'

theorem **isLieComb_one_iff** {G : Over (Sp A)} [GrpObj G] (c : Fin 1 → A) (v w : epsR A ⟶ G) : IsLieComb c ![v] w ↔ w.left = spMap (epsScaleR (c 0)) ≫ v.left

theorem **isLieBasis_eps** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R) (R' : Type) [CommRing R'] (f : R →+* R') (A : Over (Sp R')) [GrpObj A] (φ : A.left ⟶ scheme W) (v : epsR R' ⟶ A) (h : IsUnimodularSol (W.map ((algebraMap R' (DualNumber R')).comp f)) ![DualNumber.eps, -1, 0]), IsPullback φ A.hom (toSpec W) (spMap f) → eSec A ≫ φ = spMap f ≫ zero W → v.left ≫ φ = ptHom W ((algebraMap R' (DualNumber R')).comp f) ![DualNumber.eps, -1, 0] h → IsLieBasis ![v]

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsoLineBundleOfRegularPullbacksSameZeroScheme

theorem **resG_appLE** {X Y : Scheme.{0}} (f : Y ⟶ X) {U : X.Opens} {V W : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ V) (a : Γ(X, U)) : resG Y h' ((f.appLE U V h).hom a) = (f.appLE U W (h'.trans h)).hom a

theorem **appLE_resG** {X Y : Scheme.{0}} (f : Y ⟶ X) {U U' : X.Opens} {V : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : U ≤ U') (h'' : V ≤ f ⁻¹ᵁ U') (a : Γ(X, U')) : (f.appLE U V h).hom (resG X h' a) = (f.appLE U' V h'').hom a

theorem **appLE_appLE** {X Y T : Scheme.{0}} (t : T ⟶ Y) (f : Y ⟶ X) {U : X.Opens} {V : Y.Opens} (h : V ≤ f ⁻¹ᵁ U) (h' : ⊤ ≤ t ⁻¹ᵁ V) (h'' : ⊤ ≤ (t ≫ f) ⁻¹ᵁ U) (a : Γ(X, U)) : (t.appLE V ⊤ h').hom ((f.appLE U V h).hom a) = ((t ≫ f).appLE U ⊤ h'').hom a

theorem **appLE_compat** {X Y : Scheme.{0}} {L : LB X} (θ : L.Sec) (f : Y ⟶ X) (i j : L.ι) {W : Y.Opens} (hi : W ≤ f ⁻¹ᵁ L.U i) (hj : W ≤ f ⁻¹ᵁ L.U j) (hij : W ≤ f ⁻¹ᵁ (L.U i ⊓ L.U j)) : (f.appLE (L.U i) W hi).hom (θ.f i) = (res hij (pullU f (L.U i ⊓ L.U j)) (L.g i j)) : Γ(Y,

W)ˣ) ∗ (f.appLE (L.U j) W hj).hom (θ.f j)

theorem **unit_eq_of_mul** {R : Type*} [CommRing R] {a a' b b' : R} {c c' g₀ g₁ : Rˣ} (ha : a ∈ nonZeroDivisors R) (h : b = c ∗ a) (h' : b' = c' ∗ a') (hb : b = g₀ ∗ b') (haa : a = g₁ ∗ a') : c = g₀ ^ (1 : ℤ) ∗ g₁ ^ (-1 : ℤ) ∗ c'

theorem **exists_isGen_linComb** : ∀ (X Y : Scheme.{0}) (L : LB X) (θ : L.Sec) (φ ψ : Y ⟶ X), L.IsLB → (∀ (i : L.ι) (V : Y.Opens) (h : V ≤ φ ⁻¹ᵁ L.U i), (φ.appLE (L.U i) V h).hom (θ.f i) ∈ nonZeroDivisors Γ(Y, V)) → (∀ (i : L.ι) (V : Y.Opens) (h : V ≤ ψ ⁻¹ᵁ L.U i), (ψ.appLE (L.U i) V h).hom (θ.f i) ∈ nonZeroDivisors Γ(Y, V)) → (∀ (T : Scheme.{0}) (t : T ⟶ Y) (i j : L.ι) (h : ⊤ ≤ (t ≫ φ) ⁻¹ᵁ L.U i) (h' : ⊤ ≤ (t ≫ ψ) ⁻¹ᵁ L.U j), ((t ≫ φ).appLE (L.U i) ⊤ h).hom (θ.f i) = 0 ↔ ((t ≫ ψ).appLE (L.U j) ⊤ h').hom (θ.f j) = 0) → ∃ t : LB.Gen (LB.LinComb L (K

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsoLineBundleOfRegularPullbacksSameZeroSchemeStatement

theorem **_root_.GlobalTheory.T.isoLineBundle_of_regularPullbacks_sameZeroScheme** : ∀ (X Y : Scheme.{0}) (L : LB X) (θ : L.Sec) (φ ψ : Y ⟶ X), L.IsLB → (∀ (i : L.ι) (V : Y.Opens) (h : V ≤ φ ⁻¹ᵁ L.U i), (φ.appLE (L.U i) V h).hom (θ.f i) ∈ nonZeroDivisors Γ(Y, V)) → (∀ (i : L.ι) (V : Y.Opens) (h : V ≤ ψ ⁻¹ᵁ L.U i), (ψ.appLE (L.U i) V h).hom (θ.f i) ∈ nonZeroDivisors Γ(Y, V)) → (∀ (T : Scheme.{0}) (t : T ⟶ Y) (i j : L.ι) (h : ⊤ ≤ (t ≫ φ) ⁻¹ᵁ L.U i) (h' : ⊤ ≤ (t ≫ ψ) ⁻¹ᵁ L.U j), ((t ≫ φ).appLE (L.U i) ⊤ h).hom (θ.f i) = 0 ↔ ((t ≫ ψ).appLE (L.U j) ⊤ h').hom (θ.f j) = 0) → ∃ t : LB.Gen (LB.LinComb L (K

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsogenyOfSurjectivePolAV

theorem **eq_bot_of_comap_eq_bot** {K R S : Type} [Field K] [CommRing R] [CommRing S] [IsDomain R] [IsDomain S] [Algebra K R] [Algebra K S] (φ : R →+* S) (hφ : φ.comp (algebraMap K R) = algebraMap K S) {g : ℕ} (hR : Algebra.trdeg K R = g) (hS : Algebra.trdeg K S = g) {p : Ideal S} (hp : p.comap φ = ⊥) : p = ⊥

theorem **eq_genericPoint_iff** {X : Scheme.{0}} [IsIntegral X] {R : CommRingCat.{0}} [IsDomain R] (j : Spec R ⟶ X) [IsOpenImmersion j] (p : Spec R) : j p = genericPoint X ↔ p.asIdeal = ⊥

theorem **eq_genericPoint_of_apply_eq** {K : Type} [Field K] {X Y : Scheme.{0}} [IsIntegral X] [IsIntegral Y] (f : X ⟶ Y) (s : Y ⟶ Spec (CommRingCat.of K)) {g : ℕ} (hY : SmoothOfRelativeDimension g s) (hX : SmoothOfRelativeDimension g (f ≫ s)) {x : X} (hx : f x = genericPoint Y) : x = genericPoint X

theorem **isPullback_kernel_mul** {C : Type*} [Category C] [CartesianMonoidalCategory C] {Z A B : C} [GrpObj A] [GrpObj B] {i : Z ⟶ A} {ψ : A ⟶ B} [IsMonHom ψ] (hp : IsPullback i (toUnit Z) ψ η[B]) : IsPullback ((fst Z A ≫ i) ∗ snd Z A) (snd Z A) ψ ψ

theorem **locallyQuasiFinite_of_finite_fibre** {S : Scheme.{0}} [Subsingleton S] {A B : Over S} [GrpObj A] [GrpObj B] (ψ : A ⟶ B) [IsMonHom ψ] [Surjective ψ.left] [LocallyOfFiniteType ψ.left] [QuasiCompact ψ.left] (a : A.left) (h : (ψ.left ⁻¹' {ψ.left a}).Finite) : LocallyQuasiFinite ψ.left

theorem **isogeny_of_surjective_of_dim_eq** : Out.Laws → ∀ (K : Type) [Field K] (P : Out.PolAV (Sp K)) (B : Over (Sp K)) [GrpObj B] (ψ : P.A ⟶ B), IsAbelianScheme B → SmoothOfRelativeDimension P.g B.hom → IsMonHom ψ → Surjective ψ.left → IsFinite ψ.left ∧ Flat ψ.left ∧ ∃ n : ℕ, 1 ≤ n ∧ ∀ (T : Over (Sp K)) (x : T ⟶ P.A), x ≫ ψ = 1 → x ^ n = 1

theorem **isogeny_of_surjective_polAV** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] (σ : K →+* ℂ) (P P' : Out.PolAV (Sp K)) (ψ : P.A ⟶ P'.A), P.g = P'.g → IsMonHom ψ → Surjective ψ.left → IsFinite ψ.left ∧ Flat ψ.left ∧ ∃ n : ℕ, 1 ≤ n ∧ ∀ (T : Over (Sp K)) (x : T ⟶ P.A), x ≫ ψ = 1 → x ^ n = 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.IsogenyOfSurjectivePolAVStatement

theorem **_root_.GlobalTheory.T.finite_flat_torsion_ker_of_surjective_polAV** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] (σ : K →+* ℂ) (P P' : Out.PolAV (Sp K)) (ψ : P.A ⟶ P'.A), P.g = P'.g → IsMonHom ψ → Surjective ψ.left → IsFinite ψ.left ∧ Flat ψ.left ∧ ∃ n : ℕ, 1 ≤ n ∧ ∀ (T : Over (Sp K)) (x : T ⟶ P.A), x ≫ ψ = 1 → x ^ n = 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KTrivOfLamEq

theorem **kTriv_of_lamEq** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L L' : LB A.left), L.IsLB → L'.IsLB → LamEq L' (𝟙 A) (𝟙 A) L (𝟙 A) (𝟙 A) → Out.KTriv A L → Out.KTriv A L'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KTrivOfPerfect

theorem **kTriv_of_perfect** : ∀ (S : Scheme.{0}) (O : Type) [CommRing O] (G A : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (M : Type) [AddCommGroup M] [Module O M] (t : Serre.IsTensor A b G a M) (L : LB G.left) (N : LB A.left) (h : M →ₗ[O] M →+ O) (n : ℕ) (x y : Fin n → M), (∀ m, m = ∑ i, h m (y i) • x i) → L.IsLB → N.IsLB → (∃ t : LB.Gen (Theta G L), t.IsGen) → (∃ t : LB.Gen (Theta A N), t.IsGen) → (∀ m m', LamEq N (t.tmul m) (t.tmul m') L (a.act (h m m')) (𝟙 G)) → Out.KTriv G L → Out.KTriv A N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KTrivOfUnimodular

theorem **LocZero.add** {π : Y ⟶ B} {c d : Pic Y} (hc : LocZero π c) (hd : LocZero π d) : LocZero π (c + d)

theorem **LocZero.pull** {π : Y ⟶ B} {c : Pic Y} (hc : LocZero π c) (k : Y' ⟶ Y) : LocZero (k ≫ π) (pull k c)

theorem **locZero_base** (π : Y ⟶ B) (d : Pic B) : LocZero π (pull π d)

theorem **locZero_iff** (π : Y ⟶ B) {M : LB Y} (hM : M.IsLB) : (∀ p : B, ∃ V : B.Opens, p ∈ V ∧ ∃ c : LB.Gen (M.pull (π ⁻¹ᵁ V).ι), c.IsGen) ↔ LocZero π (cls M hM)

theorem **translDiff_iff** {L : LB A.left} (hL : L.IsLB) (x : T ⟶ A) : (∀ p : T.left, ∃ V : T.left.Opens, p ∈ V ∧ ∃ c : LB.Gen ((Out.translDiff A L x).pull ((fst T A).left ⁻¹ᵁ V).ι), c.IsGen) ↔ LocZero (fst T A).left (lam (cls L hL) (fst T A ≫ x) (snd T A))

theorem **lam_mul_of_right** [IsCommMonObj A] (c : Pic A.left) (u v y : T ⟶ A) (H : lam c u (v ∗ y) = lam c u v + lam c u y) : lam c (u ∗ v) y = lam c u y + lam c v y

theorem **kTriv_of_unimodular** : ∀ (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G] (L : LB G.left) (O : Type) [CommRing O] (bar : O →+* O) (M : Type) [AddCommGroup M] [Module O M] (h : M → M → O) [Module.Finite O M] [Module.Projective O M] (A : Over S) [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (t : IsTensor A b G a M) (N : LB A.left), L.IsLB → N.IsLB → (∃ tθ : LB.Gen (Theta A N), tθ.IsGen) → (∀ m n, LBIso (cubeLB A ![fst G G ≫ t.tmul m, snd G G ≫ t.tmul n] N) (cubeLB G ![fst G G ≫ a.act (h m n), snd G G] L)) → ((∀ c, bar (bar c) = c) ∧ (∀ m m' n, h (m + m') n = h m n + h m' n) ∧ (∀ (c : O) m n, h (c • m) n = c ∗ h m n) ∧ ∀ m n, h n m = bar (h m n)) → (∀ l : M →ₗ[O] O, ∃! n, ∀ m, l m = h m n) → Out.KTriv G L → Out.KTriv A N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KernelPolarisationTrivialOfSectionsVanishZero

theorem **appLE_top_eq_zero_iff_gs** {X Z : Scheme.{0}} (r : Z ⟶ X) (U : X.Opens) (h : ⊤ ≤ r ⁻¹ᵁ U) (a : Γ(X, U)) : (r.appLE U ⊤ h).hom a = 0 ↔ ∀ z, gs r z U a = 0

theorem **exists_pull_sec_sval** {X Y : Scheme.{0}} {L : LB X} (θ : L.Sec) (F : Y ⟶ X) : ∃ s : (L.pull F).Sec, ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J : (Z ⟶ X) ⟶ L.ι), Good L z J → sval s r z J = gs (r ≫ F) z (L.U (J (r ≫ F))) (θ.f (J (r ≫ F)))

theorem **exists_sec_of_gen_diff** {X Y : Scheme.{0}} {L : LB X} (hL : L.IsLB) (θ : L.Sec) (ψ φ : Y ⟶ X) (d : LB.Gen (LB.LinComb L (K

theorem **kTriv_of_sections_vanish_on_zero** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] (L : LB A.left) (θ : L.Sec), L.IsLB → (∀ (T : Scheme.{0}) (t : T ⟶ A.left) (i : L.ι) (h : ⊤ ≤ t ⁻¹ᵁ L.U i), (t.appLE (L.U i) ⊤ h).hom (θ.f i) = 0 ↔ t = t ≫ A.hom ≫ eSec A) → (∀ (R' : Type) [CommRing R'] (f : Sp R' ⟶ S) (Y : Scheme.{0}) (φ : Y ⟶ A.left) (q : Y ⟶ Sp R'), IsPullback φ q A.hom f → ∀ (s : (L.pull φ).Sec) (T : Scheme.{0}) (t : T ⟶ Y) (j : (L.pull φ).ι) (h : ⊤ ≤ t ⁻¹ᵁ (L.pull φ).U j), t ≫ φ = t ≫ φ ≫ A.hom ≫ eSec A → (t.appLE ((L.pull φ).U j) ⊤ h).hom (s.f j) = 0) → Out.KTriv A L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KernelPolarisationTrivialOfSectionsVanishZeroStatement

theorem **_root_.GlobalTheory.T.kernelPolarisation_trivial_of_sections_vanish_zero** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] (L : LB A.left) (θ : L.Sec), L.IsLB → (∀ (T : Scheme.{0}) (t : T ⟶ A.left) (i : L.ι) (h : ⊤ ≤ t ⁻¹ᵁ L.U i), (t.appLE (L.U i) ⊤ h).hom (θ.f i) = 0 ↔ t = t ≫ A.hom ≫ eSec A) → (∀ (R' : Type) [CommRing R'] (f : Sp R' ⟶ S) (Y : Scheme.{0}) (φ : Y ⟶ A.left) (q : Y ⟶ Sp R'), IsPullback φ q A.hom f → ∀ (s : (L.pull φ).Sec) (T : Scheme.{0}) (t : T ⟶ Y) (j : (L.pull φ).ι) (h : ⊤ ≤ t ⁻¹ᵁ (L.pull φ).U j), t ≫ φ = t ≫ φ ≫ A.hom ≫ eSec A → (t.appLE ((L.pull φ).U j) ⊤ h).hom (s.f j) = 0) → Out.KTriv A L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KeyLocAtOfIsLocalModel

theorem **mem_range_iff_valuation_le_one** {R K Γ : Type*} [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] [Field K] [Algebra R K] [IsFractionRing R K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) (hR : ∀ r : R, v (algebraMap R K r) ≤ 1) {π : K} (hπ0 : v π ≠ 0) (hπ : v π < 1) (y : K) : y ∈ Set.range (algebraMap R K) ↔ v y ≤ 1

theorem **keyLocAt_of_isLocalModel** : ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (p : HeightOneSpectrum (𝒪 K)) (a : K) (M : Matrix (Fin x.P.g) (Fin x.P.g) K), KeyPow.IsLocalModel x p a M → IsLocGen p a T.I → IsLocGen p M.det abelian₂.I → KeyLocAt x T abelian₂ p

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KeyLocAtOfIsLocalModelStatement

theorem **_root_.GlobalTheory.T.keyLocAt_of_isLocalModel'** : ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (p : HeightOneSpectrum (𝒪 K)) (a : K) (M : Matrix (Fin x.P.g) (Fin x.P.g) K), KeyPow.IsLocalModel x p a M → IsLocGen p a T.I → IsLocGen p M.det abelian₂.I → KeyLocAt x T abelian₂ p

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.KeyPBIdOfKeyPB

theorem **keyPB_id_of_keyPB** : ∀ (R K : Type) [CommRing R] [Field K] (f : R →+* K) (x : KeyIn R) (x₁ x₂ : KeyIn K), KeyPB x₁ x f → KeyPB x₂ x f → KeyPB x₁ x₂ (RingHom.id K)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LamDegStLogOm

theorem **GlobalTheory.LamF.lamDegSt_logOm** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → LamDegSt H (logOm E χ).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LamEqOfWeilIsAllPoints

private theorem **pnsmulPt_eq_pow** {S : Scheme.{0}} {G T : Over S} [GrpObj G] (n : ℕ) (a : T ⟶ G) : pnsmulPt n a = a ^ n

theorem **lamEq_of_weilIs_all_points** : Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A B : Out.PolAV (Sp (𝒪 K))) (f : A.A ⟶ B.A) (g : A.A ⟶ A.A), IsMonHom f → IsMonHom g → (∀ (Tb : Over (Sp (𝒪 K))) (k : ℕ) (x y : Tb ⟶ A.A) (c : Γ(Tb.left, ⊤)ˣ), 0 < k → pnsmulPt k x = toUnit Tb ≫ η[A.A] → pnsmulPt k y = toUnit Tb ≫ η[A.A] → (Out.WeilIs B.A B.L k (x ≫ f) (y ≫ f) c ↔ Out.WeilIs A.A A.L k (x ≫ g) y c)) → LamEq B.L f f A.L g (𝟙 A.A)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LamIndLenDeg

theorem **GlobalTheory.LamF.lam_ind_len_deg** : Formalism.L2.Pj.abelian₂.LamIndSt ∧ Formalism.L2.Pj.abelian₂.LamLenSt ∧ ∀ (D : D7) (Q : Par D), Q.PinHost → ∀ large₂ : SizeData₂ D, Formalism.L2.Pj.abelian₂.LamDegSt Q.H large₂.logOm

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LamLenOfIsHerm

theorem **lamLen_of_isHerm** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' f : ℕ) (o : LamObj c nΨ n') (La : ℝ), LamObjGood o f La → ∀ L : LamLetters o, L.IsGood → ∀ (hf : IsPolFibre (algebraMap (𝒪 o.K₁) o.K₁) o.B o.xB.P o.φB) (h : HermFam o.K₁ (Fin o.xB.P.g → o.K₁)), (GenFib.mk o.xB.P o.φB hf o.xB.v o.xB.basis : GenFib o.B).IsHerm h → ∀ (σ : o.K₁ →+* ℂ), o.LamL.n σ = Real.sqrt (h.gram σ (fun i : Fin o.xB.P.g => Pi.single i 1)).det.re ∧ ∀ k, (h.form σ (L.U k) (L.U k)).re ≤ (La * (L.ℓ k).n σ) ^ 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LambdaBundleIsoOfWeilPairingCompatible

theorem **prod_sq** (ρ : T' ⟶ T) (a b : T ⟶ A) {U : Type} [CommGroup U] (h : (T'.left ⟶ A.left) → U) : ∏ I, h (ρ.left ≫ sqφ a b I) ^ sqε I = h ((ρ ≫ a) * (ρ ≫ b)).left * (h (ρ ≫ a).left)⁻¹ * (h (ρ ≫ b).left)⁻¹ * h (1 : T' ⟶ A).left

theorem **exists_gen_axis** {L : LB A.left} (hL : L.IsLB) (ρ : T' ⟶ T) (a b : T ⟶ A) (H : ρ ≫ a = 1 ∨ ρ ≫ b = 1) : ∃ t : LB.Gen (LB.LinComb L (fun I => ρ.left ≫ sqφ a b I) sqε), t.IsGen

theorem **exists_gen_div** {X X' Y Y' : Scheme.{0}} {K K' : Type} [Fintype K] [Fintype K'] (L : LB X) (L' : LB X') (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X')) (ε' : K' → ℤ) (f : Y' ⟶ Y) (H : LBIso (LB.LinComb L (fun k => f ≫ φ k) ε) (LB.LinComb L' (fun k => f ≫ φ' k) ε')) : ∃ t : LB.Gen ((LB.div (LB.LinComb L' φ' ε') (LB.LinComb L φ ε)).pull f), t.IsGen

theorem **exists_gen_fibre** (hG : GlobalFnsSt) (hC : CubeSt) (P : PolAV (Sp k)) (f₁ : X ⟶ P.A) (f₂ : Y ⟶ P.A) [IsMonHom f₁] [IsMonHom f₂] (n : ℕ) (x : 𝟙_ (Over (Sp k)) ⟶ X) (hx : x ^ n = 1) : ∃ u : LB.Gen (LB.LinComb P.L (fun I => (mulA Y n).left ≫ (lift (toUnit Y ≫ x) (𝟙 Y)).left ≫ sqφ (fst X Y ≫ f₁) (snd X Y ≫ f₂) I) sqε), u.IsGen ∧ ∀ (y : 𝟙_ (Over (Sp k)) ⟶ Y) (c : Γ((𝟙_ (Over (Sp k))).left, ⊤)ˣ), y ^ n = 1 → WeilIs P.A P.L n (x ≫ f₁) (y ≫ f₂) c → ∀ (z : Y.left) (J), Good P.L z J → val u ((toUnit Y ≫ y) * 𝟙 Y).left z J = val u (𝟙 _) z J * gm (toUnit Y).left z ⊤ c

theorem **lamEq_of_weilEq** : Out.Laws → ∀ (k : Type) [Field k] [IsAlgClosed k] [CharZero k] (X Y A B : Out.PolAV (Sp k)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → WeilEq A.L f₁ f₂ B.L g₁

g₂ (𝟙 (Sp k)) → LamEq A.L f₁ f₂ B.L g₁ g₂

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LambdaBundleIsoOfWeilPairingCompatibleStatement

theorem **_root_.GlobalTheory.T.lambdaBundleIso_of_weilPairing_compatible** : Out.Laws → ∀ (k : Type) [Field k] [IsAlgClosed k] [CharZero k] (X Y A B : Out.PolAV (Sp k)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → WeilEq A.L f₁ f₂ B.L g₁ g₂ (𝟙 (Sp k)) → LamEq A.L f₁ f₂ B.L g₁ g₂

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LatticeCmEqIdealAndCovolume

theorem **exists_im_ne_zero** (σ : K →+* ℂ) (hσ : ¬ ComplexEmbedding.IsReal σ) : ∃ a : 𝒪 K, (σ (a : K)).im ≠ 0

theorem **ω₁_ne_zero** (L : PeriodPair) : L.ω₁ ≠ 0

theorem **exists_ideal** (σ : K →+* ℂ) (hσ : ¬ ComplexEmbedding.IsReal σ) (L : PeriodPair) (hL : ∀ (a : 𝒪 K), ∀ z ∈ L.lattice, σ (a : K) * z ∈ L.lattice) : ∃ (𝔞 : Ideal (𝒪 K)) (c : ℂ), 𝔞 ≠ ⊥ ∧ c ≠ 0 ∧ ∀ z, z ∈ L.lattice ↔ ∃ a ∈ 𝔞, z = c * σ (a : K)

theorem **im_mul_conj_eq_det** (σ : K →+* ℂ) (b : Basis (Fin 2) ℚ K) (v : Fin 2 → K) : (σ (v 0) * starRingEnd ℂ (σ (v 1))).im = (b.det v : ℚ) * (σ (b 0) * starRingEnd ℂ (σ (b 1))).im

theorem **covol_eq** (σ : K →+* ℂ) (b : Basis (Fin 2) ℤ (𝒪 K)) (L : PeriodPair) (I : FractionalIdeal (𝒪 K)⁰ K) (c : ℂ) (hc : c ≠ 0) (h : ∀ z, z ∈ L.lattice ↔ ∃ x ∈ I, z = c * σ x) : |(L.ω₁ * starRingEnd ℂ L.ω₂).im| = ‖c‖ ^ 2 * (FractionalIdeal.absNorm I : ℚ) * |(σ (b 0 : K) * starRingEnd ℂ (σ (b 1 : K))).im|

theorem **mem_iff** (σ : K →+* ℂ) (L L' : PeriodPair) (𝔟 : Ideal (𝒪 K)) (β : ℂ) (h𝔟 : 𝔟 ≠ ⊥) (hβ : β ≠ 0) (h1 : ∀ x : K, (∀ b ∈ 𝔟, IsIntegral ℤ (x * (b : K))) → ∀ z ∈ L.lattice, β * σ x * z ∈ L'.lattice) (h2 : ∀ y ∈ 𝔟, ∀ z ∈ L'.lattice, β⁻¹ * σ (y : K) * z ∈ L.lattice) (z : ℂ) : z ∈ L'.lattice ↔ ∀ y ∈ 𝔟, β⁻¹ * σ (y : K) * z ∈ L.lattice

theorem **lattice_cm_eq_ideal_and_covolume** : ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ : K →+* ℂ), ¬ NumberField.ComplexEmbedding.IsReal σ → ∀ (L L' : PeriodPair), (∀ (a : 𝒪 K), ∀ z ∈ L.lattice, σ (a : K) * z ∈ L.lattice) → (∃ (𝔞 : Ideal (𝒪 K)) (c : ℂ), 𝔞 ≠ ⊥ ∧ c ≠ 0 ∧ ∀ z, z ∈ L.lattice ↔ ∃ a ∈ 𝔞, z = c * σ (a : K)) ∧ ∀ (𝔟 : Ideal (𝒪 K)) (β : ℂ), 𝔟 ≠ ⊥ → β ≠ 0 → (∀ x : K, (∀ b ∈ 𝔟, IsIntegral ℤ (x * (b : K))) → ∀ z ∈ L.lattice, β * σ x * z ∈ L'.lattice) → (∀ y ∈ 𝔟, ∀ z ∈ L'.lattice, β⁻¹ * σ (y : K) * z ∈ L.lattice) → (∀ z, z ∈ L'.lattice ↔ ∀ y ∈ 𝔟, β⁻¹ * σ (y : K) * z ∈ L.lattice) ∧ |(L'.ω₁ * starRingEnd ℂ L'.ω₂).im| * Ideal.absNorm 𝔟 = ‖β‖ ^ 2 * |(L.ω₁ * starRingEnd ℂ L.ω₂).im|

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LatticeDeterminedByInvariants

theorem **lattice_le_of_g₂_eq_of_g₃_eq** {L L' : PeriodPair} (h₂ : L.g₂ = L'.g₂) (h₃ : L.g₃ = L'.g₃) : L.lattice ≤ L'.lattice

theorem **lattice_eq_of_g₂_eq_of_g₃_eq** {L L' : PeriodPair} (h₂ : L.g₂ = L'.g₂) (h₃ : L.g₃ = L'.g₃) : L.lattice = L'.lattice

theorem **g₂_cube_sub_g₃_sq_ne_zero** (L : PeriodPair) : L.g₂ ^ 3 - 27 * L.g₃ ^ 2 ≠ 0

theorem **IsHomothetic.g₂_eq** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') : L'.g₂ = (c ^ 4)⁻¹ * L.g₂

theorem **IsHomothetic.g₃_eq** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') : L'.g₃ = (c ^ 6)⁻¹ * L.g₃

theorem **exists_scale_of_j_eq** {a b a' b' : ℂ} (h : a ^ 3 - 27 * b ^ 2 ≠ 0) (h' : a' ^ 3 - 27 * b' ^ 2 ≠ 0) (e : a ^ 3 * (a' ^ 3 - 27 * b' ^ 2) = a' ^ 3 * (a ^ 3 - 27 * b ^ 2)) : ∃ u : ℂ, u ≠ 0 ∧ a' = u ^ 2 * a ∧ b' = u ^ 3 * b

theorem **exists_isHomothetic_of_j_eq** {L L' : PeriodPair} (e : L.g₂ ^ 3 * (L'.g₂ ^ 3 - 27 * L'.g₃ ^ 2) = L'.g₂ ^ 3 * (L.g₂ ^ 3 - 27 * L.g₃ ^ 2)) : ∃ c : ℂ, IsHomothetic c L L'

theorem **lattice_determined_by_invariants** (L L' : PeriodPair) : (L.g₂ = L'.g₂ → L.g₃ = L'.g₃ → L.lattice = L'.lattice) ∧ (L.g₂ ^ 3 * (L'.g₂ ^ 3 - 27 * L'.g₃ ^ 2) = L'.g₂ ^ 3 * (L.g₂ ^ 3 - 27 * L.g₃ ^ 2) → ∃ c : ℂ, c ≠ 0 ∧ ∀ z, z ∈ L'.lattice ↔ ∃ w ∈ L.lattice, z = c * w)

theorem **g₂_cube_sub_g₃_sq_eq_discriminant** : L.g₂ ^ 3 - 27 * L.g₃ ^ 2 = (2 * π / a) ^ 12 * ModularForm.discriminant τ

theorem **g₂_cube_sub_g₃_sq_eq_eta** : L.g₂ ^ 3 - 27 * L.g₃ ^ 2 = (2 * π / a) ^ 12 * ModularForm.eta τ ^ 24

theorem **g₂_cube_div_eq_modularJ** : 1728 * L.g₂ ^ 3 / (L.g₂ ^ 3 - 27 * L.g₃ ^ 2) = modularJ τ

theorem **mem_lattice_iff_of_periods_eq** {L : PeriodPair} {τ : ℍ} (h : (L.ω₁ = 1 ∧ L.ω₂ = τ) ∨ (L.ω₁ = τ ∧ L.ω₂ = 1)) (x : ℂ) : x ∈ L.lattice ↔ ∃ m n : ℤ, 1 * (m * (τ : ℂ) + n) = x

theorem **invariants_of_periods_eq** {L : PeriodPair} {τ : ℍ} (h : (L.ω₁ = 1 ∧ L.ω₂ = τ) ∨ (L.ω₁ = τ ∧ L.ω₂ = 1)) : L.g₂ ^ 3 - 27 * L.g₃ ^ 2 = (2 * π) ^ 12 * ModularForm.discriminant τ ∧ L.g₂ ^ 3 - 27 * L.g₃ ^ 2 = (2 * π) ^ 12 * ModularForm.eta τ ^ 24 ∧ 1728 * L.g₂ ^ 3 / (L.g₂ ^ 3 - 27 * L.g₃ ^ 2) = modularJ τ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LatticeEqSpanHostIdealSmul

theorem **dataX_lattice_eq_smul_of_le** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : DataX E r) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (hone : FractionalIdeal.spanSingleton (nonZeroDivisors (𝒪 E)) X.ξ0 * 𝔄 * conjFrac 𝔄 * (differentIdeal ℤ (𝒪 E) : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) = 1) (hle : (𝔄 : Submodule (𝒪 E) E) • X.L' ≤ X.Λ) : X.Λ = (𝔄 : Submodule (𝒪 E) E) • X.L'

theorem **lattice_eq_span_hostIdeal_smul** : ∀ (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ), InR D j → ∀ (k : (T.U j T.σ₀).Comp) (Xx : DataX D.host.E D.R.r), Xx.toData = (T.U j T.σ₀).X k → (∀ a ∈ D.host.AA, ∀ v ∈ Xx.L', a • v ∈ Xx.Λ) → Xx.Λ = Submodule.span (𝒪 D.host.E) {v | ∃ a ∈ D.host.AA, ∃ u ∈ Xx.L', v = a • u}

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LatticeUnifFibreEqOfPeriod

theorem **lattice_unif_fibre_eq_of_period** : ThetaUnifAV → ∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (s : R →+* ℂ) (L : PeriodPair) (c₀ ω : ℂ), P.g = 1 → IsTangent v → (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s P P' φ' → IsVecOver s φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ L.lattice ↔ z • w' ∈ u'.Λ) → c₀ ≠ 0 → (∀ z, z ∈ L.lattice ↔ ∃ m n : ℤ, z = c₀ * (m + n * ω)) → ∀ (P' : Out.PolAV (Sp ℂ)) (φ : P'.A.left ⟶ P.A.left), IsPolFibre s P P' φ → ∀ u : HolUnif P'.A P'.L P'.g, ∃ c : ℂ, c ≠ 0 ∧ ∀ z, z ∈ u.Λ ↔ ∃ m n : ℤ, z = fun _ => c * (m + n * ω)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LbIsoCubeLBMulRightOf

theorem **exists_gen_tensor** {C A B : LB Y} (a : C.ι → A.ι) (b : C.ι → B.ι) (la : ∀ i, C.U i ≤ A.U (a i)) (lb : ∀ i, C.U i ≤ B.U (b i)) (hg : ∀ (y : Y) i j, y ∈ C.U i → y ∈ C.U j → gm (𝟙 Y) y (C.U i ⊓ C.U j) (C.g i j) = gm (𝟙 Y) y (A.U (a i) ⊓ A.U (a j)) (A.g (a i) (a j)) * gm (𝟙 Y) y (B.U (b i) ⊓ B.U (b j)) (B.g (b i) (b j))) (s : LB.Gen A) (hs : s.IsGen) (t : LB.Gen B) (ht : t.IsGen) : ∃ u : LB.Gen C, u.IsGen

theorem **lbIso_sum** {L : LB X} {M : LB X'} {K K' J J' : Type} [Fintype K] [Fintype K'] [Fintype J] [Fintype J'] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} {ψ : J → (Y ⟶ X')} {δ : J → ℤ} {ψ' : J' → (Y ⟶ X')} {δ' : J' → ℤ} (h : LBIso (LB.LinComb L φ ε) (LB.LinComb M ψ δ)) (h' : LBIso (LB.LinComb L φ' ε') (LB.LinComb M ψ' δ')) : LBIso (LB.LinComb L (Sum.elim φ φ') (Sum.elim ε ε')) (LB.LinComb M (Sum.elim ψ ψ') (Sum.elim δ δ'))

theorem **cubeLB_isLB** (hL : L.IsLB) {n : ℕ} (x : Fin n → (T ⟶ A)) : (cubeLB A x L).IsLB

theorem **class_cubeLB** {U : Type*} [CommGroup U] (cl : LB T.left → U) (m : (T.left ⟶ A.left) → U) (h : ∀ (I : Type) [Fintype I] (φ : I → (T.left ⟶ A.left)) (ε : I → ℤ), cl (LB.LinComb L φ ε) = ∏ k, m (φ k) ^ ε k) (x y : T ⟶ A) : cl (cubeLB A ![x, y] L) = m (x ∗ y).left ∗ (m x.left)⁻¹ ∗ (m y.left)⁻¹ ∗ m (1 : T ⟶ A).left

theorem **lbIso_cubeLB_swap** (hL : L.IsLB) (x y : T ⟶ A) : LBIso (cubeLB A ![x, y] L) (cubeLB A ![y, x] L)

theorem **lbIso_cubeLB_mul_left** (hL : L.IsLB) (hT : ∃ t : LB.Gen (Theta A L), t.IsGen) (x x' y : T ⟶ A) : LBIso (cubeLB A ![x ∗ x', y] L) (cubeProd L x y x' y)

theorem **lbIso_cubeLB_mul_right** (hL : L.IsLB) (hT : ∃ t : LB.Gen (Theta A L), t.IsGen) (x y y' : T ⟶ A) : LBIso (cubeLB A ![x, y ∗ y'] L) (cubeProd L x y x y')

theorem **lbIso_cubeLB_mul_left_of** (hL : L.IsLB) (hM : M.IsLB) (hT : ∃ t : LB.Gen (Theta A L), t.IsGen) (hT' : ∃ t : LB.Gen (Theta B M), t.IsGen) {x x' y : T ⟶ A} {u u' v : T ⟶ B} (h : LBIso (cubeLB A ![x, y] L) (cubeLB B ![u, v] M)) (h' : LBIso (cubeLB A ![x', y] L) (cubeLB B ![u', v] M)) : LBIso (cubeLB A ![x ∗ x', y] L) (cubeLB B ![u ∗ u', v] M)

theorem **lbIso_cubeLB_mul_right_of** (hL : L.IsLB) (hM : M.IsLB) (hT : ∃ t : LB.Gen (Theta A L), t.IsGen) (hT' : ∃ t : LB.Gen (Theta B M), t.IsGen) {x y y' : T ⟶ A} {u v v' : T ⟶ B} (h : LBIso (cubeLB A ![x, y] L) (cubeLB B ![u, v] M)) (h' : LBIso (cubeLB A ![x, y'] L) (cubeLB B ![u, v'] M)) : LBIso (cubeLB A ![x, y ∗ y'] L) (cubeLB B ![u, v ∗ v'] M)

theorem **lbIso_cubeLB_hom** (ψ : A ⟶ B) [IsMonHom ψ] (h : LBIso L (M.pull ψ.left)) (x y : T ⟶ A) : LBIso (cubeLB A ![x, y] L) (cubeLB B ![x ≫ ψ, y ≫ ψ] M)

theorem **lbIso_of_lamEq** (h : LamEq L f₁ f₂ M g₁ g₂) (x : T ⟶ X) (y : T ⟶ Y) : LBIso (cubeLB A ![x ≫ f₁, y ≫ f₂] L) (cubeLB B ![x ≫ g₁, y ≫ g₂] M)

theorem **lamEq_calculus** : ∀ (S : Scheme.{0}) (X Y A B C : Over S) [GrpObj A] [IsCommMonObj A] [GrpObj B] [IsCommMonObj B] [GrpObj C] [IsCommMonObj C] (L : LB A.left) (M : LB B.left) (N : LB C.left), L.IsLB → M.IsLB → N.IsLB → ∀ (f₁ f₁' : X ⟶ A) (f₂ f₂' : Y ⟶ A) (g₁ g₁' : X ⟶ B) (g₂ g₂' : Y ⟶ B) (k₁ : X ⟶ C) (k₂ : Y ⟶ C), LamEq L f₁ f₂ L f₁ f₂ ∧ (LamEq L f₁ f₂ M g₁ g₂ → LamEq M g₁ g₂ L f₁ f₂ ∧ LamEq L f₂ f₁ M g₂ g₁ ∧ (LamEq M g₁ g₂ N k₁ k₂ → LamEq L f₁ f₂ N k₁ k₂) ∧ (∀ (X' Y' : Over S) (a : X' ⟶ X) (b : Y' ⟶ Y), LamEq L (a ≫ f₁) (b ≫ f₂) M (a ≫ g₁) (b ≫ g₂)) ∧ ∀ (T : Over S) (x : T ⟶ X) (y : T ⟶ Y), LBIso (cubeLB A ![x ≫ f₁, y ≫ f₂] L) (cubeLB B ![x ≫ g₁, y ≫ g₂] M)) ∧ ((∃ t : LB.Gen (Theta A L), t.IsGen) → (∃ t : LB.Gen (Theta B M), t.IsGen) → LamEq L f₁ f₂ M g₁ g₂ → (LamEq L f₁' f₂ M g₁' g₂ → LamEq L (f₁ ∗ f₁') f₂ M (g₁ ∗ g₁') g₂) ∧ (LamEq L f₁ f₂' M g₁ g₂' → LamEq L f₁ (f₂ ∗ f₂') M g₁ (

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LbIsoPowOfHermFormPullEq

lemma **val_resG** (x : Sp ℂ ⟶ X) {U V : X.Opens} (hVU : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵁ V) (f : Γ(X, U)) : valAt' x V (resG X hVU f) = valAt' x U f

lemma **val_mul** (x : Sp ℂ ⟶ X) {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (f f' : Γ(X, U)) : valAt' x U (f ∗ f') = valAt' x U f ∗ valAt' x U f'

lemma **val_unit_inv** (x : Sp ℂ ⟶ X) {U : X.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (f : Γ(X, U)ˣ) : valAt' x U (f⁻¹).val = (valAt' x U f.val)⁻¹

lemma **exists_of_iSup** (x : Sp ℂ ⟶ X) {I : Type*} (U : I → X.Opens) (h : ⊤ ≤ x ⁻¹ᵁ (⨆ i, U i)) : ∃ i, ⊤ ≤ x ⁻¹ᵁ U i

lemma **exists_of_local_quotient** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] {E I : Type} (e : E → (Sp ℂ ⟶ X)) (he : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ∃ z, e z = x) (W : X.Opens) (φ : E → ℂ) (a b : I → Γ(X, W)) (hb : ∀ z, ⊤ ≤ e z ⁻¹ᵁ W → ∃ i, valAt' (e z) W (b i) ≠ 0) (hab : ∀ i z, ⊤ ≤ e z ⁻¹ᵁ W → valAt' (e z) W (a i) = φ z ∗ valAt' (e z) W (b i)) : ∃ c : Γ(X, W), ∀ z, ⊤ ≤ e z ⁻¹ᵁ W → valAt' (e z) W c = φ z

lemma **pow_chart_iff** (x : Sp ℂ ⟶ X) (L : LB X) (n : ℕ) (J : PUnit → L.ι) : ⊤ ≤ x ⁻¹ᵁ (lbPow L n).U J ↔ ⊤ ≤ x ⁻¹ᵁ L.U (J PUnit.unit)

lemma **pull_chart_iff** {Y : Scheme.{0}} (x : Sp ℂ ⟶ X) (L : LB Y) (f : X ⟶ Y) (J : PUnit → L.ι) : ⊤ ≤ x ⁻¹ᵁ (L.pull f).U J ↔ ⊤ ≤ (x ≫ f) ⁻¹ᵁ L.U (J PUnit.unit)

lemma **val_pow_g** (x : Sp ℂ ⟶ X) (L : LB X) (n : ℕ) (J J' : PUnit → L.ι) (h : ⊤ ≤ x ⁻¹ᵁ (lbPow L n).U J) (h' : ⊤ ≤ x ⁻¹ᵁ (lbPow L n).U J') : valAt' x ((lbPow L n).U J ⊓ (lbPow L n).U J') ((lbPow L n).g J J').val = valAt' x (L.U (J PUnit.unit) ⊓ L.U (J' PUnit.unit)) (L.g _ _).val ^ n

lemma **val_pull_g** {Y : Scheme.{0}} (x : Sp ℂ ⟶ X) (L : LB Y) (f : X ⟶ Y) (J J' : PUnit → L.ι) (h : ⊤ ≤ x ⁻¹ᵁ (L.pull f).U J) (h' : ⊤ ≤ x ⁻¹ᵁ (L.pull f).U J') : valAt' x ((L.pull f).U J ⊓ (L.pull f).U J') ((L.pull f).g J J').val = valAt' (x ≫ f) (L.U (J PUnit.unit) ⊓ L.U (J' PUnit.unit)) (L.g _ _).val

lemma **lbIso_of_frames** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] {E : Type} (e : E → (Sp ℂ ⟶ X)) (he : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ∃ z, e z = x) (M N : LB X) (α : M.ι → E → ℂ) (β : N.ι → E → ℂ) (hα : ∀ i z, ⊤ ≤ e z ⁻¹ᵁ M.U i → α i z ≠ 0) (hβ : ∀ i z, ⊤ ≤ e z ⁻¹ᵁ N.U i → β i z ≠ 0) (hα' : ∀ i j z, ⊤ ≤ e z ⁻¹ᵁ M.U i → ⊤ ≤ e z ⁻¹ᵁ M.U j → α i z = valAt' (e z) (M.U i ⊓ M.U j) (M.g i j).val ∗ α j z) (hβ' : ∀ i j z, ⊤ ≤ e z ⁻¹ᵁ N.U i → ⊤ ≤ e z ⁻¹ᵁ N.U j → β i z = valAt' (e z) (N.U i ⊓ N.U j) (N.g i j).val ∗ β j z) (h : ∀ i j, ∃ c : Γ(X, M.U i ⊓ N.U j), ∀ z, ⊤ ≤ e z ⁻¹ᵁ (M.U i ⊓ N.U j) → valAt' (e z) _ c = α i z / β j z) : LBIso N M

lemma **exists_generating_sections** (P : Out.PolAV (Sp ℂ)) : ∃ N : ℕ, 1 ≤ N ∧ ∃ (m : ℕ) (s : Fin (m + 1) → (lbPow P.L N).Sec), ⊔ i, Out.nonzeroLocus (s i) = ⊤

theorem **lbIso_pow_of_hermForm_pull_eq** : ∀ (P P' : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g) (u' : HolUnif P'.A P'.L P'.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ) (Hm' : Matrix (Fin P'.g) (Fin P'.g) ℂ) (χ' : u'.Λ → ℂ), u.IsAH Hm χ → u'.IsAH Hm' χ' → (∀ (k : ℕ) θ, u.IsThetaFn Hm χ k θ → ∃! s, u.IsFnPow k s θ) → (∀ (k : ℕ) s, ∃! θ, u'.IsThetaFn Hm' χ' k θ ∧ u'.IsFnPow k s θ) → ∀ (F : (Fin P.g → ℂ) →₁[ℂ] (Fin P'.g → ℂ)) (n : ℕ), (∀ l ∈ u.Λ, F l ∈ u'.Λ) → (∀ z w, hermForm Hm' (F z) (F w) = (n : ℂ) ∗ hermForm Hm z w) → ∀ f : P.A ⟶ P'.A, (∀ z, u'.ex (F z) = u.ex z ≫ f) → ∃ N : ℕ, 1 ≤ N ∧ LBIso (lbPow (P'.L.pull f.left) N) (lbPow P.L (N ∗ n))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LeftEqOfIsLieCombSingle

theorem **prod_ofFn_eq_single** {M : Type*} [Monoid M] : ∀ {n : ℕ} (u : Fin n → M) (i : Fin n), (∀ j, j ≠ i → u j = 1) → (List.ofFn u).prod = u i | 0, _, i, _ => i.elim0 | n + 1, u, i, h => by rw [List.ofFn_succ, List.prod_cons] refine Fin.cases (motive

theorem **left_eq_of_isLieComb_single** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] {n : ℕ} {v : Fin n → (epsR R ⟶ G)} (hv : ∀ i, IsTangent (v i)) (i : Fin n) (t : R) {w : epsR R ⟶ G} (hw : IsLieComb (Pi.single i t) v w) : w.left = spMap (epsScaleR t) ≫ (v i).left

theorem **exists_scale_of_eigen** {K : Type} [Field K] [NumberField K] {A B : Over (Sp (𝒪 K))} {A' B' : Over (Sp K)} [GrpObj A'] [GrpObj B'] {n : ℕ} {v : Fin n → (epsR K ⟶ A')} (hv : ∀ i, IsTangent (v i)) (i : Fin n) (t : K) (φA : A'.left ⟶ A.left) (φB : B'.left ⟶ B.left) (α : A ⟶ A) (β : B ⟶ B) (F : A ⟶ B) (hF : α ≫ F = F ≫ β) {w₁ : epsR K ⟶ A'} (hw₁ : IsLieComb (Pi.single i t) v w₁) (he : (v

i).left ≫ φA ≫ α.left = w₁.left ≫ φA) {w : epsR K ⟶ B'} (hw : w.left ≫ φB = (v i).left ≫ φA ≫ F.left) : ∃ w' : epsR K ⟶ B', IsLieComb ![t] ![w] w' ∧ w.left ≫ φB ≫ β.left = w'.left ≫ φB

theorem **letter_eigen** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' f : ℕ) (o : LamObj c nΨ n') (La : ℝ), LamObjGood o f La → ∀ L : LamLetters o, L.IsGood → ∀ (b : E →+* ℂ) (i : Fin (L.m b)) (w : epsR o.K₁ ⟶ o.xB.P.A), L.IsLetter (L.e (b, i)) w → ∀ a : 𝒪 E, ∃ w' : epsR o.K₁ ⟶ o.xB.P.A, IsLieComb ![labK c o.K₁ b (a : E)] ![w] w' ∧ w.left ≫ o.φB ≫ (o.oa.act a).left = w'.left ≫ o.φB

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieAlgebraEigenDecompositionCMSignature

private theorem **hom_ext_lemma** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {f : S' ⟶ S} {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) {x y : G' ⟶ G'} (hxy : x.left ≫ φ = y.left ≫ φ) : x = y

theorem **exists_oAct** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] [GrpObj G'] {f : S' ⟶ S} {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) {O : Type} [CommRing O] (act : OAct O G) (b : O → (G' ⟶ G')) (hb : ∀ a, IsMonHom (b a)) (e : ∀ a, (b a).left ≫ φ = φ ≫ (act.act a).left) : ∃ act' : OAct O G', act'.act = b

theorem **finrank_eigenSub** {K : Type} [Field K] [NumberField K] {P : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) {O : Type} [CommRing O] (act : OAct O P.A) (r : O →+* Module.End K (Fin F.P₀.g → K)) (hr : ∀ a, F.IsDiffOf F (act.act a) (r a)) {k : Type} [Field k] (j : K →+* k) (χ : O →+* K) {N : ℕ} (hN : LieMult act (spMap (j.comp (algebraMap (𝒪 K) K))) (j.comp χ) N) : Module.finrank K (EigenSplit.eigenSub r χ) = N

theorem **bar_rt** : H.bar H.rt = -H.rt

theorem **card_half** {K : Type} [Field K] [NumberField K] {ρ : 𝒪 K} (hs : Splits E K) (hρ : ρ ^ 2 = -(H.d : 𝒪 K)) : (∀ τ : 𝒪 E →+* 𝒪 K, ¬ τ H.rt = ρ ↔ τ H.rt = -ρ) ∧ Nat.card {τ : 𝒪 E →+* 𝒪 K // τ H.rt = ρ} = Module.finrank ℚ E / 2 ∧ Nat.card {τ : 𝒪 E →+* 𝒪 K // τ H.rt = -ρ} = Module.finrank ℚ E / 2

private theorem **isSplitting_of_splits** {E₁ K₁ : Type} [Field E₁] [NumberField E₁] [Field K₁] [NumberField K₁] [Fintype (𝒪 E₁ →+* 𝒪 K₁)] (hs : Splits E₁ K₁) : EigenSplit.IsSplitting fun τ : 𝒪 E₁ →+* 𝒪 K₁ => (algebraMap (𝒪 K₁) K₁).comp τ

theorem **eigen_decomposition** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K : Type) [Field K] [NumberField K] (ρ : 𝒪 K) (T : PAT H (Sp (𝒪 K))) (F : GenFib T.pol) (p q : ℕ), Splits E K → HasType H T.t.act ρ p q → ρ ^ 2 = -(H.d : 𝒪 K) → (∀ τ : 𝒪 E →+* 𝒪 K, latRank K (F.eigLat T.t.act τ) = open Classical in if τ H.rt = ρ then p else q) ∧ iSupIndep (fun τ : 𝒪 E →+* 𝒪 K => Submodule.span K (F.eigLat T.t.act τ : Set (Fin F.P₀.g → K))) ∧ ⊔ τ : 𝒪 E →+* 𝒪 K, Submodule.span K (F.eigLat T.t.act τ : Set (Fin F.P₀.g → K)) = ⊤ ∧ T.g = Module.finrank ℚ E / 2 * (p + q) ∧ Nat.card {τ : 𝒪 E →+* 𝒪 K // τ H.rt = ρ} = Module.finrank ℚ E / 2 ∧ Nat.card {τ : 𝒪 E →+* 𝒪 K // τ H.rt = -ρ} = Module.finrank ℚ E / 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieAlgebraEigenDecompositionCMSignatureStatement

theorem **_root_.GlobalTheory.T.lieAlgebra_eigenDecomposition_CM_signature** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K : Type) [Field K] [NumberField K] (ρ : 𝒪 K) (T : PAT H (Sp (𝒪 K))) (F : GenFib T.pol) (p q : ℕ), Splits E K → HasType H T.t.act ρ p q → ρ ^ 2 = -(H.d : 𝒪 K) → (∀ τ : 𝒪 E →+* 𝒪 K, latRank K (F.eigLat T.t.act τ) = open Classical in if τ H.rt = ρ then p else q) ∧ iSupIndep (fun τ : 𝒪 E →+* 𝒪 K => Submodule.span K (F.eigLat T.t.act τ : Set (Fin F.P₀.g → K))) ∧ ⊔ τ : 𝒪 E →+* 𝒪 K, Submodule.span K (F.eigLat T.t.act τ : Set (Fin F.P₀.g → K)) = ⊤ ∧ T.g = Module.finrank ℚ E / 2 * (p + q) ∧ Nat.card {τ : 𝒪 E →+* 𝒪 K // τ H.rt = ρ} = Module.finrank ℚ E / 2 ∧ Nat.card {τ : 𝒪 E →+* 𝒪 K // τ H.rt = -ρ} = Module.finrank ℚ E / 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieAlgebraSquaredLengthCompPairingHom

theorem **push_inj** (h : IsGrpPullback A' A f φ) {T : Over S'} {x y : T ⟶ A'} (e : pushHom h T x = pushHom h T y) : x = y

theorem **push_comp** (h : IsGrpPullback A' A f φ) (h' : IsGrpPullback C' C f ψ) {p : A ⟶ C} {p' : A' ⟶ C'} (e : p'.left ≫ ψ = φ ≫ p.left) {T : Over S'} (x : T ⟶ A') : pushHom h' T (x ≫ p') = pushHom h T x ≫ p

theorem **exists_monHom_lift** (h : IsGrpPullback A' A f φ) (h' : IsGrpPullback C' C f ψ) (p : A ⟶ C) [IsMonHom p] : ∃ p' : A' ⟶ C', IsMonHom p' ∧ p'.left ≫ ψ = φ ≫ p.left

theorem **isVecOver_comp** {R R' : Type} [CommRing R] [CommRing R'] {s : R →+* R'} {A C : Over (Sp R)} {A' C' : Over (Sp R')} {φ : A'.left ⟶ A.left} {ψ : C'.left ⟶ C.left} {v : epsR R ⟶ A} {v' : epsR R' ⟶ A'} (hv : IsVecOver s φ v v') {p : A ⟶ C} {p' : A' ⟶ C'} (e : p'.left ≫ ψ = φ ≫ p.left) : IsVecOver s ψ (v ≫ p) (v' ≫ p')

theorem **lieNormSq_comp** : ThetaUnifAV → Out.Laws → ∀ (R : Type) [CommRing R] (P P' : Out.PolAV (Sp R)) (f : P.A ⟶ P'.A) (g : P.A ⟶ P.A) (v : epsR R ⟶ P.A) (a : R) (s : R →+* ℂ) (r r' : ℝ), IsMonHom f → IsMonHom g → LamEq P'.L f f P.L g (𝟙 P.A) → IsTangent v → IsLieComb ![a] ![v] (v ≫ g) → IsLieNormSq P v s r → IsLieNormSq P' (v ≫ f) s r' → r' = (s a).re * r

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieAlgebraSquaredLengthCompPairingHomStatement

theorem **_root_.GlobalTheory.T.lieAlgebraSquaredLength_comp_pairing_hom** : ThetaUnifAV → Out.Laws → ∀ (R : Type) [CommRing R] (P P' : Out.PolAV (Sp R)) (f : P.A ⟶ P'.A) (g : P.A ⟶ P.A) (v : epsR R ⟶ P.A) (a : R) (s : R →+* ℂ) (r r' : ℝ), IsMonHom f → IsMonHom g → LamEq P'.L f f P.L g (𝟙 P.A) → IsTangent v → IsLieComb ![a] ![v] (v ≫ g) → IsLieNormSq P v s r → IsLieNormSq P' (v ≫ f) s r' → r' = (s a).re * r

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieCombAddPowMap

theorem **isLieComb_zero** [IsCommMonObj G] (hv : ∀ i, IsTangent (v i)) : IsLieComb 0 v 1

theorem **isLieComb_add** [IsCommMonObj G] (hv : ∀ i, IsTangent (v i)) {c c' : Fin n → R} {w w' : epsR R ⟶ G} (h : IsLieComb c v w) (h' : IsLieComb c' v w') : IsLieComb (c + c') v (w * w')

theorem **isLieComb_nsmul** [IsCommMonObj G] (hv : ∀ i, IsTangent (v i)) {c : Fin n → R} {w : epsR R ⟶ G} (h : IsLieComb c v w) (N : ℕ) : IsLieComb (N • c) v (w ^ N)

theorem **isTangent_comp** (f : G ⟶ G') [IsMonHom f] {x : epsR R ⟶ G} (hx : IsTangent x) : IsTangent (x ≫ f)

theorem **list_prod_comp** (f : G ⟶ G') [IsMonHom f] (l : List (epsR R ⟶ G)) : l.prod ≫ f = (l.map (· ≫ f)).prod

theorem **isLieComb_comp** (f : G ⟶ G') [IsMonHom f] {c : Fin n → R} {w : epsR R ⟶ G} (h : IsLieComb c v w) : IsLieComb c (fun i => v i ≫ f) (w ≫ f)

theorem **lieComb_add_pow_map** : ∀ (R : Type) [CommRing R] (G G' : Over (Sp R)) [GrpObj G] [IsCommMonObj G] [GrpObj G'] (n : ℕ) (v : Fin n → (epsR R ⟶ G)), (∀ i, IsTangent (v i)) → IsLieComb 0 v 1 ∧ (∀ c c' w w', IsLieComb c v w → IsLieComb c' v w' → IsLieComb (c + c') v (w * w')) ∧ (∀ c w (N : ℕ), IsLieComb c v w → IsLieComb (N • c) v (w ^ N)) ∧ ∀ f : G ⟶ G', IsMonHom f → (∀ i, IsTangent (v i ≫ f)) ∧ ∀ c w, IsLieComb c v w → IsLieComb c (fun i => v i ≫ f) (w ≫ f)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeFgAndSpanEqTopRingOfIntegers

theorem **exists_smul_eq_sum_rows** {O K : Type} [CommRing O] [IsDomain O] [Field K] [Algebra O K] [IsFractionRing O K] {g : ℕ} (M : Matrix (Fin g) (Fin g) K) (hM : M.det ≠ 0) (c : Fin g → K) : ∃ d : O, d ≠ 0 ∧ ∃ a : Fin g → O, d • c = ∑ j, a j • M j

theorem **fg_and_span_eq_top** {O K : Type} [CommRing O] [IsDomain O] [IsNoetherianRing O] [Field K] [Algebra O K] [IsFractionRing O K] {g n : ℕ} (L : Submodule O (Fin g → K)) (r : Fin n → O) (M : Fin n → Matrix (Fin g) (Fin g) K) (hr : Ideal.span (Set.range r) = ⊤) (hM : ∀ i, (M i).det ≠ 0) (hL : ∀ c, c ∈ L ↔ ∀ i, ∃ (k : ℕ) (a : Fin g → O), r i ^ k • c = ∑ j, a j • M i j) : L.FG ∧ Submodule.span K (L : Set (Fin g → K)) = ⊤

theorem **lieLat_fg_and_span_eq_top** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P), F.lieLat.FG ∧ Submodule.span K (F.lieLat : Set (Fin F.P₀.g → K)) = ⊤

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeFgAndSpanEqTopRingOfIntegersStatement

theorem **_root_.GlobalTheory.T.lieLattice_fg_and_span_eq_top_ringOfIntegers** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P), F.lieLat.FG ∧ Submodule.span K (F.lieLat : Set (Fin F.P₀.g → K)) = ⊤

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeLocalBasisDescription

theorem **isPullback_dualMap** (f : R →+* R') : IsPullback (spMap (Chart.dualMapR f)) (epsR R').hom (epsR R).hom (spMap f)

theorem **exists_glue** (f : R →+* R') (hf : Function.Injective f) {A : Over (Sp R)} [IsSeparated A.hom] {I : Type} (r : I → R) (hr : Ideal.span (Set.range r) = ⊤) (g : ∀ i, Localization.Away (r i) →+* R') (hg : ∀ i, (g i).comp (algebraMap R _) = f) (w : Sp (DualNumber R') ⟶ A.left) (t : ∀ i, Sp (DualNumber (Localization.Away (r i))) ⟶ A.left) (ht : ∀ i, t i ≫ A.hom = (epsR _).hom ≫ spMap (algebraMap R (Localization.Away (r i)))) (hw : ∀ i, spMap (Chart.dualMapR (g i)) ≫ t i = w) : ∃ u : epsR R ⟶ A, spMap (Chart.dualMapR f) ≫ u.left = w

theorem **isTangent_of_isVecOver** {f : R →+* R'} (hf : Function.Injective f) {A : Over (Sp R)} {A' : Over (Sp R')} [GrpObj A] [GrpObj A'] [IsSeparated A.hom] {φ : A'.left ⟶ A.left} (hφ : IsGrpPullback A' A (spMap f) φ) {v : epsR R ⟶ A} {v' : epsR R' ⟶ A'} (h : IsVecOver f φ v v') (hv : IsTangent v') : IsTangent v

theorem **exists_matrix** (hψ : IsGrpPullback A G (spMap g) ψ) (hs : Smooth G.hom) (he : IsLieBasis e) (hv : IsLieBasis v) : ∃ M : Matrix (Fin n) (Fin n) K, M.det ≠ 0 ∧ ∀ j, ∃ w, IsVecOver g ψ (e j) w ∧ IsLieComb (M j) v w

theorem **isVecOver_iff** (hψ : IsGrpPullback A G (spMap g) ψ) (hv : IsLieBasis v) {M : Matrix (Fin n) (Fin n) K} (hM : ∀ j, ∃ w, IsVecOver g ψ (e j) w ∧ IsLieComb (M j) v w) {a : Fin n → R} {t : epsR R ⟶ G} (ht : IsLieComb a e t) (w : epsR K ⟶ A) : IsVecOver g ψ t w ↔ IsLieComb (∑ j, g (a j) • M j) v w

theorem **exists_comb_iff** {R K : Type} [CommRing R] [Field K] [Algebra R K] (r : R) (g : Localization.Away r →+* K) (hg : g.comp (algebraMap R _) = algebraMap R K) {n : ℕ} (M : Matrix (Fin n) (Fin n) K) (c : Fin n → K) : (∃ a : Fin n → Localization.Away r, c = ∑ j, g (a j) • M j) ↔ ∃ (k : ℕ) (a : Fin n → R), r ^ k • c = ∑ j, a j • M j

theorem **exists_finite_cover** {R : Type} [CommRing R] (Q : R → Prop) (h : ∀ p : PrimeSpectrum R, ∃ r ∉ p.asIdeal, Q r) : ∃ (n : ℕ) (r : Fin n → R), Ideal.span (Set.range r) = ⊤ ∧ ∀ i, r i ≠ 0 ∧ Q (r i)

theorem **exists_lift** {R R₁ R₂ : Type} [CommRing R] [CommRing R₁] [CommRing R₂] {f : R →+* R₁} {g : R₁ →+* R₂} {h : R →+* R₂} (e : g.comp f = h) {G : Over (Sp R)} {G₁ : Over (Sp R₁)} {G₂ : Over (Sp R₂)} [GrpObj G] [GrpObj G₁] [GrpObj G₂] {φ : G₁.left ⟶ G.left} {χ : G₂.left ⟶ G.left} (h₁ : IsGrpPullback G₁ G (spMap f) φ) (h₂ : IsGrpPullback G₂ G (spMap h) χ) : ∃ ψ, ψ ≫ φ = χ ∧ IsGrpPullback G₂ G₁ (spMap g) ψ

theorem **exists_cover_lieLat** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P), ∃ (n : ℕ) (r : Fin n → 𝒪 K) (M : Fin n → Matrix (Fin F.P₀.g) (Fin F.P₀.g) K), Ideal.span (Set.range r) = ⊤ ∧ (∀ i, r i ≠ 0 ∧ (M i).det ≠ 0) ∧ (∀ c, c ∈ F.lieLat ↔ ∀ i, ∃ (k : ℕ) (a : Fin F.P₀.g → 𝒪 K), r i ^ k • c = ∑ j, a j • M i j) ∧ ∀ i (R : Type) [CommRing R] [Algebra (𝒪 K) R] [Algebra R K] [IsScalarTower (𝒪 K) R K] (G : Over (Sp R)) [GrpObj G] (φ : G.left ⟶ P.A.left) (ψ : F.P₀.A.left ⟶ G.left), IsUnit (algebraMap (𝒪 K) R (r i)) → IsGrpPullback G P.A (spMap (algebraMap (𝒪 K) R)) φ → IsGrpPullback F.P₀.A G (spMap (algebraMap R K)) ψ → ψ ≫ φ = F.φ → ∃ e : Fin F.P₀.g → (epsR R ⟶ G), IsLieBasis e ∧ ∀ j, ∃ w, IsVecOver (algebraMap R K) ψ (e j) w ∧ IsLieComb (M i j) F.v w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeLocalBasisDescriptionStatement

theorem **_root_.GlobalTheory.T.lie_lattice_local_basis_description** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P), ∃ (n : ℕ) (r : Fin n → 𝒪 K) (M : Fin n → Matrix (Fin F.P₀.g) (Fin F.P₀.g) K), Ideal.span (Set.range r) = ⊤ ∧ (∀ i, r i ≠ 0 ∧ (M i).det ≠ 0) ∧ (∀ c, c ∈ F.lieLat ↔ ∀ i, ∃ (k : ℕ) (a : Fin F.P₀.g → 𝒪 K), r i ^ k • c = ∑ j, a j • M i j) ∧ ∀ i (R : Type) [CommRing R] [Algebra (𝒪 K) R] [Algebra R K] [IsScalarTower (𝒪 K) R K] (G : Over (Sp R)) [GrpObj G] (φ : G.left ⟶ P.A.left) (ψ : F.P₀.A.left ⟶ G.left), IsUnit (algebraMap (𝒪 K) R (r i)) → IsGrpPullback G P.A (spMap (algebraMap (𝒪 K) R)) φ → IsGrpPullback F.P₀.A G (spMap (algebraMap R K)) ψ → ψ ≫ φ = F.φ → ∃ e : Fin F.P₀.g → (epsR R ⟶ G), IsLieBasis e ∧ ∀ j, ∃ w, IsVecOver (algebraMap R K) ψ (e j) w ∧ IsLieComb (M i j) F.v w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeLocalizationModelLieAlgebra

theorem **coord_eq_sum** {R₀ R K : Type} [CommRing R₀] [CommRing R] [CommRing K] {f : R₀ →+* R} {g : R →+* K} {A₀ : Over (Sp R₀)} {G : Over (Sp R)} {A : Over (Sp K)} [GrpObj A₀] [GrpObj G] [GrpObj A] [IsCommMonObj A] {φ : G.left ⟶ A₀.left} {ψ : A.left ⟶ G.left} (hφ : IsGrpPullback G A₀ (spMap f) φ) (hψ : IsGrpPullback A G (spMap g) ψ) {n m : ℕ} {v : Fin n → (epsR K ⟶ A)} (hv : IsLieBasis v) {e : Fin m → (epsR R ⟶ G)} (he : IsLieBasis e) {M : Fin m → Fin n → K} (hM : ∀ i, ∃ w, IsVecOver g ψ (e i) w ∧ IsLieComb (M i) v w) {u : epsR R₀ ⟶ A₀} (hu : IsTangent u) {w : epsR K ⟶ A} (hw : IsVecOver (g.comp f) (ψ ≫ φ) u w) {c : Fin n → K} (hc : IsLieComb c v w) : ∃ a : Fin m → R, c = ∑ i, g (a i) • M i

theorem **exists_denom** {O R R' K : Type} [CommRing O] [CommRing R] [CommRing R'] [CommRing K] [Algebra O R] [Algebra R R'] [Algebra O K] (N : Submonoid O) [IsLocalization N R] (r : O) [IsLocalization.Away (algebraMap O R r) R'] (g : R' →+* K) (hg : ∀ x, g (algebraMap R R' (algebraMap O R x)) = algebraMap O K x) {m : ℕ} (b : Fin m → R') : ∃ s ∈ N, ∃ (k : ℕ) (a : Fin m → O), ∀ i, (r ^ k * s) • g (b i) = algebraMap O K (a i)

theorem **lieLat_localization** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (R : Type) [CommRing R] [Algebra (𝒪 K) R] [Algebra R K] [IsScalarTower (𝒪 K) R K] (N : Submonoid (𝒪 K)) [IsLocalization N R] (G : Over (Sp R)) [GrpObj G] (φ : G.left ⟶ P.A.left) (ψ : F.P₀.A.left ⟶ G.left) (e : Fin F.P₀.g → (epsR R ⟶ G)) (e₀ : Fin F.P₀.g → (epsR K ⟶ F.P₀.A)) (M : Matrix (Fin F.P₀.g) (Fin F.P₀.g) K), IsGrpPullback G P.A (spMap (algebraMap (𝒪 K) R)) φ → IsGrpPullback F.P₀.A G (spMap (algebraMap R K)) ψ → ψ ≫ φ = F.φ → IsLieBasis e → (∀ i, IsVecOver (algebraMap R K) ψ (e i) (e₀ i) ∧ IsLieComb (M i) F.v (e₀ i)) → (∀ i, ∃ s ∈ N, s • M i ∈ F.lieLat) ∧ ∀ c ∈ F.lieLat, ∃ a : Fin F.P₀.g → R, c = ∑ i, algebraMap R K (a i) • M i

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeLocalizationModelLieAlgebraStatement

theorem **_root_.GlobalTheory.T.lieLattice_localization_modelLieAlgebra** : ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (R : Type) [CommRing R] [Algebra (𝒪 K) R] [Algebra R K] [IsScalarTower (𝒪 K) R K] (N : Submonoid (𝒪 K)) [IsLocalization N R] (G : Over (Sp R)) [GrpObj G] (φ : G.left ⟶ P.A.left) (ψ : F.P₀.A.left ⟶ G.left) (e : Fin F.P₀.g → (epsR R ⟶ G)) (e₀ : Fin F.P₀.g → (epsR K ⟶ F.P₀.A)) (M : Matrix (Fin F.P₀.g) (Fin F.P₀.g) K), IsGrpPullback G P.A (spMap (algebraMap (𝒪 K) R)) φ → IsGrpPullback F.P₀.A G (spMap (algebraMap R K)) ψ → ψ ≫ φ = F.φ → IsLieBasis e → (∀ i, IsVecOver (algebraMap R K) ψ (e i) (e₀ i) ∧ IsLieComb (M i) F.v (e₀ i)) → (∀ i, ∃ s ∈ N, s • M i ∈ F.lieLat) ∧ ∀ c ∈ F.lieLat, ∃ a : Fin F.P₀.g → R, c = ∑ i, algebraMap R K (a i) • M i

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeOfIsLocalModelAtPrime

theorem **exists_ringHom** : ∃ f : 𝒪 K →+* R, ∀ x, algebraMap R K (f x) = algebraMap (𝒪 K) K x

theorem **isUnit_of_valuation_eq_one** (𝔭 : HeightOneSpectrum (𝒪 K)) (hR : ∀ r : R, 𝔭.valuation K (algebraMap R K r) ≤ 1) {r : R} (hr : 𝔭.valuation K (algebraMap R K r) = 1) : IsUnit r

theorem **isLocalization** [Algebra (𝒪 K) R] [IsScalarTower (𝒪 K) R K] (𝔭 : HeightOneSpectrum (𝒪 K)) (hR : ∀ r : R, 𝔭.valuation K (algebraMap R K r) ≤ 1) : IsLocalization 𝔭.asIdeal.primeCompl R

theorem **exists_iso_of_models** (h : HomExtDvr) (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'] (hG : IsSemiAbelian G) (hG' : IsSemiAbelian G') (H : Over (Sp K)) [GrpObj H] {ψ : H.left ⟶ G.left} {ψ' : H.left ⟶ G'.left} (hψ : IsGrpPullback H G (spMap (algebraMap T K)) ψ) (hψ' : IsGrpPullback H G' (spMap (algebraMap T K)) ψ') : ∃ e : G ≅ G', IsMonHom e.hom ∧ ψ ≫ e.hom.left = ψ'

theorem **eq_dualMapR** {R R' : Type} [CommRing R] [CommRing R'] (f : R →+* R') (F : DualNumber R →+* DualNumber R') (h₁ : ∀ a, F (algebraMap R _ a) = algebraMap R' _ (f a)) (h₂ : F DualNumber.eps = DualNumber.eps) : F = Chart.dualMapR f

theorem **lieLat_of_isLocalModel** : HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (hg rig hr th hth) (𝔭 : HeightOneSpectrum (𝒪 K)) (a : K) (M : Matrix (Fin F.P₀.g) (Fin F.P₀.g) K), KeyPow.IsLocalModel (F.key hg rig hr th hth) 𝔭 a M → (∀ i, ∃ s : 𝒪 K, s ∉ 𝔭.asIdeal ∧ s • M i ∈ F.lieLat) ∧ ∀ c ∈ F.lieLat, ∃ s : 𝒪 K, s ∉ 𝔭.asIdeal ∧ s • c ∈ Submodule.span (𝒪 K) (Set.range fun i => M i)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLatticeOfIsLocalModelAtPrimeStatement

theorem **_root_.GlobalTheory.T.lieLattice_of_isLocalModel_at_prime** : HomExtDvr → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (hg rig hr th hth) (𝔭 : HeightOneSpectrum (𝒪 K)) (a : K) (M : Matrix (Fin F.P₀.g) (Fin F.P₀.g) K), KeyPow.IsLocalModel (F.key hg rig hr th hth) 𝔭 a M → (∀ i, ∃ s : 𝒪 K, s ∉ 𝔭.asIdeal ∧ s • M i ∈ F.lieLat) ∧ ∀ c ∈ F.lieLat, ∃ s : 𝒪 K, s ∉ 𝔭.asIdeal ∧ s • c ∈ Submodule.span (𝒪 K) (Set.range fun i => M i)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLetterIntegralInIntegralEigenbasis

theorem **prod_pull** {S' S₀ : Scheme.{0}} {G' : Over S'} [GrpObj G'] {G : Over S₀} [GrpObj G] {f : S' ⟶ S₀} {φ : G'.left ⟶ G.left} (h : IsGrpPullback G' G f φ) (T' : Over S') (T : Over S₀) (g : T.left ⟶ T'.left) (hg : g ≫ T'.hom ≫ f = T.hom) {m : ℕ} (a : Fin m → (T' ⟶ G')) (a₀ : Fin m → (T ⟶ G)) (ha : ∀ j, (a₀ j).left = g ≫ (a j).left ≫ φ) : ((List.ofFn a₀).prod).left = g ≫ ((List.ofFn a).prod).left ≫ φ

theorem **comb_match** {V K F : Type} [CommRing V] [CommRing K] [CommRing F] {X : Scheme.{0}} {B : Over X} [GrpObj B] {G₁ : Over (Sp V)} [GrpObj G₁] {G₂ : Over (Sp K)} [GrpObj G₂] {f₁ : Sp V ⟶ X} {f₂ : Sp K ⟶ X} {φ₁ : G₁.left ⟶ B.left} {φ₂ : G₂.left ⟶ B.left} (h₁ : IsGrpPullback G₁ B f₁ φ₁) (h₂ : IsGrpPullback G₂ B f₂ φ₂) (α : V →+* F) (β : K →+* F) (hf : spMap α ≫ f₁ = spMap β ≫ f₂) {m : ℕ} {v₁ : Fin m → (epsR V ⟶ G₁)} {v₂ : Fin m → (epsR K ⟶ G₂)} (hv : ∀ i, spMap (dualMap α) ≫ (v₁ i).left ≫ φ₁ = spMap (dualMap β) ≫ (v₂ i).left ≫ φ₂) {c₁ : Fin m → V} {c₂ : Fin m → K} (hc : ∀ i, α (c₁ i) = β (c₂ i)) {w₁ : epsR V ⟶ G₁} {w₂ : epsR K ⟶ G₂} (hw₁ : IsLieComb c₁ v₁ w₁) (hw₂ : IsLieComb c₂ v₂ w₂) : spMap (dualMap α) ≫ w₁.left ≫ φ₁ = spMap (dualMap β) ≫ w₂.left ≫ φ₂

theorem **comb_eq_scale** {K : Type} [Field K] {G : Over (Sp K)} [GrpObj G] [IsCommMonObj G] {n : ℕ} {v wK : Fin n → (epsR K ⟶ G)} (hv : ∀ i, IsTangent (v i)) {A : Matrix (Fin n) (Fin n) K} (hA : A.det ≠ 0) (hw : ∀ i, IsLieComb (A i) v (wK i)) {U : Fin n → K} {w : epsR K ⟶ G} (hU : IsLieComb U v w) (cl : K) {s : epsR K ⟶ G} (hs : IsLieComb (Matrix.vecMul (cl • U) A⁻¹) wK s) : s.left = spMap (epsMul cl) ≫ w.left

theorem **epsMul_comm** {R : Type} [CommRing R] (a b : R) : spMap (epsMul a) ≫ spMap (epsMul b) = spMap (epsMul b) ≫ spMap (epsMul a)

theorem **dualMap_inj** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (hf : Function.Injective f) : Function.Injective (dualMap f)

theorem **coord_eq** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (b : E →+* ℂ) (u' : Fin (Dw.mult b) → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)) (g : Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) S) (hg : IsUnit g.det) (hu : ∀ i, IsCombAt y.x (y.u b) (g i) (u' i)) (r : S) (j : Fin (Dw.mult b)) (a : Fin (Dw.mult b) → S) (ha : IsCombAt y.x u' a (spMap (epsMul r) ≫ y.u b j)) (k : Fin (Dw.mult b)) : r * g⁻¹ j k = a k

theorem **isEigV_of_generic** {V K : Type} [CommRing V] [Field K] [Algebra V K] [IsFractionRing V K] [Algebra c.F K] {x : Sp K ⟶ Xo.X} {G : SAB H (Sp V)} [IsSeparated G.G.hom] {t : SAB H (Sp K)} (m₁ : SABMap t Xo.univ.t x) (m₂ : SABMap t G (spMap (algebraMap V K))) (b : E →+* ℂ) (w : epsOver V V ⟶ G.G) (w' : Sp (DualNumber K) ⟶ t.G.left) (hw : w' ≫ m₂.φ = spMap (dualMap (algebraMap V K)) ≫ w.left) (ho : w' ≫ t.G.hom = spMap (algebraMap K (DualNumber K))) (he : IsEigAt c b (w' ≫ m₁.φ)) : IsEigV c (algebraMap c.F K) G b w

theorem **letter_integral_in_eigenbasis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (p : HeightOneSpectrum (𝒪 z.K)) (T : CDVR) (iK : z.K →+* T.Fr) (M : LocMod z T iK), (∀ y : z.K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation z.K y ≤ 1) → letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLetterIntegralInIntegralEigenbasisStatement

theorem **_root_.GlobalTheory.T.lieLetter_integral_in_integral_eigenbasis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (p : HeightOneSpectrum (𝒪 z.K)) (T : CDVR) (iK : z.K →+* T.Fr) (M : LocMod z T iK), (∀ y : z.K, iK y ∈ Set.range (algebraMap T.V T.Fr) ↔ p.valuation z.K y ≤ 1) → letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLetterIntegralOfLocalGenerator

theorem **isDiffOf_mem_lieLat** {K : Type} [Field K] [NumberField K] {P P' : Out.PolAV (Sp (𝒪 K))} {F : GenFib P} {F' : GenFib P'} {f : P.A ⟶ P'.A} {D : (Fin F.P₀.g → K) →ₗ[K] (Fin F'.P₀.g → K)} (hD : F.IsDiffOf F' f D) {c : Fin F.P₀.g → K} (hc : c ∈ F.lieLat) : D c ∈ F'.lieLat

theorem **smul_image_mem_lieLat** {K : Type} [Field K] [NumberField K] {P P' : Out.PolAV (Sp (𝒪 K))} (F : GenFib P) (F' : GenFib P') (f : P.A ⟶ P'.A) (hf : IsMonHom f) (i : Fin F.P₀.g) (w : epsR K ⟶ F'.P₀.A) (U : Fin F'.P₀.g → K) (hw : w.left ≫ F'.φ = (F.v i).left ≫ F.φ ≫ f.left) (hU : IsLieComb U F'.v w) (c : K) (hc : (Pi.single i c : Fin F.P₀.g → K) ∈ F.lieLat) : c • U ∈ F'.lieLat

theorem **image_integral_of_isLocGen** {K : Type} [Field K] [NumberField K] {A B : Out.PolAV (Sp (𝒪 K))} {x y : KeyIn K} {φ : x.P.A.left ⟶ A.A.left} {ψ : y.P.A.left ⟶ B.A.left} (hx : IsFibreOf A x (gmap K) φ) (hy : IsFibreOf B y (gmap K) ψ) (f : A.A ⟶ B.A) (hf : IsMonHom f) {i : Fin x.P.g} {ℓ : MLine K} (hℓ : IsVecLine A x φ i ℓ) {w : epsR K ⟶ y.P.A} {U : Fin y.P.g → K} (hw : w.left ≫ ψ = (x.v i).left ≫ φ ≫ f.left) (hU : IsLieComb U y.v w) {p : HeightOneSpectrum (𝒪 K)} {M : Matrix (Fin y.P.g) (Fin y.P.g) K} (hM : IsModelAt B y ψ p M) {cl : K} (hcl : IsLocGen p cl ℓ.I) (m) : p.valuation K (Matrix.vecMul (cl • U) M⁻¹ m) ≤ 1

theorem **letter_integral_of_isLocGen** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' f : ℕ) (o : LamObj c nΨ n') (La : ℝ), LamObjGood o f La → ∀ L : LamLetters o, L.IsGood → ∀ (p : HeightOneSpectrum (𝒪 o.K₁)) (M : Matrix (Fin o.xB.P.g) (Fin o.xB.P.g) o.K₁), IsModelAt o.B o.xB o.φB p M → ∀ (k : Fin o.xB.P.g) (cl : o.K₁), IsLocGen p cl (L.ℓ k).I → ∀ m, p.valuation o.K₁ (Matrix.vecMul (cl • L.U k) M⁻¹ m) ≤ 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieLetterIntegralOfLocalGeneratorStatement

theorem **_root_.GlobalTheory.T.lieLetter_integral_of_local_generator** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' f : ℕ) (o : LamObj c nΨ n') (La : ℝ), LamObjGood o f La → ∀ L : LamLetters o, L.IsGood → ∀ (p : HeightOneSpectrum (𝒪 o.K₁)) (M : Matrix (Fin o.xB.P.g) (Fin o.xB.P.g) o.K₁), IsModelAt o.B o.xB o.φB p M → ∀ (k : Fin o.xB.P.g) (cl : o.K₁), IsLocGen p cl (L.ℓ k).I → ∀ m, p.valuation o.K₁ (Matrix.vecMul (cl • L.U k) M⁻¹ m) ≤ 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieMultOfEigenbasis

theorem **repr_eq_zero** {k O V J : Type} [Field k] [CommRing O] [AddCommGroup V] [Module k V] [Module O V] [SMulCommClass O k V] (b : Module.Basis J k V) (χ : J → (O →+* k)) (hb : ∀ i a, a • b i = χ i a • b i) (ψ : O →+* k) (w : V) (hw : ∀ a : O, a • w = ψ a • w) (j : J) (hj : χ j ≠ ψ) : b.repr w j = 0

theorem **lieMult_of_eigenbasis** {k O : Type} [Field k] [CommRing O] {G : Over (Sp k)} [GrpObj G] [IsCommMonObj G] (a : OAct O G) {n : ℕ} {v : Fin n → (epsR k ⟶ G)} (hv : IsLieBasis v) (χ : Fin n → (O →+* k)) (hχ : Function.Injective χ) (h : ∀ x i, IsLieComb (Pi.single i (χ i x)) v (v i ≫ a.act x)) : (∀ i, LieMult a (𝟙 _) (χ i) 1) ∧ ∀ ψ, (∀ i, χ i ≠ ψ) → LieMult a (𝟙 _) ψ 0

theorem **exists_label** {E K k : Type} [Field E] [NumberField E] [Field K] [CharZero K] [Field k] [IsAlgClosed k] (j : K →+* k) (lab : (E →+* ℂ) → (E →+* K)) (hl : Function.Injective lab) (τ : 𝒪 E →+* k) : ∃ t, τ = j.comp ((lab t).comp (algebraMap (𝒪 E) E))

theorem **hasType_of_isLieType** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (oa : A.OAct (𝒪 E)) (act : OAct (𝒪 E) A.A) (S : Set (E →+* ℂ)) (τ : Fin x.P.g ≃ S) (lab : (E →+* ℂ) → (E →+* K)) (ρ : 𝒪 K), act.act = oa.act → Function.Injective lab → (∀ t, t ∈ S ↔ lab t (H.rt : E) = (ρ : K)) → IsLieType A oa x φ τ lab → HasType H act ρ 1 0

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieMultTensorOfFinrank

theorem **lieMult_tensor_of_finrank** {O O' M R k : Type} [CommRing O] [CommRing O'] [Algebra O O'] [AddCommGroup M] [Module O M] [Module O' M] [IsScalarTower O O' M] [Module.Finite O M] [Module.Projective O M] [CommRing R] [Field k] [Algebra O k] {G A : Over (Sp R)} [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (hs : Smooth G.hom) (v : epsR R ⟶ G) (hv : IsLieBasis ![v]) (a : OAct O G) (κ : O →+* R) (b₀ : OAct O' A) (τ : IsTensor A (comapAct b₀ (algebraMap O O')) G a M) (hκ : ∀ c, IsLieComb ![κ c] ![v] (v ≫ a.act c)) (hlin : ∀ (x : O') m, τ.tmul (x • m) = τ.tmul m ≫ b₀.act x) (β : R →+* k) (hβ : ∀ c, algebraMap O k c = β (κ c)) (χ : O' →+* k) (n : ℕ) (hn : Module.finrank k (EigenSplit.eigenSub (rightAction k O O' M) χ) = n) : LieMult b₀ (spMap β) χ n

theorem **hasType_of_isTen** : Out.PolBaseChangeSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [CharZero R] (ω : R) (C : Ell H R ω) (D : Datum H) (h : D.La → D.La → OK H) (T : PAT H (Sp R)), Nonempty (IsTen C D h T) → HasType H T.t.act (2 * ω - 1) (D.p + D.q) 0

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LieTensorLine

private theorem **isLieComb_one_iff** {R : Type} [CommRing R] {G : Over (Sp R)} [GrpObj G] (r : R) (v w : epsR R ⟶ G) : IsLieComb ![r] ![v] w ↔ w.left = spMap (epsScaleR r) ≫ v.left

theorem **lie_tensor_line** : ∀ (R O : Type) [CommRing R] [CommRing O] [Algebra O R] (G A : Over (Sp R)) [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (M : Type) [AddCommGroup M] [Module O M] (t : Serre.IsTensor A b G a M) (v : epsR R ⟶ G), Module.Finite O M → Module.Projective O M → IsLieBasis ![v] → (∀ c, IsLieComb ![algebraMap O R c] ![v] (v ≫ a.act c)) → ∃ e : TensorProduct O R M → (epsR R ⟶ A), Function.Injective e ∧ (∀ u, IsTangent u ↔ u ∈ Set.range e) ∧ (∀ z z', e (z + z') = e z * e z') ∧ (∀ (r : R) z, IsLieComb ![r] ![e z] (e (r • z))) ∧ ∀ m, e (TensorProduct.tmul O 1 m) = v ≫ t.tmul m

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LineBundleIsoDescendsFieldToDedekindDomain

theorem **exists_gen_pull_div_iff** {X Y : Scheme.{0}} (f : Y ⟶ X) {N N' : LB X} (hN : N.IsLB) (hN' : N'.IsLB) : (∃ t : LB.Gen ((LB.div N N').pull f), t.IsGen) ↔ LBIso (N.pull f) (N'.pull f)

theorem **lbIso_of_field** : ∀ (R k : Type) [CommRing R] [IsDedekindDomain R] [Field k] (β : R →+* k), Function.Injective β → ∀ (A : Over (Sp R)) [GrpObj A], IsAbelianScheme A → ∀ (X' : Scheme.{0}) (π : X' ⟶ A.left) (p : X' ⟶ Sp k), IsPullback π p A.hom (spMap β) → ∀ N N' : LB A.left, N.IsLB → N'.IsLB → LBIso (N.pull (eSec A)) (N'.pull (eSec A)) → LBIso (N.pull π) (N'.pull π) → LBIso N N'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LineBundleIsoDescendsFieldToDedekindDomainStatement

theorem **_root_.GlobalTheory.T.lineBundleIso_descends_field_to_dedekindDomain** : ∀ (R k : Type) [CommRing R] [IsDedekindDomain R] [Field k] (β : R →+* k), Function.Injective β → ∀ (A : Over (Sp R)) [GrpObj A], IsAbelianScheme A → ∀ (X' : Scheme.{0}) (π : X' ⟶ A.left) (p : X' ⟶ Sp k), IsPullback π p A.hom (spMap β) → ∀ N N' : LB A.left, N.IsLB → N'.IsLB → LBIso (N.pull (eSec A)) (N'.pull (eSec A)) → LBIso (N.pull π) (N'.pull π) → LBIso N N'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LinearIndependentOfIsLieBasis

theorem **ofGram_single** (G : (K →+* ℂ) → Matrix (Fin g) (Fin g) ℂ) (hG : ∀ s, (G s).PosDef) (s : K →+* ℂ) (i j : Fin g) : (ofGram G hG).form s (Pi.single i 1) (Pi.single j 1) = G s i j

theorem **form_sum_right** (h : HermFam K (Fin g → K)) (s : K →+* ℂ) (x y : Fin g → K) : h.form s x y = ∑ j, s (y j) * h.form s x (Pi.single j 1)

theorem **ext_single** {h h' : HermFam K (Fin g → K)} (e : ∀ s i j, h.form s (Pi.single i 1) (Pi.single j 1) = h'.form s (Pi.single i 1) (Pi.single j 1)) : h = h'

theorem **posDef_gram** {n : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (hp : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (z ⬝ᵥ Hm.mulVec (star z)).re) {W : Fin n → Fin g → ℂ} (hW : LinearIndependent ℂ W) : (Matrix.of fun i j => (Real.pi : ℂ) * hermForm Hm (W j) (W i)).PosDef

theorem **linearIndependent_of_isLieBasis** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g n : ℕ} (u : HolUnif A L g) (hu : ∀ w, ∃! v, IsTangent v ∧ u.IsDiff w v) {v : Fin n → (epsR ℂ ⟶ A)} (hv : IsLieBasis v) {W : Fin n → Fin g → ℂ} (hW : ∀ i, u.IsDiff (W i) (v i)) : LinearIndependent ℂ W

theorem **hermForm_eq** (hU : ThetaUnifAV) {R : Type} [CommRing R] {P : Out.PolAV (Sp R)} {s : R →+* ℂ} {Q Q' : Out.PolAV (Sp ℂ)} {ψ : Q.A.left ⟶ P.A.left} {ψ' : Q'.A.left ⟶ P.A.left} (h : IsPolFibre s P Q ψ) (h' : IsPolFibre s P Q' ψ') {g g' : ℕ} {u : HolUnif Q.A Q.L g} {u' : HolUnif Q'.A Q'.L g'} {Hm : Matrix (Fin g) (Fin g) ℂ} {Hm' : Matrix (Fin g') (Fin g') ℂ} {χ : u.Λ → ℂ} {χ' : u'.Λ → ℂ} (hu : u.IsAH Hm χ) (hu' : u'.IsAH Hm' χ') {I : Type} {v : I → (epsR R ⟶ P.A)} (hv : ∀ i, IsTangent (v i)) {w : I → (epsR ℂ ⟶ Q.A)} {w' : I → (epsR ℂ ⟶ Q'.A)} {W : I → Fin g → ℂ} {W' : I → Fin g' → ℂ} (hw : ∀ i, IsVecOver s ψ (v i) (w i) ∧ u.IsDiff (W i) (w i)) (hw' : ∀ i, IsVecOver s ψ' (v i) (w' i) ∧ u'.IsDiff (W' i) (w' i)) (i j : I) : hermForm Hm (W i) (W j) = hermForm Hm' (W' i) (W' j)

theorem **exists_unique_isHerm** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝓞 K))) (F : GenFib P), (∃ h : HermFam K (Fin F.P₀.g → K), F.IsHerm h) ∧ ∀ h h' : HermFam K (Fin F.P₀.g → K), F.IsHerm h → F.IsHerm h' → h = h'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LocZeroTranslDiffComp

theorem **pull_eq_zero_of_range** {W : Y.Opens} {c : Pic Y} (H : pull W.ι c = 0) (h : Y' ⟶ Y) (hW : ∀ y, h y ∈ W) : pull h c = 0

theorem **LocZero.sub** {π : Y ⟶ B} {c d : Pic Y} (H : LocZero π c) (H' : LocZero π d) : LocZero π (c - d)

theorem **LocZero.pull_over** {π : Y ⟶ B} {c : Pic Y} (H : LocZero π c) {π' : Y' ⟶ B} (h : Y' ⟶ Y) (sq : h ≫ π = π') : LocZero π' (pull h c)

theorem **locZero_base** (π : Y ⟶ B) (c : Pic B) : LocZero π (pull π c)

theorem **locZero_iff** (π : Y ⟶ B) {N : LB Y} (hN : N.IsLB) : (∀ p : B, ∃ V : B.Opens, p ∈ V ∧ ∃ c : LB.Gen (N.pull (π ⁻¹ᵁ V).ι), c.IsGen) ↔ LocZero π (cls N hN)

theorem **cls_translDiff** {L : LB A.left} (hL : L.IsLB) {T : Over S} (x : T ⟶ A) : cls (Out.translDiff A L x) (LB.isLB_linComb hL _ _) = pull ((fst T A ≫ x) * snd T A).left (cls L hL) - pull (snd T A).left (cls L hL)

theorem **locZero_translDiff_comp** {N : LB A.left} {L : LB G.left} (hN : N.IsLB) (hL : L.IsLB) (b : Pic H.left) (p : A ⟶ G) (q : A ⟶ H) [IsMonHom p] [IsMonHom q] (u : G ⟶ A) (hp : u ≫ p = 𝟙 G) (hq : u ≫ q = 1) (e : cls N hN = pull p.left (cls L hL) + pull q.left b) {T : Over S} (x : T ⟶ A) (hx : LocZero (fst T A).left (cls (Out.translDiff A N x) (LB.isLB_linComb hN _ _))) : LocZero (fst T G).left (cls (Out.translDiff G L (x ≫ p)) (LB.isLB_linComb hL _ _))

theorem **kTriv_prod** {N : LB (G ⊗ H).left} {L : LB G.left} {L' : LB H.left} (hN : N.IsLB) (hL : L.IsLB) (hL' : L'.IsLB) (e : cls N hN = pull (fst G H).left (cls L hL) + pull (snd G H).left (cls L' hL')) (hK : Out.KTriv G L) (hK' : Out.KTriv H L') : Out.KTriv (G ⊗ H) N

theorem **cls_cubeLB** {T : Over S} (x : Fin n → (T ⟶ G)) {L : LB G.left} (hL : L.IsLB) : cls (cubeLB G x L) (LB.isLB_linComb hL _ _) = cubeCls x (cls L hL)

theorem **cubeCls_add** {T : Over S} (x : Fin n → (T ⟶ G)) (c d : Pic G.left) : cubeCls x (c + d) = cubeCls x c + cubeCls x d

theorem **pull_cubeCls** {T T' : Over S} (ρ : T' ⟶ T) (x : Fin n → (T ⟶ G)) (c : Pic G.left) (y : Fin n → (T' ⟶ G)) (h : ∀ i, y i = ρ ≫ x i) : cubeCls y c = pull ρ.left (cubeCls x c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LocalLowerBoundCmPointFramePolynomialEveryPrime

theorem **exists_isLocGen** {K : Type} [Field K] [NumberField K] (p : HeightOneSpectrum (𝓞 K)) (I : FractionalIdeal (nonZeroDivisors (𝓞 K)) K) : ∃ c : K, IsLocGen p c I

theorem **map_evalFr** {B : Type} {m : B → ℕ} {R S : Type} [CommRing R] [CommRing S] (κ : R →+* S) (P : MvPolynomial (FrVar m) R) (Y : ∀ b, Matrix (Fin (m b)) (Fin (m b)) R) : κ (evalFr P Y) = evalFr (MvPolynomial.map κ P) (fun b => (Y b).map κ)

theorem **secBC_one_of_frMap** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] {φ : S →ₐ[c.F] T} {y : FrPt Xo c S} {y' : FrPt Xo c T} (h : FrMap φ y y') : SecBC φ.toRingHom 1 y.x y'.x y.s y'.s

theorem **evR_scale** {S : Type} [Field S] [Algebra c.F S] {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {F : Rul Xo c} (hF : F ∈ sec Xo c a n) (s : Sˣ) {y y' : FrPt Xo c S} (h : FrChg (fun _ => 1) (s : S) y y') : evR Xo c F y' = (s : S) ^ n * evR Xo c F y

theorem **integral_of_eq** (T : CDVR) (io : c.F →+* T.Fr) (y : letI

theorem **secInt_baseChange** (T T' : CDVR) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr) (hψ : (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr)) {x x₁ : Sp T.Fr ⟶ Xo.X} {x' x₁' : Sp T'.Fr ⟶ Xo.X} {G : PAT H (Sp T.V)} {G' : PAT H (Sp T'.V)} {t : SAB H (Sp T.Fr)} {t' : SAB H (Sp T'.Fr)} (m1 : SABMap t Xo.univ.t x) (m2 : SABMap t G.t (spMap (algebraMap T.V T.Fr))) (m1' : SABMap t' Xo.univ.t x') (m2' : SABMap t' G'.t (spMap (algebraMap T'.V T'.Fr))) (n : SABMap G'.t G.t (spMap ψ)) (n' : SABMap t' t (spMap i)) (hx : x' = spMap i ≫ x) (h1 : m1'.φ = n'.φ ≫ m1.φ) (h2 : m2'.φ ≫ n.φ = n'.φ ≫ m2.φ) (e : x₁ = x) (e' : x₁' = x') (s : LB.Sec (PPx x₁)) (s' : LB.Sec (PPx x₁')) (hs : SecBC i 1 x₁ x₁' s s') (h : SecInt T m1 m2 s) : SecInt T' m1' m2' s'

theorem **integral_baseChange_chg** [StarRing (𝓞 E)] (z : CMPt Xo c) (T T' : CDVR) (iK : z.K →+* T.Fr) (M : LocMod z T iK) (ψ : T.V →+* T'.V) (i : T.Fr →+* T'.Fr) (hψ : (algebraMap T'.V T'.Fr).comp ψ = i.comp (algebraMap T.V T.Fr)) (io : c.F →+* T.Fr) : letI

theorem **evR_chg** {K S : Type} [Field K] [Algebra c.F K] [Field S] [Algebra c.F S] {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {F : Rul Xo c} (hF : F ∈ sec Xo c a n) {y₀ : FrPt Xo c K} {G : MvPolynomial (FrVar Dw.mult) K} (hG : IsFrPoly F y₀ G) (φ : K →ₐ[c.F] S) {y y' w : FrPt Xo c S} (hy : FrMap φ y₀ y) {g X Y : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) S} {s : S} (hs : s ≠ 0) (hY : ∀ b, IsUnit (Y b).det) (hXY : ∀ b, (Y b).transpose = X b * g b) (h : FrChg g s y y') (hw : FrChg X 1 y' w) : evR Xo c F w = s ^ n * evalFr (MvPolynomial.map φ.toRingHom G) Y

theorem **step5Loc_cmPt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝓞 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, (∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v F) → ∀ G : MvPolynomial (FrVar Dw.mult) z.K, IsFrPoly F z.y G → ∀ p, Step5Loc z.e z.Tl.dual z.LamL z.ℓ z.U G n p

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LocalLowerBoundCmPointFramePolynomialEveryPrimeStatement

theorem **_root_.GlobalTheory.T.localLowerBound_cmPoint_framePolynomial_every_prime** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝓞 E)] (z : CMPt Xo c) (f : ℕ) (La : ℝ), z.Good f La → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ F ∈ sec Xo c a n, (∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v F) → ∀ G : MvPolynomial (FrVar Dw.mult) z.K, IsFrPoly F z.y G → ∀ p, Step5Loc z.e z.Tl.dual z.LamL z.ℓ z.U G n p

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LocallyQuasiFiniteOfQuasiFiniteAt

theorem **eq_genericPoint** {K : Type} [Field K] {X Y : Scheme.{0}} [IsIntegral X] [IsIntegral Y] (p : Y ⟶ Spec (CommRingCat.of K)) (f : X ⟶ Y) (g : ℕ) (hX : SmoothOfRelativeDimension g (f ≫ p)) (hY : SmoothOfRelativeDimension g p) (x : X) (hx : f x = genericPoint Y) : x = genericPoint X

theorem **quasiFiniteAt_genericPoint** {K : Type} [Field K] {X Y : Scheme.{0}} [IsIntegral X] [IsIntegral Y] (p : Y ⟶ Spec (CommRingCat.of K)) (f : X ⟶ Y) [LocallyOfFiniteType f] [Surjective f] (g : ℕ) (hX : SmoothOfRelativeDimension g (f ≫ p)) (hY : SmoothOfRelativeDimension g p) : f.QuasiFiniteAt (genericPoint X)

theorem **isIso_transl** {S : Scheme.{0}} {A : Over S} [GrpObj A] (x : 𝟙_ (Over S) ⟶ A) : IsIso ((toUnit A ≫ x) ∗ 𝟙 A).left

theorem **locallyQuasiFinite_of_quasiFiniteAt** {K : Type} [Field K] [IsAlgClosed K] {A B : Over (Spec (CommRingCat.of K))} [GrpObj A] [GrpObj B] [LocallyOfFiniteType A.hom] (f : A ⟶ B) [IsMonHom f] [LocallyOfFiniteType f.left] (x : A.left) (hx : f.left.QuasiFiniteAt x) : LocallyQuasiFinite f.left

theorem **isFinite_of_surjective** {K : Type} [Field K] {A B : Over (Sp K)} [GrpObj A] [GrpObj B] (hA : IsAbelianScheme A) (hB : IsAbelianScheme B) {g : ℕ} (dA : SmoothOfRelativeDimension g A.hom) (dB : SmoothOfRelativeDimension g B.hom) (f : A ⟶ B) [IsMonHom f] [Surjective f.left] : IsFinite f.left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.LogPiSubMeanLogLatticeNormDisc

theorem **IsIdealClassPoint.exists_lattice_eq** {K : Type} [Field K] [NumberField K] {σ : K →+* ℂ} {𝔟 : (Ideal (𝒪 K))⁰} {τ : ℍ} (hτ : IsIdealClassPoint σ (ClassGroup.mk0 𝔟) τ) {P : PeriodPair} (hP : ∀ z, z ∈ P.lattice ↔ ∃ a ∈ (𝔟 : Ideal (𝒪 K)), z = σ (a : K)) : ∃ a : ℂ, a ≠ 0 ∧ ∀ x, x ∈ P.lattice ↔ ∃ m n : ℤ, a ∗ (m ∗ (τ : ℂ) + n) = x

theorem **log_pi_sub_mean_log_latticeNormDisc** : ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ ComplexEmbedding.IsReal σ₀ → ∀ (𝔟 : ClassGroup (𝒪 K) → nonZeroDivisors (Ideal (𝒪 K))), ∀ (Lc : ClassGroup (𝒪 K) → PeriodPair), (∀ C, ClassGroup.mk0 (𝔟 C) = C) → (∀ C z, z ∈ (Lc C).lattice ↔ ∃ a ∈ (𝔟 C : Ideal (𝒪 K)), z = σ₀ (a : K)) → ∀ (σ : K →+* ℂ) (τ : ClassGroup (𝒪 K) → ℍ), (∀ C, IsIdealClassPoint σ C (τ C)) → Real.log Real.pi - (∑ C, Real.log (|((Lc C).ω₁ ∗ starRingEnd ℂ (Lc C).ω₂).im| ^ 6 ∗ ‖(Lc C).g₂ ^ 3 - 27 ∗ (Lc C).g₃ ^ 2‖)) / (6 ∗ Fintype.card (ClassGroup (𝒪 K))) = (∑ C, -Real.log (4 ∗ Real.pi ∗ (τ C).im ∗ ‖ModularForm.eta (τ C)‖ ^ 4)) / classNumber K

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MapRtEqNegIff

theorem **map_rt_eq_neg_iff** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) {R : Type} [CommRing R] [CharZero R] {ρ : R} (hρ : ρ ^ 2 = -(H.d : R)) {k : Type} [Field k] (β : R →+* k) (hβ : Function.Injective β) (τ : 𝒪 E →+* k) : τ H.rt = β (-ρ) ↔ τ H.rt ≠ β ρ

theorem **hasType_neg_iff** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [CharZero R] (G : Over (Sp R)) [GrpObj G] (act : OAct (𝒪 E) G) (ρ : R) (p q : ℕ), ρ ^ 2 = -(H.d : R) → (HasType H act ρ p q ↔ HasType H act (-ρ) q p)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogCovolumeEqMeanClassGroup

theorem **latticeArea_eq_abs_im_mul_conj** (L : PeriodPair) : latticeArea L = |(L.ω₁ ∗ starRingEnd ℂ L.ω₂).im|

theorem **latticeNormDisc_eq** (L : PeriodPair) : latticeNormDisc L = |(L.ω₁ ∗ starRingEnd ℂ L.ω₂).im| ^ 6 ∗ ‖L.g₂ ^ 3 - 27 ∗ L.g₃ ^ 2‖

theorem **latticeNormDisc_pos** (L : PeriodPair) : 0 < latticeNormDisc L

theorem **IsHomothetic.latticeNormDisc_eq** {L L' : PeriodPair} {c : ℂ} (h : IsHomothetic c L L') : latticeNormDisc L' = latticeNormDisc L

theorem **mem_conjPeriods_lattice** (L : PeriodPair) (z : ℂ) : z ∈ (conjPeriods L).lattice ↔ starRingEnd ℂ z ∈ L.lattice

theorem **G_conjPeriods** (L : PeriodPair) (n : ℕ) : (conjPeriods L).G n = starRingEnd ℂ (L.G n)

theorem **latticeNormDisc_conjPeriods** (L : PeriodPair) : latticeNormDisc (conjPeriods L) = latticeNormDisc L

theorem **latticeNormDisc_eq_eta** {L : PeriodPair} {a : ℂ} {τ : ℍ} (ha : a ≠ 0) (h : ∀ x, x ∈ L.lattice ↔ ∃ m n : ℤ, a ∗ (m ∗ (τ : ℂ) + n) = x) : latticeNormDisc L = (4 ∗ π ^ 2 ∗ τ.im ∗ ‖ModularForm.eta τ‖ ^ 4) ^ 6

theorem **log_pi_sub_log_latticeNormDisc** {L : PeriodPair} {a : ℂ} {τ : ℍ} (ha : a ≠ 0) (h : ∀ x, x ∈ L.lattice ↔ ∃ m n : ℤ, a ∗ (m ∗ (τ : ℂ) + n) = x) : Real.log π - Real.log (|(L.ω₁ ∗ starRingEnd ℂ L.ω₂).im| ^ 6 ∗ ‖L.g₂ ^ 3 - 27 ∗ L.g₃ ^ 2‖) / 6 = -Real.log (4 ∗ π ∗ τ.im ∗ ‖ModularForm.eta τ‖ ^ 4)

theorem **embedding_eq_or_eq_conjugate** (hK : Module.finrank ℚ K = 2) {σ₀ : K →+* ℂ} (h₀ : ¬ ComplexEmbedding.IsReal σ₀) (σ : K →+* ℂ) : σ = σ₀ ∨ σ = ComplexEmbedding.conjugate σ₀

theorem **exists_mul_mem_iff_of_mk0_eq** (𝔞 𝔟 𝔡 : (Ideal (𝒪 K))⁰) (h : ClassGroup.mk0 𝔡 ∗ ClassGroup.mk0 𝔟 = ClassGroup.mk0 𝔞) : ∃ u : K, u ≠ 0 ∧ ∀ t : K, (∀ b ∈ (𝔟 : Ideal (𝒪 K)), ∃ a ∈ (𝔞 : Ideal (𝒪 K)), u ∗ t ∗ (b : K) = a) ↔ ∃ d ∈ (𝔡 : Ideal (𝒪 K)), t = d

theorem **exists_isHomothetic_of_mk0_eq** (σ : K →+* ℂ) (𝔞 𝔟 𝔡 : (Ideal (𝒪 K))⁰) (h : ClassGroup.mk0 𝔡 ∗ ClassGroup.mk0 𝔟 = ClassGroup.mk0 𝔞) {L L' P : PeriodPair} {c β : ℂ} (hc : c ≠ 0) (hβ : β ≠ 0) (hL : ∀ z, z ∈ L.lattice ↔ ∃ a ∈ (𝔞 : Ideal (𝒪 K)), z = c ∗ σ (a : K)) (hL' : ∀ z, z ∈ L'.lattice ↔ ∀ y ∈ (𝔟 : Ideal (𝒪 K)), β⁻¹ ∗ σ (y : K) ∗ z ∈ L.lattice) (hP : ∀ z, z ∈ P.lattice ↔ ∃ d ∈ (𝔡 : Ideal (𝒪 K)), z = σ (d : K)) : ∃ κ : ℂ, IsHomothetic κ P L'

theorem **mean_log_covolume_eq_mean_classGroup** : ∀ (K M : Type) [Field K] [NumberField K] [Field M] [NumberField M] [Algebra K M], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ NumberField.ComplexEmbedding.IsReal σ₀ → ∀ (𝔟 : ClassGroup (𝒪 K) → nonZeroDivisors (Ideal (𝒪 K))) (Lc : ClassGroup (𝒪 K) → PeriodPair), (∀ C, ClassGroup.mk0 (𝔟 C) = C) → (∀ C z, z ∈ (Lc C).lattice ↔ ∃ a ∈ (𝔟 C : Ideal (𝒪 K)), z = σ₀ (a : K)) → ∀ (L₀ : (M →+* ℂ) → PeriodPair) (L : ClassGroup (𝒪 K) → (M →+* ℂ) → PeriodPair) (δ : ClassGroup (𝒪 K) → (𝒪 M)ˣ) (β : ClassGroup (𝒪 K) → M), (∀ s (a : 𝒪 K), ∀ z ∈ (L₀ s).lattice, s (algebraMap K M a) ∗ z ∈ (L₀ s).lattice) → (∀ C s, (L C s).g₂ ^ 3 - 27 ∗ (L C s).g₃ ^ 2 = s ((δ C : 𝒪 M) : M)) → (∀ C, β C ≠ 0) → (∀ C (x : K), (∀ b ∈ (𝔟 C : Ideal (𝒪 K)), IsIntegral ℤ (x ∗ (b : K))) → IsIntegral ℤ (β C ∗ algebraMap K M x) ∧ ∀ s, ∀ z ∈ (L₀ s).lattice, s (β C ∗ algebraMap K M x) ∗ z ∈

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogLieNormSqEqMeanLogEta

theorem **exists_periodPair_image_ideal** {K : Type} [Field K] [NumberField K] (hK : Module.finrank ℚ K = 2) {σ : K →+* ℂ} (hσ : ¬ ComplexEmbedding.IsReal σ) {I : Ideal (𝒪 K)} (hI : I ≠ ⊥) : ∃ L : PeriodPair, ∀ z, z ∈ L.lattice ↔ ∃ a ∈ I, z = σ (a : K)

theorem **mean_log_lieNormSq_eq_mean_log_eta** : ThetaUnifAV → Out.PolBaseChangeSt → (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ ComplexEmbedding.IsReal σ₀ → ∀ (𝔟 : ClassGroup (𝒪 K) → nonZeroDivisors (Ideal (𝒪 K))), (∀

C, ClassGroup.mk0 (𝔟 C) = C) → ∀ (M : IntermediateField ℚ ℂ) [NumberField M] (κ : K →+* M) (P : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A) (L₀ : (𝒪 M →+* ℂ) → PeriodPair) (Q : ClassGroup (𝒪 K) → Out.PolAV (Sp (𝒪 M))) (t : ∀ C, epsR (𝒪 M) ⟶ (Q C).

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogLieNormSqEqMeanLogEtaStatement

theorem **_root_.GlobalTheory.T.mean_log_lieNormSq_eq_mean_log_eta_alt** : ThetaUnifAV → Out.PolBaseChangeSt → (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (K : Type) [Field K] [NumberField K], Module.finrank ℚ K = 2 → ∀ (σ₀ : K →+* ℂ), ¬ ComplexEmbedding.IsReal σ₀ → ∀ (𝔟 : ClassGroup (𝒪 K) → nonZeroDivisors (Ideal (𝒪 K))), (∀ C, ClassGroup.mk0 (𝔟 C) = C) → ∀ (M : IntermediateField ℚ ℂ) [NumberField M] (κ : K →+* M) (P : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A) (L₀ : (𝒪 M →+* ℂ) → PeriodPair) (Q : ClassGroup (𝒪 K) → Out.PolAV (Sp (𝒪 M))) (t

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogSupNormLowerBoundCmFrameNonvanishing

theorem **letters_lines** {E : Type} [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] {H : Host E} {l : Level H} {c : Coeff l} {nΨ n' f : ℕ} {o : LamF.LamObj c nΨ n'} {La t : ℝ} (hG : LamF.LamObjGood o f La) (L : LamF.LamLetters o) (hL : L.IsGood) (hdeg : (∀ k, (o.ℓΨ k).ndeg = t / 2) ∧ ∀ k, (o.ℓ' k).ndeg = t / 2) : (∀ k, (L.ℓ k).Good) ∧ ∀ k, (L.ℓ k).ndeg = t / 2

theorem **meanLogSup_lowerBound** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, (∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v F) → (∀ σ, 0 < supC Xo c U ξ (goodOf c σ) a n F) → ∀ (k : (U c.F.val.toRingHom).Comp) (x₀ : ((U c.F.val.toRingHom).X k).dom) (Xx : FD.DataX E r) (hX : Xx.toData = (U c.F.val.toRingHom).X k) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E), 𝔄 ≠ 0 → Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ Xx.L', v = a • u} → ∀ (La : ℝ) (lam : Fin r → Fin r → E), 1 ≤ La → FD.LamBasis Xx La (hX ▸ x₀.1) lam → rdU (U c.F.val.

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogSupNormLowerBoundCmFrameNonvanishingStatement

theorem **_root_.GlobalTheory.T.meanLogSupNorm_lowerBound_cmFrame_nonvanishing** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (ξ : ℝ), (∀ σ, GoodEmb c σ → IsUnf (U σ)) → (∀ σ, GoodEmb c σ → ∀ k (τ : E →+* ℂ), ‖τ ((U σ).X k).ξ0‖ = ξ) → 1 ≤ Dw.p + Dw.q → ∀ [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → ∀ a n, Dom a n → ∀ F ∈ cusp Xo c U a n, (∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v F) → (∀ σ, 0 < supC Xo c U ξ (goodOf c σ) a n F) → ∀ (k : (U c.F.val.toRingHom).Comp) (x₀ : ((U c.F.val.toRingHom).X k).dom) (Xx : FD.DataX E r) (hX : Xx.toData = (U c.F.val.toRingHom).X k) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E), 𝔄 ≠ 0 → Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ Xx.L', v = a • u} → ∀ (La : ℝ) (lam : Fin r → Fin r → E), 1 ≤ La → FD.L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogSupNormLowerBoundOfCoreLaws

theorem **log_two_mul_mul_le** {g₁ g₂ s d L : ℝ} (h1 : 0 < g₁) (h2 : 0 < g₂) (hs : 0 < s) (hsd : s ≤ d) (hL : 0 ≤ L) (b1 : Real.log g₁ ≤ 9 * L + 1) (b2 : Real.log g₂ ≤ 9 * L + 1) : Real.log (2 * (g₁ * g₂) * s) ≤ 19 * L + 3 + Real.log d

theorem **lamwAt_of_lawsCore** : ∃ C0 : ℝ, 0 < C0 ∧ ∀ (D : D7) (T : Core D) (h : LawsCore T), GamAt T → ∀ (large₂ : SizeData₂ D), (∀ i, Real.log (D.host.g i) ≤ 9 * large₂.sc.L + 1) → ∀ (j : ℕ) (w : WeightFamilies.Wt D j) (C A : ℝ), C0 ≤ C → 2 ≤ A → Formalism.MainNotions.Gaps.EtaleWeightGaps.LAMwAt large₂ (recB h) j w C A

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MeanLogSupNormLowerBoundOfCoreLawsStatement

theorem **_root_.GlobalTheory.T.meanLogSupNorm_lower_bound_of_coreLaws** : ∃ C0 : ℝ, 0 < C0 ∧ ∀ (D : D7) (T : Core D) (h : LawsCore T), GamAt T → ∀ (large₂ : SizeData₂ D), (∀ i, Real.log (D.host.g i) ≤ 9 * large₂.sc.L + 1) → ∀ (j : ℕ) (w : WeightFamilies.Wt D j) (C A : ℝ), C0 ≤ C → 2 ≤ A → Formalism.MainNotions.Gaps.EtaleWeightGaps.LAMwAt large₂ (recB h) j w C A

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MemLieVectorLineIdealIffMemLieLattice

theorem **isVecLine_mem_iff_mem_lieLat** : ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (i : Fin x.P.g) (ℓ : MLine K), IsVecLine A x φ i ℓ → ∀ c : K, c ∈ ℓ.I ↔ (Pi.single i c : Fin x.P.g → K) ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).lieLat

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MemLieVectorLineIdealIffMemLieLatticeStatement

theorem **_root_.GlobalTheory.T.mem_lieVector_line_ideal_iff_mem_lie_lattice** : ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (i : Fin x.P.g) (ℓ : MLine K), IsVecLine A x φ i ℓ → ∀ c : K, c ∈ ℓ.I ↔ (Pi.single i c : Fin x.P.g → K) ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).lieLat

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MetricChangeOfFrame

theorem **lift_comb** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {G' : Over (Sp S)} [GrpObj G'] {φ : G'.left ⟶ Xo.univ.t.G.left} (hp : IsGrpPullback G' Xo.univ.t.G x φ) : (∀ w : epsR S ⟶ G', IsTanAt x (w.left ≫ φ) → IsTangent w) ∧ ∀ (n : ℕ) (c : Fin n → S) (v : Fin n → (epsR S ⟶ G')) (w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left), IsCombAt x (fun i => (v i).left ≫ φ) c w → ∃ w' : epsR S ⟶ G', w'.left ≫ φ = w ∧ IsLieComb c v w'

theorem **nonzeroLocus_scaleSec** {R : Type} [CommRing R] {X : Scheme.{0}} (p : X ⟶ Sp R) {L : LB X} {a : R} (ha : IsUnit a) (s : LB.Sec L) : Out.nonzeroLocus (scaleSec p a s) = Out.nonzeroLocus s

theorem **hermForm_sum** {g n : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (W : Fin n → Fin g → ℂ) (c d : Fin n → ℂ) : hermForm Hm (∑ l, c l • W l) (∑ j, d j • W j) = ∑ l, ∑ j, star (d j) * hermForm Hm (W l) (W j) * c l

theorem **metricAt_change** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} (x : Sp ℂ ⟶ Xo.X) (s s' : LB.Sec (PPx x)) (uu uu' : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left)) (N : ℝ) (Gr Y : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ) (a : ℂ) (ha : a ≠ 0) (ht : ∀ b i, IsTanAt x (uu b i)) (hc : ∀ b i, IsCombAt x (uu b) (fun j => Y b j i) (uu' b i)) (hs : SecBC (RingHom.id ℂ) a x x s s') (h : IsMetricAt x s uu N Gr) : IsMetricAt x s' uu' (‖a‖ ^ 2 * N) (fun b => (Y b).conjTranspose * Gr b * Y b)

theorem **isMetric_frChg** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (y y' : FrPt Xo c (CAt σ)) (N : ℝ) (Gr Y : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ)) (a : ℂ), a ≠ 0 → FrChg (S

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MetricChangeOfFrameStatement

theorem **_root_.GlobalTheory.T.metric_change_of_frame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (y y' : FrPt Xo c (CAt σ)) (N : ℝ) (Gr Y : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ)) (a : ℂ), a ≠ 0 → FrChg (S

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MetricDataUnique

theorem **uniq_cast** (x : KeyIn ℂ) (g : ℕ) (hg : x.P.g = g) (u : HolUnif x.P.A x.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)) (u' : HolUnif x.P.A x.P.L x.P.g) (Hm' : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (χ' : u'.Λ → ℂ) (θ' : (Fin x.P.g → ℂ) → ℂ) (Fd' : Set (Fin x.P.g → ℂ)) (h : u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume) (h' : u'.IsAH Hm' χ' ∧ u'.IsRig x.rig ∧ u'.IsFn x.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume) : (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal = (∫ z in Fd', ‖θ' z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm' z z).re)) / (volume Fd').toReal ∧ ∀ (v w : epsR ℂ ⟶ x.P.A) (V W : Fin g → ℂ) (V' W' : Fin x.P.g → ℂ), IsTangent v → IsTangent w → u.IsDiff V v → u.IsDiff W w → u'.IsDiff V' v → u'.IsDiff W' w → hermForm Hm V W = hermForm Hm' V' W'

theorem **spMap_dualMap_id** (S : Type) [CommRing S] : spMap (Formalism.L2.Pj.dualMap (RingHom.id S)) = 𝟙 (Sp (DualNumber S))

theorem **isMetricAt_unique** (x : Sp ℂ ⟶ Xo.X) (s : LB.Sec (PPx x)) (uu : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left)) (ht : ∀ τ i, IsTanAt x (uu τ i)) (N N' : ℝ) (Gr Gr' : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ) (h : IsMetricAt x s uu N Gr) (h' : IsMetricAt x s uu N' Gr') : N = N' ∧ Gr = Gr'

theorem **isMetric_unique** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) (N N' : ℝ) (Gr Gr' : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ)), IsMetric y N Gr → IsMetric y N' Gr' → N = N' ∧ Gr = Gr'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MetricDataUniqueStatement

theorem **_root_.GlobalTheory.T.metric_data_unique** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (y : FrPt Xo c (CAt σ)) (N N' : ℝ) (Gr Gr' : (∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ)), IsMetric y N Gr → IsMetric y N' Gr' → N = N' ∧ Gr = Gr'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MorphismsSpreadOutToFgSubalgebra

theorem **spMap_algHom_comp** {R A : Type} [CommRing R] [CommRing A] [Algebra k R] [Algebra k A] (f : R →ₐ[k] A) : spMap f.toRingHom ≫ spMap (algebraMap k R) = spMap (algebraMap k A)

theorem **bcMap_fst** (f : R →ₐ[k] A) : bcMap pB f ≫ pullback.fst _ _ = pullback.fst _ _

theorem **bcMap_snd** (f : R →ₐ[k] A) : bcMap pB f ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap f.toRingHom

theorem **bcMap_comp** (f : R →ₐ[k] R') (g : R' →ₐ[k] A) : bcMap pB g ≫ bcMap pB f = bcMap pB (g.comp f)

theorem **bcMap_comp_of_eq** (f : R →ₐ[k] R') (g : R' →ₐ[k] A) (h : R →ₐ[k] A) (e : ∀ x, g (f x) = h x) : bcMap pB g ≫ bcMap pB f = bcMap pB h

theorem **bcMap_eq_id** (f : R →ₐ[k] R) (e : ∀ x, f x = x) : bcMap pB f = 𝟙 _

theorem **bcMap_id** : bcMap pB (AlgHom.id k R) = 𝟙 _

theorem **eq_bcMap** (f : R →ₐ[k] A) {j : pullback pB (spMap (algebraMap k A)) ⟶ pullback pB (spMap (algebraMap k R))} (h₁ : j ≫ pullback.fst _ _ = pullback.fst _ _) (h₂ : j ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap f.toRingHom) : j = bcMap pB f

theorem **isPullback_bcMap** (f : R →ₐ[k] A) : IsPullback (bcMap pB f) (pullback.snd _ _) (pullback.snd _ _) (spMap f.toRingHom)

theorem **isSchemeTheoreticallyDominant_bcMap** [Flat pB] (f : R →ₐ[k] A) (hf : Function.Injective f) : IsSchemeTheoreticallyDominant (bcMap pB f)

theorem **compactSpace_bc** [QuasiCompact pB] {T : Scheme.{0}} [CompactSpace T] (g : T ⟶ Sp k) : CompactSpace ↥(pullback pB g)

theorem **exists_fg_comp_eq** [QuasiCompact pB] [QuasiSeparated pB] [LocallyOfFinitePresentation pZ] (φ : pullback pB (spMap (algebraMap k A)) ⟶ Z) (hφ : φ ≫ pZ = pullback.fst _ _ ≫ pB) : ∃ (R : Subalgebra k A) (ψ : pullback pB (spMap (algebraMap k R)) ⟶ Z), R.FG ∧ ψ ≫ pZ = pullback.fst _ _ ≫ pB ∧ bcMap pB R.val ≫ ψ = φ

theorem **fg_iSup** {ι : Type*} [Finite ι] {R : ι → Subalgebra k A} (hR : ∀ i, (R i).FG) : (⨆ i, R i).FG

theorem **exists_fg_forall_comp_eq** [QuasiCompact pB] [QuasiSeparated pB] [LocallyOfFinitePresentation pZ] {ι : Type*} [Finite ι] (φ : ι → (pullback pB (spMap (algebraMap k A)) ⟶ Z)) (hφ : ∀ i, φ i ≫ pZ = pullback.fst _ _ ≫ pB) (R₀ : Subalgebra k A) (hR₀ : R₀.FG) : ∃ (R : Subalgebra k A) (ψ : ι → (pullback pB (spMap (algebraMap k R)) ⟶ Z)), R.FG ∧ R₀ ≤ R ∧ ∀ i, ψ i ≫ pZ = pullback.fst _ _ ≫ pB ∧ bcMap pB R.val ≫ ψ i = φ i

theorem **exists_fg_comp_eq_comp** [QuasiCompact pB] [LocallyOfFiniteType pZ] (R : Subalgebra k A) (hR : R.FG) (a b : pullback pB (spMap (algebraMap k R)) ⟶ Z) (ha : a ≫ pZ = pullback.fst _ _ ≫ pB) (hb : b ≫ pZ = pullback.fst _ _ ≫ pB) (hab : bcMap pB R.val ≫ a = bcMap pB R.val ≫ b) : ∃ (R' : Subalgebra k A) (h : R ≤ R'), R'.FG ∧ bcMap pB (Subalgebra.inclusion h) ≫ a = bcMap pB (Subalgebra.inclusion h) ≫ b

theorem **hom_ext_bcMap** [Flat pB] {R : Type} [CommRing R] [Algebra k R] (f : R →ₐ[k] A) (hf : Function.Injective f) {S : Scheme.{0}} (s : Z ⟶ S) [IsSeparated s] {a b : pullback pB (spMap (algebraMap k R)) ⟶ Z} (h : a ≫ s = b ≫ s) (hab : bcMap pB f ≫ a = bcMap pB f ≫ b) : a = b

theorem **exists_fg_points** [LocallyOfFinitePresentation pZ] {ι : Type*} [Finite ι] (x : ι → (Sp A ⟶ Z)) (hx : ∀ i, x i ≫ pZ = spMap (algebraMap k A)) (R₀ : Subalgebra k A) (hR₀ : R₀.FG) : ∃ (R : Subalgebra k A) (y : ι → (Sp R ⟶ Z)), R.FG ∧ R₀ ≤ R ∧ ∀ i, y i ≫ pZ = spMap (algebraMap k R) ∧ spMap R.val.toRingHom ≫ y i = x i

theorem **point_ext** {R A : Type} [CommRing R] [CommRing A] (f : R →+* A) (hf : Function.Injective f) {S : Scheme.{0}} (s : Z ⟶ S) [IsSeparated s] {a b : Sp R ⟶ Z} (h : a ≫ s = b ≫ s) (hab : spMap f ≫ a = spMap f ≫ b) : a = b

theorem **dualMapR_injective** {R A : Type} [CommRing R] [CommRing A] {f : R →+* A} (hf : Function.Injective f) : Function.Injective (dualMapR f)

theorem **exists_fg_tangents** [LocallyOfFinitePresentation pZ] {ι : Type*} [Finite ι] (v : ι → (Sp (DualNumber A) ⟶ Z)) (hv : ∀ i, v i ≫ pZ = spMap (algebraMap k (DualNumber A))) (R₀ : Subalgebra k A) (hR₀ : R₀.FG) : ∃ (R : Subalgebra k A) (w : ι → (Sp (DualNumber R) ⟶ Z)), R.FG ∧ R₀ ≤ R ∧ ∀ i, w i ≫ pZ = spMap (algebraMap k (DualNumber R)) ∧ spMap (dualMapR R.val.toRingHom) ≫ w i = v i

theorem **tangent_ext** {R A : Type} [CommRing R] [CommRing A] (f : R →+* A) (hf : Function.Injective f) {S : Scheme.{0}} (s : Z ⟶ S) [IsSeparated s] {a b : Sp (DualNumber R) ⟶ Z} (h : a ≫ s = b ≫ s) (hab : spMap (dualMapR f) ≫ a = spMap (dualMapR f) ≫ b) : a = b

theorem **dominant_of_compatible** (R : Subalgebra k A) {j : pullback pB (spMap (algebraMap k A)) ⟶ pullback pB (spMap (algebraMap k R))} (h₁ : j ≫ pullback.fst _ _ = pullback.fst _ _) (h₂ : j ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap R.val.toRingHom) : QuasiCompact j ∧ IsSchemeTheoreticallyDominant j

theorem **hom_ext_of_compatible** (R : Subalgebra k A) {j : pullback pB (spMap (algebraMap k A)) ⟶ pullback pB (spMap (algebraMap k R))} (h₁ : j ≫ pullback.fst _ _ = pullback.fst _ _) (h₂ : j ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap R.val.toRingHom) {S : Scheme.{0}} (s : Z ⟶ S) [IsSeparated s] {a b : pullback pB (spMap (algebraMap k R)) ⟶ Z} (h : a ≫ s = b ≫ s) (hab : j ≫ a = j ≫ b) : a = b

theorem **morphisms_spread_out_to_fg_subalgebra** (k : IntermediateField ℚ ℂ) (B Z : Scheme.{0}) (pB : B ⟶ Sp k) (pZ : Z ⟶ Sp k) [QuasiCompact pB] [QuasiSeparated pB] [LocallyOfFinitePresentation pZ] (n : ℕ) (φ : Fin n → (pullback pB (spMap (algebraMap k ℂ)) ⟶ Z)) (hφ : ∀ i, φ i ≫ pZ = pullback.fst _ _ ≫ pB) (R₀ : Subalgebra k ℂ) (hR₀ : R₀.FG) : ∃ (R : Subalgebra k ℂ) (ψ : Fin n → (pullback pB (spMap (algebraMap k R)) ⟶ Z)) (j : pullback pB (spMap (algebraMap k ℂ)) ⟶ pullback pB (spMap (algebraMap k R))), R.FG ∧ R₀ ≤ R ∧ j ≫ pullback.fst _ _ = pullback.fst _ _ ∧ j ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap R.val.toRingHom ∧ ∀ i, ψ i ≫ pZ = pullback.fst _ _ ≫ pB ∧ j ≫ ψ i = φ i

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MultipleOfHermitianFormSumOfNorms

theorem **IsSumOfNorms.congr** {B B' : M → M → R} (h : IsSumOfNorms br B) (e : ∀ x y, B' x y = B x y) : IsSumOfNorms br B'

theorem **IsSumOfNorms.zero** : IsSumOfNorms br (fun (_ _ : M) => (0 : R))

theorem **IsSumOfNorms.one** (f : M →ₗ[R] R) : IsSumOfNorms br (fun x y => f x * br (f y))

theorem **IsSumOfNorms.add** {B B' : M → M → R} (h : IsSumOfNorms br B) (h' : IsSumOfNorms br B') : IsSumOfNorms br (fun x y => B x y + B' x y)

theorem **IsSumOfNorms.nsmul** {B : M → M → R} (h : IsSumOfNorms br B) (n : ℕ) : IsSumOfNorms br (fun x y => (n : R) * B x y)

theorem **IsSumOfNorms.comp** {B : M → M → R} (h : IsSumOfNorms br B) (Z : M →ₗ[R] M) : IsSumOfNorms br (fun x y => B (Z x) (Z y))

theorem **IsSumOfNorms.sum** {κ : Type} (s : Finset κ) {B : κ → M → M → R} (h : ∀ i, IsSumOfNorms br (B i)) : IsSumOfNorms br (fun x y => ∑ i ∈ s, B i x y)

theorem **exists_step** (e : M) (q : ℕ) (hq : g e e = q) : ∃ Z : M →ₗ[R] M, Z e = 0 ∧ ∀ x y, g (Z y) (Z x) = (q : R) ^ 2 * g y x - q * (g e x * br (g e y))

theorem **exists_kill** {κ : Type} (v : κ → M) (s : Finset κ) : ∃ (N : ℕ) (Z : M →ₗ[R] M), 1 ≤ N ∧ (∀ i ∈ s, Z (v i) = 0) ∧ IsSumOfNorms br (fun x y => (N : R) * g y x - g (Z y) (Z x))

theorem **exists_split** {n : ℕ} (u w : Fin n → M) (fr : ∀ a, a = ∑ i, g (w i) a • u i) : ∃ (N m : ℕ) (l : M →ₗ[R] (Fin m → R)) (r : (Fin m → R) →ₗ[R] M), 1 ≤ N ∧ r ∘ₗ l = LinearMap.id ∧ ∀ x y, (N : R) * g y x = ∑ k, l x k * br (l y k)

theorem **kForm_sum_sq** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → ∃ (N m : ℕ) (l : D.La →ₗ[OK H] (Fin m → OK H)) (r : (Fin m → OK H) →ₗ[OK H] D.La), 1 ≤ N ∧ r ∘ₗ l = LinearMap.id ∧ ∀ x y, (N : 𝒪 E) * (h x y : OK H) = ∑ k, ((l x k : OK H) : 𝒪 E) * H.bar (l y k : OK H)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MultipleOfHermitianFormSumOfNormsStatement

theorem **_root_.GlobalTheory.T.cmHermitianForm_sum_of_squares** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H), IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → ∃ (N m : ℕ) (l : D.La →ₗ[OK H] (Fin m → OK H)) (r : (Fin m → OK H) →ₗ[OK H] D.La), 1 ≤ N ∧ r ∘ₗ l = LinearMap.id ∧ ∀ x y, (N : 𝒪 E) * (h x y : OK H) = ∑ k, ((l x k : OK H) : 𝒪 E) * H.bar (l y k : OK H)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.MultiplierIffExistsHom

theorem **exists_coord** {g : ℕ} (hg : g = 1) {w : Fin g → ℂ} (hw : w ≠ 0) : ∃ c : (Fin g → ℂ) →ₗ[ℂ] ℂ, (∀ x, c x • w = x) ∧ c w = 1

theorem **exists_index** {L L' : PeriodPair} {α : ℂ} (h : ∀ z ∈ L.lattice, α * z ∈ L'.lattice) : ∃ n : ℕ, Complex.normSq α * |(L.ω₁ * starRingEnd ℂ L.ω₂).im| = n * |(L'.ω₁ * starRingEnd ℂ L'.ω₂).im|

theorem **covolume_ne_zero** (L : PeriodPair) : (L.ω₁ * starRingEnd ℂ L.ω₂).im ≠ 0

theorem **multiplier_iff_exists_hom** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (R : Type) [CommRing R] (P Q : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A) (t : epsR R ⟶ Q.A) (s : R →+* ℂ) (L L' : PeriodPair) (α : ℂ), P.g = 1 → Q.g = 1 → IsTangent v → IsTangent t → (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s P P' φ' → IsVecOver s φ' v v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ L.lattice ↔ z • w' ∈ u'.Λ) → (∀ (P' : Out.PolAV (Sp ℂ)) (φ' : P'.A.left ⟶ Q.A.left) (v' : epsR ℂ ⟶ P'.A) (u' : HolUnif P'.A P'.L P'.g) (w' : Fin P'.g → ℂ), IsPolFibre s Q P' φ' → IsVecOver s φ' t v' → u'.IsDiff w' v' → w' ≠ 0 ∧ ∀ z : ℂ, z ∈ L'.lattice ↔ z • w' ∈ u'.Λ) → ((∀ z ∈ L.lattice, α * z ∈ L'.lattice) ↔ (∀ (P₁ Q₁ : Out.PolAV (Sp ℂ)) (φ₁ : P₁.A.left ⟶ P.A.left) (ψ₁ : Q₁.A.left ⟶ Q.A.left) (v₁ : epsR ℂ ⟶ P₁.A) (t₁ : epsR ℂ ⟶ Q₁

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NonemptyIsLimitBaseChange

theorem **exists_gen_of_refinement** (L : LB Y) (hL : L.IsLB) {A : Type} (W : A → Y.Opens) (hW : ∀ y : Y, ∃ a, y ∈ W a) (e : A → L.ι) (he : ∀ a, W a ≤ L.U (e a)) (u : ∀ a, Γ(Y, W a)ˣ) (hu : ∀ a b (y : Y), y ∈ W a → y ∈ W b → gm (𝟙 Y) y (W a) (u a) = gm (𝟙 Y) y _ (L.g (e a) (e b)) * gm (𝟙 Y) y (W b) (u b)) : ∃ t : L.Gen, t.IsGen

theorem **gm_id_pullU** {X : Scheme.{0}} (F : Y ⟶ X) (y : Y) (U : X.Opens) (hy : F y ∈ U) (t : Γ(X, U)ˣ) : gm (𝟙 Y) y (F ⁻¹ᵁ U) (pullU F U t) = gm F y U t

theorem **gm_res_pullU** {X Z : Scheme.{0}} (r : Z ⟶ Y) (F : Y ⟶ X) (z : Z) {U : X.Opens} {W : Y.Opens} (le : W ≤ F ⁻¹ᵁ U) (hz : r z ∈ W) (t : Γ(X, U)ˣ) : gm r z W (res le (pullU F U t)) = gm (r ≫ F) z U t

theorem **gm_pullU_mul_res** {X Z : Scheme.{0}} (r : Z ⟶ Y) (F : Y ⟶ X) (z : Z) {U : X.Opens} {W : Y.Opens} (le : F ⁻¹ᵁ U ≤ W) (hz : r z ∈ F ⁻¹ᵁ U) (G : Γ(X, U)ˣ) (t : Γ(Y, W)ˣ) : gm r z (F ⁻¹ᵁ U) (pullU F U G * res le t) = gm (r ≫ F) z U G * gm r z W t

theorem **exists_common_refinement** {i : I} {A : Type} [Finite A] (P : A → ∀ j : I, (j ⟶ i) → Prop) (stable : ∀ a j (f : j ⟶ i) k (g : k ⟶ j), P a j f → P a k (g ≫ f)) (h : ∀ a, ∃ j f, P a j f) : ∃ j f, ∀ a, P a j f

theorem **cone_preimage_le** {i j : I} (f : j ⟶ i) (U : (D.obj i).Opens) : π i ⁻¹ᵁ U ≤ π j ⁻¹ᵁ (D.map f ⁻¹ᵁ U)

theorem **exists_appLE_π_eq** {i : I} {U : (D.obj i).Opens} (hU : IsAffineOpen U) (s : Γ(X, π i ⁻¹ᵁ U)) : ∃ (j : I) (f : j ⟶ i) (t : Γ(D.obj j, D.map f ⁻¹ᵁ U)), ((π j).appLE _ _ (cone_preimage_le D π w f U)).hom t = s

theorem **exists_unit_π_eq** {i : I} {U : (D.obj i).Opens} (hU : IsAffineOpen U) (s : Γ(X, π i ⁻¹ᵁ U)ˣ) : ∃ (j : I) (f : j ⟶ i) (t : Γ(D.obj j, D.map f ⁻¹ᵁ U)ˣ), ∀ y : X, y ∈ π i ⁻¹ᵁ U → gm (π j) y (D.map f ⁻¹ᵁ U) t = gm (𝟙 X) y (π i ⁻¹ᵁ U) s

theorem **exists_pullU_eq** {i : I} {U : (D.obj i).Opens} (hU : IsCompact (U : Set (D.obj i))) (s t : Γ(D.obj i, U)ˣ) (h : ∀ y : X, π i y ∈ U → gm (π i) y U s = gm (π i) y U t) : ∃ (j : I) (f : j ⟶ i), pullU (D.map f) U s = pullU (D.map f) U t

theorem **exists_units_of_isLimit** {i : I} [QuasiSeparatedSpace (D.obj i)] {A : Type} [Finite A] (V : A → (D.obj i).Opens) (hV : ∀ a, IsAffineOpen (V a)) (G : ∀ a b, Γ(D.obj i, V a ⊓ V b)ˣ) (s : ∀ a, Γ(X, π i ⁻¹ᵁ V a)ˣ) (hs : ∀ a b (y : X), y ∈ π i ⁻¹ᵁ V a → y ∈ π i ⁻¹ᵁ V b → gm (𝟙 X) y _ (s a) = gm (π i) y _ (G a b) * gm (𝟙 X) y _ (s b)) : ∃ (j : I) (f : j ⟶ i) (u : ∀ a, Γ(D.obj j, D.map f ⁻¹ᵁ V a)ˣ), ∀ a b (y : D.obj j), y ∈ D.map f ⁻¹ᵁ V a → y ∈ D.map f ⁻¹ᵁ V b → gm (𝟙 _) y _ (u a) = gm (D.map f) y _ (G a b) * gm (𝟙 _) y _ (u b)

theorem **lb_div_isLB** {Y : Scheme.{0}} {M N : LB Y} (hM : M.IsLB) (hN : N.IsLB) : (LB.div N M).IsLB where

theorem **gm_div_pull_g** {Y Z : Scheme.{0}} (M N : LB Y) (F : Z ⟶ Y) (z : Z) (p q : N.ι × M.ι) (hz : F z ∈ (LB.div N M).U p ⊓ (LB.div N M).U q) : gm (𝟙 Z) z _ ((LB.div (N.pull F) (M.pull F)).g (fun _ => p.1, fun _ => p.2) (fun _ => q.1, fun _ => q.2)) = gm F z _ ((LB.div N M).g p q)

theorem **exists_lbIso_of_isLimit** {i : I} [CompactSpace (D.obj i)] [QuasiSeparatedSpace (D.obj i)] (M N : LB (D.obj i)) (hM : M.IsLB) (hN : N.IsLB) (h : LBIso (M.pull (π i)) (N.pull (π i))) : ∃ (j : I) (f : j ⟶ i), LBIso (M.pull (D.map f)) (N.pull (D.map f))

theorem **FGAbove.exists_ge** (R S : FGAbove R₀) : ∃ T : FGAbove R₀, R ≤ T ∧ S ≤ T

theorem **FGAbove.exists_mem** (h : R₀.FG) (x : C) : ∃ R : FGAbove R₀, x ∈ R.1

theorem **FGAbove.isFiltered** (h : R₀.FG) : IsFiltered (FGAbove R₀)

theorem **FGAbove.exists_ringHom** (h : R₀.FG) {S : Type} [CommRing S] (φ : ∀ R : FGAbove R₀, R.1 →+* S) (hφ : ∀ (R T : FGAbove R₀) (hRT : R.1 ≤ T.1) (x : R.1), φ T (Subalgebra.inclusion hRT x) = φ R x) : ∃ ψ : C →+* S, ∀ (R : FGAbove R₀) (x : R.1), ψ x = φ R x

theorem **spMap_comp_ring** {A A' A'' : Type} [CommRing A] [CommRing A'] [CommRing A''] (f : A →+* A') (g : A' →+* A'') : spMap g ≫ spMap f = spMap (g.comp f)

theorem **baseChangeMap_fst** (f : A →ₐ[k] A') : baseChangeMap pB f ≫ pullback.fst _ _ = pullback.fst _ _

theorem **baseChangeMap_snd** (f : A →ₐ[k] A') : baseChangeMap pB f ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap f.toRingHom

theorem **baseChangeMap_comp** (f : A →ₐ[k] A') (g : A' →ₐ[k] A'') : baseChangeMap pB g ≫ baseChangeMap pB f = baseChangeMap pB (g.comp f)

theorem **baseChangeMap_id** : baseChangeMap pB (AlgHom.id k A) = 𝟙 _

theorem **isAffineHom_baseChangeMap** (f : A →ₐ[k] A') : IsAffineHom (baseChangeMap pB f)

theorem **hom_spec_ext** {T : Scheme.{0}} {A : CommRingCat.{0}} {f g : T ⟶ Spec A} (h : (Scheme.ΓSpecIso A).inv ≫ f.appTop = (Scheme.ΓSpecIso A).inv ≫ g.appTop) : f = g

theorem **appTop_comp_spMap** {T : Scheme.{0}} {A A' : Type} [CommRing A] [CommRing A'] (φ : A →+* A') (f : T ⟶ Sp A') (x : A) : ((Scheme.ΓSpecIso (.of A)).inv ≫ (f ≫ spMap φ).appTop).hom x = ((Scheme.ΓSpecIso (.of A')).inv ≫ f.appTop).hom (φ x)

theorem **nonempty_isLimit_baseChange** (h : R₀.FG) : Nonempty (IsLimit (schemeConeOfFamily (baseChangeDiagram R₀ pB) (X

theorem **exists_fg_lbIso** [QuasiCompact pB] [QuasiSeparated pB] (h : R₀.FG) (M N : LB (pullback pB (spMap (algebraMap k R₀)))) (hM : M.IsLB) (hN : N.IsLB) (H : LBIso (M.pull (baseChangeMap pB R₀.val)) (N.pull (baseChangeMap pB R₀.val))) : ∃ (R : Subalgebra k C) (hR : R₀ ≤ R), R.FG ∧ LBIso (M.pull (baseChangeMap pB (Subalgebra.inclusion hR))) (N.pull (baseChangeMap pB (Subalgebra.inclusion hR)))

theorem **lbIso_spreads_out** (k : IntermediateField ℚ ℂ) (B : Scheme.{0}) (pB : B ⟶ Sp k) [QuasiCompact pB] [QuasiSeparated pB] (R₀ : Subalgebra k ℂ) (h : R₀.FG) (M N : LB (pullback pB (spMap (algebraMap k R₀)))) (hM : M.IsLB) (hN : N.IsLB) (j₀ : pullback pB (spMap (algebraMap k ℂ)) ⟶ pullback pB (spMap (algebraMap k R₀))) (h₁ : j₀ ≫ pullback.fst _ _ = pullback.fst _ _) (h₂ : j₀ ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap R₀.val.toRingHom) (H : LBIso (M.pull j₀) (N.pull j₀)) : ∃ (R : Subalgebra k ℂ) (hR : R₀ ≤ R) (i : pullback pB (spMap (algebraMap k R)) ⟶ pullback pB (spMap (algebraMap k R₀))), R.FG ∧ i ≫ pullback.fst _ _ = pullback.fst _ _ ∧ i ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap (Subalgebra.inclusion hR).toRingHom ∧ LBIso (M.pull i) (N.pull i)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NonemptyPolAVGenericFibreWithLieBasis

theorem **GenFib.nonempty_of_baseChange** : Out.PolBaseChangeSt → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))), Nonempty (GenFib P)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NonemptyPolAVGenericFibreWithLieBasisStatement

theorem **_root_.GlobalTheory.T.nonempty_polAVGenericFibreWithLieBasis** : Out.PolBaseChangeSt → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))), Nonempty (GenFib P)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NonemptyTensorOfOntoFullFamily

theorem **OntoAt.exists_eq_comp** {S : Scheme.{0}} {I : Type} {B : I → Over S} {A A' : Over S} [GrpObj A] [GrpObj A'] {φ : ∀ i, B i ⟶ A} {k : Type} [Field k] {t : Sp k ⟶ S} (hφ : OntoAt φ t) (Φ : A' ⟶ A) [IsMonHom Φ] (χ : ∀ i, B i ⟶ A') (e : ∀ i, χ i ≫ Φ = φ i) (y : Over.mk t ⟶ A) : ∃ x : Over.mk t ⟶ A', y = x ≫ Φ

theorem **nonempty_isTen_of_isFam_top** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsDomain R] [IsNoetherianRing R] (σ : R →+* ℂ) (ω : R) (C : Ell H R ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (ψ : (⊤ : Submodule (𝒪 E) D.La) → (C.P.A ⟶ T.t.G)), Function.Injective σ → IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → IsFam C T h ⊤ ψ → OntoAt ψ (spMap σ) → Nonempty (IsTen C D h T)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NonemptyTensorOfOntoFullFamilyStatement

theorem **_root_.GlobalTheory.T.nonempty_tensor_of_onto_full_family** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsDomain R] [IsNoetherianRing R] (σ : R →+* ℂ) (ω : R) (C : Ell H R ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (T : PAT H (Sp R)) (ψ : (⊤ : Submodule (𝒪 E) D.La) → (C.P.A ⟶ T.t.G)), Function.Injective σ → IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → IsFam C T h ⊤ ψ → OntoAt ψ (spMap σ) → Nonempty (IsTen C D h T)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NormMulDetGram

theorem **eq_dualMapR** {R R' : Type} [CommRing R] [CommRing R'] {f : R →+* R'} {F : DualNumber R →+* DualNumber R'} (h₁ : ∀ a, F (algebraMap R _ a) = algebraMap R' _ (f a)) (h₂ : F DualNumber.eps = DualNumber.eps) : F = Chart.dualMapR f

theorem **det_gram** {g : ℕ} (Hm W : Matrix (Fin g) (Fin g) ℂ) : (Matrix.of fun i j => (Real.pi : ℂ) * hermForm Hm (W j) (W i)).det.re = Real.pi ^ g * ‖W.det‖ ^ 2 * Hm.det.re

theorem **norm_mul_det_gram** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (F : GenFib P) (hg rig hr th hth) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ (σ : K →+* ℂ) (x' : KeyIn ℂ) (nth nη : ℝ), KeyPB x' (F.key hg rig

hr th hth) σ → KeyPow.IsArchNorm x' nth nη → nη * (h.gram σ (Pi.basisFun K (Fin F.P₀.g))).det.re = 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NormalisedDegreeCmEigenLineEq

theorem **splits_of_injective_label** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] {lab : (E →+* ℂ) → (E →+* K)} (hl : Function.Injective lab) : GlobalTheory.CM.Splits E K

theorem **ndeg_vecLine_eq** : DegSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (oa : A.OAct (𝒪 E)) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (S : Set (E →+* ℂ)) (τ : Fin x.P.g ≃ S) (lab : (E →+* ℂ) → (E →+* K)) (ρ : 𝒪 K), IsFibreOf A x (gmap K) φ → Function.Injective lab → ρ ^ 2 = -(H.d : 𝒪 K) → (∀ t, t ∈ S ↔ lab t (H.rt : E) = (ρ : K)) → IsLieType A oa x φ τ lab → ∀ (i : Fin x.P.g) (ℓ : MLine K), IsVecLine A x φ i ℓ → ℓ.ndeg = (logOm E χ).re / 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.NormalisedDegreeCmEigenLineEqStatement

theorem **_root_.GlobalTheory.T.normalisedDegree_cmEigenLine_eq** : DegSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → (∀ a : 𝒪 E, star a = H.bar a) → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (oa : A.OAct (𝒪 E)) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (S : Set (E →+* ℂ)) (τ : Fin x.P.g ≃ S) (lab : (E →+* ℂ) → (E →+* K)) (ρ : 𝒪 K), IsFibreOf A x (gmap K) φ → Function.Injective lab → ρ ^ 2 = -(H.d : 𝒪 K) → (∀ t, t ∈ S ↔ lab t (H.rt : E) = (ρ : K)) → IsLieType A oa x φ τ lab → ∀ (i : Fin x.P.g) (ℓ : MLine K), IsVecLine A x φ i ℓ → ℓ.ndeg = (logOm E χ).re / 2

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.OneMulConjFracOneEqInvDifferent

theorem **deg_lineHerm** (n : (L →+* ℂ) → ℝ) (hn : ∀ s, 0 < n s) {𝔠 : FractionalIdeal (𝒪 L)⁰ L} (h𝔠 : 𝔠 ≠ 0) : deg (lineHerm n hn) (𝔠 : Submodule (𝒪 L) L) = -Real.log (FractionalIdeal.absNorm 𝔠) / finrank ℚ L - meanLog L n / 2

theorem **absNorm_extendedHom** [Algebra E L] (𝔞 : FractionalIdeal (𝒪 E)⁰ E) : FractionalIdeal.absNorm (FractionalIdeal.extendedHom L (𝒪 L) 𝔞) = FractionalIdeal.absNorm 𝔞 ^ finrank E L

theorem **exists_fractionalIdeal_eq_span_image** (τ : E →+* L) {𝔞 : FractionalIdeal (𝒪 E)⁰ E} (h𝔞 : 𝔞 ≠ 0) : ∃ 𝔠 : FractionalIdeal (𝒪 L)⁰ L, 𝔠 ≠ 0 ∧ (𝔠 : Submodule (𝒪 L) L) = Submodule.span (𝒪 L) (τ '' 𝔞) ∧ Real.log (FractionalIdeal.absNorm 𝔠) / finrank ℚ L = Real.log (FractionalIdeal.absNorm 𝔞) / finrank ℚ E

theorem **meanLog_comp** (j : E →+* L) (n : (E →+* ℂ) → ℝ) : meanLog L (fun s => n (s.comp j)) = meanLog E n

theorem **log_absNorm_conjFrac** [IsCMField E] {𝔞 : FractionalIdeal (𝒪 E)⁰ E} (h𝔞 : 𝔞 ≠ 0) : Real.log (FractionalIdeal.absNorm (conjFrac 𝔞)) = Real.log (FractionalIdeal.absNorm 𝔞)

theorem **log_rootDiscr_tower** [Algebra E L] : Real.log (rootDiscr L) = Real.log (Ideal.absNorm (differentIdeal (𝒪 E) (𝒪 L))) / finrank ℚ L + Real.log (rootDiscr E)

theorem **two_mul_deg_lineHerm_different** {𝔄 : FractionalIdeal (𝒪 E)⁰ E} (h𝔄 : 𝔄 * conjFrac 𝔄 = (differentIdeal (𝒪 K) (𝒪 E) : FractionalIdeal (𝒪 E)⁰ E)⁻¹) (n : (M →+* ℂ) → ℝ) (hn : ∀ s, 0 < n s) (j : M →+* L) (τ : E →+* L) : 2 * deg (lineHerm (fun s => n (s.comp j)) fun _ => hn _) (Submodule.span (𝒪 L) (τ '' (𝔄 * differentIdeal (𝒪 K) (𝒪 E) : FractionalIdeal (𝒪 E)⁰ E))) = -meanLog M n + Real.log (rootDiscr K) - Real.log (rootDiscr E)

theorem **rootDiscr_eq_sqrt** (h2 : finrank ℚ K = 2) (hD : discr K = -d) : rootDiscr K = √d

theorem **chowla_selberg_of_mean_log** (h2 : finrank ℚ K = 2) (hD : discr K = -d) (hd : 4 < d) {χ : DirichletCharacter ℂ d} (hχ : IsKroneckerChar χ) {M : Type} [Field M] [NumberField M] (n : (M →+* ℂ) → ℝ) (hn : ∀ s, 0 < n s) (h : ∀ (σ : K →+* ℂ) (τ : ClassGroup (𝒪 K) → ℍ), (∀ C, IsIdealClassPoint σ C (τ C)) → (∑ s, Real.log (n s)) / finrank ℚ M = (∑ C, -Real.log (4 * Real.pi * (τ C).im * ‖ModularForm.eta (τ C)‖ ^ 4)) / classNumber K) : ((-meanLog M n : ℝ) : ℂ) = (Real.eulerMascheroniConstant + Real.log (2 * Real.pi / √d) : ℝ) - deriv χ.LFunction 1 / χ.LFunction 1 ∧ ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [Algebra K E] (𝔄 : FractionalIdeal (𝒪 E)⁰ E), 𝔄 * conjFrac 𝔄 = (differentIdeal (𝒪 K) (𝒪 E) : FractionalIdeal (𝒪 E)⁰ E)⁻¹ → ∀ (L : Type) [Field L] [NumberField L] (j : M →+* L) (τ : E →+* L), ((2 * deg (lineHerm (fun s => n (s.comp j)) fun _ => hn _) (Submodule.span (𝒪 L) (τ ''

theorem **chowla_selberg_lie_lines** (hC : CMCurve) (h7 : 7 ≤ d) (hs : Squarefree d) (h4 : d % 4 = 3) (h2 : finrank ℚ K = 2) (hr : ∃ r : K, r ^ 2 = -(d : K)) {χ : DirichletCharacter ℂ d} (hχ : IsKroneckerChar χ) : ∃ (M : IntermediateField ℚ ℂ) (_ : NumberField M) (w : M) (P : Out.PolAV (Sp (𝒪 M))) (v : epsR (𝒪 M) ⟶ P.A), (w : ℂ) = (1 + Complex.I * √d) / 2 ∧ P.g = 1 ∧ IsLieBasis ![v] ∧ (∃ (P₀ : Out.PolAV (Sp M)) (φ : P₀.A.left ⟶ P.A.left) (v₀ : epsR M ⟶ P₀.A) (W : P₀.A ⟶ P₀.A), IsPolFibre (algebraMap (𝒪 M) M) P P₀ φ ∧ IsVecOver (algebraMap (𝒪 M) M) φ v v₀ ∧ IsLieBasis ![v₀] ∧ IsMonHom W ∧ (W ≫ W) * (𝟙 P₀.A) ^ ((1 + d) / 4) = W ∧ IsLieComb ![w] ![v₀] (v₀ ≫ W)) ∧ (∀ (P' : Out.PolAV (Sp ℂ)) (φ : P'.A.left ⟶ P.A.left), IsPolFibre ((algebraMap M ℂ).comp (algebraMap (𝒪 M) M)) P P' φ → ∀ u : HolUnif P'.A P'.L P'.g, ∃ c : ℂ, c ≠ 0 ∧ ∀ z, z ∈ u.Λ ↔ ∃ m n : ℤ, z = fun _ => c * (m + n * (w : ℂ))) ∧ ∃ (n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.OriginLineBundlePolarisation

theorem **appLE_top_eq_zero_iff** {X T : Scheme.{0}} (t : T ⟶ X) {U : X.Opens} (h : ⊤ ≤ t ⁻¹ᵁ U) (a : Γ(X, U)) : (t.appLE U ⊤ h).hom a = 0 ↔ (t.app U).hom a = 0

theorem **appLE_originSec_eq_zero_iff** {R : Type} [CommRing R] (W : WeierstrassCurve R) {T : Scheme.{0}} (t : T ⟶ scheme W) (i : (originLB W).ι) (h : ⊤ ≤ t ⁻¹ᵁ (originLB W).U i) : (t.appLE ((originLB W).U i) ⊤ h).hom ((originSec W).f i) = 0 ↔ t = t ≫ toSpec W ≫ zero W

theorem **appLE_sec_pull_originLB_eq_zero** {R : Type} [CommRing R] (W : WeierstrassCurve R) {R' : Type} [CommRing R'] {f : Sp R' ⟶ Sp R} {Y : Scheme.{0}} {φ : Y ⟶ scheme W} {q : Y ⟶ Sp R'} (H : IsPullback φ q (toSpec W) f) (s : ((originLB W).pull φ).Sec) {T : Scheme.{0}} (t : T ⟶ Y) (j : ((originLB W).pull φ).ι) (h : ⊤ ≤ t ⁻¹ᵁ ((originLB W).pull φ).U j) (e : t ≫ φ = t ≫ φ ≫ toSpec W ≫ zero W) : (t.appLE (((originLB W).pull φ).U j) ⊤ h).hom (s.f j) = 0

theorem **originLB_polarisation** (R : Type) [CommRing R] (W : WeierstrassCurve R) [GrpObj (schemeOver W)] (_hc : IsCommMonObj (schemeOver W)) (he : η[schemeOver W].left = zero W) : Out.Rigidified (schemeOver W) (originLB W) ∧ Out.Sym (schemeOver W) (originLB W) ∧ Out.RelAmple (schemeOver W) (originLB W) ∧ Out.KTriv (schemeOver W) (originLB W)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.OriginLineBundlePolarisationStatement

theorem **_root_.GlobalTheory.T.originLineBundle_polarisation** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R) [GrpObj (schemeOver W)], IsCommMonObj (schemeOver W) → η[schemeOver W].left = zero W → Out.Rigidified (schemeOver W) (originLB W) ∧ Out.Sym (schemeOver W) (originLB W) ∧ Out.RelAmple (schemeOver W) (originLB W) ∧ Out.KTriv (schemeOver W) (originLB W)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PointsSumsEqProducts

theorem **points_sums_eq_products** : (∀ (S : Scheme.{0}) (G T : Over S) [GrpObj G], (toUnit T ≫ η[G] = (1 : T ⟶ G)) ∧ (∀ a b : T ⟶ G, padd G a b = a ∗ b) ∧ (∀ (n : ℕ) (a : T ⟶ G), pnsmulPt n a = a ^ n) ∧ ∀ (n : ℕ) (f : Fin n → (T ⟶ G)), psumPt f = (List.ofFn f).prod) ∧ ∀ (R R' : Type) [CommRing R] [CommRing R'] (f : R →+* R'), Formalism.L2.Pj.dualMap f = Chart.dualMapR f

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PolCountableAndTangentExt

theorem **pol_countable_and_tangent_ext** (hT : ThetaUnifAV) (Q Q' : Out.PolAV (Sp ℂ)) : {f : Q.A ⟶ Q'.A | IsMonHom f}.Countable ∧ ∀ f f' : Q.A ⟶ Q'.A, IsMonHom f → IsMonHom f' → (∀ v, IsTangent v → v ≫ f = v ≫ f') → f = f'

theorem **isTangent_comp** {R : Type} [CommRing R] {G H : Over (Sp R)} [GrpObj G] [GrpObj H] (f : G ⟶ H) [IsMonHom f] {v : epsR R ⟶ G} (hv : IsTangent v) : IsTangent (v ≫ f)

theorem **exists_pol_iso** (hB : Out.PolBaseChangeSt) {S S' : Scheme.{0}} (t : S' ⟶ S) (P : Out.PolAV S) : ∃ (Q : Out.PolAV S') (e : Q.A ≅ (Over.pullback t).obj P.A), IsMonHom e.hom

theorem **countable_and_tangent_ext** (hT : ThetaUnifAV) (hB : Out.PolBaseChangeSt) {S : Scheme.{0}} (t : Sp ℂ ⟶ S) (P P' : Out.PolAV S) : {f : (Over.pullback t).obj P.A ⟶ (Over.pullback t).obj P'.A | IsMonHom f}.Countable ∧ ∀ f f' : (Over.pullback t).obj P.A ⟶ (Over.pullback t).obj P'.A, IsMonHom f → IsMonHom f' → (∀ v, IsTangent v → v ≫ f = v ≫ f') → f = f'

theorem **basis_repr_map** {J : Type} (b : Module.Basis J k B) (ρ : R →ₐ[k] S) (x : R ⊗[k] B) (i : J) : (Algebra.TensorProduct.basis S b).repr (Algebra.TensorProduct.map ρ (AlgHom.id k B) x) i = ρ ((Algebra.TensorProduct.basis R b).repr x i)

theorem **specialise_eq_of_countable** (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (R : Subalgebra k ℂ) (hR : R.FG) {B : Type} [CommRing B] [Algebra k B] {Z : Scheme.{0}} (Φ : Sp (R ⊗[k] B) ⟶ Z) (U : Z.Opens) (hU : IsAffineOpen U) (hΦ : ∀ x, Φ x ∈ U) (hc : (Set.range fun ρ : R →ₐ[k] ℂ => spMap (Algebra.TensorProduct.map ρ (AlgHom.id k B)).toRingHom ≫ Φ).Countable) (ρ₁ : R →ₐ[k] ℂ) (hρ₁ : ∀ c : R, IsAlgebraic k c → ρ₁ c = c) : spMap (Algebra.TensorProduct.map ρ₁ (AlgHom.id k B)).toRingHom ≫ Φ = spMap (Algebra.TensorProduct.map R.val (AlgHom.id k B)).toRingHom ≫ Φ

theorem **exists_isMonHom_pull** {S S' : Scheme.{0}} {j : S' ⟶ S} {G H : Over S} [GrpObj G] [GrpObj H] {G' H' : Over S'} [GrpObj G'] [GrpObj H'] {φ : G'.left ⟶ G.left} {ψ : H'.left ⟶ H.left} (hG : IsGrpPullback G' G j φ) (hH : IsGrpPullback H' H j ψ) (f : G ⟶ H) [IsMonHom f] : ∃ f' : G' ⟶ H', IsMonHom f' ∧ f'.left ≫ ψ = φ ≫ f.left

theorem **isGrpPullback_bcMap** (A : Over (Sp k)) [GrpObj A] (ρ : R →ₐ[k] S) : IsGrpPullback ((Over.pullback (spMap (algebraMap k S))).obj A) ((Over.pullback (spMap (algebraMap k R))).obj A) (spMap ρ.toRingHom) (bcMap A.hom ρ)

theorem **exists_specialise** {A A' : Over (Sp k)} [GrpObj A] [GrpObj A'] (f : (Over.pullback (spMap (algebraMap k R))).obj A ⟶ (Over.pullback (spMap (algebraMap k R))).obj A') [IsMonHom f] (ρ : R →ₐ[k] S) : ∃ g : (Over.pullback (spMap (algebraMap k S))).obj A ⟶ (Over.pullback (spMap (algebraMap k S))).obj A', IsMonHom g ∧ g.left ≫ pullback.fst _ _ = bcMap A.hom ρ ≫ f.left ≫ pullback.fst _ _

theorem **spMap_comp** {R₁ R₂ R₃ : Type} [CommRing R₁] [CommRing R₂] [CommRing R₃] (f : R₁ →+* R₂) (g : R₂ →+* R₃) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **spMap_ptBC_eq** {t : Sp B ⟶ X} (ht : t ≫ p = spMap (algebraMap k B)) {C : Type} [CommRing C] (m : R ⊗[k] B →+* C) (x : Sp C ⟶ pullback p (spMap (algebraMap k R))) (h₁ : x ≫ pullback.fst _ _ = spMap (m.comp (Algebra.TensorProduct.includeRight : B →ₐ[k] R ⊗[k] B).toRingHom) ≫ t) (h₂ : x ≫ pullback.snd _ _ = spMap (m.comp (Algebra.TensorProduct.includeLeft : R →ₐ[k] R ⊗[k] B).toRingHom)) : spMap m ≫ ptBC p t ht R = x

theorem **spMap_map_ptBC** {t : Sp B ⟶ X} (ht : t ≫ p = spMap (algebraMap k B)) (ρ : R →ₐ[k] S) : spMap (Algebra.TensorProduct.map ρ (AlgHom.id k B)).toRingHom ≫ ptBC p t ht R = ptBC p t ht S ≫ bcMap p ρ

theorem **surjective_spMap_map_fst** {C : Type} [CommRing C] [Algebra k C] : Function.Surjective (spMap (Algebra.TensorProduct.map (AlgHom.id k R) ((TrivSqZeroExt.fstHom C C C).restrictScalars k)).toRingHom)

theorem **tangent_over** (v : epsR C ⟶ (Over.pullback (spMap (algebraMap k C))).obj A) : (v.left ≫ pullback.fst _ _) ≫ A.hom = spMap (algebraMap k (DualNumber C))

theorem **tangent_factor** (v : epsR C ⟶ (Over.pullback (spMap (algebraMap k C))).obj A) : ∃ m : Sp (DualNumber C) ⟶ Sp (C ⊗[k] DualNumber C), m ≫ ptBC A.hom _ (tangent_over v) C = v.left

theorem **tangent_range** {R : Type} [CommRing R] [Algebra k R] (f : (Over.pullback (spMap (algebraMap k R))).obj A ⟶ (Over.pullback (spMap (algebraMap k R))).obj A') [IsMonHom f] (v : epsR C ⟶ (Over.pullback (spMap (algebraMap k C))).obj A) (hv : IsTangent v) (x : Sp (R ⊗[k] DualNumber C)) : (ptBC A.hom _ (tangent_over v) R ≫ f.left ≫ pullback.fst _ _) x ∈ Set.range (eSec A')

theorem **exists_fg_isMonHom** [QuasiCompact A.hom] [QuasiSeparated A.hom] [Flat A.hom] [LocallyOfFinitePresentation A'.hom] [IsSeparated A'.hom] (f : (Over.pullback (spMap (algebraMap k C))).obj A ⟶ (Over.pullback (spMap (algebraMap k C))).obj A') [IsMonHom f] : ∃ (R : Subalgebra k C) (g : (Over.pullback (spMap (algebraMap k R))).obj A ⟶ (Over.pullback (spMap (algebraMap k R))).obj A'), R.FG ∧ IsMonHom g ∧ bcMap A.hom R.val ≫ g.left ≫ pullback.fst _ _ = f.left ≫ pullback.fst _ _

theorem **hom_descends_to_numberField** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (P P' : Out.PolAV (Sp k)) (f : (Over.pullback (spMap (algebraMap k ℂ))).obj P.A ⟶ (Over.pullback (spMap (algebraMap k ℂ))).obj P'.A), IsMonHom f → ∃ (K₁ : IntermediateField k ℂ) (f₁ : (Over.pullback (spMap (algebraMap k K₁))).obj P.A ⟶ (Over.pullback (spMap (algebraMap k K₁))).obj P'.A) (j : pullback P.A.hom (spMap (algebraMap k ℂ)) ⟶ pullback P.A.hom (spMap (algebraMap k K₁))), FiniteDimensional k K₁ ∧ IsMonHom f₁ ∧ j ≫ pullback.fst _ _ = pullback.fst _ _ ∧ j ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap K₁.val.toRingHom ∧ j ≫ f₁.left ≫ pullback.fst _ _ = f.left ≫ pullback.fst _ _

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PolFibreIdCompCancel

theorem **dualMapR_id** : Chart.dualMapR (RingHom.id R) = RingHom.id _

theorem **dualMapR_comp** (f : R →+* R') (g : R' →+* R'') : (Chart.dualMapR g).comp (Chart.dualMapR f) = Chart.dualMapR (g.comp f)

theorem **spMap_dual_hom** (f : R →+* R') : spMap (Chart.dualMapR f) ≫ (epsR R).hom = (epsR R').hom ≫ spMap f

theorem **lbIso_pull_comp** {X Y Z : Scheme.{0}} {L : LB X} {M : LB Y} {φ : Y ⟶ X} (h : LBIso (L.pull φ) M) (ψ : Z ⟶ Y) : LBIso (L.pull (ψ ≫ φ)) (M.pull ψ)

theorem **refl** (P : Out.PolAV (Sp R)) : IsPolFibre (RingHom.id R) P P (𝟙 _)

theorem **vec_refl** {A : Over (Sp R)} (v : epsR R ⟶ A) : IsVecOver (RingHom.id R) (𝟙 _) v v

theorem **comp** (h : IsPolFibre f P P' φ) {ψ : P''.A.left ⟶ P'.A.left} (h' : IsPolFibre g P' P'' ψ) : IsPolFibre (g.comp f) P P'' (ψ ≫ φ)

theorem **cancel** (h : IsPolFibre f P P' φ) {χ : P''.A.left ⟶ P.A.left} (h' : IsPolFibre (g.comp f) P P'' χ) : ∃ ψ, IsPolFibre g P' P'' ψ ∧ ψ ≫ φ = χ

theorem **vec_iff** {A : Over (Sp R)} {A' : Over (Sp R')} {A'' : Over (Sp R'')} {φ : A'.left ⟶ A.left} (hφ : IsPullback φ A'.hom A.hom (spMap f)) {ψ : A''.left ⟶ A'.left} {v : epsR R ⟶ A} {v' : epsR R' ⟶ A'} {v'' : epsR R'' ⟶ A''} (w : ψ ≫ A'.hom = A''.hom ≫ spMap g) (h : IsVecOver f φ v v') : IsVecOver g ψ v' v'' ↔ IsVecOver (g.comp f) (ψ ≫ φ) v v''

theorem **polFibre_id_comp_cancel** : (∀ (R : Type) [CommRing R] (P : Out.PolAV (Sp R)) (v : epsR R ⟶ P.A), IsPolFibre (RingHom.id R) P P (𝟙 _) ∧ IsVecOver (RingHom.id R) (𝟙 _) v v) ∧ ∀ (R R' R'' : Type) [CommRing R] [CommRing R'] [CommRing R''] (f : R →+* R') (g : R' →+* R'') (P : Out.PolAV (Sp R)) (P' : Out.PolAV (Sp R')) (P'' : Out.PolAV (Sp R'')) (φ : P'.A.left ⟶ P.A.left), IsPolFibre f P P' φ → (∀ ψ : P''.A.left ⟶ P'.A.left, IsPolFibre g P' P'' ψ → IsPolFibre (g.comp f) P P'' (ψ ≫ φ)) ∧ (∀ χ : P''.A.left ⟶ P.A.left, IsPolFibre (g.comp f) P P'' χ → ∃ ψ, IsPolFibre g P' P'' ψ ∧ ψ ≫ φ = χ) ∧ ∀ (ψ : P''.A.left ⟶ P'.A.left) (v : epsR R ⟶ P.A) (v' : epsR R' ⟶ P'.A) (v'' : epsR R'' ⟶ P''.A), ψ ≫ P'.A.hom = P''.A.hom ≫ spMap g → IsVecOver f φ v v' → (IsVecOver g ψ v' v'' ↔ IsVecOver (g.comp f) (ψ ≫ φ) v v'')

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PrincipalRootReZeroImSq

theorem **coe_add'** (a b : 𝒪 E) : ((a + b : 𝒪 E) : E) = a + b

theorem **coe_mul'** (a b : 𝒪 E) : ((a * b : 𝒪 E) : E) = a * b

theorem **exists_clear** (𝔞 : Submodule (𝒪 E) E) (h : 𝔞.FG) : ∃ N : ℤ, N ≠ 0 ∧ ∃ ψ : 𝔞 →ₗ[𝒪 E] 𝒪 E, ∀ x, (ψ x : E) = N * (x : E)

@[simp] theorem **act_tmul** (a : 𝒪 E) (r : ℝ) (x : M) : act M a (r ⊗ₜ x) = r ⊗ₜ (a • x)

theorem **act_act** (a b : 𝒪 E) (v : ℝ ⊗[ℤ] M) : act M a (act M b v) = act M (a * b) v

theorem **act_natCast** (n : ℕ) (v : ℝ ⊗[ℤ] M) : act M (n : 𝒪 E) v = (n : ℝ) • v

theorem **act_neg** (a : 𝒪 E) (v : ℝ ⊗[ℤ] M) : act M (-a) v = -act M a v

theorem **eq_act** (a : 𝒪 E) (A : ℝ ⊗[ℤ] M →ₗ[ℝ] ℝ ⊗[ℤ] M) (hA : ∀ x, A (1 ⊗ₜ x) = 1 ⊗ₜ (a • x)) : A = act M a

theorem **trace_act** (ψ : M →ₗ[𝒪 E] 𝒪 E) (hψ : Function.Injective ψ) (hr : Module.finrank ℤ M = Module.finrank ℤ (𝒪 E)) (a : 𝒪 E) : LinearMap.trace ℝ _ (act M a) = (Algebra.trace ℤ (𝒪 E) a : ℤ)

theorem **principal_root_re_zero_im_sq** (H : Host E) : H.rt0.re = 0 ∧ H.rt0.im ^ 2 = H.d

theorem **exists_complexStructure** (H : Host E) {M : Type} [AddCommGroup M] [Module (𝒪 E) M] (ψ : M →ₗ[𝒪 E] 𝒪 E) (hψ : Function.Injective ψ) (hr : Module.finrank ℤ M = Module.finrank ℤ (𝒪 E)) (EF : LinearMap.BilinForm ℤ M) (adj : ∀ (a : 𝒪 E) x y, EF (a • x) y = EF x (H.bar a • y)) (c : ℤ) (hc : 0 < (c : ℝ) * H.rt0.im) (hEF : ∀ x y, c * EF (H.rt • x) y = trForm H (ψ x) (ψ y)) : ∃ J : ℝ ⊗[ℤ] M →ₗ[ℝ] ℝ ⊗[ℤ] M, (∀ x, J (J x) = -x) ∧ (∀ x y, EF.baseChange ℝ (J x) (J y) = EF.baseChange ℝ x y) ∧ (∀ x, x ≠ 0 → 0 < EF.baseChange ℝ (J x) x) ∧ ∀ (a : 𝒪 E) (A : ℝ ⊗[ℤ] M →ₗ[ℝ] ℝ ⊗[ℤ] M) (t : ℂ), (∀ x : M, A (1 ⊗ₜ x) = 1 ⊗ₜ (a • x)) → t = ∑ τ ∈ Finset.univ.filter (fun τ : E →+* ℂ => τ (H.rt : E) = H.rt0), (((1 : ℕ) : ℂ) * τ (a : E) + ((0 : ℕ) : ℂ) * starRingEnd ℂ (τ (a : E))) → J ∘ₗ A = A ∘ₗ J ∧ LinearMap.trace ℝ _ A = 2 * t.re ∧ LinearMap.trace ℝ _ (J ∘ₗ A) = -2 * t.im

theorem **exists_datum** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (𝔞 : Submodule (𝒪 E) E), 𝔞.FG → 𝔞 ≠ ⊥ → (∀ y : E, y ∈ 𝔞 ↔ ∀ x ∈ 𝔞, ∃ n : ℤ, Algebra.trace ℚ E (x * IsCMField.complexConj E y / (H.rt : E)) = n) → ∃ (D : Datum H) (e : D.La ≃ₗ[𝒪 E] 𝔞), D.p = 1 ∧ D.q = 0 ∧ ∀ x y, ((D.EF x y : ℚ) : ℝ) * H.rt0.im = ((Algebra.trace ℚ E ((e x : E) * IsCMField.complexConj E (e y : E) / (H.rt : E)) : ℚ) : ℝ) * √(H.d)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PrincipalTripleExtendsOfCurveHomomorphisms

theorem **comp_prod** {T B A : Over S} [GrpObj A] [IsCommMonObj A] {I : Type} (s : Finset I) (x : T ⟶ B) (f : I → (B ⟶ A)) : x ≫ ∏ i ∈ s, f i = ∏ i ∈ s, x ≫ f i

theorem **prod_comp** {T B A : Over S} [GrpObj A] [IsCommMonObj A] [GrpObj B] [IsCommMonObj B] {I : Type} (s : Finset I) (x : I → (T ⟶ B)) (f : B ⟶ A) [IsMonHom f] : (∏ i ∈ s, x i) ≫ f = ∏ i ∈ s, x i ≫ f

theorem **exists_hom_pow** {G B A : Over S} [GrpObj G] [GrpObj A] [IsCommMonObj A] [GrpObj B] [IsCommMonObj B] {n : ℕ} (p : Out.IsPowGrp G B n) (ψ : Fin n → (B ⟶ A)) (hψ : ∀ i, IsMonHom (ψ i)) {k : Type} [Field k] (t : Sp k ⟶ S) (h : OntoAt ψ t) : ∃ Ψ : G ⟶ A, IsMonHom Ψ ∧ ∀ y : Over.mk t ⟶ A, ∃ x : Over.mk t ⟶ G, y = x ≫ Ψ

theorem **exists_extension** : ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (σ : K →+* ℂ) (P : Out.PolAV (Sp R)) (P₀ : Out.PolAV (Sp K)) (φ : P₀.A.left ⟶ P.A.left) (T₀ : PAT H (Sp K)) (ψ : Fin T₀.g → (P₀.A ⟶ T₀.t.G)), IsPolFibre (algebraMap R K) P P₀ φ → P.g = 1 → (∀ i, IsMonHom (ψ i)) → OntoAt ψ (spMap σ) → ∃ (T : PAT H (Sp R)) (_ : SABMap T₀.t T.t (spMap (algebraMap R K))), T.g = T₀.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PrincipalTripleExtendsOfCurveHomomorphismsStatement

theorem **_root_.GlobalTheory.T.principal_triple_extends_of_curve_homomorphisms** : ThetaUnifAV → Out.Laws → Out.Laws2 → HomExtDvr → Published.KernelFiniteFlatSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (σ : K →+* ℂ) (P : Out.PolAV (Sp R)) (P₀ : Out.PolAV (Sp K)) (φ : P₀.A.left ⟶ P.A.left) (T₀ : PAT H (Sp K)) (ψ : Fin T₀.g → (P₀.A ⟶ T₀.t.G)), IsPolFibre (algebraMap R K) P P₀ φ → P.g = 1 → (∀ i, IsMonHom (ψ i)) → OntoAt ψ (spMap σ) → ∃ (T : PAT H (Sp R)) (_ : SABMap T₀.t T.t (spMap (algebraMap R K))), T.g = T₀.g

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PrincipalTriplePullback

theorem **pat_pullback** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R R' : Type) [CommRing R] [CommRing R'] (f : R →+* R') (T : PAT H (Sp R)), (∃ (T' : PAT H (Sp R')) (_ : SABMap T'.t T.t (spMap f)), T'.g = T.g) ∧ ∀ (T' : PAT H (Sp R')) (m : SABMap T'.t T.t (spMap f)), Nontrivial R' → T'.g = T.g ∧ IsPolFibre f T.pol T'.pol m.φ ∧ ∃ w : LB.Iso T'.t.PP.L (T.t.PP.L.pull m.φ), RigIso m w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PrincipalTriplePullbackStatement

theorem **_root_.GlobalTheory.T.principalTriple_pullback** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R R' : Type) [CommRing R] [CommRing R'] (f : R →+* R') (T : PAT H (Sp R)), (∃ (T' : PAT H (Sp R')) (_ : SABMap T'.t T.t (spMap f)), T'.g = T.g) ∧ ∀ (T' : PAT H (Sp R')) (m : SABMap T'.t T.t (spMap f)), Nontrivial R' → T'.g = T.g ∧ IsPolFibre f T.pol T'.pol m.φ ∧ ∃ w : LB.Iso T'.t.PP.L (T.t.PP.L.pull m.φ), RigIso m w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ProdSurjectiveGeometricPoints

theorem **surjective_of_points** {k K : Type} [Field k] [Field K] [IsAlgClosed K] (σ : k →+* K) {P A : Scheme.{0}} (a : A ⟶ Sp k) [LocallyOfFiniteType a] (ψ : P ⟶ A) (hψ : IsClosed (Set.range ψ)) (h : ∀ x : Sp K ⟶ A, x ≫ a = spMap σ → ∃ p : Sp K ⟶ P, x = p ≫ ψ) : Surjective ψ

theorem **_root_.GlobalTheory.GeomSurj.exists_lift_of_surjective** {Ω : Type} [Field Ω] [IsAlgClosed Ω] {P A : Scheme.{0}} (ψ : P ⟶ A) [LocallyOfFiniteType ψ] [Surjective ψ] (x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp Ω ⟶ A) : ∃ p : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp Ω ⟶ P, x = p ≫ ψ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PtHomEqPtHomIffProportional

theorem **ptHom_eq_ptHom_iff_proportional** (R : Type) [CommRing R] (W : WeierstrassCurve R) (A : Type) [CommRing A] (ρ : R →+* A) (P Q : Fin 3 → A) (hP : IsUnimodularSol (W.map ρ) P) (hQ : IsUnimodularSol (W.map ρ) Q) : ptHom W ρ P hP = ptHom W ρ Q hQ ↔ Proportional P Q

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.PureTypeVectorSpansQuadraticLattice

theorem **_root_.GlobalTheory.PureTypeVector.hermForm_smul_smul** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z w : ℂ) (u v : Fin g → ℂ) : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm (z • u) (w • v) = z * starRingEnd ℂ w * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm u v

theorem **_root_.GlobalTheory.PureTypeVector.hermForm_self_im** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (v : Fin g → ℂ) : (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm v v).im = 0

theorem **_root_.GlobalTheory.pureType_vector_spans_quadratic_lattice** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (D : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum H) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pe : D.La →+ (Fin g → ℂ)) (Hm : Matrix (Fin g) (Fin g) ℂ), _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.IsPeriod D ty Pe → Hm.IsHermitian → (∀ z : Fin g → ℂ, z ≠ 0 → 0 < (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm z z).re) → (∀ x y, (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm (Pe x) (Pe y)).im = D.EF x y) → ∀ (ρ : ℂ) (a : D.La), ρ ^ 2 = -(H.d : ℂ) → (∀ i, ty i (H.rt : E) ≠ ρ → Pe a i = 0) → (∀ m n : ℤ, ((m : ℂ) + n * ((1 + Complex.I * Real.sqrt H.d) / 2)) • Pe a ∈ Pe.range) ∧ ∀ z w : ℂ, _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.hermForm Hm (z • Pe

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.QuadraticEquationOfDerivative

theorem **isLieComb_natCast_pow** {v : epsR R ⟶ G} (hv : IsTangent v) (n : ℕ) : IsLieComb ![(n : R)] ![v] (v ^ n)

theorem **isMonHom_id_pow** [IsCommMonObj G] (n : ℕ) : IsMonHom ((𝟙 G) ^ n)

theorem **quadratic_equation_of_derivative** : ThetaUnifAV → Out.PolBaseChangeSt → ∀ (k : Type) [Field k] (s : k →+* ℂ) (P : Out.PolAV (Sp k)) (v : epsR k ⟶ P.A) (W : P.A ⟶ P.A) (w : k) (n : ℕ), IsTangent v → (∀ (R' : Type) [CommRing R'] (f' : k →+* R') (P' : Out.PolAV (Sp R')) (φ' : P'.A.left ⟶ P.A.left) (v' : epsR R' ⟶ P'.A), IsPolFibre f' P P' φ' → IsVecOver f' φ' v v' → IsLieBasis ![v']) → IsMonHom W → IsLieComb ![w] ![v] (v ≫ W) → w * w + n = w → (W ≫ W) = (𝟙 P.A) ^ n = W

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RangeEqValuationRingIffIsLocalization

theorem **range_eq_valuationRing_iff_isLocalization** : ∀ (K : Type) [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)) (R : Type) [CommRing R] [IsDomain R] [Algebra R K] [IsFractionRing R K] [Algebra (𝒪 K) R] [IsScalarTower (𝒪 K) R K], (∀ y : K, y ∈ Set.range (algebraMap R K) ↔ p.valuation K y ≤ 1) ↔ IsLocalization p.asIdeal.primeCompl R

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RangeSubsetAbLocusOfIsProper

theorem **valuative_lift_of_isProper** {V K : Type} [CommRing V] [IsDomain V] [ValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] {X S : Scheme.{0}} (s : X ⟶ S) [IsProper s] (y : Sp K ⟶ X) (r : Sp V ⟶ S) (h : y ≫ s = spMap (algebraMap V K) ≫ r) : ∃ x : Sp V ⟶ X, spMap (algebraMap V K) ≫ x = y ∧ x ≫ s = r

theorem **range_subset_abLocus_of_isProper** {S S' : Scheme.{0}} {G : Over S} {G' : Over S'} [GrpObj G] (hG : IsSemiAbelian G) {f : S' ⟶ S} {φ : G'.left ⟶ G.left} (h : IsPullback φ G'.hom G.hom f) [IsProper G'.hom] : Set.range f ⊆ abLocus G

theorem **model_point_extends** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (c : Coeff l) (O : Type) [CommRing O] (β : O →+* c.F) (Y : Model Xo c β) (V K : Type) [CommRing V] [IsDomain V] [ValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] (b : O →+* V) (y : Sp K ⟶ Xo.XF c), y ≫ Y.j ≫ Y.s = spMap (algebraMap V K) ≫ spMap b → ∃ (y' : Sp V ⟶ Y.X) (t : SAB H (Sp V)) (_ : SABMap t Y.t y'), y' ≫ Y.s = spMap b ∧ spMap (algebraMap V K) ≫ y' = y ≫ Y.j ∧ (IsAbelianScheme t.G → Set.range y' ⊆ abLocus Y.t.G)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ReEqZeroAndImSq

theorem **re_im_of_sq_eq_neg** {z : ℂ} {c : ℝ} (hc : 0 ≤ c) (hz : z ^ 2 = -(c : ℂ)) : z.re = 0 ∧ z.im ^ 2 = c

theorem **re_eq_zero_and_im_sq** {E : Type} [Field E] [NumberField E] [NumberField.IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : H.rt0.re = (0 : ℝ) ∧ H.rt0.im ^ (2 : ℕ) = (H.d : ℝ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RelAmpleLineBundleOfPositiveHermitianCmForm

theorem **isFinite_left_of_comp_eq_id** {S : Scheme.{0}} {A B : Over S} [IsSeparated B.hom] (i : A ⟶ B) (r : B ⟶ A) (h : i ≫ r = 𝟙 A) : IsFinite i.left

theorem **relAmple_of_kForm_pos** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsNoetherianRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (A : Over (Sp R)) [GrpObj A] [IsCommMonObj A] (b : OAct (OK H) A) (t : Serre.IsTensor A b C.P.A C.act D.La) (N : LB A.left), IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → N.IsLB → (∃ t : LB.Gen (Theta A N), t.IsGen) → LBIso N (cubeLB A ![𝟙 A, b.act (wK H)] N) → (∀ m m', LamEq N (t.tmul m) (t.tmul m') C.P.L (C.act.act (h m m')) (𝟙 C.P.A)) → Out.RelAmple A N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RelAmpleLineBundleOfPositiveHermitianCmFormStatement

theorem **_root_.GlobalTheory.T.relAmple_lineBundle_of_positive_hermitian_cm_form** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (R : Type) [CommRing R] [IsNoetherianRing R] (ω : R) (C : Ell H R ω) (D : Datum H) (ε : ℤˣ) (h : D.La → D.La → OK H) (A : Over (Sp R)) [GrpObj A] [IsCommMonObj A] (b : OAct (OK H) A) (t : Serre.IsTensor A b C.P.A C.act D.La) (N : LB A.left), IsKForm D ε h → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) * D.EF (H.w0 • x) x) → N.IsLB → (∃ t : LB.Gen (Theta A N), t.IsGen) → LBIso N (cubeLB A ![𝟙 A, b.act (wK H)] N) → (∀ m m', LamEq N (t.tmul m) (t.tmul m') C.P.L (C.act.act (h m m')) (𝟙 C.P.A)) → Out.RelAmple A N

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RelAmpleOfGenericIso

theorem **isPullCubical_of_lbIso** (hG : Out.GlobalFnsSt) {S S' : Scheme.{0}} {G' : Over S'} {G : Over S} [GrpObj G'] [IsCommMonObj G'] [GrpObj G] [IsCommMonObj G] (hA : IsAbelianScheme G') {f : S' ⟶ S} {α : G'.left ⟶ G.left} (gp : IsGrpPullback G' G f α) {L' : RigLB G'} {L : RigLB G} (c' : CubicalStr G' L') (c : CubicalStr G L) (h : LBIso L'.L (L.L.pull α)) : (cubeMapOf gp).IsPullCubical c' c

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RelAmpleOfGenericIsogenousDedekind

theorem **range_spMap_fractionRing** (R K : Type) [CommRing R] [IsDomain R] [Field K] [Algebra R K] [IsFractionRing R K] : Set.range (spMap (algebraMap R K)) = {genericPoint (Sp R)}

theorem **relAmple_of_generic_of_isogenous** : Out.Laws → Out.Laws2 → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (P : Out.PolAV (Sp R)) (B : Over (Sp R)) [GrpObj B] (Ψ : P.A ⟶ B) (M : LB B.left) (B₀ : Over (Sp K)) [GrpObj B₀] (φ : B₀.left ⟶ B.left) (M₀ : LB B₀.left), IsAbelianScheme B → M.IsLB → M₀.IsLB → IsFinite Ψ.left → Flat Ψ.left → Surjective Ψ.left → IsGrpPullback B₀ B (spMap (algebraMap R K)) φ → LBIso M₀ (M.pull φ) → Out.RelAmple B₀ M₀ → Out.RelAmple B M

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RelAmpleOfGenericIsogenousDedekindStatement

theorem **_root_.GlobalTheory.T.relAmple_of_generic_isogenous_dedekind** : Out.Laws → Out.Laws2 → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (P : Out.PolAV (Sp R)) (B : Over (Sp R)) [GrpObj B] (Ψ : P.A ⟶ B) (M : LB B.left) (B₀ : Over (Sp K)) [GrpObj B₀] (φ : B₀.left ⟶ B.left) (M₀ : LB B₀.left), IsAbelianScheme B → M.IsLB → M₀.IsLB → IsFinite Ψ.left → Flat Ψ.left → Surjective Ψ.left → IsGrpPullback B₀ B (spMap (algebraMap R K)) φ → LBIso M₀ (M.pull φ) → Out.RelAmple B₀ M₀ → Out.RelAmple B M

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RigidifiedOfChart

theorem **res_pullU_eq** (f : Y ⟶ X) (U : X.Opens) (V : Y.Opens) (h : V ≤ f ⁻¹ᵁ U) (x : Γ(X, U)ˣ) : res h (pullU f U x) = Units.map (f.appLE U V h).hom.toMonoidHom x

theorem **res_res_pullU** (f : Y ⟶ X) (U : X.Opens) (V W : Y.Opens) (h : V ≤ f ⁻¹ᵁ U) (h' : W ≤ V) (x : Γ(X, U)ˣ) : res h' (res h (pullU f U x)) = res (h'.trans h) (pullU f U x)

theorem **res_pullU_res** (f : Y ⟶ X) (U U' : X.Opens) (V : Y.Opens) (h : V ≤ f ⁻¹ᵁ U') (h' : U' ≤ U) (h'' : V ≤ f ⁻¹ᵁ U) (x : Γ(X, U)ˣ) : res h (pullU f U' (res h' x)) = res h'' (pullU f U x)

theorem **exists_gen_pull** (L : LB X) (f : Y ⟶ X) (i : L.ι) (hL : L.IsLB) (hi : ⊤ ≤ f ⁻¹ᵁ L.U i) : ∃ t : LB.Gen (L.pull f), t.IsGen

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RigidifiedSectionCorrespondenceExistsUniqueNonva

theorem **exists_sec_back** {L : LB X} {L' : LB X'} {F : X' ⟶ X} {w : LB.Iso L' (L.pull F)} (hw : PGen w) (hL : L.IsLB) (hL' : L'.IsLB) (s : LB.Sec (L.pull F)) : ∃ σ : LB.Sec L', ∀ (Z : Scheme.{0}) (r : Z ⟶ X') (z : Z) (J : PUnit → L.ι) (k : L'.ι), r z ∈ (L.pull F).U J → r z ∈ L'.U k → gs r z _ (s.f J) = (gw w r z (J PUnit.unit) k).val ∗ gs r z _ (σ.f k)

theorem **chart_le_preimage'** (L : LB X) {F : X' ⟶ X} {φ : Y ⟶ X} (g : X' ⟶ Y) (e : g ≫ φ = F) (J : PUnit → L.ι) : (L.pull F).U J ≤ g ⁻¹ᵁ LB.wchart L (fun _ : Fin 1 => φ) (fun _ => J PUnit.unit)

theorem **isoMaps_iff** {L : LB X} {L' : LB X'} {F : X' ⟶ X} (w : LB.Iso L' (L.pull F)) (σ : LB.Sec L') (s : LB.Sec (L.pull F)) : IsoMaps w σ s ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ X') (z : Z) (J : PUnit → L.ι) (k : L'.ι), r z ∈ (L.pull F).U J → r z ∈ L'.U k → gs r z _ (s.f J) = (gw w r z (J PUnit.unit) k).val ∗ gs r z _ (σ.f k)

theorem **secTie_exists_unique** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x : Sp S ⟶ Xo.X) (t' : SAB H (Sp S)) (m : SABMap t' Xo.univ.t x), (∀ th, ∃ s : LB.Sec (PPx x), SecTie m th s) ∧ (∀ s : LB.Sec (PPx x), ∃ th, SecTie m th s) ∧ ∀ th (s : LB.Sec (PPx x)), SecTie m th s → (∀ th' (s' : LB.Sec (PPx x)), SecTie m th' s' → ((∀ i, th'.f i = th.f i) ↔ ∀ i, s'.f i = s.f i)) ∧ (SecNZ x s ↔ ∀ p : Sp S, ∃ q : t'.G.left, t'.G.hom q = p ∧ q ∈ Out.nonzeroLocus th)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RigidifiedSectionCorrespondenceExistsUniqueNonvaStatement

theorem **_root_.GlobalTheory.T.rigidifiedSectionCorrespondence_existsUnique_nonvanishing** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x : Sp S ⟶ Xo.X) (t' : SAB H (Sp S)) (m : SABMap t' Xo.univ.t x), (∀ th, ∃ s : LB.Sec (PPx x), SecTie m th s) ∧ (∀ s : LB.Sec (PPx x), ∃ th, SecTie m th s) ∧ ∀ th (s : LB.Sec (PPx x)), SecTie m th s → (∀ th' (s' : LB.Sec (PPx x)), SecTie m th' s' → ((∀ i, th'.f i = th.f i) ↔ ∀ i, s'.f i = s.f i)) ∧ (SecNZ x s ↔ ∀ p : Sp S, ∃ q : t'.G.left, t'.G.hom q = p ∧ q ∈ Out.nonzeroLocus th)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RingActIffOAct

theorem **_root_.GlobalTheory.LamF.ringAct_iff_oAct** : ∀ (O : Type) [CommRing O] (S : Scheme.{0}) (G : Over S) [GrpObj G] [IsCommMonObj G], (∀ a : _root_.ReciprocityTheory.ChartRestriction.G.Out.RingAct G O, ∃ b : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.OAct O G, b.act = a.act) ∧ ∀ b : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.OAct O G, ∃ a : _root_.ReciprocityTheory.ChartRestriction.G.Out.RingAct G O, a.act = b.act

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RingHomExtRingOfIntegers

theorem **ringHom_ext_ringOfIntegers** {E K : Type} [Field E] [NumberField E] [Field K] {f g : E →+* K} (h : ∀ a : 𝒪 E, f (a : E) = g (a : E)) : f = g

theorem **forall_apply_eq_smul_iff** {K O : Type} [Field K] {n : ℕ} (r : O → Module.End K (Fin n → K)) (χ : Fin n → O → K) (hχ : Function.Injective χ) (hr : ∀ a j, r a (Pi.single j 1) = Pi.single j (χ j a)) (i : Fin n) (c : Fin n → K) : (∀ a, r a c = χ i a • c) ↔ ∃ s : K, c = Pi.single i s

theorem **mem_eigLat_iff** : ∀ (E : Type) [Field E] [NumberField E] [StarRing (𝒪 E)] (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (oa : A.OAct (𝒪 E)) (act : OAct (𝒪 E) A.A) (S : Set (E →+* ℂ)) (τ : Fin x.P.g ≃ S) (lab : (E →+* ℂ) → (E →+* K)), act.act = oa.act → Function.Injective lab → IsLieType A oa x φ τ lab → ∀ (i : Fin x.P.g) (t : 𝒪 E →+* 𝒪 K), (∀ a : 𝒪 E, (t a : K) = lab (τ i) (a : E)) → ∀ c, c ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).eigLat act t ↔ c ∈ (GenFib.mk x.P φ hf x.v x.basis : GenFib A).lieLat ∧ ∃ r : K, c = Pi.single i r

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RingOfIntegersActionInjectiveOnAbelianScheme

theorem **act_zero** (act : OAct O G) : act.act 0 = 1

theorem **act_natCast** (act : OAct O G) (n : ℕ) : act.act (n : O) = 𝟙 G ^ n

theorem **act_mul_eq_one** (act : OAct O G) {c : O} (hc : act.act c = 1) (d : O) : act.act (d ∗ c) = 1

theorem **pow_eq_one_of_act** (act : OAct O G) {n : ℕ} (hn : act.act (n : O) = 1) {T : Over S} (x : T ⟶ G) : x ^ n = 1

theorem **act_injective** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (act : OAct (𝒪 E) P.A), 1 ≤ P.g → Function.Injective act.act

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RingOfIntegersActionInjectiveOnAbelianSchemeStatement

theorem **_root_.GlobalTheory.T.ringOfIntegersAction_injective_on_abelianScheme** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] (K : Type) [Field K] [NumberField K] (P : Out.PolAV (Sp (𝒪 K))) (act : OAct (𝒪 E) P.A), 1 ≤ P.g → Function.Injective act.act

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RingOfIntegersSqrtNegThreeModFour

private theorem **norm_sq_add_norm_sq_sub_eq_complete_square** {K : Type*} [Field K] [NumberField K] {d : ℕ} (h2 : finrank ℚ K = 2) (hd : d % 4 = 3) {r : K} (hr : r ^ 2 = -d) : (∀ x : K, ∃ a b : ℚ, x = a + b * r) ∧ ∃ τ : K →+* K, τ r = -r

private theorem **exists_int_eq_of_isIntegral_rat** {K : Type*} [Field K] [CharZero K] (q : ℚ) (h : IsIntegral ℤ (q : K)) : ∃ z : ℤ, q = z

private theorem **exists_int_eq_of_squarefree_mul_sq_eq_int** {d : ℕ} (hsq : Squarefree d) (q : ℚ) (M : ℤ) (h : (d : ℚ) * q ^ 2 = M) : ∃ z : ℤ, q = z

theorem **exists_integer_generator_of_sq_eq_neg** {K : Type*} [Field K] [NumberField K] {d : ℕ} (h2 : finrank ℚ K = 2) (hsq : Squarefree d) (hd : d % 4 = 3) {r : K} (hr : r ^ 2 = -d) : ∃ w : 𝒪 K, 2 * (w : K) = 1 + r ∧ ∀ c : 𝒪 K, ∃ m n : ℤ, c = m + n * w

theorem **eq_I_mul_sqrt_or_of_sq_eq_neg** {d : ℕ} {z : ℂ} (hz : z ^ 2 = -d) : z = Complex.I * √d ∨ z = -(Complex.I * √d)

theorem **ringOfIntegers_sqrt_neg_three_mod_four** (d : ℕ) (hsq : Squarefree d) (hd : d % 4 = 3) (K : Type) [Field K] [NumberField K] (h2 : Module.finrank ℚ K = 2) (r : K) (hr : r ^ 2 = -(d : K)) : ∃ (w : 𝒪 K) (bar : 𝒪 K →+* 𝒪 K), 2 * (w : K) = 1 + r ∧ w + bar w = 1 ∧ (∀ c : 𝒪 K, ∃ m n : ℤ, c = m + n * w) ∧ (∀ σ : K →+* ℂ, ¬ ComplexEmbedding.IsReal σ ∧ ∀ c : 𝒪 K, σ (bar c : K) = starRingEnd ℂ (σ (c : K))) ∧ ∃ σ : K →+* ℂ, σ r = Complex.I * √d

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RootOfUnityEqOneOfConjCongruence

theorem **isIntegral_eigenvalue_sub_one_div** {E : Type} [Field E] [NumberField E] {r M : ℕ} (Λ : Submodule (𝒪 E) (Fin r → E)) (γ : Matrix (Fin r) (Fin r) E) (τ : E →+* ℂ) (hM : (M : ℂ) ≠ 0) (hfg : Λ.FG) (hsp : Submodule.span E (Λ : Set (Fin r → E)) = ⊤) (hγ : ∀ x ∈ Λ, ∃ y ∈ Λ, γ *ᵥ x - x = (M : E) • y) (w : Fin r → ℂ) (hw : w ≠ 0) (l : ℂ) (hl : w ᵥ* γ.map τ = l • w) : IsIntegral ℤ ((l - 1) / M)

theorem **root_of_unity_eq_one_of_isIntegral_div** {E : Type} [Field E] [NumberField E] {M n : ℕ} (hM : 3 ≤ M) (hn : 0 < n) {u : E} (hu : u ^ n = 1) (h : IsIntegral ℤ ((u - 1) / M)) : u = 1

theorem **root_of_unity_eq_one_of_conj_congruence** : ∀ (E : Type) [Field E] [NumberField E] (r M n : ℕ) (Λ Λ' : Submodule (𝒪 E) (Fin r → E)) (γ δ : Matrix (Fin r) (Fin r) E) (u : E) (τ : E →+* ℂ) (Q : Matrix (Fin r) (Fin r) ℂ), 0 < r → 3 ≤ M → 0 < n → u ^ n = 1 → Λ.FG → Λ'.FG → Submodule.span E (Λ : Set (Fin r → E)) = ⊤ → Submodule.span E (Λ' : Set (Fin r → E)) = ⊤ → (∀ x ∈ Λ, ∃ y ∈ Λ, Matrix.mulVec (γ : Matrix (Fin r) (Fin r) E) x - x = (M : E) • y) → (∀ x ∈ Λ', ∃ y ∈ Λ', Matrix.mulVec (δ : Matrix (Fin r) (Fin r) E) x - x = (M : E) • y) → Q.det ≠ 0 → (u • γ).map τ * Q = Q * δ.map τ → u = 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RosatiAdjIffLamEq

theorem **form_rosati** {S : Scheme.{0}} {G : Over S} [GrpObj G] [IsCommMonObj G] (f g : G ⟶ G) {V : Type} [CommGroup V] (k : ((G ⊗ G).left ⟶ G.left) → V) : (∏ I : ULift.{0} (Finset (Fin 2)), k (psum G ![fst G G ≫ f, snd G G] I.down).left ^ ((-1 : ℤ) ^ (2 - I.down.card))) / (∏ I : ULift.{0} (Finset (Fin 2)), k (psum G ![fst G G, snd G G ≫ g] I.down).left ^ ((-1 : ℤ) ^ (2 - I.down.card))) = ∏ i, k (Out.rosatiφ G f g i) ^ Out.rosatiε i

theorem **lbIso_of_gen** {S : Scheme.{0}} {G : Over S} [GrpObj G] [IsCommMonObj G] {L : LB G.left} (hL : L.IsLB) (f g : G ⟶ G) (t : LB.Gen (LB.LinComb L (Out.rosatiφ G f g) Out.rosatiε)) (ht : t.IsGen) : LBIso (cubeLB G ![fst G G ≫ f, snd G G] L) (cubeLB G ![fst G G, snd G G ≫ g] L)

theorem **rosatiAdj_iff_lamEq** : Out.GlobalFnsSt → ∀ (S : Scheme.{0}) (P : Out.PolAV S) (f g : P.A ⟶ P.A), IsMonHom f → IsMonHom g → (Out.RosatiAdj P.A P.L f g ↔ LamEq P.L f (𝟙 P.A) P.L (𝟙 P.A) g)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RuleValueIntegralOfSupLatticeValuesNonneg

theorem **mem_range_of_nval_nonneg** {V K' : Type} [CommRing V] [IsDomain V] [IsDiscreteValuationRing V] [Field K'] [Algebra V K'] [IsFractionRing V K'] {e : ℕ} (he : 0 < e) {x : K'} (h : 0 ≤ Formalism.L2.Pj.nval V e x) : x ∈ Set.range (algebraMap V K')

theorem **evR_mem_range_of_valSet_nonneg** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (F : Rul Xo c), (∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v F) → ∀ (T : CDVR) (io : c.F →+* T.Fr), T.FinRes → letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.RuleValueIntegralOfSupLatticeValuesNonnegStatement

theorem **_root_.GlobalTheory.T.ruleValue_integral_of_supLattice_values_nonneg** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (F : Rul Xo c), (∀ la, ∀ v ∈ valSet Xo c (algCls Xo c) la, 0 ≤ v F) → ∀ (T : CDVR) (io : c.F →+* T.Fr), T.FinRes → letI

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SabMapCancelAndCompOver

theorem **sabMap_cancel_and_comp_over** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (S S₁ S₂ : Scheme.{0}) (t : SAB H S) (t₁ : SAB H S₁) (t₂ : SAB H S₂) (f₁ : S₁ ⟶ S) (f₂ : S₂ ⟶ S) (g : S₂ ⟶ S₁) (m₁ : SABMap t₁ t f₁) (m₂ : SABMap t₂ t f₂), f₂ = g ≫ f₁ → (∃ n : SABMap t₂ t₁ g, n.φ ≫ m₁.φ = m₂.φ) ∧ ∀ n : SABMap t₂ t₁ g, ∃ m : SABMap t₂ t f₂, m.φ = n.φ ≫ m₁.φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SecTieBaseChange

theorem **isoMaps_iff** {X X' : Scheme.{0}} {L : LB X} {L' : LB X'} {F : X' ⟶ X} (w : LB.Iso L' (L.pull F)) (σ : LB.Sec L') (s : LB.Sec (L.pull F)) : IsoMaps w σ s ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ X') (z : Z) (j : L.ι) (k : L'.ι), F (r z) ∈ L.U j → r z ∈ L'.U k → gs r z _ (s.f (fun _ => j)) = (gw w r z j k).val * gs r z _ (σ.f k)

theorem **exists_canon** : ∃ g : t.G.left ⟶ Gx x, g ≫ (snd (Over.mk x) Xo.univ.t.G).left = m.φ ∧ g ≫ (fst (Over.mk x) Xo.univ.t.G).left = t.G.hom

theorem **canon_mem** {g : t.G.left ⟶ Gx x} (hg : g ≫ (snd (Over.mk x) Xo.univ.t.G).left = m.φ) (j : Xo.univ.t.PP.L.ι) (y : t.G.left) (hy : m.φ y ∈ Xo.univ.t.PP.L.U j) : g y ∈ (PPx x).U (fun _ => j)

theorem **rigIso_unique** {w w' : LB.Iso t.PP.L (Xo.univ.t.PP.L.pull m.φ)} (h : RigIso m w) (h' : RigIso m w') : w = w'

theorem **secTie_iff** {w : LB.Iso t.PP.L (Xo.univ.t.PP.L.pull m.φ)} (hw : RigIso m w) {g : t.G.left ⟶ Gx x} (hs : g ≫ (snd (Over.mk x) Xo.univ.t.G).left = m.φ) (hf : g ≫ (fst (Over.mk x) Xo.univ.t.G).left = t.G.hom) (th : LB.Sec t.PP.L) (s : LB.Sec (PPx x)) : SecTie m th s ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ t.G.left) (z : Z) (j : Xo.univ.t.PP.L.ι) (k : t.PP.L.ι), m.φ (r z) ∈ Xo.univ.t.PP.L.U j → r z ∈ t.PP.L.U k → gs (r ≫ g) z ((PPx x).U (fun _ => j)) (s.f (fun _ => j)) = (gw w r z j k).val * gs r z _ (th.f k)

theorem **secTie_scale** {th th' : LB.Sec t.PP.L} {s s' : LB.Sec (PPx x)} (h : SecTie m th s) (a : S) (hs' : ∀ i, s'.f i = scal x a i ∗ s.f i) : SecTie m th' s' ↔ ∀ i, th'.f i = constFn t.G.hom _ a ∗ th.f i

theorem **secTie_baseChange** {th : LB.Sec t.PP.L} {s : LB.Sec (PPx x)} (h : SecTie m th s) {S' : Type} [CommRing S'] (φ : S →+* S') {x' : Sp S' ⟶ Xo.X} {t'' : SAB H (Sp S')} (m' : SABMap t'' Xo.univ.t x') (n : SABMap t'' t (spMap φ)) (s' : LB.Sec (PPx x')) (w : LB.Iso t''.PP.L (t.PP.L.pull n.φ)) (θ : LB.Sec (t.PP.L.pull n.φ)) (th' : LB.Sec t''.PP.L) (hx : x' = spMap φ ≫ x) (hφ : m'.φ = n.φ ≫ m.φ) (hb : SecBC φ 1 x x' s s') (hw : RigIso n w) (hθ : PullSec n.φ th θ) : SecTie m' th' s' ↔ IsoMaps w th' θ

theorem **secTie_calc** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x : Sp S ⟶ Xo.X) (t' : SAB H (Sp S)) (m : SABMap t' Xo.univ.t x) (th : LB.Sec t'.PP.L) (s : LB.Sec (PPx x)), SecTie m th s → (∀ (a : S) (x' : Sp S ⟶ Xo.X) (s' : LB.Sec (PPx x')) (th' : LB.Sec t'.PP.L), x' = x → SecBC (RingHom.id S) a x x' s s' → (SecTie m th' s' ↔ ∀ i, th'.f i = constFn t'.G.hom _ a ∗ th.f i)) ∧ ∀ (S' : Type) [CommRing S'] (φ : S →+* S') (x' : Sp S' ⟶ Xo.X) (t'' : SAB H (Sp S')) (m' : SABMap t'' Xo.univ.t x') (n : SABMap t'' t' (spMap φ)) (s' : LB.Sec (PPx x')) (w : LB.Iso t''.PP.L (t'.PP.L.pull n.φ)) (θ : LB.Sec (t'.PP.L.pull n.φ)) (th' : LB.Sec t''.PP.L), x' = spMap φ ≫ x → m'.φ = n.φ ≫ m.φ → SecBC φ 1 x x' s s' → RigIso n w → PullSec n.φ th θ → (SecTie m' th' s' ↔ IsoMaps w th' θ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SemiAbelianHomExtensionUnique

theorem **Agree.comp** {φ : X' ⟶ X} {f : X' ⟶ Y} {W W' : Scheme.{0}} {i : W ⟶ X} {g : W ⟶ Y} (h : Agree φ f i g) (c : W' ⟶ W) : Agree φ f (c ≫ i) (c ≫ g)

theorem **Agree.of_dominant** {φ : X' ⟶ X} [Flat φ] {f : X' ⟶ Y} (sX : X ⟶ S) (sY : Y ⟶ S) [IsSeparated sY] (hf : f ≫ sY = φ ≫ sX) {W W' : Scheme.{0}} {i : W ⟶ X} {g : W ⟶ Y} (hg : g ≫ sY = i ≫ sX) (c : W' ⟶ W) [QuasiCompact c] [IsSchemeTheoreticallyDominant c] (h : Agree φ f (c ≫ i) (c ≫ g)) : Agree φ f i g

theorem **exists_open_agree** [IsLocallyNoetherian X] {φ : X' ⟶ X} [Flat φ] {f : X' ⟶ Y} (sX : X ⟶ S) (sY : Y ⟶ S) [IsSeparated sY] [LocallyOfFiniteType sY] (hf : f ≫ sY = φ ≫ sX) (x : X) (g : Spec (X.presheaf.stalk x) ⟶ Y) (hg : g ≫ sY = X.fromSpecStalk x ≫ sX) (h : Agree φ f (X.fromSpecStalk x) g) : ∃ (W : Scheme.{0}) (i : W ⟶ X) (_ : IsOpenImmersion i) (g' : W ⟶ Y), x ∈ Set.range i ∧ g' ≫ sY = i ≫ sX ∧ Agree φ f i g'

theorem **exists_of_locally** {U : Scheme.{0}} {j : U ⟶ S} [QuasiCompact j] [IsSchemeTheoreticallyDominant j] {φ : X' ⟶ X} {p : X' ⟶ U} {f : X' ⟶ Y} (sX : X ⟶ S) [Flat sX] (sY : Y ⟶ S) [IsSeparated sY] (hp : IsPullback φ p sX j) (h : ∀ x : X, ∃ (W : Scheme.{0}) (i : W ⟶ X) (_ : IsOpenImmersion i) (g' : W ⟶ Y), x ∈ Set.range i ∧ g' ≫ sY = i ≫ sX ∧ Agree φ f i g') : ∃ F : X ⟶ Y, F ≫ sY = sX ∧ φ ≫ F = f

theorem **exists_of_stalks** [IsLocallyNoetherian X] {U : Scheme.{0}} {j : U ⟶ S} [QuasiCompact j] [Flat j] [IsSchemeTheoreticallyDominant j] {φ : X' ⟶ X} {p : X' ⟶ U} {f : X' ⟶ Y} (sX : X ⟶ S) [Flat sX] (sY : Y ⟶ S) [IsSeparated sY] [LocallyOfFiniteType sY] (hp : IsPullback φ p sX j) (hf : f ≫ sY = φ ≫ sX) (h : ∀ s : S, ∃ (T : Scheme.{0}) (t : T ⟶ S) (a : Spec (S.presheaf.stalk s) ⟶ T) (g : pullback sX t ⟶ Y), a ≫ t = S.fromSpecStalk s ∧ g ≫ sY = pullback.fst sX t ≫ sX ∧ Agree φ f (pullback.fst sX t) g) : ∃ F : X ⟶ Y, F ≫ sY = sX ∧ φ ≫ F = f

theorem **exists_spec_factor** (R T : Type) [CommRing R] [CommRing T] [Algebra R T] (M : Submonoid R) [IsLocalization M T] {A : CommRingCat.{0}} (ψ : CommRingCat.of R ⟶ A) (hu : ∀ y : M, IsUnit (ψ.hom y)) : ∃ a : Spec A ⟶ Sp T, a ≫ spMap (algebraMap R T) = Spec.map ψ

theorem **exists_fromSpecStalk_factor** (R T : Type) [CommRing R] [CommRing T] [Algebra R T] (M : Submonoid R) [IsLocalization M T] (s : PrimeSpectrum R) (hM : M ≤ s.asIdeal.primeCompl) : ∃ a : Spec ((Sp R).presheaf.stalk s) ⟶ Sp T, a ≫ spMap (algebraMap R T) = (Sp R).fromSpecStalk s

theorem **exists_agree_of_extend** {U T : Scheme.{0}} {k : U ⟶ T} {t : T ⟶ S} [Mono t] {j : U ⟶ S} (hj : k ≫ t = j) (G G' : Over S) [GrpObj G] [GrpObj G'] (f : (Over.pullback j).obj G ⟶ (Over.pullback j).obj G') [IsMonHom f] (h : ∀ f' : (Over.pullback k).obj ((Over.pullback t).obj G) ⟶ (Over.pullback k).obj ((Over.pullback t).obj G'), IsMonHom f' → ∃ F, (Over.pullback k).map F = f') : ∃ g : pullback G.hom t ⟶ G'.left, g ≫ G'.hom = pullback.fst G.hom t ≫ G.hom ∧ Agree (pullback.fst G.hom j) (f.left ≫ pullback.fst G'.hom j) (pullback.fst G.hom t) g

theorem **abelianScheme_hom_extends_dedekind** : (∀ (T K : Type) [CommRing T] [IsDomain T] [IsDiscreteValuationRing T] [Field K] [Algebra T K] [IsFractionRing T K] (G G' : Over (Sp T)) [GrpObj G] [GrpObj G'], IsSemiAbelian G → IsSemiAbelian G' → ∀ f : (Over.pullback (spMap (algebraMap T K))).obj G ⟶ (Over.pullback (spMap (algebraMap T K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap T K))).map F = f) → ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (G G' : Over (Sp R)) [GrpObj G] [GrpObj G'], IsAbelianScheme G → IsAbelianScheme G' → ∀ f : (Over.pullback (spMap (algebraMap R K))).obj G ⟶ (Over.pullback (spMap (algebraMap R K))).obj G', IsMonHom f → ∃! F : G ⟶ G', IsMonHom F ∧ (Over.pullback (spMap (algebraMap R K))).map F = f

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SemicharacterSqEqOneOfSymmetric

lemma **valueRules** : ValueRules

lemma **ex_neg_left** (z : Fin g → ℂ) : (u.ex z).left ≫ (GrpObj.inv : A ⟶ A).left = (u.ex (-z)).left

lemma **exists_chart** (hv : ValueRules) (hL : L.IsLB) (z : Fin g → ℂ) : ∃ i, ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U i

lemma **mul_neg_eq_one** (hv : ValueRules) (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (h : u.IsAH Hm χ) (l : u.Λ) : χ l ∗ χ (-l) = 1

lemma **exists_symmetry_values** (hv : ValueRules) (hs : Out.Sym A L) : ∃ T : L.ι → L.ι → (Fin g → ℂ) → ℂ, (∀ i j, DifferentiableOn ℂ (T i j) {z | ⊤ ≤ (u.ex (-z)).left ⁻¹ᵁ L.U i ∧ ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U j}) ∧ (∀ i j z, ⊤ ≤ (u.ex (-z)).left ⁻¹ᵁ L.U i → ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U j → T i j z ≠ 0) ∧ (∀ i j i' j' z, ⊤ ≤ (u.ex (-z)).left ⁻¹ᵁ L.U i → ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U j → ⊤ ≤ (u.ex (-z)).left ⁻¹ᵁ L.U i' → ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U j' → T i j z = valAt' (u.ex (-z)).left (L.U i ⊓ L.U i') (L.g i i').val ∗ (valAt' (u.ex z).left (L.U j ⊓ L.U j') (L.g j j').val)⁻¹ ∗ T i' j' z) ∧ ∀ i j z (l : u.Λ), T i j (z + l) = T i j z

lemma **neg_eq** (hv : ValueRules) (hL : L.IsLB) (hs : Out.Sym A L) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (h : u.IsAH Hm χ) (l : u.Λ) : χ (-l) = χ l

theorem **semicharacter_sq_eq_one_of_symmetric** : ∀ (P : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g) (Hm : Matrix (Fin P.g) (Fin P.g) ℂ) (χ : u.Λ → ℂ), u.IsAH Hm χ → ∀ l, χ l ^ 2 = 1

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SingleMemSpanIffOfSelfDual

theorem **single_mem_span_iff_of_selfDual** : ∀ (O R K La I : Type) [CommRing O] [CommRing R] [Field K] [Algebra R K] [AddCommGroup La] [Module O La] [Fintype I] [DecidableEq I] (τ₀ : O →+* R) (h : La → La → O) (ev dv : La → I → K) (n : ℕ) (x y : Fin n → La), (∀ a b,

algebraMap R K (τ₀ (h a b)) = ∑ i, ev a i * dv b i) → (∀ a, ev a = ∑ m, algebraMap R K (τ₀ (h a (y m))) • ev (x m)) → Submodule.span K (Set.range ev) = ⊤ → ∀ (i : I) (t : K), Pi.single i t ∈ Submodule.span R (Set.range ev) ↔ ∀ b, t * dv b i ∈ (algebraMap R K).range

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SmoothOfRelativeDimensionOfFibre

theorem **smoothOfRelativeDimension_of_fibre** : ∀ (R K : Type) [CommRing R] [IsDomain R] [Field K] (β : R →+* K) (B : Over (Sp R)) [GrpObj B] (X' : Scheme.{0}) (π : X' ⟶ B.left) (p : X' ⟶ Sp K) (g : ℕ), IsAbelianScheme B → IsPullback π p B.hom (spMap β) → SmoothOfRelativeDimension g p → SmoothOfRelativeDimension g B.hom

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SmoothOfRelativeDimensionTensor

theorem **finrank_tensor_of_algebra** (O M k B L : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] [Field k] [CommRing B] [Algebra O B] [Algebra k B] [Module.Finite k B] [AddCommGroup L] [Module k L] [Module O L] [Module B L] [IsScalarTower k B L] [IsScalarTower O B L] [SMulCommClass O k L] [Module.Finite k L] : Module.finrank k (L ⊗[O] M) = Module.finrank O M * Module.finrank k L

theorem **finrank_tensor** (O M k L : Type) [CommRing O] [IsDomain O] [Module.Finite ℤ O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] [Field k] [AddCommGroup L] [Module k L] [Module O L] [SMulCommClass O k L] [Module.Finite k L] : Module.finrank k (L ⊗[O] M) = Module.finrank O M * Module.finrank k L

theorem **dim_eq_of_isPullback** {X Y S T X' S' : Scheme.{0}} {f : X ⟶ S} {g : Y ⟶ T} {f' : X' ⟶ S'} {a : X' ⟶ X} {b : S' ⟶ S} {c : X' ⟶ Y} {d : S' ⟶ T} (h : IsPullback a f' f b) (h' : IsPullback c f' g d) {n m : ℕ} {W : X.Opens} {V : Y.Opens} (hW : SmoothOfRelativeDimension n (W.ι ≫ f)) (hV : SmoothOfRelativeDimension m (V.ι ≫ g)) (x : X') (hx : a x ∈ W) (hx' : c x ∈ V) : n = m

theorem **isPullback_mul** {S : Scheme.{0}} (G : Over S) [GrpObj G] : IsPullback (fst G G * snd G G).left (snd G G).left G.hom G.hom

theorem **exists_dim_of_grpObj** {k : Type} [Field k] (G : Over (Sp k)) [GrpObj G] [Smooth G.hom] : ∃ n, SmoothOfRelativeDimension n G.hom

theorem **dim_eq_field** {k O M : Type} [Field k] [CommRing O] [IsDomain O] [Module.Finite ℤ O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] {G A : Over (Sp k)} [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] {a : OAct O G} {b : OAct O A} (t : Serre.IsTensor A b G a M) {g : ℕ} (hG : SmoothOfRelativeDimension g G.hom) {n : ℕ} (hA : SmoothOfRelativeDimension n A.hom) : n = Module.finrank O M * g

theorem **smoothOfRelativeDimension_tensor** : ∀ (S : Scheme.{0}) (O : Type) [CommRing O] [IsDomain O] [Module.Finite ℤ O] (M : Type) [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (G A : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (g : ℕ), Nonempty (Serre.IsTensor A b G a M) → SmoothOfRelativeDimension g G.hom → SmoothOfRelativeDimension (Module.finrank O M * g) A.hom

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SpecialisationNumberFieldAndCountableValues

theorem **exists_algHom_isAlgebraic_of_finiteType** (F A L : Type*) [Field F] [CommRing A] [Nontrivial A] [Algebra F A] [Algebra.FiniteType F A] [Field L] [IsAlgClosed L] [Algebra F L] : ∃ φ : A →ₐ[F] L, ∀ x, IsAlgebraic F (φ x)

theorem **_root_.GlobalTheory.exists_ringHom_extend_of_finiteType** (F A L : Type*) [Field F] [CommRing A] [Nontrivial A] [Algebra F A] [Algebra.FiniteType F A] [Field L] [IsAlgClosed L] (σ : F →+* L) : ∃ φ : A →+* L, ∀ x, φ (algebraMap F A x) = σ x

theorem **exists_fg_subalgebra_over** {k L : Type*} [Field k] [Field L] [Algebra k L] (F : IntermediateField k L) (R : Subalgebra k L) (hR : R.FG) (u : L) : ∃ A : Subalgebra F L, Algebra.FiniteType F A ∧ R ≤ A.restrictScalars k ∧ u ∈ A

private lemma **exists_algHom_extend_of_fg_ne_zero** {k L M : Type*} [Field k] [Field L] [Algebra k L] [Field M] [IsAlgClosed M] [Algebra k M] (F : IntermediateField k L) (σ : F →ₐ[k] M) (R : Subalgebra k L) (hR : R.FG) (r : R) (hr : r ≠ 0) : ∃ ρ : R →ₐ[k] M, (∀ (x : R) (hx : (x : L) ∈ F), ρ x = σ (x, hx)) ∧ ρ r ≠ 0

theorem **_root_.GlobalTheory.exists_specialisation_finite** {k L : Type*} [Field k] [Field L] [Algebra k L] [IsAlgClosed L] (R : Subalgebra k L) (hR : R.FG) (r : R) (hr : r ≠ 0) : ∃ (K₁ : IntermediateField k L) (ρ : R →ₐ[k] K₁), FiniteDimensional k K₁ ∧ ρ r ≠ 0 ∧ ∀ c : R, IsAlgebraic k c → (ρ c : L) = c

theorem **exists_algHom_apply_eq_of_transcendental** {k L : Type*} [Field k] [Field L] [Algebra k L] [IsAlgClosed L] (R : Subalgebra k L) (hR : R.FG) (c : R) (hc : Transcendental k (c : L)) (t : L) (ht : Transcendental k t) : ∃ ρ : R →ₐ[k] L, ρ c = t

theorem **_root_.GlobalTheory.specialisation_numberField_and_countable_values** : ∀ (k : IntermediateField ℚ ℂ) [FiniteDimensional ℚ k] (R : Subalgebra k ℂ), R.FG → ∀ (c r : R), r ≠ 0 → (∃ (K₁ : IntermediateField k ℂ) (ρ : R →ₐ[k] K₁), FiniteDimensional k K₁ ∧ ρ r ≠ 0 ∧ (IsAlgebraic k c → (ρ c : ℂ) = c)) ∧ (∀ S : Set ℂ, S.Countable → (∀ ρ : R →ₐ[k] ℂ, ρ c ∈ S) → IsAlgebraic k c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SumExtendByZeroLeSum

theorem **extendByZero_le_sum** {p : ℕ} (f : Fin p → ℕ) (i : ℕ) : Formalism.L2.Pj.ext0 f i ≤ ∑ k, f k

theorem **sum_extendByZero_le_sum** {p : ℕ} (f : Fin p → ℕ) (m : ℕ) : ∑ i : Fin m, Formalism.L2.Pj.ext0 f i ≤ ∑ k, f k

theorem **wtA_le_tot** (w : WeightFamilies.Wt D j) (b : D.host.E →+* ℂ) (i : ℕ) : wtA w (b, i) ≤ tot w b

theorem **sum_wtA_le_tot** (w : WeightFamilies.Wt D j) (b : D.host.E →+* ℂ) (m : ℕ) : ∑ i : Fin m, wtA w (b, i) ≤ tot w b

theorem **sum_tot** (w : WeightFamilies.Wt D j) : ∑ b, tot w b = ∑ t, ((∑ i, w.aP t i) + ∑ i, w.aM t i)

theorem **tot_le_size** (w : WeightFamilies.Wt D j) : ((∑ b, tot w b : ℕ) : ℝ) + ((D.R.f * D.R.r : ℕ) : ℝ) * w.n ≤ w.size

theorem **spread_le** {r : ℕ} (hr : 0 < r) (x : Fin r → ℝ) (A B : ℝ) (h1 : ∀ i, x i ≤ A) (h2 : ∀ i, -B ≤ x i) : (⊔ i, x i) - ⊓ i, x i ≤ A + B

theorem **invFun_relab** (T : Core D) (τ : D.host.E →+* ℂ) : Function.invFun (relab T.Q.c T.σ₀) τ = τ

theorem **spread_laU_le** (T : Core D) (hr : 0 < D.R.r) (w : WeightFamilies.Wt D j) (k : (T.U j T.σ₀).Comp) (τ : ((T.U j T.σ₀).X k).Ψ) : ((T.U j T.σ₀).X k).spread (laU (T.U j T.σ₀) (wtA w) w.n k) τ ≤ (tot w τ.1 : ℝ) + tot w (conjEmb τ.1) + w.n

theorem **sum_cm_le** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : Data E r) (B : (E →+* ℂ) → ℕ) : ∑ τ : X.Ψ, (B τ.1 + B (conjEmb τ.1)) ≤ ∑ b, B b

theorem **fdeg_eq** (D : D7) : fdeg D.host.E = D.R.f

theorem **one_le_dE** (D : D7) : 1 ≤ dE D.host.E

theorem **sqrt_le_dE** (D : D7) : Real.sqrt D.host.d ≤ dE D.host.E

theorem **sizes** : ∀ (D : D7) (T : Core D) (h : LawsCore T) (j : ℕ), InR D j → ∀ w : WeightFamilies.Wt D j, 0 < D.R.r ∧ (T.Q.Dw j).p + (T.Q.Dw j).q = D.R.r ∧ fdeg D.host.E = D.R.f ∧ Real.sqrt D.host.d ≤ dE D.host.E ∧ 1 ≤ dE D.host.E ∧ Real.log (dE D.host.E) = logdE D ∧ (∀ k, (∑ τ, ((T.U j T.σ₀).X k).spread (laU (T.U j T.σ₀) (wtA w) w.n k) τ) ≤ w.size) ∧ ((∑ b : D.host.E →+* ℂ, ∑ i : Fin ((T.Q.Dw j).mult b), wtA w (b, i) : ℕ) : ℝ) + ((D.R.f * D.R.r : ℕ) : ℝ) * w.n ≤ w.size

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SumLogNormEmbeddingEqFinrankMulLogAbsNorm

theorem **exists_span_singleton_eq_map_of_integral** (K M : Type) [Field K] [NumberField K] [Field M] [NumberField M] [Algebra K M] (𝔟 : Ideal (𝒪 K)) (β : M) (h𝔟 : 𝔟 ≠ ⊥) (hβ : β ≠ 0) (h₁ : ∀ x : K, (∀ b ∈ 𝔟, IsIntegral ℤ (x * (b : K))) → IsIntegral ℤ (β * algebraMap K M x)) (h₂ : ∀ y ∈ 𝔟, IsIntegral ℤ (β⁻¹ * algebraMap K M (y : K))) : ∃ β₀ : 𝒪 M, (β₀ : M) = β ∧ Ideal.span {β₀} = 𝔟.map (algebraMap (𝒪 K) (𝒪 M))

theorem **abs_norm_eq_absNorm_pow_of_integral** (K M : Type) [Field K] [NumberField K] [Field M] [NumberField M] [Algebra K M] (𝔟 : Ideal (𝒪 K)) (β : M) (h𝔟 : 𝔟 ≠ ⊥) (hβ : β ≠ 0) (h₁ : ∀ x : K, (∀ b ∈ 𝔟, IsIntegral ℤ (x * (b : K))) → IsIntegral ℤ (β * algebraMap K M x)) (h₂ : ∀ y ∈ 𝔟, IsIntegral ℤ (β⁻¹ * algebraMap K M (y : K))) : |Algebra.norm ℚ β| = (Ideal.absNorm 𝔟 : ℚ) ^ Module.finrank K M

theorem **sum_log_norm_embedding_eq_finrank_mul_log_absNorm** (K M : Type) [Field K] [NumberField K] [Field M] [NumberField M] [Algebra K M] (𝔟 : Ideal (𝒪 K)) (β : M) (h𝔟 : 𝔟 ≠ ⊥) (hβ : β ≠ 0) (h₁ : ∀ x : K, (∀ b ∈ 𝔟, IsIntegral ℤ (x * (b : K))) → IsIntegral ℤ (β * algebraMap K M x)) (h₂ : ∀ y ∈ 𝔟, IsIntegral ℤ (β⁻¹ * algebraMap K M (y : K))) : ∑ σ : M →+* ℂ, Real.log ‖σ β‖ = (Module.finrank K M : ℝ) * Real.log (Ideal.absNorm 𝔟 : ℝ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SurjectiveAppTopOfSection

theorem **surjective_appTop_of_section** {K : Type} [Field K] {Y : Scheme.{0}} [IsIntegral Y] (p : Y ⟶ Sp K) [UniversallyClosed p] (e : Sp K ⟶ Y) (he : e ≫ p = 𝟙 _) : Function.Surjective p.appTop.hom

theorem **exists_gen_of_generic** : ∀ (R K : Type) [CommRing R] [IsDedekindDomain R] [Field K] [Algebra R K] [IsFractionRing R K] (A : Over (Sp R)) [GrpObj A], IsAbelianScheme A → ∀ (X' : Scheme.{0}) (π : X' ⟶ A.left) (p : X' ⟶ Sp K), IsPullback π p A.hom (spMap (algebraMap R K)) → ∀ N : LB A.left, N.IsLB → (∃ t : LB.Gen (N.pull (eSec A)), t.IsGen) → (∃ t : LB.Gen (N.pull π), t.IsGen) → ∃ t : LB.Gen N, t.IsGen

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SurjectiveFst

theorem **equation_inr** (c : R) : (W.map (algebraMap R (DualNumber R))).toProjective.Equation ![TrivSqZeroExt.inr c, -1, 0]

theorem **scale_vec** (c : R) : spMap (epsScaleR c) ≫ vec W 1 = vec W c

theorem **vec_inj** {c d : R} (e : vec W c = vec W d) : c = d

theorem **fst_vec** (c : R) : spMap (TrivSqZeroExt.fstHom R R R).toRingHom ≫ vec W c = zero W

theorem **surjective_fst** : Function.Surjective (PrimeSpectrum.comap (TrivSqZeroExt.fstHom R R R).toRingHom)

theorem **exists_eq_vec** (g : Spec (.of (DualNumber R)) ⟶ scheme W) (hg : g ≫ toSpec W = spMap (algebraMap R (DualNumber R))) (h0 : spMap (TrivSqZeroExt.fstHom R R R).toRingHom ≫ g = zero W) : ∃ c, g = vec W c

theorem **isLieBasis_eps** : ∀ (R : Type) [CommRing R] (W : WeierstrassCurve R) [GrpObj (schemeOver W)], η[schemeOver W].left = zero W → ∃ v : epsR R ⟶ schemeOver W, (∃ h : IsUnimodularSol (W.map (algebraMap R (DualNumber R))) ![DualNumber.eps, -1, 0], v.left = ptHom W (algebraMap R (DualNumber R)) ![DualNumber.eps, -1, 0] h) ∧ IsLieBasis ![v]

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SurjectiveOfComplexPointsLiftOfUniversallyClosed

theorem **_root_.GlobalTheory.ComplexPoints.exists_point_mem** {X : Scheme.{u}} (p : X ⟶ Spec (.of K)) [LocallyOfFiniteType p] {Z : Set X} (hZ : IsLocallyClosed Z) (hne : Z.Nonempty) : ∃ x : Spec (.of K) ⟶ X, x ≫ p = 𝟙 _ ∧ ∀ a, x a ∈ Z

theorem **exists_geomPoint_mem** {X S : Scheme.{u}} (p : X ⟶ S) [LocallyOfFiniteType p] (t : Spec (.of K) ⟶ S) [Surjective t] {Z : Set X} (hZ : IsLocallyClosed Z) (hne : Z.Nonempty) : ∃ x : Spec (.of K) ⟶ X, x ≫ p = t ∧ ∀ a, x a ∈ Z

theorem **surjective_specMap_field** {k L : Type u} [Field k] [Field L] (σ : k →+* L) : Surjective (Spec.map (CommRingCat.ofHom σ))

theorem **_root_.GlobalTheory.ComplexPoints.top_le_preimage_iff** {L : Type u} [Field L] {X : Scheme.{u}} (x : Spec (.of L) ⟶ X) (U : X.Opens) (a : ↥(Spec (.of L))) : ⊤ ≤ x ⁻¹ᵁ U ↔ x a ∈ U

theorem **opens_le_of_geomPoints** {X S : Scheme.{u}} (p : X ⟶ S) [LocallyOfFiniteType p] (t : Spec (.of K) ⟶ S) [Surjective t] (U V : X.Opens) (h : ∀ x : Spec (.of K) ⟶ X, x ≫ p = t → ⊤ ≤ x ⁻¹ᵁ U → ⊤ ≤ x ⁻¹ᵁ V) : U ≤ V

theorem **closed_eq_univ_of_geomPoints** {X S : Scheme.{u}} (p : X ⟶ S) [LocallyOfFiniteType p] (t : Spec (.of K) ⟶ S) [Surjective t] {Z : Set X} (hZ : IsClosed Z) (h : ∀ x : Spec (.of K) ⟶ X, x ≫ p = t → ∃ a, x a ∈ Z) : Z = Set.univ

theorem **_root_.GlobalTheory.ComplexPoints.surjective_of_geomPoints_lift** {A B S : Scheme.{u}} (p : A ⟶ S) [LocallyOfFiniteType p] (t : Spec (.of K) ⟶ S) [Surjective t] (ψ : B ⟶ A) (hψ : IsClosed (Set.range ψ)) (h : ∀ x : Spec (.of K) ⟶ A, x ≫ p = t → ∃ y : Spec (.of K) ⟶ B, y ≫ ψ = x) : Surjective ψ

theorem **_root_.GlobalTheory.ComplexPoints.hom_ext_of_points** {X Y S : Scheme.{u}} (p : X ⟶ Spec (.of K)) [LocallyOfFiniteType p] [IsReduced X] (q : Y ⟶ S) [IsSeparated q] {f g : X ⟶ Y} (hfg : f ≫ q = g ≫ q) (h : ∀ x : Spec (.of K) ⟶ X, x ≫ p = 𝟙 _ → x ≫ f = x ≫ g) : f = g

theorem **_root_.GlobalTheory.ComplexPoints.top_le_preimage_basicOpen_iff** (x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ ⟶ X) (U : X.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (f : Γ(X, U)) : ⊤ ≤ x ⁻¹ᵁ X.basicOpen f ↔ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.valAt x U h f ≠ 0

theorem **isUnit_of_valAt_ne_zero** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] (U : X.Opens) (f : Γ(X, U)) (h : ∀ (x : Sp ℂ ⟶ X) (hx : ⊤ ≤ x ⁻¹ᵁ U), x ≫ p = 𝟙 _ → valAt x U hx f ≠ 0) : IsUnit f

theorem **eq_zero_of_valAt_eq_zero** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (U : X.Opens) (f : Γ(X, U)) (h : ∀ (x : Sp ℂ ⟶ X) (hx : ⊤ ≤ x ⁻¹ᵁ U), x ≫ p = 𝟙 _ → valAt x U hx f = 0) : f = 0

theorem **eq_of_valAt_eq** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (U : X.Opens) (f g : Γ(X, U)) (h : ∀ (x : Sp ℂ ⟶ X) (hx : ⊤ ≤ x ⁻¹ᵁ U), x ≫ p = 𝟙 _ → valAt x U hx f = valAt x U hx g) : f = g

private theorem **opens_le_of_forall_complex_section** (p : X ⟶ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ) [LocallyOfFiniteType p] (U V : X.Opens) (h : ∀ x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ⊤ ≤ x ⁻¹ᵁ U → ⊤ ≤ x ⁻¹ᵁ V) : U ≤ V

theorem **_root_.GlobalTheory.ComplexPoints.eq_of_valAt'_eq** (p : X ⟶ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (U : X.Opens) (f g : Γ(X, U)) (h : ∀ x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ⊤ ≤ x ⁻¹ᵁ U → _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.valAt' x U f = _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.valAt' x U g) : f = g

theorem **_root_.GlobalTheory.ComplexPoints.isUnit_of_valAt'_ne_zero** (p : X ⟶ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ) [LocallyOfFiniteType p] (U : X.Opens) (f : Γ(X, U)) (h : ∀ x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ⊤ ≤ x ⁻¹ᵁ U → _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.valAt' x U f ≠ 0) : IsUnit f

theorem **hom_ext_of_complexPoints** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (q : Y ⟶ Sp ℂ) [IsSeparated q] {f g : X ⟶ Y} (hf : f ≫ q = p) (hg : g ≫ q = p) (h : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → x ≫ f = x ≫ g) : f = g

theorem **_root_.GlobalTheory.ComplexPoints.over_hom_ext_of_complexPoints** {A B : Over (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ)} [LocallyOfFiniteType A.hom] [IsReduced A.left] [IsSeparated B.hom] {f g : A ⟶ B} (h : ∀ x : 𝟙_ (Over (_root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ)) ⟶ A, x ≫ f = x ≫ g) : f = g

theorem **_root_.GlobalTheory.ComplexPoints.opens_le_of_complexPoints** (p : X ⟶ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ) [LocallyOfFiniteType p] (U V : X.Opens) (h : ∀ x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ⊤ ≤ x ⁻¹ᵁ U → ⊤ ≤ x ⁻¹ᵁ V) : U ≤ V

theorem **opens_eq_top_of_complexPoints** (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] (U : X.Opens) (h : ∀ x : Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ⊤ ≤ x ⁻¹ᵁ U) : U = ⊤

theorem **_root_.GlobalTheory.ComplexPoints.iSup_eq_top_of_complexPoints** (p : X ⟶ _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ) [LocallyOfFiniteType p] {I : Type*} (U : I → X.Opens) (h : ∀ x : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Sp ℂ ⟶ X, x ≫ p = 𝟙 _ → ∃ i, ⊤ ≤ x ⁻¹ᵁ U i) : ⨆ i, U i = ⊤

theorem **surjective_of_complexPoints_lift** {k : Type} [Field k] (σ : k →+* ℂ) {A B : Scheme.{0}} (a : A ⟶ Sp k) [LocallyOfFiniteType a] (ψ : B ⟶ A) (hψ : IsClosed (Set.range ψ)) (h : ∀ x : Sp ℂ ⟶ A, x ≫ a = spMap σ → ∃ y : Sp ℂ ⟶ B, y ≫ ψ = x) : Surjective ψ

theorem **surjective_of_complexPoints_lift_of_universallyClosed** {k : Type} [Field k] (σ : k →+* ℂ) {A B : Scheme.{0}} (a : A ⟶ Sp k) [LocallyOfFiniteType a] [IsSeparated a] (b : B ⟶ Sp k) [UniversallyClosed b] (ψ : B ⟶ A) (hw : ψ ≫ a = b) (h : ∀ x : Sp ℂ ⟶ A, x ≫ a = spMap σ → ∃ y : Sp ℂ ⟶ B, y ≫ ψ = x) : Surjective ψ

theorem **eq_zero_and_isUnit_of_valAt** : ∀ (X : Scheme.{0}) (p : X ⟶ Sp ℂ) [LocallyOfFiniteType p] [IsReduced X] (U : X.Opens) (f : Γ(X, U)), ((∀ (x : Sp ℂ ⟶ X) (h : ⊤ ≤ x ⁻¹ᵁ U), x ≫ p = 𝟙 _ → valAt x U h f = 0) → f = 0) ∧ ((∀ (x : Sp ℂ ⟶ X) (h : ⊤ ≤ x ⁻¹ᵁ U), x ≫ p = 𝟙 _ → valAt x U h f ≠ 0) → IsUnit f)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SymmetricLineBundleOfRegularSection

theorem **appLE_mem_nonZeroDivisors** {X Y : Scheme.{0}} (f : Y ⟶ X) [IsOpenImmersion f] (W : X.Opens) (a : Γ(X, W)) (ha : ∀ (V : X.Opens) (h : V ≤ W), resG X h a ∈ nonZeroDivisors Γ(X, V)) (V : Y.Opens) (h : V ≤ f ⁻¹ᵁ W) : (f.appLE W V h).hom a ∈ nonZeroDivisors Γ(Y, V)

theorem **inv_eq_zero_iff** {S : Scheme.{0}} (A : Over S) [GrpObj A] {T : Scheme.{0}} (t : T ⟶ A.left) : t ≫ (GrpObj.inv : A ⟶ A).left = (t ≫ (GrpObj.inv : A ⟶ A).left) ≫ A.hom ≫ eSec A ↔ t = t ≫ A.hom ≫ eSec A

theorem **sym_of_regular_section** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] (L : LB A.left) (θ : L.Sec), L.IsLB → (∀ (i : L.ι) (V : Scheme.Opens A.left) (h : V ≤ L.U i), resG A.left h (θ.f i) ∈ nonZeroDivisors Γ(A.left, V)) → (∀ (T : Scheme.{0}) (t : T ⟶ A.left) (i : L.ι) (h : ⊤ ≤ t ⁻¹ᵁ L.U i), (t.appLE (L.U i) ⊤ h).hom (θ.f i) = 0 ↔ t = t ≫ A.hom ≫ eSec A) → Out.Sym A L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.SymmetricLineBundleOfRegularSectionStatement

theorem **_root_.GlobalTheory.T.symmetric_lineBundle_of_regular_section** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] (L : LB A.left) (θ : L.Sec), L.IsLB → (∀ (i : L.ι) (V : Scheme.Opens A.left) (h : V ≤ L.U i), resG A.left h (θ.f i) ∈ nonZeroDivisors Γ(A.left, V)) → (∀ (T : Scheme.{0}) (t : T ⟶ A.left) (i : L.ι) (h : ⊤ ≤ t ⁻¹ᵁ L.U i), (t.appLE (L.U i) ⊤ h).hom (θ.f i) = 0 ↔ t = t ≫ A.hom ≫ eSec A) → Out.Sym A L

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TangentVectorExistsUniqueUniformisingVector

theorem **existsUnique_isDiff_and_linear** : ∀ (x : KeyIn ℂ) (u : HolUnif x.P.A x.P.L x.P.g), (∀ w, IsTangent w → ∃! W, u.IsDiff W w) ∧ ∀ (n : ℕ) (v : Fin n → (epsR ℂ ⟶ x.P.A)) (W : Fin n → Fin x.P.g → ℂ) (cf : Fin n → ℂ) (w : epsR ℂ ⟶ x.P.A), (∀ i, IsTangent (v i)) → (∀ i, u.IsDiff (W i) (v i)) → IsLieComb cf v w → u.IsDiff (∑ i, cf i • W i) w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TangentVectorExistsUniqueUniformisingVectorStatement

theorem **_root_.GlobalTheory.T.tangentVector_existsUnique_uniformisingVector** : ∀ (x : KeyIn ℂ) (u : HolUnif x.P.A x.P.L x.P.g), (∀ w, IsTangent w → ∃! W, u.IsDiff W w) ∧ ∀ (n : ℕ) (v : Fin n → (epsR ℂ ⟶ x.P.A)) (W : Fin n → Fin x.P.g → ℂ) (cf : Fin n → ℂ) (w : epsR ℂ ⟶ x.P.A), (∀ i, IsTangent (v i)) → (∀ i, u.IsDiff (W i) (v i)) → IsLieComb cf v w → u.IsDiff (∑ i, cf i • W i) w

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TateIsoExistsAlgClosed

theorem **exists_algClosure_lift** {K k : Type} [Field K] [Field k] [IsAlgClosed k] (β : K →+* k) : ∃ j : AlgebraicClosure K →+* k, j.comp (algebraMap K (AlgebraicClosure K)) = β

theorem **spMap_comp_eq** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap (g.comp f) = spMap g ≫ spMap f

theorem **onTors_iff** {Ω Q' : Type} [Field Ω] [Group Q'] (i : k →+* Ω) (r : Q →* Q') (hr : Function.Injective r) {e : ℕ → Q → Q → kˣ → Prop} {e' : ℕ → Q' → Q' → Ωˣ → Prop} (h : ∀ (n : ℕ) x y (c : kˣ), 0 < n → x ^ n = 1 → y ^ n = 1 → (e n x y c ↔ e' n (r x) (r y) (Units.map i.toMonoidHom c))) (n : ℕ) (x y : Q) (c : kˣ) : OnTors e' n (r x) (r y) (Units.map i.toMonoidHom c) ↔ OnTors e n x y c

theorem **onto_tors** {Q' : Type} [Group Q'] (r : Q →* Q') (hr : Function.Injective r) (h : ∀ (n : ℕ) y, 0 < n → y ^ n = 1 → ∃ x, r x = y) (n : ℕ) (hn : 0 < n) (z : Q') (hz : z ^ n = 1) : ∃ y : Q, y ^ n = 1 ∧ r y = z

theorem **cmTorsionIsometry_transport_geometricPoint** (hL : Out.Laws) {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (D : Datum H) {S : Scheme.{0}} [IsLocallyNoetherian S] (P : Out.PolAV S) (act : 𝒪 E → (P.A ⟶ P.A)) (hm : ∀ c, IsMonHom (act c)) {F Ω k : Type} [Field F] [Field Ω] [Field k] [IsAlgClosed F] [IsAlgClosed Ω] [IsAlgClosed k] [CharZero F] (i : F →+* Ω) (j : F →+* k) (x : Sp F ⟶ S) (y : Sp Ω ⟶ S) (z : Sp k ⟶ S) (hy : spMap i ≫ x = y) (hz : spMap j ≫ x = z) (I : TateIso D Ω (Over.mk y ⟶ P.A) (fun c y => y ≫ act c) (fun n y y' c => Out.WeilIs P.A P.L n y y' (unitOn c))) : ∃ I' : TateIso D k (Over.mk z ⟶ P.A) (fun c y => y ≫ act c) (fun n y y' c => Out.WeilIs P.A P.L n y y' (unitOn c)), (∀ (n : ℕ) (u : F), 0 < n → (I.ζ n : Ω) = i u → (I'.ζ n : k) = j u) ∧ ∀ (n : ℕ) a (s : 𝟙_ (Over S) ⟶ P.A), 0 < n → toUnit _ ≫ s = I.ev n a → toUnit _ ≫ s = I'.ev n a

theorem **tateIso_move** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (T : PAT H (Sp K)) (I : TateIso D ℂ (Over.mk (spMap σ) ⟶ T.t.G) (fun a x => x ≫ T.t.act.act a) (fun n x x' c => Out.WeilIs T.t.G T.t.PP.L n x x' (unitOn c))) (k : Type) [Field k] [IsAlgClosed k] (β : K →+* k), ∃ I' : TateIso D k (Over.mk (spMap β) ⟶ T.t.G) (fun a x => x ≫ T.t.act.act a) (fun n x x' c => Out.WeilIs T.t.G T.t.PP.L n x x' (unitOn c)), (∀ (n : ℕ) (z : K), 0 < n → (I.ζ n : ℂ) = σ z → (I'.ζ n : k) = β z) ∧ ∀ (n : ℕ) a (s : 𝟙_ (Over (Sp K)) ⟶ T.t.G), 0 < n → toUnit _ ≫ s = I.ev n a → toUnit _ ≫ s = I'.ev n a

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TateIsoExistsAlgClosedStatement

theorem **_root_.GlobalTheory.T.tate_iso_exists_alg_closed** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (K : Type) [Field K] [NumberField K] (σ : K →+* ℂ) (T : PAT H (Sp K)) (I : TateIso D ℂ (Over.mk (spMap σ) ⟶ T.t.G) (fun a x => x ≫ T.t.act.act a) (fun n x x' c => Out.WeilIs T.t.G T.t.PP.L n x x' (unitOn c))) (k : Type) [Field k] [IsAlgClosed k] (β : K →+* k), ∃ I' : TateIso D k (Over.mk (spMap β) ⟶ T.t.G) (fun a x => x ≫ T.t.act.act a) (fun n x x' c => Out.WeilIs T.t.G T.t.PP.L n x x' (unitOn c)), (∀ (n : ℕ) (z : K), 0 < n → (I.ζ n : ℂ) = σ z → (I'.ζ n : k) = β z) ∧ ∀ (n : ℕ) a (s : 𝟙_ (Over (Sp K)) ⟶ T.t.G), 0 < n → toUnit _ ≫ s = I.ev n a → toUnit _ ≫ s = I'.ev n a

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TensorLieBasisExists

theorem **isLieComb_one_iff** (c : R) (v x : epsR R ⟶ G) : IsLieComb ![c] ![v] x ↔ x = GrpTangent.epsScale c ≫ v

theorem **hom_multiples** (a : OAct O G) {v : epsR R ⟶ G} (hv : IsTangent v) (ha : ∀ c, v ≫ a.act c = GrpTangent.epsScale (algebraMap O R c) ≫ v) (r : R) : (multiples a hv ha r).hom = GrpTangent.epsScale r ≫ v

theorem **pre_eq_zero_iff** (a : OAct O G) (x : Pts a (epsR R)) : Pts.pre a (toUnit (epsR R) ≫ epsZero R) x = 0 ↔ IsTangent x.hom

theorem **isTangent_sub_pre** (a : OAct O G) (x : Pts a (epsR R)) : IsTangent (x - Pts.pre a (toUnit (epsR R) ≫ epsZero R) x).hom

theorem **exists_retraction** (a : OAct O G) {v : epsR R ⟶ G} (hv : IsLieBasis ![v]) (ha : ∀ c, v ≫ a.act c = GrpTangent.epsScale (algebraMap O R c) ≫ v) : ∃ ρ : Pts a (epsR R) →ₗ[O] R, ρ ∘ₗ multiples a (hv.1 0) ha = LinearMap.id ∧ multiples a (hv.1 0) ha ∘ₗ ρ = LinearMap.id - Pts.pre a (toUnit (epsR R) ≫ epsZero R)

theorem **exists_lie_tensor** : ∀ (R : Type) [CommRing R] (G : Over (Sp R)) [GrpObj G] [IsCommMonObj G] (O : Type) [CommRing O] [Algebra O R] (M : Type) [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (A : Over (Sp R)) [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (t : IsTensor A b G a M) (v : epsR R ⟶ G), IsLieBasis ![v] → (∀ c, IsLieComb ![algebraMap O R c] ![v] (v ≫ a.act c)) → ∃ Φ : R ⊗[O] M → (epsR R ⟶ A), Function.Injective Φ ∧ (∀ x, IsTangent x ↔ x ∈ Set.range Φ) ∧ (∀ y z, Φ (y + z) = Φ y ∗ Φ z) ∧ (∀ (c : R) y, IsLieComb ![c] ![Φ y] (Φ (c • y))) ∧ (∀ m, Φ (1 ⊗ₜ m) = v ≫ t.tmul m) ∧ ∀ (φ : M →ₗ[O] M) y, Φ (φ.baseChange R y) = Φ y ≫ t.mapLeft t φ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TensorMapFiniteOfSplit

theorem **_root_.GlobalTheory.TensorFinite.isFinite_of_comp** {X Y Z : Over S} (f : X ⟶ Y) (g : Y ⟶ Z) [IsSeparated Y.hom] (h : IsFinite (f ≫ g).left) : IsFinite f.left

theorem **_root_.GlobalTheory.TensorFinite.isFinite_tensorHom** {X Y X' Y' : Over S} (f : X ⟶ Y) (g : X' ⟶ Y') (hf : IsFinite f.left) (hg : IsFinite g.left) : IsFinite (f ⊗ₘ g).left

theorem **id_pow_tensor** (G H : Over S) [GrpObj G] [GrpObj H] (n : ℕ) : (𝟙 (G ⊗ H)) ^ n = ((𝟙 G) ^ n ⊗ₘ (𝟙 H) ^ n)

theorem **isFinite_id_pow_powObj** (G : Over S) [GrpObj G] (n : ℕ) (h : IsFinite ((𝟙 G) ^ n).left) : ∀ m : ℕ, IsFinite ((𝟙 (Out.powObj G m)) ^ n).left | 0 => by have e : (𝟙 (Out.powObj G 0)) ^ n = 𝟙 _

theorem **isFinite_id_pow_of_retract** {H K : Over S} [GrpObj H] [GrpObj K] [IsSeparated H.hom] [IsSeparated K.hom] (ρ : Retract H K) [IsMonHom ρ.i] (n : ℕ) (h : IsFinite ((𝟙 K) ^ n).left) : IsFinite ((𝟙 H) ^ n).left

theorem **mapLeft_nsmul_id** {O : Type} [CommRing O] {M : Type} [AddCommGroup M] [Module O M] {G H : Over S} [GrpObj G] [GrpObj H] [IsCommMonObj G] [IsCommMonObj H] {a : OAct O G} {b : OAct O H} (t : IsTensor H b G a M) (n : ℕ) : t.mapLeft t (n • LinearMap.id) = (𝟙 H) ^ n

theorem **isFinite_mapLeft** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) [IsCommMonObj P.A] (O : Type) [CommRing O] (M : Type) [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (A : Over S) [GrpObj A] [IsCommMonObj A] (a : OAct O P.A) (b : OAct O A) (t : IsTensor A b P.A a M) (n k : ℕ) (l : M →ₗ[O] (Fin k → O)) (r : (Fin k → O) →ₗ[O] M), 0 < n → (∀ m, r (l m) = n • m) → IsFinite (t.mapLeft (IsTensor.pow (Out.powIsPowGrp P.A k) a) l).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TensorMapFiniteOfSplitStatement

theorem **_root_.GlobalTheory.T.tensor_map_finite_of_split** : Out.Laws → ∀ (S : Scheme.{0}) [IsLocallyNoetherian S] (P : Out.PolAV S) [IsCommMonObj P.A] (O : Type) [CommRing O] (M : Type) [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (A : Over S) [GrpObj A] [IsCommMonObj A] (a : OAct O P.A) (b : OAct O A) (t : IsTensor A b P.A a M) (n k : ℕ) (l : M →ₗ[O] (Fin k → O)) (r : (Fin k → O) →ₗ[O] M), 0 < n → (∀ m, r (l m) = n • m) → IsFinite (t.mapLeft (IsTensor.pow (Out.powIsPowGrp P.A k) a) l).left

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TensorPairingIdentityOfPureTensors

theorem **isTensor_induction** {S : Scheme.{0}} {O : Type} [CommRing O] {G A T : Over S} [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] {a : OAct O G} {b : OAct O A} {M : Type} [AddCommGroup M] [Module O M] (t : Serre.IsTensor A b G a M) {P : (T ⟶ A) → Prop} (y : T ⟶ A) (tmul : ∀ (m : M) (x : T ⟶ G), P (x ≫ t.tmul m)) (mul : ∀ y z, P y → P z → P (y ∗ z)) : P y

theorem **lamEq_of_tmul** : ∀ (S : Scheme.{0}) (O : Type) [CommRing O] (G A : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (M : Type) [AddCommGroup M] [Module O M] (t : Serre.IsTensor A b G a M) (A' : Over S) [GrpObj A'] [IsCommMonObj A'] (b' : OAct O A') (M' : Type) [AddCommGroup M'] [Module O M'] (t' : Serre.IsTensor A' b' G a M') (B B' : Over S) [GrpObj B] [IsCommMonObj B] [GrpObj B'] [IsCommMonObj B'] (L : LB B.left) (L' : LB B'.left) (f₁ : A ⟶ B) (f₂ : A' ⟶ B) (g₁ : A ⟶ B') (g₂ : A' ⟶ B'), L.IsLB → L'.IsLB → (∃ t : LB.Gen (Theta B L), t.IsGen) → (∃ t : LB.Gen (Theta B' L'), t.IsGen) → IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → (∀ m m', LamEq L (t.tmul m ≫ f₁) (t'.tmul m' ≫ f₂) L' (t.tmul m ≫ g₁) (t'.tmul m' ≫ g₂)) → LamEq L f₁ f₂ L' g₁ g₂

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TensorPairingIdentityOfPureTensorsStatement

theorem **_root_.GlobalTheory.T.tensor_pairing_identity_of_pure_tensors** : ∀ (S : Scheme.{0}) (O : Type) [CommRing O] (G A : Over S) [GrpObj G] [IsCommMonObj G] [GrpObj A] [IsCommMonObj A] (a : OAct O G) (b : OAct O A) (M : Type) [AddCommGroup M] [Module O M] (t : Serre.IsTensor A b G a M) (A' : Over S) [GrpObj A'] [IsCommMonObj A'] (b' : OAct O A') (M' : Type) [AddCommGroup M'] [Module O M'] (t' : Serre.IsTensor A' b' G a M') (B B' : Over S) [GrpObj B] [IsCommMonObj B] [GrpObj B'] [IsCommMonObj B'] (L : LB B.left) (L' : LB B'.left) (f₁ : A ⟶ B) (f₂ : A' ⟶ B) (g₁ : A ⟶ B') (g₂ : A' ⟶ B'), L.IsLB → L'.IsLB → (∃ t : LB.Gen (Theta B L), t.IsGen) → (∃ t : LB.Gen (Theta B' L'), t.IsGen) → IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → (∀ m m', LamEq L (t.tmul m ≫ f₁) (t'.tmul m' ≫ f₂) L' (t.tmul m ≫ g₁) (t'.tmul m' ≫ g₂)) → LamEq L f₁ f₂ L' g₁ g₂

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaLineDegreeAndHeightLowerBound

theorem **thetaDegree_and_lowerBound** : (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → T.dual.deg = (1 / 2) ∗ abelian₂.dual.deg + (x.P.g :

ℝ) / 4 * (Module.finrank ℚ K : ℝ) * Real.log (2 * Real.pi)) ∧ (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (H : (K →+* ℂ) → Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (La logOm : ℝ) (V : Type) [Fintype V] (c : V → Fin x.P.g) (G : MvPolynomial V K) (μ : V →₀ ℕ) (n : ℕ) (S : (K →+* ℂ) → ℝ) (Rr : ℝ), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → U.det ≠ 0 → 1 ≤ La → (∀ σ, (H σ).PosDef) → (∀ σ, abelian₂.n σ = Real.sqrt (H σ).det.re) → (∀ p : HeightOneSpectrum (𝒪 K), ∃ (cd : K) (cl : Fin

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaLineDegreeAndHeightLowerBoundStatement

theorem **_root_.GlobalTheory.T.thetaLine_degree_and_heightLowerBound** : (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → T.dual.deg = (1 / 2) * abelian₂.dual.deg + (x.P.g : ℝ) / 4 * (Module.finrank ℚ K : ℝ) * Real.log (2 * Real.pi)) ∧ (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (H : (K →+* ℂ) → Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (La logOm : ℝ) (V : Type) [Fintype V] (c : V → Fin x.P.g) (G : MvPolynomial V K) (μ : V →₀ ℕ) (n : ℕ) (S : (K →+* ℂ) → ℝ) (Rr : ℝ), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → U.det ≠ 0 → 1 ≤ La → (∀ σ, (H σ).PosDef) → (∀ σ, abelian₂.n σ = Real.sqrt (H σ).det.re) → (∀ p : HeightOneSpe

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaLineDualDegreeOfKeyFormulaPowerForm

theorem **isLocGen_pow_div_pow** {K : Type} [Field K] [NumberField K] {p : HeightOneSpectrum (𝒪 K)} {ca cd : K} {A B : FractionalIdeal (𝒪 K)⁰ K} (m n : ℕ) (hA : A ≠ 0) (hB : B ≠ 0) (ha : IsLocGen p ca A) (hb : IsLocGen p cd B) : IsLocGen p (ca ^ m / cd ^ n) (A ^ m * (B ^ n)⁻¹)

theorem **thetaLine_dual_deg_of_keyFormulaPow** : GlobalTheory.KeyFormulaPow → ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → T.dual.deg = (1 / 2) * abelian₂.dual.deg + (x.P.g : ℝ) / 4 * (Module.finrank ℚ K : ℝ) * Real.log (2 * Real.pi)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaLineDualDegreeOfKeyFormulaPowerFormStatement

theorem **_root_.GlobalTheory.T.thetaLine_dual_degree_of_keyFormulaPowerForm** : GlobalTheory.KeyFormulaPow → ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → T.dual.deg = (1 / 2) * abelian₂.dual.deg + (x.P.g : ℝ) / 4 * (Module.finrank ℚ K : ℝ) * Real.log (2 * Real.pi)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaPolynomialLowerBoundOfThetaLineDegree

theorem **thetaLine_lower_bound_of_degree** : (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → T.dual.deg = (1 / 2) * abelian₂.dual.deg + (x.P.g : ℝ) / 4 * (Module.finrank ℚ K : ℝ) * Real.log (2 * Real.pi)) → ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (H : (K →+* ℂ) → Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (La logOm : ℝ) (V : Type) [Fintype V] (c : V → Fin x.P.g) (G : MvPolynomial V K) (μ : V →₀ ℕ) (n : ℕ) (S : (K →+* ℂ) → ℝ) (Rr : ℝ), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → U.det ≠ 0 → 1 ≤ La → (∀ σ, (H σ).PosDef) → (∀ σ, abelian₂.n σ = Real.sqrt (H σ).det.re) → (∀ p : HeightOneSpectrum (𝒪 K), ∃ (cd : K) (cl :

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaPolynomialLowerBoundOfThetaLineDegreeStatement

theorem **_root_.GlobalTheory.T.thetaPolynomial_lower_bound_of_thetaLine_degree** : (∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → T.dual.deg = (1 / 2) * abelian₂.dual.deg + (x.P.g : ℝ) / 4 * (Module.finrank ℚ K : ℝ) * Real.log (2 * Real.pi)) → ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (T abelian₂ : MLine K) (ℓ : Fin x.P.g → MLine K) (U : Matrix (Fin x.P.g) (Fin x.P.g) K) (H : (K →+* ℂ) → Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (La logOm : ℝ) (V : Type) [Fintype V] (c : V → Fin x.P.g) (G : MvPolynomial V K) (μ : V →₀ ℕ) (n : ℕ) (S : (K →+* ℂ) → ℝ) (Rr : ℝ), T.Good → abelian₂.Good → (∀ p, KeyLocAt x T abelian₂ p) → (∀ σ, KeyArchAt x T abelian₂ σ) → (∀ j, (ℓ j).Good) → U.det ≠ 0 → 1 ≤ La → (∀ σ, (H σ).PosDef) → (∀ σ, abelian₂.n σ = Real.sqrt (H σ).det.re) → (∀ p : Hei

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaSectionNormAndHermitianFormIndepUniformisat

theorem **volume_image** {g : ℕ} (F : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (s : Set (Fin g → ℂ)) : (volume (F '' s)).toReal = ‖LinearMap.det F‖ ^ 2 * (volume s).toReal

theorem **volume_fund** {g : ℕ} (Λ Λ' : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ'] (F : (Fin g → ℂ) ≃ₗ[ℂ] (Fin g → ℂ)) (hF : ∀ z, F z ∈ Λ' ↔ z ∈ Λ) {s s' : Set (Fin g → ℂ)} (hs : IsAddFundamentalDomain Λ s volume) (hs' : IsAddFundamentalDomain Λ' s' volume) : (volume s').toReal = ‖LinearMap.det F.toLinearMap‖ ^ 2 * (volume s).toReal

theorem **norm_eq_and_hermForm_eq** : ∀ (x : KeyIn ℂ) (u : HolUnif x.P.A x.P.L x.P.g) (Hm : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin x.P.g → ℂ) → ℂ) (Fd : Set (Fin x.P.g → ℂ)) (u' : HolUnif x.P.A x.P.L x.P.g) (Hm' : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (χ' : u'.Λ → ℂ) (θ' : (Fin x.P.g → ℂ) → ℂ) (Fd' : Set (Fin x.P.g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume → u'.IsAH Hm' χ' ∧ u'.IsRig x.rig ∧ u'.IsFn x.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume → (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal = (∫ z in Fd', ‖θ' z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm' z z).re)) / (volume Fd').toReal ∧ ∀ (v w : epsR ℂ ⟶ x.P.A) (V W V' W' : Fin x.P.g → ℂ), IsTangent v → IsTangent w → u.IsDiff V v → u.IsDiff W w → u'.IsDiff V' v → u'.IsDiff W' w → hermForm Hm V W = hermForm Hm' V' W'

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ThetaSectionNormAndHermitianFormIndepUniformisatStatement

theorem **_root_.GlobalTheory.T.thetaSection_norm_and_hermitianForm_indep_uniformisation** : ∀ (x : KeyIn ℂ) (u : HolUnif x.P.A x.P.L x.P.g) (Hm : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin x.P.g → ℂ) → ℂ) (Fd : Set (Fin x.P.g → ℂ)) (u' : HolUnif x.P.A x.P.L x.P.g) (Hm' : Matrix (Fin x.P.g) (Fin x.P.g) ℂ) (χ' : u'.Λ → ℂ) (θ' : (Fin x.P.g → ℂ) → ℂ) (Fd' : Set (Fin x.P.g → ℂ)), u.IsAH Hm χ ∧ u.IsRig x.rig ∧ u.IsFn x.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume → u'.IsAH Hm' χ' ∧ u'.IsRig x.rig ∧ u'.IsFn x.th θ' ∧ IsAddFundamentalDomain u'.Λ Fd' volume → (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm z z).re)) / (volume Fd).toReal = (∫ z in Fd', ‖θ' z‖ ^ 2 * Real.exp (-Real.pi * (hermForm Hm' z z).re)) / (volume Fd').toReal ∧ ∀ (v w : epsR ℂ ⟶ x.P.A) (V W V' W' : Fin x.P.g → ℂ), IsTangent v → IsTangent w → u.IsDiff V v → u.IsDiff W w → u'.IsDiff V' v → u'.IsDiff W' w →

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TopLieVectorNormSqEqGramDet

theorem **sq_eq_det_gram** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (h : HermFam K (Fin x.P.g → K)), (GenFib.mk x.P φ hf x.v x.basis : GenFib A).IsHerm h → ∀ (T abelian₂ : MLine K) (s : K →+* ℂ), KeyArchAt x T abelian₂ s → abelian₂.n s ^ 2 = (h.gram s (fun i : Fin x.P.g => Pi.single i 1)).det.re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TopLieVectorNormSqEqGramDetStatement

theorem **_root_.GlobalTheory.T.topLieVector_norm_sq_eq_gramDet** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (h : HermFam K (Fin x.P.g → K)), (GenFib.mk x.P φ hf x.v x.basis : GenFib A).IsHerm h → ∀ (T abelian₂ : MLine K) (s : K →+* ℂ), KeyArchAt x T abelian₂ s → abelian₂.n s ^ 2 = (h.gram s (fun i : Fin x.P.g => Pi.single i 1)).det.re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TraceFormIntegralPerfectOfIsUnf

theorem **traceForm_integral_perfect_of_isUnf** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.IsUnf U → ∀ k, (∀ x ∈ (U.X k).Λ, ∀ y ∈ (U.X k).Λ, ∃ n : ℤ, EFf (U.X k).ξ0 (U.X k).Gm x y = n) ∧ ∀ φ : (U.X k).Λ →+ ℤ, ∃ x ∈ (U.X k).Λ, ∀ y : (U.X k).Λ, (φ y : ℚ) = EFf (U.X k).ξ0 (U.X k).Gm x y

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TraceMulConjIntIffMemOfMulDifferentEqOne

private theorem **fractionalIdeal_eq_one_of_smul_submodule_eq_self** {E : Type} [Field E] [NumberField E] {r : ℕ} {c : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (hc : c ≠ 0) {L : Submodule (𝒪 E) (Fin r → E)} (hL : L ≠ ⊥) (hfg : L.FG) (h : (↑c : Submodule (𝒪 E) E) • L = L) : c = 1

theorem **_root_.GlobalTheory.trace_xi_mul_conj_int_iff_mem** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : _root_.ReciprocityTheory.ChartRestriction.G.FD.DataX E r) (𝔄 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E), 0 < r → 𝔄 ≠ 0 → X.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ 𝔄, ∃ u ∈ X.L', v = a • u} → ∀ x : E, (∀ y ∈ 𝔄, ∃ n : ℤ, Algebra.trace ℚ E (X.ξ0 * x * _root_.ReciprocityTheory.ChartRestriction.G.FD.cc y) = n) ↔ x ∈ 𝔄

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TwoMulDegDualLineCmIdeal

theorem **_root_.GlobalTheory.CM.deg_ideal_smul** {K V : Type} [Field K] [NumberField K] [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K V) (Λ : Submodule (𝒪 K) V) {v : V} (hv : v ≠ 0) {m : (K →+* ℂ) → ℝ} (hm : ∀ s, 0 < m s) (hf : ∀ s, (h.form s v v).re = m s) {c : FractionalIdeal (𝒪 K)⁰ K} (hc : c ≠ 0) (hΛ : ∀ z, z ∈ Λ ↔ ∃ t : K, z = t • v ∧ t ∈ c) : _root_.ReciprocityTheory.GeometryOfNumbersTheory.deg h Λ = -Real.log (FractionalIdeal.absNorm c) / finrank ℚ K - _root_.ReciprocityTheory.GeometryOfNumbersTheory.meanLog K m / 2

theorem **_root_.GlobalTheory.CM.two_mul_deg_dual_line** : ∀ (E K V : Type) [Field E] [NumberField E] [IsCMField E] [Field K] [NumberField K] [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] (h : _root_.ReciprocityTheory.GeometryOfNumbersTheory.HermFam K V) (Λ : Submodule (𝒪 K) V) (v : V) (τ : E →+* K) (a : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (β : E) (n : (K →+* ℂ) → ℝ), a ≠ 0 → v ≠ 0 → (∀ s, 0 < n s) → (∀ s : E →+* ℂ, 0 < (s β).re ∧ (s β).im = 0) → (∀ s, (h.form s v v).re = (s (τ β)).re * n s) → (∀ z, z ∈ Λ ↔ ∃ t : K, z = t • v ∧ b ∈ a, IsIntegral ℤ (t * τ (β * IsCMField.complexConj E b))) → 2 * _root_.ReciprocityTheory.GeometryOfNumbersTheory.deg h Λ = (2 * Real.log (FractionalIdeal.absNorm a : ℚ) + Real.log |(Algebra.norm ℚ β : ℚ)|) / Module.finrank ℚ E - _root_.ReciprocityTheory.GeometryOfNumbersTheory.meanLog K n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TwoMulDegDualLineCmIdealStatement

theorem **_root_.GlobalTheory.T.two_mul_deg_dual_line_cmIdeal** : ∀ (E K V : Type) [Field E] [NumberField E] [IsCMField E] [Field K] [NumberField K] [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] (h : HermFam K V) (Λ : Submodule (𝒪 K) V) (v : V) (τ : E →+* K) (a : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (β : E) (n : (K →+* ℂ) → ℝ), a ≠ 0 → v ≠ 0 → (∀ s, 0 < n s) → (∀ s : E →+* ℂ, 0 < (s β).re ∧ (s β).im = 0) → (∀ s, (h.form s v v).re = (s (τ β)).re * n s) → (∀ z, z ∈ Λ ↔ ∃ t : K, z = t • v ∧ ∀ b ∈ a, IsIntegral ℤ (t * τ (β * IsCMField.complexConj E b))) → 2 * deg h Λ = (2 * Real.log (FractionalIdeal.absNorm a : ℚ) + Real.log |(Algebra.norm ℚ β : ℚ)|) / Module.finrank ℚ E - meanLog K n

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TwoMulDegEigenLatticePolarizedAbelianScheme

theorem **two_mul_deg_eigLat_of_polAV** : DegSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → ∀ (K : Type) [Field K] [NumberField K] (ρ : 𝒪 K) (P : Out.PolAV (Sp (𝒪 K))) (act : OAct (𝒪 E) P.A), (∀ a, LamEq P.L (act.act a) (𝟙 P.A) P.L (𝟙 P.A) (act.act (H.bar a))) → Splits E K → ρ ^ 2 = -(H.d : 𝒪 K) → HasType H act ρ 1 0 → ∀ (F : GenFib P) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ τ : 𝒪 E →+* 𝒪 K, τ H.rt = ρ → ((2 * deg h (F.eigLat act τ) : ℝ) : ℂ) = logOm E χ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TwoMulDegEigenLatticePolarizedAbelianSchemeStatement

theorem **_root_.GlobalTheory.T.two_mul_deg_eigenLattice_polarizedAbelianScheme** : DegSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) [NeZero H.d] (χ : DirichletCharacter ℂ H.d), IsKroneckerChar χ → ∀ (K : Type) [Field K] [NumberField K] (ρ : 𝒪 K) (P : Out.PolAV (Sp (𝒪 K))) (act : OAct (𝒪 E) P.A), (∀ a, LamEq P.L (act.act a) (𝟙 P.A) P.L (𝟙 P.A) (act.act (H.bar a))) → Splits E K → ρ ^ 2 = -(H.d : 𝒪 K) → HasType H act ρ 1 0 → ∀ (F : GenFib P) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ τ : 𝒪 E →+* 𝒪 K, τ H.rt = ρ → ((2 * deg h (F.eigLat act τ) : ℝ) : ℂ) = logOm E χ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TwoMulDegreeEigenLatticeTensorRankOne

theorem **eq_zero_of_re_diag** {L : Type} [AddCommGroup L] (B : L → L → ℂ) (W : L → L) (l : ℂ) (hl : l.im ≠ 0) (hadd : ∀ a b c, B a (b + c) = B a b + B a c) (hsym : ∀ a b, starRingEnd ℂ (B a b) = B b a) (hW : ∀ a b, B a (W b) = l * B a b) (hd : ∀ a, (B a a).re = 0) (a b : L) : B a b = 0

theorem **span_embeddings_eq_top** {E K L I : Type} [Field E] [NumberField E] [Field K] [AddCommGroup L] [Module (𝒪 E) L] [Fintype I] (σ : I → (E →+* K)) (hσ : Function.Injective σ) (e : L →ₗ[𝒪 E] E) (a₀ : L) (h₀ : e a₀ ≠ 0) : Submodule.span K (Set.range fun a i => σ i (e a)) = ⊤

theorem **form_eq_of_span** {K V L I : Type} [Field K] [AddCommGroup V] [Module K V] [Fintype I] (h : HermFam K V) (s : K →+* ℂ) (Φ : (I → K) →ₗ[K] V) (ev : L → I → K) (b : I → ℝ) (hsp : Submodule.span K (Set.range ev) = ⊤) (hf : ∀ x y, h.form s (Φ (ev x)) (Φ (ev y)) = ∑ i, b i * (starRingEnd ℂ (s (ev x i)) * s (ev y i))) (z w : I → K) : h.form s (Φ z) (Φ w) = ∑ i, b i * (starRingEnd ℂ (s (z i)) * s (w i))

theorem **exists_eigenline** {E K O L V I : Type} [Field E] [NumberField E] [IsCMField E] [Field K] [NumberField K] [CommRing O] [AddCommGroup L] [Module (𝒪 E) L] [Module O L] [AddCommGroup V] [Module K V] [Module (𝒪 K) V] [IsScalarTower (𝒪 K) K V] [Fintype I] (σ : I → (E →+* K)) (hσ : Function.Injective σ) (i₀ : I) (τ₀ : O →+* 𝒪 K) (w : O) (hw : ∀ s : K →+* ℂ, (s (τ₀ w)).im ≠ 0) (e : L →ₗ[𝒪 E] E) (he : ∀ i (k : O) a, σ i (e (k • a)) = τ₀ k * σ i (e a)) (a₀ : L) (h₀ : e a₀ ≠ 0) (β : E) (hβ : ∀ s : E →+* ℂ, 0 < (s β).re ∧ (s β).im = 0) (hK : L → L → O) (hh : ∀ a b, (τ₀ (hK a b) : K) = ∑ i, σ i (β * e a * IsCMField.complexConj E (e b))) (N : ℕ) (x y : Fin N → L) (hdual : ∀ a, a = ∑ m, hK a (y m) • x m) (h : HermFam K V) (n : (K →+* ℂ) → ℝ) (hn : ∀ s, 0 < n s) (c : L →+ V) (hc : ∀ (k : O) a, c (k • a) = (τ₀ k : K) • c a) (r : 𝒪 E → V →ₗ[K] V) (hr : ∀ s a, c (s • a) = r s (c a)) (hnorm : ∀ s a,

theorem **two_mul_deg_eigLat** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K : Type) [Field K] [NumberField K] (ω : 𝒪 K) (C : Ell H (𝒪 K) ω) (D : Datum H) (ε : ℤˣ) (hK : D.La → D.La → OK H) (T : PAT H (Sp (𝒪 K))) (x : ℝ), Splits E K → D.p + D.q = 1 → IsKForm D ε hK → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) ∗ D.EF (H.w0 • x) x) → Nonempty (IsTen C D hK T) → C.HasPeriod x → ∀ (F : GenFib T.pol) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ τ : 𝒪 E →+* 𝒪 K, τ H.w0 = ω → 2 ∗ deg h (F.eigLat T.t.act τ) = x + Real.log H.d / 2 − Real.log (NumberField.rootDiscr E)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.TwoMulDegreeEigenLatticeTensorRankOneStatement

theorem **_root_.GlobalTheory.T.two_mul_degree_eigenLattice_tensor_rankOne** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (K : Type) [Field K] [NumberField K] (ω : 𝒪 K) (C : Ell H (𝒪 K) ω) (D : Datum H) (ε : ℤˣ) (hK : D.La → D.La → OK H) (T : PAT H (Sp (𝒪 K))) (x : ℝ), Splits E K → D.p + D.q = 1 → IsKForm D ε hK → (∀ x : D.La, x ≠ 0 → 0 < (ε : ℤ) ∗ D.EF (H.w0 • x) x) → Nonempty (IsTen C D hK T) → C.HasPeriod x → ∀ (F : GenFib T.pol) (h : HermFam K (Fin F.P₀.g → K)), F.IsHerm h → ∀ τ : 𝒪 E →+* 𝒪 K, τ H.w0 = ω → 2 ∗ deg h (F.eigLat T.t.act τ) = x + Real.log H.d / 2 − Real.log (NumberField.rootDiscr E)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UnfGroupContainsCongruenceUptoRootsOfUnity

private lemma **isUStab_one_cubical_line_bundles** {E : Type} [Field E] [NumberField E] [NumberField.IsCMField E] {r : ℕ} {G : Matrix (Fin r) (Fin r) E} {L : Submodule (NumberField.RingOfIntegers E) (Fin r → E)} : _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab G L (1 : GL (Fin r) E)

private lemma **isUStab_mul_cubical_line_bundles** {E : Type} [Field E] [NumberField E] [NumberField.IsCMField E] {r : ℕ} {G : Matrix (Fin r) (Fin r) E} {L : Submodule (NumberField.RingOfIntegers E) (Fin r → E)} {g h : GL (Fin r) E} (hg : _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab G L g) (hh : _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab G L h) : _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab G L (g ∗ h)

private lemma **isUStab_pow'** {E : Type} [Field E] [NumberField E] [NumberField.IsCMField E] {r : ℕ} {G : Matrix (Fin r) (Fin r) E} {L : Submodule (NumberField.RingOfIntegers E) (Fin r → E)} {g : GL (Fin r) E} (hg : _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab G L g) (n : ℕ) : _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab G L (g ^ n)

theorem **_root_.GlobalTheory.unf_group_contains_congruence_upto_rootsOfUnity** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum H) (l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H) (Xo : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Moduli Dw l) (c : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : _root_.Formalism.L2.PjFree.Unf Xo c σ r), _root_.Formalism.L2.PjFree.IsUnf U → l.A = 1 → 0 < r → ∀ k (γ : GL (Fin r) E), _root_.ReciprocityTheory.ChartRestriction.G.FD.IsUStab (U.X k).Gm (U.X k).Λ γ → (∀ x ∈ (U.X k).Λ, ∃ y ∈ (U.X k).Λ, Matrix.mulVec (γ : Matrix (Fin r) (Fin r) E) x − x = (l.M : E) • y) → ∃ γ' ∈ (U.X k).Γ, ∃ (u : E) (n : ℕ), 0 < n ∧ u ^ n = 1 ∧ (γ' : Matrix (Fin r) (Fin r) E) = u • (γ :

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UnifMetricScalar

theorem **liftPt_comp** (p : Over.mk y ⟶ G) : (liftPt h p).left ≫ φ = p.left

theorem **pt_ext** {a b : 𝟙_ (Over (Sp ℂ))} ⟶ A} (e : a.left ≫ φ = b.left ≫ φ) : a = b

theorem **liftPt_mul** (a b : Over.mk y ⟶ G) : liftPt h (a ∗ b) = liftPt h a ∗ liftPt h b

theorem **liftPt_injective** : Function.Injective (liftPt h)

theorem **liftPt_surjective** : Function.Surjective (liftPt h)

theorem **liftPt_one** : liftPt h 1 = 1

theorem **lift_le** {L : LB G.left} (p : Over.mk y ⟶ G) (j : (L.pull φ).ι) : ⊤ ≤ (liftPt h p).left ⁻¹ᵁ (L.pull φ).U j ↔ ⊤ ≤ p.left ⁻¹ᵁ L.U (j PUnit.unit)

theorem **val_tie** {y : Sp ℂ ⟶ Xo.X} {t' : SAB H (Sp ℂ)} (m : SABMap t' Xo.univ.t y) {th : LB.Sec t'.PP.L} {s : LB.Sec (PPx y)} (ht : SecTie m th s) : ∃ w : LB.Iso t'.PP.L (Xo.univ.t.PP.L.pull m.φ), RigIso m w ∧ ∀ (p : Sp ℂ ⟶ t'.G.left), p ≫ t'.G.hom = 𝟙 _ → ∃ pt : Sp ℂ ⟶ Gx y, pt ≫ gxFst y = p ≫ m.φ ∧ pt ≫ gxSnd y = 𝟙 _ ∧ ∀ j i, ⊤ ≤ p ⁻¹ᵁ ((Xo.univ.t.PP.L.pull m.φ).U (fun _ => j) ⊓ t'.PP.L.U i) → valAt' pt ((PPx y).U fun _ => j) (s.f fun _ => j) = valAt' p ((Xo.univ.t.PP.L.pull m.φ).U (fun _ => j) ⊓ t'.PP.L.U i) (w.s (fun _ => j, i)).val ∗ valAt' p (t'.PP.L.U i) (th.f i)

theorem **isDerivAt0_comp** {A : Over (Sp ℂ)} [GrpObj A] {L : LB A.left} {g : ℕ} (u : HolUnif A L g) {Y : Scheme.{0}} (φ : A.left ⟶ Y) {w : Fin g → ℂ} {v : epsR ℂ ⟶ A} {e : (Fin g → ℂ) → (Sp ℂ ⟶ Y)} (he : ∀ z, (u.ex z).left ≫ φ = e z) (hv : IsTangent v) (hd : u.IsDiff w v) : IsDerivAt0 e w ((show Sp (DualNumber ℂ) ⟶ _ from v.left) ≫ φ)

theorem **isHolTo_exp** {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (F : FamilyUnif Xo σ ty) (i : Fin F.n) {P : (F.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (F.Di i) ty P) : IsHolTo Set.univ fun z => (F.ex i P z).left

theorem **eq_of_isDerivAt0** {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (F : FamilyUnif Xo σ ty) (i : Fin F.n) {P : (F.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (F.Di i) ty P) {w : Fin g → ℂ} {v v' : Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left} (o : v ≫ Xo.univ.t.G.hom = spMap (algebraMap ℂ (DualNumber ℂ)) ≫ F.u i P) (o' : v' ≫ Xo.univ.t.G.hom = spMap (algebraMap ℂ (DualNumber ℂ)) ≫ F.u i P) (d : IsDerivAt0 (fun z => (F.ex i P z).left) w v) (d' : IsDerivAt0 (fun z => (F.ex i P z).left) w v') : v = v'

theorem **exists_unif** {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (F : FamilyUnif Xo σ ty) (i : Fin F.n) {P : (F.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (F.Di i) ty P) {y : Sp ℂ ⟶ Xo.X} (hy : y = F.u i P) {s : LB.Sec (PPx y)} (Φ : FibKey Xo ℂ y) (ht : SecTie Φ.m Φ.th s) {Hm : Matrix (Fin g) (Fin g) ℂ} {χl θ : (Fin g → ℂ) → ℂ} {fr : Xo.univ.t.PP.L.ι → (Fin g → ℂ) → ℂ} (herm : Hm.IsHermitian) (pos : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) (c₁ : ∀ j, DifferentiableOn ℂ (fr j) {z | ⊤ ≤ (F.ex i P z).left ⁻¹ᵁ Xo.univ.t.PP.L.U j}) (c₂ : ∀ j z, ⊤ ≤ (F.ex i P z).left ⁻¹ᵁ Xo.univ.t.PP.L.U j → fr j z ≠ 0) (c₃ : ∀ j j' z, ⊤ ≤ (F.ex i P z).left ⁻¹ᵁ (Xo.univ.t.PP.L.U j ⊓ Xo.univ.t.PP.L.U j') → fr j z = valAt' (F.ex i P z).left (Xo.univ.t.PP.L.U j ⊓ Xo.univ.t.PP.L.U j') (Xo.univ.t.PP.L.g j j').val ∗ fr j' z) (c₄ : ∀ j, ⊤ ≤ (F.ex i P 0).left ⁻¹ᵁ Xo.univ.t.PP.L.U j → fr j 0 = valAt' y ((Xo

theorem **isMetricAt_of** {y : Sp ℂ ⟶ Xo.X} {s : LB.Sec (PPx y)} {uu : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left)} {N : ℝ} {Gr : ∀ b : E →+* ℂ, Matrix (Fin (Dw.mult b)) (Fin (Dw.mult b)) ℂ} (Φ : FibKey Xo ℂ y) {g : ℕ} (hg : g = Φ.key.P.g) (u : HolUnif Φ.key.P.A Φ.key.P.L g) (Hm : Matrix (Fin g) (Fin g) ℂ) (χ : u.Λ → ℂ) (θ : (Fin g → ℂ) → ℂ) (Fd : Set (Fin g → ℂ)) (w : ∀ b : E →+* ℂ, Fin (Dw.mult b) → (epsR ℂ ⟶ Φ.key.P.A)) (W : ∀ b : E →+* ℂ, Fin (Dw.mult b) → Fin g → ℂ) (h : SecTie Φ.m Φ.th s ∧ u.IsAH Hm χ ∧ u.IsRig Φ.key.rig ∧ u.IsFn Φ.key.th θ ∧ IsAddFundamentalDomain u.Λ Fd volume ∧ N = (∫ z in Fd, ‖θ z‖ ^ 2 ∗ Real.exp (− Real.pi ∗ (hermForm Hm z z).re)) / (volume Fd).toReal ∧ (∀ b i, (w b i).left ≫ Φ.m.φ = uu b i ∧ u.IsDiff (W b i) (w b i)) ∧ ∀ b i k, Gr b i k = Real.pi ∗ hermForm Hm (W b k) (W b i)) : IsMetricAt y s uu N Gr

theorem **isMetric_of_isOrtho** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (ξ : ℝ), GoodEmb c σ → IsUnf U → 1 ≤ Dw.p + Dw.q → ∀ (k : U.Comp)

(x : (U.X k).dom), (∀ τ : E →+* ℂ, ‖τ (U.X k).ξ0‖ = ξ) → (U.X k).IsOrtho x.1 → IsMetric (U.un k x) 1 (fun _ => ((2 * Real.pi * ξ : ℝ) : ℂ) • 1)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UnifMetricScalarStatement

theorem **_root_.GlobalTheory.T.unif_metric_scalar** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (ξ : ℝ), GoodEmb c σ → IsUnf U → 1 ≤ Dw.p + Dw.q → ∀ (k : U.Comp) (x : (U.X k).dom), (∀ τ : E →+* ℂ, ‖τ (U.X k).ξ0‖ = ξ) → (U.X k).IsOrtho x.1 → IsMetric (U.un k x) 1 (fun _ => ((2 * Real.pi * ξ : ℝ) : ℂ) • 1)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UniversalTripleFibreKeyDataBaseChangeKeyPullback

theorem **fibKey_keyPB** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] [Nontrivial S'] (φ : S →+* S') (x : Sp S ⟶ Xo.X) (x' : Sp S' ⟶ Xo.X) (Φ : FibKey Xo S x) (Φ' : FibKey Xo S' x') (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')), x' = spMap φ ≫ x → SecTie Φ.m Φ.th s → SecTie Φ'.m Φ'.th s' → SecBC φ 1 x x' s s' → ∃ (hg : Φ'.T.g = Φ.T.g) (v' : Fin Φ'.T.g → (epsR S' ⟶ Φ'.T.t.G)) (hv' : IsLieBasis v'), KeyPB ({ Φ' with v

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UniversalTripleFibreKeyDataBaseChangeKeyPullbackStatement

theorem **_root_.GlobalTheory.T.universalTripleFibreKeyData_baseChange_keyPullback** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] [Nontrivial S'] (φ : S →+* S') (x : Sp S ⟶ Xo.X) (x' : Sp S' ⟶ Xo.X) (Φ : FibKey Xo S x) (Φ' : FibKey Xo S' x') (s : LB.Sec (PPx x)) (s' : LB.Sec (PPx x')), x' = spMap φ ≫ x → SecTie Φ.m Φ.th s → SecTie Φ'.m Φ'.th s' → SecBC φ 1 x x' s s' → ∃ (hg : Φ'.T.g = Φ.T.g) (v' : Fin Φ'.T.g → (epsR S' ⟶ Φ'.T.t.G)) (hv' : IsLieBasis v'), KeyPB ({ Φ' with v

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UniversalTripleFibreKeyDataNonemptyField

theorem **nonempty_fibKey** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (K : Type) [Field K] (x : Sp K ⟶ Xo.X), 1 ≤ Dw.p + Dw.q → Nonempty (FibKey Xo K x)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.UniversalTripleFibreKeyDataNonemptyFieldStatement

theorem **_root_.GlobalTheory.T.universalTripleFibreKeyData_nonempty_field** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (K : Type) [Field K] (x : Sp K ⟶ Xo.X), 1 ≤ Dw.p + Dw.q → Nonempty (FibKey Xo K x)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ValuationDetLeOneOfEntries

theorem **valuation_det_le_one_of_entries** {K G J : Type} [Field K] [LinearOrderedCommGroupWithZero G] [Fintype J] [DecidableEq J] (v : Valuation K G) (N : Matrix J J K) (h : ∀ i j, v (N i j) ≤ 1) : v N.det ≤ 1

theorem **valuation_prod_mul_det_le** {K G J : Type} [Field K] [LinearOrderedCommGroupWithZero G] [Fintype J] [DecidableEq J] (v : Valuation K G) (U M : Matrix J J K) (c : J → K) (hM : M.det ≠ 0) (h : ∀ k m, v (Matrix.vecMul (c k • U k) M⁻¹ m) ≤ 1) : v ((∏ k, c k) * U.det) ≤ v M.det

theorem **letters_det_valuation_le** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] [StarRing (𝒪 E)] (H : Host E) (l : Level H) (c : Coeff l) (nΨ n' f : ℕ) (o : LamObj c nΨ n') (La : ℝ), LamObjGood o f La → ∀ L : LamLetters o, L.IsGood → ∀ (p : HeightOneSpectrum (𝒪 o.K₁)) (cd : o.K₁) (cl : Fin o.xB.P.g → o.K₁), IsLocGen p cd o.LamL.I → (∀ k, IsLocGen p (cl k) (L.ℓ k).I) → p.valuation o.K₁ ((∏ k, cl k) * L.U.det) ≤ p.valuation o.K₁ cd

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ValuationEqOfIsLocGen

theorem **exists_unit_of_pow** {R K' : Type} [CommRing R] [IsDomain R] [IsIntegrallyClosed R] [Field K'] [Algebra R K'] [IsFractionRing R K'] {z : K'} {m : ℕ} (hm : 0 < m) (w : Rˣ) (h : z ^ m = algebraMap R K' w) : ∃ u : Rˣ, z = algebraMap R K' u

theorem **unit_iff** {K : Type} [Field K] [NumberField K] (p : HeightOneSpectrum (𝒪 K)) {R K' : Type} [CommRing R] [Field K'] [Algebra R K'] (hi : Function.Injective (algebraMap R K')) (f : K →+* K') (hv : ∀ y : K, f y ∈ Set.range (algebraMap R K') ↔ p.valuation K y ≤ 1) (y : K) : (∃ u : Rˣ, f y = algebraMap R K' u) ↔ p.valuation K y = 1

theorem **valuation_eq_of_isLocGen** {K : Type} [Field K] [NumberField K] {p : HeightOneSpectrum (𝒪 K)} {c c' : K} {I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K} (h : IsLocGen p c I) (h' : IsLocGen p c' I) : p.valuation K c = p.valuation K c'

theorem **exists_pow_rel** (K : Type) [Field K] [NumberField K] (x : KeyIn K) : ∃ (n : ℕ) (k : K), 1 ≤ n ∧ k ≠ 0 ∧ ∀ (R K' : Type) [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] [Field K'] [Algebra R K'] [IsFractionRing R K'] (f : K →+* K') (x' : KeyIn K') (a d : K'), KeyPB x' x f → KeyIntegral R x' a d → ∃ u : Rˣ, a ^ (8 * n) * algebraMap R K' u = d ^ (4 * n) * f k

theorem **integral_section_unit** : ∀ (K : Type) [Field K] [NumberField K] (x : KeyIn K) (Tl abelian₂ : MLine K) (p : HeightOneSpectrum (𝒪 K)), KeyLocAt x Tl abelian₂ p → ∀ (R K' : Type) [CommRing R] [IsDomain R] [IsDiscreteValuationRing R] [Field K'] [Algebra R K'] [IsFractionRing R K'] (f : K →+* K') (x' : KeyIn K') (a d : K') (ca cd : K) (ud : Rˣ), (∀ y : K, f y ∈ Set.range (algebraMap R K') ↔ p.valuation K y ≤ 1) → KeyPB x' x f → KeyIntegral R x' a d → IsLocGen p cd abelian₂.I → d = f cd * algebraMap R K' ud → IsLocGen p ca Tl.I → ∃ u : Rˣ, a = f ca * algebraMap R K' u

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.ValuationLeIffExistsSmulMem

theorem **isLocGen_iff_valuation_eq** (p : HeightOneSpectrum (𝒪 K)) (c : K) (I : FractionalIdeal (𝒪 K)⁰ K) : IsLocGen p c I ↔ p.valuation K c = exp (-FractionalIdeal.count K p I)

theorem **valuation_le_iff_exists_smul_mem** (p : HeightOneSpectrum (𝒪 K)) {I : FractionalIdeal (𝒪 K)⁰ K} (hI : I ≠ 0) (c : K) : p.valuation K c ≤ exp (-FractionalIdeal.count K p I) ↔ ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • c ∈ I

theorem **mem_iff_forall_exists_smul_mem** {V : Type} [AddCommGroup V] [Module (𝒪 K) V] (Λ : Submodule (𝒪 K) V) (z : V) : z ∈ Λ ↔ ∀ p : HeightOneSpectrum (𝒪 K), ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • z ∈ Λ

theorem **local_global_ideal_lattice** : ∀ (K : Type) [Field K] [NumberField K], (∀ (I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K) (cl : HeightOneSpectrum (𝒪 K) → K), I ≠ 0 → (∀ p, IsLocGen p (cl p) I) → ∀ c : K, c ∈ I ↔ ∀ p : HeightOneSpectrum (𝒪 K), p.valuation K c ≤ p.valuation K (cl p)) ∧ (∀ (I : FractionalIdeal (nonZeroDivisors (𝒪 K)) K) (p : HeightOneSpectrum (𝒪 K)), I ≠ 0 → (∃ cl ∈ I, IsLocGen p cl I) ∧ ∀ cl, IsLocGen p cl I ↔ cl ≠ 0 ∧ ∀ c : K, p.valuation K c ≤ p.valuation K cl ↔ ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • c ∈ I) ∧ ∀ (V : Type) [AddCommGroup V] [Module (𝒪 K) V] (Λ : Submodule (𝒪 K) V) (z : V), z ∈ Λ ↔ ∀ p : HeightOneSpectrum (𝒪 K), ∃ s : 𝒪 K, s ∉ p.asIdeal ∧ s • z ∈ Λ

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.VecLineNormSqEqForm

theorem **vecLine_norm_sq_eq_form** : ThetaUnifAV → Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A : Out.PolAV (Sp (𝒪 K))) (x : KeyIn K) (φ : x.P.A.left ⟶ A.A.left) (hf : IsPolFibre (algebraMap (𝒪 K) K) A x.P φ) (h : HermFam K (Fin x.P.g → K)), (GenFib.mk x.P φ hf x.v x.basis : GenFib A).IsHerm h → ∀ (i : Fin x.P.g) (ℓ : MLine K), IsVecLine A x φ i ℓ → ∀ s : K →+* ℂ, ℓ.n s ^ 2 = (h.form s (Pi.single i 1) (Pi.single i 1)).re

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.VolumeFundamentalDomainStdRealBasis

theorem **volume_fundamentalDomain_stdRealBasis** (g : ℕ) : volume (ZSpan.fundamentalDomain (stdRealBasis g)) = 1

theorem **re_dotProduct_star** {g : ℕ} (z w : Fin g → ℂ) : (z ⬝ᵥ star w).re = ∑ k, (stdRealBasis g).repr z k * (stdRealBasis g).repr w k

theorem **im_hermForm** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z w : Fin g → ℂ) : (hermForm Hm z w).im = (z ⬝ᵥ star ((Complex.I • Hm.map star) *ᵥ w)).re

theorem **det_gram_im** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (v : (Σ _ : Fin g, Fin 2) → Fin g → ℂ) : (Matrix.of fun i j => (hermForm Hm (v i) (v j)).im).det = ((stdRealBasis g).det v) ^ 2 * ‖Hm.det‖ ^ 2

theorem **posDef_of_hermForm** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (hpos : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) : Hm.PosDef

theorem **det_re_pos** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (hpos : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) : 0 < Hm.det.re ∧ ‖Hm.det‖ = Hm.det.re

theorem **volume_mul_det_eq_one_of_unimodular** (g : ℕ) (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ] (hfull : Submodule.span ℝ (Λ : Set (Fin g → ℂ)) = ⊤) (Hm : Matrix (Fin g) (Fin g) ℂ) (Ef : LinearMap.BilinForm ℤ Λ) (Fd : Set (Fin g → ℂ)) (hH : Hm.IsHermitian) (hpos : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) (hE : ∀ a b : Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) (hbij : Function.Bijective Ef) (hFd : IsAddFundamentalDomain Λ Fd volume) : (volume Fd).toReal * Hm.det.re = 1

theorem **det_gram_pi_hermForm** {g : ℕ} (Hm W : Matrix (Fin g) (Fin g) ℂ) : (Matrix.of fun i j => (Real.pi : ℂ) * hermForm Hm (W i) (W j)).det = (Real.pi : ℂ) ^ g * Hm.det * ((‖W.det‖ ^ 2 : ℝ) : ℂ)

theorem **volume_div_eq_inv_of_unimodular** (g : ℕ) (Λ : Submodule ℤ (Fin g → ℂ)) [DiscreteTopology Λ] (hfull : Submodule.span ℝ (Λ : Set (Fin g → ℂ)) = ⊤) (Hm : Matrix (Fin g) (Fin g) ℂ) (Ef : LinearMap.BilinForm ℤ Λ) (Fd : Set (Fin g → ℂ)) (hH : Hm.IsHermitian) (hpos : ∀ z : Fin g → ℂ, z ≠ 0 → 0 < (hermForm Hm z z).re) (hE : ∀ a b : Λ, (Ef a b : ℝ) = (hermForm Hm a b).im) (hbij : Function.Bijective Ef) (hFd : IsAddFundamentalDomain Λ Fd volume) (W : Matrix (Fin g) (Fin g) ℂ) (hW : W.det ≠ 0) : 0 < Real.pi ^ g * Hm.det.re * ‖W.det‖ ^ 2 ∧ (volume Fd).toReal / (Real.pi ^ g * ‖W.det‖ ^ 2) = (Real.pi ^ g * Hm.det.re * ‖W.det‖ ^ 2)⁻¹ ∧ (Matrix.of fun i j => (Real.pi : ℂ) * hermForm Hm (W i) (W j)).det = ((Real.pi ^ g * Hm.det.re * ‖W.det‖ ^ 2 : ℝ) : ℂ)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeightModuleIsLatticeOfShortSeriesSpan

theorem **latAt_record** : ∀ (D : D7) (T : Core D) (h : LawsCore T), GamAt T → ∀ (large₂ : SizeData₂ D) (CA : ℝ), WeightAlgebra.ParA large₂ CA → ∀ (j : ℕ) (ka : WeightFamilies.Wt D j) (P : ModuliIntegralCharts.Prov D) (X : WeightAlgebra.PJ D (recB h) P j) (S : WeightAlgebra.Scal (recB h)) (wL wF : WeightFamilies.Wt D j), S.sK ≠ 0 → PullbackSections.SpanAt X ka → WeightAlgebra.H2 X S ka wL wF → Formalism.MainNotions.Gaps.EtaleWeightGaps.LatAt (recB h) j ka

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeightModuleIsLatticeOfShortSeriesSpanStatement

theorem **_root_.GlobalTheory.T.weightModule_isLattice_of_shortSeries_span** : ∀ (D : D7) (T : Core D) (h : LawsCore T), GamAt T → ∀ (large₂ : SizeData₂ D) (CA : ℝ), WeightAlgebra.ParA large₂ CA → ∀ (j : ℕ) (ka : WeightFamilies.Wt D j) (P : ModuliIntegralCharts.Prov D) (X : WeightAlgebra.PJ D (recB h) P j) (S : WeightAlgebra.Scal (recB h)) (wL wF : WeightFamilies.Wt D j), S.sK ≠ 0 → PullbackSections.SpanAt X ka → WeightAlgebra.H2 X S ka wL wF → Formalism.MainNotions.Gaps.EtaleWeightGaps.LatAt (recB h) j ka

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilEqIffHermFormEqComplex

theorem **ex_zero** : u.ex 0 = 1

theorem **ex_natSmul** (n : ℕ) (z : Fin g → ℂ) : u.ex ((n : ℂ) • z) = u.ex z ^ n

theorem **ex_div_pow** {n : ℕ} (hn : 0 < n) (a : u.Λ) : u.ex ((n : ℂ)⁻¹ • (a : Fin g → ℂ)) ^ n = 1

theorem **exists_eq_ex_div** {n : ℕ} (hn : 0 < n) (x : 𝟙_ (Over (Sp ℂ)) ⟶ A) (hx : x ^ n = 1) : ∃ a : u.Λ, x = u.ex ((n : ℂ)⁻¹ • (a : Fin g → ℂ))

theorem **map_mem** {B : Over (Sp ℂ)} [GrpObj B] {M : LB B.left} {g' : ℕ} (v : HolUnif B M g') {f : A ⟶ B} {F : (Fin g → ℂ) →ₗ[ℂ] (Fin g' → ℂ)} (hF : ∀ z, v.ex (F z) = u.ex z ≫ f) (a : u.Λ) : F a ∈ v.Λ

theorem **eq_of_rootVal** {m m' : ℤ} (h : ∀ n : ℕ, 0 < n → rootVal m n = rootVal m' n) : m = m'

theorem **hermForm_eq_of_im** {p q a b : ℕ} (HA : Matrix (Fin a) (Fin a) ℂ) (HB : Matrix (Fin b) (Fin b) ℂ) (F₁ : (Fin p → ℂ) →ₗ[ℂ] (Fin a → ℂ)) (F₂ : (Fin q → ℂ) →ₗ[ℂ] (Fin a → ℂ)) (G₁ : (Fin p → ℂ) →ₗ[ℂ] (Fin b → ℂ)) (G₂ : (Fin q → ℂ) →ₗ[ℂ] (Fin b → ℂ)) {S : Set (Fin p → ℂ)} {T : Set (Fin q → ℂ)} (hS : Submodule.span ℝ S = ⊤) (hT : Submodule.span ℝ T = ⊤) (h : ∀ z ∈ S, ∀ w ∈ T, (hermForm HA (F₁ z) (F₂ w)).im = (hermForm HB (G₁ z) (G₂ w)).im) (z w) : hermForm HA (F₁ z) (F₂ w) = hermForm HB (G₁ z) (G₂ w)

theorem **weilIs_iff** (hU : ThetaUnifAV) (P : Out.PolAV (Sp ℂ)) (u : HolUnif P.A P.L P.g) {Hm : Matrix (Fin P.g) (Fin P.g) ℂ} {χ : u.Λ → ℂ} (h : u.IsAH Hm χ) (a b : u.Λ) : ∃ m : ℤ, (m : ℝ) = (hermForm Hm a b).im ∧ ∀ (n : ℕ) (c : ℂˣ), 0 < n → (Out.WeilIs P.A P.L n (u.ex ((n : ℂ)⁻¹ • (a : Fin P.g → ℂ))) (u.ex ((n : ℂ)⁻¹ • (b : Fin P.g → ℂ))) (unitOn c) ↔ c = rootVal m n)

theorem **weilEq_iff_hermForm_eq** : ThetaUnifAV → Out.Laws → ∀ (X Y A B : Out.PolAV (Sp ℂ)) (uX : HolUnif X.A X.L X.g) (uY : HolUnif Y.A Y.L Y.g) (uA : HolUnif A.A A.L A.g) (uB : HolUnif B.A B.L B.g) (HA : Matrix (Fin A.g) (Fin A.g) ℂ) (HB : Matrix (Fin B.g) (Fin B.g) ℂ) (χA : uA.Λ → ℂ) (χB : uB.Λ → ℂ), uA.IsAH HA χA → uB.IsAH HB χB → ∀ (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A) (F₁ : (Fin X.g → ℂ) →ₗ[ℂ] (Fin A.g → ℂ)) (F₂ : (Fin Y.g → ℂ) →ₗ[ℂ] (Fin A.g → ℂ)) (G₁ : (Fin X.g → ℂ) →ₗ[ℂ] (Fin B.g → ℂ)) (G₂ : (Fin Y.g → ℂ) →ₗ[ℂ] (Fin B.g → ℂ)), (∀ z, uA.ex (F₁ z) = uX.ex z ≫ f₁) → (∀ z, uA.ex (F₂ z) = uY.ex z ≫ f₂) → (∀ z, uB.ex (G₁ z) = uX.ex z ≫ g₁) → (∀ z, uB.ex (G₂ z) = uY.ex z ≫ g₂) → (WeilEq A.L f₁ f₂ B.L g₁ g₂ (𝟙 (Sp ℂ)) ↔ ∀ z w, hermForm HA (F₁ z) (F₂ w) = hermForm HB (G₁ z) (G₂ w))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilEqIffHermFormEqComplexStatement

theorem **_root_.GlobalTheory.T.weilEq_iff_hermForm_eq_complex** : ThetaUnifAV → Out.Laws → ∀ (X Y A B : Out.PolAV (Sp ℂ)) (uX : HolUnif X.A X.L X.g) (uY : HolUnif Y.A Y.L Y.g) (uA : HolUnif A.A A.L A.g) (uB : HolUnif B.A B.L B.g) (HA : Matrix (Fin A.g) (Fin A.g) ℂ) (HB : Matrix (Fin B.g) (Fin B.g) ℂ) (χA : uA.Λ → ℂ) (χB : uB.Λ → ℂ), uA.IsAH HA χA → uB.IsAH HB χB → ∀ (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A) (F₁ : (Fin X.g → ℂ) →ₗ[ℂ] (Fin A.g → ℂ)) (F₂ : (Fin Y.g → ℂ) →ₗ[ℂ] (Fin A.g → ℂ)) (G₁ : (Fin X.g → ℂ) →ₗ[ℂ] (Fin B.g → ℂ)) (G₂ : (Fin Y.g → ℂ) →ₗ[ℂ] (Fin B.g → ℂ)), (∀ z, uA.ex (F₁ z) = uX.ex z ≫ f₁) → (∀ z, uA.ex (F₂ z) = uY.ex z ≫ f₂) →

(∀ z, uB.ex (G₁ z) = uX.ex z ≫ g₁) → (∀ z, uB.ex (G₂ z) = uY.ex z ≫ g₂) → (WeilEq A.L f₁ f₂ B.L g₁ g₂ (𝟙 (Sp ℂ)) ↔ ∀ z w, hermForm HA (F₁ z) (F₂ w) = hermForm HB (G₁ z) (G₂ w))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilIsOfLbIsoCubeLB

theorem **val_ratio** {X X' Y Y₀ B : Scheme.{0}} {L : LB X} {N : LB X'} (hL : L.IsLB) (hN : N.IsLB) {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y₀ ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y₀ ⟶ X')} {ε' : K' → ℤ} (H : LBIso (LB.LinComb N φ' ε') (LB.LinComb L φ ε)) (h : Y ⟶ Y₀) (t : LB.Gen (LB.LinComb L (fun k => h ≫ φ k) ε)) (ht : t.IsGen) (t' : LB.Gen (LB.LinComb N (fun k => h ≫ φ' k) ε')) (ht' : t'.IsGen) (p : Y ⟶ B) (hp : Function.Surjective p.appTop.hom) {Z : Scheme.{0}} (r r' : Z ⟶ Y) (e₁ : r ≫ h = r' ≫ h) (e₂ : r ≫ p = r' ≫ p) (z : Z) {J J'} (hJ : Good L z J) (hJ' : Good N z J') : val t r z J ∗ (val t' r z J')⁻¹ = val t r' z J ∗ (val t' r' z J')⁻¹

theorem **exists_gen_lam** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {N : RigLB A} (cb : CubicalStr A N) {T : Over S} (p₀ : T.left) (hT : ∀ p : T.left, p = p₀) (a g : B ⟶ A) [IsMonHom g] (n : ℕ) (y : T ⟶ B) (hy : (y ≫ a) ^ n = 1) : ∃ t : LB.Gen (LB.LinComb N.L (fun I : ULift.{0} (Finset (Fin 2)) => (lift (fst T B ≫ y) (snd T B ^ n)).left ≫ (psum A ![fst B B ≫ a, snd B B ≫ g] I.down).left) (fun I => (-1) ^ (2 - I.down.card))), t.IsGen ∧ ∀ (z : T ⟶ B) (e : Γ(T.left, ⊤)ˣ), z ^ n = 1 → WeilIs A N.L n (y ≫ a) (z ≫ g) e → ∀ (p : T.left) (J), Good N.L p J → val t (lift (𝟙 T) z).left p J = val t (lift (𝟙 T) (1 : T ⟶ B)).left p J ∗ gm (𝟙 T.left) p ⊤ e

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilPairingEqOfIsomorphicCubePairings

theorem **exists_gen_lam_two** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) {N : RigLB A} (cb : CubicalStr A N) {T : Over S} (p₀ : T.left) (hT : ∀ p : T.left, p = p₀) (a : X ⟶ A) (g : Y ⟶ A) [IsMonHom g] (n : ℕ) (y : T ⟶ X) (hy : (y ≫ a) ^ n = 1) : ∃ t : LB.Gen (LB.LinComb N.L (fun I : ULift.{0} (Finset (Fin 2)) => (lift (fst T Y ≫ y) (snd T Y ^ n)).left ≫ (psum A ![fst X Y ≫ a, snd X Y ≫ g] I.down).left) (fun I => (-1) ^ (2 - I.down.card))), t.IsGen ∧ ∀ (z : T ⟶ Y) (e : Γ(T.left, ⊤)ˣ), z ^ n = 1 → WeilIs A N.L n (y ≫ a) (z ≫ g) e → ∀ (p : T.left) (J), Good N.L p J → val t (lift (𝟙 T) z).left p J = val t (lift (𝟙 T) (1 : T ⟶ Y)).left p J ∗ gm (𝟙 T.left) p ⊤ e

theorem **weil_value_eq_of_lamEq** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hB : IsAbelianScheme B) (hY : IsAbelianScheme Y) {N : RigLB A} {L : RigLB B} (cN : CubicalStr A N) (cL : CubicalStr B L) (f₁ : X ⟶ A) (f₂ : Y ⟶ A) (g₁ : X ⟶ B) (g₂ : Y ⟶ B) [IsMonHom f₂] [IsMonHom g₂] (H : LamEq N.L f₁ f₂ L.L g₁ g₂) {T : Over S} (p₀ : T.left) (hT : ∀ p : T.left, p = p₀) (n : ℕ) (y : T ⟶ X) (z : T ⟶ Y) (hf : (y ≫ f₁) ^ n = 1) (hg : (y ≫ g₁) ^ n = 1) (hz : z ^ n = 1) (e e' : Γ(T.left, ⊤)ˣ) (he : WeilIs B L.L n (y ≫ g₁) (z ≫ g₂) e) (he' : WeilIs A N.L n (y ≫ f₁) (z ≫ f₂) e') : e = e'

theorem **weilEq_of_lamEq** : Out.Laws → ∀ (k : Type) [Field k] [IsAlgClosed k] [CharZero k] (X Y A B : Out.PolAV (Sp k)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → LamEq A.L f₁ f₂ B.L g₁ g₂ → WeilEq A.L f₁ f₂ B.L g₁ g₂ (𝟙 (Sp k))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilPairingEqOfIsomorphicCubePairingsStatement

theorem **_root_.GlobalTheory.T.weil_pairing_eq_of_isomorphic_cube_pairings** : Out.Laws → ∀ (k : Type) [Field k] [IsAlgClosed k] [CharZero k] (X Y A B : Out.PolAV (Sp k)) (f₁ : X.A ⟶ A.A) (f₂ : Y.A ⟶ A.A) (g₁ : X.A ⟶ B.A) (g₂ : Y.A ⟶ B.A), IsMonHom f₁ → IsMonHom f₂ → IsMonHom g₁ → IsMonHom g₂ → LamEq A.L f₁ f₂ B.L g₁ g₂ → WeilEq A.L f₁ f₂ B.L g₁ g₂ (𝟙 (Sp k))

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilPairingIffOfCubeLBPairingIso

theorem **weilIs_iff_of_lamEq_id** : Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A B : Out.PolAV (Sp (𝒪 K))) (f : A.A ⟶ B.A) (g : A.A ⟶ A.A), IsMonHom f → IsMonHom g → LamEq B.L f f A.L g (𝟙 A.A) → ∀ (Tb : Over (Sp (𝒪 K))) (k : ℕ) (x y : Tb ⟶ A.A) (c : Γ(Tb.left, ⊤)ˣ), 0 < k → pnsmulPt k x = toUnit Tb ≫ η[A.A] → pnsmulPt k y = toUnit Tb ≫ η[A.A] → (Out.WeilIs B.A B.L k (x ≫ f) (y ≫ f) c ↔ Out.WeilIs A.A A.L k (x ≫ g) y c)

Development.AbelianVarietyTheory.Proofs.CubicalLineBundles.WeilPairingIffOfCubeLBPairingIsoStatement

theorem **_root_.GlobalTheory.T.weilPairing_iff_of_cubeLBPairingIso** : Out.Laws → ∀ (K : Type) [Field K] [NumberField K] (A B : Out.PolAV (Sp (𝒪 K))) (f : A.A ⟶ B.A) (g : A.A ⟶ A.A), IsMonHom f → IsMonHom g → LamEq B.L f f A.L g (𝟙 A.A) → ∀ (Tb : Over (Sp (𝒪 K))) (k : ℕ) (x y : Tb ⟶ A.A) (c : Γ(Tb.left, ⊤)ˣ), 0 < k → pnsmulPt k x = toUnit Tb ≫ η[A.A] → pnsmulPt k y = toUnit Tb ≫ η[A.A] → (Out.WeilIs B.A B.L k (x ≫ f) (y ≫ f) c ↔ Out.WeilIs A.A A.L k (x ≫ g) y c)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AbelianHolomorphicFrameReadingConstOfFormsCancel

theorem **GlobalTheory.AnVal.genRd_const** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)), t.IsGen → ∀ (N : ℕ) (q : (Fin N → ℂ) → (Sp ℂ ⟶ Z)) (Λ : K → (Fin N → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (Ga : AddSubgroup (Fin N → ℂ)), IsHolFam q → (∀ z k, q z ≫ φ k = (Φ.e (Λ k z)).left) → Submodule.span ℝ (Ga : Set (Fin N → ℂ)) = ⊤ → (∀ gm ∈ Ga, (∀ z, q (z + gm) = q z) ∧ ∀ k, Λ k gm ∈ Φ.lat) → (∀ z w, ∑ k, (ε k : ℂ) ∗ hermForm Φ.Hm (Λ k z) (Λ k w) = 0) → (∃ v : ℂ, v ≠ 0 ∧ ∀ z, Φ.GenRd φ ε t (q z) (fun k => Λ k z) v) ∧ ∀ gm ∈ Ga, ∏ k, Φ.χ (Λ k gm) ^ ε k = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AbelianHolomorphicFrameReadingConstOfFormsCancelStatement

theorem **_root_.GlobalTheory.T.abelianHolomorphicFrame_reading_const_of_forms_cancel** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)), t.IsGen → ∀ (N : ℕ) (q : (Fin N → ℂ) → (Sp ℂ ⟶ Z)) (Λ : K → (Fin N → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (Ga : AddSubgroup (Fin N → ℂ)), IsHolFam q → (∀ z k, q z ≫ φ k = (Φ.e (Λ k z)).left) → Submodule.span ℝ (Ga : Set (Fin N → ℂ)) = ⊤ → (∀ gm ∈ Ga, (∀ z, q (z + gm) = q z) ∧ ∀ k, Λ k gm ∈ Φ.lat) → (∀ z w, ∑ k, (ε k : ℂ) ∗ hermForm Φ.Hm (Λ k z) (Λ k w) = 0) → (∃ v : ℂ, v ≠ 0 ∧ ∀ z, Φ.GenRd φ ε t (q z) (fun k => Λ k z) v) ∧ ∀ gm ∈ Ga, ∏ k, Φ.χ (Λ k gm) ^ ε k = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ActLevAndActBC

theorem **actLev_of_le** {D : D7} {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c c' : Coeff l} (hle : c.F ≤ c'.F) {G : Type} [Group G] (lev : G → (Dw.La →ₗ[𝒪 D.host.E] Dw.La) × (Dw.La →ₗ[𝒪 D.host.E] Dw.La)) (hL : ActLev (Xo

theorem **actLev_and_actBC** (D : D7) (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host D.host.E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (G : Type) [Group G] (lev : G → (Dw.La →ₗ[𝒪 D.host.E] Dw.La) × (Dw.La →ₗ[𝒪 D.host.E] Dw.La)) (hL : ActLev (Xo

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AdmissibleColumnsLeAndHarmonicMapCover

theorem **map_lap_one** {T : Type} (r : T → ℕ) (n : ℕ) {k k' : Type} [Field k] [Field k'] (f : k →+* k') (τ τ' : T) (a : Fin (r τ)) (a' : Fin (r τ')) (P : MvPolynomial (Letter T r n) k) : lap r n k' 1 τ τ' a a' (MvPolynomial.map f P) = MvPolynomial.map f (lap r n k 1 τ τ' a a' P)

theorem **map_rowShift** {T : Type} (r : T → ℕ) (n : ℕ) {k k' : Type} [Field k] [Field k'] (f : k →+* k') (σ : T) (i i' : Fin (r σ)) (t : k) (P : MvPolynomial (Letter T r n) k) : rowShift r n k' σ i i' (f t) (MvPolynomial.map f P) = MvPolynomial.map f (rowShift r n k σ i i' t P)

theorem **map_mem_HHa** {T : Type} (r : T → ℕ) (n : ℕ) {k k' : Type} [Field k] [Field k'] (f : k →+* k') (hf : Function.Surjective f) (cc : T → T) (isX : T → Prop) (a : T → ℕ → ℕ) (P : MvPolynomial (Letter T r n) k) (hP : P ∈ HHa r n k cc isX a) : MvPolynomial.map f P ∈ HHa r n k' cc isX a

theorem **finrank_le_of_span_sup_smul_top** {R M : Type} [CommRing R] [IsDomain R] [AddCommGroup M] [Module R M] [Module.Finite R M] [Module.Projective R M] {n : ℕ} (x : Fin n → M) (I : Ideal R) (hI : I ≠ ⊤) (h : Submodule.span R (Set.range x) ⊔ (I • ⊤ : Submodule R M) = ⊤) : Module.finrank R M ≤ n

theorem **Mfr_ne_top** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (l : Level H) : Mfr l ≠ ⊤

theorem **admissible_columns_le_and_harmonic_map_cover** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), (∀ (rk : Bool) (x0 : Fin n → Dw.La), AdmX (l

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AeCoefficientsCommonZerosSimple

theorem **colInc_apply** {n N : ℕ} (j0 : Fin N) (v : EuclideanSpace ℂ (Fin n)) (k : Fin n) (j : Fin N) : colInc j0 v (k.succ, j) = if j = j0 then v k else 0

theorem **colProj_colInc** {n N : ℕ} (j0 : Fin N) (v : EuclideanSpace ℂ (Fin n)) : colProj j0 (colInc j0 v) = v

theorem **ae_simple_col** {n N : ℕ} (U : Set (EuclideanSpace ℂ (Fin n))) (hU : IsOpen U) (w : Fin N → EuclideanSpace ℂ (Fin n) → ℂ) (hw : ∀ j, DifferentiableOn ℂ (w j) U) (j0 : Fin N) : ∀ᵐ t : Fin (n + 1) × Fin N → ℂ, ∀ z ∈ U, w j0 z ≠ 0 → ∑ j, t (0, j) * w j z ≠ 0 → (∀ k : Fin n, ∑ j, t (k.succ, j) * w j z = 0) → (fderiv ℂ (fun y => (WithLp.toLp 2 fun k : Fin n => (∑ j, t (k.succ, j) * w j y) / ∑ j, t (0, j) * w j y : EuclideanSpace ℂ (Fin n))) z).det ≠ 0

theorem **ae_simple** {n N : ℕ} (U : Set (EuclideanSpace ℂ (Fin n))) (hU : IsOpen U) (w : Fin N → EuclideanSpace ℂ (Fin n) → ℂ) (hw : ∀ j, DifferentiableOn ℂ (w j) U) : ∀ᵐ t : Fin (n + 1) × Fin N → ℂ, ∀ z ∈ U, ∑ j, t (0, j) * w j z ≠ 0 → (∀ k : Fin n, ∑ j, t (k.succ, j) * w j z = 0) → (fderiv ℂ (fun y => (WithLp.toLp 2 fun k : Fin n => (∑ j, t (k.succ, j) * w j y) / ∑ j, t (0, j) * w j y : EuclideanSpace ℂ (Fin n))) z).det ≠ 0

theorem **exists_cell** {r : ℕ} (X : Data E r) {x0 : Frame X.Ψ r} (ho : IsOrth X x0) : ∃ O : Set (Frame X.Ψ r), IsOpen O ∧ (∀ b ∈ Bor X.Ψ r, rmul x0 b ∈ O) ∧ ∀ y ∈ O, ∃ z ∈ OmF X x0, ∃ b ∈ Bor X.Ψ r, y = rmul (cpt X x0 z) b ∧ ∀ (L : Wt X.Ψ r) (w0 : Frame X.Ψ r → ℂ) (s : Fin (nd X) → Frame X.Ψ r → ℂ), IsBEq X L w0 → (∀ k, IsBEq X L (s k)) → w0 (cpt X x0 z) ≠ 0 → (fderiv ℂ (fun u : V (nd X) => (WithLp.toLp 2 fun k => s k (cpt X x0 u) / w0 (cpt X x0 u) : V (nd X))) z).det ≠ 0 → (fderiv ℂ (fun u : V (nd X) => (WithLp.toLp 2 fun k => s k (cpt X y u) / w0 (cpt X y u) : V (nd X))) 0).det ≠ 0

theorem **main** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) , ∀ᵐ t : Fin (nd X + 1) × Fin Xa.N → ℂ, let sf : Fin (nd X + 1) → Frame X.Ψ r → ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AeCoefficientsCommonZerosSimpleStatement

theorem **_root_.GlobalTheory.T.ae_coefficients_common_zeros_simple** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) , ∀ᵐ t : Fin (nd X + 1) × Fin Xa.N → ℂ, let sf : Fin (nd X + 1) → Frame X.Ψ r → ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AeExistsFlatDatumLinearCombination

theorem **ae_exists_flat** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (K : Type) [Field K] [Countable K] (σ : K →+* ℂ), RelIn σ X Xa → ∀ᵐ t : Fin (nd X + 1) × Fin Xa.N → ℂ, ∃ Fl : FlatM X Xa, (∀ x, Fl.w0 x = ∑ j, t (0, j) * Xa.w j x) ∧ ∀ k x, Fl.s k x = ∑ j, t (k.succ, j) * Xa.w j x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AeExistsFlatDatumLinearCombinationStatement

theorem **_root_.GlobalTheory.T.ae_exists_flat_datum_linearCombination** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (K : Type) [Field K] [Countable K] (σ : K →+* ℂ), RelIn σ X Xa → ∀ᵐ t : Fin (nd X + 1) × Fin Xa.N → ℂ, ∃ Fl : FlatM X Xa, (∀ x, Fl.w0 x = ∑ j, t (0, j) * Xa.w j x) ∧ ∀ k x, Fl.s k x = ∑ j, t (k.succ, j) * Xa.w j x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AeTranscendentalPointNonvanishing

theorem **ae_eval₂_ne_zero_of_countable** (K : Type) [Field K] [Countable K] (σ : K →+* ℂ) (ι : Type) [Fintype ι] : ∀ᵐ t : ι → ℂ, ∀ Q : MvPolynomial ι K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AeTranscendentalPointNonvanishingStatement

theorem **_root_.GlobalTheory.T.ae_transcendental_point_nonvanishing** : ∀ (K : Type) [Field K] [Countable K] (σ : K →+* ℂ) (ι : Type) [Fintype ι], ∀ᵐ t : ι → ℂ, ∀ Q : MvPolynomial ι K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFactsOfThetaDataOnModels

theorem **GlobalTheory.Shares.exists_anFacts_record** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (N : ℕ) (ka wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (CR c : ℝ), addL wF wL Tn = ka → Nonempty (ArchAt h j w wF Tn (WeightFamilies.c1 D CR Tn c j)) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → MggOnto h j CR Tn c ka → TieMaxCuspSt h j CR Tn c ka → ∃ (R : Rdg (recPlusN h j hj n)) (t : Idx D j N → ℂ) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t ∧ AnFacts R w t ka wF Tn (WeightFamilies.MggK n.base CR Tn c j ka) xs

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFactsOfThetaDataOnModelsStatement

theorem **_root_.GlobalTheory.T.analyticFacts_of_thetaData_on_models** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (N : ℕ) (ka wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (CR c : ℝ), addL wF wL Tn = ka → Nonempty (ArchAt h j w wF Tn (WeightFamilies.c1 D CR Tn c j)) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → MggOnto h j CR Tn c ka → TieMaxCuspSt h j CR Tn c ka → ∃ (R : Rdg (recPlusN h j hj n)) (t : Idx D j N → ℂ) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t ∧ AnFacts R w t ka wF Tn (WeightFamilies.MggK n.base CR Tn c j ka) xs

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameAdjointnessSeriesTrivial

theorem **tay_coord** {g : ℕ} (c : ℂ) (k : Fin g) : tay (fun v : Fin g → ℂ => c * v k) 0 = MvPowerSeries.C c * MvPowerSeries.X ((0 : Fin 1), k)

theorem **psJet_CX** {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (c : ℂ) (s : κ) : psJet m (MvPowerSeries.C c * MvPowerSeries.X s : MvPowerSeries κ ℂ) = Ideal.Quotient.mk _ (MvPolynomial.C c * MvPolynomial.X s)

theorem **jetN_coord** {n g : ℕ} (m : ℕ) (c : ℂ) (i : Fin n) (j : Fin g) : jetN m (fun w : Fin n → Fin g → ℂ => c * w i j) = Ideal.Quotient.mk _ (MvPolynomial.C c * MvPolynomial.X (i, j))

theorem **subst_X_eq** {n n' g : ℕ} (m : ℕ) (sb : Fin n × Fin g → MvPolynomial (Fin n' × Fin g) ℂ) (p : Fin n × Fin g) (hc : MvPolynomial.constantCoeff (sb p) = 0) : (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (psJet m (MvPowerSeries.X p)) = Ideal.Quotient.mk _ (sb p)

theorem **mk_psIn_lin** {n' g : ℕ} (m : ℕ) (i : Fin n') (c : ℂ) (j : Fin g) : Ideal.Quotient.mk (orig ℂ (Fin n' × Fin g) ^ (m + 1)) (psIn g m ![i] (MvPowerSeries.C c * MvPowerSeries.X ((0 : Fin 1), j))) = Ideal.Quotient.mk _ (MvPolynomial.C c * MvPolynomial.X (i, j))

theorem **jetMap_unique** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) {n m : ℕ} {ψ ψ' : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (h : Φ.JetMap m ψ Λ) (h' : Φ.JetMap m ψ' Λ) : ψ = ψ'

theorem **sameSum_cancel** {X Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K ⊕ K → (Y ⟶ X)) (ψ ψ' : K → (Y ⟶ X)) (ε : K → ℤ) (h1 : ∀ k, φ (Sum.inl k) = ψ' k) (h2 : ∀ k, φ (Sum.inr k) = ψ k) : LB.SameSum (Sum.elim φ ψ) (Sum.elim (Sum.elim ε fun k => -ε k) ε) ψ' ε

theorem **scaleGen_one** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {A : Type} [CommRing A] {K : Type} [Fintype K] (φ : K → (Sp A ⟶ Xo.univ.t.G.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb Xo.univ.t.PP.L φ ε)) : scaleGen φ ε (unitΓ 1) t = t

theorem **pairL_apply** {g : ℕ} (A B : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (w : Fin 2 → Fin g → ℂ) : pairL A B w = ![A (w 0), B (w 1)]

theorem **adjoint_series_trivial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → (∀ a : 𝒪 E, star a = H.bar a) → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ a : 𝒪 E, ∃ wL : LB.Gen (LB.LinComb Xo.univ.t.PP.L (Sum.elim (cφ ![fst Xo.univ.t.G Xo.univ.t.G, snd Xo.univ.t.G Xo.univ.t.G ≫ Xo.univ.t.act.act (H.bar a)]) (cφ ![fst Xo.univ.t.G Xo.univ.t.G ≫ Xo.univ.t.act.act a, snd Xo.univ.t.G Xo.univ.t.G])) (Sum.elim (cε 2) fun I => -cε 2 I)), wL.IsGen ∧ ∀ m, ∃ u : (Jet ℂ (Fin 2 × Fin g) m)ˣ, (u : Jet ℂ (Fin 2 × Fin g) m) = psJet m ((anFB ty).ad a) ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L _ _ wL (lift (jpt y g Φ.jx m (0 : Fin

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameAdjointnessSeriesTrivialStatement

theorem **_root_.GlobalTheory.T.analyticFrame_adjointness_series_trivial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → (∀ a : 𝒪 E, star a = H.bar a) → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ a : 𝒪 E, ∃ wL : LB.Gen (LB.LinComb Xo.univ.t.PP.L (Sum.elim (cφ ![fst Xo.univ.t.G Xo.univ.t.G, snd Xo.univ.t.G Xo.univ.t.G ≫ Xo.univ.t.act.act (H.bar a)]) (cφ ![fst Xo.univ.t.G Xo.univ.t.G ≫ Xo.univ.t.act.act a, snd Xo.univ.t.G Xo.univ.t.G])) (Sum.elim (cε 2) fun I => -cε 2 I)), wL.IsGen ∧ ∀ m, ∃ u : (Jet ℂ (Fin 2 × Fin g) m)ˣ, (u : Jet ℂ (Fin 2 × Fin g) m) = psJet m ((anFB ty).ad a) ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameExponentialImmersionAtOrigin

theorem **deriv_one** (h0 : e 0 = (1 : Over.mk y ⟶ G).left) : IsDerivAt0 e 0 (1 : epsPt y ⟶ G).left

theorem **deriv_list** (h0 : e 0 = (1 : Over.mk y ⟶ G).left) : ∀ (m : ℕ) (w : Fin m → Fin g → ℂ) (v : Fin m → (epsPt y ⟶ G)), (∀ i, IsDerivAt0 e (w i) (v i).left) → IsDerivAt0 e (∑ i, w i) (List.ofFn v).prod.left | 0, w, v, _ => by simpa using deriv_one h0 | m + 1, w, v, h => by rw [Fin.sum_univ_succ, List.ofFn_succ, List.prod_cons] exact GlobalTheory.PointUnif.deriv_mul h0 (h 0) (deriv_list h0 m _ _ fun i => h i.succ) end Group theorem anPt_imm : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameExponentialImmersionAtOriginStatement

theorem **_root_.GlobalTheory.T.analyticFrame_exponential_immersion_at_origin** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameFirstPartialLawTrivial

theorem **tay_coord** {N : ℕ} (k : Fin N) : tay (fun v : Fin N → ℂ => v k) 0 = MvPowerSeries.X ((0 : Fin 1), k)

theorem **jetOf_coord** {N : ℕ} (m : ℕ) (k : Fin N) : jetOf m (fun v : Fin N → ℂ => v k) = Ideal.Quotient.mk _ (MvPolynomial.X ((0 : Fin 1), k))

theorem **jetOf_sum_coord** {N : ℕ} (m : ℕ) {T : Type} (I : Finset T) (k : T → Fin N) : jetOf m (fun v : Fin N → ℂ => ∑ i ∈ I, v (k i)) = Ideal.Quotient.mk _ (∑ i ∈ I, MvPolynomial.X ((0 : Fin 1), k i))

theorem **psumL_apply** (n g : ℕ) (I : Finset (Fin n)) (w : Fin n → Fin g → ℂ) : psumL n g ⟨I⟩ w = ∑ i ∈ I, w i

theorem **jetN_psum** {n g : ℕ} (m : ℕ) (I : Finset (Fin n)) (j : Fin g) : jetN m (fun w => psumL n g ⟨I⟩ w j) = Ideal.Quotient.mk _ (∑ i ∈ I, MvPolynomial.X (i, j))

theorem **sameSum_of_factor** {X Y : Scheme.{0}} {K K' J : Type} [Fintype K] [Fintype K'] [Fintype J] [DecidableEq J] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (F : J → (Y ⟶ X)) (σ : K → J) (σ' : K' → J) (h : ∀ k, φ k = F (σ k)) (h' : ∀ k, φ' k = F (σ' k)) (hc : ∀ I, (∑ k, if σ k = I then ε k else 0) = ∑ k, if σ' k = I then ε' k else 0) : LB.SameSum φ ε φ' ε'

theorem **sameSum_of_jetMap** {n : ℕ} (m : ℕ) {K K' J : Type} [Fintype K] [Fintype K'] [Fintype J] [DecidableEq J] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (φ' : K' → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε' : K' → ℤ) (Λ : J → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (σ : K → J) (σ' : K' → J) (h : ∀ k, Φ.JetMap m (φ k) (Λ (σ k))) (h' : ∀ k, Φ.JetMap m (φ' k) (Λ (σ' k))) (hc : ∀ I, (∑ k, if σ k = I then ε k else 0) = ∑ k, if σ' k = I then ε' k else 0) : LB.SameSum φ ε φ' ε'

theorem **lift_jet** {S : Scheme.{0}} {P Q : Over S} {N m : ℕ} {R : Type} [CommRing R] (r : Jet ℂ (Fin 1 × Fin N) m →+* R) (u : Sp R ⟶ S) (x : Sp ℂ ⟶ S) (a : Over.mk u ⟶ P) (b : Over.mk u ⟶ Q) (f : (Fin N → ℂ) → (Over.mk x ⟶ P)) (h : (Fin N → ℂ) → (Over.mk x ⟶ Q)) (hf :

IsHolFam fun v => (f v).left) (hh : IsHolFam fun v => (h v).left) (ea : a.left = spMap r ≫ jetPt _ hf m) (eb : b.left = spMap r ≫ jetPt _ hh m) : ∃ hq : IsHolFam (fun v => (lift (f v) (h v)).left), (lift a b).left = spMap r ≫ jetPt _ hq m

theorem **pe_jetRd** (e : ∀ m, FrTy y g Φ.jx m) (he : Φ.IsAnE e) (m : ℕ) {n : ℕ} (T : Fin 2 → Finset (Fin n)) (hT : Disjoint (T 0) (T 1)) (φ : Fin 2 → Fin g → MvPolynomial (Fin n × Fin g) ℂ) (hφ : ∀ k j, Ideal.Quotient.mk (orig ℂ (Fin n × Fin g) ^ (m + 1)) (φ k j - MvPolynomial.C (MvPolynomial.constantCoeff (φ k j))) = Ideal.Quotient.mk _ (∑ i ∈ T k, MvPolynomial.X (i, j))) : Φ.toAnFr.JetRd m _ _ (pe y g Φ.jx e m φ) (fun K => psumL n g (K.down.biUnion T))

theorem **jet_triple_points_eq_jet_exponentials** (m : ℕ) {n : ℕ} (a b d : jO y g m n ⟶ Xo.univ.t.G) (Λa Λb Λd : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (ha : Φ.toAnFr.JetMap m a.left Λa) (hb : Φ.toAnFr.JetMap m b.left Λb) (hd : Φ.toAnFr.JetMap m d.left Λd) : ∃ hq : IsHolFam (fun v : Fin (n * g) → ℂ => (lift (Φ.e (Λa (colsOf n g v))) (lift (Φ.e (Λb (colsOf n g v))) (Φ.e (Λd (colsOf n g v)))))).left), (lift a (lift b d)).left = spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ jetPt _ hq m

theorem **cub_jetRd** (m : ℕ) : Φ.toAnFr.JetRd m _ _ (LB.pullGen Xo.univ.t.PP.L (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (jpt y g Φ.jx m (0 : Fin 3)) (lift (jpt y g Φ.jx m (1 : Fin 3)) (jpt y g Φ.jx m (2 : Fin 3)))).left) (fun I => psumL 3 g (I.down))

theorem **scaleGen_one** {Y : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ Xo.univ.t.G.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb Xo.univ.t.PP.L φ ε)) : Formalism.L2.Pj.scaleGen φ ε 1 t = t

theorem **hX** {n g : ℕ} (m : ℕ) (i : Fin n) (j : Fin g) : Ideal.Quotient.mk (orig ℂ (Fin n × Fin g) ^ (m + 1)) ((MvPolynomial.X (i, j) : MvPolynomial (Fin n × Fin g) ℂ) - MvPolynomial.C (MvPolynomial.constantCoeff (MvPolynomial.X (i, j) : MvPolynomial (Fin n × Fin g) ℂ))) = Ideal.Quotient.mk _ (∑ i ∈ ({i} : Finset (Fin n)), MvPolynomial.X (i, j))

theorem **hAdd** {n g : ℕ} (m : ℕ) (a b : Fin n) (hab : a ≠ b) (j : Fin g) : Ideal.Quotient.mk (orig ℂ (Fin n × Fin g) ^ (m + 1)) (psIn g m ![a, b] (addF ℂ g j) - MvPolynomial.C (MvPolynomial.constantCoeff (psIn g m ![a, b] (addF ℂ g j)))) = Ideal.Quotient.mk _ (∑ i ∈ ({a, b} : Finset (Fin n)), MvPolynomial.X (i, j))

theorem **first_partial_law** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ m, ∃ u : (Jet ℂ (Fin 3 × Fin g) m)ˣ, (u : Jet ℂ (Fin 3 × Fin g) m) = psJet m (anFB ty).ph1 ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (jpt y g Φ.jx m (0 : Fin 3)) (lift (jpt y g Φ.jx m (1 : Fin 3)) (jpt y g Φ.jx m (2 : Fin 3)))).left) (scaleGen _ _ (unitΓ u) (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (pe y g Φ.jx e m ![fun j => psIn g m ![0, 1] ((anFB ty).G.F j), fun j => MvPolynomial.X ((2 : Fin 3), j)]) (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (LB.invGen Xo.univ.t.PP.L _ _ (pe y g Φ.jx

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameFirstPartialLawTrivialStatement

theorem **_root_.GlobalTheory.T.analyticFrame_first_partial_law_trivial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ m, ∃ u : (Jet ℂ (Fin 3 × Fin g) m)ˣ, (u : Jet ℂ (Fin 3 × Fin g) m) = psJet m (anFB ty).ph1 ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (jpt y g Φ.jx m (0 : Fin 3)) (lift (jpt y g Φ.jx m (1 : Fin 3)) (jpt y g Φ.jx m (2 : Fin 3)))).left) (scaleGen _ _ (unitΓ u) (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (pe y g Φ.jx e m ![fun j => psIn g m ![0, 1] ((anFB ty).G.F j), fun j => MvPolynomial.X ((2 : Fin 3), j)]) (LB.mulGen Xo.univ.t.PP.L _ _ _ _

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameOfAppellHumbertFramePeriodFamily

theorem **form_eq_zero_of_im_span** {g : ℕ} (s : Set (Fin g → ℂ)) (hs : Submodule.span ℝ s = ⊤) (B : (Fin g → ℂ) → (Fin g → ℂ) → ℂ) (h1 : ∀ z z' w, B (z + z') w = B z w + B z' w) (h2 : ∀ (r : ℂ) z w, B (r • z) w = r * B z w) (h3 : ∀ z w w', B z (w + w') = B z w + B z w') (h4 : ∀ (r : ℝ) z w, B z (r • w) = r * B z w) (h0 : ∀ z ∈ s, ∀ w ∈ s, (B z w).im = 0) (z w : Fin g → ℂ) : B z w = 0

theorem **hermForm_adj_of_im_span** {g : ℕ} (s : Set (Fin g → ℂ)) (hs : Submodule.span ℝ s = ⊤) (Hm : Matrix (Fin g) (Fin g) ℂ) (A A' : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h0 : ∀ z ∈ s, ∀ w ∈ s, (hermForm Hm (A z) w).im = (hermForm Hm z (A' w)).im) (z w : Fin g → ℂ) : hermForm Hm (A z) w = hermForm Hm z (A' w)

theorem **anPt_of_family** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (σ' : KA l →+* ℂ) (F : FamilyUnif Xo σ' ty) (i : Fin F.n) (Pm : (F.Di i).La →+ (Fin g → ℂ)) (Hm : Matrix (Fin g) (Fin g) ℂ) (χl : (Fin g → ℂ) → ℂ) (fr : Xo.univ.t.PP.L.ι → (Fin g → ℂ) → ℂ), IsPeriod (F.Di i) ty Pm → y.x = F.u i Pm → AHFrame F i Pm Hm χl fr → (∀ ℓ : Lt1 Dw, IsDerivAt0 (fun z => (F.ex i Pm z).left) (Pi.single (el.symm ℓ) 1) (y.u ℓ.1 ℓ.2)) → ∃ Φ : AnPt y g ty el, Φ.OfFam F i Pm ∧ Φ.Hm = Hm ∧ Φ.χ = χl ∧ Φ.fr = fr ∧ (∀ (α : 𝒪 E) a, Pm (F.t i (α • a)) = tyAct ty α (Pm (F.t i a))) ∧ ∀ a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La), levPt Xo y.x a = Φ.e ((l.M : ℂ)⁻¹ • Pm (F.t i a))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameOfAppellHumbertFramePeriodFamilyStatement

theorem **_root_.GlobalTheory.T.analyticFrame_of_appellHumbertFramePeriodFamily** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (σ' : KA l →+* ℂ) (F : FamilyUnif Xo σ' ty) (i : Fin F.n) (Pm : (F.Di i).La →+ (Fin g → ℂ)) (Hm : Matrix (Fin g) (Fin g) ℂ) (χl : (Fin g → ℂ) → ℂ) (fr : Xo.univ.t.PP.L.ι → (Fin g → ℂ) → ℂ), IsPeriod (F.Di i) ty Pm → y.x = F.u i Pm → AHFrame F i Pm Hm χl fr → (∀ ℓ : Lt1 Dw, IsDerivAt0 (fun z => (F.ex i Pm z).left) (Pi.single (el.symm ℓ) 1) (y.u ℓ.1 ℓ.2)) → ∃ Φ : AnPt y g ty el, Φ.OfFam F i Pm ∧ Φ.Hm = Hm ∧ Φ.χ = χl ∧ Φ.fr = fr ∧ (∀ (α : 𝒪 E) a, Pm (F.t i (α • a)) = tyAct ty α (Pm (F.t i a))) ∧ ∀ a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La), levPt Xo y.x a = Φ.e ((l.M : ℂ)⁻¹ • Pm (F.t i a))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameSecondPartialLawTrivial

theorem **tay_coord** {N : ℕ} (k : Fin N) : tay (fun v : Fin N → ℂ => v k) 0 = MvPowerSeries.X ((0 : Fin 1), k)

theorem **psJet_X'** {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (p : κ) : psJet m (MvPowerSeries.X p : MvPowerSeries κ ℂ) = Ideal.Quotient.mk _ (MvPolynomial.X p)

theorem **jetN_coord** {n g : ℕ} (m : ℕ) (i : Fin n) (j : Fin g) : jetN m (fun w : Fin n → Fin g → ℂ => w i j) = Ideal.Quotient.mk _ (MvPolynomial.X (i, j))

theorem **jetN_coord_add** {n g : ℕ} (m : ℕ) (i i' : Fin n) (j : Fin g) : jetN m (fun w : Fin n → Fin g → ℂ => w i j + w i' j) = Ideal.Quotient.mk _ (MvPolynomial.X (i, j) + MvPolynomial.X (i', j))

theorem **subst_X** {κ κ' : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (sb : κ → MvPolynomial κ' ℂ) (p : κ) : (jetSubstP m sb : Jet ℂ κ m →ₐ[ℂ] Jet ℂ κ' m) (psJet m (MvPowerSeries.X p)) = Ideal.Quotient.mk _ (sb p - MvPolynomial.C (MvPolynomial.constantCoeff (sb p)))

theorem **psIn_add** {g n' : ℕ} (m : ℕ) (r : Fin 2 → Fin n') (j : Fin g) : (Ideal.Quotient.mk _ (psIn g m r (addF ℂ g j) - MvPolynomial.C (MvPolynomial.constantCoeff (psIn g m r (addF ℂ g j)))) : Jet ℂ (Fin n' × Fin g) m) = Ideal.Quotient.mk _ (MvPolynomial.X (r 0, j) + MvPolynomial.X (r 1, j))

theorem **sameSum_of_jetMap** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) {n m : ℕ} {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)} {ε : K → ℤ} {Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} {φ' : K' → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)} {ε' : K' → ℤ} {Λ' : K' → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (h : ∀ k, Φ.JetMap m (φ k) (Λ k)) (h' : ∀ k, Φ.JetMap m (φ' k) (Λ' k)) (hs : ∀ f : ((Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) → ℤ, ∑ k, ε k * f (Λ k) = ∑ k, ε' k * f (Λ' k)) : LB.SameSum φ ε φ' ε'

theorem **sum_two** {M : Type} [AddCommMonoid M] (F : ULift.{0} (Finset (Fin 2)) → M) : ∑ I, F I = F (∅) + F ({0}) + F ({1}) + F ({0, 1})

theorem **sum_three** {M : Type} [AddCommMonoid M] (F : thK.{0} → M) : ∑ I, F I = F (∅) + F ({0}) + F ({1}) + F ({2}) + F ({0, 1}) + F ({0, 2}) + F ({1, 2}) + F ({0, 1, 2})

theorem **psumL_pi** {n n' g : ℕ} (I : ULift.{0} (Finset (Fin n))) (B : Fin n → (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) : psumL n g I ∘ₗ LinearMap.pi B = ∑ i ∈ I.down, B i

theorem **count** {g : ℕ} (P Q R : (Fin 3 → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (f : ((Fin 3 → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) → ℤ) : ∑ I : thK.{0}, thε I * f (psumL 3 g I ∘ₗ LinearMap.pi ![Q, R, P]) = ∑ k : ULift.{0} (Finset (Fin 2)) ⊕ ULift.{0} (Finset (Fin 2)) ⊕ ULift.{0} (Finset (Fin 2)), Sum.elim (cε 2) (Sum.elim (fun k => - cε 2 k) (fun k => - cε 2 k)) k * f (Sum.elim (fun k => psumL 2 g k ∘ₗ LinearMap.pi ![P, Q + R]) (Sum.elim (fun k => psumL 2 g k ∘ₗ LinearMap.pi ![P, Q]) (fun k => psumL 2 g k ∘ₗ LinearMap.pi ![P, R])) k)

theorem **jetRd_scale_one** {n m : ℕ} {K : Type} [Fintype K] {φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ Xo.univ.t.G.left)} {ε : K → ℤ} (τ : LB.Gen (LB.LinComb Xo.univ.t.PP.L φ ε)) : scaleGen φ ε (unitΓ (1 : (Jet ℂ (Fin n × Fin g) m)ˣ)) τ = τ

theorem **jetRd_pe** (e : ∀ m, FrTy y g Φ.jx m) (he : Φ.IsAnE e) (m : ℕ) {n' : ℕ} (φ : Fin 2 → Fin g → MvPolynomial (Fin n' × Fin g) ℂ) (B : Fin 2 → (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : ∀ i j, (Ideal.Quotient.mk _ (φ i j - MvPolynomial.C (MvPolynomial.constantCoeff (φ i j))) : Jet ℂ (Fin n' × Fin g) m) = jetN m (fun w => B i w j)) : Φ.toAnFr.JetRd m _ _ (pe y g Φ.jx e m φ) (fun k => psumL 2 g k ∘ₗ LinearMap.pi B)

theorem **jetRd_pe_two** (e : ∀ m, FrTy y g Φ.jx m) (he : Φ.IsAnE e) (m : ℕ) {n' : ℕ} (φ φ' : Fin g → MvPolynomial (Fin n' × Fin g) ℂ) (B B' : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : ∀ j, (Ideal.Quotient.mk _ (φ j - MvPolynomial.C (MvPolynomial.constantCoeff (φ j))) : Jet ℂ (Fin n' × Fin g) m) = jetN m (fun w => B w j)) (h' : ∀ j, (Ideal.Quotient.mk _ (φ' j - MvPolynomial.C (MvPolynomial.constantCoeff (φ' j))) : Jet ℂ (Fin n' × Fin g) m) = jetN m (fun w => B' w j)) : Φ.toAnFr.JetRd m _ _ (pe y g Φ.jx e m ![φ, φ']) (fun k => psumL 2 g k ∘ₗ LinearMap.pi ![B, B'])

theorem **jetRd_cube** {n : ℕ} (m : ℕ) (a b d : jO y g m n ⟶ Xo.univ.t.G) (P Q R : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (ha : Φ.toAnFr.JetMap m a.left P) (hb : Φ.toAnFr.JetMap m b.left Q) (hd : Φ.toAnFr.JetMap m d.left R) : Φ.toAnFr.JetRd m _ _ (LB.pullGen Xo.univ.t.PP.L (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift a (lift b d)).left) (fun I => psumL 3 g I ∘ₗ LinearMap.pi ![P, Q, R])

theorem **second_law** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ m, ∃ u : (Jet ℂ (Fin 3 × Fin g) m)ˣ, (u : Jet ℂ (Fin 3 × Fin g) m) = psJet m (anFB ty).ph2 ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (jpt y g Φ.jx m (1 : Fin 3)) (lift (jpt y g Φ.jx m (2 : Fin 3)) (jpt y g Φ.jx m (0 : Fin 3))))).left) (scaleGen _ _ (unitΓ u) (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (pe y g Φ.jx e m ![fun j => MvPolynomial.X ((0 : Fin 3), j), fun j => psIn g m ![1, 2] ((anFB ty).G.F j)]) (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (LB.invGen Xo.univ.t.PP.L _ _ (pe y g Φ.jx e m ![f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameSecondPartialLawTrivialStatement

theorem **_root_.GlobalTheory.T.analyticFrame_second_partial_law_trivial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ m, ∃ u : (Jet ℂ (Fin 3 × Fin g) m)ˣ, (u : Jet ℂ (Fin 3 × Fin g) m) = psJet m (anFB ty).ph2 ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (jpt y g Φ.jx m (1 : Fin 3)) (lift (jpt y g Φ.jx m (2 : Fin 3)) (jpt y g Φ.jx m (0 : Fin 3))))).left) (scaleGen _ _ (unitΓ u) (LB.mulGen Xo.univ.t.PP.L _ _ _ _ (pe y g Φ.jx e m ![fun j => MvPolynomial.X ((0 : Fin 3), j), fun j => psIn g m ![1, 2] ((anFB ty).G.F j)]) (LB.mulGen Xo.univ.t.PP.L _ _ _

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameSymmetryTrivial

theorem **tay_coord** {g : ℕ} (k : Fin g) : tay (fun v : Fin g → ℂ => v k) 0 = MvPowerSeries.X ((0 : Fin 1), k)

theorem **psJet_X** {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (s : κ) : psJet m (MvPowerSeries.X s : MvPowerSeries κ ℂ) = Ideal.Quotient.mk _ (MvPolynomial.X s)

theorem **jetN_coord** {n g : ℕ} (m : ℕ) (i : Fin n) (j : Fin g) : jetN m (fun w : Fin n → Fin g → ℂ => w i j) = Ideal.Quotient.mk _ (MvPolynomial.X (i, j))

theorem **psumL_swap** (g : ℕ) (I : ULift.{0} (Finset (Fin 2))) : psumL 2 g I ∘ₗ swapL g = psumL 2 g (swapI I)

theorem **cε_swap** (I : ULift.{0} (Finset (Fin 2))) : cε 2 (swapI I) = cε 2 I

theorem **jetMap_unique** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) {n m : ℕ} {ψ ψ' : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (h : Φ.JetMap m ψ Λ) (h' : Φ.JetMap m ψ' Λ) : ψ = ψ'

theorem **sy_trivial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ m, ∃ u : (Jet ℂ (Fin 2 × Fin g) m)ˣ, (u : Jet ℂ (Fin 2 × Fin g) m) = psJet m (anFB ty).sy ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e m) (scaleGen _ _ (unitΓ u⁻¹) (pe y g Φ.jx e m ![fun j => MvPolynomial.X ((1 : Fin 2), j), fun j => MvPolynomial.X ((0 : Fin 2), j)]))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticFrameSymmetryTrivialStatement

theorem **_root_.GlobalTheory.T.analyticFrame_symmetry_trivial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e → ∀ m, ∃ u : (Jet ℂ (Fin 2 × Fin g) m)ˣ, (u : Jet ℂ (Fin 2 × Fin g) m) = psJet m (anFB ty).sy ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e m) (scaleGen _ _ (unitΓ u⁻¹) (pe y g Φ.jx e m ![fun j => MvPolynomial.X ((1 : Fin 2), j), fun j => MvPolynomial.X ((0 : Fin 2), j)]))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticLineBundleFrameEqAppellHumbert

theorem **tay_coord** {N : ℕ} (k : Fin N) : tay (fun v : Fin N → ℂ => v k) 0 = MvPowerSeries.X ((0 : Fin 1), k)

theorem **jetN_coord** {n g : ℕ} (m : ℕ) (a : ℂ) (i : Fin n) (j : Fin g) : jetN m (fun w : Fin n → Fin g → ℂ => a * w i j) = Ideal.Quotient.mk _ (MvPolynomial.C a * MvPolynomial.X (i, j))

theorem **jetSubstP_X** {κ κ' : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (sb : κ → MvPolynomial κ' ℂ) (p : κ) : (jetSubstP m sb : Jet ℂ κ m →ₐ[ℂ] Jet ℂ κ' m) (psJet m (MvPowerSeries.X p)) = Ideal.Quotient.mk _ (sb p − MvPolynomial.C (MvPolynomial.constantCoeff (sb p)))

theorem **jetAug_jetN** {n g : ℕ} (m : ℕ) (f : (Fin n → Fin g → ℂ) → ℂ) : (jetAug : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] ℂ) (jetN m f) = f 0

theorem **valAt'_id_inj** {W : (Sp ℂ).Opens} (hW : ⊤ ≤ W) {a b : Γ(Sp ℂ, W)} (h : valAt' (𝟙 (Sp ℂ)) W a = valAt' (𝟙 (Sp ℂ)) W b) : a = b

theorem **valAt'_spMap_toΓ** {R : Type} [CommRing R] (f : R →+* ℂ) (a : R) : valAt' (spMap f) ⊤ (toΓ R a) = f a

theorem **sameSum_cancel** {X Y : Scheme.{0}} {P : Type} [Fintype P] (x x' : Y ⟶ X) (hx : x = x') (k : ℤ) (φ : Fin 1 ⊕ P → (Y ⟶ X)) (φ' : P → (Y ⟶ X)) (ε : P → ℤ) (h1 : ∀ i, φ (Sum.inl i) = x') (h2 : ∀ I, φ (Sum.inr I) = φ' I) : LB.SameSum (fun _ : Fin 1 => x) (fun _ => k) (Sum.elim φ φ') (fun i => k * Sum.elim (Sum.elim (fun _ => 1) fun I => -ε I) ε i)

theorem **prod_zpow_sum** {K : Type} {a : ℂ} (ha : a ≠ 0) (s : Finset K) (ε : K → ℤ) : ∏ k ∈ s, a ^ ε k = a ^ ∑ k ∈ s, ε k

theorem **jetMap_uniq** {n m : ℕ} {ψ ψ' : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ Λ' : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (h : Φ.JetMap m ψ Λ) (h' : Φ.JetMap m ψ' Λ') (e : Λ = Λ') : ψ = ψ'

theorem **e_zero_left** : (Φ.e 0).left = x0 ≫ eSec A

theorem **jetMap_aug** {n m : ℕ} {ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (h : Φ.JetMap m ψ Λ) : spMap ((jetAug : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] ℂ) : Jet ℂ (Fin n × Fin g) m →+* ℂ) ≫ ψ = (Φ.e 0).left

theorem **corr_rig** {n m : ℕ} {K : Type} [Fintype K] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : Φ.JetRd m φ ε τ Λ) (hε : ∑ k, ε k = 1) : LB.Gen.Corr L.L _ _ _ _ (LB.pullGen L.L φ ε τ (homAt x0 (jetAug : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] ℂ)).left) (rigT A L.L L.rig (overAt x0 ℂ) 1)

theorem **graphL_apply** (a : 𝒪 E) (w : Fin 1 → Fin g → ℂ) : graphL ty a w = ![w 0, tyAct ty a (w 0)]

theorem **jetRd_prim** (wP : PrimTy H Xo) (hP : IsPrim wP) (m : ℕ) : Φ.toAnFr.JetRd (n

theorem **jetRd_pe** (e : ∀ m, FrTy y g Φ.jx m) (he : Φ.IsAnE e) (m : ℕ) : Φ.toAnFr.JetRd (n

theorem **eL_appellHumbert** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ (e : ∀ m, FrTy y g Φ.jx m) (wP : PrimTy H Xo), Φ.IsAnE e → IsPrim wP → (∀ (m k : ℕ) (τ : LB.Gen (LB.LinComb Xo.univ.t.PP.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AnalyticLineBundleFrameEqAppellHumbertStatement

theorem **_root_.GlobalTheory.T.analytic_line_bundle_frame_eq_appellHumbert** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), Φ.toAnFr.Imm → ∀ (e : ∀ m, FrTy y g Φ.jx m) (wP : PrimTy H Xo), Φ.IsAnE e → IsPrim wP → (∀ (m k : ℕ) (τ : LB.Gen (LB.LinComb Xo.univ.t.PP.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AppLESpMapJetPt

theorem **appLE_spMap_jetPt** {N m : ℕ} {I : Type} {Z : Scheme.{0}} (q : (Fin N → ℂ) → (Sp ℂ ⟶ Z)) (hq : IsHolFam q) (χ : Jet ℂ (Fin 1 × Fin N) m →+* Jet ℂ I m) [IsLocalHom χ] (U : Z.Opens) (h : ⊤ ≤ (spMap χ ≫ jetPt q hq m) ⁻¹ᵁ U) : q 0 cpt ∈ U ∧ ∀ f : Γ(Z, U), (spMap χ ≫ jetPt q hq m).appLE U ⊤ h f = toΓ _ (χ (jetOf m fun v => valAt' (q v) U f))

theorem **jetRd_pullGen** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (n m : ℕ) (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (q : (Fin (n * g) → ℂ) → (Sp ℂ ⟶ Z)) (hq : IsHolFam q), (∀ v, Φ.GenRd φ ε t (q v) (fun k => Λ k (colsOf n g v)) 1) → Φ.JetRd m (fun k => (spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ jetPt q hq m) ≫ φ k) ε (LB.pullGen L.L φ ε t (spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ jetPt q hq m)) Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AppellHumbertFrameSemicharacter

theorem **semichar_herm_zero_left** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (w : Fin g → ℂ) : hermForm Hm 0 w = 0

theorem **semichar_herm_add_left** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z z' w : Fin g → ℂ) : hermForm Hm (z + z') w = hermForm Hm z w + hermForm Hm z' w

theorem **semichar_herm_add_right** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z w w' : Fin g → ℂ) : hermForm Hm z (w + w') = hermForm Hm z w + hermForm Hm z w'

theorem **semichar_herm_swap** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (z w : Fin g → ℂ) : hermForm Hm w z = star (hermForm Hm z w)

theorem **AnFr.exists_chart_lat** : ∃ j, ∀ a ∈ Φ.lat, ⊤ ≤ (Φ.e a).left ⁻¹ᵁ L.L.U j

theorem **AnFr.χ_add** {a b : Fin g → ℂ} (ha : a ∈ Φ.lat) (hb : b ∈ Φ.lat) : Φ.χ (a + b) = Φ.χ a * Φ.χ b * Complex.exp (Real.pi * Complex.I * ((hermForm Φ.Hm a b).im : ℂ))

theorem **AnFr.χ_ne_zero** {a : Fin g → ℂ} (ha : a ∈ Φ.lat) : Φ.χ a ≠ 0

theorem **AnFr.herm_im_swap** (a b : Fin g → ℂ) : (hermForm Φ.Hm b a).im = -(hermForm Φ.Hm a b).im

theorem **AnFr.herm_im_int** {a b : Fin g → ℂ} (ha : a ∈ Φ.lat) (hb : b ∈ Φ.lat) : ∃ k : ℤ, (hermForm Φ.Hm a b).im = k

theorem **AnFr.herm_im_self** (a : Fin g → ℂ) : (hermForm Φ.Hm a a).im = 0

theorem **AnFr.χ_add_self** {a : Fin g → ℂ} (ha : a ∈ Φ.lat) : Φ.χ (a + a) = Φ.χ a ^ 2

theorem **AnFr.χ_zero** : Φ.χ 0 = 1

theorem **anFr_semichar** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), (∀ a ∈ Φ.lat, ∀ b ∈ Φ.lat, (∃ k : ℤ, (hermForm Φ.Hm a b).im = k) ∧ Φ.χ (a + b) = Φ.χ a * Φ.χ b * Complex.exp (Real.pi * Complex.I * ((hermForm Φ.Hm a b).im : ℂ))) ∧ Φ.χ 0 = 1 ∧ ((∀ a ∈ Φ.lat, Φ.χ a ^ 2 = 1) → ∀ a ∈ Φ.lat, Φ.χ ((2 : ℂ) • a) = 1)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.AppellHumbertFrameSemicharacterStatement

theorem **_root_.GlobalTheory.T.appellHumbert_frame_semicharacter** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), (∀ a ∈ Φ.lat, ∀ b ∈ Φ.lat, (∃ k : ℤ, (hermForm Φ.Hm a b).im = k) ∧ Φ.χ (a + b) = Φ.χ a * Φ.χ b * Complex.exp (Real.pi * Complex.I * ((hermForm Φ.Hm a b).im : ℂ))) ∧ Φ.χ 0 = 1 ∧ ((∀ a ∈ Φ.lat, Φ.χ a ^ 2 = 1) → ∀ a ∈ Φ.lat, Φ.χ ((2 : ℂ) • a) = 1)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ArchimedeanFlatDecompositionIndexing

theorem **finrank_le_of_parts** {Y ι : Type} {Z : ι → Type} (V : Submodule ℂ (Y → ℂ)) (e : Y ≃ Σ k, Z k) (d : ι → ℕ) (C δ : ℕ) (en : Fin δ ≃ Σ k, Fin (d k)) (hb : ∀ (k : ι) (m : ℕ) (g : Fin m → V), LinearIndependent ℂ (fun q (z : Z k) => (g q).1 (e.symm ⟨k, z⟩))) → m ≤ d k * C) : Module.finrank ℂ V ≤ δ * C

theorem **upper** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → Module.finrank ℂ (R.sp (addL wF wL Tn) Set.univ) ≤ δ * (Tn + 2 * WeightFamilies.dimY D j + 6).choose (WeightFamilies.dimY D j)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ArchimedeanFlatDecompositionIndexingStatement

theorem **_root_.GlobalTheory.T.archimedean_flat_decomposition_indexing** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → Module.finrank ℂ (R.sp (addL wF wL Tn) Set.univ) ≤ δ * (Tn + 2 * WeightFamilies.dimY D j + 6).choose (WeightFamilies.dimY D j)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.BetaEqZeroOfCharacter

@[simp] theorem **beta_apply** (ω : LinearMap.BilinForm R M) (σ σ' : V →ₗ[R] M) (z w : V) : beta ω σ σ' z w = ω (σ z) (σ' w)

theorem **omegaCx_apply** (a b : Fin g ⊕ Fin g → ℂ) : omegaCx g a b = ∑ i, (a (.inl i) * b (.inr i) − a (.inr i) * b (.inl i))

theorem **omegaCx_eq_dot** (a b : Fin g ⊕ Fin g → ℂ) : omegaCx g a b = (a ∘ Sum.inl) ⬝ᵥ (b ∘ Sum.inr) − (a ∘ Sum.inr) ⬝ᵥ (b ∘ Sum.inl)

theorem **omegaCx_elim** (m n m' n' : Fin g → ℂ) : omegaCx g (Sum.elim m n) (Sum.elim m' n') = m ⬝ᵥ n' − n ⬝ᵥ m'

theorem **omegaCx_intCast** (a b : Fin g ⊕ Fin g → ℤ) : omegaCx g (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) = (omegaStd a b : ℂ)

theorem **eq_omegaCx_of_intCast** (B : LinearMap.BilinForm ℂ (Fin g ⊕ Fin g → ℂ)) (hB : ∀ a b : Fin g ⊕ Fin g → ℤ, B (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) = (omegaStd a b : ℂ)) : B = omegaCx g

theorem **perProj_apply** (τ : Matrix (Fin g) (Fin g) ℂ) (u : Fin g ⊕ Fin g → ℂ) : perProj τ u = u ∘ Sum.inl + τ *ᵥ (u ∘ Sum.inr)

theorem **perProj_elim** (τ : Matrix (Fin g) (Fin g) ℂ) (m n : Fin g → ℂ) : perProj τ (Sum.elim m n) = m + τ *ᵥ n

theorem **secL_apply** (z : Fin g → ℂ) : secL g z = Sum.elim z 0

theorem **secH_apply** (τ : Matrix (Fin g) (Fin g) ℂ) (z : Fin g → ℂ) : secH τ z = Sum.elim (−(τ.map star *ᵥ (((2 * I) • imC τ)⁻¹ *ᵥ z))) (((2 * I) • imC τ)⁻¹ *ᵥ z)

theorem **dot_mulVec_symm** {A : Matrix (Fin g) (Fin g) ℂ} (hA : A.IsSymm) (x y : Fin g → ℂ) : (A *ᵥ x) ⬝ᵥ y = x ⬝ᵥ (A *ᵥ y)

theorem **perProj_eq_zero_iff** (τ : Matrix (Fin g) (Fin g) ℂ) (u : Fin g ⊕ Fin g → ℂ) : perProj τ u = 0 ↔ u ∘ Sum.inl = −(τ *ᵥ (u ∘ Sum.inr))

theorem **secL_isLagSection** (τ : Matrix (Fin g) (Fin g) ℂ) : IsLagSection (omegaCx g) (perProj τ) (secL g)

theorem **sub_map_star** (τ : Matrix (Fin g) (Fin g) ℂ) : τ − τ.map star = (2 * I) • imC τ

theorem **imC_mul_inv** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) : imC τ * ((τ.map Complex.im)⁻¹).map (fun x : ℝ => (x : ℂ)) = 1

theorem **imC_inv** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) : (imC τ)⁻¹ = ((τ.map Complex.im)⁻¹).map (fun x : ℝ => (x : ℂ))

theorem **two_I_imC_mul** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) : ((2 * I) • imC τ) * ((2 * I)⁻¹ • (imC τ)⁻¹) = 1

theorem **two_I_imC_inv** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) : ((2 * I) • imC τ)⁻¹ = (2 * I)⁻¹ • (imC τ)⁻¹

theorem **two_I_imC_mul_inv** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) : ((2 * I) • imC τ) * ((2 * I) • imC τ)⁻¹ = 1

theorem **two_I_imC_inv_mul** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) : ((2 * I) • imC τ)⁻¹ * ((2 * I) • imC τ) = 1

theorem **perProj_secH** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) (z : Fin g → ℂ) : perProj τ (secH τ z) = z

theorem **star_mulVec_map** (τ : Matrix (Fin g) (Fin g) ℂ) (n : Fin g → ℂ) : star (τ *ᵥ n) = τ.map star *ᵥ star n

theorem **mem_range_secH_iff** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) (u : Fin g ⊕ Fin g → ℂ) : u ∈ LinearMap.range (secH τ) ↔ perProj τ (star u) = 0

theorem **eq_secH_of_conj_mem_ker** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) {u : Fin g ⊕ Fin g → ℂ} {z : Fin g → ℂ} (hu : perProj τ u = z) (hc : perProj τ (star u) = 0) : u = secH τ z

theorem **secH_equivariant** {τ : Matrix (Fin g) (Fin g) ℂ} (hY : IsUnit (τ.map Complex.im).det) (A : (Fin g ⊕ Fin g → ℂ) → (Fin g ⊕ Fin g → ℂ)) (B : (Fin g → ℂ) → (Fin g → ℂ)) (hB : B 0 = 0) (hA : ∀ u, A (star u) = star (A u)) (hAB : ∀ u, perProj τ (A u) = B (perProj τ u)) (z : Fin g → ℂ) : secH τ (B z) = A (secH τ z)

theorem **beta_secL** (σ : (Fin g → ℂ) →ₗ[ℂ] (Fin g ⊕ Fin g → ℂ)) (z w : Fin g → ℂ) : beta (omegaCx g) σ (secL g) z w = −((σ z ∘ Sum.inr) ⬝ᵥ w)

theorem **beta_secH_secL'** {τ : Matrix (Fin g) (Fin g) ℂ} (hτ : τ.IsSymm) (hY : IsUnit (τ.map Complex.im).det) (z w : Fin g → ℂ) : beta (omegaCx g) (secH τ) (secL g) z w = −(2 * I)⁻¹ * (z ⬝ᵥ (imC τ)⁻¹ *ᵥ w)

theorem **beta_secH_secL** {τ : Matrix (Fin g) (Fin g) ℂ} (hτ : τ.IsSymm) (hY : IsUnit (τ.map Complex.im).det) (z w : Fin g → ℂ) : beta (omegaCx g) (secH τ) (secL g) z w = −(2 * I)⁻¹ * (z ⬝ᵥ ((τ.map Complex.im)⁻¹).map (fun x : ℝ => (x : ℂ)) *ᵥ w)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.BetaExistsOfNeOne

theorem **betaExists_of_ne_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n bad : ℕ), bad ≠ 1 → BetaExists (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardCompLePeriodSpaces

theorem **dom_nonempty** {r : ℕ} (X : Data E r) : ∃ x, x ∈ X.dom

theorem **im_eq_of_span** {g : ℕ} {Λ : Type} (P : Λ → Fin g → ℂ) (hspan : Submodule.span ℝ (Set.range P) = ⊤) (A B : Matrix (Fin g) (Fin g) ℂ) (h : ∀ x y, (hermForm A (P x) (P y)).im = (hermForm B (P x) (P y)).im) (z z' : Fin g → ℂ) : (hermForm A z z').im = (hermForm B z z').im

theorem **not_im_neg** (H : Host E) (D : Datum H) {r : ℕ} (hr : 1 ≤ r) (ξ : E) (Gm : Matrix (Fin r) (Fin r) E) (φ : D.La → (Fin r → E)) (Ψ : Finset (E →+* ℂ)) (hcm : ∀ τ : E →+* ℂ, τ ∈ Ψ ↔ ComplexEmbedding.conjugate τ ∉ Ψ) {g : ℕ} (idx : Fin g ≃ Ψ × Fin r) (ty : Fin g → (E →+* ℂ)) (q : ℕ) (hξ : cc ξ = -ξ) (hG : ∀ a b, cc (Gm a b) = Gm b a) (hEF : ∀ x y, ((D.EF x y : ℤ) : ℚ) = Algebra.trace ℚ E (ξ * herm Gm (φ x) (φ y))) (x : Frame Ψ r) (hx : x ∈ GG Ψ Gm q) (P : D.La →+ (Fin g → ℂ)) (hP : ∀ v k, P v k = GlobalTheory.FramePeriod.locPer (idx k).1.1 Gm q (mat x (idx k).1) (Data.emb (idx k).1.1 (φ v)) (idx k).2) (hper : IsPeriod D ty P) (hneg : ∀ τ ∈ Ψ, (τ ξ).im < 0) : False

theorem **exists_idx** {c : Coeff l} {σ : c.F →+* ℂ} (hσ : GoodEmb c σ) {r g : ℕ} (hr : r = Dw.p + Dw.q) (e : Fin g ≃ Σ τ : E →+* ℂ, Fin (Dw.mult τ)) (X : Data E r) (hp0 : X.p0 = Dw.q) (hty : ∀ τ : E →+* ℂ, τ ∈ X.Ψ ↔ τ (H.rt : E) = H.rt0) : ∃ idx : Fin g ≃ X.Ψ × Fin r, (∀ j, relab c σ (e j).1 = if ((idx j).2 : ℕ) < Dw.q then ComplexEmbedding.conjugate (idx j).1.1 else (idx j).1.1) ∧ ∀ (x : Frame X.Ψ r) (v : Fin r → E) j, perFr X x v (relab c σ (e j).1) (e j).2 = if ((idx j).2 : ℕ) < Dw.q then hermC (idx j).1.1 X.Gm (col (mat x (idx j).1) (idx j).2) (Data.emb (idx j).1.1 v) else ((mat x (idx j).1)⁻¹ *ᵥ Data.emb (idx j).1.1 v) (idx j).2

theorem **exists_frame** {c : Coeff l} {σ : c.F →+* ℂ} (hσ : GoodEmb c σ) {r g : ℕ} (h1 : 1 ≤ r) (hr : r = Dw.p + Dw.q) (e : Fin g ≃ Σ τ : E →+* ℂ, Fin (Dw.mult τ)) (X : Data E r) (hp0 : X.p0 = Dw.q) (hty : ∀ τ : E →+* ℂ, τ ∈ X.Ψ ↔ τ (H.rt : E) = H.rt0) (D : Datum H) (hp : D.p = Dw.p) (hq : D.q = Dw.q) (φ : D.La ≃ₗ[𝒪 E] X.Λ) (hEF : ∀ a b, ((D.EF a b : ℤ) : ℚ) = Algebra.trace ℚ E (-X.ξ0 * herm X.Gm (φ a : Fin r → E) (φ b))) (x₀ : Frame X.Ψ r) (hx₀ : x₀ ∈ X.dom) (P₀ : D.La →+ (Fin g → ℂ)) (hP₀ : ∀ a j, P₀ a j = perFr X x₀ (φ a : Fin r → E) (relab c σ (e j).1) (e j).2) (hper₀ : IsPeriod D (fun j => relab c σ (e j).1) P₀) (P : D.La →+ (Fin g → ℂ)) (hper : IsPeriod D (fun j => relab c σ (e j).1) P) : ∃ x ∈ X.dom, ∀ a j, P a j = perFr X x (φ a : Fin r → E) (relab c σ (e j).1) (e j).2

theorem **frBor_of_eq** {Xo : Moduli Dw l} {c : Coeff l} {S : Type} [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (hx : y'.x = y.x) (hu : ∀ τ i, y'.u τ i = y.u τ i) : ∃ t : Ix Dw → Sˣ, FrBor t y y'

theorem **card_comp_le_periodSpaces** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), 1 ≤ r → Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)) (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) ty), IsTypeFn H D.p D.q ty ∧ Fintype.card U.Comp ≤ F.n

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardCompLePeriodSpacesStatement

theorem **_root_.GlobalTheory.T.exists_familyUnif_isTypeFn_card_comp_le** : Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), 1 ≤ r → Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → ∃ (g : ℕ) (ty : Fin g → (E →+* ℂ)) (F : FamilyUnif Xo (σ.comp (IntermediateField.inclusion c.le).toRingHom) ty), IsTypeFn H D.p D.q ty ∧ Fintype.card U.Comp ≤ F.n

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardExponentsAndChooseRatioBoundsPair

theorem **card_exponents_sum_lt_sub** (D T : ℕ) (hT : D < T) : Nat.card {α : Fin D → ℕ // ∑ i, α i < T - D} = (T - 1).choose D

theorem **le_of_add_mul_choose_pred_le** (D T δ d ρ : ℕ) (hD : 2 ≤ D) (hT : 11 * D ^ 2 ≤ T) (hd : d ≤ δ * (T + 2 * D + 6).choose D) (hρ : ρ + δ * (T - 1).choose D ≤ d) : (ρ : ℝ) ≤ 22 * (D : ℝ) ^ 2 * d / T

theorem **card_exponents_and_choose_ratio_bounds** : ∀ (D T : ℕ), (0 < D → D < T → Nat.card {α : Fin D → ℕ // ∑ i, α i < T - D} = (T - 1).choose D) ∧ ∀ (δ d ρ : ℕ), 2 ≤ D → 11 * D ^ 2 ≤ T → d ≤ δ * (T + 2 * D + 6).choose D → (ρ + δ * (T - 1).choose D ≤ d → (ρ : ℝ) ≤ 22 * (D : ℝ) ^ 2 * d / T) ∧ (ρ + δ * (T + D).choose D ≤ d → (ρ : ℝ) ≤ 22 * (D : ℝ) ^ 2 * d / T)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardExponentsAndChooseRatioBoundsPairStatement

theorem **_root_.GlobalTheory.T.card_exponents_and_choose_ratio_bounds_pair** : ∀ (D T : ℕ), (0 < D → D < T → Nat.card {α : Fin D → ℕ // ∑ i, α i < T - D} = (T - 1).choose D) ∧ ∀ (δ d ρ : ℕ), 2 ≤ D → 11 * D ^ 2 ≤ T → d ≤ δ * (T + 2 * D + 6).choose D → (ρ + δ * (T - 1).choose D ≤ d → (ρ : ℝ) ≤ 22 * (D : ℝ) ^ 2 * d / T) ∧ (ρ + δ * (T + D).choose D ≤ d → (ρ : ℝ) ≤ 22 * (D : ℝ) ^ 2 * d / T)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardLatticeQuotientHalfLevel

theorem **natCard_quot_smul_top_of_projective** {R N : Type} [CommRing R] [IsDomain R] [AddCommGroup N] [Module R N] [Module.Finite R N] [Module.Projective R N] (I : Ideal R) [Finite (R ⧸ I)] : Nat.card (N ⧸ (I • ⊤ : Submodule R N)) = Nat.card (R ⧸ I) ^ Module.finrank R N

theorem **absNorm_level_half** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (l : Level H) : Ideal.absNorm l.Mb = l.M ^ FD.fdeg E

theorem **card_levQ** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H), Finite (LevQ Dw l) ∧ Nat.card (LevQ Dw l) = l.M ^ (FD.fdeg E * (Dw.p + Dw.q))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardLatticeQuotientHalfLevelStatement

theorem **_root_.GlobalTheory.T.card_lattice_quotient_half_level** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H), Finite (LevQ Dw l) ∧ Nat.card (LevQ Dw l) = l.M ^ (FD.fdeg E * (Dw.p + Dw.q))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardLevGrpAndSpan

theorem **finrank_fibre** (O M k : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] [Field k] [Algebra O k] : Module.finrank k (k ⊗[O] M) = Module.finrank O M

theorem **nonempty_equiv_mod** (O M : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (I : Ideal O) [Finite (O ⧸ I)] (r : ℕ) (hr : Module.finrank O M = r) : Nonempty ((M ⧸ (I • ⊤ : Submodule O M)) ≃ₗ[O] (Fin r → O ⧸ I))

theorem **nonempty_aut_equiv** (O N : Type) [CommRing O] [AddCommGroup N] [Module O N] (I : Ideal O) (r : ℕ) (e : N ≃ₗ[O] (Fin r → O ⧸ I)) : Nonempty ((N ≃ₗ[O] N) ≃ GL (Fin r) (O ⧸ I))

theorem **span_sup_pow** (O M : Type) [CommRing O] [AddCommGroup M] [Module O M] (a : O) (S : Submodule O M) (h : S ⊔ Ideal.span {a} • ⊤ = ⊤) (k : ℕ) : S ⊔ Ideal.span {a ^ k} • ⊤ = ⊤

theorem **finite_quot** (k : ℕ) (hk : k ≠ 0) : Finite (𝒪 E ⧸ Ideal.span {(k : 𝒪 E)})

theorem **card_quot** (k : ℕ) : Nat.card (𝒪 E ⧸ Ideal.span {(k : 𝒪 E)}) = k ^ Module.finrank ℚ E

theorem **charP_quot** (k : ℕ) : CharP (𝒪 E ⧸ Ideal.span {(k : 𝒪 E)}) k

theorem **finrank_eq** [IsCMField E] : Module.finrank ℚ E = 2 * FD.fdeg E

theorem **card_levGrp_and_span** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (R : ℕ), 1 < R → Finite (LevGrp Dw (R ^ 4)) ∧ Finite (LatMod Dw R) ∧ Nat.card (LatMod Dw R) = R ^ (2 * FD.fdeg E * (Dw.p + Dw.q)) ∧ Nat.card (LevGrp Dw (R ^ 4)) ≤ R ^ (8 * FD.fdeg E * (Dw.p + Dw.q) ^ 2) ∧ (Odd R → 2 ≤ Dw.p + Dw.q → 2 * R ^ 4 ∣ Nat.card (LevGrp Dw (R ^ 4))) ∧ (∃ x₀ : Fin (Dw.p + Dw.q) → Dw.La, Submodule.span (𝒪 E) (Set.range x₀) ⊔ (Ideal.span {(R : 𝒪 E)} • ⊤) = ⊤) ∧ ∀ (n : ℕ) (x₀ : Fin n → Dw.La), Submodule.span (𝒪 E) (Set.range x₀) ⊔ (Ideal.span {(R : 𝒪 E)} • ⊤) = ⊤ → ∀ (y : Fin n → LatMod Dw R) (g : LevGrp Dw (R ^ 4)), Submodule.span (𝒪 E) (Set.range fun b => g (shiftFam R x₀ y b)) = ⊤

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardNonisometricCmTorusDataLe

theorem **mem_of_forall_mul_smul** {V : Type} [AddCommGroup V] [Module E V] (AA : Submodule (𝒪 E) E) (hfg : AA.FG) (h0 : AA ≠ ⊥) (N : AddSubmonoid V) (v : V) (h : ∀ a ∈ AA, ∀ b : E, (∀ c ∈ AA, ∃ o : 𝒪 E, (o : E) = b * c) → (a * b) • v ∈ N) : v ∈ N

theorem **smul_mem_divLat** {r : ℕ} {AA : Submodule (𝒪 E) E} {M : Submodule (𝒪 E) (Fin r → E)} {v : Fin r → E} (hv : v ∈ M) {b : E} (hb : ∀ c ∈ AA, ∃ o : 𝒪 E, (o : E) = b * c) : b • v ∈ divLat AA M

theorem **mem_iff_divLat** {r : ℕ} {AA : Submodule (𝒪 E) E} (hfg : AA.FG) (h0 : AA ≠ ⊥) (M : Submodule (𝒪 E) (Fin r → E)) (v : Fin r → E) : v ∈ M ↔ ∀ b : E, (∀ c ∈ AA, ∃ o : 𝒪 E, (o : E) = b * c) → b • v ∈ divLat AA M

theorem **isLattice_divLat** {r : ℕ} {AA : Submodule (𝒪 E) E} (hfg : AA.FG) (h0 : AA ≠ ⊥) {M : Submodule (𝒪 E) (Fin r → E)} (hM : M.FG) (hs : Submodule.span E (M : Set (Fin r → E)) = ⊤) : ReciprocityTheory.MultiplicativeHeights.IsLattice (divLat AA M)

theorem **herm_smul_smul** {r : ℕ} (G : Matrix (Fin r) (Fin r) E) (ξ a b : E) (x y : Fin r → E) : ξ * FD.herm G (a • x) (b • y) = ξ * FD.herm G x y * a * FD.cc b

theorem **selfDual_divLat** {r : ℕ} {AA : Submodule (𝒪 E) E} (hfg : AA.FG) (h0 : AA ≠ ⊥) (ξ : E) (hA : ∀ x : E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ E (ξ * x * a * FD.cc b) = z) ↔ IsIntegral ℤ x) (G : Matrix (Fin r) (Fin r) E) {M : Submodule (𝒪 E) (Fin r → E)} (hM : ∀ v, (∀ y ∈ M, ∃ n : ℤ, Algebra.trace ℚ E (ξ * FD.herm G v y) = n) ↔ v ∈ M) : (∀ x ∈ divLat AA M, ∀ y ∈ divLat AA M, IsIntegral ℤ (FD.herm G x y)) ∧ ∀ x, (∀ y ∈ divLat AA M, IsIntegral ℤ (FD.herm G x y)) → x ∈ divLat AA M

theorem **exists_isometry** {H : Host E} (D D' : Datum H) {r : ℕ} (ξ : E) (G : Matrix (Fin r) (Fin r) E) (φ : D.La →ₗ[𝒪 E] (Fin r → E)) (φ' : D'.La →ₗ[𝒪 E] (Fin r → E)) (hφ : Function.Injective φ) (hφ' : Function.Injective φ') (hE : ∀ x y, ((D.EF x y : ℤ) : ℚ) = Algebra.trace ℚ E (ξ * FD.herm G (φ x) (φ y))) (hE' : ∀ x y, ((D'.EF x y : ℤ) : ℚ) = Algebra.trace ℚ E (ξ * FD.herm G (φ' x) (φ' y))) (g : (Fin r → E) ≃ₗ[E] (Fin r → E)) (hg : ∀ x y, FD.herm G (g x) (g y) = FD.herm G x y) (hm : ∀ x, x ∈ LinearMap.range φ ↔ g x ∈ LinearMap.range φ') : ∃ gam : D.La ≃ₗ[𝒪 E] D'.La, ∀ a b, D'.EF (gam a) (gam b) = D.EF a b

theorem **card_nonisometric_data_le** : GlobalTheory.Published.HasseNormSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (p q r : ℕ), r = p + q → ∀ (ξ : E) (AA : Submodule (𝒪 E) E), FD.cc ξ = -ξ → ξ ≠ 0 → AA.FG → AA ≠ ⊥ → (∀ x : E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ E (ξ * x * a * FD.cc b) = z) ↔ IsIntegral ℤ x) → ∀ (m : ℕ) (Di : Fin m → Datum H), (∀ i, (Di i).p = p ∧ (Di i).q = q) → (∀ i j (gam : (Di i).La ≃ₗ[𝒪 E] (Di j).La), (∀ a b, (Di j).EF (gam a) (gam b) = (Di i).EF a b) → i = j) → (m : ℝ) ≤ (1 + 2 * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2) * (r * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardNonisometricCmTorusDataLeStatement

theorem **_root_.GlobalTheory.T.card_nonisometric_cmTorusData_le** : GlobalTheory.Published.HasseNormSt → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (p q r : ℕ), r = p + q → ∀ (ξ : E) (AA : Submodule (𝒪 E) E), FD.cc ξ = -ξ → ξ ≠ 0 → AA.FG → AA ≠ ⊥ → (∀ x : E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ E (ξ * x * a * FD.cc b) = z) ↔ IsIntegral ℤ x) → ∀ (m : ℕ) (Di : Fin m → Datum H), (∀ i, (Di i).p = p ∧ (Di i).q = q) → (∀ i j (gam : (Di i).La ≃ₗ[𝒪 E] (Di j).La), (∀ a b, (Di j).EF (gam a) (gam b) = (Di i).EF a b) → i = j) → (m : ℝ) ≤ (1 + 2 * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2) * (r * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardPeriodSpacesFamilyUniformisationLe

theorem **exists_injective_pi** {R M : Type} [CommRing R] [IsDomain R] [AddCommGroup M] [Module R M] [Module.Finite R M] [Module.IsTorsionFree R M] {r : ℕ} (hr : Module.finrank R M = r) : ∃ φ : M →ₗ[R] (Fin r → R), Function.Injective φ

theorem **exists_generators** (D : Datum H) {k : ℕ} (hk : 0 < k) : ∃ x : Fin (D.p + D.q) → D.La, ∀ m : D.La, ∃ (c : Fin (D.p + D.q) → 𝒪 E) (m' : D.La), m = ∑ i, c i • x i + k • m'

theorem **Isometric.refl** (D : Datum H) : Isometric D D

theorem **Isometric.symm** {D D' : Datum H} (h : Isometric D D') : Isometric D' D

theorem **Isometric.trans** {D D' D'' : Datum H} (h : Isometric D D') (h' : Isometric D' D'') : Isometric D D''

theorem **card_isometric_le** {D D₀ : Datum H} (h₀ : D₀.p + D₀.q = D.p + D.q) {k : ℕ} (hk : 0 < k) {n : ℕ} (Di : Fin n → Datum H) (t : ∀ i, D.La →ₗ[𝒪 E] (Di i).La) (hmatch : ∀ (i j : Fin n) (gam : (Di i).La ≃ₗ[𝒪 E] (Di j).La), (∀ a b, (Di j).EF (gam a) (gam b) = (Di i).EF a b) → (∀ a, ∃ e, gam (t i a) = t j a + k • e) → i = j) (s : Finset (Fin n)) (hs : ∀ i ∈ s, Isometric (Di i) D₀) : s.card ≤ k ^ (2 * FD.fdeg E * (D.p + D.q) ^ 2)

theorem **exists_representatives** {n : ℕ} (Di : Fin n → Datum H) : ∃ (m : ℕ) (rep : Fin m → Fin n), (∀ c d, Isometric (Di (rep c)) (Di (rep d)) → c = d) ∧ ∀ i, ∃ c, Isometric (Di i) (Di (rep c))

theorem **familyUnif_card_le** : GlobalTheory.Published.HasseNormSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (σ : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)), 1 ≤ D.p + D.q → σ = (baseKA l).ρ → H.rt0 → IsTypeFn H D.p D.q ty → ∀ U : FamilyUnif Xo σ ty, ∀ (r : ℕ), r = D.p + D.q → ∀ (ξ : E) (AA : Submodule (𝒪 E) E), FD.cc ξ = -ξ → ξ ≠ 0 → AA.FG → AA ≠ ⊥ → (∀ x : E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ E (ξ * x * a * FD.cc b) = z) ↔ IsIntegral ℤ x) → (U.n : ℝ) ≤ (l.M' : ℝ) ^ (2 * FD.fdeg E * r ^ 2) * ((1 + 2 * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2) * (r * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2)) ∧ ∀ L : ℝ, 16 ≤ FD.fdeg E → 2 ≤ L → Real.log (FD.dE E) ≤ 4 * FD.fdeg E * L → Real.log r

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardPeriodSpacesFamilyUniformisationLeStatement

theorem **_root_.GlobalTheory.T.card_periodSpaces_familyUniformisation_le** : GlobalTheory.Published.HasseNormSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (σ : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)), 1 ≤ D.p + D.q → σ (baseKA l).ρ = H.rt0 → IsTypeFn H D.p D.q ty → ∀ U : FamilyUnif Xo σ ty, ∀ (r : ℕ), r = D.p + D.q → ∀ (ξ : E) (AA : Submodule (𝒪 E) E), FD.cc ξ = -ξ → ξ ≠ 0 → AA.FG → AA ≠ ⊥ → (∀ x : E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ E (ξ * x * a * FD.cc b) = z) ↔ IsIntegral ℤ x) → (U.n : ℝ) ≤ (l.M' : ℝ) ^ (2 * FD.fdeg E * r ^ 2) * ((1 + 2 * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2) * (r * max 1 ((2 * FD.fdeg E * r : ℝ) ^ 4 * FD.dE E ^ 2 / (4 * FD.fdeg E))) ^ (2 * FD.fdeg E * r ^ 2)) ∧ ∀ L : ℝ, 16 ≤ FD.fdeg E → 2 ≤ L → Real.lo

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CardToMultisetEqDegree

theorem **card_toMultiset_eq_degree** (m : σ →₀ ℕ) : Multiset.card (Finsupp.toMultiset m) = m.degree

theorem **degree_eq_of_mem_support** {P : MvPolynomial σ ℂ} {a : ℕ} (hP : P.IsHomogeneous a) {m : σ →₀ ℕ} (hm : m ∈ P.support) : m.degree = a

theorem **polyDeriv_eq_sum_monomial** (P : MvPolynomial σ ℂ) (u : σ → W) (F : W → ℂ) (x : W) : polyDeriv P u F x = ∑ m ∈ P.support, P.coeff m * polyDeriv (MvPolynomial.monomial m (1 : ℂ)) u F x

theorem **polyDeriv_add_fun** [FiniteDimensional ℂ W] (P : MvPolynomial σ ℂ) (u : σ → W) {F G : W → ℂ} (hF : Differentiable ℂ F) (hG : Differentiable ℂ G) (x : W) : polyDeriv P u (F + G) x = polyDeriv P u F x + polyDeriv P u G x

theorem **polyDeriv_smul_fun** [FiniteDimensional ℂ W] (P : MvPolynomial σ ℂ) (u : σ → W) {F : W → ℂ} (hF : Differentiable ℂ F) (c : ℂ) (x : W) : polyDeriv P u (c • F) x = c * polyDeriv P u F x

theorem **polyDeriv_sum_fun** [FiniteDimensional ℂ W] {κ : Type*} (P : MvPolynomial σ ℂ) (u : σ → W) (s : Finset κ) (c : κ → ℂ) (F : κ → W → ℂ) (hF : ∀ i ∈ s, Differentiable ℂ (F i)) (x : W) : polyDeriv P u (fun w => ∑ i ∈ s, c i * F i w) x = ∑ i ∈ s, c i * polyDeriv P u (F i) x

theorem **polyDeriv_comp_equiv** (P : MvPolynomial σ ℂ) (u : σ → W) (F : W' → ℂ) (φ : W ≃L[ℂ] W') (x : W) : polyDeriv P u (F ∘ φ) x = polyDeriv P (φ ∘ u) F (φ x)

theorem **polyDeriv_comp_clm** [FiniteDimensional ℂ W'] (P : MvPolynomial σ ℂ) (u : σ → W) {F : W' → ℂ} (hF : Differentiable ℂ F) (φ : W →L[ℂ] W') (x : W) : polyDeriv P u (F ∘ φ) x = polyDeriv P (φ ∘ u) F (φ x)

theorem **polyDeriv_smul_dir** {P : MvPolynomial σ ℂ} {a : ℕ} (hP : P.IsHomogeneous a) (c : ℂ) (u : σ → W) (F : W → ℂ) (x : W) : polyDeriv P (fun i => c • u i) F x = c ^ a * polyDeriv P u F x

theorem **polyDeriv_comp_smul_at** [FiniteDimensional ℂ W] {P : MvPolynomial σ ℂ} {a : ℕ} (hP : P.IsHomogeneous a) (c : ℂ) (u : σ → W) {F : W → ℂ} (hF : Differentiable ℂ F) (x : W) : polyDeriv P u (fun v => F (c • v)) x = c ^ a * polyDeriv P u F (c • x)

theorem **polyDeriv_comp_smul** [FiniteDimensional ℂ W] {P : MvPolynomial σ ℂ} {a : ℕ} (hP : P.IsHomogeneous a) (c : ℂ) (u : σ → W) {F : W → ℂ} (hF : Differentiable ℂ F) : polyDeriv P u (fun v => F (c • v)) 0 = c ^ a * polyDeriv P u F 0

theorem **Hf_eq_inner** (v w : EuclideanSpace ℂ ι) : Hf v w = inner ℂ w v

theorem **differentiable_translate** (q : ℕ) (y : EuclideanSpace ℂ ι) {F : EuclideanSpace ℂ ι → ℂ} (hF : Differentiable ℂ F) : Differentiable ℂ fun v => cexp (q * Real.pi * Hf v y - q * Real.pi * Hf y y / 2) * F (v - y)

theorem **norm_translate** (q : ℕ) (y : EuclideanSpace ℂ ι) (F : EuclideanSpace ℂ ι → ℂ) (v : EuclideanSpace ℂ ι) : ‖cexp (q * Real.pi * Hf v y - q * Real.pi * Hf y y / 2) * F (v - y)‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v‖ ^ 2) = ‖F (v - y)‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v - y‖ ^ 2)

theorem **norm_polyDeriv_le** (q : ℕ) (hq : 1 ≤ q) (F : EuclideanSpace ℂ ι → ℂ) (hF : Differentiable ℂ F) (S : ℝ) (hS : ∀ v, ‖F v‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v‖ ^ 2) ≤ S) (P : MvPolynomial ι ℂ) (a : ℕ) (hP : P.IsHomogeneous a) : ‖polyDeriv P (fun i => EuclideanSpace.single i 1) F 0‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * Real.sqrt S * (q * Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2)

theorem **translate_jet_bound** {ι : Type} [Fintype ι] (U : ℕ → EuclideanSpace ℂ ι → (EuclideanSpace ℂ ι → ℂ) → EuclideanSpace ℂ ι → ℂ) (hU : ∀ (q : ℕ) a F v, U q a F v = cexp (q * Real.pi * Hf v a - q * Real.pi * Hf a a / 2) * F (v - a)) (q : ℕ) (hq : 1 ≤ q) (F : EuclideanSpace ℂ ι → ℂ) (hF : Differentiable ℂ F) (S : ℝ) (hS : ∀ v, ‖F v‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v‖ ^ 2) ≤ S) (P : MvPolynomial ι ℂ) (a : ℕ) (hP : P.IsHomogeneous a) (y : EuclideanSpace ℂ ι) : ‖polyDeriv P (fun i => EuclideanSpace.single i 1) (U q y F) 0‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * Real.sqrt S * (q * Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CmFrameExponentialJetsIsChart

theorem **chart_isHolFam_comp_clm** {Y : Scheme.{0}} {g N : ℕ} (e : (Fin g → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (A : (Fin N → ℂ) →L[ℂ] (Fin g → ℂ)) : IsHolFam fun v => e (A v)

theorem **chart_holTo_of_isHolFam** {Y : Scheme.{0}} {N : ℕ} {e : (Fin N → ℂ) → (Sp ℂ ⟶ Y)} (h : IsHolFam e) : IsHolTo Set.univ e

theorem **AnPt.chart_jx_left** (m : ℕ) : (Φ.jx m).left = jetPt (fun z => (Φ.e z).left) Φ.hol m

theorem **AnPt.chart_e_zero** : Φ.e 0 = 1

theorem **AnPt.chart_trunc** (m : ℕ) : homAt y.x (jetTrunc : Jet ℂ (Fin 1 × Fin g) (m + 1) →ₐ[ℂ] Jet ℂ (Fin 1 × Fin g) m) ≫ Φ.jx (m + 1) = Φ.jx m

theorem **AnPt.chart_zero** (m : ℕ) : homAt y.x (jetAug : Jet ℂ (Fin 1 × Fin g) m →ₐ[ℂ] ℂ) ≫ Φ.jx m = 1

theorem **AnPt.chart_act** (m : ℕ) (a : 𝒪 E) : Φ.jx m ≫ Xo.univ.t.act.act a = homAt y.x (jetSubstP m fun p : Fin 1 × Fin g => psTrunc m (anAct ty a p.2)) ≫ Φ.jx m

theorem **AnPt.chart_tan** (j0 : Fin g) (vt : Sp (DualNumber ℂ) ⟶ Xo.univ.t.G.left) (hd : IsDerivAt0 (fun z => (Φ.e z).left) (Pi.single j0 1) vt) : IsCombAt y.x (m

theorem **AnPt.chart_pair** : IsHolTo (Set.univ : Set (Fin (2 * g) → ℂ)) fun v => pullback.lift (Φ.e (anPr 0 v)).left (Φ.e (anPr 1 v)).left ((Over.w (Φ.e (anPr 0 v))).trans (Over.w (Φ.e (anPr 1 v))).symm)

theorem **AnPt.chart_law** (m : ℕ) : homAt y.x (jetSubstP m fun p : Fin 1 × Fin g => psTrunc m (addF ℂ g p.2)) ≫ Φ.jx m = jpt y g Φ.jx m (0 : Fin 2) * jpt y g Φ.jx m (1 : Fin 2)

theorem **anPt_isChart** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), IsChart y g (anFB ty).G.F (anFB ty).A.act (anUvec el) Φ.jx

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CmFrameExponentialJetsIsChartStatement

theorem **_root_.GlobalTheory.T.cmFrame_exponential_jets_isChart** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), IsChart y g (anFB ty).G.F (anFB ty).A.act (anUvec el) Φ.jx

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CmFrameJetPointsAreExponentialJets

theorem **isHolFam_precomp_clm** {Y : Scheme.{0}} {g N : ℕ} (e : (Fin g → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (A : (Fin N → ℂ) →L[ℂ] (Fin g → ℂ)) : IsHolFam (fun v => e (A v))

theorem **colPr_apply** (n g : ℕ) (i : Fin n) (v : Fin (n * g) → ℂ) (j : Fin g) : colPr n g i v j = v (finProdFinEquiv (i, j))

theorem **anLin_colPr** (n g : ℕ) (i : Fin n) (p : Fin 1 × Fin g) : anLin (colPr n g i) p = MvPolynomial.X ((0 : Fin 1), finProdFinEquiv (i, p.2))

theorem **renN_colPr** (n g m : ℕ) (i : Fin n) (a : Jet ℂ (Fin 1 × Fin g) m) : renN n g m (jetMap (anLin (colPr n g i)) (anLin_mem _) a) = jetCopy (S

theorem **jetCopy_one_zero** (g m : ℕ) (a : Jet ℂ (Fin 1 × Fin g) m) : jetCopy (S

theorem **spMap_jetCopy_jetPt** {Y : Scheme.{0}} {g : ℕ} (e : (Fin g → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (n m : ℕ) (i : Fin n) : spMap (jetCopy (S

theorem **AnFr.e_zero** : Φ.e 0 = 1

theorem **AnFr.jetMap_copy** (n m : ℕ) (i : Fin n) : Φ.JetMap m (homAt x0 (jetCopy (S

theorem **AnFr.jetMap_self** (m : ℕ) : Φ.JetMap (n

theorem **AnFr.jetMap_one** (n m : ℕ) : Φ.JetMap m (1 : overAt x0 (Jet ℂ (Fin n × Fin g) m) ⟶ A).left 0

theorem **AnFr.jetMap_comp** {n m : ℕ} (a : overAt x0 (Jet ℂ (Fin n × Fin g) m) ⟶ A) (Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (φ : A ⟶ A) (T : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (hT : ∀ z, Φ.e (T z) = Φ.e z ≫ φ) (ha : Φ.JetMap m a.left Λ) : Φ.JetMap m (a ≫ φ).left (T ∘ₗ Λ)

theorem **AnFr.jetMap_pair** {n m : ℕ} (a b : overAt x0 (Jet ℂ (Fin n × Fin g) m) ⟶ A) (Λ Λ' : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (ha : Φ.JetMap m a.left Λ) (hb : Φ.JetMap m b.left Λ') : ∃ hq : IsHolFam (fun v : Fin (n * g) → ℂ => (lift (Φ.e (Λ (colsOf n g v))) (Φ.e (Λ' (colsOf n g v))))).left), (lift a b).left = spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ jetPt _ hq m

theorem **AnFr.jetMap_mul** {n m : ℕ} (a b : overAt x0 (Jet ℂ (Fin n × Fin g) m) ⟶ A) (Λ Λ' : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (ha : Φ.JetMap m a.left Λ) (hb : Φ.JetMap m b.left Λ') : Φ.JetMap m (a * b).left (Λ + Λ')

theorem **jetPts_jetMap** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ (n m : ℕ), (∀ i : Fin n, Φ.toAnFr.JetMap m (jpt y g Φ.jx m i).left (LinearMap.proj i)) ∧ Φ.toAnFr.JetMap m (1 : jO y g m n ⟶ Xo.univ.t.G).left 0 ∧ (∀ (a b : jO y g m n ⟶ Xo.univ.t.G) (Λ Λ' : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.toAnFr.JetMap m a.left Λ → Φ.toAnFr.JetMap m b.left Λ' → Φ.toAnFr.JetMap m (a * b).left (Λ + Λ') ∧ ∃ hq : IsHolFam (fun v : Fin (n * g) → ℂ => (lift (Φ.e (Λ (colsOf n g v))) (Φ.e (Λ' (colsOf n g v))))).left), (lift a b).left = spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ jetPt _ hq m) ∧ (∀ (a : jO y g m n ⟶ Xo.univ.t.G) (Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CmFrameJetPointsAreExponentialJetsStatement

theorem **_root_.GlobalTheory.T.cmFrame_jetPoints_are_exponential_jets** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ (n m : ℕ), (∀ i : Fin n, Φ.toAnFr.JetMap m (jpt y g Φ.jx m i).left (LinearMap.proj i)) ∧ Φ.toAnFr.JetMap m (1 : jO y g m n ⟶ Xo.univ.t.G).left 0 ∧ (∀ (a b : jO y g m n ⟶ Xo.univ.t.G) (Λ Λ' : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.toAnFr.JetMap m a.left Λ → Φ.toAnFr.JetMap m b.left Λ' → Φ.toAnFr.JetMap m (a * b).left (Λ + Λ') ∧ ∃ hq : IsHolFam (fun v : Fin (n * g) → ℂ => (lift (Φ.e (Λ (colsOf n g v))) (Φ.e (Λ' (colsOf n g v))))).left), (lift a b).left = spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ jetPt _ hq m) ∧ (∀ (a : jO y g m n ⟶ Xo.univ.t.G

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CoefficientSpaceEqSpanOrbitExtremalVector

theorem **coeff_mem_of_eval_mem** {K A : Type} [Field K] [CharZero K] [CommRing A] [Algebra K A] (M : Submodule K A) (p : Polynomial A) (h : ∀ c : ℕ, p.eval (c : A) ∈ M) (k : ℕ) : p.coeff k ∈ M

theorem **eval_rowGen** {K : Type} [Field K] {r n : ℕ} (a b : Fin r) (c : K) (f : MvPolynomial (Fin r × Fin n) K) : (rowGen K a b f).eval (C c) = rowAct K (1 + Matrix.single b a c) f

theorem **isUnit_elem** {K : Type} [Field K] [CharZero K] {r : ℕ} (a b : Fin r) (c : ℕ) : IsUnit (1 + Matrix.single b a (c : K))

theorem **trGL_coe** {K : Type} [Field K] {r : ℕ} (g : GL (Fin r) K) : (trGL g : Matrix (Fin r) (Fin r) K) = (g : Matrix (Fin r) (Fin r) K)ᵀ

theorem **rowDiv_mem_of_stable** {K : Type} [Field K] [CharZero K] {r m : ℕ} (M : Submodule K (MvPolynomial (Fin r × Fin m) K)) (hst : ∀ u : GL (Fin r) K, ∀ f ∈ M, rowAct K (u : Matrix (Fin r) (Fin r) K) f ∈ M) (a b : Fin r) (k : ℕ) (f : MvPolynomial (Fin r × Fin m) K) (hf : f ∈ M) : rowDiv K a b k f ∈ M

theorem **applyWord_mem_of_stable** {K : Type} [Field K] [CharZero K] {r m : ℕ} (M : Submodule K (MvPolynomial (Fin r × Fin m) K)) (hst : ∀ u : GL (Fin r) K, ∀ f ∈ M, rowAct K (u : Matrix (Fin r) (Fin r) K) f ∈ M) (f : MvPolynomial (Fin r × Fin m) K) (hf : f ∈ M) (w : List (Fin r × Fin r × ℕ)) : applyWord K w f ∈ M

theorem **rowAct_one** {K : Type} [Field K] {r m : ℕ} : rowAct K (n

theorem **orbit_stable** {K : Type} [Field K] {r m : ℕ} (v : MvPolynomial (Fin r × Fin m) K) (u : GL (Fin r) K) (f : MvPolynomial (Fin r × Fin m) K) (hf : f ∈ Submodule.span K (Set.range fun g : GL (Fin r) K => rowAct K (g : Matrix (Fin r) (Fin r) K)ᵀ v)) : rowAct K (u : Matrix (Fin r) (Fin r) K) f ∈ Submodule.span K (Set.range fun g : GL (Fin r) K => rowAct K (g : Matrix (Fin r) (Fin r) K)ᵀ v)

theorem **dualWeyl_le_span_orbit** (K : Type) [Field K] [CharZero K] {r m : ℕ} (hm : m ≤ r) (ν : Fin m → ℕ) (hν : Antitone ν) : dualWeyl K r m ν ≤ Submodule.span K (Set.range fun g : GL (Fin r) K => rowAct K (g : Matrix (Fin r) (Fin r) K)ᵀ (hw K hm ν))

theorem **prod_mem_span_prod** {K A ι : Type} [Field K] [CommRing A] [Algebra K A] (s : Finset ι) (S : ι → Set A) (x : ι → A) (hx : ∀ i, x i ∈ Submodule.span K (S i)) : ∏ i ∈ s, x i ∈ Submodule.span K (∏ i ∈ s, S i)

theorem **trAct_ten** {K : Type} [Field K] {Ix : Type} [Fintype Ix] {n : ℕ} (g : Ix → GL (Fin n) K) (f : Ix → MvPolynomial (Fin n × Fin n) K) : trAct g (ten f) = ten fun t => rowAct K (g t : Matrix (Fin n) (Fin n) K)ᵀ (f t)

theorem **VV_eq_span_orbit** (K : Type) [Field K] [CharZero K] (Ix : Type) [Fintype Ix] (n : ℕ) (lam : Ix → DomWeight n) : VV K lam = Submodule.span K (Set.range fun g : Ix → GL (Fin n) K => actP lam g (ten fun t => extremal K n (lam t)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ColumnDegreeLe

theorem **eval_rowOps_eq** {K σ : Type*} [Field K] [DecidableEq σ] {p n : ℕ} (z : Fin p × Fin n → σ) (hz : Function.Injective z) (P : MvPolynomial σ K) (hinv : ∀ a b : Fin p, a < b → ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = z (b, k) then C t * X (z (a, k)) else 0) P = P) (i : ℕ) (c : Fin p × Fin p → K) (x : σ → K) (T : Finset (Fin p × Fin p)) (hT : ∀ q ∈ T, (q.1 : ℕ) < i ∧ i ≤ (q.2 : ℕ)) : eval (fun s => x s + ∑ q ∈ T, ∑ k, if s = z (q.2, k) then c q * x (z (q.1, k)) else 0) P = eval x P

theorem **eval_scale_eq** {K σ : Type*} [Field K] (w : σ → ℕ) (P : MvPolynomial σ K) (x : σ → K) (t : K) : eval (fun s => t ^ w s * x s) P = (∑ m ∈ P.support, Polynomial.monomial (∑ s ∈ m.support, w s * m s) (coeff m P * ∏ s ∈ m.support, x s ^ m s)).eval t

theorem **column_degree_le** {K σ : Type*} [Field K] [CharZero K] [DecidableEq σ] {p n : ℕ} (z : Fin p × Fin n → σ) (hz : Function.Injective z) (P : MvPolynomial σ K) (lam : Fin p → ℕ) (hrow : ∀ a, ∀ m ∈ P.support, ∑ k, m (z (a, k)) = lam a) (hinv : ∀ a b : Fin p, a < b → ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = z (b, k) then C t * X (z (a, k)) else 0) P = P) (S : Finset (Fin n)) : ∀ m ∈ P.support, ∑ a, ∑ k ∈ S, m (z (a, k)) ≤ ∑ a ∈ Finset.univ.filter (fun a : Fin p => (a : ℕ) < S.card), lam a

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CommuteRowOpRightMixedOp

theorem **pderiv_swap** (u v : σ) (P : MvPolynomial σ K) : pderiv u (pderiv v P) = pderiv v (pderiv u P)

theorem **pderiv_X_mul'** [DecidableEq σ] (v w : σ) (Q : MvPolynomial σ K) : pderiv v (X w * Q) = X w * pderiv v Q + (if v = w then Q else 0)

@[simp] theorem **polar_apply** (u v : σ) (P : MvPolynomial σ K) : polar K u v P = X u * pderiv v P

theorem **polar_lie** [DecidableEq σ] (u v w s : σ) : polar K u v * polar K w s − polar K w s * polar K u v = (if v = w then polar K u s else 0) − (if s = u then polar K w v else 0)

theorem **colOp_apply** [Fintype I] (z : I × J → σ) (k l : J) (P : MvPolynomial σ K) : colOp K z k l P = ∑ a, X (z (a, k)) * pderiv (z (a, l)) P

theorem **rowOp_apply** [Fintype J] (z : I × J → σ) (a b : I) (P : MvPolynomial σ K) : rowOp K z a b P = ∑ k, X (z (a, k)) * pderiv (z (b, k)) P

theorem **mixedOp_apply** [Fintype I] [Fintype I'] (z : I × J → σ) (y : I' × J → σ) (k l : J) (P : MvPolynomial σ K) : mixedOp K z y k l P = ∑ a, X (z (a, k)) * pderiv (z (a, l)) P − ∑ b, X (y (b, l)) * pderiv (y (b, k)) P

theorem **rowOp_eq_colOp** [Fintype J] (z : I × J → σ) (a b : I) : rowOp K z a b = colOp K (fun w : J × I => z w.swap) a b

theorem **colOp_lie** [Fintype I] [DecidableEq J] (z : I × J → σ) (hz : Function.Injective z) (i j k l : J) : colOp K z i j * colOp K z k l − colOp K z k l * colOp K z i j = (if j = k then colOp K z i l else 0) − (if l = i then colOp K z k j else 0)

theorem **rowOp_lie** [Fintype J] [DecidableEq I] (z : I × J → σ) (hz : Function.Injective z) (i j k l : I) : rowOp K z i j * rowOp K z k l − rowOp K z k l * rowOp K z i j = (if j = k then rowOp K z i l else 0) − (if l = i then rowOp K z k j else 0)

theorem **commute_polar** {u v w s : σ} (h₁ : v ≠ w) (h₂ : s ≠ u) : Commute (polar K u v) (polar K w s)

theorem **commute_colOp_colOp** [Fintype I] [Fintype I'] (z : I × J → σ) (y : I' × J' → σ) (hzy : ∀ u w, z u ≠ y w) (i j : J) (k l : J') : Commute (colOp K z i j) (colOp K y k l)

theorem **commute_rowOp_rowOp** [Fintype J] [Fintype J'] (z : I × J → σ) (y : I' × J' → σ) (hzy : ∀ u w, z u ≠ y w) (a b : I) (c d : I') : Commute (rowOp K z a b) (rowOp K y c d)

theorem **commute_rowOp_colOp** [Fintype J] [Fintype I'] (z : I × J → σ) (y : I' × J' → σ) (hzy : ∀ u w, z u ≠ y w) (a b : I) (k l : J') : Commute (rowOp K z a b) (colOp K y k l)

theorem **commute_rowOp_colOp_self** [Fintype I] [Fintype J] (z : I × J → σ) (hz : Function.Injective z) (a b : I) (k l : J) : Commute (rowOp K z a b) (colOp K z k l)

theorem **mixedOp_lie** [Fintype I] [Fintype I'] [DecidableEq J] (z : I × J → σ) (hz : Function.Injective z) (y : I' × J → σ) (hy : Function.Injective y) (hzy : ∀ u w, z u ≠ y w) (i j k l : J) : mixedOp K z y i j * mixedOp K z y k l − mixedOp K z y k l * mixedOp K z y i j = (if j = k then mixedOp K z y i l else 0) − (if l = i then mixedOp K z y k j else 0)

theorem **commute_mixedOp_mixedOp** {I₁ I₂ I₃ I₄ : Type*} [Fintype I₁] [Fintype I₂] [Fintype I₃] [Fintype I₄] (z : I₁ × J → σ) (y : I₂ × J → σ) (z' : I₃ × J' → σ) (y' : I₄ × J' → σ) (hzz : ∀ u w, z u ≠ z' w) (hzy : ∀ u w, z u ≠ y' w) (hyz : ∀ u w, y u ≠ z' w) (hyy : ∀ u w, y u ≠ y' w) (i j : J) (k l : J') : Commute (mixedOp K z y i j) (mixedOp K z' y' k l)

theorem **commute_rowOp_left_mixedOp** [Fintype I] [Fintype I'] [Fintype J] (z : I × J → σ) (hz : Function.Injective z) (y : I' × J → σ) (hzy : ∀ u w, z u ≠ y w) (a b : I) (k l : J) : Commute (rowOp K z a b) (mixedOp K z y k l)

theorem **commute_rowOp_right_mixedOp** [Fintype I] [Fintype I'] [Fintype J] (z : I × J → σ) (y : I' × J → σ) (hy : Function.Injective y) (hzy : ∀ u w, z u ≠ y w) (a b : I') (k l : J) : Commute (rowOp K y a b) (mixedOp K z y k l)

theorem **cross_term** {K σ : Type*} [CommRing K] {p q n : ℕ} (z : Fin p × Fin n → σ) (y : Fin q × Fin n → σ) (hzy : ∀ u w, z u ≠ y w) (P : MvPolynomial σ K) : (∑ k, ∑ l, ∑ a, X (z (a, k)) * pderiv (z (a, l)) (∑ b, X (y (b, k)) * pderiv (y (b, l)) P)) = ∑ a, ∑ b, (∑ k, X (z (a, k)) * X (y (b, k))) * ∑ l, pderiv (z (a, l)) (pderiv (y (b, l)) P)

theorem **cross_term'** {K σ : Type*} [CommRing K] {p q n : ℕ} (z : Fin p × Fin n → σ) (y : Fin q × Fin n → σ) (hzy : ∀ u w, z u ≠ y w) (P : MvPolynomial σ K) : (∑ k, ∑ l, ∑ b, X (y (b, k)) * pderiv (y (b, l)) (∑ a, X (z (a, k)) * pderiv (z (a, l)) P)) = ∑ a, ∑ b, (∑ k, X (z (a, k)) * X (y (b, k))) * ∑ l, pderiv (z (a, l)) (pderiv (y (b, l)) P)

theorem **capelli_casimir** {K σ : Type*} [CommRing K] [DecidableEq σ] {p n : ℕ} (z : Fin p × Fin n → σ) (hz : Function.Injective z) (P : MvPolynomial σ K) : (∑ k, ∑ l, ∑ a, X (z (a, k)) * pderiv (z (a, l)) (∑ b, X (z (b, l)) * pderiv (z (b, k)) P)) = (∑ a, ∑ b, ∑ k, X (z (a, k)) * pderiv (z (b, k)) (∑ l, X (z (b, l)) * pderiv (z (a, l)) P)) + ((n : K) − (p : K)) • ∑ w : Fin p × Fin n, X (z w) * pderiv (z w) P

theorem **casimir_colOp** (z : Fin p × Fin n → σ) (hz : Function.Injective z) (P : MvPolynomial σ K) : ∑ k, ∑ l, colOp K z k l (colOp K z l k P) = ∑ a, ∑ b, rowOp K z a b (rowOp K z b a P) + ((n : K) − (p : K)) • ∑ a, rowOp K z a a P

theorem **casimir_mixedOp** (z : Fin p × Fin n → σ) (hz : Function.Injective z) (y : Fin q × Fin n → σ) (hy : Function.Injective y) (hzy : ∀ u w, z u ≠ y w) (P : MvPolynomial σ K) : ∑ k, ∑ l, mixedOp K z y k l (mixedOp K z y l k P) = (∑ a, ∑ b, rowOp K z a b (rowOp K z b a P) + ((n : K) − (p : K)) • ∑ a, rowOp K z a a P) + (∑ a, ∑ b, rowOp K y a b (rowOp K y b a P) + ((n : K) − (q : K)) • ∑ a, rowOp K y a a P) − (2 : K) • ∑ a, ∑ b, (∑ k, X (z (a, k)) * X (y (b, k))) * ∑ l, pderiv (z (a, l)) (pderiv (y (b, l)) P)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ComponentModelWeightDimensionDictionary

theorem **tauEmb_inj** {t t' : D.what₂.Tau} (h : DV.tauEmb D t = DV.tauEmb D t') : t = t'

theorem **conjEmb_conjEmb** {E : Type} [Field E] (τ : E →+* ℂ) : conjEmb (conjEmb τ) = τ

theorem **tauEmb_ne_conj** (t t' : D.what₂.Tau) : DV.tauEmb D t ≠ conjEmb (DV.tauEmb D t')

theorem **wtA_tau** {j : ℕ} (w : WeightFamilies.Wt D j) (t : D.what₂.Tau) (i : ℕ) : wtA w (DV.tauEmb D t, i) = ext0 (w.aP t) i

theorem **wtA_conj** {j : ℕ} (w : WeightFamilies.Wt D j) (t : D.what₂.Tau) (i : ℕ) : wtA w (conjEmb (DV.tauEmb D t), i) = ext0 (w.aM t) i

theorem **matrixExtendByZero_mul** {p : ℕ} (c : ℕ) (f : Fin p → ℕ) (i : ℕ) : ext0 (fun k => c * f k) i = c * ext0 f i

theorem **wtA_wsmul** {j : ℕ} (c : ℕ) (w : WeightFfamilies.Wt D j) (x : (D.host.E →+* ℂ) × ℕ) : wtA (wsmul c w) x = c * wtA w x

theorem **wtA_wscalar** (j n : ℕ) (hn : Even n) (x : (D.host.E →+* ℂ) × ℕ) : wtA (wscalar D j n hn) x = 0

theorem **laR_wadd** (T : Top D) (j : ℕ) (k : (T.U j T.σ₀).Comp) (u u' : WeightFamilies.Wt D j) : laR T j (wadd u u') k = laR T j u k + laR T j u' k

theorem **laR_wsmul** (T : Top D) (j : ℕ) (k : (T.U j T.σ₀).Comp) (m : ℕ) (u : WeightFamilies.Wt D j) : laR T j (wsmul m u) k = m • laR T j u k

theorem **card_lower** (r : ℕ) : Fintype.card {p : Fin r × Fin r // p.2 < p.1} = r * (r - 1) / 2

theorem **choose_split** (p q : ℕ) : (p + q) * (p + q - 1) / 2 = p * q + p * (p - 1) / 2 + q * (q - 1) / 2

theorem **weights_dictionary** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (k : (T.U j T.σ₀).Comp), (∀ u u' : WeightFamilies.Wt D j, laR T j (wadd u u') k = laR T j u k + laR T j u' k) ∧ (∀ (m : ℕ) (u : WeightFamilies.Wt D j), laR T j (wsmul m u) k = m • laR T j u k) ∧ (∀ (u v : WeightFamilies.Wt D j) (m : ℕ), laR T j (addL u v m) k = laR T j u k + m • laR T j v k) ∧ (∀ i : ℕ, laR T j (etWt D j i) k = (i * (g0 D / 2)) • de2W ((T.U j T.σ₀).X k)) ∧ nd ((T.U j T.σ₀).X k) = WeightFamilies.dimY D j ∧ ((T.U j T.σ₀).X k).p0 = (D.R.sigW j).2 ∧ ((T.U j T.σ₀).X k).Ψ.card = D.R.f ∧ ∃ tt : ((T.U j T.σ₀).X k).Ψ ≃ D.what₂.Tau, ∀ (u : WeightFamilies.Wt D j) (τ : ((T.U j T.σ₀).X k).Ψ) (a : Fin D.R.r), laR T j u k τ a = if ((T.U j T.σ₀).X k).p0 ≤ (a : ℕ) then (ext0 (u.aP (tt τ)) ((a : ℕ) - ((T.U j T.σ₀).X k).p0) : ℤ) + ((u.n / 2 : ℕ) : ℤ) else -(ext0 (u.aM (tt τ)) (((T.U j T.σ₀).X k).p0 - 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ComponentModelWeightDimensionDictionaryStatement

theorem **_root_.GlobalTheory.T.componentModelWeight_dimension_dictionary** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (k : (T.U j T.σ₀).Comp), (∀ u u' : WeightFamilies.Wt D j, laR T j (wadd u u') k = laR T j u k + laR T j u' k) ∧ (∀ (m : ℕ) (u : WeightFamilies.Wt D j), laR T j (wsmul m u) k = m • laR T j u k) ∧ (∀ (u v : WeightFamilies.Wt D j) (m : ℕ), laR T j (addL u v m) k = laR T j u k + laR T j v k) ∧ (∀ i : ℕ, laR T j (etWt D j i) k = (i * (g0 D / 2)) • de2W ((T.U j T.σ₀).X k)) ∧ nd ((T.U j T.σ₀).X k) = WeightFamilies.dimY D j ∧ ((T.U j T.σ₀).X k).p0 = (D.R.sigW j).2 ∧ ((T.U j T.σ₀).X k).Ψ.card = D.R.f ∧ ∃ tt : ((T.U j T.σ₀).X k).Ψ ≃ D.what₂.Tau, ∀ (u : WeightFamilies.Wt D j) (τ : ((T.U j T.σ₀).X k).Ψ) (a : Fin D.R.r), laR T j u k τ a = if ((T.U j T.σ₀).X k).p0 ≤ (a : ℕ) then (ext0 (u.aP (tt τ)) ((a : ℕ) - ((T.U j T.σ₀).X k).p0) : ℤ) + ((u.n / 2 : ℕ) : ℤ) else -(

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConjCmHalfIntegralGeneratorEqOneSub

theorem **exists_multiple_mem** (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) (v : Fin r → E) : ∃ c : 𝒪 E, c ≠ 0 ∧ (c : E) • v ∈ X.Λ

theorem **herm_smul_smul** (G : Matrix (Fin r) (Fin r) E) (a b : E) (x y : Fin r → E) : herm G (a • x) (b • y) = a * ReciprocityTheory.ChartRestriction.G.FD.cc b * herm G x y

theorem **exists_isIso** (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) {ρ : ℕ} (cu : FJ.CuspFD X ρ) (hρ : 0 < ρ) : ∃ ξ : Fin r → E, X.IsIso ξ

theorem **conj_cmHalfIntegralGenerator_eq_one_sub** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ReciprocityTheory.ChartRestriction.G.FD.cc (H.w0 : E) = 1 - (H.w0 : E)

theorem **even_im_hermitian_sub** (X : DataX E r) (w0 : 𝒪 E) (hw : ReciprocityTheory.ChartRestriction.G.FD.cc (w0 : E) = 1 - (w0 : E)) : ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - ReciprocityTheory.ChartRestriction.G.FD.cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k

theorem **extra_weight_exists** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (w : WeightFamilies.Wt D j), GamAt T.toCore → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → j < D.R.m → (D.R.f : ℝ) ^ 2 ≤ D.host.rdE → 16 ≤ D.host.rdE → D.R.m ^ 4 ≤ D.R.f → ∀ k : (T.U j T.σ₀).Comp, BasePotentialExistence ((T.U j T.σ₀).X k) (de2W ((T.U j T.σ₀).X k)) D.R.r (1 / (2 * (5 : ℝ) ^ 4 * |(NumberField.discr D.host.E : ℝ)| ^ ((1 : ℝ) / D.R.f)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConjEmbRt

theorem **conj_eq_neg_of_sq** {z : ℂ} {r : ℝ} (hr : 0 < r) (h : z ^ 2 = -(r : ℂ)) : (starRingEnd ℂ) z = -z

theorem **conjEmb_rt** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (τ : E →+* ℂ) : conjEmb τ (H.rt : E) = -τ (H.rt : E)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConjHalfIntegralGenerator

theorem **conj_eq_neg_of_sq_eq_neg** {z : ℂ} {d : ℝ} (hd : 0 < d) (h : z ^ 2 = -(d : ℂ)) : conj z = -z

theorem **complexConj_rt** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : IsCMField.complexConj E ((H.rt : 𝒪 E) : E) = -((H.rt : 𝒪 E) : E)

theorem **conj_halfIntegralGenerator** (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) : ReciprocityTheory.ChartRestriction.G.FD.cc ((H.w0 : 𝒪 E) : E) = (1 : E) - ((H.w0 : 𝒪 E) : E)

theorem **main** : ∀ {E : Type} [inst : Field E] [inst_1 : NumberField E] [inst_2 : NumberField.IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E), ReciprocityTheory.ChartRestriction.G.FD.cc (↑(ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host.w0 H) : E) = (1 : E) - (↑(ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host.w0 H) : E)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConjugateSectionValuesSemilinear

theorem **fixes_symm** (h : Fixes c ta) : Fixes c ta.symm

theorem **fromT_tw_test** (h : Fixes c ta) (hA : AllTest Xo c) (p : TRing c.F) (y : FrPt Xo c p.R) : FromT (twPt h y)

theorem **fromT_of_map** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] {φ : S →ₐ[c.F] T} {y : FrPt Xo c S} {z : FrPt Xo c T} (hy : FromT y) (hm : FrMap φ y z) : FromT z

theorem **frMap_tw** (h : Fixes c ta) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] {φ : S →ₐ[c.F] T} {y : FrPt Xo c S} {z : FrPt Xo c T} (hm : FrMap φ y z) : FrMap (twHom ta φ) (twPt h y) (twPt h z)

theorem **fromT_of_tw** (h : Fixes c ta) (hA : AllTest Xo c) {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} (hy : FromT (twPt h y)) : FromT y

theorem **evR_galRul** (h : Fixes c ta) (hWD : ValueWD Xo c) (hA : AllTest Xo c) {f : Rul Xo c} (hf : Natural c f) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : evR Xo c (galRul h f) y = (evR Xo c f (twPt h y)).val

theorem **galRul_mem_sec** (h : Fixes c ta) (hWD : ValueWD Xo c) (hA : AllTest Xo c) (hB : BorChartF Xo c) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {f : Rul Xo c} (hf : f ∈ sec Xo c a n) : galRul h f ∈ sec Xo c a n

theorem **galRul_sec_val_semilinear** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (ta : c.F ≃+* c.F) (h : Fixes c ta), (∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ f ∈ sec Xo c a n, galRul h f ∈ sec Xo c a n ∧ ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), evR Xo c (galRul h f) y = (evR Xo c f (twPt h y)).val) ∧ (∀ (t : c.F) (f g : Rul Xo c), galRul h (t • f + g) = ta t • galRul h f + galRul h g) ∧ ∀ (c₀ : Coeff l) (hle : c₀.F ≤ c.F), (∀ x : c₀.F, ta (incF c₀ c hle x) = incF c₀ c hle x) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n m : ℕ) (as : Fin m → c.F) (fs : Fin m → Rul Xo c₀), ∀ f ∈ sec Xo c a n, IsBC hle f as fs → IsBC hle (galRul h f) (fun i => ta (as i)) fs

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConjugateSectionValuesSemilinearStatement

theorem **_root_.GlobalTheory.T.conjugateSection_values_semilinear** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (ta : c.F ≃+* c.F) (h : Fixes c ta), (∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ f ∈ sec Xo c a n, galRul h f ∈ sec Xo c a n ∧ ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), evR Xo c (galRul h f) y = (evR Xo c f (twPt h y)).val) ∧ (∀ (t : c.F) (f g : Rul Xo c), galRul h (t • f + g) = ta t • galRul h f + galRul h g) ∧ ∀ (c₀ : Coeff l) (hle : c₀.F ≤ c.F), (∀ x : c₀.F, ta (incF c₀ c hle x) = incF c₀ c hle x) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n m : ℕ) (as : Fin m → c.F) (fs : Fin m → Rul Xo c₀), ∀ f ∈ sec Xo c a n, IsBC hle f as fs → IsBC hle (galRul h f) (fun i => ta (as i)) fs

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConnectionLinearInSection

theorem **dof_C_mul** (F : Fin d → PS k 2 d) (u : Fin d → k) (c : k) (f : PS k 1 d) : Germ.Dof F u (MvPowerSeries.C c * f) = MvPowerSeries.C c * Germ.Dof F u f

theorem **dof_iter_C_mul** (F : Fin d → PS k 2 d) (u : Fin d → k) (c : k) (n : ℕ) (f : PS k 1 d) : (Germ.Dof F u)^[n] (MvPowerSeries.C c * f) = MvPowerSeries.C c * (Germ.Dof F u)^[n] f

theorem **dmon_C_mul** (F : Fin d → PS k 2 d) {κ : Type} [Fintype κ] (u : κ → Fin d → k) (m : κ →₀ ℕ) (c : k) (f : PS k 1 d) : Germ.DmonOf F u m (MvPowerSeries.C c * f) = MvPowerSeries.C c * Germ.DmonOf F u m f

theorem **nabVal_C_mul** [Algebra ℚ k] {R : Type} [CommRing R] [Algebra R k] {O : Type} [CommRing O] [StarRing O] (B : Germ.FB R d O) (h : PS k 2 d) (w0 : O) (q : ℕ) (c : k) (f : PS k 1 d) {κ : Type} [Fintype κ] (u : κ → Fin d → k) (P : MvPolynomial κ k) : B.nabVal k h w0 q (MvPowerSeries.C c * f) u P = c * B.nabVal k h w0 q f u P

theorem **psJet_C** {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (c : k) : psJet m (MvPowerSeries.C c : MvPowerSeries κ k) = algebraMap k (Jet k κ m) c

theorem **gs_liftFun** {T : Over S0} (W : (T ⊗ A).left.Opens) (r : Γ(T.left, ⊤)) {Z : Scheme.{0}} (q : Z ⟶ (T ⊗ A).left) (z : Z) (hz : q z ∈ W) : gs q z W (liftFun A W r) = gs (q ≫ (fst T A).left) z ⊤ r

theorem **isValueAt_scale** {T T' : Over S0} (g : T' ⟶ T) {F F' : LB.Sec (LPow A L a T)} {r : Γ(T.left, ⊤)} (hF : ∀ j, F'.f j = liftFun A _ (T

theorem **appTop_toΓ** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (a : A) : ((spMap f).appTop).hom (toΓ A a) = toΓ B (f a)

theorem **nab_smul_section** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (N : ℕ) (F F' : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (r : S), (∀ j, F'.f j = liftFun Xo.univ.t.G _ (T

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConnectionLinearInSectionStatement

theorem **_root_.GlobalTheory.T.connection_linear_in_section** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (N : ℕ) (F F' : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (r : S), (∀ j, F'.f j = liftFun Xo.univ.t.G _ (T

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConnectionPolynomialExpressionCoeffFieldExtensio

theorem **totalDegree_map_inj** {σ R S : Type*} [CommSemiring R] [CommSemiring S] (f : R →+* S) (hf : Function.Injective f) (P : MvPolynomial σ R) : (MvPolynomial.map f P).totalDegree = P.totalDegree

theorem **polyNab_fieldBC** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (nab : NabV Xo c) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (S : Type) [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S) (k : ℤ) (F : Fin n → LB.Sec (Out.LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))), (MvPolynomial.map (incF c c' hle) P).totalDegree = P.totalDegree ∧ ∑ m ∈ (MvPolynomial.map (incF c c' hle) P).support, algebraMap c'.F S ((MvPolynomial.map (incF c c' hle) P).coeff m) * ∏ b, nabBC hle nab y k (F b) (colOf m b) = ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab (y.res hle h) k (F b) (colOf m b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConnectionPolynomialExpressionCoeffFieldExtensioStatement

theorem **_root_.GlobalTheory.T.connectionPolynomialExpression_coeffFieldExtension** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (nab : NabV Xo c) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (S : Type) [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S) (k : ℤ) (F : Fin n → LB.Sec (Out.LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))), (MvPolynomial.map (incF c c' hle) P).totalDegree = P.totalDegree ∧ ∑ m ∈ (MvPolynomial.map (incF c c' hle) P).support, algebraMap c'.F S ((MvPolynomial.map (incF c c' hle) P).coeff m) * ∏ b, nabBC hle nab y k (F b) (colOf m b) = ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab (y.res hle h) k (F b) (colOf m b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ConstantCoeffDpolyOfAdditiveTaylorPts

theorem **hasSubst_taylorDir** (u : τ → W) {φ : W → σ → ℂ} (h0 : φ 0 = 0) : HasSubst fun i => taylorDir u fun z => φ z i

private theorem **fderiv_comp_apply_eq_sum_pi_single** {φ : W → σ → ℂ} {f : (σ → ℂ) → ℂ} {z : W} (hφ : DifferentiableAt ℂ φ z) (hf : DifferentiableAt ℂ f (φ z)) (v : W) : fderiv ℂ (f ∘ φ) z v = ∑ i, fderiv ℂ f (φ z) (Pi.single i 1) * fderiv ℂ (fun y => φ y i) z v

private theorem **taylorDir_comp_eq_subst_taylor_of_analyticAt** (u : τ → W) {φ : W → σ → ℂ} (hφ : AnalyticAt ℂ φ 0) (h0 : φ 0 = 0) : ∀ D ∈ {D : MvPowerSeries τ ℂ | ∃ f : (σ → ℂ) → ℂ, AnalyticAt ℂ f 0 ∧ D = taylorDir u (f ∘ φ) - subst (fun i => taylorDir u fun z => φ z i) (taylor f)}, D = 0

theorem **taylorDir_comp** (u : τ → W) {φ : W → σ → ℂ} (hφ : AnalyticAt ℂ φ 0) (h0 : φ 0 = 0) {f : (σ → ℂ) → ℂ} (hf : AnalyticAt ℂ f 0) : taylorDir u (f ∘ φ) = subst (fun i => taylorDir u fun z => φ z i) (taylor f)

@[simp] theorem **curryPts_apply** (z : Fin n × Fin d → ℂ) (j : Fin n) (i : Fin d) : curryPts n d z j i = z (j, i)

@[simp] theorem **constantCoeff_taylorPts** (F : (Fin n → Fin d → ℂ) → ℂ) : constantCoeff (taylorPts n F) = F 0

@[simp] theorem **taylorPts_coord** (j : Fin n) (i : Fin d) : taylorPts n (fun z : Fin n → Fin d → ℂ => z j i) = X (j, i)

@[simp] theorem **taylorPts_const** (c : ℂ) : taylorPts n (fun _ : Fin n → Fin d → ℂ => c) = C c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ContinuousRmulPair

theorem **bor_eq_pi** (Ψ : Finset (E →+* ℂ)) (r : ℕ) : Bor Ψ r = Set.univ.pi fun _ : Ψ => Set.univ.pi fun a : Fin r => Set.univ.pi fun c : Fin r => if c < a then ({0} : Set ℂ) else if c = a then ({0}ᶜ : Set ℂ) else Set.univ

theorem **isPreconnected_bor** (Ψ : Finset (E →+* ℂ)) (r : ℕ) : IsPreconnected (Bor Ψ r)

theorem **continuous_rmul_pair** (Ψ : Finset (E →+* ℂ)) (r : ℕ) : Continuous fun p : Frame Ψ r × Frame Ψ r => rmul p.1 p.2

theorem **frames_isConnected** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E], GGConnStatement (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ContinuousThetaSectionPointwiseNormSq

theorem **eq_zero_of_im_lattice** (Y : GT ι) {a : EuclideanSpace ℂ ι} (h : ∀ y ∈ lat Y.b, (Hf y a).im = 0) : a = 0

theorem **mem_lattice_of_int** (Y : GT ι) {a : EuclideanSpace ℂ ι} (h : ∀ y ∈ lat Y.b, ∃ k : ℤ, (Hf y a).im = k) : a ∈ lat Y.b

theorem **exists_common_sublattice** (Y Y' : GT ι) (R : Finset (EuclideanSpace ℂ ι)) (hR' : ∀ y ∈ lat Y'.b, ∃ r ∈ R, y - r ∈ lat Y.b) : ∃ b : Module.Basis (Fin (2 * Fintype.card ι)) ℝ (EuclideanSpace ℂ ι), lat b ≤ lat Y.b ∧ lat b ≤ lat Y'.b

theorem **pairing_density_add** (Y : GT ι) {F : EuclideanSpace ℂ ι → ℂ} (hF : F ∈ Y.Sec 1) (y y' : EuclideanSpace ℂ ι) {x : EuclideanSpace ℂ ι} (hx : x ∈ lat Y.b) (v : EuclideanSpace ℂ ι) : U 1 y F (v + x) * (starRingEnd ℂ) (U 1 y' F (v + x)) * cexp (-(Real.pi * Hf (v + x) (v + x))) = cexp (2 * Real.pi * I * (Hf x (y - y')).im) * (U 1 y F v * (starRingEnd ℂ) (U 1 y' F v) * cexp (-(Real.pi * Hf v v)))

theorem **translate_differentiable** {F : EuclideanSpace ℂ ι → ℂ} (hF : Differentiable ℂ F) (a : EuclideanSpace ℂ ι) : Differentiable ℂ (U 1 a F)

theorem **pairing_density_continuous** {F : EuclideanSpace ℂ ι → ℂ} (hF : Differentiable ℂ F) (y y' : EuclideanSpace ℂ ι) : Continuous fun v => U 1 y F v * (starRingEnd ℂ) (U 1 y' F v) * cexp (-(Real.pi * Hf v v))

theorem **integral_pairing_density_eq_zero** (Y : GT ι) {F : EuclideanSpace ℂ ι → ℂ} (hF : F ∈ Y.Sec 1) {κ : Type} [Fintype κ] (b : Module.Basis κ ℝ (EuclideanSpace ℂ ι)) (hb : lat b ≤ lat Y.b) (y y' : EuclideanSpace ℂ ι) (hint : ∀ x ∈ lat b, ∃ k : ℤ, (Hf x (y - y')).im = k) (hne : y - y' ∉ lat Y.b) : ∫ v in ZSpan.fundamentalDomain b, U 1 y F v * (starRingEnd ℂ) (U 1 y' F v) * cexp (-(Real.pi * Hf v v)) = 0

theorem **translate_translate** (Y : GT ι) (hY : ∀ x ∈ lat Y.b, ∀ y ∈ lat Y.b, Y.ka (x + y) = Y.ka x * Y.ka y * cexp (Real.pi * I * (Hf x y).im)) {x y : EuclideanSpace ℂ ι} (hx : x ∈ lat Y.b) (hy : y ∈ lat Y.b) (F : EuclideanSpace ℂ ι → ℂ) : Y.ka x • U 1 x (Y.ka y • U 1 y F) = Y.ka (x + y) • U 1 (x + y) F

theorem **thetaSectionPointwiseNormSq_periodic** (Y : GT ι) {F : EuclideanSpace ℂ ι → ℂ} (hF : F ∈ Y.Sec 1) {γ : EuclideanSpace ℂ ι} (hγ : γ ∈ lat Y.b) (v : EuclideanSpace ℂ ι) : Y.ptN2 1 F (v + γ) = Y.ptN2 1 F v

theorem **continuous_thetaSectionPointwiseNormSq** (Y : GT ι) {F : EuclideanSpace ℂ ι → ℂ} (hF : Continuous F) : Continuous (Y.ptN2 1 F)

theorem **thetaSectionPointwiseNormSq_eq** (Y : GT ι) (F : EuclideanSpace ℂ ι → ℂ) (v : EuclideanSpace ℂ ι) : ((Y.ptN2 1 F v : ℝ) : ℂ) = F v * (starRingEnd ℂ) (F v) * cexp (-(Real.pi * Hf v v))

theorem **isogeny_sum_mem_and_norm** {ι : Type} [Fintype ι] (U : ℕ → EuclideanSpace ℂ ι → (EuclideanSpace ℂ ι → ℂ) → EuclideanSpace ℂ ι → ℂ) (hU : ∀ (q : ℕ) a F v, U q a F v = cexp (q * Real.pi * Hf v a - q * Real.pi * Hf a a / 2) * F (v - a)) (Y Y' : GT ι) (hY : (∀ x ∈ lat Y.b, ∀ y ∈ lat Y.b, Y.ka (x + y) = Y.ka x * Y.ka y * cexp (Real.pi * I * (Hf x y).im))) (hY' : (∀ x ∈ lat Y'.b, ∀ y ∈ lat Y'.b, Y'.ka (x + y) = Y'.ka x * Y'.ka y * cexp (Real.pi * I * (Hf x y).im))) (hag : ∀ x ∈ lat Y.b, x ∈ lat Y'.b → Y.ka x = Y'.ka x) (R : Finset (EuclideanSpace ℂ ι)) (hR : ∀ y ∈ R, y ∈ lat Y'.b) (hR' : ∀ y ∈ lat Y'.b, ∃! r ∈ R, y - r ∈ lat Y.b) (F : EuclideanSpace ℂ ι → ℂ) (hF : F ∈ Y.Sec 1) : (∑ y ∈ R, Y'.ka y • U 1 y F) ∈ Y'.Sec 1 ∧ Y'.l2 1 (∑ y ∈ R, Y'.ka y • U 1 y F) = R.card * Y.l2 1 F

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CoordLetterEquivApplyConjugate

theorem **isWeightedHomogeneous_rename_iff** {R σ τ M : Type*} [CommSemiring R] [AddCommMonoid M] {f : σ → τ} (hf : Function.Injective f) (w : τ → M) (P : MvPolynomial σ R) (d : M) : IsWeightedHomogeneous w (rename f P) d ↔ IsWeightedHomogeneous (w ∘ f) P d

theorem **rowDegree_le_of_rowAdd_eq** {R : Type*} [CommRing R] [IsDomain R] [CharZero R] {t : T} {r r' : Fin p ⊕ Fin q} (hr : r ≠ r') {P : Poly R T p q κ} (hP : P ≠ 0) {d d' : ℕ} (hd : IsWeightedHomogeneous (rowInd t r) P d) (hd' : IsWeightedHomogeneous (rowInd t r') P d') (h : ∀ s : ℤ, rowAdd t r r' (s : R) P = P) : d' ≤ d

@[simp] theorem **coordLetterEquiv_symm_xVar** (t : {τ // Ψ τ}) (a : Fin p) (k : Fin n) : (coordLetterEquiv n hΨ hp hq).symm (xVar t a k) = (k, (t.1, a.cast (hp _ t.2).symm))

@[simp] theorem **coordLetterEquiv_symm_yVar** (t : {τ // Ψ τ}) (b : Fin q) (k : Fin n) : (coordLetterEquiv n hΨ hp hq).symm (yVar t b k) = (k, (ComplexEmbedding.conjugate t.1, b.cast (hq _ t.2).symm))

theorem **conjEmb_eq_conjugate** (τ : E →+* ℂ) : Tor.conjEmb τ = ComplexEmbedding.conjugate τ

@[simp] theorem **conjEmb_conjEmb** (τ : E →+* ℂ) : Tor.conjEmb (Tor.conjEmb τ) = τ

theorem **emb_rt_sq** (τ : E →+* ℂ) : τ (H.rt : E) ^ 2 = -(H.d : ℂ)

theorem **conjEmb_rt** (τ : E →+* ℂ) : Tor.conjEmb τ (H.rt : E) = -τ (H.rt : E)

theorem **not_mem_hostType_iff** (τ : E →+* ℂ) : ¬ τ (H.rt : E) = H.rt0 ↔ Tor.conjEmb τ (H.rt : E) = H.rt0

theorem **mult_of_mem** {τ : E →+* ℂ} (h : τ (H.rt : E) = H.rt0) : D.mult τ = D.p

theorem **mult_conjEmb_of_mem** {τ : E →+* ℂ} (h : τ (H.rt : E) = H.rt0) : D.mult (Tor.conjEmb τ) = D.q

@[simp] theorem **letterEquiv_symm_xVar** (t : HostType H) (a : Fin D.p) (k : Fin n) : (letterEquiv D n).symm (xVar t a k) = Tor.letter t.1 a k (a.isLt.trans_eq (mult_of_mem D t.2).symm)

@[simp] theorem **letterEquiv_symm_yVar** (t : HostType H) (b : Fin D.q) (k : Fin n) : (letterEquiv D n).symm (yVar t b k) = Tor.letter (Tor.conjEmb t.1) b k (b.isLt.trans_eq (mult_conjEmb_of_mem D t.2).symm)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CorrOfJetsOfAnalyticFrameProducts

theorem **jet_point_eq** {K : Type} [Field K] {J : Type} {m : ℕ} (p : PrimeSpectrum (Jet K J m)) : p = IsLocalRing.closedPoint (Jet K J m)

theorem **jetOf_one'** : jetOf m (fun _ : Fin N → ℂ => (1 : ℂ)) = 1

theorem **jetOf_pow'** {F : (Fin N → ℂ) → ℂ} (hF : AnalyticAt ℂ F 0) (k : ℕ) : jetOf m (fun v => F v ^ k) = jetOf m F ^ k

theorem **jetOf_mul_inv** {F : (Fin N → ℂ) → ℂ} (hF : AnalyticAt ℂ F 0) (h0 : F 0 ≠ 0) : jetOf m F * jetOf m (fun v => (F v)⁻¹) = 1

theorem **jetOf_zpow** {F : (Fin N → ℂ) → ℂ} (hF : AnalyticAt ℂ F 0) (h0 : F 0 ≠ 0) (u : (Jet ℂ (Fin 1 × Fin N) m)ˣ) (hu : u.val = jetOf m F) (a : ℤ) : jetOf m (fun v => F v ^ a) = (u ^ a).val

theorem **jetOf_prod** {K : Type} (s : Finset K) {F : K → (Fin N → ℂ) → ℂ} (hF : ∀ k, AnalyticAt ℂ (F k) 0) : jetOf m (fun v => ∏ k ∈ s, F k v) = ∏ k ∈ s, jetOf m (F k)

theorem **jetN_prod_zpow** {n g : ℕ} (m : ℕ) {K : Type} [Fintype K] (f : K → (Fin g → ℂ) → ℂ) (hf : ∀ k, AnalyticAt ℂ (f k) 0) (h0 : ∀ k, f k 0 ≠ 0) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (ε : K → ℤ) : ∃ u : K → (Jet ℂ (Fin n × Fin g) m)ˣ, (∀ k, (u k).val = jetN m fun w => f k (Λ k w)) ∧ jetN m (fun w => ∏ k, f k (Λ k w) ^ ε k) = (∏ k, u k ^ ε k).val

theorem **prod_zpow_fiber** {M T K : Type*} [CommGroup M] [Fintype K] (s : Finset T) (φ : K → T) (hs : ∀ k, φ k ∈ s) (ε : K → ℤ) (w : T → M) : ∏ k, w (φ k) ^ ε k = ∏ ψ ∈ s, w ψ ^ (∑ k, if φ k = ψ then ε k else 0)

theorem **prod_zpow_eq** {M T K K' : Type*} [CommGroup M] [Fintype K] [Fintype K'] (φ : K → T) (ε : K → ℤ) (φ' : K' → T) (ε' : K' → ℤ) (u : K → M) (u' : K' → M) (h₁ : ∀ k l, φ k = φ l → u k = u l) (h₂ : ∀ k l, φ' k = φ' l → u' k = u' l) (h₃ : ∀ k l, φ k = φ' l → u k = u' l) (hs : ∀ ψ, (∑ k, if φ k = ψ then ε k else 0) = ∑ k', if φ' k' = ψ then ε' k' else 0) : ∏ k, u k ^ ε k = ∏ k, u' k ^ ε' k

theorem **origin_mem** {n m : ℕ} {ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (h : Φ.JetMap m ψ Λ) {U : A.left.Opens} (p : Sp (Jet ℂ (Fin n × Fin g) m)) (hp : ψ p ∈ U) : ⊤ ≤ (Φ.e 0).left ⁻¹ᵁ U

theorem **fr_analyticAt** (j : L.L.ι) (h : ⊤ ≤ (Φ.e 0).left ⁻¹ᵁ L.L.U j) : AnalyticAt ℂ (Φ.fr j) 0

theorem **corr_of_jetRd** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), Φ.Imm → ∀ (n m : ℕ) (K K' : Type) [Fintype K] [Fintype K'] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (φ' : K' → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε' : K' → ℤ) (τ' : LB.Gen (LB.LinComb L.L φ' ε')) (Λ' : K' → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.JetRd m φ ε τ Λ → Φ.JetRd m φ' ε' τ' Λ' → LB.SameSum φ ε φ' ε' → LB.Gen.Corr L.L φ ε φ' ε' τ τ'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CorrOfJetsOfAnalyticFrameProductsStatement

theorem **_root_.GlobalTheory.T.corr_of_jets_of_analytic_frame_products** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), Φ.Imm → ∀ (n m : ℕ) (K K' : Type) [Fintype K] [Fintype K'] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (φ' : K' → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε' : K' → ℤ) (τ' : LB.Gen (LB.LinComb L.L φ' ε')) (Λ' : K' → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.JetRd m φ ε τ Λ → Φ.JetRd m φ' ε' τ' Λ' → LB.SameSum φ ε φ' ε' → LB.Gen.Corr L.L φ ε φ' ε' τ τ'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CoverMulShortThetaSeriesIndepOfFramedLift

theorem **shortValAt_smul** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) (c : Coeff l) (nab : NabV Xo c) {n : ℕ} (w w' : (Fin n → LevQ Dw l) → c.F) (t : c.F) (hw : ∀ q, w' q = t * w q) (Λ : GenLat E n) (β : c.F) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S) (h : ShortValAt Xo c nab w Λ β N P y v) : ShortValAt Xo c nab w' Λ β N P y (algebraMap c.F S t * v)

theorem **cover_product_indep_of_lift** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), @ToroidalLevelCompatibility E _ _ _ H Dw → d.IsLev → d.l₂.A = l.A → ∀ [StarRing (𝒪 E)], IsNab d.Xo₂ d.c₂ d.nab₂ → (∃ ph ψ x₀, d.PhiPin ph ψ x₀ ∧ d.PsiTie ψ) → ∀ (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt d.l₂.M → Λ.LevOK N d.l₂.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) d.c₂.F), P ∈ HHa Dw.mult n d.c₂.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (x0 : Fin n → Dw.La) (β : d.c₂.F), ∀ (r' : d.Xo₂.X ⟶ Xo.X) (m' : SABMap d.Xo₂.univ.t Xo.univ.t r'), DeltaDat.EtaOK { d with r

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CoverMulShortThetaSeriesIndepOfFramedLiftStatement

theorem **_root_.GlobalTheory.T.cover_mul_shortThetaSeries_indep_of_framed_lift** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), @ToroidalLevelCompatibility E _ _ _ H Dw → d.IsLev → d.l₂.A = l.A → ∀ [StarRing (𝒪 E)], IsNab d.Xo₂ d.c₂ d.nab₂ → (∃ ph ψ x₀, d.PhiPin ph ψ x₀ ∧ d.PsiTie ψ) → ∀ (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt d.l₂.M → Λ.LevOK N d.l₂.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) d.c₂.F), P ∈ HHa Dw.mult n d.c₂.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (x0 : Fin n → Dw.La) (β : d.c₂.F), ∀ (r' : d.Xo₂.X ⟶ Xo.X) (m' : SABMap d.Xo₂.univ.t Xo.univ.t r'), DeltaDat.EtaOK { d with r

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CubicalReadingOneAnalyticFrame

theorem **psumL_apply** {n g : ℕ} (I : ULift.{0} (Finset (Fin n))) (x : Fin n → Fin g → ℂ) : psumL n g I x = ∑ a ∈ I.down, x a

theorem **sum_finset_fin_three** (f : Finset (Fin 3) → ℂ) : ∑ I : ULift.{0} (Finset (Fin 3)), f I.down = f ∅ + f {0} + f {1} + f {2} + f {0, 1} + f {0, 2} + f {1, 2} + f {0, 1, 2}

theorem **third_difference** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z w : Fin 3 → Fin g → ℂ) : ∑ I : thK, ((thε I : ℤ) : ℂ) * hermForm Hm (psumL 3 g I z) (psumL 3 g I w) = 0

theorem **e_zero** : Φ.e 0 = 1

theorem **e_add_period** (z : Fin g → ℂ) {a : Fin g → ℂ} (ha : a ∈ Φ.lat) : Φ.e (z + a) = Φ.e z

theorem **hol_lin** {N : ℕ} (Λ : (Fin N → ℂ) →ₗ[ℂ] (Fin g → ℂ)) : IsHolFam fun v => (Φ.e (Λ v)).left

theorem **hol_lift** {N : ℕ} {B C : Over S} (p : (Fin N → ℂ) → (Over.mk x0 ⟶ B)) (q : (Fin N → ℂ) → (Over.mk x0 ⟶ C)) (hp : IsHolFam fun v => (p v).left) (hq : IsHolFam fun v => (q v).left) : IsHolFam fun v => (lift (p v) (q v)).left

theorem **hol_tri** : IsHolFam fun z => (tri Φ (colsEquiv 3 g z)).left

theorem **tripleProductPoint_comp_projection** (x : Fin 3 → Fin g → ℂ) (k : Fin 3) : tri Φ x ≫ x3 A k = Φ.e (x k)

private theorem **tri_comp_foldr_lift_mul_eq_e_sum** (x : Fin 3 → Fin g → ℂ) (l : List (Fin 3)) : tri Φ x ≫ l.foldr (fun k acc => lift (x3 A k) acc ≫ μ) (toUnit (G3 A) ≫ η) = Φ.e (l.map x).sum

theorem **tri_psum** (x : Fin 3 → Fin g → ℂ) (I : Finset (Fin 3)) : tri Φ x ≫ psum A (x3 A) I = Φ.e (∑ a ∈ I, x a)

theorem **origin_face** : (lift (Φ.e 0) (Φ.e 0)).left ≫ (faceMap A 0).left = (tri Φ 0).left

theorem **tripleColumnPeriodLattice_span_real_top** : Submodule.span ℝ (lat3 Φ : Set (Fin (3 * g) → ℂ)) = ⊤

theorem **genRd_cubical** (cub : CubicalStr A L) (x : Fin 3 → Fin g → ℂ) : Φ.GenRd (thφ A (x3 A)) thε cub.t (tri Φ x).left (fun I => psumL 3 g I x) 1

theorem **GlobalTheory.cubical_reading_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ a b d : Fin g → ℂ, Φ.toAnFr.GenRd (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (Φ.e a) (lift (Φ.e b) (Φ.e d))).left (fun I => psumL 3 g I ![a, b, d]) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CubicalReadingOneAnalyticFrameStatement

theorem **_root_.GlobalTheory.T.cubical_reading_one_analyticFrame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ a b d : Fin g → ℂ, Φ.toAnFr.GenRd (thφ Xo.univ.t.G (x3 Xo.univ.t.G)) thε Xo.univ.t.cub.t (lift (Φ.e a) (lift (Φ.e b) (Φ.e d))).left (fun I => psumL 3 g I ![a, b, d]) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormFamiliesComplexCombinationOfReadings

theorem **rdOnto_rec** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) , LawsSh T → GamAt T.toCore → 0 < badN D → (∀ j, InR D j → ∀ k k', ((T.U j T.σ₀).X k).Gm = ((T.U j T.σ₀).X k').Gm ∧ ((T.U j T.σ₀).X k).ξ0 = ((T.U j T.σ₀).X k').ξ0) → ∀ u : WeightFamilies.Wt D j, 8 * D.R.f * D.R.r + 4 ≤ u.n → LastPos (T.Q.Dw j) (wtA u) → RdOnto h j u

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormFamiliesComplexCombinationOfReadingsStatement

theorem **_root_.GlobalTheory.T.cuspForm_families_complex_combination_of_readings** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) , LawsSh T → GamAt T.toCore → 0 < badN D → (∀ j, InR D j → ∀ k k', ((T.U j T.σ₀).X k).Gm = ((T.U j T.σ₀).X k').Gm ∧ ((T.U j T.σ₀).X k).ξ0 = ((T.U j T.σ₀).X k').ξ0) → ∀ u : WeightFamilies.Wt D j, 8 * D.R.f * D.R.r + 4 ≤ u.n → LastPos (T.Q.Dw j) (wtA u) → RdOnto h j u

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormHalfSupAtOrthonormalFrameFirstMin

theorem **half_sup_at_ortho_frame** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → LastPos (T.Q.Dw j) (wtA w) → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∀ f ∈ T.C j w, 0 < supN (T.U j σ).X (laU (T.U j σ) (wtA w) w.n) (rdU (T.U j σ) f) → ∃ (k : (T.U j σ).Comp) (xf : ((T.U j σ).X k).dom), (∀ τ' i', ‖dmin ((T.U j σ).X k).Gm xf.1 τ' i'‖ = 1) ∧ ((T.U j σ).X k).IsOrtho xf.1 ∧ supN (T.U j σ).X (laU (T.U j σ) (wtA w) w.n) (rdU (T.U j σ) f) ≤ 2 * ‖toC (evR (T.Q.Xo j) T.Q.c f ((T.U j σ).un k xf))‖ ∧ 0 < ((T.U j σ).X k).firstMinimumSq xf.1 ∧ Real.log (1 / ((T.U j σ).X k).firstMinimumSq xf.1) ≤ 80 * large₂.sc.L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormHalfSupAtOrthonormalFrameFirstMinStatement

theorem **_root_.GlobalTheory.T.cuspForm_halfSup_at_orthonormalFrame_firstMin** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → LastPos (T.Q.Dw j) (wtA w) → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∀ f ∈ T.C j w, 0 < supN (T.U j σ).X (laU (T.U j σ) (wtA w) w.n) (rdU (T.U j σ) f) → ∃ (k : (T.U j σ).Comp) (xf : ((T.U j σ).X k).dom), (∀ τ' i', ‖dmin ((T.U j σ).X k).Gm xf.1 τ' i'‖ = 1) ∧ ((T.U j σ).X k).IsOrtho xf.1 ∧ supN (T.U j σ).X (laU (T.U j σ) (wtA w) w.n) (rdU (T.U j σ) f) ≤ 2 * ‖toC (evR (T.Q.Xo j) T.Q.c f ((T.U j σ).un k xf))‖ ∧ 0 < ((T.U j σ).X k).firstMinimumSq xf.1 ∧ Real.log (1 / ((T.U j σ).X k).firstMinimumSq xf.1) ≤ 80 * large₂.sc.L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormVanishingAtPointsMulPowMemSpan

theorem **leviO_phla_add_nsmul** (x0 : Frame X.Ψ r) (F L : Wt X.Ψ r) (k : ℕ) {z : V (nd X)} (hz : z ∈ OmF X x0) (ξ : Fin (nd X) → ℂ) : leviO X x0 (phla X (F + k • L)) z ξ = leviO X x0 (phla X F) z ξ + k * leviO X x0 (phla X L) z ξ

theorem **pnorm_mul_pow_sq** (F L : Wt X.Ψ r) (m : ℕ) (g w : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : X.pnorm (F + m • L) (fun x => g x * w x ^ m) x ^ 2 = X.pnorm F g x ^ 2 * (X.pnorm L w x ^ 2) ^ m

theorem **model_division** (Xa : ArchInM X) (Fl : FlatM X Xa) (hm : Margins X Xa) (k : ℕ) (hfin : ∃ (K : ℕ) (g : Fin K → Frame X.Ψ r → ℂ), ∀ f ∈ X.SS (Xa.Fp + (k + (nd X + 1)) • Xa.L), ∃ c : Fin K → ℂ, ∀ x ∈ X.dom, f x = ∑ i, c i * g i x) (g : Frame X.Ψ r → ℂ) (hg : X.IsBddSec (Xa.Fp + k • Xa.L) g) (hv : ∀ j, g (Fl.xs j) = 0) : ∃ h : Fin (nd X) → Frame X.Ψ r → ℂ, (∀ i, h i ∈ X.SS (Xa.Fp + (k + nd X) • Xa.L)) ∧ ∀ x ∈ X.dom, g x * Fl.w0 x ^ (nd X + 1) = ∑ i, Fl.s i x * h i x

theorem **sp_le_span** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} {X : PJ D B P j} (R : Rdg X) (u : WeightFamilies.Wt D j) : ∃ (K : ℕ) (b : Fin K → B.Cusp j u), R.sp u Set.univ ≤ Submodule.span ℂ (Set.range fun i => R.ev (X.inc u (b i)))

theorem **vanishing_division** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _)⟩) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → ∀ k : ℕ, ∀ g ∈ R.sp (addL wF wL k) Set.univ, (∀ q, g (xs q) = 0) → ∃ m : ℕ, R.s w t 0 ^ (m + 1) * g ∈ R.deep w t (addL wF wL (k + m)) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormVanishingAtPointsMulPowMemSpanStatement

theorem **_root_.GlobalTheory.T.cuspForm_vanishing_at_points_mul_pow_mem_span** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _)⟩) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → ∀ k : ℕ, ∀ g ∈ R.sp (addL wF wL k) Set.univ, (∀ q, g (xs q) = 0) → ∃ m : ℕ, R.s w t 0 ^ (m + 1) * g ∈ R.deep w t (addL wF wL (k + m)) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormVanishingLowCoeffMemLeadingAddDeepSpan

theorem **exists_valBasis** (X : Data E r) (G : Wt X.Ψ r) {δ m : ℕ} (xs : Fin δ → Frame X.Ψ r) (t : Fin m → Frame X.Ψ r → ℂ) (ht : ∀ k, X.IsBddSec G (t k)) (hs : ∀ v : Fin δ → ℂ, ∃ a : Fin m → ℂ, ∀ j, ∑ k, a k * t k (xs j) = v j) : ∃ t' : Fin δ → Frame X.Ψ r → ℂ, (∀ k, X.IsBddSec G (t' k)) ∧ IsValBasisM X xs t'

theorem **division_one** (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (hm : Margins X Xa) (hfin : ∃ (K : ℕ) (g : Fin K → Frame X.Ψ r → ℂ), ∀ f ∈ X.SS (Xa.Fp + Xa.T • Xa.L), ∃ c : Fin K → ℂ, ∀ x ∈ X.dom, f x = ∑ i, c i * g i x) {m : ℕ} (t : Fin m → Frame X.Ψ r → ℂ) (ht : ∀ k, X.IsBddSec Xa.Fp (t k)) (hs : ∀ v : Fin Fl.δ → ℂ, ∃ a : Fin m → ℂ, ∀ j, ∑ k, a k * t k (Fl.xs j) = v j) (f : Frame X.Ψ r → ℂ) (hf : f ∈ X.SSgt (Xa.Fp + Xa.T • Xa.L) Xa.c1) (cc : (Fin (nd X) → ℕ) → Fin m → ℂ) (hO : ∀ j, (fun z : V (nd X) => f (cpt X (Fl.xs j) z) - leadM X Fl.s Fl.w0 t Xa.T (Xa.T - nd X) cc (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (Xa.T - nd X)) : ∃ h : (Fin (nd X) → ℕ) → Frame X.Ψ r → ℂ, (∀ α ∈ Finset.Nat.antidiagonalTuple (nd X) (Xa.T - nd X), h α ∈ X.SS (Xa.Fp + nd X • Xa.L)) ∧ ∀ x ∈ X.dom, f x - leadM X Fl.s Fl.w0 t Xa.T (Xa.T - nd X) cc x = ∑ α ∈ Finset.Nat.antidiagonalTuple (nd X) (Xa.T - nd

theorem **division_one'** (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (hm : Margins X Xa) {D Tn : ℕ} (hD : nd X = D) (hT : Xa.T = Tn) (hfin : ∃ (K : ℕ) (g : Fin K → Frame X.Ψ r → ℂ), ∀ f ∈ X.SS (Xa.Fp + Tn • Xa.L), ∃ c : Fin K → ℂ, ∀ x ∈ X.dom, f x = ∑ i, c i * g i x) {m : ℕ} (t : Fin m → Frame X.Ψ r → ℂ) (ht : ∀ k, X.IsBddSec Xa.Fp (t k)) (hs : ∀ v : Fin Fl.δ → ℂ, ∃ a : Fin m → ℂ, ∀ j, ∑ k, a k * t k (Fl.xs j) = v j) (f : Frame X.Ψ r → ℂ) (hf : f ∈ X.SSgt (Xa.Fp + Tn • Xa.L) Xa.c1) (cc : (Fin D → ℕ) → Fin m → ℂ) (hO : ∀ j, (fun z : V (nd X) => f (cpt X (Fl.xs j) z) - ∑ α ∈ multiLt D (Tn - D), ∑ k, cc α k * (∏ q, Fl.s (Fin.cast hD.symm q) (cpt X (Fl.xs j) z) ^ α q) * Fl.w0 (cpt X (Fl.xs j) z) ^ (Tn - ∑ q, α q) * t k (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (Tn - D)) : ∃ h : (Fin D → ℕ) → Frame X.Ψ r → ℂ, (∀ α ∈ Finset.Nat.antidiagonalTuple D (Tn - D), h α ∈ X.SS (Xa.Fp + D • Xa.L))

theorem **eventually_diff** (X : Data E r) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {P : Frame X.Ψ r → ℂ} (hP : DifferentiableOn ℂ P X.dom) : ∀ᶠ z in 𝒩 (0 : V (nd X)), DifferentiableAt ℂ (fun z => P (cpt X x z)) z

theorem **division_all** {κ : Type} (X : κ → Data E r) (Xa : ∀ k, ArchInM (X k)) (Fl : ∀ k, FlatM (X k) (Xa k)) (hm : ∀ k, Margins (X k) (Xa k)) {D Tn : ℕ} (hD : ∀ k, nd (X k) = D) (hT : ∀ k, (Xa k).T = Tn) (hfin : ∀ k, ∃ (K : ℕ) (g : Fin K → Frame (X k).Ψ r → ℂ), ∀ f ∈ (X k).SS ((Xa k).Fp + Tn • (Xa k).L), ∃ c : Fin K → ℂ, ∀ x ∈ (X k).dom, f x = ∑ i, c i * g i x) {m : ℕ} (t : Fin m → ∀ k, Frame (X k).Ψ r → ℂ) (ht : ∀ i k, (X k).IsBddSec (Xa k).Fp (t i k)) (hs : ∀ v : (Σ k, Fin (Fl k).δ) → ℂ, ∃ a : Fin m → ℂ, ∀ p, ∑ i, a i * t i p.1 ((Fl p.1).xs p.2) = v p) (f : ∀ k, Frame (X k).Ψ r → ℂ) (hf : ∀ k, f k ∈ (X k).SSgt ((Xa k).Fp + Tn • (Xa k).L) (Xa k).c1) : ∃ (cc : (Fin D → ℕ) → Fin m → ℂ) (h : ∀ k, (Fin D → ℕ) → Frame (X k).Ψ r → ℂ), (∀ k, ∀ α ∈ Finset.Nat.antidiagonalTuple D (Tn - D), h k α ∈ (X k).SS ((Xa k).Fp + D • (Xa k).L)) ∧ ∀ k, ∀ x ∈ (X k).dom, f k x - ∑ α ∈ multiLt D (Tn - D), ∑ i,

theorem **sp_le_span_fin** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} {X : PJ D B P j} (R : Rdg X) (u : WeightFamilies.Wt D j) : ∃ (m : ℕ) (g : Fin m → R.Y → ℂ), R.sp u Set.univ ≤ Submodule.span ℂ (Set.range g)

theorem **main** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨(Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → ∀ (CR c : ℝ), c1 = WeightFamilies.c1 D CR Tn c j → (∀ F ∈ WeightFamilies.MggK (recB h.toLawsCore) CR Tn c j (addL wF wL Tn), ∀ k : (T.U j T.σ₀).Comp, rdR h j F k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j (addL wF wL Tn) k) c1) → ∀ (m : ℕ) (θ : Fin m → R.Y → ℂ), (∀ i, θ i ∈ R.sp wF Set.univ) → Submodule.span ℂ (Set.range (vals xs θ)) =

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspFormVanishingLowCoeffMemLeadingAddDeepSpanStatement

theorem **_root_.GlobalTheory.T.cuspForm_vanishing_low_coeff_mem_leading_add_deep_span** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨(Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → ∀ (CR c : ℝ), c1 = WeightFamilies.c1 D CR Tn c j → (∀ F ∈ WeightFamilies.MggK (recB h.toLawsCore) CR Tn c j (addL wF wL Tn), ∀ k : (T.U j T.σ₀).Comp, rdR h j F k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j (addL wF wL Tn) k) c1) → ∀ (m : ℕ) (θ : Fin m → R.Y → ℂ), (∀

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.CuspReadingsLinearIndependent

theorem **cusp_readings_linearIndependent** : ReciprocityTheory.ChartRestriction.G.PullbackSections.AlgInjAtSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DatumNsmulEqZero

theorem **datum_nsmul_eq_zero** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (D : Datum H) {k : ℕ} (hk : k ≠ 0) {x : D.La} (h : k • x = 0) : x = 0

theorem **marking_iso_mod** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ' : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (F : FamilyUnif Xo σ' ty) (i : Fin F.n), (∀ a, (∃ b, F.t i a = l.M • b) → ∃ b, a = l.M • b) ∧ ∀ b, ∃ a e, b = F.t i a + l.M • e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DeepSubMulRecAll

theorem **deepSubMul_rec_all** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j), EtaleWeightGaps.DeepSubMul (recPJ h j hj) ∧ EtaleWeightGaps.DeepSubMul (recPlus h j hj) ∧ ∀ (j' : ℕ) (ka : WeightFamilies.Wt D j') (Nt : NetData (recB h.toLawsCore) j' ka) (hM : NetMax Nt) (hL : NetLeSup Nt), EtaleWeightGaps.DeepSubMul (B

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DeterminantBlockProductRenameTorusPolynomial

theorem **polyDeriv_eq_sum** {θ : W → ℂ} {x : W} (hθ : ContDiffAt ℂ ω θ x) {T : Type} (P : MvPolynomial T ℂ) (u : T → W) : I2.polyDeriv P u θ x = ∑ m ∈ P.support, P.coeff m * mD θ x ((Finsupp.toMultiset m).map u)

theorem **polyDeriv_rename** {θ : W → ℂ} {x : W} (hθ : ContDiffAt ℂ ω θ x) {T κ : Type} (ρ : T → κ) (hρ : Function.Injective ρ) (P : MvPolynomial T ℂ) (v : κ → W) : I2.polyDeriv (MvPolynomial.rename ρ P) v θ x = I2.polyDeriv P (v ∘ ρ) θ x

theorem **contDiffAt_of_diff** [FiniteDimensional ℂ W] {F : W → ℂ} (hF : Differentiable ℂ F) (x : W) : ContDiffAt ℂ ω F x

theorem **polyDeriv_scale** {J W : Type} [NormedAddCommGroup W] [NormedSpace ℂ W] {P : MvPolynomial J ℂ} {d : ℕ} (hP : P.IsHomogeneous d) (c : ℂ) (u : J → W) (F : W → ℂ) (x : W) : I2.polyDeriv P (fun l => c • u l) F x = c ^ d * I2.polyDeriv P u F x

theorem **det_X_isHomogeneous** {J K : Type} [CommRing K] (i : ℕ) (f : Fin i → Fin i → J) : (Matrix.of fun a b : Fin i => (MvPolynomial.X (f a b) : MvPolynomial J K)).det.IsHomogeneous i

theorem **blockProd_rename** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (K : Type) [Field K] (σ : K →+* ℂ) (P : MvPolynomial (LtN Dw n) K), IsBlockProd P → ∀ (dim : (E →+* ℂ) → ℕ) (rl : (E →+* ℂ) → (E →+* ℂ)) (hd : ∀ τ,

Dw.mult τ ≤ dim (rl τ)), Function.Injective rl → P ≠ 0 ∧ P.IsHomogeneous P.totalDegree ∧ ∃ (nb : ℕ) (bs : Fin nb → I2.Block E n dim), I2.Disj bs ∧ I2.DF bs = P.totalDegree ∧ ∀ (t : ℂ) (F : (Fin n → I2.Vsp dim) → ℂ), Differentiable ℂ F → I2.polyDeriv (MvPolynomial.map σ P) (fun ℓ => t • I2.dir ((ℓ.2.2, (rl ℓ.1, Fin.castLE (hd ℓ.1) ℓ.2.1)) : I2.Letter n dim)) F 0 = t ^ P.totalDegree * I2.polyDeriv (I2.P1 bs) I2.dir F 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DeterminantBlockProductRenameTorusPolynomialStatement

theorem **_root_.GlobalTheory.T.determinantBlockProduct_rename_torusPolynomial** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (K : Type) [Field K] (σ : K →+* ℂ) (P : MvPolynomial (LtN Dw n) K), IsBlockProd P → ∀ (dim : (E →+* ℂ) → ℕ) (rl : (E →+* ℂ) → (E →+* ℂ)) (hd : ∀ τ, Dw.mult τ ≤ dim (rl τ)), Function.Injective rl → P ≠ 0 ∧ P.IsHomogeneous P.totalDegree ∧ ∃ (nb : ℕ) (bs : Fin nb → I2.Block E n dim), I2.Disj bs ∧ I2.DF bs = P.totalDegree ∧ ∀ (t : ℂ) (F : (Fin n → I2.Vsp dim) → ℂ), Differentiable ℂ F → I2.polyDeriv (MvPolynomial.map σ P) (fun ℓ => t • I2.dir ((ℓ.2.2, (rl ℓ.1, Fin.castLE (hd ℓ.1) ℓ.2.1)) : I2.Letter n dim)) F 0 = t ^ P.totalDegree * I2.polyDeriv (I2.P1 bs) I2.dir F 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DifferentiableOnDetHMat

theorem **borMat_of_lt** (b : Frame X.Ψ r) (τ : X.Ψ) {a : Fin r} (h : (a : ℕ) < X.p0) (c : Fin r) : borMat X b (τ, a) (τ, c) = b τ c a

theorem **borMat_of_le** (b : Frame X.Ψ r) (τ : X.Ψ) {a : Fin r} (h : X.p0 ≤ (a : ℕ)) (c : Fin r) : borMat X b (τ, a) (τ, c) = (mat b τ)⁻¹ a c

theorem **borMat_of_ne** (b : Frame X.Ψ r) {p q : X.Ψ × Fin r} (h : q.1 ≠ p.1) : borMat X b p q = 0

theorem **borMat_eq_zero_of_lt** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) {p q : X.Ψ × Fin r} (h : (p.2 : ℕ) < X.p0) (hq : p.2 < q.2) : borMat X b p q = 0

theorem **borMat_eq_zero_of_le** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) {p q : X.Ψ × Fin r} (h : X.p0 ≤ (p.2 : ℕ)) (hq : q.2 < p.2) : borMat X b p q = 0

theorem **borMat_diag** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (p : X.Ψ × Fin r) : borMat X b p p = rowFac X (fun τ a => b τ a a) p

theorem **det_borMat** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (idx : Fin g ≃ X.Ψ × Fin r) : ((borMat X b).submatrix idx idx).det = ∏ p, rowFac X (fun τ a => b τ a a) p

theorem **det_borMat_ne_zero** {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (idx : Fin g ≃ X.Ψ × Fin r) : ((borMat X b).submatrix idx idx).det ≠ 0

theorem **per_rmul** (x : Frame X.Ψ r) {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (v : Fin r → E) : per X (rmul x b) v = borMat X b *ᵥ per X x v

theorem **per_rmul_idx** (idx : Fin g ≃ X.Ψ × Fin r) (x : Frame X.Ψ r) {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) (v : Fin r → E) : per X (rmul x b) v ∘ idx = (borMat X b).submatrix idx idx *ᵥ (per X x v ∘ idx)

theorem **per_lact** (x : Frame X.Ψ r) {γ : GL (Fin r) E} (hγ : ∀ u v, herm X.Gm ((γ : Matrix (Fin r) (Fin r) E) *ᵥ u) ((γ : Matrix (Fin r) (Fin r) E) *ᵥ v) = herm X.Gm u v) (v : Fin r → E) : per X (lact γ x) ((γ : Matrix (Fin r) (Fin r) E) *ᵥ v) = per X x v

theorem **differentiableOn_per** (v : Fin r → E) (p : X.Ψ × Fin r) : DifferentiableOn ℂ (fun x => per X x v p) X.dom

theorem **hermForm_Hm** (x : Frame X.Ψ r) (z w : Fin g → ℂ) : hermForm (Hm X bas idx x) z w = hermForm (imInv X bas idx x) (hMat X bas idx x *ᵥ z) (hMat X bas idx x *ᵥ w)

theorem **perMat_rmul** (s : Fin g → Fin g ⊕ Fin g) : perMat X bas idx s (rmul x b) = (borMat X b).submatrix idx idx * perMat X bas idx s x

theorem **hMat_rmul** : hMat X bas idx (rmul x b) = hMat X bas idx x * ((borMat X b).submatrix idx idx)⁻¹

theorem **hMat_rmul_mul** : hMat X bas idx (rmul x b) * (borMat X b).submatrix idx idx = hMat X bas idx x

theorem **det_hMat_rmul** : (hMat X bas idx (rmul x b)).det = (hMat X bas idx x).det * ∏ p, (rowFac X (fun τ a => b τ a a) p)⁻¹

theorem **tauMat_rmul** : tauMat X bas idx (rmul x b) = tauMat X bas idx x

theorem **imInv_rmul** : imInv X bas idx (rmul x b) = imInv X bas idx x

theorem **hermForm_Hm_rmul** (z w : Fin g → ℂ) : hermForm (Hm X bas idx (rmul x b)) ((borMat X b).submatrix idx idx *ᵥ z) ((borMat X b).submatrix idx idx *ᵥ w) = hermForm (Hm X bas idx x) z w

theorem **c_pos** (τ : X.Ψ) : 0 < X.c τ

theorem **emb_neg_half** (τ : X.Ψ) : τ.1 (-(X.κ0 / 2)) = I * (X.c τ / 2 : ℝ)

theorem **EF_eq_sum** (u v : Fin r → E) : (EF X u v : ℝ) = ∑ τ : X.Ψ, X.c τ * -(hermC τ.1 X.Gm (Data.emb τ.1 u) (Data.emb τ.1 v)).im

theorem **hermForm_diagonal_c** (z w : X.Ψ × Fin r → ℂ) : hermForm (Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)) (z ∘ idx) (w ∘ idx) = ∑ τ, (X.c τ : ℂ) * ∑ a, z (τ, a) * conj (w (τ, a))

theorem **im_hermForm_diagonal_c** {x : Frame X.Ψ r} (ho : X.IsOrtho x) (u v : Fin r → E) : (hermForm (Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)) (per X x u ∘ idx) (per X x v ∘ idx)).im = EF X u v

theorem **norm_per_of_isOrtho** {x : Frame X.Ψ r} (hx : x ∈ X.dom) (ho : X.IsOrtho x) (v : Fin r → E) (p : X.Ψ × Fin r) : ‖per X x v p‖ = ‖((mat x p.1)⁻¹ *ᵥ Data.emb p.1.1 v) p.2‖

theorem **hermForm_diagonal_c_self** {x : Frame X.Ψ r} (hx : x ∈ X.dom) (ho : X.IsOrtho x) (v : Fin r → E) : hermForm (Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)) (per X x v ∘ idx) (per X x v ∘ idx) = ((∑ τ, X.c τ * X.maj x τ (Data.emb τ.1 v) : ℝ) : ℂ)

theorem **gram_mul_im** (G A τ : Matrix (Fin g) (Fin g) ℂ) (hM : ∀ j k, ((Aᵀ * G * A.map conj) j k).im = 0) (hN : ∀ j k, ((Aᵀ * G * (A * τ).map conj) j k).im = -(1 : Matrix (Fin g) (Fin g) ℝ) j k) : Aᵀ * G * A.map conj * τ.map (fun z => (z.im : ℂ)) = 1

theorem **riemann_of_isOrtho** (hs : IsSympl X bas) {x : Frame X.Ψ r} (ho : X.IsOrtho x) : IsUnit (perMat X bas idx Sum.inl x).det ∧ (tauMat X bas idx x).IsSymm ∧ ((tauMat X bas idx x).map Complex.im).PosDef ∧ Hm X bas idx x = Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)

theorem **Hm_of_isOrtho** (hs : IsSympl X bas) {x : Frame X.Ψ r} (ho : X.IsOrtho x) : Hm X bas idx x = Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)

theorem **det_Hm_of_isOrtho** (hs : IsSympl X bas) {x : Frame X.Ψ r} (ho : X.IsOrtho x) : (Hm X bas idx x).det = ((∏ τ, X.c τ ^ r : ℝ) : ℂ)

theorem **isUnit_det_perMat** : IsUnit (perMat X bas idx Sum.inl x).det

theorem **isUnit_det_hMat** : IsUnit (hMat X bas idx x).det

theorem **hMat_mul_perMat** : hMat X bas idx x * perMat X bas idx Sum.inl x = 1

theorem **tauMat_isSymm** : (tauMat X bas idx x).IsSymm

theorem **tauMat_im_posDef** : ((tauMat X bas idx x).map Complex.im).PosDef

theorem **hMat_mulVec_per_bas** (m n : Fin g → ℤ) : hMat X bas idx x *ᵥ (per X x (bas (Sum.elim m n)) ∘ idx) = (fun k => (m k : ℂ)) + tauMat X bas idx x *ᵥ fun k => (n k : ℂ)

theorem **imInv_mul_im** : imInv X bas idx x * (tauMat X bas idx x).map (fun z => (z.im : ℂ)) = 1

theorem **imInv_isSymm** : (imInv X bas idx x).IsSymm

theorem **Hm_isHermitian** : (Hm X bas idx x).IsHermitian

theorem **hermForm_Hm_eq_diagonal** (z w : Fin g → ℂ) : hermForm (Hm X bas idx x) z w = hermForm (Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)) ((borMat X (FrameDegreeData.bOf X x)).submatrix idx idx *ᵥ z) ((borMat X (FrameDegreeData.bOf X x)).submatrix idx idx *ᵥ w)

theorem **hermForm_Hm_pos** {z : Fin g → ℂ} (hz : z ≠ 0) : 0 < (hermForm (Hm X bas idx x) z z).re

theorem **im_hermForm_Hm** (u v : Fin r → E) : (hermForm (Hm X bas idx x) (per X x u ∘ idx) (per X x v ∘ idx)).im = EF X u v

theorem **det_im_mul_det_Hm** : (((tauMat X bas idx x).map Complex.im).det : ℂ) * (Hm X bas idx x).det = ((‖(hMat X bas idx x).det‖ ^ 2 : ℝ) : ℂ)

theorem **det_Hm** : (Hm X bas idx x).det = ((‖(hMat X bas idx x).det‖ ^ 2 / ((tauMat X bas idx x).map Complex.im).det : ℝ) : ℂ)

theorem **det_Hm_re_pos** : 0 < (Hm X bas idx x).det.re

theorem **det_im_tauMat** : ((tauMat X bas idx x).map Complex.im).det = ‖(hMat X bas idx x).det‖ ^ 2 / (Hm X bas idx x).det.re

theorem **differentiableOn_hMat** (hs : IsSympl X bas) (i j : Fin g) : DifferentiableOn ℂ (fun x => hMat X bas idx x i j) X.dom

theorem **differentiableOn_det_hMat** (hs : IsSympl X bas) : DifferentiableOn ℂ (fun x => (hMat X bas idx x).det) X.dom

theorem **differentiableOn_tauMat** (hs : IsSympl X bas) (i j : Fin g) : DifferentiableOn ℂ (fun x => tauMat X bas idx x i j) X.dom

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DifferentiablePolyDeriv

theorem **differentiable_polyDeriv** (hF : ContDiff ℂ (⊤ : ℕ∞) F) (P : MvPolynomial σ ℂ) (u : σ → W) : Differentiable ℂ fun y => polyDeriv P u F y

theorem **polyDeriv_mul_X** (hF : ContDiff ℂ (⊤ : ℕ∞) F) (P : MvPolynomial σ ℂ) (u : σ → W) (l : σ) (x : W) : polyDeriv (P * X l) u F x = fderiv ℂ (fun y => polyDeriv P u F y) x (u l)

theorem **polyDeriv_algHom** (hF : ContDiff ℂ (⊤ : ℕ∞) F) (Φ : MvPolynomial σ ℂ →ₐ[ℂ] MvPolynomial σ ℂ) (s : σ → ℂ) (j : σ → σ) (hΦ : ∀ l, Φ (X l) = X l + C (s l) * X (j l)) (u : σ → W) (P : MvPolynomial σ ℂ) (x : W) : polyDeriv (Φ P) u F x = polyDeriv P (fun l => u l + s l • u (j l)) F x

theorem **eigen_of_triangular** (f : (J → V) → ℂ) (lt : J → J → Prop) (rk : J → ℕ) (hrk : ∀ i j, lt i j → rk i < rk j) (d : J → ℕ) (hadd : ∀ i j, lt i j → ∀ (s : ℂ) u, f (Function.update u j (u j + s • u i)) = f u) (hsc : ∀ j (s : ℂ) u, f (Function.update u j (s • u j)) = s ^ d j * f u) (N : J → J → ℂ) (hN : ∀ i j, i ≠ j → ¬ lt i j → N i j = 0) (u : J → V) : f (fun j => ∑ i, N i j • u i) = (∏ j, N j j ^ d j) * f u

theorem **polyDeriv_triangular** {T κ W : Type*} [Fintype T] [Fintype κ] {p q : ℕ} [NormedAddCommGroup W] [NormedSpace ℂ W] {F : W → ℂ} (hF : ContDiff ℂ (⊤ : ℕ∞) F) {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {P : Poly ℂ T p q κ} (hP : P ∈ rowHighest ℂ c c') (N : T × (Fin p ⊕ Fin q) → T × (Fin p ⊕ Fin q) → ℂ) (hN : ∀ i j, i ≠ j → ¬ (i.1 = j.1 ∧ RowLT i.2 j.2) → N i j = 0) (u : Var T p q κ → W) (x : W) : polyDeriv P (fun l => ∑ i, N i (l.1, l.2.1) • u (i.1, i.2, l.2.2)) F x = (∏ j, N j j ^ Sum.elim (partOf (c j.1)) (partOf (c' j.1)) j.2) * polyDeriv P u F x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.DivisionOrderOne

theorem **division_order_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r), (∃ bb m : ℝ, 4 * (X.Ψ.card : ℝ) * (r : ℝ) ^ 2 < bb ∧ bb ^ 2 * ((X.Ψ.card * min X.p0 (r - X.p0) : ℕ) : ℝ) + bb < m ∧ ∀ x0, IsOrth X x0 → ∀ z ∈ OmF X x0, ∀ ξ : Fin (nd X) → ℂ, m * leviRh X x0 z ξ ≤ leviO X x0 (phla X G) z ξ - leviDens X x0 z ξ) → (∀ P, X.IsBddSec (G + (nd X + 1) • Xa.L) P → P ∈ X.SS (G + (nd X + 1) • Xa.L)) → (∃ (K : ℕ) (g : Fin K → Frame X.Ψ r → ℂ), ∀ f ∈ X.SS (G + (nd X + 1) • Xa.L), ∃ c : Fin K → ℂ, ∀ x ∈ X.dom, f x = ∑ i, c i * g i x) → ∀ f, X.IsBddSec (G + (nd X + 1) • Xa.L) f → (∃ C : ℝ, ∀ x ∈ X.dom, X.pnorm (G + (nd X + 1) • Xa.L) f x ^ 2 ≤ C * (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (nd X + 1)) → (∀ j, f (Fl.xs j) = 0) → ∃ h : Fin (nd X) → Frame X.Ψ r → ℂ, (∀ k, X.IsBddSec (G + nd X • Xa.L) (h k)) ∧ ∀ x ∈ X.dom, f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EmbeddingHermSelf

theorem **herm_one_eq** {n : ℕ} (x y : Fin n → E) : herm 1 x y = ∑ k, x k * cc (y k)

theorem **herm_smul_left** {n : ℕ} (a : E) (x y : Fin n → E) : herm 1 (a • x) y = a * herm 1 x y

theorem **herm_smul_right** {n : ℕ} (a : E) (x y : Fin n → E) : herm 1 x (a • y) = cc a * herm 1 x y

theorem **cc_cc** (a : E) : cc (cc a) = a

theorem **cc_herm_self** {n : ℕ} (x : Fin n → E) : cc (herm 1 x x) = herm 1 x x

theorem **trace_cc** (a : E) : Algebra.trace ℚ E (cc a) = Algebra.trace ℚ E a

theorem **isIntegral_cc** {a : E} (ha : IsIntegral ℤ a) : IsIntegral ℤ (cc a)

theorem **embedding_herm_self** {n : ℕ} (σ : E →+* ℂ) (x : Fin n → E) : σ (herm 1 x x) = ((∑ k, ‖σ (x k)‖ ^ 2 : ℝ) : ℂ)

theorem **mem_inv_different** {x : E} (hx : ∀ a : 𝒪 E, ∃ k : ℤ, Algebra.trace ℚ E (x * a) = k) : x ∈ (differentIdeal ℤ (𝒪 E) : FractionalIdeal (𝒪 E)⁰ E)⁻¹

theorem **trace_dual_min** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) → x ∈ Λ) (u : Fin n → E) (hu : u ≠ 0) (htr : ∀ y ∈ Λ, ∃ k : ℤ, Algebra.trace ℚ E (herm 1 u y) = k) : 2 * (fdeg E : ℝ) / rdE E ^ 2 ≤ ∑ σ : E →+* ℂ, ∑ k, ‖σ (u k)‖ ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EmbeddingsDifferOnInteger

theorem **embeddings_differ_on_integer** {E : Type} [Field E] [NumberField E] (τ τ' : E →+* ℂ) (h : τ ≠ τ') : ∃ a : 𝒪 E, τ (a : E) ≠ τ' (a : E)

theorem **balanced_pairs_conjugate** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} (c : Coeff l) {S : Type} [CommRing S] [Algebra c.F S] {d : ℕ} (A : 𝒪 E → Matrix (Fin d) (Fin d) S) (uvec : Lt1 Dw → Fin d → S) (heig : ∀ (a : 𝒪 E) (ℓ : Lt1 Dw) (i : Fin d), ∑ j, A a i j * uvec ℓ j = algebraMap c.F S (embK c ℓ.1 a) * uvec ℓ i) (M : Fin d → Fin d → S) (hbal : ∀ a, (A a).transpose * Matrix.of M = Matrix.of M * A (H.bar a)) (ℓ ℓ' : Lt1 Dw) (hne : ∑ i, ∑ j, M i j * uvec ℓ i * uvec ℓ' j ≠ 0) : ℓ'.1 = ComplexEmbedding.conjugate ℓ.1

theorem **twist_on_eigenDirections** (hFix : FrTruncFixSt) (hLE : LinEigSt) : FrTruncFixBSt

theorem **eigenDirections** : LinEigSt

theorem **chartIndep_of_globalFns** (hGlob : ReciprocityTheory.ChartRestriction.G.Out.GlobalFnsSt) : ChartIndepSt2

theorem **chartIndep** : ChartIndepSt2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EqOfForallAlgClosed

theorem **eq_of_forall_algClosed** {K R : Type} [Field K] [CommRing R] [Algebra K R] [IsReduced R] (u v : R) (h : ∀ (k : Type) [Field k] [IsAlgClosed k] [Algebra K k] (φ : R →ₐ[K] k), φ u = φ v) : u = v

theorem **sec_eq_of_geometric** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {a : (E →+* ℂ) × ℕ → ℕ} {m : ℕ} {f f' : Rul Xo c} (hf : f ∈ sec Xo c a m) (hf' : f' ∈ sec Xo c a m) (h : ∀ (k : Type) [Field k] [IsAlgClosed k] [Algebra c.F k] (y : FrPt Xo c k), evR Xo c f y = evR Xo c f' y) : f = f'

theorem **isThClsD_unique** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n) (Cl : Type) [Fintype Cl] (w : Cl → (Fin n → LevQ Dw d.l₂) → c.F) (La : Cl → GenLat E n) (Nc gam : Cl → ℕ) (vol : c.F) (ep : Cl → c.F) (Vs : Type) [AddCommGroup Vs] [Module c.F Vs] (io : Vs →ₗ[c.F] MvPolynomial (LtN Dw n) c.F), d.IsLev → (∃ _ : StarRing (𝒪 E), IsNab d.Xo₂ d.c₂ d.nab₂) → (∀ k, (La k).Unimod ∧ (La k).TrivAt d.l₂.M ∧ (La k).LevOK (Nc k) d.l₂.M) → ∀ (a : (E →+* ℂ) × ℕ → ℕ), (∀ v, io v ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) → ∀ (x : Cl → Vs) (a' : (E →+* ℂ) × ℕ → ℕ) (m : ℕ), ∀ f ∈ sec Xo c a' m, ∀ f' ∈ sec Xo c a' m, IsThClsD Xo c d w La Nc gam vol ep io x f → IsThClsD Xo c d w La Nc gam vol ep

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EqZeroOfForallAlgHom

theorem **eq_zero_of_forall_algHom** {K R Ω : Type} [Field K] [CommRing R] [Algebra K R] [IsReduced R] [Algebra.FiniteType K R] [Field Ω] [IsAlgClosed Ω] [Algebra K Ω] (x : R) (h : ∀ φ : R →ₐ[K] Ω, φ x = 0) : x = 0

theorem **pow_eq_one_of_complex** {K : Type} [Field K] (pts : ∀ (S : Type) [CommRing S] [Algebra K S], Type) (mapR : ∀ {S T : Type} [CommRing S] [Algebra K S] [CommRing T] [Algebra K T], (S →ₐ[K] T) → pts S → pts T → Prop) (hW : EvalWD pts mapR) (hB : ∀ (S : Type) [CommRing S] [Algebra K S] (T : Type) [CommRing T] [Algebra K T] (φ : S →ₐ[K] T) (y : pts S), ∃ z : pts T, mapR φ y z) (σ : K →+* ℂ) (f : Rules pts) (hf : IsNatural pts mapR f) (k : ℕ) (h : ∀ y : pts (CAt σ), evalAt pts mapR f y ^ k = 1) (p : TRing K) (y : pts p.R) : f p y ^ k = 1

theorem **evalAt_pow_eq_one** {K : Type} [Field K] (pts : ∀ (S : Type) [CommRing S] [Algebra K S], Type) (mapR : ∀ {S T : Type} [CommRing S] [Algebra K S] [CommRing T] [Algebra K T], (S →ₐ[K] T) → pts S → pts T → Prop) (f : Rules pts) (k : ℕ) (h : ∀ (p : TRing K) (y : pts p.R), f p y ^ k = 1) {S : Type} [CommRing S] [Algebra K S] (y : pts S) (hT : FromTest pts mapR y) : evalAt pts mapR f y ^ k = 1

theorem **GlobalTheory.FiniteOrderRule.unitC_of_cmp** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n) (Cls : Type) [CommGroup Cls] [Finite Cls] (δ : Cls →* d.c₂.Fˣ) (gdet : FrPt d.Xo₂ d.c₂ (CAt d.c₂.F.val.toRingHom) → Cls), CMP Xo c d _ Cls δ gdet false → UnitC d

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EqZeroOfPartialSumsNonneg

theorem **_root_.GlobalTheory.Harm.sum_filter_lt_eq_sum_range** {m : ℕ} (f : Fin m → ℤ) (i : ℕ) : ∑ k ∈ Finset.univ.filter (fun k : Fin m => (k : ℕ) < i), f k = ∑ j ∈ range i, if h : j < m then f ⟨j, h⟩ else 0

private theorem **sum_range_sub_mul_eq_abel_summation** (D e : ℕ → ℤ) (n : ℕ) : ∑ i ∈ range n, (D (i + 1) - D i) * e i = D n * e n + ∑ i ∈ range n, D (i + 1) * (e i - e (i + 1)) - D 0 * e 0

theorem **_root_.GlobalTheory.Harm.casimir_value_strict** {m : ℕ} (κ κ₀ : Fin m → ℤ) (h : Antitone κ) (h₀ : Antitone κ₀) (hle : ∀ i : ℕ, ∑ k ∈ Finset.univ.filter (fun k : Fin m => (k : ℕ) < i), κ k ≤ ∑ k ∈ Finset.univ.filter (fun k : Fin m => (k : ℕ) < i), κ₀ k) (hsum : ∑ k, κ k = ∑ k, κ₀ k) (hC : ∑ i, κ₀ i * (κ₀ i + m - 1 - 2 * (i : ℕ)) ≤ ∑ i, κ i * (κ i + m - 1 - 2 * (i : ℕ))) : κ = κ₀

theorem **mixedWeight_apply** (m : ℕ) {p q : ℕ} (a : Fin p → ℕ) (b : Fin q → ℕ) (k : Fin m) : mixedWeight m a b k = (if h : (k : ℕ) < p then (a ⟨k, h⟩ : ℤ) else 0) - (if h : m - 1 - (k : ℕ) < q then (b ⟨m - 1 - k, h⟩ : ℤ) else 0)

theorem **_root_.GlobalTheory.Harm.antitone_mixedWeight** (m : ℕ) {p q : ℕ} {a : Fin p → ℕ} {b : Fin q → ℕ} (ha : Antitone a) (hb : Antitone b) : Antitone (_root_.GlobalTheory.Harm.mixedWeight m a b)

theorem **_root_.GlobalTheory.Harm.sum_dite_lt** {m p : ℕ} (hp : p ≤ m) (f : Fin p → ℤ) : ∑ k : Fin m, (if h : (k : ℕ) < p then f ⟨k, h⟩ else 0) = ∑ k, f k

theorem **_root_.GlobalTheory.Harm.sum_dite_rev_lt** {m q : ℕ} (hq : q ≤ m) (f : Fin q → ℤ) : ∑ k : Fin m, (if h : m - 1 - (k : ℕ) < q then f ⟨m - 1 - k, h⟩ else 0) = ∑ k, f k

theorem **_root_.GlobalTheory.Harm.sum_mixedWeight** {m p q : ℕ} (h : p + q ≤ m) (a : Fin p → ℕ) (b : Fin q → ℕ) : ∑ k, _root_.GlobalTheory.Harm.mixedWeight m a b k = ∑ i, (a i : ℤ) - ∑ i, (b i : ℤ)

theorem **_root_.GlobalTheory.Harm.casimir_value_mixedWeight** {m p q : ℕ} (h : p + q ≤ m) (a : Fin p → ℕ) (b : Fin q → ℕ) : ∑ k : Fin m, _root_.GlobalTheory.Harm.mixedWeight m a b k * (_root_.GlobalTheory.Harm.mixedWeight m a b k + m - 1 - 2 * (k : ℕ)) = (∑ i : Fin p, (a i : ℤ) * (a i + p - 1 - 2 * (i : ℕ)) + ((m : ℤ) - p) * ∑ i, (a i : ℤ)) + (∑ i : Fin q, (b i : ℤ) * (b i + q - 1 - 2 * (i : ℕ)) + ((m : ℤ) - q) * ∑ i, (b i : ℤ))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EtaValAtExistsUniqueNatural

theorem **finite_latMod** (Dw : Datum H) {k : ℕ} (hk : 0 < k) : Finite (LatMod Dw k)

theorem **finite_levGrp** (Dw : Datum H) {k : ℕ} (hk : 0 < k) : Finite (LevGrp Dw k)

theorem **map_etaExpr** {S T : Type} [CommRing S] [CommRing T] (φ : S →+* T) {k : ℕ} (hk : 0 < k) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) : φ (etaExpr k cf x w) = etaExpr k cf x (fun q => φ (w q))

theorem **etaExpr_comp** {S : Type} [CommRing S] {k : ℕ} {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) (g₀ : LevGrp Dw k) : etaExpr k cf x (fun q => w (g₀ q)) = etaExpr k cf x w

theorem **exists_level_baseChange** (hL : Laws) {R : ℕ} (hR : Odd R) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) {y : FrPt Xo c S} {z : FrPt Xo c T} (hm : FrMap ψ y z) {ev : LatMod Dw (R ^ 4) → (ptO Xo y.x ⟶ Xo.univ.t.G)} {w : LatMod Dw (R ^ 4) → S} (hev : IsFullLev Xo (R ^ 4) y.x ev) (hw : ∀ q, IsNullVal Xo c (R ^ 2) y (ev q) (w q)) : ∃ ev' : LatMod Dw (R ^ 4) → (ptO Xo z.x ⟶ Xo.univ.t.G), IsFullLev Xo (R ^ 4) z.x ev' ∧ ∀ q, IsNullVal Xo c (R ^ 2) z (ev' q) (ψ (w q))

theorem **existsUnique_nullVal** (hL : Laws) {R : ℕ} (hR : Odd R) {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} {ev : LatMod Dw (R ^ 4) → (ptO Xo y.x ⟶ Xo.univ.t.G)} (hev : IsFullLev Xo (R ^ 4) y.x ev) (q : LatMod Dw (R ^ 4)) : ∃! w, IsNullVal Xo c (R ^ 2) y (ev q) w

theorem **etaExpr_eq_of_levels** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw (R ^ 4)) {T : Type} [CommRing T] [Algebra c.F T] {z : FrPt Xo c T} {ev ev' : LatMod Dw (R ^ 4) → (ptO Xo z.x ⟶ Xo.univ.t.G)} {w w' : LatMod Dw (R ^ 4) → T} (hev : IsFullLev Xo (R ^ 4) z.x ev) (hw : ∀ q, IsNullVal Xo c (R ^ 2) z (ev q) (w q)) (hev' : IsFullLev Xo (R ^ 4) z.x ev') (hw' : ∀ q, IsNullVal Xo c (R ^ 2) z (ev' q) (w' q)) : etaExpr (R ^ 4) cf x w = etaExpr (R ^ 4) cf x w'

theorem **map_etaExpr_eq_of_levels** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw (R ^ 4)) {S₁ S₂ T : Type} [CommRing S₁] [Algebra c.F S₁] [CommRing S₂] [Algebra c.F S₂] [CommRing T] [Algebra c.F T] (ψ₁ : S₁ →ₐ[c.F] T) (ψ₂ : S₂ →ₐ[c.F] T) {y₁ : FrPt Xo c S₁} {y₂ : FrPt Xo c S₂} {z : FrPt Xo c T} (h₁ : FrMap ψ₁ y₁ z) (h₂ : FrMap ψ₂ y₂ z) {ev₁ : LatMod Dw (R ^ 4) → (ptO Xo y₁.x ⟶ Xo.univ.t.G)} {w₁ : LatMod Dw (R ^ 4) → S₁} {ev₂ : LatMod Dw (R ^ 4) → (ptO Xo y₂.x ⟶ Xo.univ.t.G)} {w₂ : LatMod Dw (R ^ 4) → S₂} (hev₁ : IsFullLev Xo (R ^ 4) y₁.x ev₁) (hw₁ : ∀ q, IsNullVal Xo c (R ^ 2) y₁ (ev₁ q) (w₁ q)) (hev₂ : IsFullLev Xo (R ^ 4) y₂.x ev₂) (hw₂ : ∀ q, IsNullVal Xo c (R ^ 2) y₂ (ev₂ q) (w₂ q)) : ψ₁ (etaExpr (R ^ 4) cf x w₁) = ψ₂ (etaExpr (R ^ 4) cf x w₂)

theorem **map_eq_of_etaValAt** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] {n : ℕ} {cf : J → ℤ} {x : J → Fin n → LatMod Dw (R ^ 4)} {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {y₀ : FrPt Xo c S} {y : FrPt Xo c T} (hm : FrMap φ y₀ y) {v₀ : S} {v : T} (h₀ : EtaValAt Xo c R cf x y₀ v₀) (h : EtaValAt Xo c R cf x y v) : φ v₀ = v

theorem **exists_etaValAt** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw (R ^ 4)) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : ∃ v, EtaValAt Xo c R cf x y v

theorem **etaValAt_existsUnique_natural** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (R : ℕ), Odd R → ∀ (J : Type) [Fintype J] (n : ℕ) (cf : J → ℤ) (x : J → Fin n → LatMod Dw (R ^ 4)), (∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), ∃! v, EtaValAt Xo c R cf x y v) ∧ ∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y0 : FrPt Xo c S) (y : FrPt Xo c T), FrMap φ y0 y → ∀ v0 v, EtaValAt Xo c R cf x y0 v0 → EtaValAt Xo c R cf x y v → φ v0 = v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EvRSumSmulFin

theorem **evR_sum_smul_fin** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) (c : Coeff l) {ι : Type} (s : Finset ι) (t : ι → c.F) (f : ι → Rul Xo c) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : evR Xo c (∑ i ∈ s, t i • f i) y = ∑ i ∈ s, algebraMap c.F S (t i) * evR Xo c (f i) y

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EvaluationLemmaForNablaLevels

theorem **eq_of_geometric_points** {R : Type} [CommRing R] [_root_.IsReduced R] (a b : R) (h : ∀ (k : Type) [Field k] [IsAlgClosed k] (χ : R →+* k), χ a = χ b) : a = b

theorem **isReduced_of_smooth_algebra** (R S : Type) [CommRing R] [CommRing S] [Algebra R S] [_root_.IsReduced R] [Algebra.Smooth R S] : _root_.IsReduced S

theorem **exists_lift_geometric** (R S : Type) [CommRing R] [CommRing S] [Algebra R S] [Module.Finite R S] [Module.FaithfullyFlat R S] (k : Type) [Field k] [IsAlgClosed k] (χ : R →+* k) : ∃ χ' : S →+* k, χ'.comp (algebraMap R S) = χ

theorem **exists_preimage_of_geometric** (R S : Type) [CommRing R] [CommRing S] [Algebra R S] [_root_.IsReduced R] [Algebra.Smooth R S] [Module.FaithfullyFlat R S] (w : S) (h : ∀ (k : Type) [Field k] [IsAlgClosed k] (χ₁ χ₂ : S →+* k), χ₁.comp (algebraMap R S) = χ₂.comp (algebraMap R S) → χ₁ w = χ₂ w) : ∃ u : R, algebraMap R S u = w

theorem **exists_affine_lift** {X Y : Scheme.{0}} (f : Y ⟶ X) [IsFinite f] [Etale f] [Surjective f] (R : Type) [CommRing R] (t : Sp R ⟶ X) : ∃ (S : Type) (_ : CommRing S) (φ : R →+* S) (t' : Sp S ⟶ Y), φ.Finite ∧ φ.Smooth ∧ φ.FaithfullyFlat ∧ t' ≫ f = spMap φ ≫ t

theorem **evR_of_test** (hWD : EvalWD (K

theorem **evR_baseChange** (hWD : EvalWD (K

theorem **exists_cover_point** {l₂ : Level H} {Xo₂ : Moduli Dw l₂} {c₂ : Coeff l₂} (hc : c.F = c₂.F) (r : Xo₂.X ⟶ Xo.X) (rF : Xo₂.XF c₂ ⟶ Xo.XF c) (h1 : rF ≫ pullback.fst _ _ = pullback.fst _ _ ≫ r) (h2 : rF ≫ pullback.snd _ _ = pullback.snd _ _ ≫ spMap (IntermediateField.inclusion hc.le).toRingHom) [IsFinite rF] [Etale rF] [Surjective rF] (R : Type) [CommRing R] [Algebra c.F R] (x : Sp R ⟶ Xo.X) (hx : x ≫ Xo.s = sOf c R) : ∃ (S : Type) (_ : CommRing S) (φ : R →+* S) (j : c₂.F →+* S) (x₂ : Sp S ⟶ Xo₂.X), φ.Finite ∧ φ.Smooth ∧ φ.FaithfullyFlat ∧ j.comp (IntermediateField.inclusion hc.le).toRingHom = φ.comp (algebraMap c.F R) ∧ x₂ ≫ r = spMap φ ≫ x ∧ x₂ ≫ Xo₂.s = spMap (j.comp (IntermediateField.inclusion c₂.le).toRingHom)

theorem **shortValue_descends** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), EvalWD (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.EvaluationLemmaForNablaLevelsStatement

theorem **_root_.GlobalTheory.T.evaluation_lemma_for_nabla_levels** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), EvalWD (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExAtMemPairChartIff

theorem **hermForm_pull** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (Φ : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (z w : Fin g → ℂ) : hermForm ((LinearMap.toMatrix' Φ).transpose * Hm * (LinearMap.toMatrix' Φ).transpose.conjTranspose) z w = hermForm Hm (Φ z) (Φ w)

theorem **exAt_mem_pairChart_iff** {m : T ⊗ A ⟶ A} {z w : Fin g → ℂ} (h : exAt u p z ≫ m.left = (u.ex w).left) (j : Fin 2 → L.ι) : ⊤ ≤ exAt u p z ⁻¹ᵁ LB.wchart L ![(snd T A).left, m.left] j ↔ ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U (j 0) ∧ ⊤ ≤ (u.ex w).left ⁻¹ᵁ L.U (j 1)

theorem **map_mem_period** {m : T ⊗ A ⟶ A} {Φ : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (hΦ : ∀ z, exAt u p z ≫ m.left = (u.ex (Φ z)).left) {l : Fin g → ℂ} (hl : l ∈ u.Λ) : Φ l ∈ u.Λ

theorem **exists_fn_of_isGen** (hL : L.IsLB) {m : T ⊗ A ⟶ A} {Φ : (Fin g → ℂ) → Fin g → ℂ} (hd : Differentiable ℂ Φ) (hΦ : ∀ z, exAt u p z ≫ m.left = (u.ex (Φ z)).left) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **exists_isPullConst** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {m : T ⊗ A ⟶ A} {Φ : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (hΦ : ∀ z, exAt u p z ≫ m.left = (u.ex (Φ z)).left) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **IsPullConst.fn_eq** (hL : L.IsLB) {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} {hm : ψ ≫ snd T A = m} {Φ : (Fin g → ℂ) → Fin g → ℂ} (hψ : ∀ z, exAt u p z ≫ ψ.left = exAt u p (Φ z)) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **IsPullConst.origin** {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) {m : T ⊗ A ⟶ A} {Φ : (Fin g → ℂ) → Fin g → ℂ} (h0 : Φ 0 = 0) (hΦ : exAt u p 0 ≫ m.left = (u.ex 0).left) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **IsPullConst.eq_one** (hL : L.IsLB) {rig : LB.Gen (L.pull (eSec A))} (hrig : u.IsRig rig) {m : T ⊗ A ⟶ A} {Φ : (Fin g → ℂ) → Fin g → ℂ} (h0 : Φ 0 = 0) (hΦ : exAt u p 0 ≫ m.left = (u.ex 0).left) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **IsPullConst.unique** (hL : L.IsLB) {m : T ⊗ A ⟶ A} {Φ : (Fin g → ℂ) → Fin g → ℂ} (hΦ : ∀ z, exAt u p z ≫ m.left = (u.ex (Φ z)).left) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **exAt_snd_comp** {f : A ⟶ A} {z w : Fin g → ℂ} (h : u.ex w = u.ex z ≫ f) : exAt u p z ≫ (snd T A ≫ f).left = (u.ex w).left

theorem **ex_natCast_smul** (k : ℕ) (z : Fin g → ℂ) : u.ex ((k : ℂ) • z) = u.ex z ^ k

theorem **exAt_mulT** (k : ℕ) (z : Fin g → ℂ) : exAt u p z ≫ (mulT A T k).left = exAt u p ((k : ℂ) • z)

theorem **pull_analytic_at** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} (hm : ψ ≫ snd T A = m) {Φ : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (hψ : ∀ z, exAt u p z ≫ ψ.left = exAt u p (Φ z)) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAdjReadingOne

private theorem **padd_lemma** {T : Over S} (a b : T ⟶ A) : padd A a b = a ∗ b

private theorem **pzero_eq_one_cusp_form_evaluations** {T : Over S} : pzero A T = 1

theorem **comp_cub** {T T' : Over S} {n : ℕ} (h : T' ⟶ T) (xs : Fin n → (T ⟶ A)) (I : ULift.{0} (Finset (Fin n))) : h.left ≫ cub xs I = cub (fun k => h ≫ xs k) I

theorem **psum_one** {T : Over S} {n : ℕ} (I : Finset (Fin n)) : psum A (fun _ : Fin n => (1 : T ⟶ A)) I = 1

theorem **comp_adφ** {T : Over S} (f f' : A ⟶ A) (x y : T ⟶ A) (k) : (lift x y).left ≫ adφ f f' k = Sum.elim (cub ![x, y ≫ f']) (cub ![x ≫ f, y]) k

theorem **one_adφ** {T : Over S} (f f' : A ⟶ A) [IsMonHom f] [IsMonHom f'] (k) : (lift (1 : T ⟶ A) (1 : T ⟶ A)).left ≫ adφ f f' k = (1 : T ⟶ A).left

theorem **prod_adε** {U : Type} [CommGroup U] (x : U) : ∏ k, x ^ adε k = 1

theorem **exists_norm** {L : LB A.left} (hL : L.IsLB) (f f' : A ⟶ A) [IsMonHom f] [IsMonHom f'] (H : LBIso (cubeLB A ![fst A A ≫ f, snd A A] L) (cubeLB A ![fst A A, snd A A ≫ f'] L)) : ∃ u : LB.Gen (LB.LinComb L (adφ f f') adε), u.IsGen ∧ LB.Gen.Corr L _ _ _ _ (LB.pullGen L _ _ u (lift (1 : A ⊗ A ⟶ A) (1 : A ⊗ A ⟶ A)).left) (LB.oneGen L)

theorem **e_zero** : Φ.e 0 = 1

theorem **psumL_apply** {n : ℕ} (I : ULift.{0} (Finset (Fin n))) (p : Fin n → Fin g → ℂ) : psumL n g I p = ∑ i ∈ I.down, p i

theorem **cub_e** (p : Fin 2 → Fin g → ℂ) (I : ULift.{0} (Finset (Fin 2))) : cub (fun i => Φ.e (p i)) I = (Φ.e (psumL 2 g I p)).left

theorem **sum_herm** (Hm : Matrix (Fin g) (Fin g) ℂ) (p q : Fin 2 → Fin g → ℂ) : ∑ I : ULift.{0} (Finset (Fin 2)), ((cε 2 I : ℤ) : ℂ) ∗ hermForm Hm (psumL 2 g I p) (psumL 2 g I q) = hermForm Hm (p 0) (q 1) + hermForm Hm (p 1) (q 0)

theorem **hol_comp** {N : ℕ} (P : (Fin N → ℂ) →ₗ[ℂ] (Fin g → ℂ)) : IsHolFam (fun v => (Φ.e (P v)).left)

theorem **span_prod** {N : ℕ} (C : (Fin N → ℂ) ≃ₗ[ℂ] (Fin g → ℂ) × (Fin g → ℂ)) (lat : AddSubgroup (Fin g → ℂ)) (h : Submodule.span ℝ (lat : Set (Fin g → ℂ)) = ⊤) : Submodule.span ℝ (((lat.prod lat).comap C.toAddEquiv.toAddMonoidHom : AddSubgroup (Fin N → ℂ)) : Set (Fin N → ℂ)) = ⊤

private theorem **genRd_of_specComplex** {Z : Scheme.{0}} (q : Sp ℂ ⟶ Z) : Φ.GenRd (K

private theorem **sameSum_adφ_adε_empty** (f f' : A ⟶ A) [IsMonHom f] [IsMonHom f'] : LB.SameSum (fun k => (lift (1 : A ⊗ A ⟶ A) (1 : A ⊗ A ⟶ A)).left ≫ adφ f f' k) adε (PEmpty.elim : PEmpty → ((A ⊗ A).left ⟶ A.left)) PEmpty.elim

theorem **reading_one** (f f' : A ⟶ A) [IsMonHom f] [IsMonHom f'] (H : LBIso (cubeLB A ![fst A A ≫ f, snd A A] L.L) (cubeLB A ![fst A A, snd A A ≫ f'] L.L)) (α β : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (hα : ∀ z, Φ.e (α z) = Φ.e z ≫ f) (hβ : ∀ z, Φ.e (β z) = Φ.e z ≫ f') (h₁ : ∀ z w, hermForm Φ.Hm (α z) w = hermForm Φ.Hm z (β w)) (h₂ : ∀ z w, hermForm Φ.Hm (β z) w = hermForm Φ.Hm z (α w)) : ∃ u : LB.Gen (LB.LinComb L.L (adφ f f') adε), u.IsGen ∧ ∀ z w : Fin g → ℂ, Φ.GenRd _ _ u (lift (Φ.e z) (Φ.e w)).left (Sum.elim (fun I => psumL 2 g I ![z, β w]) (fun I => psumL 2 g I ![α z, w])) 1

theorem **exists_adj_reading_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ a : 𝒪 E, ∃ wL : LB.Gen (LB.LinComb Xo.univ.t.PP.L (Sum.elim (cφ ![fst Xo.univ.t.G Xo.univ.t.G, snd Xo.univ.t.G Xo.univ.t.G ≫ Xo.univ.t.act.act (H.bar a)]) (cφ ![fst Xo.univ.t.G Xo.univ.t.G ≫ Xo.univ.t.act.act a, snd Xo.univ.t.G Xo.univ.t.G])) (Sum.elim (cε 2) fun I => -cε 2 I)), wL.IsGen ∧ ∀ z w : Fin g → ℂ, Φ.toAnFr.GenRd _ _ wL (lift (Φ.e z) (Φ.e w)).left (Sum.elim (fun I => psumL 2 g I ![z, tyAct ty (H.bar a) w]) (fun I => psumL 2 g I ![tyAct ty a z, w])) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAdjointnessSeriesOfFrames

theorem **subst_copy** (m : ℕ) {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (i : Fin 2) : (jetSubstP m (fun p : Fin 2 × Fin d => φ p.1 p.2)).comp (jetCopy d m i) = jetSubstP m (fun p : Fin 1 × Fin d => φ i p.2)

theorem **substX** (m : ℕ) {n : ℕ} (k : Fin n) : jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (k, p.2) : MvPolynomial (Fin n × Fin d) S)) = jetCopy d m k

theorem **ren_copy** (m : ℕ) {n n' : ℕ} (r : Fin n → Fin n') (i : Fin n) : (jetRen (S

theorem **ren_one** (m : ℕ) {n' : ℕ} (r : Fin 1 → Fin n') : jetRen (S

theorem **subst_psIn** (m : ℕ) {n n' : ℕ} (r : Fin n → Fin n') (g : Fin d → MvPowerSeries (Fin n × Fin d) S) : jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (g p.2)) = (jetRen d m r).comp (jetSubstP m (fun p : Fin 1 × Fin d => psTrunc m (g p.2)))

theorem **jetSubstP_zero_eq** {S κ κ' : Type} [CommRing S] (φ ψ : κ → MvPolynomial κ' S) : (jetSubstP 0 φ : Jet S κ 0 →ₐ[S] Jet S κ' 0) = jetSubstP 0 ψ

theorem **pt_sub** (m : ℕ) {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (i : Fin 2) : homAt x0 (jetSubstP m (fun p : Fin 2 × Fin d => φ p.1 p.2)) ≫ jptG t x0 d x m i = homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => φ i p.2)) ≫ x m

theorem **pt_X** (m : ℕ) {n : ℕ} (k : Fin n) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (k, p.2) : MvPolynomial (Fin n × Fin d) S))) ≫ x m = jptG t x0 d x m k

theorem **pt_law** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (r : Fin 2 → Fin n) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (F p.2))) ≫ x m = jptG t x0 d x m (r 0) ∗ jptG t x0 d x m (r 1)

theorem **pt_act** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (r : Fin 1 → Fin n) (a : 𝒪 E) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (act a p.2))) ≫ x m = jptG t x0 d x m (r 0) ≫ t.act.act a

theorem **pe_pt** (m : ℕ) {n : ℕ} (σ : Jet S (Fin 2 × Fin d) m →ₐ[S] Jet S (Fin n × Fin d) m) (I : ULift.{0} (Finset (Fin 2))) : (homAt x0 σ).left ≫ cφG t ![jptG t x0 d x m (0 : Fin 2), jptG t x0 d x m (1 : Fin 2)] I = (psum t.G ![homAt x0 σ ≫ jptG t x0 d x m (0 : Fin 2),

homAt x0 σ ≫ jptG t x0 d x m (1 : Fin 2)] I.down).left

theorem **wl_pt** {T : Over X0} (a b : T ⟶ t.G) (c : 𝒪 E) (I : ULift.{0} (Finset (Fin 2))) : (lift a b).left ≫ cφG t ![fst t.G t.G, snd t.G t.G ≫ t.act.act c] I = (psum t.G ![a, b ≫ t.act.act c] I.down).left

theorem **wl_pt'** {T : Over X0} (a b : T ⟶ t.G) (c : 𝒪 E) (I : ULift.{0} (Finset (Fin 2))) : (lift a b).left ≫ cφG t ![fst t.G t.G ≫ t.act.act c, snd t.G t.G] I = (psum t.G ![a ≫ t.act.act c, b] I.down).left

theorem **frame_origin_jet_one** (hch : IsChartG t x0 d F act tg uvec x) : x 0 = 1

theorem **jpt_trunc'** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (i : Fin n) : homAt x0 (jetTrunc : Jet S (Fin n × Fin d) (m + 1) →ₐ[S] _) ≫ jptG t x0 d x (m + 1) i = jptG t x0 d x m i

theorem **pair_trunc** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) : (ρn x0 d 2 m).left ≫ (lift (jptG t x0 d x (m + 1) (0 : Fin 2)) (jptG t x0 d x (m + 1) (1 : Fin 2))).left = 𝟙 _ ≫ (lift (jptG t x0 d x m (0 : Fin 2)) (jptG t x0 d x m (1 : Fin 2))).left

theorem **exists_ad_one** (hch : IsChartG t x0 d F act tg uvec x) (e : ∀ m, FrTyG t x0 d x m) (heg : ∀ m, (e m).IsGen) (htr : FrTrunc t x0 d x e) (a : 𝒪 E) (wL : LB.Gen (LB.LinComb t.PP.L (Sum.elim (cφG t ![fst t.G t.G, snd t.G t.G ≫ t.act.act (H.bar a)]) (cφG t ![fst t.G t.G ≫ t.act.act a, snd t.G t.G])) (Sum.elim (cε 2) fun I => -cε 2 I))) (hwLg : wL.IsGen) (hwL0 : ∀ (Z : Scheme.{0}) (r : Z ⟶ (𝟙_ (Over X0)).left) (z : Z) (J : (Z ⟶ t.G.left) → t.PP.L.ι), Good t.PP.L z J → val wL (r ≫ (zz t).left) z J = 1) : ∃ ad : MvPowerSeries (Fin 2 × Fin d) S, MvPowerSeries.constantCoeff ad = 1 ∧ ∀ m, ∃ u : (Jet S (Fin 2 × Fin d) m)ˣ, (u : Jet S (Fin 2 × Fin d) m) = psJet m ad ∧ LB.Gen.Corr t.PP.L _ _ _ _ (LB.mulGen t.PP.L _ _ _ _ (LB.pullGen t.PP.L _ _ wL (lift (jptG t x0 d x m (0 : Fin 2)) (jptG t x0 d x m (1 : Fin 2))).left) (peG t x0 d x e m ![fun j => psIn d m ![0] (act a j), fun j => MvPolynomial.X

theorem **GlobalTheory.FrameAdjSeries.exists_ad** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ ad : 𝒪 E → PS S 2 d, (∀ a, MvPowerSeries.constantCoeff (ad a) = 1) ∧ ∀ a : 𝒪 E, ∃ wL : LB.Gen (LB.LinComb t.PP.L (Sum.elim (cφG t ![fst t.G t.G, snd t.G t.G ≫ t.act.act (H.bar a)]) (cφG t ![fst t.G t.G ≫ t.act.act a, snd t.G t.G])) (Sum.elim (cε 2) fun I =>

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAdjointnessSeriesOfFramesStatement

theorem **_root_.GlobalTheory.T.exists_adjointness_series_of_frames** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ ad : 𝒪 E → PS S 2 d, (∀ a, MvPowerSeries.constantCoeff (ad a) = 1) ∧ ∀ a : 𝒪 E, ∃ wL : LB.Gen (LB.LinComb t.PP.L (Sum.elim (cφG t ![fst t.G t.G, snd t.G t.G ≫ t.act.act (H.bar a)]) (cφG t ![fst t.G t.G ≫ t.act.act a, snd t.G t.G])) (Sum.e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAdmissibleThetaLatticeGenerators

theorem **tiedW_scaledFamily** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (S : Scal B) {N : ℕ} (w0 : Fin N → B.Cusp j wL) (a b G Prw : ℝ) (ha : 0 ≤ a) (hb : 0 ≤ b) (hs : ∀ σ : B.KO →+* ℂ, Real.log ‖σ (S.sW : B.KO)‖ ≤ a) (hspan : Submodule.span (𝒪 B.KO) (Set.range w0) = X.thAll wL) (hw : ∀ k σ, Real.log (B.supN j wL σ (w0 k)) ≤ b) (hG : a + b ≤ G * wL.size) (hM : X.WWc S.sW wL ≤ B.Mlat j wL) (hN : Real.log N ≤ Prw * (wL.size - 1)) : (scaledFamily X ka wL wF S w0 (a + b)).TiedW S G Prw

theorem **log_le_of_le_two_mul_pow** {N a b d m : ℕ} (hN0 : 0 < N) (hN : N ≤ 2 * (a ^ m * (d * b ^ m))) : Real.log N ≤ Real.log 2 + (m : ℝ) * (Real.log a + Real.log b) + Real.log d

private theorem **containsTh_of_eq_smul** {D : D7} {T : Top D} (j : ℕ) {n N : ℕ} (P₁ : MvPolynomial (LtN (T.Q.Dw j) n) T.Q.c.F) (s : T.Q.c.F) (u v : Fin N → T.R j) (huv : ∀ k p y, v k p y = algebraMap T.Q.c.F _ s * u k p y) (hc : ContainsTh (T.Q.Xo j) T.Q.c (T.nab j) 1 P₁ u) : ContainsTh (T.Q.Xo j) T.Q.c (T.nab j) s P₁ v

theorem **exists_genOK** : ∀ (D : D7) (T : Top D) (h : LawsP T) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), GlobalTheory.PtTie.Oriented T EX → PageOK large₂ (recPJ h j hj) H C2 A2 CT ka wL wF Tn nP nW h0 → ∃ G : GlobalFamilyData (recPJ h j hj) ka wL wF, GenOK h j hj large₂ G (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) (cc D large₂ C2 A2 7 wL)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAdmissibleThetaLatticeGeneratorsStatement

theorem **_root_.GlobalTheory.T.exists_admissible_thetaLatticeGenerators** : ∀ (D : D7) (T : Top D) (h : LawsP T) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), GlobalTheory.PtTie.Oriented T EX → PageOK large₂ (recPJ h j hj) H C2 A2 CT ka wL wF Tn nP nW h0 → ∃ G : GlobalFamilyData (recPJ h j hj) ka wL wF, GenOK h j hj large₂ G (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) (cc D large₂ C2 A2 7 wL)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAlgIndependentCoeffsFlatDatum

theorem **ae_cast** {a b n m : ℕ} (hab : a = b) (hnm : n = m) (P : (Fin (a + 1) × Fin n → ℂ) → Prop) (hP : ∀ᵐ t : Fin (a + 1) × Fin n → ℂ, P t) : ∀ᵐ t : Fin (b + 1) × Fin m → ℂ, P fun p => t (Fin.cast (congrArg (· + 1) hab) p.1, Fin.cast hnm p.2)

theorem **exists_generic_flatAt** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1), ∃ t : Idx D j N → ℂ, (∀ Q : MvPolynomial (Idx D j N) T.Q.c.F, Q ≠ 0 → MvPolynomial.eval₂ T.σ₀ t Q ≠ 0) ∧ Nonempty (FlatAt h j A t)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAlgIndependentCoeffsFlatDatumStatement

theorem **_root_.GlobalTheory.T.exists_algIndependent_coeffs_flatDatum** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1), ∃ t : Idx D j N → ℂ, (∀ Q : MvPolynomial (Idx D j N) T.Q.c.F, Q ≠ 0 → MvPolynomial.eval₂ T.σ₀ t Q ≠ 0) ∧ Nonempty (FlatAt h j A t)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticBiextensionChartComplexPoint

theorem **isFrame_of_isAnE** (hst : ∀ a : 𝒪 E, star a = H.bar a) (e : ∀ m, FrTy y g Φ.jx m) (he : Φ.IsAnE e) : IsFrame y g Φ.jx (anFB ty).G.F (anFB ty).A.act (anFB ty).ph1 (anFB ty).ph2 (anFB ty).sy (anFB ty).ad e

theorem **isGerm_tay_of_rd** (e : ∀ m, FrTy y g Φ.jx m) (he : Φ.IsAnE e) (wP : PrimTy H Xo) (hw : IsPrim wP) (k : ℕ) (F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) (θ : (Fin g → ℂ) → ℂ) (hF : Φ.toAnFr.Rd k F θ) : IsGerm y g Φ.jx (anFB ty).A.act e wP k F (tay θ 0)

theorem **rename_cast_refl** {g : ℕ} (f : MvPowerSeries (Fin 1 × Fin g) ℂ) : MvPowerSeries.rename (fun p : Fin 1 × Fin g => (p.1, Fin.cast (rfl : g = g).symm p.2)) f = f

theorem **exists_analytic_chart** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), (∀ a : 𝒪 E, star a = H.bar a) → ∃ (C : ChartAt Xo c y) (hd : C.d = g), C.h = 0 ∧ (∀ i, C.B.G.F i = (MvPowerSeries.X (R

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticBiextensionChartComplexPointStatement

theorem **_root_.GlobalTheory.T.exists_analytic_biextension_chart_complex_point** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), (∀ a : 𝒪 E, star a = H.bar a) → ∃ (C : ChartAt Xo c y) (hd : C.d = g), C.h = 0 ∧ (∀ i, C.B.G.F i = (MvPowerSeries.X (R

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticDataUnitFrameFramedPoint

theorem **hostHalfRoot_add_conj_eq_one** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : (H.w0 : E) + cc (H.w0 : E) = 1

theorem **exists_setup** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r), GoodEmb c σ → IsUnf U → 1 ≤ r → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → ∀ k (x : (U.X k).dom), IsUnitFr (U.X k) x.1 → letI : Algebra c.F ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticDataUnitFrameFramedPointStatement

theorem **_root_.GlobalTheory.T.exists_analyticData_unitFrame_framedPoint** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r), GoodEmb c σ → IsUnf U → 1 ≤ r → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → ∀ k (x : (U.X k).dom), IsUnitFr (U.X k) x.1 → letI : Algebra c.F ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticFrameJetPairIsAnE

theorem **jetMap_psum** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} [Algebra c.F ℂ] {y : FrPt Xo c ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} {el : Fin g ≃ Lt1 Dw} (Φ : AnPt y g ty el) (n m : ℕ) (xs : Fin n → (jO y g m n ⟶ Xo.univ.t.G)) (Λ : Fin n → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (hx : ∀ i, Φ.toAnFr.JetMap m (xs i).left (Λ i)) (I : Finset (Fin n)) : Φ.toAnFr.JetMap m (psum Xo.univ.t.G xs I).left (∑ i ∈ I, Λ i)

theorem **exists_isAnE** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∃ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticFrameJetPairIsAnEStatement

theorem **_root_.GlobalTheory.T.exists_analyticFrame_jetPair_isAnE** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∃ e : ∀ m, FrTy y g Φ.jx m, Φ.IsAnE e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticFrameOfUnitUniformisationFrame

theorem **exists_anPt_of_unitAt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (W : UnfAt U k x) (Wu : UnitAt W), letI : Algebra c.F ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAnalyticFrameOfUnitUniformisationFrameStatement

theorem **_root_.GlobalTheory.T.exists_analytic_frame_of_unit_uniformisation_frame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (W : UnfAt U k x) (Wu : UnitAt W), letI : Algebra c.F ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsAuxiliaryWeightsAtRunWeights

theorem **antitone_nonneg_of_gaps** {p : ℕ} (g : Fin p → ℤ) (hg : ∀ (i : Fin p) (h : i.1 + 1 < p), g (i.1 + 1, h) ≤ g i) (hl : ∀ i : Fin p, i.1 + 1 = p → 0 ≤ g i) : Antitone g ∧ ∀ i, 0 ≤ g i

theorem **gap_key** (c cL T' r : ℤ) (hT : 0 < T') (hr : 0 ≤ r) (hc : T' * (T' + 2 * r) ≤ c + 2) (hL : cL = Int.fdiv (c + 2) T') : c ≤ (T' + 2) * cL

theorem **exists_block** {p : ℕ} (a l f : Fin p → ℕ) (T Dm r T' : ℕ) (hT' : T' = T + Dm + 2) (hf : ∀ i, (f i : ℤ) = a i − T * l i) (hgap : ∀ i : Fin p, i.1 + 1 < p → ((T' * (T' + 2 * r) : ℕ) : ℤ) ≤ gapZ a i + 2) (hL : ∀ i : Fin p, i.1 + 1 < p → gapZ l i = Int.fdiv (gapZ a i + 2) T') (hlast : ∀ i : Fin p, i.1 + 1 = p → (a i : ℤ) ≤ ((T : ℤ) + Dm + 4) * l i) : ∃ w : Fin p → ℕ, Antitone w ∧ ∀ i, w i + f i = (Dm + 4) * l i

private theorem **le_mul_fdiv_sub_of_sq_le** (a c T Dm : ℤ) (hD : 0 ≤ Dm) (h40 : 40 * (Dm + 2) ≤ T) (hc : c ≤ T) (ha : 2 * T ^ 2 ≤ a) : a ≤ (T + Dm + 4) * Int.fdiv (a − c) (T + Dm + 2)

theorem **exists_auxiliaryWeights_at_runWeights** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → (∃ w2 : WeightFamilies.Wt D j, IsG2 wL wF w2) ∧ ∃ w1 : WeightFamilies.Wt D j, IsDoubledWeightWithShift wL w1 nW

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBaseEmbeddingReadingForms

theorem **sumParts_single** (R M : Type) [CommRing R] [AddCommMonoid M] (m : M) (f : R) : sumParts R M (AddMonoidAlgebra.single m f) = f

theorem **sumParts_smul** (K R M : Type) [Field K] [CommRing R] [Algebra K R] [AddCommMonoid M] (a : K) (x : AddMonoidAlgebra R M) : sumParts R M (a • x) = a • sumParts R M x

theorem **exists_isRdgU** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)), ∃ (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBaseEmbeddingReadingFormsStatement

theorem **_root_.GlobalTheory.T.exists_base_embedding_reading_forms** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)), ∃ (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBasisSpanEqOfCommensurable

theorem **exists_basis_span_eq_of_commensurable** {V κ : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [Fintype κ] (b : Module.Basis κ ℝ V) (L : Submodule ℤ V) {M : ℕ} (hM : 1 ≤ M) (h1 : ∀ x ∈ Submodule.span ℤ (Set.range b), M • x ∈ L) (h2 : ∀ x ∈ L, M • x ∈ Submodule.span ℤ (Set.range b)) : ∃ b' : Module.Basis κ ℝ V, Submodule.span ℤ (Set.range b') = L

theorem **setAverage_comp_equiv** {V W F : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] [NormedAddCommGroup W] [NormedSpace ℝ W] [FiniteDimensional ℝ W] [MeasurableSpace W] [BorelSpace W] [NormedAddCommGroup F] [NormedSpace ℝ F] (μ : Measure V) [μ.IsAddHaarMeasure] (ν : Measure W) [ν.IsAddHaarMeasure] (e : V ≃L[ℝ] W) (s : Set W) (f : W → F) : ∱ w in s, f w ∂ν = ∱ v in e ⁻¹' s, f (e v) ∂μ

theorem **translate_comp_flat_symm** (X : PT E) (U : ℕ → EuclideanSpace ℂ (Fin n × Idx X.dim) → (EuclideanSpace ℂ (Fin n × Idx X.dim) → ℂ) → EuclideanSpace ℂ (Fin n × Idx X.dim) → ℂ) (hU : ∀ (q : ℕ) a F v, U q a F v = cexp (q * Real.pi * Hf v a - q * Real.pi * Hf a a / 2) * F (v - a)) (q : ℕ) (a : Fin n → Vsp X.dim) (F : (Fin n → Vsp X.dim) → ℂ) : U q (flat a) (F ∘ flat.symm) = X.U q a F ∘ flat.symm

theorem **polyDeriv_comp_flat_symm** (P : MvPolynomial (Letter n dim) ℂ) (F : (Fin n → Vsp dim) → ℂ) (v : Fin n → Vsp dim) : polyDeriv P (fun l => EuclideanSpace.single l 1) (F ∘ flat.symm) (flat v) = polyDeriv P dir F v

theorem **pointwiseNorm_comp_flatten_symm_eq** (X : PT E) (Y : GT (Fin n × Idx X.dim)) (q : ℕ) (F : (Fin n → Vsp X.dim) → ℂ) (v : Fin n → Vsp X.dim) : Y.ptN2 q (F ∘ flat.symm) (flat v) = X.ptN2 q F v

theorem **l2_comp_flat_symm_eq_l2** (X : PT E) (Y : GT (Fin n × Idx X.dim)) (hY : ∀ γ, flat γ ∈ lat Y.b ↔ ∀ k, γ k ∈ X.L) (q : ℕ) {F : (Fin n → Vsp X.dim) → ℂ} (hF : F ∈ X.Sec q n) : Y.l2 q (F ∘ flat.symm) = X.l2 q F

theorem **comp_flat_symm_mem_sec_iff** (X : PT E) (Y : GT (Fin n × Idx X.dim)) (hY : ∀ γ, flat γ ∈ lat Y.b ↔ ∀ k, γ k ∈ X.L) (q : ℕ) (hka : ∀ γ : Fin n → Vsp X.dim, (∀ k, γ k ∈ X.L) → Y.ka (flat γ) = (∏ k, (X.eps (γ k) : ℂ)) ^ q) (F : (Fin n → Vsp X.dim) → ℂ) : F ∘ flat.symm ∈ Y.Sec q ↔ F ∈ X.Sec q n

theorem **im_HC_act_add** (w₀ : 𝒪 E) (hw : (w₀ : E) + cc (w₀ : E) = 1) (γ δ : Fin n → Vsp dim) : (HC (γ + δ) fun k => act w₀ ((γ + δ) k)).im = (HC γ fun k => act w₀ (γ k)).im + (HC δ fun k => act w₀ (δ k)).im + (2 * (HC γ fun k => act w₀ (δ k)).im - (HC γ δ).im)

theorem **tensorEps_sq_of_mem** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (w₀ : 𝒪 E) {γ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') : tensorEps w₀ γ ^ 2 = 1

theorem **tensorEps_add_of_mem** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (w₀ : 𝒪 E) (hw : (w₀ : E) + cc (w₀ : E) = 1) {γ δ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') (hδ : δ ∈ tensorLat X Λ') : tensorEps w₀ (γ + δ) = tensorEps w₀ γ * tensorEps w₀ δ * cexp (Real.pi * I * (HC γ δ).im)

theorem **exists_basis_tensorLat** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') : ∃ b : Module.Basis (Fin (2 * Fintype.card (Fin n × Idx X.dim))) ℝ (EuclideanSpace ℂ (Fin n × Idx X.dim)), ∀ γ, flat γ ∈ lat b ↔ γ ∈ tensorLat X Λ'

theorem **exists_tensorTorus** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (w₀ : 𝒪 E) : ∃ Y : GT (Fin n × Idx X.dim), (∀ γ, flat γ ∈ lat Y.b ↔ γ ∈ tensorLat X Λ') ∧ ∀ γ, Y.ka (flat γ) = tensorEps w₀ γ

theorem **principal_torus_of_selfDual** (X : PT E) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (w₀ : 𝒪 E) (hw : (w₀ : E) + cc (w₀ : E) = 1) : (∃ Y : GT (Fin n × Idx X.dim), (∀ γ, flat γ ∈ lat Y.b ↔ γ ∈ tensorLat X Λ') ∧ ∀ γ, Y.ka (flat γ) = tensorEps w₀ γ) ∧ (∀ Y : GT (Fin n × Idx X.dim), (∀ γ, flat γ ∈ lat Y.b ↔ γ ∈ tensorLat X Λ') → (∀ γ, Y.ka (flat γ) = tensorEps w₀ γ) → (∀ x ∈ lat Y.b, Y.ka x ^ 2 = 1) ∧ ∀ x ∈ lat Y.b, ∀ y ∈ lat Y.b, Y.ka (x + y) = Y.ka x * Y.ka y * cexp (Real.pi * I * (Hf x y).im)) ∧ ((∀ x ∈ X.L, (X.eps x : ℝ) = Real.cos (Real.pi * (Hf x (act w₀ x)).im)) → ∀ Y : GT (Fin n × Idx X.dim), (∀ γ, flat γ ∈ lat Y.b ↔ γ ∈ tensorLat X (stdLat E n)) → (∀ γ, Y.ka (flat γ) = tensorEps w₀ γ) → (∀ F, F ∘ flat.symm ∈ Y.Sec 1 ↔ F ∈ X.Sec 1 n) ∧ ∀ F ∈ X.Sec 1 n, F ∘ f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBiextensionChartWithGermsFramedPoint

theorem **bijective_appTop_powObj** (hGF : GlobalFnsSt) {S : Scheme.{0}} (G : Over S) [GrpObj G] (hG : IsAbelianScheme G) (n : ℕ) : Function.Bijective (ReciprocityTheory.ChartRestriction.G.Out.powObj G n).hom.appTop.hom

theorem **exists_chartAt_with_germs** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)], (∀ a : 𝒪 E, star a = H.bar a) → ∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra ℚ S] (y : FrPt Xo c S), ∃ C : ChartAt Xo c y, FrTruncC y C ∧ ∀ (k : ℕ) (F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))), ∃ f, IsGerm y C.d C.x C.B.A.act C.e C.wP k F f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBiextensionChartWithGermsFramedPointStatement

theorem **_root_.GlobalTheory.T.exists_biextension_chart_with_germs_framed_point** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)], (∀ a : 𝒪 E, star a = H.bar a) → ∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra ℚ S] (y : FrPt Xo c S), ∃ C : ChartAt Xo c y, FrTruncC y C ∧ ∀ (k : ℕ) (F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))), ∃ f, IsGerm y C.d C.x C.B.A.act C.e C.wP k F f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBlockProductMemIntegralHarmonic

theorem **minorMonomial_map** {R R' : Type} [CommRing R] [CommRing R'] (φ : R →+* R') (a : T → ℕ → ℕ) : MvPolynomial.map φ (minorsP r n col R a) = minorsP r n col R' a

theorem **minorMonomial_coeff_mem** (a : T → ℕ → ℕ) (O : Subring k) (m : Letter T r n →₀ ℕ) : (minorsP r n col k a).coeff m ∈ O

theorem **minor_ne_zero** (hcol : ∀ σ s, Function.Injective (col σ s)) (σ : T) (s : Fin (r σ)) : minor r n col k σ s ≠ 0

theorem **minorMonomial_ne_zero** (hcol : ∀ σ s, Function.Injective (col σ s)) (a : T → ℕ → ℕ) : minorsP r n col k a ≠ 0

theorem **minorMonomial_mem_harmonic** (cc : T → T) (isX : T → Prop) (hdis : ∀ τ, isX τ → ∀ s s' i j, col τ s i ≠ col (cc τ) s' j) (a : T → ℕ → ℕ) : minorsP r n col k a ∈ harmK r n k cc isX (fun _ => 1)

theorem **rowShift_minor** (σ : T) (i i' : Fin (r σ)) (hi : i' < i) (t : k) (σ' : T) (s : Fin (r σ')) : rowShift r n k σ i i' t (minor r n col k σ' s) = minor r n col k σ' s

theorem **minorMonomial_rowInvariant** (a : T → ℕ → ℕ) : RowInv r n k (minorsP r n col k a)

theorem **minor_weighted** (w : (Σ σ, Fin (r σ)) → ℕ) (σ : T) (s : Fin (r σ)) : (minor r n col k σ s).IsWeightedHomogeneous (fun ℓ : Letter T r n => w (ℓ.1, ℓ.2.1)) (∑ j : Fin (s + 1), w (σ, Fin.castLE s.isLt j))

theorem **minorMonomial_rowDegree** (a : T → ℕ → ℕ) (ha : ∀ σ, Antitone (a σ)) (h0 : ∀ σ, a σ (r σ) = 0) : RowDeg r n k a (minorsP r n col k a)

theorem **minorMonomial_isHomogeneous** (a : T → ℕ → ℕ) : (minorsP r n col k a).IsHomogeneous (∑ σ, ∑ s : Fin (r σ), (s + 1) * (a σ s - a σ (s + 1)))

theorem **minors_polynomial_coefficient_norm_le_pow** (a : T → ℕ → ℕ) (N : ℕ) (hN : ∀ σ, r σ ≤ N) : MvPoly.l1Norm (minorsP r n col ℂ a) ≤ (N : ℝ) ^ (∑ σ, ∑ s : Fin (r σ), (s + 1) * (a σ s - a σ (s + 1)))

theorem **exists_cols** (h : ∑ σ, r σ * (r σ + 1) ≤ 2 * n) : ∃ e : (Σ x : (Σ σ, Fin (r σ)), Fin (x.2 + 1)) → Fin n, Function.Injective e

theorem **inPsi_conjugate_of_not** {τ : E →+* ℂ} (h : ¬ InPsi H τ) : InPsi H (ComplexEmbedding.conjugate τ)

theorem **sum_mult_le** (Dw : Datum H) : ∑ τ : E →+* ℂ, Dw.mult τ * (Dw.mult τ + 1) ≤ I2.PT.f E * (Dw.p + Dw.q) * (Dw.p + Dw.q + 1)

theorem **isBlockProd_minorsP** (Dw : Datum H) {n : ℕ} (K : Type) [CommRing K] (e : (Σ x : (Σ σ, Fin (Dw.mult σ)), Fin (x.2 + 1)) → Fin n) (he : Function.Injective e) (a : (E →+* ℂ) → ℕ → ℕ) : IsBlockProd (Dw

theorem **exists_blockProd_mem_harmA** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), (∀ τ i k, i ≤ k → a (τ, k) ≤ a (τ, i)) → I2.PT.f E * (Dw.p + Dw.q) * (Dw.p + Dw.q + 1) ≤ 2 * n → ∃ P : MvPolynomial (LtN Dw n) c.F, P ∈ harmA c a n ∧ IsBlockProd P ∧ P.totalDegree = Formalism.L2.Pj.DF Dw a ∧ ∀ σ : c.F →+* ℂ, ∑ m ∈ P.support, ‖σ (P.coeff m)‖ ≤ ((Dw.p + Dw.q : ℕ) : ℝ) ^ Formalism.L2.Pj.DF Dw a

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsBlockProductMemIntegralHarmonicStatement

theorem **_root_.GlobalTheory.T.exists_blockProduct_mem_integralHarmonic** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), (∀ τ i k, i ≤ k → a (τ, k) ≤ a (τ, i)) → I2.PT.f E * (Dw.p + Dw.q) * (Dw.p + Dw.q + 1) ≤ 2 * n → ∃ P : MvPolynomial (LtN Dw n) c.F, P ∈ harmA c a n ∧ IsBlockProd P ∧ P.totalDegree = Formalism.L2.Pj.DF Dw a ∧ ∀ σ : c.F →+* ℂ, ∑ m ∈ P.support, ‖σ (P.coeff m)‖ ≤ ((Dw.p + Dw.q : ℕ) : ℝ) ^ Formalism.L2.Pj.DF Dw a

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsClearingIntegersPrincipalDatumHost

theorem **absNorm_pos** (D : D7) : 0 < Ideal.absNorm D.host.ccK

theorem **d_pos** (D : D7) : 0 < D.host.d

theorem **sqrt_ne_zero** (D : D7) : D.host.sqrtMinusD ≠ 0

theorem **norm_embedding_hostFormConstant_eq** (D : D7) (τ : D.host.E →+* ℂ) : ‖τ D.host.xi0‖ = 1 / ((Ideal.absNorm D.host.ccK : ℝ) * Real.sqrt D.host.d)

theorem **different_le** (D : D7) : differentIdeal ℤ (𝒪 D.host.E) ≤ D.host.𝔟

theorem **isIntegral_mul_of_mem** (D : D7) {z : 𝒪 D.host.E} (hz : z ∈ D.host.𝔟) {a : D.host.E} (ha : a ∈ D.host.AA) : IsIntegral ℤ ((z : D.host.E) * a)

theorem **absNorm_mem_AA** (D : D7) : ((Ideal.absNorm D.host.ccK : ℕ) : D.host.E) ∈ D.host.AA

theorem **host_clear** : ∀ (D : D7) (L : ℝ), 2 ≤ L → lrd D ≤ 2 * L → Real.log D.host.d ≤ 2 * L → Real.log (Ideal.absNorm D.host.ccK) ≤ 2 * L → (∃ NA : ℕ, 0 < NA ∧ (∀ a ∈ D.host.AA, IsIntegral ℤ ((NA : D.host.E) * a)) ∧ Real.log NA ≤ 4 * L) ∧ ((Ideal.absNorm D.host.ccK : ℕ) : D.host.E) ∈ D.host.AA ∧ 0 < Ideal.absNorm D.host.ccK ∧ (∃ m : ℕ, 0 < m ∧ IsIntegral ℤ ((m : D.host.E) * (2 * D.host.xi0)) ∧ Real.log m ≤ 4 * L) ∧ Real.log (1 / (2 * xiR D)) ≤ 3 * L ∧ fdeg D.host.E = D.R.f ∧ Real.log (dE D.host.E) ≤ 4 * fdeg D.host.E * L ∧ Real.log (rdE D.host.E) ≤ 2 * L ∧ D.R.f ≤ D.R.r ∧ 1 ≤ D.R.f ∧ ReciprocityTheory.ChartRestriction.G.FD.cc D.host.xi0 = -D.host.xi0 ∧ D.host.xi0 ≠ 0 ∧ (∀ τ : D.host.E →+* ℂ, ‖τ D.host.xi0‖ = xiR D)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsClearingIntegersPrincipalDatumHostStatement

theorem **_root_.GlobalTheory.T.exists_clearing_integers_principalDatum_host** : ∀ (D : D7) (L : ℝ), 2 ≤ L → lrd D ≤ 2 * L → Real.log D.host.d ≤ 2 * L → Real.log (Ideal.absNorm D.host.ccK) ≤ 2 * L → (∃ NA : ℕ, 0 < NA ∧ (∀ a ∈ D.host.AA, IsIntegral ℤ ((NA : D.host.E) * a)) ∧ Real.log NA ≤ 4 * L) ∧ ((Ideal.absNorm D.host.ccK : ℕ) : D.host.E) ∈ D.host.AA ∧ 0 < Ideal.absNorm D.host.ccK ∧ (∃ m : ℕ, 0 < m ∧ IsIntegral ℤ ((m : D.host.E) * (2 * D.host.xi0)) ∧ Real.log m ≤ 4 * L) ∧ Real.log (1 / (2 * xiR D)) ≤ 3 * L ∧ fdeg D.host.E = D.R.f ∧ Real.log (dE D.host.E) ≤ 4 * fdeg D.host.E * L ∧ Real.log (rdE D.host.E) ≤ 2 * L ∧ D.R.f ≤ D.R.r ∧ 1 ≤ D.R.f ∧ ReciprocityTheory.ChartRestriction.G.FD.cc D.host.xi0 = -D.host.xi0 ∧ D.host.xi0 ≠ 0 ∧ (∀ τ : D.host.E →+* ℂ, ‖τ D.host.xi0‖ = xiR D)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsCoeEqSpanDetOfIntVecLe

theorem **relIndex_eq_of_hom** {M M' : Type*} [AddCommGroup M] [AddCommGroup M'] {A₁ A₂ : AddSubgroup M} {B₁ B₂ : AddSubgroup M'} (f : M →+ M') (hf : ∀ v ∈ A₂, f v ∈ B₂) (hker : ∀ v ∈ A₂, f v ∈ B₁ ↔ v ∈ A₁) (hsurj : ∀ u ∈ B₂, ∃ v ∈ A₂, u - f v ∈ B₁) : A₁.relIndex A₂ = B₁.relIndex B₂

theorem **relIndex_pi_univ** {ι : Type*} [Fintype ι] {G : ι → Type*} [∀ i, AddCommGroup (G i)] (L' L : ∀ i, AddSubgroup (G i)) : (AddSubgroup.pi Set.univ L').relIndex (AddSubgroup.pi Set.univ L) = ∏ i, (L' i).relIndex (L i)

theorem **card_eq_relIndex_of_reps** {M : Type*} [AddCommGroup M] {A B : AddSubgroup M} {R : Finset M} (hR : ∀ y ∈ R, y ∈ B) (hR' : ∀ y ∈ B, ∃! r ∈ R, y - r ∈ A) : R.card = A.relIndex B

theorem **exists_reps_of_relIndex_ne_zero** {M : Type*} [AddCommGroup M] {A B : AddSubgroup M} (h : A.relIndex B ≠ 0) : ∃ R : Finset M, (∀ y ∈ R, y ∈ B) ∧ ∀ y ∈ B, ∃! r ∈ R, y - r ∈ A

theorem **exists_coe_eq_span_det_of_intVec_le** (X : Submodule (𝒪 K) (Fin n → K)) (hfg : X.FG) (hstd : intVec K n ≤ X) : ∃ J : FractionalIdeal (𝒪 K)⁰ K, (J : Submodule (𝒪 K) K) = Submodule.span (𝒪 K) {d | ∃ v : Fin n → Fin n → K, (∀ i, v i ∈ X) ∧ d = (Matrix.of v).det} ∧ FractionalIdeal.absNorm J * ((intVec K n).toAddSubgroup.relIndex X.toAddSubgroup : ℚ) = 1

theorem **exists_coe_eq_span_det** {Λ Λ' : Submodule (𝒪 K) (Fin n → K)} (hle : Λ ≤ Λ') (hfg : Λ'.FG) (hsp : Submodule.span K (Λ : Set (Fin n → K)) = ⊤) : ∃ J J' : FractionalIdeal (𝒪 K)⁰ K, J ≠ 0 ∧ J' ≠ 0 ∧ (J : Submodule (𝒪 K) K) = Submodule.span (𝒪 K) {d | ∃ v : Fin n → Fin n → K, (∀ i, v i ∈ Λ) ∧ d = (Matrix.of v).det} ∧ (J' : Submodule (𝒪 K) K) = Submodule.span (𝒪 K) {d | ∃ v : Fin n → Fin n → K, (∀ i, v i ∈ Λ') ∧ d = (Matrix.of v).det} ∧ FractionalIdeal.absNorm J = (Λ.toAddSubgroup.relIndex Λ'.toAddSubgroup : ℚ) * FractionalIdeal.absNorm J'

theorem **smul_coe_fractionalIdeal** (𝔫 : Ideal (𝒪 K)) (J : FractionalIdeal (𝒪 K)⁰ K) : 𝔫 • (J : Submodule (𝒪 K) K) = (((𝔫 : FractionalIdeal (𝒪 K)⁰ K) * J : FractionalIdeal (𝒪 K)⁰ K) : Submodule (𝒪 K) K)

theorem **exists_indexIdeal_span_det** {Λ Λ' : Submodule (𝒪 K) (Fin n → K)} (hle : Λ ≤ Λ') (hfg : Λ'.FG) (hsp : Submodule.span K (Λ : Set (Fin n → K)) = ⊤) : ∃ n : Ideal (𝒪 K), n ≠ ⊥ ∧ Ideal.absNorm n = Λ.toAddSubgroup.relIndex Λ'.toAddSubgroup ∧ ∀ m : Ideal (𝒪 K), Submodule.span (𝒪 K) {d | ∃ v : Fin n → Fin n → K, (∀ i, v i ∈ Λ) ∧ d = (Matrix.of v).det} = m • Submodule.span (𝒪 K) {d | ∃ v : Fin n → Fin n → K, (∀ i, v i ∈ Λ') ∧ d = (Matrix.of v).det} ↔ m = n

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsCoordsNoHomogeneousRelationGenericHyperpla

theorem **aeval_mul_of_isHomogeneous_cone** {K L ι : Type} [CommRing K] [CommRing L] [Algebra K L] {Q : MvPolynomial ι K} {d : ℕ} (hQ : Q.IsHomogeneous d) (c : L) (x : ι → L) : aeval (fun a => c * x a) Q = c ^ d * aeval x Q

theorem **aeval_eq_pow_mul_aeval_dehomogenise** {K L : Type} [CommRing K] [Field L] [Algebra K L] {m : ℕ} {Q : MvPolynomial (Fin (m + 1)) K} {d : ℕ} (hQ : Q.IsHomogeneous d) (x : Fin (m + 1) → L) (hx : x 0 ≠ 0) : aeval x Q = x 0 ^ d * aeval (fun a : Fin m => x a.succ / x 0) (aeval (Fin.cons 1 X : Fin (m + 1) → MvPolynomial (Fin m) K) Q)

theorem **dehomogenise_ne_zero_of_isHomogeneous** {K : Type} [Field K] {m : ℕ} {Q : MvPolynomial (Fin (m + 1)) K} {d : ℕ} (hQ : Q.IsHomogeneous d) (h0 : Q ≠ 0) : aeval (Fin.cons 1 X : Fin (m + 1) → MvPolynomial (Fin m) K) Q ≠ 0

theorem **exists_coordinates_no_homogeneous_relation_of_generic_zero** : ∀ (K : Type) [Field K] (σ : K →+* ℂ) (m N : ℕ) (t : Fin m × Fin N → ℂ) (ζ : Fin N → ℂ), (∀ Q : MvPolynomial (Fin m × Fin N) K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0) → ζ ≠ 0 → (∀ i, ∑ j, t (i, j) * ζ j = 0) → ∃ jj : Fin (m + 1) → Fin N, ∀ (Q : MvPolynomial (Fin (m + 1)) K) (d : ℕ), Q ≠ 0 → Q.IsHomogeneous d → MvPolynomial.eval₂ σ (fun a => ζ (jj a)) Q ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsCoordsNoHomogeneousRelationGenericHyperplaStatement

theorem **_root_.GlobalTheory.T.exists_coords_no_homogeneous_relation_generic_hyperplanes** : ∀ (K : Type) [Field K] (σ : K →+* ℂ) (m N : ℕ) (t : Fin m × Fin N → ℂ) (ζ : Fin N → ℂ), (∀ Q : MvPolynomial (Fin m × Fin N) K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0) → ζ ≠ 0 → (∀ i, ∑ j, t (i, j) * ζ j = 0) → ∃ jj : Fin (m + 1) → Fin N, ∀ (Q : MvPolynomial (Fin (m + 1)) K) (d : ℕ), Q ≠ 0 → Q.IsHomogeneous d → MvPolynomial.eval₂ σ (fun a => ζ (jj a)) Q ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsEmbExt

theorem **_root_.GlobalTheory.FieldBC.exists_emb_ext** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H} {c c' : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l} (hle : c.F ≤ c'.F) (σ : c.F →+* ℂ) : ∃ σ' : c'.F →+* ℂ, σ'.comp (_root_.Formalism.L2.Pj.incF c c' hle) = σ

theorem **_root_.GlobalTheory.FieldBC.rulBC_cusp** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum H) (l : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H) (Xo : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Moduli Dw l) (c c' : _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l) (hle : c.F ≤ c'.F) (r : ℕ) (U : ∀ σ : c.F →+* ℂ, _root_.Formalism.L2.Pj.Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ f ∈ _root_.Formalism.L2.Pj.sec Xo c a n, (∀ σ' : c'.F →+* ℂ, _root_.Formalism.L2.Pj.rdU (_root_.Formalism.L2.Pj.unfBC hle U σ') (_root_.GlobalTheory.FieldBC.rulBC hle f) = _root_.Formalism.L2.Pj.rdU (U (σ'.comp (_root_.Formalism.L2.Pj.incF c c' hle))) f) ∧ (_root_.GlobalTheory.FieldBC.rulBC hle f ∈ _root_.Formalism.L2.Pj.cusp Xo c' (_roo

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsEqMulProdMinors

theorem **coeff_X_mul_pderiv** (u : σ) (m : σ →₀ ℕ) (P : MvPolynomial σ K) : coeff m (X u * pderiv u P) = (m u : K) * coeff m P

theorem **eval_congr_vars** {P : MvPolynomial σ K} {x x' : σ → K} (h : ∀ s ∈ P.vars, x s = x' s) : eval x P = eval x' P

theorem **eval_mul_prod_eq** {J : Type*} [Fintype J] (B : J → Set σ) (hB : ∀ i j, i ≠ j → ∀ s ∈ B i, s ∉ B j) (D : J → MvPolynomial σ K) (hD : ∀ j (x x' : σ → K), (∀ s ∈ B j, x s = x' s) → eval x (D j) = eval x' (D j)) (P : MvPolynomial σ K) (hP : ∀ j, ∃ Q : MvPolynomial σ K, (∀ s ∈ B j, s ∉ Q.vars) ∧ P = Q * D j) (hv : ∀ s ∈ P.vars, ∃ j, s ∈ B j) (x₀ x : σ → K) : eval x P * ∏ j, eval x₀ (D j) = eval x₀ P * ∏ j, eval x (D j)

theorem **weight_of_mem_support** (z : Fin p × Fin n → σ) (y : Fin q × Fin n → σ) {P : MvPolynomial σ K} {k : Fin n} {w : ℤ} (h : mixedOp K z y k k P = (w : K) • P) {m : σ →₀ ℕ} (hm : m ∈ P.support) : ∑ a, (m (z (a, k)) : ℤ) − ∑ b, (m (y (b, k)) : ℤ) = w

theorem **rowOp_self** (z : Fin p × Fin n → σ) {P : MvPolynomial σ K} {a : Fin p} {d : ℕ} (h : ∀ m ∈ P.support, ∑ k, m (z (a, k)) = d) : rowOp K z a a P = ((d : ℤ) : K) • P

theorem **casimir_rows** [DecidableEq σ] (z : Fin p × Fin n → σ) (hz : Function.Injective z) {P : MvPolynomial σ K} (l : Fin p → ℕ) (hrow : ∀ a, ∀ m ∈ P.support, ∑ k, m (z (a, k)) = l a) (hinv : ∀ a b : Fin p, a < b → ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = z (b, k) then C t * X (z (a, k)) else 0) P = P) : ∑ a, ∑ b, rowOp K z a b (rowOp K z b a P) + ((n : K) − (p : K)) • ∑ a, rowOp K z a a P = ((∑ i : Fin p, (l i : ℤ) * (l i + p − 1 − 2 * (i : ℕ)) + ((n : ℤ) − p) * ∑ i, (l i : ℤ) : ℤ) : K) • P

theorem **sum_filter_pad** (hp : p ≤ n) (a : Fin p → ℕ) (i : ℕ) : ∑ k ∈ Finset.univ.filter (fun k : Fin n => (k : ℕ) < i), (if h : (k : ℕ) < p then (a ⟨k, h⟩ : ℤ) else 0) = ∑ j ∈ Finset.univ.filter (fun j : Fin p => (j : ℕ) < i), (a j : ℤ)

theorem **sum_filter_pad_rev** (hq : q ≤ n) (b : Fin q → ℕ) (i : ℕ) : ∑ k ∈ Finset.univ.filter (fun k : Fin n => ¬ (k : ℕ) < i), (if h : n − 1 − (k : ℕ) < q then (b ⟨n − 1 − k, h⟩ : ℤ) else 0) = ∑ j ∈ Finset.univ.filter (fun j : Fin q => (j : ℕ) < n − i), (b j : ℤ)

theorem **weight_eq_mixedWeight** (hP : P ≠ 0) : κ = mixedWeight n a b

theorem **notMem_vars** : (∀ i (k : Fin n), p ≤ k → z (i, k) ∉ P.vars) ∧ ∀ j (k : Fin n), (k : ℕ) < n − q → y (j, k) ∉ P.vars

theorem **exists_eq_mul_prod_minors** [DecidableEq σ] (z : Fin p × Fin n → σ) (hz : Function.Injective z) (e : Fin p → Fin n) (he : Function.Injective e) {P : MvPolynomial σ K} (c : Fin p → ℕ) (hrow : ∀ i, ∀ m ∈ P.support, ∑ k, m (z (i, k)) = partOf c i) (hinv : ∀ i j : Fin p, i < j → ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = z (j, k) then C t * X (z (i, k)) else 0) P = P) (hv : ∀ i k, k ∉ Set.range e → z (i, k) ∉ P.vars) (hcol : ∀ k l, k < l → colOp K z (e k) (e l) P = 0) : ∃ Q : MvPolynomial σ K, (∀ w : Fin p × Fin p, z (w.1, e w.2) ∉ Q.vars) ∧ P = Q * ∏ i : Fin p, (Matrix.of fun a b : Fin (i.val + 1) => (X (z (lead i a, e (lead i b))) : MvPolynomial σ K)).det ^ c i

theorem **exists_eq_smul_basePoly** {K T : Type*} [Field K] [CharZero K] [Fintype T] {p q n : ℕ} (h : p + q ≤ n) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {P : Poly K T p q (Fin n)} (hP : P ∈ rowHighest K c c') (κ : T → Fin n → ℤ) (hκ : ∀ t, Antitone (κ t)) (hup : ∀ t (k l : Fin n), k < l → ∑ a, X (xVar t a k) * pderiv (xVar t a l) P − ∑ b, X (yVar t b l) * pderiv (yVar t b k) P = 0) (hwt : ∀ t (k : Fin n), ∑ a, X (xVar t a k) * pderiv (xVar t a k) P − ∑ b, X (yVar t b k) * pderiv (yVar t b k) P = (κ t k : K) • P) : ∃ r : K, P = r • basePoly K h c c'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsExpansionOfEqSumThFr

theorem **hasSum_prod_of_fintype** {J ι : Type*} [Fintype J] {f : J → ι → ℂ} {y : J → ℂ} (hf : ∀ j, Summable fun m => ‖f j m‖) (h : ∀ j, HasSum (f j) (y j)) : (Summable fun p : J × ι => ‖f p.1 p.2‖) ∧ HasSum (fun p : J × ι => f p.1 p.2) (∑ j, y j)

theorem **exists_expansion_of_rowHighest** (hs : IsSympl X bas) (hn : Even n) (hpq : X.p0 + q0 ≤ n) (hc : ∃ τ, (∃ i : Fin q0, i.val + 1 = q0 ∧ 1 ≤ c τ i) ∨ ∃ i : Fin X.p0, i.val + 1 = X.p0 ∧ 1 ≤ c' τ i) (a b : Fin n → Fin g → ℝ) {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : P ∈ Harm.rowHighest ℂ c c') {ξ : Fin r → E} (hξ : X.IsIso ξ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : ∃ (A : (Fin n → Fin g → ℤ) → ℂ) (ν : (Fin n → Fin g → ℤ) → X.Ψ → ℝ), (Summable fun m => ‖A m‖) ∧ (∀ s : X.Ψ → ℝ, HasSum (fun m => A m * cexp (4 * Real.pi * I * ∑ τ, (ν m τ : ℂ) * s τ)) (thFr X hr bas idx n a b P (X.flow ξ (fun τ => (s τ : ℂ)) x))) ∧ ∀ m, A m ≠ 0 → ν m ≠ 0

theorem **isSec_of_eq_sum_thFr** (hs : IsSympl X bas) (hP : ∀ j ∈ s, P j ∈ Harm.rowHighest ℂ c c') (hQ : ∀ x ∈ X.dom, Q x = ∑ j ∈ s, t j * thFr X hr bas idx n (a j) (b j) (P j) x) (hΓ : ∀ γ ∈ X.Γ, ∀ x ∈ X.dom, Q (lact γ x) = X.χ (wtOf X hr c c' n) γ * Q x) : X.IsSec (wtOf X hr c c' n) Q

theorem **isModerate_of_eq_sum_thFr** (hs : IsSympl X bas) (hn : Even n) (hP : ∀ j ∈ s, P j ∈ Harm.rowHighest ℂ c c') (hQ : ∀ x ∈ X.dom, Q x = ∑ j ∈ s, t j * thFr X hr bas idx n (a j) (b j) (P j) x) : X.IsModerate (wtOf X hr c c' n) Q

theorem **exists_expansion_of_eq_sum_thFr** (hs : IsSympl X bas) (hn : Even n) (hpq : X.p0 + q0 ≤ n) (hc : ∃ τ, (∃ i : Fin q0, i.val + 1 = q0 ∧ 1 ≤ c τ i) ∨ ∃ i : Fin X.p0, i.val + 1 = X.p0 ∧ 1 ≤ c' τ i) (hP : ∀ j ∈ s, P j ∈ Harm.rowHighest ℂ c c') (hQ : ∀ x ∈ X.dom, Q x = ∑ j ∈ s, t j * thFr X hr bas idx n (a j) (b j) (P j) x) {ξ : Fin r → E} (hξ : X.IsIso ξ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : ∃ (A : s × (Fin n → Fin g → ℤ) → ℂ) (ν : s × (Fin n → Fin g → ℤ) → X.Ψ → ℝ), (Summable fun m => ‖A m‖) ∧ (∀ u : X.Ψ → ℝ, HasSum (fun m => A m * cexp (4 * Real.pi * I * ∑ τ, (ν m τ : ℂ) * u τ)) (Q (X.flow ξ (fun τ => (u τ : ℂ)) x))) ∧ ∀ m, A m ≠ 0 → ν m ≠ 0

theorem **mem_SS_of_eq_sum_thFr** (hs : IsSympl X bas) (hn : Even n) (hc : (X.p0 + q0 ≤ n ∧ ∃ τ, (∃ i : Fin q0, i.val + 1 = q0 ∧ 1 ≤ c τ i) ∨ ∃ i : Fin X.p0, i.val + 1 = X.p0 ∧ 1 ≤ c' τ i) ∨ ∀ ξ, ¬ X.IsIso ξ) (hP : ∀ j ∈ s, P j ∈ Harm.rowHighest ℂ c c') (hQ : ∀ x ∈ X.dom, Q x = ∑ j ∈ s, t j * thFr X hr bas idx n (a j) (b j) (P j) x) (hΓ : ∀ γ ∈ X.Γ, ∀ x ∈ X.dom, Q (lact γ x) = X.χ (wtOf X hr c c' n) γ * Q x) : Q ∈ X.SS (wtOf X hr c c' n)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatCoverLevelSections

theorem **exists_faithfullyFlat_isVtOver** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (n : ℕ) (x : Fin n → Dw.La), ∃ (S' : Type) (_ : CommRing S') (_ : Algebra c.F S') (φ : S →ₐ[c.F] S') (y' : FrPt Xo c S') (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y'.x))), φ.toRingHom.FaithfullyFlat ∧ FrMap φ y y' ∧ ∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) y'.s (levPt Xo y'.x (x b)) (F b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatCoverLevelSectionsStatement

theorem **_root_.GlobalTheory.T.exists_faithfullyFlat_cover_level_sections** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (n : ℕ) (x : Fin n → Dw.La), ∃ (S' : Type) (_ : CommRing S') (_ : Algebra c.F S') (φ : S →ₐ[c.F] S') (y' : FrPt Xo c S') (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y'.x))), φ.toRingHom.FaithfullyFlat ∧ FrMap φ y y' ∧ ∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) y'.s (levPt Xo y'.x (x b)) (F b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatCoverShortThetaValueData

theorem **finite_levQ** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (Dw : Datum H) (l : Level H) : Finite (LevQ Dw l)

theorem **exists_cover_shortVal_data** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), ∃ (S' : Type) (_ : CommRing S') (_ : Algebra c.F S') (φ : S →ₐ[c.F] S') (y' : FrPt Xo c S') (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y'.x ⟶ Xo.univ.t.G)) (tp : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y'.x ⟶ Xo.univ.t.G)) (F : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y'.x))), φ.toRingHom.FaithfullyFlat ∧ FrMap φ y y' ∧ (∀ i, InZh Xo Λ N (z i)) ∧ GeoInj Xo y'.x z ∧ (∀ q b, tp q b ^ N = levPt Xo y'.x (levRep (q b)) ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) (tp q b)) ∧ (∀ q i b, IsUUOver Xo.univ.t.G Xo.un

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatCoverShortThetaValueDataStatement

theorem **_root_.GlobalTheory.T.exists_faithfullyFlat_cover_shortThetaValue_data** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), ∃ (S' : Type) (_ : CommRing S') (_ : Algebra c.F S') (φ : S →ₐ[c.F] S') (y' : FrPt Xo c S') (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y'.x ⟶ Xo.univ.t.G)) (tp : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y'.x ⟶ Xo.univ.t.G)) (F : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y'.x))), φ.toRingHom.FaithfullyFlat ∧ FrMap φ y y' ∧ (∀ i, InZh Xo Λ N (z i)) ∧ GeoInj Xo y'.x z ∧ (∀ q b, tp q b ^ N = levPt Xo y'.x (levRep (q b)) ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) (tp

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatFullLinearLevelStructure

theorem **exists_finite_quotient** (Dw : Datum H) (k : ℕ) (hk : 0 < k) : ∃ (J : Type) (_ : Fintype J) (π : Dw.La → J) (r : J → Dw.La), (∀ j, π (r j) = j) ∧ ∀ a b, π a = π b ↔ Dw.ModN k (a − b)

private lemma **eq_one_of_isIdempotentElem_of_isUnit** {T : Type} [CommRing T] {x : T} (h : IsIdempotentElem x) (hu : IsUnit x) : x = 1

theorem **exists_element** (hL : Laws) {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) (B : Type) [CommRing B] (x0 : Sp B ⟶ Xo.X) (k : ℕ) (hk : 0 < k) (I I' : Type) [Fintype I] [Fintype I'] (a b : I → (ptO Xo x0 ⟶ Xo.univ.t.G)) (c d : I' → (ptO Xo x0 ⟶ Xo.univ.t.G)) (ha : ∀ i, a i ^ k = 1) (hb : ∀ i, b i ^ k = 1) (hc : ∀ i, c i ^ k = 1) (hd : ∀ i, d i ^ k = 1) : ∃ e : B, (∀ (T : Type) [CommRing T] (χ : B →+* T), IsUnit (χ e) → (∀ i, spMap χ ≫ (a i).left = spMap χ ≫ (b i).left) ∧ (Nontrivial T → ∀ j, spMap χ ≫ (c j).left ≠ spMap χ ≫ (d j).left)) ∧ ∀ (K : Type) [Field K] (χ : B →+* K), (∀ i, spMap χ ≫ (a i).left = spMap χ ≫ (b i).left) → (∀ j, spMap χ ≫ (c j).left ≠ spMap χ ≫ (d j).left) → IsUnit (χ e)

theorem **lin_of_generators** {X : Scheme.{0}} {O M : Type} [CommRing O] [AddCommGroup M] [Module O M] {G T : Over X} [GrpObj G] (A : Geo.Tor.OAct O G) (ev : M → (T ⟶ G)) (hadd : ∀ a b, ev (a + b) = ev a ∗ ev b) {s : Set O} (hs : AddSubgroup.closure s = ⊤) (h : ∀ o ∈ s, ∀ a, ev (o • a) = ev a ≫ A.act o) (o : O) (a : M) : ev (o • a) = ev a ≫ A.act o

private lemma **spMap_comp_moduli_spaces** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R →+* R') (g : R' →+* R'') : spMap (g.comp f) = spMap g ≫ spMap f

private lemma **modN_add'** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {k : ℕ} {x y : Dw.La} (hx : Dw.ModN k x) (hy : Dw.ModN k y) : Dw.ModN k (x + y)

private lemma **modN_neg'** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {k : ℕ} {x : Dw.La} (hx : Dw.ModN k x) : Dw.ModN k (-x)

private lemma **modN_smul'** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {k : ℕ} (o : 𝒪 E) {x : Dw.La} (hx : Dw.ModN k x) : Dw.ModN k (o • x)

theorem **exists_faithfullyFlat_linear_level** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (k : ℕ), 0 < k → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (ev : Dw.La → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ (∀ a b, ev (a + b) = ev a ∗ ev b) ∧ (∀ (o : 𝒪 E) a, ev (o • a) = ev a ≫ Xo.univ.t.act.act o) ∧ (∀ a, ev (k • a) = 1) ∧ ∀ (K : Type) [Field K] (χ : B →+* K) (a b : Dw.La), spMap χ ≫ (ev a).left = spMap χ ≫ (ev b).left → Dw.ModN k (a - b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatFullLinearLevelStructureStatement

theorem **_root_.GlobalTheory.T.exists_faithfullyFlat_full_linear_level_structure** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (k : ℕ), 0 < k → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (ev : Dw.La → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ (∀ a b, ev (a + b) = ev a ∗ ev b) ∧ (∀ (o : 𝒪 E) a, ev (o • a) = ev a ≫ Xo.univ.t.act.act o) ∧ (∀ a, ev (k • a) = 1) ∧ ∀ (K : Type) [Field K] (χ : B →+* K) (a b : Dw.La), spMap χ ≫ (ev a).left = spMap χ ≫ (ev b).left → Dw.ModN k (a - b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatRoots

theorem **exists_lift** {X Y : Scheme.{0}} (f : X ⟶ Y) [IsFinite f] [Flat f] [Surjective f] (S : Type) [CommRing S] (t : Sp S ⟶ Y) : ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (w : Sp B ⟶ X), φ.FaithfullyFlat ∧ w ≫ f = spMap φ ≫ t

theorem **spMap_comp** {R S T : Type} [CommRing R] [CommRing S] [CommRing T] (f : R →+* S) (g : S →+* T) : spMap (g.comp f) = spMap g ≫ spMap f

theorem **exists_lifts** {X Y : Scheme.{0}} (f : X ⟶ Y) [IsFinite f] [Flat f] [Surjective f] (S : Type) [CommRing S] (J : Type) (t : J → (Sp S ⟶ Y)) (s : Finset J) : ∃ (B : Type) (_ : CommRing B) (φ : S →+* B), φ.FaithfullyFlat ∧ ∀ i ∈ s, ∃ w : Sp B ⟶ X, w ≫ f = spMap φ ≫ t i

theorem **exists_faithfullyFlat_roots** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (N k : ℕ), 0 < N → 0 < k → ∀ (J : Type) [Fintype J] (t : J → (ptO Xo x0 ⟶ Xo.univ.t.G)), (∀ i, t i ^ k = 1) → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (w : J → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ ∀ i, (w i ^ N).left = spMap φ ≫ (t i).left

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatRootsStatement

theorem **_root_.GlobalTheory.T.exists_faithfullyFlat_nthRoots_of_torsion** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (N k : ℕ), 0 < N → 0 < k → ∀ (J : Type) [Fintype J] (t : J → (ptO Xo x0 ⟶ Xo.univ.t.G)), (∀ i, t i ^ k = 1) → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (w : J → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ ∀ i, (w i ^ N).left = spMap φ ≫ (t i).left

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatSplitIsotropicGroup

theorem **map_sum_eq_prod_ofFn_of_add_eq_mul** {A G : Type} [AddCommGroup A] [Group G] (ev : A → G) (h : ∀ a b, ev (a + b) = ev a ∗ ev b) : ∀ (n : ℕ) (f : Fin n → A), ev (∑ j, f j) = (List.ofFn fun j => ev (f j)).prod

theorem **map_nsmul_eq_pow_of_add_eq_mul** {A G : Type} [AddCommGroup A] [Group G] (ev : A → G) (h : ∀ a b, ev (a + b) = ev a ∗ ev b) (a : A) (k : ℕ) : ev (k • a) = ev a ^ k

theorem **exists_faithfullyFlat_split_Zh** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ (∀ i, InZh Xo Λ N (z i)) ∧ GeoInj Xo (spMap φ ≫ x0) z

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFaithfullyFlatSplitIsotropicGroupStatement

theorem **_root_.GlobalTheory.T.exists_faithfullyFlat_split_isotropicGroup** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ (∀ i, InZh Xo Λ N (z i)) ∧ GeoInj Xo (spMap φ ≫ x0) z

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFiniteSpanningShortGeneratorsSmallSupNorm

theorem **exists_finset_span_top_of_span** {O K R : Type} [CommRing O] [Field K] [Algebra O K] [AddCommGroup R] [Module K R] [Module O R] [IsScalarTower O K R] (C : Submodule K R) [FiniteDimensional K C] (s A : Set R) (hA : A ⊆ C) (hs : ∀ g ∈ s, g ∈ Submodule.span K A) (htop : Submodule.span K (((Submodule.span O s).comap (C.subtype.restrictScalars O) : Submodule O C) : Set C) = ⊤) : ∃ Sh : Finset C, (∀ g ∈ Sh, (g : R) ∈ A) ∧ Submodule.span K (Sh : Set C) = ⊤

theorem **exists_short_family_small** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → LawsSh T → (T.dδ j w.n).l₂.A = 1 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) w → ∃ Sh : Finset ((recB h.toLawsCore).Cusp j w), (∀ g ∈ Sh, g ∈ (recPJ h j hj).shortTh w ∧ ∀ σ, Real.log ((recB h.toLawsCore).supN j w σ g) ≤ 400 ∗ Zb D large₂ ∗ w.size) ∧ Submodule.span (recB h.toLawsCore).KO (Sh : Set ((recB h.toLawsCore).Cusp j w)) = ⊤

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFiniteSpanningShortGeneratorsSmallSupNormStatement

theorem **_root_.GlobalTheory.T.exists_finite_spanning_short_generators_small_supNorm** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → LawsSh T → (T.dδ j w.n).l₂.A = 1 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) w → ∃ Sh : Finset ((recB h.toLawsCore).Cusp j w), (∀ g ∈ Sh, g ∈ (recPJ h j hj).shortTh w ∧ ∀ σ, Real.log ((recB h.toLawsCore).supN j w σ g) ≤ 400 ∗ Zb D large₂ ∗ w.size) ∧ Submodule.span (recB h.toLawsCore).KO (Sh : Set ((recB h.toLawsCore).Cusp j w)) = ⊤

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFinsetMeetingFibres

theorem **norm_det_le** {r : ℕ} {m : Matrix (Fin r) (Fin r) ℂ} {s : ℝ} (hs : 0 ≤ s) (hm : ∀ i k, ‖m i k‖ ≤ s) : ‖m.det‖ ≤ (r : ℝ) ^ r * s ^ r

theorem **norm_mul_inv_sub_one_le** {r : ℕ} {m m' : Matrix (Fin r) (Fin r) ℂ} {s ε : ℝ} (hs : 1 ≤ s) (hm : ∀ i k, ‖m' i k‖ ≤ s) (hd : m'.det ≠ 0) (hε : ∀ i k, ‖m i k - m' i k‖ ≤ ε) (a b : Fin r) : ‖(m * m'⁻¹) a b - (if a = b then 1 else 0)‖ ≤ r * (ε * ((r : ℝ) ^ r * s ^ r / ‖m'.det‖))

theorem **abs_sub_le_of_floor_eq** {s δ a b : ℝ} (hδ : 0 < δ) (h : ⌊(a + s) / δ⌋ = ⌊(b + s) / δ⌋) : |a - b| ≤ δ

theorem **norm_sub_le_of_cell_eq** {s δ : ℝ} (hδ : 0 < δ) {z w : ℂ} (h : cell s δ z = cell s δ w) : ‖z - w‖ ≤ 2 * δ

theorem **floor_mem_Icc** {s δ a : ℝ} (hδ : 0 < δ) (h : |a| ≤ s) : ⌊(a + s) / δ⌋ ∈ Finset.Icc (0 : ℤ) ⌊2 * s / δ⌋

theorem **cell_mem** {s δ : ℝ} (hδ : 0 < δ) {z : ℂ} (h : ‖z‖ ≤ s) : cell s δ z ∈ Finset.Icc (0 : ℤ) ⌊2 * s / δ⌋ ×ˢ Finset.Icc (0 : ℤ) ⌊2 * s / δ⌋

theorem **card_Icc_floor_le** {t : ℝ} (ht : 0 ≤ t) : ((Finset.Icc (0 : ℤ) ⌊t⌋).card : ℝ) ≤ t + 1

theorem **exists_finset_meeting_fibres** {α β : Type*} (O : Set α) (g : α → β) (S : Finset β) (hS : ∀ x ∈ O, g x ∈ S) : ∃ N : Finset α, (N : Set α) ⊆ O ∧ N.card ≤ S.card ∧ ∀ x ∈ O, ∃ y ∈ N, g x = g y

theorem **exists_net_of_matrices** {α ι : Type*} [Fintype ι] {r : ℕ} (O : Set α) (m : α → ι → Matrix (Fin r) (Fin r) ℂ) (D : ι → ℝ) (hD0 : ∀ τ, 0 < D τ) (hD : 1 ≤ ∏ τ, D τ) {B η : ℝ} (hB : 1 ≤ B) (hη0 : 0 < η) (hη : η ≤ 1) (hm : ∀ x ∈ O, ∀ τ i k, ‖m x τ i k‖ ^ 2 ≤ B) (hdet : ∀ x ∈ O, ∀ τ, ‖(m x τ).det‖ = D τ) : ∃ N : Finset α, (N : Set α) ⊆ O ∧ (N.card : ℝ) ≤ (6 * (r : ℝ) ^ 2 / η * ((r : ℝ) * B) ^ r) ^ (2 * Fintype.card ι * r ^ 2) ∧ ∀ x ∈ O, ∃ y ∈ N, ∀ τ a b, ‖(m x τ * (m y τ)⁻¹) a b - (if a = b then 1 else 0)‖ ≤ η

theorem **gram_basisMat** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (v : Fin r → Fin r → E) : gram τ G (basisMat τ v) = (gramMat G v).map τ

theorem **norm_det_gram** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (M : Matrix (Fin r) (Fin r) ℂ) : ‖(gram τ G M).det‖ = ‖M.det‖ ^ 2 * ‖τ G.det‖

theorem **det_basisMat_ne_zero** (τ : E →+* ℂ) {v : Fin r → Fin r → E} (hv : LinearIndependent E v) : (basisMat τ v).det ≠ 0

theorem **norm_emb_det_gramMat** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (v : Fin r → Fin r → E) : ‖τ (gramMat G v).det‖ = ‖(basisMat τ v).det‖ ^ 2 * ‖τ G.det‖

theorem **det_gramMat_ne_zero** {v : Fin r → Fin r → E} (hv : LinearIndependent E v) : (gramMat X.Gm v).det ≠ 0

theorem **norm_det_sq_of_ortho** {x : Frame X.Ψ r} (hx : X.IsOrtho x) (τ : X.Ψ) : ‖(mat x τ).det‖ ^ 2 * ‖τ.1 X.Gm.det‖ = 1

theorem **det_ne_zero_of_ortho** {x : Frame X.Ψ r} (hx : X.IsOrtho x) (τ : X.Ψ) : (mat x τ).det ≠ 0

theorem **norm_det_coord** {x : Frame X.Ψ r} (hx : X.IsOrtho x) (τ : X.Ψ) (v : Fin r → Fin r → E) : ‖((mat x τ)⁻¹ * basisMat τ.1 v).det‖ = Real.sqrt ‖τ.1 (gramMat X.Gm v).det‖

theorem **maj_eq_sum_coord** {x : Frame X.Ψ r} (hx : X.IsOrtho x) (τ : X.Ψ) (v : Fin r → Fin r → E) (k : Fin r) : X.maj x τ (Data.emb τ.1 (v k)) = ∑ i, ‖((mat x τ)⁻¹ * basisMat τ.1 v) i k‖ ^ 2

theorem **mem_shortFrames** {v : Fin r → Fin r → E} {B : ℝ} {x : Frame X.Ψ r} : x ∈ shortFrames X v B ↔ (∀ τ : X.Ψ, gram τ.1 X.Gm (mat x τ) = Matrix.diagonal fun a : Fin r => ((sgn X.p0 (a : ℕ) : ℝ) : ℂ)) ∧ ∀ i, ∑ τ : X.Ψ, X.maj x τ (Data.emb τ.1 (v i)) ≤ B

theorem **shortFrames_subset_dom** (v : Fin r → Fin r → E) (B : ℝ) : shortFrames X v B ⊆ X.dom

theorem **exists_net_card_le** (v : Fin r → Fin r → E) (hv : LinearIndependent E v) (hint : ∀ i k, IsIntegral ℤ (herm X.Gm (v i) (v k))) {B η : ℝ} (hB : 1 ≤ B) (hη0 : 0 < η) (hη : η ≤ 1) : ∃ N : Finset (Frame X.Ψ r), (N : Set (Frame X.Ψ r)) ⊆ shortFrames X v B ∧ (∀ x ∈ shortFrames X v B, ∃ A : Frame X.Ψ r, rmul x A ∈ N ∧ ∀ τ a b, ‖A τ a b - (if a = b then 1 else 0)‖ ≤ η) ∧ (N.card : ℝ) ≤ (6 * (r : ℝ) ^ 2 / η * ((r : ℝ) * B) ^ r) ^ (2 * fdeg E * r ^ 2)

theorem **log_le_of_card_le** {n f r : ℕ} {B η : ℝ} (hB : 1 ≤ B) (hη0 : 0 < η) (hη : η ≤ 1) (h : (n : ℝ) ≤ (6 * (r : ℝ) ^ 2 / η * ((r : ℝ) * B) ^ r) ^ (2 * f * r ^ 2)) : (n : ℝ) ≤ Real.exp (2 * f * (r : ℝ) ^ 2 * Real.log (6 * (r : ℝ) ^ 2 / η) + 2 * f * (r : ℝ) ^ 3 * Real.log (r * B)) ∧ Real.log n ≤ 2 * f * (r : ℝ) ^ 2 * Real.log (6 * (r : ℝ) ^ 2 / η) + 2 * f * (r : ℝ) ^ 3 * Real.log (r * B)

theorem **exists_frame_net** (v : Fin r → Fin r → E) (hv : LinearIndependent E v) (hint : ∀ i k, IsIntegral ℤ (herm X.Gm (v i) (v k))) {B η : ℝ} (hB : 1 ≤ B) (hη0 : 0 < η) (hη : η ≤ 1) : ∃ N : Finset (Frame X.Ψ r), (N : Set (Frame X.Ψ r)) ⊆ shortFrames X v B ∧ (∀ x ∈ shortFrames X v B, ∃ A : Frame X.Ψ r, rmul x A ∈ N ∧ ∀ τ a b, ‖A τ a b - (if a = b then 1 else 0)‖ ≤ η) ∧ (N.card : ℝ) ≤ Real.exp (2 * (fdeg E : ℝ) * (r : ℝ) ^ 2 * Real.log (6 * (r : ℝ) ^ 2 / η) + 2 * (fdeg E : ℝ) * (r : ℝ) ^ 3 * Real.log (r * B)) ∧ Real.log N.card ≤ 2 * (fdeg E : ℝ) * (r : ℝ) ^ 2 * Real.log (6 * (r : ℝ) ^ 2 / η) + 2 * (fdeg E : ℝ) * (r : ℝ) ^ 3 * Real.log (r * B)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFinsetSelfDualClassesHerm

theorem **hq_embVec_pos** {H : Matrix (Fin r) (Fin r) ℂ} (hH : H.PosDef) (σ : K →+* ℂ) {v : Fin r → K} (hv : v ≠ 0) : 0 < hq H (embVec σ v)

theorem **sum_log_hq_nonneg** {H : (K →+* ℂ) → Matrix (Fin r) (Fin r) ℂ} {L : Submodule (𝒪 K) (Fin r → K)} (hss : IsSemistableZero H L) {v : Fin r → K} (hv : v ∈ L) (hv0 : v ≠ 0) : 0 ≤ ∑ σ : K →+* ℂ, Real.log (hq (H σ) (embVec σ v))

theorem **finrank_le_sum_hq** {H : (K →+* ℂ) → Matrix (Fin r) (Fin r) ℂ} (hH : ∀ σ, (H σ).PosDef) {L : Submodule (𝒪 K) (Fin r → K)} (hss : IsSemistableZero H L) {v : Fin r → K} (hv : v ∈ L) (hv0 : v ≠ 0) : (finrank ℚ K : ℝ) ≤ ∑ σ : K →+* ℂ, hq (H σ) (embVec σ v)

theorem **normSq_mulVec** (C : Matrix (Fin r) (Fin r) ℂ) (u : Fin r → ℂ) : normSq (C *ᵥ u) = hq (Cᴴ * C) u

theorem **exists_coord** {H G : Matrix (Fin r) (Fin r) ℂ} (hH : H.PosDef) (hG : G.IsHermitian) (hdet : G.det ≠ 0) (hmaj : H * G⁻¹ * H = G) : ∃ C D : Matrix (Fin r) (Fin r) ℂ, (∀ u, normSq (C *ᵥ u) = hq H u) ∧ (∀ u, normSq (D *ᵥ u) = hq H u) ∧ ∀ u w, star (C *ᵥ u) ⬝ᵥ (D *ᵥ w) = star u ⬝ᵥ (G *ᵥ w)

theorem **emb_h** (S : HermForm K r) (σ : K →+* ℂ) (x y : Fin r → K) : σ (S.h x y) = star (embVec σ x) ⬝ᵥ (S.G.map σ *ᵥ embVec σ y)

theorem **isHermitian_map** (S : HermForm K r) (σ : K →+* ℂ) : (S.G.map σ).IsHermitian

theorem **norm_dotProduct_le** (u w : Fin r → ℂ) : ‖star u ⬝ᵥ w‖ ≤ (normSq u + normSq w) / 2

theorem **norm_h_le** {S : HermForm K r} (hdet : S.G.det ≠ 0) (M : Majorant S) (σ : K →+* ℂ) (x y : Fin r → K) : ‖σ (S.h x y)‖ ≤ (hq (M.H σ) (embVec σ x) + hq (M.H σ) (embVec σ y)) / 2

theorem **hq_conjugate** {S : HermForm K r} (M : Majorant S) (σ : K →+* ℂ) (x : Fin r → K) : hq (M.H (ComplexEmbedding.conjugate σ)) (embVec (ComplexEmbedding.conjugate σ) x) = hq (M.H σ) (embVec σ x)

theorem **embedding_injective** : Function.Injective (fun w : InfinitePlace K => w.embedding)

theorem **embedding_cmType** (τ : K →+* ℂ) : (∃ w : InfinitePlace K, w.embedding = τ) ↔ ¬ ∃ w : InfinitePlace K, w.embedding = ComplexEmbedding.conjugate τ

theorem **card_infinitePlace** : Fintype.card (InfinitePlace K) = fdeg K

theorem **sum_hq_eq** {S : HermForm K r} (M : Majorant S) (x : Fin r → K) : ∑ σ : K →+* ℂ, hq (M.H σ) (embVec σ x) = 2 * ∑ w : InfinitePlace K, hq (M.H w.embedding) (embVec w.embedding x)

theorem **two_mul_hq_le** {S : HermForm K r} (M : Majorant S) (σ : K →+* ℂ) (x : Fin r → K) : 2 * hq (M.H σ) (embVec σ x) ≤ ∑ τ : K →+* ℂ, hq (M.H τ) (embVec τ x)

theorem **exists_short_basis** {S : HermForm K r} {L : Submodule (𝓞 K) (Fin r → K)} (hL : IsLattice L) (hsd : IsSelfDual S L) (M : Majorant S) {ε : ℝ} (hε : 0 < ε) (hmin : ∀ v ∈ L, v ≠ 0 → ε ≤ (∑ σ : K →+* ℂ, hq (M.H σ) (embVec σ v)) / 2) : ∃ v : Fin r → Fin r → K, (∀ i, v i ∈ L) ∧ LinearIndependent K v ∧ ∀ i, (∑ σ : K →+* ℂ, hq (M.H σ) (embVec σ (v i))) / 2 ≤ (2 * fdeg K * r : ℝ) ^ 4 * dE K ^ 2 / (4 * ε)

theorem **fdeg_pos** : 0 < fdeg K

theorem **exists_bounded_basis** {S : HermForm K r} {L : Submodule (𝓞 K) (Fin r → K)} (hL : IsLattice L) (hsd : IsSelfDual S L) : ∃ v : Fin r → Fin r → K, (∀ i, v i ∈ L) ∧ LinearIndependent K v ∧ ∀ (σ : K →+* ℂ) (i k : Fin r), ‖σ (S.h (v i) (v k))‖ ≤ (2 * fdeg K * r : ℝ) ^ 4 * dE K ^ 2 / (4 * fdeg K)

theorem **exists_finset_submodule** {R : Type} [CommRing R] [IsDedekindDomain R] {r : ℕ} (N : Matrix (Fin r) (Fin r) R) (hN : N.det ≠ 0) [Finite ((Fin r → R) ⧸ LinearMap.range N.vecMulLinear)] : ∃ T : Finset (Submodule R (Fin r → R)), T.card ≤ Nat.card ((Fin r → R) ⧸ LinearMap.range N.vecMulLinear) ^ r ∧ ∀ P, LinearMap.range N.vecMulLinear ≤ P → P ∈ T

theorem **range_le_pairings** {G : Matrix (Fin r) (Fin r) K} {L : Submodule (𝓞 K) (Fin r → K)} {v : Fin r → Fin r → K} (hv : ∀ i, v i ∈ L) {A : Matrix (Fin r) (Fin r) (𝓞 K)} (hA : gramOf G v = A.map (algebraMap (𝓞 K) K)) : LinearMap.range A.vecMulLinear ≤ pairings G L v

theorem **vecMul_mem_of_pairings_le** {G : Matrix (Fin r) (Fin r) K} (hG : G.det ≠ 0) {L L' : Submodule (𝓞 K) (Fin r → K)} {v w : Fin r → Fin r → K} (hint : ∀ x ∈ L, ∀ k, IsIntegral ℤ (herm G x (v k))) (hw : LinearIndependent K w) (hgram : gramOf G v = gramOf G w) (h : pairings G L v ≤ pairings G L' w) (c : Fin r → K) (hc : c ᵥ* Matrix.of v ∈ L) : c ᵥ* Matrix.of w ∈ L'

theorem **exists_finset_types** {B : ℝ} (hB : 0 ≤ B) : ∃ T : Finset (Matrix (Fin r) (Fin r) (𝓞 K) × Submodule (𝓞 K) (Fin r → 𝓞 K)), (T.card : ℝ) ≤ (1 + 2 * B) ^ (2 * fdeg K * r ^ 2) * (r * B) ^ (2 * fdeg K * r ^ 2) ∧ ∀ (A : Matrix (Fin r) (Fin r) (𝓞 K)) P, A.det ≠ 0 → (∀ (σ : K →+* ℂ) i k, ‖σ (A i k : K)‖ ≤ B) → LinearMap.range A.vecMulLinear ≤ P → (A, P) ∈ T

theorem **exists_finset_classes** {G : Matrix (Fin r) (Fin r) K} (ℒ : Set (Submodule (𝓞 K) (Fin r → K))) {B : ℝ} (hB : 0 ≤ B) (hℒ : ∀ L ∈ ℒ, G.det ≠ 0 ∧ (∀ x ∈ L, ∀ y ∈ L, IsIntegral ℤ (herm G x y)) ∧ ∃ v : Fin r → Fin r → K, (∀ i, v i ∈ L) ∧ LinearIndependent K v ∧ ∀ (σ : K →+* ℂ) (i k : Fin r), ‖σ (herm G (v i) (v k))‖ ≤ B) : ∃ 𝒮 : Finset (Submodule (𝓞 K) (Fin r → K)), (∀ L ∈ 𝒮, L ∈ ℒ) ∧ (∀ L ∈ ℒ, ∃ L₀ ∈ 𝒮, ∃ γ : GL (Fin r) K, (∀ x y, herm G ((γ : Matrix (Fin r) (Fin r) K) *ᵥ x) ((γ : Matrix (Fin r) (Fin r) K) *ᵥ y) = herm G x y) ∧ ∀ x, x ∈ L₀ ↔ (γ : Matrix (Fin r) (Fin r) K) *ᵥ x ∈ L) ∧ (𝒮.card : ℝ) ≤ (1 + 2 * B) ^ (2 * fdeg K * r ^ 2) * (r * B) ^ (2 * fdeg K * r ^ 2)

theorem **pow_mul_bound_le_exp** (f r : ℕ) {d ℓ M : ℝ} (hf : 16 ≤ f) (hℓ : 2 ≤ ℓ) (hd0 : 0 ≤ d) (hd : Real.log d ≤ 4 * f * ℓ) (hr : Real.log r ≤ ℓ) (hfℓ : Real.log f ≤ 2 * ℓ) (hM0 : 0 ≤ M) (hM : Real.log M ≤ 30 * ℓ) : M ^ (2 * f * r ^ 2) * ((1 + 2 * max 1 ((2 * f * r : ℝ) ^ 4 * d ^ 2 / (4 * f))) ^ (2 * f * r ^ 2) * (r * max 1 ((2 * f * r : ℝ) ^ 4 * d ^ 2 / (4 * f))) ^ (2 * f * r ^ 2)) ≤ Real.exp (20 * (f * r * ℓ) ^ 2)

theorem **c_eq_cc** (S : HermForm K r) (x : K) : S.c x = cc x

theorem **herm_transpose** (S : HermForm K r) (x y : Fin r → K) : herm S.Gᵀ x y = S.h y x

theorem **isIntegralElt_iff** {x : K} : IsIntegralElt x ↔ IsIntegral ℤ x

theorem **hermFormOf_h** (G : Matrix (Fin r) (Fin r) K) (hG : ∀ a b, cc (G a b) = G b a) (x y : Fin r → K) : (hermFormOf G hG).h x y = herm G y x

theorem **exists_finset_selfDual_classes** (S : HermForm K r) : ∃ 𝒮 : Finset (Submodule (𝓞 K) (Fin r → K)), (∀ L ∈ 𝒮, IsLattice L ∧ IsSelfDual S L) ∧ (∀ L, IsLattice L → IsSelfDual S L → ∃ L₀ ∈ 𝒮, ∃ γ : GL (Fin r) K, (∀ x y, S.h ((γ : Matrix (Fin r) (Fin r) K) *ᵥ x) ((γ : Matrix (Fin r) (Fin r) K) *ᵥ y) = S.h x y) ∧ (∀ x, x ∈ L₀ ↔ (γ : Matrix (Fin r) (Fin r) K) *ᵥ x ∈ L) ∧ L = L₀.map ((Matrix.toLin' (γ : Matrix (Fin r) (Fin r) K)).restrictScalars (𝓞 K))) ∧ (𝒮.card : ℝ) ≤ (1 + 2 * max 1 ((2 * fdeg K * r : ℝ) ^ 4 * dE K ^ 2 / (4 * fdeg K))) ^ (2 * fdeg K * r ^ 2) * (r * max 1 ((2 * fdeg K * r : ℝ) ^ 4 * dE K ^ 2 / (4 * fdeg K))) ^ (2 * fdeg K * r ^ 2) ∧ ∀ ℓ M : ℝ, 16 ≤ fdeg K → 2 ≤ ℓ → Real.log (dE K) ≤ 4 * fdeg K * ℓ → Real.log r ≤ ℓ → Real.log (fdeg K) ≤ 2 * ℓ → 0 < M → Real.log M ≤ 30 * ℓ → M ^ (2 * fdeg K * r ^ 2) * 𝒮.card ≤ Real.exp (20 * (fdeg K * r * ℓ) ^ 2) ∧ Real.log (M ^ (2 * fdeg K

theorem **exists_finset_selfDual_classes_herm** (G : Matrix (Fin r) (Fin r) K) (hG : ∀ a b, cc (G a b) = G b a) : ∃ 𝒮 : Finset (Submodule (𝓞 K) (Fin r → K)), (∀ L ∈ 𝒮, IsLattice L ∧ (∀ x ∈ L, ∀ y ∈ L, IsIntegral ℤ (herm G x y)) ∧ ∀ x, (∀ y ∈ L, IsIntegral ℤ (herm G x y)) → x ∈ L) ∧ (∀ L, IsLattice L → (∀ x ∈ L, ∀ y ∈ L, IsIntegral ℤ (herm G x y)) → (∀ x, (∀ y ∈ L, IsIntegral ℤ (herm G x y)) → x ∈ L) → ∃ L₀ ∈ 𝒮, ∃ γ : GL (Fin r) K, (∀ x y, herm G ((γ : Matrix (Fin r) (Fin r) K) *ᵥ x) ((γ : Matrix (Fin r) (Fin r) K) *ᵥ y) = herm G x y) ∧ ∀ x, x ∈ L₀ ↔ (γ : Matrix (Fin r) (Fin r) K) *ᵥ x ∈ L) ∧ (𝒮.card : ℝ) ≤ (1 + 2 * max 1 ((2 * fdeg K * r : ℝ) ^ 4 * dE K ^ 2 / (4 * fdeg K))) ^ (2 * fdeg K * r ^ 2) * (r * max 1 ((2 * fdeg K * r : ℝ) ^ 4 * dE K ^ 2 / (4 * fdeg K))) ^ (2 * fdeg K * r ^ 2) ∧ ∀ ℓ M : ℝ, 16 ≤ fdeg K → 2 ≤ ℓ → Real.log (dE K) ≤ 4 * fdeg K * ℓ → Real.log r ≤ ℓ → Real.log (fdeg K) ≤ 2

theorem **isSelfDual_hyperbolicHermForm** : (hyperbolicHermForm K).h ![1, 0] ![1, 0] = 0 ∧ IsLattice (intLat K 2) ∧ IsSelfDual (hyperbolicHermForm K) (intLat K 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFinsetTypesOfBoundedBases

theorem **exists_generators_mod_smul** {R : Type*} [CommRing R] [IsDedekindDomain R] {a : R} (ha : a ≠ 0) (r : ℕ) : ∀ (M : Type) [AddCommGroup M] [Module R M] (φ : M →ₗ[R] (Fin r → R)), Function.Injective φ → ∃ x : Fin r → M, ∀ m : M, ∃ (c : Fin r → R) (m' : M), m = ∑ i, c i • x i + a • m'

theorem **exists_finset_representatives** {α β : Type*} [Finite β] (S : Set α) (f : S → β) : ∃ V : Finset S, V.card ≤ Nat.card β ∧ ∀ s : S, ∃ t ∈ V, f t = f s

theorem **card_quotient_range_vecMulLinear** {r : ℕ} (N : Matrix (Fin r) (Fin r) (𝓞 E)) (hN : N.det ≠ 0) : Nat.card ((Fin r → 𝓞 E) ⧸ LinearMap.range N.vecMulLinear) = (Algebra.norm ℤ N.det).natAbs

theorem **natAbs_norm_le_pow** {x : 𝓞 E} {a : ℝ} (h : ∀ σ : E →+* ℂ, ‖σ (x : E)‖ ≤ a) : ((Algebra.norm ℤ x).natAbs : ℝ) ≤ a ^ finrank ℚ E

theorem **norm_det_le_pow** {r : ℕ} (M : Matrix (Fin r) (Fin r) ℂ) {b : ℝ} (hb : 0 ≤ b) (h : ∀ i k, ‖M i k‖ ≤ b) : ‖M.det‖ ≤ (r * b) ^ r

theorem **finrank_eq_two_mul_fdeg** [IsCMField E] : finrank ℚ E = 2 * fdeg E

theorem **herm_add_left** (G : Matrix (Fin r) (Fin r) E) (x x' y : Fin r → E) : herm G (x + x') y = herm G x y + herm G x' y

theorem **herm_smul_left** (G : Matrix (Fin r) (Fin r) E) (c : E) (x y : Fin r → E) : herm G (c • x) y = c * herm G x y

theorem **herm_eq_dotProduct** (G : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : herm G x y = (x ᵥ* G) ⬝ᵥ fun b => cc (y b)

theorem **gramOf_eq** (G : Matrix (Fin r) (Fin r) E) (v : Fin r → Fin r → E) : gramOf G v = Matrix.of v * G * ((Matrix.of v).map (IsCMField.complexConj E).toRingHom)ᵀ

theorem **det_of_ne_zero** {v : Fin r → Fin r → E} (hv : LinearIndependent E v) : (Matrix.of v).det ≠ 0

theorem **det_gramOf_ne_zero** {G : Matrix (Fin r) (Fin r) E} (hG : G.det ≠ 0) {v : Fin r → Fin r → E} (hv : LinearIndependent E v) : (gramOf G v).det ≠ 0

theorem **vecMul_of** (c : Fin r → E) (v : Fin r → Fin r → E) : c ᵥ* Matrix.of v = ∑ i, c i • v i

theorem **herm_vecMul_left** (G : Matrix (Fin r) (Fin r) E) (v : Fin r → Fin r → E) (c : Fin r → E) (k : Fin r) : herm G (c ᵥ* Matrix.of v) (v k) = (c ᵥ* gramOf G v) k

theorem **herm_vecMul** (G : Matrix (Fin r) (Fin r) E) (v : Fin r → Fin r → E) (c d : Fin r → E) : herm G (c ᵥ* Matrix.of v) (d ᵥ* Matrix.of v) = herm (gramOf G v) c d

theorem **exists_vecMul_eq** {v : Fin r → Fin r → E} (hv : LinearIndependent E v) (x : Fin r → E) : ∃ c, x = c ᵥ* Matrix.of v

theorem **exists_gl_mulVec** {v w : Fin r → Fin r → E} (hv : LinearIndependent E v) (hw : LinearIndependent E w) : ∃ γ : GL (Fin r) E, ∀ c : Fin r → E, (γ : Matrix (Fin r) (Fin r) E) *ᵥ (c ᵥ* Matrix.of w) = c ᵥ* Matrix.of v

theorem **coe_pairing** (v : Fin r → Fin r → E) (hv : ∀ i, v i ∈ X.L') (x : X.Λ) (c : Fin r → E) (hc : (x : Fin r → E) = c ᵥ* Matrix.of v) (k : Fin r) : ((pairing X v hv x k : 𝒪 E) : E) = (((X.NA : E) • c) ᵥ* gramOf X.Gm v) k

theorem **pairing_injective** {v : Fin r → Fin r → E} (hv : ∀ i, v i ∈ X.L') (hi : LinearIndependent E v) : Function.Injective (pairing X v hv)

theorem **smul_vecMul_mem** {v : Fin r → Fin r → E} (hv : ∀ i, v i ∈ X.L') (o : Fin r → 𝒪 E) : (X.NC : E) • ((fun i => (o i : E)) ᵥ* Matrix.of v) ∈ X.Λ

theorem **exists_mem_of_pairing** {v w : Fin r → Fin r → E} (hv : ∀ i, v i ∈ X.L') (hw : ∀ i, w i ∈ X.L') (hvi : LinearIndependent E v) (hwi : LinearIndependent E w) (A : Matrix (Fin r) (Fin r) (𝒪 E)) (hvA : gramOf X.Gm v = A.map (algebraMap (𝒪 E) E)) (hwA : gramOf X.Gm w = A.map (algebraMap (𝒪 E) E)) (h : ∀ x : X.Λ, pairing X w hw x − pairing X v hv x ∈ LinearMap.range (((X.NA * X.NC * X.g : ℕ) : 𝒪 E) • A).vecMulLinear) : ∃ γ ∈ X.Γ, ∀ i, v i = (γ : Matrix (Fin r) (Fin r) E) *ᵥ w i

theorem **exists_finset_boundedBasis_of_gram** (X : DataX E r) {B : ℝ} (hB : 0 ≤ B) (A : Matrix (Fin r) (Fin r) (𝒪 E)) : ∃ V : Finset (Fin r → Fin r → E), (∀ v ∈ V, IsBoundedBasis X B v) ∧ (∀ v, IsBoundedBasis X B v → gramOf X.Gm v = A.map (algebraMap (𝒪 E) E) → ∃ γ ∈ X.Γ, ∃ w ∈ V, ∀ i, v i = (γ : Matrix (Fin r) (Fin r) E) *ᵥ w i) ∧ (V.card : ℝ) ≤ ((X.NA : ℝ) * X.NC * X.g * r * B) ^ (2 * fdeg E * r ^ 2)

theorem **exists_finset_boundedBasis** (X : DataX E r) {B : ℝ} (hB : 0 ≤ B) : ∃ V : Finset (Fin r → Fin r → E), (∀ v ∈ V, IsBoundedBasis X B v) ∧ (∀ v, IsBoundedBasis X B v → ∃ γ ∈ X.Γ, ∃ w ∈ V, ∀ i, v i = (γ : Matrix (Fin r) (Fin r) E) *ᵥ w i) ∧ (V.card : ℝ) ≤ (1 + 2 * B) ^ (2 * fdeg E * r ^ 2) * ((X.NA : ℝ) * X.NC * r * B) ^ (2 * fdeg E * r ^ 2) * (X.g : ℝ) ^ (2 * fdeg E * r ^ 2)

theorem **exists_finset_types_of_bounded_bases** (X : DataX E r) {B : ℝ} (hB : 1 ≤ B) : ∃ V : Finset (Fin r → Fin r → E), (∀ v ∈ V, (∀ i, v i ∈ X.L') ∧ LinearIndependent E v ∧ ∀ (τ : E →+* ℂ) (i k : Fin r), ‖τ (herm X.Gm (v i) (v k))‖ ≤ B) ∧ (∀ v : Fin r → Fin r → E, (∀ i, v i ∈ X.L') → LinearIndependent E v → (∀ (τ : E →+* ℂ) (i k : Fin r), ‖τ (herm X.Gm (v i) (v k))‖ ≤ B) → ∃ γ ∈ X.Γ, ∃ w ∈ V, ∀ i, v i = Matrix.mulVec (γ : Matrix (Fin r) (Fin r) E) (w i)) ∧ (V.card : ℝ) ≤ (1 + 2 * B) ^ (2 * fdeg E * r ^ 2) * ((X.NA : ℝ) * X.NC * r * B) ^ (2 * fdeg E * r ^ 2) * (X.g : ℝ) ^ (2 * fdeg E * r ^ 2) ∧ (V.card : ℝ) ≤ (3 * (X.NA : ℝ) * X.NC * X.g * r * B ^ 2) ^ (2 * fdeg E * r ^ 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFirstMinimumLowerBoundNearMax

theorem **transvMat_natCast_smul** {r : ℕ} (G : Matrix (Fin r) (Fin r) E) (v : Fin r → E) (n : ℕ) (α : E) : transvMat G ((n : E) • v) α = transvMat G v ((n : E) ^ 2 * α)

theorem **exists_transvection_level** {r : ℕ} (X : DataX E r) (la : Wt X.Ψ r) : ∃ M : ℝ, 0 < M ∧ ∀ v ∈ X.L', herm X.Gm v v = 0 → ∀ b : E, cc b = b → IsIntegral ℤ b → ∃ γ ∈ X.Γ, X.χ la γ = 1 ∧ ∀ τ : X.Ψ, (γ : Matrix (Fin r) (Fin r) E).map τ.1 = X.flowMat v τ ((M : ℂ) * τ.1 b)

theorem **exists_firstMinimum_lower_bound_near_max** {r : ℕ} (hr : 0 < r) (X : DataX E r) (la : Wt X.Ψ r) : ∃ ε : ℝ, 0 < ε ∧ ∀ P ∈ X.SS la, ∀ z ∈ X.dom, (∀ x ∈ X.dom, X.pnorm la P x ≤ 2 * X.pnorm la P z) → 0 < X.pnorm la P z → ε ≤ X.la1 z

theorem **exists_finset_rmul_near_one** {r : ℕ} (X : Data E r) {K : Set (Frame X.Ψ r)} (hK : IsCompact K) (hKd : K ⊆ X.dom) {δ : ℝ} (hδ : 0 < δ) : ∃ N : Finset (Frame X.Ψ r), (∀ y ∈ N, y ∈ K) ∧ ∀ x ∈ K, ∃ A : Frame X.Ψ r, (∀ τ a b, ‖A τ a b − (if a = b then 1 else 0)‖ ≤ δ) ∧ rmul x A ∈ N

theorem **continuousOn_rmul_bOf** {r : ℕ} (X : Data E r) : ContinuousOn (fun x => rmul x (FrameDegreeData.bOf X x)) X.dom

theorem **ss_finite_dimensional** (hM2 : MahlerCompactnessCriterion (E

theorem **span_restrict_ss_eq** {r : ℕ} (X : Data E r) (la : Wt X.Ψ r) {f : X.dom → ℂ} (hf : f ∈ Submodule.span ℂ ((fun P : Frame X.Ψ r → ℂ => X.dom.domRestrict P) '' X.SS la)) : ∃ P ∈ X.SS la, X.dom.domRestrict P = f

theorem **rank_span_restrict_ss_le** {r : ℕ} (X : Data E r) (la : Wt X.Ψ r) (N : Finset (Frame X.Ψ r)) (hN : ∀ y ∈ N, y ∈ X.dom) {C : ℝ} (hb : ∀ P ∈ X.SS la, ∀ x ∈ X.dom, ∃ y ∈ N, X.pnorm la P x ≤ C * X.pnorm la P y) : Module.rank ℂ (Submodule.span ℂ ((fun P : Frame X.Ψ r → ℂ => X.dom.domRestrict P) '' X.SS la)) ≤ N.card

theorem **ss_card_le_of_linearIndependent** (hM2 : MahlerCompactnessCriterion (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFirstPartialLawSeriesOfFrames

theorem **ren_copy** (m : ℕ) {n n' : ℕ} (r : Fin n → Fin n') (i : Fin n) : (jetRen (S

theorem **subst_psIn** (m : ℕ) {n n' : ℕ} (r : Fin n → Fin n') (g : Fin d → MvPowerSeries (Fin n × Fin d) S) : jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (g p.2)) = (jetRen d m r).comp (jetSubstP m (fun p : Fin 1 × Fin d => psTrunc m (g p.2)))

theorem **pt_law** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (r : Fin 2 → Fin n) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (F p.2))) ≫ x m = jptG t x0 d x m (r 0) * jptG t x0 d x m (r 1)

theorem **th_pt** {T : Over X0} (a b c : T ⟶ t.G) (I : thK) : (lift a (lift b c)).left ≫ thφ t.G (x3 t.G) I = (psum t.G ![a, b, c] I.down).left

theorem **jetPt_trunc** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (i : Fin n) : homAt x0 (jetTrunc : Jet S (Fin n × Fin d) (m + 1) →ₐ[S] _) ≫ jptG t x0 d x (m + 1) i = jptG t x0 d x m i

theorem **threePts_trunc** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) (i j k : Fin 3) : (ρn x0 d 3 m).left ≫ (threePts (x

theorem **exists_firstPartialLaw_series_of_frames** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F

act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ ph1 : PS S 3 d, ∀ m, ∃ u : (Jet S (Fin 3 × Fin d) m)ˣ, (u : Jet S (Fin 3 × Fin d) m) = psJet m ph1 ∧ LB.Gen.Corr t.PP.L _ _ _ _ (LB.pullGen t.PP.L (thφ t.G (x3 t.G)) thε t.cub.t (lift (jptG t x0 d x m (0 : Fin 3)) (lift (jptG t x0 d x m (1 : Fin 3)) (jptG

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFirstPartialLawSeriesOfFramesStatement

theorem **_root_.GlobalTheory.T.exists_firstPartialLaw_series_of_frames_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ ph1 : PS S 3 d, ∀ m, ∃ u : (Jet S (Fin 3 × Fin d) m)ˣ, (u : Jet S (Fin 3 × Fin d) m) = psJet m ph1 ∧ LB.Gen.Corr t.PP.L _ _ _ _ (LB.pullGen t.PP.L (thφ t.G (x3 t.G)) thε t.cub.t (lift (jptG t x0 d x m (0 : Fin 3)) (lift (jptG t x0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFlatUniversalTorsionTuple

theorem **exists_power** {X : Scheme.{0}} (Q : Over X) [IsAffineHom Q.hom] [Flat Q.hom] (J : Type) [Finite J] : ∃ (R : Over X) (p : J → (R ⟶ Q)), IsAffineHom R.hom ∧ Flat R.hom ∧ ∀ (T : Over X) (u : J → (T ⟶ Q)), ∃ v : T ⟶ R, ∀ i, v ≫ p i = u i

theorem **exists_point** {X : Scheme.{0}} (R : Over X) [IsAffineHom R.hom] [Flat R.hom] (S : Type) [CommRing S] (x0 : Sp S ⟶ X) : ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (w : Over.mk (spMap φ ≫ x0) ⟶ R), φ.Flat ∧ ∀ (T : Type) [CommRing T] (ψ : S →+* T) (u : Over.mk (spMap ψ ≫ x0) ⟶ R), ∃ χ : B →+* T, χ.comp φ = ψ ∧ u.left = spMap χ ≫ w.left

theorem **exists_kernel** {X : Scheme.{0}} (A : Over X) [GrpObj A] (k : ℕ) [IsFinite (mulA A k).left] [Flat (mulA A k).left] : ∃ (Q : Over X) (j : Q ⟶ A), IsAffineHom Q.hom ∧ Flat Q.hom ∧ j ^ k = 1 ∧ ∀ (T : Over X) (u : T ⟶ A), u ^ k = 1 → ∃ v : T ⟶ Q, v ≫ j = u

theorem **exists_tuple** {X : Scheme.{0}} (A : Over X) [GrpObj A] (k : ℕ) [IsFinite (mulA A k).left] [Flat (mulA A k).left] (S : Type) [CommRing S] (x0 : Sp S ⟶ X) (J : Type) [Finite J] : ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (w : J → (Over.mk (spMap φ ≫ x0) ⟶ A)), φ.Flat ∧ (∀ i, w i ^ k = 1) ∧ ∀ (T : Type) [CommRing T] (ψ : S →+* T) (u : J → (Over.mk (spMap ψ ≫ x0) ⟶ A)), (∀ i, u i ^ k = 1) → ∃ χ : B →+* T, χ.comp φ = ψ ∧ ∀ i, (u i).left = spMap χ ≫ (w i).left

theorem **exists_flat_universal_torsion_tuple** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X) (k : ℕ), 0 < k → ∀ (J : Type) [Fintype J], ∃ (B : Type) (_ : CommRing B) (φ : S →+* B) (w : J → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.Flat ∧ (∀ i, w i ^ k = 1) ∧ ∀ (T : Type) [CommRing T] (ψ : S →+* T) (u : J → (ptO Xo (spMap ψ ≫ x0) ⟶ Xo.univ.t.G)), (∀ i, u i ^ k = 1) → ∃ χ : B →+* T, χ.comp φ = ψ ∧ ∀ i, (u i).left = spMap χ ≫ (w i).left

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFramedPointBaseChangeAlgHom

theorem **spMap_spMap** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

theorem **dualMap_comp_algebraMap** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) : (dualMap ψ).comp (algebraMap S (DualNumber S)) = (algebraMap T (DualNumber T)).comp ψ

theorem **fstHom_comp_dualMap** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) : (TrivSqZeroExt.fstHom T T T).toRingHom.comp (dualMap ψ) = ψ.comp (TrivSqZeroExt.fstHom S S S).toRingHom

theorem **dualMap_epsMul** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) (a : S) : (dualMap ψ).comp (epsMul a) = (epsMul (ψ a)).comp (dualMap ψ)

theorem **isTanAt_pull** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) {x : Sp S ⟶ Xo.X} {v : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hv : IsTanAt x v) : IsTanAt (spMap ψ ≫ x) (spMap (dualMap ψ) ≫ v)

theorem **isEigAt_pull** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (τ : E →+* ℂ) {v : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hv : IsEigAt c τ v) : IsEigAt c τ (spMap (dualMap ψ.toRingHom) ≫ v)

theorem **exists_frMap** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S), ∃ z : FrPt Xo c T, FrMap ψ y z

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFramedPointBaseChangeAlgHomStatement

theorem **_root_.GlobalTheory.T.exists_framedPoint_baseChange_algHom** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S), ∃ z : FrPt Xo c T, FrMap ψ y z

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFramesCubeDefectAlongChartOverRing

theorem **nilpotent_of_le** (m : ℕ) (I : Ideal (Jet S Ix m)) (h : ∀ p : MvPolynomial Ix S, Ideal.Quotient.mk _ p ∈ I → p ∈ orig S Ix) : IsNilpotent I

theorem **trunc_surj** (m : ℕ) : Function.Surjective ((jetTrunc : Jet S Ix (m + 1) →ₐ[S] Jet S Ix m) : Jet S Ix (m + 1) →+* Jet S Ix m)

theorem **trunc_nil** (m : ℕ) : IsNilpotent (RingHom.ker ((jetTrunc : Jet S Ix (m + 1) →ₐ[S] Jet S Ix m) : Jet S Ix (m + 1) →+* Jet S Ix m))

theorem **aug_surj** (m : ℕ) : Function.Surjective ((jetAug : Jet S Ix m →ₐ[S] S) : Jet S Ix m →+* S)

theorem **aug_nil** (m : ℕ) : IsNilpotent (RingHom.ker ((jetAug : Jet S Ix m →ₐ[S] S) : Jet S Ix m →+* S))

theorem **aug_axis0** (m : ℕ) : (jetAug : Jet S (Fin 1 × Fin d) m →ₐ[S] S).comp (axis0 d m) = jetAug

theorem **aug_axis1** (m : ℕ) : (jetAug : Jet S (Fin 1 × Fin d) m →ₐ[S] S).comp (axis1 d m) = jetAug

theorem **trunc_axis0** (m : ℕ) : (jetTrunc : Jet S (Fin 1 × Fin d) (m + 1) →ₐ[S] _).comp (axis0 d (m + 1)) = (axis0 d m).comp (jetTrunc : Jet S (Fin 2 × Fin d) (m + 1) →ₐ[S] _)

theorem **trunc_axis1** (m : ℕ) : (jetTrunc : Jet S (Fin 1 × Fin d) (m + 1) →ₐ[S] _).comp (axis1 d (m + 1)) = (axis1 d m).comp (jetTrunc : Jet S (Fin 2 × Fin d) (m + 1) →ₐ[S] _)

theorem **aug_trunc** (m : ℕ) : (jetAug : Jet S Ix m →ₐ[S] S).comp (jetTrunc : Jet S Ix (m + 1) →ₐ[S] _) = jetAug

theorem **cancel₀** {U : Type} [CommGroup U] (a b : U) : a * b⁻¹ * a⁻¹ * b = 1

theorem **cancel₁** {U : Type} [CommGroup U] (a b : U) : a * a⁻¹ * b⁻¹ * b = 1

theorem **exists_norm** {T : Over X0} (p q : T ⟶ t.G) (s₁ s₂ s₀ : T ⟶ T) (h₁ : s₁ ≫ (p ∗ q) = p) (h₂ : s₂ ≫ (p ∗ q) = q) (h₀ : s₀ ≫ (p ∗ q) = 1) (g₁ : LB.Gen (LB.LinComb t.PP.L (fun _ : PUnit.{1} => (p ∗ q).left) (fun _ => 1))) (hg₁ : g₁.IsGen) : ∃ e : LB.Gen (LB.LinComb t.PP.L (cφG t ![p, q]) (cε 2)), e.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ T.left) (z : Z) (J), Good t.PP.L z J → val e r z J = val g₁ r z J ∗ (val g₁ (r ≫ s₁.left) z J)⁻¹ ∗ (val g₁ (r ≫ s₂.left) z J)⁻¹ ∗ val g₁ (r ≫ s₀.left) z J

theorem **exists_tower** {Ix : Type} (P : ∀ m, overAt x0 (Jet S Ix m) ⟶ t.G) (h0 : Formalism.L2.Pj.homAt x0 (jetAug : Jet S Ix 0 →ₐ[S] S) ≫ P 0 = 1) (htr : ∀ m, Formalism.L2.Pj.homAt x0 (jetTrunc : Jet S Ix (m + 1) →ₐ[S] Jet S Ix m) ≫ P (m + 1) = P m) : ∃ g₁ : ∀ m, LB.Gen (LB.LinComb t.PP.L (fun _ : PUnit.{1} => (P m).left) (fun _ => 1)), (∀ m, (g₁ m).IsGen) ∧ ∀ m (Z : Scheme.{0}) (r : Z ⟶ (overAt x0 (Jet S Ix m)).left) (z : Z) (J), Good t.PP.L z J → val (g₁ (m + 1)) (r ≫ (Formalism.L2.Pj.homAt x0 (jetTrunc : Jet S Ix (m + 1) →ₐ[S] Jet S Ix m)).left) z J = val (g₁ m) r z J

theorem **exists_frames** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∃ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) ∧ FrTrunc t x0 d x e ∧ FrAxes t x0 d x e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFramesCubeDefectAlongChartOverRingStatement

theorem **_root_.GlobalTheory.T.exists_frames_cubeDefect_along_chart_over_ring** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∃ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) ∧ FrTrunc t x0 d x e ∧ FrAxes t x0 d x e

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFramesOfFormalChart

theorem **exists_frames_of_chart** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (G : FGLaw S d) (A : Germ.OAct (𝒪 E) G) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), IsChartG t x0 d G.F A.act tg uvec x → ∃ (e : ∀ m, FrTyG t x0 d x m) (wP : PrimTyG t) (ph1 ph2 : PS S 3 d) (sy : PS S 2 d) (ad : 𝒪 E → PS S 2 d), IsFrameG t x0 d x G.F A.act ph1 ph2 sy ad e ∧ IsPrimG t wP ∧ FrTrunc t x0 d x e ∧ FrAxes t x0 d x e ∧ (∀ a, MvPowerSeries.constantCoeff (ad a) = 1) ∧ Rig0G t x0 d x A.act e wP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsFramesOfFormalChartStatement

theorem **_root_.GlobalTheory.T.exists_frames_of_formal_chart** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (G : FGLaw S d) (A : Germ.OAct (𝒪 E) G) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), IsChartG t x0 d G.F A.act tg uvec x → ∃ (e : ∀ m, FrTyG t x0 d x m) (wP : PrimTyG t) (ph1 ph2 : PS S 3 d) (sy : PS S 2 d) (ad : 𝒪 E → PS S 2 d), IsFrameG t x0 d x G.F A.act ph1 ph2 sy ad e ∧ IsPrimG t wP ∧ FrTrunc t x0 d x e ∧ FrAxes t x0 d x e ∧ (∀ a, MvPowerSeries.constantCoeff (ad a) = 1) ∧ Rig0G t x0 d x A.act e wP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsGenLiftOfIsNilpotent

theorem **exists_ratio** (hL : L.IsLB) (p : Z ⟶ Y) (g : LB.Gen (LB.LinComb L φ' ε')) (hg : g.IsGen) (hF : FormEq (fun k => p ≫ φ k) ε φ' ε') (u : LB.Sec (LB.LinComb L φ ε)) : ∃ h : Γ(Z, ⊤), ∀ y J, Good L y J → sval u p y J = gs (𝟙 Z) y ⊤ h ∗ (val g (𝟙 Z) y J).val

theorem **ratioIdeal_eq_top** [IsAffine Y] [IsAffine Z] (hL : L.IsLB) (p : Z ⟶ Y) (hp : Function.Surjective p.appTop.hom) (g : LB.Gen (LB.LinComb L φ' ε')) (hg : g.IsGen) (hF : FormEq (fun k => p ≫ φ k) ε φ' ε') : ratioIdeal p hp φ ε g = ⊤

theorem **exists_gen_lift** [IsAffine Y] [IsAffine Z] (hL : L.IsLB) (p : Z ⟶ Y) (hp : Function.Surjective p.appTop.hom) (hp' : Function.Surjective p) (g : LB.Gen (LB.LinComb L φ' ε')) (hg : g.IsGen) (hF : FormEq (fun k => p ≫ φ k) ε φ' ε') : ∃ g' : LB.Gen (LB.LinComb L φ ε), g'.IsGen ∧ LB.Gen.Corr L _ _ _ _ (LB.pullGen L φ ε g' p) g

theorem **surjective_appTop_spMap** {A B : Type} [CommRing A] [CommRing B] (π : A →+* B) (hπ : Function.Surjective π) : Function.Surjective (spMap π).appTop.hom

theorem **surjective_spMap** {A B : Type} [CommRing A] [CommRing B] (π : A →+* B) (hπ : Function.Surjective π) (hn : IsNilpotent (RingHom.ker π)) : Function.Surjective (spMap π)

theorem **exists_gen_lift_of_isNilpotent** : ∀ (X : Scheme.{0}) (L : LB X), L.IsLB → ∀ (A B : Type) [CommRing A] [CommRing B] (π : A →+* B), Function.Surjective π → IsNilpotent (RingHom.ker π) → ∀ (K : Type) [Fintype K] (φ : K → (Sp A ⟶ X)) (ε : K → ℤ) (K' : Type) [Fintype K'] (φ' : K' → (Sp B ⟶ X)) (ε' : K' → ℤ) (g : LB.Gen (LB.LinComb L φ' ε')), g.IsGen → (∀ (U : Type) [CommGroup U] (h : (Sp B ⟶ X) → U), ∏ k, h (spMap π ≫ φ k) ^ ε k = ∏ k, h (φ' k) ^ ε' k) → ∃ g' : LB.Gen (LB.LinComb L φ ε), g'.IsGen ∧ LB.Gen.Corr L _ _ _ _ (LB.pullGen L φ ε g' (spMap π)) g

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsGeneratorJetReading

theorem **jet_point_eq** {K : Type} [Field K] {I : Type} {m : ℕ} (p : PrimeSpectrum (Jet K I m)) : p = IsLocalRing.closedPoint (Jet K I m)

theorem **res_res'** {X : Scheme.{0}} {U V W : X.Opens} (h : V ≤ U) (h' : W ≤ V) (a : Γ(X, U)ˣ) : res h' (res h a) = res (h'.trans h) a

theorem **res_id'** {X : Scheme.{0}} {U : X.Opens} (h : U ≤ U) (a : Γ(X, U)ˣ) : res h a = a

theorem **res_pullU** {X Y : Scheme.{0}} (f : Y ⟶ X) (U : X.Opens) {W : Y.Opens} (h : W ≤ f ⁻¹ᵁ U) (h' : ⊤ ≤ f ⁻¹ᵁ U) (a : Γ(X, U)ˣ) (b : Γ(Y, ⊤)ˣ) (hb : (f.appLE U ⊤ h').hom a.val = b.val) : res h (pullU f U a) = res (le_top : W ≤ ⊤) b

theorem **exists_gen_of_units** {X Y : Scheme.{0}} (L : LB X) (hY : ∀ p q : Y, p = q) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (u : K → L.ι → Γ(Y, ⊤)ˣ) (hu : ∀ k j j' (h : ⊤ ≤ φ k ⁻¹ᵁ (L.U j ⊓ L.U j')), (u k j).val = ((φ k).appLE _ ⊤ h).hom (L.g j j').val ∗ (u k j').val) : ∃ τ : LB.Gen (LB.LinComb L φ ε), τ.IsGen ∧ ∀ (j : K → L.ι) (h : ⊤ ≤ LB.wchart L φ j), (Y.presheaf.map (homOfLE h).op).hom (τ.s j).val = (∏ k, u k (j k) ^ ε k).val

theorem **jetOf_one** {d : ℕ} (m : ℕ) : jetOf m (fun _ : Fin d → ℂ => (1 : ℂ)) = 1

theorem **jetOf_pow** {d : ℕ} (m : ℕ) {f : (Fin d → ℂ) → ℂ} (hf : ContDiffAt ℂ ω f 0) (k : ℕ) : jetOf m (fun z => f z ^ k) = jetOf m f ^ k

theorem **contDiffAt_zpow'** {d : ℕ} {f : (Fin d → ℂ) → ℂ} (hf : ContDiffAt ℂ ω f 0) (h0 : f 0 ≠ 0) (k : ℤ) : ContDiffAt ℂ ω (fun z => f z ^ k) 0

theorem **jetOf_zpow** {d : ℕ} (m : ℕ) {f : (Fin d → ℂ) → ℂ} (hf : ContDiffAt ℂ ω f 0) (h0 : f 0 ≠ 0) (u : (Jet ℂ (Fin 1 × Fin d) m)ˣ) (hu : u.val = jetOf m f) (k : ℤ) : jetOf m (fun z => f z ^ k) = (u ^ k).val

theorem **jetOf_prod** {d : ℕ} (m : ℕ) {K : Type} (s : Finset K) {f : K → (Fin d → ℂ) → ℂ} (hf : ∀ k, ContDiffAt ℂ ω (f k) 0) : jetOf m (fun z => ∏ k ∈ s, f k z) = ∏ k ∈ s, jetOf m (f k)

theorem **appLE_spMap_jetPt** {d : ℕ} {Y : Scheme.{0}} (e : (Fin d → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (m : ℕ) {R : Type} [CommRing R] (r : Jet ℂ (Fin 1 × Fin d) m →+* R) (U : Y.Opens) (hU : e 0 cpt ∈ U) (h : ⊤ ≤ (spMap r ≫ jetPt e hol m) ⁻¹ᵁ U) (a : Γ(Y, U)) : ((spMap r ≫ jetPt e hol m).appLE U ⊤ h).hom a = toΓ R (r (jetOf m fun z => valAt' (e z) U a))

theorem **mem_of_spMap_jetPt** {d : ℕ} {Y : Scheme.{0}} (e : (Fin d → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (m : ℕ) {I : Type} (r : Jet ℂ (Fin 1 × Fin d) m →+* Jet ℂ I m) (U : Y.Opens) (h : ⊤ ≤ (spMap r ≫ jetPt e hol m) ⁻¹ᵁ U) : e 0 cpt ∈ U

theorem **contDiffAt_fr** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) {n : ℕ} (Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (j : L.L.ι) (h : (Φ.e (Λ (colsOf n g 0))).left cpt ∈ L.L.U j) : ContDiffAt ℂ ω (fun v => Φ.fr j (Λ (colsOf n g v))) 0

theorem **exists_gen_jetRd** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (n m : ℕ) (K : Type) [Fintype K] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), (∀ k, Φ.JetMap m (φ k) (Λ k)) → ∃ τ : LB.Gen (LB.LinComb L.L φ ε), τ.IsGen ∧ Φ.JetRd m φ ε τ Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsGeneratorJetReadingStatement

theorem **_root_.GlobalTheory.T.exists_generator_jet_reading** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (n m : ℕ) (K : Type) [Fintype K] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), (∀ k, Φ.JetMap m (φ k) (Λ k)) → ∃ τ : LB.Gen (LB.LinComb L.L φ ε), τ.IsGen ∧ Φ.JetRd m φ ε τ Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsGeneratorsLattice

theorem **finrank_fibre** (O M k : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] [Field k] [Algebra O k] : Module.finrank k (k ⊗[O] M) = Module.finrank O M

theorem **exists_generators_mod** (O M : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (m : O) [Finite (O ⧸ Ideal.span {m})] (r : ℕ) (hr : Module.finrank O M = r) : ∃ e : Fin r → M, ∀ x : M, ∃ (c : Fin r → O) (y : M), x = ∑ k, c k • e k + m • y

theorem **exists_generators_lattice** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ.FG) (h : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) : ∃ e : Fin n → Λ, ∀ x : Λ, ∃ (c : Fin n → 𝒪 E) (y : Λ), x = ∑ k, c k • e k + (5 : 𝒪 E) • y

theorem **act_add_vec** {dim : (E →+* ℂ) → ℕ} (α : 𝒪 E) (v w : Vsp dim) : act α (v + w) = act α v + act α w

theorem **act_smul_vec** {dim : (E →+* ℂ) → ℕ} (α : 𝒪 E) (c : ℂ) (v : Vsp dim) : act α (c • v) = c • act α v

theorem **five_smul_mem** (T : PT E) {y : Vsp T.dim} (hy : y ∈ T.L) : (5 : ℂ) • y ∈ T.L

theorem **finite_torsion** (T : PT E) : {v : Vsp T.dim | v ∈ ZSpan.fundamentalDomain T.b ∧ ((5 : ℂ) ^ 2) • v ∈ T.L}.Finite

theorem **exists_generators_period** {r : ℕ} {X : Data E r} {x : Frame X.Ψ r} (A : TorusAt X x) : ∃ e : Fin A.T.r → Vsp A.T.dim, (∀ j, e j ∈ A.T.L) ∧ ∀ v ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, v = ∑ j, act (c j) (e j) + (5 : ℂ) • y

theorem **tuple_sum_pos** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r) (x : Frame X.Ψ r) (A : TorusAt X.toData x), 0 < ∑' (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsHolomorphicGeneratorReadingAutomorphy

theorem **differentiableAt_prod** {E K : Type} [NormedAddCommGroup E] [NormedSpace ℂ E] (s : Finset K) (f : K → E → ℂ) (z : E) (h : ∀ k, DifferentiableAt ℂ (f k) z) : DifferentiableAt ℂ (fun w => ∏ k ∈ s, f k w) z

private theorem **preimage_eq_univ_iff_forall_mem** {W : Scheme.{0}} (x : Sp ℂ ⟶ W) (U : W.Opens) : ⊤ ≤ x ⁻¹ᵁ U ↔ ∀ p, x p ∈ U

theorem **top_le_wchart_iff** (φ : K → (Y ⟶ X)) (j : K → L.ι) (x : Sp ℂ ⟶ Y) : ⊤ ≤ x ⁻¹ᵁ LB.wchart L φ j ↔ ∀ k, ⊤ ≤ (x ≫ φ k) ⁻¹ᵁ L.U (j k)

theorem **top_le_inf** {x : Sp ℂ ⟶ Y} {U V : Y.Opens} (h : ⊤ ≤ x ⁻¹ᵁ U) (h' : ⊤ ≤ x ⁻¹ᵁ V) : ⊤ ≤ x ⁻¹ᵁ (U ⊓ V)

theorem **val_prod** (x : Sp ℂ ⟶ Y) (U : Y.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (u : K → Γ(Y, U)ˣ) : valAt' x U (∏ k, u k).val = ∏ k, valAt' x U (u k).val

private theorem **valAt'_app_hom_eq_valAt'_comp** {W V : Scheme.{0}} (x : Sp ℂ ⟶ V) (ψ : V ⟶ W) (U : W.Opens) (f : Γ(W, U)) : valAt' x (ψ ⁻¹ᵁ U) ((ψ.app U).hom f) = valAt' (x ≫ ψ) U f

private theorem **valAt'_pullU_eq_valAt'_comp** {W V : Scheme.{0}} (x : Sp ℂ ⟶ V) (ψ : V ⟶ W) (U : W.Opens) (u : Γ(W, U)ˣ) : valAt' x (ψ ⁻¹ᵁ U) (pullU ψ U u).val = valAt' (x ≫ ψ) U u.val

private theorem **valAt'_res_unit_eq** {W : Scheme.{0}} (x : Sp ℂ ⟶ W) {U V : W.Opens} (hVU : V ≤ U) (h : ⊤ ≤ x ⁻¹ᵁ V) (u : Γ(W, U)ˣ) : valAt' x V (res hVU u).val = valAt' x U u.val

private theorem **valAt'_unit_zpow'** {W : Scheme.{0}} (x : Sp ℂ ⟶ W) (U : W.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (u : Γ(W, U)ˣ) (n : ℤ) : valAt' x U (u ^ n).val = valAt' x U u.val ^ n

theorem **val_linComb_g** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (p : Sp ℂ ⟶ Y) (j j' : K → L.ι) (h : ⊤ ≤ p ⁻¹ᵁ (LB.wchart L φ j ⊓ LB.wchart L φ j')) : valAt' p (LB.wchart L φ j ⊓ LB.wchart L φ j') ((LB.LinComb L φ ε).g j j').val = ∏ k, valAt' (p ≫ φ k) (L.U (j k) ⊓ L.U (j' k)) (L.g (j k) (j' k)).val ^ ε k

theorem **val_isGen** (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) (p : Sp ℂ ⟶ Y) (j j' : K → L.ι) (h : ⊤ ≤ p ⁻¹ᵁ (LB.wchart L φ j ⊓ LB.wchart L φ j')) : valAt' p (LB.wchart L φ j) (t.s j).val = (∏ k, valAt' (p ≫ φ k) (L.U (j k) ⊓ L.U (j' k)) (L.g (j k) (j' k)).val ^ ε k) * valAt' p (LB.wchart L φ j') (t.s j').val

theorem **exists_chart** (hL : L.IsLB) (φ : K → (Y ⟶ X)) (x : Sp ℂ ⟶ Y) : ∃ j : K → L.ι, ⊤ ≤ x ⁻¹ᵁ LB.wchart L φ j

theorem **top_le_chart** {q : Sp ℂ ⟶ Z} {p : K → Fin g → ℂ} (hq : ∀ k, q ≫ φ k = (Φ.e (p k)).left) {j : K → L.L.ι} (hj : ⊤ ≤ q ⁻¹ᵁ LB.wchart L.L φ j) (k : K) : ⊤ ≤ (Φ.e (p k)).left ⁻¹ᵁ L.L.U (j k)

theorem **prod_fr_ne** {q : Sp ℂ ⟶ Z} {p : K → Fin g → ℂ} (hq : ∀ k, q ≫ φ k = (Φ.e (p k)).left) {j : K → L.L.ι} (hj : ⊤ ≤ q ⁻¹ᵁ LB.wchart L.L φ j) : ∏ k, Φ.fr (j k) (p k) ^ ε k ≠ 0

theorem **genRd_of_chart** (ht : t.IsGen) {q : Sp ℂ ⟶ Z} {p : K → Fin g → ℂ} (hq : ∀ k, q ≫ φ k = (Φ.e (p k)).left) {j' : K → L.L.ι} (hj' : ⊤ ≤ q ⁻¹ᵁ LB.wchart L.L φ j') : Φ.GenRd φ ε t q p (valAt' q (LB.wchart L.L φ j') (t.s j').val / ∏ k, Φ.fr (j' k) (p k) ^ ε k)

theorem **genRd_eq** {q : Sp ℂ ⟶ Z} {p : K → Fin g → ℂ} {u : ℂ} (hu : Φ.GenRd φ ε t q p u) {j : K → L.L.ι} (hj : ⊤ ≤ q ⁻¹ᵁ LB.wchart L.L φ j) : u = valAt' q (LB.wchart L.L φ j) (t.s j).val / ∏ k, Φ.fr (j k) (p k) ^ ε k

private theorem **valAt'_unit_val_ne_zero** {W : Scheme.{0}} (x : Sp ℂ ⟶ W) (U : W.Opens) (h : ⊤ ≤ x ⁻¹ᵁ U) (u : Γ(W, U)ˣ) : valAt' x U u.val ≠ 0

theorem **exists_reading** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)), t.IsGen → ∀ (N : ℕ) (q : (Fin N → ℂ) → (Sp ℂ ⟶ Z)) (p : K → (Fin N → ℂ) → Fin g → ℂ), IsHolFam q → (∀ k, Differentiable ℂ (p k)) → (∀ z k, q z ≫ φ k = (Φ.e (p k z)).left) → ∃ v : (Fin N → ℂ) → ℂ, Differentiable ℂ v ∧ (∀ z, v z ≠ 0) ∧ (∀ z, Φ.GenRd φ ε t (q z) (fun k => p k z) (v z)) ∧ (∀ z u, Φ.GenRd φ ε t (q z) (fun k => p k z) u → u = v z) ∧ ∀ z z' (a : K → Fin g → ℂ), q z' = q z → (∀ k, a k ∈ Φ.lat ∧ p k z' = p k z + a k) → v z' = v z * ∏ k, Φ.fac (a k) (p k z) ^ ε k

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsHolomorphicGeneratorReadingAutomorphyStatement

theorem **_root_.GlobalTheory.T.exists_holomorphic_generator_reading_automorphy** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)), t.IsGen → ∀ (N : ℕ) (q : (Fin N → ℂ) → (Sp ℂ ⟶ Z)) (p : K → (Fin N → ℂ) → Fin g → ℂ), IsHolFam q → (∀ k, Differentiable ℂ (p k)) → (∀ z k, q z ≫ φ k = (Φ.e (p k z)).left) → ∃ v : (Fin N → ℂ) → ℂ, Differentiable ℂ v ∧ (∀ z, v z ≠ 0) ∧ (∀ z, Φ.GenRd φ ε t (q z) (fun k => p k z) (v z)) ∧ (∀ z u, Φ.GenRd φ ε t (q z) (fun k => p k z) u → u = v z) ∧ ∀ z z' (a : K → Fin g → ℂ), q z' = q z → (∀ k, a k ∈ Φ.lat ∧ p k z' = p k z + a k) → v z' = v z * ∏ k, Φ.fac (a k) (p k z) ^ ε k

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsInBasisSharesInContain

theorem **exists_inBasis_shares_inContain** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (ka wL wF : WeightFamilies.Wt D j) (G₀ : GlobalFamilyData (recPlusN h j hj n) ka wL wF) (Tn : ℕ) (CR c : ℝ) (S : Scal n.base), addL wF wL Tn = ka → 2 ≤ WeightFamilies.dimY D j → 11 * WeightFamilies.dimY D j ^ 2 ≤ Tn → Nonempty (ArchAt h j G₀.w wF Tn (WeightFamilies.c1 D CR Tn c j)) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → MggOnto h j CR Tn c ka → LatAt n.base j ka → (∃ β : ℝ, ShortG (recPlusN h j hj n) ((recPlusN h j hj n).wF0 wL wF) ((recPlusN h j hj n).Thc S.s0 ((recPlusN h j hj n).wF0 wL wF)) β) → ∀ Sh : Set ((recB h.toLawsCore).Cusp j wF), Submodule.span (recB h.toLawsCore).KO Sh = ⊤ → ∃ (del : ℕ) (th' : Fin del → n.base.Cusp j wF), (∀ i, th' i ∈ Sh) ∧ ∀ rz alz th0 zc rq th iet et, (∀ la, InBasis (FamilyHypotheses.refill G₀ del th' rz alz th0 zc rq t

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIntegralAuxiliaryCuspFormsSupBound

theorem **auxiliaryWeights_last_entry_pos** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} {X : PJ D B P j} {large₂ : SizeData₂ D} {H : KHyp} {C2 A2 CR CT : ℝ} {ka wL wF : WeightFamilies.Wt D j} {Tn nP nW : ℕ} {h0 : ℝ} (hP : FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0) : (∀ w2 : WeightFamilies.Wt D j, IsG2 wL wF w2 → w2.LastPos) ∧ ∀ w1 : WeightFamilies.Wt D j, IsDoubledWeightWithShift wL w1 nW → w1.LastPos

theorem **familyHypothesesI4_recPJ_of_supBound_thInt** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → 16 ≤ D.R.f → (∀ w2, IsG2 wL wF w2 → ThInt (recPJ h j hj) hinj EX w2) → (j < D.R.m → ∀ w1, IsDoubledWeightWithShift wL w1 nW → ThInt (recPJ h j hj) hinj EX w1) → FamilyHypotheses.I4 large₂ (recPJ h j hj) wL wF nW

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIntegralAuxiliaryCuspFormsSupBoundStatement

theorem **_root_.GlobalTheory.T.exists_integral_auxiliary_cuspForms_supBound** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → 16 ≤ D.R.f → (∀ w2, IsG2 wL wF w2 → ThInt (recPJ h j hj) hinj EX w2) → (j < D.R.m → ∀ w1, IsDoubledWeightWithShift wL w1 nW → ThInt (recPJ h j hj) hinj EX w1) → FamilyHypotheses.I4 large₂ (recPJ h j hj) wL wF nW

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIntegralHarmonicNeZeroAndIsHomogeneous

theorem **isHomogeneous_of_rowDeg** {T : Type} [Fintype T] {r : T → ℕ} {n : ℕ} {k : Type} [Field k] {a : T → ℕ → ℕ} {P : MvPolynomial (Letter T r n) k} (hP : RowDeg r n k a P) : P.IsHomogeneous (∑ τ, ∑ i : Fin (r τ), a τ i)

theorem **exists_harmA_ne_zero_and_isHomogeneous** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), (Dom a n → Dw.p + Dw.q ≤ n → ∃ P ∈ harmA (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIntegralHarmonicNeZeroAndIsHomogeneousStatement

theorem **_root_.GlobalTheory.T.exists_integralHarmonic_ne_zero_and_isHomogeneous** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), (Dom a n → Dw.p + Dw.q ≤ n → ∃ P ∈ harmA (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIntegralThetaCombinationNonzeroOnComponent

private lemma **card_filter_sum_eq_zero_mul_le_pow** (N : ℕ) (v : Fin N → Fin N → ℂ) (hv : ∀ c, v c c ≠ 0) (c : Fin N) : (Finset.univ.filter (fun n : Fin N → Fin (N + 1) => ∑ i, ((n i : ℕ) : ℂ) * v i c = 0)).card * (N + 1) ≤ (N + 1) ^ N

theorem **exists_box_comb_ne_zero** (N : ℕ) (v : Fin N → Fin N → ℂ) (hv : ∀ c, v c c ≠ 0) : ∃ n : Fin N → ℕ, (∀ c, n c ≤ N) ∧ ∀ c, ∑ i, (n i : ℂ) * v i c ≠ 0

theorem **exists_mem_Mlat_ne_zero** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} (X : PJ D B P j) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (w : WeightFamilies.Wt D j) (hInt : ThInt X hinj EX w) (N : ℕ) (F : Fin N → B.Cusp j w) (hF : ∀ i, F i ∈ X.thAll w) (b : ℝ) (hb : 0 ≤ b) (hFi : ∀ σ i, Real.log (B.supN j w σ (F i)) ≤ b) (σ₀ : B.KO →+* ℂ) (ρ : Fin N → B.Cusp j w → ℂ) (hadd : ∀ c G G', ρ c (G + G') = ρ c G + ρ c G') (hsmul : ∀ c (a : B.KO) G, ρ c (a • G) = σ₀ a * ρ c G) (hne : ∀ c, ρ c (F c) ≠ 0) : ∃ th ∈ B.Mlat j w, (∀ c, ρ c th ≠ 0) ∧ ∀ σ, Real.log (B.supN j w σ th) ≤ 2 * Real.log N + b + lamStar D * w.size

theorem **exists_integral_theta_nonzero_on_components** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → WeightAlgebra.DF w + 2 ≤ qq D → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D), ThInt (recPJ h j hj) hinj EX w → ∃ th ∈

(recB h.toLawsCore).Mlat j w, (recPJ h j hj).NonZeroComp w th ∧ ∀ σ, Real.log ((recB h.toLawsCore).supN j w σ th) ≤ 2 * Real.log (Fintype.card (T.U j T.σ₀).Comp) + (400 * Zb D large₂ + lamStar D) * w.size

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIntegralThetaCombinationNonzeroOnComponentStatement

theorem **_root_.GlobalTheory.T.exists_integral_theta_combination_nonzero_on_components** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → WeightAlgebra.DF w + 2 ≤ qq D → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D), ThInt (recPJ h j hj) hinj EX w → ∃ th ∈ (recB h.toLawsCore).Mlat j w, (recPJ h j hj).NonZeroComp w th ∧ ∀ σ, Real.log ((recB h.toLawsCore).supN j w σ th) ≤ 2 * Real.log (Fintype.card (T.U j T.σ₀).Comp) + (400 * Zb D large₂ + lamStar D) * w.size

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsChartGOfBasis

theorem **mk_psTrunc** {S : Type} [CommRing S] {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (f : MvPowerSeries κ S) (hf : MvPowerSeries.constantCoeff f = 0) : ChartJet.pj m f = Ideal.Quotient.mk _ (psTrunc m f − MvPolynomial.C (MvPolynomial.constantCoeff (psTrunc m f)))

theorem **jetMap_zR** {S : Type} [CommRing S] {κ κ' : Type} (m : ℕ) (φ : κ → MvPolynomial κ' S) (h) (s : κ) : (jetMap φ h : Jet S κ m →ₐ[S] Jet S κ' m) (Ideal.Quotient.mk _ (MvPolynomial.X s)) = Ideal.Quotient.mk _ (φ s)

theorem **subst_zR** {S : Type} [CommRing S] {d n : ℕ} (m : ℕ) (p : Q S d n) (s : Fin 1 × Fin d) : jetSubstP m (fun s : Fin 1 × Fin d => psTrunc m (p.1 s.2)) (Chart.zR m s) = ChartJet.pj m (p.1 s.2)

theorem **copy_zR** {S : Type} [CommRing S] {d n : ℕ} (m : ℕ) (i : Fin n) (s : Fin 1 × Fin d) : jetCopy (S

theorem **exists_isChartG_of_basis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)), (∀ ℓ, IsTanAtG t x0 (tg ℓ)) → (∀ w, IsTanAtG t x0 w → ∃! a : κ → S, IsCombAtG t x0 (m

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsHomogeneousAevalEqZero

theorem **coeff_aeval_C_mul_X** {K Ω σ : Type} [Field K] [Field Ω] [Algebra K Ω] [Fintype σ] (v : σ → Ω) (P : MvPolynomial σ K) (d : ℕ) : (MvPolynomial.aeval (fun i => Polynomial.C (v i) * Polynomial.X) P).coeff d = MvPolynomial.aeval v (MvPolynomial.homogeneousComponent d P)

theorem **exists_isHomogeneous_aeval_eq_zero** {K Ω : Type} [Field K] [Field Ω] [Algebra K Ω] {D m : ℕ} (hD : Algebra.trdeg K Ω = D) (hm : D + 2 ≤ m) (v : Fin m → Ω) : ∃ (Q : MvPolynomial (Fin m) K) (d : ℕ), Q ≠ 0 ∧ Q.IsHomogeneous d ∧ MvPolynomial.aeval v Q = 0

theorem **one_mem_sec** : (1 : Rul Xo c) ∈ sec Xo c 0 0

theorem **multiset_prod_mem_sec** {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {σ : Type} (u : σ → Rul Xo c) (hu : ∀ i, u i ∈ sec Xo c a n) (s : Multiset σ) : (s.map u).prod ∈ sec Xo c (s.card • a) (s.card • n)

theorem **aeval_mem_sec** {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {σ : Type} (u : σ → Rul Xo c) (hu : ∀ i, u i ∈ sec Xo c a n) {Q : MvPolynomial σ c.F} {d : ℕ} (hQ : Q.IsHomogeneous d) : MvPolynomial.aeval u Q ∈ sec Xo c (d • a) (d • n)

theorem **exists_algHom_evR** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (h : FromTest (K

theorem **exists_homogeneous_relation** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n m : ℕ), Module.finrank ℚ E / 2 * (Dw.p * Dw.q + Dw.p * (Dw.p − 1) / 2 + Dw.q * (Dw.q − 1) / 2) + 2 ≤ m → ∀ u : Fin m → Rul Xo c, (∀ i, u i ∈ sec Xo c a n) → ∃ (Q : MvPolynomial (Fin m) c.F) (d : ℕ), Q ≠ 0 ∧ Q.IsHomogeneous d ∧ MvPolynomial.aeval u Q = 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsIdempotentElemRangeSpMapSubset

private theorem **eq_one_iff_isUnit_of_isIdempotentElem** {R : Type} [CommRing R] {u : R} (hu : IsIdempotentElem u) : u = 1 ↔ IsUnit u

private theorem **isUnit_iff_forall_primeSpectrum_notMem** {R : Type} [CommRing R] (x : R) : IsUnit x ↔ ∀ q : PrimeSpectrum R, x ∉ q.asIdeal

theorem **exists_isIdempotentElem_range_spMap_subset** {B : Type} [CommRing B] {U : Set (Sp B)} (hU : IsClopen U) : ∃ e : B, IsIdempotentElem e ∧ ∀ (T : Type) [CommRing T] (χ : B →+* T), χ e = 1 ↔ Set.range (spMap χ) ⊆ U

theorem **exists_isIdempotentElem_torsion_eq** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (B : Type) [CommRing B] (x0 : Sp B ⟶ Xo.X) (k : ℕ), 0 < k → ∀ (a b : ptO Xo x0 ⟶ Xo.univ.t.G), a ^ k = 1 → b ^ k = 1 → ∃ e : B, IsIdempotentElem e ∧ ∀ (T : Type) [CommRing T] (χ : B →+* T), χ e = 1 ↔ spMap χ ≫ a.left = spMap χ ≫ b.left

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsNab

theorem **exists_isNab** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), ∃ (_ : StarRing (𝒪 E)) (nab : NabV Xo c), IsNab Xo c nab

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsUUOver

theorem **exists_good_pt** {Z : Scheme.{0}} (hL : L₀.IsLB) (z : Z) : ∃ J, Good L₀ z J

theorem **gs_appLE** {Z : Scheme.{0}} (r : Z ⟶ Y) (g : Y ⟶ X) (z : Z) (U : X.Opens) (W : Y.Opens) (h : W ≤ g ⁻¹ᵁ U) (hz : r z ∈ W) (a : Γ(X, U)) : gs r z W ((g.appLE U W h).hom a) = gs (r ≫ g) z U a

theorem **gs_pow** {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (W : Y.Opens) (hz : r z ∈ W) (a : Γ(Y, W)) (n : ℕ) : gs r z W (a ^ n) = gs r z W a ^ n

theorem **val_move** {w : LB.Iso L₁ (L₀.pull F)} {K : Type} [Fintype K] {φ₁ : K → (Y' ⟶ X')} {φ₀ : K → (Y ⟶ X)} {ε : K → ℤ} {a : LB.Gen (LB.LinComb L₁ φ₁ ε)} {t : LB.Gen (LB.LinComb L₀ φ₀ ε)} {h : Y' ⟶ Y} (hc : ∀ k, φ₁ k ≫ F = h ≫ φ₀ k) (H : PMaps w φ₁ φ₀ ε a t h) {Z : Scheme.{0}} (r : Z ⟶ Y') {z : Z} {J₀ J} (hJ₀ : Good L₀ z J₀) (hJ : Good L₁ z J) : val t (r ≫ h) z J₀ = (∏ k, wv w z J₀ J (r ≫ φ₁ k) ^ ε k) * val a r z J

theorem **exists_genRel** (hL₀ : L₀.IsLB) (hL₁ : L₁.IsLB) {w : LB.Iso L₁ (L₀.pull F)} (hw : w.IsGen) (h : Y' ⟶ Y) {a₀ b₀ : Y ⟶ X} {a b : Y' ⟶ X'} (ha : h ≫ a₀ = a ≫ F) (hb : h ≫ b₀ = b ≫ F) {m n : ℤ} (t : LB.Gen (LB.LinComb L₀ (K

theorem **base_comm** {x : T ⟶ A} {x₀ : T₀ ⟶ G} (hx : τ ≫ x₀.left = x.left ≫ φ) : τ ≫ T₀.hom = T.hom ≫ f

theorem **pt_iff** (hτ : τ ≫ T₀.hom = T.hom ≫ f) (x : T ⟶ A) (x₀ : T₀ ⟶ G) : τ ≫ x₀.left = x.left ≫ φ ↔ preHom (Over.homMk τ hτ : Over.mk (T.hom ≫ f) ⟶ T₀) x₀ = pushHom hp T x

theorem **pt_mul** {x y : T ⟶ A} {x₀ y₀ : T₀ ⟶ G} (hx : τ ≫ x₀.left = x.left ≫ φ) (hy : τ ≫ y₀.left = y.left ≫ φ) : τ ≫ (x₀ ∗ y₀).left = (x ∗ y).left ≫ φ

theorem **pt_pow** {x : T ⟶ A} {x₀ : T₀ ⟶ G} (hx : τ ≫ x₀.left = x.left ≫ φ) (n : ℕ) : τ ≫ (x₀ ^ n).left = (x ^ n).left ≫ φ

theorem **pt_inv** {x : T ⟶ A} {x₀ : T₀ ⟶ G} (hx : τ ≫ x₀.left = x.left ≫ φ) : τ ≫ (x₀⁻¹).left = (x⁻¹).left ≫ φ

theorem **pt_transl** {x : T ⟶ A} {x₀ : T₀ ⟶ G} (hx : τ ≫ x₀.left = x.left ≫ φ) : Ψ ≫ (translPt G x₀).left = (translPt A x).left ≫ φ

theorem **lift_comm** {m : T ⊗ A ⟶ A} {m₀ : T₀ ⊗ G ⟶ G} (hm : Ψ ≫ m₀.left = m.left ≫ φ) (h₂ : Ψ ≫ (snd T₀ G).left = (snd T A).left ≫ φ) : (lift (fst T A) m).left ≫ Ψ = Ψ ≫ (lift (fst T₀ G) m₀).left

theorem **secRel_twist** {ψ₀ : T₀ ⊗ G ⟶ T₀ ⊗ G} {m₀ : T₀ ⊗ G ⟶ G} {hm₀ : ψ₀ ≫ snd T₀ G = m₀} {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} {hm : ψ ≫ snd T A = m} (e : ψ.left ≫ Ψ = Ψ ≫ ψ₀.left) {a b : ℤ} {α₀ : LB.Gen (LB.LinComb L₀ (K

theorem **secRel_iff** (hL₀ : L₀.IsLB) (hL₁ : L₁.IsLB) (h₂ : Ψ ≫ (snd T₀ G).left = (snd T A).left ≫ φ) {a : ℤ} (n : ℕ) (hn : a = n) (s₀ : LB.Sec (LPow G L₀ a T₀)) (s : LB.Sec (LPow A L₁ a T)) : SecRel w Ψ s₀ s ↔ ∀ (i : L₀.ι) (j : L₁.ι) (W : (T ⊗ A).left.Opens) (h₀ : W ≤ (snd T A).left ⁻¹ᵁ ((LB.div (L₀.pull φ) L₁).U ((fun _ => i), j))) (h₁ : W ≤ Ψ ⁻¹ᵁ (LPow G L₀ a T₀).U (fun _ => i)) (h₂ : W ≤ (LPow A L₁ a T).U (fun _ => j)), (Ψ.appLE ((LPow G L₀ a T₀).U (fun _ => i)) W h₁).hom (s₀.f (fun _ => i)) = ((snd T A).left.appLE ((LB.div (L₀.pull φ) L₁).U ((fun _ => i), j)) W h₀).hom (Units.val (w.s ((fun _ => i), j))) ^ n ∗ ((T ⊗ A).left.presheaf.map (homOfLE h₂).op).hom (s.f (fun _ => j))

theorem **transl_snd** (x : T ⟶ A) : (translT A x).left ≫ (snd T A).left = (translPt A x).left

theorem **transl_transl** (x y : T ⟶ A) : (translT A y).left ≫ (translPt A x).left = (translPt A (x ∗ y)).left

theorem **translPt_one** : translPt A (1 : T ⟶ A) = snd T A

theorem **neg_snd** : (negT A T).left ≫ (snd T A).left = (snd T A ≫ (GrpObj.inv : A ⟶ A)).left

theorem **neg_transl** [IsCommMonObj A] (x : T ⟶ A) : (negT A T).left ≫ (translPt A x).left = (translT A x⁻¹).left ≫ (snd T A ≫ (GrpObj.inv : A ⟶ A)).left

theorem **val_rigPow** (ρ : LB.Gen (L₁.pull (eSec A))) (k : ℤ) {Z : Scheme.{0}} (r : Z ⟶ (𝟙_ (Over S')).left) (z : Z) (J) : val (rigPow A L₁ ρ k) r z J = val (L

theorem **telescope** {U : Type} [CommGroup U] (n : ℕ) (g : ℕ → U) (v : Fin n → U) (hg : g n = g 0) (h : ∏ j : Fin n, g j ∗ (g (j + 1))⁻¹ ∗ v j = 1) : ∏ j, v j = 1

theorem **sym_cancel** {U : Type} [CommGroup U] {p q r s a b c d : U} (h : p ∗ q⁻¹ ∗ a ∗ (q ∗ r⁻¹ ∗ b) ∗ (r ∗ s⁻¹ ∗ c ∗ (p ∗ s⁻¹ ∗ d)⁻¹) = 1) : a ∗ b ∗ (c ∗ d⁻¹) = 1

theorem **exists_symLift** [IsCommMonObj A] {k : ℤ} {n : ℕ} (hk : ∀ θ : ThetaGrp A L₁ k T, θ.pt ^ n = 1) {x : T ⟶ A} {x₀ : T₀ ⟶ G} (hx : τ ≫ x₀.left = x.left ≫ φ) {θ₀ : ThetaGrp G L₀ k T₀} (H : SymLift G L₀ k n x₀ θ₀) : ∃ θ : ThetaGrp A L₁ k T, SymLift A L₁ k n x θ ∧ ∀ s₀ t₀ s, IsActOf G L₀ k θ₀ s₀ t₀ → SecRel w Ψ s₀ s → ∃ t, IsActOf A L₁ k θ s t ∧ SecRel w Ψ t₀ t

theorem **exists_normMul** {ρ₀ : LB.Gen (L₀.pull (eSec G))} {ρ : LB.Gen (L₁.pull (eSec A))} {e : (𝟙_ (Over S')).left ⟶ (𝟙_ (Over S)).left} (he : eSec A ≫ φ = e ≫ eSec G) (hρ : PMaps w (K

theorem **exists_isUUOver** {S S' : Scheme.{0}} {G : Over S} [GrpObj G] [IsCommMonObj G] {A : Over S'} [GrpObj A] [IsCommMonObj A] {f : S' ⟶ S} {φ : A.left ⟶ G.left} (hp : IsGrpPullback A G f φ) {T : Over S'} {T₀ : Over S} {τ : T.left ⟶ T₀.left} {L₀ : RigLB G} {L₁ : RigLB A} {w : LB.Iso L₁.L (L₀.L.pull φ)} (hw : w.IsGen) {Ψ : (T ⊗ A).left ⟶ (T₀ ⊗ G).left} (h₁ : Ψ ≫ (fst T₀ G).left = (fst T A).left ≫ τ) (h₂ : Ψ ≫ (snd T₀ G).left = (snd T A).left ≫ φ) {e : (𝟙_ (Over S')).left ⟶ (𝟙_ (Over S)).left} (he : eSec A ≫ φ = e ≫ eSec G) (hρ : PMaps w (K

theorem **pt_pow_eq_one** (hK : KerPowSt) {S : Scheme.{0}} [IsLocallyNoetherian S] (P : PolAV S) (n : ℕ) {k : ℤ} (hk : k = n) {T : Over S} (θ : ThetaGrp P.A P.L k T) : θ.pt ^ n = 1

theorem **isUUOver_move** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S : Type) [CommRing S] [Algebra c.F S] (ρ : Xo.X ⟶ Xo.X) (ν : SABMap Xo.univ.t Xo.univ.t ρ) (y y' : FrPt Xo c S) (pm : Gx y.x ⟶ Gx y'.x) (w : LB.Iso Xo.univ.t.PP.L (Xo.univ.t.PP.L.pull ν.cm.F)), y'.x = y.x ≫ ρ → pm ≫ (fst (Over.mk y'.x) Xo.univ.t.G).left = (fst (Over.mk y.x) Xo.univ.t.G).left → pm ≫ (snd (Over.mk y'.x) Xo.univ.t.G).left = (snd (Over.mk y.x) Xo.univ.t.G).left ≫ ν.φ → CubIso ν w → (∀ (i j : Xo.univ.t.PP.L.ι) (W : (Gx y.x).Opens) (h₀ : W ≤ (snd (Over.mk y.x) Xo.univ.t.G).left ⁻¹ᵁ ((LB.div (Xo.univ.t.PP.L.pull ν.cm.F) Xo.univ.t.PP.L).U ((fun _ => i), j))) (h₁ : W ≤ pm ⁻¹ᵁ (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 (ptO Xo y'.x)).U (fun _ => i)) (h₂ : W ≤ (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 (ptO Xo y.x)).U (fun _ => j)), (pm.ap

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsotropicGroupLatticeClassRepresentatives

theorem **finrank_fibre** (O M k : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] [Field k] [Algebra O k] : Module.finrank k (k ⊗[O] M) = Module.finrank O M

theorem **exists_basis_mod** (O M : Type) [CommRing O] [IsDomain O] [AddCommGroup M] [Module O M] [Module.Finite O M] [Module.Projective O M] (m : O) [Finite (O ⧸ Ideal.span {m})] (r : ℕ) (hr : Module.finrank O M = r) : ∃ e : Fin r → M, (∀ x : M, ∃ (c : Fin r → O) (y : M), x = ∑ k, c k • e k + m • y) ∧ ∀ (c : Fin r → O) (y : M), ∑ k, c k • e k = m • y → ∀ k, m ∣ c k

theorem **mem_stdLat** (v : Fin n → E) : v ∈ stdLat E n ↔ ∃ u : Fin n → 𝒪 E, ∀ b, v b = (u b : E)

theorem **mem_smul_stdLat** (N : ℕ) (v : Fin n → E) : v ∈ Ideal.span {(N : 𝒪 E)} • stdLat E n ↔ ∃ u : Fin n → 𝒪 E, ∀ b, v b = (N : E) ∗ (u b : E)

theorem **mul_integral** [IsCMField E] (Λ : GenLat E n) (N : ℕ) (hU : Λ.Unimod) (hN : Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La) (w : Fin n → E) (hw : w ∈ Λ.La) : ∃ v : Fin n → 𝒪 E, ∀ b, (v b : E) = (N : E) ∗ w b

theorem **exists_class_representatives** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∃ zz : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → Dw.La, (∀ i₀, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • zz i₀ j))) ∧ (∀ i i', (∀ b, Dw.ModN (N ^ 2) (zz i b − zz i' b)) → i = i') ∧ ∀ a : Fin n → Dw.La, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a j)) → ∃ i, ∀ b, Dw.ModN (N ^ 2) (a b − zz i b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsIsotropicGroupLatticeClassRepresentativesStatement

theorem **_root_.GlobalTheory.T.exists_isotropic_group_lattice_class_representatives** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∃ zz : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → Dw.La, (∀ i₀, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • zz i₀ j))) ∧ (∀ i i', (∀ b,

Dw.ModN (N ^ 2) (zz i b - zz i' b)) → i = i') ∧ ∀ a : Fin n → Dw.La, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) * mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a j)) → ∃ i, ∀ b, Dw.ModN (N ^ 2) (a b - zz i b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsJointIntEigenvectorOfStable

theorem **exists_int_eigenvector_of_stable** {K V : Type*} [Field K] [AddCommGroup V] [Module K V] (f : Module.End K V) (hf : ⊔ n : ℤ, f.eigenspace (n : K) = ⊤) (U : Submodule K V) (hU : ∀ x ∈ U, f x ∈ U) (hU0 : U ≠ ⊥) : ∃ n : ℤ, ∃ u ∈ U, u ≠ 0 ∧ f u = (n : K) • u

theorem **exists_joint_int_eigenvector_of_stable** {K V ι : Type*} [Field K] [AddCommGroup V] [Module K V] [Fintype ι] (f : ι → Module.End K V) (hc : ∀ a b, f a * f b = f b * f a) (hf : ∀ a, ⊔ n : ℤ, (f a).eigenspace (n : K) = ⊤) (W : Submodule K V) (hW : ∀ a, ∀ x ∈ W, f a x ∈ W) (hW0 : W ≠ ⊥) : ∃ κ : ι → ℤ, ∃ u ∈ W, u ≠ 0 ∧ ∀ a, f a u = (κ a : K) • u

theorem **exists_highest_vector** {K V T : Type*} [Field K] [CharZero K] [AddCommGroup V] [Module K V] [Fintype T] [DecidableEq T] {m : ℕ} (e : T → Fin m → Fin m → Module.End K V) (he : ∀ t t' i k l, e t i i * e t' k l - e t' k l * e t i i = (if t = t' then ((if i = k then 1 else 0) - (if i = l then 1 else 0) : K) else 0) • e t' k l) (hdiag : ⊔ κ : T → Fin m → ℤ, (⊓ t, ⊓ i, Module.End.eigenspace (e t i i) (κ t i : K)) = ⊤) (W : Submodule K V) [FiniteDimensional K W] (hW : ∀ t i j, ∀ w ∈ W, e t i j w ∈ W) (hW0 : W ≠ ⊥) : ∃ v ∈ W, v ≠ 0 ∧ (∀ t i j, i < j → e t i j v = 0) ∧ ∃ κ : T → Fin m → ℤ, ∀ t i, e t i i v = (κ t i : K) • v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsLevMoveOfFrOver

theorem **gw_eq_of_pmaps** {S S₀ : Scheme.{0}} (hL : L.IsLB) (hL' : L'.IsLB) (p : X' ⟶ S) (e : S ⟶ X') (he : e ≫ p = 𝟙 _) (hp : Function.Surjective p.appTop.hom) {F₁ F₂ : X' ⟶ X} (hF : F₁ = F₂) {w₁ : LB.Iso L' (L.pull F₁)} {w₂ : LB.Iso L' (L.pull F₂)} (g₁ : PGen w₁) (g₂ : PGen w₂) {e₀ : S₀ ⟶ X} {a : LB.Gen (L'.pull e)} {t : LB.Gen (L.pull e₀)} {h₁ h₂ : S ⟶ S₀} (hh : h₁ = h₂) (m₁ : PMaps w₁ (K

theorem **gs_of_secMaps** {F : X' ⟶ X} (w : LB.Iso L' (L.pull F)) {p : Y ⟶ X} {p' : Y' ⟶ X'} {pm : Y' ⟶ Y} (hpm : pm ≫ p = p' ≫ F) (s : LB.Sec (LB.LinComb L (K

theorem **rigIso_of** {t : SAB H S} {t₂ : SAB H S₂} {r : S₂ ⟶ S} (m : SABMap t₂ t r) {w : LB.Iso t₂.PP.L (t.PP.L.pull m.φ)} (h₁ : w.IsGen) (h₂ : LB.Gen.Maps (LB.linCombIso w (K

theorem **gw_triangle** {t : SAB H S} {t₂ : SAB H S₂} {u : SAB H T} {r r' : S₂ ⟶ S} {ρ : S₂ ⟶ S₂} {x : T ⟶ S₂} (hu : Function.Surjective u.G.hom.appTop.hom) (m : SABMap t₂ t r) (m' : SABMap t₂ t r') (ν : SABMap t₂ t₂ ρ) (n : SABMap u t₂ x) (hφ : n.φ ≫ m.φ = (n.φ ≫ ν.φ) ≫ m'.φ) (hx : x ≫ r = (x ≫ ρ) ≫ r') {w₀ : LB.Iso t₂.PP.L (t.PP.L.pull m.φ)} {w₀' : LB.Iso t₂.PP.L (t.PP.L.pull m'.φ)} {w : LB.Iso t₂.PP.L (t₂.PP.L.pull ν.φ)} (h₀ : RigIso m w₀) (h₀' : RigIso m' w₀') (h : RigIso ν w) {Z : Scheme.{0}} (f : Z ⟶ u.G.left) (z : Z) (j : t.PP.L.ι) (i' i : t₂.PP.L.ι) (hj : m.φ (n.φ (f z)) ∈ t.PP.L.U j) (hi' : ν.φ (n.φ (f z)) ∈ t₂.PP.L.U i') (hi : n.φ (f z) ∈ t₂.PP.L.U i) : gw w₀ (f ≫ n.φ) z j i = gw w₀' ((f ≫ n.φ) ≫ ν.φ) z j i' * gw w (f ≫ n.φ) z i' i

theorem **pow_left_comp** {t : SAB H S} {t₂ : SAB H S₂} {f : S₂ ⟶ S} (m : SABMap t₂ t f) {σ : 𝟙_ (Over S₂) ⟶ t₂.G} {τ : 𝟙_ (Over S) ⟶ t.G} (h : σ.left ≫ m.φ = f ≫ τ.left) (d : ℕ) : (σ ^ d).left ≫ m.φ = f ≫ (τ ^ d).left

theorem **sec_lift** {t : SAB H S} {t₂ : SAB H S₂} {u : SAB H T} {r : S₂ ⟶ S} {x : T ⟶ S₂} {z : T ⟶ S} (hz : x ≫ r = z) (m : SABMap t₂ t r) (n : SABMap u t₂ x) {s : 𝟙_ (Over T) ⟶ u.G} {σ : 𝟙_ (Over S₂) ⟶ t₂.G} {τ : 𝟙_ (Over S) ⟶ t.G} (h₁ : σ.left ≫ m.φ = r ≫ τ.left) (h₂ : s.left ≫ n.φ ≫ m.φ = z ≫ τ.left) : s.left ≫ n.φ = x ≫ σ.left

theorem **exists_fib** {s : T ⟶ Sp (KA l)} {o : Obj D l (baseKA l) s} {z : T ⟶ Xo.X} (p : ObjMap o Xo.univ z) {r : Xo₂.X ⟶ Xo.X} {m : SABMap Xo₂.univ.t Xo.univ.t r} (hm : Lev m) {x : T ⟶ Xo₂.X} (hz : x ≫ r = z) : ∃ n : SABMap o.t Xo₂.univ.t x, n.φ ≫ m.φ = p.toSABMap.φ ∧ Fib o n

theorem **exists_objMap** {s : T ⟶ Sp (KA l)} {o : Obj D l (baseKA l) s} {x : T ⟶ Xo₂.X} {n : SABMap o.t Xo₂.univ.t x} (hn : Fib o n) {r : Xo₂.X ⟶ Xo.X} {m : SABMap Xo₂.univ.t Xo.univ.t r} (hm : Lev m) : ∃ q : ObjMap o Xo.univ (x ≫ r), q.toSABMap.φ = n.φ ≫ m.φ

theorem **fib_move** (hL : Laws) (hd : l.Dvd l₂) {k : Type} [Field k] {s : Sp k ⟶ Sp (KA l)} {o : Obj D l (baseKA l) s} {x x' : Sp k ⟶ Xo₂.X} {n : SABMap o.t Xo₂.univ.t x} {n' : SABMap o.t Xo₂.univ.t x'} (hn : Fib o n) (hn' : Fib o n') {r : Xo₂.X ⟶ Xo.X} {m : SABMap Xo₂.univ.t Xo.univ.t r} (hm : Lev m) {g : Sim D} (hg : g.InGam l l₂) {ρ : Xo₂.X ⟶ Xo₂.X} (ν : SABMap Xo₂.univ.t Xo₂.univ.t ρ) (hν : ∀ a, (Xo₂.univ.eta (g.f l₂.M a)).left ≫ ν.φ = ρ ≫ (Xo₂.univ.eta a).left) (hx : x' = x ≫ ρ) : n'.φ = n.φ ≫ ν.φ

theorem **exists_levMove_of_frOver** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), @ToroidalLevelCompatibility E _ _ _ H Dw → l.Dvd d.l₂ → d.l₂.M = l.M * d.pd → d.EtaOK → ∀ (r' : d.Xo₂.X ⟶ Xo.X) (m' : SABMap d.Xo₂.univ.t Xo.univ.t r'), DeltaDat.EtaOK { d with r

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsLevelTranslateJetNeZero

theorem **polyDeriv_const_mul_entire** {σ W : Type} [NormedAddCommGroup W] [NormedSpace ℂ W] [FiniteDimensional ℂ W] (P : MvPolynomial σ ℂ) (u : σ → W) (c : ℂ) {G : W → ℂ} (hG : Differentiable ℂ G) (x : W) : I2.polyDeriv P u (fun z => c * G z) x = c * I2.polyDeriv P u G x

theorem **polyDeriv_const_mul_of_entire** {σ W : Type} [NormedAddCommGroup W] [NormedSpace ℂ W] [FiniteDimensional ℂ W] (P : MvPolynomial σ ℂ) (u : σ → W) (c : ℂ) {T : W → ℂ} (hG : Differentiable ℂ fun z => c * T z) (x : W) : I2.polyDeriv P u (fun z => c * T z) x = c * I2.polyDeriv P u T x

theorem **exists_level_translate_jet_ne_zero** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (𝔐 : Ideal (𝒪 E)) (M : ℕ), Odd M → 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)} → ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∀ (θ : (Fin g → ℂ) → ℂ) (κ : ℂ), IsTheta1 X bas idx (w : E) x L θ → MeanOne X bas idx x L θ → (∀ z : Fin n → Fin g → ℂ, ∏ b, θ (z b) = κ * thA X bas idx v x z) → ∀ P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n), P ≠ 0 → 3 * P.totalDegree + 2 ≤ M → ∃ y : Matrix (Fin n) (Fin r) E, (∀ b, (M : E) • y b ∈ 𝔐 • L) ∧ ∀ y' : Matrix (Fin n) (Fin r) E, (∀ b, y' b - y b ∈ L) → κ * thP X hr bas idx (heisG X

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsLevelTranslateJetNeZeroStatement

theorem **_root_.GlobalTheory.T.exists_level_translate_jet_ne_zero_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (𝔐 : Ideal (𝒪 E)) (M : ℕ), Odd M → 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)} → ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∀ (θ : (Fin g → ℂ) → ℂ) (κ : ℂ), IsTheta1 X bas idx (w : E) x L θ → MeanOne X bas idx x L θ → (∀ z : Fin n → Fin g → ℂ, ∏ b, θ (z b) = κ * thA X bas idx v x z) → ∀ P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n), P ≠ 0 → 3 * P.totalDegree + 2 ≤ M → ∃ y : Matrix (Fin n) (Fin r) E, (∀ b, (M : E) • y b ∈ 𝔐 • L) ∧ ∀ y' : Matrix (Fin n) (Fin r) E, (∀ b, y' b - y b ∈ L) → κ *

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsMulNe

theorem **pnorm_pos** (X : Data E r) (la : Wt X.Ψ r) {P : Frame X.Ψ r → ℂ} {x : Frame X.Ψ r} (hx : x ∈ X.dom) (hP : P x ≠ 0) : 0 < X.pnorm la P x

theorem **isBddSec_sum** (X : Data E r) {la : Wt X.Ψ r} {m : ℕ} {f : Fin m → Frame X.Ψ r → ℂ} (hf : ∀ q, X.IsBddSec la (f q)) (u : Fin m → ℂ) : X.IsBddSec la fun x => ∑ q, u q * f q x

theorem **s_xs** {X : Data E r} {Xa : ArchInM X} (Fl : FlatM X Xa) (j : Fin Fl.δ) (k : Fin (nd X)) : Fl.s k (Fl.xs j) = 0

theorem **zero_mem_OmF** {X : Data E r} {x : Frame X.Ψ r} (hx : x ∈ X.dom) : (0 : V (nd X)) ∈ OmF X x

theorem **exists_s_ne** {X : Data E r} {Xa : ArchInM X} (Fl : FlatM X Xa) : ∃ k, ∃ x ∈ X.dom, Fl.s k x ≠ 0

theorem **exists_comb** {X : Data E r} {Xa : ArchInM X} (Fl : FlatM X Xa) (e m : ℕ) (f : Fin m → Frame X.Ψ r → ℂ) (hf : ∀ q, DifferentiableOn ℂ (f q) X.dom) (hm : Fl.δ * (e + nd X).choose (nd X) < m) : ∃ u : Fin m → ℂ, u ≠ 0 ∧ ∀ j, (fun z : V (nd X) => ∑ q, u q * f q (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (e + 1)

theorem **exists_mul_ne** {X : Data E r} {f g : Frame X.Ψ r → ℂ} (hf : DifferentiableOn ℂ f X.dom) (hg : DifferentiableOn ℂ g X.dom) (hf0 : ∃ x ∈ X.dom, f x ≠ 0) (hg0 : ∃ x ∈ X.dom, g x ≠ 0) : ∃ x ∈ X.dom, f x * g x ≠ 0

theorem **upper** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (m : ℕ) (f : Fin m → Frame X.Ψ r → ℂ), (∀ q, X.IsBddSec (Xa.Fp + Xa.T • Xa.L) (f q)) → (∀ u : Fin m → ℂ, (∀ x ∈ X.dom, ∑ q, u q * f q x = 0) → u = 0) → m ≤ Fl.δ * (Xa.T + 2 * nd X + 6).choose (nd X)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNatCastMemDifferentIdealPowLe

theorem **exists_prime_dvd_natCast_mem** {R : Type*} [CommRing R] {P : Ideal R} (hP : P.IsPrime) : ∀ N : ℕ, N ≠ 0 → (N : R) ∈ P → ∃ p : ℕ, p.Prime ∧ p ∣ N ∧ (p : R) ∈ P

theorem **span_natCast_dvd_differentIdeal_sq** (E : Type) [Field E] [NumberField E] [IsGalois ℚ E] (N : ℕ) (hN0 : 0 < N) (hN : ((N : ℕ) : 𝒪 E) ∈ differentIdeal ℤ (𝒪 E)) (hmin : ∀ M : ℕ, 0 < M → M < N → ((M : ℕ) : 𝒪 E) ∉ differentIdeal ℤ (𝒪 E)) : Ideal.span {((N : ℕ) : 𝒪 E)} ∣ differentIdeal ℤ (𝒪 E) ^ 2

theorem **exists_natCast_mem_differentIdeal_pow_le** : ∀ (E : Type) [Field E] [NumberField E] [IsGalois ℚ E], ∃ N : ℕ, 0 < N ∧ ((N : ℕ) : 𝒪 E) ∈ differentIdeal ℤ (𝒪 E) ∧ (N : ℝ) ^ Module.finrank ℚ E ≤ |(NumberField.discr E : ℝ)| ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNatCastMemDifferentIdealPowLeStatement

theorem **_root_.GlobalTheory.T.exists_natCast_mem_differentIdeal_pow_le_discr_sq** : ∀ (E : Type) [Field E] [NumberField E] [IsGalois ℚ E], ∃ N : ℕ, 0 < N ∧ ((N : ℕ) : 𝒪 E) ∈ differentIdeal ℤ (𝒪 E) ∧ (N : ℝ) ^ Module.finrank ℚ E ≤ |(NumberField.discr E : ℝ)| ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNatCastSmulCoeffIsIntegral

theorem **exists_natCast_smul_coeff_isIntegral** {σ K : Type} [Field K] [CharZero K] [FiniteDimensional ℚ K] (P : MvPolynomial σ K) : ∃ N : ℕ, 0 < N ∧ ∀ m, IsIntegral ℤ (((N : K) • P).coeff m)

theorem **smul_mem_HHa** {T : Type} {r : T → ℕ} {n : ℕ} {k : Type} [Field k] {cc : T → T} {isX : T → Prop} {a : T → ℕ → ℕ} {P : MvPolynomial (Letter T r n) k} (t : k) (hP : P ∈ HHa r n k cc isX a) : t • P ∈ HHa r n k cc isX a

theorem **exists_natCast_smul_mem_harmA** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∃ N : ℕ, 0 < N ∧ (N : c.F) • P ∈ harmA c a n ∧ ((N : c.F) • P).support = P.support

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNatSmulCoeffIsIntegral

theorem **exists_nat_smul_coeff_isIntegral** {σ K : Type} [Field K] [CharZero K] [FiniteDimensional ℚ K] (P : MvPolynomial σ K) : ∃ N : ℕ, 0 < N ∧ ∀ m : σ →₀ ℕ, IsIntegral ℤ (MvPolynomial.coeff m ((N : K) • P))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNeZeroOnCoset

theorem **eps_zero** (X : PT E) : X.eps 0 = 1

theorem **natCast_smul_mem** (X : PT E) {x : Vsp X.dim} (hx : x ∈ X.L) (k : ℕ) : (k : ℂ) • x ∈ X.L

theorem **eps_natCast_smul** (X : PT E) {x : Vsp X.dim} (hx : x ∈ X.L) (k : ℕ) : (X.eps ((k : ℂ) • x) : ℝ) = (X.eps x : ℝ) ^ k

theorem **isPi_natCast** (X : PT E) (n : ℕ) : IsPi X (n ^ 2) ((n : ℂ) • LinearMap.id)

theorem **exists_isReps** (X : PT E) (n : ℕ) (hn : n ≠ 0) : ∃ K, IsReps X (n ^ 2) 1 ((n : ℂ) • LinearMap.id) K

theorem **exists_ne_zero_on_coset** (E : Type) [Field E] [NumberField E] [IsCMField E] (X : PT E) (s : (Fin 1 → Vsp X.dim) → ℂ) (hs : s ∈ X.Sec 1 1) (h0 : s ≠ 0) (n : ℕ) (hn : 2 ≤ n) (a : Vsp X.dim) : ∃ y ∈ X.L, s (fun _ => a + ((n : ℂ)⁻¹) • y) ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNeZeroPair

theorem **exists_ne_zero_pair** (g : ℂ → ℂ) (hg : Differentiable ℂ g) (h1 : g 1 ≠ 0) (m : ℂ) (hm : m ≠ 0) : ∃ t : ℂ, g t ≠ 0 ∧ g (-(m * t)) ≠ 0

theorem **HC_add_smul_left** {ι : Type} [Fintype ι] {n : ℕ} (w c γ : Fin n → EuclideanSpace ℂ ι) (a : ℂ) : HC (w + a • c) γ = HC w γ + a * HC c γ

theorem **prod_translate_mem** {E : Type} [Field E] [NumberField E] (X : PT E) {n : ℕ} (s : (Fin n → Vsp X.dim) → ℂ) (hs : s ∈ X.Sec 1 n) (m : ℕ) (c : Fin n → Vsp X.dim) : (fun w => s (w + (1 : ℂ) • c) ^ m * s (w + (-(m : ℂ)) • c)) ∈ X.Sec (m + 1) n

theorem **exists_sec_ne_zero** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (X : PT E) (s : (Fin 1 → Vsp X.dim) → ℂ), s ∈ X.Sec 1 1 → s ≠ 0 → ∀ q : ℕ, 2 ≤ q → ∀ v : Fin 1 → Vsp X.dim, ∃ F ∈ X.Sec q 1, F v ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNetAllComponents

theorem **component_mem_SS** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} {Comp : Type} (X : Comp → Data E r) (la : ∀ k, Wt (X k).Ψ r) {P : FDSec X} (h : P ∈ holSub X la) : ∀ k, P k ∈ (X k).SS (la k)

theorem **one_le_absDisc** (E : Type) [Field E] [NumberField E] [IsCMField E] : 1 ≤ dE E

theorem **exists_net_all_components** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (Comp : Type) [Fintype Comp] [Nonempty Comp] (Xx : Comp → DataX E r) (la : ∀ k, Wt (Xx k).Ψ r) (M w L : ℝ), 1 ≤ r → 1 ≤ L → 1 ≤ M → 0 < w → (∀ k, (∑ τ, (Xx k).spread (la k) τ) ≤ w) → (∀ k, ∀ v ∈ (Xx k).L', herm (Xx k).Gm v v = 0 → ∀ b : E, cc b = b → IsIntegral ℤ b → ∃ γ ∈ (Xx k).Γ, (Xx k).χ (la k) γ = 1 ∧ ∀ τ : (Xx k).Ψ, (γ : Matrix (Fin r) (Fin r) E).map τ.1 = (Xx k).flowMat v τ ((M : ℂ) * τ.1 b)) → Real.log (fdeg E) ≤ 2 * L → Real.log r

≤ L → Real.log M ≤ 30 * L → Real.log w ≤ 3 * L → Real.log (dE E) ≤ 4 * fdeg E * L → (∀ k, Real.log (1 + ∑ τ, ∑ i, |(la k τ i : ℝ)|) ≤ 3 * L ∧ Real.log (Xx k).NA ≤ 30 * fdeg E * L ∧ Real.log (Xx k).NC ≤ 30 * fdeg E * L ∧ Real.log (Xx k).g ≤ 30 * fdeg E * L) → Real.log (Fintype.card Comp) ≤ 41 * (fdeg E * r * L) ^ 2 → ∃ NN : Finset (FPt (fun k => (Xx k).toDat

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNetAllComponentsMax

theorem **exists_net_all_components_max** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (Comp : Type) [Fintype Comp] [Nonempty Comp] (Xx : Comp → DataX E r) (la : ∀ k, Wt (Xx k).Ψ r) (M w L : ℝ), 1 ≤ r → 1 ≤ L → 1 ≤ M → 0 < w → (∀ k, (∑ τ, (Xx k).spread (la k) τ) ≤ w) → (∀ k, ∀ v ∈ (Xx k).L', herm (Xx k).Gm v v = 0 → ∀ b : E, cc b = b → IsIntegral ℤ b → ∃ γ ∈ (Xx k).Γ, (Xx k).χ (la k) γ = 1 ∧ ∀ τ : (Xx k).Ψ, (γ : Matrix (Fin r) (Fin r) E).map τ.1 = (Xx k).flowMat v τ ((M : ℂ) * τ.1 b)) → Real.log (fdeg E) ≤ 2 * L → Real.log r ≤ L → Real.log M ≤ 30 * L → Real.log w ≤ 3 * L → Real.log (dE E) ≤ 4 * fdeg E * L → (∀ k, Real.log (1 + ∑ τ, ∑ i, |(la k τ i : ℝ)|) ≤ 3 * L ∧ Real.log (Xx k).NA ≤ 30 * fdeg E * L ∧ Real.log (Xx k).NC ≤ 30 * fdeg E * L ∧ Real.log (Xx k).g ≤ 30 * fdeg E * L) → Real.log (Fintype.card Comp) ≤ 41 * (fdeg E * r * L) ^ 2 → ∃ NN : Finset (FPt (fun k => (Xx k).t

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNetDataFromHalfConjugateEmbeddings

theorem **linearIndependent_conj_coords** {ι κ : Type*} [Fintype ι] (v : ι → κ → ℂ) (h : LinearIndependent ℂ v) : LinearIndependent ℂ (fun i y => conj (v i y))

theorem **exists_netData_of_readings_on_half** : ∀ (D : D7) (B : WeightFamilies.Prov D) (j : ℕ) (ka : WeightFamilies.Wt D j) (good : (B.KO →+* ℂ) → Prop), (∀ σ, good σ ↔ ¬ good (ComplexEmbedding.conjugate σ)) → (∀ σ F, B.supN j ka (ComplexEmbedding.conjugate σ) F = B.supN j ka σ F) → ∀ (n : (B.KO →+* ℂ) → ℕ) (ev : ∀ σ, Fin (n σ) → (B.Cusp j ka →ₛₗ[σ] ℂ)) (C : ℝ), (∀ σ, good σ → (∀ F, ∃ y, B.supN j ka σ F / 4 ≤ ‖ev σ y F‖) ∧ (∀ y F, ‖ev σ y F‖ ≤ B.supN j ka σ F) ∧ Real.log (16 * (n σ : ℝ)) ≤ C ∧ ∀ (m : ℕ) (F : Fin m → B.Cusp j ka), LinearIndependent B.KO F → LinearIndependent ℂ (fun i => fun y => ev σ y (F i))) → ∃ Nt : NetData B j ka, NetMax Nt ∧ NetLeSup Nt ∧ NetSize Nt C ∧ ∀ σ, NetInj Nt σ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNetDataFromHalfConjugateEmbeddingsStatement

theorem **_root_.GlobalTheory.T.exists_netData_from_half_conjugate_embeddings** : ∀ (D : D7) (B : WeightFamilies.Prov D) (j : ℕ) (ka : WeightFamilies.Wt D j) (good : (B.KO →+* ℂ) → Prop), (∀ σ, good σ ↔ ¬ good (ComplexEmbedding.conjugate σ)) → (∀ σ F, B.supN j ka (ComplexEmbedding.conjugate σ) F = B.supN j ka σ F) → ∀ (n : (B.KO →+* ℂ) → ℕ) (ev : ∀ σ, Fin (n σ) → (B.Cusp j ka →ₛₗ[σ] ℂ)) (C : ℝ), (∀ σ, good σ → (∀ F, ∃ y, B.supN j ka σ F / 4 ≤ ‖ev σ y F‖) ∧ (∀ y F, ‖ev σ y F‖ ≤ B.supN j ka σ F) ∧ Real.log (16 * (n σ : ℝ)) ≤ C ∧ ∀ (m : ℕ) (F : Fin m → B.Cusp j ka), LinearIndependent B.KO F → LinearIndependent ℂ (fun i => fun y => ev σ y (F i))) → ∃ Nt : NetData B j ka, NetMax Nt ∧ NetLeSup Nt ∧ NetSize Nt C ∧ ∀ σ, NetInj Nt σ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNetMaxAtGoodEmb

theorem **exists_net_max_at_goodEmb** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → 16 ≤ D.R.f → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∃ NN : Finset (FPt (T.U j σ).X), NN.Nonempty ∧ (∀ P ∈ holSub (T.U j σ).X (laU (T.U j σ) (wtA w) w.n), ∃ p ∈ NN, supN (T.U j σ).X (laU (T.U j σ) (wtA w) w.n) P ≤ 4 * ((T.U j σ).X p.1).pnorm (laU (T.U j σ) (wtA w) w.n p.1) (P p.1) p.2) ∧ Real.log (16 * NN.card) ≤ 1041 * Zb D large₂ ^ 3

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNetOfThickPoints

theorem **norm_hermC_le_of_ortho** (X : Data E r) {x : Frame X.Ψ r} (hx : X.IsOrtho x) (τ : X.Ψ) (u v : Fin r → ℂ) : ‖hermC τ.1 X.Gm u v‖ ≤ (X.maj x τ u + X.maj x τ v) / 2

theorem **norm_herm_le_of_ortho** (X : Data E r) {x : Frame X.Ψ r} (hx : X.IsOrtho x) {B : ℝ} {u v : Fin r → E} (hu : ∑ τ, X.maj x τ (Data.emb τ.1 u) ≤ B) (hv : ∑ τ, X.maj x τ (Data.emb τ.1 v) ≤ B) (σ : E →+* ℂ) : ‖σ (herm X.Gm u v)‖ ≤ B

theorem **pnorm_lact_rmul** (X : Data E r) {la : Wt X.Ψ r} {P : Frame X.Ψ r → ℂ} (hP : X.IsSec la P) {γ : GL (Fin r) E} (hγ : γ ∈ X.Γ) {x b : Frame X.Ψ r} (hx : x ∈ X.dom) (hb : b ∈ Bor X.Ψ r) : X.pnorm la P (lact γ (rmul x b)) = X.pnorm la P x

theorem **dom_nonempty** (X : Data E r) : X.dom.Nonempty

theorem **epStar_pos** {M f w k : ℝ} (hM : 0 < M) (hf : 0 ≤ f) (hw : 0 < w) (hk : 0 < k) : 0 < epStar M f w k

theorem **exists_net_of_thick_points** {r : ℕ} (X : DataX E r) {ε B η : ℝ} (hε : 0 < ε) (hB : 1 ≤ B) (hεB : (2 * fdeg E * r : ℝ) ^ 4 * dE E ^ 2 / (4 * ε) ≤ B) (hη0 : 0 < η) (hη : η ≤ 1) : ∃ N : Finset (Frame X.Ψ r), (∀ y ∈ N, y ∈ X.dom) ∧ (∀ x ∈ X.dom, (∀ v ∈ X.L', v ≠ 0 → ε ≤ ∑ τ, X.maj x τ (Data.emb τ.1 v)) → ∃ γ ∈ X.Γ, ∃ b ∈ Bor X.Ψ r, ∃ A : Frame X.Ψ r, X.IsOrtho (lact γ (rmul x b)) ∧ rmul (lact γ (rmul x b)) A ∈ N ∧ ∀ τ a c, ‖A τ a c − (if a = c then 1 else 0)‖ ≤ η) ∧ (N.card : ℝ) ≤ ((1 + 2 * B) ^ (2 * fdeg E * r ^ 2) * ((X.NA : ℝ) * X.NC * r * B) ^ (2 * fdeg E * r ^ 2) * (X.g : ℝ) ^ (2 * fdeg E * r ^ 2)) * Real.exp (2 * (fdeg E : ℝ) * (r : ℝ) ^ 2 * Real.log (6 * (r : ℝ) ^ 2 / η) + 2 * (fdeg E : ℝ) * (r : ℝ) ^ 3 * Real.log (r * B))

theorem **exists_finite_net** {r : ℕ} (X : DataX E r) (la : Wt X.Ψ r) (M w : ℝ) (hM : 0 < M) (hw : (∑ τ, X.spread la τ) ≤ w) (hw0 : 0 < w) (hT : ∀ v ∈ X.L', herm X.Gm v v = 0 → ∀ b : E, cc b = b → IsIntegral ℤ b → ∃ γ ∈ X.Γ, X.χ la γ = 1 ∧ ∀ τ : X.Ψ, (γ : Matrix (Fin r) (Fin r) E).map τ.1 = X.flowMat v τ ((M : ℂ) * τ.1 b)) : let ε : ℝ

theorem **exists_finite_net_rank_one** : ∃ (X : DataX E4 1) (la : Wt X.Ψ 1) (P : Frame X.Ψ 1 → ℂ) (N : Finset (Frame X.Ψ 1)), P ∈ X.SS la ∧ la ≠ 0 ∧ (∃ x ∈ X.dom, P x ≠ 0) ∧ N.Nonempty ∧ (∀ y ∈ N, y ∈ X.dom) ∧ ∀ Q ∈ X.SS la, ∀ x ∈ X.dom, ∃ y ∈ N, X.pnorm la Q x ≤ 4 * X.pnorm la Q y

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNormalisedIsoReadingOneAnalyticFrame

theorem **finset_fin_two_eq_cases** (I : Finset (Fin 2)) : I = ∅ ∨ I = {0} ∨ I = {1} ∨ I = {0, 1}

theorem **e_psum** {S : Scheme.{0}} {A : Over S} [GrpObj A] [IsCommMonObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) (w : A ⟶ A) (z z' : Fin g → ℂ) (h : Φ.e z' = Φ.e z ≫ w) (I : ULift.{0} (Finset (Fin 2))) : Φ.e z ≫ psum A ![𝟙 A, w] I.down = Φ.e (psumL 2 g I ![z, z'])

theorem **sum_ulift_finset_fin_two_expand** {M : Type} [AddCommMonoid M] (f : ULift.{0} (Finset (Fin 2)) → M) : ∑ I, f I = f (∅) + f ({0}) + f ({1}) + f ({0, 1})

theorem **sameSum_of_const** {X Y : Scheme.{0}} {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y ⟶ X)} {ε : K → ℤ} {φ' : K' → (Y ⟶ X)} {ε' : K' → ℤ} (f : Y ⟶ X) (h : ∀ k, φ k = f) (h' : ∀ k, φ' k = f) (hs : ∑ k, ε k = ∑ k, ε' k) : LB.SameSum φ ε φ' ε'

private theorem **w0_add_bar_w0_eq_one** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : H.w0 + H.bar H.w0 = 1

private theorem **hf_eq_cj_hf_swap_of_isHerm** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (T Tb : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (hadj : ∀ z w, hermForm Hm (T z) w = hermForm Hm z (Tb w)) (hTT : ∀ w, T w + Tb w = w) (z w : Fin g → ℂ) : ∑ k : Fin 1 ⊕ ULift.{0} (Finset (Fin 2)), (((Sum.elim (fun _ => (1 : ℤ)) fun I => -cε 2 I) k : ℤ) : ℂ) * hermForm Hm ((Sum.elim (fun _ => LinearMap.id) (fun I => (psumL 2 g

I).comp (LinearMap.pi ![LinearMap.id, T])) : (Fin 1 ⊕ ULift.{0} (Finset (Fin 2))) → (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) k z) ((Sum.elim (fun _ => LinearMap.id) (fun I => (psumL 2 g I).comp (LinearMap.pi ![LinearMap.id, T])) : (Fin 1 ⊕ ULift.{0} (Finset (Fin 2))) → (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) k w) = 0

theorem **prim_reading** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), (∃ wP : PrimTy H Xo, IsPrim wP) ∧ ∀ wP : PrimTy H Xo, IsPrim wP → ∀ z : Fin g → ℂ, Φ.toAnFr.GenRd _ _ wP (Φ.e z).left (Sum.elim (fun _ => z) (fun I => psumL 2 g I ![z, tyAct ty H.w0 z])) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsNormalisedIsoReadingOneAnalyticFrameStatement

theorem **_root_.GlobalTheory.T.exists_normalisedIso_reading_one_analyticFrame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), (∃ wP : PrimTy H Xo, IsPrim wP) ∧ ∀ wP : PrimTy H Xo, IsPrim wP → ∀ z : Fin g → ℂ, Φ.toAnFr.GenRd _ _ wP (Φ.e z).left (Sum.elim (fun _ => z) (fun I => psumL 2 g I ![z, tyAct ty H.w0 z])) 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsOrthoFrame

theorem **exists_ortho_frame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : FD.Data E r), ∀ x ∈ X.dom, ∃ b ∈ FD.Bor X.Ψ r, FD.rmul x b ∈ X.dom ∧ X.IsOrtho (FD.rmul x b) ∧ (∀ τ i, ‖FD.dmin X.Gm (FD.rmul x b) τ i‖ = 1) ∧ (∀ th, I2.phiFD X th (FD.rmul x b) = I2.phiFD X th x) ∧ ∀ (la : FD.Wt X.Ψ r) (P : FD.Frame X.Ψ r → ℂ), X.IsSec la P → X.pnorm la P x = ‖P (FD.rmul x b)‖

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsPowMulCoprimeLe

theorem **exists_pow_mul_coprime_le** {R : Type} [CommRing R] [IsDedekindDomain R] (A K : Ideal R) (hK : K ≠ 0) : ∃ (j : ℕ) (J : Ideal R), A ^ j ∗ J ≤ K ∧ A ⊔ J = ⊤

private theorem **comp_act_eq_one_of_pow_eq_killed** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) {T : Over Xo.X} (t w : T ⟶ Xo.univ.t.G) (N : ℕ) (ht : KilledBy Xo l.Mb t) (hw : w ^ N = t) : ∀ x ∈ Ideal.span {(N : 𝒪 E)} ∗ l.Mb, w ≫ Xo.univ.t.act.act x = 1

theorem **exists_root_killedBy** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) {T : Over Xo.X} (t w : T ⟶ Xo.univ.t.G) (N : ℕ) (ht : KilledBy Xo l.Mb t) (hw : w ^ N = t) (I J : Ideal (𝒪 E)) (hI : I ≤ l.Mb) (hIJ : I ⊔ J = ⊤) (hle : I ∗ J ≤ Ideal.span {(N : 𝒪 E)} ∗ l.Mb) : ∃ t' : T ⟶ Xo.univ.t.G, t' ^ N = t ∧ KilledBy Xo I t'

theorem **exists_root_killedBy_pow** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (T : Over Xo.X) (t w : T ⟶ Xo.univ.t.G) (N : ℕ), 0 < N → KilledBy Xo l.Mb t → w ^ N = t → (∃ t' : T ⟶ Xo.univ.t.G, t' ^ N = t ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) t') ∧ (N = l.M → ∃ t' : T ⟶ Xo.univ.t.G, t' ^ N = t ∧ KilledBy Xo (l.Mb ^ 2) t')

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsPtJ

theorem **czQ** (b : Fin n → Q k d l) : CZ (fun j => (b j).1)

theorem **sb_apply** (b : Fin n → Q k d l) (f : PS k n d) : sb b f = ev (fun j => (b j).1) f

@[simp] theorem **sq_gen** (b : Fin n → Q k d l) (j : Fin n) : sq b (gen j) = b j

theorem **ac_val** (A : Q k d 1) (u : Q k d n) : (ac A u).1 = comp1 A.1 u.1

theorem **op_val** (F : Q k d 2) (u v : Q k d n) : (op F u v).1 = addOf F.1 u.1 v.1

theorem **pj_eq_mk** {m : ℕ} {f : MvPowerSeries σ R} {q : MvPolynomial σ R} : pj m f = Ideal.Quotient.mk _ q ↔ ∀ e : σ →₀ ℕ, e.degree < m + 1 → coeff e f = q.coeff e

theorem **pj_coe** (m : ℕ) (q : MvPolynomial σ R) : pj m (q : MvPowerSeries σ R) = Ideal.Quotient.mk _ q

theorem **pj_X** (m : ℕ) (s : σ) : pj m (X s : MvPowerSeries σ R) = zR m s

theorem **pj_trunc** (m : ℕ) (f : MvPowerSeries σ R) : jetTruncR R σ m (pj (m + 1) f) = pj m f

theorem **aug_pj** (m : ℕ) (f : MvPowerSeries σ R) : jetAug R σ m (pj m f) = constantCoeff f

theorem **pj_ext** {f g : MvPowerSeries σ R} (h : ∀ m, pj m f = pj m g) : f = g

private theorem **coeff_add_degree_eq_coeff_degree_of_stable** (q : ℕ → MvPolynomial σ R) (hq : ∀ m, ∀ e : σ →₀ ℕ, e.degree < m + 1 → (q (m + 1)).coeff e = (q m).coeff e) (e : σ →₀ ℕ) : ∀ j, (q (e.degree + j)).coeff e = (q e.degree).coeff e

theorem **exists_pj** (z : ∀ m, JetR R σ m) (hz : ∀ m, jetTruncR R σ m (z (m + 1)) = z m) : ∃ f : MvPowerSeries σ R, ∀ m, pj m f = z m

theorem **jS_zR** {κ : Type} (m : ℕ) (a : κ → MvPowerSeries τ R) (ha) (s : κ) : jS m a ha (zR m s) = pj m (a s)

private theorem **pj_add** {κ : Type} [Finite κ] (m : ℕ) (g h : MvPowerSeries κ R) : pj m (g + h) = pj m g + pj m h

private theorem **pj_mul'** {κ : Type} [Finite κ] (m : ℕ) (g h : MvPowerSeries κ R) : pj m (g ∗ h) = pj m g ∗ pj m h

private theorem **pj_aeval** (m : ℕ) (a : σ → MvPowerSeries τ R) (q : MvPolynomial σ R) : pj m (MvPolynomial.aeval a q) = MvPolynomial.aeval (fun s => pj m (a s)) q

theorem **jS_pj** (m : ℕ) (a : σ → MvPowerSeries τ R) (ha) (f : MvPowerSeries σ R) : jS m a ha (pj m f) = pj m (subst a f)

theorem **jP_zR** (m : ℕ) (p : Q R d n) (j : Fin d) : jP m p (zR m j) = pj m (p.1 j)

private theorem **jet_spMap_square_comm** {A B C : Type} [CommRing A] [Algebra R A] [CommRing B] [Algebra R B] [CommRing C] [Algebra R C] (f : A →ₐ[R] B) (g : B →ₐ[R] C) : spMap (g.comp f) = spMap g ≫ spMap f

private theorem **jetR_algHom_ext_of_eq_on_zR** {κ A : Type} [CommRing A] [Algebra R A] {m : ℕ} (f g : JetR R κ m →ₐ[R] A) (h : ∀ j, f (zR m j) = g (zR m j)) : f = g

theorem **ptJ_sq** (m : ℕ) (b : Fin n → Q R d l) (p : Q R d n) : spMap (jQ m b) ≫ ptJ x m p = ptJ x m (sq b p)

theorem **aug_jS** (m : ℕ) (p : Q R d n) : (jetAug R _ m).comp (jP m p) = jetAug R (Fin d) m

private theorem **spMap_jetAug_comp_x_eq_one** (m : ℕ) : spMap (jetAug R (Fin d) m) ≫ x m = 1

theorem **ptJ_aug** (m : ℕ) (p : Q R d n) : spMap (jetAug R _ m) ≫ ptJ x m p = 1

theorem **ptJ_trunc** (m : ℕ) (p : Q R d n) : spMap (jetTruncR R _ m) ≫ ptJ x (m + 1) p = ptJ x m p

theorem **ptJ_inj** {p q : Q R d n} (h : ∀ m, ptJ x m p = ptJ x m q) : p = q

private theorem **jetAug_comp_jetTruncR_eq_jetAug_succ** (κ : Type) (m : ℕ) : (jetAug R κ m).comp (jetTruncR R κ m) = jetAug R κ (m + 1)

theorem **exists_ptJ** (y : ∀ m, sp (R₀

theorem **ptJ_zero** (m : ℕ) : ptJ x m (0 : Q R d n) = 1

theorem **exists_series** [IsCommMonObj G] {O : Type} [CommRing O] (a : Geo.Tor.OAct O G) : ∃ (F : FGLaw R d) (A : OAct O F), (∀ m, ptJ x m ((F.F, F.const) : Q R d 2) = ptJ x m (gen 0) * ptJ x m (gen 1)) ∧ ∀ c m, ptJ x m ((A.act c, A.const c) : Q R d 1) = ptJ x m (gen 0) » a.act c

private theorem **prod_ofFn_eq_one_of_forall_eq_one** {M : Type*} [Monoid M] (n : ℕ) : ∀ g : Fin n → M, (∀ j, g j = 1) → (List.ofFn g).prod = 1

private theorem **prod_ofFn_eq_apply_of_forall_ne_eq_one** {M : Type*} [Monoid M] (n : ℕ) : ∀ (f : Fin n → M) (i : Fin n), (∀ j, j ≠ i → f j = 1) → (List.ofFn f).prod = f i

theorem **comb_single** {v : Fin d → (Geo.Tor.epsR R ⟶ G)} (hv : ∀ j, v j = spMap (dualLift (Pi.single j 1)) » x 1) (i : Fin d) : Geo.Tor.IsLieComb (Pi.single i 1) v (v i)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsQuotEquiv

theorem **bar_bar** (H : Host E) (a : 𝒪 E) : H.bar (H.bar a) = a

theorem **bar_mem** {I : Ideal (𝒪 E)} {j : 𝒪 E} (hj : j ∈ I.map (H.bar : 𝒪 E →+* 𝒪 E)) : H.bar j ∈ I

theorem **dvd_EF_smul** (D : Datum H) {N : ℕ} {t : 𝒪 E} (ht : t ∈ Ideal.span {(N : 𝒪 E)}) (u v : D.La) : (N : ℤ) ∣ D.EF u (t • v)

theorem **dvd_right** (D : Datum H) {N : ℕ} {I : Ideal (𝒪 E)} (hI : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) {c : 𝒪 E} (hc : c ∈ I.map (H.bar : 𝒪 E →+* 𝒪 E)) (u : D.La) {w : D.La} (hw : w ∈ (I • ⊤ : Submodule (𝒪 E) D.La)) : (N : ℤ) ∣ D.EF u (c • w)

theorem **dvd_left** (D : Datum H) {N : ℕ} {I : Ideal (𝒪 E)} (hI : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) {c : 𝒪 E} (hc : c ∈ I.map (H.bar : 𝒪 E →+* 𝒪 E)) (v : D.La) {w : D.La} (hw : w ∈ (I.map (H.bar : 𝒪 E →+* 𝒪 E) • ⊤ : Submodule (𝒪 E) D.La)) : (N : ℤ) ∣ D.EF w (c • v)

theorem **exists_adjoint** (D : Datum H) (f : D.La →ₗ[𝒪 E] D.La) : ∃ f' : D.La →ₗ[𝒪 E] D.La, ∀ u v, D.EF (f' u) v = D.EF u (f v)

theorem **modN_of_dvd** (D : Datum H) {n : ℕ} (hn : 0 < n) {w : D.La} (h : ∀ v, (n : ℤ) ∣ D.EF w v) : D.ModN n w

theorem **dvd_of_modN** (D : Datum H) {n : ℕ} (u : D.La) {w : D.La} (h : D.ModN n w) : (n : ℤ) ∣ D.EF u w

theorem **mem_of_modN** (D : Datum H) {n : ℕ} {K : Ideal (𝒪 E)} (hK : (n : 𝒪 E) ∈ K) {w : D.La} (h : D.ModN n w) : w ∈ (K • ⊤ : Submodule (𝒪 E) D.La)

theorem **exists_quot_equiv** (D : Datum H) {n : ℕ} {K : Ideal (𝒪 E)} (hK : (n : 𝒪 E) ∈ K) (f g : D.La →ₗ[𝒪 E] D.La) (h : ∀ a, D.ModN n (g (f a) - a) ∧ D.ModN n (f (g a) - a)) : ∃ k : (D.La ⧸ (K • ⊤ : Submodule (𝒪 E) D.La)) ≃ₗ[𝒪 E] (D.La ⧸ (K • ⊤ : Submodule (𝒪 E) D.La)), ∀ a, k (Submodule.Quotient.mk a) = Submodule.Quotient.mk (f a)

theorem **zpow_congr** {F : Type} [Field F] {z : F} {N : ℕ} (hN : 0 < N) (hz : z ^ N = 1) {a b : ℤ} (h : (N : ℤ) ∣ a - b) : z ^ a = z ^ b

theorem **zeta_pow** {n : ℕ} (hn : 0 < n) : zeta n ^ n = 1

theorem **root_pow** {l : Level H} (c : Coeff l) {N : ℕ} (hN : N ∣ l.M) : (((zeta l.M', c.zeta_mem⟩ : c.F) ^ (l.M' / N)) ^ N = 1

theorem **cHalf_mem** (I : Ideal (𝒪 E)) : cHalf H I ∈ I.map (H.bar : 𝒪 E →+* 𝒪 E)

theorem **cSep_mem** (l : Level H) : cSep l ∈ l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)

theorem **qRep_spec** {D : Datum H} {K : Submodule (𝒪 E) D.La} (q : D.La ⧸ K) : (Submodule.Quotient.mk (qRep q) : D.La ⧸ K) = q

theorem **levRep_spec** {D : Datum H} {l : Level H} (q : LevQ D l) : (Submodule.Quotient.mk (levRep q) : LevQ D l) = q

theorem **half_prod** {l l₂ : Level H} {pd : ℕ} {I : Ideal (𝒪 E)} (hM : l₂.M = l.M * pd) (hb : l₂.Mb = l.Mb * I) : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(pd : 𝒪 E)}

theorem **datum_inv_char** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), d.l₂.M = l.M * d.pd → Nat.Coprime d.pd l.M → d.l₂.Mb = l.Mb * d.I → d.I ⊔ d.I.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤ → l.Dvd d.l₂ → ∀ (ph : (Fin n → LaJ Xo c d) → c.F) (ψ : (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) →* c.Fˣ) (x₀ : Fin n → LaJ Xo c d), d.PhiPin ph ψ x₀ → ∀ g : Sim Dw, g.InGam l d.l₂ → ∃ (kI : LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) (σq : LevQ Dw d.l₂ ≃ LevQ Dw d.l₂), (∀ a : Dw.La, (Submodule.Quotient.mk (g.f d.l₂.M a) : LaI Xo c d) = kI (Submodule.Quotient.mk a)) ∧ (∀ a, σq (Submodule.Quotient.mk a) = Submodule.Quotient.mk (g.f d.l₂.M a)) ∧ ∀ (x0 : Fin n → Dw.La) (q : Fin n → LevQ Dw d.l₂), d.w x0 (fun b => σq.symm (q b)) = d.eF (((ψ kI)⁻¹ : c.Fˣ) : c.F) * d.w x0 q

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsRankOneCuspDataLogBounds

theorem **exists_ext** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (Xx : DataX E r) (X : Data E r) (hX : Xx.toData = X) (S : Set E) (hΛ : Xx.Λ = Submodule.span (𝒪 E) {v | ∃ a ∈ S, ∃ u ∈ Xx.L', v = a • u}) (NA NC : ℕ) (hA : 0 < NA) (hC : 0 < NC) (hAi : ∀ a ∈ S, IsIntegral ℤ ((NA : E) * a)) (hCS : (NC : E) ∈ S) : ∃ e : XExt X, e.NA = NA ∧ e.NC = NC ∧ e.g = Xx.g ∧ e.k0 = Xx.k0

theorem **weight_bounds** (L f r n d : ℝ) (hL : 2 ≤ L) (hf : 1 ≤ f) (hfr : f ≤ r) (hrn : r ≤ n) (hd : 0 ≤ d) (hn : Real.log n ≤ L) (hD : Real.log (1 + d) ≤ 2 * L) : Real.log (d + f * n + 1) ≤ 3 * L ∧ Real.log (f * (d + f * n + 1) + 2) ≤ 5 * L ∧ ∀ s : ℝ, 0 ≤ s → s ≤ d + f * r * n / 2 → Real.log (1 + s) ≤ 3 * L

private theorem **level_M_eq_g_zero_mul_g_one_of_lawsM** {D : D7} (T : Core D) (h : Formalism.L2.Pj.LawsM T) : T.Q.l.M = D.host.g 0 * D.host.g 1

theorem **cusp_data_bounds** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∃ (e : ∀ k : (T.U j σ).Comp, XExt ((T.U j σ).X k)) (M wg : ℝ), 1 ≤ M ∧ Real.log M ≤ 30 * large₂.sc.L ∧ 0 < wg ∧ Real.log wg ≤ 3 * large₂.sc.L ∧ Real.log ((fdeg D.host.E : ℝ) * wg + 2) ≤ 5 * large₂.sc.L ∧ ∀ k, (∑ τ, ((T.U j σ).X k).spread (laU (T.U j σ) (wtA w) w.n k) τ) ≤ wg ∧ (∀ v ∈ (e k).L', herm ((T.U j σ).X k).Gm v v = 0 → ∀ b : D.host.E, ReciprocityTheory.ChartRestriction.G.FD.cc b = b → IsIntegral ℤ b → ∃ γ ∈ ((T.U j σ).X k).Γ, ((T.U j σ).X k).χ (laU (T.U j σ) (wtA w) w.n k) γ = 1 ∧ ∀ τ : ((T.U j σ).X k).Ψ, (γ : Matrix (Fin D.R.r) (Fin D.R.r) D.host.E).map τ.1 = ((T.U j σ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsRankOneCuspDataLogBoundsStatement

theorem **_root_.GlobalTheory.T.exists_rankOne_cuspData_log_bounds** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∃ (e : ∀ k : (T.U j σ).Comp, XExt ((T.U j σ).X k)) (M wg : ℝ), 1 ≤ M ∧ Real.log M ≤ 30 * large₂.sc.L ∧ 0 < wg ∧ Real.log wg ≤ 3 * large₂.sc.L ∧ Real.log ((fdeg D.host.E : ℝ) * wg + 2) ≤ 5 * large₂.sc.L ∧ ∀ k, (∑ τ, ((T.U j σ).X k).spread (laU (T.U j σ) (wtA w) w.n k) τ) ≤ wg ∧ (∀ v ∈ (e k).L', herm ((T.U j σ).X k).Gm v v = 0 → ∀ b : D.host.E,

ReciprocityTheory.ChartRestriction.G.FD.cc b = b → IsIntegral ℤ b → ∃ γ ∈ ((T.U j σ).X k).Γ, ((T.U j σ).X k).χ (laU (T.U j σ) (wtA w) w.n k) γ = 1 ∧ ∀ τ : ((T.U j σ).X k).Ψ, (γ : Matrix (Fin D.R.r) (

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsRawChartFramesBaseChange

private theorem **pull2_lemma_9** {Z' Y' Z Y : Scheme.{0}} (a : Z' ⟶ Y') (g : Y' ⟶ Y) (r : Z' ⟶ Z) (b : Z ⟶ Y) (hab : a ≫ g = r ≫ b) (U : Y.Opens) (V : Y'.Opens) (hV : V ≤ g ⁻¹ᵁ U) (W1 : Z'.Opens) (hW1 : W1 ≤ a ⁻¹ᵁ V) (V' : Z.Opens) (hV' : V' ≤ b ⁻¹ᵁ U) (W1' : Z'.Opens) (hW1' : W1' ≤ r ⁻¹ᵁ V') (W2 : Z'.Opens) (h12 : W2 ≤ W1) (h12' : W2 ≤ W1') (u : Γ(Y, U)ˣ) : Units.map (Z'.presheaf.map (homOfLE h12).op).hom.toMonoidHom (Units.map (Z'.presheaf.map (homOfLE hW1).op).hom.toMonoidHom (Units.map (a.app V).hom.toMonoidHom (Units.map (Y'.presheaf.map (homOfLE hV).op).hom.toMonoidHom (Units.map (g.app U).hom.toMonoidHom u)))) = Units.map (Z'.presheaf.map (homOfLE h12').op).hom.toMonoidHom (Units.map (Z'.presheaf.map (homOfLE hW1').op).hom.toMonoidHom (Units.map (r.app V').hom.toMonoidHom (Units.map (Z.presheaf.map (homOfLE hV').op).hom.toMonoidHom (Units.map (b.app U).hom.toMonoidHom u))))

theorem **exists_reindexed_pull** {X Y Y' : Scheme.{0}} {L : LB X} (hL : L.IsLB) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (c : LB.Gen (LB.LinComb L φ ε)) (hc : c.IsGen) (g : Y' ⟶ Y) (φ' : K → (Y' ⟶ X)) (h : φ' = fun k => g ≫ φ k) : ∃ c' : LB.Gen (LB.LinComb L φ' ε), c'.IsGen ∧ LB.Gen.Corr L _ _ _ _ c' (LB.pullGen L φ ε c g) ∧ ∀ (Z Z' : Scheme.{0}) (a : Z' ⟶ Y') (b : Z ⟶ Y) (r : Z' ⟶ Z), a ≫ g = r ≫ b → LB.Gen.Corr L _ _ _ _ (LB.pullGen L φ' ε c' a) (LB.pullGen L _ _ (LB.pullGen L φ ε c b) r)

theorem **spMap_comp_eq** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **sq_of_ring** {m n n' : ℕ} (f : Jet S (Fin n × Fin d) m →ₐ[S] Jet S (Fin n' × Fin d) m) (f' : Jet S' (Fin n × Fin d) m →ₐ[S'] Jet S' (Fin n' × Fin d) m) (h : f'.toRingHom.comp (jetBC ψ _ m) = (jetBC ψ _ m).comp f.toRingHom) : homAt (spMap ψ ≫ x0) f' ≫ rjBC ψ x0 d m n = rjBC ψ x0 d m n' ≫ homAt x0 f

private theorem **jetCopy_eq_lemma** {R : Type} [CommRing R] {d n : ℕ} (m : ℕ) (i : Fin n) : (jetCopy d m i : Jet R (Fin 1 × Fin d) m →ₐ[R] Jet R (Fin n × Fin d) m) = jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (i, p.2) : MvPolynomial (Fin n × Fin d) R))

theorem **sq_copy** {n : ℕ} (m : ℕ) (i : Fin n) : homAt (spMap ψ ≫ x0) (jetCopy d m i) ≫ rjBC ψ x0 d m 1 = rjBC ψ x0 d m n ≫ homAt x0 (jetCopy d m i)

theorem **sq_subst** {n n' : ℕ} (m : ℕ) (φ : Fin n × Fin d → MvPolynomial (Fin n' × Fin d) S) : homAt (spMap ψ ≫ x0) (jetSubstP m fun p => MvPolynomial.map ψ (φ p)) ≫ rjBC ψ x0 d m n = rjBC ψ x0 d m n' ≫ homAt x0 (jetSubstP m φ)

theorem **rjpt_bc** (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G) (hx : ∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) {n : ℕ} (m : ℕ) (i : Fin n) : rjpt (spMap ψ ≫ x0) d x' m i = rjBC ψ x0 d m n ≫ rjpt x0 d x m i

private theorem **padd_lemma_2** {T T' : Over Xo.X} (ρ : T' ⟶ T) (a b : T ⟶ Xo.univ.t.G) : ρ ≫ padd Xo.univ.t.G a b = padd Xo.univ.t.G (ρ ≫ a) (ρ ≫ b)

private theorem **pzero_lemma_3** {T T' : Over Xo.X} (ρ : T' ⟶ T) : ρ ≫ pzero Xo.univ.t.G T = pzero Xo.univ.t.G T'

private theorem **foldr_lemma_4** {T T' : Over Xo.X} (ρ : T' ⟶ T) {n : ℕ} (v : Fin n → (T ⟶ Xo.univ.t.G)) (v' : Fin n → (T' ⟶ Xo.univ.t.G)) (hv : ∀ k, v' k = ρ ≫ v k) (l : List (Fin n)) : ρ ≫ l.foldr (fun k acc => padd Xo.univ.t.G (v k) acc) (pzero Xo.univ.t.G T) = l.foldr (fun k acc => padd Xo.univ.t.G (v' k) acc) (pzero Xo.univ.t.G T')

theorem **cφ_comp** {T T' : Over Xo.X} (ρ : T' ⟶ T) (a b : T ⟶ Xo.univ.t.G) : cφ ![ρ ≫ a, ρ ≫ b] = fun I => ρ.left ≫ cφ ![a, b] I

theorem **exists_frames** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (e : ∀ m, RFrTy x0 d x m), (∀ m, (e m).IsGen) → ∃ e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m, (∀ m, (e' m).IsGen) ∧ (∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) ∧ ∀ (m n' : ℕ) (φ : Fin 2 → Fin d → MvPolynomial (Fin n' × Fin d) S), LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (rpe (spMap ψ ≫ x0) d x' e' m (fun i j => MvPolynomial.map

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsRawChartFramesBaseChangeStatement

theorem **_root_.GlobalTheory.T.exists_rawChart_frames_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (e : ∀ m, RFrTy x0 d x m), (∀ m, (e m).IsGen) → ∃ e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m, (∀ m, (e' m).IsGen) ∧ (∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) ∧ ∀ (m n' : ℕ) (φ : Fin 2 → Fin d → MvPolynomial (Fin n' × Fin d) S), LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (rpe (spMap ψ ≫ x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsReducedBasisIntegralPartOfLinearIndependen

theorem **exists_short_basis_of_ratSpan** {V E ι : Type*} [AddCommGroup V] [Module ℚ V] [NormedAddCommGroup E] [InnerProductSpace ℝ E] [FiniteDimensional ℝ E] (H : Submodule ℤ V) (φ : V →+ E) {ε : ℝ} (hε : 0 < ε) (hφ : ∀ x ∈ H, x ≠ 0 → ε ≤ ‖φ x‖) (v : ι → V) (hvH : ∀ j, v j ∈ H) (hsp : (H : Set V) ⊆ span ℚ (Set.range v)) {r : ℝ} (hr : ∀ j, ‖φ (v j)‖ ≤ r) {R : ℕ} (hR : R = finrank ℚ (span ℚ (Set.range v))) : ∃ b : Basis (Fin R) ℤ H, ∀ i, ‖φ (b i)‖ ^ 2 ≤ ((R + 3 : ℝ) / 4) * r ^ 2

theorem **mem_integralPart** {W : Submodule K (MvPolynomial σ K)} {P : MvPolynomial σ K} : P ∈ integralPart W ↔ P ∈ W ∧ ∀ α, IsIntegral ℤ (P.coeff α)

theorem **smul_mem_integralPart** {W : Submodule K (MvPolynomial σ K)} (c : 𝒪 K) {P : MvPolynomial σ K} (hP : P ∈ integralPart W) : c • P ∈ integralPart W

theorem **span_ringOfIntegers_basis_integralPart** {W : Submodule K (MvPolynomial σ K)} {ι : Type*} (b : Basis ι ℤ (integralPart W)) : (span (𝒪 K) (Set.range fun i => (b i : MvPolynomial σ K))).restrictScalars ℤ = integralPart W

theorem **norm_conjCoeffs_sq** [DecidableEq σ] {D : ℕ} {P : MvPolynomial σ K} (hP : P.totalDegree ≤ D) : ‖conjCoeffs K σ D P‖ ^ 2 = ∑ s : K →+* ℂ, l2Norm (map s P) ^ 2

theorem **finrank_le_one_add_pow** {W : Submodule K (MvPolynomial σ K)} {D : ℕ} (hD : ∀ P ∈ W, P.totalDegree ≤ D) : finrank K W ≤ (1 + D) ^ Fintype.card σ

theorem **exists_basis_integralPart_sum_l2Norm_sq_le** {κ : Type*} (W : Submodule K (MvPolynomial σ K)) {D : ℕ} (hD : ∀ P ∈ W, P.totalDegree ≤ D) (S : Set (MvPolynomial σ K)) (hSH : S ⊆ integralPart W) (hSW : span K S = W) {B : ℝ} (hB : ∀ P ∈ S, ∀ s : K →+* ℂ, l2Norm (map s P) ≤ B) (ω : κ → 𝒪 K) (hω : span ℚ (Set.range fun k => (ω k : K)) = ⊤) {A : ℝ} (hA : ∀ (s : K →+* ℂ) k, ‖s (ω k)‖ ≤ A) {R : ℕ} (hR : R = finrank ℚ K * finrank K W) : ∃ b : Basis (Fin R) ℤ (integralPart W), ∀ i, √(∑ s : K →+* ℂ, l2Norm (map s (b i : MvPolynomial σ K)) ^ 2) ≤ √((R + 3) / 4) * √(finrank ℚ K) * A * B

theorem **exists_reduced_basis_integralPart** {κ : Type*} (W : Submodule K (MvPolynomial σ K)) {D : ℕ} (hD : ∀ P ∈ W, P.totalDegree ≤ D) (S : Set (MvPolynomial σ K)) (hSH : S ⊆ integralPart W) (hSW : span K S = W) {B : ℝ} (hB : ∀ P ∈ S, ∀ s : K →+* ℂ, l1Norm (map s P) ≤ B) (ω : κ → 𝒪 K) (hω : span ℚ (Set.range fun k => (ω k : K)) = ⊤) {A : ℝ} (hA : ∀ (s : K →+* ℂ) k, ‖s (ω k)‖ ≤ A) {R : ℕ} (hR : R = finrank ℚ K * finrank K W) : R ≤ finrank ℚ K * (1 + D) ^ Fintype.card σ ∧ ∃ b : Basis (Fin R) ℤ (integralPart W), (∀ i (s : K →+* ℂ), l2Norm (map s (b i : MvPolynomial σ K)) ≤ √((R + 3) / 4) * √(finrank ℚ K) * A * B) ∧ (∀ i (s : K →+* ℂ), l1Norm (map s (b i : MvPolynomial σ K)) ≤ (1 + D : ℝ) ^ ((Fintype.card σ : ℝ) / 2) * (√((R + 3) / 4) * √(finrank ℚ K) * A * B)) ∧ (span (𝒪 K) (Set.range fun i => (b i : MvPolynomial σ K))).restrictScalars ℤ = integralPart W

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsReducedIntegralBasisHarmonicPolynomials

theorem **mem_HHa_of_mem_span** {T : Type} {r : T → ℕ} {n : ℕ} {k : Type} [Field k] {cc : T → T} {isX : T → Prop} {a : T → ℕ → ℕ} {S : Set (MvPolynomial (Letter T r n) k)} (hS : S ⊆ HHa r n k cc isX a) {P : MvPolynomial (Letter T r n) k} (hP : P ∈ Submodule.span k S) : P ∈ HHa r n k cc isX a

theorem **totalDegree_le_of_rowDeg** {T : Type} [Fintype T] {r : T → ℕ} {n : ℕ} {k : Type} [Field k] {a : T → ℕ → ℕ} {P : MvPolynomial (Letter T r n) k} (hP : RowDeg r n k a P) : P.totalDegree ≤ ∑ σ, ∑ i : Fin (r σ), a σ i

theorem **l1At_eq_l1Norm_map** {K ι : Type} [Field K] (σ : K →+* ℂ) (P : MvPolynomial ι K) : l1At σ P = MvPoly.l1Norm (map σ P)

theorem **mult_add_mult_conjugate** (Dw : Datum H) (τ : E →+* ℂ) : Dw.mult τ + Dw.mult (ComplexEmbedding.conjugate τ) = Dw.p + Dw.q

theorem **card_LtN** (Dw : Datum H) (n : ℕ) : Fintype.card (LtN Dw n) = FD.fdeg E * (Dw.p + Dw.q) * n

theorem **bound_le_exp** {R d D N : ℕ} {L x Δ : ℝ} (hR : 1 ≤ R) (hd : 1 ≤ d) (hΔ : 0 < Δ) (hle : R ≤ d * (1 + D) ^ N) (hL : Real.log (1 + D) ≤ 2 * L) : (1 + D : ℝ) ^ ((N : ℝ) / 2) * (√((R + 3) / 4) * √d * ((d : ℝ) ^ 3 * Δ ^ (1 / (d : ℝ))) * Real.exp x) ≤ Real.exp (x + 2 * N * L + 4 * Real.log d + Real.log Δ / d)

theorem **exists_reduced_basis_harmA** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n : ℕ), Dw.p + Dw.q ≤ n → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (L : ℝ), 2 ≤ L → Real.log (FD.rdE E) ≤ 2 * L → Real.log (FD.fdeg E) ≤ 2 * L → Real.log (1 + DF Dw a) ≤ 2 * L → Real.log n ≤ L → ∃ (R : ℕ) (b : Fin R → MvPolynomial (LtN Dw n) c.F), (∀ i, b i ∈ harmA (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsReducedIntegralBasisHarmonicPolynomialsStatement

theorem **_root_.GlobalTheory.T.exists_reduced_integral_basis_harmonicPolynomials** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n : ℕ), Dw.p + Dw.q ≤ n → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (L : ℝ), 2 ≤ L → Real.log (FD.rdE E) ≤ 2 * L → Real.log (FD.fdeg E) ≤ 2 * L → Real.log (1 + DF Dw a) ≤ 2 * L → Real.log n ≤ L → ∃ (R : ℕ) (b : Fin R → MvPolynomial (LtN Dw n) c.F), (∀ i, b i ∈ harmA (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsRunDataGeneratingScaledThetaLatticeWeight

theorem **exists_tiedW_rec** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (Gc Prw : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → ThInt (recPJ h j hj) hinj EX wL → EtaleWeightGaps.GrowthConstantBound large₂ (recB h.toLawsCore) Gc → 38 * large₂.sc.L + PO D large₂ ≤ Prw → ∃ G : EtaleWeightGaps.GlobalFamilyData (recPJ h j hj) ka wL wF, G.TiedW (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) Gc Prw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsRunDataGeneratingScaledThetaLatticeWeightStatement

theorem **_root_.GlobalTheory.T.exists_runData_generating_scaled_thetaLattice_weight** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (Gc Prw : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → ThInt (recPJ h j hj) hinj EX wL → EtaleWeightGaps.GrowthConstantBound large₂ (recB h.toLawsCore) Gc → 38 * large₂.sc.L + PO D large₂ ≤ Prw → ∃ G : EtaleWeightGaps.GlobalFamilyData (recPJ h j hj) ka wL wF, G.TiedW (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) Gc Prw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsScalarEvaluationsCuspFormsFiniteNet

private theorem **linearIndependent_inclusion_of_le** {K R ι : Type} [Field K] [AddCommGroup R] [Module K R] {A B : Submodule K R} (hAB : A ≤ B) (F : ι → A) (hF : LinearIndependent K F) : LinearIndependent K (fun i => ((F i, hAB (F i).2) : B))

theorem **exists_scalar_readings_on_net** : ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → 16 ≤ D.R.f → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∃ (n : ℕ) (ev : Fin n → ((recB h.toLawsCore).Cusp j w →ₗ[σ] ℂ)), (∀ F, ∃ y, (recB h.toLawsCore).supN j w σ F / 4 ≤ ‖ev y F‖) ∧ (∀ y F, ‖ev y F‖ ≤ (recB h.toLawsCore).supN j w σ F) ∧ Real.log (16 * (n : ℝ)) ≤ 1041 * Zb D large₂ ^ 3 ∧ ∀ (m : ℕ) (F : Fin m → (recB h.toLawsCore).Cusp j w), LinearIndependent T.Q.c.F F → LinearIndependent ℂ (fun i => fun y => ev y (F i))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsScalarEvaluationsCuspFormsFiniteNetStatement

theorem **_root_.GlobalTheory.T.exists_scalar_evaluations_cusp_forms_finite_net** : ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → 16 ≤ D.R.f → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∃ (n : ℕ) (ev : Fin n → ((recB h.toLawsCore).Cusp j w →ₗ[σ] ℂ)), (∀ F, ∃ y, (recB h.toLawsCore).supN j w σ F / 4 ≤ ‖ev y F‖) ∧ (∀ y F, ‖ev y F‖ ≤ (recB h.toLawsCore).supN j w σ F) ∧ Real.log (16 * (n : ℝ)) ≤ 1041 * Zb D large₂ ^ 3 ∧ ∀ (m : ℕ) (F : Fin m → (recB h.toLawsCore).Cusp j w), LinearIndependent T.Q.c.F F → LinearIndependent ℂ (fun i => fun y => ev y (F i))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSecondPartialLaw

theorem **prod_cube3** {X0 : Scheme.{0}} (G : Over X0) [GrpObj G] [IsCommMonObj G] {T : Over X0} {M : Type} [CommGroup M] (h : (T ⟶ G) → M) (xs : Fin 3 → (T ⟶ G)) : ∏ I : thK, h (psum G xs I.down) ^ thε I = (h 1)⁻¹ * h (xs 0 * 1) * h (xs 1 * 1) * h (xs 2 * 1) * (h (xs 0 * (xs 1 * 1)))⁻¹ * (h (xs 0 * (xs 2 * 1)))⁻¹ * (h (xs 1 * (xs 2 * 1)))⁻¹ * h (xs 0 * (xs 1 * (xs 2 * 1)))

theorem **subst_copy** (m : ℕ) {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (i : Fin 2) : (jetSubstP m (fun p : Fin 2 × Fin d => φ p.1 p.2)).comp (jetCopy d m i) = jetSubstP m (fun p : Fin 1 × Fin d => φ i p.2)

theorem **substX** (m : ℕ) {n : ℕ} (k : Fin n) : jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (k, p.2) : MvPolynomial (Fin n × Fin d) S)) = jetCopy d m k

theorem **ren_copy** (m : ℕ) {n n' : ℕ} (r : Fin n → Fin n') (i : Fin n) : (jetRen (S

theorem **ren_one** (m : ℕ) {n' : ℕ} (r : Fin 1 → Fin n') : jetRen (S

theorem **subst_psIn** (m : ℕ) {n n' : ℕ} (r : Fin n → Fin n') (g : Fin d → MvPowerSeries (Fin n × Fin d) S) : jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (g p.2)) = (jetRen d m r).comp (jetSubstP m (fun p : Fin 1 × Fin d => psTrunc m (g p.2)))

theorem **pt_sub** (m : ℕ) {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (i : Fin 2) : homAt x0 (jetSubstP m (fun p : Fin 2 × Fin d => φ p.1 p.2)) ≫ jptG t x0 d x m i = homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => φ i p.2)) ≫ x m

theorem **pt_X** (m : ℕ) {n : ℕ} (k : Fin n) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (k, p.2) : MvPolynomial (Fin n × Fin d) S))) ≫ x m = jptG t x0 d x m k

theorem **pt_law** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (r : Fin 2 → Fin n) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => psIn d m r (F p.2))) ≫ x m = jptG t x0 d x m (r 0) * jptG t x0 d x m (r 1)

theorem **pe_pt** (m : ℕ) {n : ℕ} (σ : Jet S (Fin 2 × Fin d) m →ₐ[S] Jet S (Fin n × Fin d) m) (I : ULift.{0} (Finset (Fin 2))) : (homAt x0 σ).left ≫ cφG t ![jptG t x0 d x m (0 : Fin 2), jptG t x0 d x m (1 : Fin 2)] I = (psum t.G ![homAt x0 σ ≫ jptG t x0 d x m (0 : Fin 2), homAt x0 σ ≫ jptG t x0 d x m (1 : Fin 2)] I.down).left

theorem **th_pt** {T : Over X0} (a b c : T ⟶ t.G) (I : thK) : (lift a (lift b c)).left ≫ thφ t.G (x3 t.G) I = (psum t.G ![a, b, c] I.down).left

theorem **ofMul_eq** {M : Type} [CommGroup M] {a b : M} (h : Additive.ofMul a = Additive.ofMul b) : a = b

theorem **prod_cube2_swap** {T : Over X0} {M : Type} [CommGroup M] (h : (T ⟶ t.G) → M) (xs : Fin 2 → (T ⟶ t.G)) : ∏ I : ULift (Finset (Fin 2)), h (psum t.G xs I.down) ^ cε 2 I = h 1 * (h (xs 0 * 1))⁻¹ * (h (xs 1 * 1))⁻¹ * h (xs 1 * (xs 0 * 1))

theorem **jpt_trunc_ring** (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) {n : ℕ} (i : Fin n) : homAt x0 (jetTrunc : Jet S (Fin n × Fin d) (m + 1) →ₐ[S] _) ≫ jptG t x0 d x (m + 1) i = jptG t x0 d x m i

theorem **triplePt_trunc_second** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {X0 : Scheme.{0}} {t : SAB H X0} {S : Type} [CommRing S] {x0 : Sp S ⟶ X0} {d : ℕ} {x : ∀ m, jOG x0 d m 1 ⟶ t.G} {F : Fin d → MvPowerSeries (Fin 2 × Fin d) S} {act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S} {κ : Type} [Fintype κ] {tg : κ → (Sp (DualNumber S) ⟶ t.G.left)} {uvec : κ → Fin d → S} (hch : IsChartG t x0 d F act tg uvec x) (m : ℕ) (i j k : Fin 3) : (ρn x0 d 3 m).left ≫ (lift (jptG t x0 d x (m + 1) i) (lift (jptG t x0 d x (m + 1) j) (jptG t x0 d x (m + 1) k))).left = 𝟙 _ ≫ (lift (jptG t x0 d x m i) (lift (jptG t x0 d x m j) (jptG t x0 d x m k))).left

theorem **exists_second_partial_law** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ ph2 : PS S 3 d, ∀ m, ∃ u : (Jet S (Fin 3 × Fin d) m)ˣ, (u : Jet S (Fin 3 × Fin d) m) = psJet m ph2 ∧ LB.Gen.Corr t.PP.L _ _ _ _ (LB.pullGen t.PP.L (thφ t.G (x3 t.G)) thε t.cub.t (lift (jptG t x0 d x m (1 : Fin 3)) (lift (jptG t x0 d x m (2 : Fin 3)) (jptG t x0 d x m (0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSecondPartialLawStatement

theorem **_root_.GlobalTheory.T.exists_second_partial_law_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ ph2 : PS S 3 d, ∀ m, ∃ u : (Jet S (Fin 3 × Fin d) m)ˣ, (u : Jet S (Fin 3 × Fin d) m) = psJet m ph2 ∧ LB.Gen.Corr t.PP.L _ _ _ _ (LB.pullGen t.PP.L (thφ t.G (x3 t.G)) thε t.cub.t (lift (jptG t x0 d x m (1 : Fin 3)) (lift (jptG t x0 d x m (2 : Fi

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShiftNullValNeZero

theorem **mk_eq_zero_iff** (k : ℕ) (m : Dw.La) : (Submodule.Quotient.mk m : LatMod Dw k) = 0 ↔ ∃ m' : Dw.La, k • m' = m

theorem **exists_shift** {R : ℕ} (hR : R ≠ 0) (q : LatMod Dw (R ^ 4)) (hq : R • q = 0) : ∃ u : LatMod Dw R, (Submodule.Quotient.mk (R ^ 3 • qRep u) : LatMod Dw (R ^ 4)) = q

theorem **gen_ne_zero** {K : Type} [CommRing K] [Nontrivial K] [Algebra c.F K] (y : FrPt Xo c K) : (SecM.ofSec Xo.univ.t.G y.s : SecM Xo.univ.t.G Xo.univ.t.PP.L 1 (Over.mk y.x)) ≠ 0

theorem **exists_nullVal_ne_zero** (hL : Laws) {K : Type} [Field K] [IsAlgClosed K] [Algebra c.F K] (y : FrPt Xo c K) (N : ℕ) {n : ℕ} (hn : 2 ≤ n) (a : ptO Xo y.x ⟶ Xo.univ.t.G) : ∃ z : ptO Xo y.x ⟶ Xo.univ.t.G, z ^ n = 1 ∧ ∀ p w, IsNullVal Xo c N y p w → p ^ N = a * z → w ≠ 0

theorem **exists_shift_nullVal_ne_zero** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (R : ℕ), Odd R → 1 < R → ∀ (K : Type) [Field K] [IsAlgClosed K] [Algebra c.F K] (y : FrPt Xo c K) (ev : LatMod Dw (R ^ 4) → (ptO Xo y.x ⟶ Xo.univ.t.G)) (w : LatMod Dw (R ^ 4) → K), IsFullLev Xo (R ^ 4) y.x ev → (∀ q, IsNullVal Xo c (R ^ 2) y (ev q) (w q)) → ∀ (n : ℕ) (x₀ : Fin n → Dw.La) (g : LevGrp Dw (R ^ 4)), ∃ u : Fin n → LatMod Dw R, ∀ b, w (g (shiftFam R x₀ u b)) ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortFiniteGeneratorsThetaLattice

theorem **span_eq_spanIn** {D : D7} {K : Type} [Field K] [NumberField K] {R : ℕ → Type} [∀ j, CommRing (R j)] [∀ j, Algebra K (R j)] [∀ j, Algebra (𝒪 K) (R j)] [∀ j, IsScalarTower (𝒪 K) K (R j)] (C : ∀ j, WeightFamilies.Wt D j → Submodule K (R j)) (j : ℕ) (w : WeightFamilies.Wt D j) (s : Set (R j)) {ι : Type} (g : ι → cutR C j w) (hg : ∀ i, (g i : R j) ∈ s) (hs : ∀ f ∈ s, f ∈ AddSubgroup.closure (Set.range fun i => (g i : R j))) : Submodule.span (𝒪 K) (Set.range g) = spanIn C j w s

theorem **exists_family** (hci : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2) {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c) (hn : IsNab Xo c nab) (ht : ThValTot Xo c nab) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) {R : ℕ} (b : Fin R → MvPolynomial (LtN Dw n) c.F) (hb : ∀ i, b i ∈ harmA (Dw

theorem **log_rdE** (D : D7) : Real.log (ReciprocityTheory.ChartRestriction.G.FD.rdE D.host.E) = lrd D

theorem **absA_eq** {D : D7} {j : ℕ} (w : WeightFamilies.Wt D j) : absA w = (WeightAlgebra.DF w : ℝ)

theorem **theta_family** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → (fun (B

: WeightFamilies.Prov D) (X : PJ D B T.G.P j) => ∃ (N : ℕ) (w0 : Fin N → B.Cusp j w), Submodule.span (𝒪 B.KO) (Set.range w0) = X.thAll w ∧ N ≤ (D.host.g 0 * D.host.g 1) ^ (D.R.f * D.R.r * w.n) * (Module.finrank ℚ B.KO * (1 + WeightAlgebra.DF w) ^ (D.R.f * D.R.r * w.n)) ∧ ∀ k σ, Real.log (B.supN j w σ (w0 k)) ≤ 400 * Zb D large₂ * w.size + 3 * D.R.f * D.R.r * w.n * large₂.sc.L + 5 * Real.log (Module.finrank ℚ B.KO) + Real.log |((NumberField.discr B.KO : ℤ) : ℝ)| / Module.finrank ℚ B.KO + 2) (recB h.toLawsCore) (recPJ h j hj)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortFiniteGeneratorsThetaLatticeStatement

theorem **_root_.GlobalTheory.T.exists_short_finite_generators_thetaLattice** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → (fun (B : WeightFamilies.Prov D) (X : PJ D B T.G.P j) => ∃ (N : ℕ) (w0 : Fin N → B.Cusp j w), Submodule.span (𝒪 B.KO) (Set.range w0) = X.thAll w ∧ N ≤ (D.host.g 0 * D.host.g 1) ^ (D.R.f * D.R.r * w.n) * (Module.finrank ℚ B.KO * (1 + WeightAlgebra.DF w) ^ (D.R.f * D.R.r * w.n)) ∧ ∀ k σ, Real.log (B.supN j w σ (w0 k)) ≤ 400 * Zb D large₂ * w.size + 3 * D.R.f * D.R.r * w.n * large₂.sc.L + 5 * Real.log (Module.finrank ℚ B.KO) + Real.log |((NumberField.discr B.KO : ℤ) : ℝ)| / Module.finrank ℚ B.KO + 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortHarmonicFamilyFewColumnsLinear

private theorem **perpVec_apply_eq_zero** {R V : Type} [CommRing R] [AddCommGroup V] [Module R V] {m : ℕ} (h : V →ₗ[R] (V → R)) (e e' : Fin m → V) (g : V) (c : Fin m) : h (perpVec h e e' g) (e c) = 0

theorem **exists_short_harmonic_family_few_columns** (hfg : Λ.FG) (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) → x ∈ Λ) {p q : ℕ} (hn : p + q ≤ n) {T : Type*} [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ha : 1 ≤ typeDegree c c') : ∃ S : Finset ((Fin p → Fin n → E) × (Fin q → Fin n → E)), (∀ s ∈ S, (∀ a, s.1 a ∈ Λ) ∧ (∀ b, s.2 b ∈ Λ) ∧ ∀ a b, herm 1 (s.2 b) (s.1 a) = 0) ∧ (∀ (K : Type) [Field K] [CharZero K] (τ : T → E →+* K), (Function.Injective fun s : T × Bool => fun x : E => τ s.1 (if s.2 then cc x else x)) → Submodule.span K ((fun s : (Fin p → Fin n → E) × (Fin q → Fin n → E) => minorProd c c' (fun t a k => τ t (cc (s.1 a k))) fun t b k => τ t (s.2 b k)) '' S) = rowHighest K c c') ∧ (∀ (τ : T → E →+* ℂ), ∀ s ∈ S, l1Norm (minorProd c c' (fun t a k => τ t (cc (s.1 a k))) fun t b k => τ t (s.2 b k)) ≤ (((p + q : ℕ) : ℝ) *

private theorem **det_mem_subalgebra_of_entries_mem'** {A B : Type*} [CommRing A] [CommRing B] [Algebra A B] (S : Subalgebra A B) {m : ℕ} (M : Matrix (Fin m) (Fin m) B) (hM : ∀ i j, M i j ∈ S) : M.det ∈ S

private theorem **minorProd_mem_Ha_of_herm_orthogonal** {K : Type} [Field K] {T : Type*} [Fintype T] {p q : ℕ} (τ : T → E →+* K) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (e : Fin p → Fin n → E) (f : Fin q → Fin n → E) (he : ∀ a, e a ∈ Λ) (hf : ∀ b, f b ∈ Λ) (hef : ∀ a b, herm 1 (e a) (f b) = 0) : minorProd c c' (fun t a k => τ t (e a k)) (fun t b k => τ t (cc (f b k))) ∈ Ha τ c c' Λ

theorem **exists_short_harmonic_family_few_columns_linear** (hfg : Λ.FG) (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) → x ∈ Λ) {p q : ℕ} (hn : p + q ≤ n) {T : Type*} [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ha : 1 ≤ typeDegree c c') : ∃ S : Finset ((Fin p → Fin n → E) × (Fin q → Fin n → E)), (∀ s ∈ S, (∀ a, s.1 a ∈ Λ) ∧ (∀ b, s.2 b ∈ Λ) ∧ ∀ a b, herm 1 (s.1 a) (s.2 b) = 0) ∧ (∀ (K : Type) [Field K] (τ : T → E →+* K), ∀ s ∈ S, (minorProd c c' (fun t a k => τ t (s.1 a k)) fun t b k => τ t (cc (s.2 b k))) ∈ Ha τ c c' Λ) ∧ (∀ (K : Type) [Field K] [CharZero K] (τ : T → E →+* K), (Function.Injective fun s : T × Bool => fun x : E => τ s.1 (if s.2 then cc x else x)) → Submodule.span K ((fun s : (Fin p → Fin n → E) × (Fin q → Fin n → E) => minorProd c c' (fun t a k => τ t (s.1 a k)) fun t b k => τ t (cc (s.2 b k))) '' S) = rowHig

theorem **short_harmonic_family_few_columns** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (n : ℕ) (Λ : Submodule (𝒪 E) (Fin n → E)), Λ.FG → (∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) → (∀ x, (∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) → x ∈ Λ) → ∀ (p q : ℕ), p + q ≤ n → ∀ (T : Type) [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ), 1 ≤ typeDegree c c' → ∃ S : Finset ((Fin p → Fin n → E) × (Fin q → Fin n → E)), (∀ (K : Type) [Field K] (τ : T → E →+* K), ∀ s ∈ S, (minorProd c c' (fun t a k => τ t (s.1 a k)) fun t b k => τ t (cc (s.2 b k))) ∈ Ha τ c c' Λ) ∧ (∀ (K : Type) [Field K] [CharZero K] (τ : T → E →+* K), (Function.Injective fun s : T × Bool => fun x : E => τ s.1 (if s.2 then cc x else x)) → Submodule.span K ((fun s : (Fin p → Fin n → E) × (Fin q → Fin n → E) => minorProd c c' (fun t a k => τ t (s.1 a k)) fun t b k => τ t (cc (s.2 b k))) '' S) = rowHighest K c c') ∧ (∀ (τ : T

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortHarmonicFamilyLinear

@[simp] theorem **conjVec_apply** (v : Fin n → E) (k : Fin n) : conjVec E n v k = cc (v k)

@[simp] theorem **_root_.GlobalTheory.Harm.conjVec_conjVec** (v : Fin n → E) : _root_.GlobalTheory.Harm.conjVec E n (_root_.GlobalTheory.Harm.conjVec E n v) = v

theorem **_root_.GlobalTheory.Harm.herm_conjVec** (v w : Fin n → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 (_root_.GlobalTheory.Harm.conjVec E n v) (_root_.GlobalTheory.Harm.conjVec E n w) = _root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 w v

theorem **_root_.GlobalTheory.Harm.mem_conjLattice** {v : Fin n → E} : v ∈ _root_.GlobalTheory.Harm.conjLattice Λ ↔ _root_.GlobalTheory.Harm.conjVec E n v ∈ Λ

@[simp] theorem **conjLattice_conjLattice** : conjLattice (conjLattice Λ) = Λ

theorem **_root_.GlobalTheory.Harm.conjLattice_selfDual** (hfg : Λ.FG) (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) → x ∈ Λ) : (_root_.GlobalTheory.Harm.conjLattice Λ).FG ∧ (∀ x ∈ _root_.GlobalTheory.Harm.conjLattice Λ, ∀ y ∈ _root_.GlobalTheory.Harm.conjLattice Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) ∧ ∀ x, (∀ y ∈ _root_.GlobalTheory.Harm.conjLattice Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) → x ∈ _root_.GlobalTheory.Harm.conjLattice Λ

theorem **_root_.GlobalTheory.Harm.one_le_rdE** : 1 ≤ _root_.ReciprocityTheory.ChartRestriction.G.FD.rdE E

theorem **_root_.GlobalTheory.Harm.one_le_fdeg** : 1 ≤ _root_.ReciprocityTheory.ChartRestriction.G.FD.fdeg E

theorem **_root_.GlobalTheory.Harm.one_le_add_of_one_le_typeDegree** {T : Type*} [Fintype T] {p q : ℕ} {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} (ha : 1 ≤ _root_.GlobalTheory.Harm.typeDegree c c') : 1 ≤ p + q

theorem **_root_.GlobalTheory.Harm.perpVec_hermPairing_mem** (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) {p : ℕ} {e e' : Fin p → Fin n → E} {g : Fin n → E} (he : ∀ a, e a ∈ Λ) (he' : ∀ a, e' a ∈ Λ) (hg : g ∈ Λ) : _root_.ReciprocityTheory.ChartRestriction.G.TB.perpVec (_root_.GlobalTheory.Harm.hermPairing E n) e e' g ∈ Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortHarmonicFamilyUnimodular

theorem **mem_iff_herm_of_unimod** {n : ℕ} {Λ : GenLat E n} (h : Λ.Unimod) (x : Fin n → E) : x ∈ Λ.La ↔ ∀ y ∈ Λ.La, IsIntegral ℤ (FD.herm 1 x y)

theorem **not_inPsi_and_conjugate** (H : Host E) (τ : E →+* ℂ) : ¬ (InPsi H τ ∧ InPsi H (ComplexEmbedding.conjugate τ))

private theorem **coe_embF** {H : Host E} {l : Level H} (c : Coeff l) (τ : E →+* ℂ) (x : E) : ((embF c τ x : c.F) : ℂ) = τ x

theorem **injective_embF_inPsi** {H : Host E} {l : Level H} (c : Coeff l) : Function.Injective fun s : {τ : E →+* ℂ // InPsi H τ} × Bool => fun x : E => embF c s.1.1 (if s.2 then FD.cc x else x)

theorem **sum_norm_coeff_eq_l1Norm** {K σ σ' : Type} [Field K] (e : σ ≃ σ') (f : K →+* ℂ) (Q : MvPolynomial σ K) : ∑ m ∈ Q.support, ‖f (Q.coeff m)‖ = MvPoly.l1Norm (MvPolynomial.map f (rename e Q))

theorem **log_le_of_le_of_log_le** {x y B : ℝ} (hx : 0 ≤ x) (h : x ≤ y) (hy : Real.log y ≤ B) (hB : 0 ≤ B) : Real.log x ≤ B

theorem **exists_short_harmL_family** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n : ℕ) (Λ : GenLat E n), Λ.Unimod → Λ.La.FG → Dw.p + Dw.q ≤ n → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (L : ℝ), 2 ≤ L → Real.log (FD.rdE E) ≤ 2 * L → Real.log (FD.fdeg E) ≤ 2 * L → Real.log (1 + DF Dw a) ≤ 2 * L → Real.log n ≤ L → ∃ S : Finset (MvPolynomial (LtN Dw n) c.F), (∀ Q ∈ S, Q ∈ harmL c Λ a ∧ ∀ σ' : c.F →+* ℂ, Real.log (∑ m ∈ Q.support, ‖σ' (Q.coeff m)‖) ≤ 140 * (Dw.p + Dw.q : ℕ) * L * DF Dw a) ∧ (∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), P ∈ Submodule.span c.F (S : Set (MvPolynomial (LtN Dw n) c.F))) ∧ S.card ≤ ((2 * (Dw.p + Dw.q) + 1) * DF Dw a + 1) ^ (2 * FD.fdeg E * n * (2 * Dw.p + Dw.q))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortHarmonicFamilyUnimodularStatement

theorem **_root_.GlobalTheory.T.exists_short_harmonic_family_unimodular** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n : ℕ) (Λ : GenLat E n), Λ.Unimod → Λ.La.FG → Dw.p + Dw.q ≤ n → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (L : ℝ), 2 ≤ L → Real.log (FD.rdE E) ≤ 2 * L → Real.log (FD.fdeg E) ≤ 2 * L → Real.log (1 + DF Dw a) ≤ 2 * L → Real.log n ≤ L → ∃ S : Finset (MvPolynomial (LtN Dw n) c.F), (∀ Q ∈ S, Q ∈ harmL c Λ a ∧ ∀ σ' : c.F →+* ℂ, Real.log (∑ m ∈ Q.support, ‖σ' (Q.coeff m)‖) ≤ 140 * (Dw.p + Dw.q : ℕ) * L * DF Dw a) ∧ (∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), P ∈ Submodule.span c.F (S : Set (MvPolynomial (LtN Dw n) c.F))) ∧ S.card ≤ ((2 * (Dw.p + Dw.q) + 1) * DF Dw a + 1) ^ (2 * FD.fdeg E * n * (2 * Dw.p + Dw.q))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortIndependentIntegers

private lemma **sum_embeddings_eq_zero_of_conjugate_neg** {K : Type} [Field K] [NumberField K] (f : (K →+* ℂ) → ℝ) (hf : ∀ σ, f (NumberField.ComplexEmbedding.conjugate σ) = - f σ) : ∑ σ, f σ = 0

private lemma **inner_realCoords_one_neg_one_eq_trace** {K : Type} [Field K] [NumberField K] {M : ℕ} (e : Fin M ≃ (K →+* ℂ)) (x y : K) : inner ℝ (_root_.GlobalTheory.ShortIntegers.realCoords e 1 x) (_root_.GlobalTheory.ShortIntegers.realCoords e (-1) y) = ((Algebra.trace ℚ K (x * y) : ℚ) : ℝ)

private lemma **norm_sq_realCoords_eq_sum_norm_sq** {K : Type} [Field K] [NumberField K] {M : ℕ} (e : Fin M ≃ (K →+* ℂ)) {ε : ℝ} (hε : ε ^ 2 = 1) (x : K) : ‖_root_.GlobalTheory.ShortIntegers.realCoords e ε x‖ ^ 2 = ∑ σ : K →+* ℂ, ‖σ x‖ ^ 2

theorem **_root_.GlobalTheory.exists_short_independent_integers** (K : Type) [Field K] [NumberField K] : ∃ ω : Fin (Module.finrank ℚ K) → 𝒪 K, LinearIndependent ℚ (fun k => (ω k : K)) ∧ ∀ (s : K →+* ℂ) k, ‖s (ω k : K)‖ ≤ (Module.finrank ℚ K : ℝ) ^ 3 * |((NumberField.discr K : ℤ) : ℝ)| ^ (1 / (Module.finrank ℚ K : ℝ))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortSpanThetaTieOfScalars

theorem **exists_short_span_thetaTie_of_scalars** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (large₂ : SizeData₂ D) (C1 A1 Gc : ℝ) (S : Scal B) (ka wL wF : WeightFamilies.Wt D j) (Tn : ℕ), ScalB large₂ X C1 A1 S ka wL wF Tn → EtaleWeightGaps.GrowthConstantBound large₂ B Gc → ∃ Sh : Finset (B.Cusp j wF), Submodule.span B.KO (Sh : Set (B.Cusp j wF)) = ⊤ ∧ (∀ g ∈ Sh, g ∈ X.Thc S.sF wF ∧ WeightFamilies.meanLogSup B g ≤ Gc * wF.size) ∧ ∀ G : EtaleWeightGaps.GlobalFamilyData X ka wL wF, (∀ i, G.th' i ∈ Sh) → G.TiedTh S Gc

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortThetaFamilyWithBlockPolynomialRules

theorem **span_sum_elim** {R M ι κ : Type} [CommRing R] [AddCommGroup M] [Module R M] (S : Submodule R M) (u : ι → M) (v : κ → M) (hu : Submodule.span R (Set.range u) = S) (hv : ∀ k, v k ∈ S) : Submodule.span R (Set.range (Sum.elim u v)) = S

private theorem **span_range_sum_elim_eq_thAll** {D : D7} {Pr : ModuliIntegralCharts.Prov D} {j : ℕ} (B : WeightFamilies.Prov D) (X : PJ D B Pr j) (w : WeightFamilies.Wt D j) {N M' : ℕ} {ι : Type} (w0 : Fin N → B.Cusp j w) (v : ι → B.Cusp j w) (e : Fin M' ≃ Fin N ⊕ ι) (h0 : Submodule.span (𝒪 B.KO) (Set.range w0) = X.thAll w) (hv : ∀ q, v q ∈ X.thAll w) : Submodule.span (𝒪 B.KO) (Set.range (fun k => Sum.elim w0 v (e k))) = X.thAll w

theorem **theta_family_block** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → D.R.f * D.R.r * (D.R.r + 1) ≤ 2 * w.n → ∃ (N : ℕ) (w0 : Fin N → (recB h.toLawsCore).Cusp j w) (P₁ : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F), P₁ ∈ harmA T.Q.c (wtA w) w.n ∧ IsBlockProd P₁ ∧ ContainsTh (T.Q.Xo j) T.Q.c (T.nab j) 1 P₁ (fun k => ((w0 k).1 : T.R j)) ∧ (fun (B : WeightFamilies.Prov D) (X : PJ D B T.G.P j) (w1 : Fin N → B.Cusp j w) => Submodule.span (𝒪 B.KO) (Set.range w1) = X.thAll w) (recB h.toLawsCore) (recPJ h j hj) w0 ∧ N ≤ 2 * ((D.host.g 0 * D.host.g 1) ^ (D.R.f * D.R.r * w.n) * (Module.finrank ℚ (recB h.toLawsCore).KO * (1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsShortThetaFamilyWithBlockPolynomialRulesStatement

theorem **_root_.GlobalTheory.T.exists_short_thetaFamily_with_block_polynomial_rules** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → D.R.f * D.R.r * (D.R.r + 1) ≤ 2 * w.n → ∃ (N : ℕ) (w0 : Fin N → (recB h.toLawsCore).Cusp j w) (P₁ : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F), P₁ ∈ harmA T.Q.c (wtA w) w.n ∧ IsBlockProd P₁ ∧ ContainsTh (T.Q.Xo j) T.Q.c (T.nab j) 1 P₁ (fun k => ((w0 k).1 : T.R j)) ∧ (fun (B : WeightFamilies.Prov D) (X : PJ D B T.G.P j) (w1 : Fin N → B.Cusp j w) => Submodule.span (𝒪 B.KO) (Set.range w1) = X.thAll w) (recB h.toLawsCore) (recPJ h j hj) w0 ∧ N ≤ 2 * ((D.host.g 0 * D.host.g 1) ^ (D.R.f * D.R.r

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSmallCoeffEtaValAtNeZero

private theorem **frMap_id_self** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : FrMap (AlgHom.id c.F S) y y

theorem **exists_small_coeff_etaValAt_ne_zero** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (R : ℕ), Odd R → 1 < R → ∀ [Fintype (LatMod Dw R)] (n : ℕ) (x₀ : Fin n → Dw.La) (K :

Type) [Field K] [IsAlgClosed K] [Algebra c.F K] (Nc : ℕ) (y : Fin Nc → FrPt Xo c K), ∃ cf : (Fin n → LatMod Dw R) → ℕ, (∀ j, cf j ≤ Nc * Nat.card (LevGrp Dw (R ^ 4))) ∧ ∀ k v, EtaValAt Xo c R (fun j => (cf j : ℤ)) (shiftFam R x₀) (y k) v → v ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSmulMemLattice

theorem **exists_smul_mem_of_span** {E : Type} [Field E] [NumberField E] {r : ℕ} (L : Submodule (RingOfIntegers E) (Fin r → E)) (v : Fin r → E) (hv : v ∈ Submodule.span E (L : Set (Fin r → E))) : ∃ c : RingOfIntegers E, c ≠ 0 ∧ (c : E) • v ∈ L

theorem **exists_smul_mem_lattice** : ∀ {E : Type} [inst : Field E] [inst_1 : NumberField E] [inst_2 : NumberField.IsCMField E] {r : ℕ} (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) (v : Fin r → E), ∃ (c : NumberField.RingOfIntegers E), c ≠ (0 : NumberField.RingOfIntegers E) ∧ (↑c : E) • v ∈ ReciprocityTheory.ChartRestriction.G.FD.Data.Λ X

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSymmetrySeriesOfFrames

private theorem **jetMap_mk_lemma** (m : ℕ) {κ κ' : Type} (f : κ → MvPolynomial κ' S) (hf : ∀ i, f i ∈ orig S κ') (q : MvPolynomial κ S) : jetMap (m

theorem **subst_comp_copy** (m : ℕ) {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (i : Fin 2) : (jetSubstP m (fun p : Fin 2 × Fin d => φ p.1 p.2)).comp (jetCopy d m i) = jetSubstP m (fun p : Fin 1 × Fin d => φ i p.2)

theorem **subst_X_eq_copy** (m : ℕ) {n : ℕ} (k : Fin n) : jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (k, p.2) : MvPolynomial (Fin n × Fin d) S)) = jetCopy d m k

theorem **pt_sub** (m : ℕ) {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (i : Fin 2) : homAt x0 (jetSubstP m (fun p : Fin 2 × Fin d => φ p.1 p.2)) ≫ jptG t x0 d x m i = homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => φ i p.2)) ≫ x m

theorem **pt_X** (m : ℕ) {n : ℕ} (k : Fin n) : homAt x0 (jetSubstP m (fun p : Fin 1 × Fin d => (MvPolynomial.X (k, p.2) : MvPolynomial (Fin n × Fin d) S))) ≫ x m = jptG t x0 d x m k

private theorem **comp_foldr_padd_eq_foldr_padd_comp** {T T' : Over X0} (f : T' ⟶ T) (xs : Fin 2 → (T ⟶ t.G)) (L : List (Fin 2)) : f ≫ L.foldr (fun k acc => padd t.G (xs k) acc) (pzero t.G T) = L.foldr (fun k acc => padd t.G (f ≫ xs k) acc) (pzero t.G T')

theorem **pe_pt** (m : ℕ) {n : ℕ} (σ : Jet S (Fin 2 × Fin d) m →ₐ[S] Jet S (Fin n × Fin d) m) (I : ULift.{0} (Finset (Fin 2))) : (homAt x0 σ).left ≫ cφG t ![jptG t x0 d x m (0 : Fin 2), jptG t x0 d x m (1 : Fin 2)] I = (psum t.G ![homAt x0 σ ≫ jptG t x0 d x m (0 : Fin 2), homAt x0 σ ≫ jptG t x0 d x m (1 : Fin 2)] I.down).left

theorem **ofMul_eq** {M : Type} [CommGroup M] {a b : M} (h : Additive.ofMul a = Additive.ofMul b) : a = b

private theorem **univ_ulift_finset_fin_two** : (Finset.univ : Finset (ULift (Finset (Fin 2)))) = {⟨∅⟩, ⟨{0}⟩, ⟨{1}⟩, ⟨{0, 1}⟩}

theorem **prod_cube2_swap** {T : Over X0} {M : Type} [CommGroup M] (h : (T ⟶ t.G) → M) (xs : Fin 2 → (T ⟶ t.G)) : ∏ I : ULift (Finset (Fin 2)), h (psum t.G xs I.down) ^ cε 2 I = h 1 * (h (xs 0 * 1))⁻¹ * (h (xs 1 * 1))⁻¹ * h (xs 1 * (xs 0 * 1))

theorem **exists_symmetry_series_of_frames** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ sy : PS S 2 d, ∀ m, ∃ u : (Jet S (Fin 2 × Fin d) m)ˣ, (u : Jet S (Fin 2 × Fin d) m) = psJet m sy ∧ LB.Gen.Corr t.PP.L _ _ _ _ (e m) (scaleGenG t _ _ (unitΓ u⁻¹) (peG t x0 d x e m ![fun j => X ((1 : Fin 2), j), fun j => X ((0 : Fin 2), j)]))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSymmetrySeriesOfFramesStatement

theorem **_root_.GlobalTheory.T.exists_symmetry_series_of_frames_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∃ sy : PS S 2 d, ∀ m, ∃ u : (Jet S (Fin 2 × Fin d) m)ˣ, (u : Jet S (Fin 2 × Fin d) m) = psJet m sy ∧ LB.Gen.Corr t.PP.L _ _ _ _ (e m) (scaleGenG t _ _ (unitΓ u⁻¹) (peG t x0 d x e m ![fun j => X ((1 : Fin 2), j), fun j => X ((0 : Fin 2), j)]))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsSymplecticCoordinatesAdapted

theorem **_root_.GlobalTheory.exists_symplectic_coordinates_adapted** {Λ : Type*} [AddCommGroup Λ] [Module.Free ℤ Λ] [Module.Finite ℤ Λ] {ω : LinearMap.BilinForm ℤ Λ} (hω : ω.IsAlt) (hp : Bijective ω) {I : Submodule ℤ Λ} (hsat : ∀ (k : ℤ) (x : Λ), k ≠ 0 → k • x ∈ I → x ∈ I) (hiso : ∀ x ∈ I, ∀ y ∈ I, ω x y = 0) {ε : ℤ} (hε : ε = 1 ∨ ε = -1) : ∃ (g : ℕ) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] Λ) (S : Finset (Fin g)), (∀ a b, ω (bas a) (bas b) = ε * _root_.ReciprocityTheory.ChartRestriction.G.Geo.Tor.omegaStd a b) ∧ I = Submodule.span ℤ ((fun k => bas (Pi.single (Sum.inl k) 1)) '' (S : Set (Fin g))) ∧ 2 * g = finrank ℤ Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaConstantOfThetaGroup

private theorem **star_hermForm_of_isHermitian** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (herm : Hm.IsHermitian) (z w : Fin g → ℂ) : star (hermForm Hm z w) = hermForm Hm w z

private theorem **norm_exp_neg_hermForm_mul_exp_hermForm_eq_one** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (herm : Hm.IsHermitian) (r : ℝ) (x a : Fin g → ℂ) : ‖Complex.exp (-((r : ℂ) * hermForm Hm x a)) * Complex.exp ((r : ℂ) * hermForm Hm a x)‖ = 1

theorem **thConst_liouville** {g : ℕ} (lat : AddSubgroup (Fin g → ℂ)) (span : Submodule.span ℝ (lat : Set (Fin g → ℂ)) = ⊤) (Hm : Matrix (Fin g) (Fin g) ℂ) (herm : Hm.IsHermitian) (k : ℤ) (x : Fin g → ℂ) (v : (Fin g → ℂ) → ℂ) (hv : Differentiable ℂ v) (hper : ∀ z, ∀ a ∈ lat, v (z + a) = v z * Complex.exp (-((k : ℂ) * Real.pi * hermForm Hm x a))) (z : Fin g → ℂ) : v z = v 0 * Complex.exp (-((k : ℂ) * Real.pi * hermForm Hm z x))

private theorem **fac_zpow_mul_fac_add_zpow_neg_eq_exp** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) (k : ℤ) (x : Fin g → ℂ) {a : Fin g → ℂ} (ha : a ∈ Φ.lat) (z : Fin g → ℂ) : Φ.fac a z ^ k * Φ.fac a (x + z) ^ (-k) = Complex.exp (-((k : ℂ) * Real.pi * hermForm Φ.Hm x a))

theorem **exists_thConst** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ), U.pt = Φ.e x → ∃ c : ℂ, c ≠ 0 ∧ Φ.ThConst k U x c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaConstantOfThetaGroupStatement

theorem **_root_.GlobalTheory.T.exists_thetaConstant_of_thetaGroup** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ), U.pt = Φ.e x → ∃ c

: ℂ, c ≠ 0 ∧ Φ.ThConst k U x c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaGeneratorReadingNeZeroAllFrames

theorem **exists_theta_ne_zero_record** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), D.R.r ≤ w.n → 3 * WeightAlgebra.DF w + 2 ≤ T.Q.l.M → ∀ (σ : T.Q.c.F →+* ℂ), GoodEmb T.Q.c σ → ∀ k, ∀ x ∈ ((T.U j σ).X k).dom, ∃ f ∈ T.th j w, rdU (T.U j σ) f k x ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaGeneratorReadingNeZeroAllFramesStatement

theorem **_root_.GlobalTheory.T.exists_thetaGenerator_reading_ne_zero_all_frames** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), D.R.r ≤ w.n → 3 * WeightAlgebra.DF w + 2 ≤ T.Q.l.M → ∀ (σ : T.Q.c.F →+* ℂ), GoodEmb T.Q.c σ → ∀ k, ∀ x ∈ ((T.U j σ).X k).dom, ∃ f ∈ T.th j w, rdU (T.U j σ) f k x ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaGeneratorReadingNeZeroAtFrame

theorem **mem_mul_smul_top** {R M : Type} [CommRing R] [AddCommGroup M] [Module R M] {I J : Ideal R} (h : I ⊔ J = ⊤) {x : M} (hI : x ∈ I • (⊤ : Submodule R M)) (hJ : x ∈ J • (⊤ : Submodule R M)) : x ∈ (I * J) • (⊤ : Submodule R M)

theorem **dmin_of_isOrtho** {X : Data E r} {x : Frame X.Ψ r} (ho : X.IsOrtho x) (τ : X.Ψ) (i : Fin (r + 1)) : dmin X.Gm x τ i = ((psgn X.p0 i : ℝ) : ℂ)

theorem **psgn_mul_self** (p i : ℕ) : psgn p i * psgn p i = 1

theorem **isUnitFr_of_isOrtho** {X : Data E r} {x : Frame X.Ψ r} (ho : X.IsOrtho x) : IsUnitFr X x

theorem **rmul_mem_dom_of_isOrtho** {X : Data E r} {x b : Frame X.Ψ r} (hx : x ∈ X.dom) (hb : b ∈ Bor X.Ψ r) (ho : X.IsOrtho (rmul x b)) : rmul x b ∈ X.dom

theorem **mark_sub_mem** {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {L : Lt E r} {mk : Dw.La →ₗ[𝒪 E] (Fin r → E)} (hm : IsMark l L mk) {a a' : Dw.La} (ha : a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La)) (ha' : a' ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La)) (h : a' - a ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) : ((l.M : E)⁻¹) • mk a' - ((l.M : E)⁻¹) • mk a ∈ L

theorem **exists_thGen_rdU_ne_zero** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → 1 ≤ r → ∀ t : ℝ, 0 < t → (∀ k τ, (U.X k).c τ = t) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → r ≤ n → 3 * Formalism.L2.Pj.DF Dw a + 2 ≤ l.M → ∀ (_ : StarRing (𝒪 E)) (nab : NabV Xo c), IsNab Xo c nab → ∀ k, ∀ x ∈ (U.X k).dom, ∃ f ∈ thGen Xo c nab a n, Formalism.L2.PjFree.rdU U f k x ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaGeneratorReadingNeZeroAtFrameStatement

theorem **_root_.GlobalTheory.T.exists_thetaGenerator_reading_ne_zero_at_frame** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → 1 ≤ r → ∀ t : ℝ, 0 < t → (∀ k τ, (U.X k).c τ = t) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → r ≤ n → 3 * Formalism.L2.Pj.DF Dw a + 2 ≤ l.M → ∀ (_ : StarRing (𝒪 E)) (nab : NabV Xo c), IsNab Xo c nab → ∀ k, ∀ x ∈ (U.X k).dom, ∃ f ∈ thGen Xo c nab a n, Formalism.L2.PjFree.rdU U f k x ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaGrpPullback

private theorem **gm_res_eq_gm** {X Z : Scheme.{0}} (r : Z ⟶ X) (z : Z) {U V : X.Opens} (hle : V ≤ U) (hz : r z ∈ V) (a : Γ(X, U)ˣ) : gm r z V (res hle a) = gm r z U a

private theorem **exists_isGen_linComb_val_eq_of_eq** {X Y : Scheme.{0}} {L : LB X} {K : Type} [Fintype K] {φ φ' : K → (Y ⟶ X)} {ε : K → ℤ} (hφ : φ = φ') (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) : ∃ t' : LB.Gen (LB.LinComb L φ' ε), t'.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (J : (Z ⟶ X) → L.ι), val t' r z J = val t r z J

theorem **whiskerRight_translPt** {T T' : Over S} (h : T' ⟶ T) (x : T ⟶ A) : h ▷ A ≫ translPt A x = translPt A (h ≫ x)

theorem **translT_whiskerRight** {T T' : Over S} (h : T' ⟶ T) (x : T ⟶ A) : translT A (h ≫ x) ≫ h ▷ A = h ▷ A ≫ translT A x

theorem **negT_whiskerRight** {T T' : Over S} (h : T' ⟶ T) : negT A T' ≫ h ▷ A = h ▷ A ≫ negT A T

theorem **val_powGen** {L : LB A.left} {k : ℤ} {T : Over S} (θ : ThetaGrp A L k T) (n : ℕ) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) {z : Z} {J} (hJ : Good L z J) : val (powGen A L k θ n) r z J = ∏ j : Fin n, val (L

theorem **val_symGen** {L : LB A.left} {k : ℤ} {T : Over S} (θ : ThetaGrp A L k T) (ν : LB.Gen (negLB A L k T)) {Z : Scheme.{0}} (r : Z ⟶ (T ⊗ A).left) {z : Z} {J} (hJ : Good L z J) : val (symGen A L k θ ν) r z J = val (L

theorem **exists_thetaGrp_pullback** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : LB A.left), L.IsLB → ∀ (k : ℤ) (T T' : Over S) (h : T' ⟶ T) (θ : ThetaGrp A L k T), ∃ θ' : ThetaGrp A L k T', θ'.pt = h ≫ θ.pt ∧ LB.Gen.Corr L _ _ _ _ θ'.φ (LB.pullGen L _ _ θ.φ (h ▷ A).left) ∧ (∀ (n : ℕ) (x : T ⟶ A), SymLift A L k n x θ → SymLift A L k n (h ≫ x) θ') ∧ ∀ (s t : LB.Sec (LPow A L k T)) (s' t' : LB.Sec (LPow A L k T')), IsActOf A L k θ s t → IsPullSec A L k (h ▷ A) (whiskerRight_snd _ _) s s' → IsPullSec A L k (h ▷ A) (whiskerRight_snd _ _) t t' → IsActOf A L k θ' s' t'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaGrpPullbackStatement

theorem **_root_.GlobalTheory.T.exists_symLift_pullback_isActOf** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : LB A.left), L.IsLB → ∀ (k : ℤ) (T T' : Over S) (h : T' ⟶ T) (θ : ThetaGrp A L k T), ∃ θ' : ThetaGrp A L k T', θ'.pt = h ≫ θ.pt ∧ LB.Gen.Corr L _ _ _ _ θ'.φ (LB.pullGen L _ _ θ.φ (h ▷ A).left) ∧ (∀ (n : ℕ) (x : T ⟶ A), SymLift A L k n x θ → SymLift A L k n (h ≫ x) θ') ∧ ∀ (s t : LB.Sec (LPow A L k T)) (s' t' : LB.Sec (LPow A L k T')), IsActOf A L k θ s t → IsPullSec A L k (h ▷ A) (whiskerRight_snd _ _) s s' → IsPullSec A L k (h ▷ A) (whiskerRight_snd _ _) t t' → IsActOf A L k θ' s' t'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaRunParametersUniformTorus

theorem **exists_large_nat** (B c_1 c_2 d C : ℝ) (hB : 0 < B) (hd : 0 ≤ d) : ∃ q : ℕ, 1 ≤ q ∧ c_1 < q ∧ c_2 ≤ (q : ℝ) / 2 ∧ 1 ≤ (q : ℝ) / B ∧ 4 * (7 + Real.log q) + d + 1 ≤ q * ((q : ℝ) / B) ∧ C ≤ q * ((q : ℝ) / B)

theorem **exists_par** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r m : ℕ), 2 ≤ PT.f E * r → 1 ≤ m → m ^ 4 ≤ PT.f E → (PT.f E : ℝ) ≤ PT.rd E → (PT.f E : ℝ) ^ 2 ≤ PT.rd E → 16 ≤ PT.rd E → ∃ (q : ℕ) (R0 : ℝ), 0 < th (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaRunParametersUniformTorusStatement

theorem **_root_.GlobalTheory.T.exists_thetaRunParameters_uniform_torus** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r m : ℕ), 2 ≤ PT.f E * r → 1 ≤ m → m ^ 4 ≤ PT.f E → (PT.f E : ℝ) ≤ PT.rd E → (PT.f E : ℝ) ^ 2 ≤ PT.rd E → 16 ≤ PT.rd E → ∃ (q : ℕ) (R0 : ℝ), 0 < th (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaSublatticeJetNeZero

theorem **exists_poly_scale** {σ W : Type} [NormedAddCommGroup W] [NormedSpace ℂ W] (P : MvPolynomial σ ℂ) {s : ℂ} (hs : s ≠ 0) : ∃ P' : MvPolynomial σ ℂ, P'.support = P.support ∧ ∀ (u : σ → W) (F : W → ℂ) (x : W), polyDeriv P (fun l => s • u l) F x = polyDeriv P' u F x

theorem **exists_jet_of_periods** {ι A : Type} [Fintype ι] [AddCommGroup A] (p : A →+ EuclideanSpace ℂ ι) (hp : Function.Injective p) (b : Module.Basis (Fin (2 * Fintype.card ι)) ℝ (EuclideanSpace ℂ ι)) (hb : ∀ y, y ∈ lat b ↔ ∃ a, p a = y) (hint : ∀ a a', ∃ k : ℤ, (Hf (p a) (p a')).im = k) (hperf : ∀ φ : A →+ ℤ, ∃ a, ∀ a', (φ a' : ℝ) = (Hf (p a) (p a')).im) (Q : A → ℝ) (hQ : ∀ a a', ∃ k : ℤ, Q (a + a') = Q a + Q a' − (Hf (p a) (p a')).im + 2 * k) (q : ℕ) (P : MvPolynomial ι ℂ) (hP : P ≠ 0) (hq : 3 * P.totalDegree + 2 ≤ q) {s : ℂ} (hs : s ≠ 0) : ∃ G : EuclideanSpace ℂ ι → ℂ, Differentiable ℂ G ∧ (∀ a v, G (v + p a) = cexp (Real.pi * I * (q * Q a : ℝ)) * cexp (q * Real.pi * Hf v (p a) + q * Real.pi * Hf (p a) (p a) / 2) * G v) ∧ polyDeriv P (fun i => s • EuclideanSpace.single i 1) G 0 ≠ 0

theorem **colEquiv_apply** {K : Type} [Fintype K] {n g : ℕ} (e : K ≃ Fin n × Fin g) (s : ℂ) (hs : s ≠ 0) (z : Fin n → Fin g → ℂ) (i : K) : colEquiv e s hs z i = s * z (e i).1 (e i).2

theorem **exists_basis_cols** {K κ : Type} [Fintype K] [Fintype κ] {n g : ℕ} (Φ : (Fin n → Fin g → ℂ) ≃ₗ[ℂ] EuclideanSpace ℂ K) (b₀ : Module.Basis κ ℝ (Fin g → ℂ)) : ∃ b : Module.Basis (Fin (2 * Fintype.card K)) ℝ (EuclideanSpace ℂ K), ∀ z, Φ z ∈ lat b ↔ ∀ k, z k ∈ Submodule.span ℤ (Set.range b₀)

theorem **EF_add_left** (u u' v : Fin r → E) : EF X (u + u') v = EF X u v + EF X u' v

theorem **EF_add_right** (u v v' : Fin r → E) : EF X u (v + v') = EF X u v + EF X u v'

theorem **EF_ratSmul_left** (c : ℚ) (u v : Fin r → E) : EF X ((c : E) • u) v = c * EF X u v

theorem **EF_mem_mul** (hL : SelfD X L) (hM : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) {u v : Fin r → E} (hu : u ∈ 𝔐 • L) (hv : v ∈ 𝔐 • L) : ∃ k : ℤ, EF X u v = M * k

theorem **mem_of_EF_mem_mul** (hL : SelfD X L) (hM0 : M ≠ 0) (hM : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) {u : Fin r → E} (hu : ∀ v ∈ 𝔐 • L, ∃ k : ℤ, EF X u v = M * k) : u ∈ 𝔐 • L

theorem **exists_EF_eq** {bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} (hs : IsSympl X bas) (hL : SelfD X L) (hM0 : M ≠ 0) (hM : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) (φ : (𝔐 • L : Lt E r) →+ ℤ) : ∃ u ∈ 𝔐 • L, ∀ v : (𝔐 • L : Lt E r), EF X u v = M * φ v

theorem **EF_quad** {w : E} (hw : w + cc w = 1) (u v : Fin r → E) : EF X (u + v) (w • (u + v)) = EF X u (w • u) + EF X v (w • v) − EF X u v + 2 * EF X u (w • v)

theorem **exists_basis_smul** {κ : Type} [Fintype κ] (pl : (Fin r → E) →+ (Fin g → ℂ)) (b₀ : Module.Basis κ ℝ (Fin g → ℂ)) (hb₀ : ∀ z, z ∈ Submodule.span ℤ (Set.range b₀) ↔ ∃ u ∈ L, z = pl u) (hM0 : M ≠ 0) (hM : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) : ∃ b₁ : Module.Basis κ ℝ (Fin g → ℂ), ∀ z, z ∈ Submodule.span ℤ (Set.range b₁) ↔ ∃ u ∈ 𝔐 • L, z = pl u

theorem **GlobalTheory.Dict.exists_theta_sublattice_jet_ne_zero** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (𝔐 : Ideal (𝒪 E)) (M : ℕ), Odd M → 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)} → ∀ P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n), P ≠ 0 → 3 * P.totalDegree + 2 ≤ M → ∃ F : (Fin n → Fin g → ℂ) → ℂ, Differentiable ℂ F ∧ (∀ u : Matrix (Fin n) (Fin r) E, (∀ b, u b ∈ 𝔐 • L) → ∀ z, F z = epsW X (w : E) u * cexp (-(Real.pi * ∑ b, hx X bas idx x (z b + (1 / 2 : ℂ) • pv X idx x (u b)) (pv X idx x (u b)))) * F (fun b => z b + pv X idx x (u b))) ∧ I2.polyDeriv P (letterDir X hr idx) F 0 ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaSublatticeJetNeZeroStatement

theorem **_root_.GlobalTheory.T.exists_theta_sublattice_jet_ne_zero_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (𝔐 : Ideal (𝒪 E)) (M : ℕ), Odd M → 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)} → ∀ P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n), P ≠ 0 → 3 * P.totalDegree + 2 ≤ M → ∃ F : (Fin n → Fin g → ℂ) → ℂ, Differentiable ℂ F ∧ (∀ u : Matrix (Fin n) (Fin r) E, (∀ b, u b ∈ 𝔐 • L) → ∀ z, F z = epsW X (w : E) u * cexp (-(Real.pi * ∑ b, hx X bas idx x (z b + (1 / 2 : ℂ) • pv X idx x (u b)) (pv X idx x (u b)))) * F (fun b => z b + pv X idx x (u b))) ∧ I2.polyDeriv P (letterDir X hr idx) F 0 ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThetaWeightData

theorem **one_le_size** {m : ℤ} (hm : 0 ≤ m) {G : (ι → ℤ) × ℤ} (hG : 0 ≤ G.2) : 1 ≤ size m G

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsThickPointHyperbolic

theorem **_root_.GlobalTheory.ThickPoint.exists_euclidean_real** {M : ℕ} [FiniteDimensional ℝ W] (hM : finrank ℝ W = M) (B : LinearMap.BilinForm ℝ W) (hs : B.IsSymm) (hp : ∀ x, 0 ≤ B x x) : ∃ φ : W →ₗ[ℝ] EuclideanSpace ℝ (Fin M), ∀ x y, inner ℝ (φ x) (φ y) = B x y

theorem **_root_.GlobalTheory.ThickPoint.transference_pair** {M : ℕ} (b : Basis (Fin M) ℚ V) (B : LinearMap.BilinForm ℚ V) (hs : B.IsSymm) (hn : B.Nondegenerate) (φ ψ : V →+ EuclideanSpace ℝ (Fin M)) (hφ : ∀ x y, inner ℝ (φ x) (ψ y) = (B x y : ℝ)) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ u, u ≠ 0 → (∀ j, ∃ k : ℤ, B u (b j) = k) → lam ^ 2 ≤ ‖ψ u‖ ^ 2) : ∃ v : Fin M → V, (∀ i, v i ∈ span ℤ (Set.range b)) ∧ LinearIndependent ℚ v ∧ ∀ i, ‖φ (v i)‖ ^ 2 ≤ ((M : ℝ) ^ 2 / lam) ^ 2

theorem **traceFormOf_apply** (G : Matrix (Fin r) (Fin r) E) (x y : Fin r → E) : traceFormOf G x y = Algebra.trace ℚ E (herm G x y)

private lemma **herm_swap_eq_cc_herm** {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (G a b) = G b a) (x y : Fin r → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm G y x = _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm G x y)

theorem **_root_.GlobalTheory.ThickPoint.traceFormOf_isSymm** {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (G a b) = G b a) : (_root_.GlobalTheory.ThickPoint.traceFormOf G).IsSymm

theorem **_root_.GlobalTheory.ThickPoint.traceFormOf_nondegenerate** {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (G a b) = G b a) (hdet : G.det ≠ 0) : (_root_.GlobalTheory.ThickPoint.traceFormOf G).Nondegenerate

theorem **discr_smul_mem** (X : DataX E r) (u : Fin r → E) (htr : ∀ y ∈ X.L', ∃ k : ℤ, Algebra.trace ℚ E (herm X.Gm u y) = k) : ((discr E : ℤ) : E) • u ∈ X.L'

theorem **majForm_apply** (X : Data E r) (x : Frame X.Ψ r) (hx : x ∈ X.dom) (u v : X.Ψ → Fin r → ℂ) : majForm X x hx u v = MajorantOperators.gI x u v

theorem **_root_.GlobalTheory.ThickPoint.pair_eq** (X : _root_.ReciprocityTheory.ChartRestriction.G.FD.Data E r) (x : _root_.ReciprocityTheory.ChartRestriction.G.FD.Frame X.Ψ r) (hx : x ∈ X.dom) (u v : Fin r → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.MajorantOperators.gI x (_root_.GlobalTheory.ThickPoint.embHom X u) (_root_.GlobalTheory.ThickPoint.reflHom X x v) = (_root_.GlobalTheory.ThickPoint.traceFormOf X.Gm u v : ℝ)

private lemma **isIntegral_natAbs_discr_mul_of_mem_inv_differentIdeal** {z : E} (hz : z ∈ (↑(differentIdeal ℤ (𝒪 E)) : FractionalIdeal (𝒪 E)⁰ E)⁻¹) : IsIntegral ℤ (((discr E).natAbs : E) * z)

theorem **_root_.GlobalTheory.exists_short_basis_at** {r : ℕ} (X : _root_.ReciprocityTheory.ChartRestriction.G.FD.DataX E r) (x : _root_.ReciprocityTheory.ChartRestriction.G.FD.Frame X.Ψ r) (hx : x ∈ X.dom) (ε : ℝ) (hε : 0 < ε) (hmin : ∀ v ∈ X.L', v ≠ 0 → ε ≤ ∑ τ, X.maj x τ (_root_.ReciprocityTheory.ChartRestriction.G.FD.Data.emb τ.1 v)) : ∃ v : Fin r → Fin r → E, (∀ i, v i ∈ X.L') ∧ LinearIndependent E v ∧ ∀ i, ∑ τ, X.maj x τ (_root_.ReciprocityTheory.ChartRestriction.G.FD.Data.emb τ.1 (v i)) ≤ (2 * _root_.ReciprocityTheory.ChartRestriction.G.FD.fdeg E * r : ℝ) ^ 4 * _root_.ReciprocityTheory.ChartRestriction.G.FD.dE E ^ 2 / (4 * ε)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTorusOfRows

private theorem **mem_lat_basis_map_iff** {J K : Type} [Fintype J] (b : Module.Basis K ℝ (EuclideanSpace ℂ J)) (f : EuclideanSpace ℂ J ≃ₗ[ℝ] EuclideanSpace ℂ J) (x : EuclideanSpace ℂ J) : x ∈ I2.lat (b.map f) ↔ f.symm x ∈ I2.lat b

theorem **exists_torusOf_rows** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pm : D.La →+ (Fin g → ℂ)) (t : ℝ) (row : Fin g → ℕ), 0 < t → IsTypeFn H D.p D.q ty → IsPeriod D ty Pm → (∀ a b, (hermForm (Matrix.diagonal fun _ => (t : ℂ)) (Pm a) (Pm b)).im = D.EF a b) → (∀ j, row j < D.mult (ty j)) → (∀ j j', ty j = ty j' → row j = row j' → j = j') → ∃ (T : I2.PT E) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim), IsTorusOf D ty Pm (Matrix.diagonal fun _ => (t : ℂ)) T ψ ∧ ∀ j, ∃ h : row j < T.dim (ty j), ψ (Pi.single j 1) = open Classical in (Real.sqrt t : ℂ) • EuclideanSpace.single ((ty j, ⟨row j, h⟩) : I2.Idx T.dim) (1 : ℂ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTorusOfSelfD

theorem **nondeg** {bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} (hs : IsSympl X bas) : LinearMap.BilinForm.Nondegenerate (EFBil X)

theorem **perf** {bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} (hs : IsSympl X bas) {L : Lt E r} (hL : SelfD X L) (φ : L →+ ℤ) : ∃ u ∈ L, ∀ v : L, (φ v : ℚ) = EF X u v

private theorem **exists_int_ne_zero_smul_mem_of_span_eq_univ** {L L' : Lt E r} (hfg : L.FG) (hsp : Submodule.span E (L' : Set (Fin r → E)) = ⊤) : ∃ n : ℤ, n ≠ 0 ∧ ∀ u ∈ L, (n : E) • u ∈ L'

theorem **exists_basis_of_full** {x : Frame X.Ψ r} (ho : X.IsOrtho x) {L : Lt E r} (hfg : L.FG) (hsp : Submodule.span E (L : Set (Fin r → E)) = ⊤) : ∃ b : Module.Basis (Fin (2 * Fintype.card (I2.Idx (dimPer X)))) ℝ (I2.Vsp (dimPer X)), ∀ y, y ∈ I2.lat b ↔ ∃ v ∈ L, scaled X x v = y

theorem **scaleEquiv_apply** {K : Type} [Fintype K] (e : Fin g ≃ K) (s : ℂ) (hs : s ≠ 0) (z : Fin g → ℂ) (i : K) : scaleEquiv e s hs z i = s * z (e.symm i)

theorem **GlobalTheory.Dict.exists_torus_of_selfD** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g : ℕ) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r), IsSympl X bas → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∃ (T : I2.PT E) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim) (e : Fin g ≃ I2.Idx T.dim), T.r = r ∧ (∀ z z', I2.Hf (ψ z) (ψ z') = hx X bas idx x z z') ∧ (∀ ga, ga ∈ T.L ↔ ∃ u ∈ L, ga = ψ (pv X idx x u)) ∧ (∀ (a : 𝒪 E) u, ψ (pv X idx x ((a : E) • u)) = I2.act a (ψ (pv X idx x u))) ∧ (∀ ga ∈ T.L, (T.eps ga : ℝ) = Real.cos (Real.pi * (I2.Hf ga (I2.act w ga)).im)) ∧ (∀ j, (e j).1 = rowType X (idx j) ∧ ψ (Pi.single j 1) = ((Real.sqrt t : ℝ) : ℂ) • EuclideanSpace.single (e j) (1 : ℂ))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTorusWithSignUniformisationFrame

theorem **cmHostUnit_add_conj_eq_one** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : H.w0 + H.bar H.w0 = 1

theorem **exists_torusAt_of_isom** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (D : Datum H) {r : ℕ} (X : FD.Data E r) (φ : D.La ≃ₗ[𝒪 E] X.Λ) (isom : ∀ a b, ((D.EF a b : ℤ) : ℚ) = Algebra.trace ℚ E (-X.ξ0 * FD.herm X.Gm (φ a : Fin r → E) (φ b))) (x : FD.Frame X.Ψ r) (hx : x ∈ X.dom) : ∃ A : I2.TorusAt X x, A.w0 = H.w0

theorem **exists_torusAt_w0_eq** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (W : UnfAt U k x), ∃ A : I2.TorusAt (U.X k) x.1, A.w0 = H.w0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTorusWithSignUniformisationFrameStatement

theorem **_root_.GlobalTheory.T.exists_torusWithSign_uniformisationFrame** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (W : UnfAt U k x), ∃ A : I2.TorusAt (U.X k) x.1, A.w0 = H.w0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTranslateJetNeZero

theorem **spans_mod_pow_succ** (m : Ideal R) (J A B : Submodule k R) (D : ℕ) (hJ : ∀ x, ∃ s ∈ J, x - s ∈ m ^ 2) (hA : ∀ x, ∃ a ∈ A, x - a ∈ m ^ (D + 1)) (hB : ∀ a ∈ A, ∀ s ∈ J, a * s ∈ B) (x : R) : ∃ b ∈ B, x - b ∈ m ^ (D + 2)

theorem **Hf_add_left** (v w γ : EuclideanSpace ℂ ι) : Hf (v + w) γ = Hf v γ + Hf w γ

theorem **twistedLaw_zero_iff** (Y : GT ι) (k : ℕ) (F : EuclideanSpace ℂ ι → ℂ) : TwistedLaw Y k 0 F ↔ F ∈ (powTorus Y k).Sec k

theorem **TwistedLaw.mul** {Y : GT ι} {k l : ℕ} {c d : EuclideanSpace ℂ ι} {F G : EuclideanSpace ℂ ι → ℂ} (hF : TwistedLaw Y k c F) (hG : TwistedLaw Y l d G) : TwistedLaw Y (k + l) (c + d) (F * G)

theorem **twistedLaw_translate** {Y : GT ι} {θ : EuclideanSpace ℂ ι → ℂ} (hθ : θ ∈ Y.Sec 1) (a : EuclideanSpace ℂ ι) : TwistedLaw Y 1 a (fun v => θ (v + a))

theorem **TwistedLaw.pow** {Y : GT ι} {c : EuclideanSpace ℂ ι} {F : EuclideanSpace ℂ ι → ℂ} (hF : TwistedLaw Y 1 c F) (p : ℕ) : TwistedLaw Y p (p • c) (F ^ p)

theorem **analyticAt_of_differentiable** {W : Type*} [NormedAddCommGroup W] [NormedSpace ℂ W] [FiniteDimensional ℂ W] {F : W → ℂ} (hF : Differentiable ℂ F) (x : W) : AnalyticAt ℂ F x

theorem **exists_sec_ne_zero** {Y : GT ι} {θ : EuclideanSpace ℂ ι → ℂ} (hθ : θ ∈ Y.Sec 1) (h0 : θ ≠ 0) {r : ℕ} (hr : 2 ≤ r) : ∃ F ∈ (powTorus Y r).Sec r, F 0 ≠ 0

theorem **mul_mem_jets** {Y : GT ι} {k l : ℕ} {x y : MvPowerSeries ι ℂ} (hx : x ∈ jets Y k) (hy : y ∈ jets Y l) : x * y ∈ jets Y (k + l)

theorem **coeff_single_taylorDir** {σ W : Type*} [Fintype σ] [NormedAddCommGroup W] [NormedSpace ℂ W] (u : σ → W) {F : W → ℂ} (hF : AnalyticAt ℂ F 0) (i : σ) : MvPowerSeries.coeff (Finsupp.single i 1) (taylorDir u F) = fderiv ℂ F 0 (u i)

theorem **sub_affine_mem_sq** {σ : Type*} [Fintype σ] (x : MvPowerSeries σ ℂ) : x − (MvPowerSeries.constantCoeff x • (1 : MvPowerSeries σ ℂ) + ∑ i, MvPowerSeries.coeff (Finsupp.single i 1) x • (MvPowerSeries.X i : MvPowerSeries σ ℂ)) ∈ germIdeal σ ℂ ^ 2

theorem **jets_three** {Y : GT ι} {θ : EuclideanSpace ℂ ι → ℂ} (hθ : θ ∈ Y.Sec 1) (h0 : θ ≠ 0) (x : MvPowerSeries ι ℂ) : ∃ s ∈ jets Y 3, x − s ∈ germIdeal ι ℂ ^ 2

theorem **jets_span** {Y : GT ι} {θ : EuclideanSpace ℂ ι → ℂ} (hθ : θ ∈ Y.Sec 1) (h0 : θ ≠ 0) {r : ℕ} (hr : 2 ≤ r) (D : ℕ) (x : MvPowerSeries ι ℂ) : ∃ s ∈ jets Y (r + 3 * D), x − s ∈ germIdeal ι ℂ ^ (D + 1)

theorem **exists_section_jet_ne_zero** {ι : Type} [Fintype ι] (Y : GT ι) (θ : EuclideanSpace ℂ ι → ℂ) (hθ : θ ∈ Y.Sec 1) (h0 : θ ≠ 0) (D q : ℕ) (hq : 3 * D + 2 ≤ q) (P : MvPolynomial ι ℂ) (hP : P ≠ 0) (hD : P.totalDegree ≤ D) : ∃ F : EuclideanSpace ℂ ι → ℂ, Differentiable ℂ F ∧ (∀ γ ∈ lat Y.b, ∀ v, F (v + γ) = Y.ka γ ^ q * cexp (q * Real.pi * Hf v γ + q * Real.pi * Hf γ γ / 2) * F v) ∧ polyDeriv P (fun i => EuclideanSpace.single i 1) F 0 ≠ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTranslationSectionsUniversalObject

private theorem **mem_wchart_lemma** {X Y : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (j : K → L.ι) (y : Y) : y ∈ L.wchart φ j ↔ ∀ k, φ k y ∈ L.U (j k)

theorem **twist_congr** (hL : L.IsLB) {T : Over S} {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} {hm : ψ ≫ snd T A = m} {a b : ℤ} {α α' : LB.Gen (LB.LinComb L (K

theorem **act_congr** (hL : L.IsLB) {T : Over S} {k : ℤ} {U U' : ThetaGrp A L k T} (hp : U.pt = U'.pt) (h : LB.Gen.Corr L _ _ _ _ U.φ U'.φ) {s s' s'' : LB.Sec (LPow A L k T)} (h' : IsActOf A L k U s s') (h'' : IsActOf A L k U' s s'') : s' = s''

theorem **exists_isVtOver_and_existsUnique_isUUOver** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), Odd N → ∀ (T : Over Xo.X) (s : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 T)), (∀ (ok : (T ⟶ Xo.univ.t.G) → Prop) (t t' : T ⟶ Xo.univ.t.G), t' ^ N = t → t' ^ (N ^ 2) = 1 → ok t' → ∃ F, IsVtOver Xo.univ.t.G Xo.univ.t.PP N ok s t F) ∧ ∀ x₁ x₂ : T ⟶ Xo.univ.t.G, x₁ ^ (N ^ 2) = 1 → x₂ ^ (N ^ 2) = 1 → ∃! F, IsUUOver Xo.univ.t.G Xo.univ.t.PP N s x₁ x₂ F

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsTranslationSectionsUniversalObjectStatement

theorem **_root_.GlobalTheory.T.exists_translation_sections_universal_object** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), Odd N → ∀ (T : Over Xo.X) (s : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 T)), (∀ (ok : (T ⟶ Xo.univ.t.G) → Prop) (t t' : T ⟶ Xo.univ.t.G), t' ^ N = t → t' ^ (N ^ 2) = 1 → ok t' → ∃ F, IsVtOver Xo.univ.t.G Xo.univ.t.PP N ok s t F) ∧ ∀ x₁ x₂ : T ⟶ Xo.univ.t.G, x₁ ^ (N ^ 2) = 1 → x₂ ^ (N ^ 2) = 1 → ∃! F, IsUUOver Xo.univ.t.G Xo.univ.t.PP N s x₁ x₂ F

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsUniqueIsTwistPull

theorem **existsUnique_isTwistPull** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] (L : LB A.left), L.IsLB → ∀ (T : Over S) (ψ : T ⊗ A ⟶ T ⊗ A) (m : T ⊗ A ⟶ A) (hm : ψ ≫ snd T A = m) (a b : ℤ) (α : LB.Gen (LB.LinComb L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsValueBasisInSpanningForms

theorem **exists_indep_of_span_image** {α : Type} {δ : ℕ} (S : Set α) (v : α → Fin δ → ℂ) (h : Submodule.span ℂ (v '' S) = ⊤) : ∃ θ : Fin δ → α, (∀ i, θ i ∈ S) ∧ LinearIndependent ℂ fun i => v (θ i)

theorem **ev_mem_sp_of_mem_span** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} {X : PJ D B P j} (R : Rdg X) (u : WeightFamilies.Wt D j) (s : Set (B.Cusp j u)) (F : B.Cusp j u) (hF : F ∈ Submodule.span B.KO s) : R.ev (X.inc u F) ∈ R.sp u s

theorem **exists_valueBasis_in_spanning** (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (N : ℕ) (w : Fin N → B.Cusp j wL) (R : Rdg X) (t : Idx D j N → ℂ) (T : ℕ) (Mg : Submodule B.KO (B.Cusp j ka)) (δ : ℕ) (xs : Fin δ → R.Y) (hA : AnFacts R w t ka wF T Mg xs) (Sh : Set (B.Cusp j wF)) (hS : Submodule.span B.KO Sh = ⊤) : ∃ θ : Fin δ → B.Cusp j wF, (∀ i, θ i ∈ Sh) ∧ LinearIndependent ℂ (vals xs fun i => R.ev (X.inc wF (θ i)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExistsValueBasisInSpanningFormsStatement

theorem **_root_.GlobalTheory.T.exists_valueBasis_in_spanning_forms** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (N : ℕ) (w : Fin N → B.Cusp j wL) (R : Rdg X) (t : Idx D j N → ℂ) (T : ℕ) (Mg : Submodule B.KO (B.Cusp j ka)) (δ : ℕ) (xs : Fin δ → R.Y), AnFacts R w t ka wF T Mg xs → ∀ Sh : Set (B.Cusp j wF), Submodule.span B.KO Sh = ⊤ → ∃ θ : Fin δ → B.Cusp j wF, (∀ i, θ i ∈ Sh) ∧ LinearIndependent ℂ (vals xs fun i => R.ev (X.inc wF (θ i)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ExpNsmulEqPowVt

private theorem **AnFr.exp_nsmul_eq_pow_vt** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) (z : Fin g → ℂ) (n : ℕ) : Φ.e (n • z) = Φ.e z ^ n

theorem **rd_vt_and_uu** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), (∀ a ∈ Φ.lat, Φ.χ a ^ 2 = 1) → ∀ (N : ℕ), Odd N → ∀ (s : LB.Sec (LPow A L.L 1 (Over.mk x0))) (θ : (Fin g → ℂ) → ℂ), Φ.Rd 1 s θ → (∀ (ok : (Over.mk x0 ⟶ A) → Prop) (t : Over.mk x0 ⟶ A) (F : LB.Sec (LPow A L.L ((N : ℤ) ^ 2) (Over.mk x0))), IsVtOver A L N ok s t F → (∀ t', ok t' → t' ^ N = t → t' ^ (N ^ 2) = 1) → ∃ x δ : Fin g → ℂ, δ ∈ Φ.lat ∧ ((N : ℂ) ^ 2) • x = (2 : ℂ) • δ ∧ ok (Φ.e x) ∧ Φ.e x ^ N = t ∧ Φ.Rd ((N : ℤ) ^ 2) F (fock Φ.Hm ((N : ℤ) ^ 2) x fun z => θ ((N : ℂ) • z))) ∧ ∀ (x₁ δ₁ x₂ δ₂ : Fin g → ℂ) (F : LB.Sec (LPow A L.L ((N : ℤ) ^ 2) (Over.mk x0))), δ₁ ∈ Φ.lat → ((N : ℂ) ^ 2) • x₁ = (2 : ℂ) • δ₁ → δ₂ ∈ Φ.lat → ((N : ℂ) ^ 2) • x₂ = (2 : ℂ) • δ₂ → IsUUOver A L N s (Φ.e x₁) (Φ.e x₂) F → Φ.Rd ((N : ℤ) ^ 2) F (fock Φ.Hm ((N : ℤ) ^ 2) x₂ (fock Φ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FaithfullyFlatPointsBaseChangeDescent

theorem **faithfullyFlat_of_geometric_points** (S B : Type) [CommRing S] [CommRing B] (φ : S →+* B) (hφ : φ.Flat) (h : ∀ (K : Type) [Field K] [IsAlgClosed K] (ψ : S →+* K), ∃ χ : B →+* K, χ.comp φ = ψ) : φ.FaithfullyFlat

theorem **flat_comp_localizationAway** (S B : Type) [CommRing S] [CommRing B] (φ : S →+* B) (e : B) (hφ : φ.Flat) : ((algebraMap B (Localization.Away e)).comp φ).Flat

theorem **_root_.GlobalTheory.exists_faithfullyFlat_pushout** (F S S₁ S₂ : Type) [Field F] [CommRing S] [CommRing S₁] [CommRing S₂] [Algebra F S] [Algebra F S₁] [Algebra F S₂] (φ₁ : S →ₐ[F] S₁) (φ₂ : S →ₐ[F] S₂) (h : φ₂.toRingHom.FaithfullyFlat) : ∃ (T : Type) (_ : CommRing T) (_ : Algebra F T) (ψ₁ : S₁ →ₐ[F] T) (ψ₂ : S₂ →ₐ[F] T), ψ₁.comp φ₁ = ψ₂.comp φ₂ ∧ ψ₁.toRingHom.FaithfullyFlat ∧ Function.Injective ψ₁

theorem **_root_.GlobalTheory.existsUnique_preimage_of_faithfullyFlat** (F S S' : Type) [Field F] [CommRing S] [CommRing S'] [Algebra F S] [Algebra F S'] (φ : S →ₐ[F] S') (h : φ.toRingHom.FaithfullyFlat) (w : S') (hw : ∀ (T : Type) [CommRing T] [Algebra F T] (ψ₁ ψ₂ : S' →ₐ[F] T), ψ₁.comp φ = ψ₂.comp φ → ψ₁ w = ψ₂ w) : ∃! v : S, φ v = w

theorem **_root_.GlobalTheory.faithfullyFlat_points_baseChange_descent** : (∀ (S B : Type) [CommRing S] [CommRing B] (φ : S →+* B), φ.Flat → (∀ (K : Type) [Field K] [IsAlgClosed K] (ψ : S →+* K), ∃ χ : B →+* K, χ.comp φ = ψ) → φ.FaithfullyFlat) ∧ (∀ (S B : Type) [CommRing S] [CommRing B] (φ : S →+* B) (e : B), φ.Flat → ((algebraMap B (Localization.Away e)).comp φ).Flat) ∧ (∀ (F S S₁ S₂ : Type) [Field F] [CommRing S] [CommRing S₁] [CommRing S₂] [Algebra F S] [Algebra F S₁] [Algebra F S₂] (φ₁ : S →ₐ[F] S₁) (φ₂ : S →ₐ[F] S₂), φ₂.toRingHom.FaithfullyFlat → ∃ (T : Type) (_ : CommRing T) (_ : Algebra F T) (ψ₁ : S₁ →ₐ[F] T) (ψ₂ : S₂ →ₐ[F] T), ψ₁.comp φ₁ = ψ₂.comp φ₂ ∧ ψ₁.toRingHom.FaithfullyFlat ∧ Function.Injective ψ₁) ∧ ∀ (F S S' : Type) [Field F] [CommRing S] [CommRing S'] [Algebra F S] [Algebra F S'] (φ : S →ₐ[F] S'), φ.toRingHom.FaithfullyFlat → ∀ w : S', (∀ (T : Type) [CommRing T] [Algebra F

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FamilyUnifIndexEqOfMarkedLatticeIsometry

lemma **isPeriod_comp_symm** {D' D'' : Datum H} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (gam : D'.La ≃ₗ[𝒪 E] D''.La) (hg : ∀ a b, D''.EF (gam a) (gam b) = D'.EF a b) {P : D'.La →+ (Fin g → ℂ)} (hP : IsPeriod D' ty P) : IsPeriod D'' ty (P.comp gam.symm.toLinearMap.toAddMonoidHom)

lemma **isHolTo_ex** {D : Datum H} {l : Level H} {Xo : Moduli D l} {σ : KA l →+* ℂ} {g : ℕ} {ty : Fin g → (E →+* ℂ)} (U : FamilyUnif Xo σ ty) (i : Fin U.n) {P : (U.Di i).La →+ (Fin g → ℂ)} (hP : IsPeriod (U.Di i) ty P) : IsHolTo Set.univ (fun z => (U.ex i P z).left)

theorem **index_eq_of_isometry** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (σ : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)), 1 ≤ D.p + D.q → σ (baseKA l).ρ = H.rt0 → IsTypeFn H D.p D.q ty → ∀ U : FamilyUnif Xo σ ty, ∀ (i j : Fin U.n) (gam : (U.Di i).La ≃ₗ[𝒪 E] (U.Di j).La), (∀ a b, (U.Di j).EF (gam a) (gam b) = (U.Di i).EF a b) → (∀ a, ∃ e, gam (U.t i a) = U.t j a + l.M' • e) → i = j

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FamilyUnifIndexEqOfMarkedLatticeIsometryStatement

theorem **_root_.GlobalTheory.T.familyUnif_index_eq_of_marked_latticeIsometry** : ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (σ : KA l →+* ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)), 1 ≤ D.p + D.q → σ (baseKA l).ρ = H.rt0 → IsTypeFn H D.p D.q ty → ∀ U : FamilyUnif Xo σ ty, ∀ (i j : Fin U.n) (gam : (U.Di i).La ≃ₗ[𝒪 E] (U.Di j).La), (∀ a b, (U.Di j).EF (gam a) (gam b) = (U.Di i).EF a b) → (∀ a, ∃ e, gam (U.t i a) = U.t j a + l.M' • e) → i = j

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FderivEqOfJetOf

theorem **jet_zero_ext** {κ : Type} (a b : Jet ℂ κ 0) (h : (jetAug : Jet ℂ κ 0 →ₐ[ℂ] ℂ) a = (jetAug : Jet ℂ κ 0 →ₐ[ℂ] ℂ) b) : a = b

private theorem **snd_aeval_eps_monomial_eq_ite** {κ : Type} [DecidableEq κ] (ℓ : κ) (e : κ →₀ ℕ) (c : ℂ) : (MvPolynomial.aeval (fun i => if i = ℓ then (DualNumber.eps : DualNumber ℂ) else 0) (MvPolynomial.monomial e c)).snd = if e = Finsupp.single ℓ 1 then c else 0

private theorem **single_one_mem_degLE_succ** {κ : Type} [Fintype κ] [DecidableEq κ] (k : ℕ) (ℓ : κ) : Finsupp.single ℓ 1 ∈ (degLE (k + 1) : Finset (κ →₀ ℕ))

private theorem **snd_jetTan_psJet_succ_eq_coeff_single** {κ : Type} [Fintype κ] [DecidableEq κ] (k : ℕ) (ℓ : κ) (F : MvPowerSeries κ ℂ) : (jetTan ℓ (psJet (k + 1) F) : DualNumber ℂ).snd = MvPowerSeries.coeff (Finsupp.single ℓ 1) F

theorem **fderiv_eq_of_jetOf** {g : ℕ} (m : ℕ) (θ θ' : (Fin g → ℂ) → ℂ) (h : PullbackSections.jetOf (m + 1) θ = PullbackSections.jetOf (m + 1) θ') : fderiv ℂ θ 0 = fderiv ℂ θ' 0

theorem **renInv_renN** (n g m : ℕ) (a : Jet ℂ (Fin 1 × Fin (n * g)) m) : renInv n g m (renN n g m a) = a

theorem **cancel_renN** {Y : Scheme.{0}} (n g m : ℕ) (x y : Sp (Jet ℂ (Fin 1 × Fin (n * g)) m) ⟶ Y) (h : spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ x = spMap (renN n g m : Jet ℂ (Fin 1 × Fin (n * g)) m →+* Jet ℂ (Fin n × Fin g) m) ≫ y) : x = y

private theorem **appLE_specMap_comp_fromSpecStalk_eq_germ** {Y : Scheme.{0}} {y : Y} {R : CommRingCat.{0}} (φ : Y.presheaf.stalk y ⟶ R) (U : Y.Opens) (hy : y ∈ U) (hx : ⊤ ≤ (Spec.map φ ≫ Y.fromSpecStalk y) ⁻¹ᵁ U) : (Spec.map φ ≫ Y.fromSpecStalk y).appLE U ⊤ hx = Y.presheaf.germ U y hy ≫ φ ≫ (Scheme.ΓSpecIso R).inv

theorem **jetOf_eq_of_jetPt** {g : ℕ} {Y : Scheme.{0}} (e e' : (Fin g → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (hol' : IsHolFam e') (m : ℕ) (h : PullbackSections.jetPt e hol m = PullbackSections.jetPt e' hol' m) (U : Y.Opens) (hU : e 0 cpt ∈ U) (hU' : e' 0 cpt ∈ U) (f : Γ(Y, U)) : PullbackSections.jetOf m (fun z => valAt' (e z) U f) = PullbackSections.jetOf m (fun z => valAt' (e' z) U f)

theorem **colsOf_surj** (n g : ℕ) : Function.Surjective (colsOf n g)

theorem **fderiv_comp_lin** {a b : ℕ} (θ : (Fin b → ℂ) → ℂ) (hθ : DifferentiableAt ℂ θ 0) (T : (Fin a → ℂ) →ₗ[ℂ] (Fin b → ℂ)) (v : Fin a → ℂ) : fderiv ℂ (fun z => θ (T z)) 0 v = fderiv ℂ θ 0 (T v)

private theorem **zero_mem_degLE'** {κ : Type} [Fintype κ] [DecidableEq κ] (k : ℕ) : (0 : κ →₀ ℕ) ∈ (degLE k : Finset (κ →₀ ℕ))

private theorem **jetAug_jetN_zero_eq_apply_zero** {n g : ℕ} (F : (Fin n → Fin g → ℂ) → ℂ) : (jetAug : Jet ℂ (Fin n × Fin g) 0 →ₐ[ℂ] ℂ) (jetN 0 F) = F 0

theorem **jet_uniq** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), Φ.Imm → ∀ (n m : ℕ) (ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left) (Λ Λ' : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.JetMap m ψ Λ → Φ.JetMap m ψ Λ' → ∀ f : (Fin g → ℂ) → ℂ, jetN m (fun w => f (Λ w)) = jetN m (fun w => f (Λ' w))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FinrankHost

theorem **finrank_host** (D : D7) : (Module.finrank ℚ D.host.E : ℝ) = 2 * (D.R.f : ℝ)

theorem **discr_rpow** (D : D7) : |(NumberField.discr D.host.E : ℝ)| ^ ((1 : ℝ) / D.R.f) = D.host.rdE ^ 2

theorem **two_le_dimY** (D : D7) {j : ℕ} (hj : j < D.R.m) (h1 : 1 ≤ j) : 2 ≤ WeightFamilies.dimY D j

theorem **core** (r f m ρ rd PP c T g DF q : ℝ) (hg : g = 1250 * r) (hr : 0 ≤ r) (hrT : r ≤ T) (hf : 1 ≤ f) (hm : 1 ≤ m) (hρ0 : 0 ≤ ρ) (hρ : ρ ≤ m) (hrd : 2 * f * m ≤ rd) (hPP : 0 < PP) (hq : 0 < q) (hc : c = rd ^ 7 * PP * DF ^ 2 / q) (hql : (rd ^ 7) ^ 4 * PP ^ 3 * DF ^ 2 ≤ q) (hgP : g * f * m ^ 3 ≤ PP ^ 2) : r * (1 + rd ^ 7 * T * c * (2 * f * ρ) ^ 2 * ρ / 2 / (1 / (2 * 5 ^ 4 * rd ^ 2))) ≤ 2 * T

theorem **numbers** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (K : KAdm) (Ch Ah Cq Aq : ℝ) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → TieC K H Ch Ah Cq Aq C2 A2 CR → PageAt large₂ C2 A2 → j < D.R.m → 1 ≤ j → 1 ≤ WeightAlgebra.DF wL → 0 < WeightAlgebra.cc D large₂ C2 A2 CR wL ∧ (Tn : ℝ) * WeightAlgebra.cc D large₂ C2 A2 CR wL * g0 D ≤ 1 ∧ 1 ≤ WeightFamilies.RRv D K.CRm ∧ (D.R.r : ℝ) * (1 + WeightFamilies.c1 D K.CRm Tn (WeightAlgebra.cc D large₂ C2 A2 CR wL) j / (1 / (2 * (5 : ℝ) ^ 4 * |(NumberField.discr D.host.E : ℝ)| ^ ((1 : ℝ) / D.R.f)))) ≤ 2 * Tn ∧ addL wF wL Tn = ka ∧ 2 ≤ WeightFamilies.dimY D j ∧ 11 * WeightFamilies.dimY D j ^ 2 ≤ Tn

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FinrankLowerBoundCuspFormVanishingLowCoeff

theorem **le_finrank_of_pieces** {K Z : Type} [Fintype K] [DecidableEq K] {Y : K → Type} (dom : ∀ k, Set (Y k)) (e : Z ≃ Σ k, dom k) (d : K → ℕ) (C δ : ℕ) (en : Fin δ ≃ Σ k, Fin (d k)) (f : ∀ k, Fin (d k * C) → Y k → ℂ) (hf : ∀ k (u : Fin (d k * C) → ℂ), (∀ x ∈ dom k, ∑ q, u q * f k q x = 0) → u = 0) (S : Submodule ℂ (Z → ℂ)) [FiniteDimensional ℂ S] (hS : ∀ k q, (fun z => (Pi.single k (f k q) : ∀ k, Y k → ℂ) (e z).1 (e z).2.1) ∈ S) : δ * C ≤ Module.finrank ℂ S

theorem **laR_wadd** (T : Top D) (j : ℕ) (u v : WeightFamilies.Wt D j) (k : (T.U j T.σ₀).Comp) : laR T j (wadd u v) k = laR T j u k + laR T j v k

theorem **laR_wsmul** (T : Top D) (j : ℕ) (m : ℕ) (u : WeightFamilies.Wt D j) (k : (T.U j T.σ₀).Comp) : laR T j (wsmul m u) k = m • laR T j u k

theorem **lower_bound** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨(Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → ∀ (CR c : ℝ), c1 = WeightFamilies.c1 D CR Tn c j → MggOnto h j CR Tn c (addL wF wL Tn) → δ * (Tn + WeightFamilies.dimY D j).choose (WeightFamilies.dimY D j) ≤ Module.finrank ℂ (R.sp (addL wF wL Tn) (WeightFamilies.MggK n.base CR Tn c j (addL wF wL Tn)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FinrankLowerBoundCuspFormVanishingLowCoeffStatement

theorem **_root_.GlobalTheory.T.finrank_lower_bound_cuspForm_vanishing_low_coeff** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨(Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → ∀ (CR c : ℝ), c1 = WeightFamilies.c1 D CR Tn c j → MggOnto h j CR Tn c (addL wF wL Tn) → δ * (Tn + WeightFamilies.dimY D j).choose (WeightFamilies.dimY D j) ≤ Module.finrank ℂ (R.sp (addL wF wL Tn) (WeightFamilies.MggK n.base CR Tn c j (addL wF wL Tn)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumBoundAtHalfSupremumCuspForm

theorem **firstMin_at_halfMax** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∀ f ∈ T.C j w, ∀ (k : (T.U j σ).Comp), ∀ z0 ∈ ((T.U j σ).X k).dom, (∀ x ∈ ((T.U j σ).X k).dom, ((T.U j σ).X k).pnorm (laU (T.U j σ) (wtA w) w.n k) (rdU (T.U j σ) f k) x ≤ 2 * ((T.U j σ).X k).pnorm (laU (T.U j σ) (wtA w) w.n k) (rdU (T.U j σ) f k) z0) → 0 < ((T.U j σ).X k).pnorm (laU (T.U j σ) (wtA w) w.n k) (rdU (T.U j σ) f k) z0 → 0 < ((T.U j σ).X k).firstMinimumSq z0 ∧ Real.log (1 / ((T.U j σ).X k).firstMinimumSq z0) ≤ 80 * large₂.sc.L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumBoundAtHalfSupremumCuspFormStatement

theorem **_root_.GlobalTheory.T.firstMinimum_bound_at_halfSupremum_cuspForm** : ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → ∀ f ∈ T.C j w, ∀ (k : (T.U j σ).Comp), ∀ z0 ∈ ((T.U j σ).X k).dom, (∀ x ∈ ((T.U j σ).X k).dom, ((T.U j σ).X k).pnorm (laU (T.U j σ) (wtA w) w.n k) (rdU (T.U j σ) f k) x ≤ 2 * ((T.U j σ).X k).pnorm (laU (T.U j σ) (wtA w) w.n k) (rdU (T.U j σ) f k) z0) → 0 < ((T.U j σ).X k).pnorm (laU (T.U j σ) (wtA w) w.n k) (rdU (T.U j σ) f k) z0 → 0 < ((T.U j σ).X k).firstMinimumSq z0 ∧ Real.log (1 / ((T.U j σ).X k).firstMinimumSq z0) ≤ 80 * large₂.sc.L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumEqZeroOfRankZero

theorem **expo_eq** (ν s : Ψ → ℝ) : expo ν s = cexp (((4 * Real.pi * ∑ τ, ν τ * s τ : ℝ) : ℂ) * I)

theorem **norm_expo** (ν s : Ψ → ℝ) : ‖expo ν s‖ = 1

theorem **expo_add** (ν s t : Ψ → ℝ) : expo ν (s + t) = expo ν s * expo ν t

theorem **expo_nsmul** (ν s : Ψ → ℝ) (k : ℕ) : expo ν (k • s) = expo ν s ^ k

theorem **continuous_expo** (ν : Ψ → ℝ) : Continuous (expo ν)

theorem **exists_expo_ne_one** {ι : Type*} [Countable ι] (ν : ι → Ψ → ℝ) : ∃ ν : Ψ → ℝ, ∀ i, ν i ≠ 0 → expo (ν i) ν ≠ 1

theorem **eq_zero_of_hasSum_pow** {ι : Type*} (B z : ι → ℂ) (c : ℂ) (hB : Summable fun i => ‖B i‖) (hz : ∀ i, ‖z i‖ = 1) (hz1 : ∀ i, B i ≠ 0 → z i ≠ 1) (hs : ∀ k : ℕ, HasSum (fun i => B i * z i ^ k) c) : c = 0

theorem **setIntegral_add_right** (L : AddSubgroup (Ψ → ℝ)) [Countable L] {D : Set (Ψ → ℝ)} (hD : IsAddFundamentalDomain L D volume) (f : (Ψ → ℝ) → ℂ) (hper : ∀ p ∈ L, ∀ s, f (s + p) = f s) (t : Ψ → ℝ) : ∫ s in D, f (s + t) = ∫ s in D, f s

theorem **setIntegral_eq_zero** (L : AddSubgroup (Ψ → ℝ)) [Countable L] {D : Set (Ψ → ℝ)} (hD : IsAddFundamentalDomain L D volume) (hfin : volume D < ⊤) (f : (Ψ → ℝ) → ℂ) (hper : ∀ p ∈ L, ∀ s, f (s + p) = f s) {ι : Type*} [Countable ι] (A : ι → ℂ) (hA : Summable fun i => ‖A i‖) (ν : ι → Ψ → ℝ) (hs : ∀ s, HasSum (fun i => A i * expo (ν i) s) (f s)) (hν : ∀ i, A i ≠ 0 → ν i ≠ 0) : ∫ s in D, f s = 0

theorem **fj_eq_zero_of_hasSum_ne** (X : Data E r) (la : Wt X.Ψ r) (ξ : Fin r → E) (x : Frame X.Ψ r) (Q : Frame X.Ψ r → ℂ) {ι : Type*} [Countable ι] (A : ι → ℂ) (hA : Summable fun i => ‖A i‖) (ν : ι → X.Ψ → ℝ) (hs : ∀ s : X.Ψ → ℝ, HasSum (fun i => A i * cexp (4 * Real.pi * I * ∑ τ, (ν i τ : ℂ) * s τ)) (Q (X.flow ξ (fun τ => (s τ : ℂ)) x))) (hper : ∀ p ∈ X.periods la, ∀ s : X.Ψ → ℝ, Q (X.flow ξ (fun τ => ((s + p) τ : ℂ)) x) = Q (X.flow ξ (fun τ => (s τ : ℂ)) x)) (μ : X.Ψ → ℝ) (hν : ∀ i, A i ≠ 0 → ν i ≠ μ) : X.fj la ξ μ Q x = 0

theorem **fj_zero_of_hasSum** (X : Data E r) (la : Wt X.Ψ r) (ξ : Fin r → E) (x : Frame X.Ψ r) (Q : Frame X.Ψ r → ℂ) {ι : Type*} [Countable ι] (A : ι → ℂ) (hA : Summable fun i => ‖A i‖) (ν : ι → X.Ψ → ℝ) (hs : ∀ s : X.Ψ → ℝ, HasSum (fun i => A i * cexp (4 * Real.pi * I * ∑ τ, (ν i τ : ℂ) * s τ)) (Q (X.flow ξ (fun τ => (s τ : ℂ)) x))) (hper : ∀ p ∈ X.periods la, ∀ s : X.Ψ → ℝ, Q (X.flow ξ (fun τ => ((s + p) τ : ℂ)) x) = Q (X.flow ξ (fun τ => (s τ : ℂ)) x)) (hν : ∀ i, A i ≠ 0 → ν i ≠ 0) : X.fj la ξ 0 Q x = 0

theorem **firstMinimum_eq_zero_of_rank_zero** (X : Data E 0) (x : Frame X.Ψ 0) : X.la1 x = 0

theorem **firstMinimum_bounded_on_domain** (X : Data E r) : ∃ B : ℝ, 0 < B ∧ ∀ x ∈ X.dom, X.la1 x ≤ B

theorem **rpow_mul_exp_neg_le** (a b : ℝ) (ha : 0 ≤ a) (hb : 0 < b) : ∃ M : ℝ, ∀ u : ℝ, 0 ≤ u → u ^ a * Real.exp (-b * u) ≤ M

theorem **fj_eq_zero_of_isSec** (X : Data E r) (la : Wt X.Ψ r) (Q : Frame X.Ψ r → ℂ) (hsec : X.IsSec la Q) {ξ : Fin r → E} (hξ : X.IsIso ξ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {ι : Type*} [Countable ι] (A : ι → ℂ) (hA : Summable fun i => ‖A i‖) (ν : ι → X.Ψ → ℝ) (hs : ∀ s : X.Ψ → ℝ, HasSum (fun i => A i * cexp (4 * Real.pi * I * ∑ τ, (ν i τ : ℂ) * s τ)) (Q (X.flow ξ (fun τ => (s τ : ℂ)) x))) (μ : X.Ψ → ℝ) (hν : ∀ i, A i ≠ 0 → ν i ≠ μ) : X.fj la ξ μ Q x = 0

theorem **bounded_of_moderate_nonneg** (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) (hsec : X.IsSec la P) (C A : ℝ) (hA0 : 0 ≤ A) (hmod : ∀ x ∈ X.dom, X.pnorm la P x ≤ C * X.la1 x ^ (-A)) (hcusp : ∀ ξ, X.IsIso ξ → ∀ x ∈ X.dom, X.fj la ξ 0 P x = 0) : ∃ B : ℝ, ∀ x ∈ X.dom, X.pnorm la P x ≤ B

theorem **mem_SS_of_isModerate** (X : Data E r) (la : Wt X.Ψ r) (Q : Frame X.Ψ r → ℂ) (hsec : X.IsSec la Q) (hmod : X.IsModerate la Q) (hcusp : ∀ ξ, X.IsIso ξ → ∀ x ∈ X.dom, X.fj la ξ 0 Q x = 0) : Q ∈ X.SS la

theorem **isModerate_of_le_one_add_inv** (X : Data E r) (la : Wt X.Ψ r) (Q : Frame X.Ψ r → ℂ) (C A : ℝ) (hC : ∀ x ∈ X.dom, X.pnorm la Q x ≤ C * (1 + 1 / X.la1 x) ^ A) : X.IsModerate la Q

theorem **mem_SS_of_le_one_add_inv** (X : Data E r) (la : Wt X.Ψ r) (Q : Frame X.Ψ r → ℂ) (hsec : X.IsSec la Q) (C A : ℝ) (hC : ∀ x ∈ X.dom, X.pnorm la Q x ≤ C * (1 + 1 / X.la1 x) ^ A) (hcusp : ∀ ξ, X.IsIso ξ → ∀ x ∈ X.dom, X.fj la ξ 0 Q x = 0) : Q ∈ X.SS la

theorem **decay_of_isModerate** (X : Data E r) (la : Wt X.Ψ r) (Q : Frame X.Ψ r → ℂ) (hsec : X.IsSec la Q) (hmod : X.IsModerate la Q) (hcusp : ∀ ξ, X.IsIso ξ → ∀ x ∈ X.dom, X.fj la ξ 0 Q x = 0) : ∃ c C : ℝ, 0 < c ∧ Q ∈ X.SSgt la c ∧ ∀ x ∈ X.dom, X.pnorm la Q x ^ 2 ≤ C * Real.exp (-2 * Real.pi * c / X.firstMinimumSq x)

theorem **main** (X : Data E r) (la : Wt X.Ψ r) (Q : Frame X.Ψ r → ℂ) : (∀ (ξ : Fin r → E) (x : Frame X.Ψ r) (ι : Type) [Countable ι] (A : ι → ℂ) (ν : ι → X.Ψ → ℝ), (Summable fun i => ‖A i‖) → (∀ s : X.Ψ → ℝ, HasSum (fun i => A i * cexp (4 * Real.pi * I * ∑ τ, (ν i τ : ℂ) * s τ)) (Q (X.flow ξ (fun τ => (s τ : ℂ)) x))) → (∀ p ∈ X.periods la, ∀ s : X.Ψ → ℝ, Q (X.flow ξ (fun τ => ((s + p) τ : ℂ)) x) = Q (X.flow ξ (fun τ => (s τ : ℂ)) x)) → (∀ i, A i ≠ 0 → ν i ≠ 0) → X.fj la ξ 0 Q x = 0) ∧ (X.IsSec la Q → (∀ ξ, X.IsIso ξ → ∀ x ∈ X.dom, X.fj la ξ 0 Q x = 0) → (X.IsModerate la Q ∨ ∃ C A : ℝ, ∀ x ∈ X.dom, X.pnorm la Q x ≤ C * (1 + 1 / X.la1 x) ^ A) → Q ∈ X.SS la)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumPrincipalTorusAtOrthonormalFrame

theorem **canFn_eq_canTh** (a b : Fin g → ℝ) (x : Frame X.Ψ r) (z : Fin g → ℂ) : canFn X bas idx a b x z = CanTh.canTh (tauMat X bas idx x) a b (hMat X bas idx x *ᵥ z)

theorem **differentiable_canFn** (hs : IsSympl X bas) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (a b : Fin g → ℝ) : Differentiable ℂ (canFn X bas idx a b x)

theorem **differentiable_prod_col** {V : Type*} [NormedAddCommGroup V] [NormedSpace ℂ V] {n : ℕ} {f : Fin n → V → ℂ} (hf : ∀ k, Differentiable ℂ (f k)) : Differentiable ℂ fun z : Fin n → V => ∏ k, f k (z k)

theorem **canFn_rmul** (a b : Fin g → ℝ) (x : Frame X.Ψ r) {u : Frame X.Ψ r} (hu : u ∈ Bor X.Ψ r) (z : Fin g → ℂ) : canFn X bas idx a b (rmul x u) ((borMat X u).submatrix idx idx *ᵥ z) = canFn X bas idx a b x z

theorem **borMat_rowOf_eq_zero** {u : Frame X.Ψ r} (hu : u ∈ Bor X.Ψ r) (i j : X.Ψ × (Fin q0 ⊕ Fin X.p0)) (hij : i ≠ j) (h : ¬ (i.1 = j.1 ∧ Harm.RowLT i.2 j.2)) : borMat X u (rowOf X hr i) (rowOf X hr j) = 0

theorem **bchar_wtOf_mul** {u : Frame X.Ψ r} (hu : u ∈ Bor X.Ψ r) (c : X.Ψ → Fin q0 → ℕ) (c' : X.Ψ → Fin X.p0 → ℕ) (n : ℕ) : bchar (wtOf X hr c c' n) u * ∏ j, rowFac X (fun τ a => u τ a a) (rowOf X hr j) ^ Sum.elim (Harm.partOf (c j.1)) (Harm.partOf (c' j.1)) j.2 = (∏ p, (rowFac X (fun τ a => u τ a a) p)⁻¹) ^ (n / 2)

theorem **thFr_rmul** (hs : IsSympl X bas) (hP : P ∈ Harm.rowHighest ℂ c c') {x : Frame X.Ψ r} (hx : x ∈ X.dom) {u : Frame X.Ψ r} (hu : u ∈ Bor X.Ψ r) : thFr X hr bas idx n a b P (rmul x u) = bchar (wtOf X hr c c' n) u * thFr X hr bas idx n a b P x

theorem **pnorm_thFr_rmul** (hs : IsSympl X bas) (hP : P ∈ Harm.rowHighest ℂ c c') {x : Frame X.Ψ r} (hx : x ∈ X.dom) {u : Frame X.Ψ r} (hu : u ∈ Bor X.Ψ r) : X.pnorm (wtOf X hr c c' n) (thFr X hr bas idx n a b P) (rmul x u) = X.pnorm (wtOf X hr c c' n) (thFr X hr bas idx n a b P) x

theorem **scaleUp_apply** (z : Fin g → ℂ) (i : Fin g) : scaleUp X idx z i = (√(X.c (idx i).1) : ℂ) * z i

theorem **Hf_scaleUp** (z w : Fin g → ℂ) : Hf (scaleUp X idx z) (scaleUp X idx w) = hermForm (Matrix.diagonal fun i => (X.c (idx i).1 : ℂ)) z w

theorem **canSec_scaleUp** (a b : Fin g → ℝ) (x : Frame X.Ψ r) (z : Fin g → ℂ) : canSec X bas idx a b x (scaleUp X idx z) = canFn X bas idx a b x z

theorem **differentiable_canSec** (hs : IsSympl X bas) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (a b : Fin g → ℝ) : Differentiable ℂ (canSec X bas idx a b x)

theorem **mem_dom_of_isOrtho** (ho : X.IsOrtho x) : x ∈ X.dom

theorem **normCoord_scaleUp** (z : Fin g → ℂ) : normCoord idx hs ho (scaleUp X idx z) = hMat X bas idx x *ᵥ z

theorem **Hf_eq_normCoord** (v w : EuclideanSpace ℂ (Fin g)) : Hf v w = hermForm (CanTh.imInv (tauMat X bas idx x)) (normCoord idx hs ho v) (normCoord idx hs ho w)

theorem **mem_lat_frameTorus** (y : EuclideanSpace ℂ (Fin g)) : y ∈ lat (frameTorus idx hs ho a b).b ↔ ∃ v ∈ X.Λ, y = scaleUp X idx (per X x v ∘ idx)

theorem **frameTorus_ka** (m k : Fin g → ℤ) : (frameTorus idx hs ho a b).ka (scaleUp X idx (per X x (bas (Sum.elim m k)) ∘ idx)) = CanTh.canMult a b m k

theorem **canSec_eq** : canSec X bas idx a b x = fun v => CanTh.canTh (tauMat X bas idx x) a b (normCoord idx hs ho v)

theorem **canSec_mem_Sec** : canSec X bas idx a b x ∈ (frameTorus idx hs ho a b).Sec 1

theorem **l2_canSec_eq_inv_sqrt_two_pow_mul_det_im** : (frameTorus idx hs ho a b).l2 1 (canSec X bas idx a b x) = (√(2 ^ g * ((tauMat X bas idx x).map Complex.im).det))⁻¹

theorem **l2_canSec_pos** : 0 < (frameTorus idx hs ho a b).l2 1 (canSec X bas idx a b x)

theorem **norm_scaleUp_per** (v : Fin r → E) : ‖scaleUp X idx (per X x v ∘ idx)‖ = √(∑ τ, X.c τ * X.maj x τ (Data.emb τ.1 v))

theorem **per_ne_zero** {v : Fin r → E} (hv : v ∈ X.Λ) (h0 : v ≠ 0) : per X x v ∘ idx ≠ 0

theorem **firstMinimum_principalTorusAtOrthonormalFrame** : (frameTorus idx hs ho a b).la1 = X.la1 x

theorem **thFr_eq_orthoJet** (hs : IsSympl X bas) (hP : P ∈ Harm.rowHighest ℂ c c') {x : Frame X.Ψ r} (hx : x ∈ X.dom) : thFr X hr bas idx n a b P x = (hMat X bas idx x).det ^ (n / 2) * (∏ τ, (√(X.c τ) : ℂ) ^ (Harm.partSize (c τ) + Harm.partSize (c' τ))) * orthoJet X bas idx hr a b P x

theorem **pnorm_thFr_of_isOrtho** (hs : IsSympl X bas) (hn : Even n) (hP : P ∈ Harm.rowHighest ℂ c c') {x : Frame X.Ψ r} (ho : X.IsOrtho x) (la : Wt X.Ψ r) : X.pnorm la (thFr X hr bas idx n a b P) x = (∏ τ, (X.c τ / 2) ^ ((r * n : ℝ) / 4) * X.c τ ^ (((Harm.partSize (c τ) + Harm.partSize (c' τ) : ℕ) : ℝ) / 2)) * ‖orthoJet X bas idx hr a b P x‖ / ∏ k, √((frameTorus idx hs ho (a k) (b k)).l2 1 (canSec X bas idx (a k) (b k) x))

theorem **norm_orthoJet_le** (hs : IsSympl X bas) {d : ℕ} (hP : P.IsHomogeneous d) {x : Frame X.Ψ r} (ho : X.IsOrtho x) : ‖orthoJet X bas idx hr a b P x‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * (π * Real.exp 1 * d) ^ ((d : ℝ) / 2) * (√2 * (1 + 4 * √(2 * g) / (√π * X.la1 x)) ^ g) ^ n * ∏ k, √((frameTorus idx hs ho (a k) (b k)).l2 1 (canSec X bas idx (a k) (b k) x))

theorem **pnorm_thFr_le** (hs : IsSympl X bas) (hn : Even n) (hP : P ∈ Harm.rowHighest ℂ c c') {x : Frame X.Ψ r} (hx : x ∈ X.dom) : X.pnorm (wtOf X hr c c' n) (thFr X hr bas idx n a b P) x ≤ (∏ τ, (X.c τ / 2) ^ ((r * n : ℝ) / 4) * X.c τ ^ (((Harm.partSize (c τ) + Harm.partSize (c' τ) : ℕ) : ℝ) / 2)) * ((∑ m ∈ P.support, ‖P.coeff m‖) * (π * Real.exp 1 * Harm.typeDegree c c') ^ ((Harm.typeDegree c c' : ℝ) / 2) * (√2 * (1 + 4 * √(2 * g) / (√π * X.la1 x)) ^ g) ^ n)

theorem **exists_pnorm_thFr_le** (hs : IsSympl X bas) (hn : Even n) (hP : P ∈ Harm.rowHighest ℂ c c') : ∃ C : ℝ, ∀ x ∈ X.dom, X.pnorm (wtOf X hr c c' n) (thFr X hr bas idx n a b P) x ≤ C * (1 + 1 / X.la1 x) ^ (g * n)

theorem **isModerate_thFr** (hs : IsSympl X bas) (hn : Even n) (hP : P ∈ Harm.rowHighest ℂ c c') : X.IsModerate (wtOf X hr c c' n) (thFr X hr bas idx n a b P)

theorem **isModerate_sum_thFr** (hs : IsSympl X bas) (hn : Even n) {J : Type*} (s : Finset J) (t : J → ℂ) (a b : J → Fin n → Fin g → ℝ) {P : J → Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : ∀ j ∈ s, P j ∈ Harm.rowHighest ℂ c c') : X.IsModerate (wtOf X hr c c' n) fun x => ∑ j ∈ s, t j * thFr X hr bas idx n (a j) (b j) (P j) x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumSqGeAndPhiEqZeroDefinite

theorem **hermC_mulVec_mulVec** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (P : Matrix (Fin r) (Fin r) ℂ) (w : Fin r → ℂ) : FD.hermC τ G (P.mulVec w) (P.mulVec w) = ∑ a, ∑ b, w a * FD.gram τ G P a b * conj (w b)

theorem **hermC_neg** (X : FD.Data E r) (h0 : X.p0 = 0) (τ : X.Ψ) {v : Fin r → ℂ} (hv : v ≠ 0) : ∃ s : ℝ, 0 < s ∧ FD.hermC τ.1 X.Gm v v = -(s : ℂ)

theorem **map_herm** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (u v : Fin r → E) : τ (FD.herm G u v) = FD.hermC τ G (FD.Data.emb τ u) (FD.Data.emb τ v)

theorem **posPart_eq_zero** (X : FD.Data E r) (h0 : X.p0 = 0) (x : FD.Frame X.Ψ r) (τ : X.Ψ) (v : Fin r → ℂ) : X.posPart x τ v = 0

theorem **sum_norm_eq_two_mul** (X : FD.Data E r) (y : E) : ∑ σ : E →+* ℂ, ‖σ y‖ = ∑ τ : X.Ψ, 2 * ‖τ.1 y‖

private theorem **one_le_abs_norm_mul_abs_discr_of_trace_int** {β : E} (hβ : ∀ a : 𝒪 E, ∃ k : ℤ, Algebra.trace ℚ E (β * a) = k) (h0 : β ≠ 0) : (1 : ℚ) ≤ |Algebra.norm ℚ β| * |(NumberField.discr E : ℚ)|

theorem **finrank_div_rd_le** {β : E} (hβ : ∀ a : 𝒪 E, ∃ k : ℤ, Algebra.trace ℚ E (β * a) = k) (h0 : β ≠ 0) : (Module.finrank ℚ E : ℝ) / I2.PT.rd E ≤ ∑ σ : E →+* ℂ, ‖σ β‖

theorem **two_f_div_rd_le_la1sq** (X : FD.Data E r) (hr : 0 < r) (h0 : X.p0 = 0) (hint : ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * FD.herm X.Gm u v) = k) (x : FD.Frame X.Ψ r) : 2 * (I2.PT.f E : ℝ) / I2.PT.rd E ≤ X.firstMinimumSq x

theorem **th_sq_le** {q : ℕ} {R0 : ℝ} (hrd : 1 ≤ I2.PT.rd E) (hq : 0 < q) (hb : 1 ≤ I2.b0 (E

theorem **firstMinimumSq_ge_and_phi_eq_zero_definite** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : FD.Data E r), 0 < r → X.p0 = 0 → (∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * FD.herm X.Gm u v) = k) → ∀ x ∈ X.dom, 2 * (I2.PT.f E : ℝ) / I2.PT.rd E ≤ X.firstMinimumSq x ∧ ∀ (q : ℕ) (R0 : ℝ), 1 ≤ I2.PT.rd E → 0 < q → 1 ≤ I2.b0 (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumSqGeAndPhiEqZeroDefiniteStatement

theorem **_root_.GlobalTheory.T.la1sq_ge_two_f_div_rd_of_p0_eq_zero** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : FD.Data E r), 0 < r → X.p0 = 0 → (∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * FD.herm X.Gm u v) = k) → ∀ x ∈ X.dom, 2 * (I2.PT.f E : ℝ) / I2.PT.rd E ≤ X.firstMinimumSq x ∧ ∀ (q : ℕ) (R0 : ℝ), 1 ≤ I2.PT.rd E → 0 < q → 1 ≤ I2.b0 (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstMinimumSqMulIInfHeightLeCard

theorem **finsum_mem_le_card_mul** {α ι : Type*} [Fintype ι] (g : ι → α) (S : Set α) (hS : S ⊆ Set.range g) (f : α → ℝ) (M : ℝ) (hM : 0 ≤ M) (hf : ∀ a ∈ S, f a ≤ M) : ∑ᶠ (a : α) (_ : a ∈ S), f a ≤ Fintype.card ι * M

theorem **exists_coords** {ι κ : Type} [Fintype ι] [Fintype κ] (b : Module.Basis κ ℝ (EuclideanSpace ℂ ι)) (v : EuclideanSpace ℂ ι) (hv : v ∈ ZSpan.fundamentalDomain b) (hL : ((5 : ℂ) ^ 2) • v ∈ lat b) : ∃ k : κ → Fin 25, ∑ i, (((k i : ℕ) : ℝ) / 25) • b i = v

theorem **log_le_of_bounds** {K S la th c k B p3 p1 f t : ℝ} (hK : 0 < K) (hS : 0 < S) (hla : 0 < la) (hth : 0 < th) (hc : 0 < c) (hk : 0 < k) (hB : 0 ≤ B) (hf : 0 < f) (ht : la ^ 2 * t ≤ f) (h1 : la < th → S ≤ K * (Real.exp p3 * (k * (c / la ^ 2)) ^ B * Real.exp (-(Real.pi * (c / la ^ 2) / 5 ^ 4))) ^ 2) (h2 : th ≤ la → S ≤ K * Real.exp p1 ^ 2) : Real.log S ≤ (Real.log K + 2 * |p3| + 2 * B * |Real.log (k * c)| + 2 * |p1| + 2 * Real.pi * c / (5 ^ 4 * th ^ 2)) + 4 * B * Real.log (1 + 1 / la) - 2 * Real.pi * c / (5 ^ 4 * f) * t

theorem **two_le_rank** (X : Data E r) (ξ : Fin r → E) (h : X.IsIso ξ) : 2 ≤ r

theorem **firstMinimum_sq_mul_iInf_height_le_card** (X : Data E r) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {ξ : Fin r → E} (hξ : ξ ∈ X.Λ) (h0 : ξ ≠ 0) : X.la1 x ^ 2 * ⊓ τ, X.tt ξ x τ ≤ Fintype.card X.Ψ

theorem **prod_eq_sq** (T : PT E) (s : (Fin 1 → Vsp T.dim) → ℂ) (a : Fin T.r → Vsp T.dim) : ∏ b, T.ptN2 1 s (fun _ => a b) / T.l2 1 s = (∏ j, Real.sqrt (T.ptN2 1 s fun _ => a j) / Real.sqrt (T.l2 1 s)) ^ 2

theorem **finsum_le** (T : PT E) (S : Set (Fin T.r → Vsp T.dim)) (hS : ∀ a ∈ S, (∀ j, a j ∈ ZSpan.fundamentalDomain T.b) ∧ ∀ j, ((5 : ℂ) ^ 2) • a j ∈ T.L) (f : (Fin T.r → Vsp T.dim) → ℝ) (M : ℝ) (hM : 0 ≤ M) (hf : ∀ a ∈ S, f a ≤ M) : ∑ᶠ (a) (_ : a ∈ S), f a ≤ (25 : ℝ) ^ (2 * (Module.finrank ℚ E * T.r) * T.r) * M

theorem **growth_decay** : TStatement → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ ξ : Fin r → E, X.toData.IsIso ξ) → (PT.f E : ℝ) ≤ PT.rd E → (PT.f E : ℝ) ^ 2 ≤ PT.rd E → 16 ≤ PT.rd E → (∃ m : ℕ, 1 ≤ m ∧ m ^ 4 ≤ PT.f E ∧ (X.p0 ≤ m ∨ r - X.p0 ≤ m)) → ∃ C A' : ℝ, 0 ≤ A' ∧ ∀ x ∈ X.dom, ∀ A : TorusAt X.toData x, 0 < (∑ᶠ (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s) → Real.log (∑ᶠ (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) ((

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FirstRaisingMemEquivariant

theorem **sq_eq** (hζ : IsPrimitiveRoot ζ 4) : ζ * ζ = -1

theorem **emb_eq** (hζ : IsPrimitiveRoot ζ 4) (τ : E →+* ℂ) : τ ζ = Complex.I ∨ τ ζ = -Complex.I

theorem **cc_root** (hζ : IsPrimitiveRoot ζ 4) : cc ζ = -ζ

theorem **cc_add** (x y : E) : cc (x + y) = cc x + cc y

theorem **cc_intCast** (n : ℤ) : cc (n : E) = n

theorem **cc_half_root** (hζ : IsPrimitiveRoot ζ 4) : cc (ζ / 2) = -(ζ / 2)

theorem **trace_root** (hζ : IsPrimitiveRoot ζ 4) : Algebra.trace ℚ E ζ = 0

theorem **finrank_eq** : Module.finrank ℚ E = 2

theorem **exists_coords** (hζ : IsPrimitiveRoot ζ 4) (a : 𝒪 E) : ∃ m n : ℤ, (a : E) = m + n * ζ

theorem **trace_form** (hζ : IsPrimitiveRoot ζ 4) (a b c d : ℤ) : Algebra.trace ℚ E (-(ζ / 2) * (((a : E) + b * ζ) * cc ((c : E) + d * ζ))) = ((b * c - a * d : ℤ) : ℚ)

theorem **mem_intLat** (x : Fin 1 → E) : x ∈ intLat E ↔ ∃ a : 𝒪 E, (a : E) = x 0

theorem **mem_intLat_iff** (x : Fin 1 → E) : x ∈ intLat E ↔ IsIntegral ℤ (x 0)

theorem **first_raising_mem_equivariant** : (e1 : Fin 1 → E) ∈ intLat E

theorem **herm_int** {x y : Fin 1 → E} (hx : x ∈ intLat E) (hy : y ∈ intLat E) : IsIntegral ℤ (herm (1 : Matrix (Fin 1) (Fin 1) E) x y)

theorem **eq_zero_of_iso** {ξ : Fin 1 → E} (h : herm (1 : Matrix (Fin 1) (Fin 1) E) ξ ξ = 0) : ξ = 0

theorem **mulVec_firstBasisVector** (γ : Matrix (Fin 1) (Fin 1) E) : Matrix.mulVec γ e1 0 = γ 0 0

theorem **norm_entry** {γ : GL (Fin 1) E} (hγ : IsUStab 1 (intLat E) γ) (φ : E →+* ℂ) : ‖φ ((γ : Matrix (Fin 1) (Fin 1) E) 0 0)‖ = 1

theorem **eq_one_of_cong** {γ : GL (Fin 1) E} (hγ : IsUStab 1 (intLat E) γ) (h : ∀ x ∈ intLat E, ∃ y ∈ intLat E, Matrix.mulVec (γ : Matrix (Fin 1) (Fin 1) E) x - x = ((3 : ℕ) : E) • y) : γ = 1

theorem **dmin_zero** {Ψ : Finset (E →+* ℂ)} (x : Frame Ψ 1) (τ : Ψ) : dmin (1 : Matrix (Fin 1) (Fin 1) E) x τ (Fin.castSucc 0) = 1

theorem **dmin_one** {Ψ : Finset (E →+* ℂ)} (x : Frame Ψ 1) (τ : Ψ) : dmin (1 : Matrix (Fin 1) (Fin 1) E) x τ (Fin.succ 0) = ((‖x τ 0 0‖ ^ 2 : ℝ) : ℂ)

theorem **pnorm_coordProd_le** (hζ : IsPrimitiveRoot ζ 4) (x : Frame (data hζ).Ψ 1) : (data hζ).pnorm 1 coordProd x ≤ 1

theorem **coordProd_mem** (hζ : IsPrimitiveRoot ζ 4) : coordProd ∈ (data hζ).SS 1

theorem **one_mem_dom** (hζ : IsPrimitiveRoot ζ 4) : (fun _ _ _ => 1 : Frame (data hζ).Ψ 1) ∈ (data hζ).dom

theorem **nonempty_type** (hζ : IsPrimitiveRoot ζ 4) : Nonempty (data hζ).Ψ

theorem **exists_cuspData_rank_one** : ∃ (X : DataX E4 1) (la : Wt X.Ψ 1) (P : Frame X.Ψ 1 → ℂ) (M : ℝ), P ∈ X.SS la ∧ la ≠ 0 ∧ (∃ x ∈ X.dom, P x ≠ 0) ∧ 0 < M ∧ X.Λ = X.L' ∧ (∀ v ∈ X.L', herm X.Gm v v = 0 → ∀ b : E4, cc b = b → IsIntegral ℤ b → ∃ γ ∈ X.Γ, X.χ la γ = 1 ∧ ∀ τ : X.Ψ, (γ : Matrix (Fin 1) (Fin 1) E4).map τ.1 = X.flowMat v τ ((M : ℂ) * τ.1 b))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FockLaws

theorem **fock_add** {g : ℕ} {Hm : Matrix (Fin g) (Fin g) ℂ} (hH : Hm.IsHermitian) (q : ℤ) (a b : Fin g → ℂ) (θ : (Fin g → ℂ) → ℂ) : fock Hm q (a + b) θ = fun z => Complex.exp ((q : ℂ) * Real.pi * Complex.I * ((hermForm Hm a b).im : ℂ)) * fock Hm q a (fock Hm q b θ) z

theorem **fock_sq** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (N : ℕ) (a : Fin g → ℂ) (θ : (Fin g → ℂ) → ℂ) : fock Hm ((N : ℤ) ^ 2) a (fun z => θ ((N : ℂ) • z)) = fun z => fock Hm 1 ((N : ℂ) • a) θ ((N : ℂ) • z)

theorem **fock_laws** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), (∀ (q : ℤ) (a b : Fin g → ℂ) (θ : (Fin g → ℂ) → ℂ), fock Φ.Hm q (a + b) θ = fun z => Complex.exp ((q : ℂ) * Real.pi * Complex.I * ((hermForm Φ.Hm a b).im : ℂ)) * fock Φ.Hm q a (fock Φ.Hm q b θ) z) ∧ (∀ (N : ℕ) (a : Fin g → ℂ) (θ : (Fin g → ℂ) → ℂ), fock Φ.Hm ((N : ℤ) ^ 2) a (fun z => θ ((N : ℂ) • z)) = fun z => fock Φ.Hm 1 ((N : ℂ) • a) θ ((N : ℂ) • z)) ∧ ∀ θ, Φ.IsTh 1 θ → ∀ a ∈ Φ.lat, fock Φ.Hm 1 a θ = fun z => Φ.χ a * θ z

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FockTranslationLaws

theorem **herm_add_left** (u v w : EuclideanSpace ℂ ι) : Hf (u + v) w = Hf u w + Hf v w

theorem **herm_add_right** (u v w : EuclideanSpace ℂ ι) : Hf u (v + w) = Hf u v + Hf u w

theorem **herm_sub_left** (u v w : EuclideanSpace ℂ ι) : Hf (u - v) w = Hf u w - Hf v w

theorem **herm_sub_right** (u v w : EuclideanSpace ℂ ι) : Hf u (v - w) = Hf u v - Hf u w

theorem **herm_neg_left** (u w : EuclideanSpace ℂ ι) : Hf (-u) w = -Hf u w

theorem **herm_neg_right** (u w : EuclideanSpace ℂ ι) : Hf u (-w) = -Hf u w

theorem **herm_zero_right** (u : EuclideanSpace ℂ ι) : Hf u 0 = 0

theorem **herm_smul_right** (c : ℂ) (u w : EuclideanSpace ℂ ι) : Hf u (c • w) = (starRingEnd ℂ) c * Hf u w

theorem **herm_conj** (u w : EuclideanSpace ℂ ι) : (starRingEnd ℂ) (Hf u w) = Hf w u

theorem **herm_sub_swap** (u w : EuclideanSpace ℂ ι) : Hf w u = Hf u w - 2 * I * ((Hf u w).im : ℂ)

theorem **herm_self** (x : EuclideanSpace ℂ ι) : Hf x x = ((‖x‖ ^ 2 : ℝ) : ℂ)

theorem **herm_self_re** (x : EuclideanSpace ℂ ι) : (Hf x x).re = ‖x‖ ^ 2

theorem **herm_self_im** (x : EuclideanSpace ℂ ι) : (Hf x x).im = 0

theorem **herm_re_swap** (u w : EuclideanSpace ℂ ι) : (Hf w u).re = (Hf u w).re

theorem **herm_im_swap** (u w : EuclideanSpace ℂ ι) : (Hf w u).im = -(Hf u w).im

theorem **herm_differentiable** (a : EuclideanSpace ℂ ι) : Differentiable ℂ fun v : EuclideanSpace ℂ ι => Hf v a

theorem **fock_translation_laws** {ι : Type} [Fintype ι] (U : ℕ → EuclideanSpace ℂ ι → (EuclideanSpace ℂ ι → ℂ) → EuclideanSpace ℂ ι → ℂ) (hU : ∀ (q : ℕ) a F v, U q a F v = cexp (q * Real.pi * Hf v a - q * Real.pi * Hf a a / 2) * F (v - a)) (q : ℕ) : (∀ a b F, U q a (U q b F) = cexp (-(q * Real.pi * I * (Hf a b).im)) • U q (a + b) F) ∧ (∀ F, U q 0 F = F) ∧ (∀ a F, (fun v => U q a F (-v)) = U q (-a) (fun v => F (-v))) ∧ (∀ a F (M : ℕ), (U q a)^[M] F = U q ((M : ℂ) • a) F) ∧ (∀ a F v, ‖U q a F v‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v‖ ^ 2) = ‖F (v - a)‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v - a‖ ^ 2)) ∧ (∀ (Y : GT ι) F, F ∈ Y.Sec q ↔ Differentiable ℂ F ∧ ∀ γ ∈ lat Y.b, Y.ka γ

• U q γ F = F) ∧ (∀ (Y : GT ι) a, (∀ γ ∈ lat Y.b, ∃ k : ℤ, q * (Hf a γ).im = k) → ∀ F ∈ Y.Sec q, U q a F ∈ Y.Sec q) ∧ (∀ (φ : EuclideanSpace ℂ ι →L[ℂ] EuclideanSpace ℂ ι) (N : ℕ), (∀ v w, Hf (φ v) (φ w) = N * Hf v w) → ∀ a F,

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FormalPointSubstitutionSeriesValueAtTwoPoints

theorem **czQ** (b : Fin n → Q k d l) : CZ (fun j => (b j).1)

theorem **sb_apply** (b : Fin n → Q k d l) (f : PS k n d) : sb b f = ev (fun j => (b j).1) f

@[simp] theorem **sq_gen** (b : Fin n → Q k d l) (j : Fin n) : sq b (gen j) = b j

theorem **sb_sb** {m : ℕ} (b : Fin n → Q k d l) (a : Fin m → Q k d n) (f : PS k m d) : sb b (sb a f) = sb (fun j => sq b (a j)) f

theorem **sb_gen** (f : PS k n d) : sb gen f = f

@[simp] theorem **formalPointSubstitutionSeries_valueAtTwoPoints** (b : Fin n → Q k d l) (f : PS k 2 d) (u v : Q k d n) : sb b (E2 f u v) = E2 f (sq b u) (sq b v)

@[simp] theorem **formalPointSubstitutionSeries_valueAtThreePoints** (b : Fin n → Q k d l) (f : PS k 3 d) (u v w : Q k d n) : sb b (E3 f u v w) = E3 f (sq b u) (sq b v) (sq b w)

theorem **formalSeriesValueAtTwoPoints_genericPoints** (f : PS k 2 d) : E2 f (gen 0) (gen 1) = f

theorem **formalSeriesValueAtThreePoints_genericPoints** (f : PS k 3 d) : E3 f (gen 0) (gen 1) (gen 2) = f

@[simp] theorem **sq_ac** (b : Fin n → Q k d l) (A : Q k d 1) (u : Q k d n) : sq b (ac A u) = ac A (sq b u)

theorem **formalSeriesValueAtTwoPoints_eq_ev** (f : PS k 2 d) (u v : Q k d n) : E2 f u v = ev ![u.1, v.1] f

theorem **formalSeriesValueAtThreePoints_eq_ev** (f : PS k 3 d) (u v w : Q k d n) : E3 f u v w = ev ![u.1, v.1, w.1] f

theorem **ac_val** (A : Q k d 1) (u : Q k d n) : (ac A u).1 = comp1 A.1 u.1

theorem **op_val** (F : Q k d 2) (u v : Q k d n) : (op F u v).1 = addOf F.1 u.1 v.1

theorem **sq_op** (b : Fin n → Q k d l) (F : Q k d 2) (u v : Q k d n) : sq b (op F u v) = op F (sq b u) (sq b v)

theorem **gen_val** (j : Fin n) : (gen j : Q k d n).1 = pt k d j

theorem **ext_of_trunc** {R σ : Type} [CommRing R] [Finite σ] {f g : MvPowerSeries σ R} (h : ∀ m : ℕ, Ideal.Quotient.mk (MvPolynomial.idealOfVars σ R ^ (m + 1)) (truncTotal (m + 1) f) = Ideal.Quotient.mk _ (truncTotal (m + 1) g)) : f = g

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameBaseChangeEigenBasis

theorem **spMap_spMap** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

private theorem **isTangent_iff_isTanAt_of_isGrpPullback** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {G' : Over (Sp S)} [GrpObj G'] {φ : G'.left ⟶ Xo.univ.t.G.left} (h : IsGrpPullback G' Xo.univ.t.G x φ) (w : epsR S ⟶ G') : IsTangent w ↔ IsTanAt x (w.left ≫ φ)

private theorem **isLieComb_iff_isCombAt_of_isGrpPullback** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {G' : Over (Sp S)} [GrpObj G'] {φ : G'.left ⟶ Xo.univ.t.G.left} (h : IsGrpPullback G' Xo.univ.t.G x φ) {n : ℕ} (c : Fin n → S) (v : Fin n → (epsR S ⟶ G')) (w : epsR S ⟶ G') : IsLieComb c v w ↔ IsCombAt x (fun i => (v i).left ≫ φ) c (w.left ≫ φ)

private theorem **exists_tangent_lift_of_isGrpPullback** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {G' : Over (Sp S)} [GrpObj G'] {φ : G'.left ⟶ Xo.univ.t.G.left} (h : IsGrpPullback G' Xo.univ.t.G x φ) (w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left) (hw : IsTanAt x w) : ∃ w' : epsR S ⟶ G', w'.left ≫ φ = w

theorem **basis_baseChange** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) (x : Sp S ⟶ Xo.X) {n : ℕ} (u : Fin n → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)) (hu : ∀ i, IsTanAt x (u i)) (hb : ∀ w, IsTanAt x w → ∃! a : Fin n → S, IsCombAt x u a w) : (∀ i, IsTanAt (spMap ψ ≫ x) (spMap (dualMap ψ) ≫ u i)) ∧ ∀ w, IsTanAt (spMap ψ ≫ x) w → ∃! a : Fin n → T, IsCombAt (spMap ψ ≫ x) (fun i => spMap (dualMap ψ) ≫ u i) a w

theorem **epsEnd_comp** (a b : T) : epsEnd x a ≫ epsEnd x b = epsEnd x (a * b)

theorem **epsEnd_zero** (W : tEps x ⟶ Xo.univ.t.G) (hW : IsTanAt x W.left) : epsEnd x 0 ≫ W = 1

theorem **isTanAt_epsEnd** (a : T) (W : tEps x ⟶ Xo.univ.t.G) (hW : IsTanAt x W.left) : IsTanAt x (epsEnd x a ≫ W).left

theorem **isCombAt_iff** {m : ℕ} (v : Fin m → (tEps x ⟶ Xo.univ.t.G)) (cf : Fin m → T) (W : tEps x ⟶ Xo.univ.t.G) : IsCombAt x (fun j => (v j).left) cf W.left ↔ ∏ j, epsEnd x (cf j) ≫ v j = W

theorem **comp_prod** {κ : Type} [Fintype κ] (f : tEps x ⟶ tEps x) (u : κ → (tEps x ⟶ Xo.univ.t.G)) : f ≫ ∏ k, u k = ∏ k, f ≫ u k

theorem **prod_comp** {κ : Type} [Fintype κ] (φ : Xo.univ.t.G ⟶ Xo.univ.t.G) [IsMonHom φ] (u : κ → (tEps x ⟶ Xo.univ.t.G)) : (∏ k, u k) ≫ φ = ∏ k, u k ≫ φ

theorem **exists_embK_ne** (c : Coeff l) {σ τ : E →+* ℂ} (h : σ ≠ τ) : ∃ b : 𝒪 E, embK c σ b ≠ embK c τ b

theorem **eigen_basis** (c : Coeff l) [Algebra c.F T] (U : Lt1 Dw → (tEps x ⟶ Xo.univ.t.G)) (hU : ∀ k, IsTanAt x (U k).left) (hE : ∀ k, IsEigAt c k.1 (U k).left) (hb : ∀ W : tEps x ⟶ Xo.univ.t.G, IsTanAt x W.left → ∃! a : Lt1 Dw → T, ∏ k, epsEnd x (a k) ≫ U k = W) (τ : E →+* ℂ) (W : tEps x ⟶ Xo.univ.t.G) (hW : IsTanAt x W.left) (hWE : IsEigAt c τ W.left) : ∃! a : Fin (Dw.mult τ) → T, ∏ i, epsEnd x (a i) ≫ U (τ, i) = W

theorem **frame_baseChange_eigen_basis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S) (τ : E →+* ℂ) (w : Sp (DualNumber T) ⟶ Xo.univ.t.G.left), IsTanAt (spMap ψ.toRingHom ≫ y.x) w → IsEigAt c τ w → ∃! a : Fin (Dw.mult τ) → T, IsCombAt (spMap ψ.toRingHom ≫ y.x) (fun i => spMap (dualMap ψ.toRingHom) ≫ y.u τ i) a w

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameBaseChangeEigenBasisStatement

theorem **_root_.GlobalTheory.T.existsUnique_isCombAt_baseChange_of_isEigAt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S) (τ : E →+* ℂ) (w : Sp (DualNumber T) ⟶ Xo.univ.t.G.left), IsTanAt (spMap ψ.toRingHom ≫ y.x) w → IsEigAt c τ w → ∃! a : Fin (Dw.mult τ) → T, IsCombAt (spMap ψ.toRingHom ≫ y.x) (fun i => spMap (dualMap ψ.toRingHom) ≫ y.u τ i) a w

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameBiextDictionarySecondPartialLaw

theorem **jQ_psJet** (b : Fin k → Q S d n) (f : PS S k d) : jQ m b (psJet m f) = psJet m (sb b f)

theorem **jQ_X** (b : Fin k → Q S d n) (s : Fin k × Fin d) : jQ m b (Ideal.Quotient.mk _ (MvPolynomial.X s)) = psJet m ((b s.1).1 s.2)

theorem **psJet_X** {κ : Type} [Fintype κ] [DecidableEq κ] (s : κ) : psJet m (MvPowerSeries.X s : MvPowerSeries κ S) = Ideal.Quotient.mk _ (MvPolynomial.X s)

theorem **psJet_zero** {κ : Type} [Fintype κ] [DecidableEq κ] : psJet m (0 : MvPowerSeries κ S) = 0

theorem **jQ_comp** (b : Fin n → Q S d n') (a : Fin k → Q S d n) : (jQ m b).comp (jQ m a) = jQ m fun j => sq b (a j)

theorem **subst_eq** (φ : Fin k × Fin d → MvPolynomial (Fin n × Fin d) S) (a : Fin k → Q S d n) (h : ∀ s, Ideal.Quotient.mk _ (φ s) = psJet m ((a s.1).1 s.2)) (hc : ∀ s, MvPolynomial.constantCoeff (φ s) = 0) : jetSubstP m φ = jQ m a

theorem **mk_psIn** (r : Fin k → Fin n) (f : PS S k d) : Ideal.Quotient.mk _ (psIn d m r f) = psJet m (sb (fun i => (gen (r i) : Q S d n)) f)

theorem **ps_ext** {κ : Type} [Fintype κ] [DecidableEq κ] {f g : MvPowerSeries κ S} (h : ∀ m, psJet m f = psJet m g) : f = g

theorem **PP_sq** (b : Fin n → Q S d n') (u : Q S d n) : homAt x0 (jQ m b) ≫ PP m x u = PP m x (sq b u)

theorem **PP_gen** (i : Fin n) : jptG t x0 d x m i = PP m x (gen i)

theorem **VV_mul** (y : (jOG x0 d m n).left) (f g : PS S n d) : VV m y (f * g) = VV m y f * VV m y g

theorem **VV_one** (y : (jOG x0 d m n).left) : VV m y (1 : PS S n d) = 1

theorem **VV_ext** (hL : t.PP.L.IsLB) {f g : PS S n d} (h : ∀ m (y : (jOG x0 d m n).left) J, Good t.PP.L y J → VV m y f = VV m y g) : f = g

theorem **val_peG** (e : ∀ m, FrTyG t x0 d x m) (b : Fin k → Q S d n) (φ : Fin 2 → Fin d → MvPolynomial (Fin k × Fin d) S) (p q : Q S d k) (h : ∀ s : Fin 2 × Fin d, Ideal.Quotient.mk _ (φ s.1 s.2) = psJet m (((![p, q] : Fin 2 → _) s.1).1 s.2)) (hc : ∀ s : Fin 2 × Fin d, MvPolynomial.constantCoeff (φ s.1 s.2) = 0) : val (peG t x0 d x e m φ) (homAt x0 (jQ m b)).left y J = ee x e y J (sq b p) (sq b q)

theorem **val_scale** {Y : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ t.G.left)} {ε : K → ℤ} (u : Γ(Y, ⊤)ˣ) (g : LB.Gen (LB.LinComb t.PP.L φ ε)) (r : (jOG x0 d m n).left ⟶ Y) : val (scaleGenG t φ ε u g) r y J = gm r y ⊤ u * val g r y J

theorem **gm_unit** (u : (Jet S (Fin k × Fin d) m)ˣ) (b : Fin k → Q S d n) (f : PS S k d) (hf : (u : Jet S (Fin k × Fin d) m) = psJet m f) : (gm (homAt x0 (jQ m b)).left y ⊤ (unitΓ u)).val = VV m y (sb b f)

theorem **PP_op** (u v : Q S d n) : PP m x (op (FQ G) u v) = PP m x u * PP m x v

theorem **PP_ac** (a : 𝒪 E) (u : Q S d n) : PP m x (ac (AQ A a) u) = PP m x u ≫ t.act.act a

theorem **PP_zero** : PP m x (0 : Q S d n) = 1

theorem **hX** (i : Fin k) (j : Fin d) : (Ideal.Quotient.mk _ (MvPolynomial.X (i, j)) : Jet S (Fin k × Fin d) m) = psJet m ((gen i : Q S d k).1 j)

theorem **hF** (i i' : Fin k) (j : Fin d) : (Ideal.Quotient.mk _ (psIn d m ![i, i'] (G.F j)) : Jet S (Fin k × Fin d) m) = psJet m ((op (FQ G) (gen i) (gen i') : Q S d k).1 j)

theorem **hA** (a : 𝒪 E) (i : Fin k) (j : Fin d) : (Ideal.Quotient.mk _ (psIn d m ![i] (A.act a j)) : Jet S (Fin k × Fin d) m) = psJet m ((ac (AQ A a) (gen i) : Q S d k).1 j)

theorem **peXX** (b : Fin k → Q S d n) (i i' : Fin k) : val (peG t x0 d x e m ![fun j => MvPolynomial.X (i, j), fun j => MvPolynomial.X (i', j)]) (homAt x0 (jQ m b)).left y J = ee x e y J (b i) (b i')

theorem **val_th** (b : Fin 3 → Q S d n) (i₀ i₁ i₂ : Fin 3) : val (LB.pullGen t.PP.L (thφ t.G (x3 t.G)) thε t.cub.t (lift (jptG t x0 d x m i₀) (lift (jptG t x0 d x m i₁) (jptG t x0 d x m i₂))).left) (homAt x0 (jQ m b)).left y J = cubV t.cub y J (PP m x (b i₀)) (PP m x (b i₁)) (PP m x (b i₂))

theorem **frame_biext_dictionary_first_partial_law** (u v w : Q S d n) : VV m y (E3 ph1 u v w) = (cubV t.cub y J (PP m x u) (PP m x v) (PP m x w) * (ee x e y J (op (FQ G) u v) w)⁻¹ * ee x e y J u w * ee x e y J v w).val

theorem **frame_biext_dictionary_second_partial_law** (u v w : Q S d n) : VV m y (E3 ph2 u v w) = (cubV t.cub y J (PP m x u) (PP m x v) (PP m x w) * (ee x e y J u (op (FQ G) v w))⁻¹ * ee x e y J u v * ee x e y J u w).val

theorem **dict_sy** (u v : Q S d n) : VV m y (E2 sy u v) = (ee x e y J v u * (ee x e y J u v)⁻¹).val

theorem **dict_ad** (a : 𝒪 E) {K : Type} [Fintype K] {φ : K → ((t.G ⊗ t.G).left ⟶ t.G.left)} {ε : K → ℤ} (wL : LB.Gen (LB.LinComb t.PP.L φ ε)) (U : (Jet S (Fin 2 × Fin d) m)ˣ) (hU : (U : Jet S (Fin 2 × Fin d) m) = psJet m (ad a)) (h : LB.Gen.Corr t.PP.L _ _ _ _ (LB.mulGen t.PP.L _ _ _ _ (LB.pullGen t.PP.L _ _ wL (lift (jptG t x0 d x m (0 : Fin 2)) (jptG t x0 d x m (1 : Fin 2))).left) (peG t x0 d x e m ![fun j => psIn d m ![0] (A.act a j), fun j => MvPolynomial.X ((1 : Fin 2), j)])) (scaleGenG t _ _ (unitΓ U) (peG t x0 d x e m ![fun j => MvPolynomial.X ((0 : Fin 2), j), fun j => psIn d m ![1] (A.act (H.bar a) j)])) (u v : Q S d n) : VV m y (E2 (ad a) u v) = (val₂ wL y J (PP m x u) (PP m x v) * ee x e y J (ac (AQ A a) u) v * (ee x e y J u (ac (AQ A (H.bar a)) v))⁻¹).val

theorem **dict_ax** (ha : FrAxes t x0 d x e) (u : Q S d n) : ee x e y J 0 u = 1 ∧ ee x e y J u 0 = 1

theorem **at₂** {p q : Q S d 2} (h : p.1 = q.1) (u v : Q S d n) : sq ![u, v] p = sq ![u, v] q

theorem **opc** (u v : Q S d n) : op (FQ G) u v = op (FQ G) v u

theorem **op0** (u : Q S d n) : op (FQ G) 0 u = u

theorem **opa** (u v w : Q S d n) : op (FQ G) (op (FQ G) u v) w = op (FQ G) u (op (FQ G) v w)

theorem **ac0** (a : 𝒪 E) : ac (AQ A a) (0 : Q S d n) = 0

theorem **ac_op** (a : 𝒪 E) (u v : Q S d n) : ac (AQ A a) (op (FQ G) u v) = op (FQ G) (ac (AQ A a) u) (ac (AQ A a) v)

theorem **ac_add** (a b : 𝒪 E) (u : Q S d n) : ac (AQ A (a + b)) u = op (FQ G) (ac (AQ A a) u) (ac (AQ A b) u)

theorem **ac_mul** (a b : 𝒪 E) (u : Q S d n) : ac (AQ A (a * b)) u = ac (AQ A a) (ac (AQ A b) u)

theorem **ac_one** (u : Q S d n) : ac (AQ A 1) u = u

theorem **fbId** : FBIdSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameLetterGLActionRepresentationKeepsHarmonic

theorem **IsEnum.pl_idx** (h : IsEnum H pl) (t : {τ : E →+* ℂ // InPsi H τ}) : pl (h.idx t) = t.1

theorem **rename_letAct** (h : IsEnum H pl) {K : Type} [Field K] (g : Ix → GL (Fin n) K) (P : MvPolynomial (LtN Dw n) K) : rename (thetaLetterEquiv Dw n) (letAct Dw pl K g P) = colSubst (fun t => (g (h.idx t) : Matrix (Fin n) (Fin n) K)ᵀ) (fun t => (((g (h.idx t))⁻¹ : GL (Fin n) K) : Matrix (Fin n) (Fin n) K)ᵀ) (rename (thetaLetterEquiv Dw n) P)

theorem **letAct_mul_one_mem_HHa** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (K : Type) [Field K] [CharZero K] (a : (E →+* ℂ) × ℕ → ℕ), (∀ g g' : Ix → GL (Fin n) K, letAct Dw pl K (g * g') =

(letAct Dw pl K g).comp (letAct Dw pl K g')) ∧ letAct Dw pl K (1 : Ix → GL (Fin n) K) = AlgHom.id K _ ∧ ∀ (g : Ix → GL (Fin n) K), ∀ P ∈ HHa Dw.mult n K ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), letAct Dw pl K g P ∈ HHa Dw.mult n K ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameReadingLeadingMinorProductExtremal

theorem **PM.rfl'** (x : R) : PM x x

theorem **PM.mul** {x y x' y' : R} (h : PM x y) (h' : PM x' y') : PM (x * x') (y * y')

theorem **PM.pow** {x y : R} (h : PM x y) (k : ℕ) : PM (x ^ k) (y ^ k)

theorem **PM.prod** {ι : Type} (S : Finset ι) {f g : ι → R} (h : ∀ i ∈ S, PM (f i) (g i)) : PM (∏ i ∈ S, f i) (∏ i ∈ S, g i)

theorem **jacobi_blocks** {α β : Type} [Fintype α] [Fintype β] [DecidableEq α] [DecidableEq β] (M : Matrix (α ⊕ β) (α ⊕ β) R) : M.toBlocks₁₁.det * M.adjugate.det = M.det ^ Fintype.card α * M.adjugate.toBlocks₂₂.det

theorem **lead_zero** {n : ℕ} (M : Matrix (Fin n) (Fin n) R) : lead M 0 = 1

theorem **lead_self** {n : ℕ} (M : Matrix (Fin n) (Fin n) R) : lead M n = M.det

theorem **jacobi_rev** [IsDomain R] {n k : ℕ} (h : k + 1 ≤ n) (M : Matrix (Fin n) (Fin n) R) (hM : M.det ≠ 0) (r c : Fin (k + 1) → Fin n) (hr : ∀ s, (r s : ℕ) = n - 1 - s) (hc : ∀ b, (c b : ℕ) = n - (k + 1) + b) : PM (Matrix.of fun s b => M.adjugate (r s) (c b)).det (M.det ^ k * lead M (n - 1 - k))

theorem **sum_mul_step** (B : ℕ → ℕ) (hB : Antitone B) (m : ℕ) : ∑ i ∈ Finset.range m, i * (B i - B (i + 1)) + B 0 + m * B m = ∑ i ∈ Finset.range (m + 1), B i

theorem **prod_steps** {n : ℕ} (L : ℕ → R) (hL : L 0 = 1) (A B : ℕ → ℕ) (hA : Antitone A) (hB : Antitone B) (hAn : A n = 0) (hBn : B n = 0) (s : ℤ) (lam : DomWeight n) (hw : ∀ i : Fin n, lam.w i = s + A i - B (n - 1 - i)) : (∏ i ∈ Finset.range n, L (i + 1) ^ (A i - A (i + 1))) * ∏ i ∈ Finset.range n, (L n ^ i * L (n - 1 - i)) ^ (B i - B (i + 1)) = L n ^ (((∑ i ∈ Finset.range n, B i : ℕ) : ℤ) + lam.detShift - s).toNat * ∏ i : Fin n, L (i + 1) ^ (lam.natW i - Formalism.GradedWeylDivisibility.nuNext lam.natW i)

theorem **cut_antitone** {a : ℕ → ℕ} (ha : Antitone a) (m : ℕ) : Antitone (cut a m)

theorem **prod_cut** {R : Type} [CommRing R] (f : ℕ → R) (a : ℕ → ℕ) {m n : ℕ} (h : m ≤ n) : ∏ i : Fin m, f i ^ (a i - if (i : ℕ) + 1 < m then a (i + 1) else 0) = ∏ i ∈ Finset.range n, f i ^ (cut a m i - cut a m (i + 1))

theorem **sum_cut** (a : ℕ → ℕ) {m n : ℕ} (h : m ≤ n) : ∑ i : Fin m, a i = ∑ i ∈ Finset.range n, cut a m i

theorem **gen_det_ne_zero** {Ix : Type} (K : Type) [Field K] (n : ℕ) (t : Ix) : (gen K n t).det ≠ 0

theorem **blockEmb_extremal** {Ix : Type} (K : Type) [Field K] {n : ℕ} (t : Ix) (lam : DomWeight n) : Formalism.GradedWeylDivisibility.blockEmb K t (extremal K n lam) = ∏ i : Fin n, lead (gen K n t) (i + 1) ^ (lam.natW i - Formalism.GradedWeylDivisibility.nuNext lam.natW i)

theorem **not_inPsi_conj** (h : IsEnum H pl) (t : Ix) : ¬ InPsi H (ComplexEmbedding.conjugate (pl t))

theorem **prod_types** {R : Type} [CommMonoid R] (h : IsEnum H pl) (f : (E →+* ℂ) → R) : ∏ σ, f σ = ∏ t, (f (pl t) * f (ComplexEmbedding.conjugate (pl t)))

theorem **cof_X_type** (h : IsEnum H pl) (R : Type) [CommRing R] (t : Ix) (i : Fin (Dw.mult (pl t))) (b : Fin n) (hi : (i : ℕ) < n) : cof Dw pl R n (X ⟨pl t, i, b⟩) = gen R n t b ⟨i, hi⟩

theorem **cof_X_conj** (h : IsEnum H pl) (R : Type) [CommRing R] (t : Ix) (i : Fin (Dw.mult (ComplexEmbedding.conjugate (pl t)))) (b : Fin n) (hi : (i : ℕ) < n) : cof Dw pl R n (X ⟨ComplexEmbedding.conjugate (pl t), i, b⟩) = (gen R n t).adjugate (Fin.rev ⟨i, hi⟩) b

theorem **mult_le** (hn : Dw.p + Dw.q ≤ n) (σ : E →+* ℂ) : Dw.mult σ ≤ n

theorem **cof_pG** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n), (∀ τ, Antitone fun i => a (τ, i)) → WtIs Dw pl a s lam → ∃ ε : ℤˣ, cof Dw pl c.F n (pG (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameReadingLeadingMinorProductExtremalStatement

theorem **_root_.GlobalTheory.T.frameReading_leadingMinorProduct_extremal** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n), (∀ τ, Antitone fun i => a (τ, i)) → WtIs Dw pl a s lam → ∃ ε : ℤˣ, cof Dw pl c.F n (pG (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameRigidificationOfCanonicalAxes

theorem **jetAug_comp_jetSubstP** {S A B : Type} [CommRing S] (m : ℕ) (φ : A → MvPolynomial B S) : (jetAug : Jet S B m →ₐ[S] S).comp (jetSubstP m φ) = (jetAug : Jet S A m →ₐ[S] S)

private theorem **homAt_comp_homAt_eq_homAt_comp** {X : Scheme.{0}} {S : Type} [CommRing S] (x0 : Sp S ⟶ X) {A B C : Type} [CommRing A] [Algebra S A] [CommRing B] [Algebra S B] [CommRing C] [Algebra S C] (f : A →ₐ[S] B) (g : B →ₐ[S] C) : Formalism.L2.Pj.homAt x0 g ≫ Formalism.L2.Pj.homAt x0 f = Formalism.L2.Pj.homAt x0 (g.comp f)

theorem **rig_of_frAxes** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∀ wP : PrimTyG t, IsPrimG t wP → Rig0G t x0 d x act e wP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FrameRigidificationOfCanonicalAxesStatement

theorem **_root_.GlobalTheory.T.frame_rigidification_of_canonical_axes** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (X0 : Scheme.{0}) (t : SAB H X0) (S : Type) [CommRing S] (x0 : Sp S ⟶ X0) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (κ : Type) [Fintype κ] (tg : κ → (Sp (DualNumber S) ⟶ t.G.left)) (uvec : κ → Fin d → S) (x : ∀ m, jOG x0 d m 1 ⟶ t.G), (∀ j, MvPowerSeries.constantCoeff (F j) = 0) → (∀ a j, MvPowerSeries.constantCoeff (act a j) = 0) → IsChartG t x0 d F act tg uvec x → ∀ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) → FrTrunc t x0 d x e → FrAxes t x0 d x e → ∀ wP : PrimTyG t, IsPrimG t wP → Rig0G t x0 d x act e wP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FramedPointBaseChangeClauses

theorem **framedPoint_bcGenR** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) : BcGenRSt Xo c

theorem **framedPoint_baseChange_clauses** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), BcLieAnySt Xo c ∧ BcLieRSt Xo c ∧ BcGenRSt Xo c ∧ FrBCFieldSt Xo c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FramedPointBaseChangeLaws

theorem **spec_comp** {S T U : Type} [CommRing S] [CommRing T] [CommRing U] (φ : S →+* T) (ψ : T →+* U) : spMap (ψ.comp φ) = spMap ψ ≫ spMap φ

theorem **dual_comp** {S T U : Type} [CommRing S] [CommRing T] [CommRing U] (φ : S →+* T) (ψ : T →+* U) : dualMap (ψ.comp φ) = (dualMap ψ).comp (dualMap φ)

theorem **spec_algHom_id** (K : Type) [Field K] (T : Type) [CommRing T] [Algebra K T] : spMap (AlgHom.id K T).toRingHom = 𝟙 _

theorem **dual_id** (T : Type) [CommRing T] : dualMap (RingHom.id T) = RingHom.id _

theorem **appLE_eq_of_eq** {X Y : Scheme.{0}} {f g : X ⟶ Y} (e : f = g) (U : Y.Opens) (V : X.Opens) (hf : V ≤ f ⁻¹ᵁ U) (hg : V ≤ g ⁻¹ᵁ U) : f.appLE U V hf = g.appLE U V hg

theorem **chart_le** {S T : Type} [CommRing S] [CommRing T] {x : Sp S ⟶ Xo.X} {x' : Sp T ⟶ Xo.X} (pm : Gx x' ⟶ Gx x) (hs : pm ≫ (snd (Over.mk x) Xo.univ.t.G).left = (snd (Over.mk x') Xo.univ.t.G).left) (i : Fin 1 → Xo.univ.t.PP.L.ι) : (PPx x').U i ≤ pm ⁻¹ᵁ (PPx x).U i

theorem **pulledGroup_hom_ext** {S : Type} [CommRing S] {x : Sp S ⟶ Xo.X} {W : Scheme.{0}} (a b : W ⟶ Gx x) (hs : a ≫ (snd (Over.mk x) Xo.univ.t.G).left = b ≫ (snd (Over.mk x) Xo.univ.t.G).left) (hf : a ≫ (fst (Over.mk x) Xo.univ.t.G).left = b ≫ (fst (Over.mk x) Xo.univ.t.G).left) : a = b

theorem **refl** {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) : FrMap (AlgHom.id c.F S) y y

theorem **trans** {S T U : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] [CommRing U] [Algebra c.F U] (φ : S →ₐ[c.F] T) (ψ : T →ₐ[c.F] U) {y : FrPt Xo c S} {z : FrPt Xo c T} {w : FrPt Xo c U} (h₁ : FrMap φ y z) (h₂ : FrMap ψ z w) : FrMap (ψ.comp φ) y w

theorem **cancel** {S T U : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] [CommRing U] [Algebra c.F U] (φ : S →ₐ[c.F] T) (κ : T →ₐ[c.F] U) {y0 : FrPt Xo c S} {y : FrPt Xo c T} {z : FrPt Xo c U} (h₁ : FrMap φ y0 y) (h₂ : FrMap (κ.comp φ) y0 z) : FrMap κ y z

theorem **unique** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {y : FrPt Xo c S} {z z' : FrPt Xo c T} (h₁ : FrMap φ y z) (h₂ : FrMap φ y z') : z = z'

theorem **isPullSec** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {y : FrPt Xo c S} {z : FrPt Xo c T} (hm : FrMap φ y z) (h : ptO Xo z.x ⟶ ptO Xo y.x) (hh : h.left = spMap φ.toRingHom) : IsPullSec Xo.univ.t.G Xo.univ.t.PP.L 1 (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) y.s z.s

theorem **framedPoint_baseChange_laws** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), (∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), FrMap (AlgHom.id c.F S) y y) ∧ (∀ (S T U : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] [CommRing U] [Algebra c.F U] (φ : S →ₐ[c.F] T) (ψ : T →ₐ[c.F] U) (y : FrPt Xo c S) (z : FrPt Xo c T) (w : FrPt Xo c U), FrMap φ y z → (FrMap ψ z w ↔ FrMap (ψ.comp φ) y w)) ∧ (∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y : FrPt Xo c S) (z z' : FrPt Xo c T), FrMap φ y z → FrMap φ y z' → z = z') ∧ ∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y : FrPt Xo c S) (z : FrPt Xo c T), FrMap φ y z → ∀ h : ptO Xo z.x ⟶ ptO Xo y.x, h.left = spMap φ.toRingHom → IsPullSec Xo.univ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FramedPointGeneratorRingBaseChange

theorem **isGenSec_pullS** (hS : SecBaseChangeSt) [IsLocallyNoetherian S] [IsAffine T.left] [IsAffine T'.left] (h : T' ⟶ T) {th : LB.Sec (LPow P.A P.L 1 T)} (H : IsGenSec P.A P.L 1 th) : IsGenSec P.A P.L 1 (pullS P.A 1 h th)

theorem **gs_eq_zero** {Y : Scheme.{0}} (f a : Γ(Y, ⊤)) (n : ℕ) (h : (Y.basicOpen f).ι.appTop.hom (a * f ^ n) = 0) (p : Y) (hp : p ∈ Y.basicOpen f) : gs (𝟙 Y) p ⊤ a = 0

theorem **basis_smul_eq_zero** {R M κ : Type} [CommRing R] [AddCommGroup M] [Module R M] (c : Module.Basis κ R M) (i : κ) (x : R) (h : x • c i = 0) : x = 0

theorem **eq_zero_of_smul** (hS : SecBaseChangeSt) [IsLocallyNoetherian S] [IsAffine T.left] {th : LB.Sec (LPow P.A P.L 1 T)} (H : IsGenSec P.A P.L 1 th) (a : Γ(T.left, ⊤)) (ha : a • (SecM.ofSec P.A th : SecM P.A P.L 1 T) = 0) : a = 0

theorem **existsUnique_of_isPullSec** (hS : SecBaseChangeSt) [IsLocallyNoetherian S] [IsAffine T.left] [IsAffine T'.left] (h : T' ⟶ T) {th : LB.Sec (LPow P.A P.L 1 T)} (H : IsGenSec P.A P.L 1 th) {th' : LB.Sec (LPow P.A P.L 1 T')} (hp : IsPullSec P.A P.L 1 (h ▷ P.A) (whiskerRight_snd _ _) th th') (s : LB.Sec (LPow P.A P.L 1 T')) : ∃! r : Γ(T'.left, ⊤), ∀ j, s.f j = liftFun P.A _ r * th'.f j

theorem **toΓ_bijective** (R : Type) [CommRing R] : Function.Bijective (toΓ R)

theorem **framedPoint_generator_baseChange** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] (ψ : S →+* T) (y : FrPt Xo c S) (h : ptO Xo (spMap ψ ≫ y.x) ⟶ ptO Xo y.x), h.left = spMap ψ → ∀ s' : LB.Sec (PPx (spMap ψ ≫ y.x)), IsPullSec Xo.univ.t.G Xo.univ.t.PP.L 1 (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) y.s s' → ∀ s'' : LB.Sec (PPx (spMap ψ ≫ y.x)), ∃! a : T, ∀ i, s''.f i = scal (spMap ψ ≫ y.x) a i * s'.f i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FramedPointGeneratorRingBaseChangeStatement

theorem **_root_.GlobalTheory.T.framedPoint_generator_ringBaseChange** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] (ψ : S →+* T) (y : FrPt Xo c S) (h : ptO Xo (spMap ψ ≫ y.x) ⟶ ptO Xo y.x), h.left = spMap ψ → ∀ s' : LB.Sec (PPx (spMap ψ ≫ y.x)), IsPullSec Xo.univ.t.G Xo.univ.t.PP.L 1 (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) y.s s' → ∀ s'' : LB.Sec (PPx (spMap ψ ≫ y.x)), ∃! a : T, ∀ i, s''.f i = scal (spMap ψ ≫ y.x) a i * s'.f i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FramedPointsLiftToLevelCover

theorem **mem_ch** {L : LB A} {q : Y ⟶ A} {i : L.ι} {y : Y} : y ∈ ch L q i ↔ q y ∈ L.U i

theorem **exists_const** {I : Type} (j : Fin 1 → I) : ∃ i, j = fun _ => i

theorem **sv_trans** {L : LB A} {q : Y ⟶ A} (s : LB.Sec (LOne L q)) {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) {i i' : L.ι} (h : q (r z) ∈ L.U i) (h' : q (r z) ∈ L.U i') : sv s r z i = (gm (r ≫ q) z (L.U i ⊓ L.U i') (L.g i i')).val * sv s r z i'

theorem **exists_move** {L : LB A} {L' : LB A₂} (hL : L.IsLB) (hL' : L'.IsLB) {q : Y ⟶ A} {q' : Y' ⟶ A₂} (h : Y' ⟶ Y) (c : ∀ (i : L.ι) (j : L'.ι), Γ(Y', ch L' q' j ⊓ h ⁻¹ᵁ ch L q i)ˣ) (hc : ∀ (y : Y') i i' j j', q (h y) ∈ L.U i ↠ q (h y) ∈ L.U i' ↠ q' y ∈ L'.U j ↠ q' y ∈ L'.U j' ↠ gm (𝟙 Y') y _ (c i j) ∗ gm (h ≫ q) y (L.U i ⊓ L.U i') (L.g i i') = gm q' y (L'.U j ⊓ L'.U j') (L'.g j j') ∗ gm (𝟙 Y') y _ (c i' j')) (s : LB.Sec (LOne L q)) : ∃ s' : LB.Sec (LOne L' q'), ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) i j, q (h (r z)) ∈ L.U i ↠ q' (r z) ∈ L'.U j ↠ sv s' r z j = (gm r z _ (c i j)).val ∗ sv s (r ≫ h) z i

theorem **pt_eq** {Z : Scheme.{0}} {f g : Z ⟶ A} (e : f = g) (z : Z) : f z = g z

theorem **exists_down** (hL : L.IsLB) (hL₂ : L₂.IsLB) {w : LB.Iso L₂ (L.pull φ)} (hw : w.IsGen) {q₂ : Y₂ ⟶ A₂} {q : Y ⟶ A} {p : Y₂ ⟶ Y} (hp : p ≫ q = q₂ ≫ φ) (s : LB.Sec (LOne L q)) : ∃ s₂ : LB.Sec (LOne L₂ q₂), Rel w p s₂ s

theorem **exists_up** (hL : L.IsLB) (hL₂ : L₂.IsLB) {w : LB.Iso L₂ (L.pull φ)} (hw : w.IsGen) {q₂ : Y₂ ⟶ A₂} {q : Y ⟶ A} {p : Y₂ ⟶ Y} (hp : p ≫ q = q₂ ≫ φ) {n : Y ⟶ Y₂} (hn : n ≫ p = 𝟙 _) (hn' : p ≫ n = 𝟙 _) (s₂ : LB.Sec (LOne L₂ q₂)) : ∃ s : LB.Sec (LOne L q), Rel w p s₂ s

theorem **gm_linCombIso_pull** (w : LB.Iso L₂ (L.pull φ)) {K : Type} [Fintype K] (φ' : K ↠ (Y₂ ⟶ A₂)) (ε : K ↠ ℤ) {Z : Scheme.{0}} (r : Z ⟶ Y₂) (z : Z) (p : K ↠ L.ι) (q : K ↠ L₂.ι) (hm : r z ∈ LB.wchart (L.pull φ) φ' (fun k _ => p k) ⊓ LB.wchart L₂ φ' q) : gm r z (LB.wchart (L.pull φ) φ' (fun k _ => p k) ⊓ LB.wchart L₂ φ' q) ((LB.linCombIso w φ' ε).s (fun k _ => p k, q)) = ∏ k, gw w (r ≫ φ' k) z (p k) (q k) ^ ε k

theorem **rel_iff** (w : LB.Iso L₂ (L.pull φ)) {q₂ : Y₂ ⟶ A₂} {q : Y ⟶ A} {p : Y₂ ⟶ Y} (hp : p ≫ q = q₂ ≫ φ) (s₂ : LB.Sec (LOne L₂ q₂)) (s : LB.Sec (LOne L q)) : (∃ t : LB.Sec (LOne (L.pull φ) q₂), SecMaps (LB.linCombIso w (K

theorem **sv_of_pull** {q : Y ⟶ A} {q' : Y' ⟶ A} {p : Y' ⟶ Y} (hp : p ≫ q = q') {s : LB.Sec (LOne L q)} {s' : LB.Sec (LOne L q')} (H : ∀ (i : Fin 1 ↠ L.ι) (h : (LOne L q').U i ≤ p ⁻¹ᵁ (LOne L q).U i), (p.appLE _ _ h).hom (s.f i) = s'.f i) {Z : Scheme.{0}} (r : Z ⟶ Y') (z : Z) (i : L.ι) (hi : q' (r z) ∈ L.U i) : sv s' r z i = sv s (r ≫ p) z i

theorem **gs_lift** (b : Y ⟶ B) (a : Γ(B, ⊤)) (W : Y.Opens) {Z : Scheme.{0}} (r : Z ⟶ Y) (z : Z) (hz : r z ∈ W) : gs r z W ((Y.presheaf.map (homOfLE (le_top : W ≤ ⊤)).op).hom (b.appTop.hom a)) = gs (r ≫ b) z ⊤ a

theorem **pull_scal_iff** {q : Y ⟶ A} {q' : Y' ⟶ A} {p : Y' ⟶ Y} (hp : p ≫ q = q') (b : Y ⟶ B) (a : Γ(B, ⊤)) (s : LB.Sec (LOne L q)) (s' : LB.Sec (LOne L q')) : (∀ (i : Fin 1 ↠ L.ι) (h : (LOne L q').U i ≤ p ⁻¹ᵁ (LOne L q).U i), (p.appLE _ _ h).hom ((Y.presheaf.map (homOfLE (le_top : (LOne L q).U i ≤ ⊤)).op).hom (b.appTop.hom a) ∗ s.f i) = s'.f i) ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) (i : L.ι), q' (r z) ∈ L.U i ↠ sv s' r z i = gs ((r ≫ p) ≫ b) z ⊤ a ∗ sv s (r ≫ p) z i

theorem **scal_iff** {q : Y ⟶ A} (b : Y ⟶ B) (a : Γ(B, ⊤)) (s s' : LB.Sec (LOne L q)) : (∀ i : Fin 1 ↠ L.ι, s'.f i = (Y.presheaf.map (homOfLE (le_top : (LOne L q).U i ≤ ⊤)).op).hom (b.appTop.hom a) ∗ s.f i) ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (i : L.ι), q (r z) ∈ L.U i ↠ sv s' r z i = gs (r ≫ b) z ⊤ a ∗ sv s r z i

theorem **rel_scal_iff** (hL : L.IsLB) (hL₂ : L₂.IsLB) {w : LB.Iso L₂ (L.pull φ)} {q₂ : Y₂ ⟶ A₂} {q : Y ⟶ A} {p : Y₂ ⟶ Y} (hp : p ≫ q = q₂ ≫ φ) {n : Y ⟶ Y₂} (hn : n ≫ p = 𝟙 _) {b : Y ⟶ B} {b₂ : Y₂ ⟶ B} (hb : p ≫ b = b₂) (a : Γ(B, ⊤)) {s₂ s₂' : LB.Sec (LOne L₂ q₂)} {s s' : LB.Sec (LOne L q)} (H : Rel w p s₂ s) (H' : Rel w p s₂' s') : (∀ (Z : Scheme.{0}) (r : Z ⟶ Y₂) (z : Z) (j : L₂.ι), q₂ (r z) ∈ L₂.U j ↠ sv s₂' r z j = gs (r ≫ b₂) z ⊤ a ∗ sv s₂ r z j) ↔ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) (i : L.ι), q (r z) ∈ L.U i ↠ sv s' r z i = gs (r ≫ b) z ⊤ a ∗ sv s r z i

theorem **exists_push** (h : IsGrpPullback G₂ G f φ) {a₂ : Z ⟶ X₂} {a : Z ⟶ X} (e : a₂ ≫ f = a) : ∃ ρ : (Over.mk a₂ ⟶ G₂) →∗ (Over.mk a ⟶ G), ∀ u, (ρ u).left = u.left ≫ φ

theorem **one_push** (h : IsGrpPullback G₂ G f φ) {a₂ : Z ⟶ X₂} {a : Z ⟶ X} (e : a₂ ≫ f = a) : (1 : Over.mk a₂ ⟶ G₂).left ≫ φ = (1 : Over.mk a ⟶ G).left

theorem **prod_push** (h : IsGrpPullback G₂ G f φ) {a₂ : Z ⟶ X₂} {a : Z ⟶ X} (e : a₂ ≫ f = a) {m : ℕ} (u₂ : Fin m ↠ (Over.mk a₂ ⟶ G₂)) (u : Fin m ↠ (Over.mk a ⟶ G)) (hu : ∀ i, (u i).left = (u₂ i).left ≫ φ) : (List.ofFn u₂).prod.left ≫ φ = (List.ofFn u).prod.left

theorem **exists_pm** {W W' : Scheme.{0}} {x : W ⟶ X} (a : W' ⟶ W) (g : W' ⟶ G.left) (e : a ≫ x = g ≫ G.hom) : ∃ p : W' ⟶ (Over.mk x ⊗ G).left, p ≫ (snd (Over.mk x) G).left = g ∧ p ≫ (fst (Over.mk x) G).left = a

theorem **pm_ext** {W W' : Scheme.{0}} {x : W ⟶ X} {p p' : W' ⟶ (Over.mk x ⊗ G).left} (h₁ : p ≫ (snd (Over.mk x) G).left = p' ≫ (snd (Over.mk x) G).left) (h₂ : p ≫ (fst (Over.mk x) G).left = p' ≫ (fst (Over.mk x) G).left) : p = p'

theorem **pm_cond** {W : Scheme.{0}} (x : W ⟶ X) : (fst (Over.mk x) G).left ≫ x = (snd (Over.mk x) G).left ≫ G.hom

theorem **eps_alg** (a : S) : spMap (epsMul a) ≫ spMap (algebraMap S (DualNumber S)) = spMap (algebraMap S (DualNumber S))

theorem **fst_alg** : spMap (TrivSqZeroExt.fstHom S S S).toRingHom ≫ spMap (algebraMap S (DualNumber S)) = 𝟙 _

theorem **tan_iff** (e : x₂ ≫ r = x) {v : Sp (DualNumber S) ⟶ Xo₂.univ.t.G.left} (hv : v ≫ Xo₂.univ.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ x₂) : IsTanAt x₂ v ↔ IsTanAt x (v ≫ m.φ)

theorem **eig_iff** {c : Coeff l} {c₂ : Coeff l₂} [Algebra c.F S] [Algebra c₂.F S] (τ : E →+* ℂ) (hc : ∀ a, algebraMap c₂.F S (embK c₂ τ a) = algebraMap c.F S (embK c τ a)) {v : Sp (DualNumber S) ⟶ Xo₂.univ.t.G.left} (hv : v ≫ Xo₂.univ.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ x₂) : IsEigAt c₂ τ v ↔ IsEigAt c τ (v ≫ m.φ)

theorem **comb_iff** (e : x₂ ≫ r = x) {k : ℕ} {v : Fin k ↠ (Sp (DualNumber S) ⟶ Xo₂.univ.t.G.left)} (hv : ∀ i, v i ≫ Xo₂.univ.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ x₂) (cf : Fin k ↠ S) {w : Sp (DualNumber S) ⟶ Xo₂.univ.t.G.left} (hw : w ≫ Xo₂.univ.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ x₂) : IsCombAt x₂ v cf w ↔ IsCombAt x (fun i => v i ≫ m.φ) cf (w ≫ m.φ)

theorem **exists_lift** (e : x₂ ≫ r = x) {v : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (hv : v ≫ Xo.univ.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ x) : ∃ v₂ : Sp (DualNumber S) ⟶ Xo₂.univ.t.G.left, v₂ ≫ m.φ = v ∧ v₂ ≫ Xo₂.univ.t.G.hom = spMap (algebraMap S (DualNumber S)) ≫ x₂

theorem **exists_pm_iso** (e : x₂ ≫ r = x) : ∃ (p : Gx x₂ ⟶ Gx x) (n : Gx x ⟶ Gx x₂), p ≫ (snd (Over.mk x) Xo.univ.t.G).left = (snd (Over.mk x₂) Xo₂.univ.t.G).left ≫ m.φ ∧ p ≫ (fst (Over.mk x) Xo.univ.t.G).left = (fst (Over.mk x₂) Xo₂.univ.t.G).left ∧ n ≫ p = 𝟙 _ ∧ p ≫ n = 𝟙 _

theorem **frOver_iff** [Algebra c.F S] [Algebra d.c₂.F S] (y₂ : FrPt d.Xo₂ d.c₂ S) (y : FrPt Xo c S) : FrOver d y₂ y ↔ y₂.x ≫ d.r = y.x ∧ (∀ τ i, y₂.u τ i ≫ d.m.φ = y.u τ i) ∧ ∃ (w : LB.Iso d.Xo₂.univ.t.PP.L (Xo.univ.t.PP.L.pull d.m.φ)) (p : Gx y₂.x ⟶ Gx y.x), IsRig d w ∧ p ≫ (snd (Over.mk y.x) Xo.univ.t.G).left = (snd (Over.mk y₂.x) d.Xo₂.univ.t.G).left ≫ d.m.φ ∧ p ≫ (fst (Over.mk y.x) Xo.univ.t.G).left = (fst (Over.mk y₂.x) d.Xo₂.univ.t.G).left ∧ Rel (q₂

theorem **embK_eF** (τ : E →+* ℂ) (a : 𝒪 E) : d.eF (embK c τ a) = embK d.c₂ τ a

theorem **exists_frOver** [Algebra c.F S] [Algebra d.c₂.F S] (hc : ∀ a : c.F, algebraMap d.c₂.F S (d.eF a) = algebraMap c.F S a) (y : FrPt Xo c S) (x₂ : Sp S ⟶ d.Xo₂.X) (e : x₂ ≫ d.r = y.x) (hx : x₂ ≫ d.Xo₂.s = sOf d.c₂ S) : ∃ y₂ : FrPt d.Xo₂ d.c₂ S, y₂.x = x₂ ∧ FrOver d y₂ y

theorem **frOver_map** [Algebra c.F S] [Algebra d.c₂.F S] {T : Type} [CommRing T] [Algebra c.F T] [Algebra d.c₂.F T] {ψ : S →ₐ[c.F] T} {ψ₂ : S →ₐ[d.c₂.F] T} (hψ : ψ₂.toRingHom = ψ.toRingHom) {y : FrPt Xo c S} {y₂ : FrPt d.Xo₂ d.c₂ S} {z : FrPt Xo c T} {z₂ : FrPt d.Xo₂ d.c₂ T} (h : FrOver d y₂ y) (hz : FrMap ψ y z) (hz₂ : FrMap ψ₂ y₂ z₂) : FrOver d z₂ z

theorem **frBor_lift** (hG : GlobalFnsSt) [Algebra c.F S] [Algebra d.c₂.F S] {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} {y₂ y₂' : FrPt d.Xo₂ d.c₂ S} (h : FrOver d y₂ y) (h' : FrOver d y₂' y') (ex : y₂'.x = y₂.x) (hb : FrBor t y y') : FrBor t y₂ y₂'

theorem **frames_lift_to_cover** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), (∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra d.c₂.F S], (∀ a : c.F, algebraMap d.c₂.F S (d.eF a) = algebraMap c.F S a) → ∀ (y : FrPt Xo c S) (x₂ : Sp S ⟶ d.Xo₂.X), x₂ ≫ d.r = y.x → x₂ ≫ d.Xo₂.s = sOf d.c₂ S → ∃ y₂ : FrPt d.Xo₂ d.c₂ S, y₂.x = x₂ ∧ FrOver d y₂ y) ∧ (∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra d.c₂.F S], (∀ a : c.F, algebraMap d.c₂.F S (d.eF a) = algebraMap c.F S a) → ∀ (T : Type) [CommRing T] [Algebra c.F T] [Algebra d.c₂.F T], (∀ a : c.F, algebraMap d.c₂.F T (d.eF a) = algebraMap c.F T a) → ∀ (ψ : S →ₐ[c.F] T) (ψ₂ : S →ₐ[d.c₂.F] T), ψ₂.toRingHom = ψ.toRingHom → ∀ (y : FrPt Xo c S) (y₂ : FrPt d.Xo₂ d.c₂ S) (z : FrPt Xo c T) (z₂ : FrPt d.Xo₂ d.c₂ T), FrOver d y₂ y →

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FramedPointsLiftToLevelCoverStatement

theorem **_root_.GlobalTheory.T.framedPoints_lift_to_levelCover** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), (∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra d.c₂.F S], (∀ a : c.F, algebraMap d.c₂.F S (d.eF a) = algebraMap c.F S a) → ∀ (y : FrPt Xo c S) (x₂ : Sp S ⟶ d.Xo₂.X), x₂ ≫ d.r = y.x → x₂ ≫ d.Xo₂.s = sOf d.c₂ S → ∃ y₂ : FrPt d.Xo₂ d.c₂ S, y₂.x = x₂ ∧ FrOver d y₂ y) ∧ (∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra d.c₂.F S], (∀ a : c.F, algebraMap d.c₂.F S (d.eF a) = algebraMap c.F S a) → ∀ (T : Type) [CommRing T] [Algebra c.F T] [Algebra d.c₂.F T], (∀ a : c.F, algebraMap d.c₂.F T (d.eF a) = algebraMap c.F T a) → ∀ (ψ : S →ₐ[c.F] T) (ψ₂ : S →ₐ[d.c₂.F] T), ψ₂.toRingHom = ψ.toRingHom → ∀ (y : FrPt Xo c S) (y₂ : FrPt d.Xo₂ d.c₂ S) (z : FrPt Xo c T) (z₂ : FrP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FullLinearLevelCoverBaseChangeTorsor

theorem **mem_iff_modN** (k : ℕ) (x : Dw.La) : x ∈ (Ideal.span {(k : 𝒪 E)} • ⊤ : Submodule (𝒪 E) Dw.La) ↔ Dw.ModN k x

theorem **mk_eq_iff** (k : ℕ) (a b : Dw.La) : (Submodule.Quotient.mk a : LatMod Dw k) = Submodule.Quotient.mk b ↔ Dw.ModN k (a − b)

theorem **finite_latMod** (k : ℕ) (hk : 0 < k) : Finite (LatMod Dw k)

theorem **nsmul_latMod** (k : ℕ) (q : LatMod Dw k) : k • q = 0

theorem **exists_descend** (k : ℕ) {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) (ev : Dw.La → (ptO Xo x0 ⟶ Xo.univ.t.G)) (hadd : ∀ a b, ev (a + b) = ev a ∗ ev b) (hlin : ∀ (o : 𝒪 E) a, ev (o • a) = ev a ≫ Xo.univ.t.act.act o) (hker : ∀ a, ev (k • a) = 1) (hinj : ∀ (K : Type) [Field K] (χ : S →+∗ K) (a b : Dw.La), spMap χ ≫ (ev a).left = spMap χ ≫ (ev b).left → Dw.ModN k (a − b)) : ∃ ev' : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G), IsFullLev Xo k x0 ev' ∧ ∀ a, ev' (Submodule.Quotient.mk a) = ev a

theorem **pow_eq_one** {k : ℕ} {S : Type} [CommRing S] {x0 : Sp S ⟶ Xo.X} {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (h : IsFullLev Xo k x0 ev) (q : LatMod Dw k) : ev q ^ k = 1

theorem **baseChange** {k : ℕ} {S T : Type} [CommRing S] [CommRing T] (ψ : S →+∗ T) {x0 : Sp S ⟶ Xo.X} {x1 : Sp T ⟶ Xo.X} (h : ptO Xo x1 ⟶ ptO Xo x0) (hh : h.left = spMap ψ) {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (hev : IsFullLev Xo k x0 ev) : IsFullLev Xo k x1 (fun q => h ≫ ev q)

theorem **eq_of_agree** (hL : Laws) {k : ℕ} (hk : 0 < k) {S T : Type} [CommRing S] [CommRing T] (χ : S →+∗ T) (hχ : ∀ e, χ e = 1 → IsUnit e) {x0 : Sp S ⟶ Xo.X} {a b : ptO Xo x0 ⟶ Xo.univ.t.G} (ha : a ^ k = 1) (hb : b ^ k = 1) (hab : spMap χ ≫ a.left = spMap χ ≫ b.left) : a = b

theorem **exists_geometric** (hL : Laws) {k : ℕ} (hk : 0 < k) (K : Type) [Field K] [IsAlgClosed K] (x0 : Sp K ⟶ Xo.X) : ∃ ev, IsFullLev Xo k x0 ev ∧ ∀ p : ptO Xo x0 ⟶ Xo.univ.t.G, p ^ k = 1 → ∃ q, ev q = p

theorem **onto_of_local** (hL : Laws) {k : ℕ} (hk : 0 < k) {T : Type} [CommRing T] [IsLocalRing T] {x0 : Sp T ⟶ Xo.X} {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (hev : IsFullLev Xo k x0 ev) (p : ptO Xo x0 ⟶ Xo.univ.t.G) (hp : p ^ k = 1) : ∃ q, ev q = p

theorem **injective_of_local** {k : ℕ} {T : Type} [CommRing T] [IsLocalRing T] {x0 : Sp T ⟶ Xo.X} {ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (hev : IsFullLev Xo k x0 ev) : Function.Injective ev

theorem **exists_levGrp** (hL : Laws) {k : ℕ} (hk : 0 < k) {T : Type} [CommRing T] [IsLocalRing T] {x0 : Sp T ⟶ Xo.X} {ev ev' : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)} (hev : IsFullLev Xo k x0 ev) (hev' : IsFullLev Xo k x0 ev') : ∃ g : LevGrp Dw k, ∀ q, ev' q = ev (g q)

theorem **fullLevel_cover_baseChange_torsor** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (k : ℕ), 0 < k → (∀ (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X), ∃ (B : Type) (_ : CommRing B) (φ : S →+∗ B) (ev : LatMod Dw k → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ IsFullLev Xo k (spMap φ ≫ x0) ev) ∧ (∀ (S T : Type) [CommRing S] [CommRing T] (ψ : S →+∗ T) (x0 : Sp S ⟶ Xo.X) (x1 : Sp T ⟶ Xo.X) (h : ptO Xo x1 ⟶ ptO Xo x0), h.left = spMap ψ → ∀ ev, IsFullLev Xo k x0 ev → IsFullLev Xo k x1 (fun q => h ≫ ev q) ∧ ∀ q, ev q ^ k = 1) ∧ (∀ (T : Type) [CommRing T] [IsLocalRing T] (x0 : Sp T ⟶ Xo.X) (ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)), IsFullLev Xo k x0 ev → (∀ p : ptO Xo x0 ⟶ Xo.univ.t.G, p ^ k = 1 → ∃ q, ev q = p) ∧ ∀ ev', IsFullLev Xo k x0 ev' → ∃ g : LevGrp Dw k, ∀ q, ev' q = ev (g q)) ∧ ∀ (K : Type

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.FullLinearLevelCoverBaseChangeTorsorStatement

theorem **_root_.GlobalTheory.T.fullLinearLevel_cover_baseChange_torsor** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (k : ℕ), 0 < k → (∀ (S : Type) [CommRing S] (x0 : Sp S ⟶ Xo.X), ∃ (B : Type) (_ : CommRing B) (φ : S →+∗ B) (ev : LatMod Dw k → (ptO Xo (spMap φ ≫ x0) ⟶ Xo.univ.t.G)), φ.FaithfullyFlat ∧ IsFullLev Xo k (spMap φ ≫ x0) ev) ∧ (∀ (S T : Type) [CommRing S] [CommRing T] (ψ : S →+∗ T) (x0 : Sp S ⟶ Xo.X) (x1 : Sp T ⟶ Xo.X) (h : ptO Xo x1 ⟶ ptO Xo x0), h.left = spMap ψ → ∀ ev, IsFullLev Xo k x0 ev → IsFullLev Xo k x1 (fun q => h ≫ ev q) ∧ ∀ q, ev q ^ k = 1) ∧ (∀ (T : Type) [CommRing T] [IsLocalRing T] (x0 : Sp T ⟶ Xo.X) (ev : LatMod Dw k → (ptO Xo x0 ⟶ Xo.univ.t.G)), IsFullLev Xo k x0 ev → (∀ p : ptO Xo x0 ⟶ Xo.univ.t.G, p ^ k = 1 → ∃ q, ev q = p) ∧ ∀ ev', IsFullLev Xo k x0 ev' → ∃ g : LevGrp Dw k, ∀ q, ev'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GamAtOfLawsCore

theorem **gamAt_of_lawsCore** : ∀ (D : D7) (T : Core D), LawsCore T → GamAt T

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GeneratorReadingCorrMulInv

theorem **chart_iff** {X Z : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Z ⟶ X)) (j : K → L.ι) (x : Sp ℂ ⟶ Z) : ⊤ ≤ x ⁻¹ᵁ LB.wchart L φ j ↔ ∀ k, ⊤ ≤ (x ≫ φ k) ⁻¹ᵁ L.U (j k)

theorem **exists_chart** {X : Scheme.{0}} (L : LB X) (hL : L.IsLB) (x : Sp ℂ ⟶ X) : ∃ i, ⊤ ≤ x ⁻¹ᵁ L.U i

theorem **corr_val** {X Y : Scheme.{0}} (L : LB X) {K K' : Type} [Fintype K] [Fintype K'] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (φ' : K' → (Y ⟶ X)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (t' : LB.Gen (LB.LinComb L φ' ε')) (hc : LB.Gen.Corr L φ ε φ' ε' t t') (J : (Y ⟶ X) → L.ι) (p : Sp ℂ ⟶ Y) (hJ : ∀ ψ, ⊤ ≤ (p ≫ ψ) ^* L.U (J ψ)) : valAt' p (LB.wchart L φ fun k => J (φ k)) (t.s fun k => J (φ k)).val = valAt' p (LB.wchart L φ' fun k => J (φ' k)) (t'.s fun k => J (φ' k)).val

theorem **val_inv** {Z : Scheme.{0}} (p : Sp ℂ ⟶ Z) {V : Z.Opens} (h : ⊤ ≤ p ^* V) (u : Γ(Z, V)ˣ) : valAt' p V (u⁻¹).val = (valAt' p V u.val)⁻¹

theorem **prod_fibre** {T K : Type} [Fintype K] (φ : K → T) (ε : K → ℤ) (f : T → ℂ) (s : Finset T) (hs : ∀ k, φ k ∈ s) (hf : ∀ ψ ∈ s, f ψ ≠ 0) : ∏ k, f (φ k) ^ ε k = ∏ ψ ∈ s, f ψ ^ (∑ k, if φ k = ψ then ε k else 0)

theorem **genRd_corr_mul_inv** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K K' : Type) [Fintype K] [Fintype K'] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (φ' : K' → (Z ⟶ A.left)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)) (t' : LB.Gen (LB.LinComb L.L φ' ε')) (q : Sp ℂ ⟶ Z) (p : K → Fin g → ℂ) (p' : K' → Fin g → ℂ) (v v' : ℂ), Φ.GenRd φ ε t q p v → Φ.GenRd φ' ε' t' q p' v' → (LB.Gen.Corr L.L φ ε φ' ε' t t' → ∃ J : (Z ⟶ A.left) → L.L.ι, (∀ ψ, ⊤ ≤ (q ≫ ψ) ^* L.L.U (J ψ)) ∧ v * ∏ k, Φ.fr (J (φ k)) (p k) ^ ε k = v' * ∏ k, Φ.fr (J (φ' k)) (p' k) ^ ε' k) ∧ (LB.Gen.Corr L.L φ ε φ' ε' t t' → LB.SameSum φ ε φ' ε' → (∃ π : (Z ⟶ A.left) → Fin g → ℂ, (∀ k, p k = π (φ k)) ∧ ∀ k, p' k = π (φ' k)) → v = v') ∧ Φ.GenRd (Sum.elim φ φ') (Sum.elim ε ε') (LB.mulGen L.L φ ε φ' ε' t t') q (Sum.elim p p') (v * v') ∧ Φ.Ge

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GeneratorReadingCorrMulInvStatement

theorem **_root_.GlobalTheory.T.generatorReading_corr_mul_inv** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K K' : Type) [Fintype K] [Fintype K'] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (φ' : K' → (Z ⟶ A.left)) (ε' : K' → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)) (t' : LB.Gen (LB.LinComb L.L φ' ε')) (q : Sp ℂ ⟶ Z) (p : K → Fin g → ℂ) (p' : K' → Fin g → ℂ) (v v' : ℂ), Φ.GenRd φ ε t q p v → Φ.GenRd φ' ε' t' q p' v' → (LB.Gen.Corr L.L φ ε φ' ε' t t' → ∃ J : (Z ⟶ A.left) → L.L.ι, (∀ ψ, ⊤ ≤ (q ≫ ψ) ^* L.L.U (J ψ)) ∧ v * ∏ k, Φ.fr (J (φ k)) (p k) ^ ε k = v' * ∏ k, Φ.fr (J (φ' k)) (p' k) ^ ε' k) ∧ (LB.Gen.Corr L.L φ ε φ' ε' t t' → LB.SameSum φ ε φ' ε' → (∃ π : (Z ⟶ A.left) → Fin g → ℂ, (∀ k, p k = π (φ k)) ∧ ∀ k, p' k = π (φ' k)) → v = v') ∧ Φ.GenRd (Sum.elim φ φ') (Sum.elim ε ε') (LB.mulGen L.L φ ε φ' ε' t t')

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GeneratorReadingPullbackPeriodRigid

theorem **AnFr.genMove_e_zero** : Φ.e 0 = 1

theorem **AnFr.genMove_e_zero_left** : (Φ.e 0).left = x0 ≫ eSec A

theorem **AnFr.genMove_e_per** (z a : Fin g → ℂ) (ha : a ∈ Φ.lat) : Φ.e (z + a) = Φ.e z

theorem **genMove_chart** {X Z : Scheme.{0}} (M : LB X) {K : Type} [Fintype K] (φ : K → (Z ⟶ X)) (q : Sp ℂ ⟶ Z) (j : K → M.ι) (h : ⊤ ≤ q ^* LB.wchart M φ j) (k : K) : ⊤ ≤ (q ≫ φ k) ^* M.U (j k)

theorem **AnFr.genRd_pullGen_iff** {Z : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)) {Z' : Scheme.{0}} (h : Z' ⟶ Z) (q : Sp ℂ ⟶ Z') (p : K → Fin g → ℂ) (v : ℂ) : Φ.GenRd (fun k => h ≫ φ k) ε (LB.pullGen L.L φ ε t h) q p v ↔ Φ.GenRd φ ε t (q ≫ h) p v

theorem **AnFr.genRd_period** {Z : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)) (q : Sp ℂ ⟶ Z) (p a : K → Fin g → ℂ) (v : ℂ) (ha : ∀ k, a k ∈ Φ.lat) (H : Φ.GenRd φ ε t q p v) : Φ.GenRd φ ε t q (fun k => p k + a k) (v * ∏ k, Φ.fac (a k) (p k) ^ ε k)

theorem **AnFr.genRd_rigPow** (k : ℤ) : Φ.GenRd (K

theorem **genRd_pull_period_rig** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)), (∀ (Z' : Scheme.{0}) (h : Z' ⟶ Z) (q : Sp ℂ ⟶ Z') (p : K → Fin g → ℂ) (v : ℂ), Φ.GenRd (fun k => h ≫ φ k) ε (LB.pullGen L.L φ ε t h) q p v ↔ Φ.GenRd φ ε t (q ≫ h) p v) ∧ (∀ (q : Sp ℂ ⟶ Z) (p a : K → Fin g → ℂ) (v : ℂ), (∀ k, a k ∈ Φ.lat) → Φ.GenRd φ ε t q p v → Φ.GenRd φ ε t q (fun k => p k + a k) (v * ∏ k, Φ.fac (a k) (p k) ^ ε k)) ∧ ∀ k : ℤ, Φ.GenRd (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GeneratorReadingPullbackPeriodRigidStatement

theorem **_root_.GlobalTheory.T.generatorReading_pullback_period_rigid** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (Z : Scheme.{0}) (K : Type) [Fintype K] (φ : K → (Z ⟶ A.left)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L.L φ ε)), (∀ (Z' : Scheme.{0}) (h : Z' ⟶ Z) (q : Sp ℂ ⟶ Z') (p : K → Fin g → ℂ) (v : ℂ), Φ.GenRd (fun k => h ≫ φ k) ε (LB.pullGen L.L φ ε t h) q p v ↔ Φ.GenRd φ ε t (q ≫ h) p v) ∧ (∀ (q : Sp ℂ ⟶ Z) (p a : K → Fin g → ℂ) (v : ℂ), (∀ k, a k ∈ Φ.lat) → Φ.GenRd φ ε t q p v → Φ.GenRd φ ε t q (fun k => p k + a k) (v * ∏ k, Φ.fac (a k) (p k) ^ ε k)) ∧ ∀ k : ℤ, Φ.GenRd (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GenericSectionsPolynomialDictionary

theorem **ofWt_C** {ι : Type} (u : WeightFamilies.Wt D j) (F : B.Cusp j u) : OfWt X u (MvPolynomial.C (X.inc u F) : MvPolynomial ι X.Forms)

theorem **ofWt_sum** {ι κ : Type} (u : WeightFamilies.Wt D j) (s : Finset κ) (p : κ → MvPolynomial ι X.Forms) (h : ∀ k ∈ s, OfWt X u (p k)) : OfWt X u (∑ k ∈ s, p k)

theorem **ofWt_X_mul** {ι : Type} (u : WeightFamilies.Wt D j) (i : ι) (p : MvPolynomial ι X.Forms) (h : OfWt X u p) : OfWt X u (MvPolynomial.X i * p)

theorem **ofWt_mul** {ι : Type} (u u' : WeightFamilies.Wt D j) (p q : MvPolynomial ι X.Forms) (hp : OfWt X u p) (hq : OfWt X u' q) : OfWt X (wadd u u') (p * q)

theorem **toRp_C** (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (F : X.Forms) : toRp X la N (MvPolynomial.C F) = emb X la N F

theorem **toRp_X** (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (p : Idx D j N) : toRp X la N (MvPolynomial.X p) = algebraMap (Op X la N) (Rp X la N) (Tv X la N p)

theorem **toFn_C** (R : Rdg X) {N : ℕ} (t : Idx D j N → ℂ) (F : X.Forms) : R.toFn t (MvPolynomial.C F) = R.ev F

theorem **toFn_X** (R : Rdg X) {N : ℕ} (t : Idx D j N → ℂ) (p : Idx D j N) : R.toFn t (MvPolynomial.X p) = fun _ => t p

private lemma **algebraMap_mul_mem_Vw** (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (u : WeightFamilies.Wt D j) (x : Op X la N) {v : Rp X la N} (hv : v ∈ Vw X la N u) : algebraMap (Op X la N) (Rp X la N) x * v ∈ Vw X la N u

theorem **toRp_mem** (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (u : WeightFamilies.Wt D j) (p : MvPolynomial (Idx D j N) X.Forms) (hp : OfWt X u p) : toRp X la N p ∈ Vw X la N u

theorem **toFn_mem** (R : Rdg X) {N : ℕ} (t : Idx D j N → ℂ) (u : WeightFamilies.Wt D j) (p : MvPolynomial (Idx D j N) X.Forms) (hp : OfWt X u p) : R.toFn t p ∈ R.sp u Set.univ

theorem **sp_mul** (R : Rdg X) (u u' : WeightFamilies.Wt D j) (f : R.Y → ℂ) (hf : f ∈ R.sp u Set.univ) (g : R.Y → ℂ) (hg : g ∈ R.sp u' Set.univ) : f * g ∈ R.sp (wadd u u') Set.univ

theorem **sp_span** (R : Rdg X) (u : WeightFamilies.Wt D j) (s : Set (B.Cusp j u)) : R.sp u (Submodule.span B.KO s) = R.sp u s

theorem **sp_finrank** (R : Rdg X) (u : WeightFamilies.Wt D j) (M : Submodule B.KO (B.Cusp j u)) : FiniteDimensional ℂ (R.sp u M) ∧ Module.finrank ℂ (R.sp u M) = Module.finrank B.KO M

theorem **polyDictionary** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (R : Rdg X) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (wL : WeightFamilies.Wt D j) (w : Fin N → B.Cusp j wL) (t : Idx D j N → ℂ), (∀ i, toRp X la N (∑ k, MvPolynomial.X (i, k) * MvPolynomial.C (X.inc wL (w k))) = sec X la w i ∧ R.toFn t (∑ k, MvPolynomial.X (i, k) * MvPolynomial.C (X.inc wL (w k))) = R.s w t i ∧ OfWt X wL (∑ k, MvPolynomial.X (i, k) * MvPolynomial.C (X.inc wL (w k)) : MvPolynomial (Idx D j N) X.Forms)) ∧ (∀ F, toRp X la N (MvPolynomial.C F) = emb X la N F ∧ R.toFn t (MvPolynomial.C F) = R.ev F) ∧ (∀ u (F : B.Cusp j u), OfWt X u (MvPolynomial.C (X.inc u F) : MvPolynomial (Idx D j N) X.Forms)) ∧ (∀ u u' (p q : MvPolynomial (Idx D j N) X.Forms), OfWt X u p → OfWt X u' q → OfWt X (wadd u u') (p * q)) ∧ (∀ u (p : MvPolynomial (Idx D j N) X.Forms), OfWt X

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GenericSectionsPolynomialDictionaryStatement

theorem **_root_.GlobalTheory.T.genericSections_polynomial_dictionary** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (R : Rdg X) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (wL : WeightFamilies.Wt D j) (w : Fin N → B.Cusp j wL) (t : Idx D j N → ℂ), (∀ i, toRp X la N (∑ k, MvPolynomial.X (i, k) * MvPolynomial.C (X.inc wL (w k))) = sec X la w i ∧ R.toFn t (∑ k, MvPolynomial.X (i, k) * MvPolynomial.C (X.inc wL (w k))) = R.s w t i ∧ OfWt X wL (∑ k, MvPolynomial.X (i, k) * MvPolynomial.C (X.inc wL (w k)) : MvPolynomial (Idx D j N) X.Forms)) ∧ (∀ F, toRp X la N (MvPolynomial.C F) = emb X la N F ∧ R.toFn t (MvPolynomial.C F) = R.ev F) ∧ (∀ u (F : B.Cusp j u), OfWt X u (MvPolynomial.C (X.inc u F) : MvPolynomial (Idx D j N) X.Forms)) ∧ (∀ u u' (p q : MvPolynomial (Idx D j N) X.Forms), OfWt X u p → OfWt X u' q → OfWt X (wadd u u') (p * q)) ∧ (∀ u (p

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.GlobalBasisAlgebra

theorem **global_basis_algebra** {E : Type} [Field E] [NumberField E] {K S M : Type} [Field K] [CharZero K] [CommRing S] [Algebra K S] [AddCommGroup M] [Module S M] {J : Type} [Fintype J] [DecidableEq J] (χ : J → (E →+* K)) (hχ : Function.Injective χ) (hcard : Fintype.card J = Module.finrank ℚ E) (ρ : 𝒪 E →+* Module.End S M) {mult : J → ℕ} (U : (Σ σ, Fin (mult σ)) → M) (hU : ∀ σ i (a : 𝒪 E), ρ a (U ⟨σ, i⟩) = algebraMap K S (χ σ (a : E)) • U ⟨σ, i⟩) (hb : ∀ σ (W : M), (∀ a : 𝒪 E, ρ a W = algebraMap K S (χ σ (a : E)) • W) → ∃! c : Fin (mult σ) → S, ∑ i, c i • U ⟨σ, i⟩ = W) (W : M) : ∃! a : (Σ σ, Fin (mult σ)) → S, ∑ ℓ, a ℓ • U ℓ = W

theorem **spMap_comp'** {A B C : Type} [CommRing A] [CommRing B] [CommRing C] (f : A →+* B) (g : B →+* C) : spMap g ≫ spMap f = spMap (g.comp f)

theorem **spec_ext_two** {R A : Type} [CommRing R] [CommRing A] (p1 p2 : R →+* A) (hs : Function.Surjective p1) (hn : ∀ x, p1 x = 0 → x * x = 0) (hk : ∀ x, p1 x = 0 → p2 x = 0 → x = 0) {X : Scheme.{0}} (f g : Sp R ⟶ X) (h1 : spMap p1 ≫ f = spMap p1 ≫ g) (h2 : spMap p2 ≫ f = spMap p2 ≫ g) : f = g

theorem **homAt_comp** {A B C : Type} [CommRing A] [Algebra S A] [CommRing B] [Algebra S B] [CommRing C] [Algebra S C] (f : A →ₐ[S] B) (g : B →ₐ[S] C) : homAt x0 g ≫ homAt x0 f = homAt x0 (g.comp f)

theorem **homAt_id** {A : Type} [CommRing A] [Algebra S A] : homAt x0 (AlgHom.id S A) = 𝟙 _

theorem **smT_left** (a : S) (v : overAt x0 (DualNumber S) ⟶ G) : (smT x0 a v).left = spMap (epsMul a) ≫ v.left

theorem **smT_mul** (a : S) (u w : overAt x0 (DualNumber S) ⟶ G) : smT x0 a (u * w) = smT x0 a u * smT x0 a w

theorem **smT_one'** (a : S) : smT x0 a (1 : overAt x0 (DualNumber S) ⟶ G) = 1

theorem **smT_smT** (a b : S) (v : overAt x0 (DualNumber S) ⟶ G) : smT x0 a (smT x0 b v) = smT x0 (a * b) v

theorem **smT_unit** (v : overAt x0 (DualNumber S) ⟶ G) : smT x0 1 v = v

theorem **smT_add** (v : overAt x0 (DualNumber S) ⟶ G) (h0 : smT x0 0 v = 1) (a b : S) : smT x0 (a + b) v = smT x0 a v * smT x0 b v

theorem **tan_pull** {P : Over X0} (ψ : P ⟶ overAt x0 (DualNumber S)) (hψ : ψ ≫ homAt x0 (tm ((0 : S) • (LinearMap.id : S →ₗ[S] S))) = ψ) (W : overAt x0 (DualNumber S) ⟶ G) (h : smT x0 0 W = 1) : ψ ≫ W = 1

theorem **smT_post** (a : S) (v : overAt x0 (DualNumber S) ⟶ G) (φ : G ⟶ G) : smT x0 a (v ≫ φ) = smT x0 a v ≫ φ

theorem **isCombAtG_iff** {m : ℕ} (v : Fin m → (tEpsG x0 ⟶ t.G)) (cf : Fin m → S) (w : Sp (DualNumber S) ⟶ t.G.left) : IsCombAtG t x0 (fun j => (v j).left) cf w ↔ (combT x0 v cf).left = w

private theorem **eps_zero_lemma** : epsMul (0 : S) = (algebraMap S (DualNumber S)).comp (TrivSqZeroExt.fstHom S S S).toRingHom

private theorem **fst_eps_lemma_2** : (TrivSqZeroExt.fstHom S S S).toRingHom.comp (epsMul (0 : S)) = (TrivSqZeroExt.fstHom S S S).toRingHom

private theorem **one_left_lemma_3** {G : Over X0} [GrpObj G] : (1 : tEpsG x0 ⟶ G).left = spMap (algebraMap S (DualNumber S)) ≫ (1 : Over.mk x0 ⟶ G).left

theorem **smT_zero_ovT** {v : Sp (DualNumber S) ⟶ t.G.left} (hv : IsTanAtG t x0 v) : smT x0 0 (ovT hv) = 1

theorem **isTanAtG_of** (W : tEpsG x0 ⟶ t.G) (h : smT x0 0 W = 1) : IsTanAtG t x0 W.left

theorem **isEigAt_iff** [Algebra c.F S] {x : Sp S ⟶ Xo.X} (τ : E →+* ℂ) (W : tEpsG x ⟶ Xo.univ.t.G) : IsEigAt c τ W.left ↔ ∀ a : 𝒪 E, W ≫ Xo.univ.t.act.act a = smT x (algebraMap c.F S (embK c τ a)) W

theorem **tv_zero'** (m : TanM x0 G) : smT x0 0 (tv m) = 1

theorem **tv_mkT** (W : overAt x0 (DualNumber S) ⟶ G) (h : smT x0 0 W = 1) : tv (mkT W h) = W

theorem **tv_injective** : Function.Injective (tv (x0

theorem **tv_add** (m m' : TanM x0 G) : tv (m + m') = tv m * tv m'

theorem **tv_zero** : tv (0 : TanM x0 G) = 1

theorem **tv_sum** {κ : Type} (s : Finset κ) (f : κ → TanM x0 G) : tv (∑ i ∈ s, f i) = ∏ i ∈ s, tv (f i)

theorem **tv_smul** (a : S) (m : TanM x0 G) : tv (a • m) = smT x0 a (tv m)

theorem **tv_postL** (φ : G ⟶ G) [IsMonHom φ] (m : TanM x0 G) : tv (postL φ m) = tv m ≫ φ

theorem **tv_actL** {O : Type} [CommRing O] (ρ : OAct O G) (a : O) (m : TanM x0 G) : tv (actL ρ a m) = tv m ≫ ρ.act a

theorem **tv_comb** {κ : Type} [Fintype κ] (v : κ → TanM x0 G) (a : κ → S) : tv (∑ k, a k • v k) = ∏ k, smT x0 (a k) (tv (v k))

theorem **frame_tangent_basis** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left), IsTanAt y.x w → ∃! a : Lt1 Dw → S, IsCombAt y.x (m

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HalfIntegralGeneratorTraceEqOne

private theorem **complexConj_rt_eq_neg** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : IsCMField.complexConj E (H.rt : E) = -(H.rt : E)

theorem **halfIntegralGenerator_trace_eq_one** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) : H.w0 + H.bar H.w0 = 1

theorem **univ_finset_fin_two** : (Finset.univ : Finset (Finset (Fin 2))) = {∅, {0}, {1}, {0, 1}}

theorem **chi_sum_two** {M : Type} [AddCommMonoid M] (f : ULift.{0} (Finset (Fin 2)) → M) : ∑ I, f I = f ⟨∅⟩ + f ⟨{0}⟩ + f ⟨{1}⟩ + f ⟨{0, 1}⟩

theorem **chi_prod_two** {M : Type} [CommMonoid M] (f : ULift.{0} (Finset (Fin 2)) → M) : ∏ I, f I = f ⟨∅⟩ * f ⟨{0}⟩ * f ⟨{1}⟩ * f ⟨{0, 1}⟩

theorem **chi_eq_sign** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ a ∈ Φ.lat, Φ.χ a ^ 2 = 1 ∧ Φ.χ a = Complex.exp (Real.pi * Complex.I * ((hermForm Φ.Hm a (tyAct ty H.w0 a)).im : ℂ))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HalfLevelGeneratorNorm

theorem **half_level_generator_norm** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (_EX : ErrX D), ∃ α : 𝒪 D.host.E, α * T.Q.H.bar α = (T.Q.l.M : 𝒪 D.host.E) ∧ T.Q.l.Mb = Ideal.span {T.Q.H.bar α}

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HalvesFreeAndAdjoint

theorem **nonempty_equiv_quot** {R N : Type} [CommRing R] [IsDomain R] [AddCommGroup N] [Module R N] [Module.Finite R N] [Module.Projective R N] (I : Ideal R) [Finite (R ⧸ I)] (r : ℕ) (hr : Module.finrank R N = r) : Nonempty ((N ⧸ (I • ⊤ : Submodule R N)) ≃ₗ[R] (Fin r → R ⧸ I))

theorem **exists_equiv_of_span** {R Q : Type} [CommRing R] [AddCommGroup Q] [Module R Q] (I : Ideal R) (r : ℕ) (e : Q ≃ₗ[R] (Fin r → R ⧸ I)) (x : Fin r → Q) (hx : Submodule.span R (Set.range x) = ⊤) : ∃ ψ : (Fin r → R ⧸ I) ≃ₗ[R] Q, ∀ b, ψ (Pi.single b 1) = x b

theorem **exists_span** {R Q : Type} [CommRing R] [AddCommGroup Q] [Module R Q] (I : Ideal R) (r : ℕ) (e : Q ≃ₗ[R] (Fin r → R ⧸ I)) : ∃ x : Fin r → Q, Submodule.span R (Set.range x) = ⊤

theorem **existsUnique_equiv** {R Q : Type} [CommRing R] [AddCommGroup Q] [Module R Q] (I : Ideal R) (r : ℕ) (e : Q ≃ₗ[R] (Fin r → R ⧸ I)) (x x' : Fin r → Q) (hx : Submodule.span R (Set.range x) = ⊤) (hx' : Submodule.span R (Set.range x') = ⊤) : ∃! g : Q ≃ₗ[R] Q, ∀ b, g (x b) = x' b

theorem **exists_adjoint** {R P Q Z : Type} [CommRing R] [AddCommGroup P] [Module R P] [AddCommGroup Q] [Module R Q] [AddCommGroup Z] (s : R → R) (B : P → Q → Z) (hadd : ∀ u v w, B u (v + w) = B u v + B u w) (hbal : ∀ a u v, B u (a • v) = B (s a • u) v) (hnd : ∀ v w, (∀ u, B u v = B u w) → v = w) (hrep : ∀ (g : P →ₗ[R] P) (v : Q), ∃ w, ∀ u, B (g u) v = B u w) : ∃ adj : (P ≃ₗ[R] P) → (Q ≃ₗ[R] Q), (∀ g g', adj (g * g') = adj g' * adj g) ∧ ∀ (k : Q ≃ₗ[R] Q) (g : P ≃ₗ[R] P), (∀ u v, B (g u) v = B u (k v)) ↔ k = adj g

theorem **cHalf_spec** (I : Ideal (𝒪 E)) (hI : I ⊔ I.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤) : cHalf H I ∈ I.map (H.bar : 𝒪 E →+* 𝒪 E) ∧ cHalf H I - 1 ∈ I

theorem **ef_swap** (x y : Dw.La) : Dw.EF y x = -Dw.EF x y

theorem **qRep_mk** {K : Submodule (𝒪 E) Dw.La} (q : Dw.La ⧸ K) : Submodule.Quotient.mk (qRep q) = q

private theorem **host_bar_bar'** (a : 𝒪 E) : H.bar (H.bar a) = a

private theorem **dvd_EF_cHalf_sub_of_sub_mem** {N : ℕ} {I : Ideal (𝒪 E)} (hp : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) (hI : I ⊔ I.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤) (a a' b b' : Dw.La) (hab : a - a' ∈ (I.map (H.bar : 𝒪 E →+* 𝒪 E) • ⊤ : Submodule (𝒪 E) Dw.La)) (hbb : b - b' ∈ (I • ⊤ : Submodule (𝒪 E) Dw.La)) : (N : ℤ) ∣ Dw.EF a' (cHalf H I • b') - Dw.EF a (cHalf H I • b)

private theorem **mem_ideal_smul_of_forall_dvd_EF_cHalf** {N : ℕ} (hN : N ≠ 0) {I : Ideal (𝒪 E)} (hp : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) (hI : I ⊔ I.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤) (z : Dw.La) (h : ∀ x : Dw.La, (N : ℤ) ∣ Dw.EF x (cHalf H I • z)) : z ∈ (I • ⊤ : Submodule (𝒪 E) Dw.La)

private theorem **exists_forall_dvd_EF_cHalf_sub_of_linearMap** {N : ℕ} {I : Ideal (𝒪 E)} (hp : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) (hI : I ⊔ I.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤) (G : Dw.La →ₗ[𝒪 E] Dw.La) (t : Dw.La) : ∃ m : Dw.La, ∀ a : Dw.La, (N : ℤ) ∣ Dw.EF a (cHalf H I • m) - Dw.EF (G a) (cHalf H I • t)

theorem **exists_adjoint_halves** (N : ℕ) (hN : N ≠ 0) (I : Ideal (𝒪 E)) (hp : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) (hI : I ⊔ I.map (H.bar : 𝒪 E →+* 𝒪 E) = ⊤) (KJ KI : Submodule (𝒪 E) Dw.La) (hKJ : KJ = I.map (H.bar : 𝒪 E →+* 𝒪 E) • ⊤) (hKI : KI = I • ⊤) : ∃ adj : ((Dw.La ⧸ KJ) ≃ₗ[𝒪 E] (Dw.La ⧸ KJ)) → ((Dw.La ⧸ KI) ≃ₗ[𝒪 E] (Dw.La ⧸ KI)), (∀ g g', adj (g * g') = adj g' * adj g) ∧ ∀ (k : (Dw.La ⧸ KI) ≃ₗ[𝒪 E] (Dw.La ⧸ KI)) (kst : (Dw.La ⧸ KJ) ≃ₗ[𝒪 E] (Dw.La ⧸ KJ)), (∀ u v : Dw.La, ((N : ℤ) ∣ Dw.EF (qRep (kst (Submodule.Quotient.mk u))) (cHalf H I • v) - Dw.EF u (cHalf H I • qRep (k (Submodule.Quotient.mk v))))) ↔ k = adj kst

theorem **halves_free_and_adjoint** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l), ∀ (n : ℕ) (d : DeltaDat Xo c n), d.IsLev → (∃ x₀ : Fin (Dw.p + Dw.q) → LaJ Xo c d, Submodule.span (𝒪 E) (Set.range x₀) = ⊤) ∧ (∀ x x' : Fin (Dw.p + Dw.q) → LaJ Xo c d, Submodule.span (𝒪 E) (Set.range x) = ⊤ → Submodule.span (𝒪 E) (Set.range x') = ⊤ → ∃! g : LaJ Xo c d ≃ₗ[𝒪 E] LaJ Xo c d, ∀ b, g (x b) = x' b) ∧ ∃ adj : (LaJ Xo c d ≃ₗ[𝒪 E] LaJ Xo c d) → (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d), (∀ g g', adj (g * g') = adj g' * adj g) ∧ ∀ (k : LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) (kst : LaJ Xo c d ≃ₗ[𝒪 E] LaJ Xo c d), (∀ u v : Dw.La, ((d.pd : ℤ) ∣ Dw.EF (qRep (kst (Submodule.Quotient.mk u))) (cHalf H d.I • v) - Dw.EF u (cHalf H d.I • qRep (k (Submodule.Quotient.mk v))))) ↔ k = adj kst

theorem **norm_of_pow** {F : Type} [Field F] (σ : F →+* ℂ) {z : F} {N : ℕ} (hN : N ≠ 0) (hz : z ^ N = 1) : ‖σ z‖ = 1

theorem **absNorm_conj** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : Host E) (I : Ideal (𝒪 E)) (N : ℕ) (hI : I * I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(N : 𝒪 E)}) : Nat.card (𝒪 E ⧸ I.map (H.bar : 𝒪 E →+* 𝒪 E)) = N ^ fdeg E

private theorem **norm_embedding_zpow_eq_one_of_pow_eq_one** {F : Type} [Field F] (σ : F →+* ℂ) {z : F} {N : ℕ} (hN : N ≠ 0) (hz : z ^ N = 1) (k : ℤ) : ‖σ (z ^ k)‖ = 1

private theorem **norm_embedding_inv_character_eq_one** {G K F : Type} [Group G] [Finite G] [Field K] [Field F] (ψ : G →* Kˣ) (f : K →+* F) (σ : F →+* ℂ) (k : G) : ‖σ (f (((ψ k)⁻¹ : Kˣ) : K))‖ = 1

private theorem **ideal_mul_map_bar_eq_span_pd_of_isLev** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} {d : DeltaDat Xo c n} (hd : d.IsLev) : d.I * d.I.map (H.bar : 𝒪 E →+* 𝒪 E) = Ideal.span {(d.pd : 𝒪 E)} ∧ d.pd ≠ 0

private theorem **finite_quotient_I_and_map_bar_of_isLev** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} {d : DeltaDat Xo c n} (hd : d.IsLev) : Finite (𝒪 E ⧸ d.I) ∧ Finite (𝒪 E ⧸ d.I.map (H.bar : 𝒪 E →+* 𝒪 E))

private theorem **finite_linearEquiv_laI_of_isLev** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} {d : DeltaDat Xo c n} (hd : d.IsLev) : Finite (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d)

private theorem **card_laJ_eq_pd_pow_of_isLev** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} {d : DeltaDat Xo c n} (hd : d.IsLev) : Nat.card (LaJ Xo c d) = d.pd ^ (fdeg E * (Dw.p + Dw.q))

private theorem **norm_eF_ph_le_one_of_phiPin** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} {d : DeltaDat Xo c n} (hd : d.IsLev) {ph : (Fin n → LaJ Xo c d) → c.F} {ψ : (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) →* c.Fˣ} {x₀ : Fin n → LaJ Xo c d} (hpin : d.PhiPin ph ψ x₀) (hn : Dw.p + Dw.q ≤ n) (σ : d.c₂.F →+* ℂ) (x : Fin n → LaJ Xo c d) : ‖σ (d.eF (ph x))‖ ≤ 1

theorem **norm_coeff_le** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), d.IsLev → 0 < d.pd → ∀ ph ψ x₀, d.PhiPin ph ψ x₀ → Dw.p + Dw.q ≤ n → ∀ (x0 : Fin n → Dw.La) (q : Fin n → LevQ Dw d.l₂) (σ : d.c₂.F →+* ℂ), ‖σ (d.w x0 q)‖ ≤ (d.pd : ℝ) ^ (fdeg E * (Dw.p + Dw.q) * n)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmAMemSpanShort

theorem **mem_std** {n : ℕ} (v : Fin n → E) : v ∈ stdLat E n ↔ ∀ j, IsIntegral ℤ (v j)

theorem **single_mem_std** {n : ℕ} (j : Fin n) : (Pi.single j 1 : Fin n → E) ∈ stdLat E n

theorem **std_fg** (n : ℕ) : (stdLat E n).FG

theorem **std_unimod** (n : ℕ) : (⟨stdLat E n⟩ : GenLat E n).Unimod

theorem **symO_std** (n : ℕ) : Formalism.L2.Pj.symO (Dw

theorem **harmL_std** (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) : harmL (Dw

theorem **harmA_mem_span_short** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n : ℕ), (∀ a : (E →+* ℂ) × ℕ → ℕ, harmL (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicSumTriangularFrameChange

theorem **basis_of_triangular** {T : Type} [Fintype T] {r : T → ℕ} {S : Type} [CommRing S] {d : ℕ} (u : (Σ τ : T, Fin (r τ)) → Fin d → S) (cf : ∀ τ : T, Fin (r τ) → Fin (r τ) → S) (h0 : ∀ τ i j, i < j → cf τ i j = 0) (h1 : ∀ τ i, IsUnit (cf τ i i)) (hb : ∀ v : Fin d → S, ∃! a : (Σ τ : T, Fin (r τ)) → S, ∀ j, ∑ ℓ, a ℓ * u ℓ j = v j) (v : Fin d → S) : ∃! a : (Σ τ : T, Fin (r τ)) → S, ∀ e, ∑ ℓ, a ℓ * (∑ j, cf ℓ.1 ℓ.2 j * u ⟨ℓ.1, j⟩ e) = v e

theorem **fst_comp_jetTan** {S : Type} [CommRing S] {κ : Type} [DecidableEq κ] (ℓ : κ) : (TrivSqZeroExt.fstHom S S S).comp (jetTan ℓ : Jet S κ (0 + 1) →ₐ[S] DualNumber S) = jetAug

theorem **left_one** {T : Scheme.{0}} {X : Over T} {M : Over T} [GrpObj M] : (1 : X ⟶ M).left = X.hom ≫ η[M].left

theorem **isTanAt_jetTan** {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) {κ : Type} [DecidableEq κ] (x : overAt x0 (Jet S κ (0 + 1)) ⟶ Xo.univ.t.G) (hx : homAt x0 (jetAug : Jet S κ (0 + 1) →ₐ[S] S) ≫ x = 1) (ℓ : κ) : IsTanAt x0 (homAt x0 (jetTan ℓ : Jet S κ (0 + 1) →ₐ[S] DualNumber S) ≫ x).left

theorem **eq_of_isPullSec_id** {X : Scheme.{0}} (A : Over X) [GrpObj A] (L : LB A.left) (k : ℤ) (T : Over X) (F F' : LB.Sec (LPow A L k T)) (h : IsPullSec A L k (𝟙 T ▷ A) (whiskerRight_snd _ _) F F') : F' = F

theorem **sum_eq** (hC : ChartIndepSt2) [StarRing (𝒪 E)] (nab : NabV Xo c) (hnab : IsNab Xo c nab) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (ho u' h1 h2 h3 s' h4) (cf : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Fin (Dw.mult τ) → S) (h0 : ∀ τ i j, i < j → cf τ i j = 0) (hu : ∀ τ i, IsUnit (cf τ i i)) (hcomb : ∀ τ i, IsCombAt y.x (y.u τ) (cf τ i) (u' τ i)) (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) (k : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) : ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab (⟨y.x, ho, u', h1, h2, h3, s', h4⟩ : FrPt Xo c S) k (F b) (colOf m b) = (∏ ℓ : Lt1 Dw, cf ℓ.1 ℓ.2 ℓ.2 ^ a ⟨ℓ.1, ℓ.2⟩) * ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab y k (F b) (colOf m b)

theorem **harmonic_sum_frBor** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (t : Ix Dw → Sˣ), FrBor t y y' → ∀ (h : ptO Xo y'.x ⟶ ptO Xo y.x), h.left = 𝟙 _ → ∀ (n : ℕ), ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (N : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y'.x))), (∀ b, IsPullSec Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) (F b) (F' b)) → ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab y' ((N : ℤ) ^ 2) (F' b) (c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicSumTriangularFrameChangeStatement

theorem **_root_.GlobalTheory.T.harmonic_sum_triangular_frameChange** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (t : Ix Dw → Sˣ), FrBor t y y' → ∀ (h : ptO Xo y'.x ⟶ ptO Xo y.x), h.left = 𝟙 _ → ∀ (n : ℕ), ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (N : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y'.x))), (∀ b, IsPullSec Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) (F b) (F' b)) → ∑ m ∈ P.support, algebraMap c.F S (P.coeff m

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicValueBaseChange

theorem **sum_map_of_harmA** (hC : ChartIndepSt2) {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} [StarRing (𝒪 E)] (nab : NabV Xo c) (hnab : IsNab Xo c nab) {S S' : Type} [CommRing S] [Algebra c.F S] [Algebra ℚ S] [CommRing S'] [Algebra c.F S'] [Algebra ℚ S'] (ψ : S →ₐ[c.F] S') (y : FrPt Xo c S) (y' : FrPt Xo c S') (hm : FrMap ψ y y') (h : ptO Xo y'.x ⟶ ptO Xo y.x) (hh : h.left = spMap ψ.toRingHom) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ harmA c a n) (K : ℤ) (k : ℕ) (hK : K = k) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L K (ptO Xo y.x))) (F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L K (ptO Xo y'.x))) (hF' : ∀ b, IsPullSec Xo.univ.t.G Xo.univ.t.PP.L K (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) (F b) (F' b)) : ψ (∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b,

theorem **harmonicValue_baseChange** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S S' : Type) [CommRing S] [Algebra c.F S] [CommRing S'] [Algebra c.F S'] (ψ : S →ₐ[c.F] S') (y : FrPt Xo c S) (y' : FrPt Xo c S'), FrMap ψ y y' → ∀ (h : ptO Xo y'.x ⟶ ptO Xo y.x), h.left = spMap ψ.toRingHom → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F

ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (N : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y'.x))), (∀ b, IsPullSec Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) (F b) (F' b)) → ψ (∑ m ∈ P.su

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicValueBaseChangeStatement

theorem **_root_.GlobalTheory.T.harmonicValue_baseChange_alt** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S S' : Type) [CommRing S] [Algebra c.F S] [CommRing S'] [Algebra c.F S'] (ψ : S →ₐ[c.F] S') (y : FrPt Xo c S) (y' : FrPt Xo c S'), FrMap ψ y y' → ∀ (h : ptO Xo y'.x ⟶ ptO Xo y.x), h.left = spMap ψ.toRingHom → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (N : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y'.x))), (∀ b, IsPullSec Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) (F

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicValueCanonicalConnectionEqPolyDeriv

theorem **sum_support_natCast_smul** {σ K : Type} [Field K] [Algebra K ℂ] (N : ℕ) (P : MvPolynomial σ K) (hs : ((N : K) • P).support = P.support) (v : (σ →₀ ℕ) → ℂ) : ∑ m ∈ ((N : K) • P).support, algebraMap K ℂ (((N : K) • P).coeff m) * v m = (N : ℂ) * ∑ m ∈ P.support, algebraMap K ℂ (P.coeff m) * v m

theorem **harm_nab_eq_polyDeriv** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ nab : NabV Xo c, IsNab Xo c nab → ∀ (k n : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) (θ : Fin n → (Fin g → ℂ) → ℂ), (∀ b, Φ.toAnFr.Rd k (F b) (θ b)) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∑ m ∈ P.support, algebraMap c.F ℂ (P.coeff m) * ∏ b, nab y k (F b) (colOf m b) = I2.polyDeriv (MvPolynomial.map (algebraMap c.F ℂ) P) (letDir el) (fun z => ∏ b, θ b (z b)) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicValueCanonicalConnectionEqPolyDerivStatement

theorem **_root_.GlobalTheory.T.harmonicValue_canonicalConnection_eq_polyDeriv** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ nab : NabV Xo c, IsNab Xo c nab → ∀ (k n : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) (θ : Fin n → (Fin g → ℂ) → ℂ), (∀ b, Φ.toAnFr.Rd k (F b) (θ b)) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∑ m ∈ P.support, algebraMap c.F ℂ (P.coeff m) * ∏ b, nab y k (F b) (colOf m b) = I2.polyDeriv (MvPolynomial.map (algebraMap c.F ℂ) P) (letDir el) (fun z => ∏ b, θ b (z b)) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicValuesInvariantUniversalEndomorphism

theorem **exists_up_om** {w : LB.Iso L' (L.pull Φ)} (hw : w.IsGen) (hL : L.IsLB) (hL' : L'.IsLB) (φ' : K → (Y ⟶ X')) (φ : K → (Y ⟶ X)) (ε : K → ℤ) (hc : ∀ k, φ' k ≫ Φ = φ k) (a : LB.Gen (LB.LinComb L' φ' ε)) (ha : a.IsGen) : ∃ t : LB.Gen (LB.LinComb L φ ε), t.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) J J', Good L z J → Good L' z J' → val t r z J = (∏ k, om w z J J' (r ≫ φ' k) ^ ε k) * val a r z J'

theorem **exists_down_om** {Y' : Scheme.{0}} {w : LB.Iso L' (L.pull Φ)} (hw : w.IsGen) (hL : L.IsLB) (hL' : L'.IsLB) {φ' : K → (Y' ⟶ X')} {φ : K → (Y ⟶ X)} {ε : K → ℤ} (h : Y' ⟶ Y) (hψ : ∀ k, φ' k ≫ Φ = h ≫ φ k) (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) : ∃ a : LB.Gen (LB.LinComb L' φ' ε), a.IsGen ∧ ∀ (Z : Scheme.{0}) (r : Z ⟶ Y') (z : Z) J J', Good L z J → Good L' z J' → val t (r ≫ h) z J = (∏ k, om w z J J' (r ≫ φ' k) ^ ε k) * val a r z J'

theorem **lam_one_left** {A U : Type} [CommGroup A] [CommGroup U] (h : A → U) (b : A) : lam h 1 b = 1

theorem **lam_one_right** {A U : Type} [CommGroup A] [CommGroup U] (h : A → U) (b : A) : lam h b 1 = 1

theorem **lam_comm** {A U : Type} [CommGroup A] [CommGroup U] (h : A → U) (a b : A) : lam h a b = lam h b a

theorem **prod_lam** {T T' : Over X} (σ : T' ⟶ T) (a b : T ⟶ t.G) {U : Type} [CommGroup U] (h : (T'.left ⟶ t.G.left) → U) : ∏ I : ULift.{0} (Finset (Fin 2)), h (σ.left ≫ cφG t ![a, b] I) ^ cε 2 I = lam (fun p : T' ⟶ t.G => h p.left) (σ ≫ a) (σ ≫ b)

theorem **cφ_push** {T : Over X} {T' : Over X'} (h : T'.left ⟶ T.left) {a' b' : T' ⟶ t'.G} {a b : T ⟶ t.G} (ha : a'.left ≫ ν.φ = h ≫ a.left) (hb : b'.left ≫ ν.φ = h ≫ b.left) (I : ULift.{0} (Finset (Fin 2))) : cφG t' ![a', b'] I ≫ ν.φ = h ≫ cφG t ![a, b] I

theorem **act_push** {T : Over X} {T' : Over X'} (h : T'.left ⟶ T.left) {a' : T' ⟶ t'.G} {a : T ⟶ t.G} (ha : a'.left ≫ ν.φ = h ≫ a.left) (c : 𝒪 E) : (a' ≫ t'.act.act c).left ≫ ν.φ = h ≫ (a ≫ t.act.act c).left

theorem **rig_val** (hr : LB.Gen.Maps (LB.linCombIso w (K

theorem **cub_val** (hc : LB.Gen.Maps (LB.linCombIso w (thφ t'.G (x3 t'.G)) thε) t'.cub.t (ν.cm.thetaPull t.PP.L t.cub.t)) {Y : Scheme.{0}} {g : Y ⟶ X} {g' : Y ⟶ X'} {a' b' c' : Over.mk g' ⟶ t'.G} {a b c : Over.mk g ⟶ t.G} (ha : a'.left ≫ ν.φ = a.left) (hb : b'.left ≫ ν.φ = b.left) (hc' : c'.left ≫ ν.φ = c.left) (r : Z ⟶ Y) : val (L

theorem **prim_val** (hG : GlobalFnsSt) (hA : IsAbelianScheme t'.G) (hw : w.IsGen) (hr : LB.Gen.Maps (LB.linCombIso w (K

theorem **ad_val** (hG : GlobalFnsSt) (hA : IsAbelianScheme t'.G) (hw : w.IsGen) {c : 𝒪 E} {wL : AdT t c} {wL' : AdT t' c} (hL : wL.IsGen) (hL' : wL'.IsGen) (hn : ∀ (r : Z ⟶ (𝟙_ (Over X)).left), val wL (r ≫ (GermB.zz t).left) z J = 1) {Y : Scheme.{0}} {g : Y ⟶ X} {g' : Y ⟶ X'} {a' b' : Over.mk g' ⟶ t'.G} {a b : Over.mk g ⟶ t.G} (ha : a'.left ≫ ν.φ = a.left) (hb : b'.left ≫ ν.φ = b.left) (r : Z ⟶ Y) (hn' : val wL' (r ≫ (lift (1 : Over.mk g' ⟶ t'.G) (1 : Over.mk g' ⟶ t'.G)).left) z J' = 1) : val wL (r ≫ (lift a b).left) z J = lam (fun p : Over.mk g' ⟶ t'.G => om w z J J' (r ≫ p.left)) a' (b' ≫ t'.act.act (H.bar c)) * (lam (fun p : Over.mk g' ⟶ t'.G => om w z J J' (r ≫ p.left)) (a' ≫ t'.act.act c) b')⁻¹ * val wL' (r ≫ (lift a' b').left) z J'

theorem **pu_left** (h0 : x0' ≫ ρ = x0) {A : Type} [CommRing A] [Algebra S A] (a : overAt x0' A ⟶ t'.G) : (pu ν h0 A a).left = a.left ≫ ν.φ

theorem **hom_pu** (h0 : x0' ≫ ρ = x0) {A B : Type} [CommRing A] [Algebra S A] [CommRing B] [Algebra S B] (f : A →ₐ[S] B) (a : overAt x0' A ⟶ t'.G) : homAt x0 f ≫ pu ν h0 A a = pu ν h0 B (homAt x0' f ≫ a)

theorem **pu_act** (h0 : x0' ≫ ρ = x0) {A : Type} [CommRing A] [Algebra S A] (a : overAt x0' A ⟶ t'.G) (c : 𝒪 E) : pu ν h0 A a ≫ t.act.act c = pu ν h0 A (a ≫ t'.act.act c)

theorem **isChart_push** (h0 : x0' ≫ ρ = x0) (hc : IsChartG t' x0' d F act tg uvec x') : IsChartG t x0 d F act (fun ℓ => tg ℓ ≫ ν.φ) uvec (fun m => pu ν h0 _ (x' m))

theorem **jpt_push** (hx : ∀ m, (x' m).left ≫ ν.φ = (x m).left) {n : ℕ} (m : ℕ) (i : Fin n) : (jptG t' x0' d x' m i).left ≫ ν.φ = (jptG t x0 d x m i).left

theorem **exists_frRel** (hw : w.IsGen) (hx : ∀ m, (x' m).left ≫ ν.φ = (x m).left) (e' : ∀ m, FrTyG t' x0' d x' m) (he' : ∀ m, (e' m).IsGen) : ∃ e : ∀ m, FrTyG t x0 d x m, (∀ m, (e m).IsGen) ∧ FrRel w e' e

theorem **val_pe** (he : FrRel w e' e) {Z : Scheme.{0}} {z : Z} {J J'} (hJ : Good t.PP.L z J) (hJ' : Good t'.PP.L z J') {m n : ℕ} (f g : Fin d → MvPolynomial (Fin n × Fin d) S) (r : Z ⟶ Sp (Jet S (Fin n × Fin d) m)) {a b : jOG x0' d m n ⟶ t'.G} (ha : homAt x0' (jetSubstP m (fun p : Fin 1 × Fin d => f p.2)) ≫ x' m = a) (hb : homAt x0' (jetSubstP m (fun p : Fin 1 × Fin d => g p.2)) ≫ x' m = b) : val (peG t x0 d x e m ![f, g]) r z J = lam (fun p : jOG x0' d m n ⟶ t'.G => om w z J J' (r ≫ p.left)) a b ∗ val (peG t' x0' d x' e' m ![f, g]) r z J'

theorem **val_scale** {Y Z : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ t.G.left)} {ε : K → ℤ} {z : Z} {J} (hJ : Good t.PP.L z J) (u : Γ(Y, ⊤)ˣ) (g : LB.Gen (LB.LinComb t.PP.L φ ε)) (r : Z ⟶ Y) : val (scaleGenG t φ ε u g) r z J = gm r z ⊤ u ∗ val g r z J

theorem **val_three** {Y Z : Scheme.{0}} {K₁ K₂ K₃ : Type} [Fintype K₁] [Fintype K₂] [Fintype K₃] {φ₁ : K₁ → (Y ⟶ t.G.left)} {ε₁ : K₁ → ℤ} {φ₂ : K₂ → (Y ⟶ t.G.left)} {ε₂ : K₂ → ℤ} {φ₃ : K₃ → (Y ⟶ t.G.left)} {ε₃ : K₃ → ℤ} {z : Z} {J} (hJ : Good t.PP.L z J) (u : Γ(Y, ⊤)ˣ) (g₁ : LB.Gen (LB.LinComb t.PP.L φ₁ ε₁)) (g₂ : LB.Gen (LB.LinComb t.PP.L φ₂ ε₂)) (g₃ : LB.Gen (LB.LinComb t.PP.L φ₃ ε₃)) (r : Z ⟶ Y) : val (scaleGenG t _ _ u (LB.mulGen t.PP.L _ _ _ _ g₁ (LB.mulGen t.PP.L _ _ _ _ (LB.invGen t.PP.L _ _ g₂) (LB.invGen t.PP.L _ _ g₃)))) r z J = gm r z ⊤ u ∗ (val g₁ r z J ∗ ((val g₂ r z J)⁻¹ ∗ (val g₃ r z J)⁻¹))

theorem **psJet_zero_of** {κ : Type} [Fintype κ] [DecidableEq κ] (f : MvPowerSeries κ S) (hf : MvPowerSeries.constantCoeff f = 1) : psJet 0 f = 1

theorem **ad_zero** (hc : IsChartG t' x0' d F act tg uvec x') {c : 𝒪 E} {wL : AdT t' c} {u : (Jet S (Fin 2 × Fin d) 0)ˣ} (hu : u = 1) (hr : LB.Gen.Corr t'.PP.L _ _ _ _ (LB.mulGen t'.PP.L _ _ _ _ (LB.pullGen t'.PP.L _ _ wL (lift (jptG t' x0' d x' 0 (0 : Fin 2)) (jptG t' x0' d x' 0 (1 : Fin 2))).left) (peG t' x0' d x' e' 0 ![fun j => psIn d 0 ![0] (act c j), fun j => MvPolynomial.X ((1 : Fin 2), j)])) (scaleGenG t' _ _ (unitΓ u) (peG t' x0' d x' e' 0 ![fun j => MvPolynomial.X ((0 : Fin 2), j), fun j => psIn d 0 ![1] (act (H.bar c) j)])))) {Z : Scheme.{0}} {z : Z} {J'} (hJ' : Good t'.PP.L z J') {m : ℕ} (r : Z ⟶ Sp (Jet S (Fin 2 × Fin d) m)) : val wL (r ≫ (lift (1 : jOG x0' d m 2 ⟶ t'.G) (1 : jOG x0' d m 2 ⟶ t'.G)).left) z J' = 1

theorem **alg_three** {U : Type} [CommGroup U] {A A' l₁ l₂ l₃ u p₁ p₂ p₃ q₁ q₂ q₃ : U} (k₀ : A = l₁ ∗ l₂⁻¹ ∗ l₃⁻¹ ∗ A') (k : A' = u ∗ (q₁ ∗ (q₂⁻¹ ∗ q₃⁻¹))) (k₁ : p₁ = l₁ ∗ q₁) (k₂ : p₂ = l₂ ∗ q₂) (k₃ : p₃ = l₃ ∗ q₃) : A = u ∗ (p₁ ∗ (p₂⁻¹ ∗ p₃⁻¹))

theorem **alg_one** {U : Type} [CommGroup U] {A A' l l' u p q : U} (k₀ : A = l' ∗ A') (e : l' = l) (k : A' = u ∗ q) (k₁ : p = l ∗ q) : A = u ∗ p

theorem **alg_two** {U : Type} [CommGroup U] {A A' l₁ l₂ u p₁ p₂ q₁ q₂ : U} (k₀ : A = l₂ ∗ l₁⁻¹ ∗ A') (k : A' ∗ q₁ = u ∗ q₂) (k₁ : p₁ = l₁ ∗ q₁) (k₂ : p₂ = l₂ ∗ q₂) : A ∗ p₁ = u ∗ p₂

theorem **corr_of_val₂** {Y : Scheme.{0}} {K K' : Type} [Fintype K] [Fintype K'] {φ : K → (Y ⟶ t.G.left)} {ε : K → ℤ} {φ' : K' → (Y ⟶ t.G.left)} {ε' : K' → ℤ} {a : LB.Gen (LB.LinComb t.PP.L φ ε)} {b : LB.Gen (LB.LinComb t.PP.L φ' ε')} (t' : SAB H X') (h : ∀ (Z : Scheme.{0}) (r : Z ⟶ Y) (z : Z) J J', Good t.PP.L z J → Good t'.PP.L z J' → val a r z J = val b r z J) : LB.Gen.Corr t.PP.L φ ε φ' ε' a b

theorem **isFrame_push** (hG : GlobalFnsSt) (hA : IsAbelianScheme t'.G) (hw : w.IsGen) (hcub : LB.Gen.Maps (LB.linCombIso w (thφ t'.G (x3 t'.G)) thε) t'.cub.t (ν.cm.thetaPull t.PP.L t.cub.t)) (hc : IsChartG t' x0' d F act tg uvec x') (hx : ∀ m, (x' m).left ≫ ν.φ = (x m).left) (he : FrRel w e' e) (hg : ∀ m, (e m).IsGen) {ph1 ph2 : MvPowerSeries (Fin 3 × Fin d) S} {sy : MvPowerSeries (Fin 2 × Fin d) S} {ad : 𝒪 E → MvPowerSeries (Fin 2 × Fin d) S} (had : ∀ c, MvPowerSeries.constantCoeff (ad c) = 1) (hf : IsFrameG t' x0' d x' F act ph1 ph2 sy ad e') : IsFrameG t x0 d x F act ph1 ph2 sy ad e

theorem **alg_prim** {U : Type} [CommGroup U] {A A' l o p q : U} (k₀ : A = o ∗ l⁻¹ ∗ A') (k₁ : p = l ∗ q) : A ∗ p = o ∗ (A' ∗ q)

theorem **val_eL** (hG : GlobalFnsSt) (hA : IsAbelianScheme t'.G) (hw : w.IsGen) (hr : LB.Gen.Maps (LB.linCombIso w (K

theorem **rig_push** (hG : GlobalFnsSt) (hA : IsAbelianScheme t'.G) (hw : w.IsGen) (h0 : x0' ≫ ρ = x0) (hr : LB.Gen.Maps (LB.linCombIso w (K

theorem **ptSecT_snd** {T : Over X} (a : T ⟶ t.G) : (ptSecT t.G a).left ≫ (snd T t.G).left = a.left

theorem **sv_of_isValueAt** {n : ℤ} {T : Over X} {s : LB.Sec (LPow t.G t.PP.L n T)} {a : T ⟶ t.G} {τ : LB.Gen (LB.LinComb t.PP.L (Y

theorem **isValueAt_of_sv** {n : ℤ} {T : Over X} {s : LB.Sec (LPow t.G t.PP.L n T)} {a : T ⟶ t.G} {τ : LB.Gen (LB.LinComb t.PP.L (Y

theorem **sv_pullSec** {n : ℤ} {T T' : Over X} {h : T' ⊗ t.G ⟶ T ⊗ t.G} {hh : h ≫ snd T t.G = snd T' t.G} {s : LB.Sec (LPow t.G t.PP.L n T)} {s' : LB.Sec (LPow t.G t.PP.L n T')} (hs : IsPullSec t.G t.PP.L n h hh s s') {Z : Scheme.{0}} (r : Z ⟶ (T' ⊗ t.G).left) (z : Z) {J} (hJ : Good t.PP.L z J) : sv t s' r z J = sv t s (r ≫ h.left) z J

theorem **isGerm_push** (hG : GlobalFnsSt) (hA : IsAbelianScheme t'.G) (hw : w.IsGen) (hr : LB.Gen.Maps (LB.linCombIso w (K

theorem **gs_appLE** {A B Z : Scheme.{0}} (g : B ⟶ A) (r : Z ⟶ B) (z : Z) {U : A.Opens} {W : B.Opens} (h : W ≤ g ⁻¹ᵁ U) (hz : r z ∈ W) (a : Γ(A, U)) : gs r z W ((g.appLE U W h).hom a) = gs (r ≫ g) z U a

theorem **gs_pow** {A Z : Scheme.{0}} (r : Z ⟶ A) (z : Z) (U : A.Opens) (h : r z ∈ U) (a : Γ(A, U)) (n : ℕ) : gs r z U (a ^ n) = gs r z U a ^ n

theorem **sv_of_charts** {T : Over X} {T' : Over X'} {pm : (T' ⊗ t'.G).left ⟶ (T ⊗ t.G).left} (p₂ : pm ≫ (snd T t.G).left = (snd T' t'.G).left ≫ ν.φ) {a : ℤ} {k : ℕ} {s' : LB.Sec (LPow t'.G t'.PP.L a T')} {s : LB.Sec (LPow t.G t.PP.L a T)} (h : ∀ (i : t.PP.L.ι) (j : t'.PP.L.ι) (W : (T' ⊗ t'.G).left.Opens) (h₀ : W ≤ (snd T' t'.G).left ⁻¹ᵁ ((LB.div (t.PP.L.pull ν.cm.F) t'.PP.L).U ((fun _ => i), j))) (h₁ : W ≤ pm ⁻¹ᵁ (LPow t.G t.PP.L a T).U (fun _ => i)) (h₂ : W ≤ (LPow t'.G t'.PP.L a T').U (fun _ => j)), (pm.appLE ((LPow t.G t.PP.L a T).U (fun _ => i)) W h₁).hom (s.f (fun _ => i)) = ((snd T' t'.G).left.appLE ((LB.div (t.PP.L.pull ν.cm.F) t'.PP.L).U ((fun _ => i), j)) W h₀).hom (Units.val (w.s ((fun _ => i), j))) ^ k ∗ ((T' ⊗ t'.G).left.presheaf.map (homOfLE h₂).op).hom (s'.f (fun _ => j))) (Z : Scheme.{0}) (r : Z ⟶ (T' ⊗ t'.G).left) (z : Z) (J J') (hJ : Good t.PP.L z J) (hJ' : Good t'.PP.L z J

theorem **const_ad** {d : ℕ} (B : Germ.FB S d (𝒪 E)) (a : 𝒪 E) : MvPowerSeries.constantCoeff (B.ad a) = 1

theorem **exists_chart_push** (hG : GlobalFnsSt) {ρ : Xo.X ⟶ Xo.X} (ν : SABMap Xo.univ.t Xo.univ.t ρ) {y y' : FrPt Xo c S} {pm : Gx y.x ⟶ Gx y'.x} {w : LB.Iso Xo.univ.t.PP.L (Xo.univ.t.PP.L.pull ν.cm.F)} (h0 : y'.x = y.x ≫ ρ) (p₁ : pm ≫ (fst (Over.mk y'.x) Xo.univ.t.G).left = (fst (Over.mk y.x) Xo.univ.t.G).left) (p₂ : pm ≫ (snd (Over.mk y'.x) Xo.univ.t.G).left = (snd (Over.mk y.x) Xo.univ.t.G).left ≫ ν.φ) (hw : CubIso ν w) (hu : ∀ τ i, y'.u τ i = y.u τ i ≫ ν.φ) (C : ChartAt Xo c y) : ∃ (x : ∀ m, jO y' C.d m 1 ⟶ Xo.univ.t.G) (e : ∀ m, FrTy y' C.d x m), IsChart y' C.d C.B.G.F C.B.A.act C.uvec x ∧ IsFrame y' C.d x C.B.G.F C.B.A.act C.B.ph1 C.B.ph2 C.B.sy C.B.ad e ∧ LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L _ _ (eL y' C.d x C.B.A.act e C.wP 0) (homAt y'.x (jetAug : Jet S (Fin 1 × Fin C.d) 0 →ₐ[S] S)).left) (rigT Xo.univ.t.G Xo.univ.t.PP.L Xo.univ.t.PP.rig (overAt y'.x S) 1)

theorem **chartVal_smul** {y : FrPt Xo c S} (C : ChartAt Xo c y) (k : ℕ) {n : ℕ} (f : Fin n → MvPowerSeries (Fin 1 × Fin C.d) S) (P : MvPolynomial (LtN Dw n) c.F) (a : c.F) (h : (a • P).support = P.support) : chartVal C k f (a • P) = algebraMap c.F S a ∗ chartVal C k f P

theorem **nab_push** (hG : GlobalFnsSt) (hC : ChartIndepSt2) {nab : NabV Xo c} (hn : IsNab Xo c nab) {ρ : Xo.X ⟶ Xo.X} (ν : SABMap Xo.univ.t Xo.univ.t ρ) {y y' : FrPt Xo c S} {pm : Gx y.x ⟶ Gx y'.x} {w : LB.Iso Xo.univ.t.PP.L (Xo.univ.t.PP.L.pull ν.cm.F)} (h0 : y'.x = y.x ≫ ρ) (p₁ : pm ≫ (fst (Over.mk y'.x) Xo.univ.t.G).left = (fst (Over.mk y.x) Xo.univ.t.G).left) (p₂ : pm ≫ (snd (Over.mk y'.x) Xo.univ.t.G).left = (snd (Over.mk y.x) Xo.univ.t.G).left ≫ ν.φ) (hw : CubIso ν w) (hu : ∀ τ i, y'.u τ i = y.u τ i ≫ ν.φ) {n : ℕ} {a : (E →+* ℂ) × ℕ → ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) (k : ℕ) (b : ℤ) (hb : b = k) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L b (ptO Xo y.x))) (F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L b (ptO Xo y'.x))) (hF : ∀ i (Z : Scheme.{0}) (r : Z ⟶ Gx y.x) (z : Z) J J', Go

theorem **nab_harmonic_invariant** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (ρ : Xo.X ⟶ Xo.X) (ν : SABMap Xo.univ.t Xo.univ.t ρ) (y y' : FrPt Xo c S) (pm : Gx y.x ⟶ Gx y'.x) (w : LB.Iso Xo.univ.t.PP.L (Xo.univ.t.PP.L.pull ν.cm.F)), y'.x = y.x ≫ ρ → pm ≫ (fst (Over.mk y'.x) Xo.univ.t.G).left = (fst (Over.mk y.x) Xo.univ.t.G).left → pm ≫ (snd (Over.mk y'.x) Xo.univ.t.G).left = (snd (Over.mk y.x) Xo.univ.t.G).left ≫ ν.φ → CubIso ν w → (∀ τ i, y'.u τ i = y.u τ i ≫ ν.φ) → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (N : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HarmonicValuesInvariantUniversalEndomorphismStatement

theorem **_root_.GlobalTheory.T.harmonic_values_invariant_universal_endomorphism** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (ρ : Xo.X ⟶ Xo.X) (ν : SABMap Xo.univ.t Xo.univ.t ρ) (y y' : FrPt Xo c S) (pm : Gx y.x ⟶ Gx y'.x) (w : LB.Iso Xo.univ.t.PP.L (Xo.univ.t.PP.L.pull ν.cm.F)), y'.x = y.x ≫ ρ → pm ≫ (fst (Over.mk y'.x) Xo.univ.t.G).left = (fst (Over.mk y.x) Xo.univ.t.G).left → pm ≫ (snd (Over.mk y'.x) Xo.univ.t.G).left = (snd (Over.mk y.x) Xo.univ.t.G).left ≫ ν.φ → CubIso ν w → (∀ τ i, y'.u τ i = y.u τ i ≫ ν.φ) → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (N : ℕ) (F : Fin n → LB

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HasAnConstThetaGrpZeroIff

theorem **exAt_mem_twist_chart_iff** {m : T ⊗ A ⟶ A} {z w : Fin g → ℂ} (hw : exAt u p z ≫ m.left = (u.ex w).left) (j : Fin 2 → L.ι) : ⊤ ≤ exAt u p z ⁻¹ᵁ LB.wchart L ![(snd T A).left, m.left] j ↔ ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U (j 0) ∧ ⊤ ≤ (u.ex w).left ⁻¹ᵁ L.U (j 1)

private theorem **exists_algChart_mem_source** (hL : L.IsLB) (f : Sp ℂ ⟶ A.left) : ∃ i, ⊤ ≤ f ⁻¹ᵁ L.U i

private theorem **valAt'_unit_ne_zero_cusp_form_evaluations** {X : Scheme.{0}} (y : Sp ℂ ⟶ X) (U : X.Opens) (h : ⊤ ≤ y ⁻¹ᵁ U) (f : Γ(X, U)ˣ) : valAt' y U f.val ≠ 0

private theorem **valAt'_mul_of_top_le_preimage_open** {X : Scheme.{0}} (y : Sp ℂ ⟶ X) (U : X.Opens) (a b : Γ(X, U)) : valAt' y U (a * b) = valAt' y U a * valAt' y U b

private theorem **valAt'_resG** {X : Scheme.{0}} (y : Sp ℂ ⟶ X) {U V : X.Opens} (hVU : V ≤ U) (h : ⊤ ≤ y ⁻¹ᵁ V) (f : Γ(X, U)) : valAt' y V (X.presheaf.map (homOfLE hVU).op f) = valAt' y U f

private theorem **valAt'_app_eq_valAt'_comp** {X Y : Scheme.{0}} (y : Sp ℂ ⟶ X) (φ : X ⟶ Y) (U : Y.Opens) (f : Γ(Y, U)) : valAt' y (φ ⁻¹ᵁ U) (φ.app U f) = valAt' (y ≫ φ) U f

private theorem **exAt_eq_ex_comp_lift** (z : Fin g → ℂ) : exAt u p z = (u.ex z).left ≫ (lift (toUnit A ≫ p) (𝟙 A)).left

private theorem **exists_ringHom_eq_valAt'** {X : Scheme.{0}} (y : Sp ℂ ⟶ X) (U : X.Opens) (h : ⊤ ≤ y ⁻¹ᵁ U) : ∃ φ : Γ(X, U) →+* ℂ, ∀ f, φ f = valAt' y U f

private theorem **preimage_inf_eq_univ** {X : Scheme.{0}} (y : Sp ℂ ⟶ X) {U V : X.Opens} (h1 : ⊤ ≤ y ⁻¹ᵁ U) (h2 : ⊤ ≤ y ⁻¹ᵁ V) : ⊤ ≤ y ⁻¹ᵁ (U ⊓ V)

private theorem **mul_zpow_div_zpow_eq_of_factor** {v v' G0 G1 f0 f1 f0' f1' : ℂ} (a b : ℤ) (hG0 : G0 ≠ 0) (hG1 : G1 ≠ 0) (hf0' : f0' ≠ 0) (hv : v = G0 ^ a * G1 ^ (-b) * v') (h0 : f0 = G0 * f0') (h1 : f1 = G1 * f1') : v * f1 ^ b / f0 ^ a = v' * f1' ^ b / f0' ^ a

private theorem **ringHom_prod_units_zpow** {R : Type} [CommRing R] (φ : R →+* ℂ) {n : ℕ} (v : Fin n → Rˣ) (e : Fin n → ℤ) : φ ((∏ k, v k ^ e k : Rˣ) : R) = ∏ k, (φ (v k : R)) ^ e k

private theorem **valAt'_res_pullU_eq_valAt'_comp** {X Y : Scheme.{0}} (y : Sp ℂ ⟶ Y) (ψ : Y ⟶ X) (V : X.Opens) {W : Y.Opens} (h : W ≤ ψ ⁻¹ᵁ V) (hy : ⊤ ≤ y ⁻¹ᵁ W) (f : Γ(X, V)ˣ) : valAt' y W (res h (pullU ψ V f)).val = valAt' (y ≫ ψ) V f.val

private theorem **valAt'_res_eq** {X : Scheme.{0}} (y : Sp ℂ ⟶ X) {U V : X.Opens} (h : V ≤ U) (hy : ⊤ ≤ y ⁻¹ᵁ V) (f : Γ(X, U)ˣ) : valAt' y V (res h f).val = valAt' y U f.val

private theorem **valAt'_linComb_g_eq_prod_zpow** {X Y : Scheme.{0}} (L : LB X) (F : Fin 2 → (Y ⟶ X)) (ε : Fin 2 → ℤ) (i j : (LB.LinComb L F ε).ι) (y : Sp ℂ ⟶ Y) (hy : ⊤ ≤ y ⁻¹ᵁ ((LB.LinComb L F ε).U i ⊓ (LB.LinComb L F ε).U j)) : valAt' y ((LB.LinComb L F ε).U i ⊓ (LB.LinComb L F ε).U j) ((LB.LinComb L F ε).g i j).val = ∏ k, valAt' (y ≫ F k) (L.U (i k) ⊓ L.U (j k)) (L.g (i k) (j k)).val ^ ε k

private theorem **div_zpow_eq_mul_zpow_neg** (x y : ℂ) (a : ℤ) : x / y ^ a = x * y ^ (-a)

theorem **exists_entire_twist** (hL : L.IsLB) {m : T ⊗ A ⟶ A} {a b : ℤ} (α : LB.Gen (LB.LinComb L (K

theorem **exAt_translPt** {y : T ⟶ A} {x : Fin g → ℂ} (hy : p ≫ y = u.ex x) (z : Fin g → ℂ) : exAt u p z ≫ (translPt A y).left = (u.ex (z + x)).left

theorem **exAt_translT** {y : T ⟶ A} {x : Fin g → ℂ} (hy : p ≫ y = u.ex x) (z : Fin g → ℂ) : exAt u p z ≫ (translT A y).left = exAt u p (z + x)

theorem **fockFactor_ne_zero** (Hm : Matrix (Fin g) (Fin g) ℂ) (k : ℤ) (x z : Fin g → ℂ) : fockFactor Hm k x z ≠ 0

theorem **HasAnConst.unique** (hL : L.IsLB) {c c' : ℂ} (h : HasAnConst u p Hm θ x c) (h' : HasAnConst u p Hm θ x c') : c = c'

theorem **HasAnConst.ne_zero** (hL : L.IsLB) {c : ℂ} (h : HasAnConst u p Hm θ x c) : c ≠ 0

private theorem **exAt_preimage_wchart_eq_univ** (z : Fin g → ℂ) (i : L.ι) (h : ⊤ ≤ (u.ex z).left ⁻¹ᵁ L.U i) : ⊤ ≤ exAt u p z ⁻¹ᵁ LB.wchart L (fun _ : Fin 1 => (snd T A).left) (fun _ => i)

theorem **HasAnConst.act** (hL : L.IsLB) {c : ℂ} (h : HasAnConst u p Hm θ x c) {s s' : LB.Sec (LPow A L k T)} (hs : IsActOf A L k θ s s') {θ θ' : (Fin g → ℂ) → ℂ} (hθ : IsFnAt u p k s θ) (hθ' : IsFnAt u p k s' θ') (z : Fin g → ℂ) : θ' z = c * fockFactor Hm k x z * θ (z + x)

theorem **exists_int_of_exp_eq_one** {t : ℝ} (h : Complex.exp (-(2 * Real.pi * Complex.I * t)) = 1) : ∃ n : ℤ, t = n

private theorem **mul_zpow_eq_of_two_zpow_relations** {V r r' q a0 f0 f1 W : ℂ} (k : ℤ) (hq : q ≠ 0) (ha : a0 ≠ 0) (hf : f0 ≠ 0) (e1 : V * (q * (a0 * W) * f1) ^ k = r * (q * a0 * f0) ^ k) (e2 : V * f1 ^ k = r' * f0 ^ k) : r' * W ^ k = r

private theorem **complex_star_eq_sub_two_I_mul_im** (w : ℂ) : star w = w - 2 * Complex.I * (w.im : ℂ)

theorem **exists_hasAnConst** (hL : L.IsLB) {χ : u.Λ → ℂ} (hAH : u.IsAH Hm χ) (hx : p ≫ θ.pt = u.ex x) : (∀ l ∈ u.Λ, ∃ n : ℤ, k * (hermForm Hm x l).im = n) ∧ ∃ c, HasAnConst u p Hm θ x c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HasSumPolyDerivMulProdAddMatrix

theorem **hasSum_polyDeriv_of_locally_normal** {ι : Type*} {F : ι → W → ℂ} {U : Set W} (hU : IsOpen U) (hF : ∀ i, DifferentiableOn ℂ (F i) U) (hle : ∀ x ∈ U, ∃ t ∈ 𝒩 x, ∃ v : ι → ℝ, Summable v ∧ ∀ i, ∀ z ∈ t, ‖F i z‖ ≤ v i) (P : MvPolynomial σ ℂ) (u : σ → W) {x : W} (hx : x ∈ U) : HasSum (fun i => I2.polyDeriv P u (F i) x) (I2.polyDeriv P u (fun w => ∑' i, F i w) x)

theorem **polyDeriv_congr_of_eventuallyEq** (P : MvPolynomial σ ℂ) (u : σ → W) {F G : W → ℂ} {x : W} (h : F =ᶠ[𝒩 x] G) : I2.polyDeriv P u F x = I2.polyDeriv P u G x

theorem **polyDeriv_const_mul** (P : MvPolynomial σ ℂ) (u : σ → W) {F : W → ℂ} {U : Set W} (hU : IsOpen U) (hF : DifferentiableOn ℂ F U) (c : ℂ) {x : W} (hx : x ∈ U) : I2.polyDeriv P u (fun w => c * F w) x = c * I2.polyDeriv P u F x

theorem **iteratedFDeriv_snd** {F : B × W → ℂ} {U : Set (B × W)} (hU : IsOpen U) (hF : DifferentiableOn ℂ F U) {s : B} {x : W} (hx : (s, x) ∈ U) (k : ℕ) (v : Fin k → W) : iteratedFDeriv ℂ k (fun w => F (s, w)) x v = iteratedFDeriv ℂ k F (s, x) (fun j => (0, v j))

theorem **differentiableOn_polyDeriv_snd** {F : B × W → ℂ} {U : Set (B × W)} (hU : IsOpen U) (hF : DifferentiableOn ℂ F U) (P : MvPolynomial σ ℂ) (u : σ → W) : DifferentiableOn ℂ (fun p : B × W => I2.polyDeriv P u (fun w => F (p.1, w)) p.2) U

theorem **differentiableOn_polyDeriv_param** {F : B × W → ℂ} {V : Set B} (hV : IsOpen V) (hF : DifferentiableOn ℂ F (V ×ˢ Set.univ)) (P : MvPolynomial σ ℂ) (u : σ → W) (x : W) : DifferentiableOn ℂ (fun s : B => I2.polyDeriv P u (fun w => F (s, w)) x) V

theorem **thetaChar_eq_tsum** (τ : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g → ℝ) (ζ : Fin g → ℂ) : thetaChar τ (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) ζ = ∑' m, thTerm τ a b ζ m

theorem **thTerm_eq_thetaTerm** (τ : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g → ℝ) (ζ : Fin g → ℂ) (m : Fin g → ℤ) : thTerm τ a b ζ m = thetaTerm a b ζ τ m

theorem **thetaChar_eq_theta** (τ : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g → ℝ) (ζ : Fin g → ℂ) : thetaChar τ (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) ζ = theta a b ζ τ

theorem **thTerm_add_matrix** (τ S : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g → ℝ) (ζ : Fin g → ℂ) (m : Fin g → ℤ) : thTerm (τ + S) a b ζ m = cexp (π * I * (((fun i => (m i : ℂ)) + fun i => (a i : ℂ)) ⬝ᵥ S *ᵥ ((fun i => (m i : ℂ)) + fun i => (a i : ℂ)))) * thTerm τ a b ζ m

theorem **thTerm_add_var** (τ : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g → ℝ) (ζ w : Fin g → ℂ) (m : Fin g → ℤ) : thTerm τ a b (ζ + w) m = cexp (2 * π * I * (((fun i => (m i : ℂ)) + fun i => (a i : ℂ)) ⬝ᵥ w)) * thTerm τ a b ζ m

theorem **mem_thetaDomain** {τ : Matrix (Fin g) (Fin g) ℂ} (hp : (τ.map Complex.im).PosDef) : τ ∈ ThetaDomain (Fin g)

theorem **summable_norm_thTerm** {τ : Matrix (Fin g) (Fin g) ℂ} (hτ : τ ∈ ThetaDomain (Fin g)) (a b : Fin g → ℝ) (ζ : Fin g → ℂ) : Summable fun m => ‖thTerm τ a b ζ m‖

theorem **hasSum_thTerm** {τ : Matrix (Fin g) (Fin g) ℂ} (hτ : τ ∈ ThetaDomain (Fin g)) (a b : Fin g → ℝ) (ζ : Fin g → ℂ) : HasSum (thTerm τ a b ζ) (thetaChar τ (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) ζ)

theorem **hasSum_prod_thTerm** {κ : Type*} [Fintype κ] {τ : κ → Matrix (Fin g) (Fin g) ℂ} (hτ : ∀ k, τ k ∈ ThetaDomain (Fin g)) (a b : κ → Fin g → ℝ) (ζ : κ → Fin g → ℂ) : HasSum (fun m : κ → Fin g → ℤ => ∏ k, thTerm (τ k) (a k) (b k) (ζ k) (m k)) (∏ k, thetaChar (τ k) (fun i => (a k i : ℂ)) (fun i => (b k i : ℂ)) (ζ k))

theorem **thTerm_locally_normal** {U : Set W} (hU : IsOpen U) {T : W → Matrix (Fin g) (Fin g) ℂ} {Z : W → Fin g → ℂ} (hT : ∀ i j, ContinuousOn (fun w => T w i j) U) (hZ : ContinuousOn Z U) (hdom : ∀ w ∈ U, T w ∈ ThetaDomain (Fin g)) (a b : Fin g → ℝ) {x : W} (hx : x ∈ U) : ∃ t ∈ 𝒩 x, ∃ v : (Fin g → ℤ) → ℝ, Summable v ∧ ∀ m, ∀ w ∈ t, ‖thTerm (T w) a b (Z w) m‖ ≤ v m

theorem **mul_prod_thTerm_locally_normal** {κ : Type*} [Fintype κ] {U : Set W} (hU : IsOpen U) {G : W → ℂ} {T : κ → W → Matrix (Fin g) (Fin g) ℂ} {Z : κ → W → Fin g → ℂ} (hG : ContinuousOn G U) (hT : ∀ k i j, ContinuousOn (fun w => T k w i j) U) (hZ : ∀ k, ContinuousOn (Z k) U) (hdom : ∀ k, ∀ w ∈ U, T k w ∈ ThetaDomain (Fin g)) (a b : κ → Fin g → ℝ) {x : W} (hx : x ∈ U) : ∃ t ∈ 𝒩 x, ∃ v : (κ → Fin g → ℤ) → ℝ, Summable v ∧ ∀ m, ∀ w ∈ t, ‖G w * ∏ k, thTerm (T k w) (a k) (b k) (Z k w) (m k)‖ ≤ v m

theorem **differentiableOn_prod_fun** {κ : Type*} (s : Finset κ) {F : κ → W → ℂ} {U : Set W} (hF : ∀ k ∈ s, DifferentiableOn ℂ (F k) U) : DifferentiableOn ℂ (fun w => ∏ k ∈ s, F k w) U

theorem **differentiableOn_thTerm** {U : Set W} {T : W → Matrix (Fin g) (Fin g) ℂ} {Z : W → Fin g → ℂ} (hT : ∀ i j, DifferentiableOn ℂ (fun w => T w i j) U) (hZ : DifferentiableOn ℂ Z U) (a b : Fin g → ℝ) (m : Fin g → ℤ) : DifferentiableOn ℂ (fun w => thTerm (T w) a b (Z w) m) U

theorem **differentiableOn_thetaChar** {U : Set W} {T : W → Matrix (Fin g) (Fin g) ℂ} {Z : W → Fin g → ℂ} (hT : ∀ i j, DifferentiableOn ℂ (fun w => T w i j) U) (hZ : DifferentiableOn ℂ Z U) (hdom : ∀ w ∈ U, T w ∈ ThetaDomain (Fin g)) (a b : Fin g → ℝ) : DifferentiableOn ℂ (fun w => thetaChar (T w) (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) (Z w)) U

theorem **differentiable_thetaChar** {τ : Matrix (Fin g) (Fin g) ℂ} (hτ : τ ∈ ThetaDomain (Fin g)) (a b : Fin g → ℝ) : Differentiable ℂ (thetaChar τ (fun i => (a i : ℂ)) (fun i => (b i : ℂ)))

theorem **hasSum_polyDeriv_mul_prod** {U : Set W} (hU : IsOpen U) {G : W → ℂ} {T : κ → W → Matrix (Fin g) (Fin g) ℂ} {Z : κ → W → Fin g → ℂ} (hG : DifferentiableOn ℂ G U) (hT : ∀ k i j, DifferentiableOn ℂ (fun w => T k w i j) U) (hZ : ∀ k, DifferentiableOn ℂ (Z k) U) (hdom : ∀ k, ∀ w ∈ U, T k w ∈ ThetaDomain (Fin g)) (a b : κ → Fin g → ℝ) (P : MvPolynomial σ ℂ) (u : σ → W) {x : W} (hx : x ∈ U) : HasSum (fun m : κ → Fin g → ℤ => I2.polyDeriv P u (fun w => G w * ∏ k, thTerm (T k w) (a k) (b k) (Z k w) (m k)) x) (I2.polyDeriv P u (fun w => G w * ∏ k, thetaChar (T k w) (fun i => (a k i : ℂ)) (fun i => (b k i : ℂ)) (Z k w)) x)

theorem **hasSum_polyDeriv_mul_prod_const** {G : W → ℂ} {τ : κ → Matrix (Fin g) (Fin g) ℂ} {Z : κ → W → Fin g → ℂ} (hG : Differentiable ℂ G) (hZ : ∀ k, Differentiable ℂ (Z k)) (hτ : ∀ k, τ k ∈ ThetaDomain (Fin g)) (a b : κ → Fin g → ℝ) (P : MvPolynomial σ ℂ) (u : σ → W) (x : W) : HasSum (fun m : κ → Fin g → ℤ => I2.polyDeriv P u (fun w => G w * ∏ k, thTerm (τ k) (a k) (b k) (Z k w) (m k)) x) (I2.polyDeriv P u (fun w => G w * ∏ k, thetaChar (τ k) (fun i => (a k i : ℂ)) (fun i => (b k i : ℂ)) (Z k w)) x) ∧ Summable fun m : κ → Fin g → ℤ => ‖I2.polyDeriv P u (fun w => G w * ∏ k, thTerm (τ k) (a k) (b k) (Z k w) (m k)) x‖

theorem **hasSum_polyDeriv_mul_prod_add_matrix** {G : W → ℂ} {τ S : κ → Matrix (Fin g) (Fin g) ℂ} {Z : κ → W → Fin g → ℂ} (hG : Differentiable ℂ G) (hZ : ∀ k, Differentiable ℂ (Z k)) (hτ : ∀ k, τ k + S k ∈ ThetaDomain (Fin g)) (a b : κ → Fin g → ℝ) (P : MvPolynomial σ ℂ) (u : σ → W) (x : W) : HasSum (fun m : κ → Fin g → ℤ => I2.polyDeriv P u (fun w => G w * ∏ k, thTerm (τ k) (a k) (b k) (Z k w) (m k)) x * cexp (π * I * ∑ k, ((fun i => (m k i : ℂ)) + fun i => (a k i : ℂ)) ⬝ᵥ S k *ᵥ ((fun i => (m k i : ℂ)) + fun i => (a k i : ℂ)))) (I2.polyDeriv P u (fun w => G w * ∏ k, thetaChar (τ k + S k) (fun i => (a k i : ℂ)) (fun i => (b k i : ℂ)) (Z k w)) x)

theorem **norm_translConst** (b a' b' : Fin g → ℝ) : ‖translConst b a' b'‖ = 1

theorem **imInv_mul** (hp : (τ.map Complex.im).PosDef) : imInv τ * (τ.map Complex.im).map (fun x : ℝ => (x : ℂ)) = 1

theorem **imInv_eq_inv_map** (hp : (τ.map Complex.im).PosDef) : imInv τ = ((τ.map Complex.im).map fun x : ℝ => (x : ℂ))⁻¹

theorem **isSymm_imInv** (hs : τ.IsSymm) : (imInv τ).IsSymm

theorem **dot_imInv_eq_hermForm_add** (hp : (τ.map Complex.im).PosDef) (u v : Fin g → ℝ) (ζ : Fin g → ℂ) : ζ ⬝ᵥ imInv τ *ᵥ ((fun i => (v i : ℂ)) + τ *ᵥ fun i => (u i : ℂ)) = hermForm (imInv τ) ζ ((fun i => (v i : ℂ)) + τ *ᵥ fun i => (u i : ℂ)) + 2 * I * (ζ ⬝ᵥ fun i => (u i : ℂ))

theorem **differentiable_canTh** (hp : (τ.map Complex.im).PosDef) (a b : Fin g → ℝ) : Differentiable ℂ (canTh τ a b)

theorem **thetaChar_ne_zero** (hs : τ.IsSymm) (hp : (τ.map Complex.im).PosDef) (a b : Fin g → ℝ) : thetaChar τ (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) ≠ 0

theorem **canTh_ne_zero** (hs : τ.IsSymm) (hp : (τ.map Complex.im).PosDef) (a b : Fin g → ℝ) : canTh τ a b ≠ 0

theorem **canTh_translate** (hs : τ.IsSymm) (hp : (τ.map Complex.im).PosDef) (a b a' b' : Fin g → ℝ) {y : Fin g → ℂ} (hy : y = (fun i => (b' i : ℂ)) + τ *ᵥ fun i => (a' i : ℂ)) (ζ : Fin g → ℂ) : cexp (-(π * hermForm (imInv τ) ζ y) - π / 2 * hermForm (imInv τ) y y) * canTh τ a b (ζ + y) = translConst b a' b' * canTh τ (a + a') (b + b') ζ

theorem **canTh_add_int** (τ : Matrix (Fin g) (Fin g) ℂ) (a b : Fin g → ℝ) (n m : Fin g → ℤ) (ζ : Fin g → ℂ) : canTh τ (a + fun i => (n i : ℝ)) (b + fun i => (m i : ℝ)) ζ = cexp (2 * π * I * ((a ⬝ᵥ fun i => (m i : ℝ)) : ℝ) : ℂ)) * canTh τ a b ζ

theorem **canTh_add_period** (hs : τ.IsSymm) (hp : (τ.map Complex.im).PosDef) (a b : Fin g → ℝ) (m n : Fin g → ℤ) {l : Fin g → ℂ} (hl : l = (fun i => (m i : ℂ)) + τ *ᵥ fun i => (n i : ℂ)) (ζ : Fin g → ℂ) : canTh τ a b (ζ + l) = canMult a b m n * cexp (π * hermForm (imInv τ) ζ l + π / 2 * hermForm (imInv τ) l l) * canTh τ a b ζ

theorem **im_hermForm_imInv** (hs : τ.IsSymm) (hp : (τ.map Complex.im).PosDef) (u v u' v' : Fin g → ℝ) : (hermForm (imInv τ) ((fun i => (v i : ℂ)) + τ *ᵥ fun i => (u i : ℂ)) ((fun i => (v' i : ℂ)) + τ *ᵥ fun i => (u' i : ℂ))).im = u ⬝ᵥ v' - v ⬝ᵥ u'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HermFormDiagonalEq

theorem **hermForm_diagonal_eq** (d z w : Fin g → ℂ) : hermForm (Matrix.diagonal d) z w = ∑ j, d j * z j * conj (w j)

theorem **selfD_of_isSympl** (hs : IsSympl X bas) : SelfD X X.Λ

theorem **hx_ortho** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g : ℕ) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r), IsSympl X bas → ∀ x ∈ X.dom, X.IsOrtho x → (∀ z z', hx X bas idx x z z' = ∑ j, ((X.c (idx j).1 : ℝ) : ℂ) * z j * conj (z' j)) ∧ (∀ u, (hx X bas idx x (pv X idx x u) (pv X idx x u)).re = ∑ τ : X.Ψ, X.c τ * X.maj x τ (Data.emb τ.1 u)) ∧ (∀ u ∈ X.Λ, u ≠ 0 → X.firstMinimumSq x ≤ (hx X bas idx x (pv X idx x u) (pv X idx x u)).re) ∧ IsUnitFr X x ∧ SelfD X X.Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HmEqDiagonal

private theorem **hermForm_eq_star_swap** {g : ℕ} (A : Matrix (Fin g) (Fin g) ℂ) (hA : A.IsHermitian) (z w : Fin g → ℂ) : hermForm A z w = star (hermForm A w z)

private theorem **eq_of_im_hermForm_eq_of_real_span_eq_univ** {g : ℕ} {La : Type} [AddCommGroup La] (A B : Matrix (Fin g) (Fin g) ℂ) (hA : A.IsHermitian) (hB : B.IsHermitian) (P : La →+ (Fin g → ℂ)) (hP : Submodule.span ℝ (Set.range P) = ⊤) (h : ∀ a b, (hermForm A (P a) (P b)).im = (hermForm B (P a) (P b)).im) : A = B

theorem **hm_eq_diagonal** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (W : UnfAt U k x) (Wu : UnitAt W), GoodEmb c σ → IsUnf U → (U.X k).IsOrtho x.1 → Wu.Hm = Matrix.diagonal fun j => ((2 * ‖relab c σ (W.e j).1 (U.X k).ξ0‖ : ℝ) : ℂ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HodgeWeightDominanceCondition

theorem **det_gen_ne_zero** {K : Type} [Field K] {Ix : Type} {n : ℕ} (t : Ix) : (gen K n t).det ≠ 0

theorem **detFac_ne_zero** (K : Type) [Field K] (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n) : detFac Dw pl K a s lam ≠ 0

theorem **cof_image** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n), (∀ τ, Antitone fun i => a (τ, i)) → WtIs Dw pl a s lam → (∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), ∃ F ∈ VV c.F lam, cof Dw pl c.F n P = detFac Dw pl c.F a s lam * F) ∧ (∀ F ∈ VV c.F lam, ∃! P, P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) ∧ cof Dw pl c.F n P = detFac Dw pl c.F a s lam * F) ∧ ∀ (g : Ix → GL (Fin n) c.F), ∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), ∀ F ∈ VV c.F lam, cof Dw pl c.F n P = detFac Dw pl c.F a s lam * F → cof Dw pl c.F n (le

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HodgeWeightDominanceConditionStatement

theorem **_root_.GlobalTheory.T.hodge_weight_dominance_condition** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n), (∀ τ, Antitone fun i => a (τ, i)) → WtIs Dw pl a s lam → (∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), ∃ F ∈ VV c.F lam, cof Dw pl c.F n P = detFac Dw pl c.F a s lam * F) ∧ (∀ F ∈ VV c.F lam, ∃! P, P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) ∧ cof Dw pl c.F n P = detFac Dw pl c.F a s lam * F) ∧ ∀ (g : Ix → GL (Fin n) c.F), ∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), ∀ F ∈ VV c.F lam, cof Dw pl c.F n P = detFa

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.HyperbolicFlagDataBasePoint

theorem **ii_prim** : IsPrimitiveRoot ii 4

theorem **ii_mul_ii** : ii * ii = -1

theorem **ii_ne_zero** : ii ≠ 0

theorem **ii_isIntegral** : IsIntegral ℤ ii

theorem **emb_ii** (τ : E4 →+* ℂ) : τ ii = Complex.I ∨ τ ii = -Complex.I

theorem **cc_ii** : cc ii = -ii

theorem **exists_int_of_isIntegral** {x : E4} (hx : IsIntegral ℤ x) : ∃ m n : ℤ, x = m + n * ii

theorem **exists_rat** (x : E4) : ∃ p q : ℚ, x = p + q * ii

theorem **trace_ii** : Algebra.trace ℚ E4 ii = 0

theorem **finrank_eq** : Module.finrank ℚ E4 = 2

theorem **trace_eq** (p q : ℚ) : Algebra.trace ℚ E4 (-(ii / 2) * (p + q * ii)) = q

theorem **trace_int** {t : E4} (ht : IsIntegral ℤ t) : ∃ k : ℤ, Algebra.trace ℚ E4 (-(ii / 2) * t) = k

theorem **isIntegral_of_trace** {t : E4} (h : ∀ a : E4, IsIntegral ℤ a → ∃ k : ℤ, Algebra.trace ℚ E4 (-(ii / 2) * (t * a)) = k) : IsIntegral ℤ t

theorem **herm_gm** (x y : Fin 2 → E4) : herm gm x y = x 0 * cc (y 0) - x 1 * cc (y 1)

theorem **gm_herm** (a b : Fin 2) : cc (gm a b) = gm b a

theorem **gm_det** : gm.det ≠ 0

theorem **mem_lat** {x : Fin 2 → E4} : x ∈ lat ↔ ∀ a, IsIntegral ℤ (x a)

theorem **single_mem_lat** (a : Fin 2) {c : E4} (hc : IsIntegral ℤ c) : Pi.single a c ∈ lat

theorem **lat_fg** : lat.FG

theorem **lat_span** : Submodule.span E4 (lat : Set (Fin 2 → E4)) = ⊤

theorem **lat_int** {x y : Fin 2 → E4} (hx : x ∈ lat) (hy : y ∈ lat) : IsIntegral ℤ (herm gm x y)

theorem **lat_dual** (x : Fin 2 → E4) (h : ∀ y ∈ lat, IsIntegral ℤ (herm gm x y)) : x ∈ lat

theorem **traceFormDual_lat** : traceFormDual (ii / 2) gm lat = lat

theorem **mem_cmt** {τ : E4 →+* ℂ} : τ ∈ cmt ↔ τ ii = Complex.I

theorem **cmt_cmType** (τ : E4 →+* ℂ) : τ ∈ cmt ↔ ComplexEmbedding.conjugate τ ∉ cmt

theorem **gram_one** (τ : E4 →+* ℂ) : gram τ gm 1 = Matrix.diagonal fun a : Fin 2 => ((sgn 1 (a : ℕ) : ℝ) : ℂ)

theorem **half_ii_ne_zero** : ii / 2 ≠ 0

theorem **exists_flagData** : ∃ X : Data E4 2, X.Ψ = cmt ∧ X.Gm = gm ∧ X.p0 = 1 ∧ X.Λ = lat ∧ X.ξ0 = ii / 2 ∧ X.κ0 = -ii ∧ X.m = 3 ∧ (∀ γ, γ ∈ X.Γ ↔ IsUStab gm lat γ ∧ (∀ v ∈ lat, ∃ w ∈ lat, Matrix.mulVec (γ : Matrix (Fin 2) (Fin 2) E4) v = v + 3 • w)) ∧ ∀ la γ, X.χ la γ = 1

theorem **flagData_Ψ** : flagData.Ψ = cmt

theorem **flagData_Gm** : flagData.Gm = gm

theorem **hyperbolicFlagData_base_point** : flagData.p0 = 1

theorem **flagData_Λ** : flagData.Λ = lat

theorem **flagData_ξ0** : flagData.ξ0 = ii / 2

theorem **flagData_κ0** : flagData.κ0 = -ii

theorem **flagData_m** : flagData.m = 3

theorem **mem_flagData_Γ** (γ : GL (Fin 2) E4) : γ ∈ flagData.Γ ↔ IsUStab gm lat γ ∧ (∀ v ∈ lat, ∃ w ∈ lat, Matrix.mulVec (γ : Matrix (Fin 2) (Fin 2) E4) v = v + 3 • w)

theorem **flagData_χ** (la : Wt flagData.Ψ 2) (γ : GL (Fin 2) E4) : flagData.χ la γ = 1

theorem **emb_ii_eq** (τ : flagData.Ψ) : τ.1 ii = Complex.I

theorem **norm_ξ0** (τ : flagData.Ψ) : ‖τ.1 flagData.ξ0‖ = 1 / 2

theorem **perfect** : (∀ x ∈ flagData.Λ, ∀ y ∈ flagData.Λ, ∃ k : ℤ, EFf flagData.ξ0 flagData.Gm x y = k) ∧ ∀ φ : flagData.Λ →+ ℤ, ∃ x ∈ flagData.Λ, ∀ y : flagData.Λ, (φ y : ℚ) = EFf flagData.ξ0 flagData.Gm x y

theorem **dom_nonempty** : flagData.dom.Nonempty

theorem **isIso_one** : flagData.IsIso ![1, 1]

theorem **map_transvMat** (v : Fin 2 → E4) (b : E4) (τ : flagData.Ψ) : (transvMat flagData.Gm v ((flagData.m : E4) * b * flagData.κ0)).map τ.1 = flagData.flowMat v τ (((3 : ℝ) : ℂ) * τ.1 b)

theorem **exists_cuspData_hyperbolic** : ∃ (X : DataX E4 2) (M : ℝ), X.p0 = 1 ∧ X.dom.Nonempty ∧ (∃ ξ, X.IsIso ξ) ∧ 0 < M ∧ X.Λ = X.L' ∧ (∀ la, ∀ v ∈ X.L', herm X.Gm v v = 0 → ∀ b : E4, cc b = b → IsIntegral ℤ b → ∃ γ ∈ X.Γ, X.χ la γ = 1 ∧ ∀ τ : X.Ψ, (γ : Matrix (Fin 2) (Fin 2) E4).map τ.1 = X.flowMat v τ ((M : ℂ) * τ.1 b))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.Imports

theorem **main** : ∀ {R : Type u_1} [inst : CommRing R] {P : Ideal R}, Ideal.IsPrime P → ∀ (N : ℕ), N ≠ (0 : ℕ) → (↑N : R) ∈ P → ∃ (p : ℕ), Nat.Prime p ∧ p ∣ N ∧ (↑p : R) ∈ P

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ImportsAlt

theorem **main** : ∀ (s : Multiset ℂ) {t : ℝ}, (0 : ℝ) ≤ t → (∀ μ ∈ s, ‖μ - (1 : ℂ)‖ ≤ t) → ‖Multiset.prod s - (1 : ℂ)‖ ≤ Real.exp ((↑(Multiset.card s) : ℝ) * t) - (1 : ℝ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InBasisIffPointValuesLinearIndependentSpan

theorem **SameLin.symm** {W : Q → Prop} {f : Q → V} {g : Q → V'} (h : SameLin K K' W f g) : SameLin K' K W g f

theorem **SameLin.indep_mod** {W : Q → Prop} {f : Q → V} {g : Q → V'} (h : SameLin K K' W f g) {n : ℕ} {ι : Type} [Fintype ι] (a : Fin n → Q) (b : ι → Q) (ha : ∀ i, W (a i)) (hb : ∀ i, W (b i)) (hi : ∀ c : Fin n → K, ∑ i, c i • f (a i) ∈ Submodule.span K (Set.range fun i => f (b i)) → ∀ i, c i = 0) (c : Fin n → K') (hc : ∑ i, c i • g (a i) ∈ Submodule.span K' (Set.range fun i => g (b i))) : ∀ i, c i = 0

theorem **sub_mem_iff** {n : ℕ} {ι : Type} [Fintype ι] (x : V) (a : Fin n → V) (b : ι → V) : (∃ c : Fin n → K, x - ∑ i, c i • a i ∈ Submodule.span K (Set.range b)) ↔ x ∈ Submodule.span K (Set.range (Sum.elim a b))

theorem **SameLin.mem_mod** {W : Q → Prop} {f : Q → V} {g : Q → V'} (h : SameLin K K' W f g) {n : ℕ} {ι : Type} [Fintype ι] (q : Q) (a : Fin n → Q) (b : ι → Q) (hq : W q) (ha : ∀ i, W (a i)) (hb : ∀ i, W (b i)) : (∃ c : Fin n → K, f q - ∑ i, c i • f (a i) ∈ Submodule.span K (Set.range fun i => f (b i))) ↔ (∃ c : Fin n → K', g q - ∑ i, c i • g (a i) ∈ Submodule.span K' (Set.range fun i => g (b i)))

theorem **eq_single_of_sum_eq_one** {d : ℕ} {α : Fin d → ℕ} (h : ∑ i, α i = 1) : ∃ i, α = Pi.single i 1

theorem **prod_pow_single** {A : Type} [CommRing A] {d : ℕ} (s : Fin d → A) (i : Fin d) : ∏ k, s k ^ (Pi.single i 1 : Fin d → ℕ) k = s i

theorem **mon_mul_mem** {A : Type} [CommRing A] [Algebra K A] {d n : ℕ} (s : Fin d → A) (e : Fin n → A) {α : Fin d → ℕ} (h : ∑ i, α i = 1) {f : A} (hf : f ∈ Submodule.span K (Set.range e)) : (∏ k, s k ^ α k) * f ∈ Submodule.span K (Set.range fun p : Fin d × Fin n => s p.1 * e p.2)

theorem **sameLin** (hg : R.Generic t) (u : WeightFamilies.Wt D j) : SameLin (Kp X la N) ℂ (OfWt X u) (toRp X la N) (R.toFn t)

theorem **ofWt_pow_mul** (u : WeightFamilies.Wt D j) (k n : ℕ) (q : MvPolynomial (Idx D j N) X.Forms) (hq : OfWt X (addL u wL k) q) : OfWt X (addL u wL (k + n)) (secPoly X w 0 ^ n * q)

theorem **exists_family** (u : WeightFamilies.Wt D j) (m : ℕ) : ∃ (n : ℕ) (b : Fin (WeightFamilies.dimY D j) × Fin n → MvPolynomial (Idx D j N) X.Forms), (∀ i, OfWt X (addL u wL (m + 1)) (b i)) ∧ Submodule.span (Kp X la N) (Set.range fun i => toRp X la N (b i)) = Mon X la w u 1 m ∧ Submodule.span ℂ (Set.range fun i => R.toFn t (b i)) = R.deep w t (addL u wL m) 1

theorem **indep_transfer** (hg : R.Generic t) (u : WeightFamilies.Wt D j) {n : ℕ} (θ : Fin n → B.Cusp j u) (m : ℕ) : (∀ c : Fin n → Kp X la N, sec X la w 0 ^ (m + 1) * ∑ i, c i • emb X la N (X.inc u (θ i)) ∈ Mon X la w u 1 m → ∀ i, c i = 0) ↔ (∀ c : Fin n → ℂ, R.s w t 0 ^ (m + 1) * ∑ i, c i • R.ev (X.inc u (θ i)) ∈ R.deep w t (addL u wL m) 1 → ∀ i, c i = 0)

theorem **mem_transfer** (hg : R.Generic t) (u : WeightFamilies.Wt D j) {n : ℕ} (θ : Fin n → B.Cusp j u) (k m : ℕ) (F : X.Forms) (hF : OfWt X (addL u wL k) (MvPolynomial.C F : MvPolynomial (Idx D j N) X.Forms)) : (∃ c : Fin n → Kp X la N, sec X la w 0 ^ (m + 1) * (emb X la N F - sec X la w 0 ^ k * ∑ i, c i • emb X la N (X.inc u (θ i))) ∈ Mon X la w u 1 (k + m)) ↔ (∃ c : Fin n → ℂ, R.s w t 0 ^ (m + 1) * (R.ev F - R.s w t 0 ^ k * ∑ i, c i • R.ev (X.inc u (θ i))) ∈ R.deep w t (addL u wL (k + m)) 1)

theorem **indep_iff** (u : WeightFamilies.Wt D j) {n : ℕ} (θ : Fin n → B.Cusp j u) : LinearIndependent (Kp X la N) (fun i => rho0 X la w u (emb X la N (X.inc u (θ i)), emb_mem_Vw X la N u _)) ↔ ∀ (m : ℕ) (c : Fin n → Kp X la N), sec X la w 0 ^ (m + 1) * ∑ i, c i • emb X la N (X.inc u (θ i)) ∈ Mon X la w u 1 m → ∀ i, c i = 0

theorem **class_eq_zero_iff** (u : WeightFamilies.Wt D j) (k : ℕ) (f : ɪ(Vw X la N (addL u wL k))) : toA0K X la w u (toHK X la w u k f) = 0 ↔ ∃ m : ℕ, sec X la w 0 ^ (m + 1) * f ∈ Mon X la w u 1 (k + m)

theorem **pow_mul_mem** (u : WeightFamilies.Wt D j) (k : ℕ) (g : ɪ(Vw X la N u)) : sec X la w 0 ^ k * g ∈ Vw X la N (addL u wL k)

theorem **class_pow_mul** (u : WeightFamilies.Wt D j) (k : ℕ) (g : ɪ(Vw X la N u)) : toA0K X la w u (toHK X la w u k ⟨sec X la w 0 ^ k * g, pow_mul_mem X la w u k g⟩) = rho0 X la w u g

theorem **class_mem_iff** (u : WeightFamilies.Wt D j) {n : ℕ} (θ : Fin n → B.Cusp j u) (k : ℕ) (f : ɪ(Vw X la N (addL u wL k))) : toA0K X la w u (toHK X la w u k f) ∈ Submodule.span (Kp X la N) (Set.range fun i => rho0 X la w u (emb X la N (X.inc u (θ i)), emb_mem_Vw X la N u _)) ↔ ∃ (m : ℕ) (c : Fin n → Kp X la N), sec X la w 0 ^ (m + 1) * (f - sec X la w 0 ^ k * ∑ i, c i • emb X la N (X.inc u (θ i))) ∈ Mon X la w u 1 (k + m)

theorem **span_top_iff** (u : WeightFamilies.Wt D j) {n : ℕ} (θ : Fin n → B.Cusp j u) : Submodule.span (Kp X la N) (Set.range fun i => rho0 X la w u (emb X la N (X.inc u (θ i)), emb_mem_Vw X la N u _)) = ⊤ ↔ ∀ (k : ℕ) (F : B.Cusp j (addL u wL k)), ∃ (m : ℕ) (c : Fin n → Kp X la N), sec X la w 0 ^ (m + 1) * (emb X la N (X.inc _ F) - sec X la w 0 ^ k * ∑ i, c i • emb X la N (X.inc u (θ i))) ∈ Mon X la w u 1 (k + m)

theorem **deep_vanish** (u : WeightFamilies.Wt D j) {y : R.Y} (hy : ∀ i : Fin (WeightFamilies.dimY D j), R.s w t i.succ y = 0) {g : R.Y → ℂ} (hg : g ∈ R.deep w t u 1) : g y = 0

theorem **s_pow_mul_mem** (u : WeightFamilies.Wt D j) (k : ℕ) {g : R.Y → ℂ} (hg : g ∈ R.sp u Set.univ) : R.s w t 0 ^ k * g ∈ R.sp (addL u wL k) Set.univ

theorem **sum_mem_sp** (u : WeightFamilies.Wt D j) {n : ℕ} (θ : Fin n → B.Cusp j u) (c : Fin n → ℂ) : ∑ i, c i • R.ev (X.inc u (θ i)) ∈ R.sp u Set.univ

theorem **inBasis_iff_vals** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (G : GlobalFamilyData X ka wL wF) (R : Rdg X) (t : Idx D j G.N → ℂ) (T : ℕ) (Mg : Submodule B.KO (B.Cusp j ka)) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t → AnFacts R G.w t ka wF T Mg xs → addL wF wL T = ka → ∀ la, InBasis G la ↔ (LinearIndependent ℂ (vals xs fun i => R.ev (X.inc wF (G.th' i))) ∧ G.del = δ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InBasisIffPointValuesLinearIndependentSpanStatement

theorem **_root_.GlobalTheory.T.inBasis_iff_pointValues_linearIndependent_span** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (G : GlobalFamilyData X ka wL wF) (R : Rdg X) (t : Idx D j G.N → ℂ) (T : ℕ) (Mg : Submodule B.KO (B.Cusp j ka)) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t → AnFacts R G.w t ka wF T Mg xs → addL wF wL T = ka → ∀ la, InBasis G la ↔ (LinearIndependent ℂ (vals xs fun i => R.ev (X.inc wF (G.th' i))) ∧ G.del = δ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InContainMROfAnFacts

theorem **addL_zero** (u wL : WeightFamilies.Wt D j) : addL u wL 0 = u

theorem **ofWt_pow_mul** (hs : ∀ i, OfWt X wL (secP X w i)) (hm : ∀ u u' (p q : MvPolynomial (Idx D j N) X.Forms), OfWt X u p → OfWt X u' q → OfWt X (wadd u u') (p * q)) (i : Fin (WeightFamilies.dimY D j + 1)) (u : WeightFamilies.Wt D j) (n m : ℕ) {p : MvPolynomial (Idx D j N) X.Forms} (hp : OfWt X (addL u wL n) p) : OfWt X (addL u wL (n + m)) (secP X w i ^ m * p)

theorem **ofWt_prod_mul** (hs : ∀ i, OfWt X wL (secP X w i)) (hm : ∀ u u' (p q : MvPolynomial (Idx D j N) X.Forms), OfWt X u p → OfWt X u' q → OfWt X (wadd u u') (p * q)) (α : Fin (WeightFamilies.dimY D j) → ℕ) (u : WeightFamilies.Wt D j) (n : ℕ) {p : MvPolynomial (Idx D j N) X.Forms} (hp : OfWt X (addL u wL n) p) (s : Finset (Fin (WeightFamilies.dimY D j))) : OfWt X (addL u wL (n + ∑ i ∈ s, α i)) ((∏ i ∈ s, secP X w i.succ ^ α i) * p)

theorem **inContainMR_of_anFacts** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (G : GlobalFamilyData X ka wL wF) (R : Rdg X) (t : Idx D j G.N → ℂ) (T : ℕ) (CR c : ℝ) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t → AnFacts R G.w t ka wF T (WeightFamilies.MggK B CR T c j ka) xs → addL wF wL T = ka → WeightFamilies.dimY D j ≤ T → ∀ S : Scal B, Submodule.span B.KO ((X.Thc S.s0 (X.wF0 wL wF) : Submodule (𝒪 B.KO) _) : Set (B.Cusp j (X.wF0 wL wF))) = ⊤ → ∀ la, InBasis G la → InContainMR G la S CR T c

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InPsiOrConjugateUnit

theorem **inPsi_or_conjugate_unit** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (τ : E →+* ℂ) : InPsi H τ ∨ InPsi H (ComplexEmbedding.conjugate τ)

theorem **gaStar_conj_eq_letAct** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (ga : Matrix (Fin n) (Fin n) E) (h : IsUnit ga.det), ga.conjTranspose * ga = 1 → gaStar (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InPsiOrInPsiConjugate

theorem **conj_eq_neg_of_sq** {z : ℂ} {d : ℕ} (hd : 0 < d) (hz : z ^ 2 = -(d : ℂ)) : starRingEnd ℂ z = -z

theorem **inPsi_or_inPsi_conjugate** {E : Type} [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (τ : E →+* ℂ) : Formalism.L2.Pj.InPsi H τ ∨ Formalism.L2.Pj.InPsi H (ComplexEmbedding.conjugate τ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InZhGeoInjStable

theorem **act_natCast** (m : ℕ) : Xo.univ.t.act.act m = 𝟙 Xo.univ.t.G ^ m

theorem **comp_list_prod** {T T' : Over Xo.X} (h : T' ⟶ T) (L : List (T ⟶ Xo.univ.t.G)) : (L.map fun x => h ≫ x).prod = h ≫ L.prod

theorem **inZh_comp** {T T' : Over Xo.X} (h : T' ⟶ T) (a : Fin n → (T ⟶ Xo.univ.t.G)) (ha : InZh Xo Λ N a) : InZh Xo Λ N (fun b => h ≫ a b)

theorem **inZh_mul** {T : Over Xo.X} (a a' : Fin n → (T ⟶ Xo.univ.t.G)) (ha : InZh Xo Λ N a) (ha' : InZh Xo Λ N a') : InZh Xo Λ N (fun b => a b * a' b)

theorem **geoInj_comp** {S T : Type} [CommRing S] [CommRing T] (x0 : Sp S ⟶ Xo.X) (x1 : Sp T ⟶ Xo.X) (ψ : S →+* T) (h : ptO Xo x1 ⟶ ptO Xo x0) (hh : h.left = spMap ψ) {J : Type} (z : J → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G)) (hz : GeoInj Xo x0 z) : GeoInj Xo x1 (fun i b => h ≫ z i b)

theorem **inZh_of_killedBy** (hΛ : Λ.TrivAt l.M) {T : Over Xo.X} (u : Fin n → (T ⟶ Xo.univ.t.G)) (hu : ∀ b, u b ^ N = 1 ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) (u b)) : InZh Xo Λ N u

theorem **inZh_geoInj_stable** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (n : ℕ) (Λ : GenLat E n) (N : ℕ), (∀ (T T' : Over Xo.X) (h : T' ⟶ T) (a : Fin n → (T ⟶ Xo.univ.t.G)), InZh Xo Λ N a → InZh Xo Λ N (fun b => h ≫ a b)) ∧ (∀ (T : Over Xo.X) (a a' : Fin n → (T ⟶ Xo.univ.t.G)), InZh Xo Λ N a → InZh Xo Λ N a' → InZh Xo Λ N (fun b => a b * a' b)) ∧ (∀ (S T : Type) [CommRing S] [CommRing T] (x0 : Sp S ⟶ Xo.X) (x1 : Sp T ⟶ Xo.X) (ψ : S →+* T) (h : ptO Xo x1 ⟶ ptO Xo x0), h.left = spMap ψ → ∀ (J : Type) (z : J → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G)), GeoInj Xo x0 z → GeoInj Xo x1 (fun i b => h ≫ z i b)) ∧ (Λ.TrivAt l.M → ∀ (T : Over Xo.X) (u : Fin n → (T ⟶ Xo.univ.t.G)), (∀ b, u b ^ N = 1 ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) (u b)) → InZh Xo Λ N u)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InZhGeoInjStableStatement

theorem **_root_.GlobalTheory.T.inZh_geoInj_stable_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (n : ℕ) (Λ : GenLat E n) (N : ℕ), (∀ (T T' : Over Xo.X) (h : T' ⟶ T) (a : Fin n → (T ⟶ Xo.univ.t.G)), InZh Xo Λ N a → InZh Xo Λ N (fun b => h ≫ a b)) ∧ (∀ (T : Over Xo.X) (a a' : Fin n → (T ⟶ Xo.univ.t.G)), InZh Xo Λ N a → InZh Xo Λ N a' → InZh Xo Λ N (fun b => a b * a' b)) ∧ (∀ (S T : Type) [CommRing S] [CommRing T] (x0 : Sp S ⟶ Xo.X) (x1 : Sp T ⟶ Xo.X) (ψ : S →+* T) (h : ptO Xo x1 ⟶ ptO Xo x0), h.left = spMap ψ → ∀ (J : Type) (z : J → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G)), GeoInj Xo x0 z → GeoInj Xo x1 (fun i b => h ≫ z i b)) ∧ (Λ.TrivAt l.M → ∀ (T : Over Xo.X) (u : Fin n → (T ⟶ Xo.univ.t.G)), (∀ b, u b ^ N = 1 ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) (u b)) → InZh Xo Λ N u)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IndepSpanRankTransferLargeFieldComplex

theorem **exists_coeffs** {K A s μ : Type} [Field K] [CommRing A] [Algebra K A] [Fintype μ] (w : μ → A) (p : MvPolynomial s A) (hp : ∀ m, p.coeff m ∈ Submodule.span K (Set.range w)) : ∃ c : μ → MvPolynomial s K, p = ∑ k, MvPolynomial.map (algebraMap K A) (c k) * MvPolynomial.C (w k)

theorem **eval_comb** {K A s μ F M : Type} [Field K] [CommRing A] [Algebra K A] [Fintype μ] [Field F] [CommRing M] [Algebra F M] (κ : K →+* F) (e : A →+* M) (he : ∀ (a : K) (x : A), e (a • x) = κ a • e x) (τ : s → F) (τ' : s → M) (hτ : ∀ i, τ' i = algebraMap F M (τ i)) (w : μ → A) (c : μ → MvPolynomial s K) : MvPolynomial.eval₂Hom e τ' (∑ k, MvPolynomial.map (algebraMap K A) (c k) * MvPolynomial.C (w k)) = Fintype.linearCombination F (fun k => e (w k)) (fun k => MvPolynomial.eval₂Hom κ τ (c k))

theorem **inj_transfer** {F V W ι : Type} [Field F] [AddCommGroup V] [Module F V] [AddCommGroup W] [Module F W] (L : V →ₗ[F] W) (hL : Function.Injective L) (v : ι → V) : (LinearIndependent F (fun i => L (v i)) ↔ LinearIndependent F v) ∧ (∀ x, L x ∈ Submodule.span F (Set.range fun i => L (v i)) ↔ x ∈ Submodule.span F (Set.range v)) ∧ Module.finrank F (Submodule.span F (Set.range fun i => L (v i))) = Module.finrank F (Submodule.span F (Set.range v))

theorem **emb_smul'** (X : PJ D B P j) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (a : B.KO) (F : X.Forms) : emb X la N (a • F) = kap X la N a • emb X la N F

theorem **emb_indep** (X : PJ D B P j) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) {μ : Type} (w : μ → X.Forms) (hw : LinearIndependent B.KO w) : LinearIndependent (Kp X la N) fun k => emb X la N (w k)

theorem **eval_injective** (X : PJ D B P j) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) : Function.Injective (MvPolynomial.eval₂Hom (kap X la N) (fun p : Idx D j N => algebraMap (Op X la N) (Kp X la N) (Tv X la N p)))

theorem **reading_transfer** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (R : Rdg X) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (t : Idx D j N → ℂ), R.Generic t → ∀ (u : WeightFamilies.Wt D j) (ι : Type) [Fintype ι] (p : ι → MvPolynomial (Idx D j N) X.Forms), (∀ i, OfWt X u (p i)) → (LinearIndependent (Kp X la N) (fun i => toRp X la N (p i)) ↔ LinearIndependent ℂ (fun i => R.toFn t (p i))) ∧ (∀ q, OfWt X u q → (toRp X la N q ∈ Submodule.span (Kp X la N) (Set.range fun i => toRp X la N (p i)) ↔ R.toFn t q ∈ Submodule.span ℂ (Set.range fun i => R.toFn t (p i)))) ∧ Module.finrank (Kp X la N) (Submodule.span (Kp X la N) (Set.range fun i => toRp X la N (p i))) = Module.finrank ℂ (Submodule.span ℂ (Set.range fun i => R.toFn t (p i)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IndepSpanRankTransferLargeFieldComplexStatement

theorem **_root_.GlobalTheory.T.indep_span_rank_transfer_largeField_complex** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (R : Rdg X) (la : HeightOneSpectrum (𝒪 B.KO)) (N : ℕ) (t : Idx D j N → ℂ), R.Generic t → ∀ (u : WeightFamilies.Wt D j) (ι : Type) [Fintype ι] (p : ι → MvPolynomial (Idx D j N) X.Forms), (∀ i, OfWt X u (p i)) → (LinearIndependent (Kp X la N) (fun i => toRp X la N (p i)) ↔ LinearIndependent ℂ (fun i => R.toFn t (p i))) ∧ (∀ q, OfWt X u q → (toRp X la N q ∈ Submodule.span (Kp X la N) (Set.range fun i => toRp X la N (p i)) ↔ R.toFn t q ∈ Submodule.span ℂ (Set.range fun i => R.toFn t (p i)))) ∧ Module.finrank (Kp X la N) (Submodule.span (Kp X la N) (Set.range fun i => toRp X la N (p i))) = Module.finrank ℂ (Submodule.span ℂ (Set.range fun i => R.toFn t (p i)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IntegrableOnFundamentalDomainOfContinuous

theorem **setIntegral_fundamentalDomain_add_right** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (μ : Measure V) [μ.IsAddHaarMeasure] (b : Module.Basis κ ℝ V) (f : V → E) (hb : ∀ γ ∈ Submodule.span ℤ (Set.range b), ∀ v, f (v + γ) = f v) (a : V) : ∫ v in ZSpan.fundamentalDomain b, f (v + a) ∂μ = ∫ v in ZSpan.fundamentalDomain b, f v ∂μ

theorem **integrableOn_fundamentalDomain_of_continuous** {E : Type} [NormedAddCommGroup E] (μ : Measure V) [μ.IsAddHaarMeasure] (b : Module.Basis κ ℝ V) (f : V → E) (hf : Continuous f) : IntegrableOn f (ZSpan.fundamentalDomain b) μ

theorem **average_fundamentalDomain_eq_real** {V κ κ' : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace V] [BorelSpace V] [Fintype κ] [Fintype κ'] (μ : MeasureTheory.Measure V) [μ.IsAddHaarMeasure] (b : Module.Basis κ ℝ V) (b' : Module.Basis κ' ℝ V) (f : V → ℝ) (hf : Continuous f) (hb : ∀ γ ∈ Submodule.span ℤ (Set.range b), ∀ v, f (v + γ) = f v) (hle : Submodule.span ℤ (Set.range b') ≤ Submodule.span ℤ (Set.range b)) (a : V) : 𝒻 v in ZSpan.fundamentalDomain b', f (v + a) ∂μ = 𝒻 v in ZSpan.fundamentalDomain b, f v ∂μ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IntegralityBoundOfFamilyContainingScaledThetaRul

theorem **totalDegree_eq_sum_rows** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ harmA c a n) (h0 : P ≠ 0) (hh : P.IsHomogeneous P.totalDegree) : P.totalDegree = ∑ τ : E →+* ℂ, ∑ i : Fin (Dw.mult τ), a (τ, i)

theorem **lieFac_sq** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (Dw : Datum H) (x : ℝ) (hx : 0 < x) (a : (E →+* ℂ) × ℕ → ℕ) (d : ℕ) (hd : d = ∑ τ : E →+* ℂ, ∑ i : Fin (Dw.mult τ), a (τ, i)) : lieFac Dw x a ^ 2 = ((2 * Real.pi * x)⁻¹) ^ d

theorem **i2iAt_of_containsTh** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (xR : ℝ) (nab : NabV Xo c), IsNab Xo c nab → (∀ σ, GoodEmb c σ → IsUnf (U σ) ∧ (∀ k x, FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IntegralityBoundOfFamilyContainingScaledThetaRulStatement

theorem **_root_.GlobalTheory.T.integralityBound_of_family_containing_scaled_thetaRules** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r) (xR : ℝ) (nab : NabV Xo c), IsNab Xo c nab → (∀ σ, GoodEmb c σ → IsUnf (U σ) ∧ (∀ k x, FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InvariantOperatorValueTriangularChange

theorem **eval₂_shift** [Infinite k] (φ : k →+* T) {P : MvPolynomial (Letter Ty r n) k} (hP : RowInv r n k P) (σ : Ty) (i i' : Fin (r σ)) (hi : i' < i) (t : T) (y y' : Letter Ty r n → T) (h₁ : ∀ b, y' ⟨σ, i, b⟩ = y ⟨σ, i, b⟩ + t * y ⟨σ, i', b⟩) (h₂ : ∀ ℓ, (¬ ∃ b, ℓ = ⟨σ, i, b⟩) → y' ℓ = y ℓ) : eval₂Hom φ y' P = eval₂Hom φ y P

private theorem **eval₂Hom_unipotent_substitution_eq** [Infinite k] (φ : k →+* T) {P : MvPolynomial (Letter Ty r n) k} (hP : RowInv r n k P) (σ : Ty) (i : Fin (r σ)) [DecidablePred fun ℓ : Letter Ty r n => ∃ b, ℓ = ⟨σ, (i, b)⟩] (N : Fin (r σ) → T) (hN : ∀ j, i ≤ j → N j = 0) (y : Letter Ty r n → T) [DecidableEq (Fin (r σ))] (S : Finset (Fin (r σ))) : eval₂Hom φ (fun ℓ : Letter Ty r n => if ∃ b, ℓ = ⟨σ, (i, b)⟩ then y ℓ + ∑ j ∈ S, N j * y ⟨σ, (j, ℓ.2.2)⟩ else y ℓ) P = eval₂Hom φ y P

private theorem **eval₂Hom_unipotent_block_substitution_eq** [Infinite k] (φ : k →+* T) {P : MvPolynomial (Letter Ty r n) k} (hP : RowInv r n k P) (N : ∀ σ, Fin (r σ) → Fin (r σ) → T) (hN : ∀ σ i j, i ≤ j → N σ i j = 0) [DecidableEq (Σ σ : Ty, Fin (r σ))] (R : Finset (Σ σ : Ty, Fin (r σ))) : ∀ y : Letter Ty r n → T, eval₂Hom φ (fun ℓ : Letter Ty r n => y ℓ + if ⟨ℓ.1, ℓ.2.1⟩ : Σ σ : Ty, Fin (r σ)) ∈ R then ∑ j, N ℓ.1 ℓ.2.1 j * y ⟨ℓ.1, (j, ℓ.2.2)⟩ else 0) P = eval₂Hom φ y P

theorem **eval₂_unip** [Infinite k] [Fintype Ty] (φ : k →+* T) {P : MvPolynomial (Letter Ty r n) k} (hP : RowInv r n k P) (N : ∀ σ, Fin (r σ) → Fin (r σ) → T) (hN : ∀ σ i j, i ≤ j → N σ i j = 0) (y : Letter Ty r n → T) : eval₂Hom φ (fun ℓ : Letter Ty r n => y ℓ + ∑ j, N ℓ.1 ℓ.2.1 j * y ⟨ℓ.1, j, ℓ.2.2⟩) P = eval₂Hom φ y P

theorem **eval₂_diag** [Fintype Ty] (φ : k →+* T) {a : Ty → ℕ → ℕ} {P : MvPolynomial (Letter Ty r n) k} (hP : RowDeg r n k a P) (c : ∀ σ, Fin (r σ) → T) (y : Letter Ty r n → T) : eval₂Hom φ (fun ℓ : Letter Ty r n => c ℓ.1 ℓ.2.1 * y ℓ) P = (∏ p : (Σ σ : Ty, Fin (r σ)), c p.1 p.2 ^ a p.1 p.2) * eval₂Hom φ y P

theorem **eval₂_triangular** [Infinite k] [Fintype Ty] (φ : k →+* T) {a : Ty → ℕ → ℕ} {P : MvPolynomial (Letter Ty r n) k} (hI : RowInv r n k P) (hD : RowDeg r n k a P) (c : ∀ σ, Fin (r σ) → Fin (r σ) → T) (hc : ∀ σ i j, i < j → c σ i j = 0) (hu : ∀ σ i, IsUnit (c σ i i)) (y : Letter Ty r n → T) : eval₂Hom φ (fun ℓ : Letter Ty r n => ∑ j, c ℓ.1 ℓ.2.1 j * y ⟨ℓ.1, j, ℓ.2.2⟩) P = (∏ p : (Σ σ : Ty, Fin (r σ)), c p.1 p.2 p.2 ^ a p.1 p.2) * eval₂Hom φ y P

theorem **dEnd_apply** (F : Fin d → PS S 2 d) (v : Fin d → S) (g : PS S 1 d) : dEnd F v g = Dof F v g

theorem **dEnd_comb** (F : Fin d → PS S 2 d) {κ : Type} [Fintype κ] (c : κ → S) (w : κ → Fin d → S) (v : Fin d → S) (hv : ∀ e, v e = ∑ j, c j * w j e) : dEnd F v = ∑ j, c j • dEnd F (w j)

theorem **dmon_eq** {A : Type} [CommRing A] [Algebra S A] (F : Fin d → PS S 2 d) (Ψ : A →ₐ[S] Module.End S (PS S 1 d)) {I : Type} [Fintype I] (u : I → Fin d → S) (x : I → A) (hx : ∀ l, Ψ (x l) = dEnd F (u l)) (m : I →₀ ℕ) (g : PS S 1 d) : DmonOf F u m g = Ψ (∏ l, x l ^ m l) g

theorem **dpoly_monomial** (F : Fin d → PS S 2 d) {I : Type} [Fintype I] (u : I → Fin d → S) (m : I →₀ ℕ) (g : PS S 1 d) : DpolyOf F u (MvPolynomial.monomial m 1) g = DmonOf F u m g

theorem **prodForm_prod** {A : Type} [CommRing A] [Algebra S A] {n : ℕ} (lam : Fin n → A →ₗ[S] S) (x : Fin n → A) : prodForm lam (∏ b, PiTensorProduct.singleAlgHom (R

theorem **nabVal_triangular** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) [StarRing (𝒪 E)] (S : Type) [CommRing S] [Algebra c.F S] [Algebra ℚ S] (d : ℕ) (B : Germ.FB S d (𝒪 E)) (hg : Germ.PS S 2 d), B.IsCanonGauge S hg → ∀ (k n : ℕ) (f : Fin n → MvPowerSeries (Fin 1 × Fin d) S) (uvec uvec' : Lt1 Dw → Fin d → S) (cf : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Fin (Dw.mult τ) → S), (∀ τ i j, i < j → cf τ i j = 0) → (∀ τ i e, uvec' ⟨τ, i⟩ e = ∑ j, cf τ i j * uvec ⟨τ, j⟩ e) → (∀ τ i, IsUnit (cf τ i i)) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a ⟨τ, i⟩) → ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, B.nabVal S hg H.w0 k (f b) uvec' (MvPolynomial.monomial (colOf m b) 1) = (∏ ℓ : Lt1 Dw, cf ℓ.1 ℓ.2 ℓ.2 ^ a ⟨ℓ.1, ℓ.2⟩) * ∑ m ∈ P.support, algeb

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.InvariantOperatorValueTriangularChangeStatement

theorem **_root_.GlobalTheory.T.invariantOperator_value_triangular_change** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) [StarRing (𝒪 E)] (S : Type) [CommRing S] [Algebra c.F S] [Algebra ℚ S] (d : ℕ) (B : Germ.FB S d (𝒪 E)) (hg : Germ.PS S 2 d), B.IsCanonGauge S hg → ∀ (k n : ℕ) (f : Fin n → MvPowerSeries (Fin 1 × Fin d) S) (uvec uvec' : Lt1 Dw → Fin d → S) (cf : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → Fin (Dw.mult τ) → S), (∀ τ i j, i < j → cf τ i j = 0) → (∀ τ i e, uvec' ⟨τ, i⟩ e = ∑ j, cf τ i j * uvec ⟨τ, j⟩ e) → (∀ τ i, IsUnit (cf τ i i)) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a ⟨τ, i⟩) → ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, B.nabVal S hg H.w0 k (f b) uvec' (MvPolynomial.monomial (colOf m b) 1) = (∏ ℓ : Lt1 Dw, cf ℓ.1 ℓ.

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsEtaRuleExistsUniqueMemSec

theorem **finite_aut** (Dw : Datum H) {k : ℕ} (hk : 0 < k) : Finite (LevGrp Dw k)

theorem **etaExpr_scale** {S : Type} [CommRing S] {k : ℕ} (hk : 0 < k) {J : Type} [Fintype J] {n : ℕ} (cf : J → ℤ) (x : J → Fin n → LatMod Dw k) (w : LatMod Dw k → S) (t : S) : etaExpr k cf x (fun q => t * w q) = t ^ (n * Nat.card (LevGrp Dw k)) * etaExpr k cf x w

theorem **frBor_baseChange** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} {z z' : FrPt Xo c T} (h : FrBor t y y') (hz : FrMap φ y z) (hz' : FrMap φ y' z') : FrBor (fun i => Units.map (φ : S →* T) (t i)) z z'

theorem **exists_level_of_eq** {k : ℕ} {S : Type} [CommRing S] {x₀ x₁ : Sp S ⟶ Xo.X} (h : x₁ = x₀) {ev : LatMod Dw k → (ptO Xo x₀ ⟶ Xo.univ.t.G)} (hev : IsFullLev Xo k x₀ ev) : ∃ ev' : LatMod Dw k → (ptO Xo x₁ ⟶ Xo.univ.t.G), IsFullLev Xo k x₁ ev' ∧ ∀ q, (ev' q).left = (ev q).left

theorem **etaValAt_frBor** (hL : Laws) {R : ℕ} (hR : Odd R) {J : Type} [Fintype J] {n : ℕ} {cf : J → ℤ} {x : J → Fin n → LatMod Dw (R ^ 4)} {S : Type} [CommRing S] [Algebra c.F S] {t : Ix Dw → Sˣ} {y y' : FrPt Xo c S} (h : FrBor t y y') {v : S} (hv : EtaValAt Xo c R cf x y v) : EtaValAt Xo c R cf x y' ((t none : S) ^ (n * Nat.card (LevGrp Dw (R ^ 4))) * v)

theorem **isEtaRule_existsUnique_mem_sec** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (R : ℕ), Odd R → ∀ (J : Type) [Fintype J] (n : ℕ) (cf : J → ℤ) (x : J → Fin n → LatMod Dw (R ^ 4)), (∃! f, IsEtaRule Xo c R cf x f) ∧ ∀ f, IsEtaRule Xo c R cf x f → f ∈ sec Xo c 0 (n * Nat.card (LevGrp Dw (R ^ 4)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsFrameRawChartBaseChange

theorem **spec_comp** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

theorem **gm_unit_map** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) (u : Aˣ) {Z : Scheme.{0}} (r : Z ⟶ Sp B) (z : Z) : gm r z ⊤ (unitΓ (Units.map f.toMonoidHom u)) = gm (r ≫ spMap f) z ⊤ (unitΓ u)

theorem **val_scale** {Y Z : Scheme.{0}} {K : Type} [Fintype K] {φ : K → (Y ⟶ Xo.univ.t.G.left)} {ε : K → ℤ} {z : Z} {J : (Z ⟶ Xo.univ.t.G.left) → Xo.univ.t.PP.L.ι} (hJ : Good Xo.univ.t.PP.L z J) (c : Γ(Y, ⊤)ˣ) (t : LB.Gen (LB.LinComb Xo.univ.t.PP.L φ ε)) (r : Z ⟶ Y) : val (Formalism.L2.Pj.scaleGen φ ε c t) r z J = gm r z ⊤ c * val t r z J

theorem **psIn_map** {d n n' : ℕ} (m : ℕ) (r : Fin n → Fin n') (f : MvPowerSeries (Fin n × Fin d) S) : psIn d m r (MvPowerSeries.map ψ f) = MvPolynomial.map ψ (psIn d m r f)

theorem **subst_square** {n n' d : ℕ} (m : ℕ) (φ : Fin n × Fin d → MvPolynomial (Fin n' × Fin d) S) : homAt (spMap ψ ≫ x0) (jetSubstP m (fun i => MvPolynomial.map ψ (φ i)) : Jet S' (Fin n × Fin d) m →ₐ[S'] Jet S' (Fin n' × Fin d) m) ≫ rjBC ψ x0 d m n = rjBC ψ x0 d m n' ≫ homAt x0 (jetSubstP m φ : Jet S (Fin n × Fin d) m →ₐ[S] Jet S (Fin n' × Fin d) m)

theorem **jetMap_congr** {R κ κ' : Type} [CommRing R] {m : ℕ} {f g : κ → MvPolynomial κ' R} (e : f = g) (hf : ∀ i, f i ∈ orig R κ') (hg : ∀ i, g i ∈ orig R κ') : jetMap (m

theorem **copy_square** (d m n : ℕ) (i : Fin n) : homAt (spMap ψ ≫ x0) (jetCopy d m i : Jet S' (Fin 1 × Fin d) m →ₐ[S'] Jet S' (Fin n × Fin d) m) ≫ rjBC ψ x0 d m 1 = rjBC ψ x0 d m n ≫ homAt x0 (jetCopy d m i : Jet S (Fin 1 × Fin d) m →ₐ[S] Jet S (Fin n × Fin d) m)

theorem **point_bc** (hx : ∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) {n : ℕ} (m : ℕ) (i : Fin n) : rjpt (spMap ψ ≫ x0) d x' m i = rjBC ψ x0 d m n ≫ rjpt x0 d x m i

theorem **val_subst** {n : ℕ} (φ : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S) (φ' : Fin 2 → Fin d → MvPolynomial (Fin n × Fin d) S') (hφ : ∀ i j, φ' i j = MvPolynomial.map ψ (φ i j)) (r : Z ⟶ (rjO (spMap ψ ≫ x0) d m n).left) : val (rpe (spMap ψ ≫ x0) d x' e' m φ') r z J = val (rpe x0 d x e m φ) (r ≫ (rjBC ψ x0 d m n).left) z J

theorem **isFrame_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (e : ∀ m, RFrTy x0 d x m) (e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m), (∀ m, (e m).IsGen) → (∀ m, (e' m).IsGen) → (∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) → ∀ (ph1 ph2 : MvPowerSeries (Fin 3 × Fin d) S) (sy : MvPowerSeries (Fin 2 × Fin d) S) (ad : 𝒪 E → MvPowerSeries (Fin 2 × Fin d) S), RIsFrame x0 d x F act ph1 p

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsFrameRawChartBaseChangeStatement

theorem **_root_.GlobalTheory.T.isFrame_rawChart_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (e : ∀ m, RFrTy x0 d x m) (e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m), (∀ m, (e m).IsGen) → (∀ m, (e' m).IsGen) → (∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) → ∀ (ph1 ph2 : MvPowerSeries (Fin 3 × Fin d) S) (sy : MvPowerSeries (Fin 2 × Fin d) S) (ad : 𝒪 E → MvPowerSeries (Fin 2 × Fin d) S)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsHarmonicAndMemMinorSpan

theorem **eq_zero_of_sum_type** {A B : X.Ψ → ℂ} (h : ∀ α : 𝒪 E, ∑ τ : X.Ψ, (τ.1 α * A τ + conj (τ.1 α) * B τ) = 0) (τ : X.Ψ) : A τ = 0 ∧ B τ = 0

theorem **nonempty_type** : Nonempty X.Ψ

theorem **conj_emb_herm** (τ : E →+* ℂ) (u v : Fin r → E) : conj (τ (herm X.Gm u v)) = τ (herm X.Gm v u)

theorem **herm_eq_zero_symm** {u v : Fin r → E} (h : herm X.Gm u v = 0) : herm X.Gm v u = 0

theorem **EF_smul_eq_sum** (α : E) (v w : Fin r → E) : ((EF X (α • v) w : ℚ) : ℂ) = ∑ τ : X.Ψ, (τ.1 α * (I / 2 * X.c τ * conj (τ.1 (herm X.Gm w v))) + conj (τ.1 α) * -(I / 2 * X.c τ * τ.1 (herm X.Gm w v)))

theorem **EF_eq_zero_of_herm** {v w : Fin r → E} (h : herm X.Gm w v = 0) : EF X v w = 0

theorem **herm_eq_zero_of_EF** {v w : Fin r → E} (h : ∀ α : 𝒪 E, EF X ((α : E) • v) w = 0) : herm X.Gm w v = 0

theorem **mul_per_smul_eq_sum** (x : Frame X.Ψ r) (s : X.Ψ → ℂ) (α : E) (v : Fin r → E) (p : X.Ψ × Fin r) : s p.1 * per X x (α • v) p = ∑ τ, s τ * (τ.1 α * perGe X x v τ p + conj (τ.1 α) * perLt X x v τ p)

theorem **mulVec_per_smul** (idx : Fin g ≃ X.Ψ × Fin r) (M : Matrix (Fin g) (Fin g) ℂ) (x : Frame X.Ψ r) (s : X.Ψ → ℂ) (α : E) (v : Fin r → E) (j : Fin g) : (M *ᵥ fun i => s (idx i).1 * per X x (α • v) (idx i)) j = ∑ τ, s τ * (τ.1 α * (M *ᵥ (perGe X x v τ ∘ idx)) j + conj (τ.1 α) * (M *ᵥ (perLt X x v τ ∘ idx)) j)

theorem **emb_neg_kappa** (τ : X.Ψ) : τ.1 (-X.κ0) = I * X.c τ

theorem **flowMat_mulVec_eq** (τ : X.Ψ) (s : ℂ) (v : Fin r → ℂ) : X.flowMat ξ τ s *ᵥ v = v + (-(I * X.c τ * s) * hermC τ.1 X.Gm v (Data.emb τ.1 ξ)) • Data.emb τ.1 ξ

theorem **col_flow_eq** (s : X.Ψ → ℂ) (x : Frame X.Ψ r) (τ : X.Ψ) (a : Fin r) : FD.col (mat (X.flow ξ s x) τ) a = FD.col (mat x τ) a + (-(I * X.c τ * s τ) * hermC τ.1 X.Gm (FD.col (mat x τ) a) (Data.emb τ.1 ξ)) • Data.emb τ.1 ξ

theorem **flow_rmul** (s : X.Ψ → ℂ) (x b : Frame X.Ψ r) : X.flow ξ s (rmul x b) = rmul (X.flow ξ s x) b

theorem **flowMat_neg_mul** (τ : X.Ψ) (s : ℂ) : X.flowMat ξ τ (-s) * X.flowMat ξ τ s = 1

theorem **inv_flow_mulVec** (s : X.Ψ → ℂ) (x : Frame X.Ψ r) (τ : X.Ψ) (v : Fin r → ℂ) : (mat (X.flow ξ s x) τ)⁻¹ *ᵥ v = (mat x τ)⁻¹ *ᵥ v + (I * X.c τ * s τ * hermC τ.1 X.Gm v (Data.emb τ.1 ξ)) • ((mat x τ)⁻¹ *ᵥ Data.emb τ.1 ξ)

theorem **per_flow** (s : X.Ψ → ℂ) (x : Frame X.Ψ r) (v : Fin r → E) (p : X.Ψ × Fin r) : per X (X.flow ξ s x) v p = per X x v p + s p.1 * per X x ((-X.κ0 * herm X.Gm v ξ) • ξ) p

theorem **per_flow_of_orth** (s : X.Ψ → ℂ) (x : Frame X.Ψ r) {v : Fin r → E} (hv : herm X.Gm v ξ = 0) : per X (X.flow ξ s x) v = per X x v

theorem **coper_flow_of_orth** (s : X.Ψ → ℂ) (x : Frame X.Ψ r) {v : Fin r → E} (hv : herm X.Gm v ξ = 0) : coper X (X.flow ξ s x) v = coper X x v

theorem **cuspNu_nonneg** (u : Fin g → ℝ) (τ : X.Ψ) : 0 ≤ cuspNu X bas ξ u τ

theorem **cuspNu_eq_zero_iff** (u : Fin g → ℝ) (τ : X.Ψ) : cuspNu X bas ξ u τ = 0 ↔ ∑ j, (u j : ℂ) * cuspLin X bas ξ τ j = 0

theorem **dot_cuspMat** (u : Fin g → ℝ) (τ : X.Ψ) : u ⬝ᵥ cuspMat X bas ξ τ *ᵥ u = 4 * cuspNu X bas ξ u τ

theorem **dot_sum_cuspMat** (s : X.Ψ → ℂ) (u : Fin g → ℝ) : (fun i => (u i : ℂ)) ⬝ᵥ (∑ τ, s τ • (cuspMat X bas ξ τ).map fun t : ℝ => (t : ℂ)) *ᵥ (fun i => (u i : ℂ)) = 4 * ∑ τ, (cuspNu X bas ξ u τ : ℂ) * s τ

theorem **coe_int_smul** (hξ : ξ ∈ X.Λ) (α : 𝒪 E) : ((α • (⟨ξ, hξ⟩ : X.Λ) : X.Λ) : Fin r → E) = (α : E) • ξ

theorem **EF_bas_inr** (v : X.Λ) (j : Fin g) : EF X v (bas (Pi.single (Sum.inr j) 1)) = -(bas.symm v (Sum.inl j) : ℤ)

theorem **EF_bas_inl** (v : X.Λ) (j : Fin g) : EF X v (bas (Pi.single (Sum.inl j) 1)) = (bas.symm v (Sum.inr j) : ℤ)

theorem **IsAdapted.symm_apply_eq_zero** {v : X.Λ} (hv : (v : Fin r → E) ∈ Submodule.span E {ξ}) {t : Fin g ⊕ Fin g} (ht : ∀ k ∈ S, t ≠ Sum.inl k) : bas.symm v t = 0

theorem **IsAdapted.symm_smul_eq_zero** (hξ : ξ ∈ X.Λ) (α : 𝒪 E) {t : Fin g ⊕ Fin g} (ht : ∀ k ∈ S, t ≠ Sum.inl k) : bas.symm (α • (⟨ξ, hξ⟩)) t = 0

theorem **IsAdapted.herm_inl** (hξ : ξ ∈ X.Λ) (j : Fin g) : herm X.Gm (bas (Pi.single (Sum.inl j) 1)) ξ = 0

theorem **IsAdapted.cuspLin_ne_zero** (hξ : X.IsIso ξ) (τ : X.Ψ) : cuspLin X bas ξ τ ≠ 0

theorem **IsAdapted.hMat_flow** (hξ : X.IsIso ξ) (s : X.Ψ → ℂ) (x : Frame X.Ψ r) : hMat X bas idx (X.flow ξ s x) = hMat X bas idx x

theorem **IsAdapted.splitMat_flow** (hξ : X.IsIso ξ) (s : X.Ψ → ℂ) (x : Frame X.Ψ r) : splitMat X bas idx (X.flow ξ s x) = splitMat X bas idx x

theorem **IsAdapted.hMat_mulVec_part** (hξ : ξ ∈ X.Λ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (τ : X.Ψ) (j : Fin g) : (hMat X bas idx x *ᵥ (perGe X x ξ τ ∘ idx)) j = -(I / 2 * X.c τ * conj (cuspLin X bas ξ τ j)) ∧ (hMat X bas idx x *ᵥ (perLt X x ξ τ ∘ idx)) j = I / 2 * X.c τ * cuspLin X bas ξ τ j

theorem **IsAdapted.tauMat_flow** (hξ : X.IsIso ξ) (s : X.Ψ → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : tauMat X bas idx (X.flow ξ s x) = tauMat X bas idx x + ∑ τ, s τ • (cuspMat X bas ξ τ).map fun t : ℝ => (t : ℂ)

theorem **exists_block** (p : ℕ) (w l : Fin r → ℂ) (h : ∑ a, w a * l a = 0) {a₀ c₀ : Fin r} (ha : (a₀ : ℕ) < p) (hc : p ≤ (c₀ : ℕ)) (hl : l a₀ ≠ 0) (hw : w c₀ ≠ 0) : ∃ K : Fin r → Fin r → ℂ, (∀ a : Fin r, ∑ c : Fin r, (if p ≤ (c : ℕ) then K a c * w c else 0) = w a) ∧ ∀ c : Fin r, ∑ a : Fin r, (if (a : ℕ) < p then K a c * l a else 0) = -l c

theorem **exists_unip_coper_eq_zero** (hξ : herm X.Gm ξ ξ = 0) {x : Frame X.Ψ r} (hx : ∀ τ, (mat x τ).det ≠ 0) (h₁ : ∀ τ, perLt X x ξ τ ≠ 0) (h₂ : ∀ τ, perGe X x ξ τ ≠ 0) : ∃ K, coper X (rmul x (unip X K)) ξ = 0

theorem **IsAdapted.part_ne_zero** (hξ : X.IsIso ξ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (τ : X.Ψ) : perLt X x ξ τ ≠ 0 ∧ perGe X x ξ τ ≠ 0

theorem **IsAdapted.splitMat_mulVec_part** (hξ : ξ ∈ X.Λ) {y : Frame X.Ψ r} (hy : y ∈ X.dom) (h : coper X y ξ = 0) (τ : X.Ψ) : splitMat X bas idx y *ᵥ (perGe X y ξ τ ∘ idx) = 0 ∧ splitMat X bas idx y *ᵥ (perLt X y ξ τ ∘ idx) = 0

theorem **differentiable_prod_gauss** {n : ℕ} (M : Matrix (Fin g) (Fin g) ℂ) : Differentiable ℂ fun z : Fin n → Fin g → ℂ => ∏ k, cexp (Real.pi / 2 * (z k ⬝ᵥ M *ᵥ z k))

theorem **differentiable_mulVec_col** {n : ℕ} (M : Matrix (Fin g) (Fin g) ℂ) (k : Fin n) : Differentiable ℂ fun z : Fin n → Fin g → ℂ => M *ᵥ z k

theorem **analyticAt_splitTerm** {n : ℕ} (a b : Fin n → Fin g → ℝ) (y : Frame X.Ψ r) (m : Fin n → Fin g → ℤ) : AnalyticAt ℂ (splitTerm X bas idx a b y m) 0

theorem **sum_letterDir_xVar** {n : ℕ} (y : Frame X.Ψ r) (v : Fin r → E) (τ : X.Ψ) (k : Fin n) : ∑ i, per X y v (rowOf X hr (τ, .inl i)) • letterDir X hr idx (Harm.xVar τ i k) = Pi.single k (perGe X y v τ ∘ idx)

theorem **sum_letterDir_yVar** {n : ℕ} (y : Frame X.Ψ r) (v : Fin r → E) (τ : X.Ψ) (k : Fin n) : ∑ i, per X y v (rowOf X hr (τ, .inr i)) • letterDir X hr idx (Harm.yVar τ i k) = Pi.single k (perLt X y v τ ∘ idx)

theorem **splitTerm_add_single** {n : ℕ} (a b : Fin n → Fin g → ℝ) {y : Frame X.Ψ r} (hy : y ∈ X.dom) (m : Fin n → Fin g → ℤ) {d : Fin g → ℂ} (hd : splitMat X bas idx y *ᵥ d = 0) (hm : ∀ k, ((fun i => (m k i : ℂ)) + fun i => (a k i : ℂ)) ⬝ᵥ hMat X bas idx y *ᵥ d = 0) (k : Fin n) (z : Fin n → Fin g → ℂ) (t : ℂ) : splitTerm X bas idx a b y m (z + t • (Pi.single k d : Fin n → Fin g → ℂ)) = splitTerm X bas idx a b y m z

theorem **summable_norm_flowCoeff** (n : ℕ) (a b : Fin n → Fin g → ℝ) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) {y : Frame X.Ψ r} (hy : y ∈ X.dom) : Summable fun m => ‖flowCoeff X hr bas idx n a b P y m‖

theorem **IsAdapted.hasSum_flowCoeff** (hξ : X.IsIso ξ) (n : ℕ) (a b : Fin n → Fin g → ℝ) {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : Harm.IsHarmonic P) {y : Frame X.Ψ r} (hy : y ∈ X.dom) {s : X.Ψ → ℂ} (hys : X.flow ξ s y ∈ X.dom) : HasSum (fun m : Fin n → Fin g → ℤ => flowCoeff X hr bas idx n a b P y m * cexp (4 * Real.pi * I * ∑ τ, (((∑ k, cuspNu X bas ξ fun i => (m k i : ℝ) + a k i) τ : ℝ) : ℂ) * s τ)) (thFr X hr bas idx n a b P (X.flow ξ s y))

theorem **IsAdapted.flowCoeff_eq_zero** (hξ : X.IsIso ξ) (n : ℕ) (a b : Fin n → Fin g → ℝ) {c : X.Ψ → Fin q0 → ℕ} {c' : X.Ψ → Fin X.p0 → ℕ} {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : P ∈ GaussGauge.minorSpan (Fin n) c c') {τ : X.Ψ} (hc : (∃ i : Fin q0, i.val + 1 = q0 ∧ 1 ≤ c τ i) ∨ ∃ i : Fin X.p0, i.val + 1 = X.p0 ∧ 1 ≤ c' τ i) {y : Frame X.Ψ r} (hy : y ∈ X.dom) (h : coper X y ξ = 0) {m : Fin n → Fin g → ℤ} (hm : (∑ k, cuspNu X bas ξ fun i => (m k i : ℝ) + a k i) τ = 0) : flowCoeff X hr bas idx n a b P y m = 0

theorem **IsAdapted.exists_unip** (hξ : X.IsIso ξ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : ∃ K, coper X (rmul x (unip X K)) ξ = 0

theorem **IsAdapted.exists_expansion** (hξ : X.IsIso ξ) (n : ℕ) (a b : Fin n → Fin g → ℝ) {c : X.Ψ → Fin q0 → ℕ} {c' : X.Ψ → Fin X.p0 → ℕ} {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hH : Harm.IsHarmonic P) (hP : P ∈ GaussGauge.minorSpan (Fin n) c c') (hc : ∃ τ, (∃ i : Fin q0, i.val + 1 = q0 ∧ 1 ≤ c τ i) ∨ ∃ i : Fin X.p0, i.val + 1 = X.p0 ∧ 1 ≤ c' τ i) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : ∃ A : (Fin n → Fin g → ℤ) → ℂ, (Summable fun m => ‖A m‖) ∧ (∀ s : X.Ψ → ℂ, X.flow ξ s x ∈ X.dom → HasSum (fun m : Fin n → Fin g → ℤ => A m * cexp (4 * Real.pi * I * ∑ τ, (((∑ k, cuspNu X bas ξ fun i => (m k i : ℝ) + a k i) τ : ℝ) : ℂ) * s τ)) (thFr X hr bas idx n a b P (X.flow ξ s x))) ∧ ∀ m : Fin n → Fin g → ℤ, (∑ k, cuspNu X bas ξ fun i => (m k i : ℝ) + a k i) = 0 → A m = 0

theorem **exists_isAdapted** (hs : IsSympl X bas) {ξ : Fin r → E} (hξ : herm X.Gm ξ ξ = 0) : ∃ (bas' : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (S : Finset (Fin g)), IsSympl X bas' ∧ IsAdapted X bas' ξ S

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsHolFamPullbackLiftOfCommonBase

theorem **isHolFam_pullback_lift** : ∀ (X Y₁ Y₂ : Scheme.{0}) (p₁ : Y₁ ⟶ X) (p₂ : Y₂ ⟶ X) (N : ℕ) (e₁ : (Fin N → ℂ) → (Sp ℂ ⟶ Y₁)) (e₂ : (Fin N → ℂ) → (Sp ℂ ⟶ Y₂)) (u : Sp ℂ ⟶ X) (h₁ : ∀ z, e₁ z ≫ p₁ = u) (h₂ : ∀ z, e₂ z ≫ p₂ = u), IsHolFam e₁ → IsHolFam e₂ → IsHolFam (fun z => pullback.lift (e₁ z) (e₂ z) ((h₁ z).trans (h₂ z).symm))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsHolFamPullbackLiftOfCommonBaseStatement

theorem **_root_.GlobalTheory.T.isHolFam_pullback_lift_of_common_base** : ∀ (X Y₁ Y₂ : Scheme.{0}) (p₁ : Y₁ ⟶ X) (p₂ : Y₂ ⟶ X) (N : ℕ) (e₁ : (Fin N → ℂ) → (Sp ℂ ⟶ Y₁)) (e₂ : (Fin N → ℂ) → (Sp ℂ ⟶ Y₂)) (u : Sp ℂ ⟶ X) (h₁ : ∀ z, e₁ z ≫ p₁ = u) (h₂ : ∀ z, e₂ z ≫ p₂ = u), IsHolFam e₁ → IsHolFam e₂ → IsHolFam (fun z => pullback.lift (e₁ z) (e₂ z) ((h₁ z).trans (h₂ z).symm))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsIntegralCoeffCof

theorem **det_gen_ne_zero** (K : Type) [Field K] {Ix : Type} (n : ℕ) (t : Ix) : (gen K n t).det ≠ 0

theorem **map_gen** {R S : Type} [CommRing R] [CommRing S] (f : R →+* S) {Ix : Type} (n : ℕ) (t : Ix) : (MvPolynomial.map f).mapMatrix (gen R n t) = gen S n t

theorem **detFac_ne_zero** (K : Type) [Field K] (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n) : detFac Dw pl K a s lam ≠ 0

theorem **map_cof** {R S : Type} [CommRing R] [CommRing S] (f : R →+* S) (Q : MvPolynomial (LtN Dw n) R) : MvPolynomial.map f (cof Dw pl R n Q) = cof Dw pl S n (MvPolynomial.map f Q)

theorem **isIntegral_coeff_cof** {K : Type} [Field K] (P : MvPolynomial (LtN Dw n) K) (hP : ∀ m, P.coeff m ∈ (integralClosure ℤ K).toSubring) (m) : IsIntegral ℤ ((cof Dw pl K n P).coeff m)

theorem **exists_ioLFRec** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n), (∀ τ, Antitone fun i => a (τ, i)) → WtIs Dw pl a s lam → ∃ (io : VVc c lam →ₗ[c.F] MvPolynomial (LtN Dw n) c.F) (hw : VVc c lam), IoLFRec c (fun t => embF c (pl t)) lam a io hw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsLocalizedModuleSubtype

lemma **_root_.GlobalTheory.DualFamily.herm_one_eq** {n : ℕ} (x y : Fin n → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y = ∑ a, x a * _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (y a)

lemma **_root_.GlobalTheory.DualFamily.herm_one_symm** {n : ℕ} (x y : Fin n → E) : _root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y = _root_.ReciprocityTheory.ChartRestriction.G.FD.cc (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 y x)

lemma **_root_.GlobalTheory.DualFamily.isIntegral_cc** {a : E} (ha : IsIntegral ℤ a) : IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.cc a)

lemma **_root_.GlobalTheory.DualFamily.exists_herm_eq** {n : ℕ} (f : (Fin n → E) →ₗ[E] E) : ∃ w : Fin n → E, ∀ u, f u = _root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 u w

lemma **_root_.GlobalTheory.DualFamily.eq_zero_of_line** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ.FG) (w : Fin n → E) (hw : ∀ c : E, c • w ∈ Λ) : w = 0

lemma **_root_.GlobalTheory.DualFamily.span_eq_top** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ.FG) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) → x ∈ Λ) : Submodule.span E (Λ : Set (Fin n → E)) = ⊤

private lemma **exists_linearForm_extend_of_span_eq_univ** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hspan : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) (g : Λ →ₗ[𝒪 E] 𝒪 E) : ∃ φ : (Fin n → E) →ₗ[E] E, ∀ y : Λ, φ y = g y

theorem **_root_.GlobalTheory.exists_dual_family** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ.FG) (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (_root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 x y)) → x ∈ Λ) : Submodule.span E (Λ : Set (Fin n → E)) = ⊤ ∧ ∃ (m : ℕ) (v w : Fin m → Fin n → E), (∀ i, v i ∈ Λ) ∧ (∀ i, w i ∈ Λ) ∧ ∀ u : Fin n → E, u = ∑ i, _root_.ReciprocityTheory.ChartRestriction.G.FD.herm 1 u (w i) • v i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsLocallyNoetherian

@[simp] theorem **pol_A** (Xo : Moduli D l) : (pol Xo).A = Xo.univ.t.G

@[simp] theorem **pol_L** (Xo : Moduli D l) : (pol Xo).L = Xo.univ.t.PP.L

@[simp] theorem **pol_g** (Xo : Moduli D l) : (pol Xo).g = Module.finrank ℚ E / 2 * (D.p + D.q)

theorem **smooth** (Xo : Moduli D l) : Smooth Xo.s

theorem **isLocallyNoetherian** (Xo : Moduli D l) : IsLocallyNoetherian Xo.X

theorem **isReduced** (Xo : Moduli D l) : IsReduced Xo.X

theorem **isUnit_natCast** (Xo : Moduli D l) (n : ℕ) (hn : 0 < n) : IsUnit (n : Γ(Xo.X, ⊤))

theorem **moduli_univ_polAV** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l), IsLocallyNoetherian Xo.X ∧ IsReduced Xo.X ∧ (∀ n : ℕ, 0 < n → IsUnit (n : Γ(Xo.X, ⊤))) ∧ IsAbelianScheme Xo.univ.t.G ∧ ∀ (p : ∀ (A : Over Xo.X) [GrpObj A] [IsCommMonObj A], LB A.left → ℕ → Prop), (∀ P : PolAV Xo.X, p P.A P.L P.g) → p Xo.univ.t.G Xo.univ.t.PP.L (Module.finrank ℚ E / 2 * (Dw.p + Dw.q))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsNabTwist

theorem **isNab_twist** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (ta : c.F ≃+* c.F) (h : Fixes c ta), ∀ [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → IsNab Xo c (fun {S} _ _ y k F α => (nab (twPt h y) k F α).val)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsPiActLOfIsTorusOf

theorem **negOnePow_natCast_mul** (M : ℕ) (k : ℤ) : Int.negOnePow ((M : ℤ) * k) = Int.negOnePow k ^ M

theorem **isPi_actL_of_isTorusOf** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (Pm : D.La →+ (Fin g → ℂ)) (Hm : Matrix (Fin g) (Fin g) ℂ) (T : I2.PT E) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim), IsPeriod D ty Pm → IsTorusOf D ty Pm Hm T ψ → ∀ (α : 𝒪 E) (M : ℕ), α * H.bar α = (M : 𝒪 E) → I2.IsPi T M (vspActL α)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsProportionalOfSqLaw

theorem **chgMat_mulVec** (bas bas' : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (m : Fin g ⊕ Fin g → ℤ) : chgMat X bas bas' *ᵥ m = bas.symm (bas' m)

theorem **chgMat_mul_chgMat** (bas bas' : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) : chgMat X bas bas' * chgMat X bas' bas = 1

theorem **IsSamePt.symm** {bas bas' : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} {u u' : Fin g ⊕ Fin g → ℝ} (h : IsSamePt X bas bas' u u') : IsSamePt X bas' bas u' u

theorem **perCx_chgMat** (bas bas' : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (x : Frame X.Ψ r) (w : Fin g ⊕ Fin g → ℂ) : perCx X bas idx x ((chgMat X bas bas').map (Int.cast : ℤ → ℂ) *ᵥ w) = perCx X bas' idx x w

theorem **IsSamePt.perCx_eq** {bas bas' : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} {u u' : Fin g ⊕ Fin g → ℝ} (h : IsSamePt X bas bas' u u') (x : Frame X.Ψ r) : perCx X bas idx x (fun s => (u s : ℂ)) = perCx X bas' idx x (fun s => (u' s : ℂ))

theorem **canFn_eq_canTh_mulVec** (a b : Fin g → ℝ) (z : Fin g → ℂ) : canFn X bas idx a b x z = canTh (tauMat X bas idx x) a b (hMat X bas idx x *ᵥ z)

theorem **hermForm_Hm_indep** (hs : IsSympl X bas) (hs' : IsSympl X bas') (hx : x ∈ X.dom) (z w : Fin g → ℂ) : hermForm (Hm X bas' idx x) z w = hermForm (Hm X bas idx x) z w

theorem **hMat_mulVec_perCx_real** (u : Fin g ⊕ Fin g → ℝ) : hMat X bas idx x *ᵥ perCx X bas idx x (fun s => (u s : ℂ)) = (fun i => (u (Sum.inl i) : ℂ)) + tauMat X bas idx x *ᵥ fun i => (u (Sum.inr i) : ℂ)

theorem **canFn_entire** (a b : Fin g → ℝ) : Differentiable ℂ (canFn X bas idx a b x)

theorem **exists_canFn_ne_zero** (a b : Fin g → ℝ) : ∃ z, canFn X bas idx a b x z ≠ 0

theorem **canFn_add_per** (a b : Fin g → ℝ) (l : X.Λ) (z : Fin g → ℂ) : canFn X bas idx a b x (z + per X x l ∘ idx) = latMult X bas a b l * cexp (π * hermForm (Hm X bas idx x) z (per X x l ∘ idx) + π / 2 * hermForm (Hm X bas idx x) (per X x l ∘ idx) (per X x l ∘ idx)) * canFn X bas idx a b x z

theorem **canFn_translate** (a b : Fin g → ℝ) (u : Fin g ⊕ Fin g → ℝ) {y : Fin g → ℂ} (hy : y = perCx X bas idx x fun s => (u s : ℂ)) (z : Fin g → ℂ) : cexp (-(π * hermForm (Hm X bas idx x) z y) - π / 2 * hermForm (Hm X bas idx x) y y) * canFn X bas idx a b x (z + y) = translConst b (u ∘ Sum.inr) (u ∘ Sum.inl) * canFn X bas idx (a + u ∘ Sum.inr) (b + u ∘ Sum.inl) x z

theorem **chgSp_mem** : chgSp bas bas' ∈ Matrix.symplecticGroup (Fin g) ℤ

theorem **period_chgSp** : tauMat X bas' idx x = spAct (chgSp bas bas') (tauMat X bas idx x) ∧ hMat X bas' idx x = ((jAut (chgSp bas bas') (tauMat X bas idx x))⁻¹)ᵀ * hMat X bas idx x ∧ (hMat X bas idx x).det = (jAut (chgSp bas bas') (tauMat X bas idx x)).det * (hMat X bas' idx x).det

theorem **dot_imInv_chgSp** (z : Fin g → ℂ) : (hMat X bas idx x *ᵥ z) ⬝ᵥ imInv X bas idx x *ᵥ (hMat X bas idx x *ᵥ z) = (hMat X bas' idx x *ᵥ z) ⬝ᵥ imInv X bas' idx x *ᵥ (hMat X bas' idx x *ᵥ z) + 2 * I * expArg (chgSp bas bas') (tauMat X bas idx x) (hMat X bas idx x *ᵥ z)

theorem **exists_sq_law_zero** : ∃ (k k' : Fin g → ℤ) (c : ℂ), c ^ 4 = 1 ∧ ∀ x ∈ X.dom, ∀ z, (hMat X bas' idx x).det * canFn X bas' idx (halfVec k) (halfVec k') x z ^ 2 = c * (hMat X bas idx x).det * canFn X bas idx 0 0 x z ^ 2

theorem **exists_int_of_canMult_eq** {a b a' b' : Fin g → ℝ} (h : ∀ m n, canMult a b m n = canMult a' b' m n) : ∃ n m : Fin g → ℤ, a' = a + (fun i => (n i : ℝ)) ∧ b' = b + fun i => (m i : ℝ)

theorem **IsProportional.add_int** (h : IsProportional X idx κ bas a b bas' a' b') (n m : Fin g → ℤ) : IsProportional X idx (κ * cexp (2 * π * I * ((a' ⬝ᵥ fun i => (m i : ℝ)) : ℝ)) ^ 2) bas a b bas' (a' + fun i => (n i : ℝ)) (b' + fun i => (m i : ℝ))

theorem **isProportional_of_sq_law** (h : ∀ x ∈ X.dom, ∀ z, (hMat X bas' idx x).det * canFn X bas' idx a' b' x z ^ 2 = κ * (hMat X bas idx x).det * canFn X bas idx a b x z ^ 2) : IsProportional X idx κ bas a b bas' a' b'

theorem **canFn_translate_eq_mul** {x : Frame X.Ψ r} (hx : x ∈ X.dom) {μ : ℂ} (h : ∀ z, canFn X bas' idx a' b' x z = μ * canFn X bas idx a b x z) {u u' : Fin g ⊕ Fin g → ℝ} (hu : IsSamePt X bas bas' u u') (z : Fin g → ℂ) : canFn X bas' idx (a' + u' ∘ Sum.inr) (b' + u' ∘ Sum.inl) x z = translConst b (u ∘ Sum.inr) (u ∘ Sum.inl) / translConst b' (u' ∘ Sum.inr) (u' ∘ Sum.inl) * μ * canFn X bas idx (a + u ∘ Sum.inr) (b + u ∘ Sum.inl) x z

theorem **IsProportional.translate** (h : IsProportional X idx κ bas a b bas' a' b') {u u' : Fin g ⊕ Fin g → ℝ} (hu : IsSamePt X bas bas' u u') : IsProportional X idx (κ * (translConst b (u ∘ Sum.inr) (u ∘ Sum.inl) / translConst b' (u' ∘ Sum.inr) (u' ∘ Sum.inl)) ^ 2) bas (a + u ∘ Sum.inr) (b + u ∘ Sum.inl) bas' (a' + u' ∘ Sum.inr) (b' + u' ∘ Sum.inl)

private theorem **chartRestrictionData_dom_nonempty** : X.dom.Nonempty

theorem **IsProportional.isMatched** (h : IsProportional X idx κ bas a b bas' a' b') : IsMatched X bas a b bas' a' b'

theorem **exists_isProportional** : ∃ k k' : Fin g → ℤ, ∀ u u' : Fin g ⊕ Fin g → ℝ, IsSamePt X bas bas' u u' → ∃ κ : ℂ, ‖κ‖ = 1 ∧ IsProportional X idx κ bas (u ∘ Sum.inr) (u ∘ Sum.inl) bas' ((fun i => (k i : ℝ) / 2) + u' ∘ Sum.inr) ((fun i => (k' i : ℝ) / 2) + u' ∘ Sum.inl)

theorem **IsMatched.exists_isProportional** (h : IsMatched X bas a b bas' a' b') : ∃ κ : ℂ, ‖κ‖ = 1 ∧ IsProportional X idx κ bas a b bas' a' b'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsRawChartBaseChangeJetPoints

theorem **spMap_spMap** {R R' R'' : Type} [CommRing R] [CommRing R'] [CommRing R''] (f : R' →+* R'') (g : R →+* R') : spMap f ≫ spMap g = spMap (f.comp g)

theorem **dualMap_alg** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) : (dualMap ψ).comp (algebraMap S (DualNumber S)) = (algebraMap T (DualNumber T)).comp ψ

theorem **dualMap_epsMul** {S T : Type} [CommRing S] [CommRing T] (ψ : S →+* T) (a : S) : (dualMap ψ).comp (epsMul a) = (epsMul (ψ a)).comp (dualMap ψ)

theorem **psTrunc_map** {S S' : Type} [CommRing S] [CommRing S'] (ψ : S →+* S') {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (f : MvPowerSeries κ S) : psTrunc m (MvPowerSeries.map ψ f) = MvPolynomial.map ψ (psTrunc m f)

theorem **basis_map** {S S' : Type} [CommRing S] [CommRing S'] (ψ : S →+* S') {σ κ : Type} [Fintype σ] [Fintype κ] (U : σ → κ → S) (h : ∀ v : κ → S, ∃! a : σ → S, ∀ j, ∑ ℓ, a ℓ * U ℓ j = v j) (v : κ → S') : ∃! a : σ → S', ∀ j, ∑ ℓ, a ℓ * ψ (U ℓ j) = v j

theorem **sq_aug** (d m : ℕ) : homAt (spMap ψ ≫ x0) (jetAug : Jet S' (Fin 1 × Fin d) m →ₐ[S'] S') ≫ rjBC ψ x0 d m 1 = baseMap ψ x0 ≫ homAt x0 (jetAug : Jet S (Fin 1 × Fin d) m →ₐ[S] S)

theorem **sq_trunc** (d m : ℕ) : homAt (spMap ψ ≫ x0) (jetTrunc : Jet S' (Fin 1 × Fin d) (m + 1) →ₐ[S'] Jet S' (Fin 1 × Fin d) m) ≫ rjBC ψ x0 d (m + 1) 1 = rjBC ψ x0 d m 1 ≫ homAt x0 (jetTrunc : Jet S (Fin 1 × Fin d) (m + 1) →ₐ[S] Jet S (Fin 1 × Fin d) m)

theorem **sq_subst** {n n' d : ℕ} (m : ℕ) (φ : Fin n × Fin d → MvPolynomial (Fin n' × Fin d) S) : homAt (spMap ψ ≫ x0) (jetSubstP m (fun i => MvPolynomial.map ψ (φ i)) : Jet S' (Fin n × Fin d) m →ₐ[S'] Jet S' (Fin n' × Fin d) m) ≫ rjBC ψ x0 d m n = rjBC ψ x0 d m n' ≫ homAt x0 (jetSubstP m φ : Jet S (Fin n × Fin d) m →ₐ[S] Jet S (Fin n' × Fin d) m)

theorem **sq_tan** (d : ℕ) (ℓ : Fin 1 × Fin d) : homAt (spMap ψ ≫ x0) (jetTan ℓ : Jet S' (Fin 1 × Fin d) (0 + 1) →ₐ[S'] DualNumber S') ≫ rjBC ψ x0 d 1 1 = dualPtMap ψ x0 ≫ homAt x0 (jetTan ℓ : Jet S (Fin 1 × Fin d) (0 + 1) →ₐ[S] DualNumber S)

theorem **jetMap_congr_fun** {R κ κ' : Type} [CommRing R] {m : ℕ} {f g : κ → MvPolynomial κ' R} (e : f = g) (hf : ∀ i, f i ∈ orig R κ') (hg : ∀ i, g i ∈ orig R κ') : jetMap (m

theorem **sq_copy** (d m n : ℕ) (i : Fin n) : homAt (spMap ψ ≫ x0) (jetCopy d m i : Jet S' (Fin 1 × Fin d) m →ₐ[S'] Jet S' (Fin n × Fin d) m) ≫ rjBC ψ x0 d m 1 = rjBC ψ x0 d m n ≫ homAt x0 (jetCopy d m i : Jet S (Fin 1 × Fin d) m →ₐ[S] Jet S (Fin n × Fin d) m)

theorem **list_prod_comp** {T T' : Over Xo.X} (f : T' ⟶ T) (L : List (T ⟶ Xo.univ.t.G)) : (L.map (f ≫ ·)).prod = f ≫ L.prod

theorem **isCombAt_map** {k : ℕ} {v : Fin k → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)} {cf : Fin k → S} {w : Sp (DualNumber S) ⟶ Xo.univ.t.G.left} (h : IsCombAt x0 v cf w) : IsCombAt (spMap ψ ≫ x0) (fun i => spMap (dualMap ψ) ≫ v i) (fun i => ψ (cf i)) (spMap (dualMap ψ) ≫ w)

theorem **rIsChart_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)) (uvec : Lt1 Dw → Fin d → S), RIsChart x0 u d F act uvec x → RIsChart (spMap ψ ≫ x0) (fun τ i => spMap (dualMap ψ) ≫ u τ i) d (fun i => MvPowerSeries.map ψ (F i)) (fun a i => MvPowerSeries.map ψ (act a i)) (fun ℓ j => ψ (uvec ℓ j)) x'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsRawChartBaseChangeJetPointsStatement

theorem **_root_.GlobalTheory.T.isRawChart_baseChange_jetPoints** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (u : ∀ τ : E →+* ℂ, Fin (Dw.mult τ) → (Sp (DualNumber S) ⟶ Xo.univ.t.G.left)) (uvec : Lt1 Dw → Fin d → S), RIsChart x0 u d F act uvec x → RIsChart (spMap ψ ≫ x0) (fun τ i => spMap (dualMap ψ) ≫ u τ i) d (fun i => MvPowerSeries.map ψ (F i)) (fun a i => MvPowerSeries.map ψ (act a i)) (fun ℓ j => ψ (uvec ℓ j)) x'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsRepsLevelTranslates

theorem **levReps_act_mul** (a b : 𝒪 E) (v : I2.Vsp dim) : I2.act a (I2.act b v) = I2.act (a * b) v

theorem **levReps_act_nat** (m : ℕ) (v : I2.Vsp dim) : I2.act (m : 𝒪 E) v = (m : ℂ) • v

theorem **levReps_act_smul** (a : 𝒪 E) (c : ℂ) (v : I2.Vsp dim) : I2.act a (c • v) = c • I2.act a v

theorem **levReps_act_add** (a : 𝒪 E) (v w : I2.Vsp dim) : I2.act a (v + w) = I2.act a v + I2.act a w

theorem **levReps_act_neg** (a : 𝒪 E) (v : I2.Vsp dim) : I2.act a (-v) = -I2.act a v

theorem **isReps_level_translates** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (T : I2.PT E) (mk : Dw.La →+ I2.Vsp T.dim) (α : 𝒪 E), α * H.bar α = (l.M : 𝒪 E) → l.Mb = Ideal.span {H.bar α} → (∀ a, mk a ∈ T.L) → (∀ (β : 𝒪 E) a, mk (β • a) = I2.act β (mk a)) → (∀ a, (∃ v ∈ T.L, mk a = (l.M : ℂ) • v) → ∃ b, a = l.M • b) → (∀ v ∈ T.L, ∃ a, ∃ v' ∈ T.L, v = mk a + (l.M : ℂ) • v') → ∀ x2 : LevQ Dw l → Dw.La, (∀ q, x2 q ∈ (Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La)) → (∀ q, x2 q - levRep q ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) → ∀ n : ℕ, ∃ K : Finset (Fin n → I2.Vsp T.dim), I2.IsReps T l.M n (vspActL α) K ∧ ∀ b ∈ K, ∃ q : Fin n → LevQ Dw l, b = fun i => I2.act (H.bar α) (-(((l.M : ℂ) ^ 2)⁻¹ • mk (x2 (q i))))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsThClsDDdBCIff

theorem **eF_incF** (a : c.F) : ((ddBC hle d).eF (incF c c' hle a) : (covCoeff hle d).F) = incF d.c₂ (covCoeff hle d) (le_covCoeff hle d) (d.eF a)

theorem **exists_algOver** (k : Type) [Field k] [IsAlgClosed k] [Algebra c.F k] : ∃ _ : Algebra c'.F k, AlgOver c c' hle k

theorem **thLa_iff** (w : (Fin n → LevQ Dw d.l₂) → c.F) (La : GenLat E n) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (k : Type) [CommRing k] [Algebra d.c₂.F k] [A₂ : Algebra (covCoeff hle d).F k] (h₂ : AlgOver d.c₂ (covCoeff hle d) (le_covCoeff hle d) k) (y₂ : FrPt d.Xo₂ (covCoeff hle d) k) (v : k) : thLaValAt d.Xo₂ (covCoeff hle d) (nabBC (le_covCoeff hle d) d.nab₂) (fun q => (ddBC hle d).eF (incF c c' hle (w q))) La N (MvPolynomial.map ((ddBC hle d).eF : c'.F →+* (ddBC hle d).c₂.F) (MvPolynomial.map (incF c c' hle) P)) y₂ v ↔ thLaValAt d.Xo₂ d.c₂ d.nab₂ (fun q => d.eF (w q)) La N (MvPolynomial.map (d.eF : c.F →+* d.c₂.F) P) (y₂.res (le_covCoeff hle d) h₂) v

theorem **isThClsD_ddBC_iff** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (n : ℕ) (d : DeltaDat Xo c n) (Cl : Type) [Fintype Cl] (w : Cl → (Fin n → LevQ Dw d.l₂) → c.F) (La : Cl → GenLat E n) (Nc gam : Cl → ℕ) (vol : c.F) (ep : Cl → c.F) (Vs : Type) [AddCommGroup Vs] [Module c.F Vs] (io : Vs →ₗ[c.F] MvPolynomial (LtN Dw n) c.F) (Vs' : Type) [AddCommGroup Vs'] [Module c'.F Vs'] (io' : Vs' →ₗ[c'.F] MvPolynomial (LtN Dw n) c'.F) (x : Cl → Vs) (x' : Cl → Vs'), (∀ k, io' (x' k) = MvPolynomial.map (incF c c' hle) (io (x k))) → d.cδ ∈ sec d.Xo₂ d.c₂ 0 0 → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (m : ℕ), ∀ f ∈ sec Xo c a m, (IsThClsD Xo c' (ddBC hle d) (fun k q => incF c c' hle (w k q)) La Nc gam (incF c c' hle vol) (fun k => incF c c' hle (ep k)) io' x' (rulBC hle f) ↔ IsThClsD Xo c d w La Nc gam vol ep io

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsThClsDGalRul

theorem **covAut_eF** {n : ℕ} (d : DeltaDat Xo c n) (ta : c.F ≃+* c.F) (a : c.F) : covAut d ta (d.eF a) = d.eF (ta a)

theorem **thLa_conj** (hC : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2) {n : ℕ} (d : DeltaDat Xo c n) (ta : c.F ≃+* c.F) (h₂ : Fixes d.c₂ (covAut d ta)) [StarRing (𝒪 E)] (hN : IsNab d.Xo₂ d.c₂ d.nab₂) (u : c.F) (hu : u ≠ 0) (w w' : (Fin n → LevQ Dw d.l₂) → c.F) (hw : ∀ q, ta (w q) = u * w' q) (La : GenLat E n) (N : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) (k : Type) [CommRing k] [Algebra d.c₂.F k] (y₂ : FrPt d.Xo₂ d.c₂ k) (v : Tw (covAut d ta) k) (hv : thLaValAt d.Xo₂ d.c₂ d.nab₂ (fun q => d.eF (w q)) La N (MvPolynomial.map (d.eF : c.F →+* d.c₂.F) P) (twPt h₂ y₂) v) : thLaValAt d.Xo₂ d.c₂ d.nab₂ (fun q => d.eF (w' q)) La N (MvPolynomial.map (d.eF : c.F →+* d.c₂.F) (MvPolynomial.map (ta : c.F →+* c.F) P)) y₂ (algebraMap d.c₂.F k (d.eF u⁻¹) * v.val)

theorem **isThClsD_galRul** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (ta : c.F ≃+* c.F) (h : Fixes c ta), ∀ (n : ℕ) (d : DeltaDat Xo c n) (h₂ : Fixes d.c₂ (covAut d ta)), (∃ _ : StarRing (𝒪 E), IsNab d.Xo₂ d.c₂ d.nab₂) → d.cδ ∈ sec d.Xo₂ d.c₂ 0 0 → ∀ (cx : Rul d.Xo₂ d.c₂) (ka u : c.F), ka ≠ 0 → u ≠ 0 → (∀ (k : Type) [Field k] [IsAlgClosed k] [Algebra d.c₂.F k] (y₂ : FrPt d.Xo₂ d.c₂ k), evR d.Xo₂ d.c₂ (galRul h₂ d.cδ) y₂ = algebraMap d.c₂.F k (d.eF ka) * evR d.Xo₂ d.c₂ cx y₂) → ∀ (Cl : Type) [Fintype Cl] (w w' : Cl → (Fin n → LevQ Dw d.l₂) → c.F) (La : Cl → GenLat E n) (Nc gam : Cl → ℕ) (vol : c.F) (ep : Cl → c.F) (Vs : Type) [AddCommGroup Vs] [Module c.F Vs] (io : Vs →ₗ[c.F] MvPolynomial (LtN Dw n) c.F) (a : (E →+* ℂ) × ℕ → ℕ), (∀ v, io v ∈ HH

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsTorusAtOfIsTorusOf

theorem **perFr_add** (v w : Fin r → E) (τ : E →+* ℂ) (i : ℕ) : Formalism.L2.Pj.perFr X x (v + w) τ i = Formalism.L2.Pj.perFr X x v τ i + Formalism.L2.Pj.perFr X x w τ i

theorem **perFr_smul** (α : E) (v : Fin r → E) (τ : E →+* ℂ) (i : ℕ) : Formalism.L2.Pj.perFr X x (α • v) τ i = τ α * Formalism.L2.Pj.perFr X x v τ i

theorem **norm_sq_eq_Hf** {J : Type} [Fintype J] (v : EuclideanSpace ℂ J) : ‖v‖ ^ 2 = (I2.Hf v v).re

theorem **isTorusAt_of_isTorusOf** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (W : UnfAt U k x) (Wu : UnitAt W) (T : I2.PT E) (ψ : (Fin W.g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim), GoodEmb c σ → IsUnf U → (U.X k).IsOrtho x.1 → IsTorusOf (W.F.Di (W.ik k)) (fun j => relab c σ (W.e j).1) W.Pm Wu.Hm T ψ → I2.IsTorusAt (U.X k) x.1 T ∧ T.r = r ∧ ∀ th, T.ph th = I2.phiFD (U.X k) th x.1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsTwistPullBaseChange

theorem **eq_pullS_of_isPullSec** {S : Scheme.{0}} (A : Over S) [GrpObj A] {L : LB A.left} (a : ℤ) {T T' : Over S} (h : T' ⟶ T) {s : LB.Sec (LPow A L a T)} {s' : LB.Sec (LPow A L a T')} (H : IsPullSec A L a (h ▷ A) (whiskerRight_snd _ _) s s') : s' = GlobalTheory.VVRank.pullS A a h s

theorem **isTwistPull_baseChange** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] (L : LB A.left), L.IsLB → ∀ (T T' : Over S) (h : T' ⟶ T) (ψ : T ⊗ A ⟶ T ⊗ A) (m : T ⊗ A ⟶ A) (hm : ψ ≫ snd T A = m) (ψ' : T' ⊗ A ⟶ T' ⊗ A) (m' : T' ⊗ A ⟶ A) (hm' : ψ' ≫ snd T' A = m'), ψ' ≫ h ▷ A = h ▷ A ≫ ψ → ∀ (a b : ℤ) (α : LB.Gen (LB.LinComb L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsValueAtJetPointCmFrameSection

theorem **jet_pt_eq** (p : PrimeSpectrum (Jet K σ m)) : p = IsLocalRing.closedPoint (Jet K σ m)

theorem **gs_appLE_top** {Y X : Scheme.{0}} (φ : Y ⟶ X) (y : Y) (U : X.Opens) (h : ⊤ ≤ φ ⁻¹ᵁ U) (a : Γ(X, U)) : gs φ y U a = gs (𝟙 Y) y ⊤ ((φ.appLE U ⊤ h).hom a)

theorem **isPullSec_pullSecLP** (s : LB.Sec (LPow A L a T)) : IsPullSec A L a h hh s (pullSecLP A L a h hh s)

theorem **lftJ_snd** (z : Fin g → ℂ) : lftJ e he z ≫ (snd (Over.mk x0) G).left = e z

theorem **lftJ_fst** (z : Fin g → ℂ) : lftJ e he z ≫ (fst (Over.mk x0) G).left = 𝟙 _

theorem **lftJ_eq** (holP : IsHolFam (lftJ e he)) (m : ℕ) : (ptSecT G (xJ e hol he m)).left ≫ (jB (x0

theorem **isValueAt_jet** (holP : IsHolFam (lftJ e he)) (m k : ℕ) (F : LB.Sec (LPow G L (k : ℤ) (overAt x0 (Jet ℂ (Fin 1 × Fin g) m)))) (hF' : IsPullSec G L (k : ℤ) (jB m ▷ G) (whiskerRight_snd _ _) F F') (θF : (Fin g → ℂ) → ℂ) (fr : L.ι → (Fin g → ℂ) → ℂ) (hfr : ∀ j, DifferentiableOn ℂ (fr j) {z | ⊤ ≤ e z ⁻¹ᵁ L.U j}) (hfr0 : ∀ j z, ⊤ ≤ e z ⁻¹ᵁ L.U j → fr j z ≠ 0) (hrd : ∀ j z (pt : Sp ℂ ⟶ (Over.mk x0 ⊗ G).left), pt ≫ (snd (Over.mk x0) G).left = e z → pt ≫ (fst (Over.mk x0) G).left = 𝟙 (Sp ℂ) → ⊤ ≤ e z ⁻¹ᵁ L.U j → valAt' pt ((LPow G L (k : ℤ) (Over.mk x0)).U (fun _ => j)) (F.f (fun _ => j)) = θF z * fr j z ^ (k : ℤ)) (τ : LB.Gen (LB.LinComb L (Y

theorem **isHolFam_lftJ** (hol : IsHolFam e) : IsHolFam (lftJ e he)

theorem **onePr_apply** {g : ℕ} (v : Fin (1 * g) → ℂ) (j : Fin g) : onePr g v j = v (finProdFinEquiv ((0 : Fin 1), j))

theorem **anLin_onePr** {g : ℕ} (p : Fin 1 × Fin g) : anLin (onePr g) p = MvPolynomial.X ((0 : Fin 1), finProdFinEquiv ((0 : Fin 1), p.2))

theorem **jetN_one** {g : ℕ} (m : ℕ) {f : (Fin g → ℂ) → ℂ} (hf : ContDiffAt ℂ ω f 0) : jetN (n

theorem **isValueAt_jetPt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ (m k : ℕ) (F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) (θ : (Fin g → ℂ) → ℂ) (τ : LB.Gen (LB.LinComb Xo.univ.t.PP.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsValueAtJetPointCmFrameSectionStatement

theorem **_root_.GlobalTheory.T.isValueAt_jetPoint_cmFrame_section** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ (m k : ℕ) (F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) (θ : (Fin g → ℂ) → ℂ) (τ : LB.Gen (LB.LinComb Xo.univ.t.PP.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsVtOverIsUUOverBaseChange

theorem **exists_mulTwist** {S : Scheme.{0}} (A : Over S) [GrpObj A] {L : LB A.left} (hL : L.IsLB) (N : ℕ) {T T' : Over S} (h : T' ⟶ T) {s : LB.Sec (LPow A L 1 T)} {s' : LB.Sec (LPow A L 1 T')} (hs : IsPullSec A L 1 (h ▷ A) (whiskerRight_snd _ _) s s') {α₀ : LB.Gen (LB.LinComb L (K

theorem **isVtOver_isUUOver_baseChange** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ) (I : Ideal (𝒪 E)) (T T' : Over Xo.X) (h : T' ⟶ T) (s : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 T)) (s' : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 T')), IsPullSec Xo.univ.t.G Xo.univ.t.PP.L 1 (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) s s' →

∀ (F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T)) (F' : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T')), IsPullSec Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ▷ Xo.univ.t.G) (whiskerRight_snd _ _) F F' → (∀ t : T ⟶ Xo.univ.t.G, IsVtOver Xo.univ.t.G Xo.univ.t.PP N (KilledBy Xo I) s t F → IsVtOver Xo.univ.t.G Xo.univ.t.PP N (KilledBy Xo I) s' (h ≫ t) F') ∧ ∀ x₁ x₂ : T ⟶ Xo.univ.t.G, IsUUOver Xo.univ.t.G Xo.univ.t.PP N s x₁ x₂ F → IsUUOver Xo.univ.t.G Xo.univ.t.PP N s'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsotropicEnumerationSurjectiveLocalRing

theorem **comp_list_prod** {S : Scheme.{0}} {G T T' : Over S} [GrpObj G] (h : T' ⟶ T) (L : List (T ⟶ G)) : h ≫ L.prod = (L.map fun x => h ≫ x).prod

theorem **inZh_comp** (Xo : Moduli Dw l) {n : ℕ} (Λ : GenLat E n) (N : ℕ) {T T' : Over Xo.X} (h : T' ⟶ T) {a : Fin n → (T ⟶ Xo.univ.t.G)} (ha : InZh Xo Λ N a) : InZh Xo Λ N (fun b => h ≫ a b)

theorem **ev_sum** {k : Type} [Field k] {Q : Type} [Group Q] {act : 𝒪 E → Q → Q} {e : ℕ → Q → Q → kˣ → Prop} (I : TateIso Dw k Q act e) {m : ℕ} (hm : 0 < m) {n : ℕ} (v : Fin n → Dw.La) : I.ev m (∑ j, v j) = (List.ofFn fun j => I.ev m (v j)).prod

theorem **exists_eq_of_isAlgClosed** (Xo : Moduli Dw l) {K : Type} [Field K] [IsAlgClosed K] (x : Sp K ⟶ Xo.X) {n : ℕ} (Λ : GenLat E n) {N : ℕ} (hΛ : Λ.Unimod) (hN : 0 < N) (hle : Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La) (w : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x ⟶ Xo.univ.t.G)) (hw : ∀ i, InZh Xo Λ N (w i)) (hinj : Function.Injective w) (a : Fin n → (ptO Xo x ⟶ Xo.univ.t.G)) (ha : InZh Xo Λ N a) : ∃ i, w i = a

theorem **isotropic_enumeration_onto_of_isLocalRing** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (T : Type) [CommRing T] [IsLocalRing T] (x0 : Sp T ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∀ z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G), (∀ i, InZh Xo Λ N (z i)) → GeoInj Xo x0 z → (∀ a, InZh Xo Λ N a → ∃ i, z i = a) ∧ ∀ z' : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G), (∀ i, InZh Xo Λ N (z' i)) → GeoInj Xo x0 z' → ∃ σ : Equiv.Perm (Fin (Λ.zCard N (Dw.p + Dw.q))), ∀ i, z' i = z (σ i)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsotropicEnumerationSurjectiveLocalRingStatement

theorem **_root_.GlobalTheory.T.isotropic_enumeration_surjective_local_ring** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (T : Type) [CommRing T] [IsLocalRing T] (x0 : Sp T ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∀ z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G), (∀ i, InZh Xo Λ N (z i)) → GeoInj Xo x0 z → (∀ a, InZh Xo Λ N a → ∃ i, z i = a) ∧ ∀ z' : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G), (∀ i, InZh Xo Λ N (z' i)) → GeoInj Xo x0 z' → ∃ σ : Equiv.Perm (Fin (Λ.zCard N (Dw.p + Dw.q))), ∀ i, z' i = z (σ i)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsotropicSumIndependentOfChoicesLocal

theorem **shortSum_toΓ_injective** (S : Type) [CommRing S] : Function.Injective (toΓ S)

private theorem **eq_one_of_sq_eq_one_of_pow_odd_sq_eq_one** {M : Type*} [CommMonoid M] (x : M) (N : ℕ) (hN : Odd N) (h2 : x ^ 2 = 1) (hN2 : x ^ (N ^ 2) = 1) : x = 1

theorem **shortSum_indep_local** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (P : MvPolynomial (LtN Dw n) c.F) (T : Type) [CommRing T] [IsLocalRing T] [Algebra c.F T] (y : FrPt Xo c T) (z1 : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (tp1 : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (F1 : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (z2 : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (tp2 : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (F2 : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.IsotropicSumIndependentOfChoicesLocalStatement

theorem **_root_.GlobalTheory.T.isotropicSum_independent_of_choices_local** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (P : MvPolynomial (LtN Dw n) c.F) (T : Type) [CommRing T] [IsLocalRing T] [Algebra c.F T] (y : FrPt Xo c T) (z1 : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (tp1 : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (F1 : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (z2 : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (tp2 : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (F2 : (Fin n → LevQ Dw l) → Fin (Λ.

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.JetRdMulInvPow

private lemma **jetN_eq_renN_psJet_tay** {n g : ℕ} (m : ℕ) (f : (Fin n → Fin g → ℂ) → ℂ) : jetN m f = renN n g m (psJet m (tay (fun v => f (colsOf n g v)) 0))

theorem **jetN_mul** {n g : ℕ} (m : ℕ) {f h : (Fin n → Fin g → ℂ) → ℂ} (hf : AnO f) (hh : AnO h) : jetN m (fun w => f w * h w) = jetN m f * jetN m h

theorem **jetN_one** {n g : ℕ} (m : ℕ) : jetN m (fun _ : Fin n → Fin g → ℂ => (1 : ℂ)) = 1

theorem **jetN_pow** {n g : ℕ} (m : ℕ) {f : (Fin n → Fin g → ℂ) → ℂ} (hf : AnO f) (p : ℕ) : jetN m (fun w => f w ^ p) = jetN m f ^ p

theorem **jetN_inv_mul** {n g : ℕ} (m : ℕ) {f : (Fin n → Fin g → ℂ) → ℂ} (hf : AnO f) (h0 : f (colsOf n g 0) ≠ 0) : jetN m (fun w => (f w)⁻¹) * jetN m f = 1

theorem **AnO.zpow** {n g : ℕ} {f : (Fin n → Fin g → ℂ) → ℂ} (hf : AnO f) (h0 : f (colsOf n g 0) ≠ 0) (k : ℤ) : AnO fun w => f w ^ k

theorem **origin_mem** {ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (hψ : Φ.JetMap m ψ Λ) {U : A.left.Opens} (hU : ⊤ ≤ ψ ⁻¹ᵁ U) : ⊤ ≤ (Φ.e (Λ (colsOf n g 0))).left ⁻¹ᵁ U

theorem **anO_fr** {ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left} {Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (hψ : Φ.JetMap m ψ Λ) {j : L.L.ι} (hU : ⊤ ≤ ψ ⁻¹ᵁ L.L.U j) : AnO fun w => Φ.fr j (Λ w)

theorem **anO_prod** (hφ : ∀ k, Φ.JetMap m (φ k) (Λ k)) (ε : K → ℤ) {j : K → L.L.ι} (h : ⊤ ≤ LB.wchart L.L φ j) : AnO fun w => ∏ k, Φ.fr (j k) (Λ k w) ^ ε k

theorem **prod_ne** (hφ : ∀ k, Φ.JetMap m (φ k) (Λ k)) (ε : K → ℤ) {j : K → L.L.ι} (h : ⊤ ≤ LB.wchart L.L φ j) : ∏ k, Φ.fr (j k) (Λ k (colsOf n g 0)) ^ ε k ≠ 0

theorem **map_res_val** {X : Scheme.{0}} {U V : X.Opens} (h : V ≤ U) (h' : ⊤ ≤ V) (u : Γ(X, U)ˣ) : (X.presheaf.map (homOfLE h').op).hom (Semi.res h u).val = (X.presheaf.map (homOfLE (h'.trans h)).op).hom u.val

theorem **GlobalTheory.AnVal.jetRd_mul_inv_pow** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (n m : ℕ) (K K' : Type) [Fintype K] [Fintype K'] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (φ' : K' → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε' : K' → ℤ) (τ' : LB.Gen (LB.LinComb L.L φ' ε')) (Λ' : K' → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.JetRd m φ ε τ Λ → Φ.JetRd m φ' ε' τ' Λ' → Φ.JetRd m (Sum.elim φ φ') (Sum.elim ε ε') (LB.mulGen L.L φ ε φ' ε' τ τ') (Sum.elim Λ Λ') ∧ Φ.JetRd m φ (fun k => -ε k) (LB.invGen L.L φ ε τ) Λ ∧ ∀ p : ℕ, Φ.JetRd m φ (fun k => (p : ℤ) * ε k) (fun j => τ.s j ^ p) Λ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.JetReadingLinearSubstitutionPoints

theorem **tay_coord** {N : ℕ} (k : Fin N) : tay (fun v : Fin N → ℂ => v k) 0 = MvPowerSeries.X ((0 : Fin 1), k)

theorem **psJet_X** {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (s : κ) : psJet m (MvPowerSeries.X s : MvPowerSeries κ ℂ) = Ideal.Quotient.mk _ (MvPolynomial.X s)

theorem **jetOf_coord** {N : ℕ} (m : ℕ) (k : Fin N) : jetOf m (fun v : Fin N → ℂ => v k) = Ideal.Quotient.mk _ (MvPolynomial.X ((0 : Fin 1), k))

theorem **flatL_apply** {n n' g : ℕ} (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (v : Fin (n' * g) → ℂ) (k : Fin (n * g)) : flatL B v k = B (colsOf n' g v) (finProdFinEquiv.symm k).1 (finProdFinEquiv.symm k).2

theorem **colsOf_flatL** {n n' g : ℕ} (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (v : Fin (n' * g) → ℂ) : colsOf n g (flatL B v) = B (colsOf n' g v)

theorem **subst_renN** {n n' g m : ℕ} (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (σ : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (hσ : ∀ p, σ (psJet m (MvPowerSeries.X p)) = jetN m (fun w => B w p.1 p.2)) (a : Jet ℂ (Fin 1 × Fin (n * g)) m) : σ (renN n g m a) = renN n' g m (jetMap (anLin (flatL B)) (anLin_mem _) a)

theorem **subst_jetN** {n n' g m : ℕ} (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (σ : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (hσ : ∀ p, σ (psJet m (MvPowerSeries.X p)) = jetN m (fun w => B w p.1 p.2)) (f : (Fin n → Fin g → ℂ) → ℂ) (hf : ContDiffAt ℂ ω (fun v => f (colsOf n g v)) 0) : σ (jetN m f) = jetN m (fun w => f (B w))

theorem **isHolFam_clm** {Y : Scheme.{0}} {N N' : ℕ} (e : (Fin N → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (T : (Fin N' → ℂ) →L[ℂ] (Fin N → ℂ)) : IsHolFam (fun v => e (T v))

theorem **jetPt_clm** {Y : Scheme.{0}} {N N' : ℕ} (e : (Fin N → ℂ) → (Sp ℂ ⟶ Y)) (hol : IsHolFam e) (T : (Fin N' → ℂ) →L[ℂ] (Fin N → ℂ)) (e' : (Fin N' → ℂ) → (Sp ℂ ⟶ Y)) (hol' : IsHolFam e') (he : ∀ v, e' v = e (T v)) (m : ℕ) : spMap ((jetMap (anLin T) (anLin_mem T) : Jet ℂ (Fin 1 × Fin N) m →ₐ[ℂ] Jet ℂ (Fin 1 × Fin N') m) : Jet ℂ (Fin 1 × Fin N) m →+* Jet ℂ (Fin 1 × Fin N') m) ≫ jetPt e hol m = jetPt e' hol' m

theorem **spMap_closedPoint** {R R' : Type} [CommRing R] [CommRing R'] [IsLocalRing R] [IsLocalRing R'] (f : R →+* R') (hf : IsLocalHom f) : spMap f (IsLocalRing.closedPoint R') = IsLocalRing.closedPoint R

theorem **appTop_toΓ** {R R' : Type} [CommRing R] [CommRing R'] (f : R →+* R') (a : R) : ((spMap f).appTop).hom (toΓ R a) = toΓ R' (f a)

theorem **res_pull** {Y Z : Scheme.{0}} (s : Z ⟶ Y) (W : Y.Opens) (W' : Z.Opens) (h : ⊤ ≤ W) (h' : ⊤ ≤ W') (le : W' ≤ s ⁻¹ᵁ W) (x : Γ(Y, W)) : (Z.presheaf.map (homOfLE h').op).hom ((Z.presheaf.map (homOfLE le).op).hom ((s.app W).hom x)) = s.appTop.hom ((Y.presheaf.map (homOfLE h).op).hom x)

theorem **jetMap_sub** {n n' m : ℕ} (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (σ : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (hσ : ∀ p, σ (psJet m (MvPowerSeries.X p)) = jetN m (fun w => B w p.1 p.2)) (ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left) (Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : Φ.JetMap m ψ Λ) : Φ.JetMap m (spMap (σ : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m) ≫ ψ) (Λ ∘ₗ B)

theorem **jetMap_closedPoint** {n m : ℕ} (ψ : Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left) (Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : Φ.JetMap m ψ Λ) : ψ (IsLocalRing.closedPoint (Jet ℂ (Fin n × Fin g) m)) = (Φ.e 0).left PullbackSections.cpt

theorem **contDiffAt_fr** {n : ℕ} (j : L.L.ι) (hj : (Φ.e 0).left PullbackSections.cpt ∈ L.L.U j) (a : ℤ) (Λ : (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) : ContDiffAt ℂ ω (fun v : Fin (n * g) → ℂ => Φ.fr j (Λ (colsOf n g v)) ^ a) 0

theorem **jetRd_sub** {n n' m : ℕ} (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (σ : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (hσ : ∀ p, σ (psJet m (MvPowerSeries.X p)) = jetN m (fun w => B w p.1 p.2)) {K : Type} [Fintype K] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (h : Φ.JetRd m φ ε τ Λ) : Φ.JetRd m (fun k => spMap (σ : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m) ≫ φ k) ε (LB.pullGen L.L φ ε τ (spMap (σ : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m))) (fun k => Λ k ∘ₗ B)

theorem **jetRd_linSubst** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (n n' m : ℕ) (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (sb : Fin n × Fin g → MvPolynomial (Fin n' × Fin g) ℂ), (∀ p, (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (psJet m (MvPowerSeries.X p)) = jetN m (fun w => B w p.1 p.2)) → ∀ (K : Type) [Fintype K] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.JetRd m φ ε τ Λ → Φ.JetRd m (fun k => spMap (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m) ≫ φ k) ε (LB.pullGen L.L φ ε τ (spMap (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m))) (fun k => Λ k ∘ₗ B)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.JetReadingLinearSubstitutionPointsStatement

theorem **_root_.GlobalTheory.T.jetReading_linearSubstitution_points** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (n n' m : ℕ) (B : (Fin n' → Fin g → ℂ) →ₗ[ℂ] (Fin n → Fin g → ℂ)) (sb : Fin n × Fin g → MvPolynomial (Fin n' × Fin g) ℂ), (∀ p, (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →ₐ[ℂ] Jet ℂ (Fin n' × Fin g) m) (psJet m (MvPowerSeries.X p)) = jetN m (fun w => B w p.1 p.2)) → ∀ (K : Type) [Fintype K] (φ : K → (Sp (Jet ℂ (Fin n × Fin g) m) ⟶ A.left)) (ε : K → ℤ) (τ : LB.Gen (LB.LinComb L.L φ ε)) (Λ : K → (Fin n → Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)), Φ.JetRd m φ ε τ Λ → Φ.JetRd m (fun k => spMap (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m) ≫ φ k) ε (LB.pullGen L.L φ ε τ (spMap (jetSubstP m sb : Jet ℂ (Fin n × Fin g) m →+* Jet ℂ (Fin n' × Fin g) m))) (fun k => Λ k ∘ₗ B)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.JetTranslateProjectedVacuumLe

theorem **polyDeriv_rename** {σ κ W : Type} [NormedAddCommGroup W] [NormedSpace ℂ W] {F : W → ℂ} (hF : ContDiff ℂ (⊤ : ℕ∞) F) (ρ : σ → κ) (u : κ → W) (P : MvPolynomial σ ℂ) (x : W) : I2.polyDeriv (MvPolynomial.rename ρ P) u F x = I2.polyDeriv P (fun l => u (ρ l)) F x

theorem **sum_norm_coeff_rename** {σ κ : Type} {ρ : σ → κ} (hρ : Function.Injective ρ) (P : MvPolynomial σ ℂ) : ∑ m ∈ (MvPolynomial.rename ρ P).support, ‖(MvPolynomial.rename ρ P).coeff m‖ = ∑ m ∈ P.support, ‖P.coeff m‖

theorem **polyDeriv_smul_dir** {σ W : Type} [NormedAddCommGroup W] [NormedSpace ℂ W] {P : MvPolynomial σ ℂ} {d : ℕ} (hP : P.IsHomogeneous d) (c : ℂ) (u : σ → W) (F : W → ℂ) (x : W) : I2.polyDeriv P (fun l => c • u l) F x = c ^ d * I2.polyDeriv P u F x

theorem **thA_heisG_sum** (idx : Fin g ≃ X.Ψ × Fin r) (hs : IsSympl X bas) (hn : Even n) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (y : Matrix (Fin n) (Fin r) E) {J : Type} (s : Finset J) (a : ℂ) (c : J → ℂ) (f : J → TF n g) (hf : ∀ i, IsSB (f i)) (z : Fin n → Fin g → ℂ) : thA X bas idx (heisG X bas y (a • ∑ i ∈ s, c i • f i)) x z = a * ∑ i ∈ s, c i * thA X bas idx (heisG X bas y (f i)) x z

theorem **rowPt_apply** (y : Matrix (Fin n) (Fin r) E) (b : Fin n) : rowPt idx x ψ y b = ψ (pv X idx x (y b))

theorem **map_pv_smul** (e : Fin g ≃ I2.Idx dim) (c : ℂ) (he : ∀ j, (e j).1 = rowType X (idx j) ∧ ψ (Pi.single j 1) = c • EuclideanSpace.single (e j) (1 : ℂ)) (a : E) (u : Fin r → E) : ψ (pv X idx x (a • u)) = I2.actE a (ψ (pv X idx x u))

theorem **mem_tensorLat_iff** (idx : Fin g ≃ X.Ψ × Fin r) (x : Frame X.Ψ r) (T : I2.PT E) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim) {L : Lt E r} (hL : ∀ ga, ga ∈ T.L ↔ ∃ u ∈ L, ga = ψ (pv X idx x u)) (hψ : ∀ (a : E) u, ψ (pv X idx x (a • u)) = I2.actE a (ψ (pv X idx x u))) {Λ : Lt E n} (hΛ : Λ ∈ Vset E n 1) (ga : Fin n → I2.Vsp T.dim) : ga ∈ TensorLat.tensorLat T Λ ↔ ∃ y ∈ tens Λ L, ga = rowPt idx x ψ y

theorem **jet_projA_le** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ Λ ∈ Vset E n 1, ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (θ : (Fin g → ℂ) → ℂ) (κ : ℂ), IsTheta1 X bas idx (w : E) x L θ → MeanOne X bas idx x L θ → (∀ z : Fin n → Fin g → ℂ, ∏ b, θ (z b) = κ * thA X bas idx v x z) → ∀ lam : ℝ, 0 < lam → (∀ u ∈ L, u ≠ 0 → lam ^ 2 ≤ (hx X bas idx x (pv X idx x u) (pv X idx x u)).re) → ∀ (y : Matrix (Fin n) (Fin r) E) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) (d : ℕ), P.IsHomogeneous d → ‖κ * thP X hr bas idx (heisG X bas y (projA X bas (w : E) (tens Λ L) v)) P x‖ * Real.sqrt ((((stdL E n

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.JetTranslateProjectedVacuumLeStatement

theorem **_root_.GlobalTheory.T.jet_translate_projected_vacuum_le** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ Λ ∈ Vset E n 1, ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (θ : (Fin g → ℂ) → ℂ) (κ : ℂ), IsTheta1 X bas idx (w : E) x L θ → MeanOne X bas idx x L θ → (∀ z : Fin n → Fin g → ℂ, ∏ b, θ (z b) = κ * thA X bas idx v x z) → ∀ lam : ℝ, 0 < lam → (∀ u ∈ L, u ≠ 0 → lam ^ 2 ≤ (hx X bas idx x (pv X idx x u) (pv X idx x u)).re) → ∀ (y : Matrix (Fin n) (Fin r) E) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) (d : ℕ), P.IsHomogeneous d → ‖κ * thP X hr bas idx (heisG X bas y (projA X bas (w : E) (

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.L1NormRowBlockLeadingMinorProductLe

theorem **l1Norm_rowBlockLinearForm_le** (t : T) (r : Fin p ⊕ Fin q) (v : Fin n → ℂ) {R : ℝ} (hR : 0 ≤ R) (hv : ∑ k, ‖v k‖ ^ 2 ≤ R ^ 2) : l1Norm (rowForm t r v) ≤ √(n : ℝ) * R

theorem **l1Norm_det_le_of_entry_bound** {σ : Type*} {i N : ℕ} (hi : i ≤ N) (M : Matrix (Fin i) (Fin i) (MvPolynomial σ ℂ)) {B : ℝ} (hB : 0 ≤ B) (h : ∀ a b, l1Norm (M a b) ≤ B) : l1Norm M.det ≤ (N * B) ^ i

theorem **l1Norm_rowBlockLeadingMinorProduct_le** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → Fin n → ℂ) (φ : T → Fin q → Fin n → ℂ) {R : ℝ} (hR : 0 ≤ R) (hε : ∀ t a, ∑ k, ‖ε t a k‖ ^ 2 ≤ R ^ 2) (hφ : ∀ t b, ∑ k, ‖φ t b k‖ ^ 2 ≤ R ^ 2) {N : ℕ} (hp : p ≤ N) (hq : q ≤ N) : l1Norm (minorProd c c' ε φ) ≤ (N * √(n : ℝ) * R) ^ typeDegree c c'

@[simp] theorem **embSeminorm_apply** (τ : E →+* ℂ) (x : E) : embSeminorm τ x = ‖τ x‖

@[simp] theorem **embVec_apply** (τ : E →+* ℂ) (v : κ → E) (k : κ) : embVec τ v k = τ (v k)

theorem **embVec_smul** (τ : E →+* ℂ) (x : E) (v : κ → E) : embVec τ (x • v) = τ x • embVec τ v

theorem **embLength_apply** (τ : E →+* ℂ) (v : κ → E) : embLength τ v = ‖embVec τ v‖

theorem **embLength_sq** (τ : E →+* ℂ) (v : κ → E) : embLength τ v ^ 2 = ∑ k, ‖τ (v k)‖ ^ 2

theorem **embLength_smul** (τ : E →+* ℂ) (x : E) (v : κ → E) : embLength τ (x • v) = ‖τ x‖ * embLength τ v

theorem **embLength_le_abs** (τ : E →+* ℂ) {v : κ → E} {R : ℝ} (h : ∑ k, ‖τ (v k)‖ ^ 2 ≤ R ^ 2) : embLength τ v ≤ |R|

theorem **sum_sq_le_of_embLength_le** (τ : E →+* ℂ) {v : κ → E} {R : ℝ} (h : embLength τ v ≤ |R|) : ∑ k, ‖τ (v k)‖ ^ 2 ≤ R ^ 2

theorem **norm_emb_sum_mul_le** (τ : E →+* ℂ) (s : E → E) (hs : ∀ x, ‖τ (s x)‖ = ‖τ x‖) (x y : κ → E) : ‖τ (∑ k, x k * s (y k))‖ ≤ embLength τ x * embLength τ y

theorem **sum_sq_combination_le** (τ : E →+* ℂ) {ι : Type*} [Fintype ι] (v : ι → κ → E) {ρ : ℝ} (hv : ∀ i, ∑ k, ‖τ (v i k)‖ ^ 2 ≤ ρ ^ 2) (A : ι → ℕ) {d : ℕ} (hA : ∀ i, A i ≤ d) : ∑ k, ‖τ ((∑ i, (A i : E) • v i) k)‖ ^ 2 ≤ ((Fintype.card ι : ℝ) * d * ρ) ^ 2

theorem **perpVec_eq_perpVector** {R V : Type} [CommRing R] [AddCommGroup V] [Module R V] {p : ℕ} (h : V →ₗ[R] (V → R)) (e e' : Fin p → V) (g : V) : perpVec h e e' g = perpVector (fun a => (LinearMap.proj (e a)).comp h) e' g

theorem **sum_sq_perpVec_le** (τ : E →+* ℂ) (h : (Fin n → E) →ₗ[E] ((Fin n → E) → E)) (hh : ∀ x y, h x y = ∑ k, x k * cc (y k)) (e e' : Fin p → Fin n → E) (g : Fin n → E) {R : ℝ} (he : ∀ a, ∑ k, ‖τ (e a k)‖ ^ 2 ≤ R ^ 2) (he' : ∀ a, ∑ k, ‖τ (e' a k)‖ ^ 2 ≤ R ^ 2) (hg : ∑ k, ‖τ (g k)‖ ^ 2 ≤ R ^ 2) : ∑ k, ‖τ (perpVec h e e' g k)‖ ^ 2 ≤ (((p + 1).factorial : ℝ) * R ^ (2 * p + 1)) ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.L2CanThEqInvSqrtTwoPowMulDetIm

theorem **realPt_apply** (p : Fin g ⊕ Fin g → ℝ) : realPt τ p = (fun i => (p (.inl i) : ℂ)) + τ *ᵥ fun i => (p (.inr i) : ℂ)

theorem **realPt_injective** (hs : τ.IsSymm) (hp : (τ.map Complex.im).PosDef) : Function.Injective (realPt τ)

theorem **norm_canTh_realPt** (hp : (τ.map Complex.im).PosDef) (a b : Fin g → ℝ) (p : Fin g ⊕ Fin g → ℝ) : ‖canTh τ a b (realPt τ p)‖ ^ 2 * Real.exp (-π * (hermForm (imInv τ) (realPt τ p) (realPt τ p)).re) = ‖thetaChar τ (fun i => (a i : ℂ)) (fun i => (b i : ℂ)) (realPt τ p)‖ ^ 2 * Real.exp (-2 * π * ((fun i => p (.inr i)) ⬝ᵥ (τ.map Complex.im) *ᵥ fun i => p (.inr i)))

theorem **apply_realCoord** (p : Fin g ⊕ Fin g → ℝ) : φ (realCoord hs hp φ p) = realPt τ p

theorem **mem_lat_perBasis** (y : EuclideanSpace ℂ (Fin g)) : y ∈ lat (perBasis hs hp φ) ↔ ∃ m : Fin g ⊕ Fin g → ℤ, y = realCoord hs hp φ fun j => (m j : ℝ)

theorem **im_Hf_realCoord** (p q : Fin g ⊕ Fin g → ℝ) : (Hf (realCoord hs hp φ p) (realCoord hs hp φ q)).im = ∑ i, p (.inr i) * q (.inl i) - ∑ i, p (.inl i) * q (.inr i)

private theorem **eq_realCoord_of_eq_int_add_tau_mulVec** {y : EuclideanSpace ℂ (Fin g)} {m n : Fin g → ℤ} (h : φ y = (fun i => (m i : ℂ)) + τ *ᵥ fun i => (n i : ℂ)) : y = realCoord hs hp φ fun j => ((Sum.elim m n j : ℤ) : ℝ)

theorem **mem_lat_torus** (y : EuclideanSpace ℂ (Fin g)) : y ∈ lat (torus hs hp φ hφ a b).b ↔ ∃ m n : Fin g → ℤ, φ y = (fun i => (m i : ℂ)) + τ *ᵥ fun i => (n i : ℂ)

theorem **torus_ka** {y : EuclideanSpace ℂ (Fin g)} {m n : Fin g → ℤ} (h : φ y = (fun i => (m i : ℂ)) + τ *ᵥ fun i => (n i : ℂ)) : (torus hs hp φ hφ a b).ka y = canMult a b m n

theorem **canTh_mem_Sec** : (fun v => canTh τ a b (φ v)) ∈ (torus hs hp φ hφ a b).Sec 1

private theorem **norm_sq_eq_re_Hf** (w : EuclideanSpace ℂ (Fin g)) : ‖w‖ ^ 2 = (Hf w w).re

theorem **l2_canTh_eq_inv_sqrt_two_pow_mul_det_im** : (torus hs hp φ hφ a b).l2 1 (fun v => canTh τ a b (φ v)) = (Real.sqrt (2 ^ g * (τ.map Complex.im).det))⁻¹

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.L2OneEqIntegralFundamentalDomainDivVolume

theorem **Hf_add_smul_left** {κ : Type} [Fintype κ] (c : ℂ) (v w u : EuclideanSpace ℂ κ) : I2.Hf (v + c • w) u = I2.Hf v u + c * I2.Hf w u

theorem **Hf_neg_right** {κ : Type} [Fintype κ] (v w : EuclideanSpace ℂ κ) : I2.Hf v (-w) = -I2.Hf v w

theorem **Hf_neg_left** {κ : Type} [Fintype κ] (v w : EuclideanSpace ℂ κ) : I2.Hf (-v) w = -I2.Hf v w

theorem **cexp_pi_int** (k : ℤ) : cexp (Real.pi * I * (k : ℂ)) = ((Real.cos (Real.pi * k) : ℝ) : ℂ)

theorem **l2_one_eq_integral_fundamentalDomain_div_volume** {E : Type} [Field E] [NumberField E] {g : ℕ} (T : I2.PT E) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim) (θ : (Fin g → ℂ) → ℂ) (hs : (fun v : Fin 1 → I2.Vsp T.dim => θ (ψ.symm (v 0))) ∈ T.Sec 1 1) (A : AddSubgroup (Fin g → ℂ)) (hL : ∀ z, z ∈ A ↔ ψ z ∈ T.L) (Fd : Set (Fin g → ℂ)) (hFd : IsAddFundamentalDomain A Fd volume) : T.l2 1 (fun v : Fin 1 → I2.Vsp T.dim => θ (ψ.symm (v 0))) = (∫ z in Fd, ‖θ z‖ ^ 2 * Real.exp (-Real.pi * (I2.Hf (ψ z) (ψ z)).re)) / (volume Fd).toReal

theorem **theta_on_torus** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g : ℕ) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (w : 𝒪 E) (L : Lt E r) (x : Frame X.Ψ r) (t : ℝ) (T : I2.PT E) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim) (e : Fin g ≃ I2.Idx T.dim), IsSympl X bas → x ∈ X.dom → (w : E) + cc (w : E) = 1 → (T.r = r ∧ (∀ z z', I2.Hf (ψ z) (ψ z') = hx X bas idx x z z') ∧ (∀ ga, ga ∈ T.L ↔ ∃ u ∈ L, ga = ψ (pv X idx x u)) ∧ (∀ (a : 𝒪 E) u, ψ (pv X idx x ((a : E) • u)) = I2.act a (ψ (pv X idx x u))) ∧ (∀ ga ∈ T.L, (T.eps ga : ℝ) = Real.cos (Real.pi * (I2.Hf ga (I2.act w ga)).im)) ∧ (∀ j, (e j).1 = rowType X (idx j) ∧ ψ (Pi.single j 1) = ((Real.sqrt t : ℝ) : ℂ) • EuclideanSpace.single (e j) (1 : ℂ))) → (∀ θ : (Fin g → ℂ) → ℂ, IsTheta1 X bas idx (w : E) x L θ → (fun v : Fin 1 → I2.Vsp T.dim => θ (ψ.symm (v 0))) ∈ T.Sec 1 1 ∧ (MeanOne X bas idx x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeExpOfLogLe

theorem **le_exp_of_log_le** {x b : ℝ} (h : Real.log x ≤ b) (hx : 0 ≤ x) : x ≤ Real.exp b

theorem **log_le_of_le_exp** {x b : ℝ} (h : x ≤ Real.exp b) (hx : 0 ≤ x) (hb : 0 ≤ b) : Real.log x ≤ b

theorem **supN_sum_le** {D : D7} (B : WeightFamilies.Prov D) (j : ℕ) (w : WeightFamilies.Wt D j) (σ : B.KO →+* ℂ) {ι : Type} (s : Finset ι) (F : ι → B.Cusp j w) : B.supN j w σ (∑ i ∈ s, F i) ≤ ∑ i ∈ s, B.supN j w σ (F i)

theorem **log_supN_bounds** : ∀ (D : D7) (B : WeightFamilies.Prov D) (j : ℕ) (w : WeightFamilies.Wt D j), (∀ (σ : B.KO →+* ℂ) (s : B.KO) (F : B.Cusp j w) (a b : ℝ), 0 ≤ a + b → Real.log ‖σ s‖ ≤ a → Real.log (B.supN j w σ F) ≤ b → 0 ≤ b → 0 ≤ a → Real.log (B.supN j w σ (s • F)) ≤ a + b) ∧ (∀ (σ : B.KO →+* ℂ) (N : ℕ) (c : Fin N → ℕ) (F : Fin N → B.Cusp j w) (b : ℝ), 0 ≤ b → (∀ i, c i ≤ N) → (∀ i, Real.log (B.supN j w σ (F i)) ≤ b) → Real.log (B.supN j w σ (∑ i, (c i : B.KO) • F i)) ≤ 2 * Real.log N + b) ∧ ∀ b : ℝ, (∑ _σ : B.KO →+* ℂ, b) / (Fintype.card (B.KO →+* ℂ) : ℝ) = b

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeMulNormFlatOfSelfDual

theorem **norm_sq_contr_apply_le** (γ : Fin n → Vsp dim) (w : Fin n → E) (i : Idx dim) : ‖contr γ w i‖ ^ 2 ≤ (∑ k, ‖i.1 (w k)‖ ^ 2) * ∑ k, ‖γ k i‖ ^ 2

theorem **norm_sq_contr_le** (γ : Fin n → Vsp dim) (w : Fin n → E) {R : ℝ} (hw : ∀ τ : E →+* ℂ, ∑ k, ‖τ (w k)‖ ^ 2 ≤ R) : ‖contr γ w‖ ^ 2 ≤ R * ‖flat γ‖ ^ 2

theorem **norm_contr_le** (γ : Fin n → Vsp dim) (w : Fin n → E) {ρ : ℝ} (hρ : 0 ≤ ρ) (hw : ∀ τ : E →+* ℂ, ∑ k, ‖τ (w k)‖ ^ 2 ≤ ρ ^ 2) : ‖contr γ w‖ ≤ ρ * ‖flat γ‖

theorem **exists_contr_ne_zero** {ι : Type*} (v : ι → Fin n → E) (hspan : Submodule.span E (Set.range v) = ⊤) {γ : Fin n → Vsp dim} (hγ : γ ≠ 0) : ∃ i, contr γ (v i) ≠ 0

theorem **le_mul_norm_flat** (X : PT E) (Λ' : Submodule (𝒪 E) (Fin n → E)) {ι : Type*} (v : ι → Fin n → E) (hv : ∀ i, v i ∈ Λ') (hspan : Submodule.span E (Set.range v) = ⊤) {ρ : ℝ} (hρ : 0 ≤ ρ) (hlen : ∀ i (τ : E →+* ℂ), ∑ k, ‖τ (v i k)‖ ^ 2 ≤ ρ ^ 2) {lam : ℝ} (hlam : ∀ x ∈ X.L, x ≠ 0 → lam ≤ ‖x‖) {γ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') (h0 : γ ≠ 0) : lam ≤ ρ * ‖flat γ‖

theorem **span_eq_top_of_rat** {S : Set (Fin n → E)} (h : Submodule.span ℚ S = ⊤) : Submodule.span E S = ⊤

theorem **le_mul_norm_flat_of_selfDual** (X : PT E) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') {lam : ℝ} (hlam : ∀ x ∈ X.L, x ≠ 0 → lam ≤ ‖x‖) {γ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') (h0 : γ ≠ 0) : lam ≤ 2 * (fdeg E : ℝ) ^ ((3 : ℝ) / 2) * n ^ 2 * rdE E * ‖flat γ‖

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeadingMinorProductHarmonicGenerates

theorem **rename_letAct_invTranspose_eq_colAct** {Ix : Type} [Fintype Ix] {pl : Ix → (E →+* ℂ)} (hpl : IsEnum H pl) (K : Type) [Field K] (g : {τ : E →+* ℂ // InPsi H τ} → GL (Fin n) K) (P : MvPolynomial (LtN Dw n) K) : rename (thetaLetterEquiv Dw n) (letAct Dw pl K (fun t => invTranspose (g (pl t, (hpl.2 _).2 ⟨t, rfl⟩)))) P) = colAct K _ Dw.p Dw.q (Fin n) g (rename (thetaLetterEquiv Dw n) P)

theorem **rename_minorP_of_mem** (hn : Dw.p + Dw.q ≤ n) {τ : E →+* ℂ} (h : InPsi H τ) (i : Fin Dw.p) (hi : i.val < Dw.mult τ) (hi' : i.val + 1 ≤ n) : rename (thetaLetterEquiv Dw n) (minorPolynomial (Dw

theorem **rename_minorP_conjugate** (hn : Dw.p + Dw.q ≤ n) {τ : E →+* ℂ} (h : InPsi H τ) (i : Fin Dw.q) (hi : i.val < Dw.mult (ComplexEmbedding.conjugate τ)) (hi' : n - (i.val + 1) + (i.val + 1) = n) : rename (thetaLetterEquiv Dw n) (minorPolynomial (Dw

theorem **rename_pG** [Fintype {τ : E →+* ℂ // InPsi H τ}] (hn : Dw.p + Dw.q ≤ n) (a : (E →+* ℂ) × ℕ → ℕ) : ∃ u : ℤˣ, rename (thetaLetterEquiv Dw n) (pG (Dw

theorem **pG_mem_ne_zero_generates** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : Datum H) (n : ℕ) (Ix : Type) [Fintype Ix] (pl : Ix → (E →+* ℂ)), IsEnum H pl → Dw.p + Dw.q ≤ n → ∀ (l : Level H) (c : Coeff l) (a : (E →+* ℂ) × ℕ → ℕ), (∀ τ, Antitone fun i => a (τ, i)) → pG (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeadingProductsIndependentModuloDeepProducts

private theorem **isBigO_norm_pow_of_mem_span_monomial_mul** {Y : Type} {D : ℕ} (a : ℕ) (x : Y) (s : Fin D → Y → ℂ) (H : Set (Y → ℂ)) (φ : V D → Y) (U : Set (V D)) (hU : IsOpen U) (h0U : (0 : V D) ∈ U) (hφ0 : φ 0 = x) (hs : ∀ k, DifferentiableOn ℂ (fun z => s k (φ z)) U) (hH : ∀ f ∈ H, DifferentiableOn ℂ (fun z => f (φ z)) U) (hs0 : ∀ k, s k x = 0) (g : Y → ℂ) (hg : g ∈ Submodule.span ℂ {g | ∃ (α : Fin D → ℕ) (f : Y → ℂ), ∑ i, α i = a ∧ f ∈ H ∧ g = (∏ i, s i ^ α i) * f}) : (fun z => g (φ z)) =O[𝓝 0] fun z : V D => ‖z‖ ^ a

private theorem **coeff_lemma_3** {D : ℕ} (M : Finset (Fin D → ℕ)) (A : (Fin D → ℕ) → ℂ) (h : ∀ y : Fin D → ℂ, ∑ α ∈ M, A α * ∏ q, y q ^ α q = 0) : ∀ α ∈ M, A α = 0

private theorem **gs_id_linComb_eq_trF_mul_sval** {Y : Type} {D m : ℕ} (T a d : ℕ) (x : Y) (s : Fin D → Y → ℂ) (w0 : Y → ℂ) (θ : Fin m → Y → ℂ) (H : Set (Y → ℂ)) (φ : V D → Y) (U : Set (V D)) (hU : IsOpen U) (h0U : (0 : V D) ∈ U) (hφ0 : φ 0 = x) (hs : ∀ k, DifferentiableOn ℂ (fun z => s k (φ z)) U) (hw : DifferentiableOn ℂ (fun z => w0 (φ z)) U) (hθ : ∀ i, DifferentiableOn ℂ (fun z => θ i (φ z)) U) (hH : ∀ f ∈ H, DifferentiableOn ℂ (fun z => f (φ z)) U) (hs0 : ∀ k, s k x = 0) (hw0 : w0 x ≠ 0) (hdet : (fderiv ℂ (fun z => (WithLp.toLp 2 fun k => s k (φ z) / w0 (φ z) : V D)) 0).det ≠ 0) (c : (Fin D → ℕ) → Fin m → ℂ) (hlow : ∀ α ∈ multiLt D a, ∀ k, c α k ≠ 0 → d ≤ ∑ q, α q) (hda : d < a) (hmem : (fun y => ∑ α ∈ multiLt D a, ∑ k, c α k * (∏ q, s q y ^ α q) * w0 y ^ (T - ∑ q, α q) * θ k y) ∈ Submodule.span ℂ {g | ∃ (α : Fin D → ℕ) (f : Y → ℂ), ∑ i, α i = a ∧ f ∈ H ∧ g = (∏ i, s i ^ α i) * f}) : ∀

theorem **coeff_eq_zero** {Y : Type} {δ D m : ℕ} (T a : ℕ) (xs : Fin δ → Y) (s : Fin D → Y → ℂ) (w0 : Y → ℂ) (θ : Fin m → Y → ℂ) (H : Set (Y → ℂ)) (hch : ∀ q, ∃ (n : ℕ) (hn : n = D) (φ : V n → Y) (U : Set (V n)), IsOpen U ∧ 0 ∈ U ∧ φ 0 = xs q ∧ (∀ k, DifferentiableOn ℂ (fun z => s k (φ z)) U) ∧ DifferentiableOn ℂ (fun z => w0 (φ z)) U ∧ (∀ i, DifferentiableOn ℂ (fun z => θ i (φ z)) U) ∧ (∀ f ∈ H, DifferentiableOn ℂ (fun z => f (φ z)) U) ∧ (∀ k, s k (xs q) = 0) ∧ w0 (xs q) ≠ 0 ∧ (fderiv ℂ (fun z => (WithLp.toLp 2 fun k => s (Fin.cast hn k) (φ z) / w0 (φ z) : V n)) 0).det ≠ 0) (hθ : LinearIndependent ℂ (vals xs θ)) (c : (Fin D → ℕ) → Fin m → ℂ) (hmem : (fun y => ∑ α ∈ multiLt D a, ∑ k, c α k * (∏ q, s q y ^ α q) * w0 y ^ (T - ∑ q, α q) * θ k y) ∈ Submodule.span ℂ {g | ∃ (α : Fin D → ℕ) (f : Y → ℂ), ∑ i, α i = a ∧ f ∈ H ∧ g = (∏ i, s i ^ α i) * f}) : ∀ α ∈ multiLt D a, ∀ k, c α k = 0

theorem **indep_disjoint** {Y : Type} {δ D m : ℕ} (T a : ℕ) (xs : Fin δ → Y) (s : Fin D → Y → ℂ) (w0 : Y → ℂ) (θ : Fin m → Y → ℂ) (H : Set (Y → ℂ)) (hch : ∀ q, ∃ (n : ℕ) (hn : n = D) (φ : V n → Y) (U : Set (V n)), IsOpen U ∧ 0 ∈ U ∧ φ 0 = xs q ∧ (∀ k, DifferentiableOn ℂ (fun z => s k (φ z)) U) ∧ DifferentiableOn ℂ (fun z => w0 (φ z)) U ∧ (∀ i, DifferentiableOn ℂ (fun z => θ i (φ z)) U) ∧ (∀ f ∈ H, DifferentiableOn ℂ (fun z => f (φ z)) U) ∧ (∀ k, s k (xs q) = 0) ∧ w0 (xs q) ≠ 0 ∧ (fderiv ℂ (fun z => (WithLp.toLp 2 fun k => s (Fin.cast hn k) (φ z) / w0 (φ z) : V n)) 0).det ≠ 0) (hθ : LinearIndependent ℂ (vals xs θ)) : LinearIndependent ℂ (fun p : {α : Fin D → ℕ // ∑ i, α i < a} × Fin m => (∏ i, s i ^ p.1.1 i) * w0 ^ (T - ∑ i, p.1.1 i) * θ p.2) ∧ Disjoint (Submodule.span ℂ (Set.range fun p : {α : Fin D → ℕ // ∑ i, α i < a} × Fin m => (∏ i, s i ^ p.1.1 i) * w0 ^ (T - ∑ i, p.1.1 i) * θ p.2)) (Sub

theorem **main** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ⟨(Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _⟩⟩) → ∀ (m : ℕ) (θ : Fin m → R.Y → ℂ), (∀ i, θ i ∈ R.sp wF Set.univ) → LinearIndependent ℂ (vals xs θ) → LinearIndependent ℂ (R.lead w t Tn (a0 D j Tn) θ) ∧ Disjoint (Submodule.span ℂ (Set.range (R.lead w t Tn (a0 D j Tn) θ))) (R.deep w t ((recPlusN h j hj n).wF0 wL wF) (a0 D j Tn))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeadingProductsIndependentModuloDeepProductsStatement

theorem **_root_.GlobalTheory.T.leadingProducts_independent_modulo_deepProducts** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _)⟩) → ∀ (m : ℕ) (θ : Fin m → R.Y → ℂ), (∀ i, θ i ∈ R.sp wF Set.univ) → LinearIndependent ℂ (vals xs θ) → LinearIndependent ℂ (R.lead w t Tn (a0 D j Tn) θ) ∧ Disjoint (Submodule.span ℂ (Set.range (R.lead w t Tn (a0 D j Tn) θ))) (R.deep w t ((recPlusN h j hj n).wF0 wL wF) (a0 D j Tn))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LevelCoverLargerCoeffFieldLyingOver

theorem **frOver_isLev_ddBC** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (n : ℕ) (d : DeltaDat Xo c n), (∀ (k : Type) [CommRing k] [Algebra c.F k] [Algebra c'.F k] [Algebra d.c₂.F k] [A₂ : Algebra (covCoeff hle d).F k] (h : AlgOver c c' hle k) (h₂ : AlgOver d.c₂ (covCoeff hle d) (le_covCoeff hle d) k) (y₂ : FrPt d.Xo₂ (covCoeff hle d) k) (y : FrPt Xo c' k), @FrOver E _ _ _ H Dw l Xo c' n (ddBC hle d) k _ _ A₂ y₂ y ↔ FrOver d (y₂.res (le_covCoeff hle d) h₂) (y.res hle h)) ∧ (d.IsLev → (ddBC hle d).IsLev)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LevelCoverLargerCoeffFieldLyingOverStatement

theorem **_root_.GlobalTheory.T.levelCover_larger_coeffField_lying_over** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (n : ℕ) (d : DeltaDat Xo c n), (∀ (k : Type) [CommRing k] [Algebra c.F k] [Algebra c'.F k] [Algebra d.c₂.F k] [A₂ : Algebra (covCoeff hle d).F k] (h : AlgOver c c' hle k) (h₂ : AlgOver d.c₂ (covCoeff hle d) (le_covCoeff hle d) k) (y₂ : FrPt d.Xo₂ (covCoeff hle d) k) (y : FrPt Xo c' k), @FrOver E _ _ _ H Dw l Xo c' n (ddBC hle d) k _ _ A₂ y₂ y ↔ FrOver d (y₂.res (le_covCoeff hle d) h₂) (y.res hle h)) ∧ (d.IsLev → (ddBC hle d).IsLev)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LevelFourierTransformAutAndHomogeneous

theorem **exists_zetaM_pow** (c : Coeff l) (ta : c.F ≃+* c.F) : ∃ a a' : ℕ, ta (zetaM c) = zetaM c ^ a ∧ a * a' ≡ 1 [MOD l.M]

theorem **mk_qRep** {K : Submodule (𝒪 E) Dw.La} (x : Dw.La ⧸ K) : (Submodule.Quotient.mk (qRep x) : Dw.La ⧸ K) = x

theorem **level_nsmul** (x : LevQc Dw l) : l.M • x = 0

theorem **nsmul_nsmul_of_modEq** {n a a' : ℕ} (h : a * a' ≡ 1 [MOD l.M]) (x : Fin n → LevQc Dw l) : a • a' • x = x

theorem **pairL_nsmul** {n : ℕ} (a : ℕ) (x : Fin n → LevQc Dw l) (y : Fin n → Dw.La) : (l.M : ℤ) ∣ (a : ℤ) * pairL (l

theorem **map_phiHat** (c : Coeff l) {n : ℕ} (ta : c.F ≃+* c.F) {a a' : ℕ} (ha : ta (zetaM c) = zetaM c ^ a) (h : a * a' ≡ 1 [MOD l.M]) (φ : (Fin n → LevQc Dw l) → c.F) (q : Fin n → LevQ Dw l) : ta (phiHat c φ q) = phiHat c (fun x => ta (φ (a' • x))) q

theorem **phiHat_mul** (c : Coeff l) {n : ℕ} (u : c.F) (φ : (Fin n → LevQc Dw l) → c.F) (q : Fin n → LevQ Dw l) : phiHat c (fun x => u * φ x) q = u * phiHat c φ q

theorem **shortValAt_mul** {Xo : Moduli Dw l} {c : Coeff l} {n : ℕ} (nab : NabV Xo c) (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (b u : c.F) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S) (h : ShortValAt Xo c nab w Λ b N P y v) : ShortValAt Xo c nab (fun q => u * w q) Λ b N P y (algebraMap c.F S u * v)

theorem **phiHat_aut_and_mul** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ), (∀ ta : c.F ≃+* c.F, (∃ a a' : ℕ, ta (zetaM c) = zetaM c ^ a ∧ a * a' ≡ 1 [MOD l.M]) ∧ ∀ a a' : ℕ, ta (zetaM c) = zetaM c ^ a → a * a' ≡ 1 [MOD l.M] → ∀ (φ : (Fin n → LevQc Dw l) → c.F) (q : Fin n → LevQ Dw l), ta (phiHat c φ q) = phiHat c (fun x => ta (φ (a' • x))) q) ∧ (∀ (u : c.F) (φ : (Fin n → LevQc Dw l) → c.F) q, phiHat c (fun x => u * φ x) q = u * phiHat c φ q) ∧ ∀ (nab : NabV Xo c) (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (b u : c.F) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S), ShortValAt Xo c nab w Λ b N P y v → ShortValAt Xo c nab (fun q => u * w q) Λ b N P y (algebraMap c.F S u * v)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LevelFourierTransformAutAndHomogeneousStatement

theorem **_root_.GlobalTheory.T.level_fourier_transform_aut_and_homogeneous** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ), (∀ ta : c.F ≃+* c.F, (∃ a a' : ℕ, ta (zetaM c) = zetaM c ^ a ∧ a * a' ≡ 1 [MOD l.M]) ∧ ∀ a a' : ℕ, ta (zetaM c) = zetaM c ^ a → a * a' ≡ 1 [MOD l.M] → ∀ (φ : (Fin n → LevQc Dw l) → c.F) (q : Fin n → LevQ Dw l), ta (phiHat c φ q) = phiHat c (fun x => ta (φ (a' • x))) q) ∧ (∀ (u : c.F) (φ : (Fin n → LevQc Dw l) → c.F) q, phiHat c (fun x => u * φ x) q = u * phiHat c φ q) ∧ ∀ (nab : NabV Xo c) (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (b u : c.F) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S), ShortValAt Xo c nab w Λ b N P y v → ShortValAt Xo c nab (fun q => u * w q) Λ b N P y (algebraMap c.F S u * v)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LevelThreeSeparatesTangents

lemma **exists_mem_four_of_open_dense** {U : Set V} (ho : IsOpen U) (hd : Dense U) (e₁ e₂ e₃ e₄ : V ≃ₜ V) : ∃ x, e₁ x ∈ U ∧ e₂ x ∈ U ∧ e₃ x ∈ U ∧ e₄ x ∈ U

lemma **exists_linear_increment** {U : Set V} (ho : IsOpen U) (hd : Dense U) {f : V → ℂ} {c₀ : ℂ} (hf : ∀ a ∈ U, ∀ b ∈ U, -a - b ∈ U → c₀ + f a + f b + f (-a - b) = 0) {a₀ : V} (ha₀ : a₀ ∈ U) (hdf : DifferentiableAt ℂ f a₀) : ∃ L : V →L[ℂ] ℂ, ∀ a t, a ∈ U → a + t ∈ U → f (a + t) - f a = L t

lemma **hasFDerivAt_Hf_left** (γ v : EuclideanSpace ℂ ι) : ∃ A : EuclideanSpace ℂ ι →L[ℂ] ℂ, (∀ c, A c = Hf c γ) ∧ HasFDerivAt (fun v => Hf v γ) A v

lemma **fderiv_sec_add** (Y : GT ι) {θ : EuclideanSpace ℂ ι → ℂ} (hθ : θ ∈ Y.Sec 1) {γ : EuclideanSpace ℂ ι} (hγ : γ ∈ lat Y.b) (a c : EuclideanSpace ℂ ι) : fderiv ℂ θ (a + γ) c * θ a = θ (a + γ) * (Real.pi * Hf c γ * θ a + fderiv ℂ θ a c)

theorem **level_three_separates_tangents** {ι : Type} [Fintype ι] (Y : GT ι) (θ : EuclideanSpace ℂ ι → ℂ) (hθ : θ ∈ Y.Sec 1) (h0 : θ ≠ 0) (c₀ : ℂ) (c : EuclideanSpace ℂ ι) (h : ∀ a b, c₀ * (θ a * θ b * θ (-a - b)) + fderiv ℂ (fun v => θ (v + a) * θ (v + b) * θ (v - a - b)) 0 c = 0) : c = 0 ∧ c₀ = 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LevelTupleSumEqRational

theorem **coe_smul_mem** (Λ : Submodule (𝒪 E) (Fin r → E)) (c : 𝒪 E) {v : Fin r → E} (hv : v ∈ Λ) : (c : E) • v ∈ Λ

theorem **nat_smul_mem** (Λ : Submodule (𝒪 E) (Fin r → E)) (n : ℕ) {v : Fin r → E} (hv : v ∈ Λ) : (n : E) • v ∈ Λ

theorem **Gen.of_sub** {Λ : Submodule (𝒪 E) (Fin r → E)} {t t' : Fin r → Fin r → E} (h : Gen Λ t) (hs : ∀ j, t j - t' j ∈ Λ) : Gen Λ t'

theorem **exists_finite_res** (Λ : Submodule (𝒪 E) (Fin r → E)) (hΛ : Λ.FG) : ∃ S : Set (Fin r → E), S.Finite ∧ ∀ v, (25 : E) • v ∈ Λ → ∃ s ∈ S, v - s ∈ Λ

theorem **exists_reps** (Λ : Submodule (𝒪 E) (Fin r → E)) (hΛ : Λ.FG) : ∃ 𝒯 : Finset (Fin r → Fin r → E), (∀ t ∈ 𝒯, Gen Λ t) ∧ ∀ t, Gen Λ t → ∃! t', t' ∈ 𝒯 ∧ ∀ j, t j - t' j ∈ Λ

theorem **torus_r** {X : Data E r} {x : Frame X.Ψ r} (A : TorusAt X x) : A.T.r = r

theorem **map_injective** {X : Data E r} {x : Frame X.Ψ r} (hx : x ∈ X.dom) (A : TorusAt X x) : Function.Injective A.φ

theorem **map_nat_smul** {X : Data E r} {x : Frame X.Ψ r} (A : TorusAt X x) (n : ℕ) (v : Fin r → E) : A.φ ((n : E) • v) = (n : ℂ) • A.φ v

theorem **map_five** {X : Data E r} {x : Frame X.Ψ r} (A : TorusAt X x) (v : Fin r → E) : A.φ ((5 : E) • v) = (5 : ℂ) • A.φ v

theorem **map_twentyfive** {X : Data E r} {x : Frame X.Ψ r} (A : TorusAt X x) (v : Fin r → E) : A.φ ((25 : E) • v) = ((5 : ℂ) ^ 2) • A.φ v

theorem **map_mem_iff** {X : Data E r} {x : Frame X.Ψ r} (hx : x ∈ X.dom) (A : TorusAt X x) (v : Fin r → E) : A.φ v ∈ A.T.L ↔ v ∈ X.Λ

theorem **gen_iff** {X : Data E r} {x : Frame X.Ψ r} (hx : x ∈ X.dom) (A : TorusAt X x) (t : Fin r → Fin r → E) : Gen X.Λ t ↔ (∀ j, ((5 : ℂ) ^ 2) • A.φ (t j) ∈ A.T.L) ∧ ∀ z ∈ A.T.L, ∃ c : Fin r → 𝒪 E, ∃ y ∈ A.T.L, z = ∑ j, act (c j) (((5 : ℂ) ^ 2) • A.φ (t j)) + (5 : ℂ) • y

theorem **sum_eq** {X : Data E r} {x : Frame X.Ψ r} (hx : x ∈ X.dom) (A : TorusAt X x) (𝒯 : Finset (Fin r → Fin r → E)) (h1 : ∀ t ∈ 𝒯, Gen X.Λ t) (h2 : ∀ t, Gen X.Λ t → ∃! t', t' ∈ 𝒯 ∧ ∀ j, t j - t' j ∈ X.Λ) (n : ℕ) (hn : n = r) : (∑ᶠ (a : Fin n → Vsp A.T.dim) (_ : a ∈ {a : Fin n → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin n → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s) = ∑ t ∈ 𝒯, ∏ b, A.T.ptN2 1 A.s (fun _ => A.φ (t b)) / A.T.l2 1 A.s

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeviLogSumSqFibreZeroAndNonposAtMax

theorem **contDiffAt_two_of_eventually** {n : ℕ} {f : V n → ℂ} {z0 : V n} (h : ∀ᶠ z in 𝒩 z0, DifferentiableAt ℂ f z) : ContDiffAt ℂ 2 f z0

theorem **sum_wd_mul** {n : ℕ} {f : V n → ℂ} {z0 : V n} (h : DifferentiableAt ℂ f z0) (ξ : Fin n → ℂ) : ∑ j, wd n j f z0 * ξ j = fderiv ℂ f z0 (WithLp.toLp 2 ξ)

theorem **fderiv_eq_of_quot** {n : ℕ} {f g : V n → ℂ} {z0 v : V n} (hf : DifferentiableAt ℂ f z0) (hg : DifferentiableAt ℂ g z0) (h0 : g z0 ≠ 0) (h : fderiv ℂ (fun z => f z / g z) z0 v = 0) : fderiv ℂ f z0 v = (fderiv ℂ g z0 v / g z0) * f z0

theorem **levi_log_sum_eq_zero** {n m : ℕ} (z0 : V n) (ξ : Fin n → ℂ) (s : Fin m → V n → ℂ) (k0 : Fin m) (hs : ∀ k, ∀ᶠ z in 𝒩 z0, DifferentiableAt ℂ (s k) z) (h0 : s k0 z0 ≠ 0) (hq : ∀ k, fderiv ℂ (fun z => s k z / s k0 z) z0 (WithLp.toLp 2 ξ) = 0) : leviV n (fun z => ((Real.log (∑ k, ‖s k z‖ ^ 2) : ℝ) : ℂ)) z0 ξ = 0

theorem **main** : ∀ (n : ℕ) (z0 : V n) (ξ : Fin n → ℂ), (∀ (m : ℕ) (s : Fin m → V n → ℂ) (k0 : Fin m), (∀ k, ∀ᶠ z in 𝒩 z0, DifferentiableAt ℂ (s k) z) → s k0 z0 ≠ 0 → (∀ k, fderiv ℂ (fun z => s k z / s k0 z) z0 (WithLp.toLp 2 ξ) = 0) → (leviV n (fun z => ((Real.log (∑ k, ‖s k z‖ ^ 2) : ℝ) : ℂ)) z0 ξ).re = 0) ∧ ∀ g : V n → ℝ, ContDiffAt ℝ 2 g z0 → IsLocalMax g z0 → (leviV n (fun z => ((g z : ℝ) : ℂ)) z0 ξ).re ≤ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeviLogSumSqFibreZeroAndNonposAtMaxStatement

theorem **_root_.GlobalTheory.T.levi_log_sum_sq_fibre_zero_and_nonpos_at_max** : ∀ (n : ℕ) (z0 : V n) (ξ : Fin n → ℂ), (∀ (m : ℕ) (s : Fin m → V n → ℂ) (k0 : Fin m), (∀ k, ∀ᶠ z in 𝒩 z0, DifferentiableAt ℂ (s k) z) → s k0 z0 ≠ 0 → (∀ k, fderiv ℂ (fun z => s k z / s k0 z) z0 (WithLp.toLp 2 ξ) = 0) → (leviV n (fun z => ((Real.log (∑ k, ‖s k z‖ ^ 2) : ℝ) : ℂ)) z0 ξ).re = 0) ∧ ∀ g : V n → ℝ, ContDiffAt ℝ 2 g z0 → IsLocalMax g z0 → (leviV n (fun z => ((g z : ℝ) : ℂ)) z0 ξ).re ≤ 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeviLogThetaSum

theorem **levi_lower** {n : ℕ} {ι : Type} (s : Finset ι) {g : ι → V n → ℂ} {D : V n → ℂ} {φ : V n → ℝ} {z : V n} (c d : ℝ) (hg : ∀ t, ContDiffAt ℂ 2 (g t) z) (hD : ContDiffAt ℂ 2 D z) (hD0 : D z ≠ 0) (hφ : ContDiffAt ℝ 2 φ z) (hpos : 0 < ∑ t ∈ s, Complex.normSq (g t

z)) (ξ : Fin n → ℂ) : -(c * (leviV n (fun w => ((φ w : ℝ) : ℂ)) z ξ).re) ≤ (leviV n (fun w => ((Real.log (∑ t ∈ s, Complex.normSq (g t w)) + ((c * Real.log (Complex.normSq (D w)) + d) + (-c) * φ w) : ℝ) : ℂ)) z ξ).re

theorem **levi_log_theta_sum** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ ξ : Fin r → E, X.toData.IsIso ξ) → (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) → ∀ ps : Frame X.Ψ r → ℝ, (∀ x ∈ X.dom, ∃ A : TorusAt X.toData x, ps x = Real.log (∑ᶠ (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s)) → ∀ x0 ∈ X.dom, ∀ z, cpt X.toData x0 z ∈ X.dom → ∀ ξ : Fin (nd X.toData) → ℂ, -((r : ℝ) / 2 * leviM X.toData (phla X.toData (fun _ j => if X.p0 ≤ (j : ℕ) then 1 else -1)) x0 z ξ) ≤ leviM X.toD

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LeviLogThetaSumStatement

theorem **_root_.GlobalTheory.T.levi_log_theta_sum_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ ξ : Fin r → E, X.toData.IsIso ξ) → (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) → ∀ ps : Frame X.Ψ r → ℝ, (∀ x ∈ X.dom, ∃ A : TorusAt X.toData x, ps x = Real.log (∑ᶠ (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s)) → ∀ x0 ∈ X.dom, ∀ z, cpt X.toData x0 z ∈ X.dom → ∀ ξ : Fin (nd X.toData) → ℂ, -((r : ℝ) / 2 * leviM X.toData (phla X.toData (fun _ j => if X.p0 ≤ (j : ℕ) then 1 else

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LinearIndependentRingHomCompIff

theorem **linearIndependent_ringHom_comp_iff** {A F : Type} [CommRing A] [Field F] (φ : A →+* F) (hφ : Function.Injective φ) {ι μ : Type} [Finite μ] (c : ι → μ → A) : LinearIndependent F (fun i m => φ (c i m)) ↔ LinearIndependent A c

theorem **finrank_span_ringHom_comp_le** {A F F' : Type} [CommRing A] [Field F] [Field F'] (φ : A →+* F) (ψ : A →+* F') (hφ : Function.Injective φ) (hψ : Function.Injective ψ) {ι μ : Type} [Finite ι] [Finite μ] (c : ι → μ → A) : Module.finrank F (Submodule.span F (Set.range fun i m => φ (c i m))) ≤ Module.finrank F' (Submodule.span F' (Set.range fun i m => ψ (c i m)))

theorem **finrank_span_ringHom_comp_eq** {A F F' : Type} [CommRing A] [Field F] [Field F'] (φ : A →+* F) (ψ : A →+* F') (hφ : Function.Injective φ) (hψ : Function.Injective ψ) {ι μ : Type} [Finite ι] [Finite μ] (c : ι → μ → A) : Module.finrank F (Submodule.span F (Set.range fun i m => φ (c i m))) = Module.finrank F' (Submodule.span F' (Set.range fun i m => ψ (c i m)))

theorem **mem_span_iff_finrank_option_eq** {F V ι : Type} [Field F] [AddCommGroup V] [Module F V] [FiniteDimensional F V] (v : Option ι → V) : v none ∈ Submodule.span F (Set.range fun i => v (some i)) ↔ Module.finrank F (Submodule.span F (Set.range fun i => v (some i))) = Module.finrank F (Submodule.span F (Set.range v))

theorem **rank_indep_mem_embedding_invariant** : ∀ (A F F' : Type) [CommRing A] [Field F] [Field F'] (φ : A →+* F) (ψ : A →+* F'), Function.Injective φ → Function.Injective ψ → ∀ (ι μ : Type) [Fintype ι] [Fintype μ] (c : ι → μ → A), (LinearIndependent F (fun i m => φ (c i m)) ↔ LinearIndependent F' (fun i m => ψ (c i m))) ∧ (∀ q : μ → A, (fun m => φ (q m)) ∈ Submodule.span F (Set.range fun i m => φ (c i m)) ↔ (fun m => ψ (q m)) ∈ Submodule.span F' (Set.range fun i m => ψ (c i m))) ∧ Module.finrank F (Submodule.span F (Set.range fun i m => φ (c i m))) = Module.finrank F' (Submodule.span F' (Set.range fun i m => ψ (c i m)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogCardCompUnfLe

theorem **log_card_comp_unf_le** : GlobalTheory.Published.HasseNormSt → ThetaUnifAV → Out.Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (D : Datum H) (l : Level H) (Xo : Moduli D l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), 1 ≤ r → Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → ∀ (ξ : E) (AA : Submodule (𝒪 E) E), FD.cc ξ = -ξ → ξ ≠ 0 → AA.FG → AA ≠ ⊥ → (∀ x : E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ E (ξ * x * a * FD.cc b) = z) ↔ IsIntegral ℤ x) → ∀ L : ℝ, 16 ≤ FD.fdeg E → 2 ≤ L → Real.log (FD.dE E) ≤ 4 * FD.fdeg E * L → Real.log r ≤ L → Real.log (FD.fdeg E) ≤ 2 * L → Real.log l.M' ≤ 30 * L → Real.log (Fintype.card U.Comp) ≤ 41 * (FD.fdeg E * r * L) ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogCardComponentsLeAtGoodEmbedding

theorem **log_card_comp_le** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → 16 ≤ D.R.f → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → Real.log (Fintype.card (T.U j σ).Comp) ≤ 41 * Zb D large₂ ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogCardComponentsLeAtGoodEmbeddingStatement

theorem **_root_.GlobalTheory.T.log_card_components_le_at_good_embedding** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → 16 ≤ D.R.f → ∀ σ : T.Q.c.F →+* ℂ, GoodEmb T.Q.c σ → Real.log (Fintype.card (T.U j σ).Comp) ≤ 41 * Zb D large₂ ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogExpMulLeOfGeneratorSupBound

private theorem **log_one_add_le_log_two_add_of_log_le** (x M : ℝ) (hx : 0 < x) (hM : 0 ≤ M) (h : Real.log x ≤ M) : Real.log (1 + x) ≤ Real.log 2 + M

theorem **log_exp_mul_le_of_generator_sup_bound** : ∀ (f r n d : ℕ) (L ℓ a rd lam t c S v p1 : ℝ), 2 ≤ L → 1 ≤ f → 1 ≤ r → r ≤ n → (d : ℝ) ≤ a → Real.log (1 + a) ≤ 2 * L → Real.log n ≤ L → Real.log f ≤ 2 * L → 1 ≤ rd → Real.log rd ≤ 2 * L → 0 < lam → Real.log (1 / lam ^ 2) ≤ 80 * L → 0 < t → 0 ≤ ℓ → 0 ≤ p1 → p1 ≤ Real.exp ℓ → 0 ≤ c → c ≤ Real.exp (f * r * n * L) → 0 ≤ S → S ≤ 2 * v → v ≤ c * (p1 * (Real.pi * Real.exp 1 * t * d) ^ ((d : ℝ) / 2) * Real.sqrt 2 * (1 + 4 * Real.sqrt (2 * (f * r : ℕ) * n) * (2 * (f : ℝ) ^ ((3 : ℝ) / 2) * n ^ 2 * rd) / (Real.sqrt Real.pi * lam)) ^ (f * r * n)) → Real.log (Real.exp (-((d : ℝ) / 2) * Real.log (Real.pi * t)) * S) ≤ ℓ + 10 * L * a + 200 * f * r * n * L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogLeOfLePowMulPow

theorem **log_le_of_le_pow_mul_pow** {N g₀ g₁ d X m : ℕ} {a c P : ℝ} (ha : 0 ≤ a) (hc : 0 ≤ c) (hP : 0 ≤ P) (h₀ : Real.log g₀ ≤ a) (h₁ : Real.log g₁ ≤ a) (hd : Real.log d ≤ P) (hX : Real.log X ≤ c) (hN : N ≤ (g₀ * g₁) ^ m * (d * X ^ m)) : Real.log N ≤ m * (2 * a + c) + P

theorem **frn_le_size_sub_one** {D : D7} {j : ℕ} (w : WeightFamilies.Wt D j) : ((D.R.f * D.R.r * w.n : ℕ) : ℝ) ≤ w.size - 1

theorem **log_count_le_mul_size** : ∀ (D : D7) (B : WeightFamilies.Prov D) (large₂ : SizeData₂ D) (j : ℕ) (w : WeightFamilies.Wt D j) (N : ℕ) (Prw : ℝ), 2 ≤ large₂.sc.L → 0 < w.n → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → OPO large₂ B → 38 * large₂.sc.L + PO D large₂ ≤ Prw → N ≤ (D.host.g 0 * D.host.g 1) ^ (D.R.f * D.R.r * w.n) * (Module.finrank ℚ B.KO * (1 + WeightAlgebra.DF w) ^ (D.R.f * D.R.r * w.n)) → Real.log N ≤ Prw * (w.size - 1)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogNormMapPowMulPowOfMulConj

theorem **norm_map_eq_sqrt_of_mul_conj** {K : Type*} [Semiring K] (σ : K →+* ℂ) (x y : K) (g : ℕ) (hxy : x * y = g) (hy : σ y = starRingEnd ℂ (σ x)) : ‖σ x‖ = (g : ℝ) ^ (1 / 2 : ℝ)

theorem **norm_map_pow_mul_pow_of_mul_conj** {K : Type*} [Semiring K] (σ : K →+* ℂ) (x y : K) (g : ℕ) (hxy : x * y = g) (hy : σ y = starRingEnd ℂ (σ x)) (a b : ℕ) : ‖σ (x ^ a * y ^ b)‖ = (g : ℝ) ^ (((a : ℝ) + b) / 2)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogNormNormaliserEqHalfDegreeLog

theorem **normaliser_factor_mul_bar** {D : D7} (EX : ErrX D) (i : Fin 2) : EX.piG i * piBar0 EX i = ((D.host.g i : ℕ) : 𝒪 D.host.E)

theorem **map_piBar_eq_conj** {D : D7} (EX : ErrX D) (i : Fin 2) (ρ : 𝒪 D.host.E →+* ℂ) (hρ : Function.Injective ρ) : ρ (piBar0 EX i) = starRingEnd ℂ (ρ (EX.piG i))

theorem **log_norm_normaliser** : ∀ (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (w : WeightFamilies.Wt D j) (σ : B.KO →+* ℂ), nnW B hinj EX w ≠ 0 ∧ Real.log ‖σ (nnW B hinj EX w : B.KO)‖ = (WeightAlgebra.DF w : ℝ) / 2 * Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogNormNormaliserEqHalfDegreeLogStatement

theorem **_root_.GlobalTheory.T.log_norm_normaliser_eq_half_degree_log** : ∀ (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (w : WeightFamilies.Wt D j) (σ : B.KO →+* ℂ), nnW B hinj EX w ≠ 0 ∧ Real.log ‖σ (nnW B hinj EX w : B.KO)‖ = (WeightAlgebra.DF w : ℝ) / 2 * Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogShortFamilyNumbersLe

theorem **log_shortFamilyNumbers_le** : ∀ (N n a f : ℕ) (rd L : ℝ), 1 ≤ N → N ≤ n → 1 ≤ a → 1 ≤ f → 1 ≤ rd → 2 ≤ L → Real.log rd ≤ 2 * L → Real.log f ≤ 2 * L → Real.log (1 + a) ≤ 2 * L → Real.log n ≤ L → Real.log (((N : ℝ) * Real.sqrt n * ((N + 1).factorial * ((2 * f * n) * ((2 * N + 1) * a) * (2 * (f : ℝ) ^ ((3 : ℝ) / 2) * n ^ 2 * rd)) ^ (2 * N + 1))) ^ a) ≤ 140 * N * L * a

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogSupNormShortThetaGeneratorLe

theorem **log_sup_short_generator_le** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → LastPos (T.Q.Dw j) (wtA w) → LawsSh T → (T.dδ j w.n).l₂.A = 1 → ∀ (Λ : GenLat D.host.E w.n) (N : ℕ) (x0 : Fin w.n → (T.Q.Dw j).La) (P : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F) (g : T.R j), Λ.Unimod → Λ.TrivAt (badN D) → Λ.TrivAt (T.dδ j w.n).l₂.M → Λ.LevOK N (T.dδ j w.n).l₂.M → AdmX (l

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogSupNormShortThetaGeneratorLeStatement

theorem **_root_.GlobalTheory.T.log_supNorm_shortThetaGenerator_le** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → LastPos (T.Q.Dw j) (wtA w) → LawsSh T → (T.dδ j w.n).l₂.A = 1 → ∀ (Λ : GenLat D.host.E w.n) (N : ℕ) (x0 : Fin w.n → (T.Q.Dw j).La) (P : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F) (g : T.R j), Λ.Unimod → Λ.TrivAt (badN D) → Λ.TrivAt (T.dδ j w.n).l₂.M → Λ.LevOK N (T.dδ j w.n).l₂.M → AdmX (l

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogSupNormThetaGeneratorLe

theorem **log_sup_theta_generator_le** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → LastPos (T.Q.Dw j) (wtA w) → ∀ (x : Fin w.n → (T.Q.Dw j).La) (P : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F), P ∈ harmA T.Q.c (wtA w) w.n → ∀ f : T.R j, IsThRule (T.Q.Xo j) T.Q.c (T.nab j) x P f → ∀ (σ : T.Q.c.F →+* ℂ) (ℓ : ℝ), 0 ≤ ℓ → l1At σ P ≤ Real.exp ℓ → Real.log (T.Ns j w σ f) ≤ ℓ + 10 * large₂.sc.L * absA w + 200 * D.R.f * D.R.r * w.n * large₂.sc.L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogSupNormThetaGeneratorLeStatement

theorem **_root_.GlobalTheory.T.log_supNorm_thetaGenerator_le** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → LastPos (T.Q.Dw j) (wtA w) → ∀ (x : Fin w.n → (T.Q.Dw j).La) (P : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F), P ∈ harmA T.Q.c (wtA w) w.n → ∀ f : T.R j, IsThRule (T.Q.Xo j) T.Q.c (T.nab j) x P f → ∀ (σ : T.Q.c.F →+* ℂ) (ℓ : ℝ), 0 ≤ ℓ → l1At σ P ≤ Real.exp ℓ → Real.log (T.Ns j w σ f) ≤ ℓ + 10 * large₂.sc.L * absA w + 200 * D.R.f * D.R.r * w.n * large₂.sc.L

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogThetaSumInvariantSmooth

theorem **gen_map** {Λ : Submodule (𝒪 E) (Fin r → E)} (e : (Fin r → E) ≃ₗ[E] (Fin r → E)) (he : ∀ v, v ∈ Λ ↔ e v ∈ Λ) {t : Fin r → Fin r → E} (h : Gen Λ t) : Gen Λ fun j => e (t j)

theorem **exists_reps_map** {Λ : Submodule (𝒪 E) (Fin r → E)} (e : (Fin r → E) ≃ₗ[E] (Fin r → E)) (he : ∀ v, v ∈ Λ ↔ e v ∈ Λ) (𝒯 : Finset (Fin r → Fin r → E)) (h1 : ∀ t ∈ 𝒯, Gen Λ t) (h2 : ∀ t, Gen Λ t → ∃! t', t' ∈ 𝒯 ∧ ∀ j, t j - t' j ∈ Λ) : ∃ 𝒯' : Finset (Fin r → Fin r → E), (∀ t ∈ 𝒯', Gen Λ t) ∧ (∀ t, Gen Λ t → ∃! t', t' ∈ 𝒯' ∧ ∀ j, t j - t' j ∈ Λ) ∧ ∀ f : (Fin r → Fin r → E) → ℝ, ∑ t ∈ 𝒯', f t = ∑ t ∈ 𝒯, f fun j => e (t j)

theorem **contDiffOn_cpt** (X : Data E r) (x0 : Frame X.Ψ r) {F : Frame X.Ψ r → ℂ} (hF : DifferentiableOn ℂ F X.dom) : ContDiffOn ℝ ∞ (fun z => F (cpt X x0 z)) (OmF X x0)

theorem **contDiffOn_nfac** (X : Data E r) (x0 : Frame X.Ψ r) (la : Wt X.Ψ r) : ContDiffOn ℝ ∞ (fun z => X.nfac la (cpt X x0 z)) (OmF X x0)

theorem **log_theta_sum_invariant_smooth** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ ξ : Fin r → E, X.toData.IsIso ξ) → (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) → ∀ ps : Frame X.Ψ r → ℝ, (∀ x ∈ X.dom, ∃ A : TorusAt X.toData x, ps x = Real.log (∑ᶠ (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s)) → IsInvFn X.toData ps ∧ ∀ x0 ∈ X.dom, ContDiffOn ℝ ∞ (fun z => ps (cpt X.toData x0 z)) {z | cpt X.toData x0 z ∈ X.dom}

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogThetaSumInvariantSmoothStatement

theorem **_root_.GlobalTheory.T.log_theta_sum_invariant_smooth_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ ξ : Fin r → E, X.toData.IsIso ξ) → (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 ∗ herm X.Gm u (((w0 : E) − cc (w0 : E)) • v)) − Algebra.trace ℚ E (X.ξ0 ∗ herm X.Gm u v) = 2 ∗ k) → ∀ ps : Frame X.Ψ r → ℝ, (∀ x ∈ X.dom, ∃ A : TorusAt X.toData x, ps x = Real.log (∑ᶠ (a : Fin A.T.r → Vsp A.T.dim) (_ : a ∈ {a : Fin A.T.r → Vsp A.T.dim | (∀ j, a j ∈ ZSpan.fundamentalDomain A.T.b) ∧ (∀ j, ((5 : ℂ) ^ 2) • a j ∈ A.T.L) ∧ ∀ x ∈ A.T.L, ∃ c : Fin A.T.r → 𝒪 E, ∃ y ∈ A.T.L, x = ∑ j, act (c j) (((5 : ℂ) ^ 2) • a j) + (5 : ℂ) • y}), ∏ b, A.T.ptN2 1 A.s (fun _ => a b) / A.T.l2 1 A.s)) → IsInvFn X.toData ps ∧ ∀ x0 ∈ X.dom, ContDiffOn ℝ ∞ (fun z => ps (cpt X.toData x0 z)) {z | cpt X.toData x0 z ∈ X.dom}

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LogTwoMulCountLeMulSize

theorem **log_two_mul_count_le_mul_size** : ∀ (D : D7) (B : WeightFamilies.Prov D) (large₂ : SizeData₂ D) (j : ℕ) (w : WeightFamilies.Wt D j) (N : ℕ) (Prw : ℝ), 2 ≤ large₂.sc.L → 0 < w.n → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → OPO large₂ B → 38 ∗ large₂.sc.L + PO D large₂ ≤ Prw → N ≤ 2 ∗ ((D.host.g 0 ∗ D.host.g 1) ^ (D.R.f ∗ D.R.r ∗ w.n) ∗ (Module.finrank ℚ B.KO ∗ (1 + WeightAlgebra.DF w) ^ (D.R.f ∗ D.R.r ∗ w.n))) → Real.log N ≤ Prw ∗ (w.size − 1)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LowerBoundAtFramesFamilyWithScaledThetaRules

theorem **totalDegree_eq_DF** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E} {Dw : Datum H} {l : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H} (c : Coeff l) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ harmA c a n) (h0 : P ≠ 0) : P.totalDegree = Formalism.L2.Pj.DF Dw a

theorem **pow_le_sq_of_log** {M x : ℝ} {d : ℕ} (hM : 1 < M) (hd : 1 ≤ d) (hx : 0 ≤ x) (h : (d : ℝ) / 2 ∗ Real.log M ≤ Real.log x) : M ^ d ≤ x ^ 2

theorem **i2iAt_record** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T) (_EX : ErrX D) (j : ℕ) (hj : InR D j) (K : KAdm) (Ch Ah Cq Aq : ℝ) (S : Scal (recB h.toLawsCore)) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → TieC K H Ch Ah Cq Aq C2 A2 CR → PageAt large₂ C2 A2 → EtaleWeightGaps.PageSupp D large₂ (recB h.toLawsCore) C2 A2 nW S wL → ∀ (P₁ : MvPolynomial (LtN (T.Q.Dw j) wL.n) T.Q.c.F), P₁ ∈ harmA T.Q.c (wtA wL) wL.n → IsBlockProd P₁ → ∀ (N : ℕ) (w : Fin (N + 1) → (recB h.toLawsCore).Cusp j wL), ContainsTh (T.Q.Xo j) T.Q.c (T.nab j) ((S.sW : (recB h.toLawsCore).KO) : T.Q.c.F) P₁ (fun k => ((w k).1 : T.R j)) → ∀ tableLineDegreeFive : ℝ, p14 D.host.E D.R.r wL.n D.R.m ≤ tableLineDegreeFive →

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.LowerBoundAtFramesFamilyWithScaledThetaRulesStatement

theorem **_root_.GlobalTheory.T.lower_bound_at_frames_family_with_scaled_thetaRules** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T) (_EX : ErrX D) (j : ℕ) (hj : InR D j) (K : KAdm) (Ch Ah Cq Aq : ℝ) (S : Scal (recB h.toLawsCore)) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → TieC K H Ch Ah Cq Aq C2 A2 CR → PageAt large₂ C2 A2 → EtaleWeightGaps.PageSupp D large₂ (recB h.toLawsCore) C2 A2 nW S wL → ∀ (P₁ : MvPolynomial (LtN (T.Q.Dw j) wL.n) T.Q.c.F), P₁ ∈ harmA T.Q.c (wtA wL) wL.n → IsBlockProd P₁ → ∀ (N : ℕ) (w : Fin (N + 1) → (recB h.toLawsCore).Cusp j wL), ContainsTh (T.Q.Xo j) T.Q.c (T.nab j) ((S.sW : (recB h.toLawsCore).KO) : T.Q.c.F) P₁ (fun k => ((w k).1 : T.R j)) → ∀ tableLineDegreeFi

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MarginsOfGaps

theorem **margins_of_gaps** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X), 0 < X.p0 → X.p0 < r → Nonempty X.Ψ → Xa.de = de2W X → IsStrictDom X Xa.L → (∀ τ (a b : Fin r), a ≤ b → ((b : ℕ) < X.p0 ∨ X.p0 ≤ (a : ℕ)) → (Xa.Fp − kapW X − nd X • Xa.L) τ b ≤ (Xa.Fp − kapW X − nd X • Xa.L) τ a) → ∀ bb m mL : ℝ, 4 ∗ (X.Ψ.card : ℝ) ∗ (r : ℝ) ^ 2 < bb → (∀ τ (a b : Fin r), (a : ℕ) < X.p0 → X.p0 ≤ (b : ℕ) → m ≤ (((Xa.Fp − kapW X − nd X • Xa.L) τ b : ℤ) : ℝ) − (((Xa.Fp − kapW X − nd X • Xa.L) τ a : ℤ) : ℝ) ∧ mL ≤ ((Xa.L τ b : ℤ) : ℝ) − ((Xa.L τ a : ℤ) : ℝ)) → bb + 2 ∗ Xa.T < m → bb ^ 2 ∗ ((X.Ψ.card ∗ min X.p0 (r − X.p0) : ℕ) : ℝ) + bb < m + nd X ∗ mL → 2 ∗ (r : ℝ) < m → Margins X Xa

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MatRmulUnipMulHeadProj

theorem **coper_of_lt** (y : Frame X.Ψ r) (v : Fin r → E) (τ : X.Ψ) {a : Fin r} (h : (a : ℕ) < X.p0) : coper X y v (τ, a) = X.c τ ∗ ((mat y τ)⁻¹ ∗ᵥ Data.emb τ.1 v) a

theorem **coper_of_le** (y : Frame X.Ψ r) (v : Fin r → E) (τ : X.Ψ) {a : Fin r} (h : X.p0 ≤ (a : ℕ)) : coper X y v (τ, a) = −(X.c τ ∗ hermC τ.1 X.Gm (FD.col (mat y τ) a) (Data.emb τ.1 v))

theorem **coper_add** (y : Frame X.Ψ r) (u v : Fin r → E) : coper X y (u + v) = coper X y u + coper X y v

theorem **coper_smul** (y : Frame X.Ψ r) (α : E) (v : Fin r → E) (p : X.Ψ × Fin r) : coper X y (α • v) p = conj (rowType X p α) ∗ coper X y v p

theorem **hermC_eq_sum_col** (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) {A : Matrix (Fin r) (Fin r) ℂ} (hA : A.det ≠ 0) (u v : Fin r → ℂ) : hermC τ G u v = ∑ a, (A⁻¹ ∗ᵥ u) a ∗ hermC τ G (FD.col A a) v

theorem **EF_eq_coper** {y : Frame X.Ψ r} (hy : ∀ τ, (mat y τ).det ≠ 0) (u v : Fin r → E) : ((EF X u v : ℚ) : ℂ) = I / 2 ∗ (coper X y u ⬝ᵥ per X y v − per X y u ⬝ᵥ coper X y v)

theorem **cxMat_mulVec_int** (f : (Fin r → E) →+ (X.Ψ × Fin r → ℂ)) (m : Fin g ⊕ Fin g → ℤ) : cxMat X bas idx f ∗ᵥ (fun s => (m s : ℂ)) = f (bas m) ∘ idx

theorem **actMat_mulVec_star** (α : 𝒪 E) (u : Fin g ⊕ Fin g → ℂ) : actMat X bas α ∗ᵥ star u = star (actMat X bas α ∗ᵥ u)

theorem **cxMat_mul_actMat** (f : (Fin r → E) →+ (X.Ψ × Fin r → ℂ)) (α : 𝒪 E) (χ : X.Ψ × Fin r → ℂ) (hf : ∀ v p, f ((α : E) • v) p = χ p ∗ f v p) : cxMat X bas idx f ∗ actMat X bas α = diagonal (fun i => χ (idx i)) ∗ cxMat X bas idx f

theorem **perCx_int** (y : Frame X.Ψ r) (m : Fin g ⊕ Fin g → ℤ) : perCx X bas idx y (fun s => (m s : ℂ)) = per X y (bas m) ∘ idx

theorem **coperCx_int** (y : Frame X.Ψ r) (m : Fin g ⊕ Fin g → ℤ) : coperCx X bas idx y (fun s => (m s : ℂ)) = coper X y (bas m) ∘ idx

theorem **perCx_apply** (y : Frame X.Ψ r) (u : Fin g ⊕ Fin g → ℂ) : perCx X bas idx y u = perMat X bas idx Sum.inl y ∗ᵥ (u ∘ Sum.inl) + perMat X bas idx Sum.inr y ∗ᵥ (u ∘ Sum.inr)

theorem **coperCx_apply** (y : Frame X.Ψ r) (u : Fin g ⊕ Fin g → ℂ) : coperCx X bas idx y u = coMat X bas idx Sum.inl y ∗ᵥ (u ∘ Sum.inl) + coMat X bas idx Sum.inr y ∗ᵥ (u ∘ Sum.inr)

theorem **perCx_actMat** (y : Frame X.Ψ r) (α : 𝒪 E) (u : Fin g ⊕ Fin g → ℂ) : perCx X bas idx y (actMat X bas α ∗ᵥ u) = diagonal (fun i => rowType X (idx i) α) ∗ᵥ perCx X bas idx y u

theorem **coperCx_actMat** (y : Frame X.Ψ r) (α : 𝒪 E) (u : Fin g ⊕ Fin g → ℂ) : coperCx X bas idx y (actMat X bas α *ᵥ u) = diagonal (fun i => conj (rowType X (idx i) α)) *ᵥ coperCx X bas idx y u

theorem **coperCx_secLin** (y : Frame X.Ψ r) (w : Fin g → ℂ) : coperCx X bas idx y (secLin X bas idx y w) = splitMat X bas idx y *ᵥ w

theorem **omegaCx_eq_coper** {y : Frame X.Ψ r} (hy : ∀ τ, (mat y τ).det ≠ 0) (u v : Fin g ⊕ Fin g → ℂ) : omegaCx g u v = I / 2 * (perCx X bas idx y u ⬝ᵥ coperCx X bas idx y v − coperCx X bas idx y u ⬝ᵥ perCx X bas idx y v)

theorem **eq_and_iff_of_rowType** {p q : X.Ψ × Fin r} (h : ∀ α : 𝒪 E, rowType X p α = conj (rowType X q α)) : p.1 = q.1 ∧ ((p.2 : ℕ) < X.p0 ↔ X.p0 ≤ (q.2 : ℕ))

theorem **sum_letterDir_apply** {n : ℕ} (z : Harm.Var X.Ψ q0 X.p0 (Fin n) → ℂ) (t : X.Ψ) (ρ : Fin q0 ⊕ Fin X.p0) (k : Fin n) : (∑ l, z l • letterDir X hr idx l) k (idx.symm (rowOf X hr (t, ρ))) = z (t, ρ, k)

theorem **IsBalanced.dot_mulVec** {M : Matrix (Fin g) (Fin g) ℂ} (hM : IsBalanced X idx M) (w : Fin g → ℂ) : w ⬝ᵥ M *ᵥ w = ∑ t, ∑ a, ∑ b, (M (idx.symm (rowOf X hr (t, .inl a))) (idx.symm (rowOf X hr (t, .inr b))) + M (idx.symm (rowOf X hr (t, .inr b))) (idx.symm (rowOf X hr (t, .inl a)))) * w (idx.symm (rowOf X hr (t, .inl a))) * w (idx.symm (rowOf X hr (t, .inr b)))

theorem **IsBalanced.polyDeriv_exp_mul** {M : Matrix (Fin g) (Fin g) ℂ} (hM : IsBalanced X idx M) {n : ℕ} {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : Harm.IsHarmonic P) (c : ℂ) {G : (Fin n → Fin g → ℂ) → ℂ} (hG : AnalyticAt ℂ G 0) : polyDeriv P (letterDir X hr idx) (fun z => cexp (∑ k, c * (z k ⬝ᵥ M *ᵥ z k)) * G z) 0 = polyDeriv P (letterDir X hr idx) G 0

theorem **hMat_mulVec_perCx** (u : Fin g ⊕ Fin g → ℂ) : hMat X bas idx x *ᵥ perCx X bas idx x u = perProj (tauMat X bas idx x) u

theorem **perCx_eq_perProj** (u : Fin g ⊕ Fin g → ℂ) : perCx X bas idx x u = perMat X bas idx Sum.inl x *ᵥ perProj (tauMat X bas idx x) u

theorem **secLin_isLagSection** : IsLagSection (omegaCx g) (perCx X bas idx x) (secLin X bas idx x)

theorem **splitMat_isSymm** : (splitMat X bas idx x).IsSymm

theorem **splitMat_mulVec_per** (m : Fin g → ℤ) : splitMat X bas idx x *ᵥ (per X x (bas (Sum.elim m 0)) ∘ idx) = coper X x (bas (Sum.elim m 0)) ∘ idx

theorem **dot_imInv_sub_splitMat** (z w : Fin g → ℂ) : z ⬝ᵥ ((hMat X bas idx x)ᵀ * imInv X bas idx x * hMat X bas idx x − splitMat X bas idx x) *ᵥ w = −(coperCx X bas idx x (secH (tauMat X bas idx x) (hMat X bas idx x *ᵥ z)) ⬝ᵥ w)

theorem **isBalanced_sub_splitMat** : IsBalanced X idx ((hMat X bas idx x)ᵀ * imInv X bas idx x * hMat X bas idx x − splitMat X bas idx x)

theorem **thFr_eq_splitMat** {q0 : ℕ} (hr : X.p0 + q0 = r) (n : ℕ) (a b : Fin n → Fin g → ℝ) {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : Harm.IsHarmonic P) : thFr X hr bas idx n a b P x = (hMat X bas idx x).det ^ (n / 2) * polyDeriv P (letterDir X hr idx) (fun z => ∏ k, cexp (Real.pi / 2 * (z k ⬝ᵥ splitMat X bas idx x *ᵥ z k)) * thetaChar (tauMat X bas idx x) (fun i => (a k i : ℂ)) (fun i => (b k i : ℂ)) (hMat X bas idx x *ᵥ z k)) 0

theorem **differentiableOn_coper** (v : Fin r → E) (p : X.Ψ × Fin r) : DifferentiableOn ℂ (fun y => coper X y v p) {y | ∀ τ, (mat y τ).det ≠ 0}

theorem **differentiableOn_splitMat** (i j : Fin g) : DifferentiableOn ℂ (fun x => splitMat X bas idx x i j) X.dom

theorem **differentiableOn_thFr** {q0 : ℕ} (hr : X.p0 + q0 = r) (n : ℕ) (a b : Fin n → Fin g → ℝ) {P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)} (hP : Harm.IsHarmonic P) : DifferentiableOn ℂ (thFr X hr bas idx n a b P) X.dom

theorem **headProj_mul_self** : headProj X * headProj X = headProj X

theorem **mat_unip** (K : Frame X.Ψ r) (τ : X.Ψ) : mat (unip X K) τ = 1 + headProj X * mat K τ * (1 − headProj X)

@[simp] theorem **mix_self** (x : Frame X.Ψ r) : mix X x x = x

theorem **mat_unip_neg_mul** (K : Frame X.Ψ r) (τ : X.Ψ) : mat (unip X (−K)) τ * mat (unip X K) τ = 1

theorem **unip_mem_Bor** (K : Frame X.Ψ r) : unip X K ∈ Bor X.Ψ r

theorem **per_rmul_unip** (x K : Frame X.Ψ r) : per X (rmul x (unip X K)) = per X x

theorem **hMat_congr** : hMat X bas idx x' = hMat X bas idx x

theorem **tauMat_congr** : tauMat X bas idx x' = tauMat X bas idx x

theorem **thFr_congr** {q0 : ℕ} (hr : X.p0 + q0 = r) (n : ℕ) (a b : Fin n → Fin g → ℝ) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) : thFr X hr bas idx n a b P x' = thFr X hr bas idx n a b P x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MemNabStdOfDetMulIntegral

theorem **mem_valuationSubring_of_indep** {K σ ι : Type} [Field K] [Fintype ι] (A : ValuationSubring K) (q : ι → MvPolynomial σ ℤ) (hq : LinearIndependent (IsLocalRing.ResidueField A) (fun i => map (Int.castRingHom (IsLocalRing.ResidueField A)) (q i))) (c : ι → K) (hc : ∀ m, (∑ i, c i • map (Int.castRingHom K) (q i)).coeff m ∈ A) (i : ι) : c i ∈ A

theorem **integral_of_indep** {K σ ι : Type} [Field K] [NumberField K] [Fintype ι] (q : ι → MvPolynomial σ ℤ) (hq : ∀ (k : Type) [Field k], LinearIndependent k (fun i => map (Int.castRingHom k) (q i))) (c : ι → K) (hc : ∀ m, IsIntegral ℤ ((∑ i, c i • map (Int.castRingHom K) (q i)).coeff m)) (i : ι) : ∃ a : 𝒪 K, algebraMap (𝒪 K) K a = c i

theorem **ssZ_indep** (k : Type) [Field k] (n : ℕ) (ν : Fin n → ℕ) : LinearIndependent k (fun m => map (Int.castRingHom k) (ssZ n ν m))

theorem **ssZ_span** (K : Type) [Field K] [CharZero K] (n : ℕ) (ν : Fin n → ℕ) : Submodule.span K (Set.range fun m => map (Int.castRingHom K) (ssZ n ν m)) = Formalism.GradedWeylDivisibility.dualWeyl K n n ν

theorem **ssZ_mem** (n : ℕ) (ν : Fin n → ℕ) (m) : ssZ n ν m ∈ Formalism.GradedWeylDivisibility.dualWeyl ℤ n n ν

theorem **map_prodZ** (k : Type) [CommRing k] (lam : Ix → DomWeight n) (i : VVIdx lam) : map (Int.castRingHom k) (prodZ lam i) = ∏ t, Formalism.GradedWeylDivisibility.blockEmb k t (map (Int.castRingHom k) (ssZ n (lam t).natW (i t)))

theorem **prodZ_basis** (k : Type) [Field k] (lam : Ix → DomWeight n) : LinearIndependent k (fun i => map (Int.castRingHom k) (prodZ lam i)) ∧ Submodule.span k (Set.range fun i => map (Int.castRingHom k) (prodZ lam i)) = Formalism.GradedWeylDivisibility.tensorLat k k fun t => Submodule.span k (Set.range fun m => map (Int.castRingHom k) (ssZ n (lam t).natW m))

theorem **prodZ_mem** (K : Type) [Field K] [NumberField K] (lam : Ix → DomWeight n) (i : VVIdx lam) : map (Int.castRingHom K) (prodZ lam i) ∈ nabStd K lam

theorem **map_detProd** (k : Type) [CommRing k] (e : Ix → ℕ) : map (Int.castRingHom k) (∏ t, (gen ℤ n t).det ^ e t) = ∏ t, (gen k n t).det ^ e t

theorem **detProd_ne_zero** (k : Type) [Field k] (e : Ix → ℕ) : (∏ t, (gen k n t).det ^ e t) ≠ 0

theorem **mem_nabStd_of_det_mul_integral** : ∀ (K : Type) [Field K] [NumberField K] (Ix : Type) [Fintype Ix] (n : ℕ) (lam : Ix → DomWeight n) (e : Ix → ℕ), ∀ F ∈ VV K lam, (∀ m, IsIntegral ℤ (((∏ t, (gen K n t).det ^ e t) * F).coeff m)) → F ∈ nabStd K lam

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MemOfForallLocP

theorem **mem_colLat_one** {K : Type} [Field K] [NumberField K] {n : ℕ} (x : Fin n → K) : x ∈ colLat (1 : GL (Fin n) K) ↔ ∀ a, ∃ y : 𝒪 K, algebraMap (𝒪 K) K y = x a

theorem **tauLat_stdLat** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] {Ix : Type} {n : ℕ} (emb : Ix → E →+* K) (t : Ix) : tauLat emb t (stdLat E n) = colLat (1 : GL (Fin n) K)

theorem **dualFr_colLat_one** {K : Type} [Field K] [NumberField K] {n : ℕ} : dualFr (colLat (1 : GL (Fin n) K)) = colLat (1 : GL (Fin n) K)

theorem **mem_of_forall_locP** {K W : Type} [Field K] [NumberField K] [AddCommGroup W] [Module (𝒪 K) W] (T : Submodule (𝒪 K) W) (x : W) (h : ∀ w : HeightOneSpectrum (𝒪 K), x ∈ Formalism.HermitianLatticesTheory.localization₂ w.asIdeal T) : x ∈ T

theorem **polyAction_one₂** {K Ix : Type} [Field K] [Fintype Ix] {n : ℕ} (lam : Ix → DomWeight n) (F : Pol K Ix n) : actP lam 1 F = F

theorem **nabP_colLat_one** {K Ix : Type} [Field K] [NumberField K] [Fintype Ix] {n : ℕ} (lam : Ix → DomWeight n) : nabP lam (fun _ => colLat (1 : GL (Fin n) K)) = nabStd K lam

theorem **eval_extremal** {R : Type} [CommRing R] {n : ℕ} (lam : DomWeight n) : eval (fun v : Fin n × Fin n => if v.1 = v.2 then (1 : R) else 0) (extremal R n lam) = 1

theorem **nabAtr_stdLat** : ∀ (E K : Type) [Field E] [NumberField E] [Field K] [NumberField K] (Ix : Type) [Fintype Ix] (n : ℕ) (emb : Ix → E →+* K) (lam : Ix → DomWeight n), (∀ F : VV K lam, F ∈ nabAtr emb lam (stdLat E n) ↔ (F : Pol K Ix n) ∈ nabStd K lam) ∧ (ten fun t => extremal K n (lam t)) ∈ nabStd K lam ∧ ∀ x : K, x • (ten fun t => extremal K n (lam t)) ∈ nabStd K lam → ∃ o : 𝒪 K, x = (o : K)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MemSpanCuspReadingsIffComponents

theorem **spanMem_ssgt_zero** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : Data E r) (la : ReciprocityTheory.ChartRestriction.G.FD.Wt X.Ψ r) {P : Frame X.Ψ r → ℂ} (hP : P ∈ X.SS la) : P ∈ X.SSgt la 0

theorem **spanMem_rdR** {D : D7} {T : Top D} (h : LawsP T) (j : ℕ) (hj : InR D j) (u : WeightFamilies.Wt D j) (F : (recB h.toLawsCore).Cusp j u) (k : (T.U j T.σ₀).Comp) : rdR h j F k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j u k) 0

theorem **span_readings_components** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ u : WeightFamilies.Wt D j, (∀ g ∈ R.sp u Set.univ, ∃ f : ∀ k : (T.U j T.σ₀).Comp, Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ, (∀ k, f k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j u k) 0) ∧ ∀ y, g y = f (e y).1 (e y).2.1) ∧ (∀ (A : ∀ k : (T.U j T.σ₀).Comp, Set (Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ)) (s : Set ((recB h.toLawsCore).Cusp j u)), OntoBy h j u A s → ∀ f : ∀ k : (T.U j T.σ₀).Comp, Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ, (∀ k, f k ∈ A k) → (fun y => f (e y).1 (e y).2.1) ∈ R.sp u s) ∧ ∀ (F : (recB h.toLawsCore).Cusp j u) (k : (T.U j T.σ₀).Comp), rdR h j F k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j u k) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MemSpanCuspReadingsIffComponentsStatement

theorem **_root_.GlobalTheory.T.mem_span_cuspReadings_iff_components** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ u : WeightFamilies.Wt D j, (∀ g ∈ R.sp u Set.univ, ∃ f : ∀ k : (T.U j T.σ₀).Comp, Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ, (∀ k, f k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j u k) 0) ∧ ∀ y, g y = f (e y).1 (e y).2.1) ∧ (∀ (A : ∀ k : (T.U j T.σ₀).Comp, Set (Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ)) (s : Set ((recB h.toLawsCore).Cusp j u)), OntoBy h j u A s → ∀ f : ∀ k : (T.U j T.σ₀).Comp, Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ, (∀ k, f k ∈ A k) → (fun y => f (e y).1 (e y).2.1) ∈ R.sp u s) ∧ ∀ (F : (recB h.toLawsCore).Cusp j u) (k : (T.U j T.σ₀).Comp), rdR h j F k ∈ ((T.U j T.σ₀).X k).SSgt (laR T j u k) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.MinkowskiDetBoundGramOfDetNeZero

theorem **minkowski_det_bound_gram_of_det_ne_zero** (m : ℕ) (hm : 0 < m) (B : Matrix (Fin m) (Fin m) ℝ) (hB : B.det ≠ 0) (a : ℝ) (ha : 0 ≤ a) (h : ∀ z : Fin m → ℤ, z ≠ 0 → a ≤ (fun i => (z i : ℝ)) ⬝ᵥ (B * Bᵀ) *ᵥ (fun i => (z i : ℝ))) : a ^ m ≤ (m : ℝ) ^ m * (B * Bᵀ).det

theorem **minkowski_det_bound_posDef** (m : ℕ) (hm : 0 < m) (G : Matrix (Fin m) (Fin m) ℝ) (hG : G.PosDef) (a : ℝ) (ha : 0 ≤ a) (h : ∀ z : Fin m → ℤ, z ≠ 0 → a ≤ (fun i => (z i : ℝ)) ⬝ᵥ G *ᵥ (fun i => (z i : ℝ))) : a ^ m ≤ (m : ℝ) ^ m * G.det

theorem **minkowski_det_bound_posSemidef** : ∀ (m : ℕ) (G : Matrix (Fin m) (Fin m) ℝ), G.PosSemidef → ∀ a : ℝ, 0 ≤ a → (∀ z : Fin m → ℤ, z ≠ 0 → a ≤ (fun i => (z i : ℝ)) ⬝ᵥ G *ᵥ (fun i => (z i : ℝ))) → a ^ m ≤ (m : ℝ) ^ m * G.det

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ModNSqIffExistsGenLatSum

theorem **exists_dualFamily_of_nsmul_integral** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) {N : ℕ} (hN : 0 < N) (hu : ∀ v ∈ L, ∀ b, ∃ u : 𝒪 E, (u : E) = (N : E) * v b) : ∃ (m : ℕ) (w : Fin m → L) (lam : Fin m → (L →ₗ[𝒪 E] 𝒪 E)), ∀ v, ∑ i, lam i v • w i = v

theorem **mem_stdLat_iff_integral** {E : Type} [Field E] [NumberField E] {n : ℕ} (v : Fin n → E) : v ∈ stdLat E n ↔ ∀ b, ∃ u : 𝒪 E, (u : E) = v b

theorem **modN_sq_iff_exists_genLat_sum** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∀ a : Fin n → Dw.La, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) * mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a j)) ↔ ∃ (m : ℕ) (x : Fin m → Dw.La) (v : Fin m → Fin n → 𝒪 E), (∀ i, ∃ w ∈ Λ.La, ∀ b, (v i b : E) = (N : E) * w b) ∧ ∀ b, a b = ∑ i, v i b • x i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ModNSqIffExistsGenLatSumStatement

theorem **_root_.GlobalTheory.T.modN_sq_sum_iff_exists_genLat_decomposition** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∀ a : Fin n → Dw.La, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) * mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a j)) ↔ ∃ (m : ℕ) (x : Fin m → Dw.La) (v : Fin m → Fin n → 𝒪 E), (∀ i, ∃ w ∈ Λ.La, ∀ b, (v i b : E) = (N : E) * w b) ∧ ∀ b, a b = ∑ i, v i b • x i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ModelWeightSizeSpreadBound

theorem **sum_cm** {M : Type} [AddCommMonoid M] {r : ℕ} (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) (g : (E →+* ℂ) → M) : ∑ τ : X.Ψ, (g τ.1 + g (ComplexEmbedding.conjugate τ.1)) = ∑ τ, g τ

theorem **two_mul_card_cm** {r : ℕ} (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) : 2 * Fintype.card X.Ψ = Module.finrank ℚ E

theorem **sum_row_le** (r p0 : ℕ) (hp : p0 ≤ r) (u v : ℕ → ℕ) : ∑ j : Fin r, (if p0 ≤ (j : ℕ) then u ((j : ℕ) − p0) else v (p0 − 1 − (j : ℕ))) ≤ ∑ i ∈ Finset.range r, u i + ∑ i ∈ Finset.range r, v i

theorem **spread_le** {r : ℕ} (x b : Fin r → ℝ) (m : ℝ) (hm : 0 ≤ m) (hb : ∀ i, 0 ≤ b i) (hx : ∀ i, |x i| = b i + m) : (⊔ i, x i) - ⊓ i, x i ≤ ∑ i, b i + 2 * m

theorem **relab_bijective** (c : Coeff l) (σ : c.F →+* ℂ) : Function.Bijective (relab (E

private lemma **entry_abs** (p0 j s t h : ℕ) : |(((if p0 ≤ j then (s : ℤ) + (h : ℤ) else -(t : ℤ) - (h : ℤ) : ℤ)) : ℝ)| = (((if p0 ≤ j then s else t : ℕ)) : ℝ) + (h : ℝ)

theorem **laU_bounds** (U : Unf Xo c σ r) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (k : U.Comp) (B : ℕ) (hB : ∑ τ, ∑ i ∈ Finset.range r, a (τ, i) ≤ B) : (∑ τ, (U.X k).spread (laU U a n k) τ) ≤ (B : ℝ) + Fintype.card (U.X k).Ψ * n ∧ (∑ τ, ∑ i, |((laU U a n k τ i : ℤ) : ℝ)|) ≤ (B : ℝ) + Fintype.card (U.X k).Ψ * r * n / 2

theorem **sum_wtA_le** {D : D7} {j : ℕ} (w : WeightFamilies.Wt D j) (m : ℕ) : ∑ τ, ∑ i ∈ Finset.range m, wtA w (τ, i) ≤ WeightAlgebra.DF w

theorem **laU_wtA_size** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ), GoodEmb T.Q.c σ → ∀ k : (T.U j σ).Comp, (∑ τ, ((T.U j σ).X k).spread (laU (T.U j σ) (wtA w) w.n k) τ) ≤ (WeightAlgebra.DF w : ℝ) + D.R.f * w.n ∧ (∑ τ, ∑ i, |((laU (T.U j σ) (wtA w) w.n k τ i : ℤ) : ℝ)|) ≤ (WeightAlgebra.DF w : ℝ) + D.R.f * D.R.r * w.n / 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ModelWeightSizeSpreadBoundStatement

theorem **_root_.GlobalTheory.T.modelWeight_size_spread_bound** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ), GoodEmb T.Q.c σ → ∀ k : (T.U j σ).Comp, (∑ τ, ((T.U j σ).X k).spread (laU (T.U j σ) (wtA w) w.n k) τ) ≤ (WeightAlgebra.DF w : ℝ) + D.R.f * w.n ∧ (∑ τ, ∑ i, |((laU (T.U j σ) (wtA w) w.n k τ i : ℤ) : ℝ)|) ≤ (WeightAlgebra.DF w : ℝ) + D.R.f * D.R.r * w.n / 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ModuliUnivEtaKilled

theorem **section_ext** [IsReduced S] {G : Over S} [IsSeparated G.hom] (f g : 𝟙_ (Over S) ⟶ G) (h : ∀ (k : Type) [Field k] [IsAlgClosed k] (x : Sp k ⟶ S), toUnit (Over.mk x) ≫ f = toUnit (Over.mk x) ≫ g) : f = g

theorem **eta_zero** (o : Obj D l b s) : o.eta 0 = 1

theorem **eta_nsmul** (o : Obj D l b s) (n : ℕ) (a : D.La) : o.eta (n • a) = o.eta a ^ n

theorem **natCast_mem** (l : Level H) : (l.M : 𝒪 E) ∈ l.Mb

theorem **eta_pow** (o : Obj D l b s) (a : D.La) : o.eta a ^ l.M = 1

theorem **eta_smul** [IsReduced S] (o : Obj D l b s) (c : 𝒪 E) (a : D.La) : o.eta (c • a) = o.eta a ≫ o.t.act.act c

theorem **moduli_univ_eta_killed** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (a : Dw.La), (∀ b ∈ l.Mb, Xo.univ.eta a ≫ Xo.univ.t.act.act b = 1) ∧ (∀ b : 𝒪 E, Xo.univ.eta (b • a) = Xo.univ.eta a ≫ Xo.univ.t.act.act b) ∧ Xo.univ.eta a ^ l.M = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NegLogSqrtLeOfJbound

theorem **neg_log_sqrt_le_of_jbound** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (X : I2.PT E) (dg n : ℕ) (P : I2.Par X dg) (S tableLineDegreeFive cc : ℝ), (Real.pi⁻¹) ^ dg * P.Jbound n ≤ S → p14 E X.r n P.m ≤ tableLineDegreeFive → 8100 * Real.pi * n * (P.m : ℝ) ^ 4 * (I2.dstar (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NetDataExists

theorem **good_conj_iff** (c : Coeff l) (σ : c.F →+* ℂ) : GoodEmb c (ComplexEmbedding.conjugate σ) ↔ ¬ GoodEmb c σ

theorem **goodOf_conj** (c : Coeff l) (σ : c.F →+* ℂ) : goodOf c (ComplexEmbedding.conjugate σ) = goodOf c σ

theorem **supN_conj** {D : D7} {T : Top D} (h : LawsP T) (j : ℕ) (w : WeightFamilies.Wt D j) (σ : T.Q.c.F →+* ℂ) (F : (recB h.toLawsCore).Cusp j w) : (recB h.toLawsCore).supN j w (ComplexEmbedding.conjugate σ) F = (recB h.toLawsCore).supN j w σ F

theorem **netData_exists** : ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → 16 ≤ D.R.f → ∃ Nt : NetData (recB h.toLawsCore) j w, NetMax Nt ∧ NetLeSup Nt ∧ NetSize Nt (1041 * Zb D large₂ ^ 3) ∧ ∀ σ, NetInj Nt σ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NetSizeCubic

lemma **log_two_le_one** : Real.log 2 ≤ 1

lemma **eps_pos** {M f w k : ℝ} (hM : 1 ≤ M) (hf : 1 ≤ f) (hw : 0 < w) (hk : 0 < k) : 0 < epStar M f w k

lemma **neg_log_eps_le** {M f w k L : ℝ} (hM : 1 ≤ M) (hf : 1 ≤ f) (hw : 0 < w) (hk : 0 < k) (hL : 1 ≤ L) (h1 : Real.log f ≤ 2 * L) (h2 : Real.log M ≤ 30 * L) (h3 : Real.log w ≤ 3 * L) (h4 : Real.log k ≤ 4 * L) : -Real.log (epStar M f w k) ≤ 41 * L

theorem **net_size_cubic** : ∀ (f r L M w k0 dE s NA NC g : ℝ), 1 ≤ f → 1 ≤ r → 1 ≤ L → 1 ≤ M → 0 < w → 0 < k0 → 1 ≤ dE → 0 ≤ s → 1 ≤ NA → 1 ≤ NC → 1 ≤ g → Real.log f ≤ 2 * L → Real.log r ≤ L → Real.log M ≤ 30 * L → Real.log w ≤ 3 * L → Real.log k0 ≤ 4 * L → Real.log dE ≤ 4 * f * L → Real.log (1 + s) ≤ 3 * L → Real.log NA ≤ 30 * f * L → Real.log NC ≤ 30 * f * L → Real.log g ≤ 30 * f * L → 2 * f * r ^ 2 * Real.log (3 * NA * NC * g * r * (max 1 ((2 * f * r) ^ 4 * dE ^ 2 / (4 * epStar M f w k0))) ^ 2) + (2 * f * r ^ 2 * Real.log (6 * r ^ 2 / (1 / (1000 * r ^ 2 * (1 + s)))) + 2 * f * r ^ 3 * Real.log (r * max 1 ((2 * f * r) ^ 4 * dE ^ 2 / (4 * epStar M f w k0)))) + Real.log 16 ≤ 1000 * (f * r * L) ^ 3

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NoBaseComponentShortPolynomialGenerator

theorem **dom_nonempty** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : ReciprocityTheory.ChartRestriction.G.FD.Data E r) : ∃ x, x ∈ X.dom

theorem **rank_of_pins** {D : D7} {Q : Par D} (h : Q.Pins) {j : ℕ} (hj : InR D j) : (Q.Dw j).p + (Q.Dw j).q = D.R.r

theorem **log_rdE** (D : D7) : Real.log (FD.rdE D.host.E) = lrd D

theorem **mem_span_short** (D : D7) (T : Top D) (h : LawsP T) (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j) (hrows : PullbackSections.SupBoundPageRows D large₂ w) (hL : 2 ≤ large₂.sc.L) (hr : D.R.r ≤ w.n) (P : MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F) (hP : P ∈ harmA T.Q.c (wtA w) w.n) : P ∈ Submodule.span T.Q.c.F {Q | Q ∈ harmA (Dw

theorem **noBase** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ReciprocityTheory.ChartRestriction.G.PullbackSections.NoBaseSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NoBaseComponentShortPolynomialGeneratorStatement

theorem **_root_.GlobalTheory.T.no_base_component_short_polynomial_generator** :
Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ReciprocityTheory.ChartRestriction.G.PullbackSections.NoBaseSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NoCommonZeroOfReadings

theorem **exists_complex_point** {K A : Type} [Field K] [CommRing A] [Algebra K A] [Algebra.FiniteType K A] (σ : K →+* ℂ) (I : Ideal A) (hI : I ≠ ⊤) : ∃ ψ : A →ₐ[K] CAt σ, ∀ x ∈ I, ψ x = 0

theorem **noCommonZero_of_readings** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (a : (D.host.E →+* ℂ) × ℕ → ℕ) (n m : ℕ) (ws : Fin m → T.R j), (∀ i, ws i ∈ sec (T.Q.Xo j) T.Q.c a n) → (∀ k, ∀ x ∈ ((T.U j T.σ₀).X k).dom, ∃ i, rdU (T.U j T.σ₀) (ws i) k x ≠ 0) → NoCommonZero (T.Q.Xo j) T.Q.c ws

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NonemptyHeightOneSpectrumRingOfIntegers

theorem **nonempty_heightOneSpectrum_ringOfIntegers** (K : Type) [Field K] [NumberField K] : Nonempty (HeightOneSpectrum (𝒪 K))

theorem **shareRhoP_of_anFacts** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (N : ℕ) (w : Fin N → B.Cusp j wL) (R : Rdg X) (t : Idx D j N → ℂ) (T : ℕ) (CR c : ℝ) (δ : ℕ) (xs : Fin δ → R.Y), AnFacts R w t ka wF T (WeightFamilies.MggK B CR T c j ka) xs → 2 ≤ WeightFamilies.dimY D j → 11 * WeightFamilies.dimY D j ^ 2 ≤ T → LatAt B j ka → ShareRhoP (D

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NonemptyTorusAt

theorem **nonempty_torusAt** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : DataX E r) (w0 : 𝒪 E) (hw : ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) : ∀ x ∈ X.dom, Nonempty (TorusAt X.toData x)

theorem **exists_weight** : TStatement → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ ξ : Fin r → E, X.toData.IsIso ξ) → ((∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) ∧ (PT.f E : ℝ) ≤ PT.rd E ∧ (PT.f E : ℝ) ^ 2 ≤ PT.rd E ∧ 16 ≤ PT.rd E ∧ ∃ m : ℕ, 1 ≤ m ∧ m ^ 4 ≤ PT.f E ∧ (X.p0 ≤ m ∨ r - X.p0 ≤ m)) → ∃ ps : Frame X.Ψ r → ℝ, IsInvFn X.toData ps ∧ (∀ x0 ∈ X.dom, ContDiffOn ℝ ∞ (fun z => ps (cpt X.toData x0 z)) {z | cpt X.toData x0 z ∈ X.dom}) ∧ (∀ x0 ∈ X.dom, ∀ z, cpt X.toData x0 z ∈ X.dom → ∀ ξ : Fin (nd X.toData) → ℂ, -((r : ℝ) / 2 * leviM X.toData (phla X.toData (fun _ j => if X.p0 ≤ (j : ℕ) then 1 else -1)) x0 z ξ) ≤ leviM X.toData ps x0 z ξ) ∧ cfwM X.toData ps (1 / (2 * (5 : ℝ) ^ 4 * |(NumberField.discr E : ℝ)| ^ ((1 : ℝ) / X.Ψ.card)))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormPowMulProdTranslConst

theorem **coe_bas_eq_sum** (m : Fin g ⊕ Fin g → ℤ) : (bas m : Fin r → E) = ∑ s, m s • (bas (Pi.single s 1) : Fin r → E)

theorem **ratPt_bijective** : Function.Bijective (ratPt bas)

theorem **polyDeriv_prod_canFn_eq_mul** (hs : IsSympl X bas) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {a b a' b' : Fin n → Fin g → ℝ} {c : Fin n → ℂ} (h : ∀ k z, canFn X bas' idx (a' k) (b' k) x z = c k * canFn X bas idx (a k) (b k) x z) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) : polyDeriv P (letterDir X hr idx) (fun z => ∏ k, canFn X bas' idx (a' k) (b' k) x (z k)) 0 = (∏ k, c k) * polyDeriv P (letterDir X hr idx) (fun z => ∏ k, canFn X bas idx (a k) (b k) x (z k)) 0

theorem **IsProportional.thFr_eq** {κ : ℂ} {a₀ b₀ a₀' b₀' : Fin g → ℝ} (h : IsProportional X idx κ bas a₀ b₀ bas' a₀' b₀') {u u' : Fin n → Fin g ⊕ Fin g → ℝ} (hu : ∀ k, IsSamePt X bas bas' (u k) (u' k)) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : thFr X hr bas' idx n (fun k => a₀' + u' k ∘ Sum.inr) (fun k => b₀' + u' k ∘ Sum.inl) P x = κ ^ (n / 2) * (∏ k, translConst b₀ (u k ∘ Sum.inr) (u k ∘ Sum.inl) / translConst b₀' (u' k ∘ Sum.inr) (u' k ∘ Sum.inl)) * thFr X hr bas idx n (fun k => a₀ + u k ∘ Sum.inr) (fun k => b₀ + u k ∘ Sum.inl) P x

theorem **norm_pow_mul_prod_translConst** {κ : ℂ} (hκ : ‖κ‖ = 1) (j : ℕ) (b b' : Fin g → ℝ) (s t s' t' : Fin n → Fin g → ℝ) : ‖κ ^ j * ∏ k, translConst b (s k) (t k) / translConst b' (s' k) (t' k)‖ = 1

theorem **exists_thFr_eq_mul** (a b : Fin n → Fin g → ℝ) : ∃ (a' b' : Fin n → Fin g → ℝ) (c : ℂ), ‖c‖ = 1 ∧ ((∀ k i, a k i ∈ Set.range (Rat.cast : ℚ → ℝ)) → (∀ k i, b k i ∈ Set.range (Rat.cast : ℚ → ℝ)) → (∀ k i, a' k i ∈ Set.range (Rat.cast : ℚ → ℝ)) ∧ ∀ k i, b' k i ∈ Set.range (Rat.cast : ℚ → ℝ)) ∧ ∀ (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)), ∀ x ∈ X.dom, thFr X hr bas idx n a b P x = c * thFr X hr bas' idx n a' b' P x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormProdSignMulDetSubmatrixSubOneLe

theorem **norm_multisetProd_sub_one_le_exp** (s : Multiset ℂ) {t : ℝ} (ht : 0 ≤ t) (hs : ∀ μ ∈ s, ‖μ - 1‖ ≤ t) : ‖s.prod - 1‖ ≤ Real.exp (s.card * t) - 1

theorem **norm_det_one_add_sub_one_le_exp** {n : Type*} [Fintype n] [DecidableEq n] (M : Matrix n n ℂ) {ε : ℝ} (hε : 0 ≤ ε) (hM : ∀ a b, ‖M a b‖ ≤ ε) : ‖(1 + M).det - 1‖ ≤ Real.exp (Fintype.card n * (Fintype.card n * ε)) - 1

theorem **norm_gram_sign_sub_le** {r : ℕ} (j : Fin r → ℂ) (hj : ∀ a, ‖j a‖ = 1) (z : Matrix (Fin r) (Fin r) ℂ) {δ : ℝ} (hδ : 0 ≤ δ) (hr : r * δ ≤ 1) (hz : ∀ a b, ‖z a b‖ ≤ δ) (a b : Fin r) : ‖((1 + z)ᵀ * diagonal j * (1 + z)).map star - diagonal j : Matrix (Fin r) (Fin r) ℂ) a b‖ ≤ 3 * δ

theorem **norm_prod_sign_mul_det_submatrix_sub_one_le** {r : ℕ} (j : Fin r → ℂ) (hj : ∀ a, j a = 1 ∨ j a = -1) (G : Matrix (Fin r) (Fin r) ℂ) {ε : ℝ} (hε : 0 ≤ ε) (hG : ∀ a b, ‖(G - diagonal j) a b‖ ≤ ε) (hr : (r : ℝ) ^ 2 * ε ≤ 1) {i : ℕ} {e : Fin i → Fin r} (he : Function.Injective e) : ‖(∏ a, j (e a)) * (G.submatrix e e).det - 1‖ ≤ 2 * ((r : ℝ) ^ 2 * ε)

theorem **re_pos_and_abs_log_norm_le** {d : ℂ} {t : ℝ} (ht : t ≤ 1 / 6) (hd : ‖d - 1‖ ≤ 2 * t) : 0 < d.re ∧ |Real.log ‖d‖| ≤ 3 * t

theorem **gram_eq_transpose_mul_mul_map_star** {r : ℕ} (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (x : Matrix (Fin r) (Fin r) ℂ) : gram τ G x = xᵀ * G.map τ * x.map star

theorem **gram_mul_eq_transpose_mul_gram_mul** {r : ℕ} (τ : E →+* ℂ) (G : Matrix (Fin r) (Fin r) E) (x A : Matrix (Fin r) (Fin r) ℂ) : gram τ G (x * A) = Aᵀ * gram τ G x * A.map star

theorem **dmin_rmul_near_one** {r : ℕ} (X : Data E r) (x : Frame X.Ψ r) (hx : ∀ τ : X.Ψ, gram τ.1 X.Gm (mat x τ) = Matrix.diagonal fun a : Fin r => ((sgn X.p0 (a : ℕ) : ℝ) : ℂ)) (A : Frame X.Ψ r) {δ : ℝ} (hδ : 0 ≤ δ) (hr : (r : ℝ) ^ 2 * δ ≤ 1 / 100) (hA : ∀ τ a b, ‖A τ a b - (if a = b then 1 else 0)‖ ≤ δ) : rmul x A ∈ X.dom ∧ ∀ τ i, dmin X.Gm (rmul x A) τ i ≠ 0 ∧ |Real.log ‖dmin X.Gm (rmul x A) τ i‖| ≤ 9 * ((r : ℝ) ^ 2 * δ)

theorem **exists_pnorm_eq_mul_exp** {r : ℕ} (X : Data E r) (la : Wt X.Ψ r) (y : Frame X.Ψ r) {t : ℝ} (hy : ∀ τ i, dmin X.Gm y τ i ≠ 0 ∧ |Real.log ‖dmin X.Gm y τ i‖| ≤ t) : ∃ s : ℝ, |s| ≤ t * ∑ τ, ∑ i, |(la τ i : ℝ)| ∧ ∀ P : Frame X.Ψ r → ℂ, X.pnorm la P y = ‖P y‖ * Real.exp s

theorem **rmul_one_frame** {Ψ : Finset (E →+* ℂ)} {r : ℕ} (x : Frame Ψ r) : rmul x (fun _ a b => if a = b then 1 else 0) = x

theorem **differentiable_rmul_add** {Ψ : Finset (E →+* ℂ)} {r : ℕ} (x I : Frame Ψ r) : Differentiable ℂ fun z : Frame Ψ r => rmul x (I + z)

theorem **pnorm_near_half_max** {r : ℕ} (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) (hP : DifferentiableOn ℂ P X.dom) (S : ℝ) (hS : ∀ x ∈ X.dom, X.pnorm la P x ≤ S) (x : Frame X.Ψ r) (hx : ∀ τ : X.Ψ, gram τ.1 X.Gm (mat x τ) = Matrix.diagonal fun a : Fin r => ((sgn X.p0 (a : ℕ) : ℝ) : ℂ)) (hhalf : S / 2 ≤ X.pnorm la P x) (A : Frame X.Ψ r) (hA : ∀ τ a b, ‖A τ a b − (if a = b then 1 else 0)‖ ≤ 1 / (1000 * (r : ℝ) ^ 2 * (1 + ∑ τ, ∑ i, |(la τ i i : ℝ)|))) : rmul x A ∈ X.dom ∧ S / 4 ≤ X.pnorm la P (rmul x A)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormQuotientThetaFlatAttainsSupOffCommonZeros

theorem **sat_open** (X : Data E r) {x1 : Frame X.Ψ r} (hx1 : x1 ∈ X.dom) : ∃ ρ0 : ℝ, 0 < ρ0 ∧ ∀ ρ : ℝ, ρ ≤ ρ0 → IsOpen {x | ∃ γ ∈ X.Γ, ∃ b ∈ Bor X.Ψ r, ∃ z' ∈ Metric.ball (0 : V (nd X)) ρ, x = lact γ (rmul (cpt X x1 z') b)}

theorem **pnorm_nonneg** (X : Data E r) (la : Wt X.Ψ r) (P : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : 0 ≤ X.pnorm la P x

theorem **thetaFac_nonneg** (x : Frame X.Ψ r) : 0 ≤ thetaFac X Xa x

theorem **sumSq_nonneg** (x : Frame X.Ψ r) : 0 ≤ sumSq X Xa Fl x

theorem **quot_inv** (hf : X.IsSec (Xa.Fp + Xa.T • Xa.L) f) {γ : GL (Fin r) E} (hγ : γ ∈ X.Γ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) : quot X Xa Fl f (lact γ (rmul x b)) = quot X Xa Fl f x

theorem **quot_contOn** (hf : X.IsSec (Xa.Fp + Xa.T • Xa.L) f) {A : Set (Frame X.Ψ r)} (hA : A ⊆ X.dom) (hS : ∀ x ∈ A, 0 < sumSq X Xa Fl x) : ContinuousOn (quot X Xa Fl f) A

theorem **thetaFac_le** : ∃ C : ℝ, 0 < C ∧ ∀ x ∈ X.dom, thetaFac X Xa x ≤ C * Real.exp (-(4 / (Xa.g0 : ℝ)) * Xa.ph x)

theorem **thetaFac_bdd** : ∃ C : ℝ, 0 < C ∧ ∀ x ∈ X.dom, thetaFac X Xa x ≤ C

theorem **sumSq_lower** (j : Fin Fl.δ) : ∃ d : ℝ, 0 < d ∧ ∀ᶠ z in 𝒩 (0 : V (nd X)), d * ‖z‖ ^ 2 ≤ sumSq X Xa Fl (cpt X (Fl.xs j) z)

theorem **quot_near** (j : Fin Fl.δ) (hO : (fun z : V (nd X) => f (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (Xa.T + nd X + 7)) {ε : ℝ} (hε : 0 < ε) : ∀ᶠ z in 𝒩 (0 : V (nd X)), cpt X (Fl.xs j) z ∈ X.dom ∧ quot X Xa Fl f (cpt X (Fl.xs j) z) < ε

theorem **sumSq_far** {U : Set (Frame X.Ψ r)} (hU : IsOpen U) (hxs : ∀ j, Fl.xs j ∈ U) (hsat : ∀ x ∈ U, ∀ γ ∈ X.Γ, ∀ b ∈ Bor X.Ψ r, lact γ (rmul x b) ∈ U) : ∃ d : ℝ, 0 < d ∧ ∀ x ∈ X.dom, x ∉ U → d * Real.exp (-(2 * Xa.c * Xa.ph x)) ≤ sumSq X Xa Fl x

theorem **rate_pos** : 0 < 4 / (Xa.g0 : ℝ) − 2 * Xa.c * ((Xa.T : ℝ) + nd X + 6 + 1 / 2)

theorem **quot_far** (hf : X.IsBddSec (Xa.Fp + Xa.T • Xa.L) f) {A : Set (Frame X.Ψ r)} {d : ℝ} (hd : 0 < d) (hA : ∀ x ∈ X.dom, x ∈ A → d * Real.exp (-(2 * Xa.c * Xa.ph x)) ≤ sumSq X Xa Fl x) {ε : ℝ} (hε : 0 < ε) : ∃ δ : ℝ, 0 < δ ∧ ∀ x ∈ X.dom, x ∈ A → X.la1 x < δ → quot X Xa Fl f x < ε

theorem **exists_max** (hf : X.IsBddSec (Xa.Fp + Xa.T • Xa.L) f) (hO : ∀ j, (fun z : V (nd X) => f (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (Xa.T + nd X + 7)) (hpos : ∃ x ∈ X.dom, 0 < quot X Xa Fl f x) : ∃ x0 ∈ X.dom, (∃ k, Fl.s k x0 ≠ 0) ∧ 0 < quot X Xa Fl f x0 ∧ ∀ x ∈ X.dom, quot X Xa Fl f x ≤ quot X Xa Fl f x0

theorem **main** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (f : Frame X.Ψ r → ℂ), X.IsBddSec (Xa.Fp + Xa.T • Xa.L) f → (∀ j, (fun z : V (nd X) => f (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (Xa.T + nd X + 7)) → let Φ : Frame X.Ψ r → ℝ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormQuotientThetaFlatAttainsSupOffCommonZerosStatement

theorem **_root_.GlobalTheory.T.normQuotientThetaFlat_attains_sup_off_common_zeros** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (f : Frame X.Ψ r → ℂ), X.IsBddSec (Xa.Fp + Xa.T • Xa.L) f → (∀ j, (fun z : V (nd X) => f (cpt X (Fl.xs j) z)) =O[𝒩 0] fun z => ‖z‖ ^ (Xa.T + nd X + 7)) → let Φ : Frame X.Ψ r → ℝ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormShortGeneratorEvalLeOrthonormalFrame

theorem **isSB_smul** (c : ℂ) {φ : TF n g} (h : IsSB φ) : IsSB (c • φ)

theorem **thA_sum** (idx : Fin g ≃ X.Ψ × Fin r) (hs : IsSympl X bas) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {T : Type} (s : Finset T) (f : T → TF n g) (hf : ∀ i, IsSB (f i)) : thA X bas idx (∑ i ∈ s, f i) x = ∑ i ∈ s, thA X bas idx (f i) x

theorem **theta_sum₂** (hr : X.p0 + q0 = r) (idx : Fin g ≃ X.Ψ × Fin r) (hs : IsSympl X bas) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {T : Type} (s : Finset T) (W : T → ℂ) (f : T → TF n g) (hf : ∀ i, IsSB (f i)) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) : thP X hr bas idx (∑ i ∈ s, W i • f i) P x = ∑ i ∈ s, W i * thP X hr bas idx (f i) P x

theorem **norm_mean_le** (hr : X.p0 + q0 = r) (idx : Fin g ≃ X.Ψ × Fin r) (hs : IsSympl X bas) {x : Frame X.Ψ r} (hx : x ∈ X.dom) {T : Type} (W : T → ℂ) (f : T → TF n g) (hf : ∀ i, IsSB (f i)) (P : Harm.Poly ℂ X.Ψ q0 X.p0 (Fin n)) (κ : ℂ) {C B : ℝ} (hC : 0 ≤ C) (hB : 0 ≤ B) (hW : ∀ i, ‖W i‖ ≤ C) (hb : ∀ i, ‖κ * thP X hr bas idx (f i) P x‖ ≤ B) : ‖((Nat.card T : ℂ))⁻¹ * (κ * thP X hr bas idx (∑ᶠ i, W i • f i) P x)‖ ≤ C * B

theorem **index_Ln** (Λ : GenLat E n) (h : stdLat E n = stdL E n) : (Λ.Ln.toAddSubgroup.addSubgroupOf Λ.La.toAddSubgroup).index = (stdL E n).toAddSubgroup.relIndex Λ.La.toAddSubgroup

theorem **zCard_eq** (Λ : GenLat E n) (h : stdLat E n = stdL E n) {N M : ℕ} (hN : Λ.LevOK N M) (r : ℕ) : Λ.zCard N r = Λ.mIdx N r * ((stdL E n).toAddSubgroup.relIndex Λ.La.toAddSubgroup) ^ r

theorem **map_mem_HHa** {T : Type} (m : T → ℕ) (n : ℕ) {F F' : IntermediateField ℚ ℂ} (h : F = F') (cc : T → T) (isX : T → Prop) (a : T → ℕ → ℕ) {P : MvPolynomial (Letter T m n) F} (hP : P ∈ HHa m n F cc isX a) : MvPolynomial.map (IntermediateField.equivOfEq h : F →+* F') P ∈ HHa m n F' cc isX a

theorem **sum_coeff_rename** {F S S' : Type} [Field F] (σ : F →+* ℂ) (e : S ≃ S') (P : MvPolynomial S F) : ∑ m ∈ (MvPolynomial.rename e (MvPolynomial.map σ P)).support, ‖(MvPolynomial.rename e (MvPolynomial.map σ P)).coeff m‖ = ∑ m ∈ P.support, ‖σ (P.coeff m)‖

theorem **norm_index_cancellation_le_bound** {b κ t : ℂ} {m z Q : ℕ} {J C B : ℝ} {j : ℕ} (hJ : J = j) (hj : 0 < j) (hz : z = m * j) (hb : ‖b‖ ^ 2 = J) (hC : 0 ≤ C * (B / Real.sqrt J)) (h : ‖((Q : ℂ))⁻¹ * (κ * t)‖ ≤ C * (B / Real.sqrt J)) : ‖b⁻¹ * (m : ℂ)⁻¹ * (Q : ℂ)⁻¹ * (z : ℂ) * κ * t‖ ≤ C * B

theorem **norm_evR_le** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → 1 ≤ r → ∀ t : ℝ, 0 < t → (∀ k τ, (U.X k).c τ = t) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → r ≤ n → ∀ d : DeltaDat Xo c n, d.IsLev → 0 < d.pd → l.A = 1 → d.l₂.A = 1 → (∃ _ : StarRing (𝒪 E), IsNab d.Xo₂ d.c₂ d.nab₂) → (∃ ph ψ x₀, d.PhiPin ph ψ x₀) → UnitC d → ∀ (Λ : GenLat E n) (N : ℕ) (x0 : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F) (f : Rul Xo c), Λ.Unimod → Λ.TrivAt d.l₂.M → Λ.LevOK N d.l₂.M → (∃ β : c.F, β ≠ 0 ∧ BetaIs (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormShortGeneratorEvalLeOrthonormalFrameStatement

theorem **_root_.GlobalTheory.T.norm_short_generator_eval_le_orthonormal_frame** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → 1 ≤ r → ∀ t : ℝ, 0 < t → (∀ k τ, (U.X k).c τ = t) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → r ≤ n → ∀ d : DeltaDat Xo c n, d.IsLev → 0 < d.pd → l.A = 1 → d.l₂.A = 1 → (∃ _ : StarRing (𝒪 E), IsNab d.Xo₂ d.c₂ d.nab₂) → (∃ ph ψ x₀, d.PhiPin ph ψ x₀) → UnitC d → ∀ (Λ : GenLat E n) (N : ℕ) (x0 : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F) (f : Rul Xo c), Λ.Unimod → Λ.TrivAt d.l₂.M → Λ.LevOK N d.l₂.M → (∃ β : c.F, β ≠ 0 ∧ BetaIs (Dw

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormalisedPullbackMulNReadsTheta

theorem **mulN_e_zero** : Φ.e 0 = 1

theorem **mulN_e_nsmul** (N : ℕ) (z : Fin g → ℂ) : Φ.e ((N : ℂ) • z) = Φ.e z ^ N

theorem **mulN_comp** {Z : Over S} (y : Z ⟶ A) (N : ℕ) : y ≫ mulA A N = y ^ N

theorem **mulN_eSec** (N : ℕ) : eSec A ≫ (mulA A N).left = eSec A

theorem **mulN_genRd** (N : ℕ) (α₀ : LB.Gen (LB.LinComb L.L (K

theorem **rd_mulN** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (N : ℕ) (α₀ : LB.Gen (LB.LinComb L.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormalisedPullbackMulNReadsThetaStatement

theorem **_root_.GlobalTheory.T.normalisedPullbackMulN_reads_theta** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (N : ℕ) (α₀ : LB.Gen (LB.LinComb L.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormalisedThetaNormInvariant

theorem **exists_eq_smul** (T : PT E) {s t : (Fin 1 → Vsp T.dim) → ℂ} (hs : s ∈ T.Sec 1 1) (ht : t ∈ T.Sec 1 1) (hs0 : s ≠ 0) : ∃ c : ℂ, t = c • s

theorem **hf_map** (v w : Vsp T.dim) : Hf (U v) (U w) = Hf v w

theorem **pointwiseNormSq_comp_isometry** (s : (Fin 1 → Vsp T'.dim) → ℂ) (y : Vsp T.dim) : T.ptN2 1 (fun w => s fun k => U (w k)) (fun _ => y) = T'.ptN2 1 s fun _ => U y

theorem **comp_mem_Sec** (hL : ∀ y, y ∈ T.L ↔ U y ∈ T'.L) (he : ∀ y ∈ T.L, T'.eps (U y) = T.eps y) {s : (Fin 1 → Vsp T'.dim) → ℂ} (hs : s ∈ T'.Sec 1 1) : (fun w => s fun k => U (w k)) ∈ T.Sec 1 1

theorem **l2_comp_eq_of_preserves_lattice_eps** (hL : ∀ y, y ∈ T.L ↔ U y ∈ T'.L) (he : ∀ y ∈ T.L, T'.eps (U y) = T.eps y) {s : (Fin 1 → Vsp T'.dim) → ℂ} (hs : s ∈ T'.Sec 1 1) : T.l2 1 (fun w => s fun k => U (w k)) = T'.l2 1 s

theorem **ratio_eq** (hL : ∀ y, y ∈ T.L ↔ U y ∈ T'.L) (he : ∀ y ∈ T.L, T'.eps (U y) = T.eps y) {s : (Fin 1 → Vsp T.dim) → ℂ} {s' : (Fin 1 → Vsp T'.dim) → ℂ} (hs : s ∈ T.Sec 1 1) (hs0 : s ≠ 0) (hs' : s' ∈ T'.Sec 1 1) (hs0' : s' ≠ 0) (y : Vsp T.dim) : T'.ptN2 1 s' (fun _ => U y) / T'.l2 1 s' = T.ptN2 1 s (fun _ => y) / T.l2 1 s

theorem **exists_diag_isometry** {ι ι' : Type} [Fintype ι] [Fintype ι'] (ε : ι' ≃ ι) (w : ι' → ℂ) (hw : ∀ i, ‖w i‖ = 1) : ∃ U : EuclideanSpace ℂ ι ≃ₗᵢ[ℂ] EuclideanSpace ℂ ι', ∀ y i, U y i = w i * y (ε i)

theorem **hermC_map_mulVec** (G : Matrix (Fin r) (Fin r) E) {γ : Matrix (Fin r) (Fin r) E} (hγ : ∀ x y, herm G (γ.mulVec x) (γ.mulVec y) = herm G x y) (τ : E →+* ℂ) (u w : Fin r → ℂ) : hermC τ G ((γ.map τ).mulVec u) ((γ.map τ).mulVec w) = hermC τ G u w

theorem **hermC_eq_zero_of_mem_span** (X : Data E r) (τ : X.Ψ) (e : Fin r → Fin r → ℂ) (he : ∀ a b, hermC τ.1 X.Gm (e a) (e b) = if a = b then ((sgn X.p0 a : ℝ) : ℂ) else 0) {i : ℕ} (hi : i ≤ r) {y : Fin r → ℂ} (hy : y ∈ Submodule.span ℂ (Set.range fun a : Fin i => e (Fin.castLE hi a))) (c : Fin r) (hc : i ≤ c) : hermC τ.1 X.Gm y (e c) = 0

theorem **exists_unit_of_flag** (X : Data E r) (τ : X.Ψ) (e e' : Fin r → Fin r → ℂ) (he : ∀ a b, hermC τ.1 X.Gm (e a) (e b) = if a = b then ((sgn X.p0 a : ℝ) : ℂ) else 0) (he' : ∀ a b, hermC τ.1 X.Gm (e' a) (e' b) = if a = b then ((sgn X.p0 a : ℝ) : ℂ) else 0) (hfl : ∀ (i : ℕ) (hi : i ≤ r), Submodule.span ℂ (Set.range fun a : Fin i => e' (Fin.castLE hi a)) ≤ Submodule.span ℂ (Set.range fun a : Fin i => e (Fin.castLE hi a))) (a : Fin r) : ∃ u : ℂ, ‖u‖ = 1 ∧ e' a = u • e a

theorem **exists_unit_frame** (X : Data E r) {x b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) {γ : GL (Fin r) E} (hγ : IsUStab X.Gm X.Λ γ) (A : TorusAt X x) (A' : TorusAt X (lact γ (rmul x b))) (τ : X.Ψ) (a : Fin r) : ∃ u : ℂ, ‖u‖ = 1 ∧ ∀ v : Fin r → E, hermC τ.1 X.Gm (Data.emb τ.1 ((γ : Matrix (Fin r) (Fin r) E).mulVec v)) (A'.e τ a) = u * hermC τ.1 X.Gm (Data.emb τ.1 v) (A.e τ a)

theorem **exists_isometry** (X : Data E r) {x b : Frame X.Ψ r} (hb : b ∈ Bor X.Ψ r) {γ : GL (Fin r) E} (hγ : IsUStab X.Gm X.Λ γ) (A : TorusAt X x) (A' : TorusAt X (lact γ (rmul x b))) : ∃ U : Vsp A.T.dim ≃ₗᵢ[ℂ] Vsp A'.T.dim, ∀ v, U (A.φ v) = A'.φ ((γ : Matrix (Fin r) (Fin r) E).mulVec v)

theorem **normalised_theta_norm_invariant** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) − cc (w0 : E)) • v)) − Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) → ∀ x ∈ X.dom, ∀ b ∈ Bor X.Ψ r, ∀ γ : GL (Fin r) E, IsUStab X.Gm X.Λ γ → ∀ (A : TorusAt X.toData x) (A' : TorusAt X.toData (lact γ (rmul x b))) (v : Fin r → E), A'.T.ptN2 1 A'.s (fun _ => A'.φ ((γ : Matrix (Fin r) (Fin r) E).mulVec v)) / A'.T.l2 1 A'.s = A.T.ptN2 1 A.s (fun _ => A.φ v) / A.T.l2 1 A.s ∧ ∀ u ∈ X.Λ, A.T.ptN2 1 A.s (fun _ => A.φ (v + u)) / A.T.l2 1 A.s = A.T.ptN2 1 A.s (fun _ => A.φ v) / A.T.l2 1 A.s

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NormalisedThetaValueBound

theorem **isogeny_sum_tensorLat** (T : PT E) (w₀ : 𝒪 E) (hw : (w₀ : E) + cc (w₀ : E) = 1) (heps : ∀ x ∈ T.L, (T.eps x : ℝ) = Real.cos (Real.pi * (Hf x (act w₀ x)).im)) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (R : Finset (Fin n → Vsp T.dim)) (hR : ∀ y ∈ R, y ∈ tensorLat T Λ') (hR' : ∀ y ∈ tensorLat T Λ', ∃! r, r ∈ R ∧ ∀ k, y k − r k ∈ T.L) (F : (Fin n → Vsp T.dim) → ℂ) (hF : F ∈ T.Sec 1 n) (Y' : GT (Fin n × Idx T.dim)) (hY' : ∀ γ, flat γ ∈ lat Y'.b ↔ γ ∈ tensorLat T Λ') (hka' : ∀ γ, Y'.ka (flat γ) = tensorEps w₀ γ) : (∑ y ∈ R, tensorEps w₀ y • T.U 1 y F) ∘ flat.symm ∈ Y'.Sec 1 ∧ Y'.l2 1 ((∑ y ∈ R, tensorEps w₀ y • T.U 1 y F) ∘ flat.symm) = R.card * T.l2 1 F

theorem **exists_min_period** (T : PT E) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') {lam : ℝ} (hlam : 0 < lam) (hmin : ∀ x ∈ T.L, x ≠ 0 → lam ≤ ‖x‖) : ∃ lam' : ℝ, 0 < lam' ∧ (∀ γ ∈ tensorLat T Λ', γ ≠ 0 → lam' ≤ ‖flat γ‖) ∧ (1 + 4 * Real.sqrt (2 * Fintype.card (Fin n × Idx T.dim)) / (Real.sqrt Real.pi * lam')) ^ Fintype.card (Fin n × Idx T.dim) = (1 + 4 * Real.sqrt (2 * Fintype.card (Idx T.dim) * n) * (2 * (fdeg E : ℝ) ^ ((3 : ℝ) / 2) * n ^ 2 * rdE E) / (Real.sqrt Real.pi * lam)) ^ (Fintype.card (Idx T.dim) * n)

theorem **norm_sum_jet_isogeny_sum_le** (T : PT E) (w₀ : 𝒪 E) (hw : (w₀ : E) + cc (w₀ : E) = 1) (heps : ∀ x ∈ T.L, (T.eps x : ℝ) = Real.cos (Real.pi * (Hf x (act w₀ x)).im)) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (R : Finset (Fin n → Vsp T.dim)) (hR : ∀ y ∈ R, y ∈ tensorLat T Λ') (hR' : ∀ y ∈ tensorLat T Λ', ∃! r, r ∈ R ∧ ∀ k, y k − r k ∈ T.L) (F : (Fin n → Vsp T.dim) → ℂ) (hF : F ∈ T.Sec 1 n) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ γ ∈ T.L, γ ≠ 0 → lam ≤ ‖γ‖) {κ : Type} (J : Finset κ) (c : κ → ℂ) (hc : ∀ j ∈ J, ‖c j‖ ≤ 1) (y : κ → Fin n → Vsp T.dim) (P : MvPolynomial (Letter n T.dim) ℂ) (a : ℕ) (hP : P.IsHomogeneous a) : ‖∑ j ∈ J, c j * polyDeriv P dir (T.U 1 (y j) (∑ r ∈ R, tensorEps w₀ r • T.U 1 r F)) 0‖ ≤ J.card * (∑ m ∈ P.support, ‖P.coeff m‖) * (Real.pi * Real.exp

theorem **normalised_theta_value_bound** {E : Type} [Field E] [NumberField E] [IsCMField E] {n : ℕ} (T : PT E) (w₀ : 𝒪 E) (hw : (w₀ : E) + cc (w₀ : E) = 1) (heps : ∀ x ∈ T.L, (T.eps x : ℝ) = Real.cos (Real.pi * (Hf x (act w₀ x)).im)) (s : (Fin 1 → Vsp T.dim) → ℂ) (hs : s ∈ T.Sec 1 1) (hs0 : s ≠ 0) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (R : Finset (Fin n → Vsp T.dim)) (hR : ∀ y ∈ R, y ∈ tensorLat T Λ') (hR' : ∀ y ∈ tensorLat T Λ', ∃! r, r ∈ R ∧ ∀ k, y k − r k ∈ T.L) (β : ℂ) (hβ : ‖β‖ ^ 2 = R.card) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ γ ∈ T.L, γ ≠ 0 → lam ≤ ‖γ‖) {κ : Type} (J : Finset κ) (c : κ → ℂ) (hc : ∀ j ∈ J, ‖c j‖ ≤ 1) (y : κ → Fin n → Vsp T.dim) (P : MvPolynomial (Letter n T.dim) ℂ) (a : ℕ) (hP : P.IsHomogeneous a) : β ≠ 0 ∧ 0 < T.l2 1 s ∧ ‖β⁻¹ * ∑ j ∈ J, c j *

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NotIsBlockProdZero

theorem **poly_ne_zero** (K : Type) [Field K] (B : Bk Dw n) : B.poly K ≠ 0

theorem **not_isBlockProd_zero** (K : Type) [Field K] : ¬ IsBlockProd (0 : MvPolynomial (LtN Dw n) K)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.NsmulApplyMemOfMemTensorLat

theorem **eq_sum_tens_contr** {ι : Type*} [Fintype ι] {v w : ι → Fin n → E} (h : ∀ u : Fin n → E, u = ∑ i, herm 1 u (w i) • v i) (γ : Fin n → Vsp dim) : γ = ∑ i, tens (contr γ (w i)) (v i)

theorem **tens_mem_of_integral** (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) {x : Vsp X.dim} (hx : x ∈ X.L) {v : Fin n → E} (hv : v ∈ Λ') : tens x v ∈ tensorLat X Λ'

theorem **tensorLat_eq_span** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') : tensorLat X Λ' = Submodule.span ℤ {γ | ∃ x ∈ X.L, ∃ v ∈ Λ', γ = tens x v}

theorem **im_HC_int** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') {γ γ' : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') (hγ' : γ' ∈ tensorLat X Λ') : ∃ k : ℤ, (HC γ γ').im = k

theorem **exists_im_HC_eq** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') (φ : tensorLat X Λ' →+ ℤ) : ∃ γ ∈ tensorLat X Λ', ∀ γ' : tensorLat X Λ', (φ γ' : ℝ) = (HC γ (γ' : Fin n → Vsp X.dim)).im

private lemma **tens_nsmul** (x : Vsp dim) (v : Fin n → E) (N : ℕ) : tens x (N • v) = N • tens x v

theorem **nsmul_mem_tensorLat** (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) {N : ℕ} (hN : ∀ u ∈ stdLat E n, N • u ∈ Λ') {γ : Fin n → Vsp X.dim} (hγ : ∀ k, γ k ∈ X.L) : N • γ ∈ tensorLat X Λ'

theorem **nsmul_apply_mem_of_mem_tensorLat** {N : ℕ} (hN : ∀ u ∈ stdLat E n, N • u ∈ Λ') {γ : Fin n → Vsp X.dim} (hγ : γ ∈ tensorLat X Λ') (k : Fin n) : (N • γ) k ∈ X.L

theorem **exists_nsmul_stdLat_le** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') : ∃ N : ℕ, 1 ≤ N ∧ ∀ u ∈ stdLat E n, N • u ∈ Λ'

theorem **exists_nsmul_commensurable** (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') : ∃ M : ℕ, 1 ≤ M ∧ (∀ γ : Fin n → Vsp X.dim, (∀ k, γ k ∈ X.L) → M • γ ∈ tensorLat X Λ') ∧ (∀ γ ∈ tensorLat X Λ', ∀ k, (M • γ) k ∈ X.L)

theorem **tensorLat_span_integral_perfect** (X : PT E) (Λ' : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ'.FG) (hint : ∀ x ∈ Λ', ∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ', IsIntegral ℤ (herm 1 x y)) → x ∈ Λ') : (∃ (m : ℕ) (v w : Fin m → Fin n → E), (∀ i, v i ∈ Λ') ∧ (∀ i, w i ∈ Λ') ∧ (∀ u : Fin n → E, u = ∑ i, herm 1 u (w i) • v i) ∧ ∀ γ : Fin n → Vsp X.dim, γ = ∑ i, tens (contr γ (w i)) (v i)) ∧ (∀ (m : ℕ) (v w : Fin m → Fin n → E), (∀ u : Fin n → E, u = ∑ i, herm 1 u (w i) • v i) → ∀ γ : Fin n → Vsp X.dim, γ = ∑ i, tens (contr γ (w i)) (v i)) ∧ tensorLat X Λ' = Submodule.span ℤ {γ | ∃ x ∈ X.L, ∃ v ∈ Λ', γ = tens x v} ∧ (∀ (α : 𝒪 E), ∀ γ ∈ tensorLat X Λ', (fun k => act α (γ k)) ∈ tensorLat X Λ') ∧ (∀ γ ∈ tensorLat X Λ', ∀ γ' ∈ tensorLat X Λ', ∃ k : ℤ, (HC γ γ').im = k) ∧ (∀ φ : tensorLat X Λ' →+ ℤ, ∃ γ ∈ tensorLat X Λ', ∀ γ' : tensorLat X Λ', (φ γ' : ℝ) = (HC γ (γ' : Fin n → Vsp

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.OneDivisionUpstairsFreeWeight

theorem **smoothWtG_contDiffOn** (Xa : ArchInM X) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r) (bb : ℝ) : ContDiffOn ℝ ∞ (smoothWtG Xa G x0 bb) (OmF X x0)

theorem **smoothWtG_levi** (Xa : ArchInM X) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r) (bb : ℝ) {z : V (nd X)} (hz : z ∈ OmF X x0) (ξ : Fin (nd X) → ℂ) (hm : bb * leviRh X x0 z ξ ≤ leviO X x0 (phla X G) z ξ − leviDens X x0 z ξ) : 0 ≤ (leviV (nd X) (fun w => ((smoothWtG Xa G x0 bb w : ℝ) : ℂ)) z ξ).re

theorem **inAt_iffG** (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r) (bb t : ℝ) (k0 : Fin (nd X)) (hnull : volume {y | y ∈ OmF X x0 ∧ Fl.s k0 (cpt X x0 y) = 0} = 0) (F : V (nd X) → ℂ) : InAt X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (−phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (−(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (−t) else 0) bb F ↔ ∫⁻ z in OmF X x0, ENNReal.ofReal (‖F z‖ ^ 2 * (∑ k, ‖Fl.s k (cpt X x0 z)‖ ^ 2) ^ (−t) * Real.exp (−smoothWtG Xa G x0 bb z)) < ⊤

theorem **divOne_up** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r), IsOrth X x0 → ∀ bb : ℝ, (∀ z ∈ OmF X x0, ∀ ξ : Fin (nd X) → ℂ, bb * leviRh X x0 z ξ ≤ leviO X x0 (phla X G) z ξ − leviDens X x0 z ξ) → (∃ k0, ∃ z ∈ OmF X x0, Fl.s k0 (cpt X x0 z) ≠ 0) → ∀ F : V (nd X) → ℂ, DifferentiableOn ℂ F (OmF X x0) → InAt X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (−phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (−(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (−(((nd X + 1 : ℕ) : ℝ) − 1 / 2)) else 0) bb F → ∃ H : Fin (nd X) → V (nd X) → ℂ, (∀ k, DifferentiableOn ℂ (H k) (OmF X x0) ∧ InAt X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (−phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (−(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k,

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.OneDivisionUpstairsFreeWeightStatement

theorem **_root_.GlobalTheory.T.one_division_upstairs_free_weight** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r), IsOrth X x0 → ∀ bb : ℝ, (∀ z ∈ OmF X x0, ∀ ξ : Fin (nd X) → ℂ, bb * leviRh X x0 z ξ ≤ leviO X x0 (phla X G) z ξ − leviDens X x0 z ξ) → (∃ k0, ∃ z ∈ OmF X x0, Fl.s k0 (cpt X x0 z) ≠ 0) → ∀ F

: V (nd X) → ℂ, DifferentiableOn ℂ F (OmF X x0) → InAt X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-(((nd X + 1 : ℕ) : ℝ) - 1 / 2)) else 0) bb F → ∃ H : Fin (nd X) → V (nd X) → ℂ, (∀ k, DifferentiableOn ℂ (H k) (OmF X x0) ∧ InAt X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PageRelationDegThree

theorem **page_relation_deg_three** : ∀ (D : D7) (large₂ : SizeData₂ D) (j : ℕ) (ka : WeightFamilies.Wt D j) (CG AG : ℝ), 1041 ≤ CG → 3 ≤ AG → 1 ≤ Zb D large₂ → 1041 * Zb D large₂ ^ 3 ≤ WeightFamilies.PPv D large₂ CG AG * (1 + Real.log ka.size)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PageRowsOfLe

theorem **pageRows_of_le** : ∀ (D : D7) (large₂ : SizeData₂ D) (j : ℕ) (ka w : WeightFamilies.Wt D j), 0 ≤ large₂.sc.L → WeightAlgebra.DF w ≤ WeightAlgebra.DF ka → w.n ≤ ka.n → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PageTwoLeLAndSigns

theorem **page_two_le_L_and_signs** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → 2 ≤ large₂.sc.L ∧ 2 ≤ D.R.f ∧ 6 ≤ D.R.r ∧ 0 ≤ lamStar D ∧ (1 ≤ j → 1 ≤ WeightFamilies.dimY D j) ∧ (FamilyHypotheses.SupBoundPageRows D large₂ ka ↔ ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PeriodTorusConstantEqBase

theorem **dimPer_mem** {σ : E →+* ℂ} (h : σ ∈ X.Ψ) : dimPer X σ = r - X.p0

theorem **dimPer_not_mem** {σ : E →+* ℂ} (h : σ ∉ X.Ψ) : dimPer X σ = X.p0

theorem **dimPer_add** (σ : E →+* ℂ) : dimPer X σ + dimPer X (ComplexEmbedding.conjugate σ) = r

theorem **card_idx_dimPer** : Fintype.card (Idx (dimPer X)) = X.Ψ.card * r

theorem **rowFn_mem** (τ : X.Ψ) (i : Fin (dimPer X τ.1)) (a : Fin r) (ha : (a : ℕ) = X.p0 + i) : rowFn X (τ.1, i) = (τ, a)

theorem **rowFn_conj** (τ : X.Ψ) (i : Fin (dimPer X (ComplexEmbedding.conjugate τ.1))) (a : Fin r) (ha : (a : ℕ) = X.p0 - 1 - i) : rowFn X (ComplexEmbedding.conjugate τ.1, i) = (τ, a)

theorem **rowFn_bijective** : Function.Bijective (rowFn X)

theorem **rowEquiv_apply** (i : Idx (dimPer X)) : rowEquiv X i = rowFn X i

theorem **rowType_rowEquiv** (i : Idx (dimPer X)) : rowType X (rowEquiv X i) = i.1

theorem **scaledR_apply** (x : Frame X.Ψ r) (y : X.Ψ → Fin r → ℂ) (i : Idx (dimPer X)) : scaledR X x y i = (Real.sqrt (X.c (rowEquiv X i).1) : ℂ) * perR X x y (rowEquiv X i)

theorem **scaled_apply** (x : Frame X.Ψ r) (v : Fin r → E) (i : Idx (dimPer X)) : scaled X x v i = (Real.sqrt (X.c (rowEquiv X i).1) : ℂ) * per X x v (rowEquiv X i)

theorem **scaled_smul** (x : Frame X.Ψ r) (α : 𝒪 E) (v : Fin r → E) : scaled X x ((α : E) • v) = act α (scaled X x v)

theorem **Hf_scaledR** (x : Frame X.Ψ r) (y y' : X.Ψ → Fin r → ℂ) : Hf (scaledR X x y) (scaledR X x y') = ∑ τ : X.Ψ, (X.c τ : ℂ) * ∑ a, perR X x y (τ, a) * conj (perR X x y' (τ, a))

theorem **col_orth** (τ : X.Ψ) (a b : Fin r) : hermC τ.1 X.Gm (FD.col (mat x τ) a) (FD.col (mat x τ) b) = if a = b then ((sgn X.p0 a : ℝ) : ℂ) else 0

theorem **det_ne_zero** (τ : X.Ψ) : (mat x τ).det ≠ 0

theorem **mem_dom** : x ∈ X.dom

theorem **norm_perR** (y : X.Ψ → Fin r → ℂ) (τ : X.Ψ) (a : Fin r) : ‖perR X x y (τ, a)‖ = ‖hermC τ.1 X.Gm (y τ) (FD.col (mat x τ) a)‖

theorem **norm_sq_scaledR** (y : X.Ψ → Fin r → ℂ) : ‖scaledR X x y‖ ^ 2 = ∑ τ : X.Ψ, X.c τ * X.maj x τ (y τ)

theorem **scaledR_injective** : Function.Injective (scaledR X x)

theorem **exists_basis** : ∃ b : Module.Basis (Fin (2 * Fintype.card (Idx (dimPer X)))) ℝ (Vsp (dimPer X)), ∀ y, y ∈ lat b ↔ ∃ v ∈ X.Λ, scaled X x v = y

theorem **im_Hf_scaled** (u v : Fin r → E) : (Hf (scaled X x u) (scaled X x v)).im = EF X u v

theorem **mem_lat_basis** (y : Vsp (dimPer Y.toData)) : y ∈ lat (basis hx) ↔ ∃ v ∈ Y.Λ, scaled Y.toData x v = y

theorem **stable** (α : 𝒪 E) (y : Vsp (dimPer Y.toData)) (hy : y ∈ lat (basis hx)) : act α y ∈ lat (basis hx)

theorem **integral** (y z : Vsp (dimPer Y.toData)) (hy : y ∈ lat (basis hx)) (hz : z ∈ lat (basis hx)) : ∃ k : ℤ, (Hf y z).im = k

theorem **perfect** (ψ : lat (basis hx) →+ ℤ) : ∃ y ∈ lat (basis hx), ∀ z : lat (basis hx), (ψ z : ℝ) = (Hf y (z : Vsp (dimPer Y.toData))).im

theorem **coe_one_sub** : ((1 - w₀ : 𝒪 E) : E) = cc (w₀ : E)

theorem **even_im_hermitian_sub** : ∀ y ∈ lat (basis hx), ∀ z ∈ lat (basis hx), ∃ k : ℤ, (Hf y (act w₀ z)).im - (Hf y (act (1 - w₀) z)).im - (Hf y z).im = 2 * k

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PhiFDEqZeroAtLastIndex

theorem **phiFD_eq_zero_at_last_index** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j), j = D.R.m → ∀ (DF q : ℕ) (R0 : ℝ), RunOK D.host.E D.R.r DF q R0 D.R.m → ∀ σ, GoodEmb T.Q.c σ → ∀ (k : (T.U j σ).Comp), ∀ x ∈ ((T.U j σ).X k).dom, I2.phiFD ((T.U j σ).X k) (I2.th (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PointsInterpolationGenericSections

theorem **addL_zero_pts** {D : D7} {j : ℕ} (u wL : WeightFamilies.Wt D j) : addL u wL 0 = u

theorem **rdg_s_apply** {D : D7} {T : Top D} {h : LawsP T} {j : ℕ} {hj : InR D j} {n : SecNorm (recB h.toLawsCore)} {R : Rdg (recPlusN h j hj n)} {e : R.Y ≃ DomPt T j} (hR : IsRdgU h j hj n R e) {N : ℕ} {wL : WeightFamilies.Wt D j} (w : Fin N → (recB h.toLawsCore).Cusp j

wL) (t : Idx D j N → ℂ) (i : Fin (WeightFamilies.dimY D j + 1)) (y : R.Y) : R.s w t i y = ∑ k, t (i, k) * rdR h j (w k) (e y).1 (e y).2.1

theorem **mem_sp_of_rdOnto** {D : D7} {T : Top D} {h : LawsP T} {j : ℕ} {hj : InR D j} {n : SecNorm (recB h.toLawsCore)} {R : Rdg (recPlusN h j hj n)} {e : R.Y ≃ DomPt T j} (hR : IsRdgU h j hj n R e) {u : WeightFamilies.Wt D j} (hO : RdOnto h j u) (f : ∀ k : (T.U j T.σ₀).Comp, Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ) (hf : ∀ k, f k ∈ ((T.U j T.σ₀).X k).SS (laR T j u k)) : (fun y => f (e y).1 (e y).2.1) ∈ R.sp u Set.univ

theorem **points_and_interp** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _)⟩) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → (∀ q, R.s w t 0 (xs q) ≠ 0 ∧ ∀ i : Fin (WeightFamilies.dimY D j), R.s w t i.succ (xs q) = 0) ∧ ∀ v : Fin δ → ℂ, ∃ f ∈ R.sp wF Set.univ, ∀ q, f (xs q) = v q

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PointsInterpolationGenericSectionsStatement

theorem **_root_.GlobalTheory.T.points_interpolation_genericSections** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (n : SecNorm (recB h.toLawsCore)) (R : Rdg (recPlusN h j hj n)) (e : R.Y ≃ DomPt T j), IsRdgU h j hj n R e → ∀ (N : ℕ) (wL wF : WeightFamilies.Wt D j) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (Tn : ℕ) (c1 : ℝ) (A : ArchAt h j w wF Tn c1) (t : Idx D j N → ℂ) (Fl : FlatAt h j A t) (δ : ℕ) (xs : Fin δ → R.Y) (en : Fin δ ≃ Σ k : (T.U j T.σ₀).Comp, Fin (Fl.Fl k).δ), (∀ q, e (xs q) = ⟨(en q).1, ((Fl.Fl (en q).1).xs (en q).2, (Fl.Fl (en q).1).xs_mem _)⟩) → (∀ k : ℕ, RdOnto h j (addL wF wL k)) → (∀ q, R.s w t 0 (xs q) ≠ 0 ∧ ∀ i : Fin (WeightFamilies.dimY D j), R.s w t i.succ (xs q) = 0) ∧ ∀ v : Fin δ → ℂ, ∃ f ∈ R.sp wF Set.univ, ∀ q, f (xs q) = v q

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PointwiseNormCanonicalThetaEqSqThetaConstant

theorem **canMult_add** (a b : Fin g → ℝ) (m m' : Fin g ⊕ Fin g → ℤ) : canMult a b (fun i => (m + m') (.inl i)) (fun i => (m + m') (.inr i)) = canMult a b (fun i => m (.inl i)) (fun i => m (.inr i)) * canMult a b (fun i => m' (.inl i)) (fun i => m' (.inr i)) * cexp (π * I * (omegaStd m m' : ℤ))

theorem **exists_canMult_eq** (ε : (Fin g ⊕ Fin g → ℤ) → ℤ) (h1 : ∀ m, ε m = 1 ∨ ε m = -1) (hadd : ∀ m m', (ε (m + m') : ℂ) = ε m * ε m' * cexp (π * I * (omegaStd m m' : ℤ))) : ∃ a b : Fin g → ℝ, ∀ m, canMult a b (fun i => m (.inl i)) (fun i => m (.inr i)) = ε m

theorem **Hf_self'** (v : EuclideanSpace ℂ ι) : Hf v v = ((‖v‖ ^ 2 : ℝ) : ℂ)

theorem **cj_apply** (y : Vsp T.dim) (i : ι) : cj T e y i = conj (y (e i))

theorem **Hf_cj** (v w : Vsp T.dim) : Hf (cj T e v) (cj T e w) = conj (Hf v w)

theorem **pointwiseNorm_conjugateFunctionOnConjugateSpace** (y : Vsp T.dim) : T.ptN2 1 (cjFn T e F) (fun _ => y) = Y.ptN2 1 F (cj T e y)

theorem **cjFn_mem_Sec** : cjFn T e F ∈ T.Sec 1 1

theorem **l2_cjFn_eq** : T.l2 1 (cjFn T e F) = Y.l2 1 F

theorem **ratio_cj** (hF0 : Y.l2 1 F ≠ 0) {s : (Fin 1 → Vsp T.dim) → ℂ} (hs : s ∈ T.Sec 1 1) (hs0 : s ≠ 0) (y : Vsp T.dim) : T.ptN2 1 s (fun _ => y) / T.l2 1 s = Y.ptN2 1 F (cj T e y) / Y.l2 1 F

theorem **exists_sign** (X : Data E r) : ∃ σ : ℤ, (σ = 1 ∨ σ = -1) ∧ ∀ u v, EF X u v = σ * B X u v

theorem **finrank_lattice** (X : Data E r) : Module.Free ℤ X.Λ ∧ Module.Finite ℤ X.Λ ∧ Module.finrank ℤ X.Λ = 2 * (X.Ψ.card * r)

theorem **exists_isSympl** (X : DataX E r) : ∃ bas : (Fin (X.Ψ.card * r) ⊕ Fin (X.Ψ.card * r) → ℤ) ≃ₗ[ℤ] X.Λ, IsSympl X.toData bas

theorem **exists_coord** (hs : IsSympl X bas) (v : Fin r → E) : ∃ a b : Fin g → ℝ, ∀ x ∈ X.dom, hMat X bas idx x *ᵥ (per X x v ∘ idx) = (fun i => (b i : ℂ)) + tauMat X bas idx x *ᵥ fun i => (a i : ℂ)

theorem **pointwiseNorm_canonicalTheta_eq_sq_thetaConstant** (hs : IsSympl X bas) {x : Frame X.Ψ r} (ho : X.IsOrtho x) (a b a' b' : Fin g → ℝ) (z : Fin g → ℂ) (hz : hMat X bas idx x *ᵥ z = (fun i => (b' i : ℂ)) + tauMat X bas idx x *ᵥ fun i => (a' i : ℂ)) : (frameTorus idx hs ho a b).ptN2 1 (canSec X bas idx a b x) (scaleUp X idx z) = ‖thetaChar (tauMat X bas idx x) (fun i => ((a + a') i : ℂ)) (fun i => ((b + b') i : ℂ)) 0‖ ^ 2

theorem **norm_det_hMat** (hs : IsSympl X bas) {x : Frame X.Ψ r} (ho : X.IsOrtho x) : √(2 ^ g * ((tauMat X bas idx x).map Complex.im).det) = √(2 ^ g / ∏ τ, X.c τ ^ r) * ‖(hMat X bas idx x).det‖

theorem **det_hMat_weight** (x : Frame X.Ψ r) {u : Frame X.Ψ r} (hu : u ∈ Bor X.Ψ r) : (hMat X bas idx (rmul x u)).det = bchar (fun _ j => if X.p0 ≤ (j : ℕ) then 1 else -1) u * (hMat X bas idx x).det

theorem **isOrtho_adapted** : X.IsOrtho (adapted A)

theorem **exists_adapted_eq** : ∃ u ∈ Bor X.Ψ r, adapted A = rmul x u

theorem **exists_equiv** : ∃ e : X.Ψ × Fin r ≃ Idx A.T.dim, ∀ v p, conj (A.φ v (e p)) = (√(X.c p.1) : ℂ) * per X (adapted A) v p

theorem **exists_char_at** (hs : IsSympl X.toData bas) {x : Frame X.Ψ r} (A : TorusAt X.toData x) : ∃ a b : Fin g → ℝ, ∀ m, canMult a b (fun i => m (.inl i)) (fun i => m (.inr i)) = A.T.eps (A.φ (bas m))

theorem **ratio_eq_theta** (idx : Fin g ≃ X.Ψ × Fin r) (hs : IsSympl X.toData bas) {x : Frame X.Ψ r} (A : TorusAt X.toData x) {a b : Fin g → ℝ} (hab : ∀ m, canMult a b (fun i => m (.inl i)) (fun i => m (.inr i)) = A.T.eps (A.φ (bas m))) (v : Fin r → E) (a' b' : Fin g → ℝ) (hv : hMat X.toData bas idx (adapted A) *ᵥ (per X.toData (adapted A) v ∘ idx) = (fun i => (b' i : ℂ)) + tauMat X.toData bas idx (adapted A) *ᵥ fun i => (a' i : ℂ)) : A.T.ptN2 1 A.s (fun _ => A.φ v) / A.T.l2 1 A.s = √(2 ^ g / ∏ τ, X.c τ ^ r) * ‖thetaChar (tauMat X.toData bas idx (adapted A)) (fun i => ((a + a') i : ℂ)) (fun i => ((b + b') i : ℂ)) 0‖ ^ 2 * ‖(hMat X.toData bas idx (adapted A)).det‖

theorem **normalised_theta_norm_holomorphic** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) - cc (w0 : E)) • v)) - Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) → ∃ (θ : (Fin r → E) → Frame X.Ψ r → ℂ) (D : Frame X.Ψ r → ℂ) (κ : ℝ), 0 < κ ∧ (∀ v, DifferentiableOn ℂ (θ v) X.dom) ∧ DifferentiableOn ℂ D X.dom ∧ (∀ x ∈ X.dom, D x ≠ 0) ∧ ∀ x ∈ X.dom, ∀ (A : TorusAt X.toData x) (v : Fin r → E), A.T.ptN2 1 A.s (fun _ => A.φ v) / A.T.l2 1 A.s = κ * ‖θ v x‖ ^ 2 * ‖D x‖ * X.nfac (fun _ j => if X.p0 ≤ (j : ℕ) then 1 else -1) x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PointwiseNormSqPullbackTensorEqProd

theorem **l2NormSq_nonneg_gaussianTorus** (Y : GT ι) (q : ℕ) (F : EuclideanSpace ℂ ι → ℂ) : 0 ≤ Y.l2 q F

theorem **le_firstMinimum_of_period_lower_bound** (Y : GT ι) (hι : Fintype.card ι ≠ 0) {lam : ℝ} (hmin : ∀ x ∈ lat Y.b, x ≠ 0 → lam ≤ ‖x‖) : lam ≤ Y.la1

theorem **pointwiseNormSq_le_l2_of_firstMinimum** (hsup : GSupStatement) (Y : GT ι) {F : EuclideanSpace ℂ ι → ℂ} (hF : F ∈ Y.Sec 1) {lam : ℝ} (hlam : 0 < lam) (hmin : ∀ x ∈ lat Y.b, x ≠ 0 → lam ≤ ‖x‖) (v : EuclideanSpace ℂ ι) : Y.ptN2 1 F v ≤ (Real.sqrt 2 * (1 + 4 * Real.sqrt (2 * Fintype.card ι) / (Real.sqrt Real.pi * lam)) ^ Fintype.card ι) ^ 2 * Y.l2 1 F

theorem **norm_polyDeriv_le** (q : ℕ) (F : EuclideanSpace ℂ ι → ℂ) (hF : Differentiable ℂ F) (S : ℝ) (hS : ∀ v, ‖F v‖ ^ 2 * Real.exp (-(q * Real.pi) * ‖v‖ ^ 2) ≤ S) (P : MvPolynomial ι ℂ) (a : ℕ) (hP : P.IsHomogeneous a) : ‖polyDeriv P (fun i => EuclideanSpace.single i 1) F 0‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * Real.sqrt S * (q * Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2)

theorem **norm_polyDeriv_U_le** (X : PT E) (q : ℕ) {n : ℕ} (F : (Fin n → Vsp X.dim) → ℂ) (hF : Differentiable ℂ F) (S : ℝ) (hS : ∀ v, X.ptN2 q F v ≤ S) (P : MvPolynomial (Letter n X.dim) ℂ) (a : ℕ) (hP : P.IsHomogeneous a) (b : Fin n → Vsp X.dim) : ‖polyDeriv P dir (X.U q b F) 0‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * Real.sqrt S * (q * Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2)

theorem **col_mem_Sec** (X : PT E) {q : ℕ} {s : (Fin 1 → Vsp X.dim) → ℂ} (hs : s ∈ X.Sec q 1) : (fun z => s fun _ => z) ∈ (gtOf X q).Sec q

theorem **oneColumn_pointwiseNorm_eq_torus** (X : PT E) (q : ℕ) (s : (Fin 1 → Vsp X.dim) → ℂ) (x : Vsp X.dim) : (gtOf X q).ptN2 q (fun z => s fun _ => z) x = X.ptN2 q s fun _ => x

theorem **oneColumn_l2Norm_eq_torus** (X : PT E) (q : ℕ) (s : (Fin 1 → Vsp X.dim) → ℂ) : (gtOf X q).l2 q (fun z => s fun _ => z) = X.l2 q s

theorem **l2_pos_of_mem_sec_of_ne_zero** (X : PT E) {q : ℕ} (hq : 1 ≤ q) {s : (Fin 1 → Vsp X.dim) → ℂ} (hs : s ∈ X.Sec q 1) (hne : s ≠ 0) : 0 < X.l2 q s

theorem **diagonal_pointwiseNormSq_le_l2** (X : PT E) {s : (Fin 1 → Vsp X.dim) → ℂ} (hs : s ∈ X.Sec 1 1) {lam : ℝ} (hlam : 0 < lam) (hmin : ∀ x ∈ X.L, x ≠ 0 → lam ≤ ‖x‖) (x : Vsp X.dim) : X.ptN2 1 s (fun _ => x) ≤ (Real.sqrt 2 * (1 + 4 * Real.sqrt (2 * Fintype.card (Idx X.dim)) / (Real.sqrt Real.pi * lam)) ^ Fintype.card (Idx X.dim)) ^ 2 * X.l2 1 s

theorem **differentiable_pullback_tensor** (X : PT E) {n : ℕ} (π : Vsp X.dim →ₗ[ℂ] Vsp X.dim) {s : (Fin 1 → Vsp X.dim) → ℂ} (hs : Differentiable ℂ s) : Differentiable ℂ (s0 X (n

theorem **pointwiseNormSq_pullback_tensor_eq_prod** (X : PT E) {q n : ℕ} {π : Vsp X.dim →ₗ[ℂ] Vsp X.dim} (hπ : ∀ v w, Hf (π v) (π w) = q * Hf v w) (s : (Fin 1 → Vsp X.dim) → ℂ) (v : Fin n → Vsp X.dim) : X.ptN2 q (s0 X π s) v = ∏ k, X.ptN2 1 s fun _ => π (v k)

theorem **norm_polyDeriv_pullback_tensor_le** (X : PT E) {q n : ℕ} {π : Vsp X.dim →ₗ[ℂ] Vsp X.dim} (hπ : ∀ v w, Hf (π v) (π w) = q * Hf v w) {s : (Fin 1 → Vsp X.dim) → ℂ} (hs : s ∈ X.Sec 1 1) {lam : ℝ} (hlam : 0 < lam) (hmin : ∀ x ∈ X.L, x ≠ 0 → lam ≤ ‖x‖) (P : MvPolynomial (Letter n X.dim) ℂ) (a : ℕ) (hP : P.IsHomogeneous a) (b : Fin n → Vsp X.dim) : ‖polyDeriv P dir (X.U q b (s0 X π s)) 0‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * (q * Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2) * (Real.sqrt 2 * (1 + 4 * Real.sqrt (2 * Fintype.card (Idx X.dim)) / (Real.sqrt Real.pi * lam)) ^ Fintype.card (Idx X.dim)) ^ n * Real.sqrt (X.l2 1 s) ^ n

theorem **theta_value_bound** (hsup : GSupStatement) {ι : Type} [Fintype ι] (U : ℕ → EuclideanSpace ℂ ι → (EuclideanSpace ℂ ι → ℂ) → EuclideanSpace ℂ ι → ℂ) (hU : ∀ (q : ℕ) a F v, U q a F v = cexp (q * Real.pi * Hf v a - q * Real.pi * Hf a a / 2) * F (v - a)) (Y : GT ι) (F : EuclideanSpace ℂ ι → ℂ) (hF : F ∈ Y.Sec 1) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ x ∈ lat Y.b, x ≠ 0 → lam ≤ ‖x‖) {κ : Type} (s : Finset κ) (c : κ → ℂ) (hc : ∀ j ∈ s, ‖c j‖ ≤ 1) (y : κ → EuclideanSpace ℂ ι) (P : MvPolynomial ι ℂ) (a : ℕ) (hP : P.IsHomogeneous a) : ‖∑ j ∈ s, c j * polyDeriv P (fun i => EuclideanSpace.single i 1) (U 1 (y j) F) 0‖ ≤ s.card * (∑ m ∈ P.support, ‖P.coeff m‖) * (Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2) * (Real.sqrt 2 * (1 + 4 * Real.sqrt (2 * Fintype.card ι) / (Real.sqrt Real.pi * lam)) ^ Fintype.card ι) * Real.sqrt (Y.l2 1 F)

theorem **norm_thAn_le** {E : Type} [Field E] [NumberField E] [NumberField.IsCMField E] {r : ℕ} {X : ReciprocityTheory.ChartRestriction.G.FD.Data E r} {x : ReciprocityTheory.ChartRestriction.G.FD.Frame X.Ψ r} (A : TorusAt X x) (q n : ℕ) (π : Vsp A.T.dim →ₗ[ℂ] Vsp A.T.dim) (hπ : IsPi A.T q π) (b : Fin n → Vsp A.T.dim) (P : MvPolynomial (Letter n A.T.dim) ℂ) (a : ℕ) (hP : P.IsHomogeneous a) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ γ ∈ A.T.L, γ ≠ 0 → lam ≤ ‖γ‖) : ‖thAn A q n π b P‖ ≤ (∑ m ∈ P.support, ‖P.coeff m‖) * (q * Real.pi * Real.exp 1 * a) ^ ((a : ℝ) / 2) * (Real.sqrt 2 * (1 + 4 * Real.sqrt (2 * Fintype.card (Idx A.T.dim)) / (Real.sqrt Real.pi * lam)) ^ Fintype.card (Idx A.T.dim)) ^ n

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PolyDerivLevelSqEqAlpha

theorem **polyDeriv_comp_clm** {J V : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] (Q : MvPolynomial J ℂ) (u : J → V) (G : V → ℂ) (hG : ContDiff ℂ ⊤ G) (g : V →L[ℂ] V) (c : J → ℂ) (hg : ∀ l, g (u l) = c l • u l) (κ : ℂ) (hκ : ∀ m ∈ Q.support, (m.prod fun l k => c l ^ k) = κ) : I2.polyDeriv Q u (G ∘ g) 0 = κ * I2.polyDeriv Q u G 0

theorem **hf_act_act** {E : Type} [Field E] [NumberField E] {dim : (E →+* ℂ) → ℕ} (β : 𝒪 E) (M : ℂ) (h : ∀ τ : E →+* ℂ, τ (β : E) * (starRingEnd ℂ) (τ (β : E)) = M) (v w : I2.Vsp dim) : I2.Hf (I2.act β v) (I2.act β w) = M * I2.Hf v w

theorem **translate_eq_comp** {E : Type} [Field E] [NumberField E] [IsCMField E] (X : I2.PT E) (α β : 𝒪 E) (M n : ℕ) (h : ∀ τ : E →+* ℂ, τ (β : E) * (starRingEnd ℂ) (τ (β : E)) = M) (h' : ∀ τ : E →+* ℂ, τ (α : E) * τ (β : E) = M) (s : (Fin 1 → I2.Vsp X.dim) → ℂ) (a v : Fin n → I2.Vsp X.dim) : X.U (M ^ 2) a (I2.s0 X ((M : ℂ) • LinearMap.id) s) v = X.U M (fun b => I2.act β (a b)) (I2.s0 X (vspActL α) s) (fun b => I2.act β (v b))

theorem **differentiable_translate** {E : Type} [Field E] [NumberField E] (X : I2.PT E) (q n : ℕ) (p : I2.Vsp X.dim →ₗ[ℂ] I2.Vsp X.dim) (s : (Fin 1 → I2.Vsp X.dim) → ℂ) (hs : Differentiable ℂ s) (a : Fin n → I2.Vsp X.dim) : Differentiable ℂ (X.U q a (I2.s0 X p s))

theorem **polyDeriv_levelSq_eq_alpha** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (X : I2.PT E) (α β : 𝒪 E) (M n : ℕ), (β : E) = IsCMField.complexConj E (α : E) → α * β = (M : 𝒪 E) → ∀ s : (Fin 1 → I2.Vsp X.dim) → ℂ, Differentiable ℂ s → ∀ (a : Fin n → I2.Vsp X.dim) (Io : Type) (P : MvPolynomial Io ℂ) (u : Io → Fin n → I2.Vsp X.dim) (sc : Io → ℂ) (κ : ℂ), (∀ ℓ b, I2.act β (u ℓ b) = sc ℓ • u ℓ b) → (∀ m ∈ P.support, (m.prod fun ℓ e => sc ℓ ^ e) = κ) → I2.polyDeriv P u (X.U (M ^ 2) a (I2.s0 X ((M : ℂ) • LinearMap.id) s)) 0 = κ * I2.polyDeriv P u (X.U M (fun b => I2.act β (a b)) (I2.s0 X (vspActL α) s)) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PolyDerivProdColumnFunctions

theorem **filter_col_toMultiset** {T : Type} [Fintype T] {r : T → ℕ} {n : ℕ} (m : Formalism.L2.Pj.Letter T r n →₀ ℕ) (b : Fin n) : (Finsupp.toMultiset m).filter (fun ℓ => ℓ.2.2 = b) = (Finsupp.toMultiset (Finsupp.equivFunOnFinite.symm fun ℓ : (Σ τ : T, Fin (r τ)) => m ⟨ℓ.1, ℓ.2, b⟩)).map (fun ℓ : (Σ τ : T, Fin (r τ)) => (⟨ℓ.1, ℓ.2, b⟩ : Formalism.L2.Pj.Letter T r n))

theorem **polyDeriv_prod_cols** : ∀ (V : Type) [NormedAddCommGroup V] [NormedSpace ℂ V] [FiniteDimensional ℂ V] (T : Type) [Fintype T] (r : T → ℕ) (n : ℕ) (u : (Σ τ : T, Fin (r τ)) → V) (θ : Fin n → V → ℂ), (∀ b, Differentiable ℂ (θ b)) → ∀ P : MvPolynomial (Formalism.L2.Pj.Letter T r n) ℂ, I2.polyDeriv P (fun ℓ => (Pi.single ℓ.2.2 (u ⟨ℓ.1, ℓ.2.1⟩) : Fin n → V)) (fun z => ∏ b, θ b (z b)) 0 = ∑ m ∈ P.support, P.coeff m * ∏ b, I2.polyDeriv (MvPolynomial.monomial (Finsupp.equivFunOnFinite.symm fun ℓ : (Σ τ : T, Fin (r τ)) => m ⟨ℓ.1, ℓ.2, b⟩) (1 : ℂ)) u (θ b) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PolyDerivProdColumnFunctionsStatement

theorem **_root_.GlobalTheory.T.polyDeriv_prod_column_functions** : ∀ (V : Type) [NormedAddCommGroup V] [NormedSpace ℂ V] [FiniteDimensional ℂ V] (T : Type) [Fintype T] (r : T → ℕ) (n : ℕ) (u : (Σ τ : T, Fin (r τ)) → V) (θ : Fin n → V → ℂ), (∀ b,

Differentiable ℂ (θ b)) → ∀ P : MvPolynomial (Formalism.L2.Pj.Letter T r n) ℂ, I2.polyDeriv P (fun ℓ => (Pi.single ℓ.2.2 (u (ℓ.1, ℓ.2.1)) : Fin n → V)) (fun z => ∏ b, θ b (z b)) 0 = ∑ m ∈ P.support, P.coeff m * ∏ b, I2.polyDeriv (MvPolynomial.monomial (Finsupp.equivFunOnFinite.symm fun ℓ : (Σ τ : T, Fin (r τ)) => m (ℓ.1, ℓ.2, b)) (1 : ℂ)) u (θ b) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PolyDerivProdFockEqTranslate

theorem **polyDeriv_comp_linear_equiv** {σ V W : Type} [NormedAddCommGroup V] [NormedSpace ℂ V] [NormedAddCommGroup W] [NormedSpace ℂ W] (e : V ≃L[ℂ] W) (P : MvPolynomial σ ℂ) (u : σ → V) (G : W → ℂ) (x : V) : I2.polyDeriv P u (fun z => G (e z)) x = I2.polyDeriv P (fun ℓ => e (u ℓ)) G (e x)

theorem **lintor_hf_neg_right** {J : Type} [Fintype J] (v w : EuclideanSpace ℂ J) : I2.Hf v (-w) = -I2.Hf v w

theorem **lintor_hf_neg_left** {J : Type} [Fintype J] (v w : EuclideanSpace ℂ J) : I2.Hf (-v) w = -I2.Hf v w

theorem **polyDeriv_prod_fock_eq_translate** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (X : I2.PT E) (g : ℕ) (ψ : (Fin g → ℂ) →ₗ[ℂ] I2.Vsp X.dim) (Hm : Matrix (Fin g) (Fin g) ℂ) (θ : (Fin g → ℂ) → ℂ), (∀ z w, I2.Hf (ψ z) (ψ w) = hermForm Hm z w) → Differentiable ℂ θ → ∀ (n N : ℕ) (a : Fin n → Fin g → ℂ) (Io : Type) (P : MvPolynomial Io ℂ) (u : Io → Fin n → Fin g → ℂ), I2.polyDeriv P u (fun z => ∏ b, fock Hm ((N : ℤ) ^ 2) (a b) (fun w => θ ((N : ℂ) • w)) (z b)) 0 = I2.polyDeriv P (fun ℓ b => ψ (u ℓ b)) (X.U (N ^ 2) (fun b => -ψ (a b)) (I2.s0 X ((N : ℂ) • LinearMap.id) fun v => θ (ψ.symm (v 0)))) 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PolyDerivVanishesOfTranslationInvariance

theorem **apolar_monomial** (m : σ →₀ ℕ) (c : ℂ) (F : MvPowerSeries σ ℂ) : apolar (MvPolynomial.monomial m c) F = c * ((∏ i, (m i).factorial : ℕ) : ℂ) * MvPowerSeries.coeff m F

theorem **apolar_apply** (P : MvPolynomial σ ℂ) (F : MvPowerSeries σ ℂ) : apolar P F = ∑ m ∈ P.support, P.coeff m * ((∏ i, (m i).factorial : ℕ) : ℂ) * MvPowerSeries.coeff m F

theorem **prod_factorial_add_single** [DecidableEq σ] (m : σ →₀ ℕ) (i : σ) : (∏ j, ((m + Finsupp.single i 1 : σ →₀ ℕ) j).factorial : ℕ) = (m i + 1) * ∏ j, (m j).factorial

theorem **apolar_X_mul** (i : σ) (P : MvPolynomial σ ℂ) (F : MvPowerSeries σ ℂ) : apolar (MvPolynomial.X i * P) F = apolar P (MvPowerSeries.pderiv ℂ i F)

theorem **apolar_mul_X** (i : σ) (P : MvPolynomial σ ℂ) (F : MvPowerSeries σ ℂ) : apolar P (MvPowerSeries.X i * F) = apolar (MvPolynomial.pderiv i P) F

theorem **polyDeriv_eq_apolar** (P : MvPolynomial σ ℂ) (u : σ → W) (f : W → ℂ) : polyDeriv P u f 0 = apolar P (taylorDir u f)

theorem **apolar_quad_mul** (P : MvPolynomial σ ℂ) (β : σ → σ → ℂ) (hP : ∑ s, ∑ t, β s t • MvPolynomial.pderiv s (MvPolynomial.pderiv t P) = 0) (F : MvPowerSeries σ ℂ) : apolar P ((∑ s, ∑ t, β s t • (MvPowerSeries.X t * MvPowerSeries.X s)) * F) = 0

theorem **apolar_eq_zero_of_mem_span** {ι : Type*} (v : ι → σ → ℂ) (F : MvPowerSeries σ ℂ) (hF : ∀ k, ∑ s, v k s • MvPowerSeries.pderiv ℂ s F = 0) {P : MvPolynomial σ ℂ} (hP : P ∈ Ideal.span (Set.range fun k => ∑ s, v k s • (MvPolynomial.X s : MvPolynomial σ ℂ))) : apolar P F = 0

theorem **taylor_quad** [DecidableEq σ] (β : σ → σ → ℂ) : taylor (fun z : σ → ℂ => ∑ s, ∑ t, β s t * z s * z t) = ∑ s, ∑ t, β s t • (MvPowerSeries.X t * MvPowerSeries.X s)

theorem **polyDeriv_exp_mul_dir** (P : MvPolynomial σ ℂ) (β : σ → σ → ℂ) (hP : ∑ s, ∑ t, β s t • MvPolynomial.pderiv s (MvPolynomial.pderiv t P) = 0) (u : σ → W) {B G : W → ℂ} (hBa : AnalyticAt ℂ B 0) (hB : ∀ z : σ → ℂ, B (∑ i, z i • u i) = ∑ s, ∑ t, β s t * z s * z t) (hG : AnalyticAt ℂ G 0) : polyDeriv P u (fun w => Complex.exp (B w) * G w) 0 = polyDeriv P u G 0

theorem **analyticAt_quad** (β : σ → σ → ℂ) (x : σ → ℂ) : AnalyticAt ℂ (fun z : σ → ℂ => ∑ s, ∑ t, β s t * z s * z t) x

theorem **polyDeriv_exp_mul** [DecidableEq σ] (P : MvPolynomial σ ℂ) (β : σ → σ → ℂ) (hP : ∑ s, ∑ t, β s t • MvPolynomial.pderiv s (MvPolynomial.pderiv t P) = 0) {G : (σ → ℂ) → ℂ} (hG : AnalyticAt ℂ G 0) : polyDeriv P (fun i => (Pi.single i 1 : σ → ℂ)) (fun z => Complex.exp (∑ s, ∑ t, β s t * z s * z t) * G z) 0 = polyDeriv P (fun i => (Pi.single i 1 : σ → ℂ)) G 0

theorem **polyDeriv_eq_zero_of_mem_span** {ι : Type*} (v : ι → σ → ℂ) (u : σ → W) {G : W → ℂ} (hG : AnalyticAt ℂ G 0) (hv : ∀ k, ∀ᶠ w in 𝓝 (0 : W), fderiv ℂ G w (∑ s, v k s • u s) = 0) {P : MvPolynomial σ ℂ} (hP : P ∈ Ideal.span (Set.range fun k => ∑ s, v k s • (MvPolynomial.X s : MvPolynomial σ ℂ))) : polyDeriv P u G 0 = 0

theorem **eventually_fderiv_eq_zero_of_invariant** {G : W → ℂ} {x : W} (hG : AnalyticAt ℂ G x) {v : W} (hv : ∀ (w : W) (s : ℂ), G (w + s • v) = G w) : ∀ᶠ w in 𝓝 x, fderiv ℂ G w v = 0

theorem **sum_balCoeff_smul** [DecidableEq T] [DecidableEq κ] {M : Type*} [AddCommMonoid M] [Module ℂ M] (γ : T → Fin p → Fin q → ℂ) (f : Var T p q κ → Var T p q κ → M) : ∑ s, ∑ s', balCoeff γ s s' • f s s' = ∑ t, ∑ a, ∑ b, ∑ k, γ t a b • f (xVar t a k) (yVar t b k)

theorem **quad_balCoeff** [DecidableEq T] [DecidableEq κ] (γ : T → Fin p → Fin q → ℂ) (z : Var T p q κ → ℂ) : ∑ s, ∑ s', balCoeff γ s s' * z s * z s' = ∑ t, ∑ a, ∑ b, ∑ k, γ t a b * z (xVar t a k) * z (yVar t b k)

theorem **balCoeff_pderiv_eq_zero** [DecidableEq T] [DecidableEq κ] {P : Poly ℂ T p q κ} (hP : IsHarmonic P) (γ : T → Fin p → Fin q → ℂ) : ∑ s, ∑ s', balCoeff γ s s' • MvPolynomial.pderiv s (MvPolynomial.pderiv s' P) = 0

theorem **polyDeriv_exp_balanced_mul_dir** {P : Poly ℂ T p q κ} (hP : IsHarmonic P) (γ : T → Fin p → Fin q → ℂ) (u : Var T p q κ → W) {B G : W → ℂ} (hBa : AnalyticAt ℂ B 0) (hB : ∀ z : Var T p q κ → ℂ, B (∑ i, z i • u i) = ∑ t, ∑ a, ∑ b, ∑ k, γ t a b * z (xVar t a k) * z (yVar t b k)) (hG : AnalyticAt ℂ G 0) : polyDeriv P u (fun w => Complex.exp (B w) * G w) 0 = polyDeriv P u G 0

theorem **det_rowForm_mem** {ι : Type*} [Fintype ι] [DecidableEq ι] (t : T) (r : ι → Fin p ⊕ Fin q) (ε : ι → κ → ℂ) {v : ι → ℂ} (hv : v ≠ 0) {I : Ideal (Poly ℂ T p q κ)} (hI : ∀ k, ∑ a, v a • MvPolynomial.X (t, r a, k) ∈ I) : (Matrix.of fun a b => (rowForm t (r a) (ε b) : Poly ℂ T p q κ)).det ∈ I

theorem **sum_lead_smul** {M : Type*} [AddCommMonoid M] [Module ℂ M] {m : ℕ} (i : Fin m) {v : Fin m → ℂ} (hv : ∀ a, i < a → v a = 0) (f : Fin m → M) : ∑ a : Fin (i.val + 1), v (lead i a) • f (lead i a) = ∑ a, v a • f a

theorem **lead_ne_zero** {m : ℕ} (i : Fin m) {v : Fin m → ℂ} (hv0 : v ≠ 0) (hv : ∀ a, i < a → v a = 0) : (fun a : Fin (i.val + 1) => v (lead i a)) ≠ 0

theorem **rowBlockLeadingMinorProduct_mem_ideal_first** {c : T → Fin p → ℕ} (c' : T → Fin q → ℕ) {t : T} {i : Fin p} (hc : 1 ≤ c t i) {v : Fin p → ℂ} (hv0 : v ≠ 0) (hv : ∀ a, i < a → v a = 0) {I : Ideal (Poly ℂ T p q κ)} (hI : ∀ k, ∑ a, v a • MvPolynomial.X (xVar t a k) ∈ I) (ε : T → Fin p → κ → ℂ) (φ : T → Fin q → κ → ℂ) : minorProd c c' ε φ ∈ I

theorem **minorProd_mem_y** (c : T → Fin p → ℕ) {c' : T → Fin q → ℕ} {t : T} {i : Fin q} (hc : 1 ≤ c' t i) {v : Fin q → ℂ} (hv0 : v ≠ 0) (hv : ∀ a, i < a → v a = 0) {I : Ideal (Poly ℂ T p q κ)} (hI : ∀ k, ∑ a, v a • MvPolynomial.X (yVar t a k) ∈ I) (ε : T → Fin p → κ → ℂ) (φ : T → Fin q → κ → ℂ) : minorProd c c' ε φ ∈ I

theorem **minorProd_mem_minorSpan** (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → ℂ) (φ : T → Fin q → κ → ℂ) : minorProd c c' ε φ ∈ minorSpan κ c c'

theorem **span_le_minorSpan** {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {S : Set (Poly ℂ T p q κ)} (hS : ∀ Q ∈ S, ∃ ε φ, Q = minorProd c c' ε φ) : Submodule.span ℂ S ≤ minorSpan κ c c'

theorem **orbitSpan_le_minorSpan** {n : ℕ} (h : p + q ≤ n) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) : orbitSpan ℂ h c c' ≤ minorSpan (Fin n) c c'

theorem **polyDeriv_eq_zero_of_mem_span_comb** {σ ι α : Type*} [Fintype σ] [Fintype α] (v : α → ℂ) (l : ι → α → σ) (u : σ → W) {G : W → ℂ} (hG : AnalyticAt ℂ G 0) (hv : ∀ k, ∀ᶠ w in 𝒩 (0 : W), fderiv ℂ G w (∑ a, v a • u (l k a)) = 0) {P : MvPolynomial σ ℂ} (hP : P ∈ Ideal.span (Set.range fun k => ∑ a, v a • (MvPolynomial.X (l k a) : MvPolynomial σ ℂ))) : polyDeriv P u G 0 = 0

theorem **polyDeriv_eq_zero_x_of_fderiv** {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {P : Poly ℂ T p q κ} (hP : P ∈ minorSpan κ c c') {t : T} {i : Fin p} (hc : 1 ≤ c t i) {v : Fin p → ℂ} (hv0 : v ≠ 0) (hv : ∀ a, i < a → v a = 0) (u : Var T p q κ → W) {G : W → ℂ} (hG : AnalyticAt ℂ G 0) (hd : ∀ k, ∀ᶠ w in 𝒩 (0 : W), fderiv ℂ G w (∑ a, v a • u (xVar t a k)) = 0) : polyDeriv P u G 0 = 0

theorem **polyDeriv_eq_zero_y_of_fderiv** {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {P : Poly ℂ T p q κ} (hP : P ∈ minorSpan κ c c') {t : T} {i : Fin q} (hc : 1 ≤ c' t i) {v : Fin q → ℂ} (hv0 : v ≠ 0) (hv : ∀ a, i < a → v a = 0) (u : Var T p q κ → W) {G : W → ℂ} (hG : AnalyticAt ℂ G 0) (hd : ∀ k, ∀ᶠ w in 𝒩 (0 : W), fderiv ℂ G w (∑ a, v a • u (yVar t a k)) = 0) : polyDeriv P u G 0 = 0

theorem **eq_zero_of_last_lt** {m : ℕ} {i : Fin m} (hi : i.val + 1 = m) (v : Fin m → ℂ) (a : Fin m) (h : i < a) : v a = 0

theorem **polyDeriv_eq_zero_x_dir** {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {P : Poly ℂ T p q κ} (hP : P ∈ minorSpan κ c c') {t : T} {i : Fin p} (hi : i.val + 1 = p) (hc : 1 ≤ c t i) {v : Fin p → ℂ} (hv0 : v ≠ 0) (u : Var T p q κ → W) {G : W → ℂ} (hG : AnalyticAt ℂ G 0) (hinv : ∀ k (w : W) (s : ℂ), G (w + s • ∑ a, v a • u (xVar t a k)) = G w) : polyDeriv P u G 0 = 0

theorem **polyDeriv_eq_zero_y_dir** {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {P : Poly ℂ T p q κ} (hP : P ∈ minorSpan κ c c') {t : T} {i : Fin q} (hi : i.val + 1 = q) (hc : 1 ≤ c' t i) {v : Fin q → ℂ} (hv0 : v ≠ 0) (u : Var T p q κ → W) {G : W → ℂ} (hG : AnalyticAt ℂ G 0) (hinv : ∀ k (w : W) (s : ℂ), G (w + s • ∑ a, v a • u (yVar t a k)) = G w) : polyDeriv P u G 0 = 0

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PrincipalDatumFgPerfectTracePairing

theorem **cc_eq_host_conj** (D : D7) (x : D.host.E) : ReciprocityTheory.ChartRestriction.G.FD.cc x = D.host.conj x

theorem **principalDatum_trace_perfect** : ∀ D : D7, ∃ AA : Submodule (𝒪 D.host.E) D.host.E, AA.FG ∧ AA ≠ ⊥ ∧ ∀ x : D.host.E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ D.host.E (D.host.xi0 * x * a * ReciprocityTheory.ChartRestriction.G.FD.cc b) = z) ↔ IsIntegral ℤ x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PrincipalDatumFgPerfectTracePairingStatement

theorem **_root_.GlobalTheory.T.principalDatum_fg_perfect_tracePairing** : ∀ D : D7, ∃ AA : Submodule (𝒪 D.host.E) D.host.E, AA.FG ∧ AA ≠ ⊥ ∧ ∀ x : D.host.E, (∀ a ∈ AA, ∀ b ∈ AA, ∃ z : ℤ, Algebra.trace ℚ D.host.E (D.host.xi0 * x * a * ReciprocityTheory.ChartRestriction.G.FD.cc b) = z) ↔ IsIntegral ℤ x

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ProdLeadMinorDiagonal

theorem **BiInvariant.transpose** {f : Matrix (Fin p) (Fin p) K → K} (h : BiInvariant f) : BiInvariant fun M => f Mᵀ

theorem **leadMinor_transpose** (i : Fin p) (M : Matrix (Fin p) (Fin p) K) : leadMinor i Mᵀ = leadMinor i M

theorem **leadMinor_rowAdd** (i a b : Fin p) (hab : a < b) (t : K) (M : Matrix (Fin p) (Fin p) K) : leadMinor i (rowAdd a b t M) = leadMinor i M

theorem **biInvariant_leadMinor** (i : Fin p) : BiInvariant (K

theorem **BiInvariant.what₂** {f : Matrix (Fin p) (Fin p) K → K} (h : BiInvariant f) (n : Fin p) (t : Fin p → K) (M : Matrix (Fin p) (Fin p) K) (S : Finset (Fin p)) (hS : ∀ b ∈ S, n < b) : f (Matrix.of fun b k => M b k + if b ∈ S then t b * M n k else 0) = f M

theorem **exists_clear_col** (n : Fin p) (M : Matrix (Fin p) (Fin p) K) (hn : M n n ≠ 0) : ∃ M' : Matrix (Fin p) (Fin p) K, (∀ f, BiInvariant f → f M' = f M) ∧ (∀ b, n < b → M' b n = 0) ∧ (∀ b k, b ≤ n → M' b k = M b k) ∧ (∀ b k, M n k = 0 → M' b k = M b k)

theorem **exists_cleared** (M : Matrix (Fin p) (Fin p) K) (hM : ∀ i, leadMinor i M ≠ 0) (n : ℕ) : ∃ M' : Matrix (Fin p) (Fin p) K, (∀ f, BiInvariant f → f M' = f M) ∧ ∀ i j : Fin p, i ≠ j → (i.val < n ∨ j.val < n) → M' i j = 0

theorem **exists_diagonal** (M : Matrix (Fin p) (Fin p) K) (hM : ∀ i, leadMinor i M ≠ 0) : ∃ d : Fin p → K, ∀ f, BiInvariant f → f (diagonal d) = f M

theorem **leadMinor_diagonal** (i : Fin p) (d : Fin p → K) : leadMinor i (diagonal d) = ∏ a ∈ Finset.Iic i, d a

theorem **prod_leadMinor_diagonal** (d : Fin p → K) (c : Fin p → ℕ) : ∏ i, leadMinor i (diagonal d) ^ c i = ∏ a, d a ^ ∑ i ∈ Finset.Ici a, c i

theorem **BiInvariant.eq_mul_prod** {f : Matrix (Fin p) (Fin p) K → K} (hf : BiInvariant f) (c : Fin p → ℕ) (hd : ∀ d, f (diagonal d) = f 1 * ∏ a, d a ^ ∑ i ∈ Finset.Ici a, c i) (M : Matrix (Fin p) (Fin p) K) (hM : ∀ i, leadMinor i M ≠ 0) : f M = f 1 * ∏ i, leadMinor i M ^ c i

theorem **setBlock_apply** {z : Fin p × Fin p → σ} (hz : Function.Injective z) (x : σ → K) (M : Matrix (Fin p) (Fin p) K) (w : Fin p × Fin p) : setBlock z x M (z w) = M w.1 w.2

theorem **setBlock_of_not** {z : Fin p × Fin p → σ} (x : σ → K) (M : Matrix (Fin p) (Fin p) K) {s : σ} (hs : ¬ ∃ w, z w = s) : setBlock z x M s = x s

theorem **eval_bind₁'** (y : σ → K) (g : σ → MvPolynomial σ K) (P : MvPolynomial σ K) : eval y (bind₁ g P) = eval (fun s => eval y (g s)) P

theorem **biInvariant_eval** {z : Fin p × Fin p → σ} (hz : Function.Injective z) (P : MvPolynomial σ K) (hr : ∀ a b : Fin p, a < b → ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = z (b, k) then C t * X (z (a, k)) else 0) P = P) (hc : ∀ k l : Fin p, k < l → ∀ t : K, bind₁ (fun s => X s + ∑ a, if s = z (a, l) then C t * X (z (a, k)) else 0) P = P) (x : σ → K) : BiInvariant fun M => eval (setBlock z x M) P

theorem **eval_setBlock_diagonal** {z : Fin p × Fin p → σ} (hz : Function.Injective z) (P : MvPolynomial σ K) (l : Fin p → ℕ) (hrow : ∀ a, ∀ m ∈ P.support, ∑ k, m (z (a, k)) = l a) (x : σ → K) (d : Fin p → K) : eval (setBlock z x (diagonal d)) P = eval (setBlock z x 1) P * ∏ a, d a ^ l a

theorem **eq_mul_prod_minors** {K σ : Type*} [Field K] [CharZero K] [DecidableEq σ] {p : ℕ} (z : Fin p × Fin p → σ) (hz : Function.Injective z) (P : MvPolynomial σ K) (c : Fin p → ℕ) (hrow : ∀ a, ∀ m ∈ P.support, ∑ k, m (z (a, k)) = ∑ i ∈ Finset.Ici a, c i) (hr : ∀ a b : Fin p, a < b → ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = z (b, k) then C t * X (z (a, k)) else 0) P = P) (hc : ∀ k l : Fin p, k < l → ∀ t : K, bind₁ (fun s => X s + ∑ a, if s = z (a, l) then C t * X (z (a, k)) else 0) P = P) : ∃ Q : MvPolynomial σ K, (∀ w, z w ∉ Q.vars) ∧ P = Q * ∏ i : Fin p, (Matrix.of fun a b : Fin (i.val + 1) => (X (z (Fin.castLE i.isLt a, Fin.castLE i.isLt b)) : MvPolynomial σ K)).det ^ c i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ProjAEqSumReps

theorem **projA_eq_sum_reps** {E : Type} [Field E] [NumberField E] [IsCMField E] {r g n : ℕ} {X : Data E r} {bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ} (idx : Fin g ≃ X.Ψ × Fin r) (hs : IsSympl X bas) (hn : Even n) {w : E} {A A' : AddSubgroup (Matrix (Fin n) (Fin r) E)} (hA : Good X w A) (hA' : Good X w A') {v : TF n g} (hv : IsVac X bas w A' v) {R : Finset (Matrix (Fin n) (Fin r) E)} (hR : ∀ y ∈ R, y ∈ A) (hR' : ∀ y ∈ A, ∃! y' ∈ R, y − y' ∈ A') : projA X bas w A v = ((R.card : ℂ))⁻¹ • ∑ y ∈ R, epsW X w y • heisG X bas y v

theorem **vac_proj_mean** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ Λ ∈ Vset E n 1, ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∃ R : Finset (Matrix (Fin n) (Fin r) E), (∀ y ∈ R, y ∈ tens Λ L) ∧ (∀ y ∈ tens Λ L, ∃! y', y' ∈ R ∧ ∀ b, y b − y' b ∈ L) ∧ R.card = ((stdL E n).toAddSubgroup.relIndex Λ.toAddSubgroup) ^ r ∧ 0 < R.card ∧ projA X bas (w : E) (tens Λ L) v = ((R.card : ℂ))⁻¹ • ∑ y ∈ R, epsW X (w : E) y • heisG X bas y v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.PushLettersIntCastOfIsHomogeneous

theorem **coeff_basisComponent** (b : Module.Basis ι R S) (i : ι) (P : MvPolynomial σ S) (m : σ →₀ ℕ) : coeff m (basisComponent b i P) = b.repr (coeff m P) i

theorem **exists_eq_sum_basisComponent** (b : Module.Basis ι R S) (P : MvPolynomial σ S) : ∃ J : Finset ι, P = ∑ i ∈ J, b i • MvPolynomial.map (algebraMap R S) (basisComponent b i P)

theorem **basisComponent_sum** (b : Module.Basis ι R S) (J : Finset ι) (Q : ι → MvPolynomial σ R) {i : ι} (hi : i ∈ J) : basisComponent b i (∑ j ∈ J, b j • MvPolynomial.map (algebraMap R S) (Q j)) = Q i

theorem **basisComponent_mem_rowHighest** (b : Module.Basis ι R S) (i : ι) {P : Poly S T p q κ} (hP : P ∈ rowHighest S c c') : basisComponent b i P ∈ rowHighest R c c'

theorem **span_map_rowHighest** [Module.Free R S] : Submodule.span S (MvPolynomial.map (algebraMap R S) '' (rowHighest R (κ

theorem **aeval_C_mul_X_of_isHomogeneous** {σ : Type*} (r : R) {P : MvPolynomial σ R} {d : ℕ} (hP : P.IsHomogeneous d) : aeval (fun v => (C r * X v : MvPolynomial σ R)) P = r ^ d • P

theorem **ccHom_apply** (x : E) : ccHom E x = cc x

theorem **herm_one_eq_sum** (x y : Fin n → E) : herm 1 x y = ∑ k, x k * cc (y k)

theorem **mem_Ha** : P ∈ Ha τ c c' Λ ↔ P ∈ rowHighest K c c' ∧ P ∈ symO p q τ Λ

theorem **xForm_mem_symO** (t : T) (a : Fin p) {v : Fin n → E} (hv : v ∈ Λ) : (xForm t a fun k => τ t (v k)) ∈ symO p q τ Λ

theorem **yForm_mem_symO** (t : T) (b : Fin q) {v : Fin n → E} (hv : v ∈ Λ) : (yForm t b fun k => τ t (cc (v k))) ∈ symO p q τ Λ

theorem **symO_le_iff** {A : Subalgebra (𝒪 K) (Poly K T p q (Fin n))} : symO p q τ Λ ≤ A ↔ (∀ t a, ∀ v ∈ Λ, (xForm t a fun k => τ t (v k)) ∈ A) ∧ ∀ t b, ∀ v ∈ Λ, (yForm t b fun k => τ t (cc (v k))) ∈ A

theorem **symO_mono** (h : Λ ≤ Λ') : symO p q τ Λ ≤ symO p q τ Λ'

theorem **Ha_mono** (h : Λ ≤ Λ') : Ha τ c c' Λ ≤ Ha τ c c' Λ'

theorem **det_mem_subalgebra** {R A ι : Type*} [CommRing R] [CommRing A] [Algebra R A] [Fintype ι] [DecidableEq ι] (B : Subalgebra R A) {M : Matrix ι ι A} (h : ∀ i j, M i j ∈ B) : M.det ∈ B

theorem **minorProd_mem_symO** [Fintype T] {e : T → Fin p → Fin n → E} {f : T → Fin q → Fin n → E} (he : ∀ t a, e t a ∈ Λ) (hf : ∀ t b, f t b ∈ Λ) : minorProd c c' (fun t a k => τ t (e t a k)) (fun t b k => τ t (cc (f t b k))) ∈ symO p q τ Λ

theorem **minorProd_mem_Ha** [Fintype T] {e : T → Fin p → Fin n → E} {f : T → Fin q → Fin n → E} (he : ∀ t a, e t a ∈ Λ) (hf : ∀ t b, f t b ∈ Λ) (h : ∀ t a b, herm 1 (e t a) (f t b) = 0) : minorProd c c' (fun t a k => τ t (e t a k)) (fun t b k => τ t (cc (f t b k))) ∈ Ha τ c c' Λ

theorem **minorProd_conj_mem_Ha** [Fintype T] (hΛ : ∀ v ∈ Λ, (fun k => cc (v k)) ∈ Λ') {e : T → Fin p → Fin n → E} {f : T → Fin q → Fin n → E} (he : ∀ t a, e t a ∈ Λ) (hf : ∀ t b, f t b ∈ Λ) (h : ∀ t a b, herm 1 (f t b) (e t a) = 0) : minorProd c c' (fun t a k => τ t (cc (e t a k))) (fun t b k => τ t (f t b k)) ∈ Ha τ c c' Λ'

theorem **pushLetters_xForm** (A : Matrix (Fin n) (Fin n) E) (t : T) (a : Fin p) (v : Fin n → E) : pushLetters p q τ A (xForm t a fun k => τ t (v k)) = xForm t a fun k => τ t ((A *ᵥ v) k)

theorem **pushLetters_yForm** (A : Matrix (Fin n) (Fin n) E) (t : T) (b : Fin q) (v : Fin n → E) : pushLetters p q τ A (yForm t b fun k => τ t (cc (v k))) = yForm t b fun k => τ t (cc ((A *ᵥ v) k))

theorem **eq_pushLetters** (A : Matrix (Fin n) (Fin n) E) (F : Poly K T p q (Fin n) →ₐ[K] Poly K T p q (Fin n)) (hx : ∀ t a (v : Fin n → E), F (xForm t a fun k => τ t (v k)) = xForm t a fun k => τ t ((A *ᵥ v) k)) (hy : ∀ t b (v : Fin n → E), F (yForm t b fun k => τ t (cc (v k))) = yForm t b fun k => τ t (cc ((A *ᵥ v) k))) : F = pushLetters p q τ A

theorem **pushLetters_mul** (A B : Matrix (Fin n) (Fin n) E) : pushLetters p q τ (A * B) = (pushLetters p q τ A).comp (pushLetters p q τ B)

theorem **map_pushLetters_symO** (A : Matrix (Fin n) (Fin n) E) (Λ : Submodule (𝒪 E) (Fin n → E)) : (symO p q τ Λ).map ((pushLetters p q τ A).restrictScalars (𝒪 K)) = symO p q τ (Λ.map (A.mulVecLin.restrictScalars (𝒪 E)))

theorem **pushLetters_mem_symO** {A : Matrix (Fin n) (Fin n) E} (h : ∀ v ∈ Λ, A *ᵥ v ∈ Λ') (hP : P ∈ symO p q τ Λ) : pushLetters p q τ A P ∈ symO p q τ Λ'

theorem **pushLetters_intCast_of_isHomogeneous** (M : ℤ) {d : ℕ} (hP : P.IsHomogeneous d) : pushLetters p q τ ((M : E) • 1) P = (M : 𝒪 K) ^ d • P

theorem **pow_smul_mem_symO** {M : ℤ} (h : ∀ v ∈ Λ, M • v ∈ Λ') {d : ℕ} (hd : P.IsHomogeneous d) (hP : P ∈ symO p q τ Λ) : (M : 𝒪 K) ^ d • P ∈ symO p q τ Λ'

theorem **pow_smul_mem_Ha** [Fintype T] {M : ℤ} (h : ∀ v ∈ Λ, M • v ∈ Λ') (hP : P ∈ Ha τ c c' Λ) : (M : 𝒪 K) ^ typeDegree c c' • P ∈ Ha τ c c' Λ'

theorem **pow_zsmul_mem_Ha** [Fintype T] {M : ℤ} (h : ∀ v ∈ Λ, M • v ∈ Λ') (hP : P ∈ Ha τ c c' Λ) : M ^ typeDegree c c' • P ∈ Ha τ c c' Λ'

theorem **transpose_mul_map_ccHom_eq_one** {A : Matrix (Fin n) (Fin n) E} (hA : ∀ x y, herm 1 (A *ᵥ x) (A *ᵥ y) = herm 1 x y) : Aᵀ * A.map (ccHom E) = 1

theorem **pushLetters_eq_colAct** {γ : GL (Fin n) E} (hγ : ∀ x y, herm 1 (γ.val *ᵥ x) (γ.val *ᵥ y) = herm 1 x y) (P : Poly K T p q (Fin n)) : pushLetters p q τ γ.val P = colAct K T p q (Fin n) (fun t => Matrix.GeneralLinearGroup.map ((τ t).comp (ccHom E)) γ) P

theorem **pushLetters_mem_rowHighest_iff** {γ : GL (Fin n) E} (hγ : ∀ x y, herm 1 (γ.val *ᵥ x) (γ.val *ᵥ y) = herm 1 x y) : pushLetters p q τ γ.val P ∈ rowHighest K c c' ↔ P ∈ rowHighest K c c'

theorem **map_pushLetters_Ha** {γ : GL (Fin n) E} (hγ : ∀ x y, herm 1 (γ.val *ᵥ x) (γ.val *ᵥ y) = herm 1 x y) (Λ : Submodule (𝒪 E) (Fin n → E)) : (Ha τ c c' Λ).map ((pushLetters p q τ γ.val).toLinearMap.restrictScalars (𝒪 K)) = Ha τ c c' (Λ.map (γ.val.mulVecLin.restrictScalars (𝒪 E)))

theorem **exists_zsmul_mem_of_mem_span** {v : Fin n → E} (hv : v ∈ Submodule.span E (Λ : Set _)) : ∃ N : ℤ, N ≠ 0 ∧ N • v ∈ Λ

theorem **exists_zsmul_le** (hΛ : Λ.FG) (hΛ' : Submodule.span E (Λ' : Set (Fin n → E)) = ⊤) : ∃ N : ℤ, N ≠ 0 ∧ ∀ v ∈ Λ, N • v ∈ Λ'

theorem **mem_pi_one_iff** {v : Fin n → E} : v ∈ Submodule.pi Set.univ (fun _ : Fin n => (1 : Submodule (𝒪 E) E)) ↔ ∀ k, IsIntegral ℤ (v k)

theorem **span_range_single_eq_pi_one** : Submodule.span (𝒪 E) (Set.range fun k : Fin n => (Pi.single k 1 : Fin n → E)) = Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)

theorem **fg_pi_one** : (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)).FG

theorem **span_pi_one_eq_top** : Submodule.span E (Submodule.pi Set.univ (fun _ : Fin n => (1 : Submodule (𝒪 E) E)) : Set (Fin n → E)) = ⊤

theorem **mem_range_map_ringOfIntegers_iff** {σ : Type*} {P : MvPolynomial σ K} : (∃ Q, MvPolynomial.map (algebraMap (𝒪 K) K) Q = P) ↔ ∀ m, IsIntegral ℤ (coeff m P)

theorem **symO_pi_one** : symO p q τ (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)) = (mapAlgHom (σ

theorem **mem_symO_pi_one_iff** : P ∈ symO p q τ (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)) ↔ ∀ m, IsIntegral ℤ (coeff m P)

theorem **Ha_pi_one_eq_map** : Ha τ c c' (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)) = (rowHighest (𝒪 K) c c').map (mapAlgHom (σ

theorem **Ha_pi_one_eq_span** : Ha τ c c' (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)) = Submodule.span (𝒪 K) (MvPolynomial.map (Int.castRingHom K) '' (rowHighest ℤ (κ

theorem **fg_Ha** (hΛ : Λ.FG) : (Ha τ c c' Λ).FG

theorem **exists_zsmul_mem_Ha** (hΛ : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) (hP : P ∈ rowHighest K c c') : ∃ N : ℤ, N ≠ 0 ∧ N • P ∈ Ha τ c c' Λ

theorem **span_Ha** (hΛ : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) : Submodule.span K (Ha τ c c' Λ : Set (Poly K T p q (Fin n))) = rowHighest K c c'

theorem **Ha_ne_bot** (h : p + q ≤ n) (hΛ : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) : Ha τ c c' Λ ≠ ⊥

theorem **pow_zsmul_Ha_of_isCoprime** {ℓ M : ℤ} (hM : IsCoprime M ℓ) (h : ∀ v ∈ Λ, M • v ∈ Λ') (h' : ∀ v ∈ Λ', M • v ∈ Λ) : IsCoprime (M ^ typeDegree c c') ℓ ∧ (∀ P ∈ Ha τ c c' Λ, M ^ typeDegree c c' • P ∈ Ha τ c c' Λ') ∧ ∀ P ∈ Ha τ c c' Λ', M ^ typeDegree c c' • P ∈ Ha τ c c' Λ

theorem **integral_harmonic_on_lattice** (hfg : Λ.FG) (hspan : Submodule.span E (Λ : Set (Fin n → E)) = ⊤) : (∀ (e : T → Fin p → Fin n → E) (f : T → Fin q → Fin n → E), (∀ t a, e t a ∈ Λ) → (∀ t b, f t b ∈ Λ) → (∀ t a b, herm 1 (e t a) (f t b) = 0) → minorProd c c' (fun t a k => τ t (e t a k)) (fun t b k => τ t (cc (f t b k))) ∈ Ha τ c c' Λ) ∧ ((Ha τ c c' Λ).FG ∧ Submodule.span K (Ha τ c c' Λ : Set (Poly K T p q (Fin n))) = rowHighest K c c') ∧ ((∀ P, P ∈ symO p q τ (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)) ↔ ∀ m, IsIntegral ℤ (coeff m P)) ∧ Ha τ c c' (Submodule.pi Set.univ fun _ : Fin n => (1 : Submodule (𝒪 E) E)) = Submodule.span (𝒪 K) (MvPolynomial.map (Int.castRingHom K) '' (rowHighest ℤ (κ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.QuotientNoPositiveMaxOffCommonZeros

theorem **exists_dir** {n : ℕ} (s : Fin n → V n → ℂ) (k0 : Fin n) (z0 : V n) (hc : ContinuousAt (s k0) z0) (h0 : s k0 z0 ≠ 0) : ∃ ξ : Fin n → ℂ, ξ ≠ 0 ∧ ∀ k, fderiv ℂ (fun z => s k z / s k0 z) z0 (WithLp.toLp 2 ξ) = 0

private theorem **contDiffAt_two_log_sum_normSq** {n : ℕ} {U : Set (V n)} (hU : IsOpen U) {z0 : V n} (hz : z0 ∈ U) {M : ℕ} {w : Fin M → V n → ℂ} (hw : ∀ k, DifferentiableOn ℂ (w k) U) (hp : 0 < ∑ k, Complex.normSq (w k z0)) : ContDiffAt ℝ 2 (fun z => Real.log (∑ k, Complex.normSq (w k z))) z0

private theorem **re_leviV_add** {n : ℕ} {f g : V n → ℝ} {x : V n} (hf : ContDiffAt ℝ 2 f x) (hg : ContDiffAt ℝ 2 g x) {ξ : Fin n → ℂ} : (leviV n (fun y => ((f y + g y : ℝ) : ℂ)) x ξ).re = (leviV n (fun y => (f y : ℂ)) x ξ).re + (leviV n (fun y => (g y : ℂ)) x ξ).re

private theorem **exists_pow_mul_mul_mem_subring** {n : ℕ} {f : V n → ℝ} {x : V n} (c : ℝ) (hf : ContDiffAt ℝ 2 f x) {ξ : Fin n → ℂ} : (leviV n (fun y => ((c * f y : ℝ) : ℂ)) x ξ).re = c * (leviV n (fun y => (f y : ℂ)) x ξ).re

theorem **no_localMax** {n : ℕ} {U : Set (V n)} (hU : IsOpen U) {z0 : V n} (hz : z0 ∈ U) {a b c d : V n → ℂ} (ha : DifferentiableOn ℂ a U) (hb : DifferentiableOn ℂ b U) (hc : DifferentiableOn ℂ c U) (hd : DifferentiableOn ℂ d U) (ha0 : a z0 ≠ 0) (hb0 : b z0 ≠ 0) (hc0 : c z0 ≠ 0) (hd0 : d z0 ≠ 0) {s : Fin n → V n → ℂ} (hs : ∀ k, DifferentiableOn ℂ (s k) U) {k0 : Fin n} (hs0 : s k0 z0 ≠ 0) {φ : V n → ℝ} (hφ : ContDiffAt ℝ 2 φ z0) (hpos : ∀ ξ : Fin n → ℂ, ξ ≠ 0 → 0 < (leviV n (fun w => ((φ w : ℝ) : ℂ)) z0 ξ).re) (t e : ℝ) : ¬ IsLocalMax (fun z => Real.log (Complex.normSq (a z)) + Real.log (Complex.normSq (b z)) + Real.log (Complex.normSq (c z)) + t * Real.log (Complex.normSq (d z)) + e * Real.log (∑ k, Complex.normSq (s k z)) + (1 / 2) * φ z) z0

private theorem **logq_lemma_4** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (f : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (hf : f x ≠ 0) (h2 : Xa.th2 x ≠ 0) (h1 : Xa.th1 x ≠ 0) (he : Xa.et x ≠ 0) (hs : 0 < ∑ k, Complex.normSq (Fl.s k x)) : 0 < X.pnorm (Xa.Fp + Xa.T • Xa.L) f x ^ 2 * X.pnorm ((nd X + 4) • Xa.L − Xa.Fp) Xa.th2 x ^ 2 * X.pnorm (2 • Xa.L − Xa.de) Xa.th1 x ^ 2 * (X.pnorm ((Xa.i * (Xa.g0 / 2)) • Xa.de) Xa.et x ^ 2) ^ ((1 : ℝ) / (Xa.i * (Xa.g0 / 2) : ℕ)) / (∑ k, X.pnorm Xa.L (Fl.s k) x ^ 2) ^ ((Xa.T : ℝ) + nd X + 6 + 1 / 2) ∧ Real.log (X.pnorm (Xa.Fp + Xa.T • Xa.L) f x ^ 2 * X.pnorm ((nd X + 4) • Xa.L − Xa.Fp) Xa.th2 x ^ 2 * X.pnorm (2 • Xa.L − Xa.de) Xa.th1 x ^ 2 * (X.pnorm ((Xa.i * (Xa.g0 / 2)) • Xa.de) Xa.et x ^ 2) ^ ((1 : ℝ) / (Xa.i * (Xa.g0 / 2) : ℕ)) / (∑ k, X.pnorm Xa.L (Fl.s k) x ^ 2) ^ ((Xa.T : ℝ)

private theorem **contDiffAt_two_phla_cpt** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} (X : Data E r) (la : Wt X.Ψ r) (x0 : Frame X.Ψ r) {z : V (nd X)} (hz : z ∈ OmF X x0) : ContDiffAt ℝ 2 (fun w => phla X la (cpt X x0 w)) z

theorem **no_pos_max** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (f : Frame X.Ψ r → ℂ), X.IsSec (Xa.Fp + Xa.T • Xa.L) f → let Φ : Frame X.Ψ r → ℝ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.QuotientNoPositiveMaxOffCommonZerosStatement

theorem **_root_.GlobalTheory.T.quotient_no_positive_max_off_common_zeros** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (f : Frame X.Ψ r → ℂ), X.IsSec (Xa.Fp + Xa.T • Xa.L) f → let Φ : Frame X.Ψ r → ℝ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RamifiedPrimePowersSizeBounds

theorem **pPow_size** : ∀ (D : D7) (j : ℕ) (w : WeightFamilies.Wt D j), 1 ≤ D.host.q ∧ 1 ≤ D.host.d ∧ 0 ≤ Real.log (4 * ((D.R.f : ℝ) / 2) ^ 2 * D.host.q * D.host.d) ∧ 0 ≤ Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ)) ∧ (c2 D w : ℝ) * Real.log D.host.QS +

(Formalism.MainNotions.DIV.Cons.cHalf w : ℝ) * Real.log D.host.q + (Formalism.MainNotions.DIV.Cons.cHalf w : ℝ) * Real.log D.host.d ≤ ((WeightAlgebra.DF w : ℝ) + 1) * Real.log (4 * ((D.R.f : ℝ) / 2) ^ 2 * D.host.q * D.host.d) ∧ ∀ a b c : ℕ, 1 ≤ pPow D a b c ∧ Real.log (pPow D a b c) = a * Real.log D.host.QS + b * Real.log D.host.q + c * Real.log D.host.d

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RamifiedPrimePowersSizeBoundsStatement

theorem **_root_.GlobalTheory.T.ramifiedPrimePowers_size_bounds** : ∀ (D : D7) (j : ℕ) (w : WeightFamilies.Wt D j), 1 ≤ D.host.q ∧ 1 ≤ D.host.d ∧ 0 ≤ Real.log (4 * ((D.R.f : ℝ) / 2) ^ 2 * D.host.q * D.host.d) ∧ 0 ≤ Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ)) ∧ (c2 D w : ℝ) * Real.log D.host.QS + (Formalism.MainNotions.DIV.Cons.cHalf w : ℝ) * Real.log D.host.q + (Formalism.MainNotions.DIV.Cons.cHalf w : ℝ) * Real.log D.host.d ≤ ((WeightAlgebra.DF w : ℝ) + 1) * Real.log (4 * ((D.R.f : ℝ) / 2) ^ 2 * D.host.q * D.host.d) ∧ ∀ a b c : ℕ, 1 ≤ pPow D a b c ∧ Real.log (pPow D a b c) = a * Real.log D.host.QS + b * Real.log D.host.q + c * Real.log D.host.d

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RawChartBaseChangeRigGerm

theorem **map_psTrunc** {κ : Type} [Fintype κ] [DecidableEq κ] (m : ℕ) (f : MvPowerSeries κ S) : MvPolynomial.map ψ (psTrunc m f) = psTrunc m (MvPowerSeries.map ψ f)

theorem **map_psIn** (d : ℕ) {n n' : ℕ} (m : ℕ) (r : Fin n → Fin n') (f : MvPowerSeries (Fin n × Fin d) S) : MvPolynomial.map ψ (psIn d m r f) = psIn d m r (MvPowerSeries.map ψ f)

theorem **spMap_comp_eq** {A B C D : Type} [CommRing A] [CommRing B] [CommRing C] [CommRing D] (f : A →+* B) (g : B →+* D) (f' : A →+* C) (g' : C →+* D) (h : g.comp f = g'.comp f') : (spMap g ≫ spMap f : Sp D ⟶ Sp A) = spMap g' ≫ spMap f'

theorem **appTop_toΓ_spMap** (a : S) : (spMap ψ).appTop.hom (toΓ S a) = toΓ S' (ψ a)

theorem **isPullSec_whisker_comp** (a : ℤ) {T1 T2 T3 : Over S0} (h1 : T2 ⟶ T1) (h2 : T3 ⟶ T2) {s1 : LB.Sec (LPow A L a T1)} {s2 : LB.Sec (LPow A L a T2)} {s3 : LB.Sec (LPow A L a T3)} (e1 : IsPullSec A L a (h1 ▷ A) (whiskerRight_snd _ _) s1 s2) (e2 : IsPullSec A L a (h2 ▷ A) (whiskerRight_snd _ _) s2 s3) : IsPullSec A L a ((h2 ≫ h1) ▷ A) (whiskerRight_snd _ _) s1 s3

theorem **subst_rjBC** (m n n' : ℕ) (φ : Fin n × Fin d → MvPolynomial (Fin n' × Fin d) S) (φ' : Fin n × Fin d → MvPolynomial (Fin n' × Fin d) S') (h : ∀ p, φ' p = MvPolynomial.map ψ (φ p)) : (Formalism.L2.Pj.homAt (spMap ψ ≫ x0) (jetSubstP m φ')).left ≫ (rjBC ψ x0 d m n).left = (rjBC ψ x0 d m n').left ≫ (Formalism.L2.Pj.homAt x0 (jetSubstP m φ)).left

theorem **aug_rjBC** (m n : ℕ) : (Formalism.L2.Pj.homAt (spMap ψ ≫ x0) (jetAug : Jet S' (Fin n × Fin d) m →ₐ[S'] S')).left ≫ (rjBC ψ x0 d m n).left = spMap ψ ≫ (Formalism.L2.Pj.homAt x0 (jetAug : Jet S (Fin n × Fin d) m →ₐ[S] S)).left

theorem **base_rjBC** (m n : ℕ) : rjBase (spMap ψ ≫ x0) d m n ≫ ptBC ψ x0 = rjBC ψ x0 d m n ≫ rjBase x0 d m n

theorem **overAt_hom_bc** : (overAt (spMap ψ ≫ x0) S').hom = spMap ψ ≫ (overAt x0 S).hom

theorem **val_genPow** {Y Z : Scheme.{0}} {K : Type} [Fintype K] (φ : K → (Y ⟶ Xo.univ.t.G.left)) (ε : K → ℤ) (g : LB.Gen (LB.LinComb Xo.univ.t.PP.L φ ε)) (k : ℕ) (r : Z ⟶ Y) (z : Z) (J) : val (genPow φ ε g k) r z J = val g r z J ^ k

theorem **val_reL_bc** (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G) (hx : ∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) (e : ∀ m, RFrTy x0 d x m) (e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m) (he : ∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) (wP : PrimTy H Xo) (m : ℕ) {Z : Scheme.{0}} (r : Z ⟶ (rjO (spMap ψ ≫ x0) d m 1).left) (z : Z) {J} (hJ : Good Xo.univ.t.PP.L z J) : val (reL (spMap ψ ≫ x0) d x' (fun a i => MvPowerSeries.map ψ (act a i)) e' wP m) r z J = val (reL x0 d x act e wP m) (r ≫ (rjBC ψ x0 d m 1).left) z J

theorem **rawChart_baseChange_rig_germ** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (e : ∀ m, RFrTy x0 d x m) (e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m), (∀ m, (e m).IsGen) → (∀ m, (e' m).IsGen) → (∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) → ∀ (wP : PrimTy H Xo), wP.IsGen → (LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L _ _ (reL x0 d x act e wP 0) (homAt x0 (jetAug : Jet

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RawChartBaseChangeRigGermStatement

theorem **_root_.GlobalTheory.T.rawChart_baseChange_rig_germ_alt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (S S' : Type) [CommRing S] [CommRing S'] (ψ : S →+* S') (x0 : Sp S ⟶ Xo.X) (d : ℕ) (F : Fin d → MvPowerSeries (Fin 2 × Fin d) S) (act : 𝒪 E → Fin d → MvPowerSeries (Fin 1 × Fin d) S) (x : ∀ m, rjO x0 d m 1 ⟶ Xo.univ.t.G) (x' : ∀ m, rjO (spMap ψ ≫ x0) d m 1 ⟶ Xo.univ.t.G), (∀ m, x' m = rjBC ψ x0 d m 1 ≫ x m) → ∀ (e : ∀ m, RFrTy x0 d x m) (e' : ∀ m, RFrTy (spMap ψ ≫ x0) d x' m), (∀ m, (e m).IsGen) → (∀ m, (e' m).IsGen) → (∀ m, LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (e' m) (LB.pullGen Xo.univ.t.PP.L (cφ ![rjpt x0 d x m (0 : Fin 2), rjpt x0 d x m (1 : Fin 2)]) (cε 2) (e m) (rjBC ψ x0 d m 2).left)) → ∀ (wP : PrimTy H Xo), wP.IsGen → (LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ (LB.pullGen Xo.univ.t.PP.L _ _ (reL x0 d x act e wP 0) (homAt x0 (jetAug : Jet

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RawChartFormalBiextensionBaseChange

theorem **isCanonGauge_mapFB** {S S' : Type} [CommRing S] [Algebra ℚ S] [CommRing S'] [Algebra ℚ S'] (ψ : S →+* S') {d : ℕ} {O : Type} [CommRing O] [StarRing O] (B : Germ.FB S d O) (h : Germ.PS S 2 d) (hh : B.IsCanonGauge S h) : (FBMap.mapFB ψ B).IsCanonGauge S' (MvPowerSeries.map ψ h)

theorem **rawChart_baseChange** : RawBCSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RawChartFormalBiextensionBaseChangeStatement

theorem **_root_.GlobalTheory.T.rawChart_formalBiextension_baseChange** : RawBCSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ReadingOfThetaGroupAction

theorem **exists_wchart_complexPt** {X Z : Scheme.{0}} (L : LB X) (hL : L.IsLB) {K : Type} [Fintype K] (φ : K → (Z ⟶ X)) (q : Sp ℂ ⟶ Z) : ∃ j : K → L.ι, ⊤ ≤ q ⁻¹ᵁ LB.wchart L φ j

theorem **AnFr.rd_of_isActOf** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ) (c : ℂ) (s s' : LB.Sec (LPow A L.L k (Over.mk x0))) (θ : (Fin g → ℂ) → ℂ), Φ.ThConst k U x c → IsActOf A L.L k U s s' → Φ.Rd k s θ → Φ.Rd k s' (fun z => c * fock Φ.Hm k x θ z)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ReadingOfThetaGroupActionStatement

theorem **_root_.GlobalTheory.T.reading_of_thetaGroup_action** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ) (c : ℂ) (s s' : LB.Sec (LPow A L.L k (Over.mk x0))) (θ : (Fin g → ℂ) → ℂ), Φ.ThConst k U x c → IsActOf A L.L k U s s' → Φ.Rd k s θ → Φ.Rd k s' (fun z => c * fock Φ.Hm k x θ z)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ReadingTwistedPullback

theorem **AnFr.rd_twistPull** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (ψ : Over.mk x0 ⊗ A ⟶ Over.mk x0 ⊗ A) (m : Over.mk x0 ⊗ A ⟶ A) (hm : ψ ≫ snd (Over.mk x0) A = m) (a b : ℤ) (α : LB.Gen (LB.LinComb L.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ReadingTwistedPullbackStatement

theorem **_root_.GlobalTheory.T.reading_twisted_pullback** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (ψ : Over.mk x0 ⊗ A ⟶ Over.mk x0 ⊗ A) (m : Over.mk x0 ⊗ A ⟶ A) (hm : ψ ≫ snd (Over.mk x0) A = m) (a b : ℤ) (α : LB.Gen (LB.LinComb L.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RealBounds

theorem **real_bounds** (f r m ρ w D q t : ℝ) (hf : 1 ≤ f) (hr : 1 ≤ r) (hm : 1 ≤ m) (hmf : m ≤ f) (hf2 : f ^ 2 ≤ ρ) (hρ : 16 ≤ ρ) (hw0 : 0 ≤ w) (hw : w ≤ f ^ 2) (hD : 1 ≤ D) (hq : (ρ ^ 7) ^ 4 * (90 * (f * r + D)) * D * r ≤ q) (ht : t = q * (q / (90 * (f * r + D) * (ρ ^ 7) ^ 4))) : (2 * f * m) ^ 2 * (2 * w * ρ) * (f * (90 * (f * r + D)) * (ρ ^ 7) ^ 3) < q ∧ 1 ≤ q / (90 * (f * r + D) * (ρ ^ 7) ^ 4) ∧ 90 * m ^ 2 * (2 * f * m) ^ 2 * ρ ≤ q / 2 ∧ D ≤ q ∧ 4 * D * (7 + Real.log (q * D)) + 2 * f * m + 1 ≤ t ∧ f * r * Real.log 2 + 2 * (2 * f * m) + 3 + (D + 1) * (r * Real.log r + 3 * r + 3) ≤ t

theorem **runOK_of_le** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r DF q m : ℕ), 1 ≤ DF → 2 ≤ I2.PT.f E * r → 1 ≤ m → m ^ 4 ≤ I2.PT.f E → (I2.PT.f E : ℝ) ^ 2 ≤ I2.PT.rd E → 16 ≤ I2.PT.rd E → I2.RR (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RowAddGenericRowAddTerm

theorem **eval_rowAddGeneric** (u w : ι → σ) (t : K) (P : MvPolynomial σ K) : Polynomial.eval (C t) (rowAddGeneric u w P) = bind₁ (fun s => X s + ∑ k, if s = w k then C t * X (u k) else 0) P

theorem **eval_zero_rowAddGeneric** (u w : ι → σ) (P : MvPolynomial σ K) : Polynomial.eval 0 (rowAddGeneric u w P) = P

theorem **rowAddGeneric_rowAddTerm** (u w : ι → σ) (huw : ∀ k l, u k ≠ w l) (s : σ) : rowAddGeneric u w (rowAddTerm K u w s) = Polynomial.C (rowAddTerm K u w s)

theorem **sum_X_mul_pderiv_X** (u w : ι → σ) (s : σ) : ∑ k, X (u k) * pderiv (w k) (X s : MvPolynomial σ K) = rowAddTerm K u w s

theorem **derivative_rowAddGeneric** (u w : ι → σ) (huw : ∀ k l, u k ≠ w l) (P : MvPolynomial σ K) : Polynomial.derivative (rowAddGeneric u w P) = rowAddGeneric u w (∑ k, X (u k) * pderiv (w k) P)

theorem **rowAdd_invariant_iff** {K σ ι : Type*} [Field K] [CharZero K] [DecidableEq σ] [Fintype ι] (u w : ι → σ) (hw : Function.Injective w) (huw : ∀ k l, u k ≠ w l) (P : MvPolynomial σ K) : List.TFAE [∀ t : ℤ, bind₁ (fun s => X s + ∑ k, if s = w k then C (t : K) * X (u k) else 0) P = P, ∀ t : K, bind₁ (fun s => X s + ∑ k, if s = w k then C t * X (u k) else 0) P = P, ∑ k, X (u k) * pderiv (w k) P = 0]

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RowHighestLeSpanColWordComplex

theorem **coeffForm_smul_map** (φ : K →ₗ[F] F) (r : K) (Q : MvPolynomial σ F) : coeffForm φ (r • MvPolynomial.map (algebraMap F K) Q) = φ r • Q

theorem **coeffForm_monomial** (φ : K →ₗ[F] F) (m : σ →₀ ℕ) (r : K) : coeffForm φ (monomial m r) = monomial m (φ r)

theorem **coeff_coeffForm** (φ : K →ₗ[F] F) (m : σ →₀ ℕ) (P : MvPolynomial σ K) : coeff m (coeffForm φ P) = φ (coeff m P)

theorem **coeffForm_comm** (φ : K →ₗ[F] F) (L : MvPolynomial σ F →ₗ[F] MvPolynomial σ F) (L' : MvPolynomial σ K →ₗ[K] MvPolynomial σ K) (h : ∀ Q, MvPolynomial.map (algebraMap F K) (L Q) = L' (MvPolynomial.map (algebraMap F K) Q)) (P : MvPolynomial σ K) : coeffForm φ (L' P) = L (coeffForm φ P)

theorem **mem_span_map_of_coeffForm_mem** (V : Set (MvPolynomial σ F)) {P : MvPolynomial σ K} (h : ∀ φ : K →ₗ[F] F, coeffForm φ P ∈ V) : P ∈ Submodule.span K (MvPolynomial.map (algebraMap F K) '' V)

theorem **coeffForm_mem_span** (φ : K →ₗ[F] F) (V : Set (MvPolynomial σ F)) {P : MvPolynomial σ K} (h : P ∈ Submodule.span K (MvPolynomial.map (algebraMap F K) '' V)) : coeffForm φ P ∈ Submodule.span F V

theorem **mem_span_of_map_mem_span** [Nontrivial K] (V : Set (MvPolynomial σ F)) {P : MvPolynomial σ F} (h : MvPolynomial.map (algebraMap F K) P ∈ Submodule.span K (MvPolynomial.map (algebraMap F K) '' V)) : P ∈ Submodule.span F V

theorem **coeffForm_mem_rowHighest** (φ : K →ₗ[F] F) {c : T → Fin p → ℕ} {c' : T → Fin q → ℕ} {P : Poly K T p q κ} (hP : P ∈ rowHighest K c c') : coeffForm φ P ∈ rowHighest F c c'

theorem **span_map_rowHighest_algebraMap** (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) : Submodule.span K (MvPolynomial.map (algebraMap F K) '' (rowHighest (κ

theorem **map_colLie** (f : R →+* S) (t : T) (k l : κ) (P : Poly R T p q κ) : MvPolynomial.map f (colLie R p q t k l P) = colLie S p q t k l (MvPolynomial.map f P)

theorem **colLie_lie** [DecidableEq κ] (t : T) (i j k l : κ) : colLie R p q t i j * colLie R p q t k l − colLie R p q t k l * colLie R p q t i j = (if j = k then colLie R p q t i l else 0) − (if l = i then colLie R p q t k j else 0)

theorem **fockInner_colLie** (t : T) (k l : κ) (P Q : Poly ℂ T p q κ) : fockInner (colLie ℂ p q t k l P) Q = fockInner P (colLie ℂ p q t l k Q)

theorem **map_colWord** (f : R →+* S) (w : List (T × Fin n × Fin n)) : MvPolynomial.map f (colWord R h c c' w) = colWord S h c c' w

theorem **image_map_range_colWord** (f : R →+* S) : MvPolynomial.map f '' Set.range (colWord R h c c') = Set.range (colWord S h c c')

theorem **colWord_mem_orbitSpan** {K : Type*} [Field K] [Infinite K] (w : List (T × Fin n × Fin n)) : colWord K h c c' w ∈ orbitSpan K h c c'

theorem **colWord_mem_rowHighest** (w : List (T × Fin n × Fin n)) : colWord R h c c' w ∈ rowHighest R c c'

theorem **colLie_mem_span_colWord** (t : T) (k l : Fin n) {P : Poly R T p q (Fin n)} (hP : P ∈ Submodule.span R (Set.range (colWord R h c c'))) : colLie R p q t k l P ∈ Submodule.span R (Set.range (colWord R h c c'))

theorem **basePoly_mem_of_colLie_stable** {K : Type*} [Field K] [CharZero K] (W : Submodule K (Poly K T p q (Fin n))) (hW : W ≤ rowHighest K c c') (hs : ∀ t k l, ∀ w ∈ W, colLie K p q t k l w ∈ W) (hW0 : W ≠ ⊥) : basePoly K h c c' ∈ W

theorem **rowHighest_le_span_colWord_complex** : rowHighest ℂ c c' ≤ Submodule.span ℂ (Set.range (colWord ℂ h c c'))

theorem **rowHighest_eq_span_colWord** [CommRing K] [Algebra ℚ K] : rowHighest K c c' = Submodule.span K (Set.range (colWord K h c c'))

theorem **rowHighest_eq_orbitSpan** [Field K] [CharZero K] : rowHighest K c c' = orbitSpan K h c c'

theorem **span_map_rowHighest_int** [CommRing K] [Algebra ℚ K] : Submodule.span K (MvPolynomial.map (Int.castRingHom K) '' (rowHighest (κ

theorem **rowHighest_orbitSpan_main** {T : Type*} [Fintype T] {p q n : ℕ} (h : p + q ≤ n) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (K : Type*) [Field K] [CharZero K] : rowHighest ℂ c c' = orbitSpan ℂ h c c' ∧ rowHighest K c c' = orbitSpan K h c c' ∧ Submodule.span K (MvPolynomial.map (Int.castRingHom K) '' (rowHighest (κ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RuleBaseChangeSectionValuesSemilinear

theorem **rulBC_sec_val_linear** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F), (∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ f ∈ sec Xo c a n, rulBC hle f ∈ sec Xo c' a n ∧ ∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S), evR Xo c' (rulBC hle f) y = evR Xo c f (y.res hle h)) ∧ (∀ (t : c.F) (f g : Rul Xo c), rulBC hle (t • f + g) = incF c c' hle t • rulBC hle f + rulBC hle g)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RuleBaseChangeSectionValuesSemilinearStatement

theorem **_root_.GlobalTheory.T.ruleBaseChange_section_values_semilinear** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F), (∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ f ∈ sec Xo c a n, rulBC hle f ∈ sec Xo c' a n ∧ ∀ (S : Type) [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S), evR Xo c' (rulBC hle f) y = evR Xo c f (y.res hle h)) ∧ (∀ (t : c.F) (f g : Rul Xo c), rulBC hle (t • f + g) = incF c c' hle t • rulBC hle f + rulBC hle g)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunHypothesesFormsScalarsOfParameterConditions

theorem **log_DF_le_of_rows** {D : D7} {j : ℕ} (large₂ : SizeData₂ D) (ka w : WeightFamilies.Wt D j) (h3 : 3 ≤ DF w) (hle : DF w ≤ DF ka) (hrow : Real.log (1 + DF ka) ≤ 2 * large₂.sc.L) : Real.log (DF w) ≤ 2 * large₂.sc.L

theorem **hypC_record** : ∀ (D : D7) (T : Top D) (h : LawsP T) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), ParOK T EX p j → PageOK large₂ (recPJ h j hj) H C2 A2 CT ka wL wF Tn nP nW h0 → HypC large₂ (recPJ h j hj) H C2 A2 7 0 CT (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) ka wL wF Tn nP nW h0 (nusR h j wL) ∧ LatAt (recB h.toLawsCore) j ka ∧ (∀ w : WeightFamilies.Wt D j, GlobalTheory.RunIn.ThInt (recPJ h j hj) hinj EX w) ∧ PageSupp D large₂ (recB h.toLawsCore) C2 A2 nW (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) wL ∧ ∃ Sh : Finset ((recB h.toLawsCore).Cusp j wF), Submodule.span (recB h.toLawsCore).KO (Sh : Set ((recB h.toLawsCore).Cusp j wF)) = ⊤ ∧ ∀ G : GlobalFamilyData (recPJ h j hj) ka wL wF, (∀ i, G.th' i ∈ Sh) → G.TiedTh (scalOf

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunHypothesesFormsScalarsOfParameterConditionsStatement

theorem **_root_.GlobalTheory.T.runHypotheses_formsScalars_of_parameterConditions** : ∀ (D : D7) (T : Top D) (h : LawsP T) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), ParOK T EX p j → PageOK large₂ (recPJ h j hj) H C2 A2 CT ka wL wF Tn nP nW h0 → HypC large₂ (recPJ h j hj) H C2 A2 7 0 CT (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) ka wL wF Tn nP nW h0 (nusR h j wL) ∧ LatAt (recB h.toLawsCore) j ka ∧ (∀ w : WeightFamilies.Wt D j, GlobalTheory.RunIn.ThInt (recPJ h j hj) hinj EX w) ∧ PageSupp D large₂ (recB h.toLawsCore) C2 A2 nW (scalOfP (recB h.toLawsCore) hinj EX p ka wL wF Tn) wL ∧ ∃ Sh : Finset ((recB h.toLawsCore).Cusp j wF), Submodule.span (recB h.toLawsCore).KO (Sh : Set ((recB h.toLawsCore).Cusp j wF)) = ⊤ ∧ ∀ G : GlobalFamilyData (re

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunScalarsNeZeroAndLogBounded

theorem **one_le_f** (D : D7) : (1 : ℝ) ≤ D.R.f

theorem **one_le_r** (D : D7) : (1 : ℝ) ≤ D.R.r

theorem **size_eq** (w : WeightFamilies.Wt D j) : w.size = (WeightAlgebra.DF w : ℝ) + (D.R.f : ℝ) * ((max (D.R.sigW j).2 1 : ℕ) : ℝ) * (D.R.r : ℝ) * (w.n : ℝ) + 1

theorem **size_ge** (w : WeightFamilies.Wt D j) : (WeightAlgebra.DF w : ℝ) + (D.R.f : ℝ) * (D.R.r : ℝ) * (w.n : ℝ) + 1 ≤ w.size

theorem **n_le_size** (w : WeightFamilies.Wt D j) : (w.n : ℝ) ≤ w.size

theorem **size_comb** (T : ℕ) (wL wF : WeightFamilies.Wt D j) : (wadd wF (wsmul T wL)).size + T = T * wL.size + wF.size

theorem **log_zOwn_le** (p : ℕ) (hp : 0 < p) (w : WeightFamilies.Wt D j) : Real.log (zOwnP D p w : ℝ) ≤ 2 * Real.log p * w.size

theorem **lamStar_nonneg** (w : WeightFamilies.Wt D j) : 0 ≤ lamStar D

theorem **comb_eq** (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (G wL wF : WeightFamilies.Wt D j) (T : ℕ) : (zOwnP D p G : 𝒪 B.KO) * nnW B hinj EX wL ^ T * nnW B hinj EX wF * (pPow D (T * c2 D wL + c2 D wF) (T * Formalism.MainNotions.DIV.Cons.cHalf wL + Formalism.MainNotions.DIV.Cons.cHalf wF) (T * Formalism.MainNotions.DIV.Cons.cHalf wL + Formalism.MainNotions.DIV.Cons.cHalf wF) : 𝒪 B.KO) = comb B hinj EX p G wL wF T

theorem **comb_ne_zero** (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (hp : 0 < p) (G wL wF : WeightFamilies.Wt D j) (T : ℕ) : comb B hinj EX p G wL wF T ≠ 0

theorem **log_comb_le** (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (hp : 0 < p) (G wL wF : WeightFamilies.Wt D j) (T : ℕ) (σ : B.KO →+* ℂ) : Real.log ‖σ (comb B hinj EX p G wL wF T : B.KO)‖ ≤ 2 * Real.log p * G.size + lamStar D * (T * wL.size + wF.size)

theorem **log_comb_le'** (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (hp : 0 < p) (wL wF : WeightFamilies.Wt D j) (hn : 1 ≤ wL.n) (T : ℕ) (σ : B.KO →+* ℂ) : Real.log ‖σ (comb B hinj EX p (wadd wF (wsmul T wL)) wL wF T : B.KO)‖ ≤ 2 * (lamStar D + 2 * Real.log p) * (wadd wF (wsmul T wL)).size

theorem **wt_eq** {ka wL wF : WeightFamilies.Wt D j} {T nP nW : ℕ} (hF : IsF' ka wL wF T nP) (hL : wL.n = nW) (hn : ka.n = nP + T * nW) : ka = wadd wF (wsmul T wL)

theorem **GlobalTheory.RunIn.scalOfP_size** : ∀ (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (p : ℕ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ), 0 < p → 1 ≤ nW → wL.n = nW → IsF' ka wL wF Tn nP → ka.n = nP + Tn * nW → (scalOfP B hinj EX p ka wL wF Tn).sF ≠ 0 ∧ (scalOfP B hinj EX p ka wL wF Tn).s0 ≠ 0 ∧ (scalOfP B hinj EX p ka wL wF Tn).sK ≠ 0 ∧ ∀ σ : B.KO →+* ℂ, (Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).sW : B.KO)‖ ≤ (lamStar D + 2 * Real.log p) * wL.size ∧ Real.log ‖σ

((scalOfP B hinj EX p ka wL wF Tn).sF : B.KO)‖ ≤ (lamStar D + 2 * Real.log p) * wF.size ∧ Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).s0 : B.KO)‖ ≤ 2 * (lamStar D + 2 * Real.log p) * (wadd wF (wsmul (WeightFamilies.dimY D j) wL)).size ∧ Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).sK : B.KO)‖ ≤ 2 * (lamStar D + 2 * Real.log p) * ka.size)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunScalarsNeZeroAndLogBoundedStatement

theorem **_root_.GlobalTheory.T.runScalars_ne_zero_and_log_bounded** : ∀ (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (p : ℕ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ), 0 < p → 1 ≤ nW → wL.n = nW → IsF' ka wL wF Tn nP → ka.n = nP + Tn * nW → (scalOfP B hinj EX p ka wL wF Tn).sF ≠ 0 ∧ (scalOfP B hinj EX p ka wL wF Tn).s0 ≠ 0 ∧ (scalOfP B hinj EX p ka wL wF Tn).sK ≠ 0 ∧ ∀ σ : B.KO →+* ℂ, (Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).sW : B.KO)‖ ≤ (lamStar D + 2 * Real.log p) * wL.size ∧ Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).sF : B.KO)‖ ≤ (lamStar D + 2 * Real.log p) * wF.size ∧ Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).s0 : B.KO)‖ ≤ 2 * (lamStar D + 2 * Real.log p) * (wadd wF (wsmul (WeightFamilies.dimY D j) wL)).size ∧ Real.log ‖σ ((scalOfP B hinj EX p ka wL wF Tn).sK : B.KO)‖ ≤ 2 * (lamStar D + 2 * Real.log p) *

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunScalarsNonzeroBoundedSupBound

theorem **scalB_rec** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → LawsSh T → (∀ j, InR D j → ∀ n, (T.dδ j n).l₂.A = 1) → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.OwnPr D p → FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) wF → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) ((recPJ h j hj).wF0 wL wF) → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) ka → ScalB large₂ (recPJ h j hj) H.C1 H.A1 (sca

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunScalarsNonzeroBoundedSupBoundStatement

theorem **_root_.GlobalTheory.T.runScalars_nonzero_bounded_supBound** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → LawsSh T → (∀ j, InR D j → ∀ n, (T.dδ j n).l₂.A = 1) → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.OwnPr D p → FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) wF → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj) ((recPJ h j hj).wF0 wL wF) → ReciprocityTheory.ChartRestriction.G.PullbackSections.SpanAt (recPJ h j hj)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.RunWeightDom

theorem **lt_of_succ_lt** {p : ℕ} {f : ℕ → ℤ} (h : ∀ i, i + 1 < p → f (i + 1) < f i) : ∀ i j, i < j → j < p → f j < f i

theorem **le_of_succ_lt** {p : ℕ} {f : ℕ → ℤ} (h : ∀ i, i + 1 < p → f (i + 1) < f i) (i : ℕ) (hi : i < p) : f (p − 1) ≤ f i

theorem **fdiv_spec** {x y T' : ℤ} (hT : 0 < T') (h : y = Int.fdiv x T') : ∃ ρ, 0 ≤ ρ ∧ ρ < T' ∧ x = T' * y + ρ

theorem **block** {p : ℕ} (hp : 0 < p) (ka wL wF : Fin p → ℕ) (T Dm r c : ℤ) (hD : 0 ≤ Dm) (hr : 0 ≤ r) (h40 : 40 * (Dm + 2) ≤ T) (hc : c ≤ T) (hg : ∀ i : Fin p, i.1 + 1 < p → (T + Dm + 2) * (T + Dm + 2 + 2 * r) ≤ gapZ ka i + 2) (hl : ∀ i : Fin p, i.1 + 1 = p → 2 * T ^ 2 ≤ ka i) (hL : ∀ i : Fin p, gapZ wL i = if i.1 + 1 < p then Int.fdiv (gapZ ka i + 2) (T + Dm + 2) else Int.fdiv ((ka i : ℤ) − c) (T + Dm + 2)) (hF : ∀ i, (wF i : ℤ) = ka i − T * wL i) : (∀ i, i + 1 < p → (ext0 wL (i + 1) : ℤ) < ext0 wL i) ∧ (∀ i, i + 1 < p → (ext0 wF (i + 1) : ℤ) − 2 + (Dm + 2) * (ext0 wL i − ext0 wL (i + 1)) ≤ ext0 wF i) ∧ 19 * T ≤ 10 * (ext0 wL (p − 1) : ℤ) ∧ (Dm + 2) * (ext0 wL (p − 1) : ℤ) + c ≤ ext0 wF (p − 1)

theorem **row** {r p q : ℕ} (hpq : p + q = r) (T Dm nL nF : ℤ) (hnL : 0 ≤ nL) (hnF : 0 ≤ nF) (LP LM FP FM : ℕ → ℤ) (L F : Fin r → ℤ) (hT : 0 ≤ T) (hL : ∀ a : Fin r, L a = if q ≤ (a : ℕ) then LP (a − q) + nL else −LM (q − 1 − a) − nL) (hF : ∀ a : Fin r, F a = if q ≤ (a : ℕ) then FP (a − q) + nF else −FM (q − 1 − a) − nF) (p1 : ∀ i, i + 1 < p → LP (i + 1) < LP i) (p2 : ∀ i, i + 1 < p → FP (i + 1) − 2 + (Dm + 2) * (LP i − LP (i + 1)) ≤ FP i) (p3 : 19 * T ≤ 10 * LP (p − 1)) (p4 : (Dm + 2) * LP (p − 1) + ((Dm + 2) * (2 * nL) + 2 * r) ≤ FP (p − 1)) (m1 : ∀ i, i + 1 < q → LM (i + 1) < LM i) (m2 : ∀ i, i + 1 < q → FM (i + 1) − 2 + (Dm + 2) * (LM i − LM (i + 1)) ≤ FM i) (m3 : 19 * T ≤ 10 * LM (q − 1)) (m4 : (Dm + 2) * LM (q − 1) + 0 ≤ FM (q − 1)) : (∀ i j : Fin r, i < j → (j : ℕ) < q → L j < L i) ∧ (∀ i j : Fin r, q ≤ (i : ℕ) → i < j → L j < L i) ∧ (∀ a b : Fin r, (b : ℕ) = a + 1 → (b : ℕ) ≠ q → F b

theorem **run_weight_dom** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → j < D.R.m → 8 ≤ D.R.r → ∀ k : (T.U j T.σ₀).Comp, IsStrictDom ((T.U j T.σ₀).X k) (laR T j wL k) ∧ IsStrictDom ((T.U j T.σ₀).X k) (2 • laR T j wL k − de2W ((T.U j T.σ₀).X k)) ∧ ∀ τ : ((T.U j T.σ₀).X k).Ψ, (∀ a b : Fin D.R.r, (b : ℕ) = a + 1 → (b : ℕ) ≠ ((T.U j T.σ₀).X k).p0 → (laR T j wF k − kapW ((T.U j T.σ₀).X k) − WeightFamilies.dimY D j • laR T j wL k) τ b + 2 * (laR T j wL k τ a − laR T j wL k τ b) ≤ (laR T j wF k − kapW ((T.U j T.σ₀).X k) − WeightFamilies.dimY D j • laR T j wL k) τ a) ∧ ∀ a b : Fin D.R.r, (a : ℕ) < ((T.U j T.σ₀).X k).p0 → ((T.U j T.σ₀).X k).p0 ≤ (b : ℕ) → 19 * (Tn : ℤ) ≤ 10 * (laR T j wL k τ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ScaledFamily

theorem **span_scaled** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} (X : PJ D B P j) (wL : WeightFamilies.Wt D j) (s : 𝒪 B.KO) {N : ℕ} (w0 : Fin N → B.Cusp j wL) (h : Submodule.span (𝒪 B.KO) (Set.range w0) = X.thAll wL) : Submodule.span (𝒪 B.KO) (Set.range fun k => (s : B.KO) • w0 k) = X.WWc s wL

theorem **scaled_family** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (S : Scal B) (N : ℕ) (w0 : Fin N → B.Cusp j wL) (a b G Prw : ℝ), 0 ≤ a → 0 ≤ b → (∀ σ : B.KO →+* ℂ, Real.log ‖σ (S.sW : B.KO)‖ ≤ a) → Submodule.span (𝒪 B.KO) (Set.range w0) = X.thAll wL → (∀ k σ, Real.log (B.supN j wL σ (w0 k)) ≤ b) → a + b ≤ G * wL.size → (∃ Sh : Finset (B.Cusp j wL), Submodule.span (𝒪 B.KO) (Sh : Set (B.Cusp j wL)) = X.WWc S.sW wL ∧ ∃ mx : (B.KO →+* ℂ) → ℝ, (∀ σ, ∀ g ∈ Sh, Real.log (B.supN j wL σ g) ≤ mx σ) ∧ (∑ σ, mx σ) / (Fintype.card (B.KO →+* ℂ) : ℝ) ≤ G * wL.size) ∧ (X.WWc S.sW wL ≤ B.Mlat j wL → Real.log N ≤ Prw * (wL.size − 1) → ∃ Gd : EtaleWeightGaps.GlobalFamilyData X ka wL wF, Gd.TiedW S G Prw)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ScaledThetaLatticeGeneratorsNoCommonZero

theorem **semilinear_eq_zero_of_span_smul** {O K L M ι : Type} [CommRing O] [Field K] [Field L] [Algebra O K] [AddCommGroup M] [Module K M] [Module O M] [IsScalarTower O K M] (σ : K →+* L) (φ : M →ₛₗ[σ] L) (w : ι → M) (hw : ∀ i, φ (w i) = 0) (s : O) (hs : algebraMap O K s ≠ 0) (f : M) (hf : s • f ∈ Submodule.span O (Set.range w)) : φ f = 0

theorem **eq_zero_of_span_eq_WWc** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} (X : PJ D B P j) (w : WeightFamilies.Wt D j) {ι : Type} (g : ι → B.Cusp j w) (s : 𝒪 B.KO) (hs : s ≠ 0) (hg : Submodule.span (𝒪 B.KO) (Set.range g) = X.WWc s w) (σ : B.KO →+* ℂ) (φ : B.Cusp j w → ℂ) (hadd : ∀ F F', φ (F + F') = φ F + φ F') (hsmul : ∀ (a : B.KO) F, φ (a • F) = σ a * φ F) (h0 : ∀ i, φ (g i) = 0) (F : B.Cusp j w) (hF : F ∈ X.thAll w) : φ F = 0

theorem **noCommonZero_of_tiedW** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T) (_EX : ErrX D) (j : ℕ) (hj : InR D j) (S : Scal (recB h.toLawsCore)) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (Gc Prw : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → S.sW ≠ 0 → ∀ G : EtaleWeightGaps.GlobalFamilyData (recPJ h j hj) ka wL wF, G.TiedW S Gc Prw → NoCommonZero (T.Q.Xo j) T.Q.c (fun i => (G.w i).1)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ScaledThetaLatticeGeneratorsNoCommonZeroStatement

theorem **_root_.GlobalTheory.T.scaledThetaLattice_generators_noCommonZero** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T) (_EX : ErrX D) (j : ℕ) (hj : InR D j) (S : Scal (recB h.toLawsCore)) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (Gc Prw : ℝ), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → S.sW ≠ 0 → ∀ G : EtaleWeightGaps.GlobalFamilyData (recPJ h j hj) ka wL wF, G.TiedW S Gc Prw → NoCommonZero (T.Q.Xo j) T.Q.c (fun i => (G.w i).1)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SecBCAndCuspBC

theorem **secBC_and_cuspBC** (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) : SecBC Xo hle ∧ ∀ (r : ℕ) (U : ∀ σ : c.F →+* ℂ, Unf Xo c σ r), CuspBC hle U

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SectionAtLevelPointEqTranslateByRepresentative

theorem **tyAct_natCast_mul** {E : Type} [Field E] [NumberField E] {g : ℕ} (ty : Fin g → (E →+* ℂ)) (n : ℕ) (a : 𝒪 E) (z : Fin g → ℂ) : tyAct ty ((n : 𝒪 E) * a) z = (n : ℂ) • tyAct ty a z

theorem **tyAct_natCast** {E : Type} [Field E] [NumberField E] {g : ℕ} (ty : Fin g → (E →+* ℂ)) (n : ℕ) (z : Fin g → ℂ) : tyAct ty (n : 𝒪 E) z = (n : ℂ) • z

theorem **hermForm_smul_left'** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (c : ℂ) (z w : Fin g → ℂ) : hermForm Hm (c • z) w = c * hermForm Hm z w

theorem **hermForm_two_right** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z w : Fin g → ℂ) : hermForm Hm z ((2 : ℂ) • w) = 2 * hermForm Hm z w

theorem **hermForm_add_left'** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (z z' w : Fin g → ℂ) : hermForm Hm (z + z') w = hermForm Hm z w + hermForm Hm z' w

theorem **e_natCast_smul** (n : ℕ) (z : Fin g → ℂ) : Φ.e ((n : ℂ) • z) = Φ.e z ^ n

theorem **sub_mem_of_e_eq** {z w : Fin g → ℂ} (h : Φ.e z = Φ.e w) : z − w ∈ Φ.lat

theorem **bar_bar** (a : 𝒪 E) : H.bar (H.bar a) = a

theorem **bar_mem** {r : 𝒪 E} (hr : r ∈ Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) : ∃ α ∈ l.Mb ^ 2, H.bar r = ((2 : ℕ) : 𝒪 E) * α

theorem **smul_eq_of_mem** {x2 : Dw.La} (hx : x2 ∈ (Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La)) {α : 𝒪 E} (hα : α ∈ l.Mb ^ 2) : ∃ w : Dw.La, α • x2 = ((l.M ^ 2 : ℕ) : 𝒪 E) • w

theorem **eq_two_smul_of_mem** {x2 : Dw.La} (hx : x2 ∈ (Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La)) : ∃ w : Dw.La, x2 = w + w

theorem **killedBy_iff** (I : Ideal (𝒪 E)) (z : Fin g → ℂ) : KilledBy Xo I (Φ.e z) ↔ ∀ α ∈ I, tyAct ty α z ∈ Φ.lat

theorem **levPt_eq** {x x2 : Dw.La} (h : x2 − x ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) : levPt Xo y.x x2 = levPt Xo y.x x

theorem **im_even** (mk : Dw.La →+ (Fin g → ℂ)) (hmk : ∀ a, mk a ∈ Φ.lat) (hlin : ∀ (α : 𝒪 E) a, mk (α • a) = tyAct ty α (mk a)) {u : Fin g → ℂ} (hu : ∀ α ∈ l.Mb ^ 2, tyAct ty α u ∈ Φ.lat) {x2 : Dw.La} (hx : x2 ∈ (Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La)) : ∃ k : ℤ, (hermForm Φ.Hm (mk x2) u).im = 2 * k

theorem **GlobalTheory.AnVal.rd_vt_levPt** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ mk : Dw.La →+ (Fin g → ℂ), (∀ a, mk a ∈ Φ.lat) → (∀ (α : 𝒪 E) a, mk (α • a) = tyAct ty α (mk a)) → (∀ a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La), levPt Xo y.x a = Φ.e ((l.M : ℂ)⁻¹ • mk a)) → ∀ θ, Φ.toAnFr.Rd 1 y.s θ → ∀ x x2 : Dw.La, x2 ∈ (Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La) → x2 − x ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La) → (KilledBy Xo (l.Mb ^ 2) (Φ.e (((l.M : ℂ) ^ 2)⁻¹ • mk x2)) ∧ Φ.e (((l.M : ℂ) ^ 2)⁻¹ • mk x2) ^ l.M = levPt Xo y.x x) ∧ ∀ F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y.x)), IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SectionAtLevelPointEqTranslateByRepresentativeStatement

theorem **_root_.GlobalTheory.T.section_at_level_point_eq_translate_by_representative** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ mk : Dw.La →+ (Fin g → ℂ), (∀ a, mk a ∈ Φ.lat) → (∀ (α : 𝒪 E) a, mk (α • a) = tyAct ty α (mk a)) → (∀ a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La), levPt Xo y.x a = Φ.e ((l.M : ℂ)⁻¹ • mk a)) → ∀ θ, Φ.toAnFr.Rd 1 y.s θ → ∀ x x2 : Dw.La, x2 ∈ (Ideal.span {(2 : 𝒪 E)} * (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La) → x2 − x ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La) → (KilledBy Xo (l.Mb ^ 2) (Φ.e (((l.M : ℂ) ^ 2)⁻¹ • mk x2)) ∧ Φ.e (((l.M : ℂ) ^ 2)⁻¹ • mk x2) ^ l.M = levPt Xo y.x x) ∧ ∀ F : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SectionReadingExistsUniqueIsTheta

theorem **isHolFam_vtfQ** {x0 : Sp ℂ ⟶ S} {g : ℕ} (e : (Fin g → ℂ) → (Over.mk x0 ⟶ A)) (hol : IsHolFam (fun z => (e z).left)) : IsHolFam (fun z => (vtfQ A (e z)).left)

theorem **AnFr.exists_chart_e** (z : Fin g → ℂ) : ∃ j, ⊤ ≤ (Φ.e z).left ⁻¹ᵁ L.L.U j

theorem **AnFr.rd_val** {k : ℤ} {F : LB.Sec (LPow A L.L k (Over.mk x0))} {θ : (Fin g → ℂ) → ℂ} (h : Φ.Rd k F θ) (j : L.L.ι) (z : Fin g → ℂ) (hj : ⊤ ≤ (Φ.e z).left ⁻¹ᵁ L.L.U j) : valAt' (vtfQ A (Φ.e z)).left ((LPow A L.L k (Over.mk x0)).U (fun _ => j)) (F.f (fun _ => j)) = θ z * Φ.fr j z ^ k

theorem **AnFr.rd_iff** (k : ℤ) (F : LB.Sec (LPow A L.L k (Over.mk x0))) (θ : (Fin g → ℂ) → ℂ) : Φ.Rd k F θ ↔ ∀ z, ∃ j, ⊤ ≤ (Φ.e z).left ⁻¹ᵁ L.L.U j ∧ valAt' (vtfQ A (Φ.e z)).left ((LPow A L.L k (Over.mk x0)).U (fun _ => j)) (F.f (fun _ => j)) = θ z * Φ.fr j z ^ k

theorem **AnFr.rd_eq_div** {k : ℤ} {F : LB.Sec (LPow A L.L k (Over.mk x0))} {θ : (Fin g → ℂ) → ℂ} (h : Φ.Rd k F θ) (j : L.L.ι) (z : Fin g → ℂ) (hj : ⊤ ≤ (Φ.e z).left ⁻¹ᵁ L.L.U j) : θ z = valAt' (vtfQ A (Φ.e z)).left ((LPow A L.L k (Over.mk x0)).U (fun _ => j)) (F.f (fun _ => j)) / Φ.fr j z ^ k

theorem **AnFr.existsUnique_rd** (k : ℤ) (F : LB.Sec (LPow A L.L k (Over.mk x0))) : ∃! θ, Φ.Rd k F θ

theorem **AnFr.e_add_lat** {a : Fin g → ℂ} (ha : a ∈ Φ.lat) (z : Fin g → ℂ) : Φ.e (z + a) = Φ.e z

theorem **AnFr.isTh_of_rd** {k : ℤ} {F : LB.Sec (LPow A L.L k (Over.mk x0))} {θ : (Fin g → ℂ) → ℂ} (h : Φ.Rd k F θ) : Φ.IsTh k θ

theorem **reading_existsUnique_isTh** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (F : LB.Sec (LPow A L.L k (Over.mk x0))), (∃! θ, Φ.Rd k F θ) ∧ (∀ θ, Φ.Rd k F θ → Φ.IsTh k θ) ∧ ∀ θ, Φ.Rd k F θ ↔ ∀ z, ∃ j, ⊤ ≤ (Φ.e z).left ⁻¹ᵁ L.L.U j ∧ valAt' (ptSecT A (Φ.e z)).left ((LPow A L.L k (Over.mk x0)).U (fun _ => j)) (F.f (fun _ => j)) = θ z * Φ.fr j z ^ k

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SectionReadingExistsUniqueIsThetaStatement

theorem **_root_.GlobalTheory.T.section_reading_existsUnique_isTheta** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (F : LB.Sec (LPow A L.L k (Over.mk x0))), (∃! θ, Φ.Rd k F θ) ∧ (∀ θ, Φ.Rd k F θ → Φ.IsTh k θ) ∧ ∀ θ, Φ.Rd k F θ ↔ ∀ z, ∃ j, ⊤ ≤ (Φ.e z).left ⁻¹ᵁ L.L.U j ∧ valAt' (ptSecT A (Φ.e z)).left ((LPow A L.L k (Over.mk x0)).U (fun _ => j)) (F.f (fun _ => j)) = θ z * Φ.fr j z ^ k

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShareRdOfInBasis

theorem **addL_zero_wt** (u wL : WeightFamilies.Wt D j) : addL u wL 0 = u

theorem **rk_Mlat_eq_finrank** (X : PJ D B P j) (ka : WeightFamilies.Wt D j) (h : LatAt B j ka) : X.rk ka (B.Mlat j ka) = Module.finrank B.KO (⊤ : Submodule B.KO (B.Cusp j ka))

private lemma **s_mem_sp_univ** (R : Rdg X) {N : ℕ} {wL : WeightFamilies.Wt D j} (w : Fin N → B.Cusp j wL) (t : Idx D j N → ℂ) (i : Fin (WeightFamilies.dimY D j + 1)) : R.s w t i ∈ R.sp wL Set.univ

theorem **Rdg.s_pow_mul_mem** (R : Rdg X) {N : ℕ} {wL : WeightFamilies.Wt D j} (w : Fin N → B.Cusp j wL) (t : Idx D j N → ℂ) (hmul : ∀ u u', ∀ f ∈ R.sp u Set.univ, ∀ g ∈ R.sp u' Set.univ, f * g ∈ R.sp (wadd u u') Set.univ) (i : Fin (WeightFamilies.dimY D j + 1)) (u : WeightFamilies.Wt D j) (n m : ℕ) {v : R.Y → ℂ} (hv : v ∈ R.sp (addL u wL n) Set.univ) : R.s w t i ^ m * v ∈ R.sp (addL u wL (n + m)) Set.univ

theorem **Rdg.prod_pow_mul_mem** (R : Rdg X) {N : ℕ} {wL : WeightFamilies.Wt D j} (w : Fin N → B.Cusp j wL) (t : Idx D j N → ℂ) (hmul : ∀ u u', ∀ f ∈ R.sp u Set.univ, ∀ g ∈ R.sp u' Set.univ, f * g ∈ R.sp (wadd u u') Set.univ) (α : Fin (WeightFamilies.dimY D j) → ℕ) (s : Finset (Fin (WeightFamilies.dimY D j))) (u : WeightFamilies.Wt D j) (n : ℕ) {v : R.Y → ℂ} (hv : v ∈ R.sp (addL u wL n) Set.univ) : (∏ i ∈ s, R.s w t i.succ ^ α i) * v ∈ R.sp (addL u wL (n + ∑ i ∈ s, α i)) Set.univ

theorem **shareRd_of_inBasis** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (G : GlobalFamilyData X ka wL wF) (R : Rdg X) (t : Idx D j G.N → ℂ) (T : ℕ) (Mg : Submodule B.KO (B.Cusp j ka)) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t → AnFacts R G.w t ka wF T Mg xs → addL wF wL T = ka → WeightFamilies.dimY D j < T → 0 < WeightFamilies.dimY D j → LatAt B j ka → (∃ la, InBasis G la) → ShareRd (D

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShareRhoAndShareRhoTOfShareRhoR

theorem **shareRho_and_shareRhoT_of_shareRhoR** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (G : GlobalFamilyData X ka wL wF) (S : Scal B) (T d : ℕ) (Gc : ℝ), ShareRhoR G S T d → G.TiedZ S T Gc → (∀ la, InContainZ G la S T) → ShareRho (D

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShareRhoROfAnFacts

theorem **finite_sum_le** (D a : ℕ) : {α : Fin D → ℕ | ∑ i, α i ≤ a}.Finite

theorem **graded_prod_pow_mul** {A ι : Type} [CommRing A] (Q : ℕ → A → Prop) (s : ι → A) (h : ∀ n i x, Q n x → Q (n + 1) (s i * x)) (α : ι → ℕ) (F : Finset ι) (n : ℕ) (x : A) (hx : Q n x) : Q (n + ∑ i ∈ F, α i) ((∏ i ∈ F, s i ^ α i) * x)

theorem **addL_zero'** {D : D7} {j : ℕ} (u wL : WeightFamilies.Wt D j) : addL u wL 0 = u

theorem **shareRhoR_of_anFacts** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (ka wL wF : WeightFamilies.Wt D j) (G : GlobalFamilyData X ka wL wF) (R : Rdg X) (t : Idx D j G.N → ℂ) (T : ℕ) (Mg : Submodule B.KO (B.Cusp j ka)) (δ : ℕ) (xs : Fin δ → R.Y), R.Generic t → AnFacts R G.w t ka wF T Mg xs → addL wF wL T = ka → 2 ≤ WeightFamilies.dimY D j → 11 * WeightFamilies.dimY D j ^ 2 ≤ T → LatAt B j ka → ∀ S : Scal B, ShareRhoR G S T (X.rk ka (B.Mlat j ka))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortGeneratorMemSpanSmallSupNormShortGenerators

theorem **fg_of_trivAt_even** {E : Type} [Field E] [NumberField E] {n : ℕ} (Λ : GenLat E n) {bad : ℕ} (h2 : 2 ∣ bad) (h : Λ.TrivAt bad) : Λ.La.FG

theorem **sh_mem_span_small** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → LawsSh T → (T.dδ j w.n).l₂.A = 1 → ∀ g ∈ T.sh j w, g ∈ Submodule.span T.Q.c.F {g' : T.R j | g' ∈ T.sh j w ∧ ∀ σ, Real.log (T.Ns j w σ g') ≤ 400 * Zb D large₂ * w.size}

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortGeneratorMemSpanSmallSupNormShortGeneratorsStatement

theorem **_root_.GlobalTheory.T.shortGenerator_mem_span_small_supNorm_shortGenerators** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → ∀ (large₂ : SizeData₂ D) (j : ℕ) (hj : InR D j) (w : WeightFamilies.Wt D j), ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ w → 2 ≤ large₂.sc.L → D.R.r ≤ w.n → w.LastPos → LawsSh T → (T.dδ j w.n).l₂.A = 1 → ∀ g ∈ T.sh j w, g ∈ Submodule.span T.Q.c.F {g' : T.R j | g' ∈ T.sh j w ∧ ∀ σ, Real.log (T.Ns j w σ g') ≤ 400 * Zb D large₂ * w.size}

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortGeneratorsScaledThetaLattice

theorem **span_image_smul_top** {K V : Type*} [Field K] [AddCommGroup V] [Module K V] (c : K) (hc : c ≠ 0) (S : Set V) (hS : Submodule.span K S = ⊤) : Submodule.span K ((fun g => c • g) '' S) = ⊤

theorem **shortG_scaled** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (w : WeightFamilies.Wt D j) (s : 𝒪 B.KO) (Sh : Finset (B.Cusp j w)) (a b : ℝ), s ≠ 0 → 0 ≤ a → 0 ≤ b → (∀ σ : B.KO →+* ℂ, Real.log ‖σ (s : B.KO)‖ ≤ a) → (∀ g ∈ Sh, g ∈ X.shortTh w ∧ ∀ σ, Real.log (B.supN j w σ g) ≤ b) → Submodule.span B.KO (Sh : Set (B.Cusp j w)) = ⊤ → ShortG X w (X.Thc s w) (a + b + lamStar D * w.size)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortGeneratorsScaledThetaLatticeStatement

theorem **_root_.GlobalTheory.T.shortGenerators_scaled_thetaLattice** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (w : WeightFamilies.Wt D j) (s : 𝒪 B.KO) (Sh : Finset (B.Cusp j w)) (a b : ℝ), s ≠ 0 → 0 ≤ a → 0 ≤ b → (∀ σ : B.KO →+* ℂ, Real.log ‖σ (s : B.KO)‖ ≤ a) → (∀ g ∈ Sh, g ∈ X.shortTh w ∧ ∀ σ, Real.log (B.supN j w σ g) ≤ b) → Submodule.span B.KO (Sh : Set (B.Cusp j w)) = ⊤ → ShortG X w (X.Thc s w) (a + b + lamStar D * w.size)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortSeriesValueTwistedEqConjugateValue

theorem **colSum_eq_of_isNab** (hC : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2) {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab nab' : NabV Xo c) (h : IsNab Xo c nab) (h' : IsNab Xo c nab') {n : ℕ} (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (K : ℤ) (k : ℕ) (hK : K = (k : ℤ)) (G : Fin n → LB.Sec (ReciprocityTheory.ChartRestriction.G.Out.LPow Xo.univ.t.G Xo.univ.t.PP.L K (ptO Xo y.x))) : ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab y K (G b) (colOf m b) = ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) * ∏ b, nab' y K (G b) (colOf m b)

theorem **coeffAutFixesCMConjugatesAndBase_symm** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {l : Level H} {c : Coeff l} {ta : c.F ≃+* c.F} (h : Fixes c ta) : Fixes c ta.symm

theorem **algebraMap_tw_symm** {K : Type} [Field K] (ta : K ≃+* K) (S : Type) [CommRing S] [Algebra K S] (x : K) : (algebraMap K (Tw ta.symm S) (ta x)).val = algebraMap K S x

theorem **shortValAt_conj** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (ta : c.F ≃+* c.F) (h : Fixes c ta), ∀ [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ), ∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), ∀ (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (b : c.F) (N : ℕ) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : Tw ta S), ShortValAt Xo c nab w Λ b N P (twPt h y) v → ShortValAt Xo c nab (fun q => ta (w q)) Λ (ta b) N (MvPolynomial.map (ta : c.F →+* c.F) P) y v.val

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortSeriesValueTwistedEqConjugateValueStatement

theorem **_root_.GlobalTheory.T.shortSeriesValue_twisted_eq_conjugate_value** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (ta : c.F ≃+* c.F) (h : Fixes c ta), ∀ [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ), ∀ P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)), ∀ (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (b : c.F) (N : ℕ) (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : Tw ta S), ShortValAt Xo c nab w Λ b N P (twPt h y) v → ShortValAt Xo c nab (fun q => ta (w q)) Λ (ta b) N (MvPolynomial.map (ta : c.F →+* c.F) P) y v.val

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaRuleLinearInHarmonicPolynomial

theorem **map_mem_HHa** {F F₂ : IntermediateField ℚ ℂ} (h : F = F₂) {n : ℕ} {a : (E →+* ℂ) × ℕ → ℕ} {P : MvPolynomial (LtN Dw n) F} (hP : P ∈ HHa Dw.mult n F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) : MvPolynomial.map (IntermediateField.equivOfEq h : F →+* F₂) P ∈ HHa Dw.mult n F₂ ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))

theorem **eq_sum_of_isShortRule** (hC : ChartIndepSt2) (Xo : Moduli Dw l) (c : Coeff l) {n : ℕ} (d : DeltaDat Xo c n) (hd : d.IsLev) [StarRing (𝒪 E)] (hn : IsNab d.Xo₂ d.c₂ d.nab₂) (a : (E →+* ℂ) × ℕ → ℕ) (Λ : GenLat E n) (N : ℕ) (x0 : Fin n → Dw.La) (hU : Λ.Unimod) (hT : Λ.TrivAt d.l₂.M) (hL : Λ.LevOK N d.l₂.M) (m : ℕ) (cf : Fin m → c.F) (Q : Fin m → MvPolynomial (LtN Dw n) c.F) (g : Rul Xo c) (gs : Fin m → Rul Xo c) (hQ : ∀ i, Q i ∈ harmL c Λ a) (hg : g ∈ sec Xo c a n) (hgs : ∀ i, gs i ∈ sec Xo c a n) (hr : IsShortRule Xo c d Λ N x0 (∑ i, cf i • Q i) g) (hrs : ∀ i, IsShortRule Xo c d Λ N x0 (Q i) (gs i)) : g = ∑ i, cf i • gs i

theorem **shortRule_linear** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T), LawsSh T → ∀ (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j) (Λ : GenLat D.host.E w.n) (N : ℕ) (x0 : Fin w.n → (T.Q.Dw j).La), Λ.Unimod → Λ.TrivAt (T.dδ j w.n).l₂.M → Λ.LevOK N (T.dδ j w.n).l₂.M → ∀ (m : ℕ) (cf : Fin m → T.Q.c.F) (Q : Fin m → MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F) (g : T.R j) (gs : Fin m → T.R j), (∀ i, Q i ∈ harmL T.Q.c Λ (wtA w)) → g ∈ sec (T.Q.Xo j) T.Q.c (wtA w) w.n → (∀ i, gs i ∈ sec (T.Q.Xo j) T.Q.c (wtA w) w.n) → IsShortRule (T.Q.Xo j) T.Q.c (T.dδ j w.n) Λ N x0 (∑ i, cf i • Q i) g → (∀ i, IsShortRule (T.Q.Xo j) T.Q.c (T.dδ j w.n) Λ N x0 (Q i) (gs i)) → g = ∑ i, cf i • gs i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaRuleLinearInHarmonicPolynomialStatement

theorem **_root_.GlobalTheory.T.shortThetaRule_linear_in_harmonic_polynomial** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (_h : LawsP T), LawsSh T → ∀ (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j) (Λ : GenLat D.host.E w.n) (N : ℕ) (x0 : Fin w.n → (T.Q.Dw j).La), Λ.Unimod → Λ.TrivAt (T.dδ j w.n).l₂.M → Λ.LevOK N (T.dδ j w.n).l₂.M → ∀ (m : ℕ) (cf : Fin m → T.Q.c.F) (Q : Fin m → MvPolynomial (LtN (T.Q.Dw j) w.n) T.Q.c.F) (g : T.R j) (gs : Fin m → T.R j), (∀ i, Q i ∈ harmL T.Q.c Λ (wtA w)) → g ∈ sec (T.Q.Xo j) T.Q.c (wtA w) w.n → (∀ i, gs i ∈ sec (T.Q.Xo j) T.Q.c (wtA w) w.n) → IsShortRule (T.Q.Xo j) T.Q.c (T.dδ j w.n) Λ N x0 (∑ i, cf i • Q i) g → (∀ i, IsShortRule (T.Q.Xo j) T.Q.c (T.dδ j w.n) Λ N x0 (Q i) (gs i)) → g = ∑ i, cf i • gs i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaSumEqOfWitnessesCommonAlgebra

theorem **exists_pull** (hC : ChartIndepSt2) [StarRing (𝒪 E)] (nab : NabV Xo c) (hnab : IsNab Xo c nab) {n : ℕ} (Λ : GenLat E n) (N : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S) (w : FrPt Xo c T) (hm : FrMap ψ y w) (z : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (tp : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y.x ⟶ Xo.univ.t.G)) (F : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) (hz : ∀ i, InZh Xo Λ N (z i)) (hg : GeoInj Xo y.x z) (ht : ∀ q b, tp q b ^ N = levPt Xo y.x (levRep (q b)) ∧ ∃ j : ℕ, KilledBy Xo (l.Mb ^ j) (tp q b)) (hF : ∀ q i b, IsUUOver Xo.univ.t.G Xo.univ.t.PP N y.

theorem **eq_of_local** {K T : Type} [Field K] [CommRing T] [Algebra K T] (u v : T) (h : ∀ (U : Type) [CommRing U] [IsLocalRing U] [Algebra K U] (κ : T →ₐ[K] U), κ u = κ v) : u = v

theorem **shortSum_eq_of_witnesses** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (S1 : Type) [CommRing S1] [Algebra c.F S1] (φ1 : S →ₐ[c.F] S1) (y1 : FrPt Xo c S1) (z1 : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y1.x ⟶ Xo.univ.t.G)) (tp1 : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y1.x ⟶ Xo.univ.t.G)) (F1 : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y1.x))) (S2 : Type) [Comm

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaSumEqOfWitnessesCommonAlgebraStatement

theorem **_root_.GlobalTheory.T.shortThetaSum_eq_of_witnesses_common_algebra** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (S1 : Type) [CommRing S1] [Algebra c.F S1] (φ1 : S →ₐ[c.F] S1) (y1 : FrPt Xo c S1) (z1 : Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → (ptO Xo y1.x ⟶ Xo.univ.t.G)) (tp1 : (Fin n → LevQ Dw l) → Fin n → (ptO Xo y1.x ⟶ Xo.univ.t.G)) (F1 : (Fin n → LevQ Dw l) → Fin (Λ.zCard N (Dw.p + Dw.q)) → Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N :

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaValueExistsUniqueNaturalOfLaws

theorem **finite_levelQuotient** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} (Dw : Datum H) (l : Geo.Tor.Level H) : Finite (LevQ Dw l)

theorem **map_weighted_finsum** {F S₁ S₂ T Q : Type} [Field F] [CommRing S₁] [CommRing S₂] [CommRing T] [Algebra F S₁] [Algebra F S₂] [Algebra F T] [Finite Q] (ψ₁ : S₁ →ₐ[F] T) (ψ₂ : S₂ →ₐ[F] T) (k : F) (w : Q → F) (A₁ : Q → S₁) (A₂ : Q → S₂) (h : ∀ q, ψ₁ (A₁ q) = ψ₂ (A₂ q)) : ψ₁ (algebraMap F S₁ k * ∑ᶠ q, algebraMap F S₁ (w q) * A₁ q) = ψ₂ (algebraMap F S₂ k * ∑ᶠ q, algebraMap F S₂ (w q) * A₂ q)

theorem **shortThetaValue_existsUnique_natural** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (w : (Fin n → LevQ Dw l) → c.F) (β : c.F), (∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), ∃! v, ShortValAt Xo c nab w Λ β N P y v) ∧ ∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y0 : FrPt Xo c S) (y : FrPt Xo c T), FrMap φ y0 y → ∀ v0 v, ShortValAt Xo c nab w Λ β N P y0 v0 → ShortValAt Xo c nab w Λ β N P y v → φ v0 = v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaValueExistsUniqueNaturalOfLawsStatement

theorem **_root_.GlobalTheory.T.shortThetaValue_existsUnique_natural_of_laws** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (w : (Fin n → LevQ Dw l) → c.F) (β : c.F), (∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), ∃! v, ShortValAt Xo c nab w Λ β N P y v) ∧ ∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y0 : FrPt Xo c S) (y : FrPt Xo c T), FrMap φ y0 y → ∀ v0 v, ShortValAt Xo c nab w Λ β N P y0 v0 → ShortValAt Xo c nab w Λ β N P y v → φ v0 = v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaValueLinearInPolynomial

theorem **isHomogeneous_of_rowDeg** {T : Type} [Fintype T] {r : T → ℕ} {n : ℕ} {k : Type} [Field k] {a : T → ℕ → ℕ} {P : MvPolynomial (Letter T r n) k} (hP : RowDeg r n k a P) : P.IsHomogeneous (∑ σ : T, ∑ i : Fin (r σ), a σ i)

theorem **sum_support_eq_constr** {ι F S : Type} [Field F] [CommRing S] [Algebra F S] (G : (ι →₀ ℕ) → S) (P : MvPolynomial ι F) : ∑ m ∈ P.support, algebraMap F S (P.coeff m) * G m = (MvPolynomial.basisMonomials ι F).constr F G P

theorem **formula_linear** {ι F S Qt J : Type} [Field F] [CommRing S] [Algebra F S] [Finite Qt] [Fintype J] (k N : F) (w : Qt → F) (G : Qt → J → (ι →₀ ℕ) → S) (d m : ℕ) (cf : Fin m → F) (Q : Fin m → MvPolynomial ι F) (hQ : ∀ i, (Q i).IsHomogeneous d) : algebraMap F S (k * (N ^ (∑ i, cf i • Q i).totalDegree)⁻¹) * ∑ᶠ q, algebraMap F S (w q) * ∑ j, ∑ μ ∈ (∑ i, cf i • Q i).support, algebraMap F S ((∑ i, cf i • Q i).coeff μ) * G q j μ = ∑ i, algebraMap F S (cf i) * (algebraMap F S (k * (N ^ (Q i).totalDegree)⁻¹) * ∑ᶠ q, algebraMap F S (w q) * ∑ j, ∑ μ ∈ (Q i).support, algebraMap F S ((Q i).coeff μ) * G q j μ)

theorem **shortThetaValue_linear** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (w : (Fin n → LevQ Dw l) → c.F) (β : c.F) (m : ℕ) (cf : Fin m → c.F) (Q : Fin m → MvPolynomial (LtN Dw n) c.F), (∀ i, Q i ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S) (u : Fin m → S), ShortValAt Xo c nab w Λ β N (∑ i, cf i • Q i) y v → (∀ i, ShortValAt Xo c nab w Λ β N (Q i) y (u i)) → v = ∑ i, algebraMap c.F S (cf i) * u i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaValueLinearInPolynomialStatement

theorem **_root_.GlobalTheory.T.short_theta_value_linear_in_polynomial** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → Λ.TrivAt l.M → Λ.LevOK N l.M → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (w : (Fin n → LevQ Dw l) → c.F) (β : c.F) (m : ℕ) (cf : Fin m → c.F) (Q : Fin m → MvPolynomial (LtN Dw n) c.F), (∀ i, Q i ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (v : S) (u : Fin m → S), ShortValAt Xo c nab w Λ β N (∑ i, cf i • Q i) y v → (∀ i, ShortValAt Xo c nab w Λ β N (Q i) y (u i)) → v = ∑ i, algebraMap c.F S (cf i) * u i

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaValueUnderLevelMove

theorem **secRel_of_charts** (hL₀ : L₀.IsLB) (hL₁ : L₁.IsLB) (h₂ : Ψ ≫ (snd T₀ G).left = (snd T A).left ≫ φ) (s₀ : LB.Sec (LPow G L₀ 1 T₀)) (s : LB.Sec (LPow A L₁ 1 T)) (H : ∀ (i : L₀.ι) (j : L₁.ι) (W : (T ⊗ A).left.Opens), W = (snd T A).left ⁻¹ᵁ ((LB.div (L₀.pull φ) L₁).U ((fun _ => i), j)) → ∀ (h₀ : W ≤ (snd T A).left ⁻¹ᵁ ((LB.div (L₀.pull φ) L₁).U ((fun _ => i), j))) (h₁ : W ≤ Ψ ⁻¹ᵁ (LPow G L₀ 1 T₀).U (fun _ => i)) (h₂ : W ≤ (LPow A L₁ 1 T).U (fun _ => j)), (Ψ.appLE ((LPow G L₀ 1 T₀).U (fun _ => i)) W h₁).hom (s₀.f (fun _ => i)) = ((snd T A).left.appLE ((LB.div (L₀.pull φ) L₁).U ((fun _ => i), j)) W h₀).hom (Units.val (w.s ((fun _ => i), j))) * ((T ⊗ A).left.presheaf.map (homOfLE h₂).op).hom (s.f (fun _ => j))) : SecRel w Ψ s₀ s

theorem **sv_pull** {S T : Type} [CommRing S] [CommRing T] {x : Sp S ⟶ Xo.X} {x' : Sp T ⟶ Xo.X} (pm : Gx x' ⟶ Gx x) (hs : pm ≫ (snd (Over.mk x) Xo.univ.t.G).left = (snd (Over.mk x') Xo.univ.t.G).left) {s : LB.Sec (PPx x)} {s' : LB.Sec (PPx x')} (ha : ∀ (i : Fin 1 → Xo.univ.t.PP.L.ι) (h : (PPx x').U i ≤ pm ⁻¹ᵁ (PPx x).U i), (pm.appLE _ _ h).hom (s.f i) = s'.f i) {Z : Scheme.{0}} (r : Z ⟶ Gx x') (z : Z) {J} (hJ : Good Xo.univ.t.PP.L z J) : sv Xo.univ.t.G s (r ≫ pm) z J = sv Xo.univ.t.G s' r z J

theorem **exists_ptEquiv** {S : Type} [CommRing S] {ρ : Xo.X ⟶ Xo.X} (ν : SABMap Xo.univ.t Xo.univ.t ρ) {x x' : Sp S ⟶ Xo.X} (hx : x' = x ≫ ρ) : ∃ e : (ptO Xo x ⟶ Xo.univ.t.G) ≃* (ptO Xo x' ⟶ Xo.univ.t.G), ∀ p, (e p).left = p.left ≫ ν.φ

theorem **exists_move_baseChange** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] {ρ : Xo.X ⟶ Xo.X} {ν : SABMap Xo.univ.t Xo.univ.t ρ} {y y' : FrPt Xo c S} (hm : FrMove ρ ν y y') {φ : S →ₐ[c.F] T} {z z' : FrPt Xo c T} (h : FrMap φ y z) (h' : FrMap φ y' z') : z'.x = z.x ≫ ρ ∧ (∀ τ i, z'.u τ i = z.u τ i ≫ ν.φ) ∧ ∃ (pm : Gx z.x ⟶ Gx z'.x) (w : LB.Iso Xo.univ.t.PP.L (Xo.univ.t.PP.L.pull ν.cm.F)), pm ≫ (fst (Over.mk z'.x) Xo.univ.t.G).left = (fst (Over.mk z.x) Xo.univ.t.G).left ∧ pm ≫ (snd (Over.mk z'.x) Xo.univ.t.G).left = (snd (Over.mk z.x) Xo.univ.t.G).left ≫ ν.φ ∧ CubIso ν w ∧ SecRel (T

theorem **eta_congr** {a b : Dw.La} (h : (Submodule.Quotient.mk a : LevQ Dw l) = Submodule.Quotient.mk b) : Xo.univ.eta a = Xo.univ.eta b

theorem **mk_levRep** (q : LevQ Dw l) : (Submodule.Quotient.mk (levRep q) : LevQ Dw l) = q

theorem **shortValAt_move** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (ρ : Xo.X ⟶ Xo.X) (ν : SABMap Xo.univ.t Xo.univ.t ρ) (gM gA : Dw.La →ₗ[𝒪 E] Dw.La) (y y' : FrPt Xo c S), IsLevMove ρ ν gM gA → FrMove ρ ν y y' → ∀ σq : LevQ Dw l ≃ LevQ Dw l, (∀ a, σq (Submodule.Quotient.mk a) = Submodule.Quotient.mk (gM a)) → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (β : c.F) (N : ℕ) (v : S), ShortValAt Xo c nab w Λ β N P y' v → ShortValAt Xo c nab (fun q => w (fun b => σq.symm (q b))) Λ β N P y v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortThetaValueUnderLevelMoveStatement

theorem **_root_.GlobalTheory.T.shortThetaValue_under_levelMove** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (ρ : Xo.X ⟶ Xo.X) (ν : SABMap Xo.univ.t Xo.univ.t ρ) (gM gA : Dw.La →ₗ[𝒪 E] Dw.La) (y y' : FrPt Xo c S), IsLevMove ρ ν gM gA → FrMove ρ ν y y' → ∀ σq : LevQ Dw l ≃ LevQ Dw l, (∀ a, σq (Submodule.Quotient.mk a) = Submodule.Quotient.mk (gM a)) → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (β : c.F) (N : ℕ) (v : S), ShortValAt Xo c nab w Λ β N P y' v → ShortValAt Xo c nab (fun q => w (fun b => σq.symm (q b))) Λ β N P y v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortValAtBc

theorem **shortExpr_bc** {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c c' : Coeff l} (hle : c.F ≤ c'.F) (nab : NabV Xo c) {n : ℕ} (w : (Fin n → LevQ Dw l) → c.F) (β : c.F) (a b N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) {S : Type} [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S) {J : Type} [Fintype J] (k : ℤ) (F : (Fin n → LevQ Dw l) → J → Fin n → LB.Sec (Out.LPow Xo.univ.t.G Xo.univ.t.PP.L k (ptO Xo y.x))) : algebraMap c'.F S ((incF c c' hle β)⁻¹ * ((a : c'.F))⁻¹ * ((b : c'.F))⁻¹ * ((N : c'.F) ^ (MvPolynomial.map (incF c c' hle) P).totalDegree)⁻¹) * ∑ᶠ q : Fin n → LevQ Dw l, algebraMap c'.F S (incF c c' hle (w q)) * ∑ i, ∑ m ∈ (MvPolynomial.map (incF c c' hle) P).support, algebraMap c'.F S ((MvPolynomial.map (incF c c' hle) P).coeff m) * ∏ j, nabBC hle nab y k (F q i j) (colOf m j) =

theorem **shortValAt_bc** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c c' : Coeff l) (hle : c.F ≤ c'.F) (nab : NabV Xo c) (n : ℕ) (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (β : c.F) (N : ℕ) (P : MvPolynomial (LtN Dw n) c.F) (S : Type) [CommRing S] [Algebra c.F S] [Algebra c'.F S] (h : AlgOver c c' hle S) (y : FrPt Xo c' S) (v : S), ShortValAt Xo c' (nabBC hle nab) (fun q => incF c c' hle (w q)) Λ (incF c c' hle β) N (MvPolynomial.map (incF c c' hle) P) y v ↔ ShortValAt Xo c nab w Λ β N P (y.res hle h) v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ShortValTotOfPinned

theorem **evR_eq_of_point_eq** (d : DeltaDat Xo c n) {ψ : (LaI Xo c d ≃ₗ[𝒪 E] LaI Xo c d) →* c.F^×} (hψ : d.PsiTie ψ) {k : Type} [Field k] [IsAlgClosed k] [Algebra d.c₂.F k] (y y' : FrPt d.Xo₂ d.c₂ k) (h : y'.x = y.x) : evR d.Xo₂ d.c₂ d.cδ y' = evR d.Xo₂ d.c₂ d.cδ y

theorem **map_wchar** {J : Type} [Fintype J] {K S T : Type} [Field K] [CommRing S] [Algebra K S] [CommRing T] [Algebra K T] (χ : S →ₐ[K] T) (e : J → ℕ) (t : J → Sˣ) : χ (wchar e t) = wchar e (fun i => Units.map (χ : S →* T) (t i))

theorem **shortValTot_of_pinned** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (n : ℕ) (d : DeltaDat Xo c n), @ToroidalLevelCompatibility E _ _ _ H Dw → EvalWD (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SingularMetricFreeDivisorInvariantLowerBound

theorem **Wfree_of_mem** {x : Frame X.Ψ r} (hx : x ∈ X.dom) : Wfree Xa Fl G n x = ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2))

theorem **Wfree_of_notMem** {x : Frame X.Ψ r} (hx : x ∉ X.dom) : Wfree Xa Fl G n x = 0

private lemma **sum_norm_sq_pos_of_exists_ne_zero** {N : ℕ} {a : Fin N → ℂ} (h : ∃ k, a k ≠ 0) : 0 < ∑ k, ‖a k‖ ^ 2

theorem **Wfree_eq_exp** {x : Frame X.Ψ r} (hx : x ∈ X.dom) (hs : ∃ k, Fl.s k x ≠ 0) : Wfree Xa Fl G n x = ENNReal.ofReal (Real.exp (-(phla X G x + (1 / 2) * Real.log (∑ k, ‖Xa.w k x‖ ^ 2) + ((n : ℝ) - 1 / 2) * Real.log (∑ k, ‖Fl.s k x‖ ^ 2))))

theorem **Wfree_eq_pnorm** {x : Frame X.Ψ r} (hx : x ∈ X.dom) : Wfree Xa Fl G n x = ENNReal.ofReal (Real.exp (-phla X (G + n • Xa.L) x)) * ENNReal.ofReal (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, X.pnorm Xa.L (Fl.s k) x ^ 2) ^ (-((n : ℝ) - 1 / 2))

private lemma **isBddSec_of_mem_SSgt** {la : Wt X.Ψ r} {c : ℝ} {P : Frame X.Ψ r → ℂ} (h : P ∈ X.SSgt la c) : X.IsBddSec la P

theorem **measurable_Wfree** : Measurable (Wfree Xa Fl G n)

theorem **Wfree_rmul** (hN : NfacRmulStatement (E

theorem **Wfree_lact** (hN : NfacLactStatement (E

theorem **Wfree_lower** (hn : 1 ≤ n) : WLower X (G + n • Xa.L) (Wfree Xa Fl G n)

theorem **freeDivMetric** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (n : ℕ), 1 ≤ n → IsMetricW X (G + n • Xa.L) (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2)) else 0) ∧ WLower X (G + n • Xa.L) (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑

k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2)) else 0) ∧ ∀ x ∈ X.dom, (∃ k, Fl.s k x ≠ 0) → (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2)) else 0) x = ENN

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SingularMetricFreeDivisorInvariantLowerBoundStatement

theorem **_root_.GlobalTheory.T.singularMetricFreeDivisor_invariant_lower_bound** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (n : ℕ), 1 ≤ n → IsMetricW X (G + n • Xa.L) (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2)) else 0) ∧ WLower X (G + n • Xa.L) (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2)) else 0) ∧ ∀ x ∈ X.dom, (∃ k, Fl.s k x ≠ 0) → (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SingularMetricFreeDivisorSmoothCurvature

theorem **divLevi_freeWeight** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (n : ℕ), 1 ≤ n → ∀ x0, IsOrth X x0 → (∃ k0, ∃ z ∈ OmF X x0, Fl.s k0 (cpt X x0 z) ≠ 0) ∧ ∀ k0 : Fin (nd X), ContDiffOn ℝ ∞ (fun z => (fun x => phla X G x + (1 / 2) * Real.log (∑ k, ‖Xa.w k x‖ ^ 2) + ((n : ℝ) - 1 / 2) * Real.log (∑ k, ‖Fl.s k x‖ ^ 2)) (cpt X x0 z)) {z | z ∈ OmF X x0 ∧ Fl.s k0 (cpt X x0 z) ≠ 0} ∧ ∀ z ∈ OmF X x0, Fl.s k0 (cpt X x0 z) ≠ 0 → ∀ ξ : Fin (nd X) → ℂ, leviO X x0 (phla X G) z ξ - leviDens X x0 z ξ ≤ (leviV (nd X) (fun w => (((fun x => phla X G x + (1 / 2) * Real.log (∑ k, ‖Xa.w k x‖ ^ 2) + ((n : ℝ) - 1 / 2) * Real.log (∑ k, ‖Fl.s k x‖ ^ 2)) (cpt X x0 w) - Real.log (densF X x0 w) : ℝ) : ℂ)) z ξ).re

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SingularMetricFreeDivisorSmoothCurvatureStatement

theorem **_root_.GlobalTheory.T.singularMetricFreeDivisor_smooth_curvature** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (n : ℕ), 1 ≤ n → ∀ x0, IsOrth X x0 → (∃ k0, ∃ z ∈ OmF X x0, Fl.s k0 (cpt X x0 z) ≠ 0) ∧ ∀ k0 : Fin (nd X), ContDiffOn ℝ ∞ (fun z => (fun x => phla X G x + (1 / 2) * Real.log (∑ k, ‖Xa.w k x‖ ^ 2) + ((n : ℝ) - 1 / 2) * Real.log (∑ k, ‖Fl.s k x‖ ^ 2)) (cpt X x0 z)) {z | z ∈ OmF X x0 ∧ Fl.s k0 (cpt X x0 z) ≠ 0} ∧ ∀ z ∈ OmF X x0, Fl.s k0 (cpt X x0 z) ≠ 0 → ∀ ξ : Fin (nd X) → ℂ, leviO X x0 (phla X G) z ξ - leviDens X x0 z ξ ≤ (leviV (nd X) (fun w => (((fun x => phla X G x + (1 / 2) * Real.log (∑ k, ‖Xa.w k x‖ ^ 2) + ((n : ℝ) - 1 / 2) * Real.log (∑ k, ‖Fl.s k x‖ ^ 2)) (cpt X x0 w) - Real.log (densF X x0 w) : ℝ) : ℂ)) z ξ).re

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SizePageRowsOfSizePageAt

theorem **pageRows_of_pageAt** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → PageAt large₂ C2 A2 → 4 * H.CK ≤ WeightFamilies.PPv D large₂ C2 A2 ^ 2 * D.R.f → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka ∧ 16 ≤ D.R.f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SizePageRowsOfSizePageAtStatement

theorem **_root_.GlobalTheory.T.sizePageRows_of_sizePageAt** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → PageAt large₂ C2 A2 → 4 * H.CK ≤ WeightFamilies.PPv D large₂ C2 A2 ^ 2 * D.R.f → ReciprocityTheory.ChartRestriction.G.PullbackSections.SupBoundPageRows D large₂ ka ∧ 16 ≤ D.R.f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.Sl2NonnegWeightOfHighestWeightVector

theorem **casimir_on_highest** {K V : Type*} [CommRing K] [AddCommGroup V] [Module K V] {m : ℕ} (e : Fin m → Fin m → Module.End K V) (he : ∀ i j k l, e i j * e k l - e k l * e i j = (if j = k then e i l else 0) - (if l = i then e k j else 0)) (v : V) (κ : Fin m → ℤ) (hv : ∀ i j, i < j → e i j v = 0) (hκ : ∀ i, e i i v = (κ i : K) • v) : ∑ i, ∑ j, e i j (e j i v) = ((∑ i, κ i * (κ i + m - 1 - 2 * (i : ℕ)) : ℤ) : K) • v

theorem **sl2_nonneg_weight_of_highest_weight_vector** {K V : Type*} [Field K] [CharZero K] [AddCommGroup V] [Module K V] (E F H : Module.End K V) (hEF : E * F - F * E = H) (hHF : H * F - F * H = -(F + F)) (W : Submodule K V) [FiniteDimensional K W] (hW : ∀ w ∈ W, F w ∈ W) (v : V) (hvW : v ∈ W) (hv0 : v ≠ 0) (l : ℤ) (hE : E v = 0) (hH : H v = (l : K) • v) : 0 ≤ l

theorem **highest_weight_antitone** {K V : Type*} [Field K] [CharZero K] [AddCommGroup V] [Module K V] {m : ℕ} (e : Fin m → Fin m → Module.End K V) (he : ∀ i j k l, e i j * e k l - e k l * e i j = (if j = k then e i l else 0) - (if l = i then e k j else 0)) (W : Submodule K V) [FiniteDimensional K W] (hW : ∀ i j, ∀ w ∈ W, e i j w ∈ W) (v : V) (hvW : v ∈ W) (hv0 : v ≠ 0) (κ : Fin m → ℤ) (hv : ∀ i j, i < j → e i j v = 0) (hκ : ∀ i, e i i v = (κ i : K) • v) : Antitone κ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SpanGridPerpVecEqOrbitSpanOfLe

@[simp] theorem **dotPairing_apply** (x y : Fin n → R) : dotPairing R n x y = x ⬝ᵥ y

theorem **map_perpVec_apply** (f : R →+* S) (h : (Fin n → R) →ₗ[R] ((Fin n → R) → R)) (h' : (Fin n → S) →ₗ[S] ((Fin n → S) → S)) (e e' : Fin p → Fin n → R) (g : Fin n → R) (ε ε' : Fin p → Fin n → S) (γ : Fin n → S) (he' : ∀ a k, f (e' a k) = ε' a k) (hg : ∀ k, f (g k) = γ k) (hM : ∀ a' a, f (h (e' a') (e a)) = h' (ε' a') (ε a)) (hge : ∀ a, f (h g (e a)) = h' γ (ε a)) (k : Fin n) : f (perpVec h e e' g k) = perpVec h' ε ε' γ k

theorem **map_perpVec_dotPairing** (f : R →+* S) (ε ε' : Fin p → Fin n → R) (γ : Fin n → R) : f ∘ perpVec (dotPairing R n) ε ε' γ = perpVec (dotPairing S n) (fun a => f ∘ ε a) (fun a => f ∘ ε' a) (f ∘ γ)

theorem **adjugate_isHomogeneous** {σ : Type*} (M : Matrix (Fin p) (Fin p) (MvPolynomial σ R)) (d : ℕ) (hM : ∀ a b, (M a b).IsHomogeneous d) (a b : Fin p) : (M.adjugate a b).IsHomogeneous ((p - 1) * d)

theorem **perpVec_dotPairing_isHomogeneous** {σ : Type} (ε ε' : Fin p → Fin n → MvPolynomial σ R) (γ : Fin n → MvPolynomial σ R) (hε : ∀ a k, (ε a k).IsHomogeneous 1) (hε' : ∀ a k, (ε' a k).IsHomogeneous 1) (hγ : ∀ k, (γ k).IsHomogeneous 1) (k : Fin n) : (perpVec (dotPairing _ n) ε ε' γ k).IsHomogeneous (2 * p + 1)

theorem **eval_swapVars** (x : σ → K) (F : MvPolynomial ω (MvPolynomial σ K)) : eval (fun i => C (x i)) (swapVars F) = MvPolynomial.map (eval x) F

theorem **swapVars_rowForm_isHomogeneous** (t : T) (r : Fin p ⊕ Fin q) (u : κ → MvPolynomial σ K) {d : ℕ} (hu : ∀ k, (u k).IsHomogeneous d) : (swapVars (rowForm t r u)).IsHomogeneous d

theorem **swapVars_minorProd_isHomogeneous** [Fintype T] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (ε : T → Fin p → κ → MvPolynomial σ K) (φ : T → Fin q → κ → MvPolynomial σ K) {d d' : ℕ} (hε : ∀ t a k, (ε t a k).IsHomogeneous d) (hφ : ∀ t b k, (φ t b k).IsHomogeneous d') : (swapVars (minorProd c c' ε φ)).IsHomogeneous (∑ t, (d ∗ partSize (c t) + d' ∗ partSize (c' t)))

theorem **map_coordProd** (f : R →+∗ S) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (w w' : T → ι → Fin n → R) (x : (Fin p ⊕ Fin p ⊕ Fin q) × ι → R) : MvPolynomial.map f (coordProd c c' w w' x) = coordProd c c' (fun t i k => f (w t i k)) (fun t i k => f (w' t i k)) fun j => f (x j)

theorem **coordProd_mem_orbitSpan** (hpq : p + q ≤ n) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (w w' : T → ι → Fin n → K) (x : (Fin p ⊕ Fin p ⊕ Fin q) × ι → K) : coordProd c c' w w' x ∈ orbitSpan K hpq c c'

theorem **span_range_coordProd** (hpq : p + q ≤ n) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (w w' : T → ι → Fin n → K) (hw : ∀ y y' : T → Fin n → K, ∃ x : ι → K, (∀ t k, ∑ i, x i ∗ w t i k = y t k) ∧ ∀ t k, ∑ i, x i ∗ w' t i k = y' t k) : Submodule.span K (Set.range (coordProd c c' w w')) = orbitSpan K hpq c c'

theorem **span_grid_coordProd** [CharZero K] (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (w w' : T → ι → Fin n → K) {dg : ℕ} (hdg : (2 ∗ p + 1) ∗ typeDegree c c' ≤ dg) : Submodule.span K ((fun x : (Fin p ⊕ Fin p ⊕ Fin q) × ι → ℕ => coordProd c c' w w' fun j => (x j : K)) '' {x | ∀ j, x j ≤ dg}) = Submodule.span K (Set.range (coordProd c c' w w'))

theorem **exists_sum_mul_embedding_eq** {E K J κ ι : Type∗} [Field E] [CharZero E] [Field K] [Finite J] [Finite κ] [Fintype ι] (σ : J → E →+∗ K) (hσ : Function.Injective σ) (v : ι → κ → E) (hv : Submodule.span ℚ (Set.range v) = ⊤) (y : J → κ → K) : ∃ x : ι → K, ∀ j k, ∑ i, x i ∗ σ j (v i k) = y j k

theorem **span_grid_perpVec_eq_orbitSpan_of_le** (τ : T → E →+∗ K) (hτ : Function.Injective fun s : T × Bool => fun x : E => τ s.1 (if s.2 then cc x else x)) (hpq : p + q ≤ n) (h : (Fin n → E) →ₗ[E] ((Fin n → E) → E)) (hh : ∀ x y, h x y = ∑ k, x k ∗ cc (y k)) (v : ι → Fin n → E) (hv : Submodule.span ℚ (Set.range v) = ⊤) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {dg : ℕ} (hdg : (2 ∗ p + 1) ∗ typeDegree c c' ≤ dg) : Submodule.span K {P | ∃ (A A' : Fin p → ι → ℕ) (B : Fin q → ι → ℕ), (∀ a i, A a i ≤ dg) ∧ (∀ a i, A' a i ≤ dg) ∧ (∀ b i, B b i ≤ dg) ∧ P = minorProd c c' (fun t a k => τ t (cc ((∑ i, (A a i : E) • v i) k))) fun t b k => τ t (perpVec h (fun a => ∑ i, (A a i : E) • v i) (fun a => ∑ i, (A' a i : E) • v i) (∑ i, (B b i : E) • v i) k)} = orbitSpan K hpq c c'

theorem **span_grid_perpVec_eq_orbitSpan** (τ : T → E →+∗ K) (hτ : Function.Injective fun s : T × Bool => fun x : E => τ s.1 (if s.2 then cc x else x)) (hpq : p + q ≤ n) (h : (Fin n → E) →ₗ[E] ((Fin n → E) → E)) (hh : ∀ x y, h x y = ∑ k, x k ∗ cc (y k)) (v : ι → Fin n → E) (hv : Submodule.span ℚ (Set.range v) = ⊤) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) {dg : ℕ} (hdg : (2 ∗ (p + q) + 1) ∗ typeDegree c c' ≤ dg) : Submodule.span K {P | ∃ (A A' : Fin p → ι → ℕ) (B : Fin q → ι → ℕ), (∀ a i, A a i ≤ dg) ∧ (∀ a i, A' a i ≤ dg) ∧ (∀ b i, B b i ≤ dg) ∧ P = minorProd c c' (fun t a k => τ t (cc ((∑ i, (A a i : E) • v i) k))) fun t b k => τ t (perpVec h (fun a => ∑ i, (A a i : E) • v i) (fun a => ∑ i, (A' a i : E) • v i) (∑ i, (B b i : E) • v i) k)} = orbitSpan K hpq c c'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SqDvdSumAlternatingFormsOfKilledLattice

theorem **sq_dvd_sum_EF_of_killed** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∀ a a' : Fin n → Dw.La, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a j)) → (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a' j)) → ((N : ℤ) ^ 2) ∣ ∑ b, Dw.EF (a b) (a' b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SqDvdSumAlternatingFormsOfKilledLatticeStatement

theorem **_root_.GlobalTheory.T.sq_dvd_sum_alternating_forms_of_killed_lattice** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → ∀ a a' : Fin n → Dw.La, (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a j)) → (∀ (mu : Matrix (Fin n) (Fin n) E), (∀ v ∈ Λ.La, mu.mulVec v ∈ stdLat E n) → ∀ cm : Fin n → Fin n → 𝒪 E, (∀ i j, (cm i j : E) = (N : E) ∗ mu i j) → ∀ i, Dw.ModN (N ^ 2) (∑ j, cm i j • a' j)) → ((N : ℤ) ^ 2) ∣ ∑ b, Dw.EF (a b) (a' b)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SquareIntegrableSingularMetricOrderOne

theorem **Wfree_of_mem** (n : ℕ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : Wfree Xa Fl G n x = ENNReal.ofReal (Real.exp (-phla X G x)) ∗ ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) ∗ ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2))

theorem **Wfree_eq_pnorm** (n : ℕ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : Wfree Xa Fl G n x = ENNReal.ofReal (Real.exp (-phla X (G + n • Xa.L) x)) ∗ ENNReal.ofReal (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (-(1 / 2 : ℝ)) ∗ ENNReal.ofReal (∑ k, X.pnorm Xa.L (Fl.s k) x ^ 2) ^ (-((n : ℝ) - 1 / 2))

theorem **phiW_of_mem** (n : ℕ) (h : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : phiW Xa Fl G n h x = ENNReal.ofReal (X.pnorm (G + n • Xa.L) h x ^ 2 ∗ (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (-(1 / 2 : ℝ)) ∗ (∑ k, X.pnorm Xa.L (Fl.s k) x ^ 2) ^ (-((n : ℝ) - 1 / 2)))

theorem **measurable_phiW** (n : ℕ) {h : Frame X.Ψ r → ℂ} (hh : X.IsSec (G + n • Xa.L) h) : Measurable (phiW Xa Fl G n h)

theorem **phiW_inv** (hN : NfacRmulStatement (E

theorem **phiW_eq** (n : ℕ) (h : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) (hs : ∃ k, Fl.s k x ≠ 0) : phiW Xa Fl G n h x = ENNReal.ofReal (‖h x‖ ^ 2) ∗ Wfree Xa Fl G n x

theorem **phiW_le_local** {n : ℕ} (hn : 1 ≤ n) (h : Frame X.Ψ r → ℂ) {x : Frame X.Ψ r} (hx : x ∈ X.dom) : phiW Xa Fl G n h x ≤ ENNReal.ofReal (X.nfac G x ^ 2 ∗ (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ))) ∗ ENNReal.ofReal (‖h x‖ ^ 2 ∗ (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-((n : ℝ) - 1 / 2)))

theorem **localW_int** {n : ℕ} (hn : 1 ≤ n) (h : Frame X.Ψ r → ℂ) {y : Frame X.Ψ r} (hy : y ∈ X.dom) {ρ : ℝ} (hρ : 0 < ρ) (hint : ∫⁻ z in Metric.ball (0 : V (nd X)) ρ, ENNReal.ofReal (‖h (cpt X y z)‖ ^ 2 ∗ (∑ k, ‖Fl.s k (cpt X y z)‖ ^ 2) ^ (-((n : ℝ) - 1 / 2))) < ⊤) : ∃ ρ' : ℝ, 0 < ρ' ∧ ∫⁻ z in Metric.ball (0 : V (nd X)) ρ', phiW Xa Fl G n h (cpt X y z) < ⊤

theorem **inFF_sub** {T : ℕ} (hT1 : 1 ≤ T) (hnT : nd X ≤ T) {x0 : Frame X.Ψ r} (ho : IsOrth X x0) {FF : Set (V (nd X))} (hF : IsFundSet X x0 FF) {f g : Frame X.Ψ r → ℂ} (hf : X.IsSec (G + T • Xa.L) f) (hg : X.IsSec (G + T • Xa.L) g) (hfi : ∫⁻ z in FF, ENNReal.ofReal (X.pnorm (G + T • Xa.L) f (cpt X x0 z) ^ 2 ∗ (∑ k, X.pnorm Xa.L (Xa.w k) (cpt X x0 z) ^ 2) ^ (-(T : ℝ))) ∂muF X x0 < ⊤) {C : ℝ} (hC : ∀ x ∈ X.dom, X.pnorm (G + T • Xa.L) g x ^ 2 ≤ C ∗ (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ T) (hvan : VanS X Fl.xs Fl.s (T - nd X) (fun x => f x - g x)) : InFF X x0 (Wfree Xa Fl G T) FF (fun x => f x - g x)

theorem **inFF_order_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r), IsOrth X x0 → ∀ FF, IsFundSet X x0 FF → ∀ f : Frame X.Ψ r → ℂ, X.IsSec (G + (nd X + 1) • Xa.L) f → (∃ C : ℝ, ∀ x ∈ X.dom, X.pnorm (G + (nd X + 1) • Xa.L) f x ^ 2 ≤ C ∗ (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (nd X + 1)) → (∀ j, f (Fl.xs

j) = 0) → InFF X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-(((nd X + 1 : ℕ) : ℝ) - 1 / 2)) else 0) FF f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SquareIntegrableSingularMetricOrderOneStatement

theorem **_root_.GlobalTheory.T.square_integrable_singular_metric_order_one** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (Fl : FlatM X Xa) (G : Wt X.Ψ r) (x0 : Frame X.Ψ r), IsOrth X x0 → ∀ FF, IsFundSet X x0 FF → ∀ f : Frame X.Ψ r → ℂ, X.IsSec (G + (nd X + 1) • Xa.L) f → (∃ C : ℝ, ∀ x ∈ X.dom, X.pnorm (G + (nd X + 1) • Xa.L) f x ^ 2 ≤ C * (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) ^ (nd X + 1)) → (∀ j, f (Fl.xs j) = 0) → InFF X x0 (fun x => open Classical in if x ∈ X.dom then ENNReal.ofReal (Real.exp (-phla X G x)) * ENNReal.ofReal (∑ k, ‖Xa.w k x‖ ^ 2) ^ (-(1 / 2 : ℝ)) * ENNReal.ofReal (∑ k, ‖Fl.s k x‖ ^ 2) ^ (-(((nd X + 1 : ℕ) : ℝ) - 1 / 2)) else 0) FF f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SumLeFinsumOfSurjectsOntoFinset

theorem **norm_dmin** {r : ℕ} (X : FD.Data E r) {x : FD.Frame X.Ψ r} (hx : X.IsOrtho x) (τ : X.Ψ) (i : Fin (r + 1)) : ‖FD.dmin X.Gm x τ i‖ = 1

theorem **finite_levQ** {H : Host E} (Dw : Datum H) (l : Geo.Tor.Level H) : Finite (LevQ Dw l)

theorem **exists_mult_le** {H : Host E} (Dw : Datum H) (m : ℕ) (h : Dw.p ≤ m ∨ Dw.q ≤ m) : ∃ τ, Dw.mult τ ≤ m

theorem **normSq_emb_bar** {H : Host E} (α : 𝒪 E) (M : ℕ) (hα : α * H.bar α = (M : 𝒪 E)) (τ : E →+* ℂ) : ‖τ ((H.bar α : 𝒪 E) : E)‖ ^ 2 = M

theorem **exists_kappa** {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} (c : Coeff l) (σ : c.F →+* ℂ) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ harmA c a n) (h0 : P ≠ 0) (hh : P.IsHomogeneous P.totalDegree) (s : (E →+* ℂ) → ℂ) (M : ℝ) (hs : ∀ τ, ‖s τ‖ ^ 2 = M) : ∃ κ : ℂ, ‖κ‖ ^ 2 = M ^ P.totalDegree ∧ ∀ m ∈ (MvPolynomial.map σ P).support, (m.prod fun ℓ e => s ℓ.1 ^ e) = κ

theorem **sum_le_finsum** {A B : Type} [Finite B] (K : Finset A) (φ : B → A) (u : A → ℝ) (v : B → ℝ) (hK : ∀ a ∈ K, ∃ b, a = φ b) (huv : ∀ b, v b = u (φ b)) (hv : ∀ b, 0 ≤ v b) : ∑ a ∈ K, u a ≤ ∑ᶠ b, v b

theorem **differentiable_translate** (X : I2.PT E) (q n : ℕ) (π : I2.Vsp X.dim →ₗ[ℂ] I2.Vsp X.dim) (s : (Fin 1 → I2.Vsp X.dim) → ℂ) (hs : Differentiable ℂ s) (b : Fin n → I2.Vsp X.dim) : Differentiable ℂ (X.U q b (I2.s0 X π s))

theorem **theta_law** {H : Host E} {Dw : Datum H} {l : Geo.Tor.Level H} {Xo : Moduli Dw l} {c : Coeff l} {σ : c.F →+* ℂ} {r : ℕ} {U : Unf Xo c σ r} (hU : IsUnf U) {k : U.Comp} {x : (U.X k).dom} (W : UnfAt U k x) (Wu : UnitAt W) (hu : ∀ τ' i', ‖FD.dmin (U.X k).Gm x.1 τ' i'‖ = 1) : Differentiable ℂ Wu.θ ∧ ∀ a z, Wu.θ (z + W.Pm a) = (((Int.negOnePow ((W.F.Di (W.ik k)).EF a (H.w0 • a)) : ℤˣ) : ℤ) : ℂ) * Complex.exp (Real.pi * hermForm Wu.Hm z (W.Pm a) + Real.pi / 2 * hermForm Wu.Hm (W.Pm a) (W.Pm a)) * Wu.θ z

theorem **value_torus** {H : Host E} {Dw : Datum H} (T : I2.PT E) {g n : ℕ} (e : Fin g ≃ Lt1 Dw) (ψ : (Fin g → ℂ) ≃ₗ[ℂ] I2.Vsp T.dim) (Hm : Matrix (Fin g) (Fin g) ℂ) (θ : (Fin g → ℂ) → ℂ) (hH : ∀ z w, I2.Hf (ψ z) (ψ w) = hermForm Hm z w) (hθ : Differentiable ℂ θ) (rl : (E →+* ℂ) → (E →+* ℂ)) (hd : ∀ τ, Dw.mult τ ≤ T.dim (rl τ)) (t : ℂ) (hrow : ∀ j, ∃ h : ((e j).2 : ℕ) < T.dim (rl (e j).1), ψ (Pi.single j 1) = t • EuclideanSpace.single ((rl (e j).1, ((e j).2, h)) : I2.Idx T.dim) (1 : ℂ)) (α : 𝒪 E) (M : ℕ) (hα : α * H.bar α = (M : 𝒪 E)) (Q : MvPolynomial (LtN Dw n) ℂ) (κ : ℂ) (hκ : ∀ m ∈ Q.support, (m.prod fun ℓ k => rl ℓ.1 ((H.bar α : 𝒪 E) : E) ^ k) = κ) (a : Fin n → Fin g → ℂ) : I2.polyDeriv Q (letDir e) (fun z => ∏ b, fock Hm ((M : ℤ) ^ 2) (a b) (fun w => θ ((M : ℂ) • w)) (z b)) 0 = κ * I2.polyDeriv Q (fun ℓ => t • I2.dir ((ℓ.2.2, (rl ℓ.1, Fin.castLE (hd ℓ.1) ℓ.2.1)) : I2.Letter n T.dim)) (

theorem **norm_sq_value** (v κ D : ℂ) (M d : ℕ) (t : ℝ) (hM : (0 : ℝ) < M) (ht : 0 ≤ t) (hκ : ‖κ‖ ^ 2 = (M : ℝ) ^ d) (hv : v = ((M : ℂ)⁻¹) ^ d * (κ * (((Real.sqrt t : ℝ) : ℂ) ^ d * D))) : ‖v‖ ^ 2 = (t / M) ^ d * ‖D‖ ^ 2

theorem **thetaRule_sum_lower** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (xR : ℝ) (nab : NabV Xo c), IsNab Xo c nab → GoodEmb c σ → IsUnf U → FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SumLeFinsumOfSurjectsOntoFinsetStatement

theorem **_root_.GlobalTheory.T.unfoldedThetaRule_archimedeanValue_bound** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (x : (U.X k).dom) (xR : ℝ) (nab : NabV Xo c), IsNab Xo c nab → GoodEmb c σ → IsUnf U → FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SumSmulMemOfSumZpowSmulMem

theorem **sum_smul_mem_of_sum_pow_smul_mem** (W : Submodule K V) (S : Finset ι) (e : ι → ℕ) (v : ι → V) (h : ∀ u : K, u ≠ 0 → ∑ i ∈ S, u ^ e i • v i ∈ W) (f : ℕ → K) : ∑ i ∈ S, f (e i) • v i ∈ W

theorem **sum_smul_mem_of_sum_zpow_smul_mem** (W : Submodule K V) (S : Finset ι) (e : ι → ℤ) (v : ι → V) (h : ∀ u : K, u ≠ 0 → ∑ i ∈ S, u ^ e i • v i ∈ W) (f : ℤ → K) : ∑ i ∈ S, f (e i) • v i ∈ W

theorem **derivation_mem_of_aeval_mem** {σ : Type*} (W : Submodule K (MvPolynomial σ K)) (D : Derivation K (MvPolynomial σ K) (MvPolynomial σ K)) (w : MvPolynomial σ K) (h : ∀ s : K, aeval (fun v => X v + s • D (X v)) w ∈ W) : D w ∈ W

theorem **colDer_apply** (t : T) (k l : κ) (P : Poly R T p q κ) : colDer R p q t k l P = ∑ a, X (xVar t a k) * pderiv (xVar t a l) P - ∑ b, X (yVar t b l) * pderiv (yVar t b k) P

theorem **colLie_apply** (t : T) (k l : κ) (P : Poly R T p q κ) : colLie R p q t k l P = ∑ a, X (xVar t a k) * pderiv (xVar t a l) P - ∑ b, X (yVar t b l) * pderiv (yVar t b k) P

theorem **colDer_X_xVar** [DecidableEq T] [DecidableEq κ] (t t' : T) (k l j : κ) (a : Fin p) : colDer R p q t k l (X (xVar t' a j)) = if t' = t ∧ j = l then X (xVar t a k) else 0

theorem **colDer_X_yVar** [DecidableEq T] [DecidableEq κ] (t t' : T) (k l j : κ) (b : Fin q) : colDer R p q t k l (X (yVar t' b j)) = if t' = t ∧ j = k then -X (yVar t b l) else 0

theorem **colWeight_add** (m m' : Var T p q κ →₀ ℕ) (t : T) (k : κ) : colWeight (m + m') t k = colWeight m t k + colWeight m' t k

theorem **colWeight_single_xVar** [DecidableEq T] [DecidableEq κ] (t t' : T) (a : Fin p) (k j : κ) : colWeight (Finsupp.single (xVar t' a j : Var T p q κ) 1) t k = if t' = t ∧ j = k then 1 else 0

theorem **colWeight_single_yVar** [DecidableEq T] [DecidableEq κ] (t t' : T) (b : Fin q) (k j : κ) : colWeight (Finsupp.single (yVar t' b j : Var T p q κ) 1) t k = if t' = t ∧ j = k then -1 else 0

theorem **colLie_monomial** (t : T) (k : κ) (m : Var T p q κ →₀ ℕ) (c : R) : colLie R p q t k k (monomial m c) = (colWeight m t k : R) • monomial m c

theorem **iSup_iInf_eigenspace_colLie** : ⨆ μ : T → κ → ℤ, (⨅ t, ⨅ k, Module.End.eigenspace (colLie R p q t k k) (μ t k : R)) = ⊤

theorem **colLie_diag_comm** [DecidableEq T] [DecidableEq κ] (t t' : T) (i k l : κ) : colLie R p q t i i * colLie R p q t' k l - colLie R p q t' k l * colLie R p q t i i = (if t = t' then ((if i = k then 1 else 0) - (if i = l then 1 else 0) : R) else 0) • colLie R p q t' k l

theorem **exists_colAct_eq_aeval** (t : T) {k l : κ} (hkl : k ≠ l) (s : R) : ∃ g : T → GL κ R, ∀ P : Poly R T p q κ, colAct R T p q κ g P = aeval (fun v => X v + s • colDer R p q t k l (X v)) P

theorem **exists_colAct_monomial** (t : T) (k : κ) (u : Rˣ) : ∃ g : T → GL κ R, ∀ (m : Var T p q κ →₀ ℕ) (c : R), colAct R T p q κ g (monomial m c) = ((u ^ colWeight m t k : Rˣ) : R) • monomial m c

theorem **colLie_mem_of_colAct_mem** {K : Type*} [Field K] [Infinite K] [Fintype κ] [DecidableEq κ] (W : Submodule K (Poly K T p q κ)) (hW : ∀ g : T → GL κ K, ∀ w ∈ W, colAct K T p q κ g w ∈ W) (t : T) (k l : κ) {w : Poly K T p q κ} (hw : w ∈ W) : colLie K p q t k l w ∈ W

theorem **colLie_mem_orbitSpan** {K : Type*} [Field K] [Infinite K] [Fintype T] {n : ℕ} (h : p + q ≤ n) (c : T → Fin p → ℕ) (c' : T → Fin q → ℕ) (t : T) (k l : Fin n) {w : Poly K T p q (Fin n)} (hw : w ∈ orbitSpan K h c c') : colLie K p q t k l w ∈ orbitSpan K h c c'

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SupBoundsLeGeneratorConstantMulSize

theorem **gc_arith** (f r L A n s lam lp a b M : ℝ) (hf : 2 ≤ f) (hr : 6 ≤ r) (hL : 2 ≤ L) (hA : 0 ≤ A) (hn : 1 ≤ n) (hs : A + f * r * n + 1 ≤ s) (hlam : 0 ≤ lam) (hlp : 0 ≤ lp) (ha : 0 ≤ a) (hb : 0 ≤ b) (hM : a + b ≤ M) : (140 * r * L * A + 10 * L * A + 200 * f * r * n * L ≤ 400 * (f * r * L) * s) ∧ (2 * (lam + 2 * lp) * s + 400 * (f * r * L) * s + lam * s ≤ (1009 * (f * r * L) + (3 * L + 2 * M) + 2 * f * r * (lam + 2 * lp) + 31 * (f * r * L)) * s) ∧ ((lam + 2 * lp) * s + (400 * (f * r * L) * s + 3 * f * r * n * L + 5 * a + b + 2) ≤ (1009 * (f * r * L) + (3 * L + 2 * M) + 2 * f * r * (lam + 2 * lp) + 31 * (f * r * L)) * s)

theorem **lamStar_nonneg_nat** (D : D7) : 0 ≤ lamStar D

theorem **sup_bounds_le_Gc** : ∀ (D : D7) (B : WeightFamilies.Prov D) (large₂ : SizeData₂ D) (j : ℕ) (w : WeightFamilies.Wt D j) (p : ℕ), 2 ≤ large₂.sc.L → 0 < w.n → (140 * D.R.r * large₂.sc.L * absA w + 10 * large₂.sc.L * absA w + 200 * D.R.f * D.R.r * w.n * large₂.sc.L ≤ 400 * Zb D large₂ * w.size) ∧ (2 * (lamStar D + 2 * Real.log p) * w.size + 400 * Zb D large₂ * w.size + lamStar D * w.size ≤ WeightAlgebra.Gc large₂ B (1009 * Zb D large₂) p * w.size) ∧ ((lamStar D + 2 * Real.log p) * w.size + (400 * Zb D large₂ * w.size + 3 * D.R.f * D.R.r * w.n * large₂.sc.L + 5 * Real.log (Module.finrank ℚ B.KO) + Real.log |((NumberField.discr B.KO : ℤ) : ℝ)| / Module.finrank ℚ B.KO + 2) ≤ WeightAlgebra.Gc large₂ B (1009 * Zb D large₂) p * w.size)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SupBoundsLeGeneratorConstantMulSizeStatement

theorem **_root_.GlobalTheory.T.supBounds_le_generatorConstant_mul_size** : ∀ (D : D7) (B : WeightFamilies.Prov D) (large₂ : SizeData₂ D) (j : ℕ) (w : WeightFamilies.Wt D j) (p : ℕ), 2 ≤ large₂.sc.L → 0 < w.n → (140 * D.R.r * large₂.sc.L * absA w + 10 * large₂.sc.L * absA w + 200 * D.R.f * D.R.r * w.n * large₂.sc.L ≤ 400 * Zb D large₂ * w.size) ∧ (2 * (lamStar D + 2 * Real.log p) * w.size + 400 * Zb D large₂ * w.size + lamStar D * w.size ≤ WeightAlgebra.Gc large₂ B (1009 * Zb D large₂) p * w.size) ∧ ((lamStar D + 2 * Real.log p) * w.size + (400 * Zb D large₂ * w.size + 3 * D.R.f * D.R.r * w.n * large₂.sc.L + 5 * Real.log (Module.finrank ℚ B.KO) + Real.log |((NumberField.discr B.KO : ℤ) : ℝ)| / Module.finrank ℚ B.KO + 2) ≤ WeightAlgebra.Gc large₂ B (1009 * Zb D large₂) p * w.size)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SupLeL2PrincipalTorus

theorem **_root_.GlobalTheory.packingCount** : _root_.ReciprocityTheory.ChartRestriction.G.I2.Ext.PackingCount

theorem **_root_.GlobalTheory.sup_le_l2_principal_torus** : _root_.ReciprocityTheory.ChartRestriction.G.I2.GSupStatement

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SupLeOfBoundAtThickPoints

theorem **maj_complex_smul_normSq** {r : ℕ} (X : Data E r) (x : Frame X.Ψ r) (τ : X.Ψ) (c : ℂ) (v : Fin r → ℂ) : X.maj x τ (c • v) = ‖c‖ ^ 2 * X.maj x τ v

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SymmetricLiftOddOrderActsAsTranslation

theorem **rd_fock_of_symLift** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), (∀ a ∈ Φ.lat, Φ.χ a ^ 2 = 1) → ∀ (k : ℤ) (n : ℕ) (U : ThetaGrp A L.L k (Over.mk x0)) (x δ : Fin g → ℂ) (s s' : LB.Sec (LPow A L.L k (Over.mk x0))) (θ : (Fin g → ℂ) → ℂ), Odd n → δ ∈ Φ.lat → (n : ℂ) • x = (2 : ℂ) • δ → SymLift A L.L k n (Φ.e x) U → IsActOf A L.L k U s s' → Φ.Rd k s θ → Φ.Rd k s' (fock Φ.Hm k x θ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.SymmetricLiftOddOrderActsAsTranslationStatement

theorem **_root_.GlobalTheory.T.symmetric_lift_odd_order_acts_as_translation** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), (∀ a ∈ Φ.lat, Φ.χ a ^ 2 = 1) → ∀ (k : ℤ) (n : ℕ) (U : ThetaGrp A L.L k (Over.mk x0)) (x δ : Fin g → ℂ) (s s' : LB.Sec (LPow A L.L k (Over.mk x0))) (θ : (Fin g → ℂ) → ℂ), Odd n → δ ∈ Φ.lat → (n : ℂ) • x = (2 : ℂ) • δ → SymLift A L.L k n (Φ.e x) U → IsActOf A L.L k U s s' → Φ.Rd k s θ → Φ.Rd k s' (fock Φ.Hm k x θ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaConstSymmetricSquareEqOne

theorem **translPt_eq** (x : T ⟶ A) : translPt A x = (fst T A ≫ x) * snd T A

theorem **translT_snd** (x : T ⟶ A) : translT A x ≫ snd T A = (fst T A ≫ x) * snd T A

theorem **translT_fst** (x : T ⟶ A) : translT A x ≫ fst T A = fst T A

theorem **negT_snd** : negT A T ≫ snd T A = (snd T A)⁻¹

theorem **negT_fst** : negT A T ≫ fst T A = fst T A

theorem **snd_inv** : snd T A ≫ (GrpObj.inv : A ⟶ A) = (snd T A)⁻¹

theorem **translT_inv_translPt** (x : T ⟶ A) : translT A x⁻¹ ≫ translPt A x = snd T A

theorem **negT_translPt** [IsCommMonObj A] (x : T ⟶ A) : negT A T ≫ translPt A x = translT A x⁻¹ ≫ (snd T A ≫ (GrpObj.inv : A ⟶ A))

theorem **ptSecT_translT** (a x : T ⟶ A) : ptSecT A a ≫ translT A x = ptSecT A (x * a)

theorem **ptSecT_negT** (a : T ⟶ A) : ptSecT A a ≫ negT A T = ptSecT A a⁻¹

theorem **e_zero** : Φ.e 0 = 1

theorem **e_neg** (z : Fin g → ℂ) : Φ.e (-z) = (Φ.e z)⁻¹

private theorem **isHolFam_comp'** {Y Y' : Scheme.{0}} {n : ℕ} {e : (Fin n → ℂ) → (Sp ℂ ⟶ Y)} (he : IsHolFam e) (h : Y ⟶ Y') : IsHolFam (fun z => e z ≫ h)

theorem **hol_ptSecT** : IsHolFam (fun z => ((ptSecT A (Φ.e z)).left : Sp ℂ ⟶ (Over.mk x0 ⊗ A).left))

theorem **genRd_one** {Z : Scheme.{0}} (q : Sp ℂ ⟶ Z) : Φ.GenRd (K

theorem **neg_const** [IsCommMonObj A] (k : ℤ) (nu : LB.Gen (negLB A L.L k (Over.mk x0))) (hnu : nu.IsGen) : ∃ v : ℂ, v ≠ 0 ∧ ∀ z, Φ.GenRd (K

theorem **sq_const_eq_one** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ) (c : ℂ), Φ.ThConst k U x c → (∃ nu : LB.Gen (negLB A L.L k (Over.mk x0)), nu.IsGen ∧ LB.Gen.Corr L.L _ _ _ _ (symGen A L.L k U nu) (LB.oneGen L.L)) → c ^ 2 = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaConstSymmetricSquareEqOneStatement

theorem **_root_.GlobalTheory.T.thetaConst_symmetric_square_eq_one** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ) (c : ℂ), Φ.ThConst k U x c → (∃ nu : LB.Gen (negLB A L.L k (Over.mk x0)), nu.IsGen ∧ LB.Gen.Corr L.L _ _ _ _ (symGen A L.L k U nu) (LB.oneGen L.L)) → c ^ 2 = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaConstantPowEqOneOfPowerTrivial

theorem **telescope** (c h : ℂ) (k : ℤ) (f : ℕ → ℂ) (hf : ∀ i, f i ≠ 0) (n : ℕ) : ∏ i ∈ Finset.range n, (c * Complex.exp (-((k : ℂ) * Real.pi * (i * h)) - (k : ℂ) * Real.pi * h / 2) * (f i ^ k * f (i + 1) ^ (-k))) = c ^ n * Complex.exp (-((k : ℂ) * (Real.pi / 2 * (n * n * h)))) * (f 0 ^ k * f n ^ (-k))

theorem **herm_smul_left** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (a : ℂ) (z w : Fin g → ℂ) : hermForm Hm (a • z) w = a * hermForm Hm z w

theorem **herm_nat_right** {g : ℕ} (Hm : Matrix (Fin g) (Fin g) ℂ) (a : ℕ) (z w : Fin g → ℂ) : hermForm Hm z ((a : ℂ) • w) = a * hermForm Hm z w

theorem **translPt_mul** (x : T ⟶ A) : translPt A x = (fst T A ≫ x) * snd T A

theorem **translT_snd** (y : T ⟶ A) : translT A y ≫ snd T A = translPt A y

theorem **translT_translPt** [IsCommMonObj A] (x y : T ⟶ A) : translT A y ≫ translPt A x = translPt A (x * y)

theorem **pt_translPt** (x z : T ⟶ A) : ptSecT A z ≫ translPt A x = x * z

theorem **pt_translT** (x z : T ⟶ A) : ptSecT A z ≫ translT A x = ptSecT A (x * z)

theorem **powφ_zero** (x : T ⟶ A) (n : ℕ) (i : Fin n) : powφ A x n (i, 0) = (translPt A (x ^ (i : ℕ))).left

theorem **powφ_one** [IsCommMonObj A] (x : T ⟶ A) (n : ℕ) (i : Fin n) : powφ A x n (i, 1) = (translPt A (x ^ ((i : ℕ) + 1))).left

private theorem **isChart_choose_and_symm_apply** (L : LB A.left) (k : ℤ) (θ : ThetaGrp A L k T) (n : ℕ) (q : Sp ℂ ⟶ (T ⊗ A).left) (j : Fin n × Fin 2 → L.ι) (hj : ⊤ ≤ q ⁻¹ᵁ LB.wchart L (powφ A θ.pt n) j) (i : Fin n) : ⊤ ≤ (q ≫ (translT A (θ.pt ^ (i : ℕ))).left) ⁻¹ᵁ LB.wchart L ![(snd T A).left, (translPt A θ.pt).left] (fun b => j (i, b))

private theorem **exists_unit_mul_eq_mul_uniformizer_pow** {Z Y : Scheme.{0}} (q : Sp ℂ ⟶ Z) (ψ : Z ⟶ Y) (V : Y.Opens) {W W' : Z.Opens} (e1 : W ≤ W') (e2 : W' ≤ ψ ⁻¹ᵁ V) (h : ⊤ ≤ q ⁻¹ᵁ W) (h' : ⊤ ≤ (q ≫ ψ) ⁻¹ᵁ V) (u : Γ(Y, V)ˣ) : (q.appLE W ⊤ h).hom ((Z.presheaf.map (homOfLE e1).op).hom ((Z.presheaf.map (homOfLE e2).op).hom ((ψ.app V).hom u))) = ((q ≫ ψ).appLE V ⊤ h').hom u

private theorem **valAt'_prod_res_pullU_eq_prod** {Z Y : Scheme.{0}} {n : ℕ} (q : Sp ℂ ⟶ Z) (W : Z.Opens) (h : ⊤ ≤ q ⁻¹ᵁ W) (ψ : Fin n → (Z ⟶ Y)) (V : Fin n → Y.Opens) (W' : Fin n → Z.Opens) (e1 : ∀ i, W ≤ W' i) (e2 : ∀ i, W' i ≤ ψ i ⁻¹ᵁ V i) (u : ∀ i, Γ(Y, V i)ˣ) (h' : ∀ i, ⊤ ≤ (q ≫ ψ i) ⁻¹ᵁ V i) : valAt' q W (∏ i, res (e1 i) (res (e2 i) (pullU (ψ i) (V i) (u i)))).val = ∏ i, valAt' (q ≫ ψ i) (V i) (u i).val

theorem **val_powGen** (L : LB A.left) (k : ℤ) (θ : ThetaGrp A L k T) (n : ℕ) (q : Sp ℂ ⟶ (T ⊗ A).left) (j : Fin n × Fin 2 → L.ι) (hj : ⊤ ≤ q ⁻¹ᵁ LB.wchart L (powφ A θ.pt n) j) : valAt' q (LB.wchart L (powφ A θ.pt n) j) ((powGen A L k θ n).s j).val = ∏ i : Fin n, valAt' (q ≫ (translT A (θ.pt ^ (i : ℕ))).left) (LB.wchart L ![(snd T A).left, (translPt A θ.pt).left] (fun b => j (i, b))) (θ.φ.s (fun b => j (i, b))).val

theorem **e_nat** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) (x : Fin g → ℂ) (i : ℕ) : Φ.e ((i : ℂ) • x) = Φ.e x ^ i

theorem **thConst_pow** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (n : ℕ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ) (c : ℂ), Φ.ThConst k U x c → (n : ℂ) • x ∈ Φ.lat → LB.Gen.Corr L.L _ _ _ _ (powGen A L.L k U n) (LB.oneGen L.L) → c ^ n * Φ.χ ((n : ℂ) • x) ^ k = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaConstantPowEqOneOfPowerTrivialStatement

theorem **_root_.GlobalTheory.T.thetaConstant_pow_eq_one_of_power_trivial** : ∀ (S : Scheme.{0}) (A : Over S) [GrpObj A] [IsCommMonObj A] (L : RigLB A) (x0 : Sp ℂ ⟶ S) (g : ℕ) (Φ : AnFr A L x0 g), ∀ (k : ℤ) (n : ℕ) (U : ThetaGrp A L.L k (Over.mk x0)) (x : Fin g → ℂ) (c : ℂ), Φ.ThConst k U x c → (n : ℂ) • x ∈ Φ.lat → LB.Gen.Corr L.L _ _ _ _ (powGen A L.L k U n) (LB.oneGen L.L) → c ^ n * Φ.χ ((n : ℂ) • x) ^ k = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaExpressionEqOfWitnesses

theorem **exists_pull** (hC : ChartIndepSt2) [StarRing (𝒪 E)] (nab : NabV Xo c) (hnab : IsNab Xo c nab) {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S) (z : FrPt Xo c T) (hm : FrMap ψ y z) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F) (hP : P ∈ harmA c a n) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y.x))) (hF : ∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) y.s (levPt Xo y.x (x b)) (F b)) : ∃ F' : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo z.x)), (∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) z.s (levPt Xo z.x (x b)) (F' b)) ∧ ψ (thExpr Xo c nab y P F) = thExpr Xo c nab z P F'

theorem **thExpr_eq_of_witnesses** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ (S1 : Type) [CommRing S1] [Algebra c.F S1] (φ1 : S →ₐ[c.F] S1) (y1 : FrPt Xo c S1) (F1 : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y1.x))) (S2 : Type) [CommRing S2] [Algebra c.F S2] (φ2 : S →ₐ[c.F] S2) (y2 : FrPt Xo c S2) (F2 : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y2.x))), FrMap φ1 y y1 → (∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) y1.s (levPt Xo y1.x (x b)) (F1 b)) → FrMap φ2 y y2 → (∀ b, IsVtOver Xo.u

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaExpressionEqOfWitnessesStatement

theorem **_root_.GlobalTheory.T.thetaExpression_eq_of_witnesses** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ (S1 : Type) [CommRing S1] [Algebra c.F S1] (φ1 : S →ₐ[c.F] S1) (y1 : FrPt Xo c S1) (F1 : Fin n →

LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y1.x))) (S2 : Type) [CommRing S2] [Algebra c.F S2] (φ2 : S →ₐ[c.F] S2) (y2 : FrPt Xo c S2) (F2 : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y2.x))), FrMap φ1 y y1 → (∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) y1.s (levPt Xo y1.x (x b)) (F1 b)) → FrMa

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaFormsBoundedByGenericCombinations

theorem **cone_sum_sq_le** {m N : ℕ} (t : Fin m × Fin N → ℂ) (Z : Set (Fin N → ℂ)) (hZ : IsClosed Z) (hc : ∀ (c : ℂ), ∀ ζ ∈ Z, c • ζ ∈ Z) (h0 : ∀ ζ ∈ Z, (∀ i, ∑ j, t (i, j) * ζ j = 0) → ζ = 0) : ∃ C : ℝ, ∀ ζ ∈ Z, ∑ k, ‖ζ k‖ ^ 2 ≤ C * ∑ i, ‖∑ j, t (i, j) * ζ j‖ ^ 2

theorem **eval₂_smul_of_isHomogeneous** {K : Type} [Field K] (σ : K →+* ℂ) {n d : ℕ} {Q : MvPolynomial (Fin n) K} (hQ : Q.IsHomogeneous d) (c : ℂ) (z : Fin n → ℂ) : MvPolynomial.eval₂ σ (fun a => c * z a) Q = c ^ d * MvPolynomial.eval₂ σ z Q

theorem **forms_le_generic_comb** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (K : Type) [Field K] (σ : K →+* ℂ) (t : Fin (nd X + 1) × Fin Xa.N → ℂ), (∀ Q : MvPolynomial (Fin (nd X + 1) × Fin Xa.N) K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0) → RelIn σ X Xa → let sf : Fin (nd X + 1) → Frame X.Ψ r → ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaFormsBoundedByGenericCombinationsStatement

theorem **_root_.GlobalTheory.T.theta_forms_bounded_by_generic_combinations** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (K : Type) [Field K] (σ : K →+* ℂ) (t : Fin (nd X + 1) × Fin Xa.N → ℂ), (∀ Q : MvPolynomial (Fin (nd X + 1) × Fin Xa.N) K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0) → RelIn σ X Xa → let sf : Fin (nd X + 1) → Frame X.Ψ r → ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaFormsNotSmallWhereCombinationsVanish

theorem **eval₂_smul_of_isHomogeneous** {ι : Type} {F : MvPolynomial ι K} {e : ℕ} (hF : F.IsHomogeneous e) (x : ι → ℂ) (c : ℂ) : eval₂ σ (fun j => c * x j) F = c ^ e * eval₂ σ x F

theorem **eval₂_cons_zero** {n : ℕ} (y : Fin n → ℂ) (Q : MvPolynomial (Fin (n + 1)) K) : eval₂ σ (Fin.cons 0 y : Fin (n + 1) → ℂ) Q = eval₂ σ y ((finSuccEquiv K n Q).coeff 0)

theorem **exists_rel_coeff_zero_ne** {Y : Type} {n : ℕ} (S : Set Y) (u : Fin (n + 1) → Y → ℂ) (hu : ∀ x ∈ S, u 0 x ≠ 0) (d : ℕ) : ∀ Q : MvPolynomial (Fin (n + 1)) K, Q ≠ 0 → Q.IsHomogeneous d → (∀ x ∈ S, eval₂ σ (fun i => u i x) Q = 0) → ∃ (Q' : MvPolynomial (Fin (n + 1)) K) (d' : ℕ), (finSuccEquiv K n Q').coeff 0 ≠ 0 ∧ Q'.IsHomogeneous d' ∧ ∀ x ∈ S, eval₂ σ (fun i => u i x) Q' = 0

theorem **continuous_eval₂** {ι : Type} (Q : MvPolynomial ι K) : Continuous fun y : ι → ℂ => eval₂ σ y Q

theorem **isHomogeneous_expand** {ι : Type} {F : MvPolynomial ι K} {e : ℕ} (hF : F.IsHomogeneous e) (k : ℕ) : (expand k F).IsHomogeneous (k * e)

theorem **eval₂_expand'** {ι : Type} (F : MvPolynomial ι K) (k : ℕ) (y : ι → ℂ) : eval₂ σ y (expand k F) = eval₂ σ (fun i => y i ^ k) F

theorem **decay_le** : ∃ C : ℝ, ∀ x ∈ X.dom, ‖Xa.et x * Xa.th1 x ^ (Xa.i * (Xa.g0 / 2))‖ ^ 2 ≤ C * (∑ k, X.pnorm Xa.L (Xa.w k) x ^ 2) * (‖fun j => Xa.w j x‖ ^ (2 * (Xa.i * (Xa.g0 / 2)))) ^ 2

theorem **exists_rel** {K : Type} [Field K] (σ : K →+* ℂ) (hR : RelIn σ X Xa) (jj : Fin (nd X + 1) → Fin Xa.N) : ∃ (Q : MvPolynomial (Fin (nd X + 2)) K) (d : ℕ), (finSuccEquiv K (nd X + 1) Q).coeff 0 ≠ 0 ∧ Q.IsHomogeneous d ∧ ∀ x ∈ X.dom, eval₂ σ (Fin.cons (Xa.et x * Xa.th1 x ^ (Xa.i * (Xa.g0 / 2))) (fun a => Xa.w (jj a) x ^ (2 * (Xa.i * (Xa.g0 / 2))))) : Fin (nd X + 2) → ℂ) Q = 0

theorem **forms_not_small** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (K : Type) [Field K] (σ : K →+* ℂ) (t : Fin (nd X + 1) × Fin Xa.N → ℂ), (∀ Q : MvPolynomial (Fin (nd X + 1) × Fin Xa.N) K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0) → RelIn σ X Xa → let sf : Fin (nd X + 1) → Frame X.Ψ r → ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaFormsNotSmallWhereCombinationsVanishStatement

theorem **_root_.GlobalTheory.T.theta_forms_not_small_where_combinations_vanish** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (Xa : ArchInM X) (K : Type) [Field K] (σ : K →+* ℂ) (t : Fin (nd X + 1) × Fin Xa.N → ℂ), (∀ Q : MvPolynomial (Fin (nd X + 1) × Fin Xa.N) K, Q ≠ 0 → MvPolynomial.eval₂ σ t Q ≠ 0) → RelIn σ X Xa → let sf : Fin (nd X + 1) → Frame X.Ψ r → ℂ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaGeneratorPointwiseBoundOrthonormalFrame

theorem **isUnitFr_of_isOrtho** {E : Type} [Field E] [NumberField E] [IsCMField E] {r : ℕ} {X : Data E r} {x : Frame X.Ψ r} (ho : X.IsOrtho x) : IsUnitFr X x

theorem **sum_norm_coeff_rename_map** {K α β : Type} [Field K] (σ : K →+* ℂ) (e : α ≃ β) (P : MvPolynomial α K) : ∑ m ∈ (MvPolynomial.rename e (MvPolynomial.map σ P)).support, ‖(MvPolynomial.rename e (MvPolynomial.map σ P)).coeff m‖ = ∑ m ∈ P.support, ‖σ (P.coeff m)‖

theorem **theta_bound_ortho** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → 1 ≤ r → ∀ t : ℝ, 0 < t → (∀ k τ, (U.X k).c τ = t) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → ∀ (_ : StarRing (𝒪 E)) (nab : NabV Xo c), IsNab Xo c nab → ∀ (xx : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ f : Rul Xo c, IsThRule Xo c nab xx P f → ∀ k (x : (U.X k).dom), (U.X k).IsOrtho x.1 → ∀ lam : ℝ, 0 < lam → lam ^ 2 ≤ (U.X k).firstMinimumSq x.1 → ‖Formalism.L2.PjFree.toC (evR Xo c f (U.un k x))‖ ≤ (∑ m ∈ P.support, ‖σ (P.coeff m)‖) * (Real.pi * Real.exp 1 * t * (Formalism.L2.Pj.DF Dw a : ℕ)) ^ (((Formalism.L2.

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaGeneratorPointwiseBoundOrthonormalFrameStatement

theorem **_root_.GlobalTheory.T.thetaGenerator_pointwise_bound_orthonormal_frame** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) (σ : c.F →+* ℂ) (r : ℕ) (U : Formalism.L2.PjFree.Unf Xo c σ r), Formalism.L2.PjFree.GoodEmb c σ → Formalism.L2.PjFree.IsUnf U → 1 ≤ r → ∀ t : ℝ, 0 < t → (∀ k τ, (U.X k).c τ = t) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), Dom a n → ∀ (_ : StarRing (𝒪 E)) (nab : NabV Xo c), IsNab Xo c nab → ∀ (xx : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ f : Rul Xo c, IsThRule Xo c nab xx P f → ∀ k (x : (U.X k).dom), (U.X k).IsOrtho x.1 → ∀ lam : ℝ, 0 < lam → lam ^ 2 ≤ (U.X k).firstMinimumSq x.1 → ‖Formalism.L2.PjFree.toC (evR Xo c f (U.un k x))‖ ≤ (∑ m ∈ P.support, ‖σ (P.coeff m)‖) * (Real.pi * Real.exp 1 *

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaGeneratorsNoCommonZeroOfChartIndependence

theorem **noCommonZero_of_chartIndep** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → NoCommonZeroSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaGeneratorsNoCommonZeroOfChartIndependenceStatement

theorem **_root_.GlobalTheory.T.thetaGenerators_noCommonZero_of_chartIndependence** :
Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → NoCommonZeroSt

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaGroupShiftHalfTorsionRoot

theorem **exists_lin_iff** (hL : L.IsLB) (θ : ThetaGrp A L k T) : ∃ f : SecM A L k T →ₗ[Γ(T.left, ⊤)] SecM A L k T, ∀ s s', IsActOf A L k θ s s' ↔ f (SecM.ofSec A s) = SecM.ofSec A s'

theorem **exists_isActOf** (hL : L.IsLB) (θ : ThetaGrp A L k T) (s : LB.Sec (LPow A L k T)) : ∃ s', IsActOf A L k θ s s'

theorem **act_smul** (hL : L.IsLB) (θ : ThetaGrp A L k T) (c : Γ(T.left, ⊤)) {a b a' b' : LB.Sec (LPow A L k T)} (h : ∀ j, b.f j = liftFun A _ c * a.f j) (ha : IsActOf A L k θ a a') (hb : IsActOf A L k θ b b') : ∀ j, b'.f j = liftFun A _ c * a'.f j

theorem **liftFun_one** (W : (T ⊗ A).left.Opens) : liftFun A W (1 : Γ(T.left, ⊤)) = 1

theorem **toΓ_bijective** (B : Type) [CommRing B] : Function.Bijective (toΓ B)

theorem **isUUOver_shift** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (B : Type) [CommRing B] (x0 : Sp B ⟶ Xo.X) (N : ℕ), Odd N → ∀ (s : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 (ptO Xo x0))) (t u z : ptO Xo x0 ⟶ Xo.univ.t.G) (e : Γ((ptO Xo x0).left, ⊤)ˣ), t ^ (N ^ 2) = 1 → u ^ (N ^ 2) = 1 → z ^ (N ^ 2) = 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) t u 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) u z e → ∀ F F', IsUUOver Xo.univ.t.G Xo.univ.t.PP N s z (t * u) F → IsUUOver Xo.univ.t.G Xo.univ.t.PP N s (u * z) t F' → ∃ r : B, (toΓ B r) ^ 2 = (e : Γ((ptO Xo x0).left, ⊤)) ∧ r ^ (N ^ 2) = 1 ∧ ∀ j, F.f j = liftFun Xo.univ.t.G _ (T

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaGroupShiftHalfTorsionRootStatement

theorem **_root_.GlobalTheory.T.thetaGroup_shift_halfTorsion_root** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (B : Type) [CommRing B] (x0 : Sp B ⟶ Xo.X) (N : ℕ), Odd N → ∀ (s : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L 1 (ptO Xo x0))) (t u z : ptO Xo x0 ⟶ Xo.univ.t.G) (e : Γ((ptO Xo x0).left, ⊤)ˣ), t ^ (N ^ 2) = 1 → u ^ (N ^ 2) = 1 → z ^ (N ^ 2) = 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) t u 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) u z e → ∀ F F', IsUUOver Xo.univ.t.G Xo.univ.t.PP N s z (t * u) F → IsUUOver Xo.univ.t.G Xo.univ.t.PP N s (u * z) t F' → ∃ r : B, (toΓ B r) ^ 2 = (e : Γ((ptO Xo x0).left, ⊤)) ∧ r ^ (N ^ 2) = 1 ∧ ∀ j, F.f j = liftFun Xo.univ.t.G _ (T

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaIdealLatticeEqSumLevelTranslates

theorem **mem_of_eigen_mem** {A W : Type} [AddCommGroup A] [AddCommGroup W] [Module ℂ W] (N : AddSubgroup A) [Finite (A ⧸ N)] (S : A → W →ₗ[ℂ] W) (hS : ∀ u v f, S u (S v f) = S (u + v) f) (H : AddSubgroup (AddChar (A ⧸ N) ℂ)) (hH : ∀ q ≠ 0, ∃ ψ ∈ H, ψ q ≠ 1) (f : W) (hf : ∀ u ∈ N, S u f = f) (U V : Submodule ℂ W) (hU : ∀ u, S u f ∈ U) (hV : ∀ ψ ∈ H, ∀ h ∈ U, (∀ u : A, S u h = ψ (u : A ⧸ N) • h) → h ∈ V) : f ∈ V

theorem **mem_span_prod** {Z J : Type} (φ : J → Z → ℂ) (Y : Set J) (hY : Y.Finite) : ∀ (n : ℕ) (F : (Fin n → Z) → ℂ), (∀ b z, (fun t => F (Function.update z b t)) ∈ Submodule.span ℂ (φ '' Y)) → F ∈ Submodule.span ℂ ((fun (y : Fin n → J) (z : Fin n → Z) => ∏ b, φ (y b) (z b)) '' {y | ∀ b, y b ∈ Y})

theorem **thetaSignExpEF_add_selfDual_lattice** {L : Lt E r} (hL : SelfD X L) {w : E} (hw : IsIntegral ℤ w) (hw1 : w + cc w = 1) {u v : Fin r → E} (hu : u ∈ L) (hv : v ∈ L) : eps1 X w (u + v) = eps1 X w u * eps1 X w v * cexp (-(Real.pi * I * ((EF X u v : ℚ) : ℂ)))

theorem **EF_zero_left** (u : Fin r → E) : EF X 0 u = 0

theorem **EF_zero_right** (u : Fin r → E) : EF X u 0 = 0

theorem **EF_int_of_mem** {L : Lt E r} (hL : SelfD X L) {𝔐 : Ideal (𝒪 E)} {M : ℕ} (hM : M ≠ 0) (h𝔐 : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) {y u : Fin r → E} (hy : (M : E) • y ∈ 𝔐 • L) (hu : u ∈ 𝔐 • L) : ∃ k : ℤ, EF X y u = k

theorem **mem_of_EF_int** {L : Lt E r} (hL : SelfD X L) {𝔐 : Ideal (𝒪 E)} {M : ℕ} (hM : M ≠ 0) (h𝔐 : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) {u : Fin r → E} (h : ∀ y : Fin r → E, (M : E) • y ∈ 𝔐 • L → ∃ k : ℤ, EF X y u = k) : u ∈ 𝔐 • L

theorem **pv_add** (u v : Fin r → E) : pv X idx x (u + v) = pv X idx x u + pv X idx x v

theorem **pv_zero** : pv X idx x 0 = 0

theorem **fockL_fockL** (hs : IsSympl X bas) (hd : x ∈ X.dom) (u v : Fin r → E) (f : (Fin g → ℂ) → ℂ) : fockL X bas idx x (pv X idx x u) (fockL X bas idx x (pv X idx x v) f) = cexp (-(Real.pi * I * ((EF X u v : ℚ) : ℂ))) • fockL X bas idx x (pv X idx x (u + v)) f

theorem **exists_span_one** (hs : IsSympl X bas) (hd : x ∈ X.dom) {w : E} (hw : IsIntegral ℤ w) (hw1 : w + cc w = 1) {L : Lt E r} (hL : SelfD X L) {𝔐 : Ideal (𝒪 E)} {M : ℕ} (hM : M ≠ 0) (h𝔐 : 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)}) {θ : (Fin g → ℂ) → ℂ} (hθ : IsTheta1 X bas idx w x L θ) (h0 : θ ≠ 0) : ∃ Y : Set (Fin r → E), Y.Finite ∧ (∀ y ∈ Y, (M : E) • y ∈ 𝔐 • L) ∧ ∀ f, IsTheta1 X bas idx w x (𝔐 • L) f → f ∈ Submodule.span ℂ ((fun y => fock X bas idx x 1 (pv X idx x y) θ) '' Y)

theorem **theta_eq_sum_translates** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (𝔐 : Ideal (𝒪 E)) (M : ℕ), Odd M → 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)} → ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∀ (θ : (Fin g → ℂ) → ℂ) (κ : ℂ), IsTheta1 X bas idx (w : E) x L θ → MeanOne X bas idx x L θ → (∀ z : Fin n → Fin g → ℂ, ∏ b, θ (z b) = κ * thA X bas idx v x z) → ∀ F : (Fin n → Fin g → ℂ) → ℂ, Differentiable ℂ F → (∀ u : Matrix (Fin n) (Fin r) E, (∀ b, u b ∈ 𝔐 • L) → ∀ z, F z = epsW X (w : E) u * cexp (-(Real.pi * ∑ b, hx X bas idx x (z b + (1 / 2 : ℂ) • pv X idx x (u b)) (pv X idx x (u b)))) *

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaIdealLatticeEqSumLevelTranslatesStatement

theorem **_root_.GlobalTheory.T.theta_ideal_lattice_eq_sum_level_translates** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : Data E r) (g q0 : ℕ) (hr : X.p0 + q0 = r) (bas : (Fin g ⊕ Fin g → ℤ) ≃ₗ[ℤ] X.Λ) (idx : Fin g ≃ X.Ψ × Fin r) (n : ℕ), IsSympl X bas → Even n → ∀ w : 𝒪 E, (w : E) + cc (w : E) = 1 → ∀ L : Lt E r, SelfD X L → ∀ x ∈ X.dom, X.IsOrtho x → ∀ t : ℝ, 0 < t → (∀ τ, X.c τ = t) → ∀ (𝔐 : Ideal (𝒪 E)) (M : ℕ), Odd M → 𝔐 * conjI 𝔐 = Ideal.span {(M : 𝒪 E)} → ∀ v : TF n g, IsVac X bas (w : E) (tens (stdL E n) L) v → ∀ (θ : (Fin g → ℂ) → ℂ) (κ : ℂ), IsTheta1 X bas idx (w : E) x L θ → MeanOne X bas idx x L θ → (∀ z : Fin n → Fin g → ℂ, ∏ b, θ (z b) = κ * thA X bas idx v x z) → ∀ F : (Fin n → Fin g → ℂ) → ℂ, Differentiable ℂ F → (∀ u : Matrix (Fin n) (Fin r) E, (∀ b, u b ∈ 𝔐 • L) → ∀ z, F z = epsW X (w : E) u * cexp (-(Real.pi * ∑ b, hx X bas idx x (z b + (1 / 2 : ℂ)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaIndexBoundHypothesesAtInstance

theorem **thetaInputs_rec** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → LawsSh T → 0 < badN D → (∀ j, InR D j → ∀ n, (T.dδ j n).l₂.A = 1) → (∀ j, InR D j → ∀ k k', ((T.U j T.σ₀).X k).Gm = ((T.U j T.σ₀).X k').Gm ∧ ((T.U j T.σ₀).X k).ξ0 = ((T.U j T.σ₀).X k').ξ0) → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j

: ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (Gc Prw : ℝ), FamilyHypotheses.OwnPr D p → FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → PageAt large₂ C2 A2 → 4 * H.CK ≤ WeightFamilies.PPv D large₂ C2 A2 ^ 2 * D.R.f → (∀ w2, IsG2 wL wF w2 → ThInt (recPJ h j hj) hinj EX w2) → (j < D.R.m → ∀ w1, IsDoubledWeightWithShift wL w1 nW → ThInt (recPJ h j hj) hinj EX w1) → H2 (recPJ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaIndexBoundHypothesesAtInstanceStatement

theorem **_root_.GlobalTheory.T.thetaIndexBound_hypotheses_at_instance** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (D : D7) (T : Top D) (h : LawsP T), GamAt T.toCore → LawsSh T → 0 < badN D → (∀ j, InR D j → ∀ n, (T.dδ j n).l₂.A = 1) → (∀ j, InR D j → ∀ k k', ((T.U j T.σ₀).X k).Gm = ((T.U j T.σ₀).X k').Gm ∧ ((T.U j T.σ₀).X k).ξ0 = ((T.U j T.σ₀).X k').ξ0) → ∀ (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (p : ℕ) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (Gc Prw : ℝ), FamilyHypotheses.OwnPr D p → FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → PageAt large₂ C2 A2 → 4 * H.CK ≤ WeightFamilies.PPv D large₂ C2 A2 ^ 2 * D.R.f → (∀ w2, IsG2 wL wF w2 → ThInt (recPJ h j hj) hinj EX w2) → (j < D.R.m → ∀ w1, IsDoubledWeightWithShift wL w1 nW →

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaLatticeWeightScalarNeZeroAndLogBounds

private theorem **log_norm_sOf_eq_DF_log_add_log_pPow** (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (w : WeightFamilies.Wt D j) (σ : B.KO →+* ℂ) : Real.log ‖σ (sOf B hinj EX w : B.KO)‖ = (WeightAlgebra.DF w : ℝ) / 2 * Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ)) + Real.log (pPow D (c2 D w) (Formalism.MainNotions.DIV.Cons.cHalf w) (Formalism.MainNotions.DIV.Cons.cHalf w) : ℝ)

theorem **sOf_ne_zero_and_log_bounds** : ∀ (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (w : WeightFamilies.Wt D j), sOf B hinj EX w ≠ 0 ∧ ∀ σ : B.KO →+* ℂ, 0 ≤ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ∧ (WeightAlgebra.DF w : ℝ) / 2 * Real.log (qq D) ≤ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ∧ (WeightAlgebra.DF w : ℝ) / 2 * Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ)) ≤ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ∧ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ≤ lamStar D * w.size

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaLatticeWeightScalarNeZeroAndLogBoundsStatement

theorem **_root_.GlobalTheory.T.thetaLatticeWeightScalar_ne_zero_and_log_bounds** : ∀ (D : D7) (B : WeightFamilies.Prov D) (hinj : Function.Injective D.what₂.OtoLL) (EX : ErrX D) (j : ℕ) (w : WeightFamilies.Wt D j), sOf B hinj EX w ≠ 0 ∧ ∀ σ : B.KO →+* ℂ, 0 ≤ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ∧ (WeightAlgebra.DF w : ℝ) / 2 * Real.log (qq D) ≤ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ∧ (WeightAlgebra.DF w : ℝ) / 2 * Real.log ((D.host.g 0 : ℝ) * (D.host.g 1 : ℝ)) ≤ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ∧ Real.log ‖σ (sOf B hinj EX w : B.KO)‖ ≤ lamStar D * w.size

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaLetterEquivConjugate

theorem **halfRowEquiv_of_mem** {τ : ι} (h : Ψ τ) (i : Fin (dim τ)) : halfRowEquiv hσ hΨ hp hq ⟨τ, i⟩ = (⟨τ, h⟩, Sum.inl (i.cast (hp _ h)))

theorem **halfRowEquiv_invol** {τ : ι} (h : Ψ τ) (i : Fin (dim (σ τ))) : halfRowEquiv hσ hΨ hp hq ⟨σ τ, i⟩ = (⟨τ, h⟩, Sum.inr (i.cast (hq _ h)))

theorem **typeLetterEquiv_apply** {τ : ι} {i : Fin (dim τ)} {t : {τ // Ψ τ}} {r : Fin p ⊕ Fin q} (h : halfRowEquiv hσ hΨ hp hq ⟨τ, i⟩ = (t, r)) (b : Fin n) : typeLetterEquiv hσ hΨ hp hq n ⟨τ, i, b⟩ = (t, r, b)

theorem **half_cases** (τ : ι) : Ψ τ ∨ ∃ τ', Ψ τ' ∧ τ = σ τ'

theorem **rename_lap_one** {τ : ι} (h : Ψ τ) (i : Fin (dim τ)) (j : Fin (dim (σ τ))) (P : MvPolynomial (Formalism.L2.Pj.Letter ι dim n) K) : rename (typeLetterEquiv hσ hΨ hp hq n) (Formalism.L2.Pj.lap dim n K 1 τ (σ τ) i j P) = laplacian ⟨τ, h⟩ (i.cast (hp _ h)) (j.cast (hq _ h)) (rename (typeLetterEquiv hσ hΨ hp hq n) P)

theorem **rename_rowShift** {τ : ι} {i i' : Fin (dim τ)} {t : {τ // Ψ τ}} {r r' : Fin p ⊕ Fin q} (h : halfRowEquiv hσ hΨ hp hq ⟨τ, i⟩ = (t, r')) (h' : halfRowEquiv hσ hΨ hp hq ⟨τ, i'⟩ = (t, r)) (s : K) (P : MvPolynomial (Formalism.L2.Pj.Letter ι dim n) K) : rename (typeLetterEquiv hσ hΨ hp hq n) (Formalism.L2.Pj.rowShift dim n K τ i i' s P) = rowAdd t r r' s (rename (typeLetterEquiv hσ hΨ hp hq n) P)

theorem **rowDeg_iff_rename** {τ : ι} {i : Fin (dim τ)} {t : {τ // Ψ τ}} {r : Fin p ⊕ Fin q} (h : halfRowEquiv hσ hΨ hp hq ⟨τ, i⟩ = (t, r)) (P : MvPolynomial (Formalism.L2.Pj.Letter ι dim n) K) (d : ℕ) : (∀ m ∈ P.support, ∑ b : Fin n, m ⟨τ, i, b⟩ = d) ↔ IsWeightedHomogeneous (rowInd t r) (rename (typeLetterEquiv hσ hΨ hp hq n) P) d

theorem **mem_HHa_iff_rename** [CharZero K] (a : ι → ℕ → ℕ) (P : MvPolynomial (Formalism.L2.Pj.Letter ι dim n) K) : P ∈ Formalism.L2.Pj.HHa dim n K σ Ψ a ↔ IsHarmonic (rename (typeLetterEquiv hσ hΨ hp hq n) P) ∧ (∀ t r, IsWeightedHomogeneous (rowInd t r) (rename (typeLetterEquiv hσ hΨ hp hq n) P) (Sum.elim (fun i : Fin p => a t.1 i) (fun j : Fin q => a (σ t.1) j) r)) ∧ ∀ t r r', RowLT r r' → ∀ s : ℤ, rowAdd t r r' (s : K) (rename (typeLetterEquiv hσ hΨ hp hq n) P) = rename (typeLetterEquiv hσ hΨ hp hq n) P

private theorem **X_mul_pderiv_mem_restrictTotalDegree** {S' F' : Type*} [Field F'] [DecidableEq S'] {N : ℕ} (v u : S') {Q : MvPolynomial S' F'} (hQ : Q ∈ restrictTotalDegree S' F' N) : X v * pderiv u Q ∈ restrictTotalDegree S' F' N

theorem **HHa_eq_zero_or_rowHighest** [CharZero K] [Fintype ι] [Fintype {τ // Ψ τ}] (a : ι → ℕ → ℕ) : (∀ P : MvPolynomial (Formalism.L2.Pj.Letter ι dim n) K, P ∈ Formalism.L2.Pj.HHa dim n K σ Ψ a → P = 0) ∨ ∃ (k : {τ // Ψ τ} → Fin p → ℕ) (k' : {τ // Ψ τ} → Fin q → ℕ), typeDegree k k' = ∑ τ, ∑ i : Fin (dim τ), a τ i ∧ ∀ P : MvPolynomial (Formalism.L2.Pj.Letter ι dim n) K, P ∈ Formalism.L2.Pj.HHa dim n K σ Ψ a ↔ rename (typeLetterEquiv hσ hΨ hp hq n) P ∈ rowHighest K k k'

theorem **thetaLetterEquiv_of_mem** {τ : E →+* ℂ} (h : InPsi H τ) (i : Fin (Dw.mult τ)) (k : Fin n) : thetaLetterEquiv Dw n ⟨τ, i, k⟩ = xVar ⟨τ, h⟩ (i.cast (mult_of_mem Dw h)) k

theorem **thetaLetterEquiv_conjugate** {τ : E →+* ℂ} (h : InPsi H τ) (i : Fin (Dw.mult (ComplexEmbedding.conjugate τ))) (k : Fin n) : thetaLetterEquiv Dw n (ComplexEmbedding.conjugate τ, i, k) = yVar ⟨τ, h⟩ (i.cast (mult_conjEmb_of_mem Dw h)) k

theorem **rename_letB_of_mem** {τ : E →+* ℂ} (h : InPsi H τ) (i : Fin (Dw.mult τ)) (v : Fin n → E) : rename (thetaLetterEquiv Dw n) (letB c τ i v) = xForm ⟨τ, h⟩ (i.cast (mult_of_mem Dw h)) fun k => embF c τ (v k)

theorem **rename_letB_conjugate** {τ : E →+* ℂ} (h : InPsi H τ) (i : Fin (Dw.mult (ComplexEmbedding.conjugate τ))) (v : Fin n → E) : rename (thetaLetterEquiv Dw n) (letB c (ComplexEmbedding.conjugate τ) i v) = yForm ⟨τ, h⟩ (i.cast (mult_conjEmb_of_mem Dw h)) fun k => embF c τ (FD.cc (v k))

theorem **mem_symO_iff_rename** (Λ : GenLat E n) (P : MvPolynomial (LtN Dw n) c.F) : P ∈ Formalism.L2.Pj.symO c Λ ↔ rename (thetaLetterEquiv Dw n) P ∈ GlobalTheory.Harm.symO Dw.p Dw.q (fun t : {τ : E →+* ℂ // InPsi H τ} => embF c t.1) Λ.La

theorem **theta_letter_dictionary** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (c : Coeff l) (n : ℕ) [Fintype {τ : E →+* ℂ // InPsi H τ}], ∃ e : LtN Dw n ≃ GlobalTheory.Harm.Var {τ : E →+* ℂ // InPsi H τ} Dw.p Dw.q (Fin n), (∀ (Λ : GenLat E n) (P : MvPolynomial (LtN Dw n) c.F), P ∈ Formalism.L2.Pj.symO c Λ ↔ MvPolynomial.rename e P ∈

GlobalTheory.Harm.symO Dw.p Dw.q (fun t : {τ : E →+* ℂ // InPsi H τ} => embF c t.1) Λ.La) ∧ ∀ (a : (E →+* ℂ) × ℕ → ℕ), (∀ P : MvPolynomial (LtN Dw n) c.F, P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → P = 0) ∨ ∃ (k : {τ : E →+* ℂ // InPsi H τ} → Fin Dw.p → ℕ) (k' : {τ : E →+* ℂ // InPsi H τ} → Fin Dw.q → ℕ), GlobalTheory.Harm.typeDegree k k' = DF Dw a ∧ ∀ P : MvPolynomial (LtN Dw n) c.F, P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) ↔ MvP

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaNetDimensionAndPointCountBound

theorem **net_count_dim** {D : D7} {B : WeightFamilies.Prov D} {j : ℕ} {ka : WeightFamilies.Wt D j} (Nt : NetNorms.NetData B j ka) (σ : B.KO →+* ℂ) (c : ℝ) (hM : NetNorms.NetMax Nt) (hS : NetNorms.NetSize Nt c) (hI : NetNorms.NetInj Nt σ) : Real.log (Fintype.card (Nt.NN σ) : ℝ) ≤ c ∧ ∀ n : ℕ, n ≤ Module.finrank B.KO (B.Cusp j ka) → Real.log n ≤ c

theorem **le_finrank_of_shareRd** {D : D7} {B : WeightFamilies.Prov D} {P : ModuliIntegralCharts.Prov D} {j : ℕ} (X : PJ D B P j) (ka : WeightFamilies.Wt D j) (N : Submodule (𝒪 B.KO) (B.Cusp j ka)) (Tn del : ℕ) (hT : WeightFamilies.dimY D j + 1 ≤ Tn) (hR : ShareRd (D

theorem **net_at_run** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), PageOK large₂ (recPJ h j hj) H C2 A2 CT ka wL wF Tn nP nW h0 → ∃ (Nt : NetNorms.NetData (recB h.toLawsCore) j ka) (σ₀ : (recB h.toLawsCore).KO →+* ℂ), NetNorms.NetMax Nt ∧ NetNorms.NetLeSup Nt ∧ NetNorms.NetPageG Nt large₂ 1041 3 ∧ NetNorms.NetInj Nt σ₀ ∧ Real.log (Fintype.card (Nt.NN σ₀) : ℝ) ≤ pdR large₂ ka ∧ DeepSubMul (recPJ h j hj) ∧ Real.log (Module.finrank (recB h.toLawsCore).KO ((recB h.toLawsCore).Cusp j ka) : ℝ) ≤ 1041 * Zb D large₂ ^ 3 ∧ ∀ del : ℕ, LatAt (recB h.toLawsCore) j ka → ShareRd (D

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaNetDimensionAndPointCountBoundStatement

theorem **_root_.GlobalTheory.T.thetaNet_dimension_and_pointCount_bound** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), PageOK large₂ (recPJ h j hj) H C2 A2 CT ka wL wF Tn nP nW h0 → ∃ (Nt : NetNorms.NetData (recB h.toLawsCore) j ka) (σ₀ : (recB h.toLawsCore).KO →+* ℂ), NetNorms.NetMax Nt ∧ NetNorms.NetLeSup Nt ∧ NetNorms.NetPageG Nt large₂ 1041 3 ∧ NetNorms.NetInj Nt σ₀ ∧ Real.log (Fintype.card (Nt.NN σ₀) : ℝ) ≤ pdR large₂ ka ∧ DeepSubMul (recPJ h j hj) ∧ Real.log (Module.finrank (recB h.toLawsCore).KO ((recB h.toLawsCore).Cusp j ka) : ℝ) ≤ 1041 * Zb D large₂ ^ 3 ∧ ∀ del : ℕ, LatAt (recB h.toLawsCore) j ka → ShareRd (D

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaNullValueExistsUniqueNaturalLinear

theorem **isValueAt_congr** {T : Over X} {s : LB.Sec (LPow A L a T)} {z z' : T ⟶ A} (e : z = z') {τ : LB.Gen (LB.LinComb L (Y

private lemma **gm_id_eq_germ_map** {Y : Scheme.{0}} {U V : Y.Opens} (hVU : V ≤ U) (y : Y) (hy : y ∈ V) (x : Γ(Y, U)ˣ) : ((gm (𝟙 Y) y U x : (Y.presheaf.stalk y)ˣ) : Y.presheaf.stalk y) = Y.presheaf.germ V y hy ((Y.presheaf.map (homOfLE hVU).op).hom (x : Γ(Y, U)))

private lemma **gm_id_eq_of_units_map_eq** {Y : Scheme.{0}} {U₁ U₂ V : Y.Opens} (h₁ : V ≤ U₁) (h₂ : V ≤ U₂) (x₁ : Γ(Y, U₁)ˣ) (x₂ : Γ(Y, U₂)ˣ) (y : Y) (hy : y ∈ V) (e : Units.map ((Y.presheaf.map (homOfLE h₁).op).hom : Γ(Y, U₁) →* Γ(Y, V)) x₁ = Units.map ((Y.presheaf.map (homOfLE h₂).op).hom : Γ(Y, U₂) →* Γ(Y, V)) x₂) : gm (𝟙 Y) y U₁ x₁ = gm (𝟙 Y) y U₂ x₂

theorem **isValueAt_one_pull** (ρ : LB.Gen (L.pull (eSec A))) {T T' : Over X} (g : T' ⟶ T) {s : LB.Sec (LPow A L a T)} {s' : LB.Sec (LPow A L a T')} (hs : IsPullSec A L a (g ▷ A) (whiskerRight_snd g A) s s') {v : Γ(T.left, ⊤)} (hv : IsValueAt A L a s 1 (rigT A L ρ T a) v) : IsValueAt A L a s' 1 (rigT A L ρ T' a) (g.left.appTop.hom v)

theorem **baseChange** {N : ℕ} {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) {y : FrPt Xo c S} {z : FrPt Xo c T} (hm : FrMap ψ y z) (p : ptO Xo y.x ⟶ Xo.univ.t.G) (p' : ptO Xo z.x ⟶ Xo.univ.t.G) (hp : p'.left = spMap ψ.toRingHom ≫ p.left) {w : S} (hw : IsNullVal Xo c N y p w) : IsNullVal Xo c N z p' (ψ w)

theorem **triangular** {N : ℕ} {S : Type} [CommRing S] [Algebra c.F S] {y y' : FrPt Xo c S} {t : Ix Dw → Sˣ} (hb : FrBor t y y') (p : ptO Xo y.x ⟶ Xo.univ.t.G) (p' : ptO Xo y'.x ⟶ Xo.univ.t.G) (hp : p'.left = p.left) {w : S} (hw : IsNullVal Xo c N y p w) : IsNullVal Xo c N y' p' ((t none : Sˣ) * w)

theorem **nullValue_existsUnique_natural_linear** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (N : ℕ), Odd N → (∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (p : ptO Xo y.x ⟶ Xo.univ.t.G), p ^ (N ^ 2) = 1 → ∃! w, IsNullVal Xo c N y p w) ∧ (∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S) (z : FrPt Xo c T), FrMap ψ y z → ∀ (p : ptO Xo y.x ⟶ Xo.univ.t.G) (p' : ptO Xo z.x ⟶ Xo.univ.t.G), p'.left = spMap ψ.toRingHom ≫ p.left → ∀ w, IsNullVal Xo c N y p w → IsNullVal Xo c N z p' (ψ w)) ∧ ∀ (S : Type) [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (t : Ix Dw → Sˣ), FrBor t y y' → ∀ (p : ptO Xo y.x ⟶ Xo.univ.t.G) (p' : ptO Xo y'.x ⟶ Xo.univ.t.G), p'.left = p.left → ∀ w, IsNullVal Xo c N y p w → IsNullVal Xo c N y' p' ((t none : Sˣ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaNullValueExistsUniqueNaturalLinearStatement

theorem **_root_.GlobalTheory.T.thetaNullValue_existsUnique_natural_linear** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) (N : ℕ), Odd N → (∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) (p : ptO Xo y.x ⟶ Xo.univ.t.G), p ^ (N ^ 2) = 1 → ∃! w, IsNullVal Xo c N y p w) ∧ (∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (ψ : S →ₐ[c.F] T) (y : FrPt Xo c S) (z : FrPt Xo c T), FrMap ψ y z → ∀ (p : ptO Xo y.x ⟶ Xo.univ.t.G) (p' : ptO Xo z.x ⟶ Xo.univ.t.G), p'.left = spMap ψ.toRingHom ≫ p.left → ∀ w, IsNullVal Xo c N y p w → IsNullVal Xo c N z p' (ψ w)) ∧ ∀ (S : Type) [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (t : Ix Dw → Sˣ), FrBor t y y' → ∀ (p : ptO Xo y.x ⟶ Xo.univ.t.G) (p' : ptO Xo y'.x ⟶ Xo.univ.t.G), p'.left = p.left → ∀ w, IsNullVal Xo c N y p w → IsNullVal

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRuleExistsAndEval

theorem **thetaRule_exists_and_eval** : Formalism.MainNotions.ModuliIntegralCharts.H1.ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (xx : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → (∃ f : Rul Xo c, IsThRule Xo c nab xx P f ∧ f ∈ sec Xo c a n) ∧ ∀ f : Rul Xo c, IsThRule Xo c nab xx P f → ∀ (k : Type) [Field k] [Algebra c.F k] (y : FrPt Xo c k), ThValAt Xo c nab y xx P (evR Xo c f y)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRuleMemSec

theorem **thetaRule_mem_sec** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), ∀ f ∈ thGen Xo c nab a n, f ∈ sec Xo c a n

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRuleValuePeriods

theorem **thetaRule_value_periods** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (σ : c.F →+* ℂ) (r : ℕ) (U : Unf Xo c σ r) (k : U.Comp) (xf : (U.X

k).dom) (W : UnfAt U k xf) (Wu : UnitAt W) (nab : NabV Xo c), IsNab Xo c nab → FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRuleValueTestRingEqThetaValue

theorem **thValAt_evR_of_fromTest** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ f : Rul Xo c, IsThRule Xo c nab x P f → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRuleValueTestRingEqThetaValueStatement

theorem **_root_.GlobalTheory.T.thetaRuleValue_testRing_eq_thetaValue** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ f : Rul Xo c, IsThRule Xo c nab x P f → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y : FrPt Xo c S), FromTest (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRulesLinearAndLevelCongruence

theorem **monomial_degree_of_mem_harm** {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ harmA (Dw

theorem **totalDegree_of_mem_harm** {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ harmA (Dw

theorem **thExpr_eq_sum_superset** (nab : NabV Xo c) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y.x))) {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ harmA (Dw

theorem **thExpr_linear** (nab : NabV Xo c) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y.x))) {N : ℕ} (cf : Fin N → c.F) (Q : Fin N → MvPolynomial (LtN Dw n) c.F) (hQ : ∀ i, Q i ∈ harmA (Dw

theorem **thValAt_linear** (hC : ChartIndepSt2) [StarRing (𝒪 E)] (nab : NabV Xo c) (hnab : IsNab Xo c nab) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (x : Fin n → Dw.La) {N : ℕ} (cf : Fin N → c.F) (Q : Fin N → MvPolynomial (LtN Dw n) c.F) (hQ : ∀ i, Q i ∈ harmA (Dw

theorem **levPt_eq_of_sub_mem** {S : Type} [CommRing S] (x0 : Sp S ⟶ Xo.X) {u u' : Dw.La} (h : u − u' ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) : levPt Xo x0 u = levPt Xo x0 u'

theorem **isThRule_linear_and_congr** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), (∀ (x : Fin n → Dw.La) (m : ℕ) (cf : Fin m → c.F) (Q : Fin m → MvPolynomial (LtN Dw n) c.F) (f : Rul Xo c) (g : Fin m → Rul Xo c), (∀ i, Q i ∈ harmA c a n) → (∑ i, cf i • Q i) ∈ harmA c a n → IsThRule Xo c nab x (∑ i, cf i • Q i) f → (∀ i, IsThRule Xo c nab x (Q i) (g i)) → f = ∑ i, cf i • g i) ∧ ∀ (x x' : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F) (f : Rul Xo c), (∀ b, x b − x' b ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) → IsThRule Xo c nab x P f → IsThRule Xo c nab x' P f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRulesLinearAndLevelCongruenceStatement

theorem **_root_.GlobalTheory.T.theta_rules_linear_and_level_congruence** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ), (∀ (x : Fin n → Dw.La) (m : ℕ) (cf : Fin m → c.F) (Q : Fin m → MvPolynomial (LtN Dw n) c.F) (f : Rul Xo c) (g : Fin m → Rul Xo c), (∀ i, Q i ∈ harmA c a n) → (∑ i, cf i • Q i) ∈ harmA c a n → IsThRule Xo c nab x (∑ i, cf i • Q i) f → (∀ i, IsThRule Xo c nab x (Q i) (g i)) → f = ∑ i, cf i • g i) ∧ ∀ (x x' : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F) (f : Rul Xo c), (∀ b, x b − x' b ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) → IsThRule Xo c nab x P f → IsThRule Xo c nab x' P f

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRunParametersAtTypePrimesProduct

theorem **lower_real** (R P d a r q : ℝ) (hR : 0 ≤ R) (hd : 1 ≤ d) (ha : 0 ≤ a) (hr : 0 ≤ r) (h1 : 90 * (a + 1) ≤ P) (h2 : r ≤ P) (hq : R ^ 4 * P ^ 3 * d ^ 2 ≤ q) : R ^ 4 * (90 * (a + d)) * d * r ≤ q

theorem **run_at_record** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (K : KAdm) (Ch Ah Cq Aq : ℝ) (S : Scal B) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → TieC K H Ch Ah Cq Aq C2 A2 CR → PageAt large₂ C2 A2 → EtaleWeightGaps.PageSupp D large₂ B C2 A2 nW S wL → RunOK D.host.E D.R.r (WeightAlgebra.DF wL) (D.host.g 0 * D.host.g 1) (Real.sqrt (90 * (D.R.f * D.R.r + WeightAlgebra.DF wL))) D.R.m ∧ 8100 * Real.pi * wL.n * (D.R.m : ℝ) ^ 4 * (I2.dstar (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaRunParametersAtTypePrimesProductStatement

theorem **_root_.GlobalTheory.T.thetaRunParameters_at_typePrimes_product** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (K : KAdm) (Ch Ah Cq Aq : ℝ) (S : Scal B) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → TieC K H Ch Ah Cq Aq C2 A2 CR → PageAt large₂ C2 A2 → EtaleWeightGaps.PageSupp D large₂ B C2 A2 nW S wL → RunOK D.host.E D.R.r (WeightAlgebra.DF wL) (D.host.g 0 * D.host.g 1) (Real.sqrt (90 * (D.R.f * D.R.r + WeightAlgebra.DF wL))) D.R.m ∧ 8100 * Real.pi * wL.n * (D.R.m : ℝ) ^ 4 * (I2.dstar (E

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueAnalyticFrameEqPolyDerivFock

theorem **AnFr.e_natCast_smul_eq_pow** {S : Scheme.{0}} {A : Over S} [GrpObj A] {L : RigLB A} {x0 : Sp ℂ ⟶ S} {g : ℕ} (Φ : AnFr A L x0 g) (z : Fin g → ℂ) (n : ℕ) : Φ.e ((n : ℂ) • z) = Φ.e z ^ n

theorem **thValAt_eq_polyDeriv_fock** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ mk : Dw.La →+ (Fin g → ℂ), (∀ a, mk a ∈ Φ.lat) → (∀ (α : 𝒪 E) a, mk (α • a) = tyAct ty α (mk a)) → (∀ a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La), levPt Xo y.x a = Φ.e ((l.M : ℂ)⁻¹ • mk a)) → ∀ nab : NabV Xo c, IsNab Xo c nab → ∀ θ, Φ.toAnFr.Rd 1 y.s θ → ∀ (n : ℕ) (x x2 : Fin n → Dw.La), (∀ b, x2 b ∈ (Ideal.span {(2 : 𝒪 E)} ∗ (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La)) → (∀ b, x2 b − x b ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ v : ℂ, ThValAt Xo c nab y x P v → v = ((l.M : ℂ)⁻

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueAnalyticFrameEqPolyDerivFockStatement

theorem **_root_.GlobalTheory.T.thetaValue_analyticFrame_eq_polyDeriv_fock** : ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Geo.Tor.Level H) (Xo : Moduli Dw l) (c : Coeff l) [Algebra c.F ℂ] [StarRing (𝒪 E)] (y : FrPt Xo c ℂ) (g : ℕ) (ty : Fin g → (E →+* ℂ)) (el : Fin g ≃ Lt1 Dw) (Φ : AnPt y g ty el), ∀ mk : Dw.La →+ (Fin g → ℂ), (∀ a, mk a ∈

Φ.lat) → (∀ (α : 𝒪 E) a, mk (α • a) = tyAct ty α (mk a)) → (∀ a ∈ Mfr l • (⊤ : Submodule (𝒪 E) Dw.La), levPt Xo y.x a = Φ.e ((l.M : ℂ)⁻¹ • mk a)) → ∀ nab : NabV Xo c, IsNab Xo c nab → ∀ θ, Φ.toAnFr.Rd 1 y.s θ → ∀ (n : ℕ) (x x2 : Fin n → Dw.La), (∀ b, x2 b ∈ (Ideal.span {(2 : 𝒪 E)} ∗ (l.Mb.map (H.bar : 𝒪 E →+* 𝒪 E)) ^ 2) • (⊤ : Submodule (𝒪 E) Dw.La)) → (∀ b, x2 b − x b ∈ l.Mb • (⊤ : Submodule (𝒪 E) Dw.La)) → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ v : ℂ, ThV

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueExistsUniqueOfChartIndependence

private theorem **thExpr_tmul_lemma** (hL : Laws) (hC : ChartIndepSt2) {E : Type} [Field E] [NumberField E] [IsCMField E] {H : Host E} {Dw : Datum H} {l : Level H} {Xo : Moduli Dw l} {c : Coeff l} [StarRing (𝒪 E)] {nab : NabV Xo c} (hnab : IsNab Xo c nab) {S : Type} [CommRing S] [Algebra c.F S] (y : FrPt Xo c S) {a : (E →+* ℂ) × ℕ → ℕ} {n : ℕ} (x : Fin n → Dw.La) {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ harmA c a n) {S1 : Type} [CommRing S1] [Algebra c.F S1] [Algebra S S1] (φ1 : S →ₐ[c.F] S1) (hφ1 : ∀ s, φ1 s = algebraMap S S1 s) (y1 : FrPt Xo c S1) (F1 : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((l.M : ℤ) ^ 2) (ptO Xo y1.x))) (h1 : FrMap φ1 y y1) (hF1 : ∀ b, IsVtOver Xo.univ.t.G Xo.univ.t.PP l.M (KilledBy Xo (l.Mb ^ 2)) y1.s (levPt Xo y1.x (x b)) (F1 b)) {S2 : Type} [CommRing S2] [Algebra c.F S2] [Algebra S S2] (φ2 : S →ₐ[c.F] S2) (hφ2 : ∀ s, φ2 s = algebraMap S S2 s) (y2 : FrPt X

theorem **thValTot_of_chartIndep** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ThValTot Xo c nab

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueNaturalAlgebraMap

theorem **thetaValue_natural** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y0 : FrPt Xo c S) (y : FrPt Xo c T), FrMap φ y0 y → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ v0 w, ThValAt Xo c nab y0 x P v0 → ThValAt Xo c nab y x P w → φ v0 = w

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueNaturalAlgebraMapStatement

theorem **_root_.GlobalTheory.T.thetaValue_natural_algebraMap** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S T : Type) [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) (y0 : FrPt Xo c S) (y : FrPt Xo c T), FrMap φ y0 y → ∀ (a : (E →+* ℂ) × ℕ → ℕ) (n : ℕ) (x : Fin n → Dw.La) (P : MvPolynomial (LtN Dw n) c.F), P ∈ harmA c a n → ∀ v0 w, ThValAt Xo c nab y0 x P v0 → ThValAt Xo c nab y x P w → φ v0 = w

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueWeightTriangularFrame

theorem **appLE_id** {Y : Scheme.{0}} (U : Y.Opens) : Scheme.Hom.appLE (𝟙 Y) U U le_rfl = 𝟙 _

theorem **unit_mul_finsum** {R J : Type} [CommRing R] {r : R} (hr : IsUnit r) (f : J → R) : r ∗ ∑ᶠ j, f j = ∑ᶠ j, r ∗ f j

theorem **res_liftFun** {T : Over X} {W W' : (T ⊗ A).left.Opens} (h : W' ≤ W) (r : Γ(T.left, ⊤)) : ((T ⊗ A).left.presheaf.map (homOfLE h).op).hom (liftFun A W r) = liftFun A W' r

theorem **pull_liftFun** {T : Over X} (ψ : T ⊗ A ⟶ T ⊗ A) (hψ : ψ ≫ fst T A = fst T A) {W W' : (T ⊗ A).left.Opens} (h : W' ≤ ψ.left ⁻¹ᵁ W) (r : Γ(T.left, ⊤)) : ((T ⊗ A).left.presheaf.map (homOfLE h).op).hom ((ψ.left.app W).hom (liftFun A W r)) = liftFun A W' r

private lemma **twist_lemma** {R₁ R₂ R₃ R₄ : Type} [CommRing R₁] [CommRing R₂] [CommRing R₃] [CommRing R₄] (ρ : R₁ →+* R₄) (ψa : R₂ →+* R₃) (ρ₂ : R₃ →+* R₄) (x x' : R₁) (z z' : R₂) (c : R₄) (hx : ρ x = ρ₂ (ψa z)) (hx' : ρ x' = c ∗ ρ₂ (ψa z')) : ρ (x ∗ x') = c ∗ ρ₂ (ψa (z ∗ z'))

theorem **isTwistPull_scale** {L : LB A.left} {T : Over X} {ψ : T ⊗ A ⟶ T ⊗ A} {m : T ⊗ A ⟶ A} {hm : ψ ≫ snd T A = m} (hψ : ψ ≫ fst T A = fst T A) {a b : ℤ} {α : LB.Gen (LB.LinComb L (K

theorem **isVtOver_scale** {N : ℕ} {T : Over X} {ok : (T ⟶ A) → Prop} {s s₀ : LB.Sec (LPow A L.L 1 T)} {t : T ⟶ A} {F : LB.Sec (LPow A L.L ((N : ℤ) ^ 2) T)} (r : Γ(T.left, ⊤)) (hs : ∀ j, s₀.f j = liftFun A _ r ∗ s.f j) (h : IsVtOver A L N ok s t F) : IsVtOver A L N ok s₀ t (scaleSec A r F)

theorem **isUUOver_scale** {N : ℕ} {T : Over X} {s s₀ : LB.Sec (LPow A L.L 1 T)} {x₁ x₂ : T ⟶ A} {F : LB.Sec (LPow A L.L ((N : ℤ) ^ 2) T)} (r : Γ(T.left, ⊤)) (hs : ∀ j, s₀.f j = liftFun A _ r ∗ s.f j) (h : IsUUOver A L N s x₁ x₂ F) : IsUUOver A L N s₀ x₁ x₂ (scaleSec A r F)

theorem **isPullSec_id** {L : LB A.left} {k : ℤ} {T : Over X} (s : LB.Sec (LPow A L k T)) : IsPullSec A L k (𝟙 T ▷ A) (whiskerRight_snd _ _) s s

theorem **exists_reframe** {S : Type} [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (h : y'.x = y.x) : ∃ d : Reframe y, y' = d.pt

theorem **gen_eq** {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} (d : Reframe y) {t : Ix Dw → Sˣ} (hb : FrBor t y d.pt) (i : Fin 1 → Xo.univ.t.PP.L.ι) : d.s.f i = liftFun Xo.univ.t.G _ (T

theorem **map_wchar** {J : Type} [Fintype J] {K S T : Type} [Field K] [CommRing S] [Algebra K S] [CommRing T] [Algebra K T] (φ : S →ₐ[K] T) (e : J → ℕ) (t : J → Sˣ) : φ (wchar e t) = wchar e (fun i => Units.map (φ : S →* T) (t i))

theorem **isUnit_wchar** {J : Type} [Fintype J] {S : Type} [CommRing S] (e : J → ℕ) (t : J → Sˣ) : IsUnit (wchar e t)

theorem **sum_eq** (hC : ChartIndepSt2) [StarRing (𝒪 E)] {nab : NabV Xo c} (hn : IsNab Xo c nab) {S : Type} [CommRing S] [Algebra c.F S] {y : FrPt Xo c S} (d : Reframe y) {t : Ix Dw → Sˣ} (hb : FrBor t y d.pt) {n : ℕ} {a : (E →+* ℂ) × ℕ → ℕ} {P : MvPolynomial (LtN Dw n) c.F} (hP : P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i))) (N : ℕ) (F : Fin n → LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (ptO Xo y.x))) : ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) ∗ ∏ b, nab d.pt ((N : ℤ) ^ 2) (scaleSec Xo.univ.t.G (toΓ S (t none : S)) (F b)) (colOf m b) = wchar (ex Dw a n) t ∗ ∑ m ∈ P.support, algebraMap c.F S (P.coeff m) ∗ ∏ b, nab y ((N : ℤ) ^ 2) (F b) (colOf m b)

private lemma **dualMap_comp_epsMul_cusp_form_evaluations** {S T : Type} [CommRing S] [CommRing T] (f : S →+* T) (a : S) : (dualMap f).comp (epsMul a) = (epsMul (f a)).comp (dualMap f)

private lemma **dualMap_comp_algebraMap_cusp_form_evaluations** {S T : Type} [CommRing S] [CommRing T] (f : S →+* T) : (dualMap f).comp (algebraMap S (DualNumber S)) = (algebraMap T (DualNumber T)).comp f

private lemma **appLE_appLE_eq_of_comp_eq** {X Y Z₁ Z₂ : Scheme.{0}} (p : X ⟶ Y) (f₁ : Y ⟶ Z₁) (f₂ : X ⟶ Z₂) (g : Z₂ ⟶ Z₁) (e : p ≫ f₁ = f₂ ≫ g) (W : Y.Opens) (W' : X.Opens) (h : W' ≤ p ⁻¹ᵁ W) (h₁ : W ≤ f₁ ⁻¹ᵁ ⊤) (h₂ : W' ≤ f₂ ⁻¹ᵁ ⊤) (r : Γ(Z₁, ⊤)) : (p.appLE W W' h).hom ((f₁.appLE ⊤ W h₁).hom r) = (f₂.appLE ⊤ W' h₂).hom (g.appTop.hom r)

private theorem **frBor_units_map_of_frMap** {S T : Type} [CommRing S] [Algebra c.F S] [CommRing T] [Algebra c.F T] (φ : S →ₐ[c.F] T) {t : Ix Dw → Sˣ} {y : FrPt Xo c S} {z : FrPt Xo c T} (d₀ : Reframe y) (d : Reframe z) (hb : FrBor t y d₀.pt) (hz : FrMap φ y z) (hz' : FrMap φ d₀.pt d.pt) : FrBor (fun i => Units.map (φ : S →* T) (t i)) z d.pt

theorem **thetaValue_weight** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (t : Ix Dw → Sˣ), FrBor t y y' → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), (P ∈ harmA c a n → ∀ (x : Fin n → Dw.La) (v : S), ThValAt Xo c nab y x P v → ThValAt Xo c nab y' x P (wchar (ex Dw a n) t ∗ v)) ∧ (P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (β : c.F) (N : ℕ) (v : S), ShortValAt Xo c nab w Λ β N P y v → ShortValAt Xo c nab w Λ β N P y' (wchar (ex Dw a n) t ∗ v))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ThetaValueWeightTriangularFrameStatement

theorem **_root_.GlobalTheory.T.thetaValue_weight_triangular_frame** : Laws → ChartIndepSt2 → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (c : Coeff l) [StarRing (𝒪 E)] (nab : NabV Xo c), IsNab Xo c nab → ∀ (S : Type) [CommRing S] [Algebra c.F S] (y y' : FrPt Xo c S) (t : Ix Dw → Sˣ), FrBor t y y' → ∀ (n : ℕ) (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (LtN Dw n) c.F), (P ∈ harmA c a n → ∀ (x : Fin n → Dw.La) (v : S), ThValAt Xo c nab y x P v → ThValAt Xo c nab y' x P (wchar (ex Dw a n) t ∗ v)) ∧ (P ∈ HHa Dw.mult n c.F ComplexEmbedding.conjugate (InPsi H) (fun τ i => a (τ, i)) → ∀ (w : (Fin n → LevQ Dw l) → c.F) (Λ : GenLat E n) (β : c.F) (N : ℕ) (v : S), ShortValAt Xo c nab w Λ β N P y v → ShortValAt Xo c nab w Λ β N P y' (wchar (ex Dw a n) t ∗ v))

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.TorsionIsotropicCommutatorPairing

theorem **ev_sum** (I : TateIso Dw k Q act e) (m : ℕ) (hm : 0 < m) {n : ℕ} (a : Fin n → Dw.La) : I.ev m (∑ j, a j) = ∏ j, I.ev m (a j)

theorem **dvd_EF_of_killed** (m j : ℕ) (a b : Dw.La) (ha : ∀ c ∈ l.Mb ^ j, Dw.ModN m (c • a)) (hb : ∀ c ∈ l.Mb ^ j, Dw.ModN m (c • b)) : (m : ℤ) ∣ Dw.EF a b

theorem **unit_eq_one** (hL : Laws) (N : ℕ) (hN : 0 < N) (u : Γ((ptO Xo x0).left, ⊤)ˣ) (hu : u ^ (N ^ 2) = 1) (h : ∀ (K : Type) [Field K] [IsAlgClosed K] (χ : B →+* K), Units.map (bc Xo x0 χ).left.appTop.hom.toMonoidHom u = 1) : u = 1

theorem **exists_tate** (hL : Laws) (N : ℕ) (hN : 0 < N) (K : Type) [Field K] [IsAlgClosed K] (χ : B →+* K) : ∃ I : TateIso Dw K (ptO Xo (spMap χ ≫ x0) ⟶ Xo.univ.t.G) (fun a g => g ≫ Xo.univ.t.act.act a) (fun n g g' c => WeilIs Xo.univ.t.G Xo.univ.t.PP.L n g g' (unitOn c)), ∀ (x y : ptO Xo x0 ⟶ Xo.univ.t.G) (e : Γ((ptO Xo x0).left, ⊤)ˣ), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e → ∀ a b, I.ev (N ^ 2) a = bc Xo x0 χ ≫ x → I.ev (N ^ 2) b = bc Xo x0 χ ≫ y → Units.map (bc Xo x0 χ).left.appTop.hom.toMonoidHom e = unitOn (I.ζ (N ^ 2) ^ Dw.EF a b)

theorem **unitOn_prod** {K : Type} [CommRing K] {n : ℕ} (c : Fin n → Kˣ) : ∏ b, unitOn (c b) = unitOn (∏ b, c b)

theorem **prod_zpow_eq** {U : Type} [CommGroup U] (z : U) {n : ℕ} (c : Fin n → ℤ) : ∏ b, z ^ c b = z ^ ∑ b, c b

theorem **unitOn_one** {K : Type} [CommRing K] : unitOn (1 : Kˣ) = 1

theorem **comp_prod** {S : Scheme.{0}} {G T T' : Over S} [GrpObj G] [IsCommMonObj G] (h : T' ⟶ T) {n : ℕ} (x : Fin n → (T ⟶ G)) : h ≫ (List.ofFn x).prod = ∏ j, h ≫ x j

theorem **zh_isotropic_and_half_isotropic** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (B : Type) [CommRing B] (x0 : Sp B ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → (∀ (a a' : Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G)) (e : Fin n → Γ((ptO Xo x0).left, ⊤)ˣ), InZh Xo Λ N a → InZh Xo Λ N a' → (∀ b, WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (a b) (a' b) (e b)) → ∏ b, e b = 1) ∧ ∀ (x y : ptO Xo x0 ⟶ Xo.univ.t.G) (j : ℕ), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → KilledBy Xo (l.Mb ^ j) x → KilledBy Xo (l.Mb ^ j) y → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.TorsionIsotropicCommutatorPairingStatement

theorem **_root_.GlobalTheory.T.torsion_isotropic_commutator_pairing** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (B : Type) [CommRing B] (x0 : Sp B ⟶ Xo.X) (n : ℕ) (Λ : GenLat E n) (N : ℕ), Λ.Unimod → 0 < N → Ideal.span {(N : 𝒪 E)} • stdLat E n ≤ Λ.La → (∀ (a a' : Fin n → (ptO Xo x0 ⟶ Xo.univ.t.G)) (e : Fin n → Γ((ptO Xo x0).left, ⊤)ˣ), InZh Xo Λ N a → InZh Xo Λ N a' → (∀ b, WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (a b) (a' b) (e b)) → ∏ b, e b = 1) ∧ ∀ (x y : ptO Xo x0 ⟶ Xo.univ.t.G) (j : ℕ), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → KilledBy Xo (l.Mb ^ j) x → KilledBy Xo (l.Mb ^ j) y → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.TorusAtEpsIndep

theorem **cc_cc** (a : E) : cc (cc a) = a

theorem **cc_mul** (a b : E) : cc (a ∗ b) = cc a ∗ cc b

theorem **cc_sub** (a b : E) : cc (a − b) = cc a − cc b

theorem **trace_conj** (a : E) : Algebra.trace ℚ E (cc a) = Algebra.trace ℚ E a

theorem **herm_smul_left** (G : Matrix (Fin r) (Fin r) E) (a : E) (u v : Fin r → E) : herm G (a • u) v = a ∗ herm G u v

theorem **herm_smul_right** (G : Matrix (Fin r) (Fin r) E) (a : E) (u v : Fin r → E) : herm G u (a • v) = cc a ∗ herm G u v

theorem **herm_add_left** (G : Matrix (Fin r) (Fin r) E) (u u' v : Fin r → E) : herm G (u + u') v = herm G u v + herm G u' v

theorem **herm_add_right** (G : Matrix (Fin r) (Fin r) E) (u v v' : Fin r → E) : herm G u (v + v') = herm G u v + herm G u v'

theorem **cc_herm** {G : Matrix (Fin r) (Fin r) E} (hG : ∀ a b, cc (G a b) = G b a) (u v : Fin r → E) : cc (herm G u v) = herm G v u

theorem **B_skew** (X : Data E r) (u v : Fin r → E) : B X u v = −B X v u

theorem **B_adj** (X : Data E r) (a : E) (u v : Fin r → E) : B X (a • u) v = B X u (cc a • v)

theorem **B_add_left** (X : Data E r) (u u' v : Fin r → E) : B X (u + u') v = B X u v + B X u' v

theorem **B_add_right** (X : Data E r) (u v v' : Fin r → E) : B X u (v + v') = B X u v + B X u v'

theorem **B_sub_smul** (X : Data E r) (a b : E) (u v : Fin r → E) : B X u ((a − b) • v) = B X u (a • v) − B X u (b • v)

theorem **B_swap** (X : Data E r) (a : E) (u v : Fin r → E) : B X v (a • u) = −B X u (cc a • v)

theorem **EFf_eq** (X : Data E r) (u v : Fin r → E) : EFf X.ξ0 X.Gm u v = −B X u v

theorem **B_int** (X : DataX E r) {u v : Fin r → E} (hu : u ∈ X.Λ) (hv : v ∈ X.Λ) : ∃ k : ℤ, B X.toData u v = k

theorem **exists_half** (X : DataX E r) (y : Fin r → E) (hy : ∀ u ∈ X.Λ, ∃ k : ℤ, B X.toData u y = 2 ∗ k) : ∃ x ∈ X.Λ, ∀ u ∈ X.Λ, B X.toData u y = 2 ∗ B X.toData u x

theorem **smul_mem** (X : Data E r) (w : 𝒪 E) {v : Fin r → E} (hv : v ∈ X.Λ) : (w : E) • v ∈ X.Λ

theorem **cc_smul_mem** (X : Data E r) (w : 𝒪 E) {v : Fin r → E} (hv : v ∈ X.Λ) : cc (w : E) • v ∈ X.Λ

theorem **parity_eq** (X : DataX E r) {w w' : 𝒪 E} (hw : Par X.toData w) (hw' : Par X.toData w') {v : Fin r → E} (hv : v ∈ X.Λ) : ∃ k : ℤ, B X.toData v ((w : E) • v) − B X.toData v ((w' : E) • v) = 2 * k

theorem **cos_sign** {s : ℝ} (hs : s = 1 ∨ s = -1) (k : ℤ) : Real.cos (Real.pi * (s * (((k : ℚ)) : ℝ))) = ((k.negOnePow : ℤ) : ℝ)

theorem **eps_val** {X : Data E r} {x : Frame X.Ψ r} (A : TorusAt X x) {v : Fin r → E} (hv : v ∈ X.Λ) {k : ℤ} (hk : B X v ((A.w0 : E) • v) = k) : A.T.eps (A.φ v) = (k.negOnePow : ℤ)

theorem **par_of_torusAt** (X : DataX E r) {x : Frame X.Ψ r} (A : TorusAt X.toData x) : Par X.toData A.w0

theorem **torusAt_eps_indep** : ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (r : ℕ) (X : DataX E r), (∃ w0 : 𝒪 E, ∀ u ∈ X.Λ, ∀ v ∈ X.Λ, ∃ k : ℤ, Algebra.trace ℚ E (X.ξ0 * herm X.Gm u (((w0 : E) − cc (w0 : E)) • v)) − Algebra.trace ℚ E (X.ξ0 * herm X.Gm u v) = 2 * k) → ∀ (x x' : Frame X.Ψ r) (A : TorusAt X.toData x) (A' : TorusAt X.toData x') (γ : GL (Fin r) E), IsUStab X.Gm X.Λ γ → ∀ v ∈ X.Λ, A'.T.eps (A'.φ ((γ : Matrix (Fin r) (Fin r) E).mulVec v)) = A.T.eps (A.φ v)

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.TraceHermNondegenerate

theorem **exists_euclidean** {M : ℕ} [FiniteDimensional ℚ V] (hM : finrank ℚ V = M) (B : LinearMap.BilinForm ℚ V) (hs : B.IsSymm) (hp : ∀ x, 0 ≤ B x x) : ∃ φ : V →ₗ[ℚ] EuclideanSpace ℝ (Fin M), ∀ x y, inner ℝ (φ x) (φ y) = (B x y : ℝ)

theorem **transference_rat** {M : ℕ} (b : Basis (Fin M) ℚ V) (B : LinearMap.BilinForm ℚ V) (hs : B.IsSymm) (hp : ∀ x, 0 ≤ B x x) (hn : B.Nondegenerate) (lam : ℝ) (hlam : 0 < lam) (hmin : ∀ u, u ≠ 0 → (∀ j, ∃ k : ℤ, B u (b j) = k) → lam ^ 2 ≤ (B u u : ℝ)) : ∃ v : Fin M → V, (∀ i, v i ∈ span ℤ (Set.range b)) ∧ LinearIndependent ℚ v ∧ ∀ i, (B (v i) (v i) : ℝ) ≤ ((M : ℝ) ^ 2 / lam) ^ 2

theorem **traceHerm_apply** {n : ℕ} (x y : Fin n → E) : traceHerm n x y = Algebra.trace ℚ E (herm 1 x y)

theorem **traceHerm_self** {n : ℕ} (x : Fin n → E) : (traceHerm n x x : ℝ) = ∑ σ : E →+* ℂ, ∑ k, ‖σ (x k)‖ ^ 2

theorem **traceHerm_isSymm** (n : ℕ) : (traceHerm (E

theorem **traceHerm_nonneg** {n : ℕ} (x : Fin n → E) : 0 ≤ traceHerm n x x

theorem **two_mul_le_traceHerm** {n : ℕ} (x : Fin n → E) (τ : E →+* ℂ) : 2 * ∑ k, ‖τ (x k)‖ ^ 2 ≤ (traceHerm n x x : ℝ)

theorem **traceHerm_nondegenerate** (n : ℕ) : (traceHerm (E

theorem **span_rat_eq_top** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (h : span E (Λ : Set (Fin n → E)) = ⊤) : span ℚ (Λ : Set (Fin n → E)) = ⊤

theorem **finrank_eq** : finrank ℚ E = 2 * fdeg E

theorem **exists_basis** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ.FG) (h : span E (Λ : Set (Fin n → E)) = ⊤) : ∃ b : Basis (Fin (2 * fdeg E * n)) ℚ (Fin n → E), ∀ x, x ∈ span ℤ (Set.range b) ↔ x ∈ Λ

theorem **exists_short_vectors** {n : ℕ} (Λ : Submodule (𝒪 E) (Fin n → E)) (hfg : Λ.FG) (hint : ∀ x ∈ Λ, ∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) (hdual : ∀ x, (∀ y ∈ Λ, IsIntegral ℤ (herm 1 x y)) → x ∈ Λ) : ∃ v : Fin (2 * fdeg E * n) → Fin n → E, (∀ i, v i ∈ Λ) ∧ LinearIndependent ℚ v ∧ Submodule.span ℚ (Set.range v) = ⊤ ∧ ∀ i (τ : E →+* ℂ), ∑ k, ‖τ (v i k)‖ ^ 2 ≤ (2 * (fdeg E : ℝ) ^ ((3 : ℝ) / 2) * n ^ 2 * rdE E) ^ 2

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.TripleProductTorusPeriodicBounded

theorem **exists_le_of_periodic** {S : Set E} {R : ℝ} (hS : ∀ z, ∃ l ∈ S, ‖z + l‖ ≤ R) (φ : E → ℝ) (hφ : Continuous φ) (hp : ∀ l ∈ S, ∀ z, φ (z + l) = φ z) : ∃ C, ∀ z, φ z ≤ C

theorem **weight_nonneg_of_periodic** {S : Set E} {R : ℝ} (hS : ∀ z, ∃ l ∈ S, ‖z + l‖ ≤ R) (r : E → ℂ) (hr : Differentiable ℂ r) (h0 : r 0 ≠ 0) (q : E → ℝ) (hq : Continuous q) (hs : ∀ (t : ℂ) v, q (t • v) = ‖t‖ ^ 2 * q v) (hp : ∀ l ∈ S, ∀ z, ‖r (z + l)‖ * Real.exp (−q (z + l)) = ‖r z‖ * Real.exp (−q z)) (v : E) : 0 ≤ q v

theorem **form_eq_zero_and_const** {S : Set E} {R : ℝ} (hS : ∀ z, ∃ l ∈ S, ‖z + l‖ ≤ R) (ψ : E → ℂ) (hψ : Differentiable ℂ ψ) (h0 : ∀ z, ψ z ≠ 0) (D : E → E → ℂ) (hc : Continuous fun z => D z z) (hadd : ∀ z z' w, D (z + z') w = D z w + D z' w) (hsm : ∀ (t : ℂ) z w, D (t • z) w = t * D z w) (hst : ∀ z w, D w z = star (D z w)) (hper : ∀ l ∈ S, ∃ ρ : ℂ, ‖ρ‖ = 1 ∧ ∀ z, ψ (z + l) = ρ * cexp (π * D z l + π / 2 * D l l) * ψ z) : (∀ z w, D z w = 0) ∧ ∀ z, ψ z = ψ 0

theorem **differentiableOn_fun_prod** {α : Type*} {f : α → E → ℂ} {S : Set E} (s : Finset α) (hf : ∀ i, DifferentiableOn ℂ (f i) S) : DifferentiableOn ℂ (fun z => ∏ i ∈ s, f i z) S

theorem **exists_period_norm_le** : ∃ R : ℝ, ∀ z : Fin g → ℂ, ∃ l ∈ u.Λ, ‖z + l‖ ≤ R

theorem **gen_linComb_in_frame** (hL : L.IsLB) {Hm : Matrix (Fin g) (Fin g) ℂ} {χ : u.Λ → ℂ} (hu : u.IsAH Hm χ) {E : Type} [NormedAddCommGroup E] [NormedSpace ℂ E] [FiniteDimensional ℂ E] {Y : Scheme.{0}} (p : E → (Sp ℂ ⟶ Y)) (hp : ∀ (U : Y.Opens) (f : Γ(Y, U)), IsOpen {z | ⊤ ≤ p z ⁻¹ᵁ U} ∧ DifferentiableOn ℂ (fun z => valAt' (p z) U f) {z | ⊤ ≤ p z ⁻¹ᵁ U}) (hS : ∃ R : ℝ, ∀ z : E, ∃ l, (∀ w, p (w + l) = p w) ∧ ‖z + l‖ ≤ R) {K : Type} [Fintype K] (φ : K → (Y ⟶ A.left)) (ε : K → ℤ) (lam : K → E →ₗ[ℂ] (Fin g → ℂ)) (hlam : ∀ k z, p z ≫ φ k = (u.ex (lam k z)).left) (t : LB.Gen (LB.LinComb L φ ε)) (ht : t.IsGen) : (∀ z w, ∑ k, (ε k : ℂ) * hermForm Hm (lam k z) (lam k w) = 0) ∧ ∃ κ : ℂ, κ ≠ 0 ∧ ∀ (j : K → L.ι) (z : E), (∀ k, ⊤ ≤ (u.ex (lam k z)).left ⁻¹ᵁ L.U (j k)) → valAt' (p z) (L.wchart φ j) (t.s j).val = κ * ∏ k, u.fr (j k) (lam k z) ^ ε k

theorem **val_eq_one_of_corr** {X Y Z : Scheme.{0}} (L : LB X) {K : Type} [Fintype K] (φ : K → (Y ⟶ X)) (ε : K → ℤ) (t : LB.Gen (LB.LinComb L φ ε)) (h : Z ⟶ Y) (hc : LB.Gen.Corr L _ _ _ _ (LB.pullGen L φ ε t h) (LB.oneGen L)) (x : Sp ℂ ⟶ Z) (j : L.ι) (hx : ∀ k, ⊤ ≤ ((x ≫ h) ≫ φ k) ⁻¹ᵁ L.U j) : valAt' (x ≫ h) (L.wchart φ fun _ => j) (t.s fun _ => j).val = 1

theorem **prod_zpow_eq_zpow_sum** {K : Type} (s : Finset K) {a : ℂ} (ha : a ≠ 0) (ε : K → ℤ) : ∏ k ∈ s, a ^ ε k = a ^ ∑ k ∈ s, ε k

theorem **primLin_apply** (W : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)) (k : Fin 5) (x : Fin g → ℂ) : primLin W k x = ![x, x + W x, x, W x, 0] k

theorem **ex_comp_primφ** {w : A ⟶ A} {W : (Fin g → ℂ) →ₗ[ℂ] (Fin g → ℂ)} (hW : ∀ z, u.ex (W z) = u.ex z ≫ w) (k : Fin 5) (z : Fin g → ℂ) : (u.ex z).left ≫ primφ w k = (u.ex (primLin W k z)).left

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UnitaryDatumNumbersForSupBounds

theorem **finrank_host** (D : D7) : Module.finrank ℚ D.host.E = 2 * D.R.f

theorem **xiR_pos** (D : D7) : 0 < xiR D

theorem **one_le_dE** (E : Type) [Field E] [NumberField E] : 1 ≤ dE E

private lemma **sum_ext0_le_sum** {l : ℕ} (a : Fin l → ℕ) (M : ℕ) : ∑ i : Fin M, ext0 a (i : ℕ) ≤ ∑ i, a i

theorem **record_dictionary** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), fdeg D.host.E = D.R.f ∧ Real.log (rdE D.host.E) = lrd D ∧ 1 ≤ rdE D.host.E ∧ 1 ≤ D.R.f ∧ 1 ≤ D.R.r ∧ 0 < xiR D ∧ Formalism.L2.Pj.DF (T.Q.Dw j) (wtA w) ≤ WeightAlgebra.DF w ∧ absA w = (WeightAlgebra.DF w : ℝ) ∧ (∀ σ, GoodEmb T.Q.c σ → ∀ k τ, ((T.U j σ).X k).c τ = 2 * xiR D) ∧ lieFac

(T.Q.Dw j) (xiR D) (wtA w) = Real.exp (-(((Formalism.L2.Pj.DF (T.Q.Dw j) (wtA w) : ℕ) : ℝ) / 2) * Real.log (Real.pi * (2 * xiR D))) ∧ T.Q.l.A = 1 ∧ 0 < D.host.pde ∧ 0 < D.host.pde' ∧ ∀ (ι : Type) (σ : T.Q.c.F →+* ℂ) (P : MvPolynomial ι T.Q.c.F), l1At (goodOf T.Q.c σ) P = l1At σ P

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UnitaryDatumNumbersForSupBoundsStatement

theorem **_root_.GlobalTheory.T.unitaryDatumNumbers_for_supBounds** : ∀ (D : D7) (T : Top D) (_h : LawsP T) (j : ℕ) (_hj : InR D j) (w : WeightFamilies.Wt D j), fdeg D.host.E = D.R.f ∧ Real.log (rdE D.host.E) = lrd D ∧ 1 ≤ rdE D.host.E ∧ 1 ≤ D.R.f ∧ 1 ≤ D.R.r ∧ 0 < xiR D ∧ Formalism.L2.Pj.DF (T.Q.Dw j) (wtA w) ≤ WeightAlgebra.DF w ∧ absA w = (WeightAlgebra.DF w : ℝ) ∧ (∀ σ, GoodEmb T.Q.c σ → ∀ k τ, ((T.U j σ).X k).c τ = 2 * xiR D) ∧ lieFac (T.Q.Dw j) (xiR D) (wtA w) = Real.exp (-(((Formalism.L2.Pj.DF (T.Q.Dw j) (wtA w) : ℕ) : ℝ) / 2) * Real.log (Real.pi * (2 * xiR D))) ∧ T.Q.l.A = 1 ∧ 0 < D.host.pde ∧ 0 < D.host.pde' ∧ ∀ (ι : Type) (σ : T.Q.c.F →+* ℂ) (P : MvPolynomial ι T.Q.c.F), l1At (goodOf T.Q.c σ) P = l1At σ P

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalNormalisedMulByNatPullbackIso

theorem **rigPow_isGen** {ρ : LB.Gen (L.pull (eSec A))} (hρ : ρ.IsGen) (k : ℤ) : (rigPow A L ρ k).IsGen

theorem **eSec_hom** : eSec A ≫ A.hom = 𝟙 S

theorem **eSec_mulA** (n : ℕ) : eSec A ≫ (mulA A n).left = eSec A

theorem **exists_normalised** (hL : L.IsLB) {ρ : LB.Gen (L.pull (eSec A))} (hρ : ρ.IsGen) (n : ℕ) (β : LB.Gen (LB.LinComb L (K

theorem **normalised_unique** (hL : L.IsLB) (hp : Function.Surjective A.hom.appTop.hom) {ρ : LB.Gen (L.pull (eSec A))} {n : ℕ} {α β : LB.Gen (LB.LinComb L (K

theorem **exists_pull** (hL : L.IsLB) {n : ℕ} (β : LB.Gen (LB.LinComb L (K

theorem **moduli_univ_normMul** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), ∃ α₀ : LB.Gen (LB.LinComb Xo.univ.t.PP.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalNormalisedMulByNatPullbackIsoStatement

theorem **_root_.GlobalTheory.T.universal_normalised_mulByNat_pullback_iso** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), ∃ α₀ : LB.Gen (LB.LinComb Xo.univ.t.PP.L (K

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalSymLiftExistsUniqueGlobal

theorem **symLift_exists_unique_of_cast** (hLaw : Laws) (S : Scheme.{0}) [IsLocallyNoetherian S] (P : PolAV S) (k : ℕ) (hk : Odd k) (hu : IsUnit (k : Γ(S, ⊤))) (K : ℤ) (hK : K = (k : ℤ)) (T : Over S) (x : T ⟶ P.A) (hx : x ^ k = 1) : (∃ U : ThetaGrp P.A P.L K T, SymLift P.A P.L K k x U) ∧ ∀ U U' : ThetaGrp P.A P.L K T, SymLift P.A P.L K k x U → SymLift P.A P.L K k x U' → LB.Gen.Corr P.L _ _ _ _ U.φ U'.φ

theorem **moduli_univ_symLift_global** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), Odd N → ∀ (T : Over Xo.X) (x : T ⟶ Xo.univ.t.G), x ^ (N ^ 2) = 1 → (∃ U : ThetaGrp Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T, SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) x U) ∧ ∀ U U' : ThetaGrp Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T, SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) x U → SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) x U' → LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ U.φ U'.φ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalSymLiftExistsUniqueGlobalStatement

theorem **_root_.GlobalTheory.T.universal_symLift_exists_unique_global** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), Odd N → ∀ (T : Over Xo.X) (x : T ⟶ Xo.univ.t.G), x ^ (N ^ 2) = 1 → (∃ U : ThetaGrp Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T, SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) x U) ∧ ∀ U U' : ThetaGrp Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T, SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) x U → SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) x U' → LB.Gen.Corr Xo.univ.t.PP.L _ _ _ _ U.φ U'.φ

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalSymLiftMulOdd

theorem **sq_factor** (hm : ∀ y, fa y * fb (a * y) = ρ * fc y) (hc : ∀ y, fa y * fb (a * y) = fb y * fa (b * y) * ε) (sa : ∀ y, fa (a⁻¹ * y) * ν y * (fa y⁻¹ * (ν (a⁻¹ * y))⁻¹) = 1) (sb : ∀ y, fb (b⁻¹ * y) * ν y * (fb y⁻¹ * (ν (b⁻¹ * y))⁻¹) = 1) (sc : ∀ y, fc ((a * b)⁻¹ * y) * ν y * (fc y⁻¹ * (ν ((a * b)⁻¹ * y))⁻¹) = 1) : ρ ^ 2 = ε

theorem **comm_pow** (hc : ∀ y, fa y * fb (a * y) = fb y * fa (b * y) * ε) (m : ℕ) (y : G) : fa y * ∏ j ∈ Finset.range m, fb (b ^ j * (a * y)) = ε ^ m * (∏ j ∈ Finset.range m, fb (b ^ j * y)) * fa (b ^ m * y)

theorem **prod_pow** (hm : ∀ y, fa y * fb (a * y) = ρ * fc y) (hc : ∀ y, fa y * fb (a * y) = fb y * fa (b * y) * ρ ^ 2) (m : ℕ) (y : G) : ρ ^ (m * m) * ∏ j ∈ Finset.range m, fc ((a * b) ^ j * y) = (∏ j ∈ Finset.range m, fa (a ^ j * y)) * ∏ j ∈ Finset.range m, fb (b ^ j * (a ^ m * y))

theorem **pow_factor** {n : ℕ} (hn : Odd n) (hm : ∀ y, fa y * fb (a * y) = ρ * fc y) (hc : ∀ y, fa y * fb (a * y) = fb y * fa (b * y) * ρ ^ 2) (pa : ∀ y, ∏ j ∈ Finset.range n, fa (a ^ j * y) = 1) (pb : ∀ y, ∏ j ∈ Finset.range n, fb (b ^ j * y) = 1) (pc : ∀ y, ∏ j ∈ Finset.range n, fc ((a * b) ^ j * y) = 1) (hp : fa (b ^ n * 1) = fa 1) : ρ ^ n = 1

theorem **lift_translT** (σ : T' ⟶ T) (y : T' ⟶ A) (x : T ⟶ A) : (lift σ y).left ≫ (translT A x).left = (lift σ ((σ ≫ x) * y)).left

theorem **lift_negT** (σ : T' ⟶ T) (y : T' ⟶ A) : (lift σ y).left ≫ (negT A T).left = (lift σ y⁻¹).left

theorem **lift_fst_left** (σ : T' ⟶ T) (y : T' ⟶ A) : (lift σ y).left ≫ (fst T A).left = σ.left

private theorem **comp_pow_cusp_form_evaluations** {X Y : Over S} (f : X ⟶ Y) (x : Y ⟶ A) (m : ℕ) : f ≫ x ^ m = (f ≫ x) ^ m

theorem **pow_val** {θ : ThetaGrp A L k T} {n : ℕ} (h : LB.Gen.Corr L _ _ _ _ (powGen A L k θ n) (LB.oneGen L)) (σ : T' ⟶ T) {z : T'.left} {J} (hJ : Good L z J) (y : T' ⟶ A) : ∏ j ∈ Finset.range n, tval θ (lift σ ((σ ≫ θ.pt) ^ j * y)).left z J = 1

private theorem **exp_le_qw_div_le_exp_of_far_letters** {U : Type} [CommGroup U] (t1 t2 a a' b b' : U) (hv : t1 * b * (t2 * b'⁻¹) = 1) (hd : a * b⁻¹ = a' * b'⁻¹) : t1 * a * (t2 * a'⁻¹) = 1

private theorem **comp_inv_cusp_form_evaluations** {X Y : Over S} (f : X ⟶ Y) (x : Y ⟶ A) : f ≫ x⁻¹ = (f ≫ x)⁻¹

theorem **sym_val'** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {θ : ThetaGrp A L k T} {n : ℕ} {x : T ⟶ A} (h : SymLift A L k n x θ) {ν : LB.Gen (negLB A L k T)} (hν : ν.IsGen) (σ : T' ⟶ T) {z : T'.left} {J} (hJ : Good L z J) (y : T' ⟶ A) : tval θ (lift σ ((σ ≫ x)⁻¹ * y)).left z J * nval ν (lift σ y).left z J * (tval θ (lift σ y⁻¹).left z J * (nval ν (lift σ ((σ ≫ x)⁻¹ * y)).left z J)⁻¹) = 1

theorem **tval_trans** (θ : ThetaGrp A L k T) (σ : T' ⟶ T) {z : T'.left} {J J'} (hJ : Good L z J) (hJ' : Good L z J') (y : T' ⟶ A) : tval θ (lift σ y).left z J = gm y.left z (L.U (J y.left) ⊓ L.U (J' y.left)) (L.g _ _) ^ k * (gm ((σ ≫ θ.pt) * y).left z (L.U (J ((σ ≫ θ.pt)

∗ y).left) ⊓ L.U (J' ((σ ≫ θ.pt) ∗ y).left)) (L.g _ _) ^ k)⁻¹ ∗ tval θ (lift σ y).left z J'

private theorem **prod_range_mul_inv_succ** {U : Type} [CommGroup U] (f : ℕ → U) (n : ℕ) : ∏ i ∈ Finset.range n, f i ∗ (f (i + 1))⁻¹ = f 0 ∗ (f n)⁻¹

theorem **trans_trivial** (hL : L.IsLB) {θ : ThetaGrp A L k T} {n : ℕ} (h : LB.Gen.Corr L _ _ _ _ (powGen A L k θ n) (LB.oneGen L)) (hn : θ.pt ^ n ≠ 1) (y : T ⊗ A ⟶ A) (z : (T ⊗ A).left) {i i' : L.ι} (hi : y.left z ∈ L.U i) (hi' : y.left z ∈ L.U i') : gm y.left z (L.U i ⊓ L.U i') (L.g i i') ^ k = 1

theorem **tval_periodic** (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {θ : ThetaGrp A L k T} {n : ℕ} (h : LB.Gen.Corr L _ _ _ _ (powGen A L k θ n) (LB.oneGen L)) (ξ : ThetaGrp A L k T) (σ : T' ⟶ T) {z : T'.left} {J} (hJ : Good L z J) : tval ξ (lift σ ((σ ≫ θ.pt) ^ n ∗ 1)).left z J = tval ξ (lift σ 1).left z J

theorem **symLift_mul** [IsCommMonObj A] (hG : GlobalFnsSt) (hA : IsAbelianScheme A) (hL : L.IsLB) {n : ℕ} (hn : Odd n) {a b : T ⟶ A} {Ua Ub Uab : ThetaGrp A L k T} {e : Γ(T.left, ⊤)ˣ} (ha : SymLift A L k n a Ua) (hb : SymLift A L k n b Ub) (hab : SymLift A L k n (a ∗ b) Uab) (he : WeilIs A L k a b e) : ∃ r : Γ(T.left, ⊤)ˣ, r ^ 2 = e ∧ r ^ n = 1 ∧ ∀ s s₁ s₂ s₃ : LB.Sec (LPow A L k T), IsActOf A L k Ub s s₁ → IsActOf A L k Ua s₁ s₂ → IsActOf A L k Uab s s₃ → ∀ j, s₂.f j = liftFun A _ (r : Γ(T.left, ⊤)) ∗ s₃.f j

theorem **moduli_univ_symLift_mul** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), Odd N → ∀ (T : Over Xo.X) (a b : T ⟶ Xo.univ.t.G) (Ua Ub Uab : ThetaGrp Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T) (e : Γ(T.left, ⊤)ˣ), SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) a Ua → SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) b Ub → SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) (a ∗ b) Uab → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) a b e → ∃ r : Γ(T.left, ⊤)ˣ, r ^ 2 = e ∧ r ^ (N ^ 2) = 1 ∧ ∀ s s₁ s₂ s₃ : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T), IsActOf Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) Ub s s₁ → IsActOf Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) Ua s₁ s₂ → IsActOf Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) Uab s s₃ → ∀ j, s₂.f j = liftFun Xo.univ.t.G _ (r :

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalSymLiftMulOddStatement

theorem **_root_.GlobalTheory.T.universal_symLift_mul_odd** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), Odd N → ∀ (T : Over Xo.X) (a b : T ⟶ Xo.univ.t.G) (Ua Ub Uab : ThetaGrp Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T) (e : Γ(T.left, ⊤)ˣ), SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) a Ua → SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) b Ub → SymLift Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (N ^ 2) (a ∗ b) Uab → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) a b e → ∃ r : Γ(T.left, ⊤)ˣ, r ^ 2 = e ∧ r ^ (N ^ 2) = 1 ∧ ∀ s s₁ s₂ s₃ : LB.Sec (LPow Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) T), IsActOf Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) Ub s s₁ → IsActOf Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) Ua s₁ s₂ → IsActOf Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) Uab s s₃ → ∀ j, s₂.f j = lif

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalWeilPairingPullbackExistsUnique

theorem **weilIs_pow_eq_one** (hW : WeilSt) {S : Scheme.{0}} [IsLocallyNoetherian S] (P : PolAV S) (k : ℕ) {T : Over S} (x y : T ⟶ P.A) (hx : x ^ k = 1) (hy : y ^ k = 1) (c : Γ(T.left, ⊤)ˣ) (hc : WeilIs P.A P.L k x y c) : c ^ k = 1

theorem **isNilpotent_of_geometric** {B : Type} [CommRing B] {b : B} (h : ∀ (K : Type) [Field K] [IsAlgClosed K] (χ : B →+∗ K), χ b = 0) : IsNilpotent b

theorem **eq_one_of_geometric** (B : Type) [CommRing B] (u : B) (k : ℕ) (hk : IsUnit (k : B)) (hu : u ^ k = 1) (h : ∀ (K : Type) [Field K] [IsAlgClosed K] (χ : B →+∗ K), χ u = 1) : u = 1

theorem **weilIs_univ_pull_exists_unique** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), 0 < N → (∀ (T T' : Over Xo.X) (h : T' ⟶ T) (x y : T ⟶ Xo.univ.t.G) (e : Γ(T.left, ⊤)ˣ), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom e)) ∧ (∀ (T : Over Xo.X) (x y : T ⟶ Xo.univ.t.G), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → ∃ e : Γ(T.left, ⊤)ˣ, WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e ∧ e ^ (N ^ 2) = 1 ∧ ∀ e', WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e' → e' = e) ∧ ∀ (B : Type) [CommRing B] (u : B) (k : ℕ), IsUnit (k : B) → u ^ k = 1 → (∀ (K : Type) [Field K] [IsAlgClosed K] (χ : B →+∗ K), χ u = 1) → u = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.UniversalWeilPairingPullbackExistsUniqueStatement

theorem **_root_.GlobalTheory.T.universal_weilPairing_pullback_exists_unique** : Laws → ∀ (E : Type) [Field E] [NumberField E] [IsCMField E] (H : Host E) (Dw : Datum H) (l : Level H) (Xo : Moduli Dw l) (N : ℕ), 0 < N → (∀ (T T' : Over Xo.X) (h : T' ⟶ T) (x y : T ⟶ Xo.univ.t.G) (e : Γ(T.left, ⊤)ˣ), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e → WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) (h ≫ x) (h ≫ y) (Units.map h.left.appTop.hom.toMonoidHom e)) ∧ (∀ (T : Over Xo.X) (x y : T ⟶ Xo.univ.t.G), x ^ (N ^ 2) = 1 → y ^ (N ^ 2) = 1 → ∃ e : Γ(T.left, ⊤)ˣ, WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e ∧ e ^ (N ^ 2) = 1 ∧ ∀ e', WeilIs Xo.univ.t.G Xo.univ.t.PP.L ((N : ℤ) ^ 2) x y e' → e' = e) ∧ ∀ (B : Type) [CommRing B] (u : B) (k : ℕ), IsUnit (k : B) → u ^ k = 1 → (∀ (K : Type) [Field K] [IsAlgClosed K] (χ : B →+∗ K), χ u = 1) → u = 1

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.ValAtAlgebraMap

theorem **one_le_rd** : 1 ≤ PT.rd E

theorem **phi_nonneg** (X : Data E r) (th : ℝ) (x : Frame X.Ψ r) : 0 ≤ phiFD X th x

theorem **phi_inv** (X : Data E r) (hr : 0 < r) (th : ℝ) : IsInvFn X (phiFD X th)

theorem **phi_lower** (X : Data E r) (hf : 0 < PT.f E) {th : ℝ} (hth : 0 < th) : ∃ ca ε0 : ℝ, 0 < ca ∧ 0 < ε0 ∧ ∀ x ∈ X.dom, X.la1 x < ε0 → ca / X.la1 x ^ 2 ≤ phiFD X th x

theorem **phi_le** (X : Data E r) (th : ℝ) (x : Frame X.Ψ r) : phiFD X th x ≤ 2 ∗ PT.f E / X.la1 x ^ 2

theorem **phi_threshold** (X : Data E r) (th : ℝ) {th' : ℝ} (h' : 0 < th') : ∃ C : ℝ, ∀ x, phiFD X th x ≤ phiFD X th' x + C

theorem **decay_threshold** (X : Data E r) (th : ℝ) {th' : ℝ} (h' : 0 < th') (i : ℕ) (g : Frame X.Ψ r → ℝ) (K : ℝ) (hK : ∀ x ∈ X.dom, g x ≤ K ∗ Real.exp (-(2 ∗ (i : ℝ) ∗ phiFD X th' x))) : ∃ K' : ℝ, ∀ x ∈ X.dom, g x ≤ K' ∗ Real.exp (-(2 ∗ (i : ℝ) ∗ phiFD X th x))

theorem **gap_of_adjacent** {p : ℕ} (G L : Fin r → ℤ) (h : ∀ a b : Fin r, (b : ℕ) = a + 1 → (b : ℕ) ≠ p → G b + 2 ∗ (L a - L b) ≤ G a) (a b : Fin r) (hab : a ≤ b) (hp : (b : ℕ) < p ∨ p ≤ (a : ℕ)) : G b + 2 ∗ (L a - L b) ≤ G a

theorem **gaps_all** (X : Data E r) (L G : Wt X.Ψ r) (T : ℕ) (hL : IsStrictDom X L) (h : ∀ τ : X.Ψ, (∀ a b : Fin r, (b : ℕ) = a + 1 → (b : ℕ) ≠ X.p0 → G τ b + 2 ∗ (L τ a - L τ b) ≤ G τ a) ∧ ∀ a b : Fin r, (a : ℕ) < X.p0 → X.p0 ≤ (b : ℕ) → 19 ∗ (T : ℤ) ≤ 10 ∗ (L τ b - L τ a) ∧ 2 ∗ (L τ (r - 1, by have

theorem **margin_numbers** {f r D T u : ℝ} (hf : 1 ≤ f) (hr : 8 ≤ r) (hD : 2 ∗ D = f ∗ (r ∗ (r - 1))) (hT : 11 ∗ D ^ 2 ≤ T) (hu0 : 0 ≤ u) (hu : 2 ∗ u ≤ f ∗ r) : 4 ∗ f ∗ r ^ 2 < 5 ∗ f ∗ r ^ 2 ∧ 5 ∗ f ∗ r ^ 2 + 2 ∗ T < 19 / 5 ∗ T ∧ (5 ∗ f ∗ r ^ 2) ^ 2 ∗ u + 5 ∗ f ∗ r ^ 2 < 19 / 5 ∗ T + D ∗ (19 / 10 ∗ T) ∧ 2 ∗ r < 19 / 5 ∗ T

theorem **finrank_host** (D : D7) : Module.finrank ℚ D.host.E / 2 = D.R.f

theorem **f_pos** (D : D7) : 0 < D.R.f

theorem **two_mul_half** (p : ℕ) : (2 : ℝ) * ((p * (p - 1) / 2 : ℕ) : ℝ) = p * (p - 1)

theorem **two_mul_dimY** : (2 : ℝ) * WeightFfamilies.dimY D j = D.R.f * (D.R.r * (D.R.r - 1))

theorem **wt_ext** {u v : WeightFfamilies.Wt D j} (h1 : u.aP = v.aP) (h2 : u.aM = v.aM) (h3 : u.n = v.n) : u = v

theorem **valAt_algebraMap** (k : (T.U j T.σ₀).Comp) (x : ((T.U j T.σ₀).X k).dom) (a : T.Q.c.F) : valAt h j hj k x (algebraMap T.Q.c.F (T.R j) a) = T.σ₀ a

theorem **homRel_of_cusp** (u : WeightFfamilies.Wt D j) {m : ℕ} (hm : WeightFfamilies.dimY D j + 2 ≤ m) (F : Fin m → T.R j) (hF : ∀ a, F a ∈ T.C j u) (k : (T.U j T.σ₀).Comp) (v : Fin m → Frame ((T.U j T.σ₀).X k).Ψ D.R.r → ℂ) (hv : ∀ a (x : ((T.U j T.σ₀).X k).dom), v a x.1 = valAt h j hj k x (F a)) : HomRel T.σ₀ ((T.U j T.σ₀).X k) v

theorem **pow_mem_cusp** (u : WeightFfamilies.Wt D j) {f : T.R j} (hf : f ∈ T.C j u) {n : ℕ} (hn : 1 ≤ n) : f ^ n ∈ T.C j (wsmul n u)

theorem **archAt_nonempty** : ∀ (D : D7) (T : Top D) (h : LawsP T) (j : ℕ) (hj : InR D j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ) (CRm c : ℝ) (N : ℕ) (w : Fin N → (recB h.toLawsCore).Cusp j wL) (i : ℕ) (et : (recB h.toLawsCore).Cusp j (etWt D j i)) (w1 w2 : WeightFamilies.Wt D j) (th1 : (recB h.toLawsCore).Cusp j w1) (th2 : (recB h.toLawsCore).Cusp j w2), FamilyHypotheses.HypPage large₂ (recPJ h j hj) H C2 A2 CR CT ka wL wF Tn nP nW h0 → j < D.R.m → 8 ≤ D.R.r → 0 < c → (Tn : ℝ) * c * g0 D ≤ 1 → 1 ≤ WeightFamilies.RRv D CRm → (D.R.r : ℝ) * (1 + WeightFamilies.c1 D CRm Tn c j / (1 / (2 * (5 : ℝ) ^ 4 * |(NumberField.discr D.host.E : ℝ)| ^ ((1 : ℝ) / D.R.f)))) ≤ 2 * Tn → 2 ≤ WeightFamilies.dimY D j → 0 < i → IsDoubledWeightWithShift wL w1 nW → IsG2 wL wF w2 → (∀ k : (T.U j T.σ₀).Comp, ∀ x ∈ ((T.U j T.σ₀).X k).dom, ∃ a, rdR h

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.WeightVariableFormula

theorem **exists_order** [Finite σ] (w : σ → ℕ) : ∃ m : MonomialOrder.{0,0} σ, ∀ c d : σ →₀ ℕ, Finsupp.weight w c < Finsupp.weight w d → c <[m] d

theorem **le_of_le_add_disj** {a s d : σ →₀ ℕ} (h : a ≤ s + d) (hd : ∀ v, a v = 0 ∨ d v = 0) : a ≤ s

theorem **div_ideal** {ι : Type} (f : ι → MvPolynomial σ K) (hf : ∀ i, f i ≠ 0) (S : Set ι) (g : MvPolynomial σ K) : ∃ s, g - s ∈ Ideal.span (f '' S) ∧ (s ≠ 0 → ∀ i ∈ S, ¬ m.degree (f i) ≤ m.degree s)

theorem **groebner** {ι : Type} (f : ι → MvPolynomial σ K) (hf : ∀ i, f i ≠ 0) (hd : ∀ i j, i ≠ j → ∀ v, m.degree (f i) v = 0 ∨ m.degree (f j) v = 0) (S : Finset ι) : ∀ h ∈ Ideal.span (f '' S), h ≠ 0 → ∃ i ∈ S, m.degree (f i) ≤ m.degree h

theorem **nzd_of_groebner** {ι : Type} [Finite ι] (f : ι → MvPolynomial σ K) (hf : ∀ i, f i ≠ 0) (hd : ∀ i j, i ≠ j → ∀ v, m.degree (f i) v = 0 ∨ m.degree (f j) v = 0) (D : MvPolynomial σ K) (hD : D ≠ 0) (hD' : ∀ i v, m.degree (f i) v = 0 ∨ m.degree D v = 0) (g : MvPolynomial σ K) (hg : D * g ∈ Ideal.span (Set.range f)) : g ∈ Ideal.span (Set.range f)

theorem **degree_sum_monomial** {κ : Type} [Fintype κ] (w : σ → ℕ) (hm : ∀ c d : σ →₀ ℕ, Finsupp.weight w c < Finsupp.weight w d → c <[m] d) (e : κ → σ →₀ ℕ) (c : κ → K) (k0 : κ) (h0 : c k0 ≠ 0) (h : ∀ k, k ≠ k0 → c k ≠ 0 → Finsupp.weight w (e k) < Finsupp.weight w (e k0)) : (∑ k, monomial (e k) (c k)) ≠ 0 ∧ m.degree (∑ k, monomial (e k) (c k)) = e k0

theorem **fac_pos** (m : σ →₀ ℕ) : 0 < fac m

theorem **fac_add** (m : σ →₀ ℕ) (v : σ) : fac (m + Finsupp.single v 1) = fac m * (m v + 1)

theorem **fock_monomial** (m : σ →₀ ℕ) (c : K) (g : MvPolynomial σ K) : fock σ K (monomial m c) g = c * (fac m * coeff m g)

theorem **fock_eq** (f g : MvPolynomial σ K) : fock σ K f g = ∑ m ∈ f.support, coeff m f * (fac m * coeff m g)

theorem **fock_symm** (f g : MvPolynomial σ K) : fock σ K f g = fock σ K g f

theorem **fock_X_mul** (v : σ) (f g : MvPolynomial σ K) : fock σ K (X v * f) g = fock σ K f (pderiv v g)

theorem **eq_zero_of_fock** [CharZero K] (g : MvPolynomial σ K) (h : ∀ m, fock σ K (monomial m 1) g = 0) : g = 0

theorem **eq_zero_of_fock_conj** (g : MvPolynomial σ ℂ) (h : fock σ ℂ g (map (starRingEnd ℂ) g) = 0) : g = 0

theorem **prod_pow_weight** {A : Type} [CommRing A] (w : σ → ρ) (t : ρ → A) (m : σ →₀ ℕ) : m.prod (fun v e => t (w v) ^ e) = ∏ r, t r ^ (Finsupp.weight (fun v => (Pi.single (w v) 1 : ρ → ℕ)) m r)

theorem **aeval_scale_monomial** {A : Type} [CommRing A] [Algebra K A] (w : σ → ρ) (t : ρ → A) (x : σ → A) (m : σ →₀ ℕ) (c : K) : aeval (fun v => t (w v) * x v) (monomial m c) = (∏ r, t r ^ (Finsupp.weight (fun v => (Pi.single (w v) 1 : ρ → ℕ)) m r)) * aeval x (monomial m c)

theorem **aeval_scale** {A : Type} [CommRing A] [Algebra K A] {w : σ → ρ} {a : ρ → ℕ} {P : MvPolynomial σ K} (hP : Deg w a P) (t : ρ → A) (x : σ → A) : aeval (fun v => t (w v) * x v) P = (∏ r, t r ^ a r) * aeval x P

theorem **coeff_scale** (w : σ → ρ) (t : ρ → K) (m : σ →₀ ℕ) (P : MvPolynomial σ K) : coeff m (aeval (fun v => (C (t (w v)) * X v : MvPolynomial σ K)) P) = (∏ r, t r ^ (Finsupp.weight (fun v => (Pi.single (w v) 1 : ρ → ℕ)) m r)) * coeff m P

theorem **deg_of_scale** [CharZero K] {w : σ → ρ} {a : ρ → ℕ} {P : MvPolynomial σ K} (h : ∀ t : ρ → K, (∀ r, t r ≠ 0) → aeval (fun v => C (t (w v)) * X v) P = C (∏ r, t r ^ a r) * P) : Deg w a P

theorem **weight_apply** [Fintype σ] (w : σ → ρ) (m : σ →₀ ℕ) (r : ρ) : Finsupp.weight (fun v => (Pi.single (w v) 1 : ρ → ℕ)) m r = ∑ v, if w v = r then m v else 0

theorem **finite_deg** [Fintype σ] (w : σ → ρ) (a : ρ → ℕ) : Module.Finite K (weightedHomogeneousSubmodule K (fun v => (Pi.single (w v) 1 : ρ → ℕ)) a)

theorem **fock_qf** (i : ι) (f g : MvPolynomial σ K) : fock σ K (f * qf x y K i) g = fock σ K f (lp x y i g)

theorem **fock_ideal** {f h : MvPolynomial σ K} (hf : f ∈ Ideal.span (Set.range (qf x y K))) (hh : ∀ i, lp x y i h = 0) : fock σ K f h = 0

theorem **harm_ideal_zero** (φ : K →+* ℂ) {h : MvPolynomial σ K} (hf : h ∈ Ideal.span (Set.range (qf x y K))) (hh : ∀ i, lp x y i h = 0) : h = 0

theorem **exists_harm** (φ : K →+* ℂ) {ρ : Type} [Fintype ρ] [DecidableEq ρ] (w : σ → ρ) (wx wy : ι → ρ) (hx : ∀ i b, w (x i b) = wx i) (hy : ∀ i b, w (y i b) = wy i) {a : ρ → ℕ} {P : MvPolynomial σ K} (hP : Deg w a P) : ∃ h, Deg w a h ∧ (∀ i, lp x y i h = 0) ∧ P - h ∈ Ideal.span (Set.range (qf x y K))

theorem **prod_emb_monomial** (e : S → Fin n × Fin n →₀ ℕ) : ∏ s, blockEmb K s (monomial (e s) (1 : K)) = monomial (Finsupp.equivFunOnFinite.symm fun v : S × (Fin n × Fin n) => e v.1 v.2) 1

theorem **monomial_eq_prod** (m : S × (Fin n × Fin n) →₀ ℕ) : monomial m (1 : K) = ∏ s, blockEmb K s (monomial (part m s) (1 : K))

theorem **tens_monomial** (φ : S → Module.End K (MvPolynomial (Fin n × Fin n) K)) (m : S × (Fin n × Fin n) →₀ ℕ) : tens φ (monomial m 1) = ∏ s, blockEmb K s (φ s (monomial (part m s) 1))

theorem **tens_prod** (φ : S → Module.End K (MvPolynomial (Fin n × Fin n) K)) (f : S → MvPolynomial (Fin n × Fin n) K) : tens φ (∏ s, blockEmb K s (f s)) = ∏ s, blockEmb K s (φ s (f s))

theorem **lin_ext** {M : Type} [AddCommGroup M] [Module K M] (F G : MvPolynomial (S × (Fin n × Fin n)) K →ₗ[K] M) (h : ∀ f : S → MvPolynomial (Fin n × Fin n) K, F (∏ s, blockEmb K s (f s)) = G (∏ s, blockEmb K s (f s))) : F = G

theorem **pev_self** (t : S) (z : S × (Fin n × Fin n) → K) (f : MvPolynomial (Fin n × Fin n) K) : pev t z (blockEmb K t f) = f

theorem **pev_ne** {s t : S} (h : s ≠ t) (z : S × (Fin n × Fin n) → K) (f : MvPolynomial (Fin n × Fin n) K) : pev t z (blockEmb K s f) = C (eval (fun v => z (s, v)) f)

theorem **pev_prod** (t : S) (z : S × (Fin n × Fin n) → K) (f : S → MvPolynomial (Fin n × Fin n) K) : pev t z (∏ s, blockEmb K s (f s)) = (∏ s ∈ Finset.univ.erase t, eval (fun v => z (s, v)) (f s)) • f t

theorem **eq_zero_of_pev** [Infinite K] (t : S) (G : MvPolynomial (S × (Fin n × Fin n)) K) (h : ∀ z, pev t z G = 0) : G = 0

theorem **mem_tensor** [Infinite K] (V : S → Submodule K (MvPolynomial (Fin n × Fin n) K)) (G : MvPolynomial (S × (Fin n × Fin n)) K) (h : ∀ t z, pev t z G ∈ V t) : G ∈ tensorLat K K V

theorem **rowGen_eval** (a b : Fin n) (c : ℚ) (f : MvPolynomial (Fin n × Fin n) ℚ) : (rowGen ℚ a b f).eval (C c) = rowAct ℚ (1 + Matrix.single b a c) f

theorem **closed_of_stable** (M : Submodule ℚ (MvPolynomial (Fin n × Fin n) ℚ)) (h : ∀ g : GL (Fin n) ℚ, ∀ f ∈ M, rowAct ℚ (g : Matrix (Fin n) (Fin n) ℚ) f ∈ M) : RowDivClosed M

theorem **irr** {K : Type} [Field K] [Algebra ℚ K] (ν : Fin n → ℕ) (hν : Antitone ν) (M : Submodule ℚ (MvPolynomial (Fin n × Fin n) ℚ)) (hle : M ≤ dualWeyl ℚ n n ν) (hne : M ≠ ⊥) (h : ∀ g : GL (Fin n) ℚ, ∀ f ∈ M, rowAct ℚ (g : Matrix (Fin n) (Fin n) ℚ) f ∈ M) (N : Submodule K (MvPolynomial (Fin n × Fin n) K)) (hN : ∀ f ∈ M, map (algebraMap ℚ K) f ∈ N) : dualWeyl K n n ν ≤ N

@[simp] theorem **val_inl** {A : Type} [CommRing A] (g : Ix → Matrix (Fin n) (Fin n) A) (t : Ix) (i : Fin p) (b : Fin n) : val h g ((t, Sum.inl i), b) = g t b (cx h i)

@[simp] theorem **val_inr** {A : Type} [CommRing A] (g : Ix → Matrix (Fin n) (Fin n) A) (t : Ix) (j : Fin q) (b : Fin n) : val h g ((t, Sum.inr j), b) = (g t).adjugate (cy h j) b

theorem **map_val** {A B : Type} [CommRing A] [CommRing B] [Algebra K A] [Algebra K B] (φ : A →ₐ[K] B) (g : Ix → Matrix (Fin n) (Fin n) A) (P : MvPolynomial (Lt Ix p q n) K) : φ (aeval (val h g) P) = aeval (val h fun t => (g t).map φ) P

theorem **cx_ne_cy** (i : Fin p) (j : Fin q) : cy h j ≠ cx h i

theorem **val_ideal** {A : Type} [CommRing A] [Algebra K A] (g : Ix → Matrix (Fin n) (Fin n) A) {P : MvPolynomial (Lt Ix p q n) K} (hP : P ∈ II Ix p q n K) : aeval (val h g) P = 0

theorem **aeval_shift** {A : Type} [CommRing A] [Algebra K A] (v : Lt Ix p q n → A) (ρ ρ' : Row Ix p q) (c : K) (P : MvPolynomial (Lt Ix p q n) K) : aeval v (shift ρ ρ' c P) = aeval (fun ℓ => if ℓ.1 = ρ then v ℓ + algebraMap K A c * v (ρ', ℓ.2) else v ℓ) P

theorem **adj_transv** {A : Type} [CommRing A] {l j : Fin n} (hlj : l ≠ j) (c : A) : (Matrix.transvection l j c).adjugate = Matrix.transvection l j (-c)

theorem **val_transv** {A : Type} [CommRing A] [Algebra K A] (g : Ix → Matrix (Fin n) (Fin n) A) (t : Ix) {l j : Fin n} (hlj : l < j) (c : K) {P : MvPolynomial (Lt Ix p q n) K} (hP : Inv P) : aeval (val h fun s => g s * if s = t then Matrix.transvection l j (algebraMap K A c) else 1) P = aeval (val h g) P

theorem **prod_block** {A : Type} [CommMonoid A] (t : Ix) (c : Fin p ⊕ Fin q → A) (a : Row Ix p q → ℕ) : ∏ r : Row Ix p q, (if r.1 = t then c r.2 else 1) ^ a r = (∏ i, c (Sum.inl i) ^ a (t, Sum.inl i)) * ∏ j, c (Sum.inr j) ^ a (t, Sum.inr j)

theorem **val_diag** {A : Type} [CommRing A] [Algebra K A] (g : Ix → Matrix (Fin n) (Fin n) A) (t : Ix) (d : Fin n → A) {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) : aeval (val h fun s => g s * if s = t then Matrix.diagonal d else 1) P = (∏ k, d k ^ mu h a t k) * aeval (val h g) P

theorem **pev_cof_mem** [CharZero K] {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) (hI : Inv P) (t : Ix) (z : Ix × (Fin n × Fin n) → K) : pev t z (aeval (val h (gen K n)) P) ∈ dualWeyl K n n (mu h a t)

theorem **cof_mem** [CharZero K] {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) (hI : Inv P) : aeval (val h (gen K n)) P ∈ tensorLat K K fun t => dualWeyl K n n (mu h a t)

theorem **aeval_act** {A : Type} [CommRing A] [Algebra K A] (v : Lt Ix p q n → A) (g : Ix → GL (Fin n) K) (P : MvPolynomial (Lt Ix p q n) K) : aeval v (act g P) = aeval (fun ℓ => ∑ j, algebraMap K A (mat g ℓ.1 ℓ.2 j) * v (ℓ.1, j)) P

theorem **cof_act** (g : Ix → GL (Fin n) K) {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) : aeval (val h (gen K n)) (act g P) = (∏ t, ((g t : Matrix (Fin n) (Fin n) K).det)⁻¹ ^ (∑ j, a (t, Sum.inr j))) • trAct g (aeval (val h (gen K n)) P)

theorem **act_deg** [CharZero K] (g : Ix → GL (Fin n) K) {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) : Deg Prod.fst a (act g P)

theorem **shift_act** (g : Ix → GL (Fin n) K) {ρ ρ' : Row Ix p q} (hm : mat g ρ = mat g ρ') (c : K) (P : MvPolynomial (Lt Ix p q n) K) : shift ρ ρ' c (act g P) = act g (shift ρ ρ' c P)

theorem **act_inv** (g : Ix → GL (Fin n) K) {P : MvPolynomial (Lt Ix p q n) K} (hP : Inv P) : Inv (act g P)

theorem **pderiv_shift** (ρ ρ' : Row Ix p q) (c : K) (v : Lt Ix p q n) (P : MvPolynomial (Lt Ix p q n) K) : pderiv v (shift ρ ρ' c P) = shift ρ ρ' c (pderiv v P) + if v.1 = ρ' then c • shift ρ ρ' c (pderiv (ρ, v.2) P) else 0

theorem **harm_shift** (t : Ix) (c : K) {P : MvPolynomial (Lt Ix p q n) K} (hP : ∀ i, lp xl yl i P = 0) : (∀ u u' : Fin p, ∀ i, lp xl yl i (shift (t, Sum.inl u) (t, Sum.inl u') c P) = 0) ∧ (∀ u u' : Fin q, ∀ i, lp xl yl i (shift (t, Sum.inr u) (t, Sum.inr u') c P) = 0)

theorem **ideal_shift_of** {ρ ρ' : Row Ix p q} {c : K} (hq : ∀ i, shift ρ ρ' c (qf xl yl K i) ∈ II Ix p q n K) {P : MvPolynomial (Lt Ix p q n) K} (hP : P ∈ II Ix p q n K) : shift ρ ρ' c P ∈ II Ix p q n K

theorem **ideal_shift** (t : Ix) (c : K) {P : MvPolynomial (Lt Ix p q n) K} (hP : P ∈ II Ix p q n K) : (∀ u u' : Fin p, shift (t, Sum.inl u) (t, Sum.inl u') c P ∈ II Ix p q n K) ∧ (∀ u u' : Fin q, shift (t, Sum.inr u) (t, Sum.inr u') c P ∈ II Ix p q n K)

theorem **exists_harm_inv** (φ : K →+* ℂ) {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) (hI : Inv P) : ∃ Q : MvPolynomial (Lt Ix p q n) K, Deg Prod.fst a Q ∧ Inv Q ∧ (∀ i, lp xl yl i Q = 0) ∧ aeval (val h (gen K n)) Q = aeval (val h (gen K n)) P

theorem **xm_cx** (t : Ix) (i : Fin p) (b : Fin n) : xm p q n K t b (cx h i) = X ((t, Sum.inl i), b)

theorem **ym_cy** (t : Ix) (j : Fin q) (b : Fin n) : ym p q n K t (cy h j) b = X ((t, Sum.inr j), b)

theorem **det_wm_ne** (t : Ix) : (wm p q n K t).det ≠ 0

theorem **th_eq_zero** {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) (h0 : aeval (val h (gen K n)) P = 0) : aeval (th h K) P = 0

theorem **th_mod** (t : Ix) (j : Fin q) (b : Fin n) : (xm p q n K t).det * X ((t, Sum.inr j), b) - th h K ((t, Sum.inr j), b) ∈ II Ix p q n K

theorem **weight_variable_formula** (t : Ix) (i : Fin p) (b : Fin n) : (wt (((t, Sum.inl i), b) : Lt Ix p q n) : ℤ) = n ^ 2 + 2 * b * i - b ^ 2

theorem **wt_y** (t : Ix) (j : Fin q) (b : Fin n) : (wt (((t, Sum.inr j), b) : Lt Ix p q n) : ℤ) = 2 * b * (j + 1)

theorem **degree_qf** (m : MonomialOrder.{0,0} (Lt Ix p q n)) (hm : ∀ c d : Lt Ix p q n →₀ ℕ, Finsupp.weight wt c < Finsupp.weight wt d → c <[m] d) (i : Ix × Fin p × Fin q) : (qf xl yl K i : MvPolynomial (Lt Ix p q n) K) ≠ 0 ∧ m.degree (qf xl yl K i) = Finsupp.single (xl i (lead h i)) 1 + Finsupp.single (yl i (lead h i)) 1

theorem **degree_det** (m : MonomialOrder.{0,0} (Lt Ix p q n)) (hm : ∀ c d : Lt Ix p q n →₀ ℕ, Finsupp.weight wt c < Finsupp.weight wt d → c <[m] d) (t : Ix) : (xm p q n K t).det ≠ 0 ∧ m.degree (xm p q n K t).det = ∑ k : Fin n, if hk : (k : ℕ) < p then Finsupp.single ((t, Sum.inl ⟨k, hk⟩), k) 1 else 0

theorem **det_xm_nzd** (t : Ix) (g : MvPolynomial (Lt Ix p q n) K) (hg : (xm p q n K t).det * g ∈ II Ix p q n K) : g ∈ II Ix p q n K

theorem **cof_inj** (φ : K →+* ℂ) {a : Row Ix p q → ℕ} {P : MvPolynomial (Lt Ix p q n) K} (hP : Deg Prod.fst a P) (hH : ∀ i, lp xl yl i P = 0) (h0 : aeval (val h (gen K n)) P = 0) : P = 0

theorem **eval_gshift** [Algebra ℚ K] (ρ ρ' : Row Ix p q) (c : K) (P : MvPolynomial (Lt Ix p q n) ℚ) : Polynomial.eval₂ (map (algebraMap ℚ K)) (C c) (gshift ρ ρ' P) = shift ρ ρ' c (map (algebraMap ℚ K) P)

theorem **shift_map** [Algebra ℚ K] {ρ ρ' : Row Ix p q} {P : MvPolynomial (Lt Ix p q n) ℚ} (hP : ∀ c : ℚ, shift ρ ρ' c P = P) (c : K) : shift ρ ρ' c (map (algebraMap ℚ K) P) = map (algebraMap ℚ K) P

theorem **NN_map** [Algebra ℚ K] (t : Ix) (a : Row Ix p q → ℕ) {f : MvPolynomial (Fin n × Fin n) ℚ} (hf : f ∈ NN h ℚ t a) : map (algebraMap ℚ K) f ∈ NN h K t a

theorem **NN_le** [CharZero K] (t : Ix) (a : Row Ix p q → ℕ) : NN h K t a ≤ dualWeyl K n n (mu h a t)

theorem **NN_act** [CharZero K] (t : Ix) (a : Row Ix p q → ℕ) (g : GL (Fin n) K) (f : MvPolynomial (Fin n × Fin n) K) (hf : f ∈ NN h K t a) : rowAct K (g : Matrix (Fin n) (Fin n) K) f ∈ NN h K t a

theorem **shift_rename_self** {m : ℕ} {ρ : Fin m → Row Ix p q} (hρ : Function.Injective ρ) (e : Fin n → Fin n) (i i' : Fin m) (c : K) (H : MvPolynomial (Fin n × Fin m) K) : shift (ρ i) (ρ i') c (rename (fun v => ((ρ v.2, e v.1) : Lt Ix p q n)) H) = rename (fun v => ((ρ v.2, e v.1) : Lt Ix p q n)) (colAdd K i' i c H)

theorem **shift_rename_ne** {m : ℕ} {ρ : Fin m → Row Ix p q} (e : Fin n → Fin n) {r : Row Ix p q} (hr : ∀ i, ρ i ≠ r) (r' : Row Ix p q) (c : K) (H : MvPolynomial (Fin n × Fin m) K) : shift r r' c (rename (fun v => ((ρ v.2, e v.1) : Lt Ix p q n)) H) = rename (fun v => ((ρ v.2, e v.1) : Lt Ix p q n)) H

theorem **aeval_hw** {m : ℕ} (hm : m ≤ n) (ν : Fin m → ℕ) (z : Fin n × Fin m → K) (hz : ∀ v, z v = if (v.1 : ℕ) = v.2 then 1 else 0) : aeval z (hw K hm ν) = 1

theorem **NN_ne** [CharZero K] (t : Ix) (a : Row Ix p q → ℕ) (ha : Antitone fun i => a (t, Sum.inl i)) (hb : Antitone fun i => a (t, Sum.inr i)) : NN h K t a ≠ ⊥

theorem **cof_surj** (φ : K →+* ℂ) (a : Row Ix p q → ℕ) (ha : ∀ t, Antitone fun i => a (t, Sum.inl i)) (hb : ∀ t, Antitone fun i => a (t, Sum.inr i)) (hμ : ∀ t, Antitone (mu h a t)) {F : MvPolynomial (Ix × Fin n × Fin n) K} (hF : F ∈ tensorLat K K fun t => dualWeyl K n n (mu h a t)) : ∃ Q : MvPolynomial (Lt Ix p q n) K, Deg Prod.fst a Q ∧ Inv Q ∧ (∀ i, lp xl yl i Q = 0) ∧ aeval (val h (gen K n)) Q = F

theorem **dd_pow_mem** (e : ℕ) : dd K n ^ e ∈ dualWeyl K n n fun _ => e

theorem **dd_mul_mem** {ν μ : Fin n → ℕ} {e : ℕ} (hμ : ∀ k, ν k + e = μ k) {f : MvPolynomial (Fin n × Fin n) K} (hf : f ∈ dualWeyl K n n ν) : dd K n ^ e * f ∈ dualWeyl K n n μ

theorem **rowAct_dd** (g : Matrix (Fin n) (Fin n) K) : rowAct K g (dd K n) = C g.det * dd K n

theorem **dd_ne** : dd K n ≠ 0

theorem **exists_dd_mul** [CharZero K] {ν μ : Fin n → ℕ} {e : ℕ} (hν : Antitone ν) (hμ : ∀ k, ν k + e = μ k) {f : MvPolynomial (Fin n × Fin n) K} (hf : f ∈ dualWeyl K n n μ) : ∃ f' ∈ dualWeyl K n n ν, dd K n ^ e * f' = f

theorem **emb_dd** (t : Ix) : blockEmb K t (dd K n) = (gen K n t).det

theorem **tensor_iff** [CharZero K] {ν μ : Ix → Fin n → ℕ} {e : Ix → ℕ} (hν : ∀ t, Antitone (ν t)) (hμ : ∀ t k, ν t k + e t = μ t k) (G : MvPolynomial (Ix × Fin n × Fin n) K) : G ∈ tensorLat K K (fun t => dualWeyl K n n (μ t)) ↔ ∃ F ∈ tensorLat K K (fun t => dualWeyl K n n (ν t)), G = (∏ t, (gen K n t).det ^ e t) * F

theorem **rt0_conj_neg_ne_zero** (H : Host E) : starRingEnd ℂ H.rt0 = -H.rt0 ∧ H.rt0 ≠ 0

theorem **not_psi_conj** {τ : E →+* ℂ} (hτ : InPsi H τ) : ¬ InPsi H (ComplexEmbedding.conjugate τ)

theorem **psi_conj** {τ : E →+* ℂ} (hτ : ¬ InPsi H τ) : InPsi H (ComplexEmbedding.conjugate τ)

theorem **conj_conj** (τ : E →+* ℂ) : ComplexEmbedding.conjugate (ComplexEmbedding.conjugate τ) = τ

theorem **psi_pl** (t : Ix) : InPsi H (pl t)

theorem **mult_pl** (t : Ix) : Dw.mult (pl t) = Dw.p

theorem **mult_conj** (t : Ix) : Dw.mult (ComplexEmbedding.conjugate (pl t)) = Dw.q

theorem **pl_ne** (s t : Ix) : pl s ≠ ComplexEmbedding.conjugate (pl t)

theorem **conj_pl_inj** {s t : Ix} (e : ComplexEmbedding.conjugate (pl s) = ComplexEmbedding.conjugate (pl t)) : s = t

theorem **cover** (σ : E →+* ℂ) : (∃ t, pl t = σ) ∨ ∃ t, ComplexEmbedding.conjugate (pl t) = σ

theorem **fw_inl** (t : Ix) (i : Fin Dw.p) (b : Fin n) : fw Dw he ((t, Sum.inl i), b) = (pl t, Fin.cast (mult_pl Dw he _).symm i, b)

theorem **fw_inr** (t : Ix) (i : Fin Dw.q) (b : Fin n) : fw Dw he ((t, Sum.inr i), b) = (ComplexEmbedding.conjugate (pl t), Fin.cast (mult_conj Dw he _).symm i, b)

theorem **fw_inj** : Function.Injective (fw (n

theorem **row_cover** (σ : E →+* ℂ) (i : Fin (Dw.mult σ)) : ∃ ρ : Row Ix Dw.p Dw.q, ∀ b : Fin n, fw Dw he (ρ, b) = (σ, i, b)

theorem **fw_surj** : Function.Surjective (fw (n

theorem **cof_fw** (h : Dw.p + Dw.q ≤ n) (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : cof Dw pl K n (rename (fw Dw he) P) = aeval (val h (gen K n)) P

theorem **letAct_fw** (g : Ix → GL (Fin n) K) (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : letAct Dw pl K g (rename (fw Dw he) P) = rename (fw Dw he) (act g P)

theorem **rowShift_fw** {ρ ρ' : Row Ix Dw.p Dw.q} {σ : E →+* ℂ} {i i' : Fin (Dw.mult σ)} (h1 : ∀ b : Fin n, fw Dw he (ρ, b) = (σ, i, b)) (h2 : ∀ b : Fin n, fw Dw he (ρ', b) = (σ, i', b)) (c : K) (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : rowShift Dw.mult n K σ i i' c (rename (fw Dw he) P) = rename (fw Dw he) (shift ρ ρ' c P)

theorem **inv_fw** (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : RowInv Dw.mult n K (rename (fw Dw he) P) ↔ Inv P

theorem **deg_fw** (a : (E →+* ℂ) × ℕ → ℕ) (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : RowDeg Dw.mult n K (fun τ i => a (τ, i)) (rename (fw Dw he) P) ↔ Deg Prod.fst (aa Dw pl a) P

theorem **lap_fw** (t : Ix) (i : Fin Dw.p) (j : Fin Dw.q) (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : lap Dw.mult n K 1 (pl t) (ComplexEmbedding.conjugate (pl t)) (Fin.cast (mult_pl Dw he _).symm i) (Fin.cast (mult_conj Dw he _).symm j) (rename (fw Dw he) P) = rename (fw Dw he) (lp xl yl (t, i, j) P)

theorem **harm_fw** (P : MvPolynomial (Lt Ix Dw.p Dw.q n) K) : rename (fw Dw he) P ∈ harmK Dw.mult n K ComplexEmbedding.conjugate (InPsi H) (fun _ => 1) ↔ ∀ i, lp xl yl i P = 0

theorem **degC_eq** (a : (E →+* ℂ) × ℕ → ℕ) (t : Ix) : degC Dw pl a t = ∑ j, aa Dw pl a (t, Sum.inr j)

theorem **mu_eq** (h : Dw.p + Dw.q ≤ n) {a : (E →+* ℂ) × ℕ → ℕ} {s : Ix → ℤ} {lam : Ix → DomWeight n} (hw : WtIs Dw pl a s lam) (t : Ix) (k : Fin n) : (mu h (aa Dw pl a) t k : ℤ) = (lam t).w k − s t + degC Dw pl a t

theorem **shift_le** (lam : DomWeight n) (k : Fin n) : lam.detShift ≤ lam.w k

theorem **exp_nonneg** (h : Dw.p + Dw.q ≤ n) {a : (E →+* ℂ) × ℕ → ℕ} {s : Ix → ℤ} {lam : Ix → DomWeight n} (hw : WtIs Dw pl a s lam) (hn : 0 < n) (t : Ix) : 0 ≤ (degC Dw pl a t : ℤ) + (lam t).detShift − s t

theorem **natW_add** (h : Dw.p + Dw.q ≤ n) {a : (E →+* ℂ) × ℕ → ℕ} {s : Ix → ℤ} {lam : Ix → DomWeight n} (hw : WtIs Dw pl a s lam) (t : Ix) (k : Fin n) : (lam t).natW k + ((degC Dw pl a t : ℤ) + (lam t).detShift − s t).toNat = mu h (aa Dw pl a) t k

theorem **natW_anti** (lam : DomWeight n) : Antitone lam.natW

theorem **main_field** (φ : K →+* ℂ) (h : Dw.p + Dw.q ≤ n) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → DomWeight n) (ha : ∀ τ : E →+* ℂ, Antitone fun i : ℕ => a (τ, i)) (hw : WtIs Dw pl a s lam) : (∀ P ∈ HHa Dw.mult n K ComplexEmbedding.conjugate (InPsi H) fun τ i => a (τ, i), ∃ F ∈ VV K lam, cof Dw pl K n P = detFac Dw pl K a s lam * F) ∧ (∀ F ∈ VV K lam, ∃! P : MvPolynomial (Letter (E →+* ℂ) Dw.mult n) K, (P ∈ HHa Dw.mult n K ComplexEmbedding.conjugate (InPsi H) fun τ i => a (τ, i)) ∧ cof Dw pl K n P = detFac Dw pl K a s lam * F) ∧ ∀ g : Ix → GL (Fin n) K, ∀ P ∈ HHa Dw.mult n K ComplexEmbedding.conjugate (InPsi H) fun τ i => a (τ, i), ∀ F ∈ VV K lam, cof Dw pl K n P = detFac Dw pl K a s lam * F → cof Dw pl K n (letAct Dw pl K g P) = (∏ t : Ix, Matrix.det (Units.val (g t)) ^ (−s t)) • (detFac Dw pl K a s lam * actP lam g F)

theorem **main** : ∀ (E : Type) [inst : Field E] [inst_1 : NumberField E] [inst_2 : NumberField.IsCMField E] (H : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Host E) (Dw : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum H) (n : ℕ) (Ix : Type) [inst_3 : Fintype Ix] (pl : Ix → E →+* ℂ), GlobalTheory.FrameRead.IsEnum H pl → ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum.p Dw + ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum.q Dw ≤ n → ∀ (l : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Level H) (c : ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff l) (a : (E →+* ℂ) × ℕ → ℕ) (s : Ix → ℤ) (lam : Ix → Formalism.BasicNotions.DomWeight n), (∀ (τ : E →+* ℂ), Antitone fun (i : ℕ) ↦ a (τ, i)) → GlobalTheory.FrameRead.WtIs Dw pl a s lam → (∀ P ∈ Formalism.L2.Pj.HHa (ReciprocityTheory.ChartRestriction.G.Geo.Tor.Datum.mult Dw) n (ɪ(ReciprocityTheory.ChartRestriction.G.Geo.Tor.Coeff.F c)) Numb

Development.AbelianVarietyTheory.Proofs.CuspFormEvaluations.WeightsBelow

theorem **size_le_mul** {D : D7} {j : ℕ} (x z : WeightFamilies.Wt D j) (c : ℕ) (hc : 1 ≤ c) (h1 : ∀ τ i, x.aP τ i ≤ c * z.aP τ i) (h2 : ∀ τ i, x.aM τ i ≤ c * z.aM τ i) (h3 : x.n ≤ c * z.n) : x.size ≤ (c : ℝ) * z.size

theorem **DF_le** {D : D7} {j : ℕ} (x z : WeightFamilies.Wt D j) (h1 : ∀ τ i, x.aP τ i ≤ z.aP τ i) (h2 : ∀ τ i, x.aM τ i ≤ z.aM τ i) : WeightAlgebra.DF x ≤ WeightAlgebra.DF z

theorem **weights_below** : ∀ (D : D7) (B : WeightFamilies.Prov D) (P : ModuliIntegralCharts.Prov D) (j : ℕ) (X : PJ D B P j) (large₂ : SizeData₂ D) (H : KHyp) (C2 A2 CR CT : ℝ) (ka wL wF : WeightFamilies.Wt D j) (Tn nP nW : ℕ) (h0 : ℝ), 1 ≤ j → FamilyHypotheses.HypPage large₂ X H C2 A2 CR CT ka wL wF Tn nP nW h0 → (∀ w ∈ ({wL, wF, X.wF0 wL wF, ka} : Set (WeightFamilies.Wt D j)), WeightAlgebra.DF w ≤ WeightAlgebra.DF ka ∧ w.n ≤ ka.n ∧ D.R.r ≤ w.n ∧ w.LastPos) ∧ (∀ w ∈ ({wF, X.wF0 wL wF, ka} : Set (WeightFamilies.Wt D j)), 8 * D.R.f * D.R.r + 4 ≤ w.n) ∧ (∀ w2, IsG2 wL wF w2 → WeightAlgebra.DF w2 ≤ WeightAlgebra.DF ka ∧ w2.n ≤ ka.n ∧ w2.size ≤ ((WeightFamilies.dimY D j : ℝ) + 4) * wL.size) ∧ (∀ w1, IsDoubledWeightWithShift wL w1 nW → WeightAlgebra.DF w1 ≤ WeightAlgebra.DF ka ∧ w1.n ≤ ka.n ∧ w1.size ≤ 2 * wL.size)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbsAllAtRecordGap

theorem **abs_all_at_record_gap** : ∀ Cg cb Cu cT CT C0 : ℝ, ∃ C : ℝ, 0 ≤ C ∧ ∀ Cn : ℝ, ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl : ℝ), thr ≤ C' → thr ≤ Cl → Formalism.MainNotions.BridgeLemma.AbsAllParameters D large₂ C' Cl Cg cb Cu cT CT C0 1 1 Cn C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbsConvOfInvariant

theorem **absConv_of_invariant** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), ∀ f : GA D B.V → ℂ, Continuous f → (∀ γ ∈ (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)), ∀ h, f (γ * h) = f h) → ∀ h, AbsConv B.vA.μ B.vA.ℱ B.vN.ν B.vN.𝒩 fun h₁ n => f (n.1 * embA B.d B.o h₁ * h)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbsRangeConstantsFirst

theorem **abs_range_constants_first** : ∀ Cg cb Cu cT CT C1 CMo : ℝ, ∃ C₀ : ℝ, ∀ C ≥ C₀, ∃ m₀ : ℝ, ∀ Cn : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl C0 : ℝ), Big D large₂ Cl m₀ → MomIn D large₂ C' Cl Cg cb Cu cT CT C0 Cn C1 25 CMo → ABSr D large₂ C' Cl Cg cb Cu cT CT C0 Cn C1 C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbsThetaIntegralLeSobolevSeminorm

theorem **tendsto_enorm_of_le** {v : ℕ → ℂ} {a : ℂ} {s : ℕ → ℝ≥0∞} (C : ℝ≥0) (hs : Filter.Tendsto s Filter.atTop (nhds 0)) (h : ∀ j, ‖v j − a‖ₑ ≤ C * s j) : Filter.Tendsto (fun j => ‖v j‖ₑ) Filter.atTop (nhds ‖a‖ₑ)

theorem **sobBdd_absThetaInt** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsCovolOne (ratL T S) μ ℱ → ∀ m : Measure (GA D T), m.IsHaarMeasure → ∀ Z : (WO S T → ℂ) → WO S T → ℂ, (∀ f f', IsThetaFunO ψ S T f → IsThetaFunO ψ S T f' → Z (f + f') = Z f + Z f') → (∀ (c : ℂ) f, IsThetaFunO ψ S T f → Z (c • f) = c • Z f) → Sobolev₂ ψ θ μ ℱ μ ℱ Z →

(∀ f, IsThetaFunO ψ S T f → ∀ x, om (x, 1) (Z f) = Z (om (x, 1) f)) → (∀ f, IsThetaFunO ψ S T f → Integrable (descend (GQ T) fun h => thetaKO om 1 h (Z f)) (autMeasure T m)) → SobBdd ψ θ μ ℱ fun f x => ∫⁻ q, ‖descend (GQ T) (fun h => thetaKO om x h (Z f)) q‖ₑ ∂autMeasure T m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbsThetaIntegralLeSobolevSeminormStatement

theorem **_root_.GlobalTheory.T.absThetaIntegral_le_sobolevSeminorm** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsCovolOne (ratL T S) μ ℱ → ∀ m : Measure (GA D T), m.IsHaarMeasure → ∀ Z : (WO S T → ℂ) → WO S T → ℂ, (∀ f f', IsThetaFunO ψ S T f → IsThetaFunO ψ S T f' → Z (f + f') = Z f + Z f') → (∀ (c : ℂ) f, IsThetaFunO ψ S T f → Z (c • f) = c • Z f) → Sobolev₂ ψ θ μ ℱ μ ℱ Z → (∀ f, IsThetaFunO ψ S T f → ∀ x, om (x, 1) (Z f) = Z (om (x, 1) f)) → (∀ f, IsThetaFunO ψ S T f → Integrable (descend (GQ T) fun h => thetaKO om 1 h (Z f)) (autMeasure T m)) → SobBdd ψ θ μ ℱ fun f x => ∫⁻ q, ‖descend (GQ T) (fun h => thetaKO om x h (Z f)) q‖ₑ ∂autMeasure T m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbstractDiagonalBoundCollisionsCountedOnce

theorem **diag_abstract_collisions_once** : ∀ (τ Rw I H H₁ : Type) [Fintype I] [NormedAddCommGroup H] [InnerProductSpace ℂ H] [NormedAddCommGroup H₁] [InnerProductSpace ℂ H₁] [CompleteSpace H₁] (X : Formalism.FourierDecayAlt.PB.Terms τ Rw I H H₁) (one : H) (Pg B A₁ A₂ : ℝ), 0 ≤ Pg → X.AbsConv → X.NormExpansionOn X.genSet → X.NormExpansionOn X.ngSet → X.GGBound Pg → X.RowOrth → X.CorankPos → Formalism.L2.SG.RowsOther.IdOne X one → X.pairAbs (X.big X.nzSet) ≤ B → X.pairSum (X.one X.genSet) ≤ A₁ → X.pairSum (X.one X.ngSet) ≤ A₂ → ‖X.K0 − one‖ ^ 2 ≤ 2 * Pg * (B + A₁ + A₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AbstractDiagonalBoundCollisionsCountedOnceStatement

theorem **_root_.GlobalTheory.T.abstractDiagonalBound_collisions_counted_once** : ∀ (τ Rw I H H₁ : Type) [Fintype I] [NormedAddCommGroup H] [InnerProductSpace ℂ H] [NormedAddCommGroup H₁] [InnerProductSpace ℂ H₁] [CompleteSpace H₁] (X : Formalism.FourierDecayAlt.PB.Terms τ Rw I H H₁) (one : H) (Pg B A₁ A₂ : ℝ), 0 ≤ Pg → X.AbsConv → X.NormExpansionOn X.genSet → X.NormExpansionOn X.ngSet → X.GGBound Pg → X.RowOrth → X.CorankPos → Formalism.L2.SG.RowsOther.IdOne X one → X.pairAbs (X.big X.nzSet) ≤ B → X.pairSum (X.one X.genSet) ≤ A₁ → X.pairSum (X.one X.ngSet) ≤ A₂ → ‖X.K0 − one‖ ^ 2 ≤ 2 * Pg * (B + A₁ + A₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AconvSingleList

theorem **cA_idem** (q : GlobalTheory.FinPl.SplitPt D k) : cA D (q.e + cA D q.e) = q.e + cA D q.e

theorem **idem_sq** (q : GlobalTheory.FinPl.SplitPt D k) : (q.e + cA D q.e) * (q.e + cA D q.e) = q.e + cA D q.e

theorem **aconv_cons** {G : Type} [AddMonoid G] (e : G →₀ ℂ) (l : List (G →₀ ℂ)) : aconv (e :: l) = ∑ g ∈ e.support, ∑ h ∈ (aconv l).support, Finsupp.single (g + h) (e g * aconv l h)

theorem **comb_aconv_cons** (e : WO S T →₀ ℂ) (l : List (WO S T →₀ ℂ)) (h : ∀ x ∈ e.support, ∀ y ∈ (aconv l).support, symplO S T x y = 0) (f : WO S T → ℂ) : comb ψ S T (aconv (e :: l)) f = comb ψ S T e (comb ψ S T (aconv l) f)

theorem **aconv_single_list** {G : Type} [AddMonoid G] (d : G →₀ ℂ) : aconv [d] = d

theorem **comb_aconv_away** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {S : Dbl2 ψ σ} (Q : QStep S) {A T : HSp D} (c d : WO A T →₀ ℂ) (hc : Q.Away c) (hd : ∀ u ∈ d.support, IsHalf ((ofSplit Q.q).e + cA D (ofSplit Q.q).e) u) (g : WO A T → ℂ) : comb ψ A T (aconv [c, d]) g = comb ψ A T c (comb ψ A T d g)

theorem **valK_comb** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (T₁ : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.B.V S.J.W → ℂ)) (T₂ : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.J.W S.B'.V → ℂ)), (ofSplit Q.q).IsModel ψ Q.F.G₀ Q.F.G₁ T₁ → (ofSplit Q.q).IsModel ψ Q.F.G₁ Q.F.G₂ T₂ → ∀ (U₁ : SB (Matrix Q.n₁ (Q.n₀ ⊕ Unit) Q.k) →ₗ[ℂ] (WO S.B.V₀ S.J.W₀ → ℂ)) (U₂ : SB (Matrix (Q.n₂ ⊕ Unit) Q.n₁ Q.k) →ₗ[ℂ] (WO S.J.W₀ S.B'.V₀ → ℂ)), (∀ Φ Ψ, Ψ.1 = GlobalTheory.LocFlat.flCol Q.ψk Q.ν (σ + 1 + 1) Φ.1 → fla

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdAtKStdOfTwist

theorem **adAt_kb_eq_frsF** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {u : B.Word} {a : I.ι} {Ch₀ : ℕ → Type} (A : TwDat I l Z u a Ch₀) (hA : A.Is) (P : BadPr I a) (hP : P.1 ∉ (B.primes X).vGs) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch₀ i) (r s : Fin (B.R.N i)) : adAt B.host.F P.1 (((A.kb i β : TC.GA B (B.R.N i)) : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) _) r s) = algebraMap B.host.E _ (frsF I.sel (I.tup a) (A.p0 i β).1.1 P (Pi.single s 1) r)

theorem **adAt_kStd_of_twist** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} (hI : I.Ok) (a : I.ι) (P : BadPr I a) (hP : P.1 ∉ (B.primes X).vGs) (u : B.Word) (n : ℕ) (r s : Fin n) : adAt B.host.F P.1 (((TC.kStd X u n : TC.GA B n) : Matrix (Fin n) (Fin n) _) r s) = algebraMap B.host.E _ (if r = s then 1 else 0)

theorem **map_eq_one_of_pr** {D : Formalism.BasicNotions.D7} {k : Type} [CommRing k] {n : ℕ} (φ : Formalism.L2.Theta.AE D →+* k) (M : Matrix (Fin n) (Fin n) (Formalism.L2.Theta.AE D)) (h : ∀ c d, φ (M c d) = φ (Formalism.L2.Theta.incA D (if c = d then 1 else 0))) : M.map φ = 1

theorem **mul_frameγ_std** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k N Y) (e' : Y ≃ Fin N) (hF : F.β = (1 : Matrix (Fin N) (Fin N) k).submatrix id e') (M : Matrix (Fin N) Y k) (r : Fin N) (b : Y) : (M * F.γ) r (e' b) = M r b

theorem **frameβ_mul_std** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k N Y) (e' : Y ≃ Fin N) (hF : F.β = (1 : Matrix (Fin N) (Fin N) k).submatrix id e') (g : Matrix Y Y k) (r b : Y) : (F.β * g) (e' r) b = g r b

theorem **map_eq_sum_frame** {E k : Type} [Field E] [CommRing k] {n : ℕ} (φ : E →+* k) (f : (Fin n → E) ≃ₗ[E] (Fin n → E)) (y : Fin n → E) (x : Fin n) : φ (y x) = ∑ c, φ (f (Pi.single c 1) x) * φ (f.symm y c)

theorem **QJ.Reads.movedAt** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → ∀ t (J : QJ a t), J.Reads → ∀ i, MovedAt J.A (J.P i) (J.E i) (J.hψ i) (J.sg i) (J.π i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdAtToEAfEqZero

theorem **eq_zero_of_le_exp** (x : WithZero (Multiplicative ℤ)) (h : ∀ n : ℕ, x ≤ WithZero.exp (-(n : ℤ))) : x = 0

theorem **afIso_indP** (w : GlobalTheory.FinPl.Pl D) (P : HeightOneSpectrum (𝒪 D.host.E)) (hP : P.under (𝒪 D.host.F) = w) : GlobalTheory.Mem.Col2.afIso D.host.F D.host.E ((1 : D.host.E) ⊗ₜ[D.host.F] GlobalTheory.FinPl.KOpen.indP D (· = w)) P = 1

theorem **indicatorAdele_mul_eq_zero** (w : GlobalTheory.FinPl.Pl D) (z : Af D.host.F ⊗[D.host.F] D.host.E) (hz : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w z = 0) : (GlobalTheory.FinPl.KOpen.indP D (· = w) ⊗ₜ[D.host.F] (1 : D.host.E)) ∗ z = 0

theorem **adAt_toEAf_eq_zero** (w : GlobalTheory.FinPl.Pl D) (a : AE D) (ha : GlobalTheory.FinPl.compAt D w a = 0) (P : HeightOneSpectrum (𝒪 D.host.E)) (hP : P.under (𝒪 D.host.F) = w) : GlobalTheory.Mem.Col2.adAt D.host.F P (toEAf D a) = 0

theorem **adAt_finM_eq_one** : ∀ (D : D7) (V : HSp D) {n : ℕ} (hn : V.rank = n) (Pset : GlobalTheory.FinPl.Pl D → Prop) (h : GA D V), h ∈ GlobalTheory.FinPl.grpAt V Pset → ∀ P : HeightOneSpectrum (𝒪 D.host.E), ¬ Pset (P.under (𝒪 D.host.F)) → ∀ r k, adAt D.host.F P (finM V hn h r k) = (1 : Matrix (Fin n) (Fin n) (P.adicCompletion D.host.E)) r k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdAtToEAfOfIsIdemSum

theorem **read_rel** {A B K E : Type} [CommRing A] [CommRing B] [CommRing K] [CommRing E] {m n : Type} [Fintype m] [Fintype n] (t : A →+* B) (φ : B →+* K) (i : E →+* A) (j : E →+* B) (hij : ∀ x, t (i x) = j x) {e : A} (he : φ (t e) = 1) (X : Matrix m m A) (Z : Matrix n n A) (γ γ₀ : Matrix m n E) (v : Matrix m n A) (k : ℕ) (h : γ₀.map (fun r => e ∗ i r) = X ∗ γ.map (fun r => e ∗ i r) ∗ Z + k • v) (r : m) (s : n) : φ ((X.map t ∗ γ.map j ∗ Z.map t - γ₀.map j) r s) = -((k : K) ∗ φ (t (v r s)))

theorem **adAt_toEAf_of_isIdemAt** {f : AE D} {P : HeightOneSpectrum (𝒪 D.host.E)} (hf : Formalism.L2.NTheta.IsIdemAt f P) : adAt D.host.F P (toEAf D f) = 1

theorem **adAt_toEAf_of_isIdemSum** {e : AE D} {Ps : Finset (HeightOneSpectrum (𝒪 D.host.E))} (he : IsIdemSum e Ps) {P : HeightOneSpectrum (𝒪 D.host.E)} (hP : P ∈ Ps) : adAt D.host.F P (toEAf D e) = 1

theorem **isIntegral_of_eq** {x : D.host.E} (h : ∃ r : 𝒪 D.host.E, algebraMap (𝒪 D.host.E) D.host.E r = x) : IsIntegral ℤ x

theorem **tmul_one_mem_rInt** {x : D.host.E} (hx : IsIntegral ℤ x) : x ⊗ₜ[D.host.F] (1 : FiniteAdeleRing (𝒪 D.host.F) D.host.F) ∈ Formalism.L2.DIV.RInt D.host.F D.host.E

theorem **one_tmul_mem_rInt** {α : FiniteAdeleRing (𝒪 D.host.F) D.host.F} (hα : α ∈ intAf D.host.F) : (1 : D.host.E) ⊗ₜ[D.host.F] α ∈ Formalism.L2.DIV.RInt D.host.F D.host.E

theorem **exists_den_stdHom** (S T : HSp D) : ∃ d : ℕ, 0 < d ∧ ∀ v ∈ stdHom S T, ∀ r s, ((d : EF D) ∗ toEAf D (v r s) : Formalism.L2.DIV.AfE D.host.F D.host.E) ∈ Formalism.L2.DIV.RInt D.host.F D.host.E

theorem **finM_rfl** (V : HSp D) (x : GA D V) : finM V rfl x = (Formalism.L2.Theta.mt V x).map (toEAf D)

theorem **mt_mul'** (V : HSp D) (x y : GA D V) : Formalism.L2.Theta.mt V (x ∗ y) = Formalism.L2.Theta.mt V x ∗ Formalism.L2.Theta.mt V y

theorem **mt_one'** (V : HSp D) : Formalism.L2.Theta.mt V 1 = 1

theorem **inv_val_eq** (V : HSp D) (x : GA D V) (Y : GL (Fin V.rank) (EF D)) (hY : finM V rfl x = Y.1) : (Y⁻¹ : GL (Fin V.rank) (EF D)).1 = (Formalism.L2.Theta.mt V x⁻¹).map (toEAf D)

theorem **cong_at_idem** : ∀ {D : D7} (S T : HSp D) {a b : ℕ} (hS : S.rank = a) (hT : T.rank = b) (e : AE D) (Ps : Finset (HeightOneSpectrum (𝒪 D.host.E))), IsIdemSum e Ps → ∃ d : ℕ, 0 < d ∧ ∀ (g : GA D S) (h : GA D T) (Y : GL (Fin a) (EF D)) (Y' : GL (Fin b) (EF D)), finM S hS g = Y.1 → finM T hT h = Y'.1 → ∀ (M : ℕ) (γ γ₀ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (v : WO S T), v ∈ stdHom S T → γ₀.map (fun r => e ∗ incA D r) = actO S T g⁻¹ h⁻¹ (γ.map fun r => e ∗ incA D r) + (d ∗ M) • v → ∀ P ∈ Ps, ∀ r s, Valued.v (adAt D.host.F P ((HMul.hMul (HMul.hMul ((Y'⁻¹ : GL (Fin b) (EF D)).1 : Matrix (Fin b) (Fin b) (EF D)) (((LinearMap.toMatrix' (toHom S T hS hT γ)).map (inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AddCharConductorIntegersContinuousNeOne

theorem **hasConductor_one_continuous_ne_one** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ψ : AddChar k Circle), HasConductor ψ 1 → Continuous ψ ∧ ψ ≠ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AddCharConductorIntegersContinuousNeOneStatement

theorem **_root_.GlobalTheory.T.addChar_conductor_integers_continuous_ne_one** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ψ : AddChar k Circle), HasConductor ψ 1 → Continuous ψ ∧ ψ ≠ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AddCharDotProductTrivialOnTsupportNhds

theorem **nonarch** : NonarchimedeanAddGroup (n → k) where

theorem **measureReal_box_ne** (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (s : kˣ) : (μ.real (box n s) : ℂ) ≠ 0

theorem **decomp_integral** (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (s : kˣ) {φ : (n → k) → ℂ} (hφ : φ ∈ SB (n → k)) (hinv : ∀ t ∈ box n s, ∀ x, φ (x + t) = φ x) : ∃ S : Finset (n → k), (∀ y, φ y = ∑ x ∈ S, φ x ∗ (box n s).indicator (fun _ => (1:ℂ)) (y - x)) ∧ ∀ F : (n → k) → ℂ, (∀ t ∈ box n s, ∀ x, F (x + t) = F x) → ∫ y, φ y ∗ F y ∂μ = (μ.real (box n s) : ℂ) ∗ ∑ x ∈ S, φ x ∗ F x

theorem **addChar_dotProduct_trivial_on_tsupport_nhds** (hχ : HasConductor χ c) {g : (n → k) → ℂ} (hg : g ∈ SB (n → k)) : {t : n → k | ∀ η ∈ tsupport g, χ (η ·ᵥ t) = 1} ∈ 𝒩 (0 : n → k)

theorem **box_nhds** (s : kˣ) : box n s ∈ 𝒩 (0 : n → k)

theorem **invariant_nhds** {f : (n → k) → ℂ} (hf : f ∈ SB (n → k)) : {t : n → k | ∀ x, f (x + t) = f x} ∈ 𝒩 (0 : n → k)

theorem **tr_mem** {a : (n → k) → ℂ} (ha : a ∈ SB (n → k)) (x : n → k) : (fun y => a (y - x)) ∈ SB (n → k)

theorem **mc_mem** (hχ : HasConductor χ c) {b : (n → k) → ℂ} (hb : b ∈ SB (n → k)) (x : n → k) : (fun η => (χ (η ·ᵥ x) : ℂ) ∗ b η) ∈ SB (n → k)

theorem **law_integrated** (hχ : HasConductor χ c) (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (B : SB (n → k) →ₗ[ℂ] SB (n → k) →ₗ[ℂ] ℂ) (h2 : ∀ (a a' b b' : SB (n → k)) (t : n → k), (∀ y, a'.1 y = a.1 (y - t)) → (∀ η, b'.1 η = (χ (η ·ᵥ t) : ℂ) ∗ b.1 η) → B a' b = B a b') (a b φ a' b' : SB (n → k)) (ha' : ∀ y, a'.1 y = ∫ t, φ.1 t ∗ a.1 (y - t) ∂μ) (hb' : ∀ η, b'.1 η = (∫ t, φ.1 t ∗ (χ (η ·ᵥ t) : ℂ) ∂μ) ∗ b.1 η) : B a' b = B a b'

theorem **box_mono** {s s' : kˣ} (h : valuation k (s : k) ≤ valuation k (s' : k)) : box n s ⊆ box n s'

theorem **box_total** (s s' : kˣ) : box n s ⊆ box n s' ∨ box n s' ⊆ box n s

theorem **ind_add** (s : kˣ) {t : n → k} (ht : t ∈ box n s) (z : n → k) : (box n s).indicator (fun _ => (1 : ℂ)) (z + t) = (box n s).indicator (fun _ => (1 : ℂ)) z

theorem **ind_mem** (s : kˣ) : (box n s).indicator (fun _ => (1 : ℂ)) ∈ SB (n → k)

theorem **conv_ind** (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (s : kˣ) {φ : (n → k) → ℂ} (hφ : φ ∈ SB (n → k)) (hinv : ∀ t ∈ box n s, ∀ x, φ (x + t) = φ x) (y : n → k) : ∫ t, φ t ∗ (box n s).indicator (fun _ => (1 : ℂ)) (y - t) ∂μ = (μ.real (box n s) : ℂ) ∗ φ y

theorem **isAdmissibleBoxForCharacter_of_subset** {s : kˣ} {g : (n → k) → ℂ} (h : box n s ⊆ {t : n → k | ∀ η ∈ tsupport g, χ (η ·ᵥ t) = 1}) : admitted χ s g

theorem **admitted.mono** {s s' : kˣ} {g : (n → k) → ℂ} (h : admitted χ s g) (hs : box n s' ⊆ box n s) : admitted χ s' g

theorem **ratio_indep** (hχ : HasConductor χ c) (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (B : SB (n → k) →ₗ[ℂ] SB (n → k) →ₗ[ℂ] ℂ) (h2 : ∀ (a a' b b' : SB (n → k)) (t : n → k), (∀ y, a'.1 y = a.1 (y − t)) → (∀ η, b'.1 η = (χ (η ·ᵥ t) : ℂ) * b.1 η) → B a' b = B a b') {s s' : kˣ} (g : SB (n → k)) (hs : admitted χ s g.1) (hss : box n s' ⊆ box n s) : (μ.real (box n s) : ℂ)⁻¹ * B (ind s) g = (μ.real (box n s') : ℂ)⁻¹ * B (ind s') g

theorem **exists_adm** (hχ : HasConductor χ c) (g : SB (n → k)) : ∃ s : kˣ, admitted χ s g.1

theorem **ell_eq** {s : kˣ} (g : SB (n → k)) (hs : admitted χ s g.1) : ell χ μ B g = (μ.real (box n s) : ℂ)⁻¹ * B (ind s) g

theorem **ell_translate** (h1 : ∀ (a a' b b' : SB (n → k)) (t : n → k), (∀ y, a'.1 y = (χ (t ·ᵥ y) : ℂ) * a.1 y) → (∀ η, b'.1 η = b.1 (η − t)) → B a' b = B a b') (g : SB (n → k)) (u : n → k) (hg' : (fun x => g.1 (x + u)) ∈ SB (n → k)) : ell χ μ B (_, hg') = ell χ μ B g

theorem **form_eq_ell** (a b : SB (n → k)) (hm : (fun η => (∫ y, a.1 y * (χ (η ·ᵥ y) : ℂ) ∂μ) * b.1 η) ∈ SB (n → k)) : B a b = ell χ μ B (_, hm)

theorem **heis_invariant_form** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] (χ : AddChar k Circle), Continuous χ → χ ≠ 1 → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (B : SB (n → k) →ₗ[ℂ] SB (n → k) →ₗ[ℂ] ℂ), (∀ (a a' b b' : SB (n → k)) (t : n → k), (∀ y, a'.1 y = (χ (t ·ᵥ y) : ℂ) * a.1 y) → (∀ η, b'.1 η = b.1 (η − t)) → B a' b = B a b') → (∀ (a a' b b' : SB (n → k)) (t : n → k), (∀ y, a'.1 y = a.1 (y − t)) → (∀ η, b'.1 η = (χ (η ·ᵥ t) : ℂ) * b.1 η) → B a' b = B a b') → ∃ c : ℂ, ∀ a b : SB (n → k), B a b = c * ∫ η, (∫ y, a.1 y * (χ (η ·ᵥ y) : ℂ) ∂μ) * b.1 η ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AddCharSymplecticFinitePartRestEqOne

theorem **finPart_finRat_sub** {S T : HSp D} (M : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (i : Fin T.rank) (j : Fin S.rank) : finPart D (GlobalTheory.VacUnique.finRat S T M i j − incA D (M i j)) = 0

theorem **infPart_of_idemSum** {e : AE D} {SS : Finset (Formalism.L2.DIV.Pl D.host.E)} (hI : IsIdemSum e SS) : infPart D e = 0

theorem **smul_rest_eq_zero** {S T : HSp D} {e : AE D} (hee : e * e = e) (hinf : infPart D e = 0) (M : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : e • (GlobalTheory.VacUnique.finRat S T M − M.map (fun r => e * incA D r)) = 0

theorem **char_sympl_rest_eq_one** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ∀ y' ∈ halfBO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ψ (symplO S T ((GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) − ((LinearMap.toMatrix' x').map fun r => e * incA D r)) (GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' y'))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AddCharSymplecticFinitePartRestEqOneStatement

theorem **_root_.GlobalTheory.T.addChar_symplectic_finitePartRest_eq_one** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ∀ y' ∈ halfBO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ψ (symplO S T ((GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) − ((LinearMap.toMatrix' x').map fun r => e * incA D r)) (GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' y'))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AddSubgroupFiniteQuotOfCommensurable

theorem **addSubgroup_finite_quot_of_commensurable** : ∀ (V : Type) [AddCommGroup V] (A L : AddSubgroup V) (N M : ℕ), 0 < N → 0 < M → (∀ v : V, N • v = 0 → v = 0) → (∀ a ∈ A, N • a ∈ L) → (∀ l ∈ L, M • l ∈ A) → (∀ j : ℕ, 0 < j → ∃ R : Finset V, ∀ u ∈ L, ∃ r ∈ R, ∃ v ∈ L, u = r + j • v) → ∀ k : ℕ, 0 < k → ∃ R : Finset V, ∀ u ∈ A, ∃ r ∈ R, ∃ v ∈ A, u = r + k • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdditiveCharacterCongruenceInvariance

theorem **psi_unif_mul_congr** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {s κ : ℕ} (S : GlobalTheory.SymChain.SymData k s κ) (d e : k) (hv : valuation k (d − e) < 1) : (S.ψ (((S.ϖ⁻¹ : kˣ) : k) * e) : ℂ) = (S.ψ (((S.ϖ⁻¹ : kˣ) : k) * d) : ℂ)

theorem **colLam_eq_psi_superSum** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → a.True → ∀ row col, (a.fam row col).c.Is → ∀ κ ∈ GfC ((I.rec7 l Z).sp.X 0), ∀ y : Formalism.UG.UPts (Formalism.L2.Forms.sigD (I.rec7 l Z)) (Formalism.L2.Forms.gramN (I.rec7 l Z) 0 (Nat.zero_le _)) (FiniteAdeleRing (𝒪 (I.rec7 l Z).host.F) (I.rec7 l Z).host.F), GlobalTheory.AdForm.finM ((I.rec7 l Z).sp.X 0) (Formalism.L2.Forms.rankN (I.rec7 l Z) 0 (Nat.zero_le _)) κ = y.1.1 → (y : GL _ _) ∈ (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdditiveCharacterCongruenceInvarianceStatement

theorem **_root_.GlobalTheory.T.true_geometry_final_volume** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → a.True → ∀ row col, (a.fam row col).c.Is → ∀ κ ∈ GfC ((I.rec7 l Z).sp.X 0), ∀ y : Formalism.UG.UPts (Formalism.L2.Forms.sigD (I.rec7 l Z)) (Formalism.L2.Forms.gramN (I.rec7 l Z) 0 (Nat.zero_le _)) (FiniteAdeleRing (𝒪 (I.rec7 l Z).host.F) (I.rec7 l Z).host.F), GlobalTheory.AdForm.finM ((I.rec7 l Z).sp.X 0) (Formalism.L2.Forms.rankN (I.rec7 l Z) 0 (Nat.zero_le _)) κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdditiveCharacterOneNe

theorem **isOrd_ordX** (R : Ranks) (j : ℕ) : IsOrd xPos (ordX R j)

theorem **intAt_nat** (n : ℕ) : IntAt P (n : E)

theorem **reduction_natCast** (n : ℕ) : redP P (n : E) = (n : 𝒪 E ⧸ P.asIdeal)

theorem **reduction_natCast_eq_zero** (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) : redP P (q : E) = 0

theorem **pow_red** (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (n : ℕ) : IntAt P ((q : E) ^ (n + 1)) ∧ redP P ((q : E) ^ (n + 1)) = 0

theorem **valuation_one₂** (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (hq0 : (q : E) ≠ 0) : LocE.vP P q (1 : E) = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdeleComponentAtFinitePlaceIntegralMemLocalTenso

theorem **compAt_mem_intV_of_intAt** : ∀ (D : D7) (v : Pl D) (a : AE D), a ∈ intAt D (· = v) → compAt D v a ∈ intV D v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdeleComponentAtFinitePlaceIntegralMemLocalTensoStatement

theorem **_root_.GlobalTheory.T.adeleComponentAtFinitePlace_integral_mem_localTensor** : ∀ (D : D7) (v : Pl D) (a : AE D), a ∈ intAt D (· = v) → compAt D v a ∈ intV D v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicGLCircleCharacterTrivialOnDetOne

theorem **glChar_transvection** {R : Type} [CommRing R] (h2 : IsUnit (2 : R)) {n : ℕ} (α : GL (Fin n) R →* Circle) (i j : Fin n) (hij : i ≠ j) (t : R) (u : GL (Fin n) R) (hu : u.1 = Matrix.transvection i j t) : α u = 1

theorem **glChar_eq_one_of_det_eq_one** : ∀ (D : D7) (n : ℕ) (α : GL (Fin n) (AE D) →* Circle), Continuous α → ∀ a, Matrix.GeneralLinearGroup.det a = 1 → α a = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicGLCircleCharacterTrivialOnDetOneStatement

theorem **_root_.GlobalTheory.T.adelic_GL_circle_character_trivial_on_det_one** : ∀ (D : D7) (n : ℕ) (α : GL (Fin n) (AE D) →* Circle), Continuous α → ∀ a, Matrix.GeneralLinearGroup.det a = 1 → α a = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicHatMulOfMulClosed

theorem **adelic_hat_mul_of_mul_closed** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (J : Set E) (hJ : ∀ x ∈ J, ∀ y ∈ J, x * y ∈ J) {z w : GlobalTheory.AdLat.RA F E} (hz : z ∈ GlobalTheory.AdLat.hat1 F J) (hw : w ∈ GlobalTheory.AdLat.hat1 F J) : z * w ∈ GlobalTheory.AdLat.hat1 F J

theorem **adelic_hat_one_mul_mem** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {z w : GlobalTheory.AdLat.RA F E} (hz : z ∈ GlobalTheory.AdLat.hat1 F (↑(1 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) : Set E)) (hw : w ∈ GlobalTheory.AdLat.hat1 F (↑(1 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) : Set E)) : z * w ∈ GlobalTheory.AdLat.hat1 F (↑(1 : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) : Set E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicIwasawaDecompositionWittIsotropicStabilize

theorem **exists_stab_proj_eq** {D : D7} {T T₀ : HSp D} {r : ℕ} (d : HypDec T T₀ r) (x : ∀ v : FinPl D, Gv T v) (hx : ∀ v, emb D.sp.cm.sig T.gram v (x v) ∈ stab d) (S : Finset (FinPl D)) : ∃ p ∈ stab d, (∀ v ∈ S, proj D.sp.cm.sig T.gram v p = x v) ∧ (∀ v ∉ S, proj D.sp.cm.sig T.gram v p = 1) ∧ ∀ w, projArch D.sp.cm.sig T.gram w p = 1

theorem **hasIw_Kmax** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (b : GL (Fin T.rank) (AE D)), (∀ v : FinPl D, (2 : 𝒪 D.host.F) ∈ v.asIdeal → Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicIwasawaDecompositionWittIsotropicStabilizeStatement

theorem **_root_.GlobalTheory.T.adelic_iwasawa_decomposition_wittIsotropicStabilizer** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (b : GL (Fin T.rank) (AE D)), (∀ v : FinPl D, (2 : 𝒪 D.host.F) ∈ v.asIdeal → Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicIwasawaDoubledGroupSelfDualLattice

theorem **hasIwasawa_KAd_selfDual** : ∀ (D : D7) (A : HSp D) (L : Lat A) (a : GL (Fin A.rank) (AE D)), IsSelfDualC A L → IsMaj A a → HasIwasawa D.sp.cm.sig (dblSp A).gram (KAd D.sp.cm.sig (dblSp A).gram L a)

theorem **hasIwasawa_Khat** (D : D7) (A : HSp D) (a : GL (Fin A.rank) (AE D)) (ha : IsMaj A a) : HasIwasawa D.sp.cm.sig (dblSp A).gram (Khat A a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicIwasawaDoubledGroupSelfDualLatticeStatement

theorem **_root_.GlobalTheory.T.adelicIwasawa_doubledGroup_selfDualLattice** : ∀ (D : D7) (A : HSp D) (L : Lat A) (a : GL (Fin A.rank) (AE D)), IsSelfDualC A L → IsMaj A a → HasIwasawa D.sp.cm.sig (dblSp A).gram (KAd D.sp.cm.sig (dblSp A).gram L a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixQuotientPeriodicIntegralContinuousPa

theorem **eventually_close** {P : Type} [TopologicalSpace P] (K : Mx D m n → P → ℂ) (hc : Continuous (Function.uncurry K)) (hp : ∀ γ ∈ ratMx D m n, ∀ x p, K (γ + x) p = K x p) (p₀ : P) {ε : ℝ} (hε : 0 < ε) : ∀ᶠ p in nhds p₀, ∀ x, ‖K x p − K x p₀‖ ≤ ε

theorem **isCovolOne_map** {μ : Measure (Mx D m n)} {𝓕 : Set (Mx D m n)} (h : IsCovolOne (ratMx D m n) μ 𝓕) (e : Mx D m n ≃+ Mx D m n) (he : Continuous e) (he' : Continuous e.symm) (hr : ∀ x, e x ∈ ratMx D m n ↔ x ∈ ratMx D m n) : IsCovolOne (ratMx D m n) (μ.map e) (e '' 𝓕) ∧ MeasurableEmbedding e

theorem **parCont** : ∀ (D : D7) (m n : ℕ) (v : VolM D m n) (P : Type) [TopologicalSpace P] (K : Mx D m n → P → ℂ), Continuous (Function.uncurry K) → (∀ γ ∈ ratMx D m n, ∀ x p, K (γ + x) p = K x p) → (∀ p, IntegrableOn (fun x => K x p) v.𝓕 v.μ) ∧ (Continuous fun p => ∫ x in v.𝓕, K x p ∂v.μ) ∧ (∀ p₀, ∃ C : ℝ, ∀ᶠ p in nhds p₀, ∀ x, ‖K x p‖ ≤ C) ∧ ∀ (e : Mx D m n ≃+ Mx D m n), Continuous e → Continuous e.symm → (∀ x, e x ∈ ratMx D m n ↔ x ∈ ratMx D m n) → ∀ p, ∫ x in v.𝓕, K (e x) p ∂v.μ = ∫ x in v.𝓕, K x p ∂v.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixQuotientPeriodicIntegralContinuousPaStatement

theorem **_root_.GlobalTheory.T.adelicMatrixQuotient_periodicIntegral_continuous_param** : ∀ (D : D7) (m n : ℕ) (v : VolM D m n) (P : Type) [TopologicalSpace P] (K : Mx D m n → P → ℂ), Continuous (Function.uncurry K) → (∀ γ ∈ ratMx D m n, ∀ x p, K (γ + x) p = K x p) → (∀ p, IntegrableOn (fun x => K x p) v.𝓕 v.μ) ∧ (Continuous fun p => ∫ x in v.𝓕, K x p ∂v.μ) ∧ (∀ p₀, ∃ C : ℝ, ∀ᶠ p in nhds p₀, ∀ x, ‖K x p‖ ≤ C) ∧ ∀ (e : Mx D m n ≃+ Mx D m n), Continuous e → Continuous e.symm → (∀ x, e x ∈ ratMx D m n ↔ x ∈ ratMx D m n) → ∀ p, ∫ x in v.𝓕, K (e x) p ∂v.μ = ∫ x in v.𝓕, K x p ∂v.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixRealSizeBoundsAlongRealParametrisati

theorem **compAt_add** (θ : InfFam D) (τ : D.what₂.Tau) (x y : Mx D m n) : compAt θ τ (x + y) = compAt θ τ x + compAt θ τ y

theorem **compAt_neg** (θ : InfFam D) (τ : D.what₂.Tau) (x : Mx D m n) : compAt θ τ (−x) = −compAt θ τ x

theorem **norm_compAt_le** (θ : InfFam D) (x : Mx D m n) (τ : D.what₂.Tau) (i : Fin m) (j : Fin n) : ‖compAt θ τ x i j‖ ≤ GlobalTheory.Unf.realSize θ x

theorem **norm_vec_le** (θ : InfFam D) (x : Mx D m n) : ‖(fun τ i j => compAt θ τ x i j : D.what₂.Tau → Fin m → Fin n → ℂ)‖ ≤ GlobalTheory.Unf.realSize θ x

theorem **realSize_le** (θ : InfFam D) (x : Mx D m n) : GlobalTheory.Unf.realSize θ x ≤ (Fintype.card D.what₂.Tau * (m * n) : ℝ) * ‖(fun τ i j => compAt θ τ x i j : D.what₂.Tau → Fin m → Fin n → ℂ)‖

theorem **realSize_add_le** (θ : InfFam D) (x y : Mx D m n) : GlobalTheory.Unf.realSize θ (x + y) ≤ GlobalTheory.Unf.realSize θ x + GlobalTheory.Unf.realSize θ y

theorem **realSize_neg** (θ : InfFam D) (x : Mx D m n) : GlobalTheory.Unf.realSize θ (-x) = GlobalTheory.Unf.realSize θ x

theorem **continuous_realVec** {θ : InfFam D} (hθ : IsInfFam θ) {k : ℕ} {L : (Fin k → ℝ) →+ Mx D m n} (hL : Continuous L) : Continuous (realVec θ L)

theorem **GlobalTheory.Unf.realParam_size_bounds** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n k : ℕ) (L : (Fin k → ℝ) →+ Mx D m n), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsRealParam L → (∃ C : ℝ, ∀ t, GlobalTheory.Unf.realSize θ (L t) ≤ C * ‖t‖) ∧ (Function.Injective L → ∃ C : ℝ, ∀ (x : Mx D m n) t, ‖t‖ ≤ C * (GlobalTheory.Unf.realSize θ (x + L t) + GlobalTheory.Unf.realSize θ x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixRealSizeBoundsAlongRealParametrisatiStatement

theorem **_root_.GlobalTheory.T.adelicMatrixRealSize_bounds_along_real_parametrisation** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n k : ℕ) (L : (Fin k → ℝ) →+ Mx D m n), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsRealParam L → (∃ C : ℝ, ∀ t, GlobalTheory.Unf.realSize θ (L t) ≤ C * ‖t‖) ∧ (Function.Injective L → ∃ C : ℝ, ∀ (x : Mx D m n) t, ‖t‖ ≤ C * (GlobalTheory.Unf.realSize θ (x + L t) + GlobalTheory.Unf.realSize θ x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixRealSizeLeRealSymplecticPairingSum

theorem **trace_conjT_mul_single** {r s : ℕ} (X : Matrix (Fin r) (Fin s) ℂ) (i : Fin r) (j : Fin s) (a : ℂ) : (Xᴴ * Matrix.single i j a).trace = star (X i j) * a

theorem **gramAt_det_ne** (D : D7) (T : HSp D) (τ : D.what₂.Tau) : (gramAt D T τ).det ≠ 0

theorem **pair_single** (D : D7) (T : HSp D) {ρ : ℕ} (τ : D.what₂.Tau) (X : Matrix (Fin T.rank) (Fin ρ) ℂ) (i : Fin T.rank) (j : Fin ρ) (a : ℂ) : tauE D τ D.host.xi0 * (Xᴴ * gramAt D T τ * ((gramAt D T τ)⁻¹ * Matrix.single i j a)).trace = tauE D τ D.host.xi0 * (star (X i j) * a)

theorem **realSize_le_pairing** : ∀ (D : D7) (θ : InfFam D) (T : HSp D) (ρ : ℕ), ∃ (k : ℕ) (τs : Fin k → D.what₂.Tau) (zs : Fin k → Matrix (Fin T.rank) (Fin ρ) ℂ) (C : ℝ), ∀ x : Mx D T.rank ρ, realSize θ x ≤ C * ∑ i, |2 * (tauE D (τs i) D.host.xi0 * ((compAt θ (τs i) x)ᴴ * gramAt D T (τs i) * zs i).trace).re|

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixRealSizeLeRealSymplecticPairingSumStatement

theorem **_root_.GlobalTheory.T.adelicMatrixRealSize_le_realSymplecticPairing_sum** : ∀ (D : D7) (θ : InfFam D) (T : HSp D) (ρ : ℕ), ∃ (k : ℕ) (τs : Fin k → D.what₂.Tau) (zs : Fin k → Matrix (Fin T.rank) (Fin ρ) ℂ) (C : ℝ), ∀ x : Mx D T.rank ρ, realSize θ x ≤ C * ∑ i, |2 * (tauE D (τs i) D.host.xi0 * ((compAt θ (τs i) x)ᴴ * gramAt D T (τs i) * zs i).trace).re|

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixSchwartzMajorantMeasurableRationalSh

theorem **norm_compAt_le_realSize** (θ : InfFam D) {m n : ℕ} (x : Mx D m n) (τ : D.what₂.Tau) (i : Fin m) (j : Fin n) : ‖compAt θ τ x i j‖ ≤ realSize θ x

theorem **continuous_realSize** {θ : InfFam D} (hθ : IsInfFam θ) {m n : ℕ} : Continuous (realSize θ : Mx D m n → ℝ)

theorem **finPart_eq_zero_iff** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : GlobalTheory.FockCompare.Chart θ σ) (a : AE D) : finPart D a = 0 ↔ a = σ (fun τ => θ τ (infPart D a))

theorem **isClosed_finZero** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : GlobalTheory.FockCompare.Chart θ σ) (m n : ℕ) : IsClosed {z : Mx D m n | z.map (finPart D) = 0}

theorem **setOf_inBox** {m n : ℕ} (K : Set (Mx D m n)) : {x | InBox K x} = K + {z : Mx D m n | z.map (finPart D) = 0}

theorem **isClosed_inBox** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : GlobalTheory.FockCompare.Chart θ σ) {m n : ℕ} (K : Set (Mx D m n)) (hK : IsCompact K) : IsClosed {x | InBox K x}

theorem **isCompact_box** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : GlobalTheory.FockCompare.Chart θ σ) {m n : ℕ} (K : Set (Mx D m n)) (hK : IsCompact K) (R : ℝ) : IsCompact {x | InBox K x ∧ realSize θ x ≤ R}

theorem **measurable_maj** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : GlobalTheory.FockCompare.Chart θ σ) {m n : ℕ} (K : Set (Mx D m n)) (hK : IsCompact K) (N : ℕ) : Measurable (maj θ K N)

theorem **exists_leftInv** {m n : ℕ} (x : Matrix (Fin m) (Fin n) D.host.E) (hx : LinearIndependent D.host.E x.transpose) : ∃ y : Matrix (Fin n) (Fin m) D.host.E, y * x = 1

theorem **compAt_mul** (θ : InfFam D) (τ : D.what₂.Tau) {m n k : ℕ} (A : Mx D m n) (B : Mx D n k) : compAt θ τ (A * B) = compAt θ τ A * compAt θ τ B

theorem **realSize_mul_le** (θ : InfFam D) {m n k : ℕ} (Y : Mx D m n) : ∃ c : ℝ, 0 ≤ c ∧ ∀ a : Mx D n k, realSize θ (Y * a) ≤ c * realSize θ a

theorem **maj_rat_le** (θ : InfFam D) (m n k : ℕ) (x : Matrix (Fin m) (Fin n) D.host.E) (hx : LinearIndependent D.host.E x.transpose) (K : Set (Mx D m k)) (hK : IsCompact K) : ∃ K' : Set (Mx D n k), IsCompact K' ∧ ∀ N : ℕ, ∃ C : ℝ, ∀ u : Mx D n k, maj θ K N ((rat x : Mx D m n) * u) ≤ C * maj θ K' N u

theorem **GlobalTheory.Unf.maj_measurable_rat** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → (∀ (m n : ℕ) (K : Set (Mx D m n)), IsCompact K → IsClosed {x | InBox K x}) ∧ Continuous (realSize θ : Mx D m n → ℝ) ∧ (∀ N, Measurable (maj θ K N)) ∧ ∀ R : ℝ, IsCompact {x | InBox K x ∧ realSize θ x ≤ R}) ∧ ∀ (m n k : ℕ) (x : Matrix (Fin m) (Fin n) D.host.E), LinearIndependent D.host.E x.transpose → ∀ K : Set (Mx D m k), IsCompact K → ∃ K' : Set (Mx D n k), IsCompact K' ∧ ∀ N : ℕ, ∃ C : ℝ, ∀ u : Mx D n k, maj θ K N ((rat x : Mx D m n) · u) ≤ C * maj θ K' N u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicMatrixSchwartzMajorantMeasurableRationalShStatement

theorem **_root_.GlobalTheory.T.adelicMatrixSchwartzMajorant_measurable_rationalShift** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → (∀ (m n : ℕ) (K : Set (Mx D m n)), IsCompact K → IsClosed {x | InBox K x} ∧ Continuous (realSize θ : Mx D m n → ℝ) ∧ (∀ N, Measurable (maj θ K N)) ∧ ∀ R : ℝ, IsCompact {x | InBox K x ∧ realSize θ x ≤ R}) ∧ ∀ (m n k : ℕ) (x : Matrix (Fin m) (Fin n) D.host.E), LinearIndependent D.host.E x.transpose → ∀ K : Set (Mx D m k), IsCompact K → ∃ K' : Set (Mx D n k), IsCompact K' ∧ ∀ N : ℕ, ∃ C : ℝ, ∀ u : Mx D n k, maj θ K N ((rat x : Mx D m n) · u) ≤ C * maj θ K' N u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicPointComponentsAtPlacesLatticesVacua

theorem **repr_map** {F E A B ι : Type} [Field F] [Field E] [Algebra F E] [CommRing A] [Algebra F A] [CommRing B] [Algebra F B] (b : Module.Basis ι F E) (f : A →ₐ[F] B) (x : A ⊗[F] E) (i : ι) : (Algebra.TensorProduct.basis B b).repr (Algebra.TensorProduct.map f (AlgHom.id F E) x) i = f ((Algebra.TensorProduct.basis A b).repr x i)

theorem **evalV_ePl** (P : Pl D → Prop) (v : Pl D) : evalV D.host.F v (ePl D P) = if P v then 1 else 0

theorem **eq_zero_of_comp** (x : AE D) (h0 : infPart D x = 0) (hw : ∀ w, compAt D w x = 0) : x = 0

theorem **ePl_int** (P : Pl D → Prop) : ePl D P ∈ intAf D.host.F

theorem **infPart_pair** (α : Af D.host.F) (e : D.host.E) : infPart D (pairA D.host.F 0 α ⊗ₜ[D.host.F] e) = 0

theorem **finPart_pair** (α : Af D.host.F) (e : D.host.E) : finPart D (pairA D.host.F 0 α ⊗ₜ[D.host.F] e) = α ⊗ₜ[D.host.F] e

theorem **compAt_pair** (w : Pl D) (α : Af D.host.F) (e : D.host.E) : compAt D w (pairA D.host.F 0 α ⊗ₜ[D.host.F] e) = evalV D.host.F w α ⊗ₜ[D.host.F] e

theorem **compAt_idem** (P : Pl D → Prop) (w : Pl D) : compAt D w (idem D P) = if P w then 1 else 0

theorem **idem_mem_idAt** (P : Pl D → Prop) : idem D P ∈ idAt D P

theorem **mem_idAt_iff** (P : Pl D → Prop) (a : AE D) : a ∈ idAt D P ↔ idem D P * a = a

theorem **mem_idOff_iff** (P : Pl D → Prop) (a : AE D) : a ∈ idOff D P ↔ idem D P * a = 0

theorem **idem_mem_intAt** (P : Pl D → Prop) : idem D P ∈ intAt D P

theorem **idem_mul_self** (P : Pl D → Prop) : idem D P * idem D P = idem D P

theorem **cA_idem** (P : Pl D → Prop) : cA D (idem D P) = idem D P

theorem **part_idem** (P : Pl D → Prop) : ∃ ε : AE D, ε * ε = ε ∧ cA D ε = ε ∧ ε ∈ intAt D P ∧ (∀ a, a ∈ idAt D P ↔ ε * a = a) ∧ (∀ a, a ∈ idOff D P ↔ ε * a = 0)

theorem **mix_mul** {e : A} (he : e * e = e) (X Y X' Y' : Matrix n n A) : mix e X Y * mix e X' Y' = mix e (X * X') (Y * Y')

theorem **mix_self** (e : A) (X : Matrix n n A) : mix e X X = X

theorem **cT_mix** (σ : A →+* A) {e : A} (hσ : σ e = e) (X Y : Matrix n n A) : Formalism.UG.cT σ (mix e X Y) = mix e (Formalism.UG.cT σ X) (Formalism.UG.cT σ Y)

theorem **mixGL_mem** (σ : A →+* A) (J : Matrix n n A) {e : A} (he : e * e = e) (hσ : σ e = e) {g h : GL n A} (hg : g ∈ Formalism.UG.UJ σ J) (hh : h ∈ Formalism.UG.UJ σ J) : mixGL he g h ∈ Formalism.UG.UJ σ J

theorem **mem_congGrp** {V : HSp D} (I : Ideal (AE D)) (x : GA D V) : x ∈ congGrp V I ↔ ∀ i j, mt V x i j − (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D)) i j ∈ I

theorem **mt_mul** (V : HSp D) (a b : GA D V) : mt V (a * b) = mt V a * mt V b

theorem **mt_inj** {V : HSp D} {a b : GA D V} (h : mt V a = mt V b) : a = b

theorem **mt_mixGA** (P : Pl D → Prop) {V : HSp D} (g h : GA D V) : mt V (mixGA P g h) = mix (idem D P) (mt V g) (mt V h)

theorem **mt_one** (V : HSp D) : mt V 1 = 1

theorem **mem_grpAt_iff** (P : Pl D → Prop) {V : HSp D} (x : GA D V) : x ∈ grpAt V P ↔ mix (idem D P) (mt V x) 1 = mt V x

theorem **mem_grpOff_iff** (P : Pl D → Prop) {V : HSp D} (x : GA D V) : x ∈ grpOff V P ↔ mix (idem D P) 1 (mt V x) = mt V x

theorem **mix_mix_left** {e : AE D} (he : e * e = e) {n : Type} [Fintype n] [DecidableEq n] (X Y Z : Matrix n n (AE D)) : mix e (mix e X Y) Z = mix e X Z

theorem **mix_mix_right** {e : AE D} (he : e * e = e) {n : Type} [Fintype n] [DecidableEq n] (X Y Z : Matrix n n (AE D)) : mix e X (mix e Y Z) = mix e X Z

theorem **part_grp** (P : Pl D → Prop) (S : HSp D) : (∀ h : GA D S, ∃ x ∈ grpAt S P, ∃ y ∈ grpOff S P, h = x * y) ∧ (∀ x ∈ grpAt S P, ∀ y ∈ grpOff S P, x * y = y * x) ∧ grpAt S P ⊓ grpOff S P = ⊥

theorem **pairA_add** (a b : Af D.host.F) : pairA D.host.F 0 (a + b) = pairA D.host.F 0 a + pairA D.host.F 0 b

theorem **pairA_neg** (a : Af D.host.F) : pairA D.host.F 0 (−a) = −pairA D.host.F 0 a

theorem **pairA_zero** : pairA D.host.F 0 (0 : Af D.host.F) = 0

theorem **pairA_mul** (a b : Af D.host.F) : pairA D.host.F 0 (a * b) = pairA D.host.F 0 a * pairA D.host.F 0 b

theorem **pairA_zsmul** (c : ℤ) (a : Af D.host.F) : pairA D.host.F 0 (c • a) = c • pairA D.host.F 0 a

theorem **pair_mem_idAt** (P : Pl D → Prop) (α : Af D.host.F) (hα : ∀ v, ¬ P v → evalV D.host.F v α = 0) (e : D.host.E) : pairA D.host.F 0 α ⊗ₜ[D.host.F] e ∈ idAt D P

theorem **latV_idAt** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : ∀ i, y i ∈ idAt D P

theorem **exists_gens** {S : HSp D} (L : Lat S) (hL : L.FG) : ∃ (t : ℕ) (g : Fin t → Fin S.rank → D.host.E), (∀ k, g k ∈ L) ∧ ∀ x ∈ L, ∃ c : Fin t → ℤ, x = ∑ k, c k • g k

theorem **intAf_zsmul** (c : ℤ) {α : Af D.host.F} (h : α ∈ intAf D.host.F) : c • α ∈ intAf D.host.F

theorem **latV_repr** {S : HSp D} {L : Lat S} {Q : Pl D → Prop} {t : ℕ} (g : Fin t → Fin S.rank → D.host.E) (hgen : ∀ x ∈ L, ∃ c : Fin t → ℤ, x = ∑ k, c k • g k) {z : Fin S.rank → AE D} (hz : z ∈ latV S L Q) : ∃ β : Fin t → Af D.host.F, (∀ k, β k ∈ intAf D.host.F ∧ ∀ w, ¬ Q w → evalV D.host.F w (β k) = 0) ∧ z = fun i => ∑ k, pairA D.host.F 0 (β k) ⊗ₜ[D.host.F] g k i

theorem **finIn_tmul** (a : Af D.host.F) (e : D.host.E) : finIn D (a ⊗ₜ[D.host.F] e) = pairA D.host.F 0 a ⊗ₜ[D.host.F] e

theorem **infPart_finIn** (z : Af D.host.F ⊗[D.host.F] D.host.E) : infPart D (finIn D z) = 0

theorem **compAt_finIn** (w : Pl D) (z : Af D.host.F ⊗[D.host.F] D.host.E) : compAt D w (finIn D z) = GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w z

theorem **infPart_of_idAt** {P : Pl D → Prop} {a : AE D} (h : a ∈ idAt D P) : infPart D a = 0

theorem **compAt_of_idAt** {P : Pl D → Prop} {a : AE D} (h : a ∈ idAt D P) {w : Pl D} (hw : ¬ P w) : compAt D w a = 0

theorem **eq_finIn** {a : AE D} (h : infPart D a = 0) : a = finIn D (finPart D a)

theorem **placeIdempotent_integral_smul_mem_latticeCompletionAtPlace** {S : HSp D} {L : Lat S} {P : Pl D → Prop} (v : Pl D) {z : Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ intAE D) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (idem D (· = v) ∗ finIn D z) • y ∈ latV S L (· = v)

theorem **smul_latV** {S : HSp D} {L : Lat S} {P : Pl D → Prop} (v : Pl D) {e : AE D} (he : e ∈ intAt D (· = v)) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : e • y ∈ latV S L (· = v)

theorem **part_lat** (P : Pl D → Prop) (S : HSp D) (L : Lat S) (y : Fin S.rank → AE D) (hL : Submodule.IsLattice D.host.E L) : y ∈ latV S L P ↔ (∀ i, y i ∈ idAt D P) ∧ ∀ v, P v → ∀ e ∈ intAt D (· = v), e • y ∈ latV S L (· = v)

theorem **GlobalTheory.FinPl.localComponents** : ∀ (D : D7) (P : Pl D → Prop), (∃ ε : AE D, ε ∗ ε = ε ∧ cA D ε = ε ∧ ε ∈ intAt D P ∧ (∀ a, a ∈ idAt D P ↔ ε ∗ a = a) ∧ (∀ a, a ∈ idOff D P ↔ ε ∗ a = 0)) ∧ ∀ S : HSp D, (∀ h : GA D S, ∃ x ∈ grpAt S P, ∃ y ∈ grpOff S P, h = x ∗ y) ∧ (∀ x ∈ grpAt S P, ∀ y ∈ grpOff S P, x ∗ y = y ∗ x) ∧ grpAt S P ⊓ grpOff S P = ⊥ ∧ (∀ (L : Lat S) (y : Fin S.rank → AE D), (Submodule.IsLattice D.host.E L) → (y ∈ latV S L P ↔ (∀ i, y i ∈ idAt D P) ∧ ∀ v, P v → ∀ e ∈ intAt D (· = v), e • y ∈ latV S L (· = v)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicPointComponentsAtPlacesLatticesVacuaStatement

theorem **_root_.GlobalTheory.T.adelicPoint_components_atPlaces_lattices_vacua** : ∀ (D : D7) (P : Pl D → Prop), (∃ ε : AE D, ε ∗ ε = ε ∧ cA D ε = ε ∧ ε ∈ intAt D P ∧ (∀ a, a ∈ idAt D P ↔ ε ∗ a = a) ∧ (∀ a, a ∈ idOff D P ↔ ε ∗ a = 0)) ∧ ∀ S : HSp D, (∀ h : GA D S, ∃ x ∈ grpAt S P, ∃ y ∈ grpOff S P, h = x ∗ y) ∧ (∀ x ∈ grpAt S P, ∀ y ∈ grpOff S P, x ∗ y = y ∗ x) ∧ grpAt S P ⊓ grpOff S P = ⊥ ∧ (∀ (L : Lat S) (y : Fin S.rank → AE D), (Submodule.IsLattice D.host.E L) → (y ∈ latV S L P ↔ (∀ i, y i ∈ idAt D P) ∧ ∀ v, P v → ∀ e ∈ intAt D (· = v), e • y ∈ latV S L (· = v)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicPointNearCuspScale

theorem **exists_scale_of_bdd** {ι : Type∗} (N : ι → ℝ) (t a : ℝ) (ht : 0 < t) (hN : ∀ i, t ≤ N i) (ha : 0 < a) : ∃ m : ℝ, 0 < m ∧ m ≤ a ∧ (∀ i, m / 2 ≤ N i) ∧ (m = a ∨ ∃ i, N i ≤ m)

theorem **exists_root_scale** (n : ℕ) (s₀ m : ℝ) (hs : 0 < s₀) (hm : 0 < m) : ∃ η : ℝ, 0 < η ∧ s₀ ∗ η ^ (n + 1) = m

theorem **exists_scale_real** (I : AutInst) (y : I.Ginf) (s₀ η₀ : ℝ) (hs : 0 < s₀) (hη : 0 < η₀) : ∃ η : ℝ, 0 < η ∧ η ≤ η₀ ∧ NoShort I y (s₀ / 2 ∗ η ^ (I.n + 1)) ∧ (η₀ ≤ η ∨ ∃ u, IntVec I u ∧ u ≠ 0 ∧ ‖vec I y u‖ ≤ s₀ ∗ η ^ (I.n + 1))

theorem **close_mono** (I : AutInst) {u : I.Ginf} {a b : ℝ} (h : Close I u a) (hab : a ≤ b) : Close I u b

theorem **nearInst** : ∀ I : AutInst, NearInst I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicPointNearCuspScaleStatement

theorem **_root_.GlobalTheory.T.adelicPoint_nearCusp_scale** : ∀ I : AutInst, NearInst I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicSchwartzSlicePolynomialGaussianOfAnnihilat

theorem **compAt_add'** (θ : InfFam D) (τ : D.what₂.Tau) {m n : ℕ} (x y : Mx D m n) : compAt θ τ (x + y) = compAt θ τ x + compAt θ τ y

theorem **finMap_add** {m n : ℕ} (x y : Mx D m n) : (x + y).map (finPart D) = x.map (finPart D) + y.map (finPart D)

theorem **slice_polyGauss** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (l : D.what₂.Tau → ℝ) (s : RealMat V), (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ (N : ℕ), ∀ φ ∈ schwartzA θ V.rank n, (∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), L.length = N → L.foldr (fun q g => annS V l s θ q.1 q.2 g) φ = 0) → ∀ x₀ : Mx D V.rank n, ∃ p : MvPolynomial (D.what₂.Tau × Fin V.rank × Fin n × Bool) ℂ, ∀ x : Mx D V.rank n, x.map (finPart D) = x₀.map (finPart D) → φ x = evalPoly p (fun τ => compAt θ τ x) ∗ gauss V l s (fun τ => compAt θ τ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicSchwartzSlicePolynomialGaussianOfAnnihilatStatement

theorem **_root_.GlobalTheory.T.adelicSchwartz_slice_polynomial_gaussian_of_annihilated** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (l : D.what₂.Tau → ℝ) (s : RealMat V), (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ (N : ℕ), ∀ φ ∈ schwartzA θ V.rank n, (∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), L.length = N → L.foldr (fun q g => annS V l s θ q.1 q.2 g) φ = 0) → ∀ x₀ : Mx D V.rank n, ∃ p : MvPolynomial (D.what₂.Tau × Fin V.rank × Fin n × Bool) ℂ, ∀ x : Mx D V.rank n, x.map (finPart D) = x₀.map (finPart D) → φ x = evalPoly p (fun τ => compAt θ τ x) ∗ gauss V l s (fun τ => compAt θ τ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicTranslateOneEqStandardLattice

theorem **finM_one** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) : finM V hn (1 : GA D V) = 1

theorem **moves_one** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (N : Submodule (𝒪 D.host.E) (Fin n → D.host.E)), Moves V hn 1 N → N = Formalism.L2.Col.stdLat D.host.E n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicTranslateOneEqStandardLatticeStatement

theorem **_root_.GlobalTheory.T.adelicTranslate_one_eq_standardLattice** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (N : Submodule (𝒪 D.host.E) (Fin n → D.host.E)), Moves V hn 1 N → N = Formalism.L2.Col.stdLat D.host.E n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicTranslateStandardLatticeIsSelfDual

theorem **moves_exists** (V : HSp D) (g : GA D V) : ∃ N, Moves V rfl g N

theorem **isSdLat_of** (V : HSp D) (hV : V.gram = 1) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (h : IsSelfDualC V N) : Formalism.L2.Col.IsSdLat N

theorem **std_moved_lattice** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n), V.gram = 1 → (∀ g : GA D V, ∃ N, Moves V hn g N) ∧ ∀ (g : GA D V) N, Moves V hn g N → Formalism.L2.Col.IsSdLat N ∧ ∃ Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E), (Λ : Set (Fin V.rank → D.host.E)) = latFr V hn N ∧ IsSelfDualC V Λ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicTranslateStandardLatticeIsSelfDualStatement

theorem **_root_.GlobalTheory.T.adelicTranslate_standardLattice_isSelfDual** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n), V.gram = 1 → (∀ g : GA D V, ∃ N, Moves V hn g N) ∧ ∀ (g : GA D V) N, Moves V hn g N → Formalism.L2.Col.IsSdLat N ∧ ∃ Λ : Submodule (𝒪 D.host.E)

(Fin V.rank → D.host.E), (Λ : Set (Fin V.rank → D.host.E)) = latFr V hn N ∧ IsSelfDualC V Λ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicUnitaryEveryPointHasCuspScale

theorem **norm_mul_mul_apply_le** {n : ℕ} (P A Q : Matrix (Fin n) (Fin n) ℂ) (c : ℝ) (hA : ∀ k l, ‖A k l‖ ≤ c) (i j : Fin n) : ‖(P * A * Q) i j‖ ≤ (∑ k, ∑ l, ‖P i k‖ * ‖Q l j‖) * c

theorem **norm_uMat_inv_apply** {n : ℕ} {ε : Fin n → Bool} (y : Formalism.REAL.UJ ε) (i j : Fin n) : ‖uMat y⁻¹ i j‖ = ‖uMat y j i‖

theorem **frameBound_nonneg** (fr : UI.Frames D V) (τ : D.what₂.Tau) (i j : Fin V.rank) : 0 ≤ frameBound fr τ i j

theorem **brgFin_apply** (fr : UI.Frames D V) (k : (UI.instV fr).Gf) : brgFin fr k = UI.brg fr (1, k)

theorem **continuous_brgFin** (fr : UI.Frames D V) : Continuous (brgFin fr)

theorem **brg_entry_bounds** (fr : UI.Frames D V) (h2 : ∀ (x : (UI.instV fr).GA) (τ : D.what₂.Tau) (A : Matrix (Fin V.rank) (Fin V.rank) ℂ), A = uMat (x.1 (Formalism.L2.Host.tauPl D τ)) → infMat D.sp.cm.sig V.gram (θ τ) (UI.brg fr x) = fr.P (Formalism.L2.Host.tauPl D τ) * A * (fr.P (Formalism.L2.Host.tauPl D τ))⁻¹) : ∃ S : ℝ, 0 ≤ S ∧ ∀ (y : (UI.instV fr).Ginf) (c : ℝ), ((∀ v i j, ‖uMat (y v) i j‖ ≤ c) → normInf θ V (UI.brg fr (y, 1)) ≤ S * c) ∧ ((∀ v i j, ‖(uMat (y v) − 1) i j‖ ≤ c) → ∀ τ i j, ‖(infMat D.sp.cm.sig V.gram (θ τ) (UI.brg fr (y, 1)) − 1) i j‖ ≤ S * c)

theorem **near** (D : D7) (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)) (hθ : IsInfFam θ) (V : HSp D) : Near θ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicUnitaryEveryPointHasCuspScaleStatement

theorem **_root_.GlobalTheory.T.adelicUnitary_every_point_has_cusp_scale** : ∀ (D : D7) (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)), IsInfFam θ → ∀ V : HSp D, Near θ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicUnitaryRealMulFiniteDecomposition

theorem **infMat_eq_map** (V : HSp D) (θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ) (h : GA D V) : infMat D.sp.cm.sig V.gram θ h = (infHom V h).1.map θ

theorem **split_parts** : ∀ (D : D7) (V : HSp D), (∀ h : GA D V, ∃ a ∈ GinfC V, ∃ k ∈ GfC V, h = a * k) ∧ ∀ a ∈ GinfC V, ∀ k ∈ GfC V, Lev.finE V (a * k) = Lev.finE V k ∧ ∀ θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ, infMat D.sp.cm.sig V.gram θ (a * k) = infMat D.sp.cm.sig V.gram θ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdelicUnitaryRealMulFiniteDecompositionStatement

theorem **_root_.GlobalTheory.T.adelicUnitary_real_mul_finite_decomposition** : ∀ (D : D7) (V : HSp D), (∀ h : GA D V, ∃ a ∈ GinfC V, ∃ k ∈ GfC V, h = a * k) ∧ ∀ a ∈ GinfC V, ∀ k ∈ GfC V, Lev.finE V (a * k) = Lev.finE V k ∧ ∀ θ : Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ, infMat D.sp.cm.sig V.gram θ (a * k) = infMat D.sp.cm.sig V.gram θ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdicCompletionValuationMapUniformiser

theorem **adicCompletion_valuation_map_uniformiser** : ∀ (F : Type) [Field F] [NumberField F] (v : IsDedekindDomain.HeightOneSpectrum (NumberField.RingOfIntegers F)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (φ : k ≃+* v.adicCompletion F), (∀ a, Valued.v (φ a) ≤ 1 ↔ a ∈ 𝒪[k]) → ∀ ϖ : 𝒪[k], Irreducible ϖ → Valued.v (φ (ϖ : k)) = ((Multiplicative.ofAdd (−1 : ℤ) : Multiplicative ℤ) : WithZero (Multiplicative ℤ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdicCompletionValuationMapUniformiserStatement

theorem **_root_.GlobalTheory.T.valued_v_map_irreducible_eq_ofAdd_neg_one** : ∀ (F : Type) [Field F] [NumberField F] (v : IsDedekindDomain.HeightOneSpectrum (NumberField.RingOfIntegers F)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (φ : k ≃+* v.adicCompletion F), (∀ a, Valued.v (φ a) ≤ 1 ↔ a ∈ 𝒪[k]) → ∀ ϖ : 𝒪[k], Irreducible ϖ → Valued.v (φ (ϖ : k)) = ((Multiplicative.ofAdd (−1 : ℤ) : Multiplicative ℤ) : WithZero (Multiplicative ℤ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdjoinRangeGLEqTopMatrix

theorem **adjoin_range_GL_eq_top** (R : Type) [CommRing R] (n : ℕ) : Algebra.adjoin R ((fun x : GL (Fin n) R => (x : Matrix (Fin n) (Fin n) R)) '' Set.univ) = ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdjoinRangeGLEqTopMatrixStatement

theorem **_root_.GlobalTheory.T.adjoin_range_GL_eq_top_matrix** : ∀ (R : Type) [CommRing R] (n : ℕ), Algebra.adjoin R ((fun x : GL (Fin n) R => (x : Matrix (Fin n) (Fin n) R)) '' Set.univ) = ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdjoinUnitaryGroupHermitianLocalEqTop

theorem **refl_unitary** (hs : ∀ x, s (s x) = x) {n : ℕ} (d : Fin n → R) (hd : ∀ k, s (d k) = d k) (c : R) (v : Fin n → R) (hc : c + s c + c * s c * ∑ k, s (v k) * d k * v k = 0) : cT s (1 + c • vecMulVec v (fun k => s (v k) * d k)) * diagonal d * (1 + c • vecMulVec v (fun k => s (v k) * d k)) = diagonal d

theorem **mem_uAlg_of_unitary** {n : ℕ} {J : Matrix (Fin n) (Fin n) R} (hJ : IsUnit J.det) {M : Matrix (Fin n) (Fin n) R} (hM : cT s M * J * M = J) : M ∈ uAlg s J

theorem **isUnit_det_diagonal** {n : ℕ} {d : Fin n → R} (hdu : ∀ k, IsUnit (d k)) : IsUnit (diagonal d).det

theorem **vec_mem** (hs : ∀ x, s (s x) = x) {ζ : R} (hζ : ζ * s ζ = 1) (hζ1 : IsUnit (ζ − 1)) {n : ℕ} {d : Fin n → R} (hdu : ∀ k, IsUnit (d k)) (hd : ∀ k, s (d k) = d k) (v : Fin n → R) (hN : IsUnit (∑ k, s (v k) * d k * v k)) : vecMulVec v (fun k => s (v k) * d k) ∈ uAlg s (diagonal d)

theorem **single_mem_of_isUnit** {n : ℕ} {J : Matrix (Fin n) (Fin n) R} (i j : Fin n) {u : R} (hu : IsUnit u) (h : single i j u ∈ uAlg s J) : single i j (1 : R) ∈ uAlg s J

theorem **ind_mem** (hs : ∀ x, s (s x) = x) {ζ : R} (hζ : ζ * s ζ = 1) (hζ1 : IsUnit (ζ − 1)) {n : ℕ} {d : Fin n → R} (hdu : ∀ k, IsUnit (d k)) (hd : ∀ k, s (d k) = d k) (S : Finset (Fin n)) (hN : IsUnit (∑ k ∈ S, d k)) : vecMulVec (fun k => if k ∈ S then (1 : R) else 0) (fun k => if k ∈ S then d k else 0) ∈ uAlg s (diagonal d)

theorem **single_mem_diag** (hs : ∀ x, s (s x) = x) [IsLocalRing R] {ζ : R} (hζ : ζ * s ζ = 1) (hζ1 : IsUnit (ζ − 1)) {n : ℕ} (hn : 3 ≤ n) {d : Fin n → R} (hdu : ∀ k, IsUnit (d k)) (hd : ∀ k, s (d k) = d k) (i j : Fin n) : single i j (1 : R) ∈ uAlg s (diagonal d)

theorem **eq_top_of_single** {n : ℕ} {A : Subalgebra R (Matrix (Fin n) (Fin n) R)} (h : ∀ i j, single i j (1 : R) ∈ A) : A = ⊤

theorem **isUnit_det_cT** {n : ℕ} {Q : Matrix (Fin n) (Fin n) R} (hQ : IsUnit Q.det) : IsUnit (cT s Q).det

theorem **uAlg_conj** {n : ℕ} {J Q : Matrix (Fin n) (Fin n) R} (hQ : IsUnit Q.det) (hJ : IsUnit J.det) {M : Matrix (Fin n) (Fin n) R} (hM : M ∈ uAlg s J) : Q⁻¹ * M * Q ∈ uAlg s (cT s Q * J * Q)

theorem **adjoin_unitary_eq_top** : ∀ (R : Type) [CommRing R] [IsLocalRing R] (s : R →+* R), (∀ x, s (s x) = x) → (∃ δ : R, IsUnit (δ − s δ)) → ∀ n : ℕ, 3 ≤ n → ∀ G : Matrix (Fin n) (Fin n) R, cT s G = G → IsUnit G → Algebra.adjoin R ((fun x : GL (Fin n) R => (x : Matrix (Fin n) (Fin n) R)) '' (UJ s G : Set (GL (Fin n) R))) = ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdjoinUnitaryGroupHermitianLocalEqTopStatement

theorem **_root_.GlobalTheory.T.adjoin_unitaryGroup_hermitian_local_eq_top** : ∀ (R : Type) [CommRing R] [IsLocalRing R] (s : R →+* R), (∀ x, s (s x) = x) → (∃ δ : R, IsUnit (δ − s δ)) → ∀ n : ℕ, 3 ≤ n → ∀ G : Matrix (Fin n) (Fin n) R, cT s G = G → IsUnit G → Algebra.adjoin R ((fun x : GL (Fin n) R => (x : Matrix (Fin n) (Fin n) R)) '' (UJ s G : Set (GL (Fin n) R))) = ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdmissibleRanksAllMembers

theorem **typePrimesLarge_of_val** {C} {cD ch : ℝ} {D : D7} (L : SizeData₂ D) (hv : Val C cD ch L) : GlobalTheory.IndefiniteRowSideFullHecke.UnipotentLevelStabilisers.TypePrimesLarge D

theorem **dstar_mass** (K : Ctx) (X : Stages) (hO : K.Over X.B.toPre) (hd : ∀ τ, X.S.dsig τ = Vol.trueDsig X.B.R (X.B.hc τ)) : X.toD7.what₂.Dstar = StandardLattice.DstarP X.toD7 (X.toD7.host.g 0) (X.toD7.host.g 1) (StandardLattice.massI X.toD7 X.toD7.R.r) X.toD7.dsig

theorem **columnRowRanks_admissible_of_massFormula** (D : D7) (Lv : LevelData D) (C : Formalism.L2.Forms.ColsTiedSym D) (hD : D.what₂.Dstar = StandardLattice.DstarP D (D.host.g 0) (D.host.g 1) (StandardLattice.massI D D.R.r) D.dsig) (hG : StandardLattice.GBig D) (hT : ∀ u, Formalism.MainNotions.S.SplitsE D (D.mem u).A.T) : LocalBesselAndTopKernel.FamilyMidVolume D

theorem **admissible_ranks_at_member** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => Formalism.MainNotions.LocalBesselAndTopKernel.FamilyMidVolume D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AdmissibleRanksAllMembersStatement

theorem **_root_.GlobalTheory.T.admissible_ranks_all_members** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => Formalism.MainNotions.LocalBesselAndTopKernel.FamilyMidVolume D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AeCharUnramAt

theorem **ae_charUnramAt** (χ : (AE D)ˣ →* Circle) (hχ : Continuous χ) : ∀ᶠ v : GlobalTheory.FinPl.Pl D in Filter.cofinite, GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D χ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AlVEmbHEqNegOneZpow

theorem **tauE_star** (D : D7) (τ : D.what₂.Tau) (x : D.host.E) : tauE D τ (star x) = starRingEnd ℂ (tauE D τ x)

theorem **sum_two_re_tauE** (D : D7) (e : D.host.E) : ∑ t, (((2 * (tauE D t e).re : ℝ)) : ℂ) = ((Algebra.trace ℚ D.host.E e : ℚ) : ℂ)

theorem **re_xi_mul_sq** (u : ℝ) (hu : 0 < u) : (Complex.I * ((u⁻¹ : ℝ) : ℂ) * ((1 + Complex.I * (u : ℂ)) / 2) ^ 2).re = -1 / 2

theorem **alV_embH_eq_neg_one_zpow** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (y : Formalism.L2.Col.HomE D.host.E a b) (k : ℤ), (k : ℚ) = Algebra.trace ℚ D.host.E (D.host.xi0 * ((1 + D.host.sqrtMinusD) / 2) ^ 2 * GlobalTheory.LatVac.hRat S T ((LinearMap.toMatrix' (y)).reindex (finCongr hT.symm) (finCongr hS.symm)) ((LinearMap.toMatrix' (y)).reindex (finCongr hT.symm) (finCongr hS.symm))) → alV D.host.d sc (embH (tauE D) δ y) = (-1 : ℂ) ^ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllPositionsHolds

theorem **allPositions_holds** : AllPositions

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllTrueFamilyDatum

theorem **shape_upd** {v : VCom c} {o : GlobalTheory.SeeSaw.TDat c k ι} (h : ShapeOff v o) (b Ψ ΨT JG eG eGT dG) : ShapeOff { v with b

theorem **moves_one** (S : HSp D) (L : GlobalTheory.FinPl.Lat S) (P : GlobalTheory.FinPl.Pl D → Prop) : GlobalTheory.FinPl.Moves S 1 L L P

theorem **exists_places** (D : D7) (T : ℕ) (hT : T ≠ 0) : ∃ s : Finset (GlobalTheory.FinPl.Pl D), ∀ p, p ∈ s ↔ ((T : ℕ) : 𝒪 D.host.F) ∈ p.asIdeal

theorem **meas_upd** {a : XGeo D} {y : YRat a} {m : YMeas y} {b} (h : GeoMeas (pk W) (mkGeo m b)) (b') : GeoMeas (pk W) (mkGeo m b')

theorem **tie_upd** {f : FamCh W u} {v : VCom f.c} (h : VTie f v) (JG eG eGT dG) : VTie f { v with JG

theorem **selfDual_top** (h : PkHyp W) : IsSelfDualC D.sp.X' (pk W).ΛX

theorem **core** (hD : GlobalTheory.Published.DiscreteSeriesCover) (hH : GlobalTheory.Published.HarishChandraModules) (hPk : PkHyp W) (hTy : GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg) (hT0 : ((D.mem u).A.T : ℕ) ≠ 0) (hGT : ∀ PP ∈ (pk W).Pg, (((D.mem u).A.T : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal) (hQG : ∀ PP, QAll W PP → PP ∉ (pk W).Pg) (jj : SmT (pk W).MX →ₗ[ℂ] (Formalism.L2.Forms.FrP D → ℂ)) (hjj : TopReadAt W u jj 1) (a : XGeo D) (c₀ : ℕ → ℂ) (hx : ∀ j, j ≤ D.R.m → (a.x j).Is) (hbT : a.bT = 1) (hLT : a.LT = (pk W).ΛX) (hline : IsLineVec a.Lz a.ζ₀) (hn : star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E)) (horth : ∀ n (h : n = D.R.m) (x : GlobalTheory.FinPl.Pl D), (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∉ x.asIdeal → ∃ LZ : GlobalTheory.FinPl.Lat a.Lz, IsSelfDualC _ LZ ∧ GlobalTheory.FinPl.OrthAt (a.oT n h) (a.x n).L a.LT LZ x) (hloc : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → ∃ s : LocX a (p

theorem **exists_geoAll** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → GlobalTheory.Published.StrongApproxSU → ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ u : B.Word, ∃ a : GeoAll W u, a.Is ∧ a.True ∧ (∀ r s, (a.g.oT B.R.m rfl).iA r s = if r.val = s.val then 1 else 0) ∧ GlobalTheory.MiTop.Framed a.d.B

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllTrueFamilyDatumStatement

theorem **_root_.GlobalTheory.T.all_true_family_datum** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → GlobalTheory.Published.StrongApproxSU → ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc :

Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ u : B.Word, ∃ a : GeoAll W u, a.Is ∧ a.True ∧ (∀ r s, (a.g.oT B.R.m rfl).iA r s = if r.val = s.val then 1 else 0) ∧ GlobalTheory.MiTop.Framed a.d.B

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllTrueFamilyGeometry

theorem **GlobalTheory.XGeoStd.exists_xgeo** : GlobalTheory.Published.StrongApproxSU → ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ u : B.Word, ∃ (a : XGeo (I.rec7 l Z)) (c₀ : ℕ → ℂ), (∀ j, j ≤ B.R.m → (a.x j).Is) ∧ GlobalTheory.AdForm.PinX I l Z (pk W).ψ (pk W).θ a.x a.vac c₀ ∧ a.bT = 1 ∧ (∀ r s, (a.oT B.R.m rfl).iA r s = if r.val = s.val then 1 else 0) ∧ a.LT = (pk W).ΛX ∧ IsLineVec a.Lz a.ζ₀ ∧ star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = ((((I.rec7 l Z).mem u).A.T : ℕ) : (I.rec7 l Z).host.E) ∧ GlobalTheory.AdForm.IsE

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllTrueFamilyGeometryColumn

theorem **pr_mem_of_fin_int** (p : SplitPt D k) {a : AE D} (ha : finPart D a ∈ intAE D) : p.pr a ∈ 𝒪[k]

theorem **splitPlace_valuation_cmImaginaryGenerator_eq_one** (p : SplitPt D k) (hp : p.UnitAA) : valuation k (p.pr (incA D D.host.xi0)) = 1

theorem **splitPlace_valuation_neg_cmGenerator_pow_eq_one** (p : SplitPt D k) (hp : p.UnitAA) (n : ℕ) : valuation k (p.pr (incA D (-D.host.xi0 ^ n))) = 1 ∧ valuation k (p.pr (incA D (star (-D.host.xi0 ^ n)))) = 1

theorem **GlobalTheory.LevPrimeData.data_preT** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ : ℕ → ℂ) (v : VCom (commOf (pk W) u g)) (ζ₀ : Fin g.Lz.rank → (I.rec7 l Z).host.E), GlobalTheory.AdForm.PinX I l Z (pk W).ψ (pk W).θ g.x g.vac c₀ → v.NT = (pk W).ΛX → (∀ p, p ∈ v.tP ↔ ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ p.asIdeal) → IsLineVec g.Lz ζ₀ → (∀ r, vecB (g.oT B.R.m rfl) ζ₀ r = (show Fin (B.R.N B.R.m) → (I.rec7 l Z).host.E from TC.z0 X u) (Fin.ca

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllTrueFamilyGeometryColumnStatement

theorem **_root_.GlobalTheory.T.all_true_family_geometry_column** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ : ℕ → ℂ) (v : VCom (commOf (pk W) u g)) (ζ₀ : Fin g.Lz.rank → (I.rec7 l Z).host.E), GlobalTheory.AdForm.PinX I l Z (pk W).ψ (pk W).θ g.x g.vac c₀ → v.NT = (pk W).ΛX → (∀ p, p ∈ v.tP ↔ ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ p.asIdeal) → IsLineVec g.Lz ζ₀ → (∀ r, vecB (g.oT B.R.m rfl) ζ₀ r = (show Fin (B.R.N B.R.m) → (I.rec7 l Z).host.E from

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AllTrueFamilyGeometryStatement

theorem **_root_.GlobalTheory.T.all_true_family_geometry** : GlobalTheory.Published.StrongApproxSU → ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ u : B.Word, ∃ (a : XGeo (I.rec7 l Z)) (c₀ : ℕ → ℂ), (∀ j, j ≤ B.R.m → (a.x j).Is) ∧ GlobalTheory.AdForm.PinX I l Z (pk W).ψ (pk W).θ a.x a.vac c₀ ∧ a.bT = 1 ∧ (∀ r s, (a.oT B.R.m rfl).iA r s = if r.val = s.val then 1 else 0) ∧ a.LT = (pk W).ΛX ∧ IsLineVec a.Lz a.ζ₀ ∧ star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = ((((I.rec7 l Z).mem u).A.T : ℕ) : (I.rec7 l Z).host.E) ∧ GlobalThe

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AltENsmulLeft

theorem **hv_apply'** (sc : ℝ) (z w : VV ι a b) : HV sc z w = (sc : ℂ) * ∑ τ, ∑ i, ∑ j, z τ i j * (starRingEnd ℂ) (w τ i j)

theorem **embH_apply'** (emb : ι → E →+* ℂ) (δ : ℂ) (x : HomE E a b) (τ : ι) (i : Fin b) (j : Fin a) : embH emb δ x τ i j = δ * emb τ (x (Pi.single j 1) i)

theorem **hv_smul_emb** (emb : ι → E →+* ℂ) (δ : ℂ) (sc : ℝ) (hst : ∀ (τ : ι) (x : E), emb τ (star x) = starRingEnd ℂ (emb τ x)) (r : E) (x y : HomE E a b) : HV sc (embH emb δ (r • x)) (embH emb δ y) = HV sc (embH emb δ x) (embH emb δ (star r • y))

theorem **embH_injective'** [Nonempty ι] (emb : ι → E →+* ℂ) {δ : ℂ} (hδ : δ ≠ 0) : Function.Injective (embH emb δ : HomE E a b → VV ι a b)

theorem **altE_nsmul_left** {V : Type} [AddCommGroup V] [Module ℂ V] (H : Sesq V) (n : ℕ) (v w : V) : altE H (n • v) w = altE H v (n • w)

theorem **npow_smul_mem** (L : Submodule (𝒪 E) W) (c n : ℕ) {y : W} (hy : y ∈ L) : ((c : E) ^ n) • y ∈ L

theorem **zpow_smul_mem** (L : Submodule (𝒪 E) W) (c : ℕ) {k : ℤ} (hk : 0 ≤ k) {y : W} (hy : y ∈ L) : ((c : E) ^ k) • y ∈ L

theorem **twO_selfDual** : ∀ (E : Type) [Field E] [NumberField E] [StarRing E] (ι : Type) [Fintype ι] [Nonempty ι] (a b : ℕ) (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)) (c : ℕ) (e : HeightOneSpectrum (𝒪 E) → ℤ), δ ≠ 0 → (∀ (τ : ι) (x : E), emb τ (star x) = starRingEnd ℂ (emb τ x)) → 0 < c → (∀ Q, e (Formalism.L2.Col.conjPrime₂ Q) = -e Q) → (∀ Q, e Q = 1 ∨ e Q = 0 ∨ e Q = -1) → Formalism.L2.Th1.H.IsSelfDual (HV sc) (laV emb δ M) → Formalism.L2.Th1.H.IsSemiChar (HV sc) (laV emb δ M) (alV d sc) → Formalism.L2.Th1.H.IsSelfDual (HV sc) (laV emb δ (twist₂ M c e))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmpQNotEqAmpG

private theorem **ampQ_not_eq_ampG** {Cl Nu : Type} (L : PN.LocalNG Cl Nu) : ampQ Not L = ampG L

private theorem **hasPt_of_isPt** {D : D7} {X : HidaSetting D} (Y : Full X) (u : D.what₂.Word) (νL νB : X.B.I.Ginf → Measure X.B.I.Ginf) (ng : PN.Pts D.host.E D.R.rS → ℝ) (γ : PN.Pts D.host.E D.R.rS) (h : PN.IsPt (genT (scT (Y.tO u u))) (NOf1 Y u νL νB ng) γ) : HasPt Y u u (Y.tO u u).genSet γ

theorem **stepFam1_of** : (∀ D : D7, 4 * D.R.rS + 7 ≤ D.R.k → Formalism.L2.SG.FamDefs.LocalLowerBoundAllCosets D.R.k D.R.rS 9) → Formalism.L2.SG.FamDefs.HTie → FamGen1 → FamilyWeightSumUnfoldingIdentity → WeightsFam → FamilyGroupSumFactorsWeightAmplitude → ExistFam → StepFam1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmpleThetaWideRangeLargeConstants

theorem **ptWide_at_record_gap** : ∀ Cg cb Cu cT CT C0 : ℝ, ∃ C : ℝ, ∀ Cn : ℝ, ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl : ℝ), thr ≤ C' → thr ≤ Cl → Ties D → Formalism.L2.SG.RowsConc.TFrameAt D → VanishingOnXAndXPlusBranching.adm1 D large₂ C' Cl Cg cb Cu cT CT C0 Cn 1 C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmpleThetaWideRangeLargeConstantsStatement

theorem **_root_.GlobalTheory.T.ampleTheta_wideRange_large_constants** : ∀ Cg cb Cu cT CT C0 : ℝ, ∃ C : ℝ, ∀ Cn : ℝ, ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl : ℝ), thr ≤ C' → thr ≤ Cl → Ties D → Formalism.L2.SG.RowsConc.TFrameAt D → VanishingOnXAndXPlusBranching.adm1 D large₂ C' Cl Cg cb Cu cT CT C0 Cn 1 C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmplitudeBoundsAtParametersOfAdmittedBoundsStatement

theorem **_root_.GlobalTheory.T.ampAt_of_admitted_bounds** : ∀ Cg CPT CDl hilbertNinetyConstant Ca CS : ℝ, 0 ≤ hilbertNinetyConstant → 0 ≤ Ca → ∃ C C2 Ch : ℝ, 0 ≤ C ∧ 0 ≤ Ch ∧ ∀ Cpr δ : ℝ, 0 < δ → ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl cb Cu cT CT C0 : ℝ), thr ≤ C' → thr ≤ Cl → 1 ≤ large₂.CGam → VanishingOnXAndXPlusBranching.adm1 D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 CPT → VanishingOnXAndXPlusBranching.adm2 D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 CDl → VanishingOnXAndXPlusBranching.adm5 D large₂ C' Cl Cg cb Cu cT CT C0 → VanishingOnXAndXPlusBranching.adm9 D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 → DegreeRatioUnitShiftBound D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 hilbertNinetyConstant → AbsAllParameters D large₂ C' Cl Cg cb Cu cT CT C0 1 1 (max Cpr 0) Ca → VanishingOnXAndXPlusBranching.adm11 D large₂ C' Cl Cg cb Cu cT CT C0 1 CS → VanishingOnXAndXPlus

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmplitudeBoundsAtParamsOfAdmittedBounds

theorem **ampAt_record_of_inputs** : ∀ Cg CPT CDl hilbertNinetyConstant Ca CS : ℝ, 0 ≤ hilbertNinetyConstant → 0 ≤ Ca → ∃ C C2 Ch : ℝ, 0 ≤ C ∧ 0 ≤ Ch ∧ ∀ Cpr δ : ℝ, 0 < δ → ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl cb Cu cT CT C0 : ℝ), thr ≤ C' → thr ≤ Cl → 1 ≤ large₂.CGam → VanishingOnXAndXPlusBranching.adm1 D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 CPT → VanishingOnXAndXPlusBranching.adm2 D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 CDl → VanishingOnXAndXPlusBranching.adm5 D large₂ C' Cl Cg cb Cu cT CT C0 → VanishingOnXAndXPlusBranching.adm9 D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 → DegreeRatioUnitShiftBound D large₂ C' Cl Cg cb Cu cT CT C0 (max Cpr 0) 1 hilbertNinetyConstant → AbsAllParameters D large₂ C' Cl Cg cb Cu cT CT C0 1 1 (max Cpr 0) Ca → VanishingOnXAndXPlusBranching.adm11 D large₂ C' Cl Cg cb Cu cT CT C0 1 CS → VanishingOnXAndXPlusBranching.adm12 D → AmpA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmplitudeSchurBoundUniformConstants

theorem **schurSt_of_big_tie_uniform** : ∀ Cg cb Cu cT CT C1 : ℝ, ∃ C₀ : ℝ, ∀ C ≥ C₀, ∃ m₀ : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl C0 : ℝ), Big D large₂ Cl m₀ → cdScaleTie D → BSch D large₂ C' Cl Cg cb Cu cT CT C0 → LocIn D large₂ C' Cl Cg cb Cu cT CT C0 → SchurSt D large₂ C' Cl Cg cb Cu cT CT C0 C1 C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AmplitudeSchurBoundUniformConstantsStatement

theorem **_root_.GlobalTheory.T.amplitudeSchurBound_uniform_constants** : ∀ Cg cb Cu cT CT C1 : ℝ, ∃ C₀ : ℝ, ∀ C ≥ C₀, ∃ m₀ : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl C0 : ℝ), Big D large₂ Cl m₀ → cdScaleTie D → BSch D large₂ C' Cl Cg cb Cu cT CT C0 → LocIn D large₂ C' Cl Cg cb Cu cT CT C0 → SchurSt D large₂ C' Cl Cg cb Cu cT CT C0 C1 C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AnalyticTiesTrueCols

theorem **analyticTies_trueCols** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (hX : B.fields.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)), ColIn.JetsIn I → TaylorLogMetric.IsDiscriminantTwelfthRootOfIdealLattice (I.rec7 l Z) ⊤ I.J.δ ∧ TaylorLogMetric.IsChi (I.rec7 l Z) I.chi ∧ IsDgen (I.rec7 l Z) I.J.cD ∧ ∀ u a, GlobalTheory.IndefiniteRowSideFullHecke.PlaceLogThetaBounds.FieldOfDefinition (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AnisotropicCoreWittFrame

theorem **form_mulVec** {D : D7} {n m : ℕ} (A : Matrix (Fin n) (Fin m) D.host.E) (G : Matrix (Fin n) (Fin n) D.host.E) (w : Fin m → D.host.E) : star (A *ᵥ w) ⬝ᵥ (G *ᵥ (A *ᵥ w)) = star w ⬝ᵥ ((Aᴴ * G * A) *ᵥ w)

theorem **isWittFrame_of_hypDec** (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (h : IsAniso T₀) : IsWittFrame D.sp.cm.sig T.gram d.e d.e'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AnnihilationOperatorDoubledHermitianSpaceDecomp

theorem **mul_split** {b n k : ℕ} (W : Matrix (Fin b) (Fin (n + n)) ℂ) (N : Matrix (Fin (n + n)) (Fin k) ℂ) : W * N = W.submatrix id (fun i => finSumFinEquiv (Sum.inl i)) * N.submatrix (fun i => finSumFinEquiv (Sum.inl i)) id + W.submatrix id (fun i => finSumFinEquiv (Sum.inr i)) * N.submatrix (fun i => finSumFinEquiv (Sum.inr i)) id

theorem **annih_dec** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (μ : D.host.E), IsSkew μ → ∀ (s' : RealMat (dblSp A)) (sB : RealMat B), IsStdInv A μ s' → (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) → ∀ (τ : D.what₂.Tau) (Y : Matrix (Fin B.rank) (Fin (dblSp A).rank) ℂ), ∃ (X X' : Matrix (Fin B.rank) (Fin A.rank) ℂ) (c : ℂ), ∀ f, IsThetaFunO ψ (dblSp A) B f → annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) Y f = annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) (X * (dblDec A).qX.map (tauE D τ)) f + c • annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) (X' * (dblDec A).qX.map (tauE D τ)) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AnnihilationOperatorDoubledHermitianSpaceDecompStatement

theorem **_root_.GlobalTheory.T.annihilationOperator_doubledHermitianSpace_decomp** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (μ : D.host.E), IsSkew μ → ∀ (s' : RealMat (dblSp A)) (sB : RealMat B), IsStdInv A μ s' → (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) → ∀ (τ : D.what₂.Tau) (Y : Matrix (Fin B.rank) (Fin (dblSp A).rank) ℂ), ∃ (X X' : Matrix (Fin B.rank) (Fin A.rank) ℂ) (c : ℂ), ∀ f, IsThetaFunO ψ (dblSp A) B f → annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) Y f = annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) (X * (dblDec A).qX.map (tauE D τ)) f + c • annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) (X' * (dblDec A).qX.map (tauE D τ)) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AnnihilationOperatorSchroedingerFunctionsDoubled

theorem **blockMatrix_swap_mul_top_rows_eq_bottom_rows** {n k : ℕ} (M : Matrix (Fin (n + n)) (Fin k) ℂ) : (blk4 (0 : Matrix (Fin n) (Fin n) ℂ) 1 (-1) 0 * M).submatrix (fun i => finSumFinEquiv (Sum.inl i)) id = M.submatrix (fun i => finSumFinEquiv (Sum.inr i)) id

theorem **row_sub_mul** {a b k l : ℕ} (M : Matrix (Fin a) (Fin k) ℂ) (N : Matrix (Fin k) (Fin l) ℂ) (r : Fin b → Fin a) : M.submatrix r id * N = (M * N).submatrix r id

theorem **qX_mul_stdInv** (A : HSp D) (μ : D.host.E) (hμ : IsSkew μ) (s : RealMat (dblSp A)) (hs : IsStdInv A μ s) (τ : D.what₂.Tau) : (dblDec A).qX.map (tauE D τ) * s τ = ((‖tauE D τ μ‖ : ℂ)⁻¹) • (dblDec A).qY.map (tauE D τ)

theorem **re_imaginary_mul_neg_I_div_mul** {a w : ℂ} (hre : a.re = 0) (him : 0 < a.im) (r : ℝ) : (a * (-(Complex.I / (r : ℂ))) * w).re = ‖a‖ / r * w.re

theorem **annih_sch** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ μ : D.host.E, IsSkew μ → ∀ (s' : RealMat (dblSp A)) (sB : RealMat

B), IsStdInv A μ s' → (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) → ∀ l : D.what₂.Tau → ℝ, (∀ τ, |l τ| = ‖tauE D τ D.host.xi0‖ / ‖tauE D τ μ‖) → ∀ f, IsThetaFunO ψ (dblSp A) B f → ∀ (τ : D.what₂.Tau) (X : Matrix (Fin B.rank) (Fin A.rank) ℂ), sch ψ A B ν 𝒴 (annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) (X ∗ (dblDec A).qX.map (tauE D τ)) f) = annS B l sB θ τ X (sch ψ A B ν 𝒴 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AnnihilationOperatorSchroedingerFunctionsDoubledStatement

theorem **_root_.GlobalTheory.T.annihilationOperator_schroedingerFunctions_doubledPair** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ μ : D.host.E, IsSkew μ → ∀ (s' : RealMat (dblSp A)) (sB : RealMat B), IsStdInv A μ s' → (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) → ∀ l : D.what₂.Tau → ℝ, (∀ τ, |l τ| = ‖tauE D τ D.host.xi0‖ / ‖tauE D τ μ‖) → ∀ f, IsThetaFunO ψ (dblSp A) B f → ∀ (τ : D.what₂.Tau) (X : Matrix (Fin B.rank) (Fin A.rank) ℂ), sch ψ A B ν 𝒴 (annih (heisO ψ (dblSp A) B) θ τ (cplxO (s' τ) (sB τ)) (X ∗ (dblDec A).qX.map (tauE D τ)) f) = annS B l sB θ τ X (sch ψ A B ν 𝒴 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AntiOfStep

theorem **anti_of_step** {r : ℕ} (f : Fin r → ℝ) (h : ∀ (i : Fin r) (hi : i.1 + 1 < r), f ⟨i.1 + 1, hi⟩ < f i) : ∀ a b : Fin r, a ≤ b → f b ≤ f a

theorem **lamS_node** (R : Ranks) (P : HCParam R) (j : ℕ) (hj : j < R.m) (hi : R.k - 1 + j < R.r) : lamS R P ⟨R.k - 1 + j, hi⟩ = -(((P.b ⟨j, hj⟩ : ℚ)) : ℝ)

theorem **node_of_sign** (R : Ranks) (a b : Fin R.r) (hab : a < b) (hne : epsS R a ≠ epsS R b) : R.k ≤ b.1

theorem **gap_lt** {D : D7} {L C' Cl Cg cb Cu cT CT : ℝ} (H : GlobalTheory.Core.Deg.HypL D L C' Cl Cg cb Cu cT CT) (τ : D.what₂.Tau) (a b : Fin D.R.r) (hab : a < b) (hne : epsS D.R a ≠ epsS D.R b) : Cl ∗ (D.R.r : ℝ) ^ 19 ∗ (D.R.f : ℝ) ^ 5 ∗ L ≤ lamS D.R (D.hc τ) a - lamS D.R (D.hc τ) b

theorem **hd_nc** : ∃ Cl₀ : ℝ, ∀ (D : D7) (L C' Cl Cg cb Cu cT CT : ℝ), Cl₀ ≤ Cl → HD D L C' Cl Cg cb Cu cT CT → ∀ X : Formalism.L2.SG.HidaSetting.HidaSetting D, GlobalTheory.Count.GapGe X.B (100 ∗ (D.R.r : ℝ) ^ 4)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanLevelVolumeMulRelIndex

theorem **mul_set_eq_comap** (V : HSp D) (U : Subgroup (GA D V)) (hU : U ≤ GfC V) : {g : GA D V | ∃ k ∈ GinfC V, ∃ κ ∈ U, g = k ∗ κ} = (((U.subgroupOf (GfC V)).comap (GlobalTheory.AutMult.toFin V) : Subgroup (GA D V)) : Set (GA D V))

theorem **volLev_mul_relIndex** : ∀ (D : D7) (V : HSp D) (v : GlobalTheory.Unf.Vol V) (U U' : Subgroup (GA D V)), U ≤ U' → U' ≤ GfC V → IsCompact (U' : Set (GA D V)) → IsOpen ((U.subgroupOf (GfC V) : Subgroup (GfC V)) : Set (GfC V)) → volLev v U ∗ (U.relIndex U' : ℝ) = volLev v U'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanLevelVolumeMulRelIndexStatement

theorem **_root_.GlobalTheory.T.archimedeanLevelVolume_mul_relIndex** : ∀ (D : D7) (V : HSp D) (v : GlobalTheory.Unf.Vol V) (U U' : Subgroup (GA D V)), U ≤ U' → U' ≤ GfC V → IsCompact (U' : Set (GA D V)) → IsOpen ((U.subgroupOf (GfC V) : Subgroup (GfC V)) : Set (GfC V)) → volLev v U ∗ (U.relIndex U' : ℝ) = volLev v U'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanMajorantStabilisersConjugate

theorem **re_form** {n : ℕ} (ε : Fin n → Bool) (x : Fin n → ℂ) : (star x ⬝ᵥ (sgnDiag ℂ ε *ᵥ x)).re = ∑ i, (if ε i then 1 else -1) ∗ Complex.normSq (x i)

theorem **card_neg_le** {n : ℕ} (ε ε' : Fin n → Bool) (M : Matrix (Fin n) (Fin n) ℂ) (h : Mᴴ ∗ sgnDiag ℂ ε ∗ M = sgnDiag ℂ ε') : Fintype.card {i // ε' i = false} ≤ Fintype.card {i // ε i = false}

theorem **exists_perm** {n : ℕ} (ε ε' : Fin n → Bool) (h : Fintype.card {i // ε' i = false} = Fintype.card {i // ε i = false}) : ∃ π : Equiv.Perm (Fin n), ∀ i, ε' (π i) = ε i

theorem **cT_mul** (M N : Matrix (Fin n) (Fin n) A) : Formalism.UG.cT s (M ∗ N) = Formalism.UG.cT s N ∗ Formalism.UG.cT s M

theorem **cT_one** : Formalism.UG.cT s (1 : Matrix (Fin n) (Fin n) A) = 1

theorem **congr_inv** {J D : Matrix (Fin n) (Fin n) A} (b : GL (Fin n) A) (h : Formalism.UG.cT s b.1 ∗ J ∗ b.1 = D) : Formalism.UG.cT s (b⁻¹).1 ∗ D ∗ (b⁻¹).1 = J

theorem **congr_mul** {J D D' : Matrix (Fin n) (Fin n) A} (b c : GL (Fin n) A) (h : Formalism.UG.cT s b.1 ∗ J ∗ b.1 = D) (h' : Formalism.UG.cT s c.1 ∗ D ∗ c.1 = D') : Formalism.UG.cT s (b ∗ c).1 ∗ J ∗ (b ∗ c).1 = D'

theorem **card_neg_le'** (φ : A →+* ℂ) (hφ : ∀ y, starRingEnd ℂ (φ y) = φ (s y)) (ε ε' : Fin n → Bool) (M : Matrix (Fin n) (Fin n) A) (h : Formalism.UG.cT s M ∗ sgnDiag A ε ∗ M = sgnDiag A ε') : Fintype.card {i // ε' i = false} ≤ Fintype.card {i // ε i = false}

theorem **permGL_diagonal** (π : Equiv.Perm (Fin n)) (d : Fin n → A) : Formalism.UG.cT s (permGL (A

theorem **exists_unitary** (φ : A →+* ℂ) (hφ : ∀ y, starRingEnd ℂ (φ y) = φ (s y)) {J : Matrix (Fin n) (Fin n) A} {ε ε' : Fin n → Bool} (b b' : GL (Fin n) A) (h : Formalism.UG.cT s b.1 ∗ J ∗ b.1 = sgnDiag A ε) (h' : Formalism.UG.cT s b'.1 ∗ J ∗ b'.1 = sgnDiag A ε') : ∃ P ∈ Formalism.UG.UJ s (1 : Matrix (Fin n) (Fin n) A), b' ∗ P ∗ b⁻¹ ∈ Formalism.UG.UJ s J

theorem **stabArch_conj** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (a a' : GL (Fin n) (AF F ⊗[F] E)), IsMajorantFrame σ J a → IsMajorantFrame σ J a' → ∀ w : InfinitePlace F, ∃ g : GArch σ J w, ∀ k : GArch σ J w, k ∈ stabArch σ J w (Formalism.UG.bc (compInf F E w) a') ↔ g⁻¹ ∗ k ∗ g ∈ stabArch σ J w (Formalism.UG.bc (compInf F E w) a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanMajorantStabilisersConjugateStatement

theorem **_root_.GlobalTheory.T.archimedean_majorant_stabilisers_conjugate** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (a a' : GL (Fin n) (AF F ⊗[F] E)), IsMajorantFrame σ J a → IsMajorantFrame σ J a' → ∀ w : InfinitePlace F, ∃ g : GArch σ J w, ∀ k : GArch σ J w, k ∈ stabArch σ J w (Formalism.UG.bc (compInf F E w) a') ↔ g⁻¹ ∗ k ∗ g ∈ stabArch σ J w (Formalism.UG.bc (compInf F E w) a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanParametersAddCharUnique

theorem **finPart_cA_eq** (z : AE D) : finPart D (cA D z) = Algebra.TensorProduct.map (AlgHom.id D.host.F (Af D.host.F)) D.sp.cm.sig (finPart D z)

theorem **exists_real_point** {θ : InfFam D} (hθ : IsInfFam θ) (τ₀ : D.what₂.Tau) (t : ℝ) : ∃ x : Ainf D.host.F, ∀ τ, θ τ (x ⊗ₜ[D.host.F] (1 : D.host.E)) = if τ = τ₀ then (t : ℂ) else 0

theorem **realPar_unique** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ l l' : D.what₂.Tau → ℝ, HasRealPar ψ θ l → HasRealPar ψ θ l' → l = l'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanParametersAddCharUniqueStatement

theorem **_root_.GlobalTheory.T.archimedean_parameters_addChar_unique** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ l l' : D.what₂.Tau → ℝ, HasRealPar ψ θ l → HasRealPar ψ θ l' → l = l'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanThetaCorrespondenceTowersRealData

theorem **archOnRecord_closed** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ D : D7, GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ArchOnRecord D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ArchimedeanThetaCorrespondenceTowersRealDataStatement

theorem **_root_.GlobalTheory.T.archimedeanThetaCorrespondence_towers_real_data** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ D : D7, GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ArchOnRecord D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AtMostOneHolds

theorem **atMostOne_holds** : AtMostOne

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.AtOneInt

theorem **Same.refl** (Q : QStep S) : Same Q Q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BadPrimesSharp

theorem **bad_primes_sharp** : ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → (∀ P ∈ rowSG Pr sel mu, P ∈ Pr.vGs) ∧ (∀ P ∈ Pr.vGs, ∃ i, P ∈ Pr.vG i ∧ Formalism.L2.Col.qOf P = Pr.g i ∧ ((Pr.g i : ℕ) : 𝒪 X.E) ∈ P.asIdeal ∧ ((Pr.g i : ℕ) : 𝒪 X.E) ∉ P.asIdeal ^ 2 ∧ ((Pr.g (1 - i) : ℕ) : 𝒪 X.E) ∉ P.asIdeal ∧ ((Pr.T u : ℕ) : 𝒪 X.E) ∉ P.asIdeal ∧ Pr.piT u ∉ P.asIdeal) ∧ (∀ P ∈ Pr.vT u, ((Pr.T u : ℕ) : 𝒪 X.E) ∈ P.asIdeal) ∧ (∏ P : rowSG Pr sel mu, Formal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainCommutationValid

theorem **levAt_is** {D : D7} {T : Type} {L : LevCh D T} {t : T} (h : L.Is t) : (levAt L t).Is ()

theorem **chrEq_mu_base** {D : D7} {T : Type} {L : LevCh D T} {t : T} (h : L.Is t) (j : Fin D.R.m) : ChrEq (L.mu t ((j : ℕ) + 1)) (L.C.mu0 j)

theorem **com_is** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → (GlobalTheory.BaseChain.com W).Is

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainCommutationValidStatement

theorem **_root_.GlobalTheory.T.base_chain_commutation_valid** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → (GlobalTheory.BaseChain.com W).Is

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStageXValid

theorem **cH_ne_zero** (W : RealPack ex) : cH W ≠ 0

theorem **cH_ne_top** (W : RealPack ex) : cH W ≠ ⊤

theorem **Q_base_empty** (W : RealPack ex) : (pk W).Q W.P.A3.A.μ0 = ∅

theorem **stageX_is** (W : RealPack ex) (_ : PkHyp W) : (stageX W).Is (com W) D.R.m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStageXValidStatement

theorem **_root_.GlobalTheory.T.base_chain_stage_X_valid** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → (GlobalTheory.BaseChain.stageX W).Is (GlobalTheory.BaseChain.com W) D.R.m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStageYValid

theorem **under_mem_S** {W : RealPack ex} (hL : (pk W).toLevCh.Is W.P.A3.A.μ0) (k : Fin 2) (P : Pl D.host.E) (hk : ((D.host.g k : ℕ) : 𝒪 D.host.E) ∈ P.asIdeal) : P.under (𝒪 D.host.F) ∈ (com W).S

theorem **stageY_is** (W : RealPack ex) (hP : PkHyp W) (j : ℕ) (hj : j < D.R.m) : (GlobalTheory.BaseChain.stageY W j).Is (GlobalTheory.BaseChain.com W) j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStageYValidStatement

theorem **_root_.GlobalTheory.T.base_chain_stage_Y_valid** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j < D.R.m → (GlobalTheory.BaseChain.stageY W j).Is (GlobalTheory.BaseChain.com W) j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStepThetaFunctions

theorem **rank_Yc_le_succ** (j : ℕ) : (Yc D j).rank ≤ (Yc D (j + 1)).rank

theorem **stepY_thetaFun_isItW** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j, j + 1 < D.R.m → (∀ φ, IsThetaFunO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W j).V (GlobalTheory.BaseChain.stageY W (j + 1)).V

((GlobalTheory.BaseChain.stepY f j).pt φ)) ∧ IsItW (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W j).M (GlobalTheory.BaseChain.stageY W (j + 1)).M (GlobalTheory.BaseChain.stepY f j).ω (GlobalTheory.BaseChain.stepY f j).pt (GlobalTheory.BaseChain.stepY f j).It

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStepThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.base_chain_step_theta_functions** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j, j + 1 < D.R.m → (∀ φ, IsThetaFunO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W j).V (GlobalTheory.BaseChain.stageY W (j + 1)).V ((GlobalTheory.BaseChain.stepY f j).pt φ)) ∧ IsItW (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W j).M (GlobalTheory.BaseChain.stageY W (j + 1)).M (GlobalTheory.BaseChain.stepY f j).ω (GlobalTheory.BaseChain.stepY f j).pt (GlobalTheory.BaseChain.stepY f j).It

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStepThetaLift

theorem **GlobalTheory.BaseChain.thetaKO_sum_smul_act** {D : D7} {S T : HSp D} {ι : Type} (s : Finset ι) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)) (g : GA D S) (h : GA D T) (c : ι → ℂ) (κ : ι → GA D S) (f : WO S T → ℂ) : thetaKO om g h (∑ p ∈ s, c p • om (κ p, 1) f) = ∑ p ∈ s, c p * thetaKO om (g * κ p) h f

theorem **GlobalTheory.BaseChain.stepY_integrable_and_eq** {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (hW : PkHyp W) (t) (j) (hj : j + 1 < D.R.m) (F : CP (pk W) (lvP W) t (j + 1)) (hF : F.1 ∈ (pk W).CC t (j + 1)) (v) (hv : IsStepVecOf (pk W).ψ (pk W).θ (Yc D j) (Yc D (j + 1)) (cart ((pk W).MY j)) (cart ((pk W).MY (j + 1))) ((pk W).ΛY j) ((pk W).ΛY (j + 1)) ((pk W).βY j) ((pk W).βY (j + 1)) ((pk W).datYt t j (lvP W)) ((pk W).chi (j + 1)) ((pk W).Q t) ((pk W).xiR t) ((pk W).vac j) v) (Q x g) : IntegrableOn (fun h => thetaKO ((pk W).ωY t j) g h (pureTensorO (pk W).ψ (pk W).θ (Yc D j) (Yc D (j + 1)) (cart ((pk W).MY j)) (cart ((pk W).MY (j + 1))) (onb ((pk W).MY j)) (onb ((pk W).MY (j + 1))) v Q) *

theorem **GlobalTheory.BaseChain.stepY_isIo** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j, j + 1 < D.R.m → Lift.IsIo (GlobalTheory.BaseChain.stageY W (j + 1)).ν (GlobalTheory.BaseChain.stageY W (j + 1)).fd (thetaKO (GlobalTheory.BaseChain.stepY f j).ω) (GlobalTheory.BaseChain.stepY f j).pt (GlobalTheory.BaseChain.stepY f j).It ((GlobalTheory.BaseChain.stageY W (j + 1)).CC (GlobalTheory.BaseChain.com W)) ((GlobalTheory.BaseChain.stageY W j).CC (GlobalTheory.BaseChain.com W)) (GlobalTheory.BaseChain.stepY f j).io

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStepThetaLiftStatement

theorem **_root_.GlobalTheory.T.base_chain_step_theta_lift** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j, j + 1 < D.R.m → Lift.IsIo (GlobalTheory.BaseChain.stageY W (j + 1)).ν (GlobalTheory.BaseChain.stageY W (j + 1)).fd (thetaKO (GlobalTheory.BaseChain.stepY f j).ω) (GlobalTheory.BaseChain.stepY f j).pt (GlobalTheory.BaseChain.stepY f j).It ((GlobalTheory.BaseChain.stageY W (j + 1)).CC (GlobalTheory.BaseChain.com W)) ((GlobalTheory.BaseChain.stageY W j).CC (GlobalTheory.BaseChain.com W)) (GlobalTheory.BaseChain.stepY f j).io

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStepWeilDatum

theorem **stepY_weil_vec** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j, j + 1 < D.R.m → IsWeilDatumO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W j).V (GlobalTheory.BaseChain.stageY W (j + 1)).V ((GlobalTheory.BaseChain.com W).chS (j + 1)) ((GlobalTheory.BaseChain.com W).chT (j + 1)) (GlobalTheory.BaseChain.stepY f j).ω ∧ ∃ ξ f₀, IsStepVecOf (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W j).V (GlobalTheory.BaseChain.stageY W (j + 1)).V (cart (GlobalTheory.BaseChain.stageY W j).M) (cart (GlobalTheory.BaseChain.stageY W (j + 1)).M) (GlobalTheory.BaseChain.stageY

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainStepWeilDatumStatement

theorem **_root_.GlobalTheory.T.base_chain_step_weil_datum** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j, j + 1 < D.R.m → IsWeilDatumO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W j).V (GlobalTheory.BaseChain.stageY W (j + 1)).V ((GlobalTheory.BaseChain.com W).chS (j + 1)) ((GlobalTheory.BaseChain.com W).chT (j + 1)) (GlobalTheory.BaseChain.stepY f j).ω ∧ ∃ ξ f₀, IsStepVecOf (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W j).V (GlobalTheory.BaseChain.stageY W (j + 1)).V (cart (GlobalTheory.BaseChain.stageY W j).M) (cart (GlobalTheory.BaseChain.stageY W (j + 1)).M).

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainTopThetaFunctions

theorem **comb_pureTensorO** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (sS sT BS BT) {e : WO S T →₀ ℂ} (he : IsFinComb e) {f₀ : WO S T → ℂ} (hf₀ : IsThetaFunO ψ S T f₀) (P) : comb ψ S T e (pureTensorO ψ θ S T sS sT BS BT f₀ P) = pureTensorO ψ θ S T sS sT BS BT (comb ψ S T e f₀) P

theorem **isItW_of_comb** {V W : HSp D} {S S' : D.what₂.Tau → SPar} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {M : DSModel V S} {M' : DSModel W S'} {χS χT : (AE D)ˣ →* Circle} {ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} (hω : IsWeilDatumO ψ V W χS χT ω) (hθ : IsInfFam θ) {f g : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) {b : WO V W →₀ ℂ} (hb : IsFinComb b) (hg : comb ψ V W b f = g) {It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M} (hI : IsItW θ M M' ω (pureTensorO ψ θ V W (cart M) (cart M') (onb M) (onb M') g) It) : IsItW θ M M' ω (pureTensorO ψ θ V W (cart M) (cart M') (onb M) (onb M') f) It

theorem **theta_and_it** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j (h : j + 2 * 0 + 2 = D.R.m), (∀ φ, IsThetaFunO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W (j + 1)).V (GlobalTheory.BaseChain.stageX W).V ((GlobalTheory.BaseChain.stepX f j h).pt φ)) ∧ IsItW (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W (j + 1)).M (GlobalTheory.BaseChain.stageX W).M (GlobalTheory.BaseChain.stepX f j h).ω (GlobalTheory.BaseChain.stepX f j h).pt (GlobalTheory.BaseChain.stepX f j h).It

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainTopThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.base_chain_top_theta_functions** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j (h : j + 2 * 0 + 2 = D.R.m), (∀ φ, IsThetaFunO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W (j + 1)).V (GlobalTheory.BaseChain.stageX W).V ((GlobalTheory.BaseChain.stepX f j h).pt φ)) ∧ IsItW (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W (j + 1)).M (GlobalTheory.BaseChain.stageX W).M (GlobalTheory.BaseChain.stepX f j h).ω (GlobalTheory.BaseChain.stepX f j h).pt (GlobalTheory.BaseChain.stepX f j h).It

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainTopThetaLift

theorem **setIntegral_smul_measure_complex** {H : Type*} [MeasurableSpace H] (c : ENNReal) (ν : Measure H) (s : Set H) (f : H → ℂ) : ∫ y in s, f y ∂(c • ν) = c.toReal • ∫ y in s, f y ∂ν

theorem **stepX_isIo** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j (h : j + 2 * 0 + 2 = D.R.m), Lift.IsIo (GlobalTheory.BaseChain.stageX W).ν (GlobalTheory.BaseChain.stageX W).fd (thetaKO (GlobalTheory.BaseChain.stepX f j h).ω) (GlobalTheory.BaseChain.stepX f j h).pt (GlobalTheory.BaseChain.stepX f j h).It ((GlobalTheory.BaseChain.stageX W).CC (GlobalTheory.BaseChain.com W)) ((GlobalTheory.BaseChain.stageY W (j + 1)).CC (GlobalTheory.BaseChain.com W)) (GlobalTheory.BaseChain.stepX f j h).io

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainTopThetaLiftStatement

theorem **_root_.GlobalTheory.T.base_chain_top_theta_lift** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j (h : j + 2 * 0 + 2 = D.R.m), Lift.IsIo (GlobalTheory.BaseChain.stageX W).ν (GlobalTheory.BaseChain.stageX W).fd (thetaKO (GlobalTheory.BaseChain.stepX f j h).ω) (GlobalTheory.BaseChain.stepX f j h).pt (GlobalTheory.BaseChain.stepX f j h).It ((GlobalTheory.BaseChain.stageX W).CC (GlobalTheory.BaseChain.com W)) ((GlobalTheory.BaseChain.stageY W (j + 1)).CC (GlobalTheory.BaseChain.com W)) (GlobalTheory.BaseChain.stepX f j h).io

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainTopWeilDatum

theorem **stepX_weil_vec** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j (h : j + 2 * 0 + 2 = D.R.m), IsWeilDatumO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W (j + 1)).V (GlobalTheory.BaseChain.stageX W).V ((GlobalTheory.BaseChain.com W).chS (j + 1 + 1)) ((GlobalTheory.BaseChain.com W).chT (j + 1 + 1)) (GlobalTheory.BaseChain.stepX f j h).ω ∧ ∃ ξ f₀, IsStepVecOf (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W (j + 1)).V (GlobalTheory.BaseChain.stageX W).V (cart (GlobalTheory.BaseChain.stageY W (j + 1)).M) (cart (GlobalTheory.BaseChain.stageX W).M) (GlobalTheory.B

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChainTopWeilDatumStatement

theorem **_root_.GlobalTheory.T.base_chain_top_weil_datum** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ j (h : j + 2 * 0 + 2 = D.R.m), IsWeilDatumO (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.stageY W (j + 1)).V (GlobalTheory.BaseChain.stageX W).V ((GlobalTheory.BaseChain.com W).chS (j + 1 + 1)) ((GlobalTheory.BaseChain.com W).chT (j + 1 + 1)) (GlobalTheory.BaseChain.stepX f j h).ω ∧ ∃ ξ f₀, IsStepVecOf (GlobalTheory.BaseChain.com W).ψ (GlobalTheory.BaseChain.com W).θ (GlobalTheory.BaseChain.stageY W (j + 1)).V (GlobalTheory.BaseChain.stageX W).V (cart (GlobalTheory.BaseChain.stageY W (j + 1)).M) (cart (GlobalTheory.BaseC

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChangeTensorConjugationInvolutiveOfInvolution

theorem **baseChangeTensorConjugation_involutive_of_involution** {F E : Type} [Field F] [Field E] [Algebra F E] (R : Type) [CommRing R] [Algebra F R] (σ : E →ₐ[F] E) (hσ : ∀ x, σ (σ x) = x) (e : R ⊗[F] E) : sigR R σ (sigR R σ e) = e

theorem **infinitePlaceComponent_tensorConjugation** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (σ : E →ₐ[F] E) (w : InfinitePlace F) (z : AF F ⊗[F] E) : compInf F E w (sigR (AF F) σ z) = sigR w.Completion σ (compInf F E w z)

theorem **mul_eq_conj_mul_of_invertible_cancel** {R : Type*} [Ring R] {P Pi F X K : R} (h1 : P * Pi = 1) (h : Pi * F * X = (Pi * K * P) * (Pi * F)) : F * X = K * F

theorem **eq_inv_conj_of_intertwine** {R : Type*} [Ring R] {Fi F X K : R} (h1 : Fi * F = 1) (h : F * X = K * F) : X = Fi * K * F

theorem **matHom_chart_conj** (A : HSp D) (μ : D.host.E) (κ : GA D (dblSp A) ≃* G D A.rank) (hκ : IsChart A μ κ) (Fr : GL (Fin (A.rank + A.rank)) (AE D)) (p : GA D (dblSp A)) (hp : mt (dblSp A) p * (rat (Tm A μ) : Mx D (A.rank + A.rank) (A.rank + A.rank)) = Fr.1) (k : GA D (dblSp A)) : GlobalTheory.AdelicU.matHom _ D.sp.cm.sig (Jst D A.rank) (κ (p⁻¹ * k * p)) = Fr⁻¹ * GlobalTheory.AdelicU.matHom (n

theorem **mem_KAd_iff_of_conj** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) (hσ : ∀ x, σ (σ x) = x) {n : ℕ} (Q J : Matrix (Fin (n + n)) (Fin (n + n)) E) (L : Submodule (𝒪 E) (Fin n → E)) (a b : GL (Fin n) (AF F ⊗[F] E)) (hb : IsLatticeFrame L b) (hab : ∀ w, Formalism.UG.bc (compInf F E w) b = Formalism.UG.bc (compInf F E w) a) (M M' : GL (Fin (n + n)) (AF F ⊗[F] E)) (u r : (AF F ⊗[F] E)ˣ) (hM' : ∀ v i j, compFin F E v (M'.1 i j) ∈ intLoc F E v ∧ compFin F E v ((M'⁻¹).1 i j) ∈ intLoc F E v) (hMu : M.1 = (u : AF F ⊗[F] E) • M'.1) (hMM : ∀ w, (Formalism.UG.cT (sigR (AF F) σ) M.1 * M.1).map (compInf F E w) = compInf F E w r • 1) (k : GAd σ Q) (g : GAd σ J) (hg : GlobalTheory.AdelicU.matHom _ σ J g = (blkGL finSumFinEquiv (b, b) * M)⁻¹ * GlobalTheory.AdelicU.matHom _ σ Q k * (blkGL finSumFinEquiv (b, b) * M)) : k ∈ KAd σ Q L a ↔ g ∈ levelAll σ J

theorem **isConj_of_frame** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∀ (a b : GL (Fin A.rank) (AE D)), IsLatticeFrame (frameLat A) b → (∀ w, Formalism.UG.bc (compInf D.host.F D.host.E w) b = Formalism.UG.bc (compInf D.host.F D.host.E w) a) → ∀ (M M' : GL (Fin (A.rank + A.rank)) (AE D)) (u : (AE D)ˣ), IntGL M' → M.1 = (u : AE D) • M'.1 → (∀ w, (adj M.1 * M.1).map (compInf D.host.F D.host.E w) = compInf D.host.F D.host.E w (2 * u * cA D u) • 1) → ∀ p : siegel A, mt (dblSp A) p * (rat (Tm A μ) : Mx D (A.rank + A.rank) (A.rank + A.rank)) = (blkGL finSumFinEquiv (b, b) * M).1 → IsConj A a κ p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChartDecorationLevelPrimeFunctionalEqConjMul

theorem **base_deco_fn** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ i, i ≤ B.R.m → ∀ (β : Ch i) (P : (B.primes X).vT u) (x : Formalism.L2.Col.Sp B.fields.E (B.R.N i)), x ∈ Formalism.L2.Col.stdLat B.fields.E (B.R.N i) → ∀ r : Fin (B.R.N i), r.1 + 1 = B.R.N i → ∀ c : 𝒪 B.fields.E, algebraMap (𝒪 B.fields.E) B.fields.E c = x r → (A.p0 i β).1.1.deco.1.f P x = Ideal.Quotient.mk P.1.asIdeal (B.fields.conjO (X.piT u) ∗ c)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseChartDecorationLevelPrimeFunctionalEqConjMulStatement

theorem **_root_.GlobalTheory.T.baseChartDecoration_levelPrimeFunctional_eq_conj_mul** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ i, i ≤ B.R.m → ∀ (β : Ch i) (P : (B.primes X).vT u) (x : Formalism.L2.Col.Sp B.fields.E (B.R.N i)), x ∈ Formalism.L2.Col.stdLat B.fields.E (B.R.N i) → ∀ r : Fin (B.R.N i), r.1 + 1 = B.R.N i → ∀ c : 𝒪 B.fields.E, algebraMap (𝒪 B.fields.E) B.fields.E c = x r → (A.p0 i β).1.1.deco.1.f P x = Ideal.Quotient.mk P.1.asIdeal (B.fields.conjO (X.piT u) ∗ c)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseLatLineOffT

theorem **baseLat_line_offT** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (hr : 1 ≤ D.R.rS) u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), GlobalTheory.MiTop.PlacedP f B → ∀ v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F), (∀ PP ∈ (pk W).Pg, PP.under (𝒪 D.host.F) ≠ v) → (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal → ∀ u'' : D.what₂.Word, (((D.mem u'').A.T : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal → ∀ c ∈ GlobalTheory.FinPl.idAt D (· = v), (Matrix.mulVec (Formalism.L2.Theta.mt (D.sp.Y 0) (show GA D (D.sp.Y 0) from f.g.b 0)⁻¹) ((fun i => c ∗ Formalism.L2.Theta.incA D (W.S.X.ιE (W.S.X.eR (0, hr) (Fin.cast (GlobalTheory.Core.Mi.rank_eq W.S.X W.Q.instEq) i))))) ∈ Glo

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseLevelDatumLevelEq

theorem **rChi_eq_chi** (W : RealPack ex) {h : GA D (D.sp.Y 0)} (hh : h ∈ rJ W) : ∃ hg : reductionYZero W h ∈ (lvP W).H, rChi W h = (((lvP W).chi (_, hg) : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseMovesVectorZ

theorem **pr_e** (q : SplitPt D k) : q.pr q.e = 1

theorem **e_mul_sec** (q : SplitPt D k) (x : k) : q.e ∗ q.sec x = q.sec x

theorem **sec_mul_sec** (q : SplitPt D k) (x y : k) : q.sec (x ∗ y) = q.sec x ∗ q.sec y

theorem **sec_chg** (q : SplitPt D k) (q' : SplitPt D k') (h : q.e = q'.e) (x : k') : q.sec (chg q q' h x) = q'.sec x

theorem **chg_pr** (q : SplitPt D k) (q' : SplitPt D k') (h : q.e = q'.e) (a : AE D) : chg q q' h (q'.pr a) = q.pr a

theorem **chg_le** (q : SplitPt D k) (q' : SplitPt D k') (h : q.e = q'.e) (x : k') : valuation k (chg q q' h x) ≤ 1 ↔ valuation k' x ≤ 1

theorem **chg_lt** (q : SplitPt D k) (q' : SplitPt D k') (h : q.e = q'.e) (x : k') (hx : valuation k (chg q q' h x) < 1) : valuation k' x < 1

theorem **unif_ratio** (q : SplitPt D k) (q' : SplitPt D k') {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (h : q.e = q'.e) (Ψ) (ψ : AddChar k Circle) (ψ' : AddChar k' Circle) (hψ : (ofSplit q).IsChar Ψ ψ) (hψ' : (ofSplit q').IsChar Ψ ψ') (hc : GlobalTheory.SplitWeil.HasConductor ψ 1) (ϖ : kˣ) (ϖ' : k'ˣ) (hu : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (hu' : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ') (hz : ∀ z : 𝒪 D.host.E, (ψ (((ϖ⁻¹ : kˣ) : k) ∗ (ofSplit q).pr (z : D.host.E)) : ℂ) = ψ' (((ϖ'⁻¹ : k'ˣ) : k') ∗ (ofSplit q').pr (z : D.host.E))) : valuation k ((ϖ : k) ∗ chg q q' h ((ϖ'⁻¹ : k'ˣ) : k') − 1) < 1

theorem **exists_theta** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (t : k) (ht : valuation k (t − 1) < 1) : ∃ (θ : Formalism.IndefR.Lev.EAf D) (hθ : θ ∈ Formalism.L2.Herm.intAf D.host.F D.host.E), Formalism.L2.DIV.resS D.host.F PP ⟨θ, hθ⟩ = 1 ∧ toEAf D q.e ∗ θ = toEAf D (q.sec t)

theorem **pow_sub_one_lt** {t : k} (ht : valuation k (t − 1) < 1) (n : ℕ) : valuation k (t ^ n − 1) < 1

theorem **e_mul_of_prF** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (x y : Formalism.IndefR.Datum.EAf D) (s : k) (h : q.prF he x = s ∗ q.prF he y) : toEAf D q.e ∗ x = toEAf D (q.sec s) ∗ y

theorem **map_inv_mulVec** {R k n : Type} [CommRing R] [CommRing k] [Fintype n] [DecidableEq n] (π : R →+* k) (A : GL n R) (z col : n → R) (s : k) (h : ∀ r, π (z r) = s ∗ π ((A.val *ᵥ col) r)) (r : n) : π (((A⁻¹).val *ᵥ z) r) = s ∗ π (col r)

theorem **edgeSum_elem** {K : Type} [CommRing K] {r κ : ℕ} (a b : Fin r) (h : a ≠ b) (hab : GlobalTheory.chainEdge κ a b) (y : K) [DecidableRel (GlobalTheory.chainEdge κ (r

theorem **isPat_elem** {K : Type} [CommRing K] {r ρ κ : ℕ} (a b : Fin r) (h : a ≠ b) (hab : GlobalTheory.SymRes.FreePos ρ κ a.val b.val) (y : K) : GlobalTheory.SymRes.IsPat ρ κ (elem a b h y)

theorem **exists_char** (W : RealPack ex) (PP : (pk W).Pg) : ∃ χ : 𝒪 D.host.E → ℂ, ∀ u (f : FamCh W u) (v : VCom f.c) (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GRead f G pl → ∀ i, pl i = PP → ∀ z : 𝒪 D.host.E, ((G i).S.ψ ((((G i).S.ϖ⁻¹ : (G i).k) : (G i).k) ∗ (ofSplit (G i).q).pr (z : D.host.E)) : ℂ) = χ z

theorem **sc_even** {c : Comm D} {v : VCom c} (G : GLoc v) (n : ℕ) : G.sc (2 ∗ n) = G.sc 0 ∗ (ofSplit G.q).xi ^ n

theorem **sc_mul_mu** {c : Comm D} {v : VCom c} (G : GLoc v) (μ lam : D.host.E) (hμ : μ ∗ lam = −D.host.xi0 ^ (D.R.m / 2)) (ζ₀ : Fin c.Lz.rank → D.host.E) (hτ : ∀ j h, v.τ j h *ᵥ ζ₀ = lam • (c.bes j 0 h).ζ) (hz : G.z₀ = fun r => (ofSplit G.q).pr (ζ₀ r)) : G.sc 0 ∗ (ofSplit G.q).pr μ = 1

theorem **base_inv_alt** (W : RealPack ex) {u : D.what₂.Word} (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} (v : VCom f.c) (o : TDat f.c k ι) {n : ℕ} (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg) (hG : GRead f G pl) (hB : BotTie f B v o G) (i : Fin n) : ∃ j₀ : Fin (Yc D 0).rank, j₀.val = D.R.k + D.R.rS ∧ ∀ r, toEAf D (G i).q.e ∗ (((Formalism.IndefR.Lev.finE (Yc D 0) (f.g.b 0))⁻¹).val *ᵥ (fun r => algebraMap D.host.E (Formalism.IndefR.Lev.EAf D) (W.S.X.ιE (W.S.X.zR u (Fin.cast (rank_eq W.S.X W.Q.instEq) r))))) r = toEAf D ((G i).q.sec ((((G i).S.ϖ⁻¹ : (G i).kˣ) : (G i).k)) ^ (D.R.rS + 1))) ∗ ((pk W).βY 0).val r j₀

theorem **adelic_embedding_inverse_vector** (W : RealPack ex) (hW : PkHyp W) {u : D.what₂.Word} (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (g : GA D (D.sp.X 0)) : ((Formalism.IndefR.Lev.finE (D.sp.Y 0) (embA B.d B.o g))⁻¹).val *ᵥ (fun r => algebraMap D.host.E (Formalism.IndefR.Lev.EAf D) (W.S.X.ιE (W.S.X.zR u (Fin.cast (rank_eq W.S.X W.Q.instEq) r)))) = fun r => algebraMap D.host.E (Formalism.IndefR.Lev.EAf D) (W.S.X.ιE (W.S.X.zR u (Fin.cast (rank_eq W.S.X W.Q.instEq) r)))

theorem **GlobalTheory.MiTop.base_moves_vector_z** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → 1 ≤ D.R.rS → ∀ PP ∈ (W.Q.Rl.dat0 (W.Q.Rl.toRealDat.levR W.Q.F.lv2 true)).Pg, ∃ Γ : Fin (Yc D 0).rank → Formalism.IndefR.Lev.EAf D, ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ a, IsAt f B a₀ a → ∃ (θ : Formalism.IndefR.Lev.EAf D) (hθ : θ ∈ Formalism.L2.Herm.intAf D.host.F D.host.E), Formalism.L2.DIV.resS D.host.F PP ⟨θ, hθ⟩ = 1 ∧ ∀ ei : AE D, Formalism.L2.NTheta.IsIdemAt ei PP → toEAf D ei • Matrix.mulVec (((Formalism.IndefR.Lev.finE (Yc D 0) (show GA D (Yc D 0) from embA B.d B.o (a₀ * a) * (show GA D (D.sp.Y 0) from f.g.b 0))))⁻¹ : GL (Fin (Yc

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseMovesVectorZStatement

theorem **_root_.GlobalTheory.T.base_moves_vector_z_alt** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → 1 ≤ D.R.rS → ∀ PP ∈ (W.Q.Rl.dat0 (W.Q.Rl.toRealDat.levR W.Q.F.lv2 true)).Pg, ∃ Γ : Fin (Yc D 0).rank → Formalism.IndefR.Lev.EAf D, ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ a, IsAt f B a₀ a → ∃ (θ : Formalism.IndefR.Lev.EAf D) (hθ : θ ∈ Formalism.L2.Herm.intAf D.host.F D.host.E), Formalism.L2.DIV.resS D.host.F PP ⟨θ, hθ⟩ = 1 ∧ ∀ ei : AE D, Formalism.L2.NTheta.IsIdemAt ei PP → toEAf D ei • Matrix.mulVec (((Formalism.IndefR.Lev.finE (Yc D 0) (show GA D (Yc D 0) from embA B.d B.o (a₀ * a) * (show GA D (D.sp.Y 0) from f.g.b 0))))⁻¹ : GL (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BaseTermNormCoords

theorem **base_term_norm_coords** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (χV χW : (AE D)ˣ →* Circle) (om : Om V W) (om' : Om V₀ W), IsWeilDatumO ψ V W χV χW om → IsWeilDatumO ψ V₀ W χV χW om' → ∀ (vS : VolM D W.rank ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (vW : Vol W) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (G : GA D W → ℂ) (Φ : WO V W → ℂ), RapidDec W G → IsThetaFunO ψ V W Φ → ∀ (𝒜 : Set (flagN d)) (ℬ : Set (GA D W)), (∀ f : GA D V → ENNReal, Measurable f → (∀ c, ∀ n ∈ flagN d, f (d.nC c * n) = f n) → ∫⁻ n in 𝒜, f n.1 ∂vN.ν = ∫⁻ u, ∫⁻ b in vb.ℱ, f (d.nB b * d.levi u.1) ∂vb.μ ∂vU.ν) → ∀ g : GA D V, ∫⁻ h in ℬ, ‖G h‖ₑ * ∫⁻ u, ∫⁻ b in vb.ℱ, ‖d.mixS W ψ vS.μ vS.ℱ (om (g, h) Φ) (rat x₀ · u.1.1) (rat x₀ · b)‖ₑ ∂vb.μ ∂vU.ν ∂vW.μ = ∫⁻ p in 𝒜 ×ˢ ℬ, ‖G p.2 * term ψ d W vS (om (p.1.1 * g, p.2) Φ) x₀‖ₑ ∂(vN.ν.prod vW.μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BcPrVMemLevelVLOfMoves

theorem **prV_toEAf_eq_swap** (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (a : AE D) : Formalism.UG.prV (E

theorem **bc_prV_mem_levelVL_of_moves** : ∀ (D : D7) (S : HSp D) (n : ℕ) (hn : S.rank = n) (L : GlobalTheory.FinPl.Lat S) (Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)), L.FG → (∀ x : Fin n → D.host.E, x ∈ Λ ↔ (fun i => x (finCongr hn i)) ∈ L) → ∀ (κ : GA D S) (g : GL (Fin n) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F)), GlobalTheory.AdForm.finM S hn κ = g.1 → ∀ v : HeightOneSpectrum (𝒪 D.host.F), GlobalTheory.FinPl.Moves S κ L L (· = v) → Formalism.UG.bc (Formalism.UG.prV (E

theorem **levelVL_eq_of_std** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) {Lb Λ : Submodule (𝒪 E) (Fin n → E)} {T : ℕ} {v : HeightOneSpectrum (𝒪 F)} (hT : (T : 𝒪 F) ∉ v.asIdeal) (h1 : ∀ x ∈ Lb, ∀ j, ∃ r : 𝒪 E, x j = (r : E)) (h2 : ∀ j, (Pi.single j (T : E) : Fin n → E) ∈ Lb) (h1' : ∀ x ∈ Λ, ∀ j, ∃ r : 𝒪 E, x j = (r : E)) (h2' : ∀ j, (Pi.single j (T : E) : Fin n → E) ∈ Λ) : Formalism.UG.levelVL σ J v Lb = Formalism.UG.levelVL σ J v Λ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BeforeLastStepDatumTransportBadPrime

theorem **intAt_iff_val** (P : HeightOneSpectrum (𝒪 E)) (a : E) : IntAt P a ↔ P.valuation E a ≤ 1

theorem **intAt_inv_pow_mul** {P : HeightOneSpectrum (𝒪 E)} {q B c : ℕ} (hq : (q : E) ≠ 0) {z : E} (hz : P.valuation E z ≤ P.valuation E (((q ^ B * c : ℕ) : ℕ) : E)) : IntAt P (((q : E) ^ B)⁻¹ * z)

theorem **GlobalTheory.Transport.TwistCpl.psiAt_transport_tw** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (hm : i + 2 = B.R.m) (al : (I.idx a).β i) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup a)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u (i)) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (i + 1)) (k y : TC.GA B (B.R.N (i))) (k' y' : TC.GA B (B.R.N (i + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1)) → TC.PtOf (y * k) q.1.1 → TC.PtOf (y' * k') q'.1.1 → cosetIn q' = cosetIn p' → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → Formalism.L2.Col.PsiAt I.Ch I.sel (I.tup a) (I.cst a) i P ((I.i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BeforeLastStepDatumTransportBadPrimeStatement

theorem **_root_.GlobalTheory.T.beforeLastStep_datum_transport_badPrime** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (hm : i + 2 = B.R.m) (al : (I.idx a).β i) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup a)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u (i)) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (i + 1)) (k y : TC.GA B (B.R.N (i))) (k' y' : TC.GA B (B.R.N (i + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1)) → TC.PtOf (y * k) q.1.1 → TC.PtOf (y' * k') q'.1.1 → cosetIn q' = cosetIn p' → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → Formalism.L2.Col.PsiAt I.Ch I.sel (I.tup a) (I.cst a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselAdaptedTranslateLevel

theorem **unitaryAdelicPointFiniteMatrix_inv_eq_of_mul_eq_one** {D : D7} (a : GA D (D.sp.X 0)) (M : Matrix (Fin (D.R.N 0)) (Fin (D.R.N 0)) (GlobalTheory.AdForm.EF D)) (hM : M * GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) a = 1) : GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) a⁻¹ = M

theorem **col_level_match** : ∀ (D : D7) (u : D.what₂.Word) (F₀ : ∀ u, Formalism.L2.Forms.Forms0 D u) (S : STrue D F₀) (d : GlobalTheory.SeeSaw.Dat D u) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), GlobalTheory.Core.Mi.IdE S.X → d.Base → LevX (F₀ u) d h0 → ∀ (B : SBes S.X S.inst u d.c.ψ (D.sp.X 0)), B.lev = d.B 0 D.R.rS h0 → B.Adapted (S.tr u).wt → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (S.tr u) a₀ → ∀ b ∈ GfC (D.sp.X 0), ((d.t 0)⁻¹ * embA (d.bes 0 D.R.rS h0).d (d.bes 0 D.R.rS h0).o b * d.t 0 ∈ d.lev 0 ↔ ((iotaX S.X S.inst).symm (1, S.X.xu u))⁻¹ * embA B.d B.o (a₀ * b * a₀⁻¹) * ((iotaX S.X S.inst).symm (1, S.X.xu u)) ∈ rowJ S.X S.inst) ∧ ((d.t 0)⁻¹ * embA (d.bes 0 D.R.rS h0).d (d.bes 0 D.R.rS h0).o b * d.t 0 ∈ d.lev 0 → rowChi S.X S.inst (((iotaX S.X S.inst).symm (1, S.X.xu u))⁻¹ * embA B.d B.o (a₀ * b * a₀⁻¹) * ((iotaX S.X S.inst).symm (1, S.X.xu u))) = levChi (d.st 0).V (d.st 0).β (d.st

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselAdaptedTranslateLevelStatement

theorem **_root_.GlobalTheory.T.bessel_adapted_translate_level** : ∀ (D : D7) (u : D.what₂.Word) (F₀ : ∀ u, Formalism.L2.Forms.Forms0 D u) (S : STrue D F₀) (d : GlobalTheory.SeeSaw.Dat D u) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), GlobalTheory.Core.Mi.IdE S.X → d.Base → LevX (F₀ u) d h0 → ∀ (B : SBes S.X S.inst u d.c.ψ (D.sp.X 0)), B.lev = d.B 0 D.R.rS h0 → B.Adapted (S.tr u).wt → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (S.tr u) a₀ → ∀ b ∈ GfC (D.sp.X 0), ((d.t 0)⁻¹ * embA (d.bes 0 D.R.rS h0).d (d.bes 0 D.R.rS h0).o b * d.t 0

∈ d.lev 0 ↔ ((iotaX S.X S.inst).symm (1, S.X.xu u))⁻¹ * embA B.d B.o (a₀ * b * a₀⁻¹) * ((iotaX S.X S.inst).symm (1, S.X.xu u)) ∈ rowJ S.X S.inst) ∧ ((d.t 0)⁻¹ * embA (d.bes 0 D.R.rS h0).d (d.bes 0 D.R.rS h0).o b * d.t 0 ∈ d.lev 0 → rowChi S.X S.inst (((iotaX S.X S.inst).symm (1, S.X.xu u))⁻¹ * embA B.d B.o (a₀ * b * a₀⁻¹) * ((iotaX S.X S.inst).symm (1, S.X.xu u)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselCharacterLocalElementEqSimpleEntrySum

theorem **psiE_ι** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (h : p.IsChar ψ ψk) (y : k) : psiE ψ (p.ι y) = ψk (y / p.xi)

theorem **psiE_cA** (ψ : AddChar (AF D.host.F) ℂ) (x : AE D) : psiE ψ (cA D x) = psiE ψ x

theorem **psiE_supp** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (h : p.IsChar ψ ψk) {u : AE D} (hu : (p.e + cA D p.e) * u = u) : psiE ψ u = ψk ((p.π u + p.π (cA D u)) / p.xi)

theorem **hA_supp** {V : HSp D} {e : AE D} {g : GA D V} (hg : SuppAt e V g) {x y : Fin V.rank → D.host.E} (h : hQ V x y = 0) : (e + cA D e) * hA V g x y = hA V g x y

theorem **π_hA** (V : HSp D) (g : GA D V) (x y : Fin V.rank → D.host.E) : p.π (hA V g x y) = (Witness.prH p ∘ (star y ᵥ* V.gram)) ⬝ᵥ (mt V g).map p.π *ᵥ (Witness.prH p ∘ x)

theorem **map_mt_inv** {V : HSp D} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k V.rank Y) {g : GA D V} {q : GL Y k} (h : (mt V g).map p.π = F.β * q.val * F.γ) : (mt V g⁻¹).map p.π = F.β * (q⁻¹).val * F.γ

theorem **col_ne_zero** {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k N Y) (j : Y) {t : k} {x : Fin N → k} (h : ∀ r, F.β r j = t * x r) : t ≠ 0

theorem **π_coef** {g : GA D V} {q : GL (n ⊕ Fin c) k} (h : (mt V g).map p.π = F.β * q.val * F.γ) (i j : Fin c) : t j * p.π (hA V g (a j) (a (c - 1 - i))) = p.pr (κ i) * t i * q.val (.inr i) (.inr j)

theorem **π_simple** {g : GA D V} {q : GL (n ⊕ Fin c) k} (h : (mt V g).map p.π = F.β * q.val * F.γ) {i : ℕ} (hi : i + 1 < c) {x : k} (hx : p.pr (κ i) * t i = x * t (i + 1)) : p.π (hA V g (a (i + 1)) (a (c - 1 - i))) = x * ent q i

theorem **scal_lo** {ρ : ℕ} (B : BesLev D ρ) (s : k) {i : ℕ} (hi : i < ρ) : p.pr (if i = ρ then hQ B.L B.ζ B.ζ else 1) * scal p B s i = p.xi * scal p B s (i + 1)

theorem **scal_hi** {ρ : ℕ} (B : BesLev D ρ) (s : k) {i : ℕ} (hi : ρ ≤ i) (hi' : i < 2 * ρ) : p.pr (if i = ρ then hQ B.L B.ζ B.ζ else 1) * scal p B s i = -p.xi * scal p B s (i + 1)

theorem **col_eq** {ρ : ℕ} {B : BesLev D ρ} {s : k} {FA : Frame k B.A.rank n} {FV : Frame k B.V.rank (n ⊕ Fin c)} (h : p.IsBesFrame B s FA FV) (i : Fin c) (r) : FV.β r (Sum.inr i) = scal p B s i * p.pr (besselAuxiliaryVectors B.d (vecB B.o B.ζ) i r)

theorem **psiBes_isLoc** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (B : BesLev D σ) (s : k) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (FA : Frame k B.A.rank n) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA FV → ∀ (g : GA D B.V) (q : GL (n ⊕ Fin c) k), q ∈ N k n c (σ + 1) → p.IsLoc FV g q → psiBes ψ B.d 1 (vecB B.o B.ζ) g = (ψk (GlobalTheory.RootEx.simpleSum q) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselCharacterLocalElementEqSimpleEntrySumStatement

theorem **_root_.GlobalTheory.T.besselCharacter_localElement_eq_simpleEntrySum** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (B : BesLev D σ) (s : k) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (FA : Frame k B.A.rank n) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA FV → ∀ (g : GA D B.V) (q : GL (n ⊕ Fin c) k), q ∈ N k n c (σ + 1) → p.IsLoc FV g q → psiBes ψ B.d 1 (vecB B.o B.ζ) g = (ψk (GlobalTheory.RootEx.simpleSum q) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnFormCentralTranslation

theorem **mem_GfC_iff** (V : HSp D) (g : GA D V) : g ∈ GfC V ↔ (Formalism.L2.Theta.mt V g).map (finf D) = 1

theorem **map_sand** {k n : ℕ} (x : Mx D k n) (m : Mx D n n) (y : Mx D n k) (hm : m.map (finf D) = 1) : (x ⬝ m ⬝ y).map (finf D) = (x ⬝ y).map (finf D)

theorem **orth_emb_mem_GfC** {S A B : HSp D} (o : OrthDec S A B) {a : GA D A} (ha : a ∈ GfC A) : o.emb (a, 1) ∈ GfC S

theorem **witt_embedding_mem_finite_unitary_points** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {g : GA D S₀} (hg : g ∈ GfC S₀) : d.emb0 g ∈ GfC S

theorem **embA_mem_GfC** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) {a : GA D A} (ha : a ∈ GfC A) : embA d o a ∈ GfC S

theorem **castGA_fst** {I I' : AutInst} (e : I = I') (x : I.GA) : (castGA e x).1 = 1 ↔ x.1 = 1

theorem **iotaX_fst** {X : HidaSetting D} (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) {k : GA D (D.sp.Y 0)} (hk : k ∈ GfC (D.sp.Y 0)) : (iotaX X h k).1 = 1

theorem **toH_mul** (B : SBes X h u ψ A) (a b : GA D A) : B.toH (a * b) = B.toH a * B.toH b

theorem **toH_fst** (B : SBes X h u ψ A) {a : GA D A} (ha : a ∈ GfC A) : (B.toH a).1 = 1

theorem **toH_surj** (B : SBes X h u ψ A) (k : Hfin X u) : ∃ a : GA D A, B.toH a = (1, k)

theorem **toH_snd_comm** (B : SBes X h u ψ A) {a : GA D A} (hc : a ∈ Subgroup.center (GA D A)) (k : Hfin X u) : (B.toH a).2 * k = k * (B.toH a).2

theorem **rtr_mem_Cu** (hK : X.LevelSubgroupCharacter) {δ : Hfin X u} (hδ : ∀ k : Hfin X u, δ * k = k * δ) {T : X.Vpi u →ₗ[ℂ] (Hinf X u × Hfin X u → ℂ)} (hT : T ∈ Cu X u hK) : Formalism.L2.SG.ColDict.rtr X δ T ∈ Cu X u hK

theorem **rtr_inv_rtr** (δ : Hfin X u) (T : X.Vpi u →ₗ[ℂ] (Hinf X u × Hfin X u → ℂ)) : Formalism.L2.SG.ColDict.rtr X δ⁻¹ (Formalism.L2.SG.ColDict.rtr X δ T) = T

theorem **GlobalTheory.Core.Mi.central_acts_on_columns** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (a : GA D A), GlobalTheory.Core.Mi.IdE X → a ∈ GfC A → a ∈ Subgroup.center (GA D A) → ∃ R : Cu X u hK →ₗ[ℂ] Cu X u hK, Function.Injective R ∧ ∀ (T : Cu X u hK) ξ, B.colForm (R T).1 ξ = rTrans a (B.colForm T.1 ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnFormCentralTranslationStatement

theorem **_root_.GlobalTheory.T.bessel_column_form_central_translation** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (a : GA D A), GlobalTheory.Core.Mi.IdE X → a ∈ GfC A → a ∈ Subgroup.center (GA D A) → ∃ R : Cu X u hK →ₗ[ℂ] Cu X u hK, Function.Injective R ∧ ∀ (T : Cu X u hK) ξ, B.colForm (R T).1 ξ = rTrans a (B.colForm T.1 ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnFormIsometry

theorem **GlobalTheory.MiTop.central_isom_ramified** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → ∀ u (d : BotDat W u) z, d.Pin z → ∃ R : D.what₂.Mu u ≃ₗᵢ[ℂ] D.what₂.Mu u, ∀ φ C C₀, IsCol W u C (R φ) → IsCol W u C₀ φ → ∀ ξ, d.B.colForm C₀.1 ξ = rTrans d.a⁻¹ (d.B.colForm C.1 ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnFormIsometryStatement

theorem **_root_.GlobalTheory.T.bessel_column_form_isometry** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → ∀ u (d : BotDat W u) z, d.Pin z → ∃ R : D.what₂.Mu u ≃ₗᵢ[ℂ] D.what₂.Mu u, ∀ φ C C₀, IsCol W u C (R φ) → IsCol W u C₀ φ → ∀ ξ, d.B.colForm C₀.1 ξ = rTrans d.a⁻¹ (d.B.colForm C.1 ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnFunctionContinuousOnBesselGroup

theorem **psiBes_mul_embA** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) (θ : D.host.E) (ζ : Fin L.rank → D.host.E) (n : GA D S) (a : GA D A) : psiBes ψ d θ (vecB o ζ) (n ∗ embA d o a) = psiBes ψ d θ (vecB o ζ) n

theorem **ph_continuousOn** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK) (y : X.B.I.Ginf), IsCharAF D.host.F ψ → ContinuousOn (fun g => Ph X u hK T (iotaX X h g ∗ (y, X.xu u))) (besGrp B.d B.o : Set (GA D (D.sp.Y 0)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnFunctionContinuousOnBesselGroupStatement

theorem **_root_.GlobalTheory.T.bessel_column_function_continuousOn_bessel_group** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK) (y : X.B.I.Ginf), IsCharAF D.host.F ψ → ContinuousOn (fun g => Ph X u hK T (iotaX X h g ∗ (y, X.xu u))) (besGrp B.d B.o : Set (GA D (D.sp.Y 0)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnScalarFormNormBound

theorem **colForm_bound** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK), ∃ C : ℝ, ∀ (ξ : X.Vpi u) (a : GA D A), ‖B.colForm T.1 ξ a‖ ≤ C ∗ ‖ξ‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselColumnScalarFormNormBoundStatement

theorem **_root_.GlobalTheory.T.bessel_column_scalar_form_norm_bound** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK), ∃ C : ℝ, ∀ (ξ : X.Vpi u) (a : GA D A), ‖B.colForm T.1 ξ a‖ ≤ C ∗ ‖ξ‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselDoubleStepLinearInSecondDatum

theorem **hand0_smul_snd_of_bessel** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (φ : GA D B.A → ℂ) (Ξ₁ : WO B.V₀ J.W₀ → ℂ) (Ξ₂ : WO J.W₀ B'.V₀ → ℂ) (c : ℂ), hand0 s₁ s₂ l vL φ Ξ₁ (c • Ξ₂) = c • hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselDoubleStepLinearInSecondDatumStatement

theorem **_root_.GlobalTheory.T.besselDoubleStep_linear_in_secondDatum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (φ : GA D B.A → ℂ) (Ξ₁ : WO B.V₀ J.W₀ → ℂ) (Ξ₂ : WO J.W₀ B'.V₀ → ℂ) (c : ℂ), hand0 s₁ s₂ l vL φ Ξ₁ (c • Ξ₂) = c • hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselDoubleStepUnfoldingOfUnfoldingsAndLink

theorem **om_comp_one** {D : D7} {S T : HSp D} (om : Om S T) (g : GA D S) (h : GA D T) (f : WO S T → ℂ) : om (1, h) (om (g, 1) f) = om (g, h) f

theorem **rTrans_liftS** {D : D7} {S T : HSp D} (om : Om S T) (ν : Measure (GA D T)) (ℱ : Set (GA D T)) (G : GA D T → ℂ) (f : WO S T → ℂ) (g : GA D S) : rTrans g (liftS om ν ℱ G f) = liftS om ν ℱ G (om (g, 1) f)

theorem **doubleStep_of_laws** : ∀ D : D7, UnfBF D → UnfFB D → LinkLaw D → HandFBLaw D → FlatLaw D → DoubleStep D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselDoubleStepUnfoldingOfUnfoldingsAndLinkStatement

theorem **_root_.GlobalTheory.T.besselDoubleStepUnfolding_of_unfoldings_and_link** : ∀ D : D7, UnfBF D → UnfFB D → LinkLaw D → HandFBLaw D → FlatLaw D → DoubleStep D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFlatMapLocalUnipotentTranslationLaw

theorem **lo_hi_bij** {ρ c : ℕ} (h : c = 2 ∗ ρ) : Function.Bijective (Sum.elim (lo h) (hi h))

theorem **sum_lo_hi** {M : Type} [AddCommMonoid M] {ρ c : ℕ} (h : c = 2 ∗ ρ) (f : Fin c → M) : ∑ l, f l = ∑ i, f (lo h i) + ∑ i, f (hi h i)

theorem **frame_mul** {k : Type} [Field k] {m ρ c : ℕ} {n : Type} [Fintype n] [DecidableEq n] (h : c = 2 ∗ ρ) (ξ s t : k) (hξ : ξ ≠ 0) (M : Matrix (Fin m) (Fin m) k) (E E' : Matrix (Fin m) (Fin ρ) k) (β : Matrix (Fin m) (n ⊕ Fin c) k) (u : Matrix (Fin c) (Fin c) k) (hu : ∀ l l', l' < l → u l l' = 0) (hb : ∀ l l' : Fin c, l.val < ρ → ρ ≤ l'.val → u l l' = 0) (h0 : ∀ r a, (M ∗ β) r (.inl a) = β r (.inl a)) (hlo : ∀ i r, β r (.inr (lo h i)) = s ∗ ξ ^ (ρ − i.val) ∗ E r i) (hhi : ∀ i r, β r (.inr (hi h i)) = t ∗ (−1 / ξ) ^ (ρ − 1 − i.val) ∗ E' r i) (hE : M ∗ E = E ∗ Matrix.of fun i j => u (lo h i) (lo h j) ∗ ξ ^ (j.val − i.val)) (hE' : M ∗ E' = E' ∗ Matrix.of fun i j => u (hi h i) (hi h j) ∗ (−ξ) ^ (i.val − j.val)) : M ∗ β = β ∗ (fromBlocks 1 0 0 u : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k)

theorem **simpleSum_split** {k : Type} [Field k] {ρ c : ℕ} (n : Type) [Fintype n] [DecidableEq n] (h : c = 2 ∗ ρ) (u : GL (Fin c) k) (hb : ∀ l l' : Fin c, l.val < ρ → ρ ≤ l'.val → u.val l l' = 0) : GlobalTheory.RootEx.simpleSum (GlobalTheory.RootEx.embA n u) = (∑ i : Fin ρ, ∑ j : Fin ρ, if j.val = i.val + 1 then u.val (lo h i) (lo h j) else 0) + ∑ i : Fin ρ, ∑ j : Fin ρ, if j.val = i.val + 1 then u.val (hi h j) (hi h i) else 0

theorem **det_of_inN** {F : Type} [Field F] {n : Type} [Fintype n] [DecidableEq n] {c p : ℕ} {g : GL (n ⊕ Fin c) F} (hg : InN p g) : Matrix.GeneralLinearGroup.det g = 1

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **jacT_levi** (ψ : AddChar (AF D.host.F) ℂ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) (a : GL (Fin ρ) (AE D)) (f : WO S T₀ → ℂ) : jacT ψ d S y (d.levi a) f = f

theorem **grpAt_of_suppAt** {e : AE D} (he : IsSplitIdem D e) {S : HSp D} {g : GA D S} (hg : SuppAt e S g) : g ∈ GlobalTheory.FinPl.grpAt S (fun _ => True)

theorem **exists_lift** {k : Type} [Field k] {ρ : ℕ} (f : k →+ AE D) (α : Matrix (Fin ρ) (Fin ρ) k) (h0 : ∀ i j, j < i → α i j = 0) (h1 : ∀ i, α i i = 1) : ∃ a ∈ UX D ρ, (∀ i j, a.val i j = (1 : Mx D ρ ρ) i j + f ((α − 1) i j)) ∧ upSum a.val = f (∑ i : Fin ρ, ∑ j : Fin ρ, if j.val = i.val + 1 then α i j else 0)

theorem **map_inv_one** {A B : Type} [CommRing A] [CommRing B] {ρ : ℕ} (φ : A →+* B) (a : GL (Fin ρ) A) (h : a.val.map φ = 1) : (a⁻¹).val.map φ = 1

theorem **smul_inv_one** {A : Type} [CommRing A] {ρ : ℕ} (η : A) (a : GL (Fin ρ) A) (h : η • a.val = η • 1) : η • (a⁻¹).val = η • 1

theorem **smul_div_one** {A : Type} [CommRing A] {ρ : ℕ} (η : A) (a b : GL (Fin ρ) A) (ha : η • a.val = η • 1) (hb : η • b.val = η • 1) : η • (a * b⁻¹).val = η • 1

theorem **smul_three** {A : Type} [CommRing A] {ρ m : ℕ} (η : A) (x y : Matrix (Fin m) (Fin ρ) A) (x' y' : Matrix (Fin ρ) (Fin m) A) (z : Matrix (Fin m) (Fin m) A) (a b : Matrix (Fin ρ) (Fin ρ) A) (ha : η • a = η • 1) (hb : η • b = η • 1) (h : x * x' + z + y * y' = 1) : η • (x * a * x' + z + y * b * y') = η • 1

theorem **suppAt_levi** {e : AE D} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (a : GL (Fin ρ) (AE D)) (h : (1 − (e + cA D e)) • a.val = (1 − (e + cA D e)) • 1) : SuppAt e S (d.levi a)

theorem **π_c_ι** (x : k) : p.π (cA D (p.ι x)) = 0

theorem **eta_ι** (x : k) : (1 − (p.e + cA D p.e)) * p.ι x = 0

theorem **eta_c_ι** (x : k) : (1 − (p.e + cA D p.e)) * cA D (p.ι x) = 0

theorem **psiE_ι** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (h : p.IsChar ψ ψk) (x : k) : psiE ψ (p.ι (p.xi * x)) = ψk x

theorem **one_map** (φ : AE D →+* k) {ρ : ℕ} (i j : Fin ρ) : φ ((1 : Mx D ρ ρ) i j) = (1 : Matrix (Fin ρ) (Fin ρ) k) i j

theorem **map_div_one** {A B : Type} [CommRing A] [CommRing B] {ρ : ℕ} (φ : A →+* B) (a b : GL (Fin ρ) A) (α : Matrix (Fin ρ) (Fin ρ) B) (ha : a.val.map φ = α) (hb : b.val.map φ = 1) : (a * b⁻¹).val.map φ = α

theorem **map_mul_of_eq** {A B : Type} [CommRing A] [CommRing B] {m l r : ℕ} (φ : A →+* B) {M : Matrix (Fin m) (Fin m) A} {x : Matrix (Fin m) (Fin l) A} {y : Matrix (Fin m) (Fin r) A} {a : Matrix (Fin r) (Fin l) A} (h : M * x = y * a) : M.map φ * x.map φ = y.map φ * a.map φ

theorem **psiE_cA** (ψ : AddChar (AF D.host.F) ℂ) (x : AE D) : psiE ψ (cA D x) = psiE ψ x

theorem **psiFlag_div** (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {a b : GL (Fin ρ) (AE D)} (ha : a ∈ UX D ρ) (hb : b ∈ UX D ρ) {x y : k} (fa : psiFlag ψ d (d.levi a) = ψk x) (fb : psiFlag ψ d (d.levi b) = ψk (−y)) : psiFlag ψ d (d.levi (a * b⁻¹)) = ψk (x + y)

theorem **levi_comp** {ρ c : ℕ} (J : JacLev D ρ) (s hh : k) {n : Type} [Fintype n] [DecidableEq n] (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)) (hJ : p.IsJacFrame J s hh G₀ FW) (u : GL (Fin c) k) (h0 : ∀ l l', l' < l → u.val l l' = 0) (hb : ∀ l l' : Fin c, l.val < ρ → ρ ≤ l'.val → u.val l l' = 0) (a : GL (Fin ρ) (AE D)) (h1 : a.val.map p.π = Matrix.of fun i j => u.val (lo hJ.1 i) (lo hJ.1 j) * p.xi ^ (j.val − i.val)) (h2 : (a⁻¹).val.map (p.π.comp (cA D)) = (Matrix.of fun i j => u.val (hi hJ.1 i) (hi hJ.1 j) * (−p.xi) ^ (i.val − j.val))ᵀ) : (mt J.W (J.d.levi a)).map p.π = FW.β * (GlobalTheory.RootEx.embA n u).val * FW.γ

theorem **exists_levi** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (hψ : p.IsChar ψ ψk) {ρ c : ℕ} (J : JacLev D ρ) (s hh : k) {n : Type} [Fintype n] [DecidableEq n] (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)) (hJ : p.IsJacFrame J s hh G₀ FW) (u : GL (Fin c) k) (h0 : ∀ l l', l' < l → u.val l l' = 0) (h1 : ∀ l, u.val l l = 1) (hb : ∀ l l' : Fin c, l.val < ρ → ρ ≤ l'.val → u.val l l' = 0) : ∃ a ∈ UX D ρ, p.IsLoc FW (J.d.levi a) (GlobalTheory.RootEx.embA n u) ∧ psiFlag ψ J.d (J.d.levi a) = ψk (GlobalTheory.RootEx.simpleSum (GlobalTheory.RootEx.embA n u))

theorem **flatBF_unip** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ρ : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ u : GL (Fin c) k, GlobalTheory.RootE

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFlatMapLocalUnipotentTranslationLawStatement

theorem **_root_.GlobalTheory.T.besselFlatMap_local_unipotent_translation_law** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsM

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFourierJacobiContractedIsThetaFunEachVariaStatement

theorem **_root_.GlobalTheory.T.besselFourierJacobi_contracted_isThetaFun_each_variable** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (χA χA' χz : ((AE D)ˣ →* Circle)) (oma : Om B.A J.W₀) (omz : Om B.L J.W₀) (omb : Om J.W₀ B'.A) (omz' : Om J.W₀ B'.L), s₁.χV = χA * χz → s₂.χW = χA' * χz → IsWeilDatumO ψ B.A J.W₀ χA s₁.χW oma → IsWeilDatumO ψ B.L J.W₀ χz s₁.χW omz → IsWeilDatumO ψ J.W₀ B'.A s₂.χV χA' omb → IsWeilDatumO ψ J.W₀ B'.L s₂.χV χz omz' → ∀ (Ξ₁ : WO B.V₀ J.W₀ → ℂ) (Ξ₂ : WO J.W₀ B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ J.W₀ Ξ₁ → IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂ → (∀ v, IsThetaFunO ψ B.A J.W₀ fun u => contr l J.W₀ vL Ξ₁ Ξ₂ u v) ∧ ∀ p, (IsThetaFunO ψ J.W₀ B'.A fun v => oma p (fun u => contr l J.W₀ vL Ξ₁ Ξ₂ u v) 0) ∧ ∀ v, oma p (fun u =>

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFourierJacobiContractedIsThetaFunctionEachVariable

theorem **continuous_linkKappa** {D : D7} {ρ : ℕ} {B : BesLev D ρ} {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (S : HSp D) : Continuous (l.κ S)

theorem **integrable_restrict_of_bdd** {X : Type} [TopologicalSpace X] [MeasurableSpace X] [OpensMeasurableSpace X] (μ : Measure X) (s : Set X) (hs : μ s = 1) (f : X → ℂ) (hf : Continuous f) (C : ℝ) (hC : ∀ x, ‖f x‖ ≤ C) : Integrable f (μ.restrict s)

theorem **contr_isTheta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (χA χA' χz : ((AE D)ˣ →* Circle)) (oma : Om B.A J.W₀) (omz : Om B.L J.W₀) (omb : Om J.W₀ B'.A) (omz' : Om J.W₀ B'.L), s₁.χV = χA * χz → s₂.χW = χA' * χz → IsWeilDatumO ψ B.A J.W₀ χA s₁.χW oma → IsWeilDatumO ψ B.L J.W₀ χz s₁.χW omz → IsWeilDatumO ψ J.W₀ B'.A s₂.χV χA' omb → IsWeilDatumO ψ J.W₀ B'.L s₂.χV χz omz' → ∀ (Ξ₁ : WO B.V₀ J.W₀ → ℂ) (Ξ₂ : WO J.W₀ B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ J.W₀ Ξ₁ → IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂ → (∀ v, IsThetaFunO ψ B.A J.W₀ fun u => contr l J.W₀ vL Ξ₁ Ξ₂ u v) ∧ ∀ ρ, (IsThetaFunO ψ J.W₀ B'.A fun v => oma ρ (fun u => contr l J.W₀ vL Ξ₁ Ξ₂ u v) 0) ∧ ∀ v, oma ρ (fun u => contr l J.W₀ vL Ξ₁ Ξ₂ u v) 0 = ∫ w in vL.𝓕, Ξ₂ ((B'.o.resT J.W₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFourierJacobiDoubleUnfoldingBilinear

theorem **besselFourierJacobi_doubleUnfolding_bilinear** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (φ : GA D B.A → ℂ), IsContAut B.A φ → (∀ (Ξ Ξ' : WO B.V₀ J.W₀ → ℂ) (θ : WO J.W₀ B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ J.W₀ Ξ → IsThetaFunO ψ B.V₀ J.W₀ Ξ' → IsThetaFunO ψ J.W₀ B'.V₀ θ → hand0 s₁ s₂ l vL φ (Ξ + Ξ') θ = hand0 s₁ s₂ l vL φ Ξ θ + hand0 s₁ s₂ l vL φ Ξ' θ) ∧ (∀ (c : ℂ) (Ξ : WO B.V₀ J.W₀ → ℂ) (θ : WO J.W₀ B'.V₀ → ℂ), hand0 s₁ s₂ l vL φ (c • Ξ) θ = c • hand0 s₁ s₂ l vL φ Ξ θ) ∧ (∀ (Ξ : WO B.V₀ J.W₀ → ℂ) (θ θ' : WO J.W₀ B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ J.W₀ Ξ → IsThetaFunO ψ J.W₀ B'.V₀ θ → IsThetaFunO ψ J.W₀ B'.V₀ θ' → hand0 s₁ s₂ l vL φ Ξ (θ + θ') = hand0 s₁ s₂ l vL φ Ξ θ + hand0 s₁ s₂ l vL φ Ξ θ') ∧ (∀ (c : ℂ) (Ξ : WO B.V₀ J.W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFourierJacobiNormalFramesExistSplitPlace

theorem **Frame.exists_of_vecMul** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (β : Matrix (Fin N) Y k) (hc : Fintype.card Y = N) (h : ∀ v, v ᵥ* β = 0 → v = 0) : ∃ F : GlobalTheory.LocMod.Frame k N Y, F.β = β

theorem **Frame.card_eq** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k N Y) : Fintype.card Y = N

theorem **Frame.eq_zero** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k N Y) {v : Fin N → k} (h : v ᵥ* F.β = 0) : v = 0

theorem **hyp_eq_zero** {D : D7} {k : Type} [Field k] (f : D.host.E →+* k) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {v : Fin S.rank → k} (h1 : v ᵥ* d.e.map f = 0) (h2 : v ᵥ* d.i₀.map f = 0) (h3 : v ᵥ* d.e'.map f = 0) : v = 0

theorem **orth_eq_zero** {D : D7} {k : Type} [Field k] (f : D.host.E →+* k) {S A L : HSp D} (o : OrthDec S A L) {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) {v : Fin S.rank → k} (h1 : v ᵥ* o.iA.map f = 0) (h2 : v ⬝ᵥ (f ∘ vecB o ζ) = 0) : v = 0

theorem **dot_eq_zero_of_smul** {k : Type} [Field k] {N : ℕ} {v w : Fin N → k} {c : k} (hc : c ≠ 0) (h : v ⬝ᵥ (fun r => c * w r) = 0) : v ⬝ᵥ w = 0

theorem **pr_eq** : p.pr = ⇑(p.π.comp (incA D))

theorem **xi_ne_zero** : p.xi ≠ 0

theorem **hz_ne_zero** (B : BesLev D ρ) : p.hz B ≠ 0

theorem **exists_isCoreFrame** (hs : s ≠ 0) (B : BesLev D ρ) (FA : GlobalTheory.LocMod.Frame k B.A.rank n) : ∃ F, p.IsCoreFrame B s FA F

theorem **exists_isJacFrame** (hs : s ≠ 0) (J : JacLev D ρ) {hh : k} (hh0 : hh ≠ 0) (F₀ : GlobalTheory.LocMod.Frame k J.W₀.rank n) : ∃ F : GlobalTheory.LocMod.Frame k J.W.rank (n ⊕ Fin (2 * ρ)), p.IsJacFrame J s hh F₀ F

theorem **exists_isBesFrame** (hs : s ≠ 0) (B : BesLev D ρ) (FA : GlobalTheory.LocMod.Frame k B.A.rank n) : ∃ F : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin (2 * ρ + 1)), p.IsBesFrame B s FA F

theorem **normalFrames_exist** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (s : k), s ≠ 0 → (∀ (B : BesLev D ρ) (FA : GlobalTheory.LocMod.Frame k B.A.rank n), (∃ F, p.IsCoreFrame B s FA F) ∧ (∃ F : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin (2 * ρ + 1)), p.IsBesFrame B s FA F) ∧ p.hz B ≠ 0) ∧ ∀ (J : JacLev D ρ) (hh : k), hh ≠ 0 → ∀ F₀ : GlobalTheory.LocMod.Frame k J.W₀.rank n, ∃ F : GlobalTheory.LocMod.Frame k J.W.rank (n ⊕ Fin (2 * ρ)), p.IsJacFrame J s hh F₀ F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFourierJacobiNormalFramesExistSplitPlaceStatement

theorem **_root_.GlobalTheory.T.bessel_fourierJacobi_normalFrames_exist_splitPlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (s : k), s ≠ 0 → (∀ (B : BesLev D ρ) (FA : GlobalTheory.LocMod.Frame k B.A.rank n), (∃ F, p.IsCoreFrame B s FA F) ∧ (∃ F : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin (2 * ρ + 1)), p.IsBesFrame B s FA F) ∧ p.hz B ≠ 0) ∧ ∀ (J : JacLev D ρ) (hh : k), hh ≠ 0 → ∀ F₀ : GlobalTheory.LocMod.Frame k J.W₀.rank n, ∃ F : GlobalTheory.LocMod.Frame k J.W.rank (n ⊕ Fin (2 * ρ)), p.IsJacFrame J s hh F₀ F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFourierJacobiUnfoldingHeisenbergExchange

theorem **kappa_mul** {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (g : GA D S) (w : WO S B'.L) : l.κ S (w ⬝ mt S g⁻¹) = mt S g ⬝ l.κ S w

theorem **sympl_resT** {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) (c d : WO S L) : symplOH S T ((o.resT S).symm (0, c)) ((o.resT S).symm (0, d)) = symplOH S L c d

theorem **sympl_resS** {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) (c d : WO L S) : symplOH T S ((o.resS S).symm (0, c)) ((o.resS S).symm (0, d)) = symplOH L S c d

theorem **act_resT** {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) (g : GA D S) (a : GA D A) (c : WO S L) : actO S T g (o.emb (a, 1)) ((o.resT S).symm (0, c)) = (o.resT S).symm (0, c ⬝ mt S g⁻¹)

theorem **act_resS** {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) (g : GA D S) (a : GA D A) (c : WO L S) : actO T S (o.emb (a, 1)) g ((o.resS S).symm (0, c)) = (o.resS S).symm (0, mt S g ⬝ c)

theorem **quasi_resT** {ψ : AddChar (AF D.host.F) ℂ} {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {γ : WO S L} (hγ : γ ∈ ratMx D L.rank S.rank) (v : WO S L) : f ((o.resT S).symm (0, γ + v)) = ψ (-symplOH S L γ v) * f ((o.resT S).symm (0, v))

theorem **quasi_resS** {ψ : AddChar (AF D.host.F) ℂ} {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) {f : WO T S → ℂ} (hf : IsThetaFunO ψ T S f) {γ : WO L S} (hγ : γ ∈ ratMx D S.rank L.rank) (v : WO L S) : f ((o.resS S).symm (0, γ + v)) = ψ (-symplOH L S γ v) * f ((o.resS S).symm (0, v))

theorem **integral_exchange** {ψ : AddChar (AF D.host.F) ℂ} {S L : HSp D} (vL : VolM D L.rank S.rank) (T Y : WO S L → ℂ) (hT : ∀ γ ∈ ratMx D L.rank S.rank, ∀ v, T (γ + v) = ψ (-symplOH S L γ v) * T v) (hY : ∀ γ ∈ ratMx D L.rank S.rank, ∀ v, Y (γ + v) = ψ (symplOH S L γ v) * Y v) (x : WO S L) : ∫ v in vL.𝒻, T v * (ψ (-symplOH S L v x) * Y (v + x)) ∂vL.μ = ∫ v in vL.𝒻, ψ (symplOH S L v (-x)) * T (v + -x) * Y v ∂vL.μ

theorem **besselFourierJacobi_unfolding_heisenberg_exchange** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (φ : GA D B.A → ℂ), IsContAut B.A φ → ∀ (Ξ : WO B.V₀ J.W₀ → ℂ) (θ : WO J.W₀ B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ J.W₀ Ξ → IsThetaFunO ψ J.W₀ B'.V₀ θ → ∀ w₀ : WO J.W₀ B'.L, hand0 s₁ s₂ l vL φ (heisO ψ B.V₀ J.W₀ ((B.o.resS J.W₀).symm (0, l.κ J.W₀ w₀)) Ξ) θ = hand0 s₁ s₂ l vL φ Ξ (heisO ψ J.W₀ B'.V₀ ((B'.o.resT J.W₀).symm (0, -w₀)) θ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFrameEmbeddingLocalElementEqEmbeddedMatrix

theorem **isLoc_embA** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.LocMod.Place D k) (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ) (B : BesLev D ρ) (s : k) (FA : GlobalTheory.LocMod.Frame k B.A.rank n) (F : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA F → ∀ (h : GA D B.A) (γ : GL n k), p.IsLoc FA h γ → p.IsLoc F (embA B.d B.o h) (GlobalTheory.RootEx.embX c γ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselFrameEmbeddingLocalElementEqEmbeddedMatrixStatement

theorem **_root_.GlobalTheory.T.besselFrame_embedding_localElement_eq_embedded_matrix** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.LocMod.Place D k) (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ) (B : BesLev D ρ) (s : k) (FA : GlobalTheory.LocMod.Frame k B.A.rank n) (F : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA F → ∀ (h : GA D B.A) (γ : GL n k), p.IsLoc FA h γ → p.IsLoc F (embA B.d B.o h) (GlobalTheory.RootEx.embX c γ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupActionFourierBesselBasePoint

theorem **snocCols_uni_iff** : ∀ (R : Type) [CommRing R] (n σ : ℕ) (M : Matrix (Fin n) (Fin σ) R) (w : Fin n → R) (Y : Matrix (Fin n) (Fin n) R), (∃ U ∈ GlobalTheory.UniIndexPrime.uniSub (σ + 1) R, Y * Matrix.of (fun k => Fin.snoc (α

theorem **fix_lineVec_emb** : ∀ (D : D7) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ k : GA D S, mt S k *ᵥ ratV (o.iB *ᵥ ζ) = ratV (o.iB *ᵥ ζ) → ∃ h : GA D A, k = o.emb (h, 1)

theorem **fourierBesselBasePoint_leftInverse_rational** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∃ P ∈ ratMx D (σ + 1) B.V.rank, P · x0 B = 1) ∧ x0 B ∈ ratMx D B.V.rank (σ + 1)

theorem **ratV_mulVec_rat** {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) (v : Fin n → D.host.E) : ratV (M *ᵥ v) = rat M *ᵥ ratV v

theorem **mem_actUni_iff** (B : BesLev D σ) (g : GA D B.V) : g ∈ actUni B ↔ ∃ u ∈ UX D σ, ∃ v : Fin σ → AE D, mt B.V g · B.d.jX = B.d.jX · u.1 ∧ mt B.V g *ᵥ (B.d.j0 *ᵥ ratV (vecB B.o B.ζ)) = B.d.j0 *ᵥ ratV (vecB B.o B.ζ) + B.d.jX *ᵥ v

theorem **flagN_le_actUni** (B : BesLev D σ) : flagN B.d ≤ actUni B

theorem **embA_mem_actUni** (B : BesLev D σ) (h : GA D B.A) : embA B.d B.o h ∈ actUni B

theorem **besGrp_le_actUni** (B : BesLev D σ) : besGrp B.d B.o ≤ actUni B

theorem **mem_besGrp_of_actUni** (B : BesLev D σ) (g : GA D B.V) (hg : g ∈ actUni B) : g ∈ besGrp B.d B.o

theorem **besselGroup_action_fourierBesselBasePoint** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∀ g : GA D B.V, g ∈ besGrp B.d B.o ↔ ∃ u ∈ UX D (σ + 1), mt B.V g · x0 B = x0 B · u.1) ∧ (∀ a a' : Mx D (σ + 1) (σ + 1), x0 B · a = x0 B · a' → a = a') ∧ ∀ g ∈ GQ B.V, ∀ u : UX D (σ + 1), mt B.V g · x0 B = x0 B · u.1.1 → u ∈ UXQ D (σ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupActionFourierBesselBasePointStatement

theorem **_root_.GlobalTheory.T.mem_besGrp_iff_mt_mul_x0_eq_x0_mul** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∀ g : GA D B.V, g ∈ besGrp B.d B.o ↔ ∃ u ∈ UX D (σ + 1), mt B.V g · x0 B = x0 B · u.1) ∧ (∀ a a' : Mx D (σ + 1) (σ + 1), x0 B · a = x0 B · a' → a = a') ∧ ∀ g ∈ GQ B.V, ∀ u : UX D (σ + 1), mt B.V g · x0 B = x0 B · u.1.1 → u ∈ UXQ D (σ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupHaarIsMulRightInvariant

theorem **besGrp_isMulRightInvariant** : ∀ (D : D7) (V V₀ A L : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (o : OrthDec V₀ A L) (μ : Measure (besGrp d o)), μ.IsHaarMeasure → μ.IsMulRightInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupHaarIsMulRightInvariantStatement

theorem **_root_.GlobalTheory.T.besselGroup_haar_isMulRightInvariant** : ∀ (D : D7) (V V₀ A L : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (o : OrthDec V₀ A L) (μ : Measure (besGrp d o)), μ.IsHaarMeasure → μ.IsMulRightInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupQuotientFundamentalDomainCocompact

theorem **besGrp_quotient** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ), B.μR.IsHaarMeasure ∧ IsClosed (besGrp B.d B.o : Set (GA D B.V)) ∧ (∃ Ω : Set (GA D B.V), IsCompact Ω ∧ ∀ s ∈ besGrp B.d B.o, ∃ γ ∈ GQ B.V ⊓ besGrp B.d B.o, γ * s ∈ Ω) ∧ ∃ 𝒮 : Set (besGrp B.d B.o), IsFundamentalDomain ((GQ B.V).subgroupOf (besGrp B.d B.o)) 𝒮 B.μR ∧ B.μR 𝒮 < ⊤ ∧ ∀ F : GA D B.V → ℂ, ContinuousOn F (besGrp B.d B.o : Set (GA D B.V)) → (∀ γ ∈ GQ B.V ⊓ besGrp B.d B.o, ∀ s ∈ besGrp B.d B.o, F (γ * s) = F s) → IntegrableOn (fun s : besGrp B.d B.o => F s) 𝒮 B.μR ∧ ∫ s in 𝒮, F s ∂B.μR = ∫ a in B.vA.𝒻, ∫ n in B.vN.𝒩, F (n.1 * embA B.d B.o a) ∂B.vN.ν ∂B.vA.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupQuotientFundamentalDomainCocompactStatement

theorem **_root_.GlobalTheory.T.besselGroup_quotient_fundamentalDomain_cocompact** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ), B.μR.IsHaarMeasure ∧ IsClosed (besGrp B.d B.o : Set (GA D B.V)) ∧ (∃ Ω : Set (GA D B.V), IsCompact Ω ∧ ∀ s ∈ besGrp B.d B.o, ∃ γ ∈ GQ B.V ⊓ besGrp B.d B.o, γ * s ∈ Ω) ∧ ∃ 𝒮 : Set (besGrp B.d B.o), IsFundamentalDomain ((GQ B.V).subgroupOf (besGrp B.d B.o)) 𝒮 B.μR ∧ B.μR 𝒮 < ⊤ ∧ ∀ F : GA D B.V → ℂ, ContinuousOn F (besGrp B.d B.o : Set (GA D B.V)) → (∀ γ ∈ GQ B.V ⊓ besGrp B.d B.o, ∀ s ∈ besGrp B.d B.o, F (γ * s) = F s) → IntegrableOn (fun s : besGrp B.d B.o => F s) 𝒮 B.μR ∧ ∫ s in 𝒮, F s ∂B.μR = ∫ a in B.vA.𝒻, ∫ n in B.vN.𝒩, F (n.1 * embA B.d B.o a) ∂B.vN.ν ∂B.vA.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupSemidirectProductRational

theorem **continuous_embA** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) : Continuous (embA d o)

theorem **unique** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) {n n' : GA D S} (hn : n ∈ flagN d) (hn' : n' ∈ flagN d) {h h' : GA D A} (e : n * embA d o h = n' * embA d o h') : n = n' ∧ h = h'

theorem **besselCompression_unipotent_mul_embedding** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {n : GA D S} (hn : n ∈ flagN d) (k : GA D S₀) : d.p0 · (mt S (n * d.emb0 k) · d.j0) = mt S₀ k

theorem **rat_iff** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) {n : GA D S} (hn : n ∈ flagN d) (h : GA D A) : n * embA d o h ∈ GQ S ↔ n ∈ GQ S ∧ h ∈ GQ A

theorem **besSemi** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), Continuous (embA B.d B.o) ∧ IsClosed (flagN B.d : Set (GA D B.V)) ∧ (∀ n ∈ flagN B.d, ∀ h, embA B.d B.o h * n * (embA B.d B.o h)⁻¹ ∈ flagN B.d) ∧ (∀ n ∈ flagN B.d, ∀ n' ∈ flagN B.d, ∀ h h', n * embA B.d B.o h = n' * embA B.d B.o h' → n = n' ∧ h = h') ∧ ∀ n ∈ flagN B.d, ∀ h, n * embA B.d B.o h ∈ GQ B.V ↔ n ∈ GQ B.V ∧ h ∈ GQ B.A

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupSemidirectProductRationalStatement

theorem **_root_.GlobalTheory.T.besselGroup_semidirectProduct_rational** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), Continuous (embA B.d B.o) ∧ IsClosed (flagN B.d : Set (GA D B.V)) ∧ (∀ n ∈ flagN B.d, ∀ h, embA B.d B.o h * n * (embA B.d B.o h)⁻¹ ∈ flagN B.d) ∧ (∀ n ∈ flagN B.d, ∀ n' ∈ flagN B.d, ∀ h h', n * embA B.d B.o h = n' * embA B.d B.o h' → n = n' ∧ h = h') ∧ ∀ n ∈ flagN B.d, ∀ h, n * embA B.d B.o h ∈ GQ B.V ↔ n ∈ GQ B.V ∧ h ∈ GQ B.A

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupSplitRealFinite

theorem **tens_ext** {F E A B : Type} [Field F] [Field E] [Algebra F E] [CommRing A] [CommRing B] [Algebra F A] [Algebra F B] (z w : (A × B) ⊗[F] E) (h1 : Algebra.TensorProduct.map (AlgHom.fst F A B) (AlgHom.id F E) z = Algebra.TensorProduct.map (AlgHom.fst F A B) (AlgHom.id F E) w) (h2 : Algebra.TensorProduct.map (AlgHom.snd F A B) (AlgHom.id F E) z = Algebra.TensorProduct.map (AlgHom.snd F A B) (AlgHom.id F E) w) : z = w

theorem **entry_eq** {R R₁ R₂ : Type} [CommRing R] [CommRing R₁] [CommRing R₂] (φ : R →+* R₁) (ψ : R →+* R₂) (hinj : ∀ z w, φ z = φ w → ψ z = ψ w → z = w) {n k m : ℕ} {M N K : Matrix (Fin n) (Fin n) R} (hi : M.map φ = N.map φ) (hf : M.map ψ = K.map ψ) (P : Matrix (Fin k) (Fin n) R) (Q : Matrix (Fin n) (Fin m) R) (i : Fin k) (j : Fin m) {c : R} (hN : (P * (N * Q)) i j = c) (hK : (P * (K * Q)) i j = c) : (P * (M * Q)) i j = c

theorem **mat_eq** {R R₁ R₂ : Type} [CommRing R] [CommRing R₁] [CommRing R₂] (φ : R →+* R₁) (ψ : R →+* R₂) (hinj : ∀ z w, φ z = φ w → ψ z = ψ w → z = w) {n k m : ℕ} {M N K : Matrix (Fin n) (Fin n) R} (hi : M.map φ = N.map φ) (hf : M.map ψ = K.map ψ) (P : Matrix (Fin k) (Fin n) R) (Q : Matrix (Fin n) (Fin m) R) {C : Matrix (Fin k) (Fin m) R} (hN : P * (N * Q) = C) (hK : P * (K * Q) = C) : P * (M * Q) = C

theorem **ext_parts** (z w : AE D) (h1 : realPart D z = realPart D w) (h2 : finitePart D z = finitePart D w) : z = w

theorem **map_real_eq** {S : HSp D} {x y : GA D S} (h : infHom S x = infHom S y) : (Formalism.L2.Theta.mt S x).map (realPart D) = (Formalism.L2.Theta.mt S y).map (realPart D)

theorem **map_finite_eq** {S : HSp D} {x y : GA D S} (h : finHom S x = finHom S y) : (Formalism.L2.Theta.mt S x).map (finitePart D) = (Formalism.L2.Theta.mt S y).map (finitePart D)

theorem **mem_besGrp_of_parts** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) {x y z : GA D S} (hi : infHom S x = infHom S y) (hf : finHom S x = finHom S z) (hy : y ∈ besGrp d o) (hz : z ∈ besGrp d o) : x ∈ besGrp d o

theorem **besGrp_split** : ∀ (D : D7) (V V₀ A L : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (o : OrthDec V₀ A L), ∀ a ∈ GinfC V, ∀ b ∈ GfC V, a * b ∈ besGrp d o → a ∈ besGrp d o ∧ b ∈ besGrp d o

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselGroupSplitRealFiniteStatement

theorem **_root_.GlobalTheory.T.besselGroup_split_real_finite** : ∀ (D : D7) (V V₀ A L : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (o : OrthDec V₀ A L), ∀ a ∈ GinfC V, ∀ b ∈ GfC V, a * b ∈ besGrp d o → a ∈ besGrp d o ∧ b ∈ besGrp d o

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLatticeSplitsAlongWittNormalFrames

theorem **mul_mem_iff** {c : k} (hc : valuation k c = 1) (y : k) : c * y ∈ 𝒪[k] ↔ y ∈ 𝒪[k]

theorem **ne_zero_of_val** {x : k} (h : valuation k x = 1) : x ≠ 0

theorem **sec_mem_intAt_iff** (q : SplitPt D k) (c : k) : q.sec c ∈ intAt D (· = q.v) ↔ c ∈ 𝒪[k]

theorem **sec_pr_eq** (q : SplitPt D k) (x : D.host.E) : q.sec ((ofSplit q).pr x) = incA D x * q.e

theorem **val_xi** (q : SplitPt D k) (hU : q.UnitAA) : valuation k (ofSplit q).xi = 1

theorem **eq_sec_add_csec** (q : SplitPt D k) {n : ℕ} {y : Fin n → AE D} (hy : ∀ i, y i ∈ idAt D (· = q.v)) : y = (fun i => q.sec (q.pr (y i))) + (fun i => cA D (q.sec (q.pr (cA D (y i)))))

theorem **mem_of_sec_iff** (q : SplitPt D k) {S : HSp D} {L L' : Lat S} (hL : IsSelfDualC S L) (hL' : IsSelfDualC S L') (h : ∀ u : Fin S.rank → k, (fun i => q.sec (u i)) ∈ latV S L (· = q.v) ↔ (fun i => q.sec (u i)) ∈ latV S L' (· = q.v)) {y : Fin S.rank → AE D} (hy : y ∈ latV S L (· = q.v)) : y ∈ latV S L' (· = q.v)

theorem **latV_eq_of_frame** (q : SplitPt D k) {S : HSp D} {L L' : Lat S} (hL : IsSelfDualC S L) (hL' : IsSelfDualC S L') {Y : Type} [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) (h1 : IntIn q S F L) (h2 : IntIn q S F L') : latV S L (· = q.v) = latV S L' (· = q.v)

theorem **besselNormalFrameAuxiliaryColumnIndex_bijective** (ρ : ℕ) : Function.Bijective (besselNormalFrameAuxiliaryColumnIndex ρ)

theorem **sum_divisible_eq_sum_mul** {M : Type} [AddCommMonoid M] (ρ : ℕ) (f : Fin (2 * ρ + 1) → M) : ∑ j, f j = ∑ i : Fin ρ, f ⟨i.val, by omega⟩ + f ⟨ρ, by omega⟩ + ∑ i : Fin ρ, f ⟨2 * ρ - i.val, by omega⟩

theorem **forall_fin_odd_iff_lower_middle_upperReversed** (ρ : ℕ) (p : Fin (2 * ρ + 1) → Prop) : (∀ j, p j) ↔ (∀ i : Fin ρ, p ⟨i.val, by omega⟩) ∧ p ⟨ρ, by omega⟩ ∧ ∀ i : Fin ρ, p ⟨2 * ρ - i.val, by omega⟩

theorem **proj_sec** (q : SplitPt D k) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (w : Fin S₀.rank → k) (a b : Fin ρ → k) : (d.pX *ᵥ fun i => q.sec ((d.i₀.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ w + d.e.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ a + d.e'.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ b) i)) = (fun i => q.sec (a i)) ∧ (d.pY *ᵥ fun i => q.sec ((d.i₀.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ w + d.e.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ a + d.e'.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ b) i)) = (fun i => q.sec (b i)) ∧ (d.p0 *ᵥ fun i => q.sec ((d.i₀.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ w + d.e.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ a + d.e'.map (GlobalTheory.SliceModel.prH (ofSplit q)) *ᵥ b) i)) = (fun i => q.sec (w i))

theorem **lineX_sec** (q : SplitPt D k) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (M : Lat S₀) (i : Fin ρ) (c : k) : colMul d.jX i (q.sec c) ∈ latV S (splitL d M fun _ => 1) (· = q.v) ↔ c ∈ 𝒪[k]

theorem **lineY_sec** (q : SplitPt D k) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (M : Lat S₀) (i : Fin ρ) (c : k) : colMul d.jY i (q.sec c) ∈ latV S (splitL d M fun _ => 1) (· = q.v) ↔ c ∈ 𝒪[k]

theorem **intIn_splitL** (q : SplitPt D k) {ρ : ℕ} (B : BesLev D ρ) {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} (s : k) (FA : GlobalTheory.LocMod.Frame k B.A.rank n) (F₀ : GlobalTheory.LocMod.Frame k B.V₀.rank (n ⊕ Unit)) (FV : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin c)) (M : Lat B.V₀) (hs : valuation k s = 1) (hxi : valuation k (ofSplit q).xi = 1) (hz : valuation k ((ofSplit q).hz B) = 1) (hC : (ofSplit q).IsCoreFrame B s FA F₀) (hB : (ofSplit q).IsBesFrame B s FA FV) (hM : IntIn q B.V₀ F₀ M) : IntIn q B.V FV (splitL B.d M fun _ => 1)

theorem **GlobalTheory.MonoBes.besLat_monoAt** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m), GlobalTheory.FinPl.MonoAt ((y.b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLatticeSplitsAlongWittNormalFramesStatement

theorem **_root_.GlobalTheory.T.besselLattice_splits_along_witt_normalFrames** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m),

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLevelLineVectorPrimitiveRowLatticeNorm

theorem **cY_eq_repr** (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (y : (VS D).V) : GlobalTheory.Core.SBesEx.cY X h y = (X.bas.reindex (finCongr (rank_eq X h)).symm).repr y

theorem **sum_bas_mem_iff** (I : AutInst) (hI : I = Formalism.L2.SG.InstExistsPf.instOf D) (ιE : I.E ≃+* D.host.E) (hId : ∀ x : I.E, HEq (ιE x) x) (hn : (D.sp.Y 0).rank = I.n) (bas : Module.Basis (Fin I.n) D.host.E (VS D).V) (hL : ∀ x : Fin I.n → I.E, x ∈ I.L ↔ (∑ i, ιE (x i) • bas i) ∈ (VS D).L0) (v : Fin (D.sp.Y 0).rank → D.host.E) : (∑ k, v k • bas (Fin.cast hn k)) ∈ (VS D).L0 ↔ (∑ k, v k • (VS D).frame k) ∈ (VS D).L0

theorem **cY_mem_latS_iff** (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (hId : GlobalTheory.Core.Mi.IdE X) (y : (VS D).V) : GlobalTheory.Core.SBesEx.cY X h y ∈ latS D ↔ y ∈ (VS D).L0

theorem **cY_smul** (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (c : D.host.E) (y : (VS D).V) : GlobalTheory.Core.SBesEx.cY X h (c • y) = c • GlobalTheory.Core.SBesEx.cY X h y

theorem **smul_zu_mem_iff** (X : HidaSetting D) (u : D.what₂.Word) (c : D.host.E) : c • X.zu u ∈ (VS D).L0 ↔ ∃ o : 𝒪 D.host.E, (o : D.host.E) = c

theorem **quad** {m n : Type} [Fintype m] [Fintype n] (M : Matrix m n D.host.E) (G : Matrix m m D.host.E) (x : n → D.host.E) : star x ⬝ᵥ (Mᴴ * G * M) *ᵥ x = star (M *ᵥ x) ⬝ᵥ G *ᵥ (M *ᵥ x)

theorem **GlobalTheory.Core.Mi.LatMatch.sz_tied** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (μ lam : D.host.E), GlobalTheory.Core.Mi.IdE X → B.FrameAt μ lam → (∀ t : D.host.E, t • vecB B.o B.ζ ∈ coreLat B.d (latS D) ↔ μ * t ∈ (1 : FI D)) ∧ star B.ζ ⬝ᵥ B.L.gram *ᵥ B.ζ = -(μ * star μ * (((D.mem u).A.T : ℕ) : D.host.E)) ∧ vecB B.o B.ζ ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLevelLineVectorPrimitiveRowLatticeNormStatement

theorem **_root_.GlobalTheory.T.besselLevelLineVector_primitive_rowLattice_norm** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (μ lam : D.host.E), GlobalTheory.Core.Mi.IdE X → B.FrameAt μ lam → (∀ t : D.host.E, t • vecB B.o B.ζ ∈ coreLat B.d (latS D) ↔ μ * t ∈ (1 : FI D)) ∧ star B.ζ ⬝ᵥ B.L.gram *ᵥ B.ζ = -(μ * star μ * (((D.mem u).A.T : ℕ) : D.host.E)) ∧ vecB B.o B.ζ ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLevelStepFourierJacobiPeriodLink

theorem **star_psi** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (a : AF D.host.F) : star (ψ a) = ψ (-a)

theorem **heisO_at_zero'** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (w : WO S T) (f : WO S T → ℂ) : heisO ψ S T w f 0 = f w

theorem **link** : ∀ D : D7, LinkLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLevelStepFourierJacobiPeriodLinkStatement

theorem **_root_.GlobalTheory.T.bessel_level_step_fourierJacobi_period_link** : ∀ D : D7, LinkLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLineContractionHeisenbergInvariant

theorem **symplOH_kappa** {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (w w' : WO S B'.L) : symplOH B.L S (l.κ S w) (l.κ S w') = -symplOH S B'.L w w'

theorem **kappa_add** {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (w w' : WO S B'.L) : l.κ S (w + w') = l.κ S w + l.κ S w'

theorem **kappa_rat** {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (w : WO S B'.L) (hw : w ∈ ratMx D B'.L.rank S.rank) : l.κ S w ∈ ratMx D S.rank B.L.rank

theorem **contr_heisO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (Ξ₁ : WO B.V₀ S → ℂ) (Ξ₂ : WO S B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ S Ξ₁ → IsThetaFunO ψ S B'.V₀ Ξ₂ → ∀ x : WO S B'.L, GlobalTheory.Unf.contr l S vL (heisO ψ B.V₀ S ((B.o.resS S).symm (0, l.κ S x)) Ξ₁) (heisO ψ S B'.V₀ ((B'.o.resT S).symm (0, x)) Ξ₂) = GlobalTheory.Unf.contr l S vL Ξ₁ Ξ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLineContractionHeisenbergInvariantStatement

theorem **_root_.GlobalTheory.T.besselLineContraction_heisenberg_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (Ξ₁ : WO B.V₀ S → ℂ) (Ξ₂ : WO S B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ S Ξ₁ → IsThetaFunO ψ S B'.V₀ Ξ₂ → ∀ x : WO S B'.L, GlobalTheory.Unf.contr l S vL (heisO ψ B.V₀ S ((B.o.resS S).symm (0, l.κ S x)) Ξ₁) (heisO ψ S B'.V₀ ((B'.o.resT S).symm (0, x)) Ξ₂) = GlobalTheory.Unf.contr l S vL Ξ₁ Ξ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLineContractionThetaFunctionIsLocalModel

theorem **ι_mul_incA** (x : k) (b : D.host.E) : p.ι x ∗ incA D b = p.ι (x ∗ p.pr b)

theorem **map_ι_mul_rat** {a b c : Type} [Fintype b] (M : Matrix a b k) (N : Matrix b c D.host.E) : M.map p.ι ∗ N.map (incA D) = (M ∗ N.map p.pr).map p.ι

theorem **map_pr_eq** {a b : Type} (M : Matrix a b D.host.E) : M.map p.pr = M.map (Witness.prH p)

theorem **core_iA** {ρ : ℕ} (B : BesLev D ρ) {s : k} (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (h : p.IsCoreFrame B s FA F₀) : F₀.γ ∗ B.o.iA.map p.pr = fromRows FA.γ (0 : Matrix Unit (Fin B.A.rank) k)

theorem **core_qA** {ρ : ℕ} (B : BesLev D ρ) {s : k} (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (h : p.IsCoreFrame B s FA F₀) : FA.γ ∗ B.o.qA.map p.pr ∗ F₀.β = fromCols (1 : Matrix n' n' k) (0 : Matrix n' Unit k)

theorem **up_fromCols** {ρ : ℕ} (B : BesLev D ρ) {s : k} (S : HSp D) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n) (h : p.IsCoreFrame B s FA F₀) (x : Matrix n n' k) : p.up F₀ FS (fromCols x (0 : Matrix n Unit k)) = (B.o.resS S).symm (p.up FA FS x, 0)

theorem **up_jA** {ρ : ℕ} (B : BesLev D ρ) {s : k} (S : HSp D) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n) (h : p.IsCoreFrame B s FA F₀) (t : Matrix n (n' ⊕ Unit) k) : p.up F₀ FS t · B.o.jA = p.up FA FS t.toCols₁

theorem **symplOH_zero_r** {S T : HSp D} (x : WO S T) : symplOH S T x 0 = 0

theorem **heis_pass** {V A L : HSp D} (o : OrthDec V A L) (S : HSp D) (f : WO V S → ℂ) (a u : WO A S) (z : WO L S) : heisO ψ V S ((o.resS S).symm (a, 0)) f ((o.resS S).symm (u, z)) = ψ (symplOH A S u a) ∗ f ((o.resS S).symm (u + a, z))

theorem **contr_isModel** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ (T₁ : SB (Matrix n (n' ⊕ Unit) k) →ₗ[ℂ] (WO B.V₀ S → ℂ)), p.IsModel ψ F₀ FS T₁ → ∀ (φ₀ : SB (n → k)) (Ξ₂ : WO S B'.V₀ → ℂ), IsThetaFunO ψ S B'.V₀ Ξ₂ → ∀ (v : WO S B'.A) (T : SB (Matrix n n' k) →ₗ[ℂ] (WO B.A S → ℂ)), (∀ Ξ₁ u, T Ξ₁ u = GlobalTheory.Unf.contr l S vL (T₁ (GlobalTheory.ContractionLocal.colTens φ₀ Ξ₁)) Ξ₂ u v) → p.IsModel ψ FA FS T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLineContractionThetaFunctionIsLocalModelStatement

theorem **_root_.GlobalTheory.T.besselLine_contraction_thetaFunction_isLocalModel** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ (T₁ : SB (Matrix n (n' ⊕ Unit) k) →ₗ[ℂ] (WO B.V₀ S → ℂ)), p.IsModel ψ F₀ FS T₁ → ∀ (φ₀ : SB (n → k)) (Ξ₂ : WO S B'.V₀ → ℂ), IsThetaFunO ψ S B'.V₀ Ξ₂ → ∀ (v : WO S B'.A) (T : SB (Matrix n n' k) →ₗ[ℂ] (WO B.A S → ℂ)), (∀ Ξ₁ u, T Ξ₁ u = GlobalTheory.Unf.contr l S vL (T₁ (GlobalTheory.ContractionLocal.colTens φ₀ Ξ₁)) Ξ₂ u v) → p.I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLineLinkGramSimilitudeBijective

theorem **exists_inv_of_rank_one** {m n : ℕ} (hm : m = 1) (hn : n = 1) (τ : Matrix (Fin m) (Fin n) D.host.E) (h : τ ≠ 0) : ∃ σ : Matrix (Fin n) (Fin m) D.host.E, σ ∗ τ = 1 ∧ τ ∗ σ = 1

theorem **scalar_one** : incA D (-D.host.xi0⁻¹) ∗ cA D (incA D D.host.xi0) = 1

theorem **ofAdj_starAdj** (L S : HSp D) (x : WO L S) : ofAdj L S (starAdj L S x) = x

theorem **link_gram** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E), IsLineVec L' ζ' → ∀ l : Link B L' ζ', l.τᴴ · (B.L.gram · l.τ) = (((1 : ℤ) : D.host.E) ∗ (D.host.xi0 ∗ star D.host.xi0)) • L'.gram ∧ ∀ S : HSp D, Function.Bijective (l.κ S) ∧ ∀ w, l.κ S w ∈ ratMx D S.rank B.L.rank → w ∈ ratMx D L'.rank S.rank

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLineLinkGramSimilitudeBijectiveStatement

theorem **_root_.GlobalTheory.T.besselLineLink_gram_similitude_bijective** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E), IsLineVec L' ζ' → ∀ l : Link B L' ζ', l.τᴴ · (B.L.gram · l.τ) = (((1 : ℤ) : D.host.E) ∗ (D.host.xi0 ∗ star D.host.xi0)) • L'.gram ∧ ∀ S : HSp D, Function.Bijective (l.κ S) ∧ ∀ w, l.κ S w ∈ ratMx D S.rank B.L.rank → w ∈ ratMx D L'.rank S.rank

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLinkKappaAdjointTauFactor

theorem **doubleSum_eq_dotProduct_mapMatrix_mulVec** {R K : Type*} [CommRing R] {m n : ℕ} (g : R → R) (f : K → R) (s : Fin m → R) (G : Matrix (Fin m) (Fin n) K) (v : Fin n → R) : (∑ i, ∑ j, g (s i) ∗ f (G i j) ∗ v j) = (fun i => g (s i)) ·ᵥ (G.map f *ᵥ v)

theorem **ringHom_conjTranspose_smul_dotProduct_adjoint** {R : Type*} [CommRing R] {m n : ℕ} (σ : R →+* R) (c : R) (s : Fin m → R) (Y : Matrix (Fin n) (Fin m) R) (G : Matrix (Fin n) (Fin n) R) (a : Fin n → R) : (fun i => σ (s i)) ·ᵥ ((c • ((Y.map σ)ᵀ ∗ G)) *ᵥ a) = c ∗ ((fun k => σ ((Y *ᵥ s) k)) ·ᵥ (G *ᵥ a))

theorem **kappa_adj** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (S : HSp D) (x : WO S L') (a : Fin B.L.rank → AE D) (s : Fin S.rank → AE D), (∑ i, ∑ j, cA D (s i) ∗ incA D (HSpace.gram S i j) ∗ ((l.κ S x : Matrix _ _ (AE D)) *ᵥ a) j) = incA D (-D.host.xi0⁻¹) ∗ ∑ i, ∑ j, cA D (((((rat l.τ : Matrix _ _ (AE D)) ∗ (x : Matrix _ _ (AE D))) *ᵥ s) i) ∗ incA D (HSpace.gram B.L i j) ∗ a j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLinkKappaAdjointTauFactorStatement

theorem **_root_.GlobalTheory.T.besselLinkKappa_adjoint_tau_factor** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (S : HSp D) (x : WO S L') (a : Fin B.L.rank → AE D) (s : Fin S.rank → AE D), (∑ i, ∑ j, cA D (s i) ∗ incA D (HSpace.gram S i j) ∗ ((l.κ S x : Matrix _ _ (AE D)) *ᵥ a) j) = incA D (-D.host.xi0⁻¹) ∗ ∑ i, ∑ j, cA D (((((rat l.τ : Matrix _ _ (AE D)) ∗ (x : Matrix _ _ (AE D))) *ᵥ s) i) ∗ incA D (HSpace.gram B.L i j) ∗ a j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLinkKappaPreservesSupportedMatrices

theorem **mxAt_linkKappa** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (S : HSp D) (P : Pl D → Prop) (w : WO S L'), MxAt P w → MxAt P (l.κ S w)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLinkKappaPreservesSupportedMatricesStatement

theorem **_root_.GlobalTheory.T.besselLink_kappa_preserves_supported_matrices** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (S : HSp D) (P : Pl D → Prop) (w : WO S L'), MxAt P w → MxAt P (l.κ S w)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLinkKappaSurjectiveSupportedConjugateHalf

theorem **isHalf_mul_l** {e : AE D} {l m n : ℕ} (A : Mx D l m) {x : Mx D m n} (hx : IsHalf e x) : IsHalf e (A * x)

theorem **isHalf_mul_r** {e : AE D} {l m n : ℕ} {x : Mx D l m} (hx : IsHalf e x) (A : Mx D m n) : IsHalf e (x * A)

theorem **isHalf_smul** {e : AE D} {m n : ℕ} (c : AE D) {x : Mx D m n} (hx : IsHalf e x) : IsHalf e (c • x)

theorem **isHalf_adj** {e : AE D} {m n : ℕ} {x : Mx D m n} (hx : IsHalf e x) : IsHalf (cA D e) (adj x)

theorem **mul_tauPlus** {ρ : ℕ} {B : BesLev D ρ} {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') : l.τ * tauPlus l = 1

theorem **kappa_inv** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (S : HSp D) (P : Pl D → Prop) (h : WO B.L S), ∃ x : WO S L', l.κ S x = h ∧ (MxAt P h → MxAt P x) ∧ ∀ e : AE D, IsHalf e h → IsHalf (cA D e) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLinkKappaSurjectiveSupportedConjugateHalfStatement

theorem **_root_.GlobalTheory.T.besselLink_kappa_surjective_supported_conjugateHalf** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (S : HSp D) (P : Pl D → Prop) (h : WO B.L S), ∃ x : WO S L', l.κ S x = h ∧ (MxAt P h → MxAt P x) ∧ ∀ e : AE D, IsHalf e h → IsHalf (cA D e) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLocalComponentData

theorem **exists_place** (E : Type) [Field E] [NumberField E] {T : ℕ} (hT : T.Prime) : ∃ P : HeightOneSpectrum (𝓞 E), (T : 𝓞 E) ∈ P.asIdeal

theorem **isU_nat** {x : GlobalTheory.FinPl.Pl D} {N : ℕ} (hN : (N : 𝓞 D.host.F) ∉ x.asIdeal) : IsU x (incA D (N : D.host.E))

theorem **isU_xi_of_prime** {x : GlobalTheory.FinPl.Pl D} {T : ℕ} (hT : T.Prime) (hv : (T : 𝓞 D.host.F) ∈ x.asIdeal) (hd : T ≠ D.host.d) (hc : Nat.Coprime (Ideal.absNorm D.host.ccK) T) : IsU x (xiU D : AE D)

theorem **isU_alpha_of_prime** {x : GlobalTheory.FinPl.Pl D} {T : ℕ} (hT : T.Prime) (hv : (T : 𝓞 D.host.F) ∈ x.asIdeal) (h2 : T ∉ D.host.small) (hq : T ≠ D.host.q) (hd : T ≠ D.host.d) (hc : Nat.Coprime (Ideal.absNorm D.host.ccK) T) {L : Formalism.BasicNotions.TH.HSp D} {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) {α : (AE D)ˣ} (hα : ∀ (S : Formalism.BasicNotions.TH.HSp D) (L : GlobalTheory.FinPl.Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) : IsU x (α : AE D)

theorem **lat_zero_line** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (hr : 1 ≤ D.R.rS) u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v' : VCom f.c) (o : TDat f.c k ι) (n : ℕ) (G : Fin n → GLoc v') (pl : Fin n ≃ (pk W).Pg), f.c.Is → v'.Is → VTie f v' → ShapeOff v' o → GPin G → GRead f G pl → BotTie f B v' o G → GlobalTheory.MiTop.PrimTop (c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLocalComponentDataStatement

theorem **_root_.GlobalTheory.T.bessel_local_component_data** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (hr : 1 ≤ D.R.rS) u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v' : VCom f.c) (o : TDat f.c k ι) (n : ℕ) (G : Fin n → GLoc v') (pl : Fin n ≃ (pk W).Pg), f.c.Is → v'.Is → VTie f v' → ShapeOff v' o → GPin G → GRead f G pl → BotTie f B v' o G → GlobalTheory.MiTop.PrimTop (c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLocalTypeData

theorem **pg_isTypePlaces** (W : RealPack ex) : GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg

theorem **stZero_is** (W : RealPack ex) (hP : PkHyp W) (hm : 0 < D.R.m) : (stZero W).st.Is (GlobalTheory.BaseChain.com W) 0

theorem **bot_of_read** (W : RealPack ex) (hP : PkHyp W) (hm : 0 < D.R.m) {u : D.what₂.Word} (f : FamCh W u) (v : VCom f.c) (hv : v.Is) (ht : VTie f v) {n : ℕ} (G : (pk W).Pg → GLoc v) (pl : Fin n ≃ (pk W).Pg) (hR : GRead f (fun i => G (pl i)) pl) : Bot (c

theorem **bot_of_placedBy** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom f.c) (o : TDat f.c k ι) (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GlobalTheory.LevZero.PlacedBy f B v o G pl → GlobalTheory.LevZero.Bot (c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselLocalTypeDataStatement

theorem **_root_.GlobalTheory.T.bessel_local_type_data** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom f.c) (o : TDat f.c k ι) (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GlobalTheory.LevZero.PlacedBy f B v o G pl → GlobalTheory.LevZero.Bot (c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselModelDoublingData

theorem **poincare_eq_zero_of_per_vanish** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (Y : Formalism.L2.SG.Full.Full X) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)), S = parB (Ptw D) 0 → IsInfFam θ → IsStdChar D ψ (-1) → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → GlobalTheory.Core.Mi.IdE X → Matches X h J χ t → InHH X u Y.hK → ∀ T : Cu X u Y.hK, (∀ G ∈ MultT θ M J χ, ∀ ξ x, B.lev.per ψ (B.colForm T.1 ξ) (rTrans ((iotaX X h).symm (1, X.xu u) * t) (G x)) = 0) → PuFun X u Y.hK (Y.Dt u) T = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselModelDoublingDataStatement

theorem **_root_.GlobalTheory.T.puFun_eq_zero_of_per_colForm_eq_zero** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (Y : Formalism.L2.SG.Full.Full X) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)), S = parB (Ptw D) 0 → IsInfFam θ → IsStdChar D ψ (-1) → GlobalTheory.Count.GapGe X.B (100 *

(D.R.r : ℝ) ^ 4) → GlobalTheory.Core.Mi.IdE X → Matches X h J χ t → InHH X u Y.hK → ∀ T : Cu X u Y.hK, (∀ G ∈ MultT θ M J χ, ∀ ξ x, B.lev.per ψ (B.colForm T.1 ξ) (rTrans ((iotaX X h).symm (1, X.xu u) ∗ t) (G x)) = 0) → PuFun X u Y.hK (Y.Dt u) T = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselNormalFrameIsotropicColumnsPairing

theorem **cols_pair** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ρ : Place D k) (ρ : ℕ) (B : BesLev D ρ) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (s : k) (FA : Frame k B.A.rank n) (F : Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA F → ∀ i : Fin ρ, ∃ m₀ m₁ : n ⊕ Fin c, (∀ r, F.β r m₀ = s ∗ p.xi ^ (ρ − i.val) ∗ p.pr (B.d.e r i)) ∧ (∀ r, F.β r m₁ = s ∗ (−p.hz B / p.xi) ∗ (−1 / p.xi) ^ (ρ − 1 − i.val) ∗ p.pr (B.d.e' r i)) ∧ ∀ m, m ≠ m₁ → ∑ r, ∑ r', p.pr (star (B.d.e r i)) ∗ p.pr (B.V.gram r r') ∗ F.β r' m = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselNormalFrameIsotropicColumnsPairingStatement

theorem **_root_.GlobalTheory.T.besselNormalFrame_isotropicColumns_pairing** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ρ : Place D k) (ρ : ℕ) (B : BesLev D ρ) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (s : k) (FA : Frame k B.A.rank n) (F : Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA F → ∀ i : Fin ρ, ∃ m₀ m₁ : n ⊕ Fin c, (∀ r, F.β r m₀ = s ∗ p.xi ^ (ρ − i.val) ∗ p.pr (B.d.e r i)) ∧ (∀ r, F.β r m₁ = s ∗ (−p.hz B / p.xi) ∗ (−1 / p.xi) ^ (ρ − 1 − i.val) ∗ p.pr (B.d.e' r i)) ∧ ∀ m, m ≠ m₁ → ∑ r, ∑ r', p.pr (star (B.d.e r i)) ∗ p.pr (B.V.gram r r') ∗ F.β r' m = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodCharacterPrimitiveNormalFrames

theorem **z_pair** {D : D7} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (v : Fin S₀.rank → D.host.E) (i : Fin ρ) : ∑ r, ∑ t, star ((d.i₀ ∗ᵥ v) r) ∗ HSpace.gram S r t ∗ d.e t i = 0

theorem **prim_of_normal** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), GlobalTheory.FinPl.PrimAt ψ ((y.bes j ρ h).lev (a.x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodCharacterPrimitiveNormalFramesStatement

theorem **_root_.GlobalTheory.T.besselPeriodCharacter_primitive_normalFrames** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m),

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodDoubleStepIteratedQuotientIntegral

theorem **quotient_integral_finset_sum_eq** {ι : Type} (s : Finset ι) (F : ι → G → ℂ) (Z : ι → ℂ) (hF : ∀ i ∈ s, IsQuotIntL1 μR μG (F i) (Z i)) : IsQuotIntL1 μR μG (fun x => ∑ i ∈ s, F i x) (∑ i ∈ s, Z i)

theorem **quotient_integral_finset_sum_smul** {ι : Type} (s : Finset ι) (a : ι → ℂ) (F : ι → G → ℂ) (Z : ι → ℂ) (hF : ∀ i ∈ s, IsQuotIntL1 μR μG (F i) (Z i)) : IsQuotIntL1 μR μG (fun x => ∑ i ∈ s, a i ∗ F i x) (∑ i ∈ s, a i ∗ Z i)

theorem **doubleStep_tuple** : ∀ (D : D7) (c : Comm D) (t : Tup c), DoubleStep D → c.Is → t.Is → ∀ j σ (h : j + 2 ∗ (σ + 1) + 2 = D.R.m) P₁ P₂ (φ : t.Ch j → GA D (c.X j) → ℂ) (f : GA D (c.V (j + 1 + 1)) → ℂ), (∀ a, IsContAut (c.X j) (φ a)) → RapidDec _ f → RapidDec _ (liftS (t.om (j + 1)) (c.vol (j + 1 + 1)).μ (c.vol (j + 1 + 1)).𝓕 f (t.pt (j + 1) P₂)) → ∃ F₁ : GA D (c.V (j + 1)) → ℂ, IsQuotIntL1 (c.J j (σ + 1) h).μR (c.vol (j + 1)).μ F₁ (t.perF j (σ + 1) h φ (liftS (t.om j) (c.vol (j + 1)).μ (c.vol (j + 1)).𝓕 (liftS (t.om (j + 1)) (c.vol (j + 1 + 1)).μ (c.vol (j + 1 + 1)).𝓕 f (t.pt (j + 1) P₂)) (t.pt j P₁))) ∧ ∀ h₁, IsQuotIntL1 (c.B (j + 1 + 1) σ (idx_step h)).μR (c.vol (j + 1 + 1)).μ (t.integrand j σ h P₁ P₂ φ f h₁) (F₁ h₁)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodDoubleStepIteratedQuotientIntegralStatement

theorem **_root_.GlobalTheory.T.besselPeriod_double_step_iterated_quotient_integral** : ∀ (D : D7) (c : Comm D) (t : Tup c), DoubleStep D → c.Is → t.Is → ∀ j σ (h : j + 2 ∗ (σ + 1) + 2 = D.R.m) P₁ P₂ (φ : t.Ch j → GA D (c.X j) → ℂ) (f : GA D (c.V (j + 1 + 1)) → ℂ), (∀ a, IsContAut (c.X j) (φ a)) → RapidDec _ f → RapidDec _ (liftS (t.om (j + 1)) (c.vol (j + 1 + 1)).μ (c.vol (j + 1 + 1)).𝓕 f (t.pt (j + 1) P₂)) → ∃ F₁ : GA D (c.V (j + 1)) → ℂ, IsQuotIntL1 (c.J j (σ + 1) h).μR (c.vol (j + 1)).μ F₁ (t.perF j (σ + 1) h φ (liftS (t.om j) (c.vol (j + 1)).μ (c.vol (j + 1)).𝓕 (liftS (t.om (j + 1)) (c.vol (j + 1 + 1)).μ (c.vol (j + 1 + 1)).𝓕 f (t.pt (j + 1) P₂)) (t.pt j P₁))) ∧ ∀ h₁, IsQuotIntL1 (c.B (j + 1 + 1) σ (idx_step h)).μR (c.vol (j + 1 + 1)).μ (t.integrand j σ h P₁ P₂ φ f h₁) (F₁ h₁)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodDoubleStepUnfoldingOfDescent

theorem **doubleStep_of_descent** : ∀ D : D7, Descent D → DoubleStep D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodDoubleStepUnfoldingOfDescentStatement

theorem **_root_.GlobalTheory.T.besselPeriod_double_step_unfolding_of_descent** : ∀ D : D7, Descent D → DoubleStep D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodEvaluationUnfoldingOfWittDescent

theorem **nonempty_orthDec_zero** (D : D7) (S : HSp D) : Nonempty (OrthDec S S (diagSp D 0 ![] 0 0))

private theorem **s_finite_of_sigma_finite** {X : Type} [MeasurableSpace X] (ν : Measure X) [SigmaFinite ν] : SFinite ν

theorem **evalBF_of_descent** : ∀ D : D7, Descent D → EvalBF D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodEvaluationUnfoldingOfWittDescentStatement

theorem **_root_.GlobalTheory.T.bessel_period_evaluation_unfolding_of_witt_descent** : ∀ D : D7, Descent D → EvalBF D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodLinearAndTranslationCombination

theorem **contAut_add** {V : HSp D} {f g : GA D V → ℂ} (hf : IsContAut V f) (hg : IsContAut V g) : IsContAut V (f + g)

theorem **contAut_smul** {V : HSp D} {f : GA D V → ℂ} (hf : IsContAut V f) (c : ℂ) : IsContAut V (c • f)

theorem **contAut_rTrans** {V : HSp D} {f : GA D V → ℂ} (h : GA D V) (hf : IsContAut V f) : IsContAut V (rTrans h f)

theorem **contAut_sum** {V : HSp D} {I : Type} (s : Finset I) (a : I → ℂ) (f : I → GA D V → ℂ) (hf : ∀ i ∈ s, IsContAut V (f i)) : IsContAut V (∑ i ∈ s, a i • f i)

theorem **per_sum_right** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} (B : BesLev D ρ) {φ : GA D B.A → ℂ} (hφ : IsContAut B.A φ) {I : Type} (s : Finset I) (a : I → ℂ) (f : I → GA D B.V → ℂ) (hf : ∀ i ∈ s, IsContAut B.V (f i)) : B.per ψ φ (∑ i ∈ s, a i • f i) = ∑ i ∈ s, a i * B.per ψ φ (f i)

theorem **per_linear_trc** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ : ℕ) (B : BesLev D ρ) (φ : GA D B.A → ℂ), IsContAut B.A φ → (∀ (I : Type) (s : Finset I) (a : I → ℂ) (f : I → GA D B.V → ℂ), (∀ i ∈ s, IsContAut B.V (f i)) → B.per ψ φ (∑ i ∈ s, a i • f i) = ∑ i ∈ s, a i * B.per ψ φ (f i)) ∧ ∀ (e : GA D B.V →ₗ ℂ) (f : GA D B.V → ℂ), IsContAut B.V f → IsContAut B.V (trc e f) ∧ B.per ψ φ (trc e f) = ∑ g ∈ e.support, e g * B.per ψ φ (rTrans g f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodLinearAndTranslationCombinationStatement

theorem **_root_.GlobalTheory.T.besselPeriod_linear_and_translationCombination** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ : ℕ) (B : BesLev D ρ) (φ : GA D B.A → ℂ), IsContAut B.A φ → (∀ (I : Type) (s : Finset I) (a : I → ℂ) (f : I → GA D B.V → ℂ), (∀ i ∈ s, IsContAut B.V (f i)) → B.per ψ φ (∑ i ∈ s, a i • f i) = ∑ i ∈ s, a i * B.per ψ φ (f i)) ∧ ∀ (e : GA D B.V →ₗ ℂ) (f : GA D B.V → ℂ), IsContAut B.V f → IsContAut B.V (trc e f) ∧ B.per ψ φ (trc e f) = ∑ g ∈ e.support, e g * B.per ψ φ (rTrans g f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodRightTranslateSobolevSepContinuous

theorem **rTrans_mem** {V : HSp D} (t : GA D V) {F : GA D V → ℂ} (hF : F ∈ contLeft V) : rTrans t F ∈ contLeft V

theorem **norm_psiBes** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (θ : D.host.E) (z : Fin S₀.rank → D.host.E) (n : GA D S) : ‖psiBes ψ d θ z n‖ = 1

theorem **per_add_bound** {ρ : ℕ} (B : BesLev D ρ) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {φ : GA D B.A → ℂ} (hφ : IsContAut B.A φ) (Ω : Set (GA D B.V)) (hΩc : IsCompact Ω) (hΩ : ∀ s ∈ besGrp B.d B.o, ∃ γ ∈ GQ B.V ⊓ besGrp B.d B.o, γ * s ∈ Ω) : ∃ C : ℝ, (∀ f g, f ∈ contLeft B.V → g ∈ contLeft B.V → B.per ψ φ (f + g) = B.per ψ φ f + B.per ψ φ g) ∧ ∀ f, f ∈ contLeft B.V → ∀ K : ℝ, (∀ x ∈ Ω, ‖f x‖ ≤ K) → ‖B.per ψ φ f‖ ≤ C * K

theorem **mult_mem_contLeft** (θ : InfFam D) {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) {T : MMap M} (hT : T ∈ Mult θ M J χ) (u : M.Vec) : T u ∈ contLeft V

theorem **mult_bound** (θ : InfFam D) (hθ : IsInfFam θ) {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) {T : MMap M} (hT : T ∈ Mult θ M J χ) (Ω : Set (GA D V)) (hΩ : IsCompact Ω) (τ : D.what₂.Tau) (v : M.Vec) : ∃ (L : List (List (Matrix (Fin V.rank) (Fin V.rank) ℂ))) (C : ℝ), (∀ l ∈ L, ∀ Y ∈ l, Y ∈ lieAt V τ) ∧ ∀ w : M.sm τ, ∀ g ∈ Ω, ‖T (Function.update v τ w) g‖ ≤ C * (L.map fun l => ‖l.foldr (dRep (M.rep τ)) (w : M.H τ)‖).sum

theorem **sepCont_per_rTrans** : ∀ (D : D7) (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (S : D.what₂.Tau → SPar) (M : DSModel B.V S) (J : Set (GA D B.V)) (χ : GA D B.V → ℂ) (t : GA D B.V) (φ : GA D B.A → ℂ), IsInfFam θ → IsStdChar D ψ (-1) → IsContAut B.A φ → ∀ T ∈ MultT θ M J χ, ∃ Λ : SmT M →ₗ[ℂ] ℂ, SepCont M Λ ∧ ∀ x, Λ x = B.per ψ φ (rTrans t (T x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodRightTranslateSobolevSepContinuousStatement

theorem **_root_.GlobalTheory.T.besselPeriod_rightTranslate_sobolevSepContinuous** : ∀ (D : D7) (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (S : D.what₂.Tau → SPar) (M : DSModel B.V S) (J : Set (GA D B.V)) (χ : GA D B.V → ℂ) (t : GA D B.V) (φ : GA D B.A → ℂ), IsInfFam θ → IsStdChar D ψ (-1) → IsContAut B.A φ → ∀ T ∈ MultT θ M J χ, ∃ Λ : SmT M →ₗ[ℂ] ℂ, SepCont M Λ ∧ ∀ x, Λ x = B.per ψ φ (rTrans t (T x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodRightTranslationInvariant

theorem **inner_left** {ρ : ℕ} (B : BesLev D ρ) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsCharAF D.host.F ψ) (φ : GA D B.A → ℂ) (f : GA D B.V → ℂ) (hφ : ∀ γ ∈ GQ B.A, ∀ g, φ (γ * g) = φ g) (hf : ∀ γ ∈ GQ B.V, ∀ g, f (γ * g) = f g) {γ : GA D B.A} (hγ : γ ∈ GQ B.A) (h : GA D B.A) : ∫ n in B.vN.𝒩, besFun ψ 1 B.d B.o B.ζ φ f (γ * h) n ∂B.vN.ν = ∫ n in B.vN.𝒩, besFun ψ 1 B.d B.o B.ζ φ f h n ∂B.vN.ν

theorem **map_mul_right** {ρ : ℕ} (B : BesLev D ρ) (a : GA D B.A) : B.vA.μ.map (· * a) = B.vA.μ

theorem **setIntegral_mul_right** {ρ : ℕ} (B : BesLev D ρ) (G : GA D B.A → ℂ) (hG : ∀ γ ∈ GQ B.A, ∀ h, G (γ * h) = G h) (a : GA D B.A) : ∫ h in B.vA.𝓕, G (h * a) ∂B.vA.μ = ∫ h in B.vA.𝓕, G h ∂B.vA.μ

theorem **GlobalTheory.Core.Mi.per_rTrans** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (φ : GA D B.A → ℂ) (f : GA D B.V → ℂ) (a : GA D B.A), IsStdChar D ψ (-1) → IsContAut B.A φ → IsContAut B.V f → B.per ψ (rTrans a φ) (rTrans (embA B.d B.o a) f) = B.per ψ φ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodRightTranslationInvariantStatement

theorem **_root_.GlobalTheory.T.bessel_period_right_translation_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (φ : GA D B.A → ℂ) (f : GA D B.V → ℂ) (a : GA D B.A), IsStdChar D ψ (-1) → IsContAut B.A φ → IsContAut B.V f → B.per ψ (rTrans a φ) (rTrans (embA B.d B.o a) f) = B.per ψ φ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodThetaLiftEqIntegralKernel

theorem **enorm_fl_le** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) {V V₀ : HSp D} {ρ : ℕ} (d : HypDec V V₀ ρ) (W : HSp D) (v : VolM D W.rank ρ) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (Th : GA D V → ℂ) (hTh : ∀ n, ‖Th n‖ ≤ 1) (x : Mx D W.rank ρ) (Ψ : WO V W → ℂ) : ‖fl ψ d W v vb vU Th x Ψ‖ₑ ≤ ∫⁻ u, ∫⁻ b in vb.𝓕, ‖d.mixS W ψ v.μ v.𝓕 Ψ (x · u.1.1) (x · b)‖ₑ ∂vb.μ ∂vU.ν

theorem **liftS_mul** {D : D7} {V W : HSp D} (om : Om V W) (μ : Measure (GA D W)) (𝓕 : Set (GA D W)) (G : GA D W → ℂ) (Φ : WO V W → ℂ) (n g : GA D V) : liftS om μ 𝓕 G Φ (n * g) = liftS om μ 𝓕 G (om (g, 1) Φ) n

theorem **besPer_lift_eq_integral_kerBF** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∀ 𝓑 : Set (GA D J.W), IsFundamentalDomain (GQ J.W ⊓ jacGrp J.d : Subgroup (GA D J.W)) 𝓑 s.vW.μ → IntegrableOn (kerBF s φ G Φ) 𝓑 s.vW.μ ∧ B.per ψ φ (s.lift G Φ) = ∫ h in 𝓑, kerBF s φ G Φ h ∂s.vW.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodThetaLiftEqIntegralKernelStatement

theorem **_root_.GlobalTheory.T.besselPeriod_thetaLift_eq_integral_kernel** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ →

RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∀ 𝓑 : Set (GA D J.W), IsFundamentalDomain (GQ J.W ⊓ jacGrp J.d : Subgroup (GA D J.W)) 𝓑 s.vW.μ → IntegrableOn (kerBF s φ G Φ) 𝓑 s.vW.μ ∧ B.per ψ φ (s.lift G Φ) = ∫ h in 𝓑, kerBF s φ G Φ h ∂s.vW.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodThetaLiftUnfolding

theorem **unfBF_of_parts** (hcpt : ∀ (D : D7) (ρ : ℕ) (J : JacLev D ρ) (K : GA D J.W → ℂ), Continuous K → (∀ γ ∈ GQ J.W ⊓ jacGrp J.d, ∀ h, K (γ ∗ h) = K h) → ∀ h, AbsConv J.v0.μ J.v0.𝓕 J.vN.ν J.vN.𝓝 fun h' n => K (n.1 ∗ J.d.emb0 h' ∗ h)) (hfub : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∀ 𝓑 : Set (GA D J.W), IsFundamentalDomain (GQ J.W ⊓ jacGrp J.d : Subgroup (GA D J.W)) 𝓑 s.vW.μ → IntegrableOn (kerBF s φ G Φ) 𝓑 s.vW.μ ∧ B.per ψ φ (s.lift G Φ) = ∫ h in 𝓑, kerBF s φ G Φ h ∂s.vW.μ) : ∀ D : D7, Descent D → UnfBF D

theorem **unfBF** : ∀ D : D7, Descent D → UnfBF D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselPeriodThetaLiftUnfoldingStatement

theorem **_root_.GlobalTheory.T.besselPeriod_thetaLift_unfolding** : ∀ D : D7, Descent D → UnfBF D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselStepPositionOneFinite

theorem **GlobalTheory.PosOne.quotInt_one_comb** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ') (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (e : GL (n' ⊕ Fin (c + 1)) k →₀ ℂ) (_ : SrcComb ψ ν P ρ' Φ e) (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselStepPositionOneFiniteStatement

theorem **_root_.GlobalTheory.T.bessel_step_position_one_finite** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ') (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (e : GL (n' ⊕ Fin (c + 1)) k →₀ ℂ) (_ : SrcComb ψ ν P ρ' Φ e) (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselThetaContractionExtendByZeroVacuumSmooth

theorem **contr_comb_main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (f₁ : WO B.V₀ S → ℂ) (f₂ : WO S B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ S f₁ → IsThetaFunO ψ S B'.V₀ f₂ → IsBiTheta ψ B.A S B'.A (contr l S vL f₁ f₂) ∧ (∀ (c : WO B.A S →₀ ℂ) (c' : WO S B'.A →₀ ℂ), contr l S vL (comb ψ B.V₀ S (extS B.o S c) f₁) (comb ψ S B'.V₀ (extT B'.o S c') f₂) = biComb ψ B.A S B'.A c c' (contr l S vL f₁ f₂)) ∧ ∀ (P : Pl D → Prop) (LA : Lat B.A) (LS : Lat S) (LA' : Lat B'.A) (M : Lat B.V₀) (M' : Lat B'.V₀), IsSelfDualC B.A LA → IsSelfDualC S LS → IsSelfDualC B'.A LA' → IsSelfDualC B.V₀ M → IsSelfDualC B'.V₀ M' → (∀ v, P v → (∃ LL, OrthAt B.o M LA LL v) ∧ ∃ LL, OrthAt B'.o M' LA' LL v) → Vac ψ B.V₀ S M LS P f₁ → Vac ψ S B'.V₀ LS M' P f₂ → BiVacAt ψ LA LS LA' P (contr l S vL f₁ f₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselThetaContractionExtendByZeroVacuumSmoothStatement

theorem **_root_.GlobalTheory.T.besselThetaContraction_extendByZero_vacuum_smooth** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (f₁ : WO B.V₀ S → ℂ) (f₂ : WO S B'.V₀ → ℂ), IsThetaFunO ψ B.V₀ S f₁ → IsThetaFunO ψ S B'.V₀ f₂ → IsBiTheta ψ B.A S B'.A (contr l S vL f₁ f₂) ∧ (∀ (c : WO B.A S →₀ ℂ) (c' : WO S B'.A →₀ ℂ), contr l S vL (comb ψ B.V₀ S (extS B.o S c) f₁) (comb ψ S B'.V₀ (extT B'.o S c') f₂) = biComb ψ B.A S B'.A c c' (contr l S vL f₁ f₂)) ∧ ∀ (P : Pl D → Prop) (LA : Lat B.A) (LS : Lat S) (LA' : Lat B'.A) (M : Lat B.V₀) (M' : Lat B'.V₀), IsSelfDualC B.A LA → IsSelfDualC S LS → IsSelfDualC B'.A LA' → IsSelfDualC B.V₀ M → IsSelfDualC B'.V₀ M' → (∀ v, P v → (∃ LL, OrthAt B.o M LA LL v) ∧ ∃ LL, OrthAt B'.o M' LA' LL v) → Vac ψ B.V₀ S M LS P f₁ → Vac ψ S B'.V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToFourierJacobiHandedFamilyContinuousAutom

theorem **star_psi_eq** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (a : AF D.host.F) : star (ψ a) = ψ (-a)

theorem **handBF_isQuasiFam** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E), IsLineVec L' ζ' → ∀ (l : Link B L' ζ') (φ : GA D B.A → ℂ) (Ξ : WO B.V₀ J.W₀ → ℂ), IsContAut B.A φ → IsThetaFunO ψ B.V₀ J.W₀ Ξ → IsQuasiFam ψ J.W₀ L' (handBF s l φ Ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToFourierJacobiHandedFamilyContinuousAutomStatement

theorem **_root_.GlobalTheory.T.besselToFourierJacobi_handedFamily_continuous_automorphic** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E), IsLineVec L' ζ' → ∀ (l : Link B L' ζ') (φ : GA D B.A → ℂ) (Ξ : WO B.V₀ J.W₀ → ℂ), IsContAut B.A φ → IsThetaFunO ψ B.V₀ J.W₀ Ξ → IsQuasiFam ψ J.W₀ L' (handBF s l φ Ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFirstStepFlatMapSupportAll

theorem **congGrp_mono** (V : HSp D) {I J : Ideal (AE D)} (hIJ : I ≤ J) : congGrp V I ≤ congGrp V J

theorem **grpOff_le_at** (V : HSp D) {P : Pl D → Prop} {v : Pl D} (hv : P v) : grpOff V P ≤ grpOff V (· = v)

theorem **grpOff_eq_top** (V : HSp D) {P : Pl D → Prop} (hP : ¬ ∃ v, P v) : grpOff V P = ⊤

theorem **grpAt_eq_bot** (V : HSp D) {P : Pl D → Prop} (hP : ¬ ∃ v, P v) : grpAt V P = ⊥

theorem **latO_eq_bot** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) {P : Pl D → Prop} (hP : ¬ ∃ v, P v) : latO S T LS LT P = ⊥

theorem **heisO_zero_eq** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **suppFirst_holds** (D : D7) : SuppFirst D

theorem **GlobalTheory.FinPl.suppFirst_all** : ∀ D : D7, GlobalTheory.FinPl.SuppFirst D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFirstStepFlatMapSupportAllStatement

theorem **_root_.GlobalTheory.T.besselToJacobiFirstStepFlatMapSupport_all** : ∀ D : D7, GlobalTheory.FinPl.SuppFirst D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatMapSupportReducesToTorus

theorem **oneBF_red_torus** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → UnrAt s v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → ∀ n ∈ flagN J.d, n ∈ grpAt J.W (· = v) → ∀ t : GL (Fin ρ) (AE D), ∀ γ ∈ grpAt J.W₀ (· = v), ∀ κ, InK J.W LW (· = v) κ → 𝔉 (s.om (1, n * J.d.levi t * J.d.emb0 γ * κ) f) ≠ 0 → 𝔉 (s.om (1, J.d.levi t) f) ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatMapSupportReducesToTorusStatement

theorem **_root_.GlobalTheory.T.besselToJacobiFlatMap_support_reduces_to_torus** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → UnrAt s v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → ∀ n ∈ flagN J.d, n ∈ grpAt J.W (· = v) → ∀ t : GL (Fin ρ) (AE D), ∀ γ ∈ grpAt J.W₀ (· = v), ∀ κ, InK J.W LW (· = v) κ → 𝔉 (s.om (1, n * J.d.levi t * J.d.emb0 γ * κ) f) ≠ 0 → 𝔉 (s.om (1, J.d.levi t) f) ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatMapVacuumAtGoodPlace

theorem **smul_mem_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (fun i => ((1 : AF D.host.F) ⊗ₜ[D.host.F] a) * y i) ∈ latA S L P

theorem **wittCoreInclusion_mem_latticeCompletionAtPlace** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {L : Lat T} {L₀ : Lat T₀} {v : Pl D} (hm : MonoAt d L L₀ v) {u : Fin T₀.rank → AE D} (hu : u ∈ latV T₀ L₀ (· = v)) : d.j0 *ᵥ u ∈ latV T L (· = v)

theorem **wittCoreInclusion_mem_latticeCompletionAAScaledAtPlace** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {L : Lat T} {L₀ : Lat T₀} {v : Pl D} (hm : MonoAt d L L₀ v) {z : Fin T₀.rank → AE D} (hz : z ∈ latA T₀ L₀ (· = v)) : d.j0 *ᵥ z ∈ latA T L (· = v)

theorem **wittExt_wittInc_mem_completed_hom_lattice** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (dS : HypDec S S₀ ρ) (dT : HypDec T T₀ σ) {LS : Lat S} {LS₀ : Lat S₀} {LT : Lat T} {LT₀ : Lat T₀} {v : Pl D} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (hS₀ : IsSelfDualC S₀ LS₀) (hmS : MonoAt dS LS LS₀ v) (hmT : MonoAt dT LT LT₀ v) {w : WO S₀ T₀} (hw : w ∈ latO S₀ T₀ (LS₀ : Set _) (LT₀ : Set _) (· = v)) : dS.ext0 T (dT.inc0 S₀ w) ∈ latO S T (LS : Set _) (LT : Set _) (· = v)

theorem **wittExt_wittInc_isHalf** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (dS : HypDec S S₀ ρ) (dT : HypDec T T₀ σ) {e : AE D} {w : WO S₀ T₀} (hw : IsHalf e w) : IsHalf e (dS.ext0 T (dT.inc0 S₀ w))

theorem **flat_vac** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → Vac ψ B.V₀ J.W₀ LV₀ LW₀ (· = v) (𝔉 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatMapVacuumAtGoodPlaceStatement

theorem **_root_.GlobalTheory.T.besselToJacobi_flatMap_vacuum_at_goodPlace** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → Vac ψ B.V₀ J.W₀ LV₀ LW₀ (· = v) (𝔉 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatSchwartzEquivariant

theorem **levi_mem_flagN** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : d.levi u ∈ flagN d

theorem **psiUni_mul** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {u v : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) (hv : v ∈ UX D ρ) : psiUni ψ (u * v) = psiUni ψ u * psiUni ψ v

theorem **flatBF_levi** (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W) (hom : IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om') (Ψ : WO B.V J.W → ℂ) (hΨ : IsThetaFunO ψ B.V J.W Ψ) (γ : UX D ρ) : flatBF s (s.om (1, J.d.levi γ.1) Ψ) = star (psiUni ψ γ.1) • flatBF s Ψ

theorem **flatBF_equivariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → ∀ h₀, ∀ n ∈ flagN J.d, ∀ h', flatBF s (s.om (embA B.d B.o h₀, n * J.d.emb0 h') Ψ) = star (psiFlag ψ J.d n) • s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (flatBF s Ψ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatSchwartzEquivariantStatement

theorem **_root_.GlobalTheory.T.besselToJacobiFlatSchwartz_equivariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → ∀ h₀, ∀ n ∈ flagN J.d, ∀ h', flatBF s (s.om (embA B.d B.o h₀, n * J.d.emb0 h') Ψ) = star (psiFlag ψ J.d n) • s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (flatBF s Ψ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatSchwartzLinear

theorem **mixS_add** (hψ : IsStdChar D ψ (-1)) {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) (x : Mx D T.rank ρ) : d.mixS T ψ v.μ v.ℱ (f + g) x = d.mixS T ψ v.μ v.ℱ f x + d.mixS T ψ v.μ v.ℱ g x

theorem **cont_bdd_of_tame** {Z : Type} [TopologicalSpace Z] {F : Z → ℂ} (t : GlobalTheory.Unf.EvalCont.Tame fun (_ : Unit) (q : Unit × Z) => F q.2) : Continuous F ∧ ∃ C : ℝ, ∀ w, ‖F w‖ ≤ C

theorem **mixS_cont_bdd** (hψ : IsStdChar D ψ (-1)) {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (x : Mx D T.rank ρ) : Continuous (d.mixS T ψ v.μ v.ℱ f x) ∧ ∃ C : ℝ, ∀ w, ‖d.mixS T ψ v.μ v.ℱ f x w‖ ≤ C

theorem **integrableOn_mixT** (hψ : IsStdChar D ψ (-1)) {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (v : VolM D ρ S.rank) {f : WO S T → ℂ} (hf : Continuous f ∧ ∃ C : ℝ, ∀ w, ‖f w‖ ≤ C) (y : Mx D S.rank ρ) (w₀ : WO S T₀) : IntegrableOn (fun ξ => ψ (symplOH S T (d.incX S ξ) (d.incAdj S y)) * f (d.incX S ξ + d.incAdj S y + d.inc0 S w₀)) v.ℱ v.μ

theorem **mixT_add** (hψ : IsStdChar D ψ (-1)) {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (v : VolM D ρ S.rank) {f g : WO S T → ℂ} (hf : Continuous f ∧ ∃ C : ℝ, ∀ w, ‖f w‖ ≤ C) (hg : Continuous g ∧ ∃ C : ℝ, ∀ w, ‖g w‖ ≤ C) (y : Mx D S.rank ρ) : d.mixT S ψ v.μ v.ℱ (f + g) y = d.mixT S ψ v.μ v.ℱ f y + d.mixT S ψ v.μ v.ℱ g y

theorem **evalAt_add** (hψ : IsStdChar D ψ (-1)) {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) {f g : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (hg : IsThetaFunO ψ V W g) (x : Mx D W.rank ρ) (y : Mx

D V₀.rank σ) : GlobalTheory.MixPer.evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (f + g) x y = GlobalTheory.MixPer.evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 f x y + GlobalTheory.MixPer.evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 g x y

theorem **flatAt_add** (hF : FlatLaw D) (hψ : IsStdChar D ψ (-1)) {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E) (hx : LinearIndependent D.host.E x.transpose) (y : Mx D V₀.rank σ) {f g : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (hg : IsThetaFunO ψ V W g) : GlobalTheory.MixPer.flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν (rat x) y (f + g) = GlobalTheory.MixPer.flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν (rat x) y f + GlobalTheory.MixPer.flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν (rat x) y g

theorem **flatBF_linear** : ∀ (D : D7), FlatLaw D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (f g : WO B.V J.W → ℂ) (a : ℂ), IsThetaFunO ψ B.V J.W f → IsThetaFunO ψ B.V J.W g → flatBF st (f + g) = flatBF st f + flatBF st g ∧ flatBF st (a • f) = a • flatBF st f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatSchwartzLinearStatement

theorem **_root_.GlobalTheory.T.besselToJacobiFlatSchwartz_linear** : ∀ (D : D7), FlatLaw D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (f g : WO B.V J.W → ℂ) (a : ℂ), IsThetaFunO ψ B.V J.W f → IsThetaFunO ψ B.V J.W g → flatBF st (f + g) = flatBF st f + flatBF st g ∧ flatBF st (a • f) = a • flatBF st f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatWeilRepresentationCores

theorem **sFinite_of_haar** {G : Type} [Group G] [TopologicalSpace G] [MeasurableSpace G] [LocallyCompactSpace G] [SecondCountableTopology G] (ν : Measure G) (h : ν.IsHaarMeasure) : SFinite ν

theorem **continuous_psiUni** (ψ : AddChar (AF D.host.F) ℂ) (hψ : Continuous ψ) (ρ : ℕ) : Continuous fun u : UX D ρ => psiUni ψ u.1

theorem **jacT_integral** (ψ : AddChar (AF D.host.F) ℂ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) (n : GA D T) {U : Type} [MeasurableSpace U] (m : Measure U) (G : U → WO S T₀ → ℂ) (w : WO S T₀) : jacT ψ d S y n (fun w' => ∫ u, G u w' ∂m) w = ∫ u, jacT ψ d S y n (G u) w ∂m

theorem **jacT_smul** (ψ : AddChar (AF D.host.F) ℂ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) (n : GA D T) (c : ℂ) (f : WO S T₀ → ℂ) : jacT ψ d S y n (c • f) = c • jacT ψ d S y n f

theorem **flatBF_om** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → ∀ h₀, ∀ n ∈ radX J.d, ∀ h', flatBF s (s.om (embA B.d B.o h₀, n ∗ J.d.emb0 h') Ψ) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (flatBF s Ψ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiFlatWeilRepresentationCoresStatement

theorem **_root_.GlobalTheory.T.besselToJacobiFlat_weilRepresentation_cores** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → ∀ h₀, ∀ n ∈ radX J.d, ∀ h', flatBF s (s.om (embA B.d B.o h₀, n ∗ J.d.emb0 h') Ψ) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (flatBF s Ψ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiMapCarriesNormalFrameLines

theorem **colsAt_xZero_of_normalFr** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊛ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), GlobalTheory.FinPl.ColsAt ((y.bes j ρ h).

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselToJacobiMapCarriesNormalFrameLinesStatement

theorem **_root_.GlobalTheory.T.besselToJacobiMap_carries_normalFrameLines** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊛ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 ∗ ρ + 2 = D.R.m), G

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselUnfoldingEvaluationIntegrandContinuousInva

theorem **fund_lt_top** {C : Set H} (hC : IsCompact C) (hcov : ∀ g, ∃ γ ∈ Γ, γ ∗ g ∈ C) (h𝓕 : IsFundamentalDomain Γ 𝓕 μ) : μ 𝓕 < ⊤

theorem **mean_mul_right** [SecondCountableTopology H] (h𝓕 : IsFundamentalDomain Γ 𝓕 μ) (hfin : μ 𝓕 < ⊤) (f : H → ℂ) (hf : ∀ γ ∈ Γ, ∀ x, f (γ ∗ x) = f x) (r : H) : ∫ x in 𝓕, f (x ∗ r) ∂μ = ∫ x in 𝓕, f x ∂μ

theorem **mean_continuous** [SecondCountableTopology H] {C : Set H} (hC : IsCompact C) (hcov : ∀ g, ∃ γ ∈ Γ, γ ∗ g ∈ C) (h𝓕 : IsFundamentalDomain Γ 𝓕 μ) {P : Type} [TopologicalSpace P] [FirstCountableTopology P] [LocallyCompactSpace P] (K : H → P → ℂ) (hK : Continuous fun q : H × P => K q.1 q.2) (hinv : ∀ γ ∈ Γ, ∀ x p, K (γ ∗ x) p = K x p) : (∀ p, IntegrableOn (fun x => K x p) 𝓕 μ) ∧ Continuous fun p => ∫ x in 𝓕, K x p ∂μ

theorem **jac_cocompact** {W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (hc : CoCpt W₀) : ∃ C : Set (jacGrp d), IsCompact C ∧ ∀ g, ∃ γ ∈ (GQ W ⊓ jacGrp d).subgroupOf (jacGrp d), γ ∗ g ∈ C

theorem **jac_mean** {ρ : ℕ} (J : JacLev D ρ) (K : GA D J.W → GA D J.W → ℂ) (hK : Continuous fun q : jacGrp J.d × GA D J.W => K q.1 q.2) (hinv : ∀ γ ∈ GQ J.W, γ ∈ jacGrp J.d → ∀ r ∈ jacGrp J.d, ∀ p, K (γ ∗ r) p = K r p) : (Continuous fun p => ∫ h in J.v0.𝓕, ∫ n in J.vN.𝒩, K (n.1 ∗ J.d.emb0 h) p ∂J.vN.ν ∂J.v0.μ) ∧ ∀ r ∈ jacGrp J.d, ∀ p p', (∀ x ∈ jacGrp J.d, K x p' = K (x ∗ r) p) → ∫ h in J.v0.𝓕, ∫ n in J.vN.𝒩, K (n.1 ∗ J.d.emb0 h) p ∂J.vN.ν ∂J.v0.μ = ∫ h in J.v0.𝓕, ∫ n in J.vN.𝒩, K (n.1 ∗ J.d.emb0 h) p' ∂J.vN.ν ∂J.v0.μ

theorem **yZero_rat** (S : HSp D) (ρ : ℕ) (z : Fin S.rank → D.host.E) : yZero S ρ z ∈ ratMx D S.rank ρ

theorem **jacT_rat** (hψ : IsStdChar D ψ (-1)) {T T₀ : HSp D} (d : HypDec T T₀ ρ) (S : HSp D) {y : Mx D S.rank ρ} (hy : y ∈ ratMx D S.rank ρ) {n : GA D T} (hn : n ∈ GQ T) {f : WO S T₀ → ℂ} (hf : IsThetaFunO ψ S T₀ f) : jacT ψ d S y n f 0 = f 0

theorem **ev_theta** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (x : Mx D J.W.rank ρ) : IsThetaFunO ψ B.V₀ J.W₀ (ev s Ψ x)

theorem **om_mul** (a b : GA D B.V) (g h : GA D J.W) (Ψ : WO B.V J.W → ℂ) : s.om (a ∗ b, g ∗ h) Ψ = s.om (a, g) (s.om (b, h) Ψ)

theorem **ev_om** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (u : UX D ρ) (h₀ : GA D B.A) {n : GA D J.W} (hn : n ∈ flagN J.d) (h' : GA D J.W₀) : ev s (s.om (embA B.d B.o h₀, n ∗ J.d.emb0 h') Ψ) (mt J.W (n ∗ J.d.emb0 h') · (J.d.jX · u.1.1)) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (ev s Ψ (J.d.jX · u.1.1)))

theorem **exists_uni** {r : GA D J.W} (hr : r ∈ jacGrp J.d) : ∃ u : UX D ρ, mt J.W r · J.d.jX = J.d.jX · u.1.1

theorem **ker_jac** {γ : GA D J.W} (hγ : γ ∈ GQ J.W) (hγ' : γ ∈ jacGrp J.d) {r : GA D J.W} (hr : r ∈ jacGrp J.d) (h₀ h) : ker s φ Φ h₀ (γ ∗ r) h = ker s φ Φ h₀ r h

theorem **ker_eq** {n : GA D J.W} (hn : n ∈ flagN J.d) (h' h₀ h) : ker s φ Φ h₀ (n ∗ J.d.emb0 h') h = φ h₀ ∗ fjKer ψ J.d B.o B.ζ s.om₀ (evalBF s (s.om (1, h) Φ)) h₀ n h'

theorem **ker_left** {γ : GA D B.A} (hγ : γ ∈ GQ B.A) {r : GA D J.W} (hr : r ∈ jacGrp J.d) (h₀ h) : ker s φ Φ (γ ∗ h₀) r h = ker s φ Φ h₀ r h

theorem **ker_cont** : Continuous fun q : GA D B.A × jacGrp J.d × GA D J.W => ker s φ Φ q.1 q.2.1 q.2.2

theorem **psiFlag_jac** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} {W W₀ : HSp D} (d : HypDec W W₀ ρ) {γ : GA D W} (hγ : γ ∈ GQ W) (hγ' : γ ∈ jacGrp d) {r : GA D W} (hr : r ∈ jacGrp d) : psiFlag ψ d (γ ∗ r) = psiFlag ψ d r

theorem **evalIntegrand_continuous_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → (Continuous fun h => J.perAll ψ B s.om₀ φ (rTrans h G) (evalBF s (s.om (1, h) Φ))) ∧ ∀ r ∈ radX J.d ⊔ J.d.emb0.range, ∀ h, J.perAll ψ B s.om₀ φ (rTrans (r ∗ h) G) (evalBF s (s.om (1, (r ∗ h)) Φ)) = J.perAll ψ B s.om₀ φ (rTrans h G) (evalBF s (s.om (1, h) Φ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselUnfoldingEvaluationIntegrandContinuousInvaStatement

theorem **_root_.GlobalTheory.T.besselUnfolding_evaluationIntegrand_continuous_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → (Continuous fun h => J.perAll ψ B s.om₀ φ (rTrans h G) (evalBF s (s.om (1, h) Φ))) ∧ ∀ r ∈ radX J.d ⊔ J.d.emb0.range, ∀ h, J.perAll ψ B s.om₀ φ (rTrans (r ∗ h) G) (evalBF s (s.om (1, (r ∗ h)) Φ)) = J.perAll ψ B s.om₀ φ (rTrans h G) (evalBF s (s.om (1, h) Φ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselUnfoldingOrbitMemParabolicMulStabiliser

theorem **mem_congGrp_iff'** (V : HSp D) (I : Ideal (AE D)) (g : GA D V) : g ∈ congGrp V I ↔ (mt V g).map (Ideal.Quotient.mk I) = 1

theorem **matrixTransform_wittCoreEmbedding** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (h : GA D S₀) : mt S (d.emb0 h) = d.jX · d.pX + d.j0 · mt S₀ h · d.p0 + d.jY · d.pY

theorem **wittCoreEmbedding_mem_supportedSubgroup** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (P : Pl D → Prop) {h : GA D S₀} (hh : h ∈ grpAt S₀ P) : d.emb0 h ∈ grpAt S P

theorem **emb_mem_grpAt'** {S A B : HSp D} (o : OrthDec S A B) {P : Pl D → Prop} {a : GA D A} (ha : a ∈ grpAt A P) : o.emb (a, 1) ∈ grpAt S P

theorem **inK_mul'** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a b : GA D S} (ha : InK S L P a) (hb : InK S L P b) : InK S L P (a ∗ b)

theorem **orbit_coset** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ) (L : Lat B.V) (L₀ : Lat B.V₀) (v : Pl D), MonoAt B.d L L₀ v → ∀ n ∈ flagN B.d, n ∈ grpAt B.V (· = v) → ∀ a ∈ grpAt B.A (· = v), ∀ κ₀, InK B.V₀ L₀ (· = v) κ₀ → ∀ κ, InK B.V L (· = v) κ → ∃ r ∈ besGrp B.d B.o, r ∈ grpAt B.V (· = v) ∧ ∃ κ', InK B.V L (· = v) κ' ∧ n ∗ B.d.emb0 (B.o.emb (a, 1) ∗ κ₀) ∗ κ = r ∗ κ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselUnfoldingOrbitMemParabolicMulStabiliserStatement

theorem **_root_.GlobalTheory.T.besselUnfolding_orbit_mem_parabolic_mul_stabiliser** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ) (L : Lat B.V) (L₀ : Lat B.V₀) (v : Pl D), MonoAt B.d L L₀ v → ∀ n ∈ flagN B.d, n ∈ grpAt B.V (· = v) → ∀ a ∈ grpAt B.A (· = v), ∀ κ₀, InK B.V₀ L₀ (· = v) κ₀ → ∀ κ, InK B.V L (· = v) κ → ∃ r ∈ besGrp B.d B.o, r ∈ grpAt B.V (· = v) ∧ ∃ κ', InK B.V L (· = v) κ' ∧ n ∗ B.d.emb0 (B.o.emb (a, 1) ∗ κ₀) ∗ κ = r ∗ κ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BesselWittDecomposition

theorem **val_lt_one_iff** (x : K) : valuation K x < 1 ↔ Tendsto (fun n : ℕ => x ^ n) atTop (𝓝 0)

theorem **ι_mul** (p : Place D k) (x y : k) : p.ι (x ∗ y) = p.ι x ∗ p.ι y

theorem **ι_one** (p : Place D k) : p.ι 1 = p.e

theorem **π_e** (p : Place D k) : p.π p.e = 1

theorem **ι_xi** (p : Place D k) (x : k) : p.ι (p.xi ∗ x) = incA D D.host.xi0 ∗ p.ι x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BeszPairings

theorem **besz_pairings** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ρ : ℕ) (B : BesLev D ρ) (s : k) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (FA : Frame k B.A.rank n) (F : Frame k B.V.rank (n ⊕ Fin c)), p.IsBesFrame B s FA F → valuation k s = 1 → valuation k (p.hz B) = 1 → ∃ m₀ : n ⊕ Fin c, (∀ r, s ∗ p.pr ((B.d.i₀ *ᵥ vecB B.o B.ζ) r) = F.β r m₀) ∧ (∀ m, ∑ r, F.β r m ∗ p.pr (star ((B.V.gram *ᵥ (B.d.i₀ *ᵥ vecB B.o B.ζ)) r)) ∈ 𝒪[k]) ∧ ∃ m, valuation k (∑ r, F.β r m ∗ p.pr (star ((B.V.gram *ᵥ (B.d.i₀ *ᵥ vecB B.o B.ζ)) r))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BiCombEqLOpROp

theorem **biComb_eq_lOp_rOp** (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' c c' C = lOp ψ A S A' c (rOp ψ A S A' c' C)

theorem **lOp_rOp** (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (C : WO A S → WO S A' → ℂ) : lOp ψ A S A' c (rOp ψ A S A' c' C) = rOp ψ A S A' c' (lOp ψ A S A' c C)

theorem **lOp_lOp** (c d e : WO A S →₀ ℂ) (h : ∀ f, comb ψ A S e f = comb ψ A S c (comb ψ A S d f)) (C : WO A S → WO S A' → ℂ) : lOp ψ A S A' c (lOp ψ A S A' d C) = lOp ψ A S A' e C

theorem **rOp_rOp** (c d e : WO S A' →₀ ℂ) (h : ∀ f, comb ψ S A' e f = comb ψ S A' c (comb ψ S A' d f)) (C : WO A S → WO S A' → ℂ) : rOp ψ A S A' c (rOp ψ A S A' d C) = rOp ψ A S A' e C

theorem **lOp_comm** (c d : WO A S →₀ ℂ) (h : ∀ f, comb ψ A S c (comb ψ A S d f) = comb ψ A S d (comb ψ A S c f)) (C : WO A S → WO S A' → ℂ) : lOp ψ A S A' c (lOp ψ A S A' d C) = lOp ψ A S A' d (lOp ψ A S A' c C)

theorem **rOp_comm** (c d : WO S A' →₀ ℂ) (h : ∀ f, comb ψ S A' c (comb ψ S A' d f) = comb ψ S A' d (comb ψ S A' c f)) (C : WO A S → WO S A' → ℂ) : rOp ψ A S A' c (rOp ψ A S A' d C) = rOp ψ A S A' d (rOp ψ A S A' c C)

theorem **biComb_comp** (c d e : WO A S →₀ ℂ) (c' d' e' : WO S A' →₀ ℂ) (hl : ∀ f, comb ψ A S e f = comb ψ A S c (comb ψ A S d f)) (hr : ∀ f, comb ψ S A' e' f = comb ψ S A' c' (comb ψ S A' d' f)) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' c c' (biComb ψ A S A' d d' C) = biComb ψ A S A' e e' C

theorem **biComb_comm** (c d : WO A S →₀ ℂ) (c' d' : WO S A' →₀ ℂ) (hl : ∀ f, comb ψ A S c (comb ψ A S d f) = comb ψ A S d (comb ψ A S c f)) (hr : ∀ f, comb ψ S A' c' (comb ψ S A' d' f) = comb ψ S A' d' (comb ψ S A' c' f)) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' c c' (biComb ψ A S A' d d' C) = biComb ψ A S A' d d' (biComb ψ A S A' c c' C)

theorem **comb_single_zero** (T T' : HSp D) (f : WO T T' → ℂ) : comb ψ T T' (Finsupp.single 0 1) f = f

theorem **comm_foldr** {α ι : Type} (F : α → α) (H : ι → α → α) (l : List ι) (h : ∀ i ∈ l, ∀ x, F (H i x) = H i (F x)) (x : α) : F (l.foldr H x) = l.foldr H (F x)

theorem **foldr_mem** {α ι : Type} (H : ι → α → α) (s : Set α) (l : List ι) (h : ∀ i ∈ l, ∀ x ∈ s, H i x ∈ s) (x : α) (hx : x ∈ s) : l.foldr H x ∈ s

theorem **prodList** {ι' : Type} (e₁ : ι' → WO A S →₀ ℂ) (e₂ : ι' → WO S A' →₀ ℂ) (l : List ι') : ∃ (e : WO A S →₀ ℂ) (e' : WO S A' →₀ ℂ), (∀ w ∈ e.support, w ∈ AddSubmonoid.closure (⋃ i ∈ l, ((e₁ i).support : Set (WO A S)))) ∧ (∀ w ∈ e'.support, w ∈ AddSubmonoid.closure (⋃ i ∈ l, ((e₂ i).support : Set (WO S A')))) ∧ ∀ C, biComb ψ A S A' e e' C = l.foldr (fun i C => biComb ψ A S A' (e₁ i) (e₂ i) C) C

theorem **keyList** {ι' : Type} (e₁ g₁ : ι' → WO A S →₀ ℂ) (e₂ g₂ : ι' → WO S A' →₀ ℂ) (Cl Vl : ι' → Set (WO A S → WO S A' → ℂ)) (l : List ι') (hl : l.Nodup) (horth : ∀ i ∈ l, ∀ i' ∈ l, i ≠ i' → (∀ x, x ∈ (e₁ i).support ∨ x ∈ (g₁ i).support → ∀ y ∈ (e₁ i').support, symplO A S x y = 0) ∧ ∀ x, x ∈ (e₂ i).support ∨ x ∈ (g₂ i).support → ∀ y ∈ (e₂ i').support, symplO S A' x y = 0) (hloc : ∀ i ∈ l, ∀ C ∈ Cl i, biComb ψ A S A' (e₁ i) (e₂ i) C ∈ Vl i ∧ C = biComb ψ A S A' (g₁ i) (g₂ i) (biComb ψ A S A' (e₁ i) (e₂ i) C)) (hstab : ∀ i ∈ l, ∀ i' ∈ l, i ≠ i' → ∀ C ∈ Cl i, biComb ψ A S A' (e₁ i') (e₂ i') C ∈ Cl i) (C : WO A S → WO S A' → ℂ) (hC : ∀ i ∈ l, C ∈ Cl i) : (∀ i ∈ l, l.foldr (fun i C => biComb ψ A S A' (e₁ i) (e₂ i) C) C ∈ Vl i) ∧ C = l.foldr (fun i C => biComb ψ A S A' (g₁ i) (g₂ i) C) (l.foldr (fun i C => biComb ψ A S A' (e₁ i) (e₂ i) C) C)

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (ι' P : Type) (l : List ι') (X : P → WO A S → WO S A' → ℂ) (e₁ g₁ : ι' → WO A S →₀ ℂ) (e₂ g₂ : ι' → WO S A' →₀ ℂ) (Cl Vl : ι' → Set (WO A S → WO S A' → ℂ)), l.Nodup → (∀ i ∈ l, ∀ i' ∈ l, i ≠ i' → (∀ x, x ∈ (e₁ i).support ∨ x ∈ (g₁ i).support → ∀ y ∈ (e₁ i').support, symplO A S x y = 0) ∧ ∀ x, x ∈ (e₂ i).support ∨ x ∈ (g₂ i).support → ∀ y ∈ (e₂ i').support, symplO S A' x y = 0) → (∀ i ∈ l, ∀ p, X p ∈ Cl i) → (∀ i ∈ l, ∀ C ∈ Cl i, biComb ψ A S A' (e₁ i) (e₂ i) C ∈ Vl i ∧ C = biComb ψ A S A' (g₁ i) (g₂ i) (biComb ψ A S A' (e₁ i) (e₂ i) C)) → (∀ i ∈ l, ∀ i' ∈ l, i ≠ i' → ∀ C ∈ Cl i, biComb ψ A S A' (e₁ i') (e₂ i') C ∈ Cl i) → ∃ (e : WO A S →₀ ℂ) (e' : WO S A' →₀ ℂ), (∀ w ∈ e.support, w ∈ AddSubmonoid.closure (⋃ i ∈ l, ((e₁ i).support : Set (WO A S)))) ∧ (∀ w ∈ e'.support, w ∈ AddSubmonoid.closure (⋃ i ∈ l, ((e₂ i).suppo

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BiInvariantEigenfunctionEqMulZonalSpherical

theorem **genVal_top_ne_zero** (q N : ℕ) (α : Fin N → ℂˣ) : genVal q 0 N (Finset.univ.val.map α) N ≠ 0

theorem **eq_mul_zonalInt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) [MeasurableSpace (GL (Fin N) k)] [BorelSpace (GL (Fin N) k)] (μK : Measure (GL (Fin N) k)) [μK.IsHaarMeasure], μK (KO k (Fin N)) = 1 → ∀ (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ α : Fin N → ℂˣ, ∃ ψ : GL (Fin N) k → ℂ, IsBorelSph k (borelParam (resCard k) α) ψ ∧ ∀ f : GL (Fin N) k → ℂ, (∀ κ ∈ KO k (Fin N), ∀ m, f (κ ∗ m) = f m ∧ f (m ∗ κ) = f m) → (∀ (s : Finset (Fin N)) R, IsRepsK (torusElt k (Fin N) ϖ (indic s)) R → ∀ m, ∑ g ∈ R, f (m ∗ g) = genVal (resCard k) 0 N (Finset.univ.val.map α) s.card ∗ f m) → ∀ m, f m = f 1 ∗ zonalInt μK ψ m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BiInvariantEigenfunctionEqMulZonalSphericalStatement

theorem **_root_.GlobalTheory.T.biInvariant_eigenfunction_eq_mul_zonalSpherical** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) [MeasurableSpace (GL (Fin N) k)] [BorelSpace (GL (Fin N) k)] (μK : Measure (GL (Fin N) k)) [μK.IsHaarMeasure], μK (KO k (Fin N)) = 1 → ∀ (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ α : Fin N → ℂˣ, ∃ ψ : GL (Fin N) k → ℂ, IsBorelSph k (borelParam (resCard k) α) ψ ∧ ∀ f : GL (Fin N) k → ℂ, (∀ κ ∈ KO k (Fin N), ∀ m, f (κ ∗ m) = f m ∧ f (m ∗ κ) = f m) → (∀ (s : Finset (Fin N)) R, IsRepsK (torusElt k (Fin N) ϖ (indic s)) R → ∀ m, ∑ g ∈ R, f (m ∗ g) = genVal (resCard k) 0 N (Finset.univ.val.map α) s.card ∗ f m) → ∀ m, f m = f 1 ∗ zonalInt μK ψ m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BiThetaOfVac

theorem **slice_left** {A S A' : HSp D} {K : WO A S → WO S A' → ℂ} (hK : IsBiTheta ψ A S A' K) (v : WO S A') : IsThetaFunO ψ A S (fun u => K u v)

theorem **slice_right** {A S A' : HSp D} {K : WO A S → WO S A' → ℂ} (hK : IsBiTheta ψ A S A' K) (u : WO A S) : IsThetaFunO ψ S A' (K u)

theorem **actO_real** {S T : HSp D} {g : GA D S} {h : GA D T} (hg : g ∈ GinfC S) (hh : h ∈ GinfC T) {x : WO S T} (hx : x.map (infPart D) = 0) : actO S T g h x = x

theorem **nsmul_fin** {S T : HSp D} {x : WO S T} (hx : x.map (infPart D) = 0) (n : ℕ) : (n • x).map (infPart D) = 0

theorem **latVac_om** {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {g : GA D S} {h : GA D T} (hg : g ∈ GinfC S) (hh : h ∈ GinfC T) {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hl : IsLatVac (heisO ψ S T) (latO S T LS LT fun _ => True) f) : IsLatVac (heisO ψ S T) (latO S T LS LT fun _ => True) (om (g, h) f)

theorem **latVac_opR** {A S A' : HSp D} {Λ : AddSubgroup (WO A S)} {K : WO A S → WO S A' → ℂ} (hK : ∀ v, IsLatVac (heisO ψ A S) Λ (fun u => K u v)) (T : Module.End ℂ (WO S A' → ℂ)) (v : WO S A') : IsLatVac (heisO ψ A S) Λ (fun u => T (K u) v)

theorem **latVac_opL** {A S A' : HSp D} {Λ : AddSubgroup (WO S A')} {K : WO A S → WO S A' → ℂ} (hK : ∀ u, IsLatVac (heisO ψ S A') Λ (K u)) (T : Module.End ℂ (WO A S → ℂ)) (u : WO A S) : IsLatVac (heisO ψ S A') Λ (fun v => T (fun u' => K u' v) u)

theorem **eq_two_smul** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {x : WO S T} (hx : x ∈ latNbhd S T LS LT) (hx' : x.map (infPart D) = 0) : ∃ l ∈ latO S T LS LT (fun _ => True), x = 2 • l

theorem **biTheta_of_vac** {A S A' : HSp D} {LA : Submodule (𝒪 D.host.E) (Fin A.rank → D.host.E)} {LS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {LA' : Submodule (𝒪 D.host.E) (Fin A'.rank → D.host.E)} (hLA : IsSelfDualC A LA) (hLS : IsSelfDualC S LS) (hLA' : IsSelfDualC A' LA') {C : WO A S → WO S A' → ℂ} (hv : IsBiVac ψ A S A' LA LS LA' C) (hs : IsBiSmooth C) : IsBiTheta ψ A S A' C

theorem **om_comb** {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {g : GA D S} {h : GA D T} (hg : g ∈ GinfC S) (hh : h ∈ GinfC T) {c : WO S T →₀ ℂ} (hc : IsFinComb c) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (comb ψ S T c f) = comb ψ S T c (om (g, h) f)

theorem **op_combR** {A S A' : HSp D} (T : Module.End ℂ (WO A S → ℂ)) (c : WO S A' →₀ ℂ) (K : WO A S → WO S A' → ℂ) (v : WO S A') : T (fun u => comb ψ S A' c (K u) v) = fun u => comb ψ S A' c (fun v' => T (fun u' => K u' v') u) v

theorem **op_combL** {A S A' : HSp D} (T : Module.End ℂ (WO S A' → ℂ)) (c : WO A S →₀ ℂ) (K : WO A S → WO S A' → ℂ) (u : WO A S) : T (fun v => comb ψ A S c (fun u' => K u' v) u) = fun v => comb ψ A S c (fun u' => T (K u') v) u

theorem **comb_apply** {S T : HSp D} (c : WO S T →₀ ℂ) (g : WO S T → ℂ) (u : WO S T) : comb ψ S T c g u = ∑ x ∈ c.support, c x * (ψ (symplOH S T u x) * g (u + x))

theorem **isOpen_ker** {S T : HSp D} (hψ : IsStdChar D ψ (-1)) {x : WO S T} (hx : x.map (infPart D) = 0) : IsOpen {w : WO S T | ψ (symplOH S T w x) = 1}

theorem **exists_orth** {S T : HSp D} (hψ : IsStdChar D ψ (-1)) {c : WO S T →₀ ℂ} (hc : IsFinComb c) : ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ w ∈ U, ∀ x ∈ c.support, ψ (symplOH S T w x) = 1

theorem **heisO_comb** {S T : HSp D} {c : WO S T →₀ ℂ} {w : WO S T} (hw : ∀ x ∈ c.support, ψ (symplOH S T w x) = 1) {g : WO S T → ℂ} (hg : heisO ψ S T w g = g) : heisO ψ S T w (comb ψ S T c g) = comb ψ S T c g

theorem **biTheta_combL** {A S A' : HSp D} (hψ : IsStdChar D ψ (-1)) {K : WO A S → WO S A' → ℂ} (hK : IsBiTheta ψ A S A' K) {c : WO A S →₀ ℂ} (hc : IsFinComb c) : IsBiTheta ψ A S A' (fun u v => comb ψ A S c (fun u' => K u' v) u)

theorem **biTheta_combR** {A S A' : HSp D} (hψ : IsStdChar D ψ (-1)) {K : WO A S → WO S A' → ℂ} (hK : IsBiTheta ψ A S A' K) {c : WO S A' →₀ ℂ} (hc : IsFinComb c) : IsBiTheta ψ A S A' (fun u v => comb ψ S A' c (K u) v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BiVacuumOperatorZeroOrLeftInvertible

theorem **twoT_apply** (x : WO A S × WO S A') (C : WO A S → WO S A' → ℂ) (u : WO A S) (v : WO S A') : twoT ψ A S A' x C u v = ψ (symplOH A S u x.1) * (ψ (symplOH S A' v x.2) * C (u + x.1) (v + x.2))

theorem **symplOH_add_l** (T T' : HSp D) (x x' y : WO T T') : symplOH T T' (x + x') y = symplOH T T' x y + symplOH T T' x' y

theorem **symplOH_add_r** (T T' : HSp D) (x y y' : WO T T') : symplOH T T' x (y + y') = symplOH T T' x y + symplOH T T' x y'

theorem **symplOH_two** (T T' : HSp D) (x y : WO T T') : symplOH T T' x y + symplOH T T' x y = symplO T T' x y

theorem **symplO_skew'** (T T' : HSp D) (x y : WO T T') : symplO T T' y x = -symplO T T' x y

theorem **psi_ne_zero** (z : AF D.host.F) : ψ z ≠ 0

theorem **twoT_zero** (C : WO A S → WO S A' → ℂ) : twoT ψ A S A' 0 C = C

theorem **twoT_mul** (x y : WO A S × WO S A') (C : WO A S → WO S A' → ℂ) : twoT ψ A S A' x (twoT ψ A S A' y C) = bet ψ A S A' x y • twoT ψ A S A' (x + y) C

theorem **bet_ne_zero** (x y : WO A S × WO S A') : bet ψ A S A' x y ≠ 0

theorem **twoT_comm** (x γ : WO A S × WO S A') (C : WO A S → WO S A' → ℂ) : twoT ψ A S A' γ (twoT ψ A S A' x C) = chi ψ A S A' x γ • twoT ψ A S A' x (twoT ψ A S A' γ C)

theorem **comb_single** (T T' : HSp D) (x : WO T T') (c : ℂ) (f : WO T T' → ℂ) : comb ψ T T' (Finsupp.single x c) f = c • heisO ψ T T' x f

theorem **biComb_single** (x : WO A S × WO S A') (c : ℂ) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' (Finsupp.single x.1 c) (Finsupp.single x.2 1) C = c • twoT ψ A S A' x C

theorem **biComb_eq** (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' c c' C = ∑ x ∈ c.support, ∑ y ∈ c'.support, (c x * c' y) • twoT ψ A S A' (x, y) C

theorem **app_eq** (op : BiOp A S A') (C : WO A S → WO S A' → ℂ) : op.app ψ C = ∑ p ∈ (kOp A S A' op).support, kOp A S A' op p • twoT ψ A S A' p C

theorem **kOp_good** (P : Pl D → Prop) (op : BiOp A S A') (hop : OpAt P op) : ∀ p ∈ (kOp A S A' op).support, p ∈ goodG A S A' P

theorem **ofK_app** (κ : (WO A S × WO S A') →₀ ℂ) (C : WO A S → WO S A' → ℂ) : (ofK A S A' κ).app ψ C = ∑ p ∈ κ.support, κ p • twoT ψ A S A' p C

theorem **ofK_at** (P : Pl D → Prop) (κ : (WO A S × WO S A') →₀ ℂ) (hκ : ∀ p ∈ κ.support, p ∈ goodG A S A' P) : OpAt P (ofK A S A' κ)

theorem **biComb_sum** {ι : Type} (s : Finset ι) (e : WO A S →₀ ℂ) (e' : WO S A' →₀ ℂ) (F : ι → WO A S → WO S A' → ℂ) : biComb ψ A S A' e e' (∑ i ∈ s, F i) = ∑ i ∈ s, biComb ψ A S A' e e' (F i)

theorem **comp_part** (P : Pl D → Prop) (op : BiOp A S A') (hop : OpAt P op) (e : WO A S →₀ ℂ) (e' : WO S A' →₀ ℂ) (he : CombAt P e) (he' : CombAt P e') : ∃ op' : BiOp A S A', OpAt P op' ∧ ∀ C, op'.app ψ C = biComb ψ A S A' e e' (op.app ψ C)

theorem **symplO_zero_left** (T T' : HSp D) (w : WO T T') : symplO T T' 0 w = 0

theorem **psi_eq_one_of_neg** {z : AF D.host.F} (h : ψ (-z) = 1) : ψ z = 1

theorem **GlobalTheory.FinPl.biInv_main** : ∀ D : D7, GlobalTheory.FinPl.BiInv D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BiVacuumOperatorZeroOrLeftInvertibleStatement

theorem **_root_.GlobalTheory.T.biVacuumOperator_zero_or_left_invertible** : ∀ D : D7, GlobalTheory.FinPl.BiInv D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BilinearVacuumCentralCharacter

theorem **contDiff_mean** {K R : Type} [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] [NormedAddCommGroup R] [NormedSpace ℝ R] [FiniteDimensional ℝ R] (μ : Measure K) [IsFiniteMeasure μ] (Q : (K → R → ℂ) → Prop) (hc : ∀ F, Q F → Continuous fun p : K × R => F p.1 p.2) (hd : ∀ F, Q F → ∀ k, Differentiable ℝ (F k)) (hq : ∀ F, Q F → ∀ u, Q fun k x => fderiv ℝ (F k) x u) : ∀ F, Q F → ContDiff ℝ ∞ fun x => ∫ k, F k x ∂μ

theorem **contDiff_mean_span** {K R : Type} [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] [NormedAddCommGroup R] [NormedSpace ℝ R] [FiniteDimensional ℝ R] (μ : Measure K) [IsFiniteMeasure μ] (G : Set (K → R → ℂ)) (hG : ∀ F ∈ G, (Continuous fun q : K × R => F q.1 q.2) ∧ (∀ k, Differentiable ℝ (F k)) ∧ ∀ u, (fun k x => fderiv ℝ (F k) x u) ∈ Submodule.span ℂ G) : ∀ F ∈ G, ContDiff ℝ ∞ fun x => ∫ k, F k x ∂μ

theorem **fderiv_om** (hom : IsWeilDatumO ψ S T χS χT om) (hc : Chart θ σ) {n : ℕ} (e : Module.Basis (Fin n) ℝ (RW T S)) (p : GA D S × GA D T) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S T) (x u : RW T S) : fderiv ℝ (fun y : RW T S => om p f (w + iotaW σ T S y)) x u = (Real.pi * Complex.I) * (omO S T (piW θ T S w + x) u : ℝ) * om p f (w + iotaW σ T S x) + ∑ j, ((e.repr (mov θ p.1⁻¹ p.2⁻¹ u) j : ℝ) : ℂ) * om p (dv θ σ S T (e j) f) (w + iotaW σ T S x)

theorem **continuous_coord** (hθ : IsInfFam θ) {n : ℕ} (e : Module.Basis (Fin n) ℝ (RW T S)) {K : Type} [TopologicalSpace K] {p : K → GA D S × GA D T} (hp : Continuous p) (u : RW T S) (j : Fin n) : Continuous fun k => ((e.repr (mov θ (p k).1⁻¹ (p k).2⁻¹ u) j : ℝ) : ℂ)

theorem **sm_mean** (hom : IsWeilDatumO ψ S T χS χT om) (hc : Chart θ σ) {K : Type} [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] (μ : Measure K) [IsFiniteMeasure μ] {p : K → GA D S × GA D T} (hp : Continuous p) {a : K → ℂ} (ha : Continuous a) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Sm (iotaW σ T S) fun w => ∫ k, a k * om (p k) f w ∂μ

theorem **exists_haar** {S : HSp D} {y : DefSp S} (hy : y.Is) : CompactSpace ↥(GinfC S ⊓ Subgroup.center (GA D S)) ∧ ∃ ν : Measure ↥(GinfC S ⊓ Subgroup.center (GA D S)), IsProbabilityMeasure ν ∧ ν.IsMulLeftInvariant

theorem **family_bounded** {S T : HSp D} (hψ : IsStdChar D ψ (-1)) {K : Type} [TopologicalSpace K] [CompactSpace K] {F : K → WO S T → ℂ} (hF : ∀ k, IsThetaFunO ψ S T (F k)) (hc : Continuous fun q : K × WO S T => F q.1 q.2) : ∃ C : ℝ, ∀ k w, ‖F k w‖ ≤ C

theorem **integral_mean_translate** {S : HSp D} (hS : CoCpt S) (ν : Vol S) {K : Type} [TopologicalSpace K] [SecondCountableTopology K] [MeasurableSpace K] [BorelSpace K] (ν : Measure K) [IsProbabilityMeasure ν] {b : K → GA D S} (hb : Continuous b) {Ψ : GA D S → ℂ} (hΨ : IsContAut S Ψ) : ∫ h in ν.𝓕, (∫ k, Ψ (h ∗ b k) ∂ν) ∂ν.μ = ∫ h in ν.𝓕, Ψ h ∂ν.μ

theorem **theta_comb** {S T : HSp D} (hψ : IsStdChar D ψ (-1)) {c : WO S T →₀ ℂ} (hc : IsFinComb c) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (comb ψ S T c f)

theorem **comb_smul** {S T : HSp D} (c : WO S T →₀ ℂ) (a : ℂ) (f : WO S T → ℂ) : comb ψ S T c (a • f) = a • comb ψ S T c f

theorem **heisO_mean** (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {u : WO S T} (hu : u.map (infPart D) = 0) : heisO ψ S T u (mean om ν χ f) = mean om ν χ (heisO ψ S T u f)

theorem **theta_mean** (hom : IsWeilDatumO ψ S T χS χT om) {θ : InfFam D} (hθ : IsInfFam θ) (hχ : Continuous fun b : ↥(GinfC S ⊓ Subgroup.center (GA D S)) => χ b.1) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (mean om ν χ f)

theorem **vac_mean** (hom : IsWeilDatumO ψ S T χS χT om) {θ : InfFam D} (hθ : IsInfFam θ) (hχ : Continuous fun b : ↥(GinfC S ⊓ Subgroup.center (GA D S)) => χ b.1) {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {f : WO S T → ℂ} (hf : f ∈ vacSub ψ S T (latO S T LS LT fun _ => True)) : mean om ν χ f ∈ vacSub ψ S T (latO S T LS LT fun _ => True)

theorem **comb_mean** (hom : IsWeilDatumO ψ S T χS χT om) (hχ : Continuous fun b : ↥(GinfC S ⊓ Subgroup.center (GA D S)) => χ b.1) {c : WO S T →₀ ℂ} (hc : IsFinComb c) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : comb ψ S T c (mean om ν χ f) = mean om ν χ (comb ψ S T c f)

theorem **om_mean** (hom : IsWeilDatumO ψ S T χS χT om) {θ : InfFam D} (hθ : IsInfFam θ) (hχ : Continuous fun b : ↥(GinfC S ⊓ Subgroup.center (GA D S)) => χ b.1) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (g : GA D S) (h : GA D T) (w : WO S T) : om (g, h) (mean om ν χ f) w = ∫ b, χ b.1 ∗ om (g ∗ b.1, h) f w ∂ν

theorem **eigen_mean** [ν.IsMulLeftInvariant] (hom : IsWeilDatumO ψ S T χS χT om) {θ : InfFam D} (hθ : IsInfFam θ) (hχ : Continuous fun b : ↥(GinfC S ⊓ Subgroup.center (GA D S)) => χ b.1) (hm : ∀ b b' : ↥(GinfC S ⊓ Subgroup.center (GA D S)), χ (b.1 ∗ b'.1) = χ b.1 ∗ χ b'.1) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (b' : ↥(GinfC S ⊓ Subgroup.center (GA D S))) : om (b'.1, 1) (mean om ν χ f) = χ b'.1⁻¹ • mean om ν χ f

theorem **lift_mean** (hom : IsWeilDatumO ψ S T χS χT om) {θ : InfFam D} (hθ : IsInfFam θ) (hS : CoCpt S) (ν : Vol S) (hχ : Continuous fun b : ↥(GinfC S ⊓ Subgroup.center (GA D S)) => χ b.1) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {Φ : GA D S → ℂ} (hΦ : IsContAut S Φ) (he : ∀ (b : ↥(GinfC S ⊓ Subgroup.center (GA D S))) h, Φ (h ∗ b.1) = χ b.1 ∗ Φ h) (h₂ : GA D T) : ∫ h in ν.𝓕, Φ h ∗ om (h, h₂) (mean om ν χ f) 0 ∂ν.μ = ∫ h in ν.𝓕, Φ h ∗ om (h, h₂) f 0 ∂ν.μ

theorem **contAut_lift** (hw : IsWeilDatumO ψ A S χA χS oma) (hA : CoCpt A) (ν : Vol A) {X : WO A S → ℂ} (hX : IsThetaFunO ψ A S X) {φ : GA D A → ℂ} (hφ : IsContAut A φ) : IsContAut S fun h => ∫ g in ν.𝓕, oma (g, h) X 0 ∗ φ g ∂ν.μ

theorem **biComb_sum** (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) {n : ℕ} (X : Fin n → WO A S → ℂ) (Y : Fin n → WO S A' → ℂ) : biComb ψ A S A' c c' (fun u v => ∑ i, X i u ∗ Y i v) = fun u v => ∑ i, comb ψ A S c (X i) u ∗ comb ψ S A' c' (Y i) v

theorem **biLift_sum** (hwa : IsWeilDatumO ψ A S χA χS oma) (hwb : IsWeilDatumO ψ S A' χS' χA' omb) (hA : CoCpt A) (hS : CoCpt S) (vA : Vol A) (vS : Vol S) {n : ℕ} {X : Fin n → WO A S → ℂ} {Y : Fin n → WO S A' → ℂ} (hX : ∀ i, IsThetaFunO ψ A S (X i)) (hY : ∀ i, IsThetaFunO ψ S A' (Y i)) {φ : GA D A → ℂ} (hφ : IsContAut A φ) (h₂ : GA D A') : biLift oma omb vA vS φ (fun u v => ∑ i, X i u ∗ Y i v) h₂ = ∑ i, ∫ h in vS.𝓕, (∫ g in vA.𝓕, oma (g, h) (X i) 0 ∗ φ g ∂vA.μ) ∗ omb (h, h₂) (Y i) 0 ∂vS.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BilinearVacuumNonzeroCuspForm

theorem **dLine_smul** {S T : HSp D} (M : ℝ →+ WO S T) (a : ℂ) (f : WO S T → ℂ) : dLine ψ S T M (a • f) = a • dLine ψ S T M f

theorem **dLine_add** (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {M : ℝ →+ WO S T} (hM : Continuous M) (hM0 : ∀ s, (M s).map (finPart D) = 0) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : dLine ψ S T M (f + g) = dLine ψ S T M f + dLine ψ S T M g

theorem **continuous_act** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) {G : WO A S → WO S A' → ℂ} (hG : IsBiTheta ψ A S A' G) : Continuous fun x : (GA D A × GA D S) × WO A S × WO S A' => oma x.1 (fun u => G u x.2.2) x.2.1

theorem **isBiSmooth_avg** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) {K : Type} [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] (μ : Measure K) [IsFiniteMeasure μ] {p : K → GA D A × GA D S} (hp : Continuous p) {c : K → ℂ} (hc : Continuous c) {G : WO A S → WO S A' → ℂ} (hG : IsBiTheta ψ A S A' G) : IsBiSmooth fun u v => ∫ k, c k ∗ oma (p k) (fun u' => G u' v) u ∂μ

theorem **exists_bound_fam** (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {F : K → WO S T → ℂ} (hF : ∀ k, IsThetaFunO ψ S T (F k)) (hc : Continuous fun q : K × WO S T => F q.1 q.2) : ∃ C : ℝ, ∀ k x, ‖F k x‖ ≤ C

theorem **om_integral** [IsFiniteMeasure μ] {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {c : K → ℂ} (hc : Continuous c) {F : K → WO S T → ℂ} (p : GA D S × GA D T) (hF : ∀ k, IsThetaFunO ψ S T (F k)) (h1 : Continuous fun q : K × WO S T => F q.1 q.2) (h2 : Continuous fun q : K × WO S T => om p (F q.1) q.2) (hI : IsThetaFunO ψ S T fun x => ∫ k, c k ∗ F k x ∂μ) (x : WO S T) : om p (fun y => ∫ k, c k ∗ F k y ∂μ) x = ∫ k, c k ∗ om p (F k) x ∂μ

theorem **integral_mean** [IsProbabilityMeasure μ] {A : HSp D} (hA : CoCpt A) (vA : Vol A) {j : K → GA D A} (hj : Continuous j) {F : GA D A → ℂ} (hF : IsContAut A F) : ∫ h in vA.𝓕, ∫ k, F (h ∗ j k) ∂μ ∂vA.μ = ∫ h in vA.𝓕, F h ∂vA.μ

theorem **avg_fix_left** (ha : IsWeilDatumO ψ A S χA χS oma) (hj : ∀ k, j k ∈ GinfC A) {v : WO S A'} (hG : IsThetaFunO ψ A S fun u => G u v) {w : WO A S} (hw : w.map (infPart D) = 0) (h : heisO ψ A S w (fun u => G u v) = fun u => G u v) : heisO ψ A S w (fun u => avg oma μ j c G u v) = fun u => avg oma μ j c G u v

theorem **avg_fix_right** {w : WO S A'} (h : ∀ u, heisO ψ S A' w (G u) = G u) (u : WO A S) : heisO ψ S A' w (avg oma μ j c G u) = avg oma μ j c G u

theorem **continuous_fam** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) (hG : IsBiTheta ψ A S A' G) {p : K → GA D A × GA D S} (hp : Continuous p) : Continuous fun q : K × WO A S × WO S A' => oma (p q.1) (fun u => G u q.2.2) q.2.1

theorem **isBiTheta_avg** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) (hjc : Continuous j) (hj : ∀ k, j k ∈ GinfC A) (hc : Continuous c) (hG : IsBiTheta ψ A S A' G) : IsBiTheta ψ A S A' (avg oma μ j c G)

theorem **isBiVac_avg** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) (hjc : Continuous j) (hj : ∀ k, j k ∈ GinfC A) (hc : Continuous c) (hG : IsBiTheta ψ A S A' G) {LA LS LA'} (hv : IsBiVac ψ A S A' LA LS LA' G) : IsBiVac ψ A S A' LA LS LA' (avg oma μ j c G)

theorem **biComb_avg** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) (hjc : Continuous j) (hj : ∀ k, j k ∈ GinfC A) (hc : Continuous c) (hG : IsBiTheta ψ A S A' G) {e : WO A S →₀ ℂ} {e' : WO S A' →₀ ℂ} (he : IsFinComb e) (he' : IsFinComb e') : biComb ψ A S A' e e' (avg oma μ j c G) = avg oma μ j c (biComb ψ A S A' e e' G)

theorem **om_avg** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) (hjc : Continuous j) (hj : ∀ k, j k ∈ GinfC A) (hc : Continuous c) (hG : IsBiTheta ψ A S A' G) (g : GA D A) (h : GA D S) (u : WO A S) (v : WO S A') : oma (g, h) (fun u' => avg oma μ j c G u' v) u = ∫ k, c k ∗ oma (g ∗ j k, h) (fun u' => G u' v) u ∂μ

theorem **ker_avg** (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hB : BiAct D) (hjc : Continuous j) (hj : ∀ k, j k ∈ GinfC A) (hc : Continuous c) (hG : IsBiTheta ψ A S A' G) (g : GA D A) (h : GA D S) (q : GA D S × GA D A') (u : WO A S) (v : WO S A') : omb q (fun v' => oma (g, h) (fun u' => avg oma μ j c G u' v') u) v = ∫ k, c k ∗ omb q (fun v' => oma (g ∗ j k, h) (fun u' => G u' v') u) v ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BlockDetLowUEmb

theorem **val_lowU** (ν : Matrix D A k) : (lowU ν).1 = fromBlocks 1 0 ν 1

theorem **val_emb_one** (m : GL D k) : (emb₂ k (Equiv.refl (A ⊕ D)) (1, m)).1 = fromBlocks 1 0 0 m.1

theorem **blockDet_lowU_emb** (ν : Matrix D A k) (m : GL D k) : blockDet (lowU ν ∗ emb₂ k (Equiv.refl (A ⊕ D)) (1, m)) = GeneralLinearGroup.det m

theorem **emb_mul_lowU** (m : GL D k) (ν : Matrix D A k) : emb₂ k (Equiv.refl (A ⊕ D)) (1, m) ∗ lowU ν = lowU (m.1 ∗ ν) ∗ emb₂ k (Equiv.refl (A ⊕ D)) (1, m)

theorem **emb_mul_emb** (m m' : GL D k) : emb₂ k (Equiv.refl (A ⊕ D)) (1, m) ∗ emb₂ k (Equiv.refl (A ⊕ D)) (1, m') = emb₂ k (Equiv.refl (A ⊕ D)) (1, m ∗ m')

theorem **mtor_eq** (ϖ : kˣ) (e : D → ℤ) : (mtor ϖ e : GL (A ⊕ D) k) = emb₂ k (Equiv.refl (A ⊕ D)) (1, torusElt k D ϖ e)

theorem **inv_lowU** (ν : Matrix D A k) : (lowU ν)⁻¹ = lowU (−ν)

theorem **lowU_mem** {ν : Matrix D A k} (h : ∀ i a, valuation k (ν i a) < 1) : lowU ν ∈ parahoric k A D

theorem **emb_mem** {κ : GL D k} (h : κ ∈ KO k D) : emb₂ k (Equiv.refl (A ⊕ D)) (1, κ) ∈ parahoric k A D

theorem **mem_mLat** {v : D → k} : v ∈ mLat k D ↔ ∀ i, valuation k (v i) < 1

theorem **mem_preLat** {m : Matrix D D k} {v : D → k} : v ∈ preLat m ↔ m.mulVec v ∈ mLat k D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BlockMatrixTwoByTwoFinStandardSkewMap

theorem **submatrix_fromBlocks_mul** (a b c d a' b' c' d' : Matrix (Fin n) (Fin n) R) : blk4 a b c d ∗ blk4 a' b' c' d' = blk4 (a ∗ a' + b ∗ c') (a ∗ b' + b ∗ d') (c ∗ a' + d ∗ c') (c ∗ b' + d ∗ d')

theorem **block_matrix_one** : blk4 (1 : Matrix (Fin n) (Fin n) R) 0 0 1 = 1

theorem **blockMatrixTwoByTwoFin_smul** (r : R) (a b c d : Matrix (Fin n) (Fin n) R) : r • blk4 a b c d = blk4 (r • a) (r • b) (r • c) (r • d)

theorem **block_matrix_map** (f : R → S) (a b c d : Matrix (Fin n) (Fin n) R) : (blk4 a b c d).map f = blk4 (a.map f) (b.map f) (c.map f) (d.map f)

theorem **blockMatrixTwoByTwoFin_transpose** (a b c d : Matrix (Fin n) (Fin n) R) : (blk4 a b c d)ᵀ = blk4 aᵀ cᵀ bᵀ dᵀ

theorem **blockMatrixTwoByTwoFin_forall_entries** (p : R → Prop) {a b c d : Matrix (Fin n) (Fin n) R} (ha : ∀ i j, p (a i j)) (hb : ∀ i j, p (b i j)) (hc : ∀ i j, p (c i j)) (hd : ∀ i j, p (d i j)) (i j : Fin (n + n)) : p (blk4 a b c d i j)

theorem **map_smul_hom** (f : R →+* S) {m : Type} (r : R) (a : Matrix m m R) : (r • a).map f = f r • a.map f

theorem **frm_mul_inv** (hKH : K ∗ H = 1) (he : e ∗ e' = 1) (ht : t + t' = 1) : frm K e t t' ∗ frmInv H e' t t' = 1

theorem **adj_frm** (c : R →+* R) (hK : (K.map c)ᵀ = K) (hct : c t = t') (hct' : c t' = t) : ((frm K e t t').map c)ᵀ = blk4 1 1 ((c e ∗ t') • K) ((−(c e ∗ t)) • K)

theorem **gram_frm** (ce : R) (hKH : K ∗ H = 1) (hHK : H ∗ K = 1) (ht : t + t' = 1) : blk4 1 1 ((ce ∗ t') • K) ((−(ce ∗ t)) • K) ∗ blk4 H 0 0 (−H) ∗ frm K e t t' = blk4 0 (e • 1) (ce • 1) 0

theorem **norm_frm** (ce : R) (f : R →+* S) (hK : (K ∗ K).map f = 1) (ht : t + t' = 1) (h2 : f (2 ∗ t) = 1) (h4 : f (e ∗ ce) = 4) : (blk4 1 1 ((ce ∗ t') • K) ((−(ce ∗ t)) • K) ∗ frm K e t t').map f = (2 : S) • 1

theorem **exists_frame_ring** (c : R →+* R) (hc : ∀ x, c (c x) = x) (p : R → Prop) (p0 : p 0) (p1 : p 1) (pmul : ∀ {x y}, p x → p y → p (x ∗ y)) (pneg : ∀ {x}, p x → p (−x)) (pc : ∀ {x}, p x → p (c x)) (Hm : GL (Fin n) R) (hH' : (Hm.1.map c)ᵀ = Hm.1) (hI1 : ∀ i j, p (Hm.1 i j)) (hI2 : ∀ i j, p ((Hm⁻¹).1 i j)) (m : R) (hm : c m = −m) (u ε : Rˣ) (t : R) (hs1 : (ε : R) ∗ u ∗ c u = m) (hs2 : p ε) (hs3 : p (ε⁻¹ : Rˣ)) (ht1 : t + c t = 1) (ht2 : p t) : ∃ M M' : GL (Fin (n + n)) R, (M.1.map c)ᵀ ∗ blk4 Hm.1 0 0 (−Hm.1) ∗ M.1 = m • blk4 0 1 (−1) 0 ∧ (∀ i j : Fin n, M.1 (finSumFinEquiv (Sum.inr i : Fin n ⊕ Fin n)) (finSumFinEquiv (Sum.inl j : Fin n ⊕ Fin n)) = M.1 (finSumFinEquiv (Sum.inl i : Fin n ⊕ Fin n)) (finSumFinEquiv (Sum.inl j : Fin n ⊕ Fin n))) ∧ ((∀ i j, p (M'.1 i j)) ∧ ∀ i j, p ((M'⁻¹).1 i j)) ∧ M.1 = (u : R) • M'.1 ∧ ∀ (S : Type) [CommRing S] (f : R →+* S), (Hm.1 ∗ Hm.1).map f = 1 → f (ε ∗

theorem **blockMatrixTwoByTwoFin_standardSkew_map** (f : R →+* S) : (blk4 (0 : Matrix (Fin n) (Fin n) R) 1 (−1) 0).map f = blk4 0 1 (−1) 0

theorem **compFin_cA** (v : HeightOneSpectrum (𝒪 D.host.F)) (z : AE D) : compFin D.host.F D.host.E v (cA D z) = sigR _ D.sp.cm.sig (compFin D.host.F D.host.E v z)

theorem **intA_cA** {z : AE D} (h : IntA z) : IntA (cA D z)

theorem **intA_mul** {x y : AE D} (hx : IntA x) (hy : IntA y) : IntA (x ∗ y)

theorem **intA_neg** {x : AE D} (hx : IntA x) : IntA (−x)

theorem **intA_zero** : IntA (0 : AE D)

theorem **intA_one** : IntA (1 : AE D)

theorem **cA_incA_skew** {μ : D.host.E} (h : IsSkew μ) : cA D (incA D μ) = −incA D μ

theorem **rat_Jst** (n : ℕ) : (rat (Jst D n) : Mx D (n + n) (n + n)) = blk4 0 1 (−1) 0

theorem **exists_frame** : ∀ (D : D7) (n : ℕ) (Hm : GL (Fin n) (AE D)), adj Hm.1 = Hm.1 → IntGL Hm → (∀ w, (Hm.1 ∗ Hm.1).map (compInf D.host.F D.host.E w) = 1) → ∀ (μ : D.host.E) (u ε : (AE D)ˣ) (t : AE D), IsSkew μ → ScalData μ u ε → HalfData t → ∃ M M' : GL (Fin (n + n)) (AE D), adj M.1 ∗ dblGram finSumFinEquiv Hm.1 ∗ M.1 = incA D μ • rat (Jst D n) ∧ (∀ i j : Fin n, M.1 (finSumFinEquiv (Sum.inr i : Fin n ⊕ Fin n)) (finSumFinEquiv (Sum.inl j : Fin n ⊕ Fin n)) = M.1 (finSumFinEquiv (Sum.inl i : Fin n ⊕ Fin n)) (finSumFinEquiv (Sum.inl j : Fin n ⊕ Fin n))) ∧ IntGL M' ∧ M.1 = (u : AE D) • M'.1 ∧ ∀ w, (adj M.1 ∗ M.1).map (compInf D.host.F D.host.E w) = compInf D.host.F D.host.E w (2 ∗ u ∗ cA D u) • 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BottomHypothesisMatchingCentral

theorem **botHyp_match** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → W.Q.Rl.toRealDat.IsModels ∧ IsInfFam W.Q.Rl.θ ∧ IsStdChar D W.Q.Rl.ψ (-1) ∧ (∀ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) → Matches W.S.X W.Q.instEq (rJ W) (rChi W) (z ∗ (show GA D (D.sp.Y 0) from W.Q.tS))) ∧ (∀ k ∈ rJ W, rChi W k ≠ 0) ∧ ∀ e : W.Q.Rl.HE.H, W.Q.Rl.HE.e e ∈ W.Q.Rl.toRealDat.EEat (W.Q.Rl.toRealDat.levR W.Q.F.lv2 true) → rowOf W e ∈ ThetaLiftSpaceZero W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BottomHypothesisMatchingCentralStatement

theorem **_root_.GlobalTheory.T.bottom_hypothesis_matching_central** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → W.Q.Rl.toRealDat.IsModels ∧ IsInfFam W.Q.Rl.θ ∧ IsStdChar D W.Q.Rl.ψ (-1) ∧ (∀ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) → Matches W.S.X W.Q.instEq (rJ W) (rChi W) (z ∗ (show GA D (D.sp.Y 0) from W.Q.tS))) ∧ (∀ k ∈ rJ W, rChi W k ≠ 0) ∧ ∀ e : W.Q.Rl.HE.H, W.Q.Rl.HE.e e ∈ W.Q.Rl.toRealDat.EEat (W.Q.Rl.toRealDat.levR W.Q.F.lv2 true) → rowOf W e ∈ ThetaLiftSpaceZero W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BottomRelationOfPin

theorem **per_smul_f** {ρ : ℕ} (B : BesLev D ρ) (ψ : AddChar (AF D.host.F) ℂ) (φ : GA D B.A → ℂ) (b : ℂ) (f : GA D B.V → ℂ) : B.per ψ φ (b • f) = b ∗ B.per ψ φ f

theorem **contAut_zero** (V : HSp D) : IsContAut V (0 : GA D V → ℂ)

theorem **contAut_add** {V : HSp D} {f g : GA D V → ℂ} (hf : IsContAut V f) (hg : IsContAut V g) : IsContAut V (f + g)

theorem **contAut_smul** {V : HSp D} {f : GA D V → ℂ} (hf : IsContAut V f) (c : ℂ) : IsContAut V (c • f)

theorem **contAut_rTrans** {V : HSp D} {f : GA D V → ℂ} (h : GA D V) (hf : IsContAut V f) : IsContAut V (rTrans h f)

theorem **contAut_multT** {V : HSp D} {S : D.what₂.Tau → SPar} {θ : InfFam D} {M : DSModel V S} {J : Set (GA D V)} {χ : GA D V → ℂ} {G : SmT M →ₗ[ℂ] (GA D V → ℂ)} (hG : G ∈ MultT θ M J χ) (x : SmT M) : IsContAut V (G x)

theorem **lev_multT** {V : HSp D} {S : D.what₂.Tau → SPar} {θ : InfFam D} {M : DSModel V S} {J : Set (GA D V)} {χ : GA D V → ℂ} {G : SmT M →ₗ[ℂ] (GA D V → ℂ)} (hG : G ∈ MultT θ M J χ) (x : SmT M) {k : GA D V} (hk : k ∈ J) (g : GA D V) : G x (g ∗ k) = χ k ∗ G x g

theorem **shift_swap** {G A B : Type} [Group G] [Group A] [Group B] (ι : G ≃∗ A × B) (g w : G) (y : A) (x : B) (hw : (ι w).1 = 1) : ι.symm (ι g ∗ (y, x)) ∗ w = g ∗ (ι.symm (1, x) ∗ w) ∗ ι.symm (y, 1)

theorem **comb_integral** {Y : Type} [MeasurableSpace Y] (ν : Measure Y) (c α α' : ℂ) (F F' : Y → ℂ) (h : ∀ y, α ∗ F y + α' ∗ F' y = 0) : α ∗ (c ∗ ∫ y, F y ∂ν) + α' ∗ (c ∗ ∫ y, F' y ∂ν) = 0

theorem **GlobalTheory.MiTop.botRel_two** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → ∀ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) → ∀ u (d : BotDat W u), d.Pin z → BotRel W z d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BottomRelationOfPinStatement

theorem **_root_.GlobalTheory.T.bottom_relation_of_pin** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → ∀ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) → ∀ u (d : BotDat W u), d.Pin z → BotRel W z d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BottomSphericalCardWeightIndex

theorem **relIndex_KO_ne_zero** {n : Type} [Fintype n] [DecidableEq n] {J : Subgroup (GL n k)} (hJ : J ≤ KO k n) (ho : IsOpen (J : Set (GL n k))) : J.relIndex (KO k n) ≠ 0

theorem **N_one_apply_inl** {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c 1) (a b : n) : u.val (Sum.inl a) (Sum.inl b) = (1 : Matrix n n k) a b

theorem **mul_embX_inl** {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c 1) (g : GL n k) (a b : n) : (u ∗ embX c g).val (Sum.inl a) (Sum.inl b) = g.val a b

theorem **embX_mul_inl** {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c 1) (g : GL n k) (a b : n) : (embX c g ∗ u).val (Sum.inl a) (Sum.inl b) = g.val a b

theorem **intM_embX** {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {g : GL n k} (h : IntM g.val) : IntM (embX c g : GL (n ⊕ Fin c) k).val

theorem **prX_Cgrp_KO_one** (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) : prX (Cgrp (KO k (n ⊕ Fin c)) 1) = KO k n

theorem **reGL_symm_mem_KO_iff** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (e : a ≃ b) {x : GL b k} : (reGL e).symm x ∈ KO k a ↔ x ∈ KO k b

theorem **twistChain_lev_zero** {s : ℕ} {t κ : Type} [Fintype t] [DecidableEq t] [Fintype κ] [DecidableEq κ] (δ : kˣ →∗ ℂˣ) (hδ : IsTame δ) (η : ℕ → kˣ →∗ ℂˣ) (c : ℕ) (h : 0 + c = s + 1) : ((twistChain s t κ δ hδ η).lev 0 c h).J = KO k _

theorem **QExt.bottom_one** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →∗ Circle) (E : QExt G χ), Fintype.card (E.Ch 0) = 1 ∧ (∀ α, E.wt 0 α = 1) ∧ E.idx 0 = 1 ∧ ∀ j, j ≤ G.s + 1 + 1 → E.idx j ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BottomSphericalCardWeightIndexStatement

theorem **_root_.GlobalTheory.T.bottom_spherical_card_weight_index** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle) (E : QExt G χ), Fintype.card (E.Ch 0) = 1 ∧ (∀ α, E.wt 0 α = 1) ∧ E.idx 0 = 1 ∧ ∀ j, j ≤ G.s + 1 + 1 → E.idx j ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoundedAutomorphicParametersStarStable

theorem **isParamVec_of_map** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} {n : ℕ} {J : Matrix (Fin n) (Fin n) E} {L : Submodule (𝒪 E) (Fin n → E)} {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} {h : w.under (𝒪 F) = v} {X Y : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup Y] [Module ℂ Y] {ρ : Representation ℂ (GLoc σ J v) X} {ρ' : Representation ℂ (GLoc σ J v) Y} (f : X →ₗ[ℂ] Y) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (hi : Function.Injective f) {x : X} {B : Multiset ℂˣ} (hx : IsParamVec σ J L h ρ' (f x) B) : IsParamVec σ J L h ρ x B

theorem **param_star** (D : D7) (V : HSp D) (S : Finset (HeightOneSpectrum (𝒪 D.host.F))) : ∀ f ∈ GlobalTheory.FinZeta.bddAut V, ∀ B : GlobalTheory.LFun.Fam D.host.E, GlobalTheory.LFun.IsParamFunV V S f B → ∀ w ∉ GlobalTheory.LFun.above D.host.E (S : Set (HeightOneSpectrum (𝒪 D.host.F))), (∀ w' : HeightOneSpectrum (𝒪 D.host.E), w'.under (𝒪 D.host.F) = w.under (𝒪 D.host.F) → w ≠ w' → B w' = (B w).map star) ∧ (¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoundedAutomorphicParametersStarStableStatement

theorem **_root_.GlobalTheory.T.boundedAutomorphic_parameters_star_stable** : ∀ (D : D7) (V : HSp D) (S : Finset (HeightOneSpectrum (𝒪 D.host.F))), ∀ f ∈ GlobalTheory.FinZeta.bddAut V, ∀ B : GlobalTheory.LFun.Fam D.host.E, GlobalTheory.LFun.IsParamFunV V S f B → ∀ w ∉ GlobalTheory.LFun.above D.host.E (S : Set (HeightOneSpectrum (𝒪 D.host.F))), (∀ w' : HeightOneSpectrum (𝒪 D.host.E), w'.under (𝒪 D.host.F) = w.under (𝒪 D.host.F) → w ≠ w' → B w' = (B w).map star) ∧ (¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoundedFunctionalThetaEstimate

theorem **line_move** (hθ : IsInfFam θ) {k l m n : ℕ} (P : Mx D k m) (Q : Mx D n l) (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : lineAt θ τ (compAt θ τ P · z · compAt θ τ Q) t = (P · lineAt θ τ z t · Q : Mx D k l)

theorem **phase_real** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {m n : ℕ} (y : Mx D n m) (κ : Mx D m n) (hκ : κ.map (finPart D) = 0) : ψ (-(Algebra.trace (AF D.host.F) (AE D) (y · κ).trace)) = Complex.exp (2 * Real.pi * Complex.I * ((∑ τ, 2 * (Matrix.trace (compAt θ τ y · compAt θ τ κ)).re : ℝ) : ℂ))

theorem **phase_line** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {m n : ℕ} (y : Mx D n m) (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : ψ (-(Algebra.trace (AF D.host.F) (AE D) (y · lineAt θ τ z t).trace)) = Complex.exp (((4 * Real.pi * (Matrix.trace (compAt θ τ y · z)).re * t : ℝ) : ℂ) * Complex.I)

theorem **deriv_char** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (ℓ : (WO S T → ℂ) → ℂ) (hadd : ∀ f g, IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → ℓ (f + g) = ℓ f + ℓ g) (hsmul : ∀ (c : ℂ) f, IsThetaFunO ψ S T f → ℓ (c • f) = c * ℓ f) (hb : ∀ f, IsThetaFunO ψ S T f → ‖ℓ f‖ₑ ≤ ⊔ w, ‖f w‖ₑ) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) (a : ℝ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (h : ∀ t : ℝ, ℓ (heisO ψ S T (L t) f) = Complex.exp (((a * t : ℝ) : ℂ) * Complex.I) * ℓ f) : ℓ (dLine ψ S T L f) = (a : ℂ) * Complex.I * ℓ f

theorem **entry_le** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {m n : ℕ} (P : Mx D T.rank m) (Q : Mx D n S.rank) (ℓ : (WO S T → ℂ) → ℂ) (y : Mx D n m) (hadd : ∀ f g, IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → ℓ (f + g) = ℓ f + ℓ g) (hsmul : ∀ (c : ℂ) f, IsThetaFunO ψ S T f → ℓ (c • f) = c * ℓ f) (hb : ∀ f, IsThetaFunO ψ S T f → ‖ℓ f‖ₑ ≤ ⊔ w, ‖f w‖ₑ) (hchar : ∀ (ξ : Mx D m n) f, IsThetaFunO ψ S T f → ℓ (heisO ψ S T (P · ξ · Q) f) = ψ (-(Algebra.trace (AF D.host.F) (AE D) (y · ξ).trace)) * ℓ f) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (τ : D.what₂.Tau) (i : Fin n) (j : Fin m) : ‖ℓ f‖ₑ * ENNReal.ofReal ‖compAt θ τ y i j‖ ≤ ‖ℓ (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ (compAt θ τ P · single j i (1 : ℂ) · compAt θ τ Q) f)‖ₑ + ‖ℓ (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ (compAt θ τ P · single j i Complex.I · compAt θ τ Q) f)‖ₑ

theorem **GlobalTheory.CharDecay.decay** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (m n : ℕ) (P : Mx D T.rank m) (Q : Mx D n S.rank) (N : ℕ), ∃ C : ℝ≥0, ∀ (ℓ : (WO S T → ℂ) → ℂ) (y : Mx D n m), (∀ f g, IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → ℓ (f + g) = ℓ f + ℓ g) → (∀ (c : ℂ) f, IsThetaFunO ψ S T f → ℓ (c • f) = c * ℓ f) → (∀ f, IsThetaFunO ψ S T f → ‖ℓ f‖ₑ ≤ ⊔ w, ‖f w‖ₑ) → (∀ (ξ : Mx D m n) f, IsThetaFunO ψ S T f → ℓ (heisO ψ S T (P · ξ · Q) f) = ψ (-(Algebra.trace (AF D.host.F) (AE D) (y · ξ).trace)) * ℓ f) → ∀ f, IsThetaFunO ψ S T f → ‖ℓ f‖ₑ * ENNReal.ofReal ((1 + realSize θ y) ^ N) ≤ C * supSob ψ θ N f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoundedFunctionalThetaEstimateStatement

theorem **_root_.GlobalTheory.T.bounded_functional_theta_estimate** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (m n : ℕ) (P : Mx D T.rank m) (Q : Mx D n S.rank) (N : ℕ), ∃ C : ℝ≥0, ∀ (ℓ : (WO S T → ℂ) → ℂ) (y : Mx D n m), (∀ f g, IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → ℓ (f + g) = ℓ f + ℓ g) → (∀ (c : ℂ) f, IsThetaFunO ψ S T f → ℓ (c • f) = c * ℓ f) → (∀ f, IsThetaFunO ψ S T f → ‖ℓ f‖ₑ ≤ ⊔ w, ‖f w‖ₑ) → (∀ (ξ : Mx D m n) f, IsThetaFunO ψ S T f → ℓ (heisO ψ S T (P · ξ · Q) f) = ψ (-(Algebra.trace (AF D.host.F) (AE D) (y · ξ).trace)) * ℓ f) → ∀ f, IsThetaFunO ψ S T f → ‖ℓ f‖ₑ * ENNReal.ofReal ((1 + realSize θ y) ^ N) ≤ C * supSob ψ θ N f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoundedSubgroupOfIndefiniteUnitaryEqMaximalCompa

theorem **norm_eigenvalue_le_one** {n : ℕ} (M : Matrix (Fin n) (Fin n) ℂ) (C : ℝ) (hC : ∀ k : ℕ, ∀ i j, ‖(M ^ k) i j‖ ≤ C) (v : Fin n → ℂ) (hv : v ≠ 0) (μ : ℂ) (h : M *ᵥ v = μ • v) : ‖μ‖ ≤ 1

theorem **eq_one_of_posSemidef** {n : ℕ} (p q : Matrix (Fin n) (Fin n) ℂ) (hp : p.PosSemidef) (hqp : q * p = 1) (C : ℝ) (hC : ∀ k : ℕ, ∀ i j, ‖(p ^ k) i j‖ ≤ C) (hC' : ∀ k : ℕ, ∀ i j, ‖(q ^ k) i j‖ ≤ C) : p = 1

theorem **eq_Kset_of_bounded** : ∀ (n : ℕ) (ε : Fin n → Bool) (G' : Set (Matrix (Fin n) (Fin n) ℂ)), G' ⊆ Formalism.MainNotions.S.UJset ε → Formalism.REAL.Kset ε ⊆ G' → (∀ a ∈ G', ∀ b ∈ G', a * b ∈ G') → (∀ a ∈ G', a⁻¹ ∈ G') → (∃ C : ℝ, ∀ a ∈ G', ∀ i j, ‖a i j‖ ≤ C) → G' = Formalism.REAL.Kset ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoundedSubgroupOfIndefiniteUnitaryEqMaximalCompaStatement

theorem **_root_.GlobalTheory.T.bounded_subgroup_of_indefinite_unitary_eq_maximalCompact** : ∀ (n : ℕ) (ε : Fin n → Bool) (G' : Set (Matrix (Fin n) (Fin n) ℂ)), G' ⊆ Formalism.MainNotions.S.UJset ε → Formalism.REAL.Kset ε ⊆ G' → (∀ a ∈ G', ∀ b ∈ G', a * b ∈ G') → (∀ a ∈ G', a⁻¹ ∈ G') → (∃ C : ℝ, ∀ a ∈ G', ∀ i j, ‖a i j‖ ≤ C) → G' = Formalism.REAL.Kset ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoxProductsOrthogonalSumTargetCartanFock

theorem **frameLat_selfDual** {D : D7} (V : HSp D) : IsSelfDualC V (frameLat V)

theorem **box_sum_target** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B W H : HSp D) (o : OrthDec B W H) (sA : RealMat A) (sW : RealMat W) (sH : RealMat H), IsStdChar D ψ (-1) → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D W τ) (sW τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → ∃ (sB : RealMat B) (LB : Lat B), (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) ∧ IsSelfDualC B LB ∧ (∀ Φ Φ', IsFockVec ψ θ A W sA sW Φ → IsFockVec ψ θ A H sA sH Φ' → IsFockVec ψ θ A B sA sB (o.boxT A Φ Φ')) ∧ ∀ (LA : Lat A) (Sf : Finset (FinPl D)) Φ Φ', IsSphOff ψ A W LA (frameLat W) Sf Φ → IsSphOff ψ A H LA (frameLat H) Sf Φ' → IsSphOff ψ A B LA LB Sf (o.boxT A Φ Φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BoxProductsOrthogonalSumTargetCartanFockStatement

theorem **_root_.GlobalTheory.T.boxProducts_orthogonalSum_target_cartan_fock** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B W H : HSp D) (o : OrthDec B W H) (sA : RealMat A) (sW : RealMat W) (sH : RealMat H), IsStdChar D ψ (-1) → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D W τ) (sW τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → ∃ (sB : RealMat B) (LB : Lat B), (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) ∧ IsSelfDualC B LB ∧ (∀ Φ Φ', IsFockVec ψ θ A W sA sW Φ → IsFockVec ψ θ A H sA sH Φ' → IsFockVec ψ θ A B sA sB (o.boxT A Φ Φ')) ∧ ∀ (LA : Lat A) (Sf : Finset (FinPl D)) Φ Φ', IsSphOff ψ A W LA (frameLat W) Sf Φ → IsSphOff ψ A H LA (frameLat H) Sf Φ' → IsSphOff ψ A B LA LB Sf (o.boxT A Φ Φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BranchingMultiplicityOneUnitaryFamilies

theorem **fnOf_ten** {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (s : ι → ℤ) (f : ι → MvPolynomial (Fin n × Fin n) ℂ) (y : Frames G) : fnOf G s 1 (ten f) y = ∏ σ, ((y.1 σ).det ^ s σ * eval (fun p : Fin n × Fin n => y.1 σ p.1 p.2) (f σ))

theorem **eval_rowAct** {n : ℕ} (g y : Matrix (Fin n) (Fin n) ℂ) (f : MvPolynomial (Fin n × Fin n) ℂ) : eval (fun p : Fin n × Fin n => y p.1 p.2) (Formalism.GradedWeylDivisibility.rowAct ℂ g f) = eval (fun p : Fin n × Fin n => (g * y) p.1 p.2) f

theorem **fnOf_ten_update** {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (s : ι → ℤ) (f : ι → MvPolynomial (Fin n × Fin n) ℂ) (τ : ι) (g : Matrix (Fin n) (Fin n) ℂ) (z z' : Frames G) (hz : IsTr (Function.update (1 : ι → Matrix (Fin n) (Fin n) ℂ) τ g) z z') : fnOf G s 1 (ten (Function.update f τ (g.det ^ s τ • Formalism.GradedWeylDivisibility.rowAct ℂ g (f τ)))) z = fnOf G s 1 (ten f) z'

theorem **eval_blockDet** {n : ℕ} (τ : ι) (x : ι × Fin n × Fin n → ℂ) : eval x (blockDet τ) = (Matrix.of fun a b : Fin n => x (τ, a, b)).det

theorem **fnOf_inj** {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (s : ι → ℤ) (P : MvPolynomial (ι × Fin n × Fin n) ℂ) (h : ∀ y, fnOf G s 1 P y = 0) : P = 0

theorem **rowAct_mem** {N : ℕ} (lam : DomWeight N) (g : Matrix (Fin N) (Fin N) ℂ) (hg : IsUnit g.det) (f : MvPolynomial (Fin N × Fin N) ℂ) (hf : f ∈ delSq ℂ N lam) : Formalism.GradedWeylDivisibility.rowAct ℂ g f ∈ delSq ℂ N lam

theorem **actL_apply** {N : ℕ} (lam : DomWeight N) (c : ℤ) (g : Matrix (Fin N) (Fin N) ℂ) (hg : IsUnit g.det) (f : delSq ℂ N lam) : (actL lam c g f).1 = g.det ^ c • Formalism.GradedWeylDivisibility.rowAct ℂ g f.1

theorem **span_map_mult_one** {V W : ι → Type} [∀ τ, AddCommGroup (V τ)] [∀ τ, Module ℂ (V τ)] [∀ τ, AddCommGroup (W τ)] [∀ τ, Module ℂ (W τ)] (C : ∀ τ, Set (V τ →ₗ[ℂ] W τ)) (hC : ∀ τ, ∀ t ∈ C τ, ∀ t₀ ∈ C τ, t₀ ≠ 0 → ∃ c : ℂ, t = c • t₀) (Q Q₀ : PiTensorProduct ℂ V →ₗ[ℂ] PiTensorProduct ℂ W) (hQ : Q ∈ Submodule.span ℂ {T | ∃ t : ∀ τ, V τ →ₗ[ℂ] W τ, (∀ τ, t τ ∈ C τ) ∧ T = PiTensorProduct.map t}) (hQ₀ : Q₀ ∈ Submodule.span ℂ {T | ∃ t : ∀ τ, V τ →ₗ[ℂ] W τ, (∀ τ, t τ ∈ C τ) ∧ T = PiTensorProduct.map t}) (h0 : Q₀ ≠ 0) : ∃ c : ℂ, Q = c • Q₀

theorem **isBlock_one** {N n : ℕ} (hN : N = n + 1) : GlobalTheory.BranchFormula.IsBlock hN (1 : Matrix (Fin N) (Fin N) ℂ) (1 : Matrix (Fin n) (Fin n) ℂ) 1

theorem **emb_update** {N : ℕ} (G : ι → Matrix (Fin N) (Fin N) ℂ) (lam : ι → DomWeight N) (e : PiTensorProduct ℂ (fun _ : ι => MvPolynomial (Fin N × Fin N) ℂ) ≃ₗ[ℂ] MvPolynomial (ι × Fin N × Fin N) ℂ) (he : ∀ f : ι → MvPolynomial (Fin N × Fin N) ℂ, e (PiTensorProduct.tprod ℂ f) = ∏ τ, rename (Prod.mk τ) (f τ)) (τ : ι) (g : Matrix (Fin N) (Fin N) ℂ) (hg : IsUnit g.det) (x : PiTensorProduct ℂ (fun τ => delSq ℂ N (lam τ))) (y y' : Frames G) (hy : IsTr (Function.update (1 : ι → Matrix (Fin N) (Fin N) ℂ) τ g) y y') : emb G lam e (PiTensorProduct.map (Function.update (fun σ => (LinearMap.id : delSq ℂ N (lam σ) →ₗ[ℂ] delSq ℂ N (lam σ))) τ (actL (lam τ) (shiftOf lam τ) g)) x) y = emb G lam e x y'

theorem **range_emb** {N : ℕ} (G : ι → Matrix (Fin N) (Fin N) ℂ) (lam : ι → DomWeight N) (e : PiTensorProduct ℂ (fun _ : ι => MvPolynomial (Fin N × Fin N) ℂ) ≃ₗ[ℂ] MvPolynomial (ι × Fin N × Fin N) ℂ) (hU : LinearMap.range (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (delSq ℂ N (lam τ)).subtype) = Submodule.span ℂ {p | ∃ f : ι → MvPolynomial (Fin N × Fin N) ℂ, (∀ τ, f τ ∈ delSq ℂ N (lam τ)) ∧ p = ∏ τ, rename (Prod.mk τ) (f τ)}) : LinearMap.range (emb G lam e) = ValC G lam

theorem **emb_inj** {N : ℕ} (G : ι → Matrix (Fin N) (Fin N) ℂ) (lam : ι → DomWeight N) (e : PiTensorProduct ℂ (fun _ : ι => MvPolynomial (Fin N × Fin N) ℂ) ≃ₗ[ℂ] MvPolynomial (ι × Fin N × Fin N) ℂ) (hI : Function.Injective (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (delSq ℂ N (lam τ)).subtype)) : Function.Injective (emb G lam e)

theorem **isUnit_of_unitary** {n : ℕ} {k : Matrix (Fin n) (Fin n) ℂ} (hk : k ∈ Matrix.unitaryGroup (Fin n) ℂ) : IsUnit k.det

theorem **mult_one_families** : ∀ (ι : Type) [Fintype ι] (N n : ℕ) (hN : N = n + 1) (G : ι → Matrix (Fin N) (Fin N) ℂ) (G' : ι → Matrix (Fin n) (Fin n) ℂ) (lam : ι → DomWeight N) (sg : ι → DomWeight n), (∀ τ, Formalism.L2.Col.Br.Interlace hN (lam τ) (sg τ)) → ∀ T₀, BrEqv hN G G' lam sg T₀ → (∃ Φ ∈ ValC G lam, T₀ Φ ≠ 0) → ∀ T, BrEqv hN G G' lam sg T → ∃ c : ℂ, ∀ Φ ∈ ValC G lam, T Φ = c • T₀ Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BranchingMultiplicityOneUnitaryFamiliesStatement

theorem **_root_.GlobalTheory.T.branching_multiplicityOne_unitary_families** : ∀ (ι : Type) [Fintype ι] (N n : ℕ) (hN : N = n + 1) (G : ι → Matrix (Fin N) (Fin N) ℂ) (G' : ι → Matrix (Fin n) (Fin n) ℂ) (lam : ι → DomWeight N) (sg : ι → DomWeight n), (∀ τ, Formalism.L2.Col.Br.Interlace hN (lam τ) (sg τ)) → ∀ T₀, BrEqv hN G G' lam sg T₀ → (∃ Φ ∈ ValC G lam, T₀ Φ ≠ 0) → ∀ T, BrEqv hN G G' lam sg T → ∃ c : ℂ, ∀ Φ ∈ ValC G lam, T Φ = c • T₀ Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.BrkSumRepsOrth

theorem **brk_sum_reps_orth** : ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (H : Formalism.L2.Th1.H.Sesq V) (α : V → ℂ) (A B Λ : AddSubgroup V), SplitHyp H α A B Λ → ∀ SB : Finset V, Formalism.L2.Th1.H.IsReps Λ B SB → ∀ w ∈ Λ ⊓ A, ∑ y ∈ SB, brk H y w = if w ∈ B then (SB.card : ℂ) else 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CaRepApply

theorem **rep_mem_CA** {A : HSp D} (α : Frame p A n) (g : GL n k) {φ : GA D A → ℂ} (h : φ ∈ CA A) : α.rep g φ ∈ CA A

@[simp] theorem **caRep_apply** {A : HSp D} (α : Frame p A n) (g : GL n k) (φ : CA A) : ((caRep α g φ : CA A) : GA D A → ℂ) = α.rep g φ

@[simp] theorem **blk_val** (g : GL n k) : (blk g).val = fromBlocks g.val (0 : Matrix n (Fin 1) k) 0 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CapelliLowestGelfandConj

theorem **single_comm** {N : ℕ} (a b c d : Fin N) : Matrix.single a b (1 : ℂ) * Matrix.single c d 1 - Matrix.single c d 1 * Matrix.single a b 1 = (if b = c then Matrix.single a d 1 else 0) - (if d = a then Matrix.single c b 1 else 0)

theorem **lie_of_glRel** {A : Type} [Ring A] [Algebra ℂ A] {N : ℕ} (Q : Matrix (Fin N) (Fin N) ℂ →ₗ[ℂ] A) (h : Formalism.REAL.GlRel (fun i j => Q (Matrix.single i j 1))) (X Y : Matrix (Fin N) (Fin N) ℂ) : Q X * Q Y - Q Y * Q X = Q (X * Y - Y * X)

theorem **glRel_conj** {A : Type} [Ring A] [Algebra ℂ A] (N : ℕ) (Q : Matrix (Fin N) (Fin N) ℂ →ₗ[ℂ] A) (g : GL (Fin N) ℂ) (h : Formalism.REAL.GlRel (fun i j => Q (Matrix.single i j 1))) : Formalism.REAL.GlRel (fun i j => Q (g.1 * Matrix.single i j 1 * (g⁻¹).1))

theorem **gelfA_conj** {A : Type} [Ring A] [Algebra ℂ A] (N : ℕ) (Q : Matrix (Fin N) (Fin N) ℂ →ₗ[ℂ] A) (g : GL (Fin N) ℂ) : GlobalTheory.SmoothForms.gelfA (fun i j => Q (g.1 * Matrix.single i j 1 * (g⁻¹).1)) = GlobalTheory.SmoothForms.gelfA (fun i j => Q (Matrix.single i j 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardFilterLowerHeckeRepsEqRelIndexPow

theorem **ko_mulVec_mem** (κ : GL D k) (hκ : κ ∈ KO k D) (v : D → k) (hv : v ∈ mLat k D) : κ.1 *ᵥ v ∈ mLat k D

theorem **dc_mulVec_mem** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) {s : Finset D} {g : GL D k} (hg : ∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') (v : D → k) (hv : v ∈ mLat k D) : g.1 *ᵥ v ∈ mLat k D

theorem **preLat_inv_le** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) {s : Finset D} {g : GL D k} (hg : ∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') : preLat (g⁻¹).1 ≤ mLat k D

theorem **smul_mem_preLat_inv** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) {s : Finset D} {g : GL D k} (hg : ∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') (u : D → k) (hu : u ∈ mLat k D) : (ϖ : k) • u ∈ preLat (g⁻¹).1

theorem **relIndex_preLat_inv_ne_zero** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) {s : Finset D} {g : GL D k} (hg : ∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') : (preLat (g⁻¹).1).relIndex (mLat k D) ≠ 0

theorem **card_eq_relIndex_of_reps** {G : Type} [AddCommGroup G] (H K : AddSubgroup G) (F : Finset G) (hF : ∀ x ∈ F, x ∈ K) (hu : ∀ x ∈ K, ∃! y, y ∈ F ∧ x - y ∈ H) : F.card = H.relIndex K

theorem **relIndex_pi_univ** {ι G : Type} [Fintype ι] [AddCommGroup G] (A B : ι → AddSubgroup G) : (AddSubgroup.pi Set.univ A).relIndex (AddSubgroup.pi Set.univ B) = ∏ i, (A i).relIndex (B i)

theorem **mem_sup_iff_of_mem** {V : Type} [AddCommGroup V] {S L large₂ : AddSubgroup V} (h : large₂ ≤ L) {x : V} (hx : x ∈ L) : x ∈ S ⊔ large₂ ↔ x ∈ (S ⊓ L) ⊔ large₂

theorem **sub_mem_iff** (g : GL D k) (ν ν' : Matrix D A k) : (∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν - ν' = g.1 * μ) ↔ ∀ a, (fun i => (ν - ν') i a) ∈ preLat (g⁻¹).1

theorem **cond_iff** (m g y κ'' : GL D k) (hκ'' : κ'' ∈ KO k D) (hy : y = m * g * κ'') (ν : Matrix D A k) : (∃ ν₁ ν₂ : Matrix D A k, (∀ i a, valuation k (ν₁ i a) < 1) ∧ (∀ i a, valuation k (ν₂ i a) < 1) ∧ m.1 * ν = ν₁ + y.1 * ν₂) ↔ ∀ a, (fun i => ν i a) ∈ preLat m.1 ⊔ preLat (g⁻¹).1

theorem **card_filter_eq** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D) (m g y : GL D k), (∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') → (∃ κ'' ∈ KO k D, y = m * g * κ'') → ∀ N : Finset (Matrix D A k), ((∀ ν ∈ N, ∀ i a, valuation k (ν i a) < 1) ∧ ∀ ν : Matrix D A k, (∀ i a, valuation k (ν i a) < 1) → ∃! ν', ν' ∈ N ∧ ∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν - ν' = g.1 * μ) → (N.filter fun ν => ∃ ν₁ ν₂ : Matrix D A k, (∀ i a, valuation k (ν₁ i a) < 1) ∧ (∀ i a, valuation k (ν₂ i a) < 1) ∧ m.1 * ν = ν₁ + y.1 * ν₂).card = ((preLat m.1 ⊓ preLat (g⁻¹).1).relIndex (preLat m.1 ⊓ mLat k D)) ^ Fintype.card A ∧ 0 < (preLat m.1 ⊓ preLat (g⁻¹).1).relIndex (preLat m.1 ⊓ mLat k D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardFilterLowerHeckeRepsEqRelIndexPowStatement

theorem **_root_.GlobalTheory.T.card_filter_lowerHecke_reps_eq_relIndex_pow** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D) (m g y : GL D k), (∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') → (∃ κ'' ∈ KO k D, y = m * g * κ'') → ∀ N : Finset (Matrix D A k), ((∀ ν ∈ N, ∀ i a, valuation k (ν i a) < 1) ∧ ∀ ν : Matrix D A k, (∀ i a, valuation k (ν i a) < 1) → ∃! ν', ν' ∈ N ∧ ∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν - ν' = g.1 * μ) → (N.filter fun ν => ∃ ν₁ ν₂ : Matrix D A k, (∀ i a, valuation k (ν₁ i a) < 1) ∧ (∀ i a, valuation k (ν₂ i a) < 1) ∧ m.1 * ν = ν₁ + y.1 * ν₂).card = ((preLat m.1 ⊓ preLat (g⁻¹).1).relIndex (preLat m.1 ⊓ mLat k D)) ^ Fintype.card A ∧ 0 < (preLat m.1 ⊓ preL

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardFilterParahoricCosetsEqSchurCount

theorem **val_lt_mul_right** {l m n : Type} [Fintype m] {c : Matrix l m k} {P : Matrix m n k} (hc : ∀ i j, valuation k (c i j) < 1) (hP : IsIntM P) (i j) : valuation k ((c * P) i j) < 1

theorem **isIntM_of_lt** {m n : Type} {M : Matrix m n k} (h : ∀ i j, valuation k (M i j) < 1) : IsIntM M

theorem **val_nlow** (v : Matrix D A k) : (nlow v).val = fromBlocks 1 0 v 1

theorem **nlow_mem** {v : Matrix D' A k} (hv : ∀ i j, valuation k (v i j) < 1) : nlow v ∈ parahoric k A D'

theorem **torus_int** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (s : Finset D) : IsIntM (torusElt k D ϖ (indic s)).val

theorem **nlow_mul_eq** {x : Matrix (A ⊕ D') (A ⊕ D) k} (hx : InSupp x) : (nlow (-(x.toBlocks₂₁ * x.toBlocks₁₁⁻¹))).val * x = fromBlocks x.toBlocks₁₁ x.toBlocks₁₂ 0 (schur x)

theorem **inSupp_mul_iff** {x : Matrix (A ⊕ D') (A ⊕ D) k} (hx : InSupp x) (V : Matrix (A ⊕ D') (A ⊕ D') k) (h11 : IsIntM V.toBlocks₁₁) (hd : valuation k V.toBlocks₁₁.det = 1) (h12 : IsIntM V.toBlocks₁₂) : InSupp (V * fromBlocks x.toBlocks₁₁ x.toBlocks₁₂ 0 (schur x)) ↔ (∀ i l, valuation k (V.toBlocks₂₁ i l) < 1) ∧ IsIntM (V.toBlocks₂₂ * schur x)

theorem **key_mem** {Y : GL (A ⊕ D') k} (hY : Y ∈ parahoric k A D') {E T : GL D' k} (hE : E ∈ KO k D') (hEY : E.val = Y.val.toBlocks₂₂) (hT : IsIntM T.val) (hl : ∀ i l, valuation k (((T⁻¹).val * Y.val.toBlocks₂₁) i l) < 1) : dg A T⁻¹ * Y * dg A (E⁻¹ * T) ∈ parahoric k A D'

theorem **card_filter_eq** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D D' : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] [Fintype D'] [DecidableEq D'] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D') (R : Finset (GL (A ⊕ D') k)) d, IsReps (mtor ϖ (indic s)) R d → ∀ RK, IsRepsK (torusElt k D' ϖ (indic s)) RK → ∀ x : Matrix (A ⊕ D') (A ⊕ D) k, InSupp x → (R.filter fun j => InSupp (((j * mtor ϖ (indic s))⁻¹).1 * x)).card = (RK.filter fun g => ∀ i l, ((g⁻¹).1 * schur x) i l ∈ 𝒪[k]).card

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardFilterParahoricCosetsEqSchurCountStatement

theorem **_root_.GlobalTheory.T.card_filter_parahoric_cosets_eq_schur_count** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D D' : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] [Fintype D'] [DecidableEq D'] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D') (R : Finset (GL (A ⊕ D') k)) d, IsReps (mtor ϖ (indic s)) R d → ∀ RK, IsRepsK (torusElt k D' ϖ (indic s)) RK → ∀ x : Matrix (A ⊕ D') (A ⊕ D) k, InSupp x → (R.filter fun j => InSupp (((j ∗ mtor ϖ (indic s))⁻¹).1 ∗ x)).card = (RK.filter fun g => ∀ i l, ((g⁻¹).1 ∗ schur x) i l ∈ 𝒪[k]).card

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardMatrixStepIndicesLeExp

theorem **card_index_le_sharp** : ∃ C : ℝ, 0 ≤ C ∧ ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ j, j + 2 ≤ N.R.m → (Fintype.card (I.β j) : ℝ) ≤ Real.exp (C ∗ errUnit N (Pr.T u))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardMatrixStepIndicesLeExpStatement

theorem **_root_.GlobalTheory.T.card_matrixStep_indices_le_exp** : ∃ C : ℝ, 0 ≤ C ∧ ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ j, j + 2 ≤ N.R.m → (Fintype.card (I.β j) : ℝ) ≤ Real.exp (C ∗ errUnit N (Pr.T u))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardOrbitsLeExp

theorem **card_orbits_le_exp** {D : D7} {N : NumDatum} {X : Fields} {s : Seam D X} {Pr : Primes X} (hH : HostTie N s Pr) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) {δ : Code N} {CY : ℝ} (hS : SizeRowsS CY N δ) {n : ℕ} (hn : 1 ≤ n) (hnN : n ≤ N.Nm) {ST SG : Finset (HeightOneSpectrum (𝒪 X.E))} (hST : ST ⊆ Pr.vT u) (hSG : SG = Pr.vGs) {G' D' : Type} [Group G'] [MulAction G' D'] (p : D' → DecoSet X.E n ST SG) (hp : Function.Injective p) (h : ∀ (g : UF X.E n) (x : D'), ∃ g' : G', p (g' • x) = (g, (1 : GLG X.E n SG)) • p x) : (Nat.card (MulAction.orbitRel.Quotient G' D') : ℝ) ≤ Real.exp (55 ∗ errUnit N (Pr.T u))

theorem **classNumbers_leS** : ∃ C : ℝ, 0 ≤ C ∧ ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → (∀ j ≤ N.R.m, ∀ b : I.β j, (Nat.card ((chF W sel mu I u j).mem b).Cl : ℝ) ≤ Real.exp (C ∗ errUnit N (Pr.T u))) ∧ (Nat.card (evGSD W Ch sel mu (rPsymF Ch) u).Cl : ℝ) ≤ Real.exp (C ∗ errUnit N (Pr.T u)) ∧ (topClassNumber Ch sel mu u : ℝ) ≤ Real.exp (C ∗ errUnit N (Pr.T u))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardParahoricDoubleCosetRepsInvEq

theorem **dcos_eq_iUnion** (J : Subgroup G) (g : G) (R : Finset G) (h1 : ∀ j ∈ R, j ∈ J) (h2 : ∀ j ∈ J, ∃! j', j' ∈ R ∧ (j' ∗ g)⁻¹ ∗ (j ∗ g) ∈ J) : dcos J g = ⋃ j ∈ R, (fun x => (j ∗ g)⁻¹ ∗ x) ⁻¹' (J : Set G)

theorem **pairwiseDisjoint_cosets** (J : Subgroup G) (g : G) (R : Finset G) (h1 : ∀ j ∈ R, j ∈ J) (h2 : ∀ j ∈ J, ∃! j', j' ∈ R ∧ (j' ∗ g)⁻¹ ∗ (j ∗ g) ∈ J) : (R : Set G).PairwiseDisjoint (fun j => (fun x => (j ∗ g)⁻¹ ∗ x) ⁻¹' (J : Set G))

theorem **dcos_inv** (J : Subgroup G) (g : G) : dcos J g⁻¹ = Inv.inv ⁻¹' dcos J g

theorem **measure_dcos** [MeasurableSpace G] [MeasurableMul G] (μ : Measure G) [μ.IsMulLeftInvariant] (J : Subgroup G) (hJ : MeasurableSet (J : Set G)) (g : G) (R : Finset G) (h1 : ∀ j ∈ R, j ∈ J) (h2 : ∀ j ∈ J, ∃! j', j' ∈ R ∧ (j' ∗ g)⁻¹ ∗ (j ∗ g) ∈ J) : μ (dcos J g) = R.card ∗ μ J

theorem **card_eq_of_unimodular** [MeasurableSpace G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [BorelSpace G] (μ : Measure G) [μ.IsHaarMeasure] [μ.IsMulRightInvariant] (J : Subgroup G) (hJo : IsOpen (J : Set G)) (hJt : μ J ≠ ⊤) (g : G) (R R' : Finset G) (h1 : ∀ j ∈ R, j ∈ J) (h2 : ∀ j ∈ J, ∃! j', j' ∈ R ∧ (j' ∗ g)⁻¹ ∗ (j ∗ g) ∈ J) (h1' : ∀ j ∈ R', j ∈ J) (h2' : ∀ j ∈ J, ∃! j', j' ∈ R' ∧ (j' ∗ g⁻¹)⁻¹ ∗ (j ∗ g⁻¹) ∈ J) : R.card = R'.card

theorem **card_isReps_inv** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (g : GL (A ⊕ D) k) (R R' : Finset (GL (A ⊕ D) k)) (d d' : GL (A ⊕ D) k → kˣ) : IsReps g R d → IsReps g⁻¹ R' d' → R.card = R'.card ∧ R.Nonempty

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardParahoricDoubleCosetRepsInvEqStatement

theorem **_root_.GlobalTheory.T.card_parahoricDoubleCosetReps_inv_eq** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (g : GL (A ⊕ D) k) (R R' : Finset (GL (A ⊕ D) k)) (d d' : GL (A ⊕ D) k → kˣ), IsReps g R d → IsReps g⁻¹ R' d' → R.card = R'.card ∧ R.Nonempty

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardRepsEq

theorem **card_reps_eq** : ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (H : Formalism.L2.Th1.H.Sesq V) (α : V → ℂ) (A B Λ : AddSubgroup V), SplitHyp H α A B Λ → ∀ SA SB : Finset V, Formalism.L2.Th1.H.IsReps B (Λ ⊓ A) SA → Formalism.L2.Th1.H.IsReps Λ B SB → SA.card = SB.card

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardResidueFieldOfRingEquivAdicCompletion

theorem **card_residue_of_ringEquiv** : ∀ (F : Type) [Field F] [NumberField F] (v : IsDedekindDomain.HeightOneSpectrum (NumberField.RingOfIntegers F)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (φ : k ≃+* v.adicCompletion F), (∀ a, Valued.v (φ a) ≤ 1 ↔ a ∈ 𝒪[k]) → Nat.card 𝓀[k] = Ideal.absNorm v.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardResidueFieldOfRingEquivAdicCompletionStatement

theorem **_root_.GlobalTheory.T.card_residueField_of_ringEquiv_adicCompletion** : ∀ (F : Type) [Field F] [NumberField F] (v : IsDedekindDomain.HeightOneSpectrum (NumberField.RingOfIntegers F)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (φ : k ≃+* v.adicCompletion F), (∀ a, Valued.v (φ a) ≤ 1 ↔ a ∈ 𝒪[k]) → Nat.card 𝓀[k] = Ideal.absNorm v.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardSubspacesContainingRangeQVandermonde

theorem **card_above_range_eq_sum** : ∀ (κ : Type) [Field κ] [Finite κ] (U U' : Type) [AddCommGroup U] [Module κ U] [FiniteDimensional κ U] [AddCommGroup U'] [Module κ U'] [FiniteDimensional κ U'] (f : U →ₗ[κ] U') (t e : ℕ), Module.finrank κ U' = Module.finrank κ U + t → e ≤ Module.finrank κ U' → Nat.card {X : Submodule κ U' // LinearMap.range f ≤ X ∧ Module.finrank κ X + e = Module.finrank κ U'} = ∑ n ∈ Finset.range (Module.finrank κ U + 1), (if n ≤ e then Nat.card κ ^ (n * (t - (e - n))) * Formalism.L2.Col.gaussBin (Nat.card κ) t (e - n) else 0) * Nat.card {X : Submodule κ U // X ≤ LinearMap.ker f ∧ Module.finrank κ X = n}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardSubspacesContainingRangeQVandermondeStatement

theorem **_root_.GlobalTheory.T.card_subspaces_containing_range_qVandermonde** : ∀ (κ : Type) [Field κ] [Finite κ] (U U' : Type) [AddCommGroup U] [Module κ U] [FiniteDimensional κ U] [AddCommGroup U'] [Module κ U'] [FiniteDimensional κ U'] (f : U →ₗ[κ] U') (t e : ℕ), Module.finrank κ U' = Module.finrank κ U + t → e ≤ Module.finrank κ U' → Nat.card {X : Submodule κ U' // LinearMap.range f ≤ X ∧ Module.finrank κ X + e = Module.finrank κ U'} = ∑ n ∈ Finset.range (Module.finrank κ U + 1), (if n ≤ e then Nat.card κ ^ (n * (t - (e - n))) * Formalism.L2.Col.gaussBin (Nat.card κ) t (e - n) else 0) * Nat.card {X : Submodule κ U // X ≤ LinearMap.ker f ∧ Module.finrank κ X = n}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CardValLt

theorem **card_val_lt** : ∀ (N n : ℕ), n ≤ N → Finset.card {i : Fin N | (↑i : ℕ) < n} = n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriedCentreAct

theorem **theta_of_tame** {r : ℕ} {Δ : (WO S T → ℂ) → (WO S T → ℂ)} (t : Tame ψ θ LinearMap.id r Δ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (Δ f)

theorem **Carried.comp** {L₁ L₂ : (GA D T → ℂ) → (GA D T → ℂ)} (h₁ : Carried ψ θ om L₁) (h₂ : Carried ψ θ om L₂) : Carried ψ θ om fun φ => L₁ (L₂ φ)

theorem **Carried.add** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {L₁ L₂ : (GA D T → ℂ) → (GA D T → ℂ)} (h₁ : Carried ψ θ om L₁) (h₂ : Carried ψ θ om L₂) : Carried ψ θ om fun φ => L₁ φ + L₂ φ

theorem **Carried.smul** (c : ℂ) {L : (GA D T → ℂ) → (GA D T → ℂ)} (hL : Carried ψ θ om L) : Carried ψ θ om fun φ => c • L φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionIffForallPlace

theorem **idAt_mono** {P Q : Pl D → Prop} (h : ∀ w, Q w → P w) {a : AE D} (ha : a ∈ idAt D Q) : a ∈ idAt D P

theorem **latV_mono** {S : HSp D} {L : Lat S} {P Q : Pl D → Prop} (h : ∀ w, Q w → P w) {y : Fin S.rank → AE D} (hy : y ∈ GlobalTheory.FinPl.latV S L Q) : y ∈ GlobalTheory.FinPl.latV S L P

theorem **mulVec_idAt** {S : HSp D} (g : GA D S) {Q : Pl D → Prop} {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D Q) : ∀ i, (mt S g *ᵥ y) i ∈ idAt D Q

theorem **latV_at_iff** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {v : Pl D} (hv : P v) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D (· = v)) : y ∈ GlobalTheory.FinPl.latV S L (· = v) ↔ y ∈ GlobalTheory.FinPl.latV S L P

theorem **moves_iff** : ∀ {D : D7} (S : HSp D) (g : GA D S) (L : GlobalTheory.FinPl.Lat S), Submodule.IsLattice D.host.E L → ∀ (P : GlobalTheory.FinPl.Pl D → Prop), GlobalTheory.FinPl.Moves S g L L P ↔ ∀ v, P v → GlobalTheory.FinPl.Moves S g L L (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionIffForallPlaceStatement

theorem **_root_.GlobalTheory.T.carriesLatticeCompletion_iff_forall_place** : ∀ {D : D7} (S : HSp D) (g : GA D S) (L : GlobalTheory.FinPl.Lat S), Submodule.IsLattice D.host.E L → ∀ (P : GlobalTheory.FinPl.Pl D → Prop), GlobalTheory.FinPl.Moves S g L L P ↔ ∀ v, P v → GlobalTheory.FinPl.Moves S g L L (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionOfEachPlace

theorem **latV_transfer** (S : HSp D) (L : Lat S) {P₁ P₂ : Pl D → Prop} (h : ∀ p, P₁ p → P₂ p) {z : Fin S.rank → AE D} (hz : ∀ i, z i ∈ idAt D P₁) : z ∈ latV S L P₁ ↔ z ∈ latV S L P₂

theorem **mem_latV_empty** (S : HSp D) (L : Lat S) {z : Fin S.rank → AE D} (hz : ∀ i, z i ∈ idAt D (· ∈ (∅ : Finset (Pl D)))) : z ∈ latV S L (· ∈ (∅ : Finset (Pl D)))

theorem **moves_of_places** : ∀ {D : D7} (S : HSp D) (g : GA D S) (L M : Lat S) (Q : Finset (Pl D)), (∀ v ∈ Q, Moves S g L M (· = v)) → Moves S g L M (· ∈ Q)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionOfEachPlaceStatement

theorem **_root_.GlobalTheory.T.carriesLatticeCompletion_of_each_place** : ∀ {D : D7} (S : HSp D) (g : GA D S) (L M : Lat S) (Q : Finset (GlobalTheory.FinPl.Pl D)), (∀ v ∈ Q, GlobalTheory.FinPl.Moves S g L M (· = v)) → GlobalTheory.FinPl.Moves S g L M (· ∈ Q)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionOfTrivialAtPlaces

theorem **GlobalTheory.fixes_latV_of_mem_grpOff** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop), ∀ g ∈ grpOff S P, Moves S g L L P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionOfTrivialAtPlacesStatement

theorem **_root_.GlobalTheory.T.carriesLatticeCompletion_of_trivialAtPlaces** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop), ∀ g ∈ grpOff S P, Moves S g L L P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionOnOffIffAllFinitePlaces

theorem **idem_eq_algebraMap** (R : Pl D → Prop) : idem D R = algebraMap (AF D.host.F) (AE D) (pairA D.host.F 0 (ePl D R))

theorem **idem_smul_mem** (S : HSp D) (L : Lat S) {P : Pl D → Prop} (R : Pl D → Prop) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : idem D R • y ∈ latV S L (fun p => P p ∧ R p)

theorem **idem_smul_eq** (S : HSp D) (R : Pl D → Prop) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D R) : idem D R • y = y

theorem **mem_latV_iff_all** (S : HSp D) (L : Lat S) (R : Pl D → Prop) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D R) : y ∈ latV S L R ↔ y ∈ latV S L (fun _ => True)

theorem **idem_add_idem** (P : Pl D → Prop) {a : AE D} (ha : a ∈ idAt D (fun _ => True)) : idem D P * a + idem D (fun p => ¬ P p) * a = a

theorem **mem_latV_iff_parts** (S : HSp D) (L : Lat S) (P : Pl D → Prop) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D (fun _ => True)) : y ∈ latV S L (fun _ => True) ↔ idem D (fun p => ¬ P p) • y ∈ latV S L (fun p => ¬ P p) ∧ idem D P • y ∈ latV S L P

theorem **idem_smul_supp** (S : HSp D) (R : Pl D → Prop) (y : Fin S.rank → AE D) (i : Fin S.rank) : (idem D R • y) i ∈ idAt D R

theorem **GlobalTheory.FinPl.moves_iff_moves_on_off** : ∀ {D : D7} (S : HSp D) (g : GA D S) (L : GlobalTheory.FinPl.Lat S) (P : GlobalTheory.FinPl.Pl D → Prop), (GlobalTheory.FinPl.Moves S g L L (fun p => ¬ P p) ∧ GlobalTheory.FinPl.Moves S g L L P) ↔ GlobalTheory.FinPl.Moves S g L L (fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CarriesLatticeCompletionOnOffIffAllFinitePlacesStatement

theorem **_root_.GlobalTheory.T.carriesLatticeCompletion_on_off_iff_all_finitePlaces** : ∀ {D : D7} (S : HSp D) (g : GA D S) (L : GlobalTheory.FinPl.Lat S) (P : GlobalTheory.FinPl.Pl D → Prop), (GlobalTheory.FinPl.Moves S g L L (fun p => ¬ P p) ∧ GlobalTheory.FinPl.Moves S g L L P) ↔ GlobalTheory.FinPl.Moves S g L L (fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanAntitone

theorem **permUnit_mem_KO** {N : ℕ} (σ : Equiv.Perm (Fin N)) : (permUnit σ : GL (Fin N) k) ∈ KO k (Fin N)

theorem **permUnit_val** {N : ℕ} (σ : Equiv.Perm (Fin N)) : ((permUnit σ : GL (Fin N) k) : Matrix (Fin N) (Fin N) k) = σ.toPEquiv.toMatrix

theorem **conj_torusElt** {N : ℕ} (ϖ : kˣ) (m : Fin N → ℤ) (σ : Equiv.Perm (Fin N)) : permUnit σ * torusElt k (Fin N) ϖ m * (permUnit σ)⁻¹ = torusElt k (Fin N) ϖ (m ∘ σ)

theorem **torusElt_conj** {N : ℕ} (ϖ : kˣ) (m : Fin N → ℤ) (σ : Equiv.Perm (Fin N)) : torusElt k (Fin N) ϖ m = (permUnit σ)⁻¹ * torusElt k (Fin N) ϖ (m ∘ σ) * permUnit σ

theorem **cartan_antitone** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ x : GL (Fin N) k, ∃ ν : Fin N → ℤ, Antitone ν ∧ ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), x = κ * torusElt k (Fin N) ϖ ν * κ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanAntitoneStatement

theorem **_root_.GlobalTheory.T.exists_antitone_cartan_decomposition** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ x : GL (Fin N) k, ∃ ν : Fin N → ℤ, Antitone ν ∧ ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), x = κ * torusElt k (Fin N) ϖ ν * κ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanDecompositionDominanceLocalField

theorem **det_mul_eq_sum_fun** {R : Type*} [CommRing R] {ι m : Type*} [Fintype ι] [DecidableEq ι] [Fintype m] [DecidableEq m] (A : Matrix ι m R) (B : Matrix m ι R) : (A * B).det = ∑ f : ι → m, (∏ β, B (f β) β) * (A.submatrix id f).det

theorem **prod_zpow_eq** {G : Type*} [CommGroupWithZero G] {ι : Type*} (s : Finset ι) {x : G} (hx : x ≠ 0) (e : ι → ℤ) : ∏ i ∈ s, x ^ e i = x ^ ∑ i ∈ s, e i

theorem **sum_le_sum_of_threshold** {α : Type*} [DecidableEq α] (e : α → ℤ) (U T : Finset α) (hc : T.card = U.card) (c : ℤ) (hU : ∀ i ∈ U, e i ≤ c) (hU' : ∀ i ∈ T, i ∉ U → c ≤ e i) : ∑ i ∈ U, e i ≤ ∑ i ∈ T, e i

theorem **sum_tail_le** {N : ℕ} (e : Fin N → ℤ) (he : Antitone e) (j : ℕ) (hj : j < N) (f : {i : Fin N // ¬ (i : ℕ) < j} → Fin N) (hf : Function.Injective f) : ∑ i ∈ Finset.univ.filter (fun i : Fin N => ¬ (i : ℕ) < j), e i ≤ ∑ x, e (f x)

theorem **val_det_le_one** {ι : Type*} [Fintype ι] [DecidableEq ι] (M : Matrix ι ι k) (hM : ∀ i j, valuation k (M i j) ≤ 1) : valuation k M.det ≤ 1

theorem **minor_mul_int** {m ι : Type*} [Fintype m] [DecidableEq m] [Fintype ι] [DecidableEq ι] (X A : Matrix m m k) (γ : ValueGroupWithZero k) (hA : ∀ i j, valuation k (A i j) ≤ 1) (hX : ∀ f h : ι → m, valuation k (X.submatrix f h).det ≤ γ) (f h : ι → m) : valuation k ((X * A).submatrix f h).det ≤ γ

theorem **minor_int_mul** {m ι : Type*} [Fintype m] [DecidableEq m] [Fintype ι] [DecidableEq ι] (A X : Matrix m m k) (γ : ValueGroupWithZero k) (hA : ∀ i j, valuation k (A i j) ≤ 1) (hX : ∀ f h : ι → m, valuation k (X.submatrix f h).det ≤ γ) (f h : ι → m) : valuation k ((A * X).submatrix f h).det ≤ γ

theorem **det_diag_mul_submatrix** {m ι : Type*} [Fintype m] [DecidableEq m] [Fintype ι] [DecidableEq ι] (d : m → k) (M : Matrix m m k) (f h : ι → m) : ((diagonal d * M).submatrix f h).det = (∏ x, d (f x)) * (M.submatrix f h).det

theorem **det_submatrix_eq_zero** {m ι : Type*} [Fintype ι] [DecidableEq ι] (M : Matrix m m k) (f h : ι → m) (hf : ¬ Function.Injective f) : (M.submatrix f h).det = 0

theorem **KO_int** {m : Type} [Fintype m] [DecidableEq m] {κ : GL m k} (hκ : κ ∈ KO k m) (i j : m) : valuation k (κ.1 i j) ≤ 1

theorem **KO_det** {m : Type} [Fintype m] [DecidableEq m] {κ : GL m k} (hκ : κ ∈ KO k m) : valuation k κ.1.det = 1

theorem **card_filter_lt** (N n : ℕ) (hn : n ≤ N) : (Finset.univ.filter fun i : Fin N => (i : ℕ) < n).card = n

theorem **indic_antitone** (N n : ℕ) : Antitone (indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n))

theorem **sum_indic** (N n : ℕ) (hn : n ≤ N) : ∑ i : Fin N, indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n) i = n

theorem **val_torus_prod** (ϖ : kˣ) {ι : Type*} [Fintype ι] (e : ι → ℤ) : valuation k (∏ x, (ϖ : k) ^ e x) = valuation k (ϖ : k) ^ ∑ x, e x

theorem **tail_diag** {N : ℕ} (ϖ : kˣ) (hϖ : valuation k (ϖ : k) = uniformizer k) (e : Fin N → ℤ) (he : Antitone e) (j : ℕ) (hj : j < N) (M : Matrix (Fin N) (Fin N) k) (γ : ValueGroupWithZero k) (hM : ∀ f h : {i : Fin N // ¬ (i : ℕ) < j} → Fin N, valuation k (M.submatrix f h).det ≤ γ) (f h : {i : Fin N // ¬ (i : ℕ) < j} → Fin N) : valuation k (((diagonal fun i => (ϖ : k) ^ e i) * M).submatrix f h).det ≤ valuation k (ϖ : k) ^ (∑ i ∈ Finset.univ.filter (fun i : Fin N => ¬ (i : ℕ) < j), e i) * γ

theorem **tail_bound** {N : ℕ} (ϖ : kˣ) (hϖ : valuation k (ϖ : k) = uniformizer k) (μ ν ε : Fin N → ℤ) (hμ : Antitone μ) (hε : Antitone ε) (g : GL (Fin N) k) (hg : ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ ε * κ') (hmain : ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') (j : ℕ) : ∑ i ∈ Finset.univ.filter (fun i : Fin N => ¬ (i : ℕ) < j), μ i + ∑ i ∈ Finset.univ.filter (fun i : Fin N => ¬ (i : ℕ) < j), ε i ≤ ∑ i ∈ Finset.univ.filter (fun i : Fin N => ¬ (i : ℕ) < j), ν i

theorem **total_eq** {N : ℕ} (ϖ : kˣ) (hϖ : valuation k (ϖ : k) = uniformizer k) (μ ν ε : Fin N → ℤ) (g : GL (Fin N) k) (hg : ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ ε * κ') (hmain : ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') : ∑ i, ν i = ∑ i, μ i + ∑ i, ε i

theorem **cartan_dominance** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (μ ν : Fin N → ℤ), Antitone μ → Antitone ν → ∀ n : ℕ, n ≤ N → ∀ g : GL (Fin N) k, (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ (indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n)) * κ') → (∃ κ

∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') → ∑ i, ν i = ∑ i, μ i + n ∧ ∀ j : ℕ, ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), ν i ≤ ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), (μ i + indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n) i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanDecompositionDominanceLocalFieldStatement

theorem **_root_.GlobalTheory.T.cartan_decomposition_dominance_local_field** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (μ ν : Fin N → ℤ), Antitone μ → Antitone ν → ∀ n : ℕ, n ≤ N → ∀ g : GL (Fin N) k, (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ (indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n)) * κ') → (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') → ∑ i, ν i = ∑ i, μ i + n ∧ ∀ j : ℕ, ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), ν i ≤ ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), (μ i + indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n) i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanDecompositionDominanceOrder

theorem **psum_succ** {N : ℕ} (ν : Fin N → ℤ) (i : Fin N) : psum ν (i + 1) = psum ν i + ν i

theorem **pot_lt** {N : ℕ} {μ ν : Fin N → ℤ} (h : ∀ j, psum μ j ≤ psum ν j) (hne : μ ≠ ν) : pot μ < pot ν

theorem **pot_nonneg** {N : ℕ} {ν : Fin N → ℤ} (h : ∀ i, 0 ≤ ν i) : 0 ≤ pot ν

theorem **pos_iff_lt** {N : ℕ} {e : Fin N → ℤ} (ha : Antitone e) (i : Fin N) : 0 < e i ↔ (i : ℕ) < (Finset.univ.filter fun j => 0 < e j).card

theorem **nonneg_of_dom** {N : ℕ} {ν e : Fin N → ℤ} (hν : Antitone ν) (he : ∀ i, 0 ≤ e i) (hle : ∀ j, psum ν j ≤ psum e j) (hsum : ∑ i, ν i = ∑ i, e i) (i : Fin N) : 0 ≤ ν i

theorem **torusElt_add'** {N : ℕ} (ϖ : kˣ) (a b : Fin N → ℤ) : torusElt k (Fin N) ϖ (a + b) = torusElt k (Fin N) ϖ a * torusElt k (Fin N) ϖ b

theorem **torusElt_zero'** {N : ℕ} (ϖ : kˣ) : torusElt k (Fin N) ϖ (fun _ => 0) = 1

theorem **torusElt_one_comm** {N : ℕ} (ϖ : kˣ) (g : GL (Fin N) k) : g * torusElt k (Fin N) ϖ (fun _ => 1) = torusElt k (Fin N) ϖ (fun _ => 1) * g

theorem **vanish_of_shift** {N : ℕ} {ϖ : kˣ} {c : ℕ → ℂ} (hc : c N ≠ 0) {h : GL (Fin N) k → ℂ} (hK : ∀ κ ∈ KO k (Fin N), ∀ m, h (κ * m) = h m ∧ h (m * κ) = h m) (hT : ∀ (s : Finset (Fin N)) R, IsRepsK (torusElt k (Fin N) ϖ (indic s)) R → ∀ m, ∑ g ∈ R, h (m * g) = c s.card * h m) (hex : ∀ g : GL (Fin N) k, ∃ R, IsRepsK g R) (j : ℕ) (m : GL (Fin N) k) (hm : h (m * torusElt k (Fin N) ϖ (fun _ => (j : ℤ))) = 0) : h m = 0

theorem **vanish_dominant** {N : ℕ} {ϖ : kˣ} {c : ℕ → ℂ} {h : GL (Fin N) k → ℂ} (hcart : ∀ x : GL (Fin N) k, ∃ ν : Fin N → ℤ, Antitone ν ∧ ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), x = κ * torusElt k (Fin N) ϖ ν * κ') (hdom : ∀ (μ ν : Fin N → ℤ), Antitone μ → Antitone ν → ∀ n : ℕ, n ≤ N → ∀ g : GL (Fin N) k, (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ (indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n)) * κ') → (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') → ∑ i, ν i = ∑ i, μ i + n ∧ ∀ j : ℕ, ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), ν i ≤ ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), (μ i + indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n) i)) (hex : ∀ g : GL (Fin N) k, ∃ R, IsRepsK g R) (hK : ∀ κ ∈ KO k (Fin N), ∀ m, h (κ * m) = h m ∧ h (m * κ) = h m) (hT : ∀ (s : Finset

theorem **zonal_vanish** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (ϖ : kˣ) (c : ℕ → ℂ), c N ≠ 0 → (∀ x : GL (Fin N) k, ∃ ν : Fin N → ℤ, Antitone ν ∧ ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), x = κ * torusElt k (Fin N) ϖ ν * κ') → (∀ (μ ν : Fin N → ℤ), Antitone μ → Antitone ν → ∀ n : ℕ, n ≤ N → ∀ g : GL (Fin N) k, (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ (indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n)) * κ') → (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') → ∑ i, ν i = ∑ i, μ i + n ∧ ∀ j : ℕ, ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), ν i ≤ ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), (μ i + indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n) i)) → (∀ g : GL (Fin N) k, ∃ R, IsRepsK g R) → ∀ h : GL (Fin N) k →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanDecompositionDominanceOrderStatement

theorem **_root_.GlobalTheory.T.cartan_decomposition_dominance_order** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (ϖ : kˣ) (c : ℕ → ℂ), c N ≠ 0 → (∀ x : GL (Fin N) k, ∃ ν : Fin N → ℤ, Antitone ν ∧ ∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), x = κ * torusElt k (Fin N) ϖ ν * κ') → (∀ (μ ν : Fin N → ℤ), Antitone μ → Antitone ν → ∀ n : ℕ, n ≤ N → ∀ g : GL (Fin N) k, (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), g = κ * torusElt k (Fin N) ϖ (indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n)) * κ') → (∃ κ ∈ KO k (Fin N), ∃ κ' ∈ KO k (Fin N), torusElt k (Fin N) ϖ μ * g = κ * torusElt k (Fin N) ϖ ν * κ') → ∑ i, ν i = ∑ i, μ i + n ∧ ∀ j : ℕ, ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), ν i ≤ ∑ i ∈ Finset.univ.filter (fun i : Fin N => (i : ℕ) < j), (μ i + indic (Finset.univ.filter fun i : Fin N => (i : ℕ) < n) i)) → (∀ g : GL (Fin N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanInvolutionEqOfCommute

theorem **isCartanInv_eq_of_commute** : ∀ (k : ℕ) (G s s' : Matrix (Fin k) (Fin k) ℂ), IsCartanInv G s → IsCartanInv G s' → s * s' = s' * s → s = s'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CartanInvolutionEqOfCommuteStatement

theorem **_root_.GlobalTheory.T.cartanInvolution_eq_of_commute** : ∀ (k : ℕ) (G s s' : Matrix (Fin k) (Fin k) ℂ), IsCartanInv G s → IsCartanInv G s' → s * s' = s' * s → s = s'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CasimirEqGelfandValueOfInfinitesimalCharacter

theorem **gelf_one_eq_sum** {n : ℕ} (ε : Fin n → Bool) (f : Matrix (Fin n) (Fin n) ℂ → ℂ) (g : Matrix (Fin n) (Fin n) ℂ) : GlobalTheory.Dec.gelf ε 1 f g = ∑ a, ∑ b, Formalism.REAL.Eab ε a b (Formalism.REAL.Eab ε b a f) g

theorem **casimir_eq_gelfVal_of_hasInfChar** : ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (ε : Fin n → Bool) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e ε → Factors R.U e σ → ∀ χ : Fin n → ℂ, GlobalTheory.Dec.HasInfChar ε χ σ → ∀ v, casimir R.HK.act v = GlobalTheory.Dec.gelfVal χ 2 • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CasimirEqGelfandValueOfInfinitesimalCharacterStatement

theorem **_root_.GlobalTheory.T.casimir_eq_gelfandValue_of_infinitesimalCharacter** : ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (ε : Fin n → Bool) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e ε → Factors R.U e σ → ∀ χ : Fin n → ℂ, GlobalTheory.Dec.HasInfChar ε χ σ → ∀ v, casimir R.HK.act v = GlobalTheory.Dec.gelfVal χ 2 • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CastGAMemGFset

theorem **castGA_mem_GFset** {I I' : Formalism.MainNotions.S.AutInst} (e : I = I') (x : I.GA) : Formalism.L2.SG.RowsJoinPinned.castGA e x ∈ I'.GFset ↔ x ∈ I.GFset

theorem **besGrp_iff_sA** (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (g : GA D (D.sp.Y 0)) : g ∈ besGrp B.d B.o ↔ (iotaX X h g).1 ∈ X.SinfU u ∧ (iotaX X h g).2 ∈ SfinU X u

theorem **besGrp_eq_sGroup** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A), (∀ g, g ∈ besGrp B.d B.o ↔ (iotaX X h g).1 ∈ X.SinfU u ∧ (iotaX X h g).2 ∈ SfinU X u) ∧ ∀ g, g ∈ GQ (D.sp.Y 0) ⊓ besGrp B.d B.o ↔ iotaX X h g ∈ Srat X u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CayleyRuleOfBiInvariantVacuumCoefficient

theorem **exists_cartanU** {L : Type} [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] {ϖ : L} {σ : L →+* L} {q : ℕ} (hU : IsUnramInv ϖ σ q) {n : ℕ} (g : GU σ n) : ∃ (d : Fin n → ℤ) (t : GU σ n), t.1.1 = Matrix.diagonal (fun i => ϖ ^ d i) ∧ ∃ k₁ ∈ KU σ n, ∃ k₂ ∈ KU σ n, g = k₁ * t * k₂

theorem **cayley_of_coeff** {L : Type} [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] {ϖ : L} {σ : L →+* L} {q : ℕ} (hU : IsUnramInv ϖ σ q) {M : Type} [AddCommGroup M] [Module ℂ M] {α β : Type} (ρ : Matrix α β L → Module.End ℂ M) {r N : ℕ} (π : GU σ r →* Module.End ℂ M) (s : M) (l : M →ₗ[ℂ] ℂ) (hls : l s ≠ 0) (hco : ∀ e : M →ₗ[ℂ] ℂ, (∀ u ∈ intMat L α β, ∀ m, e (ρ u m) = e m) → ∀ m, e m * l s = l m * e s) (φ : GU σ r → ℂ) (hφ : KerU ϖ q N ((-1) ^ N) φ) (hl : ∀ g, l (π g s) = φ g * l s) : CayleyRule q N ρ π s

theorem **GlobalTheory.ChainOnto.cayleyRule_of_vacCoeff** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : L) (σ : L →+* L) (q : ℕ) (δ : L) (ψ : AddChar L ℂ) (n n' : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (ρ : Matrix (Fin n') (Fin n) L → Module.End ℂ M) (ω : GU σ n × GU σ n' →* Module.End ℂ M) (s : M), IsStd ϖ σ q δ ψ ρ ω s → ∀ l : M →ₗ[ℂ] ℂ, l s ≠ 0 → (∀ u ∈ intMat L (Fin n') (Fin n), ∀ m, l (ρ u m) = l m) → (∀ φ : GU σ n → ℂ, KerU ϖ q n' ((-1) ^ n') φ → (∀ g, l (ω (g, 1) s) = φ g * l s) → CayleyRule q n' ρ (ω.comp (MonoidHom.inl _ _)) s) ∧ ∀ φ : GU σ n' → ℂ, KerU ϖ q n ((-1) ^ n) φ → (∀ g, l (ω (1, g) s) = φ g * l s) → CayleyRule q n ρ (ω.comp (MonoidHom.inr _ _)) s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CayleyRuleOfBiInvariantVacuumCoefficientStatement

theorem **_root_.GlobalTheory.T.cayleyRule_of_biInvariantVacuumCoefficient** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : L) (σ : L →+* L) (q : ℕ) (δ : L) (ψ : AddChar L ℂ) (n n' : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (ρ : Matrix (Fin n') (Fin n) L → Module.End ℂ M) (ω : GU σ n × GU σ n' →* Module.End ℂ M) (s : M), IsStd ϖ σ q δ ψ ρ ω s → ∀ l : M →ₗ[ℂ] ℂ, l s ≠ 0 → (∀ u ∈ intMat L (Fin n') (Fin n), ∀ m, l (ρ u m) = l m) → (∀ φ : GU σ n → ℂ, KerU ϖ q n' ((-1) ^ n') φ → (∀ g, l (ω (g, 1) s) = φ g * l s) → CayleyRule q n' ρ (ω.comp (MonoidHom.inl _ _)) s) ∧ ∀ φ : GU σ n' → ℂ, KerU ϖ q n ((-1) ^ n) φ → (∀ g, l (ω (1, g) s) = φ g * l s) → CayleyRule q n ρ (ω.comp (MonoidHom.inr _ _)) s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterActionSmoothCommuting

theorem **CommOnSm.symm** {A B : (GA D V → ℂ) → (GA D V → ℂ)} (h : CommOnSm θ V A B) : CommOnSm θ V B A

theorem **lin_sum** {T : (GA D V → ℂ) → (GA D V → ℂ)} (hT : LinOnSm θ V T) {α : Type} (s : Finset α) (g : α → GA D V → ℂ) (hg : ∀ a ∈ s, g a ∈ SmFun θ V) : T (∑ a ∈ s, g a) = ∑ a ∈ s, T (g a)

theorem **comm_foldr** {T : (GA D V → ℂ) → (GA D V → ℂ)} {α : Type} (A : α → (GA D V → ℂ) → (GA D V → ℂ)) (l : List α) (hl : ∀ a ∈ l, LinOnSm θ V (A a)) (hc : ∀ a ∈ l, CommOnSm θ V T (A a)) : CommOnSm θ V T fun f => l.foldr (fun a g => A a g) f

theorem **lin_opPow** {ι : Type} {A : ι → (GA D V → ℂ) → (GA D V → ℂ)} (hA : ∀ i, LinOnSm θ V (A i)) (k : ι →₀ ℕ) : LinOnSm θ V (opPow A k)

theorem **lin_polyAct** {ι : Type} {A : ι → (GA D V → ℂ) → (GA D V → ℂ)} (hA : ∀ i, LinOnSm θ V (A i)) (p : MvPolynomial ι ℂ) : LinOnSm θ V (polyAct A p)

theorem **comm_polyAct** {T : (GA D V → ℂ) → (GA D V → ℂ)} (hT : LinOnSm θ V T) {ι : Type} {A : ι → (GA D V → ℂ) → (GA D V → ℂ)} (hA : ∀ i, LinOnSm θ V (A i)) (hc : ∀ i, CommOnSm θ V T (A i)) (p : MvPolynomial ι ℂ) : CommOnSm θ V T (polyAct A p)

theorem **comm_gelf** {T : (GA D V → ℂ) → (GA D V → ℂ)} (hT : LinOnSm θ V T) {τ : D.what₂.Tau} (hc : ∀ Z, CommOnSm θ V T (RderC θ V τ Z)) (k : ℕ) : CommOnSm θ V T (gelf θ V τ k)

theorem **gelf_RderC** (hθ : IsInfFam θ) (τ τ' : D.what₂.Tau) (k : ℕ) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : CommOnSm θ V (gelf θ V τ k) (RderC θ V τ' Z)

theorem **lin_centreAct** (τ : D.what₂.Tau) (p : MvPolynomial ℕ ℂ) : LinOnSm θ V (centreAct θ τ V p)

theorem **centreAct_RderC** (hθ : IsInfFam θ) (τ τ' : D.what₂.Tau) (p : MvPolynomial ℕ ℂ) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : CommOnSm θ V (centreAct θ τ V p) (RderC θ V τ' Z)

theorem **centreAct_mem_comm** : ∀ (D : D7) (V : HSp D), ∀ θ : InfFam D, IsInfFam θ → ∀ (τ τ' : D.what₂.Tau) (p : MvPolynomial ℕ ℂ) (f : GA D V → ℂ), f ∈ SmFun θ V → centreAct θ τ V p f ∈ SmFun θ V ∧ (∀ Z, centreAct θ τ V p (RderC θ V τ' Z f) = RderC θ V τ' Z (centreAct θ τ V p f)) ∧ ∀ p', centreAct θ τ V p (centreAct θ τ' V p' f) = centreAct θ τ' V p' (centreAct θ τ V p f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterActionTransportChart

theorem **trace_pow_conj** {R : Type} [Ring R] [Algebra ℂ R] {n : ℕ} (E : Fin n → Fin n → R) (P Q : Matrix (Fin n) (Fin n) ℂ) (hQP : Q * P = 1) (m : ℕ) : ∑ c, (Matrix.of (fun a b => ∑ c, ∑ d, (P a c * Q d b) • E c d) ^ m) c c = ∑ c, (Matrix.of E ^ m) c c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterAdelicUnitaryGroupFinitePartScalar

theorem **cT_smul'** (σ : E →+* E) (c : E) (M : Matrix n n E) : cT σ (c • M) = σ c • cT σ M

theorem **cT_add'** (σ : E →+* E) (M N : Matrix n n E) : cT σ (M + N) = cT σ M + cT σ N

theorem **cT_sub'** (σ : E →+* E) (M N : Matrix n n E) : cT σ (M - N) = cT σ M - cT σ N

theorem **cT_natCast'** (σ : E →+* E) (N : ℕ) : cT σ (N : Matrix n n E) = N

theorem **map_natCast'** {R : Type} [CommRing R] (ψ : E →+* R) (N : ℕ) : (N : Matrix n n E).map ψ = N

theorem **commute_of_skew** {R : Type} [CommRing R] (ψ : E →+* R) (σ : E →+* E) (J : Matrix n n E) (hJ : J.det ≠ 0) (Z : Matrix n n R) (hZ : ∀ g ∈ UJ σ J, Commute Z ((g : Matrix n n E).map ψ)) (A : Matrix n n E) (hA : cT σ A * J + J * A = 0) : Commute Z (A.map ψ)

theorem **commute_all** {R : Type} [CommRing R] (ψ : E →+* R) (σ : E →+* E) (hσσ : ∀ x, σ (σ x) = x) (δ : E) (hδ : σ δ = -δ) (hδ0 : δ ≠ 0) (J : Matrix n n E) (hJh : cT σ J = J) (hJ : J.det ≠ 0) (Z : Matrix n n R) (hZ : ∀ g ∈ UJ σ J, Commute Z ((g : Matrix n n E).map ψ)) (M : Matrix n n E) : Commute Z (M.map ψ)

theorem **eq_smul_one** {R : Type} [CommRing R] (ψ : E →+* R) (σ : E →+* E) (hσσ : ∀ x, σ (σ x) = x) (δ : E) (hδ : σ δ = -δ) (hδ0 : δ ≠ 0) (J : Matrix n n E) (hJh : cT σ J = J) (hJ : J.det ≠ 0) (Z : Matrix n n R) (hZ : ∀ g ∈ UJ σ J, Commute Z ((g : Matrix n n E).map ψ)) : ∃ c : R, Z = c • (1 : Matrix n n R)

theorem **finE_ratPt** {D : D7} (V : Formalism.BasicNotions.TH.HSp D) (γ : GL (Fin V.rank) D.host.E) (hγ : γ ∈ UJ (D.sp.cm.sig : D.host.E →+* D.host.E) V.gram) : (Formalism.IndefR.Lev.finE V (ratPt V γ hγ)).val = (γ : Matrix (Fin V.rank) (Fin V.rank) D.host.E).map (ratE D)

theorem **finE_center_scalar** : ∀ {D : D7} (V : Formalism.BasicNotions.TH.HSp D) (z : GA D V), z ∈ Subgroup.center (GA D V) → z ∈ GfC V → ∃ c : Formalism.IndefR.Lev.EAf D, (Formalism.IndefR.Lev.finE V z).val = c • (1 : Matrix (Fin V.rank) (Fin V.rank) (Formalism.IndefR.Lev.EAf D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterAdelicUnitaryGroupFinitePartScalarStatement

theorem **_root_.GlobalTheory.T.center_adelicUnitaryGroup_finitePart_scalar** : ∀ {D : D7} (V : Formalism.BasicNotions.TH.HSp D) (z : GA D V), z ∈ Subgroup.center (GA D V) → z ∈ GfC V → ∃ c : Formalism.IndefR.Lev.EAf D, (Formalism.IndefR.Lev.finE V z).val = c • (1 : Matrix (Fin V.rank) (Fin V.rank) (Formalism.IndefR.Lev.EAf D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterAdelicUnitaryGroupScalarAtFinitePlace

theorem **mem_commSub** {φ : E →+* R'} {Z : Matrix n n R'} {A : Matrix n n E} : A ∈ commSub φ Z ↔ Z ∗ A.map φ = A.map φ ∗ Z

theorem **one_mem_commSub** (φ : E →+* R') (Z : Matrix n n R') : (1 : Matrix n n E) ∈ commSub φ Z

theorem **mem_commSub_of_inv** {φ : E →+* R'} {Z : Matrix n n R'} {A : Matrix n n E} (hA : IsUnit A.det) (h : A⁻¹ ∈ commSub φ Z) : A ∈ commSub φ Z

theorem **cT_add'** (A B : Matrix n n E) : cT σ (A + B) = cT σ A + cT σ B

theorem **cT_sub'** (A B : Matrix n n E) : cT σ (A - B) = cT σ A - cT σ B

theorem **cT_smul'** (c : E) (A : Matrix n n E) : cT σ (c • A) = σ c • cT σ A

theorem **det_cT'** (A : Matrix n n E) : (cT σ A).det = σ A.det

theorem **cayley_mem** (hZ : ∀ g ∈ UJ σ J, (g : Matrix n n E) ∈ commSub φ Z) (hJ : IsUnit J.det) {Y : Matrix n n E} (hY : cT σ Y ∗ J = -(J ∗ Y)) (hA : IsUnit (1 - Y).det) : Y ∈ commSub φ Z

theorem **eval_det_one_sub** (X : Matrix n n E) (t : E) : X.charpolyRev.eval t = (1 - t • X).det

theorem **lie_mem** (hZ : ∀ g ∈ UJ σ J, (g : Matrix n n E) ∈ commSub φ Z) (hJ : IsUnit J.det) {X : Matrix n n E} (hX : cT σ X ∗ J = -(J ∗ X)) : X ∈ commSub φ Z

theorem **cT_cT'** (hσσ : ∀ x, σ (σ x) = x) (A : Matrix n n E) : cT σ (cT σ A) = A

theorem **all_mem** (hσσ : ∀ x, σ (σ x) = x) {δ : E} (hδ0 : δ ≠ 0) (hδ : σ δ = -δ) (hJh : cT σ J = J) (hJ : IsUnit J.det) (hZ : ∀ g ∈ UJ σ J, (g : Matrix n n E) ∈ commSub φ Z) (M : Matrix n n E) : M ∈ commSub φ Z

theorem **map_single_one** (i j : n) : (Matrix.single i j (1 : E)).map φ = Matrix.single i j 1

theorem **scalar_of_comm** (hσσ : ∀ x, σ (σ x) = x) {δ : E} (hδ0 : δ ≠ 0) (hδ : σ δ = -δ) (hJh : cT σ J = J) (hJ : IsUnit J.det) (hZ : ∀ g ∈ UJ σ J, (g : Matrix n n E) ∈ commSub φ Z) : ∃ ζ : R', Z = ζ • (1 : Matrix n n R')

theorem **exists_sig_ne** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] (cm : CMData F E) : ∃ a : E, cm.sig a ≠ a

theorem **scalar_at_finite_place** : ∀ (D : D7) (V : Formalism.BasicNotions.TH.HSp D) (z : GA D V), z ∈ Subgroup.center (GA D V) → ∀ v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F), ∃ ζ : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, (GlobalTheory.AdelicU.matHom (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) D.sp.cm.sig V.gram (GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v z)).val = ζ • (1 : Matrix (Fin V.rank) (Fin V.rank) (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterAdelicUnitaryGroupScalarAtFinitePlaceStatement

theorem **_root_.GlobalTheory.T.center_adelicUnitaryGroup_scalar_at_finitePlace** : ∀ (D : D7) (V : Formalism.BasicNotions.TH.HSp D) (z : GA D V), z ∈ Subgroup.center (GA D V) → ∀ v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F), ∃ ζ : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, (GlobalTheory.AdelicU.matHom (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) D.sp.cm.sig V.gram (GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v z)).val = ζ • (1 : Matrix (Fin V.rank) (Fin V.rank) (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterEnvelopingAlgebraDirectSum

theorem **adjoin_range_ι_eq_top** (L : Type) [LieRing L] [LieAlgebra ℂ L] : Algebra.adjoin ℂ (Set.range (ι ℂ : L → UniversalEnvelopingAlgebra ℂ L)) = ⊤

theorem **mem_of_ι_mem** {L C : Type} [LieRing L] [LieAlgebra ℂ L] [Ring C] [Algebra ℂ C] (f : UniversalEnvelopingAlgebra ℂ L →ₐ[ℂ] C) (S : Subalgebra ℂ C) (h : ∀ x, f (ι ℂ x) ∈ S) (a : UniversalEnvelopingAlgebra ℂ L) : f a ∈ S

theorem **commute_of_ι** {L₁ L₂ C : Type} [LieRing L₁] [LieAlgebra ℂ L₁] [LieRing L₂] [LieAlgebra ℂ L₂] [Ring C] [Algebra ℂ C] (f : UniversalEnvelopingAlgebra ℂ L₁ →ₐ[ℂ] C) (g : UniversalEnvelopingAlgebra ℂ L₂ →ₐ[ℂ] C) (h : ∀ x y, Commute (f (ι ℂ x)) (g (ι ℂ y))) (a : UniversalEnvelopingAlgebra ℂ L₁) (b : UniversalEnvelopingAlgebra ℂ L₂) : Commute (f a) (g b)

theorem **bwdLie_apply** (p₁ : M →ₗ[ℂ] L₁) (p₂ : M →ₗ[ℂ] L₂) (x : M) : bwdLie p₁ p₂ x = ι ℂ (p₁ x) ⊗ₜ[ℂ] 1 + 1 ⊗ₜ[ℂ] ι ℂ (p₂ x)

theorem **fwd_tmul** (j₁ : L₁ →ₗ[ℂ] M) (j₂ : L₂ →ₗ[ℂ] M) (hc : ∀ x y, ⁅j₁ x, j₂ y⁆ = 0) (a b) : fwd j₁ j₂ hc (a ⊗ₜ b) = lift ℂ ((ι ℂ).comp j₁) a ∗ lift ℂ ((ι ℂ).comp j₂) b

theorem **decomp_left** (j₁ : L₁ →ₗ[ℂ] M) (j₂ : L₂ →ₗ[ℂ] M) (p₁ p₂ hc h11 h21 h12 h22 hs) (a) : decomp j₁ j₂ p₁ p₂ hc h11 h21 h12 h22 hs (a ⊗ₜ 1) = lift ℂ ((ι ℂ).comp j₁) a

theorem **decomp_right** (j₁ : L₁ →ₗ[ℂ] M) (j₂ : L₂ →ₗ[ℂ] M) (p₁ p₂ hc h11 h21 h12 h22 hs) (b) : decomp j₁ j₂ p₁ p₂ hc h11 h21 h12 h22 hs (1 ⊗ₜ b) = lift ℂ ((ι ℂ).comp j₂) b

theorem **toLieAlgebra_of** {N : Type} [LieRing N] [LieAlgebra ℂ N] (f : ∀ i, L i →ₗ[ℂ] N) (hf) (i : σ) (x : L i) : toLieAlgebra N f hf (of L i x) = f i x

theorem **projRestAt_self** (a : σ) (x : L a) : projRestAt L a a x = 0

theorem **projRestAt_ne** (a i : σ) (h : i ≠ a) (x : L i) : projRestAt L a i x = of (fun k : {i // i ≠ a} => L k.1) ⟨i, h⟩ x

theorem **inclRest_of** (a : σ) (k : {i // i ≠ a}) (x : L k.1) : inclRest L a (of (fun k : {i // i ≠ a} => L k.1) k x) = of L k.1 x

theorem **projRest_of** (a i : σ) (x : L i) : projRest L a (of L i x) = projRestAt L a i x

theorem **comp_of_same** (a : σ) (x : L a) : lieAlgebraComponent ℂ σ L a (of L a x) = x

theorem **comp_of_ne** (a i : σ) (h : i ≠ a) (x : L i) : lieAlgebraComponent ℂ σ L a (of L i x) = 0

theorem **algHom_ext_ι** {L C : Type} [LieRing L] [LieAlgebra ℂ L] [Ring C] [Algebra ℂ C] {f g : UniversalEnvelopingAlgebra ℂ L →ₐ[ℂ] C} (h : ∀ x, f (ι ℂ x) = g (ι ℂ x)) : f = g

theorem **center_directSum_card** (n : ℕ) : ∀ (σ : Type) [Fintype σ] [DecidableEq σ], Fintype.card σ = n → ∀ (L : σ → Type) [∀ i, LieRing (L i)] [∀ i, LieAlgebra ℂ (L i)] (m : ∀ i, UniversalEnvelopingAlgebra ℂ (L i) →ₐ[ℂ] UniversalEnvelopingAlgebra ℂ (⊕ i, L i)), (∀ i x, m i (ι ℂ x) = ι ℂ (DirectSum.of L i x : ⊕ i, L i)) → Subalgebra.center ℂ (UniversalEnvelopingAlgebra ℂ (⊕ i, L i)) = Algebra.adjoin ℂ (∪ i, m i '' (Subalgebra.center ℂ (UniversalEnvelopingAlgebra ℂ (L i)) : Set _))

theorem **center_directSum** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (L : ι → Type) [∀ i, LieRing (L i)] [∀ i, LieAlgebra ℂ (L i)] (m : ∀ i, UniversalEnvelopingAlgebra ℂ (L i) →ₐ[ℂ] UniversalEnvelopingAlgebra ℂ (⊕ i, L i)), (∀ i x, m i (UniversalEnvelopingAlgebra.ι ℂ x) = UniversalEnvelopingAlgebra.ι ℂ (DirectSum.of L i x : ⊕ i, L i)) → Subalgebra.center ℂ (UniversalEnvelopingAlgebra ℂ (⊕ i, L i)) = Algebra.adjoin ℂ (∪ i, m i '' (Subalgebra.center ℂ (UniversalEnvelopingAlgebra ℂ (L i)) : Set _))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterLocalUnitaryGroupFinitePlaceScalar

theorem **cT_add'** (σ : A →+* A) (M N : Matrix n n A) : cT σ (M + N) = cT σ M + cT σ N

theorem **cT_sub'** (σ : A →+* A) (M N : Matrix n n A) : cT σ (M − N) = cT σ M − cT σ N

theorem **cT_smul'** (σ : A →+* A) (c : A) (M : Matrix n n A) : cT σ (c • M) = σ c • cT σ M

theorem **cT_cT'** (σ : A →+* A) (hσ : ∀ x, σ (σ x) = x) (M : Matrix n n A) : cT σ (cT σ M) = M

theorem **cayley_unitary** (σ : A →+* A) (J B V : Matrix n n A) (s : A) (hs : σ s = s) (hB : cT σ B * J + J * B = 0) (hV1 : (s • (1 : Matrix n n A) − B) * V = 1) : cT σ ((s • (1 : Matrix n n A) + B) * V) * J * ((s • (1 : Matrix n n A) + B) * V) = J

theorem **comm_of_comm_cayley** (g B V : Matrix n n A) (s : A) (h2s : IsUnit (2 * s)) (hV1 : (s • (1 : Matrix n n A) − B) * V = 1) (hV2 : V * (s • (1 : Matrix n n A) − B) = 1) (hg : g * ((s • (1 : Matrix n n A) + B) * V) = ((s • (1 : Matrix n n A) + B) * V) * g) : g * B = B * g

theorem **comm_of_comm_skew** (σ : A →+* A) (hσ : ∀ x, σ (σ x) = x) (J : Matrix n n A) (hJ : cT σ J = J) (hJu : IsUnit J.det) (h2 : IsUnit (2 : A)) (δ : A) (hδ : IsUnit δ) (hδσ : σ δ = −δ) (g : Matrix n n A) (hg : ∀ B : Matrix n n A, cT σ B * J + J * B = 0 → g * B = B * g) (M : Matrix n n A) : g * M = M * g

theorem **exists_nat_isUnit_sub** (K : Type) [Field K] [CharZero K] [Algebra K A] [FiniteDimensional K A] (B : Matrix n n A) : ∃ m : ℕ, m ≠ 0 ∧ IsUnit ((m : A) • (1 : Matrix n n A) − B)

theorem **scalar_of_comm_UJ** (K : Type) [Field K] [CharZero K] [Algebra K A] [FiniteDimensional K A] (σ : A →+* A) (hσ : ∀ x, σ (σ x) = x) (J : Matrix n n A) (hJ : cT σ J = J) (hJu : IsUnit J.det) (δ : A) (hδ : IsUnit δ) (hδσ : σ δ = −δ) (g : Matrix n n A) (hg : ∀ h ∈ UJ σ J, g * (h : Matrix n n A) = (h : Matrix n n A) * g) : ∃ c : A, g = c • (1 : Matrix n n A)

theorem **exists_anti** (D : D7) : ∃ e : D.host.E, D.sp.cm.sig e = −e ∧ e ≠ 0

theorem **center_scalar_loc** : ∀ (D : D7) (V : Formalism.BasicNotions.TH.HSp D) (v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F)) (g : GlobalTheory.AdelicU.GLoc D.sp.cm.sig V.gram v), g ∈ Subgroup.center (GlobalTheory.AdelicU.GLoc D.sp.cm.sig V.gram v) → ∃ c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, (GlobalTheory.AdelicU.matHom (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) D.sp.cm.sig V.gram g).1 = c • (1 : Matrix (Fin V.rank) (Fin V.rank) (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterLocalUnitaryGroupFinitePlaceScalarStatement

theorem **_root_.GlobalTheory.T.center_localUnitaryGroup_finitePlace_scalar** : ∀ (D : D7) (V : Formalism.BasicNotions.TH.HSp D) (v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F)) (g : GlobalTheory.AdelicU.GLoc D.sp.cm.sig V.gram v), g ∈ Subgroup.center (GlobalTheory.AdelicU.GLoc D.sp.cm.sig V.gram v) → ∃ c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, (GlobalTheory.AdelicU.matHom (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) D.sp.cm.sig V.gram g).1 = c • (1 : Matrix (Fin V.rank) (Fin V.rank) (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterMtEqSmulOne

theorem **eval_charpolyRev_eq** (M : Matrix n n E) (z : E) : M.charpolyRev.eval z = (1 − z • M).det

theorem **cT_smul** (σ : E →+* E) (c : E) (M : Matrix n n E) : Formalism.UG.cT σ (c • M) = σ c • Formalism.UG.cT σ M

theorem **cT_add** (σ : E →+* E) (M N : Matrix n n E) : Formalism.UG.cT σ (M + N) = Formalism.UG.cT σ M + Formalism.UG.cT σ N

theorem **cT_sub** (σ : E →+* E) (M N : Matrix n n E) : Formalism.UG.cT σ (M − N) = Formalism.UG.cT σ M − Formalism.UG.cT σ N

theorem **cT_cT** (σ : E →+* E) (hσ : ∀ x, σ (σ x) = x) (M : Matrix n n E) : Formalism.UG.cT σ (Formalism.UG.cT σ M) = M

theorem **cayley_unitary** (σ : E →+* E) (J X : Matrix n n E) (t : E) (ht : σ t = t) (hX : Formalism.UG.cT σ X * J + J * X = 0) (hA : (1 − t • X).det ≠ 0) : Formalism.UG.cT σ ((2 : E) • (1 − t • X)⁻¹ − 1) * J * ((2 : E) • (1 − t • X)⁻¹ − 1) = J

theorem **map_smulE** (c : E) (M : Matrix n n E) : (c • M).map ι = ι c • M.map ι

theorem **map_subE** (M N : Matrix n n E) : (M − N).map ι = M.map ι − N.map ι

theorem **map_addE** (M N : Matrix n n E) : (M + N).map ι = M.map ι + N.map ι

theorem **map_oneE** : (1 : Matrix n n E).map ι = 1

theorem **map_mulE** (M N : Matrix n n E) : (M * N).map ι = M.map ι * N.map ι

theorem **commute_skew** (σ : E →+* E) (J : Matrix n n E) (hF : {t : E | σ t = t}.Infinite) (h2 : (2 : E) ≠ 0) (Z : Matrix n n R) (hZ : ∀ g : Matrix n n E, Formalism.UG.cT σ g * J * g = J → Commute Z (g.map ι)) (X : Matrix n n E) (hX : Formalism.UG.cT σ X * J + J * X = 0) : Commute Z (X.map ι)

theorem **commute_all** (σ : E →+* E) (hσ : ∀ x, σ (σ x) = x) (J : Matrix n n E) (hJh : Formalism.UG.cT σ J = J) (hJ : J.det ≠ 0) (ξ : E) (hξ : σ ξ = −ξ) (hξ0 : ξ ≠ 0) (hF : {t : E | σ t = t}.Infinite) (h2 : (2 : E) ≠ 0) (Z : Matrix n n R) (hZ : ∀ g : Matrix n n E, Formalism.UG.cT σ g * J * g = J → Commute Z (g.map ι)) (M : Matrix n n E) : Commute Z (M.map ι)

theorem **eq_smul_one** (σ : E →+* E) (hσ : ∀ x, σ (σ x) = x) (J : Matrix n n E) (hJh : Formalism.UG.cT σ J = J) (hJ : J.det ≠ 0) (ξ : E) (hξ : σ ξ = −ξ) (hξ0 : ξ ≠ 0) (hF : {t : E | σ t = t}.Infinite) (h2 : (2 : E) ≠ 0) (Z : Matrix n n R) (hZ : ∀ g : Matrix n n E,

Formalism.UG.cT σ g * J * g = J → Commute Z (g.map ι)) : ∃ c : R, Z = c • (1 : Matrix n n R)

theorem **exists_skew** (D : D7) : ∃ ξ : D.host.E, (D.sp.cm.sig : D.host.E →+* D.host.E) ξ = -ξ ∧ ξ ≠ 0

theorem **fixed_infinite** (D : D7) : {t : D.host.E | (D.sp.cm.sig : D.host.E →+* D.host.E) t = t}.Infinite

theorem **center_mt_eq_smul_one** : ∀ (D : D7) (V : Formalism.BasicNotions.TH.HSp D) (z : GA D V), z ∈ Subgroup.center (GA D V) → ∃ c : AE D, Formalism.L2.Theta.mt V z = c • (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CenterTensorProductMemRange

theorem **center_tensorProduct_mem_range** {A B : Type} [Ring A] [Algebra ℂ A] [Ring B] [Algebra ℂ B] (z : A ⊗[ℂ] B) (hz : z ∈ Subalgebra.center ℂ (A ⊗[ℂ] B)) : ∃ u : (Subalgebra.center ℂ A) ⊗[ℂ] (Subalgebra.center ℂ B), Algebra.TensorProduct.map (Subalgebra.center ℂ A).val (Subalgebra.center ℂ B).val u = z

theorem **center_eq_adjoin** : ∀ (A B C : Type) [Ring A] [Algebra ℂ A] [Ring B] [Algebra ℂ B] [Ring C] [Algebra ℂ C] (e : (A ⊗[ℂ] B) ≃ₐ[ℂ] C), Subalgebra.center ℂ C = Algebra.adjoin ℂ ((fun a => e (a ⊗ₜ 1)) '' (Subalgebra.center ℂ A : Set A) ∪ (fun b => e (1 ⊗ₜ b)) '' (Subalgebra.center ℂ B : Set B))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralAdaptedElementProperty

theorem **h0_eq** (D : D7) : 0 + 2 * D.R.rS + 2 = D.R.m

theorem **conj_mem_K_two** (W : RealPack ex) (hPk : PkHyp W) {u : D.what₂.Word} (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) {v : VCom f.c} (G : (GlobalTheory.BaseChain.com W).Pg → GLoc v) (hBot : GlobalTheory.LevZero.Bot (c

theorem **mem_K_two_iff** (W : RealPack ex) (hPk : PkHyp W) {u : D.what₂.Word} (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) {v : VCom f.c} (G : (GlobalTheory.BaseChain.com W).Pg → GLoc v) (hBot : GlobalTheory.LevZero.Bot (c

theorem **mem_rowJ_iff_rTup** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∀ a ∈ GfC (D.sp.X 0), a ∈ Subgroup.center (GA D (D.sp.X 0)) → GlobalTheory.LevZero.StabAt f B a₀ a → ∀ κ : GA D (D.sp.Y 0), κ ∈ GfC (D.sp.Y 0) → finE (D.sp.Y 0) κ ∈ Formalism.L2.DIV.St (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralAdaptedElementPropertyStatement

theorem **_root_.GlobalTheory.T.central_adapted_element_property** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∀ a ∈ GfC (D.sp.X 0), a ∈ Subgroup.center (GA D (D.sp.X 0)) → GlobalTheory.LevZero.StabAt f B a₀ a → ∀ κ : GA D (D.sp.Y 0), κ ∈ GfC (D.sp.Y 0) → finE (D.sp.Y 0) κ ∈ Formalism.L2.DIV.St (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralAdaptedElementValue

theorem **rowChi_conj_of_mem** (W : RealPack ex) (hW : PkHyp W) (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (hPl : Placed f B) (a₀ : GA D (D.sp.X 0)) (hH0 : IsTransportAdelicCorrection (W.S.tr u) a₀) (a : GA D (D.sp.X 0)) (hac : a ∈ Subgroup.center (GA D (D.sp.X 0))) (t s : GA D (D.sp.Y 0)) (ht : t = f.g.b 0) (hs : s = ((iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u))⁻¹ * embA B.d B.o (a₀ * a) * t) (β : GL (Fin (D.sp.Y 0).rank) (Lev.EAf D)) (hβ : β = (pk W).βY 0) (Lz : Formalism.L2.DIV.LevDat D.host.E ℂ (D.sp.Y 0).rank) (hLz : Lz = GlobalTheory.LevZero.datZero W) (hSt : ∀ κ : GA D (D.sp.Y 0), (κ ∈ GfC (D.sp.Y 0) ∧ finE (D.sp.Y 0) κ ∈ St (finE (D.sp.Y 0) s * β)) ↔ ((iotaX W.S.X W.Q.instEq κ).1 = 1 ∧ (iotaX W.S.X W.Q.instEq κ).2 ∈ W.S.X.B.I.Kset)) (hmem : ∀ k ∈ levS (D.sp.Y 0) β Lz, s * k * s⁻¹ ∈ rowJ W.S.X W.Q.instEq) : ∀ k ∈ levS (D.sp.Y 0) β Lz, rowChi

theorem **row_character_eq_extension** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∀ a ∈ GfC (D.sp.X 0), a ∈ Subgroup.center (GA D (D.sp.X 0)) → GlobalTheory.LevZero.StabAt f B a₀ a → ∀ κ : GA D (D.sp.Y 0), κ ∈ GfC (D.sp.Y 0) → finE (D.sp.Y 0) κ ∈ Formalism.L2.DIV.St (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralAdaptedElementValueStatement

theorem **_root_.GlobalTheory.T.central_adapted_element_value** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∀ a ∈ GfC (D.sp.X 0), a ∈ Subgroup.center (GA D (D.sp.X 0)) → GlobalTheory.LevZero.StabAt f B a₀ a → ∀ κ : GA D (D.sp.Y 0), κ ∈ GfC (D.sp.Y 0) → finE (D.sp.Y 0) κ ∈ Formalism.L2.DIV.St (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralEigenRealDatumFiniteSumFockProducts

theorem **span_eval_eq_top** {α : Type*} (E : Submodule ℂ (α → ℂ)) [FiniteDimensional ℂ E] : Submodule.span ℂ (Set.range fun u : α => (LinearMap.proj u : (α → ℂ) →ₗ[ℂ] ℂ).comp E.subtype) = ⊤

theorem **sum_prod_of_slices** {α β : Type*} (C : α → β → ℂ) (E : Submodule ℂ (α → ℂ)) [FiniteDimensional ℂ E] (E' : Submodule ℂ (β → ℂ)) (hl : ∀ v, (fun u => C u v) ∈ E) (hr : ∀ u, C u ∈ E') : ∃ (n : ℕ) (e : Fin n → α → ℂ) (g : Fin n → β → ℂ), (∀ i, e i ∈ E) ∧ (∀ i, g i ∈ E') ∧ ∀ u v, C u v = ∑ i, e i u * g i v

theorem **biVac_sum_fock** : ∀ (D : D7), CenFock D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → ∀ (χ : GA D A → ℂ) (χ' : GA D S → ℂ) (C : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L y.L z.L C → (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k.om (a, 1) (fun u => C u v) = χ a • fun u => C u v) → (∀ b ∈ GinfC S, b ∈ Subgroup.center (GA D S) → ∀ u, k'.om (b, 1) (C u) = χ' b • C u) → ∃ (n : ℕ) (P : Fin n → PolF D S.rank A.rank) (P' : Fin n → PolF D A'.rank S.rank), ∀ u v, C u v = ∑ i, pureTensorO ψ θ A S x.s y.s x.B y.B k.vac (P i) u * pureTensorO ψ θ S A' y.s z.s y.B z.B k'.vac (P' i) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralEigenRealDatumFiniteSumFockProductsStatement

theorem **_root_.GlobalTheory.T.centralEigen_realDatum_finite_sum_fockProducts** : ∀ (D : D7), CenFock D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → ∀ (χ : GA D A → ℂ) (χ' : GA D S → ℂ) (C : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L

y.L z.L C → (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k.om (a, 1) (fun u => C u v) = χ a • fun u => C u v) → (∀ b ∈ GinfC S, b ∈ Subgroup.center (GA D S) → ∀ u, k'.om (b, 1) (C u) = χ' b • C u) → ∃ (n : ℕ) (P : Fin n → PolF D S.rank A.rank) (P' : Fin n → PolF D A'.rank S.rank), ∀ u v, C u v = ∑ i, pureTensorO ψ θ A S x.s y.s x.B y.B k.vac (P i) u * pureTensorO ψ θ S A' y.s z.s y.B z.B k'.vac (P' i) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralEigenThetaIsFockVectorBoundedDegree

theorem **thPair_smul_smul** {m n : ℕ} (μ : Measure (Mx D m n)) (ℱ : Set (Mx D m n)) (a b : ℂ) (f g : Mx D m n → ℂ) : thPair μ ℱ (a • f) (b • g) = (a * star b) * thPair μ ℱ f g

theorem **heisO_sub** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (u : WO S T) (f g : WO S T → ℂ) : heisO ψ S T u (f - g) = heisO ψ S T u f - heisO ψ S T u g

theorem **isLatVac_sub** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) {Λ : AddSubgroup (WO S T)} {f g : WO S T → ℂ} (hf : IsLatVac (heisO ψ S T) Λ f) (hg : IsLatVac (heisO ψ S T) Λ g) : IsLatVac (heisO ψ S T) Λ (f - g)

theorem **comb_single_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single 0 1) f = f

theorem **isFinComb_single_zero** (S T : HSp D) : IsFinComb (Finsupp.single (0 : WO S T) (1 : ℂ))

theorem **thPair_sum_right** {ι : Type} (s : Finset ι) (a : ι → ℂ) (g : Mx D m n → ℂ) (h : ι → Mx D m n → ℂ) (hi : ∀ i, Integrable (fun w => g w * star (h i w)) (μ.restrict ℱ)) : thPair μ ℱ g (∑ i ∈ s, a i • h i) = ∑ i ∈ s, star (a i) * thPair μ ℱ g (h i)

theorem **thPair_sub_sum_left** {ι : Type} (s : Finset ι) (a : ι → ℂ) (f g : Mx D m n → ℂ) (h : ι → Mx D m n → ℂ) (hf : Integrable (fun w => f w * star (g w)) (μ.restrict ℱ)) (hi : ∀ i, Integrable (fun w => h i w * star (g w)) (μ.restrict ℱ)) : thPair μ ℱ (f - ∑ i ∈ s, a i • h i) g = thPair μ ℱ f g - ∑ i ∈ s, a i * thPair μ ℱ (h i) g

theorem **exists_orth_coeff** {X : Type} [MeasurableSpace X] (ν : Measure X) {ι : Type} [Fintype ι] (f : X → ℂ) (h : ι → X → ℂ) (hf : MemLp f 2 ν) (hh : ∀ i, MemLp (h i) 2 ν) : ∃ c : ι → ℂ, ∀ j, ∫ w, f w * star (h j w) ∂ν = ∑ i, c i * ∫ w, h i w * star (h j w) ∂ν

theorem **isLatVac_pureTensorO** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {x : DefSp S} {y : DefSp T} {f₀ : WO S T → ℂ} (hvac : IsVacO ψ θ S T x.s y.s x.L y.L f₀) (φ : MvPolynomial (FVar D T.rank S.rank) ℂ) : IsLatVac (heisO ψ S T) (latO S T x.L y.L fun _ => True) (pureTensorO ψ θ S T x.s y.s x.B y.B f₀ φ)

theorem **thPair_eq_zero_of_eigen** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χ χ' : (AE D)ˣ →* Circle} {om : Om S T} (hW : IsWeilDatumO ψ S T χ χ' om) (v : VolM D T.rank S.rank) (ρ : GA D S × GA D T) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) {a b : ℂ} (ha : om ρ f = a • f) (hb : om ρ g = b • g) (hb1 : ‖b‖ = 1) (hab : b ≠ a) : thPair v.μ v.ℱ f g = 0

theorem **cenFock** : ∀ D : D7, CenFock D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CentralEigenThetaIsFockVectorBoundedDegreeStatement

theorem **_root_.GlobalTheory.T.centralEigenTheta_isFockVector_bounded_degree** : ∀ D : D7, CenFock D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CexpTwoPiINegInvNatSmulAddStar

theorem **phi_star** (v : InfinitePlace I.F) (x : I.E) : I.φ v (star x) = star (I.φ v x)

theorem **hJ_frame** (v : InfinitePlace I.F) (x y : Fin I.n → ℂ) (N : Matrix (Fin I.n) (Fin I.n) ℂ) : hJ (I.ε v) ((I.P v)⁻¹ *ᵥ x) (((I.P v)⁻¹ * N * I.P v) *ᵥ ((I.P v)⁻¹ *ᵥ y)) = star x ⬝ᵥ (I.J.map (I.φ v) *ᵥ (N *ᵥ y))

theorem **theta_form** (v : InfinitePlace I.F) (M : Matrix (Fin I.n) (Fin I.n) (AF I.F ⊗[I.F] I.E)) (p q : Fin I.n → I.E) : thetaV I v (∑ i, ∑ j, inclR (AF I.F) (star (p i) * I.J i j) * (M *ᵥ fun k => inclR (AF I.F) (q k)) j) = hJ (I.ε v) (rv I v p) (((I.P v)⁻¹ * M.map (thetaV I v) * I.P v) *ᵥ rv I v q)

theorem **sig_inclE** (x : I.E) : Formalism.UG.sigR I.cm.sig (Formalism.MainNotions.S.Af I.F) (Formalism.UG.inclE (F

theorem **tau_form** (M : Matrix (Fin I.n) (Fin I.n) (AF I.F ⊗[I.F] I.E)) (p q : Fin I.n → I.E) : tauF I (∑ i, ∑ j, inclR (AF I.F) (star (p i) * I.J i j) * (M *ᵥ fun k => inclR (AF I.F) (q k)) j) = hA I (incl I p) (M.map (tauF I) *ᵥ incl I q)

theorem **trace_tmul'** (a : AF I.F) (x : I.E) : Algebra.trace (AF I.F) (AF I.F ⊗[I.F] I.E) (a ⊗ₜ[I.F] x) = a * algebraMap I.F (AF I.F) (Algebra.trace I.F I.E x)

theorem **evV_trace** (v : InfinitePlace I.F) (y : AF I.F ⊗[I.F] I.E) : evV I v (Algebra.trace (AF I.F) (AF I.F ⊗[I.F] I.E) y) = thetaV I v y + star (thetaV I v y)

theorem **tmul_mem_int** {x : I.E} (hx : IsIntegral ℤ x) {b : Formalism.MainNotions.S.Af I.F} (hb : b ∈ Formalism.UG.intAd I.F) : x ⊗ₜ[I.F] b ∈ IntAf I.F I.E

theorem **fin_char** (ψ : AddChar (AF I.F) ℂ) (hψr : ∀ x : I.F, ψ (pairA I.F 0 (algebraMap I.F (Formalism.MainNotions.S.Af I.F) x)) = Complex.exp (2 * Real.pi * Complex.I * ((Algebra.trace ℚ I.F x : ℚ) : ℂ))) (hψi : ∀ β ∈ Formalism.UG.intAd I.F, ψ (pairA I.F 0 β) = 1) (psiEf : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F → ℂ) (hadd : ∀ x y, psiEf (x + y) = psiEf x * psiEf y) (hint : ∀ x ∈ IntAf I.F I.E, psiEf x = 1) (hrat : ∀ x : I.E, psiEf (Formalism.UG.inclE (F

theorem **hN_smul** {D : D7} (n : GA D (D.sp.Y 0)) (a b : D.host.E) (p q : Fin (D.sp.Y 0).rank → D.host.E) : hN n (a • p) (b • q) = incA D (star a * b) * hN n p q

theorem **pX_mulVec** {D : D7} {V₀ : HSp D} {ρ : ℕ} (d : HypDec (D.sp.Y 0) V₀ ρ) (n : GA D (D.sp.Y 0)) (q : Fin (D.sp.Y 0).rank → D.host.E) (a : Fin ρ) : (d.pX *ᵥ (mt (D.sp.Y 0) n *ᵥ ratV q)) a = hN n (fun k => d.e' k a) q

theorem **blockX_eq** {D : D7} {V₀ : HSp D} {ρ : ℕ} (d : HypDec (D.sp.Y 0) V₀ ρ) (n : GA D (D.sp.Y 0)) (a b : Fin ρ) : blockX d n a b = hN n (fun k => d.e' k a) (fun k => d.e k b)

theorem **coefX_eq** {D : D7} {V₀ : HSp D} {ρ : ℕ} (d : HypDec (D.sp.Y 0) V₀ ρ) (z : Fin V₀.rank → D.host.E) (n : GA D (D.sp.Y 0)) (a : Fin ρ) : coefX d z n a = hN n (fun k => d.e' k a) (d.i₀ *ᵥ z)

theorem **bridge_comp** (D : D7) (g : GA D (D.sp.Y 0)) : (∀ v, uMat ((Formalism.IndefR.RationalEmbedding.iota D g).1 v) = ofHAInf (instOf D) g v) ∧ (instOf D).fmat (Formalism.IndefR.RationalEmbedding.iota D g).2 = ofHAFin (instOf D) g

theorem **simpleSum_eq** (D : D7) {V₀ : HSp D} {ρ : ℕ} (d : HypDec (D.sp.Y 0) V₀ ρ) (E F : Fin ρ → Fin (D.sp.Y 0).rank → D.host.E) (Z : Fin (D.sp.Y 0).rank → D.host.E) (z : Fin V₀.rank → D.host.E) (t lam : D.host.E) (ht : t ≠ 0) (hl : lam ≠ 0) (he : d.e = Matrix.of fun k (a : Fin ρ) => t * lam ^ (ρ - (a : ℕ)) * E a k) (he' : d.e' = Matrix.of fun k (a : Fin ρ) => (star (t * lam ^ (ρ - (a : ℕ))))⁻¹ * F a k) (hz : d.i₀ *ᵥ z = t • Z) (n : GA D (D.sp.Y 0)) : simpleSum d z n = incA D lam⁻¹ * ∑ a, hN n (F a) (nxt E Z a)

theorem **thetaV_hN** (D : D7) (n : GA D (D.sp.Y 0)) (v : InfinitePlace (instOf D).F) (p q : Fin (D.sp.Y 0).rank → D.host.E) : thetaV (instOf D) v (hN n p q) = hJ ((instOf D).ε v) (rv (instOf D) v p) (uMat ((Formalism.IndefR.RationalEmbedding.iota D n).1 v) *ᵥ rv (instOf D) v q)

theorem **tauF_hN** (D : D7) (n : GA D (D.sp.Y 0)) (p q : Fin (D.sp.Y 0).rank → D.host.E) : tauF (instOf D) (hN n p q) = hA (instOf D) (incl (instOf D) p) ((instOf D).fmat (Formalism.IndefR.RationalEmbedding.iota D n).2 *ᵥ incl (instOf D) q)

theorem **psi_inf_trace** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (y : AE D) : ψ (pairA D.host.F (Algebra.trace (AF D.host.F) (AE D) y).1 0) = ∏ v, Complex.exp (2 * Real.pi * Complex.I * -(thetaV (instOf D) v y + star (thetaV (instOf D) v y)))

theorem **psi_fin_trace** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (psiEf : (instOf D).E ⊗[(instOf D).F] Formalism.MainNotions.S.Af (instOf D).F → ℂ) (hadd : ∀ x y, psiEf (x + y) = psiEf x * psiEf y) (hint : ∀ x ∈ IntAf (instOf D).F (instOf D).E, psiEf x = 1) (hrat : ∀ x : (instOf D).E, psiEf (Formalism.UG.inclE (F

theorem **thetaV_incA'** (D : D7) (v : InfinitePlace (instOf D).F) (y : D.host.E) : thetaV (instOf D) v (incA D y) = (instOf D).φ v y

theorem **tauF_incA'** (D : D7) (y : D.host.E) : tauF (instOf D) (incA D y) = Formalism.UG.inclE (F

theorem **thetaV_mulAE** (D : D7) (v : InfinitePlace (instOf D).F) (a b : AE D) : thetaV (instOf D) v (a * b) = thetaV (instOf D) v a * thetaV (instOf D) v b

theorem **thetaV_negAE** (D : D7) (v : InfinitePlace (instOf D).F) (a : AE D) : thetaV (instOf D) v (-a) = -thetaV (instOf D) v a

theorem **thetaV_sumAE** (D : D7) (v : InfinitePlace (instOf D).F) {ρ : ℕ} (x : Fin ρ → AE D) : thetaV (instOf D) v (∑ a, x a) = ∑ a, thetaV (instOf D) v (x a)

theorem **tauF_mulAE** (D : D7) (a b : AE D) : tauF (instOf D) (a * b) = tauF (instOf D) a * tauF (instOf D) b

theorem **tauF_sumAE** (D : D7) {ρ : ℕ} (x : Fin ρ → AE D) : tauF (instOf D) (∑ a, x a) = ∑ a, tauF (instOf D) (x a)

theorem **star_psi_eq** {F : Type} [Field F] [NumberField F] (ψ : AddChar (AF F) ℂ) (hψ : IsCharAF F ψ) (a : AF F) : star (ψ a) = ψ (-a)

theorem **psi_split** {F : Type} [Field F] [NumberField F] (ψ : AddChar (AF F) ℂ) (a : AF F) : ψ a = ψ (pairA F a.1 0) * ψ (pairA F 0 a.2)

theorem **cexp_two_pi_I_neg_inv_nat_smul_add_star** {g : ℕ} (c x : ℂ) (hc : c = ((g : ℂ))⁻¹) : Complex.exp (2 * Real.pi * Complex.I * -(-(c * x) + star (-(c * x)))) = Complex.exp (2 * Real.pi * Complex.I * ((x + star x) / (g : ℂ)))

theorem **star_psiE_div** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (psiEf : (instOf D).E ⊗[(instOf D).F] Formalism.MainNotions.S.Af (instOf D).F → ℂ) (hadd : ∀ x y, psiEf (x + y) = psiEf x * psiEf y) (hint : ∀ x ∈ IntAf (instOf D).F (instOf D).E, psiEf x = 1) (hrat : ∀ x : (instOf D).E, psiEf (Formalism.UG.inclE (F

theorem **sum_add_star_eq_of_eq_fin_sum** {ρ : ℕ} (s : ℂ) (c : Fin ρ → ℂ) (h : s = ∑ a, c a) : s + star s = ∑ a, (c a + star (c a))

theorem **rv_nxt** (I : Formalism.MainNotions.S.AutInst) (v : InfinitePlace I.F) {ρ : ℕ} (e : Fin ρ → Fin I.n → I.E) (z : Fin I.n → I.E) (a : Fin ρ) : rv I v (nxt e z a) = nxt (fun b => rv I v (e b)) (rv I v z) a

theorem **castGA_self'** {I : Formalism.MainNotions.S.AutInst} (e : I = I) (x : I.GA) : castGA e x = x

theorem **core** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (I : Formalism.MainNotions.S.AutInst) (hI : I = instOf D) (ιE : I.E ≃+* D.host.E) (hId : ∀ y, HEq (ιE y) y) (hr : (D.sp.Y 0).rank = I.n) (psiEf : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F → ℂ) (hadd : ∀ x y, psiEf (x + y) = psiEf x * psiEf y) (hint : ∀ x ∈ IntAf I.F I.E, psiEf x = 1) (hrat : ∀ x : I.E, psiEf (Formalism.UG.inclE (F

theorem **GlobalTheory.Core.Mi.tiepsi_sign** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ), IdE X → IsStdChar D ψ (-1) → ∃ lam : D.host.E, (lam = (X.B.g : D.host.E) ∨ lam = -(X.B.g : D.host.E)) ∧ ∀ t : D.host.E, t ≠ 0 → ∀ (V₀ L : HSp D) (d : HypDec (D.sp.Y 0) V₀ D.R.rS) (o : OrthDec V₀ (D.sp.X 0) L) (ζ : Fin L.rank → D.host.E), d.e = GlobalTheory.Core.SBesEx.eM X h t lam → d.e' = GlobalTheory.Core.SBesEx.e'M X h t lam → d.i₀ *ᵥ vecB o ζ = t • GlobalTheory.Core.SBesEx.cY X h (X.zu u) → ∀ n ∈ flagN d, star (psiBes ψ d 1 (vecB o ζ) n) = X.psiInfU u (iotaX X h n).1 * X.psiFinU u (iotaX X h n).2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChLApply

theorem **chL_apply** (x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : chL x = ch x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChOfDil

theorem **chOf_dil** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ j, j ≤ B.R.m → ∀ (v : (I.atT l u a).vT) (s : ((I.atT l u a).Fv v)ˣ) (c : (I.chain l u a).cls j), A.chOf j ((I.fm l u a).dil j v s c) = A.chOf j c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainEachStepSplitPlaceWithoutExcludedParams

theorem **chainNS_holds** : ∀ D : D7, ChainNS D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainEachStepSplitPlaceWithoutExcludedParamsStatement

theorem **_root_.GlobalTheory.T.chain_each_step_splitPlace_without_excludedParams** : ∀ D : D7, ChainNS D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainFirstStageLValuesPositive

theorem **isPar_flat_of_le** {D : D7} {c : Com D} {w : Twist D} {j : ℕ} {s : PiLocalization₂ D} (hr : D.R.r ≤ s.V.rank) (hle : s.CC c ≤ (flat c w j s hr).CC c) (u : s.CC c) (B : GlobalTheory.LFun.Fam D.host.E) (hB : IsPar u B) : IsPar ((u.1, hle u.2) : (flat c w j s hr).CC c) B

theorem **bounded_first_member** (hCRD : GlobalTheory.Published.CuspRapidDecay) {D : D7} {c : Com D} (hc : c.Is) (hj : 0 ≤ D.R.m) {s' : PiLocalization₂ D} (hflat : s'.Is c 0) (u' : s'.CC c) (B : GlobalTheory.LFun.Fam D.host.E) (hB' : IsPar u' B) : GlobalTheory.LFun.Bounded (GlobalTheory.LFun.above D.host.E (c.S : Set (HeightOneSpectrum (𝒪 D.host.F)))) B D.R.r (((D.R.r : ℝ) - 1) / 2)

theorem **conjStable_first_member** (hCRD : GlobalTheory.Published.CuspRapidDecay) {D : D7} {c : Com D} (hc : c.Is) (hj : 0 ≤ D.R.m) {s' : PiLocalization₂ D} (hflat : s'.Is c 0) (u' : s'.CC c) (B : GlobalTheory.LFun.Fam D.host.E) (hB' : IsPar u' B) : ∀ w ∉ GlobalTheory.LFun.above D.host.E (c.S : Set (HeightOneSpectrum (𝒪 D.host.F))), (∀ w' : HeightOneSpectrum (𝒪 D.host.E), w'.under (𝒪 D.host.F) = w.under (𝒪 D.host.F) → w ≠ w' → B w' = (B w).map star) ∧ (¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **lPos_of_rapidDecay** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, LPos D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainFirstStageLValuesPositiveStatement

theorem **_root_.GlobalTheory.T.chain_first_stage_L_values_positive** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, LPos D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainFractionalIdealRelations

theorem **host_ne_zero** (D : D7) : FractionalIdeal.spanSingleton (𝒪 D.host.E)⁰ D.host.xi0 ≠ 0 ∧ D.host.AA ≠ 0 ∧ D.host.AAbar ≠ 0 ∧ dI D ≠ 0

theorem **step** (D : D7) {j : ℕ} (h : j + 1 + 1 ≤ D.R.m) : D.host.AA⁻¹ * (dI D * cI D j) = D.host.AA * cI D (j + 1 + 1) ∧ D.host.AA⁻¹ * (dI D * cI D j) = FractionalIdeal.spanSingleton _ D.host.xi0 * D.host.AAbar * cI D j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainIdealIndexProduct

theorem **local_part** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (V : HSp D) (g : GA D V) : ∃ c : GA D V, SuppAt q.e V c ∧ (mt V c).map q.pr = (mt V g).map q.pr ∧ GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram q.v (g * c⁻¹) = 1

theorem **moves_of_St** {V : HSp D} {β : GL (Fin V.rank) (Lev.EAf D)} {Λ : GlobalTheory.FinPl.Lat V} (hβ : IsFrame β Λ) (hΛ : Λ.FG) {h : GA D V} (hh : finE V h ∈ St β) (P : GlobalTheory.FinPl.Pl D → Prop) : GlobalTheory.FinPl.Moves V h Λ Λ P

theorem **mem_levG_of_inK** {V : HSp D} {β : GL (Fin V.rank) (Lev.EAf D)} {Λ : GlobalTheory.FinPl.Lat V} (hβ : IsFrame β Λ) (hΛ : Λ.FG) (L : LevDat D.host.E ℂ V.rank) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : ∀ PP, PP ∈ L.Pg ∨ PP ∈ L.Q → ¬ P (PP.under (𝒪 D.host.F))) {κ : GA D V} (hκ : GlobalTheory.FinPl.InK V Λ P κ) : κ ∈ levG V β L

theorem **mem_levG_iff_base** {V : HSp D} (β : GL (Fin V.rank) (Lev.EAf D)) {r t j : ℕ} (hr : r ≤ V.rank) (Pg Q Q₀ : Finset (Pl D.host.E)) (P : LevelP Pg r ℂ) (χ : ℕ → ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (ξ ξ₀ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (h₀ : Q₀ = ∅) (h : GA D V) : h ∈ levG V β (datY _ r t j hr Pg Q P χ ξ Frm.B) ↔ h ∈ levG V β (datY _ r t j hr Pg Q₀ P χ ξ₀ Frm.B) ∧ ∀ PP ∈ Q, redFn β PP (finE V h) ∈ blockTri (blk2 r)

theorem **red_iff_local** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {V : HSp D} {Y : Type} [Fintype Y] [DecidableEq Y] (Fr : GlobalTheory.LocMod.Frame k V.rank Y) (J : Subgroup (GL Y k)) {β : GL (Fin V.rank) (Lev.EAf D)} {Λ N : GlobalTheory.FinPl.Lat V} (hβ : IsFrame β Λ) (hΛ : Λ.FG) (L : LevDat D.host.E ℂ V.rank) {PP : Pl D.host.E} (hPP : PP ∈ L.Q) (hv : q.v = PP.under (𝒪 D.host.F)) (hg : ∀ PP' ∈ L.Pg, PP'.under (𝒪 D.host.F) ≠ q.v) (hq : ∀ PP' ∈ L.Q, PP'.under (𝒪 D.host.F) = q.v → PP' = PP) {b : GA D V} (hb : b ∈ GfC V) (hm : GlobalTheory.FinPl.Moves V b Λ N (· = q.v)) (hN : IntIn q V Fr N) (hJ : ∀ x ∈ KO k Y, ∀ c, (ofSplit q).IsLoc Fr c x → (b⁻¹ * c * b ∈ levG V β L ↔ x ∈ J)) {g : GA D V} (hS : finE V (b⁻¹ * g * b) ∈ St β) {x : GL Y k} (hx : (mt V g).map q.pr = Fr.β * x.val * Fr.γ) : redFn β PP (finE V (b⁻

theorem **mem_moved** {V : HSp D} (b : GA D V) (K : Subgroup (GA D V)) (g : GA D V) : g ∈ moved b K ↔ b⁻¹ * g * b ∈ K

theorem **moved_set** {V : HSp D} (b : GA D V) (K : Subgroup (GA D V)) : {g : GfC V | b * g.1 * b⁻¹ ∈ moved b K} = {k : GfC V | k.1 ∈ (K : Set (GA D V))}

theorem **moved_level** {V : HSp D} {β : GL (Fin V.rank) (Lev.EAf D)} {Λ N : GlobalTheory.FinPl.Lat V} (hβ : IsFrame β Λ) (hΛ : Λ.FG) (L : LevDat D.host.E ℂ V.rank) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : ∀ PP, PP ∈ L.Pg ∨ PP ∈ L.Q → ¬ P (PP.under (𝒪 D.host.F))) {b : GA D V} (hb : b ∈ GfC V) (hm : GlobalTheory.FinPl.Moves V b Λ N P) : moved b (levG V β L) ≤ GfC V ∧ IsCompact (moved b (levG V β L) : Set (GA D V)) ∧ IsOpen (((moved b (levG V β L)).subgroupOf (GfC V) : Subgroup (GfC V)) : Set (GfC V)) ∧ (∀ κ, GlobalTheory.FinPl.InK V N P κ → κ ∈ moved b (levG V β L)) ∧ ∀ g ∈ moved b (levG V β L), GlobalTheory.FinPl.Moves V g N N P

theorem **exists_mat** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {V : HSp D} {Y : Type} [Fintype Y] [DecidableEq Y] (Fr : GlobalTheory.LocMod.Frame k V.rank Y) (g : GA D V) : ∃ x : GL Y k, (mt V g).map q.pr = Fr.β * x.val * Fr.γ

theorem **under_ne** {T : Type} {L : LevCh D T} {t₀ : T} (h : L.Is t₀) (t : T) {PP PP' : Pl D.host.E} (hQ : PP ∈ L.Q t) (hg : PP' ∈ L.Pg) : PP'.under (𝒪 D.host.F) ≠ PP.under (𝒪 D.host.F)

theorem **base_empty** {T : Type} {L : LevCh D T} {t₀ : T} (h : L.Is t₀) : L.Q t₀ = ∅

theorem **cI_mul_idx** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ t (w : VTup (f.t t)) (F : GlobalTheory.QPl.QFam v (f.t t)), F.Pins w → QLev f F → ∀ j, j < D.R.m → (f.t t).cI j * ∏ i, ((F.Q i).idx (j + 1) : ℂ) = (f.t W.P.A3.A.μ0).cI j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainIdealIndexProductStatement

theorem **_root_.GlobalTheory.T.chain_ideal_index_product** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ t (w : VTup (f.t t)) (F : GlobalTheory.QPl.QFam v (f.t t)), F.Pins w → QLev f F → ∀ j, j < D.R.m → (f.t t).cI j * ∏ i, ((F.Q i).idx (j + 1) : ℂ) = (f.t W.P.A3.A.μ0).cI j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainKernelBoundAtEmbeddingAnyLevelPrime

theorem **kernel_le_sharp** (hB : GlobalTheory.IndefiniteRowSideFullHecke.PaddedLastEntry.BlockBound) (hT : FamilyTie) (hV : FamilyVectors) (hL : PeriodLattices) (hR : CosetRefinement) (hP : GlobalTheory.IndefiniteRowSideFullHecke.MeanPeriodLogarithms.MeanPeriod) (C : ℕ) : ∃ CK : ℝ, 0 ≤ CK ∧ ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ B : Base D D.host.ccK dl, FieldOfDefinition W sel mu cst u (ambOf s B χ cD) → ∀ i j, j = i + 1 → i < N.R.m →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainKernelBoundAtEmbeddingAnyLevelPrimeStatement

theorem **_root_.GlobalTheory.T.chainKernel_bound_at_embedding_anyLevelPrime** : GlobalTheory.IndefiniteRowSideFullHecke.PaddedLastEntry.BlockBound → FamilyTie → FamilyVectors → PeriodLattices → CosetRefinement → GlobalTheory.IndefiniteRowSideFullHecke.MeanPeriodLogarithms.MeanPeriod → ∀ C : ℕ, ∃ CK : ℝ, 0 ≤ CK ∧ ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ B : Base D D.host.ccK dl, FieldOfDefinition W sel mu cst u (ambOf s B χ cD) → ∀ i j, j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainKernelPreservesRamifiedFormsLiveCharts

theorem **kernels_keep_ramL** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch₀ Ch : ℕ → Type) (A : TwDat I l Z u a Ch₀) (ι : ∀ j, Ch j → Ch₀ j), A.Is → A.Live ι → ∀ i, i < B.R.m → ∀ φ, A.RamL ι i φ → A.RamL ι (i + 1) ((I.chain l u a).K i φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainKernelPreservesRamifiedFormsLiveChartsStatement

theorem **_root_.GlobalTheory.T.chainKernel_preserves_ramifiedForms_liveCharts** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch₀ Ch : ℕ → Type) (A : TwDat I l Z u a Ch₀) (ι : ∀ j, Ch j → Ch₀ j), A.Is → A.Live ι → ∀ i, i < B.R.m → ∀ φ, A.RamL ι i φ → A.RamL ι (i + 1) ((I.chain l u a).K i φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainLevelIntegralRealZetaOfTwistedTransformLaws

theorem **pet_tr** {c : Com D} {s : PiLocalization₂ D} (hc : c.Is) {ipD : SmT s.M →ₗ[ℂ] SmT s.M →ₗ*[ℂ] ℂ} {ip : s.CC c →ₗ[ℂ] s.CC c →ₗ*[ℂ] ℂ} (hM : IsMet c s ipD ip) (u₁ u₂ : s.CC c) (v : s.M.Vec) (b : GinfC s.V) (κ : GA D s.V) (hκ : κ ∈ Lev.levG s.V s.β s.L) : s.pet (rTr s.V ((b : GA D s.V) ∗ κ) (PiLocalization₂.val u₁ (PiTensorProduct.tprod ℂ v))) (PiLocalization₂.val u₂ (PiTensorProduct.tprod ℂ v)) = Lev.levChi s.V s.β s.L κ * ip u₁ u₂ * ∏ τ, ⟪(v τ : s.M.H τ), s.M.rep τ (infMat D.sp.cm.sig s.V.gram (c.θ τ) (b : GA D s.V)) (v τ : s.M.H τ)⟫_ℂ

theorem **leftInv_conj** {V : HSp D} {F : GA D V → ℂ} (h : GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ V) F) : GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ V) (fun x => conj (F x))

theorem **rcoef_conj** {s : PiLocalization₂ D} (hh : s.ν.IsHaarMeasure) (hf : IsFundamentalDomain (GQ s.V) s.fd s.ν) {F₁ F₂ : GA D s.V → ℂ} (h₁ : GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ s.V) F₁) (h₂ : GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ s.V) F₂) (g : GA D s.V) : GlobalTheory.Core.Rk.IP.rcoef s.ν (fun x => conj (F₁ x)) (fun x => conj (F₂ x)) g = conj (s.pet (rTr s.V g F₁) F₂)

theorem **leftInv_val** {c : Com D} {s : PiLocalization₂ D} (u : s.CC c) (v : s.M.Vec) : GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ s.V) (PiLocalization₂.val u (PiTensorProduct.tprod ℂ v))

theorem **hgtL_real_lev** {s : PiLocalization₂ D} (hsd : IsSelfDualC s.V s.Λ) {a : GL (Fin s.V.rank) (AE D)} (hmaj : GlobalTheory.Doubling.IsMaj s.V a) (b : GinfC s.V) (κ : GA D s.V) (hκf : κ ∈ GfC s.V) (hpr : ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig s.V.gram v κ ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig s.V.gram s.Λ v) : GlobalTheory.Core.Rk.IP.hgtL s.V s.Λ a ((b : GA D s.V) ∗ κ) = GlobalTheory.Core.Rk.IP.hgtL s.V s.Λ a (b : GA D s.V) ∧ ((GlobalTheory.Core.Rk.IP.hgtL s.V s.Λ a (b : GA D s.V) : ℝ)) = ∏ w, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW s.V a w (GlobalTheory.AdelicU.projArch D.sp.cm.sig s.V.gram w (b : GA D s.V))

theorem **flatInt_mul_lev** {c : Com D} {s s' : PiLocalization₂ D} (hc : c.Is) (hh : s.ν.IsHaarMeasure) (hf : IsFundamentalDomain (GQ s.V) s.fd s.ν) (hsd : IsSelfDualC s.V s.Λ) {ipD : SmT s.M →ₗ[ℂ] SmT s.M →ₗ*[ℂ] ℂ} {ip : s.CC c →ₗ[ℂ] s.CC c →ₗ*[ℂ] ℂ} (hM : IsMet c s ipD ip) (p : Step c s s') (m : Measure (WO s.V s'.V)) (X : Set (WO s.V s'.V)) {a : GL (Fin s.V.rank) (AE D)} (hmaj : GlobalTheory.Doubling.IsMaj s.V a) (φ : PolF D s'.V.rank s.V.rank) (v : s.M.Vec) (hlaw : ∀ κ ∈ Lev.levG s.V s.β s.L, (p.ω (κ, 1)) (p.pt φ) = Lev.levChi s.V s.β s.L κ • p.pt φ) (hstab : ∀ κ ∈ Lev.levG s.V s.β s.L, ‖Lev.levChi s.V s.β s.L κ‖ = 1 ∧ ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig s.V.gram v κ ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig s.V.gram s.Λ v) (u₁ u₂ : s.CC c) (z : ℂ) (b : GinfC s.V) (κ : GA D s.V) (hκ : κ ∈ Lev.levG s.V s.β s.L) : flatInt p m X a φ (PiLocalization₂.val u₁ (PiTensorProduct.tprod ℂ v

theorem **levInt** : ∀ D : D7, LevelLawQ D → LevInt D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainLevelIntegralRealZetaOfTwistedTransformLawsStatement

theorem **_root_.GlobalTheory.T.chainLevelIntegralRealZeta_of_twisted_transform_laws** : ∀ D : D7, LevelLawQ D → LevInt D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainMoveCharactersEqRhoUpToBase

theorem **chr_root_mul_base** {B : Base} {X : PreX B.toPre} {I : TC B X} (hI : I.Ok) {l : I.Laws} {Z : TC.Later B (I.cols l)} {u : B.Word} {a : I.ι} {Ch : ℕ → Type} {A : TwDat I l Z u a Ch} (hA : A.Is) (j : Fin B.R.m) (k : ℕ) (hk : k ≤ B.R.m) (β : Ch k) (c : {c // A.chOf k c = β}) (x : (AE (I.rec7 l Z))ˣ) (hx : IsDetRoot _ (A.h k β c) x) : (A.chr ((j : ℕ) + 1) x : ℂ) * (A.chr ((j : ℕ) + 1) (A.xr k β) : ℂ) = ((tildeQ B.host I.SQ (I.C.mu0 j * I.xi a) (detPt B.host I.SQ ((I.stp u a k).rep c.1).1.1) : ℂˣ) : ℂ)

theorem **rho_eq** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i + 1 ≤ B.R.m) (c : (I.stp u a i).Cl) : (I.chIn u a).rho i c = ((tildeQ B.host I.SQ (I.C.mu0 (i, hi) * I.xi a) (detPt B.host I.SQ ((I.stp u a i).rep c).1.1) : ℂˣ) : ℂ)⁻¹ * chiId I.C.chiPsi (Th.detId B.host (Th.stdL B.host (B.R.N i)) ((I.stp u a i).rep c).1.1.deco.1.N) ^ (-((B.R.N (i + 1) / 2 : ℕ) : ℤ))

theorem **rho'_eq** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i + 1 ≤ B.R.m) (c' : (I.stp u a (i + 1)).Cl) : (I.chIn u a).rho' i c' = ((tildeQ B.host I.SQ (I.C.mu0 (i, hi) * I.xi a) (detPt B.host I.SQ ((I.stp u a (i + 1)).rep c').1.1) : ℂˣ) : ℂ) * chiId I.C.chiPsi (Th.detId B.host (Th.stdL B.host (B.R.N (i + 1))) ((I.stp u a (i + 1)).rep c').1.1.deco.1.N) ^ ((B.R.N i / 2 : ℕ) : ℤ)

theorem **moves_alg** (X Y r₀ r₁ a b : ℂ) (p q : ℤ) (hX : X ≠ 0) (hY : Y ≠ 0) (h₀ : r₀ ≠ 0) (h₁ : r₁ ≠ 0) (ha : a ≠ 0) (hb : b ≠ 0) : X⁻¹ * Y = r₀ * r₁⁻¹ * ((X * r₀)⁻¹ * a ^ (-p) * (Y * r₁ * b ^ q)) * (a ^ p * b ^ (-q))

theorem **chr_moves_eq_rho_base** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (c : {c // A.chOf i c = α}) (c' : {c // A.chOf (i + 1) c = γ}) x y, IsDetRoot _ (A.h i α c) x → IsDetRoot _ (A.h (i + 1) γ c') y → (A.chr (i + 1) x : ℂ)⁻¹ * (A.chr (i + 1) y : ℂ) = (A.chr (i + 1) (A.xr i α) : ℂ) * (A.chr (i + 1) (A.xr (i + 1) γ) : ℂ)⁻¹ * ((I.chIn u a).rho i c.1 * (I.chIn u a).rho' i c'.1) * I.A.chiF (B.R.N i) (B.R.N (i + 1)) ((I.stp u a i).rep c.1).1.1.deco.1.N ((I.stp u a (i + 1)).rep c'.1).1.1.deco.1.N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainMoveCharactersEqRhoUpToBaseStatement

theorem **_root_.GlobalTheory.T.chainMove_characters_eq_rho_up_to_base** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (c : {c // A.chOf i c = α}) (c' : {c // A.chOf (i + 1) c = γ}) x y, IsDetRoot _ (A.h i α c) x → IsDetRoot _ (A.h (i + 1) γ c') y → (A.chr (i + 1) x : ℂ)⁻¹ * (A.chr (i + 1) y : ℂ) = (A.chr (i + 1) (A.xr i α) : ℂ) * (A.chr (i + 1) (A.xr (i + 1) γ) : ℂ)⁻¹ * ((I.chIn u a).rho i c.1 * (I.chIn u a).rho' i c'.1) * I.A.chiF (B.R.N i) (B.R.N (i + 1)) ((I.stp u a i).rep c.1).1.1.deco.1.N ((I.stp u a (i + 1)).rep c'.1).1.1.deco.1.N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainProjectionLevelCharacter

theorem **multT_val** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (θ : InfFam D) (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) (u : SmT M →ₗ[ℂ] (GA D V → ℂ)) (hu : u ∈ MultT θ M J χ) (v : SmT M) : (∀ γ ∈ GQ V, ∀ g, u v (γ ∗ g) = u v g) ∧ ∀ k ∈ J, ∀ g, u v (g ∗ k) = χ k ∗ u v g

theorem **io_levChi** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t j, j + 1 < D.R.m → ∀ F : (pk W).CC t (j + 1), F.1 ∈ CP (pk W) (lvP W) t (j + 1) → ∀ κ ∈ levS (Yc D j) ((pk W).βY j) ((pk W).datYt t j (lvP W)), ∀ φ x g, ((pk W).ioY t j F).1 ((pk W).ItY j φ x) (g ∗ κ) = levChi (Yc D j) ((pk W).βY j) ((pk W).datYt t j (lvP W)) κ ∗ ((pk W).ioY t j F).1 ((pk W).ItY j φ x) g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainProjectionLevelCharacterStatement

theorem **_root_.GlobalTheory.T.chain_projection_level_character** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t j, j + 1 < D.R.m → ∀ F : (pk W).CC t (j + 1), F.1 ∈ CP (pk W) (lvP W) t (j + 1) → ∀ κ ∈ levS (Yc D j) ((pk W).βY j) ((pk W).datYt t j (lvP W)), ∀ φ x g, ((pk W).ioY t j F).1 ((pk W).ItY j φ x) (g ∗ κ) = levChi (Yc D j) ((pk W).βY j) ((pk W).datYt t j (lvP W)) κ ∗ ((pk W).ioY t j F).1 ((pk W).ItY j φ x) g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStageFormsTwistedLevelLeFinerLevel

theorem **prod_diagonal_block_dets_eq_lower_block_det** {κ : Type} [CommRing κ] {r t j n : ℕ} (hn : n = r + j ∗ t) {g : GL (Fin n) κ} (hg : g ∈ blockTri (blkOf r t)) : ∏ c ∈ Finset.range j, dBlk (blkOf r t) (c + 1) g = dBlk (blk2 r) 1 g

theorem **multT_le** {V : HSp D} {S : D.what₂.Tau → SPar} (θ : InfFam D) (M : DSModel V S) {J J' : Set (GA D V)} {χ χ' : GA D V → ℂ} (h : ∀ k ∈ J', k ∈ J ∧ χ k = χ' k) : MultT θ M J χ ≤ MultT θ M J' χ'

theorem **lev_flat_le** (c : Com D) (w : Twist D) (hw : w.Is c) (j : ℕ) {n : ℕ} (hn : n = D.R.r + j ∗ D.R.t) (J : Matrix (Fin n) (Fin n) D.host.E) (β : GL (Fin n) (Lev.EAf D)) (hr hr' : D.R.r ≤ n) (x : GL (Fin n) (Lev.EAf D)) (hx : x ∈ lev D.sp.cm J β (Lev.datY n D.R.r D.R.t j hr c.Pg ∅ c.P c.chi w.ξ Lev.Frm.B)) : x ∈ lev D.sp.cm J β (Lev.datY n D.R.r D.R.t j hr' w.Pg w.Q (LevelP.triv _ _ _) w.chi w.ξ Lev.Frm.B) ∧ levFn β (Lev.datY n D.R.r D.R.t j hr c.Pg ∅ c.P c.chi w.ξ Lev.Frm.B) x = levFn β (Lev.datY n D.R.r D.R.t j hr' w.Pg w.Q (LevelP.triv _ _ _) w.chi w.ξ Lev.Frm.B) x

theorem **flatLe** : ∀ D : D7, FlatLe D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStageFormsTwistedLevelLeFinerLevelStatement

theorem **_root_.GlobalTheory.T.chain_stage_forms_twisted_level_le_finer_level** : ∀ D : D7, FlatLe D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStepKernelPreservesLevelRamifiedForms

theorem **tr_one** {E K ι : Type} [Field E] [NumberField E] [Field K] [Fintype ι] {n : ℕ} (S : GStep E K n) (emb : ι → E →+* K) (σ : K →+* ℂ) (b : S.β) (F : Fr emb σ n → ℂ) : S.tr emb σ b 1 F = F

theorem **ent_dia** (I : TC B X) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) {b : (I.stp u a i).β} {b' : (I.stp u a (i + 1)).β} (p : (I.stp u a i).D b) (p' : (I.stp u a (i + 1)).D b') (P : (I.atIn u a).ST) (s : (Res P.1)ˣ) : ent I l u a i (((I.atIn u a).A i).dia b P s p) (((I.atIn u a).A (i + 1)).dia b' P s p') = ent I l u a i p p'

theorem **Kmap_stab** (I : TC B X) (l : I.Laws) (hI : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i < B.R.m) (c : (I.stp u a i).Cl) (c' : (I.stp u a (i + 1)).Cl) (g : (I.stp u a (i + 1)).Stab c') (f : Fr B.fields.emb B.embLL (B.R.N i) → ℂ) (hf : f ∈ ValC (Gr B.fields.emb B.embLL (B.R.N i)) (lamT (B.hcW u) i)) : (I.stp u a (i + 1)).tr B.fields.emb B.embLL c'.1 g.1 ((I.chain l u a).Kmap i c c' f) = (I.chain l u a).Kmap i c c' f

theorem **K_mem_Fi** (I : TC B X) (l : I.Laws) (hI : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i < B.R.m) : ∀ φ ∈ (I.stp u a i).Fi B.fields.emb B.embLL (lamT (B.hcW u) i), (I.chain l u a).K i φ ∈ (I.stp u a (i + 1)).Fi B.fields.emb B.embLL (lamT (B.hcW u) (i + 1))

theorem **entry_dia** (I : TC B X) (l : I.Laws) (hI : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i < B.R.m) (φ : ColT I l u a i) (hφ : φ ∈ (I.fm l u a).Fi i) (v : (I.atT l u a).vT) (s : ((I.atT l u a).Fv v)ˣ) (L : (I.chain l u a).cls i) (c' : (I.chain l u a).cls (i + 1)) : (I.fm l u a).dilT (i + 1) v s c' ((I.chain l u a).Kmap i ((I.fm l u a).dil i v s L) ((I.fm l u a).dil (i + 1) v s c') (φ ((I.fm l u a).dil i v s L))) = (I.chain l u a).Kmap i L c' ((I.fm l u a).dilT i v s L (φ ((I.fm l u a).dil i v s L)))

theorem **dia_K_eq** (I : TC B X) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (v : (I.atT l u a).vT) (s : ((I.atT l u a).Fv v)ˣ) (φ : ColT I l u a i) (c' : (I.chain l u a).cls (i + 1)) : (I.fm l u a).dilT (i + 1) v s c' ((I.chain l u a).K i φ ((I.fm l u a).dil (i + 1) v s c')) = ∑ L ∈ @Finset.univ _ ((I.chain l u a).clsFin i), (I.chain l u a).wt i L • (I.fm l u a).dilT (i + 1) v s c' ((I.chain l u a).Kmap i ((I.fm l u a).dil i v s L) ((I.fm l u a).dil (i + 1) v s c') (φ ((I.fm l u a).dil i v s L)))

theorem **kernel_keeps_ramT** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (i : ℕ), i < B.R.m → ∀ φ : ColT I l u a i, RamT i φ → RamT (i + 1) ((I.chain l u a).K i φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStepKernelPreservesLevelRamifiedFormsStatement

theorem **_root_.GlobalTheory.T.chainStepKernel_preserves_levelRamifiedForms** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (i : ℕ), i < B.R.m → ∀ φ : ColT I l u a i, RamT i φ → RamT (i + 1) ((I.chain l u a).K i φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStepScalarsOfRealisation

theorem **sesq_eq_of_tprod** {ι : Type} {s : ι → Type} [∀ i, AddCommGroup (s i)] [∀ i, Module ℂ (s i)] (f g : (⊗[ℂ] i, s i) →ₗ[ℂ] (⊗[ℂ] i, s i) →ₗ⋆[ℂ] ℂ) (h : ∀ v v' : ∀ i, s i, f (PiTensorProduct.tprod ℂ v) (PiTensorProduct.tprod ℂ v') = g (PiTensorProduct.tprod ℂ v) (PiTensorProduct.tprod ℂ v')) : f = g

theorem **adjVec_unique** {c : Com D} {s s' : PiLocalization₂ D} {p : Step c s s'} {ipD : SmT s.M →ₗ[ℂ] SmT s.M →ₗ⋆[ℂ] ℂ} {ipD' : SmT s'.M →ₗ[ℂ] SmT s'.M →ₗ⋆[ℂ] ℂ} {ip' : s'.CC c →ₗ[ℂ] s'.CC c →ₗ⋆[ℂ] ℂ} (hm : IsMet c s' ipD' ip') {φ : PolF D s'.V.rank s.V.rank} {v : SmT s.M} {w₀ w₁ : SmT s'.M} (h₀ : IsAdjVec p ipD ipD' φ v w₀) (h₁ : IsAdjVec p ipD ipD' φ v w₁) : w₀ = w₁

theorem **eq_zero_of_ip** {c : Com D} {s : PiLocalization₂ D} {ipD : SmT s.M →ₗ[ℂ] SmT s.M →ₗ⋆[ℂ] ℂ} {ip : s.CC c →ₗ[ℂ] s.CC c →ₗ⋆[ℂ] ℂ} (hm : IsMet c s ipD ip) {x : s.CC c} (h : ∀ y, ip y x = 0) : x = 0

theorem **isUp_iff** (R : RealDat D A Ks) (hIP : IPDict R) {t : T} {k : ℕ} (hk : k + 1 < D.R.m) {ipD ip ipD' ip'} (hm : IsMet (comR R t) (stY R t k) ipD ip) (hm' : IsMet (comR R t) (stY R t (k + 1)) ipD' ip') (U : (stY R t k).CC (comR R t)) (X : (stY R t (k + 1)).CC (comR R t)) : IsUp R t k (PiTensorProduct.lift.symm U.1) (PiTensorProduct.lift.symm X.1) ↔ ∀ y, ip' y X = ip ((stepY R t k).io y) U

theorem **isUpTop_iff** (R : RealDat D A Ks) (hIP : IPDict R) (h : RangeX R) (hm2 : 2 ≤ D.R.m) {t : T} {ipD ip} (hm : IsMet (comR R t) (stY R t (D.R.m − 1)) ipD ip) (U : (stY R t (D.R.m − 1)).CC (comR R t)) (x : (R.HX t).H) : IsUpTop R t (PiTensorProduct.lift.symm U.1) x ↔ ∀ y, ⟪x, y⟫_ℂ = ip ((stepX R h t).io (toCC R h t y)) U

theorem **reached_good** (R : RealDat D A Ks) (hN : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hSoft : SoftQ D) (hQ : QOfVal D) (hSL : StepLam D) (h : RangeX R) (hRC : RChain R h) (hRU : RUnif R h) (hRP : RowPar R) (hIP : IPDict R) (t : T) (c : EigSys D) (B : GlobalTheory.LFun.Fam D.host.E) (hB : GlobalTheory.LFun.IsSysParamD D (badPl R t) D.R.r c B) (hb : GlobalTheory.LFun.Bounded (GlobalTheory.LFun.above D.host.E (badPl R t : Set (HeightOneSpectrum (𝒪 D.host.F)))) B D.R.r ((D.R.r : ℝ) / 2)) : ∀ k < D.R.m, (∀ u ∈ reached R t c k, GoodAt R t B k u) ∧ (rowPart R (badPl R t) c ≠ ⊥ → ∃ u ∈ reached R t c k, u ≠ 0)

theorem **conj_step** {P Q K d : ℂ} {l : ℝ} (hd : conj d = d) (hd0 : d ≠ 0) (hf : P * d = conj (K * l) * Q) : conj P = K / d * l * conj Q

theorem **pos_ratio** {a b d K : ℂ} {l : ℝ} (ha : conj a = a) (ha0 : 0 < a.re) (hb : conj b = b) (hb0 : 0 < b.re) (hd : conj d = d) (hd0 : 0 < d.re) (hl : 0 < l) (hf : a * d = conj (K * l) * b) : ∃ q : ℝ, 0 < q ∧ (q : ℂ) = K / d

theorem **mid_scalar** (R : RealDat D A Ks) (hN : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hSoft : SoftQ D) (hQ : QOfVal D) (hLP : LPos D) (hLF : LFull D) (hSL : StepLam D) (h : RangeX R) (hRC : RChain R h) (hRU : RUnif R h) (hRP : RowPar R) (hIP : IPDict R) (k : ℕ) (hk1 : k + 1 < D.R.m) : ∃ r : ℝ, 0 < r ∧ ∀ (t : T) (c : EigSys D), rowPart R (badPl R t) c ≠ ⊥ → ∀ B, GlobalTheory.LFun.IsSysParamD D (badPl R t) D.R.r c B → ∃ l, IsLValue R t B (k + 1) l ∧ StepScalar R t c k (r * l)

theorem **top_key** (R : RealDat D A Ks) (hN : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hSoft : SoftQ D) (hQ : QOfVal D) (hLP : LPos D) (hLF : LFull D) (hSL : StepLam D) (h : RangeX R) (hRC : RChain R h) (hRU : RUnif R h) (hRP : RowPar R) (hIP : IPDict R) (δ : ℝ) (hδ : 0 < δ) (hNXt : ∀ t, Function.Bijective (toCC R h t) ∧ ∀ ipD ip, IsMet (comR R t) (stX R t) ipD ip → ∀ x y : (R.HX t).H, ⟪x, y⟫_ℂ = (δ : ℂ) * ip (toCC R h t y) (toCC R h t x)) (K : ℂ) (φ : PolF D D.sp.X'.rank (Yc D (D.R.m − 1)).rank) (v : (R.MY (D.R.m − 1)).Vec) (hKU : IsUnif R (D.R.m − 1) (fun t => stY R t (D.R.m − 1)) (fun t => stX R t) (fun t => stepX R h t) K (fun _ => φ) fun _ => v) : ∀ (ipD₀ : SmT (R.MY (D.R.m − 1)) →ₗ[ℂ] SmT (R.MY (D.R.m − 1)) →ₗ*[ℂ] ℂ) (ipD₀' : SmT R.MX →ₗ[ℂ] SmT R.MX →ₗ*[ℂ] ℂ), (∀ v v' : (R.MY (D.R.m − 1)).Vec, ipD₀ (PiTensorProduct.tprod ℂ v) (PiTensorProduct.tprod ℂ v') =

theorem **top_scalar** (R : RealDat D A Ks) (hN : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D) (hSoft : SoftQ D) (hQ : QOfVal D) (hLP : LPos D) (hLF : LFull D) (hSL : StepLam D) (h : RangeX R) (hRC : RChain R h) (hNX : NormX R h) (hRU : RUnif R h) (hRP : RowPar R) (hIP : IPDict R) : ∃ r : ℝ, 0 < r ∧ ∀ (t : T) (c : EigSys D), rowPart R (badPl R t) c ≠ ⊥ → ∀ B, GlobalTheory.LFun.IsSysParamD D (badPl R t) D.R.r c B → ∃ l, IsLValue R t B D.R.m l ∧ TopScalar R t c (r * l)

theorem **scalars_of_realisation** : ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), Given R F → GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D → SoftQ D → QOfVal D → LPos D → LFull D → StepLam D → ∀ h : RangeX R, RChain R h → NormX R h → RUnif R h → RowPar R → IPDict R → GlobalTheory.IndefiniteRowSideFullHecke.LabelSettingScalars.Scalars R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStepScalarsOfRealisationStatement

theorem **_root_.GlobalTheory.T.chainStep_scalars_of_realisation** : ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), Given R F → GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D → SoftQ D → QOfVal D → LPos D → LFull D → StepLam D → ∀ h : RangeX R, RChain R h → NormX R h → RUnif R h → RowPar R → IPDict R → GlobalTheory.IndefiniteRowSideFullHecke.LabelSettingScalars.Scalars R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStepTwistedNormSqEqUntwisted

theorem **level_trivial_eq_ofLocal** {E : Type} [Field E] [NumberField E] (Pg : Finset (Pl E)) (r : ℕ) : LevelP.triv Pg r ℂ = LevelP.ofLocal (fun _ => ⊥) (fun _ => 1)

theorem **norm_sq_eq_inner_product** {D : D7} {S T : HSp D} (μ : Measure (WO S T)) (ℱ : Set (WO S T)) (g : WO S T → ℂ) : ((nrm2 μ ℱ g : ℝ) : ℂ) = ipO μ ℱ g g

theorem **chainStep_twisted_normSq_eq_untwisted_mul_localFactors** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (f₀ : WO S T → ℂ) (m : Measure (WO S T)) (X : Set (WO S T)), IsBase ψ θ S T sS sT ΛS ΛT f₀ → IsCovolOne (ratL T S) m X → Formalism.L2.DIV.IsFrame βS ΛS → Formalism.L2.DIV.IsFrame βT ΛT → ∀ (r t j t' : ℕ) (hr : r ≤ S.rank), S.rank + t' = T.rank → ∀ (Pg Q : Finset (Pl D.host.E)), Disjoint Pg Q → (∀ PP ∈ Q, ((D.host.QS * D.host.q * Ideal.absNorm D.host.ccK : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal) → ∀ (χ : ℕ → ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (χj ξ ξ' : ∀ PP : Pl D.host.E, (kP PP

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainStepTwistedNormSqEqUntwistedStatement

theorem **_root_.GlobalTheory.T.nrm2_twisted_eq_untwisted_mul_prod_zinv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (f₀ : WO S T → ℂ) (m : Measure (WO S T)) (X : Set (WO S T)), IsBase ψ θ S T sS sT ΛS ΛT f₀ → IsCovolOne (ratL T S) m X → Formalism.L2.DIV.IsFrame βS ΛS → Formalism.L2.DIV.IsFrame βT ΛT → ∀ (r t j t' : ℕ) (hr : r ≤ S.rank), S.rank + t' = T.rank → ∀ (Pg Q : Finset (Pl D.host.E)), Disjoint Pg Q → (∀ PP ∈ Q, ((D.host.QS * D.host.q * Ideal.absNorm D.host.ccK : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal) → ∀ (χ : ℕ → ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (χj ξ ξ' : ∀ PP : Pl D.host.E,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftUpEqAdjointValue

theorem **upVal_of_span_adj** : ∀ D : D7, SoftQ D → UpSpanQ D → UpAdjQ D → UpVal D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftUpEqAdjointValueStatement

theorem **_root_.GlobalTheory.T.chain_theta_lift_up_eq_adjoint_value** : ∀ D : D7, SoftQ D → UpSpanQ D → UpAdjQ D → UpVal D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftUpLowerBlockParameterProof

theorem **isGen_of_injective** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A B₀ : Type} [Fintype A] [DecidableEq A] [Fintype B₀] [DecidableEq B₀] {X Y : Type} [AddCommGroup X] [Module ℂ X] [AddCommGroup Y] [Module ℂ Y] {ρ : Representation ℂ (GL (A ⊕ B₀) k) X} {ρ' : Representation ℂ (GL (A ⊕ B₀) k) Y} (f : X →ₗ[ℂ] Y) (hf : ∀ g y, f (ρ g y) = ρ' g (f y)) (hi : Function.Injective f) {ξ : kˣ →* ℂˣ} {ϖ : kˣ} {B : Multiset ℂˣ} {x : X} (h : GlobalTheory.ParHecke.IsGen ρ' ξ ϖ B (f x)) : GlobalTheory.ParHecke.IsGen ρ ξ ϖ B x

theorem **seg_eq_segL** {D : D7} {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) (PP : Pl D.host.E) (hP : IsProjAt φ PP) {ϖ : (Fv D.host.F v)ˣ} (hϖ : valuation (Fv D.host.F v) (ϖ : Fv D.host.F v) = uniformizer (Fv D.host.F v)) (z : ℂˣ) (t : ℕ) : GlobalTheory.LFun.seg PP z t = GlobalTheory.ParHecke.segL ϖ z t

theorem **isUnram_of_lifts** {D : D7} {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) (PP : Pl D.host.E) (hP : IsProjAt φ PP) (a : (kP PP)ˣ →* ℂˣ) (μ μ' : (Fv D.host.F v)ˣ →* ℂˣ) (h : LiftsChar φ PP μ a) (h' : LiftsChar φ PP μ' a) : GlobalTheory.SplitWeil.IsUnram (μ * μ'⁻¹)

theorem **GlobalTheory.ChainTw.upQ_holds** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, UpQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftUpLowerBlockParameterProofStatement

theorem **_root_.GlobalTheory.T.chainThetaLiftUpLowerBlockParameter_proof** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, UpQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftUpParameterOfTransfer

theorem **upParQ_of_transfer** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Hk3.HoweUnram → ∀ D : D7, UpParQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftUpParameterOfTransferStatement

theorem **_root_.GlobalTheory.T.chainThetaLiftUp_parameter_of_transfer** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Hk3.HoweUnram → ∀ D : D7, UpParQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftsAdjointOfCuspRapidDecay

theorem **upAdjQ_of_cuspRapidDecay** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, UpAdjQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainThetaLiftsAdjointOfCuspRapidDecayStatement

theorem **_root_.GlobalTheory.T.chainThetaLiftsAdjoint_of_cuspRapidDecay** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, UpAdjQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainTwistEulerFactorsRestoreProof

theorem **restore_holds** : ∀ D : D7, Restore D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainTwistEulerFactorsRestoreProofStatement

theorem **_root_.GlobalTheory.T.chainTwistEulerFactorsRestore_proof** : ∀ D : D7, Restore D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainTwistPlacesIntegralPeelOfDoubling

theorem **qPeel_holds** : GlobalTheory.Published.UnramifiedDoublingGL → GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, QPeel D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainTwistPlacesIntegralPeelOfDoublingStatement

theorem **_root_.GlobalTheory.T.chainTwistPlacesIntegralPeel_of_doubling** : GlobalTheory.Published.UnramifiedDoublingGL → GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, QPeel D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainTwistPlacesSplitUnramifiedQuotients

theorem **absNorm_under_of_ne** {w w' : HeightOneSpectrum (𝒪 D.host.E)} (h : w.under (𝒪 D.host.F) = w'.under (𝒪 D.host.F)) (hne : w ≠ w') : Ideal.absNorm (w.under (𝒪 D.host.F)).asIdeal = Ideal.absNorm w.asIdeal

theorem **conjPrime_ne_of_mem** {P : Finset (Pl D.host.E)} {e : AE D} (h : IsIdemSum e P) {PP : Pl D.host.E} (hPP : PP ∈ P) : GlobalTheory.IdemSupport.cP PP ≠ PP

theorem **exists_fin_of_mem** {m i : ℕ} (hi : i ∈ Finset.Icc 1 m) : ∃ k : Fin m, (k : ℕ) + 1 = i

theorem **GlobalTheory.ChainTw.twist_places** : ∀ (D : D7) (c : Com D) (w : Twist D), w.Is c → (∀ v, v ∈ c.S ↔ v ∈ w.S₀ ∨ ∃ PP ∈ w.Q, PP.under (𝒪 D.host.F) = v) ∧ (∀ PP ∈ w.Q, PP.under (𝒪 D.host.F) ∉ w.S₀) ∧ Set.InjOn (fun PP : Pl D.host.E => PP.under (𝒪 D.host.F)) w.Q ∧ (∀ PP ∈ w.Q, Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChainTwistPlacesSplitUnramifiedQuotientsStatement

theorem **_root_.GlobalTheory.T.chainTwistPlaces_split_unramifiedQuotients** : ∀ (D : D7) (c : Com D) (w : Twist D), w.Is c → (∀ v, v ∈ c.S ↔ v ∈ w.S₀ ∨ ∃ PP ∈ w.Q, PP.under (𝒪 D.host.F) = v) ∧ (∀ PP ∈ w.Q, PP.under (𝒪 D.host.F) ∉ w.S₀) ∧ Set.InjOn (fun PP : Pl D.host.E => PP.under (𝒪 D.host.F)) w.Q ∧ (∀ PP ∈ w.Q, Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharConjFixedEqOne

theorem **nontrivial_AF** (D : D7) : Nontrivial (AF D.host.F)

theorem **finrank_AE** (D : D7) : Module.finrank (AF D.host.F) (AE D) = 2

theorem **cA_tmul** (D : D7) (a : AF D.host.F) (e : D.host.E) : cA D (a ⊗ₜ[D.host.F] e) = a ⊗ₜ[D.host.F] star e

theorem **trace_mem** (D : D7) (y : AE D) : ∃ b : AF D.host.F, algebraMap (AF D.host.F) (AE D) b = y + cA D y

theorem **fixed_mem** (D : D7) (y : AE D) (hy : cA D y = y) : ∃ b : AF D.host.F, algebraMap (AF D.host.F) (AE D) b = y

theorem **norm_scalar** (D : D7) (b : AF D.host.F) : Algebra.norm (AF D.host.F) (algebraMap (AF D.host.F) (AE D) b) = b ^ 2

theorem **fixed_unit** (D : D7) (x : (AE D)ˣ) (hx : cA D (x : AE D) = x) : ∃ a : (AF D.host.F)ˣ, idF D a = x

theorem **GlobalTheory.char_conj_fixed_eq_one** : ∀ (D : D7) (μ : (AE D)ˣ →* Circle) (n : ℕ), ResEta D μ n → Even n → ∀ (V : HSp D) (x : (AE D)ˣ), IsDetRoot V 1 x → μ x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharConjFixedEqOneStatement

theorem **_root_.GlobalTheory.T.character_eq_one_of_isDetRoot_of_even** : ∀ (D : D7) (μ : (AE D)ˣ →* Circle) (n : ℕ), ResEta D μ n → Even n → ∀ (V : HSp D) (x : (AE D)ˣ), IsDetRoot V 1 x → μ x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharIntegrable

theorem **isHalf_smul** {e : AE D} {u : Mx D m n} : IsHalf e u ↔ e • u = u

theorem **heis_add** {x y : WO S T} (h : symplO S T x y = 0) (f : WO S T → ℂ) : heisO ψ S T (x + y) f = heisO ψ S T x (heisO ψ S T y f)

theorem **heis_comm** (x y : WO S T) (f : WO S T → ℂ) : heisO ψ S T x (heisO ψ S T y f) = ψ (symplO S T x y) • heisO ψ S T y (heisO ψ S T x f)

theorem **heis_smul** (x : WO S T) (c : ℂ) (f : WO S T → ℂ) : heisO ψ S T x (c • f) = c • heisO ψ S T x f

theorem **fix_half** (h₁ : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e Λ 1 f₀ Tm) {u : WO S T} (hu : u ∈ Λ) (hu' : IsHalf e u) : heisO ψ S T u f₀ = f₀

theorem **fix_conj** (hLL' : ∀ u ∈ Λ, IsHalf e u → ∀ y ∈ Λ, IsHalf (cA D e) y → ψ (symplO S T u y) = 1) (h₁ : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e Λ 1 f₀ Tm) {u : WO S T} (hu : u ∈ Λ) (hu' : IsHalf (cA D e) u) : heisO ψ S T u f₀ = f₀

theorem **lattice_facts** (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) (he : e * e = e) (hinf : infPart D e = 0) : (∀ u ∈ latO S T ΛS ΛT fun _ => True, IsHalf e u → ∀ y ∈ latO S T ΛS ΛT fun _ => True, IsHalf (cA D e) y → ψ (symplO S T u y) = 1) ∧ (∀ x, IsHalf e x → (∀ y ∈ latO S T ΛS ΛT fun _ => True, IsHalf (cA D e) y → ψ (symplO S T x y) = 1) → x ∈ latO S T ΛS ΛT fun _ => True) ∧ (∀ y, IsHalf (cA D e) y → (∀ x ∈ latO S T ΛS ΛT fun _ => True, IsHalf e x → ψ (symplO S T x y) = 1) → y ∈ latO S T ΛS ΛT fun _ => True) ∧ (∀ k : ℕ, 0 < k → ∃ R : Set (WO S T), R.Finite ∧ ∀ t ∈ (latO S T ΛS ΛT fun _ => True) ⊓ half S T e, ∃ r ∈ R, ∃ u ∈ (latO S T ΛS ΛT fun _ => True) ⊓ half S T e, t = r + k • u) ∧ ∀ y, IsHalf (cA D e) y → ∃ k : ℕ, 0 < k ∧ k • y ∈ latO S T ΛS ΛT fun _ => True

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharSymplOHConjHalfRest

theorem **comm4** {R : Type} [CommRing R] (c a h H : R) : c * (a * h * H) = c * a * H * h

theorem **split_half** {E R : Type} [Field E] [CharZero E] [CommRing R] (f : E →+* R) (x k : R) : x * k = x * (k * f ((2 : ℕ) : E)⁻¹) + x * (k * f ((2 : ℕ) : E)⁻¹)

theorem **half_add** {F R : Type} [Field F] [CharZero F] [CommRing R] (g : F →+* R) (t : R) : g (1 / 2) * (t + t) = t

theorem **hO_smul_right** (a : AE D) (x y : WO S T) : hO S T x (a • y) = cA D a * hO S T x y

theorem **hO_nsmul_right** (n : ℕ) (x y : WO S T) : hO S T x (n • y) = (n : AE D) * hO S T x y

theorem **hO_nsmul_left** (n : ℕ) (x y : WO S T) : hO S T (n • x) y = (n : AE D) * hO S T x y

theorem **char_symplOH_conjHalf_rest** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, (2 : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (c : ℕ), 0 < c → ∀ (Y Z : WO S T), c • Y ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) (fun _ => True) → Z ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) (fun _ => True) → e • Y = c • (e • Z) → ψ (symplOH S T (cA D e • Y) ((1 - cA D e) • Y)) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharTieTrueCols

theorem **charTie_trueCols** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) (L : SizeData₂ (I.rec7 l Z)), GlobalTheory.IndefiniteRowSideFullHecke.PlaceLogThetaBounds.CharTie (numDatumOfParameters (I.rec7 l Z) L) (seamOf B (I.cols l) Z.W Z.V Z.S Z.N Z.G) (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterAnnihilatorCompactBox

theorem **trace_mul_smul_single** (D : D7) {m n : ℕ} (y : Mx D n m) (r : AF D.host.F) (i : Fin n) (j : Fin m) (a : D.host.E) : Algebra.trace (AF D.host.F) (AE D) (y · (r • Matrix.single j i (incA D a) : Mx D m n)).trace = r * Algebra.trace (AF D.host.F) (AE D) (incA D a * y i j)

theorem **finPart_sum_tmul** (D : D7) {d : ℕ} (x : Fin d → Af D.host.F) (e : Fin d → D.host.E) : finPart D (∑ l, pairA D.host.F 0 (x l) ⊗ₜ[D.host.F] e l) = ∑ l, x l ⊗ₜ[D.host.F] e l

theorem **annih_mem_box** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (m n : ℕ), ∀ V ∈ nhds (0 : Mx D m n), ∃ K : Set (Mx D n m), IsCompact K ∧ ∀ y : Mx D n m, (∀ ξ ∈ V, ξ.map (infPart D) = 0 → ψ (Algebra.trace (AF D.host.F) (AE D) (y · ξ).trace) = 1) → InBox K y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterAnnihilatorCompactBoxStatement

theorem **_root_.GlobalTheory.T.character_annihilator_compact_box** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (m n : ℕ), ∀ V ∈ nhds (0 : Mx D m n), ∃ K : Set (Mx D n m), IsCompact K ∧ ∀ y : Mx D n m, (∀ ξ ∈ V, ξ.map (infPart D) = 0 → ψ (Algebra.trace (AF D.host.F) (AE D) (y · ξ).trace) = 1) → InBox K y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterArgumentCongrRootEntrySumMod

theorem **rootSum_eq** {A : Type} [CommRing A] {n ρ k : ℕ} (hn : n = 2 * ρ + k + 1) (ε : A) (m : Matrix (Fin n) (Fin n) A) : rootSum ρ k ε m = (∑ a : Fin ρ, m ⟨a.1, by omega⟩ ⟨if a.1 + 1 < ρ then a.1 + 1 else ρ + k, by split_ifs <;> omega⟩) + ∑ i : Fin k, if h : i.1 + 1 < k then ε * m ⟨ρ + i.1, by omega⟩ ⟨ρ + (i.1 + 1), by omega⟩ else 0

theorem **hatL_eq_Lf** (I : AutInst) : Formalism.L2.Herm.hatL (F

theorem **matIn_int** (I : AutInst) {β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} {x : I.Gf} (hx : (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β) (i j : Fin I.n) : matIn I β x i j ∈ IntAf I.F I.E

theorem **mul_matIn** (I : AutInst) (β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) (x : I.Gf) : (β : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) * matIn I β x = I.fmat x * (β : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F))

theorem **mulVec_mem_Lf** (I : AutInst) {β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (hβ : IsFrame β I.L) {x : I.Gf} (hx : (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β) {v : VA I} (hv : v ∈ I.Lf) : I.fmat x *ᵥ v ∈ I.Lf

theorem **col_mem_Lf** (I : AutInst) {β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (hβ : IsFrame β I.L) (j : Fin I.n) : colOf I β j ∈ I.Lf

theorem **mL_le_Lf** (I : AutInst) (g : ℕ) : mL I g ≤ I.Lf

theorem **mulVec_mem_mL** (I : AutInst) (g : ℕ) {M : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (hM : ∀ y ∈ I.Lf, M *ᵥ y ∈ I.Lf) {v : VA I} (hv : v ∈ mL I g) : M *ᵥ v ∈ mL I g

theorem **hA_col** (I : AutInst) (β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) (x : I.Gf) (p q : Fin I.n) : hA I (colOf I β p) (I.fmat x *ᵥ colOf I β q) = ∑ l, matIn I β x l q * gramIn I β p l

theorem **pair_zmod** (I : AutInst) (g ρ k : ℕ) (b : ℕ → VA I) (β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) (hβ : IsLiftFrame I g b β) (hG : ∀ i j, ZMod' I g (gramIn I β i j - flagGram (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) I.n ρ k i j)) (x :

I.Gf) (hx : (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β) (p q t : Fin I.n) (hp : p.1 ≠ ρ + k) (ht : t.1 = dualIdx ρ k p.1) : ZMod' I g (hA I (b p.1) (I.fmat x *ᵥ b q.1) − matIn I β x t q)

theorem **chiArg_sub_rootSum** : ∀ (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → ∀ β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsLiftFrame I g (flagVec e0 f0 z0 bw bwd) β → (∀ i j, ZMod' I g (gramIn I β i j − flagGram (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) I.n ρ k i j)) → ∀ x : I.Gf, (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β → ZMod' I g (chiArg I e0 f0 z0 bw bwd x − rootSum ρ k ew (matIn I β x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterArgumentCongrRootEntrySumModStatement

theorem **_root_.GlobalTheory.T.characterArgument_congr_rootEntrySum_mod** : ∀ (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → ∀ β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsLiftFrame I g (flagVec e0 f0 z0 bw bwd) β → (∀ i j, ZMod' I g (gramIn I β i j − flagGram (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) I.n ρ k i j)) → ∀ x : I.Gf, (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β → ZMod' I g (chiArg I e0 f0 z0 bw bwd x − rootSum ρ k ew (matIn I β x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterResidueFieldsEqProdGenericShift

theorem **exists_shift** {F : Type} [Field F] [Finite F] (ψ : AddChar F ℂ) (hψ : ψ ≠ 1) (χ : AddChar F ℂ) : ∃ a : F, ∀ x, χ x = ψ (a * x)

theorem **prod_shift** {ι : Type} (k : ι → Type) [∀ i, CommRing (k i)] {S : Type} [CommRing S] (Pg : Finset ι) (p p' : ∀ i, S →+* k i) (χ : S → ℂ) (ψ : ∀ i, AddChar (k i) ℂ) (hψ : ∀ i ∈ Pg, ∀ χ' : AddChar (k i) ℂ, ∃ a, ∀ x, χ' x = ψ i (a * x)) (hadd : ∀ s t, χ (s + t) = χ s * χ t) (hker : ∀ s, (∀ i ∈ Pg, p i s = 0 ∧ p' i s = 0) → χ s = 1) (hsurj : ∀ u u' : ∀ i, k i, ∃ s, ∀ i ∈ Pg, p i s = u i ∧ p' i s = u' i) (hposition : ∀ i ∈ Pg, (∃ s, (∀ j ∈ Pg, (j ≠ i → p j s = 0) ∧ p' j s = 0) ∧ χ s ≠ 1) ∧ ∃ s, (∀ j ∈ Pg, p j s = 0 ∧ (j ≠ i → p' j s = 0)) ∧ χ s ≠ 1) : ∃ a₁ a₂ : ∀ i, k i, (∀ i ∈ Pg, a₁ i ≠ 0 ∧ a₂ i ≠ 0) ∧ ∀ s, χ s = ∏ i ∈ Pg, ψ i (a₁ i * p i s + a₂ i * p' i s)

theorem **char_eq_prod_shift** : ∀ (E : Type) [Field E] [NumberField E] (S : Type) [CommRing S] (Pg : Finset (Pl E)) (p p' : ∀ PP : Pl E, S →+* kP PP) (χ : S → ℂ) (ψ0 : ∀ PP : Pl E, AddChar (kP PP) ℂ), IsGenericPsi Pg ψ0 → (∀ s t, χ (s + t) = χ s * χ t) → (∀ s, (∀ PP ∈ Pg, p PP s = 0 ∧ p' PP s = 0) → χ s = 1) → (∀ u u' : ∀ PP : Pl E, kP PP, ∃ s, ∀ PP ∈ Pg, p PP s = u PP ∧ p' PP s = u' PP) → (∀ PP ∈ Pg, (∃ s, (∀ PP' ∈ Pg, (PP' ≠ PP → p PP' s = 0) ∧ p' PP' s = 0) ∧ χ s ≠ 1) ∧ ∃ s, (∀ PP' ∈ Pg, p PP' s = 0 ∧ (PP' ≠ PP → p' PP' s = 0)) ∧ χ s ≠ 1) → ∃ a₁ a₂ : ∀ PP : Pl E, kP PP, (∀ PP ∈ Pg, a₁ PP ≠ 0 ∧ a₂ PP ≠ 0) ∧ ∀ s, χ s = ∏ PP ∈ Pg, ψ0 PP (a₁ PP * p PP s + a₂ PP * p' PP s)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterResidueFieldsEqProdGenericShiftStatement

theorem **_root_.GlobalTheory.T.character_residueFields_eq_prod_genericShift** : ∀ (E : Type) [Field E] [NumberField E] (S : Type) [CommRing S] (Pg : Finset (Pl E)) (p p' : ∀ PP : Pl E, S →+* kP PP) (χ : S → ℂ) (ψ0 : ∀ PP : Pl E, AddChar (kP PP) ℂ), IsGenericPsi Pg ψ0 → (∀ s t, χ (s + t) = χ s * χ t) → (∀ s, (∀ PP ∈ Pg, p PP s = 0 ∧ p' PP s = 0) → χ s = 1) → (∀ u u' : ∀ PP : Pl E, kP PP, ∃ s, ∀ PP ∈ Pg, p PP s = u PP ∧ p' PP s = u' PP) → (∀ PP ∈ Pg, (∃ s, (∀ PP' ∈ Pg, (PP' ≠ PP → p PP' s = 0) ∧ p' PP' s = 0) ∧ χ s ≠ 1) ∧ ∃ s, (∀ PP' ∈ Pg, p PP' s = 0 ∧ (PP' ≠ PP → p' PP' s = 0)) ∧ χ s ≠ 1) → ∃ a₁ a₂ : ∀ PP : Pl E, kP PP, (∀ PP ∈ Pg, a₁ PP ≠ 0 ∧ a₂ PP ≠ 0) ∧ ∀ s, χ s = ∏ PP ∈ Pg, ψ0 PP (a₁ PP * p PP s + a₂ PP * p' PP s)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterUnramifiedAwayLevel

theorem **mu_charUnramAt** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (t : W.P.T) (n : ℕ) (x : HeightOneSpectrum (𝒪 D.host.F)), 1 ≤ n → n ≤ D.R.m → (∀ PP ∈ (pk W).Pg, PP.under (𝒪 D.host.F) ≠ x) → (∀ PP ∈ (pk W).Q t, PP.under (𝒪 D.host.F) ≠ x) → GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D ((pk W).toLevCh.mu t n) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharacterUnramifiedAwayLevelStatement

theorem **_root_.GlobalTheory.T.character_unramified_away_level** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (t : W.P.T) (n : ℕ) (x : HeightOneSpectrum (𝒪 D.host.F)), 1 ≤ n → n ≤ D.R.m → (∀ PP ∈ (pk W).Pg, PP.under (𝒪 D.host.F) ≠ x) → (∀ PP ∈ (pk W).Q t, PP.under (𝒪 D.host.F) ≠ x) → GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D ((pk W).toLevCh.mu t n) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharactersMatchBundle

theorem **GlobalTheory.Fam.realAt_cols** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → (∀ j : Fin q.B.R.m, Formalism.IndefR.Links.ChrEq ((pk (packB b)).mu0 ((j : ℕ) + 1)) (q.w.I.C.mu0 j)) ∧ (∀ t, Formalism.IndefR.Links.ChrEq ((pk (packB b)).xi t) (q.w.I.xi (b.P.I.tup t))) ∧ (∀ t PP, PP ∈ (pk (packB b)).Q t ↔ PP ∈ Formalism.L2.Col.QE q.w.I.sel (q.w.I.tup (b.P.I.tup t) : Formalism.L2.DIV.Chr q.B.fields.F q.B.fields.E q.B.fields.K)) ∧ LevTrue (packB b) ∧ Nonempty (Formalism.IndefR.RealRec b.B.Q) ∧ Formalism.IndefR.RealPins b.B.Q ∧ (∃ s, b.P.A3.fam3.Ok3 s) ∧ GlobalTheory.Core.Mi.IdE b.S.X ∧ ∀ u, Formalism.L2.SG.ColDict.LevelMatch b.S.X (b.S.tr u) b.S.Y.hK

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CharactersMatchBundleStatement

theorem **_root_.GlobalTheory.T.characters_match_bundle** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → (∀ j : Fin q.B.R.m, Formalism.IndefR.Links.ChrEq ((pk (packB b)).mu0 ((j : ℕ) + 1)) (q.w.I.C.mu0 j)) ∧ (∀ t, Formalism.IndefR.Links.ChrEq ((pk (packB b)).xi t) (q.w.I.xi (b.P.I.tup t))) ∧ (∀ t PP, PP ∈ (pk (packB b)).Q t ↔ PP ∈ Formalism.L2.Col.QE q.w.I.sel (q.w.I.tup (b.P.I.tup t) : Formalism.L2.DIV.Chr q.B.fields.F q.B.fields.E q.B.fields.K)) ∧ LevTrue (packB b) ∧ Nonempty (Formalism.IndefR.RealRec b.B.Q) ∧ Formalism.IndefR.RealPins b.B.Q ∧ (∃ s, b.P.A3.fam3.Ok3 s) ∧ GlobalTheory.Core.Mi.IdE b.S.X ∧ ∀ u, Formalism.L2.SG.ColDict.LevelMatch b.S.X (b.S.tr u) b.S.Y.hK

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartBasePairDatumRationalApproxEqTwisted

theorem **phiT_tw** {B : Base} {X : PreX B.toPre} (I : TC B X) (u : B.Word) (a : I.ι) (i : ℕ) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u i) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (i + 1)) (k y : TC.GA B (B.R.N i)) (k' y' : TC.GA B (B.R.N (i + 1))) (hp : TC.PtOf k p.1.1) (hpN : p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i)) (hp' : TC.PtOf k' p'.1.1) (hpN' : p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1))) (hq : TC.PtOf (y * k) q.1.1) (hq' : TC.PtOf (y' * k') q'.1.1) : ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), (∀ P ∈ (B.primes X).vT u, GlobalTheory.Transport.CongAt B P M₀ y y' x x₀) → Formalism.L2.Col.phiT B.fields.K p.1.1.deco.1 p'.1.1.deco.1 x₀ = Formalism.L2.Col.phiT B.fields.K q.1.1.deco.1 q'.1.1.deco.1 x

theorem **xiAt_transport_tw** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (hok : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i + 1 ≤ B.R.m) (b : Idx I.sel (I.tup a) (I.idx a) i) (b' : Idx I.sel (I.tup a) (I.idx a) (i + 1)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u i) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (i + 1)) (k y : TC.GA B (B.R.N i)) (k' y' : TC.GA B (B.R.N (i + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1)) → TC.PtOf (y ∗ k) q.1.1 → TC.PtOf (y' ∗ k') q'.1.1 → cosetIn q = cosetIn p → cosetIn q' = cosetIn p' → ∃ M₀ : ℕ, 0 < M₀ ∧ ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), (∀ P ∈ Formalism.L2.Col.Sall (Pr

theorem **tw_resid_datum** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) (c : (I.stp u a i).Cl) (c' : (I.stp u a (i + 1)).Cl) g g', g ∈ GfC _ → g' ∈ GfC _ → finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) g = ((A.K i (Nat.le_of_succ_le hi) (A.chOf i c)).y (c, rfl)).1.1 → finM _ (rankN (I.rec7 l Z) (i + 1) hi) g' = ((A.K (i + 1) hi (A.chOf (i + 1) c')).y (c', rfl)).1.1 → ∃ M : ℕ, 0 < M ∧ ∀ (γ γ₀ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (v : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), v ∈ stdHom _ _ → IsHalf A.e v → γ₀.map (fun r => A.e ∗ incA (I.rec7 l Z) r) = actO _ _ g⁻¹ g'⁻¹ (γ.map fun r => A.e ∗ incA (I.rec7 l Z) r) + M • v → A.xi i hi (A.chOf i c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartBasePairDatumRationalApproxEqTwistedStatement

theorem **_root_.GlobalTheory.T.chartBasePair_datum_rational_approx_eq_twisted** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) (c : (I.stp u a i).Cl) (c' : (I.stp u a (i + 1)).Cl) g g', g ∈ GfC _ → g' ∈ GfC _ → finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) g = ((A.K i (Nat.le_of_succ_le hi) (A.chOf i c)).y (c, rfl)).1.1 → finM _ (rankN (I.rec7 l Z) (i + 1) hi) g' = ((A.K (i + 1) hi (A.chOf (i + 1) c')).y (c', rfl)).1.1 → ∃ M : ℕ, 0 < M ∧ ∀ (γ γ₀ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (v : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), v ∈ stdHom _ _ → IsHalf A.e v → γ₀.map (fun r => A.e ∗ incA (I.rec7 l Z) r) = actO _ _ g⁻¹ g'⁻¹ (γ.map fun r => A.e

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartBasePointsLevelDatumEqLocalProd

theorem **phiT_base_eq_prod** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ J : List ι, Places p J (fun v => ∃ i, v = (p i).v) → ∀ j (hj : j + 1 ≤ B.R.m) (α : Ch j) (γ : Ch (j + 1)) (z : Matrix (Fin ((I.rec7 l Z).sp.X (j + 1)).rank) (Fin ((I.rec7 l Z).sp.X j).rank) (I.rec7 l Z).host.E), B.embLL (Formalism.L2.Col.phiT _ (A.p0 j α).1.1.deco.1 (A.p0 (j + 1) γ).1.1.deco.1 (toHom _ _ (rankN (I.rec7 l Z) j (Nat.le_of_succ_le hj)) (ran

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartBasePointsLevelDatumEqLocalProdStatement

theorem **_root_.GlobalTheory.T.chartBasePoints_levelDatum_eq_local_prod** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ J : List ι, Places p J (fun v => ∃ i, v = (p i).v) → ∀ j (hj : j + 1 ≤ B.R.m) (α : Ch j) (γ : Ch (j + 1)) (z : Matrix (Fin ((I.rec7 l Z).sp.X (j + 1)).rank) (Fin ((I.rec7 l Z).sp.X j).rank) (I.rec7 l Z).host.E), B.embLL (Formalism.L2.Col.phiT _ (A.p0 j α).1.1.deco.1 (A.p0 (j + 1) γ).1.1.deco.1 (toHom _ _ (rank

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartCharacterTrivialOnStandardPairStabilizer

theorem **prV_toEAf_idem_mul** {v v' : HeightOneSpectrum (𝒪 D.host.F)} {ε : AF D.host.F} (hε : IsPlaceIdem D v ε) (hv : v' ≠ v) (a : AE D) : Formalism.UG.prV (E

theorem **finM_gl_map_prV** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) {n : ℕ} (hn : S.rank = n) {m : Type} [Fintype m] [DecidableEq m] (β : Frame p S m) (g : GL m k) {v' : HeightOneSpectrum (𝒪 D.host.F)} (hv : v' ≠ p.v) : (finM S hn (β.gl g)).map (Formalism.UG.prV (E

theorem **GlobalTheory.AdForm.ChiStd.not_both_mem** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) {a c : ℕ} (ha : a.Prime) (hc : c.Prime) (hne : a ≠ c) (h1 : ((a : ℕ) : 𝒪 F) ∈ v.asIdeal) (h2 : ((c : ℕ) : 𝒪 F) ∈ v.asIdeal) : False

theorem **GlobalTheory.AdForm.chi_std_trivial** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : GlobalTheory.AdForm.TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), GlobalTheory.AdForm.StdT I l Z u p β Q L → ∀ j, j ≤ B.R.m → ∀ b, ∀ i (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k), g • Q j i = Q j i → ∀ h : (β j i).gl g ∈ A.U j b, A.chi j b ⟨_, h⟩ = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartCharacterTrivialOnStandardPairStabilizerStatement

theorem **_root_.GlobalTheory.T.chartCharacter_trivial_on_standardPairStabilizer** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : GlobalTheory.AdForm.TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), GlobalTheory.AdForm.StdT I l Z u p β Q L → ∀ j, j ≤ B.R.m → ∀ b, ∀ i (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k), g • Q j i = Q j i → ∀ h : (β j i).gl g ∈ A.U j b, A.chi j b ⟨_, h⟩ = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelCharacterRepsOnAdelicUnitaryPoints

theorem **exists_finM_GfC** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (y : Formalism.UG.UPts (Formalism.L2.Forms.sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ a ∈ GfC V, GlobalTheory.AdForm.finM V hn a = y.1.1

theorem **levData** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (i : ℕ) (hi : i ≤ B.R.m) {T : ℕ} {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {cl : Type} {stab : cl → Type} [∀ c, Group (stab c)] {lat : cl → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)} (K : LevelChart (I.rec7 l Z) T bad i hi stab lat), ∃ (U : Subgroup (GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i))) (chi : U →* ℂˣ) (h : cl → GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i)), (∀ g, g ∈ U ↔ g ∈ GfC _ ∧ ∃ κ : K.lev.U, κ.1.1 = finM _ (rankN (I.rec7 l Z) i hi) g) ∧ (∀ (g : U) (κ : K.lev.U), κ.1.1 = finM _ (rankN (I.rec7 l Z) i hi) g.1 → chi g = K.lev.chi κ) ∧ (∀ c, h c ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) (h c) = (K.y c).1.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelCharacterRepsOnAdelicUnitaryPointsStatement

theorem **_root_.GlobalTheory.T.chartLevelCharacterReps_on_adelicUnitaryPoints** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (i : ℕ) (hi : i ≤ B.R.m) {T : ℕ} {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {cl : Type} {stab : cl → Type} [∀ c, Group (stab c)] {lat : cl → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)} (K : LevelChart (I.rec7 l Z) T bad i hi stab lat), ∃ (U : Subgroup (GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i))) (chi : U →* ℂˣ) (h : cl → GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i)), (∀ g, g ∈ U ↔ g ∈ GfC _ ∧ ∃ κ : K.lev.U, κ.1.1 = finM _ (rankN (I.rec7 l Z) i hi) g) ∧ (∀ (g : U) (κ : K.lev.U), κ.1.1 = finM _ (rankN (I.rec7 l Z) i hi) g.1 → chi g = K.lev.chi κ) ∧ (∀ c, h c ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) (h c) = (K.y c).1.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelColumnSubOneMemBaseLattice

theorem **GlobalTheory.AdForm.level_col_sub_one_mem** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : GlobalTheory.AdForm.TwDat I l Z u a Ch), A.Is → ∀ j (hj : j ≤ B.R.m) b, ∀ κ ∈ A.U j b, ∀ v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F), ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ v.asIdeal → ∀ s : Fin ((I.rec7 l Z).R.N j), (fun r => Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelColumnSubOneMemBaseLatticeStatement

theorem **_root_.GlobalTheory.T.chartLevel_column_sub_one_mem_baseLattice** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : GlobalTheory.AdForm.TwDat I l Z u a Ch), A.Is → ∀ j (hj : j ≤ B.R.m) b, ∀ κ ∈ A.U j b, ∀ v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F), ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ v.asIdeal → ∀ s : Fin ((I.rec7 l Z).R.N j), (fun r => Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelComponentFixesStandardPair

theorem **GlobalTheory.AdForm.CompFix.intMat_keeps** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : Type} [Fintype n] [DecidableEq n] (M M' : Matrix n n k) (hMM : M * M' = 1) (hM : ∀ r s, M r s ∈ 𝒪[k]) (hM' : ∀ r s, M' r s ∈ 𝒪[k]) (r₀ : n) (hr : ∀ s, valuation k (M' r₀ s - (1 : Matrix n n k) r₀ s) < 1) (z : n → k) : ((∀ r, (M' *ᵥ z) r ∈ 𝒪[k]) ↔ ∀ r, z r ∈ 𝒪[k]) ∧ ((∀ r, z r ∈ 𝒪[k]) → valuation k ((M' *ᵥ z) r₀ - z r₀) < 1)

theorem **GlobalTheory.AdForm.comp_fixes_stdPair** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : GlobalTheory.AdForm.TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), GlobalTheory.AdForm.StdT I l Z u p β Q L → ∀ j, j ≤ B.R.m → ∀ b, ∀ κ ∈ A.U j b, ∀ i (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k), κ * ((β j i).gl g)⁻¹ ∈ grpOff ((I.rec7 l Z).sp.X j) (· = (p i).v) → g • Q j i = Q j i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelComponentFixesStandardPairStatement

theorem **_root_.GlobalTheory.T.chartLevel_component_fixes_standardPair** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : GlobalTheory.AdForm.TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), GlobalTheory.AdForm.StdT I l Z u p β Q L → ∀ j, j ≤ B.R.m → ∀ b, ∀ κ ∈ A.U j b, ∀ i (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k), κ * ((β j i).gl g)⁻¹ ∈ grpOff ((I.rec7 l Z).sp.X j) (· = (p i).v) → g • Q j i = Q j i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelExistsAdelicLevelTrivialRealLift

theorem **exists_mem_U_finM** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) (z : (A.K i hi β).lev.U), ∃ w : A.U i β, z.1.1 = finM _ (rankN (I.rec7 l Z) i hi) w.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelExistsAdelicLevelTrivialRealLiftStatement

theorem **_root_.GlobalTheory.T.chartLevel_exists_adelicLevel_trivialReal_lift** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) (z : (A.K i hi β).lev.U), ∃ w : A.U i β, z.1.1 = finM _ (rankN (I.rec7 l Z) i hi) w.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelPairActionFiniteDatumLaw

theorem **latFr_stdLat** (D : D7) (S : HSp D) (a : ℕ) (hS : S.rank = a) : latFr S hS (Formalism.L2.Col.stdLat D.host.E a) = intVec D S.rank

theorem **mem_stdHom_iff_act0** (D : D7) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (k : GA D S) (k' : GA D T) (h1 : Moves S hS 1 (Formalism.L2.Col.stdLat D.host.E a)) (h1' : Moves T hT 1 (Formalism.L2.Col.stdLat D.host.E b)) (hk : Moves S hS k (Formalism.L2.Col.stdLat D.host.E a)) (hk' : Moves T hT k' (Formalism.L2.Col.stdLat D.host.E b)) : ∀ x, x ∈ stdHom S T ↔ act0 S T k k' x ∈ stdHom S T

theorem **lev_law** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch₀ Ch : ℕ → Type) (A : TwDat I l Z u a Ch₀) (ι : ∀ j, Ch j → Ch₀ j) ψ θ x vac c₀ kom dd w Ram rd cK, PinX I l Z ψ θ x vac c₀ → PinColL A ι ψ θ x vac c₀ kom dd w Ram rd cK → ∀ j, j < B.R.m → ∀ α γ (κ : A.U j (ι j α)) (κ' : A.U (j + 1) (ι (j + 1) γ)), kom j (κ.1, κ'.1) (comb ψ _ _ (dd j α γ) (vac j)) = (((A.chi j _ κ : ℂˣ) : ℂ)⁻¹ * ((A.chi (j + 1) _ κ' : ℂˣ) : ℂ)) • comb ψ _ _ (dd j α γ) (vac j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelPairActionFiniteDatumLawStatement

theorem **_root_.GlobalTheory.T.chartLevelPair_action_finiteDatum_law** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch₀ Ch : ℕ → Type) (A : TwDat I l Z u a Ch₀) (ι : ∀ j, Ch j → Ch₀ j) ψ θ x vac c₀ kom dd w Ram rd cK, PinX I l Z ψ θ x vac c₀ → PinColL A ι ψ θ x vac c₀ kom dd w Ram rd cK → ∀ j, j < B.R.m → ∀ α γ (κ : A.U j (ι j α)) (κ' : A.U (j + 1) (ι (j + 1) γ)), kom j (κ.1, κ'.1) (comb ψ _ _ (dd j α γ) (vac j)) = (((A.chi j _ κ : ℂˣ) : ℂ)⁻¹ * ((A.chi (j + 1) _ κ' : ℂˣ) : ℂ)) • comb ψ _ _ (dd j α γ) (vac j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelsKeepStandardLattice

theorem **inclE_mem_IA** (r : 𝒪 E) : inclE (F

theorem **sigR_mem_IA** (σ : E →ₐ[F] E) {y : E ⊗[F] FiniteAdeleRing (𝒪 F) F} (hy : y ∈ IA F E) : Formalism.UG.sigR σ (FiniteAdeleRing (𝒪 F) F) y ∈ IA F E

theorem **hatP_of_mem** (π : 𝒪 E) (L : Submodule (𝒪 E) (Fin n → E)) (hL : ∀ x, x ∈ L → ∀ j : Fin n, ∃ r : 𝒪 E, x j = (r : E) ∧ (j.1 + 1 = n → π ∣ r)) {x : Fin n → E ⊗[F] FiniteAdeleRing (𝒪 F) F} (hx : x ∈ latS (F

theorem **inv_eq_cT** {J : Matrix (Fin n) (Fin n) E} (hJ : J = 1) (lev : StepLevel F E σ J T f bad) {g : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)} (hg : g ∈ lev.U) (a b : Fin n) : (g⁻¹ : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)).1 a b = Formalism.UG.sigR σ (FiniteAdeleRing (𝒪 F) F) (g.1 b a)

theorem **img_hatP** {g : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)} (hg : g ∈ lev.U) (y : Fin n → E) (hy : y ∈ lev.Lb) : HatP π (g.1 *ᵥ fun i => inclE (F

theorem **incl_single** (j : Fin n) (c : E) : (fun i => inclE (F

theorem **col_mem** {g : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)} (hg : g ∈ lev.U) (j : Fin n) (hj : ¬ j.1 + 1 = n) (a : Fin n) : g.1 a j ∈ IA F E

theorem **pi_ne_zero** (l : Fin n) (hl : l.1 + 1 = n) : π ≠ 0

theorem **last_mem** {g : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)} (hg : g ∈ lev.U) (l : Fin n) (hl : l.1 + 1 = n) : g.1 l l ∈ IA F E

theorem **entry_mem** (hJ : J = 1) {g : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)} (hg : g ∈ lev.U) (a b : Fin n) : g.1 a b ∈ IA F E

theorem **entry_memD** {D : D7} {n : ℕ} {J : Matrix (Fin n) (Fin n) D.host.E} {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (π : 𝒪 D.host.E) (lev : StepLevel D.host.F D.host.E (sigD D) J T f bad) (hLb : ∀ x, x ∈ lev.Lb ↔ ∀ j : Fin n, ∃ r : 𝒪 D.host.E, x j = (r : D.host.E) ∧ (j.1 + 1 = n → π ∣ r)) (hJ : J = 1) {g : GL (Fin n) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F)} (hg : g ∈ lev.U) (a b : Fin n) : g.1 a b ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **levels_keep_stdLat** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) β, ∀ k ∈ A.U i β, Moves _ (rankN (I.rec7 l Z) i hi) k (Formalism.L2.Col.stdLat B.fields.E _)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelsKeepStandardLatticeStatement

theorem **_root_.GlobalTheory.T.chartLevels_keep_standardLattice** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) β, ∀ k ∈ A.U i β, Moves _ (rankN (I.rec7 l Z) i hi) k (Formalism.L2.Col.stdLat B.fields.E _)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelsRepsRepresentUnitaryDoubleCosets

theorem **finHomN_val** (g : GA D V) : (finHomN V hn g).1 = finM V hn g

theorem **finHomN_eq_one** (g : GA D V) : finHomN V hn g = 1 ↔ g ∈ GinfC V

theorem **lift_mem** (e : ɪ(GfC V) ≃* ɪ(UPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F))) (he : ∀ g, (e g).1.1 = finM V hn g.1) (κ : S.U) : (e.symm (κ.1, ((S.U_eq κ.1).1 κ.2).1)).1 ∈ U ∧ finM V hn (e.symm (κ.1, ((S.U_eq κ.1).1 κ.2).1)).1 = κ.1.1

theorem **reps_of_chart** (hy : Function.Bijective (fun c => (Quotient.mk'' (y c) : Cls S))) : IsReps V U h

theorem **stabOrd_of_chart** {stab : cl → Type} [∀ c, Group (stab c)] (stabI : ∀ c, stab c →* UJ (sigD D : D.host.E →+* D.host.E) (V.gram.reindex (finCongr hn) (finCongr hn))) (hinj : ∀ c, Function.Injective (stabI c)) (hran : ∀ c (γ : UJ (sigD D : D.host.E →+* D.host.E) (V.gram.reindex (finCongr hn) (finCongr hn))), γ ∈ (stabI c).range ↔ bc (inclE (F

theorem **twDat_reps** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) β, IsReps ((I.rec7 l Z).sp.X i) (A.U i β) (A.h i β) ∧ (Continuous fun κ => ((A.chi i β κ : ℂˣ) : ℂ)) ∧ ∀ c, stabOrd (A.U i β) (A.h i β c) = Nat.card ((I.chain l u a).stab i c.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartLevelsRepsRepresentUnitaryDoubleCosetsStatement

theorem **_root_.GlobalTheory.T.chartLevels_reps_represent_unitaryDoubleCosets** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) β, IsReps ((I.rec7 l Z).sp.X i) (A.U i β) (A.h i β) ∧ (Continuous fun κ => ((A.chi i β κ : ℂˣ) : ℂ)) ∧ ∀ c, stabOrd (A.U i β) (A.h i β c) = Nat.card ((I.chain l u a).stab i c.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartScalarFormAdelicPointEqClassTransport

theorem **finM_inf'** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (k : GA D V) (hk : k ∈ GinfC V) : finM V hn k = 1

theorem **infM_fin''** {D : D7} (θ : InfFam D) (V : HSp D) {n : ℕ} (hn : V.rank = n) (k : GA D V) (hk : k ∈ GfC V) (τ : D.what₂.Tau) : infM θ V hn k τ = 1

theorem **atInf_mul_inv** {E : Type} [Field E] {n : ℕ} {ι : Type} (emb : ι → E →+* ℂ) (γ : GL (Fin n) E) (τ : ι) : atInf emb γ.val τ * atInf emb (γ⁻¹).val τ = 1

theorem **eq_of_mul_eq_one** {n : ℕ} (A B C D : Matrix (Fin n) (Fin n) ℂ) (h1 : A * B = 1) (h2 : C * D = 1) (h3 : B = C) : A = D

theorem **exists_level** {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) {Ch : Type} {chOf : (I.stp u a i).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)} {U : Ch → Subgroup (TC.GA B (B.R.N i))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a i).Cl → TC.GA B (B.R.N i)} {st : ∀ c, (I.stp u a i).Stab c → GL (Fin (B.R.N i)) B.host.E} {k : Ch → TC.GA B (B.R.N i)} (h : I.Reads u a i chOf Lb U chi y st k) (b : (I.stp u a i).β) (d : (I.stp u a i).D b) (hco : cosetIn d = cosetIn ((I.stp u a i).rep ((I.stp u a i).cl b d))) (w : TC.GA B (B.R.N i)) (hw : TC.PtOf (w * k (chOf ((I.stp u a i).cl b d))) d.1.1) : ∃ x : U (chOf ((I.stp u a i).cl b d)), w = bc (inclE (F

theorem **GlobalTheory.AdForm.rd_trTo** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ j (hj : j ≤ B.R.m), ∀ φ ∈ (I.fm l u a).Fi j, ∀ b (c : (I.chain l u a).cls j) (d : (I.stp u a j).D c.1) (x : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X j)) (z : TC.GA B (B.R.N j)), x ∈ GfC _ → z.1 = finM _ (rankN (I.rec7 l Z) j hj) x → TC.PtOf (z * A.kb j b) (extTo I.SQ d.1.1) → A.chOf j ((I.stp u a j).cl c.1 d) = b → cosetIn d = cosetIn ((I.stp u a j).rep ((I.stp u a j).cl c.1 d)) → ∀ y, A.rd θ j hj b φ y x = (I.stp u a j).trTo B.fields.emb B.embLL c.1 d (φ ((I.stp u a j).cl c.1 d)) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChartScalarFormAdelicPointEqClassTransportStatement

theorem **_root_.GlobalTheory.T.chartScalarForm_adelicPoint_eq_classTransport** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ j (hj : j ≤ B.R.m), ∀ φ ∈ (I.fm l u a).Fi j, ∀ b (c : (I.chain l u a).cls j) (d : (I.stp u a j).D c.1) (x : GA (I.rec7 l

Z) ((I.rec7 l Z).sp.X j)) (z : TC.GA B (B.R.N j)), x ∈ GfC _ → z.1 = finM _ (rankN (I.rec7 l Z) j hj) x → TC.PtOf (z ∗ A.kb j b) (extTo I.SQ d.1.1) → A.chOf j ((I.stp u a j).cl c.1 d) = b → cosetIn d = cosetIn ((I.stp u a j).rep ((I.stp u a j).cl c.1 d)) → ∀ y, A.rd θ j hj b φ y x = (I.stp u a j).trTo B.fields.emb B.embLL c.1 d (φ ((I.stp u a j).cl c.1 d)) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChebotarevDegreeOneValuationSubrings

theorem **chebOmSt_holds** : ReciprocityTheory.ChartRestriction.L1.ChebOmSt

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChebotarevDegreeOneValuationSubringsStatement

theorem **_root_.GlobalTheory.T.chebotarev_degreeOne_valuationSubrings** : ReciprocityTheory.ChartRestriction.L1.ChebOmSt

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ChiPowOfLevG

theorem **Deep.mono** {n : ℕ} {P : HeightOneSpectrum (𝒪 E)} {N N' : ℕ} (h : N ≤ N') {w : GL (Fin n) (AfE F E)} (hw : Deep P N' w) : Deep P N w

theorem **Deep.ext** {n M : ℕ} (hM : M = n + 1) {P : HeightOneSpectrum (𝒪 E)} {N : ℕ} {w : GL (Fin n) (AfE F E)} (hw : Deep P N w) : Deep P N (extGL hM w)

theorem **exists_deep** (hc : CMData F E) {n T f : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : J = 1) {bad : Set (HeightOneSpectrum (𝒪 F))} (S : StepLevel F E hc.sig J T f bad) (P : HeightOneSpectrum (𝒪 E)) (hq : Valued.v ((qOf P : ℕ) : P.adicCompletion E) = WithZero.exp (-1 : ℤ)) (hu : (conjPrime₂ P).under (𝒪 F) = P.under (𝒪 F)) : ∃ N : ℕ, ∀ w ∈ UAd hc n, Deep P N w → w ∈ S.U

theorem **exists_rat** (hc : CMData F E) {n m : ℕ} (hm : 0 < m) (Q : Finset (HeightOneSpectrum (𝒪 E))) (hQ : ∀ P ∈ Q, conjPrime₂ P ∉ Q) (H : Subgroup (GL (Fin n) (AfE F E))) (hH : ∀ P ∈ Q, ∃ N : ℕ, ∀ w ∈ UAd hc n, Deep P N w → w ^ m ∈ H) (g : GL (Fin n) (AfE F E)) : ∃ z ∈ H, ∀ P ∈ Q, ∃ γ : GL (Fin n) E, ∀ r s, adAt F P ((g ∗ z).1 r s) = algebraMap E (P.adicCompletion E) (γ.1 r s)

theorem **exists_ka** (u : B.Word) (a : I.ι) {Ch : Type} {chOf : (I.stp u a B.R.m).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)} {U : Ch → Subgroup (GA B (B.R.N B.R.m))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a B.R.m).Cl → GA B (B.R.N B.R.m)} {st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E} {k : Ch → GA B (B.R.N B.R.m)} (hR : I.Reads u a B.R.m chOf Lb U chi y st k) (c : (I.stp u a B.R.m).Cl) (g : GA B (B.R.N B.R.m)) (m : (I.stp u a B.R.m).G c.1) (hp : PtOf (g ∗ k (chOf c)) ((I.stp u a B.R.m).act c.1 m ((I.stp u a B.R.m).rep c)).1.1) : ∃ κ : U (chOf c), bc (inclE (F

theorem **ext_mem_of_pt** (l : I.Laws) (u : B.Word) (a : I.ι) {Ch : Type} {chOf : (I.stp u a B.R.m).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)} {U : Ch → Subgroup (GA B (B.R.N B.R.m))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a B.R.m).Cl → GA B (B.R.N B.R.m)} {st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E} {k : Ch → GA B (B.R.N B.R.m)} (hR : I.Reads u a B.R.m chOf Lb U chi y st k) (hT : ∀ β (d : XPt (B.primes X) B.R I.sel (I.tup a)) (g' : GA B (B.R.N B.R.m - 1)), PtOf g' d → PtOf (extGL (N_m_succ B.R) g' ∗ k β) (tilF (Pr

theorem **mem_sb** {a : I.ι} {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ I.Q a) : P ∈ SBof I a

theorem **ext_mem** (hI : I.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) {Ch : Type} {chOf : (I.stp u a B.R.m).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)} {U : Ch → Subgroup (GA B (B.R.N B.R.m))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a B.R.m).Cl → GA B (B.R.N B.R.m)} {st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E} {k : Ch → GA B (B.R.N B.R.m)} (hR : I.Reads u a B.R.m chOf Lb U chi y st k) (hT : ∀ β (d : XPt (B.primes X) B.R I.sel (I.tup a)) (g' : GA B (B.R.N B.R.m - 1)), PtOf g' d → PtOf (extGL (N_m_succ B.R) g' ∗ k β) (tilF (Pr

theorem **extM_map** {A A' : Type} [CommRing A] [CommRing A'] {N n : ℕ} (hN : N = n + 1) (f : A →+* A') (g : Matrix (Fin n) (Fin n) A) : (extM hN g).map f = extM hN (g.map f)

theorem **isExt_iff** (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) {R' : Type} [CommRing R'] (φ : (I.rec7 l Z).host.E →+* R') (g' : Matrix (Fin (B.R.N B.R.m - 1)) (Fin (B.R.N B.R.m - 1)) R') (g : Matrix (Fin (B.R.N B.R.m)) (Fin (B.R.N B.R.m)) R') : IsExt (I.rec7 l Z) (z0 X u) φ g' g ↔ g = extM (N_m_succ B.R) g'

theorem **deep_twist** (hI : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) {n T f : ℕ} {J : Matrix (Fin n) (Fin n) (I.rec7 l Z).host.E} (hJ : J = 1) {bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))} (S : StepLevel (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) J T f bad) : ∀ P ∈ I.Q a, ∃ N : ℕ, ∀ w ∈ UAd B.sp.cm n, Deep (F

theorem **mem_GfC_iff** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (g : GA D V) : g ∈ GfC V ↔ ((mt V g).reindex (finCongr hn) (finCongr hn)).map (infPart D) = 1

theorem **finM_of_hasMat** {D : D7} {h : GA D D.sp.X'} {y} (hy : GlobalTheory.Core.Rk.Top.HasMat h y) : ((mt _ h).reindex (finCongr (rankP D)) (finCongr (rankP D))).map (toEAf D) = y.1

theorem **emb_mem_chi** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch₀ : ℕ → Type) (A : TwDat I l Z u a Ch₀), A.Is → ∀ emb : GA (I.rec7 l Z) (I.rec7 l Z).sp.X' →* GA (I.rec7 l Z) ((I.rec7 l Z).sp.X B.R.m), IsEmbTop (D

theorem **levFn_pow** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} {LL : Type} [CommRing LL] (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) (L : LevDat E LL n) : ∃ m : ℕ, 0 < m ∧ ∀ x, levFn β L x ^ m = 1

theorem **chi_pow_of_levG** {D : D7} {u : D.what₂.Word} {C : ColumnChain D.R D.what₂.Tau (D.hcW u)} {A : ColumnsAtT D.R C} {Pm : PeriodMap D.R C} {M : MemberForms D.what₂.LL D.what₂.embLL D.what₂.places D.what₂.Tau (D.hcW u) C A Pm} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (K : ClassesAtPrimes M bad) {x : GA D D.sp.X'} {J : Set (GA D D.sp.X')} {β : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {L : LevDat D.host.E ℂ D.sp.X'.rank} (h : LevG K x J (levChi _ β L)) : ∃ n : ℕ, 0 < n ∧ ∀ κ, K.lev.chi κ ^ n = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ClassLatticeSelfDual

theorem **mem_span_of_mem_latS** (R : Type) [CommRing R] [Algebra F R] [Nontrivial R] (S : Subring (E ⊗[F] R)) (L : Submodule (𝒪 E) (n → E)) (x : n → E) (hx : (fun i => inclE (F

theorem **eq_globLat** (Λ L : Submodule (𝒪 E) (n → E)) (hfin : Module.Finite (𝒪 E) Λ) (hspan : Submodule.span E (Λ : Set (n → E)) = ⊤) (hL : Module.Finite (𝒪 E) L) (g : GL n (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (h : ∀ x : n → E ⊗[F] FiniteAdeleRing (𝒪 F) F, x ∈ latS (F

theorem **classLattice_selfDual** : ∀ (D : D7) (u : D.what₂.Word) (K : ClsPTied u) (c : (D.fm u).clsP), Submodule.IsLattice D.host.E (K.lat c) ∧ ∀ y, y ∈ K.lat c ↔ ∀ x ∈ K.lat c, Formalism.UG.hform (sigD D : D.host.E →+* D.host.E) (gramP D) x y ∈ (algebraMap (𝒪 D.host.E) D.host.E).range

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ClassicalThetaOfDatumSmulMove

theorem **hatS_mul_left** {V : Type} [AddCommGroup V] [Module ℂ V] (H : Sesq V) (vol : ℝ) (SA : Finset V) (Φ : V → ℂ) (p : ℂ) (y : V) : hatS H vol SA (fun s => p ∗ Φ s) y = p ∗ hatS H vol SA Φ y

theorem **thC_smul_move** : ∀ {E : Type} [Field E] [NumberField E] [StarRing E] {ι : Type} [Fintype ι] {a b : ℕ} (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (AA : Submodule (𝓞 E) E) (SS : Finset (IsDedekindDomain.HeightOneSpectrum (𝓞 E))) (N : Submodule (𝓞 E) (Formalism.L2.Col.Sp E a)) (N' : Submodule (𝓞 E) (Formalism.L2.Col.Sp E b)) (c : ℕ) (p : ℂ) (Ξ : Formalism.L2.Col.HomE E a b → ℂ) (θ : Formalism.L2.Ker.VV ι a b → ℂ), Formalism.L2.Ker.thC emb δ d sc AA SS N N' c (fun x => p * Ξ x) θ = Formalism.L2.Ker.thC emb δ d sc AA SS N N' c Ξ (fun z => p * θ z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ClassicalThetaOfDatumSmulMoveStatement

theorem **_root_.GlobalTheory.T.classicalThetaOfDatum_smul_move** : ∀ {E : Type} [Field E] [NumberField E] [StarRing E] {ι : Type} [Fintype ι] {a b : ℕ} (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (AA : Submodule (𝓞 E) E) (SS : Finset (IsDedekindDomain.HeightOneSpectrum (𝓞 E))) (N : Submodule (𝓞 E) (Formalism.L2.Col.Sp E a)) (N' : Submodule (𝓞 E) (Formalism.L2.Col.Sp E b)) (c : ℕ) (p : ℂ) (Ξ : Formalism.L2.Col.HomE E a b → ℂ) (θ : Formalism.L2.Ker.VV ι a b → ℂ), Formalism.L2.Ker.thC emb δ d sc AA SS N N' c (fun x => p * Ξ x) θ = Formalism.L2.Ker.thC emb δ d sc AA SS N N' c Ξ (fun z => p * θ z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CloudAddNodeLeTenBaseSize

theorem **cloud_add_node_le_ten_b** : ∀ (C : Formalism.MainNotions.Inhab.FC) (cD ch : ℝ) (D : D7) (L : SizeData₂ D), Val C cD ch L → SizesAt L → L.sc.x ≤ L.sc.X' → ∀ τ : D.what₂.Tau, (∀ i j, (((D.hc τ).v i : ℚ) : ℝ) + (((D.hc τ).b j : ℚ) : ℝ) ≤ 10 * L.sc.b) ∧ ∀ j j', (((D.hc τ).b j' : ℚ) : ℝ) − (((D.hc τ).b j : ℚ) : ℝ) ≤ 10 * L.sc.b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CloudAddNodeLeTenBaseSizeStatement

theorem **_root_.GlobalTheory.T.cloud_add_node_le_ten_base_size** : ∀ (C : Formalism.MainNotions.Inhab.FC) (cD ch : ℝ) (D : D7) (L : SizeData₂ D), Val C cD ch L → SizesAt L → L.sc.x ≤ L.sc.X' → ∀ τ : D.what₂.Tau, (∀ i j, (((D.hc τ).v i : ℚ) : ℝ) + (((D.hc τ).b j : ℚ) : ℝ) ≤ 10 * L.sc.b) ∧ ∀ j j', (((D.hc τ).b j' : ℚ) : ℝ) − (((D.hc τ).b j : ℚ) : ℝ) ≤ 10 * L.sc.b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmGoodModelOfGlobalArtinReciprocity

theorem **cmGoodModel_of_globalArtinReciprocity** (hA : GlobalTheory.Published.GlobalArtinReciprocity) : Formalism.L2.Th1.CM.CM2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmHostImaginaryGeneratorNeZero

theorem **nonempty_vol** (V : HSp D) : Nonempty (Vol V)

theorem **nonempty_volN** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : Nonempty (VolN d)

theorem **nonempty_volM** (D : D7) (m n : ℕ) : Nonempty (VolM D m n)

theorem **nonempty_volU** (D : D7) (ρ : ℕ) : Nonempty (VolU D ρ)

theorem **coCpt_of_mul** {V : HSp D} (h : ∃ K : Set (GA D V), IsCompact K ∧ ∀ g, ∃ γ ∈ GQ V, ∃ k ∈ K, g = γ * k) : CoCpt V

theorem **coCpt_of_isAniso** {V : HSp D} (h : IsAniso V) : CoCpt V

theorem **coCpt_of_posDef** {V : HSp D} (h : ∀ τ, (gramAt D V τ).PosDef) : CoCpt V

theorem **exists_weil** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S T : HSp D) {χS χT : (AE D)ˣ →* Circle} (h1 : IsHeckeChar D χS) (h2 : IsHeckeChar D χT) (h3 : ResEta D χS S.rank) (h4 : ResEta D χT T.rank) : ∃ om, IsWeilDatumO ψ S T χS χT om

theorem **exists_weil_pair** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S T : HSp D) : ∃ χS χT om, IsWeilDatumO ψ S T χS χT om

theorem **nonempty_step** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (hV : V.rank % 2 = V₀.rank % 2) (hW : W.rank % 2 = W₀.rank % 2) : Nonempty (Step ψ V V₀ W W₀ ρ σ)

theorem **even_im_hermitian_sub** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : S.rank % 2 = S₀.rank % 2

theorem **nonempty_stepBF** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} (B : BesLev D ρ) (J : JacLev D ρ) : Nonempty (StepBF ψ B J)

theorem **nonempty_stepFB** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {σ : ℕ} (J : JacLev D (σ + 1)) (B : BesLev D σ) : Nonempty (StepFB ψ J B)

theorem **cmHost_imaginaryGenerator_ne_zero** (D : D7) : D.host.xi0 ≠ 0

theorem **posDef_posLine** (D : D7) (τ : D.what₂.Tau) : (gramAt D (posLine D) τ).PosDef

theorem **coCpt_posLine** (D : D7) : CoCpt (posLine D)

theorem **isLineVec_negLine** {c : D.host.E} (hc : c ≠ 0) : IsLineVec (negLine D) (fun _ => c)

theorem **exists_last** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (B : BesLev D 0) (J : JacLev D 0) (hJ : CoCpt J.W) {T : HSp D} (hT : CoCpt T) : ∃ t : Last ψ T B, t.J = J

theorem **exists_data** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {T : HSp D} {ρ : ℕ} {B : BesLev D ρ} (c : Chain ψ T ρ B) : ∃ Φ : c.Data, c.IsTheta Φ ∧ AllNonzero c Φ

theorem **exists_chain_of_isStdChar** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ (T : HSp D) (B : BesLev D 1) (c : Chain ψ T 1 B), 0 < B.A.rank ∧ 0 < T.rank ∧ CoresPos c ∧ ∃ Φ : c.Data, c.IsTheta Φ ∧ AllNonzero c Φ

theorem **isThetaFam_om** {ψ : AddChar (AF D.host.F) ℂ} {S L : HSp D} {χS χL : (AE D)ˣ →* Circle} {om : Om S L} (hom : IsWeilDatumO ψ S L χS χL om) {Φ₀ : WO S L → ℂ} (hΦ : IsThetaFunO ψ S L Φ₀) : IsThetaFam ψ S L fun h => om (h, 1) Φ₀ where

theorem **exists_isThetaFam** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S L : HSp D) : ∃ Y, IsThetaFam ψ S L Y ∧ ∀ h, Y h ≠ 0

theorem **exists_chain** (D : D7) : (∃ ψ : AddChar (AF D.host.F) ℂ, IsStdChar D ψ (-1)) ∧ ∀ ψ : AddChar (AF D.host.F) ℂ, IsStdChar D ψ (-1) → (∃ (T : HSp D) (B : BesLev D 1) (c : Chain ψ T 1 B), 0 < B.A.rank ∧ 0 < T.rank ∧ CoresPos c ∧ ∃ Φ : c.Data, c.IsTheta Φ ∧ AllNonzero c Φ) ∧ ∀ S L : HSp D, ∃ Y, IsThetaFam ψ S L Y ∧ ∀ h, Y h ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmLevelHermitianIntegral

theorem **lvl_consequence_one** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (v0 : InfinitePlace K), IsConjAt v0 → ∀ S : CSD K, S.Good 1 → (∀ i j, S.H i j ∈ cmMax K) → (∀ v : InfinitePlace K, (S.H.map v.embedding).PosDef) → ∀ (G : Type) [Group G] [Fintype G] (r : Representation K G S.Forms), S.ActGood r → S.ActRat r → ∀ (P : Subgroup G) (ψ : P →* Kˣ), (∀ i j, S.lvlH r P ψ i j ∈ cmMax K) ∧ (∀ v : InfinitePlace K, ((S.lvlH r P ψ).map v.embedding).PosDef) ∧ Formalism.HermitianLatticesTheory.dual (hform (S.lvlH r P ψ)) (S.lvlNab r P ψ) ≤ S.lvlNab r P ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmLevelHermitianIntegralStatement

theorem **_root_.GlobalTheory.T.cm_level_hermitian_integral** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (v0 : InfinitePlace K), IsConjAt v0 → ∀ S : CSD K, S.Good 1 → (∀ i j, S.H i j ∈ cmMax K) → (∀ v : InfinitePlace K, (S.H.map v.embedding).PosDef) → ∀ (G : Type) [Group G] [Fintype G] (r : Representation K G S.Forms), S.ActGood r → S.ActRat r → ∀ (P : Subgroup G) (ψ : P →* Kˣ), (∀ i j, S.lvlH r P ψ i j ∈ cmMax K) ∧ (∀ v : InfinitePlace K, ((S.lvlH r P ψ).map v.embedding).PosDef) ∧ Formalism.HermitianLatticesTheory.dual (hform (S.lvlH r P ψ)) (S.lvlNab r P ψ) ≤ S.lvlNab r P ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmMultiplierIntegralSeriesAtInertPrimeProof

theorem **norm_eq_one_of_quad_root** {l : ℕ} [Fact l.Prime] (x : ℂ_[l]) (N : ℕ) (hN : ¬ l ∣ N) (hx : x ^ 2 - x + (N : ℂ_[l]) = 0) : ‖x‖ = 1

theorem **endOmega_holds** : EndOmega

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmMultiplierIntegralSeriesAtInertPrimeProofStatement

theorem **_root_.GlobalTheory.T.cmMultiplierIntegralSeriesAtInertPrime_proof** : EndOmega

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CmulMemPerOOfMemHalfBO

theorem **cmul_mem_perO_of_mem_halfBO** : ∀ (D : D7) (a b : ℕ) (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (SS : Finset (HeightOneSpectrum (𝒪 D.host.E))) (c : ℕ), 0 < c → ∀ y ∈ halfBO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, (c : D.host.E) • y ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CntOnSingleton

theorem **cntOn_singleton** : Formalism.HermitianLatticesTheory.SiegelMassRatio → ∀ c : ℝ, CntOn {c}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoCptOfIsAnisoOrPosDef

theorem **coCpt_of_mul** {D : D7} {V : HSp D} (h : ∃ K : Set (GA D V), IsCompact K ∧ ∀ g, ∃ γ ∈ GQ V, ∃ k ∈ K, g = γ * k) : CoCpt V

theorem **coCpt_of_isAniso_or_posDef** : ∀ (D : D7) (V : HSp D), (IsAniso V → CoCpt V) ∧ ((∀ τ, (gramAt D V τ).PosDef) → CoCpt V)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoCptXZero

theorem **coCpt_X_zero** : ∀ (D : D7), GlobalTheory.Unf.CoCpt (D.sp.X 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CocRZeroLeft

theorem **cocR_add_left** (x y z : Formalism.L2.Col.HomE E a b) : cocR emb δ d sc (x + y) z = cocR (ι

theorem **cocR_zero_left** (z : Formalism.L2.Col.HomE E a b) : cocR (ι

theorem **hv_embH_smul** (hconj : ∀ τ x, emb τ (star x) = starRingEnd ℂ (emb τ x)) (r : E) (x y : Formalism.L2.Col.HomE E a b) : HV (ι

theorem **cocR_smul** (hconj : ∀ τ x, emb τ (star x) = starRingEnd ℂ (emb τ x)) (r : E) (x y : Formalism.L2.Col.HomE E a b) : cocR (ι

theorem **semiChar_twO** : ∀ (E : Type) [Field E] [NumberField E] [StarRing E] (ι : Type) [Fintype ι] [Nonempty ι] (a b : ℕ) (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)) (c : ℕ) (e : HeightOneSpectrum (𝒪 E) → ℤ), δ ≠ 0 → (∀ (τ : ι) (x : E), emb τ (star x) = starRingEnd ℂ (emb τ x)) → 0 < c → (∀ Q, e (Formalism.L2.Col.conjPrime₂ Q) = -e Q) → (∀ Q, e Q = 1 ∨ e Q = 0 ∨ e Q = -1) → Formalism.L2.Th1.H.IsSelfDual (HV sc) (laV emb δ M) → Formalism.L2.Th1.H.IsSemiChar (HV sc) (laV emb δ M) (alV d sc) → Formalism.L2.Th1.H.IsSemiChar (HV sc) (laV emb δ (twist₂ M c e)) (alV d sc)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoeLatFrameOfEq

theorem **coe_latFrame_of_eq** {D : D7} (S : HSp D) (hs : ∀ x : D.host.E, IsIntegral ℤ x → IsIntegral ℤ (star x)) (N : ℕ) (h : S = Formalism.BasicNotions.Examples₂.posSpace hs N) : (GlobalTheory.PlaceModel.latFrame S : Set (Fin S.rank → D.host.E)) = GlobalTheory.AdForm.intVec D S.rank

theorem **gram_of_eq** {D : D7} (S : HSp D) (hs : ∀ x : D.host.E, IsIntegral ℤ x → IsIntegral ℤ (star x)) (N : ℕ) (h : S = Formalism.BasicNotions.Examples₂.posSpace hs N) : S.gram = 1

theorem **cartan_of_gram** {D : D7} {S : HSp D} (h : S.gram = 1) (τ : D.what₂.Tau) : IsCartanInv (gramAt D S τ) 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoeffAtNegOneZetaOfEisCoeffAt

theorem **isExtAt_sub** (a s₀ : ℂ) : IsExtAt (fun s : ℂ => s - a) s₀ fun s => s - a

theorem **coeffAt_neg_one_of_holValueAt** {h : ℂ → ℂ} {a c : ℂ} (hv : HolValueAt (fun s => (s - a) * h s) a c) : CoeffAt h a (-1) c

theorem **cont_sub_mul** {h : ℂ → ℂ} (hc : Continues h) (a s : ℂ) (hs : ContinuousAt (cont h) s) : cont (fun t => (t - a) * h t) s = (s - a) * cont h s

theorem **eisSer_sub_smul** (s₀ : ℂ) (g : GA D (dblSp V)) : eisSer V ((fun s => s - s₀) • Φ) g = fun s => (s - s₀) * eisSer V Φ g s

theorem **isZetaFamily_sub_smul** {μ : (AE D)ˣ →* Circle} {m : Measure (GA D V)} {f₁ f₂ : GA D V → ℂ} (hZ : IsZetaFamily V μ m Φ f₁ f₂) (s₀ : ℂ) : IsZetaFamily V μ m ((fun s => s - s₀) • Φ) f₁ f₂

theorem **eisContinues_sub_smul** (hE : EisContinues V Φ) (s₀ : ℂ) : EisContinues V ((fun s => s - s₀) • Φ)

theorem **eisRegularAt_sub_smul** (hE : EisContinues V Φ) (s₀ s₁ : ℂ) {k : ℕ} (h : EisRegularAt V Φ s₁ k) : EisRegularAt V ((fun s => s - s₀) • Φ) s₁ k

theorem **eisValueAt_sub_smul** {s₀ : ℂ} {E₁ : GA D (dblSp V) → ℂ} (hp : EisPoleLE V Φ s₀ 1) (hc : EisCoeffAt V Φ s₀ (-1) E₁) : EisValueAt V ((fun s => s - s₀) • Φ) s₀ E₁

theorem **coeffAt_neg_one_zeta_of_eisCoeffAt** : ∀ (D : D7) (V : HSp D) (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ (m : Measure (GA D V)), m.IsHaarMeasure → ∀ (Φ : ℂ → GA D (dblSp V) → ℂ), GlobalTheory.Eis.EisContinues V Φ → (∀ g, AnalyticOnNhd ℂ (GlobalTheory.Eis.eisSer V Φ g) (rightOf (V.rank / 2))) → (∀ s₀, ∃ k, GlobalTheory.Eis.EisRegularAt V Φ s₀ k) → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → GlobalTheory.Eis.IsZetaFamily V μ m Φ f₁ f₂ → ∀ (s₀ : ℂ) (E₁ : GA D (dblSp V) → ℂ), GlobalTheory.Eis.EisPoleLE V Φ s₀ 1 → GlobalTheory.Eis.EisCoeffAt V Φ s₀ (-1) E₁ → CoeffAt (GlobalTheory.Eis.zeta V μ m Φ f₁ f₂) s₀ (-1) (GlobalTheory.Eis.dblInt V μ m E₁ f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColCrdHolds

theorem **colC_eq_lin** (W : RealPack ex) (u : D.what₂.Word) : colC W u = Formalism.IndefR.Cx.lin (fun k => D.jN u (W.P.I.eNu u (Pi.single k 1)))

theorem **colCrd_holds** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), ColCrd W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColMatchAll

theorem **colMatchAll** : ColMatchAll

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColPairSmul

theorem **colPair** (q : GlobalTheory.FinPl.SplitPt D k) (S : HSp D) (x y : Fin S.rank → D.host.E) : ∑ r, ∑ t, q.pr (incA D (x r)) ∗ q.pr (cA D (incA D (HSpace.gram S r t))) ∗ q.pr (cA D (incA D (y t))) = q.pr (incA D (star (∑ r, ∑ t, star (x r) ∗ HSpace.gram S r t ∗ y t)))

theorem **mat_entry** (S : HSp D) {m n : Type} [Fintype m] [Fintype n] (A : Matrix (Fin S.rank) m D.host.E) (B : Matrix (Fin S.rank) n D.host.E) (b : m) (i : n) : (Aᴴ ∗ HSpace.gram S ∗ B) b i = ∑ r, ∑ t, star (A r b) ∗ HSpace.gram S r t ∗ B t i

theorem **colPair_smul** (q : GlobalTheory.FinPl.SplitPt D k) (S : HSp D) (c : k) (x y : Fin S.rank → D.host.E) (col : Fin S.rank → k) (hcol : ∀ r, col r = c ∗ q.pr (incA D (x r))) : ∑ r, ∑ t, col r ∗ q.pr (cA D (incA D (HSpace.gram S r t))) ∗ q.pr (cA D (incA D (y t))) = c ∗ q.pr (incA D (star (∑ r, ∑ t, star (x r) ∗ HSpace.gram S r t ∗ y t)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColStep0MemLat

theorem **cl_rep_reRep** (k : ((stepY ST y β Λ 0 pref).reRep pf).Cl) : (stepY ST y β Λ 0 pref).cl k.1 (((stepY ST y β Λ 0 pref).reRep pf).rep k) = k

theorem **mem_lat_reRep** [Unique β] (hy : ∀ N, y.ys N = {∅}) (hΛ : ∀ b, Λ b = Λ₀) (h0 : ∀ b, 0 b = LevQFam.top X.E X.K n Q) (h : (((stepY ST y β Λ 0 pref).reRep pf).fam X.emb lam).Ok) (h₀ : (base ST lam Λ₀).Ok) (x : (((stepY ST y β Λ 0 pref).reRep pf).hlat X.emb lam h).V) (φ : (base ST lam Λ₀).EqFun) (hx : ∀ c, x.1 c = φ.1 (((stepY ST y β Λ 0 pref).reRep pf).rep c).1.1.deco) : x ∈ (((stepY ST y β Λ 0 pref).reRep pf).hlat X.emb lam h).L ↔ φ ∈ (base ST lam Λ₀).intDel

theorem **colStep0_mem_lat** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (hX : B.fields.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) (x : ((I.fm l u a).lat 0).V) (φ : ColsK I u), IsCol I u a l x φ → (x ∈ ((I.fm l u a).lat 0).L ↔ φ ∈ MfL (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CollSOn

theorem **collS_on** : Formalism.HermitianLatticesTheory.SiegelMassRatio → ∀ S : Set ℝ, AmpOn S → CountBoundOn S → SmallOn S → TieOn S → On S CollS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CollisionAbsSumBoundOfTwoRowBounds

theorem **collS_of_rowBounds** : Formalism.HermitianLatticesTheory.SiegelMassRatio → ∀ CT C : ℝ, 408 ∗ (Real.exp 1 ∗ Real.pi) ≤ C → CT ≤ C → ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (X : HidaSetting D) (Y : Full X) (hF : FinX Y) (hI : SIso D.s (sOf hF)) (u : D.what₂.Word) (T : TermTie D) (Z : LoopTie D large₂ T u u 1) (C' Cl Cg cb Cu cT C0 : ℝ), thr ≤ C' → thr ≤ Cl → Cnt.Hyp0 D large₂ C' Cl Cg cb Cu cT CT C0 → ConcTie X Y → (Y.tO u u).AbsConv → TieAt hF hI large₂ u T Z → UnitaryVolumeMass.FormAt D u → BridgeLemma.RowsND T Z C → BridgeLemma.RowsDeg T Z C → CollS Y u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CollisionAbsSumBoundOfTwoRowBoundsStatement

theorem **_root_.GlobalTheory.T.collisionAbsSum_bound_of_two_rowBounds** : Formalism.HermitianLatticesTheory.SiegelMassRatio → ∀ CT C : ℝ, 408 ∗ (Real.exp 1 ∗ Real.pi) ≤ C → CT ≤ C → ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (X : HidaSetting D) (Y : Full X) (hF : FinX Y) (hI : SIso D.s (sOf hF)) (u : D.what₂.Word) (T : TermTie D) (Z : LoopTie D large₂ T u u 1) (C' Cl Cg cb Cu cT C0 : ℝ), thr ≤ C' → thr ≤ Cl → Cnt.Hyp0 D large₂ C' Cl Cg cb Cu cT CT C0 → ConcTie X Y → (Y.tO u u).AbsConv → TieAt hF hI large₂ u T Z → UnitaryVolumeMass.FormAt D u → BridgeLemma.RowsND T Z C → BridgeLemma.RowsDeg T Z C → CollS Y u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CollisionTypingsCompare

theorem **uOne_last** (D : D7) (u w : D.what₂.Word) (τ : D.what₂.Tau) : uOne D w τ D.R.k = uOne D u τ D.R.k

theorem **splits_level** (D : D7) (w : D.what₂.Word) : SplitsE D (D.mem w).A.T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColsAtBfOfNormalFrames

theorem **xi_unit** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (hU : q.UnitAA) : valuation k ((ofSplit q).xi) = 1

theorem **line_crit** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {S : HSp D} {L : GlobalTheory.FinPl.Lat S} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k S.rank Y) (u u' : Fin S.rank → D.host.E) (m₀ m₁ : Y) (c₀ c₁ : k) (hL : IsSelfDualC S L) (hI : IntIn q S F L) (h0 : valuation k c₀ = 1) (h1 : valuation k c₁ = 1) (hc0 : ∀ r, F.β r m₀ = c₀ ∗ (ofSplit q).pr (u r)) (hc1 : ∀ r, F.β r m₁ = c₁ ∗ (ofSplit q).pr (u' r)) (hd : ∑ r, ∑ r', star (u r) ∗ S.gram r r' ∗ u' r' = 1) (hp : ∀ m, m ≠ m₁ → ∑ r, ∑ r', (ofSplit q).pr (star (u r)) ∗ (ofSplit q).pr (S.gram r r') ∗ F.β r' m = 0) (a : AE D) (ha : a ∈ GlobalTheory.FinPl.idAt D (· = q.v)) : (fun r => a ∗ incA D (u r)) ∈ GlobalTheory.FinPl.latV S L (· = q.v) ↔ a ∈ GlobalTheory.FinPl.intAt D (· = q.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColsStdOfBWit

theorem **colsStd_of_bWit** {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c) (h : Bl.BWit q b) : q.w.I.Ok ∧ Nonempty (GlobalTheory.Mem.Col2.TC.Pins q.B q.w.px) ∧ GlobalTheory.Mem.Col2.TC.IsTw q.w.I ∧ Nonempty (GlobalTheory.Mem.Col2.TC.StdC q.w.I q.w.l q.Z b.R.T.cols) ∧ GlobalTheory.AdForm.TwStd (packB b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColumnDimensionWindowThreeOfColumnCount

theorem **standingNumericalData_three_of_dataNW_rankMu** : ∀ D : D7, DataNW D → (∀ u, GlobalTheory.Core.Rk.RankMu D u) → StandingNumericalData D 3

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColumnDimensionWindowThreeOfColumnCountStatement

theorem **_root_.GlobalTheory.T.columnDimension_window_three_of_columnCount** : ∀ D : D7, DataNW D → (∀ u, GlobalTheory.Core.Rk.RankMu D u) → StandingNumericalData D 3

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColumnLevelIffRowLatticeKeptAtPlace

theorem **embA_mem_grpAt** {D : D7} {S S₀ A B : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A B) (P : GlobalTheory.FinPl.Pl D → Prop) {a : GA D A} (ha : a ∈ grpAt A P) : embA d o a ∈ grpAt S P

theorem **levrow** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (F : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F) (hK : X.LevelSubgroupCharacter) (B : SBes X h u ψ (D.sp.X 0)) (a₀ : GA D (D.sp.X 0)), GlobalTheory.Core.Mi.IdE X → Formalism.L2.SG.ColDict.LevelMatch X tr hK → B.Adapted tr.wt → IsTransportAdelicCorrection tr a₀ → ∀ v : GlobalTheory.FinPl.Pl D, ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal → ∀ a ∈ grpAt (D.sp.X 0) (· = v), ∃ y : Formalism.UG.UPts (Formalism.L2.Forms.sigD D) (Formalism.L2.Forms.gramN D 0 (Nat.zero_le _)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F), GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) a = y.1.1 ∧ ((y : GL _ _) ∈ F.cls0.lev.U ↔ Moves (D.sp.Y 0) (embA B.d B.o (a₀ * a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ColumnLevelIffRowLatticeKeptAtPlaceStatement

theorem **_root_.GlobalTheory.T.columnLevel_iff_rowLattice_kept_at_place** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (F : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F) (hK : X.LevelSubgroupCharacter) (B : SBes X h u ψ (D.sp.X 0)) (a₀ : GA D (D.sp.X 0)), GlobalTheory.Core.Mi.IdE X → Formalism.L2.SG.ColDict.LevelMatch X tr hK → B.Adapted tr.wt → IsTransportAdelicCorrection tr a₀ → ∀ v : GlobalTheory.FinPl.Pl D, ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal → ∀ a ∈ grpAt (D.sp.X 0) (· = v), ∃ y : Formalism.UG.UPts (Formalism.L2.Forms.sigD D) (Formalism.L2.Forms.gramN D 0 (Nat.zero_le _)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F), GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) a = y.1.1 ∧ ((y : GL _ _)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CommuteOfOff

theorem **commute_of_off** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} {v : HeightOneSpectrum (𝒪 F)} {x k : GL (Fin n) (AfE F E)} (hx : ∀ w, w ≠ v → bc (prV (E

theorem **fr_eq** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] {hc : Formalism.BasicNotions.CMData F E} {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} {g g' : GL (Fin n) (AfE F E)} {p p' : DecoF E n ST SG Q} (h : IsPtOf hc g p) (h' : IsPtOf hc g' p') (P : Q) (e0 : ∀ i j, adAt F P.1 (g.1 i j) = adAt F P.1 (g'.1 i j)) : p.fr P = p'.fr P

theorem **exists_pt** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [StarRing E] [Algebra F E] {hc : Formalism.BasicNotions.CMData F E} {n : ℕ} [NeZero n] {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} {g g' : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) (hg' : g' ∈ UAd hc n) (e0 : ∀ P ∈ Q, ∀ i j, adAt F P (g'.1 i j) = adAt F P (g.1 i j)) : ∃ p' : DecoF E n ST SG Q, IsPtOf hc g' p'

theorem **coset_at** (q : PtAt R sel mu cst u j) (h : j + 1 = R.m) {y : Sp X.E (R.N j)} (hy0 : y ∈ q.1.2) (w : Pr.vGs) : cosetIn q ⟨w.1, Finset.mem_union_left _ w.2⟩ = {z | z - (q.1.1.deco.1.liftFr w).symm y ∈ stdAt X.E (R.N j) w.1} ∧ ((qOf w.1 : ℕ) : X.E) • (q.1.1.deco.1.liftFr w).symm y ∈ stdAt X.E (R.N j) w.1 ∧ bar q w = {redV w.1 (((qOf w.1 : ℕ) : X.E) • (q.1.1.deco.1.liftFr w).symm y)}

theorem **cos_iff_bar** (h : j + 1 = R.m) (w : Pr.vGs) (hq : ((qOf w.1 : ℕ) : 𝒪 X.E) ∉ w.1.asIdeal ^ 2) (q : PtAt R sel mu cst u j) (q' : PtAt R sel mu' cst' u j) : cosetIn q ⟨w.1, Finset.mem_union_left _ w.2⟩ = cosetIn q' ⟨w.1, Finset.mem_union_left _ w.2⟩ ↔ bar q w = bar q' w

theorem **levGAt_eq** (hTw : I.IsTw) (a : I.ι) (j : ℕ) (hj : j < B.R.m) (b : Idx I.sel (I.tup a) (I.idx a) j) : levGAt I.sel (I.tup a) (I.idx a) (I.Tp a) j b = colLevSym I.Ch j

theorem **mem_lev_iff** (hI : I.Ok) (hTw : I.IsTw) (hPr : (B.primes X).Ok) (u : B.Word) (a : I.ι) (j : ℕ) (hj : j + 1 = B.R.m) [NeZero (B.R.N j)] {Ch : Type} {chOf : (I.stp u a j).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N j) → B.host.E)} {U : Ch → Subgroup (TC.GA B (B.R.N j))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a j).Cl → TC.GA B (B.R.N j)} {st : ∀ c, (I.stp u a j).Stab c → GL (Fin (B.R.N j)) B.host.E} {k : Ch → TC.GA B (B.R.N j)} (hR : I.Reads u a j chOf Lb U chi y st k) (c : (I.stp u a j).Cl) (x : TC.GA B (B.R.N j)) (hx : x ∈ UAd B.sp.cm (B.R.N j)) (hc : x * k (chOf c) = k (chOf c) * x) (hc₀ : x * TC.kStd X u (B.R.N j) = TC.kStd X u (B.R.N j) * x) (h1 : ∀ P ∈ I.Q a, x.1.map (adAt B.host.F P) = 1) : x ∈ U (chOf c) ↔ x ∈ levU (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompIncAAlgebraMap

theorem **incA_algebraMap** (t : D.host.F) : incA D (algebraMap D.host.F D.host.E t) = algebraMap (AF D.host.F) (AE D) (algebraMap D.host.F (AF D.host.F) t)

theorem **comp_incA_algebraMap** {v : Pl D} {e : AE D} (he : SplitAt D v e) (t : D.host.F) : comp he (incA D (algebraMap D.host.F D.host.E t)) = algebraMap D.host.F (Fv D.host.F v) t

theorem **comp_sec** {v : Pl D} {e : AE D} (he : SplitAt D v e) (a : Fv D.host.F v) : comp he (sec D v e a) = a

theorem **sec_comp** {v : Pl D} {e : AE D} (he : SplitAt D v e) (x : AE D) : sec D v e (comp he x) = e * x

theorem **int_iff_sec** {v : Pl D} {e : AE D} (he : SplitAt D v e) (a : Fv D.host.F v) : a ∈ 𝒪[Fv D.host.F v] ↔ finPart D (sec D v e a) ∈ intAE D

theorem **emb_apply** {v : Pl D} {e : AE D} (he : SplitAt D v e) (a : D.host.E) : emb he a = comp he (incA D a)

theorem **splitPrime_emb** {v : Pl D} {e : AE D} (he : SplitAt D v e) (PP : HeightOneSpectrum (𝒪 D.host.E)) (h : Formalism.L2.NTheta.IsIdemAt e PP) : GlobalTheory.AdelicU.splitPrime (emb he) = PP

theorem **exists_place** : ∀ (D : D7) (PP : Formalism.L2.DIV.Pl D.host.E) (e : AE D), Formalism.L2.NTheta.IsIdemAt e PP → ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (q : GlobalTheory.FinPl.SplitPt D k), q.e = e ∧ q.v = PP.under (NumberField.RingOfIntegers D.host.F) ∧ (∀ a, (ofSplit q).pr a ∈ 𝒪[k] ↔ Formalism.L2.Col.IntAt PP a) ∧ DenseRange (ofSplit q).pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactAverageProjectionThetaSchrodinger

theorem **realSize_mul_le** (θ : InfFam D) {m n k : ℕ} (Y : Mx D m n) (a : Mx D n k) : realSize θ (Y · a) ≤ (k * realSize θ Y) * realSize θ a

theorem **maj_mt_le** {θ : InfFam D} (hθ : IsInfFam θ) (T : HSp D) (n : ℕ) {K : Set (GA D T)} (hK : IsCompact K) {K₀ : Set (Mx D T.rank n)} (hK₀ : IsCompact K₀) : ∃ K' : Set (Mx D T.rank n), IsCompact K' ∧ ∀ N : ℕ, ∃ C : ℝ, 0 ≤ C ∧ ∀ k ∈ K, ∀ u : Mx D T.rank n, maj θ K₀ N (mt T k⁻¹ · u) ≤ C * maj θ K' N u

theorem **exists_summable_bound** {θ : InfFam D} (hθ : IsInfFam θ) (T : HSp D) (n : ℕ) {K : Set (GA D T)} (hK : IsCompact K) {Φ : Mx D T.rank n → ℂ} (hΦ : Φ ∈ schwartzA θ T.rank n) (x : Mx D T.rank n) : ∃ b : Matrix (Fin T.rank) (Fin n) D.host.E → ℝ, Summable b ∧ ∀ k ∈

K, ∀ ξ, ‖Φ (mt T k⁻¹ · (x + rat ξ : Mx D T.rank n))‖ ≤ b ξ

theorem **hasSum_sch** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (A B : HSp D) {ν : Measure (WO A B)} {𝒴 : Set (WO A B)} (hν : IsCovolOne (ratL B A) ν 𝒴) {f : WO (dblSp A) B → ℂ} (hf : IsThetaFunO ψ (dblSp A) B f) (x y : Mx D B.rank A.rank) : HasSum (fun ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E => ψ (symplO (dblSp A) B ((GlobalTheory.Core.Rk.SWLaw.dblDec A).extX B (x + rat ξ)) ((GlobalTheory.Core.Rk.SWLaw.dblDec A).extY B y)) * sch ψ A B ν 𝒴 f (x + rat ξ)) (ψ (symplOH (dblSp A) B ((GlobalTheory.Core.Rk.SWLaw.dblDec A).extX B x) ((GlobalTheory.Core.Rk.SWLaw.dblDec A).extY B y)) * f ((GlobalTheory.Core.Rk.SWLaw.dblDec A).extX B x + (GlobalTheory.Core.Rk.SWLaw.dblDec A).extY B y))

theorem **projK_theta_sch** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χ : (AE D)ˣ →* Circle) (om : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)), IsWeilDatumO ψ (dblSp A) B 1 χ om → ∀ (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ K : Subgroup (GA D B), IsCompact (K : Set (GA D B)) → ∀ νK : Measure K, IsProbabilityMeasure νK → ∀ φ, IsThetaFunO ψ (dblSp A) B φ → IsThetaFunO ψ (dblSp A) B (projK om νK φ) ∧ sch ψ A B ν 𝒴 (projK om νK φ) = projS νK (sch ψ A B ν 𝒴 φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactAverageProjectionThetaSchrodingerStatement

theorem **_root_.GlobalTheory.T.compactAverageProjection_theta_schrodinger** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χ : (AE D)ˣ →* Circle) (om : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)), IsWeilDatumO ψ (dblSp A) B 1 χ om → ∀ (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ K : Subgroup (GA D B), IsCompact (K : Set (GA D B)) → ∀ νK : Measure K, IsProbabilityMeasure νK → ∀ φ, IsThetaFunO ψ (dblSp A) B φ → IsThetaFunO ψ (dblSp A) B (projK om νK φ) ∧ sch ψ A B ν 𝒴 (projK om νK φ) = projS νK (sch ψ A B ν 𝒴 φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactAverageSchwartzProjectionMemAdelicSchwart

theorem **mt_mul_inv_self** (V : HSp D) (g : GA D V) : Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V g⁻¹ = 1

theorem **map_cancel** {R : Type} [CommRing R] (φ : AE D →+* R) (V : HSp D) (g : GA D V) : (Formalism.L2.Theta.mt V g).map φ * (Formalism.L2.Theta.mt V g⁻¹).map φ = 1

theorem **map_mul_entry** {R R' : Type} [CommRing R] [CommRing R'] (φ : R →+* R') {m n : ℕ} (M : Matrix (Fin m) (Fin m) R) (x : Matrix (Fin m) (Fin n) R) : (fun i j => φ ((M * x) i j)) = fun i j => ∑ c, φ (M i c) * φ (x c j)

theorem **integrable_and_projS_mem** {θ : InfFam D} (hθ : IsInfFam θ) {T : HSp D} {n : ℕ} {K : Subgroup (GA D T)} (hK : IsCompact (K : Set (GA D T))) (ν : Measure K) [IsFiniteMeasure ν] {Φ : Mx D T.rank n → ℂ} (hΦ : Φ ∈ schwartzA θ T.rank n) : (∀ x : Mx D T.rank n, Integrable (fun k : K => Φ (Formalism.L2.Theta.mt T (k : GA D T)⁻¹ · x)) ν) ∧ projS ν Φ ∈ schwartzA θ T.rank n

theorem **projS_mem** : ∀ (D : D7), ∀ θ : InfFam D, IsInfFam θ → ∀ (T : HSp D) (n : ℕ) (K : Subgroup (GA D T)), IsCompact (K : Set (GA D T)) → ∀ ν : Measure K, IsProbabilityMeasure ν → ∀ Φ ∈ schwartzA θ T.rank n, projS ν Φ ∈ schwartzA θ T.rank n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactAverageSchwartzProjectionMemAdelicSchwartStatement

theorem **_root_.GlobalTheory.T.compactAverageSchwartzProjection_mem_adelicSchwartz** : ∀ (D : D7), ∀ θ : InfFam D, IsInfFam θ → ∀ (T : HSp D) (n : ℕ) (K : Subgroup (GA D T)), IsCompact (K : Set (GA D T)) → ∀ ν : Measure K, IsProbabilityMeasure ν → ∀ Φ ∈ schwartzA θ T.rank n, projS ν Φ ∈ schwartzA θ T.rank n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactAverageThetaFamilySmooth

theorem **key** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {lam : AF D.host.F →+ ℂ} (hl : Continuous lam) (hlam : ∀ a, sndA D.host.F a = 0 → ψ a = Complex.exp (lam a)) {n : ℕ} (L : (Fin n → ℝ) →+ WO S T) (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) (w : WO S T) (f : WO S T → ℂ) (hf : IsThetaFunO ψ S T f) (t : Fin n → ℝ) (i : Fin n) : fderiv ℝ (fun t => f (w + L t)) t (Pi.single i 1) = dLine ψ S T (L.comp (AddMonoidHom.single (fun _ : Fin n => ℝ) i)) f (w + L t) + -phase lam S T (L.comp (AddMonoidHom.single (fun _ : Fin n => ℝ) i) 1) (w + L t) * f (w + L t)

theorem **avg_smooth** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (K : Type) [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] (μ : Measure K) [IsFiniteMeasure μ] (𝒢 : Set (K → WO S T → ℂ)), (∀ F ∈ 𝒢, Continuous fun p : K × WO S T => F p.1 p.2) → (∀ F ∈ 𝒢, ∀ k, IsThetaFunO ψ S T (F k)) → (∀ F ∈ 𝒢, ∀ M : ℝ →+ WO S T, Continuous M → (∀ s, (M s).map (finPart D) = 0) → (fun k => GlobalTheory.WeilVacMove.dLine ψ S T M (F k)) ∈ 𝒢) → ∀ F ∈ 𝒢, ∀ (n : ℕ) (L : (Fin n → ℝ) →+ WO S T), Continuous L → (∀ t, (L t).map (finPart D) = 0) → ∀ w, ContDiff ℝ ∞ fun t => ∫ k, F k (w + L t) ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactAverageThetaFamilySmoothStatement

theorem **_root_.GlobalTheory.T.compactAverage_thetaFamily_smooth** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (K : Type) [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] (μ : Measure K) [IsFiniteMeasure μ] (𝒢 : Set (K → WO S T → ℂ)), (∀ F ∈ 𝒢, Continuous fun p : K × WO S T => F p.1 p.2) → (∀ F ∈ 𝒢, ∀ k, IsThetaFunO ψ S T (F k)) → (∀ F ∈ 𝒢, ∀ M : ℝ →+ WO S T, Continuous M → (∀ s, (M s).map (finPart D) = 0) → (fun k => GlobalTheory.WeilVacMove.dLine ψ S T M (F k)) ∈ 𝒢) → ∀ F ∈ 𝒢, ∀ (n : ℕ) (L : (Fin n → ℝ) →+ WO S T), Continuous L → (∀ t, (L t).map (finPart D) = 0) → ∀ w, ContDiff ℝ ∞ fun t => ∫ k, F k (w + L t) ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactCommutingCartanInvolutionMovedStandard

theorem **mt_mul'** (V : HSp D) (g h : GA D V) : Formalism.L2.Theta.mt V (g * h) = Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V h

theorem **compAt_mt_one** (θ : InfFam D) (τ : D.what₂.Tau) (V : HSp D) : compAt θ τ (Formalism.L2.Theta.mt V 1) = 1

theorem **compAt_mt_mul** (θ : InfFam D) (τ : D.what₂.Tau) (V : HSp D) (g h : GA D V) : compAt θ τ (Formalism.L2.Theta.mt V (g * h)) = compAt θ τ (Formalism.L2.Theta.mt V g) * compAt θ τ (Formalism.L2.Theta.mt V h)

theorem **compAt_isom** {θ : InfFam D} (hθ : IsInfFam θ) (V : HSp D) (g : GA D V) (τ : D.what₂.Tau) : (compAt θ τ (Formalism.L2.Theta.mt V g))ᴴ * gramAt D V τ * compAt θ τ (Formalism.L2.Theta.mt V g) = gramAt D V τ

theorem **isCartanInv_conj** {k : ℕ} {G s P Pi : Matrix (Fin k) (Fin k) ℂ} (hs : IsCartanInv G s) (hPi : Pi * P = 1) (hPPi : P * Pi = 1) (hiso : Piᴴ * G * Pi = G) : IsCartanInv G (Pi * s * P)

theorem **cartan_move_std** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) =* G D A.rank, IsChart A μ κ → ∀ (a : GL (Fin A.rank) (AE D)) (p : siegel A), IsConj A a κ p → ∀ θ : InfFam D, IsInfFam θ → ∀ sS : RealMat (dblSp A), (∀ τ, IsCartanInv (gramAt D (dblSp A) τ) (sS τ)) → (∀ k ∈ Khat A a, ∀ τ, compAt θ τ (mt (dblSp A) k) * sS τ = sS τ * compAt θ τ (mt (dblSp A) k)) → ∃ s' : RealMat (dblSp A), IsStdInv A μ s' ∧ ∀ τ, s' τ * compAt θ τ (mt (dblSp A) (p : GA D (dblSp A))⁻¹) = compAt θ τ (mt (dblSp A) (p : GA D (dblSp A))⁻¹) * sS τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactCommutingCartanInvolutionMovedStandardStatement

theorem **_root_.GlobalTheory.T.compactCommuting_cartanInvolution_moved_standard** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∀ (a : GL (Fin A.rank) (AE D)) (p : siegel A), IsConj A a κ p → ∀ θ : InfFam D, IsInfFam θ → ∀ sS : RealMat (dblSp A), (∀ τ, IsCartanInv (gramAt D (dblSp A) τ) (sS τ)) → (∀ k ∈ Khat A a, ∀ τ, compAt θ τ (mt (dblSp A) k) * sS τ = sS τ * compAt θ τ (mt (dblSp A) k)) → ∃ s' : RealMat (dblSp A), IsStdInv A μ s' ∧ ∀ τ, s' τ * compAt θ τ (mt (dblSp A) (p : GA D (dblSp A))⁻¹) = compAt θ τ (mt (dblSp A) (p : GA D (dblSp A))⁻¹) * sS τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactUnitaryFiniteInvariantSubspace

theorem **continuous_rankOne_self** : Continuous fun w : H => rankOne ℂ w w

theorem **isCompactOperator_rankOne_self** (w : H) : IsCompactOperator (rankOne ℂ w w)

theorem **convex_compactOperators** : Convex ℝ {f : H →L[ℂ] H | IsCompactOperator f}

theorem **isCompactOperator_integral** [IsProbabilityMeasure μ] {F : X → H →L[ℂ] H} (hFi : Integrable F μ) (hF : ∀ k, IsCompactOperator (F k)) : IsCompactOperator (∫ k, F k ∂μ : H →L[ℂ] H)

theorem **inner_integral_rankOne** {w : X → H} (hFi : Integrable (fun k => rankOne ℂ (w k) (w k)) μ) (x y : H) : ⟪y, (∫ k, rankOne ℂ (w k) (w k) ∂μ : H →L[ℂ] H) x⟫_ℂ = ∫ k, ⟪w k, x⟫_ℂ * ⟪y, w k⟫_ℂ ∂μ

theorem **isSymmetric_integral_rankOne** {w : X → H} (hFi : Integrable (fun k => rankOne ℂ (w k) (w k)) μ) : (∫ k, rankOne ℂ (w k) (w k) ∂μ : H →L[ℂ] H).IsSymmetric

theorem **integral_rankOne_ne_zero** {w : X → H} (hFi : Integrable (fun k => rankOne ℂ (w k) (w k)) μ) (v : H) (hpos : 0 < ∫ k, ‖⟪v, w k⟫_ℂ‖ ^ 2 ∂μ) : (∫ k, rankOne ℂ (w k) (w k) ∂μ : H →L[ℂ] H) ≠ 0

theorem **exists_finiteDimensional_invariant** : ∀ (K : Type) [Group K] [TopologicalSpace K] [IsTopologicalGroup K] [CompactSpace K] [T2Space K] [SecondCountableTopology K] (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] [Nontrivial H] (π : K →* (H →L[ℂ] H)), (∀ k x, ‖π k x‖ = ‖x‖) → (∀ x, Continuous fun k => π k x) → ∃ S : Submodule ℂ H, S ≠ ⊥ ∧ FiniteDimensional ℂ S ∧ ∀ k, ∀ x ∈ S, π k x ∈ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompactUnitaryFiniteInvariantSubspaceStatement

theorem **_root_.GlobalTheory.T.compact_unitary_finite_invariant_subspace** : ∀ (K : Type) [Group K] [TopologicalSpace K] [IsTopologicalGroup K] [CompactSpace K] [T2Space K] [SecondCountableTopology K] (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] [Nontrivial H] (π : K →* (H →L[ℂ] H)), (∀ k x, ‖π k x‖ = ‖x‖) → (∀ x, Continuous fun k => π k x) → ∃ S : Submodule ℂ H, S ≠ ⊥ ∧ FiniteDimensional ℂ S ∧ ∀ k, ∀ x ∈ S, π k x ∈ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletedHomLatticeStableSplitIdempotent

theorem **finIncl_tmul** (β : Af D.host.F) (x : D.host.E) : finIncl D (β ⊗ₜ[D.host.F] x) = pairA D.host.F 0 β ⊗ₜ[D.host.F] x

theorem **infPart_finIncl** (y : Af D.host.F ⊗[D.host.F] D.host.E) : infPart D (finIncl D y) = 0

theorem **finPart_finIncl** (y : Af D.host.F ⊗[D.host.F] D.host.E) : finPart D (finIncl D y) = y

theorem **eq_finIncl** {z : AE D} (hz : infPart D z = 0) : z = finIncl D (finPart D z)

theorem **finIncl_mul_mem** {S T : HSp D} (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T) (v : GlobalTheory.FinPl.Pl D) {α : Af D.host.F} (hα : α ∈ intAf D.host.F) (hsupp : ∀ w, ¬ w = v → evalV D.host.F w α = 0) {a : D.host.E} (ha : a ∈ D.host.AA) {s : Fin S.rank → D.host.E} (hs : s ∈ LS) {t : Fin T.rank → D.host.E} (ht : t ∈ LT) {y : Af D.host.F ⊗[D.host.F] D.host.E} (hy : y ∈ intAE D) : (Matrix.of fun i j => finIncl D y * (pairA D.host.F 0 α ⊗ₜ[D.host.F] (a * t i * ∑ k, star (s k) * S.gram k j)) : WO S T) ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = v)

theorem **idem_latO** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T) (u : WO S T), u ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v) → (Matrix.of fun i j => q.e * u i j : WO S T) ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v) ∧ ∀ i j, (q.e + cA D q.e) * u i j = u i j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletedHomLatticeStableSplitIdempotentStatement

theorem **_root_.GlobalTheory.T.completed_hom_lattice_stable_split_idempotent** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T) (u : WO S T), u ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v) → (Matrix.of fun i j => q.e * u i j : WO S T) ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v) ∧ ∀ i j, (q.e + cA D q.e) * u i j = u i j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionChainLatticeEntriesAgree

theorem **typeFrame_bas_val** (ϖ : kˣ) (ord : ι ≃ n ⊕ Fin c) : (typeFrame ϖ ord).bas.val = diagonal (fun y : n ⊕ Fin c => (ϖ : k) ^ ordExp y)

theorem **typeFrame_bas_inv_val** (ϖ : kˣ) (ord : ι ≃ n ⊕ Fin c) : ((typeFrame ϖ ord).bas⁻¹).val = diagonal (fun y : n ⊕ Fin c => (ϖ : k) ^ (-ordExp y))

theorem **typeFrame_coords_apply** (ϖ : kˣ) (ord : ι ≃ n ⊕ Fin c) (ord' : ι' ≃ n' ⊕ Fin c') (x : Matrix (n ⊕ Fin c) (n' ⊕ Fin c') k) (a : ι) (b : ι') : (typeFrame ϖ ord).coords (typeFrame ϖ ord') x a b = (ϖ : k) ^ (ordExp (ord' b) - ordExp (ord a)) * x (ord a) (ord' b)

theorem **fIdx_of_lt** {κ c n v : ℕ} (h : v < κ) : LocE.fIdx κ c n v = v

theorem **fIdx_of_mid** {κ c n v : ℕ} (h₁ : κ ≤ v) (h₂ : v < κ + c) : LocE.fIdx κ c n v = n + (v - κ)

theorem **fIdx_of_ge** {κ c n v : ℕ} (h : κ + c ≤ v) : LocE.fIdx κ c n v = v - c

theorem **scE_of_lt** {n v : ℕ} (ex : ℕ → ℕ) (h : v < n) : LocE.scE n ex v = 0

theorem **scE_add** (n i : ℕ) (ex : ℕ → ℕ) : LocE.scE n ex (n + i) = ex i

theorem **stdOrd_place** {κ τ j c l n N : ℕ} (h : j + c = l) (hn : n = κ + j * (τ + 1)) (e : XIx (Fin κ) (Fin τ) j ⊕ Fin c ≃ Fin N) (he : IsOrd (yPos xPos n) e) (a : Ix (Fin κ ⊕ Fin l) (Fin τ) j) : (e (stdOrd (Fin κ) (Fin τ) j c h a)).val = LocE.fIdx κ c n (zPos a) ∧ ordExp (stdOrd (Fin κ) (Fin τ) j c h a) = LocE.scE n (fun i => i + 1) (LocE.fIdx κ c n (zPos a))

theorem **atN_val** {N : ℕ} (v : Fin N → E) (i : Fin N) : Loc.atN v i.val = v i

theorem **eN_val** {N : ℕ} (i : Fin N) : Loc.eN E N i.val = Pi.single i 1

theorem **typeFrame_coords_toLoc** (ι : E →+* k) (q : ℕ) (ϖ : kˣ) (hϖ : (ϖ : k) = ι (q : E)) {κ τ l j c j' c' n N n' N' : ℕ} (h : j + c = l) (h' : j' + c' = l) (hn : n = κ + j * (τ + 1)) (hn' : n' = κ + j' * (τ + 1)) (e : XIx (Fin κ) (Fin τ) j ⊕ Fin c ≃ Fin N) (e' : XIx (Fin κ) (Fin τ) j' ⊕ Fin c' ≃ Fin N') (he : IsOrd (yPos xPos n) e) (he' : IsOrd (yPos xPos n') e') (x : (Fin N' → E) →ₗ[E] (Fin N → E)) (a : Ix (Fin κ ⊕ Fin l) (Fin τ) j) (b : Ix (Fin κ ⊕ Fin l) (Fin τ) j') : (typeFrame ϖ (stdOrd (Fin κ) (Fin τ) j c h)).coords (typeFrame ϖ (stdOrd (Fin κ) (Fin τ) j' c' h')) (toLoc ι e e' x) a b = ι (LocE.entB q κ c' n' c n (fun i => i + 1) (fun i => i + 1) x (zPos a) (zPos b))

theorem **entries_agree** : ∀ (E K k : Type) [Field E] [NumberField E] [StarRing E] [Field K] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K) (χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (s κ : ℕ) (S : SymData k s κ), Comparison₂ P q ι σ₀ ψ₀ χ S → ∀ (τ j c : ℕ) (h : j + c = s) (n' c' n c₂ : ℕ), n' = κ + j * (τ + 1) → n = κ + (j + 1) * (τ + 1) → c' = c + 1 → c₂ = c → ∀ (e' : XIx (Fin κ) (Fin τ) j ⊕ Fin (c + 1) ≃ Fin (n' + c')) (e : XIx (Fin κ) (Fin τ) (j + 1) ⊕ Fin c ≃ Fin (n + c₂)), IsOrd (yPos xPos n') e' → IsOrd (yPos xPos n) e → ∀ (x : (Fin (n' + c') → E) →ₗ[E] (Fin (n + c₂) → E)) (a : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) (j + 1)) (b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) j), (typeFrame S.ϖ (stdOrd (Fin κ) (Fin τ) (j + 1) c (by omega))).coords (typeFrame S.ϖ (stdOrd (Fin κ) (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionChainLatticeEntriesAgreeStatement

theorem **_root_.GlobalTheory.T.completionChain_lattice_entries_agree** : ∀ (E K k : Type) [Field E] [NumberField E] [StarRing E] [Field K] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K) (χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (s κ : ℕ) (S : SymData k s κ), Comparison₂ P q ι σ₀ ψ₀ χ S → ∀ (τ j c : ℕ) (h : j + c = s) (n' c' n c₂ : ℕ), n' = κ + j * (τ + 1) → n = κ + (j + 1) * (τ + 1) → c' = c + 1 → c₂ = c → ∀ (e' : XIx (Fin κ) (Fin τ) j ⊕ Fin (c + 1) ≃ Fin (n' + c')) (e : XIx (Fin κ) (Fin τ) (j + 1) ⊕ Fin c ≃ Fin (n + c₂)), IsOrd (yPos xPos n') e' → IsOrd (yPos xPos n) e → ∀ (x : (Fin (n' + c') → E) →ₗ[E] (Fin (n + c₂) → E)) (a : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) (j + 1)) (b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) j), (typeFrame S.ϖ (stdOrd (Fin κ) (Fin τ) (j + 1) c (by om

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionMatrixBoundedRationalAutIntegral

theorem **isGL_int_of_bdd** {X : Fields} {Pr : Primes X} {R : Ranks} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (ϖ : kˣ) (P : HeightOneSpectrum (𝒪 X.E)) (hP : P ∉ Pr.vGs) (hc : GlobalTheory.SymBridge.IsCompl P (qOf P) ι ϖ) (j c : ℕ) (a : AutY X R j) (g : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin c) k) (hg : IsGL ι a g) (hb : ∀ x ∈ lamYat (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionMatrixBoundedRationalAutIntegralStatement

theorem **_root_.GlobalTheory.T.completionMatrix_boundedRationalAut_integral** : ∀ {X : Fields} {Pr : Primes X} {R : Ranks} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (ϖ : kˣ) (P : HeightOneSpectrum (𝒪 X.E)), P ∉ Pr.vGs → GlobalTheory.SymBridge.IsCompl P (qOf P) ι ϖ → ∀ (j c : ℕ) (a : AutY X R j) (g : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin c) k), IsGL ι a g → (∀ x ∈ lamYat (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionRepresentativeMemPreviousIndexSet

theorem **exists_rep_flat_ne** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionRepresentativeMemPreviousIndexSetStatement

theorem **_root_.GlobalTheory.T.completion_representative_mem_previous_index_set** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionRepresentativesNormalFormModUnipotent

theorem **val_unif_of** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ϖ : kˣ} (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) : valuation k (ϖ : k) = uniformizer k

theorem **isGL_one** {X : Fields} {R : Ranks} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : X.E →+* k} {j c : ℕ} {g : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin c) k} (hg : IsGL ι (1 : AutY X R j) g) : g = 1

theorem **exists_normal_rep** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionRepresentativesNormalFormModUnipotentStatement

theorem **_root_.GlobalTheory.T.completion_representatives_normal_form_mod_unipotent** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionSmallIntegersInPrimeDense

theorem **small_in_prime_and_dense** : ∀ (E : Type) [Field E] [NumberField E] (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (PP : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (ϖ : kˣ), GlobalTheory.SymBridge.IsCompl PP q ι ϖ → GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → (∀ z : 𝒪 E, valuation k (ι z) < 1 → z ∈ PP.asIdeal) ∧ ∀ y : k, y ∈ 𝒪[k] → ∃ z : 𝒪 E, valuation k (y − ι z) < 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionSmallIntegersInPrimeDenseStatement

theorem **_root_.GlobalTheory.T.completion_smallIntegers_in_prime_dense** : ∀ (E : Type) [Field E] [NumberField E] (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (PP : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (ϖ : kˣ),

GlobalTheory.SymBridge.IsCompl PP q ι ϖ → GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → (∀ z : 𝒪 E, valuation k (ι z) < 1 → z ∈ PP.asIdeal) ∧ ∀ y : k, y ∈ 𝒪[k] → ∃ z : 𝒪 E, valuation k (y - ι z) < 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionTensorRelativeBasisCoordsUnique

theorem **coords_unique_of_field** {K E A : Type*} [Field K] [Ring E] [Nontrivial E] [Algebra K E] [AddCommGroup A] [Module K A] (w : E) (hw : ∀ f : K, algebraMap K E f ≠ w) (a b : A) (h : a ⊗ₜ[K] (1 : E) + b ⊗ₜ[K] w = 0) : a = 0 ∧ b = 0

theorem **coords_unique** : ∀ (D : D7) (v : Pl D) (w : D.host.E), (∀ f : D.host.F, algebraMap D.host.F D.host.E f ≠ w) → ∀ a b : Fv D.host.F v, a ⊗ₜ[D.host.F] (1 : D.host.E) + b ⊗ₜ[D.host.F] w = 0 → a = 0 ∧ b = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CompletionTensorRelativeBasisCoordsUniqueStatement

theorem **_root_.GlobalTheory.T.completion_tensor_relativeBasis_coords_unique** : ∀ (D : D7) (v : Pl D) (w : D.host.E), (∀ f : D.host.F, algebraMap D.host.F D.host.E f ≠ w) → ∀ a b : Fv D.host.F v, a ⊗ₜ[D.host.F] (1 : D.host.E) + b ⊗ₜ[D.host.F] w = 0 → a = 0 ∧ b = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CongruenceUnipotentMatrixReduction

theorem **mul_invariant** {a b : Type} [Fintype a] [DecidableEq a] {f : Matrix a b k → ℂ} (hs : ∀ x, f x ≠ 0 → IntM x) (hp : ∀ x y, IntM x → MaxM y → f (x + y) = f x) {g : Matrix a a k} (hg : MaxM (g - 1)) (x : Matrix a b k) : f (g ∗ x) = f x

theorem **hat_eq_zero** [MeasurableSpace k] [BorelSpace k] {n : Type} [Fintype n] [DecidableEq n] {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {ε : k} (hε : valuation k ε = 1) (y₀ : n → k) {η : n → k} (hη : ¬ ∀ i, η i ∈ 𝒪[k]) : hat ψ ν ε (cosetInd (box n (1 : kˣ)) (Submodule.Quotient.mk y₀)) η = 0

theorem **phiTop_eq** (z : Matrix (Ix (Fin κ ⊕ Fin (s + 1)) t (s + 2)) (XIx (Fin κ) t (s + 1) ⊕ Fin 0) k) : (S.T t).PhiTop z = (S.P.iter t S.μ S.tame (s + 1)).Phi t (S.μ (s + 1 + 1)) (z.submatrix (ixSucc _ t (s + 1)) (stdOrd (Fin κ) t (s + 1) 0 rfl))

theorem **symmetricChain_psi_base_eq_one_of_maximalIdeal** {d : XIx (Fin κ) t (s + 1) → k} (hd : ∀ i, valuation k (d i) < 1) : S.ψ (-1 ∗ (d ·ᵥ S.y0 t)) = 1

theorem **xiTop_mul_congr** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t] (g : Matrix (Ix (Fin κ ⊕ Fin (s + 1)) t (s + 2) ⊕ Unit) (Ix (Fin κ ⊕ Fin (s + 1)) t (s + 2) ⊕ Unit) k), (∀ r c, valuation k ((g - 1) r c) < 1) → ∀ x : Matrix (Ix (Fin κ ⊕ Fin (s + 1)) t (s + 2) ⊕ Unit) (XIx (Fin κ) t (s + 1)) k, (S.T t).xiTop (box _ (1 : kˣ)) S.ψ ν (-1) (Submodule.Quotient.mk (S.y0 t)) (g ∗ x) = (S.T t).xiTop (box _ (1 : kˣ)) S.ψ ν (-1) (Submodule.Quotient.mk (S.y0 t)) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjBesselCharacterIsUnfoldingThetaFunctionOfGra

theorem **exists_psiE_ne_one** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∃ s : AE D, psiE ψ s ≠ 1

theorem **vecB_norm_ne** : ∀ (D : D7) (V₀ A L : HSp D) (o : OrthDec V₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → star (vecB o ζ) ·ᵥ V₀.gram ∗ᵥ vecB o ζ ≠ 0

theorem **psi_qS_eq_one** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (W : HSp D) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E), gramQ W x₀ = 0 → ∀ c : skewA D ρ, ψ (d.qS W c (rat x₀)) = 1

theorem **conjBessel_isTh** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ : ℕ) (B : BesLev D ρ) (W : HSp D) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E), gramQ W x₀ = 0 → IsTh ψ B.d W x₀ fun n => star (psiBes ψ B.d 1 (vecB B.o B.ζ) n)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjBesselCharacterIsUnfoldingThetaFunctionOfGraStatement

theorem **_root_.GlobalTheory.T.conjBesselCharacter_isUnfoldingThetaFunction_of_gram_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ : ℕ) (B : BesLev D ρ) (W : HSp D) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E), gramQ W x₀ = 0 → IsTh ψ B.d W x₀ fun n => star (psiBes ψ B.d 1 (vecB B.o B.ζ) n)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjByColumnUnipotentPreservesUnipotentHaar

theorem **colE_inv_val_eq_zero** (ρ : ℕ) (β : n → k) (t s : n ⊕ Fin c) (h : lev (ρ + 1) t ≠ lev (ρ + 1) s) : (colE c (ρ + 1) β)⁻¹.val t s = 0

theorem **colE_uni** (ρ : ℕ) (β : n → k) : ∀ r s, (Sum.elim (fun _ => 0) (fun _ => 1) : n ⊕ Fin c → ℕ) s ≤ Sum.elim (fun _ => 0) (fun _ => 1) r → (colE c (ρ + 1) β).val r s = (1 : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) r s

theorem **colE_inv_uni** (ρ : ℕ) (β : n → k) : ∀ r s, (Sum.elim (fun _ => 0) (fun _ => 1) : n ⊕ Fin c → ℕ) s ≤ Sum.elim (fun _ => 0) (fun _ => 1) r → (colE c (ρ + 1) β)⁻¹.val r s = (1 : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) r s

theorem **extendAboveDiagonalEntries_mul_blockRespecting** (p : ℕ) (h : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) (hh : ∀ t s, lev p t ≠ lev p s → h t s = 0) (e : Ix n c p → k) : ext0 (Up n c p) e ∗ h = ext0 (Up n c p) ((rm (Up n c p) h).mulVec e)

theorem **cjE_mkNel** (ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (β : n → k) (e : Ix n c (ρ + 1) → k) : cjE ρ h1 h2 β (mkNel (ρ + 1) e) = mkNel (ρ + 1) ((lm (Up n c (ρ + 1)) (colE c (ρ + 1) β).val ∗ rm (Up n c (ρ + 1)) (colE c (ρ + 1) β)⁻¹.val).mulVec e)

theorem **det_L** (ρ : ℕ) (β : n → k) : (lm (Up n c (ρ + 1)) (colE c (ρ + 1) β).val ∗ rm (Up n c (ρ + 1)) (colE c (ρ + 1) β)⁻¹.val).det = 1

theorem **GlobalTheory.HaarR.map_volN_cjE** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (ν : Measure k), ν.IsAddHaarMeasure → ∀ β : n → k, (volN ν n c (ρ + 1)).map (cjE ρ h1 h2 β) = volN ν n c (ρ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjByColumnUnipotentPreservesUnipotentHaarStatement

theorem **_root_.GlobalTheory.T.conjByColumnUnipotent_preserves_unipotentHaar** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (ν : Measure k), ν.IsAddHaarMeasure → ∀ β : n → k, (volN ν n c (ρ + 1)).map (cjE ρ h1 h2 β) = volN ν n c (ρ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjDoublingIsotropicUnipotentMemSiegel

theorem **exists_leftInv_of_linInd** {K : Type} [Field K] {a b : ℕ} (C : Matrix (Fin a) (Fin b) K) (hC : LinearIndependent K C.transpose) : ∃ L : Matrix (Fin b) (Fin a) K, L ∗ C = 1

theorem **conj_io_gen_mem** (V : HSp D) (γ : GA D (dblSp V)) (hγ : γ ∈ GQ (dblSp V)) (γi : Matrix (Fin (V.rank + V.rank)) (Fin (V.rank + V.rank)) D.host.E) (hγi : (Formalism.L2.Theta.matHom (dblSp V) γ⁻¹).1 = γi.map (incA D)) (r : ℕ) (K : Matrix (Fin V.rank) (Fin r) D.host.E) (hK : diagSnd γi ∗ K = 0) (hC : IsIsoFrame V (leviMat finSumFinEquiv γi ∗ K)) (g : GA D V) (h1 : HMul.hMul ((Formalism.L2.Theta.matHom V g).1 - 1) ((leviMat finSumFinEquiv γi ∗ K).map (incA D)) = 0) (h3 : ∀ x : Fin V.rank → D.host.E, (Formalism.UG.cT (D.sp.cm.sig : D.host.E →+* D.host.E) (leviMat finSumFinEquiv γi ∗ K) ∗ V.gram).mulVec x = 0 → ∃ a : Fin r → AE D,

((Formalism.L2.Theta.matHom V g).1 − 1).mulVec (fun i => incA D (x i)) = ((leviMat finSumFinEquiv γi ∗ K).map (incA D)).mulVec a) : ∃ h : γ ∗ io V (g, 1) ∗ γ⁻¹ ∈ siegel V, aP V (_, h) = 1

theorem **conj_io_unip_siegel** : ∀ (D : D7) (V : HSp D), ∀ γ ∈ GQ (dblSp V), ∀ γi : Matrix (Fin (V.rank + V.rank)) (Fin (V.rank + V.rank)) D.host.E, (matHom (dblSp V) γ⁻¹).1 = γi.map (incA D) → ∀ (r : ℕ) (K : Matrix (Fin V.rank) (Fin r) D.host.E), diagSnd γi ∗ K = 0 → IsIsoFrame V (leviMat finSumFinEquiv γi ∗ K) → ∀ n ∈ NI V (leviMat finSumFinEquiv γi ∗ K), ∃ p : siegel V, γ ∗ io V (n, 1) = p ∗ γ ∧ aP V p = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjDoublingIsotropicUnipotentMemSiegelStatement

theorem **_root_.GlobalTheory.T.conj_doubling_isotropic_unipotent_mem_siegel** : ∀ (D : D7) (V : HSp D), ∀ γ ∈ GQ (dblSp V), ∀ γi : Matrix (Fin (V.rank + V.rank)) (Fin (V.rank + V.rank)) D.host.E, (matHom (dblSp V) γ⁻¹).1 = γi.map (incA D) → ∀ (r : ℕ) (K : Matrix (Fin V.rank) (Fin r) D.host.E), diagSnd γi ∗ K = 0 → IsIsoFrame V (leviMat finSumFinEquiv γi ∗ K) → ∀ n ∈ NI V (leviMat finSumFinEquiv γi ∗ K), ∃ p : siegel V, γ ∗ io V (n, 1) = p ∗ γ ∧ aP V p = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjFixedIntegralIsUnitOrIdemSubUnit

theorem **evalV_single_same** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v) : evalV F v (singleV F v y) = y

theorem **evalV_single_ne** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v) {w : HeightOneSpectrum (𝒪 F)} (h : w ≠ v) : evalV F w (singleV F v y) = 0

theorem **integral_compFin** (v : Pl D) {z : Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ intAE D) : (intIn D v).IsIntegralElem (GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D v z)

theorem **mem_int_of_integral** (v : Pl D) (a : Fv D.host.F v) (h : (intIn D v).IsIntegralElem (a ⊗ₜ[D.host.F] (1 : D.host.E))) : a ∈ v.adicCompletionIntegers D.host.F

theorem **exists_inv** (v : Pl D) {x : AE D} (hx : x ∈ idAt D (· = v)) (u : (v.adicCompletionIntegers D.host.F)ˣ) (hu : compAt D v x = ((u : v.adicCompletionIntegers D.host.F) : Fv D.host.F v) ⊗ₜ[D.host.F] (1 : D.host.E)) : ∃ y ∈ intAt D (· = v), x ∗ y = idem D (· = v)

theorem **fixed_unit_or** : ∀ (D : D7) (v : Pl D) (ε : AE D), Tame D v → ε ∗ ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) → ∀ x ∈ intAt D (· = v), cA D x = x → (∃ y ∈ intAt D (· = v), x ∗ y = ε) ∨ (∃ y ∈ intAt D (· = v), (ε − x) ∗ y = ε)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjFixedIntegralIsUnitOrIdemSubUnitStatement

theorem **_root_.GlobalTheory.T.conjFixed_integral_isUnit_or_idem_sub_unit** : ∀ (D : D7) (v : Pl D) (ε : AE D), Tame D v → ε ∗ ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) → ∀ x ∈ intAt D (· = v), cA D x = x → (∃ y ∈ intAt D (· = v), x ∗ y = ε) ∨ (∃ y ∈ intAt D (· = v), (ε − x) ∗ y = ε)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjIntertwinerPointValueSobolevContinuous

theorem **reg_star** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (g : X.B.I.GA) (φ : L2X X) : X.B.R g (star φ) = star (X.B.R g φ)

theorem **iotaX_rat_iff** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (g : GA D (D.sp.Y 0)) : g ∈ GQ (D.sp.Y 0) ↔ iotaX X h g ∈ X.B.I.GFset

theorem **exists_conj_multilinear** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) {S : D.what₂.Tau → SPar} (M : DSModel (D.sp.Y 0) S) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)) (Φ : X.B.Hs →L[ℂ] L2X X) : ∃ Ψ : ContinuousMultilinearMap ℂ M.H (L2X X), ∀ y, Ψ y = star (Φ (X.B.tens (Equiv.piCongrLeft X.B.H X.pl.symm fun τ => c τ (y τ))))

theorem **conjMap_cont** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 ∗ (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowMult.vecC X M c u))) ψ) → ∀ (τ : D.what₂.Tau)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjIntertwinerPointValueSobolevContinuousStatement

theorem **_root_.GlobalTheory.T.conjIntertwiner_pointValue_sobolevContinuous** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 ∗ (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheor

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjPEqCP

theorem **cP_cP** (P : HeightOneSpectrum (𝒪 D.host.E)) : cP (cP P) = P

theorem **conjP_eq_cP** (Q : HeightOneSpectrum (𝒪 D.host.E)) : Formalism.L2.Col.conjPrime₂ Q = cP Q

theorem **delta_mem** {Φ : AE D →+∗ FiniteAdeleRing (𝒪 D.host.E) D.host.E} {P : HeightOneSpectrum (𝒪 D.host.E)} {x : AE D} (h : IsDelta Φ P x) (Q : HeightOneSpectrum (𝒪 D.host.E)) : Φ x Q ∈ Q.adicCompletionIntegers D.host.E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjPNotMemOfIdemSum

theorem **conjP_notMem_of_isIdemSum** {D : D7} {e : AE D} {SS : Finset (Formalism.L2.DIV.Pl D.host.E)} (he : IsIdemSum e SS) : ∀ P ∈ SS, Formalism.L2.Col.conjPrime₂ P ∉ SS

theorem **natCast_smul_inv_smul** {E : Type} [Field E] [NumberField E] {a b : ℕ} (c : ℕ) (hc : 0 < c) (y : Formalism.L2.Col.HomE E a b) : ((c : 𝒪 E)) • ((c : E)⁻¹ • y) = y

theorem **exists_rep_add_mem** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ c : ℕ, 0 < c → ∀ (SX : Finset (VV D.what₂.Tau S.rank T.rank)), Formalism.L2.Th1.H.IsReps (LamC (tauE D) δ (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') (AC (tauE

D) δ (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) SS N N' c) SX → ∀ (px : VV D.what₂.Tau S.rank T.rank → Formalism.L2.Col.HomE D.host.E S.rank T.rank), (∀ x ∈ SX, px x ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c ∧ embH (tauE D) δ (px x) = x) → ∀ g : Formalism.L2.Col.HomE D.host.E S.rank T.rank, (∀ P ∈ SS, (c : D.host.E) • g ∈ Formali

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjPlEqCP

theorem **ev_sum** {ι : Type} (s : Finset ι) (f : ι → Formalism.L2.Theta.AE D) (Q : HeightOneSpectrum (𝒪 D.host.E)) : Φ (∑ i ∈ s, f i) Q = ∑ i ∈ s, Φ (f i) Q

theorem **conjPl_eq_cP** (P : HeightOneSpectrum (𝒪 D.host.E)) : Formalism.DivisionValues.conjPl P = cP P

theorem **under_injOn** {P : Finset (Formalism.L2.DIV.Pl D.host.E)} {e : AE D} (h : Formalism.IndefR.Datum.IsIdemSum e P) : Set.InjOn (fun PP : Formalism.L2.DIV.Pl D.host.E => PP.under (𝒪 D.host.F)) (P : Set (Formalism.L2.DIV.Pl D.host.E))

theorem **injOn_under** : ∀ (D : D7) (c : GlobalTheory.ChainOnto.Com D), c.Is → Set.InjOn (fun PP : Formalism.L2.DIV.Pl D.host.E => PP.under (NumberField.RingOfIntegers D.host.F)) (c.Pg : Set (Formalism.L2.DIV.Pl D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjRowEqTranslatedEvaluationCuspFormMap

theorem **see_onto** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (v₀ : M.Vec), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → ConjVec X h θ M v₀ → ∀ φ : L2X X, star φ ∈ HHX X → ∃ T ∈ MultT θ M J χ, (φ : X.B.I.GA → ℂ) =ᵐ[muX X] evS X h t (T (PiTensorProduct.tprod ℂ v₀))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjRowEqTranslatedEvaluationCuspFormMapStatement

theorem **_root_.GlobalTheory.T.conj_row_eq_translated_evaluation_cusp_form_map** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (v₀ : M.Vec), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → ConjVec X h θ M v₀ → ∀ φ : L2X X, star φ ∈ HHX X → ∃ T ∈ MultT θ M J χ, (φ : X.B.I.GA → ℂ) =ᵐ[muX X] evS X h t (T (PiTensorProduct.tprod ℂ v₀))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjSplitLocalModelBySupportedPair

theorem **twist_apply** (c : ℂ) (gk : GL Y' k) (hk : GL Y k) (Φ : SB (Matrix Y Y' k)) (x) : (twist c gk hk Φ).1 x = c * Φ.1 ((hk⁻¹).val * x * gk.val)

theorem **mt_inv_mul'** {S : HSp D} (g : GA D S) : mt S g⁻¹ * mt S g = 1

theorem **actO_up** {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {g : GA D S} {gk : GL Y' k} {h : GA D T} {hk : GL Y k} (hg : p.IsLoc FS g gk) (hh : p.IsLoc FT h hk) (x : Matrix Y Y' k) : actO S T g h (p.up FS FT x) = p.up FS FT (hk.val * x * (gk⁻¹).val)

theorem **conj_isModel** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (ω : Om S T), IsWeilDatumO ψ S T χS χT ω → ∀ (μS μT : kˣ →* ℂˣ) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)), p.IsModel ψ FS FT Tm → ∀ (g : GA D S) (gk : GL Y' k) (h : GA D T) (hk : GL Y k), p.IsLoc FS g gk → p.IsLoc FT h hk → ∃ T₂ : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ), p.IsModel ψ FS FT T₂ ∧ ∀ Φ Φ', Φ'.1 = (((μS * μT⁻¹) (Matrix.GeneralLinearGroup.det gk) : ℂˣ) : ℂ) • om μS⁻¹ gk hk Φ.1 → T₂ Φ = ω (g⁻¹, h⁻¹) (Tm Φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjSplitLocalModelBySupportedPairStatement

theorem **_root_.GlobalTheory.T.conj_splitLocalModel_by_supported_pair** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (ω : Om S T), IsWeilDatumO ψ S T χS χT ω → ∀ (μS μT : kˣ →* ℂˣ) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)), p.IsModel ψ FS FT Tm → ∀ (g : GA D S) (gk : GL Y' k) (h : GA D T) (hk : GL Y k), p.IsLoc FS g gk → p.IsLoc FT h hk → ∃ T₂ : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ), p.IsModel ψ FS FT T₂ ∧ ∀ Φ Φ' : SB (Matrix Y Y' k), Φ'.1 = (((μS * μT⁻¹) (Matrix.GeneralLinearGroup.det gk) : ℂˣ) : ℂ) • om μS⁻¹ gk hk Φ.1 → T₂ Φ = ω (g⁻¹, h⁻¹) (Tm Φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjSymmEntry

theorem **xPos_lt** {κ τ j : ℕ} (a : XIx (Fin κ) (Fin τ) j) : xPos a < κ + j * (τ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjTranslatedEvaluationMemAllRowsSpace

theorem **inner_antiunitary_swap** {E F : Type*} [NormedAddCommGroup E] [InnerProductSpace ℂ E] [NormedAddCommGroup F] [InnerProductSpace ℂ F] (U : E ≃ₗᵢ⋆[ℂ] F) (x y : E) : ⟪U x, U y⟫_ℂ = ⟪y, x⟫_ℂ

theorem **star_toLp_eq** {D : D7} (X : HidaSetting D) {f : X.B.I.GA → ℂ} (hm : MemLp f 2 (muX X)) (hs : MemLp (fun x => star (f x)) 2 (muX X)) : star (hm.toLp f) = hs.toLp (fun x => star (f x))

theorem **re_fix_of_row** {D : D7} (X : HidaSetting D) {f : X.B.I.GA → ℂ} (hf : RowFun X f) (hm : MemLp f 2 (muX X)) : X.B.Re (star (hm.toLp f)) = star (hm.toLp f)

theorem **real_shift** {D : D7} (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) {S : D.what₂.Tau → SPar} (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (ht : t ∈ GfC (D.sp.Y 0)) (hR : RealSplit X h θ) (v₀ : M.Vec) (g : X.B.I.Ginf) : ∃ a : D.what₂.Tau → Matrix (Fin (D.sp.Y 0).rank) (Fin (D.sp.Y 0).rank) ℂ, (∀ τ, a τ ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ)) ∧ (∀ τ, (iotaX X h (atTau θ (D.sp.Y 0) τ (a τ))).1 (X.pl.symm τ) = g (X.pl.symm τ)) ∧ ∃ w : M.Vec, (∀ τ, (w τ : M.H τ) = M.rep τ (a τ) (v₀ τ : M.H τ)) ∧ ∀ U ∈ Mult θ M J χ, ∀ x, U v₀ ((iotaX X h).symm (x * (g, 1)) * t) = U w ((iotaX X h).symm x * t)

theorem **coef_eq** {D : D7} (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) {S : D.what₂.Tau → SPar} (M : DSModel (D.sp.Y 0) S) (v₀ : M.Vec) (hCV : ConjVec X h θ M v₀) (g : X.B.I.Ginf) (a : D.what₂.Tau → Matrix (Fin (D.sp.Y 0).rank) (Fin (D.sp.Y 0).rank) ℂ) (ha : ∀ τ, a τ ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ)) (hb : ∀ τ, (iotaX X h (atTau θ (D.sp.Y 0) τ (a τ))).1 (X.pl.symm τ) = g (X.pl.symm τ)) (w : M.Vec) (hw : ∀ τ, (w τ : M.H τ) = M.rep τ (a τ) (v₀ τ : M.H τ)) : ∏ τ, ⟪(w τ : M.H τ), (v₀ τ : M.H τ)⟫_ℂ = Formalism.L2.SG.ProjRows.mInf X.B.I X.B.σ X.B.w g

theorem **GlobalTheory.Core.Mi.SeeHH.conj_evS_mem_HHX** : GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (v₀ : M.Vec), S = parB (Ptw D) 0

→ IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → ConjVec X h θ M v₀ → ∀ T ∈ MultT θ M J χ, ∀ φ : L2X X, (φ : X.B.I.GA → ℂ) =ᵐ[muX X] evS X h t (T (PiTensorProduct.tprod ℂ v₀)) → star φ ∈ HHX X

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjTranslatedEvaluationMemAllRowsSpaceStatement

theorem **_root_.GlobalTheory.T.conj_translated_evaluation_mem_all_rows_space** : GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (v₀ : M.Vec), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → ConjVec X h θ M v₀ → ∀ T ∈ MultT θ M J χ, ∀ φ : L2X X, (φ : X.B.I.GA → ℂ) =ᵐ[muX X] evS X h t (T (PiTensorProduct.tprod ℂ v₀)) → star φ ∈ HHX X

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjWeilDatumViaAdjointExchangedPair

theorem **dagger_eq** (S T : HSp D) (x : WO S T) : dagger S T x = adjO T S x

theorem **cj_dagger** (S T : HSp D) (f : WO S T → ℂ) : cj (dagger T S) f = dag S T f

theorem **conjAdjoint_apply** {S T : HSp D} (om : Rep S T) (g : GA D T) (h : GA D S) (f : WO T S → ℂ) : conjAdjoint om (g, h) f = dag S T (om (h, g) (dag T S f))

theorem **chi_conj** {χ : (AE D)ˣ →* Circle} {n : ℕ} (hR : ResEta D χ n) (y : (AE D)ˣ) : χ (Units.map (cA D).toMonoidHom y) = (χ y)⁻¹

theorem **isSplitIdem_conj** {e : AE D} (h : IsSplitIdem D e) : IsSplitIdem D (cA D e)

theorem **suppAt_conj** {e : AE D} {V : HSp D} {g : GA D V} (h : SuppAt e V g) : SuppAt (cA D e) V g

theorem **det_inv_eq** (V : HSp D) (g : GA D V) : (mt V g⁻¹).det = cA D (mt V g).det

theorem **isDetAt_conj** {e : AE D} (he : e * e = e) {V : HSp D} {g : GA D V} {x : (AE D)ˣ} (hx : IsDetAt e V g x) : IsDetAt (cA D e) V g (Units.map (cA D).toMonoidHom x⁻¹)

theorem **isWeilDatumO_conjAdjoint** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → IsWeilDatumO ψ T S χT χS (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.conjAdjoint om)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjWeilDatumViaAdjointExchangedPairStatement

theorem **_root_.GlobalTheory.T.conj_weil_datum_via_adjoint_exchanged_pair** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → IsWeilDatumO ψ T S χT χS (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.conjAdjoint om)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateCompactMemLevel

theorem **mem_idAt_of_eq** {P : Pl D → Prop} {v : Pl D} (hv : P v) {a : AE D} (ha : a ∈ idAt D (· = v)) : a ∈ idAt D P

theorem **latV_supp_iff** (S : HSp D) (L : Lat S) (hL : Submodule.IsLattice D.host.E L) {P : Pl D → Prop} {v : Pl D} (hv : P v) (y : Fin S.rank → AE D) (hy : ∀ i, y i ∈ idAt D (· = v)) : y ∈ latV S L P ↔ y ∈ latV S L (· = v)

theorem **moves_at_of_moves** {S : HSp D} {L : Lat S} (hL : Submodule.IsLattice D.host.E L) {P : Pl D → Prop} {v : Pl D} (hv : P v) {g : GA D S} (h : Moves S g L L P) : Moves S g L L (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateEvaluationMapHasLevel

theorem **e_left** (k : X.B.I.Gf) (hk : k ∈ X.K2) (y : X.B.I.Gf) : X.B.e y = X.chi2 k * X.B.e (k * y)

theorem **R_comp_Re** (k : X.B.I.Gf) (hk : k ∈ X.K2) : (X.B.R (1, k)).comp X.B.Re = X.chi2 k • X.B.Re

theorem **conjMap_lev** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowMult.vecC X M c u))) ψ) → ∀ u : M.Vec, ∀ k ∈ J

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateEvaluationMapHasLevelStatement

theorem **_root_.GlobalTheory.T.conjugate_evaluation_map_has_level** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowM

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateEvaluationMapIntertwinesRealPlaces

theorem **conjMap_equiv** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowMult.vecC X M c u))) ψ) → ∀ τ : D.what₂.Tau,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateEvaluationMapIntertwinesRealPlacesStatement

theorem **_root_.GlobalTheory.T.conjugate_evaluation_map_intertwines_real_places** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ)

(uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateHalfAnnihilatorMemCompletedHomLattice

theorem **trace_rankOne** {k m n Y Y' : Type} [Field k] [Fintype m] [DecidableEq m] [Fintype n] [Fintype Y] [Fintype Y'] (A G : Matrix m m k) (hAG : G ∗ A = 1) (N : Matrix m n k) (B : Matrix n n k) (βT : Matrix n Y k) (βS : Matrix m Y' k) (γS : Matrix Y' m k) (hS : βS ∗ γS = 1) (c : Y → k) (x : m → k) : Matrix.trace (A ∗ N ∗ B ∗ (βT ∗ vecMulVec c ((x ᵥ∗ G) ᵥ∗ βS) ∗ γS)) = x ⬝ᵥ (N ∗ B) ∗ᵥ (βT ∗ᵥ c)

theorem **int_of_char** {ψk : AddChar k Circle} (h : HasConductor ψk 1) {b : k} (hb : ∀ a ∈ 𝒪[k], ψk (a ∗ b) = 1) : b ∈ 𝒪[k]

theorem **annihilator_mem_latO** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β ∗ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β ∗ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), IsStdChar D ψ (−1) → (ofSplit q).IsChar ψ ψk → HasConductor ψk 1 → ∀ y : WO S T, IsHalf (cA D q.e) y → (∀ x : Matrix Y Y' k, (∀ r s, x r s ∈ 𝒪[k]) → ψ (symplO S T ((ofSplit q).up FS FT x) y) = 1) → y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateHalfAnnihilatorMemCompletedHomLatticeStatement

theorem **_root_.GlobalTheory.T.conjugate_half_annihilator_mem_completed_hom_lattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β ∗ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β ∗ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), IsStdChar D ψ (−1) → (ofSplit q).IsChar ψ ψk → HasConductor ψk 1 → ∀ y : WO S T, IsHalf (cA D q.e) y → (∀ x : Matrix Y Y' k, (∀ r s, x r s ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateHalfLineMultipleMemIffIntegral

theorem **vecMul_jA_eq_zero** (p : SplitPt D k) {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) : (fun i => p.pr (incA D (star ((HSpace.gram S₀ ∗ᵥ vecB o ζ) i)))) ᵥ∗ o.jA.map p.pr = 0

theorem **sum_mul_star_eq** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) : ∑ i, vecB o ζ i ∗ star ((HSpace.gram S₀ ∗ᵥ vecB o ζ) i) = star ζ ⬝ᵥ HSpace.gram L ∗ᵥ ζ

theorem **sumFrame_dot** (p : SplitPt D k) {n : Type} [Fintype n] [DecidableEq n] {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n) (α₀ : Frame p S₀ (n ⊕ Fin 1)) (hsum : IsSumFrame o ζ α α₀) (x' : n ⊕ Fin 1 → k) : (α₀.b ∗ᵥ x') ⬝ᵥ (fun i => p.pr (incA D (star ((HSpace.gram S₀ ∗ᵥ vecB o ζ) i)))) = x' (Sum.inr 0) ∗ p.pr (incA D (star ζ ⬝ᵥ HSpace.gram L ∗ᵥ ζ))

theorem **pair_eq** (p : SplitPt D k) (S₀ : HSp D) (z : Fin S₀.rank → D.host.E) (c : AE D) (x : Fin S₀.rank → k) : ∑ i, ∑ j, x i ∗ p.pr (cA D (incA D (HSpace.gram S₀ i j))) ∗ (p.pr (cA D c) ∗ p.pr (incA D (star (z j)))) = p.pr (cA D c) ∗ (x ⬝ᵥ fun i => p.pr (incA D (star ((HSpace.gram S₀ ∗ᵥ z) i))))

theorem **conj_half** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n) (α₀ : Frame p S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : Submodule 𝒪[k] (n → k)), IsSelfDualC S₀ M → IsSumFrame o ζ α α₀ → α₀.IsLat M (lastKer Λ) → ∀ c : AE D, ((fun r => cA D p.e ∗ (c ∗ incA D (vecB o ζ r))) ∈ latV S₀ M (· = p.v) ↔ p.pr (cA D c) ∗ p.pr (incA D (star ζ ⬝ᵥ HSpace.gram L ∗ᵥ ζ)) ∈ 𝒪[k])

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateHalfLineMultipleMemIffIntegralStatement

theorem **_root_.GlobalTheory.T.conjugateHalf_lineMultiple_mem_iff_integral** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n) (α₀ : Frame p S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : Submodule 𝒪[k] (n → k)), IsSelfDualC S₀ M → IsSumFrame o ζ α α₀ → α₀.IsLat M (lastKer Λ) → ∀ c : AE D, ((fun r => cA D p.e ∗ (c ∗ incA D (vecB o ζ r))) ∈ latV S₀ M (· = p.v) ↔ p.pr (cA D c) ∗ p.pr (incA D (star ζ ⬝ᵥ HSpace.gram L ∗ᵥ ζ)) ∈ 𝒪[k])

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateHalfPointNegMultipliesByCharacter

theorem **e_mul_sec** (p : SplitPt D k) (a : k) : p.e ∗ p.sec a = p.sec a

theorem **pr_trace** (p : SplitPt D k) {m : ℕ} (M : Mx D m m) : p.pr M.trace = (M.map p.pr).trace

theorem **map_rat** (p : SplitPt D k) {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) : (rat M).map p.pr = M.map (iota p)

theorem **conj_half_modul** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (χ : AddChar k Circle), p.IsChar ψ χ → ∀ (n n'' : Type) [Fintype n] [DecidableEq n] [Fintype n''] [DecidableEq n''] (S₀ A L W₀ : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (β : Frame p W₀ n) (γ : Frame p A n'') (γ₀ : Frame p S₀ (n'' ⊕ Fin 1)), IsSumFrame o ζ γ γ₀ → ∀ t : n → k, IsHalf (cA D p.e) ((o.resT W₀).symm (0, −GlobalTheory.FinPl.LineAct.wCol p ζ β t)) ∧ ∀ x : WO W₀ S₀, IsHalf p.e x → ψ (symplO W₀ S₀ x ((o.resT W₀).symm (0, −GlobalTheory.FinPl.LineAct.wCol p ζ β t))) = (χ (−((fun j => rd β γ₀ x (Sum.inr 0) j) ⬝ᵥ t)) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateHalfPointNegMultipliesByCharacterStatement

theorem **_root_.GlobalTheory.T.conjugateHalfPoint_neg_multiplies_by_character** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (χ : AddChar k Circle), p.IsChar ψ χ → ∀ (n n'' : Type) [Fintype n] [DecidableEq n] [Fintype n''] [DecidableEq n''] (S₀ A L W₀ : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (β : Frame p W₀ n) (γ : Frame p A n'') (γ₀ : Frame p S₀ (n'' ⊕ Fin 1)), IsSumFrame o ζ γ γ₀ → ∀ t : n → k, IsHalf (cA D p.e) ((o.resT W₀).symm (0, −GlobalTheory.FinPl.LineAct.wCol p ζ β t)) ∧ ∀ x : WO W₀ S₀, IsHalf p.e x → ψ (symplO W₀ S₀ x ((o.resT W₀).symm (0, −GlobalTheory.FinPl.LineAct.wCol p ζ β t))) = (χ (−((fun j => rd β γ₀ x (Sum.inr 0) j) ⬝ᵥ t)) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateIdealLatticeDiscrOneHomotheticIdeal

theorem **conj_lattice_core** : ∀ (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) (M : IntermediateField ℚ ℂ) [FiniteDimensional ℚ M] (t : (M →ₐ[ℚ] ℂ) → ℂ), SignedLogMeans.HidaSetting D cc δ M t → ∀ ρ : M →ₐ[ℚ] ℂ, Formalism.MainNotions.TaylorLogMetric.discr (lamConj M ρ (lamOf D cc δ)) = 1 ∧ ∃ c : Ideal (𝒪 D.host.K), c ≠ ⊥ ∧ ∃ x : ℂ, x ≠ 0 ∧ lamConj M ρ (lamOf D cc δ) = lamOf D c x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateIdealLatticeDiscrOneHomotheticIdealStatement

theorem **_root_.GlobalTheory.T.conjugateIdealLattice_discr_one_homothetic_ideal** : ∀ (D : D7) (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) (M : IntermediateField ℚ ℂ) [FiniteDimensional ℚ M] (t : (M →ₐ[ℚ] ℂ) → ℂ), SignedLogMeans.HidaSetting D cc δ M t → ∀ ρ : M →ₐ[ℚ] ℂ, Formalism.MainNotions.TaylorLogMetric.discr (lamConj M ρ (lamOf D cc δ)) = 1 ∧ ∃ c : Ideal (𝒪 D.host.K), c ≠ ⊥ ∧ ∃ x : ℂ, x ≠ 0 ∧ lamConj M ρ (lamOf D cc δ) = lamOf D c x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateIntertwinerValuesIsCuspFun

theorem **mem_NI_iff_iota** : ∀ (D : D7) {r : ℕ} (C : Matrix (Fin (D.sp.Y 0).rank) (Fin r) D.host.E), IsIsoFrame (D.sp.Y 0) C → ∀ g : GA D (D.sp.Y 0), g ∈ NI (D.sp.Y 0) C ↔ Formalism.IndefR.RationalEmbedding.iota D g ∈ (Formalism.IndefR.RationalEmbedding.inst D).NA (Submodule.span (Formalism.IndefR.RationalEmbedding.inst D).E (Set.range (C : Matrix (Fin (Formalism.IndefR.RationalEmbedding.inst D).n) (Fin r) (Formalism.IndefR.RationalEmbedding.inst D).E).col))

theorem **isCuspFun_of_isCuspFn_iota** : ∀ (D : D7) (I : Formalism.MainNotions.S.AutInst) (hI : I = Formalism.L2.SG.InstExistsPf.instOf D) (μf : Measure I.Gf) [μf.IsHaarMeasure] [SigmaFinite μf] (f : I.GA → ℂ), Continuous f → (∀ γ ∈ I.GFset, ∀ y, f (γ * y) = f y) → I.IsCuspFn (I.muA μf) f → IsCuspFun (D.sp.Y 0) (fun y => f (((Formalism.IndefR.RationalEmbedding.iota D).trans (castGA hI.symm)) y))

theorem **isCuspFn_of_contRep** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) {F : L2X X} {ψ : X.B.I.GA → ℂ} (hψ : Formalism.L2.SG.EvalTOntoRep.IsContRep X F ψ) (hF : X.B.I.IsCuspFn (X.B.I.muA X.B.μf) (X.B.I.ext F)) : X.B.I.IsCuspFn (X.B.I.muA X.B.μf) ψ

theorem **conjMap_isCuspFun** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowMult.vecC X M c u))) ψ) → ∀ u : M.Vec, I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateIntertwinerValuesIsCuspFunStatement

theorem **_root_.GlobalTheory.T.conjugateIntertwiner_values_isCuspFun** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjR

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateIntertwinerValuesSmoothAutomorphic

theorem **conj_lie** {n : ℕ} {P H J E : Matrix (Fin n) (Fin n) ℂ} (hP : IsUnit P) (hs : Pᴴ * H * P = J) (hE : Eᴴ * J + J * E = 0) : (P * E * P⁻¹)ᴴ * H + H * (P * E * P⁻¹) = 0

theorem **exp_conj** {n : ℕ} {P : Matrix (Fin n) (Fin n) ℂ} (hP : IsUnit P) (E : Matrix (Fin n) (Fin n) ℂ) (t : ℝ) : P⁻¹ * NormedSpace.exp (t • (P * E * P⁻¹)) * P = NormedSpace.exp (t • E)

theorem **iota_data** {D : D7} (θ : InfFam D) (hθ : IsInfFam θ) (hexp : ∀ τ, ∀ Y ∈ lieAt (D.sp.Y 0) τ, ∀ t : ℝ, NormedSpace.exp (t • Y) ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ)) (I : AutInst) (h : I = Formalism.L2.SG.InstExistsPf.instOf D) : let ι

theorem **conjMap_isAutFunSm** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowMult.vecC X M c u))) ψ) → ∀ u : M.Vec,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateIntertwinerValuesSmoothAutomorphicStatement

theorem **_root_.GlobalTheory.T.conjugateIntertwiner_values_smoothAutomorphic** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheo

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateRowPairingPoincareSeriesBesselPeriods

theorem **perLaw_holds** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (Y : Formalism.L2.SG.Full.Full X), IsStdChar D ψ (-1) → InHH X u Y.hK → B.PerLaw Y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateRowPairingPoincareSeriesBesselPeriodsStatement

theorem **_root_.GlobalTheory.T.conjugateRow_pairing_poincareSeries_besselPeriods** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (Y : Formalism.L2.SG.Full.Full X), IsStdChar D ψ (-1) → InHH X u Y.hK → B.PerLaw Y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateSymmetricUnipotentPattern

theorem **mem_unipSym_iff** {K : Type} [CommRing K] {r : ℕ} (ρ κ : ℕ) (g : GL (Fin r) K) : g ∈ unipSym K r ρ κ ↔ ∀ a b : Fin r, ¬ (a.val < b.val ∧ ¬ (ρ ≤ a.val ∧ a.val < ρ + κ ∧ b.val = ρ + κ)) → (g : Matrix (Fin r) (Fin r) K) a b = (1 : Matrix (Fin r) (Fin r) K) a b

theorem **perm_val** {r ρ κ : ℕ} (hr : κ + ρ ≤ r) (i : Fin r) : (perm r ρ κ hr i).val = permFn ρ κ i.val

theorem **freePos_iff** (ρ κ i j : ℕ) : GlobalTheory.SymRes.FreePos ρ κ i j ↔ (permFn ρ κ i < permFn ρ κ j ∧ ¬ (ρ ≤ permFn ρ κ i ∧ permFn ρ κ i < ρ + κ ∧ permFn ρ κ j = ρ + κ))

theorem **reidx_apply** {K : Type} [CommRing K] {r ρ κ : ℕ} (hr : κ + ρ ≤ r) (g : GL (Fin r) K) (i j : Fin r) : ((reidx r ρ κ hr g : GL (Fin r) K) : Matrix (Fin r) (Fin r) K) (perm r ρ κ hr i) (perm r ρ κ hr j) = (g : Matrix (Fin r) (Fin r) K) i j

theorem **reidx_mem_iff** {K : Type} [CommRing K] {r ρ κ : ℕ} (hr : κ + ρ ≤ r) (g : GL (Fin r) K) : reidx r ρ κ hr g ∈ unipSym K r ρ κ ↔ GlobalTheory.SymRes.IsPat ρ κ g

theorem **conj_pat_onto** : ∀ (K : Type) [CommRing K] [Finite K] (r ρ κ : ℕ), r = κ + (2 * ρ + 1) → ∀ m : GL (Fin r) K, (∀ g : GL (Fin r) K, GlobalTheory.SymRes.IsPat ρ κ g → m * g * m⁻¹ ∈ Formalism.L2.DIV.unipSym K r ρ κ) → ∀ g : GL (Fin r) K, m * g * m⁻¹ ∈ Formalism.L2.DIV.unipSym K r ρ κ → GlobalTheory.SymRes.IsPat ρ κ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateSymmetricUnipotentPatternStatement

theorem **_root_.GlobalTheory.T.conjugate_symmetric_unipotent_pattern** : ∀ (K : Type) [CommRing K] [Finite K] (r ρ κ : ℕ), r = κ + (2 * ρ + 1) → ∀ m : GL (Fin r) K, (∀ g : GL (Fin r) K, GlobalTheory.SymRes.IsPat ρ κ g → m * g * m⁻¹ ∈ Formalism.L2.DIV.unipSym K r ρ κ) → ∀ g : GL (Fin r) K, m * g * m⁻¹ ∈ Formalism.L2.DIV.unipSym K r ρ κ → GlobalTheory.SymRes.IsPat ρ κ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateVectorCentralTwist

theorem **anchor_pairs** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → PuNZ D → ∀ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) → ∀ v : (W.Q.Rl.MY 0).Vec, ConjVec W.S.X W.Q.instEq W.Q.Rl.θ (W.Q.Rl.MY 0) v → ∀ u, (∀ φ, ∃ C, IsCol W u C φ) ∧ (∀ T ∈ ThetaLiftSpaceZero W, ∀ C, ∃ b, IsPair W u (z * (show GA D (D.sp.Y 0) from W.Q.tS)) (T (PiTensorProduct.tprod ℂ v)) C b) ∧ ∃ (φ : D.what₂.Mu u) (C : _) (b : ℂ), ∃ T ∈ ThetaLiftSpaceZero W, IsCol W u C φ ∧ IsPair W u (z * (show GA D (D.sp.Y 0) from W.Q.tS)) (T (PiTensorProduct.tprod ℂ v)) C b ∧ b ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateVectorCentralTwistStatement

theorem **_root_.GlobalTheory.T.conjugate_vector_central_twist** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → PuNZ D → ∀ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) → ∀ v : (W.Q.Rl.MY 0).Vec, ConjVec W.S.X W.Q.instEq W.Q.Rl.θ (W.Q.Rl.MY 0) v → ∀ u, (∀ φ, ∃ C, IsCol W u C φ) ∧ (∀ T ∈ ThetaLiftSpaceZero W, ∀ C, ∃ b, IsPair W u (z * (show GA D (D.sp.Y 0) from W.Q.tS)) (T (PiTensorProduct.tprod ℂ v)) C b) ∧ ∃ (φ : D.what₂.Mu u) (C : _) (b : ℂ), ∃ T ∈ ThetaLiftSpaceZero W, IsCol W u C φ ∧ IsPair W u (z * (show GA D (D.sp.Y 0) from W.Q.tS)) (T (PiTensorProduct.tprod ℂ v)) C b ∧ b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateVectorPairForm

theorem **inner_sp** {St : Formalism.L2.SG.HidaSetting.HidaSetting D} (s : Bool) {Bs : DefBase D.what₂.LL} {Df : DefSide Bs} (Dc : DefCols Df) (hK : St.LevelSubgroupCharacter) (Gd : GoodOp St) (wdD : Dc.Mem → D.what₂.Word) (pw : ∀ x, PerW St hK (wdD x)) (bR : ∀ x, Fin (Dc.a x) → (D.mem (wdD x)).A.Mram) (Sp : SPer Dc) (hSp : Sp = spOf s Dc hK Gd wdD pw bR) {E : Type} [NormedAddCommGroup E] [InnerProductSpace ℂ E] (sEE : E ≃ₗ⋆[ℂ] Sp.HH) (sEEc : E ≃ₗ⋆[ℂ] Tw s (HHX St)) (h : HEq sEE sEEc) (e : E) (m : Dc.Mem) (c : Fin (Dc.a m) → ℂ) : ⟪sEE e, Sp.Pu m (cjv s c)⟫_ℂ = cj s ⟪Tw.out (sEEc e), (pw m).op (ramCrd (bR m) c)⟫_ℂ

theorem **inner_join** {St : Formalism.L2.SG.HidaSetting.HidaSetting D} (s : Bool) {Bs : DefBase D.what₂.LL} {Z : Type} {sd : Z → DefSide Bs} (Dcf : ∀ z, DefCols (sd z)) (Spf : ∀ z, SPer (Dcf z)) (z0 z1 : Z) (hz : z0 = z1) (Dc : DefCols (sd z1)) (hdc : Dcf z1 = Dc) (hK : St.LevelSubgroupCharacter) (Gd : GoodOp St) (wdD : Dc.Mem → D.what₂.Word) (pw : ∀ x, PerW St hK (wdD x)) (bR : ∀ x, Fin (Dc.a x) → (D.mem (wdD x)).A.Mram) (hsp : HEq (Spf z1) (spOf s Dc hK Gd wdD pw bR)) {E : Type} [NormedAddCommGroup E] [InnerProductSpace ℂ E] (sEE : E ≃ₗ⋆[ℂ] (Spf z0).HH) (sEEc : E ≃ₗ⋆[ℂ] Tw s (HHX St)) (h : HEq sEE sEEc) (e : E) (m : (Dcf z0).Mem) (m' : Dc.Mem) (hm : HEq m m') (c : Fin ((Dcf z0).a m) → ℂ) (c' : Fin (Dc.a m') → ℂ) (hc : HEq c c') : ⟪sEE e, (Spf z0).Pu m (cjv s c)⟫_ℂ = cj s ⟪Tw.out (sEEc e), (pw m').op (ramCrd (bR m') c')⟫_ℂ

theorem **iota_symm_eq** {I : Formalism.MainNotions.S.AutInst} (h : I = Formalism.L2.SG.InstExistsPf.instOf D) (ι : GA D (D.sp.Y 0) ≃* I.GA) (hι : HEq ι (Formalism.IndefR.RationalEmbedding.iota D).toMulEquiv) (x : I.GA) : ι.symm x = ((Formalism.IndefR.RationalEmbedding.iota D).trans (castGA h.symm)).symm x

theorem **op_eq** {X : Formalism.L2.SG.HidaSetting.HidaSetting D} {hK : X.LevelSubgroupCharacter} {u : D.what₂.Word} (p : PerW X hK u) (y : (D.mem u).A.Mu0) (T : Cu X u hK) (hT : Formalism.L2.SG.ColDict.colDictL X p.Tr hK T = (y : (D.mem u).C.cls 0 → (D.mem u).C.Y 0 → ℂ)) : ((p.op y : HHX X) : L2X X) = PuFun X u hK p.Dt T

theorem **op_col** (W : RealPack ex) (u w : D.what₂.Word) (hwu : w = u) (hc : W.P.I.wU w = W.P.I.wU u) (x : Fin (W.P.Dc.a (W.P.I.wU u)) → ℂ) (T : Cu W.S.X u W.S.Y.hK) (hT : IsCol W u T (colC W u x)) : (((pwOf W.S W.Q.hin w).op (ramCrd (fun i => D.eMu w (D.jN w (W.P.I.eNu w (Pi.single (Fin.cast (congrArg W.P.Dc.a hc).symm i) (1 : D.what₂.LL))))) x) : HHX W.S.X) : L2X W.S.X) = PuFun W.S.X u W.S.Y.hK (W.S.Y.Dt u) T

theorem **exists_eval** {D : D7} {S : D.what₂.Tau → Formalism.BasicNotions.TH.SPar} {M : DSModel (D.sp.Y 0) S} {μ : Measure (GA D (D.sp.Y 0))} {ℱ : Set (GA D (D.sp.Y 0))} {E : Submodule ℂ (MMap M)} {St : Formalism.L2.SG.HidaSetting.HidaSetting D} {Gd : GoodOp St} {ι : GA D (D.sp.Y 0) ≃* St.B.I.GA} {t : GA D (D.sp.Y 0)} {c : ℝ} (hE : EvalTieT M μ ℱ E St Gd ι t c) (h : St.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (hι : HEq ι (Formalism.IndefR.RationalEmbedding.iota D).toMulEquiv) : ∃ v₀ : M.Vec, (∀ τ, ‖(v₀ τ : M.H τ)‖ = 1) ∧ ∀ T : E, ∃ F : L2X St, ((F : St.B.I.GA → ℂ) =ᵐ[muX St] evS St h t (PiTensorProduct.lift (T : MMap M) (PiTensorProduct.tprod ℂ v₀))) ∧ star F ∈ HHX St ∧ ((sEEofTT M μ ℱ E St Gd ι t c T : HHX St) : L2X St) = star F

theorem **inner_smul_star** {D : D7} {St : Formalism.L2.SG.HidaSetting.HidaSetting D} (h : St.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (t : GA D (D.sp.Y 0)) (f g : GA D (D.sp.Y 0) → ℂ) (l a : ℂ) (hfg : f = l • g) (F φ P : L2X St) (hF : (F : St.B.I.GA → ℂ) =ᵐ[muX St] evS St h t f) (hφ : (φ : St.B.I.GA → ℂ) =ᵐ[muX St] evS St h t g) (y z : HHX St) (hy : (y : L2X St) = star F) (hz : (z : L2X St) = P) : ⟪a • y, z⟫_ℂ = starRingEnd ℂ a * l * ⟪star φ, P⟫_ℂ

theorem **pairForm** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → ∀ v : (W.Q.Rl.MY 0).Vec, ConjVec W.S.X W.Q.instEq W.Q.Rl.θ (W.Q.Rl.MY 0) v → PairForm W v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConjugateVectorPairFormStatement

theorem **_root_.GlobalTheory.T.conjugate_vector_pair_form** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl :

Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → ∀ v : (W.Q.Rl.MY 0).Vec, ConjVec W.S.X W.Q.instEq W.Q.Rl.θ (W.Q.Rl.MY 0) v → PairForm W v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConstantsExistForBundle

theorem **ties_at** : GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => Formalism.BasicNotions.Ties D)

theorem **scale_at** (H : Formalism.MainNotions.Inhab.FC → ℝ) : GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun κ _ O _ _ => H κ.C ≤ O.L.sc.L ∧ O.L.CGam = κ.C.CGam ∧ 1 ≤ κ.C.CGam)

theorem **sizes** (F G H : ℝ → ℝ → ℝ) : GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D O _ _ => ∃ C' Cl Cg CT : ℝ, F Cg CT ≤ C' ∧ G Cg CT ≤ Cl ∧ H Cg CT ≤ O.L.sc.L ∧ GlobalTheory.Core.Rk.HD D O.L.sc.L C' Cl Cg (1 / 4) 3 (1 / 2) CT ∧ GlobalTheory.Core.Rk.IsScaleWithBoundedGamma D O.L.sc.L C' ∧ (∀ u, GlobalTheory.Core.Rk.RankMu D u) ∧ GlobalTheory.Core.Rk.BigGap D ∧ 1 ≤ D.R.rS)

theorem **member_sizes** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL) (F G H : ℝ → ℝ → ℝ), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D O _ _ => ∃ C' Cl Cg CT : ℝ, F Cg CT ≤ C' ∧ G Cg CT ≤ Cl ∧ H Cg CT ≤ O.L.sc.L ∧ GlobalTheory.Core.Rk.HD D O.L.sc.L C' Cl Cg (1 / 4) 3 (1 / 2) CT ∧ GlobalTheory.Core.Rk.IsScaleWithBoundedGamma D O.L.sc.L C' ∧ (∀ u, GlobalTheory.Core.Rk.RankMu D u) ∧ GlobalTheory.Core.Rk.BigGap D ∧ 1 ≤ D.R.rS)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConstantsExistForBundleStatement

theorem **_root_.GlobalTheory.T.constants_exist_for_bundle** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL) (F G H : ℝ → ℝ → ℝ), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D O _ _ => ∃ C' Cl Cg CT : ℝ, F Cg CT ≤ C' ∧ G Cg CT ≤ Cl ∧ H Cg CT ≤ O.L.sc.L ∧ GlobalTheory.Core.Rk.HD D O.L.sc.L C' Cl Cg (1 / 4) 3 (1 / 2) CT ∧ GlobalTheory.Core.Rk.IsScaleWithBoundedGamma D O.L.sc.L C' ∧ (∀ u, GlobalTheory.Core.Rk.RankMu D u) ∧ GlobalTheory.Core.Rk.BigGap D ∧ 1 ≤ D.R.rS)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContAutRapidDecBoundedIntegrable

theorem **contAut_rapidDec_bounded_integrable** : ∀ (D : D7) (V : HSp D), CoCpt V → ∀ φ : GA D V → ℂ, IsContAut V φ → RapidDec V φ ∧ (∃ C : ℝ, ∀ g, ‖φ g‖ ≤ C) ∧ ∀ v : Vol V, IntegrableOn φ v.𝓕 v.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffCompactAverageOfClosedPartials

theorem **integrable_cont** (μ : Measure K) [IsFiniteMeasure μ] {f : K → ℂ} (hf : Continuous f) : Integrable f μ

theorem **clm_expand** (L : (Fin n → ℝ) →L[ℝ] ℂ) : L = ∑ i, L (Pi.single i 1) • coordC n i

theorem **hasFDerivAt_avg** (μ : Measure K) [IsFiniteMeasure μ] {F : K → (Fin n → ℝ) → ℂ} {G : Fin n → K → (Fin n → ℝ) → ℂ} (hF : Continuous fun p : K × (Fin n → ℝ) => F p.1 p.2) (hG : ∀ i, Continuous fun p : K × (Fin n → ℝ) => G i p.1 p.2) (hd : ∀ k, Differentiable ℝ (F k)) (he : ∀ i k t, fderiv ℝ (F k) t (Pi.single i 1) = G i k t) (t₀ : Fin n → ℝ) : HasFDerivAt (fun t => ∫ k, F k t ∂μ) (∑ i, (∫ k, G i k t₀ ∂μ) • coordC n i) t₀

theorem **GlobalTheory.AvgSmooth.main** : ∀ (K : Type) [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] (μ : Measure K) [IsFiniteMeasure μ] (n : ℕ) (φ : Fin n → (Fin n → ℝ) → ℂ), (∀ i, ContDiff ℝ ∞ (φ i)) → ∀ 𝒢 : Set (K → (Fin n → ℝ) → ℂ), (∀ F ∈ 𝒢, Continuous fun p : K × (Fin n → ℝ) => F p.1 p.2) → (∀ F ∈ 𝒢, ∀ k, Differentiable ℝ (F k)) → (∀ F ∈ 𝒢, ∀ i, ∃ F' ∈ 𝒢, ∀ k t, fderiv ℝ (F k) t (Pi.single i 1) = F' k t + φ i t * F k t) → ∀ F ∈ 𝒢, ContDiff ℝ ∞ fun t => ∫ k, F k t ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffCompactAverageOfClosedPartialsStatement

theorem **_root_.GlobalTheory.T.contDiff_compactAverage_of_closed_partials** : ∀ (K : Type) [TopologicalSpace K] [CompactSpace K] [MeasurableSpace K] [BorelSpace K] (μ : Measure K) [IsFiniteMeasure μ] (n : ℕ) (φ : Fin n → (Fin n → ℝ) → ℂ), (∀ i, ContDiff ℝ ∞ (φ i)) → ∀ 𝒢 : Set (K → (Fin n → ℝ) → ℂ), (∀ F ∈ 𝒢, Continuous fun p : K × (Fin n → ℝ) => F p.1 p.2) → (∀ F ∈ 𝒢, ∀ k, Differentiable ℝ (F k)) → (∀ F ∈ 𝒢, ∀ i, ∃ F' ∈ 𝒢, ∀ k t, fderiv ℝ (F k) t (Pi.single i 1) = F' k t + φ i t * F k t) → ∀ F ∈ 𝒢, ContDiff ℝ ∞ fun t => ∫ k, F k t ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffHeisenbergSuperpositionThetaFunctionReal

theorem **clm_expand** (φ : RW T S →L[ℝ] ℂ) : φ = ∑ i, φ (bas S T i) • coordC S T i

theorem **norm_heisO_le** (hψ : IsStdChar D ψ (-1)) (x : WO S T) {G : WO S T → ℂ} {C : ℝ} (hC : ∀ y, ‖G y‖ ≤ C) (w : WO S T) : ‖heisO ψ S T x G w‖ ≤ C

theorem **theta_dl** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {F : WO S T → ℂ} (hF : IsThetaFunO ψ S T F) (u : RW T S) : IsThetaFunO ψ S T (dl ψ σ S T u F)

theorem **hasFDerivAt_pt** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {F : WO S T → ℂ} (hF : IsThetaFunO ψ S T F) (w : WO S T) (c : ℂ) (z : RW T S) : HasFDerivAt (fun z : RW T S => c * heisO ψ S T (iotaW σ T S z) F w) (∑ i, (c * sco ψ σ S T F (bas S T i) z w) • coordC S T i) z

theorem **aesm_heisO** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {G : WO S T → ℂ} (hG : IsThetaFunO ψ S T G) {Ω : Type} [MeasurableSpace Ω] (μ : Measure Ω) {v : Ω → WO S T} (hv : Measurable v) (z : RW T S) : AEStronglyMeasurable (fun x => heisO ψ S T (iotaW σ T S z) G (v x)) μ

theorem **integrable_heisO** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {G : WO S T → ℂ} (hG : IsThetaFunO ψ S T G) {Ω : Type} [MeasurableSpace Ω] (μ : Measure Ω) {v : Ω → WO S T} (hv : Measurable v) {a : Ω → ℂ} (ha : Integrable a μ) (z : RW T S) : Integrable (fun x => a x * heisO ψ S T (iotaW σ T S z) G (v x)) μ

theorem **aesm_sco** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {F : WO S T → ℂ} (hF : IsThetaFunO ψ S T F) {Ω : Type} [MeasurableSpace Ω] (μ : Measure Ω) {v : Ω → WO S T} (hv : Measurable v) (u z : RW T S) : AEStronglyMeasurable (fun x => sco ψ σ S T F u z (v x)) μ

theorem **integral_sco** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {F : WO S T → ℂ} (hF : IsThetaFunO ψ S T F) {Ω : Type} [MeasurableSpace Ω] (μ : Measure Ω) {v : Ω → WO S T} (hv : Measurable v) {a : Ω → ℂ} (ha : Integrable a μ) (u z : RW T S) : Integrable (fun x => a x * sco ψ σ S T F u z (v x)) μ ∧ ∫ x, a x * sco ψ σ S T F u z (v x) ∂μ = (Real.pi * Complex.I) * ((omO S T z u : ℝ) : ℂ) * (∫ x, a x * heisO ψ S T (iotaW σ T S z) F (v x) ∂μ) + ∫ x, a x * heisO ψ S T (iotaW σ T S z) (dl ψ σ S T u F) (v x) ∂μ

theorem **hasFDerivAt_superpos** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {F : WO S T → ℂ} (hF : IsThetaFunO ψ S T F) {Ω : Type} [MeasurableSpace Ω] (μ : Measure Ω) {v : Ω → WO S T} (hv : Measurable v) {a : Ω → ℂ} (ha : Integrable a μ) (z₀ : RW T S) : HasFDerivAt (fun z : RW T S => ∫ x, a x * heisO ψ S T (iotaW σ T S z) F (v x) ∂μ) (∑ i, ((Real.pi * Complex.I) * ((omO S T z₀ (bas S T i) : ℝ) : ℂ) * (∫ x, a x * heisO ψ S T (iotaW σ T S z₀) F (v x) ∂μ) + ∫ x, a x * heisO ψ S T (iotaW σ T S z₀) (dl ψ σ S T (bas S T i) F) (v x) ∂μ) • coordC S T i) z₀

theorem **contDiff_superpos_nat** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {Ω : Type} [MeasurableSpace Ω] (μ : Measure Ω) {v : Ω → WO S T} (hv : Measurable v) {a : Ω → ℂ} (ha : Integrable a μ) (n : ℕ) : ∀ F : WO S T → ℂ, IsThetaFunO ψ S T F → ContDiff ℝ n (fun z : RW T S => ∫ x, a x * heisO ψ S T (iotaW σ T S z) F (v x) ∂μ)

theorem **GlobalTheory.SuperposSmooth.main** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), GlobalTheory.FockCompare.Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ (Ω : Type) [MeasurableSpace Ω] (μ : Measure Ω) (v : Ω → WO S T) (a : Ω → ℂ), Measurable v → Integrable a μ → ContDiff ℝ ∞ (fun z : GlobalTheory.FockCompare.RW T S => ∫ x, a x * heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F (v x) ∂μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffHeisenbergSuperpositionThetaFunctionRealStatement

theorem **_root_.GlobalTheory.T.contDiff_heisenbergSuperposition_thetaFunction_real** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), GlobalTheory.FockCompare.Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ (Ω : Type) [MeasurableSpace Ω] (μ : Measure Ω) (v : Ω → WO S T) (a : Ω → ℂ), Measurable v → Integrable a μ → ContDiff ℝ ∞ (fun z : GlobalTheory.FockCompare.RW T S => ∫ x, a x * heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F (v x) ∂μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffHeisenbergTranslateThetaFunctionRealCoor

theorem **hasDerivAt_expLine** {H : ℝ → ℂ} {h' : ℂ} (hH : HasDerivAt H h' 0) (a k : ℝ) : HasDerivAt (fun s : ℝ => Complex.exp (-(Real.pi * Complex.I) * ((a + s * k : ℝ) : ℂ)) * H s) (Complex.exp (-(Real.pi * Complex.I) * (a : ℂ)) * (-(Real.pi * Complex.I) * (k : ℂ) * H 0 + h')) 0

theorem **contDiff_heisO_iota_fderiv** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), GlobalTheory.FockCompare.Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ v : WO S T, ContDiff ℝ ∞ (fun z : GlobalTheory.FockCompare.RW T S => heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F v) ∧ ∀ z u : GlobalTheory.FockCompare.RW T S, fderiv ℝ (fun z : GlobalTheory.FockCompare.RW T S => heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F v) z u = (Real.pi * Complex.I) * ((GlobalTheory.OrientedHeis.omO S T z u : ℝ) : ℂ) * heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F v + heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) (fun y => deriv (fun s : ℝ => heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S (s • u)) F y) 0) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffHeisenbergTranslateThetaFunctionRealCoorStatement

theorem **_root_.GlobalTheory.T.contDiff_heisenbergTranslate_thetaFunction_realCoords** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), GlobalTheory.FockCompare.Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ v : WO S T, ContDiff ℝ ∞ (fun z : GlobalTheory.FockCompare.RW T S => heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F v) ∧ ∀ z u : GlobalTheory.FockCompare.RW T S, fderiv ℝ (fun z : GlobalTheory.FockCompare.RW T S => heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F v) z u = (Real.pi * Complex.I) * ((GlobalTheory.OrientedHeis.omO S T z u : ℝ) : ℂ) * heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) F v + heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S z) (fun y => deriv (fun s : ℝ => heisO ψ S T (GlobalTheory.FockCompare.iotaW σ T S (s • u)) F y) 0) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffOfHeisenbergDerivativeClosedClass

theorem **differentiableAt_phase** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {M : ℝ →+ WO S T} (hM : Continuous M) (hMf : ∀ s, (M s).map (finPart D) = 0) (w : WO S T) : DifferentiableAt ℝ (fun s : ℝ => ψ (symplOH S T w (M s))) 0

theorem **hasDerivAt_line** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {M : ℝ →+ WO S T} (hM : Continuous M) (hMf : ∀ s, (M s).map (finPart D) = 0) {f : WO S T → ℂ} {w : WO S T} {c : ℂ} (hd : HasDerivAt (fun s : ℝ => heisO ψ S T (M s) f w) c 0) : HasDerivAt (fun s : ℝ => f (w + M s)) (c + -deriv (fun s : ℝ => ψ (symplOH S T w (M s))) 0 * f w) 0

theorem **contDiff_of_heis_closed** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (V : Set (WO S T → ℂ)), (∀ f ∈ V, Continuous f) → (∀ f ∈ V, ∀ M : ℝ →+ WO S T, Continuous M → (∀ t, (M t).map (finPart D) = 0) → ∃ f' ∈ V, ∀ w, HasDerivAt (fun t : ℝ => heisO ψ S T (M t) f w) (f' w) 0) → ∀ f ∈ V, ∀ (k : ℕ) (L : (Fin k → ℝ) →+ WO S T), Continuous L → (∀ t, (L t).map (finPart D) = 0) → ∀ w, ContDiff ℝ ∞ fun t => f (w + L t)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContDiffOfHeisenbergDerivativeClosedClassStatement

theorem **_root_.GlobalTheory.T.contDiff_of_heisenbergDerivativeClosedClass** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (V : Set (WO S T → ℂ)), (∀ f ∈ V, Continuous f) → (∀ f ∈ V, ∀ M : ℝ →+ WO S T, Continuous M → (∀ t, (M t).map (finPart D) = 0) → ∃ f' ∈ V, ∀ w, HasDerivAt (fun t : ℝ => heisO ψ S T (M t) f w) (f' w) 0) → ∀ f ∈ V, ∀ (k : ℕ) (L : (Fin k → ℝ) →+ WO S T), Continuous L → (∀ t, (L t).map (finPart D) = 0) → ∀ w, ContDiff ℝ ∞ fun t => f (w + L t)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuesWithModerateGrowthShiftTranslatePow

theorem **holValueAt_of_continuousAt** {h : ℂ → ℂ} (a : GlobalTheory.Cont.Continues h) {t : ℂ} (b : ContinuousAt (GlobalTheory.Cont.cont h) t) : GlobalTheory.Cont.HolValueAt h t (GlobalTheory.Cont.cont h t)

theorem **differentiable_npow_sub** {c : ℝ≥0} (hc : 0 < c) (s₁ : ℂ) : Differentiable ℂ fun s => npow c (s₁ - s)

theorem **contMod_shift** (D : D7) (T : HSp D) (e e' : ℂ → GA D T → ℂ) (x : GA D T) (c : ℝ≥0) (s₁ s₂ : ℂ) (hc : 0 < c) (he : ∀ s g, e' s g = npow c (s₁ - s) * e (s + s₂) (g * x)) (hm : ContMod T e) : ContMod T e'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuesWithModerateGrowthShiftTranslatePowStatement

theorem **_root_.GlobalTheory.T.continuesWithModerateGrowth_shift_translate_pow** : ∀ (D : D7) (T : HSp D) (e e' : ℂ → GA D T → ℂ) (x : GA D T) (c : ℝ≥0) (s₁ s₂ : ℂ), 0 < c → (∀ s g, e' s g = npow c (s₁ - s) * e (s + s₂) (g * x)) → ContMod T e → ContMod T e'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousChartLevelCharactersAdelicTuple

theorem **chi_continuous** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) β, Continuous fun κ => ((A.chi i β κ : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousChartLevelCharactersAdelicTupleStatement

theorem **_root_.GlobalTheory.T.continuous_chartLevelCharacters_adelic_tuple** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i ≤ B.R.m) β, Continuous fun κ => ((A.chi i β κ : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousComplEquivSymm

theorem **sigR_compFin'** (σ : E →ₐ[F] E) (v : HeightOneSpectrum (𝒪 F)) (z : AF F ⊗[F] E) : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ (compFin F E v z) = compFin F E v (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ z)

theorem **continuous_complEquiv_symm** {σ : E →ₐ[F] E} {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (hE : IsCMExt F E σ) (h : w.under (𝒪 F) = v) {τ : E →ₐ[F] Fv F v} (hτ : splitPrime τ = w) : Continuous (complEquiv hE h hτ).symm

theorem **continuous_compW** {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (h : w.under (𝒪 F) = v) : Continuous fun a : AF F ⊗[F] E => GlobalTheory.LFun.toCompl h (compFin F E v a)

theorem **exists_iota_gen** {σ : E →ₐ[F] E} {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (hE : IsCMExt F E σ) (h : w.under (𝒪 F) = v) {τ : E →ₐ[F] Fv F v} (hτ : splitPrime τ = w) : ∃ ι : w.adicCompletion E →+ AF F ⊗[F] E, Continuous ι ∧ (∀ x, GlobalTheory.LFun.toCompl h (compFin F E v (ι x)) = x) ∧ (∀ x, GlobalTheory.LFun.toCompl h (compFin F E v (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ (ι x))) = 0) ∧ (∀ x, algebraMap (AF F) (AF F ⊗[F] E) (pairA F 0 (singleFin v 1)) * ι x = ι x) ∧ (∀ a x, a * ι x = ι (GlobalTheory.LFun.toCompl h (compFin F E v a) * x)) ∧ (∀ a, algebraMap (AF F) (AF F ⊗[F] E) (pairA F 0 (singleFin v 1)) * a = ι (GlobalTheory.LFun.toCompl h (compFin F E v a)) + Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ (ι (GlobalTheory.LFun.toCompl h (compFin F E v (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR

theorem **GlobalTheory.Core.Rk.SplitIota.exists_iota** : ∀ (D : D7) (q : FinPl D) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = q), Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousFinGL

theorem **continuous_comm** : letI

theorem **finGL_val** (V : HSp D) (h : GA D V) : (finGL V h).val = finM V rfl h

theorem **continuous_finGL** (V : HSp D) : letI

theorem **law_left** (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) V.gram T f bad) (cl : Type) (yc : cl → UPts (sigD D) V.gram (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (hb : Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC V.gram (embT D))) : ∀ γ ∈ GQ V, ∀ g, rdS θ V rfl S yc (fun c => Ψ (yc c)) y (γ * g) = rdS θ V rfl S yc (fun c => Ψ (yc c)) y g

theorem **law_level** (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) V.gram T f bad) (cl : Type) (yc : cl → UPts (sigD D) V.gram (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (hb : Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC V.gram (embT D))) : ∀ k ∈ GfC V, ∀ κ : S.U, κ.1.1 = finM V rfl k → ∀ g, rdS θ V rfl S yc (fun c => Ψ (yc c)) y (g * k) = (S.chi κ : ℂ) * rdS θ V rfl S yc (fun c => Ψ (yc c)) y g

theorem **law_real** (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) V.gram T f bad) (cl : Type) (yc : cl → UPts (sigD D) V.gram (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (hb : Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (y : Frames (gramC V.gram (embT D))) : ∀ k ∈ GinfC V, ∀ y', IsTr (infM θ V rfl k) y y' → ∀ g, rdS θ V rfl S yc (fun c => Ψ (yc c)) y (g * k) = rdS θ V rfl S yc (fun c => Ψ (yc c)) y' g

theorem **continuous_infM** (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) (τ : D.what₂.Tau) : Continuous fun g : GA D V => infM θ V rfl g τ

theorem **law_cont** (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) V.gram T f bad) (cl : Type) (yc : cl → UPts (sigD D) V.gram (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (hb : Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S))) (Ψ) (hΨ : Ψ ∈ AdFun S (embT D)) (lam) (hval : ∀ x, Ψ x ∈ ValC (gramC V.gram (embT D)) lam) (y : Frames (gramC V.gram (embT D))) (hW : ∃ W : Subgroup (GA D V), IsOpen (W : Set (GA D V)) ∧ ∀ k ∈ W, k ∈ GfC V → ∀ κ : S.U, κ.1.1 = finM V rfl k → S.chi κ = 1) : Continuous (rdS θ V rfl S yc (fun c => Ψ (yc c)) y)

theorem **rdS_laws** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (hn : V.rank = n) (J : Matrix (Fin n) (Fin n) D.host.E), J = V.gram.reindex (finCongr hn) (finCongr hn) → ∀ (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) J T f bad) (cl : Type) (yc : cl → UPts (sigD D) J (FiniteAdeleRing (𝒪 D.host.F) D.host.F)), Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S)) → ∀ lam, ∀ φ ∈ FiC S (embT D) lam yc, ∀ y : Frames (gramC J (embT D)), (∀ γ ∈ GQ V, ∀ g, rdS θ V hn S yc φ y (γ * g) = rdS θ V hn S yc φ y g) ∧ (∀ k ∈ GfC V, ∀ κ : S.U, κ.1.1 = finM V hn k → ∀ g, rdS θ V hn S yc φ y (g * k) = (S.chi κ : ℂ) * rdS θ V hn S yc φ y g) ∧ (∀ k ∈ GinfC V, ∀ y', IsTr (infM θ V hn k) y y' → ∀ g, rdS θ V hn S yc φ y (g * k) = rdS θ V hn S yc φ y' g) ∧ ((∃ W : Subgroup (GA D V), IsOpen (W : Set (GA D V)) ∧ ∀ k ∈ W, k ∈ GfC V → ∀ κ : S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousFlatMapWeilRepresentationJoint

theorem **flatAt_om_continuous** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (χV χW : (AE D)ˣ →* Circle) (om : Om V W), IsWeilDatumO ψ V W χV χW om → ∀ (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∀ Φ, IsThetaFunO ψ V W Φ → Continuous fun q : (GA D V × GA D W) × Mx D V₀.rank σ × WO V₀ W₀ => flatAt ψ dV dW vS.μ vS.ℱ vT.μ vT.ℱ vU.ν (rat x) q.2.1 (om q.1 Φ) q.2.2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousFlatMapWeilRepresentationJointStatement

theorem **_root_.GlobalTheory.T.continuous_flatMap_weilRepresentation_joint** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (χV χW : (AE D)ˣ →* Circle) (om : Om V W), IsWeilDatumO ψ V W χV χW om → ∀ (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∀ Φ, IsThetaFunO ψ V W Φ → Continuous fun q : (GA D V × GA D W) × Mx D V₀.rank σ × WO V₀ W₀ => flatAt ψ dV dW vS.μ vS.ℱ vT.μ vT.ℱ vU.ν (rat x) q.2.1 (om q.1 Φ) q.2.2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIdempotentCutIsClosedScaledKernel

theorem **continuous_mt_cutTop** (V : HSp D) : Continuous (fun g : GA D V => Formalism.L2.Theta.mt V g)

theorem **continuous_mt_cutG** (V : HSp D) (ε : AF D.host.F) (hε : ε * ε = ε) : Continuous (fun g : GA D V => Formalism.L2.Theta.mt V (cutG V ε hε g))

theorem **continuous_cutG** (V : HSp D) (ε : AF D.host.F) (hε : ε * ε = ε) : Continuous (cutG V ε hε)

theorem **isClosed_smul_sub_one_eq_zero** (V : HSp D) (c : AF D.host.F) : IsClosed {g : GA D V | ∀ i j, algebraMap (AF D.host.F) (AE D) c * (Formalism.L2.Theta.mt V g - 1) i j = 0}

theorem **continuous_cutG_and_isClosed** : ∀ (D : D7) (V : HSp D), (∀ (ε : AF D.host.F) (hε : ε * ε = ε), Continuous (GlobalTheory.FinPl.SetCut.cutG V ε hε)) ∧ ∀ c : AF D.host.F, IsClosed {g : GA D V | ∀ i j, algebraMap (AF D.host.F) (AE D) c * (Formalism.L2.Theta.mt V g - 1) i j = 0}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIdempotentCutIsClosedScaledKernelStatement

theorem **_root_.GlobalTheory.T.continuous_idempotentCut_and_isClosed_scaledKernel** : ∀ (D : D7) (V : HSp D), (∀ (ε : AF D.host.F) (hε : ε * ε = ε), Continuous (GlobalTheory.FinPl.SetCut.cutG V ε hε)) ∧ ∀ c : AF D.host.F, IsClosed {g : GA D V | ∀ i j, algebraMap (AF D.host.F) (AE D) c * (Formalism.L2.Theta.mt V g - 1) i j = 0}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIntegralCompactAutomorphicQuotientPara

theorem **eventually_close** {G P : Type*} [TopologicalSpace G] [TopologicalSpace P] (K : G → P → ℂ) (hK : Continuous (Function.uncurry K)) {C : Set G} (hC : IsCompact C) (p₀ : P) {ε : ℝ} (hε : 0 < ε) : ∀ᶠ p in nhds p₀, ∀ g ∈ C, ‖K g p - K g p₀‖ < ε

theorem **continuous_setIntegral_of_unif** {G P : Type*} [MeasurableSpace G] [TopologicalSpace P] (μ : Measure G) (𝓕 : Set G) (hfin : μ 𝓕 < ⊤) (K : G → P → ℂ) (hint : ∀ p, IntegrableOn (fun g => K g p) 𝓕 μ) (hcl : ∀ p₀, ∀ ε : ℝ, 0 < ε → ∀ᶠ p in nhds p₀, ∀ g, ‖K g p - K g p₀‖ < ε) : Continuous fun p => ∫ g in 𝓕, K g p ∂μ

theorem **continuous_integral_param** : ∀ (D : D7) (V : HSp D), CoCpt V → ∀ (v : Vol V) (P : Type) [TopologicalSpace P] (K : GA D V → P → ℂ), Continuous (Function.uncurry K) → (∀ γ ∈ GQ V, ∀ g p, K (γ * g) p = K g p) → (∀ p, IntegrableOn (fun g => K g p) v.𝓕 v.μ) ∧ (Continuous fun p => ∫ g in v.𝓕, K g p ∂v.μ) ∧ ∀ p₀, ∃ C : ℝ, ∀ᶠ p in nhds p₀, ∀ g, ‖K g p‖ ≤ C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIntegralCompactAutomorphicQuotientParaStatement

theorem **_root_.GlobalTheory.T.continuous_integral_compactAutomorphicQuotient_param** : ∀ (D : D7) (V : HSp D), CoCpt V → ∀ (v : Vol V) (P : Type) [TopologicalSpace P] (K : GA D V → P → ℂ), Continuous (Function.uncurry K) → (∀ γ ∈ GQ V, ∀ g p, K (γ * g) p = K g p) → (∀ p, IntegrableOn (fun g => K g p) v.𝓕 v.μ) ∧ (Continuous fun p => ∫ g in v.𝓕, K g p ∂v.μ) ∧ ∀ p₀, ∃ C : ℝ, ∀ᶠ p in nhds p₀, ∀ g, ‖K g p‖ ≤ C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIntegrand

theorem **Tame.comp** {F : B → Z → ℂ} (h : Tame F) {u : Z' → Z} (hu : Continuous u) : Tame fun b z => F b (u z)

theorem **Tame.mul** {F G : B → Z → ℂ} (h : Tame F) (h' : Tame G) : Tame fun b z => F b z * G b z

theorem **tame_of_bounded** {F : B → Z → ℂ} (hc : Continuous fun p : B × Z => F p.1 p.2) {C : ℝ} (hb : ∀ b z, ‖F b z‖ ≤ C) : Tame F

theorem **Tame.integral** [FirstCountableTopology B] [FirstCountableTopology Z] [MeasurableSpace Y] [OpensMeasurableSpace Y] (μ : Measure Y) (s : Set Y) (hs : μ s < ⊤) {F : B → Z → Y → ℂ} (h : Tame fun b (q : Z × Y) => F b q.1 q.2) : Tame fun b z => ∫ y in s, F b z y ∂μ

theorem **Tame.continuous_integral** [FirstCountableTopology B] [MeasurableSpace Y] [OpensMeasurableSpace Y] (μ : Measure Y) (s : Set Y) (hs : μ s < ⊤) {F : B → Y → ℂ} (h : Tame F) : Continuous fun b => ∫ y in s, F b y ∂μ

theorem **vol_lt_top** {V : HSp D} (hV : CoCpt V) (v : Vol V) : v.μ v.𝓕 < ⊤

theorem **bounded_of_contAut** {V : HSp D} (hV : CoCpt V) {φ : GA D V → ℂ} (hφ : IsContAut V φ) : ∃ C : ℝ, ∀ g, ‖φ g‖ ≤ C

theorem **heisO_apply_zero** (S T : HSp D) (w : WO S T) (f : WO S T → ℂ) : heisO ψ S T w f 0 = f w

theorem **om_rat** {S T : HSp D} {χ χ' : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χ χ' om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {γ : GA D S} (hγ : γ ∈ GQ S) {γ' : GA D T} (hγ' : γ' ∈ GQ T) (w : WO S T) : om (γ, γ') f w = f (actO S T γ⁻¹ γ'⁻¹ w)

theorem **tame_om** {S T A T₀ : HSp D} {χ χ' : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χ χ' om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hA : CoCpt A) (hT : CoCpt T₀) (a : GA D A →* GA D S) (b : GA D T₀ →* GA D T) (ha : Continuous a) (hb : Continuous b) (ha' : ∀ γ ∈ GQ A, a γ ∈ GQ S) (hb' : ∀ γ ∈ GQ T₀, b γ ∈ GQ T) : Tame fun (h : GA D T) (q : (GA D A × GA D T₀) × WO S T) => om (a q.1.1, b q.1.2 * h) f q.2

theorem **tame_mixS** (hψ : IsStdChar D ψ (-1)) {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (x : Mx D T.rank ρ) {F : B → Z → WO S T → ℂ} (h : Tame fun b (q : Z × WO S T) => F b q.1 q.2) : Tame fun b (q : Z × WO S₀ T) => d.mixS T ψ v.μ v.𝓕 (F b q.1) x q.2

theorem **tame_mixT** (hψ : IsStdChar D ψ (-1)) {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (v : VolM D ρ S.rank) (y : Mx D S.rank ρ) {F : B → Z → WO S T → ℂ} (h : Tame fun b (q : Z × WO S T) => F b q.1 q.2) : Tame fun b (q : Z × WO S T₀) => d.mixT S ψ v.μ v.𝓕 (F b q.1) y q.2

theorem **isTheta_evalBF** (hv : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D) {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) : IsThetaFunO ψ B.V₀ J.W₀ (evalBF s Ψ)

theorem **om_evalBF** (hd : Descent D) (hm : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedModelFormulas D) (hv : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D) {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (h₀ : GA D B.A) (h' : GA D J.W₀) : s.om₀ (B.o.emb (h₀, 1), h') (evalBF s Ψ) = evalBF s (s.om (embA B.d B.o h₀, J.d.emb0 h') Ψ)

theorem **fjKer_evalBF** (hd : Descent D) (hm : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedModelFormulas D) (hv : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D) {Φ : WO B.V J.W → ℂ} (hΦ : IsThetaFunO ψ B.V J.W Φ) (h : GA D J.W) (h₀ : GA D B.A) (n : GA D J.W) (h' : GA D J.W₀) : fjKer ψ J.d B.o B.ζ s.om₀ (evalBF s (s.om (1, h) Φ)) h₀ n h' = ψ (cenT J.d B.V₀ (yZero B.V₀ ρ (vecB B.o B.ζ)) n) * evalBF s (s.om (embA B.d B.o h₀, J.d.emb0 h' * h) Φ) (actO B.V₀ J.W₀ (B.o.emb (h₀, 1)) 1 (ptT J.d B.V₀ (yZero B.V₀ ρ (vecB B.o B.ζ)) n))

theorem **firstCountable_GA** (V : HSp D) : FirstCountableTopology (GA D V)

theorem **continuous_embL** (B : BesLev D ρ) : Continuous fun h₀ : GA D B.A => B.o.emb (h₀, 1)

theorem **tame_evalBF** {Φ : WO B.V J.W → ℂ} (hΦ : IsThetaFunO ψ B.V J.W Φ) : Tame fun (h : GA D J.W) (q : (GA D B.A × GA D J.W₀) × WO B.V₀ J.W₀) => evalBF s (s.om (embA B.d B.o q.1.1, J.d.emb0 q.1.2 * h) Φ) q.2

theorem **continuous_mt_inv** (V : HSp D) : Continuous fun g : GA D V => mt V g⁻¹

theorem **continuous_ptT** {T T₀ : HSp D} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) : Continuous (ptT d S y)

theorem **continuous_cenT** {T T₀ : HSp D} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) : Continuous (cenT d S y)

theorem **continuous_psiFlag** (hψ : Continuous ψ) {T T₀ : HSp D} (d : HypDec T T₀ ρ) : Continuous (psiFlag ψ d)

theorem **tame_fjKer** (hd : Descent D) (hm : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedModelFormulas D) (hv : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D) {Φ : WO B.V J.W → ℂ} (hΦ : IsThetaFunO ψ B.V J.W Φ) : Tame fun (h : GA D J.W) (q : GA D B.A × GA D J.W × GA D J.W₀) => fjKer ψ J.d B.o B.ζ s.om₀ (evalBF s (s.om (1, h) Φ)) q.1 q.2.1 q.2.2

theorem **tame_inner** (hd : Descent D) (hm : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedModelFormulas D) (hv : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D) {Φ : WO B.V J.W → ℂ} (hΦ : IsThetaFunO ψ B.V J.W Φ) {φ : GA D B.A → ℂ} (hφ : IsContAut B.A φ) : Tame fun (h : GA D J.W) (q : GA D J.W × GA D J.W₀) => ∫ h₀ in B.vA.𝓕, φ h₀ * fjKer ψ J.d B.o B.ζ s.om₀ (evalBF s (s.om (1, h) Φ)) h₀ q.1 q.2 ∂B.vA.μ

theorem **tame_fjAllFun** (hd : Descent D) (hm : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedModelFormulas D) (hv : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D) {Φ : WO B.V J.W → ℂ} (hΦ : IsThetaFunO ψ B.V J.W Φ) {φ : GA D B.A → ℂ} (hφ : IsContAut B.A φ) {G : GA D J.W → ℂ} (hG : RapidDec J.W G) : Tame fun (h : GA D J.W) (q : GA D J.W₀ × flagN J.d) => fjAllFun ψ J.d B.o B.ζ s.om₀ B.vA.μ B.vA.𝓕 φ (rTrans h G) (evalBF s (s.om (1, h) Φ)) q.1 q.2

theorem **continuous_integrand** : ∀ D : D7, Descent D → GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedModelFormulas D → GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.MixedValues D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → Continuous fun h => J.perAll ψ B s.om₀ φ (rTrans h G) (evalBF s (s.om (1, h) Φ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIteratedWeilActionBiTheta

theorem **four_eq** (C : WO A S → WO S A' → ℂ) (φ₁ : WO A A' → ℂ) (φ₂ : WO S S → ℂ) (x : WO X Y) : four d e C φ₁ φ₂ x = C (e.resT A (d.resS Y x).2).1 (e.resT S (d.resS Y x).1).2 * (φ₁ (e.resT A (d.resS Y x).2).2 * φ₂ (e.resT S (d.resS Y x).1).1)

theorem **four_symm** (C : WO A S → WO S A' → ℂ) (φ₁ : WO A A' → ℂ) (φ₂ : WO S S → ℂ) (s₁ : WO S S) (s₂ : WO S A') (u₁ : WO A S) (u₂ : WO A A') : four d e C φ₁ φ₂ ((d.resS Y).symm ((e.resT S).symm (s₁, s₂), (e.resT A).symm (u₁, u₂))) = C u₁ s₂ * (φ₁ u₂ * φ₂ s₁)

theorem **symplOH_four** (x y : WO X Y) : symplOH X Y x y = symplOH A S (sandwich e.pA d.jB x) (sandwich e.pA d.jB y) + symplOH S A' (sandwich e.pB d.jA x) (sandwich e.pB d.jA y) + (symplOH A A' (sandwich e.pB d.jB x) (sandwich e.pB d.jB y) + symplOH S S (sandwich e.pA d.jA x) (sandwich e.pA d.jA y))

theorem **isThetaFunO_four** (h : IsBiTheta ψ A S A' C) (h₁ : IsThetaFunO ψ A A' φ₁) (h₂ : IsThetaFunO ψ S S φ₂) : IsThetaFunO ψ X Y (four d e C φ₁ φ₂) where

theorem **end_const_mul** {Z : Type} (L : Module.End ℂ (Z → ℂ)) (c : ℂ) (f : Z → ℂ) (z : Z) : L (fun y => c * f y) z = c * L f z

theorem **om_four** {χA χS χS' χA' : (AE D)ˣ →* Circle} {oma : Om A S} {omb : Om S A'} {Ω : Om X Y} {ΩS : Om S Y} {ΩA : Om A Y} {ω₁ : Om A A'} {ω₂ : Om S S} (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S A' χS' χA' omb) (hΩ : IsWeilDatumO ψ X Y (χS' * χA) (χS * χA') Ω) (hS : IsWeilDatumO ψ S Y χS' (χS * χA') ΩS) (hA : IsWeilDatumO ψ A Y χA (χS * χA') ΩA) (hω₁ : IsWeilDatumO ψ A A' χA χA' ω₁) (hω₂ : IsWeilDatumO ψ S S χS' χS ω₂) (hΦ : IsThetaFunO ψ X Y (four d e C φ₁ φ₂)) (a : GA D A) (h h' : GA D S) (b : GA D A') (w : WO A S) (w' : WO S A') : Ω (d.emb (h', a), e.emb (h, b)) (four d e C φ₁ φ₂) ((d.resS Y).symm ((e.resT S).symm (0, w'), (e.resT A).symm (w, 0))) = (ω₁ (a, b) φ₁ 0 * ω₂ (h', h) φ₂ 0) * omb (h', b) (fun v => oma (a, h) (fun u => C u v) w) w'

theorem **exists_kernel_ne_zero** {V W : HSp D} {χ χ' : (AE D)ˣ →* Circle} {om : Om V W} (h : IsWeilDatumO ψ V W χ χ' om) (p : GA D V × GA D W) : ∃ f, IsThetaFunO ψ V W f ∧ om p f 0 ≠ 0

theorem **hecke_mul** {χ χ' : (AE D)ˣ →* Circle} (h : IsHeckeChar D χ) (h' : IsHeckeChar D χ') : IsHeckeChar D (χ * χ')

theorem **continuous_biAct** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (χA χS χS' χA' : (AE D)ˣ →* Circle) (oma : Om A S) (omb : Om S A'), IsWeilDatumO ψ A S χA χS oma → IsWeilDatumO ψ S A' χS' χA' omb → ∀ C, IsBiTheta ψ A S A' C → Continuous fun x : (GA D A × GA D S) × (GA D S × GA D A') × WO A S × WO S A' => omb x.2.1 (fun v => oma x.1 (fun u => C u v) x.2.2.1) x.2.2.2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousIteratedWeilActionBiThetaStatement

theorem **_root_.GlobalTheory.T.continuous_iteratedWeilAction_biTheta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (χA χS χS' χA' : (AE D)ˣ →* Circle) (oma : Om A S) (omb : Om S A'), IsWeilDatumO ψ A S χA χS oma → IsWeilDatumO ψ S A' χS' χA' omb → ∀ C, IsBiTheta ψ A S A' C → Continuous fun x : (GA D A × GA D S) × (GA D S × GA D A') × WO A S × WO S A' => omb x.2.1 (fun v => oma x.1 (fun u => C u v) x.2.2.1) x.2.2.2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousKappa

@[simp] theorem **colTens_apply** (φ : SB (n → k)) (Ξ : SB (Matrix n n' k)) (x : Matrix n (n' ⊕ Unit) k) : (colTens φ Ξ).1 x = Ξ.1 x.toCols₁ * φ.1 (fun r => x r (Sum.inr ()))

@[simp] theorem **rowTens_apply** (φ : SB (n → k)) (Ξ : SB (Matrix n'' n k)) (x : Matrix (n'' ⊕ Unit) n k) : (rowTens φ Ξ).1 x = Ξ.1 x.toRows₁ * φ.1 (x (Sum.inr ()))

theorem **continuous_kappa** {ρ : ℕ} {B : BesLev D ρ} {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (S : HSp D) : Continuous (l.κ S)

theorem **integrable_contr** (hψ : IsStdChar D ψ (-1)) {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) {Ξ₁ : WO B.V₀ S → ℂ} {Ξ₂ : WO S B'.V₀ → ℂ} (h₁ : IsThetaFunO ψ B.V₀ S Ξ₁) (h₂ : IsThetaFunO ψ S B'.V₀ Ξ₂) (u : WO B.A S) (v : WO S B'.A) : Integrable (fun w => Ξ₂ ((B'.o.resT S).symm (v, w)) * Ξ₁ ((B.o.resS S).symm (u, l.κ S w))) (vL.μ.restrict vL.𝓕)

@[simp] theorem **contrBil_apply** (hψ : IsStdChar D ψ (-1)) {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) {M₁ M₂ : Type} [AddCommGroup M₁] [Module ℂ M₁] [AddCommGroup M₂] [Module ℂ M₂] (T₁ : M₁ →ₗ[ℂ] (WO B.V₀ S → ℂ)) (T₂ : M₂ →ₗ[ℂ] (WO S B'.V₀ → ℂ)) (h₁ : ∀ Φ, IsThetaFunO ψ B.V₀ S (T₁ Φ)) (h₂ : ∀ Φ, IsThetaFunO ψ S B'.V₀ (T₂ Φ)) (Ψ₁ : M₁) (Ψ₂ : M₂) : contrBil hψ l S vL T₁ T₂ h₁ h₂ Ψ₁ Ψ₂ = GlobalTheory.Unf.contr l S vL (T₁ Ψ₁) (T₂ Ψ₂)

theorem **exists_normPair** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψk : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψk) (hν : ν.IsAddHaarMeasure) (n : Type) [Fintype n] [DecidableEq n] : ∃ φ₀ φ₀' : SB (n → k), ∫ u, φ₀'.1 u * GlobalTheory.StepData.hat ψk ν (-1) φ₀.1 u ∂(Measure.pi fun _ => ν) = 1

theorem **contr_tens** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψk : AddChar k Circle) (ν : Measure k) {n n' n'' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (φ₀ φ₀' : SB (n → k)) (Ξ₁ : SB (Matrix n n' k)) (Ξ₂ : SB (Matrix n'' n k)) (q : Matrix n n' k × Matrix n'' n k) : GlobalTheory.LocFlat.contr ψk ν (colTens φ₀ Ξ₁).1 (rowTens φ₀' Ξ₂).1 q = Ξ₁.1 q.1 * Ξ₂.1 q.2 * ∫ u, φ₀'.1 u * GlobalTheory.StepData.hat ψk ν (-1) φ₀.1 u ∂(Measure.pi fun _ => ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousLeviBlock

theorem **lower_eq_zero** {p : G D n} (hp : p ∈ P D n) (i j : Fin n) : mat p (finSumFinEquiv (.inr i)) (finSumFinEquiv (.inl j)) = 0

theorem **blkA_mul** {p q : G D n} (hq : q ∈ P D n) : blkA (mat (p * q)) = blkA (mat p) * blkA (mat q)

theorem **blkA_one** : blkA (mat (1 : G D n)) = 1

theorem **continuous_mat** : Continuous (mat : G D n → Mx D (n + n) (n + n))

theorem **continuous_leviBlock** : Continuous (leviBlock (D

theorem **continuous_detGL** : Continuous (Matrix.GeneralLinearGroup.det : GL (Fin n) (AE D) → (AE D)ˣ)

theorem **isCompact_K** : IsCompact (K D n : Set (G D n))

theorem **isClosed_P** : IsClosed (P D n : Set (G D n))

theorem **exists_height** : ∀ (D : D7) (n : ℕ), ∃ H : G D n → ℝ≥0, IsHeight H

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousLocHom

theorem **mt_mul** (S : HSp D) (g h : GA D S) : mt S (g * h) = mt S g * mt S h

theorem **mt_one** (S : HSp D) : mt S 1 = 1

theorem **locHom_val** (q : SplitPt D k) (S : HSp D) (F : GlobalTheory.LocMod.Frame k S.rank Y) (g : GA D S) : (locHom q S F g).val = F.γ * (mt S g).map q.pr * F.β

theorem **eq_locHom_iff** (q : SplitPt D k) (S : HSp D) (F : GlobalTheory.LocMod.Frame k S.rank Y) (g : GA D S) (x : GL Y k) : (mt S g).map q.pr = F.β * x.val * F.γ ↔ x = locHom q S F g

theorem **continuous_locHom** (q : SplitPt D k) (S : HSp D) (F : GlobalTheory.LocMod.Frame k S.rank Y) : Continuous (locHom q S F)

theorem **vol_mul_relIndex** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] (μ : Measure G) [μ.IsMulLeftInvariant] (H H' : Subgroup G) (hle : H' ≤ H) (hc : IsCompact (H : Set G)) (ho : IsOpen (H' : Set G)) : μ.real H' * (H'.relIndex H : ℝ) = μ.real H

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousOfCond

theorem **adapted_of_normal** (hϖ : valuation k (ϖ : k) = uniformizer k) (hb : ∀ j c h, latOf (T.bas j c h) = latOf 1) (hs : ∀ j c h x, T.Phi j c h x ≠ 0 → IntM x) (hp : ∀ j c h x y, MaxM y → T.Phi j c h (x + y) = T.Phi j c h x) {M : ℕ} (hM : s + 1 ≤ M) {g : GL (X s ⊕ Fin (0 + 1)) k} (hg : IsNormalRep T ϖ s 0 rfl g) (x₀ : Matrix (X (s + 1)) (X s) k) (y : X (s + 1) → k) : (T.Psi g x₀ y ≠ 0 → y ∈ box _ (ϖ ^ M)⁻¹) ∧ ∀ z ∈ box _ (ϖ ^ M), T.Psi g x₀ (y + z) = T.Psi g x₀ y

theorem **val_unif** (h : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) : valuation k (ϖ : k) = uniformizer k

theorem **continuous_of_cond** {ψ : AddChar k Circle} (h : HasConductor ψ 1) : Continuous ψ

theorem **exists_system** {σ₀ : X.K →+* ℂ} {P : rowSG Pr sel mu} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : X.E →+* k} {W : TwData k} [MeasurableSpace k] [BorelSpace k] {ν : Measure k} (hν : IsUnitVol ν) (hA : AtTwist (R

theorem **idx_succ** (hX : X.Ok) (hPr : Pr.Ok) (hCh : Ch.Ok) (hsel : IsSel X sel) (hs : cst.sharp = true) (hL : L.Ok Ch sel mu cst G s) {j : ℕ} (hj : j < s) (hne : Nonempty (L.β j)) (h : ∀ P : rowSG Pr sel mu, P.1 ∉ Pr.vGs → ∃ γ : AutY X R (j + 1), Bdd Pr (j + 1) P.1 γ ∧ ∃ al : L.β j, FiniteSumNonzeroLargeDepth Ch sel mu cst j P (L.h j al P) γ) : Nonempty (L.β (j + 1))

theorem **isGL_unique** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : X.E →+* k} {j c : ℕ} {γ γ' : AutY X R j} {g g' : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin c) k} (h : IsGL ι γ g) (h' : IsGL ι γ' g') : γ = γ' ↔ g = g'

theorem **exists_bij** (hX : X.Ok) (hPr : Pr.Ok) (hCh : Ch.Ok) (hsel : IsSel X sel) (hs : cst.sharp = true) (hL : L.Ok Ch sel mu cst G s) (hprod : IdxIsProd sel mu s L) (hone : ∀ j b (P : rowSG Pr sel mu), P.1 ∈ Pr.vGs → L.h j b P = 1) {T : Type} (P : T → rowSG Pr sel mu) (hinj : Function.Injective P) (hoff : ∀ i, (P i).1 ∉ Pr.vGs) (hall : ∀ Q : rowSG Pr sel mu, Q.1 ∉ Pr.vGs → ∃ i, P i = Q) (C : T → ℕ → Type) (Rel : ∀ i j, AutY X R j → C i j → Prop) (ha : ∀ i j, j ≤ s → ∀ b : L.β j, ∃ y, Rel i j (L.h j b (P i)) y) (hb : ∀ i j, j ≤ s → Nonempty (L.β j) → ∀ y, ∃ b, Rel i j (L.h j b (P i)) y) (hc : ∀ i j, j ≤ s → ∀ γ γ' y y', Rel i j γ y → Rel i j γ' y' → (γ = γ' ↔ y = y')) (hd : ∀ i j, j < s → Nonempty (L.β j) → Nonempty (C i (j + 1)) → ∃ γ : AutY X R (j + 1), Bdd Pr (j + 1) (P i).1 γ ∧ ∃ al : L.β j, FiniteSumNonzeroLargeDepth Ch sel mu cst j (P i) (L.h j al (P i)) γ) (he : ∀ i j, j < s → None

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousPf

theorem **PM_map** {r : ℕ} {B : Type} [CommRing B] (f : AE D →+* B) (M : Matrix (Fin r) (Fin r) (AE D)) : (PM M).map f = f (eps D) • M.map f + f (1 − eps D) • (1 : Matrix (Fin r) (Fin r) B)

theorem **cT_PM** {r : ℕ} (M : Matrix (Fin r) (Fin r) (AE D)) : cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF D.host.F) D.sp.cm.sig) (PM M) = eps D • cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF D.host.F) D.sp.cm.sig) M + (1 − eps D) • (1 : Matrix (Fin r) (Fin r) (AE D))

theorem **PM_unit** {r : ℕ} (J M : Matrix (Fin r) (Fin r) (AE D)) (hM : cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF D.host.F) D.sp.cm.sig) M * J * M = J) : cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF D.host.F) D.sp.cm.sig) (PM M) * J * PM M = J

theorem **PGL_mem** (V : HSp D) (g : GL (Fin V.rank) (AE D)) (hg : g ∈ HR (AF D.host.F) D.sp.cm.sig V.gram) : PGL D V.rank g ∈ HR (AF D.host.F) D.sp.cm.sig V.gram

theorem **mt_pf** (V : HSp D) (h : GA D V) : Formalism.L2.Theta.mt V (pf V h) = PM (Formalism.L2.Theta.mt V h)

theorem **pf_one** (V : HSp D) : pf V 1 = 1

theorem **pf_fin** (V : HSp D) (h : GA D V) : pf V h ∈ GfC V

theorem **finM_pf** (V : HSp D) (h : GA D V) : finM V rfl (pf V h) = finM V rfl h

theorem **continuous_pf** (V : HSp D) : Continuous (pf V)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousPhiPrV

theorem **redM_eq_one** (v : HeightOneSpectrum (𝒪 F)) (X : Matrix (Fin n) (Fin n) (v.adicCompletion F)) (hX : ∀ i j, Valued.v (X i j − (1 : Matrix (Fin n) (Fin n) (v.adicCompletion F)) i j) < 1) : redM F v X = 1

theorem **continuous_phi_prV** (G : GLoc F E σ gp) (a : Fin 2) (v : HeightOneSpectrum (𝒪 F)) (h : (gp a : 𝒪 F) ∈ v.asIdeal) : letI

theorem **isOpen_ker_chi** {S : StepLevel F E σ J T f bad} {G : GLoc F E σ gp} {H : ∀ (a : Fin 2) (v : HeightOneSpectrum (𝒪 F)), (gp a : 𝒪 F) ∈ v.asIdeal → Subgroup (GL (Fin n) (kv F v))} {lam : ∀ a v h, H a v h →* ℂˣ} (L : GLevH S G H lam) (hgp : ∀ a, gp a ≠ 0) : letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousSingleAt

theorem **continuous_single_at** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : Continuous (singleV F v)

theorem **continuous_eval_at** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) : Continuous (evalV F v)

theorem **mem_integer_iff** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (a : Fv F v) : a ∈ 𝒪[Fv F v] ↔ a ∈ v.adicCompletionIntegers F

theorem **splitAt_of** {ε : AF D.host.F} (h : IsSplitIdem D e) (hε : IsPlaceIdem D v ε) (he : e + cA D e = algebraMap (AF D.host.F) (AE D) ε) : SplitAt D v e

theorem **sec_apply** (a : Fv D.host.F v) : sec D v e a = fj D (singleV D.host.F v a) ∗ e

theorem **continuous_sec** : Continuous (sec D v e)

theorem **continuous_comp** (he : SplitAt D v e) : Continuous (comp he)

theorem **exists_splitPt** : ∀ (D : D7) (v : Pl D) (e : AE D) (ε : AF D.host.F), IsSplitIdem D e → IsPlaceIdem D v ε → e + cA D e = algebraMap (AF D.host.F) (AE D) ε → ∃ p : SplitPt D (Fv D.host.F v), p.e = e ∧ p.v = v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousSplitPlaceHermitianFrameGLEmbedding

theorem **gram_det_isUnit** {E : Type} [Field E] [NumberField E] [StarRing E] (S : HSpace E) : IsUnit S.gram.det

theorem **continuous_of_parts** {A T m : Type} [CommRing A] [TopologicalSpace A] [IsTopologicalRing A] [TopologicalSpace T] [Fintype m] [DecidableEq m] (σ : A →+* A) (hσ : Continuous σ) (e : A) (J : Matrix m m A) (hJ : IsUnit J.det) (M N : T → Matrix m m A) (hU : ∀ t, Formalism.UG.cT σ (N t) ∗ J ∗ N t = J) (hNM : ∀ t, N t ∗ M t = 1) (hs : ∀ t i j, (e + σ e) ∗ (M t − 1) i j = (M t − 1) i j) (hM : Continuous fun t => e • M t) (hN : Continuous fun t => e • N t) : Continuous M

theorem **continuous_sigR** {F E : Type} [Field F] [Field E] [Algebra F E] (R : Type) [CommRing R] [Algebra F R] [TopologicalSpace R] [IsTopologicalRing R] (σ : E →ₐ[F] E) : Continuous (sigR R σ)

theorem **unitary_mt** {D : D7} {S : HSp D} (x : GA D S) : Formalism.UG.cT (cA D) (mt S x) ∗ S.gram.map (incA D) ∗ mt S x = S.gram.map (incA D)

theorem **mt_inv_mul** {D : D7} {S : HSp D} (x : GA D S) : mt S x⁻¹ ∗ mt S x = 1

theorem **gram_map_det_isUnit** {D : D7} (S : HSp D) : IsUnit (S.gram.map (incA D)).det

theorem **e_mul_entry** (β : Frame p S n) (g : GL n k) (i j : Fin S.rank) : p.e ∗ mt S (β.gl g) i j = p.sec ((β.b ∗ g.val ∗ β.b') i j)

theorem **continuous_e_smul** (β : Frame p S n) : Continuous fun g : GL n k => p.e • mt S (β.gl g)

theorem **continuous_mt** (β : Frame p S n) : Continuous fun g : GL n k => mt S (β.gl g)

theorem **continuous_gl** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (β : Frame p S n), Continuous β.gl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousSplitPlaceHermitianFrameGLEmbeddingStatement

theorem **_root_.GlobalTheory.T.continuous_splitPlaceHermitianFrame_GL_embedding** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (β : Frame p S n), Continuous β.gl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousStandardUnitaryChartAndSymm

theorem **continuous_mat_pt** (D : D7) (n : ℕ) : Continuous (mat : G D n → _)

theorem **continuous_mat_pt_inv** (D : D7) (n : ℕ) : Continuous (fun g : G D n => mat g⁻¹)

theorem **continuous_of_mat_pt** {D : D7} {n : ℕ} {X : Type} [TopologicalSpace X] {φ : X → G D n} (h : Continuous fun x => mat (φ x)) (h' : Continuous fun x => mat (φ x)⁻¹) : Continuous φ

theorem **chart_continuous** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → Continuous κ ∧ Continuous κ.symm

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousStandardUnitaryChartAndSymmStatement

theorem **_root_.GlobalTheory.T.continuous_standardUnitaryChart_and_symm** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → Continuous κ ∧ Continuous κ.symm

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousSwapR

theorem **continuous_bc** {A B : Type} [CommRing A] [CommRing B] [TopologicalSpace A] [TopologicalSpace B] {n : ℕ} (φ : A →+* B) (h : Continuous φ) : Continuous (bc (n

theorem **bc_mem** {A B : Type} [CommRing A] [CommRing B] {n : ℕ} (σ : A →+* A) (σ' : B →+* B) (φ : A →+* B) (h : ∀ x, σ' (φ x) = φ (σ x)) (J : Matrix (Fin n) (Fin n) A) (J' : Matrix (Fin n) (Fin n) B) (hJ : J.map φ = J') {g : GL (Fin n) A} (hg : g ∈ UJ σ J) : bc φ g ∈ UJ σ' J'

theorem **continuous_swapR** : @Continuous _ _ _ (tensTopL (F

theorem **swapR_sig** (x : R ⊗[F] E) : Formalism.UG.sigR σ R (swapR R x) = swapR R (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR R σ x)

theorem **exists_finIso** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n), letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousTraceMul

theorem **GlobalTheory.Core.Rk.SplitPsi.continuous_trace_mul** (D : D7) (a : AE D) : Continuous fun z : AE D => Algebra.trace (AF D.host.F) (AE D) (a ∗ z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousWeilCoefficientFockVectors

theorem **continuous_l2_weil_action_ptAt** : ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), ∀ i t σ s P cμ cη m ω BW f₀ f c io, IsLiftDataC R F i t σ s P cμ cη m ω BW f₀ f c io → ∀ (μ : Measure (WO (Vc D i) (Wc D i))) (𝒳 : Set (WO (Vc D i) (Wc D i))), IsCovolOne (ratL (Wc D i) (Vc D i)) μ 𝒳 → ∀ Q₁ Q₂, Continuous fun g : GA D (Wc D i) => l2 μ 𝒳 (ω (1, g) (ptAt R i σ BW f Q₁)) (ptAt R i σ BW f Q₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousWeilCoefficientFockVectorsStatement

theorem **_root_.GlobalTheory.T.continuous_weilCoefficient_fockVectors** : ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), ∀

i t σ s P cμ cη m ω BW f₀ f c io, IsLiftDataC R F i t σ s P cμ cη m ω BW f₀ f c io → ∀ (μ : Measure (WO (Vc D i) (Wc D i))) (𝒳 : Set (WO (Vc D i) (Wc D i))), IsCovolOne (ratL (Wc D i) (Vc D i)) μ 𝒳 → ∀ Q₁ Q₂, Continuous fun g : GA D (Wc D i) => l2 μ 𝒳 (ω (1, g) (ptAt R i σ BW f Q₁)) (ptAt R i σ BW f Q₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousWeilSchrodingerStandardUnnEvalZero

theorem **weilSch_apply_zero_continuous** : ∀ (D : D7) (n : ℕ) (V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (Mx D V.rank n)) (𝓕 : Set (Mx D V.rank n)), IsCovolOne (ratMx D V.rank n) μ 𝓕 → ∀ ω : G D n →* Module.End ℂ (schwartzA θ V.rank n), IsWeilSch ψ χ θ V μ ω → ∀ Φ : schwartzA θ V.rank n, Continuous fun g : G D n => (ω g Φ).1 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContinuousWeilSchrodingerStandardUnnEvalZeroStatement

theorem **_root_.GlobalTheory.T.continuous_weilSchrodinger_standardUnn_eval_zero** : ∀ (D : D7) (n : ℕ) (V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (Mx D V.rank n)) (𝓕 : Set (Mx D V.rank n)), IsCovolOne (ratMx D V.rank n) μ 𝓕 → ∀ ω : G D n →* Module.End ℂ (schwartzA θ V.rank n), IsWeilSch ψ χ θ V μ ω → ∀ Φ : schwartzA θ V.rank n, Continuous fun g : G D n => (ω g Φ).1 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContractionOfVacuumCombinationsProof

theorem **mxAt_mul_right** {P : Pl D → Prop} {m n r : ℕ} {u : Mx D m n} (hu : MxAt P u) (N : Mx D n r) : MxAt P (u * N)

theorem **mxAt_mul_left** {P : Pl D → Prop} {m n r : ℕ} (N : Mx D r m) {u : Mx D m n} (hu : MxAt P u) : MxAt P (N * u)

theorem **mxAt_sub** {P : Pl D → Prop} {m n : ℕ} {u v : Mx D m n} (hu : MxAt P u) (hv : MxAt P v) : MxAt P (u - v)

theorem **mxAt_neg** {P : Pl D → Prop} {m n : ℕ} {u : Mx D m n} (hu : MxAt P u) : MxAt P (-u)

theorem **fin_of_mxAt** {P : Pl D → Prop} {m n : ℕ} {u : Mx D m n} (hu : MxAt P u) : u.map (infPart D) = 0

theorem **contr_smul_left** {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (c : ℂ) (f₁ : WO B.V₀ S → ℂ) (f₂ : WO S B'.V₀ → ℂ) : GlobalTheory.Unf.contr l S vL (c • f₁) f₂ = c • GlobalTheory.Unf.contr l S vL f₁ f₂

theorem **biComb_single** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (a : WO A S) (a' : WO S A') (r : ℂ) (C : WO A S → WO S A' → ℂ) (u : WO A S) (v : WO S A') : biComb ψ A S A' (Finsupp.single a r) (Finsupp.single a' 1) C u v = r * heisO ψ A S a (fun u' => heisO ψ S A' a' (C u') v) u

theorem **symplOH_self** {S T : HSp D} (x : WO S T) : symplOH S T x (-x) = 0

theorem **heisO_neg_cancel** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (x : WO S T) (g : WO S T → ℂ) : heisO ψ S T x (heisO ψ S T (-x) g) = g

theorem **line_step** {ρ σ : ℕ} {B : BesLev D ρ} {B' : BesLev D σ} (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (l : Link B B'.L B'.ζ) (S : HSp D) (vL : VolM D B'.L.rank S.rank) (f₁ : WO B.V₀ S → ℂ) (f₂ : WO S B'.V₀ → ℂ) (hf₁ : IsThetaFunO ψ B.V₀ S f₁) (hf₂ : IsThetaFunO ψ S B'.V₀ f₂) (y : WO B.L S) (w : WO S B'.L) (hy : ((B.o.resS S).symm (0, y)).map (infPart D) = 0) (hk : (-(B.o.resS S).symm (0, l.κ S w)).map (infPart D) = 0) (c : ℂ) (hc : GlobalTheory.Unf.contr l S vL (heisO ψ B.V₀ S ((B.o.resS S).symm (0, y - l.κ S w)) f₁) f₂ = c • GlobalTheory.Unf.contr l S vL f₁ f₂) : GlobalTheory.Unf.contr l S vL (heisO ψ B.V₀ S ((B.o.resS S).symm (0, y)) f₁) (heisO ψ S B'.V₀ ((B'.o.resT S).symm (0, w)) f₂) = (ψ (symplOH B.V₀ S (-(B.o.resS S).symm (0, l.κ S w)) ((B.o.resS S).symm (0, y))) * c) • GlobalTheory.Unf.contr l S vL f₁ f₂

theorem **contrOp** : ∀ D : D7, ContrOp D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ContractionOfVacuumCombinationsProofStatement

theorem **_root_.GlobalTheory.T.contractionOfVacuumCombinations_proof** : ∀ D : D7, GlobalTheory.FinPl.ContrOp D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConvergentInnerProductConjugateOrientationWeil

theorem **integrable_star_comp** {X : Type*} [MeasurableSpace X] {μ : Measure X} {f : X → ℂ} (hf : Integrable f μ) : Integrable (fun x => conj (f x)) μ

theorem **memL2_conj_comp** (V : HSp D) (μ : Measure (GA D V)) {f : GA D V → ℂ} (hf : MemL2Quotient (GQ V) μ f) : MemL2Quotient (GQ V) μ (fun x => conj (f x))

theorem **petersson_star_comp** (V : HSp D) (μ : Measure (GA D V)) (f₁ f₂ : GA D V → ℂ) : petersson V μ (fun x => conj (f₁ x)) (fun x => conj (f₂ x)) = conj (petersson V μ f₁ f₂)

theorem **innerProduct_convergent_conj** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.WeilSiegelFormulaUnitary → GlobalTheory.Published.SiegelEisensteinContinuation → ∀ D : D7, ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (sA : RealMat A) (sB : RealMat B) (t : ℕ) (cA : CptF A) (K₁ K₂ : GA D A → GA D B → ℂ) (c : GA D A → ℂ) (m : Measure (GA D A)) (ν : Measure (GA D B)) (f₁ f₂ : GA D A → ℂ), HidaSetting θ A B sA sB t cA → (wittIndex B = 0 ∨ wittIndex B < t) → A.rank < t → GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.IsKernelPair ψ θ A B χA χB sA sB false (fun _ => True) (fun _ => True) K₁ K₂ c → m.IsHaarMeasure → ν.IsHaarMeasure → f₁ ∈ A0 θ A cA → f₂ ∈ A0 θ A cA → GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.LiftConv m ν K₁ f₁ ∧ GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.LiftConv m ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ConvergentInnerProductConjugateOrientationWeilStatement

theorem **_root_.GlobalTheory.T.convergentInnerProduct_conjugateOrientation_weil** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.WeilSiegelFormulaUnitary → GlobalTheory.Published.SiegelEisensteinContinuation → ∀ D : D7, ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (sA : RealMat A) (sB : RealMat B) (t : ℕ) (cA : CptF A) (K₁ K₂ : GA D A → GA D B → ℂ) (c : GA D A → ℂ) (m : Measure (GA D A)) (ν : Measure (GA D B)) (f₁ f₂ : GA D A → ℂ), HidaSetting θ A B sA sB t cA → (wittIndex B = 0 ∨ wittIndex B < t) → A.rank < t → GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.IsKernelPair ψ θ A B χA χB sA sB false (fun _ => True) (fun _ => True) K₁ K₂ c → m.IsHaarMeasure → ν.IsHaarMeasure → f₁ ∈ A0 θ A cA → f₂ ∈ A0 θ A cA → GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.LiftConv m ν K₁ f₁ ∧ GlobalTheory.IndefiniteRowSideF

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoprimeScalingIdealValuationLe

theorem **natInv_mem_adicIntegers** (F : Type) [Field F] [NumberField F] (T : ℕ) (v : HeightOneSpectrum (𝒪 F)) (h : ((T : ℕ) : 𝒪 F) ∉ v.asIdeal) : algebraMap F (v.adicCompletion F) ((T : F)⁻¹) ∈ v.adicCompletionIntegers F

theorem **natInv_smul_mem_intAf** (F : Type) [Field F] [NumberField F] (T : ℕ) (v : HeightOneSpectrum (𝒪 F)) (h : ((T : ℕ) : 𝒪 F) ∉ v.asIdeal) (α : Af F) (h1 : α ∈ intAf F) (h2 : ∀ w, w ≠ v → evalV F w α = 0) : ((T : F)⁻¹ : F) • α ∈ intAf F ∧ ∀ w, w ≠ v → evalV F w (((T : F)⁻¹ : F) • α) = 0

theorem **smul_pairA_zero** (F : Type) [Field F] [NumberField F] (t : F) (α : Af F) : t • pairA F 0 α = pairA F 0 (t • α)

theorem **pairA_zero_tmul_smul** (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] (t : F) (α : Af F) (c : E) : pairA F 0 α ⊗ₜ[F] (t • c) = pairA F 0 (t • α) ⊗ₜ[F] c

theorem **idV_natInv_mul_le** (D : D7) (𝔠 : FI D) (T : ℕ) (v : Pl D) (h : ((T : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal) : idV (FractionalIdeal.spanSingleton _ ((T : D.host.E)⁻¹) * 𝔠) v ≤ idV 𝔠 v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankOneIsEquivalenceNonzeroGramTerms

theorem **corankOne_isEquivalence_nonzeroGram_terms** : CorankOneIsEquivalenceOnRow

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankOneOfSameFlagDoubleCoset

theorem **frame_mapLF** (γ : S.V ≃ₗ[E] S.V) (hγ : γ ∈ Formalism.HermitianLatticesTheory.isom S.hT) (F : LFlag S r T) (a : Fin (r + 1)) : frame (mapLF γ F) a = γ (frame F a)

theorem **Xflag_mapLF** (γ : S.V ≃ₗ[E] S.V) (hγ : γ ∈ Formalism.HermitianLatticesTheory.isom S.hT) (F : LFlag S r T) (j : ℕ) : Xflag (mapLF γ F) j = (Xflag F j).map (γ : S.V →ₗ[E] S.V)

theorem **orth_mapLF** (γ : S.V ≃ₗ[E] S.V) (hγ : γ ∈ Formalism.HermitianLatticesTheory.isom S.hT) (F : LFlag S r T) (x : S.V) (hx : x ∈ orth S (Xsp (mapLF γ F))) : γ.symm x ∈ orth S (Xsp F)

theorem **conj_mem_NRat** {D : D7} {T : D.host.E} (γ : (VS D).V ≃ₗ[D.host.E] (VS D).V) (hγ : γ ∈ Formalism.HermitianLatticesTheory.isom (VS D).hT) (F : LFlag (VS D) D.R.rS T) (n : (VS D).V ≃ₗ[D.host.E] (VS D).V) (hn : n ∈ NRat F) : γ * n * γ⁻¹ ∈ NRat (mapLF γ F)

theorem **corankOne_of_same_flagDoubleCoset** : ∀ (D : D7) (X : HidaSetting D) (Y : Full X) (u u' : D.what₂.Word), ∀ t t' : TauC Y u u', t.G ≠ 0 → (Y.tO u u').rowOf t = (Y.tO u u').rowOf t' → nmlOf Y u u' t = nmlOf Y u u' t' → (Y.tO u u').s0 t t' = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankOnePairsFamilyBoundsOfSiegelWeil

theorem **oneOn_singleton_of_siegelWeil** : HeckeCharacterZeroLocus.SiegelWeilU.SiegelWeilUnitaryConvergent → ∀ c : ℝ, GlobalTheory.Core.Diag.OneOn {c} true ∧ GlobalTheory.Core.Diag.OneOn {c} false

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankOnePairsFamilyBoundsOfSiegelWeilStatement

theorem **_root_.GlobalTheory.T.corankOnePairsFamilyBounds_of_siegelWeil** : HeckeCharacterZeroLocus.SiegelWeilU.SiegelWeilUnitaryConvergent → ∀ c : ℝ, GlobalTheory.Core.Diag.OneOn {c} true ∧ GlobalTheory.Core.Diag.OneOn {c} false

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankSizeHypothesisExplicitConstantSiegelMass

theorem **xiSizes_explicit** : Formalism.HermitianLatticesTheory.SiegelMassRatio → ∀ C : ℝ, 408 * (Real.exp 1 * Real.pi) ≤ C → ∀ (D : D7) (large₂ : SizeData₂ D) (u u' : D.what₂.Word) (C' Cl Cg cb Cu cT CT C0 A : ℝ), 1 ≤ Cl → Cnt.Hyp D large₂ u u' C' Cl Cg cb Cu cT CT C0 A → Formalism.HermitianLatticeCounting.LatticeSumBounds.XiHyp C D.R.k (2 * D.R.rS + 2) D.R.f large₂.sc.b D.host.v (Cnt.NT D u) (BridgeLemma.XiB D large₂ u C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankSizeHypothesisExplicitConstantSiegelMassStatement

theorem **_root_.GlobalTheory.T.corankSizeHypothesis_explicitConstant_siegelMass** : Formalism.HermitianLatticesTheory.SiegelMassRatio → ∀ C : ℝ, 408 * (Real.exp 1 * Real.pi) ≤ C → ∀ (D : D7) (large₂ : SizeData₂ D) (u u' : D.what₂.Word) (C' Cl Cg cb Cu cT CT C0 A : ℝ), 1 ≤ Cl → Cnt.Hyp D large₂ u u' C' Cl Cg cb Cu cT CT C0 A → Formalism.HermitianLatticeCounting.LatticeSumBounds.XiHyp C D.R.k (2 * D.R.rS + 2) D.R.f large₂.sc.b D.host.v (Cnt.NT D u) (BridgeLemma.XiB D large₂ u C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankSumsBoundWithExpLoss

theorem **rpow_le_rpow_both** {x y u w : ℝ} (hx : 1 ≤ x) (hxy : x ≤ y) (hu : 0 ≤ u) (huw : u ≤ w) : x ^ u ≤ y ^ w

theorem **bigH_one_le** (C : ℝ) (hC : 0 ≤ C) (m f : ℕ) {b : ℝ} (hb : 1 ≤ b) : 1 ≤ Formalism.HermitianLatticeCounting.LatticeSumBounds.bigH C m f b

theorem **bigH_mono** {C D : ℝ} (hC : 0 ≤ C) (hCD : C ≤ D) (m f : ℕ) {b : ℝ} (hb : 1 ≤ b) : Formalism.HermitianLatticeCounting.LatticeSumBounds.bigH C m f b ≤ Formalism.HermitianLatticeCounting.LatticeSumBounds.bigH D m f b

theorem **xiHyp_mono** {C D : ℝ} (hC : 1 ≤ C) (hCD : C ≤ D) {k m f : ℕ} {b v NT : ℝ} {Xi : ℕ → ℝ} (hb : 1 ≤ b) (hNT : 1 ≤ NT) (h : XiHyp C k m f b v NT Xi) : XiHyp D k m f b v NT Xi

theorem **params_mono** {C D K K₀ : ℝ} (hC : 1 ≤ C) (hCD : C ≤ D) (hK : K ≤ K₀) {k m f : ℕ} {b g v NT : ℝ} (h : Params C K₀ k m f b g v NT 1) : Params D K k m f b g v NT 1

theorem **gam_absorb** {C Gam : ℝ} (hC : 0 ≤ C) {k f : ℕ} (hf : 1 ≤ f) (hG : Gam ≤ Real.exp ((k : ℝ) / 8)) : C ^ (k * f) * Gam ≤ (Real.exp 1 * C) ^ (k * f)

theorem **ndRow_mono** {C Gam : ℝ} (hC : 1 ≤ C) {k m f s : ℕ} {b g v : ℝ} {Xi : ℕ → ℝ} (hf : 1 ≤ f) (hb : 1 ≤ b) (hg : 0 < g) (hv : 0 ≤ v) (hXi : 0 ≤ Xi s) (hG : Gam ≤ Real.exp ((k : ℝ) / 8)) : ndRow C k m f s b g v Gam Xi ≤ ndRow (Real.exp 1 * C) k m f s b g v 1 Xi

theorem **degRow_mono** {C Gam : ℝ} (hC : 1 ≤ C) {k m f s : ℕ} {b g v : ℝ} {Xi : ℕ → ℝ} (hf : 1 ≤ f) (hb : 1 ≤ b) (hg : 0 < g) (hv : 0 ≤ v) (hXi : 0 ≤ Xi s) (hG : Gam ≤ Real.exp ((k : ℝ) / 8)) : degRow C k m f s b g v Gam Xi ≤ degRow (Real.exp 1 * C) k m f s b g v 1 Xi

theorem **corankSums_expLoss** : ∀ C : ℝ, 1 ≤ C → ∃ K₀ : ℝ, ∀ (k m f : ℕ) (b g v NT Gam : ℝ) (Xi : ℕ → ℝ), Formalism.HermitianLatticeCounting.LatticeSumBounds.Params C K₀ k m f b g v NT 1 → Formalism.HermitianLatticeCounting.LatticeSumBounds.XiHyp C k m f b v NT Xi → 0 ≤ Gam → Gam ≤ Real.exp ((k : ℝ) / 8) → ∑ s ∈ Finset.Icc 2 k, Formalism.HermitianLatticeCounting.LatticeSumBounds.ndRow C k m f s b g v Gam Xi ≤ (Real.sqrt g)⁻¹ ^ (k * f) ∧ ∑ s ∈ Finset.Icc 2 k, Formalism.HermitianLatticeCounting.LatticeSumBounds.degRow C k m f s b g v Gam Xi ≤ Real.exp (-((k : ℝ) * f * Real.log k) / 2)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankSumsBoundWithExpLossStatement

theorem **_root_.GlobalTheory.T.corankSums_bound_with_expLoss** : ∀ C : ℝ, 1 ≤ C → ∃ K₀ : ℝ, ∀ (k m f : ℕ) (b g v NT Gam : ℝ) (Xi : ℕ → ℝ), Formalism.HermitianLatticeCounting.LatticeSumBounds.Params C K₀ k m f b g v NT 1 → Formalism.HermitianLatticeCounting.LatticeSumBounds.XiHyp C k m f b v NT Xi → 0 ≤ Gam → Gam ≤ Real.exp ((k : ℝ) / 8) → ∑ s ∈ Finset.Icc

2 k, Formalism.HermitianLatticeCounting.LatticeSumBounds.ndRow C k m f s b g ν Gam Xi ≤ (Real.sqrt g)⁻¹ ^ (k * f) ∧ ∑ s ∈ Finset.Icc 2 k, Formalism.HermitianLatticeCounting.LatticeSumBounds.degRow C k m f s b g ν Gam Xi ≤ Real.exp (-((k : ℝ) * f * Real.log k) / 2)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CorankTwoCountBoundsOfRowCountBoundsUniform

theorem **corankTwoCountBounds_of_rowCountBoundsUniform** : ∀ S : Set ℝ, CntOn S → CountBoundOn S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoreIffCollapsesWith

theorem **exists_average** (ρ : Representation ℂ G V) (hρ : IsSmoothRep ρ) (K : Subgroup G) (hK : IsCompact (K : Set G)) : ∃ e : V →ₗ[ℂ] V, (∀ g ∈ K, ∀ w, e (ρ g w) = e w) ∧ (∀ w, (∀ g ∈ K, ρ g w = w) → e w = w) ∧ ∀ j : V →ₗ[ℂ] V, (∀ g w, j (ρ g w) = ρ g (j w)) → ∀ w, j (e w) = e (j w)

theorem **exists_equivariant** {H : Type} [Group H] (ρ : Representation ℂ G V) (hρ : IsSmoothRep ρ) (π : Representation ℂ H V) (hcomm : ∀ g h w, π h (ρ g w) = ρ g (π h w)) (K : Subgroup G) (hKo : IsOpen (K : Set G)) (hK : IsCompact (K : Set G)) : ∃ (P : Type) (_ : AddCommGroup P) (_ : Module ℂ P) (π' : Representation ℂ G P) (θ : P →ₗ[ℂ] V →ₗ[ℂ] V) (φ : P), IsSmoothRep π' ∧ (∀ g h T w, θ (π' g T) (ρ g (π h w)) = π h (θ T w)) ∧ ∀ w, (∀ g ∈ K, ρ g w = w) → θ φ w = w

theorem **omRep_smooth** {μ : kˣ →* ℂˣ} (hμ : IsOpen (μ.ker : Set kˣ)) : IsSmoothRep (omRep (n

theorem **Core.collapsesWith** {Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)} (H : Core ν μX μG J bas ψ μ lam Φ.1 κ) : CollapsesWith ν μX μG J bas ψ μ lam Φ κ

theorem **isQuotInt_zero_iff** {Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)} (H : CollapsesWith ν μX μG J bas ψ μ lam Φ κ) (z : ℂ) : IsQuotInt ν μX μG 1 (fun _ => 0) z ↔ z = 0

theorem **core_of_collapsesWith** (hc : 1 ≤ c) [ν.IsAddHaarMeasure] (hμ : IsOpen (μ.ker : Set kˣ)) (hJ : IsOpen (J : Set (GL (n ⊕ Fin c) k))) {Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)} (hΦ : HasTargetType μ J lam Φ.1) (H : CollapsesWith ν μX μG J bas ψ μ lam Φ κ) : Core ν μX μG J bas ψ μ lam Φ.1 κ

theorem **flat_eq_smul** (μ : kˣ →* ℂˣ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (h : GL (n ⊕ Fin c) k) : flat ψ ν μ Φ 1 h = (((μ (Matrix.GeneralLinearGroup.det h))⁻¹ : ℂˣ) : ℂ) • flat ψ ν 1 Φ 1 h

theorem **om_eq_smul** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (Φ : Matrix b a k → ℂ) (h : GL b k) : om 1 1 h Φ = (μ (Matrix.GeneralLinearGroup.det h) : ℂ) • om μ 1 h Φ

theorem **hasTargetType_untwist** {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (hΦ : HasTargetType μ J lam Φ) : HasTargetType 1 J (untwist μ lam) Φ

theorem **oneCoset_untwist** {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (hΦ : OneCoset ψ ν μ Φ J) : OneCoset ψ ν 1 Φ J

theorem **core_of_untwist** {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (H : Core ν μX μG J bas ψ 1 (untwist μ lam) Φ κ) : Core ν μX μG J bas ψ μ lam Φ κ

theorem **cjG_apply** (t h : GL (n ⊕ Fin c) k) : cjG t h = t * h * t⁻¹

theorem **Adapted.mem_N** {t : GL (n ⊕ Fin c) k} (ht : Adapted t) {u : GL (n ⊕ Fin c) k} : t * u * t⁻¹ ∈ N k n c 1 ↔ u ∈ N k n c 1

theorem **Adapted.inv** {t : GL (n ⊕ Fin c) k} (ht : Adapted t) : Adapted t⁻¹

theorem **Adapted.conj_mem_R** {t : GL (n ⊕ Fin c) k} (ht : Adapted t) {u : GL (n ⊕ Fin c) k} (hu : u ∈ R k n c 1) : t * u * t⁻¹ ∈ R k n c 1

theorem **Adapted.prX_cjJ** {t : GL (n ⊕ Fin c) k} (ht : Adapted t) (J : Subgroup (GL (n ⊕ Fin c) k)) : prX (Cgrp (cjJ t J) 1) = prX (Cgrp J 1)

theorem **Adapted.exists_map_volN** {t : GL (n ⊕ Fin c) k} (ht : Adapted t) (ν ν₀ : Measure k) [ν.IsAddHaarMeasure] [ν₀.IsAddHaarMeasure] : ∃ r : ℝ≥0, r ≠ 0 ∧ (∀ F : GL (n ⊕ Fin c) k → ℂ, ∫ u : N k n c 1, F (t * u * t⁻¹) ∂(volN ν₀ n c 1) = (r : ℝ) • ∫ u : N k n c 1, F u ∂(volN ν n c 1)) ∧ ∀ S S₀ : Subgroup (GL (n ⊕ Fin c) k), (∀ u ∈ N k n c 1, u ∈ S₀ ↔ t * u * t⁻¹ ∈ S) → volN ν₀ n c 1 (S₀.subgroupOf (N k n c 1)) = r * volN ν n c 1 (S.subgroupOf (N k n c 1))

theorem **isHaarMeasure_cj** (t : GL (n ⊕ Fin c) k) (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] : (μG.map (cjG t).symm).IsHaarMeasure

theorem **map_cjJ** (t : GL (n ⊕ Fin c) k) (μG : Measure (GL (n ⊕ Fin c) k)) (J : Subgroup (GL (n ⊕ Fin c) k)) : μG.map (cjG t).symm (cjJ t J) = μG J

theorem **Adapted.isQuotInt_iff** {t : GL (n ⊕ Fin c) k} (ht : Adapted t) {ν ν₀ : Measure k} {r : ℝ≥0} (hr : r ≠ 0) (H : ∀ F : GL (n ⊕ Fin c) k → ℂ, ∫ u : N k n c 1, F (t * u * t⁻¹) ∂(volN ν₀ n c 1) = (r : ℝ) • ∫ u : N k n c 1, F u ∂(volN ν n c 1)) (μX : Measure (GL n k)) (μG : Measure (GL (n ⊕ Fin c) k)) (f : GL (n ⊕ Fin c) k → ℂ) (z : ℂ) : IsQuotInt ν₀ (r⁻¹ • μX) (μG.map (cjG t).symm) 1 (fun h => f (cjG t h)) z ↔ IsQuotInt ν μX μG 1 f z

theorem **frame_conj_apply** (b : ℕ → kˣ) (u : GL (n ⊕ Fin c) k) (r s : n ⊕ Fin c) : (frame n c b * u * (frame n c b)⁻¹).val r s = Sum.elim (fun _ : n => (1 : k)) (fun i : Fin c => (b (i.val + 1) : k)) r * Sum.elim (fun _ : n => (1 : k)) (fun i : Fin c => (b (i.val + 1) : k)⁻¹) s * u.val r s

theorem **adapted_frame** (b : ℕ → kˣ) : Adapted (frame n c b)

theorem **simpleSum_frame** (β : kˣ) (u : GL (n ⊕ Fin c) k) : simpleSum (frame n c (fun i => 1⁻¹ * β⁻¹ ^ i) * u * (frame n c fun i => 1⁻¹ * β⁻¹ ^ i)⁻¹) = β * simpleSum u

theorem **hasConductor_mulShift** {ψ : AddChar k Circle} {β : kˣ} (h : HasConductor ψ β) : HasConductor (ψ.mulShift (β : k)) 1

theorem **exists_rescaled** (β : kˣ) (ν ν₀ : Measure k) [ν.IsAddHaarMeasure] [ν₀.IsAddHaarMeasure] {t : GL (n ⊕ Fin c) k} (ht : t = frame n c fun i => 1⁻¹ * β⁻¹ ^ i) {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (hΦ : Φ ∈ SB _) : ∃ Φ₀ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), (∀ ψ μ h, flat (ψ.mulShift (β : k)) ν₀ μ Φ₀ 1 h = flat ψ ν μ Φ 1 (t * h * t⁻¹)) ∧ ∀ μ h (z : ℂ), om μ 1 (t * h * t⁻¹) Φ = z • Φ → om μ 1 h Φ₀ = z • Φ₀

theorem **exists_reduction** (β : kˣ) (ν ν₀ : Measure k) [ν.IsAddHaarMeasure] [ν₀.IsAddHaarMeasure] (ψ : AddChar k Circle) (μ : kˣ →* ℂˣ) (J : Subgroup (GL (n ⊕ Fin c) k)) (lam : J →* ℂˣ) (bas : GL (n ⊕ Fin c) k) {Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ} (hΦ : Φ ∈ SB _) (h1 : HasTargetType μ J lam Φ) (h2 : OneCoset ψ ν μ Φ J) : ∃ (t : GL (n ⊕ Fin c) k) (r : ℝ≥0), ∃ Φ₀ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), Adapted t ∧ r ≠ 0 ∧ HasTargetType μ (cjJ t J) (cjLam t lam) Φ₀ ∧ OneCoset (ψ.mulShift (β : k)) ν₀ μ Φ₀ (cjJ t J) ∧ (∀ m : Measure (GL n k), IndefiniteRowSideFullHecke.FramePositiveSum.rescale ν₀ (t⁻¹ * bas) m = r⁻¹ • IndefiniteRowSideFullHecke.FramePositiveSum.rescale ν bas m) ∧ ∀ (μX : Measure (GL n k)) (μG : Measure (GL (n ⊕ Fin c) k)) (κ : ℝ), Core ν₀ (r⁻¹ • μX) (μG.map (cjG t).symm) (cjJ t J) (t⁻¹ * bas) (ψ.mulShift (β : k)) μ (cjLam t lam) Φ₀ κ → Core ν μX μG J bas ψ

theorem **core_iff_collapsesWith** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (hc : 1 ≤ c) (ν :

Measure k) [ν.IsAddHaarMeasure] (μX : Measure (GL n k)) (μG : Measure (GL (n ⊕ Fin c) k)) {J : Subgroup (GL (n ⊕ Fin c) k)} (hJ : IsOpen (J : Set (GL (n ⊕ Fin c) k))) (bas : GL (n ⊕ Fin c) k) (ψ : AddChar k Circle) {μ : kˣ →* ℂˣ} (hμ : IsOpen (μ.ker : Set kˣ)) (lam : J →* ℂˣ) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (hΦ : HasTargetType μ J lam Φ.1) (κ : ℝ) : Core ν μX μG J bas ψ μ lam Φ.1 κ ↔ CollapsesWith ν μX μG J bas ψ μ lam Φ κ

theorem **collapse_of_weak** (H : CollapseWeak) : SymChain.Collapse

theorem **isCO_cjJ** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} (t : GL (n ⊕ Fin c) k) {J : Subgroup (GL (n ⊕ Fin c) k)} (ho : IsOpen (J : Set (GL (n ⊕ Fin c) k))) (hc : IsCompact (J : Set (GL (n ⊕ Fin c) k))) : IndefiniteRowSideFullHecke.FramePositiveSum.IsCO (cjJ t J)

theorem **weak_of_collapse** (H : SymChain.Collapse) : CollapseWeak

theorem **collapse_of_exact** (H : SymChain.CollapseExact) : IndefiniteRowSideFullHecke.FlatOmegaSeries.Collapse

theorem **uniform_iff** : CollapseUniform ↔ IndefiniteRowSideFullHecke.FlatOmegaSeries.Collapse

theorem **collapse_compare** : (SymChain.Collapse ↔ CollapseWeak) ∧ (CollapseUniform ↔ IndefiniteRowSideFullHecke.FlatOmegaSeries.Collapse) ∧ (IndefiniteRowSideFullHecke.FlatOmegaSeries.Collapse → SymChain.Collapse) ∧ (SymChain.CollapseExact → IndefiniteRowSideFullHecke.FlatOmegaSeries.Collapse)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CosetSumAlongMulEquivSubgroups

private theorem **bijective_quot_mk_map_out** {G₁ G₂ : Type} [Group G₁] [Group G₂] (κ : G₁ ≃* G₂) (P₁ Γ₁ : Subgroup G₁) (P₂ Γ₂ : Subgroup G₂) (hP : ∀ g, g ∈ P₁ ↔ κ g ∈ P₂) (hΓ : ∀ g, g ∈ Γ₁ ↔ κ g ∈ Γ₂) : Function.Bijective fun q : _root_.GlobalTheory.AutomorphicL2Unitary.Quot (P₁.subgroupOf Γ₁) => _root_.GlobalTheory.AutomorphicL2Unitary.mk (P₂.subgroupOf Γ₂) ((κ (Quotient.out q : Γ₁), (hΓ _).1 (Quotient.out q : Γ₁).2) : Γ₂)

theorem **_root_.GlobalTheory.StdUnn.cosetSum_along_mulEquiv** : ∀ (G₁ G₂ : Type) [Group G₁] [Group G₂] (κ : G₁ ≃* G₂) (P₁ Γ₁ : Subgroup G₁) (P₂ Γ₂ : Subgroup G₂), (∀ g, g ∈ P₁ ↔ κ g ∈ P₂) → (∀ g, g ∈ Γ₁ ↔ κ g ∈ Γ₂) → ∀ (Φ₂ : G₂ → ℂ) (x : G₁) (c : ℂ) (Φ₁ : G₁ → ℂ), (∀ g, Φ₁ g = c * Φ₂ (κ (g * x))) → _root_.GlobalTheory.AutomorphicL2Unitary.LeftInv (P₁ ⊓ Γ₁) Φ₁ ∧ ∀ g, (_root_.GlobalTheory.Eis.CosetConv P₂ Γ₂ Φ₂ (κ (g * x)) → _root_.GlobalTheory.Eis.CosetConv P₁ Γ₁ Φ₁ g) ∧ _root_.GlobalTheory.Eis.cosetSum P₁ Γ₁ Φ₁ g = c * _root_.GlobalTheory.Eis.cosetSum P₂ Γ₂ Φ₂ (κ (g * x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CosetSumAlongMulEquivSubgroupsStatement

theorem **_root_.GlobalTheory.T.cosetSum_along_mulEquiv_subgroups** : ∀ (G₁ G₂ : Type) [Group G₁] [Group G₂] (κ : G₁ ≃* G₂) (P₁ Γ₁ : Subgroup G₁) (P₂ Γ₂ : Subgroup G₂), (∀ g, g ∈ P₁ ↔ κ g ∈ P₂) → (∀ g, g ∈ Γ₁ ↔ κ g ∈ Γ₂) → ∀ (Φ₂ : G₂ → ℂ) (x : G₁) (c : ℂ) (Φ₁ : G₁ → ℂ), (∀ g, Φ₁ g = c * Φ₂ (κ (g * x))) → LeftInv (P₁ ⊓ Γ₁) Φ₁ ∧ ∀ g, (GlobalTheory.Eis.CosetConv P₂ Γ₂ Φ₂ (κ (g * x)) → GlobalTheory.Eis.CosetConv P₁ Γ₁ Φ₁ g) ∧ GlobalTheory.Eis.cosetSum P₁ Γ₁ Φ₁ g = c * GlobalTheory.Eis.cosetSum P₂ Γ₂ Φ₂ (κ (g * x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoverDiscreteSeriesUnitarilyEquivOfSameParam

theorem **unitEquiv_of_dsUnique** : GlobalTheory.Dec.DSUnique → ∀ (p q : ℕ), 1 ≤ p + q → ∀ (T : GrpData p q) (R R' : DSRep T) (lam : Ix p q → ℚ), IsParam lam → HasParam R lam → HasParam R' lam → UnitEquiv R R'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoverDiscreteSeriesUnitarilyEquivOfSameParamStatement

theorem **_root_.GlobalTheory.T.coverDiscreteSeriesUnitarilyEquiv_of_sameParam** : GlobalTheory.Dec.DSUnique → ∀ (p q : ℕ), 1 ≤ p + q → ∀ (T : GrpData p q) (R R' : DSRep T) (lam : Ix p q → ℚ), IsParam lam → HasParam R lam → HasParam R' lam → UnitEquiv R R'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoverFactoringRepresentationUnitaryGroupDictiona

theorem **signCompat_of_blockOrd** {p q n : ℕ} {e : Ix p q ≃ Fin n} {ε : Fin n → Bool} (h : BlockOrd e ε) : SignCompat ε e

theorem **dictionary** : ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (ε : Fin n → Bool) (T : GrpData p q) (U : URep T) (σ : Matrix (Fin n) (Fin n) ℂ → (U.H →L[ℂ] U.H)), BlockOrd e ε → Factors U e σ → IsURepOn (UJset ε) σ ∧ (IrrU U → IsIrredOn (UJset ε) σ) ∧ (∀ x, KFin T U.π x ↔ IsKFin ε σ x) ∧ (∀ Y, Y ∈ upq p q ↔ Formalism.REAL.inG ε (Matrix.reindex e e Y)) ∧ (∀ Y ∈ upq p q, ∀ t : ℝ, U.π (T.expt (t • Y)) = σ (NormedSpace.exp (t • Matrix.reindex e e Y))) ∧ ∀ k ∈ Formalism.REAL.Kset ε, ∃ x ∈ T.Kset, U.π x = σ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CoverFactoringRepresentationUnitaryGroupDictionaStatement

theorem **_root_.GlobalTheory.T.coverFactoringRepresentation_unitaryGroup_dictionary** : ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (ε : Fin n → Bool) (T : GrpData p q) (U : URep T) (σ : Matrix (Fin n) (Fin n) ℂ → (U.H →L[ℂ] U.H)), BlockOrd e ε → Factors U e σ → IsURepOn (UJset ε) σ ∧ (IrrU U → IsIrredOn (UJset ε) σ) ∧ (∀ x, KFin T U.π x ↔ IsKFin ε σ x) ∧ (∀ Y, Y ∈ upq p q ↔ Formalism.REAL.inG ε (Matrix.reindex e e Y)) ∧ (∀ Y ∈ upq p q, ∀ t : ℝ, U.π (T.expt (t • Y)) = σ (NormedSpace.exp (t • Matrix.reindex e e Y))) ∧ ∀ k ∈ Formalism.REAL.Kset ε, ∃ x ∈ T.Kset, U.π x = σ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormMapEqZeroOfTranslatedEvalAeZero

theorem **GlobalTheory.Core.Mi.multT_eq_zero_of_evS_ae** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (v₀ : M.Vec), IsInfFam θ → (∀ τ, (v₀ τ : M.H τ) ≠ 0) → ∀ T ∈ MultT θ M J χ, evS X h t (T (PiTensorProduct.tprod ℂ v₀)) =ᵐ[muX X] 0 → T = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormMapEqZeroOfTranslatedEvalAeZeroStatement

theorem **_root_.GlobalTheory.T.cuspFormMap_eq_zero_of_translatedEval_ae_zero** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (v₀ : M.Vec), IsInfFam θ → (∀ τ, (v₀ τ : M.H τ) ≠ 0) → ∀ T ∈ MultT θ M J χ, evS X h t (T (PiTensorProduct.tprod ℂ v₀)) =ᵐ[muX X] 0 → T = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormMapValuesCompactSobolevBound

theorem **fam_smooth** [CompleteSpace H] {σ : Mx n → (H →L[ℂ] H)} (hσ : IsURepOn (UJset ε) σ) {v : List (GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε)) → H} (hv : v ∈ Fam ε σ) : SmoothVec (lieSet ε) σ (v []) ∧ ∀ l : List (GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε)), Dw σ (l.map Subtype.val) (v []) = v l

theorem **dRep_smul_of_sm** {σ : Mx n → (H →L[ℂ] H)} {w : H} (hw : Sm ε σ w) {X : Mx n} (hX : X ∈ lieSet ε) (a : ℝ) : dRep σ (a • X) w = (a : ℂ) • dRep σ X w

theorem **smooth_fam** [CompleteSpace H] {σ : Mx n → (H →L[ℂ] H)} (hσ : IsURepOn (UJset ε) σ) {w : H} (hw : SmoothVec (lieSet ε) σ w) : (fun l : List (GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε)) => Dw σ (l.map Subtype.val) w) ∈ Fam ε σ

theorem **fam_baire** [CompleteSpace H] {σ : Mx n → (H →L[ℂ] H)} (hσ : IsURepOn (UJset ε) σ) : BaireSpace (Fam ε σ)

theorem **exists_lift** (l : List (Mx n)) (hl : ∀ X ∈ l, X ∈ lieSet ε) : ∃ l' : List (GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε)), l'.map Subtype.val = l

theorem **exists_cont_bound** [CompleteSpace H] {σ : Mx n → (H →L[ℂ] H)} (hσ : IsURepOn (UJset ε) σ) {d : ℕ} {b : Fin d → GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε)} {φ : Fin d → (Mx n →ₗ[ℝ] ℝ)} (hb : ∀ X : GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε), ∑ j, φ j X • b j = X) (S : List (List (Mx n))) (hS : ∀ l ∈ S, ∀ X ∈ l, X ∈ lieSet ε) : ∃ q : Fam ε σ → ℝ, Continuous q ∧ q 0 = 0 ∧ ∀ v : Fam ε σ, (S.map fun l => ‖Dw σ l (v.1 [])‖).sum ≤ q v

theorem **continuous_of_bound** {E : Type} [AddCommGroup E] [Module ℂ E] [TopologicalSpace E] [IsTopologicalAddGroup E] (f : E →ₗ[ℂ] ℂ) (q : E → ℝ) (hq : Continuous q) (hq0 : q 0 = 0) (h : ∀ x, ‖f x‖ ≤ q x) : Continuous f

theorem **sn_eq_sum** {σ : Mx n → (H →L[ℂ] H)} {d : ℕ} (b : Fin d → GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε)) (k : ℕ) : ∃ L : List (List (GlobalTheory.LowTypeDetermined.lie (Formalism.REAL.Jm ε))), ∀ v : Fam ε σ, sn σ b k v = (L.map fun l => ‖v.1 l‖).sum

theorem **unif_model** [CompleteSpace H] {σ : Mx n → (H →L[ℂ] H)} (hσ : IsURepOn (UJset ε) σ) {ι : Type} (ℓ : ι → H →ₗ[ℂ] ℂ) (hℓ : ∀ i, Sobolev₂ (lieSet ε) σ (ℓ i)) (hbd : ∀ w, SmoothVec (lieSet ε) σ w → ∃ B, ∀ i, ‖ℓ i w‖ ≤ B) : ∃ (S : List (List (Mx n))) (C : ℝ), (∀ l ∈ S, ∀ X ∈ l, X ∈ lieSet ε) ∧ ∀ i w, SmoothVec (lieSet ε) σ w → ‖ℓ i w‖ ≤ C * sobNorm σ S w

theorem **dRep_conj_eq** (σ : Mx n → (H →L[ℂ] H)) {P : Mx n} (hP : IsUnit P) (X : Mx n) : dRep (fun g => σ (P * g * P⁻¹)) (P⁻¹ * X * P) = dRep σ X

theorem **foldr_conj_eq** (σ : Mx n → (H →L[ℂ] H)) {P : Mx n} (hP : IsUnit P) (l : List (Mx n)) (w : H) : (l.map fun X => P⁻¹ * X * P).foldr (dRep (fun g => σ (P * g * P⁻¹))) w = l.foldr (dRep σ) w

theorem **GlobalTheory.Core.Mi.sobolev_unif_bound** : ∀ (D : D7) (θ : InfFam D) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ), IsInfFam θ → ∀ T : MMap M, IsMultMap θ M J χ T → ∀ Ω : Set (GA D V), IsCompact Ω → ∀ τ (v : M.Vec), ∃ (L : List (List (Matrix (Fin V.rank) (Fin V.rank) ℂ))) (C : ℝ), (∀ l ∈ L, ∀ Y ∈ l, Y ∈ lieAt V τ) ∧ ∀ w : M.sm τ, ∀ g ∈ Ω, ‖T (Function.update v τ w) g‖ ≤ C * (L.map fun l => ‖l.foldr (dRep (M.rep τ)) (w : M.H τ)‖).sum

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormMapValuesCompactSobolevBoundStatement

theorem **_root_.GlobalTheory.T.cuspFormMap_values_compact_sobolev_bound** : ∀ (D : D7) (θ : InfFam D) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ), IsInfFam θ → ∀ T : MMap M, IsMultMap θ M J χ T → ∀ Ω : Set (GA D V), IsCompact Ω → ∀ τ (v : M.Vec), ∃ (L : List (List (Matrix (Fin V.rank) (Fin V.rank) ℂ))) (C : ℝ), (∀ l ∈ L, ∀ Y ∈ l, Y ∈ lieAt V τ) ∧ ∀ w : M.sm τ, ∀ g ∈ Ω, ‖T (Function.update v τ w) g‖ ≤ C * (L.map fun l => ‖l.foldr (dRep (M.rep τ)) (w : M.H τ)‖).sum

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormsArchimedeanKFiniteRightTranslateFiniteA

theorem **cuspFormsArchimedeanKFinite_rightTranslate_finite** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (cA : CptF A), IsInfFam θ → ∀ k ∈ GfC A, ∀ u ∈ A0 θ A cA, rTr A k u ∈ A0 θ A cA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormsArchimedeanKFiniteRightTranslateFiniteAStatement

theorem **_root_.GlobalTheory.T.rTr_mem_A0** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (cA : CptF A), IsInfFam θ → ∀ k ∈ GfC A, ∀ u ∈ A0 θ A cA, rTr A k u ∈ A0 θ A cA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormsArchimedeanKFiniteStarCompMem

theorem **isKFinite_conj** {G : Type} [Group G] {K : Subgroup G} {Φ : G → ℂ} (h : GlobalTheory.Doubling.IsKFinite K Φ) : GlobalTheory.Doubling.IsKFinite K fun g => conj (Φ g)

theorem **conj_comp_mem_A0** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (cA : CptF A), IsInfFam θ → ∀ u ∈ A0 θ A cA, (fun g => conj (u g)) ∈ A0 θ A cA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormsArchimedeanKFiniteStarCompMemStatement

theorem **_root_.GlobalTheory.T.cuspFormsArchimedeanKFinite_star_comp_mem** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (cA : CptF A), IsInfFam θ → ∀ u ∈ A0 θ A cA, (fun g => conj (u g)) ∈ A0 θ A cA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormsRapidDecay

theorem **GlobalTheory.RapidDecay.cuspRapidDecay** : GlobalTheory.Published.CuspRapidDecay

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspFormsRapidDecayStatement

theorem **_root_.GlobalTheory.T.cusp_forms_rapid_decay** : GlobalTheory.Published.CuspRapidDecay

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspIntegralBasisCovolumeConstant

theorem **normSq_det_UJ** {n : ℕ} {ε : Fin n → Bool} (g : Formalism.REAL.UJ ε) : Complex.normSq (uMat g).det = 1

theorem **card_eq** : Fintype.card (Σ _ : InfinitePlace I.F, Σ _ : Fin I.n, Fin 2) = Fintype.card (Idx I)

theorem **rho_isRealCoord** : IsRealCoord I (rho I)

theorem **bQ_apply** (p : Idx I) : bQ I p = Pi.single p.1 (integralBasis I.E p.2)

theorem **eB_isZBasis** : IsZBasis I (eB I)

theorem **vec_eq** (y : I.Ginf) (u : Fin I.n → I.E) : GlobalTheory.CuspGeo.vec I y u = bm I y (GlobalTheory.CuspGeo.vec I 1 u)

theorem **det_bm** (y : I.Ginf) : LinearMap.det (bm I y) = 1

theorem **gm_eq** {N : ℕ} (ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)) (y : I.Ginf) (e : Fin N → Fin I.n → I.E) : gm I ρ y e = LinearMap.toMatrix' (ρ.toLinearMap ∘ₗ bm I y ∘ₗ ρ.symm.toLinearMap) * gm I ρ 1 e

theorem **det_gm** {N : ℕ} (ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)) (y : I.Ginf) (e : Fin N → Fin I.n → I.E) : (gm I ρ y e).det = (gm I ρ 1 e).det

theorem **qm_injective** : Function.Injective (qm I)

theorem **sw_injective** : Function.Injective (sw I)

theorem **vec_one_bQ** (ρ : Idx I) : GlobalTheory.CuspGeo.vec I 1 (bQ I ρ) = qm I (sw I (Formalism.L2.SG.Pf.Tau.placeCoordsBasis I I.n ρ))

theorem **det_gm_one_ne** : (gm I (rho I) 1 (eB I)).det ≠ 0

theorem **GlobalTheory.CuspGeo.covolume_const** : ∀ I : AutInst, ∃ (N : ℕ) (e : Fin N → Fin I.n → I.E) (ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)) (d : ℝ), 0 < d ∧ IsZBasis I e ∧ IsRealCoord I ρ ∧ ∀ y : I.Ginf, |(gm I ρ y e).det| = d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspIntegralBasisCovolumeConstantStatement

theorem **_root_.GlobalTheory.T.cusp_integralBasis_covolume_constant** : ∀ I : AutInst, ∃ (N : ℕ) (e : Fin N → Fin I.n → I.E) (ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)) (d : ℝ), 0 < d ∧ IsZBasis I e ∧ IsRealCoord I ρ ∧ ∀ y : I.Ginf, |(gm I ρ y e).det| = d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspidalModerateGrowthRapidDecay

theorem **closeR_mono** {D : D7} {θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)} {V : HSp D} {u : GA D V} {a a' : ℝ} (h : CloseR θ V u a) (hle : a ≤ a') : CloseR θ V u a'

theorem **cuspDecayBound** : GlobalTheory.Published.CuspDecayBound

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.CuspidalModerateGrowthRapidDecayStatement

theorem **_root_.GlobalTheory.T.cuspidal_moderateGrowth_rapidDecay** : GlobalTheory.Published.CuspDecayBound

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DatExists

theorem **comb_smul_coef** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (κ : ℂ) (d : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (κ • d) f = κ • comb ψ S T d f

theorem **isRatDatum_const_mul** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {e : AE D} {Ξ : Matrix (Fin T.rank) (Fin S.rank) D.host.E → ℂ} {f₀ f : WO S T → ℂ} (h : IsRatDatum ψ S T e Ξ f₀ f) (κ : ℂ) : IsRatDatum ψ S T e (fun z => κ * Ξ z) f₀ (κ • f)

theorem **datRoot_ne_zero** {x : ℂ} (h : ∃ n : ℕ, 0 < n ∧ x ^ n = 1) : x ≠ 0

theorem **datRoot_mul** {x y : ℂ} (hx : ∃ n : ℕ, 0 < n ∧ x ^ n = 1) (hy : ∃ n : ℕ, 0 < n ∧ y ^ n = 1) : ∃ n : ℕ, 0 < n ∧ (x * y) ^ n = 1

theorem **datRoot_prod** {ι : Type*} (s : Finset ι) (x : ι → ℂ) (h : ∀ i ∈ s, ∃ n : ℕ, 0 < n ∧ x i ^ n = 1) : ∃ n : ℕ, 0 < n ∧ (∏ i ∈ s, x i) ^ n = 1

theorem **QJ.dat_exists** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → a.True → (∀ r s, (a.g.oT B.R.m rfl).iA r s = if r.val = s.val then 1 else 0) → ∃ C : ℕ → ℂ, (∀ j, j < B.R.m → C j ≠ 0) ∧ ∀ t (J : QJ a t), J.Reads → ∃ s ζ ρ, J.Dat C s ζ ρ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DataAsCombinationsAtSplitPlacesProof

theorem **vac_mono** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P Q : Pl D → Prop} (h : ∀ x, Q x → P x) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) : Vac ψ S T LS LT Q f

theorem **combAt_mono** {S T : HSp D} {P Q : Pl D → Prop} (h : ∀ x, P x → Q x) {c : WO S T →₀ ℂ} (hc : CombAt P c) : CombAt Q c

theorem **overAll_nil** {f f' : WO S T → ℂ} : OverAll ψ α β LS LT Ψ [] f f' ↔ f' = f

theorem **overAll_cons** {i : ι} {J : List ι} {f f' : WO S T → ℂ} : OverAll ψ α β LS LT Ψ (i :: J) f f' ↔ ∃ g, OverAll ψ α β LS LT Ψ J f g ∧ Over ψ (α i) (β i) LS LT g (Ψ i) f'

theorem **over_unique** (hψ : IsStdChar D ψ (-1)) {i : ι} {g f' f'' : WO S T → ℂ} (h' : Over ψ (α i) (β i) LS LT g (Ψ i) f') (h'' : Over ψ (α i) (β i) LS LT g (Ψ i) f'') : f' = f''

theorem **overAll_unique** (hψ : IsStdChar D ψ (-1)) (J : List ι) : ∀ f f' f'' : WO S T → ℂ, OverAll ψ α β LS LT Ψ J f f' → OverAll ψ α β LS LT Ψ J f f'' → f' = f''

theorem **over_comb** (hψ : IsStdChar D ψ (-1)) {i : ι} (hu : (p i).UnitAA) {Q : Pl D → Prop} (hQ : ¬ Q (p i).v) {d : WO S T →₀ ℂ} (hd : CombAt Q d) {g f : WO S T → ℂ} (h : Over ψ (α i) (β i) LS LT g (Ψ i) f) : Over ψ (α i) (β i) LS LT (comb ψ S T d g) (Ψ i) (comb ψ S T d f)

theorem **overAll_comb** (hψ : IsStdChar D ψ (-1)) (hu : ∀ i, (p i).UnitAA) {Q : Pl D → Prop} (hQ : ∀ i, ¬ Q (p i).v) {d : WO S T →₀ ℂ} (hd : CombAt Q d) (J : List ι) (f : WO S T → ℂ) : ∀ f', OverAll ψ α β LS LT Ψ J f f' → OverAll ψ α β LS LT Ψ J (comb ψ S T d f) (comb ψ S T d f')

theorem **exists_comb** (hψ : IsStdChar D ψ (-1)) (hu : ∀ i, (p i).UnitAA) (hinj : Function.Injective fun i => (p i).v) (hLS : IsSelfDualC S LS) (hLT : IsSelfDualC T LT) (hΨ : ∀ i, Ψ i ∈ SB (Matrix n n' k)) (J : List ι) (hJ : J.Nodup) : ∃ c : WO S T →₀ ℂ, CombAt (fun x => ∃ i ∈ J, x = (p i).v) c ∧ ∀ f, (∀ i ∈ J, Vac ψ S T LS LT (· = (p i).v) f) → OverAll ψ α β LS LT Ψ J f (comb ψ S T c f)

theorem **allComb** : ∀ D : D7, AllComb D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DataAsCombinationsAtSplitPlacesProofStatement

theorem **_root_.GlobalTheory.T.dataAsCombinationsAtSplitPlaces_proof** : ∀ D : D7, GlobalTheory.FinPl.AllComb D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DataNonvanishingAllMembers

theorem **dataNW_of_val** {C : Formalism.MainNotions.Inhab.FC} {cD ch : ℝ} (hC : Std C) {D : D7} (L : SizeData₂ D) (hV : Val C cD ch L) (hS : SizesAt L) (hR : ∀ u, GlobalTheory.Core.Rk.RankMu D u) : GlobalTheory.Core.Deg.DataNW D

theorem **val_congr** {C C' : Formalism.MainNotions.Inhab.FC} {cD cD' ch ch' : ℝ} {D : D7} {L : SizeData₂ D} (h1 : C = C') (h2 : cD = cD') (h3 : ch = ch') (h : Val C cD ch L) : Val C' cD' ch' L

theorem **val_sizes_at** (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL) : GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun κ _ O _ _ => Std κ.C ∧ Val κ.C κ.cD κ.ch O.L ∧ SizesAt O.L)

theorem **dataNW_at** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => GlobalTheory.Core.Deg.DataNW D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DataNonvanishingAllMembersStatement

theorem **_root_.GlobalTheory.T.data_nonvanishing_all_members** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => GlobalTheory.Core.Deg.DataNW D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DatumListConsEqIdempotentTensor

theorem **datJ_cons** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : ∀ i, Frame (p i) S n') (β : ∀ i, Frame (p i) T n) (Ψ : ι → Matrix n n' k → ℂ) (i : ι) (J : List ι), i ∉ J → (∀ j ∈ J, (p i).e ∗ (p j).e = 0) → GlobalTheory.FinPl.PutIn.datJ p α β Ψ (i :: J) = GlobalTheory.FinPl.PutIn.tens (p i).e (GlobalTheory.FinPl.PutIn.datJ p α β Ψ J) (extFun (α i) (β i) (Ψ i))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DatumListConsEqIdempotentTensorStatement

theorem **_root_.GlobalTheory.T.datum_list_cons_eq_idempotent_tensor** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : ∀ i, Frame (p i) S n') (β : ∀ i, Frame (p i) T n) (Ψ : ι → Matrix n n' k → ℂ) (i : ι) (J : List ι), i ∉ J → (∀ j ∈ J, (p i).e ∗ (p j).e = 0) → GlobalTheory.FinPl.PutIn.datJ p α β Ψ (i :: J) = GlobalTheory.FinPl.PutIn.tens (p i).e (GlobalTheory.FinPl.PutIn.datJ p α β Ψ J) (extFun (α i) (β i) (Ψ i))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DatumReadingSplitPlacesEqClassicalTheta

theorem **reads_datum_eq_thC** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ e (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS ∗ D.host.q ∗ D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → (∃ ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E), IsSelfDualC S ΛS ∧ (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N) → (∃ ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E), IsSelfDualC T ΛT ∧ (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N') → ∀ f₀ Φ f (Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ), IsVacO ψ θ S T (fun _ => 1) (fun _ => 1) (latFr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DatumReadingSplitPlacesEqClassicalThetaStatement

theorem **_root_.GlobalTheory.T.datumReading_splitPlaces_eq_classicalTheta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ e (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS ∗ D.host.q ∗ D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → (∃ ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E), IsSelfDualC S ΛS ∧ (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N) → (∃ ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E), IsSelfDualC T ΛT ∧ (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N') → ∀ f₀ Φ f (Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ), I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DecoratedLatticeOfAdelicMatrixEqRowPoint

theorem **hasRes_red** {P : HeightOneSpectrum (𝒪 E)} {a : E} (ha : IntAt P a) : HasRes P (algebraMap E _ a) (redP P a)

theorem **adAt_pull** {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) (P : Q) (x : Sp E n) (i : Fin n) : adAt F P.1 (GlobalTheory.Mem.Col2.pullV g x i) = algebraMap E _ ((p.fr P).symm x i)

theorem **isPtOf_noFr** {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) : IsPtOf hc g (noFr p)

theorem **decoOf_eq_forgetFr** [NeZero n] {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) : decoOf ST (uSG SG Q) g = forgetFr p

theorem **latOf_eq_toRow** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : GlobalTheory.Mem.Col2.TC B X), I.Ok → ∀ (a : I.ι) (y : GlobalTheory.Mem.Col2.TC.GA B (B.R.N B.R.m - 1)) (p : GlobalTheory.Mem.Col2.XPt (B.primes X) B.R I.sel (I.tup a)), GlobalTheory.Mem.Col2.TC.PtOf y p → GlobalTheory.TopVal.latOf I a y = GlobalTheory.Mem.Col2.toRow I.sel (I.tup a) p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DecoratedLatticeOfAdelicMatrixEqRowPointStatement

theorem **_root_.GlobalTheory.T.decoratedLatticeOfAdelicMatrix_eq_rowPoint** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : GlobalTheory.Mem.Col2.TC B X), I.Ok → ∀ (a : I.ι) (y : GlobalTheory.Mem.Col2.TC.GA B (B.R.N B.R.m - 1)) (p : GlobalTheory.Mem.Col2.XPt (B.primes X) B.R I.sel (I.tup a)), GlobalTheory.Mem.Col2.TC.PtOf y p → GlobalTheory.TopVal.latOf I a y = GlobalTheory.Mem.Col2.toRow I.sel (I.tup a) p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DecoratedPointMoveDiamondOperatorEq

theorem **decoratedPoint_move_diamondOperator_eq** {X : Fields} {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 X.E))} {y : Ys X n} (p : PtY ST SG Q y) (P : ST) (a : (𝒪 X.E ⧸ P.1.asIdeal)ˣ) : p.mv (DecoSet.dia1 P a p.1.1.deco) = ((((DecoSet.dia1 P a p.1.1.deco, p.1.1.1.2), p.1.1.2), p.1.2), p.2)

theorem **xiAt_dia** {X : Fields} {Pr : Primes X} {R : Ranks} (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S : Set ℕ} (mu : Tup X.F X.E X.K S) (cst : LocCst X (rowSG Pr sel mu)) (L : ChIdx X Pr R (QE sel (mu : Chr X.F X.E X.K))) (u : Pr.Word) (i : ℕ) (b : Idx sel mu L i) (b' : Idx sel mu L (i + 1)) (p : PtAt R sel mu cst u i) (p' : PtAt R sel mu cst u (i + 1)) (P : Pr.vT u) (s : (𝒪 X.E ⧸ P.1.asIdeal)ˣ) : xiAt Ch sel mu cst L u i b b' (p.mv (DecoSet.dia1 P s p.1.1.deco)) (p'.mv (DecoSet.dia1 P s p'.1.1.deco)) = xiAt Ch sel mu cst L u i b b' p p'

theorem **th_dia** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Laws → ∀ (u : B.Word) (a : I.ι) (i : ℕ) (b : (I.stp u a i).β) (b' : (I.stp u a (i + 1)).β) (p : (I.stp u a i).D b) (p' : (I.stp u a (i + 1)).D b') (P : (I.atIn u a).ST) (s : (Res P.1)ˣ), I.th u a i b b' (((I.atIn u a).A i).dia b P s p) (((I.atIn u a).A (i + 1)).dia b' P s p') = I.th u a i b b' p p'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteLiftCommLevelPrimeOperators

theorem **combAt_mono** {P P' : Pl D → Prop} (h : ∀ v, P v → P' v) {S T : HSp D} {c : WO S T →₀ ℂ} (hc : CombAt P c) : CombAt P' c

theorem **comb_pureTensorO** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} (hstd : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {c : WO S T →₀ ℂ} (hc : IsFinComb c) (φ : MvPolynomial (FVar D T.rank S.rank) ℂ) : pureTensorO ψ θ S T sS sT BS BT (comb ψ S T c f) φ = comb ψ S T c (pureTensorO ψ θ S T sS sT BS BT f φ)

theorem **vac_pureTensorO** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {x : DefSp S} {y : DefSp T} {f₀ : WO S T → ℂ} (hvac : IsVacO ψ θ S T x.s y.s x.L y.L f₀) (P : Pl D → Prop) (φ : MvPolynomial (FVar D T.rank S.rank) ℂ) : Vac ψ S T x.L y.L P (pureTensorO ψ

θ S T x.s y.s x.B y.B f₀ φ)

theorem **sOp_ram** {ι : Type} {p : ι → SplitPt D k} (hinj : Function.Injective fun i => (p i).v) {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : ∀ i, Frame (p i) S n) (Q : ι → Pair k n) (q : Unif k) (χ₀ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ₀ i) 1) (ε : 𝔸[k]ˣ) (i : ι) {φ : GA D S → ℂ} (hφ : IsContAut S φ ∧ ∀ i', RamAt (β i') (Q i') φ) : IsContAut S (sOp (β i) (Q i) q (resChar (χ₀ i) (hχ i) q) ε φ) ∧ ∀ i', RamAt (β i') (Q i') (sOp (β i) (Q i) q (resChar (χ₀ i) (hχ i) q) ε φ)

theorem **foldr_ram** {ι : Type} {p : ι → SplitPt D k} (hinj : Function.Injective fun i => (p i).v) {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : ∀ i, Frame (p i) S n) (Q : ι → Pair k n) (q : Unif k) (χ₀ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ₀ i) 1) (ε : ι → 𝔸[k]ˣ) (l : List ι) {φ : GA D S → ℂ} (hφ : IsContAut S φ ∧ ∀ i', RamAt (β i') (Q i') φ) : IsContAut S (l.foldr (fun i φ' => sOp (β i) (Q i) q (resChar (χ₀ i) (hχ i) q) (ε i) φ') φ) ∧ ∀ i', RamAt (β i') (Q i') (l.foldr (fun i φ' => sOp (β i) (Q i) q (resChar (χ₀ i) (hχ i) q) (ε i) φ') φ)

theorem **isIdele_idele** (p : SplitPt D k) (x : kˣ) : p.IsIdele x (p.idele x)

theorem **chrPow_mul_neg** (μ : (AE D)ˣ →* Circle) (a : (AE D)ˣ) (e : ℤ) : chrPow μ a e * chrPow μ a (-e) = 1

theorem **pt_over** {D : D7} {c : Comm D} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} {v : VCom c} {o : TDat c k ι} {t : Tup c} {w : VTup t} (hc : c.Is) (ht : t.Is) (hv : v.Is) (hVF : VForm v w) (hSh : ShapeOff v o) (j : ℕ) (hj : j < D.R.m) (α : t.Ch j) (γ : t.Ch (j + 1)) (P : PolF D (c.X (j + 1)).rank (c.X j).rank) (i : ι) : ∃ f₀, Over c.ψ (o.β j i) (o.β (j + 1) i) (c.x j).L (c.x (j + 1)).L f₀ (phiOne (o.Q j i) (o.Q (j + 1) i)) ((t.ker j α γ).pt c.ψ c.θ (c.x j) (c.x (j + 1)) P)

theorem **GlobalTheory.SeeSaw.liftS_of_exact** : ∀ (D : D7) (c : Comm D) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom c) (o : TDat c k ι) (t : Tup c) (w : VTup t), LiftExact D → c.Is → t.Is → v.Is → VForm v w → ShapeOff v o → (∀ j, j < D.R.m → ∃ μ, IsWeilDatumO c.ψ _ _ μ μ (t.kom j) ∧ ∀ i, GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D μ (o.p i).v) → LiftS o t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteLiftCommLevelPrimeOperatorsStatement

theorem **_root_.GlobalTheory.T.definiteLift_comm_levelPrimeOperators** : ∀ (D : D7) (c : Comm D) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom c) (o : TDat c k ι) (t : Tup c) (w : VTup t), LiftExact D → c.Is → t.Is → v.Is → VForm v w → ShapeOff v o → (∀ j, j < D.R.m → ∃ μ, IsWeilDatumO c.ψ _ _ μ μ (t.kom j) ∧ ∀ i, GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D μ (o.p i).v) → LiftS o t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefinitePositivityBothLevelsShortList

theorem **pos_short** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraFinite → GlobalTheory.Published.HarishChandraModules → GlobalTheory.Published.CasselmanWallach → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Hk3.HoweUnram → GlobalTheory.Published.WeilSiegelFormulaUnitary → (∀ D' : Formalism.BasicNotions.D7, GlobalTheory.Core.Rk.SW.HeckeFormNS D') → GlobalTheory.Published.SiegelEisensteinContinuation → GlobalTheory.Published.UnramifiedDoublingGL → GlobalTheory.Published.GreenCuspidal → ∀ eps : ℝ, 0 < eps → ∃ x₀ : ℝ, ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 I0 T : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat3 D A3 Ks) (F : Face3 D A3) {s : ℕ}, ∀ (L : SizeData₂ D) (_ : ParameterChoice' D L), L.sc.eps = eps → x₀ ≤ L.sc.x → Formalism.MainNotions.SizesAt L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefinitePositivityBothLevelsShortListStatement

theorem **_root_.GlobalTheory.T.definite_positivity_both_levels_short_list** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraFinite → GlobalTheory.Published.HarishChandraModules → GlobalTheory.Published.CasselmanWallach → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Hk3.HoweUnram → GlobalTheory.Published.WeilSiegelFormulaUnitary → (∀ D' : Formalism.BasicNotions.D7, GlobalTheory.Core.Rk.SW.HeckeFormNS D') → GlobalTheory.Published.SiegelEisensteinContinuation → GlobalTheory.Published.UnramifiedDoublingGL → GlobalTheory.Published.GreenCuspidal → ∀ eps : ℝ, 0 < eps → ∃ x₀ : ℝ, ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 I0 T : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat3 D A3 Ks) (F : Face3 D A3) {s : ℕ}, ∀ (L : SizeData₂ D) (_ : ParameterChoice' D L), L.sc.ep

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteRealGroupFixesSplitSpaceWeilVacuum

theorem **trace_cartan_of_sig** {k t : ℕ} (G s : Matrix (Fin k) (Fin k) ℂ) (hs : IsCartanInv G s) (hG : Formalism.UG.HasSignature G t t) (hk : k = 2 * t) : s.trace = 0

theorem **vacuum_fixed** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (t : ℕ) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) 1) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → HSpace.SigOK H → H.rank = 2 * t → (∀ τ, H.sig τ = (t, t)) → ∀ f, IsThetaFunO ψ A H f → (∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO 1 (sH τ)) f) → ∀ b ∈ GinfC A, om (b, 1) f = f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteRealGroupFixesSplitSpaceWeilVacuumStatement

theorem **_root_.GlobalTheory.T.definite_real_group_fixes_split_space_weil_vacuum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (t : ℕ) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) 1) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → HSpace.SigOK H → H.rank = 2 * t → (∀ τ, H.sig τ = (t, t)) → ∀ f, IsThetaFunO ψ A H f → (∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO 1 (sH τ)) f) → ∀ b ∈ GinfC A, om (b, 1) f = f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteScalarComponentIsCuspAndCongruenceFixedA

theorem **extV_mul_inf** (R : RealDat D A Ks) (i : Fin D.R.m) (t : T) (y : (R.HV i t).H) {k : GA D (Vc D i)} (hk : k ∈ GinfC (Vc D i)) (h : GA D (Vc D i)) : R.extV i t y (h * k) = (R.cV i).ρ k⁻¹ (R.extV i t y h)

theorem **extV_mul_Kg** (R : RealDat D A Ks) (hM : R.IsModels) (i : Fin D.R.m) (t : T) (y : (R.HV i t).H) {κ : GA D (Vc D i)} (hκ : κ ∈ GfC (Vc D i)) (hK : finE (Vc D i) κ ∈ Kg D.sp.cm (Vc D i).gram (R.βV i) (R.Pg ∪ R.Q t)) (h : GA D (Vc D i)) : R.extV i t y (h * κ) = R.extV i t y h

theorem **mul_inf_mem_range** (R : RealDat D A Ks) (i : Fin D.R.m) (t : T) (y : (R.HV i t).H) (l : (R.cV i).W →ₗ[ℂ] ℂ) {k : GA D (Vc D i)} (hk : k ∈ GinfC (Vc D i)) : (fun g => l (R.extV i t y (g * k))) ∈ LinearMap.range (compL R i t y)

theorem **definiteScalarComponent_isCusp_and_congruence_fixed** : ∀ (D : D7) (G : Type) [Group G] [Fintype G] (Pl0 T : Type) (A : FamIn D.what₂.LL G Pl0 T D.R.m) (Ks : Type) [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), GlobalTheory.LiftRows.Given R F → R.FullP → ∀ i t cA a, Defin R.θ (Vc D i) cA a → ∀ (y : (R.HV i t).H) (l : (R.cV i).W →ₗ[ℂ] ℂ), (fun g

=> l (R.extV i t y g)) ∈ GlobalTheory.Cusp.A0 R.θ (Vc D i) cA ∧ ∀ κ ∈ prin (Vc D i) (R.βV i) (R.Pg ∪ R.Q t), ∀ g, l (R.extV i t y (g ∗ κ)) = l (R.extV i t y g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteScalarComponentIsCuspAndCongruenceFixedAStatement

theorem **_root_.GlobalTheory.T.linearForm_extV_mem_A0_and_prin_invariant** : ∀ (D : D7) (G : Type) [Group G] [Fintype G] (Pl0 T : Type) (A : FamIn D.what₂.LL G Pl0 T D.R.m) (Ks : Type) [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), GlobalTheory.LiftRows.Given R F → R.FullP → ∀ i t cA a, Defin R.θ (Vc D i) cA a → ∀ (y : (R.HV i t).H) (l : (R.cV i).W →ₗ[ℂ] ℂ), (fun g => l (R.extV i t y g)) ∈ GlobalTheory.Cusp.A0 R.θ (Vc D i) cA ∧ ∀ κ ∈ prin (Vc D i) (R.βV i) (R.Pg ∪ R.Q t), ∀ g, l (R.extV i t y (g ∗ κ)) = l (R.extV i t y g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteStepSmallVectorFockVacuumSpherical

theorem **smallVec_vac_sph** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sA : RealMat A) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → A.rank ≤ H.rank → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∀ (βH : GL (Fin H.rank) (Formalism.IndefR.Lev.EAf D)) (f₀ f : WO A H → ℂ), Formalism.L2.DIV.IsFrame βH (frameLat H) → IsStepVecOf ψ θ A H sA sH (frameLat A) (frameLat H) β βH (prinDat A Z) (fun _ => 1) ∅ (fun _ => 1) f₀ f → IsThetaFunO ψ A H f ∧ f ≠ 0 ∧ (∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO (sA τ) (sH τ)) f) ∧ IsSphOff ψ A H (frameLat A) (frameLat H) (under Z) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteStepSmallVectorFockVacuumSphericalStatement

theorem **_root_.GlobalTheory.T.definiteStep_smallVector_fockVacuum_spherical** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sA : RealMat A) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → A.rank ≤ H.rank → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∀ (βH : GL (Fin H.rank) (Formalism.IndefR.Lev.EAf D)) (f₀ f : WO A H → ℂ), Formalism.L2.DIV.IsFrame βH (frameLat H) → IsStepVecOf ψ θ A H sA sH (frameLat A) (frameLat H) β βH (prinDat A Z) (fun _ => 1) ∅ (fun _ => 1) f₀ f → IsThetaFunO ψ A H f ∧ f ≠ 0 ∧ (∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO (sA τ) (sH τ)) f) ∧ IsSphOff ψ A H (frameLat A) (frameLat H) (under Z) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteUnitaryQuotientCompactFundamentalDomain

theorem **exists_measurable_fundamentalDomain_subset** [SecondCountableTopology G] (Γ : Subgroup G) [DiscreteTopology Γ] {K : Set G} (hK : IsClosed K) (hcov : ∀ x : G, ∃ γ ∈ Γ, γ ∗ x ∈ K) (μ : Measure G) : ∃ S ⊆ K, MeasurableSet S ∧ IsFundamentalDomain Γ S μ

theorem **GlobalTheory.DefFund.coCpt_and_fundamentalDomain_subset_compact** : ∀ (D : D7) (A : HSp D), HSpace.SigOK A → HSpace.IsPosDef A → ∀ m : Measure (GA D A), m.IsHaarMeasure → GlobalTheory.Unf.CoCpt A ∧ ∃ (X K : Set (GA D A)), MeasurableSet X ∧ IsFundamentalDomain (GQ A) X m ∧ IsCompact K ∧ X ⊆ K

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DefiniteUnitaryQuotientCompactFundamentalDomainStatement

theorem **_root_.GlobalTheory.T.definite_unitary_quotient_compact_fundamentalDomain** : ∀ (D : D7) (A : HSp D), HSpace.SigOK A → HSpace.IsPosDef A → ∀ m : Measure (GA D A), m.IsHaarMeasure → GlobalTheory.Unf.CoCpt A ∧ ∃ (X K : Set (GA D A)), MeasurableSet X ∧ IsFundamentalDomain (GQ A) X m ∧ IsCompact K ∧ X ⊆ K

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DegenerateCollisionRowBoundAtParametersStatement

theorem **_root_.GlobalTheory.T.degenerateCollision_rowBound_at_sizeData** : ∀ CT cH cV Ca C2 Ch c C : ℝ, 1 ≤ c → CT ≤ c → |cH| + 2 ≤ c → |cV| + 2 ≤ c → 0 ≤ Ch → c ≤ C → 4 + (c ^ 2 + 2 ∗ max (Ch − c) 0) ∗ (6 ∗ c + 2) ≤ C → 2 ∗ C2 ≤ C → |Ca| ≤ C → ∀ (D : D7) (large₂ : SizeData₂ D) (u : D.what₂.Word) (T : TermTie D) (Z : LoopTie D large₂ T u u 1) (C' Cl Cg cb Cu cT C0 : ℝ), 3 ∗ |C2| ≤ C' → Cnt.Hyp D large₂ u u C' Cl Cg cb Cu cT CT C0 1 → UnitaryVolumeMass.MeanFlagPairCount D large₂ u u C' Cl Cg cb Cu cT CT C0 1 3 c → UnitaryVolumeMass.stHH D large₂ u u C' Cl Cg cb Cu cT CT C0 1 cH → UnitaryVolumeMass.stVol D large₂ T u u 1 Z C' Cl Cg cb Cu cT CT C0 1 cV → AmpAt D large₂ C' Cl Cg cb Cu cT CT C0 (9 ∗ c + 3) Ca C2 Ch 1 → SmallAt D large₂ C' Cl Cg cb Cu cT CT C0 1 → BridgeLemma.TermPtsTie T Z → @BridgeLemma.LatticeDeterminantFactorization D T u u → BridgeLemma₂.wordPairEntriesInLattice T u u → Bri

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DegenerateCollisionRowBoundAtParams

theorem **rowsDeg_at_record** : ∀ CT cH cV Ca C2 Ch c C : ℝ, 1 ≤ c → CT ≤ c → |cH| + 2 ≤ c → |cV| + 2 ≤ c → 0 ≤ Ch → c ≤ C → 4 + (c ^ 2 + 2 ∗ max (Ch − c) 0) ∗ (6 ∗ c + 2) ≤ C → 2 ∗ C2 ≤ C → |Ca| ≤ C → ∀ (D : D7) (large₂ : SizeData₂ D) (u : D.what₂.Word) (T : TermTie D) (Z : LoopTie D large₂ T u u 1) (C' Cl Cg cb Cu cT C0 : ℝ), 3 ∗ |C2| ≤ C' → Cnt.Hyp D large₂ u u C' Cl Cg cb Cu cT CT C0 1 → UnitaryVolumeMass.MeanFlagPairCount D large₂ u u C' Cl Cg cb Cu cT CT C0 1 3 c → UnitaryVolumeMass.stHH D large₂ u u C' Cl Cg cb Cu cT CT C0 1 cH → UnitaryVolumeMass.stVol D large₂ T u u 1 Z C' Cl Cg cb Cu cT CT C0 1 cV → AmpAt D large₂ C' Cl Cg cb Cu cT CT C0 (9 ∗ c + 3) Ca C2 Ch 1 → SmallAt D large₂ C' Cl Cg cb Cu cT CT C0 1 → BridgeLemma.TermPtsTie T Z → @BridgeLemma.LatticeDeterminantFactorization D T u u → BridgeLemma₂.wordPairEntriesInLattice T u u → BridgeLemma.RowsDeg T Z C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DegenerateTermsAndCollisionBoundsUniformWidths

theorem **degenerate_and_collisions_uncond** : (∀ c : ℝ, On {c} VanishingGramTermsBound) ∧ ∀ c : ℝ, On {c} CollS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DegenerateTermsAndCollisionBoundsUniformWidthsStatement

theorem **_root_.GlobalTheory.T.degenerateTerms_and_collisionBounds_uniform_widths** : (∀ c : ℝ, On {c} VanishingGramTermsBound) ∧ ∀ c : ℝ, On {c} CollS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DegreeHypothesisAllMembers

theorem **del_lo** {x eps k f CD cD : ℝ} (hx : 1 ≤ x) (he : eps ≤ 1) (hk0 : 0 ≤ f) (hfk : f ≤ k) (hk : k ≤ x ^ (5 : ℝ)) (hCD : 0 ≤ CD) (hc : 11 ∗ CD ≤ cD) : (k ∗ f ∗ x ^ eps) ^ CD ≤ ((⌈x ^ cD⌉₊ : ℕ) : ℝ)

theorem **evP_hypL** (A B : Formalism.MainNotions.Inhab.FC → ℝ) : EvP fun κ K => ∀ β : BPt K, (∀ u : β.D.what₂.Word, β.O.L.sc.b ^ ((1 / 2 : ℝ) ∗ (β.D.R.m : ℝ)) ≤ Formalism.MainNotions.S.Tr β.D u ∧ Formalism.MainNotions.S.Tr β.D u ≤ β.O.L.sc.b ^ (κ.C.CTu ∗ (β.D.R.m : ℝ))) → GlobalTheory.Core.Deg.HypL β.D β.O.L.sc.L (max (A κ.C) (8 ∗ κ.C.CGam)) (max (B κ.C) 40) (κ.C.P.CYL + 1) (1 / 4) 3 (1 / 2) κ.C.CTu

theorem **hypL_at_points** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL) (A B : Formalism.MainNotions.Inhab.FC → ℝ), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun κ D O _ _ => GlobalTheory.Core.Deg.HypL D O.L.sc.L (max (A κ.C) (8 ∗ κ.C.CGam)) (max (B κ.C) 40) (κ.C.P.CYL + 1) (1 / 4) 3 (1 / 2) κ.C.CTu)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DegreeHypothesisAllMembersStatement

theorem **_root_.GlobalTheory.T.degree_hypothesis_all_members** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL) (A B : Formalism.MainNotions.Inhab.FC → ℝ), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked ρ) (fun κ D O _ _ => GlobalTheory.Core.Deg.HypL D O.L.sc.L (max (A κ.C) (8 * κ.C.CGam)) (max (B κ.C) 40) (κ.C.P.CYL + 1) (1 / 4) 3 (1 / 2) κ.C.CTu)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DenseKFiniteVectorsUnitaryGroupRep

theorem **dense_finite_vectors** (K : Type) [Group K] [TopologicalSpace K] [IsTopologicalGroup K] [CompactSpace K] [T2Space K] [SecondCountableTopology K] (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (π : K →* H →L[ℂ] H) (hiso : ∀ k x, ‖π k x‖ = ‖x‖) (hcont : ∀ x, Continuous fun k => π k x) : Dense {v : H | ∃ S : Submodule ℂ H, FiniteDimensional ℂ S ∧ v ∈ S ∧ ∀ k, ∀ x ∈ S, π k x ∈ S}

theorem **dense_isKFin** (n : ℕ) (ε : Fin n → Bool) (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)) (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) : Dense {v : H | Formalism.MainNotions.S.IsKFin ε σ v}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DenseKFiniteVectorsUnitaryGroupRepStatement

theorem **_root_.GlobalTheory.T.dense_kFinite_vectors_unitaryGroupRep** : ∀ (n : ℕ) (ε : Fin n → Bool) (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)), Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ → Dense {v : H | Formalism.MainNotions.S.IsKFin ε σ v}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DependentFrameFirstColumnRelation

theorem **mulVec_apply_eq_zero_of_upper** {u : Matrix (Fin ρ) (Fin ρ) E} (hu : ∀ i k, k < i → u i k = 0) {a : Fin ρ → E} {i : Fin ρ} (ha : ∀ k, i ≤ k → a k = 0) : (u *ᵥ a) i = 0

theorem **mulVec_apply_eq_of_upper** {u : Matrix (Fin ρ) (Fin ρ) E} (hu : ∀ i k, k < i → u i k = 0) (hd : ∀ i, u i i = 1) {a : Fin ρ → E} {j : Fin ρ} (ha : ∀ i, j < i → a i = 0) : (u *ᵥ a) j = a j

theorem **sum_smul_transpose** (x : Matrix (Fin n) (Fin ρ) E) (a : Fin ρ → E) : ∑ i, a i • x.transpose i = x *ᵥ a

theorem **not_linearIndependent_iff_exists** (x : Matrix (Fin n) (Fin ρ) E) : ¬ LinearIndependent E x.transpose ↔ ∃ j, FirstRel x j

theorem **firstRel_unique** (x : Matrix (Fin n) (Fin ρ) E) {j j' : Fin ρ} (h : FirstRel x j) (h' : FirstRel x j') : j = j'

theorem **firstRel_mul** (x : Matrix (Fin n) (Fin ρ) E) (j : Fin ρ) {u v : Matrix (Fin ρ) (Fin ρ) E} (hu : ∀ i k, k < i → u i k = 0) (hv : ∀ i k, k < i → v i k = 0) (hvd : ∀ i, v i i = 1) (huv : u * v = 1) (hvu : v * u = 1) (h : FirstRel x j) : FirstRel (x * u) j

theorem **firstRel_mul_iff** (x : Matrix (Fin n) (Fin ρ) E) (j : Fin ρ) (u : Matrix (Fin ρ) (Fin ρ) E) (hd : ∀ i, u i i = 1) (hu : ∀ i k, k < i → u i k = 0) : FirstRel x j ↔ FirstRel (x * u) j

theorem **dependent_frame_first_relation** : ∀ (E : Type) [Field E] (n ρ : ℕ) (x : Matrix (Fin n) (Fin ρ) E), (¬ LinearIndependent E x.transpose ↔ ∃ j : Fin ρ, (∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0) ∧ (∀ j j' : Fin ρ, ((∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0) → ((∀ a : Fin ρ → E, (∀ i, j' ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j' = 1 ∧ (∀ i, j' < i → a i = 0) ∧ x *ᵥ a = 0) → j = j') ∧ ∀ (j : Fin ρ) (u : Matrix (Fin ρ) (Fin ρ) E), (∀ i, u i i = 1) → (∀ i k, k < i → u i k = 0) → (((∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0) ↔ ((∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → (x * u) *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DependentFrameFirstColumnRelationStatement

theorem **_root_.GlobalTheory.T.dependent_frame_first_column_relation** : ∀ (E : Type) [Field E] (n ρ : ℕ) (x : Matrix (Fin n) (Fin ρ) E), (¬ LinearIndependent E x.transpose ↔ ∃ j : Fin ρ, (∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0) ∧ (∀ j j' : Fin ρ, ((∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0) → ((∀ a : Fin ρ → E, (∀ i, j' ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j' = 1 ∧ (∀ i, j' < i → a i = 0) ∧ x *ᵥ a = 0) → j = j') ∧ ∀ (j : Fin ρ) (u : Matrix (Fin ρ) (Fin ρ) E), (∀ i, u i i = 1) → (∀ i k, k < i → u i k = 0) → (((∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0) ↔ ((∀ a : Fin ρ → E, (∀ i, j ≤ i → a i = 0) → (x * u) *ᵥ a = 0 →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DependentFramesFirstRelationMeanZero

theorem **gramQ_mul** {D : D7} (W : HSp D) {ρ : ℕ} (x : Matrix (Fin W.rank) (Fin ρ) D.host.E) (u : Matrix (Fin ρ) (Fin ρ) D.host.E) : gramQ W (x * u) = uᴴ * gramQ W x * u

theorem **dep_frames_mean_zero** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (χV χW : (AE D)ˣ →* Circle) (om : Om V W) (om' : Om V₀ W), IsWeilDatumO ψ V W χV χW om → IsWeilDatumO ψ V₀ W χV χW om' → ∀ (vS : VolM D W.rank ρ) (vN : VolN d) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (Th : GA D V → ℂ), IsBase W x₀ → IsTh ψ d W x₀ Th → ∀ Ψ : WO V W → ℂ, IsThetaFunO ψ V W Ψ → ∀ j : Fin ρ, ∫ n in vN.𝒩, ∑' x : {x : Matrix (Fin W.rank) (Fin ρ) D.host.E // gramQ W x = gramQ W x₀ ∧ (∀ a : Fin ρ → D.host.E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → D.host.E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0}, Th n.1 * term ψ d W vS (om (n.1, 1) Ψ) x.1 ∂vN.ν = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DependentFramesFirstRelationMeanZeroStatement

theorem **_root_.GlobalTheory.T.dependentFrames_firstRelation_mean_zero** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (χV χW : (AE D)ˣ →* Circle) (om : Om V W) (om' : Om V₀ W), IsWeilDatumO ψ V W χV χW om → IsWeilDatumO ψ V₀ W χV χW om' → ∀ (vS : VolM D W.rank ρ) (vN : VolN d) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (Th : GA D V → ℂ), IsBase W x₀ → IsTh ψ d W x₀ Th → ∀ Ψ : WO V W → ℂ, IsThetaFunO ψ V W Ψ → ∀ j : Fin ρ, ∫ n in vN.𝒩, ∑' x : {x : Matrix (Fin W.rank) (Fin ρ) D.host.E // gramQ W x = gramQ W x₀ ∧ (∀ a : Fin ρ → D.host.E, (∀ i, j ≤ i → a i = 0) → x *ᵥ a = 0 → a = 0) ∧ ∃ a : Fin ρ → D.host.E, a j = 1 ∧ (∀ i, j < i → a i = 0) ∧ x *ᵥ a = 0}, Th n.1 * term ψ d W vS (om (n.1, 1) Ψ) x.1 ∂vN.ν = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DerivativeClosedFunctionClassSmoothAndSchwartz

theorem **smooth_schwartz** : ∀ (k : ℕ) (V : Set ((Fin k → ℝ) → ℂ)), (∀ φ ∈ V, Continuous φ) → (∀ φ ∈ V, ∀ i : Fin k, ∃ φ' ∈ V, ∀ x : Fin k → ℝ, HasDerivAt (fun t : ℝ => φ (x + t • (Pi.single i (1 : ℝ) : Fin k → ℝ))) (φ' x) 0) → (∀ φ ∈ V, ContDiff ℝ ∞ φ) ∧ ((∀ φ ∈ V, ∀ N : ℕ, ∃ C : ℝ, ∀ x, ‖x‖ ^ N * ‖φ x‖ ≤ C) → ∀ φ ∈ V, ∃ g : SchwartzMap (Fin k → ℝ) ℂ, ∀ x, g x = φ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DerivativeClosedFunctionClassSmoothAndSchwartzStatement

theorem **_root_.GlobalTheory.T.derivativeClosedFunctionClass_smooth_and_schwartz** : ∀ (k : ℕ) (V : Set ((Fin k → ℝ) → ℂ)), (∀ φ ∈ V, Continuous φ) → (∀ φ ∈ V, ∀ i : Fin k, ∃ φ' ∈ V, ∀ x : Fin k → ℝ, HasDerivAt (fun t : ℝ => φ (x + t • (Pi.single i (1 : ℝ) : Fin k → ℝ))) (φ' x) 0) → (∀ φ ∈ V, ContDiff ℝ ∞ φ) ∧ ((∀ φ ∈ V, ∀ N : ℕ, ∃ C : ℝ, ∀ x, ‖x‖ ^ N * ‖φ x‖ ≤ C) → ∀ φ ∈ V, ∃ g : SchwartzMap (Fin k → ℝ) ℂ, ∀ x, g x = φ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataChainStep

theorem **vac_true_mono** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : GlobalTheory.FinPl.Lat S} {LT : GlobalTheory.FinPl.Lat T} (P : GlobalTheory.FinPl.Pl D → Prop) {F : WO S T → ℂ} (h : GlobalTheory.FinPl.Vac ψ S T LS LT (fun _ => True) F) : GlobalTheory.FinPl.Vac ψ S T LS LT P F

theorem **moved_datum_eq** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (o' : WO (Yc D j) (Yc D (j + 1)) →₀ ℂ), IsSelfDualC (Yc D j) M ∧ IsSelfDualC (Yc D (j + 1)) M' ∧ GlobalTheory.FinPl.CombAt (· ∈ v.gP) o' ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M M' (· ∈ v.gP) ((pk W).ωY W.P.A3.A.μ0 j (f.g.b j, f.g.b (j + 1)) ((pk W).vac j)) ∧ G

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataChainStepStatement

theorem **_root_.GlobalTheory.T.descent_data_chain_step** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (o' : WO (Yc D j) (Yc D (j + 1)) →₀ ℂ), IsSelfDualC (Yc D j) M ∧ IsSelfDualC (Yc D (j + 1)) M' ∧ GlobalTheory.FinPl.CombAt (· ∈ v.gP) o' ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M M' (· ∈ v.gP) ((pk W).ωY W.P.A3.A.μ0 j (f.g.b j, f.g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepExponents

theorem **mt_mul** {D : D7} (V : HSp D) (a b : GA D V) : mt V (a * b) = mt V a * mt V b

theorem **mt_one** {D : D7} (V : HSp D) : mt V 1 = 1

theorem **mt_inv_mul** {D : D7} (V : HSp D) (a : GA D V) : mt V a⁻¹ * mt V a = 1

theorem **isHalf_mul** {D : D7} {S T : HSp D} (e : AE D) (A : Matrix (Fin T.rank) (Fin T.rank) (AE D)) (B : Matrix (Fin S.rank) (Fin S.rank) (AE D)) (y : WO S T) (hy : IsHalf e y) : IsHalf e (HMul.hMul (HMul.hMul A (y : Matrix (Fin T.rank) (Fin S.rank) (AE D))) B : WO S T)

theorem **isHalf_actO_iff** {D : D7} (S T : HSp D) (g : GA D S) (h : GA D T) (e : AE D) (x : WO S T) : IsHalf e (actO S T g h x) ↔ IsHalf e x

theorem **down_moved_actO** (p : Place D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (g : GA D S) (h : GA D T) (x : WO S T) : p.down (moved p FS g⁻¹) (moved p FT h⁻¹) (actO S T g⁻¹ h⁻¹ x) = p.down FS FT x

theorem **phi_down_eq** (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) {u : D.what₂.Word} (f : FamCh W u) (v : VCom f.c) (G0 : GLoc v) (π' : Formalism.IndefR.Datum.EAf D →+* G0.k) (hπ : ∀ a, π' (Formalism.IndefR.Datum.toEAf D a) = G0.q.pr a) (j a : ℕ) (ha : j + a = G0.s) (ea : Ix (Fin D.R.k ⊕ Fin (G0.s + 1)) (Fin D.R.t) j ≃ Fin (Yc D j).rank) (eb : Ix (Fin D.R.k ⊕ Fin (G0.s + 1)) (Fin D.R.t) (j + 1) ≃ Fin (Yc D (j + 1)).rank) (mS : MovedTo (ofSplit G0.q) (Yc D j) (f.g.b j) ((pk W).βY j) (((G0.FY j (a + 1) (by omega)).β * ((G0.S.T (Fin D.R.t)).bas j (a + 1) (by omega)).val).submatrix id fun i => stdOrd (Fin D.R.k) (Fin D.R.t) j (a + 1) (by omega) (ea.symm i))) (mT : MovedTo (ofSplit G0.q) (Yc D (j +

theorem **moved_datum_eq_prod** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ (a : Fin n → ℕ) (ha : ∀ i, j + a i = (G i).s), ∃ κ : ℂ, κ ≠ 0 ∧ ∀ x : WO (Yc D j) (Yc D (j + 1)), (if IsHalf (∑ i, (G i).q.e) (actO _ _ (f.g.b j)⁻¹ (f.g.b (j + 1))⁻¹ x) ∧ IsIntM (frMapO (Yc D j) (Yc D (j + 1)) ((pk W).βY j) ((pk W).βY (j + 1)) (actO _ _ (f.g.b j)⁻¹ (f.g.b (j + 1))⁻¹ x)) then omegaG (datY _ D.R.r D.R.t j (r_le_Y j) (pk W).Pg ∅ (lvP W) (pk W).chi (fun _ => 1) Frm.B) ((pk W).chi (j + 1)) (fun PP => redM PP (frMap

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepExponentsStatement

theorem **_root_.GlobalTheory.T.descent_data_step_exponents** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ (a : Fin n → ℕ) (ha : ∀ i, j + a i = (G i).s), ∃ κ : ℂ, κ ≠ 0 ∧ ∀ x : WO (Yc D j) (Yc D (j + 1)), (if IsHalf (∑ i, (G i).q.e) (actO _ _ (f.g.b j)⁻¹ (f.g.b (j + 1))⁻¹ x) ∧ IsIntM (frMapO (Yc D j) (Yc D (j + 1)) ((pk W).βY j) ((pk W).βY (j + 1)) (actO _ _ (f.g.b j)⁻¹ (f.g.b (j + 1))⁻¹ x)) then omegaG (datY _ D.R.r D.R.t j (r_le_Y j) (pk W).Pg ∅ (lvP W) (pk W).chi (fun _ => 1) Frm.B) ((pk W).chi (j +

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepLattice

theorem **mt_mul'** (V : HSp D) (g h : GA D V) : mt V (g * h) = mt V g * mt V h

theorem **mt_one'** (V : HSp D) : mt V 1 = 1

theorem **mt_mul_inv_mulVec** (V : HSp D) (g : GA D V) (y : Fin V.rank → AE D) : mt V g *ᵥ (mt V g⁻¹ *ᵥ y) = y

theorem **mulVec_mem_idAt** {n : ℕ} (A : Matrix (Fin n) (Fin n) (AE D)) {P : GlobalTheory.FinPl.Pl D → Prop} {y : Fin n → AE D} (hy : ∀ i, y i ∈ GlobalTheory.FinPl.idAt D P) (i : Fin n) : (A *ᵥ y) i ∈ GlobalTheory.FinPl.idAt D P

theorem **moves_inv** {S : HSp D} {g : GA D S} {L M : GlobalTheory.FinPl.Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} (h : GlobalTheory.FinPl.Moves S g L M P) : GlobalTheory.FinPl.Moves S g⁻¹ M L P

theorem **smul_mem_latA** {S : HSp D} {L : GlobalTheory.FinPl.Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {y : Fin S.rank → AE D} (hy : y ∈ GlobalTheory.FinPl.latV S L P) : ((1 : AF D.host.F) ⊗ₜ[D.host.F] a) • y ∈

GlobalTheory.FinPl.latA S L P

theorem **mulVec_mem_latA** {S : HSp D} {g : GA D S} {L M : GlobalTheory.FinPl.Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} (h : GlobalTheory.FinPl.Moves S g L M P) {z : Fin S.rank → AE D} (hz : z ∈ GlobalTheory.FinPl.latA S L P) : mt S g *ᵥ z ∈ GlobalTheory.FinPl.latA S M P

theorem **actO_mem_latO** {S T : HSp D} {Λ M : GlobalTheory.FinPl.Lat S} {Λ' M' : GlobalTheory.FinPl.Lat T} (hΛ : IsSelfDualC S Λ) (hM : IsSelfDualC S M) (hΛ' : IsSelfDualC T Λ') (hM' : IsSelfDualC T M') {P : GlobalTheory.FinPl.Pl D → Prop} {g : GA D S} {h : GA D T} (hg : GlobalTheory.FinPl.Moves S g Λ M P) (hh : GlobalTheory.FinPl.Moves T h Λ' M' P) {u : WO S T} (hu : u ∈ latO S T (Λ : Set _) (Λ' : Set _) P) : actO S T g h u ∈ latO S T (M : Set _) (M' : Set _) P

theorem **actO_smul_left** {S T : HSp D} (g : GA D S) (h : GA D T) (e : AE D) (x : WO S T) : actO S T g h (e • x) = e • actO S T g h x

theorem **actO_two_nsmul** {S T : HSp D} (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g h (2 • x) = 2 • actO S T g h x

theorem **isHalf_actO** {S T : HSp D} (g : GA D S) (h : GA D T) {e : AE D} {x : WO S T} (hx : IsHalf e x) : IsHalf e (actO S T g h x)

theorem **actO_actO_inv** {S T : HSp D} (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g h (actO S T g⁻¹ h⁻¹ x) = x

theorem **isLatVac_move** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hW : IsWeilDatumO ψ S T χS χT om) {Λ M : GlobalTheory.FinPl.Lat S} {Λ' M' : GlobalTheory.FinPl.Lat T} (hΛ : IsSelfDualC S Λ) (hM : IsSelfDualC S M) (hΛ' : IsSelfDualC T Λ') (hM' : IsSelfDualC T M') {b : GA D S} {b' : GA D T} (hb : GlobalTheory.FinPl.Moves S b Λ M fun _ => True) (hb' : GlobalTheory.FinPl.Moves T b' Λ' M' fun _ => True) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hv : IsLatVac (heisO ψ S T) (latO S T (Λ : Set _) (Λ' : Set _) fun _ => True) f) : IsLatVac (heisO ψ S T) (latO S T (M : Set _) (M' : Set _) fun _ => True) (om (b, b') f)

theorem **main** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))), IsSelfDualC (Yc D j) M ∧ IsSelfDualC (Yc D (j + 1)) M' ∧ GlobalTheory.FinPl.Moves (Yc D j) (f.g.b j) ((pk W).ΛY j) M (fun _ => True) ∧ GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M' (fun _ => True) ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepLatticeStatement

theorem **_root_.GlobalTheory.T.descent_data_step_lattice** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))), IsSelfDualC (Yc D j) M ∧ IsSelfDualC (Yc D (j + 1)) M' ∧ GlobalTheory.FinPl.Moves (Yc D j) (f.g.b j) ((pk W).ΛY j) M (fun _ => True) ∧ GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M' (fun _ =>

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepSelfDual

theorem **mem_latV_at_iff_all** (S : HSp D) (L : Lat S) (p : Pl D) {z : Fin S.rank → AE D} (hz : ∀ i, z i ∈ idAt D (· = p)) : z ∈ latV S L (· = p) ↔ z ∈ latV S L (fun _ => True)

theorem **latV_eq_of_moves** (S : HSp D) (g : GA D S) (Λ N M : Lat S) (p : Pl D) (hN : Moves S g Λ N (· = p)) (hM : Moves S g Λ M (fun _ => True)) : latV S N (· = p) = latV S M (· = p)

theorem **GlobalTheory.Fam.latV_N_eq_off_level** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))), GlobalTheory.FinPl.Moves (Yc D j) (f.g.b j) ((pk W).ΛY j) M (fun _ => True) → GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M' (fun _ => True) → ∀ p, p ∉ v.gP → GlobalTheory.FinPl.latV (Yc D j) (v.N j) (· = p) = GlobalTheory.FinPl.latV (Yc D j) M (· = p) ∧ GlobalTheory.FinPl.latV (Yc D (j + 1)) (v.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepSelfDualStatement

theorem **_root_.GlobalTheory.T.descent_data_step_self_dual** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))), GlobalTheory.FinPl.Moves (Yc D j) (f.g.b j) ((pk W).ΛY j) M (fun _ => True) → GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M' (fun _ => True) → ∀ p, p ∉ v.gP → GlobalTheory.FinPl.latV (Yc D j) (v.N j) (· = p) = GlobalTheory.FinPl.latV (Yc D j) M (· = p) ∧ GlobalTheory.FinPl.latV (Yc D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepVector

theorem **exists_o** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {S T : HSp D} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (q : GlobalTheory.FinPl.SplitPt D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hS : IsSelfDualC S M) (hT : IsSelfDualC T M') : ∃ o : WO S T →₀ ℂ, (∀ x ∈ o.support, IsHalf (q.e + cA D q.e) x) ∧ ∀ Tm, (ofSplit q).IsModel ψ FS FT Tm → ∀ Φ₀ Φ₁, IsOne Φ₁ → Φ₀.1 = Set.indicator {x | (ofSplit q).up FS FT x ∈ latO S T (M : Set _) (M' : Set _) (· = q.v)} (fun _ => 1) → Tm Φ₁ = comb ψ S T o (Tm Φ₀)

theorem **idem_eq** {D : D7} {P : Finset (Pl D.host.E)} {e : AE D} (he : IsIdemSum e P) {n : ℕ} (a : Fin n → AE D) (pl : Fin n ≃ P) (ha : ∀ i, Formalism.L2.NTheta.IsIdemAt (a i) (pl i).1) : e = ∑ i, a i

theorem **isThetaAt_smul** {D : D7} {m n : ℕ} {ψ : AddChar (AF D.host.F) ℂ} {B : Mx D m n → Mx D m n → AF D.host.F} {hs : Mx D m n → (Mx D m n → ℂ) → (Mx D m n → ℂ)} {e : AE D} {Λ : AddSubgroup (Mx D m n)} {c : ℂ} {f₀ Φ f : Mx D m n → ℂ} {κ : ℂ} (hκ : κ ≠ 0) (h : IsThetaAt ψ B hs e Λ c f₀ (κ • Φ) f) : IsThetaAt ψ B hs e Λ c f₀ Φ (κ⁻¹ • f)

theorem **GlobalTheory.MidDat.moved_datum_prod** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1))

((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∀ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))), IsSelfDualC (Yc D j) M → IsSelfDualC (Yc D (j + 1)) M' → GlobalTheory.FinPl.Moves (Yc D j) (f.g.b j) ((pk W).ΛY j) M (fun _ => True) → GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M' (fun _ => True) → Gl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataStepVectorStatement

theorem **_root_.GlobalTheory.T.descent_data_step_vector** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∀ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))), IsSelfDualC (Yc D j) M → IsSelfDualC (Yc D (j + 1)) M' → GlobalTheory.FinPl.Moves (Yc D j) (f.g.b j) ((pk W).ΛY j) M (fun _ => True) → GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M' (fun _ =>

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataTopStep

theorem **combAt_mono'** {P Q : GlobalTheory.FinPl.Pl D → Prop} (h : ∀ x, P x → Q x) {c : WO S T →₀ ℂ} (hc : GlobalTheory.FinPl.CombAt P c) : GlobalTheory.FinPl.CombAt Q c

theorem **vac_mono'** {LS : GlobalTheory.FinPl.Lat S} {LT : GlobalTheory.FinPl.Lat T} (P : GlobalTheory.FinPl.Pl D → Prop) {f : WO S T → ℂ} (h : GlobalTheory.FinPl.Vac ψ S T LS LT (fun _ => True) f) : GlobalTheory.FinPl.Vac ψ S T LS LT P f

theorem **locPl_combAt** {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (Lp : LocPl ψ S T M M') : GlobalTheory.FinPl.CombAt (· = Lp.q.v) Lp.o ∧ GlobalTheory.FinPl.CombAt (· = Lp.q.v) Lp.e

theorem **comm_list** (hψ : IsStdChar D ψ (-1)) {n : ℕ} (pv : Fin n → GlobalTheory.FinPl.Pl D) (hv : Function.Injective pv) (a b : Fin n → (WO S T →₀ ℂ)) (ha : ∀ i, GlobalTheory.FinPl.CombAt (· = pv i) (a i)) (hb : ∀ i, GlobalTheory.FinPl.CombAt (· = pv i) (b i)) : ∀ l : List (Fin n), l.Nodup → ∀ g, l.foldr (fun i g' => comb ψ S T (b i) (comb ψ S T (a i) g')) g = l.foldr (fun i g' => comb ψ S T (b i) g') (l.foldr (fun i g' => comb ψ S T (a i) g') g)

theorem **comp_list** (hψ : IsStdChar D ψ (-1)) {n : ℕ} (Q : GlobalTheory.FinPl.Pl D → Prop) (a : Fin n → (WO S T →₀ ℂ)) (ha : ∀ i, GlobalTheory.FinPl.CombAt Q (a i)) : ∀ l : List (Fin n), ∃ o, GlobalTheory.FinPl.CombAt Q o ∧ ∀ g, comb ψ S T o g = l.foldr (fun i g' => comb ψ S T (a i) g') g

theorem **latV_eq_of_moves** {V : HSp D} {g : GA D V} {Λ M N : GlobalTheory.FinPl.Lat V} {P : GlobalTheory.FinPl.Pl D → Prop} (hM : Submodule.IsLattice D.host.E M) (hN : Submodule.IsLattice D.host.E N) (h1 : GlobalTheory.FinPl.Moves V g Λ M P) (h2 : GlobalTheory.FinPl.Moves V g Λ N P) : GlobalTheory.FinPl.latV V M P = GlobalTheory.FinPl.latV V N P

theorem **datum_top** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j (h : j + 2 * 0 + 2 = D.R.m), ∀ d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (M' : GlobalTheory.FinPl.Lat D.sp.X') (o' : WO (Yc D (j + 1)) D.sp.X' →₀ ℂ), IsSelfDualC (Yc D (j + 1)) M ∧ IsSelfDualC D.sp.X' M' ∧ GlobalTheory.FinPl.CombAt (· ∈ v.gP) o' ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M M' (· ∈ v.gP) (omT (pk W) W.P.A3.A.μ0 j h (f.g.b (j + 1), f.g.bT) (vacT (pk W) j h)) ∧ GlobalTheory.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentDataTopStepStatement

theorem **_root_.GlobalTheory.T.descent_data_top_step** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j (h : j + 2 * 0 + 2 = D.R.m), ∀ d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (M' : GlobalTheory.FinPl.Lat D.sp.X') (o' : WO (Yc D (j + 1)) D.sp.X' →₀ ℂ), IsSelfDualC (Yc D (j + 1)) M ∧ IsSelfDualC D.sp.X' M' ∧ GlobalTheory.FinPl.CombAt (· ∈ v.gP) o' ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M M' (· ∈ v.gP) (omT (pk W) W.P.A3.A.μ0 j h (f.g.b (j + 1), f.g.bT)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentEVMemLevel

theorem **descent_eV_mem_level** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → LevHyp W → (∀ t F, (cxW W).0 t ((cxW W).jR t (W.Q.F.eV (cxW W).zs t F)) ∈ (cxW W).EEl true)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentTopSelfDual

theorem **isWeil_top** (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) : ∃ χS χT, IsWeilDatumO (pk W).ψ (Yc D (j + 1)) D.sp.X' χS χT (omT (pk W) W.P.A3.A.μ0 j h)

theorem **moved_vacuum** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ j (h : j + 2 * 0 + 2 = D.R.m) d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → IsSelfDualC (Yc D (j + 1)) ((pk W).ΛY (j + 1)) ∧ IsSelfDualC D.sp.X' (pk W).ΛX ∧ ∃ M : GlobalTheory.FinPl.Lat (Yc D (j + 1)), IsSelfDualC (Yc D (j + 1)) M ∧ (∀ P, GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M P) ∧ GlobalTheory.FinPl.Vac f.c.ψ (Yc D (j + 1)) D.sp.X' M f.g.LT (fun _ => True) (omT (pk W) W.P.A3.A.μ0 j h (f.g.b (j + 1), f.g.bT) (vacT (pk W) j h))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentTopSelfDualStatement

theorem **_root_.GlobalTheory.T.descent_top_self_dual** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ j (h : j + 2 * 0 + 2 = D.R.m) d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → IsSelfDualC (Yc D (j + 1)) ((pk W).ΛY (j + 1)) ∧ IsSelfDualC D.sp.X' (pk W).ΛX ∧ ∃ M : GlobalTheory.FinPl.Lat (Yc D (j + 1)), IsSelfDualC (Yc D (j + 1)) M ∧ (∀ P, GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M P) ∧ GlobalTheory.FinPl.Vac f.c.ψ (Yc D (j + 1)) D.sp.X' M f.g.LT (fun _ => True) (omT (pk W) W.P.A3.A.μ0 j h (f.g.b (j + 1), f.g.bT) (vacT (pk W) j h))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentTopStandardCharacter

theorem **down_moved** (p : Place D k) (GS : Frame k S.rank Y') (GT : Frame k T.rank Y) (g : GA D S) (h : GA D T) (x : WO S T) : p.down (moved p GS g) (moved p GT h) (actO S T g h x) = p.down GS GT x

theorem **coords_moved** (p : Place D k) (GS : Frame k S.rank Y') (GT : Frame k T.rank Y) {π' : Datum.EAf D →+* k} (hπ : ∀ a, π' (toEAf D a) = p.π a) {ι ι' : Type} [Fintype ι] [DecidableEq ι] [Fintype ι'] [DecidableEq ι'] (LS : LatFrame k ι' Y') (LT : LatFrame k ι Y) (ea : ι' ≃ Fin S.rank) (eb : ι ≃ Fin T.rank) {βS : GL (Fin S.rank) (Datum.EAf D)} {βT : GL (Fin T.rank) (Datum.EAf D)} {g : GA D S} {h : GA D T} (mS : MovedTo p S g βS ((GS.β ∗ LS.bas.val).submatrix id fun i => LS.ord (ea.symm i))) (mT : MovedTo p T h βT ((GT.β ∗ LT.bas.val).submatrix id fun i => LT.ord (eb.symm i))) (x : WO S T) : LT.coords LS (p.down GS GT x) = ((frMapO S T βS βT (actO S T g⁻¹ h⁻¹ x)).map π').submatrix eb ea

theorem **isHalf_actO_iff** {e : AE D} (g : GA D S) (h : GA D T) (x : WO S T) : IsHalf e (actO S T g h x) ↔ IsHalf e x

theorem **part_int** (q : GlobalTheory.FinPl.SplitPt D k) {π' : Datum.EAf D →+* k} (hπ : ∀ a, π' (toEAf D a) = q.pr a) (hint : ∀ z : Datum.EAf D, z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ π' z ∈ 𝒪[k]) {z : Datum.EAf D} (hz : π' z ∈ 𝒪[k]) : toEAf D q.e ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **sum_part_mem** {R R' ι : Type} [CommRing R] [CommRing R'] [Fintype ι] (φ : R →+* R') (A : Subring R') (a : ι → R) (z : R') (h : ∀ i, φ (a i) ∗ z ∈ A) : φ (∑ i, a i) ∗ z ∈ A

theorem **idem_eq** {P : Finset (Pl D.host.E)} {e : AE D} (he : IsIdemSum e P) {n : ℕ} (a : Fin n → AE D) (pl : Fin n ≃ P) (ha : ∀ i, Formalism.L2.NTheta.IsIdemAt (a i) (pl i).1) : e = ∑ i, a i

theorem **rank_first** (G : GLoc v) : D.R.k + (G.s + 1) = D.R.r

theorem **red_iff** {w : Pl D.host.E} (G : GLoc v) {π' : Datum.EAf D →+* G.k} (hπ : ∀ a, π' (toEAf D a) = G.q.pr a) (hred : IsRed (resS D.host.F w) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype)) (x : G.S.P.J) (g : GL (Fin D.R.r) (kP w)) (M : Matrix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.k ⊕ Fin (G.s + 1)) (Formalism.L2.Herm.intAf D.host.F D.host.E)) (hM : g.val.submatrix (firstOrd (rank_first G)) (firstOrd (rank_first G)) = M.map (resS D.host.F w)) : Red G w x g ↔ MaxM ((x : GL _ G.k).val − M.map (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype))

theorem **chS_base** (hPk : PkHyp W) (j : ℕ) : (GlobalTheory.BaseChain.com W).chS j = (Lift.chrStep (pk W).mu0 (pk W).eta j).1

theorem **locF_iter** {w : Pl D.host.E} (G : GLoc v) {π' : Datum.EAf D →+* G.k} (hπ : ∀ a, π' (toEAf D a) = G.q.pr a) (hred : IsRed (resS D.host.F w) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype)) {j a b : ℕ} (ha : a = D.R.r + j ∗ D.R.t) (hb : b = D.R.r + (j + 1) ∗ D.R.t) (hr : D.R.r ≤ a) {ea : Ix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t) j ≃ Fin a} {eb : Ix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t) (j + 1) ≃ Fin b} (oa : GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.zPos ea) (ob : GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.zPos eb) (χ : ℕ → (kP w)ˣ →* ℂˣ) (hχ : ∀ c, c ≤ j → GlobalTheory.FunObj.Comparison₂.Compat (resS D.host.F w) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype) (χ (c + 1)) (G.S.μ (c + 1))) {N : Matrix (Fin b) (Fin a) (Datum.EAf D)} (hN : IsIntM N) : locF D.R.t j hr (Red G w) (fun y => G.S.P.lam y) χ (redM w N) = G.S.P.iterPhi (Fin D.R

theorem **chi_compat** (hPk : PkHyp W) (G : GLoc v) (w : (pk W).Pg) (hq : Formalism.L2.NTheta.IsIdemAt G.q.e w.1) {π' : Datum.EAf D →+* G.k} (hπ : ∀ a, π' (toEAf D a) = G.q.pr a) (hred : IsRed (resS D.host.F w.1) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype)) {c : ℕ} (h1 : 1 ≤ c) (h2 : c ≤ D.R.m) (hμ : (ofSplit G.q).IsComp (Lift.chrStep (pk W).mu0 (pk W).eta c).1⁻¹ (G.S.μ c)) : GlobalTheory.FunObj.Comparison₂.Compat (resS D.host.F w.1) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype) ((pk W).chi c w.1) (G.S.μ c)

theorem **place** (hPk : PkHyp W) {n : ℕ} {G : Fin n → GLoc v} {pl : Fin n ≃ (pk W).Pg} (hGR : GRead f G pl) (i : Fin n) {j : ℕ} (h : j + 2 ∗ 0 + 2 = D.R.m) (x : WO (Yc D (j + 1)) D.sp.X') {N : Matrix (Fin D.sp.X'.rank) (Fin (Yc D (j + 1)).rank) (Datum.EAf D)} (hN : N = frMapO (Yc D (j + 1)) D.sp.X' ((pk W).βY (j + 1)) (pk W).βX (actO _ _ (f.g.b (j + 1))⁻¹ f.g.bT⁻¹ x)) : (((G i).S.T (Fin D.R.t)).PhiTop ((ofSplit (G i).q).down (GlobalTheory.TopLocData.frS (G i) h) (G i).FT x) ≠ 0 → ∀ a b, toEAf D (G i).q.e ∗ N a b ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) ∧ (IsIntM N → locF D.R.t (j + 1) (r_le_Y (j + 1)) (Red (G i) (pl i).1) (fun y => (G i).S.P.lam y) (fun c => (pk W).chi c (pl i).1) (redM (pl i).1 N) = ((G i).S.T (Fin D.R.t)).PhiTop ((ofSplit (G i).q).down (GlobalTheory.TopLocData.frS (G i) h) (G i).FT x))

theorem **datum_eq** (hPk : PkHyp W) {n : ℕ} {G : Fin n → GLoc v} {pl : Fin n ≃ (pk W).Pg} (hGR : GRead f G pl) {j : ℕ} (h : j + 2 ∗ 0 + 2 = D.R.m) (x : WO (Yc D (j + 1)) D.sp.X') : (if IsHalf (∑ i, (G i).q.e) (actO _ _ (f.g.b (j + 1))⁻¹ f.g.bT⁻¹ x) ∧ IsIntM (frMapO (Yc D (j + 1)) D.sp.X' ((pk W).βY (j + 1)) (pk W).βX (actO _ _ (f.g.b (j + 1))⁻¹ f.g.bT⁻¹ x)) then omegaG (datY _ D.R.r D.R.t (j + 1) (r_le_Y (j + 1)) (pk W).Pg ⌀ (lvP W) (pk W).chi (fun _ => 1) Frm.B) ((pk W).chi (j + 1 + 1)) (fun PP => redM PP (frMapO (Yc D (j + 1)) D.sp.X' ((pk W).βY (j + 1)) (pk W).βX (actO _ _ (f.g.b (j + 1))⁻¹ f.g.bT⁻¹ x))) ∗ omegaQ ⌀ (datY _ D.R.r D.R.t (j + 1) (r_le_Y (j + 1)) (pk W).Pg ⌀ (lvP W) (pk W).chi (fun _ => 1) Frm.B).r (fun _ => 1) 1 (fun PP => redM PP (frMapO (Yc D (j + 1)) D.sp.X' ((pk W).βY (j + 1)) (pk W).βX (actO _ _ (f.g.b (j + 1))⁻¹ f.g.bT⁻¹ x))) else 0) = if IsHalf (∑ i, (G i).q.e) x the

theorem **weil_top** (W : RealPack ex) (j : ℕ) (h : j + 2 ∗ 0 + 2 = D.R.m) : ∃ χS χT, IsWeilDatumO (pk W).ψ (Yc D (j + 1)) D.sp.X' χS χT (omT (pk W) W.P.A3.A.μ0 j h)

theorem **moved_top_datum** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg) (hGP : GPin G), GRead f G pl → ∀ j (h : j + 2 ∗ 0 + 2 = D.R.m) (hψ : IsStdChar D f.c.ψ (−1)) d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → IsSelfDualC (Yc D (j + 1)) ((pk W).ΛY (j + 1)) → IsSelfDualC D.sp.X' (pk W).ΛX → ∀ (M : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (hM : IsSelfDualC (Yc D (j + 1)) M) (hLT : IsSelfDualC D.sp.X' f.g.LT), (∀ P, GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M P) → ∃ κ : ℂ, κ ≠ 0 ∧ ∀ d, IsDG W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentTopStandardCharacterStatement

theorem **_root_.GlobalTheory.T.descent_top_standard_character** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg) (hGP : GPin G), GRead f G pl → ∀ j (h : j + 2 ∗ 0 + 2 = D.R.m) (hψ : IsStdChar D f.c.ψ (−1)) d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → IsSelfDualC (Yc D (j + 1)) ((pk W).ΛY (j + 1)) → IsSelfDualC D.sp.X' (pk W).ΛX → ∀ (M : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (hM : IsSelfDualC (Yc D (j + 1)) M) (hLT : IsSelfDualC D.sp.X' f.g.LT), (∀ P, GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentTopStepVector

theorem **main** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc

v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j (h : j + 2 * 0 + 2 = D.R.m) d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → IsSelfDualC (Yc D (j + 1)) ((pk W).ΛY (j + 1)) → IsSelfDualC D.sp.X' (pk W).ΛX → ∀ M : GlobalTheory.FinPl.Lat (Yc D (j + 1)), IsSelfDualC (Yc D (j + 1)) M → (∀ P, GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M P) → ∃ (Lp : Fin n → LocPl f.c.ψ (Yc D (j + 1)) D.sp.X' M f.g.LT) (κ : ℂ), κ ≠ 0 ∧ (∀ i, (Lp i).q.v = (G i).q.v) ∧ (∀ i, In

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DescentTopStepVectorStatement

theorem **_root_.GlobalTheory.T.descent_top_step_vector** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j (h : j + 2 * 0 + 2 = D.R.m) d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → IsSelfDualC (Yc D (j + 1)) ((pk W).ΛY (j + 1)) → IsSelfDualC D.sp.X' (pk W).ΛX → ∀ M : GlobalTheory.FinPl.Lat (Yc D (j + 1)), IsSelfDualC (Yc D (j + 1)) M → (∀ P, GlobalTheory.FinPl.Moves (Yc D (j + 1)) (f.g.b (j + 1)) ((pk W).ΛY (j + 1)) M P) → ∃ (Lp : Fin n → LocPl f.c.ψ (Yc D (j + 1)) D.sp.X' M f.g.LT) (κ : ℂ), κ ≠ 0 ∧

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DetIntegralUnitIdeleOfIntegralFinitePart

theorem **det_mem_subring** {R : Type} [CommRing R] (S : Subring R) {n : ℕ} (M : Matrix (Fin n) (Fin n) R) (h : ∀ i j, M i j ∈ S) : M.det ∈ S

theorem **det_finM** (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (g : GA D V) : (finM V hn g).det = toEAf D (Formalism.L2.Theta.mt V g).det

theorem **det_int_unit** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (g : GA D V) (k : GL (Fin n) (EF D)), finM V hn g = k.1 → (∀ i j, k.1 i j ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) → (∀ i j, (k⁻¹).1 i j ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) → ∀ d : (Formalism.L2.Theta.AE D)ˣ, (d : Formalism.L2.Theta.AE D) = (Formalism.L2.Theta.mt V g).det → finPart D (d : Formalism.L2.Theta.AE D) ∈ intAE D ∧ finPart D ((d⁻¹ : (Formalism.L2.Theta.AE D)ˣ) : Formalism.L2.Theta.AE D) ∈ intAE D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DetIntegralUnitIdeleOfIntegralFinitePartStatement

theorem **_root_.GlobalTheory.T.det_integralUnitIdele_of_integral_finitePart** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (g : GA D V) (k : GL (Fin n) (EF D)), finM V hn g = k.1 → (∀ i j, k.1 i j ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) → (∀ i j, (k⁻¹).1 i j ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) → ∀ d : (Formalism.L2.Theta.AE D)ˣ, (d : Formalism.L2.Theta.AE D) = (Formalism.L2.Theta.mt V g).det → finPart D (d : Formalism.L2.Theta.AE D) ∈ intAE D ∧ finPart D ((d⁻¹ : (Formalism.L2.Theta.AE D)ˣ) : Formalism.L2.Theta.AE D) ∈ intAE D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagCorollary

theorem **dstar_pos_of** (D : D7) (u : D.what₂.Word) : 0 < D.what₂.Dstar

theorem **diagS_mono** {D : D7} {u : D.what₂.Word} {e e' : ℝ} (h : DiagS D u e) (he : e ≤ e') : DiagS D u e'

theorem **kfk_nonneg** (D : D7) : 0 ≤ kfk D

theorem **diag_corollary** : (∀ (D : D7) (u : D.what₂.Word) (A x κ : ℝ), 0 ≤ A → TwoTerm D u A x → (d0 D u : ℝ) ≤ κ * D.what₂.Dstar → DiagS D u (A * κ + x)) ∧ (∀ (D : D7) (u : D.what₂.Word) (L C : ℝ), 0 ≤ C * mfL D L → SDiag D u L C → Small D u L C → DiagS D u (eDiag D L C)) ∧ (∀ (D : D7) (u : D.what₂.Word) (Γ₀ : ℝ), SAbs D u Γ₀ ↔ TwoTerm D u Γ₀ (1 / 4))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagSOfSDiagAt

theorem **diagS_of_sDiagAt** : ∀ (D : D7) (u : D.what₂.Word) (Γ c : ℝ), 1 ≤ Γ → c ≤ 1 → SDiagAt D u Γ c → (D.duS u : ℝ) * Γ ^ 2 ≤ D.what₂.Dstar → GlobalTheory.Core.Rk.DiagS D u (8 / Γ + 6 * Real.exp (-(c * D.R.k * D.R.f * Real.log D.R.k)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalAdelicEntriesCongrOneOfRatiosAtPlace

theorem **finMul** {D : D7} {x y : Af D.host.F ⊗[D.host.F] D.host.E} (hx : x ∈ intAE D) (hy : y ∈ intAE D) : x * y ∈ intAE D

theorem **intAt_add** {D : D7} {v : Pl D} {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a + b ∈ intAt D (· = v)

theorem **intAt_mul** {D : D7} {v : Pl D} {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a * b ∈ intAt D (· = v)

theorem **mul_sub_one** {D : D7} {v : Pl D} {a b : AE D} (ha : a - 1 ∈ intAt D (· = v)) (hb : b - 1 ∈ intAt D (· = v)) : a * b - 1 ∈ intAt D (· = v)

theorem **inv_diag_mul** {R : Type*} [CommRing R] {ρ : ℕ} (t : GL (Fin ρ) R) (ht : ∀ i j, i ≠ j → t.1 i j = 0) (j : Fin ρ) : (t⁻¹).1 j j * t.1 j j = 1

theorem **diag_int** : ∀ (D : D7) (ρ : ℕ) (v : Pl D) (t : GL (Fin ρ) (AE D)), (∀ i j, i ≠ j → t.1 i j = 0) → (∀ i, (t⁻¹).1 i i - 1 ∈ intAt D (· = v)) → (∀ i j : Fin ρ, j.1 = i.1 + 1 → t.1 i i * (t⁻¹).1 j j - 1 ∈ intAt D (· = v)) → (∀ i : Fin ρ, i.1 + 1 = ρ → t.1 i i - 1 ∈ intAt D (· = v)) → ∀ i, t.1 i i - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalAdelicEntriesCongrOneOfRatiosAtPlaceStatement

theorem **_root_.GlobalTheory.T.diagonalAdelic_entries_congr_one_of_ratios_at_place** : ∀ (D : D7) (ρ : ℕ) (v : Pl D) (t : GL (Fin ρ) (AE D)), (∀ i j, i ≠ j → t.1 i j = 0) → (∀ i, (t⁻¹).1 i i - 1 ∈ intAt D (· = v)) → (∀ i j : Fin ρ, j.1 = i.1 + 1 → t.1 i i * (t⁻¹).1 j j - 1 ∈ intAt D (· = v)) → (∀ i : Fin ρ, i.1 + 1 = ρ → t.1 i i - 1 ∈ intAt D (· = v)) → ∀ i, t.1 i i - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalRowBoundFromWindowWidthThree

theorem **exp_add_exp_le_exp_max_succ** {C₁ C₂ y : ℝ} (hy : 1 ≤ y) : Real.exp (C₁ * y) + Real.exp (C₂ * y) ≤ Real.exp ((max C₁ C₂ + 1) * y)

theorem **diagonalRow_bound_from_windowWidthThree** : Formalism.L2.SG.Pf.WuDecayOfGap → On {3} VanishingGramTermsBound → On {3} CollS → OneOn {3} true → OneOn {3} false → GlobalTheory.Core.Rk.RowDiag (fun D L C' => GlobalTheory.Core.Rk.IsScaleWithBoundedGamma D L C' ∧ ∀ u, GlobalTheory.Core.Rk.RankMu D u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalRowOfSiegelWeilUnitary

theorem **rowDiag_of_siegelWeilUnitary** (hSW : HeckeCharacterZeroLocus.SiegelWeilU.SiegelWeilUnitaryConvergent) : GlobalTheory.Core.Rk.RowDiag (fun D L C' => GlobalTheory.Core.Rk.IsScaleWithBoundedGamma D L C' ∧ ∀ u, GlobalTheory.Core.Rk.RankMu D u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalRowOfSiegelWeilUnitaryStatement

theorem **_root_.GlobalTheory.T.diagonalRow_of_siegelWeil_unitary** : HeckeCharacterZeroLocus.SiegelWeilU.SiegelWeilUnitaryConvergent → GlobalTheory.Core.Rk.RowDiag (fun D L C' => GlobalTheory.Core.Rk.IsScaleWithBoundedGamma D L C' ∧ ∀ u, GlobalTheory.Core.Rk.RankMu D u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalTheoremOfFourPointBounds

theorem **exp_neg_le_two_fifths** {x : ℝ} (hx : 2 ≤ x) : Real.exp (-x) ≤ 2 / 5 * Real.exp (-x / 2)

theorem **diag_arith** {H N₁ N₂ P s dd d E₂ E Γ₁ Γ₂ Γ : ℝ} (hH : H ≤ 3 / 2 * N₁ + 3 * N₂) (h₁ : N₁ ≤ 2 * P * (s * dd + E₂ * d + Γ₁ * dd + Γ₂ * dd)) (h₂ : N₂ ≤ d * E) (hP : P ≤ 6 / 5) (hs0 : 0 ≤ s) (hs1 : s ≤ 1) (hdd : 0 ≤ dd) (hd : 0 ≤ d) (hE₂ : 0 ≤ E₂) (hE : E ≤ 2 / 5 * E₂) (hΓ₁ : 0 ≤ Γ₁) (hΓ₂ : 0 ≤ Γ₂) (hΓ : Γ₁ + Γ₂ ≤ Γ) : H ≤ 4 * ((Γ + 1) * dd + d * E₂) + 2 * d * E

theorem **sDiagAt_of_points** : ∀ (D : D7) (c : Formalism.L2.SG.ThmSOwn.ConcK D) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT C0 Γ₁ Γ₂ Γ : ℝ) (u : D.what₂.Word), 29 ≤ C' → 29 ≤ Cl → Cnt.Hyp0 D large₂ C' Cl Cg cb Cu cT CT C0 → 0 ≤ Γ₁ → 0 ≤ Γ₂ → Γ₁ + Γ₂ ≤ Γ → VanishingGramTermsBound c.Y u → CollS c.Y u → (c.Y.tO u u).pairSum ((c.Y.tO u u).one (c.Y.tO u u).genSet) ≤ Γ₁ * ddC c.X c.Y u u → (c.Y.tO u u).pairSum ((c.Y.tO u u).one (c.Y.tO u u).ngSet) ≤ Γ₂ * ddC c.X c.Y u u → SDiagAt D u Γ (1 / 2)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiagonalTheoremOfFourPointBoundsStatement

theorem **_root_.GlobalTheory.T.diagonalTheorem_of_fourPointBounds** : ∀ (D : D7) (c : Formalism.L2.SG.ThmSOwn.ConcK D) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT C0 Γ₁ Γ₂ Γ : ℝ) (u : D.what₂.Word), 29 ≤ C' → 29 ≤ Cl → Cnt.Hyp0 D large₂ C' Cl Cg cb Cu cT CT C0 → 0 ≤ Γ₁ → 0 ≤ Γ₂ → Γ₁ + Γ₂ ≤ Γ → VanishingGramTermsBound c.Y u → CollS c.Y u → (c.Y.tO u u).pairSum ((c.Y.tO u u).one (c.Y.tO u u).genSet) ≤ Γ₁ * ddC c.X c.Y u u → (c.Y.tO u u).pairSum ((c.Y.tO u u).one (c.Y.tO u u).ngSet) ≤ Γ₂ * ddC c.X c.Y u u → SDiagAt D u Γ (1 / 2)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiamondTranslateSumEqZeroOfChartRepresentatives

theorem **conj_level** {G ι : Type*} [Group G] (U : Subgroup G) (χ : U →* ℂˣ) (J : List ι) (f : ι → G) (x κ' : G) (κ : U) (hκ : κ.1 = (J.map f).prod * κ') (hc : x * κ' = κ' * x) (h1 : ∀ i, ∃ h : f i ∈ U, χ ⟨_, h⟩ = 1) (h2 : ∀ i, ∃ h : x⁻¹ * f i * x ∈ U, χ ⟨_, h⟩ = 1) : ∃ κ'' : U, κ.1 * x = x * κ''.1 ∧ χ κ'' = χ κ

theorem **real_le_off** (V : HSp D) (P : Pl D → Prop) : GinfC V ≤ grpOff V P

theorem **grpOff_anti** (V : HSp D) {P P' : Pl D → Prop} (h : ∀ w, P' w → P w) : grpOff V P ≤ grpOff V P'

theorem **GlobalTheory.AdForm.DiaReduce.rd_laws** {B : Base} {X : PreX B.toPre} (I : TC B X) (hok : I.Ok) (hpins : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch) (hA : A.Is) (θ : InfFam (I.rec7 l Z)) (hθ : IsInfFam θ) (i : ℕ) (hi : i ≤ B.R.m) (φ) (hφ : φ ∈ (I.fm l u a).Fi i) (β : Ch i) (y : (I.chain l u a).Y i) (k : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i)) (hk : k ∈ GinfC ((I.rec7 l Z).sp.X i)) : ∃ y' : (I.chain l u a).Y i, ∀ γ ∈ GQ ((I.rec7 l Z).sp.X i), ∀ g, ∀ κ : A.U i β, A.rd θ i hi β φ y (γ * (g * k * κ.1)) = (A.chi i β κ : ℂ) * A.rd θ i hi β φ y' g

theorem **GlobalTheory.AdForm.diaSum_eq_zero_of_reps** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m), ∀ φ ∈ (I.fm l u a).Fi j, ∀ (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ g : 𝓚[k]ˣ → GL (Fin ((I.rec7 l Z).sp.X j).rank) k, (∀ s, g s • Q j i = s • Q j i) → ∀ b y, (∀ (c : {c // A.chOf j c = b}) y', ∑ᶠ s, A.rd θ j hj b φ y' (A.h j b c * (β j i).gl (g s)) = 0) →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiamondTranslateSumEqZeroOfChartRepresentativesStatement

theorem **_root_.GlobalTheory.T.diamondTranslateSum_eq_zero_of_chartRepresentatives** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m), ∀ φ ∈ (I.fm l u a).Fi j, ∀ (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ g : 𝓚[k]ˣ → GL (Fin ((I.rec7 l Z).sp.X j).rank) k, (∀ s, g s • Q j i = s • Q j i) → ∀ b y, (∀ (c : {c // A.chOf j c = b}) y', ∑ᶠ s, A.rd θ j hj b φ y' (A.h j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiamondTranslatesScalarFormsSumIff

theorem **diaSum_iff** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m), ∀ φ ∈ (I.fm l u a).Fi j, ∀ (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ g : 𝓚[k]ˣ → GL (Fin ((I.rec7 l Z).sp.X j).rank) k, (∀ s, g s • Q j i = s • Q j i) → ((∀ b y, ∑ᶠ s, rTrans ((β j i).gl (g s)) (A.rd θ j hj b φ y) = 0) ↔ ∀ c, ∑ᶠ s : ((I.atT l u a).Fv v)ˣ, (I.fm l u a).dilT j v s c (φ ((I.f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiamondTranslatesScalarFormsSumIffStatement

theorem **_root_.GlobalTheory.T.diamondTranslates_scalarForms_sum_iff** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m), ∀ φ ∈ (I.fm l u a).Fi j, ∀ (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ g : 𝓚[k]ˣ → GL (Fin ((I.rec7 l Z).sp.X j).rank) k, (∀ s, g s • Q j i = s • Q j i) → ((∀ b y, ∑ᶠ s, rTrans ((β j i).gl (g s)) (A.rd θ j hj b φ y) = 0) ↔ ∀ c, ∑ᶠ s : ((I.atT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DictionaryColumnScalarFormEqTraceColumnForm

theorem **colForm_of_dec** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (B : SBes X h u ψ (D.sp.X 0)) (F : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F) (hK : X.LevelSubgroupCharacter) (θ : InfFam D) (a₀ : GA D (D.sp.X 0)) (hθ : IsInfFam θ) (hE : IdE X) (hAd : B.Adapted tr.wt) (hH0 : IsTransportAdelicCorrection tr a₀) (hLM : Formalism.L2.SG.ColDict.LevelMatch X tr hK) (T : Cu X u hK) (y : (D.mem u).C.Y 0) (g : GA D (D.sp.X 0)) (c : (D.mem u).C.cls 0) (γ : GL (Fin (D.R.N 0)) D.host.E) (κ : F.cls0.lev.U) (hγ : γ ∈ Formalism.UG.UJ (Formalism.L2.Forms.sigD D : D.host.E →+* D.host.E) (Formalism.L2.Forms.gramN D 0 (Nat.zero_le _))) (hx : GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) g = (Formalism.UG.bc (Formalism.UG.i

theorem **GlobalTheory.Core.Mi.colDict_rd_eq_colForm** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (B : SBes X h u ψ (D.sp.X 0)) (F : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F) (hK : X.LevelSubgroupCharacter) (θ : InfFam D) (a₀ : GA D (D.sp.X 0)), IsInfFam θ → GlobalTheory.Core.Mi.IdE X → B.Adapted tr.wt → IsTransportAdelicCorrection tr a₀ → Formalism.L2.SG.ColDict.LevelMatch X tr hK → ∃ ξ : (D.mem u).C.Y 0 → X.Vpi u, Submodule.span ℂ (Set.range ξ) = ⊤ ∧ ∀ (T : Cu X u hK) (y : (D.mem u).C.Y 0) (g : GA D (D.sp.X 0)), (chart0 F).rd θ (Formalism.L2.SG.ColDict.colDictL X tr hK T) y g = B.colForm T.1 (ξ y) (g * a₀⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DictionaryColumnScalarFormEqTraceColumnFormStatement

theorem **_root_.GlobalTheory.T.dictionaryColumnScalarForm_eq_traceColumnForm** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (B : SBes X h u ψ (D.sp.X 0)) (F : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F) (hK : X.LevelSubgroupCharacter) (θ : InfFam D) (a₀ : GA D (D.sp.X 0)), IsInfFam θ → GlobalTheory.Core.Mi.IdE X → B.Adapted tr.wt → IsTransportAdelicCorrection tr a₀ → Formalism.L2.SG.ColDict.LevelMatch X tr hK → ∃ ξ : (D.mem u).C.Y 0 → X.Vpi u, Submodule.span ℂ (Set.range ξ) = ⊤ ∧ ∀ (T : Cu X u hK) (y : (D.mem u).C.Y 0) (g : GA D (D.sp.X 0)), (chart0 F).rd θ (Formalism.L2.SG.ColDict.colDictL X tr hK T) y g = B.colForm T.1 (ξ y) (g * a₀⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DifferentiableAtLimitOfDirectionalDerivsConverge

theorem **tendstoLocallyUniformly_sum_smul_const** {E : Type} [TopologicalSpace E] {F : Type} [NormedAddCommGroup F] [NormedSpace ℂ F] {A : Type} (s : Finset A) (L : A → F) (f : A → ℕ → E → ℂ) (ψ : A → E → ℂ) (h : ∀ a, TendstoLocallyUniformly (f a) (ψ a) atTop) : TendstoLocallyUniformly (fun j y => ∑ a ∈ s, f a j y • L a) (fun y => ∑ a ∈ s, ψ a y • L a) atTop

theorem **differentiableAt_of_tendstoLocallyUniformly_directional** : ∀ {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] {A : Type} [Fintype A] (v : A → E) (c : A → E →L[ℝ] ℝ), (∀ x, ∑ a, c a x • v a = x) → ∀ (φ : E → ℂ) (g : ℕ → E → ℂ) (ψ : A → E → ℂ), (∀ j, Differentiable ℝ (g j)) → TendstoLocallyUniformly g φ Filter.atTop → (∀ a, TendstoLocallyUniformly (fun j y => fderiv ℝ (g j) y (v a)) (ψ a) Filter.atTop) → ∀ y, DifferentiableAt ℝ φ y ∧ ∀ a, fderiv ℝ φ y (v a) = ψ a y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DifferentiableAtLimitOfDirectionalDerivsConvergeStatement

theorem **_root_.GlobalTheory.T.differentiableAt_limit_of_directional_derivs_converge** : ∀ {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] {A : Type} [Fintype A] (v : A → E) (c : A → E →L[ℝ] ℝ), (∀ x, ∑ a, c a x • v a = x) → ∀ (φ : E → ℂ) (g : ℕ → E → ℂ) (ψ : A → E → ℂ), (∀ j, Differentiable ℝ (g j)) → TendstoLocallyUniformly g φ Filter.atTop → (∀ a, TendstoLocallyUniformly (fun j y => fderiv ℝ (g j) y (v a)) (ψ a) Filter.atTop) → ∀ y, DifferentiableAt ℝ φ y ∧ ∀ a, fderiv ℝ φ y (v a) = ψ a y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiscreteSeriesDetSqrtCoverOfUnitaryClassificatio

theorem **discreteSeriesCover_of_classification** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.DiscreteSeriesCover

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiscreteSeriesDetSqrtCoverOfUnitaryClassificatioStatement

theorem **_root_.GlobalTheory.T.discreteSeries_detSqrtCover_of_unitary_classification** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.DiscreteSeriesCover

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiscreteSeriesModelHolomorphicPositiveSignsFirst

theorem **eq_self_of_strictMono** {n : ℕ} {F K : Fin n → Fin n} (hK : StrictMono K) (hF : StrictMono F) (hFK : ∀ x, F (K x) = x) (x : Fin n) : K x = x

theorem **holShape_signs_first** : ∀ (D : D7) (σ : D.what₂.LL →+* ℂ), (∀ ρ, Formalism.IndefR.Links.labE σ (Formalism.IndefR.Links.lab σ ρ) = tauE D ρ) → ∀ (i : Fin D.R.m) (M : DSModel (Wc D i) (Formalism.IndefR.Links.parAt (Wc D i) σ)), GlobalTheory.HolDict.HolShape M ∧ (∃ wt, GlobalTheory.HolDict.WtIs M wt) ∧ ∀ ρ a, M.ε ρ a = decide ((a : ℕ) < ρW D i ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiscreteSeriesModelHolomorphicPositiveSignsFirstStatement

theorem **_root_.GlobalTheory.T.discreteSeriesModel_holomorphic_positiveSignsFirst** : ∀ (D : D7) (σ : D.what₂.LL →+* ℂ), (∀ ρ, Formalism.IndefR.Links.labE σ (Formalism.IndefR.Links.lab σ ρ) = tauE D ρ) → ∀ (i : Fin D.R.m) (M : DSModel (Wc D i) (Formalism.IndefR.Links.parAt (Wc D i) σ)), GlobalTheory.HolDict.HolShape M ∧ (∃ wt, GlobalTheory.HolDict.WtIs M wt) ∧ ∀ ρ a, M.ε ρ a = decide ((a : ℕ) < ρW D i ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiscreteSeriesTestDataOfNoncompactGaps

private theorem **isHCParam_comp_fin_cast** {n m : ℕ} (h : n = m) {f : Fin m → ℝ} (hf : Formalism.QuantumDeterminants.IsHCParam f) : Formalism.QuantumDeterminants.IsHCParam (fun i : Fin n => f (Fin.cast h i))

theorem **isHCParam_lam_of_setting** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (v : InfinitePlace X.B.I.F) : Formalism.QuantumDeterminants.IsHCParam (X.B.lam v)

private theorem **card_filter_true_add_card_filter_false** {n : ℕ} (ε : Fin n → Bool) : (Finset.univ.filter fun a : Fin n => ε a = true).card + (Finset.univ.filter fun a : Fin n => ε a = false).card = n

theorem **two_mul_rkR_le** {n : ℕ} (ε : Fin n → Bool) : 2 * GlobalTheory.Count.rkR ε ≤ n

private theorem **natCast_rpow_three_halves_le_sq** (q : ℕ) : (q : ℝ) ^ ((3 : ℝ) / 2) ≤ (q : ℝ) ^ 2

private theorem **mul_rpow_three_halves_le_pow_four** (q n : ℕ) (h : 2 * q ≤ n) : 120 * (n : ℝ) * ((q : ℕ) : ℝ) ^ ((3 : ℝ) / 2) ≤ 100 * (n : ℝ) ^ 4

theorem **rkR_bound** {n : ℕ} (ε : Fin n → Bool) : 120 * (n : ℝ) * ((GlobalTheory.Count.rkR ε : ℕ) : ℝ) ^ ((3 : ℝ) / 2) ≤ 100 * (n : ℝ) ^ 4

private theorem **mul_sq_mul_sqrt_le_pow_four** (q n : ℕ) (h : 2 * q ≤ n) : 40 * (n : ℝ) ^ 2 * Real.sqrt (q : ℝ) ≤ 100 * (n : ℝ) ^ 4

theorem **rkR_bound_sqrt** {n : ℕ} (ε : Fin n → Bool) : 40 * (n : ℝ) ^ 2 * Real.sqrt (GlobalTheory.Count.rkR ε) ≤ 100 * (n : ℝ) ^ 4

theorem **seeData_of_gap** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D), GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → GlobalTheory.Count.IsDSData X.B X.dv ∧ GlobalTheory.Dec.LocDec X.B (100 * (D.R.r : ℝ) ^ 4) ∧ GlobalTheory.Dec.HBig X.B 40 (100 * (D.R.r : ℝ) ^ 4) ∧ (∀ v, 120 * (X.B.I.n : ℝ) * ((GlobalTheory.Count.rkR (X.B.I.ε v) : ℕ) : ℝ) ^ ((3 : ℝ) / 2) ≤ 100 * (D.R.r : ℝ) ^ 4) ∧ Integrable X.ph (Formalism.L2.SG.ProjRows.muInf X.B.I) ∧ ∀ v, Integrable (Formalism.L2.SG.ProjRows.phLoc (ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DiscreteSeriesTestDataOfNoncompactGapsStatement

theorem **_root_.GlobalTheory.T.discreteSeriesTestData_of_noncompactGaps** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D), GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → GlobalTheory.Count.IsDSData X.B X.dv ∧ GlobalTheory.Dec.LocDec X.B (100 * (D.R.r : ℝ) ^ 4) ∧ GlobalTheory.Dec.HBig X.B 40 (100 * (D.R.r : ℝ) ^ 4) ∧ (∀ v, 120 * (X.B.I.n : ℝ) * ((GlobalTheory.Count.rkR (X.B.I.ε v) : ℕ) : ℝ) ^ ((3 : ℝ) / 2) ≤ 100 * (D.R.r : ℝ) ^ 4) ∧ Integrable X.ph (Formalism.L2.SG.ProjRows.muInf X.B.I) ∧ ∀ v, Integrable (Formalism.L2.SG.ProjRows.phLoc (ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DisjointReferenceTwistPrimes

theorem **map_mem_span_range** {K L M N J : Type*} [Semiring K] [Semiring L] [AddCommMonoid M] [AddCommMonoid N] [Module K M] [Module L N] {σ : K →+* L} (f : M →ₛₗ[σ] N) {φ : J → M} (h : Submodule.span K (Set.range φ) = ⊤) (z : M) : f z ∈ Submodule.span L (Set.range fun a => f (φ a))

theorem **valMtP_injective** : Function.Injective (valMtP W Ch sel mu P σ)

theorem **rowEmb_linearIndependent** (H : ∀ b, Subgroup ((stepX Ch sel mu P).G b)) {J : Type} {φ : J → MtildeV W Ch sel mu P} (h : LinearIndependent X.K φ) : LinearIndependent ℂ fun a => rowEmb W Ch sel mu P b0 til ext tz H hd σ (φ a)

theorem **rowEmb_span** (hX : X.Ok) {J : Type} {φ : J → MtildeV W Ch sel mu P} (h : Submodule.span X.K (Set.range φ) = ⊤) : Submodule.span ℂ (Set.range fun a => rowEmb W Ch sel mu P b0 til ext tz (fun _ => ⊤) hd σ (φ a)) = ⊤

theorem **disjoint_reference_twist_primes** (h : I.Ok) : Disjoint (B.primes X).vGs (QE I.sel (I.tup I.mu0 : Chr B.fields.F B.fields.E B.fields.K))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DivisiblePontryaginAnnihilatorDuality

theorem **exists_hom** {A B : Type} [AddCommGroup A] [AddCommGroup B] (e : A → B → ℂ) (h1 : ∀ a a' b, e (a + a') b = e a b * e a' b) (h2 : ∀ a b b', e a (b + b') = e a b * e a b') (h3 : ∀ b, e 0 b = 1) (h4 : ∀ a, e a 0 = 1) (M : AddSubgroup A) (hM : ∀ a ∈ M, ∀ b, e a b = 1) : ∃ f : B →+ AddChar (A ⧸ M) ℂ, ∀ b (a : A), f b (a : A ⧸ M) = e a b

theorem **card_range** {A B : Type} [AddCommGroup A] [AddCommGroup B] (e : A → B → ℂ) (M : AddSubgroup A) (K : AddSubgroup B) (hK : ∀ b, b ∈ K ↔ ∀ a, e a b = 1) (f : B →+ AddChar (A ⧸ M) ℂ) (hf : ∀ b (a : A), f b (a : A ⧸ M) = e a b) : Nat.card f.range = Nat.card (B ⧸ K)

theorem **exists_separating** {A B : Type} [AddCommGroup A] [AddCommGroup B] (e : A → B → ℂ) (h1 : ∀ a a' b, e (a + a') b = e a b * e a' b) (h2 : ∀ a b b', e a (b + b') = e a b * e a b') (h3 : ∀ b, e 0 b = 1) (h4 : ∀ a, e a 0 = 1) (M : AddSubgroup A) (K : AddSubgroup B) [Finite (A ⧸ M)] [Finite (B ⧸ K)] (hM : ∀ a, a ∈ M ↔ ∀ b, e a b = 1) (hK : ∀ b, b ∈ K ↔ ∀ a, e a b = 1) (H : AddSubgroup A) (hMH : M ≤ H) (x : A) (hx : x ∉ H) : ∃ b, (∀ u ∈ H, e u b = 1) ∧ e x b ≠ 1

theorem **finite_quot** {G : Type} [AddCommGroup G] (A L₀ : AddSubgroup G) (N : ℕ) (hN : 0 < N) (tf : Function.Injective fun t : G => N • t) (hA : ∀ t ∈ A, N • t ∈ L₀) (fin : ∀ k : ℕ, 0 < k → ∃ R : Set G, R.Finite ∧ ∀ t ∈ L₀, ∃ r ∈ R, ∃ u ∈ L₀, t = r + k • u) : Finite (A ⧸ (L₀.map (nsmulAddMonoidHom N)).addSubgroupOf A)

theorem **mem_of_double_annihilator** : ∀ {G : Type} [AddCommGroup G] (χ : G → G → ℂ), GlobalTheory.PlaceModel.IsBichar χ → ∀ (X Y L₀ L₀' L : AddSubgroup G), L₀ ≤ X → L₀' ≤ Y → L ≤ X → (∀ u ∈ L₀, ∀ y ∈ L₀', χ u y = 1) → (∀ x ∈ X, (∀ y ∈ L₀', χ x y = 1) → x ∈ L₀) → (∀ y ∈ Y, (∀ u ∈ L₀, χ u y = 1) → y ∈ L₀') → (∀ k : ℕ, 0 < k → ∀ x ∈ X, ∃ x' ∈ X, k • x' = x) → (∀ k : ℕ, 0 < k → ∀ y ∈ Y, ∃ y' ∈ Y, k • y' = y) → (∀ k : ℕ, 0 < k → Function.Injective fun t : G => k • t) → (∀ k : ℕ, 0 < k → ∃ R : Set G, R.Finite ∧ ∀ t ∈ L₀, ∃ r ∈ R, ∃ u ∈ L₀, t = r + k • u) → (∀ k : ℕ, 0 < k → ∃ R : Set G, R.Finite ∧ ∀ t ∈ L₀', ∃ r ∈ R, ∃ u ∈ L₀', t = r + k • u) → ∀ N : ℕ, 0 < N → (∀ u ∈ L₀, N • u ∈ L) → (∀ u ∈ L, N • u ∈ L₀) → ∀ x ∈ X, N • x ∈ L₀ → (∀ y ∈ Y, (∀ u ∈ L, χ u y = 1) → χ x y = 1) → x ∈ L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleBlockTriangularMemIntegralMulLevi

theorem **shape_aX** (hp : 1 ≤ p) (B : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) (h1 : B.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1)))) (h2 : B.BlockTriangular (fun r => OrderDual.toDual (lev p r))) (l : Fin c) (y : n) : B (Sum.inr l) (Sum.inl y) = 0

theorem **shape_aa** (B : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) (h1 : B.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1)))) (h2 : B.BlockTriangular (fun r => OrderDual.toDual (lev p r))) (i j : Fin c) (hij : i ≠ j) : B (Sum.inr i) (Sum.inr j) = 0

theorem **shape_Xa** (B : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) (h1 : B.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1)))) (h2 : B.BlockTriangular (fun r => OrderDual.toDual (lev p r))) (x : n) (i : Fin c) (hi : i.val + 1 ≠ p) : B (Sum.inl x) (Sum.inr i) = 0

theorem **eq_fromBlocks** (hp : 1 ≤ p) (B : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) (h1 : B.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1)))) (h2 : B.BlockTriangular (fun r => OrderDual.toDual (lev p r))) : B = fromBlocks (toBlocks₁₁ B) (toBlocks₁₂ B) 0 (toBlocks₂₂ B)

theorem **diag_block** (B : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) (h1 : B.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1)))) (h2 : B.BlockTriangular (fun r => OrderDual.toDual (lev p r))) : toBlocks₂₂ B = diagonal fun i => B (Sum.inr i) (Sum.inr i)

theorem **blockDiag_mem_KO** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ϖ : kˣ) (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (A : GL (n ⊕ Fin c) k), A.val.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1))) → A.val.BlockTriangular (fun r => OrderDual.toDual (lev p r)) → (A⁻¹).val.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1))) → (A⁻¹).val.BlockTriangular (fun r => OrderDual.toDual (lev p r)) → (∀ i : Fin c, A.val (Sum.inr i) (Sum.inr i) * (ϖ : k) ^ (i.val + 1) ∈ 𝒪[k] ∧ (A⁻¹).val (Sum.inr i) (Sum.inr i) * ((ϖ : k) ^ (i.val + 1))⁻¹ ∈ 𝒪[k]) → (∀ i : Fin c, i.val + 1 = p → ∀ x : n, A.val (Sum.inl x) (Sum.inr i) * (ϖ : k) ^ (i.val + 1) ∈ 𝒪[k]) → ∃ γ : GL n k, A * embX c γ * embA n (GlobalTheory.SplitFlip.diagGL fun i : Fin c => ϖ ^ (i.val + 1)) ∈ KO

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleBlockTriangularMemIntegralMulLeviStatement

theorem **_root_.GlobalTheory.T.doubleBlockTriangular_mem_integral_mul_levi** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ϖ : kˣ) (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (A : GL (n ⊕ Fin c) k), A.val.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1))) → A.val.BlockTriangular (fun r => OrderDual.toDual (lev p r)) → (A⁻¹).val.BlockTriangular (Sum.elim (fun _ : n => 2 * p - 1) (fun i : Fin c => 2 * (i.val + 1))) → (A⁻¹).val.BlockTriangular (fun r => OrderDual.toDual (lev p r)) → (∀ i : Fin c, A.val (Sum.inr i) (Sum.inr i) * (ϖ : k) ^ (i.val + 1) ∈ 𝒪[k] ∧ (A⁻¹).val (Sum.inr i) (Sum.inr i) * ((ϖ : k) ^ (i.val + 1))⁻¹ ∈ 𝒪[k]) → (∀ i : Fin c, i.val + 1 = p → ∀ x : n, A.val (Sum.inl x) (Sum.inr i) * (ϖ : k) ^ (i.val + 1) ∈ 𝒪[k]) → ∃ γ : GL n k, A * embX c γ * embA n (GlobalTheory.SplitFl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftFockSumEqSumIteratedLifts

theorem **comb_sum_mul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) {n : ℕ} (a : Fin n → ℂ) (F : Fin n → WO S T → ℂ) : comb ψ S T c (fun w => ∑ i, a i * F i w) = fun w => ∑ i, a i * comb ψ S T c (F i) w

theorem **comb_pureTensorO** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (sS sT BS BT) (f : WO S T → ℂ) (hf : IsThetaFunO ψ S T f) (c : WO S T →₀ ℂ) (hc : IsFinComb c) (P : PolF D T.rank S.rank) : comb ψ S T c (pureTensorO ψ θ S T sS sT BS BT f P) = pureTensorO ψ θ S T sS sT BS BT (comb ψ S T c f) P

theorem **integrableOn_kernel** {S T : HSp D} (hS : CoCpt S) (v : Vol S) {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hw : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {φ : GA D S → ℂ} (hφ : IsContAut S φ) (h : GA D T) : IntegrableOn (fun g => om (g, h) f 0 * φ g) v.𝓕 v.μ

theorem **biLift_biComb_fockSum** : ∀ (D : D7), BiAct D → CenFock D → LiftSum D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → ∀ (n : ℕ) (P : Fin n → PolF D S.rank A.rank) (P' : Fin n → PolF D A'.rank S.rank) (C : WO A S → WO S A' → ℂ), (∀ u v, C u v = ∑ i, pureTensorO ψ θ A S x.s y.s x.B y.B k.vac (P i) u * pureTensorO ψ θ S A' y.s z.s y.B z.B k'.vac (P' i) v) → ∀ φ, IsContAut A φ → biLift k.om k'.om x.vol y.vol φ (biComb ψ A S A' k.c k'.c C) = ∑ i, k'.lift ψ θ y z (P' i) (k.lift ψ θ x y (P i) φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftFockSumEqSumIteratedLiftsStatement

theorem **_root_.GlobalTheory.T.doubleLift_fockSum_eq_sum_iteratedLifts** : ∀ (D : D7), BiAct D → CenFock D → LiftSum D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → ∀ (n : ℕ) (P : Fin n → PolF D S.rank A.rank) (P' : Fin n → PolF D A'.rank S.rank) (C : WO A S → WO S A' → ℂ), (∀ u v, C u v = ∑ i, pureTensorO ψ θ A S x.s y.s x.B y.B k.vac (P i) u * pureTensorO ψ θ S A' y.s z.s y.B z.B k'.vac (P' i) v) → ∀ φ, IsContAut A φ → biLift k.om k'.om x.vol y.vol φ (biComb ψ A S A' k.c k'.c C) = ∑ i, k'.lift ψ θ y z (P' i) (k.lift ψ θ x y (P i) φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftPeriodRankZeroFubiniCompactQuotient

theorem **flagN_rank_zero** {S S₀ : HSp D} (d : HypDec S S₀ 0) {n : GA D S} (hn : n ∈ flagN d) : n = 1

theorem **per_rank_zero** (B : BesLev D 0) (ψ : AddChar (AF D.host.F) ℂ) (φ : GA D B.A → ℂ) (f : GA D B.V → ℂ) : B.per ψ φ f = ∫ h in B.vA.𝓕, φ h * f (embA B.d B.o h) ∂B.vA.μ

theorem **swap_bdd** {X Y : Type} [TopologicalSpace X] [TopologicalSpace Y] [MeasurableSpace X] [MeasurableSpace Y] [BorelSpace X] [BorelSpace Y] [SecondCountableTopology X] [SecondCountableTopology Y] (μ : Measure X) (ν : Measure Y) (s : Set X) (t : Set Y) (hs : μ s < ⊤) (ht : ν t < ⊤) (f : X → Y → ℂ) (hf : Continuous (Function.uncurry f)) (C : ℝ) (hC : ∀ x y, ‖f x y‖ ≤ C) : ∫ x in s, ∫ y in t, f x y ∂ν ∂μ = ∫ y in t, ∫ x in s, f x y ∂μ ∂ν

theorem **om_bdd** {S T A T₀ : HSp D} {χ χ' : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χ χ' om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hA : CoCpt A) (hT : CoCpt T₀) (a : GA D A →* GA D S) (b : GA D T₀ →* GA D T) (ha : Continuous a) (hb : Continuous b) (ha' : ∀ γ ∈ GQ A, a γ ∈ GQ S) (hb' : ∀ γ ∈ GQ T₀, b γ ∈ GQ T) : Continuous (fun p : GA D A × GA D T₀ => om (a p.1, b p.2) f 0) ∧ ∃ C : ℝ, ∀ x y, ‖om (a x, b y) f 0‖ ≤ C

theorem **bound_mul** {x y : ℂ} {C C' : ℝ} (hx : ‖x‖ ≤ C) (hy : ‖y‖ ≤ C') : ‖x * y‖ ≤ C * C'

theorem **triple** {X Y Z : Type} [TopologicalSpace X] [TopologicalSpace Y] [TopologicalSpace Z] [MeasurableSpace X] [MeasurableSpace Y] [MeasurableSpace Z] [BorelSpace X] [BorelSpace Y] [BorelSpace Z] [SecondCountableTopology X] [SecondCountableTopology Y] [SecondCountableTopology Z] (μ : Measure X) (ν : Measure Y) (κ : Measure Z) (s : Set X) (u : Set Y) (w : Set Z) (hs : μ s < ⊤) (hu : ν u < ⊤) (hw : κ w < ⊤) (φ : X → Y → ℂ) (F : Z → ℂ) (a : X → Y → ℂ) (b : Y → Z → ℂ) (hφ : Continuous φ) (hF : Continuous F) (ha : Continuous fun p : X × Y => a p.1 p.2) (hb : Continuous fun p : Y × Z => b p.1 p.2) {Cφ CF Ca Cb : ℝ} (hCφ : ∀ x, ‖φ x‖ ≤ Cφ) (hCF : ∀ z, ‖F z‖ ≤ CF) (hCa : ∀ x y, ‖a x y‖ ≤ Ca) (hCb : ∀ y z, ‖b y z‖ ≤ Cb) : ∫ x in s, φ x * ∫ y in u, a x y * ∫ z in w, b y z * F z ∂κ ∂ν ∂μ = ∫ z in w, F z * ∫ y in u, b y z * ∫ x in s, φ x * a x y ∂μ ∂ν ∂κ

theorem **lastLaw** : ∀ D : D7, LastLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftPeriodRankZeroFubiniCompactQuotientStatement

theorem **_root_.GlobalTheory.T.doubleLiftPeriod_rankZero_fubini_compactQuotient** : ∀ D : D7, LastLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftReductionToFockVectorsUniform

theorem **exists_central_average_uniform** : ∀ (D : D7), BiAct D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k₀ : KerDat A S) (k₀' : KerDat S A'), k₀.Is ψ θ x y → k₀'.Is ψ θ y z → ∀ (χ : GA D A → ℂ) (C : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L y.L z.L C → IsBiSmooth C → (∃ φ₀ : GA D A → ℂ, IsContAut A φ₀ ∧ HasCen A χ φ₀ ∧ φ₀ ≠ 0) → ∃ (C₁ : WO A S → WO S A' → ℂ) (χ₁ : GA D A → ℂ), IsBiVac ψ A S A' x.L y.L z.L C₁ ∧ IsBiSmooth C₁ ∧ (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k₀.om (a, 1) (fun u => C₁ u v) = χ₁ a • fun u => C₁ u v) ∧ ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → (∀ a ∈ GinfC A, ∀ b ∈ GinfC S, ∀ f, IsThetaFunO ψ A S f → k.om (a, b) f = k₀.om (a, b) f) → ∀ φ, IsContAut A φ → HasCen A χ φ → biLift k.om k'.om x.vol y.vol φ (biComb ψ A S A' k.c k'.c C) = biLif

theorem **fockRed_uniform** : ∀ D : D7, BiAct D → CenFock D → LiftSum D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k₀ : KerDat A S) (k₀' : KerDat S A'), k₀.Is ψ θ x y → k₀'.Is ψ θ y z → ∀ (χ : GA D A → ℂ) (C : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L y.L z.L C → IsBiSmooth C → ∃ (n : ℕ) (P : Fin n → PolF D S.rank A.rank) (P' : Fin n → PolF D A'.rank S.rank), ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → k.vac = k₀.vac → k'.vac = k₀'.vac → (∀ a ∈ GinfC A, ∀ b ∈ GinfC S, ∀ f, IsThetaFunO ψ A S f → k.om (a, b) f = k₀.om (a, b) f) → (∀ a ∈ GinfC S, ∀ b ∈ GinfC A', ∀ f, IsThetaFunO ψ S A' f → k'.om (a, b) f = k₀'.om (a, b) f) → ∀ φ, IsContAut A φ → HasCen A χ φ → biLift k.om k'.om x.vol y.vol φ (biComb ψ A S A' k.c k'.c C) = ∑ i, k'.lift ψ θ y z (P' i) (k.lift ψ θ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftReductionToFockVectorsUniformStatement

theorem **_root_.GlobalTheory.T.doubleLift_reduction_to_fockVectors_uniform** : ∀ D : D7, BiAct D → CenFock D → LiftSum D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k₀ : KerDat A S) (k₀' : KerDat S A'), k₀.Is ψ θ x y → k₀'.Is ψ θ y z → ∀ (χ : GA D A → ℂ) (C : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L y.L z.L C → IsBiSmooth C → ∃ (n : ℕ) (P : Fin n → PolF D S.rank A.rank) (P' : Fin n → PolF D A'.rank S.rank), ∀ (k : KerDat A S) (k' : KerDat S A'), k.Is ψ θ x y → k'.Is ψ θ y z → k.vac = k₀.vac → k'.vac = k₀'.vac → (∀ a ∈ GinfC A, ∀ b ∈ GinfC S, ∀ f, IsThetaFunO ψ A S f → k.om (a, b) f = k₀.om (a, b) f) → (∀ a ∈ GinfC S, ∀ b ∈ GinfC A', ∀ f, IsThetaFunO ψ S A' f → k'.om (a, b) f = k₀'.om (a, b) f) → ∀ φ, IsContAut A φ → HasCen A χ φ → biLift k.om k'.om x.vol y.vol φ (biComb ψ A S A' k.c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftValueEquivariantSymmetricUnipotent

theorem **valK_nEquiv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ Ξ, ∀ g ∈ N Q.k Q.n₂ (Q.c + 1) (σ + 1), ∀ f ∈ GlobalTheory.TriStep.space Q, GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ Ξ (rTrans (Q.gl₂ g) f) = (Q.ψk (simpleSum g) : ℂ) * GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ Ξ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleLiftValueEquivariantSymmetricUnipotentStatement

theorem **_root_.GlobalTheory.T.doubleLiftValue_equivariant_symmetricUnipotent** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ Ξ, ∀ g ∈ N Q.k Q.n₂ (Q.c + 1) (σ + 1), ∀ f ∈ GlobalTheory.TriStep.space Q, GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ Ξ (rTrans (Q.gl₂ g) f) = (Q.ψk (simpleSum g) : ℂ) * GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ Ξ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleMixedModelBaseValueEquivariant

theorem **heisO_smul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w₀ : WO S T) (c : ℂ) (f : WO S T → ℂ) : heisO ψ S T w₀ (c • f) = c • heisO ψ S T w₀ f

theorem **det_UX** {ρ : ℕ} (γ : UX D ρ) : Matrix.GeneralLinearGroup.det γ.1 = 1

theorem **halfMod_UX** {D : D7} {ρ : ℕ} (n : ℕ) (γ : UX D ρ) : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.halfMod n γ.1 = 1

theorem **GlobalTheory.Unf.evalAt_xZero_yZero_rad** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → ∀ (u : UX D ρ) h₀, ∀ n ∈ radX J.d, ∀ h', evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (embA B.d B.o h₀, n * J.d.emb0 h') Ψ) (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ)) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ))))

theorem **GlobalTheory.Unf.evalAt_xZero_yZero_levi** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → ∀ (u γ : UX D ρ), evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (1, J.d.levi γ.1) Ψ) (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ)) = evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (xZero J.d · (γ⁻¹ * u).1.1) (yZero B.V₀ ρ (vecB B.o B.ζ))

theorem **GlobalTheory.Unf.evalAt_xZero_yZero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → (∀ (u : UX D ρ) h₀, ∀ n ∈ radX J.d, ∀ h', evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (embA B.d B.o h₀, n * J.d.emb0 h') Ψ) (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ)) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ))))) ∧ ∀ (u γ : UX D ρ), evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (1, J.d.levi γ.1) Ψ) (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ)) = evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (xZero J.d · (γ⁻¹ * u).1.1) (yZero B.V₀ ρ (vecB B.o B.ζ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleMixedModelBaseValueEquivariantStatement

theorem **_root_.GlobalTheory.T.doubleMixedModel_baseValue_equivariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → (∀ (u : UX D ρ) h₀, ∀ n ∈ radX J.d, ∀ h', evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (embA B.d B.o h₀, n * J.d.emb0 h') Ψ) (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ)) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ))))) ∧ ∀ (u γ : UX D ρ), evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (1, J.d.levi γ.1) Ψ) (xZero J.d · u.1.1) (yZero B.V₀ ρ (vecB B.o B.ζ)) = evalAt ψ B.d J.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (xZero J.d · (γ⁻¹ * u).1.1) (yZero B.V₀ ρ (vecB B.o B.ζ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleMixedModelValueAtZeroEquivariant

theorem **det_of_mem_UX** {ρ : ℕ} {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : Matrix.GeneralLinearGroup.det u = 1

theorem **halfMod_of_mem_UX** {ρ : ℕ} (m : ℕ) {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.halfMod m u = 1

theorem **mixT_zero_flagN** {S T T₀ : HSp D} {σ : ℕ} (d : HypDec T T₀ σ) (ψ : AddChar (AF D.host.F) ℂ) (χS : (AE D)ˣ →* Circle) (om : Om S T) (om₀ : Om S T₀) (μ : Measure (Mx D σ S.rank)) (ℱ : Set (Mx D σ S.rank)) (hpres : ∀ p f, IsThetaFunO ψ S T f → IsThetaFunO ψ S T (om p f)) (hF : ∀ f, IsThetaFunO ψ S T f → GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.FormulasT d ψ χS om om₀ μ ℱ f) (f : WO S T → ℂ) (hf : IsThetaFunO ψ S T f) (n : GA D T) (hn : n ∈ flagN d) : d.mixT S ψ μ ℱ (om (1, n) f) 0 = d.mixT S ψ μ ℱ f 0

theorem **evalAt_zero_equivariant** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (Ψ : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V Ψ → ∀ (g : GA D J.W₀), ∀ n ∈ flagN B.d, ∀ (h : GA D B.A) (x : Mx D B.V.rank (σ + 1)), evalAt ψ J.d B.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (J.d.emb0 g, n * embA B.d B.o h) Ψ) x 0 = s.om₀ (g, B.o.emb (h, 1)) (evalAt ψ J.d B.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (mt B.V (n * embA B.d B.o h)⁻¹ · x) 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleMixedModelValueAtZeroEquivariantStatement

theorem **_root_.GlobalTheory.T.doubleMixedModel_valueAtZero_equivariant** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (Ψ : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V Ψ → ∀ (g : GA D J.W₀), ∀ n ∈ flagN B.d, ∀ (h : GA D B.A) (x : Mx D B.V.rank (σ + 1)), evalAt ψ J.d B.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ (s.om (J.d.emb0 g, n * embA B.d B.o h) Ψ) x 0 = s.om₀ (g, B.o.emb (h, 1)) (evalAt ψ J.d B.d s.vS.μ s.vS.ℱ s.vT.μ s.vT.ℱ Ψ (mt B.V (n * embA B.d B.o h)⁻¹ · x) 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepAdelicIntegralFiniteTwistPlaces

theorem **valQr_succ** (D : D7) (r : ℕ) (IH : ValQr D r) : ValQr D (r + 1)

theorem **valQ** : ∀ D : D7, ValQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepAdelicIntegralFiniteTwistPlacesStatement

theorem **_root_.GlobalTheory.T.doubleStep_adelicIntegral_finiteTwistPlaces** : ∀ D : D7, ValQ D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepEqTwoThetaColumnLifts

theorem **bdd_of_contAut** {V : HSp D} (hV : CoCpt V) {φ : GA D V → ℂ} (hφ : IsContAut V φ) : ∃ C : ℝ, ∀ g, ‖φ g‖ ≤ C

theorem **bdd_om_core** {ψ : AddChar (AF D.host.F) ℂ} {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {Ξ : WO B.V₀ J.W₀ → ℂ} (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (h₁ : GA D J.W₀) : ∃ C : ℝ, ∀ h₀ x, ‖s.om₀ (B.o.emb (h₀, 1), h₁) Ξ x‖ ≤ C

theorem **continuous_kappa** {ρ : ℕ} {B : BesLev D ρ} {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (S : HSp D) : Continuous (l.κ S)

theorem **continuous_core** {ψ : AddChar (AF D.host.F) ℂ} {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') {Ξ : WO B.V₀ J.W₀ → ℂ} (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (h₁ : GA D J.W₀) : Continuous fun p : WO J.W₀ L' × GA D B.A => s.om₀ (B.o.emb (p.2, 1), h₁) Ξ ((B.o.resS J.W₀).symm (0, l.κ J.W₀ p.1))

theorem **exchange_gen** {α β : Type} [TopologicalSpace α] [MeasurableSpace α] [OpensMeasurableSpace α] [TopologicalSpace β] [MeasurableSpace β] [OpensMeasurableSpace β] [SecondCountableTopologyEither α β] (μ : Measure α) (ν : Measure β) [IsFiniteMeasure μ] [IsFiniteMeasure ν] (X₂ : α → ℂ) (φ : β → ℂ) (X₁ : β → α → ℂ) (hc : Continuous fun p : α × β => X₂ p.1 * (φ p.2 * X₁ p.2 p.1)) (C : ℝ) (hb : ∀ w h, ‖X₂ w * (φ h * X₁ h w)‖ ≤ C) : ∫ w, X₂ w * ∫ h, φ h * X₁ h w ∂ν ∂μ = ∫ h, φ h * ∫ w, X₂ w * X₁ h w ∂μ ∂ν

theorem **exchange** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) {φ : GA D B.A → ℂ} {Ξ₁ : WO B.V₀ J.W₀ → ℂ} {Ξ₂ : WO J.W₀ B'.V₀ → ℂ} (hφ : IsContAut B.A φ) (hΞ₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) (hΞ₂ : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂) (h₁ : GA D J.W₀) (h₂ : GA D B'.A) : ∫ w in vL.𝒻, s₂.om₀ (h₁, B'.o.emb (h₂, 1)) Ξ₂ ((B'.o.resT J.W₀).symm (0, w)) * ∫ h₀ in B.vA.𝒻, φ h₀ * s₁.om₀ (B.o.emb (h₀, 1), h₁) Ξ₁ ((B.o.resS J.W₀).symm (0, l.κ J.W₀ w)) ∂B.vA.μ ∂vL.μ = ∫ h₀ in B.vA.𝒻, φ h₀ * ∫ w in vL.𝒻, s₂.om₀ (h₁, B'.o.emb (h₂, 1)) Ξ₂ ((B'.o.resT J.W₀).symm (0, w)) * s₁.om₀ (B.o.emb (h₀, 1), h₁) Ξ₁ ((B.o.resS J.W₀).symm (0, l.κ J.W₀ w)) ∂vL.μ ∂B.vA.μ

theorem **contrLaw** (D : D7) : ContrLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepEqTwoThetaColumnLiftsStatement

theorem **_root_.GlobalTheory.T.doubleStep_eq_two_thetaColumnLifts** : ∀ D : D7, ContrLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandBesselPeriodInvariantProof

theorem **mul_star_psi** (hψ : IsStdChar D ψ (-1)) (x : AF D.host.F) : ψ x * star (ψ x) = 1

theorem **star_psi_neg** (hψ : IsStdChar D ψ (-1)) (x : AF D.host.F) : star (ψ (-x)) = ψ x

theorem **om_mul** {S T : HSp D} (om : Om S T) (a b : GA D S) (a' b' : GA D T) (f : WO S T → ℂ) : om (a * b, a' * b') f = om (a, a') (om (b, b') f)

theorem **mixT_zero_heis** {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (v : VolM D ρ S.rank) {g : WO S T → ℂ} (hg : IsThetaFunO ψ S T g) (c : ℂ) (η : Mx D ρ S.rank) (w : WO S T₀) : d.mixT S ψ v.μ v.𝒻 (heisO ψ S T (d.incX S η + d.inc0 S w) (c • g)) 0 = c • heisO ψ S T₀ w (d.mixT S ψ v.μ v.𝒻 g 0)

theorem **fourierBessel_centralTerm_const_on_orbit** {σ : ℕ} (B : BesLev D σ) {S S₀ : HSp D} (d : HypDec S S₀ (σ + 1)) (c : skewA D (σ + 1)) (v : UX D (σ + 1)) : d.qS B.V c (x0 B · v.1.1) = d.qS B.L c (zl B)

theorem **flatFB_source** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) {Ψ : WO J.W B.V → ℂ} (hΨ : IsThetaFunO ψ J.W B.V Ψ) {n : GA D J.W} (hn : n ∈ flagN J.d) : flatFB s (s.om (n, 1) Ψ) = (psiFlag ψ J.d n * ψ (cenS J.d B.L (zl B) n)) • heisO ψ J.W₀ B.V₀ ((B.o.resT J.W₀).symm (0, ptS J.d B.L (zl B) n)) (flatFB s Ψ)

theorem **handBF_heis** {ρ σ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} {B' : BesLev D σ} (s : StepBF ψ B J) (l : Link B B'.L B'.ζ) (φ : GA D B.A → ℂ) {Ξ : WO B.V₀ J.W₀ → ℂ} (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (a : ℂ) (p : WO J.W₀ B'.L) (h h₀ : GA D J.W₀) (w : WO J.W₀ B'.L) : handBF s l φ (a • heisO ψ B.V₀ J.W₀ ((B.o.resS J.W₀).symm (0, l.κ J.W₀ p)) (s.om₀ (1, h) Ξ)) h₀ w = star a * ψ (symplOH J.W₀ B'.L w (p · mt J.W₀ h₀⁻¹)) * handBF s l φ Ξ (h₀ * h) (w + p · mt J.W₀ h₀⁻¹)

theorem **pairKer_heis** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) {Y Y' : GA D J.W₀ → WO J.W₀ B.L → ℂ} (hY : IsQuasiFam ψ J.W₀ B.L Y) {Ξ : WO J.W₀ B.V₀ → ℂ} (hΞ : IsThetaFunO ψ J.W₀ B.V₀ Ξ) (a b : ℂ) (hab : a * b = 1) (p : WO J.W₀ B.L) (h : GA D J.W₀) (hY' : ∀ h₀ w, Y' h₀ w = star a * ψ (symplOH J.W₀ B.L w (p · mt J.W₀ h₀⁻¹)) * Y (h₀ * h) (w + p · mt J.W₀ h₀⁻¹)) (h₀ : GA D J.W₀) (x : GA D B.A) : pairKer s vL Y' (b • heisO ψ J.W₀ B.V₀ ((B.o.resT J.W₀).symm (0, p)) (s.om₀ (h, 1) Ξ)) h₀ x = pairKer s vL Y Ξ (h₀ * h) x

theorem **handFB_smul_om** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (Ξ : WO J.W₀ B.V₀ → ℂ) (c : ℂ) (h : GA D B.A) : handFB s vL Y (c • s.om₀ (1, B.o.emb (h, 1)) Ξ) = c • rTrans h (handFB s vL Y Ξ)

theorem **per_flag** {ρ : ℕ} (B : BesLev D ρ) (hψ : IsStdChar D ψ (-1)) (φ : GA D B.A → ℂ) {f : GA D B.V → ℂ} (hf : ∀ γ ∈ GQ B.V, ∀ g, f (γ * g) = f g) {n : GA D B.V} (hn : n ∈ flagN B.d) : B.per ψ (star (psiBes ψ B.d 1 (vecB B.o B.ζ) n) • φ) (rTrans n f) = B.per ψ φ f

theorem **handInv** : ∀ D : D7, GlobalTheory.FinPl.HandInv D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandBesselPeriodInvariantProofStatement

theorem **_root_.GlobalTheory.T.doubleStepIntegrandBesselPeriodInvariant_proof** : ∀ D : D7, GlobalTheory.FinPl.HandInv D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandLeftInvariantBesselSecond

theorem **contAut_translate** {V : HSp D} {f : GA D V → ℂ} (h : GA D V) (hf : IsContAut V f) : IsContAut V (rTrans h f)

theorem **contAut_of_mem_multT** {V : HSp D} {S : D.what₂.Tau → SPar} {θ : InfFam D} {M : DSModel V S} {J : Set (GA D V)} {χ : GA D V → ℂ} {G : SmT M →ₗ[ℂ] (GA D V → ℂ)} (hG : G ∈ MultT θ M J χ) (x : SmT M) : IsContAut V (G x)

theorem **theta_comb** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {P : Pl D → Prop} {e : WO S T →₀ ℂ} (he : CombAt P e) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (comb ψ S T e f)

theorem **integ_left_second** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandLeftInvariantBesselSecondStatement

theorem **_root_.GlobalTheory.T.doubleStep_integrand_leftInvariant_bessel_second** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandLeftInvariantFourierJacobiFir

theorem **vac_empty_comb** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {P : Pl D → Prop} {e : WO S T →₀ ℂ} (he : CombAt P e) {LS : Lat S} {LT : Lat T} {f : WO S T → ℂ} (hf : Vac ψ S T LS LT (fun _ => False) f) : Vac ψ S T LS LT (fun _ => False) (comb ψ S T e f)

theorem **vac_empty** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT (fun _ => True) f) : Vac ψ S T LS LT (fun _ => False) f

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandLeftInvariantFourierJacobiFirStatement

theorem **_root_.GlobalTheory.T.doubleStep_integrand_leftInvariant_fourierJacobi_first** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantFirstFixedLevel

theorem **vac_of_all** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : GlobalTheory.FinPl.Lat S} {LT : GlobalTheory.FinPl.Lat T} (P : Pl D → Prop) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT (fun _ => True) f) : Vac ψ S T LS LT P f

theorem **hand_mul_right** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (Φ₁ : WO B.V J.W → ℂ) (Φ₂ : WO J.W B'.V → ℂ) (φ : GA D B.A → ℂ) (k : GA D J.W) (a : ℂ) (ha : a ≠ 0) (h₁ : s₁.om (1, k) Φ₁ = a⁻¹ • Φ₁) (h₂ : s₂.om (k, 1) Φ₂ = a • Φ₂) (g : GA D B.V) (q₁ : GA D J.W) (q₂ : GA D B'.V) : Chain.hand s₁ s₂ l vL Φ₁ Φ₂ φ g (q₁ ∗ k) q₂ = Chain.hand s₁ s₂ l vL Φ₁ Φ₂ φ g q₁ q₂

theorem **integ_lev** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantFirstFixedLevelStatement

theorem **_root_.GlobalTheory.T.doubleStepIntegrand_rightInvariant_first_fixedLevel** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantFirstOffLevel

theorem **vac_off** {D : D7} {c : Comm D} {v : VCom c} {t : Tup c} {w : VTup t} (hc : c.Is) (hv : v.Is) (hf : VForm v w) (j : ℕ) (hj : j + 1 < D.R.m) (Ψ : WO (c.V j) (c.V (j + 1)) → ℂ) (hΨ : Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (fun _ => True) Ψ) : Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (offP v w) (comb c.ψ _ _ (v.eG j) (comb c.ψ _ _ (w.eQ j) Ψ))

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantFirstOffLevelStatement

theorem **_root_.GlobalTheory.T.doubleStepIntegrand_rightInvariant_first_offLevel** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantSecondFixedLeve

theorem **comm_inf_fin** {D : D7} (V : HSp D) {a b : GA D V} (ha : a ∈ GinfC V) (hb : b ∈ GfC V) : a ∗ b = b ∗ a

theorem **per_hand_mul** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (Φ₁ : WO B.V J.W → ℂ) (Φ₂ : WO J.W B'.V → ℂ) (φ : GA D B.A → ℂ) (g : GA D B.V) (q₁ : GA D J.W) (q₂ k : GA D B'.V) (s : ℂ) (hs : s ≠ 0) (hom : s₂.om (1, k) Φ₂ = s⁻¹ • Φ₂) (f f' : GA D B'.V → ℂ) (hf : f' = s • f) : B'.per ψ (Chain.hand s₁ s₂ l vL Φ₁ Φ₂ φ g q₁ (q₂ ∗ k)) f' = B'.per ψ (Chain.hand s₁ s₂ l vL Φ₁ Φ₂ φ g q₁ q₂) f

theorem **rTrans_eigen** {G : Type} [Group G] (a b k q : G) (F : G → ℂ) (s : ℂ) (hcm : a ∗ k = k ∗ a) (h : rTrans k (rTrans b F) = s • rTrans b F) : rTrans (q ∗ k) (rTrans (a ∗ b) F) = s • rTrans q (rTrans (a ∗ b) F)

theorem **vac_comb** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (LS : Lat S) (LT : Lat T) (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (P Q : Pl D → Prop) (hPQ : ∀ p, Q p → ¬ P p) (e : WO S T →₀ ℂ) (he : CombAt Q e) (f : WO S T → ℂ) (hf : Vac ψ S T LS LT (fun _ => True) f) : Vac ψ S T LS LT P (comb ψ S T e f)

theorem **integ_mul_level** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantSecondFixedLeveStatement

theorem **_root_.GlobalTheory.T.doubleStepIntegrand_rightInvariant_second_fixedLevel** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantSecondOffLevel

theorem **vac_off** {D : D7} {c : Comm D} (v : VCom c) {t : Tup c} (w : VTup t) (hc : c.Is) (hv : v.Is) (hw : VForm v w) (i : ℕ) (hi : i + 1 < D.R.m) (Ψ : WO (c.V i) (c.V (i + 1)) → ℂ) (hΨ : Vac c.ψ _ _ (v.N i) (v.N (i + 1)) (fun _ => True) Ψ) : Vac c.ψ _ _ (v.N i) (v.N (i + 1)) (offP v w) (comb c.ψ _ _ (v.eG i) (comb c.ψ _ _ (w.eQ i) Ψ))

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepIntegrandRightInvariantSecondOffLevelStatement

theorem **_root_.GlobalTheory.T.doubleStepIntegrand_rightInvariant_second_offLevel** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepLocalWeilActionLineCharactersCancel

theorem **isComp_mul** (p : Place D k) {χ χ' : (AE D)ˣ →* Circle} {μ μ' : kˣ →* ℂˣ} (h : p.IsComp χ μ) (h' : p.IsComp χ' μ') : p.IsComp (χ ∗ χ') (μ ∗ μ')

theorem **isComp_inv** (p : Place D k) {χ : (AE D)ˣ →* Circle} {μ : kˣ →* ℂˣ} (h : p.IsComp χ μ) : p.IsComp χ⁻¹ μ⁻¹

theorem **om_mul** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ ν : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (Φ : Matrix b a k → ℂ) : om (μ ∗ ν) g h Φ = (((ν (Matrix.GeneralLinearGroup.det g) ∗ (ν (Matrix.GeneralLinearGroup.det h))⁻¹ : ℂˣ)) : ℂ) • om μ g h Φ

theorem **val_smul_inv** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} (S : Dbl2 ψ σ) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (A : WO S.B.V S.J.W → ℂ) (B : WO S.J.W S.B'.V → ℂ) (φ : GA D S.B.A → ℂ) (f : GA D S.B'.V → ℂ) (g : GA D S.B.V) (h₁ : GA D S.J.W) (h₂ : GA D S.B'.V) (z : ℂˣ) : S.val c₁ c₂ ((z : ℂ) • A) (((z⁻¹ : ℂˣ) : ℂ) • B) φ f g h₁ h₂ = S.val c₁ c₂ A B φ f g h₁ h₂

theorem **scal_one** {G : Type} [Group G] (μ ζ : G →* ℂˣ) (a x : G) : ζ x = ((μ⁻¹ ∗ ζ) ∗ μ) a ∗ (ζ⁻¹ a ∗ (ζ⁻¹ x)⁻¹)

theorem **scal_two** {G : Type} [Group G] (μ ζ : G →* ℂˣ) (x : G) : (ζ x)⁻¹ = (μ⁻¹ * (μ * ζ⁻¹)) x

theorem **val_om_loc** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (T₁ : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.B.V S.J.W → ℂ)) (T₂ : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.J.W S.B'.V → ℂ)), (ofSplit Q.q).IsModel ψ Q.F.G₀ Q.F.G₁ T₁ → (ofSplit Q.q).IsModel ψ Q.F.G₁ Q.F.G₂ T₂ → ∀ (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (φ : GA D S.B.A → ℂ) (F : GA D S.B'.V → ℂ) (a : GL (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) (x : GL (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) (y : GL (Q.n₂ ⊕ Fin (Q.c + 1)) Q.k) (Φ₁ Φ₁' : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k)) (Φ₂ Φ₂' : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k)), Φ₁'.1 = om Q.μ₁ a x Φ₁.1 → Φ₂'.1 = om Q.μ₂ x y Φ₂.1 → S.val c₁ c₂ (S.s₁.om (Q.gl₀ a, Q.gl₁ x) (T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepLocalWeilActionLineCharactersCancelStatement

theorem **_root_.GlobalTheory.T.doubleStep_local_weil_action_line_characters_cancel** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (T₁ : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.B.V S.J.W → ℂ)) (T₂ : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.J.W S.B'.V → ℂ)), (ofSplit Q.q).IsModel ψ Q.F.G₀ Q.F.G₁ T₁ → (ofSplit Q.q).IsModel ψ Q.F.G₁ Q.F.G₂ T₂ → ∀ (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (φ : GA D S.B.A → ℂ) (F : GA D S.B'.V → ℂ) (a : GL (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) (x : GL (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) (y : GL (Q.n₂ ⊕ Fin (Q.c + 1)) Q.k) (Φ₁ Φ₁' : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k)) (Φ₂ Φ₂' : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k)), Φ₁'.1 = om Q.μ₁ a x Φ₁.1 → Φ₂'.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepSmallPairsValueAllGluedSplitPlacesProo

theorem **over_self** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {p : SplitPt D k} {m m' : Type} [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] {α : Frame p S m'} {β : Frame p T m} {g f : WO S T → ℂ} {Φ : Matrix m m' k → ℂ} (h : Over ψ α β LS LT g Φ f) (hU : p.UnitAA) (ΛS : Submodule 𝒪[k] (m' → k)) (ΛT : Submodule 𝒪[k] (m → k)) (hLS : α.IsLat LS ΛS) (hLT : β.IsLat LT ΛT) : Over ψ α β LS LT g (latInd ΛS ΛT) g

theorem **overAll_nil** {ι : Type} {p : ι → SplitPt D k} {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {α : ∀ i, Frame (p i) S n'} {β : ∀ i, Frame (p i) T n} {LS : Lat S} {LT : Lat T} {Ψ : ι → Matrix n n' k → ℂ} {f f' : WO S T → ℂ} : OverAll ψ α β LS LT Ψ [] f f' ↔ f' = f

theorem **overAll_cons** {ι : Type} {p : ι → SplitPt D k} {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {α : ∀ i, Frame (p i) S n'} {β : ∀ i, Frame (p i) T n} {LS : Lat S} {LT : Lat T} {Ψ : ι → Matrix n n' k → ℂ} {i : ι} {l : List ι} {f f' : WO S T → ℂ} : OverAll ψ α β LS LT Ψ (i :: l) f f' ↔ ∃ g, OverAll ψ α β LS LT Ψ l f g ∧ Over ψ (α i) (β i) LS LT g (Ψ i) f'

theorem **GlobalTheory.FinPl.vertexAll** : ∀ D : D7, GlobalTheory.FinPl.VertexAll D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepSmallPairsValueAllGluedSplitPlacesProoStatement

theorem **_root_.GlobalTheory.T.doubleStepSmallPairsValueAllGluedSplitPlaces_proof** : ∀ D : D7, GlobalTheory.FinPl.VertexAll D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepValueEqTrilinearTensorValue

theorem **val_eq_tensorVal** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (T₁ : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.B.V S.J.W → ℂ)) (T₂ : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.J.W S.B'.V → ℂ)), (ofSplit Q.q).IsModel ψ Q.F.G₀ Q.F.G₁ T₁ → (ofSplit Q.q).IsModel ψ Q.F.G₁ Q.F.G₂ T₂ → ∀ (U₁ : SB (Matrix Q.n₁ (Q.n₀ ⊕ Unit) Q.k) →ₗ[ℂ] (WO S.B.V₀ S.J.W₀ → ℂ)) (U₂ : SB (Matrix (Q.n₂ ⊕ Unit) Q.n₁ Q.k) →ₗ[ℂ] (WO S.J.W₀ S.B'.V₀ → ℂ)), (∀ Φ Ψ, Ψ.1 = GlobalTheory.LocFlat.flCol Q.ψk Q.ν (σ + 1 + 1) Φ.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleStepValueEqTrilinearTensorValueStatement

theorem **_root_.GlobalTheory.T.doubleStepValue_eq_trilinearTensorValue** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (T₁ : SB (Matrix (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) (Q.n₀ ⊕ Fin (Q.c + 1 + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.B.V S.J.W → ℂ)) (T₂ : SB (Matrix (Q.n₂ ⊕ Fin (Q.c + 1)) (Q.n₁ ⊕ Fin (Q.c + 1 + 1)) Q.k) →ₗ[ℂ] (WO S.J.W S.B'.V → ℂ)), (ofSplit Q.q).IsModel ψ Q.F.G₀ Q.F.G₁ T₁ → (ofSplit Q.q).IsModel ψ Q.F.G₁ Q.F.G₂ T₂ → ∀ (U₁ : SB (Matrix Q.n₁ (Q.n₀ ⊕ Unit) Q.k) →ₗ[ℂ] (WO S.B.V₀ S.J.W₀ → ℂ)) (U₂ : SB (Matrix (Q.n₂ ⊕ Unit) Q.n₁ Q.k) →ₗ[ℂ] (WO S.J.W₀ S.B'.V₀ → ℂ)), (∀ Φ Ψ, Ψ.1 = Globa

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleSumWeightedBound

theorem **realSize_nsmul** (θ : InfFam D) {m k : ℕ} (n : ℕ) (x : Mx D m k) : realSize θ (n • x) = n * realSize θ x

theorem **pointwise** (θ : InfFam D) {a b c e : ℕ} (K : Set (Mx D a b)) (K' : Set (Mx D c e)) (N N' n : ℕ) (hn : 0 < n) (B : ℝ≥0∞) (F : ratMx D a b → ratMx D c e → ℝ≥0∞) (hF : ∀ x y, F x y ≠ 0 → InBox K (n • (x : Mx D a b)) ∧ InBox K' (n • (y : Mx D c e))) (h1 : ∀ x y, F x y * ENNReal.ofReal ((1 + realSize θ (x : Mx D a b)) ^ (N + N')) ≤ B) (h2 : ∀ x y, F x y * ENNReal.ofReal ((1 + realSize θ (y : Mx D c e)) ^ (N + N')) ≤ B) (x : ratMx D a b) (y : ratMx D c e) : F x y ≤ (n : ℝ≥0∞) ^ (N + N') * B * (ENNReal.ofReal (maj θ K N (n • (x : Mx D a b))) * ENNReal.ofReal (maj θ K' N' (n • (y : Mx D c e))))

theorem **GlobalTheory.Unf.doubleSum_le_maj** : ∀ (D : D7) (θ : InfFam D) (a b c e : ℕ) (K : Set (Mx D a b)) (K' : Set (Mx D c e)) (N N' n : ℕ), 0 < n → ∀ (B : ℝ≥0∞) (F : ratMx D a b → ratMx D c e → ℝ≥0∞), (∀ x y, F x y ≠ 0 → InBox K (n • (x : Mx D a b)) ∧ InBox K' (n • (y : Mx D c e))) → (∀ x y, F x y * ENNReal.ofReal ((1 + realSize θ (x : Mx D a b)) ^ (N + N')) ≤ B) → (∀ x y, F x y * ENNReal.ofReal ((1 + realSize θ (y : Mx D c e)) ^ (N + N')) ≤ B) → ∑' x, ∑' y, F x y ≤ (n : ℝ≥0∞) ^ (N + N') * B * ((∑' x : ratMx D a b, ENNReal.ofReal (maj θ K N x)) * ∑' y : ratMx D c e, ENNReal.ofReal (maj θ K' N' y))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleSumWeightedBoundStatement

theorem **_root_.GlobalTheory.T.double_sum_weighted_bound** : ∀ (D : D7) (θ : InfFam D) (a b c e : ℕ) (K : Set (Mx D a b)) (K' : Set (Mx D c e)) (N N' n : ℕ), 0 < n → ∀ (B : ℝ≥0∞) (F : ratMx D a b → ratMx D c e → ℝ≥0∞), (∀ x y, F x y ≠ 0 → InBox K (n • (x : Mx D a b)) ∧ InBox K' (n • (y : Mx D c e))) → (∀ x y, F x y * ENNReal.ofReal ((1 + realSize θ (x : Mx D a b)) ^ (N + N')) ≤ B) → (∀ x y, F x y * ENNReal.ofReal ((1 + realSize θ (y : Mx D c e)) ^ (N + N')) ≤ B) → ∑' x, ∑' y, F x y ≤ (n : ℝ≥0∞) ^ (N + N') * B * ((∑' x : ratMx D a b, ENNReal.ofReal (maj θ K N x)) * ∑' y : ratMx D c e, ENNReal.ofReal (maj θ K' N' y))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaLiftLastStepsCouplingCosetOrbitInvari

theorem **setIntegral_mul_right** {V : HSp D} (v : Vol V) (G : GA D V → ℂ) (hG : ∀ γ ∈ GQ V, ∀ h, G (γ ∗ h) = G h) (a : GA D V) : ∫ h in v.𝓕, G (h ∗ a) ∂v.μ = ∫ h in v.𝓕, G h ∂v.μ

theorem **biLift_mid** {A S X : HSp D} {χA χS χS' χX : (AE D)ˣ →∗ Circle} {oma : Om A S} {omb : Om S X} (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S X χS' χX omb) (vA : Vol A) (vS : Vol S) (φ : GA D A → ℂ) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) (g : GA D S) : biLift oma omb vA vS φ (fun u v => oma (1, g) (fun u' => omb (g, 1) (C u') v) u) = biLift oma omb vA vS φ C

theorem **biTheta_biComb** (hψ : IsStdChar D ψ (-1)) {A S X : HSp D} {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) (c : WO A S →₀ ℂ) (c' : WO S X →₀ ℂ) : IsBiTheta ψ A S X (biComb ψ A S X c c' C)

theorem **comb_aconv** {S T : HSp D} {e : AE D} (hc : cA D e = e) (he : e ∗ e = e) {c d : WO S T →₀ ℂ} (h₁ : ∀ u ∈ c.support, IsHalf (1 - e) u) (h₂ : ∀ u ∈ d.support, IsHalf e u) (f : WO S T → ℂ) : comb ψ S T (aconv [c, d]) f = comb ψ S T c (comb ψ S T d f)

theorem **biComb_aconv** {A S X : HSp D} {e : AE D} (hc : cA D e = e) (he : e ∗ e = e) {c₁ d₁ : WO A S →₀ ℂ} {c₂ d₂ : WO S X →₀ ℂ} (h₁ : ∀ u ∈ c₁.support, IsHalf (1 - e) u) (h₁' : ∀ u ∈ d₁.support, IsHalf e u) (h₂ : ∀ u ∈ c₂.support, IsHalf (1 - e) u) (h₂' : ∀ u ∈ d₂.support, IsHalf e u) (C : WO A S → WO S X → ℂ) : biComb ψ A S X (aconv [c₁, d₁]) (aconv [c₂, d₂]) C = biComb ψ A S X c₁ c₂ (biComb ψ A S X d₁ d₂ C)

theorem **actO_away** {S T : HSp D} {e : AE D} (he : (e + cA D e) ∗ (e + cA D e) = e + cA D e) {g : GA D S} {h : GA D T} (hg : SuppAt e S g) (hh : SuppAt e T h) {x : WO S T} (hx : IsHalf (1 - (e + cA D e)) x) : actO S T g h x = x

theorem **om_comb** {S T : HSp D} {χS χT : (AE D)ˣ →∗ Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {g : GA D S} {h : GA D T} {c : WO S T →₀ ℂ} (hc : ∀ x ∈ c.support, actO S T g h x = x) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (comb ψ S T c f) = comb ψ S T c (om (g, h) f)

theorem **act_biComb** {A S X : HSp D} {χA χS χS' χX : (AE D)ˣ →∗ Circle} {oma : Om A S} {omb : Om S X} (ha : IsWeilDatumO ψ A S χA χS oma) (hb : IsWeilDatumO ψ S X χS' χX omb) {g : GA D S} {c₁ : WO A S →₀ ℂ} {c₂ : WO S X →₀ ℂ} (h₁ : ∀ x ∈ c₁.support, actO A S 1 g x = x) (h₂ : ∀ x ∈ c₂.support, actO S X g 1 x = x) {C C₁ C₂ : WO A S → WO S X → ℂ} (hR : ∀ u, IsThetaFunO ψ S X (C u)) (eR : ∀ u, omb (g, 1) (C u) = C₁ u) (hL : ∀ v, IsThetaFunO ψ A S fun u => C₁ u v) (eL : ∀ v, oma (1, g) (fun u => C₁ u v) = fun u => C₂ u v) : (fun u v => oma (1, g) (fun u' => omb (g, 1) (biComb ψ A S X c₁ c₂ C u') v) u) = biComb ψ A S X c₁ c₂ C₂

theorem **blk_mul** (a b : Matrix (n ⊕ Fin 0) (n ⊕ Fin 0) k) : (a ∗ b).toBlocks₁₁ = a.toBlocks₁₁ ∗ b.toBlocks₁₁

theorem **det_blk** (g : GL (n ⊕ Fin 0) k) : GeneralLinearGroup.det (blk g) = GeneralLinearGroup.det g

theorem **box_mulVec** {g : GL (n ⊕ Fin 0) k} (hg : g ∈ KO k _) (c : kˣ) {y : n → k} (hy : y ∈ box n c) : (blk g).val ∗ᵥ y ∈ box n c

theorem **cosetInd_move** {g : GL (n ⊕ Fin 0) k} (hg : g ∈ KO k _) (c : kˣ) (y z : n → k) : cosetInd (box n c) (Submodule.Quotient.mk y) (((blk g)⁻¹).val ∗ᵥ z) = cosetInd (box n c) (Submodule.Quotient.mk ((blk g).val ∗ᵥ y)) z

theorem **om_xiPen** {T : Chain k s X Z} (hT : T.Laws) {A : RepFam k s X} {S M} (hA : T.Adapted A S M) {a} (ha : a ∈ A s 0 rfl) (g : (T.lev (s + 1) 0 rfl).J) (hg : g.1 ∈ KO k _) (y : X (s + 1) → k) : om (T.mu (s + 1)) 1 (blk g.1) (T.xiPen M a (Submodule.Quotient.mk y)) = ((((T.lev (s + 1) 0 rfl).lam g)⁻¹ : ℂˣ) : ℂ) • T.xiPen M a (Submodule.Quotient.mk ((blk g.1).val ∗ᵥ y))

theorem **om_xiTop** [MeasurableSpace k] [BorelSpace k] {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {T : Chain k s X Z} (hT : T.Laws) (c : kˣ) (g : (T.lev (s + 1) 0 rfl).J) (hg : g.1 ∈ KO k _) (y : X (s + 1) → k) : om (T.mu (s + 1 + 1)) (blk g.1) 1 (T.xiTop (box _ c) ψ ν (-1) (Submodule.Quotient.mk y)) = (((T.lev (s + 1) 0 rfl).lam g : ℂˣ) : ℂ) • T.xiTop (box _ c) ψ ν (-1) (Submodule.Quotient.mk ((blk g.1).val ∗ᵥ y))

theorem **isComp_inv** {p : Place D k} {χ : (AE D)ˣ →∗ Circle} {μ : kˣ →∗ ℂˣ} (h : p.IsComp χ⁻¹ μ) : p.IsComp χ μ⁻¹

theorem **isLoc_one** (p : Place D k) {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (F : Frame k S.rank n) : p.IsLoc F 1 1

theorem **biLift_model** (p : Place D k) {ψ : AddChar (AF D.host.F) ℂ} {A S X : HSp D} {χ χ' : (AE D)ˣ →∗ Circle} {oma : Om A S} {omb : Om S X} (ha : IsWeilDatumO ψ A S χ χ oma) (hb : IsWeilDatumO ψ S X χ' χ' omb) {μ μ' : kˣ →∗ ℂˣ} (hμ : p.IsComp χ⁻¹ μ) (hμ' : p.IsComp χ'⁻¹ μ') {n n' n'' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] {FA : Frame k A.rank n'} {FS : Frame k S.rank n} {FX : Frame k X.rank n''} {Λ} (hΛ : p.IsBiModel ψ FA FS FX Λ) (κ : GL n k) {c₁ : WO A S →₀ ℂ} {c₂ : WO S X →₀ ℂ} (h₁ : ∀ u ∈ c₁.support, IsHalf (1 - (p.e + cA D p.e)) u) (h₂ : ∀ u ∈ c₂.support, IsHalf (1 - (p.e + cA D p.e)) u) (he : (p.e + cA D p.e) ∗ (p.e + cA D p.e) = p.e + cA D p.e) (Ξ₁ : SB (Matrix n n' k)) (Ξ₂ : SB (Matrix n'' n k)) (Ξ Ξ' : SB (Matrix n n' k × Matrix n'' n k)) (hΞ : ∀ q, Ξ.1 q = Ξ₁.1 q.1 ∗ Ξ₂.1 q.2) (hΞ' : ∀ q, Ξ'.1 q = om μ 1 κ Ξ₁.1 q.1 ∗ om

theorem **lst_model** {D : D7} {c : Comm D} {v : VCom c} {χ} (Q : QAt v χ) (hc : c.Is) (hv : v.Is) (h : Q.s + 2 ∗ 0 + 2 = D.R.m) (hr : c.Lz.rank = 1) (hτ : (v.τ Q.s h)ᴴ ∗ ((c.bes Q.s 0 h).L.gram ∗ v.τ Q.s h) = (-(D.host.xi0 ∗ star D.host.xi0)) • c.Lz.gram) (s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h)) {X₁ X₂} (h₁ : GlobalTheory.FinPl.Vac c.ψ _ _ (v.N Q.s) (v.N (Q.s + 1)) (· = Q.q.v) X₁) (h₂ : GlobalTheory.FinPl.Vac c.ψ _ _ (v.N (Q.s + 1)) v.NT (· = Q.q.v) X₂) : ∃ Λ O, (ofSplit Q.q).IsBiModel c.ψ (Q.FX Q.s (Nat.le_succ _)) (Q.FX (Q.s + 1) le_rfl) Q.FM Λ ∧ (∀ q, O.1 q = Set.indicator {x | ∀ r l, x r l ∈ 𝒪[Q.k]} (fun _ => (1 : ℂ)) q.1 ∗ Set.indicator {x | ∀ r l, x r l ∈ 𝒪[Q.k]} (fun _ => (1 : ℂ)) q.2) ∧ c.lst Q.s h (v.τ Q.s h) s₁ X₁ X₂ = Λ O

theorem **orb** : ∀ (D : D7) (c : Comm D) (v : VCom c) χ (Q : QAt v χ), c.Is → v.Is → ∀ (h : Q.s + 2 ∗ 0 + 2 = D.R.m), c.Lz.rank = 1 → (v.τ Q.s h)ᴴ ∗ ((c.bes Q.s 0 h).L.gram ∗ v.τ Q.s h) = (-(D.host.xi0 ∗ star D.host.xi0)) • c.Lz.gram → ∀ s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h), s₁.χW = χ Q.s → ∀ oma omb, IsWeilDatumO c.ψ _ _ (χ Q.s) (χ Q.s) oma → IsWeilDatumO c.ψ _ _ (χ (Q.s + 1)) (χ (Q.s + 1)) omb → ∀ c₁ c₂, (∀ u ∈ c₁.support, IsHalf (1 - (Q.q.e + cA D Q.q.e)) u) → (∀ u ∈ c₂.support, IsHalf (1 - (Q.q.e + cA D Q.q.e)) u) → ∀ Ψ₁ Ψ₂, GlobalTheory.FinPl.Vac c.ψ _ _ (v.N Q.s) (v.N (Q.s + 1)) (· = Q.q.v) Ψ₁ → GlobalTheory.FinPl.Vac c.ψ _ _ (v.N (Q.s + 1)) v.NT (· = Q.q.v) Ψ₂ → ∀ φ, IsContAut (c.X Q.s) φ → ∀ α u y yb, SameOrb (Q.T.lev (Q.s + 1) 0 rfl).J (box _ Q.c₂) (Q.cos y) yb → ∀ d₁ d₂, (ofSplit Q.q).IsCombAt c.ψ (Q.FX Q.s (Nat.le_succ _)) (Q.FX (Q.s + 1) le_rfl) d₁ (Q.T.xiPen (box _ Q.c₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaLiftLastStepsCouplingCosetOrbitInvariStatement

theorem **_root_.GlobalTheory.T.doubleThetaLift_lastSteps_couplingCoset_orbit_invariant** : ∀ (D : D7) (c : Comm D) (v : VCom c) χ (Q : QAt v χ), c.Is → v.Is → ∀ (h : Q.s + 2 ∗ 0 + 2 = D.R.m), c.Lz.rank = 1 → (v.τ Q.s h)ᴴ ∗ ((c.bes Q.s 0 h).L.gram ∗ v.τ Q.s h) = (-(D.host.xi0 ∗ star D.host.xi0)) • c.Lz.gram → ∀ s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h), s₁.χW = χ Q.s → ∀ oma omb, IsWeilDatumO c.ψ _ _ (χ Q.s) (χ Q.s) oma → IsWeilDatumO c.ψ _ _ (χ (Q.s + 1)) (χ (Q.s + 1)) omb → ∀ c₁ c₂, (∀ u ∈ c₁.support, IsHalf (1 - (Q.q.e + cA D Q.q.e)) u) → (∀ u ∈ c₂.support, IsHalf (1 - (Q.q.e + cA D Q.q.e)) u) → ∀ Ψ₁ Ψ₂, GlobalTheory.FinPl.Vac c.ψ _ _ (v.N Q.s) (v.N (Q.s + 1)) (· = Q.q.v) Ψ₁ → GlobalTheory.FinPl.Vac c.ψ _ _ (v.N (Q.s + 1)) v.NT (· = Q.q.v) Ψ₂ → ∀ φ, IsContAut (c.X Q.s) φ → ∀ α u y yb, SameOrb (Q.T.lev (Q.s + 1) 0 rfl).J (box _ Q.c₂) (Q.cos y) yb → ∀ d₁ d₂, (ofSplit Q.q).IsCombAt c.ψ (Q.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaLiftValueBilinear

theorem **per_int** {ρ : ℕ} (B : BesLev D ρ) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {φ : GA D B.A → ℂ} (hφ : IsContAut B.A φ) {f : GA D B.V → ℂ} (hf : f ∈ contLeft B.V) : (∀ h, IntegrableOn (GlobalTheory.MixPer.besFun ψ 1 B.d B.o B.ζ φ f h) B.vN.𝒩 B.vN.ν) ∧ IntegrableOn (fun h => ∫ n in B.vN.𝒩, GlobalTheory.MixPer.besFun ψ 1 B.d B.o B.ζ φ f h n ∂B.vN.ν) B.vA.𝓕 B.vA.μ

theorem **per_add_left** {ρ : ℕ} (B : BesLev D ρ) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {φ₁ φ₂ : GA D B.A → ℂ} (h₁ : IsContAut B.A φ₁) (h₂ : IsContAut B.A φ₂) {f : GA D B.V → ℂ} (hf : f ∈ contLeft B.V) : B.per ψ (φ₁ + φ₂) f = B.per ψ φ₁ f + B.per ψ φ₂ f

theorem **biComb_add** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (K K' : WO A S → WO S A' → ℂ) : biComb ψ A S A' c c' (K + K') = biComb ψ A S A' c c' K + biComb ψ A S A' c c' K'

theorem **biComb_smul** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (a : ℂ) (K : WO A S → WO S A' → ℂ) : biComb ψ A S A' c c' (a • K) = a • biComb ψ A S A' c c' K

theorem **GlobalTheory.TriStep.valK_bilin** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → (∀ Ξ, IsBiTheta ψ S.B.A S.J.W₀ S.B'.A (Λ Ξ)) → ∃ T : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] GlobalTheory.TriStep.space Q →ₗ[ℂ] ℂ, ∀ Ξ f, T Ξ f = GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ Ξ f.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaLiftValueBilinearStatement

theorem **_root_.GlobalTheory.T.doubleThetaLiftValue_bilinear** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → (∀ Ξ, IsBiTheta ψ S.B.A S.J.W₀ S.B'.A (Λ Ξ)) → ∃ T : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] GlobalTheory.TriStep.space Q →ₗ[ℂ] ℂ, ∀ Ξ f, T Ξ f = GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ Ξ f.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaLiftValueInvariantMiddleGL

theorem **comb_right_comm** (ψ : AddChar (AF D.host.F) ℂ) {A S A' : HSp D} (L : Module.End ℂ (WO A S → ℂ)) (K : WO A S → WO S A' → ℂ) (c : WO S A' →₀ ℂ) (u : WO A S) (v : WO S A') : comb ψ S A' c (fun v' => L (fun u' => K u' v') u) v = L (fun u' => comb ψ S A' c (K u') v) u

theorem **comb_left_comm** (ψ : AddChar (AF D.host.F) ℂ) {A S A' : HSp D} (L : Module.End ℂ (WO S A' → ℂ)) (K : WO A S → WO S A' → ℂ) (c : WO A S →₀ ℂ) (u : WO A S) (v : WO S A') : comb ψ A S c (fun u' => L (K u') v) u = L (fun v' => comb ψ A S c (fun u' => K u' v') u) v

theorem **comb_om_comm** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {ω : Om S T} (h : IsWeilDatumO ψ S T χS χT ω) (g : GA D S) (g' : GA D T) (c : WO S T →₀ ℂ) (hc : ∀ t ∈ c.support, actO S T g g' t = t) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : comb ψ S T c (ω (g, g') f) = ω (g, g') (comb ψ S T c f)

theorem **idem_mul** {e : AE D} (he : IsSplitIdem D e) : (e + cA D e) ∗ (1 − (e + cA D e)) = 0

theorem **actO_fix_left** {e : AE D} (he : IsSplitIdem D e) {S T : HSp D} {a : GA D T} (ha : SuppAt e T a) {t : WO S T} (ht : IsHalf (1 − (e + cA D e)) t) : actO S T 1 a t = t

theorem **actO_fix_right** {e : AE D} (he : IsSplitIdem D e) {S T : HSp D} {a : GA D S} (ha : SuppAt e S a⁻¹) {t : WO S T} (ht : IsHalf (1 − (e + cA D e)) t) : actO S T a 1 t = t

theorem **valK_glMid** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (g : GL Q.n₁ Q.k) (Ξ₁ Ξ₁' : SB (Matrix Q.n₁ Q.n₀ Q.k)) (Ξ₂ Ξ₂' : SB (Matrix Q.n₂ Q.n₁ Q.k)), Ξ₁'.1 = om Q.μ₁ 1 g Ξ₁.1 → Ξ₂'.1 = om Q.μ₂ g 1 Ξ₂.1 → IsBiTheta ψ S.B.A S.J.W₀ S.B'.A (biComb ψ S.B.A S.J.W₀ S.B'.A c₁ c₂ (Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂))) → ∀ f ∈ GlobalTheory.TriStep.space Q, GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁' Ξ₂') f = GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaLiftValueInvariantMiddleGLStatement

theorem **_root_.GlobalTheory.T.doubleThetaLiftValue_invariant_middleGL** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (g : GL Q.n₁ Q.k) (Ξ₁ Ξ₁' : SB (Matrix Q.n₁ Q.n₀ Q.k)) (Ξ₂ Ξ₂' : SB (Matrix Q.n₂ Q.n₁ Q.k)), Ξ₁'.1 = om Q.μ₁ 1 g Ξ₁.1 → Ξ₂'.1 = om Q.μ₂ g 1 Ξ₂.1 → IsBiTheta ψ S.B.A S.J.W₀ S.B'.A (biComb ψ S.B.A S.J.W₀ S.B'.A c₁ c₂ (Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂))) → ∀ f ∈ GlobalTheory.TriStep.space Q, GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁' Ξ₂') f = GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaStepIntegrandFiniteDoubleCosets

theorem **prod_mul_inv_mem** {G : Type} [Group G] (H : Subgroup G) : ∀ {n : ℕ} (f : Fin n → G) (i : Fin n), (∀ j, j ≠ i → f j ∈ H) → (∀ h ∈ H, f i ∗ h = h ∗ f i) → (List.ofFn f).prod ∗ (f i)⁻¹ ∈ H

theorem **prod_ofFn_mul** {G : Type} [Monoid G] : ∀ {n : ℕ} (f g : Fin n → G), (∀ i j, i ≠ j → Commute (g i) (f j)) → (List.ofFn fun i => f i ∗ g i).prod = (List.ofFn f).prod ∗ (List.ofFn g).prod

theorem **cells_prod** {G : Type} [Group G] {r : ℕ} (H : Fin (r + 1) → Type) [∀ i, Group (H i)] (φ : ∀ i, H i →* G) (O : Fin (r + 1) → Subgroup G) (hO : ∀ i j, j ≠ i → ∀ h, φ j h ∈ O i) (hc : ∀ i h, ∀ y ∈ O i, φ i h ∗ y = y ∗ φ i h) (A : Subgroup G) (hA : ∀ (i : Fin r) h, φ i.succ h ∈ A) (hA' : ∀ a ∈ A, ∃ x : ∀ i : Fin r, H i.succ, a = (List.ofFn fun i => φ i.succ (x i)).prod) (K : Subgroup G) (J : ∀ i, Subgroup (H i)) (R₀ : Subgroup (H 0)) (hR : ∀ h, φ 0 h ∈ K → h ∈ R₀) (N : G → Prop) (hN : ∀ i, ∃ T : Finset (H i), ∀ z ∈ O i, ∀ x, N (z ∗ φ i x) → ∃ t ∈ T, ∃ ρ, φ i ρ ∈ K ∧ ∃ j ∈ J i, x = ρ ∗ t ∗ j) : ∃ T : Finset (A × H 0), ∀ x : A × H 0, N (x.1.1 ∗ φ 0 x.2) → ∃ t ∈ T, ∃ ρ ∈ (K.subgroupOf A).prod R₀, ∃ κ ∈ ((⊔ i : Fin r, (J i.succ).map (φ i.succ)).subgroupOf A).prod (J 0), x = ρ ∗ t ∗ κ

theorem **conv_support** {G : Type} [Monoid G] : ∀ {n : ℕ} (H : Fin n → Type) (φ : ∀ i, H i → G) (e : ∀ i, H i →₀ ℂ), ∀ a ∈ (conv (List.ofFn fun i => (e i).mapDomain (φ i))).support, ∃ b : ∀ i, H i, (∀ i, b i ∈ (e i).support) ∧ a = (List.ofFn fun i => φ i (b i)).prod

theorem **aconv_support** {G : Type} [AddMonoid G] (p : G → Prop) (h0 : p 0) (hadd : ∀ x y, p x → p y → p (x + y)) : ∀ l : List (G →₀ ℂ), (∀ c ∈ l, ∀ u ∈ c.support, p u) → ∀ u ∈ (aconv l).support, p u

theorem **comb_aconv_cons** (c : WO S T →₀ ℂ) (l : List (WO S T →₀ ℂ)) (h : ∀ x ∈ c.support, ∀ y ∈ (aconv l).support, symplO S T x y = 0) (f : WO S T → ℂ) : comb ψ S T (aconv (c :: l)) f = comb ψ S T c (comb ψ S T (aconv l) f)

theorem **off_zero** (e : AE D) : Off S T e 0

theorem **off_add** (e : AE D) (x y : WO S T) (hx : Off S T e x) (hy : Off S T e y) : Off S T e (x + y)

theorem **off_of_half** {e e' : AE D} (h : e ∗ e' = 0) (hi : infPart D e' = 0) (u : WO S T) (hu : IsHalf e' u) : Off S T e u

theorem **symplO_off** {e : AE D} (hc : cA D e = e) (he : e ∗ e = e) {x y : WO S T} (hx : IsHalf e x) (hy : Off S T e y) : symplO S T x y = 0

theorem **comb_aconv_strip** : ∀ {n : ℕ} (E : Fin n → AE D) (c : Fin n → WO S T →₀ ℂ), (∀ i, cA D (E i) = E i) → (∀ i, E i ∗ E i = E i) → (∀ i j, i ≠ j → E i ∗ E j = 0) → (∀ i, infPart D (E i) = 0) → (∀ i, ∀ u ∈ (c i).support, IsHalf (E i) u) → ∀ i, ∃ c' : WO S T →₀ ℂ, (∀ u ∈ c'.support, Off S T (E i) u) ∧ ∀ f, comb ψ S T (aconv (List.ofFn c)) f = comb ψ S T (c i) (comb ψ S T c' f)

theorem **comb_single** (y : WO S T) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single y 1) f = heisO ψ S T y f

theorem **model_comb** (hψ : IsStdChar D ψ (-1)) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (hc : cA D (p.e + cA D p.e) = p.e + cA D p.e) (he : (p.e + cA D p.e) ∗ (p.e + cA D p.e) = p.e + cA D p.e) (c : WO S T →₀ ℂ) (h : ∀ u ∈ c.support, Off S T (p.e + cA D p.e) u) (f : WO S T → ℂ) (hf : ∃ M O, p.IsModel ψ FS FT M ∧ IsOne O ∧ M O = f) : ∃ M O, p.IsModel ψ FS FT M ∧ IsOne O ∧ M O = comb ψ S T c f

theorem **sum_eq_eFin** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) : q.e + cA D q.e = algebraMap (AF D.host.F) (AE D) (eFin D.host.F q.v)

theorem **exists_vac** (hψ : IsStdChar D ψ (-1)) {r : ℕ} (Q : Fin r → QStep S) (hinj : Function.Injective fun i => (Q i).q.v) (f₁ : WO S.B.V S.J.W → ℂ) (f₂ : WO S.J.W S.B'.V → ℂ) (hv : ∀ i, (Q i).Vac f₁ f₂) (i : Fin r) : ∃ f₁' f₂', (Q i).Vac f₁' f₂' ∧ comb ψ _ _ (aconv (List.ofFn fun i => (Q i).D₁)) f₁ = comb ψ _ _ (Q i).D₁ f₁' ∧ comb ψ _ _ (aconv (List.ofFn fun i => (Q i).D₂)) f₂ = comb ψ _ _ (Q i).D₂ f₂'

theorem **con_ne_zero** {c₁ c₂ Φ₁ Φ₂ φ f g h₁ h₂} (h : S.val c₁ c₂ Φ₁ Φ₂ φ f g h₁ h₂ ≠ 0) : S.con Φ₁ Φ₂ g h₁ h₂ ≠ 0

theorem **flat_ne_zero** {Φ₁ Φ₂ g h₁ h₂} (h : S.con Φ₁ Φ₂ g h₁ h₂ ≠ 0) : flatBF S.s₁ (S.s₁.om (g, h₁) Φ₁) ≠ 0

theorem **gl_mem_at** (h : ∀ i g, (ofSplit (q i)).IsLoc (F i) (gl i g) g) {P : GlobalTheory.FinPl.Pl D → Prop} (i : Fin r) (hP : P (q i).v) (g : GL (Y i) (k i)) : gl i g ∈ GlobalTheory.FinPl.grpAt V P

theorem **gl_mem_off** (hinj : Function.Injective fun i => (q i).v) (h : ∀ i g, (ofSplit (q i)).IsLoc (F i) (gl i g) g) (i j : Fin r) (hij : j ≠ i) (g : GL (Y j) (k j)) : gl j g ∈ GlobalTheory.FinPl.grpOff V (· = (q i).v)

theorem **gl_comm** (h : ∀ i g, (ofSplit (q i)).IsLoc (F i) (gl i g) g) (i : Fin r) (g : GL (Y i) (k i)) : ∀ y ∈ GlobalTheory.FinPl.grpOff V (· = (q i).v), gl i g ∗ y = y ∗ gl i g

theorem **exists_prod** (hinj : Function.Injective fun i => (q i).v) (h : ∀ i g, (ofSplit (q i)).IsLoc (F i) (gl i g) g) : ∀ a ∈ GlobalTheory.FinPl.grpAt V fun v => ∃ i, (q i).v = v, ∃ x : ∀ i, GL (Y i) (k i), a = (List.ofFn fun i => gl i (x i)).prod

theorem **exists_con** {r : ℕ} (Q : Fin r → QStep S) (hinj : Function.Injective fun i => (Q i).q.v) {c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y} (hg : g ∈ grpOff S.B.V (places Q)) (h : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y ≠ 0) (i : Fin r) : ∃ a ∈ (Q i).e₀.support, ∃ g' ∈ grpOff S.B.V (· = (Q i).q.v), S.con (comb ψ _ _ (aconv (List.ofFn fun i => (Q i).D₁)) f₁) (comb ψ _ _ (aconv (List.ofFn fun i => (Q i).D₂)) f₂) (g' ∗ (Q i).gl₀ a) (h₁ ∗ x) (h₂ ∗ y) ≠ 0

theorem **cells_first** (hψ : IsStdChar D ψ (-1)) {r : ℕ} (Q : Fin r → QStep S) (hinj : Function.Injective fun i => (Q i).q.v) {c₁ c₂ f₁ f₂ φ f g h₁ h₂} (hH : Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂) (i : Fin r) : ∃ T : Finset (GL ((Q i).n₁ ⊕ Fin ((Q i).c + 1 + 1)) (Q i).k), ∀ z ∈ grpOff S.J.W (· = (Q i).q.v), ∀ x, (∃ y, integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (z ∗ (Q i).gl₁ x) y ≠ 0) → ∃ t ∈ T, ∃ ρ, (Q i).gl₁ ρ ∈ jacGrp S.J.d ∧ ∃ j ∈ (Q i).L₁.J, x = ρ ∗ t ∗ j

theorem **cells_second** (hψ : IsStdChar D ψ (-1)) {r : ℕ} (Q : Fin r → QStep S) (hinj : Function.Injective fun i => (Q i).q.v) {c₁ c₂ f₁ f₂ φ f g h₁ h₂} (hH : Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂) (x : GA D S.J.W) (i : Fin r) : ∃ T : Finset (GL ((Q i).n₂ ⊕ Fin ((Q i).c + 1)) (Q i).k), ∀ z ∈ grpOff S.B'.V (· = (Q i).q.v), ∀ y, integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x (z ∗ (Q i).gl₂ y) ≠ 0 → ∃ t ∈ T, ∃ ρ, (Q i).gl₂ ρ ∈ besGrp S.B'.d S.B'.o ∧ ∃ j ∈ (Q i).L₂.J, y = ρ ∗ t ∗ j

theorem **finite_cells** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → (∃ T : Finset (grpAt S.J.W (places fun i => Q i.succ) × GL ((Q 0).n₁ ⊕ Fin ((Q 0).c + 1 + 1)) (Q 0).k), ∀ x y, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y ≠ 0 → ∃ t ∈ T, ∃ r ∈ ((jacGrp S.J.d).subgroupOf (grpAt S.J.W (places fun i => Q i.succ))).prod (RFJ (Q 0).k (Q 0).n₁ ((Q 0).c + 1 + 1) (σ + 1)), ∃ κ ∈ (levJ fun i => Q i.succ).prod (Q 0).L₁.J, x = r ∗ t ∗ κ) ∧ ∀ x, ∃ T : Finset (grpAt S.B'.V (places fun i => Q i.succ) × GL ((Q 0).n₂ ⊕ Fin ((Q 0).c + 1)) (Q 0).k), ∀ y, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y ≠ 0 → ∃ t ∈ T, ∃ r ∈ ((besGrp S.B'.d S.B'.o).subgroupOf (grpAt S.B'.V (places fun i => Q i.succ))).prod (R (Q 0).k (Q 0).n₂ ((Q 0).c + 1) (σ + 1)), ∃

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaStepIntegrandFiniteDoubleCosetsStatement

theorem **_root_.GlobalTheory.T.doubleThetaStep_integrand_finite_doubleCosets** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → (∃ T : Finset (grpAt S.J.W (places fun i => Q i.succ) × GL ((Q 0).n₁ ⊕ Fin ((Q 0).c + 1 + 1)) (Q 0).k), ∀ x y, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y ≠ 0 → ∃ t ∈ T, ∃ r ∈ ((jacGrp S.J.d).subgroupOf (grpAt S.J.W (places fun i => Q i.succ))).prod (RFJ (Q 0).k (Q 0).n₁ ((Q 0).c + 1 + 1) (σ + 1)), ∃ κ ∈ (levJ fun i => Q i.succ).prod (Q 0).L₁.J, x = r ∗ t ∗ κ) ∧ ∀ x, ∃ T : Finset (grpAt S.B'.V (places fun i => Q i.succ) × GL ((Q 0).n₂ ⊕ Fin ((Q 0).c + 1)) (Q 0).k), ∀ y, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y ≠ 0 → ∃ t ∈ T, ∃ r ∈ ((besGrp S.B'.d S.B'.o).subgroupOf (grpAt S.B'.V (places fun i => Q i.suc

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaStepIntegrandInvariantAtSuccPlaces

theorem **infPart_idem** (q : GlobalTheory.FinPl.SplitPt D k) : infPart D (q.e + cA D q.e) = 0

theorem **cA_idem** (q : GlobalTheory.FinPl.SplitPt D k) : cA D (q.e + cA D q.e) = q.e + cA D q.e

theorem **idem_mul_idem** (q : GlobalTheory.FinPl.SplitPt D k) (q' : GlobalTheory.FinPl.SplitPt D k') (hne : q.v ≠ q'.v) : (q.e + cA D q.e) ∗ (q'.e + cA D q'.e) = 0

theorem **idem_sq** (q : GlobalTheory.FinPl.SplitPt D k) : (q.e + cA D q.e) ∗ (q.e + cA D q.e) = q.e + cA D q.e

theorem **aconv_cons** {G : Type} [AddMonoid G] (e : G →₀ ℂ) (l : List (G →₀ ℂ)) : aconv (e :: l) = ∑ g ∈ e.support, ∑ h ∈ (aconv l).support, Finsupp.single (g + h) (e g ∗ aconv l h)

theorem **support_aconv** {G : Type} [AddMonoid G] (p : G → Prop) (h0 : p 0) (hadd : ∀ x y, p x → p y → p (x + y)) (l : List (G →₀ ℂ)) (hl : ∀ e ∈ l, ∀ u ∈ e.support, p u) : ∀ u ∈ (aconv l).support, p u

theorem **comb_aconv_cons** (e : WO S T →₀ ℂ) (l : List (WO S T →₀ ℂ)) (h : ∀ x ∈ e.support, ∀ y ∈ (aconv l).support, symplO S T x y = 0) (f : WO S T → ℂ) : comb ψ S T (aconv (e :: l)) f = comb ψ S T e (comb ψ S T (aconv l) f)

theorem **half_add** {e : AE D} {x y : WO S T} (hx : IsHalf e x) (hy : IsHalf e y) : IsHalf e (x + y)

theorem **half_zero** (e : AE D) : IsHalf e (0 : WO S T)

theorem **half_compl** {e e' : AE D} (h : e ∗ e' = 0) {x : WO S T} (hx : IsHalf e' x) : IsHalf (1 − e) x

theorem **fin_add** {x y : WO S T} (hx : x.map (infPart D) = 0) (hy : y.map (infPart D) = 0) : (x + y).map (infPart D) = 0

theorem **fin_of_half** {e : AE D} (he : infPart D e = 0) {u : WO S T} (hu : IsHalf e u) : u.map (infPart D) = 0

theorem **comb_aconv_isolate** : ∀ (n : ℕ) (ε : Fin n → AE D) (d : Fin n → WO S T →₀ ℂ), (∀ j, cA D (ε j) = ε j) → (∀ j, ε j ∗ ε j = ε j) → (∀ j j', j ≠ j' → ε j ∗ ε j' = 0) → (∀ j, infPart D (ε j) = 0) → (∀ j, ∀ u ∈ (d j).support, IsHalf (ε j) u) → ∀ i, ∃ c : WO S T →₀ ℂ, (∀ u ∈ c.support, IsHalf (1 − ε i) u ∧ u.map (infPart D) = 0) ∧ ∀ f, comb ψ S T (aconv (List.ofFn d)) f = comb ψ S T (d i) (comb ψ S T c f)

theorem **comb_single_one** (y : WO S T) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single y 1) f = heisO ψ S T y f

theorem **isModel_comb** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (hψ : IsStdChar D ψ (−1)) (q : GlobalTheory.FinPl.SplitPt D k) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (c : WO S T →₀ ℂ) (hc : ∀ u ∈ c.support, IsHalf (1 − (q.e + cA D q.e)) u ∧ u.map (infPart D) = 0) (Tm) (h : (ofSplit q).IsModel ψ FS FT Tm) : (ofSplit q).IsModel ψ FS FT (combLin ψ S T c ∘ₗ Tm)

theorem **eq_one_of_grpAt_empty** {x : GA D V} (hx : x ∈ grpAt V fun _ => False) : x = 1

theorem **induction_places** {n : ℕ} (v : Fin n → GlobalTheory.FinPl.Pl D) (H : Subgroup (GA D V)) (C : GA D V → Prop) (h1 : C 1) (hmul : ∀ x y, C x → C y → C (x ∗ y)) (hloc : ∀ i, ∀ x ∈ H, ∃ a ∈ H, C a ∧ a ∈ grpAt V (· = v i) ∧ x ∗ a⁻¹ ∈ grpOff V (· = v i)) : ∀ x ∈ H, x ∈ grpAt V (fun p => ∃ i, v i = p) → C x

theorem **grpOff_of_places** {P : GlobalTheory.FinPl.Pl D → Prop} {v : GlobalTheory.FinPl.Pl D} (hv : P v) {x : GA D V} (hx : x ∈ grpOff V P) : x ∈ grpOff V (· = v)

theorem **val_inv** (hψ : IsStdChar D ψ (−1)) (Q : QStep S) {c₁ c₂} (h₁ : Q.Away c₁) (h₂ : Q.Away c₂) {f₁ f₂} (hv : Q.Vac f₁ f₂) {φ f} (hφ : IsContAut S.B.A φ) (hf : IsContAut S.B'.V f) (ht : Q.Typed f) (g : GA D S.B.V) (x : GA D S.J.W) (y : GA D S.B'.V) {a j a' j'} (ha : a ∈ RFJ Q.k Q.n₁ (Q.c + 1 + 1) (σ + 1)) (hj : j ∈ Q.L₁.J) (ha' : a' ∈ R Q.k Q.n₂ (Q.c + 1) (σ + 1)) (hj' : j' ∈ Q.L₂.J) : S.val c₁ c₂ (comb ψ _ _ Q.D₁ f₁) (comb ψ _ _ Q.D₂ f₂) φ f g (Q.gl₁ a ∗ x ∗ Q.gl₁ j) (Q.gl₂ a' ∗ y ∗ Q.gl₂ j') = S.val c₁ c₂ (comb ψ _ _ Q.D₁ f₁) (comb ψ _ _ Q.D₂ f₂) φ f g x y

theorem **integrand_inv** (hψ : IsStdChar D ψ (−1)) {n : ℕ} (Q : Fin n → QStep S) (hinj : Function.Injective fun i => (Q i).q.v) {c₁ c₂ f₁ f₂ φ f g h₁ h₂} (hH : Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂) (i : Fin n) (x : GA D S.J.W) (y : GA D S.B'.V) {a j a' j'} (ha : a ∈ RFJ (Q i).k (Q i).n₁ ((Q i).c + 1 + 1) (σ + 1)) (hj : j ∈ (Q i).L₁.J) (ha' : a' ∈ R (Q i).k (Q i).n₂ ((Q i).c + 1) (σ + 1)) (hj' : j' ∈ (Q i).L₂.J) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ ((Q i).gl₁ a ∗ x ∗ (Q i).gl₁ j) ((Q i).gl₂ a' ∗ y ∗ (Q i).gl₂ j') = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **left₁** (i : Fin n) {a} (ha : a ∈ RFJ (Q i).k (Q i).n₁ ((Q i).c + 1 + 1) (σ + 1)) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ ((Q i).gl₁ a ∗ x) y = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **right₁** (i : Fin n) {j} (hj : j ∈ (Q i).L₁.J) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (x ∗ (Q i).gl₁ j) y = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **left₂** (i : Fin n) {a} (ha : a ∈ R (Q i).k (Q i).n₂ ((Q i).c + 1) (σ + 1)) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x ((Q i).gl₂ a ∗ y) = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **right₂** (i : Fin n) {j} (hj : j ∈ (Q i).L₂.J) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x (y ∗ (Q i).gl₂ j) = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **left_jac** {m : ℕ} (τ : Fin m → Fin n) {ρ : GA D S.J.W} (hρ : ρ ∈ jacGrp S.J.d) (hP : ρ ∈ grpAt S.J.W (places fun i => Q (τ i))) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (ρ ∗ x) y = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **left_bes** {m : ℕ} (τ : Fin m → Fin n) {ρ : GA D S.B'.V} (hρ : ρ ∈ besGrp S.B'.d S.B'.o) (hP : ρ ∈ grpAt S.B'.V (places fun i => Q (τ i))) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x (ρ ∗ y) = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **right_lev₁** {m : ℕ} (τ : Fin m → Fin n) {κ : GA D S.J.W} (hκ : κ ∈ ⊔ i, (Q (τ i)).L₁.J.map (Q (τ i)).gl₁) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (x ∗ κ) y = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **right_lev₂** {m : ℕ} (τ : Fin m → Fin n) {κ : GA D S.B'.V} (hκ : κ ∈ ⊔ i, (Q (τ i)).L₂.J.map (Q (τ i)).gl₂) (x y) : integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x (y ∗ κ) = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y

theorem **pint_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → (∀ x y, ∀ r ∈ ((besGrp S.B'.d S.B'.o).subgroupOf (grpAt S.B'.V (places fun i => Q i.succ))).prod (R (Q 0).k (Q 0).n₂ ((Q 0).c + 1) (σ + 1)), ∀ κ ∈ (levB fun i => Q i.succ).prod (Q 0).L₂.J, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x (r ∗ y ∗ κ) = pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y) ∧ (∀ x y, ∀ κ ∈ (levJ fun i => Q i.succ).prod (Q 0).L₁.J, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (x ∗ κ) y = pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y) ∧ (∀ x y, ∀ r ∈ ((jacGrp S.J.d).subgroupOf (grpAt S.J.W (places fun i => Q i.succ))).prod (RFJ (Q 0).k (Q 0).n₁ ((Q 0).c + 1 + 1) (σ + 1)), pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (r ∗ x) y = pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaStepIntegrandInvariantAtSuccPlacesStatement

theorem **_root_.GlobalTheory.T.doubleThetaStep_integrand_invariant_at_succ_places** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → (∀ x y, ∀ r ∈ ((besGrp S.B'.d S.B'.o).subgroupOf (grpAt S.B'.V (places fun i => Q i.succ))).prod (R (Q 0).k (Q 0).n₂ ((Q 0).c + 1) (σ + 1)), ∀ κ ∈ (levB fun i => Q i.succ).prod (Q 0).L₂.J, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x (r ∗ y ∗ κ) = pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y) ∧ (∀ x y, ∀ κ ∈ (levJ fun i => Q i.succ).prod (Q 0).L₁.J, pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (x ∗ κ) y = pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y) ∧ (∀ x y, ∀ r ∈ ((jacGrp S.J.d).subgroupOf (grpAt S.J.W (places fun i => Q i.succ))).prod (RFJ (Q 0).k (Q 0).n₁ ((Q 0).c + 1 + 1) (σ + 1)), pint Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (r ∗ x) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaStepSplitPlaceValuePeriodGroupInvaria

theorem **val_move** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} (S : Dbl2 ψ σ) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (θ₁ : WO S.B.V S.J.W → ℂ) (θ₂ : WO S.J.W S.B'.V → ℂ) (φ : GA D S.B.A → ℂ) (f : GA D S.B'.V → ℂ) (g : GA D S.B.V) (h₁ : GA D S.J.W) (h₂ : GA D S.B'.V) : S.val c₁ c₂ θ₁ θ₂ φ f g h₁ h₂ = S.val c₁ c₂ (S.s₁.om (g, h₁) θ₁) (S.s₂.om (h₁, h₂) θ₂) φ (rTrans h₂ f) 1 1 1

theorem **val_inv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) c₁ c₂, Q.Away c₁ → Q.Away c₂ → ∀ f₁ f₂, Q.Vac f₁ f₂ → ∀ φ f, IsContAut S.B.A φ → IsContAut S.B'.V f → Q.Typed f → ∀ g h₁ h₂, g ∈ GlobalTheory.FinPl.grpOff S.B.V (· = Q.q.v) → h₁ ∈ GlobalTheory.FinPl.grpOff S.J.W (· = Q.q.v) → h₂ ∈ GlobalTheory.FinPl.grpOff S.B'.V (· = Q.q.v) → ∀ a x y, ∀ r ∈ RFJ Q.k Q.n₁ (Q.c + 1 + 1) (σ + 1), ∀ j ∈ Q.L₁.J, ∀ r' ∈ R Q.k Q.n₂ (Q.c + 1) (σ + 1), ∀ j' ∈ Q.L₂.J,

S.val c₁ c₂ (comb ψ _ _ Q.D₁ f₁) (comb ψ _ _ Q.D₂ f₂) φ f (g ∗ Q.gl₀ a) (h₁ ∗ Q.gl₁ (r ∗ x ∗ j)) (h₂ ∗ Q.gl₂ (r' ∗ y ∗ j')) = S.val c₁ c₂ (comb ψ _ _ Q.D₁ f₁) (comb ψ _ _ Q.D₂ f₂) φ f (g ∗ Q.gl₀ a) (h₁ ∗ Q.gl₁ x) (h₂ ∗ Q.gl₂ y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaStepSplitPlaceValuePeriodGroupInvariaStatement

theorem **_root_.GlobalTheory.T.doubleThetaStep_splitPlace_value_periodGroup_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) c₁ c₂, Q.Away c₁ → Q.Away c₂ → ∀ f₁ f₂, Q.Vac f₁ f₂ → ∀ φ f, IsContAut S.B.A φ → IsContAut S.B'.V f → Q.Typed f → ∀ g h₁ h₂, g ∈ GlobalTheory.FinPl.grpOff S.B.V (· = Q.q.v) → h₁ ∈ GlobalTheory.FinPl.grpOff S.J.W (· = Q.q.v) → h₂ ∈ GlobalTheory.FinPl.grpOff S.B'.V (· = Q.q.v) → ∀ a x y, ∀ r ∈ RFJ Q.k Q.n₁ (Q.c + 1 + 1) (σ + 1), ∀ j ∈ Q.L₁.J, ∀ r' ∈ R Q.k Q.n₂ (Q.c + 1) (σ + 1), ∀ j' ∈ Q.L₂.J, S.val c₁ c₂ (comb ψ _ _ Q.D₁ f₁) (comb ψ _ _ Q.D₂ f₂) φ f (g ∗ Q.gl₀ a) (h₁ ∗ Q.gl₁ (r ∗ x ∗ j)) (h₂ ∗ Q.gl₂ (r' ∗ y ∗ j')) = S.val c₁ c₂ (comb ψ _ _ Q.D₁ f₁) (comb ψ _ _ Q.D₂ f₂) φ f (g ∗ Q.gl₀ a) (h₁ ∗ Q.gl₁ x) (h₂ ∗ Q.gl₂ y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleThetaUnfoldingEquivariantUnitaryGroups

theorem **integral_right** {V : HSp D} (v : Vol V) {F : GA D V → ℂ} (hF : ∀ γ ∈ GQ V, ∀ g, F (γ ∗ g) = F g) (a : GA D V) : ∫ g in v.𝓕, F (g ∗ a) ∂v.μ = ∫ g in v.𝓕, F g ∂v.μ

theorem **IsContAut.integrableOn** {V : HSp D} (hV : CoCpt V) (v : Vol V) {F : GA D V → ℂ} (hF : IsContAut V F) : IntegrableOn F v.𝓕 v.μ

theorem **contAut_kerBF** {φ : GA D B.A → ℂ} {Ξ : WO B.V₀ J.W₀ → ℂ} (hφ : IsContAut B.A φ) (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (h : GA D J.W₀) (u : WO B.L J.W₀) : IsContAut B.A fun a => φ a ∗ s.om₀ (B.o.emb (a, 1), h) Ξ ((B.o.resS J.W₀).symm (0, u))

theorem **handBF_left** {φ φ' : GA D B.A → ℂ} {Ξ : WO B.V₀ J.W₀ → ℂ} (hφ : IsContAut B.A φ) (hφ' : IsContAut B.A φ') (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (c : ℂ) : handBF s l (c • φ + φ') Ξ = star c • handBF s l φ Ξ + handBF s l φ' Ξ

theorem **handBF_right** {φ : GA D B.A → ℂ} {Ξ Ξ' : WO B.V₀ J.W₀ → ℂ} (hφ : IsContAut B.A φ) (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (hΞ' : IsThetaFunO ψ B.V₀ J.W₀ Ξ') (c : ℂ) : handBF s l φ (c • Ξ + Ξ') = star c • handBF s l φ Ξ + handBF s l φ Ξ'

theorem **handBF_equiv** {φ : GA D B.A → ℂ} {Ξ : WO B.V₀ J.W₀ → ℂ} (hφ : IsContAut B.A φ) (hΞ : IsThetaFunO ψ B.V₀ J.W₀ Ξ) (a : GA D B.A) (g : GA D J.W₀) : handBF s l (fun x => φ (x ∗ a)) (s.om₀ (B.o.emb (a, 1), g) Ξ) = fun h => handBF s l φ Ξ (h ∗ g)

theorem **integrableOn_pair** {Y : GA D J.W₀ → WO J.W₀ B.L → ℂ} {F : WO J.W₀ B.V₀ → ℂ} (hY : IsQuasiFam ψ J.W₀ B.L Y) (hF : IsThetaFunO ψ J.W₀ B.V₀ F) (h : GA D J.W₀) : IntegrableOn (fun v => F ((B.o.resT J.W₀).symm (0, v)) ∗ star (Y h v)) vL.𝓕 vL.μ

theorem **contAut_pairKer** {Y : GA D J.W₀ → WO J.W₀ B.L → ℂ} {Ξ : WO J.W₀ B.V₀ → ℂ} (hY : IsQuasiFam ψ J.W₀ B.L Y) (hΞ : IsThetaFunO ψ J.W₀ B.V₀ Ξ) (h₁ : GA D B.A) : IsContAut J.W₀ fun h₀ => pairKer s vL Y Ξ h₀ h₁

theorem **handFB_left** {Y Y' : GA D J.W₀ → WO J.W₀ B.L → ℂ} {Ξ : WO J.W₀ B.V₀ → ℂ} (hY : IsQuasiFam ψ J.W₀ B.L Y) (hY' : IsQuasiFam ψ J.W₀ B.L Y') (hΞ : IsThetaFunO ψ J.W₀ B.V₀ Ξ) (c : ℂ) : handFB s vL (star c • Y + Y') Ξ = c • handFB s vL Y Ξ + handFB s vL Y' Ξ

theorem **handFB_right** {Y : GA D J.W₀ → WO J.W₀ B.L → ℂ} {Ξ Ξ' : WO J.W₀ B.V₀ → ℂ} (hY : IsQuasiFam ψ J.W₀ B.L Y) (hΞ : IsThetaFunO ψ J.W₀ B.V₀ Ξ) (hΞ' : IsThetaFunO ψ J.W₀ B.V₀ Ξ') (c : ℂ) : handFB s vL Y (c • Ξ + Ξ') = c • handFB s vL Y Ξ + handFB s vL Y Ξ'

theorem **handFB_equiv** {Y : GA D J.W₀ → WO J.W₀ B.L → ℂ} {Ξ : WO J.W₀ B.V₀ → ℂ} (hY : IsQuasiFam ψ J.W₀ B.L Y) (hΞ : IsThetaFunO ψ J.W₀ B.V₀ Ξ) (g : GA D J.W₀) (h : GA D B.A) : handFB s vL (fun x => Y (x ∗ g)) (s.om₀ (g, B.o.emb (h, 1)) Ξ) = fun x => handFB s vL Y Ξ (x ∗ h)

theorem **quasi_handBF** (hφ : IsContAut B.A φ) (h₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) : IsQuasiFam ψ J.W₀ B'.L (handBF s₁ l φ Ξ₁)

theorem **contAut_hand0** (hφ : IsContAut B.A φ) (h₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) (h₂ : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂) : IsContAut B'.A (hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂)

theorem **doubleThetaUnfolding_linear_function** (hφ : IsContAut B.A φ) (hφ' : IsContAut B.A φ') (h₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) (h₂ : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂) (c : ℂ) : hand0 s₁ s₂ l vL (c • φ + φ') Ξ₁ Ξ₂ = c • hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂ + hand0 s₁ s₂ l vL φ' Ξ₁ Ξ₂

theorem **doubleThetaUnfolding_linear_first_datum** (hφ : IsContAut B.A φ) (h₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) (h₁' : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁') (h₂ : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂) (c : ℂ) : hand0 s₁ s₂ l vL φ (c • Ξ₁ + Ξ₁') Ξ₂ = c • hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂ + hand0 s₁ s₂ l vL φ Ξ₁' Ξ₂

theorem **doubleThetaUnfolding_linear_second_datum** (hφ : IsContAut B.A φ) (h₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) (h₂ : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂) (h₂' : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂') (c : ℂ) : hand0 s₁ s₂ l vL φ Ξ₁ (c • Ξ₂ + Ξ₂') = c • hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂ + hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂'

theorem **doubleThetaUnfolding_equivariant_unitaryGroups** (hφ : IsContAut B.A φ) (h₁ : IsThetaFunO ψ B.V₀ J.W₀ Ξ₁) (h₂ : IsThetaFunO ψ J.W₀ B'.V₀ Ξ₂) (a : GA D B.A) (g : GA D J.W₀) (h : GA D B'.A) : hand0 s₁ s₂ l vL (fun x => φ (x ∗ a)) (s₁.om₀ (B.o.emb (a, 1), g) Ξ₁) (s₂.om₀ (g, B'.o.emb (h, 1)) Ξ₂) = fun x => hand0 s₁ s₂ l vL φ Ξ₁ Ξ₂ (x ∗ h)

theorem **eq_of_map_pr** {S : HSp D} {g g' : GA D S} (hg : SuppAt p.e S g) (hg' : SuppAt p.e S g') (h : (mt S g).map p.pr = (mt S g').map p.pr) : g = g'

theorem **suppAt_emb** {S₀ A L : HSp D} (o : OrthDec S₀ A L) {e : AE D} {a : GA D A} (ha : SuppAt e A a) : SuppAt e S₀ (o.emb (a, 1))

theorem **emb_vecB** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) (a : GA D A) : mt S₀ (o.emb (a, 1)) ∗ᵥ ratV (vecB o ζ) = ratV (vecB o ζ)

theorem **block_calc** {R K m r n : Type} [CommRing R] [CommRing K] [Fintype m] [Fintype r] [Fintype n] [DecidableEq n] (f : R →+∗ K) (M : Matrix m m R) (N : Matrix r r R) (j : Matrix m r R) (z : m → R) (b : Matrix r n K) (b' : Matrix n r K) (g : Matrix n n K) (h1 : M ∗ j = j ∗ N) (h2 : M ∗ᵥ z = z) (h3 : N.map f = b ∗ g ∗ b') (h4 : b' ∗ b = 1) : M.map f ∗ fromCols (j.map f ∗ b) (replicateCol (Fin 1) fun i => f (z i)) = fromCols (j.map f ∗ b) (replicateCol (Fin 1) fun i => f (z i)) ∗ (fromBlocks g 0 0 1 : Matrix (n ⊕ Fin 1) (n ⊕ Fin 1) K)

theorem **emb_gl** {S₀ A L : HSp D} {o : OrthDec S₀ A L} {ζ : Fin L.rank → D.host.E} {n : Type} [Fintype n] [DecidableEq n] {β : Frame p A n} {β₀ : Frame p S₀ (n ⊕ Fin 1)} (h : IsSumFrame o ζ β β₀) (g : GL n k) : o.emb (β.gl g, 1) = β₀.gl (RootEx.embX 1 g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleUnfoldingStepAtGluedFinitePlace

theorem **exists_trilinear** {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) {M₁ M₂ : Type} [AddCommGroup M₁] [Module ℂ M₁] [AddCommGroup M₂] [Module ℂ M₂] (U₁ : M₁ →ₗ[ℂ] WO B.V₀ J.W₀ → ℂ) (U₂ : M₂ →ₗ[ℂ] WO J.W₀ B'.V₀ → ℂ) (h₁ : ∀ x, IsThetaFunO ψ B.V₀ J.W₀ (U₁ x)) (h₂ : ∀ y, IsThetaFunO ψ J.W₀ B'.V₀ (U₂ y)) : ∃ θ : caSub B.A →ₗ[ℂ] M₁ →ₗ[ℂ] M₂ →ₗ[ℂ] caSub B'.A, ∀ v x y, (θ v x y).1 = hand0 s₁ s₂ l vL v.1 (U₁ x) (U₂ y)

theorem **handFB_smul'** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (c : ℂ) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (Ξ : WO J.W₀ B.V₀ → ℂ) : handFB s vL (c • Y) Ξ = star c • handFB s vL Y Ξ

theorem **handFB_zero** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (Ξ : WO J.W₀ B.V₀ → ℂ) : handFB s vL 0 Ξ = 0

theorem **handBF_sum** {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') {φ : GA D B.A → ℂ} (hφ : IsContAut B.A φ) {M : Type} [AddCommGroup M] [Module ℂ M] (U : M →ₗ[ℂ] WO B.V₀ J.W₀ → ℂ) (hU : ∀ x, IsThetaFunO ψ B.V₀ J.W₀ (U x)) {ι : Type} (S : Finset ι) (x : ι → M) : ∑ i ∈ S, handBF s l φ (U (x i)) = handBF s l φ (U (∑ i ∈ S, x i))

theorem **lift_step** {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) {c₀ : ℂ} {φ₀ φ' : GA D B.A → ℂ} (hφ' : IsContAut B.A φ') {M : Type} [AddCommGroup M] [Module ℂ M] (U : M →ₗ[ℂ] WO B.V₀ J.W₀ → ℂ) (hU : ∀ x, IsThetaFunO ψ B.V₀ J.W₀ (U x)) {ι : Type} (S : Finset ι) (x : ι → M) (Ξ₁ : WO B.V₀ J.W₀ → ℂ) (Ξ₂ : WO J.W₀ B'.V₀ → ℂ) (h : liftA B.o s₁.om₀ B.vA φ₀ Ξ₁ = c₀ • ∑ i ∈ S, liftA B.o s₁.om₀ B.vA φ' (U (x i))) : hand0 s₁ s₂ l vL φ₀ Ξ₁ Ξ₂ = c₀ • hand0 s₁ s₂ l vL φ' (U (∑ i ∈ S, x i)) Ξ₂

theorem **charInd_mem'** (χ : AddChar 𝓀[k] ℂ) (Q : Pair k n) : charInd χ Q ∈ SB (n → k)

theorem **inDual_vecMul** {L : Submodule 𝒪[k] (n → k)} {g : GL n k} (hg : g • L = L) (η : n → k) : InDual L (η ᵥ* g.val) ↔ InDual L η

theorem **dualInd_vecMul** {L : Submodule 𝒪[k] (n → k)} {g : GL n k} (hg : g • L = L) (η : n → k) : dualInd L (η ᵥ* g.val) = dualInd L η

theorem **vecInd_mulVec** {L : Submodule 𝒪[k] (n → k)} {g : GL n k} (hg : g • L = L) (y : n → k) : vecInd L (g.val *ᵥ y) = vecInd L y

theorem **hasVal_neg** {P : Pair k n} {x : n → k} {a : 𝓀[k]} (h : P.HasVal x a) : P.HasVal (-x) (-a)

theorem **neg_one_smul_pair** (P : Pair k n) : ((-1 : GL n k) • P) = ((-1 : 𝓀[k]ˣ) • P)

theorem **val_det_neg_one** : valuation k ((Matrix.GeneralLinearGroup.det (-1 : GL n k) : kˣ) : k) = 1

theorem **val_FFStar'** {S : HSp D} (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (φ : Form k n (caSub S)) (P : Pair k n) : (FFStar q χ φ P).1 = FFStar q χ (fun Q => (φ Q).1) P

theorem **det_embX_one'** (g : GL n k) : Matrix.GeneralLinearGroup.det (GlobalTheory.RootEx.embX 1 g) = Matrix.GeneralLinearGroup.det g

theorem **overMap** (hψ : IsStdChar D ψ (-1)) {α : Frame p S n'} {β : Frame p T n} {LS : Lat S} {LT : Lat T} {f₀ : WO S T → ℂ} {Ψ₀ : Matrix n n' k → ℂ} {Ξ : WO S T → ℂ} (h : Over ψ α β LS LT f₀ Ψ₀ Ξ) (hlin : ∀ (Ψ : Matrix n n' k → ℂ) (f : WO S T → ℂ), Over ψ α β LS LT f₀ Ψ f → ∀ (c : ℂ) (Ψ' : Matrix n n' k → ℂ) (f' : WO S T → ℂ), Over ψ α β LS LT f₀ Ψ' f' → Over ψ α β LS LT f₀ (c • Ψ + Ψ') (c • f + f')) : ∃ U : SB (Matrix n n' k) →ₗ[ℂ] WO S T → ℂ, (∀ Ψ, Over ψ α β LS LT f₀ Ψ.1 (U Ψ)) ∧ (∀ Ψ f, Over ψ α β LS LT f₀ Ψ.1 f → f = U Ψ) ∧ ∀ Ψ, IsThetaFunO ψ S T (U Ψ)

theorem **tensCol_subst'** {Ψ Ψ' : Matrix n n' k → ℂ} (a : (n → k) → ℂ) (u : Matrix n n k) (g : GL n' k) (h : ∀ x, Ψ' x = Ψ (u * x * g.val)) (x : Matrix n (n' ⊕ Fin 1) k) : tensCol Ψ a (u * x * (GlobalTheory.RootEx.embX 1 g).val) = tensCol Ψ' (fun y => a (u *ᵥ y)) x

theorem **tensRow_subst'** {Ψ Ψ' : Matrix n n' k → ℂ} (b : (n' → k) → ℂ) (h₀ : GL n k) (u : Matrix n' n' k) (h : ∀ x, Ψ' x = Ψ ((h₀⁻¹ : GL n k).val * x * u)) (x : Matrix (n ⊕ Fin 1) n' k) : tensRow Ψ b (((GlobalTheory.RootEx.embX 1 h₀)⁻¹ : GL (n ⊕ Fin 1) k).val * x * u) = tensRow Ψ' (fun η => b (η ᵥ* u)) x

theorem **vertexGlued** (D : D7) : VertexGlued D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubleUnfoldingStepAtGluedFinitePlaceStatement

theorem **_root_.GlobalTheory.T.doubleUnfoldingStep_at_glued_finite_place** : ∀ D : D7, GlobalTheory.FinPl.VertexGlued D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledBasisSkewColumnMatrixGramInverse

theorem **gram_fromCols_smul** (G : Matrix (Fin (n + n)) (Fin (n + n)) E) (e e' : Matrix (Fin (n + n)) (Fin n) E) (μ : E) (hμ : star μ = -μ) (h₁ : eᴴ * G * e = 0) (h₂ : e'ᴴ * G * e' = 0) (h₃ : e'ᴴ * G * e = 1) (h₄ : eᴴ * G * e' = 1) : ((Matrix.fromCols e (μ • e')).submatrix id finSumFinEquiv.symm)ᴴ * G * (Matrix.fromCols e (μ • e')).submatrix id finSumFinEquiv.symm = μ • blk4 0 1 (-1) 0

theorem **fromRows_mul_fromCols_smul** (G : Matrix (Fin (n + n)) (Fin (n + n)) E) (e e' : Matrix (Fin (n + n)) (Fin n) E) (μ : E) (hμ : μ ≠ 0) (h₁ : eᴴ * G * e = 0) (h₂ : e'ᴴ * G * e' = 0) (h₃ : e'ᴴ * G * e = 1) (h₄ : eᴴ * G * e' = 1) : (Matrix.fromRows (e'ᴴ * G) (μ⁻¹ • (eᴴ * G))).submatrix finSumFinEquiv.symm id * (Matrix.fromCols e (μ • e')).submatrix id finSumFinEquiv.symm = (1 : Matrix (Fin (n + n)) (Fin (n + n)) E)

theorem **Tm_gram_inv** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → (Tm A μ)ᴴ * (dblSp A).gram * Tm A μ = μ • Jst D A.rank ∧ IsUnit (Tm A μ).det ∧ (Tm A μ)⁻¹ = (Matrix.fromRows (dblDec A).qX (μ⁻¹ • (dblDec A).qY)).submatrix finSumFinEquiv.symm id

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledBasisSkewColumnMatrixGramInverseStatement

theorem **_root_.GlobalTheory.T.doubledBasisSkewColumnMatrix_gram_inverse** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → (Tm A μ)ᴴ * (dblSp A).gram * Tm A μ = μ • Jst D A.rank ∧ IsUnit (Tm A μ).det ∧ (Tm A μ)⁻¹ = (Matrix.fromRows (dblDec A).qX (μ⁻¹ • (dblDec A).qY)).submatrix finSumFinEquiv.symm id

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledCartanInvolutionCommuteMaximalCompact

theorem **herm_of_isCartanInv** (hs : IsCartanInv G s) : Gᴴ = G

theorem **frame_gram_eq** (hbc : b * c = 1) (hs : s * s = 1) (h : bᴴ * (G * s) * b = 1) : bᴴ * G * b = c * s * b

theorem **frame_gram_sq** (hbc : b * c = 1) (hs : s * s = 1) (h : bᴴ * (G * s) * b = 1) : (bᴴ * G * b) * (bᴴ * G * b) = 1

theorem **frame_inv_eq** (hbc : b * c = 1) (hs : s * s = 1) (h : bᴴ * (G * s) * b = 1) : b * bᴴ * G = s

theorem **blkMat_conjTranspose** (x y : Matrix (Fin n) (Fin n) ℂ) : (blkMat finSumFinEquiv x y)ᴴ = blkMat finSumFinEquiv xᴴ yᴴ

theorem **compAt_blkGL** (τ : D.what₂.Tau) {n : ℕ} (a : GL (Fin n) (AE D)) : compAt θ τ (blkGL finSumFinEquiv (a, a)).1 = blkMat finSumFinEquiv (compAt θ τ a.1) (compAt θ τ a.1)

theorem **commute_of_mem_Kmax** (hθ : IsInfFam θ) {V : HSp D} {a : GL (Fin V.rank) (AE D)} (τ : D.what₂.Tau) (hd : ((compAt θ τ a.1)ᴴ * gramAt D V τ * compAt θ τ a.1) * ((compAt θ τ a.1)ᴴ * gramAt D V τ * compAt θ τ a.1) = 1) {g : GA D V} (hg : g ∈ Kmax V a) : Commute (Formalism.MainNotions.StandardLattice.infMat D.sp.cm.sig V.gram (θ τ) g) (frameInv θ V a τ)

theorem **cartan_dbl** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (sA : RealMat A) (a : GL (Fin A.rank) (AE D)), IsInfFam θ → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, (compAt θ τ a.1)ᴴ * (gramAt D A τ * sA τ) * compAt θ τ a.1 = 1) → (∀ τ, IsCartanInv (gramAt D (dblSp A) τ) ((Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm)) ∧ ∀ k ∈ Khat A a, ∀ τ, compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k) * (Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm = (Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm * compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledCartanInvolutionCommuteMaximalCompactStatement

theorem **_root_.GlobalTheory.T.doubled_cartanInvolution_commute_maximalCompact** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (sA : RealMat A) (a : GL (Fin A.rank) (AE D)), IsInfFam θ → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, (compAt θ τ a.1)ᴴ * (gramAt D A τ * sA τ) * compAt θ τ a.1 = 1) → (∀ τ, IsCartanInv (gramAt D (dblSp A) τ) ((Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm)) ∧ ∀ k ∈ Khat A a, ∀ τ, compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k) * (Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm = (Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm * compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledCompactSubgroupConjHermitianSpace

theorem **stab_conj_iff** {R ι : Type} [CommRing R] [Fintype ι] [DecidableEq ι] (M M' : Set (ι → R)) (B X : GL ι R) (hB : ∀ Y, Y ∈ M ↔ B.1 *ᵥ Y ∈ M') : (∀ Y, Y ∈ M' ↔ X.1 *ᵥ Y ∈ M') ↔ (∀ Y, Y ∈ M ↔ (B⁻¹ * X * B).1 *ᵥ Y ∈ M)

theorem **map_conj_of_eq_one** {G H : Type} [Group G] [Group H] (f : G →* H) {g : G} (hg : f g = 1) (x : G) : f (g⁻¹ * x * g) = f x

theorem **map_conj'** {G H : Type} [Group G] [Group H] (f : G →* H) (g x : G) : f (g⁻¹ * x * g) = (f g)⁻¹ * f x * f g

theorem **mem_KFin_conj** (h : Q = dblGram finSumFinEquiv J) {L L' : Submodule (𝒪 E) (Fin n → E)} {v : HeightOneSpectrum (𝒪 F)} (g : GLoc σ J v) (hg : ∀ z, z ∈ latLoc F L v ↔ (GlobalTheory.AdelicU.matHom _ σ J g).val *ᵥ z ∈ latLoc F L' v) (x : GLoc σ Q v) : x ∈ GlobalTheory.Doubling.KFin σ Q L' v ↔ (ioH _ σ h (g, g))⁻¹ * x * ioH _ σ h (g, g) ∈ GlobalTheory.Doubling.KFin σ Q L v

theorem **mem_KAd_conj** (h : Q = dblGram finSumFinEquiv J) {L L' : Submodule (𝒪 E) (Fin n → E)} (k : GAd σ J) (hk : k ∈ finPts σ J) (hL : ∀ (v : HeightOneSpectrum (𝒪 F)) z, z ∈ latLoc F L v ↔ (GlobalTheory.AdelicU.matHom _ σ J (proj σ J v k)).val *ᵥ z ∈ latLoc F L' v) (a : GL (Fin n) (AF F ⊗[F] E)) (x : GAd σ Q) : x ∈ KAd σ Q L' a ↔ (ioH _ σ h (k, k))⁻¹ * x * ioH _ σ h (k, k) ∈ KAd σ Q L a

theorem **mem_KAd_io_conj** : ∀ (D : D7) (A : HSp D) (L L' : Submodule (𝒪 D.host.E) (Fin A.rank → D.host.E)) (k : GA D A), k ∈ GfC A → (∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (z : Fin A.rank → TensorProduct D.host.F (Fv D.host.F v) D.host.E), z ∈ latLoc D.host.F L v ↔ (GlobalTheory.AdelicU.matHom _ D.sp.cm.sig A.gram (proj D.sp.cm.sig A.gram v k)).val *ᵥ z ∈ latLoc D.host.F L' v) → ∀ (a : GL (Fin A.rank) (AE D)) (x : GA D (dblSp A)), x ∈ KAd D.sp.cm.sig (dblSp A).gram L' a ↔ (io A (k, k))⁻¹ * x * io A (k, k) ∈ KAd D.sp.cm.sig (dblSp A).gram L a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledCompactSubgroupConjHermitianSpaceStatement

theorem **_root_.GlobalTheory.T.doubledCompactSubgroup_conj_hermitianSpace** : ∀ (D : D7) (A : HSp D) (L L' : Submodule (𝒪 D.host.E) (Fin A.rank → D.host.E)) (k : GA D A), k ∈ GfC A → (∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (z : Fin A.rank → TensorProduct D.host.F (Fv D.host.F v) D.host.E), z ∈ latLoc D.host.F L v ↔ (GlobalTheory.AdelicU.matHom _ D.sp.cm.sig A.gram (proj D.sp.cm.sig A.gram v k)).val *ᵥ z ∈ latLoc D.host.F L' v) → ∀ (a : GL (Fin A.rank) (AE D)) (x : GA D (dblSp A)), x ∈ KAd D.sp.cm.sig (dblSp A).gram L' a ↔ (io A (k, k))⁻¹ * x * io A (k, k) ∈ KAd D.sp.cm.sig (dblSp A).gram L a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledFlatSectionEqTranslateStandardSection

theorem **flat_swSec_eq_sect** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ θ : InfFam D, IsInfFam θ → ∀ μ : D.host.E, IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∀ omS : G D A.rank →* Module.End ℂ (schwartzA θ B.rank A.rank), IsTransport ψ θ A B ν 𝒴 omh κ omS → ∀ lam : D.host.F, algebraMap D.host.F D.host.E lam = -D.host.xi0 / μ → ∀ (a : GL (Fin A.rank) (AE D)) (p : siegel A) (f : WO (dblSp A) B → ℂ), IsThetaFunO ψ (dblSp A) B f → ∀ Φ : schwartzA θ B.rank A.rank, Φ.1 = sch ψ A B ν 𝒴 (omh ((p : GA D (dblSp A))⁻¹, 1) f) → s0 B A.rank = sZero A B ∧ (∀ (z : ℂ) (g : GA D (dblSp A)), flat A a (swSec A B omh ν 𝒴 f) (sZero A B) z g = npow (absA D (aP A p)) (sZero

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledFlatSectionEqTranslateStandardSectionStatement

theorem **_root_.GlobalTheory.T.doubledFlatSection_eq_translate_standardSection** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ θ : InfFam D, IsInfFam θ → ∀ μ : D.host.E, IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∀ omS : G D A.rank →* Module.End ℂ (schwartzA θ B.rank A.rank), IsTransport ψ θ A B ν 𝒴 omh κ omS → ∀ lam : D.host.F, algebraMap D.host.F D.host.E lam = -D.host.xi0 / μ → ∀ (a : GL (Fin A.rank) (AE D)) (p : siegel A) (f : WO (dblSp A) B → ℂ), IsThetaFunO ψ (dblSp A) B f → ∀ Φ : schwartzA θ B.rank A.rank, Φ.1 = sch ψ A B ν 𝒴 (omh ((p : GA D (dblSp A))⁻¹, 1) f) → s0 B A.rank = sZero A B ∧ (∀ (z : ℂ) (g : GA D (dblSp A)), flat A a (swSec A B omh ν 𝒴

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledLagrangianTransverseInOpenOrbit

theorem **exists_mem_siegel_of_diagSnd** : ∀ (D : D7) (V : HSp D), ∀ γ ∈ GQ (dblSp V), ∀ γi : Matrix (Fin (V.rank + V.rank)) (Fin (V.rank + V.rank)) D.host.E, (matHom (dblSp V) γ⁻¹).1 = γi.map (incA D) → (∀ x : Fin V.rank → D.host.E, diagSnd γi *ᵥ x = 0 → x = 0) → ∃ δ ∈ GQ V, γ * (io V (δ, 1))⁻¹ ∈ siegel V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledLagrangianTransverseInOpenOrbitStatement

theorem **_root_.GlobalTheory.T.doubled_lagrangian_transverse_in_open_orbit** : ∀ (D : D7) (V : HSp D), ∀ γ ∈ GQ (dblSp V), ∀ γi : Matrix (Fin (V.rank + V.rank)) (Fin (V.rank + V.rank)) D.host.E, (matHom (dblSp V) γ⁻¹).1 = γi.map (incA D) → (∀ x : Fin V.rank → D.host.E, diagSnd γi *ᵥ x = 0 → x = 0) → ∃ δ ∈ GQ V, γ * (io V (δ, 1))⁻¹ ∈ siegel V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledLatticeHeightFunProperties

theorem **hgtL_eq_prod** (hL : IsSelfDualC A L) (ha : IsMaj A a) (g : GA D A) : hgtL A L a g = (∏ w, hgtArch D.sp.cm.sig (dblSp A).gram w (Formalism.UG.bc (compInf _ _ w) a) (ioH _ D.sp.cm.sig (gram_hdbl A) (projArch D.sp.cm.sig A.gram w g, 1))) * ∏ᶠ v, hgtFin D.sp.cm.sig (dblSp A).gram L v (ioH _ D.sp.cm.sig (gram_hdbl A) (proj D.sp.cm.sig A.gram v g, 1))

theorem **isOpen_openOff** (hL : IsSelfDualC A L) (Sf : Finset (FinPl D)) : IsOpen (openOff A L Sf)

theorem **continuous_hgtL** (hL : IsSelfDualC A L) (ha : IsMaj A a) : Continuous (hgtL A L a)

theorem **hgtL_le_one** (hL : IsSelfDualC A L) (ha : IsMaj A a) (g : GA D A) : hgtL A L a g ≤ 1

theorem **hgtL_eq_prod_off** (hL : IsSelfDualC A L) (ha : IsMaj A a) (Sf : Finset (FinPl D)) (g : GA D A) (hg : g ∈ openOff A L Sf) : (hgtL A L a g : ℝ) = (∏ w, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW A a w (projArch D.sp.cm.sig A.gram w g)) * ∏ v ∈ Sf, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV A L v (proj D.sp.cm.sig A.gram v g)

theorem **hgtL_props** : ∀ (D : D7) (A : HSp D) (L : Lat A) (a : GL (Fin A.rank) (AE D)), IsSelfDualC A L → IsMaj A a → Continuous (hgtL A L a) ∧ (∀ g, 0 < hgtL A L a g ∧ hgtL A L a g ≤ 1) ∧ ∀ (Sf : Finset (FinPl D)), IsOpen (openOff A L Sf) ∧ ∀ g ∈ openOff A L Sf, (hgtL A

L a g : ℝ) = (∏ w, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW A a w (projArch D.sp.cm.sig A.gram w g)) * ∏ v ∈ Sf, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV A L v (proj D.sp.cm.sig A.gram v g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledLatticeHeightFunPropertiesStatement

theorem **_root_.GlobalTheory.T.doubledLatticeHeightFun_properties** : ∀ (D : D7) (A : HSp D) (L : Lat A) (a : GL (Fin A.rank) (AE D)), IsSelfDualC A L → IsMaj A a → Continuous (hgtL A L a) ∧ (∀ g, 0 < hgtL A L a g ∧ hgtL A L a g ≤ 1) ∧ ∀ (Sf : Finset (FinPl D)), IsOpen (openOff A L Sf) ∧ ∀ g ∈ openOff A L Sf, (hgtL A L a g : ℝ) = (∏ w, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW A a w (projArch D.sp.cm.sig A.gram w g)) * ∏ v ∈ Sf, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV A L v (proj D.sp.cm.sig A.gram v g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledLatticeHeightSelfDualLatticesEqConj

theorem **hgtL_eq_conj_of_selfDual** : ∀ (D : D7) (A : HSp D) (L L' : Lat A), IsSelfDualC A L → IsSelfDualC A L' → ∃ k ∈ GfC A, ∀ (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ g, hgtL A L' a g = hgtL A L a (k⁻¹ * g * k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledLatticeHeightSelfDualLatticesEqConjStatement

theorem **_root_.GlobalTheory.T.doubledLatticeHeight_selfDual_lattices_eq_conj** : ∀ (D : D7) (A : HSp D) (L L' : Lat A), IsSelfDualC A L → IsSelfDualC A L' → ∃ k ∈ GfC A, ∀ (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ g, hgtL A L' a g = hgtL A L a (k⁻¹ * g * k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledPairSchrodingerFunctionContinuousIntegrab

theorem **mixS_continuous_integrable** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (ν : GlobalTheory.Unf.VolM D T.rank ρ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S₀ T) : Continuous (fun x => d.mixS T ψ ν.μ ν.𝓕 f x w) ∧ Integrable (fun x => d.mixS T ψ ν.μ ν.𝓕 f x w) ν.μ

theorem **sch_continuous_integrable** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → Continuous (sch ψ A B ν 𝒴 f) ∧ Integrable (sch ψ A B ν 𝒴 f) ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledPairSchrodingerFunctionContinuousIntegrabStatement

theorem **_root_.GlobalTheory.T.doubledPairSchrodingerFunction_continuous_integrable** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → Continuous (sch ψ A B ν 𝒴 f) ∧ Integrable (sch ψ A B ν 𝒴 f) ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledPairThetaFunctionEqZeroOfSchrodingerEqZer

theorem **sch_heisO_extY** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (A B : HSp D) {ν : Measure (WO A B)} {𝒴 : Set (WO A B)} (hν : IsCovolOne (ratL B A) ν 𝒴) {f : WO (dblSp A) B → ℂ} (hf : IsThetaFunO ψ (dblSp A) B f) (x y : Mx D B.rank A.rank) : sch ψ A B ν 𝒴 (heisO ψ (dblSp A) B ((dblDec A).extY B y) f) x = ψ (symplO (dblSp A) B ((dblDec A).extX B x) ((dblDec A).extY B y)) * sch ψ A B ν 𝒴 f x

theorem **exists_extX_add_extY** {D : D7} (A B : HSp D) (w : WO (dblSp A) B) : ∃ x y : Mx D B.rank A.rank, w = (dblDec A).extX B x + (dblDec A).extY B y

theorem **eq_zero_of_sch_eq_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → sch ψ A B ν 𝒴 f = 0 → f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledPairThetaFunctionEqZeroOfSchrodingerEqZerStatement

theorem **_root_.GlobalTheory.T.doubledPairThetaFunction_eq_zero_of_schrodinger_eq_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → sch ψ A B ν 𝒴 f = 0 → f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledSpaceMaximalLevelLattice

theorem **jX_dblDec_mulVec** (V : HSp D) (y : Fin V.rank → AE D) : (dblDec V).jX *ᵥ y = pairVec finSumFinEquiv (GlobalTheory.MixModel.giA V *ᵥ y) (GlobalTheory.MixModel.giA V *ᵥ y)

theorem **mem_siegel_of_mem_stab** (V : HSp D) {p : GA D (dblSp V)} (h : p ∈ stab (dblDec V)) : p ∈ siegel V

theorem **mul_jX_dblDec** (V : HSp D) {p : GA D (dblSp V)} (h : p ∈ siegel V) : mt (dblSp V) p · (dblDec V).jX = (dblDec V).jX · (GlobalTheory.MixModel.gA V * leviMat finSumFinEquiv (mt (dblSp V) p) * GlobalTheory.MixModel.giA V)

theorem **mem_stab_of_mem_siegel** (V : HSp D) {p : GA D (dblSp V)} (h : p ∈ siegel V) : p ∈ stab (dblDec V)

theorem **onX_dblDec** (V : HSp D) {p : GA D (dblSp V)} (h : p ∈ siegel V) (h' : p ∈ stab (dblDec V)) : onX (dblDec V) ⟨p, h'⟩ = gramGL V * leviOf finSumFinEquiv (Formalism.L2.Theta.matHom (dblSp V)) ⟨p, h⟩ * (gramGL V)⁻¹

theorem **isAniso_zeroSp** (D : D7) : IsAniso (zeroSp D)

theorem **siegel_eq_stab_dblDec** : ∀ (D : D7) (V : HSp D), siegel V = stab (dblDec V) ∧ (∀ (p : GA D (dblSp V)) (h : p ∈ siegel V) (h' : p ∈ stab (dblDec V)), aP V ⟨p, h⟩ = Matrix.GeneralLinearGroup.det (onX (dblDec V) ⟨p, h'⟩)) ∧ IsAniso (zeroSp D) ∧ ∀ a : GL (Fin V.rank) (AE D), IsMaj V a → ∃ (L : Submodule (𝒪 D.host.E) (Fin (dblSp V).rank → D.host.E)) (b : GL (Fin (dblSp V).rank) (AE D)), IsLevel (dblSp V) L b ∧ Khat V a = lev (dblSp V) L b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledSpaceMaximalLevelLatticeStatement

theorem **_root_.GlobalTheory.T.doubled_space_maximal_level_lattice** : ∀ (D : D7) (V : HSp D), siegel V = stab (dblDec V) ∧ (∀ (p : GA D (dblSp V)) (h : p ∈ siegel V) (h' : p ∈ stab (dblDec V)), aP V ⟨p, h⟩ = Matrix.GeneralLinearGroup.det (onX (dblDec V) ⟨p, h'⟩)) ∧ IsAniso (zeroSp D) ∧ ∀ a : GL (Fin V.rank) (AE D), IsMaj V a → ∃ (L : Submodule (𝒪 D.host.E) (Fin (dblSp V).rank → D.host.E)) (b : GL (Fin (dblSp V).rank) (AE D)), IsLevel (dblSp V) L b ∧ Khat V a = lev (dblSp V) L b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledThetaEqSumSchrodingerFunctionPeriodisatio

theorem **theta_eq_sum_sch** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → ∀ x : Mx D B.rank A.rank, (Summable fun ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E => ‖sch ψ A B ν 𝒴 f (x + rat ξ)‖) ∧ HasSum (fun ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E => sch ψ A B ν 𝒴 f (x + rat ξ)) (f ((dblDec A).extX B x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledThetaEqSumSchrodingerFunctionPeriodisatioStatement

theorem **_root_.GlobalTheory.T.doubledTheta_eq_sum_schrodingerFunction_periodisation** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → ∀ x : Mx D B.rank A.rank, (Summable fun ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E => ‖sch ψ A B ν 𝒴 f (x + rat ξ)‖) ∧ HasSum (fun ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E => sch ψ A B ν 𝒴 f (x + rat ξ)) (f ((dblDec A).extX B x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledThetaSeriesLawAndLevel

theorem **bt_add_left** (x x' y : Mx D T.rank ρ) : bt d T (x + x') y = bt d T x y + bt d T x' y

theorem **bt_add_right** (x y y' : Mx D T.rank ρ) : bt d T x (y + y') = bt d T x y + bt d T x y'

theorem **bt_zero_right** (x : Mx D T.rank ρ) : bt d T x 0 = 0

theorem **symplO_bt** (x y : Mx D T.rank ρ) : symplO S T (d.extX T x) (d.extY T y) = bt d T x y + bt d T x y

theorem **psi_bt_rat** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) (ξ η : Matrix (Fin T.rank) (Fin ρ) D.host.E) : ψ (bt d T (rat ξ) (rat η)) = 1

theorem **symplOH_coord** (h0 : d.j0 = 0) (w w' : WO S T) : symplOH S T w w' = bt d T (w · d.jX) (w' · d.jY) - bt d T (w' · d.jX) (w · d.jY)

theorem **map_mul_zero** {R : Type} [CommRing R] (q : AE D →+* R) {l m n : ℕ} {x : Mx D l m} (hx : x.map q = 0) (y : Mx D m n) : (x · y : Mx D l n).map q = 0

theorem **snd_bt** {x : Mx D T.rank ρ} (hx : x.map (finPart D) = 0) (y : Mx D T.rank ρ) : (bt d T x y).2 = 0

theorem **bt_eq_pair** (x : Mx D T.rank ρ) {y : Mx D T.rank ρ} (hy : y.map (infPart D) = 0) : bt d T x y = pairA D.host.F 0 (bt d T x y).2

theorem **exists_section** (D : D7) : ∃ s : Af D.host.F ⊗[D.host.F] D.host.E → AE D, Continuous s ∧ ∀ z, finPart D (s z) = z

theorem **exists_open_ann** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) (Z : Set (Mx D T.rank ρ)) {K : Set (Matrix (Fin T.rank) (Fin ρ) (Af D.host.F ⊗[D.host.F] D.host.E))} (hK : IsCompact K) (hZ : ∀ z ∈ Z, z.map (finPart D) ∈ K) : ∃ V : AddSubgroup (Mx D T.rank ρ), IsOpen (V : Set (Mx D T.rank ρ)) ∧ ∀ b ∈ V, b.map (infPart D) = 0 → ∀ z ∈ Z, ψ (bt d T z b) = 1

theorem **addChar_theta_law_rearrangement_identity** {R : Type} [AddCommGroup R] (χ : AddChar R ℂ) (a b c e g h : R) (t : ℂ) (ha : χ a = 1) (hg : χ g = 1) : χ (-(a + b + (c + e))) * (χ (c + e + (g + h) + (c + e + (g + h))) * t) = χ (-(b - c)) * χ (-e) * (χ (e + h + (e + h)) * t)

theorem **law** (hψ : IsCharAF D.host.F ψ) (h0 : d.j0 = 0) (hf : ∀ w : WO S T, f w = ψ (-(bt d T (w · d.jX) (w · d.jY))) * ∑' ξ : Matrix (Fin T.rank) (Fin ρ) D.host.E, ψ (symplO S T (d.extX T (w · d.jX + rat ξ)) (d.extY T (w · d.jY))) * φ (w · d.jX + rat ξ)) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (w : WO S T) : f (rat γ + w) = ψ (-(symplOH S T (rat γ) w)) * f w

theorem **level** (hψ : IsCharAF D.host.F ψ) (h0 : d.j0 = 0) (hφ : GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ) (hf : ∀ w : WO S T, f w = ψ (-(bt d T (w · d.jX) (w · d.jY))) * ∑' ξ : Matrix (Fin T.rank) (Fin ρ) D.host.E, ψ (symplO S T (d.extX T (w · d.jX + rat ξ)) (d.extY T (w · d.jY))) * φ (w · d.jX + rat ξ)) : ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u f = f

theorem **law_and_level** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (φ : Mx D B.rank A.rank → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → ∀ f : WO (dblSp A) B → ℂ, (∀ w : WO (dblSp A) B, f w = ψ (-(symplOH (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY)))) * ∑' ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E, ψ (symplO (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX + rat ξ)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY))) * φ (HMul.hMul w (dblDec A).jX + rat ξ)) → (∀ γ w, f (ratW B (dblSp A) γ + w) = ψ (-(symplOH (dblSp A) B (ratW B (dblSp A) γ) w)) * f w) ∧ ∃ U : AddSubgroup (WO (dblSp A) B), IsOpen (U : Set (WO (dblSp A) B)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ (dblSp A) B u f = f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledThetaSeriesLawAndLevelStatement

theorem **_root_.GlobalTheory.T.doubled_theta_series_law_and_level** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (φ : Mx D B.rank A.rank → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → ∀ f : WO (dblSp A) B → ℂ, (∀ w : WO (dblSp A) B, f w = ψ (-(symplOH (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY)))) * ∑' ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E, ψ (symplO (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX + rat ξ)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY))) * φ (HMul.hMul w (dblDec A).jX + rat ξ)) → (∀ γ w, f (ratW B (dblSp A) γ + w) = ψ (-(symplOH (dblSp A) B (ratW B (dblSp A) γ) w)) * f w) ∧ ∃ U : AddSubgroup (WO (dblSp A) B), IsOpen (U : Set (WO (dblSp A) B)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ (dblSp A) B u f = f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledThetaVectorFixedUnramifiedStabilizer

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : WRep (dblSp A) B), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (LA : Lat A) (LB : Lat B) (Sf : Finset (FinPl D)) (Φ₁ Φ₂ : WO A B → ℂ), IsThetaFunO ψ A B Φ₁ → IsThetaFunO ψ A B Φ₂ → Unram ψ A B χB LA LB Sf Φ₁ Φ₂ → ∀ v ∉ Sf, ∀ k ∈ KFin D.sp.cm.sig (dblSp A).gram LA v, omh (emb D.sp.cm.sig (dblSp A).gram v k, 1) (delta A B Φ₁ Φ₂) = delta A B Φ₁ Φ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoubledThetaVectorFixedUnramifiedStabilizerStatement

theorem **_root_.GlobalTheory.T.doubledTheta_vector_fixed_unramified_stabilizer** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : WRep (dblSp A) B), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (LA : Lat A) (LB : Lat B) (Sf : Finset (FinPl D)) (Φ₁ Φ₂ : WO A B → ℂ), IsThetaFunO ψ A B Φ₁ → IsThetaFunO ψ A B Φ₂ → Unram ψ A B χB LA LB Sf Φ₁ Φ₂ → ∀ v ∉ Sf, ∀ k ∈ KFin D.sp.cm.sig (dblSp A).gram LA v, omh (emb D.sp.cm.sig (dblSp A).gram v k, 1) (delta A B Φ₁ Φ₂) = delta A B Φ₁ Φ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingBasicIdentityFlatSectionCuspForms

theorem **basicIdentity_flat** : GlobalTheory.Published.SiegelEisensteinContinuation → GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (V : HSp D), 0 < V.rank → ∀ a : GL (Fin V.rank) (AE D), IsMaj V a → HasIwasawa D.sp.cm.sig (dblSp V).gram (Khat V a) → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ (s₁ : ℂ) (φ : GA D (dblSp V) → ℂ), φ ∈ degPS V a μ s₁ → ∀ θ : GlobalTheory.Mem.Th.InfF D, IsInfFam θ → ∀ A : CptF V, ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ ∈ GlobalTheory.Cusp.A0 θ V A, ∀ f₂ ∈ GlobalTheory.Cusp.A0 θ V A, ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → GlobalTheory.Eis.BasicIdentity V μ m (flat V a φ s₁ s) f₁ f₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingBasicIdentityFlatSectionCuspFormsStatement

theorem **_root_.GlobalTheory.T.doublingBasicIdentity_flatSection_cuspForms** : GlobalTheory.Published.SiegelEisensteinContinuation → GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (V : HSp D), 0 < V.rank → ∀ a : GL (Fin V.rank) (AE D), IsMaj V a → HasIwasawa D.sp.cm.sig (dblSp V).gram (Khat V a) → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ (s₁ : ℂ) (φ : GA D (dblSp V) → ℂ), φ ∈ degPS V a

μ s₁ → ∀ θ : GlobalTheory.Mem.Th.InfF D, IsInfFam θ → ∀ A : CptF V, ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ ∈ GlobalTheory.Cusp.A0 θ V A, ∀ f₂ ∈ GlobalTheory.Cusp.A0 θ V A, ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → GlobalTheory.Eis.BasicIdentity V μ m (flat V a φ s₁ s) f₁ f₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingBasicIdentityUnfolding

theorem **measurable_tsum_of_summable'** {X I : Type} [MeasurableSpace X] [Countable I] (F : I → X → ℂ) (hF : ∀ i, Measurable (F i)) (hs : ∀ x, Summable fun i => F i x) : Measurable fun x => ∑' i, F i x

theorem **out_mul_inv_mem** {D : D7} (V : HSp D) (x : GQ (dblSp V)) : (((mk ((siegel V).subgroupOf (GQ (dblSp V))) x).out : GQ (dblSp V)) : GA D (dblSp V)) * ((x : GA D (dblSp V))⁻¹ ∈ siegel V

theorem **basicIdentity_unfold** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V (GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → IsCuspFun V f₁ → GlobalTheory.Eis.BasicIdentity V μ m φ f₁ f₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingBasicIdentityUnfoldingStatement

theorem **_root_.GlobalTheory.T.doubling_basicIdentity_unfolding** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V (GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → IsCuspFun V f₁ → GlobalTheory.Eis.BasicIdentity V μ m φ f₁ f₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingClassOffOpenOrbitFixedByUnipotentRadical

theorem **offOpen** : ∀ (D : D7) (V : HSp D), ∀ γ ∈ GQ (dblSp V), ¬ (∃ δ ∈ GQ V, γ * (io V (δ, 1))⁻¹ ∈ siegel V) → ∃ (r : ℕ) (C : Matrix (Fin V.rank) (Fin r) D.host.E), IsIsoFrame V C ∧ (∀ n ∈ NI V C, ∃ p : siegel V, γ * io V (n, 1) = p * γ ∧ aP V p = 1) ∧ ∀ δ ∈ GQ V, γ * io V (δ, 1) * γ⁻¹ ∈ siegel V → ∀ n ∈ NI V C, δ * n * δ⁻¹ ∈ NI V C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingClassOffOpenOrbitFixedByUnipotentRadicalStatement

theorem **_root_.GlobalTheory.T.doublingClass_offOpenOrbit_fixedByUnipotentRadical** : ∀ (D : D7) (V : HSp D), ∀ γ ∈ GQ (dblSp V), ¬ (∃ δ ∈ GQ V, γ * (io V (δ, 1))⁻¹ ∈ siegel V) → ∃ (r : ℕ) (C : Matrix (Fin V.rank) (Fin r) D.host.E), IsIsoFrame V C ∧ (∀ n ∈ NI V C, ∃ p : siegel V, γ * io V (n, 1) = p * γ ∧ aP V p = 1) ∧ ∀ δ ∈ GQ V, γ * io V (δ, 1) * γ⁻¹ ∈ siegel V → ∀ n ∈ NI V C, δ * n * δ⁻¹ ∈ NI V C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingEulerProductFarRight

theorem **limit_eq_prod_mul** {ι : Type} (S : Finset ι) (f : ι → ℂ) (I : Finset ι → ℂ) {P J : ℂ} (hP : HasProd (fun v : {v : ι // v ∉ (S : Set ι)} => f v.1) P) (hJ : Tendsto I atTop (𝓝 J)) (h : ∀ T U : Finset ι, Disjoint S T → (∀ u, u ∈ U ↔ u ∈ S ∨ u ∈ T) → I U = (∏ v ∈ T, f v) * I S) : J = P * I S

theorem **bounded_of_isParamFun** (hP : GlobalTheory.Published.CuspRapidDecay) {D : D7} (hB : ParamBoundOn D (fun _ => True)) {θ : InfFam D} {A : HSp D} {cA : CptF A} {L : Lat A} (hθ : IsInfFam θ) (hL : IsSelfDualC A L) (m : Measure (GA D A)) (hm : m.IsHaarMeasure) {Sf : Finset (FinPl D)} {u : GA D A → ℂ} (hu : u ∈ A0 θ A cA) {par : GlobalTheory.LFun.Fam D.host.E} (hpar : GlobalTheory.LFun.IsParamFun D.sp.cm.sig A.gram L (Sf : Set (FinPl D)) u par) : GlobalTheory.LFun.Bounded (GlobalTheory.LFun.above D.host.E (Sf : Set (FinPl D))) par A.rank ((A.rank : ℝ) / 2)

theorem **eulerRight** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, SphFactor D → LocalZetaOn D (fun _ => True) → ParamBoundOn D (fun _ => True) → EulerRight D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingEulerProductFarRightOfRapidDecayUnramifi

theorem **eulerRight_of_unramifiedDoubling** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.UnramifiedDoublingGL → ∀ D : D7, GlobalTheory.Core.Rk.IP.EulerRight D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingEulerProductFarRightOfRapidDecayUnramifiStatement

theorem **_root_.GlobalTheory.T.doublingEulerProductFarRight_of_rapidDecay_unramified** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.UnramifiedDoublingGL → ∀ D : D7, GlobalTheory.Core.Rk.IP.EulerRight D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingEulerProductFarRightStatement

theorem **_root_.GlobalTheory.T.doubling_eulerProduct_farRight** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, SphFactor D → LocalZetaOn D (fun _ => True) → ParamBoundOn D (fun _ => True) → EulerRight D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightAtRealPlaceContinuousPosLeOne

theorem **continuous_hgtW** (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)) (w : InfinitePlace D.host.F) (hI : GlobalTheory.Doubling.HasIwasawa D.sp.cm.sig (GlobalTheory.Doubling.dblSp V).gram (GlobalTheory.Doubling.KArch D.sp.cm.sig (GlobalTheory.Doubling.dblSp V).gram w (Formalism.UG.bc (GlobalTheory.AdelicU.compInf D.host.F D.host.E w) a))) : Continuous (GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w)

theorem **hgtW_pos** (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)) (w : InfinitePlace D.host.F) (g : GlobalTheory.Doubling.Gw V w) : 0 < GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w g

theorem **hgtW_cont_pos_le_one** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), GlobalTheory.Doubling.IsMaj V a → ∀ w, Continuous (GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w) ∧ ∀ g, 0 < GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w g ∧ GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w g ≤ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightAtRealPlaceContinuousPosLeOneStatement

theorem **_root_.GlobalTheory.T.doublingHeightAtRealPlace_continuous_pos_le_one** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), GlobalTheory.Doubling.IsMaj V a → ∀ w, Continuous (GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w) ∧ ∀ g, 0 < GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w g ∧ GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW V a w g ≤ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightAtRealPlaceLeOne

theorem **cT_fromBlocks** (P Q R S : Matrix m m A) : Formalism.UG.cT ρ (fromBlocks P Q R S) = fromBlocks (Formalism.UG.cT ρ P) (Formalism.UG.cT ρ R) (Formalism.UG.cT ρ Q) (Formalism.UG.cT ρ S)

theorem **cT_zero** : Formalism.UG.cT ρ (0 : Matrix m m A) = 0

theorem **diag_identity** (Z b binv : Matrix m m A) (P : Matrix (m ⊕ m) (m ⊕ m) A) (hbb : b ∗ binv = 1) (hKD : P.toBlocks₂₁ + P.toBlocks₂₂ = P.toBlocks₁₁ + P.toBlocks₁₂) (hU : Formalism.UG.cT ρ (fromBlocks binv 0 0 binv ∗ (fromBlocks Z 0 0 1 ∗ P) ∗ fromBlocks b 0 0 b) ∗ (fromBlocks binv 0 0 binv ∗ (fromBlocks Z 0 0 1 ∗ P) ∗ fromBlocks b 0 0 b) = 1) : Formalism.UG.cT ρ (Z ∗ (P.toBlocks₁₁ + P.toBlocks₁₂)) ∗ (Formalism.UG.cT ρ binv ∗ binv) ∗ (Z ∗ (P.toBlocks₁₁ + P.toBlocks₁₂)) + Formalism.UG.cT ρ (P.toBlocks₁₁ + P.toBlocks₁₂) ∗ (Formalism.UG.cT ρ binv ∗ binv) ∗ (P.toBlocks₁₁ + P.toBlocks₁₂) = Formalism.UG.cT ρ binv ∗ binv + Formalism.UG.cT ρ binv ∗ binv

theorem **map_identity** (φ : A →+∗ ℂ) (hφ : ∀ x, φ (ρ x) = star (φ x)) (Z α binv : Matrix m m A) (h : Formalism.UG.cT ρ (Z ∗ α) ∗ (Formalism.UG.cT ρ binv ∗ binv) ∗ (Z ∗ α) + Formalism.UG.cT ρ α ∗ (Formalism.UG.cT ρ binv ∗ binv) ∗ α = Formalism.UG.cT ρ binv ∗ binv + Formalism.UG.cT ρ binv ∗ binv) : (Z.map φ ∗ α.map φ)ᴴ ∗ ((binv.map φ)ᴴ ∗ binv.map φ) ∗ (Z.map φ ∗ α.map φ) + (α.map φ)ᴴ ∗ ((binv.map φ)ᴴ ∗ binv.map φ) ∗ α.map φ = (binv.map φ)ᴴ ∗ binv.map φ + (binv.map φ)ᴴ ∗ binv.map φ

theorem **det_one_add_ge** (W : Matrix (Fin n) (Fin n) ℂ) (hW : ‖W.det‖ = 1) : ∃ r : ℝ, 2 ^ n ≤ r ∧ (1 + Wᴴ ∗ W).det = r

theorem **norm_det_le_one** (a Z c : Matrix (Fin n) (Fin n) ℂ) (hc : c.det ≠ 0) (hZ : ‖Z.det‖ = 1) (h : (Z ∗ a)ᴴ ∗ (cᴴ ∗ c) ∗ (Z ∗ a) + aᴴ ∗ (cᴴ ∗ c) ∗ a = cᴴ ∗ c + cᴴ ∗ c) : ‖a.det‖ ≤ 1

theorem **submatrix_mul_self_equiv** {ι κ : Type} [Fintype ι] [Fintype κ] {A : Type} [CommRing A] (e : κ ≃ ι) (X Y : Matrix ι ι A) : (X ∗ Y).submatrix e e = X.submatrix e e ∗ Y.submatrix e e

theorem **cT_submatrix** {ι κ : Type} {A : Type} [CommRing A] (ρ : A →+∗ A) (e : κ ≃ ι) (X : Matrix ι ι A) : (Formalism.UG.cT ρ X).submatrix e e = Formalism.UG.cT ρ (X.submatrix e e)

theorem **absEw_aH_le_one** (hc : IsCMExt F E σ) (hJ : J.det ≠ 0) (h : Q = dblGram finSumFinEquiv J) {w : InfinitePlace F} (b : GL (Fin n) (w.Completion ⊗[F] E)) (y : GArch σ J w) (p : siegelH w.Completion σ Q) (k : GArch σ Q w) (hk : k ∈ KArch σ Q w b) (e : ioH _ σ h (y, 1) = p ∗ k) : absEw F E w (aH _ σ Q p) ≤ 1

theorem **hgtArch_ioH_le_one** (hc : IsCMExt F E σ) (hJ : J.det ≠ 0) (h : Q = dblGram finSumFinEquiv J) {w : InfinitePlace F} (b : GL (Fin n) (w.Completion ⊗[F] E)) (y : GArch σ J w) : hgtArch σ Q w b (ioH _ σ h (y, 1)) ≤ 1

theorem **hgtW_le_one** : ∀ (D : D7) (A : HSp D) (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ (w : InfinitePlace D.host.F) (y : Gw A w), GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW A a w y ≤ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightAtRealPlaceLeOneStatement

theorem **_root_.GlobalTheory.T.doublingHeightAtRealPlace_le_one** : ∀ (D : D7) (A : HSp D) (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ (w : InfinitePlace D.host.F) (y : Gw A w), GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtW A a w y ≤ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightEqOneLatticeStabilizerComponents

theorem **hgt_one_components** : ∀ (D : D7) (V : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)), IsSelfDualC V Λ → (∀ v, ∀ k ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV V Λ v k = 1) ∧ (∀ κ ∈ GfC V, ∀ w, GlobalTheory.AdelicU.projArch D.sp.cm.sig V.gram w κ = 1) ∧ ∀ b ∈ GinfC V, ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v b = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightEqOneLatticeStabilizerComponentsStatement

theorem **_root_.GlobalTheory.T.doublingHeight_eq_one_lattice_stabilizer_components** : ∀ (D : D7) (V : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)), IsSelfDualC V Λ → (∀ v, ∀ k ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV V Λ v k = 1) ∧ (∀ κ ∈ GfC V, ∀ w, GlobalTheory.AdelicU.projArch D.sp.cm.sig V.gram w κ = 1) ∧ ∀ b ∈ GinfC V, ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v b = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightSplitPlaceChartBiInvariantTorusVal

theorem **hgtV_mul_stab** (D : D7) (V : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hΛ : IsSelfDualC V Λ) (v : HeightOneSpectrum (𝒪 D.host.F)) (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) {k k' : GlobalTheory.AdelicU.GLoc D.sp.cm.sig V.gram v} (hk : k ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v) (hk' : k' ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v) (g : GlobalTheory.AdelicU.GLoc D.sp.cm.sig V.gram v) : GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV V Λ v (k ∗ g ∗ k') = GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV V Λ v g

theorem **nnnorm_unif** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (ϖ : (Fv F v)ˣ) (h : letI

theorem **hgtFin_ioH_eq** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (Q : Matrix (Fin (n + n)) (Fin (n + n)) E) (hQ : Q = GlobalTheory.Doubling.dblGram finSumFinEquiv J) (L : Submodule (𝒪 E) (Fin n → E)) (v : HeightOneSpectrum (𝒪 F)) (g : GlobalTheory.AdelicU.GLoc σ J v) : GlobalTheory.Doubling.hgtFin σ Q L v (GlobalTheory.Doubling.ioH (Fv F v) σ hQ (g, 1)) = GlobalTheory.Doubling.hgtFin σ (GlobalTheory.Doubling.dblGram finSumFinEquiv J) L v (GlobalTheory.Doubling.ioH (Fv F v) σ rfl (g, 1))

theorem **min_zpow_inv** (q : ℝ≥0) (hq : 1 ≤ q) (m : ℤ) : min ((q⁻¹) ^ m) (((q⁻¹) ^ m)⁻¹) = q ^ (-|m|)

theorem **prod_zpow_eq** {ι : Type} (s : Finset ι) (q : ℝ≥0) (hq : q ≠ 0) (f : ι → ℤ) : ∏ i ∈ s, q ^ f i = q ^ (∑ i ∈ s, f i)

theorem **bridge** : ∀ (D : D7) (s : PiLocalization₂ D) (k : ℕ) (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v) (ι : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) →∗ GA D s.V), IsSelfDualC s.V s.Λ → Formalism.L2.DIV.IsFrame s.β s.Λ → IsChart s k φ ι → ∀ e : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) → GlobalTheory.FinZeta.Gv s.V v, (∀ y, ι y = GlobalTheory.AdelicU.emb D.sp.cm.sig s.V.gram v (e y)) → ∃ (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) (β : GL (Fin s.V.rank) (Fv D.host.F v)) (ε : Fin D.R.r ⊕ Blocks k D.R.t ≃ Fin s.V.rank), GlobalTheory.AdelicU.IsSplitBasis τ s.Λ β ∧ ∀ y, (GlobalTheory.AdelicU.splitEquivIn (GlobalTheory.AdelicU.isCMExt_host D) (GlobalTheory.Doubling.isHermND_gram s.V) τ β (e y)).1.submatrix ε ε = y.1

theorem **isHgt_chart** : ∀ (D : D7) (s : PiLocalization₂ D) (k : ℕ) (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v) (ι : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) →∗ GA D s.V), IsSelfDualC s.V s.Λ → Formalism.L2.DIV.IsFrame s.β s.Λ → IsChart s k φ ι → ∀ e : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) → GlobalTheory.FinZeta.Gv s.V v, (∀ y, ι y = GlobalTheory.AdelicU.emb D.sp.cm.sig s.V.gram v (e y)) → ∀ x ϖ, IsLocUnif φ x ϖ → GlobalTheory.ParHecke.IsHgt ϖ fun y => GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV s.V s.Λ v (e y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingHeightSplitPlaceChartBiInvariantTorusValStatement

theorem **_root_.GlobalTheory.T.doublingHeight_splitPlaceChart_biInvariant_torus_value** : ∀ (D : D7) (s : PiLocalization₂ D) (k : ℕ) (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v) (ι : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) →* GA D s.V), IsSelfDualC s.V s.Λ → Formalism.L2.DIV.IsFrame s.β s.Λ → IsChart s k φ ι → ∀ e : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) → GlobalTheory.FinZeta.Gv s.V v, (∀ y, ι y = GlobalTheory.AdelicU.emb D.sp.cm.sig s.V.gram v (e y)) → ∀ x ϖ, IsLocUnif φ x ϖ → GlobalTheory.ParHecke.IsHgt ϖ fun y => GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV s.V s.Λ v (e y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingIntegralNegligibleOrbitsEqZeroCuspForm

theorem **isNeg_siegel_mul** {p x : GA D (dblSp V)} (hp : p ∈ siegel V) : IsNeg V (p * x) ↔ IsNeg V x

theorem **isNeg_mul_io** {x : GA D (dblSp V)} {γ₁ γ₂ : GA D V} (h₁ : γ₁ ∈ GQ V) (h₂ : γ₂ ∈ GQ V) : IsNeg V (x * io V (γ₁, γ₂)) → IsNeg V x

theorem **isNeg_mul_io_iff** {x : GA D (dblSp V)} {γ₁ γ₂ : GA D V} (h₁ : γ₁ ∈ GQ V) (h₂ : γ₂ ∈ GQ V) : IsNeg V (x * io V (γ₁, γ₂)) ↔ IsNeg V x

theorem **exists_out_rmul** (q : SiegelCosets V) (g : GQ (dblSp V)) : ∃ p ∈ siegelQ V, (((rmul _ g q).out : GQ (dblSp V)) : GA D (dblSp V)) = p * (q.out : GQ (dblSp V)) * g

theorem **isNeg_out_rmul_iff** (q : SiegelCosets V) {γ₁ γ₂ : GA D V} (h₁ : γ₁ ∈ GQ V) (h₂ : γ₂ ∈ GQ V) : IsNeg V ((rmul _ ⟨io V (γ₁, γ₂), io_mem_GQ V h₁ h₂⟩ q).out : GQ (dblSp V)) ↔ IsNeg V (q.out : GQ (dblSp V))

theorem **apply_out_rmul** {a : GL (Fin V.rank) (AE D)} {μ : (AE D)ˣ →* Circle} (hμ : IsHeckeChar D μ) {s : ℂ} {φ : GA D (dblSp V) → ℂ} (hφ : φ ∈ degPS V a μ s) (q : SiegelCosets V) (g : GQ (dblSp V)) (y : GA D (dblSp V)) : φ ((((rmul _ g q).out : GQ (dblSp V)) : GA D (dblSp V)) * y) = φ (((q.out : GQ (dblSp V)) * g * y)

@[simp] theorem **io_one_one** : io V (1, 1) = 1

theorem **smul_val** (δ : GQ V) (q : NegCosets V) : (δ • q).1 = rmul _ ⟨io V ((δ : GA D V)⁻¹, 1), io_mem_GQ V (inv_mem δ.2) (one_mem _)⟩ q.1

theorem **mem_map_stabilizer_iff** (y : NegCosets V) (x : GA D V) : x ∈ (MulAction.stabilizer (GQ V) y).map (GQ V).subtype ↔ x ∈ GQ V ∧ ((y.1.out : GQ (dblSp V)) : GA D (dblSp V)) * io V (x, 1) * ((y.1.out : GQ (dblSp V)) : GA D (dblSp V))⁻¹ ∈ siegel V

theorem **leftInv_negIntegrand** {a : GL (Fin V.rank) (AE D)} {μ : (AE D)ˣ →* Circle} (hμ : IsHeckeChar D μ) {s : ℂ} {φ : GA D (dblSp V) → ℂ} (hφ : φ ∈ degPS V a μ s) {f₁ f₂ : GA D V → ℂ} (hf₁ : LeftInv (GQ V) f₁) (hf₂ : LeftInv (GQ V) f₂) : LeftInv ((GQ V).prod (GQ V)) (dblIntegrand V μ (fun y => ∑' q : NegCosets V, φ (((q.1.out : GQ (dblSp V)) : GA D (dblSp V)) * y)) f₁ f₂)

theorem **integral_neg_eq_zero** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V (GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → IsCuspFun V f₁ → ∫ x, GlobalTheory.Eis.dblIntegrandQ V μ (fun y => ∑' q : {q : GlobalTheory.Eis.SiegelCosets V // ¬ ∃ δ ∈ GQ V, ((q.out : GQ (dblSp V)) : GA D (dblSp V)) * (io V (δ, 1))⁻¹ ∈ siegel V}, φ (((q.1.out : GQ (dblSp V)) : GA D (dblSp V)) * y)) f₁ f₂ x ∂(autMeasure V m).prod (autMeasure V m) = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingIntegralNegligibleOrbitsEqZeroCuspFormStatement

theorem **_root_.GlobalTheory.T.doubling_integral_negligible_orbits_eq_zero_cuspForm** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V (GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → IsCuspFun V f₁ → ∫ x, GlobalTheory.Eis.dblIntegrandQ V μ (fun y => ∑' q : {q : GlobalTheory.Eis.SiegelCosets V // ¬ ∃ δ ∈ GQ V, ((q.out : GQ (dblSp V)) : GA D (dblSp V)) * (io V (δ, 1))⁻¹ ∈ siegel V}, φ (((q.1.out : GQ (dblSp V)) : GA D (dblSp V)) * y)) f₁ f₂ x ∂(autMeasure V m).prod (autMeasure V m) = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingIntegralsVanishOfAllLiftsVanish

theorem **exists_bound_of_fundamentalDomain** {A : HSp D} {m : Measure (GA D A)} [m.IsHaarMeasure] {X K : Set (GA D A)} (hX : IsFundamentalDomain (GQ A) X m) (hK : IsCompact K) (hXK : X ⊆ K) {u : GA D A → ℂ} (hu : Continuous u) (hi : ∀ γ ∈ GQ A, ∀ g, u (γ * g) = u g) : ∃ C : ℝ, ∀ g, ‖u g‖ ≤ C

theorem **thetaKO_rat** {S T : HSp D} {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {om : WRep S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (h : GA D T) : ∀ γ ∈ GQ S, ∀ g, thetaKO om (γ * g) h f = thetaKO om g h f

theorem **continuous_thetaKO** {S T : HSp D} {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {om : WRep S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (h : GA D T) : Continuous fun g => thetaKO om g h f

theorem **kills_of_noLift** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A W : HSp D) (χA χW : (AE D)ˣ →* Circle) (om : WRep A W) (omh : WRep (dblSp A) W), HSpace.SigOK A → HSpace.IsPosDef A → IsWeilDatumO ψ A W χA χW om → IsWeilDatumO ψ (dblSp A) W 1 χW omh → ∀ m : Measure (GA D A), m.IsHaarMeasure → ∀ u : GA D A → ℂ, Continuous u → (∀ γ ∈ GQ A, ∀ g, u (γ * g) = u g) → NoLift ψ om m u → Kills ψ χW omh m u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingIntegralsVanishOfAllLiftsVanishStatement

theorem **_root_.GlobalTheory.T.doubling_integrals_vanish_of_all_lifts_vanish** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A W : HSp D) (χA χW : (AE D)ˣ →* Circle) (om : WRep A W) (omh : WRep (dblSp A) W), HSpace.SigOK A → HSpace.IsPosDef A → IsWeilDatumO ψ A W χA χW om → IsWeilDatumO ψ (dblSp A) W 1 χW omh → ∀ m : Measure (GA D A), m.IsHaarMeasure → ∀ u : GA D A → ℂ, Continuous u → (∀ γ ∈ GQ A, ∀ g, u (γ * g) = u g) → NoLift ψ om m u → Kills ψ χW omh m u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingMapSpanDenseHeisenbergStable

theorem **span_delta_eq** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (A W : HSp D) : (Submodule.span ℂ {φ | ∃ f₁ f₂, IsThetaFunO ψ A W f₁ ∧ IsThetaFunO ψ A W f₂ ∧ φ = delta A W f₁ f₂} : Submodule ℂ (WO (dblSp A) W → ℂ)) = GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.prodSpan ((GlobalTheory.DoublingTheta.sumDec A (negSp A)).resS W) (IsThetaFunO ψ A W) (IsThetaFunO ψ (negSp A) W)

theorem **prodSpan_theta_heis** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S A B : HSp D} (o : GlobalTheory.MixModel.OrthDec S A B) (W : HSp D) : (∀ g ∈ GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.prodSpan (o.resS W) (IsThetaFunO ψ A W) (IsThetaFunO ψ B W), IsThetaFunO ψ S W g) ∧ ∀ g ∈ GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.prodSpan (o.resS W) (IsThetaFunO ψ A W) (IsThetaFunO ψ B W), ∀ w, heisO ψ S W w g ∈ GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.prodSpan (o.resS W) (IsThetaFunO ψ A W) (IsThetaFunO ψ B W)

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A W : HSp D), IsStdChar D ψ (-1) → let M : Submodule ℂ (WO (dblSp A) W → ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingMapSpanDenseHeisenbergStableStatement

theorem **_root_.GlobalTheory.T.doublingMap_span_dense_heisenbergStable** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A W : HSp D), IsStdChar D ψ (-1) → let M : Submodule ℂ (WO (dblSp A) W → ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingMapThetaIdentityAnyWeilData

theorem **inL_map** {R' : Type} [CommRing R'] (f : R →+* R') : (inL R m n).map f = inL R' m n

theorem **inR_map** {R' : Type} [CommRing R'] (f : R →+* R') : (inR R m n).map f = inR R' m n

theorem **mul_inL** {l : ℕ} (x : Matrix (Fin l) (Fin (m + n)) R) : x * inL R m n = Matrix.of fun i j => x i (finSumFinEquiv (Sum.inl j))

theorem **mul_inR** {l : ℕ} (x : Matrix (Fin l) (Fin (m + n)) R) : x * inR R m n = Matrix.of fun i j => x i (finSumFinEquiv (Sum.inr j))

theorem **blkMat_inL** (a : Matrix (Fin m) (Fin m) R) (b : Matrix (Fin n) (Fin n) R) : blkMat finSumFinEquiv a b * inL R m n = inL R m n * a

theorem **blkMat_inR** (a : Matrix (Fin m) (Fin m) R) (b : Matrix (Fin n) (Fin n) R) : blkMat finSumFinEquiv a b * inR R m n = inR R m n * b

theorem **inL_inL** : (inL R m n)ᴴ * inL R m n = 1

theorem **inR_inR** : (inR R m n)ᴴ * inR R m n = 1

theorem **inR_inL** : (inR R m n)ᴴ * inL R m n = 0

theorem **sumDec_jA** (V W : HSp D) : (sumDec V W).jA = inL (AE D) V.rank W.rank

theorem **sumDec_jB** (V W : HSp D) : (sumDec V W).jB = inR (AE D) V.rank W.rank

theorem **sumDec_emb** (V W : HSp D) (g : GA D V) (g' : GA D W) : (sumDec V W).emb (g, g') = ioSum V W (g, g')

theorem **boxS_apply** (A B : HSp D) (f : WO A B → ℂ) (f' : WO (negSp A) B → ℂ) (x : WO (dblSp A) B) : (sumDec A (negSp A)).boxS B f f' x = f (resL A B x) * f' (resR A B x)

theorem **thetaClause** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (om : WRep A B) (omh : WRep (dblSp A) B), IsWeilDatumO ψ A B χA χB om → IsWeilDatumO ψ (dblSp A) B 1 χB omh → ThetaClause ψ A B χB om omh

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingMapThetaIdentityAnyWeilDataStatement

theorem **_root_.GlobalTheory.T.doublingMap_thetaIdentity_any_weilData** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (om : WRep A B) (omh : WRep (dblSp A) B), IsWeilDatumO ψ A B χA χB om → IsWeilDatumO ψ (dblSp A) B 1 χB omh → ThetaClause ψ A B χB om omh

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingMethodOfEisensteinFacts

theorem **doublingMethod_of_three** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.SiegelEisensteinContinuation → GlobalTheory.Published.SiegelEisensteinSimplePole → GlobalTheory.Published.DoublingMethod

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingMethodOfEisensteinFactsStatement

theorem **_root_.GlobalTheory.T.doubling_method_of_eisenstein_facts** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.SiegelEisensteinContinuation → GlobalTheory.Published.SiegelEisensteinSimplePole → GlobalTheory.Published.DoublingMethod

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingNegligibleOrbitUnfoldedIntegralEqZero

theorem **integral_eq_zero_of_not_open** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)) (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, ∀ φ ∈ degPS V a μ s, ∀ γ ∈ GQ (dblSp V), (¬ ∃ δ ∈ GQ V, γ * (io V (δ, 1))⁻¹ ∈ siegel V) → ∀ f₁ f₂ : GA D V → ℂ, Continuous f₁ → Continuous f₂ → GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ V) f₁ → IsCuspFun V f₁ → ∀ (g₂ : GA D V) (Γ' : Subgroup (GA D V)), (∀ x, x ∈ Γ' ↔ x ∈ GQ V ∧ γ * io V (x, 1) * γ⁻¹ ∈ siegel V) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ 𝒢 : Set (GA D V), IsFundamentalDomain Γ' 𝒢 m → IntegrableOn (fun h => GlobalTheory.Eis.dblIntegrand V μ (fun x => φ (γ * x)) f₁ f₂ (h, g₂)) 𝒢 m → ∫ h in 𝒢, GlobalTheory.Eis.dblIntegrand V μ (fun x => φ (γ * x)) f₁ f₂ (h, g₂) ∂m = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingNegligibleOrbitUnfoldedIntegralEqZeroStatement

theorem **_root_.GlobalTheory.T.doublingNegligibleOrbit_unfoldedIntegral_eq_zero** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)) (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, ∀ φ ∈ degPS V a μ s, ∀ γ ∈ GQ (dblSp V), (¬ ∃ δ ∈ GQ V, γ * (io V (δ, 1))⁻¹ ∈ siegel V) → ∀ f₁ f₂ : GA D V → ℂ, Continuous f₁ → Continuous f₂ → GlobalTheory.AutomorphicL2Unitary.LeftInv (GQ V) f₁ → IsCuspFun V f₁ → ∀ (g₂ : GA D V) (Γ' : Subgroup (GA D V)), (∀ x, x ∈ Γ' ↔ x ∈ GQ V ∧ γ * io V (x, 1) * γ⁻¹ ∈ siegel V) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ 𝒢 : Set (GA D V), IsFundamentalDomain Γ' 𝒢 m → IntegrableOn (fun h => GlobalTheory.Eis.dblIntegrand V μ (fun x => φ (γ * x)) f₁ f₂ (h, g₂)) 𝒢 m → ∫ h in 𝒢, GlobalTheory.Eis.dblIntegrand V μ (fun x => φ (γ * x)) f₁ f₂ (h, g₂) ∂m = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingOpenOrbitUnfoldsToAdelicIntegral

theorem **io_one_mul** (δ : GA D V) (x : GA D V × GA D V) : io V (δ, 1) * io V x = io V (δ * x.1, x.2)

theorem **term_eq** (hf₁ : ∀ γ ∈ GQ V, ∀ g, f₁ (γ * g) = f₁ g) (δ : GQ V) (x : GA D V × GA D V) : dblIntegrand V μ (fun y => φ (io V (δ, 1) * y)) f₁ f₂ x = dblIntegrand V μ φ f₁ f₂ ((δ : GA D V) * x.1, x.2)

theorem **tsum_open_ne_top** (hμ : IsHeckeChar D μ) {a : GL (Fin V.rank) (AE D)} {s : ℂ} (hφ : φ ∈ degPS V a μ s) {m : Measure (GA D V)} [SFinite m] {𝓕 : Set (GA D V)} (h : ∑' q : SiegelCosets V, ∫⁻ x in 𝓕 ×ˢ 𝓕, ‖dblIntegrand V μ (fun x => φ (((q.out : GQ (dblSp V)) : GA D (dblSp V)) * x)) f₁ f₂ x‖ₑ ∂m.prod m ≠ ⊤) : ∑' δ : GQ V, ∫⁻ x, ‖dblIntegrand V μ (fun y => φ (io V (δ, 1) * y)) f₁ f₂ x‖ₑ ∂(m.restrict 𝓕).prod (m.restrict 𝓕) ≠ ⊤

theorem **leftInv_open** (hμ : IsHeckeChar D μ) {a : GL (Fin V.rank) (AE D)} {s : ℂ} (hφ : φ ∈ degPS V a μ s) (hf₁ : ∀ γ ∈ GQ V, ∀ g, f₁ (γ * g) = f₁ g) (hf₂ : ∀ γ ∈ GQ V, ∀ g, f₂ (γ * g) = f₂ g) : LeftInv ((GQ V).prod (GQ V)) (dblIntegrand V μ (fun y => ∑' δ : GQ V, φ (io V (δ, 1) * y)) f₁ f₂)

theorem **open_orbit** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V (GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → ZetaConvH D.sp.cm.sig (gram_hdbl V) m φ (GlobalTheory.Eis.petCoef V m f₁ f₂) ∧ ∫ x,

GlobalTheory.Eis.dblIntegrandQ V μ (fun y => ∑' δ : GQ V, φ (io V (δ, 1) * y)) f₁ f₂ x ∂(autMeasure V m).prod (autMeasure V m) = zetaH D.sp.cm.sig (gram_hdbl V) m φ (GlobalTheory.Eis.petCoef V m f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingOpenOrbitUnfoldsToAdelicIntegralStatement

theorem **_root_.GlobalTheory.T.doubling_open_orbit_unfolds_to_adelic_integral** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V (GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → ZetaConvH D.sp.cm.sig (gram_hdbl V) m φ (GlobalTheory.Eis.petCoef V m f₁ f₂) ∧ ∫ x, GlobalTheory.Eis.dblIntegrandQ V μ (fun y => ∑' δ : GQ V, φ (io V (δ, 1) * y)) f₁ f₂ x ∂(autMeasure V m).prod (autMeasure V m) = zetaH D.sp.cm.sig (gram_hdbl V) m φ (GlobalTheory.Eis.petCoef V m f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSeesawTensorMapPolynomialFockVector

theorem **killed_of_leibniz** {P Q U V : Type} [Zero U] [AddCommGroup V] (a : P → U → U) (b : Q → V → V) (TU : U → Prop) (TV : V → Prop) (δ : U → U → V) (c₁ c₂ : Q → P) (ha : ∀ p f, TU f → TU (a p f)) (hb : ∀ q f, TV f → TV (b q f)) (hadd : ∀ q f g, TV f → TV g → b q (f + g) = b q f + b q g) (hb0 : ∀ q, b q 0 = 0) (hδ : ∀ f g, TU f → TU g → TV (δ f g)) (h1 : ∀ g, δ 0 g = 0) (h2 : ∀ f, δ f 0 = 0) (hL : ∀ q f g, TU f → TU g → b q (δ f g) = δ (a (c₁ q) f) g + δ f (a (c₂ q) g)) (N : ℕ) : ∀ (N₁ N₂ : ℕ) (f g : U), N₁ + N₂ ≤ N → TU f → TU g → (∀ l : List P, l.length = N₁ → l.foldr a f = 0) → (∀ l : List P, l.length = N₂ → l.foldr a g = 0) → ∀ l : List Q, l.length = N → l.foldr b (δ f g) = 0

theorem **lineAt_neg** (hθ : IsInfFam θ) {m n : ℕ} (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) : lineAt θ τ (-z) = -lineAt θ τ z

theorem **lineAt_col** (hθ : IsInfFam θ) {m n k : ℕ} (c : Fin k → Fin n) (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : lineAt θ τ (z.submatrix id c) t = (lineAt θ τ z t).submatrix id c

theorem **hasDerivAt_dRho** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S T) : HasDerivAt (fun t : ℝ => heisO ψ S T (lineAt θ τ z t) f w) (dRho (heisO ψ S T) θ τ z f w) 0

theorem **dRho_add** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : dRho (heisO ψ S T) θ τ z (f + g) = dRho (heisO ψ S T) θ τ z f + dRho (heisO ψ S T) θ τ z g

theorem **dRho_neg** (hθ : IsInfFam θ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (f : WO S T → ℂ) : dRho (heisO ψ S T) θ τ (-z) f = -dRho (heisO ψ S T) θ τ z f

theorem **symplOH_dbl** (x y : WO (dblSp A) B) : symplOH (dblSp A) B x y = symplOH A B (resL A B x) (resL A B y) - symplOH A B (-resR A B x) (-resR A B y)

theorem **heisO_delta** (hψ : IsStdChar D ψ (-1)) (u : WO (dblSp A) B) (f₁ f₂ : WO A B → ℂ) : heisO ψ (dblSp A) B u (delta A B f₁ f₂) = delta A B (heisO ψ A B (resL A B u) f₁) (heisO ψ A B (-resR A B u) f₂)

theorem **dRho_delta** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (z : Matrix (Fin B.rank) (Fin (dblSp A).rank) ℂ) {f₁ f₂ : WO A B → ℂ} (h₁ : IsThetaFunO ψ A B f₁) (h₂ : IsThetaFunO ψ A B f₂) : dRho (heisO ψ (dblSp A) B) θ τ z (delta A B f₁ f₂) = delta A B (dRho (heisO ψ A B) θ τ (z.submatrix id (colL A)) f₁) f₂ + delta A B f₁ (dRho (heisO ψ A B) θ τ (-z.submatrix id (colR A)) f₂)

theorem **cplxO_colL** (s : Matrix (Fin A.rank) (Fin A.rank) ℂ) (s' : Matrix (Fin B.rank) (Fin B.rank) ℂ) (z : Matrix (Fin B.rank) (Fin (dblSp A).rank) ℂ) : (cplxO (n

theorem **cplxO_colR** (s : Matrix (Fin A.rank) (Fin A.rank) ℂ) (s' : Matrix (Fin B.rank) (Fin B.rank) ℂ) (z : Matrix (Fin B.rank) (Fin (dblSp A).rank) ℂ) : (cplxO (n

theorem **annih_delta** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (s : Matrix (Fin A.rank) (Fin A.rank) ℂ) (s' : Matrix (Fin B.rank) (Fin B.rank) ℂ) (z : Matrix (Fin B.rank) (Fin (dblSp A).rank) ℂ) {f₁ f₂ : WO A B → ℂ} (h₁ : IsThetaFunO ψ A B f₁) (h₂ : IsThetaFunO ψ A B f₂) : annih (heisO ψ (dblSp A) B) θ τ (cplxO (n

theorem **isFockVec_delta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B : HSp D) (sA : RealMat A) (sB : RealMat B), IsStdChar D ψ (-1) → IsInfFam θ → ∀ Φ₁ Φ₂, IsFockVec ψ θ A B sA sB Φ₁ → IsFockVec ψ θ A B sA sB Φ₂ → IsFockVec ψ θ (dblSp A) B (fun τ => (Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm) sB (delta A B Φ₁ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSeesawTensorMapPolynomialFockVectorStatement

theorem **_root_.GlobalTheory.T.doublingSeesawTensorMap_polynomial_fock_vector** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B : HSp D) (sA : RealMat A) (sB : RealMat B), IsStdChar D ψ (-1) → IsInfFam θ → ∀ Φ₁ Φ₂, IsFockVec ψ θ A B sA sB Φ₁ → IsFockVec ψ θ A B sA sB Φ₂ → IsFockVec ψ θ (dblSp A) B (fun τ => (Matrix.fromBlocks (sA τ) 0 0 (-sA τ)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm) sB (delta A B Φ₁ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSiegelEqLeviMulUnipotent

theorem **stkRows_mul** {R : Type} [CommRing R] {a k l : ℕ} (X Y : Matrix (Fin a) (Fin k) R) (B : Matrix (Fin k) (Fin l) R) : stkRows X Y * B = stkRows (X * B) (Y * B)

theorem **keepsDiag_mul_stkRows** {R : Type} [CommRing R] {n k : ℕ} {M : Matrix (Fin (n + n)) (Fin (n + n)) R} (h : KeepsDiag finSumFinEquiv M) (X : Matrix (Fin n) (Fin k) R) : M * stkRows X X = stkRows (leviMat finSumFinEquiv M * X) (leviMat finSumFinEquiv M * X)

theorem **rat_stackRows** {a k : ℕ} (X Y : Matrix (Fin a) (Fin k) D.host.E) : rat (stackRows X Y) = stkRows (rat X) (rat Y)

theorem **dblDec_jX** (A : HSp D) : (dblDec A).jX = stkRows (giA A) (giA A)

theorem **siegel_mt_jX** (A : HSp D) {g : GA D (dblSp A)} (hg : g ∈ siegel A) : mt (dblSp A) g · (dblDec A).jX = (dblDec A).jX · (gA A · leviMat finSumFinEquiv (mt (dblSp A) g) · giA A)

theorem **siegel_eq_levi_nC** : ∀ (D : D7) (A : HSp D) (p : siegel A), ∃ (a : GL (Fin A.rank) (AE D)) (c : skewA D A.rank), (p : GA D (dblSp A)) = (dblDec A).levi a * (dblDec A).nC c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSiegelEqLeviMulUnipotentStatement

theorem **_root_.GlobalTheory.T.doublingSiegel_eq_levi_mul_unipotent** : ∀ (D : D7) (A : HSp D) (p : siegel A), ∃ (a : GL (Fin A.rank) (AE D)) (c : skewA D A.rank), (p : GA D (dblSp A)) = (dblDec A).levi a * (dblDec A).nC c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSiegelLeviDetHomLeviMulUnipotent

theorem **aP_levi_mul_nC** : ∀ (D : D7) (A : HSp D) (p : siegel A) (a : GL (Fin A.rank) (AE D)) (c : skewA D A.rank), (p : GA D (dblSp A)) = (dblDec A).levi a * (dblDec A).nC c → aP A p = Matrix.GeneralLinearGroup.det a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSiegelLeviDetHomLeviMulUnipotentStatement

theorem **_root_.GlobalTheory.T.doublingSiegelLeviDetHom_levi_mul_unipotent** : ∀ (D : D7) (A : HSp D) (p : siegel A) (a : GL (Fin A.rank) (AE D)) (c : skewA D A.rank), (p : GA D (dblSp A)) = (dblDec A).levi a * (dblDec A).nC c → aP A p = Matrix.GeneralLinearGroup.det a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSiegelWeilSectionMemDegeneratePrincipalS

theorem **swSec_mul** (φ : WO (dblSp A) B → ℂ) (g k : GA D (dblSp A)) : swSec A B omh ν ℱ φ (g * k) = swSec A B omh ν ℱ (omh (k, 1) φ) g

theorem **swSec_mem_degPS** : GlobalTheory.Published.WeilRepresentationUnitaryPair → ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : WRep (dblSp A) B), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (ν : Measure (WO A B)) (ℱ : Set (WO A B)), IsCovolOne (ratL B A) ν ℱ → ∀ φ, IsThetaFunO ψ (dblSp A) B φ → ∀ a : GL (Fin A.rank) (AE D), IsKFinVec omh (Khat A a) φ → swSec A B omh ν ℱ φ ∈ degPS A a χB⁻¹ (sZero A B)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSiegelWeilSectionMemDegeneratePrincipalSStatement

theorem **_root_.GlobalTheory.T.doublingSiegelWeilSection_mem_degeneratePrincipalSeries** : GlobalTheory.Published.WeilRepresentationUnitaryPair → ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : WRep (dblSp A) B), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (ν : Measure (WO A B)) (ℱ : Set (WO A B)), IsCovolOne (ratL B A) ν ℱ → ∀ φ, IsThetaFunO ψ (dblSp A) B φ → ∀ a : GL (Fin A.rank) (AE D), IsKFinVec omh (Khat A a) φ → swSec A B omh ν ℱ φ ∈ degPS A a χB⁻¹ (sZero A B)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSphericalFactorizationOfWeilDualPair

theorem **unrAt_inv_of_unrAt** {D : D7} {χ : (AE D)ˣ →* Circle} {w : HeightOneSpectrum (𝒪 D.host.E)} (h : GlobalTheory.LFun.UnrAt χ w) : GlobalTheory.LFun.UnrAt χ⁻¹ w

theorem **isHeckeChar_inv_sph** {D : D7} {χ : (AE D)ˣ →* Circle} (h : IsHeckeChar D χ) : IsHeckeChar D χ⁻¹

theorem **sphFactor_of_weilRepresentationUnitaryPair** (hW : GlobalTheory.Published.WeilRepresentationUnitaryPair) (D : D7) : SphFactor D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSphericalFactorizationOfWeilDualPairStatement

theorem **_root_.GlobalTheory.T.doublingSphericalFactorization_of_weilDualPair** : GlobalTheory.Published.WeilRepresentationUnitaryPair → ∀ D : D7, SphFactor D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSphericalVacuaToDoubledLatticeVacuum

theorem **gram_dbl_inv** (A : HSp D) : (dblSp A).gram⁻¹ = blkMat finSumFinEquiv A.gram⁻¹ (-A.gram⁻¹)

theorem **hO_dbl** (A B : HSp D) (x y : WO (dblSp A) B) : hO (dblSp A) B x y = hO A B (resL A B x) (resL A B y) - hO A B (resR A B x) (resR A B y)

theorem **hO_neg_neg** (A B : HSp D) (x y : WO A B) : hO A B (-x) (-y) = hO A B x y

theorem **symplOH_dbl** (A B : HSp D) (x y : WO (dblSp A) B) : symplOH (dblSp A) B x y = symplOH A B (resL A B x) (resL A B y) - symplOH A B (resR A B x) (resR A B y)

theorem **symplOH_neg_neg** (A B : HSp D) (x y : WO A B) : symplOH A B (-x) (-y) = symplOH A B x y

theorem **resL_add** (A B : HSp D) (x y : WO (dblSp A) B) : resL A B (x + y) = resL A B x + resL A B y

theorem **resR_add** (A B : HSp D) (x y : WO (dblSp A) B) : resR A B (x + y) = resR A B x + resR A B y

theorem **conj_psi_eq** {ψ : AddChar (AF D.host.F) ℂ} (hψ : ∀ x, ‖ψ x‖ = 1) (z : AF D.host.F) : conj (ψ z) = ψ (-z)

theorem **heisO_delta** {ψ : AddChar (AF D.host.F) ℂ} (hψ : ∀ x, ‖ψ x‖ = 1) (A B : HSp D) (w₀ : WO (dblSp A) B) (f₁ f₂ : WO A B → ℂ) : heisO ψ (dblSp A) B w₀ (delta A B f₁ f₂) = delta A B (heisO ψ A B (resL A B w₀) f₁) (heisO ψ A B (-resR A B w₀) f₂)

theorem **sum_dbl_gram_inl** (A : HSp D) (s : Fin (dblSp A).rank → D.host.E) (j : Fin A.rank) : ∑ k, star (s k) * (dblSp A).gram k (finSumFinEquiv (Sum.inl j)) = ∑ k, star ((fun k => s (finSumFinEquiv (Sum.inl k))) k) * A.gram k j

theorem **sum_dbl_gram_inr** (A : HSp D) (s : Fin (dblSp A).rank → D.host.E) (j : Fin A.rank) : ∑ k, star (s k) * (dblSp A).gram k (finSumFinEquiv (Sum.inr j)) = ∑ k, star ((-fun k => s (finSumFinEquiv (Sum.inr k))) k) * A.gram k j

theorem **res_mem_latO** {A B : HSp D} {LA : Lat A} {LB : Lat B} {P : FinPl D → Prop} {u : WO (dblSp A) B} (hu : u ∈ latO (dblSp A) B (sumLat finSumFinEquiv LA LA : Lat (dblSp A)) LB P) : resL A B u ∈ latO A B LA LB P ∧ resR A B u ∈ latO A B LA LB P

theorem **latO_mono_at** {A B : HSp D} (LA : Lat A) (LB : Lat B) {Sf : Finset (FinPl D)} {v : FinPl D} (hv : v ∉ Sf) : latO A B LA LB (fun u => u = v) ≤ latO A B LA LB (fun u => u ∉ Sf)

theorem **pair_dbl** (A : HSp D) (x y : Fin (dblSp A).rank → D.host.E) : ∑ i, ∑ j, star (x i) * (dblSp A).gram i j * y j = (∑ i, ∑ j, star (x (finSumFinEquiv (Sum.inl i))) * A.gram i j * y (finSumFinEquiv (Sum.inl j))) - ∑ i, ∑ j, star (x (finSumFinEquiv (Sum.inr i))) * A.gram i j * y (finSumFinEquiv (Sum.inr j))

theorem **selfDual_sumLat** {A : HSp D} {L : Lat A} (hL : IsSelfDualC A L) : IsSelfDualC (dblSp A) (sumLat finSumFinEquiv L L)

theorem **doubling_isLatVac** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (LA : Lat A) (LB : Lat B), IsStdChar D ψ (-1) → IsSelfDualC A LA → IsSelfDualC (dblSp A) (sumLat finSumFinEquiv LA LA) ∧ ∀ (Sf : Finset (FinPl D)) (Φ₁ Φ₂ : WO A B → ℂ), IsSphOff ψ A B LA LB Sf Φ₁ → IsSphOff ψ A B LA LB Sf Φ₂ → ∀ v ∉ Sf, IsLatVac (heisO ψ (dblSp A) B) (latO (dblSp A) B (sumLat finSumFinEquiv LA LA : Lat (dblSp A)) LB fun u => u = v) (delta A B Φ₁ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingSphericalVacuaToDoubledLatticeVacuumStatement

theorem **_root_.GlobalTheory.T.doubling_sphericalVacua_to_doubled_lattice_vacuum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (LA : Lat A) (LB : Lat B), IsStdChar D ψ (-1) → IsSelfDualC A LA → IsSelfDualC (dblSp A) (sumLat finSumFinEquiv LA LA) ∧ ∀ (Sf : Finset (FinPl D)) (Φ₁ Φ₂ : WO A B → ℂ), IsSphOff ψ A B LA LB Sf Φ₁ → IsSphOff ψ A B LA LB Sf Φ₂ → ∀ v ∉ Sf, IsLatVac (heisO ψ (dblSp A) B) (latO (dblSp A) B (sumLat finSumFinEquiv LA LA : Lat (dblSp A)) LB fun u => u = v) (delta A B Φ₁ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingVectorFockPairIsCompactFinite

theorem **delta_isKFinVec** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (sA : RealMat A) (sB : RealMat B) (t : ℕ) (cA : CptF A) (a : GL (Fin A.rank) (AE D)) (om : WRep A B) (ν : Measure (WO A B)) (ℱ : Set (WO A B)), Pair ψ θ A B χA χB sA sB t cA a om ν ℱ → ∀ omh : WRep (dblSp A) B, IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ Φ₁ Φ₂, IsFockVec ψ θ A B sA sB Φ₁ → IsFockVec ψ θ A B sA sB Φ₂ → IsKFinVec omh (Khat A a) (delta A B Φ₁ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingVectorFockPairIsCompactFiniteStatement

theorem **_root_.GlobalTheory.T.doublingVector_fockPair_isCompactFinite** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (sA : RealMat A) (sB : RealMat B) (t : ℕ) (cA : CptF A) (a : GL (Fin A.rank) (AE D)) (om : WRep A B) (ν : Measure (WO A B)) (ℱ : Set (WO A B)), Pair ψ θ A B χA χB sA sB t cA a om ν ℱ → ∀ omh : WRep (dblSp A) B, IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ Φ₁ Φ₂, IsFockVec ψ θ A B sA sB Φ₁ → IsFockVec ψ θ A B sA sB Φ₂ → IsKFinVec omh (Khat A a) (delta A B Φ₁ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingWittTowerReduction

theorem **dblInt_smul_left** {D : D7} {V : HSp D} {μ : (AE D)ˣ →* Circle} {m : Measure (GA D V)} (c : ℂ) (E : GA D (dblSp V) → ℂ) (f : GA D V → ℂ) : GlobalTheory.Eis.dblInt V μ m (c • E) f f = c * GlobalTheory.Eis.dblInt V μ m E f f

theorem **res_killed** : GlobalTheory.Published.RegularisedSiegelWeilFirstTerm → ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A : HSp D) (cA : CptF A) (a : GL (Fin A.rank) (AE D)), Defin θ A cA a → (∀ v : FinPl D, (2 : 𝒪 D.host.F) ∈ v.asIdeal → Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingWittTowerReductionStatement

theorem **_root_.GlobalTheory.T.doubling_witt_tower_reduction** : GlobalTheory.Published.RegularisedSiegelWeilFirstTerm → ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A : HSp D) (cA : CptF A) (a : GL (Fin A.rank) (AE D)), Defin θ A cA a → (∀ v : FinPl D, (2 : 𝒪 D.host.F) ∈ v.asIdeal → Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingZetaValueAtEisensteinHolomorphicPoint

theorem **zetaHolValueAt_of_heightBounds** (l1 : ∀ (D : D7) (V : HSp D), ∃ (C : ℝ) (c : ℕ), ∀ g₁ g₂ : GA D V, GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ C * (GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂) ^ c) (l2 : ∀ (D : D7) (V : HSp D) (μ : (AE D)ˣ →* Circle) (m : Measure (GA D V)), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → ∀ (C' : ℝ) (c : ℕ), (∀ g₁ g₂ : GA D V, GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ C' * (GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂) ^ c) → ∀ (C : ℝ) (N : ℕ), ∃ bound : AutQ V × AutQ V → ℝ, Integrable bound ((autMeasure V m).prod (autMeasure V m)) ∧ ∀ E : GA D (dblSp V) → ℂ, LeftInv (GQ (dblSp V)) E → (∀ g, ‖E g‖ ≤ C * GlobalTheory.Grow.height (dblSp V) g ^ N) → ∀ q, ‖GlobalTheory.Eis.dblIntegrandQ V μ E f₁ f₂ q‖ ≤ bound q) : ∀ (D : D7) (V : HSp D) (μ

theorem **zetaHolValueAt_of_eisValueAt** : ∀ (D : D7) (V : HSp D) (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ (m : Measure (GA D V)), m.IsHaarMeasure → ∀ (Φ : ℂ → GA D (dblSp V) → ℂ), GlobalTheory.Eis.EisContinues V Φ → (∀ g, AnalyticOnNhd ℂ (GlobalTheory.Eis.eisSer V Φ g) (rightOf (V.rank / 2))) → (∀ s₀, ∃ k, GlobalTheory.Eis.EisRegularAt V Φ s₀ k) → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → GlobalTheory.Eis.IsZetaFamily V μ m Φ f₁ f₂ → ∀ (s₀ : ℂ) (E₀ : GA D (dblSp V) → ℂ), GlobalTheory.Eis.EisValueAt V Φ s₀ E₀ → GlobalTheory.Eis.DblConv V μ m E₀ f₁ f₂ ∧ GlobalTheory.Eis.ZetaHolValueAt V μ m Φ f₁ f₂ s₀ (GlobalTheory.Eis.dblInt V μ m E₀ f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DoublingZetaValueAtEisensteinHolomorphicPointStatement

theorem **_root_.GlobalTheory.T.doubling_zeta_value_at_eisenstein_holomorphic_point** : ∀ (D : D7) (V : HSp D) (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ (m : Measure (GA D V)), m.IsHaarMeasure → ∀ (Φ : ℂ → GA D (dblSp V) → ℂ), GlobalTheory.Eis.EisContinues V Φ → (∀ g, AnalyticOnNhd ℂ (GlobalTheory.Eis.eisSer V Φ g) (rightOf (V.rank / 2))) → (∀ s₀, ∃ k, GlobalTheory.Eis.EisRegularAt V Φ s₀ k) → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → GlobalTheory.Eis.IsZetaFamily V μ m Φ f₁ f₂ → ∀ (s₀ : ℂ) (E₀ : GA D (dblSp V) → ℂ), GlobalTheory.Eis.EisValueAt V Φ s₀ E₀ → GlobalTheory.Eis.DblConv V μ m E₀ f₁ f₂ ∧ GlobalTheory.Eis.ZetaHolValueAt V μ m Φ f₁ f₂ s₀ (GlobalTheory.Eis.dblInt V μ m E₀ f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualDirectionDerivativeMixedCoordinateBound

theorem **gram_posSp_one** (D : D7) (n : ℕ) : (GlobalTheory.Size.posSp D n).gram = 1

theorem **giA_posSp** (D : D7) (n : ℕ) : giA (GlobalTheory.Size.posSp D n) = 1

theorem **hO_posSp** (D : D7) (T : HSp D) (ρ : ℕ) (x y : Mx D T.rank ρ) : hO (GlobalTheory.Size.posSp D ρ) T x y = pairS T x y

theorem **symplO_eq_two_half** {D : D7} (S T : HSp D) (x y : WO S T) : symplO S T x y = symplOH S T x y + symplOH S T x y

theorem **symplAt_posSp** (D : D7) (T : HSp D) (ρ : ℕ) (τ : D.what₂.Tau) (z X : Matrix (Fin T.rank) (Fin ρ) ℂ) : GlobalTheory.IndefiniteRowSideFullHecke.HermitianSymplecticScalars.symplAt (GlobalTheory.Size.posSp D ρ) T τ z X = 2 * (tauE D τ D.host.xi0 * (Xᴴ * gramAt D T τ * z).trace).re

theorem **norm_mixS_dRho_ge** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin ρ) ℂ), ∃ Z : Matrix (Fin T.rank) (Fin S.rank) ℂ, ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → ∀ x w, |2 * (tauE D τ D.host.xi0 * ((compAt θ τ x)ᴴ * gramAt D T τ * z).trace).re| * ‖d.mixS T ψ v.μ v.ℱ f x w‖ ≤ ‖d.mixS T ψ v.μ v.ℱ (dRho (heisO ψ S T) θ τ Z f) x w‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualDirectionDerivativeMixedCoordinateBoundStatement

theorem **_root_.GlobalTheory.T.dualDirection_derivative_mixedCoordinate_bound** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin ρ) ℂ), ∃ Z : Matrix (Fin T.rank) (Fin S.rank) ℂ, ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → ∀ x w, |2 * (tauE D τ D.host.xi0 * ((compAt θ τ x)ᴴ * gramAt D T τ * z).trace).re| * ‖d.mixS T ψ v.μ v.ℱ f x w‖ ≤ ‖d.mixS T ψ v.μ v.ℱ (dRho (heisO ψ S T) θ τ Z f) x w‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualFormInvMemSubfield

theorem **dualForm_ringHom_inv** {F E : Type} [Field F] [Field E] (f : F →+* E) {n : ℕ} (A : Matrix (Fin n) (Fin n) F) : A⁻¹.map f = (A.map f)⁻¹

theorem **dualForm_inv_mem_subfield** {K : Type} [Field K] (K0 : Subfield K) {n : ℕ} (H : Matrix (Fin n) (Fin n) K) (hH : ∀ i j, H i j ∈ K0) (i j : Fin n) : H⁻¹ i j ∈ K0

theorem **isRatHerm_dualFrame** (K : Type) [Field K] [NumberField K] [StarRing K] (K0 : Subfield K) (n : ℕ) (H : Matrix (Fin n) (Fin n) K) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ) : IsRatHerm K0 H X → IsRatHerm K0 (H⁻¹)ᵀ (Formalism.HN.dualFrame X)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualIsotropicLineMemCompletionIff

theorem **idV_le_idAt** {a : FI D} {v : Pl D} {a : AE D} (ha : a ∈ idV a v) : a ∈ idAt D (· = v)

theorem **cA_mem_idAt** {v : Pl D} {a : AE D} (h : a ∈ idAt D (· = v)) : cA D a ∈ idAt D (· = v)

theorem **cA_mem_idV_one** {v : Pl D} {c : AE D} (hc : c ∈ idV (1 : FI D) v) : cA D c ∈ idV (1 : FI D) v

theorem **mul_incA_mem** {S : HSp D} {L : Lat S} {v : Pl D} {c : AE D} (hc : c ∈ idV (1 : FI D) v) {s : Fin S.rank → D.host.E} (hs : s ∈ L) : (fun a => c * incA D (s a)) ∈ latV S L (· = v)

theorem **hA_add_left** (S : HSp D) (w w' y : Fin S.rank → AE D) : hA S (w + w') y = hA S w y + hA S w' y

theorem **hA_neg_left** (S : HSp D) (w y : Fin S.rank → AE D) : hA S (-w) y = -hA S w y

theorem **hA_zero_left** (S : HSp D) (y : Fin S.rank → AE D) : hA S 0 y = 0

theorem **hA_add_right** (S : HSp D) (w y y' : Fin S.rank → AE D) : hA S w (y + y') = hA S w y + hA S w y'

theorem **hA_neg_right** (S : HSp D) (w y : Fin S.rank → AE D) : hA S w (-y) = -hA S w y

theorem **hA_zero_right** (S : HSp D) (w : Fin S.rank → AE D) : hA S w 0 = 0

theorem **hA_mem** {S : HSp D} {L : Lat S} (hL : IsSelfDualC S L) {v : Pl D} {w y : Fin S.rank → AE D} (hw : w ∈ latV S L (· = v)) (hy : y ∈ latV S L (· = v)) : hA S w y ∈ idV (1 : FI D) v

theorem **cA_cA'** (x : AE D) : cA D (cA D x) = x

theorem **cA_gA** (S : HSp D) (a b : Fin S.rank) : cA D (gA S a b) = incA D (S.gram b a)

theorem **mem_of_hA** {S : HSp D} {L : Lat S} (hL : IsSelfDualC S L) {v : Pl D} {y : Fin S.rank → AE D} (hyAt : ∀ a, y a ∈ idAt D (· = v)) (h : ∀ w ∈ latV S L (· = v), hA S w y ∈ idV (1 : FI D) v) : y ∈ latV S L (· = v)

theorem **dualLine_mem_iff** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (ɓ : ℕ → FI D) (v : Pl D), IsSelfDualC S L → MonoId d L L₀ ɓ v → ∀ i : Fin ρ, ɓ i ≠ 0 → ∀ t ∈ idAt D (· = v), colMul d.jY i t ∈ latV S L (· = v) ↔ cA D t ∈ idV (ɓ i)⁻¹ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualIsotropicLineMemCompletionIffStatement

theorem **_root_.GlobalTheory.T.dualIsotropicLine_mem_completion_iff** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (ɓ : ℕ → FI D) (v : Pl D), IsSelfDualC S L → MonoId d L L₀ ɓ v → ∀ i : Fin ρ, ɓ i ≠ 0 → ∀ t ∈ idAt D (· = v), colMul d.jY i t ∈ latV S L (· = v) ↔ cA D t ∈ idV (ɓ i)⁻¹ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualLatticeCosetIndicatorIsSchwartzBruhat

theorem **cosetInd_mem_SB** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (f : Module.Dual 𝓀[k] (Red L)), cosetInd q L f ∈ SB (n → k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualLatticeCosetIndicatorIsSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.dualLatticeCoset_indicator_isSchwartzBruhat** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (f : Module.Dual 𝓀[k] (Red L)), cosetInd q L f ∈ SB (n → k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.DualPickDot

theorem **dualPick_dot** {m : Type} [Fintype m] [DecidableEq m] (g : m → k) (hg : g ≠ 0) : dualPick g ·ᵥ g = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EZeroNe

theorem **places** (h : ColOut k w₀ w F) (t) (p) : p ∈ (w t).Q ↔ ∃ PP ∈ (pk W).Q t, p = PP.under (𝒪 D.host.F)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EZeroThetaDescentEval

theorem **good_fix** {K : Type} [Field K] [NumberField K] [StarRing K] {G : Type} [Group G] [Fintype G] {Pl0 : Type} (B : BlockDat K G Pl0) (ℓ : Lvl K G) (hX : ∀ χ ∈ ℓ.X, IsAbsIrrChar χ) (w : Fin (B.nV ℓ) → K) : ∑ g, (idemOf ℓ.X).coeff g • B.r g (B.S.basis.equivFun.symm (B.vE ℓ *ᵥ w)) = B.S.basis.equivFun.symm (B.vE ℓ *ᵥ w)

theorem **thDn_act** (hCR : GlobalTheory.Published.CuspRapidDecay) {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T0 : Type} {A : FamIn D.what₂.LL G Pl0 T0 D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (h : R.toLevCh.Is A.μ0) (hM : R.IsModels) (hC : R.IsChain) (hred : R.grp_red_R) (t : T0) (g : G) (x : (A.fam.R t).S.Forms) (v : (R.MY 0).Vec) : R.thDn t (R.jX t ((A.fam.R t).r g x)) (PiTensorProduct.tprod ℂ v) = actL (Yc D 0) (R.βY 0) (R.dat0 R.triv) (R.eG g) (R.thDn t (R.jX t x) (PiTensorProduct.tprod ℂ v))

theorem **goodProjector_fixes_compositeThetaLift_of_fixed** (hCR : GlobalTheory.Published.CuspRapidDecay) {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T0 : Type} {A : FamIn D.what₂.LL G Pl0 T0 D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (h : R.toLevCh.Is A.μ0) (hM : R.IsModels) (hC : R.IsChain) (hred : R.grp_red_R) (t : T0) (x : A.fam.amb.FR t) (hx : ∑ g, (idemOf A.X).coeff g • A.fam.amb.opR t (Sum.inr g) x = x) (v : (R.MY 0).Vec) : R.e0 (R.thDn t (R.jX t x) (PiTensorProduct.tprod ℂ v)) = R.thDn t (R.jX t x) (PiTensorProduct.tprod ℂ v)

theorem **GlobalTheory.GoodFix.e_zero_theta_descent_eval** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t (F : Fin ((W.Q.F.fc (cxW W).zs).Df1.nV t) → D.what₂.LL) (v : ((pk W).MY 0).Vec), (pk W).e0 ((pk W).thDn t ((pk W).jX t (W.Q.F.eV (cxW W).zs t F)) (PiTensorProduct.tprod ℂ v)) = (pk W).thDn t ((pk W).jX t (W.Q.F.eV (cxW W).zs t F)) (PiTensorProduct.tprod ℂ v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EZeroThetaDescentEvalStatement

theorem **_root_.GlobalTheory.T.e0_thDn_tprod_eq** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t (F : Fin ((W.Q.F.fc (cxW W).zs).Df1.nV t) → D.what₂.LL) (v : ((pk W).MY 0).Vec), (pk W).e0 ((pk

W).thDn t ((pk W).jX t (W.Q.F.eV (cxW W).zs t F)) (PiTensorProduct.tprod ℂ v)) = (pk W).thDn t ((pk W).jX t (W.Q.F.eV (cxW W).zs t F)) (PiTensorProduct.tprod ℂ v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EisensteinMaximalParabolicIff

theorem **coord_map** {ι : Type} [DecidableEq ι] (b : Module.Basis ι F E) {R R' : Type} [CommRing R] [Algebra F R] [CommRing R'] [Algebra F R'] (f : R →ₐ[F] R') (x : R ⊗[F] E) (i : ι) : equivFinsuppOfBasisRight b (Algebra.TensorProduct.map f (AlgHom.id F E) x) i = f (equivFinsuppOfBasisRight b x i)

theorem **tens_ext** {x y : AF F ⊗[F] E} (hw : ∀ w, compInf F E w x = compInf F E w y) (hv : ∀ v, compFin F E v x = compFin F E v y) : x = y

theorem **keeps_iff** {A : Type} [CommRing A] {n j : ℕ} {M : Matrix (Fin n) (Fin n) A} {C : Matrix (Fin n) (Fin j) A} {D : Matrix (Fin j) (Fin n) A} (h : D ∗ C = 1) : (∃ β : Matrix (Fin j) (Fin j) A, M ∗ C = C ∗ β) ↔ M ∗ C = C ∗ (D ∗ (M ∗ C))

theorem **keeps_inv** {A : Type} [CommRing A] {n j : ℕ} (M : GL (Fin n) A) {C : Matrix (Fin n) (Fin j) A} {β : Matrix (Fin j) (Fin j) A} (h : M.val ∗ C = C ∗ β) : ∃ β' : Matrix (Fin j) (Fin j) A, (M⁻¹).val ∗ C = C ∗ β'

theorem **keeps_one** (R : Type) [CommRing R] [Algebra F R] {σ : E →ₐ[F] E} {n j : ℕ} {J : Matrix (Fin n) (Fin n) E} (C : Matrix (Fin n) (Fin j) E) : Keeps R C (1 : HA R σ J)

theorem **mem_parFlag_of_local** {σ : E →ₐ[F] E} {n j : ℕ} {J : Matrix (Fin n) (Fin n) E} {C : Matrix (Fin n) (Fin j) E} {D : Matrix (Fin j) (Fin n) E} (h : D ∗ C = 1) (p : GAd σ J) (hw : ∀ w : InfinitePlace F, Keeps w.Completion C (projArch σ J w p)) (hv : ∀ v, Keeps (Fv F v) C (proj σ J v p)) : p ∈ parFlag σ J C fun _ => 0

theorem **iwasawa_fin** {σ : E →ₐ[F] E} (hc : IsCMExt F E σ) (hu : Algebra.Unramified (𝒪 F) (𝒪 E)) {n j : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) {C C' : Matrix (Fin n) (Fin j) E} (hCC : cT (σ : E →+* E) C ∗ J ∗ C = 0) (hCC' : cT (σ : E →+* E) C ∗ J ∗ C' = 1) (hC'C : cT (σ : E →+* E) C' ∗ J ∗ C = 1) (hC'C' : cT (σ : E →+* E) C' ∗ J ∗ C' = 0) {L : Submodule (𝒪 E) (Fin n → E)} [Submodule.IsLattice E L] (v : HeightOneSpectrum (𝒪 F)) (hL : IsSelfDualAt σ J L v) (y : GLoc σ J v) : ∃ q : GLoc σ J v, ∃ k ∈ stabLoc σ J L v, Keeps (Fv F v) C q ∧ y = q ∗ k

theorem **iwasawa_arch** {σ : E →ₐ[F] E} (hc : IsCMExt F E σ) {n j : ℕ} {J : Matrix (Fin n) (Fin n) E} {C : Matrix (Fin n) (Fin j) E} (hCC : cT (σ : E →+* E) C ∗ J ∗ C = 0) {a : GL (Fin n) (AF F ⊗[F] E)} (ha : IsMajorantFrame σ J a) (w : InfinitePlace F) (y : GArch σ J w) : ∃ q : GArch σ J w, ∃ k ∈ stabArch σ J w (bc (compInf F E w) a), Keeps w.Completion C q ∧ y = q ∗ k

theorem **iwasawa** {σ : E →ₐ[F] E} (hc : IsCMExt F E σ) (hu : Algebra.Unramified (𝒪 F) (𝒪 E)) {n j : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) {C C' : Matrix (Fin n) (Fin j) E} (hCC : cT (σ : E →+* E) C ∗ J ∗ C = 0) (hCC' : cT (σ : E →+* E) C ∗ J ∗ C' = 1) (hC'C : cT (σ : E →+* E) C' ∗ J ∗ C = 1) (hC'C' : cT (σ : E →+* E) C' ∗ J ∗ C' = 0) {L : Submodule (𝒪 E) (Fin n → E)} [Submodule.IsLattice E L] (hL : ∀ v, IsSelfDualAt σ J L v) {a : GL (Fin n) (AF F ⊗[F] E)} (ha : IsMajorantFrame σ J a) (g : GAd σ J) : ∃ p ∈ parFlag σ J C (fun _ => 0), ∃ k ∈ levelAll σ J L a, g = p ∗ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EisensteinSeriesContinuationModuli

theorem **unramified_host** (D : D7) : Algebra.Unramified (𝒪 D.host.F) (𝒪 D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ElementarySizeBoundsOfSizeHypotheses

theorem **elementary_size_bounds_of_size_hypotheses** : ∀ M : ℝ, ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT : ℝ), thr ≤ C' → thr ≤ Cl → StandingHypotheses D large₂ C' Cl Cg cb Cu cT CT → 1000 ≤ Cl ∧ M ≤ Cl ∧ M ≤ (D.R.m : ℝ) ∧ M ≤ (D.R.k : ℝ) ∧ D.host.rdE ≤ large₂.sc.b ∧ (D.R.k : ℝ) ^ M ≤ large₂.sc.b ∧ (D.R.r : ℝ) ≤ large₂.sc.b ∧ (D.R.m : ℝ) ^ 2 ≤ D.R.k ∧ M ∗ (D.R.m : ℝ) ^ 5 ∗ D.R.f ∗ large₂.sc.L ≤ D.R.k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ElementarySizeBoundsOfSizeHypothesesStatement

theorem **_root_.GlobalTheory.T.size_bounds_of_standingHypotheses** : ∀ M : ℝ, ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT : ℝ), thr ≤ C' → thr ≤ Cl → StandingHypotheses D large₂ C' Cl Cg cb Cu cT CT → 1000 ≤ Cl ∧ M ≤ Cl ∧ M ≤ (D.R.m : ℝ) ∧ M ≤ (D.R.k : ℝ) ∧ D.host.rdE ≤ large₂.sc.b ∧ (D.R.k : ℝ) ^ M ≤ large₂.sc.b ∧ (D.R.r : ℝ) ≤ large₂.sc.b ∧ (D.R.m : ℝ) ^ 2 ≤ D.R.k ∧ M ∗ (D.R.m : ℝ) ^ 5 ∗ D.R.f ∗ large₂.sc.L ≤ D.R.k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EmbFColFormMemMram

theorem **col_smul** (φ : ColsK I u) (b : Idx I.sel (I.tup a) (I.idx a) 0) (g : (I.stp u a 0).G b) (p : (I.stp u a 0).D b) : φ.1 ((I.stp u a 0).act b g p).1.1.deco = (I.stp u a 0).rho B.fields.emb (lamT (B.hcW u) 0) b g (φ.1 p.1.1.deco)

theorem **colForm_eq_zero** (φ : ColsK I u) (h0 : colForm I h u a hf φ = 0) : φ = 0

theorem **trTo_colForm** (φ : ColsK I u) (b : Idx I.sel (I.tup a) (I.idx a) 0) (p : (I.stp u a 0).D b) : (I.stp u a 0).trTo B.fields.emb B.embLL b p ((I.stp u a 0).embF B.fields.emb B.embLL (lamT (B.hcW u) 0) hf (colForm I h u a hf φ) ((I.stp u a 0).cl b p)) = evV B.fields.emb (1 : Matrix (Fin (B.R.N 0)) (Fin (B.R.N 0)) B.fields.E) (lamT (B.hcW u) 0) B.embLL (φ.1 p.1.1.deco)

theorem **embF_colForm_diaOp** (P : (B.primes X).vT u) (e : (𝒪 B.fields.E ⧸ P.1.asIdeal)ˣ) (φ : ColsK I u) : (I.stp u a 0).embF B.fields.emb B.embLL (lamT (B.hcW u) 0) hf (colForm I h u a hf (diaOp (Pr

theorem **embF_colForm_mem_Mram** (l : I.Laws) (φ : ColsK I u) : (I.stp u a 0).embF B.fields.emb B.embLL (lamT (B.hcW u) 0) hf (colForm I h u a hf φ) ∈ (I.atT l u a).Mram ↔ φ ∈ MramV (Pr

theorem **isCol_mem_ramSub_iff** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (_hX : B.fields.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) (x : ((I.fm l u a).lat 0).V) (φ : ColsK I u) (hc : IsCol I u a l x φ) : x ∈ (I.fm l u a).ramSub ↔ φ ∈ MramV (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EmbZeroTrivialWittDecomposition

theorem **eventually_charUnramAt** (D : D7) (χ : (AE D)ˣ →* Circle) (hχ : Continuous χ) : ∀ᶠ v : Pl D in Filter.cofinite, CharUnramAt D χ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EntryDatumTransportBadPrimeAnyTuple

theorem **entry_transport_any** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (hm : i + 2 < B.R.m) (al : (I.idx a).β i) (be : (I.idx a).β (i + 1)) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup a)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u (i)) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (i + 1)) (k y : TC.GA B (B.R.N (i))) (k' y' : TC.GA B (B.R.N (i + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1)) → TC.PtOf (y ∗ k) q.1.1 → TC.PtOf (y' ∗ k') q'.1.1 → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → Formalism.L2.Col.XiAt I.Ch I.sel (I.tup a) (I.cst a) i P ((I.idx a).h i al P) ((I.idx a).h (i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EntryDatumTransportBadPrimeAnyTupleStatement

theorem **_root_.GlobalTheory.T.entryDatum_transport_badPrime_anyTuple** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (hm : i + 2 < B.R.m) (al : (I.idx a).β i) (be : (I.idx a).β (i + 1)) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup a)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u (i)) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (i + 1)) (k y : TC.GA B (B.R.N (i))) (k' y' : TC.GA B (B.R.N (i + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1)) → TC.PtOf (y ∗ k) q.1.1 → TC.PtOf (y' ∗ k') q'.1.1 → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → Formalism.L2.Col.XiAt I.Ch I.sel (I.tup a) (I.cst a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqEvalOfDirDerivVanish

theorem **taylor_of_chain** (N : ℕ) : ∀ (G : ℕ → ℝ → ℂ) (_ : ∀ k t, HasDerivAt (G k) (G (k+1) t) t) (_ : G N = 0) (t : ℝ), G 0 t = ∑ k ∈ Finset.range N, G k 0 ∗ ((t ^ k / k.factorial : ℝ) : ℂ)

theorem **hasDerivAt_iter** {u : E → ℂ} (hu : ContDiff ℝ ∞ u) (k : ℕ) (y x : E) (t : ℝ) : HasDerivAt (fun t : ℝ => iteratedFDeriv ℝ k u (y + t • x) (fun _ => x)) (iteratedFDeriv ℝ (k+1) u (y + t • x) (fun _ => x)) t

theorem **foldr_replicate** {u : E → ℂ} (hu : ContDiff ℝ ∞ u) (x : E) (k : ℕ) : (List.replicate k x).foldr (fun y g => fun z => deriv (fun t : ℝ => g (z + t • y)) 0) u = fun y => iteratedFDeriv ℝ k u y (fun _ => x)

theorem **multilinear_diag_poly** {ι : Type} [Fintype ι] [DecidableEq ι] {k : ℕ} (M : ContinuousMultilinearMap ℝ (fun _ : Fin k => ι → ℝ) ℂ) : ∃ q : MvPolynomial ι ℂ, ∀ x : ι → ℝ, M (fun _ => x) = MvPolynomial.eval (fun i => (x i : ℂ)) q

theorem **eq_eval_of_dirDeriv_vanish** : ∀ (ι : Type) [Fintype ι] [DecidableEq ι] (N : ℕ) (u : (ι → ℝ) → ℂ), ContDiff ℝ ∞ u → (∀ L : List (ι → ℝ), L.length = N → L.foldr (fun y g => fun x => deriv (fun t : ℝ => g (x + t • y)) 0) u = 0) → ∃ p : MvPolynomial ι ℂ, ∀ x, u x = MvPolynomial.eval (fun i => (x i : ℂ)) p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqLatFrameOfIsSDFrame

theorem **gram_X'** {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} (hI : I.Ok) : (I.rec7 l Z).sp.X'.gram = 1

theorem **norm_zeta** {I : TC B X} (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) : star (zeta I l Z u) ·ᵥ (lineZ (I.rec7 l Z)).gram *ᵥ zeta I l Z u = ((((I.rec7 l Z).mem u).A.T : ℕ) : (I.rec7 l Z).host.E)

theorem **isLineVec_zeta** {I : TC B X} (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) : IsLineVec (lineZ (I.rec7 l Z)) (zeta I l Z u)

theorem **eq_latFrame_of_isSDFrame** {D : D7} {V : HSp D} {β : GL (Fin V.rank) (Lev.EAf D)} {Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)} (h : IsSDFrame V β Λ 1) : Λ = GlobalTheory.PlaceModel.latFrame V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqMulOfValLtOne

theorem **exists_eq_mul** (v : HeightOneSpectrum (𝒪 F)) (T b : 𝒪 F) (h1 : T ∈ v.asIdeal) (h2 : T ∉ v.asIdeal ^ 2) (hb : b ∈ v.asIdeal) : ∃ c ∈ v.adicCompletionIntegers F, algebraMap F (v.adicCompletion F) (b : F) = algebraMap F (v.adicCompletion F) (T : F) ∗ c

theorem **incl_eq_mul** [StarRing E] (hc : CMData F E) (v : HeightOneSpectrum (𝒪 F)) (P : Ideal (𝒪 E)) (hv : ∀ b : 𝒪 F, algebraMap (𝒪 F) (𝒪 E) b ∈ P → b ∈ v.asIdeal) (T : 𝒪 F) (t t' : 𝒪 E) (ht' : (t' : E) = star (t : E)) (htt : t ∗ t' = algebraMap (𝒪 F) (𝒪 E) T) (hco : IsCoprime t t') (ht : t ∈ P) (h1 : T ∈ v.asIdeal) (h2 : T ∉ v.asIdeal ^ 2) (a : 𝒪 E) (ha : a ∈ P) : ∃ c ∈ intV E v, Formalism.UG.inclE (F

theorem **eq_mul_of_val_lt_one** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.FinPl.SplitPt D k) (P : HeightOneSpectrum (𝒪 D.host.E)), Formalism.L2.NTheta.IsIdemAt p.e P → ∀ (T : ℕ) (t t' : 𝒪 D.host.E), (t' : D.host.E) = star (t : D.host.E) → t ∗ t' = (T : 𝒪 D.host.E) → IsCoprime t t' → t ∈ P.asIdeal → (T : 𝒪 D.host.F) ∈ p.v.asIdeal → (T : 𝒪 D.host.F) ∉ p.v.asIdeal ^ 2 → ∀ y : AE D, Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqOfPsiE

theorem **first_col** {K : Type} [CommRing K] {σ : ℕ} (T S : Matrix (Fin (σ + 1)) (Fin (σ + 1)) K) (h : S ∗ T = 1) (hl : T (Fin.last σ) (Fin.last σ) = 1) (hu : ∀ j j' : Fin (σ + 1), j'.1 = j.1 + 1 → ∀ a b, b ≤ a → T a j ∗ S j' b = 0) (hs : ∀ j j' : Fin (σ + 1), j'.1 = j.1 + 1 → (∑ a : Fin (σ + 1), ∑ b : Fin (σ + 1), if b.1 = a.1 + 1 then T a j ∗ S j' b else 0) = 1) : ∀ a, T a 0 = if a = 0 then 1 else 0

theorem **eq_of_psiE** {ψ : AddChar (AF D.host.F) ℂ} (hψ : ∃ s : AE D, psiE ψ s ≠ 1) {s s' : D.host.E} (h : ∀ t : AE D, psiE ψ (t ∗ incA D s) = psiE ψ (t ∗ incA D s')) : s = s'

theorem **exists_b** (hz' : z' ≠ 0) (i : Fin ρ) (t : AE D) : ∃ b : Mx D ρ S₀'.rank, b *ᵥ ratV z' = Pi.single i t

theorem **coefX_eq** (hz : d.i₀ *ᵥ z = d'.i₀ *ᵥ z') (n : GA D S) : coefX d z n = d.pX *ᵥ (mt S n *ᵥ (d'.j0 *ᵥ ratV z'))

theorem **coefX_nB'** (hz : d.i₀ *ᵥ z = d'.i₀ *ᵥ z') (b : Mx D ρ S₀'.rank) : coefX d z (d'.nB b) = (d.pX · d'.jX) *ᵥ (b *ᵥ ratV z')

theorem **e_eq** (hz : d.i₀ *ᵥ z = d'.i₀ *ᵥ z') (hz' : z' ≠ 0) (hN : ∀ n ∈ flagN d', n ∈ flagN d) : d'.e = d.e ∗ (d.qX ∗ d'.e)

theorem **first_eq** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} (d : HypDec S S₀ ρ) (d' : HypDec S S₀' ρ) {z : Fin S₀.rank → D.host.E} {z' : Fin S₀'.rank → D.host.E} (hz : d.i₀ *ᵥ z = d'.i₀ *ᵥ z') (hz0 : z ≠ 0) (hz' : z' ≠ 0) (hN : ∀ n, n ∈ flagN d ↔ n ∈ flagN d') (hc : ∀ n ∈ flagN d, psiBes ψ d 1 z n = psiBes ψ d' 1 z' n) (r : Fin ρ) (hr : r.1 = 0) (k : Fin S.rank) : d'.e k r = d.e k r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqOfSameDoubleCosetCompletionRowRepresentatives

theorem **reps_injOn** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqOfSameDoubleCosetCompletionRowRepresentativesStatement

theorem **_root_.GlobalTheory.T.eq_of_same_doubleCoset_completion_rowRepresentatives** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqOfSplitPartAgreeSupportedAtSplitPlace

theorem **mt_mul** (S : HSp D) (g g' : GA D S) : mt S (g * g') = mt S g * mt S g'

theorem **mt_one** (S : HSp D) : mt S 1 = 1

theorem **eq_of_parts** {A : Type*} [CommRing A] {e f m m' o : A} (k1 : (e + f) * (m - o) = m - o) (k2 : (e + f) * (m' - o) = m' - o) (k3 : e * m = e * m') (k4 : f * m' = f * m) : m = m'

theorem **mt_unit** (S : HSp D) (g : GA D S) : ((mt S g).map (cA D)).transpose * (HSpace.gram S).map (incA D) * mt S g = (HSpace.gram S).map (incA D)

theorem **gram_mul_conj_inv** (S : HSp D) (e : AE D) (g : GA D S) : (HSpace.gram S).map (incA D) * (cA D e • mt S g⁻¹) = ((e • mt S g).map (cA D)).transpose * (HSpace.gram S).map (incA D)

theorem **eq_of_split_part** : ∀ (D : D7) (S : HSp D) (e : AE D) (h h' : GA D S), IsSplitIdem D e → SuppAt e S h → SuppAt e S h' → (∀ i j, e * Formalism.L2.Theta.mt S h i j = e * Formalism.L2.Theta.mt S h' i j) → h = h'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqOfSplitPartAgreeSupportedAtSplitPlaceStatement

theorem **_root_.GlobalTheory.T.eq_of_splitPart_agree_supportedAtSplitPlace** : ∀ (D : D7) (S : HSp D) (e : AE D) (h h' : GA D S), IsSplitIdem D e → SuppAt e S h → SuppAt e S h' → (∀ i j, e * Formalism.L2.Theta.mt S h i j = e * Formalism.L2.Theta.mt S h' i j) → h = h'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqOneOfNormPowTwoPowSubOneLt

lemma **norm_sq_sub_one_ge** (w : ℂ) (h : ‖w - 1‖ < (1 / 2 : ℝ)) : (3 / 2 : ℝ) * ‖w - 1‖ ≤ ‖w ^ 2 - 1‖

theorem **eq_one_of_norm_pow_two_pow_sub_one_lt** (z : ℂ) (h : ∀ (m : ℕ), ‖z ^ (2 : ℕ) ^ m - (1 : ℂ)‖ < (1 / 2 : ℝ)) : z = (1 : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqSecAddCsec

theorem **eq_sec_add_csec** (q : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = q.v)) : a = q.sec (q.pr a) + cA D (q.sec (q.pr (cA D a)))

theorem **mem_intAt_iff** (q : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = q.v)) : a ∈ intAt D (· = q.v) ↔ q.pr a ∈ 𝒪[k] ∧ q.pr (cA D a) ∈ 𝒪[k]

theorem **val_xi** (q : SplitPt D k) (h : q.UnitAA) : valuation k ((ofSplit q).xi) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqSemiMeasOfCovolOne

theorem **map_eq_self_of_fd** (Γ : Subgroup N) [Countable Γ] (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain Γ 𝒩 ν) (hfin : ν 𝒩 ≠ ⊤) (φ : N ≃ₜ N) [(ν.map φ).IsMulLeftInvariant] (e : Γ → Γ) (he : Bijective e) (hφ : ∀ (g : Γ) (x : N), φ ((e g).1 * x) = g.1 * φ x) : ν.map φ = ν

theorem **rightInv_of_fd** (Γ : Subgroup N) [Countable Γ] (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain Γ 𝒩 ν) (hfin : ν 𝒩 ≠ ⊤) : ν.IsMulRightInvariant

theorem **invInv_of_rightInv** (ν : Measure N) [ν.IsHaarMeasure] [ν.IsMulRightInvariant] : ν.IsInvInvariant

theorem **eq_smul_of_rightInv** (κ l : Measure N) [κ.IsHaarMeasure] [κ.IsMulRightInvariant] [l.IsMulRightInvariant] [IsFiniteMeasureOnCompacts l] : ∃ c : ℝ≥0, l = c • κ

theorem **fd_ne_top** (Γ : Subgroup N) [Countable Γ] {K : Set N} (hK : IsCompact K) (hcov : ∀ g, ∃ γ ∈ Γ, γ * g ∈ K) (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain Γ 𝒩 ν) : ν 𝒩 ≠ ⊤

theorem **countable_subgroupOf** {G : Type} [Group G] (Γ N : Subgroup G) [Countable Γ] : Countable (Γ.subgroupOf N)

theorem **eq_one_of_bdd** {H : Type} [Group H] (χ : H →* ℝ≥0) (B : ℝ≥0) (hB : ∀ h, χ h ≤ B) (h : H) : χ h = 1

theorem **cj_coe** (h : H) (n : N) : (cj N s hnorm h n : G) = s h * n.1 * (s h)⁻¹

theorem **mk_surjective** : Surjective (mk N s)

theorem **mk_mul** (q q' : H × N) : mk N s q * mk N s q' = mk N s (q.1 * q'.1, q.2 * cj N s hnorm q.1 q'.2)

theorem **exists_homeo** (hs : Continuous s) (hp : ∃ p : (N ⊔ s.range : Subgroup G) → H, Continuous p ∧ ∀ (h : H) (n : N), p (mk N s (h, n)) = h) : ∃ e : H × N ≃ₜ (N ⊔ s.range : Subgroup G), ⇑e = mk N s

theorem **map_cj_of_mem** (Γ : Subgroup G) [Countable Γ] (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain (Γ.subgroupOf N) 𝒩 ν) (hfin : ν 𝒩 ≠ ⊤) {δ : H} (hδ : s δ ∈ Γ) : ν.map (cj N s hnorm δ) = ν

theorem **map_cj_of_cocompact** (Γ : Subgroup G) [Countable Γ] (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain (Γ.subgroupOf N) 𝒩 ν) (hfin : ν 𝒩 ≠ ⊤) {K : Set H} (hK : IsCompact K) (hcov : ∀ h, ∃ δ, s δ ∈ Γ ∧ δ * h ∈ K) (h : H) : ν.map (cj N s hnorm h) = ν

theorem **semiMeas_rightInv** (μ : Measure H) (ν : Measure N) [μ.IsHaarMeasure] [ν.IsHaarMeasure] [μ.IsMulRightInvariant] [ν.IsMulRightInvariant] : (semiMeas N s μ ν).IsMulRightInvariant

theorem **semiMeas_haar_inv** (hp : ∃ p : (N ⊔ s.range : Subgroup G) → H, Continuous p ∧ ∀ (h : H) (n : N), p (mk N s (h, n)) = h) (Γ : Subgroup G) [Countable Γ] (Λ : Subgroup H) [Countable Λ] (hΛ : ∀ δ ∈ Λ, s δ ∈ Γ) {K : Set H} (hK : IsCompact K) (hcov : ∀ h, ∃ δ ∈ Λ, δ * h ∈ K) (μ : Measure H) [μ.IsHaarMeasure] {ℱ : Set H} (hℱ : IsFundamentalDomain Λ ℱ μ) (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain (Γ.subgroupOf N) 𝒩 ν) (hfin : ν 𝒩 ≠ ⊤) : (semiMeas N s μ ν).IsHaarMeasure ∧ (semiMeas N s μ ν).IsInvInvariant

theorem **eq_semiMeas_of_covolOne** (hp : ∃ p : (N ⊔ s.range : Subgroup G) → H, Continuous p ∧ ∀ (h : H) (n : N), p (mk N s (h, n)) = h) (Γ : Subgroup G) [Countable Γ] (Λ : Subgroup H) [Countable Λ] (hrat : ∀ n ∈ N, ∀ h, n * s h ∈ Γ ↔ n ∈ Γ ∧ h ∈ Λ) (μ : Measure H) [μ.IsHaarMeasure] {ℱ : Set H} (hℱ : IsFundamentalDomain Λ ℱ μ) (hℱ1 : μ ℱ = 1) (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain (Γ.subgroupOf N) 𝒩 ν) (h𝒩1 : ν 𝒩 = 1) (κ : Measure (N ⊔ s.range : Subgroup G)) [κ.IsHaarMeasure] {ℛ : Set (N ⊔ s.range : Subgroup G)} (hℛ : IsFundamentalDomain (Γ.subgroupOf (N ⊔ s.range)) ℛ κ) (hℛ1 : κ ℛ = 1) : κ = semiMeas N s μ ν

theorem **lintegral_semiMeas** (μ : Measure H) (ν : Measure N) [SFinite ν] (F : G → ℝ≥0∞) (hF : Measurable F) : ∫⁻ r : (N ⊔ s.range : Subgroup G), F r ∂(semiMeas N s μ ν) = ∫⁻ h, ∫⁻ n, F (n.1 * s h) ∂ν ∂μ

theorem **lintegral_of_covolOne** (R : Subgroup G) (hR : N ⊔ s.range = R) (hp : ∃ p : R → H, Continuous p ∧ ∀ (h : H) (n : N) (hm : n.1 * s h ∈ R), p (n.1 * s h, hm) = h) (Γ : Subgroup G) [Countable Γ] (Λ : Subgroup H) [Countable Λ] (hrat : ∀ n ∈ N, ∀ h, n * s h ∈ Γ ↔ n ∈ Γ ∧ h ∈ Λ) (μ : Measure H) [μ.IsHaarMeasure] {ℱ : Set H} (hℱ : IsFundamentalDomain Λ ℱ μ) (hℱ1 : μ ℱ = 1) (ν : Measure N) [ν.IsHaarMeasure] {𝒩 : Set N} (h𝒩 : IsFundamentalDomain (Γ.subgroupOf N) 𝒩 ν) (h𝒩1 : ν 𝒩 = 1) (κ : Measure R) [κ.IsHaarMeasure] {ℛ :

Set R} (hℛ : IsFundamentalDomain (Γ.subgroupOf R) ℛ κ) (hℛ1 : κ ℛ = 1) (F : G → ℝ≥0∞) (hF : Measurable F) : ∫⁻ r : R, F r ∂κ = ∫⁻ h, ∫⁻ n, F (n.1 * s h) ∂ν ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqSmulOfUnitaryEquivariant

theorem **eval_blkPoly** (N n : ℕ) (h : Matrix (Fin n) (Fin n) ℂ) (v : Fin N × Fin N) : MvPolynomial.eval (fun x : Fin n × Fin n => h x.1 x.2) (blkPoly N n v) = blkMat N n h v.1 v.2

theorem **isBlock_blkMat** {N n : ℕ} (hN : N = n + 1) (h : Matrix (Fin n) (Fin n) ℂ) : GlobalTheory.BranchFormula.IsBlock hN (blkMat N n h) h 1

theorem **eq_blkMat_of_isBlock** {N n : ℕ} (hN : N = n + 1) {kt : Matrix (Fin N) (Fin N) ℂ} {k : Matrix (Fin n) (Fin n) ℂ} (H : GlobalTheory.BranchFormula.IsBlock hN kt k 1) : kt = blkMat N n k

theorem **pow_toNat_eq** (x : ℂ) (hx : x ≠ 0) (c m : ℤ) (hc : 0 ≤ c + m) : x ^ (c + m).toNat = x ^ m * x ^ c

theorem **det_eval_mvPolynomialX** (n : ℕ) (h : Matrix (Fin n) (Fin n) ℂ) : MvPolynomial.eval (fun x : Fin n × Fin n => h x.1 x.2) (Matrix.mvPolynomialX (Fin n) (Fin n) ℂ).det = h.det

theorem **det_ne_zero_of_unitary** {n : ℕ} {u : Matrix (Fin n) (Fin n) ℂ} (hu : u ∈ Matrix.unitaryGroup (Fin n) ℂ) : u.det ≠ 0

theorem **unitary_to_all** {N n : ℕ} (hN : N = n + 1) (lam : DomWeight N) (sg : DomWeight n) (u : delSq ℂ N lam →ₗ[ℂ] delSq ℂ n sg) (E : MvPolynomial (Fin N × Fin N) ℂ →ₗ[ℂ] MvPolynomial (Fin n × Fin n) ℂ) (hE : ∀ f : delSq ℂ N lam, E f.1 = (u f).1) (hu : ∀ k ∈ Matrix.unitaryGroup (Fin n) ℂ, ∀ kt : Matrix (Fin N) (Fin N) ℂ, GlobalTheory.BranchFormula.IsBlock hN kt k 1 → ∀ f f' : delSq ℂ N lam, f'.1 = Formalism.GradedWeylDivisibility.rowAct ℂ kt f.1 → k.det ^ lam.detShift • (u f').1 = k.det ^ sg.detShift • Formalism.GradedWeylDivisibility.rowAct ℂ k (u f).1) (g : Matrix (Fin n) (Fin n) ℂ) (hg : g.det ≠ 0) (f : MvPolynomial (Fin N × Fin N) ℂ) (hf : f ∈ delSq ℂ N lam) : g.det ^ lam.detShift • E (Formalism.GradedWeylDivisibility.rowAct ℂ (blkMat N n g) f) = g.det ^ sg.detShift • Formalism.GradedWeylDivisibility.rowAct ℂ g (E f)

theorem **eq_smul_of_unitary_equivariant** : ∀ (N n : ℕ) (hN : N = n + 1) (lam : DomWeight N) (sg : DomWeight n), Formalism.L2.Col.Br.Interlace hN lam sg → ∀ t t₀ : delSq ℂ N lam →ₗ[ℂ] delSq ℂ n sg, (∀ s, s = t ∨ s = t₀ → ∀ k ∈ Matrix.unitaryGroup (Fin n) ℂ, ∀ kt : Matrix (Fin N) (Fin N) ℂ, GlobalTheory.BranchFormula.IsBlock hN kt k 1 → ∀ f f' : delSq ℂ N lam, f'.1 = Formalism.GradedWeylDivisibility.rowAct ℂ kt f.1 → k.det ^ lam.detShift • (s f').1 = k.det ^ sg.detShift • Formalism.GradedWeylDivisibility.rowAct ℂ k (s f).1) → t₀ ≠ 0 → ∃ c : ℂ, t = c • t₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqZeroOfMulInt

theorem **exists_frame** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (hc : Fintype.card Y = N) (M : Matrix (Fin N) Y k) (hM : ∀ y, M *ᵥ y = 0 → y = 0) : ∃ F : GlobalTheory.LocMod.Frame k N Y, F.β = M

theorem **eq_zero_of_mul_int** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (a : k) (h : ∀ c : k, c * a ∈ 𝒪[k]) : a = 0

theorem **top_frames** : ∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (s : ℕ) (hs : s + 1 + 1 = D.R.m), d.Lz.rank = 1 → Fintype.card (Ix (Fin D.R.k ⊕ Fin (s + 1)) (Fin D.R.t) (s + 2)) = D.sp.X'.rank → ∀ (a : k) (z₀ : Fin d.Lz.rank → k), valuation k a = 1 → (∀ h, (τ s h).map (ofSplit q).pr *ᵥ z₀ = (-((ofSplit q).xi * a)) • fun i => (ofSplit q).pr ((d.bes s 0 h).ζ i)) → (∀ B : Matrix (Fin D.sp.X'.rank) (Fin D.sp.X'.rank) (AE D), B.map (toEAf D) = d.top.β.val → ∀ (y : Fin D.sp.X'.rank → k) (b : k), (fun i => q.sec (((d.oT (s + 1 + 1) hs).iA.map (ofSplit q).pr *ᵥ (B.map (ofSplit q).π *ᵥ y) + b • ((d.oT (s + 1 + 1) hs).iB.map (ofSplit q).pr *ᵥ z₀)) i)) ∈ GlobalTheory.FinPl.latV (D.sp.X (s + 1 + 1)) (d.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqZeroOfOrthFock

theorem **eq_zero_of_orth_fock** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T), x.Is → y.Is → ∀ vac, IsVacO ψ θ S T x.s y.s x.L y.L vac → ∀ (v : VolM D T.rank S.rank) f, IsThetaFunO ψ S T f → IsLatVac (heisO ψ S T) (latO S T x.L y.L fun _ => True) f → (∀ P, thPair v.μ v.𝒯 f (pureTensorO ψ θ S T x.s y.s x.B y.B vac P) = 0) → f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqZeroOfOrthogonalAllThetaFunctions

theorem **eq_zero_of_orthogonal_theta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (μ : Measure (WO S T)) (𝒯 : Set (WO S T)), μ.IsAddHaarMeasure → IsAddFundamentalDomain (ratL T S) 𝒯 μ → ∀ q : WO S T → ℂ, Continuous q → (∀ γ w, q (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * q w) → (∀ f, IsThetaFunO ψ S T f → ∫ w in 𝒯, q w * star (f w) ∂μ = 0) → q = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqZeroOfOrthogonalAllThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.eq_zero_of_orthogonal_all_thetaFunctions** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (μ : Measure (WO S T)) (𝒯 : Set (WO S T)), μ.IsAddHaarMeasure → IsAddFundamentalDomain (ratL T S) 𝒯 μ → ∀ q : WO S T → ℂ, Continuous q → (∀ γ w, q (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * q w) → (∀ f, IsThetaFunO ψ S T f → ∫ w in 𝒯, q w * star (f w) ∂μ = 0) → q = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqualIdentificationPairPriceLeSixFifths

theorem **exp_le_inv_one_sub_pg** (x : ℝ) (hx1 : x < 1) : Real.exp x * (1 - x) ≤ 1

theorem **one_factor_sharp** (k b g : ℝ) (f : ℕ) (hk : 1 ≤ k) (hb : 2 ≤ b) (hg : (k * b) ^ 8 ≤ g) (hf : (f : ℝ) ≤ k) : (1 + 3 / g) ^ f ≤ Real.exp (3 / 256)

theorem **exp_pair_le_six_fifths** : Real.exp (3 / 256) * Real.exp (3 / 256) ≤ 6 / 5

theorem **pgC_le_six_fifths** : ∀ (C' Cl Cg cb Cw cT CT C0 : ℝ) (D : D7) (large₂ : SizeData₂ D) (X : HidaSetting D) (Y : Full X), Formalism.MainNotions.NonGenericAndKTwoTheorems.KTwoHypotheses D large₂ C' Cl Cg cb Cw cT CT C0 → PgC Y ≤ 6 / 5

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EqualIdentificationPairPriceLeSixFifthsStatement

theorem **_root_.GlobalTheory.T.equalIdentificationPairPrice_le_six_fifths** : ∀ (C' Cl Cg cb Cw cT CT C0 : ℝ) (D : D7) (large₂ : SizeData₂ D) (X : HidaSetting D) (Y : Full X), Formalism.MainNotions.NonGenericAndKTwoTheorems.KTwoHypotheses D large₂ C' Cl Cg cb Cw cT CT C0 → PgC Y ≤ 6 / 5

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EquivariantHomMultiplicityFreeIrreducible

theorem **tensor_eq_of_coord_contract** {A B ι : Type} [AddCommGroup A] [Module ℂ A] [AddCommGroup B] [Module ℂ B] (b : Module.Basis ι ℂ B) (x y : A ⊗[ℂ] B) (h : ∀ i, TensorProduct.rid ℂ A (LinearMap.lTensor A (b.coord i) x) = TensorProduct.rid ℂ A (LinearMap.lTensor A

(b.coord i) y)) : x = y

theorem **contract_map_left** {A B : Type} [AddCommGroup A] [Module ℂ A] [AddCommGroup B] [Module ℂ B] (l : Module.Dual ℂ B) (f : A →ₗ[ℂ] A) (y : A ⊗[ℂ] B) : TensorProduct.rid ℂ A (LinearMap.lTensor A l (TensorProduct.map f LinearMap.id y)) = f (TensorProduct.rid ℂ A (LinearMap.lTensor A l y))

theorem **intertwining_eq_zero_of_isEmpty** {G V A : Type} [Group G] [AddCommGroup V] [Module ℂ V] [AddCommGroup A] [Module ℂ A] (ρ : Representation ℂ G V) (α : Representation ℂ G A) (hp : ρ.IsIrreducible) (hα : α.IsIrreducible) (hne : IsEmpty (ρ.Equiv α)) (f : ρ.IntertwiningMap α) : f = 0

theorem **intertwining_eq_smul** {G V A : Type} [Group G] [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] [AddCommGroup A] [Module ℂ A] (ρ : Representation ℂ G V) (α : Representation ℂ G A) (hp : ρ.IsIrreducible) (ι : ρ.Equiv α) (f : ρ.IntertwiningMap α) : ∃ c : ℂ, ∀ v, f v = c • ι v

theorem **equivariantHom_mem_of_stable** : ∀ (G H J : Type) [Group G] [Group H] [DecidableEq J] (A B : J → Type) [∀ j, AddCommGroup (A j)] [∀ j, Module ℂ (A j)] [∀ j, AddCommGroup (B j)] [∀ j, Module ℂ (B j)] [∀ j, FiniteDimensional ℂ (A j)] [∀ j, FiniteDimensional ℂ (B j)] (α : ∀ j, Representation ℂ G (A j)) (β : ∀ j, Representation ℂ H (B j)), (∀ j, (α j).IsIrreducible) → (∀ j, (β j).IsIrreducible) → (∀ j j', Nonempty ((α j).Equiv (α j')) → j = j') → ∀ (W : Type) [AddCommGroup W] [Module ℂ W] (ω : Representation ℂ (G × H) W), Nonempty ((Representation.directSum fun j => Representation.tprod ((α j).comp (MonoidHom.fst G H)) ((β j).comp (MonoidHom.snd G H))).Equiv ω) → ∀ (V : Type) [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (ρ : Representation ℂ G V), ρ.IsIrreducible → ∀ S : Submodule ℂ (V →ₗ[ℂ] W), (∀ T ∈ S, ∀ g v, T (ρ g v) = ω (g, 1) (T v)) → (∀ T ∈ S, ∀ h, ω (1, h) ∘ₗ T ∈ S) →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EquivariantHomMultiplicityFreeIrreducibleStatement

theorem **_root_.GlobalTheory.T.equivariantHom_multiplicityFree_irreducible** : ∀ (G H J : Type) [Group G] [Group H] [DecidableEq J] (A B : J → Type) [∀ j, AddCommGroup (A j)] [∀ j, Module ℂ (A j)] [∀ j, AddCommGroup (B j)] [∀ j, Module ℂ (B j)] [∀ j, FiniteDimensional ℂ (A j)] [∀ j, FiniteDimensional ℂ (B j)] (α : ∀ j, Representation ℂ G (A j)) (β : ∀ j, Representation ℂ H (B j)), (∀ j, (α j).IsIrreducible) → (∀ j, (β j).IsIrreducible) → (∀ j j', Nonempty ((α j).Equiv (α j')) → j = j') → ∀ (W : Type) [AddCommGroup W] [Module ℂ W] (ω : Representation ℂ (G × H) W), Nonempty ((Representation.directSum fun j => Representation.tprod ((α j).comp (MonoidHom.fst G H)) ((β j).comp (MonoidHom.snd G H))).Equiv ω) → ∀ (V : Type) [AddCommGroup V] [Module ℂ V] [FiniteDimensional ℂ V] (ρ : Representation ℂ G V), ρ.IsIrreducible → ∀ S : Submodule ℂ (V →ₗ[ℂ] W), (∀ T ∈ S, ∀ g v, T (ρ g v) = ω (g, 1) (T v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ErrXSPiAndVal

theorem **errXS_pi_and_val** : ∀ (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt) (b : CoreB K w Z), (Formalism.L2.Herm.toErrXS b.t.full.1).piG = w.px.piG ∧ (Formalism.L2.Herm.toErrXS b.t.full.1).piT = w.px.piT ∧ Formalism.L2.Host.ValLL0 (w.D Z) (TC.cf w.I w.l Z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ErrorExponentStatement

theorem **exists_castE** {D : D7} {I : Formalism.MainNotions.S.AutInst} (h : I = instOf D) : ∃ ιE : I.E ≃+* D.host.E, ∀ y, ιE y ≍ y

theorem **crdY_eq** {D : D7} {I : Formalism.MainNotions.S.AutInst} (h : I = instOf D) (ιE : I.E ≃+* D.host.E) (hι : ∀ y, ιE y ≍ y) (hr : (D.sp.Y 0).rank = I.n) (x : Fin I.n → I.E) : crdY h x = fun k => ιE (x (Fin.cast hr k))

theorem **hyp_linIndep** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : LinearIndependent D.host.E d.e.transpose

theorem **hyp_e_i₀** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.e.conjTranspose ∗ HSpace.gram S ∗ d.i₀ = 0

theorem **Scaled.tieψ** (hS : Scaled X h u d o ζ lam) {ψ : AddChar (AF ↥D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {θ : D.host.E} (hθ : θ ∗ lam⁻¹ = thetaS D) (n : GA D (D.sp.Y 0)) : star (psiBes ψ d θ (vecB o ζ) n) = X.psiInfU u ((RowsJoinPinned.iotaX X h) n).1 ∗ X.psiFinU u ((RowsJoinPinned.iotaX X h) n).2

theorem **besPer_domN** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) {ψ : AddChar (AF ↥D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) (θ : D.host.E) (μ 𝓕) (ν : Measure (flagN d)) (hν : ν.IsHaarMeasure) {𝒩 𝒩' : Set (flagN d)} (h𝒩 : IsFundamentalDomain (NFlagQ S d.e) 𝒩 ν) (h𝒩' : IsFundamentalDomain (NFlagQ S d.e) 𝒩' ν) (φ : GA D A → ℂ) (f : GA D S → ℂ) (hf : ∀ γ ∈ GQ S, ∀ g, f (γ ∗ g) = f g) : besPer ψ θ d o ζ μ 𝓕 ν 𝒩 φ f = besPer ψ θ d o ζ μ 𝓕 ν 𝒩' φ f

theorem **besPer_domA** {ρ : ℕ} (B : GlobalTheory.Unf.BesLev D ρ) {ψ : AddChar (AF ↥D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) {𝓕 : Set (GA D B.A)} (h𝓕 : IsFundamentalDomain (GQ B.A) 𝓕 B.vA.μ) (φ : GA D B.A → ℂ) (f : GA D B.V → ℂ) (hφ : ∀ γ ∈ GQ B.A, ∀ g, φ (γ ∗ g) = φ g) (hf : ∀ γ ∈ GQ B.V, ∀ g, f (γ ∗ g) = f g) : besPer ψ 1 B.d B.o B.ζ B.vA.μ 𝓕 B.vN.ν B.vN.𝒩 φ f = B.per ψ φ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EulerFactorFromDoubling

theorem **fockVec_of_vac** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {A H : HSp D} {sA : RealMat A} {sH : RealMat H} {f : WO A H → ℂ} (hf : IsThetaFunO ψ A H f) (hv : ∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO (sA τ) (sH τ)) f) : IsFockVec ψ θ A H sA sH f

theorem **sphFun_of_param** {V : HSp D} {Λ : Lat V} {Sf : Finset (FinPl D)} {f : GA D V → ℂ} {B : GlobalTheory.LFun.Fam D.host.E} (hf : GlobalTheory.LFun.IsParamFun D.sp.cm.sig V.gram Λ (Sf : Set (FinPl D)) f B) : SphFun V Λ Sf f

theorem **downNe_of_hypotheses** : GlobalTheory.Published.DoublingMethod → GlobalTheory.Published.RegularisedSiegelWeilFirstTerm → GlobalTheory.IndefiniteRowSideFullHecke.SimplePoleTwistedLift.SimplePole → ∀ D : D7, EulerRight D → DownNe D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EulerFactorFromDoublingStatement

theorem **_root_.GlobalTheory.T.euler_factor_from_doubling** : GlobalTheory.Published.DoublingMethod → GlobalTheory.Published.RegularisedSiegelWeilFirstTerm → GlobalTheory.IndefiniteRowSideFullHecke.SimplePoleTwistedLift.SimplePole → ∀ D : D7, EulerRight D → DownNe D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EulerProductConjugationStableParametersPos

theorem **lsd_pos** : ∀ (D : D7) (S : Finset (HeightOneSpectrum (𝒪 D.host.F))) (B : GlobalTheory.LFun.Fam D.host.E) (μ : (AE D)ˣ →* Circle) (r : ℕ) (κ s : ℝ), IsHeckeChar D μ → ResEta D μ 0 → GlobalTheory.LFun.UnrOff (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EulerProductConjugationStableParametersPosStatement

theorem **_root_.GlobalTheory.T.euler_product_conjugation_stable_parameters_pos** : ∀ (D : D7) (S : Finset (HeightOneSpectrum (𝒪 D.host.F))) (B : GlobalTheory.LFun.Fam D.host.E) (μ : (AE D)ˣ →* Circle) (r : ℕ) (κ s : ℝ), IsHeckeChar D μ → ResEta D μ 0 → GlobalTheory.LFun.UnrOff (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EvGSDOkOfEqOne

theorem **evGSD_ok_of_eq_one** : ∀ (X : Fields) [IsTotallyComplex X.K] (hX : X.Ok) (Pr : Primes X) (R : Ranks) (W : Wts X Pr.Word R) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (P : LevFam X.E X.K R.r Pr.vGs) (u : Pr.Word), Pr.Ok → mu = 1 → (evGSD W Ch sel mu P u).Ok

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EvalAtJetMap

theorem **evalAt_jetLin** (y' : ι → Matrix (Fin b) (Fin b) ℂ) (Λ : FP ι a b →₁[ℂ] ℂ) (Q : MvPolynomial ((ι × Fin b × Fin b) ⊕ Formalism.L2.Ker.DVar ι a b) ℂ) : evalAt y' (Formalism.L2.Ker.jetLin (fun m => Λ (monomial m 1)) Q) = Λ (aeval (Sum.elim (fun x : ι × Fin b × Fin b => (C (y' x.1 x.2.1 x.2.2) : FP ι a b)) X) Q)

theorem **evalAt_jetMap** (hab : a ≤ b) (y' : ι → Matrix (Fin b) (Fin b) ℂ) (Λ : FP ι a b →₁[ℂ] ℂ) (k : ι → ℕ) (P : PY ι a) : evalAt y' (Formalism.L2.Ker.jetMap (fun m => Λ (monomial m 1)) k P) = Λ (subY hab y' (P * ∏ τ, Formalism.L2.Ker.frDet ℂ a τ ^ k τ))

theorem **cSpan_eq_VV'** {m : ℕ} (lam : ι → DomWeight m) : CSpan lam = Formalism.L2.DIV.VV ℂ lam

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EvalVIndP

theorem **evalV_indP** (X : Set (Pl D)) (v : Pl D) : evalV D.host.F v (indP D X) = if v ∈ X then 1 else 0

theorem **indicatorAdele_mem_integralAdeles** (X : Set (Pl D)) : indP D X ∈ intAf D.host.F

theorem **finiteAdele_ext** {a b : Af D.host.F} (h : ∀ v, evalV D.host.F v a = evalV D.host.F v b) : a = b

theorem **indicatorAdele_singleton_mul_singleton** (X Y : Set (Pl D)) : indP D X * indP D Y = indP D (X ∩ Y)

theorem **indicatorAdele_union_of_disjoint** {X Y : Set (Pl D)} (h : Disjoint X Y) : indP D X + indP D Y = indP D (X ∪ Y)

theorem **indicatorAdele_univ** : indP D Set.univ = 1

theorem **indicatorAdele_empty** : indP D ∅ = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EvaluationFormAbsoluteConvergence

theorem **sFinite_of_haar** {G : Type} [Group G] [TopologicalSpace G] [MeasurableSpace G] [LocallyCompactSpace G] [SecondCountableTopology G] (μ : Measure G) [μ.IsHaarMeasure] : SFinite μ

theorem **lintegral_three_lt_top** {α β γ : Type} [MeasurableSpace α] [MeasurableSpace β] [MeasurableSpace γ] {μ : Measure α} {ν : Measure β} {κ : Measure γ} [SFinite μ] [SFinite ν] [SFinite κ] {φ : α → ℂ} {G : β → ℂ} {E : α → β → γ → ℂ} (hφ : Measurable φ) (hG : Measurable G) (hE : Measurable fun p : α × β × γ => E p.1 p.2.1 p.2.2) (hi : Integrable φ μ) {M : ℝ} (hM : ∀ a, ∫⁻ b, ‖G b‖ₑ * ∫⁻ c, ‖E a b c‖ₑ ∂κ ∂ν ≤ ENNReal.ofReal M) : ∫⁻ b, ∫⁻ c, ‖G b‖ₑ * ∫⁻ a, ‖φ a‖ₑ * ‖E a b c‖ₑ ∂μ ∂κ ∂ν < ∞

theorem **norm_theta_add** (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {γ : WO S T} (hγ : γ ∈ ratMx D T.rank S.rank) (w : WO S T) : ‖f (γ + w)‖ = ‖f w‖

theorem **yZero_mem** (S : HSp D) (ρ : ℕ) (z : Fin S.rank → D.host.E) : yZero S ρ z ∈ ratMx D S.rank ρ

theorem **incAdj_mem** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) {y : Mx D S.rank ρ} (hy : y ∈ ratMx D S.rank ρ) : d.incAdj S y ∈ ratMx D T.rank S.rank

theorem **val_theta** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (x : Mx D J.W.rank ρ) : IsThetaFunO ψ B.V₀ J.W₀ (val s Ψ x)

theorem **enorm_val_le** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (x : Mx D J.W.rank ρ) : ‖val s Ψ x 0‖ₑ ≤ ∫⁻ b in s.vT.𝓕, ‖B.d.mixS J.W ψ s.vS.μ s.vS.𝓕 Ψ x (xZero J.d · b)‖ₑ ∂s.vT.μ

theorem **val_flag** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (h₀ : GA D B.A) {n : GA D J.W} (hn : n ∈ flagN J.d) (h' : GA D J.W₀) (u : UX D ρ) : val s (s.om (embA B.d B.o h₀, n * J.d.emb0 h') Ψ) (mt J.W (n * J.d.emb0 h') · (xZero J.d · u.1.1)) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (val s Ψ (xZero J.d · u.1.1)))

theorem **val_ratA** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) {γ : GA D B.A} (hγ : γ ∈ GQ B.A) (u : UX D ρ) : val s (s.om (embA B.d B.o γ, 1) Ψ) (xZero J.d · u.1.1) 0 = val s Ψ (xZero J.d · u.1.1) 0

theorem **val_ratR** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) {γ : GA D J.W} (hγ : γ ∈ GQ J.W ⊓ jacGrp J.d) (u : UX D ρ) : ‖val s (s.om (1, γ) Ψ) (mt J.W γ · (xZero J.d · u.1.1)) 0‖ = ‖val s Ψ (xZero J.d · u.1.1) 0‖

theorem **continuous_val** (hΦ : IsThetaFunO ψ B.V J.W Φ) {X : Type} [TopologicalSpace X] {a : X → GA D B.A} {g : X → GA D J.W} {x : X → Mx D J.W.rank ρ} (ha : Continuous a) (hg : Continuous g) (hx : Continuous x) : Continuous fun p => val s (s.om (embA B.d B.o (a p), g p) Φ) (x p) 0

theorem **measurable_maj** (hφ : IsContAut B.A φ) (hG : RapidDec J.W G) (hΦ : IsThetaFunO ψ B.V J.W Φ) : Measurable (Function.uncurry (evalMaj s φ G Φ))

theorem **exists_mt_xZero** {r : GA D J.W} (hr : r ∈ jacGrp J.d) : ∃ v : UX D ρ, mt J.W r · xZero J.d = xZero J.d · v.1.1

theorem **exists_levi_conj** {r : GA D J.W} (hr : r ∈ jacGrp J.d) : ∃ v : UX D ρ, ∀ u : UX D ρ, ∃ t ∈ radX J.d ⊔ J.d.emb0.range, J.d.levi u.1 * r = t * J.d.levi (u * v).1

theorem **maj_left** (hG : RapidDec J.W G) (hΦ : IsThetaFunO ψ B.V J.W Φ) {γ : GA D J.W} (hγ : γ ∈ GQ J.W ⊓ jacGrp J.d) (g : GA D J.W) {r : GA D J.W} (hr : r ∈ jacGrp J.d) : evalMaj s φ G Φ (γ * g) (γ * r) = evalMaj s φ G Φ g r

theorem **maj_levi** {r : GA D J.W} (hr : r ∈ jacGrp J.d) : ∃ v : UX D ρ, ∀ g (u : UX D ρ), evalMaj s φ G Φ g (r * J.d.levi u.1) = evalMaj s φ G Φ g (J.d.levi (v * u).1)

theorem **enorm_fjAllFun_le** (hΦ : IsThetaFunO ψ B.V J.W Φ) (h : GA D J.W) (h' : GA D J.W₀) (n : flagN J.d) : ‖fjAllFun ψ J.d B.o B.ζ s.om₀ B.vA.μ B.vA.𝓕 φ (rTrans h G) (evalBF s (s.om (1, h) Φ)) h' n‖ₑ ≤ evalMaj s φ G Φ (n.1 * J.d.emb0 h' * h) (n.1 * J.d.emb0 h')

theorem **exists_bound** (hG : RapidDec J.W G) (hΦ : IsThetaFunO ψ B.V J.W Φ) {𝓑 : Set (GA D J.W)} (h𝓑 : IsFundamentalDomain (GQ J.W ⊓ jacGrp J.d : Subgroup (GA D J.W)) 𝓑 s.vW.μ) : ∃ M : ℝ, ∀ h₀, ∫⁻ g in 𝓑, ‖G g‖ₑ * ∫⁻ u : UX D ρ, ‖val s (s.om (embA B.d B.o h₀, g) Φ) (xZero J.d · u.1.1) 0‖ₑ ∂s.vU.ν ∂s.vW.μ ≤ ENNReal.ofReal M

theorem **lintegral_maj_lt_top** (hφ : IsContAut B.A φ) (hG : RapidDec J.W G) (hΦ : IsThetaFunO ψ B.V J.W Φ) {𝓑 : Set (GA D J.W)} (h𝓑 : IsFundamentalDomain (GQ J.W ⊓ jacGrp J.d : Subgroup (GA D J.W)) 𝓑 s.vW.μ) : ∫⁻ g in 𝓑, ∫⁻ u : UX D ρ, evalMaj s φ G Φ g (J.d.levi u.1) ∂s.vU.ν ∂s.vW.μ < ∞

theorem **exists_domain** (J : JacLev D ρ) : IsClosed (jacGrp J.d : Set (GA D J.W)) ∧ J.μR.IsHaarMeasure ∧ J.μR.IsInvInvariant ∧ ∃ 𝓡 : Set (jacGrp J.d), IsFundamentalDomain ((GQ J.W ⊓ jacGrp J.d).subgroupOf (jacGrp J.d)) 𝓡 J.μR ∧ ∀ F : GA D J.W → ℝ≥0∞, Measurable F → ∫⁻ r in 𝓡, F r ∂J.μR = ∫⁻ h in J.v0.𝓕, ∫⁻ n in J.vN.𝓝, F (n.1 * J.d.emb0 h) ∂J.vN.ν ∂J.v0.μ

theorem **evalBF_abs** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ →

RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∃ z', IsQuotIntL1 J.μR s.vW.μ (fun h => ((∫ u : UX D ρ, ‖J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1 * h) G) (evalBF s (s.om (1, (J.d.levi u.1 * h)) Φ))‖ ∂s.vU.ν : ℝ) : ℂ)) z'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EvaluationFormAbsoluteConvergenceStatement

theorem **_root_.GlobalTheory.T.evaluationForm_absolute_convergence** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∃ z', IsQuotIntL1 J.μR s.vW.μ (fun h => ((∫ u : UX D ρ, ‖J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1 * h) G) (evalBF s (s.om (1, (J.d.levi u.1 * h)) Φ))‖ ∂s.vU.ν : ℝ) : ℂ)) z'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EventuallyExtUnramTwoNotMem

theorem **eventually_extUnram_two_notMem** : ∀ D : D7, ∀ᶠ v : Pl D in cofinite, GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.ExtUnramAt D v ∧ (2 : 𝒪 D.host.F) ∉ v.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EventuallyIncAMulMemIntAt

theorem **idem_mul_finPart** (D : D7) (v : Pl D) {a : AE D} (ha : a ∈ idAt D (· = v)) : (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) * finPart D a = finPart D a

theorem **inv_smul_singleV_mem** {F : Type} [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (N : ℤ) (h : (N : 𝒪 F) ∉ v.asIdeal) : (N : F)⁻¹ • singleV F v 1 ∈ intAf F

theorem **tmul_eq_inv_smul_tmul** {F M P : Type} [Field F] [AddCommGroup M] [Module F M] [AddCommGroup P] [Module F P] (N : ℤ) (hN : (N : F) ≠ 0) (x : M) (c : P) : x ⊗ₜ[F] c = ((N : F)⁻¹ • x) ⊗ₜ[F] (N • c)

theorem **eventually_incA_mul_mem_intAt** : ∀ (D : D7) (c : D.host.E), ∀ᶠ v : Pl D in cofinite, ∀ a ∈ intAt D (· = v), incA D c * a ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EventuallyIntIn

theorem **exists_place** (D : D7) {S : Set (Pl D)} (hS : S.Finite) : ∃ PP : Formalism.L2.DIV.Pl D.host.E, (qOf PP).Prime ∧ ((qOf PP : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal ^ 2 ∧ PP.under (𝒪 D.host.F) ∉ S ∧ ∃ e, IsIdemAt e PP

theorem **int_dense** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) : (∀ a, (ofSplit q).pr a ∈ 𝒪[k] ↔ IntAt PP a) ∧ DenseRange (ofSplit q).pr

theorem **exists_compl** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) (h2 : ((qOf PP : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal ^ 2) : ∃ ϖ : kˣ, IsUnif ϖ ∧ IsCompl PP (qOf PP) (q.pr.comp (incA D)) ϖ

theorem **xi_unit** (hu : q.UnitAA) : valuation k (ofSplit q).xi = 1

theorem **exists_char** (hu : q.UnitAA) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ ψk, (ofSplit q).IsChar ψ ψk ∧ HasConductor ψk 1

theorem **Ranked.line** {c : Comm D} (h : Ranked c) : c.Lz.rank = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EventuallyNhdsOneAdelicMatrixCongruentOneAtPrime

theorem **congAt_eventually_nhds_one** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n N : ℕ) (P : HeightOneSpectrum (𝒪 E)), letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EventuallyNhdsOneAdelicMatrixCongruentOneAtPrimeStatement

theorem **_root_.GlobalTheory.T.eventually_nhds_one_adelicMatrixCongruentOneAtPrime** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n N : ℕ) (P : HeightOneSpectrum (𝒪 E)), letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.EventuallyOnto

theorem **eventually_onto** : ∀ (X : Type) (W : Submodule ℂ (X → ℂ)) [FiniteDimensional ℂ W] (T : ℕ → W →ₗ[ℂ] W), (∀ (ψ : W) (x : X), Tendsto (fun i => (T i ψ).1 x) atTop (𝓝 (ψ.1 x))) → ∃ i, Function.Surjective (T i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeCombinationMovesLastStepDatumToChart

theorem **exchange_posOne_last** : ∀ (D : D7) (c : Comm D) (v : VCom c) χ (Q : QAt v χ), c.Is → ∀ (h : Q.s + 2 * 0 + 2 = D.R.m) (s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h)), s₁.χW = χ Q.s → ∀ α (Tm : SB (Matrix (Q.X (Q.s + 1) ⊕ Fin 0) (Q.X Q.s ⊕ Fin (0 + 1)) Q.k) →ₗ[ℂ] (WO (c.V Q.s) (c.V (Q.s + 1)) → ℂ)), (ofSplit Q.q).IsModel c.ψ (Q.FY Q.s (0 + 1) (by omega)) (Q.FY (Q.s + 1) 0 (by omega)) Tm → ∀ Φ Φα : SB (Matrix (Q.X (Q.s + 1) ⊕ Fin 0) (Q.X Q.s ⊕ Fin (0 + 1)) Q.k), Φ.1 = Q.T.Phi Q.s 0 rfl → Φα.1 = om (Q.T.mu (Q.s + 1)) (Q.rep Q.s 0 rfl α) 1 (Q.T.Phi Q.s 0 rfl) → ∑ g ∈ (Q.εL Q.s α).support, Q.εL Q.s α g • s₁.om (g, 1) (Tm Φ) = Tm Φα

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeCombinationMovesLastStepDatumToChartStatement

theorem **_root_.GlobalTheory.T.exchangeCombination_moves_lastStepDatum_to_chart** : ∀ (D : D7) (c : Comm D) (v : VCom c) χ (Q : QAt v χ), c.Is → ∀ (h : Q.s + 2 * 0 + 2 = D.R.m) (s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h)), s₁.χW = χ Q.s → ∀ α (Tm : SB (Matrix (Q.X (Q.s + 1) ⊕ Fin 0) (Q.X Q.s ⊕ Fin (0 + 1)) Q.k) →ₗ[ℂ] (WO (c.V Q.s) (c.V (Q.s + 1)) → ℂ)), (ofSplit Q.q).IsModel c.ψ (Q.FY Q.s (0 + 1) (by omega)) (Q.FY (Q.s + 1) 0 (by omega)) Tm → ∀ Φ Φα : SB (Matrix (Q.X (Q.s + 1) ⊕ Fin 0) (Q.X Q.s ⊕ Fin (0 + 1)) Q.k), Φ.1 = Q.T.Phi Q.s 0 rfl → Φα.1 = om (Q.T.mu (Q.s + 1)) (Q.rep Q.s 0 rfl α) 1 (Q.T.Phi Q.s 0 rfl) → ∑ g ∈ (Q.εL Q.s α).support, Q.εL Q.s α g • s₁.om (g, 1) (Tm Φ) = Tm Φα

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeEqFiniteTranslateCombOfLevel

theorem **comb_sum** {G ι : Type} (T : Finset ι) (g : ι → G) (a : ℂ) (f : G → ℂ) : comb (∑ t ∈ T, Finsupp.single (g t) a) f = a * ∑ t ∈ T, f (g t)

theorem **exists_comb** {ψ : AddChar k Circle} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (g : (n → k) → GL (n ⊕ Fin c) k) (hg : Continuous g) (hg1 : g 0 = 1) (hadd : ∀ a b, g (a + b) = g a * g b) (U : Subgroup (GL (n ⊕ Fin c) k)) (hU : IsOpen (U : Set (GL (n ⊕ Fin c) k))) (W : Submodule 𝒪[k] (n → k)) (hW : W.IsLattice k) : ∃ e : GL (n ⊕ Fin c) k →ₐ₀ ℂ, ∀ f : GL (n ⊕ Fin c) k → ℂ, (∀ x b, g b ∈ U → f (g x * g b) = f (g x)) → (∀ x, x ∉ perp ψ W → f (g x) = 0) → ∫ x, f (g x) ∂(Measure.pi fun _ => ν) = comb e f

theorem **exchange_eq_comb** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ) (_ : 1 ≤ p) (_ : p < c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (U : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsOpen (U : Set (GL (n ⊕ Fin c) k))), ∃ e e' : GL (n ⊕ Fin c) k →ₐ₀ ℂ, ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (l : A

→ₗ[ℂ] ℂ) (v : A), (∀ u ∈ U, σ u v = v) → (IsNEquiv ψ (p + 1) σ l → exDown ν p σ l v = comb e fun g => l (σ g v)) ∧ (IsNEquiv ψ p σ l → exUp ν p σ l v = comb e' fun g => l (σ g v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeEqFiniteTranslateCombOfLevelStatement

theorem **_root_.GlobalTheory.T.exchange_eq_finiteTranslateComb_of_level** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ) (_ : 1 ≤ p) (_ : p < c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (U : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsOpen (U : Set (GL (n ⊕ Fin c) k))), ∃ e e' : GL (n ⊕ Fin c) k →₀ ℂ, ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (l : A →ₗ[ℂ] ℂ) (v : A), (∀ u ∈ U, σ u v = v) → (IsNEquiv ψ (p + 1) σ l → exDown ν p σ l v = comb e fun g => l (σ g v)) ∧ (IsNEquiv ψ p σ l → exUp ν p σ l v = comb e' fun g => l (σ g v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeToPositionOneIsPositionOne

theorem **toOne_lin** (ψ : AddChar k Circle) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] {A : Type} [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (hσ : IsSmoothRep σ) : ∀ m : ℕ, m + 1 ≤ c → ∀ l : A →ₗ[ℂ] ℂ, IsNEquiv ψ (m + 1) σ l → ∃ l₁ : A →ₗ[ℂ] ℂ, (∀ v, l₁ v = toOne ν σ (m + 1) l v) ∧ IsNEquiv ψ 1 σ l₁

theorem **toOne_bil** (ψ : AddChar k Circle) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] {A : Type} [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (hσ : IsSmoothRep σ) {P : Type} [AddCommGroup P] [Module ℂ P] (π : Representation ℂ (GL n k) P) : ∀ m : ℕ, m + 1 ≤ c → ∀ b : P →ₗ[ℂ] A →ₗ[ℂ] ℂ, IsPos ψ (m + 1) π σ b → ∃ b₁ : P →ₗ[ℂ] A →ₗ[ℂ] ℂ, (∀ φ v, b₁ φ v = toOne ν σ (m + 1) (b φ) v) ∧ IsPos ψ 1 π σ b₁

theorem **bilPos_iff** (ψ : AddChar k Circle) (μ : kˣ →* ℂˣ) (hμ : IsOpen (μ.ker : Set kˣ)) (p : ℕ) {A : Type} [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) : BilPos ψ μ p σ B ↔ IsPos ψ p (omRep μ hμ) σ B

theorem **GlobalTheory.PosOne.toOne_isPosOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ), (∀ l : A →ₗ[ℂ] ℂ, IsNEquiv ψ P σ l → ∃ l₁ : A →ₗ[ℂ] ℂ, (∀ v, l₁ v = toOne ν σ P l v) ∧ IsNEquiv ψ 1 σ l₁) ∧ ∀ (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ), BilPos ψ μ P σ B → ∃ B₁ : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ, (∀ Ψ v, B₁ Ψ v = toOne ν σ P (B Ψ) v) ∧ BilPos ψ μ 1 σ B₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeToPositionOneIsPositionOneStatement

theorem **_root_.GlobalTheory.T.exchangeToPositionOne_isPositionOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ), (∀ l : A →ₗ[ℂ] ℂ, IsNEquiv ψ P σ l → ∃ l₁ : A →ₗ[ℂ] ℂ, (∀ v, l₁ v = toOne ν σ P l v) ∧ IsNEquiv ψ 1 σ l₁) ∧ ∀ (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ), BilPos ψ μ P σ B → ∃ B₁ : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ, (∀ Ψ v, B₁ Ψ v = toOne ν σ P (B Ψ) v) ∧ BilPos ψ μ 1 σ B₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeToPositionOneUniformInLevel

theorem **comb_eq_lc** {G : Type} (e : G →₀ ℂ) (F : G → ℂ) : comb e F = Finsupp.linearCombination ℂ F e

theorem **comb_translate** {G : Type} [Group G] (e : G →₀ ℂ) (f : G → G →₀ ℂ) (F : G → ℂ) : comb e (fun g => comb (f g) (fun h => F (h * g))) = comb (e.sum fun g a => a • (f g).mapDomain (· * g)) F

theorem **mem_conjSub** {G : Type} [Group G] (U : Subgroup G) (g u : G) : u ∈ conjSub U g ↔ g⁻¹ * u * g ∈ U

theorem **isOpen_conjSub** {G : Type} [Group G] [TopologicalSpace G] [ContinuousMul G] (U : Subgroup G) (hU : IsOpen (U : Set G)) (g : G) : IsOpen (conjSub U g : Set G)

theorem **fixed_conjSub** {G A : Type} [Group G] [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ G A) (U : Subgroup G) (v : A) (hv : ∀ u ∈ U, σ u v = v) (g : G) : ∀ u ∈ conjSub U g, σ u (σ g v) = σ g v

theorem **toOne_comb** (ψ : AddChar k Circle) (hψc : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) (hsd : IsSelfDual ψ ν) : ∀ q : ℕ, q + 1 ≤ c → ∀ (U : Subgroup (GL (n ⊕ Fin c) k)), IsOpen (U : Set (GL (n ⊕ Fin c) k)) → ∃ e e' : GL (n ⊕ Fin c) k →₀ ℂ, ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (l : A →ₗ[ℂ] ℂ) (v : A), IsNEquiv ψ (q + 1) σ l → (∀ u ∈ U, σ u v = v) → toOne ν σ (q + 1) l v = comb e (fun g => l (σ g v)) ∧ l v = comb e' fun g => toOne ν σ (q + 1) l (σ g v)

theorem **GlobalTheory.PosOne.toOne_uniform** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (U : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsOpen (U : Set (GL (n ⊕ Fin c) k))), ∃ e e' : GL (n ⊕ Fin c) k →₀ ℂ, ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (l : A →ₗ[ℂ] ℂ) (v : A), IsNEquiv ψ P σ l → (∀ u ∈ U, σ u v = v) → toOne ν σ P l v = comb e (fun g => l (σ g v)) ∧ l v = comb e' fun g => toOne ν σ P l (σ g v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExchangeToPositionOneUniformInLevelStatement

theorem **_root_.GlobalTheory.T.exchangeToPositionOne_uniform_in_level** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (U : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsOpen (U : Set (GL (n ⊕ Fin c) k))), ∃ e e' : GL (n ⊕ Fin c) k →₀ ℂ, ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (l : A →ₗ[ℂ] ℂ) (v : A), IsNEquiv ψ P σ l → (∀ u ∈ U, σ u v = v) → toOne ν σ P l v = comb e (fun g => l (σ g v)) ∧ l v = comb e' fun g => toOne ν σ P l (σ g v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdaptedLatticeFrame

theorem **exists_frame_of_sign** {κ : Type} [Fintype κ] [DecidableEq κ] (e : κ ≃ Fin n) (A : Matrix (Fin n) κ R) (ε : κ → Bool) {X Z : Matrix (Fin n) (Fin r) R} (M N : Matrix κ (Fin r) R) (hA : cT s A * J * A = diagonal fun i => if ε i then 1 else -1) (hM : A * M = X) (hN : A * N = Z) (h : (diagonal fun i => if ε i then (1 : R) else -1) * N = M) : ∃ (b : GL (Fin n) R) (δ : Fin n → Bool), (cT s b.1 * J * b.1 = diagonal fun i => if δ i then 1 else -1) ∧ cT s (b⁻¹).1 * (b⁻¹).1 * X = J * Z

theorem **exists_frame_hyp** (hn : m + 2 * r = n) {X Y : Matrix (Fin n) (Fin r) R} {I : Matrix (Fin n) (Fin m) R} (ε : Fin m → Bool) {t : R} (ht : t + t = 1) (hs : s t = t) (XX : cT s X * J * X = 0) (YY : cT s Y * J * Y = 0) (YX : cT s Y * J * X = 1) (XY : cT s X * J * Y =

1) (IX : cT s I * J * X = 0) (IY : cT s I * J * Y = 0) (XI : cT s X * J * I = 0) (YI : cT s Y * J * I = 0) (II : cT s I * J * I = diagonal fun i => if ε i then 1 else -1) : ∃ (b : GL (Fin n) R) (δ : Fin n → Bool), (cT s b.1 * J * b.1 = diagonal fun i => if δ i then 1 else -1) ∧ cT s (b⁻¹).1 * (b⁻¹).1 * X = (J * Y : Matrix (Fin n) (Fin r) R) * diagonal fun _ : Fin r => t

theorem **exists_frame_hyp'** (hn : m + 2 * r = n) {X Y : Matrix (Fin n) (Fin r) R} {I : Matrix (Fin n) (Fin m) R} {G c : Matrix (Fin m) (Fin m) R} (ε : Fin m → Bool) {t : R} (ht : t + t = 1) (hs : s t = t) (XX : cT s X * J * X = 0) (YY : cT s Y * J * Y = 0) (YX : cT s Y * J * X = 1) (XY : cT s X * J * Y = 1) (IX : cT s I * J * X = 0) (IY : cT s I * J * Y = 0) (XI : cT s X * J * I = 0) (YI : cT s Y * J * I = 0) (II : cT s I * J * I = G) (hc : cT s c * G * c = diagonal fun i => if ε i then 1 else -1) : ∃ (b : GL (Fin n) R) (δ : Fin n → Bool), (cT s b.1 * J * b.1 = diagonal fun i => if δ i then 1 else -1) ∧ cT s (b⁻¹).1 * (b⁻¹).1 * X = (J * Y : Matrix (Fin n) (Fin r) R) * diagonal fun _ : Fin r => t

theorem **mulVec_mem_span_imLat** (P : Matrix (Fin n) (Fin m) E) {L : Submodule (𝒪 E) (Fin m → E)} (hL : Submodule.IsLattice E L) (u : Fin m → E) : P *ᵥ u ∈ Submodule.span E (imLat P L : Set (Fin n → E))

theorem **hE_eq** (x y : Fin n → E) : hE σ J x y = (fun i => σ (x i)) ·ᵥ J *ᵥ y

theorem **hE_mulVec** {P : Matrix (Fin n) (Fin m) E} {G : Matrix (Fin m) (Fin m) E} {Q : Matrix (Fin m) (Fin n) E} (h : G * Q = cT (σ : E →+* E) P * J) (u : Fin m → E) (y : Fin n → E) : hE σ J (P *ᵥ u) y = hE σ G u (Q *ᵥ y)

theorem **hE_add** (x x' y : Fin n → E) : hE σ J (x + x') y = hE σ J x y + hE σ J x' y

theorem **smul_col_mem** {X : Matrix (Fin n) (Fin r) E} {u : Fin r → E} (hu : u ∈ stdLattice E r) (i : Fin r) : u i • Xᵀ i ∈ imLat X (stdLattice E r)

theorem **exists_lattice_hyp** (hJ : J.det ≠ 0) {G : Matrix (Fin m) (Fin m) E} (hG : G.det ≠ 0) {L₀ : Submodule (𝒪 E) (Fin m → E)} (hL : Submodule.IsLattice E L₀) (hd : ∀ v, IsSelfDualAt σ G L₀ v) {X Y : Matrix (Fin n) (Fin r) E} {I : Matrix (Fin n) (Fin m) E} {Q : Matrix (Fin m) (Fin n) E} (XX : cT (σ : E →+* E) X * J * X = 0) (YY : cT (σ : E →+* E) Y * J * Y = 0) (YX : cT (σ : E →+* E) Y * J * X = 1) (XY : cT (σ : E →+* E) X * J * Y = 1) (XI : cT (σ : E →+* E) X * J * I = 0) (YI : cT (σ : E →+* E) Y * J * I = 0) (QX : Q * X = 0) (QY : Q * Y = 0) (QI : Q * I = 1) (hQ : G * Q = cT (σ : E →+* E) I * J) (h1 : X * (cT (σ : E →+* E) Y * J) + I * Q + Y * (cT (σ : E →+* E) X * J) = 1) : ∃ L : Submodule (𝒪 E) (Fin n → E), Submodule.IsLattice E L ∧ (∀ v, IsSelfDualAt σ J L v) ∧ ∀ z ∈ L, ∀ i, ((cT (σ : E →+* E) Y * J) *ᵥ z) i • Xᵀ i ∈ L ∧ ((cT (σ : E →+* E) X * J) *ᵥ z) i • Yᵀ i ∈ L

theorem **exists_arch_frame_hyp** (h : IsCMExt F E σ) (hn : m + 2 * r = n) {G : Matrix (Fin m) (Fin m) E} (hG : IsHermND σ G) {X Y : Matrix (Fin n) (Fin r) E} {I : Matrix (Fin n) (Fin m) E} (XX : cT (σ : E →+* E) X * J * X = 0) (YY : cT (σ : E →+* E) Y * J * Y = 0) (YX : cT (σ : E →+* E) Y * J * X = 1) (XY : cT (σ : E →+* E) X * J * Y = 1) (IX : cT (σ : E →+* E) I * J * X = 0) (IY : cT (σ : E →+* E) I * J * Y = 0) (XI : cT (σ : E →+* E) X * J * I = 0) (YI : cT (σ : E →+* E) Y * J * I = 0) (II : cT (σ : E →+* E) I * J * I = G) (w : InfinitePlace F) : ∃ (b : GL (Fin n) (w.Completion ⊗[F] E)) (δ : Fin n → Bool), (cT (sigR _ σ) b.1 * JR w.Completion J * b.1 = diagonal fun i => if δ i then 1 else -1) ∧ cT (sigR _ σ) (b⁻¹).1 * (b⁻¹).1 * X.map (inclR (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAddChar

theorem **hom_extend** {A B C : Type} [AddCommGroup A] [AddCommGroup B] [CommGroup C] (hC : ∀ n : ℕ, n ≠ 0 → ∀ c : C, ∃ d : C, d ^ n = c) (i : A →+ B) (hi : Function.Injective i) (f : A →+ Additive C) : ∃ g : B →+ Additive C, ∀ a, g (i a) = f a

theorem **circle_roots** (n : ℕ) (hn : n ≠ 0) (c : Circle) : ∃ d : Circle, d ^ n = c

theorem **units_roots** (n : ℕ) (hn : n ≠ 0) (c : ℂˣ) : ∃ d : ℂˣ, d ^ n = c

theorem **exists_addChar** (ϖ : kˣ) (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (r : 𝒪[k] →+* F) (hr : Function.Surjective r) (h0 : ∀ x : 𝒪[k], valuation k (x : k) < 1 → r x = 0) (α : AddChar F Circle) (hα : α ≠ 1) : ∃ ψ : AddChar k Circle, HasConductor ψ 1 ∧ ∀ x : 𝒪[k], ψ ((x : k) / ϖ) = α (r x)

theorem **exists_tame** (r : 𝒪[k] →+* F) (h0 : ∀ x : 𝒪[k], valuation k (x : k) < 1 → r x = 0) (χ : Fˣ →* ℂˣ) : ∃ μ : kˣ →* ℂˣ, IsTame μ ∧ ∀ u : (𝒪[k])ˣ, μ (Units.map (𝒪[k]).subtype.toMonoidHom u) = χ (Units.map r.toMonoidHom u)

theorem **reduction_spec** (P : HeightOneSpectrum (𝒪 E)) {a : E} (h : IntAt P a) : ∃ r, ∃ s ∉ P.asIdeal, algebraMap (𝒪 E) E s * a = algebraMap (𝒪 E) E r ∧ Ideal.Quotient.mk P.asIdeal s * redP P a = Ideal.Quotient.mk P.asIdeal r

theorem **exists_residue** (P : HeightOneSpectrum (𝒪 E)) (ι : E →+* k) (hint : ∀ a, ι a ∈ 𝒪[k] ↔ IntAt P a) (hd : DenseRange ι) : ∃ r : 𝒪[k] →+* 𝒪 E ⧸ P.asIdeal, Function.Surjective r ∧ (∀ x : 𝒪[k], r x = 0 ↔ valuation k (x : k) < 1) ∧ ∀ a (h : ι a ∈ 𝒪[k]), r (ι a, h) = redP P a

theorem **exists_tie** : ∀ (E K k : Type) [Field E] [NumberField E] [StarRing E] [Field K] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (ϖ : kˣ) (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K) (χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (ρ κ : ℕ), IsCompl P q ι ϖ → GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → ψ₀ ≠ 1 → ∃ S : SymData k (2 * ρ) κ, Comparison₂ P q ι σ₀ ψ₀ χ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdeleEquivFin

theorem **exists_adele_equiv_fin** (D : D7) (ι : (Af D.host.F ⊗[D.host.F] D.host.E) ≃+* FiniteAdeleRing (𝒪 D.host.E) D.host.E) (hι : ∀ x : D.host.E, ι ((1 : Af D.host.F) ⊗ₜ[D.host.F] x) = algebraMap D.host.E (FiniteAdeleRing (𝒪 D.host.E) D.host.E) x) (hint : ∀ y, y ∈ intAE D ↔ ∀ P : HeightOneSpectrum (𝒪 D.host.E), Valued.v (ι y P) ≤ 1) : ∃ Φ : AE D ≃+* AdeleRing (𝒪 D.host.E) D.host.E, Continuous Φ ∧ Continuous Φ.symm ∧ (∀ x : D.host.E, Φ (incA D x) = algebraMap D.host.E (AdeleRing (𝒪 D.host.E) D.host.E) x) ∧ (∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (e : AE D), IsPlIdem D e P → ∀ k : D.host.Eˣ, Φ (locE D e (k : D.host.E)) = ((localIdele D.host.E P (atFin P k) : Ideles D.host.E) : AdeleRing (𝒪 D.host.E) D.host.E)) ∧ ∀ z : AE D, ((Φ z).1 = 1 → infPart D z = 1) ∧ (Φ z).2 = ι (finPart D z)

theorem **read_of_equiv** (D : D7) (c : (AE D)ˣ →* Circle) (χ : HChar D.host.E) (h : ChrEq c χ) (Φ : AE D ≃+* AdeleRing (𝒪 D.host.E) D.host.E) (hΦ : Continuous Φ.symm) (h1 : ∀ x : D.host.E, Φ (incA D x) = algebraMap D.host.E (AdeleRing (𝒪 D.host.E) D.host.E) x) (h2 : ∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (e : AE D), IsPlIdem D e P → ∀ k : D.host.Eˣ, Φ (locE D e (k : D.host.E)) = ((localIdele D.host.E P (atFin P k) : Ideles D.host.E) : AdeleRing (𝒪 D.host.E) D.host.E)) (z : (AE D)ˣ) : ((c z : Circle) : ℂ) = ((χ.1 (Units.map Φ.toRingHom.toMonoidHom z) : ℂˣ) : ℂ)

theorem **comm_sigR** (D : D7) (w : Af D.host.F ⊗[D.host.F] D.host.E) : Algebra.TensorProduct.comm D.host.F (Af D.host.F) D.host.E (sigR (Af D.host.F) D.sp.cm.sig w) = Formalism.UG.sigR D.sp.cm.sig (Af D.host.F) (Algebra.TensorProduct.comm D.host.F (Af D.host.F) D.host.E w)

theorem **chr_at_detRoot** : ∀ (D : D7) (c : (AE D)ˣ →* Circle) (χ : Formalism.L2.Host.HChar D.host.E), ChrEq c χ → χ.TrivOn D.host.F → ∀ (V : HSp D) {n : ℕ} (hn : V.rank = n) (g : GA D V) (y : GL (Fin n) (Formalism.L2.DIV.AfE D.host.F D.host.E)), g ∈ GfC V → finM V hn g = (y : Matrix (Fin n) (Fin n) (Formalism.L2.DIV.AfE D.host.F D.host.E)) → ∀ x, IsDetRoot V g x → ∀ a : Neutral.B1.Ideles D.host.E, IsDetLift (host_cmData D.R D.host) y a → (c x : ℂ) = ((χ.1 (Chr.finPartI D.host.E a) : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdeleSupportedAtPlaceNotIntegral

theorem **trAt_tmul** (v : Pl D) (a : Af D.host.F) (x : D.host.E) : trAt D v (a ⊗ₜ[D.host.F] x) = algebraMap D.host.F (Fv D.host.F v) (Algebra.trace D.host.F D.host.E x) * evalV D.host.F v a

theorem **algebraMap_int_mem** (v : Pl D) (r : 𝒪 D.host.F) : algebraMap D.host.F (Fv D.host.F v) (r : D.host.F) ∈ v.adicCompletionIntegers D.host.F

theorem **trAt_int** (v : Pl D) {y : Af D.host.F ⊗[D.host.F] D.host.E} (hy : y ∈ intAE D) : trAt D v y ∈ v.adicCompletionIntegers D.host.F

theorem **evalV_single_same** (v : Pl D) (y : Fv D.host.F v) : evalV D.host.F v (singleV D.host.F v y) = y

theorem **evalV_single_ne** (v : Pl D) (y : Fv D.host.F v) {w : Pl D} (hw : w ≠ v) : evalV D.host.F w (singleV D.host.F v y) = 0

theorem **inv_not_mem** (v : Pl D) {π : 𝒪 D.host.F} (hπ : π ∈ v.asIdeal) (hπ0 : π ≠ 0) : (algebraMap D.host.F (Fv D.host.F v) (π : D.host.F))⁻¹ ∉ v.adicCompletionIntegers D.host.F

theorem **exists_not_intAt** : ∀ (D : D7) (v : Pl D), ∃ u ∈ idAt D (· = v), u ∉ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdeleSupportedAtPlaceNotIntegralStatement

theorem **_root_.GlobalTheory.T.exists_adele_supportedAt_place_not_integral** : ∀ (D : D7) (v : Pl D), ∃ u ∈ idAt D (· = v), u ∉ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicMajorantFrameForCartanInvolutions

theorem **exists_maj_cart** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (sA : RealMat A), IsInfFam θ → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → ∃ a : GL (Fin A.rank) (AE D), IsMaj A a ∧ ∀ τ, (compAt θ τ a.1)ᴴ * (gramAt D A τ * sA τ) * compAt θ τ a.1 = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicMajorantFrameForCartanInvolutionsStatement

theorem **_root_.GlobalTheory.T.exists_adelicMajorantFrame_for_cartanInvolutions** : ∀ (D : D7) (θ : InfFam D) (A : HSp D) (sA : RealMat A), IsInfFam θ → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → ∃ a : GL (Fin A.rank) (AE D), IsMaj A a ∧ ∀ τ, (compAt θ τ a.1)ᴴ * (gramAt D A τ * sA τ) * compAt θ τ a.1 = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicReadingLastStepTrivialOffLevel

theorem **offPl_finite** (I : TC B X) (u : B.Word) (a : I.ι) : (offPl I u a).Finite

theorem **reads_move'** (I : TC B X) {u : B.Word} {a : I.ι} {i : ℕ} {Ch : Type} {chOf : (I.stp u a i).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)} {U : Ch → Subgroup (TC.GA B (B.R.N i))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a i).Cl → TC.GA B (B.R.N i)} {st : ∀ c, (I.stp u a i).Stab c → GL (Fin (B.R.N i)) B.host.E} {k : Ch → TC.GA B (B.R.N i)} (R : I.Reads u a i chOf Lb U chi y st k) (k' : Ch → TC.GA B (B.R.N i)) (U' : Ch → Subgroup (TC.GA B (B.R.N i))) (chi' : ∀ β, U' β →* ℂˣ) (hU : ∀ β x, x ∈ U' β ↔ k' β * x * (k' β)⁻¹ ∈ U β) (hchi : ∀ β (x : U' β), chi' β x = chi β (k' β * x.1 * (k' β)⁻¹, (hU β x.1).1 x.2)) (hstd : ∀ β, ∃ p : DecoF B.fields.E (B.R.N i) ⌀ ⌀ ⌀, TC.PtOf ((k' β)⁻¹ * k β) p ∧ p.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i)) (hoff : ∀ β (v : HeightOneSpectrum (𝒪 B.host.F)), (B.T u : 𝒪 B.host.F) ∉ v.asIdeal → ¬ I.UnderQ a v → (a = I.mu0 v ∀ j,

theorem **exists_basePt_on** (S : StepLevel F E σ J T f bad) (k : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (hkU : k ∈ UPts σ J (FiniteAdeleRing (𝒪 F) F)) (hkV : ∀ v, bc (prV (E

theorem **exists_mv** {I : TC B X} (hI : I.Ok) (hP : TC.Pins B X) {l : I.Laws} {Z : TC.Later B (I.cols l)} {u : B.Word} {a : I.ι} {Ch : ℕ → Type} (A : TwDat I l Z u a Ch) (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) (S : Set (HeightOneSpectrum (𝒪 B.host.F))) (hSf : S.Finite) (hST : ∀ v ∈ S, (B.T u : 𝒪 B.host.F) ∉ v.asIdeal) : ∃ w : basePts (A.K i hi β).lev, (∀ v ∈ S, bc (prV (E

theorem **rInt_of_stab** {m : GL (Fin n) (AfE F E)} (h : m ∈ stabGL (latS (F

theorem **adAt_congr_bc** {P : HeightOneSpectrum (𝒪 E)} {x y : GL (Fin n) (AfE F E)} (e : bc (prV (E

theorem **isPtOf_move** (S : Set (HeightOneSpectrum (𝒪 F))) {g k k₁ : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (hp : IsPtOf hc (g * k) p) (hkU : k ∈ Formalism.L2.Herm.UstdAf hc n) (hkl : Formalism.L2.Herm.Carries k (Formalism.L2.Col.stdLat E n) (Formalism.L2.Col.stdLat E n)) (hk₁ : k₁ ∈ Formalism.L2.Herm.UstdAf hc n) (hS : ∀ v ∈ S, bc (prV (E

theorem **std_move'** (k w : GL (Fin n) (AfE F E)) (hwU : w ∈ Formalism.L2.Herm.UstdAf hc n) (hloc : ∀ v : HeightOneSpectrum (𝒪 F), bc (prV (E

theorem **val_le_one_of_carries** {k : GL (Fin n) (AfE F E)} (hkl : Formalism.L2.Herm.Carries k (Formalism.L2.Col.stdLat E n) (Formalism.L2.Col.stdLat E n)) (P : HeightOneSpectrum (𝒪 E)) (i j : Fin n) : Valued.v (adAt F P ((k : Matrix (Fin n) (Fin n) (AfE F E)) i j)) ≤ 1 ∧ Valued.v (adAt F P (((k⁻¹ : GL (Fin n) (AfE F E)) : Matrix (Fin n) (Fin n) (AfE F E)) i j)) ≤ 1

theorem **isRed_one_of_inv** {P : HeightOneSpectrum (𝒪 E)} {k : GL (Fin n) (AfE F E)} (hkl : Formalism.L2.Herm.Carries k (Formalism.L2.Col.stdLat E n) (Formalism.L2.Col.stdLat E n)) (h : LevRead.IsRed P k⁻¹ 1) : LevRead.IsRed P k 1

theorem **isRed_one_of_topPt** (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (k : TC.GA B (B.R.N B.R.m)) (hkl : Formalism.L2.Herm.Carries k (Formalism.L2.Col.stdLat B.host.E (B.R.N B.R.m)) (Formalism.L2.Col.stdLat B.host.E (B.R.N B.R.m))) (h : I.TopPt l Z u a k) : ∀ w ∈ (B.primes X).vGs, LevRead.IsRed w k 1

theorem **exists_gfC** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (y : UPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ g ∈ GfC V, GlobalTheory.AdForm.finM V hn g = y.1.1

theorem **mvEl_spec** (hI : I.Ok) (hP : TC.Pins B X) (A : TwDat I l Z u a Ch) (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) : (∀ v : HeightOneSpectrum (𝒪 B.host.F), bc (prV (E

theorem **mvLift_spec** (hI : I.Ok) (hP : TC.Pins B X) (A : TwDat I l Z u a Ch) (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) : mvLift hI hP A hA i hi β ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) (mvLift hI hP A hA i hi β) = (mvEl hI hP A hA i hi β).1.1

theorem **mvB_spec** (hI : I.Ok) (hP : TC.Pins B X) (A : TwDat I l Z u a Ch) (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) : (∀ v : HeightOneSpectrum (𝒪 B.host.F), bc (prV (E

theorem **mvG_eq** (hI : I.Ok) (hP : TC.Pins B X) (A : TwDat I l Z u a Ch) (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) : mvG hI hP A hA i β = mvB hI hP A hA i hi β

theorem **mvW_spec** (hI : I.Ok) (hP : TC.Pins B X) (A : TwDat I l Z u a Ch) (hA : A.Is) (i : ℕ) (hi : i ≤ B.R.m) (β : Ch i) : mvW hI hP A hA i β ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) (mvW hI hP A hA i β) = (mvEl hI hP A hA i hi β).1.1

theorem **T_mem_of_vT** (hPr : (B.primes X).Ok) (u : B.Word) {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ (B.primes X).vT u) : (B.T u : 𝒪 B.host.F) ∈ (P.under (𝒪 B.host.F)).asIdeal

theorem **not_offPl_SQ** (hI : I.Ok) {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ I.SQ) : P.under (𝒪 B.host.F) ∉ offPl I u a

theorem **finM_inv_of** {D : D7} {V : HSp D} {n : ℕ} {hn : V.rank = n} {W : GA D V} {w : GL (Fin n) (EF D)} (h : finM V hn W = w.1) : finM V hn W⁻¹ = (w⁻¹ : GL (Fin n) (EF D)).1

theorem **moved_is** (hI : I.Ok) (hP : TC.Pins B X) (A : TwDat I l Z u a Ch) (hA : A.Is) : (moved hI hP A hA).Is

theorem **exists_twDat_top_trivial** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → Nonempty (Unique (Ch B.R.m)) → ∃ A' : TwDat I l Z u a Ch, A'.Is ∧ (∀ i, i < B.R.m → ∀ β, A'.kb i β = A.kb i β) ∧ ∀ β (v : HeightOneSpectrum (𝒪 B.host.F)), ((B.T u : ℕ) : 𝒪 B.host.F) ∉ v.asIdeal → ¬ I.UnderQ a v → Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicReadingLastStepTrivialOffLevelStatement

theorem **_root_.GlobalTheory.T.exists_adelicReading_lastStep_trivial_off_level** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → Nonempty (Unique (Ch B.R.m)) → ∃ A' : TwDat I l Z u a Ch, A'.Is ∧ (∀ i, i < B.R.m → ∀ β, A'.kb i β = A.kb i β) ∧ ∀ β (v : HeightOneSpectrum (𝒪 B.host.F)), ((B.T u : ℕ) : 𝒪 B.host.F) ∉ v.asIdeal → ¬ I.UnderQ a v → Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicTraceCharacterSplitSectionNeOne

theorem **exists_psiV_ne_one** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (v : HeightOneSpectrum (𝒪 D.host.F)) : ∃ y, ψ (pairA D.host.F 0 (singleV D.host.F v y)) ≠ 1

theorem **algebraMap_injective** (D : D7) : Function.Injective (algebraMap (AF D.host.F) (AE D))

theorem **exists_psiE_sec_ne_one** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∃ a : k, GlobalTheory.MixPer.psiE ψ (p.sec a) ≠ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicTraceCharacterSplitSectionNeOneStatement

theorem **_root_.GlobalTheory.T.exists_adelicTraceCharacter_splitSection_ne_one** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∃ a : k, GlobalTheory.MixPer.psiE ψ (p.sec a) ≠ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicUnitaryConjSelfDualMajorantLevels

theorem **exists_proj_eq** (x : ∀ v, GLoc σ J v) (S : Finset (HeightOneSpectrum (𝒪 F))) : ∃ g : GAd σ J, (∀ v ∈ S, proj σ J v g = x v) ∧ (∀ v ∉ S, proj σ J v g = 1) ∧ ∀ w, projArch σ J w g = 1

theorem **exists_projArch_eq** (x : ∀ w, GArch σ J w) (S : Finset (InfinitePlace F)) : ∃ g : GAd σ J, (∀ w ∈ S, projArch σ J w g = x w) ∧ (∀ w ∉ S, projArch σ J w g = 1) ∧ ∀ v, proj σ J v g = 1

theorem **exists_conj_of_local** {L L' : Submodule (𝒪 E) (Fin n → E)} (hL : Submodule.IsLattice E L) (hL' : Submodule.IsLattice E L') (a' : GL (Fin n) (AF F ⊗[F] E)) (hf : ∀ v : HeightOneSpectrum (𝒪 F), ∃ g : GLoc σ J v, ∀ k : GLoc σ J v, k ∈ stabLoc σ J L' v ↔ g⁻¹ * k * g ∈ stabLoc σ J L v) (hi : ∀ w : InfinitePlace F, ∃ g : GArch σ J w, ∀ k : GArch σ J w, k ∈ stabArch σ J w (Formalism.UG.bc (compInf F E w) a') ↔ g⁻¹ * k * g ∈ stabArch σ J w (Formalism.UG.bc (compInf F E w) a)) : ∃ g : GAd σ J, ∀ k : GAd σ J, k ∈ levelAll σ J L' a' ↔ g⁻¹ * k * g ∈ levelAll σ J L a

theorem **exists_conj** : ∀ (D : D7) (T : HSp D) (L L' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (a a' : GL (Fin T.rank) (AE D)), IsLevel T L a → IsLevel T L' a' → ∃ g : GA D T, ∀ k : GA D T, k ∈ lev T L' a' ↔ g⁻¹ * k * g ∈ lev T L a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicUnitaryConjSelfDualMajorantLevelsStatement

theorem **_root_.GlobalTheory.T.exists_adelic_unitary_conj_selfDual_majorant_levels** : ∀ (D : D7) (T : HSp D) (L L' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (a a' : GL (Fin T.rank) (AE D)), IsLevel T L a → IsLevel T L' a' → ∃ g : GA D T, ∀ k : GA D T, k ∈ lev T L' a' ↔ g⁻¹ * k * g ∈ lev T L a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicUnitaryLiftOfChartBasePoint

theorem **exists_lift_of_ptOf** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (i : ℕ) (hi : i ≤ B.R.m) (k : TC.GA B (B.R.N i)), (∃ p : Formalism.L2.Col.DecoF B.fields.E (B.R.N i) ⌀ ⌀ ⌀, TC.PtOf k p ∧ p.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i)) → ∃ kA : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i), kA ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) kA = k.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAdelicUnitaryLiftOfChartBasePointStatement

theorem **_root_.GlobalTheory.T.exists_adelicUnitaryLift_of_chartBasePoint** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (i : ℕ) (hi : i ≤ B.R.m) (k : TC.GA B (B.R.N i)), (∃ p : Formalism.L2.Col.DecoF B.fields.E (B.R.N i) ⌀ ⌀ ⌀, TC.PtOf k p ∧ p.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i)) → ∃ kA : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i), kA ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) kA = k.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAtTypeAndOrd

theorem **intAt_iff** (a : E) : IntAt P a ↔ P.valuation E a ≤ 1

theorem **valuation_norm** (h : ((qOf P : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) : P.valuation E ((qOf P : ℕ) : E) = WithZero.exp (-1 : ℤ)

theorem **exists_compl** (h : ((qOf P : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) : letI

theorem **exists_atType** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (s : ℕ), P.1 ∈ Pr.vGs → R.m = s + 2 → cst.Ok → cst.sharp = true → R.m ≤ cst.A → (∀ a, IntAt P.1 a → cst.psi P (a / (qOf P.1 : X.E)) = psi0P Ch P.1 (redP P.1 a)) → ((qOf P.1 : ℕ) : 𝒪 X.E) ∉ P.1.asIdeal ^ 2 → ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (ι : X.E →+* k) (S : SymData k s R.k), AtType Ch sel mu cst σ₀ P ι S

theorem **exists_ord** : (∀ κ τ j : ℕ, ∃ e : XIx (Fin κ) (Fin τ) j ≃ Fin (κ + j * (τ + 1)), IsOrd xPos e) ∧ (∀ κ τ l j : ℕ, ∃ e : Ix (Fin κ ⊕ Fin l) (Fin τ) j ≃ Fin (κ + l + j * τ), IsOrd zPos e) ∧ (∀ (A : Type) (n c : ℕ) (pos : A → ℕ) (d : A ≃ Fin n), IsOrd pos d → ∃ e : A ⊕ Fin c ≃ Fin (n + c), IsOrd (yPos pos n) e) ∧ (∀ (A : Type) (n N : ℕ) (pos : A → ℕ) (d : A ≃ Fin n), IsOrd pos d → N = n + 1 → ∃ e : A ⊕ Unit ≃ Fin N, IsOrd (Sum.elim pos fun _ => N - 1) e)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsAvgForm

theorem **app_smul** {X V : Type} [TopologicalSpace X] [AddCommGroup V] [Module ℂ V] (l : SB X →ₗ[ℂ] V) {z : ℂ} (hz : z ≠ 0) (f : X → ℂ) : app l (z • f) = z • app l f

theorem **om_smul'** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (z : ℂ) (Φ : Matrix b a k → ℂ) : om μ g h (z • Φ) = z • om μ g h Φ

theorem **opE_smul** (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (z : ℂ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) : opE ψ ν p (z • Φ) = z • opE ψ ν p Φ

theorem **opEFJ_smul** (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (z : ℂ) (Φ₀ : (n' → k) → ℂ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) : opEFJ ψ ν p Φ₀ (z • Φ) = z • opEFJ ψ ν p Φ₀ Φ

theorem **tInt_left** {p : ℕ} (hp : 1 ≤ p) (hpc : p ≤ c) {ψ : AddChar k Circle} (hψ : Continuous ψ) (ν : Measure k) [ν.IsAddHaarMeasure] (μ : kˣ →* ℂˣ) {σ : Representation ℂ (GL (n ⊕ Fin c) k) A} {B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ} (hB : BilPos ψ μ p σ B) (F : A) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (x : GL n k) (u : GL (n ⊕ Fin c) k) (hu : u ∈ N k n c p) (h : GL (n ⊕ Fin c) k) : tInt ψ ν μ p σ B F Φ.1 (embX c x * u * h) = ((nabs k (Matrix.GeneralLinearGroup.det x) ^ ((c : ℝ) + 1 − 2 * p) : ℝ) : ℂ) * tInt ψ ν μ p σ B F Φ.1 h

theorem **tInt_right** (ψ : AddChar k Circle) (ν : Measure k) (μ : kˣ →* ℂˣ) (p : ℕ) (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (F : A) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (j : GL (n ⊕ Fin c) k) (l : ℂˣ) (h1 : om μ 1 j Φ = ((l⁻¹ : ℂˣ) : ℂ) • Φ) (h2 : σ j F = (l : ℂ) • F) (h : GL (n ⊕ Fin c) k) : tInt ψ ν μ p σ B F Φ (h * j) = tInt ψ ν μ p σ B F Φ h

theorem **avg_RFJ** {ρ : ℕ} (h1 : 1 ≤ ρ) (h2 : ρ < c) (ν : Measure k) [ν.IsAddHaarMeasure] (t : GL (n ⊕ Fin c) k → ℂ) (ht : ∀ (x : GL n k), ∀ u ∈ N k n c (ρ + 1), ∀ h, t (embX c x * u * h) = (((nabs k (Matrix.GeneralLinearGroup.det x))⁻¹ : ℝ) : ℂ) * t h) {r : GL (n ⊕ Fin c) k} (hr : r ∈ RFJ k n c ρ) (h : GL (n ⊕ Fin c) k) : ∫ β : n → k, t (colE c (ρ + 1) β * (r * h)) ∂(Measure.pi fun _ => ν) = ∫ β : n → k, t (colE c (ρ + 1) β * h) ∂(Measure.pi fun _ => ν)

theorem **exists_avg_form** {ρ : ℕ} (hcρ : c + 1 = 2 * ρ) {ψ : AddChar k Circle} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) {ν : Measure k} (hν : IsSelfDual ψ ν) {μ₁ μ₂ : kˣ →* ℂˣ} (hμ₂ : IsOpen (μ₂.ker : Set kˣ)) {τ : Representation ℂ (GL (n₂ ⊕ Fin c) k) A} {T : SB (Matrix n₁ n₀ k) →ₗ[ℂ] SB (Matrix n₂ n₁ k) →ₗ[ℂ] A →ₗ[ℂ] ℂ} (hT : TriPos ψ μ₁ μ₂ ρ τ T) (v : A) : ∃ (ρ₁ : Representation ℂ (GL (n₁ ⊕ Fin (c + 1)) k) (SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k))) (B : SB (Matrix n₁ n₀ k) →ₗ[ℂ] SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k) →ₗ[ℂ] ℂ), IsSrc μ₂ ρ₁ ∧ BilPos ψ μ₁ (ρ + 1) ρ₁ B ∧ ∀ (Φ₁ : Matrix (n₁ ⊕ Fin (c + 1)) (n₀ ⊕ Fin (c + 1 + 1)) k → ℂ) (Φ₂ : SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k)) h w, HasTensorVal (fun Ψ Φ₀ => app (T Ψ) (opEFJ ψ ν ρ Φ₀.1 (om μ₂ h 1 Φ₂.1)) v) (fl ψ ν μ₁ ρ (om μ₁ 1 h Φ₁)) w → w = ∫ β : n₁ → k, tInt ψ ν μ₁ (ρ + 1) ρ₁ B Φ₂ Φ₁ (colE (c + 1) (ρ + 1) β * h) ∂(Measure.pi

theorem **pair_last** {ρ : ℕ} (hcρ : c + 1 = 2 * ρ) {ψ : AddChar k Circle} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) {ν : Measure k} (hν : IsSelfDual ψ ν) {μ : kˣ →* ℂˣ} (hμ : IsOpen (μ.ker : Set kˣ)) {τ : Representation ℂ (GL (n ⊕ Fin c) k) A} {B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ} (hB : BilPos ψ μ ρ τ B) (F : A) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (j : GL (n ⊕ Fin c) k) (l : ℂˣ) (h1 : om μ 1 j Φ.1 = ((l⁻¹ : ℂˣ) : ℂ) • Φ.1) (h2 : τ j F = (l : ℂ) • F) {r : GL (n ⊕ Fin c) k} (hr : r ∈ R k n c ρ) (x : GL (n' ⊕ Fin (c + 1)) k) (y : GL (n ⊕ Fin c) k) (Φ₀ : SB (n' → k)) : app B (opEFJ ψ ν ρ Φ₀.1 (om μ x (r * y * j) Φ.1)) (τ (r * y * j) F) = app B (opEFJ ψ ν ρ Φ₀.1 (om μ x y Φ.1)) (τ y F)

theorem **tensorVal_inv** {ρ : ℕ} (hcρ : c + 1 = 2 * ρ) {ψ : AddChar k Circle} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) {ν : Measure k} (hν : IsSelfDual ψ ν) {μ₁ μ₂ : kˣ →* ℂˣ} (hμ₂ : IsOpen (μ₂.ker : Set kˣ)) {τ : Representation ℂ (GL (n₂ ⊕ Fin c) k) A} {T : SB (Matrix n₁ n₀ k) →ₗ[ℂ] SB (Matrix n₂ n₁ k) →ₗ[ℂ] A →ₗ[ℂ] ℂ} (hT : TriPos ψ μ₁ μ₂ ρ τ T) (F : A) {L₀ : Level (GL (n₀ ⊕ Fin (c + 1 + 1)) k)} {L₁ : Level (GL (n₁ ⊕ Fin (c + 1)) k)} {L₂ : Level (GL (n₂ ⊕ Fin c) k)} (Φ₁ : SB (Matrix (n₁ ⊕ Fin (c + 1)) (n₀ ⊕ Fin (c + 1 + 1)) k)) (Φ₂ : SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k)) (t₁ : HasTypes μ₁ L₀ L₁ Φ₁.1) (t₂ : HasTypes μ₂ L₁ L₂ Φ₂.1) (hF : ∀ j : L₂.J, τ (j : GL (n₂ ⊕ Fin c) k) F = (L₂.lam j : ℂ) • F) (a x y) {r} (hr : r ∈ RFJ k n₁ (c + 1) ρ) {j} (hj : j ∈ L₁.J) {r'} (hr' : r' ∈ R k n₂ c ρ) {j'} (hj' : j' ∈ L₂.J) {z z' : ℂ} (hz' : HasTensorVal (fun Ψ Φ₀ => app (T Ψ) (opEFJ ψ ν ρ Φ₀.1 (om μ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBadPrimeOverChartBadPlace

theorem **exists_SBof_over_sub** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)), v ∈ Formalism.L2.Forms.badG (I.rec7 l Z) v I.UnderQ a v → ∃ P : SBof I a, Formalism.L2.Col.IsOver v (GlobalTheory.TwU.seamP (sm I l Z) P.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBadPrimeOverChartBadPlaceStatement

theorem **_root_.GlobalTheory.T.exists_badPrime_over_chart_bad_place** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)), v ∈ Formalism.L2.Forms.badG (I.rec7 l Z) v I.UnderQ a v → ∃ P : SBof I a, Formalism.L2.Col.IsOver v (GlobalTheory.TwU.seamP (sm I l Z) P.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBasePointCarryingLatticeAndFrames

theorem **algebraMap_placeIdem_mem** {v : Pl D} {ε : AF D.host.F} (hε : IsPlaceIdem D v ε) : algebraMap (AF D.host.F) (AE D) ε ∈ idAt D (· = v)

theorem **idAt_mono** {P Q : Pl D → Prop} (h : ∀ v, P v → Q v) {a : AE D} (ha : a ∈ idAt D P) : a ∈ idAt D Q

theorem **mulVec_mem'** {I : Ideal (AE D)} {m n : ℕ} (A : Matrix (Fin m) (Fin n) (AE D)) {y : Fin n → AE D} (hy : ∀ i, y i ∈ I) : ∀ i, (A *ᵥ y) i ∈ I

theorem **mulVec_fix** {V : HSp D} {e : AE D} {ε : AF D.host.F} {u v : Pl D} (hε : IsPlaceIdem D u ε) (he : e + cA D e = algebraMap (AF D.host.F) (AE D) ε) {h : GA D V} (hs : SuppAt e V h) (huv : v ≠ u) {y : Fin V.rank → AE D} (hy : ∀ l, y l ∈ idAt D (· = v)) : mt V h *ᵥ y = y

theorem **pr_e_mul** (q : SplitPt D k) (a : AE D) : q.pr (q.e * a) = q.pr a

theorem **e_mul_eps** (q : SplitPt D k) {ε : AF D.host.F} (he : q.e + cA D q.e = algebraMap (AF D.host.F) (AE D) ε) : q.e * algebraMap (AF D.host.F) (AE D) ε = q.e

theorem **map_mt_eq_one** (q : SplitPt D k) (q' : SplitPt D k') (hne : q.v ≠ q'.v) {V : HSp D} {h : GA D V} (hs : SuppAt q'.e V h) : (mt V h).map (GlobalTheory.LocMod.ofSplit q).π = 1

theorem **xi_ne** (q : SplitPt D k) : (GlobalTheory.LocMod.ofSplit q).xi ≠ 0

theorem **latV_iff_hat'** (S : HSp D) (L : Lat S) (P : Pl D → Prop) (y : Fin S.rank → AE D) (hy : ∀ i, y i ∈ idAt D P) : y ∈ latV S L P ↔ (fun i => Formalism.IndefR.Lev.swapAE D (finPart D (y i))) ∈ GlobalTheory.IndefiniteRowSideFullHecke.FlatSpaceComponents.hat L

theorem **finE_mulVec'** (S : HSp D) (x : GA D S) (y : Fin S.rank → AE D) : (fun i => Formalism.IndefR.Lev.swapAE D (finPart D ((mt S x *ᵥ y) i))) = (Formalism.IndefR.Lev.finE S x).val *ᵥ (fun i => Formalism.IndefR.Lev.swapAE D (finPart D (y i)))

theorem **exists_base** (V : HSp D) (Λ L₀ : Lat V) (hΛ : IsSelfDualC V Λ) (hL₀ : IsSelfDualC V L₀) : ∃ t₀ ∈ GfC V, ∀ P, Moves V t₀ Λ L₀ P

theorem **exists_corr** (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (q : ∀ i, SplitPt D (kk i)) (hinj : Function.Injective (fun i => (q i).v)) (M : ∀ i, GL (Fin V.rank) (kk i)) (t₀ : GA D V) (s : Finset J) : ∃ g ∈ GfC V, (∀ v, (∀ i, v ≠ (q i).v) → ∀ y : Fin V.rank → AE

D, (∀ l, y l ∈ idAt D (· = v)) → mt V g *ᵥ y = y) ∧ ∀ i ∈ s, GlobalTheory.LocMod.MovedTo (GlobalTheory.LocMod.ofSplit (q i)) V (g * t₀) β (M i).val

theorem **GlobalTheory.FinPl.BasePoint.exists_move** : ∀ (D : D7) (V : HSp D) (Λ L₀ : Lat V) (β : GL (Fin V.rank) (Lev.EAf D)), IsSelfDualC V Λ → IsSelfDualC V L₀ → ∀ (J : Type) [Fintype J] (k : J → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, SplitPt D (k i)), Function.Injective (fun i => (q i).v) → ∀ M : ∀ i, GL (Fin V.rank) (k i), ∃ t ∈ GfC V, ∃ L : Lat V, IsSelfDualC V L ∧ (∀ P, Moves V t Λ L P) ∧ (∀ v, (∀ i, v ≠ (q i).v) → latV V L (· = v) = latV V L₀ (· = v)) ∧ ∀ i, GlobalTheory.LocMod.MovedTo (GlobalTheory.LocMod.ofSplit (q i)) V t β (M i).val

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBasePointCarryingLatticeAndFramesStatement

theorem **_root_.GlobalTheory.T.exists_base_point_carrying_lattice_and_frames** : ∀ (D : D7) (V : HSp D) (Λ L₀ : Lat V) (β : GL (Fin V.rank) (Lev.EAf D)), IsSelfDualC V Λ → IsSelfDualC V L₀ → ∀ (J : Type) [Fintype J] (k : J → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, SplitPt D (k i)), Function.Injective (fun i => (q i).v) → ∀ M : ∀ i, GL (Fin V.rank) (k i), ∃ t ∈ GfC V, ∃ L : Lat V, IsSelfDualC V L ∧ (∀ P, Moves V t Λ L P) ∧ (∀ v, (∀ i, v ≠ (q i).v) → latV V L (· = v) = latV V L₀ (· = v)) ∧ ∀ i, GlobalTheory.LocMod.MovedTo (GlobalTheory.LocMod.ofSplit (q i)) V t β (M i).val

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBesselEmbeddingEqLastGLRootExchange

theorem **gl₂_embX_eq_embA** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (g : GL Q.n₂ Q.k), ∃ a₀ : GA D S.B'.A, GlobalTheory.MixPer.embA S.B'.d S.B'.o a₀ = Q.gl₂ (embX (Q.c + 1) g) ∧ (ofSplit Q.q).IsLoc Q.F.F₂ a₀ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBesselEmbeddingEqLastGLRootExchangeStatement

theorem **_root_.GlobalTheory.T.exists_besselEmbedding_eq_lastGL_rootExchange** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (g : GL Q.n₂ Q.k), ∃ a₀ : GA D S.B'.A, GlobalTheory.MixPer.embA S.B'.d S.B'.o a₀ = Q.gl₂ (embX (Q.c + 1) g) ∧ (ofSplit Q.q).IsLoc Q.F.F₂ a₀ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBesselRationalCutoffFundamentalDomain

theorem **exists_cutoff_srat** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (Fs : Set X.B.I.GA), IsFundamentalDomain (Srat X u) Fs (X.B.I.muA X.B.μf) → ∃ β : X.B.I.GA → ENNReal, Measurable β ∧ (∀ x, ∑' σ : Srat X u, β ((σ : X.B.I.GA) * x) = 1) ∧ ∀ F : X.B.I.GA → ENNReal, Measurable F → (∀ γ ∈ Srat X u, ∀ x, F (γ * x) = F x) → ∫⁻ x in Fs, F x ∂(X.B.I.muA X.B.μf) = ∫⁻ x, β x * F x ∂(X.B.I.muA X.B.μf)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBesselRationalCutoffFundamentalDomainStatement

theorem **_root_.GlobalTheory.T.exists_besselRational_cutoff_fundamentalDomain** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (Fs : Set X.B.I.GA), IsFundamentalDomain (Srat X u) Fs (X.B.I.muA X.B.μf) → ∃ β : X.B.I.GA → ENNReal, Measurable β ∧ (∀ x, ∑' σ : Srat X u, β ((σ : X.B.I.GA) * x) = 1) ∧ ∀ F : X.B.I.GA → ENNReal, Measurable F → (∀ γ ∈ Srat X u, ∀ x, F (γ * x) = F x) → ∫⁻ x in Fs, F x ∂(X.B.I.muA X.B.μf) = ∫⁻ x, β x * F x ∂(X.B.I.muA X.B.μf)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBilinearEquivariantOrthogonalSummandLift

theorem **contAut_ker** (hW : IsWeilDatumO ψ S T χS χT om) {φ : GA D A → ℂ} {Ξ : WO S T → ℂ} (hφ : IsContAut A φ) (hΞ : IsThetaFunO ψ S T Ξ) (h : GA D T) (u : WO L T) : IsContAut A fun a => φ a * om (o.emb (a, 1), h) Ξ ((o.resS T).symm (0, u))

theorem **liftA_add_left** (hA : CoCpt A) (vA : Vol A) (hW : IsWeilDatumO ψ S T χS χT om) {φ φ' : GA D A → ℂ} {Ξ : WO S T → ℂ} (hφ : IsContAut A φ) (hφ' : IsContAut A φ') (hΞ : IsThetaFunO ψ S T Ξ) : liftA o om vA (φ + φ') Ξ = liftA o om vA φ Ξ + liftA o om vA φ' Ξ

theorem **liftA_add_right** (hA : CoCpt A) (vA : Vol A) (hW : IsWeilDatumO ψ S T χS χT om) {φ : GA D A → ℂ} {Ξ Ξ' : WO S T → ℂ} (hφ : IsContAut A φ) (hΞ : IsThetaFunO ψ S T Ξ) (hΞ' : IsThetaFunO ψ S T Ξ') : liftA o om vA φ (Ξ + Ξ') = liftA o om vA φ Ξ + liftA o om vA φ Ξ'

theorem **liftA_smul_left** (vA : Vol A) (c : ℂ) (φ : GA D A → ℂ) (Ξ : WO S T → ℂ) : liftA o om vA (c • φ) Ξ = c • liftA o om vA φ Ξ

theorem **liftA_smul_right** (vA : Vol A) (c : ℂ) (φ : GA D A → ℂ) (Ξ : WO S T → ℂ) : liftA o om vA φ (c • Ξ) = c • liftA o om vA φ Ξ

theorem **liftA_equiv** (vA : Vol A) (hW : IsWeilDatumO ψ S T χS χT om) {φ : GA D A → ℂ} {Ξ : WO S T → ℂ} (hφ : IsContAut A φ) (hΞ : IsThetaFunO ψ S T Ξ) (a : GA D A) : liftA o om vA (fun x => φ (x * a)) (om (o.emb (a, 1), 1) Ξ) = liftA o om vA φ Ξ

theorem **exists_linear_of_rel** {M N : Type} [AddCommGroup M] [Module ℂ M] [AddCommGroup N] [Module ℂ N] (P : Submodule ℂ M) (R : M → N → Prop) (hex : ∀ x ∈ P, ∃! y, R x y) (hlin : ∀ (c : ℂ) x y x' y', R x y → R x' y' → R (c • x + x') (c • y + y')) : ∃ G : M →ₗ[ℂ] N, (∀ x, ∃ x', R x' (G x)) ∧ ∀ x ∈ P, R x (G x)

theorem **exists_theta** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → CoCpt A → ∀ (vA : Vol A) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n), IsSumFrame o ζ α α₀ → ∃ lS : kˣ →* ℂˣ, (CharUnramAt D χT p.v → IsUnram lS) ∧ ∀ (LS : Lat S) (LT : Lat T) (f₀ : WO S T → ℂ), IsSelfDualC S LS → IsSelfDualC T LT → Vac ψ S T LS LT (· = p.v) f₀ → ∃ θ : GlobalTheory.FinPl.SrcGl.CA A →ₗ[ℂ] (Matrix n (n' ⊕ Fin 1) k → ℂ) →ₗ[ℂ] (GA D T → WO L T → ℂ), (∀ (v : GlobalTheory.FinPl.SrcGl.CA A) (Φ : Matrix n (n' ⊕ Fin 1) k → ℂ) (f : WO S T → ℂ), Over ψ α₀ β LS LT f₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBilinearEquivariantOrthogonalSummandLiftStatement

theorem **_root_.GlobalTheory.T.exists_bilinear_equivariant_orthogonalSummandLift** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → CoCpt A → ∀ (vA : Vol A) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n), IsSumFrame o ζ α α₀ → ∃ lS : kˣ →* ℂˣ, (CharUnramAt D χT p.v → IsUnram lS) ∧ ∀ (LS : Lat S) (LT : Lat T) (f₀ : WO S T → ℂ), IsSelfDualC S LS → IsSelfDualC T LT → Vac ψ S T LS LT (· = p.v) f₀ → ∃ θ : GlobalTheory.FinPl.SrcGl.CA A →ₗ[ℂ] (Matrix n (n' ⊕ Fin 1) k → ℂ) →ₗ[ℂ] (GA D T → WO L T → ℂ), (∀ (v : GlobalTheory.FinPl.SrcGl.CA A) (Φ : Matrix n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBilinearPositionMiddleVertex

theorem **exists_bilPos_mid** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] (c ρ : ℕ) (_ : c + 1 = 2 * ρ) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ₁ μ₂ : kˣ →* ℂˣ) (_ : IsOpen (μ₁.ker : Set kˣ)) (_ : IsOpen (μ₂.ker : Set kˣ)) (m₁ : Measure (GL n₁ k)) [m₁.IsHaarMeasure] (G₁ :

Measure (GL (n₁ ⊕ Fin (c + 1)) k)) [G₁.IsHaarMeasure] (m₂ : Measure (GL n₂ k)) [m₂.IsHaarMeasure] (G₂ : Measure (GL (n₂ ⊕ Fin c) k)) [G₂.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n₂ ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (T : SB (Matrix n₁ n₀ k) →ₗ[ℂ] SB (Matrix n₂ n₁ k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : TriPos ψ μ₁ μ₂ ρ σ T) (F : A) (ρ₁ : Representation ℂ (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBilinearPositionMiddleVertexStatement

theorem **_root_.GlobalTheory.T.exists_bilinear_position_middle_vertex** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] (c ρ : ℕ) (_ : c + 1 = 2 * ρ) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ₁ μ₂ : kˣ →* ℂˣ) (_ : IsOpen (μ₁.ker : Set kˣ)) (_ : IsOpen (μ₂.ker : Set kˣ)) (m₁ : Measure (GL n₁ k)) [m₁.IsHaarMeasure] (G₁ : Measure (GL (n₁ ⊕ Fin (c + 1)) k)) [G₁.IsHaarMeasure] (m₂ : Measure (GL n₂ k)) [m₂.IsHaarMeasure] (G₂ : Measure (GL (n₂ ⊕ Fin c) k)) [G₂.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n₂ ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (T : SB (Matrix n₁ n₀ k) →ₗ[ℂ] SB (Matrix n₂ n₁ k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : TriPos ψ μ₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBorelSphericalFunction

theorem **unrBorel_apply** (α : Fin N → ℂˣ) (b : borel k (Fin N)) : unrBorel k α b = ∏ i : Fin N, α i ^ ord k (diagU b i)

theorem **unrBorel_eq_one** (α : Fin N → ℂˣ) (b : borel k (Fin N)) (hb : b.1 ∈ KO k (Fin N)) : unrBorel k α b = 1

theorem **exists_isBorelSph** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (γ : Fin N → ℂ), (∀ i, γ i ≠ 0) → ∃ ψ : GL (Fin N) k → ℂ, IsBorelSph k γ ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBorelSphericalFunctionStatement

theorem **_root_.GlobalTheory.T.exists_borelSphericalFunction** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (N : ℕ) (γ : Fin N → ℂ), (∀ i, γ i ≠ 0) → ∃ ψ : GL (Fin N) k → ℂ, IsBorelSph k γ ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBorelSphericalSameDoubleCosetSumsZonal

theorem **exists_fixed_sum_eq** {ρ : Representation ℂ G X} {π : Representation ℂ G V} (hρ : IsSmooth ρ) {K : Subgroup G} (hK : IsCompact (K : Set G)) (f : ρ.IntertwiningMap π) (hf : Function.Surjective f) (E : X →ₗ[ℂ] ℂ) (hE : ∀ u : X, (∀ x ∈ K, ρ x u = u) → E u = 0 → u = 0) {v : V} (hv : v ∈ fixed π K) (l : Module.Dual ℂ V) (hlv : l v = 1) : ∃ z : X, (∀ x ∈ K, ρ x z = z) ∧ E z = 1 ∧ ∀ S : Finset (G ⧸ K), (∀ x ∈ K, ∀ c ∈ S, x • c ∈ S) → ∑ c ∈ S, l (π c.out v) = ∑ c ∈ S, E (ρ c.out z)

theorem **exists_isBorelSph_hAct_eq** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (V : Type) [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (Fin n) k) V) (α : Fin n → ℂˣ), HasSatake π α → ∀ ω : GL (Fin n) k → ℂ, IsZonal π (KO k (Fin n)) ω → ∃ φ : GL (Fin n) k → ℂ, IsBorelSph k (fun i : Fin n => (α i : ℂ)) ∗ (((resCard k : ℝ) ^ (((i : ℕ) : ℝ) - ((n : ℝ) - 1) / 2) : ℝ) : ℂ)) φ ∧ ∀ g, Formalism.UG.hAct (KO k (Fin n)) (Formalism.UG.dosetInd ℂ (KO k (Fin n)) g) ω 1 = Formalism.UG.hAct (KO k (Fin n)) (Formalism.UG.dosetInd ℂ (KO k (Fin n)) g) φ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBorelSphericalSameDoubleCosetSumsZonalStatement

theorem **_root_.GlobalTheory.T.exists_borel_spherical_same_double_coset_sums_zonal** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (V : Type) [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (Fin n) k) V) (α : Fin n → ℂˣ), HasSatake π α → ∀ ω : GL (Fin n) k → ℂ, IsZonal π (KO k (Fin n)) ω → ∃ φ : GL (Fin n) k → ℂ, IsBorelSph k (fun i : Fin n => (α i : ℂ)) ∗ (((resCard k : ℝ) ^ (((i : ℕ) : ℝ) - ((n : ℝ) - 1) / 2) : ℝ) : ℂ)) φ ∧ ∀ g, Formalism.UG.hAct (KO k (Fin n)) (Formalism.UG.dosetInd ℂ (KO k (Fin n)) g) ω 1 = Formalism.UG.hAct (KO k (Fin n)) (Formalism.UG.dosetInd ℂ (KO k (Fin n)) g) φ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBoundAbsLocalLatticeCompletionEntries

theorem **absFin_latLoc_le** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (v : HeightOneSpectrum (𝒪 F)), ∃ C : ℝ≥0, ∀ y ∈ latLoc F L v, ∀ i, absFin F v (y i) ≤ C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBoundAbsLocalLatticeCompletionEntriesStatement

theorem **_root_.GlobalTheory.T.exists_bound_abs_localLatticeCompletion_entries** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (v : HeightOneSpectrum (𝒪 F)), ∃ C : ℝ≥0, ∀ y ∈ latLoc F L v, ∀ i, absFin F v (y i) ≤ C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBoundLatLoc

theorem **abs_zero_le** (v : HeightOneSpectrum (𝒪 F)) (C : ℝ≥0) : absFin F v (0 : Fv F v ⊗[F] E) ≤ C

theorem **abs_neg_le** (v : HeightOneSpectrum (𝒪 F)) {C : ℝ≥0} {y : Fv F v ⊗[F] E} (h : absFin F v y ≤ C) : absFin F v (-y) ≤ C

theorem **abs_int_mul_le** (v : HeightOneSpectrum (𝒪 F)) {C : ℝ≥0} {c y : Fv F v ⊗[F] E} (hc : c ∈ intLoc F E v) (h : absFin F v y ≤ C) : absFin F v (c * y) ≤ C

theorem **exists_bound_rat** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hL : L.FG) (v : HeightOneSpectrum (𝒪 F)) : ∃ C : ℝ≥0, ∀ x ∈ L, ∀ i, absFin F v ((1 : Fv F v) ⊗ₜ[F] x i) ≤ C

theorem **exists_bound_latLoc** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (v : HeightOneSpectrum (𝒪 F)) : ∃ C : ℝ≥0, ∀ y ∈ latLoc F L v, ∀ i : Fin n, absFin F v (y i) ≤ C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBoxFinitePosProductHaarUnitaryGroups

theorem **haar_box_prod_finite_pos** : ∀ (ι : Type) [Fintype ι] (p q : ι → ℕ) (m : ∀ i, Measure (UpqT (p i) (q i))), (∀ i, (m i).IsHaarMeasure) → ∃ B : ∀ i, Set (UpqT (p i) (q i)), (∀ i, MeasurableSet (B i)) ∧ ∏ i, m i (B i) ≠ 0 ∧ ∏ i, m i (B i) ≠ ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsBoxFinitePosProductHaarUnitaryGroupsStatement

theorem **_root_.GlobalTheory.T.exists_box_finite_pos_product_haar_unitaryGroups** : ∀ (ι : Type) [Fintype ι] (p q : ι → ℕ) (m : ∀ i, Measure (UpqT (p i) (q i))), (∀ i, (m i).IsHaarMeasure) → ∃ B : ∀ i, Set (UpqT (p i) (q i)), (∀ i, MeasurableSet (B i)) ∧ ∏ i, m i (B i) ≠ 0 ∧ ∏ i, m i (B i) ≠ ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralAdaptedElement

theorem **facts** {c : Comm D} {v : VCom c} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} (o : TDat c k ι) {n : ℕ} (G : Fin n → GLoc v) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m) (hc : c.Is) (hS : ShapeOff v o) (PP : Fin n →

Formalism.L2.DIV.Pl D.host.E) (b : GA D (c.V 0)) (β : GL (Fin (c.V 0).rank) (Lev.EAf D)) (hloc : ∀ i, LocAt (G i) (PP i) b β) (Λ Lm : Lat (c.V 0)) (hfr : Formalism.L2.DIV.IsFrame β Λ) (hmv : ∀ P, Moves (c.V 0) b Λ Lm P) : Facts o G Lm (c.bes 0 D.R.rS h0)

theorem **sMoves_of_moves** {S : HSp D} {t g : GA D S} {Λ L L' M : Lat S} {p : GlobalTheory.FinPl.Pl D} {c : AE D} (h : Moves S t Λ L (· = p)) (h' : Moves S t Λ L' (· = p)) (hs : SMoves S c g L M p) : SMoves S c g L' M p

theorem **stab_match** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → Framed B → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∃ a ∈ GfC (D.sp.X 0), a ∈ Subgroup.center (GA D (D.sp.X 0)) ∧ GlobalTheory.LevZero.StabAt f B a₀ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralAdaptedElementStatement

theorem **_root_.GlobalTheory.T.exists_central_adapted_element** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → Framed B → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∃ a ∈ GfC (D.sp.X 0), a ∈ Subgroup.center (GA D (D.sp.X 0)) ∧ GlobalTheory.LevZero.StabAt f B a₀ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralBottomDatum

theorem **GlobalTheory.MiTop.levChi_mul_lev** {D : D7} (V : Formalism.BasicNotions.TH.HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ V.rank) {k k' : GA D V} (hk : k ∈ Lev.levG V β L) (hk' : k' ∈ Lev.levG V β L) : Lev.levChi V β L (k * k') = Lev.levChi V β L k * Lev.levChi V β L k'

theorem **GlobalTheory.MiTop.conj_in_level** {G : Type} [Group G] (J : Subgroup G) (χ : G → ℂ) (hm : ∀ a ∈ J, ∀ b ∈ J, χ (a * b) = χ a * χ b) (h0 : ∀ a, χ a ≠ 0) {k : G} (hk : k ∈ J) (j : G) : (j ∈ J ↔ k * j * k⁻¹ ∈ J) ∧ (j ∈ J → χ (k * j * k⁻¹) = χ j)

theorem **GlobalTheory.MiTop.matches_mul_lev** {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) {t k : GA D (D.sp.Y 0)} (ht : Matches W.S.X W.Q.instEq (rJ W) (rChi W) t) (hk : k ∈ rJ W) : Matches W.S.X W.Q.instEq (rJ W) (rChi W) (t * k)

theorem **GlobalTheory.MiTop.one_central_pin** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → 1 ≤ D.R.rS → ∃ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) ∧ ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), GlobalTheory.MiTop.PlacedP f B → GlobalTheory.MiTop.Framed B → ∃ d : BotDat W u, d.B = B ∧ d.b = (show GA D (D.sp.Y 0) from f.g.b 0) ∧ d.Pin z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralBottomDatumStatement

theorem **_root_.GlobalTheory.T.exists_central_bottom_datum** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → 1 ≤ D.R.rS → ∃ z ∈ GfC (D.sp.Y 0), z ∈ Subgroup.center (GA D (D.sp.Y 0)) ∧ ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), GlobalTheory.MiTop.PlacedP f B → GlobalTheory.MiTop.Framed B → ∃ d : BotDat W u, d.B = B ∧ d.b = (show GA D (D.sp.Y 0) from f.g.b 0) ∧ d.Pin z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralElementHomotheticGluedCores

theorem **mulVec_mem_idAt** {P : GlobalTheory.FinPl.Pl D → Prop} {m l : ℕ} (X : Matrix (Fin m) (Fin l) (AE D)) {y : Fin l → AE D} (hy : ∀ i, y i ∈ idAt D P) (i : Fin m) : (X *ᵥ y) i ∈ idAt D P

theorem **moves_inv** {S : HSp D} {g : GA D S} {L M : Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} (h : Moves S g L M P) : Moves S g⁻¹ M L P

theorem **moves_trans** {S : HSp D} {g h : GA D S} {L M N : Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} (h1 : Moves S g L M P) (h2 : Moves S h M N P) : Moves S (h * g) L N P

theorem **moves_conj** {S : HSp D} {h : GA D S} {L M : Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} (hM : Moves S h L M P) (a : GA D S) : Moves S (h * a * h⁻¹) M M P ↔ Moves S a L L P

theorem **sMoves_trans** {S : HSp D} {c : AE D} {g h : GA D S} {L M N : Lat S} {v : GlobalTheory.FinPl.Pl D} (hS : SMoves S c g L M v) (hM : Moves S h M N (· = v)) : SMoves S c (h * g) L N v

theorem **mem_gluedS_smul** (q : GlobalTheory.LatPair.Unif k) {P P' : GlobalTheory.LatPair.Pair k n} {g : GL n k} {s : k} (hs : s ≠ 0) (hg : g.val = s • (1 : Matrix n n k)) (hP : P' = g • P) (u : n → k) (t : Fin 1 → k) : Sum.elim u t ∈ GlobalTheory.LatPair.gluedS q 1 P.ker P.vec ↔ Sum.elim (s • u) t ∈ GlobalTheory.LatPair.gluedS q 1 P'.ker P'.vec

theorem **emb_central_frame** {p : SplitPt D k} {S A L : HSp D} {o : OrthDec S A L} {ζ : Fin L.rank → D.host.E} {β : GlobalTheory.FinPl.Frame p A n} {β₀ : GlobalTheory.FinPl.Frame p S (n ⊕ Fin 1)} (hsum : IsSumFrame o ζ β β₀) {a : GA D A} {c' : AE D} (ha : mt A a = c' • (1 : Matrix (Fin A.rank) (Fin A.rank) (AE D))) (y : n ⊕ Fin 1 → k) : (mt S (o.emb (a, 1))).map p.pr *ᵥ (β₀.b *ᵥ y) = β₀.b *ᵥ Sum.elim (fun i => p.pr c' * y (Sum.inl i)) (fun j => y (Sum.inr j))

theorem **exists_central_sMoves** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (n : Type) [Fintype n] [DecidableEq n] (β : GlobalTheory.FinPl.Frame p A n) (β₀ : GlobalTheory.FinPl.Frame p S (n ⊕ Fin 1)) (M M' : Lat S) (μ : D.host.E) (T : ℕ) (q : GlobalTheory.LatPair.Unif k) (P : GlobalTheory.LatPair.Pair k n) (a₀ : GA D A), 3 ≤ Fintype.card n → IsSumFrame o ζ β β₀ → IsSelfDualC S M → IsSelfDualC S M' → β₀.IsLat M (GlobalTheory.LatPair.gluedS q 1 P.ker P.vec) → μ ≠ 0 → LineAt (vecB o ζ) μ M' p.v → star ζ ·ᵥ L.gram *ᵥ ζ = −(μ * star μ * (T : D.host.E)) → valuation k (p.pr (incA D (T : D.host.E))) = valuation k ((q.1 : 𝒪[k]) : k) → (∀ a ∈ grpAt A (· = p.v), Moves S (o.emb (a, 1)) M M (· = p.v) ↔ Moves S (o.emb (a₀ * a * a₀⁻¹, 1)) M' M' (· = p.v)) → ∃ (a : G

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralElementHomotheticGluedCoresStatement

theorem **_root_.GlobalTheory.T.exists_central_element_homothetic_gluedCores** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (n : Type) [Fintype n] [DecidableEq n] (β : GlobalTheory.FinPl.Frame p A n) (β₀ : GlobalTheory.FinPl.Frame p S (n ⊕ Fin 1)) (M M' : Lat S) (μ : D.host.E) (T : ℕ) (q : GlobalTheory.LatPair.Unif k) (P : GlobalTheory.LatPair.Pair k n) (a₀ : GA D A), 3 ≤ Fintype.card n → IsSumFrame o ζ β β₀ → IsSelfDualC S M → IsSelfDualC S M' → β₀.IsLat M (GlobalTheory.LatPair.gluedS q 1 P.ker P.vec) → μ ≠ 0 → LineAt (vecB o ζ) μ M' p.v → star ζ ·ᵥ L.gram *ᵥ ζ = −(μ * star μ * (T : D.host.E)) → valuation k (p.pr (incA D (T : D.host.E))) = valuation k ((q.1 : 𝒪[k]) : k) → (∀ a ∈ grpAt A (· = p.v), Moves S (o.emb (a, 1)) M M (· = p.v) ↔ Moves S (o.emb (a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralGlueLocalHomotheties

theorem **mt_mul** (S : HSp D) (a b : GA D S) : Formalism.L2.Theta.mt S (a * b) = Formalism.L2.Theta.mt S a * Formalism.L2.Theta.mt S b

theorem **mt_one** (S : HSp D) : Formalism.L2.Theta.mt S 1 = 1

theorem **mem_congGrp** {V : HSp D} {I : Ideal (AE D)} {g : GA D V} : g ∈ GlobalTheory.FinPl.congGrp V I ↔ ∀ i j, Formalism.L2.Theta.mt V g i j - (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D)) i j ∈ I

theorem **idAt_le_idOff** {v w : GlobalTheory.FinPl.Pl D} (h : v ≠ w) : idAt D (· = w) ≤ GlobalTheory.FinPl.idOff D (· = v)

theorem **mulVec_fix** {S : HSp D} {v w : GlobalTheory.FinPl.Pl D} (hvw : v ≠ w) {h : GA D S} (hh : h ∈ grpAt S (· = w)) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D (· = v)) : Formalism.L2.Theta.mt S h *ᵥ y = y

theorem **scal_mulVec** {n : ℕ} (c : AE D) (y : Fin n → AE D) : (c • (1 : Matrix (Fin n) (Fin n) (AE D))) *ᵥ y = c • y

theorem **glue_finset** {V A : HSp D} (e : GA D A →* GA D V) (he : ∀ v : GlobalTheory.FinPl.Pl D, ∀ a ∈ grpAt A (· = v), e a ∈ grpAt V (· = v)) (a : GlobalTheory.FinPl.Pl D → GA D A) (c : GlobalTheory.FinPl.Pl D → AE D) (z : GlobalTheory.FinPl.Pl D → GA D V) (ha : ∀ v, CenAt A v (a v)) (hz : ∀ v, CenAt V v (z v)) (hzm : ∀ v, Formalism.L2.Theta.mt V (z v) = c v • (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D))) (s : Finset (GlobalTheory.FinPl.Pl D)) : ∃ α ∈ GfC A, α ∈ Subgroup.center (GA D A) ∧ ∃ ζ ∈ Subgroup.center (GA D V), (∀ v ∈ s, ∀ y : Fin V.rank → AE D, (∀ i, y i ∈ idAt D (· = v)) → Formalism.L2.Theta.mt V (e α) *ᵥ y = Formalism.L2.Theta.mt V (e (a v)) *ᵥ y ∧ Formalism.L2.Theta.mt V ζ *ᵥ y = c v • y) ∧ (∀ v ∉ s, ∀ y : Fin V.rank → AE D, (∀ i, y i ∈ idAt D (· = v)) → Formalism.L2.Theta.mt V (e α) *ᵥ y = y ∧ Formalism.L2.Theta.mt V ζ *ᵥ y = y)

theorem **glue** : ∀ (D : D7) (V A : HSp D) (e : GA D A →* GA D V) (g : GA D V) (L M : Lat V), IsSelfDualC V L → IsSelfDualC V M → (∀ v : GlobalTheory.FinPl.Pl D, ∀ a ∈ grpAt A (· = v), e a ∈ grpAt V (· = v)) → (∀ v : GlobalTheory.FinPl.Pl D, ∃ (a : GA D A) (c : AE D), CenAt A v a ∧ NormOneAt v c ∧ SMoves V c (g * e a) L M v) → ∃ a ∈ GfC A, a ∈ Subgroup.center (GA D A) ∧ ∃ z ∈ Subgroup.center (GA D V), Moves V (z * (g * e a)) L M (fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralGlueLocalHomothetiesStatement

theorem **_root_.GlobalTheory.T.exists_central_glue_localHomotheties** : ∀ (D : D7) (V A : HSp D) (e : GA D A →* GA D V) (g : GA D V) (L M : Lat V), IsSelfDualC V L → IsSelfDualC V M → (∀ v : GlobalTheory.FinPl.Pl D, ∀ a ∈ grpAt A (· = v), e a ∈ grpAt V (· = v)) → (∀ v : GlobalTheory.FinPl.Pl D, ∃ (a : GA D A) (c : AE D), CenAt A v a ∧ NormOneAt v c ∧ SMoves V c (g * e a) L M v) → ∃ a ∈ GfC A, a ∈ Subgroup.center (GA D A) ∧ ∃ z ∈ Subgroup.center (GA D V), Moves V (z * (g * e a)) L M (fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralRealPointUnitScalarSinglePlace

theorem **exists_unit** {θ : InfFam D} (hθ : IsInfFam θ) (τ : D.what₂.Tau) {ζ : ℂ} (hζ : ‖ζ‖ = 1) : ∃ a : AE D, finPart D a = 1 ∧ (∀ ρ, θ ρ (infPart D a) = if ρ = τ then ζ else 1) ∧ cA D a * a = 1

theorem **scalar_mem** (S : HSp D) {a : AE D} (ha : cA D a * a = 1) : Formalism.UG.cT (cA D) (Matrix.diagonal fun _ : Fin S.rank => a) * JR (AF D.host.F) S.gram * (Matrix.diagonal fun _ : Fin S.rank => a) = JR (AF D.host.F) S.gram

theorem **mt_scalarElt** (S : HSp D) {a : AE D} (ha : cA D a * a = 1) : Formalism.L2.Theta.mt S (scalarElt S ha) = Matrix.diagonal fun _ => a

theorem **scalarElt_mem_GinfC** (S : HSp D) {a : AE D} (ha : cA D a * a = 1) (h1 : finPart D a = 1) : scalarElt S ha ∈ GinfC S

theorem **scalarElt_mem_center** (S : HSp D) {a : AE D} (ha : cA D a * a = 1) : scalarElt S ha ∈ Subgroup.center (GA D S)

theorem **compAt_scalarElt** (θ : InfFam D) (τ : D.what₂.Tau) (S : HSp D) {a : AE D} (ha : cA D a * a = 1) : compAt θ τ (Formalism.L2.Theta.mt S (scalarElt S ha)) = θ τ (infPart D a) • (1 : Matrix (Fin S.rank) (Fin S.rank) ℂ)

theorem **exists_central_of_norm_one** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (S : HSp D) (τ : D.what₂.Tau) (ζ : ℂ), ‖ζ‖ = 1 → ∃ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) ∧ compAt θ τ (Formalism.L2.Theta.mt S z) = ζ • (1 : Matrix (Fin S.rank) (Fin S.rank) ℂ) ∧ ∀ ρ : D.what₂.Tau, ρ ≠ τ → compAt θ ρ (Formalism.L2.Theta.mt S z) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralRealPointUnitScalarSinglePlaceStatement

theorem **_root_.GlobalTheory.T.exists_central_realPoint_unit_scalar_single_place** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (S : HSp D) (τ : D.what₂.Tau) (ζ : ℂ), ‖ζ‖ = 1 → ∃ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) ∧ compAt θ τ (Formalism.L2.Theta.mt S z) = ζ • (1 : Matrix (Fin S.rank) (Fin S.rank) ℂ) ∧ ∀ ρ : D.what₂.Tau, ρ ≠ τ → compAt θ ρ (Formalism.L2.Theta.mt S z) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralUnitaryScalarOfNormOneIdele

theorem **scalGL_mem** (σ : A →+* A) (J : Matrix n n A) (c : A) (h : c * σ c = 1) : scalGL (n

theorem **mt_scalGA** (S : HSp D) (c : AE D) (hc : c * cA D c = 1) : Formalism.L2.Theta.mt S (scalGA S c hc) = c • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D))

theorem **scalGA_mem_center** (S : HSp D) (c : AE D) (hc : c * cA D c = 1) : scalGA S c hc ∈ Subgroup.center (GA D S)

theorem **scalGA_mem_grpAt** (S : HSp D) (v : GlobalTheory.FinPl.Pl D) (c : AE D) (hc : NormOneAt v c) : scalGA S c hc.1 ∈ grpAt S (· = v)

theorem **exists_cenAt_scalar** : ∀ (D : D7) (S : HSp D) (v : GlobalTheory.FinPl.Pl D) (c : AE D), NormOneAt v c → ∃ z : GA D S, CenAt S v z ∧ Formalism.L2.Theta.mt S z = c • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralUnitaryScalarOfNormOneIdeleStatement

theorem **_root_.GlobalTheory.T.exists_centralUnitary_scalar_of_normOneIdele** : ∀ (D : D7) (S : HSp D) (v : GlobalTheory.FinPl.Pl D) (c : AE D), NormOneAt v c → ∃ z : GA D S, CenAt S v z ∧ Formalism.L2.Theta.mt S z = c • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralValueNeOneOfGramNe

theorem **gramSkew_rat** {D : D7} (W : HSp D) {ρ : ℕ} (y : Matrix (Fin W.rank) (Fin ρ) D.host.E) : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew W (rat y) = rat ((-D.host.xi0) • (gramQ W y)ᵀ)

theorem **gramQ_eq_of_gramSkew** {D : D7} (W : HSp D) {ρ : ℕ} {x y : Matrix (Fin W.rank) (Fin ρ) D.host.E} (h : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew W (rat x) = GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew W (rat y)) : gramQ W x = gramQ W y

theorem **exists_central_ne** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (x₀ x : Matrix (Fin W.rank) (Fin ρ) D.host.E), gramQ W x ≠ gramQ W x₀ → ∃ c : skewA D ρ, star (ψ (d.qS W c (rat x₀))) * ψ (d.qS W c (rat x)) ≠ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCentralValueNeOneOfGramNeStatement

theorem **_root_.GlobalTheory.T.exists_centralValue_ne_one_of_gram_ne** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (x₀ x : Matrix (Fin W.rank) (Fin ρ) D.host.E), gramQ W x ≠ gramQ W x₀ → ∃ c : skewA D ρ, star (ψ (d.qS W c (rat x₀))) * ψ (d.qS W c (rat x)) ≠ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCharExponentInfinityKernelShift

theorem **int_eq_sub_half** (R : Ranks) (j : ℕ) (s z : ℤ) (h : (s : ℚ) = z - (R.N j : ℚ) / 2) : s = z - (R.N j : ℤ) / 2

theorem **chr_exp_shifts** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m), ∃ e : (I.rec7 l Z).what₂.Tau → ℤ, (∀ τ, HasInfExp (I.rec7 l Z) (A.chr (i + 1)) τ (e τ)) ∧ ∀ τ, (I.chain l u a).sSharp i τ = -e τ - ((I.rec7 l Z).sp.X (i + 1)).rank / 2 ∧ -(I.chain l u a).sDiv i τ = e τ + ((I.rec7 l Z).sp.X i).rank / 2 ∧ 0 ≤ shiftOf (Formalism.L2.Forms.lamT (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCharExponentInfinityKernelShiftStatement

theorem **_root_.GlobalTheory.T.exists_charExponent_infinity_kernel_shift** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m), ∃ e : (I.rec7 l Z).what₂.Tau → ℤ, (∀ τ, HasInfExp (I.rec7 l Z) (A.chr (i + 1)) τ (e τ)) ∧ ∀ τ, (I.chain l u a).sSharp i τ = -e τ - ((I.rec7 l Z).sp.X (i + 1)).rank / 2 ∧ -(I.chain l u a).sDiv i τ = e τ + ((I.rec7 l Z).sp.X i).rank / 2 ∧ 0 ≤ shiftOf (Formalism.L2.Forms.lamT (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsChartPointRightTranslateStabilizerSplitPla

theorem **decoratedLattice_unitScale_functional_keeps_frames** {X : Formalism.L2.Col.Fields} {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 X.E))} {y} (p : PtY (X

theorem **finM_mul'** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (g h : GA D V) : finM V hn (g * h) = finM V hn g * finM V hn h

theorem **ptOf_gl_dia** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ g : 𝓀[k]ˣ → GL (Fin ((I.rec7 l Z).sp.X j).rank) k, (∀ s, g s • Q j i = s • Q j i) → ∀ b (c : {c // A.chOf j c = b}), ∃ e : 𝓀[k]ˣ ≃ ((I.atT l u a).Fv v)ˣ, ∀ s, ∃ z : TC.GA B (B.R.N j), z.1 = finM _ (rankN (I.rec7 l Z) j hj) (A.h j b c * (β j i).gl (g s)) ∧ TC.PtOf (z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsChartPointRightTranslateStabilizerSplitPlaStatement

theorem **_root_.GlobalTheory.T.exists_chartPoint_rightTranslate_stabilizer_splitPlace** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ g : 𝓀[k]ˣ → GL (Fin ((I.rec7 l Z).sp.X j).rank) k, (∀ s, g s • Q j i = s • Q j i) → ∀ b (c : {c // A.chOf j c = b}), ∃ e : 𝓀[k]ˣ ≃ ((I.atT l u a).Fv v)ˣ, ∀ s, ∃ z : TC.GA B (B.R.N j), z.1 = finM _ (ran

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsChartsAtTwistPlacesChain

theorem **isIdemOf_idem** {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) : IsIdemOf φ (GlobalTheory.EmbChart.idem (GlobalTheory.SplitProjChart.emb φ))

theorem **exists_unif** (v : HeightOneSpectrum (𝒪 D.host.F)) : ∃ ϖ : (Fv D.host.F v)ˣ, ValuativeRel.valuation (Fv D.host.F v) (ϖ : Fv D.host.F v) = ValuativeRel.uniformizer (Fv D.host.F v)

theorem **unit_section** {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) {e : AE D} (he : IsIdemOf φ e) : ∃ U : (Fv D.host.F v)ˣ →* (AE D)ˣ, (∀ y, (1 - e) * (U y : AE D) = 1 - e ∧ rdA φ (U y : AE D) = y) ∧ ∀ (x : AE D) (y : (Fv D.host.F v)ˣ), (1 - e) * x = 1 - e → rdA φ x = y → x = U y

theorem **blockEquiv_inl** (n r k t : ℕ) (h : n = r + k * t) (i : Fin r) : ((blockEquiv n r k t h (.inl i) : Fin n) : ℕ) = i

theorem **exists_isChart** (s : PiLocalization₂ D) (k : ℕ) (hrank : s.V.rank = D.R.r + k * D.R.t) {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) {e : AE D} (he : IsIdemOf φ e) : ∃ ι, IsChart s k φ ι

theorem **charts_at_twist** : ∀ D : D7, ForStageQ D fun c w j s => ∀ PP ∈ w.Q, (∃ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP) ∧ ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → (∃ e, IsIdemOf φ e) ∧ (∃ x ϖ, IsLocUnif φ x ϖ) ∧ (∃ ι, IsChart s j φ ι) ∧ ∀ χ : (AE D)ˣ →* Circle, ∃ μ : (Fv D.host.F v)ˣ →* ℂˣ, GlobalTheory.SmGL.IsUnitaryChar μ ∧ (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μ y = Circle.toUnits (χ x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsChartsAtTwistPlacesChainStatement

theorem **_root_.GlobalTheory.T.exists_charts_at_twist_places_chain** : ∀ D : D7, ForStageQ D fun c w j s => ∀ PP ∈ w.Q, (∃ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP) ∧ ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → (∃ e, IsIdemOf φ e) ∧ (∃ x ϖ, IsLocUnif φ x ϖ) ∧ (∃ ι, IsChart s j φ ι) ∧ ∀ χ : (AE D)ˣ →* Circle, ∃ μ : (Fv D.host.F v)ˣ →* ℂˣ, GlobalTheory.SmGL.IsUnitaryChar μ ∧ (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μ y = Circle.toUnits (χ x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsClopenCompactNbhdZero

theorem **isOpen_intAf** : IsOpen (intAf F)

theorem **isCompact_intAf** : IsCompact (intAf F)

theorem **isClosed_intAf** : IsClosed (intAf F)

theorem **zero_mem_intAf** : (0 : FiniteAdeleRing (𝒪 F) F) ∈ intAf F

theorem **continuous_coordEquiv** : Continuous (coordEquiv F E)

theorem **continuous_coordEquiv_symm** : Continuous (coordEquiv F E).symm

theorem **isClopen_intTens** : IsClopen (intTens F E)

theorem **isCompact_intTens** : IsCompact (intTens F E)

theorem **zero_mem_intTens** : (0 : A ⊗[F] E) ∈ intTens F E

theorem **exists_clopen_compact_nbhd_zero** : ∀ (D : D7) (m n : ℕ), ∃ U : Set (Fin m → Fin n → Af D.host.F ⊗[D.host.F] D.host.E), IsClopen U ∧ IsCompact U ∧ (0 : Fin m → Fin n → Af D.host.F ⊗[D.host.F] D.host.E) ∈ U

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoefficientVectorReadingRealRepresentation

theorem **exists_pairB_of_range_eq** {ι : Type} [Fintype ι] {n : ℕ} (G G' : ι → Matrix (Fin n) (Fin n) ℂ) (lam : ι → DomWeight n) (o : Frames G') (ho : (fun τ => (o.1 τ)⁻¹) = 1) {V : Type} [AddCommGroup V] [Module ℂ V] (j : V →ₗ[ℂ] (Frames G → ℂ)) (hinj : Function.Injective j) (hr : LinearMap.range j = ValC G lam) (ℓ : V →ₗ[ℂ] ℂ) : ∃ w ∈ Formalism.L2.DIV.VV ℂ lam, ∀ v, ℓ v = Formalism.L2.DIV.pairB lam (Formalism.L2.Col.polOf G' (shiftOf lam) o (j v)) w

theorem **sgT_eq_perIn_sgT** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) : Formalism.L2.Forms.sgT (D

theorem **exists_pairB_of_isJ** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) {S : (I.rec7 l Z).what₂.Tau → SPar} (MX : DSModel (I.rec7 l Z).sp.X' S) (j : SmT MX →ₗ[ℂ] (Formalism.L2.Forms.FrP (I.rec7 l Z) → ℂ)), GlobalTheory.Core.Rk.Top.IsJ u MX j → ∀ ℓ : SmT MX →ₗ[ℂ] ℂ, ∃ w ∈ Formalism.L2.DIV.VV ℂ (I.perIn u a).sgT, ∀ v, ℓ v = Formalism.L2.DIV.pairB (I.perIn u a).sgT (Formalism.L2.Col.polOf (GlobalTheory.Mem.Col2.Gr B.fields.emb B.embLL (B.R.N B.R.m − 1)) (Formalism.L2.Forms.shiftOf (I.perIn u a).sgT) (GlobalTheory.Mem.Col2.fr0 B.fields.emb B.embLL (B.R.N B.R.m − 1)) (j v)) w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoefficientVectorReadingRealRepresentationStatement

theorem **_root_.GlobalTheory.T.exists_coefficientVector_reading_realRepresentation_form** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) {S : (I.rec7 l Z).what₂.Tau → SPar} (MX : DSModel (I.rec7 l Z).sp.X' S) (j : SmT MX →ₗ[ℂ] (Formalism.L2.Forms.FrP (I.rec7 l Z) → ℂ)), GlobalTheory.Core.Rk.Top.IsJ u MX j → ∀ ℓ : SmT MX →ₗ[ℂ] ℂ, ∃ w ∈ Formalism.L2.DIV.VV ℂ (I.perIn u a).sgT, ∀ v, ℓ v = Formalism.L2.DIV.pairB (I.perIn u a).sgT (Formalism.L2.Col.polOf (GlobalTheory.Mem.Col2.Gr B.fields.emb B.embLL (B.R.N B.R.m − 1)) (Formalism.L2.Forms.shiftOf (I.perIn u a).sgT) (GlobalTheory.Mem.Col2.fr0 B.fields.emb B.embLL (B.R.N B.R.m − 1)) (j v)) w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsColSpaceAutToLocalGLMemOpenDense

theorem **GlobalTheory.denseGL_completion** : ∀ {E k : Type} [Field E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : E →+* k), DenseRange ι → ∀ {A : Type} [Fintype A] [DecidableEq A] {M : ℕ} (e : A ≃ Fin M) (y : GL A k) (V : Set (GL A k)), IsOpen V → y ∈ V → ∃ x : Sp E M ≃ₗ[E] Sp E M, locHom ι e x ∈ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsColSpaceAutToLocalGLMemOpenDenseStatement

theorem **_root_.GlobalTheory.T.exists_colSpaceAutToLocalGL_mem_open_dense** : ∀ {E k : Type} [Field E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : E →+* k), DenseRange ι → ∀ {A : Type} [Fintype A] [DecidableEq A] {M : ℕ} (e : A ≃ Fin M) (y : GL A k) (V : Set (GL A k)), IsOpen V → y ∈ V → ∃ x : Sp E M ≃ₗ[E] Sp E M, locHom ι e x ∈ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCombinationRebaseVacuumLatticesInjective

theorem **zero_placeModel** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (e : AE D) (Λ : AddSubgroup (WO S T)) : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e Λ 1 0 0

theorem **local_step** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) {S T : HSp D} {M N : GlobalTheory.FinPl.Lat S} {M' N' : GlobalTheory.FinPl.Lat T} (hM : IsSelfDualC S M) (hN : IsSelfDualC S N) (hM' : IsSelfDualC T M') (hN' : IsSelfDualC T N') (p : GlobalTheory.FinPl.Pl D) (hsp : ∃ e ε, IsSplitIdem D e ∧ IsPlaceIdem D p ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε) (hu1 : ∀ a ∈ GlobalTheory.FinPl.intAt D (· = p), incA D D.host.xi0 ∗ a ∈ GlobalTheory.FinPl.intAt D (· = p) ∧ incA D D.host.xi0⁻¹ ∗ a ∈ GlobalTheory.FinPl.intAt D (· = p)) (hu2 : ∀ (V : HSp D) (Λ : GlobalTheory.FinPl.Lat V), GlobalTheory.FinPl.latA V Λ (· = p) = GlobalTheory.FinPl.latV V Λ (· = p)) : ∃ c : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt (· = p) c ∧ ∀ f, GlobalTheory.FinPl.Vac ψ S T M M' (· = p) f → GlobalTheory.FinPl.Vac ψ S T N N' (· = p) (comb ψ S T c f) ∧ (f ≠ 0 → comb ψ S T c f ≠ 0)

theorem **idAt_mono'** {D : D7} {P P' : GlobalTheory.FinPl.Pl D → Prop} (h : ∀ v, P v → P' v) : GlobalTheory.FinPl.idAt D P ≤ GlobalTheory.FinPl.idAt D P'

theorem **foldr_vac_off** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) {S T : HSp D} (c : GlobalTheory.FinPl.Pl D → WO S T →₀ ℂ) (X : GlobalTheory.FinPl.Pl D → Prop) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T) : ∀ l : List (GlobalTheory.FinPl.Pl D), (∀ p ∈ l, GlobalTheory.FinPl.CombAt (· = p) (c p)) → (∀ p ∈ l, ¬ X p) → ∀ f, GlobalTheory.FinPl.Vac ψ S T LS LT X f → GlobalTheory.FinPl.Vac ψ S T LS LT X (l.foldr (fun p g => comb ψ S T (c p) g) f)

theorem **foldr_vac_on** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) {S T : HSp D} {M N : GlobalTheory.FinPl.Lat S} {M' N' : GlobalTheory.FinPl.Lat T} (hM : IsSelfDualC S M) (hN : IsSelfDualC S N) (hM' : IsSelfDualC T M') (hN' : IsSelfDualC T N') (c : GlobalTheory.FinPl.Pl D → WO S T →₀ ℂ) : ∀ l : List (GlobalTheory.FinPl.Pl D), l.Nodup → (∀ p ∈ l, GlobalTheory.FinPl.CombAt (· = p) (c p) ∧ ∀ f, GlobalTheory.FinPl.Vac ψ S T M M' (· = p) f → GlobalTheory.FinPl.Vac ψ S T N N' (· = p) (comb ψ S T (c p) f) ∧ (f ≠ 0 → comb ψ S T (c p) f ≠ 0)) → ∀ f, GlobalTheory.FinPl.Vac ψ S T M M' (· ∈ l) f → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ l) (l.foldr (fun p g => comb ψ S T (c p) g) f) ∧ (f ≠ 0 → l.foldr (fun p g => comb ψ S T (c p) g) f ≠ 0)

theorem **GlobalTheory.VacRebase.rebase_vac** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (S T : HSp D) (M N : GlobalTheory.FinPl.Lat S) (M' N' : GlobalTheory.FinPl.Lat T) (Q : Finset (GlobalTheory.FinPl.Pl D)), IsSelfDualC S M → IsSelfDualC S N → IsSelfDualC T M' → IsSelfDualC T N' → (∀ p ∈ Q, ((∃ e ε, IsSplitIdem D e ∧ IsPlaceIdem D p ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε) ∧ (∀ a ∈ GlobalTheory.FinPl.intAt D (· = p), incA D D.host.xi0 ∗ a ∈ GlobalTheory.FinPl.intAt D (· = p) ∧ incA D D.host.xi0⁻¹ ∗ a ∈ GlobalTheory.FinPl.intAt D (· = p)) ∧ ∀ (V : HSp D) (Λ : GlobalTheory.FinPl.Lat V), GlobalTheory.FinPl.latA V Λ (· = p) = GlobalTheory.FinPl.latV V Λ (· = p))) → (∀ p, p ∉ Q → GlobalTheory.FinPl.latV S M (· = p) = GlobalTheory.FinPl.latV S N (· = p) ∧ GlobalTheory.FinPl.latV T M' (· = p) = GlobalTheory.FinPl.latV T N' (· = p)) → ∃ o : WO S T →₀ ℂ, GlobalTh

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCombinationRebaseVacuumLatticesInjectiveStatement

theorem **_root_.GlobalTheory.T.exists_combination_rebase_vacuumLattices_injective** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (S T : HSp D) (M N : GlobalTheory.FinPl.Lat S) (M' N' : GlobalTheory.FinPl.Lat T) (Q : Finset (GlobalTheory.FinPl.Pl D)), IsSelfDualC S M → IsSelfDualC S N → IsSelfDualC T M' → IsSelfDualC T N' → (∀ p ∈ Q, ((∃ e ε, IsSplitIdem D e ∧ IsPlaceIdem D p ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε) ∧ (∀ a ∈ GlobalTheory.FinPl.intAt D (· = p), incA D D.host.xi0 ∗ a ∈ GlobalTheory.FinPl.intAt D (· = p) ∧ incA D D.host.xi0⁻¹ ∗ a ∈ GlobalTheory.FinPl.intAt D (· = p)) ∧ ∀ (V : HSp D) (Λ : GlobalTheory.FinPl.Lat V), GlobalTheory.FinPl.latA V Λ (· = p) = GlobalTheory.FinPl.latV V Λ (· = p))) → (∀ p, p ∉ Q →

GlobalTheory.FinPl.latV S M (· = p) = GlobalTheory.FinPl.latV S N (· = p) ∧ GlobalTheory.FinPl.latV T M' (· = p) = GlobalTheory.FinPl.latV T N'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCommonHeisenbergLevelWeilTransformCompact

theorem **mt_mul'** (V : HSp D) (g h : GA D V) : Formalism.L2.Theta.mt V (g * h) = Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V h

theorem **mt_one'** (V : HSp D) : Formalism.L2.Theta.mt V 1 = 1

theorem **actO_actO'** (S T : HSp D) (g g' : GA D S) (h h' : GA D T) (x : WO S T) : actO S T g h (actO S T g' h' x) = actO S T (g * g') (h * h') x

theorem **actO_inv_actO'** (S T : HSp D) (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g h (actO S T g⁻¹ h⁻¹ x) = x

theorem **actO_add'** (S T : HSp D) (g : GA D S) (h : GA D T) (x y : WO S T) : actO S T g h (x + y) = actO S T g h x + actO S T g h y

theorem **actO_map_infPart** (S T : HSp D) (g : GA D S) (h : GA D T) {x : WO S T} (hx : x.map (infPart D) = 0) : (actO S T g h x).map (infPart D) = 0

theorem **continuous_actO_inv** (S T : HSp D) : Continuous fun q : (GA D S × GA D T) × WO S T => actO S T q.1.1⁻¹ q.1.2⁻¹ q.2

theorem **exists_level_compact** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ f, IsThetaFunO ψ S T f → ∀ K : Set (GA D S × GA D T), IsCompact K → ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ p ∈ K, om p f ∈ GlobalTheory.Size.thetaLevel ψ S T U

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCommonHeisenbergLevelWeilTransformCompactStatement

theorem **_root_.GlobalTheory.T.exists_common_heisenbergLevel_weilTransform_compact** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ f, IsThetaFunO ψ S T f → ∀ K : Set (GA D S × GA D T), IsCompact K → ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ p ∈ K, om p f ∈ GlobalTheory.Size.thetaLevel ψ S T U

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCommonLevelCompactTranslatesThetaFunction

theorem **mt_mul'** {D : D7} (V : HSp D) (g h : GA D V) : Formalism.L2.Theta.mt V (g * h) = Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V h

theorem **mt_one'** {D : D7} (V : HSp D) : Formalism.L2.Theta.mt V 1 = 1

theorem **continuous_mt'** {D : D7} (V : HSp D) : Continuous (Formalism.L2.Theta.mt V)

theorem **actO_mul_mt** {D : D7} (S T : HSp D) (k : GA D S) (u : WO S T) : actO S T k 1 (HMul.hMul (u : Matrix (Fin T.rank) (Fin S.rank) (AE D)) (Formalism.L2.Theta.mt S k)) = u

theorem **exists_common_level** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : WRep S T), IsWeilDatumO ψ S T χS χT om → ∀ K : Set (GA D S), IsCompact K → ∀ f, IsThetaFunO ψ S T f → ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ k ∈ K, ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u (om (k, 1) f) = om (k, 1) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCommonLevelCompactTranslatesThetaFunctionStatement

theorem **_root_.GlobalTheory.T.exists_common_level_compact_translates_thetaFunction** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : WRep S T), IsWeilDatumO ψ S T χS χT om → ∀ K : Set (GA D S), IsCompact K → ∀ f, IsThetaFunO ψ S T f → ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ k ∈ K, ∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u (om (k, 1) f) = om (k, 1) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCommonLevelSobolevBoundWeilCompact

theorem **mt_mul_eq** (V : HSp D) (g h : GA D V) : Formalism.L2.Theta.mt V (g * h) = Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V h

theorem **mt_one_eq** (V : HSp D) : Formalism.L2.Theta.mt V 1 = 1

theorem **continuous_mt_eq** (V : HSp D) : Continuous (Formalism.L2.Theta.mt V)

theorem **actO_actO_inv** {S T : HSp D} (g : GA D S) (h : GA D T) (x : WO S T) : Formalism.IndefR.Datum.actO S T g h (Formalism.IndefR.Datum.actO S T g⁻¹ h⁻¹ x) = x

theorem **dRho_om** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z (om (g, h) f) = om (g, h) (Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ (compAt θ τ (Formalism.L2.Theta.mt T h⁻¹) * z * compAt θ τ (Formalism.L2.Theta.mt S g)) f)

theorem **continuous_dirSize** {θ : InfFam D} (hθ : IsInfFam θ) (S T : HSp D) : Continuous (dirSize θ S T)

theorem **actO_add'** {S T : HSp D} (g : GA D S) (h : GA D T) (x y : WO S T) : Formalism.IndefR.Datum.actO S T g h (x + y) = Formalism.IndefR.Datum.actO S T g h x + Formalism.IndefR.Datum.actO S T g h y

theorem **actO_zero'** {S T : HSp D} (g : GA D S) (h : GA D T) : Formalism.IndefR.Datum.actO S T g h 0 = 0

theorem **actO_neg'** {S T : HSp D} (g : GA D S) (h : GA D T) (x : WO S T) : Formalism.IndefR.Datum.actO S T g h (-x) = - Formalism.IndefR.Datum.actO S T g h x

theorem **word_om_le** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) {μ : Measure (WO S T)} {ℱ : Set (WO S T)} (hμ : IsCovolOne (ratL T S) μ ℱ) {K : Set (GA D S × GA D T)} (B : ℝ≥0) (hB : ∀ x ∈ K, dirSize θ S T x ≤ B) (k : ℕ) : ∃ C : ℝ≥0, ∀ l : List (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool), l.length = k → ∀ x ∈ K, ∀ g, IsThetaFunO ψ S T g → l2O μ ℱ (wordAct (heisDer ψ θ) l (om x g)) ≤ C * sobO ψ θ μ ℱ k g

theorem **GlobalTheory.ThetaDens.om_compact_sobCont** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsCovolOne (ratL T S) μ ℱ → ∀ K : Set (GA D S × GA D T), IsCompact K → ∀ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) → ∃ U' : AddSubgroup (WO S T), IsOpen (U' : Set (WO S T)) ∧ (∀ f ∈ thetaLevel ψ S T U, ∀ x ∈ K, om x f ∈ thetaLevel ψ S T U') ∧ ∀ d : ℕ, ∃ (d' : ℕ) (C : ℝ≥0), ∀ f ∈ thetaLevel ψ S T U, ∀ x ∈ K, sobO ψ θ μ ℱ d (om x f) ≤ C * sobO ψ θ μ ℱ d' f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCommonLevelSobolevBoundWeilCompactStatement

theorem **_root_.GlobalTheory.T.exists_common_level_sobolev_bound_weil_compact** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ θ : InfFam D, IsInfFam

θ → ∀ (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsCovolOne (ratL T S) μ ℱ → ∀ K : Set (GA D S × GA D T), IsCompact K → ∀ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) → ∃ U' : AddSubgroup (WO S T), IsOpen (U' : Set (WO S T)) ∧ (∀ f ∈ thetaLevel ψ S T U, ∀ x ∈ K, om x f ∈ thetaLevel ψ S T U') ∧ ∀ d : ℕ, ∃ (d' : ℕ) (C : ℝ≥0), ∀ f ∈ thetaLevel ψ S T U, ∀ x ∈ K, sobO ψ θ μ ℱ d (om x f) ≤ C * sobO ψ θ μ ℱ d' f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCompactIntegralUnitaryFinitePart

theorem **sub_of_red** {E : Type} [Field E] [NumberField E] {n : ℕ} (M : ℕ) {A B : Matrix (Fin n) (Fin n) E} (hA : ∀ i j, IsIntegral ℤ (A i j)) (hB : ∀ i j, IsIntegral ℤ (B i j)) (h : red M A = red M B) : ∀ i j, ∃ b : E, IsIntegral ℤ b ∧ (A − B) i j = (M : E) * b

theorem **mul_cong** {E : Type} [Field E] [NumberField E] {n : ℕ} (M : ℕ) {C D : Matrix (Fin n) (Fin n) E} (hC : ∀ i j, IsIntegral ℤ (C i j)) (hD : ∀ i j, ∃ b : E, IsIntegral ℤ b ∧ D i j = (M : E) * b) : ∀ i j, ∃ a : E, IsIntegral ℤ a ∧ (C * D) i j = (M : E) * a

theorem **kset_nhds** (I : AutInst) : I.Kset ∈ 𝒩 (1 : I.Gf)

theorem **fp_mul** (I : AutInst) {γ δ : GL (Fin I.n) I.E} (hγ : γ ∈ Formalism.UG.UJ (starRingEnd I.E) I.J) (hδ : δ ∈ Formalism.UG.UJ (starRingEnd I.E) I.J) : fp I (γ⁻¹ * δ) (mul_mem (inv_mem hγ) hδ) = (fp I γ hγ)⁻¹ * fp I δ hδ

theorem **eq_of_real** (I : AutInst) (v : InfinitePlace I.F) {A B : Matrix (Fin I.n) (Fin I.n) I.E} (h : (I.P v)⁻¹ * A.map (I.φ v) * I.P v = (I.P v)⁻¹ * B.map (I.φ v) * I.P v) : A = B

theorem **intUnit_compact** : ∀ I : AutInst, ∃ K : Set I.Gf, IsCompact K ∧ ∀ γ : GL (Fin I.n) I.E, γ ∈ Formalism.UG.UJ (starRingEnd I.E) I.J → (∀ i j, IsIntegral ℤ ((γ : Matrix (Fin I.n) (Fin I.n) I.E) i j)) → (∀ i j, IsIntegral ℤ (((γ⁻¹ : GL (Fin I.n) I.E) : Matrix (Fin I.n) (Fin I.n) I.E) i j)) → ∃ x ∈ I.GFset, x.2 ∈ K ∧ ∀ v, uMat (x.1 v) = (I.P v)⁻¹ * (γ : Matrix (Fin I.n) (Fin I.n) I.E).map (I.φ v) * I.P v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCompactIntegralUnitaryFinitePartStatement

theorem **_root_.GlobalTheory.T.exists_compact_integral_unitary_finitePart** : ∀ I : AutInst, ∃ K : Set I.Gf, IsCompact K ∧ ∀ γ : GL (Fin I.n) I.E, γ ∈ Formalism.UG.UJ (starRingEnd I.E) I.J → (∀ i j, IsIntegral ℤ ((γ : Matrix (Fin I.n) (Fin I.n) I.E) i j)) → (∀ i j, IsIntegral ℤ (((γ⁻¹ : GL (Fin I.n) I.E) : Matrix (Fin I.n) (Fin I.n) I.E) i j)) → ∃ x ∈ I.GFset, x.2 ∈ K ∧ ∀ v, uMat (x.1 v) = (I.P v)⁻¹ * (γ : Matrix (Fin I.n) (Fin I.n) I.E).map (I.φ v) * I.P v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCompactOpenLatticeStabilizerAtPlaces

theorem **evalV_indP** (P : Pl D → Prop) (w : Pl D) : evalV D.host.F w (indP D P) = if P w then 1 else 0

theorem **indicatorAdele_integral** (P : Pl D → Prop) : indP D P ∈ intAf D.host.F

theorem **one_sub_indP_int** (P : Pl D → Prop) : 1 − indP D P ∈ intAf D.host.F

theorem **compFin_e** (P : Pl D → Prop) (w : Pl D) : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w (indP D P ⊗ₜ[D.host.F] (1 : D.host.E)) = if P w then 1 else 0

theorem **e_mul_of_supp** (P : Pl D → Prop) (X : Af D.host.F ⊗[D.host.F] D.host.E) (h : ∀ w, ¬ P w → GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w X = 0) : (indP D P ⊗ₜ[D.host.F] (1 : D.host.E)) * X = X

theorem **e_mul_fin** {P : Pl D → Prop} {a : AE D} (ha : a ∈ idAt D P) : (indP D P ⊗ₜ[D.host.F] (1 : D.host.E)) * finPart D a = finPart D a

theorem **exists_latticeCompletion_cut_to_places_finiteReal** (S : HSp D) (L : Lat S) (P : Pl D → Prop) {z : Fin S.rank → Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ latHat D.host.F D.host.E (L : Set (Fin S.rank → D.host.E))) : ∃ y ∈ latV S L P, (∀ i, infPart D (y i) = 0) ∧ ∀ i, finPart D (y i) = (indP D P ⊗ₜ[D.host.F] (1 : D.host.E)) * z i

theorem **mem_idAt_of_fin** (P : Pl D → Prop) {a : AE D} (h1 : infPart D a = 0) {z : Af D.host.F ⊗[D.host.F] D.host.E} (h2 : finPart D a = (indP D P ⊗ₜ[D.host.F] (1 : D.host.E)) * z) : a ∈ idAt D P

theorem **exists_lift_latV** (S : HSp D) (L : Lat S) (P : Pl D → Prop) {z : Fin S.rank → Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ latHat D.host.F D.host.E (L : Set (Fin S.rank → D.host.E))) : ∃ y ∈ latV S L P, (∀ i, y i ∈ idAt D P) ∧ ∀ i, finPart D (y i) = (indP D P ⊗ₜ[D.host.F] (1 : D.host.E)) * z i

theorem **mem_latV_iff** (S : HSp D) (L : Lat S) (P : Pl D → Prop) (y : Fin S.rank → AE D) (hy : ∀ i, y i ∈ idAt D P) : y ∈ latV S L P ↔ (finPart D ∘ y) ∈ latHat D.host.F D.host.E (L : Set (Fin S.rank → D.host.E))

theorem **supp_mulVec** (S : HSp D) (P : Pl D → Prop) (g : GA D S) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D P) : ∀ i, (Formalism.L2.Theta.mt S g *ᵥ y) i ∈ idAt D P

theorem **fin_mulVec** (S : HSp D) (g : GA D S) (y : Fin S.rank → AE D) : finPart D ∘ (Formalism.L2.Theta.mt S g *ᵥ y) = (finMat D.sp.cm.sig S.gram g) *ᵥ (finPart D ∘ y)

theorem **moves_of_levelOpen** (S : HSp D) (L : Lat S) (hL : L.FG) (P : Pl D → Prop) {g : GA D S} (hg : g ∈ GlobalTheory.AdelicU.levelOpen D.sp.cm.sig S.gram L) : Moves S g L L P

theorem **keeps_of_core** {R : Type} [CommRing R] {n : ℕ} (H : AddSubgroup (Fin n → R)) (e : R) (hee : e * e = e) (h1 : ∀ z ∈ H, e • z ∈ H) (h2 : ∀ z ∈ H, (1 − e) • z ∈ H) (M : Matrix (Fin n) (Fin n) R) (hN : ∀ i j, e * (M − 1) i j = (M − 1) i j) (hcore : ∀ z, e • z ∈ H ↔ M *ᵥ (e • z) ∈ H) (y : Fin n → R) : y ∈ H ↔ M *ᵥ y ∈ H

theorem **fin_sub_one** (S : HSp D) (g : GA D S) (i j : Fin S.rank) : finPart D ((Formalism.L2.Theta.mt S g − 1) i j) = (finMat D.sp.cm.sig S.gram g − 1) i j

theorem **levelOpen_of_inK** (S : HSp D) (L : Lat S) (hL : L.FG) (P : Pl D → Prop) {g : GA D S} (hg : InK S L P g) : g ∈ GlobalTheory.AdelicU.levelOpen D.sp.cm.sig S.gram L

theorem **compAt_eq** (v : Pl D) (a : AE D) : compAt D v a = GlobalTheory.AdelicU.compFin D.host.F D.host.E v a

theorem **map_eq_one_iff** {R R' : Type} [CommRing R] [CommRing R'] {n : ℕ} (f : R →+* R') (M : Matrix (Fin n) (Fin n) R) : M.map f = 1 ↔ ∀ i j, f ((M − 1) i j) = 0

theorem **mem_GfC_iff** (S : HSp D) (g : GA D S) : g ∈ GfC S ↔ ∀ i j, infPart D ((Formalism.L2.Theta.mt S g − 1) i j) = 0

theorem **proj_eq_one_iff** (S : HSp D) (v : Pl D) (g : GA D S) : GlobalTheory.AdelicU.proj D.sp.cm.sig S.gram v g = 1 ↔ ∀ i j, compAt D v ((Formalism.L2.Theta.mt S g − 1) i j) = 0

theorem **mem_grpAt_iff** (S : HSp D) (P : Pl D → Prop) (g : GA D S) : g ∈ grpAt S P ↔ g ∈ GfC S ∧ ∀ v, ¬ P v → GlobalTheory.AdelicU.proj D.sp.cm.sig S.gram v g = 1

theorem **isClosed_grpAt** (S : HSp D) (P : Pl D → Prop) : IsClosed (grpAt S P : Set (GA D S))

theorem **exists_compact_open** : ∀ (D : D7) (V : HSp D) (L : Lat V), IsSelfDualC V L → ∀ P : Pl D → Prop, ∃ K : Subgroup (GA D V), (∀ g, g ∈ K ↔ InK V L P g) ∧ IsCompact (K : Set (GA D V)) ∧ IsOpen ((K.subgroupOf (grpAt V P) : Subgroup _) : Set (grpAt V P))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCompactOpenLatticeStabilizerAtPlacesStatement

theorem **_root_.GlobalTheory.T.exists_compactOpen_latticeStabilizerAtPlaces** : ∀ (D : D7) (V : HSp D) (L : Lat V), IsSelfDualC V L → ∀ P : Pl D → Prop, ∃ K : Subgroup (GA D V), (∀ g, g ∈ K ↔ InK V L P g) ∧ IsCompact (K : Set (GA D V)) ∧ IsOpen ((K.subgroupOf (grpAt V P) : Subgroup _) : Set (grpAt V P))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsComplMapEq

theorem **exists_gen** {E : Type} [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (hI : I ≠ 0) : ∃ a₀ ∈ I, a₀ ≠ 0 ∧ ∀ a ∈ I, ∃ c ∈ Ov E w, algebraMap E (w.adicCompletion E) a = algebraMap E (w.adicCompletion E) a₀ * c

theorem **mul_mem_gen** {S : HSp D} {𝔞 : Set D.host.E} {L : Lat S} {P : FinPl D → Prop} {z : AE D} (hz : finPart D z ∈ intAE D) {β : Af D.host.F} (hβi : β ∈ Formalism.MainNotions.StandardLattice.intAf D.host.F) (hβ : ∀ v, ¬ P v → evalV D.host.F v β = 0) {a : D.host.E} (ha : a ∈ 𝔞) {x : Fin S.rank → D.host.E} (hx : x ∈ L) : (fun i => z * (pairA D.host.F 0 β ⊗ₜ[D.host.F] (a * x i))) ∈ latMul S 𝔞 L P

theorem **cw_tmul** (h : w.under (𝒪 D.host.F) = q) (α : Af D.host.F) (x : D.host.E) : cw h (pairA D.host.F 0 α ⊗ₜ[D.host.F] x) = complMap h (evalV D.host.F q α) * algebraMap D.host.E _ x

theorem **cw_mem** (h : w.under (𝒪 D.host.F) = q) {S : HSp D} {𝔞 : Set D.host.E} {L : Lat S} {P : FinPl D → Prop} {a₀ : w.adicCompletion D.host.E} (ha : ∀ a ∈ 𝔞, ∃ c ∈ Ov D.host.E w, algebraMap D.host.E _ a = a₀ * c) {y : Fin S.rank → AE D} (hy : y ∈ latMul S 𝔞 L P) : ∃ z ∈ latAt w L, ∀ i, cw h (y i) = a₀ * z i

theorem **exists_complMap_eq** (h : w.under (𝒪 D.host.F) = q) (hs : Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **singleFin_supp** (α : Fv D.host.F q) : ∀ v, ¬ v = q → evalV D.host.F v (singleFin q α) = 0

theorem **iota_mem** (h : w.under (𝒪 D.host.F) = q) (hs : Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **GlobalTheory.Core.Rk.SWQ.LatHalf.mem_latO_iff_intM** : ∀ (D : D7) (q : FinPl D) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = q), Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCompletionTwistDataChainColumns

theorem **val_eq_one_of_sub_one** {x : k} (hx : valuation k (x - 1) < 1) : valuation k x = 1

theorem **delta_unit** (μ₁ : kˣ →* ℂˣ) (ϖ : kˣ) {x : kˣ} (hx : valuation k (x : k) = 1) : delta μ₁ ϖ x = μ₁ x

theorem **delta_tame** {μ₁ : kˣ →* ℂˣ} (h : IsTame μ₁) (ϖ : kˣ) : IsTame (delta μ₁ ϖ)

theorem **eta_mul** (μ : ℕ → kˣ →* ℂˣ) (m : ℕ) (δ : kˣ →* ℂˣ) {i : ℕ} (hi : i < m) : eta μ m δ (i + 1) * δ = μ (i + 1)

theorem **eta_unram** (μ : ℕ → kˣ →* ℂˣ) (m : ℕ) (δ : kˣ →* ℂˣ) (h : ∀ i, i < m → ∀ x : kˣ, valuation k (x : k) = 1 → μ (i + 1) x = δ x) (j : ℕ) : IsUnram (eta μ m δ j)

theorem **exists_near** {E : Type} [Field E] {ι : E →+* k} (hd : DenseRange ι) (x : k) : ∃ a : E, valuation k (ι a - x) < 1

theorem **tame_near** {μ : kˣ →* ℂˣ} (h : IsTame μ) {x u : kˣ} (hx : valuation k (x : k) = 1) (hu : valuation k ((u : k) - x) < 1) : μ u = μ x

theorem **unitV_inv** {P : HeightOneSpectrum (𝒪 E)} (χ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (a : E) : LocE.unitV χ⁻¹ a = (LocE.unitV χ a)⁻¹

theorem **valuation_self** (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (h0 : 0 < q) : LocE.vP P q (q : E) = 1

theorem **locChar_q** {P : HeightOneSpectrum (𝒪 E)} (q : ℕ) (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (h0 : 0 < q) (cq : K) (χ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) : LocE.locChar q cq χ (q : E) = cq

theorem **muAt_unit** (hP : P.1 ∉ Pr.vGs) (i : ℕ) {a : X.E} (ha : IsUnit (redP P.1 a)) : muAt Ch sel mu cst i P a = LocE.unitV (resChar (mu : Chr X.F X.E X.K) P.1)⁻¹ a

theorem **muAt_q** (hP : P.1 ∉ Pr.vGs) (i : ℕ) (hq : ((qOf P.1 : ℕ) : 𝒪 X.E) ∈ P.1.asIdeal) (h0 : 0 < qOf P.1) : muAt Ch sel mu cst i P ((qOf P.1 : ℕ) : X.E) = cst.muq (i + 1) P

theorem **exists_twData** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (L : ChIdx X Pr R (QE sel (mu : Chr X.F X.E X.K))) (G : ChGlob X R.m) (s : ℕ), L.Ok Ch sel mu cst G s → ∀ (P : rowSG Pr sel mu), P.1 ∉ Pr.vGs → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (ϖ : kˣ), GlobalTheory.SymBridge.IsCompl P.1 (qOf P.1) ι ϖ → GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → ∀ ψ : AddChar k Circle, Continuous ψ → HasConductor ψ 1 → (∀ (a : X.E) (b : ℤ) (n : ℕ), IntAt P.1 (G.xi0 * a - (b : X.E) / (qOf P.1 : X.E) ^ n) → (ψ (ι a) : ℂ) = Complex.exp (2 * Real.pi * Complex.I * b / (qOf P.1 : ℂ) ^ n)) → ∀ μ : ℕ → kˣ →* ℂˣ, (∀ i, i < R.m → (∀ (a : X.E) (u : kˣ), (u : k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCompletionTwistDataChainColumnsStatement

theorem **_root_.GlobalTheory.T.exists_completionTwistData_chainColumns** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (L : ChIdx X Pr R (QE sel (mu : Chr X.F X.E X.K))) (G : ChGlob X R.m) (s : ℕ), L.Ok Ch sel mu cst G s → ∀ (P : rowSG Pr sel mu), P.1 ∉ Pr.vGs → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (ϖ : kˣ), GlobalTheory.SymBridge.IsCompl P.1 (qOf P.1) ι ϖ → GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → ∀ ψ : AddChar k Circle, Continuous ψ → HasConductor ψ 1 → (∀ (a : X.E) (b : ℤ) (n : ℕ), IntAt P.1 (G.xi0 * a - (b : X.E) / (qOf P.1 : X.E) ^ n) → (ψ (ι a) : ℂ) = Complex.exp (2 * Real.pi * Complex.I * b / (qOf P.1 : ℂ) ^ n)) → ∀ μ : ℕ → kˣ →* ℂˣ,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCongruenceModulusIdelicCharacterTrivial

theorem **char_trivial_modSet** : ∀ (D : D7) (ν : (AE D)ˣ →* Circle), Continuous ν → ∃ N : ℕ, 0 < N ∧ ∀ x ∈ GlobalTheory.HeckeSplitAll.modSet D N, ν x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCongruenceModulusIdelicCharacterTrivialStatement

theorem **_root_.GlobalTheory.T.exists_congruenceModulus_idelicCharacter_trivial** : ∀ (D : D7) (ν : (AE D)ˣ →* Circle), Continuous ν → ∃ N : ℕ, 0 < N ∧ ∀ x ∈ GlobalTheory.HeckeSplitAll.modSet D N, ν x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConjStabilizersSelfDualLatticesUnramified

theorem **exists_sq_conj_sub** (h : IsCMExt F E σ) (hU : Algebra.Unramified (𝒪 F) (𝒪 E)) (v : HeightOneSpectrum (𝒪 F)) : ∃ (x : 𝒪 E) (d : 𝒪 F), d ∉ v.asIdeal ∧ algebraMap F E d = ((x : E) - σ x) ^ 2

theorem **exists_isUnit_sub_conj** (h : IsCMExt F E σ) (hU : Algebra.Unramified (𝒪 F) (𝒪 E)) (v : HeightOneSpectrum (𝒪 F)) : ∃ ω : intLoc F E v, IsUnit (ω - conjInt σ v ω)

theorem **exists_norm** (h : IsCMExt F E σ) (hU : Algebra.Unramified (𝒪 F) (𝒪 E)) (v : HeightOneSpectrum (𝒪 F)) {n : ℕ} {B : Matrix (Fin n) (Fin n) (intLoc F E v)} (hB : cT (conjInt σ v) B = B) (hu : IsUnit B) : ∃ u : GL (Fin n) (intLoc F E v), cT (conjInt σ v) u.1 ∗ u.1 = B

theorem **cT_mul** {A : Type} [CommRing A] (s : A →+∗ A) {n : ℕ} (P Q : Matrix (Fin n) (Fin n) A) : cT s (P ∗ Q) = cT s Q ∗ cT s P

theorem **cT_one** {A : Type} [CommRing A] (s : A →+∗ A) {n : ℕ} : cT s (1 : Matrix (Fin n) (Fin n) A) = 1

theorem **exists_orthonormal_frame** (h : IsCMExt F E σ) (hU : Algebra.Unramified (𝒪 F) (𝒪 E)) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] {v : HeightOneSpectrum (𝒪 F)} (hL : IsSelfDualAt σ J L v) : ∃ b : GL (Fin n) (Fv F v ⊗[F] E), cT (sigR _ σ) b.1 ∗ JR (Fv F v) J ∗ b.1 = 1 ∧ ∀ y, (∀ i, y i ∈ intLoc F E v) ↔ b.1 ∗ᵥ y ∈ latLoc F L v

theorem **exists_conj_stabLoc** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → Algebra.Unramified (𝒪 F) (𝒪 E) → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (L L' : Submodule (𝒪 E) (Fin n → E)), Submodule.IsLattice E L → Submodule.IsLattice E L' → ∀ v : HeightOneSpectrum (𝒪 F), IsSelfDualAt σ J L v → IsSelfDualAt σ J L' v → ∃ g : GLoc σ J v, ∀ k : GLoc σ J v, k ∈ stabLoc σ J L' v ↔ g⁻¹ ∗ k ∗ g ∈ stabLoc σ J L v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConjStabilizersSelfDualLatticesUnramifiedStatement

theorem **_root_.GlobalTheory.T.exists_conj_stabilizers_selfDual_lattices_unramified** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → Algebra.Unramified (𝒪 F) (𝒪 E) → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (L L' : Submodule (𝒪 E) (Fin n → E)), Submodule.IsLattice E L → Submodule.IsLattice E L' → ∀ v : HeightOneSpectrum (𝒪 F), IsSelfDualAt σ J L v → IsSelfDualAt σ J L' v → ∃ g : GLoc σ J v, ∀ k : GLoc σ J v, k ∈ stabLoc σ J L' v ↔ g⁻¹ ∗ k ∗ g ∈ stabLoc σ J L v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConjVec

theorem **exists_conjVec** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 ∗ (D.R.r : ℝ) ^ 4) → IdE X → ∃ v₀ : M.Vec, ConjVec X h θ M v₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConjugateWeilDatumHermitianSpaceNeg

theorem **symplOH_neg_neg** (S T : HSp D) (x y : WO S T) : symplOH S T (−x) (−y) = symplOH S T x y

theorem **heisO_refl** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (u : WO S T) (f : WO S T → ℂ) : (fun x => heisO ψ S T u f (−x)) = heisO ψ S T (−u) (fun x => f (−x))

theorem **ratW_neg** (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : ratW T S (−γ) = −ratW T S γ

theorem **actO_neg** (S T : HSp D) (g : GA D S) (h : GA D T) (w : WO S T) : actO S T g h (−w) = −actO S T g h w

theorem **theta_refl** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T fun x => f (−x)

theorem **theta_smul** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (c : ℂ) : IsThetaFunO ψ S T (c • f)

theorem **reflRep_apply** {S T : HSp D} (om : WRep S T) (p : GA D S × GA D T) (f : WO S T → ℂ) : reflRep om p f = fun x => om p (fun y => f (−y)) (−x)

theorem **weil_refl** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →∗ Circle} {om : WRep S T} (hw : IsWeilDatumO ψ S T χS χT om) : IsWeilDatumO ψ S T χS χT (reflRep om)

theorem **det_sub_one** {R : Type} [CommRing R] {n : Type} [Fintype n] [DecidableEq n] (ε : R) (M : Matrix n n R) (h : ∀ i j, ε ∗ (M − 1) i j = (M − 1) i j) : ∃ t, t ∗ ε = M.det − 1

theorem **idem_identity** {R : Type} [CommRing R] {e e' d d' t : R} (e1 : e ∗ e = e) (e2 : e ∗ e' = 0) (e3 : e' ∗ e' = e') (h1 : d' ∗ d = 1) (ht : t ∗ (e + e') = d − 1) : d ∗ (e' ∗ d' + (1 − e')) = e ∗ d + (1 − e)

theorem **conj_det_mul** (V : HSp D) (g : GA D V) : cA D (Formalism.L2.Theta.mt V g).det ∗ (Formalism.L2.Theta.mt V g).det = 1

theorem **detA_val** (V : HSp D) (g : GA D V) : ((detA V g : (AE D)ˣ) : AE D) = (Formalism.L2.Theta.mt V g).det

theorem **conj_detAt** {e : AE D} {V : HSp D} {g : GA D V} {x : (AE D)ˣ} (hx : IsDetAt e V g x) : cA D (x : AE D) = cA D e ∗ cA D (Formalism.L2.Theta.mt V g).det + (1 − cA D e)

theorem **conj_idem** {e : AE D} (he : IsSplitIdem D e) : cA D e ∗ cA D e = cA D e

theorem **det_sub_one_of_suppAt** {e : AE D} {V : HSp D} {g : GA D V} (hg : SuppAt e V g) : ∃ t, t ∗ (e + cA D e) = (Formalism.L2.Theta.mt V g).det − 1

theorem **det_mul_conj** {e : AE D} (he : IsSplitIdem D e) {V : HSp D} {g : GA D V} (hg : SuppAt e V g) {x : (AE D)ˣ} (hx : IsDetAt e V g x) : detA V g ∗ Units.map (cA D).toMonoidHom x = x

theorem **chi_det** {χ : (AE D)ˣ →∗ Circle} {n : ℕ} (hR : ResEta D χ n) {e : AE D} (he : IsSplitIdem D e) {V : HSp D} {g : GA D V} (hg : SuppAt e V g) {x : (AE D)ˣ} (hx : IsDetAt e V g x) : χ (detA V g) = χ x ∗ χ x

theorem **twistRep_apply** {S T : HSp D} (χ : (AE D)ˣ →∗ Circle) (om : WRep S T) (p : GA D S × GA D T) (f : WO S T → ℂ) : twistRep χ om p f = (((χ (detA S p.1))⁻¹ : Circle) : ℂ) • om p f

theorem **weil_twist** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →∗ Circle} {om : WRep S T} (hH : IsHeckeChar D χT) (hR : ResEta D χT T.rank) (hw : IsWeilDatumO ψ S T χS χT⁻¹ om) : IsWeilDatumO ψ S T χS χT (twistRep χT om)

theorem **exists_neg_datum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χA χB : (AE D)ˣ →∗ Circle) (om : WRep A B), IsWeilDatumO ψ A B χA χB om → (∀ f : WO A B → ℂ, IsThetaFunO ψ A B f → IsThetaFunO ψ (negSp A) B fun x => conj (f (−x))) ∧ ∃ om' : WRep (negSp A) B, IsWeilDatumO ψ (negSp A) B χA⁻¹ χB om' ∧ ∀ (g : GA D (negSp A)) (h : GA D B) (f : WO A B → ℂ), IsThetaFunO ψ A B f → om' (g, h) (fun x => conj (f (−x))) = fun x => (((χB (detA A (negEquiv A g)))⁻¹ : Circle) : ℂ) ∗ conj (om (negEquiv A g, h) f (−x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConjugateWeilDatumHermitianSpaceNegStatement

theorem **_root_.GlobalTheory.T.exists_conjugate_weilDatum_hermitianSpaceNeg** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χA χB : (AE D)ˣ →∗ Circle) (om : WRep A B), IsWeilDatumO ψ A B χA χB om → (∀ f : WO A B → ℂ, IsThetaFunO ψ A B f → IsThetaFunO ψ (negSp A) B fun x => conj (f (−x))) ∧ ∃ om' : WRep (negSp A) B, IsWeilDatumO ψ (negSp A) B χA⁻¹ χB om' ∧ ∀ (g : GA D (negSp A)) (h : GA D B) (f : WO A B → ℂ), IsThetaFunO ψ A B f → om' (g, h) (fun x => conj (f (−x))) = fun x => (((χB (detA A (negEquiv A g)))⁻¹ : Circle) : ℂ) ∗ conj (om (negEquiv A g, h) f (−x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConsistentRowChoices

theorem **GlobalTheory.Fam.LiftSide.rowCh_exists** : GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ g : GlobalTheory.Fam.Geo D, ∃ r : ∀ t, RowCh (pk W) (lvP W) u g t, (∀ t, (r t).Is) ∧ ∀ t t', (r t).Same (r t')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConsistentRowChoicesStatement

theorem **_root_.GlobalTheory.T.exists_consistent_row_choices** : GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ g : GlobalTheory.Fam.Geo D, ∃ r : ∀ t, RowCh (pk W) (lvP W) u g t, (∀ t, (r t).Is) ∧ ∀ t t', (r t).Same (r t')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConstMovedVacuumAllTuples

theorem **vacT_twist** {B : Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {θ : InfFam (I.rec7 l Z)} {i : ℕ} {hi : i + 1 ≤ B.R.m} {c₀ : ℂ} {ψ : AddChar (AF (I.rec7 l Z).host.F) 𝕋} {μ₀ μ : (AE (I.rec7 l Z))ˣ →* Circle} {om₀ om : Om ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))} {vac : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) → ℂ} (hv : IsThetaFunO ψ _ _ vac) (h0 : IsWeilDatumO ψ _ _ μ₀ μ₀ om₀) (h1 : IsWeilDatumO ψ _ _ μ μ om) (H : VacT I l Z θ i hi c₀ μ₀ om₀ vac) : VacT I l Z θ i hi c₀ μ om vac

theorem **vacT_one** : GlobalTheory.Published.StrongApproxSU → ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) ψ θ, IsStdChar (I.rec7 l Z) ψ (-1) → IsInfFam θ → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) vac, IsVacO ψ θ ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) (fun _ => 1) (fun _ => 1) (intVec _ _) (intVec _ _) vac → ∀ (a : I.ι) μ om, ChrEq μ (I.C.mu0 (i, hi) * I.xi a) → IsWeilDatumO ψ _ _ μ μ om → ∃ c₀ : ℂ, c₀ ≠ 0 ∧ VacT I l Z θ i hi c₀ μ om vac

theorem **vacT_all** : GlobalTheory.Published.StrongApproxSU → ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ ψ θ, IsStdChar (I.rec7 l Z) ψ (-1) → IsInfFam θ → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) vac, IsVacO ψ θ ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) (fun _ => 1) (fun _ => 1) (intVec _ _) (intVec _ _) vac → ∃ c₀ : ℂ, c₀ ≠ 0 ∧ ∀ (a : I.ι) μ om, ChrEq μ (I.C.mu0 (i, hi) * I.xi a) → IsWeilDatumO ψ _ _ μ μ om → VacT I l Z θ i hi c₀ μ om vac

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsConstMovedVacuumAllTuplesStatement

theorem **_root_.GlobalTheory.T.exists_const_movedVacuum_all_tuples** : GlobalTheory.Published.StrongApproxSU → ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ ψ θ, IsStdChar (I.rec7 l Z) ψ (-1) → IsInfFam θ → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) vac, IsVacO ψ θ ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) (fun _ => 1) (fun _ => 1) (intVec _ _) (intVec _ _) vac → ∃ c₀ : ℂ, c₀ ≠ 0 ∧ ∀ (a : I.ι) μ om, ChrEq μ (I.C.mu0 (i, hi) * I.xi a) → IsWeilDatumO ψ _ _ μ μ om → VacT I l Z θ i hi c₀ μ om vac

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsContinuousLeftInvariantRepConjugateTensor

theorem **act_list** {D : D7} (θ : InfFam D) {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) {K : Type} [NormedAddCommGroup K] [InnerProductSpace ℂ K] (ρ : GA D V →* (K →L[ℂ] K)) (F : (∀ τ, M.H τ) → K) (h : ∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ), ∀ y, F (Function.update y τ (M.rep τ g (y τ))) = ρ (atTau θ V τ g) (F y)) (y : ∀ τ, M.H τ) (l : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin V.rank) ℂ)) (hl : ∀ p ∈ l, p.2 ∈ Formalism.QuantumDeterminants.UH (gramAt D V p.1)) : ρ (l.map fun p => atTau θ V p.1 p.2).prod (F y) = F fun τ => M.rep τ (l.map fun p => if p.1 = τ then p.2 else 1).prod (y τ)

theorem **smooth_of_contDiff** {D : D7} (θ : InfFam D) {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) {K : Type} [NormedAddCommGroup K] [InnerProductSpace ℂ K] (ρ : GA D V →* (K →L[ℂ] K)) (F : (∀ τ, M.H τ) → K) (hF : ContDiff ℝ ∞ F) (h : ∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ), ∀ y, F (Function.update y τ (M.rep τ g (y τ))) = ρ (atTau θ V τ g) (F y)) (v : M.Vec) : GlobalTheory.HolDict.IsSmoothVector' θ V ρ (F fun τ => (v τ : M.H τ))

theorem **contDiff_anti** {A B : Type} [NormedAddCommGroup A] [InnerProductSpace ℂ A] [NormedAddCommGroup B] [InnerProductSpace ℂ B] (c : A ≃ₗᵢ⋆[ℂ] B) : ContDiff ℝ ∞ c

theorem **exists_contRep** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (GlobalTheory.ConjRowMult.vecC X M c u))) ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsContinuousLeftInvariantRepConjugateTensorStatement

theorem **_root_.GlobalTheory.T.exists_continuous_leftInvariant_rep_conjugateTensor** : GlobalTheory.Published.DiscreteSeriesClassification → ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (θ : InfFam D) (S : D.what₂.Tau → SPar) (M : DSModel (D.sp.Y 0) S) (J : Set (GA D (D.sp.Y 0))) (χ : GA D (D.sp.Y 0) → ℂ) (t : GA D (D.sp.Y 0)) (c : ∀ τ, M.H τ ≃ₗᵢ⋆[ℂ] X.B.H (X.pl.symm τ)), S = parB (Ptw D) 0 → IsInfFam θ → GlobalTheory.Count.GapGe X.B (100 * (D.R.r : ℝ) ^ 4) → IdE X → Matches X h J χ t → (∀ τ, ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D (D.sp.Y 0) τ), ∀ w, c τ (M.rep τ g w) = X.B.σ (X.pl.symm τ) (uMat ((iotaX X h (atTau θ (D.sp.Y 0) τ g)).1 (X.pl.symm τ))) (c τ w)) → ∀ Φ : X.B.Hs →L[ℂ] L2X X, Φ ∈ X.B.I.eHom X.B.σs X.B.μf X.B.fund X.B.R X.B.Re → (∀ u : M.Vec, ∃ ψ, Formalism.L2.SG.EvalTOntoRep.IsContRep X (Φ (X.B.tens (Glob

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsContinuousLocalGLEmbeddingSplitPlace

theorem **exists_gl** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (V : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k V.rank Y), ∃ gl : GL Y k →* GA D V, Continuous gl ∧ (∀ x, SuppAt q.e V (gl x) ∧ (mt V (gl x)).map q.pr = F.β * x.val * F.γ) ∧ ∀ g x, SuppAt q.e V g → (mt V g).map q.pr = F.β * x.val * F.γ → g = gl x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsContinuousLocalGLEmbeddingSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_continuous_local_GL_embedding_splitPlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (V : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k V.rank Y), ∃ gl : GL Y k →* GA D V, Continuous gl ∧ (∀ x, SuppAt q.e V (gl x) ∧ (mt V (gl x)).map q.pr = F.β * x.val * F.γ) ∧ ∀ g x, SuppAt q.e V g → (mt V g).map q.pr = F.β * x.val * F.γ → g = gl x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsContrBilinear

theorem **comp_mem_SB** {X Y : Type} [TopologicalSpace X] [TopologicalSpace Y] [T2Space X] {f : Y → ℂ} (hf : f ∈ SB Y) {i : X → Y} (hi : Continuous i) {r : Y → X} (hr : Continuous r) (hri : ∀ x, r (i x) = x) : (fun x => f (i x)) ∈ SB X

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierHyperbolicTermsNearRefere

theorem **nat_smul_mL** (k : ℕ) {x : VA X.B.I} (hx : x ∈ mL X.B.I X.B.g) : (k : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • x ∈ mL X.B.I X.B.g

theorem **dvd_smul_mL** {d M : ℕ} (hd : d ∣ M) {x : VA X.B.I} (hx : (d : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • x ∈ mL X.B.I X.B.g) : (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • x ∈ mL X.B.I X.B.g

theorem **term_near** {v0 w0 v w y : VA X.B.I} (hw0 : w0 ∈ X.B.I.Lf) (hy : y ∈ X.B.I.Lf) {c1 c2 c3 : ℕ} (h1 : (c1 : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (v0 − v) ∈ mL X.B.I X.B.g) (h2 : (c2 : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (w0 − w) ∈ mL X.B.I X.B.g) (h3 : (c3 : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • v ∈ X.B.I.Lf) : ((c1 ∗ c2 ∗ c3 : ℕ) : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I w y • v − hA X.B.I w0 y • v0) ∈ mL X.B.I X.B.g

theorem **ef_near_ref** : ∀ (D : D7) (X : HidaSetting D), ∃ M : ℕ, Nat.Coprime M X.B.g ∧ ∀ y ∈ X.B.I.Lf, ∀ a, (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I (GlobalTheory.ScalW.fV X a) y • GlobalTheory.ScalW.eV X a − hA X.B.I (X.f0 a) y • X.e0 a) ∈ mL X.B.I X.B.g ∧ (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I (GlobalTheory.ScalW.eV X a) y • GlobalTheory.ScalW.fV X a − hA X.B.I (X.e0 a) y • X.f0 a) ∈ mL X.B.I X.B.g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierHyperbolicTermsNearRefereStatement

theorem **_root_.GlobalTheory.T.exists_coprime_multiplier_hyperbolic_terms_near_reference** : ∀ (D : D7) (X : HidaSetting D), ∃ M : ℕ, Nat.Coprime M X.B.g ∧ ∀ y ∈ X.B.I.Lf, ∀ a, (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I (GlobalTheory.ScalW.fV X a) y • GlobalTheory.ScalW.eV X a − hA X.B.I (X.f0 a) y • X.e0 a) ∈ mL X.B.I X.B.g ∧ (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I (GlobalTheory.ScalW.eV X a) y • GlobalTheory.ScalW.fV X a − hA X.B.I (X.e0 a) y • X.f0 a) ∈ mL X.B.I X.B.g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierLevelCharacterSumCongr

theorem **nsmul_mL'** {g k : ℕ} {v : Formalism.L2.SG.HidaSetting.VA I} (hv : v ∈ Formalism.L2.SG.HidaSetting.mL I g) : ((k : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • v) ∈ Formalism.L2.SG.HidaSetting.mL I g

theorem **nsmul_Lf'** {k : ℕ} {v : Formalism.L2.SG.HidaSetting.VA I} (hv : v ∈ I.Lf) : ((k : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • v) ∈ I.Lf

theorem **mL_le_Lf'** (g : ℕ) : Formalism.L2.SG.HidaSetting.mL I g ≤ I.Lf

theorem **K_mL'** {x : I.Gf} (hx : x ∈ I.Kset) {g : ℕ} {v : Formalism.L2.SG.HidaSetting.VA I} (hv : v ∈ Formalism.L2.SG.HidaSetting.mL I g) : I.fmat x ∗ᵥ v ∈ Formalism.L2.SG.HidaSetting.mL I g

theorem **K_sub_mL'** {x : I.Gf} (hx : x ∈ I.Kset) {g : ℕ} {v : Formalism.L2.SG.HidaSetting.VA I} (hv : v ∈ Formalism.L2.SG.HidaSetting.mL I g) : (I.fmat x − 1) ∗ᵥ v ∈ Formalism.L2.SG.HidaSetting.mL I g

theorem **hcn_of** {x : I.Gf} (hx : x ∈ I.Kset) {g : ℕ} {n N N0 : Formalism.L2.SG.HidaSetting.VA I} (hN : I.fmat x ∗ᵥ N0 = N) {c : ℕ} (h : (c : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • (n − N0) ∈ Formalism.L2.SG.HidaSetting.mL I g) : (c : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • (I.fmat x ∗ᵥ n) − (c : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • N ∈ Formalism.L2.SG.HidaSetting.mL I g

theorem **conj_sub** (x m : I.Gf) (n : Formalism.L2.SG.HidaSetting.VA I) : (I.fmat (x⁻¹ ∗ m ∗ x) − 1) ∗ᵥ n = I.fmat x⁻¹ ∗ᵥ ((I.fmat m − 1) ∗ᵥ (I.fmat x ∗ᵥ n))

theorem **term_cong** {g : ℕ} {x m : I.Gf} (hx : x ∈ I.Kset) (hm : m ∈ I.Kset) {f F n N : Formalism.L2.SG.HidaSetting.VA I} (hf : f ∈ I.Lf) (hn : n ∈ I.Lf) (hFx : I.fmat x ∗ᵥ F = F) {c : ℕ} (hcf : (c : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • (f − F) ∈ Formalism.L2.SG.HidaSetting.mL I g) (hcn : (c : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • (I.fmat x ∗ᵥ n) − (c : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) • N ∈ Formalism.L2.SG.HidaSetting.mL I g) : Formalism.L2.SG.HidaSetting.ZMod' I g (((c ∗ c : ℕ) : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) ∗ (Formalism.L2.SG.HidaSetting.hA I f ((I.fmat (x⁻¹ ∗ m ∗ x) − 1) ∗ᵥ n) − Formalism.L2.SG.HidaSetting.hA I F ((I.fmat m − 1) ∗ᵥ N)))

theorem **term_each** (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (m : X.B.I.Gf) (hm : m ∈ X.B.I.Kset) (a : Fin D.R.rS) : ∃ c : ℕ, Nat.Coprime c X.B.g ∧ ZMod' X.B.I X.B.g ((c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) ∗ (hA X.B.I (X.f0 a) ((X.B.I.fmat ((X.xu u)⁻¹ ∗ m ∗ X.xu u) − 1) ∗ᵥ nxt X.e0 X.z0 a) − hA X.B.I (incl X.B.I (X.fR a)) ((X.B.I.fmat m − 1) ∗ᵥ incl X.B.I (nxt X.eR (X.zR u) a))))

theorem **nsum_cong** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (m : X.B.I.Gf), m ∈ X.B.I.Kset → ∃ c : ℕ, Nat.Coprime c X.B.g ∧ ZMod' X.B.I X.B.g ((c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) ∗ ∑ a, (hA X.B.I (X.f0 a) ((X.B.I.fmat ((X.xu u)⁻¹ ∗ m ∗ X.xu u) − 1) ∗ᵥ nxt X.e0 X.z0 a) − hA X.B.I (incl X.B.I (X.fR a)) ((X.B.I.fmat m − 1) ∗ᵥ incl X.B.I (nxt X.eR (X.zR u) a))))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierLevelCharacterSumCongrStatement

theorem **_root_.GlobalTheory.T.exists_coprime_multiplier_levelCharacter_sum_congr** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (m : X.B.I.Gf), m ∈ X.B.I.Kset → ∃ c : ℕ, Nat.Coprime c X.B.g ∧ ZMod' X.B.I X.B.g ((c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) ∗ ∑ a, (hA X.B.I (X.f0 a) ((X.B.I.fmat ((X.xu u)⁻¹ ∗ m ∗ X.xu u) − 1) ∗ᵥ nxt X.e0 X.z0 a) − hA X.B.I (incl X.B.I (X.fR a)) ((X.B.I.fmat m − 1) ∗ᵥ incl X.B.I (nxt X.eR (X.zR u) a))))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierNegativeVectorTermNear

theorem **znear_term** : ∀ (D : D7) (X : HidaSetting D) (u : D.what₂.Word), ∃ M : ℕ, Nat.Coprime M X.B.g ∧ ∀ y ∈ X.B.I.Lf, (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I (X.B.I.fmat (X.xu u)⁻¹ ∗ᵥ GlobalTheory.ScalW.zV X u) y • (X.B.I.fmat (X.xu u)⁻¹ ∗ᵥ GlobalTheory.ScalW.zV X u) − hA X.B.I X.z0 y • X.z0) ∈ mL X.B.I X.B.g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierNegativeVectorTermNearStatement

theorem **_root_.GlobalTheory.T.exists_coprime_multiplier_negativeVector_term_near** : ∀ (D : D7) (X : HidaSetting D) (u : D.what₂.Word), ∃ M : ℕ, Nat.Coprime M X.B.g ∧ ∀ y ∈ X.B.I.Lf, (M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) • (hA X.B.I (X.B.I.fmat (X.xu u)⁻¹ ∗ᵥ GlobalTheory.ScalW.zV X u) y • (X.B.I.fmat (X.xu u)⁻¹ ∗ᵥ GlobalTheory.ScalW.zV X u) − hA X.B.I X.z0 y • X.z0) ∈ mL X.B.I X.B.g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierOneSubNormCongrZero

theorem **one_sub_T_zmod** : ∀ (D : D7) (X : HidaSetting D) (u : D.what₂.Word), ∃ M : ℕ, Nat.Coprime M X.B.g ∧ ZMod' X.B.I X.B.g ((M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) * (1 - (((D.mem u).A.T : ℕ) : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierOneSubNormCongrZeroStatement

theorem **_root_.GlobalTheory.T.exists_coprime_multiplier_one_sub_norm_congr_zero** : ∀ (D : D7) (X : HidaSetting D) (u : D.what₂.Word), ∃ M : ℕ, Nat.Coprime M X.B.g ∧ ZMod' X.B.I X.B.g ((M : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F) * (1 - (((D.mem u).A.T : ℕ) : X.B.I.E ⊗[X.B.I.F] Af X.B.I.F)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierUnipotentPartCongrZero

theorem **natCast_int** (c : ℕ) : (c : RA) ∈ IntAf X.B.I.F X.B.I.E

theorem **sigA_natCast** (c : ℕ) : sigA X.B.I (c : RA) = c

theorem **natsmul_Lf** (c : ℕ) {y : VA X.B.I} (hy : y ∈ X.B.I.Lf) : (c : RA) • y ∈ X.B.I.Lf

theorem **natsmul_mL** (c : ℕ) {y : VA X.B.I} (hy : y ∈ mL X.B.I X.B.g) : (c : RA) • y ∈ mL X.B.I X.B.g

theorem **mL_le_Lf** : mL X.B.I X.B.g ≤ X.B.I.Lf

theorem **intsmul_Lf** {t : RA} (ht : t ∈ IntAf X.B.I.F X.B.I.E) {y : VA X.B.I} (hy : y ∈ X.B.I.Lf) : t • y ∈ X.B.I.Lf

theorem **Z_int** {t : RA} (ht : ZMod' X.B.I X.B.g t) : t ∈ IntAf X.B.I.F X.B.I.E

theorem **hA_ef** (a b : Fin D.R.rS) : hA X.B.I (incl X.B.I (X.eR a)) (incl X.B.I (X.fR b)) = if a = b then 1 else 0

theorem **hA_fe** (a b : Fin D.R.rS) : hA X.B.I (incl X.B.I (X.fR b)) (incl X.B.I (X.eR a)) = if a = b then 1 else 0

theorem **exists_mult** : ∃ c : ℕ, Nat.Coprime c X.B.g ∧ ∀ b : Fin D.R.rS, (c : RA) • (X.e0 b - incl X.B.I (X.eR b)) ∈ mL X.B.I X.B.g ∧ (c : RA) • (X.f0 b - incl X.B.I (X.fR b)) ∈ mL X.B.I X.B.g

theorem **hA_sum_right** (p : VA X.B.I) (v : Fin D.R.rS → VA X.B.I) : hA X.B.I p (∑ b, v b) = ∑ b, hA X.B.I p (v b)

theorem **core** (u : D.what₂.Word) (m : X.B.I.Gf) (hmK : m ∈ X.B.I.Kset) (hmN : m ∈ X.NfinU u) (c : ℕ) (hc : ∀ b : Fin D.R.rS, (c : RA) • (X.e0 b - incl X.B.I (X.eR b)) ∈ mL X.B.I X.B.g ∧ (c : RA) • (X.f0 b - incl X.B.I (X.fR b)) ∈ mL X.B.I X.B.g) {p w : VA X.B.I} (hp : p ∈ X.B.I.Lf) (hw : w ∈ X.B.I.Lf) (ht : ∀ b, ZMod' X.B.I X.B.g (hA X.B.I (incl X.B.I (X.eR b)) ((c : RA) • w))) (hpe : ∀ b, ZMod' X.B.I X.B.g ((c : RA) * hA X.B.I p (incl X.B.I (X.eR b)))) : ZMod' X.B.I X.B.g ((c : RA) ^ 4 * hA X.B.I p ((X.B.I.fmat m - 1) *ᵥ w))

theorem **conj_sub** (x m : X.B.I.Gf) (v : VA X.B.I) : (X.B.I.fmat (x⁻¹ * m * x) - 1) *ᵥ v = X.B.I.fmat x⁻¹ *ᵥ ((X.B.I.fmat m - 1) *ᵥ (X.B.I.fmat x *ᵥ v))

theorem **term** (u : D.what₂.Word) (m : X.B.I.Gf) (hmK : m ∈ X.B.I.Kset) (hmN : m ∈ X.NfinU u) (c : ℕ) (hc : ∀ b : Fin D.R.rS, (c : RA) • (X.e0 b - incl X.B.I (X.eR b)) ∈ mL X.B.I X.B.g ∧ (c : RA) • (X.f0 b - incl X.B.I (X.fR b)) ∈ mL X.B.I X.B.g) (i j : Fin D.R.k) : ZMod' X.B.I X.B.g ((c : RA) ^ 4 * hA X.B.I (X.bwd i) ((X.B.I.fmat ((X.xu u)⁻¹ * m * X.xu u) - 1) *ᵥ X.bw j))

theorem **ukPart_zmod** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (m : X.B.I.Gf), m ∈ X.B.I.Kset → m ∈ X.NfinU u → ∃ c : ℕ, Nat.Coprime c X.B.g ∧ ZMod' X.B.I X.B.g ((c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) * ∑ i : Fin D.R.k, if h : i.1 + 1 < D.R.k then hA X.B.I (X.bwd i) ((X.B.I.fmat ((X.xu u)⁻¹ * m * X.xu u) - 1) *ᵥ X.bw ⟨i.1 + 1, h⟩) else 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCoprimeMultiplierUnipotentPartCongrZeroStatement

theorem **_root_.GlobalTheory.T.exists_coprime_multiplier_unipotentPart_congr_zero** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (m : X.B.I.Gf), m ∈ X.B.I.Kset → m ∈ X.NfinU u → ∃ c : ℕ, Nat.Coprime c X.B.g ∧ ZMod' X.B.I X.B.g ((c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) * ∑ i : Fin D.R.k, if h : i.1 + 1 < D.R.k then hA X.B.I (X.bwd i) ((X.B.I.fmat ((X.xu u)⁻¹ * m * X.xu u) - 1) *ᵥ X.bw ⟨i.1 + 1, h⟩) else 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCopy

theorem **chiSymF_map** {κ K : Type} [CommRing κ] [Field K] (emb : K →+* ℂ) (ψ : AddChar κ K) (n k rS : ℕ) (M : HPsymF κ n k rS) : ((chiSymF ((emb : K →* ℂ).compAddChar ψ) n k rS M : ℂˣ) : ℂ) = emb ((chiSymF ψ n k rS M : Kˣ) : K)

theorem **lev_iff** (hc : CMData F E) {n n' : ℕ} (e : n' = n) [NeZero n] {J' : Matrix (Fin n') (Fin n') E} {J : Matrix (Fin n) (Fin n) E} (hJ' : J' = 1) (hJ : J = 1) {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 F))} (S : StepLevel F E hc.sig J T f bad) (hL : S.Lb = Formalism.L2.Col.stdLat E n) {r t s : ℕ} (h : r ≤ n) (h' : r ≤ n') (ht : 0 < t) (hn : n = r + s * t) {SG : Finset (Pl E)} (P : LevFam E K r SG) (χ : ∀ w : SG, ℕ → (𝒪 E ⧸ w.1.asIdeal)ˣ →* Kˣ) (emb : K →+* ℂ) (hR : RdLev hc S.U S.chi Set.univ (parFam h t s P χ) emb) (P' : LevelP SG r ℂ) (cg : ℕ → ∀ PP : Pl E, (kP PP)ˣ →* ℂˣ) (xq : ∀ PP : Pl E, (kP PP)ˣ →* ℂˣ) (hH : ∀ g, g ∈ P'.H ↔ ∀ w, g w ∈ (P w).H) (hchi : ∀ (g : P'.H) (hg : ∀ w, g.1 w ∈ (P w).H), ((P'.chi g : ℂˣ) : ℂ) = ∏ w, emb (((P w).lam (g.1 w, hg w) : Kˣ) : K)) (hcg : ∀ c (w : SG) a, ((cg (c + 1) w.1 a : ℂˣ) : ℂ) = emb ((χ w c a : Kˣ) : K)) (g : GL (Fin n') (E ⊗[F] Formalism.L

theorem **exists_copy** {A : Type} [CommRing A] {n n' : ℕ} (e : n' = n) (g : GL (Fin n') A) : ∃ y : GL (Fin n) A, ∀ i j, g.1 i j = y.1 (finCongr e i) (finCongr e j)

theorem **rd_all** {B : Base} {X : PreX B.toPre} (I : TC B X) [NeZero (B.R.N B.R.m - 1)] {Lb U chi y lat st} (h : I.ReadsP I.mu0 Lb U chi y lat st) (c : (I.stX I.mu0).Cl) : RdLev (E

theorem **topLev_std** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z) (v : B.Word), ∀ c, IsComOf I l Z c → ∀ top : StAt (I.rec7 l Z).sp.X', top.β = 1 → top.st.Is c B.R.m → GlobalTheory.Core.Rk.Top.TopLev (w.C.forms v).clsP (levS _ top.β top.L) (Lev.levChi _ top.β top.L)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCutoffOfAddCharTranslationInvariantFunctio

theorem **dot_eq_one** {ψ : AddChar k Circle} {c : kˣ} (h : HasConductor ψ c) {n : Type} [Fintype n] (R : kˣ) {t y : n → k} (ht : t ∈ box n (c * R⁻¹)) (hy : y ∈ box n R) : ψ (t ⬝ᵥ y) = 1

theorem **exists_dot_ne_one** {ψ : AddChar k Circle} {c : kˣ} (h : HasConductor ψ c) {n : Type} [Fintype n] (s : kˣ) {y : n → k} (hy : y ∉ box n s) : ∃ b ∈ box n (c * s⁻¹), ψ (b ⬝ᵥ y) ≠ 1

theorem **sum_char** {ψ : AddChar k Circle} {c : kˣ} (h : HasConductor ψ c) {n : Type} [Fintype n] (s R : kˣ) (S : Finset (n → k)) (hS : ∀ b ∈ S, b ∈ box n (c * s⁻¹)) (hU : ∀ a ∈ box n (c * s⁻¹), ∃! b, b ∈ S ∧ a - b ∈ box n (c * R⁻¹)) {y : n → k} (hy : y ∈ box n R) : ∑ b ∈ S, (ψ (b ⬝ᵥ y) : ℂ) = (box n s).indicator (fun _ => (1 : ℂ)) y * S.card

theorem **exists_cutoff_of_char_invariant** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ (I₀ I₁ I₂ : Type) [Fintype I₀] [Fintype I₁] [Fintype I₂] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB ((I₀ → k) × (I₁ → k) × (I₂ → k)) →ₗ[ℂ] M), (∀ (b : I₁ → k) (Φ Φ' : SB ((I₀ → k) × (I₁ → k) × (I₂ → k))), (∀ x, Φ'.1

x = (ψ (b ·ᵥ x.2.1) : ℂ) * Φ.1 x) → l Φ' = l Φ) → ∀ (Φ : SB ((I₀ → k) × (I₁ → k) × (I₂ → k))) (s : kˣ), ∃ Φ' : SB ((I₀ → k) × (I₁ → k) × (I₂ → k)), (∀ x, Φ'.1 x = (box I₁ s).indicator (fun _ => (1 : ℂ)) x.2.1 * Φ.1 x) ∧ l Φ = l Φ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsCutoffOfAddCharTranslationInvariantFunctioStatement

theorem **_root_.GlobalTheory.T.exists_cutoff_of_addChar_translation_invariant_functional** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ (I₀ I₁ I₂ : Type) [Fintype I₀] [Fintype I₁] [Fintype I₂] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB ((I₀ → k) × (I₁ → k) × (I₂ → k)) →ₗ[ℂ] M), (∀ (b : I₁ → k) (Φ Φ' : SB ((I₀ → k) × (I₁ → k) × (I₂ → k))), (∀ x, Φ'.1 x = (ψ (b ·ᵥ x.2.1) : ℂ) * Φ.1 x) → l Φ' = l Φ) → ∀ (Φ : SB ((I₀ → k) × (I₁ → k) × (I₂ → k))) (s : kˣ), ∃ Φ' : SB ((I₀ → k) × (I₁ → k) × (I₂ → k)), (∀ x, Φ'.1 x = (box I₁ s).indicator (fun _ => (1 : ℂ)) x.2.1 * Φ.1 x) ∧ l Φ = l Φ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDenomNotMemOfPrInt

theorem **mem_prime_iff_pr_inv_notMem** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (P : HeightOneSpectrum (𝒪 D.host.E)) (h : Formalism.L2.NTheta.IsIdemAt p.e P) (x : 𝒪 D.host.E) (hx : x ≠ 0) : x ∈ P.asIdeal ↔ p.pr (incA D ((x : D.host.E)⁻¹)) ∉ 𝒪[k]

theorem **exists_denom_notMem_of_pr_int** : ∀ {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (P : HeightOneSpectrum (𝒪 D.host.E)), Formalism.L2.NTheta.IsIdemAt p.e P → ∀ a : D.host.E, p.pr (incA D a) ∈ 𝒪[k] → ∃ s : 𝒪 D.host.E, s ∉ P.asIdeal ∧ ∃ c : 𝒪 D.host.E, algebraMap (𝒪 D.host.E) D.host.E c = algebraMap (𝒪 D.host.E) D.host.E s * a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDenominatorAvoidingGivenPrimes

theorem **exists_den_avoid** : ∀ (E : Type) [Field E] [NumberField E] (a b : ℕ) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)) (G : HeightOneSpectrum (𝒪 E) → Prop) (c : ℕ), 0 < c → ∀ x : Formalism.L2.Col.HomE E a b, (c : E) • x ∈ M → (∀ Q : HeightOneSpectrum (𝒪 E), ¬ G Q → x ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M) → ∃ d : 𝒪 E, d ≠ 0 ∧ (d : E) • x ∈ M ∧ ∀ Q : HeightOneSpectrum (𝒪 E), (c : 𝒪 E) ∈ Q.asIdeal → d ∈ Q.asIdeal → G Q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDenominatorAvoidingGivenPrimesStatement

theorem **_root_.GlobalTheory.T.exists_denominator_avoiding_given_primes** : ∀ (E : Type) [Field E] [NumberField E] (a b : ℕ) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)) (G : HeightOneSpectrum (𝒪 E) → Prop) (c : ℕ), 0 < c → ∀ x : Formalism.L2.Col.HomE E a b, (c : E) • x ∈ M → (∀ Q : HeightOneSpectrum (𝒪 E), ¬ G Q → x ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M) → ∃ d : 𝒪 E, d ≠ 0 ∧ (d : E) • x ∈ M ∧ ∀ Q : HeightOneSpectrum (𝒪 E), (c : 𝒪 E) ∈ Q.asIdeal → d ∈ Q.asIdeal → G Q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDescentDataChain

theorem **main** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → (∀ j, j + 1 < D.R.m → ∃ dG, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) dG ∧ ∀ t, ∃ dQ, IsDQ W t j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) dQ ∧ (f.row t).vP j = comb (pk W).ψ _ _ dQ (comb (pk W).ψ _ _ dG ((pk W).vac j))) ∧ ∀ j h, ∃ dG, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) dG ∧ ∀ t, ∃ dQ, IsDQ W t (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) dQ ∧ (f.row t).vT j h = comb (pk W).ψ _ _ dQ (comb (pk W).ψ _ _ dG (vacT (pk W) j h))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDescentDataChainStatement

theorem **_root_.GlobalTheory.T.exists_descent_data_chain** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → (∀ j, j + 1 < D.R.m → ∃ dG, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) dG ∧ ∀ t, ∃ dQ, IsDQ W t j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) dQ ∧ (f.row t).vP j = comb (pk W).ψ _ _ dQ (comb (pk W).ψ _ _ dG ((pk W).vac j))) ∧ ∀ j h, ∃ dG, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) dG ∧ ∀ t, ∃ dQ, IsDQ W t (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) dQ ∧ (f.row t).vT j h = comb (pk W).ψ _ _ dQ (comb (pk W).ψ _ _ dG (vacT (pk W) j h))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDiagonalFrameChangeUnweightingSymEdges

theorem **isUnit_res** {E : Type} [Field E] [NumberField E] (PP : Formalism.L2.DIV.Pl E) (x : Formalism.L2.DIV.kP PP) (hx : x ≠ 0) : IsUnit x

theorem **exists_diag_frame** : ∀ {D : D7} {n : ℕ} (ρ k : ℕ) (Pg : Finset (Pl D.host.E)) (Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) (β₀ : GL (Fin n) (Formalism.IndefR.Lev.EAf D)), Formalism.L2.DIV.IsFrame β₀ Λ → ∀ a : ∀ PP : Pl D.host.E, Fin n → Fin n → kP PP, (∀ PP ∈ Pg, ∀ i j, Formalism.L2.DIV.symEdge ρ k i j → a PP i j ≠ 0) → ∃ β₃ : GL (Fin n) (Formalism.IndefR.Lev.EAf D), Formalism.L2.DIV.IsFrame β₃ Λ ∧ Formalism.L2.DIV.St β₃ = Formalism.L2.DIV.St β₀ ∧ ∀ PP ∈ Pg, ∀ y ∈ Formalism.L2.DIV.St β₀, (Formalism.L2.DIV.redFn β₃ PP y ∈ Formalism.L2.DIV.unipSym (kP PP) n ρ k ↔ Formalism.L2.DIV.redFn β₀ PP y ∈ Formalism.L2.DIV.unipSym (kP PP) n ρ k) ∧ (∑ i, ∑ j, if Formalism.L2.DIV.symEdge ρ k i j then a PP i j * ((Formalism.L2.DIV.redFn β₀ PP y : GL (Fin n) (kP PP)) : Matrix (Fin n) (Fin n) (kP PP)) i j else 0) = Formalism.L2.DIV.edgeSum (Formalism.L2.DIV.symEdge ρ k) (Formalism.L2.DIV.redFn β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDiagonalFrameChangeUnweightingSymEdgesStatement

theorem **_root_.GlobalTheory.T.exists_diagonal_frame_change_unweighting_symEdges** : ∀ {D : D7} {n : ℕ} (ρ k : ℕ) (Pg : Finset (Pl D.host.E)) (Λ : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) (β₀ : GL (Fin n) (Formalism.IndefR.Lev.EAf D)), Formalism.L2.DIV.IsFrame β₀ Λ → ∀ a : ∀ PP : Pl D.host.E, Fin n → Fin n → kP PP, (∀ PP ∈ Pg, ∀ i j, Formalism.L2.DIV.symEdge ρ k i j → a PP i j ≠ 0) → ∃ β₃ : GL (Fin n) (Formalism.IndefR.Lev.EAf D), Formalism.L2.DIV.IsFrame β₃ Λ ∧ Formalism.L2.DIV.St β₃ = Formalism.L2.DIV.St β₀ ∧ ∀ PP ∈ Pg, ∀ y ∈ Formalism.L2.DIV.St β₀, (Formalism.L2.DIV.redFn β₃ PP y ∈ Formalism.L2.DIV.unipSym (kP PP) n ρ k ↔ Formalism.L2.DIV.redFn β₀ PP y ∈ Formalism.L2.DIV.unipSym (kP PP) n ρ k) ∧ (∑ i, ∑ j, if Formalism.L2.DIV.symEdge ρ k i j then a PP i j * ((Formalism.L2.DIV.redFn β₀ PP y : GL (Fin n) (kP PP)) : Matrix (Fin n) (Fin n) (kP PP)) i j else 0) = Formalism.L2.DIV.edgeSum (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDoubleDescentOfParity

theorem **isHalf_add_left** (p : Place D k) {u : WO S T} (hu : IsHalf p.e u) : IsHalf (p.e + cA D p.e) u

theorem **isHalf_add_right** (p : Place D k) {u : WO S T} (hu : IsHalf (cA D p.e) u) : IsHalf (p.e + cA D p.e) u

theorem **isHalf_of_mem_vacSet** {ψ : AddChar (AF D.host.F) ℂ} {p : Place D k} {FS : GlobalTheory.LocMod.Frame k S.rank Y'} {FT : GlobalTheory.LocMod.Frame k T.rank Y} {u : WO S T} (hu : u ∈ vacSet ψ p FS FT) : IsHalf (p.e + cA D p.e) u

theorem **exists_fixed_vacSet** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (p : Place D k) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (U : Set (WO S T)) (hU : ∀ u ∈ U, ∀ w, IsHalf (p.e + cA D p.e) w → symplO S T u w = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (f0 : f ≠ 0) (hfU : ∀ u ∈ U, heisO ψ S T u f = f) : ∃ g, IsThetaFunO ψ S T g ∧ g ≠ 0 ∧ ∀ u ∈ U ∪ vacSet ψ p FS FT, heisO ψ S T u g = g

theorem **exists_model_of_fixed** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (p : Place D k) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (h : ∀ u ∈ vacSet ψ p FS FT, heisO ψ S T u f = f) : ∃ Tm O, p.IsModel ψ FS FT Tm ∧ IsOne O ∧ Tm O = f

theorem **idem_orth** {k' : Type} [Field k'] [ValuativeRel k'] [TopologicalSpace k'] [IsNonarchimedeanLocalField k'] (q : GlobalTheory.FinPl.SplitPt D k) (q' : GlobalTheory.FinPl.SplitPt D k') (h : q.v ≠ q'.v) : cA D (q.e + cA D q.e) * (q'.e + cA D q'.e) = 0

theorem **exists_fixed_all** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ι : Type} [Finite ι] (a : ι → AE D) (ha : ∀ i j, i ≠ j → cA D (a i) * a j = 0) (V : ι → Set (WO S T)) (hV : ∀ i, ∀ u ∈ V i, IsHalf (a i) u) (step : ∀ i (U : Set (WO S T)), (∀ u ∈ U, ∀ w, IsHalf (a i) w → symplO S T u w = 0) → ∀ f, IsThetaFunO ψ S T f → f ≠ 0 → (∀ u ∈ U, heisO ψ S T u f = f) → ∃ g, IsThetaFunO ψ S T g ∧ g ≠ 0 ∧ ∀ u ∈ U ∪ V i, heisO ψ S T u g = g) : ∃ f, IsThetaFunO ψ S T f ∧ f ≠ 0 ∧ ∀ i, ∀ u ∈ V i, heisO ψ S T u f = f

theorem **exists_vac** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {σ : ℕ} {S : Dbl2 ψ σ} {ι : Type} [Finite ι] (Q : ι → QStep S) (hQ : Function.Injective fun i => (Q i).q.v) : ∃ f₁ f₂, f₁ ≠ 0 ∧ f₂ ≠ 0 ∧ IsThetaFunO ψ _ _ f₁ ∧ IsThetaFunO ψ _ _ f₂ ∧ ∀ i, (Q i).Vac f₁ f₂

theorem **exists_open_fix** {Y : Type} [Fintype Y] [DecidableEq Y] (L : Level (GL Y k)) (hL : IsOpen (L.J : Set (GL Y k))) (α : GL Y k) : ∃ U : Subgroup (GL Y k), IsOpen (U : Set (GL Y k)) ∧ ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL Y k) A), IsSmoothRep σ → ∀ v : A, (∀ j : L.J, σ (j : GL Y k) v = (L.lam j : ℂ) • v) → ∀ u ∈ U, σ u (σ α v) = σ α v

theorem **exists_isOneComb** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {P : ℕ} (h1 : 1 ≤ P) (hP : P ≤ c) (L : Level (GL (n ⊕ Fin c) k)) (hL : IsOpen (L.J : Set (GL (n ⊕ Fin c) k))) (α : GL (n ⊕ Fin c) k) : ∃ e, IsOneComb ψ ν P L α e

theorem **exists_covers** {ι V : Type} [Zero V] {p : ℕ} {J : Subgroup (GL (n ⊕ Fin c) k)} (s : Finset ι) (f : ι → GL (n ⊕ Fin c) k → V) (hf : ∀ i, ∃ T, Covers p J T (f i)) (t₀ : GL (n ⊕ Fin c) k) : ∃ T, t₀ ∈ T ∧ ∀ i ∈ s, Covers p J T (f i)

theorem **exists_covers_flat** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (ν : Measure k) [ν.IsAddHaarMeasure] {μ : kˣ →* ℂˣ} (hμ : IsOpen (μ.ker : Set kˣ)) {J : Subgroup (GL (n ⊕ Fin (c + 1)) k)} (hJ : IsOpen (J : Set (GL (n ⊕ Fin (c + 1)) k))) (Φ : SB (Matrix (n ⊕ Fin (c + 1)) (n' ⊕ Fin (c + 1 + 1)) k)) (α : GL (n' ⊕ Fin (c + 1 + 1)) k) : ∃ T, Covers 1 J T fun h => GlobalTheory.StepData.flat ψ ν μ Φ.1 α h

theorem **xi_mem_SB** (ψ : AddChar k Circle) (ν : Measure k) (μ : kˣ →* ℂˣ) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (J : Subgroup (GL (n ⊕ Fin c) k)) (b : GL (n ⊕ Fin c) k) (h' : GL (n' ⊕ Fin (c + 1)) k) (h : GL (n ⊕ Fin c) k) : GlobalTheory.StepData.xi ψ ν μ Φ.1 J b h' h ∈ SB (Matrix n n' k)

theorem **exists_isCombAt** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (p : Place D k) {S T : HSp D} (FS : GlobalTheory.LocMod.Frame k S.rank n') (FT : GlobalTheory.LocMod.Frame k T.rank n) {Ξ : Matrix n n' k → ℂ} (hΞ : Ξ ∈ SB (Matrix n n' k)) : ∃ d, p.IsCombAt ψ FS FT d Ξ

theorem **prX_Cgrp_KO** : prX (Cgrp (KO k (n ⊕ Fin c)) 1) = KO k n

theorem **exists_qstep** (hψ : IsStdChar D ψ (-1)) (S : Dbl2 ψ σ) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (hν : IsUnitVol ν) (q : GlobalTheory.FinPl.SplitPt D k) (ψk : AddChar k Circle) (hk : (ofSplit q).IsChar ψ ψk) (hc : HasConductor ψk 1) (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] (c : ℕ) (F : Dbl (ofSplit q) S.B S.J S.B' n₀ n₁ n₂ c) (μ₁ μ₂ : kˣ →* ℂˣ) (c₁ : (ofSplit q).IsComp S.χ₁⁻¹ μ₁) (c₂ : (ofSplit q).IsComp S.χ₂⁻¹ μ₂) (o₁ : IsOpen (μ₁.ker : Set kˣ)) (o₂ : IsOpen (μ₂.ker : Set kˣ)) (L₀ : Level (GL (n₀ ⊕ Fin (c + 1 + 1 + 1)) k)) (L₁ : Level (GL (n₁ ⊕ Fin (c + 1 + 1)) k)) (L₂ : Level (GL (n₂ ⊕ Fin (c + 1)) k)) (op₀ : IsOpen (L₀.J : Set (GL (n₀ ⊕ Fin (c + 1 + 1 + 1)) k))) (co₁ : GlobalTheory.IndefiniteRowSideFullHecke.FrameP

theorem **exists_spherical_at** (hψ : IsStdChar D ψ (-1)) (S : Dbl2 ψ σ) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {ψk : AddChar k Circle} (hk : (ofSplit q).IsChar ψ ψk) (hc : HasConductor ψk 1) {μ₁ μ₂ : kˣ →* ℂˣ} (c₁ : (ofSplit q).IsComp S.χ₁⁻¹ μ₁) (c₂ : (ofSplit q).IsComp S.χ₂⁻¹ μ₂) (u₁ : IsUnram μ₁) (u₂ : IsUnram μ₂) : ∃ Q : QStep S, Q.q.v = q.v ∧ IsSpherical Q

theorem **infinite_spherical** (hψ : IsStdChar D ψ (-1)) (S : Dbl2 ψ σ) : {v | ∃ Q : QStep S, Q.q.v = v ∧ IsSpherical Q}.Infinite

theorem **IsSpherical.xi_eq** {S : Dbl2 ψ σ} {Q : QStep S} (h : IsSpherical Q) (γ : Q.A₁) (β : Q.A₂) : GlobalTheory.StepData.xi Q.ψk Q.ν Q.μ₁ Q.Φ₁.1 Q.L₁.J 1 Q.α γ.1 = Model.ind Q.k _ _ ∧ GlobalTheory.StepData.xi Q.ψk Q.ν Q.μ₂ Q.Φ₂.1 Q.L₂.J 1 γ.1 β.1 = Model.ind Q.k _ _

theorem **IsSpherical.constants** {S : Dbl2 ψ σ} {Q : QStep S} (h : IsSpherical Q) : Q.volJ = 1 ∧ (∀ γ, Q.w₁ γ = 1) ∧ ∀ β, Q.w₂ β = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDoubleStep

theorem **exists_dbl** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] (F₀ : GlobalTheory.LocMod.Frame k B.A.rank n₀) (F₁ : GlobalTheory.LocMod.Frame k J.W₀.rank n₁) (F₂ : GlobalTheory.LocMod.Frame k B'.A.rank n₂) (s : k), s ≠ 0 → ∃ F : Dbl p B J B' n₀ n₁ n₂ (2 * σ), F.s = s ∧ F.F₀ = F₀ ∧ F.F₁ = F₁ ∧ F.F₂ = F₂

theorem **five_gram** (e e' : Matrix (Fin 5) (Fin 1) E) (i : Matrix (Fin 5) (Fin 3) E) (he : e = !![0; 0; 0; 1; 1]) (he' : e' = !![0; 0; 0; 1 / 2; -(1 / 2)]) (hi : i = !![1, 0, 0; 0, 1, 0; 0, 0, 1; 0, 0, 0; 0, 0, 0]) : eᴴ * sgnM E ![false, false, false, false, true] * e = 0 ∧ e'ᴴ * sgnM E ![false, false, false, false, true] * e' = 0 ∧ e'ᴴ * sgnM E ![false, false, false, false, true] * e = 1 ∧ iᴴ * sgnM E ![false, false, false, false, true] * i = sgnM E (fun _ => false) ∧ iᴴ * sgnM E ![false, false, false, false, true] * e = 0 ∧ iᴴ * sgnM E ![false, false, false, false, true] * e' = 0

theorem **sub_gram** {N a b : ℕ} (M : Matrix (Fin N) (Fin N) E) (f : Fin a → Fin N) (g : Fin b → Fin N) : ((1 : Matrix (Fin N) (Fin N) E).submatrix id f)ᴴ * M * (1 : Matrix (Fin N) (Fin N) E).submatrix id g = M.submatrix f g

theorem **last_gram** (N : ℕ) (a : Matrix (Fin (N + 1)) (Fin N) E) (b : Matrix (Fin (N + 1)) (Fin 1) E) (ha : a = (1 : Matrix (Fin (N + 1)) (Fin (N + 1)) E).submatrix id Fin.castSucc) (hb : b = (1 : Matrix (Fin (N + 1)) (Fin (N + 1)) E).submatrix id fun _ => Fin.last N) : aᴴ * sgnM E (fun i => decide (i = Fin.last N)) * a = sgnM E (fun _ => false) ∧ bᴴ * sgnM E (fun i => decide (i = Fin.last N)) * b = sgnM E ![true] ∧ bᴴ * sgnM E (fun i => decide (i = Fin.last N)) * a = 0

theorem **coCpt_posSp** (D : D7) (N : ℕ) : CoCpt (posSp D N)

theorem **exists_levels** (D : D7) : ∃ (B : BesLev D (0 + 1)) (J : JacLev D (0 + 1)) (B' : BesLev D 0), 1 = B.A.rank ∧ 3 = J.W₀.rank ∧ 5 = B'.A.rank ∧ Nonempty (Link B B'.L B'.ζ)

theorem **sec_one** : q.sec 1 = q.e

theorem **sec_mul_pr** (a : k) (o : AE D) : q.sec (a * q.pr o) = q.sec a * o

theorem **sec_mem_idAt** (a : k) : q.sec a ∈ idAt D (· = q.v)

theorem **sec_mem_intAt** (a : k) : q.sec a ∈ intAt D (· = q.v) ↔ valuation k a ≤ 1

theorem **pr_le_one** {o : AE D} (ho : o ∈ intAt D (· = q.v)) : valuation k (q.pr o) ≤ 1

theorem **hasConductor** {ψ : AddChar (AF D.host.F) ℂ} (hc : ∀ s ∈ idAt D (· = q.v), (∀ o ∈ intAt D (· = q.v), psiE ψ (s * o) = 1) ↔ s ∈ intAt D (· = q.v)) (h1 : ∀ a ∈ intAt D (· = q.v), incA D D.host.xi0 * a ∈ intAt D (· = q.v)) (h2 : ∀ a ∈ intAt D (· = q.v), incA D D.host.xi0⁻¹ * a ∈ intAt D (· = q.v)) {ψk : AddChar k Circle} (hk : (ofSplit q).IsChar ψ ψk) : HasConductor ψk 1

theorem **eventually_isUnram** (D : D7) (χ : (AE D)ˣ →* Circle) (hχ : Continuous χ) : ∀ᶠ v : GlobalTheory.FinPl.Pl D in cofinite, ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.v = v → ∀ μ, (ofSplit q).IsComp χ μ → IsUnram μ

theorem **isTame_of_isUnram** {k : Type} [Field k] [ValuativeRel k] {μ : kˣ →* ℂˣ} (h : IsUnram μ) : IsTame μ

theorem **exists_good_place** (D : D7) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (χ₁ χ₂ : (AE D)ˣ →* Circle) (h₁ : Continuous χ₁) (h₂ : Continuous χ₂) : ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (p : Place D k) (ψk : AddChar k Circle) (μ₁ μ₂ : kˣ →* ℂˣ), p.IsChar ψ ψk ∧ HasConductor ψk 1 ∧ p.IsComp χ₁ μ₁ ∧ IsUnram μ₁ ∧ p.IsComp χ₂ μ₂ ∧ IsUnram μ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDoubledUnitaryChartSplitPlace

theorem **blkGL_mem_diag** (a : GL m A) : blkGL e (a, a) ∈ Formalism.UG.stabGL (diagSub A e)

theorem **keepsDiag_conj_iff** (a : GL m A) (x : GL ι A) : KeepsDiag e ((blkGL e (a, a))⁻¹ * x * blkGL e (a, a)).1 ↔ KeepsDiag e x.1

theorem **det_leviMat_conj** (a : GL m A) {x : GL ι A} (hx : KeepsDiag e x.1) : (leviMat e ((blkGL e (a, a))⁻¹ * x * blkGL e (a, a)).1).det = (leviMat e x.1).det

theorem **det_quotMat_conj** (a : GL m A) {x : GL ι A} (hx : KeepsDiag e x.1) : (quotMat e ((blkGL e (a, a))⁻¹ * x * blkGL e (a, a)).1).det = (quotMat e x.1).det

theorem **quotMat_map** {B : Type} [CommRing B] (f : A →+* B) (M : Matrix ι ι A) : quotMat e (M.map f) = (quotMat e M).map f

theorem **ordL_eq_neg_log** {K : Type} [Field K] [Valued K (WithZero (Multiplicative ℤ))] (x : K) : ordL x = -WithZero.log (Valued.v x)

theorem **isHermND_dbl** {F E : Type} [Field F] [Field E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) : IsHermND σ (dblGram finSumFinEquiv J)

theorem **exists_dbl_chart** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (Q : Matrix (Fin (n + n)) (Fin (n + n)) E) (hQ : Q = dblGram finSumFinEquiv J) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L], ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDoubledUnitaryChartSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_doubled_unitary_chart_split_place** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (Q : Matrix (Fin (n + n)) (Fin (n + n)) E) (hQ : Q = dblGram finSumFinEquiv J) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L], ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDoublingChartInertPlace

theorem **bc_mem_iff_gram** (hw : IsInertAt F E v w) (hc : GlobalTheory.LFun.IsConjAt σ h c) {J : Matrix (Fin m) (Fin m) E} {T : Matrix (Fin m) (Fin m) (w.adicCompletion E)} {a : GL (Fin m) (w.adicCompletion E)} (ha : cT c a.1 * J.map (algebraMap E _) * a.1 = T) (g : GL (Fin m) (Fv F v ⊗[F] E)) : g ∈ HR (Fv F v) σ J ↔ a⁻¹ * bc (toCompl h) g * a ∈ UJ c T

theorem **blkGL_mem_diag** (a : GL n A) : blkGL e (a, a) ∈ stabGL (diagSub A e)

theorem **conj_mem_diag** (a : GL n A) (x : GL ι A) : (blkGL e (a, a))⁻¹ * x * blkGL e (a, a) ∈ stabGL (diagSub A e) ↔ x ∈ stabGL (diagSub A e)

theorem **bc_mem_diag** {f : A →+* B} (hf : Function.Injective f) (x : GL ι A) : bc f x ∈ stabGL (diagSub B e) ↔ x ∈ stabGL (diagSub A e)

theorem **det_leviMat_conj** (a : GL n A) {x : GL ι A} (hx : x ∈ stabGL (diagSub A e)) : (leviMat e ((blkGL e (a, a))⁻¹ * x * blkGL e (a, a)).1).det = (leviMat e x.1).det

theorem **exists_dblChart** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (Q : Matrix (Fin (n + n)) (Fin (n + n)) E) (hQ : Q = dblGram finSumFinEquiv J) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L], ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), IsInertAt F E v w → ∀ (c : w.adicCompletion E →+* w.adicCompletion E) (e : GLoc σ J v ≃* GU c n), IsInertChart σ J L h c e → ∃ f : GLoc σ Q v ≃* GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.GD c n, (∀ g, g ∈ KFin σ Q L v ↔ f g ∈ GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.KD c n) ∧ (∀ y, f (ioH _ σ hQ (y, 1)) = GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.io1 c n (e y)) ∧ (∀ g, g ∈ siegelH (Fv F v) σ Q ↔ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsDoublingChartInertPlaceStatement

theorem **_root_.GlobalTheory.T.exists_doubling_chart_inert_place** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E), IsCMExt F E σ → ∀ (n : ℕ) (J : Matrix (Fin n) (Fin n) E), IsHermND σ J → ∀ (Q : Matrix (Fin (n + n)) (Fin (n + n)) E) (hQ : Q = dblGram finSumFinEquiv J) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L], ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), IsInertAt F E v w → ∀ (c : w.adicCompletion E →+* w.adicCompletion E) (e : GLoc σ J v ≃* GU c n), IsInertChart σ J L h c e → ∃ f : GLoc σ Q v ≃* GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.GD c n, (∀ g, g ∈ KFin σ Q L v ↔ f g ∈ GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.KD c n) ∧ (∀ y, f (ioH _ σ hQ (y, 1)) = GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.io1 c n (e y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsEqAdicCompletionIntegers

theorem **exists_eq_of_henselian** {R : Type*} [CommRing R] [IsLocalRing R] [HenselianRing R (maximalIdeal R)] [Finite (ResidueField R)] (t n u : R) (hD : IsUnit (t ^ 2 - 4 * n)) (hu : IsUnit u) : ∃ r s : R, r ^ 2 + t * r * s + n * s ^ 2 = u

theorem **exists_eq_adicCompletionIntegers** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (t n u : v.adicCompletionIntegers F), IsUnit (t ^ 2 - 4 * n) → IsUnit u → ∃ r s : v.adicCompletionIntegers F, r ^ 2 + t * r * s + n * s ^ 2 = u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsEqAdicCompletionIntegersStatement

theorem **_root_.GlobalTheory.T.exists_binary_quadratic_eq_unit_of_disc_isUnit** : ∀ (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (t n u : v.adicCompletionIntegers F), IsUnit (t ^ 2 - 4 * n) → IsUnit u → ∃ r s : v.adicCompletionIntegers F, r ^ 2 + t * r * s + n * s ^ 2 = u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsEqSumIndicator

theorem **dc_subset** {G : Type} [Group G] (R K : Subgroup G) {s t x : G} (hs : x ∈ dc R K s) (ht : x ∈ dc R K t) : dc R K s ⊆ dc R K t

theorem **exists_eq_sum_indicator** {G : Type} [Group G] {M : Type} [AddCommMonoid M] (R K : Subgroup G) (T : Finset G) (φ : G → M) (hR : ∀ r ∈ R, ∀ x : G, φ (r * x) = φ x) (hK : ∀ κ ∈ K, ∀ x : G, φ (x * κ) = φ x) (hT : ∀ x : G, φ x ≠ 0 → ∃ t ∈ T, x ∈ dc R K t) : ∃ c : G → M, (∀ t : G, c t = 0 ∨ c t = φ t) ∧ ∀ x : G, φ x = ∑ t ∈ T, Set.indicator (dc R K t) (fun _ => c t) x

theorem **main** : (let d1 : {G : Type} → [inst : Group G] → Subgroup G → Subgroup G → G → Set G

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsEvalAtOfMem

theorem **qdet_map** (f : A →ₐ[ℂ] B) (E : Fin n → Fin n → A) (u : ℂ) : Formalism.REAL.qdet (fun a b => f (E a b)) u = f (Formalism.REAL.qdet E u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsExtGeometryTupleOfRepresentatives

theorem **corner_mul** (g h : GL (n ⊕ Fin 0) k) : (g * h).val.toBlocks₁₁ = g.val.toBlocks₁₁ * h.val.toBlocks₁₁

theorem **corner_one** : (1 : GL (n ⊕ Fin 0) k).val.toBlocks₁₁ = 1

theorem **mulVec_mem_box** {A : Matrix n n k} (hA : IntM A) {c : kˣ} {y : n → k} (hy : y ∈ box n c) : A *ᵥ y ∈ box n c

theorem **sameOrb_equivalence** (J : Subgroup (GL (n ⊕ Fin 0) k)) (hJ : J ≤ KO k _) (c : kˣ) : Equivalence (SameOrb J (box n c))

theorem **stabLat_one** {Y : Type} [Fintype Y] [DecidableEq Y] : stabLat (1 : GL Y k) = KO k Y

theorem **twist_lev_le** (δ : kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → kˣ →* ℂˣ) (j c : ℕ) (h : j + c = s + 1) : ((twistChain s t κ δ hδ η).lev j c h).J ≤ KO k _

theorem **twist_lev_zero** (δ : kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → kˣ →* ℂˣ) (c : ℕ) (h : 0 + c = s + 1) : ((twistChain s t κ δ hδ η).lev 0 c h).J = KO k _ ∧ ∀ x, ((twistChain s t κ δ hδ η).lev 0 c h).lam x = 1

theorem **piA_finite** (hfin : ∀ j d h, (A j d h).Finite) (j : ℕ) : Finite (PiA G A j)

theorem **chT_finite** (hfin : ∀ j d h, (A j d h).Finite) (j : ℕ) : Finite (ChT G δ hδ η A M j)

theorem **mT_spec** (j : ℕ) : (mT G j).IsHaarMeasure ∧ mT G j (KO G.k (XIx (Fin D.R.k) (Fin D.R.t) j) : Set _) = 1

theorem **combOf_spec** (hstd : IsStdChar D c.ψ (-1)) {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame G.k S.rank Y') (FT : Frame G.k T.rank Y) (Ξ : Matrix Y Y' G.k → ℂ) : (ofSplit G.q).IsCombAt c.ψ FS FT (combOf G hstd FS FT Ξ) Ξ

theorem **repT_mem** {D : D7} {c : Comm D} {v : VCom c} {B : GlobalTheory.Fam.Base c} {PP : Formalism.L2.DIV.Pl D.host.E} (G : QGeo v B PP) (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ) (A : RepFam G.k G.s (XIx (Fin D.R.k) (Fin D.R.t))) (M : ℕ) (j d : ℕ) (h : j + d = G.s) (α : ChT G δ hδ η A M j) : repT G δ hδ η A M j d h α ∈ A j d h

theorem **exists_qExt** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle), c.Is → ∀ (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ), (∀ j, IsUnram (η j)) → (∀ j, j ≤ G.s + 1 → (ofSplit G.q).IsComp (χ j)⁻¹ (η (j + 1) * δ)) → ∀ A, (G.T δ hδ η).IsRepSystem A G.ψk G.ν → ∀ M : ℕ, (G.T δ hδ η).Adapted A (box _ (G.ϖ ^ M)⁻¹) (box _ (G.ϖ ^ M)) → ∃ E : QExt G χ, E.δ = δ ∧ E.η = η ∧ E.A = A ∧ E.M = M

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsExtGeometryTupleOfRepresentativesStatement

theorem **_root_.GlobalTheory.T.exists_extGeometry_tuple_of_representatives** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle), c.Is → ∀ (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ), (∀ j, IsUnram (η j)) → (∀ j, j ≤ G.s + 1 → (ofSplit G.q).IsComp (χ j)⁻¹ (η (j + 1) * δ)) → ∀ A, (G.T δ hδ η).IsRepSystem A G.ψk G.ν → ∀ M : ℕ, (G.T δ hδ η).Adapted A (box _ (G.ϖ ^ M)⁻¹) (box _ (G.ϖ ^ M)) → ∃ E : QExt G χ, E.δ = δ ∧ E.η = η ∧ E.A = A ∧ E.M = M

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFactorThroughFlatMapFourierJacobiColumn

theorem **flCol_integrand_integrable** (ψ : AddChar k Circle) (hψ : Continuous ψ) (ν : Measure k) (hν : ν.IsAddHaarMeasure) (p : ℕ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (hΦ : Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (x : Matrix n (n' ⊕ Unit) k) : Integrable (fun e => Φ (ptCol p x e) * (ψ (-ch (affPt p (0 : Matrix n n' k) e)) : ℂ)) (Measure.pi fun _ => ν)

theorem **flCol_add_of_SB** (ψ : AddChar k Circle) (hψ : Continuous ψ) (ν : Measure k) (hν : ν.IsAddHaarMeasure) (p : ℕ) (Φ₁ Φ₂ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (h₁ : Φ₁ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (h₂ : Φ₂ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) : flCol ψ ν p (Φ₁ + Φ₂) = flCol ψ ν p Φ₁ + flCol ψ ν p Φ₂

theorem **flCol_smul** (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (a : ℂ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) : flCol ψ ν p (a • Φ) = a • flCol ψ ν p Φ

theorem **flCol_factor** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c + 1 → ∀ (M : Type) [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), (∀ Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), (∀ x, GlobalTheory.LocFlat.flCol ψ ν p Φ.1 x = 0) → L Φ = 0) → ∃ L₀ : SB (Matrix n (n' ⊕ Unit) k) →ₗ[ℂ] M, ∀ Φ Ψ, Ψ.1 = GlobalTheory.LocFlat.flCol ψ ν p Φ.1 → L Φ = L₀ Ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFactorThroughFlatMapFourierJacobiColumnStatement

theorem **_root_.GlobalTheory.T.exists_factor_through_flatMapFourierJacobiColumn** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k),

Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c + 1 → ∀ (M : Type) [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), (∀ Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), (∀ x, GlobalTheory.LocFlat.flCol ψ ν p Φ.1 x = 0) → L Φ = 0) → ∃ L₀ : SB (Matrix n (n' ⊕ Unit) k) →ₗ[ℂ] M, ∀ Φ Ψ, Ψ.1 = GlobalTheory.LocFlat.flCol ψ ν p Φ.1 → L Φ = L₀ Ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinPartOneMulInvMemHeckeSplitCongruence

theorem **finite_mem** {d : 𝒪 E} (hd : d ≠ 0) : {Q : HeightOneSpectrum (𝒪 E) | d ∈ Q.asIdeal}.Finite

theorem **exists_avoid** (c : 𝒪 E ≃+* 𝒪 E) (hc : ∀ x, c (c x) = x) (δ : 𝒪 E) (hδ : c δ ≠ δ) (M : ℕ) (hM : 0 < M) (y₀ : 𝒪 E) : ∃ t : 𝒪 E, y₀ + M * t ≠ 0 ∧ ∀ Q : HeightOneSpectrum (𝒪 E), (∀ x ∈ Q.asIdeal, c x ∈ Q.asIdeal) → (M : 𝒪 E) ∉ Q.asIdeal → y₀ + M * t ∉ Q.asIdeal

theorem **local_close** {K Γ : Type*} [Field K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) {s y n m : K} (hs : s ≠ 0) (hm : v m ≤ v s) (hm1 : v m ≤ 1) (hlt : v (n * m) < 1) (hy : v (y - s) ≤ v (n * m * m)) : v (s * y⁻¹) = 1 ∧ v (s * y⁻¹ - 1) ≤ v n

theorem **local_unit** {K Γ : Type*} [Field K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) {s y n m : K} (hs : v s ≤ 1) (hm : v m ≤ v s) (hm1 : v m ≤ 1) (hn : v n ≤ 1) (hlt : ¬ v (n * m) < 1) (hy : v y = 1) : v (s * y⁻¹) = 1 ∧ v (s * y⁻¹ - 1) ≤ v n

theorem **val_int** (Q : HeightOneSpectrum (𝒪 E)) (x : 𝒪 E) : Valued.v (algebraMap E (Q.adicCompletion E) (x : E)) < 1 ↔ x ∈ Q.asIdeal

theorem **val_int_le** (Q : HeightOneSpectrum (𝒪 E)) (x : 𝒪 E) : Valued.v (algebraMap E (Q.adicCompletion E) (x : E)) ≤ 1

theorem **val_nat_le** (Q : HeightOneSpectrum (𝒪 E)) (n : ℕ) : Valued.v (n : Q.adicCompletion E) ≤ 1

theorem **exists_den** (s : FiniteAdeleRing (𝒪 E) E) : ∃ m : ℕ, 0 < m ∧ ∀ Q : HeightOneSpectrum (𝒪 E), Valued.v ((m : Q.adicCompletion E) * s Q) ≤ 1

theorem **approx_int** (c : 𝒪 E ≃+* 𝒪 E) (hc : ∀ x, c (c x) = x) (δ : 𝒪 E) (hδ : c δ ≠ δ) (N : ℕ) (hN : 0 < N) (s s' : FiniteAdeleRing (𝒪 E) E) (h : s * s' = 1) (hs : ∀ Q, Valued.v (s Q) ≤ 1) : ∃ y : E, y ≠ 0 ∧ ∃ B : Set (HeightOneSpectrum (𝒪 E)), B.Finite ∧ (∀ Q ∈ B, ¬ ∀ x ∈ Q.asIdeal, c x ∈ Q.asIdeal) ∧ ∀ Q ∉ B, Valued.v (s Q * (algebraMap E (Q.adicCompletion E) y)⁻¹) = 1 ∧ Valued.v (s Q * (algebraMap E (Q.adicCompletion E) y)⁻¹ - 1) ≤ Valued.v (N : Q.adicCompletion E)

theorem **ev_sub** (a b : AE D) (Q : HeightOneSpectrum (𝒪 D.host.E)) : Φ (a - b) Q = Φ a Q - Φ b Q

theorem **ev_one** (Q : HeightOneSpectrum (𝒪 D.host.E)) : Φ 1 Q = 1

theorem **ev_inc** (hΦ : Comp D Φ) (y : D.host.E) (Q : HeightOneSpectrum (𝒪 D.host.E)) : Φ (incA D y) Q = algebraMap D.host.E (Q.adicCompletion D.host.E) y

theorem **isSplitIdem_of_delta** (hΦ : Comp D Φ) {P : HeightOneSpectrum (𝒪 D.host.E)} (hc : cP P ≠ P) {e : AE D} (he : IsDelta Φ P e) : IsSplitIdem D e

theorem **mul_part_inv** {A : Type} [CommRing A] {e : A} (he : e * e = e) (x : Aˣ) : ((x * (part he x)⁻¹ : Aˣ) : A) = e + (1 - e) * x

theorem **mem_modSet** (hΦ : Comp D Φ) {N : ℕ} (hN : 0 < N) (x : (AE D)ˣ) (h0 : infPart D (x : AE D) = 1) (h : ∀ Q, Valued.v (Φ x Q) = 1 ∧ Valued.v (Φ x Q - 1) ≤ Valued.v (N : Q.adicCompletion D.host.E)) : x ∈ modSet D N

theorem **mem_grpH_of_good** (hΦ : Comp D Φ) {N : ℕ} (hN : 0 < N) {B : Set (HeightOneSpectrum (𝒪 D.host.E))} (hB : B.Finite) : (∀ P ∈ B, cP P ≠ P) → ∀ x : (AE D)ˣ, infPart D (x : AE D) = 1 → (∀ Q ∉ B, Valued.v (Φ x Q) = 1 ∧ Valued.v (Φ x Q - 1) ≤ Valued.v (N : Q.adicCompletion D.host.E)) → x ∈ grpH D N

theorem **exists_fin_idem** (hΦ : Comp D Φ) : ∃ f : AE D, f * f = f ∧ infPart D f = 0 ∧ finPart D f = 1 ∧ Φ f = 1

theorem **map_mix_zero** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) {f : A} (h : φ f = 0) (x : A) : φ (f * x + (1 - f)) = 1

theorem **map_mix_one** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) {f : A} (h : φ f = 1) (x : A) : φ (f * x + (1 - f)) = φ x ∧ φ (f + (1 - f) * x) = 1

theorem **exists_of_good** (hΦ : Comp D Φ) {N : ℕ} (hN : 0 < N) {B : Set (HeightOneSpectrum (𝒪 D.host.E))} (hB : B.Finite) (hB1 : ∀ P ∈ B, cP P ≠ P) (x : (AE D)ˣ) (h : ∀ Q ∉ B, Valued.v (Φ x Q) = 1 ∧ Valued.v (Φ x Q - 1) ≤ Valued.v (N : Q.adicCompletion D.host.E)) : ∃ z : (AE D)ˣ, finPart D (z : AE D) = 1 ∧ x * z⁻¹ ∈ grpH D N

theorem **exists_not_fixed** (D : D7) : ∃ δ : 𝒪 D.host.E, cO D δ ≠ δ

theorem **mem_of_mul_mem** {G : Type} [CommGroup G] {H : Subgroup G} {a b x z : G} (ha : a ∈ H) (hb : b ∈ H) (h : a * x * b * z ∈ H) : x * z ∈ H

theorem **exists_finPart_eq_one_mul_inv_mem** : ∀ (D : D7) (N : ℕ), 0 < N → ∀ x : (AE D)ˣ, ∃ z : (AE D)ˣ, finPart D (z : AE D) = 1 ∧ x * z⁻¹ ∈ GlobalTheory.HeckeSplitAll.grpH D N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinPartOneMulInvMemHeckeSplitCongruenceStatement

theorem **_root_.GlobalTheory.T.exists_finPartOne_mul_inv_mem_heckeSplitCongruence** : ∀ (D : D7) (N : ℕ), 0 < N → ∀ x : (AE D)ˣ, ∃ z : (AE D)ˣ, finPart D (z : AE D) = 1 ∧ x * z⁻¹ ∈ GlobalTheory.HeckeSplitAll.grpH D N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteAdditiveCharNeOneAtEachPrime

theorem **trace_eq_tau** (I : AutInst) (x : I.E) : Algebra.trace ℚ I.E x = Algebra.trace ℚ I.F (PI.ScPsiExists.UnitaryGroupMassFormula.tau I.cm.fixed x)

theorem **exists_away** {E : Type} [Field E] [NumberField E] (g : 𝒪 E) (hg : g ≠ 0) (Q : HeightOneSpectrum (𝒪 E)) : ∃ j : 𝒪 E, j ∉ Q.asIdeal ∧ ∀ Q' : HeightOneSpectrum (𝒪 E), g ∈ Q'.asIdeal → Q' ≠ Q → j ∈ Q'.asIdeal

theorem **core** (I : AutInst) (psiEf : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F → ℂ) (hrat : ∀ x : I.E, psiEf (Formalism.UG.inclE (F

theorem **exists_psi_ne_one** : ∀ (D : D7) (psiEf : (instOf D).E ⊗[(instOf D).F] Formalism.MainNotions.S.Af (instOf D).F → ℂ), (∀ x y, psiEf (x + y) = psiEf x * psiEf y) → (∀ x ∈ IntAf (instOf D).F (instOf D).E, psiEf x = 1) → (∀ x : (instOf D).E, psiEf (Formalism.UG.inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteAdditiveCharNeOneAtEachPrimeStatement

theorem **_root_.GlobalTheory.T.exists_finiteAdditiveChar_ne_one_at_each_prime** : ∀ (D : D7) (psiEf : (instOf D).E ⊗[(instOf D).F] Formalism.MainNotions.S.Af (instOf D).F → ℂ), (∀ x y, psiEf (x + y) = psiEf x * psiEf y) → (∀ x ∈ IntAf (instOf D).F (instOf D).E, psiEf x = 1) → (∀ x : (instOf D).E, psiEf (Formalism.UG.inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteAdeleProjectionIntegralReduction

theorem **pr_eq_zero_of_toEAf** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (a : AE D) (ha : toEAf D a = 0) : q.pr a = 0

theorem **prF_toEAf** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (a : AE D) : q.prF he (toEAf D a) = q.pr a

theorem **pr_mem_iff** (q : SplitPt D k) (a : AE D) : q.pr a ∈ 𝒪[k] ↔ toEAf D q.e ∗ toEAf D a ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **mem_iff_val_lt** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (x : 𝒪 D.host.E) : x ∈ PP.asIdeal ↔ valuation k (q.prF he (Formalism.L2.DIV.oToS D.host.F D.host.E x : EAf D)) < 1

theorem **val_lt_of_mem_idS** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) {s : Formalism.L2.Herm.intAf D.host.F D.host.E} (hs : s ∈ Formalism.L2.DIV.idS D.host.F PP.asIdeal) : valuation k (q.prF he s.1) < 1

theorem **resS_eq_zero_of_mem** {PP : Formalism.L2.DIV.Pl D.host.E} {s : Formalism.L2.Herm.intAf D.host.F D.host.E} (hs : s ∈ Formalism.L2.DIV.idS D.host.F PP.asIdeal) : Formalism.L2.DIV.resS D.host.F PP s = 0

theorem **prF_mem_iff** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (z : EAf D) : q.prF he z ∈ 𝒪[k] ↔ toEAf D q.e ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **resS_eq_zero_iff** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (s : Formalism.L2.Herm.intAf D.host.F D.host.E) : Formalism.L2.DIV.resS D.host.F PP s = 0 ↔ valuation k (q.prF he s.1) < 1

theorem **mem_intAf_iff** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (z : EAf D) : z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ q.prF he z ∈ 𝒪[k]

theorem **resS_off** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (PP' : Formalism.L2.DIV.Pl D.host.E) (hne : PP' ≠ PP) (z : EAf D) (hz : z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) (hz' : (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) : Formalism.L2.DIV.resS D.host.F PP' ⟨(1 − toEAf D q.e) ∗ z, hz'⟩ = Formalism.L2.DIV.resS D.host.F PP' ⟨z, hz⟩

theorem **exists_lift_isRed** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (PP : Formalism.L2.DIV.Pl D.host.E), Formalism.L2.NTheta.IsIdemAt q.e PP → ∃ π' : EAf D →+∗ k, (∀ a, π' (toEAf D a) = q.pr a) ∧ GlobalTheory.FunObj.Comparison₂.IsRed (Formalism.L2.DIV.resS D.host.F PP) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype) ∧ (∀ z : EAf D, z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ π' z ∈ 𝒪[k]) ∧ ∀ PP' : Formalism.L2.DIV.Pl D.host.E, PP' ≠ PP → ∀ (z : EAf D) (hz : z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) (hz' : (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E), Formalism.L2.DIV.resS D.host.F PP' ⟨(1 − toEAf D q.e) ∗ z, hz'⟩ = Formalism.L2.DIV.resS D.host.F PP' ⟨z, hz⟩

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteAdeleProjectionIntegralReductionStatement

theorem **_root_.GlobalTheory.T.exists_finiteAdele_projection_integral_reduction** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (PP : Formalism.L2.DIV.Pl D.host.E), Formalism.L2.NTheta.IsIdemAt q.e PP → ∃ π' : EAf D →+∗ k, (∀ a, π' (toEAf D a) = q.pr a) ∧ GlobalTheory.FunObj.Comparison₂.IsRed (Formalism.L2.DIV.resS D.host.F PP) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype) ∧ (∀ z : EAf D, z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ π' z ∈ 𝒪[k]) ∧ ∀ PP' : Formalism.L2.DIV.Pl D.host.E, PP' ≠ PP → ∀ (z : EAf D) (hz : z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) (hz' : (1 − toEAf D q.e) ∗ z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E), Formalism.L2.DIV.resS D.host.F PP' ⟨(1 − toEAf D q.e) ∗ z, hz'⟩ = Formalism.L2.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteAdelicConjSelfDualDoublingCompact

theorem **exists_conj_KAd_of_selfDual** : ∀ (D : D7) (A : HSp D) (L L' : Lat A), IsSelfDualC A L → IsSelfDualC A L' → ∃ k ∈ GfC A, ∀ (a : GL (Fin A.rank) (AE D)) (x : GA D (dblSp A)), x ∈ KAd D.sp.cm.sig (dblSp A).gram L' a ↔ (io A (k, k))⁻¹ ∗ x ∗ io A (k, k) ∈ KAd D.sp.cm.sig (dblSp A).gram L a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteAdelicConjSelfDualDoublingCompactStatement

theorem **_root_.GlobalTheory.T.exists_finiteAdelic_conj_selfDual_doublingCompact** : ∀ (D : D7) (A : HSp D) (L L' : Lat A), IsSelfDualC A L → IsSelfDualC A L' → ∃ k ∈ GfC A, ∀ (a : GL (Fin A.rank) (AE D)) (x : GA D (dblSp A)), x ∈ KAd D.sp.cm.sig (dblSp A).gram L' a ↔ (io A (k, k))⁻¹ ∗ x ∗ io A (k, k) ∈ KAd D.sp.cm.sig (dblSp A).gram L a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteComponentUnitaryActingOnFiniteSuppor

theorem **GlobalTheory.GlobFin.cong_entry** {D : D7} {S : HSp D} {I : Ideal (AE D)} {h : GA D S} (hh : h ∈ congGrp S I) (i j : Fin S.rank) : mt S h i j − (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) i j ∈ I

theorem **GlobalTheory.GlobFin.mem_GfC** {D : D7} {S : HSp D} {x : GA D S} (hx : x ∈ congGrp S (idAt D (fun _ => True))) : x ∈ GfC S

theorem **GlobalTheory.GlobFin.fin_rep** : ∀ (D : D7) (S : HSp D) (g : GA D S), ∃ x ∈ GfC S, ∀ (P : Pl D → Prop) (y : Fin S.rank → AE D), (∀ i, y i ∈ idAt D P) → mt S g ∗ᵥ y = mt S x ∗ᵥ y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteComponentUnitaryActingOnFiniteSupporStatement

theorem **_root_.GlobalTheory.T.exists_finiteComponent_unitary_acting_on_finiteSupported** : ∀ (D : D7) (S : HSp D) (g : GA D S), ∃ x ∈ GfC S, ∀ (P : Pl D → Prop) (y : Fin S.rank → AE D), (∀ i, y i ∈ idAt D P) → mt S g ∗ᵥ y = mt S x ∗ᵥ y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteLocalSupportExchangeDataTwistPlaces

theorem **prod_mul_inv_mem** {G : Type} [Group G] (H : Subgroup G) : ∀ (n : ℕ) (f : Fin n → G) (i : Fin n), (∀ i', i' ≠ i → f i' ∈ H) → (∀ y ∈ H, f i ∗ y = y ∗ f i) → (List.ofFn f).prod ∗ (f i)⁻¹ ∈ H

theorem **mem_support_conv** {G : Type} [Monoid G] : ∀ (n : ℕ) (f : Fin n → (G →₀ ℂ)) (g : G), g ∈ (conv (List.ofFn f)).support → ∃ x : Fin n → G, (∀ i, x i ∈ (f i).support) ∧ g = (List.ofFn x).prod

theorem **GlobalTheory.QSuppTwo.main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t) (F : QFam v t), F.Pins w → ∀ j σ (h : j + 2 ∗ (σ + 1) + 2 = D.R.m), ∃ A : ∀ i, Finset (GL ((F.Q i).X j ⊕ Fin (2 ∗ σ + 2 + 1)) (F.Q i).k), ∀ α, ∀ g ∈ (w.ε j α).support, ∃ a : ∀ i, GL ((F.Q i).X j ⊕ Fin (2 ∗ σ + 2 + 1)) (F.Q i).k, ∀ i, a i ∈ A i ∧ g ∗ ((F.Q i).gl j (2 ∗ σ + 2 + 1) (by have

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteLocalSupportExchangeDataTwistPlacesStatement

theorem **_root_.GlobalTheory.T.exists_finite_local_support_exchangeData_twistPlaces** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t) (F : QFam v t), F.Pins w → ∀ j σ (h : j + 2 ∗ (σ + 1) + 2 = D.R.m), ∃ A : ∀ i, Finset (GL ((F.Q i).X j ⊕ Fin (2 ∗ σ + 2 + 1)) (F.Q i).k), ∀ α, ∀ g ∈ (w.ε j α).support, ∃ a : ∀ i, GL ((F.Q i).X j ⊕ Fin (2 ∗ σ + 2 + 1)) (F.Q i).k, ∀ i, a i ∈ A i ∧ g ∗ ((F.Q i).gl j (2 ∗ σ + 2 + 1) (by have

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteRepsHalfStandardHomModNsmul

theorem **exists_reps_half_stdHom** : ∀ (D : D7) (S T : HSp D) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∀ k : ℕ, 0 < k → ∃ R : Finset (WO S T), ∀ u ∈ stdHom S T, IsHalf E₂ u → ∃ r ∈ R, ∃ v ∈ stdHom S T, IsHalf E₂ v ∧ u = r + k • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFiniteRepsHalfStandardHomModNsmulStatement

theorem **_root_.GlobalTheory.T.exists_finite_reps_half_standardHom_mod_nsmul** : ∀ (D : D7) (S T : HSp D) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∀ k : ℕ, 0 < k → ∃ R : Finset (WO S T), ∀ u ∈ stdHom S T, IsHalf E₂ u → ∃ r ∈ R, ∃ v ∈ stdHom S T, IsHalf E₂ v ∧ u = r + k • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinsetEqSumIndicatorOfPeriodic

theorem **exists_finset_eq_sum_indicator_of_periodic** {X : Type*} [AddCommGroup X] [TopologicalSpace X] [IsTopologicalAddGroup X] {Φ : X → ℂ} (hcs : IsCompact (tsupport Φ)) (V : AddSubgroup X) (hV : IsOpen (V : Set X)) (hper : ∀ x, ∀ y ∈ V, Φ (x + y) = Φ x) : ∃ s : Finset X, ∀ t, Φ t = ∑ a ∈ s, Φ a * (V : Set X).indicator (fun _ => (1 : ℂ)) (t - a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinsetIntegralScaledBoxPiCosetReps

theorem **exists_finset_box_reps** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] (r R : kˣ), valuation k (r : k) ≤ valuation k (R : k) → ∃ S : Finset (n → k), (∀ b ∈ S, b ∈ box n R) ∧ ∀ a ∈ box n R, ∃! b, b ∈ S ∧ a - b ∈ box n r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinsetIntegralScaledBoxPiCosetRepsStatement

theorem **_root_.GlobalTheory.T.exists_finset_integralScaledBoxPi_coset_reps** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] (r R : kˣ), valuation k (r : k) ≤ valuation k (R : k) → ∃ S : Finset (n → k), (∀ b ∈ S, b ∈ box n R) ∧ ∀ a ∈ box n R, ∃! b, b ∈ S ∧ a - b ∈ box n r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinsetNormOne

theorem **norm_alg_inf** (w : InfinitePlace F) (x : F) : ‖algebraMap F w.Completion x‖ = w x

theorem **norm_le_one_iff_int** (v : HeightOneSpectrum (𝒪 F)) (x : v.adicCompletion F) : ‖x‖ ≤ 1 ↔ x ∈ v.adicCompletionIntegers F

theorem **norm_nat_le_one** (v : HeightOneSpectrum (𝒪 F)) (m : ℕ) : ‖(m : v.adicCompletion F)‖ ≤ 1

theorem **norm_int_le_one** (v : HeightOneSpectrum (𝒪 F)) (o : 𝒪 F) : ‖algebraMap F (v.adicCompletion F) (o : F)‖ ≤ 1

theorem **norm_int_eq_one** (v : HeightOneSpectrum (𝒪 F)) (o : 𝒪 F) (ho : o ∉ v.asIdeal) : ‖algebraMap F (v.adicCompletion F) (o : F)‖ = 1

theorem **nat_ne_zero_fin** (v : HeightOneSpectrum (𝒪 F)) {m : ℕ} (hm : m ≠ 0) : (m : v.adicCompletion F) ≠ 0

theorem **norm_natCast_lt_one** (v : HeightOneSpectrum (𝒪 F)) (c : ℕ) (hc : Ideal.absNorm v.asIdeal ∣ c) : ‖(c : v.adicCompletion F)‖ < 1

theorem **exists_finset_norm_one** (x : F) (hx : x ≠ 0) : ∃ S : Finset (HeightOneSpectrum (𝒪 F)), ∀ v ∉ S, ‖algebraMap F (v.adicCompletion F) x‖ = 1

theorem **exists_nat_small** (x : F) (hx : x ≠ 0) : ∃ P : ℕ, P ≠ 0 ∧ ∀ v : HeightOneSpectrum (𝒪 F), ‖algebraMap F (v.adicCompletion F) x‖ ≠ 1 → ‖(P : v.adicCompletion F)‖ < 1

theorem **exists_int_near** (F : Type) [Field F] [NumberField F] : ∃ C : ℝ, 0 < C ∧ ∀ t : InfiniteAdeleRing F, ∃ o : 𝒪 F, ∀ w : InfinitePlace F, ‖algebraMap F w.Completion (o : F) - t w‖ < C

theorem **exists_small_elt** (v₀ : HeightOneSpectrum (𝒪 F)) (η : ℝ) (hη : 0 < η) : ∃ x : F, x ≠ 0 ∧ (∀ w : InfinitePlace F, w x < η) ∧ ∀ v : HeightOneSpectrum (𝒪 F), v ≠ v₀ → ‖algebraMap F (v.adicCompletion F) x‖ ≤ 1

theorem **strong_approx** (v₀ : HeightOneSpectrum (𝒪 F)) (t : InfiniteAdeleRing F) (s : FiniteAdeleRing (𝒪 F) F) (M : ℕ) (hM : M ≠ 0) (ε : ℝ) (hε : 0 < ε) : ∃ c : F, c ≠ 0 ∧ (∀ w : InfinitePlace F, ‖algebraMap F w.Completion c - t w‖ < ε) ∧ ∀ v : HeightOneSpectrum (𝒪 F), v ≠ v₀ → ‖algebraMap F (v.adicCompletion F) c - s v‖ ≤ ‖(M : v.adicCompletion F)‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinsetRepsMaximalIdealMatrixModMul

theorem **inv_mul_mem** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) {x : k} (hx : valuation k x < 1) : (ϖ : k)⁻¹ * x ∈ 𝒪[k]

theorem **exists_int_right_inv** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (s : Finset D) (g : GL D k) (hg : ∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') : ∃ Q : Matrix D D k, GlobalTheory.StepData.IntM Q ∧ g.1 * Q = (ϖ : k) • (1 : Matrix D D k)

theorem **GlobalTheory.LowerHecke.exists_finset_reps_maxLat_mod_mul** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D) (g : GL D k), (∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') → ∃ N : Finset (Matrix D A k), (∀ ν ∈ N, ∀ i a, valuation k (ν i a) < 1) ∧ ∀ ν : Matrix D A k, (∀ i a, valuation k (ν i a) < 1) → ∃! ν', ν' ∈ N ∧ ∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν - ν' = g.1 * μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFinsetRepsMaximalIdealMatrixModMulStatement

theorem **_root_.GlobalTheory.T.exists_finset_reps_maximalIdealMatrix_mod_mul** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D) (g : GL D k), (∃ κ ∈ KO k D, ∃ κ' ∈ KO k D, g = κ * torusElt k D ϖ (indic s) * κ') → ∃ N : Finset (Matrix D A k), (∀ ν ∈ N, ∀ i a, valuation k (ν i a) < 1) ∧ ∀ ν : Matrix D A k, (∀ i a, valuation k (ν i a) < 1) → ∃! ν', ν' ∈ N ∧ ∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν - ν' = g.1 * μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagDualBasisSelfDualLatticeAtPlace

theorem **mul_mem_intAt** {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a * b ∈ intAt D (· = v)

theorem **exists_nat_isIntegral** (z : D.host.E) : ∃ n : ℕ, 0 < n ∧ IsIntegral ℤ ((n : D.host.E) * z)

theorem **exists_nat_isIntegral_fin** {ι : Type} [Fintype ι] (z : ι → D.host.E) : ∃ d : ℕ, 0 < d ∧ ∀ l, IsIntegral ℤ ((d : D.host.E) * z l)

theorem **mul_incA_mem** {ε : AE D} (hεI : ε ∈ intAt D (· = v)) {o : D.host.E} (ho : IsIntegral ℤ o) : ε * incA D o ∈ intAt D (· = v)

theorem **princ_intGrpAt** {ε : AE D} (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) (ht : Tame D v) : Princ ε (intGrpAt D v)

theorem **exists_flag_basis** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (ε : AE D) (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a), Tame D v → ∀ M : AddSubgroup (Fin S.rank → AE D), IsSDAt S v M → ∃ b : Mx D ρ ρ, (∀ i j, j < i → b i j = 0) ∧ (∀ i j, b i j ∈ idAt D (· = v)) ∧ (∀ i, (fun k => (d.jX · b) k i) ∈ M) ∧ ∃ c' : Mx D S.rank ρ, (∀ i, (fun k => c' k i) ∈ M) ∧ adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagDualBasisSelfDualLatticeAtPlaceStatement

theorem **_root_.GlobalTheory.T.exists_flag_dual_basis_selfDualLatticeAtPlace** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (ε : AE D) (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a), Tame D v → ∀ M : AddSubgroup (Fin S.rank → AE D), IsSDAt S v M → ∃ b : Mx D ρ ρ, (∀ i j, j < i → b i j = 0) ∧ (∀ i j, b i j ∈ idAt D (· = v)) ∧ (∀ i, (fun k => (d.jX · b) k i) ∈ M) ∧ ∃ c' : Mx D S.rank ρ, (∀ i, (fun k => c' k i) ∈ M) ∧ adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagDualMatrix

theorem **smul_mem_of_sd** (hmul : ∀ a ∈ O, ∀ b ∈ O, a * b ∈ O) (hM₁ : ∀ y ∈ M, ε • y = y) (hM₂ : ∀ y, ε • y = y → (y ∈ M ↔ ∀ x ∈ M, h y x ∈ O)) {o : R} (ho : o ∈ O) {y : V} (hy : y ∈ M) : o • y ∈ M

theorem **exists_triangular** (hεc : c ε = ε) (hmul : ∀ a ∈ O, ∀ b ∈ O, a * b ∈ O) (hP : Princ ε O) (hM₁ : ∀ y ∈ M, ε • y = y) (hM₂ : ∀ y, ε • y = y → (y ∈ M ↔ ∀ x ∈ M, h y x ∈ O)) (X : (Fin ρ → R) →ₗ[R] V) (f : Fin ρ → V) (hXf : ∀ t k, h (X t) (f k) = t k) (N : ℕ) (hN : 0 < N) (he : ∀ k, X (Pi.single k ((N : R) * ε)) ∈ M) (hf : ∀ k, ((N : R) * ε) • f k ∈ M) (i : Fin ρ) : ∃ (b : Fin ρ → R) (α' : R), (∀ l, i < l → b l = 0) ∧ X b ∈ M ∧ ε * α' = α' ∧ b i * α' = ε ∧ ∀ t : Fin ρ → R, (∀ l, i < l → t l = 0) → X t ∈ M → α' * t i ∈ O

theorem **exists_step** (hcc : ∀ a, c (c a) = a) (hεε : ε * ε = ε) (hεc : c ε = ε) (hεO : ε ∈ O) (hOε : ∀ a ∈ O, ε * a = a) (hmul : ∀ a ∈ O, ∀ b ∈ O, a * b ∈ O) (hcO : ∀ a ∈ O, c a ∈ O) (hP : Princ ε O) (hM₁ : ∀ y ∈ M, ε • y = y) (hM₂ : ∀ y, ε • y = y → (y ∈ M ↔ ∀ x ∈ M, h y x ∈ O)) (X : (Fin ρ → R) →ₗ[R] V) (f : Fin ρ → V) (hXf : ∀ t k, h (X t) (f k) = t k) (N : ℕ) (hN : 0 < N) (he : ∀ k, X (Pi.single k ((N : R) * ε)) ∈ M) (hf : ∀ k, ((N : R) * ε) • f k ∈ M) (b : Fin ρ → Fin ρ → R) (hb₁ : ∀ i l, i < l → b i l = 0) (hb₂ : ∀ i, X (b i) ∈ M) (j : Fin ρ) (α' : R) (hα : b j j * α' = ε) (hJ : ∀ t : Fin ρ → R, (∀ l, j < l → t l = 0) → X t ∈ M → α' * t j ∈ O) (ih : ∀ z : Fin ρ → R, (∀ i, z i ∈ O) → ∃ y ∈ M, ∀ i, i < j → h (X (b i)) y = z i) : ∃ w ∈ M, (∀ i, i < j → h (X (b i)) w = 0) ∧ h (X (b j)) w = ε

theorem **exists_flag_dual** (hcc : ∀ a, c (c a) = a) (hεε : ε * ε = ε) (hεc : c ε = ε) (hεO : ε ∈ O) (hOε : ∀ a ∈ O, ε * a = a) (hmul : ∀ a ∈ O, ∀ b ∈ O, a * b ∈ O) (hcO : ∀ a ∈ O, c a ∈ O) (hP : Princ ε O) (hM₁ : ∀ y ∈ M, ε • y = y) (hM₂ : ∀ y, ε • y = y → (y ∈ M ↔ ∀ x ∈ M, h y x ∈ O)) (X : (Fin ρ → R) →ₗ[R] V) (f : Fin ρ → V) (hXf : ∀ t k, h (X t) (f k) = t k) (N : ℕ) (hN : 0 < N) (he : ∀ k, X (Pi.single k ((N : R) * ε)) ∈ M) (hf : ∀ k, ((N : R) * ε) • f k ∈ M) : ∃ b : Fin ρ → Fin ρ → R, (∀ i l, i < l → b i l = 0) ∧ (∀ i l, ε * b i l = b i l) ∧ (∀ i, X (b i) ∈ M) ∧ ∃ y : Fin ρ → V, (∀ j, y j ∈ M) ∧ ∀ i j, h (X (b i)) (y j) = if i = j then ε else 0

theorem **sesq_col** {n p q : ℕ} (G : Matrix (Fin n) (Fin n) R) (A : Matrix (Fin n) (Fin p) R) (B : Matrix (Fin n) (Fin q) R) (i : Fin p) (j : Fin q) : sesq c G (fun k => A k i) (fun k => B k j) = ((B.map c)ᵀ * (G * A)) j i

theorem **exists_flag_dual_matrix** {n : ℕ} {M : AddSubgroup (Fin n → R)} (G : Matrix (Fin n) (Fin n) R) (hcc : ∀ a, c (c a) = a) (hεε : ε * ε = ε) (hεc : c ε = ε) (hεO : ε ∈ O) (hOε : ∀ a ∈ O, ε * a = a) (hmul : ∀ a ∈ O, ∀ b ∈ O, a * b ∈ O) (hcO : ∀ a ∈ O, c a ∈ O) (hP : Princ ε O) (hM₁ : ∀ y ∈ M, ∀ i, ε * y i = y i) (hM₂ : ∀ y : Fin n → R, (∀ i, ε * y i = y i) → (y ∈ M ↔ ∀ x ∈ M, ∑ i, ∑ j, c (x i) * G i j * y j ∈ O)) (E F : Matrix (Fin n) (Fin ρ) R) (hEF : (F.map c)ᵀ * (G * E) = 1) (N : ℕ) (hN : 0 < N) (he : ∀ k, (fun l => (N : R) * ε * E l k) ∈ M) (hf : ∀ k, (fun l => (N : R) * ε * F l k) ∈ M) : ∃ b : Matrix (Fin ρ) (Fin ρ) R, (∀ i j, j < i → b i j = 0) ∧ (∀ i j, ε * b i j = b i j) ∧ (∀ i, (fun k => (E * b) k i) ∈ M) ∧ ∃ c' : Matrix (Fin n) (Fin ρ) R, (∀ i, (fun k => c' k i) ∈ M) ∧ (c'.map c)ᵀ * (G * (E * b)) = ε • (1 : Matrix (Fin ρ) (Fin ρ) R)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagStabiliserMoveLatticeSplitWitt

theorem **iwasawa_flag_move** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSDAt S v (latV S L (· = v)) → MonoAt d L L₀ v → ∀ h ∈ grpAt S (· = v), ∃ q ∈ grpAt S (· = v), (∃ a : Mx D ρ ρ, (∀ i j, j < i → a i j = 0) ∧ mt S q · d.jX = d.jX · a) ∧ ∃ Λ₀ : AddSubgroup (Fin S₀.rank → AE D), IsSDAt S₀ v Λ₀ ∧ ∀ z : Fin S.rank → AE D, (∀ i, z i ∈ idAt D (· = v)) → (mt S (h⁻¹ * q) *ᵥ z ∈ latV S L (· = v) ↔ (∀ i, colMul d.jX i ((d.pX *ᵥ z) i) ∈ latV S L (· = v)) ∧ (∀ i, colMul d.jY i ((d.pY *ᵥ z) i) ∈ latV S L (· = v)) ∧ d.p0 *ᵥ z ∈ Λ₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagStabiliserMoveLatticeSplitWittStatement

theorem **_root_.GlobalTheory.T.exists_flag_stabiliser_move_lattice_split_witt** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSDAt S v (latV S L (· = v)) → MonoAt d L L₀ v → ∀ h ∈ grpAt S (· = v), ∃ q ∈ grpAt S (· = v), (∃ a : Mx D ρ ρ, (∀ i j, j < i → a i j = 0) ∧ mt S q · d.jX = d.jX · a) ∧ ∃ Λ₀ : AddSubgroup (Fin S₀.rank → AE D), IsSDAt S₀ v Λ₀ ∧ ∀ z : Fin S.rank → AE D, (∀ i, z i ∈ idAt D (· = v)) → (mt S (h⁻¹ * q) *ᵥ z ∈ latV S L (· = v) ↔ (∀ i, colMul d.jX i ((d.pX *ᵥ z) i) ∈ latV S L (· = v)) ∧ (∀ i, colMul d.jY i ((d.pY *ᵥ z) i) ∈ latV S L (· = v)) ∧ d.p0 *ᵥ z ∈ Λ₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagUnipotentCoordinateMeasurableEquiv

theorem **coordFlip_apply** (D : D7) (ρ m : ℕ) (p : skewA D ρ × (Mx D ρ m × UX D ρ)) : coordFlip D ρ m p = (p.2.2⁻¹, (-p.2.1, -p.1))

theorem **flagCoord_measEquiv** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), ∃ e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d, ∀ p, (e p : GA D V) = d.nC p.1 * (d.nB p.2.1 * d.levi p.2.2.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFlagUnipotentCoordinateMeasurableEquivStatement

theorem **_root_.GlobalTheory.T.exists_flagUnipotent_coordinate_measurableEquiv** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), ∃ e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d, ∀ p, (e p : GA D V) = d.nC p.1 * (d.nB p.2.1 * d.levi p.2.2.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFractionalIdealOfHermIdele

theorem **one_tmul_mem** (o : 𝒪 D.host.E) : (1 : Af D.host.F) ⊗ₜ[D.host.F] (o : D.host.E) ∈ intAE D

theorem **one_tmul_mem_iff** (e : D.host.E) : (1 : Af D.host.F) ⊗ₜ[D.host.F] e ∈ intAE D ↔ e ∈ (1 : Submodule (𝒪 D.host.E) D.host.E)

theorem **exists_int_mul** (y : ZF) : ∃ N : 𝒪 D.host.E, N ≠ 0 ∧ ((1 : Af D.host.F) ⊗ₜ[D.host.F] (N : D.host.E)) * y ∈ intAE D

theorem **approx_mul** (z : ZF) {N : 𝒪 D.host.E} (hN : N ≠ 0) : ∃ x : D.host.E, ∃ u ∈ intAE D, z = (1 : Af D.host.F) ⊗ₜ[D.host.F] x + ((1 : Af D.host.F) ⊗ₜ[D.host.F] (N : D.host.E)) * u

theorem **mem_subOf** {y : Af D.host.F ⊗[D.host.F] D.host.E} {d : D.host.E} : d ∈ subOf y ↔ y * ((1 : Af D.host.F) ⊗ₜ[D.host.F] d) ∈ intAE D

theorem **isFractional_subOf** {y y' : Af D.host.F ⊗[D.host.F] D.host.E} (h : y' * y = 1) : IsFractional (𝒪 D.host.E)⁰ (subOf y)

theorem **mem_idealOf** {y y' : Af D.host.F ⊗[D.host.F] D.host.E} (h : y' * y = 1) {d : D.host.E} : d ∈ idealOf h ↔ y * ((1 : Af D.host.F) ⊗ₜ[D.host.F] d) ∈ intAE D

theorem **GlobalTheory.SeeSaw.exists_idealOfIdele** : ∀ (D : D7) (m : (AE D)ˣ), ∃ a : FI D, ∀ x : Pl D, ∀ a ∈ idAt D (· = x), a ∈ idV a x ↔ ((m⁻¹ : (AE D)ˣ) : AE D) ∗ a ∈ intAt D (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFractionalIdealOfHermIdeleStatement

theorem **_root_.GlobalTheory.T.exists_fractionalIdeal_of_hermIdele** : ∀ (D : D7) (m : (AE D)ˣ), ∃ a : FI D, ∀ x : Pl D, ∀ a ∈ idAt D (· = x), a ∈ idV a x ↔ ((m⁻¹ : (AE D)ˣ) : AE D) ∗ a ∈ intAt D (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFrameCol

theorem **integrable_row** (ψk : AddChar k Circle) (hψ : Continuous ψk) (ν : Measure k) [ν.IsAddHaarMeasure] {Φ : Matrix Unit n k → ℂ} (hΦ : Φ ∈ SB (Matrix Unit n k)) (x : Matrix n Unit k) : Integrable (fun η : n → k => Φ (Matrix.of fun _ r => η r) ∗ (ψk (η ⬝ᵥ fun r => x r ()) : ℂ)) (Measure.pi fun _ => ν)

theorem **finv_add** (ψk : AddChar k Circle) (hψ : Continuous ψk) (ν : Measure k) [ν.IsAddHaarMeasure] (C : ℂ) {Φ Ψ : Matrix Unit n k → ℂ} (hΦ : Φ ∈ SB (Matrix Unit n k)) (hΨ : Ψ ∈ SB (Matrix Unit n k)) : finv ψk ν C (Φ + Ψ) = finv ψk ν C Φ + finv ψk ν C Ψ

theorem **finv_smul** (ψk : AddChar k Circle) (ν : Measure k) (C a : ℂ) (Φ : Matrix Unit n k → ℂ) : finv ψk ν C (a • Φ) = a • finv ψk ν C Φ

theorem **finv_trans** (ψk : AddChar k Circle) (ν : Measure k) [ν.IsAddHaarMeasure] (C : ℂ) (x : Matrix Unit n k) {Φ Φ' : Matrix Unit n k → ℂ} (h : ∀ t, Φ' t = Φ (t + x)) (a : Matrix n Unit k) : finv ψk ν C Φ' a = (ψk (-(∑ r, x () r ∗ a r ())) : ℂ) ∗ finv ψk ν C Φ a

theorem **finv_mod** (ψk : AddChar k Circle) (ν : Measure k) (C : ℂ) (t₀ : Matrix n Unit k) {Φ Φ' : Matrix Unit n k → ℂ} (h : ∀ t, Φ' t = (ψk (∑ r, t () r ∗ t₀ r ()) : ℂ) ∗ Φ t) (a : Matrix Unit k) : finv ψk ν C Φ' a = finv ψk ν C Φ (a + t₀)

theorem **heis_comp** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (L S L'' : HSp D) (κ : WO S L'' → WO L S) (hadd : ∀ w w', κ (w + w') = κ w + κ w') (hs : ∀ w w', symplO L S (κ w) (κ w') = symplO S L'' w w') (u : WO S L'') (f : WO L S → ℂ) : (fun w => heisO ψ L S (κ u) f (κ w)) = heisO ψ S L'' u (fun w => f (κ w))

theorem **exists_frame_col** {k : Type} [Field k] {N : ℕ} (hN : N = 1) (z₀ : Fin N → k) (hz : z₀ ≠ 0) : ∃ F : Frame k N Unit, ∀ r, F.β r () = z₀ r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsFullLatticeOfFg

theorem **exists_fullLattice_of_fg** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] (n : ℕ) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ), (∀ v, IsUnit (X v).det) → ∀ N : Submodule (𝒪 K) (Fin n → K), N.FG → ∃ (j : ℕ) (e : Fin j → Fin n → K) (Y : InfinitePlace K → Matrix (Fin j) (Fin j) ℂ) (L : Submodule (𝒪 K) (Fin j → K)), LinearIndependent K e ∧ (∀ v, IsUnit (Y v).det) ∧ Submodule.IsLattice K L ∧ (∀ x : Fin j → K, x ∈ L ↔ ∑ i, x i • e i ∈ N) ∧ (∀ v, gram Y v (fun i => (Pi.single i 1 : Fin j → K)) = gram X v e) ∧ lcv Y L = lcv X N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGAtAllPlaces

theorem **ev_sum** {D : D7} {Φ : Formalism.L2.Theta.AE D →+* FiniteAdeleRing (𝒪 D.host.E) D.host.E} {ι : Type} (s : Finset ι) (f : ι → Formalism.L2.Theta.AE D) (Q : HeightOneSpectrum (𝒪 D.host.E)) : Φ (∑ i ∈ s, f i) Q = ∑ i ∈ s, Φ (f i) Q

theorem **cP_cP** {D : D7} (P : HeightOneSpectrum (𝒪 D.host.E)) : cP (D

theorem **mul_conj_add_eq_zero** {D : D7} {Pg : Finset (HeightOneSpectrum (𝒪 D.host.E))} {E : Formalism.L2.Theta.AE D} (hE : IsIdemSum E Pg) {P P' : HeightOneSpectrum (𝒪 D.host.E)} (hP : P ∈ Pg) (hP' : P' ∈ Pg) (hne : P ≠ P') {e e' : Formalism.L2.Theta.AE D} (he : Formalism.L2.NTheta.IsIdemAt e P) (he' : Formalism.L2.NTheta.IsIdemAt e' P') : e ∗ (e' + cA D e') = 0

theorem **pr_eq_zero** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {f : Formalism.L2.Theta.AE D} (h : q.e ∗ f = 0) : q.pr f = 0

theorem **map_mt_eq_one** {D : D7} {k : Type} [Field k] (π : Formalism.L2.Theta.AE D →+* k) {e : Formalism.L2.Theta.AE D} (V : HSp D) (g : GA D V) (hs : Formalism.L2.Theta.SuppAt e V g) (h0 : π (e + cA D e) = 0) : (mt V g).map π = 1

theorem **prod_map_single** {M α : Type} [Monoid M] (g : α → M) (a : α) : ∀ l : List α, l.Nodup → a ∈ l → (∀ x ∈ l, x ≠ a → g x = 1) → (l.map g).prod = g a | [], _, ha, _ => by simp at ha | x :: l, hn, ha, h1 => by rw [List.map_cons, List.prod_cons] obtain ⟨hx, hl⟩

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGAtAllPlacesMul

theorem **e_mul_add** (q : SplitPt D k) : q.e ∗ (q.e + cA D q.e) = q.e

theorem **v_eq_under** (q : SplitPt D k) {P : HeightOneSpectrum (𝒪 D.host.E)} (h : Formalism.L2.NTheta.IsIdemAt q.e P) : q.v = P.under (𝒪 D.host.F)

theorem **idem_unique** {P : HeightOneSpectrum (𝒪 D.host.E)} {e e' : Formalism.L2.Theta.AE D} (h : Formalism.L2.NTheta.IsIdemAt e P) (h' : Formalism.L2.NTheta.IsIdemAt e' P) : e = e'

theorem **mem_congGrp** {V : HSp D} {I : Ideal (Formalism.L2.Theta.AE D)} {g : GA D V} : g ∈ congGrp V I ↔ ∀ i j, (mt V g - 1) i j ∈ I

theorem **mem_idAt_of_supp** (q : SplitPt D k) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : P q.v) {a : Formalism.L2.Theta.AE D} (h : (q.e + cA D q.e) ∗ a = a) : a ∈ idAt D P

theorem **mem_grpAt_of_supp** (q : SplitPt D k) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : P q.v) {V : HSp D} {g : GA D V} (hg : SuppAt q.e V g) : g ∈ grpAt V P

theorem **pr_eq_zero_of_off** (q : SplitPt D k) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : P q.v) {a : Formalism.L2.Theta.AE D} (ha : a ∈ idOff D P) : q.pr a = 0

theorem **map_mt_eq_one_of_off** (q : SplitPt D k) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : P q.v) {V : HSp D} {g : GA D V} (hg : g ∈ grpOff V P) : (mt V g).map q.pr = 1

theorem **exists_gAt_all_places_mul** : ∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E), d.c.Is → (∀ PP : d.c.Pg, ∃ (e : Formalism.L2.Theta.AE D) (t : ∀ j, GA D (d.st j).V), Formalism.L2.NTheta.IsIdemAt e PP.1 ∧ (∀ j, SuppAt e (d.st j).V (t j)) ∧ Nonempty (GAt (d

theorem **main** : (∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) τ (G : ∀ PP : d.c.Pg, GAt d τ PP.1), d.c.Is → (∃ P : Formalism.L2.DIV.LevelP d.c.Pg D.R.r ℂ, (∀ g, g ∈ P.H ↔ ∀ PP, ∃ x, (G PP).Red x (g PP)) ∧ ∀ (g : P.H) (x : ∀ PP, (G PP).S.P.J), (∀ PP, (G PP).Red (x PP) (g.1 PP)) → P.chi g = ∏ PP, (G PP).S.P.lam (x PP)) ∧ ∃ cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ, (∀ j, ∀ w ∈ (cG j).support, GlobalTheory.FinPl.MxAt (· ∈ d.c.S) w) ∧ ∀ j, j < D.R.m → ∀ f, IsThetaFunO d.c.ψ _ _ f → comb d.c.ψ _ _ (cG j) f = d.c.Pg.attach.toList.foldr (fun PP f' => comb d.c.ψ _ _ ((G PP).cg j) f') f) ∧ (∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E), d.c.Is → (∀ PP : d.c.Pg, ∃ (t : ∀ j, GA D (d.st j).V) (G : GAt (d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGLEmbeddingUnitaryAdelicSplitPlaceBasis

theorem **cT_add'** (c : A →+* A) (X Y : Matrix m m A) : Formalism.UG.cT c (X + Y) = Formalism.UG.cT c X + Formalism.UG.cT c Y

theorem **cT_smul'** (c : A →+* A) (a : A) (X : Matrix m m A) : Formalism.UG.cT c (a • X) = c a • Formalism.UG.cT c X

theorem **smul_mul_smul'** (a b : A) (P Q : Matrix m m A) : (a • P) * (b • Q) = (a * b) • (P * Q)

theorem **blk_mul** (e f u : A) (hef : e * f = 0) (hue : u * e = 0) (huf : u * f = 0) (huu : u * u = u) {X1 Y1 X2 Y2 M : Matrix m m A} (h1 : e • X1 = X1) (h2 : f • Y1 = Y1) (h3 : e • X2 = X2) (h4 : f • Y2 = Y2) : (X1 + Y1 + u • M) * (X2 + Y2 + u • 1) = X1 * X2 + Y1 * Y2 + u • M

theorem **sec_eq** (pr_sec : ∀ a, pr (sec a) = a) (sec_pr : ∀ x, sec (pr x) = e * x) (a : k) : e * sec a = sec a

theorem **sec_mul** (pr_sec : ∀ a, pr (sec a) = a) (sec_pr : ∀ x, sec (pr x) = e * x) (a b : k) : sec (a * b) = sec a * sec b

theorem **secM_mul** (pr_sec : ∀ a, pr (sec a) = a) (sec_pr : ∀ x, sec (pr x) = e * x) (Z W : Matrix m m k) : (Z * W).map sec = Z.map sec * W.map sec

theorem **secM_one** (pr_sec : ∀ a, pr (sec a) = a) (sec_pr : ∀ x, sec (pr x) = e * x) : (1 : Matrix m m k).map sec = e • (1 : Matrix m m A)

theorem **secM_supp** (pr_sec : ∀ a, pr (sec a) = a) (sec_pr : ∀ x, sec (pr x) = e * x) (Z : Matrix m m k) : e • Z.map sec = Z.map sec

theorem **cT_cT'** (hcc : ∀ x, c (c x) = x) (X : Matrix m m A) : Formalism.UG.cT c (Formalism.UG.cT c X) = X

theorem **cT_three** (hh : Formalism.UG.cT c J = J) (P Q : Matrix m m A) : Formalism.UG.cT c P * J * Formalism.UG.cT c Q = Formalism.UG.cT c (Q * J * P)

theorem **hQ** (G : Matrix m m k) : e • (J * G.map sec * Ji) = J * G.map sec * Ji

theorem **hY** (G : Matrix m m k) : c e • Formalism.UG.cT c (J * G.map sec * Ji) = Formalism.UG.cT c (J * G.map sec * Ji)

theorem **scal** : c e * c e = c e ∧ (1 - e - c e) * e = 0 ∧ (1 - e - c e) * c e = 0 ∧ (1 - e - c e) * (1 - e - c e) = 1 - e - c e

theorem **phi_mul** (hJ : Ji * J = 1) (G Gi G' Gi' : Matrix m m k) : phi c e sec J Ji G Gi * phi c e sec J Ji G' Gi' = phi c e sec J Ji (G * G') (Gi' * Gi)

theorem **phi_one** (hJ : Ji * J = 1) : phi c e sec J Ji 1 1 = 1

theorem **phi_unitary** (hcc : ∀ x, c (c x) = x) (hJ : Ji * J = 1) (hh : Formalism.UG.cT c J = J) (G Gi : Matrix m m k) (hG : Gi * G = 1) : Formalism.UG.cT c (phi c e sec J Ji G Gi) * J * phi c e sec J Ji G Gi = J

theorem **phi_map** (G Gi : Matrix m m k) : (phi c e sec J Ji G Gi).map pr = G

theorem **phi_supp** (G Gi : Matrix m m k) : (e + c e) • (phi c e sec J Ji G Gi - 1) = phi c e sec J Ji G Gi - 1

theorem **cT_JR** (S : HSp D) : Formalism.UG.cT (cA D) (S.gram.map (incA D)) = S.gram.map (incA D)

theorem **JR_inv_mul** (S : HSp D) (h : IsUnit S.gram.det) : (S.gram⁻¹).map (incA D) * S.gram.map (incA D) = 1

theorem **exists_gl** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (b : Matrix (Fin S.rank) n k) (b' : Matrix n (Fin S.rank) k) : b * b' = 1 → b' * b = 1 → ∃ gl : GL n k →* GA D S, (∀ g, SuppAt p.e S (gl g)) ∧ ∀ g, (mt S (gl g)).map p.pr = b * g.val * b'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGLEmbeddingUnitaryAdelicSplitPlaceBasisStatement

theorem **_root_.GlobalTheory.T.exists_GL_embedding_unitaryAdelic_splitPlace_basis** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (b : Matrix (Fin S.rank) n k) (b' : Matrix n (Fin S.rank) k), b * b' = 1 → b' * b = 1 → ∃ gl : GL n k →* GA D S, (∀ g, SuppAt p.e S (gl g)) ∧ ∀ g, (mt S (gl g)).map p.pr = b * g.val * b'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGaussIndicator

theorem **exists_gauss_indicator** (D : D7) (θ : InfFam D) (A B : HSp D) (l : D.what₂.Tau → ℝ) (hl : ∀ τ, l τ ≠ 0) (s : D.what₂.Tau → Matrix (Fin B.rank) (Fin B.rank) ℂ) (hs : ∀ τ, IsCartanInv (gramAt D B τ) (s τ)) : ∃ Φ : schwartzA θ B.rank A.rank, Φ.1 ∈ fockA θ B A.rank l s ∧ Φ.1 0 ≠ 0

theorem **hypotheses_inhabited** : ∀ (D : D7) (A B : HSp D), ∃ (ψ : AddChar (AF D.host.F) ℂ) (χ : (AE D)ˣ →* Circle) (θ : InfFam D) (m : Measure (Mx D B.rank A.rank)) (ℱ : Set (Mx D B.rank A.rank)) (omS : G D A.rank →* Module.End ℂ (schwartzA θ B.rank A.rank)) (H : G D A.rank → ℝ≥0) (l : D.what₂.Tau → ℝ) (s : RealMat B) (Φ : schwartzA θ B.rank A.rank), IsCharAF D.host.F ψ ∧ IsHeckeChar D χ ∧ ResEta D χ B.rank ∧ IsInfFam θ ∧ IsCovolOne (ratMx D B.rank A.rank) m ℱ ∧ IsWeilSch ψ χ θ B m omS ∧ IsHeight H ∧ HasRealPar ψ θ l ∧ (∀ τ, IsCartanInv (gramAt D B τ) (s τ)) ∧ Φ.1 ∈ fockA θ B A.rank l s ∧ Φ.1 0 ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGeometricMeasureVolume

theorem **common_measure** : ∀ (D : D7) (V : HSp D) (T : Type) (K : T → Subgroup (GL (Fin V.rank) (Lev.EAf D))) (good : ∀ ρ : D.what₂.Tau, Measure (UpqT (V.sig (tauE D ρ)).1 (V.sig (tauE D ρ)).2) → Prop), (∀ ρ m, good ρ m → m.IsHaarMeasure) → ∀ (μf : Measure (GfC V)), μf.IsHaarMeasure → ∀ (ν : T → Measure (GA D V)), (∀ t, Lev.IsMeasLevW V (K t) good (ν t)) → ∃ ν₀ : Measure (GA D V), ν₀.IsHaarMeasure ∧ ∀ t, μf {k | Lev.finE V k.1 ∈ K t} ≠ 0 ∧ μf {k | Lev.finE V k.1 ∈ K t} ≠ ⊤ ∧ μf {k | Lev.finE V k.1 ∈ K t} • ν t = ν₀

@[simp] theorem **volOf_μ** {D : D7} {V : HSp D} (μ : Measure (GA D V)) (h : μ.IsHaarMeasure) : (volOf μ h).μ = μ

theorem **measures_common** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (b : ∀ j, GA D (Yc D j)), (∀ j, j ≤ D.R.m → (a.x j).Is) → ∃ m : YMeas y, GeoMeas (pk W) (mkGeo m b) ∧ (commOf (pk W) u (mkGeo m b)).VolLast

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGeometricMeasureVolumeStatement

theorem **_root_.GlobalTheory.T.exists_geometric_measure_volume** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (b : ∀ j, GA D (Yc D j)), (∀ j, j ≤ D.R.m → (a.x j).Is) → ∃ m : YMeas y, GeoMeas (pk W) (mkGeo m b) ∧ (commOf (pk W) u (mkGeo m b)).VolLast

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsGoodCompatThree

theorem **exists_frame_of_inverse** {κ : Type} [Fintype κ] [DecidableEq κ] (e : κ ≃ Fin n) (A : Matrix (Fin n) κ R) (B : Matrix κ (Fin n) R) (ε : κ → Bool) (hA : cT s A * J * A = diagonal fun i => if ε i then 1 else -1) (h2 : B * A = 1) : ∃ (b : GL (Fin n) R) (δ : Fin n →

Bool), (cT s b.1 * J * b.1 = diagonal fun i => if δ i then 1 else -1) ∧ cT s (b⁻¹).1 * (b⁻¹).1 = cT s B * B

theorem **exists_frame_std** (hn : m + 2 * r = n) {X Y : Matrix (Fin n) (Fin r) R} {I : Matrix (Fin n) (Fin m) R} {Q : Matrix (Fin m) (Fin n) R} {G : Matrix (Fin m) (Fin m) R} (c : GL (Fin m) R) (ε : Fin m → Bool) {u : R} (hu : u * u + u * u = 1) (hs : s u = u) (XX : cT s X * J * X = 0) (YY : cT s Y * J * Y = 0) (YX : cT s Y * J * X = 1) (XY : cT s X * J * Y = 1) (IX : cT s I * J * X = 0) (IY : cT s I * J * Y = 0) (XI : cT s X * J * I = 0) (YI : cT s Y * J * I = 0) (II : cT s I * J * I = G) (QX : Q * X = 0) (QY : Q * Y = 0) (QI : Q * I = 1) (hc : cT s c.1 * G * c.1 = diagonal fun i => if ε i then 1 else -1) : ∃ (b : GL (Fin n) R) (δ : Fin n → Bool), (cT s b.1 * J * b.1 = diagonal fun i => if δ i then 1 else -1) ∧ cT s (b⁻¹).1 * (b⁻¹).1 = cT s (cT s Y * J) * (cT s Y * J) + cT s Q * ((cT s (c⁻¹).1 * (c⁻¹).1) * Q) + cT s (cT s X * J) * (cT s X * J)

theorem **map_cT** {A : Type} [CommRing A] {s' : A →+* A} {φ : A →+* R} (h : ∀ x, φ (s' x) = s (φ x)) {k l : ℕ} (M : Matrix (Fin k) (Fin l) A) : (cT s' M).map φ = cT s (M.map φ)

theorem **map_form** {A : Type} [CommRing A] {s' : A →+* A} {φ : A →+* R} (h : ∀ x, φ (s' x) = s (φ x)) (c : GL (Fin m) A) : (cT s' (c⁻¹).1 * (c⁻¹).1).map φ = cT s ((bc φ c)⁻¹).1 * ((bc φ c)⁻¹).1

theorem **map_std** {A : Type} [CommRing A] {s' : A →+* A} {φ : A →+* R} (h : ∀ x, φ (s' x) = s (φ x)) (c : GL (Fin m) A) (P P' : Matrix (Fin r) (Fin n) A) (Q : Matrix (Fin m) (Fin n) A) : (cT s' P * P + cT s' Q * ((cT s' (c⁻¹).1 * (c⁻¹).1) * Q) + cT s' P' * P').map φ = cT s (P.map φ) * P.map φ + cT s (Q.map φ) * ((cT s ((bc φ c)⁻¹).1 * ((bc φ c)⁻¹).1) * Q.map φ) + cT s (P'.map φ) * P'.map φ

theorem **hE_mulVec** {P : Matrix (Fin n) (Fin m) E} {G : Matrix (Fin m) (Fin m) E} {Q : Matrix (Fin m) (Fin n) E} (h : G * Q = cT (σ : E →+* E) P * J) (u : Fin m → E) (y : Fin n → E) : hE σ J (P *ᵥ u) y = hE σ G u (Q *ᵥ y)

theorem **exists_lattice_std** (hJ : J.det ≠ 0) {G : Matrix (Fin m) (Fin m) E} (hG : G.det ≠ 0) {L₀ : Submodule (𝒪 E) (Fin m → E)} (hL : Submodule.IsLattice E L₀) (hd : ∀ v, IsSelfDualAt σ G L₀ v) {X Y : Matrix (Fin n) (Fin r) E} {I : Matrix (Fin n) (Fin m) E} {Q : Matrix (Fin m) (Fin n) E} (XX : cT (σ : E →+* E) X * J * X = 0) (YY : cT (σ : E →+* E) Y * J * Y = 0) (YX : cT (σ : E →+* E) Y * J * X = 1) (XY : cT (σ : E →+* E) X * J * Y = 1) (XI : cT (σ : E →+* E) X * J * I = 0) (YI : cT (σ : E →+* E) Y * J * I = 0) (QX : Q * X = 0) (QY : Q * Y = 0) (QI : Q * I = 1) (hQ : G * Q = cT (σ : E →+* E) I * J) (h1 : X * (cT (σ : E →+* E) Y * J) + I * Q + Y * (cT (σ : E →+* E) X * J) = 1) : ∃ L : Submodule (𝒪 E) (Fin n → E), Submodule.IsLattice E L ∧ (∀ v, IsSelfDualAt σ J L v) ∧ ∀ z, z ∈ L ↔ (cT (σ : E →+* E) Y * J) *ᵥ z ∈ stdLattice E r ∧ Q *ᵥ z ∈ L₀ ∧ (cT (σ : E →+* E) X * J) *ᵥ z ∈ stdLattice E r

theorem **exists_sqrt_half** [IsTotallyReal F] (w : InfinitePlace F) : ∃ u : w.Completion ⊗[F] E, u * u + u * u = 1 ∧ sigR _ σ u = u

theorem **exists_arch_frame_std** [IsTotallyReal F] (hn : m + 2 * r = n) {G : Matrix (Fin m) (Fin m) E} {X Y : Matrix (Fin n) (Fin r) E} {I : Matrix (Fin n) (Fin m) E} {Q : Matrix (Fin m) (Fin n) E} (XX : cT (σ : E →+* E) X * J * X = 0) (YY : cT (σ : E →+* E) Y * J * Y = 0) (YX : cT (σ : E →+* E) Y * J * X = 1) (XY : cT (σ : E →+* E) X * J * Y = 1) (IX : cT (σ : E →+* E) I * J * X = 0) (IY : cT (σ : E →+* E) I * J * Y = 0) (XI : cT (σ : E →+* E) X * J * I = 0) (YI : cT (σ : E →+* E) Y * J * I = 0) (II : cT (σ : E →+* E) I * J * I = G) (QX : Q * X = 0) (QY : Q * Y = 0) (QI : Q * I = 1) (w : InfinitePlace F) (c : GL (Fin m) (w.Completion ⊗[F] E)) (ε : Fin m → Bool) (hc : cT (sigR _ σ) c.1 * JR w.Completion G * c.1 = diagonal fun i => if ε i then 1 else -1) {P P' : Matrix (Fin r) (Fin n) (w.Completion ⊗[F] E)} {Q' : Matrix (Fin m) (Fin n) (w.Completion ⊗[F] E)} (hP : P = (cT (σ : E →+* E) Y * J

theorem **compInf_sigR** (w : InfinitePlace F) (z : AF F ⊗[F] E) : compInf F E w (sigR _ σ z) = sigR _ σ (compInf F E w z)

theorem **map_inclR_compInf** {k l : ℕ} (M : Matrix (Fin k) (Fin l) E) (w : InfinitePlace F) : (M.map (inclR (F

theorem **exists_adelic_frame_std** [IsTotallyReal F] (hn : m + 2 * r = n) {G : Matrix (Fin m) (Fin m) E} {X Y : Matrix (Fin n) (Fin r) E} {I : Matrix (Fin n) (Fin m) E} {Q : Matrix (Fin m) (Fin n) E} (XX : cT (σ : E →+* E) X * J * X = 0) (YY : cT (σ : E →+* E) Y * J * Y = 0) (YX : cT (σ : E →+* E) Y * J * X = 1) (XY : cT (σ : E →+* E) X * J * Y = 1) (IX : cT (σ : E →+* E) I * J * X = 0) (IY : cT (σ : E →+* E) I * J * Y = 0) (XI : cT (σ : E →+* E) X * J * I = 0) (YI : cT (σ : E →+* E) Y * J * I = 0) (II : cT (σ : E →+* E) I * J * I = G) (QX : Q * X = 0) (QY : Q * Y = 0) (QI : Q * I = 1) (c : GL (Fin m) (AF F ⊗[F] E)) (hc : IsMajorantFrame σ G c) {P P' : Matrix (Fin r) (Fin n) (AF F ⊗[F] E)} {Q' : Matrix (Fin m) (Fin n) (AF F ⊗[F] E)} (hP : P = (cT (σ : E →+* E) Y * J).map (inclR (F

theorem **exists_good_compat** : ∀ (D : D7) (T T' : HSp D) (t : ℕ) (d : HypDec T T' t) (l' : GlobalTheory.RegSch.Lvl T'), l'.Good → ∃ l : GlobalTheory.RegSch.Lvl T, l.Good ∧ GlobalTheory.RegSch.Compat d l l'

theorem **exists_good_frameLat** (D : D7) (V : HSp D) : ∃ a, Lvl.Good ((frameLat V, a) : Lvl V)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHaarAdelicUnitaryProductRealFinite

theorem **haar_prod** : ∀ (D : D7) (V : HSp D) (μf : Measure (GfC V)), μf.IsHaarMeasure → ∃ (μR : Measure (GA D V)) (μi : Measure (GinfC V)), μR.IsHaarMeasure ∧ ∀ (S : Set (GinfC V)) (E : Set (GfC V)), μR {h | ∃ a ∈ S, ∃ k ∈ E, h = (a : GA D V) * k} = μi S * μf E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHaarAdelicUnitaryProductRealFiniteStatement

theorem **_root_.GlobalTheory.T.exists_haar_adelicUnitary_product_real_finite** : ∀ (D : D7) (V : HSp D) (μf : Measure (GfC V)), μf.IsHaarMeasure → ∃ (μR : Measure (GA D V)) (μi : Measure (GinfC V)), μR.IsHaarMeasure ∧ ∀ (S : Set (GinfC V)) (E : Set (GfC V)), μR {h | ∃ a ∈ S, ∃ k ∈ E, h = (a : GA D V) * k} = μi S * μf E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHaarCovolumeOneSkewHermitianAdelic

theorem **skewPart_mem** (x : Mx D ρ ρ) : skewPart D ρ x ∈ skewA D ρ

theorem **skewPart_of_mem** {c : Mx D ρ ρ} (hc : c ∈ skewA D ρ) : skewPart D ρ c = c

theorem **skewPart_add** (x y : Mx D ρ ρ) : skewPart D ρ (x + y) = skewPart D ρ x + skewPart D ρ y

theorem **skewPart_rat** {x : Mx D ρ ρ} (hx : x ∈ ratMx D ρ ρ) : skewPart D ρ x ∈ ratMx D ρ ρ

theorem **continuous_skewPart** : Continuous (skewPart D ρ)

theorem **isClosed_skewA** : IsClosed (skewA D ρ : Set (Mx D ρ ρ))

theorem **isCoLat_skew** : GlobalTheory.CuspGeneric.IsCoLat ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ))

theorem **exists_haar_covol_one** : ∀ (D : D7) (ρ : ℕ), ∃ (lC : Measure (skewA D ρ)) (𝒸 : Set (skewA D ρ)), lC.IsAddHaarMeasure ∧ SFinite lC ∧ IsAddFundamentalDomain ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ)) 𝒸 lC ∧ lC 𝒸 = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHaarCovolumeOneSkewHermitianAdelicStatement

theorem **_root_.GlobalTheory.T.exists_haar_covolume_one_skewHermitian_adelic** : ∀ (D : D7) (ρ : ℕ), ∃ (lC : Measure (skewA D ρ)) (𝒸 : Set (skewA D ρ)), lC.IsAddHaarMeasure ∧ SFinite lC ∧ IsAddFundamentalDomain ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ)) 𝒸 lC ∧ lC 𝒸 = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHaarFinitePointsFourierJacobiBessel

theorem **haar_exists** : ∀ (D : D7) (V : HSp D), (∃ μ : Measure (GfC V), μ.IsHaarMeasure) ∧ ∀ (V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), (∃ μ : Measure ((jacGrp d).subgroupOf (GfC V)), μ.IsHaarMeasure) ∧ ∀ (A L : HSp D) (o : OrthDec V₀ A L), ∃ μ : Measure ((besGrp d

o).subgroupOf (GfC V)), μ.IsHaarMeasure

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHaarFinitePointsFourierJacobiBesselStatement

theorem **_root_.GlobalTheory.T.exists_haar_finitePoints_fourierJacobi_bessel** : ∀ (D : D7) (V : HSp D), (∃ μ : Measure (GfC V), μ.IsHaarMeasure) ∧ ∀ (V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), (∃ μ : Measure ((jacGrp d).subgroupOf (GfC V)), μ.IsHaarMeasure) ∧ ∀ (A L : HSp D) (o : OrthDec V₀ A L), ∃ μ : Measure ((besGrp d o).subgroupOf (GfC V)), μ.IsHaarMeasure

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHalfData

theorem **compFin_half** (v : HeightOneSpectrum (𝒪 F)) (x c : E) : compFin F E v (pairA F 0 1 ⊗ₜ[F] x + pairA F 1 0 ⊗ₜ[F] c) = (1 : Fv F v) ⊗ₜ[F] x

theorem **compInf_half** (w : InfinitePlace F) (x : E) : compInf F E w (2 * (pairA F 0 1 ⊗ₜ[F] x + pairA F 1 0 ⊗ₜ[F] (1 / 2 : E))) = 1

theorem **sigR_half** (σ : E →ₐ[F] E) (x : E) : sigR (AF F) σ (pairA F 0 1 ⊗ₜ[F] x + pairA F 1 0 ⊗ₜ[F] (1 / 2 : E)) = pairA F 0 1 ⊗ₜ[F] σ x + pairA F 1 0 ⊗ₜ[F] (1 / 2 : E)

theorem **half_add** (x y : E) (h : x + y = 1) : (pairA F 0 1 ⊗ₜ[F] x + pairA F 1 0 ⊗ₜ[F] (1 / 2 : E)) + (pairA F 0 1 ⊗ₜ[F] y + pairA F 1 0 ⊗ₜ[F] (1 / 2 : E)) = (1 : AF F ⊗[F] E)

theorem **exists_halfData** : ∀ D : D7, ∃ t : AE D, HalfData t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHalfRankSub

theorem **fg_of_le_fg** {R M : Type} [CommRing R] [IsNoetherianRing R] [AddCommGroup M] [Module R M] {L X : Submodule R M} (hL : L.FG) (hX : X ≤ L) : X.FG

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHeightDoublingEmbeddingLePowMulHeights

theorem **height_io_le_mul** {D : D7} (V : HSp D) (g₁ g₂ : GA D V) : GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂

theorem **height_io_bound** : ∀ (D : D7) (V : HSp D), ∃ (C : ℝ) (c : ℕ), ∀ g₁ g₂ : GA D V, GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ C * (GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂) ^ c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHeightDoublingEmbeddingLePowMulHeightsStatement

theorem **_root_.GlobalTheory.T.exists_height_doublingEmbedding_le_pow_mul_heights** : ∀ (D : D7) (V : HSp D), ∃ (C : ℝ) (c : ℕ), ∀ g₁ g₂ : GA D V, GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ C * (GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂) ^ c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHeisenbergAverageCompactSymplecticTrivial

theorem **exists_average_of_compact** {X : Type} [AddCommGroup X] [TopologicalSpace X] [IsTopologicalAddGroup X] [LocallyCompactSpace X] [SecondCountableTopology X] [T2Space X] (K : AddSubgroup X) (hK : IsCompact (K : Set X)) (f : X → X → ℂ) (hf : Continuous fun p : X × X => f p.1 p.2) : ∃ g : X → ℂ, Continuous g ∧ (∀ (c : ℂ) (x y : X), (∀ u ∈ K, f x u = c * f y u) → g x = c * g y) ∧ (∀ (c : ℂ) (x y : X), ∀ u₀ ∈ K, (∀ u ∈ K, c * f x u = f y (u₀ + u)) → c * g x = g y) ∧ ∀ (δ : ℝ) (z : ℂ) (w : X), (∀ a ∈ K, ‖f w a − z‖ ≤ δ) → ‖g w − z‖ ≤ δ

theorem **exists_average** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (A T : HSp D) (K : AddSubgroup (WO A T)), IsCompact (K : Set (WO A T)) → (∀ u ∈ K, ∀ u' ∈ K, ψ (symplOH A T u u') = 1) → ∀ (q : WO A T → ℂ), Continuous q → (∀ (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) (u : WO A T), q (ratW T A γ + u) = ψ (−(symplOH A T (ratW T A γ) u)) * q u) → ∃ g : WO A T → ℂ, Continuous g ∧ (∀ (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) (u : WO A T), g (ratW T A γ + u) = ψ (−(symplOH A T (ratW T A γ) u)) * g u) ∧ (∀ u ∈ K, heisO ψ A T u g = g) ∧ ∀ δ : ℝ, (∀ a ∈ K, ∀ w, ‖heisO ψ A T a q w − q w‖ ≤ δ) → ∀ w, ‖g w − q w‖ ≤ δ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHeisenbergAverageCompactSymplecticTrivialStatement

theorem **_root_.GlobalTheory.T.exists_heisenbergAverage_compactSymplecticTrivial** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (A T : HSp D) (K : AddSubgroup (WO A T)), IsCompact (K : Set (WO A T)) → (∀ u ∈ K, ∀ u' ∈ K, ψ (symplOH A T u u') = 1) → ∀ (q : WO A T → ℂ), Continuous q → (∀ (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) (u : WO A T), q (ratW T A γ + u) = ψ (−(symplOH A T (ratW T A γ) u)) * q u) → ∃ g : WO A T → ℂ, Continuous g ∧ (∀ (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) (u : WO A T), g (ratW T A γ + u) = ψ (−(symplOH A T (ratW T A γ) u)) * g u) ∧ (∀ u ∈ K, heisO ψ A T u g = g) ∧ ∀ δ : ℝ, (∀ a ∈ K, ∀ w, ‖heisO ψ A T a q w − q w‖ ≤ δ) → ∀ w, ‖g w − q w‖ ≤ δ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHeisenbergCombinationSplitPlaceDatum

theorem **comb_add** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c c' : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (c + c') f = comb ψ S T c f + comb ψ S T c' f

theorem **comb_smul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (a : ℂ) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (a • c) f = a • comb ψ S T c f

theorem **comb_single** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (y : WO S T) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single y 1) f = heisO ψ S T y f

theorem **heis_comb** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) {e : AE D} {y : WO S T} (hy : IsHalf e y) {c : WO S T →₀ ℂ} (hc : ∀ w ∈ c.support, IsHalf e w) : ∃ c' : WO S T →₀ ℂ, (∀ w ∈ c'.support, IsHalf e w) ∧ ∀ f, comb ψ S T c' f = heisO ψ S T y (comb ψ S T c f)

theorem **trans_mem_SB** {Φ : Matrix Y Y' k → ℂ} (hΦ : Φ ∈ SB (Matrix Y Y' k)) (x : Matrix Y Y' k) : (fun t => Φ (t + x)) ∈ SB (Matrix Y Y' k)

theorem **modul_mem_SB** {ψ : AddChar k Circle} {d : kˣ} (hd : HasConductor ψ d) {Φ : Matrix Y Y' k → ℂ} (hΦ : Φ ∈ SB (Matrix Y Y' k)) (b : Matrix Y Y' k) : (fun t => Inhab.pairing ψ t b * Φ t) ∈ SB (Matrix Y Y' k)

theorem **exists_comb** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ Φ₀ Φ : SB (Matrix Y Y' k), Φ₀ ≠ 0 → ∃ c : WO S T →₀ ℂ, (∀ w ∈ c.support, IsHalf (p.e + cA D p.e) w) ∧ ∀ Tm, p.IsModel ψ FS FT Tm → Tm Φ = comb ψ S T c (Tm Φ₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHeisenbergCombinationSplitPlaceDatumStatement

theorem **_root_.GlobalTheory.T.exists_heisenberg_combination_split_place_datum** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ Φ₀ Φ : SB

(Matrix Y Y' k), Φ₀ ≠ 0 → ∃ c : WO S T →₀ ℂ, (∀ w ∈ c.support, IsHalf (p.e + cA D p.e) w) ∧ ∀ Tm, p.IsModel ψ FS FT Tm → Tm Φ = comb ψ S T c (Tm Φ₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHermIsUnitGeneric

theorem **isUnit_glue** {R ι : Type} [CommRing R] [Fintype ι] {n : ℕ} (e t : ι → R) (A : ι → Matrix (Fin n) (Fin n) R) (h1 : ∑ i, e i = 1) (ho : ∀ i j, i ≠ j → e i * e j = 0) (ht : ∀ i, t i * (A i).det = e i) : IsUnit (∑ i, e i • A i)

theorem **exists_herm_notMem** (σ : E →+* E) (δ : E) (hδ : δ ≠ 0) (hσδ : σ δ = -δ) (ι : E →+* R) (τ : R →+* R) (hτι : ∀ z, τ (ι z) = ι (σ z)) (c d p q : Matrix (Fin n) (Fin n) R) (h : c * p + d * q = 1) (M : Ideal R) (hM : M.IsMaximal) : ∃ x : Matrix (Fin n) (Fin n) E, (∀ i j, σ (x i j) = x j i) ∧ (c * x.map ι + d).det * τ (c * x.map ι + d).det ∉ M

theorem **exists_herm_isUnit_generic** {A : Type} [CommRing A] (σ : E →+* E) (δ : E) (hδ : δ ≠ 0) (hσδ : σ δ = -δ) (ι : E →+* R) (τ : R →+* R) (hτι : ∀ z, τ (ι z) = ι (σ z)) (hττ : ∀ z, τ (τ z) = z) (a : A →+* R) (ha : Function.Injective a) (hτa : ∀ z, τ (a z) = a z) (hfix : ∀ y, τ y = y → ∃ z, a z = y) (k : A) (hk : k * 2 = 1) (hid : ∀ (κ : Type) [Fintype κ] (f s : κ → A), ∑ i, s i * f i = 1 → ∃ e t : κ → A, ∑ i, e i = 1 ∧ (∀ i j, i ≠ j → e i * e j = 0) ∧ ∀ i, t i * f i = e i) (c d p q : Matrix (Fin n) (Fin n) R) (h : c * p + d * q = 1) : ∃ x : Matrix (Fin n) (Fin n) R, (x.map τ)ᵀ = x ∧ IsUnit (c * x + d)

theorem **cA_algebraMap** {D : D7} (a : AF D.host.F) : cA D (algebraMap (AF D.host.F) (AE D) a) = algebraMap (AF D.host.F) (AE D) a

theorem **algebraMap_inj** {D : D7} : Function.Injective (algebraMap (AF D.host.F) (AE D))

theorem **half_mul_two** {F A : Type} [Field F] [CharZero F] [CommRing A] [Algebra F A] : algebraMap F A (1 / 2) * 2 = 1

theorem **exists_herm_isUnit** (D : D7) (n : ℕ) (c d p q : Mx D n n) (h : c * p + d * q = 1) : ∃ x : Mx D n n, GlobalTheory.MixModel.adj x = x ∧ IsUnit (c * x + d)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHermitianDetNeZeroTwo

theorem **hermGenericMatrix_map_eval** {κ : Type} [Field κ] (n : ℕ) (e : κ) (u : Fin n × Fin n → κ) : (hermGenericMatrix n e).map (MvPolynomial.eval u) = Matrix.of fun i j => u (i, j) + u (j, i) + e * (u (i, j) - u (j, i))

theorem **hermGenericMatrix_onto** {κ : Type} [Field κ] [CharZero κ] (n : ℕ) (e : κ) (he : e ≠ 0) (y : Matrix (Fin n) (Fin n) κ) : ∃ u : Fin n × Fin n → κ, (hermGenericMatrix n e).map (MvPolynomial.eval u) = y

theorem **eval_det_affine_matrix** {κ ι : Type} [Field κ] {n : ℕ} (c d : Matrix (Fin n) (Fin n) κ) (G : Matrix (Fin n) (Fin n) (MvPolynomial ι κ)) (u : ι → κ) : MvPolynomial.eval u (c.map C * G + d.map C).det = (c * G.map (MvPolynomial.eval u) + d).det

theorem **exists_hermitian_det_ne_zero_two** : ∀ (E κ : Type) [Field E] [CharZero E] [Field κ] (σ : E →+* E) (φ : E →+* κ) (δ : E), δ ≠ 0 → σ δ = -δ → ∀ (n : ℕ) (c d y c' d' y' : Matrix (Fin n) (Fin n) κ), (c * y + d).det ≠ 0 → (c' * y' + d').det ≠ 0 → ∃ x : Matrix (Fin n) (Fin n) E, x.map σ = x.transpose ∧ (c * x.map φ + d).det ≠ 0 ∧ (c' * x.transpose.map φ + d').det ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHermitianDetNeZeroTwoStatement

theorem **_root_.GlobalTheory.T.exists_hermitian_two_det_ne_zero** : ∀ (E κ : Type) [Field E] [CharZero E] [Field κ] (σ : E →+* E) (φ : E →+* κ) (δ : E), δ ≠ 0 → σ δ = -δ → ∀ (n : ℕ) (c d y c' d' y' : Matrix (Fin n) (Fin n) κ), (c * y + d).det ≠ 0 → (c' * y' + d').det ≠ 0 → ∃ x : Matrix (Fin n) (Fin n) E, x.map σ = x.transpose ∧ (c * x.map φ + d).det ≠ 0 ∧ (c' * x.transpose.map φ + d').det ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHermitianLineVectorPrescribedNorm

theorem **signLine_norm** (D : D7) (c : D.host.E) (p q : ℕ) (h : p + q = 1) : star (fun _ : Fin (signLine D p q).rank => c) ·ᵥ (signLine D p q).gram *ᵥ (fun _ : Fin (signLine D p q).rank => c) = (Examples₂.sgn (decide (p ≤ 0)) : D.host.E) * (c * star c)

theorem **exists_line_vec** (D : D7) (c : D.host.E) (ε : ℤˣ) (hc : c ≠ 0) : ∃ (L : HSp D) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ ∧ L.SigOK ∧ star ζ ·ᵥ L.gram *ᵥ ζ = ((ε : ℤ) : D.host.E) * (c * star c) ∧ ∀ τ, L.sig τ = if ε = 1 then (1, 0) else (0, 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHermitianLineVectorPrescribedNormStatement

theorem **_root_.GlobalTheory.T.exists_hermitianLine_vector_prescribed_norm** : ∀ (D : D7) (c : D.host.E) (ε : ℤˣ), c ≠ 0 → ∃ (L : HSp D) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ ∧ L.SigOK ∧ star ζ ·ᵥ L.gram *ᵥ ζ = ((ε : ℤ) : D.host.E) * (c * star c) ∧ ∀ τ, L.sig τ = if ε = 1 then (1, 0) else (0, 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHomotheticPeriodPairLattice

theorem **homotheticLattice** : ∀ (L : PeriodPair) (c : ℂ), c ≠ 0 → ∃ L' : PeriodPair, (∀ z, z ∈ L'.lattice ↔ c * z ∈ L.lattice) ∧ L'.g₂ = c ^ 4 * L.g₂ ∧ L'.g₃ = c ^ 6 * L.g₃ ∧ (∀ z, L'.weierstrassP z = c ^ 2 * L.weierstrassP (c * z) ∧ L'.derivWeierstrassP z = c ^ 3 * L.derivWeierstrassP (c * z)) ∧ ∀ d, HasCMBy d L → HasCMBy d L'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHomotheticPeriodPairLatticeStatement

theorem **_root_.GlobalTheory.T.exists_homothetic_periodPair_lattice** : ∀ (L : PeriodPair) (c : ℂ), c ≠ 0 → ∃ L' : PeriodPair, (∀ z, z ∈ L'.lattice ↔ c * z ∈ L.lattice) ∧ L'.g₂ = c ^ 4 * L.g₂ ∧ L'.g₃ = c ^ 6 * L.g₃ ∧ (∀ z, L'.weierstrassP z = c ^ 2 * L.weierstrassP (c * z) ∧ L'.derivWeierstrassP z = c ^ 3 * L.derivWeierstrassP (c * z)) ∧ ∀ d, HasCMBy d L → HasCMBy d L'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHondaTypeEvenFrobeniusPowers

theorem **ord_add** {k : ℕ∞} {A B : MvPowerSeries ι K} (hA : k ≤ A.order) (hB : k ≤ B.order) : k ≤ (A + B).order

theorem **ord_mul** {j k : ℕ∞} {A B : MvPowerSeries ι K} (hA : j ≤ A.order) (hB : k ≤ B.order) : j + k ≤ (A * B).order

theorem **ord_pow_sub** {A B : MvPowerSeries ι K} (hA : MvPowerSeries.constantCoeff A = 0) (hB : MvPowerSeries.constantCoeff B = 0) (h : (2 : ℕ∞) ≤ (A - B).order) (n : ℕ) : ((n + 1 : ℕ) : ℕ∞) ≤ (A ^ n - B ^ n).order

theorem **ord_subst_sub** {g : K⟦X⟧} (h0 : constantCoeff g = 0) (h1 : coeff 1 g = 1) {A : MvPowerSeries ι K} (hA : MvPowerSeries.constantCoeff A = 0) : (2 : ℕ∞) ≤ (g.subst A - A).order

theorem **subst_comb** {A : MvPowerSeries ι K} (hA : MvPowerSeries.constantCoeff A = 0) (a : K) (f : K⟦X⟧) (s : Finset ℕ) (c : ℕ → K) (g : ℕ → K⟦X⟧) : (C a * f - ∑ k ∈ s, C (c k) * g k).subst A = MvPowerSeries.C a * f.subst A - ∑ k ∈ s, MvPowerSeries.C (c k) * (g k).subst A

theorem **norm_sub_le_of** {x y : K} {r : ℝ} (hx : ‖x‖ ≤ r) (hy : ‖y‖ ≤ r) : ‖x - y‖ ≤ r

theorem **norm_coeff_subst_sub** {r : ℝ} (hr : 0 ≤ r) {A B : MvPowerSeries ι K} (hA : MvPowerSeries.constantCoeff A = 0) (hB : MvPowerSeries.constantCoeff B = 0) (hAB : (2 : ℕ∞) ≤ (A - B).order) (hI : IntMv (Formalism.G.unitBall K) A) (h : K⟦X⟧) (d : ι →₀ ℕ) (hlow : ∀ m < d.degree, ‖coeff m h‖ ≤ r) : ‖MvPowerSeries.coeff d (h.subst A) - coeff d.degree h * MvPowerSeries.coeff d (B ^ d.degree)‖ ≤ r

theorem **cob_degree** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] (l : ℕ) (f e : K⟦X⟧), IntLog f e → ∀ (h : K⟦X⟧), (∀ m, ‖MvPowerSeries.coeff m (Formalism.L2.Th1.C.be0 f e h (MvPowerSeries.X (0 : Fin 2)) (MvPowerSeries.X (1 : Fin 2)))‖ ≤ ‖(l : K)‖) → ∀ n : ℕ, (∀ m < n, ‖coeff m h‖ ≤ ‖(l : K)‖) → ∀ k : ℕ, 0 < k → k < n → ‖coeff n h * (n.choose k : K)‖ ≤ ‖(l : K)‖

theorem **eig_degree** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] (l : ℕ) (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → ∀ h : K⟦X⟧, CoefLe ‖(l : K)‖ (h.subst (Formalism.L2.Th1.C.mulBy f e w) - C w * h) → ∀ n : ℕ, (∀ m < n, ‖coeff m h‖ ≤ ‖(l : K)‖) → ‖coeff n h * (w ^ n - w)‖ ≤ ‖(l : K)‖

theorem **pow_cong_frob** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ w : K, ‖w‖ ≤ 1 → ∀ i : ℕ, ‖w ^ (l ^ i) - (σ ^ i) w‖ < 1

theorem **coeff_frobOp_ite** : ∀ (K : Type) [NormedField K] (τ : K →+* K) (q i : ℕ) (f : K⟦X⟧), 0 < q → ∀ m : ℕ, coeff m (frobOp τ q i f) = if q ^ i ∣ m then (τ ^ i) (coeff (m / q ^ i) f) else 0

theorem **cob_even_sum** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ (s : Finset ℕ) (c : ℕ → K), (∀ k ∈ s, ‖c k‖ ≤ 1) → ∀ m, ‖MvPowerSeries.coeff m (Formalism.L2.Th1.C.be0 f e (C (l : K) * f - ∑ k ∈ s, C (c k) * frobOp σ l (2 * k + 2) f) (MvPowerSeries.X (0 : Fin 2)) (MvPowerSeries.X (1 : Fin 2)))‖ ≤ ‖(l : K)‖

theorem **eig_even_sum** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → σ (σ w) = w → ∀ (s : Finset ℕ) (c : ℕ → K), (∀ k ∈ s, ‖c k‖ ≤ 1) → CoefLe ‖(l : K)‖ ((C (l : K) * f - ∑ k ∈ s, C (c k) * frobOp σ l (2 * k + 2) f).subst (Formalism.L2.Th1.C.mulBy f e w) - C w * (C (l : K) * f - ∑ k ∈ s, C (c k) * frobOp σ l (2 * k + 2) f))

theorem **mulBy_low_coeff** : ∀ (K : Type) [Field K] (f e : K⟦X⟧), Formalism.L2.Th1.C.IsLogPair f e → ∀ w : K, constantCoeff (Formalism.L2.Th1.C.mulBy f e w) = 0 ∧ coeff 1 (Formalism.L2.Th1.C.mulBy f e w) = w

theorem **exists_triangular** {K : Type} [Field K] (a : ℕ → K) (b : ℕ → ℕ → K) : ∃ c : ℕ → K, ∀ j, c j = a j - ∑ k ∈ Finset.range j, c k * b j k

theorem **exists_even_type** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → σ (σ w) = w → ‖σ w - w‖ = 1 → ∃ c : ℕ → K, (∀ j, ‖c j‖ ≤ 1) ∧ CoefLe ‖(l : K)‖ (C (l : K) * f - hondaSum (σ ^ 2) (l ^ 2) c f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHondaTypeEvenFrobeniusPowersStatement

theorem **_root_.GlobalTheory.T.exists_hondaType_evenFrobeniusPowers** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → σ (σ w) = w → ‖σ w - w‖ = 1 → ∃ c : ℕ → K, (∀ j, ‖c j‖ ≤ 1) ∧ CoefLe ‖(l : K)‖ (C (l : K) * f - hondaSum (σ ^ 2) (l ^ 2) c f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsHypDec

theorem **gram_scaled** {E : Type} [Field E] [StarRing E] {n k l : Type} [Fintype n] (G : Matrix n n E) (A : Matrix n k E) (B : Matrix n l E) (α : k → E) (β : l → E) (x : k → n → E) (y : l → n → E) (hA : ∀ i a, A i a = α a * x a i) (hB : ∀ i b, B i b = β b * y b i) (a : k) (b : l) : (Aᴴ * G * B) a b = star (α a) * β b * (star (x a) ⬝ᵥ (G *ᵥ y b))

theorem **gram_sum** {E : Type} [Field E] [StarRing E] {n k l m : Type} [Fintype n] [Fintype k] [Fintype l] (G : Matrix n n E) (I : Matrix n k E) (J : Matrix n l E) (P : Matrix k m E) (Q : Matrix l m E) (hIJ : Iᴴ * G * J = 0) (hJI : Jᴴ * G * I = 0) : (I * P + J * Q)ᴴ * G * (I * P + J * Q) = Pᴴ * (Iᴴ * G * I) * P + Qᴴ * (Jᴴ * G * J) * Q

theorem **gram_sum_zero** {E : Type} [Field E] [StarRing E] {n k l m p : Type} [Fintype n] [Fintype k] [Fintype l] (G : Matrix n n E) (I : Matrix n k E) (J : Matrix n l E) (P : Matrix k m E) (Q : Matrix l m E) (C : Matrix n p E) (hI : Iᴴ * G * C = 0) (hJ : Jᴴ * G * C = 0) : (I * P + J * Q)ᴴ * G * C = 0

theorem **exists_hypDec** : ∀ (D : D7) (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (F₀ : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F₀) (t lam : D.host.E), t ≠ 0 → lam ≠ 0 → (∀ y y' : (VS D).V, star (GlobalTheory.Core.SBesEx.cY X h y') ⬝ᵥ ((D.sp.Y 0).gram *ᵥ GlobalTheory.Core.SBesEx.cY X h y) = (VS D).hT y' y) → (∀ (i j : Fin (D.sp.X 0).rank) (a : Fin D.R.rS), (VS D).hT (Formalism.L2.SG.ColDict.qFr tr.wt (Fin.cast (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) i)) (Formalism.L2.SG.ColDict.qFr tr.wt (Fin.cast (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) j)) = (D.sp.X 0).gram i j ∧ (VS D).hT (Formalism.L2.SG.ColDict.qFr tr.wt (Fin.cast (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) i)) (X.eX a) = 0 ∧ (VS D).hT (Formalism.L2.SG.ColDict.qFr tr.wt (Fin.cast (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) i)) (X.fY a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIdeleChangeFinitelyManySplitPlaces

theorem **idAt_le_idOff** {x y : Pl D} (hxy : x ≠ y) : idAt D (· = x) ≤ idOff D (· = y)

theorem **mul_eq_zero_of_ne** {x y : Pl D} (hxy : x ≠ y) {a b : AE D} (ha : a ∈ idAt D (· = x)) (hb : b ∈ idAt D (· = y)) : a * b = 0

theorem **exists_change_one** (q : SplitPt D k) (h : D.host.E) (hh : h ≠ 0) (hs : star h = h) (m : (AE D)ˣ) : ∃ m' : (AE D)ˣ, (∀ x, x ≠ q.v → ∀ a ∈ idAt D (· = x), (m' : AE D) * a = (m : AE D) * a ∧ cA D (m' : AE D) * a = cA D (m : AE D) * a) ∧ ∀ a ∈ idAt D (· = q.v), (m' : AE D) * cA D (m' : AE D) * incA D h * a = a

theorem **exists_change_list** {ι : Type} (p : ι → SplitPt D k) (hp : Function.Injective fun i => (p i).v) (h : D.host.E) (hh : h ≠ 0) (hs : star h = h) (m : (AE D)ˣ) (l : List ι) : ∃ m' : (AE D)ˣ, (∀ x, (∀ i ∈ l, x ≠ (p i).v) → ∀ a ∈ idAt D (· = x), (m' : AE D) * a = (m : AE D) * a ∧ cA D (m' : AE D) * a = cA D (m : AE D) * a) ∧ ∀ i ∈ l, ∀ a ∈ idAt D (· = (p i).v), (m' : AE D) * cA D (m' : AE D) * incA D h * a = a

theorem **GlobalTheory.FinPl.exists_idele_change_split** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ h : D.host.E, h ≠ 0 → star h = h → ∀ m : (AE D)ˣ, ∃ m' : (AE D)ˣ, (∀ x, ¬ P x → ∀ a ∈ idAt D (· = x), (m' : AE D) * a = (m : AE D) * a ∧ cA D (m' : AE D) * a = cA D (m : AE D) * a) ∧ ∀ x, P x → ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (m' : AE D) * cA D (m' : AE D) * incA D h * a ∈ intAt D (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIdeleChangeFinitelyManySplitPlacesStatement

theorem **_root_.GlobalTheory.T.exists_idele_change_finitely_many_split_places** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ h : D.host.E, h ≠ 0 → star h = h → ∀ m : (AE D)ˣ, ∃ m' : (AE D)ˣ, (∀ x, ¬ P x → ∀ a ∈ idAt D (· = x), (m' : AE D) * a = (m : AE D) * a ∧ cA D (m' : AE D) * a = cA D (m : AE D) * a) ∧ ∀ x, P x → ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (m' : AE D) * cA D (m' : AE D) * incA D h * a ∈ intAt D (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIdeleTraceZeroNormAdjustRealComponents

private theorem **tmulOne_map** {R S : Type} [CommRing R] [CommRing S] [Algebra F R] [Algebra F S] (f : R →ₐ[F] S) (σ : E →ₐ[F] E) (x : R) : (Algebra.TensorProduct.map f σ).toRingHom (x ⊗ₜ[F] (1 : E)) = f x ⊗ₜ[F] (1 : E)

private theorem **scal_gen** (σ : E →ₐ[F] E) (μ : E) (hμ : μ ≠ 0) (r : ∀ w : InfinitePlace F, w.Completion) (hr : ∀ w, r w ≠ 0) (hN : ∀ w, ((1 : w.Completion) ⊗ₜ[F] (μ * σ μ) : w.Completion ⊗[F] E) = 4 * (r w ⊗ₜ[F] (1 : E)) ^ 4) (u₀ : (AF F ⊗[F] E)ˣ) : ∃ u ε : (AF F ⊗[F] E)ˣ, (ε : AF F ⊗[F] E) * (u : AF F ⊗[F] E) * sigR (AF F) σ (u : AF F ⊗[F] E) = inclR (AF F) μ ∧ (∀ v, compFin F E v u = compFin F E v u₀) ∧ ∀ w, compInf F E w ((ε : AF F ⊗[F] E) * sigR (AF F) σ (ε : AF F ⊗[F] E)) = 4

private theorem **scal_pos** (D : D7) (μ : D.host.E) (hμ : μ ≠ 0) (w : InfinitePlace D.host.F) : 0 < InfinitePlace.Completion.ringEquivRealOfIsReal (NumberField.IsTotallyReal.isReal w) (algebraMap D.host.F w.Completion (μ * star μ, (IsCMField.complexConj_eq_self_iff D.host.E _).1 (by change star (μ * star μ) = μ * star μ; rw [star_mul, star_star])))

private theorem **scal_root** (F E : Type) [Field F] [Field E] [Algebra F E] (w : InfinitePlace F) (hw : w.IsReal) (n : F) (hn : 0 < InfinitePlace.Completion.ringEquivRealOfIsReal hw (algebraMap F w.Completion n)) : ∃ r : w.Completion, r ≠ 0 ∧ ((1 : w.Completion) ⊗ₜ[F] (algebraMap F E n) : w.Completion ⊗[F] E) = 4 * (r ⊗ₜ[F] (1 : E)) ^ 4

theorem **exists_scal_real** : ∀ (D : D7) (μ : D.host.E), IsSkew μ → ∀ u₀ : (AE D)ˣ, ∃ u ε : (AE D)ˣ, (ε : AE D) * u * cA D u = incA D μ ∧ (∀ v, compFin D.host.F D.host.E v u = compFin D.host.F D.host.E v u₀) ∧ ∀ w, compInf D.host.F D.host.E w (ε * cA D ε) = 4

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIdeleTraceZeroNormAdjustRealComponentsStatement

theorem **_root_.GlobalTheory.T.exists_idele_traceZero_norm_adjust_realComponents** : ∀ (D : D7) (μ : D.host.E), IsSkew μ → ∀ u₀ : (AE D)ˣ, ∃ u ε : (AE D)ˣ, (ε : AE D) * u * cA D u = incA D μ ∧ (∀ v, compFin D.host.F D.host.E v u = compFin D.host.F D.host.E v u₀) ∧ ∀ w, compInf D.host.F D.host.E w (ε * cA D ε) = 4

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIdemSumLeft

theorem **latO_le** (S T : HSp D) (LS LT) {p q : GlobalTheory.FinPl.Pl D → Prop} (h : ∀ v, p v → q v) : latO S T LS LT p ≤ latO S T LS LT q

theorem **mem_latO_at** (q : GlobalTheory.FinPl.SplitPt D k) {S T : HSp D} {LS LT} {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) (hh : IsHalf q.e u) : u ∈ latO S T LS LT (· = q.v)

theorem **int_iff_up_mem** (q : GlobalTheory.FinPl.SplitPt D k) (hq : q.UnitAA) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {LS : GlobalTheory.FinPl.Lat S} {LT : GlobalTheory.FinPl.Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (eS : (LS : Set (Fin S.rank → D.host.E)) = intVec D S.rank) (eT : (LT : Set (Fin T.rank → D.host.E)) = intVec D T.rank) (hFS : ∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) (hFT : ∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) (x : Matrix Y Y' k) : (∀ r s, x r s ∈ 𝒪[k]) ↔ (ofSplit q).up FS FT x ∈ stdHom S T

theorem **gamma_int** (q : GlobalTheory.FinPl.SplitPt D k) {Y : Type} [Fintype Y] [DecidableEq Y] {T : HSp D} (FT : Frame k T.rank Y) {LT : GlobalTheory.FinPl.Lat T} (eT : (LT : Set (Fin T.rank → D.host.E)) = intVec D T.rank) (hFT : ∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) (r s) : FT.γ r s ∈ 𝒪[k]

theorem **mk_eq_mk_of_int** {n : Type} {x y : n → k} (h : ∀ c, y c − x c ∈ 𝒪[k]) : (Submodule.Quotient.mk x : _ ⧸ box n (1 : kˣ)) = Submodule.Quotient.mk y

theorem **mul_toLoc** (ι : E →+* k) {Y Y' : Type} [Fintype Y] [Fintype Y'] {a b nS nT : ℕ} (hS : nS = a) (hT : nT = b) (f : (Fin a → E) ≃ₗ[E] (Fin a → E)) (f' : (Fin b → E) ≃ₗ[E] (Fin b → E)) (e' : Y' ≃ Fin a) (e : Y ≃ Fin b) (β : Matrix (Fin nS) Y' k) (M : Matrix (Fin nT) Y k) (hβ : ∀ r c, β r c = ι (f (Pi.single (e' c) 1) (finCongr hS r))) (hM : ∀ r c, M r c = ι (f' (Pi.single (e c) 1) (finCongr hT r))) (γ : Matrix (Fin nT) (Fin nS) E) : M * toLoc ι e e' (Formalism.L2.Col.inFr f f' (Matrix.toLin' (γ.reindex (finCongr hT) (finCongr hS)))) = γ.map ι * β

theorem **twSt_h** {X : Formalism.L2.Col.Fields} {Pr : Formalism.L2.Col.Primes X} {R : Ranks} (Ch : Formalism.L2.Col.Chars X Pr R.m) (sel) {S₀ : Set ℕ} (mu : Formalism.L2.DIV.Tup X.F X.E X.K S₀) (cst : Formalism.L2.Col.LocCst X (Formalism.L2.Col.rowSG Pr sel mu)) (m2 j : ℕ) (al : (Formalism.L2.Col.twSt Ch sel mu cst m2 j).β) (P : Formalism.L2.Col.rowSG Pr sel mu) (hP : P.1 ∈ Pr.vGs) : (Formalism.L2.Col.twSt Ch sel mu cst m2 j).h al P = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIndexZeroLevelElementGivenReductionAtPlace

theorem **red_at** {PP : Formalism.L2.DIV.Pl D.host.E} (G : GLoc v) {j n : ℕ} (hr : D.R.r ≤ n) {e : Ix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t) j ≃ Fin n} (he : IsOrd zPos e) (π : Lev.EAf D →+* G.k) (h1 : ∀ a, π (toEAf D a) = G.q.pr a) (h3 : ∀ s : Formalism.L2.Herm.intAf D.host.F D.host.E, π s ∈ 𝒪[G.k]) (h4 : ∀ s : Formalism.L2.Herm.intAf D.host.F D.host.E, resS D.host.F PP s = 0 ↔ valuation G.k (π s) < 1) (Y : GL (Fin n) (Formalism.L2.Herm.intAf D.host.F D.host.E)) (hy : ∀ a b, ¬ FreeIx G.S.ρ a b → valuation G.k (π (Y.val (e a) (e b)) − (1 : Matrix _ _ G.k) a b) < 1) : Formalism.UG.bc (resS D.host.F PP) Y ∈ blockTri (blkOf D.R.r D.R.t) ∧ ∃ x, GlobalTheory.LevZero.Red G PP x (upperLeftBlock hr (Formalism.UG.bc (resS D.host.F PP) Y))

theorem **lev_back** {cm : GlobalTheory.ChainOnto.Com D} {st : StAt (c.V 0)} {b : GA D (c.V 0)} {G : cm.Pg → GLoc v} (hB : GlobalTheory.LevZero.Bot cm st b G) (PP : cm.Pg) (c' : ℕ) (hc : 0 + c' = (G PP).s + 1) (x : GA D (c.V 0)) (xk : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin c') (G PP).k) (hx : (ofSplit (G PP).q).IsLoc ((G PP).FY 0 c' hc) x xk) (hxk : xk ∈ (((G PP).S.T (Fin D.R.t)).lev 0 c' hc).J) : b⁻¹ * x * b ∈ levS (c.V 0) st.β st.L

theorem **finE_mem_St** {cm : GlobalTheory.ChainOnto.Com D} {st : StAt (c.V 0)} {b : GA D (c.V 0)} {G : cm.Pg → GLoc v} (hB : GlobalTheory.LevZero.Bot cm st b G) (g : GA D (c.V 0)) (hg : g ∈ levS (c.V 0) st.β st.L) : finE (c.V 0) g ∈ St st.β

theorem **red_of_congr** (G : GLoc v) (PP : Formalism.L2.DIV.Pl D.host.E) (x x' : G.S.P.J) (g : GL (Fin D.R.r) (kP PP)) (h : GlobalTheory.LevZero.Red G PP x' g) (hc : (x'.1 : GL (Fin D.R.k ⊕ Fin (G.s + 1)) G.k)⁻¹ * x.1 ∈ GlobalTheory.SplitWeil.KOne G.k (Fin D.R.k ⊕ Fin (G.s + 1))) : GlobalTheory.LevZero.Red G PP x g

theorem **GlobalTheory.LevZeroPlace.one_place** : ∀ {D : D7} {c : Comm D} {v : VCom c} (cm : GlobalTheory.ChainOnto.Com D) (st : StAt (c.V 0)) (b : GA D (c.V 0)) (G : cm.Pg → GLoc v) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), GlobalTheory.LevZero.Bot cm st b G → ∀ Q : GA D (c.V 0) → Prop, (∀ k ∈ levS (c.V 0) st.β st.L, ∀ k' ∈ levS (c.V 0) st.β st.L, Q k → Q k' → Q (k * k')) → (∀ (PP : cm.Pg) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (c.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → b⁻¹ * n * b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ * n * b)) → (∀ x ∈ GfC (c.X 0), b⁻¹ * embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS h0).o x * b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ * embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS h0).o x * b)) → ∀ (PP : cm.Pg)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIndexZeroLevelElementGivenReductionAtPlaceStatement

theorem **_root_.GlobalTheory.T.exists_indexZeroLevel_element_given_reduction_at_place** : ∀ {D : D7} {c : Comm D} {v : VCom c} (cm : GlobalTheory.ChainOnto.Com D) (st : StAt (c.V 0)) (b : GA D (c.V 0)) (G : cm.Pg → GLoc v) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), GlobalTheory.LevZero.Bot cm st b G → ∀ Q : GA D (c.V 0) → Prop, (∀ k ∈ levS (c.V 0) st.β st.L, ∀ k' ∈ levS (c.V 0) st.β st.L, Q k → Q k' → Q (k * k')) → (∀ (PP : cm.Pg) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (c.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → b⁻¹ * n * b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ * n * b)) → (∀ x ∈ GfC (c.X 0), b⁻¹ * embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS h0).o x * b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ * embA (c.bes 0 D.R.rS h0).d (c.be

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsInjectiveModel

theorem **e_mul_ι** (x : k) : p.e * p.ι x = p.ι x

theorem **loc_gram_mul** (V : HSp D) : loc p V.gram * loc p V.gram⁻¹ = 1

theorem **loc_mul_gram** (V : HSp D) : loc p V.gram⁻¹ * loc p V.gram = 1

theorem **up_mem** (x : Matrix Y Y' k) : p.up FS FT x ∈ half S T p.e

theorem **dual_mem** (b : Matrix Y Y' k) : dual p FS FT b ∈ half S T (cA D p.e)

theorem **up_add** (x y : Matrix Y Y' k) : p.up FS FT (x + y) = p.up FS FT x + p.up FS FT y

theorem **dual_add** (x y : Matrix Y Y' k) : dual p FS FT (x + y) = dual p FS FT x + dual p FS FT y

theorem **up_zero** : p.up FS FT 0 = 0

theorem **dual_zero** : dual p FS FT 0 = 0

theorem **continuous_up** : Continuous (p.up FS FT)

theorem **continuous_dual** : Continuous (dual p FS FT)

theorem **π_hO** (x y : WO S T) : p.π (hO S T x y) = Matrix.trace (loc p S.gram⁻¹ * (Matrix.map y fun z => p.π (cA D z))ᵀ * loc p T.gram * Matrix.map x p.π)

theorem **π_hO_dual** (t b : Matrix Y Y' k) : p.π (hO S T (p.up FS FT t) (dual p FS FT b)) = ∑ r, ∑ q, b r q * t r q

theorem **exists_dual** {y : WO S T} (hy : IsHalf (cA D p.e) y) : ∃ b, dual p FS FT b = y

theorem **char_sympl** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (h : p.IsChar ψ ψk) (t b : Matrix Y Y' k) : ψ (symplO S T (p.up FS FT t) (dual p FS FT b)) = pairing ψk t b

theorem **exists_isLoc** (F : GlobalTheory.LocMod.Frame k S.rank Y) (gk : GL Y k) : ∃ g, p.IsLoc F g gk

theorem **exists_injective_model** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y), ∃ Tm, p.IsModel ψ FS FT Tm ∧ Function.Injective Tm

theorem **exists_place** : ∀ (D : D7) (e : Formalism.L2.Theta.AE D), IsSplitIdem D e → ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (p : Place D k), p.e = e

theorem **exists_place_closed** (D : D7) : ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k), Nonempty (Place D k)

theorem **exists_char_comp_loc** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k), (∀ ψ, IsStdChar D ψ (-1) → ∃ ψk : AddChar k Circle, p.IsChar ψ ψk ∧ Continuous ψk ∧ ψk ≠ 1) ∧ (∀ χ : (Formalism.L2.Theta.AE D)ˣ →* Circle, ∃ μ : kˣ →* ℂˣ, p.IsComp χ μ) ∧ (∀ (S : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) (gk : GL Y k), ∃ g, p.IsLoc F g gk)

theorem **exists_frames** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (s : k), s ≠ 0 → (∀ (B : BesLev D ρ) (FA : GlobalTheory.LocMod.Frame k B.A.rank n), (∃ F, p.IsCoreFrame B s FA F) ∧ (∃ F : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin (2 * ρ + 1)), p.IsBesFrame B s FA F) ∧ p.hz B ≠ 0) ∧ ∀ (J : JacLev D ρ) (hh : k), hh ≠ 0 → ∀ F₀ : GlobalTheory.LocMod.Frame k J.W₀.rank n, ∃ F : GlobalTheory.LocMod.Frame k J.W.rank (n ⊕ Fin (2 * ρ)), p.IsJacFrame J s hh F₀ F

theorem **inhabitants** : (∀ D : D7, ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k), Nonempty (Place D k)) ∧ (∀ (D : D7) (e : Formalism.L2.Theta.AE D), IsSplitIdem D e → ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (p : Place D k), p.e = e) ∧ (∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k), (∀ ψ, IsStdChar D ψ (-1) → ∃ ψk : AddChar k Circle, p.IsChar ψ ψk ∧ Continuous ψk ∧ ψk ≠ 1) ∧ (∀ χ : (Formalism.L2.Theta.AE D)ˣ →* Circle, ∃ μ : kˣ →* ℂˣ, p.IsComp χ μ) ∧ (∀ (S : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) (gk : GL Y k), ∃ g, p.IsLoc F g gk)) ∧ (∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Plac

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsInnerProductSmoothVectorsTensor

theorem **ellA_tprod** (a b : ∀ i, E i) : ellA a (PiTensorProduct.tprod ℂ b) = ∏ i, ⟪a i, b i⟫_ℂ

theorem **prod_inner_update** [DecidableEq ι] (a b : ∀ i, E i) (i : ι) (z : E i) : ∏ j, ⟪Function.update a i z j, b j⟫_ℂ = ⟪z, b i⟫_ℂ * ∏ j ∈ Finset.univ \ {i}, ⟪a j, b j⟫_ℂ

theorem **ellA_update_add** [DecidableEq ι] (a : ∀ i, E i) (i : ι) (x y : E i) : ellA (Function.update a i (x + y)) = ellA (Function.update a i x) + ellA (Function.update a i y)

theorem **ellA_update_smul** [DecidableEq ι] (a : ∀ i, E i) (i : ι) (c : ℂ) (x : E i) : ellA (Function.update a i (c • x)) = conj c • ellA (Function.update a i x)

theorem **ellC_apply** (a : ∀ i, E i) (w : ⊗[ℂ] i, E i) : ellC a w = conj (ellA a w)

theorem **ellC_tprod** (a b : ∀ i, E i) : ellC a (PiTensorProduct.tprod ℂ b) = ∏ i, ⟪b i, a i⟫_ℂ

theorem **ipT_tprod** (a b : ∀ i, E i) : ipT (PiTensorProduct.tprod ℂ a) (PiTensorProduct.tprod ℂ b) = ∏ i, ⟪b i, a i⟫_ℂ

theorem **ipT_herm** (w w' : ⊗[ℂ] i, E i) : ipT w w' = conj (ipT w' w)

theorem **ipT_map** {G : ι → Type*} [∀ i, NormedAddCommGroup (G i)] [∀ i, InnerProductSpace ℂ (G i)] (f : ∀ i, G i →ₗ[ℂ] E i) (hf : ∀ i x y, ⟪f i x, f i y⟫_ℂ = ⟪x, y⟫_ℂ) (x y : ⊗[ℂ] i, G i) : ipT (PiTensorProduct.map f x) (PiTensorProduct.map f y) = ipT x y

theorem **ipT_pos_fd** [∀ i, FiniteDimensional ℂ (E i)] (x : ⊗[ℂ] i, E i) (hx : x ≠ 0) : 0 < (ipT x x).re

theorem **exists_fd** (w : ⊗[ℂ] i, E i) : ∃ F : ∀ i, Submodule ℂ (E i), (∀ i, FiniteDimensional ℂ (F i)) ∧ ∃ w₀ : ⊗[ℂ] i, F i, PiTensorProduct.map (fun i => (F i).subtype) w₀ = w

theorem **ipT_pos** (w : ⊗[ℂ] i, E i) (hw : w ≠ 0) : 0 < (ipT w w).re

theorem **exists_ipD_smT** : ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S), ∃ ipD : SmT M →ₗ[ℂ] SmT M →ₗ⋆[ℂ] ℂ, (∀ v v' : M.Vec, ipD (PiTensorProduct.tprod ℂ v) (PiTensorProduct.tprod ℂ v') = ∏ τ, ⟪(v' τ : M.H τ), (v τ : M.H τ)⟫_ℂ) ∧ (∀ w w', ipD w w' = conj (ipD w' w)) ∧ ∀ w, w ≠ 0 → 0 < (ipD w w).re

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsInnerProductSmoothVectorsTensorStatement

theorem **_root_.GlobalTheory.T.exists_inner_product_smooth_vectors_tensor** : ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S), ∃ ipD : SmT M →ₗ[ℂ] SmT M →ₗ⋆[ℂ] ℂ, (∀ v v' : M.Vec, ipD (PiTensorProduct.tprod ℂ v) (PiTensorProduct.tprod ℂ v') = ∏ τ, ⟪(v' τ : M.H τ), (v τ : M.H τ)⟫_ℂ) ∧ (∀ w w', ipD w w' = conj (ipD w' w)) ∧ ∀ w, w ≠ 0 → 0 < (ipD w w).re

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntSmulMemOfFg

theorem **exists_int_smul_mem_of_fg** {E : Type} [Field E] {V : Type} [AddCommGroup V] [Module (NumberField.RingOfIntegers E) V] {L M : Submodule (NumberField.RingOfIntegers E) V} (h : ∀ x : V, ∃ N : ℤ, N ≠ 0 ∧ N • x ∈ L) (hM : M.FG) : ∃ N : ℤ, N ≠ 0 ∧ ∀ x ∈ M, N • x ∈ L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegerOfRelativeTraceNormIntegral

theorem **integral_of_trace_norm_integral** : ∀ (D : D7) (w : D.host.E) (t n : 𝒪 D.host.F), w + D.sp.cm.sig w = algebraMap D.host.F D.host.E t → w * D.sp.cm.sig w = algebraMap D.host.F D.host.E n → ∃ x : 𝒪 D.host.E, (x : D.host.E) = w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegerOfRelativeTraceNormIntegralStatement

theorem **_root_.GlobalTheory.T.exists_integer_of_relative_trace_norm_integral** : ∀ (D : D7) (w : D.host.E) (t n : 𝒪 D.host.F), w + D.sp.cm.sig w = algebraMap D.host.F D.host.E t → w * D.sp.cm.sig w = algebraMap D.host.F D.host.E n → ∃ x : 𝒪 D.host.E, (x : D.host.E) = w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegerSubConjNotMemSelfConjugatePlace

theorem **exists_sub_star_notMem** : ∀ (D : D7) (Q : Pl D.host.E), IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegerSubConjNotMemSelfConjugatePlaceStatement

theorem **_root_.GlobalTheory.T.exists_integer_sub_conj_notMem_self_conjugate_place** : ∀ (D : D7) (Q : Pl D.host.E), IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegrableBoundDoublingIntegrandQuotient

theorem **exists_integrable_bound_dblIntegrandQ** : ∀ (D : D7) (V : HSp D) (μ : (AE D)ˣ →* Circle) (m : Measure (GA D V)), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → ∀ (C' : ℝ) (c : ℕ), (∀ g₁ g₂ : GA D V, GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ C' * (GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂) ^ c) → ∀ (C : ℝ) (N : ℕ), ∃ bound : AutQ V × AutQ V → ℝ, Integrable bound ((autMeasure V m).prod (autMeasure V m)) ∧ ∀ E : GA D (dblSp V) → ℂ, LeftInv (GQ (dblSp V)) E → (∀ g, ‖E g‖ ≤ C * GlobalTheory.Grow.height (dblSp V) g ^ N) → ∀ q, ‖GlobalTheory.Eis.dblIntegrandQ V μ E f₁ f₂ q‖ ≤ bound q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegrableBoundDoublingIntegrandQuotientStatement

theorem **_root_.GlobalTheory.T.exists_integrable_bound_doubling_integrand_quotient** : ∀ (D : D7) (V : HSp D) (μ : (AE D)ˣ →* Circle) (m : Measure (GA D V)), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → ∀ (C' : ℝ) (c : ℕ), (∀ g₁ g₂ : GA D V, GlobalTheory.Grow.height (dblSp V) (io V (g₁, g₂)) ≤ C' * (GlobalTheory.Grow.height V g₁ * GlobalTheory.Grow.height V g₂) ^ c) → ∀ (C : ℝ) (N : ℕ), ∃ bound : AutQ V × AutQ V → ℝ, Integrable bound ((autMeasure V m).prod (autMeasure V m)) ∧ ∀ E : GA D (dblSp V) → ℂ, LeftInv (GQ (dblSp V)) E → (∀ g, ‖E g‖ ≤ C * GlobalTheory.Grow.height (dblSp V) g ^ N) → ∀ q, ‖GlobalTheory.Eis.dblIntegrandQ V μ E f₁ f₂ q‖ ≤ bound q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralAdeleAtPlaceWithComponent

theorem **evalAtPlace_singleAtPlace_self** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v) : evalV F v (singleV F v y) = y

theorem **evalAtPlace_singleAtPlace_of_ne** (F : Type) [Field F] [NumberField F] (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v) {w : HeightOneSpectrum (𝒪 F)} (hw : w ≠ v) : evalV F w (singleV F v y) = 0

theorem **exists_integralAdele_atPlace_component_tmul** (D : D7) (v : Pl D) (a : Fv D.host.F v) (ha : a ∈ v.adicCompletionIntegers D.host.F) (x : 𝒪 D.host.E) : ∃ y ∈ intAt D (· = v), compAt D v y = a ⊗ₜ[D.host.F] (x : D.host.E)

theorem **exists_intAt_compAt_eq** : ∀ (D : D7) (v : Pl D) (z : Fv D.host.F v ⊗[D.host.F] D.host.E), z ∈ intV D v → ∃ x ∈ intAt D (· = v), compAt D v x = z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralAdeleAtPlaceWithComponentStatement

theorem **_root_.GlobalTheory.T.exists_integralAdele_atPlace_with_component** : ∀ (D : D7) (v : Pl D) (z : Fv D.host.F v ⊗[D.host.F] D.host.E), z ∈ intV D v → ∃ x ∈ intAt D (· = v), compAt D v x = z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralAdelePrescribedResiduesAndConj

theorem **sigInt_oToS** {D : D7} (a : 𝒪 D.host.E) : sigInt (instOf D) (oToS D.host.F D.host.E a) = oToS D.host.F D.host.E (cO D a)

theorem **exists_res_and_conj** (D : D7) (Pg : Finset (Pl D.host.E)) (h : ∀ PP ∈ Pg, ∀ PP' ∈ Pg, ¬ Formalism.IndefR.Lev.IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralAdelePrescribedResiduesAndConjStatement

theorem **_root_.GlobalTheory.T.exists_integralAdele_prescribed_residues_and_conj** : ∀ (D : D7) (Pg : Finset (Pl D.host.E)), (∀ PP ∈ Pg, ∀ PP' ∈ Pg, ¬ Formalism.IndefR.Lev.IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralBasisOfLatticeAtPlaceStable

theorem **exists_basis_of_dense** {A E : Type} [CommRing A] [Field E] [NumberField E] (ι : E →+* A) (ε : A) (O : NonUnitalSubring A) (n : ℕ) (Λ : AddSubgroup (Fin n → A)) (hε : ε * ε = ε) (hOε : ∀ a ∈ O, ε * a = a) (hint : ∀ u : 𝒪 E, ε * ι u ∈ O) (hdense : ∀ T : ℕ, 0 < T → ∀ a ∈ O, ∃ u : 𝒪 E, ∃ z ∈ O, a = ε * ι u + T * z) (hframe : ∀ L : Submodule (𝒪 E) (Fin n → E), L.FG → Submodule.span E (L : Set (Fin n

→ E)) = ⊤ → ∃ (B : GL (Fin n) E) (c : E), c ≠ 0 ∧ ε ∗ ι c⁻¹ ∈ O ∧ (∀ x ∈ L, ∀ i, c ∗ (((B⁻¹ : GL (Fin n) E) : Matrix (Fin n) (Fin n) E) *ᵥ x) i ∈ (algebraMap (𝒪 E) E).range) ∧ ∀ j, (fun i => c ∗ (B : Matrix (Fin n) (Fin n) E) i j) ∈ L) (hst : ∀ a ∈ O, ∀ y ∈ Λ, a • y ∈ Λ) (m : ℕ) (hm : 0 < m) (hmΛ : ∀ y : Fin n → A, (∀ i, y i ∈ O) → (m : A) • y ∈ Λ) (N : ℕ) (hN : 0 < N) (hNΛ : ∀ y ∈ Λ, ∀ i, (N : A) ∗ y i ∈ O) : ∃ β β' : Matrix (Fin n) (Fin n) A, (∀ i j, ε ∗ β i j = β i j) ∧ (∀ i j, ε ∗ β' i

theorem **finPart_finIn** (z : Af D.host.F ⊗[D.host.F] D.host.E) : finPart D (finIn D z) = z

theorem **idem_mul_incA** (P : Pl D → Prop) (x : D.host.E) : idem D P ∗ incA D x = pairA D.host.F 0 (ePl D P) ⊗ₜ[D.host.F] x

theorem **idem_mul_int** (P : Pl D → Prop) (u : 𝒪 D.host.E) : idem D P ∗ incA D u ∈ intAt D P

theorem **idem_mul_inv** (v : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ v.asIdeal) : idem D (· = v) ∗ incA D (algebraMap D.host.F D.host.E (c : D.host.F))⁻¹ ∈ intAt D (· = v)

theorem **intAE_dense** (T : ℕ) (hT : 0 < T) {z : Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ intAE D) : ∃ u : 𝒪 D.host.E, ∃ w ∈ intAE D, z = (1 : Af D.host.F) ⊗ₜ[D.host.F] (u : D.host.E) + T • w

theorem **intAt_dense** (P : Pl D → Prop) (T : ℕ) (hT : 0 < T) (a : AE D) (ha : a ∈ intAt D P) : ∃ u : 𝒪 D.host.E, ∃ z ∈ intAt D P, a = idem D P ∗ incA D u + T ∗ z

theorem **exists_basis** : ∀ (D : D7) (n : ℕ) (v : Pl D) (ε : AE D) (Λ : AddSubgroup (Fin n → AE D)), Tame D v → ε ∗ ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) → (∀ y ∈ Λ, ∀ i, y i ∈ idAt D (· = v)) → (∀ a ∈ intAt D (· = v), ∀ y ∈ Λ, a • y ∈ Λ) → (∃ m : ℕ, 0 < m ∧ ∀ y : Fin n → AE D, (∀ i, y i ∈ intAt D (· = v)) → (m : AE D) • y ∈ Λ) → (∃ N : ℕ, 0 < N ∧ ∀ y ∈ Λ, ∀ i, (N : AE D) ∗ y i ∈ intAt D (· = v)) → ∃ β β' : Matrix (Fin n) (Fin n) (AE D), (∀ i j, β i j ∈ idAt D (· = v)) ∧ (∀ i j, β' i j ∈ idAt D (· = v)) ∧ β ∗ β' = ε • (1 : Matrix (Fin n) (Fin n) (AE D)) ∧ β' ∗ β = ε • (1 : Matrix (Fin n) (Fin n) (AE D)) ∧ (∀ y : Fin n → AE D, (∀ i, y i ∈ idAt D (· = v)) → (y ∈ Λ ↔ ∀ i, (β' *ᵥ y) i ∈ intAt D (· = v)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralBasisOfLatticeAtPlaceStableStatement

theorem **_root_.GlobalTheory.T.exists_integralBasis_of_latticeAtPlace_stable** : ∀ (D : D7) (n : ℕ) (v : Pl D) (ε : AE D) (Λ : AddSubgroup (Fin n → AE D)), Tame D v → ε ∗ ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) → (∀ y ∈ Λ, ∀ i, y i ∈ idAt D (· = v)) → (∀ a ∈ intAt D (· = v), ∀ y ∈ Λ, a • y ∈ Λ) → (∃ m : ℕ, 0 < m ∧ ∀ y : Fin n → AE D, (∀ i, y i ∈ intAt D (· = v)) → (m : AE D) • y ∈ Λ) → (∃ N : ℕ, 0 < N ∧ ∀ y ∈ Λ, ∀ i, (N : AE D) ∗ y i ∈ intAt D (· = v)) → ∃ β β' : Matrix (Fin n) (Fin n) (AE D), (∀ i j, β i j ∈ idAt D (· = v)) ∧ (∀ i j, β' i j ∈ idAt D (· = v)) ∧ β ∗ β' = ε • (1 : Matrix (Fin n) (Fin n) (AE D)) ∧ β' ∗ β = ε • (1 : Matrix (Fin n) (Fin n) (AE D)) ∧ (∀ y : Fin n → AE D, (∀ i, y i ∈ idAt D (· = v)) → (y ∈ Λ ↔ ∀ i, (β' *ᵥ y) i ∈ intAt D (· = v)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralFramePointOfInertHalfLatticePoint

theorem **cA_algebraMap_place** (D : D7) (ε : AF D.host.F) : cA D (algebraMap (AF D.host.F) (AE D) ε) = algebraMap (AF D.host.F) (AE D) ε

theorem **placeIdem_mul** {D : D7} {v : HeightOneSpectrum (𝒪 D.host.F)} {ε : AF D.host.F} (hε : IsPlaceIdem D v ε) (a : AE D) : algebraMap (AF D.host.F) (AE D) ε ∗ a = GlobalTheory.EmbChart.tensAt D.host.E v (compFin D.host.F D.host.E v a)

theorem **pair_mem_intL** (S : HSp D) {s : Fin S.rank → D.host.E} (hs : s ∈ L0c D S) {z : Fin S.rank → w.adicCompletion D.host.E} (hz : z ∈ latAt w (frameLat S)) : ∑ m, algebraMap D.host.E (w.adicCompletion D.host.E) (∑ k, star (s k) ∗ S.gram k m) ∗ z m ∈ Formalism.UG.intL (w.adicCompletion D.host.E)

theorem **secW_piW** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (a : AE D) : secW hw h (piW hw h a) = GlobalTheory.EmbChart.tensAt D.host.E v (compFin D.host.F D.host.E v a)

theorem **piW_tmul** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (α : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F) (m : D.host.E) : piW hw h (pairA D.host.F 0 α ⊗ₜ[D.host.F] m) = complMap h (evalV D.host.F v α) ∗ algebraMap D.host.E (w.adicCompletion D.host.E) m

theorem **latO_frame_int** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) {S T : HSp D} {k : w.adicCompletion D.host.E} {aS : GL (Fin S.rank) (w.adicCompletion D.host.E)} {aT : GL (Fin T.rank) (w.adicCompletion D.host.E)} (hk : GlobalTheory.InertLoc.IsGen D w k) (haS : IsFrameAt w (frameLat S) aS) (haT : IsFrameAt w (frameLat T) aT) {u : WO S T} (hu : u ∈ latO S T (L0c D S) (L0c D T) (fun _ => True)) (i : Fin T.rank) (j : Fin S.rank) : (k⁻¹ • ((aT⁻¹).1 ∗ u.map (piW hw h) ∗ aS.1)) i j ∈ Formalism.UG.intL (w.adicCompletion D.host.E)

theorem **exists_uPt_of_half** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (S T : HSp D) (k : w.adicCompletion D.host.E) (aS : GL (Fin S.rank) (w.adicCompletion D.host.E)) (aT : GL (Fin T.rank) (w.adicCompletion D.host.E)), GlobalTheory.InertLoc.IsGen D w k → IsFrameAt w (frameLat S) aS → IsFrameAt w (frameLat T) aT → ∀ e e', IsPrimeIdem w e e' → ∀ u ∈ latO S T (L0c D S) (L0c D T) (fun _ => True), IsHalf e u v IsHalf (cA D e) u → ∃ x : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E), (∀ i j, x i j ∈ Formalism.UG.intL (w.adicCompletion D.host.E)) ∧ uPt hw h k aS aT x = u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralFramePointOfInertHalfLatticePointStatement

theorem **_root_.GlobalTheory.T.exists_integralFramePoint_of_inertHalf_latticePoint** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (S T : HSp D) (k : w.adicCompletion D.host.E) (aS : GL (Fin S.rank) (w.adicCompletion D.host.E)) (aT : GL (Fin T.rank) (w.adicCompletion D.host.E)), GlobalTheory.InertLoc.IsGen D w k → IsFrameAt w (frameLat S) aS → IsFrameAt w (frameLat T) aT → ∀ e e', IsPrimeIdem w e e' → ∀ u ∈ latO S T (L0c D S) (L0c D T) (fun _ => True), IsHalf e u v IsHalf (cA D e) u → ∃ x : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E), (∀ i j, x i j ∈ Formalism.UG.intL (w.adicCompletion D.host.E)) ∧ uPt hw h k aS aT x = u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralKeepsDoublingDiagonalAdicCompletio

theorem **mem_GLint_of_entries** {L : Type} [Field L] {ι : Type} [Fintype ι] [DecidableEq ι] {O : Subring L} {x : GL ι L} (h1 : ∀ i j, x.1 i j ∈ O) (h2 : ∀ i j, (x⁻¹).1 i j ∈ O) : x ∈ Formalism.UG.GLint ι O

theorem **exists_kappa** {L : Type} [Field L] {O : Subring L} {n : ℕ} {ϖ : L} (h0 : ϖ ≠ 0) (hϖO : ϖ ∈ O) (d : Fin n → ℤ) (t k₁ k₂ : GL (Fin n) L) (ht : t.1 = Matrix.diagonal (fun i => ϖ ^ d i)) (hk₁ : k₁ ∈ Formalism.UG.GLint (Fin n) O) (hk₂ : k₂ ∈ Formalism.UG.GLint (Fin n) O) : ∃ κ ∈ Formalism.UG.GLint (Fin (n + n)) O, KeepsDiag finSumFinEquiv (blkGL finSumFinEquiv (k₁ ∗ t ∗ k₂, 1) ∗ κ).1 ∧ leviMat finSumFinEquiv (blkGL finSumFinEquiv (k₁ ∗ t ∗ k₂, 1) ∗ κ).1 = k₁.1 ∗ Matrix.diagonal (fun i => ϖ ^ max (d i) 0) ∧ quotMat finSumFinEquiv (blkGL finSumFinEquiv (k₁ ∗ t ∗ k₂, 1) ∗ κ).1 = k₁.1 ∗ Matrix.diagonal (fun i => ϖ ^ min (d i) 0)

theorem **ordL_det_mul_diagonal** {L : Type} [Field L] [Valued L (WithZero (Multiplicative ℤ))] {O : Subring L} (hO : ∀ x ∈ O, Valued.v x ≤ (1 : WithZero (Multiplicative ℤ))) {n : ℕ} {ϖ : L} (hϖ : Valued.v ϖ = ((Multiplicative.ofAdd (-1 : ℤ)) : Multiplicative ℤ) : WithZero (Multiplicative ℤ))) {k : GL (Fin n) L} (hk : k ∈ Formalism.UG.GLint (Fin n) O) (f : Fin n → ℤ) : ordL (k.1 ∗ Matrix.diagonal fun i => ϖ ^ f i).det = ∑ i, f i

theorem **graph_shear** : ∀ (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ) (ϖ : w.adicCompletion E), IsUnifV E w ϖ → ∀ (d : Fin n → ℤ) (t k₁ k₂ : GL (Fin n) (w.adicCompletion E)), t.1 = Matrix.diagonal (fun i => ϖ ^ d i) → k₁ ∈ KGL w n → k₂ ∈ KGL w n → ∃ κ ∈ KGL w (n + n), KeepsDiag finSumFinEquiv (blkGL finSumFinEquiv (k₁ * t * k₂, 1) * κ).1 ∧ ordL (leviMat finSumFinEquiv (blkGL finSumFinEquiv (k₁ * t * k₂, 1) * κ).1).det = ∑ i, max (d i) 0 ∧ ordL (quotMat finSumFinEquiv (blkGL finSumFinEquiv (k₁ * t * k₂, 1) * κ).1).det = ∑ i, min (d i) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralKeepsDoublingDiagonalAdicCompletioStatement

theorem **_root_.GlobalTheory.T.exists_integral_keepsDoublingDiagonal_adicCompletion** : ∀ (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ) (ϖ : w.adicCompletion E), IsUnifV E w ϖ → ∀ (d : Fin n → ℤ) (t k₁ k₂ : GL (Fin n) (w.adicCompletion E)), t.1 = Matrix.diagonal (fun i => ϖ ^ d i) → k₁ ∈ KGL w n → k₂ ∈ KGL w n → ∃ κ ∈ KGL w (n + n), KeepsDiag finSumFinEquiv (blkGL finSumFinEquiv (k₁ * t * k₂, 1) * κ).1 ∧ ordL (leviMat finSumFinEquiv (blkGL finSumFinEquiv (k₁ * t * k₂, 1) * κ).1).det = ∑ i, max (d i) 0 ∧ ordL (quotMat finSumFinEquiv (blkGL finSumFinEquiv (k₁ * t * k₂, 1) * κ).1).det = ∑ i, min (d i) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralMatrixRelatingIntegralVectorZBases

theorem **exists_integralMatrix_of_same_multiplicationMatrices** {E : Type*} [Field E] [NumberField E] {n N : ℕ} (e e₀ : Fin N → Fin n → E) (hli : LinearIndependent ℤ e) (he : ∀ u : Fin n → E, (∀ i, IsIntegral ℤ (u i)) ↔ ∃ z : Fin N → ℤ, u = ∑ i, z i • e i) (he₀ : ∀ u : Fin n → E, (∀ i, IsIntegral ℤ (u i)) ↔ ∃ z : Fin N → ℤ, u = ∑ i, z i • e₀ i) (h : ∀ (k : Module.Free.ChooseBasisIndex ℤ (𝒪 E)) (j : Fin N) (z : Fin N → ℤ), (integralBasis E k : E) • e j = ∑ i, z i • e i → (integralBasis E k : E) • e₀ j = ∑ i, z i • e₀ i) : ∃ M : Matrix (Fin n) (Fin n) E, (∀ i j, IsIntegral ℤ (M i j)) ∧ ∀ j, M *ᵥ e j = e₀ j

theorem **exists_integral_matrix_of_same_mul** : ∀ (I : AutInst) {N : ℕ} (e e₀ : Fin N → Fin I.n → I.E), IsZBasis I e → IsZBasis I e₀ → (∀ (k : Module.Free.ChooseBasisIndex ℤ (𝒪 I.E)) (j : Fin N) (z : Fin N → ℤ), (integralBasis I.E k : I.E) • e j = ∑ i, z i • e i → (integralBasis I.E k : I.E) • e₀ j = ∑ i, z i • e₀ i) → ∃ M : Matrix (Fin I.n) (Fin I.n) I.E, (∀ i j, IsIntegral ℤ (M i j)) ∧ ∀ j, M *ᵥ e j = e₀ j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralMatrixRelatingIntegralVectorZBasesStatement

theorem **_root_.GlobalTheory.T.exists_integralMatrix_relating_integralVectorZBases** : ∀ (I : AutInst) {N : ℕ} (e e₀ : Fin N → Fin I.n → I.E), IsZBasis I e → IsZBasis I e₀ → (∀ (k : Module.Free.ChooseBasisIndex ℤ (𝒪 I.E)) (j : Fin N) (z : Fin N → ℤ), (integralBasis I.E k : I.E) • e j = ∑ i, z i • e i → (integralBasis I.E k : I.E) • e₀ j = ∑ i, z i • e₀ i) → ∃ M : Matrix (Fin I.n) (Fin I.n) I.E, (∀ i j, IsIntegral ℤ (M i j)) ∧ ∀ j, M *ᵥ e j = e₀ j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralMatrixUpImageCompletedHomLattice

theorem **sec_mul'** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (a b : k) : q.sec (a * b) = q.sec a * q.sec b

private lemma **mulVec_sec_eq_sec_mulVec** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {m n : Type} [Fintype n] (u : Matrix m n (AE D)) (z : n → k) : u *ᵥ (fun j => q.sec (z j)) = fun i => q.sec ((u.map q.pr *ᵥ z) i)

theorem **exists_int_up** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → ∀ u : WO S T, u ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v) → IsHalf q.e u → ∃ x : Matrix Y Y' k, (∀ r s, x r s ∈ 𝒪[k]) ∧ u = (ofSplit q).up FS FT x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralMatrixUpImageCompletedHomLatticeStatement

theorem **_root_.GlobalTheory.T.exists_integral_matrix_up_image_completed_hom_lattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → ∀ u : WO S T, u ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v) → IsHalf q.e u → ∃ x : Matrix Y Y' k, (∀ r s, x r s ∈ 𝒪[k]) ∧ u = (ofSplit q).up FS FT x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralScaledBoxPiSupersetOfIsCompact

theorem **box_subset_box_of_valuation_le** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) {s t : kˣ} (h : valuation k (s : k) ≤ valuation k (t : k)) : box n s ⊆ box n t

theorem **exists_box_superset_of_isCompact** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] (K : Set (n → k)) : IsCompact K → ∀ r : kˣ, ∃ R : kˣ, valuation k (r : k) ≤ valuation k (R : k) ∧ K ⊆ box n R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralScaledBoxPiSupersetOfIsCompactStatement

theorem **_root_.GlobalTheory.T.exists_integralScaledBoxPi_superset_of_isCompact** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] (K : Set (n → k)), IsCompact K → ∀ r : kˣ, ∃ R : kˣ, valuation k (r : k) ≤ valuation k (R : k) ∧ K ⊆ box n R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralThetaEtaTwelfthPowerConjugates

theorem **exists_integral_theta_ratio** : PeriodLatticeAgreement.LatticePeriodEta → ThetaConstantWeber.ThetaWeberStatement → ∀ (D : D7), Formalism.MainNotions.GL3Theta.admJ D → ∀ (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) (M : IntermediateField ℚ ℂ) [FiniteDimensional ℚ M] (t : (M →ₐ[ℚ] ℂ) → ℂ), SignedLogMeans.HidaSetting D cc δ M t → ∃ X : M, X ≠ 0 ∧ IsIntegral ℤ (X : ℂ) ∧ ∀ (ρ : M →ₐ[ℚ] ℂ) (ω₁ τ : ℂ), ω₁ ≠ 0 → 0 < τ.im → JacobiThetaHalfCharacteristic.IsPeriodLattice (lamConj M ρ (lamOf D cc δ)) ω₁ τ → ∀ a b : Fin 2, JacobiThetaHalfCharacteristic.IsEvenChar a b → JacobiThetaHalfCharacteristic.thetaChar a b (t ρ / ω₁) τ = 0 → ‖ρ X‖ = (‖JacobiThetaHalfCharacteristic.thetaChar a b 0 τ‖ / ‖ModularForm.eta τ‖) ^ 12

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralThetaEtaTwelfthPowerConjugatesStatement

theorem **_root_.GlobalTheory.T.exists_integral_thetaEta_twelfth_power_conjugates** : PeriodLatticeAgreement.LatticePeriodEta → ThetaConstantWeber.ThetaWeberStatement → ∀ (D : D7), Formalism.MainNotions.GL3Theta.admJ D → ∀ (cc : Ideal (𝒪 D.host.K)) (δ : ℂ) (M : IntermediateField ℚ ℂ) [FiniteDimensional ℚ M] (t : (M →ₐ[ℚ] ℂ) → ℂ), SignedLogMeans.HidaSetting D cc δ M t → ∃ X : M, X ≠ 0 ∧ IsIntegral ℤ (X : ℂ) ∧ ∀ (ρ : M →ₐ[ℚ] ℂ) (ω₁ τ : ℂ), ω₁ ≠ 0 → 0 < τ.im → JacobiThetaHalfCharacteristic.IsPeriodLattice (lamConj M ρ (lamOf D cc δ)) ω₁ τ → ∀ a b : Fin 2, JacobiThetaHalfCharacteristic.IsEvenChar a b → JacobiThetaHalfCharacteristic.thetaChar a b (t ρ / ω₁) τ = 0 → ‖ρ X‖ = (‖JacobiThetaHalfCharacteristic.thetaChar a b 0 τ‖ / ‖ModularForm.eta τ‖) ^ 12

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralTraceIdempotentAtTamePlace

theorem **idAt₂** (v : Pl D) : algebraMap (AF D.host.F) (AE D) (cP (· = v)) ∈ idAt D (· = v)

theorem **eq_alg_cP** {v : Pl D} {ε : AE D} (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) : ε = algebraMap (AF D.host.F) (AE D) (cP (· = v))

theorem **placeIdem_eq_cP** {v : Pl D} {εF : AF D.host.F} (h : IsPlaceIdem D v εF) : εF = cP (· = v)

theorem **half_mem_int** (v : Pl D) (h2 : (2 : 𝒪 D.host.F) ∉ v.asIdeal) : algebraMap D.host.F (Fv D.host.F v) (2⁻¹) ∈ v.adicCompletionIntegers D.host.F

theorem **exists_trace_one_at_tame** : ∀ (D : D7) (v : Pl D) (ε : AE D) (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a), Tame D v → ∃ w ∈ intAt D (· = v), w + cA D w = ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIntegralTraceIdempotentAtTamePlaceStatement

theorem **_root_.GlobalTheory.T.exists_integral_trace_idempotent_at_tamePlace** : ∀ (D : D7) (v : Pl D) (ε : AE D) (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a), Tame D v → ∃ w ∈ intAt D (· = v), w + cA D w = ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsInvOfNear

theorem **MLe.mono** {A : Matrix (Fin a) (Fin b) L} {γ δ : Γ} (h : MLe v A γ) (hγ : γ ≤ δ) : MLe v A δ

theorem **MLe.mul** {A : Matrix (Fin a) (Fin b) L} {B : Matrix (Fin b) (Fin c) L} {γ δ : Γ} (h : MLe v A γ) (h' : MLe v B δ) : MLe v (A * B) (γ * δ)

theorem **MLe.add** {A B : Matrix (Fin a) (Fin b) L} {γ : Γ} (h : MLe v A γ) (h' : MLe v B γ) : MLe v (A + B) γ

theorem **MLe.sub** {A B : Matrix (Fin a) (Fin b) L} {γ : Γ} (h : MLe v A γ) (h' : MLe v B γ) : MLe v (A - B) γ

theorem **mle_one** : MLe v (1 : Matrix (Fin a) (Fin a) L) 1

theorem **exists_mle** (A : Matrix (Fin a) (Fin b) L) : ∃ m : Γ, 1 ≤ m ∧ MLe v A m

theorem **mle_of_fix** {Z W : Matrix (Fin a) (Fin b) L} {D : Matrix (Fin a) (Fin a) L} {γ δ : Γ} (hδ : δ < 1) (hD : MLe v D δ) (hW : MLe v W γ) (e : Z = W + D * Z) : MLe v Z γ

theorem **exists_inv_of_near** {A : Matrix (Fin a) (Fin a) L} {δ : Γ} (hδ : δ < 1) (hA : MLe v (A - 1) δ) : ∃ B : Matrix (Fin a) (Fin a) L, A * B = 1 ∧ B * A = 1 ∧ MLe v (B - 1) δ

theorem **MLe.of_near** {A : Matrix (Fin a) (Fin a) L} {δ : Γ} (hδ : δ ≤ 1) (hA : MLe v (A - 1) δ) : MLe v A 1

theorem **MLe.near_mul** {A B : Matrix (Fin a) (Fin a) L} {δ : Γ} (hδ : δ ≤ 1) (hA : MLe v (A - 1) δ) (hB : MLe v (B - 1) δ) : MLe v (A * B - 1) δ

theorem **exists_approx** (U : Matrix (Fin a) (Fin b) (P.adicCompletion E)) {δ : WithZero (Multiplicative ℤ)} (hδ : δ ≠ 0) : ∃ u : Matrix (Fin a) (Fin b) E, MLe Valued.v (u.map (algebraMap E (P.adicCompletion E)) - U) δ

theorem **mle_map_iff** {A : Matrix (Fin a) (Fin b) E} {γ : WithZero (Multiplicative ℤ)} : MLe Valued.v (A.map (algebraMap E (P.adicCompletion E))) γ ↔ MLe (P.valuation E) A γ

theorem **intAt_inv_mul** {c z : E} (hc : c ≠ 0) (h : P.valuation E z ≤ P.valuation E c) : IntAt P (c⁻¹ * z)

theorem **eq_of_near** {K : Type} (Ξ : HomE E a b → K) {c d : E} (hc0 : c ≠ 0) (hc : P.valuation E c < 1) (hd0 : d ≠ 0) (hcong : ∀ (k : Sp E a ≃ₗ[E] Sp E a) (k' : Sp E b ≃ₗ[E] Sp E b), (∀ s l, IntAt P (c⁻¹ * (k (Pi.single s (1 : E)) l - (Pi.single s (1 : E) : Sp E a) l))) → (∀ s l, IntAt P (c⁻¹ * (k' (Pi.single s (1 : E)) l - (Pi.single s (1 : E) : Sp E b) l))) → ∀ x, Ξ (inFr k k' x) = Ξ x) (hinv : ∀ x z : HomE E a b, (∀ s l, IntAt P (d⁻¹ * z (Pi.single s 1) l)) → Ξ (x + z) = Ξ x) (x x' : HomE E a b) (U : Matrix (Fin a) (Fin a) (P.adicCompletion E)) (U' : Matrix (Fin b) (Fin b) (P.adicCompletion E)) (hU : MLe Valued.v (U - 1) (P.valuation E c)) (hU' : MLe Valued.v (U' - 1) (P.valuation E c)) (h : MLe Valued.v (U' * (LinearMap.toMatrix' x').map (algebraMap E (P.adicCompletion E)) - (LinearMap.toMatrix' x).map (algebraMap E (P.adicCompletion E)) * U) (P.valuation E d)) : Ξ x' = Ξ x

theorem **val_qOf_lt_one** (P : HeightOneSpectrum (𝒪 E)) : P.valuation E ((qOf P : ℕ) : E) < 1

theorem **qOf_pos** (P : HeightOneSpectrum (𝒪 E)) : 0 < qOf P

theorem **mle_frame** {g : GL (Fin a) (AfE F E)} {p : DecoF E a ST SG Q} (h : IsPtOf hc g p) (w : SG) : MLe Valued.v ((g⁻¹ : GL (Fin a) (AfE F E)).1.map (adAt F w.1) * (LinearMap.toMatrix' (p.deco.1.liftFr w).toLinearMap).map (algebraMap E (w.1.adicCompletion E)) - 1) (WithZero.exp (-((1 : ℕ) : ℤ)))

theorem **frame_mul_inFr** (f : Sp E a ≃ₗ[E] Sp E a) (f' : Sp E b ≃ₗ[E] Sp E b) (x : HomE E a b) : LinearMap.toMatrix' f'.toLinearMap * LinearMap.toMatrix' (inFr f f' x) = LinearMap.toMatrix' x * LinearMap.toMatrix' f.toLinearMap

theorem **exists_modulus** {k y : GL (Fin a) (AfE F E)} {k' y' : GL (Fin b) (AfE F E)} {p q : DecoF E a ST SG Q} {p' q' : DecoF E b ST SG Q} (hp : IsPtOf hc k p) (hp' : IsPtOf hc k' p') (hq : IsPtOf hc (y * k) q) (hq' : IsPtOf hc (y' * k') q') (w : SG) {d : E} (hd : d ≠ 0) : ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : HomE E a b, (∀ r s, Valued.v (adAt F w.1 (((y'⁻¹ : GL (Fin b) (AfE F E)).1 * ((LinearMap.toMatrix' x).map (inclE (F

theorem **xiAt_inFr** {X : Fields} {Pr : Primes X} {R : Ranks} {Ch : Chars X Pr R.m} {sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)} {S : Set ℕ} {mu : Tup X.F X.E X.K S} {cst : LocCst X (rowSG Pr sel mu)} {L : ChIdx X Pr R (QE sel (mu : Chr X.F X.E X.K))} {G : ChGlob X R.m} {m2 : ℕ} (hX : X.AbelianUnramifiedExtensionConditions) (hPr : Pr.Ok) (hCh : Ch.Ok) (hsel : IsSel X sel) (hL : L.Ok Ch sel mu cst G m2) {i : ℕ} (hi : i < m2) (al : L.β i) (be : L.β (i + 1)) (P : rowSG Pr sel mu) (k : Sp X.E (R.N i) ≃ₗ[X.E] Sp X.E (R.N i)) (k' : Sp X.E (R.N (i + 1)) ≃ₗ[X.E] Sp X.E (R.N (i + 1))) (hk : ∀ s l, IntAt P.1 (((qOf P.1 : X.E) ^ cst.D P)⁻¹ * (k (Pi.single s (1 : X.E)) l - (Pi.single s (1 : X.E) : Sp X.E (R.N i)) l))) (hk' : ∀ s l, IntAt P.1 (((qOf P.1 : X.E) ^ cst.D P)⁻¹ * (k' (Pi.single s (1 : X.E)) l - (Pi.single s (1 : X.E) : Sp X.E (R.N (i + 1))) l))) (x : HomE X.E (R.N i) (R.N (i

theorem **exp_le_val_qOf** {E : Type} [Field E] [NumberField E] {P : HeightOneSpectrum (𝒪 E)} (h : GlobalTheory.IndefiniteRowSideFullHecke.SourceLatticeInvariance.DegOneUnram P) : WithZero.exp (-((1 : ℕ) : ℤ)) ≤ P.valuation E ((qOf P : ℕ) : E)

theorem **entry_transport** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (u : B.Word) (i : ℕ) (hm : i + 2 < B.R.m) (al : (I.idx I.mu0).β i) (be : (I.idx I.mu0).β (i + 1)) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup I.mu0)) (p q : PtAt B.R I.sel (I.tup I.mu0) (I.cst I.mu0) u (i)) (p' q' : PtAt B.R I.sel (I.tup I.mu0) (I.cst I.mu0) u (i + 1)) (k y : TC.GA B (B.R.N (i))) (k' y' : TC.GA B (B.R.N (i + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (i + 1)) → TC.PtOf (y ∗ k) q.1.1 → TC.PtOf (y' ∗ k') q'.1.1 → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → Formalism.L2.Col.XiAt I.Ch I.sel (I.tup I.mu0) (I.cst I.mu0) i P ((I.idx I.mu0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsInvertibleFamilyEvalSpan

theorem **evalLin_apply** {ι : Type} {a : ℕ} (y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : evalLin y F = evalAt y F

theorem **exists_evalAt_span** (ι : Type) [Fintype ι] (a : ℕ) (lam : ι → DomWeight a) (n : ι → ℕ) (W : Type) [AddCommGroup W] [Module ℂ W] (T : PY ι a →ₗ[ℂ] W) : ∃ (m : ℕ) (y : Fin m → ι → Matrix (Fin a) (Fin a) ℂ) (P : Fin m → W), (∀ j, IsInv (y j)) ∧ ∀ F ∈ Vn lam n, T F = ∑ j, evalAt (y j) F • P j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsInvertibleFamilyEvalSpanStatement

theorem **_root_.GlobalTheory.T.exists_invertibleFamily_eval_span** : ∀ (ι : Type) [Fintype ι] (a : ℕ) (lam : ι → DomWeight a) (n : ι → ℕ) (W : Type) [AddCommGroup W] [Module ℂ W] (T : PY ι a →ₗ[ℂ] W), ∃ (m : ℕ) (y : Fin m → ι → Matrix (Fin a) (Fin a) ℂ) (P : Fin m → W), (∀ j, IsInv (y j)) ∧ ∀ F ∈ Vn lam n, T F = ∑ j, evalAt (y j) F • P j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsCombAt

theorem **exists_isCombAt** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (p : Place D k) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (Ξ : Matrix Y Y' k → ℂ) : ∃ c : WO S T →₀ ℂ, p.IsCombAt ψ FS FT c Ξ

theorem **exists_cg** : ∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (q : GlobalTheory.FinPl.SplitPt D k) (s : ℕ) (S : SymData k s D.R.k) (FX : ∀ j, j ≤ s + 1 → GlobalTheory.LocMod.Frame k (D.sp.X j).rank (XIx (Fin D.R.k) (Fin D.R.t) j)) (FM : GlobalTheory.LocMod.Frame k (D.sp.X (s + 1 + 1)).rank (Ix (Fin D.R.k ⊕ Fin (s + 1)) (Fin D.R.t) (s + 2) ⊕ Unit)), IsStdChar D d.c.ψ (-1) → ∃ cg : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ, (∀ j c (H : j + 1 + c = s), (ofSplit q).IsCombAt d.c.ψ (FX j (by omega)) (FX (j + 1) (by omega)) (cg j) ((S.T (Fin D.R.t)).entry S.ψ ν j c H 1 1)) ∧ (ofSplit q).IsCombAt d.c.ψ (FX s (by omega)) (FX (s + 1) le_rfl) (cg s) (fun x₀ => (S.T (Fin D.R.t)).Psi 1 x₀ (S.y0 (Fin D.R.t))) ∧ (ofSplit q).IsCombAt d.c.ψ (FX (s + 1) le_rfl) FM (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsComplIsUnif

theorem **intAt_iff** (a : E) : IntAt P a ↔ P.valuation E a ≤ 1

theorem **valuation_absNorm** (h : ((Ideal.absNorm P.asIdeal : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) : P.valuation E ((Ideal.absNorm P.asIdeal : ℕ) : E) = WithZero.exp (-1 : ℤ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsConjAtOfEmbedding

theorem **exists_isConjAt_of_embedding** (K : Type) [Field K] [NumberField K] [StarRing K] (h : ∃ σ : K →+* ℂ, ∀ x : K, σ (star x) = starRingEnd ℂ (σ x)) : ∃ v0 : InfinitePlace K, IsConjAt v0

theorem **isConjAt_exists_main** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K], (∃ σ : K →+* ℂ, ∀ x : K, σ (star x) = starRingEnd ℂ (σ x)) → ∃ v0 : InfinitePlace K, IsConjAt v0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsIdemSumConjOfIsIdemSum

theorem **exists_isIdemSum_cA** : ∀ (D : D7) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∃ Q' : Finset (Formalism.L2.DIV.Pl D.host.E), IsIdemSum (cA D E₂) Q'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsIdemSumConjOfIsIdemSumStatement

theorem **_root_.GlobalTheory.T.exists_isIdemSum_conj_of_isIdemSum** : ∀ (D : D7) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∃ Q' : Finset (Formalism.L2.DIV.Pl D.host.E), IsIdemSum (cA D E₂) Q'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsJ

theorem **exists_isJ** : ∀ (D : D7) (u : D.what₂.Word) (M : DSModel D.sp.X' (parB (Ptw D) D.R.m)), ∃ j : SmT M →ₗ[ℂ] (FrP D → ℂ), (∀ (i : Fin (D.R.N D.R.m - 1)) τ, (((D.fm u).sg τ).w i : ℚ) = sgHC D.R (D.hcW u τ) i - rhoVec (D.R.N D.R.m - 1) i) → IsJ u M j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsLatticeFrameBcEq

theorem **idem_glue_mul** {R : Type} [CommRing R] {m : Type} [Fintype m] [DecidableEq m] {e : R} (hee : e ∗ e = e) {x y x' y' : Matrix m m R} (h1 : x ∗ x' = 1) (h2 : y ∗ y' = 1) : ((1 - e) • x + e • y) ∗ ((1 - e) • x' + e • y') = 1

theorem **bc_idemGlue_of_zero** {R S : Type} [CommRing R] [CommRing S] {m : Type} [Fintype m] [DecidableEq m] {e : R} (hee : e ∗ e = e) (x y : GL m R) (φ : R →+* S) (h : φ e = 0) : Formalism.UG.bc φ (idemGlue e hee x y) = Formalism.UG.bc φ x

theorem **bc_idemGlue_of_one** {R S : Type} [CommRing R] [CommRing S] {m : Type} [Fintype m] [DecidableEq m] {e : R} (hee : e ∗ e = e) (x y : GL m R) (φ : R →+* S) (h : φ e = 1) : Formalism.UG.bc φ (idemGlue e hee x y) = Formalism.UG.bc φ y

theorem **isLatticeFrame_idemGlue** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] {n : ℕ} {L : Submodule (𝒪 E) (Fin n → E)} {e : AF F ⊗[F] E} (hee : e ∗ e = e) {x : GL (Fin n) (AF F ⊗[F] E)} (hx : IsLatticeFrame L x) (y : GL (Fin n) (AF F ⊗[F] E)) (he : ∀ v, compFin F E v e = 0) : IsLatticeFrame L (idemGlue e hee x y)

theorem **exists_isLatticeFrame_bc_eq** (D : D7) (A : HSp D) (a : GL (Fin A.rank) (AE D)) : ∃ b : GL (Fin A.rank) (AE D), IsLatticeFrame (frameLat A) b ∧ ∀ w, Formalism.UG.bc (compInf D.host.F D.host.E w) b = Formalism.UG.bc (compInf D.host.F D.host.E w) a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsModelChg

theorem **chg_inv** (FS : Frame k M Y') (FT : Frame k N Y) (GS : Frame k M Z') (GT : Frame k N Z) (x : Matrix Y Y' k) : chg GS GT FS FT (chg FS FT GS GT x) = x

theorem **frame_sandwich** (FS : Frame k M Y') (FT : Frame k N Y) (GS : Frame k M Z') (GT : Frame k N Z) (x : Matrix Y Y' k) : GT.β ∗ chg FS FT GS GT x ∗ GS.γ = FT.β ∗ x ∗ FS.γ

theorem **chg_add** (FS : Frame k M Y') (FT : Frame k N Y) (GS : Frame k M Z') (GT : Frame k N Z) (x y : Matrix Y Y' k) : chg FS FT GS GT (x + y) = chg FS FT GS GT x + chg FS FT GS GT y

theorem **continuous_chg** (FS : Frame k M Y') (FT : Frame k N Y) (GS : Frame k M Z') (GT : Frame k N Z) : Continuous (chg FS FT GS GT)

theorem **up_chg** {Y Y' Z Z' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] [Fintype Z] [DecidableEq Z] [Fintype Z'] [DecidableEq Z'] {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (GS : Frame k S.rank Z') (GT : Frame k T.rank Z) (x : Matrix Y Y' k) : p.up GS GT (chg FS FT GS GT x) = p.up FS FT x

theorem **exists_isModel_chg** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (Z Z' : Type) [Fintype Z] [DecidableEq Z] [Fintype Z'] [DecidableEq Z'] (GS : Frame k S.rank Z') (GT : Frame k T.rank Z) (ψ : AddChar (AF D.host.F) ℂ) : ∀ Tm, p.IsModel ψ FS FT Tm → ∃ Tm', p.IsModel ψ GS GT Tm' ∧ ∀ (Φ : SB (Matrix Y Y' k)) (Φ' : SB (Matrix Z Z' k)), (∀ x, Φ.1 x = Φ'.1 (GT.γ ∗ FT.β ∗ x ∗ (FS.γ ∗ GS.β))) → Tm' Φ' = Tm Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsParamVecOne

theorem **chiDelta_trivParGL** (b : GL (Fin n) (w.adicCompletion E)) : Formalism.UG.chiDelta E w (fun i => (trivParGL w n i : ℂ)) b = 1

theorem **isSatVecGL_one** : IsSatVecGL w (1 : Representation ℂ (GL (Fin n) (w.adicCompletion E)) ℂ) 1 (Finset.univ.val.map (trivParGL w n))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsSphSec

theorem **npow_zpow** {q : ℕ} (hq : q ≠ 0) (r : ℝ≥0) (m : ℤ) (hr : (r : ℝ) = (q : ℝ) ^ (-(2 ∗ m))) (t : ℂ) : GlobalTheory.Doubling.npow r t = (q : ℂ) ^ (-(2 ∗ (m : ℂ) ∗ t))

theorem **exists_isSphSec** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} (hc : IsCMExt F E σ) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) {Q : Matrix (Fin (n + n)) (Fin (n + n)) E} (hQ : Q = dblGram finSumFinEquiv J) (L : Submodule (𝒪 E) (Fin n → E)) (hL : Submodule.IsLattice E L) (μ : (AF F ⊗[F] E)ˣ →∗ Circle) {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (h : w.under (𝒪 F) = v) (hw : IsInertAt F E v w) (hq : qV F v ≠ 0) {s : ℂ} {φ : GLoc σ Q v → ℂ} (h1 : φ ∈ degPSFin σ Q L v (charAt μ v) s) (h2 : ∀ k ∈ KFin σ Q L v, ∀ g, φ (g ∗ k) = φ g) (h3 : φ 1 = 1) {c : w.adicCompletion E →+∗ w.adicCompletion E} {e : GLoc σ J v ≃∗ GU c n} (he : IsInertChart σ J L h c e) : ∃ Φ, GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsSphSec c (qV F v) n (locChar μ w) s Φ ∧ ∀ y, φ (ioH _ σ hQ (y, 1)) = Φ (GlobalTheo

theorem **sphSec_inert** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ (χ : (AE D)ˣ →∗ Circle) (b : ℕ), ResEta D χ b → ∀ (v : FinPl D) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), IsInertAt D.host.F D.host.E v w → UnrAt χ w → ∀ (s : ℂ) (φ : Gv (dblSp A) v → ℂ), IsSphSecAt A L v χ⁻¹ s φ → ∀ (c : w.adicCompletion D.host.E →+∗ w.adicCompletion D.host.E) (ϖ : (w.adicCompletion D.host.E)ˣ) (e : Gv A v ≃∗ GU c A.rank), IsUnramInv (ϖ : w.adicCompletion D.host.E) c (qV D.host.F v) → IsInertChart D.sp.cm.sig A.gram L h c e → GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsUnr (locChar χ⁻¹ w) ∧ locChar χ⁻¹ w ϖ = (GlobalTheory.LFun.val χ w)⁻¹ ∧ ((locChar χ⁻¹ w ϖ : ℂˣ) : ℂ) = (-1) ^ b ∧ ∃ Φ, GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsSphSec c (qV D.host.F v) A.rank (locChar χ⁻¹ w) s Φ ∧ ∀ y, φ (ioAt A v y) = Φ (GlobalTheory

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsotropicDualColumnsSelfDualLatticeAtPlace

theorem **intAE_mul_mem'** {x y : Af D.host.F ⊗[D.host.F] D.host.E} (hx : x ∈ intAE D) (hy : y ∈ intAE D) : x ∗ y ∈ intAE D

theorem **intAt_mul_mem** {v : Pl D} {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a ∗ b ∈ intAt D (· = v)

theorem **hermAt_smul_left** (S : HSp D) (o : AE D) (y x : Fin S.rank → AE D) : hermAt S (o • y) x = o ∗ hermAt S y x

theorem **smul_mem_of_isSDAt** {S : HSp D} {v : Pl D} {M : AddSubgroup (Fin S.rank → AE D)} (hM : IsSDAt S v M) {o : AE D} (ho : o ∈ intAt D (· = v)) {y : Fin S.rank → AE D} (hy : y ∈ M) : o • y ∈ M

theorem **col_mul_mem** {S : HSp D} {v : Pl D} {M : AddSubgroup (Fin S.rank → AE D)} (hM : IsSDAt S v M) {ρ : ℕ} (A : Mx D S.rank ρ) (B : Mx D ρ ρ) (hA : ∀ i, (fun k => A k i) ∈ M) (hB : ∀ m i, B m i ∈ intAt D (· = v)) (i : Fin ρ) : (fun k => (A · B) k i) ∈ M

theorem **gram_entry** (S : HSp D) {ρ : ℕ} (c' : Mx D S.rank ρ) (i j : Fin ρ) : (adj c' · (gA S · c')) i j = hermAt S (fun k => c' k j) (fun k => c' k i)

theorem **smul_mul_smul_one** {ρ : ℕ} (ε w : AE D) (H : Mx D ρ ρ) : (ε • (1 : Mx D ρ ρ)) · (w • H) = (ε ∗ w) • H

theorem **smul_mul_smul_one'** {ρ : ℕ} (ε w : AE D) (H : Mx D ρ ρ) : (w • H) · (ε • (1 : Mx D ρ ρ)) = (ε ∗ w) • H

theorem **adj_gram_self** (S : HSp D) {ρ : ℕ} (c' : Mx D S.rank ρ) : adj (adj c' · (gA S · c')) = adj c' · (gA S · c')

theorem **gram_swap_smul** (S : HSp D) {ρ : ℕ} {ε : AE D} (hεc : cA D ε = ε) (c c' : Mx D S.rank ρ) (hcc : adj c' · (gA S · c) = ε • (1 : Mx D ρ ρ)) : adj c · (gA S · c') = ε • (1 : Mx D ρ ρ)

theorem **pair_sub** (S : HSp D) {ρ : ℕ} {ε : AE D} (c c' : Mx D S.rank ρ) (K : Mx D ρ ρ) (hcc0 : adj c · (gA S · c) = 0) (hcc : adj c' · (gA S · c) = ε • (1 : Mx D ρ ρ)) : adj (c' − c · K) · (gA S · c) = ε • (1 : Mx D ρ ρ)

theorem **pair_sub'** (S : HSp D) {ρ : ℕ} {ε : AE D} (c c' : Mx D S.rank ρ) (K : Mx D ρ ρ) (hcc' : adj c · (gA S · c') = ε • (1 : Mx D ρ ρ)) : adj (c' − c · K) · (gA S · c') = adj c' · (gA S · c') − adj K · (ε • (1 : Mx D ρ ρ))

theorem **self_sub** (S : HSp D) {ρ : ℕ} (c'' c c' : Mx D S.rank ρ) (K : Mx D ρ ρ) (h : c'' = c' − c · K) : adj c'' · (gA S · c'') = adj c'' · (gA S · c') − adj c'' · (gA S · c) · K

theorem **isotropic_core** {S : HSp D} {v : Pl D} {ε : AE D} (hεε : ε ∗ ε = ε) (hεc : cA D ε = ε) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) {M : AddSubgroup (Fin S.rank → AE D)} (hM : IsSDAt S v M) {w : AE D} (hw : w ∈ intAt D (· = v)) (hwc : w + cA D w = ε) {ρ : ℕ} (c c' : Mx D S.rank ρ) (hc : ∀ i, (fun k => c k i) ∈ M) (hc' : ∀ i, (fun k => c' k i) ∈ M) (hcc0 : adj c · (gA S · c) = 0) (hcc : adj c' · (gA S · c) = ε • (1 : Mx D ρ ρ)) : ∃ c'' : Mx D S.rank ρ, (∀ i, (fun k => c'' k i) ∈ M) ∧ adj c'' · (gA S · c) = ε • (1 : Mx D ρ ρ) ∧ adj c'' · (gA S · c'') = 0

theorem **dual_isotropic** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (ε : AE D) (hεε : ε ∗ ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) (M : AddSubgroup (Fin S.rank → AE D)), IsSDAt S v M → ∀ w ∈ intAt D (· = v), w + cA D w = ε → ∀ (b : Mx D ρ ρ) (c' : Mx D S.rank ρ), (∀ i j, b i j ∈ idAt D (· = v)) → (∀ i, (fun k => (d.jX · b) k i) ∈ M) → (∀ i, (fun k => c' k i) ∈ M) → adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ) → ∃ c'' : Mx D S.rank ρ, (∀ i, (fun k => c'' k i) ∈ M) ∧ adj c'' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ) ∧ adj c'' · (gA S · c'') = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsIsotropicDualColumnsSelfDualLatticeAtPlaceStatement

theorem **_root_.GlobalTheory.T.exists_isotropic_dual_columns_selfDualLatticeAtPlace** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (ε : AE D) (hεε : ε ∗ ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε ∗ a = a) (M : AddSubgroup (Fin S.rank → AE D)), IsSDAt S v M → ∀ w ∈ intAt D (· = v), w + cA D w = ε → ∀ (b : Mx D ρ ρ) (c' : Mx D S.rank ρ), (∀ i j, b i j ∈ idAt D (· = v)) → (∀ i, (fun k => (d.jX · b) k i) ∈ M) → (∀ i, (fun k => c' k i) ∈ M) → adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ) → ∃ c'' : Mx D S.rank ρ, (∀ i, (fun k => c'' k i) ∈ M) ∧ adj c'' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ) ∧ adj c'' · (gA S · c'') = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsKerTie

theorem **GlobalTheory.Core.DefKer.stdRd_keep** {B : Base} {X : PreX B.toPre} (I : TC B X) (hOk : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : G.Inst (I.rec7 l Z) L} {Gl : G.LatInst (I.rec7 l Z)} {K : G.Consts} {Y : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl K Y cX} (t : KerEntTied ex) (hS : I.StdKer l Z t) : ∃ t' : KerEntTied ex, I.StdKer l Z t' ∧ GlobalTheory.Pdv.StdRd I l Z t' ∧ t'.TopPinned ∧ t'.O = t.O ∧ t'.ent = t.ent

theorem **GlobalTheory.Core.DefKer.all_pins** {B : Base} {X : PreX B.toPre} (I : TC B X) (hOk : I.Ok) (hP : TC.Pins B X) (hTw : I.IsTw) (l : I.Laws) (Z : TC.Later B (I.cols l)) {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : G.Inst (I.rec7 l Z) L} {Gl : G.LatInst (I.rec7 l Z)} {K : G.Consts} {Y : DivX (I.rec7 l Z)} {cX : ExtraC} (ex : Extra (I.rec7 l Z) L hT Gi Gl K Y cX) (O : ∀ u a, GL3Theta.LiftObj (I.rec7 l Z) ((I.rec7 l Z).chTw u a)) (E : KerEnt (I.rec7 l Z) Y ex.tw) (hQ : I.QpIs Y.Qp) (hO : I.StdO l Z Y.Qp O) (hU : ∀ u a i, i < B.R.m → UnfIsAll (O u a) i) (hE : EntOk E O) (hS : ∀ u w h, ThAtServed (ex.ker u w h) E) (hC : ∃ (Pl : Examples₂.EntPl (I.rec7 l Z)) (Bs : PlaceMatrixFamily (I.rec7 l Z)) (h : KerEntHyp Pl O Bs Y ex.tw), E = kerEntC Pl O Bs h) (hpi : ex.errX.piT = X.piT) : ∃ t : KerEntTied ex, I.StdKer l Z t ∧ GlobalTheory.Pdv.StdRd I l Z t ∧ GlobalTheory.Pdv.IsEnt

theorem **GlobalTheory.Core.DefKer.exists_kerTie** : ∀ (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Fr B.toPre) (T : TwDat B) (w : ColW B f T) (Z : w.Lt) (b : CoreB K w Z) (O : Obj (w.D Z)), ofObj O = b.obj → ∃ t : KerEntTied O.ex, TC.StdKer w.I w.l Z t ∧ GlobalTheory.Pdv.StdRd w.I w.l Z t ∧ GlobalTheory.Pdv.IsEntC t ∧ t.Adapted ∧ t.TopPinned ∧ t.AAUnit ∧ t.UnfAll ∧ t.ChTie ∧ t.MemSideU ∧ t.MtTwTie ∧ ∀ a, (t.cst a).sharp = true

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLastPeriodClassDataOfLastStep

theorem **exists_clsT_of_pinTop** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLastPeriodClassDataOfLastStepStatement

theorem **_root_.GlobalTheory.T.exists_lastPeriod_classData_of_lastStep** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLatticeFrameLiftingSpanningVectorsMod

theorem **det_one_sub_smul** (c : R) (X : Matrix (Fin n) (Fin n) R) : ∃ a : R, (1 - c • X).det = 1 - c ∗ a

theorem **isUnit_mix** {e g : R} (he : e ∗ e = e) (hu : ∀ a : R, IsUnit (1 - e ∗ g ∗ a)) (M C X : Matrix (Fin n) (Fin n) R) (h : M ∗ C = 1 - g • X) : IsUnit (e • M + (1 - e) • (1 : Matrix (Fin n) (Fin n) R))

theorem **exists_lift** {A : Type} [CommRing A] (S : Subring A) (g : ℕ) {e t : S} (he : e ∗ e = e) (ht : 1 - e = g ∗ t) (hu : ∀ a : S, IsUnit (1 - e ∗ g ∗ a)) (Λ : Submodule S (Fin n → A)) (β₀ : GL (Fin n) A) (hβ : ∀ x, x ∈ Λ ↔ ∀ i, ((β₀⁻¹).val ∗ᵥ x) i ∈ S) (b : Fin n → Fin n → A) (hb : ∀ j, b j ∈ Λ) (hs : ∀ y ∈ Λ, y ∈ Submodule.span S (Set.range b) ⊔ Submodule.span S {y | ∃ x ∈ Λ, y = (g : A) • x}) : ∃ β : GL (Fin n) A, (∀ x, x ∈ Λ ↔ ∀ i, ((β⁻¹).val ∗ᵥ x) i ∈ S) ∧ ∀ j, ∃ x ∈ Λ, (fun i => β.val i j) - b j = (g : A) • x

theorem **valued_natCast_lt_one_iff** {E : Type} [Field E] [NumberField E] (Q : HeightOneSpectrum (𝒪 E)) (N : ℕ) : Valued.v (N : Q.adicCompletion E) < 1 ↔ (N : 𝒪 E) ∈ Q.asIdeal

theorem **exists_inv_of_valued** {D : D7} (z : Ef D) (h : ∀ Q : Pl D.host.E, Valued.v (cpt Q z) = 1) : ∃ y ∈ OA D, z ∗ y = 1

theorem **coe_one_sub** {A : Type} [CommRing A] (S : Subring A) (g : ℕ) (e a : S) : ((1 - e ∗ g ∗ a : S) : A) = 1 - e ∗ g ∗ a

theorem **one_sub_eq** {A : Type} [CommRing A] (S : Subring A) {g : ℕ} {c e : A} (hc : (g : A) ∗ c = 1) (he : e ∈ S) (ht : c ∗ (1 - e) ∈ S) : (1 - ⟨e, he⟩ : S) = g ∗ ⟨_, ht⟩

theorem **val_quot_le** {g : ℕ} {c e : A} (hc : (g : A) ∗ c = 1) (h : φ e = 1 ∨ φ e = 0 ∧ v (g : K) = 1) : v (φ (c ∗ (1 - e))) ≤ 1

theorem **val_one_sub** {g : ℕ} {a e : A} (ha : v (φ a) ≤ 1) (h : φ e = 1 ∧ v (g : K) < 1 ∨ φ e = 0) : v (φ (1 - e ∗ g ∗ a)) = 1

theorem **quot_mem** {D : D7} {g : ℕ} {c e : Ef D} (hc : (g : Ef D) ∗ c = 1) (he1 : ∀ Q : Pl D.host.E, (g : 𝒪 D.host.E) ∈ Q.asIdeal → cpt Q e = 1) (he0 : ∀ Q : Pl D.host.E, (g : 𝒪 D.host.E) ∉ Q.asIdeal → cpt Q e = 0) : c ∗ (1 - e) ∈ OA D

theorem **valued_one_sub** {D : D7} {g : ℕ} {a e : Ef D} (ha : a ∈ OA D) (he1 : ∀ Q : Pl D.host.E, (g : 𝒪 D.host.E) ∈ Q.asIdeal → cpt Q e = 1) (he0 : ∀ Q : Pl D.host.E, (g : 𝒪 D.host.E) ∉ Q.asIdeal → cpt Q e = 0) (Q : Pl D.host.E) : Valued.v (cpt Q (1 - e ∗ g ∗ a)) = 1

theorem **exists_idem_over** (D : D7) {g : ℕ} (hg : g ≠ 0) : ∃ e c : Ef D, e ∗ e = e ∧ e ∈ OA D ∧ (g : Ef D) ∗ c = 1 ∧ c ∗ (1 - e) ∈ OA D ∧ ∀ a ∈ OA D, ∃ y ∈ OA D, (1 - e ∗ g ∗ a) ∗ y = 1

theorem **idem_subring** {A : Type} [CommRing A] (S : Subring A) {g : ℕ} (h : ∃ e c : A, e ∗ e = e ∧ e ∈ S ∧ (g : A) ∗ c = 1 ∧ c ∗ (1 - e) ∈ S ∧ ∀ a ∈ S, ∃ y ∈ S, (1 - e ∗ g ∗ a) ∗ y = 1) : ∃ e t : S, e ∗ e = e ∧ 1 - e = g ∗ t ∧ ∀ a : S, IsUnit (1 - e ∗ g ∗ a)

theorem **hatL_eq_Lf** (I : AutInst) : Formalism.L2.Herm.hatL (F

theorem **exists_isLiftFrame** : ∀ (D : D7) (g : ℕ), g ≠ 0 → ∀ b : ℕ → VA (instOf D), (∀ j, b j ∈ (instOf D).Lf) → (∀ y ∈ (instOf D).Lf, InMod (instOf D) g (Set.range fun j : Fin (instOf D).n => b j.val) y) → ∃ β, IsLiftFrame (instOf D) g b β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLatticeFrameLiftingSpanningVectorsModStatement

theorem **_root_.GlobalTheory.T.exists_latticeFrame_lifting_spanning_vectors_mod** : ∀ (D : D7) (g : ℕ), g ≠ 0 → ∀ b : ℕ → VA (instOf D), (∀ j, b j ∈ (instOf D).Lf) → (∀ y ∈ (instOf D).Lf, InMod (instOf D) g (Set.range fun j : Fin (instOf D).n => b j.val) y) → ∃ β,

IsLiftFrame (instOf D) g b β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLatticeStabilizerTwistAndLevelAway

private theorem **congGrp_mono'** (V : HSp D) {I J : Ideal (AE D)} (h : I ≤ J) : congGrp V I ≤ congGrp V J

private theorem **idAt_mono_decorated_lattice** {P P' : Pl D → Prop} (h : ∀ v, P v → P' v) : idAt D P ≤ idAt D P'

theorem **grpAt_le_of_imp** (V : HSp D) {P P' : Pl D → Prop} (h : ∀ v, P v → P' v) : grpAt V P ≤ grpAt V P'

private theorem **idAt_inf_le_idAt_and** (P P' : Pl D → Prop) : idAt D P ⊓ idAt D P' ≤ idAt D (fun p => P p ∧ P' p)

private theorem **congGrp_inf_le_congGrp_inf** (V : HSp D) (I J : Ideal (AE D)) : congGrp V I ⊓ congGrp V J ≤ congGrp V (I ⊓ J)

private theorem **mem_grpAt_and_of_mem_grpAt** (V : HSp D) {P P' : Pl D → Prop} {g : GA D V} (h : g ∈ grpAt V P) (h' : g ∈ grpAt V P') : g ∈ grpAt V (fun p => P p ∧ P' p)

private theorem **inK_congr_of_iff** {V : HSp D} {L : Lat V} {P P' : Pl D → Prop} (h : ∀ p, P p ↔ P' p) (κ : GA D V) : InK V L P κ ↔ InK V L P' κ

private theorem **mem_grpAt_of_inK** {V : HSp D} {L : Lat V} {P : Pl D → Prop} {κ : GA D V} (h : InK V L P κ) : κ ∈ grpAt V P

theorem **levels** : ∀ (D : D7) (V : HSp D) (L : Lat V) (gP Q : Finset (Pl D)) (JG : Subgroup (GA D V)), IsSelfDualC V L → Disjoint Q gP → JG ≤ grpAt V (· ∈ gP) → IsCompact (JG : Set (GA D V)) → IsOpen (((JG.subgroupOf (grpAt V (· ∈ gP))) : Subgroup _) : Set (grpAt V (· ∈ gP))) → ∃ K₁ K₂ : Subgroup (GA D V), (∀ g, g ∈ K₁ ↔ InK V L (· ∈ Q) g) ∧ K₁ ≤ grpAt V (· ∈ Q) ∧ K₂ ≤ grpAt V (· ∉ Q) ∧ IsCompact (K₁ : Set (GA D V)) ∧ IsCompact (K₂ : Set (GA D V)) ∧ IsOpen ((K₁.subgroupOf (grpAt V (· ∈ Q)) : Subgroup _) : Set (grpAt V (· ∈ Q))) ∧ IsOpen ((K₂.subgroupOf (grpAt V (· ∉ Q)) : Subgroup _) : Set (grpAt V (· ∉ Q))) ∧ (∀ g, g ∈ K₂ ↔ ∃ κ, InK V L (fun p => p ∉ gP ∧ p ∉ Q) κ ∧ ∃ k ∈ JG, g = κ ∗ k) ∧ ∀ g, (∃ a ∈ K₁, ∃ b ∈ K₂, g = a ∗ b) ↔ ∃ κ, InK V L (· ∉ gP) κ ∧ ∃ k ∈ JG, g = κ ∗ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLatticeStabilizerTwistAndLevelAwayStatement

theorem **_root_.GlobalTheory.T.exists_latticeStabilizer_twist_and_level_away** : ∀ (D : D7) (V : HSp D) (L : Lat V) (gP Q : Finset (Pl D)) (JG : Subgroup (GA D V)), IsSelfDualC V L → Disjoint Q gP → JG ≤ grpAt V (· ∈ gP) → IsCompact (JG : Set (GA D V)) → IsOpen (((JG.subgroupOf (grpAt V (· ∈ gP))) : Subgroup _) : Set (grpAt V (· ∈ gP))) → ∃ K₁ K₂ : Subgroup (GA D V), (∀ g, g ∈ K₁ ↔ InK V L (· ∈ Q) g) ∧ K₁ ≤ grpAt V (· ∈ Q) ∧ K₂ ≤ grpAt V (· ∉ Q) ∧ IsCompact (K₁ : Set (GA D V)) ∧ IsCompact (K₂ : Set (GA D V)) ∧ IsOpen ((K₁.subgroupOf (grpAt V (· ∈ Q)) : Subgroup _) : Set (grpAt V (· ∈ Q))) ∧ IsOpen ((K₂.subgroupOf (grpAt V (· ∉ Q)) : Subgroup _) : Set (grpAt V (· ∉ Q))) ∧ (∀ g, g ∈ K₂ ↔ ∃ κ, InK V L (fun p => p ∉ gP ∧ p ∉ Q) κ ∧ ∃ k ∈ JG, g = κ ∗ k) ∧ ∀ g, (∃ a ∈ K₁, ∃ b ∈ K₂, g = a ∗ b) ↔ ∃ κ, InK V L (· ∉ gP) κ ∧ ∃ k ∈ JG, g = κ ∗ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLatticesGoodPositionAllPlaces

theorem **tau_norm** {E : Type} [Field E] [StarRing E] {m n : ℕ} {G : Matrix (Fin m) (Fin m) E} {G' : Matrix (Fin n) (Fin n) E} {τ : Matrix (Fin m) (Fin n) E} {t lam : E} {ζ : Fin m → E} {ζ₀ : Fin n → E} (h : τᴴ ∗ (G ∗ τ) = t • G') (hτ : τ ∗ᵥ ζ₀ = lam • ζ) : lam ∗ star lam ∗ (star ζ ⬝ᵥ G ∗ᵥ ζ) = t ∗ (star ζ₀ ⬝ᵥ G' ∗ᵥ ζ₀)

theorem **val_norm_xi** {p : SplitPt D k} (h : p.UnitAA) : valuation k (p.pr (incA D (D.host.xi0 ∗ star D.host.xi0))) = 1

theorem **top_idele** (hc : c.Is) (pre : PreT v o) (hs : ∀ n (h : n = D.R.m) x, x ∉ v.tP → ∃ LZ : Lat c.Lz, IsSelfDualC _ LZ ∧ OrthAt (c.oT n h) (c.x n).L v.NT LZ x) {α : (AE D)ˣ} (hα : ∀ (x : Pl D) (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) : ∃ m₀ : (AE D)ˣ, (∀ x ∈ v.tP, IsU x (m₀ : AE D)) ∧ ∀ j (h : j + 2 ∗ 0 + 2 = D.R.m), (∀ x, x ∉ v.tP → IsU x ((m₀ : AE D) ∗ cA D m₀ ∗ incA D (star (c.bes j 0 h).ζ ⬝ᵥ (c.bes j 0 h).L.gram ∗ᵥ (c.bes j 0 h).ζ))) ∧ (∀ i, valuation k ((o.p i).pr (incA D (star (c.bes j 0 h).ζ ⬝ᵥ (c.bes j 0 h).L.gram ∗ᵥ (c.bes j 0 h).ζ))) = valuation k ((o.q.1 : 𝒪[k]) : k)) ∧ ∀ x, x ∉ v.tP → ∀ LL : Lat (c.bes j 0 h).L, IsSelfDualC _ LL → (∀ a ∈ idAt D (· = x), a • ratV (c.bes j 0 h).ζ ∈ latV _ LL (· = x)) ↔ ((m₀⁻¹ : (AE D)ˣ) : AE D) ∗ a ∈ intAt D (· = x)) → ∃ LZ : Lat c.Lz, IsSelfDualC _ LZ ∧ OrthAt (c.oT (j + 1 + 1) h) (c.x

theorem **gam_norm** (α : (AE D)ˣ) : (gam α : AE D) ∗ cA D (gam α : AE D) ∗ incA D (D.host.xi0 ∗ star D.host.xi0) = 1

theorem **level_norm** (pre : PreT v o) (α m₀ : (AE D)ˣ) (h0 : ∀ j (h : j + 2 ∗ 0 + 2 = D.R.m), (∀ x, x ∉ v.tP → IsU x ((m₀ : AE D) ∗ cA D m₀ ∗ incA D (star (c.bes j 0 h).ζ ⬝ᵥ (c.bes j 0 h).L.gram ∗ᵥ (c.bes j 0 h).ζ))) ∧ (∀ i, valuation k ((o.p i).pr (incA D (star (c.bes j 0 h).ζ ⬝ᵥ (c.bes j 0 h).L.gram ∗ᵥ (c.bes j 0 h).ζ))) = valuation k ((o.q.1 : 𝒪[k]) : k))) (ρ : ℕ) : ∀ j (h : j + 2 ∗ ρ + 2 = D.R.m), (∀ x, x ∉ v.tP → IsU x (((gam α ^ ρ ∗ m₀ : (AE D)ˣ) : AE D) ∗ cA D ((gam α ^ ρ ∗ m₀ : (AE D)ˣ) : AE D) ∗ incA D (star (c.bes j ρ h).ζ ⬝ᵥ (c.bes j ρ h).L.gram ∗ᵥ (c.bes j ρ h).ζ))) ∧ (∀ i, valuation k ((o.p i).pr (incA D (star (c.bes j ρ h).ζ ⬝ᵥ (c.bes j ρ h).L.gram ∗ᵥ (c.bes j ρ h).ζ))) = valuation k ((o.q.1 : 𝒪[k]) : k))

theorem **bB_step** (α m₀ : (AE D)ˣ) (ρ : ℕ) (i j : Fin ρ) (h : j.1 = i.1 + 1) : bB α m₀ ρ i = cond α ∗ bB α m₀ ρ j

theorem **bJ_step** (α m₀ : (AE D)ˣ) (ρ : ℕ) (i j : Fin ρ) (h : j.1 = i.1 + 1) : bJ α m₀ ρ i = cond α ∗ bJ α m₀ ρ j

theorem **cond_gam** (α : (AE D)ˣ) : cond α ∗ gam α = α ∗ α

theorem **bJ_succ** (α m₀ : (AE D)ˣ) (σ : ℕ) (i : Fin (σ + 1)) : bJ α m₀ (σ + 1) i = α ∗ (cond α ^ (σ − i.1) ∗ (gam α ^ σ ∗ m₀))

theorem **bJ_last** (α m₀ : (AE D)ˣ) (ρ : ℕ) (r : Fin ρ) (hr : r.1 + 1 = ρ) : bJ α m₀ ρ r = xiU D ∗ cU α ∗ (gam α ^ ρ ∗ m₀)

theorem **exists_N** (hc : c.Is) (M : ∀ j ρ h, Lat (c.bes j ρ h).V₀) (hM : ∀ j ρ h, IsSelfDualC _ (M j ρ h)) (b b' : ∀ ρ, Fin ρ → (AE D)ˣ) : ∃ N : ∀ j, Lat (c.V j), (∀ j, IsSelfDualC _ (N j)) ∧ (∀ j ρ h, HasIdeals (c.bes j ρ h).d (N j) (M j ρ h) (b ρ)) ∧ ∀ j ρ h, HasIdeals (c.jac j ρ h).d (N (j + 1)) (c.x (j + 1)).L (b' ρ)

theorem **exists_goodAt** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι), c.Is → IsSelfDualC c.T v.NT → PreT v o → (∀ n (h : n = D.R.m) x, x ∉ v.tP → ∃ LZ : Lat c.Lz, IsSelfDualC _ LZ ∧ OrthAt (c.oT n h) (c.x n).L v.NT LZ x) → ∃ (N : ∀ j, Lat (c.V j)) (M : ∀ j ρ h, Lat (c.bes j ρ h).V₀) (β₀ : ∀ j ρ h i, Frame (o.p i) (c.bes j ρ h).V₀ (Fin (c.X j).rank ⊕ Fin 1)), (∀ j, j < D.R.m → IsSelfDualC (c.V j) (N j)) ∧ (∀ x, GoodAt { v with N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLatticesGoodPositionAllPlacesStatement

theorem **_root_.GlobalTheory.T.exists_lattices_goodPosition_all_places** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι), c.Is → IsSelfDualC c.T v.NT → PreT v o → (∀ n (h : n = D.R.m) x, x ∉ v.tP → ∃ LZ : Lat c.Lz, IsSelfDualC _ LZ ∧ OrthAt (c.oT n h) (c.x n).L v.NT LZ x) → ∃ (N : ∀ j, Lat (c.V j)) (M : ∀ j ρ h, Lat (c.bes j ρ h).V₀) (β₀ : ∀ j ρ h i, Frame (o.p i) (c.bes j ρ h).V₀ (Fin (c.X j).rank ⊕ Fin 1)), (∀ j, j < D.R.m → IsSelfDualC (c.V j) (N j)) ∧ (∀ x, GoodAt { v with N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLevelData

theorem **exists_levelData** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b : ∀ j, GA D (Yc D j)) (v : VCom (commOf (pk W) u (mkGeo m b))), (commOf (pk W) u (mkGeo m b)).Is → v.τ = y.τ → v.NT = a.LT → (∀ p, p ∈ v.gP ↔ ∃ PP ∈ (pk W).Pg, p = PP.under (𝒪 D.host.F)) → GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg → ∀ (l : ∀ PP : (pk W).Pg, LocX a (pk W).βX PP.1) (S : ∀ PP, SymData (l PP).k (l PP).s D.R.k), SymRead W l S → (∀ PP, LocY (l PP) y v.N (pk W).βY b fun j c hc => ((S PP).T (Fin D.R.t)).bas j c hc) → (∀ PP, (∀ j hj, IsStdFr GlobalTheory.SymBridge.xPos ((l PP).FX j hj)) ∧ IsStdFr GlobalTheory.SymBridge.zPos (l PP).FT) →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLevelElementsStepVector

theorem **fockVec_eq_sum** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t j, j + 1 < D.R.m → ∀ [Fintype (lvP W).H], ∃ κ : (lvP W).H → GA D (Yc D j), (∀ p, κ p ∈ levS (Yc D j) ((pk W).βY j) ((pk W).datYt t j (lvP W))) ∧ ∀ v, IsStepVecOf (pk W).ψ (pk W).θ (Yc D j) (Yc D (j + 1)) (cart ((pk W).MY j)) (cart ((pk W).MY (j + 1))) ((pk W).ΛY j) ((pk W).ΛY (j + 1)) ((pk W).βY j) ((pk W).βY (j + 1)) ((pk W).datYt t j (lvP W)) ((pk W).chi (j + 1)) ((pk W).Q t) ((pk W).xiR t) ((pk W).vac j) v → ∀ Q, pureTensorO (pk W).ψ (pk W).θ (Yc D j) (Yc D (j + 1)) (cart ((pk W).MY j)) (cart ((pk W).MY (j + 1))) (onb ((pk W).MY j)) (onb ((pk W).MY (j + 1))) v Q = ∑ p, (levChi (Yc D j) ((pk W).βY j) ((pk

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLevelElementsStepVectorStatement

theorem **_root_.GlobalTheory.T.exists_level_elements_step_vector** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t j, j + 1 < D.R.m → ∀ [Fintype (lvP W).H], ∃ κ : (lvP W).H → GA D (Yc D j), (∀ p, κ p ∈ levS (Yc D j) ((pk W).βY j) ((pk W).datYt t j (lvP W))) ∧ ∀ v, IsStepVecOf (pk W).ψ (pk W).θ (Yc D j) (Yc D (j + 1)) (cart ((pk W).MY j)) (cart ((pk W).MY (j + 1))) ((pk W).ΛY j) ((pk W).ΛY (j + 1)) ((pk W).βY j) ((pk W).βY (j + 1)) ((pk W).datYt t j (lvP W)) ((pk W).chi (j + 1)) ((pk W).Q t) ((pk W).xiR t) ((pk W).vac j) v → ∀ Q, pureTensorO (pk W).ψ (pk W).θ (Yc D j) (Yc D (j + 1)) (cart ((pk W).MY j)) (cart ((pk W).MY (j + 1))) (onb ((pk W).MY j)) (onb ((pk W).MY (j + 1))) v Q =

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLeviPreimageOfFixIsotropicFrames

theorem **gram_mt** {S : HSp D} (g : GA D S) {k l : ℕ} (x : Mx D S.rank k) (y : Mx D S.rank l) : adj (mt S g · x) · (gA S · (mt S g · y)) = adj x · (gA S · y)

theorem **exists_emb0_eq_of_fix_jX_jY** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (g : GA D S) (hX : mt S g · d.jX = d.jX) (hY : mt S g · d.jY = d.jY) : ∃ b : GA D S₀, d.emb0 b = g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLeviPreimageOfFixIsotropicFramesStatement

theorem **_root_.GlobalTheory.T.exists_levi_preimage_of_fix_isotropic_frames** : ∀ {D : D7} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (g : GA D S), mt S g · d.jX = d.jX → mt S g · d.jY = d.jY → ∃ b : GA D S₀, d.emb0 b = g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLinearIsQuotIntTInt

theorem **exists_open_translations** {M : Type*} [AddCommGroup M] [TopologicalSpace M] [IsTopologicalAddGroup M] {Φ : M → ℂ} (hΦ : Φ ∈ SB M) : ∃ Λ : AddSubgroup M, IsOpen (Λ : Set M) ∧ ∀ x, ∀ y ∈ Λ, Φ (x + y) = Φ x

theorem **exists_isQuotInt_tInt** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c p : ℕ} (hp : 1 ≤ p) (hpc : p ≤ c) {ψ : AddChar k Circle} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] {μ : kˣ →* ℂˣ} (hμ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] {A : Type} [AddCommGroup A] [Module ℂ A] {σ : Representation ℂ (GL (n ⊕ Fin c) k) A} (hσ : IsSmoothRep σ) {B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ} (hB : BilPos ψ μ p σ B) (F : A) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) : ∃ z, IsQuotInt ν μX μG p (tInt ψ ν μ p σ B F Φ.1) z

theorem **exists_linear_isQuotInt_tInt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (_ : 1 ≤ p) (_ : p ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ p σ B) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ'), ∃ Rr : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] ℂ, (∀ Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLinearIsQuotIntTIntStatement

theorem **_root_.GlobalTheory.T.exists_linear_isQuotInt_tInt_alt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (_ : 1 ≤ p) (_ : p ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ p σ B) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ'), ∃ Rr : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] ℂ, (∀ Φ : SB (Matrix

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLinearMapVaryPureTensorFactorSplitPlace

theorem **GlobalTheory.VarySplitFactor.exists_linearMap_pure** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : OmT S T) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (f₀ : WO S T → ℂ) (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → IsBase ψ θ S T sS sT ΛS ΛT f₀ → IsCovolOne (ratL T S) μ ℱ → ∀ (x : Place D → WO S T → ℂ) (f : WO S T → ℂ), IsPure ψ θ S T sS sT ΛS ΛT μ ℱ f₀ x f → ∀ (v : FinPl D) (φ : SplitProj D v) (e : AE D), IsIdemOf φ e → ∀ (BS : GL (Fin S.rank) (Fv D.host.F v)) (BT : GL (Fin T.rank) (Fv D.host.F v)) (Φ₀ : GlobalTheory.SplitWeil.SB (Matrix (Fin T.rank) (Fin S.rank) (Fv D.host.F v))), ∀ Tm, IsPlaceModel ψ (sympl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLinearMapVaryPureTensorFactorSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_linearMap_vary_pureTensor_factor_split_place** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : OmT S T) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (f₀ : WO S T → ℂ) (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → IsBase ψ θ S T sS sT ΛS ΛT f₀ → IsCovolOne (ratL T S) μ ℱ → ∀ (x : Place D → WO S T → ℂ) (f : WO S T → ℂ), IsPure ψ θ S T sS sT ΛS ΛT μ ℱ f₀ x f → ∀ (v : FinPl D) (φ : SplitProj D v) (e : AE D), IsIdemOf φ e → ∀ (BS : GL (Fin S.rank) (Fv D.host.F v)) (BT : GL (Fin T.rank) (Fv D.host.F v)) (Φ₀ : GlobalTheory.SplitWeil.SB (Matrix (Fin T.rank) (Fin S.rank) (Fv D.host.F v))), ∀ Tm

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLinearOfBilinearSchwartzBruhatProduct

theorem **tensorToProd_tmul** {X Y : Type} [TopologicalSpace X] [TopologicalSpace Y] (f : SB X) (g : SB Y) : tensorToProd X Y (f ⊗ₜ g) = prodSB f g

theorem **tensorToProd_injective** (X Y : Type) [TopologicalSpace X] [TopologicalSpace Y] : Function.Injective (tensorToProd X Y)

theorem **exists_linear_of_bilinear_prodSB** : ∀ (X Y : Type) [TopologicalSpace X] [TopologicalSpace Y] (M : Type) [AddCommGroup M] [Module ℂ M] (B : SB X →ₗ[ℂ] SB Y →ₗ[ℂ] M), ∃ l : SB (X × Y) →ₗ[ℂ] M, ∀ f g, l (prodSB f g) = B f g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLinearOfBilinearSchwartzBruhatProductStatement

theorem **_root_.GlobalTheory.T.exists_linear_of_bilinear_schwartzBruhatProduct** : ∀ (X Y : Type) [TopologicalSpace X] [TopologicalSpace Y] (M : Type) [AddCommGroup M] [Module ℂ M] (B : SB X →ₗ[ℂ] SB Y →ₗ[ℂ] M), ∃ l : SB (X × Y) →ₗ[ℂ] M, ∀ f g, l (prodSB f g) = B f g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLocComb

theorem **idAt_mono** {P P' : GlobalTheory.FinPl.Pl D → Prop} (h : ∀ w, P w → P' w) : idAt D P ≤ idAt D P'

theorem **mxAt_of_half** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) {P : GlobalTheory.FinPl.Pl D → Prop} (hP : P q.v) {m n : ℕ} (w : Mx D m n) (h : IsHalf (q.e + cA D q.e) w) : MxAt P w

theorem **exists_fold** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (P : GlobalTheory.FinPl.Pl D → Prop) {ι : Type} (c : ι → WO S T →₀ ℂ) (hc : ∀ i, ∀ w ∈ (c i).support, MxAt P w) (l : List ι) : ∃ c' : WO S T →₀ ℂ, (∀ w ∈ c'.support, MxAt P w) ∧ ∀ f, comb ψ S T c' f = l.foldr (fun i f' => comb ψ S T (c i) f') f

theorem **mxAt_cast** {F : ℕ → HSp D} {T : HSp D} {P : GlobalTheory.FinPl.Pl D → Prop} {s j : ℕ} (h : s = j) (x : WO (F s) T →₀ ℂ) (hx : ∀ w ∈ x.support, MxAt P w) : ∀ w ∈ (cast (congrArg (fun n => WO (F n) T →₀ ℂ) h) x).support, MxAt P w

theorem **conj_iff** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k N Y) (M : GL (Fin N) k) (x : GL Y k) : M.val = F.β * x.val * F.γ ↔ x = conjHom F M

theorem **mem_frameSub** {G : Type} [Group G] {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (ρ : G →* GL (Fin N) k) (F : Frame k N Y) (J : Subgroup (GL Y k)) (g : G) : g ∈ frameSub ρ F J ↔ ∀ x : GL Y k, (ρ g).val = F.β * x.val * F.γ → x ∈ J

theorem **under_inj** {Pg : Finset (Formalism.L2.DIV.Pl D.host.E)} (hT : GlobalTheory.Core.Mi.IsTypePlaces Pg) {PP PP' : Formalism.L2.DIV.Pl D.host.E} (h : PP ∈ Pg) (h' : PP' ∈ Pg) (e : PP.under (𝒪 D.host.F) = PP'.under (𝒪 D.host.F)) : PP = PP'

theorem **exists_locComb** {a : XGeo D} {y : YRat a} {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (S : SymData l.k l.s D.R.k) {N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)} {β : ∀ j, GL (Fin (Yc D j).rank) (Lev.EAf D)} {b : ∀ j, GA D (Yc D j)} {bas : ∀ j c, j + c = l.s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) l.k} (Y : LocY l y N β b bas) (ν : Measure l.k) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ (eg : ∀ j, WO (Yc D j) (Yc D (j + 1)) →₀ ℂ) (egT : WO (Yc D (l.s + 1)) D.sp.X' →₀ ℂ) (cg : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ), (∀ j a' (H : j + a' = l.s), (ofSplit l.q).IsCombAt ψ (Y.FY j (a' + 1) (by omega)) (Y.FY (j + 1) a' (by omega)) (eg j) ((S.T (Fin D.R.t)).Phi j a' H)) ∧ (ofSplit l.q).IsCombAt ψ (Y.FY (l.s + 1) 0 rfl) l.FT egT (S.T (Fin D.R.t)).PhiTop ∧ (∀ j a' (H : j + 1 + a' = l.s), (ofSplit l.q).IsCombAt ψ (l.FX j (by omega))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLocPlOff

theorem **exists_latInd** (q : GlobalTheory.FinPl.SplitPt D k) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hS : IsSelfDualC S M) (hT : IsSelfDualC T M') : ∃ Φ₀ : SB (Matrix Y Y' k), Φ₀.1 0 = 1 ∧ Φ₀.1 = Set.indicator {x | (ofSplit q).up FS FT x ∈ latO S T (M : Set _) (M' : Set _) (· = q.v)} (fun _ => 1)

theorem **exists_locPl** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (q : GlobalTheory.FinPl.SplitPt D k) (hq : q.UnitAA) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (Φ : SB (Matrix Y Y' k)) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hS : IsSelfDualC S M) (hT : IsSelfDualC T M') : ∃ L : LocPl ψ S T M M', L.q.v = q.v ∧ L.q.e = q.e ∧ ∀ x, prodDat (fun _ : Fin 1 => L) ((ofSplit q).up FS FT x) = Φ.1 x

theorem **exists_locPl_off** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hS : IsSelfDualC S M) (hT : IsSelfDualC T M') (Z : Finset (GlobalTheory.FinPl.Pl D)) : ∃ L : LocPl ψ S T M M', L.q.v ∉ Z ∧ ∃ x, prodDat (fun _ : Fin 1 => L) x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLocalModelSliceSourceLine

theorem **ι_e** (x : k) : p.e * p.ι x = p.ι x

theorem **ι_mul** (x y : k) : p.ι (x * y) = p.ι x * p.ι y

theorem **ι_mul_right** (x : k) (z : AE D) : p.ι x * z = p.ι (x * p.π z)

theorem **ι_mul_left** (x : k) (z : AE D) : z * p.ι x = p.ι (p.π z * x)

theorem **prH_apply** (a : D.host.E) : prH p a = p.pr a

theorem **map_ι_mul_rat** {l m o : Type} [Fintype m] (M : Matrix l m k) (N : Matrix m o D.host.E) : M.map p.ι * N.map (incA D) = (M * N.map (prH p)).map p.ι

theorem **rat_mul_map_ι** {l m o : Type} [Fintype m] (N : Matrix l m D.host.E) (M : Matrix m o k) : N.map (incA D) * M.map p.ι = (N.map (prH p) * M).map p.ι

theorem **qA_iA_k** : B.o.qA.map (prH p) * B.o.iA.map (prH p) = 1

theorem **qA_iB_k** : B.o.qA.map (prH p) * B.o.iB.map (prH p) = 0

theorem **β_eq** (h : p.IsCoreFrame B s FA F₀) : F₀.β = B.o.iA.map (prH p) * fromCols FA.β 0 + B.o.iB.map (prH p) * fromCols 0 (Matrix.of fun i (_ : Unit) => s * prH p (B.ζ i))

theorem **qA_β** (h : p.IsCoreFrame B s FA F₀) : B.o.qA.map (prH p) ∗ F₀.β = fromCols FA.β 0

theorem **γ_qA** (h : p.IsCoreFrame B s FA F₀) : FA.γ ∗ B.o.qA.map (prH p) = F₀.γ.toRows₁

theorem **γ_iA** (h : p.IsCoreFrame B s FA F₀) : F₀.γ ∗ B.o.iA.map (prH p) = fromRows FA.γ (0 : Matrix Unit (Fin B.A.rank) k)

theorem **up_slice** (h : p.IsCoreFrame B s FA F₀) (x : Matrix n n' k) : (B.o.resS S).symm (p.up FA FS x, 0) = p.up F₀ FS (fromCols x 0)

theorem **res_up** (h : p.IsCoreFrame B s FA F₀) (x : Matrix n (n' ⊕ Unit) k) : (B.o.resS S (p.up F₀ FS x)).1 = p.up FA FS x.toCols₁

theorem **tens_apply** (a : (n → k) → ℂ) (ha : a ∈ SB (n → k)) (Ξ : SB (Matrix n n' k)) (x : Matrix n (n' ⊕ Unit) k) : (tens a ha Ξ).1 x = Ξ.1 x.toCols₁ ∗ a fun r => x r (Sum.inr ())

theorem **symplO_zero_right''** (S T : HSp D) (x : WO S T) : symplO S T x 0 = 0

theorem **exists_isModel_slice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∀ (a : (n → k) → ℂ), a ∈ SB _ → ∀ v : WO B.L S, ∃ T₁, p.IsModel ψ FA FS T₁ ∧ ∀ (Ξ : SB (Matrix n n' k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ ∗ a fun r => x r (Sum.inr ())) → ∀ u, T₀ Ψ ((B.o.resS S).symm (u, v)) = T₁ Ξ u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLocalModelSliceSourceLineStatement

theorem **_root_.GlobalTheory.T.exists_local_model_slice_source_line** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∀ (a : (n → k) → ℂ), a ∈ SB _ → ∀ v : WO B.L S, ∃ T₁, p.IsModel ψ FA FS T₁ ∧ ∀ (Ξ : SB (Matrix n n' k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ ∗ a fun r => x r (Sum.inr ())) → ∀ u, T₀ Ψ ((B.o.resS S).symm (u, v)) = T₁ Ξ u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLocalModelSliceTargetLine

theorem **resT_symm_left** {T A B : HSp D} (d : OrthDec T A B) (S : HSp D) (u : WO S A) : (d.resT S).symm (u, 0) = GlobalTheory.MixModel.sandwich d.jA (1 : Mx D S.rank S.rank) u

theorem **isThetaFunO_sliceT_left** {ψ : AddChar (AF D.host.F) ℂ} {T A B S : HSp D} (d : OrthDec T A B) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (v : WO S B) : IsThetaFunO ψ S A fun u => f ((d.resT S).symm (u, v)) where

theorem **ι_mul** (a : AE D) (x : k) : p.ι (p.π a ∗ x) = a ∗ p.ι x

theorem **mul_map_ι** {m l n : Type} [Fintype l] (N : Matrix m l (AE D)) (M : Matrix l n k) : N ∗ M.map p.ι = (N.map p.π ∗ M).map p.ι

theorem **beta_eq** (hF : p.IsCoreFrame B s FA F₀) : F₀.β = fromCols (B.o.iA.map (prH p) ∗ FA.β) (Matrix.of fun r (_ : Unit) => s ∗ prH p (vecB B.o B.ζ r))

theorem **col_eq** : (Matrix.of fun r (_ : Unit) => s ∗ prH p (vecB B.o B.ζ r)) = s • (B.o.iB.map (prH p) ∗ Matrix.of fun j (_ : Unit) => prH p (B.ζ j))

theorem **qA_beta** (hF : p.IsCoreFrame B s FA F₀) : B.o.qA.map (prH p) ∗ F₀.β = fromCols FA.β 0

theorem **up_fromRows** (hF : p.IsCoreFrame B s FA F₀) (x : Matrix n' n k) : p.up FS F₀ (fromRows x 0) = (B.o.resT S).symm (p.up FS FA x, 0)

theorem **resT_up_fst** (hF : p.IsCoreFrame B s FA F₀) (x : Matrix (n' ⊕ Unit) n k) : (B.o.resT S (p.up FS F₀ x)).1 = p.up FS FA x.toRows₁

theorem **symplO_zero_right'** (S T : HSp D) (x : WO S T) : symplO S T x 0 = 0

theorem **tens_apply** (a : SB (n → k)) (Ξ : SB (Matrix n' n k)) (x : Matrix (n' ⊕ Unit) n k) : (tens a Ξ).1 x = Ξ.1 x.toRows₁ ∗ a.1 (x (Sum.inr ()))

theorem **GlobalTheory.SliceModelT.main** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ FS F₀ T₀ → ∀ (a : (n → k) → ℂ), a ∈ SB _ → ∀ v : WO S B.L, ∃ T₁, p.IsModel ψ FS FA T₁ ∧ ∀ (Ξ : SB (Matrix n' n k)) (Ψ : SB (Matrix (n' ⊕ Unit) n k)), (∀ x, Ψ.1 x = Ξ.1 x.toRows₁ ∗ a (x (Sum.inr ()))) → ∀ u, T₀ Ψ ((B.o.resT S).symm (u, v)) = T₁ Ξ u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLocalModelSliceTargetLineStatement

theorem **_root_.GlobalTheory.T.exists_local_model_slice_target_line** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ FS F₀ T₀ → ∀ (a : (n → k) → ℂ), a ∈ SB _ → ∀ v : WO S B.L, ∃ T₁, p.IsModel ψ FS FA T₁ ∧ ∀ (Ξ : SB (Matrix n' n k)) (Ψ : SB (Matrix (n' ⊕ Unit) n k)), (∀ x, Ψ.1 x = Ξ.1 x.toRows₁ ∗ a (x (Sum.inr ()))) → ∀ u, T₀ Ψ ((B.o.resT S).symm (u, v)) = T₁ Ξ u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLogNearOneUnitaryGroupOfFormLinearBound

theorem **entry_le_linfty** {n : ℕ} (X : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : ‖X i j‖ ≤ ‖X‖

theorem **linfty_le_of_entries** {n : ℕ} (M : Matrix (Fin n) (Fin n) ℂ) {δ : ℝ} (hδ : 0 ≤ δ) (h : ∀ i j, ‖M i j‖ ≤ δ) : ‖M‖ ≤ n ∗ δ

theorem **exp_conj_neg_eq** {n : ℕ} (U : (Matrix (Fin n) (Fin n) ℂ)ˣ) (X u : Matrix (Fin n) (Fin n) ℂ) (hX : NormedSpace.exp X = u) (hu : uᴴ ∗ (U : Matrix (Fin n) (Fin n) ℂ) ∗ u = U) : NormedSpace.exp (−(((U⁻¹ : (Matrix (Fin n) (Fin n) ℂ)ˣ) : Matrix (Fin n) (Fin n) ℂ) ∗ Xᴴ ∗ (U : Matrix (Fin n) (Fin n) ℂ))) = u

theorem **lie_of_fixed** {n : ℕ} (U : (Matrix (Fin n) (Fin n) ℂ)ˣ) (X : Matrix (Fin n) (Fin n) ℂ) (h : X = −(((U⁻¹ : (Matrix (Fin n) (Fin n) ℂ)ˣ) : Matrix (Fin n) (Fin n) ℂ) ∗ Xᴴ ∗ (U : Matrix (Fin n) (Fin n) ℂ))) : Xᴴ ∗ (U : Matrix (Fin n) (Fin n) ℂ) + (U : Matrix (Fin n) (Fin n) ℂ) ∗ X = 0

private lemma **linftyOp_norm_le_mul_of_entry_le** {n : ℕ} (A : Matrix (Fin n) (Fin n) ℂ) (δ : ℝ) (h : ∀ i j, ‖A i j‖ ≤ δ) : ‖A‖ ≤ n ∗ δ

private lemma **norm_entry_le_linftyOp_norm** {n : ℕ} (A : Matrix (Fin n) (Fin n) ℂ) (i j : Fin n) : ‖A i j‖ ≤ ‖A‖

theorem **_root_.GlobalTheory.UnitaryLog.exists_log_unitary** (n : ℕ) (G : Matrix (Fin n) (Fin n) ℂ) (hG : IsUnit G.det) : ∃ δ₁ C : ℝ, 0 < δ₁ ∧ ∀ (u : Matrix (Fin n) (Fin n) ℂ) (δ : ℝ), uᴴ ∗ G ∗ u = G → δ ≤ δ₁ → (∀ i j, ‖(u − 1) i j‖ ≤ δ) → ∃ X : Matrix (Fin n) (Fin n) ℂ, Xᴴ ∗ G + G ∗ X = 0 ∧ NormedSpace.exp X = u ∧ ∀ i j, ‖X i j‖ ≤ C ∗ δ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLogNearOneUnitaryGroupOfFormLinearBoundStatement

theorem **_root_.GlobalTheory.T.exists_log_near_one_unitaryGroup_of_form_linear_bound** : ∀ (n : ℕ) (G : Matrix (Fin n) (Fin n) ℂ), IsUnit G.det → ∃ δ₁ C : ℝ, 0 < δ₁ ∧ ∀ (u : Matrix (Fin n) (Fin n) ℂ) (δ : ℝ), uᴴ * G * u = G → δ ≤ δ₁ → (∀ i j, ‖(u - 1) i j‖ ≤ δ) → ∃ X : Matrix (Fin n) (Fin n) ℂ, Xᴴ * G + G * X = 0 ∧ NormedSpace.exp X = u ∧ ∀ i j, ‖X i j‖ ≤ C * δ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLowerUnipotentBlockDiagCosetRep

theorem **lowU_conj_blocks** {k : Type} [Field k] {A D : Type} [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (M W : Matrix D D k) (ν : Matrix D A k) (a : Matrix A A k) (b : Matrix A D k) (c : Matrix D A k) (e : Matrix D D k) : fromBlocks 1 0 0 M * fromBlocks 1 0 (-ν) 1 * fromBlocks a b c e * fromBlocks 1 0 0 W = fromBlocks a (b * W) (M * (c - ν * a)) (M * (e - ν * b) * W)

theorem **exists_lowU_diag_coset** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D) (R : Finset (GL D k)), IsRepsK (torusElt k D ϖ (indic s)) R → ∀ (κ κ' : GL D k → GL D k), (∀ g ∈ R, κ g ∈ KO k D ∧ κ' g ∈ KO k D ∧ g = κ g * torusElt k D ϖ (indic s) * κ' g) → ∀ j ∈ parahoric k A D, ∃ g ∈ R, ∃ ν₀ : Matrix D A k, (∀ i a, valuation k (ν₀ i a) < 1) ∧ ∀ ν : Matrix D A k, (∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν₀ - ν = g.1 * μ) → (lowU ν * emb₂ k (Equiv.refl (A ⊕ D)) (1, κ g) * mtor ϖ (indic s))⁻¹ * (j * mtor ϖ (indic s)) ∈ parahoric k A D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsLowerUnipotentBlockDiagCosetRepStatement

theorem **_root_.GlobalTheory.T.exists_lowerUnipotent_blockDiag_coset_rep** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (s : Finset D) (R : Finset (GL D k)), IsRepsK (torusElt k D ϖ (indic s)) R → ∀ (κ κ' : GL D k → GL D k), (∀ g ∈ R, κ g ∈ KO k D ∧ κ' g ∈ KO k D ∧ g = κ g * torusElt k D ϖ (indic s) * κ' g) → ∀ j ∈ parahoric k A D, ∃ g ∈ R, ∃ ν₀ : Matrix D A k, (∀ i a, valuation k (ν₀ i a) < 1) ∧ ∀ ν : Matrix D A k, (∃ μ : Matrix D A k, (∀ i a, valuation k (μ i a) < 1) ∧ ν₀ - ν = g.1 * μ) → (lowU ν * emb₂ k (Equiv.refl (A ⊕ D)) (1, κ g) * mtor ϖ (indic s))⁻¹ * (j * mtor ϖ (indic s)) ∈ parahoric k A D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMaxCompactRealPointsOfOrthonormalFrames

theorem **comps_unit** (R : RealPoints θ V) (h : IsFrameFam V ε fr) (k : ∀ τ, Formalism.REAL.KG (ε τ)) : sigInf V (comps fr k) * comps fr k = 1

theorem **continuous_comps** : Continuous (comps (ε

theorem **frameHom_spec** (R : RealPoints θ V) (h : IsFrameFam V ε fr) (k : ∀ τ, Formalism.REAL.KG (ε τ)) : frameHom R h k ∈ GinfC V ∧ ∀ τ, infMat D.sp.cm.sig V.gram (θ τ) (frameHom R h k) = (fr τ)⁻¹ * Formalism.REAL.toMK (k τ) * fr τ

theorem **frameHom_single** (hθ : IsInfFam θ) (R : RealPoints θ V) (h : IsFrameFam V ε fr) (τ : D.what₂.Tau) (k : Formalism.REAL.KG (ε τ)) : frameHom R h (Pi.mulSingle τ k) = kAt θ V τ (ε τ) (fr τ) k

theorem **isCompact_frameRange** (hθ : IsInfFam θ) (R : RealPoints θ V) (h : IsFrameFam V ε fr) : IsCompact ((frameHom R h).range : Set (GA D V))

theorem **conj_mem_Kset** (hθ : IsInfFam θ) (R : RealPoints θ V) (h : IsFrameFam V ε fr) (K' : Subgroup (GA D V)) (hc : IsCompact (K' : Set (GA D V))) (hle : (frameHom R h).range ≤ K') (τ : D.what₂.Tau) {g : GA D V} (hg : g ∈ K') : fr τ * infMat D.sp.cm.sig V.gram (θ τ) g * (fr τ)⁻¹ ∈ Formalism.REAL.Kset (ε τ)

theorem **isMaxCompactInf_range** (hθ : IsInfFam θ) (R : RealPoints θ V) (h : IsFrameFam V ε fr) : IsMaxCompactInf V (frameHom R h).range

theorem **exists_maxCompact_of_frames** : ∀ (D : D7) (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)), IsInfFam θ → ∀ (V : HSp D), ∀ (ε : D.what₂.Tau → Fin V.rank → Bool) (fr : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ), IsFrameFam V ε fr → ∃ K : Subgroup (GA D V), IsMaxCompactInf V K ∧ (∃ j : (∀ τ, Formalism.REAL.KG (ε τ)) →* K, Function.Surjective j ∧ ∀ τ k, (j (Pi.mulSingle τ k) : GA D V) = kAt θ V τ (ε τ) (fr τ) k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMaxCompactRealPointsOfOrthonormalFramesStatement

theorem **_root_.GlobalTheory.T.exists_maxCompact_realPoints_of_orthonormalFrames** : ∀ (D : D7) (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)), IsInfFam θ → ∀ (V : HSp D), ∀ (ε : D.what₂.Tau → Fin V.rank → Bool) (fr : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ), IsFrameFam V ε fr → ∃ K : Subgroup (GA D V), IsMaxCompactInf V K ∧ (∃ j : (∀ τ, Formalism.REAL.KG (ε τ)) →* K, Function.Surjective j ∧ ∀ τ k, (j (Pi.mulSingle τ k) : GA D V) = kAt θ V τ (ε τ) (fr τ) k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMaximalCompactUnnComponentStandardCartanIn

theorem **block_matrix_map** {S : Type} [CommRing S] (f : R →+* S) (a b c d : Matrix (Fin n) (Fin n) R) : (blk4 a b c d).map f = blk4 (a.map f) (b.map f) (c.map f) (d.map f)

theorem **blkJ_mul** : (blk4 (0 : Matrix (Fin n) (Fin n) R) 1 (-1) 0) * blk4 0 1 (-1) 0 = -1

theorem **blkJ_transpose** : (blk4 (0 : Matrix (Fin n) (Fin n) R) 1 (-1) 0)ᵀ = -blk4 0 1 (-1) 0

theorem **blkJ_map** {S : Type} [CommRing S] (f : R →+* S) : (blk4 (0 : Matrix (Fin n) (Fin n) R) 1 (-1) 0).map f = blk4 0 1 (-1) 0

theorem **mixJ_mul** (a j a' j' : R) : mixJ n a j * mixJ n a' j' = mixJ n (a * a' - j * j') (a * j' + j * a')

theorem **mixJ_one** : mixJ n (1 : R) 0 = 1

theorem **mixJ_J** : mixJ n (0 : R) 1 = blk4 0 1 (-1) 0

theorem **mixJ_map** {S : Type} [CommRing S] (f : R →+* S) (a j : R) : (mixJ n a j).map f = mixJ n (f a) (f j)

theorem **mixJ_transpose** (a j : R) : (mixJ n a j)ᵀ = mixJ n a (-j)

theorem **cT_mixJ** (σ : R →+* R) (a j : R) : Formalism.UG.cT σ (mixJ n a j) = mixJ n (σ a) (-σ j)

theorem **phase** (A B t : ℂ) (hB : B ≠ 0) (h1 : A * B * conj B = t) (h2 : A * conj A = 4) : A * (1 / 2) = t / (‖t‖ : ℂ)

theorem **four_iff** {S : Type} [CommRing S] (f : R →+* S) (z : R) : f (z - 4) = 0 ↔ f z = 4

theorem **four_left** {S : Type} [CommRing S] (f : R →+* S) {e : R} (z : R) (h : f e = 0) : f (e * (z - 4)) = 0

theorem **four_right** {S : Type} [CommRing S] (f : R →+* S) (e : R) {z : R} (h : f z = 4) : f (e * (z - 4)) = 0

theorem **alg_phase** (c : R →+* R) (hc : ∀ z, c (c z) = z) {e h m : R} {u ε : Rˣ} (he : e * e = e) (hce : c e = e) (hε : (ε : R) * u * c u = m) (hm : c m = -m) (hh : 2 * h = 1) (h4 : e * (ε * c ε - 4) = 0) : e * (e * ε * h) = e * ε * h ∧ (e * ε * h) * (e * ε * h) = -e ∧ c (e * ε * h) = -(e * ε * h)

theorem **cx_phase** (c : R →+* R) (φ : R →+* ℂ) (hφ : ∀ z, φ (c z) = conj (φ z)) {e h m : R} {u ε : Rˣ} (he : φ e = 1) (hε : (ε : R) * u * c u = m) (hh : 2 * h = 1) (h4 : φ (ε * c ε) = 4) : φ (e * ε * h) = φ m / (‖φ m‖ : ℂ)

theorem **mixJ_real** (φ : R →+* ℂ) {e j : R} (he : φ e = 1) : (mixJ n (1 − e) j).map φ = φ j • blk4 0 1 (−1) 0

theorem **sigR_sigR** (A : Type) [CommRing A] [Algebra F A] (hσ : ∀ x, σ (σ x) = x) (z : A ⊗[F] E) : sigR A σ (sigR A σ z) = z

theorem **sigR_compInf** (w : InfinitePlace F) (z : AF F ⊗[F] E) : sigR w.Completion σ (compInf F E w z) = compInf F E w (sigR (AF F) σ z)

theorem **incl_half** [CharZero E] (A : Type) [CommRing A] [Algebra F A] : 2 ∗ inclR (F

theorem **exists_point** (n : ℕ) {e j : AF F ⊗[F] E} (hee : e ∗ e = e) (hej : e ∗ j = j) (hjj : j ∗ j = −e) (hce : sigR (AF F) σ e = e) (hcj : sigR (AF F) σ j = −j) (hef : ∀ v, compFin F E v e = 0) : ∃ k ∈ levelAll σ (blk4 (0 : Matrix (Fin n) (Fin n) E) 1 (−1) 0) (stdLattice E (n + n)) 1, (AdelicU.matHom _ σ _ k).1 = mixJ n (1 − e) j

theorem **real_four** (D : D7) (θ : InfFam D) (hθ : IsInfFam θ) (z : AE D) (hz : ∀ w, compInf D.host.F D.host.E w z = 4) (τ : D.what₂.Tau) : ((θ τ).comp (infPart D)) z = 4

theorem **idem_four** (D : D7) (θ : InfFam D) (hθ : IsInfFam θ) {e z : AE D} (he : finPart D e = 0) (hz : ∀ τ, ((θ τ).comp (infPart D)) z = 4) : e ∗ (z − 4) = 0

theorem **cA_skew** (D : D7) (μ : D.host.E) (hμ : IsSkew μ) : cA D (incA D μ) = −incA D μ

theorem **cA_cA** (D : D7) (z : AE D) : cA D (cA D z) = z

theorem **exists_phase_adele** (D : D7) (θ : InfFam D) (hθ : IsInfFam θ) (μ : D.host.E) (hμ : IsSkew μ) : ∃ e j : AE D, e ∗ e = e ∧ e ∗ j = j ∧ j ∗ j = −e ∧ cA D e = e ∧ cA D j = −j ∧ (∀ v, compFin D.host.F D.host.E v e = 0) ∧ (∀ τ, ((θ τ).comp (infPart D)) e = 1) ∧ ∀ τ, ((θ τ).comp (infPart D)) j = tauE D τ μ / (‖tauE D τ μ‖ : ℂ)

theorem **exists_mem_K_compAt_eq_stdInv** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (n : ℕ) (μ : D.host.E), IsSkew μ → ∃ k ∈ K D n, ∀ τ, compAt θ τ (mat k) = stdInv n μ τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMaximalCompactUnnComponentStandardCartanInStatement

theorem **_root_.GlobalTheory.T.exists_maximalCompactUnn_component_standardCartanInv** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (n : ℕ) (μ : D.host.E), IsSkew μ → ∃ k ∈ K D n, ∀ τ, compAt θ τ (mat k) = stdInv n μ τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMemMultT

theorem **finN_mem_UPts** (g : GA D V) : finN V hn g ∈ Formalism.UG.UPts D.sp.cm.sig (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)

theorem **finN_eq** (g : GA D V) (y : GL (Fin n) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (h : ∀ i j, (Formalism.UG.bc (Lev.swapAE D) (finHom V g)).1 i j = y.1 (finCongr hn i) (finCongr hn j)) : finN V hn g = y

theorem **exists_rat'** (hθ : IsInfFam θ) {γ : GA D V} (hγ : γ ∈ GQ V) : ∃ δ ∈ Formalism.UG.UJ (D.sp.cm.sig : D.host.E →+* D.host.E) (V.gram.reindex (finCongr hn) (finCongr hn)), finN V hn γ = Formalism.UG.bc (Formalism.UG.inclE (F

theorem **infN_atTau** (hθ : IsInfFam θ) (τ : D.what₂.Tau) {g : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hg : g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ)) (ρ : D.what₂.Tau) : (infN θ V hn (Equiv.refl _) (atTau θ V τ g) ρ : Matrix (Fin n) (Fin n) ℂ) = (Function.update (fun _ => (1 : Matrix (Fin V.rank) (Fin V.rank) ℂ)) τ g ρ).reindex (finCongr hn) (finCongr hn)

theorem **continuous_finN** : letI

theorem **finPt_mul** (h k : GA D D.sp.X') : (finPt (h ∗ k)).1 = (finPt h).1 ∗ (finPt k).1

theorem **finPt_one** : (finPt (1 : GA D D.sp.X')) = 1

theorem **finPt_of_hasMat** {h : GA D D.sp.X'} {y} (hh : HasMat h y) : (finPt h).1 = y

theorem **form_level** (hg : g ∈ AdForms K.lev (embT D) (sgT u)) (h k : GA D D.sp.X') (κ : K.lev.U) (hk : (finPt k).1 = κ.1) : g (finPt (h ∗ k)) = (K.lev.chi κ : ℂ) • g (finPt h)

theorem **form_left** (hθ : IsInfFam θ) (hg : g ∈ AdForms K.lev (embT D) (sgT u)) {γ : GA D D.sp.X'} (hγ : γ ∈ GQ D.sp.X') (h : GA D D.sp.X') : lT _ (infPt θ γ) (g (finPt (γ ∗ h))) = g (finPt h)

theorem **form_locally** (hK : letI

theorem **finPt_mul_inf** (h : GA D D.sp.X') {k : GA D D.sp.X'} (hk : k ∈ GinfC D.sp.X') : finPt (h ∗ k) = finPt h

theorem **infPt_of_fin** {k : GA D D.sp.X'} (hk : k ∈ GfC D.sp.X') : infPt θ k = 1

theorem **valAt_mem** (hg : g ∈ AdForms K.lev (embT D) (sgT u)) (h : GA D D.sp.X') : valAt θ g h ∈ ValC (gramC (gramP D) (embT D)) (sgT u)

theorem **valAt_level** (hg : g ∈ AdForms K.lev (embT D) (sgT u)) (h : GA D D.sp.X') {k : GA D D.sp.X'} {κ : K.lev.U} (hk : HasMat k κ.1) : valAt θ g (h ∗ k) = (K.lev.chi κ : ℂ) • valAt θ g h

theorem **valAt_of_hasMat** {h : GA D D.sp.X'} {y : Formalism.UG.UPts (sigD D) (gramP D) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)} (hh : HasMat h y.1) : valAt θ g h = g y

theorem **valAt_atTau** (hθ : IsInfFam θ) (h : GA D D.sp.X') (τ : D.what₂.Tau) {a : Matrix (Fin D.sp.X'.rank) (Fin D.sp.X'.rank) ℂ} (ha : a ∈ Formalism.QuantumDeterminants.UH (gramAt D D.sp.X' τ)) (y y' : FrP D) (hy : IsTr (fun ρ => (Function.update (fun _ => (1 : Matrix (Fin D.sp.X'.rank) (Fin D.sp.X'.rank) ℂ)) τ a ρ).reindex (finCongr (rankP D)) (finCongr (rankP D))) y' y) : valAt θ g h y = valAt θ g (h ∗ atTau θ D.sp.X' τ a) y'

theorem **isAutFunSm_pair** (hθ : IsInfFam θ) (hK : letI

theorem **exists_mem_multT** : ∀ (D : D7) (u : D.what₂.Word) (K : ClsPTied u) (θ : InfFam D), IsInfFam θ → ∀ (S : D.what₂.Tau → SPar) (M : DSModel D.sp.X' S) (j : SmT M →ₗ[ℂ] (FrP D → ℂ)), IsJ u M j → ∀ (J : Set (GA D D.sp.X')) (χ : GA D D.sp.X' → ℂ), TopLev K J χ → (letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMoveSelfDualLatticeSplitFrames

theorem **exists_selfDual_frames** : ∀ (D : D7) (S : HSp D) (L₀ : GlobalTheory.FinPl.Lat S), IsSelfDualC S L₀ → ∀ (I : Type) [Finite I] (K : I → Type) [∀ i, Field (K i)] [∀ i, ValuativeRel (K i)] [∀ i, TopologicalSpace (K i)] [∀ i, IsNonarchimedeanLocalField (K i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (K i)) (Y : I → Type) [∀ i, Fintype (Y i)] [∀ i, DecidableEq (Y i)] (F : ∀ i, GlobalTheory.LocMod.Frame (K i) S.rank (Y i)), (∀ i i', (q i).v = (q i').v → i = i') → ∃ L : GlobalTheory.FinPl.Lat S, IsSelfDualC S L ∧ ∀ i, IntIn (q i) S (F i) L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMoveSelfDualLatticeSplitFramesStatement

theorem **_root_.GlobalTheory.T.exists_move_selfDual_lattice_split_frames** : ∀ (D : D7) (S : HSp D) (L₀ : GlobalTheory.FinPl.Lat S), IsSelfDualC S L₀ → ∀ (I : Type) [Finite I] (K : I → Type) [∀ i, Field (K i)] [∀ i, ValuativeRel (K i)] [∀ i, TopologicalSpace (K i)] [∀ i, IsNonarchimedeanLocalField (K i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (K i)) (Y : I → Type) [∀ i, Fintype (Y i)] [∀ i,

DecidableEq (Y i)] (F : ∀ i, GlobalTheory.LocMod.Frame (K i) S.rank (Y i)), (∀ i i', (q i).v = (q i').v → i = i') → ∃ L : GlobalTheory.FinPl.Lat S, IsSelfDualC S L ∧ ∀ i, IntIn (q i) S (F i) L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsMovedNotProportional

theorem **exists_unram_char** (D : D7) (n : ℕ) : ∃ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ ∧ ResEta D χ n ∧ UnramEverywhere D χ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsN1LocalHomMemOpen

theorem **autOf_apply** (e : A ⊕ Fin c ≃ Fin (M + c')) (g : GL (A ⊕ Fin c) E) (v : Fin (M + c') → E) : autOf e g v = (g.val.submatrix e.symm e.symm).mulVec v

theorem **autOf_inv_apply** (e : A ⊕ Fin c ≃ Fin (M + c')) (g : GL (A ⊕ Fin c) E) (v : Fin (M + c') → E) : (autOf e g)⁻¹ v = ((g⁻¹).val.submatrix e.symm e.symm).mulVec v

theorem **lev_symm** (e : A ⊕ Fin c ≃ Fin (M + c')) (hcc : c' = c) (he : ∀ i : Fin c, (e (Sum.inr i)).val = M + i.val) (l : Fin (M + c')) : lev (n

theorem **mulVec_fix** (e : A ⊕ Fin c ≃ Fin (M + c')) (m : Matrix (A ⊕ Fin c) (A ⊕ Fin c) E) (hm : ∀ r s, lev 1 s ≤ lev 1 r → m r s = (1 : Matrix (A ⊕ Fin c) (A ⊕ Fin c) E) r s) (v : Fin (M + c') → E) (l : Fin (M + c')) (hv : ∀ j, v j ≠ 0 → lev (n

theorem **mem_normOneIdeles** (e : A ⊕ Fin c ≃ Fin (M + c')) (hcc : c' = c) (he : ∀ i : Fin c, (e (Sum.inr i)).val = M + i.val) (g : GL (A ⊕ Fin c) E) (hg : g ∈ N E A c 1) : autOf e g ∈ Loc.N1 E M c'

theorem **locHom_autOf** {k : Type} [Field k] (ι : E →+* k) (e : A ⊕ Fin c ≃ Fin (M + c')) (g : GL (A ⊕ Fin c) E) : (locHom ι e (autOf e g)).val = g.val.map ι

theorem **exists_N1_local_hom_mem_open** : ∀ {E k : Type} [Field E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : E →+* k), DenseRange ι → ∀ {A : Type} [Fintype A] [DecidableEq A] {M c c' : ℕ} (e : A ⊕ Fin c ≃ Fin (M + c')), c' = c → (∀ i : Fin c, (e (Sum.inr i)).val = M + i.val) → ∀ u ∈ N k A c 1, ∀ V : Set (GL (A ⊕ Fin c) k), IsOpen V → u ∈ V → ∃ x ∈ Loc.N1 E M c', locHom ι e x ∈ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsN1LocalHomMemOpenStatement

theorem **_root_.GlobalTheory.T.exists_N1_locHom_mem_open** : ∀ {E k : Type} [Field E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : E →+* k), DenseRange ι → ∀ {A : Type} [Fintype A] [DecidableEq A] {M c c' : ℕ} (e : A ⊕ Fin c ≃ Fin (M + c')), c' = c → (∀ i : Fin c, (e (Sum.inr i)).val = M + i.val) → ∀ u ∈ N k A c 1, ∀ V : Set (GL (A ⊕ Fin c) k), IsOpen V → u ∈ V → ∃ x ∈ Loc.N1 E M c', locHom ι e x ∈ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNEmbXInvMulMemKOne

theorem **exists_N_embX_inv_mul_mem_KOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (s κ : ℕ) (S : GlobalTheory.SymChain.SymData k s κ), ∀ x ∈ S.P.J, ∃ (n : GL (Fin κ ⊕ Fin (s + 1)) k) (γ : GL (Fin κ) k), n ∈ GlobalTheory.RootEx.N k (Fin κ) (s + 1) (S.ρ + 1) ∧ n ∈ S.P.J ∧ GlobalTheory.RootEx.embX (s + 1) γ ∈ S.P.J ∧ (n * GlobalTheory.RootEx.embX (s + 1) γ)⁻¹ * x ∈ GlobalTheory.SplitWeil.KOne k (Fin κ ⊕ Fin (s + 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNEmbXInvMulMemKOneStatement

theorem **_root_.GlobalTheory.T.exists_N_embX_inv_mul_mem_KOne'** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (s κ : ℕ) (S : GlobalTheory.SymChain.SymData k s κ), ∀ x ∈ S.P.J, ∃ (n : GL (Fin κ ⊕ Fin (s + 1)) k) (γ : GL (Fin κ) k), n ∈ GlobalTheory.RootEx.N k (Fin κ) (s + 1) (S.ρ + 1) ∧ n ∈ S.P.J ∧ GlobalTheory.RootEx.embX (s + 1) γ ∈ S.P.J ∧ (n * GlobalTheory.RootEx.embX (s + 1) γ)⁻¹ * x ∈ GlobalTheory.SplitWeil.KOne k (Fin κ ⊕ Fin (s + 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNatEqMul

theorem **exists_nat_eq_mul** {E : Type} [Field E] [NumberField E] {c : 𝒪 E} (hc : c ≠ 0) : ∃ n : ℕ, n ≠ 0 ∧ ∃ c' : 𝒪 E, (n : 𝒪 E) = c * c'

theorem **exists_nat_bounds** {E : Type} [Field E] [NumberField E] {I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E} (hI : I ≠ 0) : ∃ N M : ℕ, N ≠ 0 ∧ M ≠ 0 ∧ (∀ a ∈ I, ∃ b : 𝒪 E, (N : E) * a = algebraMap (𝒪 E) E b) ∧ (M : E) ∈ I

theorem **eventually_notMem** {F : Type} [Field F] [NumberField F] {N : ℕ} (hN : N ≠ 0) : ∀ᶠ v : HeightOneSpectrum (𝒪 F) in cofinite, (N : 𝒪 F) ∉ v.asIdeal

theorem **inv_mem_integers** {F : Type} [Field F] [NumberField F] {N : ℕ} {v : HeightOneSpectrum (𝒪 F)} (hv : (N : 𝒪 F) ∉ v.asIdeal) : algebraMap F (Fv F v) (N : F)⁻¹ ∈ v.adicCompletionIntegers F

theorem **smul_inv_mem** {F : Type} [Field F] [NumberField F] {N : ℕ} {v : HeightOneSpectrum (𝒪 F)} (hv : (N : 𝒪 F) ∉ v.asIdeal) {α : Af F} (hα : α ∈ intAf F) (h0 : ∀ w, ¬ w = v → evalV F w α = 0) : (N : F)⁻¹ • α ∈ intAf F ∧ ∀ w, ¬ w = v → evalV F w ((N : F)⁻¹ • α) = 0

theorem **pairA_smul** {F : Type} [Field F] [NumberField F] (c : F) (α : Af F) : pairA F 0 (c • α) = c • pairA F 0 α

theorem **tmul_eq** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] {N : ℕ} (hN : N ≠ 0) (α : Af F) (z : E) : pairA F 0 α ⊗ₜ[F] z = pairA F 0 ((N : F)⁻¹ • α) ⊗ₜ[F] ((N : E) * z)

theorem **latA_eq_latV** (D : D7) : ∀ᶠ v : Pl D in cofinite, ∀ (S : HSp D) (L : Lat S), latA S L (· = v) = latV S L (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNatMulSelfDualLatticeAtPlaceIntegral

theorem **exists_nat_mul_isIntegral** (z : D.host.E) : ∃ n : ℕ, 0 < n ∧ IsIntegral ℤ ((n : D.host.E) * z)

theorem **exists_nat_mul_isIntegral_fin** {n : ℕ} (z : Fin n → Fin n → D.host.E) : ∃ d : ℕ, 0 < d ∧ ∀ l k, IsIntegral ℤ ((d : D.host.E) * z l k)

theorem **finPart_incA_mem** {o : D.host.E} (ho : IsIntegral ℤ o) : finPart D (incA D o) ∈ intAE D

theorem **hermAt_single** (S : HSp D) (y : Fin S.rank → AE D) (k : Fin S.rank) (a : AE D) : hermAt S y (Pi.single k a) = cA D a * (gA S *ᵥ y) k

theorem **exists_mul_mem_intAt** : ∀ (D : D7) (S : HSp D) (v : Pl D) (Λ : AddSubgroup (Fin S.rank → AE D)), IsSDAt S v Λ → ∃ N : ℕ, 0 < N ∧ ∀ y ∈ Λ, ∀ i, (N : AE D) * y i ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNatMulSelfDualLatticeAtPlaceIntegralStatement

theorem **_root_.GlobalTheory.T.exists_nat_mul_selfDualLatticeAtPlace_integral** : ∀ (D : D7) (S : HSp D) (v : Pl D) (Λ : AddSubgroup (Fin S.rank → AE D)), IsSDAt S v Λ → ∃ N : ℕ, 0 < N ∧ ∀ y ∈ Λ, ∀ i, (N : AE D) * y i ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNatMulSplitHalfLatticeCoordinatesIntegral

theorem **exists_nat_mul_mem** (x : k) : ∃ N : ℕ, 0 < N ∧ (N : k) * x ∈ 𝒪[k]

theorem **exists_nat_mul_mem_family** {ι : Type} [Fintype ι] (f : ι → k) : ∃ N : ℕ, 0 < N ∧ ∀ i, (N : k) * f i ∈ 𝒪[k]

theorem **GlobalTheory.FinPl.RdBound.exists_nat_mul_rd_mem** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∃ N : ℕ, 0 < N ∧ ∀ u ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v), IsHalf p.e u → ∀ i j, (N : k) * rd α β u i j ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNatMulSplitHalfLatticeCoordinatesIntegralStatement

theorem **_root_.GlobalTheory.T.exists_nat_mul_splitHalf_latticeCoordinates_integral** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∃ N : ℕ, 0 < N ∧ ∀ u ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v), IsHalf p.e u → ∀ i j, (N : k) * rd α β u i j ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNeZeroOfEventually

theorem **exists_nat_mem_mul_integer** (c : k) (hc : c ≠ 0) : ∃ n : ℕ, 0 < n ∧ c⁻¹ * (n : k) ∈ 𝒪[k]

theorem **exists_ne_zero_of_eventually** {P : k → Prop} (h : ∀ᶠ s in nhds (0 : k), P s) : ∃ c : k, c ≠ 0 ∧ P c

theorem **sb_lattice_bounds** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [CharZero k] (Y Y' : Type) [Fintype Y] [Fintype Y'] (Φ : Matrix Y Y' k → ℂ), Φ ∈ SB (Matrix Y Y' k) → ∃ n : ℕ, 0 < n ∧ (∀ u : Matrix Y Y' k, (∀ r q, u r q ∈ 𝒪[k]) → ∀ t, Φ (t + n • u) = Φ t) ∧ ∀ t, Φ t ≠ 0 → ∀ r q, (n • t) r q ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNonzeroJetScaleTrueColumns

theorem **exists_jetS** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ ψ θ, IsStdChar (I.rec7 l Z) ψ (-1) → IsInfFam θ → ∃ s, JetS I l Z ψ θ s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNonzeroJetScaleTrueColumnsStatement

theorem **_root_.GlobalTheory.T.exists_nonzero_jetScale_trueColumns** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ ψ θ, IsStdChar (I.rec7 l Z) ψ (-1) → IsInfFam θ → ∃ s, JetS I l Z ψ θ s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNonzeroLocalScalars

theorem **theta_comb** (hψ : IsStdChar D ψ (-1)) {c : WO S T →₀ ℂ} (hc : IsFinComb c) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (comb ψ S T c f)

theorem **comb_smul** (c : WO S T →₀ ℂ) (a : ℂ) (f : WO S T → ℂ) : comb ψ S T c (a • f) = a • comb ψ S T c f

theorem **comb_zero** (c : WO S T →₀ ℂ) : comb ψ S T c 0 = 0

theorem **comb_pureTensorO** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (sS sT BS BT) {e : WO S T →₀ ℂ} (he : IsFinComb e) {f₀ : WO S T → ℂ} (hf₀ : IsThetaFunO ψ S T f₀) (P) : comb ψ S T e (pureTensorO ψ θ S T sS sT BS BT f₀ P) = pureTensorO ψ θ S T sS sT BS BT (comb ψ S T e f₀) P

theorem **fock** {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hθ : IsInfFam θ) (hW : IsWeilDatumO ψ S T χS χT om) (sS sT BS BT) {d o e : WO S T →₀ ℂ} {g : WO S T → ℂ} {s : ℂ} (hd : IsFinComb d) (ho : IsFinComb o) (he : IsFinComb e) (hg : IsThetaFunO ψ S T g) {b : GA D S} (hb : b ∈ GfC S) {b' : GA D T} (hb' : b' ∈ GfC T) (h : om (b, b') (comb ψ S T d g) = s • comb ψ S T e (comb ψ S T o (om (b, b') g))) (P) {a : GA D S} (ha : a ∈ GinfC S) {a' : GA D T} (ha' : a' ∈ GinfC T) : om (a * b, a' * b') (comb ψ S T d (pureTensorO ψ θ S T sS sT BS BT g P)) = s • comb ψ S T e (comb ψ S T o (om (a * b, a' * b') (pureTensorO ψ θ S T sS sT BS BT g P)))

theorem **pureTensorO_zero** (sS sT BS BT) (P) : pureTensorO ψ θ S T sS sT BS BT (0 : WO S T → ℂ) P = 0

theorem **vac_mono** {LS : GlobalTheory.FinPl.Lat S} {LT : GlobalTheory.FinPl.Lat T} (P : GlobalTheory.FinPl.Pl D → Prop) {f : WO S T → ℂ} (h : GlobalTheory.FinPl.Vac ψ S T LS LT (fun _ => True) f) : GlobalTheory.FinPl.Vac ψ S T LS LT P f

theorem **datum_mid** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j, j + 1 < D.R.m → ∀ d₀, IsDG W j ((pk W).MY (j + 1)) ((pk W).ΛY (j + 1)) ((pk W).βY (j + 1)) ((pk W).vac j) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D j)) (M' : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (o' : WO (Yc D j) (Yc D (j + 1)) →₀ ℂ), IsSelfDualC (Yc D j) M ∧ IsSelfDualC (Yc D (j + 1)) M' ∧ GlobalTheory.FinPl.CombAt (· ∈ v.gP) o' ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M M' (· ∈ v.gP) ((pk W).ωY W.P.A3.A.μ0 j (f.g.b j, f.g.b (j + 1)) ((pk W).vac j)) ∧ Global

theorem **datum_top** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∀ j (h : j + 2 * 0 + 2 = D.R.m), ∀ d₀, IsDG W (j + 1) (pk W).MX (pk W).ΛX (pk W).βX (vacT (pk W) j h) d₀ → ∃ (M : GlobalTheory.FinPl.Lat (Yc D (j + 1))) (M' : GlobalTheory.FinPl.Lat D.sp.X') (o' : WO (Yc D (j + 1)) D.sp.X' →₀ ℂ), IsSelfDualC (Yc D (j + 1)) M ∧ IsSelfDualC D.sp.X' M' ∧ GlobalTheory.FinPl.CombAt (· ∈ v.gP) o' ∧ GlobalTheory.FinPl.Vac f.c.ψ _ _ M M' (· ∈ v.gP) (omT (pk W) W.P.A3.A.μ0 j h (f.g.b (j + 1), f.g.bT) (vacT (pk W) j h)) ∧ GlobalTheory.

theorem **scalar** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M N : GlobalTheory.FinPl.Lat S) (M' N' : GlobalTheory.FinPl.Lat T) (Q : Finset (GlobalTheory.FinPl.Pl D)) (o o' : WO S T →₀ ℂ) (g₀ : WO S T → ℂ), IsStdChar D ψ (-1) → IsSelfDualC S M → IsSelfDualC T M' → IsSelfDualC S N → IsSelfDualC T N' → GlobalTheory.FinPl.CombAt (· ∈ Q) o → GlobalTheory.FinPl.CombAt (· ∈ Q) o' → GlobalTheory.FinPl.Vac ψ S T M M' (· ∈ Q) g₀ → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ Q) (comb ψ S T o g₀) → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ Q) (comb ψ S T o' g₀) → comb ψ S T o' g₀ ≠ 0 → ∃ c : ℂ, comb ψ S T o g₀ = c • comb ψ S T o' g₀

theorem **weil_top** (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) : ∃ χS χT, IsWeilDatumO (pk W).ψ (Yc D (j + 1)) D.sp.X' χS χT (omT (pk W) W.P.A3.A.μ0 j h)

theorem **main** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∃ s : ℕ → ℂ, (∀ j, s j ≠ 0) ∧ GVec f v s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNonzeroLocalScalarsStatement

theorem **_root_.GlobalTheory.T.exists_nonzero_local_scalars** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u), PkHyp W → f.RowOK → ∀ v : VCom f.c, v.Is → VTie f v → ∀ (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GPin G → GRead f G pl → ∃ s : ℕ → ℂ, (∀ j, s j ≠ 0) ∧ GVec f v s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNonzeroMoveConstant

theorem **vacuum_reads_base** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z) (u : B.Word) (A : ColDat I l Z u (w.C.herm.base.Ch u)) ψ θ, A.Is (w.C.forms u) ψ θ → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m), ∃ c₀ : ℂ, c₀ ≠ 0 ∧ A.MoveAt θ i hi c₀ 1 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNonzeroMoveConstantStatement

theorem **_root_.GlobalTheory.T.exists_nonzero_move_constant** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z) (u : B.Word) (A : ColDat I l Z u (w.C.herm.base.Ch u)) ψ θ, A.Is (w.C.forms u) ψ θ → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m), ∃ c₀ : ℂ, c₀ ≠ 0 ∧ A.MoveAt θ i hi c₀ 1 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormNearNegOneAtRealPlace

theorem **ext_algebraMap** {K : Type} [Field K] [NumberField K] (v : InfinitePlace K) (x : K) : InfinitePlace.Completion.extensionEmbedding v (algebraMap K v.Completion x) = v.embedding x

theorem **tendsto_completion_iff** {K : Type} [Field K] [NumberField K] (v : InfinitePlace K) {α : Type} {l : Filter α} (f : α → K) (c : v.Completion) : Tendsto (fun x => algebraMap K v.Completion (f x)) l (𝓝 c) ↔ Tendsto (fun x => v.embedding (f x)) l (𝓝 (InfinitePlace.Completion.extensionEmbedding v c))

theorem **mk_eq_mk_iff_of_isReal** {K : Type} [Field K] [NumberField K] {j σ : K →+* ℂ} (hj : ComplexEmbedding.IsReal j) : InfinitePlace.mk σ = InfinitePlace.mk j ↔ σ = j

theorem **tendsto_embedding** {K : Type} [Field K] [NumberField K] {j : K →+* ℂ} (hj : ComplexEmbedding.IsReal j) {l : Filter K} (hl : ∀ u : InfinitePlace K, Tendsto (fun x => algebraMap K u.Completion x) l (𝓝 (if u = InfinitePlace.mk j then -1 else 1))) (σ : K →+* ℂ) : Tendsto (fun x => σ x) l (𝓝 (if σ = j then -1 else 1))

theorem **tendsto_norm** {E K : Type} [Field E] [NumberField E] [Field K] [NumberField K] [Algebra E K] {j : K →+* ℂ} (hj : ComplexEmbedding.IsReal j) {l : Filter K} (hσ : ∀ σ : K →+* ℂ, Tendsto (fun x => σ x) l (𝓝 (if σ = j then -1 else 1))) (u : InfinitePlace E) : Tendsto (fun x => u.embedding (Algebra.norm E x)) l (𝓝 (if u = InfinitePlace.mk (j.comp (algebraMap E K)) then -1 else 1))

theorem **exists_norm_sign** : ∀ (E K : Type) [Field E] [NumberField E] [Field K] [NumberField K] [Algebra E K] (j : K →+* ℂ), NumberField.ComplexEmbedding.IsReal j → ∀ (T : Finset (HeightOneSpectrum (𝒪 K))) (M : ℕ) (W : Set (Ideles E)), W ∈ nhds (ReciprocityTheory.Neutral.Bridge.HCFR.infIdele E (InfinitePlace.mk (j.comp (algebraMap E K))) (-1)) → ∃ a : Kˣ, (∀ w ∈ T, w.valuation K ((a : K) - 1) ≤ WithZero.exp (-(M : ℤ))) ∧ Units.map (ReciprocityTheory.Neutral.Bridge.HCFR.infPart E) (diagUnits E (Units.map (Algebra.norm E (S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormNearNegOneAtRealPlaceStatement

theorem **_root_.GlobalTheory.T.exists_norm_near_neg_one_at_real_place** : ∀ (E K : Type) [Field E] [NumberField E] [Field K] [NumberField K] [Algebra E K] (j : K →+* ℂ), NumberField.ComplexEmbedding.IsReal j → ∀ (T : Finset (HeightOneSpectrum (𝒪 K))) (M : ℕ) (W : Set (Ideles E)), W ∈ nhds (ReciprocityTheory.Neutral.Bridge.HCFR.infIdele E (InfinitePlace.mk (j.comp (algebraMap E K))) (-1)) → ∃ a : Kˣ, (∀ w ∈ T, w.valuation K ((a : K) - 1) ≤ WithZero.exp (-(M : ℤ))) ∧ Units.map (ReciprocityTheory.Neutral.Bridge.HCFR.infPart E) (diagUnits E (Units.map (Algebra.norm E (S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOfConjFixedUnitTamePlace

theorem **compAt_cA** (D : D7) (v : Pl D) (a : AE D) : compAt D v (cA D a) = sigV D v (compAt D v a)

theorem **tmul_algebraMap** (D : D7) (v : Pl D) (c : Fv D.host.F v) (f : D.host.F) : c ⊗ₜ[D.host.F] (algebraMap D.host.F D.host.E f) = (algebraMap D.host.F (Fv D.host.F v) f * c) ⊗ₜ[D.host.F] (1 : D.host.E)

theorem **fixed_unit_norm** : ∀ (D : D7) (v : Pl D) (ε : AE D), Tame D v → ε * ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) → ∀ u ∈ intAt D (· = v), cA D u = u → (∃ y ∈ intAt D (· = v), u * y = ε) → ∃ x ∈ intAt D (· = v), x * cA D x = u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOfConjFixedUnitTamePlaceStatement

theorem **_root_.GlobalTheory.T.exists_norm_of_conjFixed_unit_tamePlace** : ∀ (D : D7) (v : Pl D) (ε : AE D), Tame D v → ε * ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) → ∀ u ∈ intAt D (· = v), cA D u = u → (∃ y ∈ intAt D (· = v), u * y = ε) → ∃ x ∈ intAt D (· = v), x * cA D x = u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOneIdeleOfSupportedRankOneUnitary

theorem **exists_normOneAt_of_rank_one** (D : D7) (L : HSp D) (v : GlobalTheory.FinPl.Pl D) (γ : GA D L) (hr : L.rank = 1) (hγ : γ ∈ grpAt L (· = v)) : ∃ c : AE D, NormOneAt v c ∧ NormOneAt v (cA D c) ∧ Formalism.L2.Theta.mt L γ = c • (1 : Matrix (Fin L.rank) (Fin L.rank) (AE D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOneIdeleOfSupportedRankOneUnitaryStatement

theorem **_root_.GlobalTheory.T.exists_normOneIdele_of_supported_rankOneUnitary** : ∀ (D : D7) (L : HSp D) (v : GlobalTheory.FinPl.Pl D) (γ : GA D L), L.rank = 1 → γ ∈ grpAt L (· = v) → ∃ c : AE D, NormOneAt v c ∧ NormOneAt v (cA D c) ∧ Formalism.L2.Theta.mt L γ = c • (1 : Matrix (Fin L.rank) (Fin L.rank) (AE D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOneIdeleWithSplitPlaceComponent

theorem **exists_normOneAt_pr** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (c₀ : k) (hc₀ : c₀ ≠ 0) : ∃ c : AE D, NormOneAt p.v c ∧ p.pr c = c₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOneIdeleWithSplitPlaceComponentStatement

theorem **_root_.GlobalTheory.T.exists_normOneIdele_with_splitPlace_component** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (c₀ : k), c₀ ≠ 0 → ∃ c : AE D, NormOneAt p.v c ∧ p.pr c = c₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOneWithResiduesAtSplitPlaces

theorem **exists_normOne_residues** {A : Type} [CommRing A] {S : Subring A} {P : Type} {k : P → Type} [∀ Q, Field (k Q)] {r : ∀ Q, S →+* k Q} (Z : PlaceSys A S P k r) (σ : A →+* A) (hσ : ∀ a, σ (σ a) = a) (hS : ∀ a ∈ S, σ a ∈ S) (T : Finset P) (hT : ∀ Q ∈ T, ∀ Q' ∈ T, σ (Z.e Q) ≠ Z.e Q') (c : ∀ Q, (k Q)ˣ) : ∃ z : S, (z : A) * σ z = 1 ∧ (∀ Q ∈ T, r Q z = c Q) ∧ ∀ b : A, (∀ Q ∈ T, Z.p Q b ∈ Z.O Q ∧ Z.p Q (σ b) ∈ Z.O Q) → (1 - (z : A)) * b ∈ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormOneWithResiduesAtSplitPlacesStatement

theorem **_root_.GlobalTheory.T.exists_normOne_with_residues_at_splitPlaces** : ∀ {A : Type} [CommRing A] {S : Subring A} {P : Type} {k : P → Type} [∀ Q, Field (k Q)] {r : ∀ Q, S →+* k Q} (Z : PlaceSys A S P k r) (σ : A →+* A), (∀ a, σ (σ a) = a) → (∀ a ∈ S, σ a ∈ S) → ∀ (T : Finset P), (∀ Q ∈ T, ∀ Q' ∈ T, σ (Z.e Q) ≠ Z.e Q') → ∀ c : ∀ Q, (k Q)ˣ, ∃ z : S, (z : A) * σ z = 1 ∧ (∀ Q ∈ T, r Q z = c Q) ∧ ∀ b : A, (∀ Q ∈ T, Z.p Q b ∈ Z.O Q ∧ Z.p Q (σ b) ∈ Z.O Q) → (1 - (z : A)) * b ∈ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormalFramesAndBasePointSplitPlace

theorem **mt_mul_gaMat** (V : HSp D) (g h : GA D V) : mt V (g * h) = mt V g * mt V h

theorem **mt_one_gaMat** (V : HSp D) : mt V 1 = 1

theorem **Place.pi_idem** (p : Place D k) : p.π p.e = 1

theorem **Place.pi_eq_of_toEAf** (p : Place D k) {a b : AE D} (h : toEAf D a = toEAf D b) : p.π a = p.π b

theorem **Place.map_pi_eq_of_toEAf** (p : Place D k) {m n : ℕ} {a b : Matrix (Fin m) (Fin n) (AE D)} (h : a.map (toEAf D) = b.map (toEAf D)) : a.map p.π = b.map p.π

theorem **exists_toEAf_lift** (z : EAf D) : ∃ w : AE D, toEAf D w = z

theorem **liftM_map** {m n : ℕ} (x : Matrix (Fin m) (Fin n) (EAf D)) : (liftM x).map (toEAf D) = x

theorem **exists_GL_mul_of_mul_eq_one** {n : ℕ} (M : GL (Fin n) k) (N Ni : Matrix (Fin n) (Fin n) k) (h : N * Ni = 1) : ∃ gk : GL (Fin n) k, gk.val = M.val * Ni

theorem **Place.exists_movedTo** (p : Place D k) (V : HSp D) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)) (M : GL (Fin V.rank) k) (g₀ : GA D V) : ∃ g, SuppAt p.e V g ∧ MovedTo p V (g * g₀) β M.val

theorem **normalFrames_and_basePoint_exist** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k), p.xi ≠ 0 → (∀ (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (B : BesLev D ρ) (s : k) (FA : Frame k B.A.rank n), s ≠ 0 → (∃ F : Frame k B.V₀.rank (n ⊕ Unit), p.IsCoreFrame B s FA F) ∧ ∃ F : Frame k B.V.rank (n ⊕ Fin (2 * ρ + 1)), p.IsBesFrame B s FA F) ∧ (∀ (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (J : JacLev D ρ) (s hh : k) (F₀ : Frame k J.W₀.rank n), s ≠ 0 → hh ≠ 0 → ∃ F : Frame k J.W.rank (n ⊕ Fin (2 * ρ)), p.IsJacFrame J s hh F₀ F) ∧ ∀ (V : HSp D) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)) (M : GL (Fin V.rank) k) (g₀ : GA D V), ∃ g, SuppAt p.e V g ∧ MovedTo p V (g * g₀) β M.val

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormalFramesAndBasePointSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_normal_frames_and_base_point_split_place** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k), p.xi ≠ 0 → (∀ (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (B : BesLev D ρ) (s : k) (FA : Frame k B.A.rank n), s ≠ 0 → (∃ F : Frame k B.V₀.rank (n ⊕ Unit), p.IsCoreFrame B s FA F) ∧ ∃ F : Frame k B.V.rank (n ⊕ Fin (2 * ρ + 1)), p.IsBesFrame B s FA F) ∧ (∀ (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (J : JacLev D ρ) (s hh : k) (F₀ : Frame k J.W₀.rank n), s ≠ 0 → hh ≠ 0 → ∃ F : Frame k J.W.rank (n ⊕ Fin (2 * ρ)), p.IsJacFrame J s hh F₀ F) ∧ ∀ (V : HSp D) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)) (M : GL (Fin V.rank) k) (g₀ : GA D V), ∃ g, SuppAt p.e V g ∧ MovedTo p V (g * g₀) β M.val

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormalFramesRationalDecompositionsOverLoca

theorem **val_xi** (p : SplitPt D k) (hU : p.UnitAA) : valuation k (p.pr (incA D D.host.xi0)) = 1

theorem **val_star_xi** (p : SplitPt D k) (hU : p.UnitAA) : valuation k (p.pr (incA D (star D.host.xi0))) = 1

theorem **val_nat** (p : SplitPt D k) {PP : Pl D.host.E} (hI : Formalism.L2.NTheta.IsIdemAt p.e PP) (n : ℕ) (hn : ((n : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal) : valuation k (p.pr (incA D (n : D.host.E))) = 1

theorem **tau_norm** {a : XGeo D} (y : YRat a) {lam : D.host.E} (hT : y.Tau lam) (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) : star lam * (lam * (star (y.bes j 0 h).ζ ·ᵥ (y.bes j 0 h).L.gram *ᵥ (y.bes j 0 h).ζ)) = -(D.host.xi0 * star D.host.xi0) * (star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀)

theorem **val_hz** {a : XGeo D} (y : YRat a) (q : SplitPt D k) (hU : q.UnitAA) {PP : Pl D.host.E} (hI : Formalism.L2.NTheta.IsIdemAt q.e PP) (n : ℕ) (hT : y.Tau (-D.host.xi0 ^ (D.R.m / 2))) (hn : star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = ((n : ℕ) : D.host.E)) (hnP : ((n : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal) : ∀ ρ j (h : j + 2 * ρ + 2 = D.R.m), valuation k ((ofSplit q).hz ((y.bes j ρ h).lev (a.x j).vol)) = 1

theorem **normalFrames_exist** : ∀ {D : D7} (a : XGeo D) (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (n : ℕ), y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = ((n : ℕ) : D.host.E) → ((n : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal → ∃ (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormalFramesRationalDecompositionsOverLocaStatement

theorem **_root_.GlobalTheory.T.exists_normalFrames_rationalDecompositions_over_local** : ∀ {D : D7} (a : XGeo D) (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (n : ℕ), y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = ((n : ℕ) : D.host.E) → ((n : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal → ∃ (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormalizedHaarPairAtFinitePlaces

theorem **normAt_exists** : ∀ (D : D7) (V : HSp D) (R : Subgroup (GA D V)), IsClosed (R : Set (GA D V)) → ∀ (L : Lat V), IsSelfDualC V L → ∀ Q : Finset (Pl D), ∃ ρ μ, IsNormAt V R L (· ∈ Q) ρ μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNormalizedHaarPairAtFinitePlacesStatement

theorem **_root_.GlobalTheory.T.exists_normalizedHaarPair_at_finitePlaces** : ∀ (D : D7) (V : HSp D) (R : Subgroup (GA D V)), IsClosed (R : Set (GA D V)) → ∀ (L : Lat V), IsSelfDualC V L → ∀ Q : Finset (Pl D), ∃ ρ μ, IsNormAt V R L (· ∈ Q) ρ μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNotMemDvdLocallyAway

theorem **exists_not_mem_dvd_locally_away** (E : Type) [Field E] [NumberField E] (Q : HeightOneSpectrum (𝒪 E)) (r : 𝒪 E) (hr : r ≠ 0) : ∃ t : 𝒪 E, t ∉ Q.asIdeal ∧ ∀ Q' : HeightOneSpectrum (𝒪 E), Q' ≠ Q → ∃ w : 𝒪 E, w ∉ Q'.asIdeal ∧ ∃ z : 𝒪 E, w * t = r * z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNsmulAnnihilatorHalfMemStandardHom

theorem **symplO_nsmul_swap** (S T : HSp D) (N : ℕ) (x y : WO S T) : symplO S T (N • x) y = symplO S T x (N • y)

theorem **exists_nsmul_frame_mem** (S T : HSp D) : ∃ N : ℕ, 0 < N ∧ ∀ y ∈ latO S T (GlobalTheory.PlaceModel.latFrame S : Set (Fin S.rank → D.host.E)) (GlobalTheory.PlaceModel.latFrame T : Set (Fin T.rank → D.host.E)) (fun _ => True), N • y ∈ stdHom S T

theorem **exists_nsmul_ann_mem_stdHom** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∃ N : ℕ, 0 < N ∧ ∀ u : WO S T, IsHalf (cA D E₂) u → (∀ x ∈ stdHom S T, IsHalf E₂ x → ψ (symplO S T x u) = 1) → N • u ∈ stdHom S T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNsmulAnnihilatorHalfMemStandardHomStatement

theorem **_root_.GlobalTheory.T.exists_nsmul_annihilator_half_mem_standardHom** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∃ N : ℕ, 0 < N ∧ ∀ u : WO S T, IsHalf (cA D E₂) u → (∀ x ∈ stdHom S T, IsHalf E₂ x → ψ (symplO S T x u) = 1) → N • u ∈ stdHom S T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNsmulHalf

theorem **exists_nsmul_half** (p : SplitPt D k) {S T : HSp D} {LS MS : Lat S} {LT MT : Lat T} (hLS : IsSelfDualC S LS) (hLT : IsSelfDualC T LT) (hMS : IsSelfDualC S MS) (hMT : IsSelfDualC T MT) : ∃ N : ℕ, 0 < N ∧ ∀ u ∈ latO S T (MS : Set (Fin S.rank → D.host.E)) (MT : Set (Fin T.rank → D.host.E)) (· = p.v), IsHalf p.e u → N • u ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = p.v)

theorem **isHalf_nsmul** {S T : HSp D} {e : AE D} {u : WO S T} (hu : IsHalf e u) (N : ℕ) : IsHalf e (N • u)

theorem **sbHalf_mono** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S T : HSp D) (LS MS : Lat S) (LT MT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → IsSelfDualC S MS → IsSelfDualC T MT → ∀ Φ : WO S T → ℂ, IsSBHalf p.e (latO S T (MS : Set (Fin S.rank → D.host.E)) (MT : Set (Fin T.rank → D.host.E)) (· = p.v)) Φ → IsSBHalf p.e (latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = p.v)) Φ

theorem **placeIdem_mul** {v : Pl D} {ε : AF D.host.F} (hε : IsPlaceIdem D v ε) {a : AE D} (ha : a ∈ idAt D (· = v)) : algebraMap (AF D.host.F) (AE D) ε * a = a

theorem **eq_add_halves** (p : SplitPt D k) {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {u : WO S T} (hu : u ∈ latO S T LS LT (· = p.v)) : p.e • u + cA D p.e • u = u

theorem **smul_conj_zero** (e e' : AE D) (h : e' * cA D e' = 0) {S T : HSp D} (u : WO S T) (hu : e' • u = u) : cA D e' • (e • u) = 0

theorem **isHalf_add** {S T : HSp D} {e : AE D} {a b : WO S T} (ha : IsHalf e a) (hb : IsHalf e b) : IsHalf e (a + b)

theorem **latVac_image** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS MS : Lat S) (LT MT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → IsSelfDualC S MS → IsSelfDualC T MT → ∀ (f₀ : WO S T → ℂ) (Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) p.e (latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = p.v)) 1 f₀ Tm → IsLatVac (heisO ψ S T) (latO S T (MS : Set (Fin S.rank → D.host.E)) (MT : Set (Fin T.rank → D.host.E)) (· = p.v)) (Tm (halfInd p.e (latO S T (MS : Set (Fin S.rank → D.host.E)) (MT : Set (Fin T.rank → D.host.E)) (· = p.v))))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNsmulMemHomLatticeOfIntegralFrameCoords

theorem **exists_nat_mul_mem** (c : k) : ∃ N : ℕ, 0 < N ∧ (N : k) * c ∈ 𝒪[k]

theorem **exists_nat_mul_mem_fin** {ι : Type} [Fintype ι] (c : ι → k) : ∃ N : ℕ, 0 < N ∧ ∀ i, (N : k) * c i ∈ 𝒪[k]

theorem **entry_mem** {n n' m m' : Type} [Fintype n] [Fintype n'] {N : ℕ} (b : Matrix m n k) (b' : Matrix n' m' k) (hb : ∀ i a, (N : k) * b i a ∈ 𝒪[k]) (hb' : ∀ a j, (N : k) * b' a j ∈ 𝒪[k]) (x : Matrix n n' k) (hx : ∀ i j, x i j ∈ 𝒪[k]) (i : m) (j : m') : ((N * N : ℕ) : k) * (b * x * b') i j ∈ 𝒪[k]

theorem **GlobalTheory.FinPl.exists_nsmul_mem_latO_of_rd_int** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∃ N : ℕ, 0 < N ∧ ∀ t : WO S T, IsHalf p.e t → (∀ i j, rd α β t i j ∈ 𝒪[k]) → N • t ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNsmulMemHomLatticeOfIntegralFrameCoordsStatement

theorem **_root_.GlobalTheory.T.exists_nsmul_mem_homLattice_of_integral_frameCoords** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∃ N : ℕ, 0 < N ∧ ∀ t : WO S T, IsHalf p.e t → (∀ i j, rd α β t i j ∈ 𝒪[k]) → N • t ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsNsmulMemStdHom

theorem **coe_colStdLat** (n : ℕ) : ((Formalism.L2.Col.stdLat D.host.E n : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) : Set (Fin n → D.host.E)) = intVec D n

theorem **exists_nsmul_mem_stdHom** (S T : HSp D) : ∃ K : ℕ, 0 < K ∧ ∀ t : WO S T, t.map (infPart D) = 0 → (∀ i j, toEAf D (t i j) ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) → K • t ∈ stdHom S T

theorem **infPart_liftFin** (x : Af D.host.F ⊗[D.host.F] D.host.E) : infPart D (GlobalTheory.LatticeStable.liftFin D x) = 0

theorem **finPart_liftFin** (x : Af D.host.F ⊗[D.host.F] D.host.E) : finPart D (GlobalTheory.LatticeStable.liftFin D x) = x

theorem **toEAf_eq_comm** (z : AE D) : toEAf D z = Algebra.TensorProduct.comm D.host.F (Af D.host.F) D.host.E (finPart D z)

theorem **toEAf_liftFin** (r : EAf D) : toEAf D (GlobalTheory.LatticeStable.liftFin D ((Algebra.TensorProduct.comm D.host.F (Af D.host.F) D.host.E).symm r)) = r

theorem **toEAf_approx_eq_zero** (t L : AE D) (x : D.host.E) (M : ℕ) (r : EAf D) (ht : toEAf D t = eToEAf D x + (M : EAf D) * r) (hL : toEAf D L = -r) : toEAf D (incA D x - t - (M : AE D) * L) = 0

theorem **approx_entry** (t : AE D) (M : ℕ) (hM : 0 < M) : ∃ (x : D.host.E) (w : AE D), infPart D w = 0 ∧ toEAf D w ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ finPart D (incA D x − t − (M : AE D) ∗ w) = 0

theorem **exists_rat_half_approx** : ∀ {D : D7} (S T : HSp D) (e : AE D) (P : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e P → ∀ (N : ℕ), 0 < N → ∀ y : WO S T, ∃ (γ₀ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (v : WO S T), v ∈ stdHom S T ∧ IsHalf e v ∧ γ₀.map (fun r => e ∗ incA D r) = e • y + N • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOpenAddSubgroupFinitePoints

theorem **exists_openAddSubgroup_finite_points** : ∀ (D : D7) (V : Set (AE D)), V ∈ nhds (0 : AE D) → ∃ U : AddSubgroup (AE D), IsOpen (U : Set (AE D)) ∧ ∀ a ∈ U, infPart D a = 0 → a ∈ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOpenAddSubgroupFinitePointsStatement

theorem **_root_.GlobalTheory.T.exists_openAddSubgroup_finite_points_alt** : ∀ (D : D7) (V : Set (AE D)), V ∈ nhds (0 : AE D) → ∃ U : AddSubgroup (AE D), IsOpen (U : Set (AE D)) ∧ ∀ a ∈ U, infPart D a = 0 → a ∈ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOpenFix

theorem **exists_open_fix'** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {Y : Type} [Fintype Y] [DecidableEq Y] (L : Level (GL Y k)) (hL : IsOpen (L.J : Set (GL Y k))) (α : GL Y k) : ∃ U : Subgroup (GL Y k), IsOpen (U : Set (GL Y k)) ∧ ∀ (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL Y k) A), IsSmoothRep σ → ∀ v : A, (∀ j : L.J, σ (j : GL Y k) v = (L.lam j : ℂ) • v) → ∀ u ∈ U, σ u (σ α v) = σ α v

theorem **exists_isOneComb** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c P : ℕ), 1 ≤ P → P ≤ c → ∀ (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ ν : Measure k, IsSelfDual ψ ν → ∀ (L : Level (GL (n ⊕ Fin c) k)), IsCO L.J → ∀ α, ∃ e, IsOneComb ψ ν P L α e

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOpenPrincipalCongruenceFinitelyManyPrimes

theorem **exists_open_princ** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Z : Finset (Pl D.host.E)), ∃ U : Subgroup (GA D V), IsOpen (U : Set (GA D V)) ∧ ∀ k ∈ U, Lev.finE V k ∈ Formalism.L2.DIV.St β ∧ ∀ PP ∈ Z, Formalism.L2.DIV.redFn β PP (Lev.finE V k) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOpenPrincipalCongruenceFinitelyManyPrimesStatement

theorem **_root_.GlobalTheory.T.exists_open_principal_congruence_finitely_many_primes** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Z : Finset (Pl D.host.E)), ∃ U : Subgroup (GA D V), IsOpen (U : Set (GA D V)) ∧ ∀ k ∈ U, Lev.finE V k ∈ Formalism.L2.DIV.St β ∧ ∀ PP ∈ Z, Formalism.L2.DIV.redFn β PP (Lev.finE V k) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthDec

theorem **exists_orthDec** (D : D7) (A L : HSp D) : Nonempty (Σ V₀ : HSp D, OrthDec V₀ A L)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthogonalComplementPreimageOfFixLine

theorem **exists_emb_of_fix_line** : ∀ {D : D7} {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (b : GA D S₀), (mt S₀ b ∗ᵥ (fun k => incA D (vecB o ζ k)) = fun k => incA D (vecB o ζ k)) → ∃ a : GA D A, o.emb (a, 1) = b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthogonalComplementPreimageOfFixLineStatement

theorem **_root_.GlobalTheory.T.exists_orthogonalComplement_preimage_of_fix_line** : ∀ {D : D7} {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (b : GA D S₀), (mt S₀ b ∗ᵥ (fun k => incA D (vecB o ζ k)) = fun k => incA D (vecB o ζ k)) → ∃ a : GA D A, o.emb (a, 1) = b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthogonalIdempotentsOfSumMulEqOne

theorem **exists_orthogonal_idempotents_of_sum_mul_eq_one** (F : Type) [Field F] [NumberField F] (ι : Type) [Fintype ι] (f s : ι → AdeleRing (𝒪 F) F) (h : ∑ i, s i ∗ f i = 1) : ∃ e t : ι → AdeleRing (𝒪 F) F, ∑ i, e i = 1 ∧ (∀ i j, i ≠ j → e i ∗ e j = 0) ∧ ∀ i, t i ∗ f i = e i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthogonalIdempotentsOfSumMulEqOneStatement

theorem **_root_.GlobalTheory.T.exists_orthogonal_idempotents_of_sum_mul_eq_one'** : ∀ (F : Type) [Field F] [NumberField F] (ι : Type) [Fintype ι] (f s : ι → NumberField.AdeleRing (NumberField.RingOfIntegers F) F), ∑ i, s i ∗ f i = 1 → ∃ e t : ι → NumberField.AdeleRing (NumberField.RingOfIntegers F) F, ∑ i, e i = 1 ∧ (∀ i j, i ≠ j → e i ∗ e j = 0) ∧ ∀ i, t i ∗ f i = e i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthogonalSumAndHyperbolicSumSpaces

theorem **col_gram** {V W : HSpace E} {k l : ℕ} (X : Matrix (Fin V.rank) (Fin k) E) (Y : Matrix (Fin W.rank) (Fin k) E) (X' : Matrix (Fin V.rank) (Fin l) E) (Y' : Matrix (Fin W.rank) (Fin l) E) : (col X Y)ᴴ ∗ (GlobalTheory.Doubling.hsum V W).gram ∗ col X' Y' = Xᴴ ∗ V.gram ∗ X' + Yᴴ ∗ W.gram ∗ Y'

theorem **posSp_sigOK** (D : D7) (ρ : ℕ) : (posSp D ρ).SigOK

theorem **posSp_gram** (D : D7) (ρ : ℕ) : (posSp D ρ).gram = 1

theorem **hypSp_sigOK** (D : D7) (ρ : ℕ) : (hypSp D ρ).SigOK

theorem **hypSp_sig** (D : D7) (ρ : ℕ) (τ) : (hypSp D ρ).sig τ = (ρ, ρ)

theorem **exists_orthSum_hypSum** : ∀ (D : D7), (∀ A L : HSp D, A.SigOK → L.SigOK → ∃ (S : HSp D) (_ : OrthDec S A L), S.SigOK ∧ ∀ τ, S.sig τ = A.sig τ + L.sig τ) ∧ ∀ (A : HSp D) (ρ : ℕ), A.SigOK → ∃ (S : HSp D) (_ : HypDec S A ρ), S.SigOK ∧ ∀ τ, S.sig τ = A.sig τ + (ρ, ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthogonalSumAndHyperbolicSumSpacesStatement

theorem **_root_.GlobalTheory.T.exists_orthogonalSum_and_hyperbolicSum_spaces** : ∀ (D : D7), (∀ A L : HSp D, A.SigOK → L.SigOK → ∃ (S : HSp D) (_ : OrthDec S A L), S.SigOK ∧ ∀ τ, S.sig τ = A.sig τ + L.sig τ) ∧ ∀ (A : HSp D) (ρ : ℕ), A.SigOK → ∃ (S : HSp D) (_ : HypDec S A ρ), S.SigOK ∧ ∀ τ, S.sig τ = A.sig τ + (ρ, ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthonormalBasisUnimodularHermitianLocalRi

theorem **exists_orthonormal_of_isLocalRing** (O : Type) [CommRing O] [IsLocalRing O] (s : O →+* O) (hs : ∀ x, s (s x) = x) (hτ : ∃ τ : O, τ + s τ = 1) (hn : ∀ u : O, IsUnit u → s u = u → ∃ x : O, s x * x = u) (n : ℕ) (Q : Matrix (Fin n) (Fin n) O) (hQ : Formalism.UG.cT s Q = Q) (hu : IsUnit Q) : ∃ a : GL (Fin n) O, Formalism.UG.cT s a.1 * Q * a.1 = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsOrthonormalBasisUnimodularHermitianLocalRiStatement

theorem **_root_.GlobalTheory.T.exists_orthonormal_basis_unimodular_hermitian_local_ring** : ∀ (O : Type) [CommRing O] [IsLocalRing O] (s : O →+* O), (∀ x, s (s x) = x) → (∃ τ : O, τ + s τ = 1) → (∀ u : O, IsUnit u → s u = u → ∃ x : O, s x * x = u) → ∀ (n : ℕ) (Q : Matrix (Fin n) (Fin n) O), Formalism.UG.cT s Q = Q → IsUnit Q → ∃ a : GL (Fin n) O, Formalism.UG.cT s a.1 * Q * a.1 = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPadicCoordinatesOfDegreeOneSplitPlace

theorem **exists_coord_padic** : ∀ (D : D7) (T : ℕ) [Fact T.Prime] (w : Formalism.L2.DIV.Pl D.host.E) (e : AE D), Formalism.L2.NTheta.IsIdemAt e w → IsSplitIdem D e → Ideal.absNorm w.asIdeal = T → ¬ (w.asIdeal ^ 2 ∣ Ideal.span {(T : 𝒪 D.host.E)}) → ∃ (v : Pl D) (pr : AE D →+* ℚ_[T]) (sec : ℚ_[T] →+ AE D), (∃ ε, IsPlaceIdem D v ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε) ∧ (∀ a, pr (sec a) = a) ∧ (∀ x, sec (pr x) = e * x) ∧ Continuous pr ∧ Continuous sec ∧ ∀ a, ‖a‖ ≤ 1 ↔ finPart D (sec a) ∈ intAE D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPadicCoordinatesOfDegreeOneSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_padicCoordinates_of_degreeOne_splitPlace** : ∀ (D : D7) (T : ℕ) [Fact T.Prime] (w : Formalism.L2.DIV.Pl D.host.E) (e : AE D), Formalism.L2.NTheta.IsIdemAt e w → IsSplitIdem D e → Ideal.absNorm w.asIdeal = T → ¬ (w.asIdeal ^ 2 ∣ Ideal.span {(T : 𝒪 D.host.E)}) → ∃ (v : Pl D) (pr : AE D →+* ℚ_[T]) (sec : ℚ_[T] →+ AE D), (∃ ε, IsPlaceIdem D v ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε) ∧ (∀ a, pr (sec a) = a) ∧ (∀ x, sec (pr x) = e * x) ∧ Continuous pr ∧ Continuous sec ∧ ∀ a, ‖a‖ ≤ 1 ↔ finPart D (sec a) ∈ intAE D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPairSpaceLinearMapViaPartialFourier

theorem **eq_zero_of_hat_eq_zero** (ν : Measure k) [ν.IsAddHaarMeasure] (ψ : AddChar k Circle) (hψ : Continuous ψ) (hne : ψ ≠ 1) {n : Type} [Fintype n] [DecidableEq n] {f : (n → k) → ℂ} (hf : f ∈ SB (n → k)) (h0 : ∀ η, GlobalTheory.StepData.hat ψ ν (-1) f η = 0) (a : n → k) : f a = 0

theorem **withLast_toCols** {n n' : Type} (x : Matrix n (n' ⊕ Unit) k) : withLast x.toCols₁ (fun r => x r (Sum.inr ())) = x

theorem **continuous_withLast** {n n' : Type} (x₀ : Matrix n n' k) : Continuous (withLast x₀)

theorem **slice_mem_SB** {n n' : Type} [Fintype n] [Fintype n'] {Ψ : Matrix n (n' ⊕ Unit) k → ℂ} (hΨ : Ψ ∈ SB (Matrix n (n' ⊕ Unit) k)) (x₀ : Matrix n n' k) : (fun y : n → k => Ψ (withLast x₀ y)) ∈ SB (n → k)

theorem **partialHat_injective** (ν : Measure k) [ν.IsAddHaarMeasure] (ψ : AddChar k Circle) (hψ : Continuous ψ) (hne : ψ ≠ 1) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] : Function.Injective (partialHat ν ψ hψ n n')

theorem **coupleBil_injective** (ν : Measure k) [ν.IsAddHaarMeasure] (ψ : AddChar k Circle) (hψ : Continuous ψ) (hne : ψ ≠ 1) (n n' Z : Type) [Fintype n] [DecidableEq n] [Fintype n'] : Function.Injective (TensorProduct.lift (coupleBil ν ψ hψ n n' Z))

theorem **exists_pair_map** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψk : AddChar k Circle), Continuous ψk → ψk ≠ 1 → ∀ (n n' Z : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype Z] [DecidableEq Z] (M : Type) [AddCommGroup M] [Module ℂ M] (Bl : SB (Matrix Z n k) →ₗ[ℂ] SB (Matrix n (n' ⊕ Unit) k) →ₗ[ℂ] M), ∃ Λ : SB (Matrix n n' k × Matrix (Z ⊕ Unit) n k) →ₗ[ℂ] M, (∀ (Ψ₁ : SB (Matrix n (n' ⊕ Unit) k)) (Ψ₂ : SB (Matrix Z n k)) (Ξ : SB (Matrix n n' k × Matrix (Z ⊕ Unit) n k)), (∀ q, Ξ.1 q = Ψ₂.1 q.2.toRows₁ * GlobalTheory.StepData.hat ψk ν (-1) (fun y => Ψ₁.1 (fromCols q.1 (Matrix.of fun r _ => y r))) (q.2 (Sum.inr ()))) → Λ Ξ = Bl Ψ₂ Ψ₁) ∧ ∀ (Ξ₁ : SB (Matrix n n' k)) (a : SB (Matrix Z n k)) (c : SB (Matrix n Unit k)) (Ψ₁ : SB (Matrix n (n' ⊕ Un

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPairSpaceLinearMapViaPartialFourierStatement

theorem **_root_.GlobalTheory.T.exists_pair_space_linear_map_via_partial_fourier** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψk : AddChar k Circle), Continuous ψk → ψk ≠ 1 → ∀ (n n' Z : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype Z] [DecidableEq Z] (M : Type) [AddCommGroup M] [Module ℂ M] (Bl : SB (Matrix Z n k) →ₗ[ℂ] SB (Matrix n (n' ⊕ Unit) k) →ₗ[ℂ] M), ∃ Λ : SB (Matrix n n' k × Matrix (Z ⊕ Unit) n k) →ₗ[ℂ] M, (∀ (Ψ₁ : SB (Matrix n (n' ⊕ Unit) k)) (Ψ₂ : SB (Matrix Z n k)) (Ξ : SB (Matrix n n' k × Matrix (Z ⊕ Unit) n k)), (∀ q, Ξ.1 q = Ψ₂.1 q.2.toRows₁ * GlobalTheory.StepData.hat ψk ν (-1) (fun y => Ψ₁.1 (fromCols q.1 (Matrix.of fun r _ => y r))) (q.2 (Sum.inr ()))) → Λ Ξ = Bl Ψ₂ Ψ₁) ∧ ∀ (Ξ₁ : SB (Matrix n n' k)) (a : SB (Matrix Z n k))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsParahoricAndIntegralGLDoubleCosetReps

theorem **exists_reps_both** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D], (∀ g : GL (A ⊕ D) k, ∃ R d, IsReps g R d) ∧ ∀ g : GL D k, ∃ R, IsRepsK g R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsParahoricAndIntegralGLDoubleCosetRepsStatement

theorem **_root_.GlobalTheory.T.exists_parahoric_and_integralGL_doubleCosetReps** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D], (∀ g : GL (A ⊕ D) k, ∃ R d, IsReps g R d) ∧ ∀ g : GL D k, ∃ R, IsRepsK g R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsParamCoverDiscreteSeries

theorem **blockOrd_std** (p q : ℕ) : BlockOrd (finSumFinEquiv : Ix p q ≃ Fin (p + q)) fun i => (finSumFinEquiv.symm i).isLeft

theorem **blockOrd_perm** {e : Ix p q ≃ Fin n} {ε : Fin n → Bool} (h : BlockOrd e ε) (s : Equiv.Perm (Fin n)) (hs : ∀ a b, a < b → ε (s a) = ε (s b) → s a < s b) : BlockOrd (e.trans s.symm) (ε ∘ s)

theorem **exists_param_of_trivial** (hE : GlobalTheory.Dec.DSExhaust) {T : GrpData p q} (R : DSRep T) (k : ∀ x, T.cov x = 1 → R.U.π x = 1) : ∃ (P : SignedPar (p + q)) (e : Ix p q ≃ Fin (p + q)), BlockOrd e P.sgn ∧ HasParam R (blockPar P e)

theorem **exists_param** : GlobalTheory.Dec.DSExhaust → ∀ (p q : ℕ), 1 ≤ p + q → ∀ (T : GrpData p q) (R : DSRep T), ∃ lam, IsParam lam ∧ HasParam R lam

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsParamCoverDiscreteSeriesStatement

theorem **_root_.GlobalTheory.T.exists_param_coverDiscreteSeries** : GlobalTheory.Dec.DSExhaust → ∀ (p q : ℕ), 1 ≤ p + q → ∀ (T : GrpData p q) (R : DSRep T), ∃ lam, IsParam lam ∧ HasParam R lam

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPartitionOne

theorem **finite_ratW_mem** {D : D7} (A T : HSp D) {C : Set (WO A T)} (hC : IsCompact C) : {γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E | ratW T A γ ∈ C}.Finite

theorem **exists_partition_one** : ∀ (D : D7) (A T : HSp D), ∃ χ : WO A T → ℝ, Continuous χ ∧ HasCompactSupport χ ∧ (∀ u, 0 ≤ χ u) ∧ ∀ u, HasSum (fun γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E => χ (ratW T A γ + u)) 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPeriodIdealLocalGeneratorAtSplitPoint

theorem **half_diff_ne_bot** (D : D7) : D.host.𝔟 ≠ ⊥

theorem **AA_ne_zero** (D : D7) : D.host.AA ≠ 0

theorem **pr_incA_int** (q : SplitPt D k) (x : 𝒪 D.host.E) : q.pr (incA D (x : D.host.E)) ∈ 𝒪[k]

theorem **exists_gen** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), ∃ a₀ ∈ (D.host.AA : Set D.host.E), q.pr (incA D a₀) ≠ 0 ∧ ∀ a ∈ (D.host.AA : Set D.host.E), q.pr (incA D a) / q.pr (incA D a₀) ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPeriodIdealLocalGeneratorAtSplitPointStatement

theorem **_root_.GlobalTheory.T.exists_periodIdeal_local_generator_at_split_point** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), ∃ a₀ ∈ (D.host.AA : Set D.host.E), q.pr (incA D a₀) ≠ 0 ∧ ∀ a ∈ (D.host.AA : Set D.host.E), q.pr (incA D a) / q.pr (incA D a₀) ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPiTensorMvPolynomialEquivBlocks

theorem **blockMul_apply** (f : ι → MvPolynomial σ ℂ) : blockMul ι σ f = ∏ τ, rename (Prod.mk τ) (f τ)

theorem **expEquiv_apply** (d : ι → σ →₀ ℕ) (a : ι) (b : σ) : expEquiv ι σ d (a, b) = d a b

theorem **sum_mapDomain_eq** (d : ι → σ →₀ ℕ) : ∑ τ, (d τ).mapDomain (Prod.mk τ) = expEquiv ι σ d

theorem **prod_rename_monomial** (d : ι → σ →₀ ℕ) : ∏ τ, rename (Prod.mk τ) (monomial (d τ) (1 : ℂ)) = monomial (expEquiv ι σ d) 1

theorem **blockEquiv_toLinearMap** : (blockEquiv ι σ).toLinearMap = PiTensorProduct.lift (blockMul ι σ)

theorem **blockEquiv_tprod** (f : ι → MvPolynomial σ ℂ) : blockEquiv ι σ (PiTensorProduct.tprod ℂ f) = ∏ τ, rename (Prod.mk τ) (f τ)

theorem **map_subtype_injective** (U : ι → Submodule ℂ (MvPolynomial σ ℂ)) : Function.Injective (PiTensorProduct.map fun τ => (U τ).subtype)

theorem **exists_blockEquiv** : ∀ (ι σ : Type) [Fintype ι] [DecidableEq ι], ∃ e : PiTensorProduct ℂ (fun _ : ι => MvPolynomial σ ℂ) ≃ₗ[ℂ] MvPolynomial (ι × σ) ℂ, (∀ f : ι → MvPolynomial σ ℂ, e (PiTensorProduct.tprod ℂ f) = ∏ τ, rename (Prod.mk τ) (f τ)) ∧ ∀ U : ι → Submodule ℂ (MvPolynomial σ ℂ), Function.Injective (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (U τ).subtype) ∧ LinearMap.range (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (U τ).subtype) = Submodule.span ℂ {p | ∃ f : ι → MvPolynomial σ ℂ, (∀ τ, f τ ∈ U τ) ∧ p = ∏ τ, rename (Prod.mk τ) (f τ)}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPiTensorMvPolynomialEquivBlocksStatement

theorem **_root_.GlobalTheory.T.exists_piTensorMvPolynomialEquiv_blocks** : ∀ (ι σ : Type) [Fintype ι] [DecidableEq ι], ∃ e : PiTensorProduct ℂ (fun _ : ι => MvPolynomial σ ℂ) ≃ₗ[ℂ] MvPolynomial (ι × σ) ℂ, (∀ f : ι → MvPolynomial σ ℂ, e (PiTensorProduct.tprod ℂ f) = ∏ τ, rename (Prod.mk τ) (f τ)) ∧ ∀ U : ι → Submodule ℂ (MvPolynomial σ ℂ), Function.Injective (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (U τ).subtype) ∧ LinearMap.range (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (U τ).subtype) = Submodule.span ℂ {p | ∃ f : ι → MvPolynomial σ ℂ, (∀ τ, f τ ∈ U τ) ∧ p = ∏ τ, rename (Prod.mk τ) (f τ)}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPlace

theorem **exists_place** : ∀ (D : D7) (e : Formalism.L2.Theta.AE D), IsSplitIdem D e → ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (p : Place D k), p.e = e

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPlaceOverTwo

theorem **exists_place_over_two** (F : Type) [Field F] [NumberField F] : ∃ v : HeightOneSpectrum (𝒪 F), (2 : 𝒪 F) ∈ v.asIdeal

theorem **exists_isSplitIdem** : ∀ D : D7, ∃ e : Formalism.L2.Theta.AE D, IsSplitIdem D e

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPlaceStabilizerPartialProductsFiniteSet

theorem **locM_apply** (e : R) (M : Matrix n n R) (i j : n) : locM e M i j = e * M i j + (1 - e) * (1 : Matrix n n R) i j

theorem **locM_one** (e : R) : locM e (1 : Matrix n n R) = 1

theorem **locM_zero** (M : Matrix n n R) : locM 0 M = 1

theorem **locM_mul_orth** {e f : R} (hef : e * f = 0) (M : Matrix n n R) : locM e M * locM f M = locM (e + f) M

theorem **locM_mul** {e : R} (he : e * e = e) (M N : Matrix n n R) : locM e M * locM e N = locM e (M * N)

theorem **locM_sandwich** {e : R} (he : e * e = e) (A B J : Matrix n n R) : locM e A * J * locM e B = e • (A * J * B) + (1 - e) • J

theorem **cT_locM** (σ : R →+* R) {e : R} (hσ : σ e = e) (M : Matrix n n R) : Formalism.UG.cT σ (locM e M) = locM e (Formalism.UG.cT σ M)

theorem **mt_mul** (S : HSp D) (a b : GA D S) : mt S (a * b) = mt S a * mt S b

theorem **mt_one** (S : HSp D) : mt S 1 = 1

theorem **mt_locG** (S : HSp D) {e : AE D} (he : e * e = e) (hc : cA D e = e) (κ : GA D S) : mt S (locG S he hc κ) = locM e (mt S κ)

theorem **pe_place** (v : Pl D) : IsPlaceIdem D v (pairA D.host.F 0 (singleV D.host.F v 1))

theorem **inf_pe** (v : Pl D) : infPart D (pe D v) = 0

theorem **comp_pe** (v w : Pl D) : compAt D w (pe D v) = if w = v then 1 else 0

theorem **cA_pe** (v : Pl D) : cA D (pe D v) = pe D v

theorem **ext_AE** {x y : AE D} (h1 : infPart D x = infPart D y) (h2 : ∀ w, compAt D w x = compAt D w y) : x = y

theorem **pe_idem** (v : Pl D) : pe D v * pe D v = pe D v

theorem **inf_peT** (T : Finset (Pl D)) : infPart D (peT D T) = 0

theorem **comp_peT** (T : Finset (Pl D)) (w : Pl D) : compAt D w (peT D T) = if w ∈ T then 1 else 0

theorem **pe_mul_peT** {v : Pl D} {T : Finset (Pl D)} (hv : v ∉ T) : pe D v * peT D T = 0

theorem **mem_idAt** {P : Pl D → Prop} {x : AE D} : x ∈ idAt D P ↔ infPart D x = 0 ∧ ∀ w, ¬ P w → compAt D w x = 0

theorem **mem_grpAt** {S : HSp D} {P : Pl D → Prop} {g : GA D S} : g ∈ grpAt S P ↔ ∀ i j, mt S g i j − (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) i j ∈ idAt D P

theorem **pe_mul_of_mem** {v : Pl D} {x : AE D} (hx : x ∈ idAt D (· = v)) : pe D v * x = x

theorem **pe_mul_mem** (v : Pl D) (x : AE D) : pe D v * x ∈ idAt D (· = v)

theorem **idAt_mono** {P : Pl D → Prop} {v : Pl D} (hv : P v) {x : AE D} (hx : x ∈ idAt D (· = v)) : x ∈ idAt D P

theorem **pairA_mul** (a b : Af D.host.F) : pairA D.host.F 0 a * pairA D.host.F 0 b = pairA D.host.F 0 (a * b)

theorem **latV_le** (S : HSp D) (L : Lat S) {P : Pl D → Prop} {v : Pl D} (hv : P v) : latV S L (· = v) ≤ latV S L P

theorem **pe_smul_latV** (S : HSp D) (L : Lat S) (P : Pl D → Prop) (v : Pl D) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : pe D v • y ∈ latV S L (· = v)

theorem **latV_iff** (S : HSp D) (L : Lat S) {P : Pl D → Prop} {v : Pl D} (hv : P v) {z : Fin S.rank → AE D} (hz : ∀ i, z i ∈ idAt D (· = v)) : z ∈ latV S L P ↔ z ∈ latV S L (· = v)

theorem **mulVec_loc** {S : HSp D} (K : Matrix (Fin S.rank) (Fin S.rank) (AE D)) {v : Pl D} {z : Fin S.rank → AE D} (hz : ∀ i, z i ∈ idAt D (· = v)) : locM (pe D v) K *ᵥ z = K *ᵥ z

theorem **inK_locG** (S : HSp D) (L : Lat S) {P : Pl D → Prop} {κ : GA D S} (hκ : InK S L P κ) {v : Pl D} (hv : P v) : InK S L (· = v) (locG S (pe_idem v) (cA_pe v) κ)

theorem **exists_list** (S : HSp D) (L : Lat S) (P : Pl D → Prop) {κ : GA D S} (hκ : InK S L P κ) (T : Finset (Pl D)) (hT : ∀ v ∈ T, P v) : ∃ l : List (GA D S), (∀ y ∈ l, ∃ v, P v ∧ InK S L (· = v) y) ∧ mt S l.prod = locM (peT D T) (mt S κ)

theorem **hom_loc** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) (e a b : A) : φ (e * a + (1 − e) * b) = φ e * φ a + (1 − φ e) * φ b

theorem **hom_loc_one** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) {e : A} (he : φ e = 1) (a b : A) : φ (e * a + (1 − e) * b) = φ a

theorem **hom_loc_zero** {A B : Type} [CommRing A] [CommRing B] (φ : A →+* B) {e : A} (he : φ e = 0) (a b : A) : φ (e * a + (1 − e) * b) = φ b

theorem **finite_place_products** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop) (κ : GA D S), IsSelfDualC S L → InK S L P κ → ∀ Fs : Finset (Pl D), ∃ l : List (GA D S), (∀ y ∈ l, ∃ v, P v ∧ InK S L (· = v) y) ∧ (∀ i j, infPart D (Formalism.L2.Theta.mt S l.prod i j) = infPart D (Formalism.L2.Theta.mt S κ i j)) ∧ (∀ w ∈ Fs, ∀ i j, GlobalTheory.FinPl.compAt D w (Formalism.L2.Theta.mt S l.prod i j) = GlobalTheory.FinPl.compAt D w (Formalism.L2.Theta.mt S κ i j)) ∧ ∀ w, w ∉ Fs → ∀ i j, GlobalTheory.FinPl.compAt D w (Formalism.L2.Theta.mt S l.prod i j) = GlobalTheory.FinPl.compAt D w ((1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) i j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPlaceStabilizerPartialProductsFiniteSetStatement

theorem **_root_.GlobalTheory.T.exists_placeStabilizer_partialProducts_finiteSet** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop) (κ : GA D S), IsSelfDualC S L → InK S L P κ → ∀ Fs : Finset (Pl D), ∃ l : List (GA D S), (∀ y ∈ l, ∃ v, P v ∧ InK S L (· = v) y) ∧ (∀ i j, infPart D (Formalism.L2.Theta.mt S l.prod i j) = infPart D (Formalism.L2.Theta.mt S κ i j)) ∧ (∀ w ∈ Fs, ∀ i j, GlobalTheory.FinPl.compAt D w (Formalism.L2.Theta.mt S l.prod i j) = GlobalTheory.FinPl.compAt D w (Formalism.L2.Theta.mt S κ i j)) ∧ ∀ w, w ∉ Fs → ∀ i j, GlobalTheory.FinPl.compAt D w (Formalism.L2.Theta.mt S l.prod i j) = GlobalTheory.FinPl.compAt D w ((1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) i j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPlacedBesselPoints

theorem **oneBlock_lev** {D : D7} (V : Formalism.BasicNotions.TH.HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Λ : LevDat D.host.E ℂ V.rank) (hn : V.rank ≤ Λ.r) (hs : Λ.s = 0) (hQ : Λ.Q = ∅) (κ : GA D V) : (κ ∈ levS V β Λ ↔ κ ∈ GfC V ∧ Lev.finE V κ ∈ Formalism.L2.DIV.St (F

theorem **exists_isAt** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → GlobalTheory.MiTop.Framed B → ∃ a₀ a, IsAt f B a₀ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPlacedBesselPointsStatement

theorem **_root_.GlobalTheory.T.exists_placed_bessel_points** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → GlobalTheory.MiTop.Framed B → ∃ a₀ a, IsAt f B a₀ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPolynomialMulGaussianOfAnnihilated

theorem **hasDerivAt_line** {g : E → ℂ} {x : E} (hg : DifferentiableAt ℝ g x) (y : E) : HasDerivAt (fun t : ℝ => g (x + t • y)) (fderiv ℝ g x y) 0

theorem **dirD_eq** {g : E → ℂ} (hg : Differentiable ℝ g) (y : E) : dirD y g = fun x => fderiv ℝ g x y

theorem **contDiff_dirD** {g : E → ℂ} (hg : ContDiff ℝ ∞ g) (y : E) : ContDiff ℝ ∞ (dirD y g)

theorem **foldr_ofFn_dirD** {u : E → ℂ} (hu : ContDiff ℝ ∞ u) : ∀ (k : ℕ) (m : Fin k → E), (List.ofFn m).foldr dirD u = fun x => iteratedFDeriv ℝ k u x m

theorem **foldr_dirD_eq_zero** {T : Type} [Fintype T] [DecidableEq T] {W : Type} [NormedAddCommGroup W] [NormedSpace ℝ W] {u : (T → W) → ℂ} (hu : ContDiff ℝ ∞ u) (N : ℕ) (h : ∀ L : List (T × W), L.length = N → L.foldr (fun q g => dirD (Pi.single q.1 q.2) g) u = 0) : ∀ L : List (T → W), L.length = N → L.foldr dirD u = 0

theorem **gam_eq** (z : D.what₂.Tau → Fin V.rank → Fin n → ℂ) : gam V l s z = Complex.exp (expo V l s z)

theorem **contDiff_expo** : ContDiff ℝ ∞ (expo V l s (n

theorem **trace_line** {k : ℕ} (M : Matrix (Fin k) (Fin k) ℂ) (A B : Matrix (Fin k) (Fin n) ℂ) (t : ℝ) : Matrix.trace ((A + t • B)ᴴ * M * (A + t • B)) = Matrix.trace (Aᴴ * M * A) + (t : ℂ) * (Matrix.trace (Bᴴ * M * A) + Matrix.trace (Aᴴ * M * B)) + (t : ℂ) ^ 2 * Matrix.trace (Bᴴ * M * B)

theorem **hasDerivAt_expo** (z : D.what₂.Tau → Fin V.rank → Fin n → ℂ) (τ : D.what₂.Tau) (X : Fin V.rank → Fin n → ℂ) : HasDerivAt (fun t : ℝ => expo V l s (z + t • (Pi.single τ X : D.what₂.Tau → Fin V.rank → Fin n → ℂ))) (-(2 * Real.pi) * ((((|l τ| : ℝ) : ℂ) * (Matrix.trace ((Matrix.of X)ᴴ * (gramAt D V τ * s τ) * Matrix.of (z τ)) + Matrix.trace ((Matrix.of (z τ))ᴴ * (gramAt D V τ * s τ) * Matrix.of X)))) 0

theorem **hasDerivAt_gam** (hM : ∀ τ, (gramAt D V τ * s τ)ᴴ = gramAt D V τ * s τ) (z : D.what₂.Tau → Fin V.rank → Fin n → ℂ) (τ : D.what₂.Tau) (X : Fin V.rank → Fin n → ℂ) : HasDerivAt (fun t : ℝ => gam V l s (z + t • (Pi.single τ X : D.what₂.Tau → Fin V.rank → Fin n → ℂ))) (-(linF V l s τ (Matrix.of X) (Matrix.of (z τ))) * gam V l s z) 0

theorem **annC_gam_mul** (hM : ∀ τ, (gramAt D V τ * s τ)ᴴ = gramAt D V τ * s τ) {h : (D.what₂.Tau → Fin V.rank → Fin n → ℂ) → ℂ} (hh : Differentiable ℝ h) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin n) ℂ) : annC V l s τ X (fun z => gam V l s z * h z) = fun z => gam V l s z * dirD (Pi.single τ (fun i j => X i j) : D.what₂.Tau → Fin V.rank → Fin n → ℂ) h z

theorem **foldr_annC** (hM : ∀ τ, (gramAt D V τ * s τ)ᴴ = gramAt D V τ * s τ) {u : (D.what₂.Tau → Fin V.rank → Fin n → ℂ) → ℂ} (hu : ContDiff ℝ ∞ u) : ∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), ContDiff ℝ ∞ (L.foldr (fun q g => dirD (Pi.single q.1 (fun i j => q.2 i j) : D.what₂.Tau → Fin V.rank → Fin n → ℂ) g) u) ∧ L.foldr (fun q g => annC V l s q.1 q.2 g) (fun z => gam V l s z * u z) = fun z => gam V l s z * L.foldr (fun q g => dirD (Pi.single q.1 (fun i j => q.2 i j) : D.what₂.Tau → Fin V.rank → Fin n → ℂ) g) u z | [] => ⟨hu, rfl⟩ | q :: L => by obtain ⟨h1, h2⟩

theorem **exists_poly_mul_gauss_of_annC** : ∀ (D : D7) (V : HSp D) (n : ℕ) (l : D.what₂.Tau → ℝ) (s : RealMat V), (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ (N : ℕ) (f : (D.what₂.Tau → Fin V.rank → Fin n → ℂ) → ℂ), ContDiff ℝ ∞ f → (∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), L.length = N → L.foldr (fun q g => annC V l s q.1 q.2 g) f = 0) → ∃ p : MvPolynomial (D.what₂.Tau × Fin V.rank × Fin n × Bool) ℂ, ∀ z, f z = evalPoly p (fun τ => Matrix.of (z τ)) * gauss V l s (fun τ => Matrix.of (z τ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPolynomialMulGaussianOfAnnihilatedStatement

theorem **_root_.GlobalTheory.T.exists_polynomial_mul_gaussian_of_annihilated** : ∀ (D : D7) (V : HSp D) (n : ℕ) (l : D.what₂.Tau → ℝ) (s : RealMat V), (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ (N : ℕ) (f : (D.what₂.Tau → Fin V.rank → Fin n → ℂ) → ℂ), ContDiff ℝ ∞ f → (∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), L.length = N → L.foldr (fun q g => annC V l s q.1 q.2 g) f = 0) → ∃ p : MvPolynomial (D.what₂.Tau × Fin V.rank × Fin n × Bool) ℂ, ∀ z, f z = evalPoly p (fun τ => Matrix.of (z τ)) * gauss V l s (fun τ => Matrix.of (z τ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPosNoShortIntegralVectorFromPoint

theorem **norm_mulVec_le** {n : ℕ} (A : Matrix (Fin n) (Fin n) ℂ) (x : Fin n → ℂ) : ‖A *ᵥ x‖ ≤ (∑ i, ∑ j, ‖A i j‖) * ‖x‖

theorem **exists_one_le_norm** {a : I.E} (ha : IsIntegral ℤ a) (h0 : a ≠ 0) : ∃ v, 1 ≤ ‖I.φ v a‖

theorem **coords_eq** (y : I.Ginf) (u : Fin I.n → I.E) (v : InfinitePlace I.F) : (fun i => I.φ v (u i)) = (I.P v * uMat (y v)) *ᵥ vec I y u v

theorem **cst_nonneg** (y : I.Ginf) (v : InfinitePlace I.F) : 0 ≤ cst I y v

theorem **norm_phi_le** (y : I.Ginf) (u : Fin I.n → I.E) (v : InfinitePlace I.F) (i : Fin I.n) : ‖I.φ v (u i)‖ ≤ cst I y v * ‖vec I y u‖

theorem **firstMin_pos** : ∀ (I : AutInst) (y : I.Ginf), ∃ t : ℝ, 0 < t ∧ NoShort I y t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPosNoShortIntegralVectorFromPointStatement

theorem **_root_.GlobalTheory.T.exists_pos_noShortIntegralVectorFromPoint** : ∀ (I : AutInst) (y : I.Ginf), ∃ t : ℝ, 0 < t ∧ NoShort I y t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPowHeckeCharUnramifiedEverywhere

theorem **exists_pow_unrAt** : ∀ (L : Type) [Field L] [NumberField L] (χ : Formalism.L2.Host.HChar L), ∃ N : ℕ, 0 < N ∧ ∀ v : HeightOneSpectrum (𝒪 L), (χ ^ N).UnrAt v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPowHeckeCharUnramifiedEverywhereStatement

theorem **_root_.GlobalTheory.T.exists_pow_heckeChar_unramified_everywhere** : ∀ (L : Type) [Field L] [NumberField L] (χ : Formalism.L2.Host.HChar L), ∃ N : ℕ, 0 < N ∧ ∀ v : HeightOneSpectrum (𝒪 L), (χ ^ N).UnrAt v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPrimeMemPl

theorem **exists_prime_mem** {A : Type} [CommRing A] (P : Ideal A) (hP : P.IsPrime) (n : ℕ) (hn : n ≠ 0) (h : (n : A) ∈ P) : ∃ ℓ : ℕ, ℓ.Prime ∧ (ℓ : A) ∈ P

theorem **exists_prime_mem_pl** {E : Type} [Field E] [NumberField E] (PP : HeightOneSpectrum (𝒪 E)) : ∃ ℓ : ℕ, ℓ.Prime ∧ (ℓ : 𝒪 E) ∈ PP.asIdeal

theorem **GlobalTheory.XGeoPlace.exists_place** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (PP : Pl (I.rec7 l Z).host.E), PP ∈ (pk W).Pg ∨ QAll W PP → ∃ (k : Type) (_ : Field k) (_ : ValuativeRel k) (_ : TopologicalSpace k) (_ : IsNonarchimedeanLocalField k) (_ : MeasurableSpace k) (_ : BorelSpace k) (q : GlobalTheory.FinPl.SplitPt (I.rec7 l Z) k), Formalism.L2.NTheta.IsIdemAt q.e PP ∧ q.v = PP.under (𝒪 (I.rec7 l Z).host.F) ∧ q.UnitAA ∧ valuation k ((GlobalTheory.LocMod.ofSplit q).pr (Global

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsPrimeToDvdAway

theorem **exists_prime_to_dvd_away** {R : Type} [CommRing R] [IsDedekindDomain R] (SS : Finset (HeightOneSpectrum R)) (m : R) (hm : m ≠ 0) : ∃ t : R, t ≠ 0 ∧ (∀ Q ∈ SS, t ∉ Q.asIdeal) ∧ ∀ Q : HeightOneSpectrum R, Q ∉ SS → ∃ u : R, u ∉ Q.asIdeal ∧ ∃ w : R, u * t = m * w

theorem **tmul_mem_of** {E : Type} [Field E] [NumberField E] [StarRing E] {W : Type} [AddCommGroup W] [Module E W] [Module (𝒪 E) W] [IsScalarTower (𝒪 E) E W] (M : Submodule (𝒪 E) W) (SS : Finset (HeightOneSpectrum (𝒪 E))) (c : ℕ) (hc : 0 < c) (y : W) (hy : y ∈ twist₂ M c (expS SS)) (t : 𝒪 E) (ht : ∀ Q : HeightOneSpectrum (𝒪 E), Q ∉ SS → ∃ u : 𝒪 E, u ∉ Q.asIdeal ∧ ∃ w : 𝒪 E, u * t = (c : 𝒪 E) ^ 2 * w) : ((t : E) * (c : E)⁻¹) • y ∈ M

theorem **exists_tmul_mem_perO** : ∀ (D : D7) (a b : ℕ) (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (SS : Finset (HeightOneSpectrum (𝒪 D.host.E))) (c : ℕ), 0 < c → ∀ y ∈ halfBO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ∃ t : 𝒪 D.host.E, t ≠ 0 ∧ (∀ Q ∈ SS, t ∉ Q.asIdeal) ∧ ((t : D.host.E) * (c : D.host.E)⁻¹) • y ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsQuotientIntegrableFoldUnimodular

theorem **lintegral_eq_of_integral** {α β γ : Type} [MeasurableSpace α] [MeasurableSpace β] [MeasurableSpace γ] {m₁ : Measure α} {m₂ : Measure β} {a : α → γ} {b : β → γ} (ha : Measurable a) (hb : Measurable b) (h : ∀ f : γ → ℂ, Measurable f → (Integrable (fun x => f (a x)) m₁ ↔ Integrable (fun y => f (b y)) m₂) ∧ (Integrable (fun y => f (b y)) m₂ → ∫ x, f (a x) ∂m₁ = ∫ y, f (b y) ∂m₂)) (F : γ → ℝ≥0∞) (hF : Measurable F) : ∫⁻ x, F (a x) ∂m₁ = ∫⁻ y, F (b y) ∂m₂

theorem **setLIntegral_mul_right** {R : Type} [Group R] [MeasurableSpace R] [MeasurableMul R] {μ : Measure R} [μ.IsMulLeftInvariant] [μ.IsMulRightInvariant] {Γ : Subgroup R} [Countable Γ] {𝓡 : Set R} (h : IsFundamentalDomain Γ 𝓡 μ) (q : R → ℝ≥0∞) (hq : ∀ γ ∈ Γ, ∀ r, q (γ * r) = q r) (m : R) : ∫⁻ r in 𝓡, q (r * m) ∂μ = ∫⁻ r in 𝓡, q r ∂μ

theorem **exists_quotientIntegrable_of_le_mean** (hΓR : Γ ≤ R) (hR : IsClosed (R : Set G)) [Countable Γ] [DiscreteTopology Γ] (h𝓡 : IsFundamentalDomain (Γ.subgroupOf R) 𝓡 μR) (h𝓑 : IsFundamentalDomain Γ 𝓑 μ) {K : G → ℝ≥0∞} (hK : Measurable K) (hKΓ : ∀ γ ∈ Γ, ∀ g, K (γ * g) = K g) (hfin : ∫⁻ g in 𝓑, K g ∂μ < ∞) {f : G → ℂ} (hf : Measurable f) (hfR : ∀ r ∈ R, ∀ g, f (r * g) = f g) (hle : ∀ h, ‖f h‖ₑ ≤ ∫⁻ r in 𝓡, K (r * h) ∂μR) : ∃ z, IsQuotIntL1 μR μ f z

theorem **exists_quotientIntegrable_fold_unimodular** (hΓR : Γ ≤ R) (hR : IsClosed (R : Set G)) [Countable Γ] [DiscreteTopology Γ] (h𝓡 : IsFundamentalDomain (Γ.subgroupOf R) 𝓡 μR) (h𝓑 : IsFundamentalDomain Γ 𝓑 μ) {U : Type} [Group U] [MeasurableSpace U] [MeasurableMul U] [MeasurableInv U] {ν : Measure U} [SFinite ν] [ν.IsMulLeftInvariant] [ν.IsInvInvariant] (j : U →* G) (hj : Measurable j) (hjR : ∀ u, j u ∈ R) {q : G → G → ℝ≥0∞} (hq : Measurable (Function.uncurry q)) (hqΓ : ∀ γ ∈ Γ, ∀ g, ∀ r ∈ R, q (γ * g) (γ * r) = q g r) (hqR : ∀ r ∈ R, ∃ v, ∀ g u, q g (r * j u) = q g (j (v * u))) (hfin : ∫⁻ g in 𝓑, ∫⁻ u, q g (j u) ∂ν ∂μ < ∞) {P : G → ℂ} (hP : Measurable P) (hPR : ∀ r ∈ R, ∃ v, ∀ g u, P (j u * (r * g)) = P (j (u * v) * g)) (hle : ∀ h, ‖P h‖ₑ ≤ ∫⁻ r in 𝓡, q (r * h) r ∂μR) : ∃ z, IsQuotIntL1 μR μ (fun h => ((∫ u, ‖P (j u * h)‖ ∂ν : ℝ) : ℂ)) z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRatOfSameOrb

theorem **image_val_mem_nhds** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {B : Type} [Fintype B] [DecidableEq B] {S : Set (GL B k)} (hS : IsOpen S) {h : GL B k} (hh : h ∈ S) : Units.val '' S ∈ nhds h.val

theorem **exists_rat_of_sameOrb** : ∀ {E k : Type} [Field E] [NumberField E] [StarRing E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (P : HeightOneSpectrum (𝒪 E)) (ι : E →+* k) (ϖ : kˣ), GlobalTheory.SymBridge.IsCompl P (qOf P) ι ϖ → ∀ {A : Type} [Fintype A] [DecidableEq A] {n : ℕ} (e : A ≃ Fin n) (Jl : Subgroup (GL (A ⊕ Fin 0) k)), IsOpen (Jl : Set (GL (A ⊕ Fin 0) k)) → Jl ≤ GlobalTheory.SplitWeil.KO k (A ⊕ Fin 0) → ∀ (M : ℕ) (c : (A → k) ⧸ GlobalTheory.StepData.box A (ϖ ^ M)) (y₀ : Sp E n), GlobalTheory.QPl.SameOrb Jl (GlobalTheory.StepData.box A (ϖ ^ M)) c (Submodule.Quotient.mk fun a => ι (y₀ (e a))) → ∃ r ∈ GLint E n P, ∃ g ∈ Jl, g.val.toBlocks₁₁ = GlobalTheory.SymBridge.toLoc ι e e r.toLinearMap ∧ ∀ y : Sp E n, ((qOf P : E) ^ M)⁻¹ • (r.symm y - y₀) ∈ stdAt E n P → c = Submodule.Quotient.mk fun a => ι (y (e a))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRatSubTwoNsmul

theorem **two_nsmul_cancel** {D : D7} (a b : AE D) (h : (2 : ℕ) • a = (2 : ℕ) • b) : a = b

theorem **map_eq_zero_apply** {D : D7} {S T : HSp D} {R : Type} [Semiring R] (f : AE D →+* R) {x : WO S T} (hx : x.map f = 0) (i : Fin T.rank) (j : Fin S.rank) : f (x i j) = 0

theorem **eq_two_nsmul** {D : D7} {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {x l : WO S T} (hx : x.map (infPart D) = 0) (hl : l ∈ latO S T LS LT fun _ => True) (hf : (x - (2 : ℕ) • l).map (finPart D) = 0) : x = (2 : ℕ) • l

theorem **exists_rat_sub_two_nsmul** : ∀ (D : D7) (S T : HSp D) (ΛS ΛS' : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT ΛT' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC S ΛS' → IsSelfDualC T ΛT → IsSelfDualC T ΛT' → ∀ u ∈ (latO S T (ΛS' : Set (Fin S.rank → D.host.E)) (ΛT' : Set (Fin T.rank → D.host.E)) fun _ => True), ∃ γ, ∃ l ∈ (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True), l ∈ (latO S T (ΛS' : Set (Fin S.rank → D.host.E)) (ΛT' : Set (Fin T.rank → D.host.E)) fun _ => True) ∧ u - GlobalTheory.LatVac.finTens 1 γ = 2 • l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalAdelicPointOfUnitaryBaseSpace

theorem **exists_GQ_of_unitary** {D : D7} (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) {n : ℕ} (hn : V.rank = n) (γ : GL (Fin n) D.host.E) (hγ : γ ∈ Formalism.UG.UJ (Formalism.L2.Forms.sigD D : D.host.E →+* D.host.E) (Matrix.reindex (finCongr hn) (finCongr hn) (HSpace.gram V))) : ∃ a : GA D V, a ∈ GQ V ∧ GlobalTheory.AdForm.finM V hn a = (Formalism.UG.bc (Formalism.UG.inclE (F

theorem **exists_GQ_of_unitary0** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ γ : GL (Fin (D.R.N 0)) D.host.E, γ ∈ Formalism.UG.UJ (Formalism.L2.Forms.sigD D : D.host.E →+* D.host.E) (Formalism.L2.Forms.gramN D 0 (Nat.zero_le _)) → ∃ a : GA D (D.sp.X 0), a ∈ GQ (D.sp.X 0) ∧ GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) a = (Formalism.UG.bc (Formalism.UG.inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalAdelicPointOfUnitaryBaseSpaceStatement

theorem **_root_.GlobalTheory.T.exists_rationalAdelicPoint_of_unitary_baseSpace** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ γ : GL (Fin (D.R.N 0)) D.host.E, γ ∈ Formalism.UG.UJ (Formalism.L2.Forms.sigD D : D.host.E →+* D.host.E) (Formalism.L2.Forms.gramN D 0 (Nat.zero_le _)) → ∃ a : GA D (D.sp.X 0), a ∈ GQ (D.sp.X 0) ∧ GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) a = (Formalism.UG.bc (Formalism.UG.inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalDoubleCosetRepresentativeBounds

theorem **val_unif** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ϖ : kˣ} (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) : valuation k (ϖ : k) = uniformizer k

theorem **denseRange_toLoc** {E k : Type} [Field E] [Field k] [TopologicalSpace k] {ι : E →+* k} (hι : DenseRange ι) {A B : Type} {a b : ℕ} (e : A ≃ Fin a) (e' : B ≃ Fin b) : DenseRange (toLoc ι e e')

theorem **exists_bounded_rep** : ∀ (R : Ranks) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ j c (h : j + 1 + c = s) (g : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin (c + 1 + 1)) k), (∃ r ∈ GlobalTheory.RootEx.R k (XIx (Fin R.k) (Fin R.t) j) (c + 1 + 1) 1, GlobalTheory.IndefiniteRowSideFullHecke.NormalRepresentationLattice.IsNormalRep (W.T s R) W.ϖ j (c + 1) (by omega) (r * g)) → ∀ h' : GL (XIx (Fin R.k) (Fin R.t) (j + 1) ⊕ Fin (c + 1)) k, flat W.ψ ν ((W.T s R).mu (j + 1)) ((W.T s R).Phi j (c + 1) (by omega)) g h' ≠ 0 → ∃ g₁ : GL (XIx (Fin R.k) (Fin R.t) (j + 1) ⊕ Fin (c + 1)) k, DoubleCoset.mk (GlobalTheory.RootEx.R k (XIx (Fin R.k) (Fin R.t) (j + 1)) (c + 1) 1) ((W.T s R).lev (j + 1) (c + 1) (by omega)).J g₁ = DoubleCoset.mk (GlobalTheory.RootEx.R k (XIx (Fin R.k)

theorem **exists_toLocGL_mem** : ∀ (E k : Type) [Field E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : E →+* k), DenseRange ι → ∀ (A : Type) [Fintype A] [DecidableEq A] (a : ℕ) (e : A ≃ Fin a) (U : Set (GL A k)), IsOpen U → U.Nonempty → ∃ γ : (Fin a → E) ≃ₗ[E] (Fin a → E), toLocGL ι e γ ∈ U

theorem **exists_toLoc_ne_zero** : ∀ (E k : Type) [Field E] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : E →+* k), DenseRange ι → ∀ (A B : Type) [Fintype A] [Fintype B] (a b : ℕ) (e : A ≃ Fin a) (e' : B ≃ Fin b) (f : Matrix A B k → ℂ), IsLocallyConstant f → f ≠ 0 → ∃ x : (Fin b → E) →ₗ[E] (Fin a → E), f (toLoc ι e e' x) ≠ 0

theorem **toLoc_mulVec** {E k : Type} [Field E] [Field k] (ι : E →+* k) {A : Type} [Fintype A] {a : ℕ} (e : A ≃ Fin a) (x : (Fin a → E) →ₗ[E] (Fin a → E)) (y : Fin a → E) : (toLoc ι e e x *ᵥ fun i => ι (y (e i))) = fun i => ι (x y (e i))

theorem **bounds_of_isGL** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (P : HeightOneSpectrum (𝒪 X.E)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (ϖ : kˣ), IsCompl P (qOf P) ι ϖ → P ∉ Pr.vGs → ∀ (j c n : ℕ) (γ : AutY X R j) (g' : GL (XIx (Fin R.k) (Fin R.t) j ⊕ Fin c) k), IsGL ι γ g' → latOf 1 ≤ latOf g' → (∀ y ∈ latOf g', ((ϖ : k) ^ n) • y ∈ latOf 1) → ∀ x ∈ lamYat (Pr

theorem **ordX_exists** (R : Ranks) (j : ℕ) : ∃ e : XIx (Fin R.k) (Fin R.t) j ≃ Fin (R.N j), IsOrd xPos e

theorem **GlobalTheory.TwBridge.repsOnto** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalDoubleCosetRepresentativeBoundsStatement

theorem **_root_.GlobalTheory.T.exists_rational_double_coset_representative_bounds** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalMatrixOfMemPeriodLattice

theorem **exists_mat_of_mem_lamC** : ∀ (D : D7) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)), ∀ x ∈ LamC (tauE D) δ (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N', ∃ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, toHom S T hS hT γ ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' ∧ embH (tauE D) δ (toHom S T hS hT γ) = x ∧ ∃ v, GlobalTheory.AdTh.IsFinOf S T γ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalMatrixOfMemPeriodLatticeStatement

theorem **_root_.GlobalTheory.T.exists_rationalMatrix_of_mem_periodLattice** : ∀ (D : D7) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)), ∀ x ∈ LamC (tauE D) δ (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N', ∃ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, toHom S T hS hT γ ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' ∧ embH (tauE D) δ (toHom S T hS hT γ) = x ∧ ∃ v, GlobalTheory.AdTh.IsFinOf S T γ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalOrbitPointEqBesselBasePoint

theorem **mul_mul_mul_of_gram** {a b c e n : Type*} [Fintype b] [Fintype c] [Fintype e] {R : Type*} [Semiring R] {A : Matrix a b R} {G : Matrix b c R} {M : Matrix c e R} {C : Matrix a e R} (h : A * G * M = C) (N : Matrix e n R) : A * (G * (M * N)) = C * N

theorem **ell_succ_eq** : ell D (σ + 1) = fun i => incA D (lastE D σ i)

theorem **zl_eq_rat** (B : BesLev D σ) : zl B = rat (vecMulVec B.ζ (lastE D σ))

theorem **rat_x0Rat** (B : BesLev D σ) : rat (besselRationalBasePointMatrix B) = x0 B

theorem **gram_x0Rat** (B : BesLev D σ) : (besselRationalBasePointMatrix B)ᴴ * B.V.gram * besselRationalBasePointMatrix B = (vecMulVec B.ζ (lastE D σ))ᴴ * B.L.gram * vecMulVec B.ζ (lastE D σ)

theorem **gramSkew_x0Rat** (B : BesLev D σ) : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew B.V (rat (besselRationalBasePointMatrix B) : Mx D B.V.rank (σ + 1)) = GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew B.L (zl B)

theorem **linearIndependent_x0Rat** (B : BesLev D σ) : LinearIndependent D.host.E (besselRationalBasePointMatrix B).transpose

theorem **exists_rationalOrbit_point_eq_besselBasePoint** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), ∃ x ∈ orb B, (rat x : Mx D B.V.rank (σ + 1)) = x0 B

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalOrbitPointEqBesselBasePointStatement

theorem **_root_.GlobalTheory.T.exists_mem_orb_rat_eq_x0** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), ∃ x ∈ orb B, (rat x : Mx D B.V.rank (σ + 1)) = x0 B

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalRepLevelDecompositionFiniteUnitary

theorem **exists_isDec_finM** : ∀ (D : D7) (u : D.what₂.Word) (F : Formalism.L2.Forms.Forms0 D u) (g : GA D (D.sp.X 0)), ∃ p, GlobalTheory.AdForm.IsDec F.cls0.lev F.cls0.y (GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) g) p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalRepLevelDecompositionFiniteUnitaryStatement

theorem **_root_.GlobalTheory.T.exists_rationalRepLevelDecomposition_finiteUnitary** : ∀ (D : D7) (u : D.what₂.Word) (F : Formalism.L2.Forms.Forms0 D u) (g : GA D (D.sp.X 0)), ∃ p, GlobalTheory.AdForm.IsDec F.cls0.lev F.cls0.y (GlobalTheory.AdForm.finM (D.sp.X 0) (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)) g) p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalWittFlagCoordinatesOfMemUnipotent

theorem **coords_rat** {D : D7} {V V₀ : HSp D} {ρ : ℕ} (d : HypDec V V₀ ρ) (c' : skewA D ρ) (b' : Mx D ρ V₀.rank) (u : GL (Fin ρ) (AE D)) (hQ : d.nC c' * (d.nB b' * d.levi u) ∈ GQ V) : c'.1 ∈ ratMx D ρ ρ ∧ b' ∈ ratMx D ρ V₀.rank ∧ u.1 ∈ ratMx D ρ ρ ∧ (u⁻¹).1 ∈ ratMx D ρ ρ

theorem **rat_coords_of_mem_flagN** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (γ : GA D V), γ ∈ GQ V → γ ∈ flagN d → ∃ (c : skewA D ρ) (b : Mx D ρ V₀.rank) (u : UX D ρ), γ = d.nC c * (d.nB b * d.levi u.1) ∧ c.1 ∈ ratMx D ρ ρ ∧ b ∈ ratMx D ρ V₀.rank ∧ u ∈ UXQ D ρ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRationalWittFlagCoordinatesOfMemUnipotentStatement

theorem **_root_.GlobalTheory.T.exists_rational_wittFlagCoordinates_of_mem_unipotent** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (γ : GA D V), γ ∈ GQ V → γ ∈ flagN d → ∃ (c : skewA D ρ) (b : Mx D ρ V₀.rank) (u : UX D ρ), γ = d.nC c * (d.nB b * d.levi u.1) ∧ c.1

∈ ratMx D ρ ρ ∧ b ∈ ratMx D ρ V₀.rank ∧ u ∈ UXQ D ρ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealCentreCharacterOnDefiniteVacuum

theorem **lintegral_enorm_sq_eq_one** {α : Type} [MeasurableSpace α] {μ : Measure α} {s : Set α} {f : α → ℂ} (h : ∫ w in s, ‖f w‖ ^ 2 ∂μ = 1) : ∫⁻ w in s, ‖f w‖ₑ ^ 2 ∂μ = 1

theorem **om_vac_eq_smul** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} (hθ : IsInfFam θ) {S T : HSp D} {x : DefSp S} {y : DefSp T} (hx : x.Is) (hy : y.Is) {k : KerDat S T} (hk : k.Is ψ θ x y) {z : GA D S} (hz : z ∈ GinfC S) : ∃ c : ℂ, ‖c‖ = 1 ∧ k.om (z, 1) k.vac = c • k.vac

theorem **exists_center_char** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T), x.Is → y.Is → ∀ k : KerDat S T, k.Is ψ θ x y → ∃ c : GA D S → ℂ, ∀ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) → ‖c z‖ = 1 ∧ k.om (z, 1) k.vac = c z • k.vac

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealCentreCharacterOnDefiniteVacuumStatement

theorem **_root_.GlobalTheory.T.exists_realCentre_character_on_definiteVacuum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T), x.Is → y.Is → ∀ k : KerDat S T, k.Is ψ θ x y → ∃ c : GA D S → ℂ, ∀ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) → ‖c z‖ = 1 ∧ k.om (z, 1) k.vac = c z • k.vac

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealDiscreteSeriesDataThetaProperties

theorem **exists_record_theta_all** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∃ X : ProvA, ChosenSeries X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFirst X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisSecond X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisUpDownLine X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFourth X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFifth X

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealDiscreteSeriesDataThetaPropertiesStatement

theorem **_root_.GlobalTheory.T.exists_real_discreteSeries_data_theta_properties** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∃ X : ProvA, ChosenSeries X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFirst X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisSecond X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisUpDownLine X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFourth X ∧ Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFifth X

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealIntertwinerNeZeroKFiniteTensor

theorem **exists_ne_zero_inclK** : GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)) (pt : PolF D W.rank V.rank → (WO V W → ℂ)) (It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M), IsItW θ M M' ω pt It → ∀ h' : GlobalTheory.Chain.IoLift.KSm M', ∃ φ y, It φ (GlobalTheory.Chain.IoLift.inclK h' y) ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealIntertwinerNeZeroKFiniteTensorStatement

theorem **_root_.GlobalTheory.T.exists_realIntertwiner_ne_zero_KFinite_tensor** : GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)) (pt : PolF D W.rank V.rank → (WO V W → ℂ)) (It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M), IsItW θ M M' ω pt It → ∀ h' : GlobalTheory.Chain.IoLift.KSm M', ∃ φ y, It φ (GlobalTheory.Chain.IoLift.inclK h' y) ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealParOfIsCharAF

theorem **exists_realPar_of_isCharAF** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ θ : InfFam D, IsInfFam θ → ∃ l : D.what₂.Tau → ℝ, HasRealPar ψ θ l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealPlacesIdempotentExtensionAdeles

private theorem **realIdem_map** {R S : Type} [CommRing R] [CommRing S] [Algebra F R] [Algebra F S] (f : R →ₐ[F] S) (σ : E →ₐ[F] E) (x : R) : (Algebra.TensorProduct.map f σ).toRingHom (x ⊗ₜ[F] (1 : E)) = f x ⊗ₜ[F] (1 : E)

private theorem **realIdem_gen** (σ : E →ₐ[F] E) : ∃ e : AF F ⊗[F] E, e * e = e ∧ sigR (AF F) σ e = e ∧ (∀ v, compFin F E v e = 0) ∧ (∀ w, compInf F E w e = 1) ∧ (Algebra.TensorProduct.map (sndA F) (AlgHom.id F E)).toRingHom e = 0 ∧ (Algebra.TensorProduct.map (fstA F) (AlgHom.id F E)).toRingHom e = 1

theorem **realIdem_exists** : ∀ D : D7, ∃ e : AE D, e * e = e ∧ cA D e = e ∧ (∀ v, compFin D.host.F D.host.E v e = 0) ∧ (∀ w, compInf D.host.F D.host.E w e = 1) ∧ finPart D e = 0 ∧ infPart D e = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealPlacesIdempotentExtensionAdelesStatement

theorem **_root_.GlobalTheory.T.exists_realPlacesIdempotent_extensionAdeles** : ∀ D : D7, ∃ e : AE D, e * e = e ∧ cA D e = e ∧ (∀ v, compFin D.host.F D.host.E v e = 0) ∧ (∀ w, compInf D.host.F D.host.E w e = 1) ∧ finPart D e = 0 ∧ infPart D e = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealThetaIntertwinerExchangedPair

theorem **itPairUp** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (k : Fin D.R.m) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ), S = parB (Ptw D) k → S' = parB (Ptw D) (k + 1) → IsInfFam θ → ChainPar D k → ∃ It : PolF D V.rank W.rank →ₗ[ℂ] SmT M →ₗ[ℂ] SmT M', ∀ (ω : (GA D W × GA D V →* Module.End ℂ (WO W V → ℂ))) (f : WO W V → ℂ), IsWeilExp ψ W V (expT D k) (expS D k) ω → IsVacVec ψ θ M' M f → IsItW θ M' M ω (ptOf ψ θ M' M f) It

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRealThetaIntertwinerExchangedPairStatement

theorem **_root_.GlobalTheory.T.exists_realThetaIntertwiner_exchangedPair** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (k : Fin D.R.m) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ), S = parB (Ptw D) k → S' = parB (Ptw D) (k + 1) → IsInfFam θ → ChainPar D k → ∃ It : PolF D V.rank W.rank →ₗ[ℂ] SmT M →ₗ[ℂ] SmT M', ∀ (ω : (GA D W × GA D V →* Module.End ℂ (WO W V → ℂ))) (f : WO W V → ℂ), IsWeilExp ψ W V (expT D k) (expS D k) ω → IsVacVec ψ θ M' M f → IsItW θ M' M ω (ptOf ψ θ M' M f) It

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRedIffIsPat

theorem **exists_red_iff_isPat** : ∀ {D : D7} {c : Comm D} {v : VCom c} (G : GLoc v) (PP : Formalism.L2.DIV.Pl D.host.E), Formalism.L2.NTheta.IsIdemAt G.q.e PP → ∀ g : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP), (∃ x, GlobalTheory.LevZero.Red G PP x g) ↔ GlobalTheory.SymRes.IsPat D.R.rS D.R.k g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsReducedBasisBoundedTypeOfInstance

theorem **vec_add** (y : I.Ginf) (u w : Fin I.n → I.E) : vec I y (u + w) = vec I y u + vec I y w

theorem **vec_sum_zsmul** (y : I.Ginf) {N : ℕ} (z : Fin N → ℤ) (e : Fin N → Fin I.n → I.E) : vec I y (∑ j, z j • e j) = ∑ j, z j • vec I y (e j)

theorem **gm_mulVec** {N : ℕ} (ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)) (y : I.Ginf) (e : Fin N → Fin I.n → I.E) (z : Fin N → ℤ) : gm I ρ y e *ᵥ (fun j => (z j : ℝ)) = ρ (vec I y (∑ j, z j • e j))

theorem **gm_change** {N : ℕ} (ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)) (y : I.Ginf) (e : Fin N → Fin I.n → I.E) (M : Matrix (Fin N) (Fin N) ℤ) : gm I ρ y (fun j => ∑ i, M i j • e i) = gm I ρ y e * M.map (Int.cast : ℤ → ℝ)

theorem **sum_change** {N : ℕ} (M : Matrix (Fin N) (Fin N) ℤ) (e : Fin N → Fin I.n → I.E) (w : Fin N → ℤ) : ∑ j, w j • (∑ i, M i j • e i) = ∑ i, (M *ᵥ w) i • e i

theorem **isZBasis_change** {N : ℕ} (e : Fin N → Fin I.n → I.E) (he : IsZBasis I e) (M : Matrix (Fin N) (Fin N) ℤ) (hM : IsUnit M.det) : IsZBasis I (fun j => ∑ i, M i j • e i)

theorem **norm_le_sqrt** {N : ℕ} {ρ : (InfinitePlace I.F → Fin I.n → ℂ) ≃ₗ[ℝ] (Fin N → ℝ)} (hρ : IsRealCoord I ρ) (x : InfinitePlace I.F → Fin I.n → ℂ) : ‖x‖ ≤ Real.sqrt N * ‖ρ x‖

theorem **abs_le_entSum** {N : ℕ} (X : Matrix (Fin N) (Fin N) ℝ) (i j : Fin N) : |X i j| ≤ entSum X

theorem **smul_conj** {N : ℕ} (l c : ℝ) (A X : Matrix (Fin N) (Fin N) ℝ) : (l • A)ᵀ * (c • X) * (l • A) = (l * c * l) • (Aᵀ * X * A)

theorem **smul_conj'** {N : ℕ} (l : ℝ) (A B : Matrix (Fin N) (Fin N) ℝ) : (l • A) * B * (l • A)ᵀ = (l * l) • (A * B * Aᵀ)

theorem **GlobalTheory.CuspGeo.redBasis_holds** : ∀ I : AutInst, RedBasis I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsReducedBasisBoundedTypeOfInstanceStatement

theorem **_root_.GlobalTheory.T.existsReducedBasisBoundedType_of_instance** : ∀ I : AutInst, RedBasis I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsReductionIffPattern

theorem **exists_red_iff_isPat** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {O : Type} [CommRing O] (I : Ideal O) (π : O →+* k) (h : GlobalTheory.FunObj.Comparison₂.IsRed (Ideal.Quotient.mk I) π) {s κ r : ℕ} (hr : r = κ + (s + 1)) (S : GlobalTheory.SymChain.SymData k s κ) (g : GL (Fin r) (O ⧸ I)) : (∃ x : S.P.J, ∀ (a b : Fin κ ⊕ Fin (s + 1)) (a' b' : Fin r), a'.val = GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.framePos _ _ a → b'.val = GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.framePos _ _ b → ∀ z : O, Ideal.Quotient.mk I z = g.val a' b' → valuation k ((x : GL (Fin κ ⊕ Fin (s + 1)) k).val a b - π z) < 1) ↔ IsPat S.ρ κ g

theorem **exists_red_iff** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (g : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP)), (∃ x, G.Red x g) ↔ GlobalTheory.SymRes.IsPat D.R.rS D.R.k g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsReductionIffPatternStatement

theorem **_root_.GlobalTheory.T.exists_reduction_iff_pattern** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (g : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP)), (∃ x, G.Red x g) ↔ GlobalTheory.SymRes.IsPat D.R.rS D.R.k g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsReductionOfMemLevelZero

theorem **mt_mul'** (V : HSp D) (g h : GA D V) : mt V (g * h) = mt V g * mt V h

theorem **mt_one'** (V : HSp D) : mt V 1 = 1

theorem **finE_val'** (V : HSp D) (g : GA D V) : (Lev.finE V g).val = (mt V g).map (toEAf D)

theorem **exists_lift** (z : EAf D) : ∃ w : AE D, toEAf D w = z

theorem **frGL_mul** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) (y : GL (Fin n) (Formalism.L2.Herm.intAf F E)) : (Formalism.L2.DIV.frGL β y).val * β.val = β.val * (Formalism.L2.DIV.intGL F E n y).val

theorem **eq_map_of_moved** {S K n : Type} [CommRing S] [CommRing K] [Fintype n] [DecidableEq n] (π : S →+* K) (β : GL n S) (x y : Matrix n n S) (h1 : x * β.val = β.val * y) (T T' H N M : Matrix n n K) (hT : T' * T = 1) (hN : T * β.val.map π = N) (hM : H * N = N * M) (hx : T * x.map π = H * T) : M = y.map π

theorem **red_of_movedTo** (q : SplitPt D k) (PP : Formalism.L2.DIV.Pl D.host.E) (hPP : Formalism.L2.NTheta.IsIdemAt q.e PP) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (t h : GA D V) (N M : Matrix (Fin V.rank) (Fin V.rank) k) (hN : MovedTo (ofSplit q) V t β N) (hM : (mt V h).map q.pr * N = N * M) (hx : Lev.finE V (t⁻¹ * h * t) ∈ Formalism.L2.DIV.St β) (i j : Fin V.rank) (z : 𝒪 D.host.E) (hz : Ideal.Quotient.mk PP.asIdeal z = (Formalism.L2.DIV.redFn β PP (Lev.finE V (t⁻¹ * h * t))).val i j) : valuation k (M i j - (ofSplit q).pr (z : D.host.E)) < 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsReindexIff

theorem **reGL_embX** (e : n ≃ m) (g : GL n k) : reGL (sE e c) (embX c g) = embX c (reGL e g)

theorem **reGL_embA** (e : n ≃ m) (g : GL (Fin c) k) : reGL (sE e c) (embA n g) = embA m g

theorem **reGL_mem_N** (e : n ≃ m) {p : ℕ} {g : GL (n ⊕ Fin c) k} : reGL (sE e c) g ∈ N k m c p ↔ g ∈ N k n c p

theorem **reGL_mem_R** (e : n ≃ m) {p : ℕ} {g : GL (n ⊕ Fin c) k} (hg : g ∈ R k n c p) : reGL (sE e c) g ∈ R k m c p

theorem **mul_embX_mem_R** {p : ℕ} {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c p) (g : GL n k) : u * embX c g ∈ R k n c p

theorem **prX_inf_eq_iff** (J : Subgroup (GL (n ⊕ Fin c) k)) : prX (R k n c 1 ⊓ J) = J.comap (embX c) ↔ ∀ u ∈ N k n c 1, ∀ g, u * embX c g ∈ J → embX c g ∈ J

theorem **Cgrp_one** (J : Subgroup (GL (n ⊕ Fin c) k)) : Cgrp J 1 = R k n c 1 ⊓ J

theorem **reGL_conj** {ι ι' : Type*} [Fintype ι] [DecidableEq ι] [Fintype ι'] [DecidableEq ι'] (F : LatFrame k ι n) (F' : LatFrame k ι' m) (a : ι ≃ ι') (b : n ≃ m) (ho : a.trans F'.ord = F.ord.trans b) (hb : F'.bas = reGL b F.bas) (g : GL ι k) : F'.conj (reGL a g) = reGL b (F.conj g)

theorem **coords_submatrix** {ι ι' κ κ' : Type*} (F : LatFrame k ι n) (F' : LatFrame k ι' m) (a : ι ≃ ι') (b : n ≃ m) (ho : a.trans F'.ord = F.ord.trans b) (hb : F'.bas = reGL b F.bas) (G : LatFrame k κ n') (G' : LatFrame k κ' m') (a' : κ ≃ κ') (b' : n' ≃ m') (ho' : a'.trans G'.ord = G.ord.trans b') (hb' : G'.bas = reGL b' G.bas) (x : Matrix m m' k) : (F'.coords G' x).submatrix a a' = F.coords G (x.submatrix b b')

theorem **om_submatrix** (μ : kˣ →* ℂˣ) (E : n ≃ m) (E' : n' ≃ m') (h : GL n k) (Φ : Matrix n n' k → ℂ) : om μ 1 (reGL E h) (fun x => Φ (x.submatrix E E')) = fun x => om μ 1 h Φ (x.submatrix E E')

theorem **opE_submatrix** [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k) [ν.IsAddHaarMeasure] (p : ℕ) (e : n ≃ m) (e' : n' ≃ m') (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (x₀ : Matrix m m' k) : opE ψ ν p (fun x => Φ (x.submatrix (sE e c) (sE e' (c + 1)))) x₀ = opE ψ ν p Φ (x₀.submatrix e e')

theorem **td_ext** {P Q : TypeDatum k n} (hJ : ∀ g, g ∈ P.J ↔ g ∈ Q.J) (hl : ∀ g (h : g ∈ P.J) (h' : g ∈ Q.J), P.lam ⟨g, h⟩ = Q.lam ⟨g, h'⟩) : P = Q

theorem **reindex_trans** (P : TypeDatum k n) (e : n ≃ m) (e' : m ≃ n') : (P.reindex e).reindex e' = P.reindex (e.trans e')

theorem **reGL_symm_mem_KO** (e : n ≃ m) {g : GL m k} : (reGL e).symm g ∈ KO k n ↔ g ∈ KO k m

theorem **exists_reindex_iff** (P : TypeDatum k n) (e : n ≃ m) (M : Matrix m m k) : (∃ g ∈ (P.reindex e).J, g.val = M) ↔ ∃ g ∈ P.J, g.val = M.submatrix e e

theorem **maxM_submatrix** (e : n ≃ m) (e' : n' ≃ m') (M : Matrix m m' k) : MaxM (M.submatrix e e') ↔ MaxM M

theorem **det_reGL_symm** (e : n ≃ m) (g : GL m k) : Matrix.GeneralLinearGroup.det ((reGL e).symm g) = Matrix.GeneralLinearGroup.det g

theorem **step_reindex** (P : TypeDatum k n) (e : n ≃ m) (f : n' ≃ m') (μ : kˣ →* ℂˣ) (hμ : IsTame μ) : (P.reindex e).step m' μ hμ = (P.step n' μ hμ).reindex (e.sumCongr f)

theorem **Phi_reindex** (P : TypeDatum k n) (e : n ≃ m) (f : n' ≃ m') (μ : kˣ →* ℂˣ) (x : Matrix (m ⊕ m') m k) : (P.reindex e).Phi m' μ x = P.Phi n' μ (x.submatrix (e.sumCongr f) e)

theorem **ixSucc_eI** (r : Type*) {t t' : Type*} (f : t ≃ t') (j : ℕ) : ((eI r f j).sumCongr f).trans (ixSucc r t' j) = (ixSucc r t j).trans (eI r f (j + 1))

theorem **iter_reindex** (f : t ≃ t') (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) : P.iter t' μ hμ j = (P.iter t μ hμ j).reindex (eI r f j)

theorem **iterPhi_reindex** (f : t ≃ t') (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) (x : Matrix (Ix r t' (j + 1)) (Ix r t' j) k) : P.iterPhi t' μ hμ j x = P.iterPhi t μ hμ j (x.submatrix (eI r f (j + 1)) (eI r f j))

theorem **stdOrd_eI** (κ : Type*) {t t' : Type*} (f : t ≃ t') {l : ℕ} (j c : ℕ) (h : j + c = l) : (eI (κ ⊕ Fin l) f j).trans (stdOrd κ t' j c h) = (stdOrd κ t j c h).trans (sE (eI (κ ⊕ Fin j) f j) c)

theorem **typeChain_lev** (f : t ≃ t') (ϖ : kˣ) (P : TypeDatum k (κ ⊕ Fin (s + 1))) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j c : ℕ) (h : j + c = s + 1) (g : GL (XIx κ t j ⊕ Fin c) k) : reGL (sE (eI (κ ⊕ Fin j) f j) c) g ∈ ((typeChain t' ϖ P μ hμ).lev j c h).J ↔ g ∈ ((typeChain t ϖ P μ hμ).lev j c h).J

theorem **typeChain_Phi** (f : t ≃ t') (ϖ : kˣ) (P : TypeDatum k (κ ⊕ Fin (s + 1))) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j c : ℕ) (h : j + c = s) : (typeChain t' ϖ P μ hμ).Phi j c h = fun x => (typeChain t ϖ P μ hμ).Phi j c h (x.submatrix (sE (eI (κ ⊕ Fin (j + 1)) f (j + 1)) c) (sE (eI (κ ⊕ Fin j) f j) (c + 1)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRingEquivAddOne

theorem **shiftVars_X** (ι k : Type) [CommRing k] (i : ι) : shiftVars ι k (MvPolynomial.X i) = MvPolynomial.X i + 1

theorem **exists_ringEquiv_add_one** {Ω : Type} [Field Ω] [IsAlgClosed Ω] (k : Subfield Ω) (c : Ω) (hc : Transcendental (ɪk) c) : ∃ (σ : Ω ≃+* Ω), (∀ a ∈ k, (σ : Ω → Ω) a = a) ∧ (σ : Ω → Ω) c = c + (1 : Ω)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRingEquivPadicOfResidueCardUniformizer

theorem **natCast_residue_eq_zero** (hirr : Irreducible (T : 𝒪[k])) : ((T : ℕ) : ResidueField 𝒪[k]) = 0

theorem **charP_residueField_of_irreducible** (hirr : Irreducible (T : 𝒪[k])) : CharP (ResidueField 𝒪[k]) T

theorem **maximalIdeal_eq** (hirr : Irreducible (T : 𝒪[k])) : maximalIdeal 𝒪[k] = Ideal.span {(T : 𝒪[k])}

theorem **dvd_of_residue** (hirr : Irreducible (T : 𝒪[k])) (a : ℤ) : (T : 𝒪[k]) ∣ (a : 𝒪[k]) ↔ (T : ℤ) ∣ a

theorem **pow_dvd_intCast_iff** (hirr : Irreducible (T : 𝒪[k])) (n : ℕ) (a : ℤ) : (T : 𝒪[k]) ^ n ∣ (a : 𝒪[k]) ↔ (T : ℤ) ^ n ∣ a

theorem **exists_residue_intCast** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) (y : 𝒪[k]) : ∃ b : ℤ, (T : 𝒪[k]) ∣ y - b

theorem **exists_intCast_sub** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) (n : ℕ) (x : 𝒪[k]) : ∃ a : ℤ, (T : 𝒪[k]) ^ n ∣ x - a

theorem **charP_quot** (hirr : Irreducible (T : 𝒪[k])) (n : ℕ) : CharP (𝒪[k] ⧸ Ideal.span {(T : 𝒪[k]) ^ n}) (T ^ n)

theorem **quotEquiv_intCast** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) (n : ℕ) (a : ℤ) : quotEquiv hcard hirr n (a : ZMod (T ^ n)) = Ideal.Quotient.mk _ (a : 𝒪[k])

theorem **red_eq** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) (n : ℕ) (x : 𝒪[k]) (a : ℤ) (h : (T : 𝒪[k]) ^ n ∣ x - a) : red hcard hirr n x = a

theorem **red_compat** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) (k1 k2 : ℕ) (hk : k1 ≤ k2) : (ZMod.castHom (pow_dvd_pow T hk) (ZMod (T ^ k1))).comp (red hcard hirr k2) = red hcard hirr k1

theorem **padic_intCast_mem_span_iff** (m : ℕ) (c : ℤ) : (c : ℤ_[T]) ∈ Ideal.span {(T : ℤ_[T]) ^ m} ↔ (T : ℤ) ^ m ∣ c

theorem **isAdicComplete_integer** : IsAdicComplete (maximalIdeal 𝒪[k]) 𝒪[k]

theorem **toZModPow_toPadic** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) (n : ℕ) (x : 𝒪[k]) : PadicInt.toZModPow n (toPadic hcard hirr x) = red hcard hirr n x

theorem **toPadic_injective** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) : Function.Injective (toPadic hcard hirr)

theorem **toPadic_surjective** (hcard : Nat.card (ResidueField 𝒪[k]) = T) (hirr : Irreducible (T : 𝒪[k])) : Function.Surjective (toPadic hcard hirr)

theorem **valuation_T_lt_one** (hirr : Irreducible (T : 𝒪[k])) : valuation k (T : k) < 1

theorem **valuation_T_ne_zero** (hirr : Irreducible (T : 𝒪[k])) : valuation k (T : k) ≠ 0

theorem **valuation_le_T_pow_iff** (hirr : Irreducible (T : 𝒪[k])) (o : 𝒪[k]) (n : ℕ) : valuation k (o : k) ≤ valuation k (T : k) ^ n ↔ (T : 𝒪[k]) ^ n ∣ o

theorem **exists_T_pow_lt** (hirr : Irreducible (T : 𝒪[k])) (γ : ValueGroupWithZero k) (hγ : γ ≠ 0) : ∃ n : ℕ, valuation k (T : k) ^ n < γ

theorem **exists_ringEquiv_padic** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [IsTopologicalRing k] [CharZero k] (T : ℕ) [Fact T.Prime], Nat.card (IsLocalRing.ResidueField 𝒪[k]) = T → Irreducible (T : 𝒪[k]) → ∃ φ : ℚ_[T] ≃+* k, Continuous φ ∧ Continuous φ.symm ∧ ∀ x : ℚ_[T], φ x ∈ 𝒪[k] ↔ ‖x‖ ≤ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRingEquivPadicOfResidueCardUniformizerStatement

theorem **_root_.GlobalTheory.T.exists_ringEquiv_padic_of_residueCard_uniformizer** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [IsTopologicalRing k] [CharZero k] (T : ℕ) [Fact T.Prime], Nat.card (IsLocalRing.ResidueField 𝒪[k]) = T → Irreducible (T : 𝒪[k]) → ∃ φ : ℚ_[T] ≃+* k, Continuous φ ∧ Continuous φ.symm ∧ ∀ x : ℚ_[T], φ x ∈ 𝒪[k] ↔ ‖x‖ ≤ 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRootExchangeParabolicMulPrincipalCongruenc

theorem **of_blockIx_lt** {a b : n ⊕ Fin c} (h : blockIx p b < blockIx p a) : lev p b ≤ lev p a ∧ ¬ (a.isLeft = true ∧ b.isLeft = true) ∧ a ≠ b

theorem **mem_R_of_pattern** (r : GL (n ⊕ Fin c) k) (hr : ∀ a b : n ⊕ Fin c, lev p b ≤ lev p a → ¬ (a.isLeft = true ∧ b.isLeft = true) → r.val a b = (1 : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) a b) : r ∈ R k n c p

theorem **exists_R_mul_KOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ϖ : kˣ) (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ) (h : GL (n ⊕ Fin c) k), (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas⁻¹ * h * (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas ∈ KO k (n ⊕ Fin c) → (∀ a b : n ⊕ Fin c, lev p b ≤ lev p a → ¬ (a.isLeft = true ∧ b.isLeft = true) → valuation k (((typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas⁻¹ * h * (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas).val a b - (1 : Matrix _ _ k) a b) < 1) → ∃ r ∈ R k n c p, ∃ u : GL (n ⊕ Fin c) k, (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas⁻¹ * u * (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas ∈ KOne k (n ⊕ Fin c) ∧ h = r * u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRootExchangeParabolicMulPrincipalCongruencStatement

theorem **_root_.GlobalTheory.T.exists_rootExchangeParabolic_mul_principalCongruence** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ϖ : kˣ) (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ) (h : GL (n ⊕ Fin c) k), (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas⁻¹ * h * (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas ∈ KO k (n ⊕ Fin c) → (∀ a b : n ⊕ Fin c, lev p b ≤ lev p a → ¬ (a.isLeft = true ∧ b.isLeft = true) → valuation k (((typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas⁻¹ * h * (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas).val a b - (1 : Matrix _ _ k) a b) < 1) → ∃ r ∈ R k n c p, ∃ u : GL (n ⊕ Fin c) k, (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas⁻¹ * u * (typeFrame ϖ (Equiv.refl (n ⊕ Fin c))).bas ∈ KOne k (n ⊕ Fin c) ∧ h = r * u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRowFormPrescribedValuesClassRepresentative

theorem **exists_Mt_of_rep_values** : ∀ {E K ι : Type} [Field E] [NumberField E] [StarRing E] [Field K] [NumberField K] [StarRing K] [Fintype ι] {R : Formalism.BasicNotions.Ranks} {I : GlobalTheory.Mem.Col2.ChainIn E K ι R} (Q : GlobalTheory.Mem.Col2.PerIn I) (emb : ι → E →+* K) (σ : K →+* ℂ), (∀ b, Q.H b = ⊤) → ∀ (w : Q.S'.Cl → Formalism.L2.DIV.Pol ℂ ι (R.N R.m - 1)), (∀ c, w c ∈ Formalism.L2.DIV.VV ℂ Q.sgT) → (∀ c (s : Q.S'.Stab c), w c = (σ (Q.S'.chr c.1 s.1 : K))⁻¹ • Formalism.L2.DIV.rhoP Q.sgT (Formalism.L2.DIV.embGL (Formalism.L2.Col.embAt σ emb) (Q.S'.gam c.1 s.1)) (w c)) → ∃ F : Q.Mt emb σ, ∀ c, F.1 c.1 (Q.S'.rep c) = w c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRowFormPrescribedValuesClassRepresentativeStatement

theorem **_root_.GlobalTheory.T.exists_rowForm_prescribed_values_class_representatives** : ∀ {E K ι : Type} [Field E] [NumberField E] [StarRing E] [Field K] [NumberField K] [StarRing K] [Fintype ι] {R : Formalism.BasicNotions.Ranks} {I : GlobalTheory.Mem.Col2.ChainIn E K ι R} (Q : GlobalTheory.Mem.Col2.PerIn I) (emb : ι → E →+* K) (σ : K →+* ℂ), (∀ b, Q.H b = ⊤) → ∀ (w : Q.S'.Cl → Formalism.L2.DIV.Pol ℂ ι (R.N R.m - 1)), (∀ c, w c ∈ Formalism.L2.DIV.VV ℂ Q.sgT) → (∀ c (s : Q.S'.Stab c), w c = (σ (Q.S'.chr c.1 s.1 : K))⁻¹ • Formalism.L2.DIV.rhoP Q.sgT (Formalism.L2.DIV.embGL (Formalism.L2.Col.embAt σ emb) (Q.S'.gam c.1 s.1)) (w c)) → ∃ F : Q.Mt emb σ, ∀ c, F.1 c.1 (Q.S'.rep c) = w c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRowFormReadingLevelFormLastSpace

theorem **GlobalTheory.AdForm.exists_readsG_of_level** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ : ℕ → ℂ), PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (a : I.ι), ∀ (bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))) (K : ClassesAtPrimes (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsRowFormReadingLevelFormLastSpaceStatement

theorem **_root_.GlobalTheory.T.exists_rowForm_reading_levelForm_lastSpace** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ : ℕ → ℂ), PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (a : I.ι), ∀ (bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))) (K : ClassesAtPrimes (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSBExtFunEqHalfInd

theorem **wr_apply** (α : Frame p S n') (β : Frame p T n) (x : Matrix n n' k) (i : Fin T.rank) (j : Fin S.rank) : wr α β x i j = p.sec ((β.b * x * α.b') i j)

theorem **rd_eq** (α : Frame p S n') (β : Frame p T n) (t : WO S T) : rd α β t = β.b' * (Matrix.map t p.pr : Matrix (Fin T.rank) (Fin S.rank) k) * α.b

theorem **isHalf_wr** (α : Frame p S n') (β : Frame p T n) (x : Matrix n n' k) : IsHalf p.e (wr α β x)

theorem **wr_rd** (α : Frame p S n') (β : Frame p T n) {t : WO S T} (ht : IsHalf p.e t) : wr α β (rd α β t) = t

theorem **open_compact_sec** (p : SplitPt D k) {LS : Lat S} {LT : Lat T} (hS : Submodule.IsLattice D.host.E LS) (hT : Submodule.IsLattice D.host.E LT) : IsOpen {y : Matrix (Fin T.rank) (Fin S.rank) k | y.map p.sec ∈ latO S T (LS : Set _) (LT : Set _) fun _ => True} ∧ IsCompact {y : Matrix (Fin T.rank) (Fin S.rank) k | y.map p.sec ∈ latO S T (LS : Set _) (LT : Set _) fun _ => True}

theorem **exists_idem** (p : SplitPt D k) : ∃ ε, IsPlaceIdem D p.v ε ∧ algebraMap (AF D.host.F) (AE D) ε * p.e = p.e

theorem **compact_open** (α : Frame p S n') (β : Frame p T n) {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) : IsCompact {x | wr α β x ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v) ∧ IsHalf p.e (wr α β x)} ∧ IsOpen {x | wr α β x ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v) ∧ IsHalf p.e (wr α β x)}

theorem **exists_SB_extFun_eq_halfInd** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S T : HSp D) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∃ Ψ : Matrix n n' k → ℂ, Ψ ∈ GlobalTheory.SplitWeil.SB (Matrix n n' k) ∧ extFun α β Ψ = halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSBesUnscaledDual

theorem **exists_sBes_scaled_dual** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (F₀ : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F₀), IdE X → IsStdChar D ψ (-1) → ∃ lam : D.host.E, (lam = (X.B.g : D.host.E) ∨ lam = -(X.B.g : D.host.E)) ∧ ∀ (μ : D.host.E) (vA : Vol (D.sp.X 0)), μ ≠ 0 → ∃ B : SBes X h u ψ (D.sp.X 0), B.Adapted tr.wt ∧ B.vA = vA ∧ (∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = μ * X.ιE (X.zR u (Fin.cast (rank_eq X h) k))) ∧ (∀ k (a : Fin D.R.rS), B.d.e k a = μ * lam ^ (D.R.rS - (a : ℕ)) * X.ιE (X.eR a (Fin.cast (rank_eq X h) k))) ∧ ∀ k (a : Fin D.R.rS), B.d.e' k a = (star (μ * lam ^ (D.R.rS - (a : ℕ))))⁻¹ * X.ιE (X.fR a (Fin.cast (rank_eq X h) k))

theorem **exists_sBes_unscaled_dual** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (F₀ : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F₀), IdE X → IsStdChar D ψ (-1) → ∃ lam : D.host.E, (lam = (X.B.g : D.host.E) ∨ lam = -(X.B.g : D.host.E)) ∧ ∃ B : SBes X h u ψ (D.sp.X 0), B.Adapted tr.wt ∧ (∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = X.ιE (X.zR u (Fin.cast (rank_eq X h) k))) ∧ (∀ k (a : Fin D.R.rS), B.d.e k a = lam ^ (D.R.rS - (a : ℕ)) * X.ιE (X.eR a (Fin.cast (rank_eq X h) k))) ∧ ∀ k (a : Fin D.R.rS), B.d.e' k a = (star (lam ^ (D.R.rS - (a : ℕ))))⁻¹ * X.ιE (X.fR a (Fin.cast (rank_eq X h) k))

theorem **exists_sBes_unscaled** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (F₀ : Formalism.L2.Forms.Forms0 D u) (tr : Formalism.L2.SG.ColDict.Transport X u F₀), IdE X → IsStdChar D ψ (-1) → ∃ lam : D.host.E, (lam = (X.B.g : D.host.E) ∨ lam = -(X.B.g : D.host.E)) ∧ ∃ B : SBes X h u ψ (D.sp.X 0), B.Adapted tr.wt ∧ (∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = X.ιE (X.zR u (Fin.cast (rank_eq X h) k))) ∧ ∀ k (a : Fin D.R.rS), B.d.e k a = lam ^ (D.R.rS - (a : ℕ)) * X.ιE (X.eR a (Fin.cast (rank_eq X h) k))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsScalarPreservesIntegralMatricesSubst

theorem **exists_integral_multiplier** {k : Type} [Field k] [ValuativeRel k] (a : k) : ∃ b : k, b ≠ 0 ∧ b ∈ 𝒪[k] ∧ b * a ∈ 𝒪[k]

theorem **exists_common_integral_multiplier** {k : Type} [Field k] [ValuativeRel k] {ι : Type} [Fintype ι] (f : ι → k) : ∃ c : k, c ≠ 0 ∧ c ∈ 𝒪[k] ∧ ∀ i, c * f i ∈ 𝒪[k]

theorem **enl_scalar_denominator** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (gk : GL Y' k) (hk : GL Y k), ∃ c : kˣ, ∀ zk : GL Y k, (zk : Matrix Y Y k) = (c : k) • (1 : Matrix Y Y k) → Enl gk (hk * zk⁻¹) ∧ Enl (1 : GL Y' k) zk⁻¹

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsScalarPreservesIntegralMatricesSubstStatement

theorem **_root_.GlobalTheory.T.exists_scalar_preservesIntegralMatricesSubst** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (gk : GL Y' k) (hk : GL Y k), ∃ c : kˣ, ∀ zk : GL Y k, (zk : Matrix Y Y k) = (c : k) • (1 : Matrix Y Y k) → Enl gk (hk * zk⁻¹) ∧ Enl (1 : GL Y' k) zk⁻¹

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSchrodingerIntertwinerIntegralMatrices

theorem **exists_intertwiner_span** {G V : Type} [AddCommGroup G] [AddCommGroup V] [Module ℂ V] {χ : G → G → ℂ} {X Y L L' : AddSubgroup G} (h : IsBichar χ) (hL : L ≤ X) (hL' : L' ≤ Y) (hLL' : ∀ u ∈ L, ∀ y ∈ L', χ u y = 1) (hX : ∀ x ∈ X, (∀ y ∈ L', χ x y = 1) → x ∈ L) (hY : ∀ y ∈ Y, (∀ u ∈ L, χ u y = 1) → y ∈ L') (dX : ∀ k : ℕ, 0 < k → ∀ x ∈ X, ∃ x' ∈ X, k • x' = x) (dY : ∀ k : ℕ, 0 < k → ∀ y ∈ Y, ∃ y' ∈ Y, k • y' = y) (tf : ∀ k : ℕ, 0 < k → Function.Injective fun t : G => k • t) (hfin : ∀ k : ℕ, 0 < k → ∃ R : Set G, R.Finite ∧ ∀ t ∈ L, ∃ r ∈ R, ∃ u ∈ L, t = r + k • u) (ρ : G → V →ₗ[ℂ] V) (ρX : ∀ x ∈ X, ∀ x' ∈ X, ∀ f, ρ x (ρ x' f) = ρ (x + x') f) (ρY : ∀ y ∈ Y, ∀ y' ∈ Y, ∀ f, ρ y (ρ y' f) = ρ (y + y') f) (ρXY : ∀ x ∈ X, ∀ y ∈ Y, ∀ f, ρ x (ρ y f) = χ x y • ρ y (ρ x f)) (f₀ : V) (hf : ∀ u ∈ L, ρ u f₀ = f₀) (hf' : ∀ u ∈ L', ρ u f₀ = f₀) : ∃ T : (G → ℂ) →ₗ[ℂ] V, T (ind L) = f₀ ∧ ∀ Φ : G → ℂ, (∀ t

theorem **mem_intLat** {k : Type} [Field k] [ValuativeRel k] {Y Y' : Type} {x : Matrix Y Y' k} : x ∈ intLat k Y Y' ↔ ∀ r q, x r q ∈ 𝒪[k]

theorem **mem_annLat** {k : Type} [Field k] [ValuativeRel k] {Y Y' Z : Type} [AddCommGroup Z] {χ : Matrix Y Y' k → Z → ℂ} {hr : ∀ a z z', χ a (z + z') = χ a z * χ a z'} {h0 : ∀ a, χ a 0 = 1} {z : Z} : z ∈ annLat χ hr h0 ↔ ∀ u : Matrix Y Y' k, (∀ r q, u r q ∈ 𝒪[k]) → χ u z = 1

theorem **inv_nsmul_nsmul** {k : Type} [Field k] [CharZero k] {Y Y' : Type} (n : ℕ) (hn : 0 < n) (m : Matrix Y Y' k) : (n : k)⁻¹ • (n • m) = m

theorem **liftZ_of_eq** {A Z : Type} [AddCommGroup Z] [TopologicalSpace A] (Φ : SB A) {g : A × Z} (h : g.2 = 0) : liftZ Φ g = Φ.1 g.1

theorem **liftZ_of_ne** {A Z : Type} [AddCommGroup Z] [TopologicalSpace A] (Φ : SB A) {g : A × Z} (h : g.2 ≠ 0) : liftZ Φ g = 0

theorem **exists_intertwiner_matrix** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [CharZero k] (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (Z : Type) [AddCommGroup Z] (χ : Matrix Y Y' k → Z → ℂ), (∀ a b z, χ (a + b) z = χ a z * χ b z) → (∀ a z z', χ a (z + z') = χ a z * χ a z') → (∀ z, χ 0 z = 1) → (∀ a, χ a 0 = 1) → (∀ n : ℕ, 0 < n → ∀ z : Z, ∃ z' : Z, n • z' = z) → (∀ n : ℕ, 0 < n → Function.Injective fun z : Z => n • z) → (∀ x : Matrix Y Y' k, (∀ z, (∀ u : Matrix Y Y' k, (∀ r q, u r q ∈ 𝒪[k]) → χ u z = 1) → χ x z = 1) → ∀ r q, x r q ∈ 𝒪[k]) → ∀ (M : Type) [AddCommGroup M] [Module ℂ M] (τ : Matrix Y Y' k → M →ₗ[ℂ] M) (μ : Z → M →ₗ[ℂ] M), (∀ x x' f, τ x (τ x' f) = τ (x + x') f) → (∀ z z' f, μ z (μ z' f) = μ (z + z') f) → (∀ x z f, τ x (μ z f) = χ x z • μ z (τ x f)) → (∀ f, τ 0 f = f) → (∀ f, μ 0 f = f) → ∀ f₀ : M,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSchrodingerIntertwinerIntegralMatricesStatement

theorem **_root_.GlobalTheory.T.exists_schrodinger_intertwiner_integralMatrices** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [CharZero k] (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (Z : Type) [AddCommGroup Z] (χ : Matrix Y Y' k → Z → ℂ), (∀ a b z, χ (a + b) z = χ a z * χ b z) → (∀ a z z', χ a (z + z') = χ a z * χ a z') → (∀ z, χ 0 z = 1) → (∀ a, χ a 0 = 1) → (∀ n : ℕ, 0 < n → ∀ z : Z, ∃ z' : Z, n • z' = z) → (∀ n : ℕ, 0 < n → Function.Injective fun z : Z => n • z) → (∀ x : Matrix Y Y' k, (∀ z, (∀ u : Matrix Y Y' k, (∀ r q, u r q ∈ 𝒪[k]) → χ u z = 1) → χ x z = 1) → ∀ r q, x r q ∈ 𝒪[k]) → ∀ (M : Type) [AddCommGroup M] [Module ℂ M] (τ : Matrix Y Y' k → M →ₗ[ℂ] M) (μ : Z → M →ₗ[ℂ] M), (∀ x x' f, τ x (τ x' f) = τ (x + x') f) → (∀ z z' f, μ z (μ z' f) = μ (z + z') f) → (∀ x z f, τ x (μ z f) = χ x z • μ z (τ x f)) → (∀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSchwartzBruhatExtByZeroOfSplitPlaceHalf

theorem **exists_sb_ext_eq** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.LocMod.Place D k) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (Λ : AddSubgroup (WO S T)), IsOpen ((Λ.comap (GlobalTheory.LocMod.Witness.upH p FS FT) : AddSubgroup (Matrix Y Y' k)) : Set (Matrix Y Y' k)) → IsCompact ((Λ.comap (GlobalTheory.LocMod.Witness.upH p FS FT) : AddSubgroup (Matrix Y Y' k)) : Set (Matrix Y Y' k)) → ∀ Φ : WO S T → ℂ, IsSBHalf p.e Λ Φ → ∃ Φ' : SB (Matrix Y Y' k), GlobalTheory.LocMod.Witness.ext p FS FT Φ' = Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSchwartzBruhatExtByZeroOfSplitPlaceHalfStatement

theorem **_root_.GlobalTheory.T.exists_schwartzBruhat_extByZero_of_splitPlaceHalf** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.LocMod.Place D k) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (Λ : AddSubgroup (WO S T)), IsOpen ((Λ.comap (GlobalTheory.LocMod.Witness.upH p FS FT) : AddSubgroup (Matrix Y Y' k)) : Set (Matrix Y Y' k)) → IsCompact ((Λ.comap (GlobalTheory.LocMod.Witness.upH p FS FT) : AddSubgroup (Matrix Y Y' k)) : Set (Matrix Y Y' k)) → ∀ Φ : WO S T → ℂ, IsSBHalf p.e Λ Φ → ∃ Φ' : SB (Matrix Y Y' k), GlobalTheory.LocMod.Witness.ext p FS FT Φ' = Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSchwartzIntegral

theorem **exists_bound** (L Li : S → V →L[ℝ] V) (hL : Continuous L) (hLi : Continuous Li) (h : ∀ s x, Li s (L s x) = x) (f : SchwartzMap V ℂ) (k n : ℕ) : ∃ M : ℝ, 0 ≤ M ∧ ∀ s x, ‖x‖ ^ k * ‖iteratedFDeriv ℝ n (fun y => f (L s y)) x‖ ≤ M

private theorem **integrable_of_continuous_isFiniteMeasure** {E : Type} [NormedAddCommGroup E] [MeasurableSpace S] [OpensMeasurableSpace S] (ν : Measure S) [IsFiniteMeasure ν] {φ : S → E} (hφ : Continuous φ) : Integrable φ ν

private theorem **hasFDerivAt_integral_of_continuous_fderiv** {E : Type} [FiniteDimensional ℝ V] [NormedAddCommGroup E] [NormedSpace ℝ E] [CompleteSpace E] [MeasurableSpace S] [OpensMeasurableSpace S] (ν : Measure S) [IsFiniteMeasure ν] (G : S → V → E) (hG : ∀ s, Differentiable ℝ (G s)) (hc : Continuous fun p : S × V => G p.1 p.2) (hd : Continuous fun p : S × V => fderiv ℝ (G p.1) p.2) (x₀ : V) : HasFDerivAt (fun x => ∫ s, G s x ∂ν) (∫ s, fderiv ℝ (G s) x₀ ∂ν) x₀

private theorem **contDiff_iteratedFDeriv_schwartz_comp** (f : SchwartzMap V ℂ) (m : ℕ) : ContDiff ℝ ∞ (fun p : (V →L[ℝ] V) × V => iteratedFDeriv ℝ m (fun y => f (p.1 y)) p.2)

theorem **exists_schwartz_integral** [FiniteDimensional ℝ V] [MeasurableSpace S] [OpensMeasurableSpace S] (ν : Measure S) [IsFiniteMeasure ν] (L Li : S → V →L[ℝ] V) (hL : Continuous L) (hLi : Continuous Li) (h : ∀ s x, Li s (L s x) = x) (f : SchwartzMap V ℂ) : ∃ F : SchwartzMap V ℂ, ∀ x, F x = ∫ s, f (L s x) ∂ν

theorem **comp_mem_SB** {a b : Y → Y} (ha : Continuous a) (hb : Continuous b) (h : ∀ y, b (a y) = y) (hg : g ∈ WeilM.SB Y) : (fun y => g (a y)) ∈ WeilM.SB Y

theorem **isLocallyConstant_comp** [CompactSpace S] (ha : Continuous fun p : S × Y => a p.1 p.2) (hb : Continuous fun p : S × Y => b p.1 p.2) (h : ∀ s y, b s (a s y) = y) (hg : g ∈ WeilM.SB Y) : IsLocallyConstant fun s y => g (a s y)

theorem **integral_tensor** {S V Y : Type} [TopologicalSpace S] [CompactSpace S] [NormedAddCommGroup V] [NormedSpace ℝ V] [FiniteDimensional ℝ V] [MeasurableSpace S] [OpensMeasurableSpace S] [TopologicalSpace Y] (ν : Measure S) [IsFiniteMeasure ν] (L Li : S → V →L[ℝ] V) (hL : Continuous L) (hLi : Continuous Li) (h : ∀ s x, Li s (L s x) = x) (a b : S → Y → Y) (ha : Continuous fun p : S × Y => a p.1 p.2) (hb : Continuous fun p : S × Y => b p.1 p.2) (hab : ∀ s y, b s (a s y) = y) (f : SchwartzMap V ℂ) {g : Y → ℂ} (hg : g ∈ WeilM.SB Y) : (∀ x y, Integrable (fun s => f (L s x) * g (a s y)) ν) ∧ ∃ (t : Finset (Y → ℂ)) (F : (Y → ℂ) → SchwartzMap V ℂ), (∀ g' ∈ t, g' ∈ WeilM.SB Y) ∧ ∀ x y, ∫ s, f (L s x) * g (a s y) ∂ν = ∑ g' ∈ t, F g' x * g' y

theorem **lmul_apply** {ι : Type} [Fintype ι] {m : ℕ} (n : ℕ) (A : ι → Matrix (Fin m) (Fin m) ℂ) (X : ι → Fin m → Fin n → ℂ) : lmul n A X = fun τ i j => ∑ c, A τ i c * X τ c j

theorem **sum_mul_cancel** {R : Type} [Ring R] {m n : ℕ} {A B : Matrix (Fin m) (Fin m) R} (h : B * A = 1) (y : Fin m → Fin n → R) : (fun i j => ∑ c, B i c * ∑ d, A c d * y d j) = y

theorem **integral_tensor_matrix** {S R ι : Type} [TopologicalSpace S] [CompactSpace S] [MeasurableSpace S] [OpensMeasurableSpace S] [Ring R] [TopologicalSpace R] [IsTopologicalRing R] [Fintype ι] {m n : ℕ} (ν : Measure S) [IsFiniteMeasure ν] (A Ai : S → ι → Matrix (Fin m) (Fin m) ℂ) (hA : ∀ τ, Continuous fun s => A s τ) (hAi : ∀ τ, Continuous fun s => Ai s τ) (h : ∀ s τ, Ai s τ * A s τ = 1) (B Bi : S → Matrix (Fin m) (Fin m) R) (hB : Continuous B) (hBi : Continuous Bi) (h' : ∀ s, Bi s * B s = 1) (f : SchwartzMap (ι → Fin m → Fin n → ℂ) ℂ) {g : (Fin m → Fin n → R) → ℂ} (hg : g ∈ WeilM.SB (Fin m → Fin n → R)) : (∀ (x : ι → Fin m → Fin n → ℂ) (y : Fin m → Fin n → R), Integrable (fun s => f (fun τ i j => ∑ c, A s τ i c * x τ c j) * g (fun i j => ∑ c, B s i c * y c j)) ν) ∧ ∃ (t : Finset ((Fin m → Fin n → R) → ℂ)) (F : ((Fin m → Fin n → R) → ℂ) → SchwartzMap (ι → Fin m → Fin n → ℂ) ℂ), (∀ g' ∈ t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSchwartzVectorField

theorem **Induced.sum** {ι : Type} (s : Finset ι) (c : ι → ℂ) (A : ι → (GA D V → ℂ) → GA D V → ℂ) (h : ∀ i, Induced θ V n (A i)) : Induced θ V n fun f => ∑ i ∈ s, c i • A i f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticeChain

theorem **exists_selfDual_latY** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j < D.R.m → ∃ L : GlobalTheory.FinPl.Lat (Yc D j), IsSelfDualC (Yc D j) L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticeChainStatement

theorem **_root_.GlobalTheory.T.exists_self_dual_lattice_chain** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX :

ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j < D.R.m → ∃ L : GlobalTheory.FinPl.Lat (Yc D j), IsSelfDualC (Yc D j) L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticeMoves

theorem **movedLat** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j < D.R.m → ∀ g : GA D (Yc D j), ∃ M : GlobalTheory.FinPl.Lat (Yc D j), IsSelfDualC (Yc D j) M ∧ ∀ P, GlobalTheory.FinPl.Moves (Yc D j) g ((pk W).ΛY j) M P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticeMovesStatement

theorem **_root_.GlobalTheory.T.exists_self_dual_lattice_moves** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j < D.R.m → ∀ g : GA D (Yc D j), ∃ M : GlobalTheory.FinPl.Lat (Yc D j), IsSelfDualC (Yc D j) M ∧ ∀ P, GlobalTheory.FinPl.Moves (Yc D j) g ((pk W).ΛY j) M P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticePatchedAtPlaces

theorem **mul_eq_zero_of_ne** {x w : Pl D} (h : w ≠ x) {a b : AE D} (ha : a ∈ idAt D (· = x)) (hb : b ∈ idAt D (· = w)) : a * b = 0

theorem **mulVec_eq_of_ne** {S : HSp D} {x w : Pl D} (h : w ≠ x) {g : GA D S} (hg : g ∈ grpAt S (· = x)) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D (· = w)) : mt S g *ᵥ y = y

theorem **latV_eq_off** {S : HSp D} {x w : Pl D} (h : w ≠ x) {g : GA D S} (hg : g ∈ grpAt S (· = x)) {L M : Lat S} (hm : Moves S g L M (· = w)) : latV S M (· = w) = latV S L (· = w)

theorem **latV_le_of_moves** {S : HSp D} {P : Pl D → Prop} {g : GA D S} {L M M' : Lat S} (hm : Moves S g L M P) (hm' : Moves S g L M' P) : latV S M P ≤ latV S M' P

theorem **patch** : ∀ (D : D7) (S : HSp D) (L L' : Lat S), IsSelfDualC S L → IsSelfDualC S L' → ∀ Z : Finset (Pl D), ∃ M : Lat S, IsSelfDualC S M ∧ (∀ x ∈ Z, latV S M (· = x) = latV S L' (· = x)) ∧ ∀ x, x ∉ Z → latV S M (· = x) = latV S L (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticePatchedAtPlacesStatement

theorem **_root_.GlobalTheory.T.exists_selfDual_lattice_patched_at_places** : ∀ (D : D7) (S : HSp D) (L L' : Lat S), IsSelfDualC S L → IsSelfDualC S L' → ∀ Z : Finset (Pl D), ∃ M : Lat S, IsSelfDualC S M ∧ (∀ x ∈ Z, latV S M (· = x) = latV S L' (· = x)) ∧ ∀ x, x ∉ Z → latV S M (· = x) = latV S L (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticePrescribedSplitPlace

theorem **suppAt_one** (e : AE D) (S : HSp D) : SuppAt e S 1

theorem **isLoc_one** (q : SplitPt D k) {S : HSp D} {Y : Type} [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) : (place q).IsLoc F 1 1

theorem **locGrp** (q : SplitPt D k) (S : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) : (∀ gk : GL Y k, ∃! g, (place q).IsLoc F g gk) ∧ (∀ g, SuppAt (place q).e S g → ∃ gk, (place q).IsLoc F g gk) ∧ ∀ g gk h hk, (place q).IsLoc F g gk → (place q).IsLoc F h hk → (place q).IsLoc F (g * h) (gk * hk)

theorem **mem_std** {Y : Type} (y : Y → k) : y ∈ stdLat k Y ↔ ∀ i, y i ∈ 𝒪[k]

theorem **std_eq_span** (Y : Type) [Fintype Y] [DecidableEq Y] : stdLat k Y = Submodule.span 𝒪[k] (Set.range fun i : Y => (Pi.single i 1 : Y → k))

theorem **isLattice_std** (Y : Type) [Fintype Y] [DecidableEq Y] : Submodule.IsLattice k (stdLat k Y) where

theorem **idAt_mono** {P Q : Pl D → Prop} (hPQ : ∀ w, P w → Q w) : idAt D P ≤ idAt D Q

theorem **latV_le_of_off** {S : HSp D} {L L' : Lat S} (hL : Submodule.IsLattice D.host.E L) (hL' : Submodule.IsLattice D.host.E L') {v₀ v : Pl D} (hv : v ≠ v₀) (h : latV S L (· ≠ v₀) = latV S L' (· ≠ v₀)) : latV S L (· = v) ≤ latV S L' (· = v)

theorem **exists_selfDual_frame** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) (S : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) (L₀ : Lat S), IsSelfDualC S L₀ → ∃ L : Lat S, IsSelfDualC S L ∧ (∀ v, v ≠ q.v → latV S L (· = v) = latV S L₀ (· = v)) ∧ ∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((F.β *ᵥ y) i)) ∈ latV S L (· = q.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticePrescribedSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_selfDualLattice_prescribed_splitPlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) (S : HSp D) (Y : Type) [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) (L₀ : Lat S), IsSelfDualC S L₀ → ∃ L : Lat S, IsSelfDualC S L ∧ (∀ v, v ≠ q.v → latV S L (· = v) = latV S L₀ (· = v)) ∧ ∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((F.β *ᵥ y) i)) ∈ latV S L (· = q.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticeWithPrescribedSplitComponen

theorem **exists_gl_of_lattices** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : Type} [Fintype n] [DecidableEq n] (Λ₁ Λ₂ : Submodule 𝒪[k] (n → k)) (h₁ : Λ₁.IsLattice k) (h₂ : Λ₂.IsLattice k) : ∃ g : GL n k, ∀ z, z ∈ Λ₁ ↔ g.val *ᵥ z ∈ Λ₂

theorem **e_mem_idAt** (p : SplitPt D k) : p.e ∈ idAt D (· = p.v)

theorem **sec_eq_mul** (p : SplitPt D k) (a : k) : p.sec a = p.e * p.sec a

theorem **sec_mem_idAt** (p : SplitPt D k) (a : k) : p.sec a ∈ idAt D (· = p.v)

theorem **mt_gl_vec** {p : SplitPt D k} (β : Frame p S n) (g : GL n k) (z : n → k) : mt S (β.gl g) *ᵥ (fun i => p.sec ((β.b *ᵥ z) i)) = fun i => p.sec ((β.b *ᵥ (g.val *ᵥ z)) i)

theorem **mt_gl_off** {p : SplitPt D k} (β : Frame p S n) (g : GL n k) (y : Fin S.rank → AE D) (hy : ∀ i, y i ∈ idAt D (· ≠ p.v)) : mt S (β.gl g) *ᵥ y = y

theorem **main** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (β : Frame p S n) (L : Lat S) (Λ : Submodule 𝒪[k] (n → k)), IsSelfDualC S L → Λ.IsLattice k → ∃ M : Lat S, IsSelfDualC S M ∧ β.IsLat M Λ ∧ latV S M (· ≠ p.v) = latV S L (· ≠ p.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLatticeWithPrescribedSplitComponenStatement

theorem **_root_.GlobalTheory.T.exists_selfDual_lattice_with_prescribed_splitComponent** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (β : Frame p S n) (L : Lat S) (Λ : Submodule 𝒪[k] (n → k)), IsSelfDualC S L → Λ.IsLattice k → ∃ M : Lat S, IsSelfDualC S M ∧ β.IsLat M Λ ∧ latV S M (· ≠ p.v) = latV S L (· ≠ p.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLineLatticeSplittingSum

theorem **mem_latV_iff_wParts** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.FinPl.SplitPt D k) {S : HSp D} {L : GlobalTheory.FinPl.Lat S} (hL : IsSelfDualC S L) (y : Fin S.rank → AE D) (hy : ∀ i, y i ∈ GlobalTheory.FinPl.idAt D (· = p.v)) : y ∈ GlobalTheory.FinPl.latV S L (· = p.v) ↔ (fun i => p.sec (p.pr (y i))) ∈ GlobalTheory.FinPl.latV S L (· = p.v) ∧ ∀ x : Fin S.rank → k, (fun i => p.sec (x i)) ∈ GlobalTheory.FinPl.latV S L (· = p.v) → ∑ i, ∑ j, x i * p.pr (cA D (incA D (S.gram i j))) * p.pr (cA D (y j)) ∈ 𝒪[k]

theorem **latV_eq_of_wPart** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S : HSp D) (L L' : GlobalTheory.FinPl.Lat S), IsSelfDualC S L → IsSelfDualC S L' → (∀ x : Fin S.rank → k, (fun i => q.sec (x i)) ∈ GlobalTheory.FinPl.latV S L (· = q.v) ↔ (fun i => q.sec (x i)) ∈ GlobalTheory.FinPl.latV S L' (· = q.v)) → GlobalTheory.FinPl.latV S L (· = q.v) = GlobalTheory.FinPl.latV S L' (· = q.v)

theorem **rat_mulVec_sec** (q : GlobalTheory.FinPl.SplitPt D k) {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) (b : Fin n → k) : ((rat M : Mx D m n) *ᵥ fun i => q.sec (b i)) = fun i => q.sec ((M.map ((ofSplit q).π.comp (incA D)) *ᵥ b) i)

theorem **exists_lineLat** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S A B : HSp D) (o : OrthDec S A B) (MS : GlobalTheory.FinPl.Lat S) (MA : GlobalTheory.FinPl.Lat A) (Y : Type) [Fintype Y] [DecidableEq Y] (FA : Frame k A.rank Y) (FS : Frame k S.rank (Y ⊕ Unit)) (z : Fin B.rank → k), IsSelfDualC S MS → IsSelfDualC A MA → (∃ LB : GlobalTheory.FinPl.Lat B, IsSelfDualC B LB) → (∀ r a, FS.β r (Sum.inl a) = (o.iA.map (ofSplit q).pr * FA.β) r a) → (∀ r, FS.β r (Sum.inr ()) = (o.iB.map (ofSplit q).pr *ᵥ z) r) → IntIn q S FS MS → IntIn q A FA MA → ∃ LB : GlobalTheory.FinPl.Lat B, IsSelfDualC B LB ∧ GlobalTheory.FinPl.OrthAt o MS MA LB q.v ∧ ∀ y : k, (y ∈ 𝒪[k] ↔ (fun i => q.sec (z i * y)) ∈ GlobalTheory.FinPl.latV B LB (· = q.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLineLatticeSplittingSumStatement

theorem **_root_.GlobalTheory.T.exists_selfDualLineLattice_splitting_sum** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S A B : HSp D) (o : OrthDec S A B) (MS : GlobalTheory.FinPl.Lat S) (MA : GlobalTheory.FinPl.Lat A) (Y : Type) [Fintype Y] [DecidableEq Y] (FA : Frame k A.rank Y) (FS : Frame k S.rank (Y ⊕ Unit)) (z : Fin B.rank → k), IsSelfDualC S MS → IsSelfDualC A MA → (∃ LB : GlobalTheory.FinPl.Lat B, IsSelfDualC B LB) → (∀ r a, FS.β r (Sum.inl a) = (o.iA.map (ofSplit q).pr * FA.β) r a) → (∀ r, FS.β r (Sum.inr ()) = (o.iB.map (ofSplit q).pr *ᵥ z) r) → IntIn q S FS MS → IntIn q A FA MA → ∃ LB : GlobalTheory.FinPl.Lat B, IsSelfDualC B LB ∧ GlobalTheory.FinPl.OrthAt o MS MA LB q.v ∧ ∀ y : k, (y ∈ 𝒪[k] ↔ (fun i => q.sec (z i * y)) ∈ GlobalTheory.FinPl.latV B LB (· = q.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLineLatticeTraceAtPlace

theorem **mul_mem_intAt** {v : GlobalTheory.FinPl.Pl D} {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a * b ∈ intAt D (· = v)

theorem **one_mem_intAf** : (1 : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F) ∈ intAf D.host.F

theorem **incA_mem_intAE** (x : 𝒪 D.host.E) : finPart D (incA D (x : D.host.E)) ∈ intAE D

theorem **mul_incA_mem** {v : GlobalTheory.FinPl.Pl D} {a : AE D} (ha : a ∈ intAt D (· = v)) (x : 𝒪 D.host.E) : a * incA D (x : D.host.E) ∈ intAt D (· = v)

theorem **cA_mem_intAt** {v : GlobalTheory.FinPl.Pl D} {a : AE D} (ha : a ∈ intAt D (· = v)) : cA D a ∈ intAt D (· = v)

theorem **unit_at** {v : GlobalTheory.FinPl.Pl D} {T : ℕ} (hT : (T : 𝒪 D.host.F) ∉ v.asIdeal) : GlobalTheory.FinPlLoc.idem D (· = v) * incA D ((T : D.host.E)⁻¹) ∈ intAt D (· = v)

theorem **exists_isSDAt_line** : ∀ (D : D7) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M' : Lat S) (μ : D.host.E) (T : ℕ) (v : GlobalTheory.FinPl.Pl D), IsLineVec L ζ → μ ≠ 0 → LineAt (vecB o ζ) μ M' v → star ζ ·ᵥ L.gram *ᵥ ζ = -(μ * star μ * (T : D.host.E)) → (T : 𝒪 D.host.F) ∉ v.asIdeal → ∃ Λ' : AddSubgroup (Fin L.rank → AE D), GlobalTheory.IwaPlace.IsSDAt L v Λ' ∧ ∀ l : Fin L.rank → AE D, (∀ i, l i ∈ idAt D (· = v)) → (l ∈ Λ' ↔ o.jB *ᵥ l ∈ latV S M' (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualLineLatticeTraceAtPlaceStatement

theorem **_root_.GlobalTheory.T.exists_selfDual_lineLatticeTrace_at_place** : ∀ (D : D7) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M' : Lat S) (μ : D.host.E) (T : ℕ) (v : GlobalTheory.FinPl.Pl D), IsLineVec L ζ → μ ≠ 0 → LineAt (vecB o ζ) μ M' v → star ζ ·ᵥ L.gram *ᵥ ζ = -(μ * star μ * (T : D.host.E)) → (T : 𝒪 D.host.F) ∉ v.asIdeal → ∃ Λ' : AddSubgroup (Fin L.rank → AE D), GlobalTheory.IwaPlace.IsSDAt L v Λ' ∧ ∀ l : Fin L.rank → AE D, (∀ i, l i ∈ idAt D (· = v)) → (l ∈ Λ' ↔ o.jB *ᵥ l ∈ latV S M' (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualOrthAt

theorem **rat_mulVec_mem_latV** {S T : HSp D} (m : Matrix (Fin T.rank) (Fin S.rank) D.host.E) {L : Lat S} {L' : Lat T} (h : ∀ x ∈ L, m *ᵥ x ∈ L') {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (rat m : Mx D T.rank S.rank) *ᵥ y ∈ latV T L' P

theorem **dsum_eq'** {S : HSp D} (x y : Fin S.rank → D.host.E) : ∑ i, ∑ j, star (x i) * S.gram i j * y j = star x ·ᵥ S.gram *ᵥ y

theorem **pair_iA'** {S A B : HSp D} (o : OrthDec S A B) (a : Fin A.rank → D.host.E) (y : Fin S.rank → D.host.E) : star (o.iA *ᵥ a) ·ᵥ S.gram *ᵥ y = star a ·ᵥ A.gram *ᵥ (o.qA *ᵥ y)

theorem **exists_selfDual_orthAt** : ∀ (D : D7) (S A B : HSp D) (o : OrthDec S A B) (LA : Lat A) (LB : Lat B), IsSelfDualC A LA → IsSelfDualC B LB → ∃ L : Lat S, IsSelfDualC S L ∧ ∀ v, OrthAt o L LA LB v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualStabilizerWithPrescribedSplitPlace

theorem **eq_zero_at_off** (P : Pl D → Prop) {a : AE D} (h1 : a ∈ idAt D P) (h2 : a ∈ idOff D P) : a = 0

theorem **mulVec_off** {S : HSp D} {P : Pl D → Prop} {b : GA D S} (hb : b ∈ grpOff S P) {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D P) : mt S b *ᵥ y = y

theorem **moves_off** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {b : GA D S} (hb : b ∈ grpOff S P) : Moves S b L L P

theorem **moves_one** {S : HSp D} {L : Lat S} {P : Pl D → Prop} : Moves S 1 L L P

theorem **moves_list_prod** {S : HSp D} {L : Lat S} {P : Pl D → Prop} : ∀ l : List (GA D S), (∀ a ∈ l, Moves S a L L P) → Moves S l.prod L L P | [], _ => by rw [List.prod_nil]; exact moves_one | a :: l, h => by rw [List.prod_cons] exact PlacesAdeles.moves_mul (h a List.mem_cons_self) (moves_list_prod l fun b hb => h b (List.mem_cons_of_mem _ hb)) theorem prod_ofFn_single {M : Type*} [Monoid M] : ∀ {r : ℕ} (f : Fin r → M) (i : Fin r), (∀ j, j ≠ i → f j = 1) → (List.ofFn f).prod = f i | 0, _, i, _ => i.elim0 | r + 1, f, i, h => by rw [List.ofFn_succ, List.prod_cons] by_cases hi : i = 0 · subst hi rw [List.prod_eq_one, mul_one] intro x hx obtain ⟨j, rfl⟩

theorem **pr_off** (q : SplitPt D k) {a : AE D} (ha : a ∈ idOff D (· = q.v)) : q.pr a = 0

theorem **map_off** (q : SplitPt D k) {S : HSp D} {b : GA D S} (hb : b ∈ grpOff S (· = q.v)) : (mt S b).map q.pr = 1

theorem **one_place** (S : HSp D) (N : Lat S) (hN : IsSelfDualC S N) (q : SplitPt D k) {Y : Type} [Fintype Y] [DecidableEq Y] (F : GlobalTheory.LocMod.Frame k S.rank Y) (hI : IntIn q S F N) (x : GL Y k) (hx : x ∈ GlobalTheory.SplitWeil.KO k Y) : ∃ g, InK S N (· = q.v) g ∧ (mt S g).map q.pr = F.β ∗ x.val ∗ F.γ

theorem **exists_of_local** : ∀ {D : D7} (S : HSp D) (N : Lat S), IsSelfDualC S N → ∀ {r : ℕ} (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : (i : Fin r) → SplitPt D (k i)) (Y : Fin r → Type) [∀ i, Fintype (Y i)] [∀ i, DecidableEq (Y i)] (F : (i : Fin r) → GlobalTheory.LocMod.Frame (k i) S.rank (Y i)), Function.Injective (fun i => (q i).v) → (∀ i, IntIn (q i) S (F i) N) → ∀ x : (i : Fin r) → GL (Y i) (k i), (∀ i, x i ∈ GlobalTheory.SplitWeil.KO (k i) (Y i)) → ∃ g, InK S N (fun p => ∃ i, (q i).v = p) g ∧ ∀ i, (mt S g).map (q i).pr = (F i).β ∗ (x i).val ∗ (F i).γ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSelfDualStabilizerWithPrescribedSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_selfDual_stabilizer_with_prescribed_splitPlace_matrices** : ∀ {D : D7} (S : HSp D) (N : Lat S), IsSelfDualC S N → ∀ {r : ℕ} (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : (i : Fin r) → SplitPt D (k i)) (Y : Fin r → Type) [∀ i, Fintype (Y i)] [∀ i, DecidableEq (Y i)] (F : (i : Fin r) → GlobalTheory.LocMod.Frame (k i) S.rank (Y i)), Function.Injective (fun i => (q i).v) → (∀ i, IntIn (q i) S (F i) N) → ∀ x : (i : Fin r) → GL (Y i) (k i), (∀ i, x i ∈ GlobalTheory.SplitWeil.KO (k i) (Y i)) → ∃ g, InK S N (fun p => ∃ i, (q i).v = p) g ∧ ∀ i, (mt S g).map (q i).pr = (F i).β ∗ (x i).val ∗ (F i).γ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSiegelParabolicConjDoubledCompact

theorem **compInf_sigR_comm** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (σ : E →ₐ[F] E) (w : NumberField.InfinitePlace F) (x : AF F ⊗[F] E) : compInf F E w (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF F) σ x) = Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR w.Completion σ (compInf F E w x)

theorem **compInf_inclR_eq** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (w : NumberField.InfinitePlace F) (y : E) : compInf F E w (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.inclR (AF F) y) = Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.inclR w.Completion y

theorem **sq_real_of_majorant** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (a b : GL (Fin n) (AF F ⊗[F] E)) (ha : IsMajorantFrame σ J a) (hbw : ∀ w, Formalism.UG.bc (compInf F E w) b = Formalism.UG.bc (compInf F E w) a) (w : NumberField.InfinitePlace F) : (((b.1.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF F) σ))ᵀ ∗ J.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.inclR (AF F)) ∗ b.1) ∗ ((b.1.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF F) σ))ᵀ ∗ J.map (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.inclR (AF F)) ∗ b.1)).map (compInf F E w) = 1

theorem **gram_blk_mul** {R : Type} [CommRing R] (c : R →+* R) {n : ℕ} (b J : Matrix (Fin n) (Fin n) R) (M : Matrix (Fin (n + n)) (Fin (n + n)) R) : Formalism.UG.cT c (blkMat finSumFinEquiv b b ∗ M) ∗ dblGram finSumFinEquiv J ∗ (blkMat finSumFinEquiv b b ∗ M) = Formalism.UG.cT c M ∗ dblGram finSumFinEquiv (Formalism.UG.cT c b ∗ J ∗ b) ∗ M

theorem **diag_blk_mul** {R : Type} [CommRing R] {n : ℕ} (b : Matrix (Fin n) (Fin n) R) (M : Matrix (Fin (n + n)) (Fin (n + n)) R) (l : Fin (n + n)) (h : ∀ i : Fin n, M (finSumFinEquiv (Sum.inr i : Fin n ⊕ Fin n)) l = M (finSumFinEquiv (Sum.inl i : Fin n ⊕ Fin n)) l) (i : Fin n) : (blkMat finSumFinEquiv b b ∗ M) (finSumFinEquiv (Sum.inr i : Fin n ⊕ Fin n)) l = (blkMat finSumFinEquiv b b ∗ M) (finSumFinEquiv (Sum.inl i : Fin n ⊕ Fin n)) l

theorem **gA_dblSp** {D : D7} (A : HSp D) : gA (dblSp A) = dblGram finSumFinEquiv (gA A)

theorem **gram_frame** {D : D7} (A : HSp D) (b : GL (Fin A.rank) (AE D)) (M : GL (Fin (A.rank + A.rank)) (AE D)) : adj (blkGL finSumFinEquiv (b, b) ∗ M).1 ∗ gA (dblSp A) ∗ (blkGL finSumFinEquiv (b, b) ∗ M).1 = adj M.1 ∗ dblGram finSumFinEquiv (adj b.1 ∗ gA A ∗ b.1) ∗ M.1

theorem **exists_siegel_frame** {D : D7} (A : HSp D) (μ : D.host.E) (hμ : IsSkew μ) (b : GL (Fin A.rank) (AE D)) (M : GL (Fin (A.rank + A.rank)) (AE D)) (hMg : adj M.1 ∗ dblGram finSumFinEquiv (adj b.1 ∗ gA A ∗ b.1) ∗ M.1 = incA D μ • rat (Jst D A.rank)) (hMc : ∀ i j : Fin A.rank, M.1 (finSumFinEquiv (Sum.inr i : Fin A.rank ⊕ Fin A.rank)) (finSumFinEquiv (Sum.inl j : Fin A.rank ⊕ Fin A.rank)) = M.1 (finSumFinEquiv (Sum.inl i : Fin A.rank ⊕ Fin A.rank)) (finSumFinEquiv (Sum.inl j : Fin A.rank ⊕ Fin A.rank))) : ∃ p : siegel A, mt (dblSp A) p ∗ (rat (Tm A μ) : Mx D (A.rank + A.rank) (A.rank + A.rank)) = (blkGL finSumFinEquiv (b, b) ∗ M).1

theorem **adj_gram** {D : D7} (A : HSp D) (b : Mx D A.rank A.rank) : adj (adj b ∗ gA A ∗ b) = adj b ∗ gA A ∗ b

theorem **sq_real** {D : D7} (A : HSp D) (a b : GL (Fin A.rank) (AE D)) (ha : IsMaj A a) (hbw : ∀ w, Formalism.UG.bc (compInf D.host.F D.host.E w) b = Formalism.UG.bc (compInf D.host.F D.host.E w) a) (w : NumberField.InfinitePlace D.host.F) : ((adj b.1 ∗ gA A ∗ b.1) ∗ (adj b.1 ∗ gA A ∗ b.1)).map (compInf D.host.F D.host.E w) = 1

theorem **exists_isConj** : ∀ D : D7, ∃ μ : D.host.E, IsSkew μ ∧ ∀ (A : HSp D) (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∃ p : siegel A, IsConj A a κ p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSiegelParabolicConjDoubledCompactStatement

theorem **_root_.GlobalTheory.T.exists_siegel_parabolic_conj_doubled_compact** : ∀ D : D7, ∃ μ : D.host.E, IsSkew μ ∧ ∀ (A : HSp D) (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∃ p : siegel A, IsConj A a κ p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSingle

theorem **edgeVal_one** {k : Type} [CommRing k] {n : ℕ} (ρ κ : ℕ) (a : Fin n → Fin n → k) : edgeVal ρ κ a 1 = 0

theorem **ext'** {a b : A} (h : ∀ Q, Z.p Q a = Z.p Q b) : a = b

theorem **e_mem** (Q : P) : Z.e Q ∈ S

theorem **e_mul_eq_iff** (Q : P) (a b : A) : Z.e Q ∗ a = Z.e Q ∗ b ↔ Z.p Q a = Z.p Q b

theorem **e_idem** (Q : P) : Z.e Q ∗ Z.e Q = Z.e Q

theorem **e_mul_mem_iff** (Q : P) (a : A) : Z.e Q ∗ a ∈ S ↔ Z.p Q a ∈ Z.O Q

theorem **e_injective** : Function.Injective Z.e

theorem **exists_single** (Q : P) (x : Z.K Q) : ∃ a : A, Z.p Q a = x ∧ ∀ Q', Q' ≠ Q → Z.p Q' a = 0

theorem **pO_coe** (Q : P) (s : S) : (Z.pO Q s : Z.K Q) = Z.p Q s

theorem **pO_surj** (Q : P) : Function.Surjective (Z.pO Q)

theorem **exists_res** (Q : P) : ∃ ρ : Z.O Q →+∗ k Q, Function.Surjective ρ ∧ ∀ s, ρ (Z.pO Q s) = r Q s

theorem **r_congr** (Q : P) {s t : S} (h : Z.p Q s = Z.p Q t) : r Q s = r Q t

theorem **gl_ext** {n : ℕ} {x y : GL (Fin n) A} (h : ∀ Q, bc (Z.p Q) x = bc (Z.p Q) y) : x = y

theorem **gl_int** {n : ℕ} {x : GL (Fin n) A} (h : ∀ Q, ∃ y : GL (Fin n) (Z.O Q), bc (Z.O Q).subtype y = bc (Z.p Q) x) : ∃ y : GL (Fin n) S, bc S.subtype y = x

theorem **gl_single** {n : ℕ} (Q : P) (y : GL (Fin n) (Z.K Q)) : ∃ x : GL (Fin n) A, bc (Z.p Q) x = y ∧ ∀ Q', Q' ≠ Q → bc (Z.p Q') x = 1

theorem **exists_bar** (hσ : ∀ a, σ (σ a) = a) (Q : P) : ∃ Q', σ (Z.e Q) = Z.e Q'

theorem **exists_conj** (hS : ∀ a ∈ S, σ a ∈ S) {Q Q' : P} (h : σ (Z.e Q) = Z.e Q') : ∃ s : Z.K Q →+∗ Z.K Q', (∀ a, s (Z.p Q a) = Z.p Q' (σ a)) ∧ ∀ x ∈ Z.O Q, s x ∈ Z.O Q'

theorem **r_conj** (hσ : ∀ a, σ (σ a) = a) (hS : ∀ a ∈ S, σ a ∈ S) {Q Q' : P} (h : σ (Z.e Q) = Z.e Q') (s : S) (hs : r Q s = 0) : r Q' (σ s, hS _ s.2) = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSingleMemLatLoc

theorem **exists_int_mul_integral** {E : Type} [Field E] [NumberField E] (e : E) : ∃ M : ℤ, M ≠ 0 ∧ ∃ r : 𝒪 E, (r : E) = (M : E) ∗ e

theorem **exists_int_smul_mem** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [hL : Submodule.IsLattice E L] (x : Fin n → E) : ∃ N : ℤ, N ≠ 0 ∧ N • x ∈ L

theorem **exists_single_mem_latLoc** : ∀ {F E : Type} [inst : Field F] [inst_1 : NumberField F] [inst_2 : Field E] [NumberField E] [inst_4 : Algebra F E] {n : ℕ} (L : Submodule (NumberField.RingOfIntegers E) (Fin n → E)) [Submodule.IsLattice E L] (v : IsDedekindDomain.HeightOneSpectrum (NumberField.RingOfIntegers F)), ∃ (c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v), c ≠ (0 : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv F v) ∧ ∀ (i : Fin n), Pi.single i (c ⊗ₜ[F] (1 : E)) ∈ GlobalTheory.AdelicU.latLoc F L v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSingleMemLocalLatticeCompletion

theorem **exists_zsmul_mem_of_isLattice** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (x : Fin n → E) : ∃ m : ℤ, m ≠ 0 ∧ m • x ∈ L

theorem **exists_single_mem_latLoc** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (v : HeightOneSpectrum (𝒪 F)) : ∃ c : Fv F v, c ≠ 0 ∧ ∀ i : Fin n, (Pi.single i (c ⊗ₜ[F] (1 : E)) : Fin n → Fv F v ⊗[F] E) ∈ latLoc F L v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSingleMemLocalLatticeCompletionStatement

theorem **_root_.GlobalTheory.T.exists_single_mem_localLatticeCompletion** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (v : HeightOneSpectrum (𝒪 F)), ∃ c : Fv F v, c ≠ 0 ∧ ∀ i : Fin n, (Pi.single i (c ⊗ₜ[F] (1 : E)) : Fin n → Fv F v ⊗[F] E) ∈ latLoc F L v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSmallCompactLevelSymplecticCharacterTrivia

theorem **exists_open_subgroup** (F : Type) [Field F] [NumberField F] (V : Set (Af F)) (hV : V ∈ 𝒩 (0 : Af F)) : ∃ S : AddSubgroup (Af F), IsOpen (S : Set (Af F)) ∧ (S : Set (Af F)) ⊆ V

theorem **infPart_smul_eq_zero** {D : D7} (y : Af D.host.F) (x : AE D) : infPart D (pairA D.host.F 0 y • x) = 0

theorem **exists_closed_level** (D : D7) (m n : ℕ) (N : Set (Mx D m n)) (hN : N ∈ 𝒩 (0 : Mx D m n)) : ∃ K : AddSubgroup (Mx D m n), IsClosed (K : Set (Mx D m n)) ∧ (K : Set (Mx D m n)) ⊆ N ∧ (∀ u ∈ K, u.map (infPart D) = 0) ∧ ∃ U : AddSubgroup (Mx D m n), IsOpen (U : Set (Mx D m n)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → u ∈ K

theorem **exists_small_level** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T : HSp D), ∀ N ∈ 𝒩 (0 : WO A T), ∃ K : AddSubgroup (WO A T), IsCompact (K : Set (WO A T)) ∧ (K : Set (WO A T)) ⊆ N ∧ (∀ u ∈ K, u.map (infPart D) = 0) ∧ (∃ U : AddSubgroup (WO A T), IsOpen (U : Set (WO A T)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → u ∈ K) ∧ ∀ u ∈ K, ∀ u' ∈ K, ψ (symplOH A T u u') = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSmallCompactLevelSymplecticCharacterTriviaStatement

theorem **_root_.GlobalTheory.T.exists_small_compact_level_symplecticCharacter_trivial** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T : HSp D), ∀ N ∈ 𝒩 (0 : WO A T), ∃ K : AddSubgroup (WO A T), IsCompact (K : Set (WO A T)) ∧ (K : Set (WO A T)) ⊆ N ∧ (∀ u ∈ K, u.map (infPart D) = 0) ∧ (∃ U : AddSubgroup (WO A T), IsOpen (U : Set (WO A T)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → u ∈ K) ∧ ∀ u ∈ K, ∀ u' ∈ K, ψ (symplOH A T u u') = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSmallPrincipalCongruenceLiftVector

theorem **exists_isVacO** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (hS : ∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) (hT : ∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (dS : IsSelfDualC S ΛS) (dT : IsSelfDualC T ΛT) : ∃ f, IsVacO ψ θ S T sS sT ΛS ΛT f

theorem **exists_small_stepVec** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (χA : (AE D)ˣ →∗ Circle) (om : WRep A H) (sA : RealMat A) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → A.rank ≤ H.rank → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∃ (βH : GL (Fin H.rank) (Formalism.IndefR.Lev.EAf D)) (f₀ f : WO A H → ℂ), Formalism.L2.DIV.IsFrame βH (frameLat H) ∧ IsStepVecOf ψ θ A H sA sH (frameLat A) (frameLat H) β βH (prinDat A Z) (fun _ => 1) ∅ (fun _ => 1) f₀ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSmallPrincipalCongruenceLiftVectorStatement

theorem **_root_.GlobalTheory.T.exists_small_principal_congruence_lift_vector** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sA : RealMat A) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → A.rank ≤ H.rank → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∃ (βH : GL (Fin H.rank) (Formalism.IndefR.Lev.EAf D)) (f₀ f : WO A H → ℂ), Formalism.L2.DIV.IsFrame βH (frameLat H) ∧ IsStepVecOf ψ θ A H sA sH (frameLat A) (frameLat H) β βH (prinDat A Z) (fun _ => 1) ⌀ (fun _ => 1) f₀ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSmoothKFiniteUnitHighestWeightVector

theorem **exists_smooth_highestUnit** : GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (τ : D.what₂.Tau), ∃ w : M.H τ, w ∈ M.sm τ ∧ w ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ ∧ IsHighestUnit (M.ε τ) (M.lam τ) (M.σ τ) w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSmoothKFiniteUnitHighestWeightVectorStatement

theorem **_root_.GlobalTheory.T.exists_smooth_kFinite_unit_highestWeight_vector** : GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (τ : D.what₂.Tau), ∃ w : M.H τ, w ∈ M.sm τ ∧ w ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ ∧ IsHighestUnit (M.ε τ) (M.lam τ) (M.σ τ) w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSobolevBoundThetaPointwise

theorem **sup_le_sobO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ S T : HSp D, ∀ (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsCovolOne (ratL T S) μ ℱ → ∀ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) → ∃ (d : ℕ) (C : ℝ≥0), ∀ f ∈ thetaLevel ψ S T U, ∀ w, ‖f w‖ₑ ≤ C * sobO ψ θ μ ℱ d f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSobolevBoundThetaPointwiseStatement

theorem **_root_.GlobalTheory.T.exists_sobolev_bound_theta_pointwise** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ S T : HSp D, ∀ (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsCovolOne (ratL T S) μ ℱ → ∀ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) → ∃ (d : ℕ) (C : ℝ≥0), ∀ f ∈ thetaLevel ψ S T U, ∀ w, ‖f w‖ₑ ≤ C * sobO ψ θ μ ℱ d f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitLatMonoId

theorem **sum_pair_eq** {n : ℕ} (G : Matrix (Fin n) (Fin n) D.host.E) (x y : Fin n → D.host.E) : ∑ i, ∑ j, star (x i) * G i j * y j = star x ⬝ᵥ (G *ᵥ y)

theorem **pair_mul** {m n : ℕ} (A B : Matrix (Fin m) (Fin n) D.host.E) (x y : Fin n → D.host.E) : star x ⬝ᵥ ((Aᴴ * B) *ᵥ y) = star (A *ᵥ x) ⬝ᵥ (B *ᵥ y)

theorem **pair_decomp** (x y : Fin S.rank → D.host.E) : star x ⬝ᵥ (S.gram *ᵥ y) = star (d.qY *ᵥ x) ⬝ᵥ (d.qX *ᵥ y) + star (d.q0 *ᵥ x) ⬝ᵥ (S₀.gram *ᵥ (d.q0 *ᵥ y)) + star (d.qX *ᵥ x) ⬝ᵥ (d.qY *ᵥ y)

theorem **star_single** (i : Fin ρ) (b : D.host.E) : star (Pi.single i b : Fin ρ → D.host.E) = Pi.single i (star b)

theorem **pair_e** (i : Fin ρ) (b : D.host.E) (y : Fin S.rank → D.host.E) : star (d.e *ᵥ Pi.single i b) ⬝ᵥ (S.gram *ᵥ y) = star b * (d.qY *ᵥ y) i

theorem **pair_e'** (i : Fin ρ) (c : D.host.E) (y : Fin S.rank → D.host.E) : star (d.e' *ᵥ Pi.single i c) ⬝ᵥ (S.gram *ᵥ y) = star c * (d.qX *ᵥ y) i

theorem **pair_i₀** (x₀ : Fin S₀.rank → D.host.E) (y : Fin S.rank → D.host.E) : star (d.i₀ *ᵥ x₀) ⬝ᵥ (S.gram *ᵥ y) = star x₀ ⬝ᵥ (S₀.gram *ᵥ (d.q0 *ᵥ y))

theorem **decomp** (y : Fin S.rank → D.host.E) : y = d.e *ᵥ (d.qX *ᵥ y) + d.i₀ *ᵥ (d.q0 *ᵥ y) + d.e' *ᵥ (d.qY *ᵥ y)

theorem **exists_ne_zero** {𝔟 : FI D} (h : 𝔟 ≠ 0) : ∃ b ∈ 𝔟, b ≠ 0

theorem **exists_ne_zero_dualC** (𝔟 : FI D) : ∃ c ∈ dualC 𝔟, c ≠ 0

theorem **dualC_le_span** {𝔟 : FI D} (h : 𝔟 ≠ 0) : ∃ γ : D.host.E, dualC 𝔟 ≤ Submodule.span (𝒪 D.host.E) {γ}

theorem **mulVec_single_smul** {n : ℕ} (M : Matrix (Fin S.rank) (Fin n) D.host.E) (i : Fin n) {a b : D.host.E} (hb : b ≠ 0) : M *ᵥ Pi.single i a = (a * b⁻¹) • (M *ᵥ Pi.single i b)

theorem **isLattice_splitL** (h𝔟 : ∀ i, 𝔟 i ≠ 0) (hL₀ : Submodule.IsLattice D.host.E L₀) : Submodule.IsLattice D.host.E (splitL d L₀ 𝔟)

theorem **isSelfDualC_splitL** (h𝔟 : ∀ i, 𝔟 i ≠ 0) (hL₀ : IsSelfDualC S₀ L₀) : IsSelfDualC S (splitL d L₀ 𝔟)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitLocalSupportedScalarComponent

theorem **place_ι_mul** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (x y : k) : p.ι (x * y) = p.ι x * p.ι y

theorem **place_ι_one** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) : p.ι 1 = p.e

theorem **place_π_e** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) : p.π p.e = 1

theorem **exists_isLoc_scalar** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (V : HSp D) (Z : Type) [Fintype Z] [DecidableEq Z] (F : Frame k V.rank Z) (c : kˣ) (zk : GL Z k), (zk : Matrix Z Z k) = (c : k) • (1 : Matrix Z Z k) → ∃ z : GA D V, p.IsLoc F z zk

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitLocalSupportedScalarComponentStatement

theorem **_root_.GlobalTheory.T.exists_splitLocal_supported_scalar_component** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (V : HSp D) (Z : Type) [Fintype Z] [DecidableEq Z] (F : Frame k V.rank Z) (c : kˣ) (zk : GL Z k), (zk : Matrix Z Z k) = (c : k) • (1 : Matrix Z Z k) → ∃ z : GA D V, p.IsLoc F z zk

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceComponentRingHomIntegral

theorem **pr_e** (p : SplitPt D k) : p.pr p.e = 1

theorem **prV_toEAf_e_ne** (p : SplitPt D k) {w : HeightOneSpectrum (𝒪 D.host.F)} (hw : w ≠ p.v) : Formalism.UG.prV (E

theorem **prV_toEAf_eps** (p : SplitPt D k) {ε : AF D.host.F} (hε : IsPlaceIdem D p.v ε) : Formalism.UG.prV (E

theorem **prF_eq_zero** (p : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt p.e PP) (y : EAf D) (hy : Formalism.UG.prV (E

theorem **prF_IE_prV** (p : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt p.e PP) (y : EAf D) : p.prF he (Formalism.UG.P.afsplitIE p.v (Formalism.UG.prV (E

theorem **phi_apply** (p : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt p.e PP) (x : D.host.E ⊗[D.host.F] p.v.adicCompletion D.host.F) : phi p he x = p.prF he (Formalism.UG.P.afsplitIE p.v x)

theorem **phi_prV** (p : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt p.e PP) (a : AE D) : phi p he (Formalism.UG.prV (E

theorem **phi_inclE** (p : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt p.e PP) (c : D.host.E) : phi p he (Formalism.UG.inclE (F

theorem **tensInt_iff** (v : HeightOneSpectrum (𝒪 D.host.F)) (z : D.host.E ⊗[D.host.F] v.adicCompletion D.host.F) : z ∈ Formalism.UG.tensInt (F

theorem **sigR_sigR** (z : EAf D) : Formalism.UG.sigR D.sp.cm.sig (Formalism.L2.Herm.Af D.host.F) (Formalism.UG.sigR D.sp.cm.sig (Formalism.L2.Herm.Af D.host.F) z) = z

theorem **toEAf_cA** (a : AE D) : toEAf D (cA D a) = Formalism.UG.sigR D.sp.cm.sig (Formalism.L2.Herm.Af D.host.F) (toEAf D a)

theorem **sigR_mem_iff** (s : EAf D) : Formalism.UG.sigR D.sp.cm.sig (Formalism.L2.Herm.Af D.host.F) s ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ s ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **one_sub_e_mul** (p : SplitPt D k) (y : EAf D) (hy : ∀ w, w ≠ p.v → Formalism.UG.prV (E

theorem **sigR_inclE** (v : HeightOneSpectrum (𝒪 D.host.F)) (c : D.host.E) : Formalism.UG.sigR (sigD D) (v.adicCompletion D.host.F) (Formalism.UG.inclE (F

theorem **exists_ringHom_int** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (PP : HeightOneSpectrum (𝒪 D.host.E)), Formalism.L2.NTheta.IsIdemAt p.e PP → ∃ φ : D.host.E ⊗[D.host.F] p.v.adicCompletion D.host.F →+* k, (∀ a : AE D, φ (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceComponentRingHomIntegralStatement

theorem **_root_.GlobalTheory.T.exists_splitPlace_component_ringHom_integral** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (PP : HeightOneSpectrum (𝒪 D.host.E)), Formalism.L2.NTheta.IsIdemAt p.e PP → ∃ φ : D.host.E ⊗[D.host.F] p.v.adicCompletion D.host.F →+* k, (∀ a : AE D, φ (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceGeometryExtensionCosetOrbitReps

theorem **exists_qExt_cos** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle), c.Is → ∀ (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ), (∀ j, IsUnram (η j)) → (∀ j, j ≤ G.s + 1 → (ofSplit G.q).IsComp (χ j)⁻¹ (η (j + 1) * δ)) → ∀ A, (G.T δ hδ η).IsRepSystem A G.ψk G.ν → ∀ M : ℕ, (G.T δ hδ η).Adapted A (box _ (G.ϖ ^ M)⁻¹) (box _ (G.ϖ ^ M)) → ∀ (C : Type) [Fintype C] (cs : C → XIx (Fin D.R.k) (Fin D.R.t) (G.s + 1) → G.k), (∀ y y', SameOrb ((G.T δ hδ η).lev (G.s + 1) 0 rfl).J (box _ (G.ϖ ^ M)) (Submodule.Quotient.mk (cs y)) (Submodule.Quotient.mk (cs y')) → y = y') → (∀ z ∈ box (XIx (Fin D.R.k) (Fin D.R.t) (G.s + 1)) (G.ϖ ^ M)⁻¹, ∃ y, SameOrb ((G.T δ hδ η).lev (G.s + 1) 0 rfl).J (box _ (G.ϖ ^ M)) (Submodule.Quotient.mk (cs y)) (Submodule.Quotient.mk z)) → ∃ E : QExt G χ, E.δ = δ ∧ E.η = η

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceGeometryExtensionCosetOrbitRepsStatement

theorem **_root_.GlobalTheory.T.exists_splitPlaceGeometryExtension_cosetOrbitReps** : ∀ (D : D7) (c : Comm D) (v : VCom c) (B : GlobalTheory.Fam.Base c) (PP : Formalism.L2.DIV.Pl D.host.E) (G : QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle), c.Is → ∀ (δ : G.kˣ →* ℂˣ) (hδ : IsTame δ) (η : ℕ → G.kˣ →* ℂˣ), (∀ j, IsUnram (η j)) → (∀ j, j ≤ G.s + 1 → (ofSplit G.q).IsComp (χ j)⁻¹ (η (j + 1) * δ)) → ∀ A, (G.T δ hδ η).IsRepSystem A G.ψk G.ν → ∀ M : ℕ, (G.T δ hδ η).Adapted A (box _ (G.ϖ ^ M)⁻¹) (box _ (G.ϖ ^ M)) → ∀ (C : Type) [Fintype C] (cs : C → XIx (Fin D.R.k) (Fin D.R.t) (G.s + 1) → G.k), (∀ y y', SameOrb ((G.T δ hδ η).lev (G.s + 1) 0 rfl).J (box _ (G.ϖ ^ M)) (Submodule.Quotient.mk (cs y)) (Submodule.Quotient.mk (cs y')) → y = y') → (∀ z ∈ box (XIx (Fin D.R.k) (Fin D.R.t) (G.s + 1)) (G.ϖ ^ M)⁻¹, ∃ y, SameOrb ((G.T δ hδ η).lev (G.s + 1) 0 rfl).J (box _ (G.ϖ ^ M)) (Submodule.Quotient.mk (cs y)) (Submod

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceInvariantFormOfWittCore

theorem **exists_form_of_core** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ ℱ → ∀ e, IsSplitIdem D e → ∀ ℓ₀ : (WO S₀ T → ℂ) →ₗ[ℂ] ℂ, (∀ u : WO S₀ T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S₀ T f → ℓ₀ (heisO ψ S₀ T u f) = ℓ₀ f) → (∃ f, IsThetaFunO ψ S₀ T f ∧ ℓ₀ f ≠ 0) → ∃ ℓ : (WO S T → ℂ) →ₗ[ℂ] ℂ, (∀ u : WO S T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S T f → ℓ (heisO ψ S T u f) = ℓ f) ∧ (∃ f, IsThetaFunO ψ S T f ∧ ℓ f ≠ 0) ∧ ∀ (h : GL (Fin T.rank) (AE D)) (a : GL (Fin ρ) (AE D)), ∃ r : ℝ, 0 < r ∧ ∀ (f f' : WO S T → ℂ) (k : ℂ), IsThetaFunO ψ S T f → IsThetaFunO ψ S T f' → (∀ x : Mx D T.rank ρ, (∀ i j, cA D e * x i j = x i j) → ℓ₀ (d.mixS T ψ μ ℱ (f') x) = k * ℓ₀ (d.mixS T ψ μ ℱ (f) (h.1 · x · a.1))) → ℓ f' = r *

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceInvariantFormOfWittCoreStatement

theorem **_root_.GlobalTheory.T.exists_splitPlace_invariantForm_of_wittCore** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ ℱ → ∀ e, IsSplitIdem D e → ∀ ℓ₀ : (WO S₀ T → ℂ) →ₗ[ℂ] ℂ, (∀ u : WO S₀ T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S₀ T f → ℓ₀ (heisO ψ S₀ T u f) = ℓ₀ f) → (∃ f, IsThetaFunO ψ S₀ T f ∧ ℓ₀ f ≠ 0) → ∃ ℓ : (WO S T → ℂ) →ₗ[ℂ] ℂ, (∀ u : WO S T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S T f → ℓ (heisO ψ S T u f) = ℓ f) ∧ (∃ f, IsThetaFunO ψ S T f ∧ ℓ f ≠ 0) ∧ ∀ (h : GL (Fin T.rank) (AE D)) (a : GL (Fin ρ) (AE D)), ∃ r : ℝ, 0 < r ∧ ∀ (f f' : WO S T → ℂ) (k : ℂ), IsThetaFunO ψ S T f → IsThetaFunO ψ S T f' → (∀ x : Mx D T.rank ρ, (∀ i j, cA D e * x i j = x i j) → ℓ₀ (d.mixS T ψ μ ℱ (f') x) = k * ℓ₀ (d.m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceModelExtensionLineLattice

theorem **isLattice_span** {D : D7} {L : HSp D} {ζ : Fin L.rank → D.host.E} (h : IsLineVec L ζ) : Submodule.IsLattice D.host.E (Submodule.span (NumberField.RingOfIntegers D.host.E) {ζ} : Lat L)

theorem **exists_placeModel_ext** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (S L'' : HSp D) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L'' ζ₀ → ∀ (LS : Lat S), IsSelfDualC S LS → ∀ (n : Type) [Fintype n] [DecidableEq n] (FS : GlobalTheory.LocMod.Frame k S.rank n) (FL'' : GlobalTheory.LocMod.Frame k L''.rank Unit) (TL : SB (Matrix Unit n k) →ₗ[ℂ] (WO S L'' → ℂ)),

(GlobalTheory.LocMod.Place.ofSplitPt p).IsModel ψ FS FL'' TL → ∃ (g₃ : WO S L'' → ℂ) (T₃ : (WO S L'' → ℂ) →ₗ[ℂ] (WO S L'' → ℂ)), IsThetaFunO ψ S L'' g₃ ∧ IsPlaceModel ψ (symplO S L'') (heisO ψ S L'') p.e (latO S L'' (LS : Set _) ((Submodule.span (NumberField.RingOfIntegers D.host.E) {ζ₀} : Lat L'') : Set _) (· = p.v)) 1 g₃ T₃ ∧ ∀ Φ, T₃ (GlobalTheory.LocMod.Witness.ext (GlobalTheory.LocMod.Plac

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPlaceModelExtensionLineLatticeStatement

theorem **_root_.GlobalTheory.T.exists_splitPlaceModel_extension_lineLattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (S L'' : HSp D) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L'' ζ₀ → ∀ (LS : Lat S), IsSelfDualC S LS → ∀ (n : Type) [Fintype n] [DecidableEq n] (FS : GlobalTheory.LocMod.Frame k S.rank n) (FL'' : GlobalTheory.LocMod.Frame k L''.rank Unit) (TL : SB (Matrix Unit n k) →ₗ[ℂ] (WO S L'' → ℂ)), (GlobalTheory.LocMod.Place.ofSplitPt p).IsModel ψ FS FL'' TL → ∃ (g₃ : WO S L'' → ℂ) (T₃ : (WO S L'' → ℂ) →ₗ[ℂ] (WO S L'' → ℂ)), IsThetaFunO ψ S L'' g₃ ∧ IsPlaceModel ψ (symplO S L'') (heisO ψ S L'') p.e (latO S L'' (LS : Set _) ((Submodule.span (NumberField.RingOfIntegers D.host.E) {ζ₀} : Lat L'') : Set _) (· = p.v)) 1 g₃ T₃ ∧ ∀ Φ, T₃ (GlobalTheory

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPointFrameCharacter

theorem **exists_char** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∃ χ : AddChar k Circle, p.IsChar ψ χ ∧ Continuous χ

theorem **exists_splitPt_frame_char** : ∀ (D : D7), (∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [IsTopologicalRing k] [CharZero k] (T : ℕ), T.Prime → Nat.card (IsLocalRing.ResidueField 𝒪[k]) = T → Irreducible (T : 𝒪[k]) → ∀ (w : Formalism.L2.DIV.Pl D.host.E) (e : AE D), Formalism.L2.NTheta.IsIdemAt e w → IsSplitIdem D e → Ideal.absNorm w.asIdeal = T → ¬ (w.asIdeal ^ 2 ∣ Ideal.span {(T : 𝒪 D.host.E)}) → ∃ p : SplitPt D k, p.e = e) ∧ ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k), (∀ (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (b : Matrix (Fin S.rank) n k) (b' : Matrix n (Fin S.rank) k), b ∗ b' = 1 → b' ∗ b = 1 → ∃ β : Frame p S n, β.b = b) ∧ ∀ ψ : AddChar (AF D.host.F) ℂ, IsStdChar D ψ (-1) → ∃ χ : AddChar k Circle, p.IsChar ψ χ ∧ Continuous χ ∧ χ ≠ 1 ∧ (p.UnitAA → GlobalTheo

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPointFrameCharacterStatement

theorem **_root_.GlobalTheory.T.exists_splitPoint_frame_character** : ∀ (D : D7), (∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [IsTopologicalRing k] [CharZero k] (T : ℕ), T.Prime → Nat.card (IsLocalRing.ResidueField 𝒪[k]) = T → Irreducible (T : 𝒪[k]) → ∀ (w : Formalism.L2.DIV.Pl D.host.E) (e : AE D), Formalism.L2.NTheta.IsIdemAt e w → IsSplitIdem D e → Ideal.absNorm w.asIdeal = T → ¬ (w.asIdeal ^ 2 ∣ Ideal.span {(T : 𝒪 D.host.E)}) → ∃ p : SplitPt D k, p.e = e) ∧ ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k), (∀ (S : HSp D) (n : Type) [Fintype n] [DecidableEq n] (b : Matrix (Fin S.rank) n k) (b' : Matrix n (Fin S.rank) k), b ∗ b' = 1 → b' ∗ b = 1 → ∃ β : Frame p S n, β.b = b) ∧ ∀ ψ : AddChar (AF D.host.F) ℂ, IsStdChar D ψ (-1) → ∃ χ : AddChar k Circle, p.IsChar ψ χ ∧ Continuous χ ∧

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPointWeilCharConductorOneOfUnits

theorem **splitPlace_polarisationDatum_inv_integral** (q : SplitPt D k) (hU : q.UnitAA) : q.pr (incA D D.host.xi0⁻¹) ∈ 𝒪[k]

theorem **valuation_pr_incA_xi0_eq_one_of_unitAA** (q : SplitPt D k) (hU : q.UnitAA) : valuation k (q.pr (incA D D.host.xi0)) = 1

theorem **weil_char_conductor_one** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → q.UnitAA → ∃ ψk : AddChar k Circle, (ofSplit q).IsChar ψ ψk ∧ HasConductor ψk 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSplitPointWeilCharConductorOneOfUnitsStatement

theorem **_root_.GlobalTheory.T.exists_splitPoint_weilChar_conductor_one_of_units** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → q.UnitAA → ∃ ψk : AddChar k Circle, (ofSplit q).IsChar ψ ψk ∧ HasConductor ψk 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStandardCartanInvolutionCommuteCompact

theorem **comm_of_unitary_isom** {m : ℕ} {u J : Matrix (Fin m) (Fin m) ℂ} (hu : uᴴ ∗ u = 1) (hJ : uᴴ ∗ J ∗ u = J) : u ∗ J = J ∗ u

theorem **compAt_K_comm** (θ : InfFam D) (hθ : IsInfFam θ) (n : ℕ) (k : G D n) (hk : k ∈ K D n) (τ : D.what₂.Tau) : compAt θ τ (mat k) ∗ blk4 0 1 (-1) 0 = blk4 0 1 (-1) 0 ∗ compAt θ τ (mat k)

theorem **exists_stdInv_comm_K** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∀ θ : InfFam D, IsInfFam θ → ∃ s' : RealMat (dblSp A), IsStdInv A μ s' ∧ ∀ k : GA D (dblSp A), κ k ∈ K D A.rank → ∀ τ, compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k) ∗ s' τ = s' τ ∗ compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStandardCartanInvolutionCommuteCompactStatement

theorem **_root_.GlobalTheory.T.exists_standardCartanInvolution_commute_compact** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ → ∀ θ : InfFam D, IsInfFam θ → ∃ s' : RealMat (dblSp A), IsStdInv A μ s' ∧ ∀ k : GA D (dblSp A), κ k ∈ K D A.rank → ∀ τ, compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k) ∗ s' τ = s' τ ∗ compAt θ τ (Formalism.L2.Theta.mt (dblSp A) k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStandardLocalisationWeilDatumInertPlace

theorem **exists_isStd_inert** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), IsInertAt D.host.F D.host.E v w → UnrAt χS w → UnrAt χT w → ∀ c ϖ eS eT, IsUnramInv ϖ c (qV D.host.F v) → IsInertChart D.sp.cm.sig S.gram (frameLat S) h c eS → IsInertChart D.sp.cm.sig T.gram (frameLat T) h c eT → ∀ f, VacFin ψ S T v f → f ≠ 0 → ∃ (δ : w.adicCompletion D.host.E) (ψ' : AddChar (w.adicCompletion D.host.E) ℂ) (M : Submodule ℂ (WO S T → ℂ)) (ρ : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E) → Module.End ℂ M) (ω : GU c S.rank × GU c T.rank →* Module.End ℂ M) (s : M), IsStd ϖ c (qV D.host.F v) δ ψ' ρ ω s ∧ (s : WO S T → ℂ) = f ∧ (∀ g g' (m : M), ((ω (g, g') m : M)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStandardLocalisationWeilDatumInertPlaceStatement

theorem **_root_.GlobalTheory.T.exists_standardLocalisation_weilDatum_inertPlace** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), IsInertAt D.host.F D.host.E v w → UnrAt χS w →

UnrAt χT w → ∀ c ϖ eS eT, IsUnramInv ϖ c (qV D.host.F v) → IsInertChart D.sp.cm.sig S.gram (frameLat S) h c eS → IsInertChart D.sp.cm.sig T.gram (frameLat T) h c eT → ∀ f, VacFin ψ S T v f → f ≠ 0 → ∃ (δ : w.adicCompletion D.host.E) (ψ' : AddChar (w.adicCompletion D.host.E) ℂ) (M : Submodule ℂ (WO S T → ℂ)) (ρ : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E) → Module.End ℂ M) (ω : GU c S.rank × GU c T.rank →* Module.End ℂ M) (s : M), IsStd ϖ c (qV D.host.F v) δ ψ' ρ ω s ∧ (s : WO

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStandardUnitaryChartDoubledSpace

theorem **cT_mul_gen** (σ : R →+* R) {m p q : Type} [Fintype p] (M : Matrix m p R) (N : Matrix p q R) : cT σ (M * N) = cT σ N * cT σ M

theorem **cT_one_gen** (σ : R →+* R) {n : Type} [Fintype n] [DecidableEq n] : cT σ (1 : Matrix n n R) = 1

theorem **conj_mem** (σ : R →+* R) (Q J : Matrix n n R) (T : GL n R) (c : Rˣ) (hT : cT σ (T : Matrix n n R) * Q * T = (c : R) • J) (g : GL n R) (hg : g ∈ UJ σ Q) : T⁻¹ * g * T ∈ UJ σ J

theorem **conj_back** (σ : R →+* R) (Q J : Matrix n n R) (T : GL n R) (c : Rˣ) (hT : cT σ (T : Matrix n n R) * Q * T = (c : R) • J) : cT σ ((T⁻¹ : GL n R) : Matrix n n R) * J * ((T⁻¹ : GL n R) : Matrix n n R) = ((c⁻¹ : Rˣ) : R) • Q

theorem **GlobalTheory.StdUnn.ChartExist.gram_adelic** (D : D7) (A : HSp D) (μ : D.host.E) (hgram : (Tm A μ)ᴴ * (dblSp A).gram * Tm A μ = μ • Jst D A.rank) (T : GL (Fin (A.rank + A.rank)) (AE D)) (hTv : T.val = rat (Tm A μ)) (c : (AE D)ˣ) (hc : (c : AE D) = incA D μ) : Formalism.UG.cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF D.host.F) D.sp.cm.sig) T.val * (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR (AF D.host.F) (dblSp A).gram : Matrix (Fin (A.rank + A.rank)) (Fin (A.rank + A.rank)) (AE D)) * T.val = (c : AE D) • Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR (AF D.host.F) (Jst D A.rank)

theorem **GlobalTheory.StdUnn.chart_exists** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∃ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStandardUnitaryChartDoubledSpaceStatement

theorem **_root_.GlobalTheory.T.exists_standardUnitaryChart_doubledSpace** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∃ κ : GA D (dblSp A) ≃* G D A.rank, IsChart A μ κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStepPointOfChartBasePoint

theorem **pointOf_base** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (u : B.Word) (a : I.ι) (i : ℕ), i ≤ B.R.m → ∀ {C : Type} (chOf : (I.stp u a i).Cl → C) (Lb : C → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)) (U : C → Subgroup (TC.GA B (B.R.N i))) (chi : ∀ β, U β →* ℂˣ) (y : (I.stp u a i).Cl → TC.GA B (B.R.N i)) (st : ∀ c, (I.stp u a i).Stab c → GL (Fin (B.R.N i)) B.host.E) (k : C → TC.GA B (B.R.N i)), I.Reads u a i chOf Lb U chi y st k → ∀ β, (∃ q : Formalism.L2.Col.ChPt (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsStepPointOfChartBasePointStatement

theorem **_root_.GlobalTheory.T.exists_stepPoint_of_chartBasePoint** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (u : B.Word) (a : I.ι) (i : ℕ), i ≤ B.R.m → ∀ {C : Type} (chOf : (I.stp u a i).Cl → C) (Lb : C → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)) (U : C → Subgroup (TC.GA B (B.R.N i))) (chi : ∀ β, U β →* ℂˣ) (y : (I.stp u a i).Cl → TC.GA B (B.R.N i)) (st : ∀ c, (I.stp u a i).Stab c → GL (Fin (B.R.N i)) B.host.E) (k : C → TC.GA B (B.R.N i)), I.Reads u a i chOf Lb U chi y st k → ∀ β, (∃ q : Formalism.L2.Col.ChPt (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSumFrame

theorem **card_eq_of_inv** {k : Type} [Field k] {m n : Type} [Fintype m] [DecidableEq m] [Fintype n] [DecidableEq n] (b : Matrix m n k) (b' : Matrix n m k) (h : b * b' = 1) (h' : b' * b = 1) : Fintype.card n = Fintype.card m

theorem **exists_sumFrame'** (p : SplitPt D k) {S A L : HSp D} (o : OrthDec S A L) {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) {n : Type} [Fintype n] [DecidableEq n] (β : Frame p A n) : ∃ β₀ : Frame p S (n ⊕ Fin 1), IsSumFrame o ζ β β₀

theorem **line_of_orth** {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M : Lat S) (LA : Lat A) (LL : Lat L) (v : Pl D) (ho : OrthAt o M LA LL v) (a : AE D) : a • ratV (vecB o ζ) ∈ latV S M (· = v) ↔ a • ratV ζ ∈ latV L LL (· = v)

theorem **GlobalTheory.FinPl.CoreIdele.lineLat_idele** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (L : HSp D) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ m : (AE D)ˣ, (∀ x, ¬ P x → ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (m : AE D) * cA D (m : AE D) * incA D (star ζ ·ᵥ L.gram *ᵥ ζ) * a ∈ intAt D (· = x)) → ∃ LL : Lat L, IsSelfDualC L LL ∧ ∀ x, ¬ P x → ∀ a ∈ idAt D (· = x), a • ratV ζ ∈ latV L LL (· = x) ↔ ((m⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **GlobalTheory.FinPl.CoreIdele.exists_core** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (q : Unif k) (n : Type) [Fintype n] [DecidableEq n] (ρ : ℕ) (B : BesLev D ρ) (β : ∀ i, Frame (p i) B.A n) (Q : ι → Pair k n) (LA : Lat B.A), IsSelfDualC B.A LA → (∀ i, (β i).IsLat LA (Q i).lat) → (∀ i, valuation k ((p i).pr (incA D (star B.ζ ·ᵥ B.L.gram *ᵥ B.ζ))) = valuation k ((q.1 : 𝒪[k]) : k)) → ∀ m : (AE D)ˣ, (∀ x, P x → ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (m : AE D) * a ∈ intAt D (· = x)) → (∀ x, ¬ P x → ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (m : AE D) * cA D (m : AE D) * incA D (star B.ζ ·ᵥ B.L.gram *ᵥ B.ζ) * a ∈ intAt D (· = x)) → ∃ (M : Lat B.V₀) (β₀ : ∀ i, Frame (p i) B.V₀ (n ⊕ Fin 1)), IsSelfDualC B.V₀ M ∧ (∀ i, GluedCore B M q (β i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSumHyperbolicPlanesOrthogonalWittDecomposi

theorem **gramAt_hypSp** (D : D7) (t : ℕ) (τ : D.what₂.Tau) : gramAt D (GlobalTheory.OrthSumSp.hypSp D t) τ = (Matrix.fromBlocks (1 : Matrix (Fin t) (Fin t) ℂ) 0 0 (-1)).submatrix finSumFinEquiv.symm finSumFinEquiv.symm

theorem **gsq** (D : D7) (t : ℕ) (τ : D.what₂.Tau) : gramAt D (GlobalTheory.OrthSumSp.hypSp D t) τ * gramAt D (GlobalTheory.OrthSumSp.hypSp D t) τ = 1

theorem **gct** (D : D7) (t : ℕ) (τ : D.what₂.Tau) : (gramAt D (GlobalTheory.OrthSumSp.hypSp D t) τ)ᴴ = gramAt D (GlobalTheory.OrthSumSp.hypSp D t) τ

theorem **exists_hypSum_orthDec_hypDec** : ∀ (D : D7) (W : HSp D) (t : ℕ), HSpace.SigOK W → ∃ (H B : HSp D) (_ : OrthDec B W H) (_ : HypDec B W t), HSpace.SigOK H ∧ HSpace.SigOK B ∧ H.rank = 2 * t ∧ (∀ τ, H.sig τ = (t, t)) ∧ ∃ sH : RealMat H, ∀ τ, IsCartanInv (gramAt D H τ) (sH τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSumHyperbolicPlanesOrthogonalWittDecomposiStatement

theorem **_root_.GlobalTheory.T.exists_sum_hyperbolicPlanes_orthogonal_wittDecomposition** : ∀ (D : D7) (W : HSp D) (t : ℕ), HSpace.SigOK W → ∃ (H B : HSp D) (_ : OrthDec B W H) (_ : HypDec B W t), HSpace.SigOK H ∧ HSpace.SigOK B ∧ H.rank = 2 * t ∧ (∀ τ, H.sig τ = (t, t)) ∧ ∃ sH : RealMat H, ∀ τ, IsCartanInv (gramAt D H τ) (sH τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSummableBoundSchwartzMajorantRationalTrans

theorem **exists_compact_sub** {G : Type*} [AddGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] (K Q : Set G) (hK : IsCompact K) (hQ : IsCompact Q) : ∃ K' : Set G, IsCompact K' ∧ ∀ k ∈ K, ∀ x ∈ Q, k − x ∈ K'

theorem **summable_of_tsum_ne_top** {α β : Type*} (f : β → ℝ) (hf : ∀ y, 0 ≤ f y) (h : (∑' y, ENNReal.ofReal (f y)) ≠ ⊤) (e : α → β) (he : Function.Injective e) : Summable fun a => f (e a)

theorem **inv_pow_le** {a c R : ℝ} (N : ℕ) (ha : 0 ≤ a) (hc : 0 ≤ c) (hR : 0 ≤ R) (h : c ≤ a + R) : ((1 + a) ^ N)⁻¹ ≤ (1 + R) ^ N * ((1 + c) ^ N)⁻¹

theorem **realSize_neg** {D : D7} (θ : InfFam D) {m n : ℕ} (x : Mx D m n) : realSize θ (−x) = realSize θ x

theorem **maj_add_le** {D : D7} (θ : InfFam D) {m n : ℕ} (K K' : Set (Mx D m n)) (N : ℕ) (x y : Mx D m n) (R : ℝ) (hR : 0 ≤ R) (hx : realSize θ x ≤ R) (hK : ∀ k ∈ K, k − x ∈ K') : maj θ K N (x + y) ≤ (1 + R) ^ N * maj θ K' N y

theorem **summable_bound** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ m n : ℕ, ∃ N : ℕ, ∀ K Q : Set (Mx D m n), IsCompact K → IsCompact Q → ∃ b : Matrix (Fin m) (Fin n) D.host.E → ℝ, Summable b ∧ ∀ x ∈ Q, ∀ ξ, GlobalTheory.Unf.maj θ K N (x + rat ξ) ≤ b ξ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSummableBoundSchwartzMajorantRationalTransStatement

theorem **_root_.GlobalTheory.T.exists_summable_bound_schwartz_majorant_rational_translates** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ m n : ℕ, ∃ N : ℕ, ∀ K Q : Set (Mx D m n), IsCompact K → IsCompact Q → ∃ b : Matrix (Fin m) (Fin n) D.host.E → ℝ, Summable b ∧ ∀ x ∈ Q, ∀ ξ, GlobalTheory.Unf.maj θ K N (x + rat ξ) ≤ b ξ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSuppVecLocEq

theorem **mem_latV_iff_latLoc** (V : HSp D) (L : Lat V) (hL : L.FG) (v : Pl D) (y : Fin V.rank → AE D) (hy : ∀ i, y i ∈ idAt D (· = v)) : y ∈ GlobalTheory.FinPl.latV V L (· = v) ↔ GlobalTheory.AdelicU.vecLoc v (finPart D ∘ y) ∈ GlobalTheory.AdelicU.latLoc D.host.F L v

theorem **exists_supp_vecLoc_eq** {n : ℕ} (v : Pl D) (z : Fin n → Fv D.host.F v ⊗[D.host.F] D.host.E) : ∃ y : Fin n → AE D, (∀ i, y i ∈ idAt D (· = v)) ∧ GlobalTheory.AdelicU.vecLoc v (finPart D ∘ y) = z

theorem **moves_iff_stabLoc** (V : HSp D) (L : Lat V) (hL : L.FG) (v : Pl D) (g : GA D V) : Moves V g L L (· = v) ↔ GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram L v

theorem **moves_mul** {V : HSp D} {g h : GA D V} {L M N : Lat V} {P : Pl D → Prop} (hg : Moves V g L M P) (hh : Moves V h M N P) : Moves V (h * g) L N P

theorem **moves_inv** {V : HSp D} {g : GA D V} {L M : Lat V} {P : Pl D → Prop} (hg : Moves V g L M P) : Moves V g⁻¹ M L P

theorem **moves_conj_iff** {V : HSp D} {t : GA D V} (h : GA D V) {L M : Lat V} {P : Pl D → Prop} (ht : Moves V t L M P) : Moves V (t⁻¹ * h * t) L L P ↔ Moves V h M M P

theorem **finE_conj_mem_St_iff** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Λ M : GlobalTheory.FinPl.Lat V), Formalism.L2.DIV.IsFrame β Λ → IsSelfDualC V Λ → IsSelfDualC V M → ∀ t h : GA D V, t ∈ GfC V → h ∈ GfC V → (∀ P, GlobalTheory.FinPl.Moves V t Λ M P) → (Lev.finE V (t⁻¹ * h * t) ∈ Formalism.L2.DIV.St β ↔ ∀ v : GlobalTheory.FinPl.Pl D, GlobalTheory.FinPl.Moves V h M M (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSymDataOfUnifCongr

theorem **apply_mul_KOne** {f : Matrix ι ι' k → ℂ} (hs : ∀ x, f x ≠ 0 → IntM x) (hp : ∀ x y, MaxM y → f (x + y) = f x) {a : GL ι k} (ha : a ∈ KOne k ι) {b : GL ι' k} (hb : b ∈ KOne k ι') (x : Matrix ι ι' k) : f (a.val * x * b.val) = f x

theorem **reGL_symm_mem_KOne** (e : ι ≃ Y) {d : GL Y k} (hd : d ∈ KOne k Y) : (reGL e).symm d ∈ KOne k ι

theorem **Near.conj** {F G : LatFrame k ι Y} (h : Near F G) : ∃ d ∈ KOne k ι, ∀ g, G.conj.symm g = d⁻¹ * F.conj.symm g * d

theorem **Near.J_eq** {F G : LatFrame k ι Y} (h : Near F G) (P : TypeDatum k ι) : (P.toLevel.map G.conj).J = (P.toLevel.map F.conj).J

theorem **Near.stabLat_eq** {F G : LatFrame k ι Y} (h : Near F G) : stabLat G.bas = stabLat F.bas

theorem **Near.transport_eq** {F G : LatFrame k ι Y} {F' G' : LatFrame k ι' Y'} (h : Near F G) (h' : Near F' G') {f : Matrix ι ι' k → ℂ} (hs : ∀ x, f x ≠ 0 → IntM x) (hp : ∀ x y, MaxM y → f (x + y) = f x) : G.transport G' f = F.transport F' f

theorem **near_refl** (F : LatFrame k ι Y) : Near F F

theorem **val_pow_sub_one** {u : k} (hu : valuation k (u − 1) < 1) (m : ℕ) : valuation k (u ^ m − 1) < 1

theorem **val_div_sub_one** (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) : valuation k ((ϖ : k)⁻¹ * ϖ' − 1) < 1

theorem **near_typeFrame** {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) (o : ι ≃ n ⊕ Fin c) : Near (typeFrame ϖ o) (typeFrame ϖ' o)

theorem **val_eq** (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) : valuation k (ϖ' : k) = valuation k (ϖ : k)

theorem **isUnif_of_near** (hϖ : IsUnif ϖ) (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) : IsUnif ϖ'

theorem **inv_sub_inv_mem** (hϖ : IsUnif ϖ) (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) : valuation k (((ϖ'⁻¹ : kˣ) : k) − ((ϖ⁻¹ : kˣ) : k)) ≤ 1

theorem **psi_inv_mul** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (hϖ : IsUnif ϖ) (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) {x : k} (hx : x ∈ 𝒪[k]) : ψ ((ϖ'⁻¹ : kˣ) * x) = ψ ((ϖ⁻¹ : kˣ) * x)

theorem **isSymLevel_of_near** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) (hϖ : IsUnif ϖ) (h : valuation k ((ϖ' : k) − (ϖ : k)) < valuation k (ϖ : k)) {ρ κ l : ℕ} {P : TypeDatum k (Fin κ ⊕ Fin l)} (hP : IsSymLevel ψ ϖ ρ P) : IsSymLevel ψ ϖ' ρ P

theorem **transport_iterPhi** (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) {F G : LatFrame k (Ix r t (j + 1)) Y} {F' G' : LatFrame k (Ix r t j) Y'} (h : Near F G) (h' : Near F' G') : G.transport G' (P.iterPhi t μ hμ j) = F.transport F' (P.iterPhi t μ hμ j)

theorem **exists_symData_of_unif_congr** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {s κ : ℕ} (S : SymData k s κ) (ϖ' : kˣ), valuation k ((ϖ' : k) − (S.ϖ : k)) < valuation k (S.ϖ : k) → ∃ S' : SymData k s κ, S'.ϖ = ϖ' ∧ S'.ψ = S.ψ ∧ S'.ρ = S.ρ ∧ S'.P = S.P ∧ S'.μ = S.μ ∧ ∀ (τ : ℕ) (ν : Measure k), (∀ j c h, (S'.T (Fin τ)).Phi j c h = (S.T (Fin τ)).Phi j c h) ∧ (∀ j c h, (S'.T (Fin τ)).entry S'.ψ ν j c h 1 1 = (S.T (Fin τ)).entry S.ψ ν j c h 1 1) ∧ (∀ x₀, (S'.T (Fin τ)).Psi 1 x₀ (S'.y0 (Fin τ)) = (S.T (Fin τ)).Psi 1 x₀ (S.y0 (Fin τ))) ∧ (S'.T (Fin τ)).xiTop (box _ (1 : kˣ)) S'.ψ ν (−1) (Submodule.Quotient.mk (S'.y0 (Fin τ))) = (S.T (Fin τ)).xiTop (box _ (1 : kˣ)) S.ψ ν (−1) (Submodule.Quotient.mk (S.y0 (Fin τ)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSymmetricDataAtTypePlaces

theorem **tame_step** {D : D7} {T : Type} (L : LevCh D T) {μ0 : T} (hL : L.Is μ0) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {PP : Pl D.host.E} (hP : PP ∈ L.Pg) (h : Formalism.L2.NTheta.IsIdemAt q.e PP) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) {μ : kˣ →* ℂˣ} (hμ : (ofSplit q).IsComp (Lift.chrStep L.mu0 L.eta j).1⁻¹ μ) : IsTame μ

theorem **red_sym** {D : D7} {a : XGeo D} {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (S : SymData l.k l.s D.R.k) {π : Equiv.Perm (Fin D.R.r)} (hπ : IsSymOrd D.R.rS D.R.k π) (g g' : GL (Fin D.R.r) (kP PP)) (hg : ∀ i j, g'.val i j = g.val (π i) (π j)) : ((∃ x, l.Red S x g) ↔ g' ∈ unipSym (kP PP) D.R.r D.R.rS D.R.k) ∧ ∀ ψ₀ : AddChar (kP PP) ℂ, (∀ z, (S.ψ (((S.ϖ⁻¹ : l.kˣ) : l.k) * prO l.q z) : ℂ) = ψ₀ (Ideal.Quotient.mk _ z)) → ∀ x, l.Red S x g → (S.P.lam x : ℂ) = ψ₀ (edgeSum (symEdge D.R.rS D.R.k) g')

theorem **exists_symRead** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg → ∀ (a : XGeo D) (l : ∀ PP : (pk W).Pg, LocX a (pk W).βX PP.1), ∃ S : ∀ PP, SymData (l PP).k (l PP).s D.R.k, SymRead W l S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSymmetricDataAtTypePlacesStatement

theorem **_root_.GlobalTheory.T.exists_symmetric_data_at_type_places** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg → ∀ (a : XGeo D) (l : ∀ PP : (pk W).Pg, LocX a (pk W).βX PP.1), ∃ S : ∀ PP, SymData (l PP).k (l PP).s D.R.k, SymRead W l S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSymmetricDataConductorOne

theorem **sec_one** : q.sec 1 = q.e

theorem **pr_e** : q.pr q.e = 1

theorem **pr_e_mul** (x : AE D) : q.pr (q.e * x) = q.pr x

theorem **place_facts** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) : (∀ a, (ofSplit q).pr a ∈ 𝒪[k] ↔ IntAt PP a) ∧ DenseRange (ofSplit q).pr

theorem **prO_le** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) (z : 𝒪 D.host.E) : valuation k (prO q z) ≤ 1

theorem **prO_lt** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) (z : 𝒪 D.host.E) : z ∈ PP.asIdeal ↔ valuation k (prO q z) < 1

theorem **prO_eq_one** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) {z : 𝒪 D.host.E} (hz : z ∉ PP.asIdeal) : valuation k (prO q z) = 1

theorem **sec_mul** (x y : k) : q.sec x * q.sec y = q.sec (x * y)

theorem **idl_mul** (x y : k) (h : x * y = 1) : (q.sec x + (1 - q.e)) * (q.sec y + (1 - q.e)) = 1

theorem **continuous_idl** : Continuous (idl q)

theorem **unram_idl** {χ : (AE D)ˣ →* Circle} (hχ : UnramEverywhere D χ) (x : kˣ) (hx : valuation k (x : k) = 1) : χ (idl q x) = 1

theorem **res_idl** {PP : Formalism.L2.DIV.Pl D.host.E} (h : IsIdemAt q.e PP) {χ : (AE D)ˣ →* Circle} {H : Formalism.L2.Host.HChar D.host.E} {c : (Formalism.L2.DIV.kP PP)ˣ →* ℂˣ} (hχ : Formalism.IndefR.Links.ChrEq χ H) (hr : Formalism.IndefR.Links.IsRes H PP c) (x : kˣ) (hx : valuation k ((x : k) - 1) < 1) : χ (idl q x) = 1

theorem **tame_comp** {c : GlobalTheory.ChainOnto.Com D} (hc : c.Is) {PP : Formalism.L2.DIV.Pl D.host.E} (hP : PP ∈ c.Pg) (h : IsIdemAt q.e PP) {j : ℕ} (h1 : 1 ≤ j) (h2 : j ≤ D.R.m) {μ : kˣ →* ℂˣ} (hμ : (ofSplit q).IsComp (c.chS j)⁻¹ μ) : IsTame μ

theorem **exists_symData_at** : ∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (PP : Formalism.L2.DIV.Pl D.host.E), d.c.Is → PP ∈ d.c.Pg → Formalism.L2.NTheta.IsIdemAt q.e PP → ∀ ψk : AddChar k Circle, (ofSplit q).IsChar d.c.ψ ψk → HasConductor ψk 1 → ∀ s ρ : ℕ, s = 2 * ρ → ∃ S : SymData k s D.R.k, S.ψ = ψk ∧ ∀ j, 1 ≤ j → j ≤ D.R.m → (ofSplit q).IsComp (d.c.chS j)⁻¹ (S.μ j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsSymmetricDataConductorOneStatement

theorem **_root_.GlobalTheory.T.exists_symmetric_data_conductor_one** : ∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (PP : Formalism.L2.DIV.Pl D.host.E), d.c.Is → PP ∈ d.c.Pg → Formalism.L2.NTheta.IsIdemAt q.e PP → ∀ ψk : AddChar k Circle, (ofSplit q).IsChar d.c.ψ ψk → HasConductor ψk 1 → ∀ s ρ : ℕ, s = 2 * ρ → ∃ S : SymData k s D.R.k, S.ψ = ψk ∧ ∀ j, 1 ≤ j → j ≤ D.R.m → (ofSplit q).IsComp (d.c.chS j)⁻¹ (S.μ j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTameRamified

theorem **mem_intRing** (z : AE D) : z ∈ intRing D ↔ finPart D z ∈ intAE D

theorem **finPart_cA** (z : AE D) : finPart D (cA D z) = sigR (Af D.host.F) D.sp.cm.sig (finPart D z)

theorem **cA_cA'** (z : AE D) : cA D (cA D z) = z

theorem **cA_mem** {z : AE D} (hz : z ∈ intRing D) : cA D z ∈ intRing D

theorem **Dvd.add** {N : ℕ} {y z : AE D} (hy : Dvd N y) (hz : Dvd N z) : Dvd N (y + z)

theorem **Dvd.mul** {N : ℕ} {y z : AE D} (hy : Dvd N y) (hz : z ∈ intRing D) : Dvd N (y * z)

theorem **dvd_mul** {N : ℕ} {z : AE D} (hz : z ∈ intRing D) : Dvd N ((N : AE D) * z)

theorem **isOpen_dvd** {N : ℕ} (hN : 0 < N) : IsOpen {z : AE D | Dvd N z}

theorem **isOpen_intRing** : IsOpen {z : AE D | z ∈ intRing D}

theorem **eq_one_of_dvd** {N : ℕ} (hN : 2 < N) {a : D.host.E} (ha : star a * a = 1) (h : Dvd N (incA D a - 1)) : a = 1

theorem **exists_unit_nat** (k : Type) [Field k] [ValuativeRel k] : ∃ N : ℕ, 2 < N ∧ valuation k (N : k) = 1

theorem **exists_unit_ne_one** (h : 2 < Nat.card (IsLocalRing.ResidueField 𝒪[k])) : ∃ u : k, valuation k u = 1 ∧ ¬ valuation k (u - 1) < 1

theorem **near_one** : Near p N 1

theorem **Near.val** {y : AE D} (h : Near p N y) : valuation k (p.pr y) = 1

theorem **Near.mul** {y z : AE D} (hy : Near p N y) (hz : Near p N z) : Near p N (y * z)

theorem **Near.conj** {y : AE D} (hy : Near p N y) : Near p N (cA D y)

theorem **sec_mem** {a : k} (ha : valuation k a ≤ 1) : p.sec a ∈ intRing D

theorem **e_mem** : p.e ∈ intRing D

theorem **mem_of_supp** {y : AE D} (h1 : cA D y * y = 1) (h2 : (p.e + cA D p.e) * (y - 1) = y - 1) (h3 : valuation k (p.pr y) = 1) : y ∈ intRing D

theorem **dvd_idem** (hN : valuation k (N : k) = 1) : Dvd N (p.e + cA D p.e)

theorem **lt_one_of_near** (hN2 : 2 < N) (hN : valuation k (N : k) = 1) {y : AE D} (h1 : cA D y * y = 1) (h2 : (p.e + cA D p.e) * (y - 1) = y - 1) (h3 : valuation k (p.pr y) = 1) {a : D.host.E} (ha : star a * a = 1) (h : Near p N (y * incA D a)) : valuation k (p.pr y - 1) < 1

theorem **isOpen_near** (hN : 0 < N) : IsOpen {y : AE D | y ∈ intRing D ∧ Dvd N ((1 - (p.e + cA D p.e)) * (y - 1)) ∧ valuation k (p.pr y - 1) < 1}

theorem **ent_mul** (g h : GA D S) : mt S (g * h) i i = mt S g i i * mt S h i i

theorem **ent_norm** (g : GA D S) : cA D (mt S g i i) * mt S g i i = 1

theorem **ent_inv** (g : GA D S) : mt S g⁻¹ i i = cA D (mt S g i i)

theorem **isOpen_ratNear** (hN : 0 < N) : IsOpen (ratNear p N i : Set (GA D S))

theorem **GQ_le_ratNear** : GQ S ≤ ratNear p N i

theorem **pr_ent** (β : Frame p S (Fin 1)) (g : GL (Fin 1) k) : p.pr (mt S (β.gl g) i i) = g.val 0 0

theorem **supp_ent** (β : Frame p S (Fin 1)) (g : GL (Fin 1) k) : (p.e + cA D p.e) * (mt S (β.gl g) i i - 1) = mt S (β.gl g) i i - 1

theorem **exists_tame_ramified** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S : HSp D), S.rank = 1 → ∀ β : Frame p S (Fin 1), 2 < Nat.card (IsLocalRing.ResidueField 𝒪[k]) → ∃ f : GA D S → ℂ, IsContAut S f ∧ (∀ g : GL (Fin 1) k, valuation k (g.val 0 0 - 1) < 1 → β.rep g f = f) ∧ ∃ g : GL (Fin 1) k, valuation k (g.val 0 0) = 1 ∧ β.rep g f ≠ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTermLoopOfStandingHypotheses

theorem **GlobalTheory.Core.Deg.exists_termLoop_of_standingHypotheses** : ∀ (D : D7) (large₂ : SizeData₂ D) (X : HidaSetting D) (Y : Full X) (hF : FinX Y) (hI : SIso D.s (sOf hF)) (u : D.what₂.Word) (C' Cl Cg cb Cu cT CT : ℝ), 29 ≤ C' → 72 ≤ Cl → Ties D → StandingHypotheses D large₂ C' Cl Cg cb Cu cT CT → ConcTie X Y → ∃ (W : WittDat X) (H : TieHyp W hI) (Am : AmpDat X) (hV : PvalForm W hI Am) (Z : LoopTie D large₂ (termTieOf W hI H Am hV) u u 1), TieAt hF hI large₂ u (termTieOf W hI H Am hV) Z ∧ ∀ t, Z.volR t = Formalism.L2.SG.LoopTerm.volRC hI t

theorem **tieOn_singleton_uncond** : ∀ c : ℝ, TieOn {c}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsThetaFunctionSchrodingerFunctionEq

theorem **schw_onto** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ φ : Mx D B.rank A.rank → ℂ, GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → ∃ f, IsThetaFunO ψ (dblSp A) B f ∧ sch ψ A B ν 𝒴 f = φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsThetaFunctionSchrodingerFunctionEqStatement

theorem **_root_.GlobalTheory.T.exists_thetaFunction_schrodingerFunction_eq** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ φ : Mx D B.rank A.rank → ℂ, GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → ∃ f, IsThetaFunO ψ (dblSp A) B f ∧ sch ψ A B ν 𝒴 f = φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsThetaLiftSpacesPositiveIndexWithLineNorms

theorem **star_eq_conj** (D : D7) (x : D.host.E) : star x = D.host.conj x

theorem **posDef_of_sig** {n : ℕ} (G : Matrix (Fin n) (Fin n) ℂ) (h : Formalism.UG.HasSignature G n 0) : G.PosDef

theorem **defSpace** {D : D7} (hX : ∀ j, j ≤ D.R.m → (D.sp.X j).IsPosDef) {j : ℕ} (hj : j ≤ D.R.m) : (D.sp.X j).SigOK ∧ (∀ τ, (D.sp.X j).sig τ = (D.R.N j, 0)) ∧ CoCpt (D.sp.X j)

theorem **spaces_of_index** : ∀ (D : D7) (u : D.what₂.Word), (∀ j, j ≤ D.R.m → (D.sp.X j).IsPosDef) → (∀ j ρ, j + 2 * ρ + 2 = D.R.m → ∃ (V : HSp D) (b : BesAt V (D.sp.X j) ρ), V.rank = D.R.rankY j ∧ V.SigOK ∧ (∀ τ, V.sig τ = sigYD D.R j) ∧ (star b.ζ ·ᵥ b.L.gram *ᵥ b.ζ) * (D.host.xi0 * star D.host.xi0) ^ (j / 2) = -(((D.mem u).A.T : ℕ) : D.host.E)) ∧ ∀ j ρ, j + 2 * ρ + 2 = D.R.m → ∃ (V : HSp D) (_ : JacAt V (D.sp.X (j + 1)) ρ), V.rank = D.R.rankY (j + 1) ∧ V.SigOK ∧ ∀ τ, V.sig τ = sigYD D.R (j + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsThetaLiftSpacesPositiveIndexWithLineNormsStatement

theorem **_root_.GlobalTheory.T.exists_thetaLiftSpaces_positiveIndex_with_lineNorms** : ∀ (D : D7) (u : D.what₂.Word), (∀ j, j ≤ D.R.m → (D.sp.X j).IsPosDef) → (∀ j ρ, j + 2 * ρ + 2 = D.R.m → ∃ (V : HSp D) (b : BesAt V (D.sp.X j) ρ), V.rank = D.R.rankY j ∧ V.SigOK ∧ (∀ τ, V.sig τ = sigYD D.R j) ∧ (star b.ζ ·ᵥ b.L.gram *ᵥ b.ζ) * (D.host.xi0 * star D.host.xi0) ^ (j / 2) = -(((D.mem u).A.T : ℕ) : D.host.E)) ∧ ∀ j ρ, j + 2 * ρ + 2 = D.R.m → ∃ (V : HSp D) (_ : JacAt V (D.sp.X (j + 1)) ρ), V.rank = D.R.rankY (j + 1) ∧ V.SigOK ∧ ∀ τ, V.sig τ = sigYD D.R (j + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsThresholdCriticalLevelLowerBound

theorem **num_bound_pow** (k m r e : ℕ) (B P ae c ka : ℝ) (hm : m = 2 * r + 2) (he : 2 * e + 40 ≤ m) (hk : 5 ≤ k) (hB : (k : ℝ) ^ 40 ≤ B) (hP : B ^ m ≤ P) (ha0 : 0 < ae) (ha : ae ≤ B ^ e) (hc : 1 / 8 ≤ c) (hka0 : 0 < ka) (hka : ka ≤ 4) : (k : ℝ) ^ (3 * r) ≤ c * (ka ^ (m - 1))⁻¹ * P / ae

theorem **le_pow_ceil** (b B u cb Cu : ℝ) (hcb : 0 < cb) (hb : 1 ≤ b) (hB : b ^ cb ≤ B) (hu : u ≤ b ^ Cu) : u ≤ B ^ ⌈max Cu 0 / cb⌉₊

theorem **tcrit_low_of_sizes** (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cb Cu cT CT C0 : ℝ) (hcb : 0 < cb) (hCl : 2 * ((⌈max Cu 0 / cb⌉₊ + 4 : ℕ) : ℝ) + 40 ≤ Cl) (hVV : ∀ τ, VV D τ = vand (lamSH D.R (D.hc τ))) : UnitaryVolumeMass.stTcritLow D large₂ C' Cl Cg cb Cu cT CT C0

theorem **GlobalTheory.CritLevelSize.tcritLow_sizes** : ∀ cb Cu : ℝ, 0 < cb → ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cT CT C0 : ℝ), thr ≤ Cl → (∀ τ, VV D τ = vand (lamSH D.R (D.hc τ))) → UnitaryVolumeMass.stTcritLow D large₂ C' Cl Cg cb Cu cT CT C0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsThresholdCriticalLevelLowerBoundStatement

theorem **_root_.GlobalTheory.T.exists_threshold_criticalLevel_lower_bound** : ∀ cb Cu : ℝ, 0 < cb → ∃ thr : ℝ, ∀ (D : D7) (large₂ : SizeData₂ D) (C' Cl Cg cT CT C0 : ℝ), thr ≤ Cl → (∀ τ, VV D τ = vand (lamSH D.R (D.hc τ))) → UnitaryVolumeMass.stTcritLow D large₂ C' Cl Cg cb Cu cT CT C0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopReadingBundle

theorem **isJ_smul** {D : D7} {u : D.what₂.Word} {S : D.what₂.Tau → SPar} {M : Formalism.IndefR.Mult.DSModel D.sp.X' S} {j : SmT M →ₗ[ℂ] (Formalism.L2.Forms.FrP D → ℂ)} (h : GlobalTheory.Core.Rk.Top.IsJ u M j) {a : ℂ} (ha : a ≠ 0) : GlobalTheory.Core.Rk.Top.IsJ u M (a • j) where

theorem **toMap_fin** (t : W.P.T) (x : W.P.A3.A.fam.amb.FR t) (g : GA (I.rec7 l Z) (I.rec7 l Z).sp.X') (hg : g ∈ GfC (I.rec7 l Z).sp.X') (v : SmT (pk W).MX) : (pk W).cX.toMap (pk W).prX (((pk W).HX t).e ((pk W).jX t x)) v g = (pk W).prX (pkAt W t x g) v

theorem **exists_j** (hg : Formalism.L2.Forms.gramP (I.rec7 l Z) = 1) (u : B.Word) (cv : Val B →ₛₗ[B.embLL] (pk W).cX.W) (hE : ValEqv W cv) (hP : PkVal W cv) (hi : ∀ (t : W.P.T) (φ : Frm I (W.P.I.tup t)) (x : W.P.A3.A.fam.amb.FR t), IsRes φ x → (∀ g y, HasMat g y → IsUni B y → pkAt W t x g = 0) → ∀ d, val φ d = 0) : ∃ j, IsJ u (pk W).MX j ∧ ∀ (t : W.P.T) (φ : Frm I (W.P.I.tup t)) (x : W.P.A3.A.fam.amb.FR t), IsRes φ x → ∀ d v, (pk W).prX (cv (val φ d)) v = pairVal B (j v) (val φ d)

theorem **reads_pk** (hOk : I.Ok) (hC : I.StdC l Z W.R.T.cols) (hW : GlobalTheory.AdForm.TwStd W) (u : B.Word) (cv : Val B →ₛₗ[B.embLL] (pk W).cX.W) (hP : PkVal W cv) (j : SmT (pk W).MX →ₗ[ℂ] (Formalism.L2.Forms.FrP (I.rec7 l Z) → ℂ)) (t : W.P.T) (φ : Frm I (W.P.I.tup t)) (x : W.P.A3.A.fam.amb.FR t) (hx : IsRes φ x) (hV : ∀ d v, (pk W).prX (cv (val φ d)) v = pairVal B (j v) (val φ d)) (G : (I.rec7 l Z).what₂.Amb (W.P.I.tup t)) (hG : RecVal u (W.P.I.tup t) φ G) : Reads W u j 1 (W.P.I.tup t) G ((pk W).cX.toMap (pk W).prX (((pk W).HX t).e ((pk W).jX t x)))

theorem **levG_one** {D : D7} {u : D.what₂.Word} {C : ColumnChain D.R D.what₂.Tau (D.hcW u)} {A : ColumnsAtT D.R C} {Pm : PeriodMap D.R C} {M : MemberForms D.what₂.LL D.what₂.embLL D.what₂.places D.what₂.Tau (D.hcW u) C A Pm} {bad} (K : Formalism.L2.Forms.ClassesAtPrimes M bad) {J : Set (GA D D.sp.X')} {χ : GA D D.sp.X' → ℂ} (h1 : ∀ g, g ∈ J ↔ ∃ κ : K.lev.U, GlobalTheory.Core.Rk.Top.HasMat g κ.1) (h2 : ∀ g (κ : K.lev.U), GlobalTheory.Core.Rk.Top.HasMat g κ.1 → χ g = star ((K.lev.chi κ : ℂˣ) : ℂ)) : LevG K 1 J χ

theorem **topReadAt_of_bWit** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → Nonempty (Bl.RXs q.x.what₂) → ∀ u, ∃ j, TopReadAt (packB b) u j 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopReadingBundleStatement

theorem **_root_.GlobalTheory.T.exists_top_reading_bundle** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → Nonempty (Bl.RXs q.x.what₂) → ∀ u, ∃ j, TopReadAt (packB b) u j 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopReadingOfSg

theorem **GlobalTheory.SeeSaw.exists_isJ_of_par** {D : D7} {u : D.what₂.Word} (hsg : ∀ (i : Fin (D.R.N D.R.m - 1)) τ, (((D.fm u).sg τ).w i : ℚ) = sgHC D.R (D.hcW u τ) i - rhoVec (D.R.N D.R.m - 1) i) {par} (M : DSModel D.sp.X' par) (hpar : par = parB (Ptw D) D.R.m) : ∃ j, GlobalTheory.Core.Rk.Top.IsJ u M j

theorem **GlobalTheory.SeeSaw.sg_trueColumns** {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (i : Fin ((I.rec7 l Z).R.N (I.rec7 l Z).R.m - 1)) (τ) : ((((I.rec7 l Z).fm u).sg τ).w i : ℚ) = sgHC (I.rec7 l Z).R ((I.rec7 l Z).hcW u τ) i - rhoVec ((I.rec7 l Z).R.N (I.rec7 l Z).R.m - 1) i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopValueWitness

theorem **matEq_of_hasMat** {D : D7} {h : GA D D.sp.X'} {y} (hm : HasMat h y) : GlobalTheory.Mem.R3.MatEq (R3.matIn D.sp.X' 1 ((finHom D.sp.X').rangeRestrict h)) y

theorem **datVal_rows** (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Top.Fr B.toPre) (T : TwDat B) (w : Bl.ColW B f T) (Z : w.Lt) (Wt : Set B.places.Pl) (p0 : BaseLevelRowSide (w.D Z)) (t : Bl.RowsT K w Z Wt p0) (hs : t.r.z.Std) : DatVal w.I w.l Z (aK

theorem **datVal_twArgs** (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Top.Fr B.toPre) (T : TwDat B) (w : Bl.ColW B f T) (Z : w.Lt) (Wt : Set B.places.Pl) (L : SizeData₂ (w.st Z).toD7) (hT : GatTies (w.st Z).toD7) (Gi : G.Inst (w.st Z).toD7 L) (Gl : G.LatInst (w.st Z).toD7) (Kc : G.Consts) (Yd : DivX (w.st Z).toD7) (cX : ExtraC) (ex : Extra (w.st Z).toD7 L hT Gi Gl Kc Yd cX) (t : Bl.RowsT K w Z Wt ex.p0) (a : R3.TwArgs ex t) (hs : t.r.z.Std) : DatVal w.I w.l Z (aK

theorem **pkVal_of_bWit** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → q.x.what₂.r.z.Std → ∃ cv, GlobalTheory.TopVal.ValEqv (I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopValueWitnessStatement

theorem **_root_.GlobalTheory.T.exists_top_value_witness** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → q.x.what₂.r.z.Std → ∃ cv, GlobalTheory.TopVal.ValEqv (I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopologicalIsoParahoricChartSplitPlace

theorem **splitProj_projEmb** {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) (x : Fv D.host.F v ⊗[D.host.F] D.host.E) : GlobalTheory.AdelicU.splitProj (projEmb φ) x = φ x

theorem **reindexGL_val** {ι κ : Type} [Fintype ι] [DecidableEq ι] [Fintype κ] [DecidableEq κ] (K : Type) [Field K] [TopologicalSpace K] [IsTopologicalRing K] (ε : ι ≃ κ) (x : GL κ K) : (reindexGL K ε x).1 = x.1.submatrix ε ε

theorem **grpCo_eq** {v : HeightOneSpectrum (𝒪 D.host.F)} (φ : SplitProj D v) (V : HSp D) (hJ : IsHermND D.sp.cm.sig V.gram) (B : GL (Fin V.rank) (Fv D.host.F v)) (g : GA D V) : grpCo φ V B g = GlobalTheory.AdelicU.splitEquivIn (GlobalTheory.AdelicU.isCMExt_host D) hJ (projEmb φ) B (GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g)

theorem **exists_equiv_emb** : ∀ (D : D7) (s : PiLocalization₂ D) (k : ℕ) (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v) (ι : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) →* GA D s.V), IsSelfDualC s.V s.Λ → Formalism.L2.DIV.IsFrame s.β s.Λ → IsChart s k φ ι → ∃ e : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) ≃ₜ* GlobalTheory.FinZeta.Gv s.V v, ∀ y, ι y = GlobalTheory.AdelicU.emb D.sp.cm.sig s.V.gram v (e y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTopologicalIsoParahoricChartSplitPlaceStatement

theorem **_root_.GlobalTheory.T.exists_topological_iso_parahoric_chart_split_place** : ∀ (D : D7) (s : PiLocalization₂ D) (k : ℕ) (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v) (ι : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) →* GA D s.V), IsSelfDualC s.V

s.Λ → Formalism.L2.DIV.IsFrame s.β s.Λ → IsChart s k φ ι → ∃ e : GL (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) ≃ₜ* GlobalTheory.FinZeta.Gv s.V v, ∀ y, ι y = GlobalTheory.AdelicU.emb D.sp.cm.sig s.V.gram v (e y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTraceZeroLocalNormMulIntegralUnit

theorem **exists_norm_eq** {R : Type} [CommRing R] (c : R →+* R) (a : R) (ha : c a = -a) (B B' : Ideal R) (hsup : B ⊔ B' = ⊤) (h1 : ∀ x ∈ B, c x ∈ B') (h2 : ∀ y ∈ B', c y ∈ B) (hP : ∀ P : Ideal R, P.IsPrime → a ∈ P → B ≤ P ∨ B' ≤ P) (hcc : ∀ x, c (c x) = x) : ∃ α η : R, α * c α = a * η ∧ IsCoprime η a

theorem **inv_natCast_mem** (v : HeightOneSpectrum (𝒪 F)) (n : ℕ) (hn : (n : 𝒪 F) ∉ v.asIdeal) : algebraMap F (Fv F v) (n : F)⁻¹ ∈ v.adicCompletionIntegers F

theorem **inv_mem_intLoc** (v : HeightOneSpectrum (𝒪 F)) (n : ℕ) (hn : (n : 𝒪 F) ∉ v.asIdeal) [CharZero E] (x y : 𝒪 E) (h : x * y = n) : (1 : Fv F v) ⊗ₜ[F] (x : E)⁻¹ ∈ intLoc F E v

theorem **evFin_idemAt** (q : ℕ) (v : HeightOneSpectrum (𝒪 F)) : evFin F v (idemAt F q) = if (q : 𝒪 F) ∈ v.asIdeal then 1 else 0

theorem **idemAt_mul** (q : ℕ) : idemAt F q * idemAt F q = idemAt F q

theorem **glue_mul** (q : ℕ) (x y x' y' : E) : (idemAt F q ⊗ₜ[F] x + (1 - idemAt F q) ⊗ₜ[F] y) * (idemAt F q ⊗ₜ[F] x' + (1 - idemAt F q) ⊗ₜ[F] y') = idemAt F q ⊗ₜ[F] (x * x') + (1 - idemAt F q) ⊗ₜ[F] (y * y')

theorem **glue_self** (q : ℕ) (x : E) : idemAt F q ⊗ₜ[F] x + (1 - idemAt F q) ⊗ₜ[F] x = (1 : AF F) ⊗ₜ[F] x

theorem **glue_mem** (q : ℕ) (x y : E) (v : HeightOneSpectrum (𝒪 F)) (h1 : (q : 𝒪 F) ∈ v.asIdeal → (1 : Fv F v) ⊗ₜ[F] x ∈ intLoc F E v) (h2 : (q : 𝒪 F) ∉ v.asIdeal → (1 : Fv F v) ⊗ₜ[F] y ∈ intLoc F E v) : compFin F E v (idemAt F q ⊗ₜ[F] x + (1 - idemAt F q) ⊗ₜ[F] y) ∈ intLoc F E v

theorem **exists_idele** (σ : E →ₐ[F] E) (q : ℕ) (hq : 1 < q) (a α η : 𝒪 E) (ha : a ^ 2 = -(q : 𝒪 E)) (hα : (α : E) * σ α = a * η) (hcop : IsCoprime η a) (hu : ¬ IsUnit a) : ∃ u ε : (AF F ⊗[F] E)ˣ, (ε : AF F ⊗[F] E) * u * Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF F) σ u = (1 : AF F) ⊗ₜ[F] (a : E) ∧ (∀ v, compFin F E v ε ∈ intLoc F E v) ∧ ∀ v, compFin F E v (ε⁻¹ : (AF F ⊗[F] E)ˣ) ∈ intLoc F E v

theorem **different_le** (D : D7) (a : 𝒪 D.host.E) (ha : a ^ 2 = -(D.host.q : 𝒪 D.host.E)) (P : Ideal (𝒪 D.host.E)) (hp : P.IsPrime) (haP : a ∈ P) : differentIdeal (𝒪 D.host.K) (𝒪 D.host.E) ≤ P

theorem **exists_skew_norm_mul_unit** : ∀ D : D7, ∃ μ : D.host.E, IsSkew μ ∧ ∃ u ε : (AE D)ˣ, (ε : AE D) * u * cA D u = incA D μ ∧ IntA (ε : AE D) ∧ IntA ((ε⁻¹ : (AE D)ˣ) : AE D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTraceZeroLocalNormMulIntegralUnitStatement

theorem **_root_.GlobalTheory.T.exists_traceZero_localNorm_mul_integralUnit** : ∀ D : D7, ∃ μ : D.host.E, IsSkew μ ∧ ∃ u ε : (AE D)ˣ, (ε : AE D) * u * cA D u = incA D μ ∧ IntA (ε : AE D) ∧ IntA ((ε⁻¹ : (AE D)ˣ) : AE D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTranslatesOfValuesFrames

theorem **isTr_unique** {ι : Type} [Fintype ι] {n : ℕ} {G : ι → Matrix (Fin n) (Fin n) ℂ} {g : ι → Matrix (Fin n) (Fin n) ℂ} {y₁ y₂ y' : Frames G} (h₁ : IsTr g y₁ y') (h₂ : IsTr g y₂ y') : y₁.1 = y₂.1

theorem **jOf_exists** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks), IsInfFam R.θ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), ∃ J, JOf R j J

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTranslatesOfValuesFramesStatement

theorem **_root_.GlobalTheory.T.exists_translates_of_values_frames** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks), IsInfFam R.θ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), ∃ J, JOf R j J

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTransvectionFixingFirstVectorsSum

theorem **form_mulVec** (S : HSp D) {n : ℕ} (y : VA S) (M : Mx D S.rank n) (c : Fin n → AE D) : form S y (M *ᵥ c) = (fun k => cA D (c k)) ·ᵥ ((adj M · gA S) *ᵥ y)

theorem **mul_apply_col** {m n k : ℕ} (g : Mx D m n) (M : Mx D n k) (r : Fin m) (c : Fin k) : (g · M) r c = (g *ᵥ (M *ᵥ Pi.single c 1)) r

theorem **col_apply** {m n : ℕ} (M : Mx D m n) (c : Fin n) (r : Fin m) : (M *ᵥ Pi.single c 1) r = M r c

theorem **submatrix_col** {m n : ℕ} (M : Mx D m n) (i : Fin n) (c : Fin 1) : (M.submatrix id fun _ : Fin 1 => i) *ᵥ Pi.single c 1 = M *ᵥ Pi.single i 1

theorem **smul_vecMulVec_single** {ρ : ℕ} (s : AE D) (x : Fin ρ → AE D) (J k : Fin ρ) : (s • vecMulVec x (Pi.single J 1)) *ᵥ Pi.single k 1 = (s * (Pi.single k 1 : Fin ρ → AE D) J) • x

theorem **upSum_smul_vecMulVec** {ρ : ℕ} (s : AE D) (x : Fin ρ → AE D) (j J : Fin ρ) (hJ : J.1 = j.1 + 1) : GlobalTheory.MixPer.upSum (s • vecMulVec x (Pi.single J 1)) = s * x j

theorem **form_jY_single** (y : VA V) (J : Fin ρ) (s : AE D) : form V y (d.jY *ᵥ Pi.single J (cA D s)) = s * (d.pX *ᵥ y) J

theorem **form_jX** (y : VA V) (b : Fin ρ → AE D) : form V y (d.jX *ᵥ b) = (fun k => cA D (b k)) ·ᵥ (d.pY *ᵥ y)

theorem **transv_fix** {w v : VA V} {u : GA D V} (hu : IsTransv V w v u) {y : VA V} (hv : form V y v = 0) (hw : form V y w = 0) : mt V u *ᵥ y = y

theorem **pX_jX_single** (k : Fin ρ) : d.pX *ᵥ (d.jX *ᵥ Pi.single k 1) = Pi.single k 1

theorem **pY_jX_vec** (x : Fin ρ → AE D) : d.pY *ᵥ (d.jX *ᵥ x) = 0

theorem **isotropicProjection_anisotropic_vec_eq_zero** (x : Fin V₀.rank → AE D) : d.pX *ᵥ (d.j0 *ᵥ x) = 0

theorem **dualProjection_anisotropic_vec_eq_zero** (x : Fin V₀.rank → AE D) : d.pY *ᵥ (d.j0 *ᵥ x) = 0

theorem **transv_apply** {w v : VA V} {u : GA D V} (hu : IsTransv V w v u) (y : VA V) : mt V u *ᵥ y = transv V w v y

theorem **mem_fixer_of_ne** (hu : IsTransv V w v u) (fv : ∀ y : VA V, form V y v = s * (d.pX *ᵥ y) J) (fw : ∀ y : VA V, form V y w = c ·ᵥ (d.pY *ᵥ y)) {i : Fin ρ} (hiJ : i ≠ J) : u ∈ GlobalTheory.MixPer.fixer V (d.jX.submatrix id fun _ : Fin 1 => i)

theorem **transv_jX** (hu : IsTransv V w v u) (fv : ∀ y : VA V, form V y v = s * (d.pX *ᵥ y) J) (fw : ∀ y : VA V, form V y w = c ·ᵥ (d.pY *ᵥ y)) {x : Fin ρ → AE D} (hw : w = d.jX *ᵥ x) : mt V u · d.jX = d.jX + d.jX · (s • vecMulVec x (Pi.single J 1))

theorem **unitaryTransvection_on_anisotropic_part** (hu : IsTransv V w v u) (fv : ∀ y : VA V, form V y v = s * (d.pX *ᵥ y) J) (fw : ∀ y : VA V, form V y w = c ·ᵥ (d.pY *ᵥ y)) : mt V u · d.j0 = d.j0 + d.jX · (0 : Mx D ρ V₀.rank)

theorem **GlobalTheory.TransvMid.exists_mid** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (z : Fin V₀.rank → D.host.E) (j : Fin ρ) (a : Fin ρ → D.host.E), a j = 1 → (∀ i, j < i → a i = 0) → j.1 + 1 < ρ → ∀ s : AE D, ∃ v : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.VA V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a)) v = 0 ∧ (∀ i k : Fin ρ, k ≤ i → GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (Pi.single k 1)) v * GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a i = 0) ∧ ∀ u : GA D V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.IsTransv V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTransvectionFixingFirstVectorsSumStatement

theorem **_root_.GlobalTheory.T.exists_transvection_fixing_first_vectors_sum** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (z : Fin V₀.rank → D.host.E) (j : Fin ρ) (a : Fin ρ → D.host.E), a j = 1 → (∀ i, j < i → a i = 0) → j.1 + 1 < ρ → ∀ s : AE D, ∃ v : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.VA V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a)) v = 0 ∧ (∀ i k : Fin ρ, k ≤ i → GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (Pi.single k 1)) v * GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a i = 0) ∧ ∀ u : GA D V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.IsTransv V (GlobalTheory.Indefin

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTransvectionFixingIsotropicLastSum

theorem **form_mulVec** (S : HSp D) {k l : ℕ} (M : Mx D S.rank k) (N : Mx D S.rank l) (x : Fin k → AE D) (y : Fin l → AE D) : form S (M *ᵥ x) (N *ᵥ y) = (fun i => cA D (y i)) ·ᵥ ((adj N · (gA S · M)) *ᵥ x)

theorem **conj_smul** {n : ℕ} (c : AE D) (y : Fin n → AE D) : (fun i => cA D ((c • y) i)) = cA D c • fun i => cA D (y i)

theorem **conj_ratV** {n : ℕ} (z : Fin n → D.host.E) : (fun i => cA D (ratV z i)) = ratV (star z)

theorem **ratV_dotProduct** {n : ℕ} (x y : Fin n → D.host.E) : ratV x ·ᵥ ratV y = incA D (x ·ᵥ y)

theorem **incA_inv_mul** {x : D.host.E} (hx : x ≠ 0) : incA D x⁻¹ * incA D x = 1

theorem **hermSpaceAdelicForm_isotropic_anisotropic_zero** (b : Fin ρ → AE D) (y : Fin V₀.rank → AE D) : form V (vecX d b) (d.j0 *ᵥ y) = 0

theorem **form_vecX_vecX** (b b' : Fin ρ → AE D) : form V (vecX d b) (vecX d b') = 0

theorem **hermSpaceAdelicForm_anisotropic_isotropic_zero** (y : Fin V₀.rank → AE D) (b : Fin ρ → AE D) : form V (d.j0 *ᵥ y) (vecX d b) = 0

theorem **hermSpaceAdelicForm_anisotropic_rational_smul** (z : Fin V₀.rank → D.host.E) (c : AE D) : form V (d.j0 *ᵥ ratV z) (d.j0 *ᵥ (c • ratV z)) = cA D c * incA D (star z ·ᵥ V₀.gram *ᵥ z)

theorem **mt_jX_of_isTransv** {w v : VA V} {u : GA D V} (hu : IsTransv V w v u) (hw : ∀ b, form V (vecX d b) w = 0) (hv : ∀ b, form V (vecX d b) v = 0) : mt V u · d.jX = d.jX

theorem **mt_of_isTransv** {S : HSp D} {w v y : VA S} {u : GA D S} (hu : IsTransv S w v u) (hy : form S y w = 0) : mt S u *ᵥ y = y + form S y v • w

theorem **simpleSum_of** {z : Fin V₀.rank → D.host.E} {u : GA D V} {c : Fin ρ → AE D} (s : AE D) (hX : mt V u · d.jX = d.jX) (hy : mt V u *ᵥ (d.j0 *ᵥ ratV z) = d.j0 *ᵥ ratV z + s • (d.jX *ᵥ c)) : simpleSum d z u = s * (ell D ρ ·ᵥ c)

theorem **mem_fixer_column** {u : GA D V} (hX : mt V u · d.jX = d.jX) (i : Fin ρ) : u ∈ fixer V (d.jX.submatrix id fun _ : Fin 1 => i)

theorem **ell_dotProduct_last** {j : Fin ρ} (hj : j.1 + 1 = ρ) {a : Fin ρ → D.host.E} (haj : a j = 1) : ell D ρ ·ᵥ ratVec a = 1

theorem **exists_transv_simpleSum_last** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (z : Fin V₀.rank → D.host.E), star z ·ᵥ V₀.gram *ᵥ z ≠ 0 → ∀ (j : Fin ρ) (a : Fin ρ → D.host.E), a j = 1 → (∀ i, j < i → a i = 0) → j.1 + 1 = ρ → ∀ s : AE D, ∃ v : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.VA V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a)) v = 0 ∧ (∀ i k : Fin ρ, k ≤ i → GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (Pi.single k 1)) v * GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a i = 0) ∧ ∀ u : GA D V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.IsTransv V (GlobalTheory.IndefiniteRow

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTransvectionFixingIsotropicLastSumStatement

theorem **_root_.GlobalTheory.T.exists_transvection_fixing_isotropic_last_sum** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (z : Fin V₀.rank → D.host.E), star z ·ᵥ V₀.gram *ᵥ z ≠ 0 → ∀ (j : Fin ρ) (a : Fin ρ → D.host.E), a j = 1 → (∀ i, j < i → a i = 0) → j.1 + 1 = ρ → ∀ s : AE D, ∃ v : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.VA V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a)) v = 0 ∧ (∀ i k : Fin ρ, k ≤ i → GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form V (GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.vecX d (Pi.single k 1)) v * GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.ratVec a i = 0) ∧ ∀ u : GA D V, GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTrivialCharLevelEltRationalAtTwistPrimes

theorem **adAt_one_apply** (P : HeightOneSpectrum (𝒪 E)) {n : ℕ} (r s : Fin n) : adAt F P ((1 : Matrix (Fin n) (Fin n) (AfE F E)) r s) = (1 : Matrix (Fin n) (Fin n) (P.adicCompletion E)) r s

theorem **conj_bound** (hc : CMData F E) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : J = 1) {u : GL (Fin n) (AfE F E)} (hu : u ∈ UPts hc.sig J (FiniteAdeleRing (𝒪 F) F)) (P : HeightOneSpectrum (𝒪 E)) (ε : WithZero (Multiplicative ℤ)) (hε : ε < 1) (hP : ∀ r s, Valued.v (adAt F P (u.1 r s) - (1 : Matrix (Fin n) (Fin n) (P.adicCompletion E)) r s) ≤ ε) (r s : Fin n) : Valued.v (adAt F (conjPrime₂ P) (u.1 r s) - (1 : Matrix (Fin n) (Fin n) ((conjPrime₂ P).adicCompletion E)) r s) ≤ ε

theorem **map_prod_cases** {G M κ : Type} [Monoid G] [Monoid M] (f : κ → G) (c : G →* M) (hpair : ∀ P P' : κ, P ≠ P' → c (f P) = 1 ∨ c (f P') = 1) : ∀ L : List κ, L.Nodup → c (L.map f).prod = 1 ∨ ∃ P ∈ L, c (L.map f).prod = c (f P)

theorem **exists_level_elt_all** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) (T i : ℕ) (hi : i ≤ B.R.m) (bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))) (S : StepLevel (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) (gramN (I.rec7 l Z) i hi) T (I.rec7 l Z).R.f bad), GlobalTheory.Mem.Col2.AdT.ChiFin I l Z S → ∀ (N : ℕ) (g : TC.GA B (B.R.N i)), ∃ z : S.U, S.chi z = 1 ∧ (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P (((g ∗ (show TC.GA B (B.R.N i) from z.1) : TC.GA B (B.R.N i)) : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) ∧ ∀ (P : HeightOneSpectrum (𝒪 B.host.E)) (r s : Fin (B.R.N i)), Valued.v (adAt B.host.F P (((show TC.GA B (B.R.N i) from z.1) : Matrix (Fin (B.R.N i)) (Fin (B.R.N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTrivialCharLevelEltRationalAtTwistPrimesStatement

theorem **_root_.GlobalTheory.T.exists_trivialCharLevel_elt_rational_at_twistPrimes** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) (T i : ℕ) (hi : i ≤ B.R.m) (bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))) (S : StepLevel (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) (gramN (I.rec7 l Z) i hi) T (I.rec7 l Z).R.f bad), GlobalTheory.Mem.Col2.AdT.ChiFin I l Z S → ∀ (N : ℕ) (g : TC.GA B (B.R.N i)), ∃ z : S.U, S.chi z = 1 ∧ (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P (((g ∗ (show TC.GA B (B.R.N i) from z.1) : TC.GA B (B.R.N i)) : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) ∧ ∀ (P : HeightOneSpectrum (𝒪 B.host.E)) (r s : Fin (B.R.N i)), Valued.v (adAt B.host.F P (((show TC.GA B (B.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTupleAdelicDataStandardBasePoints

theorem **exists_twDat_baseStd** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι), ∃ (Ch : ℕ → Type) (_ : ∀ i, Fintype (Ch i)) (A : TwDat I l Z u a Ch), A.Is ∧ Nonempty (Unique (Ch B.R.m)) ∧ (∀ i, Nonempty (Idx I.sel (I.tup a) (I.idx a) i) → Nonempty (Ch i)) ∧ GlobalTheory.QTw.BaseCommon A ∧ (∀ i, i < B.R.m → ∀ β, A.kb i β = TC.kStd X u (B.R.N i)) ∧ ∀ i, i ≤ B.R.m → ∀ β (v : HeightOneSpectrum (𝒪 B.host.F)), ((B.T u : ℕ) : 𝒪 B.host.F) ∉ v.asIdeal → (i < B.R.m ∨ ¬ I.UnderQ a v) → Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTupleAdelicDataStandardBasePointsStatement

theorem **_root_.GlobalTheory.T.exists_tupleAdelicData_standardBasePoints** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι), ∃ (Ch : ℕ → Type) (_ : ∀ i, Fintype (Ch i)) (A : TwDat I l Z u a Ch), A.Is ∧ Nonempty (Unique (Ch B.R.m)) ∧ (∀ i, Nonempty (Idx I.sel (I.tup a) (I.idx a) i) → Nonempty (Ch i)) ∧ GlobalTheory.QTw.BaseCommon A ∧ (∀ i, i < B.R.m → ∀ β, A.kb i β = TC.kStd X u (B.R.N i)) ∧ ∀ i, i ≤ B.R.m → ∀ β (v : HeightOneSpectrum (𝒪 B.host.F)), ((B.T u : ℕ) : 𝒪 B.host.F) ∉ v.asIdeal → (i < B.R.m ∨ ¬ I.UnderQ a v) → Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTupleChainAdelicData

theorem **kStd_at_SQ** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (u : B.Word) (n : ℕ) (P : HeightOneSpectrum (𝒪 B.host.E)) (hP : P ∈ I.SQ) : (kStd X u n).1.map (adAt B.host.F P) = 1

theorem **kStd_pt** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (u : B.Word) (a : I.ι) (i : ℕ) : ∃ q : ChPt (Pr

theorem **stepCh_zero** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) : StepCh I l Z u a 0

theorem **topBaseOk** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) {C : Type} (chOf : (I.stp u a B.R.m).Cl → C) (Lb : C → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)) (U : C → Subgroup (TC.GA B (B.R.N B.R.m))) (chi : ∀ β, U β →∗ ℂˣ) (y : (I.stp u a B.R.m).Cl → TC.GA B (B.R.N B.R.m)) (st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E) (k : C → TC.GA B (B.R.N B.R.m)) (hR : I.Reads u a B.R.m chOf Lb U chi y st k) (htop : ∀ β, I.TopPt l Z u a (k β)) (β : C) : BaseOk I u a B.R.m (k β)

theorem **stepCh_mid** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) (h1 : 1 ≤ i) (h2 : i < B.R.m) : StepCh I l Z u a i

theorem **stepCh_top** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) : StepCh I l Z u a B.R.m

theorem **stepCh_above** {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) (hi : B.R.m < i) : StepCh I l Z u a i

theorem **stepCh_all** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) : StepCh I l Z u a i

theorem **liftKA** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (i : ℕ) (hi : i ≤ B.R.m) (k : TC.GA B (B.R.N i)) (hk : ∃ p : Formalism.L2.Col.DecoF B.fields.E (B.R.N i) ∅ ∅ ∅, TC.PtOf k p ∧ p.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i)) : ∃ kA : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i), kA ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) kA = k.1

theorem **levData** {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (i : ℕ) (hi : i ≤ B.R.m) {T : ℕ} {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {cl : Type} {stab : cl → Type} [∀ c, Group (stab c)] {lat : cl → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)} (K : LevelChart (I.rec7 l Z) T bad i hi stab lat) : ∃ (U : Subgroup (GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i))) (chi : U →∗ ℂˣ) (h : cl → GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i)), (∀ g, g ∈ U ↔ g ∈ GfC _ ∧ ∃ κ : K.lev.U, κ.1.1 = finM _ (rankN (I.rec7 l Z) i hi) g) ∧ (∀ (g : U) (κ : K.lev.U), κ.1.1 = finM _ (rankN (I.rec7 l Z) i hi) g.1 → chi g = K.lev.chi κ) ∧ (∀ c, h c ∈ GfC _ ∧ finM _ (rankN (I.rec7 l Z) i hi) (h c) = (K.y c).1.1)

theorem **pointOf** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i ≤ B.R.m) {C : Type} (chOf : (I.stp u a i).Cl → C) (Lb : C → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)) (U : C → Subgroup (TC.GA B (B.R.N i))) (chi : ∀ β, U β →∗ ℂˣ) (y : (I.stp u a i).Cl → TC.GA B (B.R.N i)) (st : ∀ c, (I.stp u a i).Stab c → GL (Fin (B.R.N i)) B.host.E) (k : C → TC.GA B (B.R.N i)) (hR : I.Reads u a i chOf Lb U chi y st k) (β : C) (hb : BaseOk I u a i (k β)) : ∃ (b0 : Idx I.sel (I.tup a) (I.idx a) i) (p0 : PtAt B.R I.sel (I.tup a) (I.cst a) u i), TC.PtOf (k β) (extTo I.SQ p0.1.1) ∧ p0.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N i) ∧ ∀ c, chOf c = β → c.1 = b0 ∧ cosetIn ((I.stp u a i).rep c) = cosetIn p0

theorem **comp** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i ≤ B.R.m) {C : Type} {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} (chOf : (I.stp u a i).Cl → C) (K : ∀ β : C, LevelChart (I.rec7 l Z) (B.T u) bad i hi (fun c : {c // chOf c = β} => (I.chain l u a).stab i c.1) (fun c => latCrd (I.rec7 l Z) i hi (I.latAt u a i c.1))) (k : C → TC.GA B (B.R.N i)) (hR : I.Reads u a i chOf (fun β => (K β).lev.Lb) (fun β => (K β).lev.U) (fun β => (K β).lev.chi) (fun c => ((K (chOf c)).y ⟨c, rfl⟩).1) (fun c s => ((K (chOf c)).stabI ⟨c, rfl⟩ s).1) k) (hb : ∀ β, BaseOk I u a i (k

β)) : ∃ (kA : C → GA (I.rec7 l Z) ((I.rec7 l Z).sp.X i)) (xr : C → (AE (I.rec7 l Z))ˣ) (b0 : C → Idx I.sel (I.tup a) (I.idx a) i) (p0 : C → PtAt B.R I.sel (I.tup a) (I.cst a) u i) (U : C → Subgroup (GA (I.rec7 l Z) ((I.rec7 l

theorem **fullStep** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) : FullStep I l Z u a i

theorem **exists_chr** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) : ∃ chr : ℕ → ((AE (I.rec7 l Z))ˣ →* Circle), ∀ j : Fin B.R.m, ChrEq (chr ((j : ℕ) + 1)) (I.C.mu0 j * I.xi a)

theorem **exists_idem** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) : ∃ e : AE (I.rec7 l Z), IsIdemSum e (Formalism.L2.Col.Sall (Pr

theorem **GlobalTheory.AdForm.exists_twDat** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι), ∃ (Ch : ℕ → Type) (_ : ∀ i, Fintype (Ch i)) (A : TwDat I l Z u a Ch), A.Is ∧ Nonempty (Unique (Ch B.R.m)) ∧ ∀ i, i < B.R.m → ∀ β, A.kb i β = TC.kStd X u (B.R.N i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTupleChainAdelicDataStatement

theorem **_root_.GlobalTheory.T.exists_tupleChain_adelicData** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι), ∃ (Ch : ℕ → Type) (_ : ∀ i, Fintype (Ch i)) (A : TwDat I l Z u a Ch), A.Is ∧ Nonempty (Unique (Ch B.R.m)) ∧ ∀ i, i < B.R.m → ∀ β, A.kb i β = TC.kStd X u (B.R.N i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTwistIndexRepresentativeOfBounds

theorem **exists_rep_of_bounds** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTwistIndexRepresentativeOfBoundsStatement

theorem **_root_.GlobalTheory.T.exists_twistIndex_representative_of_bounds** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTwistRationalLevelElementsSameCharacters

theorem **mt_mul'** {D : D7} (V : HSp D) (g h : GA D V) : Formalism.L2.Theta.mt V (g * h) = Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V h

theorem **actO_actO'** {D : D7} (S T : HSp D) (g g' : GA D S) (h h' : GA D T) (x : WO S T) : actO S T g h (actO S T g' h' x) = actO S T (g * g') (h * h') x

theorem **isHalf_actO'** {D : D7} {S T : HSp D} {e : AE D} (g : GA D S) (h : GA D T) {u : WO S T} (hu : IsHalf e u) : IsHalf e (actO S T g h u)

theorem **approx** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ (z : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E), ∃ (κ₁ : A.U i α) (κ₁' : A.U (i + 1) γ) (x₁ y₁ : (AE (I.rec7 l Z))ˣ) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ₁.1 : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) ∧ (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N (i + 1))) (Fi

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsTwistRationalLevelElementsSameCharactersStatement

theorem **_root_.GlobalTheory.T.exists_twistRational_levelElements_same_characters** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ (z : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E), ∃ (κ₁ : A.U i α) (κ₁' : A.U (i + 1) γ) (x₁ y₁ : (AE (I.rec7 l Z))ˣ) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ₁.1 : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnifUnit

theorem **cA_tmul** (a : AF D.host.F) (x : D.host.E) : cA D (a ⊗ₜ[D.host.F] x) = a ⊗ₜ[D.host.F] star x

theorem **cA_cA** (z : AE D) : cA D (cA D z) = z

theorem **finPart_tmul** (a : AF D.host.F) (x : D.host.E) : finPart D (a ⊗ₜ[D.host.F] x) = sndA D.host.F a ⊗ₜ[D.host.F] x

theorem **approx_gen** {F A B E : Type} [Field F] [CharZero F] [CommRing A] [CommRing B] [CommRing E] [Algebra F A] [Algebra F B] [Algebra F E] (φ : A → B) (σ : E →+* E) (fp : A ⊗[F] E →+* B ⊗[F] E) (c : A ⊗[F] E →+* A ⊗[F] E) (hfp : ∀ a x, fp (a ⊗ₜ[F] x) = φ a ⊗ₜ[F] x) (h1 : φ 1 = 1) (hc : ∀ a x, c (a ⊗ₜ[F] x) = a ⊗ₜ[F] σ x) (O : Set B) (I : AddSubgroup (B ⊗[F] E)) (hI : ∀ b ∈ O, ∀ x : E, IsIntegral ℤ x → b ⊗ₜ[F] x ∈ I) (hE : ∀ x : E, ∃ m : ℤ, m ≠ 0 ∧ IsIntegral ℤ (m • x)) (N : ℕ) (hB : ∀ y : B, ∃ (f : F) (β : B), β ∈ O ∧ y = algebraMap F B f + N • β) (z : A ⊗[F] E) : ∃ x : E, ∃ a ∈ I, ∃ a' ∈ I, fp (z - 1 ⊗ₜ[F] x) = N * a ∧ fp (c z - c (1 ⊗ₜ[F] x)) = N * a'

theorem **cancel_gen** {F R : Type} [Field F] [CharZero F] [CommRing R] [Algebra F R] {N : ℕ} (hN : 0 < N) {a b : R} (h : (N : R) * a = N * b) : a = b

theorem **tmul_mem** {a : Af D.host.F} (ha : a ∈ intAf D.host.F) {x : D.host.E} (hx : IsIntegral ℤ x) : a ⊗ₜ[D.host.F] x ∈ intAE D

theorem **mul_mem_intAE** {a b : Af D.host.F ⊗[D.host.F] D.host.E} (ha : a ∈ intAE D) (hb : b ∈ intAE D) : a * b ∈ intAE D

theorem **approx** (N : ℕ) (hN : 0 < N) (z : AE D) : ∃ c : D.host.E, ∃ A ∈ intAE D, ∃ A' ∈ intAE D, finPart D (z - incA D c) = N * A ∧ finPart D (cA D z - incA D (star c)) = N * A'

theorem **val_le** (p : SplitPt D k) (N : ℕ) (hN : 0 < N) (z : AE D) (A : Af D.host.F ⊗[D.host.F] D.host.E) (hA : A ∈ intAE D) (h : finPart D z = N * A) : valuation k (p.pr z) ≤ valuation k (N : k)

theorem **exists_nat** (q : Unif k) : valuation k ((q.1 : 𝒪[k]) : k) < 1 ∧ ∃ N : ℕ, 0 < N ∧ valuation k (N : k) < valuation k ((q.1 : 𝒪[k]) : k)

theorem **exists_unif_unit** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k), ∃ c : D.host.E, valuation k (p.pr (incA D c)) = valuation k ((q.1 : 𝒪[k]) : k) ∧ valuation k (p.pr (incA D (star c))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniformIsotypicAverageRealCentre

theorem **exists_isotypic_uniform** : ∀ (D : D7), BiAct D → CenFock D → LiftSum D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k₀ : KerDat A S) (k₀' : KerDat S A'), k₀.Is ψ θ x y → k₀'.Is ψ θ y z → ∀ (χ₁ : GA D A → ℂ) (χ₃ : GA D S → ℂ) (C₁ : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L y.L z.L C₁ → IsBiSmooth C₁ → (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k₀.om (a, 1) (fun u => C₁ u v) = χ₁ a • fun u => C₁ u v) → (∀ b ∈ GinfC S, b ∈ Subgroup.center (GA D S) → ∀ v, k₀.om (1, b) (fun u => C₁ u v) = χ₃ b • fun u => C₁ u v) → ∃ (C₂ : WO A S → WO S A' → ℂ) (χ₂ : GA D S → ℂ), IsBiVac ψ A S A' x.L y.L z.L C₂ ∧ (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k₀.om (a, 1) (fun u => C₂ u v) = χ₁ a • fun u => C₂ u v) ∧ (∀ b ∈ GinfC S, b ∈ Subgroup.center (GA D S) → ∀ u, k₀'.om (b, 1) (C

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniformIsotypicAverageRealCentreStatement

theorem **_root_.GlobalTheory.T.exists_uniform_isotypic_average_real_centre** : ∀ (D : D7), BiAct D → CenFock D → LiftSum D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (A S A' : HSp D) (x : DefSp A) (y : DefSp S) (z : DefSp A'), x.Is → y.Is → z.Is → ∀ (k₀ : KerDat A S) (k₀' : KerDat S A'), k₀.Is ψ θ x y → k₀'.Is ψ θ y z → ∀ (χ₁ : GA D A → ℂ) (χ₃ : GA D S → ℂ) (C₁ : WO A S → WO S A' → ℂ), IsBiVac ψ A S A' x.L y.L z.L C₁ → IsBiSmooth C₁ → (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k₀.om (a, 1) (fun u => C₁ u v) = χ₁ a • fun u => C₁ u v) → (∀ b ∈ GinfC S, b ∈ Subgroup.center (GA D S) → ∀ v, k₀.om (1, b) (fun u => C₁ u v) = χ₃ b • fun u => C₁ u v) → ∃ (C₂ : WO A S → WO S A' → ℂ) (χ₂ : GA D S → ℂ), IsBiVac ψ A S A' x.L y.L z.L C₂ ∧ (∀ a ∈ GinfC A, a ∈ Subgroup.center (GA D A) → ∀ v, k₀.om (a, 1) (fun u => C₂ u v) = χ₁ a • fun u => C₂ u v) ∧ (∀ b ∈ GinfC S, b ∈ Subgro

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnimodularBasisSelfDualLatticeAtPlace

theorem **exists_basis_unimodular** : ∀ (D : D7) (S : HSp D) (v : Pl D) (ε : AE D) (Λ : AddSubgroup (Fin S.rank → AE D)), Tame D v → ε * ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) → IsSDAt S v Λ → ∃ β β' : Matrix (Fin S.rank) (Fin S.rank) (AE D), (∀ i j, β i j ∈ idAt D (· = v)) ∧ (∀ i j, β' i j ∈ idAt D (· = v)) ∧ β * β' = ε • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) ∧ β' * β = ε • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) ∧ (∀ y : Fin S.rank → AE D, (∀ i, y i ∈ idAt D (· = v)) → (y ∈ Λ ↔ ∀ i, (β' *ᵥ y) i ∈ intAt D (· = v))) ∧ (∀ i j, (adj β * (gA S * β)) i j ∈ intAt D (· = v)) ∧ (∀ i j, cA D ((adj β * (gA S * β)) i j) = (adj β * (gA S * β)) j i) ∧ ∃ H' : Matrix (Fin S.rank) (Fin S.rank) (AE D), (∀ i j, H' i j ∈ intAt D (· = v)) ∧ adj β * (gA S * β) * H' = ε • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnimodularBasisSelfDualLatticeAtPlaceStatement

theorem **_root_.GlobalTheory.T.exists_unimodular_basis_selfDual_lattice_at_place** : ∀ (D : D7) (S : HSp D) (v : Pl D) (ε : AE D) (Λ : AddSubgroup (Fin S.rank → AE D)), Tame D v → ε * ε = ε → cA D ε = ε → ε ∈ intAt D (· = v) → (∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) → IsSDAt S v Λ → ∃ β β' : Matrix (Fin S.rank) (Fin S.rank) (AE D), (∀ i j, β i j ∈ idAt D (· = v)) ∧ (∀ i j, β' i j ∈ idAt D (· = v)) ∧ β * β' = ε • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) ∧ β' * β = ε • (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) ∧ (∀ y : Fin S.rank → AE D, (∀ i, y i ∈ idAt D (· = v)) → (y ∈ Λ ↔ ∀ i, (β' *ᵥ y) i ∈ intAt D (· = v))) ∧ (∀ i j, (adj β * (gA S * β)) i j ∈ intAt D (· = v)) ∧ (∀ i j, cA D ((adj β * (gA S * β)) i j) = (adj β * (gA S * β)) j i) ∧ ∃ H' : Matrix (Fin S.rank) (Fin S.rank) (AE D), (∀ i j, H' i j ∈ intAt D (· = v)) ∧ adj β * (gA S * β) * H' = ε • (1 : Matrix (Fin S.rank) (Fin S.rank)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniqueAdelicMatrixRealComponentsFinitePart

theorem **exists_unique_of_comp_finPart** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ) (z : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ) (x₀ : Mx D m n), ∃! x : Mx D m n, (∀ τ, compAt θ τ x = z τ) ∧ x.map (finPart D) = x₀.map (finPart D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniqueAdelicMatrixRealComponentsFinitePartStatement

theorem **_root_.GlobalTheory.T.existsUnique_adelicMatrix_realComponents_finitePart** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ) (z : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ) (x₀ : Mx D m n), ∃! x : Mx D m n, (∀ τ, compAt θ τ x = z τ) ∧ x.map (finPart D) = x₀.map (finPart D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniqueCompletionMatrixOfRationalTwistAut

theorem **ordY_isOrd** (κ τ j c n : ℕ) (h : κ + j + j * τ + c = n) : IsOrd (yPos xPos (κ + j + j * τ)) (ordY κ τ j c n h)

theorem **isOrd_unique** {A : Type} {n : ℕ} (pos : A → ℕ) (e e' : A ≃ Fin n) (he : IsOrd pos e) (he' : IsOrd pos e') : e = e'

theorem **toLoc_eq** (ι : E →+* k) {A B : Type} {a b : ℕ} (e : A ≃ Fin a) (e' : B ≃ Fin b) (x : (Fin b → E) →ₗ[E] (Fin a → E)) : toLoc ι e e' x = ((LinearMap.toMatrix' x).submatrix e e').map ι

theorem **toLoc_comp** (ι : E →+* k) {A : Type} [Fintype A] {a : ℕ} (e : A ≃ Fin a) (x y : (Fin a → E) →ₗ[E] (Fin a → E)) : toLoc ι e e (x ∘ₗ y) = toLoc ι e e x * toLoc ι e e y

theorem **toLoc_id** (ι : E →+* k) {A : Type} [Fintype A] [DecidableEq A] {a : ℕ} (e : A ≃ Fin a) : toLoc ι e e (LinearMap.id : (Fin a → E) →ₗ[E] (Fin a → E)) = 1

theorem **existsUnique_isGL** : ∀ (X : Fields) (R : Ranks) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (j c : ℕ), c = cOf R j → ∀ γ : AutY X R j, ∃! g : GL (XIx (Fin R.k) (Fin R.t) (j) ⊕ Fin c) k, IsGL ι γ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniqueCompletionMatrixOfRationalTwistAutStatement

theorem **_root_.GlobalTheory.T.existsUnique_completionMatrix_of_rationalTwistAut** : ∀ (X : Fields) (R : Ranks) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (j c : ℕ), c = cOf R j → ∀ γ : AutY X R j, ∃! g : GL (XIx (Fin R.k) (Fin R.t) (j) ⊕ Fin c) k, IsGL ι γ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniqueSphericalDoublingSectionUnramified

theorem **gen_wd** (hK : ∀ p : P, (p : G) ∈ K → f p = 1) (p p' : P) (k k' : G) (hk : k ∈ K) (hk' : k' ∈ K) (e : (p : G) * k = p' * k') : f p = f p'

theorem **genFun_mul** (hK : ∀ p : P, (p : G) ∈ K → f p = 1) (p : P) {k : G} (hk : k ∈ K) : genFun P K f hI (p * k) = f p

theorem **genFun_right** (hK : ∀ p : P, (p : G) ∈ K → f p = 1) (g : G) {k : G} (hk : k ∈ K) : genFun P K f hI (g * k) = genFun P K f hI g

theorem **genFun_left** (hK : ∀ p : P, (p : G) ∈ K → f p = 1) (p : P) (g : G) : genFun P K f hI (p * g) = f p * genFun P K f hI g

theorem **genFun_one** (hK : ∀ p : P, (p : G) ∈ K → f p = 1) : genFun P K f hI 1 = 1

theorem **genFun_unique** (Ψ : G → ℂ) (hΨ : ∀ (p : P) g, Ψ (p * g) = f p * Ψ g) (hΨK : ∀ k ∈ K, ∀ g, Ψ (g * k) = Ψ g) (g : G) : Ψ g = Ψ 1 * genFun P K f hI g

theorem **continuous_genFun** [TopologicalSpace G] [IsTopologicalGroup G] (hK : ∀ p : P, (p : G) ∈ K → f p = 1) (ho : IsOpen (K : Set G)) : Continuous (genFun P K f hI)

theorem **gen_exists** [TopologicalSpace G] [IsTopologicalGroup G] (P K : Subgroup G) (f : P →* ℂ) (hI : ∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) (hK : ∀ p : P, (p : G) ∈ K → f p = 1) (ho : IsOpen (K : Set G)) : ∃ φ : G → ℂ, (∀ (p : P) g, φ (p * g) = f p * φ g) ∧ Continuous φ ∧ (∀ g, ∀ k ∈ K, φ (g * k) = φ g) ∧ φ 1 = 1 ∧ ∀ Ψ : G → ℂ, (∀ (p : P) g, Ψ (p * g) = f p * Ψ g) → (∀ k ∈ K, ∀ g, Ψ (g * k) = Ψ g) → ∀ g, Ψ g = Ψ 1 * φ g

theorem **exists_isSphSecAt_unique** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ v : FinPl D, (∀ w : HeightOneSpectrum (𝒪 D.host.E), w.under (𝒪 D.host.F) = v → GlobalTheory.LFun.UnrAt μ w) → ∀ s : ℂ, ∃ φ, IsSphSecAt A L v μ s φ ∧ ∀ Ψ : Gv (dblSp A) v → ℂ, (∀ (p : siegelH (Fv D.host.F v) D.sp.cm.sig (dblSp A).gram) g, Ψ (p * g) = charAt μ v (aH _ D.sp.cm.sig (dblSp A).gram p) * npow (absEv D.host.F D.host.E v (aH _ D.sp.cm.sig (dblSp A).gram p)) (s + (A.rank : ℂ) / 2) * Ψ g) → (∀ k ∈ KFin D.sp.cm.sig (dblSp A).gram L v, ∀ g, Ψ (g * k) = Ψ g) → ∀ g, Ψ g = Ψ 1 * φ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUniqueSphericalDoublingSectionUnramifiedStatement

theorem **_root_.GlobalTheory.T.exists_unique_spherical_doubling_section_unramified** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ v : FinPl D, (∀ w : HeightOneSpectrum (𝒪 D.host.E), w.under (𝒪 D.host.F) = v → GlobalTheory.LFun.UnrAt μ w) → ∀ s : ℂ, ∃ φ, IsSphSecAt A L v μ s φ ∧ ∀ Ψ : Gv (dblSp A) v → ℂ, (∀ (p : siegelH (Fv D.host.F v) D.sp.cm.sig (dblSp A).gram) g, Ψ (p * g) = charAt μ v (aH _ D.sp.cm.sig (dblSp A).gram p) * npow (absEv D.host.F D.host.E v (aH _ D.sp.cm.sig (dblSp A).gram p)) (s + (A.rank : ℂ) / 2) * Ψ g) → (∀ k ∈ KFin D.sp.cm.sig (dblSp A).gram L v, ∀ g, Ψ (g * k) = Ψ g) → ∀ g, Ψ g = Ψ 1 * φ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitMultipliersLineInMovedLatticeSplit

theorem **exists_dominant** {k : ℕ} (f : Fin k → Fv F v) (hf : ∃ j, f j ≠ 0) : ∃ j, f j ≠ 0 ∧ ∀ c : Fv F v, (∀ i, c * f i ∈ v.adicCompletionIntegers F) ↔ c * f j ∈ v.adicCompletionIntegers F

theorem **exists_unit_gen** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) {k : ℕ} {u l : Fin k → Fv F v ⊗[F] E} (hl : ∑ j, l j * u j = 1) : ∃ α : (Fv F v ⊗[F] E)ˣ, ∀ c : Fv F v ⊗[F] E, (∀ j, c * u j ∈ intLoc F E v) ↔ ((α⁻¹ : (Fv F v ⊗[F] E)ˣ) : Fv F v ⊗[F] E) * c ∈ intLoc F E v

theorem **exists_unit_mult_latLoc** : ∀ (D : D7) (v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F)), Nonempty (D.host.E →ₐ[D.host.F] Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) → ∀ (n : ℕ) (L : Submodule (𝒪 D.host.E) (Fin n → D.host.E)), Submodule.IsLattice D.host.E L → ∀ (g : GL (Fin n) (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E)) (ε : Fin n → D.host.E), ε ≠ 0 → ∃ α : (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ, ∀ c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, Matrix.mulVec g.val (c • fun i => (1 : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) ⊗ₜ[D.host.F] ε i) ∈ GlobalTheory.AdelicU.latLoc D.host.F L v ↔ ((α⁻¹ : (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitMultipliersLineInMovedLatticeSplitStatement

theorem **_root_.GlobalTheory.T.exists_unit_multipliers_line_in_movedLattice_split** : ∀ (D : D7) (v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F)), Nonempty (D.host.E →ₐ[D.host.F] Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) → ∀ (n : ℕ) (L : Submodule (𝒪 D.host.E) (Fin n → D.host.E)), Submodule.IsLattice D.host.E L → ∀ (g : GL (Fin n) (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E)) (ε : Fin n → D.host.E), ε ≠ 0 → ∃ α : (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ, ∀ c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, Matrix.mulVec g.val (c • fun i => (1 : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) ⊗ₜ[D.host.F] ε i) ∈ GlobalTheory.AdelicU.latLoc D.host.F L v ↔ ((α⁻¹ : (Formalism.MainNotio

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitSolutionTwistedFrobeniusSquare

theorem **unitFix_subMax** {K : Type} [NormedField K] [IsUltrametricDist K] (x y : K) : ‖x - y‖ ≤ max ‖x‖ ‖y‖

theorem **unitFix_powSub** {K : Type} [NormedField K] [IsUltrametricDist K] (a b : K) (ha : ‖a‖ ≤ 1) (hb : ‖b‖ ≤ 1) (hab : ‖a - b‖ < 1) : ∀ n : ℕ, ‖a ^ n - b ^ n‖ < 1

theorem **unitFix_sq_apply** {K : Type} [NormedField K] (σ : K →+* K) (x : K) : (σ ^ 2) x = σ (σ x)

theorem **unitFix_frob2** {K : Type} [NormedField K] [IsUltrametricDist K] (l : ℕ) (σ : K →+* K) (hU : Unr K l σ) (x : K) (hx : ‖x‖ ≤ 1) : ‖(σ ^ 2) x - x ^ (l ^ 2)‖ < 1

theorem **unitFix_root** {K : Type} [NormedField K] [IsUltrametricDist K] (hR : ResAlg K) (m : ℕ) (hm : 2 ≤ m) (a b : K) (ha : ‖a‖ ≤ 1) (hb : ‖b‖ ≤ 1) : ∃ y : K, ‖y‖ ≤ 1 ∧ ‖y ^ m - a * y - b‖ < 1

theorem **unitFix_step** {K : Type} [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K) (hU : Unr K l σ) (hR : ResAlg K) (c : K) (hc : ‖c‖ = 1) (n : ℕ) (ε : K) (hε : ‖ε - c * (σ ^ 2) ε‖ ≤ ‖(l : K)‖ ^ n) : ∃ η : K, ‖η‖ ≤ ‖(l : K)‖ ^ n ∧ ‖(ε + η) - c * (σ ^ 2) (ε + η)‖ ≤ ‖(l : K)‖ ^ (n + 1)

theorem **unitFix_init** {K : Type} [NormedField K] [IsUltrametricDist K] (l : ℕ) (σ : K →+* K) (hU : Unr K l σ) (hR : ResAlg K) (c : K) (hc : ‖c‖ = 1) : ∃ ε : K, ‖ε‖ = 1 ∧ ‖ε - c * (σ ^ 2) ε‖ ≤ ‖(l : K)‖ ^ 1

theorem **unit_solution_twisted_frobenius** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] [CompleteSpace K] (l : ℕ) (σ : K →+* K), Unr K l σ → ResAlg K → ∀ c : K, ‖c‖ = 1 → ∃ ε : K, ‖ε‖ = 1 ∧ ε = c * (σ ^ 2) ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitSolutionTwistedFrobeniusSquareStatement

theorem **_root_.GlobalTheory.T.exists_unit_solution_twisted_frobenius_square** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] [CompleteSpace K] (l : ℕ) (σ : K →+* K), Unr K l σ → ResAlg K → ∀ c : K, ‖c‖ = 1 → ∃ ε : K, ‖ε‖ = 1 ∧ ε = c * (σ ^ 2) ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitWeightRealCenterActionFockMonomial

theorem **central_wt** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) 𝕋) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T), x.Is → y.Is → ∀ k : KerDat S T, k.Is ψ θ x y → ∃ wt : (FVar D T.rank S.rank →₀ ℕ) → GA D S → ℂ, (∀ α, ∀ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) → ‖wt α z‖ = 1 ∧ ∀ c : WO S T →₀ ℂ, IsFinComb c → k.om (z, 1) (pureTensorO ψ θ S T x.s y.s x.B y.B (comb ψ S T c k.vac) (MvPolynomial.monomial α 1)) = wt α z • pureTensorO ψ θ S T x.s y.s x.B y.B (comb ψ S T c k.vac) (MvPolynomial.monomial α 1)) ∧ ∀ χ : GA D S → ℂ, {α | ∀ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) → wt α z = χ z}.Finite

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitWeightRealCenterActionFockMonomialStatement

theorem **_root_.GlobalTheory.T.exists_unit_weight_real_center_action_fock_monomial** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T), x.Is → y.Is → ∀ k : KerDat S T, k.Is ψ θ x y → ∃ wt : (FVar D T.rank S.rank →₀ ℕ) → GA D S → ℂ, (∀ α, ∀ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) → ‖wt α z‖ = 1 ∧ ∀ c : WO S T →₀ ℂ, IsFinComb c → k.om (z, 1) (pureTensorO ψ θ S T x.s y.s x.B y.B (comb ψ S T c k.vac) (MvPolynomial.monomial α 1)) = wt α z • pureTensorO ψ θ S T x.s y.s x.B y.B (comb ψ S T c k.vac) (MvPolynomial.monomial α 1)) ∧ ∀ χ : GA D S → ℂ, {α | ∀ z ∈ GinfC S, z ∈ Subgroup.center (GA D S) → wt α z = χ z}.Finite

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryChartsAllFinitePlacesFrameLattice

theorem **exists_charts** : ∀ (D : D7) (V : HSp D) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), (IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryChartsAllFinitePlacesFrameLatticeStatement

theorem **_root_.GlobalTheory.T.exists_unitaryCharts_allFinitePlaces_frameLattice** : ∀ (D : D7) (V : HSp D) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), (IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryEmbeddingPreimageOfFixesAnisotropic

theorem **mt_jB_eq_of_fixes_lineVec** (D : D7) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (hζ : IsLineVec L ζ) (k : GA D S) (hk : mt S k *ᵥ ratV (o.iB *ᵥ ζ) = ratV (o.iB *ᵥ ζ)) : mt S k · o.jB = o.jB

theorem **mat_eq_of_splitting** {R : Type} [CommRing R] {n a b : ℕ} (M : Matrix (Fin n) (Fin n) R) (jA : Matrix (Fin n) (Fin a) R) (pA : Matrix (Fin a) (Fin n) R) (jB : Matrix (Fin n) (Fin b) R) (pB : Matrix (Fin b) (Fin n) R) (H : Matrix (Fin a) (Fin a) R) (hX : M * jA = jA * H) (hfix : M * jB = jB) (hsum : jA * pA + jB * pB = 1) : M = jA * H * pA + jB * pB

theorem **exists_emb_of_mt_jB_eq** (D : D7) (S A B : HSp D) (o : OrthDec S A B) (k : GA D S) (hfix : mt S k · o.jB = o.jB) : ∃ h : GA D A, k = o.emb (h, 1)

theorem **exists_emb_of_fixes_lineVec** (D : D7) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (hζ : IsLineVec L ζ) (k : GA D S) (hk : mt S k *ᵥ ratV (o.iB *ᵥ ζ) = ratV (o.iB *ᵥ ζ)) : ∃ h : GA D A, k = o.emb (h, 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryEmbeddingPreimageOfFixesAnisotropicStatement

theorem **_root_.GlobalTheory.T.exists_unitaryEmbedding_preimage_of_fixes_anisotropicLine** : ∀ (D : D7) (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ k : GA D S, mt S k *ᵥ ratV (o.iB *ᵥ ζ) = ratV (o.iB *ᵥ ζ) → ∃ h : GA D A, k = o.emb (h, 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryFiniteAdelicMapSelfDualCompletions

theorem **mul_inv_mem_UJ** {R : Type} [CommRing R] {n : ℕ} (σ : R →+* R) (J : Matrix (Fin n) (Fin n) R) (β β' : GL (Fin n) R) (h : Formalism.UG.cT σ β'.1 * J * β'.1 = Formalism.UG.cT σ β.1 * J * β.1) : β' * β⁻¹ ∈ Formalism.UG.UJ σ J

theorem **mem_iff_of_frames** {R : Type} [CommRing R] {n : ℕ} (S : Subring R) (β β' : GL (Fin n) R) (H H' : Set (Fin n → R)) (h : ∀ y, y ∈ H ↔ ∀ i, ((β⁻¹).val *ᵥ y) i ∈ S) (h' : ∀ y, y ∈ H' ↔ ∀ i, ((β'⁻¹).val *ᵥ y) i ∈ S) (y : Fin n → R) : y ∈ H ↔ (β' * β⁻¹).val *ᵥ y ∈ H'

theorem **swap_symm_mem_HR** (σ : E →ₐ[F] E) (J : Matrix (Fin n) (Fin n) E) (g : GL (Fin n) (E ⊗[F] R)) (hg : g ∈ Formalism.UG.UPts σ J R) : Formalism.UG.bc (Algebra.TensorProduct.comm F R E).symm.toAlgHom.toRingHom g ∈ HR R σ J

theorem **swap_mulVec** (g : GL (Fin n) (E ⊗[F] R)) (z : Fin n → R ⊗[F] E) : (fun i => Algebra.TensorProduct.comm F R E (((Formalism.UG.bc (Algebra.TensorProduct.comm F R E).symm.toAlgHom.toRingHom g).val *ᵥ z) i)) = g.val *ᵥ fun i => Algebra.TensorProduct.comm F R E (z i)

theorem **mem_latHat_iff_swap** (Λ : Submodule (𝒪 E) (Fin n → E)) (z : Fin n → Af F ⊗[F] E) : z ∈ Formalism.MainNotions.StandardLattice.latHat F E (Λ : Set (Fin n → E)) ↔ (fun i => Algebra.TensorProduct.comm F (Af F) E (z i)) ∈ Formalism.L2.Herm.hatL (F

theorem **exists_of_swap** (σ : E →ₐ[F] E) (J : Matrix (Fin n) (Fin n) E) (Λ Λ' : Submodule (𝒪 E) (Fin n → E)) (g : GL (Fin n) (E ⊗[F] Af F)) (hg : g ∈ Formalism.UG.UPts σ J (Af F)) (h : ∀ y, y ∈ Formalism.L2.Herm.hatL (F

theorem **exists_unitary_latHat** : ∀ (D : D7) (A : HSp D) (L L' : Submodule (𝒪 D.host.E) (Fin A.rank → D.host.E)), IsSelfDualC A L → IsSelfDualC A L' → ∃ b ∈ HR (Af D.host.F) D.sp.cm.sig A.gram, ∀ w : Fin A.rank → TensorProduct D.host.F (Af D.host.F) D.host.E, w ∈ Formalism.MainNotions.StandardLattice.latHat D.host.F D.host.E (L : Set (Fin A.rank → D.host.E)) ↔ b.val *ᵥ w ∈ Formalism.MainNotions.StandardLattice.latHat D.host.F D.host.E (L' : Set (Fin A.rank → D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryFiniteAdelicMapSelfDualCompletionsStatement

theorem **_root_.GlobalTheory.T.exists_unitary_finiteAdelic_map_selfDual_completions** : ∀ (D : D7) (A : HSp D) (L L' : Submodule (𝒪 D.host.E) (Fin A.rank → D.host.E)), IsSelfDualC A L → IsSelfDualC A L' → ∃ b ∈ HR (Af D.host.F) D.sp.cm.sig A.gram, ∀ w : Fin A.rank → TensorProduct D.host.F (Af D.host.F) D.host.E, w ∈ Formalism.MainNotions.StandardLattice.latHat D.host.F D.host.E (L : Set (Fin A.rank → D.host.E)) ↔ b.val *ᵥ w ∈ Formalism.MainNotions.StandardLattice.latHat D.host.F D.host.E (L' : Set (Fin A.rank → D.host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryIntegralOfSplitPlaceComponent

theorem **matrix_mul_orthogonal_three_idempotents_smul** (a b c : R) (hab : a * b = 0) (hac : a * c = 0) (hbc : b * c = 0) (ha : a * a = a) (hb : b * b = b) (hc : c * c = c) (M₁ M₂ M₃ N₁ N₂ N₃ : Matrix (Fin n) (Fin n) R) : (a • M₁ + b • M₂ + c • M₃) * (a • N₁ + b • N₂ + c • N₃) = a • (M₁ * N₁) + b • (M₂ * N₂) + c • (M₃ * N₃)

theorem **cT_smul'** (τ : R →+* R) (a : R) (M : Matrix (Fin n) (Fin n) R) : cT τ (a • M) = τ a • cT τ M

theorem **cT_add'** (τ : R →+* R) (M N : Matrix (Fin n) (Fin n) R) : cT τ (M + N) = cT τ M + cT τ N

theorem **cT_cT'** (τ : R →+* R) (hτ : ∀ x, τ (τ x) = x) (M : Matrix (Fin n) (Fin n) R) : cT τ (cT τ M) = M

theorem **map_idem3** (a b c : R) (X Y : Matrix (Fin n) (Fin n) R) {T : Type} [CommRing T] (f : R →+* T) : (a • X + b • Y + c • (1 : Matrix (Fin n) (Fin n) R)).map f = f a • X.map f + f b • Y.map f + f c • (1 : Matrix (Fin n) (Fin n) T)

theorem **idem_split_inv** (τ : R →+* R) (a b : R) (ha : a * a = a) (hb : b * b = b) (hab : a * b = 0) (G Gi X Xi : Matrix (Fin n) (Fin n) R) (hG : G * Gi = 1) (hGi : Gi * G = 1) (hX : X * Xi = 1) : (a • X + b • (Gi * cT τ Xi * G) + (1 − a − b) • (1 : Matrix (Fin n) (Fin n) R)) * (a • Xi + b • (Gi * cT τ X * G) + (1 − a − b) • (1 : Matrix (Fin n) (Fin n) R)) = 1

theorem **idem_split_unitary** (τ : R →+* R) (hτ : ∀ x, τ (τ x) = x) (a b : R) (ha : a * a = a) (hb : b * b = b) (hab : a * b = 0) (hτa : τ a = b) (G Gi X Xi : Matrix (Fin n) (Fin n) R) (hG : G * Gi = 1) (hXi : Xi * X = 1) (hGh : cT τ G = G) : cT τ (a • X + b • (Gi * cT τ Xi

∗ G) + (1 − a − b) • (1 : Matrix (Fin n) (Fin n) R)) ∗ G ∗ (a • X + b • (Gi ∗ cT τ Xi ∗ G) + (1 − a − b) • (1 : Matrix (Fin n) (Fin n) R)) = G

theorem **exists_unitary_int_of_component** {A : Type} [CommRing A] {S : Subring A} {P : Type} {k : P → Type} [∀ Q, Field (k Q)] {r : ∀ Q, S →+* k Q} (Z : PlaceSys A S P k r) (σ : A →+* A) (hσ : ∀ a, σ (σ a) = a) (hS : ∀ a ∈ S, σ a ∈ S) (n : ℕ) (G : GL (Fin n) S) (hG : cT σ ((bc S.subtype G : GL (Fin n) A) : Matrix (Fin n) (Fin n) A) = ((bc S.subtype G : GL (Fin n) A) : Matrix (Fin n) (Fin n) A)) (Q Q' : P) (hQ : σ (Z.e Q) = Z.e Q') (hne : Q ≠ Q') (y : GL (Fin n) (Z.O Q)) : ∃ u : GL (Fin n) S, bc S.subtype u ∈ UJ σ ((bc S.subtype G : GL (Fin n) A) : Matrix (Fin n) (Fin n) A) ∧ bc (Z.pO Q) u = y ∧ ∀ Q'', Q'' ≠ Q → Q'' ≠ Q' → bc (Z.pO Q'') u = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryIntegralOfSplitPlaceComponentStatement

theorem **_root_.GlobalTheory.T.exists_unitary_integral_of_splitPlace_component** : ∀ {A : Type} [CommRing A] {S : Subring A} {P : Type} {k : P → Type} [∀ Q, Field (k Q)] {r : ∀ Q, S →+* k Q} (Z : PlaceSys A S P k r) (σ : A →+* A), (∀ a, σ (σ a) = a) → (∀ a ∈ S, σ a ∈ S) → ∀ (n : ℕ) (G : GL (Fin n) S), cT σ ((bc S.subtype G : GL (Fin n) A) : Matrix (Fin n) (Fin n) A) = ((bc S.subtype G : GL (Fin n) A) : Matrix (Fin n) (Fin n) A) → ∀ Q Q' : P, σ (Z.e Q) = Z.e Q' → Q ≠ Q' → ∀ y : GL (Fin n) (Z.O Q), ∃ u : GL (Fin n) S, bc S.subtype u ∈ UJ σ ((bc S.subtype G : GL (Fin n) A) : Matrix (Fin n) (Fin n) A) ∧ bc (Z.pO Q) u = y ∧ ∀ Q'', Q'' ≠ Q → Q'' ≠ Q' → bc (Z.pO Q'') u = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryIsometryIsotropicFrameLocalRing

theorem **cT_mul_r** (s : O →+* O) {a b c : Type} [Fintype b] (M : Matrix a b O) (N : Matrix b c O) : Formalism.UG.cT s (M ∗ N) = Formalism.UG.cT s N ∗ Formalism.UG.cT s M

theorem **cT_one_r** (s : O →+* O) {a : Type} [DecidableEq a] : Formalism.UG.cT s (1 : Matrix a a O) = 1

theorem **cT_zero_r** (s : O →+* O) {a b : Type} : Formalism.UG.cT s (0 : Matrix a b O) = 0

theorem **cT_sub_r** (s : O →+* O) {a b : Type} (M N : Matrix a b O) : Formalism.UG.cT s (M − N) = Formalism.UG.cT s M − Formalism.UG.cT s N

theorem **cT_cT_r** {s : O →+* O} (hs : ∀ x, s (s x) = x) {a b : Type} (M : Matrix a b O) : Formalism.UG.cT s (Formalism.UG.cT s M) = M

theorem **cT_fromCols_r** (s : O →+* O) {a b c : Type} (X : Matrix a b O) (Y : Matrix a c O) : Formalism.UG.cT s (fromCols X Y) = fromRows (Formalism.UG.cT s X) (Formalism.UG.cT s Y)

theorem **assoc3** {a b c d : Type} [Fintype b] [Fintype c] {X : Matrix a b O} {Y : Matrix b c O} {Z : Matrix c d O} {V : Matrix a d O} (h : X ∗ Y ∗ Z = V) : X ∗ (Y ∗ Z) = V

theorem **tail3** {a b c d : Type} [Fintype b] [Fintype c] [Fintype d] {X : Matrix a b O} {Y : Matrix b c O} {Z : Matrix c d O} {V : Matrix a d O} (h : X ∗ Y ∗ Z = V) {p : Type} (W : Matrix d p O) : X ∗ (Y ∗ (Z ∗ W)) = V ∗ W

theorem **tail2** {a b c : Type} [Fintype b] [Fintype c] {X : Matrix a b O} {Y : Matrix b c O} {V : Matrix a c O} (h : X ∗ Y = V) {p : Type} (W : Matrix c p O) : X ∗ (Y ∗ W) = V ∗ W

theorem **idem_factor** [IsLocalRing O] {ι : Type} [Fintype ι] [DecidableEq ι] (P : Matrix ι ι O) (hP : P ∗ P = P) : ∃ (m : ℕ) (B : Matrix ι (Fin m) O) (B' : Matrix (Fin m) ι O), B' ∗ B = 1 ∧ B ∗ B' = P

theorem **card_eq_of_inv** [Nontrivial O] {a b : Type} [Fintype a] [Fintype b] [DecidableEq a] [DecidableEq b] {S : Matrix b a O} {T : Matrix a b O} (h1 : S ∗ T = 1) (h2 : T ∗ S = 1) : Fintype.card a = Fintype.card b

theorem **exists_completion** [IsLocalRing O] {s : O →+* O} (hs : ∀ x, s (s x) = x) {τ : O} (hτ : τ + s τ = 1) (hnorm : ∀ u : O, IsUnit u → s u = u → ∃ x : O, s x ∗ x = u) {n r : ℕ} {Q : Matrix (Fin n) (Fin n) O} (hQ : Formalism.UG.cT s Q = Q) (hu : IsUnit Q) {C : Matrix (Fin n) (Fin r) O} {L : Matrix (Fin r) (Fin n) O} (hL : L ∗ C = 1) (hC : Formalism.UG.cT s C ∗ Q ∗ C = 0) : ∃ (m : ℕ) (T : Matrix (Fin n) ((Fin r ⊕ Fin r) ⊕ Fin m) O) (S : Matrix ((Fin r ⊕ Fin r) ⊕ Fin m) (Fin n) O), S ∗ T = 1 ∧ T ∗ S = 1 ∧ Formalism.UG.cT s T ∗ Q ∗ T = fromBlocks (fromBlocks 0 1 1 0) 0 0 1 ∧ T ∗ (fromRows (fromRows (1 : Matrix (Fin r) (Fin r) O) 0) (0 : Matrix (Fin m) (Fin r) O) : Matrix ((Fin r ⊕ Fin r) ⊕ Fin m) (Fin r) O) = C

theorem **exists_isometry_frame** : ∀ (O : Type) [CommRing O] [IsLocalRing O] (s : O →+* O), (∀ x, s (s x) = x) → (∃ τ : O, τ + s τ = 1) → (∀ u : O, IsUnit u → s u = u → ∃ x : O, s x ∗ x = u) → ∀ (n r : ℕ) (Q : Matrix (Fin n) (Fin n) O), Formalism.UG.cT s Q = Q → IsUnit Q → ∀ C C' : Matrix (Fin n) (Fin r) O, (∃ L : Matrix (Fin r) (Fin n) O, L ∗ C = 1) → (∃ L : Matrix (Fin r) (Fin n) O, L ∗ C' = 1) → Formalism.UG.cT s C ∗ Q ∗ C = 0 → Formalism.UG.cT s C' ∗ Q ∗ C' = 0 → ∃ k : GL (Fin n) O, Formalism.UG.cT s k.1 ∗ Q ∗ k.1 = Q ∧ k.1 ∗ C = C'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryIsometryIsotropicFrameLocalRingStatement

theorem **_root_.GlobalTheory.T.exists_unitaryIsometry_isotropicFrame_localRing** : ∀ (O : Type) [CommRing O] [IsLocalRing O] (s : O →+* O), (∀ x, s (s x) = x) → (∃ τ : O, τ + s τ = 1) → (∀ u : O, IsUnit u → s u = u → ∃ x : O, s x ∗ x = u) → ∀ (n r : ℕ) (Q : Matrix (Fin n) (Fin n) O), Formalism.UG.cT s Q = Q → IsUnit Q → ∀ C C' : Matrix (Fin n) (Fin r) O, (∃ L : Matrix (Fin r) (Fin n) O, L ∗ C = 1) → (∃ L : Matrix (Fin r) (Fin n) O, L ∗ C' = 1) → Formalism.UG.cT s C ∗ Q ∗ C = 0 → Formalism.UG.cT s C' ∗ Q ∗ C' = 0 → ∃ k : GL (Fin n) O, Formalism.UG.cT s k.1 ∗ Q ∗ k.1 = Q ∧ k.1 ∗ C = C'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryScalarBlendSelfAdjointIdempotent

theorem **hform_mulVec_two** {A : Type} [CommRing A] {n : Type} [Fintype n] (σ : A →+* A) (J M M' : Matrix n n A) (x y : n → A) : Formalism.UG.hform σ J (M *ᵥ x) (M' *ᵥ y) = Formalism.UG.hform σ (Formalism.UG.cT σ M ∗ J ∗ M') x y

theorem **hform_single** {A : Type} [CommRing A] {n : Type} [Fintype n] [DecidableEq n] (σ : A →+* A) (N : Matrix n n A) (i j : n) : Formalism.UG.hform σ N (Pi.single i 1) (Pi.single j 1) = N i j

theorem **matrix_eq_of_hform** {A : Type} [CommRing A] {n : Type} [Fintype n] [DecidableEq n] (σ : A →+* A) (N N' : Matrix n n A) (h : ∀ x y, Formalism.UG.hform σ N x y = Formalism.UG.hform σ N' x y) : N = N'

theorem **unit_of_proj** {A : Type} [CommRing A] {n : Type} [Fintype n] [DecidableEq n] (σ : A →+* A) (J P : Matrix n n A) (c : A) (hPP : P ∗ P = P) (hSA : Formalism.UG.cT σ P ∗ J = J ∗ P) (hc : c ∗ σ c = 1) : (c • 1 + (1 − c) • P) ∗ (σ c • 1 + (1 − σ c) • P) = 1 ∧ (σ c • 1 + (1 − σ c) • P) ∗ (c • 1 + (1 − c) • P) = 1 ∧ Formalism.UG.cT σ (c • 1 + (1 − c) • P) ∗ J ∗ (c • 1 + (1 − c) • P) = J

theorem **GlobalTheory.unitTorus_unit** : ∀ (I : AutInst) (P : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Af I.F)) (c : I.E ⊗[I.F] Af I.F), (∀ y, P *ᵥ (P *ᵥ y) = P *ᵥ y) → (∀ x y, hA I (P *ᵥ x) y = hA I x (P *ᵥ y)) → c ∗ sigA I c = 1 → ∃ h : I.Gf, I.fmat h = c • 1 + (1 − c) • P ∧ I.fmat h⁻¹ = sigA I c • 1 + (1 − sigA I c) • P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitaryScalarBlendSelfAdjointIdempotentStatement

theorem **_root_.GlobalTheory.T.exists_unitary_scalarBlend_selfAdjoint_idempotent** : ∀ (I : AutInst) (P : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Af I.F)) (c : I.E ⊗[I.F] Af I.F), (∀ y, P *ᵥ (P *ᵥ y) = P *ᵥ y) → (∀ x y, hA I (P *ᵥ x) y = hA I x (P *ᵥ y)) → c ∗ sigA I c = 1 → ∃ h : I.Gf, I.fmat h = c • 1 + (1 − c) • P ∧ I.fmat h⁻¹ = sigA I c • 1 + (1 − sigA I c) • P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitarySupportedAtPlaceOfDualIsotropicFram

theorem **mulVec_colMul** {m n k : ℕ} (M : Mx D m n) (X : Mx D n k) (i : Fin k) (a : AE D) : M *ᵥ colMul X i a = fun r => a * (M · X) r i

theorem **adj_diagonal** {n : ℕ} (f : Fin n → AE D) : adj (Matrix.diagonal f) = Matrix.diagonal (fun i => cA D (f i))

theorem **mul_tail** {a m n k : ℕ} {A : Mx D a m} {B : Mx D m n} {C : Mx D a n} (h : A · B = C) (X : Mx D n k) : A · (B · X) = C · X

theorem **mul_tail₃** {a m n p k : ℕ} {A : Mx D a m} {B : Mx D m n} {B' : Mx D n p} {C : Mx D a p} (h : A · (B · B') = C) (X : Mx D p k) : A · (B · (B' · X)) = C · X

theorem **mul_tail₄** {a m n p q k : ℕ} {A : Mx D a m} {B : Mx D m n} {B' : Mx D n p} {B'' : Mx D p q} {C : Mx D a q} (h : A · (B · (B' · B'')) = C) (X : Mx D q k) : A · (B · (B' · (B'' · X))) = C · X

theorem **gXX** {k : ℕ} (X : Mx D ρ k) : adj d.jX · (gA S · (d.jX · X)) = 0

theorem **gYY** {k : ℕ} (X : Mx D ρ k) : adj d.jY · (gA S · (d.jY · X)) = 0

theorem **gYX** {k : ℕ} (X : Mx D ρ k) : adj d.jY · (gA S · (d.jX · X)) = X

theorem **gXY** {k : ℕ} (X : Mx D ρ k) : adj d.jX · (gA S · (d.jY · X)) = X

theorem **g0X** {k : ℕ} (X : Mx D ρ k) : adj d.j0 · (gA S · (d.jX · X)) = 0

theorem **g0Y** {k : ℕ} (X : Mx D ρ k) : adj d.j0 · (gA S · (d.jY · X)) = 0

theorem **adjoint_jX_gram_j_zero_eq_zero** {k : ℕ} (X : Mx D S₀.rank k) : adj d.jX · (gA S · (d.j0 · X)) = 0

theorem **g00** {k : ℕ} (X : Mx D S₀.rank k) : adj d.j0 · (gA S · (d.j0 · X)) = gA S₀ · X

theorem **smul_eq_of_mem** {P : Pl D → Prop} {ε : AE D} (hε : ∀ a, a ∈ idAt D P ↔ ε * a = a) {m n : ℕ} {x : Mx D m n} (hx : ∀ i j, x i j ∈ idAt D P) : ε • x = x

theorem **one_sub_mul_self** {ε : AE D} (hεε : ε * ε = ε) : (1 − ε) * ε = 0

theorem **one_sub_sq** {ε : AE D} (hεε : ε * ε = ε) : (1 − ε) * (1 − ε) = 1 − ε

theorem **kill_left** {ε : AE D} (hεε : ε * ε = ε) {m n k : ℕ} {x : Mx D m n} (hx : ε • x = x) (X : Mx D n k) : (1 − ε) • (x · X) = 0

theorem **kill_right** {ε : AE D} (hεε : ε * ε = ε) {m n k : ℕ} {x : Mx D n k} (hx : ε • x = x) (X : Mx D m n) : (1 − ε) • (X · x) = 0

theorem **diag_absorb** {ε : AE D} {n : ℕ} {α : Fin n → AE D} (hαε : ∀ i, ε * α i = α i) : ε • Matrix.diagonal α = Matrix.diagonal α

theorem **diag_const** {ε : AE D} {n : ℕ} {α β : Fin n → AE D} (hαβ : ∀ i, α i * β i = ε) : (Matrix.diagonal fun i => α i * β i) = ε • (1 : Mx D n n)

theorem **imYX** {ε : AE D} (hεε : ε * ε = ε) (hεc : cA D ε = ε) {α β : Fin ρ → AE D} (hαε : ∀ i, ε * α i = α i) (hαβ : ∀ i, α i * β i = ε) {b : Mx D ρ ρ} (hb : ε • b = b) {c' : Mx D S.rank ρ} (hc : ε • c' = c') (H1 : adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ)) : adj (c' · Matrix.diagonal (fun i => cA D (α i)) + (1 − ε) • d.jY) · (gA S · (d.jX · (b · Matrix.diagonal β + (1 − ε) • (1 : Mx D ρ ρ)))) = 1

theorem **kill_mid₂** {ε : AE D} (hεε : ε * ε = ε) {m n p q k : ℕ} {x : Mx D p q} (hx : ε • x = x) (A : Mx D m n) (B : Mx D n p) (X : Mx D q k) : (1 − ε) • (A · (B · (x · X))) = 0

theorem **absorb_mul** {ε : AE D} {m n k : ℕ} {x : Mx D m n} (hx : ε • x = x) (X : Mx D n k) : x · (ε • X) = x · X

theorem **imXX** {ε : AE D} {β : Fin ρ → AE D} {b : Mx D ρ ρ} : adj (d.jX · (b · Matrix.diagonal β + (1 − ε) • (1 : Mx D ρ ρ))) · (gA S · (d.jX · (b · Matrix.diagonal β + (1 − ε) • (1 : Mx D ρ ρ)))) = 0

theorem **imYY** {ε : AE D} (hεε : ε * ε = ε) (hεc : cA D ε = ε) {α : Fin ρ → AE D} (hαε : ∀ i, ε * α i = α i) {c' : Mx D S.rank ρ} (hc : ε • c' = c') (H2 : adj c' · (gA S · c') = 0) : adj (c' · Matrix.diagonal (fun i => cA D (α i)) + (1 − ε) • d.jY) · (gA S · (c' · Matrix.diagonal (fun i => cA D (α i)) + (1 − ε) • d.jY)) = 0

theorem **imOO** {b : Mx D ρ ρ} {c' : Mx D S.rank ρ} : adj (d.j0 − (d.jX · b) · (adj c' · (gA S · d.j0))) · (gA S · (d.j0 − (d.jX · b) · (adj c' · (gA S · d.j0)))) = gA S₀

theorem **imOX** {ε : AE D} {β : Fin ρ → AE D} {b : Mx D ρ ρ} {c' : Mx D S.rank ρ} : adj (d.j0 − (d.jX · b) · (adj c' · (gA S · d.j0))) · (gA S · (d.jX · (b · Matrix.diagonal β + (1 − ε) • (1 : Mx D ρ ρ)))) = 0

theorem **imOY** {ε : AE D} (hεε : ε * ε = ε) (hεc : cA D ε = ε) {α : Fin ρ → AE D} {b : Mx D ρ ρ} (hb : ε • b = b) {c' : Mx D S.rank ρ} (hc : ε • c' = c') (H1 : adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ)) : adj (d.j0 − (d.jX · b) · (adj c' · (gA S · d.j0))) · (gA S · (c' · Matrix.diagonal (fun i => cA D (α i)) + (1 − ε) • d.jY)) = 0

theorem **smul_sub_one** {ε : AE D} (M : Mx D S.rank S.rank) (hX : ε • (M · d.jX − d.jX) = M · d.jX − d.jX) (h0 : ε • (M · d.j0 − d.j0) = M · d.j0 − d.j0) (hY : ε • (M · d.jY − d.jY) = M · d.jY − d.jY) : ε • (M − 1) = M − 1

theorem **smul_shift** {ε : AE D} (hεε : ε * ε = ε) {m n : ℕ} {x : Mx D m n} (hx : ε • x = x) (y : Mx D m n) : ε • (x + (1 − ε) • y − y) = x + (1 − ε) • y − y

theorem **scalar_X** {R : Type*} [CommRing R] {ε a b x y : R} (ha : ε * a = a) (hab : a * b = ε) (hx : ε * x = x) : a * (x * b + (1 − ε) * y) = x

theorem **GlobalTheory.IwaPlace.exists_grpAt_of_dual_frames** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (ε : AE D) (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) (α β : Fin ρ → AE D) (b : Mx D ρ ρ) (c' : Mx D S.rank ρ), (∀ i, α i ∈ idAt D (· = v)) → (∀ i, β i ∈ idAt D (· = v)) → (∀ i, α i * β i = ε) → (∀ i j, b i j ∈ idAt D (· = v)) → (∀ i j, c' i j ∈ idAt D (· = v)) → adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ) → adj c' · (gA S · c') = 0 → ∃ q ∈ grpAt S (· = v), mt S q · d.jX = d.jX · (b · Matrix.diagonal β + (1 − ε) • (1 : Mx D ρ ρ)) ∧ (∀ i, mt S q *ᵥ colMul d.jX i (α i) = fun k => (d.jX · b) k i) ∧ (∀ i, mt S q *ᵥ colMul d.jY i (cA D (β i)) = fun k => c' k i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnitarySupportedAtPlaceOfDualIsotropicFramStatement

theorem **_root_.GlobalTheory.T.exists_unitarySupportedAtPlace_of_dual_isotropic_frames** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (ε : AE D) (hεε : ε * ε = ε) (hεc : cA D ε = ε) (hεI : ε ∈ intAt D (· = v)) (hε : ∀ a, a ∈ idAt D (· = v) ↔ ε * a = a) (α β : Fin ρ → AE D) (b : Mx D ρ ρ) (c' : Mx D S.rank ρ), (∀ i, α i ∈ idAt D (· = v)) → (∀ i, β i ∈ idAt D (· = v)) → (∀ i, α i * β i = ε) → (∀ i j, b i j ∈ idAt D (· = v)) → (∀ i j, c' i j ∈ idAt D (· = v)) → adj c' · (gA S · (d.jX · b)) = ε • (1 : Mx D ρ ρ) → adj c' · (gA S · c') = 0 → ∃ q ∈ grpAt S (· = v), mt S q · d.jX = d.jX · (b · Matrix.diagonal β + (1 − ε) • (1 : Mx D ρ ρ)) ∧ (∀ i, mt S q *ᵥ colMul d.jX i (α i) = fun k => (d.jX · b) k i) ∧ (∀ i, mt S q *ᵥ colMul d.jY i (cA D (β i)) = fun k => c' k i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnramifiedNearIntegralComplexPadic

theorem **norm_algebraMap_padicInt** (z : ℤ_[l]) : ‖algebraMap ℤ_[l] (PadicAlgCl l) z‖ ≤ 1

theorem **norm_aeval_le** (x : PadicAlgCl l) (hx : ‖x‖ ≤ 1) (g : ℤ_[l][X]) : ‖aeval x g‖ ≤ 1

theorem **norm_aeval_lt** (x : PadicAlgCl l) (hx : ‖x‖ ≤ 1) (g : ℤ_[l][X]) (hg : g.map (PadicInt.toZMod (p

theorem **exists_monic** (x : PadicAlgCl l) (hx : ‖x‖ ≤ 1) : ∃ q : ℤ_[l][X], q.Monic ∧ aeval x q = 0

theorem **exists_pow_near_one** (x : PadicAlgCl l) (hx : ‖x‖ = 1) : ∃ m : ℕ, 0 < m ∧ ‖x ^ m - 1‖ < 1

theorem **frob_inj** (z : PadicAlgCl l) (hz : ‖z‖ ≤ 1) (h : ‖z ^ l - 1‖ < 1) : ‖z - 1‖ < 1

theorem **frob_pow** (a : ℕ) : ∀ z : PadicAlgCl l, ‖z‖ ≤ 1 → ‖z ^ (l ^ a) - 1‖ < 1 → ‖z - 1‖ < 1

theorem **one_le_norm_prod** (s : Multiset ℂ_[l]) (h : ∀ a ∈ s, 1 ≤ ‖a‖) : 1 ≤ ‖s.prod‖

theorem **exists_root_near** (x : ℂ_[l]) (m : ℕ) (hm : 0 < m) (h : ‖x ^ m - 1‖ < 1) : ∃ ζ : ℂ_[l], ζ ^ m = 1 ∧ ‖x - ζ‖ < 1

theorem **exists_near_kur** : ∀ (l : ℕ) [Fact l.Prime] (x : ℂ_[l]), ‖x‖ ≤ 1 → ∃ y ∈ MaximalUnramifiedExtension l, ‖y‖ ≤ 1 ∧ ‖x - y‖ < 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsUnramifiedNearIntegralComplexPadicStatement

theorem **_root_.GlobalTheory.T.exists_unramified_near_integral_complexPadic** : ∀ (l : ℕ) [Fact l.Prime] (x : ℂ_[l]), ‖x‖ ≤ 1 → ∃ y ∈ MaximalUnramifiedExtension l, ‖y‖ ≤ 1 ∧ ‖x - y‖ < 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsVTOver

theorem **incA_mul_mem_intAt** {v : Pl D} {c : D.host.E} {N : ℕ} (hv : (N : 𝒪 D.host.F) ∉ v.asIdeal) (hint : IsIntegral ℤ ((N : D.host.E) * c)) : ∀ a ∈ intAt D (· = v), incA D c * a ∈ intAt D (· = v)

theorem **latA_eq_latV_of** {v : Pl D} {N M : ℕ} (hvN : (N : 𝒪 D.host.F) ∉ v.asIdeal) (hvM : (M : 𝒪 D.host.F) ∉ v.asIdeal) (hNI : ∀ a ∈ D.host.AA, ∃ b : 𝒪 D.host.E, algebraMap _ _ b = a * (N : D.host.E)) (hMI : (M : D.host.E) ∈ D.host.AA) (S : HSp D) (L : Lat S) : latA S L (· = v) = latV S L (· = v)

theorem **notMem_of_coprime** {R : Type} [CommRing R] {P : Ideal R} (hP : P.IsPrime) {a b : ℕ} (h : a.Coprime b) (hb : (b : R) ∈ P) : (a : R) ∉ P

theorem **unitAA_of** (p : SplitPt D k) {l : ℕ} (hl : l.Prime) (hv : (l : 𝒪 D.host.F) ∈ p.v.asIdeal) (hq : (D.host.QS * D.host.q).Coprime l) (hd : D.host.d.Coprime l) (hc : (Ideal.absNorm D.host.ccK).Coprime l) : p.UnitAA

theorem **mxAt_of_half** (q : SplitPt D k) {m n : ℕ} {w : Mx D m n} (h : IsHalf (q.e + cA D q.e) w) : MxAt (· = q.v) w

theorem **sec_mem_idAt** (p : SplitPt D k) (a : k) : p.sec a ∈ idAt D (· = p.v)

theorem **idAt_mono** {P Q : Pl D → Prop} (h : ∀ v, P v → Q v) : idAt D P ≤ idAt D Q

theorem **mem_latV_int** {S : HSp D} {L : Lat S} (hL : ∀ z, z ∈ L ↔ z ∈ intVec D S.rank) (v : Pl D) (y : Fin S.rank → AE D) : y ∈ latV S L (· = v) ↔ ∀ i, y i ∈ intAt D (· = v)

theorem **exists_vT_over** {X : Formalism.L2.Col.Fields} {Pr : Formalism.L2.Col.Primes X} (hX : X.Ok) (hPr : Pr.Ok) (u : Pr.Word) (v' : HeightOneSpectrum (𝒪 X.F)) (hT : ((Pr.T u : ℕ) : 𝒪 X.F) ∈ v'.asIdeal) : ∃ P ∈ Pr.vT u, P.under (𝒪 X.F) = v'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsVacuumVectorPrincipalCongruenceSupport

theorem **exists_small_vac_vec** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sA : RealMat A) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → A.rank ≤ H.rank → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∃ f, IsThetaFunO ψ A H f ∧ f ≠ 0 ∧ (∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO (sA τ) (sH τ)) f) ∧ IsSphOff ψ A H (frameLat A) (frameLat H) (under Z) f ∧ (∀ κ ∈ prin A β Z, om (κ, 1) f = f) ∧ ∀ ν 𝓕, IsCovolOne (ratL H A) ν 𝓕 → ∀ g ∈ openOff A (frameLat A) (under Z), g ∉ prinInf A β Z → wcoef om ν 𝓕 f f g = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsVacuumVectorPrincipalCongruenceSupportStatement

theorem **_root_.GlobalTheory.T.exists_vacuumVector_principalCongruenceSupport** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A H : HSp D) (χA : (AE D)ˣ →* Circle) (om : WRep A H) (sA : RealMat A) (sH : RealMat H), IsInfFam θ → (∃ η, GlobalTheory.WeilParts.IsEtaTilde D η) → IsWeilDatumO ψ A H χA 1 om → (∀ τ, IsCartanInv (gramAt D A τ) (sA τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → A.rank ≤ H.rank → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∃ f, IsThetaFunO ψ A H f ∧ f ≠ 0 ∧ (∀ τ, IsFockVacAt (heisO ψ A H) θ τ (cplxO (sA τ) (sH τ)) f) ∧ IsSphOff ψ A H (frameLat A) (frameLat H) (under Z) f ∧ (∀ κ ∈ prin A β Z, om (κ, 1) f = f) ∧ ∀ ν 𝓕, IsCovolOne (ratL H A) ν 𝓕 → ∀ g ∈ openOff A (frameLat A) (under Z), g ∉ prinInf A β Z → wcoef om ν 𝓕 f f g = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumDoubledOfSchrodingerFormulas

theorem **exists_datum** : ∀ (D : D7) (A V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (ν : Measure (WO A V)) (𝒴 : Set (WO A V)), IsCovolOne (ratL V A) ν 𝒴 → ∀ ω : G D A.rank →* Module.End ℂ (schwartzA θ V.rank A.rank), IsWeilSch ψ χ θ V ν ω → ∃ (ψ₀ : AddChar (AF D.host.F) ℂ) (μ : D.host.E) (lam : D.host.F) (κ : GA D (dblSp A) ≃* G D A.rank) (omh : GA D (dblSp A) × GA D V →* Module.End ℂ (WO (dblSp A) V → ℂ)), IsStdChar D ψ₀ (-1) ∧ IsSkew μ ∧ algebraMap D.host.F D.host.E lam = -D.host.xi0 / μ ∧ ψ = shift ψ₀ lam ∧ IsChart A μ κ ∧ IsWeilDatumO ψ₀ (dblSp A) V 1 χ⁻¹ omh ∧ IsTransport ψ₀ θ A V ν 𝒴 omh κ ω

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumDoubledOfSchrodingerFormulasStatement

theorem **_root_.GlobalTheory.T.exists_weilDatum_doubled_of_schrodingerFormulas** : ∀ (D : D7) (A V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (ν : Measure (WO A V)) (𝒴 : Set (WO A V)), IsCovolOne (ratL V A) ν 𝒴 → ∀ ω : G D A.rank →* Module.End ℂ (schwartzA θ V.rank A.rank), IsWeilSch ψ χ θ V ν ω → ∃ (ψ₀ : AddChar (AF D.host.F) ℂ) (μ : D.host.E) (lam : D.host.F) (κ : GA D (dblSp A) ≃* G D A.rank) (omh : GA D (dblSp A) × GA D V →* Module.End ℂ (WO (dblSp A) V → ℂ)), IsStdChar D ψ₀ (-1) ∧ IsSkew μ ∧ algebraMap D.host.F D.host.E lam = -D.host.xi0 / μ ∧ ψ = shift ψ₀ lam ∧ IsChart A μ κ ∧ IsWeilDatumO ψ₀ (dblSp A) V 1 χ⁻¹ omh ∧ IsTransport ψ₀ θ A V ν 𝒴 omh κ ω

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumHermitianPair

theorem **weilDatumExists** : ∀ D : D7, GlobalTheory.IndefiniteRowSideFullHecke.WeilDatumExistenceCriterion.WeilDatumExists D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumHermitianPairStatement

theorem **_root_.GlobalTheory.T.exists_weilDatum_hermitianPair** : ∀ D : D7, GlobalTheory.IndefiniteRowSideFullHecke.WeilDatumExistenceCriterion.WeilDatumExists D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumHermitianSpaceDoubled

theorem **exists_doubled_weilDatum** : GlobalTheory.Published.WeilRepresentationUnitaryPair → ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (om : WRep A B), IsWeilDatumO ψ A B χA χB om → ∃ omh : WRep (dblSp A) B, IsWeilDatumO ψ (dblSp A) B 1 χB omh

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumHermitianSpaceDoubledStatement

theorem **_root_.GlobalTheory.T.exists_weilDatum_hermitianSpaceDoubled** : GlobalTheory.Published.WeilRepresentationUnitaryPair → ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χA χB : (AE D)ˣ →* Circle) (om : WRep A B), IsWeilDatumO ψ A B χA χB om → ∃ omh : WRep (dblSp A) B, IsWeilDatumO ψ (dblSp A) B 1 χB omh

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumTransportSchwartzBruhatStandard

theorem **exists_hom_of_model** (P : X → Prop) (s : X → Y) (S : Submodule ℂ Y) (hadd : ∀ f f', P f → P f' → P (f + f') ∧ s (f + f') = s f + s f') (hsmul : ∀ (c : ℂ) f, P f → P (c • f) ∧ s (c • f) = c • s f) (hin : ∀ f, P f → s f ∈ S) (honto : ∀ φ ∈ S, ∃ f, P f ∧ s f = φ) (hinj : ∀ f f', P f → P f' → s f = s f' → f = f') (om : G →* Module.End ℂ X) (hpres : ∀ g f, P f → P (om g f)) (κ : G ≃* H) : ∃ ρ : H →* Module.End ℂ S, ∀ g f, P f → ∀ Φ : S, Φ.1 = s f → (ρ (κ g) Φ).1 = s (om g f)

theorem **exists_transport_std** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ θ : InfFam D, IsInfFam θ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, ∃ omS : G D A.rank →* Module.End ℂ (schwartzA θ B.rank A.rank), IsTransport ψ θ A B ν 𝒴 omh κ omS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilDatumTransportSchwartzBruhatStandardStatement

theorem **_root_.GlobalTheory.T.exists_weilDatum_transport_schwartzBruhat_standard** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D) (χB : (AE D)ˣ →* Circle) (omh : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)), IsWeilDatumO ψ (dblSp A) B 1 χB omh → ∀ (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ θ : InfFam D, IsInfFam θ → ∀ κ : GA D (dblSp A) ≃* G D A.rank, ∃ omS : G D A.rank →* Module.End ℂ (schwartzA θ B.rank A.rank), IsTransport ψ θ A B ν 𝒴 omh κ omS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilSourceActionAndSmooth

theorem **isSrc_exists_isSmoothRep** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (a b : Type) [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)), (∃ ρ' : Representation ℂ (GL a k) (SB (Matrix b a k)), IsSrc μ ρ') ∧ ∀ ρ' : Representation ℂ (GL a k) (SB (Matrix b a k)), IsSrc μ ρ' → IsSmoothRep ρ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsWeilSourceActionAndSmoothStatement

theorem **_root_.GlobalTheory.T.exists_weilSourceAction_and_smooth** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (a b : Type) [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)), (∃ ρ' : Representation ℂ (GL a k) (SB (Matrix b a k)), IsSrc μ ρ') ∧ ∀ ρ' : Representation ℂ (GL a k) (SB (Matrix b a k)), IsSrc μ ρ' → IsSmoothRep ρ'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExistsZsmulMemLattice

theorem **exists_int_mul_mem** {E : Type} [Field E] [NumberField E] (a : E) : ∃ (m : ℤ) (b : RingOfIntegers E), m ≠ 0 ∧ algebraMap (RingOfIntegers E) E b = m • a

theorem **exists_zsmul_mem_lattice** {E : Type} [Field E] [NumberField E] {n : ℕ} (L : Submodule (RingOfIntegers E) (Fin n → E)) [Submodule.IsLattice E L] (x : Fin n → E) : ∃ (m : ℤ), m ≠ (0 : ℤ) ∧ m • x ∈ L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExpTraceEqBrk

theorem **exp_trace_eq_brk** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ M Y : Matrix (Fin T.rank) (Fin S.rank) D.host.E, Complex.exp (2 * Real.pi * Complex.I * ((Algebra.trace ℚ D.host.E (D.host.xi0 * GlobalTheory.LatVac.hRat S T M Y) : ℚ) : ℂ)) = brk (HV sc) (embH (tauE D) δ (Matrix.toLin' Y)) (embH (tauE D) δ (Matrix.toLin' M))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtXCastAdd

theorem **locHom_val** (x : Sp E N ≃ₗ[E] Sp E N) : (locHom ι e x).val = toLoc ι e e x.toLinearMap

theorem **eX_lt** (a : A) : (e (Sum.inl a)).val < N

theorem **eX_val** (a : A) : ((eX e hcc he a : Fin N) : ℕ) = (e (Sum.inl a)).val

theorem **extX_castAdd** (y : Sp E N ≃ₗ[E] Sp E N) (v : Sp E (N + c)) (i : Fin N) : extX c y v (Fin.castAdd c i) = y (fun i => v (Fin.castAdd c i)) i

theorem **extX_natAdd** (y : Sp E N ≃ₗ[E] Sp E N) (v : Sp E (N + c)) (i : Fin c) : extX c y v (Fin.natAdd N i) = v (Fin.natAdd N i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtXUnipotentMemLattice

theorem **latV_mono** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {v : Pl D} (hv : P v) : latV S L (· = v) ≤ latV S L P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtZeroIncZeroMemLattice

theorem **mulVec_idAt** {m n : ℕ} (M : Mx D m n) {P : Pl D → Prop} {y : Fin n → AE D} (hy : ∀ i, y i ∈ idAt D P) : ∀ i, (M *ᵥ y) i ∈ idAt D P

theorem **wittProjectionZero_mem_latticeCompletion_of_splits** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} {P : Pl D → Prop} (hL₀ : Submodule.IsLattice D.host.E L₀) (hm : ∀ v, P v → MonoAt d L L₀ v) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : d.p0 *ᵥ y ∈ latV S₀ L₀ P

theorem **wittCoreInclusion_mem_latticeCompletionAtPlace** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {L : Lat T} {L₀ : Lat T₀} {P : Pl D → Prop} (hL : Submodule.IsLattice D.host.E L) (hm : ∀ v, P v → MonoAt d L L₀ v) {u : Fin T₀.rank → AE D} (hu : u ∈ latV T₀ L₀ P) : d.j0 *ᵥ u ∈ latV T L P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroLeftMemCoreLattice

theorem **extL_mem_latO** : ∀ (D : D7) (V₀ A L S : HSp D) (o : OrthDec V₀ A L) (P : Pl D → Prop) (LA : Lat A) (LS : Lat S) (M : Lat V₀), IsSelfDualC A LA → IsSelfDualC S LS → IsSelfDualC V₀ M → (∀ v, P v → ∃ LL, OrthAt o M LA LL v) → ∀ u ∈ latO A S (LA : Set _) (LS : Set

_) P, (o.resS S).symm (u, 0) ∈ latO V₀ S (M : Set _) (LS : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroLeftMemCoreLatticeStatement

theorem **_root_.GlobalTheory.T.extendByZero_left_mem_core_lattice** : ∀ (D : D7) (V₀ A L S : HSp D) (o : OrthDec V₀ A L) (P : Pl D → Prop) (LA : Lat A) (LS : Lat S) (M : Lat V₀), IsSelfDualC A LA → IsSelfDualC S LS → IsSelfDualC V₀ M → (∀ v, P v → ∃ LL, OrthAt o M LA LL v) → ∀ u ∈ latO A S (LA : Set _) (LS : Set _) P, (o.resS S).symm (u, 0) ∈ latO V₀ S (M : Set _) (LS : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroMemCompletedLatticeSource

theorem **mxAt_mul** {P : Pl D → Prop} {a b c : ℕ} {x : Mx D a b} (hx : MxAt P x) (y : Mx D b c) : MxAt P (x · y)

theorem **symplO_add_left** (S T : HSp D) (x x' y : WO S T) : symplO S T (x + x') y = symplO S T x y + symplO S T x' y

theorem **symplO_add_right** (S T : HSp D) (x y y' : WO S T) : symplO S T x (y + y') = symplO S T x y + symplO S T x y'

theorem **symplO_conjHalf** {S T : HSp D} {e : AE D} (he : IsSplitIdem D e) {x y : WO S T} (hx : IsHalf (cA D e) x) (hy : IsHalf (cA D e) y) : symplO S T x y = 0

theorem **symplO_resS_symm_left** {S A L T : HSp D} (o : OrthDec S A L) (a : WO A T) (t : WO S T) : symplO S T ((o.resS T).symm (a, 0)) t = symplO A T a (t · o.jA)

theorem **smul_isHalf** {e : AE D} (he : e ∗ e = e) {m n : ℕ} (u : Mx D m n) : IsHalf e (e • u)

theorem **finPart_idem_mem** {e : AE D} (he : e ∗ e = e) : finPart D e ∈ intAE D

theorem **GlobalTheory.FinPl.ExtZero.extZero_mem_latO** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), (∀ u : WO A T, MxAt (· = p.v) u → (u ∈ latO A T (LA : Set _) (LT : Set _) (· = p.v) ↔ ∀ u' ∈ latO A T (LA : Set _) (LT : Set _) (· = p.v), ψ (symplO A T u u') = 1)) → (∀ u : WO S T, MxAt (· = p.v) u → (u ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v) ↔ ∀ u' ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v), ψ (symplO S T u u') = 1)) → halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v)) = o.boxS T (halfInd p.e (latO A T (LA : Set _) (LT : Set _) (· = p.v))) (GlobalTheory.FinPl.ExtZero.colFun β ζ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroMemCompletedLatticeSourceStatement

theorem **_root_.GlobalTheory.T.extendByZero_mem_completed_lattice_source** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), (∀ u : WO A T, MxAt (· = p.v) u → (u ∈ latO A T (LA : Set _) (LT : Set _) (· = p.v) ↔ ∀ u' ∈ latO A T (LA : Set _) (LT : Set _) (· = p.v), ψ (symplO A T u u') = 1)) → (∀ u : WO S T, MxAt (· = p.v) u → (u ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v) ↔ ∀ u' ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v), ψ (symplO S T u u') = 1)) → halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v)) = o.boxS T (halfInd p.e (latO A T (LA : Set _) (LT : Set _) (· = p.v))) (GlobalTheory.FinPl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroRightMemCoreLattice

theorem **latV_entry_mem** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) (i : Fin S.rank) : y i ∈ idAt D P

theorem **jA_latV** {V₀ A L : HSp D} (o : OrthDec V₀ A L) {P : Pl D → Prop} {LA : Lat A} {M : Lat V₀} (hLA : IsSelfDualC A LA) (hM : IsSelfDualC V₀ M) (hloc : ∀ v, P v → ∃ LL, OrthAt o M LA LL v) {y : Fin A.rank → AE D} (hy : y ∈ latV A LA P) : o.jA ∗ᵥ y ∈ latV V₀ M P

theorem **smul_latV_mem_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {w : Fin S.rank → AE D} (hw : w ∈ latV S L P) : ((1 : AF D.host.F) ⊗ₜ[D.host.F] a : AE D) • w ∈ latA S L P

theorem **jA_latA** {V₀ A L : HSp D} (o : OrthDec V₀ A L) {P : Pl D → Prop} {LA : Lat A} {M : Lat V₀} (hLA : IsSelfDualC A LA) (hM : IsSelfDualC V₀ M) (hloc : ∀ v, P v → ∃ LL, OrthAt o M LA LL v) {z : Fin A.rank → AE D} (hz : z ∈ latA A LA P) : o.jA ∗ᵥ z ∈ latA V₀ M P

theorem **latO_extR** : ∀ (D : D7) (V₀ A L S : HSp D) (o : OrthDec V₀ A L) (P : Pl D → Prop) (LA : Lat A) (LS : Lat S) (M : Lat V₀), IsSelfDualC A LA → IsSelfDualC S LS → IsSelfDualC V₀ M → (∀ v, P v → ∃ LL, OrthAt o M LA LL v) → ∀ u ∈ latO S A (LS : Set _) (LA : Set _) P, (o.resT S).symm (u, 0) ∈ latO S V₀ (LS : Set _) (M : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroRightMemCoreLatticeStatement

theorem **_root_.GlobalTheory.T.extendByZero_right_mem_core_lattice** : ∀ (D : D7) (V₀ A L S : HSp D) (o : OrthDec V₀ A L) (P : Pl D → Prop) (LA : Lat A) (LS : Lat S) (M : Lat V₀), IsSelfDualC A LA → IsSelfDualC S LS → IsSelfDualC V₀ M → (∀ v, P v → ∃ LL, OrthAt o M LA LL v) → ∀ u ∈ latO S A (LS : Set _) (LA : Set _) P, (o.resT S).symm (u, 0) ∈ latO S V₀ (LS : Set _) (M : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroSourceTensorColumnProduct

theorem **isHalf_mul_right** {e : AE D} {l m n : ℕ} {u : Mx D l m} (hu : IsHalf e u) (M : Mx D m n) : IsHalf e (u ∗ M)

theorem **isHalf_add_of** {e : AE D} {m n : ℕ} {u v : Mx D m n} (hu : IsHalf e u) (hv : IsHalf e v) : IsHalf e (u + v)

theorem **isHalf_resS_iff** {S A L T : HSp D} (o : OrthDec S A L) (e : AE D) (t : WO S T) : IsHalf e t ↔ IsHalf e (t ∗ o.jA) ∧ IsHalf e (t ∗ o.jB)

theorem **rd_sum_inl** {p : SplitPt D k} {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {S A L T : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n) (hs : IsSumFrame o ζ α α₀) (t : WO S T) : (rd α₀ β t).submatrix id Sum.inl = rd α β (t ∗ o.jA)

theorem **rd_sum_inr** {p : SplitPt D k} {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {S A L T : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n) (hs : IsSumFrame o ζ α α₀) (t : WO S T) : (fun i => rd α₀ β t i (Sum.inr 0)) = zcol β ζ (t ∗ o.jB)

theorem **extFun_tensCol_sum** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n) : IsSumFrame o ζ α α₀ → ∀ (Ψ : Matrix n n' k → ℂ) (a : (n → k) → ℂ), extFun α₀ β (tensCol Ψ a) = o.boxS T (extFun α β Ψ) (GlobalTheory.FinPl.ExtZero.colFun β ζ a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroSourceTensorColumnProductStatement

theorem **_root_.GlobalTheory.T.extendByZero_source_tensor_column_product** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n']

[DecidableEq n'] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n), IsSumFrame o ζ α α₀ → ∀ (Ψ : Matrix n n' k → ℂ) (a : (n → k) → ℂ), extFun α₀ β (tensCol Ψ a) = o.boxS T (extFun α β Ψ) (GlobalTheory.FinPl.ExtZero.colFun β ζ a)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroTargetTensorRowProduct

theorem **isHalf_mul** {e : AE D} {l m n : ℕ} (M : Mx D l m) {x : Mx D m n} (hx : IsHalf e x) : IsHalf e (M · x)

theorem **isHalf_add** {e : AE D} {m n : ℕ} {x y : Mx D m n} (hx : IsHalf e x) (hy : IsHalf e y) : IsHalf e (x + y)

theorem **isHalf_resT_iff** {S A L T : HSp D} (o : OrthDec S A L) {e : AE D} (t : WO T S) : IsHalf e t ↔ IsHalf e (o.pA · t) ∧ IsHalf e (o.pB · t)

theorem **map_rat_pr** {l m : ℕ} (M : Matrix (Fin l) (Fin m) D.host.E) : (rat M).map p.pr = M.map (p.pr.comp (incA D))

theorem **sumFrame_b'** {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (hζ : IsLineVec L ζ) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (hs : IsSumFrame o ζ α α₀) : α₀.b' = fromRows (α.b' ∗ o.pA.map p.pr) (replicateRow (Fin 1) ((p.pr (incA D (star ζ ᵥ· L.gram ∗ᵥ ζ)))⁻¹ • ((fun j => p.pr (incA D ((star ζ ᵥ* L.gram) j))) ᵥ* o.pB.map p.pr)))

theorem **rd_sumFrame** {S A L T : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (hζ : IsLineVec L ζ) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n) (hs : IsSumFrame o ζ α α₀) (t : WO T S) : rd β α₀ t = fromRows (rd β α (o.pA · t)) (replicateRow (Fin 1) (zrow β ζ (o.pB · t)))

theorem **extFun_tensRow** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (hζ : IsLineVec L ζ) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n) (hs : IsSumFrame o ζ α α₀) (Ψ : Matrix n' n k → ℂ) (b : (n → k) → ℂ) : extFun β α₀ (tensRow Ψ b) = o.boxT T (extFun β α Ψ) (GlobalTheory.FinPl.ExtZero.rowFun β ζ b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtendByZeroTargetTensorRowProductStatement

theorem **_root_.GlobalTheory.T.extendByZero_target_tensor_row_product** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n), IsSumFrame o ζ α α₀ → ∀ (Ψ : Matrix n' n k → ℂ) (b : (n → k) → ℂ), extFun β α₀ (tensRow Ψ b) = o.boxT T (extFun β α Ψ) (GlobalTheory.FinPl.ExtZero.rowFun β ζ b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionAdeleRealComponentsCmType

theorem **eq_zero_of_map_eq_zero** {F E A : Type} [Field F] [Field E] [Algebra F E] [CommRing A] [Algebra F A] {I : Type} {B : I → Type} [∀ i, CommRing (B i)] [∀ i, Algebra F (B i)] (f : ∀ i, A →ₐ[F] B i) (hf : ∀ a, (∀ i, f i a = 0) → a = 0) (z : A ⊗[F] E) (hz : ∀ i, Algebra.TensorProduct.map (f i) (AlgHom.id F E) z = 0) : z = 0

theorem **compInf_eq_map** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (w : InfinitePlace F) (z : AF F ⊗[F] E) : compInf F E w z = Algebra.TensorProduct.map (GlobalTheory.AutFunCompare.evW w) (AlgHom.id F E) (Algebra.TensorProduct.map (fstA F) (AlgHom.id F E) z)

theorem **compInf_eq** (w : InfinitePlace D.host.F) (z : AE D) : compInf D.host.F D.host.E w z = Algebra.TensorProduct.map (GlobalTheory.AutFunCompare.evW w) (AlgHom.id D.host.F D.host.E) (infPart D z)

theorem **compInf_zero_iff** (z : AE D) : (∀ w, compInf D.host.F D.host.E w z = 0) ↔ infPart D z = 0

theorem **theta_zero_iff** {θ : InfFam D} (hθ : IsInfFam θ) (y : Ainf D.host.F ⊗[D.host.F] D.host.E) : (∀ τ, θ τ y = 0) ↔ y = 0

theorem **infPart_conj** (z : AE D) : infPart D (cA D z) = sigR (Ainf D.host.F) D.sp.cm.sig (infPart D z)

theorem **eq_zero_of_parts** (z : AE D) (h1 : infPart D z = 0) (h2 : finPart D z = 0) : z = 0

theorem **real_components** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → (∀ z : AE D, (∀ w, compInf D.host.F D.host.E w z = 0) ↔ ∀ τ, θ τ (infPart D z) = 0) ∧ (∀ z τ, θ τ (infPart D (cA D z)) = conj (θ τ (infPart D z))) ∧ (∀ x τ, θ τ (infPart D (incA D x)) = tauE D τ x) ∧ (∀ z : AE D, (∀ τ, θ τ (infPart D z) = 0) → finPart D z = 0 → z = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionAdeleRealComponentsCmTypeStatement

theorem **_root_.GlobalTheory.T.extensionAdele_realComponents_cmType** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → (∀ z : AE D, (∀ w, compInf D.host.F D.host.E w z = 0) ↔ ∀ τ, θ τ (infPart D z) = 0) ∧ (∀ z τ, θ τ (infPart D (cA D z)) = conj (θ τ (infPart D z))) ∧ (∀ x τ, θ τ (infPart D (incA D x)) = tauE D τ x) ∧ (∀ z : AE D, (∀ τ, θ τ (infPart D z) = 0) → finPart D z = 0 → z = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionByZeroAtSplitPlacesProof

theorem **vac_zero'** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} (P : Pl D → Prop) : Vac ψ S T LS LT P 0

theorem **vac_at'** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P : Pl D → Prop} {v : Pl D} {f : WO S T → ℂ} (hv : P v) (h : Vac ψ S T LS LT P f) : Vac ψ S T LS LT (· = v) f

theorem **idAt_mono'** {P Q : Pl D → Prop} (h : ∀ v, P v → Q v) : idAt D P ≤ idAt D Q

theorem **combAt_extS** {P : Pl D → Prop} {S A L : HSp D} (o : OrthDec S A L) (T : HSp D) {c : WO A T →₀ ℂ} (hc : CombAt P c) : CombAt P (extS o T c)

theorem **combAt_extT** {P : Pl D → Prop} {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) {c : WO S A →₀ ℂ} (hc : CombAt P c) : CombAt P (extT o S c)

theorem **comb_eq_sum** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) (F : WO S T → ℂ) : comb ψ S T c F = c.sum (fun u a => a • heisO ψ S T u F)

theorem **comb_extS_slice** (ψ : AddChar (AF D.host.F) ℂ) {S A L T : HSp D} (o : OrthDec S A L) (c : WO A T →₀ ℂ) (F : WO S T → ℂ) (w : WO A T) (y : WO L T) : comb ψ S T (extS o T c) F ((o.resS T).symm (w, y)) = comb ψ A T c (fun w' => F ((o.resS T).symm (w', y))) w

theorem **comb_extT_slice** (ψ : AddChar (AF D.host.F) ℂ) {S A L T : HSp D} (o : OrthDec S A L) (c : WO T A →₀ ℂ) (F : WO T S → ℂ) (w : WO T A) (y : WO T L) : comb ψ T S (extT o T c) F ((o.resT T).symm (w, y)) = comb ψ T A c (fun w' => F ((o.resT T).symm (w', y))) w

theorem **overAll_sb** {α : ∀ i, Frame (p i) S n'} {β : ∀ i, Frame (p i) T n} {LS : Lat S} {LT : Lat T} {Ψ : ι → Matrix n n' k → ℂ} : ∀ (l : List ι) (f f' : WO S T → ℂ), OverAll ψ α β LS LT Ψ l f f' → ∀ i ∈ l, Ψ i ∈ SB (Matrix n n' k)

theorem **chainOp_vac** {LS : Lat S} {LT : Lat T} (e : ι → WO S T →₀ ℂ) (hpres : ∀ i j, i ≠ j → ∀ g, Vac ψ S T LS LT (· = (p j).v) g → Vac ψ S T LS LT (· = (p j).v) (comb ψ S T (e i) g)) : ∀ (l : List ι) (f : WO S T → ℂ), (∀ i, Vac ψ S T LS LT (· = (p i).v) f) → ∀ j, j ∉ l → Vac ψ S T LS LT (· = (p j).v) (chainOp ψ S T e l f)

theorem **overAll_chain** (α : ∀ i, Frame (p i) S n') (β : ∀ i, Frame (p i) T n) (LS : Lat S) (LT : Lat T) (Φ : ι → Matrix n n' k → ℂ) (e : ι → WO S T →₀ ℂ) (hE : ∀ i f₀, Vac ψ S T LS LT (· = (p i).v) f₀ → Over ψ (α i) (β i) LS LT f₀ (Φ i) (comb ψ S T (e i) f₀)) (hpres : ∀ i j, i ≠ j → ∀ g, Vac ψ S T LS LT (· = (p j).v) g → Vac ψ S T LS LT (· = (p j).v) (comb ψ S T (e i) g)) : ∀ (l : List ι), l.Nodup → ∀ f, (∀ i, Vac ψ S T LS LT (· = (p i).v) f) → OverAll ψ α β LS LT Φ l f (chainOp ψ S T e l f)

theorem **pres_of_allComb** (hAC : AllComb D) {I : List ι} {P : Pl D → Prop} (hP : Places p I P) (hψ : IsStdChar D ψ (-1)) (α : ∀ i, Frame (p i) S n') (β : ∀ i, Frame (p i) T n) {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (e : ι → WO S T →₀ ℂ) (he : ∀ i, CombAt (· = (p i).v) (e i)) : ∀ i j, i ≠ j → ∀ g, Vac ψ S T LS LT (· = (p j).v) g → Vac ψ S T LS LT (· = (p j).v) (comb ψ S T (e i) g)

theorem **chainOp_sliceS** (ψ : AddChar (AF D.host.F) ℂ) {ι : Type} {S A L T : HSp D} (o : OrthDec S A L) (d : ι → WO A T →₀ ℂ) (y : WO L T) : ∀ (l : List ι) (h : WO S T → ℂ) (w : WO A T), chainOp ψ S T (fun i => extS o T (d i)) l h ((o.resS T).symm (w, y)) = chainOp ψ A T d l (fun w' => h ((o.resS T).symm (w', y))) w

theorem **chainOp_sliceT** (ψ : AddChar (AF D.host.F) ℂ) {ι : Type} {S A L T : HSp D} (o : OrthDec S A L) (d : ι → WO T A →₀ ℂ) (y : WO T L) : ∀ (l : List ι) (h : WO T S → ℂ) (w : WO T A), chainOp ψ T S (fun i => extT o T (d i)) l h ((o.resT T).symm (w, y)) = chainOp ψ T A d l (fun w' => h ((o.resT T).symm (w', y))) w

theorem **vac_sliceS** {ψ : AddChar (AF D.host.F) ℂ} {S A L T : HSp D} (o : OrthDec S A L) {P : Pl D → Prop} {LA : Lat A} {LS : Lat S} {LT : Lat T} (hA : IsSelfDualC A LA) (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (hO : ∀ v, P v → ∃ LL, OrthAt o LS LA LL v) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) (y : WO L T) : Vac ψ A T LA LT P (fun w => f ((o.resS T).symm (w, y)))

theorem **vac_sliceT** {ψ : AddChar (AF D.host.F) ℂ} {S A L T : HSp D} (o : OrthDec S A L) {P : Pl D → Prop} {LA : Lat A} {LS : Lat S} {LT : Lat T} (hA : IsSelfDualC A LA) (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (hO : ∀ v, P v → ∃ LL, OrthAt o LS LA LL v) {f : WO T S → ℂ} (hf : Vac ψ T S LT LS P f) (y : WO T L) : Vac ψ T A LT LA P (fun w => f ((o.resT T).symm (w, y)))

theorem **ext_all** : ∀ D : D7, GlobalTheory.FinPl.ExtAll D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionByZeroAtSplitPlacesProofStatement

theorem **_root_.GlobalTheory.T.extensionByZeroAtSplitPlaces_proof** : ∀ D : D7, GlobalTheory.FinPl.ExtAll D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionByZeroMapsCompletedHomLattice

theorem **mxAt_mul_left** {P : Pl D → Prop} {a b c : ℕ} (y : Mx D a b) {x : Mx D b c} (hx : MxAt P x) : MxAt P (y · x)

theorem **symplO_resT_symm_left** {S A L T : HSp D} (o : OrthDec S A L) (a : WO T A) (t : WO T S) : symplO T S ((o.resT T).symm (a, 0)) t = symplO T A a (o.pA · t)

theorem **GlobalTheory.FinPl.ExtZero.extZero_mem_latO_target** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), (∀ u : WO T A, MxAt (· = p.v) u → (u ∈ latO T A (LT : Set _) (LA : Set _) (· = p.v) ↔ ∀ u' ∈ latO T A (LT : Set _) (LA : Set _) (· = p.v), ψ (symplO T A u u') = 1)) → (∀ u : WO T S, MxAt (· = p.v) u → (u ∈ latO T S (LT : Set _) (LS : Set _) (· = p.v) ↔ ∀ u' ∈ latO T S (LT : Set _) (LS : Set _) (· = p.v), ψ (symplO T S u u') = 1)) → halfInd p.e (latO T S (LT : Set _) (LS : Set _) (· = p.v)) = o.boxT T (halfInd p.e (latO T A (LT : Set _) (LA : Set _) (· = p.v))) (GlobalTheory.FinPl.ExtZero.rowF

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionByZeroMapsCompletedHomLatticeStatement

theorem **_root_.GlobalTheory.T.extension_by_zero_maps_completed_hom_lattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), (∀ u : WO T A, MxAt (· = p.v) u → (u ∈ latO T A (LT : Set _) (LA : Set _) (· = p.v) ↔ ∀ u' ∈ latO T A (LT : Set _) (LA : Set _) (· = p.v), ψ (symplO T A u u') = 1)) → (∀ u : WO T S, MxAt (· = p.v) u → (u ∈ latO T S (LT : Set _) (LS : Set _) (· = p.v) ↔ ∀ u' ∈ latO T S (LT : Set _) (LS : Set _) (· = p.v), ψ (symplO T S u u') = 1)) → halfInd p.e (latO T S (LT : Set _) (LS : Set _) (· = p.v)) = o.boxT T (halfInd p.e (latO T A (LT : Set _) (LA : Set _) (· = p.v))) (GlobalTheory.Fi

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtensionFunctionIncZero

theorem **mul_colRow** {S T T' : HSp D} (M : Mx D T'.rank T.rank) (r : Fin S.rank → AE D) (y : Fin T.rank → AE D) : HMul.hMul M (colRow S T r y) = colRow S T' r (M *ᵥ y)

theorem **gen_eq_colRow** {S T : HSp D} (α : AF D.host.F) (a : D.host.E) (s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) : (Matrix.of fun i j => α ⊗ₜ[D.host.F] (a * t i * ∑ k, star (s k) * S.gram k j) : WO S T) = colRow S T (fun j => incA D (a * ∑ k, star (s k) * S.gram k j)) (fun i => α ⊗ₜ[D.host.F] (1 * t i))

theorem **colRow_mem_latO** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Lat T} {P : GlobalTheory.FinPl.Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {s : Fin S.rank → D.host.E} (hs : s ∈ LS) {y : Fin T.rank → AE D} (hy : y ∈ latV T LT P) : colRow S T (fun j => incA D (a * ∑ k, star (s k) * S.gram k j)) y ∈ latO S T LS (LT : Set _) P

theorem **mul_mem_latO** {S T T' : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Lat T} {LT' : Lat T'} {P : GlobalTheory.FinPl.Pl D → Prop} (M : Mx D T'.rank T.rank) (hM : ∀ y ∈ latV T LT P, M *ᵥ y ∈ latV T' LT' P) {w : WO S T} (hw : w ∈ latO S T LS (LT : Set _) P) : HMul.hMul M w ∈ latO S T' LS (LT' : Set _) P

theorem **isHalf_mul** {m n l : ℕ} {e : AE D} (M : Mx D m l) {u : Mx D l n} (h : IsHalf e u) : IsHalf e (HMul.hMul M u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtraColumnRangeMembership

theorem **lift_lev_pt** (hl : R.toLevCh.Is A.μ0) (hθ : IsInfFam R.θ) (t : T) (k : ℕ) (hk : k < D.R.m) {V : HSp D} (hV : V.SigOK) (hr : V.rank = D.R.r + (k + 1) * D.R.t) {S : D.what₂.Tau → SPar} (M : DSModel V S) {Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)} {β : GL (Fin V.rank) (Lev.EAf D)} {N : ℕ} (hβ : IsSDFrame V β Λ N) {om : Om (Yc D k) V} (hom : IsWeilDatumO R.ψ (Yc D k) V (Lift.chrStep (R.toLevCh.mu t) R.eta (k + 1)).1 (Lift.chrStep (R.toLevCh.mu t) R.eta (k + 1)).2 om) {f₀ f f' : WO (Yc D k) V → ℂ} (hf : IsStepVecOf R.ψ R.θ (Yc D k) V (cart (R.MY k)) (cart M) (R.ΛY k) Λ (R.βY k) β (R.datYt t k R.triv) (R.chi (k + 1)) (R.Q t) (R.xiR t) f₀ f) (hf' : IsStepVecOf R.ψ R.θ (Yc D k) V (cart (R.MY k)) (cart M) (R.ΛY k) Λ (R.βY k) β (R.datYt t k P) (R.chi (k + 1)) (R.Q t) (R.xiR t) f₀ f') (ν : Measure (GA D V)) (ℱ : Set (GA D V)) (F : GA D V → ℂ) (φ : PolF D V.rank (Yc D k).rank) (hi :

theorem **lift_ioX_pt** (hl : R.toLevCh.Is A.μ0) (hm : R.IsModels) (hc : R.IsChain) (t : T) (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) (F : LinearMap.range (R.cX.toMap R.prX ∘ₗ (R.HX t).e)) (hF : ∀ κ ∈ levS (Yc D (D.R.m - 1)) (R.βY (D.R.m - 1)) (R.datYt t (D.R.m - 1) P), ∀ φ x g, (R.ioX t F).1 (R.ItX φ x) (g * κ) = levChi (Yc D (D.R.m - 1)) (R.βY (D.R.m - 1)) (R.datYt t (D.R.m - 1) P) κ * (R.ioX t F).1 (R.ItX φ x) g) {v : WO (Yc D (j + 1)) D.sp.X' → ℂ} (hv : IsStepVecOf R.ψ R.θ (Yc D (j + 1)) D.sp.X' (cart (R.MY (j + 1))) (cart R.MX) (R.ΛY (j + 1)) R.ΛX (R.βY (j + 1)) R.βX (R.datYt t (j + 1) P) (R.chi D.R.m) (R.Q t) (R.xiR t) (vacT R j h) v) (Q x) : liftS (omT R t j

h) (R.νX t) (R.fdX t) (F.1 x) (pureTensor0 R.ψ R.θ (Yc D (j + 1)) D.sp.X' (cart (R.MY (j + 1))) (cart R.MX) (onb (R.MY (j + 1))) (onb R.MX) v Q) = (Nat.card P.H : ℂ) • (ioXT R t j h F).1 (itXT R j h Q x)

theorem **ioXT_levelP** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (g : GlobalTheory.Fam.Geo D) t j (h : j + 2 * 0 + 2 = D.R.m) (F : CXP (pk W) (lvP W) t), ∃ h' : F.1 ∈ LinearMap.range ((pk W).cX.toMap (pk W).prX ∘ₗ ((pk W).HX t).e), (GlobalTheory.Published.CuspRapidDecay → (ioXT (pk W) t j h (F.1, h')).1 ∈ CP (pk W) (lvP W) t (j + 1)) ∧ ∀ v, IsStepVecOf (pk W).ψ (pk W).θ (Yc D (j + 1)) D.sp.X' (cart ((pk W).MY (j + 1))) (cart (pk W).MX) ((pk W).ΛY (j + 1)) (pk W).ΛX ((pk W).βY (j + 1)) (pk W).βX ((pk W).datYt t (j + 1) (lvP W)) ((pk W).chi D.R.m) ((pk W).Q t) ((pk W).xiR t) (vacT (pk W) j h) v → ∀ Q x, liftS (omT (pk W) t j h) ((pk W).νX t) ((pk W).fdX t) (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtraColumnRangeMembershipStatement

theorem **_root_.GlobalTheory.T.extra_column_range_membership** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (g : GlobalTheory.Fam.Geo D) t j (h : j + 2 * 0 + 2 = D.R.m) (F : CXP (pk W) (lvP W) t), ∃ h' : F.1 ∈ LinearMap.range ((pk W).cX.toMap (pk W).prX ∘ₗ ((pk W).HX t).e), (GlobalTheory.Published.CuspRapidDecay → (ioXT (pk W) t j h (F.1, h')).1 ∈ CP (pk W) (lvP W) t (j + 1)) ∧ ∀ v, IsStepVecOf (pk W).ψ (pk W).θ (Yc D (j + 1)) D.sp.X' (cart ((pk W).MY (j + 1))) (cart (pk W).MX) ((pk W).ΛY (j + 1)) (pk W).ΛX ((pk W).βY (j + 1)) (pk W).βX ((pk W).datYt t (j + 1) (lvP W)) ((pk W).chi D.R.m) ((pk W).Q t) ((pk W).xiR t) (vacT (pk W) j h) v → ∀ Q x, liftS (omT (pk W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtraProjectionLevelCharacter

theorem **eq_mul_of_inv_mul** {a b c d : ℂ} (ha : a ≠ 0) (h : a⁻¹ * b = c * d) : b = a * c * d

theorem **mul_eq_of** {a c d u v p q : ℂ} (h1 : u = a * p) (h2 : v = c * d * q) : u * v = a * (c * d) * (p * q)

theorem **const_eq** {a b c I : ℂ} (hb : b ≠ 0) : a * b⁻¹ * (b * c) * I = c * a * I

theorem **ioX_levChi_gen** {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (P : LevelP R.Pg D.R.r ℂ) (hL : R.toLevCh.Is A.μ0) (hm : R.IsModels) (hc : R.IsChain) (t : T) (F : LinearMap.range (R.cX.toMap R.prX ∘ₗ (R.HX t).e)) (hF : F.1 ∈ CXP R P t) (κ : GA D (Yc D (D.R.m − 1))) (hκ : κ ∈ levS (Yc D (D.R.m − 1)) (R.βY (D.R.m − 1)) (R.datYt t (D.R.m − 1) P)) (φ x g) : (R.ioX t F).1 (R.ItX φ x) (g * κ) = levChi (Yc D (D.R.m − 1)) (R.βY (D.R.m − 1)) (R.datYt t (D.R.m − 1) P) κ * (R.ioX t F).1 (R.ItX φ x) g

theorem **ioX_levChi** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t (F : LinearMap.range ((pk W).cX.toMap (pk W).prX ∘ₗ ((pk W).HX t).e)), F.1 ∈ CXP (pk W) (lvP W) t → ∀ κ ∈ levS (Yc D (D.R.m − 1)) ((pk W).βY (D.R.m − 1)) ((pk W).datYt t (D.R.m − 1) (lvP W)), ∀ φ x g, ((pk W).ioX t F).1 ((pk W).ItX φ x) (g * κ) = levChi (Yc D (D.R.m − 1)) ((pk W).βY (D.R.m − 1)) ((pk W).datYt t (D.R.m − 1) (lvP W)) κ * ((pk W).ioX t F).1 ((pk W).ItX φ x) g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ExtraProjectionLevelCharacterStatement

theorem **_root_.GlobalTheory.T.extra_projection_level_character** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ t (F : LinearMap.range ((pk W).cX.toMap (pk W).prX ∘ₗ ((pk W).HX t).e)), F.1 ∈ CXP (pk W) (lvP W) t → ∀ κ ∈ levS (Yc D (D.R.m − 1)) ((pk W).βY (D.R.m − 1)) ((pk W).datYt t (D.R.m − 1) (lvP W)), ∀ φ x g, ((pk W).ioX t F).1 ((pk W).ItX φ x) (g * κ) = levChi (Yc D (D.R.m − 1)) ((pk W).βY (D.R.m − 1)) ((pk W).datYt t (D.R.m − 1) (lvP W)) κ * ((pk W).ioX t F).1 ((pk W).ItX φ x) g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamGen1

theorem **famGen1** : CorankOneIsEquivalenceOnRow → FamGen1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyLaws

theorem **family_laws** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (hX : B.fields.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)), (I.J.toRec B (I.cols l) Z.W Z.V Z.S Z.N Z.G).EqP I.chi ∧ (I.J.toRec B (I.cols l) Z.W Z.V Z.S Z.N Z.G).VecP I.chi

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyLinkDownward

theorem **topOf_eq** (W : RealPack ex) (t : W.P.T) (F : Fin ((W.Q.F.fc (cxW W).zs).Df1.nV t) → D.what₂.LL) : topOf W t F = (pk W).cX.toMap (pk W).prX (((pk W).HX t).e (xOf W t F))

theorem **dn_eq** {W : RealPack ex} {u : D.what₂.Word} (f : FamCh W u) (t : W.P.T) (hr : (f.row t).Is) : ∀ k, k + 1 ≤ D.R.m → ∀ (F : (f.t t).CC k) (F' : (pk W).CC t k), F.1 = F'.1 → ((f.t t).dn k F).1 = (Lift.dnFrom ((pk W).CC t) ((pk W).ioY t) k F').1

theorem **dnFrom_heq** (W : RealPack ex) (t : W.P.T) {k : ℕ} (hk : D.R.m − 1 = k) (X : (pk W).CC t k) (X' : (pk W).CC t (D.R.m − 1)) (e : HEq X X') : (Lift.dnFrom ((pk W).CC t) ((pk W).ioY t) k X).1 = (Lift.dnFrom ((pk W).CC t) ((pk W).ioY t) (D.R.m − 1) X').1

theorem **app_heq** (W : RealPack ex) (t : W.P.T) {M : Type} [AddCommGroup M] [Module ℂ M] {k k' : ℕ} (p : k = k') (g : M →ₗ[ℂ] (pk W).CC t k) (z : M) : HEq ((Eq.rec (motive

theorem **down_eq** {W : RealPack ex} {u : D.what₂.Word} (f : FamCh W u) (t : W.P.T) (hr : (f.row t).Is) (n : ℕ) (h : 2 * n + 2 * 0 + 2 = D.R.m) (G : (f.t t).CT) (y : ((pk W).HX t).H) (hG : G.1 = (pk W).cX.toMap (pk W).prX (((pk W).HX t).e y)) : down f n h t G = (pk W).thDn t y

theorem **lift_postOn_fixed** (W : RealPack ex) (T : SmT ((pk W).MY 0) →ₗ[ℂ] (GA D (Yc D 0) → ℂ)) (hT : ∀ v : ((pk W).MY 0).Vec, (pk W).e0 (T (PiTensorProduct.tprod ℂ v)) = T (PiTensorProduct.tprod ℂ v)) : PiTensorProduct.lift (postOn ((pk W).MY 0) (pk W).e0 (PiTensorProduct.lift.symm T)) = T

theorem **GlobalTheory.MiTop.linkDown_of_rowOK** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ f : FamCh W u, f.RowOK → LinkDown f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyLinkDownwardStatement

theorem **_root_.GlobalTheory.T.family_link_downward** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ f : FamCh W u, f.RowOK → LinkDown f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyLinkOnto

theorem **stepY_is** {W : RealPack ex} {u : D.what₂.Word} (f : FamCh W u) (hP : PkHyp W) (hf : f.RowOK) (j : ℕ) (hj : j + 1 < D.R.m) : (stepY f j).Is (j + 1)

theorem **stepX_is** {W : RealPack ex} {u : D.what₂.Word} (f : FamCh W u) (hP : PkHyp W) (hf : f.RowOK) (j : ℕ) (h : j + 2 * 0 + 2 = D.R.m) : (stepX f j h).Is (j + 1 + 1)

theorem **reach_dn** {W : RealPack ex} {u : D.what₂.Word} (f : FamCh W u) (hP : PkHyp W) (hf : f.RowOK) : ∀ k, k + 1 ≤ D.R.m → Reach (com W) (stageY W 0) k (stageY W k) ((f.t W.P.A3.A.μ0).dn k) | 0, _ => Reach.zero | k + 1, hk => Reach.step (reach_dn f hP hf k (by omega)) (GlobalTheory.T.base_chain_stage_Y_valid W hP (k + 1) (by omega)) (stepY f k) (stepY_is f hP hf k (by omega)) theorem linkOnto_of_ontoAll : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → GlobalTheory.ChainOnto.OntoAll D → ∀ f : FamCh W u, f.RowOK → LinkOnto f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyLinkOntoStatement

theorem **_root_.GlobalTheory.T.family_link_onto** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → GlobalTheory.ChainOnto.OntoAll D → ∀ f : FamCh W u, f.RowOK → LinkOnto f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyRowsColumnsValid

theorem **main** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ f : FamCh W u, f.RowOK → f.c.Is ∧ (f.ColOK → ∀ t, (f.t t).Is)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FamilyRowsColumnsValidStatement

theorem **_root_.GlobalTheory.T.family_rows_columns_valid** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ f : FamCh W u, f.RowOK → f.c.Is ∧ (f.ColOK → ∀ t, (f.t t).Is)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FdegHCRowsMulSuperfactorialLeVandermonde

theorem **pairs_append** {p q : ℕ} (g : Fin (p + q) → Fin (p + q) → ℝ) : ∏ a : Fin (p + q), ∏ b ∈ univ.filter (fun b => a < b), g a b = (∏ i : Fin p, ∏ j ∈ univ.filter (fun j => i < j), g (Fin.castAdd q i) (Fin.castAdd q j)) * (∏ i : Fin p, ∏ j : Fin q, g (Fin.castAdd q i) (Fin.natAdd p j)) * (∏ i : Fin q, ∏ j ∈ univ.filter (fun j => i < j), g (Fin.natAdd p i) (Fin.natAdd p j))

theorem **pairs_cast** {n n' : ℕ} (h : n = n') (f : Fin n' → Fin n' → ℝ) : ∏ a : Fin n, ∏ b ∈ univ.filter (fun b => a < b), f (Fin.cast h a) (Fin.cast h b) = ∏ a : Fin n', ∏ b ∈ univ.filter (fun b => a < b), f a b

theorem **den_eq** (n : ℕ) : ∏ a : Fin n, ∏ b ∈ univ.filter (fun b : Fin n => a < b), (((b : ℕ) : ℝ) - ((a : ℕ) : ℝ)) = Formalism.MainNotions.StandardLattice.facProd n

theorem **facProd_pos** (n : ℕ) : 0 < Formalism.MainNotions.StandardLattice.facProd n

theorem **anti_of** {K n : ℕ} (hn : K + 1 = n) (u : Fin n → ℚ) (v : Fin K → ℚ) (h1 : ∀ i : Fin K, v i < u ⟨i.1, by omega⟩) (h2 : ∀ i : Fin K, u ⟨i.1 + 1, by omega⟩ < v i) : StrictAnti u

theorem **mono_of** {m : ℕ} (f : Fin m → ℚ) (h : ∀ j : Fin m, ∀ hj : j.1 + 1 < m, f j < f ⟨j.1 + 1, hj⟩) : StrictMono f

theorem **vand_cast** {K : ℕ} (u : Fin (K + 1) → ℝ) : Formalism.MainNotions.StandardLattice.vand u = ∏ i : Fin K, ∏ j : Fin K, (if i ≤ j then u (Fin.castSucc i) - u (Fin.succ j) else 1)

theorem **interlace_bound** {K n : ℕ} (hn : K + 1 = n) (u : Fin n → ℝ) (v : Fin K → ℝ) (h1 : ∀ i : Fin K, v i < u ⟨i.1, by omega⟩) (h2 : ∀ i : Fin K, u ⟨i.1 + 1, by omega⟩ < v i) (h3 : ∀ i : Fin K, 1 ≤ u ⟨i.1, by omega⟩ - u ⟨i.1 + 1, by omega⟩) : ∏ a : Fin K, ∏ b ∈ univ.filter (fun b => a < b), |v a - v b| ≤ Formalism.MainNotions.StandardLattice.vand u

theorem **lamSH_congr** (R : Ranks) (P Q : HCParam R) (hv : Q.v = P.v) (hb : Q.b = P.b) : Formalism.MainNotions.S.lamSH R P = Formalism.MainNotions.S.lamSH R Q

theorem **lamS_left** (R : Ranks) (P : HCParam R) (h : (R.k - 1) + R.m = R.r) (i : Fin (R.k - 1)) : Formalism.MainNotions.S.lamS R P (Fin.cast h (Fin.castAdd R.m i)) = (P.v i : ℝ)

theorem **lamS_right** (R : Ranks) (P : HCParam R) (h : (R.k - 1) + R.m = R.r) (j : Fin R.m) : Formalism.MainNotions.S.lamS R P (Fin.cast h (Fin.natAdd (R.k - 1) j)) = -(P.b j : ℝ)

theorem **count_pairs** (m : ℕ) : ∑ j : Fin m, (univ.filter (fun j' : Fin m => j < j')).card = m * (m - 1) / 2

theorem **fdeg_rows_le** : ∀ (R : Ranks) (P Q : HCParam R), Q.v = P.v → Q.b = P.b → ∀ B : ℝ, 1 ≤ B → (∀ i j, ((P.v i : ℚ) : ℝ) + ((P.b j : ℚ) : ℝ) ≤ B) → (∀ j j', ((P.b j' : ℚ) : ℝ) - ((P.b j : ℚ) : ℝ) ≤ B) → Formalism.QuantumDeterminants.fdegHC (Formalism.MainNotions.S.lamSH R P) * Formalism.MainNotions.StandardLattice.facProd R.r ≤ B ^ ((R.k - 1) * R.m + R.m * (R.m - 1) / 2) * Formalism.MainNotions.StandardLattice.vand (fun i => ((Q.u i : ℚ) : ℝ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FdegHCRowsMulSuperfactorialLeVandermondeStatement

theorem **_root_.GlobalTheory.T.fdegHC_rows_mul_superfactorial_le_vandermonde** : ∀ (R : Ranks) (P Q : HCParam R), Q.v = P.v → Q.b = P.b → ∀ B : ℝ, 1 ≤ B → (∀ i j, ((P.v i : ℚ) : ℝ) + ((P.b j : ℚ) : ℝ) ≤ B) → (∀ j j', ((P.b j' : ℚ) : ℝ) - ((P.b j : ℚ) : ℝ) ≤ B) → Formalism.QuantumDeterminants.fdegHC (Formalism.MainNotions.S.lamSH R P) * Formalism.MainNotions.StandardLattice.facProd R.r ≤ B ^ ((R.k - 1) * R.m + R.m * (R.m - 1) / 2) * Formalism.MainNotions.StandardLattice.vand (fun i => ((Q.u i : ℚ) : ℝ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinMEqOneIffMemGinfC

theorem **finM_eq_one_iff_mem_GinfC** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (g : GA D V), finM V hn g = 1 ↔ g ∈ GinfC V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinPartIdemSumMulInvMem

theorem **finPart_idemSum_mul_inv_mem** : ∀ (D : D7) (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ t : 𝒪 D.host.E, (∀ Q ∈ SS, t ∉ Q.asIdeal) → finPart D (e * incA D ((t : D.host.E))⁻¹) ∈ intAE D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinRatMemLatOOfPerO

theorem **finRat_mem_latO_of_perO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)), ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N → (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N' → ∀ x ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N', GlobalTheory.VacUnique.finRat S T ((LinearMap.toMatrix' (x)).reindex (finCongr hT.symm) (finCongr hS.symm)) ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) (fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteAdeleIdempotentOfFMulSelf

theorem **finM_rfl** (V : HSp D) (g : GA D V) : finM V rfl g = (Formalism.L2.Theta.mt V g).map (toEAf D)

theorem **infM_eq** (θ : InfFam D) (V : HSp D) (g : GA D V) (τ : D.what₂.Tau) : infM θ V rfl g τ = (Formalism.L2.Theta.mt V g).map ((θ τ).comp (infPart D))

theorem **infM_rfl** (θ : InfFam D) (V : HSp D) (g : GA D V) (τ : D.what₂.Tau) : infM θ V rfl g τ = infMat D.sp.cm.sig V.gram (θ τ) g

theorem **mt_mul** (V : HSp D) (g h : GA D V) : Formalism.L2.Theta.mt V (g * h) = Formalism.L2.Theta.mt V g * Formalism.L2.Theta.mt V h

theorem **finM_mul** (V : HSp D) (g h : GA D V) : finM V rfl (g * h) = finM V rfl g * finM V rfl h

theorem **infM_mul** (θ : InfFam D) (V : HSp D) (g h : GA D V) (τ) : infM θ V rfl (g * h) τ = infM θ V rfl g τ * infM θ V rfl h τ

theorem **infM_fin** (θ : InfFam D) (V : HSp D) (k : GA D V) (hk : k ∈ GfC V) (τ) : infM θ V rfl k τ = 1

theorem **finM_inf** (V : HSp D) (k : GA D V) (hk : k ∈ GinfC V) : finM V rfl k = 1

theorem **rat_mt** (V : HSp D) (γ : GA D V) (hγ : γ ∈ GQ V) : ∃ γ0 : GL (Fin V.rank) D.host.E, γ0 ∈ UJ (D.sp.cm.sig : D.host.E →+* D.host.E) V.gram ∧ Formalism.L2.Theta.mt V γ = γ0.val.map (incA D)

theorem **rat_finM** (V : HSp D) (γ0 : GL (Fin V.rank) D.host.E) (γ : GA D V) (e : Formalism.L2.Theta.mt V γ = γ0.val.map (incA D)) : finM V rfl γ = (bc (inclE (F

theorem **rat_infM** (θ : InfFam D) (hθ : IsInfFam θ) (V : HSp D) (γ0 : GL (Fin V.rank) D.host.E) (γ : GA D V) (e : Formalism.L2.Theta.mt V γ = γ0.val.map (incA D)) (τ) : infM θ V rfl γ τ = atInf (embT D) γ0.val τ

theorem **finiteAdeleIdempotentOfF_mul_self** : e0 D * e0 D = e0 D

theorem **eps_mul_self** : eps D * eps D = eps D

theorem **eps_mul_one_sub** : eps D * (1 - eps D) = 0

theorem **one_sub_mul_self** : (1 - eps D) * (1 - eps D) = 1 - eps D

theorem **sig_eps** : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR (AF D.host.F) D.sp.cm.sig (eps D) = eps D

theorem **infPart_eps** : infPart D (eps D) = 0

theorem **finPart_eps** : finPart D (eps D) = 1

theorem **toEAf_eps** : toEAf D (eps D) = 1

theorem **PM_mul** {r : ℕ} (M N : Matrix (Fin r) (Fin r) (AE D)) : PM (M * N) = PM M * PM N

theorem **PM_one** {r : ℕ} : PM (1 : Matrix (Fin r) (Fin r) (AE D)) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteAnnihilatorOpenSubgroupInBox

theorem **cA_smul** (r : AF D.host.F) (z : AE D) : cA D (r • z) = r • cA D z

theorem **continuous_cA'** : Continuous (cA D)

theorem **infPart_cA_eq** (a : AE D) : infPart D (cA D a) = sigR (Ainf D.host.F) D.sp.cm.sig (infPart D a)

theorem **infPart_cA'** (a : AE D) (h : infPart D a = 0) : infPart D (cA D a) = 0

theorem **symplO_extX_extY_single** (S S₀ T : HSp D) {ρ : ℕ} (d : HypDec S S₀ ρ) (x : Mx D T.rank ρ) (i : Fin T.rank) (j : Fin ρ) (c : AE D) : symplO S T (d.extX T x) (d.extY T (Matrix.single i j (cA D c))) = Algebra.trace (AF D.host.F) (AE D) (incA D D.host.xi0 * (gA T · x) i j * c)

theorem **map_finPart_mul_map** {n m : ℕ} (M : Mx D n n) (c : AE D) (z : Mx D n m) (hz : ∀ i j, finPart D (z i j) = 0) : (M · z.map (fun w => c * w)).map (finPart D) = 0

theorem **GlobalTheory.AnnihBox.annihilator_inBox** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (U : AddSubgroup (WO S T)), IsOpen (U : Set (WO S T)) → ∃ K : Set (Mx D T.rank ρ), IsCompact K ∧ ∀ x : Mx D T.rank ρ, (∀ a : Mx D T.rank ρ, a.map (infPart D) = 0 → d.extY T a ∈ U → ψ (symplO S T (d.extX T x) (d.extY T a)) = 1) → InBox K x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteAnnihilatorOpenSubgroupInBoxStatement

theorem **_root_.GlobalTheory.T.finiteAnnihilator_open_subgroup_in_box** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (U : AddSubgroup (WO S T)), IsOpen (U : Set (WO S T)) → ∃ K : Set (Mx D T.rank ρ), IsCompact K ∧ ∀ x : Mx D T.rank ρ, (∀ a : Mx D T.rank ρ, a.map (infPart D) = 0 → d.extY T a ∈ U → ψ (symplO S T (d.extX T x) (d.extY T a)) = 1) → InBox K x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteCommCreation

theorem **finite_comm_creation** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (μ : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T μ μ om → ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ f, IsHol θ S T ψ f → IsHol θ S T ψ (om (g, h) f) ∧ ∀ P, IsThetaFunO ψ S T (pureTensorO ψ θ S T (fun _ => 1) (fun _ => 1) (fun _ => 1) (fun _ => 1) f P) ∧ om (g, h) (pureTensorO ψ θ S T (fun _ => 1) (fun _ => 1) (fun _ => 1) (fun _ => 1) f P) = pureTensorO ψ θ S T (fun _ => 1) (fun _ => 1) (fun _ => 1) (fun _ => 1) (om (g, h) f) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteCommCreationStatement

theorem **_root_.GlobalTheory.T.om_pureTensorO_comm** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (μ : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T μ μ om → ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ f, IsHol θ S T ψ f → IsHol θ S T ψ (om (g, h) f) ∧ ∀ P, IsThetaFunO ψ S T (pureTensorO ψ θ S T (fun _ => 1) (fun _ => 1) (fun _ => 1) (fun _ => 1) f P) ∧ om (g, h) (pureTensorO ψ θ S T (fun _ => 1) (fun _ => 1) (fun _ => 1) (fun _ => 1) f P) = pureTensorO ψ θ S T (fun _ => 1) (fun _ => 1) (fun _ => 1) (fun _ => 1) (om (g, h) f) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumBasePointsLawRationalTwistPrimes

theorem **transport_base** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → (∀ (z₁ z₂ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), t ∈ stdHom _ _ → IsHalf A.e t → z₁.map (fun r => A.e ∗ incA (I.rec7 l Z) r) = z₂.map (fun r => A.e ∗ incA (I.rec7 l Z) r) + k • t → A.xi i hi α γ z₁ = A.xi i hi α γ z₂) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) (κ₀ : TC.GA B (B.R.N i)) (κ₀' : TC.GA B (B.R.N (i + 1))), finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ.1 = κ₀.1 → finM _ (rankN (I.rec7 l Z) (i + 1) hi) κ'.1 = κ₀'.1 → ∀ (q : PtAt B.R I.sel (I.tup a) (I.cst a) u i) (q' : PtAt B.R I.sel (I.tup a) (I.cs

theorem **law_read** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → (∀ (z₁ z₂ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), t ∈ stdHom _ _ → IsHalf A.e t → z₁.map (fun r => A.e ∗ incA (I.rec7 l Z) r) = z₂.map (fun r => A.e ∗ incA (I.rec7 l Z) r) + k • t → A.xi i hi α γ z₁ = A.xi i hi α γ z₂) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ), (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ.1 : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumBasePointsLawRationalTwistPrimesStatement

theorem **_root_.GlobalTheory.T.finiteDatum_basePoints_law_rational_twistPrimes** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → (∀ (z₁ z₂ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), t ∈ stdHom _ _ → IsHalf A.e t → z₁.map (fun r => A.e ∗ incA (I.rec7 l Z) r) = z₂.map (fun r => A.e ∗ incA (I.rec7 l Z) r) + k • t → A.xi i hi α γ z₁ = A.xi i hi α γ z₂) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ), (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ.1 : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumFactorsOverTwistPrimesLocal

theorem **xiAt_eq_prod** (i : ℕ) (hi : i + 1 ≤ B.R.m) (α : Ch₀ i) (γ : Ch₀ (i + 1)) (x : Formalism.L2.Col.HomE B.fields.E (B.R.N i) (B.R.N (i + 1))) : xiAt I.Ch I.sel (I.tup a) (I.cst a) (I.idx a) u i (A.b0 i α) (A.b0 (i + 1) γ) (A.p0 i α) (A.p0 (i + 1) γ) x = Formalism.L2.Col.phiT B.fields.K (A.p0 i α).1.1.deco.1 (A.p0 (i + 1) γ).1.1.deco.1 x ∗ ∏ P : BadPr I a, locXi A i α γ P x

theorem **xi_eq_xiOff_mul** (i : ℕ) (hi : i + 1 ≤ B.R.m) (α : Ch₀ i) (γ : Ch₀ (i + 1)) (z : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) : A.xi i hi α γ z = xiOff A i hi α γ z ∗ ∏ P ∈ Finset.univ.filter (fun P : BadPr I a => P.1 ∉ (B.primes X).vGs), B.embLL (locXi A i α γ P (homOf I l Z i hi z))

theorem **entry_eq_sgn_mul** (hI : I.Ok) : ∀ (c : Comm (I.rec7 l Z)) (v : VCom c) (Bs : GlobalTheory.Fam.Base c) PP (G : QGeo v Bs PP) χ (E : QExt G χ) hψ sg π (P : BadPr I a), ReadsAt A P E hψ sg π → ∀ j d (H : j + 1 + d = G.s) (hj : j + 1 ≤ B.R.m) (hj' : j + 1 < B.R.m) α γ (e' : XIx (Fin B.R.k) (Fin B.R.t) j ≃ Fin (B.R.N j)) (e : XIx (Fin B.R.k) (Fin B.R.t) (j + 1) ≃ Fin (B.R.N (j + 1))), GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.xPos e' → GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.xPos e → ∀ x, (G.T E.δ E.tame E.η).entry G.ψk G.ν j d H (E.rep j (d + 1) (by omega) (π j α)) (E.rep (j + 1) d H (π (j + 1) γ)) (GlobalTheory.SymBridge.toLoc (G.q.pr.comp (incA (I.rec7 l Z))) e e' (Formalism.L2.Col.inFr (Formalism.L2.Col.frsF I.sel (I.tup a) (A.p0 j α).1.1 P) (Formalism.L2.Col.frsF I.sel (I.tup a) (A.p0 (j + 1) γ).1.1 P) x)) = GlobalTheory.TwBridge.sgn I.sel (I.tup a) P (E.tw h

theorem **xi_factor_and_entry** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → I.IsTw → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch₀ : ℕ → Type) (A : TwDat I l Z u a Ch₀), A.Is → (∀ i (hi : i + 1 ≤ B.R.m) α γ z, A.xi i hi α γ z = xiOff A i hi α γ z ∗ ∏ P ∈ Finset.univ.filter (fun P : BadPr I a => P.1 ∉ (B.primes X).vGs), B.embLL (locXi A i α γ P (homOf I l Z i hi z))) ∧ ∀ (c : Comm (I.rec7 l Z)) (hX : c.X = (I.rec7 l Z).sp.X) (v : VCom c) (Bs : GlobalTheory.Fam.Base c) PP (G : QGeo v Bs PP) χ (E : QExt G χ) hψ sg π (P : BadPr I a), ReadsAt A P E hψ sg π → ∀ j d (H : j + 1 + d = G.s) (hj : j + 1 ≤ B.R.m) (hj' : j + 1 < B.R.m) α γ (e' : XIx (Fin B.R.k) (Fin B.R.t) j ≃ Fin (B.R.N j)) (e : XIx (Fin B.R.k) (Fin B.R.t) (j + 1) ≃ Fin (B.R.N (j + 1))), GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.xPos e' → GlobalTheory.SymBridge.IsOrd GlobalT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumFactorsOverTwistPrimesLocalStatement

theorem **_root_.GlobalTheory.T.finiteDatum_factors_over_twistPrimes_local** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → I.IsTw → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch₀ : ℕ → Type) (A : TwDat I l Z u a Ch₀), A.Is → (∀ i (hi : i + 1 ≤ B.R.m) α γ z, A.xi i hi α γ z = xiOff A i hi α γ z ∗ ∏ P ∈ Finset.univ.filter (fun P : BadPr I a => P.1 ∉ (B.primes X).vGs), B.embLL (locXi A i α γ P (homOf I l Z i hi z))) ∧ ∀ (c : Comm (I.rec7 l Z)) (hX : c.X = (I.rec7 l Z).sp.X) (v : VCom c) (Bs : GlobalTheory.Fam.Base c) PP (G : QGeo v Bs PP) χ (E : QExt G χ) hψ sg π (P : BadPr I a), ReadsAt A P E hψ sg π → ∀ j d (H : j + 1 + d = G.s) (hj : j + 1 ≤ B.R.m) (hj' : j + 1 < B.R.m) α γ (e' : XIx (Fin B.R.k) (Fin B.R.t) j ≃ Fin (B.R.N j)) (e : XIx (Fin B.R.k) (Fin B.R.t) (j + 1) ≃ Fin (B.R.N (j + 1))), GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawBetweenBasePointsMovedPoints

theorem **mem_ratAt_of** {P : HeightOneSpectrum (𝒪 E)} {g : GL (Fin n) (AfE F E)} (M : Matrix (Fin n) (Fin n) E) (h : ∀ r s, adAt F P ((g : Matrix (Fin n) (Fin n) (AfE F E)) r s) = algebraMap E _ (M r s)) : g ∈ ratAt F n P

theorem **of_mem_ratAt** {P : HeightOneSpectrum (𝒪 E)} {g : GL (Fin n) (AfE F E)} (h : g ∈ ratAt F n P) : ∃ γ : GL (Fin n) E, ∀ r s, adAt F P ((g : Matrix (Fin n) (Fin n) (AfE F E)) r s) = algebraMap E (P.adicCompletion E) ((γ : Matrix (Fin n) (Fin n) E) r s)

theorem **ratAt_of_pt** [StarRing E] {hc : Formalism.BasicNotions.CMData F E} {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) {P : HeightOneSpectrum (𝒪 E)} (hP : P ∈ Q) : g ∈ ratAt F n P

theorem **fst_eq_one** {u : B.Word} {a : I.ι} {j : ℕ} {b : (I.stp u a j).β} {g : (I.stp u a j).G b} (h : (I.stp u a j).gam b g = 1) : g.1 = 1

theorem **act_eq** {u : B.Word} {a : I.ι} {j : ℕ} {b : (I.stp u a j).β} {g : (I.stp u a j).G b} (h : (I.stp u a j).gam b g = 1) (p : (I.stp u a j).D b) : ((I.stp u a j).act b g p).1.1 = ((1, g.2.1.1, g.2.2.1) : UF B.fields.E (B.R.N j) × GLG B.fields.E (B.R.N j) (B.primes X).vGs × GLQ B.fields.E (B.R.N j) (I.Q a)) • p.1.1

theorem **cosetIn_act_congr** (hI : I.Ok) {u : B.Word} {a : I.ι} {j : ℕ} {b : (I.stp u a j).β} {g : (I.stp u a j).G b} (h : (I.stp u a j).gam b g = 1) {p p' : (I.stp u a j).D b} (e : cosetIn p = cosetIn p') : cosetIn ((I.stp u a j).act b g p) = cosetIn ((I.stp u a j).act b g p')

theorem **exists_move_of_reads** (hI : I.Ok) {u : B.Word} {a : I.ι} {i : ℕ} {Ch : Type} {chOf : (I.stp u a i).Cl → Ch} {Lb : Ch → Submodule (𝒪 B.host.E) (Fin (B.R.N i) → B.host.E)} {U : Ch → Subgroup (TC.GA B (B.R.N i))} {chi : ∀ β, U β →* ℂˣ} {y : (I.stp u a i).Cl → TC.GA B (B.R.N i)} {st : ∀ c, (I.stp u a i).Stab c → GL (Fin (B.R.N i)) B.host.E} {k : Ch → TC.GA B (B.R.N i)} (hR : I.Reads u a i chOf Lb U chi y st k) (c : (I.stp u a i).Cl) (x : U (chOf c)) (hx : x.1 ∈ UAd B.sp.cm (B.R.N i)) (hr : ∀ P ∈ I.Q a, x.1 ∈ ratAt B.host.F (B.R.N i) P) (p : (I.stp u a i).D c.1) (hp : TC.PtOf (k (chOf c)) p.1.1) (hc : cosetIn ((I.stp u a i).rep c) = cosetIn p) : ∃ g : (I.stp u a i).G c.1, (I.stp u a i).gam c.1 g = 1 ∧ TC.PtOf (x.1 * k (chOf c)) ((I.stp u a i).act c.1 g p).1.1 ∧ cosetIn ((I.stp u a i).act c.1 g p) = cosetIn p ∧ ((chi (chOf c) x : ℂˣ) : ℂ) = B.embLL (((I.stp u a i).chr c.1 g : B.LLˣ) : B

theorem **exists_move** (hI : I.Ok) (hA : A.Is) (j : ℕ) (hj : j ≤ B.R.m) (β : Ch j) (κ : A.U j β) (hr : ∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N j)) (Fin (B.R.N j)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) j hj) κ.1 : Matrix (Fin (B.R.N j)) (Fin (B.R.N j)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) (κ₀ : TC.GA B (B.R.N j)) (h0 : finM _ (rankN (I.rec7 l Z) j hj) κ.1 = κ₀.1) : ∃ g : (I.stp u a j).G (A.b0 j β), (I.stp u a j).gam (A.b0 j β) g = 1 ∧ TC.PtOf (κ₀ * A.kb j β) ((I.stp u a j).act (A.b0 j β) g (A.p0 j β)).1.1 ∧ cosetIn ((I.stp u a j).act (A.b0 j β) g (A.p0 j β)) = cosetIn (A.p0 j β) ∧ ((A.chi j β κ : ℂˣ) : ℂ) = B.embLL (((I.stp u a j).chr (A.b0 j β) g : B.LLˣ) : B.LL)

theorem **chr_root_eq** (hI : I.Ok) (hA : A.Is) (j : Fin B.R.m) (k : ℕ) (hk : k ≤ B.R.m) (g : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X k)) (hg : g ∈ GfC _) (y : TC.GA B (B.R.N k)) (hy : finM _ (rankN (I.rec7 l Z) k hk) g = y.1) (x : (AE (I.rec7 l Z))ˣ) (hx : IsDetRoot _ g x) (p : ChPt (Pr

theorem **chr_root_eq_prod** (hI : I.Ok) (hA : A.Is) (j : Fin B.R.m) (k : ℕ) (hk : k ≤ B.R.m) (β : Ch k) (κ : A.U k β) (κ₀ : TC.GA B (B.R.N k)) (h0 : finM _ (rankN (I.rec7 l Z) k hk) κ.1 = κ₀.1) (x : (AE (I.rec7 l Z))ˣ) (hx : IsDetRoot _ κ.1 x) (g : (I.stp u a k).G (A.b0 k β)) (hg : (I.stp u a k).gam (A.b0 k β) g = 1) (hq : TC.PtOf (κ₀ * A.kb k β) ((I.stp u a k).act (A.b0 k β) g (A.p0 k β)).1.1) (t : GLQ B.fields.E (B.R.N k) (rowSG (B.primes X) I.sel (I.tup a))) (ht : ∀ P, frsF I.sel (I.tup a) ((I.stp u a k).act (A.b0 k β) g (A.p0 k β)).1.1 P = g.1.1 * (frsF I.sel (I.tup a) (A.p0 k β).1.1 P * (t P).1)) : (A.chr ((j : ℕ) + 1) x : ℂ)⁻¹ = B.embLL (∏ P : rowSG (B.primes X) I.sel (I.tup a), rhoS (R

theorem **moved_law** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (κ : A.U i α) (κ' : A.U (i + 1) γ), (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ.1 : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) → (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N (i + 1))) (Fin (B.R.N (i + 1))) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) (i + 1) hi) κ'.1 : Matrix (Fin (B.R.N (i + 1))) (Fin (B.R.N (i + 1))) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) → ∀ x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ (κ₀ : TC.GA B (B.R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawBetweenBasePointsMovedPointsStatement

theorem **_root_.GlobalTheory.T.finiteDatum_law_between_basePoints_movedPoints** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (κ : A.U i α) (κ' : A.U (i + 1) γ), (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) i (Nat.le_of_succ_le hi)) κ.1 : Matrix (Fin (B.R.N i)) (Fin (B.R.N i)) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) → (∀ P ∈ I.Q a, ∃ M : Matrix (Fin (B.R.N (i + 1))) (Fin (B.R.N (i + 1))) B.host.E, ∀ r s, adAt B.host.F P ((finM _ (rankN (I.rec7 l Z) (i + 1) hi) κ'.1 : Matrix (Fin (B.R.N (i + 1))) (Fin (B.R.N (i + 1))) (Formalism.L2.DIV.AfE B.host.F B.host.E)) r s) = algebraMap B.host.E _ (M r s)) → ∀ x y, Is

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawRationalPointsHalf

theorem **isHalf_actO** {D : D7} {S T : HSp D} {e : AE D} (g : GA D S) (h : GA D T) {u : WO S T} (hu : IsHalf e u) : IsHalf e (actO S T g h u)

theorem **law_rat** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ, ∀ Φ : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) → ℂ, IsSBHalf A.e (stdHom _ _) Φ → ∀ p : ℂ, (∀ z, Φ (z.map fun r => A.e * incA (I.rec7 l Z) r) = p * A.xi i hi α γ z) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ z : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1))).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (Fin ((I.rec7 l Z).host.E, (A.chr (i + 1) x : ℂ)⁻¹ * (A.chr (i + 1) y : ℂ) * Φ (actO _ _ κ.1⁻¹ κ'.1⁻¹ (z.map fun r => A.e * incA (I.rec7 l Z) r)) = ((A.chi i α κ : ℂˣ) : ℂ)⁻¹ * ((A.chi (i + 1) γ κ' : ℂˣ) : ℂ) * Φ (z.map fun r => A.e * incA (I.rec7 l Z) r)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawRationalPointsHalfStatement

theorem **_root_.GlobalTheory.T.finiteDatumLaw_rationalPoints_half** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ, ∀ Φ : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) → ℂ, IsSBHalf A.e (stdHom _ _) Φ → ∀ p : ℂ, (∀ z, Φ (z.map fun r => A.e * incA (I.rec7 l Z) r) = p * A.xi i hi α γ z) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ z : Matrix (Fin ((I.rec7 l Z).sp.X i)).rank) (Fin ((I.rec7 l Z).sp.X (i + 1))).rank) (Fin ((I.rec7 l Z).host.E, (A.chr (i + 1) x : ℂ)⁻¹ * (A.chr (i + 1) y : ℂ) * Φ (actO _ _ κ.1⁻¹ κ'.1⁻¹ (z.map fun r => A.e * incA (I.rec7 l Z) r)) = ((A.chi i α κ : ℂˣ) : ℂ)⁻¹ * ((A.chi (i + 1) γ κ' : ℂˣ) : ℂ) * Φ (z.map fun r => A.e * incA (I.rec7 l Z) r)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawUnderTwoChartLevels

theorem **half_nsmul** {D : D7} {m n : ℕ} (e : AE D) (v : Mx D m n) (h : IsHalf e v) (k : ℕ) : IsHalf e (k • v)

theorem **sb_step** {Y : Type} [AddCommGroup Y] (Λ : AddSubgroup Y) (hf : Y → Prop) (Φ : Y → ℂ) (k : ℕ) (hinv : ∀ u ∈ Λ, hf u → ∀ t, Φ (t + k • u) = Φ t) (k' : ℕ) (v : Y) (hv : v ∈ Λ) (hh : hf (k' • v)) (t : Y) : Φ (t + (k * k') • v) = Φ t

theorem **levDat_law** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ, ∀ Φ : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) → ℂ, IsSBHalf A.e (stdHom _ _) Φ → ∀ p : ℂ, (∀ z, Φ (z.map fun r => A.e * incA (I.rec7 l Z) r) = p * A.xi i hi α γ z) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ z, IsHalf A.e z → (A.chr (i + 1) x : ℂ)⁻¹ * (A.chr (i + 1) y : ℂ) * Φ (actO _ _ κ.1⁻¹ κ'.1⁻¹ z) = ((A.chi i α κ : ℂˣ) : ℂ)⁻¹ * ((A.chi (i + 1) γ κ' : ℂˣ) : ℂ) * Φ z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawUnderTwoChartLevelsStatement

theorem **_root_.GlobalTheory.T.finite_datum_law_under_two_chart_levels** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ, ∀ Φ : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1)) → ℂ, IsSBHalf A.e (stdHom _ _) Φ → ∀ p : ℂ, (∀ z, Φ (z.map fun r => A.e * incA (I.rec7 l Z) r) = p * A.xi i hi α γ z) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x →

IsDetRoot _ κ'.1 y → ∀ z, IsHalf A.e z → (A.chr (i + 1) x : ℂ)⁻¹ * (A.chr (i + 1) y : ℂ) * Φ (actO _ _ κ.1⁻¹ κ'.1⁻¹ z) = ((A.chi i α κ : ℂˣ) : ℂ)⁻¹ * ((A.chi (i + 1) γ κ' : ℂˣ) : ℂ) * Φ z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawUnderTwoLevelsModHalfLattice

theorem **isHalf_add** {D : D7} {m n : ℕ} {e : AE D} {u v : Mx D m n} (hu : IsHalf e u) (hv : IsHalf e v) : IsHalf e (u + v)

theorem **xi_law** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → (∀ (z₁ z₂ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), t ∈ stdHom _ _ → IsHalf A.e t → z₁.map (fun r => A.e * incA (I.rec7 l Z) r) = z₂.map (fun r => A.e * incA (I.rec7 l Z) r) + k • t → A.xi i hi α γ z₁ = A.xi i hi α γ z₂) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ (z z' : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (v : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), v ∈ stdHom _ _ → IsHalf A.e v → actO _ _ κ.1⁻¹ κ'.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDatumLawUnderTwoLevelsModHalfLatticeStatement

theorem **_root_.GlobalTheory.T.finiteDatum_law_under_two_levels_mod_half_lattice** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ (i : ℕ) (hi : i + 1 ≤ B.R.m) α γ (k : ℕ), 0 < k → (∀ (z₁ z₂ : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (t : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).sp.X (i + 1))), t ∈ stdHom _ _ → IsHalf A.e t → z₁.map (fun r => A.e * incA (I.rec7 l Z) r) = z₂.map (fun r => A.e * incA (I.rec7 l Z) r) + k • t → A.xi i hi α γ z₁ = A.xi i hi α γ z₂) → ∀ (κ : A.U i α) (κ' : A.U (i + 1) γ) x y, IsDetRoot _ κ.1 x → IsDetRoot _ κ'.1 y → ∀ (z z' : Matrix (Fin ((I.rec7 l Z).sp.X (i + 1)).rank) (Fin ((I.rec7 l Z).sp.X i).rank) (I.rec7 l Z).host.E) (v : WO ((I.rec7 l Z).sp.X i) ((I.rec7 l Z).s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimUnitaryKTypeEqDualWeylModule

theorem **dual_weyl_type** : ∀ (k n : ℕ) (h : k = n) (lam : DomWeight n) (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [FiniteDimensional ℂ H] (σ : Matrix (Fin k) (Fin k) ℂ → (H →L[ℂ] H)) (lamH : Fin k → ℝ), IsURepOn (UJset (allPos k)) σ → IsKTypeOf (allPos k) σ (Formalism.QuantumDeterminants.lowK (allPos k) lamH) ⊤ → (∀ i : Fin k, -lamH (Fin.rev i) = (lam.w (Fin.cast h i) : ℝ) + ((rhoVec k i : ℚ) : ℝ)) → ∃ (j : H →ₗ[ℂ] MvPolynomial (Fin n × Fin n) ℂ) (d : ℕ) (b : Module.Basis (Fin d) ℂ H), (∀ ξ, j ξ ∈ delSq ℂ n lam) ∧ (∀ p ∈ delSq ℂ n lam, ∃ ξ, j ξ = p) ∧ (∀ a a', normFock lam (j (b a)) (j (b a')) = if a = a' then 1 else 0) ∧ ∀ g ∈ Matrix.unitaryGroup (Fin n) ℂ, ∀ ξ, j (σ (g.submatrix (Fin.cast h) (Fin.cast h)) ξ) = (g.det ^ (-lam.detShift)) • Formalism.GradedWeylDivisibility.rowAct ℂ gᴴ (j ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimUnitaryKTypeEqDualWeylModuleStatement

theorem **_root_.GlobalTheory.T.finite_dim_unitary_K_type_eq_dual_weyl_module** : ∀ (k n : ℕ) (h : k = n) (lam : DomWeight n) (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [FiniteDimensional ℂ H] (σ : Matrix (Fin k) (Fin k) ℂ → (H →L[ℂ] H)) (lamH : Fin k → ℝ), IsURepOn (UJset (allPos k)) σ → IsKTypeOf (allPos k) σ (Formalism.QuantumDeterminants.lowK (allPos k) lamH) ⊤ → (∀ i : Fin k, -lamH (Fin.rev i) = (lam.w (Fin.cast h i) : ℝ) + ((rhoVec k i : ℚ) : ℝ)) → ∃ (j : H →ₗ[ℂ] MvPolynomial (Fin n × Fin n) ℂ) (d : ℕ) (b : Module.Basis (Fin d) ℂ H), (∀ ξ, j ξ ∈ delSq ℂ n lam) ∧ (∀ p ∈ delSq ℂ n lam, ∃ ξ, j ξ = p) ∧ (∀ a a', normFock lam (j (b a)) (j (b a')) = if a = a' then 1 else 0) ∧ ∀ g ∈ Matrix.unitaryGroup (Fin n) ℂ, ∀ ξ, j (σ (g.submatrix (Fin.cast h) (Fin.cast h)) ξ) = (g.det ^ (-lam.detShift)) • Formalism.GradedWeylDivisibility.rowAct ℂ gᴴ (j ξ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimensionalSpanCoefficients

theorem **finiteDimensional_span_coefficients** {G W : Type} [Group G] [AddCommGroup W] [Module ℂ W] [FiniteDimensional ℂ W] (ρ : Representation ℂ G W) : FiniteDimensional ℂ ↥(GlobalTheory.Published.AutFin.coeffSpan ρ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimensionalSpanMultilinearValues

theorem **finiteDimensional_span_multilinear_values** : ∀ {ι : Type} [Fintype ι] (E : ι → Type) [∀ i, AddCommGroup (E i)] [∀ i, Module ℂ (E i)] (F : Type) [AddCommGroup F] [Module ℂ F] (T : MultilinearMap ℂ E F) (W : ∀ i, Submodule ℂ (E i)), (∀ i, FiniteDimensional ℂ (W i)) → FiniteDimensional ℂ (Submodule.span ℂ {f | ∃ v : ∀ i, E i, (∀ i, v i ∈ W i) ∧ f = T v})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimensionalSpanMultilinearValuesStatement

theorem **_root_.GlobalTheory.T.finiteDimensional_span_multilinear_values'** : ∀ {ι : Type} [Fintype ι] (E : ι → Type) [∀ i, AddCommGroup (E i)] [∀ i, Module ℂ (E i)] (F : Type) [AddCommGroup F] [Module ℂ F] (T : MultilinearMap ℂ E F) (W : ∀ i, Submodule ℂ (E i)), (∀ i, FiniteDimensional ℂ (W i)) → FiniteDimensional ℂ (Submodule.span ℂ {f | ∃ v : ∀ i, E i, (∀ i, v i ∈ W i) ∧ f = T v})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimensionalSpanSeparately

theorem **exists_finset_eval_injective** {Z : Type} (T : Submodule ℂ (Z → ℂ)) [FiniteDimensional ℂ T] : ∃ Y : Finset Z, ∀ g ∈ T, (∀ y ∈ Y, g y = 0) → g = 0

theorem **finiteDimensional_span_separately** (ι : Type) [Fintype ι] [DecidableEq ι] (X : ι → Type) (S : ∀ i, Submodule ℂ (X i → ℂ)) (hS : ∀ i, FiniteDimensional ℂ (S i)) : FiniteDimensional ℂ (Submodule.span ℂ {f : (∀ i, X i) → ℂ | ∀ i x, (fun y => f (Function.update x i y)) ∈ S i})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimensionalThetaFunctionsLevelFockDegree

theorem **norm_le_half** {R : Type} [NormedAddCommGroup R] [NormedSpace ℝ R] {F : R → ℂ} (hF : Differentiable ℝ F) (x y : R) (α β q C : ℝ) (hcr : ∀ a b : ℝ, fderiv ℝ F (a • x + b • y) x + I * fderiv ℝ F (a • x + b • y) y = π * I * ((α + b * q) + I * (β - a * q)) * F (a • x + b • y)) (hC : ∀ z, ‖F z‖ ≤ C) (h0 : F 0 = 0) (hq : Real.exp (32 * π * |q|) ≤ 2) : ‖F x‖ ≤ C / 2

theorem **fg_of_compact** {G : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] (V : Submodule ℂ (G → ℂ)) {K N : Set G} (hK : IsCompact K) (hN : N ∈ 𝓝 (0 : G)) (hb : ∀ f ∈ V, ∃ C, ∀ x, ‖f x‖ ≤ C) (hk : ∀ x, ∃ k ∈ K, ∀ f ∈ V, ‖f x‖ = ‖f k‖) (hs : ∀ f ∈ V, ∀ C, (∀ x, ‖f x‖ ≤ C) → ∀ w, f w = 0 → ∀ n ∈ N, ‖f (w + n)‖ ≤ C / 2) : V.FG

theorem **exists_fg_of_words** {M ι : Type} [AddCommGroup M] [Module ℂ M] [Finite ι] (P : Submodule ℂ M) (A : ι → M →ₗ[ℂ] M) (a : ι → M → M) (ha : ∀ i, ∀ f ∈ P, A i f = a i f) (hP : ∀ i, ∀ f ∈ P, a i f ∈ P) (h0 : (P ⊓ LinearMap.ker (LinearMap.pi A)).FG) (N : ℕ) : ∃ V : Submodule ℂ M, V.FG ∧ ∀ f ∈ P, (∀ l : List ι, l.length = N → l.foldr (fun i g => a i g) f = 0) → f ∈ V

theorem **annihilation_eq** (h : Chart θ σ) (hψ : IsStdChar D ψ (-1)) {κ : Type} (b : Module.Basis κ ℝ (Fin T.rank → Fin S.rank → ℂ)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hv : ∀ τ k, annih (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) (b k) f = 0) (w : WO S T) (x : RW T S) : fderiv ℝ (fun z : RW T S => f (w + iotaW σ T S z)) 0 x + I * fderiv ℝ (fun z : RW T S => f (w + iotaW σ T S z)) 0

(GlobalTheory.VacUnique.jO S T sS sT x) = π * I * (omO S T (piW θ T S w) x + I * omO S T (piW θ T S w) (GlobalTheory.VacUnique.jO S T sS sT x)) * f w

theorem **fg_vacua** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {κ : Type} (b : Module.Basis κ ℝ (Fin T.rank → Fin S.rank → ℂ)) {U : AddSubgroup (WO S T)} (hU : IsOpen (U : Set (WO S T))) (V : Submodule ℂ (WO S T → ℂ)) (hV : V ≤ levelSub ψ S T U) (hv : ∀ f ∈ V, ∀ τ k, annih (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) (b k) f = 0) : V.FG

theorem **finiteDimensional_level_degree** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : RealMat S) (sT : RealMat T), IsStdChar D ψ (-1) → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ (U : AddSubgroup (WO S T)), IsOpen (U : Set (WO S T)) → ∀ N : ℕ, FiniteDimensional ℂ (Submodule.span ℂ {f : WO S T → ℂ | IsThetaFunO ψ S T f ∧ (∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u f = f) ∧ ∀ l : List (D.what₂.Tau × Matrix (Fin T.rank) (Fin S.rank) ℂ), l.length = N → l.foldr (fun p f => annih (heisO ψ S T) θ p.1 (cplxO (sS p.1) (sT p.1)) p.2 f) f = 0})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteDimensionalThetaFunctionsLevelFockDegreeStatement

theorem **_root_.GlobalTheory.T.finiteDimensional_thetaFunctions_level_fockDegree** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : RealMat S) (sT : RealMat T), IsStdChar D ψ (-1) → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ (U : AddSubgroup (WO S T)), IsOpen (U : Set (WO S T)) → ∀ N : ℕ, FiniteDimensional ℂ (Submodule.span ℂ {f : WO S T → ℂ | IsThetaFunO ψ S T f ∧ (∀ u ∈ U, u.map (infPart D) = 0 → heisO ψ S T u f = f) ∧ ∀ l : List (D.what₂.Tau × Matrix (Fin T.rank) (Fin S.rank) ℂ), l.length = N → l.foldr (fun p f => annih (heisO ψ S T) θ p.1 (cplxO (sS p.1) (sT p.1)) p.2 f) f = 0})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteElementsActFockVectorsThroughDatum

theorem **pt_finite** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T) (k : KerDat S T), x.Is → y.Is → k.Is ψ θ x y → ∀ P, IsThetaFunO ψ S T (k.pt ψ θ x y P) ∧ ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ p : ℂ, k.om (g, h) (comb ψ S T k.c k.vac) = p • comb ψ S T k.c k.vac → k.om (g, h) (k.pt ψ θ x y P) = p • k.pt ψ θ x y P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteElementsActFockVectorsThroughDatumStatement

theorem **_root_.GlobalTheory.T.finiteElements_act_fockVectors_through_datum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D), IsInfFam θ → ∀ (S T : HSp D) (x : DefSp S) (y : DefSp T) (k : KerDat S T), x.Is → y.Is → k.Is ψ θ x y → ∀ P, IsThetaFunO ψ S T (k.pt ψ θ x y P) ∧ ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ p : ℂ, k.om (g, h) (comb ψ S T k.c k.vac) = p • comb ψ S T k.c k.vac → k.om (g, h) (k.pt ψ θ x y P) = p • k.pt ψ θ x y P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteFourierExpansionHalfSplitPlaces

theorem **nsmul_mem_perO_of_halfB** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} (hc : 0 < c) {y : Formalism.L2.Col.HomE E a b} (hy : y ∈ halfBO SS M c) : c • y ∈ M

theorem **xi_add_halfB** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {N : Submodule (𝒪 E) (Formalism.L2.Col.Sp E a)} {N' : Submodule (𝒪 E) (Formalism.L2.Col.Sp E b)} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} (hc : 0 < c) (Ξ : Formalism.L2.Col.HomE E a b → ℂ) (hΞ2 : ∀ x, ∀ y ∈ Formalism.L2.Col.homAt SS N N', Ξ (x + (c : E) • y) = Ξ x) (hH : ∀ z, (∀ P ∈ SS, z ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal M) → z ∈ Formalism.L2.Col.homAt SS N N') {β : Formalism.L2.Col.HomE E a b} (hβ : β ∈ halfBO SS M c) (x : Formalism.L2.Col.HomE E a b) : Ξ (x + β) = Ξ x

theorem **nsmul_toMatrix_mem_latRat** {D : D7} {S T : HSp D} (hcc : D.host.ccK = ⊤) {N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (dS : IsSelfDualC S N) (k c : ℕ) {y : Formalism.L2.Col.HomE D.host.E S.rank T.rank} (hy : c • y ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') : (k * c) • LinearMap.toMatrix' y ∈ GlobalTheory.LatVac.latRat S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E))

theorem **char_brk** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {δ : ℂ} {sc : ℝ} (hP : StdPair θ S T δ sc ψ) {ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (dS : IsSelfDualC S ΛS) (dT : IsSelfDualC T ΛT) {e : AE D} (he0 : infPart D e = 0) {k : ℕ} {t u : WO S T} (ht : IsHalf e t) (hu : u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) {M : Matrix (Fin T.rank) (Fin S.rank) D.host.E} (htM : t = GlobalTheory.LatVac.finTens 1 M + k • u) (y' : Formalism.L2.Col.HomE D.host.E S.rank T.rank) (hY : k • LinearMap.toMatrix' y' ∈ GlobalTheory.LatVac.latRat S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E))) : ψ (symplO S T t (eXm S T (cA D e) y')) = brk (HV sc) (embH (tauE D) δ y') (embH (tauE D) δ (Matrix.toLin' M))

theorem **fin_sum** {V : Type} [AddCommGroup V] [Module ℂ V] (H : Formalism.L2.Th1.H.Sesq V) {A B Λ : AddSubgroup V} {SA SB : Finset V} (f g : V → ℂ) {v : V} (hv : v ∈ Λ ⊓ A) (hRA : Formalism.L2.Th1.H.IsReps B (Λ ⊓ A) SA) (horth : ∀ w ∈ Λ ⊓ A, ∑ y ∈ SB, brk H y w = if w ∈ B then (SB.card : ℂ) else 0) (hcard : SA.card = SB.card) (hne : SB.card ≠ 0) (hg : ∀ y ∈ SB, g y = brk H y v) : ∃ s0 ∈ SA, v - s0 ∈ B ∧ ∑ y ∈ SB, (((SA.card : ℂ))⁻¹ * ∑ s ∈ SA, f s * (brk H y s)⁻¹) * g y = f s0

theorem **onV_embH** {E : Type} [Field E] [NumberField E] [StarRing E] {ι : Type} [Fintype ι] [Nonempty ι] {a b : ℕ} (emb : ι → E →+* ℂ) {δ : ℂ} (hδ : δ ≠ 0) (Ξ : Formalism.L2.Col.HomE E a b → ℂ) (z : Formalism.L2.Col.HomE E a b) : onV emb δ Ξ (embH emb δ z) = Ξ z

theorem **mem_BC** {E : Type} [Field E] [NumberField E] [StarRing E] {ι : Type} [Fintype ι] {a b : ℕ} {emb : ι → E →+* ℂ} {δ : ℂ} {AA0 : Submodule (𝒪 E) E} {SS : Finset (HeightOneSpectrum (𝒪 E))} {N : Submodule (𝒪 E) (Formalism.L2.Col.Sp E a)} {N' : Submodule (𝒪 E) (Formalism.L2.Col.Sp E b)} {c : ℕ} {w : VV ι a b} (h : w ∈ BC emb δ AA0 SS N N' c) : ∃ β ∈ halfBO SS (perO AA0 N N') c, embH emb δ β = w

theorem **embH_sub_eq** {E : Type} [Field E] [NumberField E] {ι : Type} {a b : ℕ} (emb : ι → E →+* ℂ) (δ : ℂ) (m p β : Formalism.L2.Col.HomE E a b) (x0 s0 : VV ι a b) (hp : embH emb δ p = x0) (hβ : embH emb δ β = embH emb δ m - s0) : s0 - x0 = embH emb δ (m - p - β)

theorem **xi_sub_halfB** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {N : Submodule (𝒪 E) (Formalism.L2.Col.Sp E a)} {N' : Submodule (𝒪 E) (Formalism.L2.Col.Sp E b)} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} (hc : 0 < c) (Ξ : Formalism.L2.Col.HomE E a b → ℂ) (hΞ2 : ∀ x, ∀ y ∈ Formalism.L2.Col.homAt SS N N', Ξ (x + (c : E) • y) = Ξ x) (hH : ∀ z, (∀ P ∈ SS, z ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal M) → z ∈ Formalism.L2.Col.homAt SS N N') {β : Formalism.L2.Col.HomE E a b} (hβ : β ∈ halfBO SS M c) (x : Formalism.L2.Col.HomE E a b) : Ξ (x - β) = Ξ x

theorem **four_expand** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → (∃ ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E), IsSelfDualC S ΛS ∧ (ΛS : Set (Fin S.rank → D.host.E)) = latFr

S hS N) → (∃ ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E), IsSelfDualC T ΛT ∧ (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N') → ∀ (Φ : WO S T → ℂ) (Ξ : Formalism.L2.Col.HomE D.host.E a b → ℂ), IsSBHalf e (latO S T (latFr S hS N)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteFourierExpansionHalfSplitPlacesStatement

theorem **_root_.GlobalTheory.T.finite_fourier_expansion_half_split_places** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS ∗ D.host.q ∗ D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → (∃ ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E), IsSelfDualC S ΛS ∧ (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N) → (∃ ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E), IsSelfDualC T ΛT ∧ (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N') → ∀ (Φ : WO S T → ℂ) (Ξ : Formalism.L2.Col.HomE D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteMatrixEntryIntegralOfKeepsStandardLattice

theorem **map_mulVec_single** {A B : Type} [CommRing A] [CommRing B] (f : A →+* B) {n : ℕ} (M : Matrix (Fin n) (Fin n) A) (y : Fin n → A) (r s : Fin n) (h1 : f (y s) = 1) (h0 : ∀ b, b ≠ s → y b = 0) : f ((M *ᵥ y) r) = f (M r s)

theorem **toEAf_pair** (D : D7) (α : Af D.host.F) (e : D.host.E) : toEAf D (pairA D.host.F 0 α ⊗ₜ[D.host.F] e) = e ⊗ₜ[D.host.F] α

theorem **infPart_pair** (D : D7) (α : Af D.host.F) (e : D.host.E) : infPart D (pairA D.host.F 0 α ⊗ₜ[D.host.F] e) = 0

theorem **coord_int** (D : D7) (S : HSp D) (L : GlobalTheory.FinPl.Lat S) (hL : ∀ z, z ∈ L ↔ z ∈ GlobalTheory.AdForm.intVec D S.rank) (v : HeightOneSpectrum (𝒪 D.host.F)) (y : Fin S.rank → AE D) (hy : y ∈ GlobalTheory.FinPl.latV S L (fun _ => True)) (i : Fin S.rank) : Formalism.UG.prV (E

theorem **finM_entry_int** : ∀ (D : D7) (S : HSp D) {n : ℕ} (hn : S.rank = n) (L : GlobalTheory.FinPl.Lat S), (∀ z, z ∈ L ↔ z ∈ GlobalTheory.AdForm.intVec D S.rank) → ∀ κ : GA D S, GlobalTheory.FinPl.Moves S κ L L (fun _ => True) → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (r s : Fin n), Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteMatrixEntryIntegralOfKeepsStandardLatticeStatement

theorem **_root_.GlobalTheory.T.finiteMatrix_entry_integral_of_keeps_standardLattice** : ∀ (D : D7) (S : HSp D) {n : ℕ} (hn : S.rank = n) (L : GlobalTheory.FinPl.Lat S), (∀ z, z ∈ L ↔ z ∈ GlobalTheory.AdForm.intVec D S.rank) → ∀ κ : GA D S, GlobalTheory.FinPl.Moves S κ L L (fun _ => True) → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (r s : Fin n), Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteMonomialsWithPrescribedPlaceDegree

theorem **finite_monomials_of_placeDegree** : ∀ (D : D7) (m n : ℕ) (d : D.what₂.Tau → ℕ), {α : FVar D m n →₀ ℕ | ∀ τ, (∑ v ∈ α.support with v.1 = τ, α v) = d τ}.Finite

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteMonomialsWithPrescribedPlaceDegreeStatement

theorem **_root_.GlobalTheory.T.finite_monomials_with_prescribed_placeDegree** : ∀ (D : D7) (m n : ℕ) (d : D.what₂.Tau → ℕ), {α : FVar D m n →₀ ℕ | ∀ τ, (∑ v ∈ α.support with v.1 = τ, α v) = d τ}.Finite

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinitePartMemFrameStabilizerIffLocalStabilizer

theorem **col_mem_hatL** {β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} {Λ : Submodule (𝒪 E) (Fin n → E)} (hβ : IsFrame β Λ) (j : Fin n) : β.1 *ᵥ Pi.single j 1 ∈ Formalism.L2.Herm.hatL (F

theorem **exists_fg_sub** {β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} {Λ : Submodule (𝒪 E) (Fin n → E)} (hβ : IsFrame β Λ) : ∃ Λ₁ ≤ Λ, Λ₁.FG ∧ Formalism.L2.Herm.hatL (F

theorem **latLoc_eq** {Λ₁ Λ : Submodule (𝒪 E) (Fin n → E)} (hle : Λ₁ ≤ Λ) (hfg : Λ₁.FG) (heq : Formalism.L2.Herm.hatL (F

theorem **finE_mem_St_iff_of_fg** (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hfr : Formalism.L2.DIV.IsFrame β Λ) (hfg : Λ.FG) (g : GA D V) : Lev.finE V g ∈ Formalism.L2.DIV.St β ↔ ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v

theorem **finE_mem_St_iff_stabLoc** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)), Formalism.L2.DIV.IsFrame β Λ → ∀ g : GA D V, Lev.finE V g ∈ Formalism.L2.DIV.St β ↔ ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinitePartMemFrameStabilizerIffLocalStabilizerStatement

theorem **_root_.GlobalTheory.T.finitePart_mem_frameStabilizer_iff_local_stabilizer** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)), Formalism.L2.DIV.IsFrame β Λ → ∀ g : GA D V, Lev.finE V g ∈ Formalism.L2.DIV.St β ↔ ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinitePartMemLevelOfThetaDatumNeZero

theorem **proj_eq_one_of_finHom** (V : HSp D) (h : GA D V) (hh : h ∈ GinfC V) (v : HeightOneSpectrum (𝒪 D.host.F)) : GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v h = 1

theorem **proj_finPartG** (V : HSp D) (g : GA D V) (v : HeightOneSpectrum (𝒪 D.host.F)) : GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v (GlobalTheory.LevelNbhd.finPartG V g) = GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v g

theorem **finE_finPartG** (V : HSp D) (g : GA D V) : Lev.finE V (GlobalTheory.LevelNbhd.finPartG V g) = Lev.finE V g

theorem **half_supp** (S T : HSp D) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (e : AE D) (x : WO S T) (h : IsHalf e x) : ∀ i j, toEAf D e ∗ frMapO S T βS βT x i j = frMapO S T βS βT x i j

theorem **datum_spec** (S T : HSp D) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ S.rank) (χ ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (e : AE D) (x : WO S T) (hx : (if IsHalf e x ∧ IsIntM (frMapO S T βS βT x) then omegaG L χ (fun PP => redM PP (frMapO S T βS βT x)) ∗ omegaQ ∅ L.r ξ 1 (fun PP => redM PP (frMapO S T βS βT x)) else 0) ≠ 0) : IsHalf e x ∧ IsIntM (frMapO S T βS βT x) ∧ (∀ PP ∈ L.Pg, IsUnit (topL S.rank (redM PP (frMapO S T βS βT x))).det) ∧ ∃ a : ∀ PP : ɪL.Pg, GL (Fin S.rank) (kP PP.1), (∀ PP : ɪL.Pg, ((a PP : GL (Fin S.rank) (kP PP.1)) : Matrix (Fin S.rank) (Fin S.rank) (kP PP.1)) = topL S.rank (redM PP.1 (frMapO S T βS βT x))) ∧ (∀ PP : ɪL.Pg, a PP ∈ blockTri (blkOf L.r L.t)) ∧ (fun PP => upperLeftBlock L.hr (a PP)) ∈ L.P.H

theorem **finE_mem_lev** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ S.rank) (χ ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (e : AE D), IsSelfDualC S ΛS → Formalism.L2.DIV.IsFrame βS ΛS → S.rank ≤ T.rank → L.Q = ∅ → IsIdemSum e (L.Pg ∪ ∅) → ∀ g : GA D S, (∀ v, (∀ PP ∈

L.Pg, PP.under (𝒪 D.host.F) ≠ v) → GlobalTheory.AdelicU.proj D.sp.cm.sig S.gram v g ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig S.gram ΛS v) → ∀ x, (fun x : WO S T => if IsHalf e x ∧ IsIntM (frMapO S T βS βT x) then omegaG L χ (fun PP => redM PP (frMapO S T βS βT x)) ∗ omegaQ ∅ L.r ξ 1 (fun PP => redM PP (frMapO S T βS βT x)) else 0) x ≠ 0 → (fun x : WO S T => if IsHalf e x ∧ IsIntM (frMapO S T βS βT x) then omegaG L χ (fun PP => redM PP (frMapO S T βS βT x)) ∗ omegaQ ∅ L.r ξ 1 (fun PP => redM PP (frMapO

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinitePartMemLevelOfThetaDatumNeZeroStatement

theorem **_root_.GlobalTheory.T.finitePart_mem_level_of_thetaDatum_ne_zero** : ∀ (D : D7) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ S.rank) (χ ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →∗ ℂˣ) (e : AE D), IsSelfDualC S ΛS → Formalism.L2.DIV.IsFrame βS ΛS → S.rank ≤ T.rank → L.Q = ∅ → IsIdemSum e (L.Pg ∪ ∅) → ∀ g : GA D S, (∀ v, (∀ PP ∈ L.Pg, PP.under (𝒪 D.host.F) ≠ v) → GlobalTheory.AdelicU.proj D.sp.cm.sig S.gram v g ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig S.gram ΛS v) → ∀ x, (fun x : WO S T => if IsHalf e x ∧ IsIntM (frMapO S T βS βT x) then omegaG L χ (fun PP => redM PP (frMapO S T βS βT x)) ∗ omegaQ ∅ L.r ξ 1 (fun PP => redM PP (frMapO S T βS βT x)) else 0) x ≠ 0 → (fun x : WO S T => if IsHalf e x ∧ IsIntM (frMapO S T βS βT x) then omegaG L χ (fun PP => redM PP (frMapO S T βS β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinitePartTraceCharacterIdempotentInverseDiffere

theorem **indicatorAdele_singleton_mul_singleton** (x v : Pl D) : indP (D

theorem **indicatorIdempotent_mul** (x v : Pl D) : cP (D

theorem **ex_mul** (x v : Pl D) : ex v ∗ ex x = if v = x then ex x else 0

theorem **sum_mul_ex** (S : Finset (Pl D)) (f : Pl D → D.host.E) (hf : ∀ v ∉ S, f v = 0) (x : Pl D) : (∑ v ∈ S, ex v ∗ incA D (f v)) ∗ ex x = ex x ∗ incA D (f x)

theorem **alpha_mul_ex** (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (x : Pl D) : alpha S a ∗ ex x = ex x ∗ incA D (a x)

theorem **alpha_mul_inv** (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (h0 : ∀ v, a v ≠ 0) : alpha S a ∗ alpha S (fun v => (a v)⁻¹) = 1

theorem **cA_incA** (t : D.host.E) : cA D (incA D t) = incA D (D.host.conj t)

theorem **tmul_eq** (p : AF D.host.F) (e : D.host.E) : p ⊗ₜ[D.host.F] e = algebraMap (AF D.host.F) (AE D) p ∗ incA D e

theorem **evalV_inv_mem** (x : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ x.asIdeal) : evalV D.host.F x (algebraMap D.host.F (Af D.host.F) ((c : D.host.F))⁻¹) ∈ x.adicCompletionIntegers D.host.F

theorem **gam_int** (x : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ x.asIdeal) : gam x c ∈ intAf D.host.F

theorem **gam_supp** (x : Pl D) (c : 𝒪 D.host.F) : ∀ w, ¬ (w = x) → evalV D.host.F w (gam x c) = 0

theorem **coe_ne_zero** (x : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ x.asIdeal) : ((c : D.host.F) : D.host.E) ≠ 0

theorem **ex_scale** (x : Pl D) (c : 𝒪 D.host.F) (e : D.host.E) : ex x ∗ incA D (((c : D.host.F) : D.host.E)⁻¹ ∗ e) = pairA D.host.F 0 (gam x c) ⊗ₜ[D.host.F] e

theorem **pair_supp** (x : Pl D) (γ : Af D.host.F) (hγ : ∀ w, ¬ (w = x) → evalV D.host.F w γ = 0) : cP (D

theorem **scale_tmul** (x : Pl D) (ρ : AE D) (t : D.host.E) (hρ : ρ ∗ ex x = ex x ∗ incA D t) (γ : Af D.host.F) (hγ : ∀ w, ¬ (w = x) → evalV D.host.F w γ = 0) (e : D.host.E) : ρ ∗ (pairA D.host.F 0 γ ⊗ₜ[D.host.F] e) = pairA D.host.F 0 γ ⊗ₜ[D.host.F] (t ∗ e)

theorem **indicatorAdele_mul_eq_self** (x : Pl D) (γ : Af D.host.F) (hγ : ∀ w, ¬ (w = x) → evalV D.host.F w γ = 0) : indP (D

theorem **pair_scale** (x : Pl D) (c : 𝒪 D.host.F) (γ : Af D.host.F) (hγ : ∀ w, ¬ (w = x) → evalV D.host.F w γ = 0) (e : D.host.E) : pairA D.host.F 0 γ ⊗ₜ[D.host.F] (((c : D.host.F) : D.host.E)⁻¹ ∗ e) = pairA D.host.F 0 (gam x c ∗ γ) ⊗ₜ[D.host.F] e

theorem **gam_mul_int** (x : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ x.asIdeal) {γ : Af D.host.F} (hγ : γ ∈ intAf D.host.F) : gam x c ∗ γ ∈ intAf D.host.F

theorem **gam_mul_supp** (x : Pl D) (c : 𝒪 D.host.F) (γ : Af D.host.F) : ∀ w, ¬ (w = x) → evalV D.host.F w (gam x c ∗ γ) = 0

theorem **smul_latMul** (S' : HSp D) (L : Lat S') (x : Pl D) (𝔞 𝔞' : Set D.host.E) (ρ : AE D) (t : D.host.E) (hρ : ρ ∗ ex x = ex x ∗ incA D t) (h𝔞 : ∀ a ∈ 𝔞, ∃ c : 𝒪 D.host.F, c ∉ x.asIdeal ∧ ∃ a' ∈ 𝔞', ∃ o : 𝒪 D.host.E, t ∗ a = ((c : D.host.F) : D.host.E)⁻¹ ∗ (a' ∗ (o : D.host.E))) (y : Fin S'.rank → AE D) (hy : y ∈ latMul S' 𝔞 L (· = x)) : ρ • y ∈ latMul S' 𝔞' L (· = x)

theorem **lat_iff** (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (h0 : ∀ v, a v ≠ 0) (hgen : ∀ v, ∃ c : 𝒪 D.host.F, c ∉ v.asIdeal ∧ ((c : D.host.F) : D.host.E) ∗ a v ∈ D.host.AA ∧ ∀ b ∈ D.host.AA, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) ∗ b = a v ∗ (o : D.host.E)) (Sp : HSp D) (L : Lat Sp) (x : Pl D) (y : Fin Sp.rank → AE D) : y ∈ latA Sp L (· = x) ↔ alpha S (fun v => (a v)⁻¹) • y ∈ latV Sp L (· = x)

theorem **trace_int_of_dual** {d : D.host.E} (hd : d ∈ Dual D) (o : 𝒪 D.host.E) : ∃ n : ℤ, Algebra.trace ℚ D.host.E (d ∗ o) = n

theorem **mul_mem_dual** {d : D.host.E} (hd : d ∈ Dual D) (o : 𝒪 D.host.E) : d ∗ (o : D.host.E) ∈ Dual D

theorem **chi_tmul_dual** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {γ : Af D.host.F} (hγ : γ ∈ intAf D.host.F) {d : D.host.E} (hd : d ∈ Dual D) : chi ψ (γ ⊗ₜ[D.host.F] d) = 1

theorem **chi_dual** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {γ : Af D.host.F} (hγ : γ ∈ intAf D.host.F) {d : D.host.E} (hd : d ∈ Dual D) {w : Af D.host.F ⊗[D.host.F] D.host.E} (hw : w ∈ intAE D) : chi ψ ((γ ⊗ₜ[D.host.F] d) ∗ w) = 1

theorem **conj_F** (c : 𝒪 D.host.F) : D.host.conj ((c : D.host.F) : D.host.E) = ((c : D.host.F) : D.host.E)

theorem **conj_ne_zero** {a : D.host.E} (ha : a ≠ 0) : D.host.conj a ≠ 0

theorem **claimC** (a : D.host.E) (ha0 : a ≠ 0) (c : 𝒪 D.host.F) (hcb : ∀ b ∈ D.host.AA, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) ∗ b = a ∗ (o : D.host.E)) {z : D.host.E} (hz : z ∈ Dual D) : ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) ∗ ((c : D.host.F) : D.host.E) ∗ (D.host.xi0⁻¹ ∗ a⁻¹ ∗ (D.host.conj a)⁻¹) ∗ z = (o : D.host.E)

theorem **claimC'** (a : D.host.E) (c : 𝒪 D.host.F) (hca : ((c : D.host.F) : D.host.E) ∗ a ∈ D.host.AA) : ((c : D.host.F) : D.host.E) ∗ ((c : D.host.F) : D.host.E) ∗ (D.host.xi0 ∗ a ∗ D.host.conj a) ∈ Dual D

theorem **cA_mul_ex** (ρ : AE D) (x : Pl D) (t : D.host.E) (h : ρ ∗ ex x = ex x ∗ incA D t) : cA D ρ ∗ ex x = ex x ∗ incA D (D.host.conj t)

theorem **B_mul_ex** (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (x : Pl D) : incA D D.host.xi0⁻¹ ∗ alpha S (fun v => (a v)⁻¹) ∗ cA D (alpha S (fun v => (a v)⁻¹)) ∗ ex x = ex x ∗ incA D (D.host.xi0⁻¹ ∗ (a x)⁻¹ ∗ (D.host.conj (a x))⁻¹)

theorem **Binv_mul_ex** (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (x : Pl D) : incA D D.host.xi0 ∗ alpha S a ∗ cA D (alpha S a) ∗ ex x = ex x ∗ incA D (D.host.xi0 ∗ a x ∗ D.host.conj (a x))

theorem **Binv_mul_B** (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (h0 : ∀ v, a v ≠ 0) : (incA D D.host.xi0 * alpha S a * cA D (alpha S a)) * (incA D D.host.xi0⁻¹ * alpha S (fun v => (a v)⁻¹) * cA D (alpha S (fun v => (a v)⁻¹))) = 1

theorem **mul_of_ex** (ρ : AE D) (x : Pl D) (t : D.host.E) (h : ρ * ex x = ex x * incA D t) {z : AE D} (hz : z ∈ idAt D (· = x)) : ρ * z = ex x * incA D t * z

theorem **intAt_mul** {x : Pl D} {z w : AE D} (hz : z ∈ intAt D (· = x)) (hw : w ∈ intAt D (· = x)) : z * w ∈ intAt D (· = x)

theorem **coeC** (c : 𝒪 D.host.F) : (((c * c : 𝒪 D.host.F) : D.host.F) : D.host.E) = ((c : D.host.F) : D.host.E) * ((c : D.host.F) : D.host.E)

theorem **C_not_mem** {x : Pl D} {c : 𝒪 D.host.F} (hc : c ∉ x.asIdeal) : c * c ∉ x.asIdeal

theorem **indicatorIdempotent_eq_single** (x : Pl D) : cP (D

theorem **finPart_incA** (o : D.host.E) : finPart D (incA D o) = (1 : Af D.host.F) ⊗ₜ[D.host.F] o

theorem **assoc4** {R : Type*} [CommRing R] (p q s o : R) (h : p * q = 1) : s * o = p * (q * s * o)

theorem **mem_of_split** (x : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ x.asIdeal) (K x0 : D.host.E) {u : Af D.host.F ⊗[D.host.F] D.host.E} (hu : u ∈ intAE D) (o1 o2 : 𝒪 D.host.E) (h1 : K = o1) (h2 : K * x0 = o2) : (gam x c ⊗ₜ[D.host.F] K) * ((1 : Af D.host.F) ⊗ₜ[D.host.F] x0 + u) ∈ intAE D

theorem **finPart_ex** (x : Pl D) (c : 𝒪 D.host.F) (hc0 : ((c : D.host.F) : D.host.E) ≠ 0) (t : D.host.E) : finPart D (ex x * incA D t) = gam x c ⊗ₜ[D.host.F] (((c : D.host.F) : D.host.E) * t)

theorem **finitePartTraceCharacter_idempotent_inverseDifferent** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (x : Pl D) (c : 𝒪 D.host.F) (hc : c ∉ x.asIdeal) (t : D.host.E) (ht : ((c : D.host.F) : D.host.E) * t ∈ Dual D) (w : AE D) (hw : finPart D w ∈ intAE D) : chi ψ (finPart D (ex x * incA D t * w)) = 1

theorem **cond_iff** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (S : Finset (Pl D)) (a : Pl D → D.host.E) (ha : ∀ v ∉ S, a v = 1) (h0 : ∀ v, a v ≠ 0) (hgen : ∀ v, ∃ c : 𝒪 D.host.F, c ∉ v.asIdeal ∧ ((c : D.host.F) : D.host.E) * a v ∈ D.host.AA ∧ ∀ b ∈ D.host.AA, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) * b = a v * (o : D.host.E)) (x : Pl D) (s : AE D) (hs : s ∈ idAt D (· = x)) : (∀ o ∈ intAt D (· = x), psiE ψ (s * o) = 1) ↔ incA D D.host.xi0⁻¹ * alpha S (fun v => (a v)⁻¹) * cA D (alpha S (fun v => (a v)⁻¹)) * s ∈ intAt D (· = x)

theorem **exists_idele_AA** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∃ α : (AE D)ˣ, ∀ x : Pl D, (∀ (S : HSp D) (L : Lat S) (y : Fin S.rank → AE D), y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) ∧ ∀ s ∈ idAt D (· = x), ((∀ o ∈ intAt D (· = x), psiE ψ (s * o) = 1) ↔ incA D D.host.xi0⁻¹ * ((α⁻¹ : (AE D)ˣ) : AE D) * cA D ((α⁻¹ : (AE D)ˣ) : AE D) * s ∈ intAt D (· = x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteQuotientAnnihilatorStandardLatticeConjHalf

theorem **symplO_add_right'** (S T : HSp D) (x u v : WO S T) : symplO S T x (u + v) = symplO S T x u + symplO S T x v

theorem **symplO_neg_right'** (S T : HSp D) (x u : WO S T) : symplO S T x (-u) = -symplO S T x u

theorem **symplO_zero_right'** (S T : HSp D) (x : WO S T) : symplO S T x 0 = 0

theorem **half_nsmul** {m n : ℕ} {f : AE D} {u : Mx D m n} (hu : IsHalf f u) (N : ℕ) : IsHalf f (N • u)

theorem **nsmul_eq_zero_WO** (S T : HSp D) {N : ℕ} (hN : 0 < N) (v : WO S T) (h : N • v = 0) : v = 0

theorem **ann_conj_half_finite** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∀ k : ℕ, 0 < k → ∃ R : Finset (WO S T), ∀ u : WO S T, IsHalf (cA D E₂) u → (∀ x ∈ stdHom S T, IsHalf E₂ x → ψ (symplO S T x u) = 1) → ∃ r ∈ R, ∃ v : WO S T, IsHalf (cA D E₂) v ∧ (∀ x ∈ stdHom S T, IsHalf E₂ x → ψ (symplO S T x v) = 1) ∧ u = r + k • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiniteQuotientAnnihilatorStandardLatticeConjHalfStatement

theorem **_root_.GlobalTheory.T.finite_quotient_annihilator_standardLattice_conjHalf** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (E₂ : AE D), IsIdemSum E₂ Q → ∀ k : ℕ, 0 < k → ∃ R : Finset (WO S T), ∀ u : WO S T, IsHalf (cA D E₂) u → (∀ x ∈ stdHom S T, IsHalf E₂ x → ψ (symplO S T x u) = 1) → ∃ r ∈ R, ∃ v : WO S T, IsHalf (cA D E₂) v ∧ (∀ x ∈ stdHom S T, IsHalf E₂ x → ψ (symplO S T x v) = 1) ∧ u = r + k • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinrankKerLeHilbertSchmidtNormSqSubId

theorem **finrank_ker_le_hsNormSq_sub_id** {V : Type} [NormedAddCommGroup V] [InnerProductSpace ℂ V] [FiniteDimensional ℂ V] (K : V →L[ℂ] V) (N : Submodule ℂ V) (hN : ∀ x ∈ N, K x = 0) : (Module.finrank ℂ N : ℝ) ≤ hsNormSq (K - ContinuousLinearMap.id ℂ V)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinrankKerLeHilbertSchmidtNormSqSubIdStatement

theorem **_root_.GlobalTheory.T.finrank_ker_le_hilbertSchmidtNormSq_sub_id** : ∀ {V : Type} [NormedAddCommGroup V] [InnerProductSpace ℂ V] [FiniteDimensional ℂ V] (K : V →L[ℂ] V) (N : Submodule ℂ V), (∀ x ∈ N, K x = 0) → (Module.finrank ℂ N : ℝ) ≤ hsNormSq (K - ContinuousLinearMap.id ℂ V)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FinrankKerPuLeHsDev

theorem **finrank_ker_Pu_le_hsDev** (D : D7) (u : D.what₂.Word) : (Module.finrank ℂ (LinearMap.ker (D.s.Pu u : D.s.Cu u →ₗ[ℂ] D.s.HH)) : ℝ) ≤ hsDev D u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstCoupledKernelFiniteDatumLevelSize

theorem **smallDatum_xiCpl1_sharp** : ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ (al : I.β (N.R.m - 2)) (p₀ : ChPt (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstCoupledKernelFiniteDatumLevelSizeStatement

theorem **_root_.GlobalTheory.T.firstCoupledKernel_finiteDatum_level_size** : ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu :

Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ (al : I.β (N.R.m - 2)) (p₀ : ChPt (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstIsotropicVectorCoordinatesNeZero

theorem **coord_first_isotropic_ne_zero** : ∀ {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (hr : 1 ≤ D.R.rS), (fun i : Fin (D.sp.Y 0).rank => X.ιE (X.eR (0, hr) (Fin.cast (GlobalTheory.Core.Mi.rank_eq X h) i))) ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstIsotropicVectorCoordinatesNeZeroStatement

theorem **_root_.GlobalTheory.T.firstIsotropicVector_coordinates_ne_zero** : ∀ {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (hr : 1 ≤ D.R.rS), (fun i : Fin (D.sp.Y 0).rank => X.ιE (X.eR (0, hr) (Fin.cast (GlobalTheory.Core.Mi.rank_eq X h) i))) ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstOccurrenceRuleThetaChainStepBothDirections

theorem **rulePar_of_pairPar** {X : ProvA} (h3 : Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisUpDownLine X) (h4 : Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFourth X) (h5 : Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFifth X) {S S' : SPar} {aS aT : ℚ} (h : PairPar S S' aS aT) : RulePar X S S' aS aT

theorem **rulePar_step** : ∀ (D : D7) (X : ProvA), Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisUpDownLine X → Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFourth X → Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFifth X → ∀ k : Fin D.R.m, ChainPar D k → ∀ τ, RulePar X (parB (Ptw D) k τ) (parB (Ptw D) (k + 1) τ) (expS D k τ) (expT D k τ) ∧ RulePar X (parB (Ptw D) (k + 1) τ) (parB (Ptw D) k τ) (expT D k τ) (expS D k τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstOccurrenceRuleThetaChainStepBothDirectionsStatement

theorem **_root_.GlobalTheory.T.firstOccurrenceRule_thetaChainStep_both_directions** : ∀ (D : D7) (X : ProvA), Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisUpDownLine X → Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFourth X → Formalism.MainNotions.SchurIrreducible.ArchimedeanRallisFifth X → ∀ k : Fin D.R.m, ChainPar D k → ∀ τ, RulePar X (parB (Ptw D) k τ) (parB (Ptw D) (k + 1) τ) (expS D k τ) (expT D k τ) ∧ RulePar X (parB (Ptw D) (k + 1) τ) (parB (Ptw D) k τ) (expT D k τ) (expS D k τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstStepCompletionRepresentativesEqOne

theorem **toLoc_id** {X : Fields} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) {A : Type} [DecidableEq A] {n : ℕ} (e : A ≃ Fin n) : toLoc ι e e (LinearMap.id : (Fin n → X.E) →ₗ[X.E] (Fin n → X.E)) = 1

theorem **exists_isOrdY_zero** (R : Ranks) (c : ℕ) (hc : c = cOf R 0) : ∃ e : XIx (Fin R.k) (Fin R.t) 0 ⊕ Fin c ≃ Fin (R.N 0 + cOf R 0), IsOrdY e

theorem **reps_zero** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstStepCompletionRepresentativesEqOneStatement

theorem **_root_.GlobalTheory.T.firstStep_completionRepresentatives_eq_one** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstTermExistenceOfUniversal

theorem **hasIw_of_good** {T T₀ : HSp D} {r : ℕ} (d : HypDec T T₀ r) {l : Lvl T} (hl : l.Good) : HasIw d l.K ∧ IsCompact (l.K : Set (GA D T))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstTermThetaHeckeIsotropic

theorem **theta_chart** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {A B : HSp D} {χ : (AE D)ˣ →* Circle} {om : GA D (dblSp A) × GA D B →* Module.End ℂ (WO (dblSp A) B → ℂ)} {ν : Measure (WO A B)} {𝓎 : Set (WO A B)} {μ : D.host.E} {κ : GA D (dblSp A) ≃* StdUnn.G D A.rank} {omS : StdUnn.G D A.rank →* Module.End ℂ (schwartzA θ B.rank A.rank)} (P : Pass ψ θ A B χ om ν 𝓎 μ κ omS) {f : WO (dblSp A) B → ℂ} (hf : IsThetaFunO ψ (dblSp A) B f) (Φ : schwartzA θ B.rank A.rank) (hΦ : Φ.1 = sch ψ A B ν 𝓎 f) (g p : GA D (dblSp A)) (h : GA D B) : StdUnn.theta omS Φ (κ (g * p)) h = thetaKO om g h (om (p, 1) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstVariableSupportFiniteCellsAtTwistPlaces

theorem **locK_congr** {D : D7} {c : Comm D} {v : VCom c} {χ : ℕ → (AE D)ˣ →* Circle} (Q : QAt v χ) (j d d' : ℕ) (e : d = d') (h1 : j + d = Q.s + 1) (h2 : j + d' = Q.s + 1) : (Q.T.lev j d h1).J.map (Q.gl j d h1) = (Q.T.lev j d' h2).J.map (Q.gl j d' h2)

theorem **cells_at** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t) (F : QFam v t), c.Is → t.Is → v.Is → WeilOK (t

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t) (F : QFam v t), c.Is → t.Is → v.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FirstVariableSupportFiniteCellsAtTwistPlacesStatement

theorem **_root_.GlobalTheory.T.firstVariable_support_finite_cells_at_twist_places** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t) (F : QFam v t), c.Is → t.Is → v.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FiveLeHalfKfk

theorem **exp_neg_five_lt** : Real.exp (-5) < 1 / 128

theorem **five_le_half_kfk** (D : D7) : 5 ≤ kfk D / 2

theorem **two_eDiag_lt** : ∀ (D : D7) (u : D.what₂.Word) (L C : ℝ), 5 ≤ C * mfL D L → 4 * D.R.f + 1 ≤ (D.mem u).A.T → 2 * eDiag D L C < 1 - (D.R.f : ℝ) / (((D.mem u).A.T : ℝ) - 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FixLevCompactOpen

theorem **continuous_mt** (V : HSp D) : Continuous (mt V)

theorem **map_mt_mul** (q : SplitPt D k) (V : HSp D) (g h : GA D V) : (mt V (g * h)).map q.pr = (mt V g).map q.pr * (mt V h).map q.pr

theorem **frame_mul** {N : ℕ} (F : GlobalTheory.LocMod.Frame k N Y) (A B : Matrix (Fin N) (Fin N) k) : F.γ * A * F.β * (F.γ * B * F.β) = F.γ * (A * B) * F.β

theorem **frame_cancel** {N : ℕ} (F : GlobalTheory.LocMod.Frame k N Y) {A B : Matrix Y Y k} (h : F.β * A * F.γ = F.β * B * F.γ) : A = B

theorem **frame_conj** {N : ℕ} (F : GlobalTheory.LocMod.Frame k N Y) (A : Matrix (Fin N) (Fin N) k) : F.β * (F.γ * A * F.β) * F.γ = A

theorem **locHom_val** (q : SplitPt D k) (V : HSp D) (F : GlobalTheory.LocMod.Frame k V.rank Y) (g : GA D V) : (locHom q V F g).val = F.γ * (mt V g).map q.pr * F.β

theorem **locHom_spec** (q : SplitPt D k) (V : HSp D) (F : GlobalTheory.LocMod.Frame k V.rank Y) (g : GA D V) : (mt V g).map q.pr = F.β * (locHom q V F g).val * F.γ

theorem **continuous_locHom** (q : SplitPt D k) (V : HSp D) (F : GlobalTheory.LocMod.Frame k V.rank Y) : Continuous (locHom q V F)

theorem **list_prod_single** {M : Type} [Monoid M] {ι : Type} [DecidableEq ι] (f : ι → M) (i : ι) (hf : ∀ i', i' ≠ i → f i' = 1) : ∀ l : List ι, l.Nodup → i ∈ l → (l.map f).prod = f i | [], _, h => absurd h List.not_mem_nil | a :: l, hl, h => by rw [List.nodup_cons] at hl rw [List.map_cons, List.prod_cons] by_cases ha : a = i · subst ha rw [List.prod_eq_one fun y hy => ?_, mul_one] obtain ⟨b, hb, rfl⟩

theorem **map_mt_list_prod** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) (V : HSp D) (l : List (GA D V)) : (mt V l.prod).map q.pr = (l.map fun g => (mt V g).map q.pr).prod

theorem **exists_section** (V : HSp D) {n : ℕ} (k : Fin n → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, SplitPt D (k i)) (hq : Function.Injective fun i => (q i).v) (Y : Fin n → Type) [∀ i, Fintype (Y i)] [∀ i, DecidableEq (Y i)] (F : ∀ i, GlobalTheory.LocMod.Frame (k i) V.rank (Y i)) : ∃ Ψ : (∀ i, GL (Y i) (k i)) → GA D V, Continuous Ψ ∧ (∀ x, Ψ x ∈ grpAt V (fun p => ∃ i, p = (q i).v)) ∧ (∀ x i, (mt V (Ψ x)).map (q i).pr = (F i).β * (x i).val * (F i).γ) ∧ ∀ g ∈ grpAt V (fun p => ∃ i, p = (q i).v), ∀ x, (∀ i, (mt V g).map (q i).pr = (F i).β * (x i).val * (F i).γ) → Ψ x = g

theorem **fixLev_compact_open** : ∀ (D : D7) (c : Comm D) (v : VCom c) (n : ℕ) (G : Fin n → GLoc v), GPin G → ∀ j, j < D.R.m → v.JG j ≤ grpAt (c.V j) (· ∈ v.gP) ∧ IsCompact (v.JG j : Set (GA D (c.V j))) ∧ IsOpen (((v.JG j).subgroupOf (grpAt (c.V j) (· ∈ v.gP)) : Subgroup _) : Set (grpAt (c.V j) (· ∈ v.gP)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagBasisLiftGramSignedPermutation

theorem **hatL_eq** (I : AutInst) : Formalism.L2.Herm.hatL (F

theorem **col_mem** (I : AutInst) {β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (hF : IsFrame β I.L) (j : Fin I.n) : (fun i => (β : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) i j) ∈ I.Lf

theorem **Z_of_eq** {a b : (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (h : a = b) (hb : ZMod' I g b) : ZMod' I g a

theorem **Z_symm** {x y : VA I} (h : ZMod' I g (hA I x y)) : ZMod' I g (hA I y x)

theorem **Z_symm_sub** {x y : VA I} {c : (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (h : ZMod' I g (hA I x y - c)) : ZMod' I g (hA I y x - sigA I c)

theorem **Z_sub_zero** {a : (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (h : ZMod' I g a) : ZMod' I g (a - 0)

theorem **Z_addR** {x y z : VA I} (h : ZMod' I g (hA I x y)) (h' : ZMod' I g (hA I x z)) : ZMod' I g (hA I x (y + z))

theorem **idem_sum** {z0 : VA I} {ew : (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (hz : z0 ∈ I.Lf) (hzz : ZMod' I g (hA I z0 z0 + 1)) (hint : ew ∈ IntAf I.F I.E) (hfull : ∀ y ∈ I.Lf, (1 - ew - sigA I ew) • y ∈ mL I g) : ZMod' I g (1 - ew - sigA I ew)

theorem **middle** {k : ℕ} {bw bwd : Fin k → VA I} {ew : (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)} (horth : ew * sigA I ew = 0) (hbw : ∀ i, bw i = ew • bw i) (hbwd : ∀ i, bwd i = sigA I ew • bwd i) (hdual : ∀ i j, ZMod' I g (hA I (bwd i) (bw j) - if i = j then ew else 0)) (hsum : ZMod' I g (1 - ew - sigA I ew)) (l m : Fin k) : ZMod' I g (hA I (bw l + bwd l.rev) (bw m + bwd m.rev) - if m = l.rev then 1 else 0)

theorem **flagVec_cases** {V : Type} [AddCommGroup V] {ρ k : ℕ} (e0 f0 : Fin ρ → V) (z0 : V) (bw bwd : Fin k → V) (j : ℕ) (hj : j < 2 * ρ + k + 1) : (∃ a : Fin ρ, j = a.1 ∧ dualIdx ρ k j = 2 * ρ + k - a.1 ∧ flagVec e0 f0 z0 bw bwd j = e0 a) ∨ (∃ l : Fin k, j = ρ + l.1 ∧ dualIdx ρ k j = ρ + (k - (l.1 + 1)) ∧ flagVec e0 f0 z0 bw bwd j = bw l + bwd l.rev) ∨ (j = ρ + k ∧ dualIdx ρ k j = ρ + k ∧ flagVec e0 f0 z0 bw bwd j = z0) ∨ (∃ a : Fin ρ, j = 2 * ρ + k - a.1 ∧ dualIdx ρ k j = a.1 ∧ flagVec e0 f0 z0 bw bwd j = f0 a)

theorem **flagGram_zero** (h : j.val ≠ dualIdx ρ k i.val) : flagGram R n ρ k i j = 0

theorem **flagGram_neg** (h : j.val = dualIdx ρ k i.val) (h' : i.val = ρ + k) : flagGram R n ρ k i j = -1

theorem **flagGram_ite** {p : Prop} [Decidable p] (h : j.val = dualIdx ρ k i.val ↔ p) (h' : i.val ≠ ρ + k) : flagGram R n ρ k i j = if p then 1 else 0

theorem **flagVec_mem** (hR : IsRefFrame I g e0 f0 z0 bw bwd ew) (j : ℕ) (hj : j < 2 * ρ + k + 1) : flagVec e0 f0 z0 bw bwd j ∈ I.Lf

theorem **flagVec_gram** (hR : IsRefFrame I g e0 f0 z0 bw bwd ew) {n : ℕ} (hn : n = 2 * ρ + k + 1) (i j : Fin n) : ZMod' I g (hA I (flagVec e0 f0 z0 bw bwd i.val) (flagVec e0 f0 z0 bw bwd j.val) - flagGram (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) n ρ k i j)

theorem **gramIn_lift** : ∀ (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)), IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → ∀ β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsLiftFrame I g (flagVec e0 f0 z0 bw bwd) β → ZMod' I g (1 - ew - sigA I ew) ∧ (∀ i j, ZMod' I g (gramIn I β i j - flagGram (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) I.n ρ k i j))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagBasisLiftGramSignedPermutationStatement

theorem **_root_.GlobalTheory.T.liftFrame_gram_congr_flagGram** : ∀ (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → ∀ β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsLiftFrame I g (flagVec e0 f0 z0 bw bwd) β → ZMod' I g (1 - ew - sigA I ew) ∧ (∀ i j, ZMod' I g (gramIn I β i j - flagGram (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) I.n ρ k i j))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagLevelCharacterEqPsiDivN

theorem **isRefFrame_setting** {D : Formalism.BasicNotions.D7} (X : HidaSetting D) : IsRefFrame X.B.I X.B.g X.e0 X.f0 X.z0 X.bw X.bwd X.ew

theorem **sigInt_coe** (I : AutInst) (s : Formalism.L2.Herm.intAf I.F I.E) : ((sigInt I s : Formalism.L2.Herm.intAf I.F I.E) : I.E ⊗[I.F] Af I.F) = sigA I s

theorem **flagLevelCharacter_eq_psi_divN** (I : AutInst) (psiEf : I.E ⊗[I.F] Af I.F → ℂ) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (x : I.Gf) : chi2f I psiEf g e0 f0 z0 bw bwd x = psiEf (divN I g (chiArg I e0 f0 z0 bw bwd x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagRootSumDualEqSymmetricEdgeSum

theorem **dualIdx_lt** {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (i : ℕ) (hi : i < n) : dualIdx ρ k i < n

theorem **dualIdx_invol** {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (i : ℕ) (hi : i < n) : dualIdx ρ k (dualIdx ρ k i) = i

theorem **tau_val** {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (i : Fin n) : (tau hn i).val = dualIdx ρ k i.val

theorem **tau_tau** {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (i : Fin n) : tau hn (tau hn i) = i

theorem **sgn_tau** (K : Type) [CommRing K] {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (i : Fin n) : sgn K ρ k (tau hn i) = sgn K ρ k i

theorem **sgn_mul_self** (K : Type) [CommRing K] (ρ k : ℕ) {n : ℕ} (i : Fin n) : sgn K ρ k i * sgn K ρ k i = 1

theorem **flagGram_apply** (K : Type) [CommRing K] {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (i j : Fin n) : flagGram K n ρ k i j = if j = tau hn i then sgn K ρ k i else 0

theorem **flagGram_mul** (K : Type) [CommRing K] {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (X : Matrix (Fin n) (Fin n) K) (i j : Fin n) : (flagGram K n ρ k * X) i j = sgn K ρ k i * X (tau hn i) j

theorem **mul_flagGram** (K : Type) [CommRing K] {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) (X : Matrix (Fin n) (Fin n) K) (i j : Fin n) : (X * flagGram K n ρ k) i j = X i (tau hn j) * sgn K ρ k j

theorem **flagGram_mul_self** (K : Type) [CommRing K] {ρ k n : ℕ} (hn : n = 2 * ρ + k + 1) : flagGram K n ρ k * flagGram K n ρ k = 1

theorem **inv_adj** {K : Type} [CommRing K] {n ρ k : ℕ} {M : GL (Fin n) K} (hM : M ∈ unipSym K n ρ k) (i j : Fin n) (hij : j.val = i.val + 1) : ((M⁻¹ : GL (Fin n) K) : Matrix (Fin n) (Fin n) K) i j = - (M : Matrix (Fin n) (Fin n) K) i j

theorem **dual_entry** {K : Type} [CommRing K] {n ρ k : ℕ} (hn : n = 2 * ρ + k + 1) (M M' : GL (Fin n) K) (h : (M' : Matrix (Fin n) (Fin n) K)ᵀ * flagGram K n ρ k * (M : Matrix (Fin n) (Fin n) K) = flagGram K n ρ k) (i j : Fin n) : (M' : Matrix (Fin n) (Fin n) K) (tau hn j) (tau hn i) = sgn K ρ k i * ((M⁻¹ : GL (Fin n) K) : Matrix (Fin n) (Fin n) K) i j * sgn K ρ k j

theorem **sum_reindex** {K : Type} [CommRing K] {n ρ k : ℕ} (hn : n = 2 * ρ + k + 1) (f : Fin n → Fin n → K) : ∑ i, ∑ j, f i j = ∑ i, ∑ j, f (tau hn j) (tau hn i)

theorem **rootSum_dual_edges** : ∀ (K : Type) [CommRing K] (n ρ k : ℕ), n = 2 * ρ + k + 1 → ∀ ε ε' a₁ a₂ : K, (ε = 1 ∧ ε' = 0) ∨ (ε = 0 ∧ ε' = 1) → a₁ ≠ 0 → a₂ ≠ 0 → ∃ a : Fin n → Fin n → K, (∀ i j, symEdge ρ k i j → a i j ≠ 0) ∧ ∀ M M' : GL (Fin n) K, (M' : Matrix (Fin n) (Fin n) K)ᵀ * flagGram K n ρ k * (M : Matrix (Fin n) (Fin n) K) = flagGram K n ρ k → M ∈ unipSym K n ρ k → a₁ * rootSum ρ k ε (M : Matrix (Fin n) (Fin n) K) + a₂ * rootSum ρ k ε' (M' : Matrix (Fin n) (Fin n) K) = ∑ i, ∑ j, if symEdge ρ k i j then a i j * (M : Matrix (Fin n) (Fin n) K) i j else 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagRootSumDualEqSymmetricEdgeSumStatement

theorem **_root_.GlobalTheory.T.flagRootSum_dual_eq_symmetricEdge_sum** : ∀ (K : Type) [CommRing K] (n ρ k : ℕ), n = 2 * ρ + k + 1 → ∀ ε ε' a₁ a₂ : K, (ε = 1 ∧ ε' = 0) ∨ (ε = 0 ∧ ε' = 1) → a₁ ≠ 0 → a₂ ≠ 0 → ∃ a : Fin n → Fin n → K, (∀ i j, symEdge ρ k i j → a i j ≠ 0) ∧ ∀ M M' : GL (Fin n) K, (M' : Matrix (Fin n) (Fin n) K)ᵀ * flagGram K n ρ k * (M : Matrix (Fin n) (Fin n) K) = flagGram K n ρ k → M ∈ unipSym K n ρ k → a₁ * rootSum ρ k ε (M : Matrix (Fin n) (Fin n) K) + a₂ * rootSum ρ k ε' (M' : Matrix (Fin n) (Fin n) K) = ∑ i, ∑ j, if symEdge ρ k i j then a i j * (M : Matrix (Fin n) (Fin n) K) i j else 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagStabilizerEqUnipotentTorusLeviAtPlace

theorem **cA_mem_idAt'** {v : Pl D} {y : AE D} (h : y ∈ idAt D (· = v)) : cA D y ∈ idAt D (· = v)

theorem **mul_left_mem** {R : Type} [CommRing R] {I : Ideal R} {l m k : ℕ} (A : Matrix (Fin l) (Fin m) R) {M : Matrix (Fin m) (Fin k) R} (h : ∀ i j, M i j ∈ I) (i : Fin l) (j : Fin k) : (A * M) i j ∈ I

theorem **mul_right_mem** {R : Type} [CommRing R] {I : Ideal R} {l m k : ℕ} {M : Matrix (Fin l) (Fin m) R} (B : Matrix (Fin m) (Fin k) R) (h : ∀ i j, M i j ∈ I) (i : Fin l) (j : Fin k) : (M * B) i j ∈ I

theorem **flag_stab** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (g : GA D S), g ∈ grpAt S (· = v) → (∃ a : Mx D ρ ρ, (∀ i j, j < i → a i j = 0) ∧ mt S g · d.jX = d.jX · a) → ∃ n ∈ flagN d, n ∈ grpAt S (· = v) ∧ ∃ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) ∧ d.levi t ∈ grpAt S (· = v) ∧ ∃ γ ∈ grpAt S₀ (· = v), g = n * d.levi t * d.emb0 γ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagStabilizerEqUnipotentTorusLeviAtPlaceStatement

theorem **_root_.GlobalTheory.T.flagStabilizer_eq_unipotent_torus_levi_atPlace** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : Pl D) (g : GA D S), g ∈ grpAt S (· = v) → (∃ a : Mx D ρ ρ, (∀ i j, j < i → a i j = 0) ∧ mt S g · d.jX = d.jX · a) → ∃ n ∈ flagN d, n ∈ grpAt S (· = v) ∧ ∃ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) ∧ d.levi t ∈ grpAt S (· = v) ∧ ∃ γ ∈ grpAt S₀ (· = v), g = n * d.levi t * d.emb0 γ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagSubgroupsClosedNormalCocompact

theorem **isClosed_fixer** {k : ℕ} (x : Mx D V.rank k) : IsClosed (fixer V x : Set (GA D V))

theorem **centre_eq** : (centre d : Set (GA D V)) = {g | mt V g · d.jX = d.jX} ∩ {g | mt V g · d.j0 = d.j0}

theorem **isClosed_centre** : IsClosed (centre d : Set (GA D V))

theorem **flag_jX** {k : GA D V} (hk : k ∈ flagN d) : ∃ u ∈ UX D ρ, mt V k · d.jX = d.jX · u.1

theorem **radX_normal** : ((radX d).subgroupOf (flagN d)).Normal

theorem **centre_le** : centre d ≤ flagN d

theorem **centre_normal** : ((centre d).subgroupOf (flagN d)).Normal

theorem **fix_col_iff** (h : GA D V) (i : Fin ρ) : h ∈ fixer V (d.jX.submatrix id fun _ : Fin 1 => i) ↔ ∀ r, (mt V h · d.jX) r i = d.jX r i

theorem **mem_Nfix_iff** (h : GA D V) (j : Fin ρ) : h ∈ Nfix d j ↔ h ∈ flagN d ∧ ∀ l : Fin ρ, l ≤ j → ∀ r, (mt V h · d.jX) r l = d.jX r l

theorem **Nfix_le** (j : Fin ρ) : Nfix d j ≤ flagN d

theorem **Nfix_normal** (j : Fin ρ) : ((Nfix d j).subgroupOf (flagN d)).Normal

theorem **flagN_eq** : (flagN d : Set (GA D V)) = {g | (mt V g · d.jX = d.jX · (d.pX · (mt V g · d.jX)) ∧ ∀ j i : Fin ρ, i ≤ j → (d.pX · (mt V g · d.jX) - 1) j i = 0) ∧ ∀ y : Fin V.rank → AE D, (GlobalTheory.BesselChar.perpMat V d.e).map (incA D) *ᵥ y = 0 → mt V g *ᵥ y - y = d.jX *ᵥ (d.pX *ᵥ (mt V g *ᵥ y - y))}

theorem **isClosed_flagN** : IsClosed (flagN d : Set (GA D V))

theorem **continuous_skPart** : Continuous (skPart (D

theorem **skPart_add_skew** (x : Mx D ρ ρ) (c : skewA D ρ) : skPart (x + c.1) = skPart x + c

theorem **skPart_rat** {x : Mx D ρ ρ} (hx : x ∈ ratMx D ρ ρ) : (skPart x).1 ∈ ratMx D ρ ρ

theorem **skew_reduce** : ∃ K0 : Set (Mx D ρ ρ), IsCompact K0 ∧ ∀ c : skewA D ρ, ∃ s : skewA D ρ, s.1 ∈ ratMx D ρ ρ ∧ ∃ x ∈ K0, c = s + skPart x

theorem **centre_cocompact** : ∃ K : Set (centre d), IsCompact K ∧ ∀ m, ∃ γ ∈ (GQ V).subgroupOf (centre d), ∃ k ∈ K, m = γ * k

theorem **radX_cocompact** : ∃ K : Set (radX d), IsCompact K ∧ ∀ m, ∃ γ ∈ (GQ V).subgroupOf (radX d), ∃ k ∈ K, m = γ * k

theorem **exists_pick** (D : D7) : ∃ C : Set (AE D), IsCompact C ∧ ∃ pick : AE D → D.host.E, pick 0 = 0 ∧ ∀ a, incA D (pick a) + a ∈ C

theorem **unip_cols_cocompact** : ∀ (D : D7) (ρ : ℕ) (j : Fin ρ), ∃ K : Set (UX D ρ), IsCompact K ∧ ∀ u : UX D ρ, (∀ l : Fin ρ, l ≤ j → ∀ s : Fin ρ, u.1.1 s l = (1 : Matrix (Fin ρ) (Fin ρ) (AE D)) s l) → ∃ γ ∈ UXQ D ρ, (∀ l : Fin ρ, l ≤ j → ∀ s : Fin ρ, γ.1.1 s l = (1 : Matrix (Fin ρ) (Fin ρ) (AE D)) s l) ∧ ∃ k ∈ K, u = γ * k

theorem **radX_le_Nfix** (j : Fin ρ) : radX d ≤ Nfix d j

theorem **levi_mem_Nfix** (j : Fin ρ) {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) (hcol : ∀ l : Fin ρ, l ≤ j → ∀ s : Fin ρ, u.1 s l = (1 : Matrix (Fin ρ) (Fin ρ) (AE D)) s l) : d.levi u ∈ Nfix d j

theorem **col_of_mem_Nfix** (j : Fin ρ) {u : GL (Fin ρ) (AE D)} {b : Mx D ρ V₀.rank} {c : skewA D ρ} (h : d.levi u * (d.nB b * d.nC c) ∈ Nfix d j) : ∀ l : Fin ρ, l ≤ j → ∀ s : Fin ρ, u.1 s l = (1 : Matrix (Fin ρ) (Fin ρ) (AE D)) s l

theorem **Nfix_cocompact** (j : Fin ρ) : ∃ K : Set (Nfix d j), IsCompact K ∧ ∀ m, ∃ γ ∈ (GQ V).subgroupOf (Nfix d j), ∃ k ∈ K, m = γ * k

theorem **flag_subgroups_closed_normal_cocompact** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), IsClosed (flagN d : Set (GA D V)) ∧ ∀ M : Subgroup (GA D V), (M = centre d ∨ M = radX d ∨ ∃ j, M = Nfix d j) → M ≤ flagN d ∧ IsClosed (M : Set (GA D V)) ∧ (M.subgroupOf (flagN d)).Normal ∧ ∃ K : Set M, IsCompact K ∧ ∀ m, ∃ γ ∈ (GQ V).subgroupOf M, ∃ k ∈ K, m = γ * k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagSubgroupsClosedNormalCocompactStatement

theorem **_root_.GlobalTheory.T.flag_subgroups_closed_normal_cocompact'** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), IsClosed (flagN d : Set (GA D V)) ∧ ∀ M : Subgroup (GA D V), (M = centre d ∨ M = radX d ∨ ∃ j, M = Nfix d j) → M ≤ flagN d ∧ IsClosed (M : Set (GA D V)) ∧ (M.subgroupOf (flagN d)).Normal ∧ ∃ K : Set M, IsCompact K ∧ ∀ m, ∃ γ ∈ (GQ V).subgroupOf M, ∃ k ∈ K, m = γ * k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagSupportSplitPlaces

theorem **hA_fR_nxt_zero** (X : HidaSetting D) (u : D.what₂.Word) (a : Fin D.R.rS) : hA X.B.I (incl X.B.I (X.fR a)) (incl X.B.I (nxt X.eR (X.zR u) a)) = 0

theorem **mem_GfC_of_suppAt** {e : AE D} (he : IsSplitIdem D e) {V : HSp D} {g : GA D V} (hs : SuppAt e V g) : g ∈ GfC V

theorem **smul_fact** {R : Type} [CommRing R] {k : ℕ} (M : Matrix (Fin k) (Fin k) R) (ε : R) (h : ∀ i j, M i j = ε * M i j) (v : Fin k → R) : M *ᵥ v = ε • (M *ᵥ v)

theorem **supp_fact** {A R : Type} [CommRing A] [CommRing R] {k : ℕ} (f : A →+* R) (M : Matrix (Fin k) (Fin k) A) (a : A) (h : ∀ i j, a * (M - 1) i j = (M - 1) i j) (v : Fin k → R) : (M.map f - 1) *ᵥ v = f a • ((M.map f - 1) *ᵥ v)

theorem **exists_eps** (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) {e : AE D} {PP : Formalism.L2.DIV.Pl D.host.E} (hi : Formalism.L2.NTheta.IsIdemAt e PP) (hg : ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) (n : GA D (D.sp.Y 0)) (hs : SuppAt e (D.sp.Y 0) n) : ∃ ε : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F, (∀ c : ℕ, Nat.Coprime c X.B.g → ∃ t ∈ IntAf X.B.I.F X.B.I.E, ε = (c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) * t) ∧ ∀ v, (X.B.I.fmat (iotaX X h n).2 - 1) *ᵥ v = ε • ((X.B.I.fmat (iotaX X h n).2 - 1) *ᵥ v)

theorem **besselLevel_chi_conj_eq_finite_bessel_character** (X : HidaSetting D) (u : D.what₂.Word) (m : X.B.I.Gf) (ε : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) (hε : ∀ c : ℕ, Nat.Coprime c X.B.g → ∃ t ∈ IntAf X.B.I.F X.B.I.E, ε = (c : X.B.I.E ⊗[X.B.I.F] Formalism.MainNotions.S.Af X.B.I.F) * t) (hs : ∀ v, (X.B.I.fmat m - 1) *ᵥ v = ε • ((X.B.I.fmat m - 1) *ᵥ v)) (hN : m ∈ X.NfinU u) (hK : (X.xu u)⁻¹ * m * X.xu u ∈ X.K2) : X.chi2 ((X.xu u)⁻¹ * m * X.xu u) = X.psiFinU u m

theorem **rowChi_conj_eq_psiBes** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A), GlobalTheory.Core.Mi.IdE X → ∀ (PP : Formalism.L2.DIV.Pl D.host.E) (e : AE D), IsSplitIdem D e → Formalism.L2.NTheta.IsIdemAt e PP → ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal → ∀ n ∈ flagN B.d, Formalism.L2.Theta.SuppAt e (D.sp.Y 0) n → ((iotaX X h).symm (1, X.xu u))⁻¹ * n * ((iotaX X h).symm (1, X.xu u)) ∈ rowJ X h → rowChi X h (((iotaX X h).symm (1, X.xu u))⁻¹ * n * ((iotaX X h).symm (1, X.xu u))) = psiBes ψ B.d 1 (vecB B.o B.ζ) n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagSupportSplitPlacesStatement

theorem **_root_.GlobalTheory.T.flag_support_split_places** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A), GlobalTheory.Core.Mi.IdE X → ∀ (PP : Formalism.L2.DIV.Pl D.host.E) (e : AE D), IsSplitIdem D e → Formalism.L2.NTheta.IsIdemAt e PP → ((D.host.g 0 * D.host.g 1 : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal → ∀ n ∈ flagN B.d, Formalism.L2.Theta.SuppAt e (D.sp.Y 0) n → ((iotaX X h).symm (1, X.xu u))⁻¹ * n * ((iotaX X h).symm (1, X.xu u)) ∈ rowJ X h → rowChi X h (((iotaX X h).symm (1, X.xu u))⁻¹ * n * ((iotaX X h).symm (1, X.xu u))) = psiBes ψ B.d 1 (vecB B.o B.ζ) n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentFundamentalDomainLayersIntegral

theorem **layers_integral** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ), ∃ 𝒜 : Set (flagN d), IsFundamentalDomain ((GQ V ⊓ radX d).subgroupOf (flagN d)) 𝒜 vN.ν ∧ ∀ f : GA D V → ENNReal, Measurable f → (∀ c, ∀ n ∈ flagN d, f (d.nC c * n) = f n) → ∫⁻ n in 𝒜, f n.1 ∂vN.ν = ∫⁻ u, ∫⁻ b in vb.ℱ, f (d.nB b * d.levi u.1) ∂vb.μ ∂vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentFundamentalDomainLayersIntegralStatement

theorem **_root_.GlobalTheory.T.flagUnipotent_fundamentalDomain_layers_integral** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ), ∃ 𝒜 : Set (flagN d), IsFundamentalDomain ((GQ V ⊓ radX d).subgroupOf (flagN d)) 𝒜 vN.ν ∧ ∀ f : GA D V → ENNReal, Measurable f → (∀ c, ∀ n ∈ flagN d, f (d.nC c * n) = f n) → ∫⁻ n in 𝒜, f n.1 ∂vN.ν = ∫⁻ u, ∫⁻ b in vb.ℱ, f (d.nB b * d.levi u.1) ∂vb.μ ∂vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentHaarEqMapCoordinates

theorem **eq_of_rightInvariant_of_fundamentalDomain** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] [LocallyCompactSpace G] [SecondCountableTopology G] (Γ : Subgroup G) [Countable Γ] (ν m : Measure G) (𝒩 ℱ : Set G) (h1 : ν.IsHaarMeasure) (h2 : IsFundamentalDomain Γ 𝒩 ν) (h3 : ν 𝒩 = 1) (k0 : SigmaFinite m) (k1 : m.IsMulRightInvariant) (k2 : IsFundamentalDomain Γ ℱ m) (k3 : m ℱ = 1) : m = ν

theorem **volN_eq_map_coords** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d), (∀ p, (e p : GA D V) = d.nC p.1 ∗ (d.nB p.2.1 ∗ d.levi p.2.2.1)) → ∀ (lC : Measure (skewA D ρ)) (𝒞 : Set (skewA D ρ)), lC.IsAddHaarMeasure → SFinite lC → IsAddFundamentalDomain ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ)) 𝒞 lC → lC 𝒞 = 1 → vN.ν = (lC.prod (vb.μ.prod vU.ν)).map e

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentHaarEqMapCoordinatesStatement

theorem **_root_.GlobalTheory.T.flagUnipotentHaar_eq_map_coordinates** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d), (∀ p, (e p : GA D V) = d.nC p.1 ∗ (d.nB p.2.1 ∗ d.levi p.2.2.1)) → ∀ (lC : Measure (skewA D ρ)) (𝒞 : Set (skewA D ρ)), lC.IsAddHaarMeasure → SFinite lC → IsAddFundamentalDomain ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ)) 𝒞 lC → lC 𝒞 = 1 → vN.ν = (lC.prod (vb.μ.prod vU.ν)).map e

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentRadicalSurjectsUnitriangularCharact

theorem **fourierBesselBasePoint_castSucc_column** (i : Fin B.V.rank) (j : Fin σ) : x0 B i (Fin.castSucc j) = B.d.jX i j

theorem **fourierBesselBasePoint_last_column** (i : Fin B.V.rank) : x0 B i (Fin.last σ) = (B.d.j0 ∗ᵥ ratV (vecB B.o B.ζ)) i

theorem **mul_fourierBesselBasePoint_castSucc_column** {m : ℕ} (M : Mx D m B.V.rank) (i : Fin m) (j : Fin σ) : (M · x0 B) i (Fin.castSucc j) = (M · B.d.jX) i j

theorem **mul_fourierBesselBasePoint_last_column** {m : ℕ} (M : Mx D m B.V.rank) (i : Fin m) : (M · x0 B) i (Fin.last σ) = (M ∗ᵥ (B.d.j0 ∗ᵥ ratV (vecB B.o B.ζ))) i

theorem **fourierBesselBasePoint_mul_eq** (U : Mx D (σ + 1) (σ + 1)) (i : Fin B.V.rank) (j : Fin (σ + 1)) : (x0 B · U) i j = ∑ k : Fin σ, B.d.jX i k ∗ U (Fin.castSucc k) j + (B.d.j0 ∗ᵥ ratV (vecB B.o B.ζ)) i ∗ U (Fin.last σ) j

theorem **besselAnisotropicEmbedding_fixes_basePoint** (h : GA D B.A) : mt B.V (embA B.d B.o h) · x0 B = x0 B

theorem **upSum_succ** (U : Mx D (σ + 1) (σ + 1)) : upSum U = upSum (fun i j : Fin σ => U (Fin.castSucc i) (Fin.castSucc j)) + ell D σ ·ᵥ (fun i => U (Fin.castSucc i) (Fin.last σ))

theorem **besselFlagProjection_mul_fourierBesselBasePoint_apply** (i : Fin σ) (k : Fin (σ + 1)) : (B.d.pX · x0 B) i k = if k = Fin.castSucc i then 1 else 0

theorem **besselFlagProjection_mul_basePoint_mul_apply** (U : Mx D (σ + 1) (σ + 1)) (i : Fin σ) (k : Fin (σ + 1)) : (B.d.pX · (x0 B · U)) i k = U (Fin.castSucc i) k

theorem **simpleSum_eq_upSum** {n : GA D B.V} {U : Mx D (σ + 1) (σ + 1)} (e : mt B.V n · x0 B = x0 B · U) : simpleSum B.d (vecB B.o B.ζ) n = upSum U

theorem **psiBes_eq_psiUni** (ψ : AddChar (AF D.host.F) ℂ) {n : GA D B.V} (h : GA D B.A) (u : UX D (σ + 1)) (e : mt B.V (n ∗ embA B.d B.o h) · x0 B = x0 B · u.1.1) : psiBes ψ B.d 1 (vecB B.o B.ζ) n = psiUni ψ u.1

theorem **vecB_ne** : vecB B.o B.ζ ≠ 0

theorem **exists_form** : ∃ r₀ : Fin B.V₀.rank → D.host.E, ratV r₀ ·ᵥ ratV (vecB B.o B.ζ) = 1

theorem **vecMulVec_mulVec'** {m k : ℕ} (w : Fin m → AE D) (r v : Fin k → AE D) (hr : r ·ᵥ v = 1) : vecMulVec w r ∗ᵥ v = w

theorem **det_blk** {U : Mx D (σ + 1) (σ + 1)} (hU : ∀ i j : Fin (σ + 1), (j < i → U i j = 0) ∧ U i i = 1) : (blk U).det = 1

theorem **blkU_val** {U : Mx D (σ + 1) (σ + 1)} (hU : ∀ i j : Fin (σ + 1), (j < i → U i j = 0) ∧ U i i = 1) : (blkU hU).1 = blk U

theorem **blkU_mem** {U : Mx D (σ + 1) (σ + 1)} (hU : ∀ i j : Fin (σ + 1), (j < i → U i j = 0) ∧ U i i = 1) : blkU hU ∈ upperUni σ (AE D)

theorem **blk_rat** (γ : Matrix (Fin (σ + 1)) (Fin (σ + 1)) D.host.E) : blk (rat γ : Mx D (σ + 1) (σ + 1)) = rat (blkQ γ)

theorem **blkU_rat** {U : Mx D (σ + 1) (σ + 1)} (hU : ∀ i j : Fin (σ + 1), (j < i → U i j = 0) ∧ U i i = 1) {γ : Matrix (Fin (σ + 1)) (Fin (σ + 1)) D.host.E} (hγ : rat γ = U) : (blkU hU).1 ∈ ratMx D σ σ ∧ ((blkU hU)⁻¹).1 ∈ ratMx D σ σ

theorem **rat_vecMulVec** {m k : ℕ} (w : Fin m → D.host.E) (r : Fin k → D.host.E) : (rat (vecMulVec w r) : Mx D m k) = vecMulVec (ratV w) (ratV r)

theorem **besselFlagElement_mul_basePoint_eq_basePoint_mul** {U : Mx D (σ + 1) (σ + 1)} (hU : ∀ i j : Fin (σ + 1), (j < i → U i j = 0) ∧ U i i = 1) (a : GL (Fin σ) (AE D)) (ha : a.1 = blk U) (b : Mx D σ B.V₀.rank) (hbz : a.1 ∗ᵥ (b ∗ᵥ ratV (vecB B.o B.ζ)) = fun i => U (Fin.castSucc i) (Fin.last σ)) : mt B.V (B.d.levi a ∗ B.d.nB b) · x0 B = x0 B · U

theorem **exists_flag** (u : UX D (σ + 1)) : ∃ n ∈ flagN B.d, mt B.V n · x0 B = x0 B · u.1.1 ∧ (u ∈ UXQ D (σ + 1) → n ∈ GQ B.V)

theorem **uniRad_surj_psi** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∀ u : UX D (σ + 1), ∃ n ∈ flagN B.d, mt B.V n · x0 B = x0 B · u.1.1 ∧ (u ∈ UXQ D (σ + 1) → n ∈ GQ B.V)) ∧ ∀ (ψ : AddChar (AF D.host.F) ℂ), ∀ n ∈ flagN B.d, ∀ (h : GA D B.A) (u : UX D (σ + 1)), mt B.V (n ∗ embA B.d B.o h) · x0 B = x0 B · u.1.1 → psiBes ψ B.d 1 (vecB B.o B.ζ) n = psiUni ψ u.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentRadicalSurjectsUnitriangularCharactStatement

theorem **_root_.GlobalTheory.T.flagUnipotentRadical_surjects_unitriangular_characters_agree** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∀ u : UX D (σ + 1), ∃ n ∈ flagN B.d, mt B.V n · x0 B = x0 B · u.1.1 ∧ (u ∈ UXQ D (σ + 1) → n ∈ GQ B.V)) ∧ ∀ (ψ : AddChar (AF D.host.F) ℂ), ∀ n ∈ flagN B.d, ∀ (h : GA D B.A) (u : UX D (σ + 1)), mt B.V (n ∗ embA B.d B.o h) · x0 B = x0 B · u.1.1 → psiBes ψ B.d 1 (vecB B.o B.ζ) n = psiUni ψ u.1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentUnfoldingBesselFourierJacobiEq

theorem **zEll_mem_ratMx** (T : HSp D) (ρ : ℕ) (z : Fin T.rank → D.host.E) : zEll T ρ z ∈ ratMx D T.rank ρ

theorem **incAdj_yZero_mem** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (z : Fin S.rank → D.host.E) : d.incAdj S (yZero S ρ z) ∈ ratMx D T.rank S.rank

theorem **star_psi** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) (x : AF D.host.F) : star (ψ x) = ψ (−x)

theorem **integrand_eq** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (−1)) {V V₀ W W₀ : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (z : Fin V₀.rank → D.host.E) (f : WO V₀ W → ℂ) (hf : IsThetaFunO ψ V₀ W f) (b : Mx D ρ V₀.rank) : star (psiBes ψ dV 1 z (dV.nB b)) ∗ f (xZero dW · b) = heisO ψ V₀ W (dW.incX V₀ b) (heisO ψ V₀ W (dW.incAdj V₀ (yZero V₀ ρ z)) (heisO ψ V₀ W (dW.inc0 V₀ 0) f)) 0

theorem **fl_eq_flatBF** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → fl ψ B.d J.W s.vS s.vT s.vU (fun n => star (psiBes ψ B.d 1 (vecB B.o B.ζ) n)) (xZero J.d) Ψ = flatBF s Ψ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnipotentUnfoldingBesselFourierJacobiEqStatement

theorem **_root_.GlobalTheory.T.flagUnipotentUnfolding_besselFourierJacobi_eq** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (Ψ : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W Ψ → fl ψ B.d J.W s.vS s.vT s.vU (fun n => star (psiBes ψ B.d 1 (vecB B.o B.ζ) n)) (xZero J.d) Ψ = flatBF s Ψ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnitaryOrbitActionTransitiveWithStabilizer

theorem **exists_leftInv** {m k : ℕ} (x : Matrix (Fin m) (Fin k) D.host.E) (hx : LinearIndependent D.host.E x.transpose) : ∃ L : Matrix (Fin k) (Fin m) D.host.E, L * x = 1

theorem **blockX_spec** {n : GA D V} (hn : n ∈ flagN d) : ∃ u ∈ UX D ρ, blockX d n = u.1 ∧ mt V n · d.jX = d.jX · u.1

theorem **mt_jX** {n : GA D V} (hn : n ∈ flagN d) : mt V n · d.jX = d.jX · blockX d n

theorem **blockX_one** : blockX d 1 = 1

theorem **blockX_mul** {n n' : GA D V} (hn : n ∈ flagN d) (hn' : n' ∈ flagN d) : blockX d (n * n') = blockX d n · blockX d n'

theorem **blockX_rat** {n : GA D V} (hn : n ∈ GQ V) : blockX d n ∈ ratMx D ρ ρ

theorem **cancel_blockX** {n : GA D V} (hn : n ∈ flagN d) {m : ℕ} {y y' : Mx D m ρ} (h : y · blockX d n = y' · blockX d n) : y = y'

theorem **orb_ext** {n : GA D V} (hn : n ∈ flagN d) {y y' : Orb W x₀} (h : (rat y.1 : Mx D W.rank ρ) · blockX d n = rat y'.1 · blockX d n) : y = y'

theorem **exists_act** (hx : IsBase W x₀) (γ : (NFlagQ V d.e).prod (GQ W)) (x : Orb W x₀) : ∃ x' ∈ Orb W x₀, (rat x' : Mx D W.rank ρ) · blockX d γ.1.1.1 = mt W γ.1.2 · rat x.1

theorem **act_spec** (hx : IsBase W x₀) (γ : (NFlagQ V d.e).prod (GQ W)) (x : Orb W x₀) : (rat (act d hx γ x).1 : Mx D W.rank ρ) · blockX d γ.1.1.1 = mt W γ.1.2 · rat x.1

theorem **smul_def** (hx : IsBase W x₀) (γ : (NFlagQ V d.e).prod (GQ W)) (x : Orb W x₀) : @HSMul.hSMul _ _ _ (@instHSMul _ _ (mulAct d hx).toSMul) γ x = act d hx γ x

theorem **base_val** (hx : IsBase W x₀) : (base hx).1 = x₀

theorem **act_base_iff** (hx : IsBase W x₀) (γ : (NFlagQ V d.e).prod (GQ W)) : act d hx γ (base hx) = base hx ↔ (rat x₀ : Mx D W.rank ρ) · blockX d γ.1.1.1 = mt W γ.1.2 · rat x₀

theorem **transitive** (hx : IsBase W x₀) (x : Orb W x₀) : ∃ γ : (NFlagQ V d.e).prod (GQ W), act d hx γ (base hx) = x

theorem **leftInv_rat** (hx : IsBase W x₀) : ∃ L : Matrix (Fin ρ) (Fin W.rank) D.host.E, (rat L : Mx D ρ W.rank) · rat x₀ = 1

theorem **stab_left** (hx : IsBase W x₀) (n : NFlag V d.e) : (∃ γ : (NFlagQ V d.e).prod (GQ W), act d hx γ (base hx) = base hx ∧ γ.1 = (n, 1)) ↔ n.1 ∈ GQ V ⊓ radX d

theorem **stab_right** (hx : IsBase W x₀) (h : GA D W) : (∃ γ : (NFlagQ V d.e).prod (GQ W), act d hx γ (base hx) = base hx ∧ γ.1.2 = h) ↔ h ∈ GQ W ⊓ Rgrp W (rat x₀)

theorem **action_transitive_stabiliser** : ∀ (D : D7) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (hx : IsBase W x₀), ∃ (_ : MulAction ((NFlagQ V d.e).prod (GQ W)) (Orb W x₀)) (y₀ : Orb W x₀), y₀.1 = x₀ ∧ (∀ (γ : (NFlagQ V d.e).prod (GQ W)) (x : Orb W x₀), (rat (γ • x).1 : Mx D W.rank ρ) · blockX d γ.1.1.1 = mt W γ.1.2 · rat x.1) ∧ (∀ x : Orb W x₀, ∃ γ : (NFlagQ V d.e).prod (GQ W), γ • y₀ = x) ∧ ((MulAction.stabilizer ((NFlagQ V d.e).prod (GQ W)) y₀).map (Subgroup.subtype _)).comap (MonoidHom.inl _ _) = (GQ V ⊓ radX d).subgroupOf (flagN d) ∧ ((MulAction.stabilizer ((NFlagQ V d.e).prod (GQ W)) y₀).map (Subgroup.subtype _)).map (MonoidHom.snd _ _) = GQ W ⊓ Rgrp W (rat x₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagUnitaryOrbitActionTransitiveWithStabilizerStatement

theorem **_root_.GlobalTheory.T.flagUnitaryOrbitAction_transitive_with_stabilizer** : ∀ (D : D7) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (hx : IsBase W x₀), ∃ (_ : MulAction ((NFlagQ V d.e).prod (GQ W)) (Orb W x₀)) (y₀ : Orb W x₀), y₀.1 = x₀ ∧ (∀ (γ : (NFlagQ V d.e).prod (GQ W)) (x : Orb W x₀), (rat (γ • x).1 : Mx D W.rank ρ) · blockX d γ.1.1.1 = mt W γ.1.2 · rat x.1) ∧ (∀ x : Orb W x₀, ∃ γ : (NFlagQ V d.e).prod (GQ W), γ • y₀ = x) ∧ ((MulAction.stabilizer ((NFlagQ V d.e).prod (GQ W)) y₀).map (Subgroup.subtype _)).comap (MonoidHom.inl _ _) = (GQ V ⊓ radX d).subgroupOf (flagN d) ∧ ((MulAction.stabilizer ((NFlagQ V d.e).prod (GQ W)) y₀).map (Subgroup.subtype _)).map (MonoidHom.snd _ _) = GQ W ⊓ Rgrp W (rat x₀)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagVecMemAndSpan

theorem **flagVec_mem_and_span** (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) : IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → (∀ j, flagVec e0 f0 z0 bw bwd j ∈ I.Lf) ∧ ∀ y ∈ I.Lf, InMod I g (Set.range fun j : Fin I.n => flagVec e0 f0 z0 bw bwd j.val) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlagVecMemAndSpanStatement

theorem **_root_.GlobalTheory.T.flagVec_mem_Lf_and_inMod** : ∀ (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → (∀ j, flagVec e0 f0 z0 bw bwd j ∈ I.Lf) ∧ ∀ y ∈ I.Lf, InMod I g (Set.range fun j : Fin I.n => flagVec e0 f0 z0 bw bwd j.val) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatBesselToFourierJacobiRowMulLaws

theorem **pr_eq** : p.pr = prH p

theorem **e_mul_ι** (x : k) : p.e * p.ι x = p.ι x

theorem **e_smul_map** {m n : Type} (K : Matrix m n k) : p.e • K.map p.ι = K.map p.ι

theorem **map_ι_π** {m n : Type} (K : Matrix m n k) : (K.map p.ι).map p.π = K

theorem **xi_ne** : p.xi ≠ 0

theorem **hz_ne** {ρ : ℕ} (B : BesLev D ρ) : p.hz B ≠ 0

theorem **char_half** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (h : p.IsChar ψ ψk) {a : AE D} (ha : p.e * a = a) : ψ (Algebra.trace (AF D.host.F) (AE D) (incA D D.host.xi0 * a)) = ψk (p.π a)

theorem **infPart_conj_e** : infPart D (cA D p.e) = 0

theorem **smul_xiPt** (S : HSp D) {ρ : ℕ} (K : Matrix (Fin S.rank) (Fin ρ) k) : cA D p.e • xiPt p S K = xiPt p S K

theorem **mxAt_xiPt** (S : HSp D) {ρ : ℕ} (K : Matrix (Fin S.rank) (Fin ρ) k) : GlobalTheory.FinPl.MxAt (fun _ => True) (xiPt p S K)

theorem **isHalf_xiPt** : IsHalf (cA D p.e) (dV.ext0 W (dW.incX V₀ (xiPt p V₀ K)))

theorem **hO_xiPt** (x : WO V W) : hO V W x (dV.ext0 W (dW.incX V₀ (xiPt p V₀ K))) = (dV.j0 · (K.map p.ι · (dW.pY · x))).trace

theorem **hO_incAdj_xiPt** (y : Mx D V₀.rank σ) : hO V₀ W (dW.incAdj V₀ y) (dW.incX V₀ (xiPt p V₀ K)) = (K.map p.ι · (incA D (-D.host.xi0⁻¹) • (adj y · gA V₀))).trace

theorem **char_up** (h : p.IsChar ψ ψk) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FV : Frame k V.rank Y') (FW : Frame k W.rank Y) (t : Matrix Y Y' k) : ψ (symplO V W (p.up FV FW t) (dV.ext0 W (dW.incX V₀ (xiPt p V₀ K)))) = ψk ((FV.γ ∗ (dV.j0.map p.π ∗ K) ∗ (dW.pY.map p.π ∗ FW.β) ∗ t).trace)

theorem **char_incAdj** (h : p.IsChar ψ ψk) (y : Mx D V₀.rank σ) : ψ (symplO V₀ W (dW.incAdj V₀ y) (dW.incX V₀ (xiPt p V₀ K))) = ψk (-p.xi⁻¹ ∗ (K ∗ (adj y · gA V₀).map p.π).trace)

theorem **mul_col** {l m n Y : Type} [Fintype m] {P : Matrix l m k} {X : Matrix m n k} {M : Matrix m Y k} {j : Y} {i : n} {a : k} (h : ∀ r, M r j = a ∗ X r i) (i₀ : l) : (P ∗ M) i₀ j = a ∗ (P ∗ X) i₀ i

theorem **jac_row** {ρ : ℕ} {J : JacLev D ρ} {s hh : k} {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {G₀ : Frame k J.W₀.rank n} {FW : Frame k J.W.rank (n ⊕ Fin c)} (h : p.IsJacFrame J s hh G₀ FW) (i : Fin ρ) (l : Fin c) (hl : l.val + i.val + 1 = 2 ∗ ρ) (r : n ⊕ Fin c) : (J.d.pY.map p.π ∗ FW.β) i r = if r = Sum.inr l then s ∗ (hh / p.xi) ∗ (-1 / p.xi) ^ (ρ - 1 - i.val) else 0

theorem **core_val** {ρ : ℕ} {B : BesLev D ρ} {s : k} {n : Type} [Fintype n] [DecidableEq n] {FA : Frame k B.A.rank n} {F₀ : Frame k B.V₀.rank (n ⊕ Unit)} (h : p.IsCoreFrame B s FA F₀) (i : Fin ρ) (q : n ⊕ Unit) : ((adj (yZero B.V₀ ρ (vecB B.o B.ζ)) · gA B.V₀).map p.π ∗ F₀.β) i q = if i.val + 1 = ρ ∧ q = Sum.inr () then -(s ∗ p.hz B) else 0

theorem **core_col** {ρ : ℕ} {B : BesLev D ρ} {s : k} {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {FA : Frame k B.A.rank n} {F₀ : Frame k B.V₀.rank (n ⊕ Unit)} {FV : Frame k B.V.rank (n ⊕ Fin c)} (h₀ : p.IsCoreFrame B s FA F₀) (h : p.IsBesFrame B s FA FV) (q : n ⊕ Fin c) (hq : ∀ i, q = Sum.inr i → i.val = ρ) (x : Fin B.V.rank) : (B.d.j0.map p.π ∗ F₀.β) x (q.map id fun _ => ()) = FV.β x q

theorem **trace_single** {m n l : Type} [Fintype m] [Fintype n] [DecidableEq n] [Fintype l] [DecidableEq l] (M : Matrix m n k) (q : n) (i : l) (a : k) (N : Matrix l m k) : (M ∗ Matrix.single q i a ∗ N).trace = a ∗ (N ∗ M) i q

theorem **flatBF_mul_row** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatBesselToFourierJacobiRowMulLawsStatement

theorem **_root_.GlobalTheory.T.flat_bessel_to_fourierJacobi_row_mul_laws** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatColumnIndicatorPureAtSymmetricPosition

theorem **typeFrame_val'** (ϖ : kˣ) (ord : ι ≃ n ⊕ Fin c) : (typeFrame ϖ ord).bas.val = diagonal (Sum.elim (fun _ => 1) fun i : Fin c => (ϖ : k) ^ (i.val + 1))

theorem **typeFrame_inv_val'** (ϖ : kˣ) (ord : ι ≃ n ⊕ Fin c) : ((typeFrame ϖ ord).bas⁻¹).val = diagonal (Sum.elim (fun _ => 1) fun i : Fin c => ((ϖ : k) ^ (i.val + 1))⁻¹)

theorem **coords_entry** (ϖ : kˣ) (o : ι ≃ n ⊕ Fin c) (o' : ι' ≃ n' ⊕ Fin c') (x : Matrix (n ⊕ Fin c) (n' ⊕ Fin c') k) (u : n ⊕ Fin c) (v : n' ⊕ Fin c') : (((typeFrame ϖ o).bas⁻¹).val ∗ x ∗ (typeFrame ϖ o').bas.val) u v = Sum.elim (fun _ => 1) (fun i : Fin c => ((ϖ : k) ^ (i.val + 1))⁻¹) u ∗ x u v ∗ Sum.elim (fun _ => 1) (fun i : Fin c' => (ϖ : k) ^ (i.val + 1)) v

theorem **affPt_col** (p : ℕ) (x₀ : Matrix n n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) (a : n) (q : Fin (c + 1)) (hq : q.val + 1 = p) : affPt p x₀ e (.inl a) (.inr q) = 0

theorem **om_one_one** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →∗ ℂˣ) (Φ : Matrix b a k → ℂ) : om μ (1 : GL a k) (1 : GL b k) Φ = Φ

theorem **le_of_mul_pow_lt** {ϖ : kˣ} (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (r : ℕ) {y : k} (h : valuation k (y ∗ (ϖ : k) ^ (r + 1)) < 1) : valuation k y ≤ valuation k (((ϖ ^ r)⁻¹ : kˣ) : k)

theorem **mul_pow_lt_of_le** {ϖ : kˣ} (hϖ : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (r : ℕ) {y : k} (h : valuation k y ≤ valuation k (((ϖ ^ r)⁻¹ : kˣ) : k)) : valuation k (y ∗ (ϖ : k) ^ (r + 1)) < 1

theorem **Phi_add_colAt** (j c : ℕ) (H : j + 1 + c = s) (r : ℕ) (y : XIx (Fin κ) t (j + 1) → k) (hy : y ∈ box (XIx (Fin κ) t (j + 1)) (S.ϖ ^ r)⁻¹) (z : Matrix (XIx (Fin κ) t (j + 1) ⊕ Fin (c + 1)) (XIx (Fin κ) t j ⊕ Fin (c + 1 + 1)) k) : (S.T t).Phi j (c + 1) (by omega) (z + colAt (XIx (Fin κ) t j) (c + 1) (r + 1) y) = (S.T t).Phi j (c + 1) (by omega) z

theorem **mem_box_of_Phi_ne_zero** (hpat : S.Pattern t) (j c : ℕ) (H : j + 1 + c = s) (r : ℕ) (hr : c + 1 = 2 ∗ r) (z : Matrix (XIx (Fin κ) t (j + 1) ⊕ Fin (c + 1)) (XIx (Fin κ) t j ⊕ Fin (c + 1 + 1)) k) (hz : ∀ a, z (.inl a) (.inr ⟨r, by omega⟩) = 0) (y : XIx (Fin κ) t (j + 1) → k) (h : (S.T t).Phi j (c + 1) (by omega) (z + colAt (XIx (Fin κ) t j) (c + 1) (r + 1) y) ≠ 0) : y ∈ box (XIx (Fin κ) t (j + 1)) (S.ϖ ^ r)⁻¹

theorem **_root_.GlobalTheory.LocFlat.pureCol_of_pattern** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], S.Pattern t → S.ExchangeAt t ν → GlobalTheory.LocFlat.PureCol S t ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatColumnIndicatorPureAtSymmetricPositionStatement

theorem **_root_.GlobalTheory.T.flatColumnIndicator_pure_at_symmetricPosition** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], S.Pattern t → S.ExchangeAt t ν → GlobalTheory.LocFlat.PureCol S t ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatColumnMapEqRootExchangeOperatorTranslate

theorem **colE_neg_inv** {k : Type} [Field k] {n : Type} [Fintype n] [DecidableEq n] (c p : ℕ) (y : n → k) : ((colE c p (-y))⁻¹ : GL (n ⊕ Fin c) k).val = (colE c p y).val

theorem **affPt_add_colAt** {k : Type} [Field k] {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (q : Fin c) (y : n → k) (x₀ : Matrix n n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free (q.val + 1) rs} → k) : affPt (q.val + 1) x₀ e + colAt n' c (q.val + 1) y = mat x₀ (ext0 (Free (q.val + 1)) e + gCol q y)

theorem **ch_add_colAt** {k : Type} [Field k] {n n' : Type} {c : ℕ} (p : ℕ) (y : n → k) (z : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : ch (z + colAt n' c p y) = ch z

theorem **flCol_eq_opE_colE** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), ∀ x : Matrix n (n' ⊕ Unit) k, GlobalTheory.LocFlat.flCol ψ ν p Φ x = opE ψ ν p (fun z => Φ ((colE c p fun r => x r (Sum.inr ())).val ∗ z)) x.toCols₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatColumnMapEqRootExchangeOperatorTranslateStatement

theorem **_root_.GlobalTheory.T.flatColumnMap_eq_rootExchangeOperator_translate** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), ∀ x : Matrix n (n' ⊕ Unit) k, GlobalTheory.LocFlat.flCol ψ ν p Φ x = opE ψ ν p (fun z => Φ ((colE c p fun r => x r (Sum.inr ())).val ∗ z)) x.toCols₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatColumnSupportOfOneIndexAtSymPosition

theorem **det_colE** {k : Type} [Field k] {n : Type} [Fintype n] [DecidableEq n] (c p : ℕ) (β : n → k) : Matrix.GeneralLinearGroup.det (colE c p β) = 1

theorem **om_colE** {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (μ : kˣ →∗ ℂˣ) (g h : GL (n ⊕ Fin c) k) (hg : Matrix.GeneralLinearGroup.det g = 1) (Φ : Matrix (n ⊕ Fin c) n' k → ℂ) : (fun z => om μ 1 h Φ (g.val ∗ z)) = om μ 1 (g⁻¹ ∗ h) Φ

theorem **GlobalTheory.LocFlat.suppCol_of_oneIndexAt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], S.OneIndexAt t ν → GlobalTheory.LocFlat.SuppCol S t ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatColumnSupportOfOneIndexAtSymPositionStatement

theorem **_root_.GlobalTheory.T.flatColumnSupport_of_oneIndexAt_symPosition** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], S.OneIndexAt t ν → GlobalTheory.LocFlat.SuppCol S t ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatContrIntIndicator

theorem **coe_psi_sum** {ι : Type} (ψ : AddChar k Circle) (s : Finset ι) (f : ι → k) : (ψ (∑ i ∈ s, f i) : ℂ) = ∏ i ∈ s, (ψ (f i) : ℂ)

theorem **intM_add_of_disj** {m m' : Type} {A B : Matrix m m' k} (h : ∀ r s, B r s = 0 ∨ A r s = 0) : IntM (A + B) ↔ IntM A ∧ IntM B

theorem **intM_colAt** (y : n → k) : IntM (colAt (k

theorem **intM_rowAt** (u : n' → k) : IntM (rowAt (k

theorem **colAt_disj** (x₀ : Matrix n n' k) (e) (y : n → k) (r : n ⊕ Fin c) (s : n' ⊕ Fin (c + 1)) : colAt n' c p y r s = 0 ∨ affPt (c

theorem **rowAt_disj** (x₀ : Matrix n n' k) (e) (u : n' → k) (r : n ⊕ Fin c) (s : n' ⊕ Fin (c + 1)) : rowAt n c p u r s = 0 ∨ affPt (c

theorem **flCol_intInd** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (p : ℕ) : flCol ψ ν p (Set.indicator {x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) = (if p = 0 ∨ c + 1 < p then fun x => Set.indicator {x : Matrix n n' k | ∀ r q, x r q ∈ 𝒪[k]} (fun _ => 1) x.toCols₁ else Set.indicator {x | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1)

theorem **flRow_intInd** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (p : ℕ) (hp1 : 1 ≤ p) (hpc : p ≤ c) : flRow ψ ν p (Set.indicator {x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) = Set.indicator {x | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1

theorem **integral_psi_O** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (a : k) : ∫ t, (if t ∈ 𝒪[k] then (1 : ℂ) else 0) ∗ (ψ (-1 ∗ (a ∗ t)) : ℂ) ∂ν = if a ∈ 𝒪[k] then 1 else 0

theorem **integral_psi_box** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (u : n → k) : ∫ y, (if ∀ r, y r ∈ 𝒪[k] then (1 : ℂ) else 0) ∗ (ψ (-1 ∗ (u ⬝ᵥ y)) : ℂ) ∂(Measure.pi fun _ : n => ν) = if ∀ r, u r ∈ 𝒪[k] then 1 else 0

theorem **ind_eq_ite** {m m' : Type} (x : Matrix m m' k) : Set.indicator {x : Matrix m m' k | ∀ r q, x r q ∈ 𝒪[k]} (fun _ => (1 : ℂ)) x = if IntM x then 1 else 0

theorem **contr_intInd** {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) (q : Matrix n n' k × Matrix n'' n k) : contr ψ ν (Set.indicator {x : Matrix n (n' ⊕ Unit) k | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) (Set.indicator {x : Matrix (n'' ⊕ Unit) n k | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) q = Set.indicator {x : Matrix n n' k | ∀ r q, x r q ∈ 𝒪[k]} (fun _ => 1) q.1 ∗ Set.indicator {x : Matrix n'' n k | ∀ r q, x r q ∈ 𝒪[k]} (fun _ => 1) q.2

theorem **GlobalTheory.LocFlat.flat_contr_intIndicator** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), HasConductor ψ 1 → IsUnitVol ν → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (c p : ℕ), GlobalTheory.LocFlat.flCol ψ ν p (Set.indicator {x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) = (if p = 0 ∨ c + 1 < p then fun x => Set.indicator {x : Matrix n n' k | ∀ r q, x r q ∈ 𝒪[k]} (fun _ => 1) x.toCols₁ else Set.indicator {x | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) ∧ (1 ≤ p → p ≤ c → GlobalTheory.LocFlat.flRow ψ ν p (Set.indicator {x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) = Set.indicator {x | ∀ r q, x r q ∈ 𝒪[k]} fun _ => 1) ∧ ∀ q, GlobalTheory.L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFactorReductionKernelAllNonFreeUnreadEntries

private theorem **mem_evC_iff** (e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))) : e ∈ _root_.GlobalTheory.FlatFactor.evC n n' c ρ ↔ (∃ i : Fin (c + 1), e.2 = Sum.inr i ∧ (i : ℕ) < ρ ∧ ¬_root_.GlobalTheory.RootEx.Free (ρ + 1) e) ∨ ∃ l : Fin c, e.1 = Sum.inr l ∧ ρ ≤ (l : ℕ) ∧ ∀ i : Fin (c + 1), e.2 = Sum.inr i → (i : ℕ) = ρ

private theorem **mem_evAll_iff** (e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))) : e ∈ _root_.GlobalTheory.FlatFactor.evAll n n' c ρ ↔ ¬_root_.GlobalTheory.RootEx.Free (ρ + 1) e ∧ ¬_root_.GlobalTheory.FlatFactor.IsRead n n' c ρ e

private theorem **evC_subset_evAll** : _root_.GlobalTheory.FlatFactor.evC n n' c ρ ⊆ _root_.GlobalTheory.FlatFactor.evAll n n' c ρ

private theorem **mem_itB_iff** (e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))) : e ∈ _root_.GlobalTheory.FlatFactor.itB n n' c ρ ↔ _root_.GlobalTheory.RootEx.Free (ρ + 1) e ∧ ∀ i : Fin (c + 1), e.2 = Sum.inr i → ρ ≤ (i : ℕ)

private theorem **disjoint_evAll_itB** : Disjoint (_root_.GlobalTheory.FlatFactor.evAll n n' c ρ) (_root_.GlobalTheory.FlatFactor.itB n n' c ρ)

private theorem **exists_inr_lt_le_of_mem_evAll_notMem_evC** {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} (he : e ∈ _root_.GlobalTheory.FlatFactor.evAll n n' c ρ) (hC : e ∉ _root_.GlobalTheory.FlatFactor.evC n n' c ρ) : ∃ (L : Fin c) (I : Fin (c + 1)), e = (Sum.inr L, Sum.inr I) ∧ ρ < (I : ℕ) ∧ (I : ℕ) ≤ L

private theorem **embA_tv_eq_one_add_single** (i j : Fin c) (h : i ≠ j) (a : k) : ((_root_.GlobalTheory.RootEx.embA n (_root_.GlobalTheory.FlatFactor.LowStage.tv i j h a) : GL (n ⊕ Fin c) k) : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) = 1 + single (Sum.inr i) (Sum.inr j) a

private theorem **comp_left_mul_mem_SB** {m m' : Type} [Fintype m] [DecidableEq m] [Fintype m'] (A : GL m k) {θ : Matrix m m' k → ℂ} (hθ : θ ∈ _root_.GlobalTheory.SplitWeil.SB (Matrix m m' k)) : (fun t => θ ((A : Matrix m m k) * t)) ∈ _root_.GlobalTheory.SplitWeil.SB (Matrix m m' k)

private theorem **killsKernel_rowShear_invariant_lemma_10** (ν : Measure k) [ν.IsAddHaarMeasure] {M : Type} [AddCommGroup M] [Module ℂ M] (l : _root_.GlobalTheory.SplitWeil.SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M) (ev : Finset ((n ⊕ Fin c) × (n' ⊕ Fin (c + 1)))) (i₀ j : Fin c) (hρ : ρ ≤ (i₀ : ℕ)) (hij : (i₀ : ℕ) < j) (hcov : ∀ e ∈ _root_.GlobalTheory.FlatFactor.evAll n n' c ρ, (∀ q : Fin (c + 1), e.2 = Sum.inr q → (q : ℕ) ≤ i₀) → e ∈ ev) (hker : _root_.GlobalTheory.FlatFactor.KerRed ν ev (_root_.GlobalTheory.FlatFactor.itB n n' c ρ) (_root_.GlobalTheory.RootEx.basePt c (0 : Matrix n n' k)) l) (s : k) (θ θ' : _root_.GlobalTheory.SplitWeil.SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (hθ' : ∀ t, (θ' : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) t = (θ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) ((1 + single (Sum.inr i₀) (Sum.inr j) s : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) * t))

private theorem **twist_invariant_of_lowerEntry_lemma_4** (ψ : AddChar k Circle) (ν : Measure k) [ν.IsAddHaarMeasure] {M : Type} [AddCommGroup M] [Module ℂ M] (l : _root_.GlobalTheory.SplitWeil.SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M) (hTw : _root_.GlobalTheory.FlatFactor.TwLaw ρ ψ l) (ev : Finset ((n ⊕ Fin c) × (n' ⊕ Fin (c + 1)))) (L : Fin c) (I : Fin (c + 1)) (hρI : ρ < (I : ℕ)) (hIL : (I : ℕ) ≤ L) (hcov : ∀ e ∈ _root_.GlobalTheory.FlatFactor.evAll n n' c ρ, (∀ q : Fin (c + 1), e.2 = Sum.inr q → (q : ℕ) < I) → e ∈ ev) (hker : _root_.GlobalTheory.FlatFactor.KerRed ν ev (_root_.GlobalTheory.FlatFactor.itB n n' c ρ) (_root_.GlobalTheory.RootEx.basePt c (0 : Matrix n n' k)) l) (b : k) (θ θ' : _root_.GlobalTheory.SplitWeil.SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (hθ' : ∀ t, (θ' : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) t = (ψ (b * (t (Sum.inr L) (Sum.inr I) − _root_.Glo

theorem **_root_.GlobalTheory.FlatFactor.LowStage.kerRed_evAll** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (l : _root_.GlobalTheory.SplitWeil.SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ → _root_.GlobalTheory.FlatFactor.TwLaw ρ ψ l → _root_.GlobalTheory.FlatFactor.KerRed ν (_root_.GlobalTheory.FlatFactor.evC n n' c ρ) (_root_.GlobalTheory.FlatFactor.itB n n' c ρ) (_root_.GlobalTheory.RootEx.basePt c (0 : Matrix n n' k)) l → _root_.GlobalTheory.FlatFactor.KerRed ν (_root_.GlobalTheory.FlatFactor.evAll n n' c ρ) (_root_.GlobalTheory.FlatFactor.itB n n' c ρ) (_root_.GlobalTheory.RootE

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFactorReductionKernelAllNonFreeUnreadEntriesStatement

theorem **_root_.GlobalTheory.T.flatFactorReduction_kernel_all_nonFreeUnread_entries** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ → GlobalTheory.FlatFactor.TwLaw ρ ψ l → GlobalTheory.FlatFactor.KerRed ν (GlobalTheory.FlatFactor.evC n n' c ρ) (GlobalTheory.FlatFactor.itB n n' c ρ) (basePt c (0 : Matrix n n' k)) l → GlobalTheory.FlatFactor.KerRed ν (GlobalTheory.FlatFactor.evAll n n' c ρ) (GlobalTheory.FlatFactor.itB n n' c ρ) (basePt c (0 : Matrix n n' k)) l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFactorReductionKernelIntegratesFreeEntries

theorem **measurePreserving_shear** {k ι : Type} [AddCommGroup k] [MeasurableSpace k] [MeasurableAdd₂ k] [Fintype ι] (ν : Measure k) [SigmaFinite ν] [ν.IsAddRightInvariant] (p : ι → Prop) [DecidablePred p] (v : (ι → k) → ι → k) (hv : Measurable v) (h0 : ∀ α i, p i → v α i = 0) (hd : ∀ α α', (∀ i, p i → α i = α' i) → v α = v α') : MeasurePreserving (fun α => α + v α) (Measure.pi fun _ : ι => ν) (Measure.pi fun _ : ι => ν)

theorem **red_rowAdd** (ν : Measure k) [ν.IsAddHaarMeasure] (ev it : Finset (m × m')) (β t t' : Matrix m m' k) (r₀ r₁ : m) (hr : r₁ ≠ r₀) (a : k) (θ : Matrix m m' k → ℂ) (hθ : Continuous θ) (h : ∀ α r q, (r, q) ∉ it → rowAdd r₀ r₁ a (fill ev it β t α) r q = fill ev it β t' α r q) : red ν ev it β (fun x => θ (rowAdd r₀ r₁ a x)) t = red ν ev it β θ t'

theorem **kerRed_eq** {M : Type} [AddCommGroup M] [Module ℂ M] (ν : Measure k) [ν.IsAddHaarMeasure] (ev it : Finset (m × m')) (β : Matrix m m' k) (l : SB (Matrix m m' k) →ₗ[ℂ] M) (hK : KerRed ν ev it β l) (θ θ' : SB (Matrix m m' k)) (h : ∀ t, red ν ev it β 0.1 t = red ν ev it β θ'.1 t) : l θ = l θ'

theorem **fill_add_single_of_notMem** (ev it : Finset (m × m')) (β t : Matrix m m' k) (x : {e : m × m' // e ∈ it} → k) (r₀ : m) (q₀ : m') (h1 : (r₀, q₀) ∉ ev) (h2 : (r₀, q₀) ∉ it) (a : k) : fill ev it β t x + Matrix.single r₀ q₀ a = fill ev it β (t + Matrix.single r₀ q₀ a) x

theorem **mem_SB_mul_left** (g : GL m k) (θ : SB (Matrix m m' k)) : (fun t => 0.1 (g.val * t)) ∈ SB (Matrix m m' k)

theorem **itUp_top** : itUp n n' c ρ ρ = itB n n' c ρ

theorem **itUp_zero** : itUp n n' c ρ 0 = itFree n n' c ρ

theorem **mem_itUp** {j : ℕ} {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} : e ∈ itUp n n' c ρ j ↔ Free (ρ + 1) e ∧ ∀ i, e.2 = Sum.inr i → ρ ≤ i.val ∨ j ≤ i.val

theorem **mem_evAll** {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} : e ∈ evAll n n' c ρ ↔ ¬ Free (ρ + 1) e ∧ ¬ IsRead n n' c ρ e

theorem **fromBlocks_transvection_mul** {m' : Type} (i j : Fin c) (a : k) (x : Matrix (n ⊕ Fin c) m' k) : (fromBlocks 1 0 0 (transvection i j a) : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k) * x = rowAdd (Sum.inr i) (Sum.inr j) a x

theorem **embA_tv_mul** {m' : Type} (i j : Fin c) (h : i ≠ j) (a : k) (x : Matrix (n ⊕ Fin c) m' k) : (embA n (tv i j h a)).val * x = rowAdd (Sum.inr i) (Sum.inr j) a x

theorem **embA_tv_inv_mul** {m' : Type} (i j : Fin c) (h : i ≠ j) (a : k) (x : Matrix (n ⊕ Fin c) m' k) : ((embA n (tv i j h a))⁻¹).val * x = rowAdd (Sum.inr i) (Sum.inr j) (-a) x

theorem **ch_rowAdd** (i j : Fin c) (a : k) (x : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : ch (rowAdd (Sum.inr i) (Sum.inr j) a x) - ch x = a * x (Sum.inr j) (Sum.inr i.succ)

theorem **simpleSum_tv** (i j : Fin c) (h : i ≠ j) (a : k) : simpleSum (embA n (tv i j h a)) = if j.val = i.val + 1 then a else 0

theorem **embA_tv_mem** (i j : Fin c) (h : i ≠ j) (hij : i.val < j.val) (a : k) (p : ℕ) : embA n (tv i j h a) ∈ N k n c p

theorem **rowAdd_fill** (l j : Fin c) (hl : l.val < j.val) (hj : j.val < ρ) (a : k) (t : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (α) (r) (q) (h : (r, q) ∉ itUp n n' c ρ (j.val + 1)) : rowAdd (Sum.inr l) (Sum.inr j) a (fill (evAll n n' c ρ) (itUp n n' c ρ (j.val + 1)) (basePt c (0 : Matrix n n' k)) t α) r q = fill (evAll n n' c ρ) (itUp n n' c ρ (j.val + 1)) (basePt c (0 : Matrix n n' k)) (t + Matrix.single (Sum.inr l) (Sum.inr j.castSucc) a) α r q

private theorem **addChar_comp_mul_mem_SB** {X : Type} [TopologicalSpace X] (ψ : AddChar k Circle) (hψ : Continuous ψ) (f : X → k) (hf : Continuous f) (θ : SB X) : (fun t => (ψ (f t) : ℂ) * θ.1 t) ∈ SB X

theorem **translate_law** (ψ : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψ) [ν.IsAddHaarMeasure] {M : Type} [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M) (hT : TwLaw ρ ψ L) (l j : Fin c) (hl : l.val < j.val) (hj : j.val < ρ) (hK : KerRed ν (evAll n n' c ρ) (itUp n n' c ρ (j.val + 1)) (basePt c (0 : Matrix n n' k)) L) (a : k) (θ θ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (h : ∀ t, θ'.1 t = θ.1 (t + Matrix.single (Sum.inr l) (Sum.inr j.castSucc) a)) : L θ' = L θ

theorem **kerRed_itFree** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ → GlobalTheory.FlatFactor.TwLaw ρ ψ l → GlobalTheory.FlatFactor.KerRed ν (GlobalTheory.FlatFactor.evAll n n' c ρ) (GlobalTheory.FlatFactor.itB n n' c ρ) (basePt c (0 : Matrix n n' k)) l → GlobalTheory.FlatFactor.KerRed ν (GlobalTheory.FlatFactor.evAll n n' c ρ) (GlobalTheory.FlatFactor.itFree n n' c ρ) (basePt c (0 : Matrix n n' k)) l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFactorReductionKernelIntegratesFreeEntriesStatement

theorem **_root_.GlobalTheory.T.flatFactorReduction_kernel_integrates_free_entries** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ → GlobalTheory.FlatFactor.TwLaw ρ ψ l → GlobalTheory.FlatFactor.KerRed ν (GlobalTheory.FlatFactor.evAll n n' c ρ) (GlobalTheory.FlatFactor.itB n n' c ρ) (basePt c (0 : Matrix n n' k)) l → GlobalTheory.FlatFactor.KerRed ν (GlobalTheory.FlatFactor.evAll n n' c ρ) (GlobalTheory.FlatFactor.itFree n n' c ρ) (basePt c (0 : Matrix n n' k)) l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFamilyZetaConvergenceHolomorphy

theorem **flatIntegrand_rebase** (f : X → ℂ) {a : X → ℝ} (ha : ∀ x, 0 < a x) (s₀ s₁ s : ℂ) (x : X) : flatIntegrand f a s₀ s x = flatIntegrand f a s₀ s₁ x * ((a x : ℝ) : ℂ) ^ (s - s₁)

theorem **norm_flatIntegrand_le** (f : X → ℂ) {a : X → ℝ} (ha : ∀ x, 0 < a x) (ha1 : ∀ x, a x ≤ 1) (s₀ : ℂ) {s₁ s : ℂ} (hs : s₁.re ≤ s.re) (x : X) : ‖flatIntegrand f a s₀ s x‖ ≤ ‖flatIntegrand f a s₀ s₁ x‖

theorem **aestronglyMeasurable_flatIntegrand** {μ : Measure X} {f : X → ℂ} {a : X → ℝ} (hm : Measurable a) (ha : ∀ x, 0 < a x) {s₀ s₁ : ℂ} (h : AEStronglyMeasurable (flatIntegrand f a s₀ s₁) μ) (s : ℂ) : AEStronglyMeasurable (flatIntegrand f a s₀ s) μ

theorem **hasDerivAt_flatIntegrand** (f : X → ℂ) {a : X → ℝ} (ha : ∀ x, 0 < a x) (s₀ s : ℂ) (x : X) : HasDerivAt (fun z => flatIntegrand f a s₀ z x) (flatIntegrand f a s₀ s x * ((Real.log (a x) : ℝ) : ℂ)) s

theorem **flatZetaHolomorphic** : FlatZetaHolomorphic

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFamilyZetaConvergenceHolomorphyStatement

theorem **_root_.GlobalTheory.T.flatFamilyZeta_convergence_holomorphy** : GlobalTheory.IndefiniteRowSideFullHecke.FlatZetaIntegral.FlatZetaHolomorphic

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFourierJacobiToBesselColumnMulLaws

theorem **frame_col** (p : Place D k) {σ : ℕ} (B : BesLev D σ) (s : k) {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} (FA : Frame k B.A.rank n) (FV : Frame k B.V.rank (n ⊕ Fin c)) (hB : p.IsBesFrame B (s * p.xi) FA FV) (j : Fin (σ + 1)) (l : Fin c) (hl : l.val = j.val) (x : Fin B.V.rank) : FV.β x (Sum.inr l) = s * p.xi ^ (σ + 1 - j.val) * ((xLast B.d (vecB B.o B.ζ)).map p.π) x j

theorem **flatFB_mulcol** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ i, q = Sum.inr i ∧ i.val ≤ σ ∧ ¬ Free (σ + 1) (r, q)) → ∀ (b : k) (Φ Φ' : SB (Ma

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatFourierJacobiToBesselColumnMulLawsStatement

theorem **_root_.GlobalTheory.T.flat_fourierJacobi_to_bessel_column_mul_laws** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ i, q = Sum.inr i ∧ i.val ≤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatIntegrandLocalFactorSplitsTwistPlace

theorem **transport** {G H : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] [Group H] [TopologicalSpace H] [IsTopologicalGroup H] [MeasurableSpace H] [BorelSpace H] (e : G ≃ₜ* H) (m : Measure G) [m.IsHaarMeasure] (P :

Subgroup G) (hP : IsOpen (P : Set G)) (h1 : m P = 1) (f : G → ℂ) (hf : Integrable f m) : ∃ (J : Subgroup H) (μ : Measure H), IsOpen (J : Set H) ∧ μ.IsHaarMeasure ∧ μ J = 1 ∧ Integrable (fun y => f (e.symm y)) μ ∧ (∀ y, y ∈ J ↔ e.symm y ∈ P) ∧ ∫ y, f (e.symm y) ∂μ = ∫ g, f g ∂m

theorem **exists_haar** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] [LocallyCompactSpace G] (P : Subgroup G) (hP : IsOpen (P : Set G)) (hc : IsCompact (P : Set G)) : ∃ m : Measure G, m.IsHaarMeasure ∧ m P = 1

theorem **rcoef_eq_coef** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {I I' : Type} [Fintype I] [DecidableEq I] [Fintype I'] [DecidableEq I'] {A : HSp D} (m : Measure (GA D A)) (hm : m.IsHaarMeasure) {u₁ u₂ : GA D A → ℂ} (h1 : MemL2Quotient (GQ A) m u₁) (h2 : MemL2Quotient (GQ A) m u₂) (ι : GL (I ⊕ I') k →* GA D A) (x : GA D A) (hx : ∀ y, ι y * x = x * ι y) (ξ : kˣ →* ℂˣ) (hξ : GlobalTheory.SmGL.IsUnitaryChar ξ) (ϖ : kˣ) (B : Multiset ℂˣ) (hg : IsGen ((GlobalTheory.PRep.rightReg ℂ (GA D A) ℂ).comp ι) ξ ϖ B u₁) (ht : ∀ j ∈ parahoric k I I', ∀ d : kˣ, (d : k) = j.1.toBlocks₂₂.det → ∀ g, u₂ (g * ι j) = (ξ d : ℂ) * u₂ g) : ∃ (V : Type) (_ : AddCommGroup V) (_ : Module ℂ V) (π : Representation ℂ (GL (I ⊕ I') k) V) (u : V) (ℓ : V →ₗ[ℂ] ℂ), IsGen π ξ ϖ B u ∧ IsCoType π ξ ℓ ∧ (∀ y, ℓ (π y u) = rcoef m u₁ u₂ (x * ι y)) ∧ ℓ u = rcoef m u₁ u₂ x

theorem **val_type** {c : Com D} {s : PiLocalization₂ D} (u : s.CC c) (v : SmT s.M) {κ : GA D s.V} (hκ : κ ∈ Lev.levG s.V s.β s.L) (g : GA D s.V) : PiLocalization₂.val u v (g * κ) = Lev.levChi s.V s.β s.L κ * PiLocalization₂.val u v g

theorem **hgt_split** {A : HSp D} {L : GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.Lat A} {a : GL (Fin A.rank) (AE D)} (hL : IsSelfDualC A L) (ha : GlobalTheory.Doubling.IsMaj A a) (Sf : Finset (HeightOneSpectrum (𝒪 D.host.F))) {v : HeightOneSpectrum (𝒪 D.host.F)} (hv : v ∈ Sf) (h1 : GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV A L v 1 = 1) {x : GA D A} (hx : x ∈ openOff A L Sf) (hxv : proj D.sp.cm.sig A.gram v x = 1) (y : GlobalTheory.FinZeta.Gv A v) (t : ℂ) : GlobalTheory.Doubling.npow (hgtL A L a (x * emb D.sp.cm.sig A.gram v y)) t = GlobalTheory.Doubling.npow (hgtL A L a x) t * ((GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.hgtV A L v y : ℝ) : ℂ) ^ t

theorem **hgt_one** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : Type} [Fintype n] [DecidableEq n] {ϖ : kˣ} {H : GL n k → ℝ} (hH : IsHgt ϖ H) {g : GL n k} (hg : g ∈ GlobalTheory.SplitWeil.KO k n) : H g = 1

theorem **local_factor** : GlobalTheory.Published.UnramifiedDoublingGL → GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, ForStepQ D fun c w k s s' p => ∀ m X, IsCovolOne (ratL s'.V s.V) m X → ∀ a, IsMajOf c s a → ∀ (u₁ u₂ : s.CC c) φ, IsQ w k u₁ → ∀ v₁ ∈ s.KF, ∀ v₂ ∈ s.KF, ∀ z : ℝ, (D.R.t : ℝ) / 2 - 1 < z → ∀ PP ∈ w.Q, ∃ (J : Subgroup (GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)))) (μ : Measure (GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)))) (ζ : GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)) → ℂ), IsOpen (J : Set (GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)))) ∧ μ.IsHaarMeasure ∧ μ J = 1 ∧ Integrable ζ μ ∧ (∀ y ∈ J, ζ y = 1) ∧ (∀ g, gAt s.V (.inl (PP.under (𝒪 D.host.F))) g ∈ Lev.levG s.V s.β s.L ↔ GlobalTheory.AdelicU.proj D.sp.cm.sig s.V.gram (PP.under (𝒪 D.host.F)) g ∈ J) ∧ (z = (D.R.t : ℝ) / 2 → ∫ y, ζ y ∂μ = blockL c k PP * (zinv PP D.R.t ((k + 1) * D.R.t) :

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatIntegrandLocalFactorSplitsTwistPlaceStatement

theorem **_root_.GlobalTheory.T.flatIntegrand_localFactor_splits_twistPlace** : GlobalTheory.Published.UnramifiedDoublingGL → GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, ForStepQ D fun c w k s s' p => ∀ m X, IsCovolOne (ratL s'.V s.V) m X → ∀ a, IsMajOf c s a → ∀ (u₁ u₂ : s.CC c) φ, IsQ w k u₁ → ∀ v₁ ∈ s.KF, ∀ v₂ ∈ s.KF, ∀ z : ℝ, (D.R.t : ℝ) / 2 - 1 < z → ∀ PP ∈ w.Q, ∃ (J : Subgroup (GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)))) (μ : Measure (GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)))) (ζ : GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)) → ℂ), IsOpen (J : Set (GlobalTheory.FinZeta.Gv s.V (PP.under (𝒪 D.host.F)))) ∧ μ.IsHaarMeasure ∧ μ J = 1 ∧ Integrable ζ μ ∧ (∀ y ∈ J, ζ y = 1) ∧ (∀ g, gAt s.V (.inl (PP.under (𝒪 D.host.F))) g ∈ Lev.levG s.V s.β s.L ↔ GlobalTheory.AdelicU.proj D.sp.cm.sig s.V.gram (PP.under (𝒪 D.host.F)) g ∈ J) ∧ (z = (D.R.t : ℝ) / 2 → ∫ y, ζ y ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapBesselToFourierJacobiLocalModel

theorem **flatBF_of_laws** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → FlatBF D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapBesselToFourierJacobiLocalModelStatement

theorem **_root_.GlobalTheory.T.flatMap_besselToFourierJacobi_localModel** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → FlatBF D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapFourierJacobiFactorsLocalFlat

theorem **ch_add_mat** {k : Type} [Field k] {n n' : Type} {c : ℕ} (a b : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : ch (a + b) = ch a + ch b

theorem **continuous_ch** {k : Type} [Field k] [TopologicalSpace k] [IsTopologicalRing k] {n n' : Type} {c : ℕ} : Continuous (ch : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → k)

theorem **kerRed_empty** {M : Type} [AddCommGroup M] [Module ℂ M] (ν : Measure k) (β : Matrix m m' k) (l : SB (Matrix m m' k) →ₗ[ℂ] M) : KerRed ν ∅ ∅ β l

theorem **mem_SB_addChar_mul** (ψ : AddChar k Circle) (hψ : Continuous ψ) {X : Type} [TopologicalSpace X] (f : X → k) (hf : Continuous f) (θ : X → ℂ) (hθ : θ ∈ SB X) : (fun t => (ψ (f t) : ℂ) * θ t) ∈ SB X

theorem **GlobalTheory.FlatFactor.flCol_factor** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ i, q = Sum.inr i ∧ i.val < ρ ∧ ¬ Free (ρ + 1) (r, q)) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = (ψ (b * (t r q - basePt c (0 : Matrix n n' k) r q)) : ℂ) * Φ.1 t) → L Φ' = L Φ) → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ l, r = Sum.inr l ∧ ρ ≤ l.val ∧ ∀ i, q = Sum.inr i → i.val = ρ) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapFourierJacobiFactorsLocalFlatStatement

theorem **_root_.GlobalTheory.T.flatMap_fourierJacobi_factors_local_flat** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ i, q = Sum.inr i ∧ i.val < ρ ∧ ¬ Free (ρ + 1) (r, q)) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = (ψ (b * (t r q - basePt c (0 : Matrix n n' k) r q)) : ℂ) * Φ.1 t) → L Φ' = L Φ) → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ l, r = Sum.inr l ∧ ρ ≤ l.val ∧ ∀ i, q = Sum.inr i → i.val = ρ) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapFourierJacobiRowAdd

theorem **add_right_mem_SB** {m m' k : Type} [AddCommGroup k] [TopologicalSpace k] [IsTopologicalAddGroup k] {Φ : Matrix m m' k → ℂ} (hΦ : Φ ∈ SB (Matrix m m' k)) (a : Matrix m m' k) : (fun z => Φ (z + a)) ∈ SB (Matrix m m' k)

theorem **flRow_add** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (Φ₁ Φ₂ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ), Φ₁ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) → Φ₂ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) → GlobalTheory.LocFlat.flRow ψ ν p (Φ₁ + Φ₂) = GlobalTheory.LocFlat.flRow ψ ν p Φ₁ + GlobalTheory.LocFlat.flRow ψ ν p Φ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapFourierJacobiRowAddStatement

theorem **_root_.GlobalTheory.T.flatMapFourierJacobiRow_add** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (Φ₁ Φ₂ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ), Φ₁ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) → Φ₂ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) → GlobalTheory.LocFlat.flRow ψ ν p (Φ₁ + Φ₂) = GlobalTheory.LocFlat.flRow ψ ν p Φ₁ + GlobalTheory.LocFlat.flRow ψ ν p Φ₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapFourierJacobiToBesselLocalModel

theorem **flatFB_holds** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → FlatFB D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapFourierJacobiToBesselLocalModelStatement

theorem **_root_.GlobalTheory.T.flatMap_fourierJacobiToBessel_localModel** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → FlatFB D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapHandedFunctionVanishingPeriodInvariant

theorem **addChar_ne_zero** {A : Type} [AddGroup A] (ψ : AddChar A ℂ) (a : A) : ψ a ≠ 0

theorem **mem_congGrp_iff** (V : HSp D) (I : Ideal (AE D)) (g : GA D V) : g ∈ congGrp V I ↔ (mt V g).map (Ideal.Quotient.mk I) = 1

theorem **unitaryMatrix_wittCoreEmbedding** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (h : GA D S₀) : mt S (d.emb0 h) = d.jX · d.pX + d.j0 · mt S₀ h · d.p0 + d.jY · d.pY

theorem **wittCoreEmbedding_preserves_supportedUnitarySubgroup** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (P : Pl D → Prop) {h : GA D S₀} (hh : h ∈ grpAt S₀ P) : d.emb0 h ∈ grpAt S P

theorem **comp_mem_grpAt** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (P : Pl D → Prop) {n : GA D S} (hn : n ∈ flagN d) {h : GA D S₀} (hr : n ∗ d.emb0 h ∈ grpAt S P) : h ∈ grpAt S₀ P

theorem **jac_decomp** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (P : Pl D → Prop) {r : GA D S} (hr : r ∈ jacGrp d) (hP : r ∈ grpAt S P) : ∃ n ∈ flagN d, n ∈ grpAt S P ∧ ∃ h ∈ grpAt S₀ P, r = n ∗ d.emb0 h

theorem **heisO_eq_zero_iff** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w₀ : WO S T) (F : WO S T → ℂ) : heisO ψ S T w₀ F = 0 ↔ F = 0

theorem **om_eq_zero_iff** {S T : HSp D} (om : Om S T) (g : GA D S × GA D T) (F : WO S T → ℂ) : om g F = 0 ↔ F = 0

theorem **orth_comp_mem** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (P : Pl D → Prop) {h : GA D A} (hh : o.emb (h, 1) ∈ grpAt S₀ P) : h ∈ grpAt A P

theorem **bes_decomp** {S S₀ A L : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (o : OrthDec S₀ A L) (P : Pl D → Prop) {r : GA D S} (hr : r ∈ besGrp d o) (hP : r ∈ grpAt S P) : ∃ n ∈ flagN d, n ∈ grpAt S P ∧ ∃ h ∈ grpAt A P, r = n ∗ embA d o h

theorem **opA_eq_zero_iff** (ψ : AddChar (AF D.host.F) ℂ) {W W₀ V₀ A L : HSp D} {ρ : ℕ} (dW : HypDec W W₀ ρ) (o : OrthDec V₀ A L) (ζ : Fin L.rank → D.host.E) (om : Om V₀ W₀) (n : GA D W) (h : GA D W₀) (Ξ : WO V₀ W₀ → ℂ) : opA ψ dW o ζ om n h Ξ = 0 ↔ Ξ = 0

theorem **handFB_smul_om_at** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (Ξ : WO J.W₀ B.V₀ → ℂ) (c : ℂ) (h₁ g : GA D B.A) : handFB s vL Y (c • s.om₀ (1, B.o.emb (h₁, 1)) Ξ) g = c ∗ handFB s vL Y Ξ (g ∗ h₁)

theorem **vanish_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (P : Pl D → Prop), (∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (f : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W f → ∀ r ∈ jacGrp J.d, r ∈ grpAt J.W P → (flatBF s (s.om (1, r) f) = 0 ↔ flatBF s f = 0)) ∧ ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (f : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V f → ∀ r ∈ besGrp B.d B.o, r ∈ grpAt B.V P → (handFB s vL Y (flatFB s (s.om (1, r) f)) = 0 ↔ handFB s vL Y (flatFB s f) = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapHandedFunctionVanishingPeriodInvariantStatement

theorem **_root_.GlobalTheory.T.flatMap_handedFunction_vanishing_period_invariant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (P : Pl D → Prop), (∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (f : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W f → ∀ r ∈ jacGrp J.d, r ∈ grpAt J.W P → (flatBF s (s.om (1, r) f) = 0 ↔ flatBF s f = 0)) ∧ ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (f : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V f → ∀ r ∈ besGrp B.d B.o, r ∈ grpAt B.V P → (handFB s vL Y (flatFB s (s.om (1, r) f)) = 0 ↔ handFB s vL Y (flatFB s f) = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapIntegrableAndBoundedLinearForm

theorem **enorm_psi** (hψ : IsStdChar D ψ (-1)) (a : AF D.host.F) : ‖ψ a‖ₑ = 1

theorem **enorm_evalAt_le** (hψ : IsStdChar D ψ (-1)) (hT : μT 𝓕T = 1) (f : WO V W → ℂ) (x : Mx D W.rank ρ) (M : ℝ≥0∞) (hM : ∀ w', ‖dV.mixS W ψ μS 𝓕S f x w'‖ₑ ≤ M) (y : Mx D V₀.rank σ) (w : WO V₀ W₀) : ‖evalAt ψ dV dW μS 𝓕S μT 𝓕T f x y w‖ₑ ≤ M

theorem **enorm_flatFun_le** (hψ : IsStdChar D ψ (-1)) (hT : μT 𝓕T = 1) (f : WO V W → ℂ) (x : Mx D W.rank ρ) (M : Mx D W.rank ρ → ℝ≥0∞) (hM : ∀ x' w', ‖dV.mixS W ψ μS 𝓕S f x' w'‖ₑ ≤ M x') (y : Mx D V₀.rank σ) (w : WO V₀ W₀) (u : UX D ρ) : ‖flatFun ψ dV dW μS 𝓕S μT 𝓕T x y f w u‖ₑ ≤ M (x · u.1.1)

theorem **integrable_integrandT** (hψ : IsStdChar D ψ (-1)) (hT : μT 𝓕T = 1) (g : WO V₀ W → ℂ) (B : ℝ) (hB : ∀ w', ‖g w'‖ ≤ B) (hg : Continuous g) (y : Mx D V₀.rank σ) (w : WO V₀ W₀) : Integrable (fun ξ => ψ (symplOH V₀ W (dW.incX V₀ ξ) (dW.incAdj V₀ y)) ∗ g (dW.incX V₀ ξ + dW.incAdj V₀ y + dW.inc0 V₀ w)) (μT.restrict 𝓕T)

theorem **continuous_evalAt** (hψ : IsStdChar D ψ (-1)) (hT : μT 𝓕T = 1) (f : WO V W → ℂ) (B : ℝ) (hB : ∀ x w', ‖dV.mixS W ψ μS 𝓕S f x w'‖ ≤ B) (hc : Continuous fun p : Mx D W.rank ρ × WO V₀ W => dV.mixS W ψ μS 𝓕S f p.1 p.2) (y : Mx D V₀.rank σ) (w : WO V₀ W₀) : Continuous fun x => evalAt ψ dV dW μS 𝓕S μT 𝓕T f x y w

theorem **continuous_upSum** : Continuous fun u : UX D ρ => upSum (u.1.1 : Mx D ρ ρ)

theorem **continuous_flatFun** (hψ : IsStdChar D ψ (-1)) (f : WO V W → ℂ) (x : Mx D W.rank ρ) (y : Mx D V₀.rank σ) (w : WO V₀ W₀) (hc : Continuous fun x' => evalAt ψ dV dW μS 𝓕S μT 𝓕T f x' y w) : Continuous (flatFun ψ dV dW μS 𝓕S μT 𝓕T x y f w)

theorem **mixS_add** (hψ : IsStdChar D ψ (-1)) (hS : IsCovolOne (ratMx D W.rank ρ) μS 𝓕S) {f g : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (hg : IsThetaFunO ψ V W g) (x : Mx D W.rank ρ) : dV.mixS W ψ μS 𝓕S (f + g) x = dV.mixS W ψ μS 𝓕S f x + dV.mixS W ψ μS 𝓕S g x

theorem **mixS_smul** (c : ℂ) (f : WO V W → ℂ) (x : Mx D W.rank ρ) : dV.mixS W ψ μS 𝓕S (c • f) x = c • dV.mixS W ψ μS 𝓕S f x

theorem **mixT_add** (g₁ g₂ : WO V₀ W → ℂ) (y : Mx D V₀.rank σ) (w : WO V₀ W₀) (h₁ : Integrable (fun ξ => ψ (symplOH V₀ W (dW.incX V₀ ξ) (dW.incAdj V₀ y)) * g₁ (dW.incX V₀ ξ + dW.incAdj V₀ y + dW.inc0 V₀ w)) (μT.restrict 𝓕T)) (h₂ : Integrable (fun ξ => ψ (symplOH V₀ W (dW.incX V₀ ξ) (dW.incAdj V₀ y)) * g₂ (dW.incX V₀ ξ + dW.incAdj V₀ y + dW.inc0 V₀ w)) (μT.restrict 𝓕T)) : dW.mixT V₀ ψ μT 𝓕T (g₁ + g₂) y w = dW.mixT V₀ ψ μT 𝓕T g₁ y w + dW.mixT V₀ ψ μT 𝓕T g₂ y w

theorem **mixT_smul** (c : ℂ) (g : WO V₀ W → ℂ) (y : Mx D V₀.rank σ) (w : WO V₀ W₀) : dW.mixT V₀ ψ μT 𝓕T (c • g) y w = c * dW.mixT V₀ ψ μT 𝓕T g y w

theorem **mixS_bounded** (hψ : IsStdChar D ψ (-1)) {θ : InfFam D} (hθ : IsInfFam θ) {V V₀ W : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (vS : VolM D W.rank ρ) {f : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) : ∃ B : ℝ, ∀ x w', ‖dV.mixS W ψ vS.μ vS.𝓕 f x w'‖ ≤ B

theorem **lintegral_flatFun_le** (hψ : IsStdChar D ψ (-1)) {θ : InfFam D} (hθ : IsInfFam θ) {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E) (hx : LinearIndependent D.host.E x.transpose) (y : Mx D V₀.rank σ) (U : AddSubgroup (WO V W)) (hU : IsOpen (U : Set (WO V W))) : ∃ (N : ℕ) (C : ℝ≥0), ∀ f ∈ GlobalTheory.Size.thetaLevel ψ V W U, ∀ w, ∫⁻ u, ‖flatFun ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (rat x) y f w u‖ₑ ∂vU.ν ≤ C * supSob ψ θ N f

theorem **flatAt_isSobForm** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∀ y : Mx D V₀.rank σ, (∀ Ψ, IsThetaFunO ψ V W Ψ → ∀ w, Integrable (flatFun ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (rat x) y Ψ w) vU.ν) ∧ ∀ w, IsSobForm ψ θ V W fun Ψ => flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν (rat x) y Ψ w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapIntegrableAndBoundedLinearFormStatement

theorem **_root_.GlobalTheory.T.flatMap_integrable_and_bounded_linearForm** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∀ y : Mx D V₀.rank σ, (∀ Ψ, IsThetaFunO ψ V W Ψ → ∀ w, Integrable (flatFun ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (rat x) y Ψ w) vU.ν) ∧ ∀ w, IsSobForm ψ θ V W fun Ψ => flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν (rat x) y Ψ w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapIntertwinesCoreHeisenbergOperators

theorem **flatAt_heisO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (vU : VolU D ρ) (x : Mx D W.rank ρ) (y : Mx D V₀.rank σ) (Ψ : WO V W → ℂ), IsThetaFunO ψ V W Ψ → (∀ w, evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (heisO ψ V W (dV.ext0 W (dW.inc0 V₀ w)) Ψ) x y = heisO ψ V₀ W₀ w (evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 Ψ x y)) ∧ (∀ w, flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y (heisO ψ V W (dV.ext0 W (dW.inc0 V₀ w)) Ψ) = heisO ψ V₀ W₀ w (flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y Ψ)) ∧ ∀ γ, flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y Ψ (ratW W₀ V₀ γ) = flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y Ψ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapIntertwinesCoreHeisenbergOperatorsStatement

theorem **_root_.GlobalTheory.T.flatMap_intertwines_core_heisenbergOperators** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (vU : VolU D ρ) (x : Mx D W.rank ρ) (y : Mx D V₀.rank σ) (Ψ : WO V W → ℂ), IsThetaFunO ψ V W Ψ → (∀ w, evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (heisO ψ V W (dV.ext0 W (dW.inc0 V₀ w)) Ψ) x y = heisO ψ V₀ W₀ w (evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 Ψ x y)) ∧ (∀ w, flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y (heisO ψ V W (dV.ext0 W (dW.inc0 V₀ w)) Ψ) = heisO ψ V₀ W₀ w (flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y Ψ)) ∧ ∀ γ, flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y Ψ (ratW W₀ V₀ γ) = flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 vU.ν x y Ψ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapLawsFactorFourierJacobiSource

theorem **fillAt_entry** (β : Matrix R Q k) (E I : Finset (R × Q)) (x : Matrix R Q k) (y : I → k) (e : R × Q) (hI : e ∉ I) (hE : e ∉ E) : fillAt β E I x y e.1 e.2 = x e.1 e.2

theorem **red_mul_entry** (ψ : AddChar k Circle) (ν : Measure k) (w : Matrix R Q k →ₗ[k] k) (β : Matrix R Q k) (E I : Finset (R × Q)) (e : R × Q) (hI : e ∉ I) (hE : e ∉ E) (g : k → ℂ) (Φ : Matrix R Q k → ℂ) (x : Matrix R Q k) : red ψ ν w β E I (fun t => g (t e.1 e.2) * Φ t) x = g (x e.1 e.2) * red ψ ν w β E I Φ x

theorem **fillAt_add_single** (β : Matrix R Q k) (E I : Finset (R × Q)) (x : Matrix R Q k) (y : I → k) (e : R × Q) (hI : e ∉ I) (hE : e ∉ E) (a : k) : fillAt β E I x y + Matrix.single e.1 e.2 a = fillAt β E I (x + Matrix.single e.1 e.2 a) y

theorem **circle_neg_mul** (ψ : AddChar k Circle) (z : k) : (ψ (-z) : ℂ) * (ψ z : ℂ) = 1

theorem **evalLaw_direct** (ψ : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψ) (L : SB (Matrix R Q k) →ₗ[ℂ] M) (w : Matrix R Q k →ₗ[k] k) (β : Matrix R Q k) (E I : Finset (R × Q)) (e : R × Q) (hI : e ∉ I) (hE : e ∉ E) (h : ∀ (b : k) (Φ Φ' : SB (Matrix R Q k)), (∀ t, Φ'.1 t = (ψ (b * (t e.1 e.2 - β e.1 e.2)) : ℂ) * Φ.1 t) → L Φ' = L Φ) : EvalLaw L ψ ν w β E I e

theorem **intLaw_direct** (ψ : AddChar k Circle) (ν : Measure k) (L : SB (Matrix R Q k) →ₗ[ℂ] M) (w : Matrix R Q k →ₗ[k] k) (β : Matrix R Q k) (E I : Finset (R × Q)) (e : R × Q) (hI : e ∉ I) (hE : e ∉ E) (h : ∀ (a : k) (Φ Φ' : SB (Matrix R Q k)), (∀ t, Φ'.1 t = Φ.1 (t + Matrix.single e.1 e.2 a)) → L Φ' = (ψ (w (Matrix.single e.1 e.2 a)) : ℂ) • L Φ) : IntLaw L ψ ν w β E I e

theorem **kills_empty** (ψ : AddChar k Circle) (ν : Measure k) (L : SB (Matrix R Q k) →ₗ[ℂ] M) (w : Matrix R Q k →ₗ[k] k) (β : Matrix R Q k) : Kills L ψ ν w β ∅ ∅

theorem **kills_evals** (ψ : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (hν : ν.IsAddHaarMeasure) (L : SB (Matrix R Q k) →ₗ[ℂ] M) (w : Matrix R Q k →ₗ[k] k) (β : Matrix R Q k) (I : Finset (R × Q)) (S : Finset (R × Q)) : ∀ E : Finset (R × Q), Disjoint S I → Kills L ψ ν w β E I → (∀ e ∈ S, ∀ E' : Finset (R × Q), E ⊆ E' → E' ⊆ E ∪ S → e ∉ E' → EvalLaw L ψ ν w β E' I e) → Kills L ψ ν w β (E ∪ S) I

theorem **kills_ints** (ψ : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψ) (hν : ν.IsAddHaarMeasure) (L : SB (Matrix R Q k) →ₗ[ℂ] M) (w : Matrix R Q k →ₗ[k] k) (β : Matrix R Q k) (E : Finset (R × Q)) (S : Finset (R × Q)) : ∀ I : Finset (R × Q), Disjoint S E → Kills L ψ ν w β E I → (∀ e ∈ S, ∀ I' : Finset (R × Q), I ⊆ I' → I' ⊆ I ∪ S → e ∉ I' → IntLaw L ψ ν w β E I' e) → Kills L ψ ν w β E (I ∪ S)

theorem **mem_fixSet** {ρ : ℕ} {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} : e ∈ fixSet ρ ↔ ((∃ i, e.2 = Sum.inr i ∧ i.val ≤ ρ ∧ ¬ Free (ρ + 1) e) ∨ (∃ l s, e.1 = Sum.inr l ∧ ρ < l.val ∧ e.2 = Sum.inl s))

theorem **mem_transSet** {ρ : ℕ} {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} : e ∈ transSet ρ ↔ (Free (ρ + 1) e ∧ ∀ i, e.2 = Sum.inr i → ρ < i.val)

theorem **mem_lowSet** {ρ : ℕ} {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} : e ∈ lowSet ρ ↔ ∃ (l : Fin c) (i : Fin (c + 1)), e = (Sum.inr l, Sum.inr i) ∧ ρ < i.val ∧ i.val ≤ l.val

theorem **mem_upSet** {ρ : ℕ} {e : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1))} : e ∈ upSet ρ ↔ ∃ (l : Fin c) (i : Fin (c + 1)), e = (Sum.inr l, Sum.inr i) ∧ l.val < i.val ∧ i.val ≤ ρ

theorem **disj_trans_fix** (ρ : ℕ) : Disjoint (transSet ρ : Finset ((n ⊕ Fin c) × (n' ⊕ Fin (c + 1)))) (fixSet ρ)

theorem **disj_low_trans** (ρ : ℕ) : Disjoint (lowSet ρ : Finset ((n ⊕ Fin c) × (n' ⊕ Fin (c + 1)))) (transSet ρ)

theorem **disj_up_fixlow** (ρ : ℕ) : Disjoint (upSet ρ : Finset ((n ⊕ Fin c) × (n' ⊕ Fin (c + 1)))) (fixSet ρ ∪ lowSet ρ)

theorem **kills_low** (ψ : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (hν : ν.IsAddHaarMeasure) (ρ : ℕ) (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M) (hc : c = 2 * ρ + 1) (H4 : ∀ u : GL (Fin c) k, embA n u ∈ N k n c (ρ + 1) → ∀ (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = Φ.1 (((embA n u)⁻¹).val * t)) → L Φ' = (ψ (-simpleSum (embA n u)) : ℂ) • L Φ) (K : Kills L ψ ν chL (basePt c (0 : Matrix n n' k)) (fixSet ρ) (transSet ρ)) : ∀ j : ℕ, Kills L ψ ν chL (basePt c (0 : Matrix n n' k)) (fixSet ρ ∪ (lowSet ρ).filter (fun e => colIx e < j)) (transSet ρ)

theorem **kills_up** (ψ : AddChar k Circle) (ν : Measure k) (hψ : Continuous ψ) (hν : ν.IsAddHaarMeasure) (ρ : ℕ) (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M) (hc : c = 2 * ρ + 1) (H4 : ∀ u : GL (Fin c) k, embA n u ∈ N k n c (ρ + 1) → ∀ (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = Φ.1 (((embA n u)⁻¹).val * t)) → L Φ' = (ψ (-simpleSum (embA n u)) : ℂ) • L Φ) (K : Kills L ψ ν chL (basePt c (0 : Matrix n n' k)) (fixSet ρ ∪ lowSet ρ) (transSet ρ)) : ∀ d j : ℕ, j + d = c + 1 → Kills L ψ ν chL (basePt c (0 : Matrix n n' k)) (fixSet ρ ∪ lowSet ρ) (transSet ρ ∪ (upSet ρ).filter (fun e => j ≤ colIx e))

theorem **flRow_factor** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ + 1 → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (q : n' ⊕ Fin (c + 1)), (∃ i, q = Sum.inr i ∧ i.val ≤ ρ ∧ ¬ Free (ρ + 1) (r, q)) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = (ψ (b * (t r q - basePt c (0 : Matrix n n' k) r q)) : ℂ) * Φ.1 t) → L Φ' = L Φ) → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ l s, r = Sum.inr l ∧ ρ < l.val ∧ q = Sum.inl s) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = (ψ (b * (t r q - basePt c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapLawsFactorFourierJacobiSourceStatement

theorem **_root_.GlobalTheory.T.flatMap_laws_factor_fourierJacobi_source** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (L : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] M), c = 2 * ρ + 1 → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ i, q = Sum.inr i ∧ i.val ≤ ρ ∧ ¬ Free (ρ + 1) (r, q)) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = (ψ (b * (t r q - basePt c (0 : Matrix n n' k) r q)) : ℂ) * Φ.1 t) → L Φ' = L Φ) → (∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ l s, r = Sum.inr l ∧ ρ < l.val ∧ q = Sum.inl s) → ∀ (b : k) (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c +

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapPreservesVacuumProof

theorem **mulVec_mem_ideal** {m n : ℕ} (I : Ideal (AE D)) (M : Matrix (Fin m) (Fin n) (AE D)) {y : Fin n → AE D} (hy : ∀ i, y i ∈ I) (i : Fin m) : (M *ᵥ y) i ∈ I

theorem **mulVec_latV_of_local** {S S' : HSp D} {L : Lat S} {L' : Lat S'} (hL : Submodule.IsLattice D.host.E L) (hL' : Submodule.IsLattice D.host.E L') {P : Pl D → Prop} (M : Matrix (Fin S'.rank) (Fin S.rank) (AE D)) (h : ∀ v, P v → ∀ y ∈ latV S L (· = v), M *ᵥ y ∈ latV S' L' (· = v)) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : M *ᵥ y ∈ latV S' L' P

theorem **smul_mem_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : ((1 : AF D.host.F) ⊗ₜ[D.host.F] a : AE D) • y ∈ latA S L P

theorem **mulVec_latA** {S S' : HSp D} {L : Lat S} {L' : Lat S'} {P : Pl D → Prop} (M : Matrix (Fin S'.rank) (Fin S.rank) (AE D)) (h : ∀ y ∈ latV S L P, M *ᵥ y ∈ latV S' L' P) {y : Fin S.rank → AE D} (hy : y ∈ latA S L P) : M *ᵥ y ∈ latA S' L' P

theorem **p0_mem** {v : Pl D} (h : MonoAt d L L₀ v) {y : Fin S.rank → AE D} (hy : y ∈ latV S L (· = v)) : d.p0 *ᵥ y ∈ latV S₀ L₀ (· = v)

theorem **wittCoreInclusion_mulVec_mem_latticeCompletionAtPlace** {v : Pl D} (h : MonoAt d L L₀ v) {y : Fin S₀.rank → AE D} (hy : y ∈ latV S₀ L₀ (· = v)) : d.j0 *ᵥ y ∈ latV S L (· = v)

theorem **wittExtendByZero_mem_completedHomLattice** {T : HSp D} {LT : Lat T} {P : Pl D → Prop} (hL : IsSelfDualC S L) (hL₀ : IsSelfDualC S₀ L₀) (hT : IsSelfDualC T LT) (h : ∀ v, P v → MonoAt d L L₀ v) {w : WO S₀ T} (hw : w ∈ latO S₀ T (L₀ : Set _) (LT : Set _) P) : d.ext0 T w ∈ latO S T (L : Set _) (LT : Set _) P

theorem **wittCoreInclusion_mem_completedHomLattice** {T : HSp D} {LT : Lat T} {P : Pl D → Prop} (hL : IsSelfDualC S L) (hL₀ : IsSelfDualC S₀ L₀) (hT : IsSelfDualC T LT) (h : ∀ v, P v → MonoAt d L L₀ v) {w : WO T S₀} (hw : w ∈ latO T S₀ (LT : Set _) (L₀ : Set _) P) : d.inc0 T w ∈ latO T S (LT : Set _) (L : Set _) P

theorem **isHalf_wittExtendByZero** {T : HSp D} {e : AE D} {w : WO S₀ T} (hw : IsHalf e w) : IsHalf e (d.ext0 T w)

theorem **isHalf_wittCoreInclusion** {T : HSp D} {e : AE D} {w : WO T S₀} (hw : IsHalf e w) : IsHalf e (d.inc0 T w)

theorem **isLatVac_of_intertwine** {m n m' n' : ℕ} {hs : Mx D m n → (Mx D m n → ℂ) → (Mx D m n → ℂ)} {hs' : Mx D m' n' → (Mx D m' n' → ℂ) → (Mx D m' n' → ℂ)} {Λ : AddSubgroup (Mx D m n)} {Λ' : AddSubgroup (Mx D m' n')} (ι : Mx D m' n' →+ Mx D m n) (hΛ : ∀ u ∈ Λ', ι u ∈ Λ) (hh : ∀ e u, IsHalf e u → IsHalf e (ι u)) (ℜ : (Mx D m n → ℂ) → (Mx D m' n' → ℂ)) {f : Mx D m n → ℂ} (hF : ∀ u ∈ Λ', ℜ (hs (ι u) f) = hs' u (ℜ f)) (hf : IsLatVac hs Λ f) : IsLatVac hs' Λ' (ℜ f)

theorem **latO_empty** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {P : Pl D → Prop} (hP : ∀ v, ¬ P v) {u : WO S T} (hu : u ∈ latO S T LS LT P) : u = 0

theorem **vac_of_heisLaws** {ψ : AddChar (AF D.host.F) ℂ} {V V₀ W W₀ : HSp D} {ρ σ : ℕ} {dV : HypDec V V₀ ρ} {dW : HypDec W W₀ σ} {x : Mx D W.rank ρ} {y : Mx D V₀.rank σ} {P : Pl D → Prop} {ℜ : (WO V W → ℂ) → (WO V₀ W₀ → ℂ)} (H : HeisLaws ψ dV dW x y P ℜ) {LV : Lat V} {LV₀ : Lat V₀} {LW : Lat W} {LW₀ : Lat W₀} (h : ∀ v, P v → IsSelfDualC V LV ∧ IsSelfDualC V₀ LV₀ ∧ IsSelfDualC W LW ∧ IsSelfDualC W₀ LW₀ ∧ MonoAt dV LV LV₀ v ∧ MonoAt dW LW LW₀ v) {Φ : WO V W → ℂ} (hΦ : Vac ψ V W LV LW P Φ) : Vac ψ V₀ W₀ LV₀ LW₀ P (ℜ Φ)

theorem **flatVac** : ∀ D : D7, GlobalTheory.FinPl.FlatVac D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapPreservesVacuumProofStatement

theorem **_root_.GlobalTheory.T.flatMapPreservesVacuum_proof** : ∀ D : D7, GlobalTheory.FinPl.FlatVac D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapRowEqZeroUnlessRowMemDualLattice

theorem **val_row_le_of_Phi_ne_zero** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {s κ : ℕ} (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t] (j c : ℕ) (H : j + 1 + c = s) (r : ℕ) (hc : c = 2 * r) (w : Matrix (XIx (Fin κ) t (j + 1) ⊕ Fin (c + 1)) (XIx (Fin κ) t j ⊕ Fin (c + 1 + 1)) k) (hw : (S.T t).Phi j (c + 1) (by omega) w ≠ 0) (y : XIx (Fin κ) t j) : valuation k (w (Sum.inr ⟨r, by omega⟩) (Sum.inl y)) ≤ valuation k (((S.ϖ ^ (r + 1) : kˣ)) : k)

theorem **flRow_eq_zero_of_not_mem_box** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], ∀ j c (H : j + 1 + c = s) (r : ℕ), c = 2 * r → ∀ x : Matrix (XIx (Fin κ) t (j + 1) ⊕ Unit) (XIx (Fin κ) t j) k, x (Sum.inr ()) ∉ box (XIx (Fin κ) t j) (S.ϖ ^ (r + 1)) → GlobalTheory.LocFlat.flRow S.ψ ν (r + 1) ((S.T t).Phi j (c + 1) (by omega)) x = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapRowEqZeroUnlessRowMemDualLatticeStatement

theorem **_root_.GlobalTheory.T.flatMapRow_eq_zero_unless_row_mem_dualLattice** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], ∀ j c (H : j + 1 + c = s) (r : ℕ), c = 2 * r → ∀ x : Matrix (XIx (Fin κ) t (j + 1) ⊕ Unit) (XIx (Fin κ) t j) k, x (Sum.inr ()) ∉ box (XIx (Fin κ) t j) (S.ϖ ^ (r + 1)) → GlobalTheory.LocFlat.flRow S.ψ ν (r + 1) ((S.T t).Phi j (c + 1) (by omega)) x = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapSupportFirstHalfAtLevelPlace

theorem **e_mul_ι** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (x : k) : p.e * p.ι x = p.ι x

theorem **ι_mul'** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (x y : k) : p.ι (x * y) = p.ι x * p.ι y

theorem **ι_one'** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) : p.ι 1 = p.e

theorem **shift_mul** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (x y : k) : (p.ι x + (1 - p.e)) * (p.ι y + (1 - p.e)) = p.ι (x * y) + (1 - p.e)

theorem **exists_comp** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (χ : (AE D)ˣ →* Circle) : ∃ μ : kˣ →* ℂˣ, p.IsComp χ μ

theorem **card_frame** {k : Type} [Field k] {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (Fr : Frame k N Y) : Fintype.card Y = N

theorem **prH_apply** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (a : D.host.E) : prH p a = p.pr a

theorem **exists_coreFrame** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {σ : ℕ} {B : BesLev D (σ + 1)} {J : JacLev D (σ + 1)} {B' : BesLev D σ} {n₀ n₁ n₂ : Type} [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] {c : ℕ} (F : Dbl p B J B' n₀ n₁ n₂ c) : ∃ C : Frame k B.V₀.rank (n₀ ⊕ Unit), p.IsCoreFrame B F.s F.F₀ C

theorem **isLoc_one** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {V : HSp D} {Y : Type} [Fintype Y] [DecidableEq Y] (Fr : Frame k V.rank Y) : p.IsLoc Fr 1 1

theorem **gsupp1** : ∀ (D : D7), OmLoc D → FlatBF D → GlobalTheory.LocFlat.SymCollapse → GSupp1 D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapSupportFirstHalfAtLevelPlaceStatement

theorem **_root_.GlobalTheory.T.flat_map_support_first_half_at_level_place** : ∀ (D : D7), OmLoc D → FlatBF D → GlobalTheory.LocFlat.SymCollapse → GSupp1 D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapsHeisenbergTranslationLaw

theorem **psiE_mul_star** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (a : AE D) : psiE ψ a * star (psiE ψ a) = 1

theorem **measEmb_val** (ρ : ℕ) : MeasurableEmbedding (fun u : UX D ρ => u.1.1)

theorem **measEmb_add** {ρ : ℕ} (N : Mx D ρ ρ) : MeasurableEmbedding (fun m : Mx D ρ ρ => m + N)

theorem **flatAt_shift** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (μS : Measure (Mx D W.rank ρ)) (ℱS : Set (Mx D W.rank ρ)) (μT : Measure (Mx D σ V₀.rank)) (ℱT : Set (Mx D σ V₀.rank)) (νU : Measure (UX D ρ)) (hν : νU.IsHaarMeasure) (x : Mx D W.rank ρ) (y : Mx D V₀.rank σ) (f : WO V W → ℂ) (N : Mx D ρ ρ) (hN : ∀ i j, j ≤ i → N i j = 0) : flatAt ψ dV dW μS ℱS μT ℱT νU x y (heisO ψ V W (dV.extX W (x · N)) f) = psiE ψ (upSum N) • flatAt ψ dV dW μS ℱS μT ℱT νU x y f

theorem **flatShift** : ∀ D : D7, GlobalTheory.FinPl.FlatShift D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapsHeisenbergTranslationLawStatement

theorem **_root_.GlobalTheory.T.flatMaps_heisenbergTranslation_law** : ∀ D : D7, GlobalTheory.FinPl.FlatShift D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapsPreserveSchwartzBruhatAndSurjective

theorem **flatSB_isLocallyConstant_indicator** {X : Type*} [TopologicalSpace X] {W : Set X} (hW : IsClopen W) (a : ℂ) : IsLocallyConstant (W.indicator fun _ => a)

theorem **affPt_eq_basePt_add_freeMat** (p : ℕ) (x₀ : Matrix n n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) : affPt p x₀ e = basePt c x₀ + freeMat p e

theorem **ch_add'** (a b : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : ch (a + b) = ch a + ch b

theorem **ch_basePt'** (x₀ : Matrix n n' k) : ch (basePt c x₀) = 0

theorem **ch_affPt_zero_eq** (p : ℕ) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) : ch (affPt p (0 : Matrix n n' k) e) = ch (freeMat p e)

theorem **freeMat_apply_free** (p : ℕ) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) (rs : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs}) : freeMat p e rs.1.1 rs.1.2 = e rs

theorem **continuous_freeMat** (p : ℕ) : Continuous (freeMat (k

theorem **isClosedEmbedding_freeMat** (p : ℕ) : Topology.IsClosedEmbedding (freeMat (k

theorem **exists_clopen_ch_near** (ψ : AddChar k Circle) (hψ : Continuous ψ) : ∃ W : Set (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), IsClopen W ∧ IsCompact W ∧ (0 : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) ∈ W ∧ ∀ z ∈ W, ‖((ψ (-ch z) : Circle) : ℂ) - 1‖ < 1 / 2

theorem **flatSB_generic** {X : Type} [TopologicalSpace X] [T2Space X] (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (A : X → ({rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (hA : Continuous fun q : X × ({rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) => A q.1 q.2) (R : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → X) (hR : Continuous R) (Q : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (hQ : Continuous Q) (hRA : ∀ x e, R (A x e) = x) (hQA : ∀ x e, Q (A x e) = freeMat p e) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (hΦ : Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) : (fun x => ∫ e, Φ (A x e) * (ψ (-ch (affPt p (0 : Matrix n n' k) e)) : ℂ) ∂(Measure.pi fun _ => ν)) ∈ SB X

theorem **flatOnto_generic** {X : Type} [TopologicalSpace X] [T2Space X] (ψ : AddChar k Circle) (hψ : Continuous ψ) (ν : Measure k) (hν : ν.IsAddHaarMeasure) (p : ℕ) (A : X → ({rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (R : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → X) (hR : Continuous R) (Q : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (hQ : Continuous Q) (G : X × Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (hG : Continuous G) (hGRQ : ∀ z, G (R z, Q z) = z) (hRA : ∀ x e, R (A x e) = x) (hQA : ∀ x e, Q (A x e) = freeMat p e) (Ψ : X → ℂ) (hΨ : Ψ ∈ SB X) : ∃ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), ∀ x, ∫ e, Φ (A x e) * (ψ (-ch (affPt p (0 : Matrix n n' k) e)) : ℂ) ∂(Measure.pi fun _ => ν) = Ψ x

theorem **continuous_restOf** (rd : n ⊕ Fin c → n' ⊕ Fin (c + 1) → Bool) : Continuous (restOf (k

theorem **continuous_glue** {X : Type} [TopologicalSpace X] (rd : n ⊕ Fin c → n' ⊕ Fin (c + 1) → Bool) {E : X → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k} (hE : Continuous E) : Continuous (glue (k

theorem **glue_restOf** {X : Type} (rd : n ⊕ Fin c → n' ⊕ Fin (c + 1) → Bool) (E : X → Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (R : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → X) (hER : ∀ z r s, rd r s = true → E (R z) r s = z r s) (z : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : glue rd E (R z, restOf rd z) = z

theorem **continuous_readCol** (p : ℕ) (hp : p - 1 < c + 1) : Continuous (readCol (k

theorem **continuous_extCol** : Continuous (extCol (k

theorem **extCol_readCol** (p : ℕ) (hp : p - 1 < c + 1) (z : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (r s) (h : rdCol p r s = true) : extCol (readCol p hp z) r s = z r s

theorem **readCol_ptCol** (p : ℕ) (hp1 : 1 ≤ p) (hp : p - 1 < c + 1) (x : Matrix n (n' ⊕ Unit) k) e : readCol p hp (ptCol p x e) = x

theorem **restOf_ptCol** (p : ℕ) (x : Matrix n (n' ⊕ Unit) k) e : restOf (rdCol p) (ptCol (c

theorem **continuous_readRow** (p : ℕ) (hp : p - 1 < c) : Continuous (readRow (k

theorem **continuous_extRow** : Continuous (extRow (k

theorem **extRow_readRow** (p : ℕ) (hp : p - 1 < c) (z : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) (r s) (h : rdRow p r s = true) : extRow (readRow p hp z) r s = z r s

theorem **readRow_ptRow** (p : ℕ) (hp1 : 1 ≤ p) (hp : p - 1 < c) (x : Matrix (n ⊕ Unit) n' k) e : readRow p hp (ptRow p x e) = x

theorem **restOf_ptRow** (p : ℕ) (x : Matrix (n ⊕ Unit) n' k) e : restOf (rdRow p) (ptRow (c

theorem **continuous_ptCol** (p : ℕ) : Continuous fun q : Matrix n (n' ⊕ Unit) k × ({rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) => ptCol p q.1 q.2

theorem **continuous_ptRow** (p : ℕ) : Continuous fun q : Matrix (n ⊕ Unit) n' k × ({rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) => ptRow p q.1 q.2

theorem **flCol_eq_integral** (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) : flCol ψ ν p Φ = fun x => ∫ e, Φ (ptCol p x e) * (ψ (-ch (affPt p (0 : Matrix n n' k) e)) : ℂ) ∂(Measure.pi fun _ => ν)

theorem **flRow_eq_integral** (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) : flRow ψ ν p Φ = fun x => ∫ e, Φ (ptRow p x e) * (ψ (-ch (affPt p (0 : Matrix n n' k) e)) : ℂ) ∂(Measure.pi fun _ => ν)

theorem **flat_SB_onto** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → (p ≤ c + 1 → (∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flCol ψ ν p Φ ∈ SB _) ∧ ∀ Ψ ∈ SB (Matrix n (n' ⊕ Unit) k), ∃ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flCol ψ ν p Φ = Ψ) ∧ (p ≤ c → (∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flRow ψ ν p Φ ∈ SB _) ∧ ∀ Ψ ∈ SB (Matrix (n ⊕ Unit) n' k), ∃ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flRow ψ ν p Φ = Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapsPreserveSchwartzBruhatAndSurjectiveStatement

theorem **_root_.GlobalTheory.T.flatMaps_preserve_schwartzBruhat_and_surjective** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → (p ≤ c + 1 → (∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flCol ψ ν p Φ ∈ SB _) ∧ ∀ Ψ ∈ SB (Matrix n (n' ⊕ Unit) k), ∃ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flCol ψ ν p Φ = Ψ) ∧ (p ≤ c → (∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flRow ψ ν p Φ ∈ SB _) ∧ ∀ Ψ ∈ SB (Matrix (n ⊕ Unit) n' k), ∃ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), GlobalTheory.LocFlat.flRow ψ ν p Φ = Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapsSatisfyFlatLawsFinitePlaces

theorem **mixT_smul** {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (ψ : AddChar (AF D.host.F) ℂ) (μ : Measure (Mx D ρ S.rank)) (ℱ : Set (Mx D ρ S.rank)) (c : ℂ) (f : WO S T → ℂ) (y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (c • f) y = c • d.mixT S ψ μ ℱ f y

theorem **symplO_eq_of_symplOH** {S T : HSp D} {a b a' b' : WO S T} (h : symplOH S T a b = symplOH S T a' b') : symplO S T a b = symplO S T a' b'

theorem **flatAt_dual** (hψ : IsStdChar D ψ (-1)) (f : WO V W → ℂ) (hf : IsThetaFunO ψ V W f) (a : Mx D W.rank ρ) (ha : GlobalTheory.FinPl.UInv W x a) : flatAt ψ dV dW vS.μ vS.ℱ vT.μ vT.ℱ νU x y (heisO ψ V W (dV.extY W a) f) = ψ (symplO V W (dV.extX W x) (dV.extY W a)) • flatAt ψ dV dW vS.μ vS.ℱ vT.μ vT.ℱ νU x y f

theorem **flatAt_iso** (hψ : IsStdChar D ψ (-1)) (f : WO V W → ℂ) (hf : IsThetaFunO ψ V W f) (ξ : Mx D σ V₀.rank) : flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 νU x y (heisO ψ V W (dV.ext0 W (dW.incX V₀ ξ)) f) = ψ (symplO V₀ W (dW.incAdj V₀ y) (dW.incX V₀ ξ)) • flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 νU x y f

theorem **flatAt_core** (hψ : IsStdChar D ψ (-1)) (f : WO V W → ℂ) (hf : IsThetaFunO ψ V W f) (w₀ : WO V₀ W₀) : flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 νU x y (heisO ψ V W (dV.ext0 W (dW.inc0 V₀ w₀)) f) = heisO ψ V₀ W₀ w₀ (flatAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 νU x y f)

theorem **flatBF_equiv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (f : WO B.V J.W → ℂ), IsThetaFunO ψ B.V J.W f → ∀ n ∈ flagN J.d, ∀ h : GA D J.W₀, flatBF s (s.om (1, n ∗ J.d.emb0 h) f) = star (psiFlag ψ J.d n) • opA ψ J.d B.o B.ζ s.om₀ n h (flatBF s f)

theorem **flatFB_equiv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B) (f : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V f → ∀ n ∈ flagN B.d, ∀ h : GA D B.A, flatFB s (s.om (1, n ∗ embA B.d B.o h) f) = star (psiBes ψ B.d 1 (vecB B.o B.ζ) n) • s.om₀ (1, B.o.emb (h, 1)) (flatFB s f)

theorem **flatLaws_all** (D : D7) : GlobalTheory.FinPl.FlatLaws D

theorem **GlobalTheory.FinPl.flatLaws_all** : ∀ D : D7, GlobalTheory.FinPl.FlatLaws D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatMapsSatisfyFlatLawsFinitePlacesStatement

theorem **_root_.GlobalTheory.T.flatMapsSatisfyFlatLaws_finitePlaces** : ∀ D : D7, GlobalTheory.FinPl.FlatLaws D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatPeriodEqIntegralEvaluationPeriods

theorem **integral_triple** (α : Measure A) (β : Measure B) (γ : Measure C) [IsFiniteMeasure α] [IsFiniteMeasure β] [IsFiniteMeasure γ] {f : A × B × C → ℂ} (hf : StronglyMeasurable f) {c : ℝ} (hc : ∀ p, ‖f p‖ ≤ c) : ∫ p, f p ∂α.prod (β.prod γ) = ∫ a, ∫ b, ∫ x, f (a, b, x) ∂γ ∂β ∂α

theorem **exchange** (ν : Measure U) [SFinite ν] (α : Measure A) (β : Measure B) (γ : Measure C) [IsFiniteMeasure α] [IsFiniteMeasure β] [IsFiniteMeasure γ] {H : U × A × B × C → ℂ} (hH : StronglyMeasurable H) {M : U → ℝ} (hM : Integrable M ν) (hb : ∀ p, ‖H p‖ ≤ M p.1) : Integrable (fun u => ∫ a, ∫ b, ∫ x, H (u, a, b, x) ∂γ ∂β ∂α) ν ∧ ∫ a, ∫ b, ∫ x, ∫ u, H (u, a, b, x) ∂ν ∂γ ∂β ∂α = ∫ u, ∫ a, ∫ b, ∫ x, H (u, a, b, x) ∂γ ∂β ∂α ∂ν

theorem **setIntegral_mul_right** {G : Type} [Group G] [MeasurableSpace G] [MeasurableMul G] (Γ : Subgroup G) [Countable Γ] {ν : Measure G} [ν.IsMulLeftInvariant] [ν.IsMulRightInvariant] {𝒩 : Set G} (h : IsFundamentalDomain Γ 𝒩 ν) (f : G → ℂ) (hf : ∀ γ ∈ Γ, ∀ n, f (γ ∗ n) = f n) (a : G) : ∫ n in 𝒩, f (n ∗ a) ∂ν = ∫ n in 𝒩, f n ∂ν

theorem **sFinite_haar** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [MeasurableSpace G] (ν : Measure G) [ν.IsHaarMeasure] : SFinite ν

theorem **yZero_mem** (S : HSp D) (ρ : ℕ) (z : Fin S.rank → D.host.E) : yZero S ρ z ∈ ratMx D S.rank ρ

theorem **jacT_rat** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) {y : Mx D S.rank ρ} (hy : y ∈ ratMx D S.rank ρ) {γ : GA D T} (hγ : γ ∈ GQ T) {f : WO S T₀ → ℂ} (hf : IsThetaFunO ψ S T₀ f) : jacT ψ d S y γ f 0 = f 0

theorem **norm_jacT** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) (n : GA D T) (f : WO S T₀ → ℂ) : ‖jacT ψ d S y n f 0‖ = ‖f (ptT d S y n)‖

theorem **levi_commutes_with_ext_zero** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (a : GL (Fin ρ) (AE D)) (h : GA D S₀) : d.levi a ∗ d.emb0 h = d.emb0 h ∗ d.levi a

theorem **ev_theta** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (x : Mx D J.W.rank ρ) : IsThetaFunO ψ B.V₀ J.W₀ (ev s Ψ x)

theorem **om_om** (p q : GA D B.V × GA D J.W) (Ψ : WO B.V J.W → ℂ) : s.om p (s.om q Ψ) = s.om (p ∗ q) Ψ

theorem **ev_om** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (v : UX D ρ) (h₀ : GA D B.A) {n : GA D J.W} (hn : n ∈ flagN J.d) (h' : GA D J.W₀) : ev s (s.om (embA B.d B.o h₀, n ∗ J.d.emb0 h') Ψ) (mt J.W n · (xZero J.d · v.1.1)) = s.om₀ (B.o.emb (h₀, 1), 1) (opA ψ J.d B.o B.ζ s.om₀ n h' (ev s Ψ (xZero J.d · v.1.1)))

theorem **ev_jac** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (v : UX D ρ) {n : GA D J.W} (hn : n ∈ flagN J.d) : ev s (s.om (1, n) Ψ) (mt J.W n · (xZero J.d · v.1.1)) = jacT ψ J.d B.V₀ (yZero B.V₀ ρ (vecB B.o B.ζ)) n (ev s Ψ (xZero J.d · v.1.1))

theorem **ev_core** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (v : UX D ρ) (h₀ : GA D B.A) (h' : GA D J.W₀) : ev s (s.om (embA B.d B.o h₀, J.d.emb0 h') Ψ) (xZero J.d · v.1.1) = s.om₀ (B.o.emb (h₀, 1), h') (ev s Ψ (xZero J.d · v.1.1))

theorem **ker_eq** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (v : UX D ρ) (h₀ : GA D B.A) {n : GA D J.W} (hn : n ∈ flagN J.d) (h' : GA D J.W₀) : fjKer ψ J.d B.o B.ζ s.om₀ (ev s Ψ (xZero J.d · v.1.1)) h₀ n h' = jacT ψ J.d B.V₀ (yZero B.V₀ ρ (vecB B.o B.ζ)) n (ev s (s.om (embA B.d B.o h₀, J.d.emb0 h') Ψ) (xZero J.d · v.1.1)) 0

theorem **ker_left** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (v : UX D ρ) (h₀ : GA D B.A) {γ n : GA D J.W} (hγ : γ ∈ flagN J.d) (hγ' : γ ∈ GQ J.W) (hn : n ∈ flagN J.d) (h' : GA D J.W₀) : fjKer ψ J.d B.o B.ζ s.om₀ (ev s Ψ (xZero J.d · v.1.1)) h₀ (γ ∗ n) h' = fjKer ψ J.d B.o B.ζ s.om₀ (ev s Ψ (xZero J.d · v.1.1)) h₀ n h'

theorem **ker_flat** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (h₀ : GA D B.A) {n : GA D J.W} (hn : n ∈ flagN J.d) (h' : GA D J.W₀) : fjKer ψ J.d B.o B.ζ s.om₀ (flatBF s Ψ) h₀ n h' = ∫ v : UX D ρ, star (psiUni ψ v.1) ∗ fjKer ψ J.d B.o B.ζ s.om₀ (ev s Ψ (xZero J.d · v.1.1)) h₀ n h' ∂s.νU.ν

theorem **ker_bound** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) : ∃ M : UX D ρ → ℝ, Integrable M s.νU.ν ∧ ∀ (v : UX D ρ) h₀, ∀ n ∈ flagN J.d, ∀ h', ‖star (psiUni ψ v.1) ∗ fjKer ψ J.d B.o B.ζ s.om₀ (ev s Ψ (xZero J.d · v.1.1)) h₀ n h'‖ ≤ M v

theorem **ker_cont** {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) : Continuous fun p : UX D ρ × GA D J.W₀ × flagN J.d × GA D B.A => fjKer ψ J.d B.o B.ζ s.om₀ (ev s Ψ (xZero J.d · p.1.1.1)) p.2.2.2 p.2.2.1.1 p.2.1

theorem **cpt_facts** {V : HSp D} (hV : CoCpt V) (v : Vol V) : IsFiniteMeasure (v.μ.restrict v.𝓕) ∧ ∀ φ, IsContAut V φ → ∃ c, ∀ g, ‖φ g‖ ≤ c

theorem **psiFlag_levi** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (a : GL (Fin ρ) (AE D)) : psiFlag ψ d (d.levi a) = psiUni ψ a

theorem **per_levi** {ψ : AddChar (AF D.host.F) ℂ} {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {om' : Om B.V₀ J.W} (hom : IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om') {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) (φ : GA D B.A → ℂ) {G : GA D J.W → ℂ} (hG : ∀ γ ∈ GQ J.W, ∀ g, G (γ ∗ g) = G g) (u v : UX D ρ) : J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1) G) (ev s Ψ (xZero J.d · v.1.1)) = star (psiUni ψ (u⁻¹).1) ∗ J.perAll ψ B s.om₀ φ G (ev s Ψ (xZero J.d · v.1.1))

theorem **perAll_flat_eq_integral** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut

B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∀ h, Integrable (fun u : UX D ρ => J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1 ∗ h) G) (evalBF s (s.om (1, (J.d.levi u.1 ∗ h)) Φ))) s.vU.ν ∧ J.perAll ψ B s.om₀ φ (rTrans h G) (flatBF s (s.om (1, h) Φ)) = ∫ u : UX D ρ, J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1 ∗ h) G) (evalBF s (s.om (1, (J.d.levi u.1 ∗ h)) Φ)) ∂s.vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatPeriodEqIntegralEvaluationPeriodsStatement

theorem **_root_.GlobalTheory.T.flatPeriod_eq_integral_evaluationPeriods** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (om' : Om B.V₀ J.W), IsWeilDatumO ψ B.V₀ J.W s.χV s.χW om' → ∀ (φ : GA D B.A → ℂ) (G : GA D J.W → ℂ) (Φ : WO B.V J.W → ℂ), IsContAut B.A φ → RapidDec J.W G → IsThetaFunO ψ B.V J.W Φ → ∀ h, Integrable (fun u : UX D ρ => J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1 ∗ h) G) (evalBF s (s.om (1, (J.d.levi u.1 ∗ h)) Φ))) s.vU.ν ∧ J.perAll ψ B s.om₀ φ (rTrans h G) (flatBF s (s.om (1, h) Φ)) = ∫ u : UX D ρ, J.perAll ψ B s.om₀ φ (rTrans (J.d.levi u.1 ∗ h) G) (evalBF s (s.om (1, (J.d.levi u.1 ∗ h)) Φ)) ∂s.vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatRowIndicatorOfSymLevelExchange

theorem **lam_rowE_symLevel** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], ∀ j c (H : j + 1 + c = s) (r : ℕ), c = 2 ∗ r → ∀ u : XIx (Fin κ) t j → k, ∀ hm : rowE (c + 1 + 1) (r + 1) u ∈ ((S.T t).lev j (c + 1 + 1) (by omega)).J, ((S.T t).lev j (c + 1 + 1) (by omega)).lam ⟨_, hm⟩ = 1

theorem **pureRow_of_exchange** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], S.Pattern t → S.ExchangeAt t ν → GlobalTheory.LocFlat.PureRow S t ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatRowIndicatorOfSymLevelExchangeStatement

theorem **_root_.GlobalTheory.T.flatRowIndicator_of_symLevelExchange** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (s κ : ℕ) (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], S.Pattern t → S.ExchangeAt t ν → GlobalTheory.LocFlat.PureRow S t ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatRowMapEqRootExchangeOperatorTranslate

theorem **flRow_eq_opE_rowE** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), ∀ x : Matrix (n ⊕ Unit) n' k, GlobalTheory.LocFlat.flRow ψ ν p Φ x = opE ψ ν p (fun z => Φ (z ∗ (rowE (c + 1) p (x (Sum.inr ())))).val)) x.toRows₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatRowMapEqRootExchangeOperatorTranslateStatement

theorem **_root_.GlobalTheory.T.flatRowMap_eq_rootExchangeOperator_translate** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ Φ ∈ SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k), ∀ x : Matrix (n ⊕ Unit) n' k, GlobalTheory.LocFlat.flRow ψ ν p Φ x = opE ψ ν p (fun z => Φ (z ∗ (rowE (c + 1) p (x (Sum.inr ())))).val)) x.toRows₁

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatSchwartzTargetUnipotentConjBesselCharacter

theorem **levi_mem_flag** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {u : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) : d.levi u ∈ flagN d

theorem **psiUni_mul_of_mem** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {u v : GL (Fin ρ) (AE D)} (hu : u ∈ UX D ρ) (hv : v ∈ UX D ρ) : psiUni ψ (u ∗ v) = psiUni ψ u ∗ psiUni ψ v

theorem **flatFB_flagN** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (Ψ : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V Ψ → ∀ n ∈ flagN B.d, flatFB s (s.om (1, n) Ψ) = star (psiBes ψ B.d 1 (vecB B.o B.ζ) n) • flatFB s Ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatSchwartzTargetUnipotentConjBesselCharacterStatement

theorem **_root_.GlobalTheory.T.flatSchwartz_targetUnipotent_conj_besselCharacter** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (Ψ : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V Ψ → ∀ n ∈ flagN B.d, flatFB s (s.om (1, n) Ψ) = star (psiBes ψ B.d 1 (vecB B.o B.ζ) n) • flatFB s Ψ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatSectionParabolicChange

theorem **degPS_chart_isSection** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ κ : GA D (dblSp A) ≃∗ G D A.rank, IsChart A μ κ → ∀ (a : GL (Fin A.rank) (AE D)) (p : siegel A), IsConj A a κ p → ∀ (χ : (AE D)ˣ →∗ Circle) (s₁ : ℂ), ∀ φ ∈ degPS A a χ s₁, IsSection χ s₁ (fun x => φ (κ.symm x ∗ (p : GA D (dblSp A))⁻¹)) ∧ ∀ (s : ℂ) (g : GA D (dblSp A)), flat A a φ s₁ s g = npow (absA D (aP A p)) (s₁ − s) ∗ stdSec (hgtStd A a κ p) (fun x => φ (κ.symm x ∗ (p : GA D (dblSp A))⁻¹)) s₁ s (κ (g ∗ p))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatThetaIntegrandLocallyUniformMajorant

theorem **flatFun_majorant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (χV χW : (AE D)ˣ →∗ Circle) (om : Om V W), IsWeilDatumO ψ V W χV χW om → ∀ (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∀ Φ, IsThetaFunO ψ V W Φ → ∀ K : Set (GA D V × GA D W), IsCompact K → ∃ M : UX D ρ → ℝ, Integrable M vU.ν ∧ ∀ p ∈ K, ∀ y w u, ‖flatFun ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (rat x) y (om p Φ) w u‖ ≤ M u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatThetaIntegrandLocallyUniformMajorantStatement

theorem **_root_.GlobalTheory.T.flatThetaIntegrand_locallyUniform_majorant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W W₀ : HSp D) (ρ σ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) (χV χW : (AE D)ˣ →∗ Circle) (om : Om V W), IsWeilDatumO ψ V W χV χW om → ∀ (vU : VolU D ρ) (x : Matrix (Fin W.rank) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∀ Φ, IsThetaFunO ψ V W Φ → ∀ K : Set (GA D V × GA D W), IsCompact K → ∃ M : UX D ρ → ℝ, Integrable M vU.ν ∧ ∀ p ∈ K, ∀ y w u, ‖flatFun ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (rat x) y (om p Φ) w u‖ ≤ M u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatThetaMapsConvergeThetaFunctions

theorem **coreMap_eq** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) : ⇑(coreMap dV dW) = fun w => (dW.j0 ∗ w) ∗ dV.p0

theorem **continuous_coreMap** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) : Continuous (coreMap dV dW)

theorem **coreMap_finPart** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (w : WO V₀ W₀) (h : w.map (finPart D) = 0) : (coreMap dV dW w).map (finPart D) = 0

theorem **coreMap_infPart** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (w : WO V₀ W₀) (h : w.map (infPart D) = 0) : (coreMap dV dW w).map (infPart D) = 0

theorem **flatLaw** : ∀ D : D7, FlatLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatThetaMapsConvergeThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.flatThetaMaps_converge_thetaFunctions** : ∀ D : D7, FlatLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatThetaMapsPreserveThetaFunctions

theorem **rat_lastQ** (B : BesLev D σ) : rat (lastQ B) = xLast B.d (vecB B.o B.ζ)

theorem **zeta_ne** (B : BesLev D σ) : B.ζ ≠ 0

theorem **lastQ_indep** (B : BesLev D σ) : LinearIndependent D.host.E (lastQ B).transpose

theorem **flat_isThetaFun** : ∀ D : D7, FlatLaw D → ∀ (ψ : AddChar (AF D.host.F) ℂ), (∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) Ψ, IsThetaFunO ψ B.V J.W Ψ → IsThetaFunO ψ B.V₀ J.W₀ (flatBF s Ψ)) ∧ ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B) Ψ, IsThetaFunO ψ J.W B.V Ψ → IsThetaFunO ψ J.W₀ B.V₀ (flatFB s Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FlatThetaMapsPreserveThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.flatThetaMaps_preserve_thetaFunctions** : ∀ D : D7, FlatLaw D → ∀ (ψ : AddChar (AF D.host.F) ℂ), (∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) Ψ, IsThetaFunO ψ B.V J.W Ψ → IsThetaFunO ψ B.V₀ J.W₀ (flatBF s Ψ)) ∧ ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B) Ψ, IsThetaFunO ψ J.W B.V Ψ → IsThetaFunO ψ J.W₀ B.V₀ (flatFB s Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FnOfMapInjective

theorem **fnOf_map_injective** {K : Type} [Field K] {ι : Type} [Fintype ι] {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (sh : ι → ℤ) (s : K →+* ℂ) {F F' : MvPolynomial (ι × Fin n × Fin n) K} (h : fnOf G sh 1 (MvPolynomial.map s F) = fnOf G sh 1 (MvPolynomial.map s F')) : F = F'

theorem **exists_ptVal_at** (D : D7) (u : D.what₂.Word) (K : ClsPTied u) (W : ValPTied u K) (x : (D.fm u).ptLat.V) (c : (D.fm u).clsP) : ∃ F : VV D.what₂.LL (sgT u), W.eW ((D.fm u).ptEmb x) c = fnOf _ (shiftOf (sgT u)) 1 (MvPolynomial.map D.what₂.embLL F.1)

theorem **exists_ptVal** : ∀ (D : D7) (u : D.what₂.Word) (K : ClsPTied u) (W : ValPTied u K), ∃ val : (D.fm u).ptLat.V →ₗ[D.what₂.LL] ((D.fm u).clsP → VV D.what₂.LL (sgT u)), (∀ x c, W.eW ((D.fm u).ptEmb x) c = fnOf _ (shiftOf (sgT u)) 1 (MvPolynomial.map D.what₂.embLL (val x c).1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockCoefficientEqConstMulLowestKTypeCoefficient

theorem **pair_of_ext** (θ : InfFam D) (hθ : IsInfFam θ) (M : DSModel W S') (hLS : LowSm M) (hLI : GlobalTheory.IndefiniteRowSideFullHecke.HolomorphicLowestType.LowIrr M) (hω : IsWeilDatumO ψ V W χV χW ω) (hcv : IsCovolOne (ratL W V) μ 𝓕) (J : MultilinearMap ℂ (fun ρ => lowT M ρ) (WO V W → ℂ)) (hJ : ∀ v, IsThetaFunO ψ V W (J v)) (hK : ∀ τ, ∀ k ∈ Formalism.REAL.Kset (M.ε τ), ∀ (v : ∀ ρ, lowT M ρ) (u : lowT M τ), ((u : M.sm τ) : M.H τ) = M.σ τ k ((v τ : M.sm τ) : M.H τ) → J (Function.update v τ u) = ω (1, atTau θ W τ ((M.fr τ)⁻¹ * k * M.fr τ)) (J v)) (hX : ∀ v : (∀ ρ, lowT M ρ), IsExtVec M (fun ρ => (v ρ : M.sm ρ)) → ∀ u ∈ GinfC W, thetaIP μ 𝓕 (J v) (ω (1, u) (J v)) = thetaIP μ 𝓕 (J v) (J v) * ∏ ρ, ⟪((v ρ : M.sm ρ) : M.H ρ), M.rep ρ (infMat D.sp.cm.sig W.gram (θ ρ) u) ((v ρ : M.sm ρ) : M.H ρ)⟫_ℂ) : ∃ N : ℂ, ∀ (v v' : ∀ ρ, lowT M ρ), ∀ g ∈ GinfC W, thetaIP μ 𝓕 (J v) (ω (1, g) (J v')) = N * ∏ ρ,

theorem **coef_one_vector** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (Sp : D.what₂.Tau → SPar) (M : DSModel T Sp), GlobalTheory.HolDict.HolShape M → (∃ wt, GlobalTheory.HolDict.WtIs M wt) → ∀ (μ : Measure (WO S T)) (𝓧 : Set (WO S T)), IsCovolOne (ratL T S) μ 𝓧 → ∀ c : CoefA S, c.Is → ∀ pt : PolF D T.rank S.rank →ₗ[ℂ] (WO S T → ℂ), (∀ P, IsThetaFunO ψ S T (pt P)) → ∀ (Δ : PolF D T.rank S.rank) (io : LowT M →ₗ[ℂ] c.W →ₗ[ℂ] PolF D T.rank S.rank), IsIoW θ M c om (fun P => pt P) Δ io → GlobalTheory.RawLift.FockCoef.Killed om θ M (pt Δ) → ∀ w : c.W, ∃ N : ℂ, ∀ g : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ, (∀ ρ, g ρ ∈ Formalism.QuantumDeterminants.UH (gramAt D T ρ)) → ∀ (e e' : ∀ ρ, lowT M ρ), l2 μ 𝓧 (om (1, realElt θ T g) (pt (io (PiTe

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockCoefficientEqConstMulLowestKTypeCoefficientStatement

theorem **_root_.GlobalTheory.T.fockCoefficient_eq_const_mul_lowestKTypeCoefficient** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (Sp : D.what₂.Tau → SPar) (M : DSModel T Sp), GlobalTheory.HolDict.HolShape M → (∃ wt, GlobalTheory.HolDict.WtIs M wt) → ∀ (μ : Measure (WO S T)) (𝓧 : Set (WO S T)), IsCovolOne (ratL T S) μ 𝓧 → ∀ c : CoefA S, c.Is → ∀ pt : PolF D T.rank S.rank →ₗ[ℂ] (WO S T → ℂ), (∀ P, IsThetaFunO ψ S T (pt P)) → ∀ (Δ : PolF D T.rank S.rank) (io : LowT M →ₗ[ℂ] c.W →ₗ[ℂ] PolF D T.rank S.rank), IsIoW θ M c om (fun P => pt P) Δ io → GlobalTheory.RawLift.FockCoef.Killed om θ M (pt Δ) → ∀ w : c.W, ∃ N : ℂ, ∀ g : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ, (∀ ρ, g ρ ∈ Formalism.QuantumDeterminants.UH (gramAt D T ρ)) → ∀ (e e' : ∀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockCoefficientRealPlaceTwoVacua

theorem **foldr_lin** {ι X : Type} {A : ι → (X → ℂ) → (X → ℂ)} {P : (X → ℂ) → Prop} (hA : ∀ i f, P f → P (A i f)) (hl : ∀ i (a : ℂ) f f', P f → P f' → A i (a • f + f') = a • A i f + A i f') (a : ℂ) {f f' : X → ℂ} (hf : P f) (hf' : P f') (l : List ι) : l.foldr (fun i g => A i g) (a • f + f') = a • l.foldr (fun i g => A i g) f + l.foldr (fun i g => A i g) f'

theorem **pure_lin** {ι X : Type} {A : ι → (X → ℂ) → (X → ℂ)} {P : (X → ℂ) → Prop} (hA : ∀ i f, P f → P (A i f)) (hl : ∀ i (a : ℂ) f f', P f → P f' → A i (a • f + f') = a • A i f + A i f') (a : ℂ) {f f' : X → ℂ} (hf : P f) (hf' : P f') (Q : MvPolynomial ι ℂ) : pureTensorL A (a • f + f') Q = a • pureTensorL A f Q + pureTensorL A f' Q

theorem **polar** {Y : Type} [AddCommGroup Y] [Module ℂ Y] {vac : Y → Prop} (hv : ∀ (a : ℂ) x y, vac x → vac y → vac (a • x + y)) (E : Y → Y → ℂ) (h1 : ∀ (a : ℂ) x y z, vac x → vac y → vac z → E (a • x + y) z = a * E x z + E y z) (h2 : ∀ (a : ℂ) x y z, vac x → vac y → vac z → E z (a • x + y) = conj a * E z x + E z y) (h0 : ∀ x, vac x → E x x = 0) {x y : Y} (hx : vac x) (hy : vac y) : E x y = 0

theorem **exists_signCompat** {n : ℕ} (ε : Fin n → Bool) : ∃ (p q : ℕ) (e : Ix p q ≃ Fin n), SignCompat ε e

theorem **fockIp_one** {ι : Type} : fockIp (1 : MvPolynomial ι ℂ) 1 = 1

theorem **exists_cov** {p q n : ℕ} (T : GrpData p q) {fr G : Mn n} {ε : Fin n → Bool} {e : Ix p q ≃ Fin n} (hfr : IsUnit fr) (hG : frᴴ * Formalism.REAL.Jm ε * fr = G) (hs : SignCompat ε e) {g : Mn n} (hg : g ∈ Formalism.QuantumDeterminants.UH G) : ∃ x : T.G, phiFr fr true

e ((T.cov x : GL (Ix p q) ℂ) : Mat p q) = g

theorem **ptAt_theta** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (τ : D.what₂.Tau) (sS : Mn V.rank) (sT : Mn W.rank) (BS : Mn V.rank) (BT : Mn W.rank) {f : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (P : LPol W.rank V.rank) : IsThetaFunO ψ V W (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ptAt ψ θ V W τ sS sT BS BT f P)

theorem **ptAt_lin** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (τ : D.what₂.Tau) (sS : Mn V.rank) (sT : Mn W.rank) (BS : Mn V.rank) (BT : Mn W.rank) (a : ℂ) {f f' : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (hf' : IsThetaFunO ψ V W f') (P : LPol W.rank V.rank) : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ptAt ψ θ V W τ sS sT BS BT (a • f + f') P = a • GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ptAt ψ θ V W τ sS sT BS BT f P + GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ptAt ψ θ V W τ sS sT BS BT f' P

theorem **vac_lin** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (τ : D.what₂.Tau) (J : Matrix (Fin W.rank) (Fin V.rank) ℂ → Matrix (Fin W.rank) (Fin V.rank) ℂ) (a : ℂ) {f f' : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (hf' : IsThetaFunO ψ V W f') (vf : IsFockVacAt (heisO ψ V W) θ τ J f) (vf' : IsFockVacAt (heisO ψ V W) θ τ J f') : IsFockVacAt (heisO ψ V W) θ τ J (a • f + f')

theorem **coef_vac** {S S' : D.what₂.Tau → SPar} (M : DSModel V S) (M' : DSModel W S') {χS χT : (AE D)ˣ →* Circle} {ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)} (hθ : IsInfFam θ) (hω : IsWeilDatumO ψ V W χS χT ω) {m : Measure (WO V W)} {X : Set (WO V W)} (hX : IsCovolOne (ratL W V) m X) (τ : D.what₂.Tau) {g : Mn V.rank} (hg : g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ)) (P Q : LPol W.rank V.rank) : ∃ K : ℂ, ∀ f, IsThetaFunO ψ V W f → IsFockVacAt (heisO ψ V W) θ τ (cplxO (cart M τ) (cart M' τ)) f → f ≠ 0 → GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ipTh m X (ω (atTau θ V τ g, 1) (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ptAt ψ θ V W τ (cart M τ) (cart M' τ) (onb M τ) (onb M' τ) f P)) (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ptAt ψ θ V W τ (cart M τ) (cart M' τ) (onb M τ) (onb M' τ) f Q) = GlobalTheory.IndefiniteRowSideFullHecke.Th

theorem **coef_at_place** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (χS χT : (AE D)ˣ →* Circle) (ω : (GA D V × GA D W →* Module.End ℂ (WO V W → ℂ))), IsInfFam θ → IsWeilDatumO ψ V W χS χT ω → ∀ (m : Measure (WO V W)) (X : Set (WO V W)), IsCovolOne (ratL W V) m X → ∀ (τ : D.what₂.Tau) (f F F' : WO V W → ℂ), (∀ x ∈ [f, F, F'], IsThetaFunO ψ V W x ∧ IsFockVacAt (heisO ψ V W) θ τ (cplxO (cart M τ) (cart M' τ)) x) → ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ), ∀ P Q : LPol W.rank V.rank, let pt

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockCoefficientRealPlaceTwoVacuaStatement

theorem **_root_.GlobalTheory.T.fockCoefficient_realPlace_two_vacua** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (χS χT : (AE D)ˣ →* Circle) (ω : (GA D V × GA D W →* Module.End ℂ (WO V W → ℂ))), IsInfFam θ → IsWeilDatumO ψ V W χS χT ω → ∀ (m : Measure (WO V W)) (X : Set (WO V W)), IsCovolOne (ratL W V) m X → ∀ (τ : D.what₂.Tau) (f F F' : WO V W → ℂ), (∀ x ∈ [f, F, F'], IsThetaFunO ψ V W x ∧ IsFockVacAt (heisO ψ V W) θ τ (cplxO (cart M τ) (cart M' τ)) x) → ∀ g ∈ Formalism.QuantumDeterminants.UH (gramAt D V τ), ∀ P Q : LPol W.rank V.rank, let pt

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockMatrixCoefficientSplitsRealFiniteFactors

theorem **actO_inf** : ∀ (D : D7) (S T : HSp D) (a : GA D S) (b : GA D T), a ∈ GinfC S → b ∈ GinfC T → ∀ w : WO S T, w.map (infPart D) = 0 → actO S T a b w = w

theorem **om_heisO_inf** {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (hom : IsWeilDatumO ψ S T χS χT om) {a : GA D S} (ha : a ∈ GinfC S) {b : GA D T} (hb : b ∈ GinfC T) {w : WO S T} (hw : w.map (infPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (a, b) (heisO ψ S T w f) = heisO ψ S T w (om (a, b) f)

theorem **fockVac_om_fin** {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) {x : GA D S} (hx : x ∈ GfC S) {y : GA D T} (hy : y ∈ GfC T) (τ : D.what₂.Tau) (J : Matrix (Fin T.rank) (Fin S.rank) ℂ → Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hv : IsFockVacAt (heisO ψ S T) θ τ J f) : IsFockVacAt (heisO ψ S T) θ τ J (om (x, y) f)

theorem **om_fin_pureTensorO** {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) {x : GA D S} (hx : x ∈ GfC S) {y : GA D T} (hy : y ∈ GfC T) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (P : MvPolynomial (FVar D T.rank S.rank) ℂ) : om (x, y) (pureTensorO ψ θ S T sS sT BS BT f P) = pureTensorO ψ θ S T sS sT BS BT (om (x, y) f) P

theorem **l2_eq_star** (μ : Measure (WO S T)) (𝓕 : Set (WO S T)) (f g : WO S T → ℂ) : l2 μ 𝓕 f g = star (thetaIP μ 𝓕 f g)

theorem **pair_eq_inf** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → ∀ (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (pS pT : D.what₂.Tau → ℕ), (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → (∀ τ, IsONBasis (gramAt D S τ) (sS τ) (BS τ) (pS τ)) → (∀ τ, IsONBasis (gramAt D T τ) (sT τ) (BT τ) (pT τ)) → ∀ (μ : Measure (WO S T)) (𝓕 : Set (WO S T)), IsCovolOne (ratL T S) μ 𝓕 → ∀ a ∈ GinfC S, ∀ b ∈ GinfC T, ∀ P P' : MvPolynomial (FVar D T.rank S.rank) ℂ, ∃ c : ℂ, ∀ f g : WO S T → ℂ, IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) f) → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT

theorem **coef_split** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (μ : Measure (WO S T)) (𝒳 : Set (WO S T)), IsCovolOne (ratL T S) μ 𝒳 → ∀ (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ pS pT : D.what₂.Tau → ℕ, (∀ τ, IsONBasis (gramAt D S τ) (sS τ) (BS τ) (pS τ)) → (∀ τ, IsONBasis (gramAt D T τ) (sT τ) (BT τ) (pT τ)) → ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) f) → (∀ Q, IsThetaFunO ψ S T (pureTensorO ψ θ S T sS sT BS BT f Q)) → ∀ a ∈ GinfC S, ∀ b ∈ GinfC T, ∀ x ∈ GfC S, ∀ y ∈ GfC T, ∀ Q Q', l2 μ 𝒳 (om (x ∗ a, y ∗ b) (pureTenso

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockMatrixCoefficientSplitsRealFiniteFactorsStatement

theorem **_root_.GlobalTheory.T.fockMatrixCoefficient_splits_real_finite_factors** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (μ : Measure (WO S T)) (𝒳 : Set (WO S T)), IsCovolOne (ratL T S) μ 𝒳 → ∀ (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ pS pT : D.what₂.Tau → ℕ, (∀ τ, IsONBasis (gramAt D S τ) (sS τ) (BS τ) (pS τ)) → (∀ τ, IsONBasis (gramAt D T τ) (sT τ) (BT τ) (pT τ)) → ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO (sS τ) (sT τ)) f) → (∀ Q, IsThetaFunO ψ S T (pureTensorO ψ θ S T sS sT BS BT f Q)) → ∀ a ∈ GinfC S, ∀ b ∈ GinfC T, ∀ x ∈ GfC

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockOfVac

theorem **coeff_sum_pderiv_eq_zero** {ι : Type} [Fintype ι] (c : ι → ℂ) (Q : MvPolynomial ι ℂ) (n : ℕ) (hQ : ∀ m : ι →₀ ℕ, n < m.degree → Q.coeff m = 0) (m : ι →₀ ℕ) (hm : n < m.degree + 1) : (∑ i, c i • MvPolynomial.pderiv i Q).coeff m = 0

theorem **fock_of_vac** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → IsInfFam θ → ∀ f, IsVacVec ψ θ M M' f → ∀ P, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.IsFockVec ψ θ V W (cart M) (cart M') (ptOf ψ θ M M' f P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockOfVacStatement

theorem **_root_.GlobalTheory.T.isFockVec_ptOf_of_isVacVec** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → IsInfFam θ → ∀ f, IsVacVec ψ θ M M' f → ∀ P, GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.IsFockVec ψ θ V W (cart M) (cart M') (ptOf ψ θ M M' f P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockOperatorsDistinctRealPlacesCommuteTheta

theorem **foldr_comm** {B : (X → ℂ) → (X → ℂ)} (hA : ∀ i f, P f → P (A i f)) (hBA : ∀ i f, P f → B (A i f) = A i (B f)) {f : X → ℂ} (hf : P f) (l : List ι) : B (l.foldr (fun i g => A i g) f) = l.foldr (fun i g => A i g) (B f)

theorem **pure_comm** {B : (X → ℂ) → (X → ℂ)} (hadd : ∀ f f', P f → P f' → P (f + f')) (hsm : ∀ (a : ℂ) f, P f → P (a • f)) (hA : ∀ i f, P f → P (A i f)) (Badd : ∀ f f', P f → P f' → B (f + f') = B f + B f') (Bsm : ∀ (a : ℂ) f, P f → B (a • f) = a • B f) (hBA : ∀ i f, P f → B (A i f) = A i (B f)) {f : X → ℂ} (hf : P f) (Q : MvPolynomial ι ℂ) : B (pureTensorL A f Q) = pureTensorL A (B f) Q

theorem **pure_fun_add** (hA : ∀ i f, P f → P (A i f)) (hadd : ∀ f f', P f → P f' → P (f + f')) (hsm : ∀ (a : ℂ) f, P f → P (a • f)) (Aadd : ∀ i f f', P f → P f' → A i (f + f') = A i f + A i f') {f f' : X → ℂ} (hf : P f) (hf' : P f') (Q : MvPolynomial ι ℂ) : pureTensorL A (f + f') Q = pureTensorL A f Q + pureTensorL A f' Q

theorem **pure_fun_smul** (hA : ∀ i f, P f → P (A i f)) (Asm : ∀ i (a : ℂ) f, P f → A i (a • f) = a • A i f) (a : ℂ) {f : X → ℂ} (hf : P f) (Q : MvPolynomial ι ℂ) : pureTensorL A (a • f) Q = a • pureTensorL A f Q

theorem **dRho_comm_ne** (z z' : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Datum.dRho (heisO ψ S T) θ τ z (Datum.dRho (heisO ψ S T) θ τ' z' f) = Datum.dRho (heisO ψ S T) θ τ' z' (Datum.dRho (heisO ψ S T) θ τ z f)

theorem **dRho_aPlus_ne** (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (J : Matrix (Fin T.rank) (Fin S.rank) ℂ → Matrix (Fin T.rank) (Fin S.rank) ℂ) (u : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Datum.dRho (heisO ψ S T) θ τ z (aPlus (heisO ψ S T) θ τ' J u f) = aPlus (heisO ψ S T) θ τ' J u (Datum.dRho (heisO ψ S T) θ τ z f)

theorem **aPlus_comm_ne** (J J' : Matrix (Fin T.rank) (Fin S.rank) ℂ → Matrix (Fin T.rank) (Fin S.rank) ℂ) (z u : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : aPlus (heisO ψ S T) θ τ J z (aPlus (heisO ψ S T) θ τ' J' u f) = aPlus (heisO ψ S T) θ τ' J' u (aPlus (heisO ψ S T) θ τ J z f)

theorem **annih_aPlus_ne** (J J' : Matrix (Fin T.rank) (Fin S.rank) ℂ → Matrix (Fin T.rank) (Fin S.rank) ℂ) (z u : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : annih (heisO ψ S T) θ τ J z (aPlus (heisO ψ S T) θ τ' J' u f) = aPlus (heisO ψ S T) θ τ' J' u (annih (heisO ψ S T) θ τ J z f)

theorem **om_dRho_fix** (z : Matrix (Fin W.rank) (Fin V.rank) ℂ) {f : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) : ω (atTau θ V τ' g, 1) (Datum.dRho (heisO ψ V W) θ τ z f) = Datum.dRho (heisO ψ V W) θ τ z (ω (atTau θ V τ' g, 1) f)

theorem **om_aPlus_fix** (J : Matrix (Fin W.rank) (Fin V.rank) ℂ → Matrix (Fin W.rank) (Fin V.rank) ℂ) (u : Matrix (Fin W.rank) (Fin V.rank) ℂ) {f : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) : ω (atTau θ V τ' g, 1) (aPlus (heisO ψ V W) θ τ J u f) = aPlus (heisO ψ V W) θ τ J u (ω (atTau θ V τ' g, 1) f)

theorem **om_annih_fix** (J : Matrix (Fin W.rank) (Fin V.rank) ℂ → Matrix (Fin W.rank) (Fin V.rank) ℂ) (u : Matrix (Fin W.rank) (Fin V.rank) ℂ) {f : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) : ω (atTau θ V τ' g, 1) (annih (heisO ψ V W) θ τ J u f) = annih (heisO ψ V W) θ τ J u (ω (atTau θ V τ' g, 1) f)

theorem **twoPlace_comm** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (χS χT : (AE D)ˣ →* Circle) (ω : (GA D V × GA D W →* Module.End ℂ (WO V W → ℂ))), IsInfFam θ → IsWeilDatumO ψ V W χS χT ω → ∀ (τ τ' : D.what₂.Tau), τ ≠ τ' → ∀ F, IsThetaFunO ψ V W F → let pt

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockOperatorsDistinctRealPlacesCommuteThetaStatement

theorem **_root_.GlobalTheory.T.fockOperators_distinct_real_places_commute_theta** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (χS χT : (AE D)ˣ →* Circle) (ω : (GA D V × GA D W →* Module.End ℂ (WO V W → ℂ))), IsInfFam θ → IsWeilDatumO ψ V W χS χT ω → ∀ (τ τ' : D.what₂.Tau), τ ≠ τ' → ∀ F, IsThetaFunO ψ V W F → let pt

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockPartitionRowsNumericDatumMemIcc

theorem **one_le_scL** (L : SizeData₂ D) : 1 ≤ L.sc.L

theorem **one_le_b** (L : SizeData₂ D) : 1 ≤ L.sc.b

theorem **b'_pos** (L : SizeData₂ D) : 0 < L.sc.b'

theorem **one_le_n** (D : D7) : (1 : ℝ) ≤ D.R.n

theorem **exists_top** {τ : D.what₂.Tau} (h : (D.eTau τ : D.R.GG) ∈ D.host.topPlaces) : ∃ t : D.host.TT, (D.eTau τ : D.R.GG) = D.R.tau1 t

theorem **uW_numRec** (large₂ : SizeData₂ D) (hD : ∃ n : ℕ, D.s.Del = n) (u : D.what₂.Word) (z : RPl D.R) (i : Fin D.R.k) : (numDatumOfParameters D large₂).uW (codeRec large₂ u) z i = (D.hcW u (tauPl D z)).u i

theorem **nu_numRec** (large₂ : SizeData₂ D) (hD : ∃ n : ℕ, D.s.Del = n) (u : D.what₂.Word) (j : ℕ) (z : RPl D.R) (i : ℕ) : (((numDatumOfParameters D large₂).nu (codeRec large₂ u) j z i : ℚ) : ℝ) = DV.nuRow D u j (tauPl D z) i

theorem **nuRow_zero_le_three_mul** (large₂ : SizeData₂ D) (P : ParameterChoice' D large₂) (hb : ∀ t, (large₂.bT t : ℝ) ≤ 2 * large₂.sc.b') (hT : ThickUp large₂) (u : D.what₂.Word) (i : ℕ) (hi : i < D.R.m) (τ : D.what₂.Tau) : DV.nuRow D u (i + 1) τ 0 ≤ 3 * large₂.sc.a

theorem **nu_numRec_mem** : ∀ (D : D7) (L : SizeData₂ D) (CY : ℝ), HtRowsS L CY → ∀ (u : D.what₂.Word), ∀ j ∈ (numDatumOfParameters D L).lifts, ∀ z, ∀ i < D.R.N (j - 1), 0 ≤ (numDatumOfParameters D L).nu (codeRec L u) j z i ∧ (((numDatumOfParameters D L).nu (codeRec L u) j z i : ℚ) : ℝ) ≤ 3 * L.sc.a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockPartitionRowsNumericDatumMemIccStatement

theorem **_root_.GlobalTheory.T.fockPartitionRows_numericDatum_mem_Icc** : ∀ (D : D7) (L : SizeData₂ D) (CY : ℝ), HtRowsS L CY → ∀ (u : D.what₂.Word), ∀ j ∈ (numDatumOfParameters D L).lifts, ∀ z, ∀ i < D.R.N (j − 1), 0 ≤ (numDatumOfParameters D L).nu (codeRec L u) j z i ∧ (((numDatumOfParameters D L).nu (codeRec L u) j z i : ℚ) : ℝ) ≤ 3 ∗ L.sc.a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockPolynomialJetValueEqJetLinearFormRescaled

theorem **mvZ_scal_monomial** {ι : Type} {a b : ℕ} (r : ι → ℂ) (k : ι × Fin b × Fin a →₀ ℕ) (c : ℂ) : mvZ (fun _ => (1 : Matrix (Fin a) (Fin a) ℂ)) (fun τ => r τ • (1 : Matrix (Fin b) (Fin b) ℂ)) (MvPolynomial.monomial k c) = MvPolynomial.monomial k (c ∗ k.prod fun v n => r v.1 ^ n)

theorem **jetP_eq_jetL** : ∀ (D : D7) (S T : HSp D) (s : ℂ) (cD : D.what₂.Tau → ℂ), (∀ τ, cD τ ≠ 0) → ∀ (G : VCx D.what₂.Tau S.rank T.rank → ℂ) (P : FP D.what₂.Tau S.rank T.rank), jetP S T rfl rfl s G P = jetL cD G (mvZ (fun _ => 1) (fun τ => (s / cD τ) • 1) P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockPolynomialJetValueEqJetLinearFormRescaledStatement

theorem **_root_.GlobalTheory.T.fockPolynomialJetValue_eq_jetLinearForm_rescaled** : ∀ (D : D7) (S T : HSp D) (s : ℂ) (cD : D.what₂.Tau → ℂ), (∀ τ, cD τ ≠ 0) → ∀ (G : VCx D.what₂.Tau S.rank T.rank → ℂ) (P : FP D.what₂.Tau S.rank T.rank), jetP S T rfl rfl s G P = jetL cD G (mvZ (fun _ => 1) (fun τ => (s / cD τ) • 1) P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockPolynomialSchwartzKilledByLongAnnihilation

theorem **hasDerivAt_eval_line** {σ : Type} [Fintype σ] [DecidableEq σ] (p : MvPolynomial σ ℂ) (v c : σ → ℂ) (t : ℝ) : HasDerivAt (fun t : ℝ => eval (v + (t : ℂ) • c) p) (∑ k, c k ∗ eval (v + (t : ℂ) • c) (pderiv k p)) t

theorem **hasDerivAt_gauss** (hM : ∀ τ, (gramAt D V τ ∗ s τ).IsHermitian) (z : D.what₂.Tau → Matrix (Fin V.rank) (Fin n) ℂ) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin n) ℂ) : HasDerivAt (fun t : ℝ => gauss V l s (z + (t : ℂ) • dirAt τ X)) (−(linF V l s τ X (z τ)) ∗ gauss V l s z) 0

theorem **compAt_add'** (τ : D.what₂.Tau) {m : ℕ} (x y : Mx D m n) : compAt θ τ (x + y) = compAt θ τ x + compAt θ τ y

theorem **compAt_add_lineAt** (hθ : IsInfFam θ) {m : ℕ} (x : Mx D m n) (τ : D.what₂.Tau) (X : Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : (fun τ' => compAt θ τ' (x + lineAt θ τ X t)) = (fun τ' => compAt θ τ' x) + (t : ℂ) • dirAt τ X

theorem **finPart_add_lineAt** (hθ : IsInfFam θ) {m : ℕ} (x : Mx D m n) (τ : D.what₂.Tau) (X : Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : (fun i j => finPart D ((x + lineAt θ τ X t) i j)) = fun i j => finPart D (x i j)

theorem **evalPoly_eq** {m : ℕ} (p : MvPolynomial (D.what₂.Tau × Fin m × Fin n × Bool) ℂ) (z : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ) : evalPoly p z = eval (ptOf z) p

theorem **ptOf_add_smul** {m : ℕ} (z Y : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : ptOf (z + (t : ℂ) • Y) = ptOf z + (t : ℂ) • ptOf Y

theorem **hasDerivAt_gen** (hθ : IsInfFam θ) (hM : ∀ τ, (gramAt D V τ ∗ s τ).IsHermitian) (e : D.what₂.Tau × Fin V.rank × Fin n × Bool →₀ ℕ) (g : (Fin V.rank → Fin n → Af D.host.F ⊗[D.host.F] D.host.E) → ℂ) (x : Mx D V.rank n) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin n) ℂ) : HasDerivAt (fun t : ℝ => gen θ V l s e g (x + lineAt θ τ X t)) ((∑ k, ptOf (dirAt τ X) k ∗ eval (ptOf (fun τ' => compAt θ τ' x)) (pderiv k (monomial e 1))) ∗ gauss V l s (fun τ' => compAt θ τ' x) ∗ g (fun i j => finPart D (x i j)) − linF V l s τ X (compAt θ τ x) ∗ gen θ V l s e g x) 0

theorem **annS_gen** (hθ : IsInfFam θ) (hM : ∀ τ, (gramAt D V τ ∗ s τ).IsHermitian) (e : D.what₂.Tau × Fin V.rank × Fin n × Bool →₀ ℕ) (g : (Fin V.rank → Fin n → Af D.host.F ⊗[D.host.F] D.host.E) → ℂ) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin n) ℂ) : annS V l s θ τ X (gen θ V l s e g) = ∑ k, (ptOf (dirAt τ X) k ∗ (e k : ℂ)) • gen θ V l s (e − Finsupp.single k 1) g

theorem **differentiableAt_of_mem** (hθ : IsInfFam θ) (hM : ∀ τ, (gramAt D V τ ∗ s τ).IsHermitian) {N : ℕ} {φ : Mx D V.rank n → ℂ} (hφ : φ ∈ lowDeg θ V n l s N) (x : Mx D V.rank n) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin n) ℂ) : DifferentiableAt ℝ (fun t : ℝ => φ (x + lineAt θ τ X t)) 0

theorem **degree_sub_single_lt** {σ : Type} {N : ℕ} (e : σ →₀ ℕ) (k : σ) (he : Finsupp.degree e < N + 1) (hk : e k ≠ 0) : Finsupp.degree (e − Finsupp.single k 1) < N

theorem **annS_mem** (hθ : IsInfFam θ) (hM : ∀ τ, (gramAt D V τ ∗ s τ).IsHermitian) {N : ℕ} {φ : Mx D V.rank n → ℂ} (hφ : φ ∈ lowDeg θ V n l s (N + 1)) (τ : D.what₂.Tau) (X : Matrix (Fin V.rank) (Fin n) ℂ) : annS V l s θ τ X φ ∈ lowDeg θ V n l s N

theorem **lowDeg_mono** {N M : ℕ} (h : N ≤ M) : lowDeg θ V n l s N ≤ lowDeg θ V n l s M

theorem **lowDeg_zero** : lowDeg θ V n l s 0 = ⊥

theorem **exists_lowDeg** {φ : Mx D V.rank n → ℂ} (hφ : φ ∈ fockA θ V n l s) : ∃ N, φ ∈ lowDeg θ V n l s N

theorem **fockA_killed_by_annS** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (l : D.what₂.Tau → ℝ) (s : RealMat V), (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ φ ∈ fockA θ V n l s, ∃ N : ℕ, ∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), L.length = N → L.foldr (fun q g => annS V l s θ q.1 q.2 g) φ = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockPolynomialSchwartzKilledByLongAnnihilationStatement

theorem **_root_.GlobalTheory.T.fockPolynomialSchwartz_killed_by_longAnnihilation** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (l : D.what₂.Tau → ℝ) (s : RealMat V), (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ φ ∈ fockA θ V n l s, ∃ N : ℕ, ∀ L : List (D.what₂.Tau × Matrix (Fin V.rank) (Fin n) ℂ), L.length = N → L.foldr (fun q g => annS V l s θ q.1 q.2 g) φ = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockSignTransposeExchangesLocalGlActions

lemma **sg_mul_self** (e : Bool) : GlobalTheory.IndefiniteRowSideFullHecke.RaisingLoweringOperators.sgn e ∗ GlobalTheory.IndefiniteRowSideFullHecke.RaisingLoweringOperators.sgn e = 1

lemma **conj_sg** (e : Bool) : starRingEnd ℂ (GlobalTheory.IndefiniteRowSideFullHecke.RaisingLoweringOperators.sgn e) = GlobalTheory.IndefiniteRowSideFullHecke.RaisingLoweringOperators.sgn e

lemma **swSign_mul_self** (n : Fin b × Fin a) : swSign ε ε' n ∗ swSign ε ε' n = 1

lemma **conj_swSign** (n : Fin b × Fin a) : starRingEnd ℂ (swSign ε ε' n) = swSign ε ε' n

lemma **swSign_swap** (n : Fin b × Fin a) : swSign ε' ε n.swap = swSign ε ε' n

lemma **C_swSign_mul_self** {σ : Type∗} (n : Fin b × Fin a) : (C (swSign ε ε' n) : MvPolynomial σ ℂ) ∗ C (swSign ε ε' n) = 1

lemma **C_sg_mul_self** {σ : Type∗} (e : Bool) : (C (GlobalTheory.IndefiniteRowSideFullHecke.RaisingLoweringOperators.sgn e) : MvPolynomial σ ℂ) ∗ C (GlobalTheory.IndefiniteRowSideFullHecke.RaisingLoweringOperators.sgn e) = 1

@[simp] lemma **swapHom_C** (c : ℂ) : swapHom ε ε' (C c) = C (starRingEnd ℂ c)

@[simp] lemma **swapHom_X** (n : Fin b × Fin a) : swapHom ε ε' (X n) = C (swSign ε ε' n) * X n.swap

lemma **swapHom_smul** (c : ℂ) (P : LPol b a) : swapHom ε ε' (c • P) = starRingEnd ℂ c • swapHom ε ε' P

lemma **swapHom_comp** : (swapHom ε' ε).comp (swapHom ε ε') = RingHom.id _

lemma **swapHom_pderiv** (n : Fin b × Fin a) (Q : LPol b a) : swapHom ε ε' (pderiv n Q) = C (swSign ε ε' n) * pderiv n.swap (swapHom ε ε' Q)

lemma **swapHom_rai** (i : Fin b) (j : Fin a) (Q : LPol b a) : swapHom ε ε' (rai ε ε' i j Q) = rai ε' ε j i (swapHom ε ε' Q)

lemma **swapHom_low** (i : Fin b) (j : Fin a) (Q : LPol b a) : swapHom ε ε' (low ε ε' i j Q) = low ε' ε j i (swapHom ε ε' Q)

lemma **swapHom_half** (c : Prop) [Decidable c] (Q : LPol b a) : swapHom ε ε' ((if c then (1 / 2 : ℂ) • (1 : Module.End ℂ (LPol b a)) else 0) Q) = (if c then (1 / 2 : ℂ) • (1 : Module.End ℂ (LPol a b)) else 0) (swapHom ε ε' Q)

lemma **swapHom_termT** (i l : Fin b) (j : Fin a) (Q : LPol b a) : swapHom ε ε' ((rai ε ε' i j * low ε ε' l j + if i = l then (1 / 2 : ℂ) • 1 else 0) Q) = (rai ε' ε j i * low ε' ε j l + if i = l then (1 / 2 : ℂ) • 1 else 0) (swapHom ε ε' Q)

lemma **swapHom_termS** (i : Fin b) (j k : Fin a) (Q : LPol b a) : swapHom ε ε' ((rai ε ε' i j * low ε ε' i k + if j = k then (1 / 2 : ℂ) • 1 else 0) Q) = (rai ε' ε j i * low ε' ε k i + if j = k then (1 / 2 : ℂ) • 1 else 0) (swapHom ε ε' Q)

lemma **swapEquiv_apply** (P : LPol b a) : swapEquiv ε ε' P = swapHom ε ε' P

theorem **locS_locT_exchange** : ∀ (a b : ℕ) (ε : Fin a → Bool) (ε' : Fin b → Bool), ∃ j : LPol b a ≃+ LPol a b, (∀ (c : ℂ) P, j (c • P) = conj c • j P) ∧ ∀ P : LPol b a, (∀ Y : Matrix (Fin b) (Fin b) ℂ, j (locT ε ε' Y P) = locS ε' ε (Y.map (starRingEnd ℂ)) (j P)) ∧ ∀ X : Matrix (Fin a) (Fin a) ℂ, j (locS ε ε' (X.map (starRingEnd ℂ)) P) = locT ε' ε X (j P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockSignTransposeExchangesLocalGlActionsStatement

theorem **_root_.GlobalTheory.T.fock_sign_transpose_exchanges_local_gl_actions** : ∀ (a b : ℕ) (ε : Fin a → Bool) (ε' : Fin b → Bool), ∃ j : LPol b a ≃+ LPol a b, (∀ (c : ℂ) P, j (c • P) = conj c • j P) ∧ ∀ P : LPol b a, (∀ Y : Matrix (Fin b) (Fin b) ℂ, j (locT ε ε' Y P) = locS ε' ε (Y.map (starRingEnd ℂ)) (j P)) ∧ ∀ X : Matrix (Fin a) (Fin a) ℂ, j (locS ε ε' (X.map (starRingEnd ℂ)) P) = locT ε' ε X (j P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockSubstitutionIdentities

theorem **sum_three_fintype_reverse** {α β γ M : Type} [Fintype α] [Fintype β] [Fintype γ] [AddCommMonoid M] (f : α → β → γ → M) : ∑ x, ∑ y, ∑ z, f x y z = ∑ z, ∑ y, ∑ x, f x y z

theorem **sum_four_fintype_permute** {α β γ δ M : Type} [Fintype α] [Fintype β] [Fintype γ] [Fintype δ] [AddCommMonoid M] (f : α → β → γ → δ → M) : ∑ k, ∑ j, ∑ i, ∑ l, f k j i l = ∑ i, ∑ l, ∑ j, ∑ k, f k j i l

theorem **mvZ_subY_comp** (hab : a ≤ b) (g : ι → Matrix (Fin a) (Fin a) ℂ) (h y' : ι → Matrix (Fin b) (Fin b) ℂ) : (mvZ g h).comp (subY hab y') = (subY hab (fun τ => h τ * y' τ)).comp (lft g)

theorem **mvZ_mvZ_comp** (g g' : ι → Matrix (Fin a) (Fin a) ℂ) (h h' : ι → Matrix (Fin b) (Fin b) ℂ) : (mvZ g h).comp (mvZ g' h') = mvZ (fun τ => g' τ * g τ) (fun τ => h τ * h' τ)

theorem **lft_lft_comp** (g g' : ι → Matrix (Fin a) (Fin a) ℂ) : (lft g).comp (lft g') = lft (fun τ => g' τ * g τ)

theorem **mvZ_one** : mvZ (ι

theorem **lft_one** : lft (ι

theorem **subY_smul** (hab : a ≤ b) (t : ι → ℂ) (y' : ι → Matrix (Fin b) (Fin b) ℂ) : subY hab (fun τ => t τ • y' τ) = (scaleFP t).comp (subY hab y')

theorem **scaleVariables_monomial** (t : ι → ℂ) (m : (ι × Fin b × Fin a) →₀ ℕ) : scaleFP t (monomial m (1 : ℂ)) = (m.prod fun k n => t k.1 ^ n) • monomial m (1 : ℂ)

theorem **jetL_monomial_one** (c : ι → ℂ) (G : VCx ι a b → ℂ) (m : (ι × Fin b × Fin a) →₀ ℕ) : jetL c G (monomial m (1 : ℂ)) = jetC c G m

theorem **jetL_comp_scale** (c t : ι → ℂ) (G : VCx ι a b → ℂ) : (jetL c G).comp (scaleFP (a

theorem **evalAt_frDet** (y : ι → Matrix (Fin a) (Fin a) ℂ) (τ : ι) : evalAt y (frDet ℂ a τ : PY ι a) = (y τ).det

theorem **evalAt_mul_prod** (y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : evalAt y (F * ∏ τ, (frDet ℂ a τ : PY ι a)) = evalAt y F * ∏ τ, (y τ).det

theorem **frDet_prod_ne_zero** : (∏ τ, (frDet ℂ a τ : PY ι a)) ≠ 0

theorem **eq_zero_of_invertible** (F : PY ι a) (hF : ∀ y, IsInv y → evalAt y F = 0) : F = 0

theorem **evalAt_lft_eq** (g y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : evalAt y (lft g F) = evalAt (fun τ => g τ * y τ) F

theorem **substIdentities** : ∀ (ι : Type) [Fintype ι] (a b : ℕ) (hab : a ≤ b), (∀ (g : ι → Matrix (Fin a) (Fin a) ℂ) (h y' : ι → Matrix (Fin b) (Fin b) ℂ) (F : PY ι a), mvZ g h (subY hab y' F) = subY hab (fun τ => h τ * y' τ) (lft g F)) ∧ (∀ (g g' : ι → Matrix (Fin a) (Fin a) ℂ) (h h' : ι → Matrix (Fin b) (Fin b) ℂ) (P : FP ι a b), mvZ g h (mvZ g' h' P) = mvZ (fun τ => g' τ * g τ) (fun τ => h τ * h' τ) P) ∧ (∀ P : FP ι a b, mvZ (fun _ => 1) (fun _ => 1) P = P) ∧ (∀ (g g' : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a), lft g (lft g' F) = lft (fun τ => g' τ * g τ) F) ∧ (∀ F : PY ι a, lft (fun _ => 1) F = F) ∧ (∀ (g y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a), evalAt y (lft g F) = evalAt (fun τ => g τ * y τ) F) ∧ (∀ F : PY ι a, (∀ y, IsInv y → evalAt y F = 0) → F = 0) ∧ ∀ (c t : ι → ℂ) (G : VCx ι a b → ℂ) (y' : ι → Matrix (Fin b) (Fin b) ℂ) (F : PY ι a), jetL c G (subY hab (fun τ => t τ • y'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockSubstitutionIdentitiesStatement

theorem **_root_.GlobalTheory.T.fockSubstitution_identities** : ∀ (ι : Type) [Fintype ι] (a b : ℕ) (hab : a ≤ b), (∀ (g : ι → Matrix (Fin a) (Fin a) ℂ) (h y' : ι → Matrix (Fin b) (Fin b) ℂ) (F : PY ι a), mvZ g h (subY hab y' F) = subY hab (fun τ => h τ * y' τ) (lft g F)) ∧ (∀ (g g' : ι → Matrix (Fin a) (Fin a) ℂ) (h h' : ι → Matrix (Fin b) (Fin b) ℂ) (P : FP ι a b), mvZ g h (mvZ g' h' P) = mvZ (fun τ => g' τ * g τ) (fun τ => h τ * h' τ) P) ∧ (∀ P : FP ι a b, mvZ (fun _ => 1) (fun _ => 1) P = P) ∧ (∀ (g g' : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a), lft g (lft g' F) = lft (fun τ => g' τ * g τ) F) ∧ (∀ F : PY ι a, lft (fun _ => 1) F = F) ∧ (∀ (g y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a), evalAt y (lft g F) = evalAt (fun τ => g τ * y τ) F) ∧ (∀ F : PY ι a, (∀ y, IsInv y → evalAt y F = 0) → F = 0) ∧ ∀ (c t : ι → ℂ) (G : VCx ι a b → ℂ) (y' : ι → Matrix (Fin b) (Fin b) ℂ) (F : PY ι a), jetL

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorFiniteUnderCompactCommutingCartan

theorem **isKFinVec_of_isFockVec** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : WRep S T) (sS : RealMat S) (sT : RealMat T), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀

τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ K : Subgroup (GA D S), IsCompact (K : Set (GA D S)) → (∀ k ∈ K, ∀ τ, compAt θ τ (Formalism.L2.Theta.mt S k) * sS τ = sS τ * compAt θ τ (Formalism.L2.Theta.mt S k)) → ∀ f, IsFockVec ψ θ S T sS sT f → IsKFinVec om K f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorFiniteUnderCompactCommutingCartanStatement

theorem **_root_.GlobalTheory.T.fock_vector_finite_under_compact_commuting_cartan** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : WRep S T) (sS : RealMat S) (sT : RealMat T), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → (∀ τ, IsCartanInv (gramAt D S τ) (sS τ)) → (∀ τ, IsCartanInv (gramAt D T τ) (sT τ)) → ∀ K : Subgroup (GA D S), IsCompact (K : Set (GA D S)) → (∀ k ∈ K, ∀ τ, compAt θ τ (Formalism.L2.Theta.mt S k) * sS τ = sS τ * compAt θ τ (Formalism.L2.Theta.mt S k)) → ∀ f, IsFockVec ψ θ S T sS sT f → IsKFinVec om K f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorPureTensorCoefficientSplitsTrivialPlac

theorem **ip_double_sum** {D : D7} {S T : HSp D} (μ : Measure (WO S T)) (ℱ : Set (WO S T)) {ι κ : Type} (s : Finset ι) (t : Finset κ) (a : ι → ℂ) (b : κ → ℂ) (u : ι → WO S T → ℂ) (w : κ → WO S T → ℂ) (hI : ∀ i j, IntegrableOn (fun z => u i z * (starRingEnd ℂ) (w j z)) ℱ μ) : ipO μ ℱ (∑ i ∈ s, a i • u i) (∑ j ∈ t, b j • w j) = ∑ i ∈ s, ∑ j ∈ t, a i * (starRingEnd ℂ) (b j) * ipO μ ℱ (u i) (w j)

theorem **monoFactor_supported** {D : D7} {m n : ℕ} (k : FVar D m n →₀ ℕ) (τ : D.what₂.Tau) : monoFactor k τ ∈ MvPolynomial.supported ℂ {y : FVar D m n | y.1 = τ}

theorem **prod_monoFactor** {D : D7} {m n : ℕ} (k : FVar D m n →₀ ℕ) : ∏ τ, monoFactor k τ = MvPolynomial.monomial k 1

theorem **fockPure_any** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ} {sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ} {ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} {f₀ : WO S T → ℂ} {μ : Measure (WO S T)} {ℱ : Set (WO S T)} (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (hb : IsBase ψ θ S T sS sT ΛS ΛT f₀) (hμ : IsCovolOne (ratL T S) μ ℱ) (x : Place D → WO S T → ℂ) (f : WO S T → ℂ) (hx : IsPure ψ θ S T sS sT ΛS ΛT μ ℱ f₀ x f) (hx0 : ∀ τ, x (.inr τ) = f₀) (P : D.what₂.Tau → MvPolynomial (FVar D T.rank S.rank) ℂ) (hP : ∀ τ, P τ ∈ MvPolynomial.supported ℂ {y : FVar D T.rank S.rank | y.1 = τ}) : IsPure ψ θ S T sS sT ΛS ΛT μ ℱ f₀ (Sum.elim (fun v => x (.inl v)) fun τ => pureTensorO ψ θ

theorem **pureTensorO_eq_sum** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (f : WO S T → ℂ) (P : MvPolynomial (FVar D T.rank S.rank) ℂ) : pureTensorO ψ θ S T sS sT BS BT f P = ∑ k ∈ P.support, P.coeff k • pureTensorO ψ θ S T sS sT BS BT f (MvPolynomial.monomial k 1)

theorem **coeff_split** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : OmT S T) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (f₀ : WO S T → ℂ) (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → IsBase ψ θ S T sS sT ΛS ΛT f₀ → IsCovolOne (ratL T S) μ ℱ → ∀ (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (x : Place D → WO S T → ℂ), (∀ τ, x (.inr τ) = f₀) → ∀ (v : FinPl D) (P : MvPolynomial (FVar D T.rank S.rank) ℂ) (g : GA D S), gAt S (.inl v) g = 1 → ∃ C : ℂ, ∀ (x₁ x₂ : Place D → WO S T → ℂ) (f₁ f₂ : WO S T → ℂ), (∀ p, p ≠ .inl v → x₁ p = x p) → (∀ p, p ≠ .inl v → x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorPureTensorCoefficientSplitsTrivialPlacStatement

theorem **_root_.GlobalTheory.T.fockVector_pureTensor_coefficient_splits_trivial_place** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : OmT S T) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (f₀ : WO S T → ℂ) (μ : Measure (WO S T)) (ℱ : Set (WO S T)), IsWeilDatumO ψ S T χS χT om → IsInfFam θ → IsBase ψ θ S T sS sT ΛS ΛT f₀ → IsCovolOne (ratL T S) μ ℱ → ∀ (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (x : Place D → WO S T → ℂ), (∀ τ, x (.inr τ) = f₀) → ∀ (v : FinPl D) (P : MvPolynomial (FVar D T.rank S.rank) ℂ) (g : GA D S), gAt S (.inl v) g = 1 → ∃ C : ℂ, ∀ (x₁ x₂ : Place D → WO S T → ℂ) (f₁ f₂ : W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorSchrodingerModelIsKFinite

theorem **fock_kfinite** : ∀ (D : D7) (n : ℕ) (V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (Mx D V.rank n)) (ℱ : Set (Mx D V.rank n)), IsCovolOne (ratMx D V.rank n) μ ℱ → ∀ ω : G D n →* Module.End ℂ (schwartzA θ V.rank n), IsWeilSch ψ χ θ V μ ω → ∀ l : D.what₂.Tau → ℝ, HasRealPar ψ θ l → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ Φ : schwartzA θ V.rank n, Φ.1 ∈ fockA θ V n l s → FiniteDimensional ℂ (Submodule.span ℂ {Ψ : schwartzA θ V.rank n | ∃ k ∈ K D n, Ψ = ω k Φ})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorSchrodingerModelIsKFiniteStatement

theorem **_root_.GlobalTheory.T.fockVector_schrodingerModel_isKFinite** : ∀ (D : D7) (n : ℕ) (V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (Mx D V.rank n)) (ℱ : Set (Mx D V.rank n)), IsCovolOne (ratMx D V.rank n) μ ℱ → ∀ ω : G D n →* Module.End ℂ (schwartzA θ V.rank n), IsWeilSch ψ χ θ V μ ω → ∀ l : D.what₂.Tau → ℝ, HasRealPar ψ θ l → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ Φ : schwartzA θ V.rank n, Φ.1 ∈ fockA θ V n l s → FiniteDimensional ℂ (Submodule.span ℂ {Ψ : schwartzA θ V.rank n | ∃ k ∈ K D n, Ψ = ω k Φ})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorsLatticeVacuaOffFiniteSet

theorem **exists_under** (v : HeightOneSpectrum (𝒪 D.host.F)) : ∃ P : HeightOneSpectrum (𝒪 D.host.E), P.under (𝒪 D.host.F) = v

theorem **latO_at_le** {S T : HSp D} {LS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {LT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} {NS NT : ℕ} (hS : ∀ x ∈ L0c D S, (NS : D.host.E) • x ∈ LS) (hT : ∀ x ∈ L0c D T, (NT : D.host.E) • x ∈ LT) {v : HeightOneSpectrum (𝒪 D.host.F)} (hv : ((NS * NT : ℕ) : 𝒪 D.host.F) ∉ v.asIdeal) : latO S T (frameLat S : Set (Fin S.rank → D.host.E)) (frameLat T : Set (Fin T.rank → D.host.E)) (fun u => u = v) ≤ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **GlobalTheory.ChainOnto.fock_vacFin_off** : ∀ D : D7, ForStep D fun c _ s s' p => ∀ φ, ∀ v ∉ c.S, VacFin c.ψ s.V s'.V v (p.pt φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FockVectorsLatticeVacuaOffFiniteSetStatement

theorem **_root_.GlobalTheory.T.fock_vectors_lattice_vacua_off_finite_set** : ∀ D : D7, ForStep D fun c _ s s' p => ∀ φ, ∀ v ∉ c.S, VacFin c.ψ s.V s'.V v (p.pt φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FoldIntegralBesselQuotientSupportedFunctions

theorem **fold_quot** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (𝒮 : Set (besGrp B.d B.o)), IsFundamentalDomain ((GQ (D.sp.Y 0)).subgroupOf (besGrp B.d B.o)) 𝒮 B.lev.μR → ∀ Fs : Set X.B.I.GA, IsFundamentalDomain (Srat X u) Fs (X.B.I.muA X.B.μf) → ∃ c : ℝ, 0 < c ∧ ∀ F : X.B.I.GA → ENNReal, Measurable F → (∀ γ ∈ Srat X u, ∀ x, F (γ * x) = F x) → (∀ k ∈ X.K2, ∀ x, F (x * (1, k)) = F x) → (∀ x, F x ≠ 0 → ∃ s ∈ SfinU X u, ∃ k ∈ X.K2, x.2 = s * X.xu u * k) → ∫⁻ x in Fs, F x ∂(X.B.I.muA X.B.μf) = ENNReal.ofReal c * ∫⁻ y, ∫⁻ s in 𝒮, F (iotaX X h s.1 * (y, X.xu u)) ∂B.lev.μR ∂(X.nuQ u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FoldIntegralBesselQuotientSupportedFunctionsStatement

theorem **_root_.GlobalTheory.T.fold_integral_besselQuotient_supportedFunctions** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (𝒮 : Set (besGrp B.d B.o)), IsFundamentalDomain ((GQ (D.sp.Y 0)).subgroupOf (besGrp B.d B.o)) 𝒮 B.lev.μR → ∀ Fs : Set X.B.I.GA, IsFundamentalDomain (Srat X u) Fs (X.B.I.muA X.B.μf) → ∃ c : ℝ, 0 < c ∧ ∀ F : X.B.I.GA → ENNReal, Measurable F → (∀ γ ∈ Srat X u, ∀ x, F (γ * x) = F x) → (∀ k ∈ X.K2, ∀ x, F (x * (1, k)) = F x) → (∀ x, F x ≠ 0 → ∃ s ∈ SfinU X u, ∃ k ∈ X.K2, x.2 = s * X.xu u * k) → ∫⁻ x in Fs, F x ∂(X.B.I.muA X.B.μf) = ENNReal.ofReal c * ∫⁻ y, ∫⁻ s in 𝒮, F (iotaX X h s.1 * (y, X.xu u)) ∂B.lev.μR ∂(X.nuQ u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FoldrExtract

theorem **foldr_extract** {ι X : Type} [DecidableEq ι] (g : ι → X → X) (i : ι) (hc : ∀ a, a ≠ i → ∀ x, g a (g i x) = g i (g a x)) : ∀ (l : List ι) (x : X), i ∈ l → l.foldr g x = g i ((l.erase i).foldr g x)

theorem **latO_mono** {S T : HSp D} (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) {P P' : Pl D → Prop} (h : ∀ p, P p → P' p) : latO S T LS LT P ≤ latO S T LS LT P'

theorem **vac_mono** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P P' : Pl D → Prop} (h : ∀ p, P p → P' p) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P' f) : Vac ψ S T LS LT P f

theorem **eg_combAt** {c : Comm D} {v : VCom c} (G : GLoc v) (j : ℕ) (hj : j ≤ G.s) : CombAt (· = G.q.v) (G.eg j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FoldrSplitPlaceSecondCombinationsVacuumNeZero

theorem **isThetaFunO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) : IsThetaFunO ψ S T 0 where

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (u : WO S T) : heisO ψ S T u 0 = 0

theorem **isModel_zero** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) : p.IsModel ψ FS FT (0 : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)) where

theorem **model_ne_zero** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {FS : Frame k S.rank Y'} {FT : Frame k T.rank Y} {Tm} (hTm : p.IsModel ψ FS FT Tm) {Φ₀ : SB (Matrix Y Y' k)} (h₀ : Tm Φ₀ ≠ 0) {O : SB (Matrix Y Y' k)} (hO : O ≠ 0) : Tm O ≠ 0

theorem **exists_adapted_frame** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S : HSp D) (M : GlobalTheory.FinPl.Lat S), IsSelfDualC S M → ∀ (Y : Type) [Fintype Y] [DecidableEq Y] (F : Frame k S.rank Y), ∃ G : Frame k S.rank Y, ∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((G.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S M (· = q.v)

theorem **down_up** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x : Matrix Y Y' k) : p.down FS FT (p.up FS FT x) = x

theorem **exists_model_vac** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ G, GlobalTheory.FinPl.Vac ψ S T M M' (· = q.v) G → ∃ Tm, (ofSplit q).IsModel ψ FS FT Tm ∧ ∀ Φ₀ : SB (Matrix Y Y' k), Φ₀.1 = Set.indicator {x | (ofSplit q).up FS FT x ∈ latO S T (M : Set (Fin S.rank → D.host.E)) (M' : Set (Fin T.rank → D.host.E)) (· = q.v)} (fun _ => 1) → Tm Φ₀ = G

theorem **comb_o_ne_zero** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hM : IsSelfDualC S M) (hM' : IsSelfDualC T M') (L : LocPl ψ S T M M') {G : WO S T → ℂ} (hG : GlobalTheory.FinPl.Vac ψ S T M M' (· = L.q.v) G) (hG0 : G ≠ 0) : comb ψ S T L.o G ≠ 0

theorem **combAt_o** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (L : LocPl ψ S T M M') : GlobalTheory.FinPl.CombAt (· = L.q.v) L.o

theorem **foldr_comb_ne_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → GlobalTheory.SeeSaw.LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ F, GlobalTheory.FinPl.Vac ψ S T M M' (fun _ => True) F → F ≠ 0 → (List.finRange n).foldr (fun i g => comb ψ S T (L i).o g) F ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FoldrSplitPlaceSecondCombinationsVacuumNeZeroStatement

theorem **_root_.GlobalTheory.T.foldr_splitPlaceSecondCombinations_vacuum_ne_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → GlobalTheory.SeeSaw.LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ F, GlobalTheory.FinPl.Vac ψ S T M M' (fun _ => True) F → F ≠ 0 → (List.finRange n).foldr (fun i g => comb ψ S T (L i).o g) F ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormMapRBijective

theorem **clMapR_cl** (hΛ : ∀ b, Λ b = Λ₀) (b : β) (p : (stepY ST y β Λ 0 pref).D b) : clMapR (Λ₀

theorem **clMapR_bijective** [Unique β] (hy : ∀ N, y.ys N = {∅}) (hΛ : ∀ b, Λ b = Λ₀) (h0 : ∀ b, 0 b = LevQFam.top X.E X.K n Q) : Function.Bijective (clMapR (Λ₀

theorem **formMapR_bijective** [Unique β] (hy : ∀ N, y.ys N = {∅}) (hΛ : ∀ b, Λ b = Λ₀) (h0 : ∀ b, 0 b = LevQFam.top X.E X.K n Q) (h : (((stepY ST y β Λ 0 pref).reRep pr).fam X.emb lam).Ok) (h₀ : (base ST lam Λ₀).Ok) : Function.Bijective (formMapR pref pr lam hΛ h0 h)

theorem **exists_colZeroR** {X : Fields} {Pr : Primes X} {R : Ranks} (W : Wts X Pr.Word R) {Ch : Chars X Pr R.m} {sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)} {S : Set ℕ} {mu : Tup X.F X.E X.K S} {cst : LocCst X (rowSG Pr sel mu)} {L : ChIdx X Pr R (QE sel

(mu : Chr X.F X.E X.K))} {G : ChGlob X R.m} {m2 : ℕ} (Tp : TopLev X Pr R (QE sel (mu : Chr X.F X.E X.K))) (u : Pr.Word) (pr : ∀ b, (stepAt sel mu cst L Tp u 0).D b → Prop) (hX : X.Ok) (hL : L.Ok Ch sel mu cst G m2) (h : (((stepAt sel mu cst L Tp u 0).reRep pr).fam X.emb (lamT (W.hc u) 0)).Ok) : ∃ κ : MfV W Ch u ≃ₗ[X.K] ((stepAt sel mu cst L Tp u 0).reRep pr).FormsK X.emb (lamT (W.hc u) 0) h, ∀ φ c, (κ φ).1 c = φ.1 (((stepAt sel mu cst L Tp u 0).reRep pr).rep c).1.1.deco

theorem **exists_colEquiv** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (hX : B.fields.Ok) (l : I.Laws) (u : B.Word) (a : I.ι) : ∃ κ : ColsK I u ≃ₗ[B.fields.K] ((I.fm l u a).lat 0).V, ∀ φ, IsCol I u a l (κ φ) φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormTranslateRealPointReadByPeriod

theorem **infM_eq_reindex** {D : D7} (θ : InfFam D) (V : HSp D) {n : ℕ} (hn : V.rank = n) (g : GA D V) (τ : D.what₂.Tau) : infM θ V hn g τ = (infMat D.sp.cm.sig V.gram (θ τ) g).reindex (finCongr hn) (finCongr hn)

theorem **infPt_eq_infM** {D : D7} (θ : InfFam D) (x : GA D D.sp.X') (τ : D.what₂.Tau) : ((infPt θ x τ : GL _ ℂ) : Matrix _ _ ℂ) = infM θ D.sp.X' (rankP D) x τ

theorem **infPt_mul_inv** {D : D7} (θ : InfFam D) (x : GA D D.sp.X') (τ : D.what₂.Tau) : (infPt θ x τ).1 * (infPt θ x⁻¹ τ).1 = 1

theorem **top_translate** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormTranslateRealPointReadByPeriodStatement

theorem **_root_.GlobalTheory.T.form_translate_real_point_read_by_period** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormalGroupHeightLeTwoGoodReduction

theorem **analyticAt_shiftFn** (L : PeriodPair) (c : ℂ) : AnalyticAt ℂ (shiftFn L c) 0

theorem **taylor0_shiftFn** (L : PeriodPair) (c : ℂ) : taylor0 (shiftFn L c) = wpShift (pSeries L) c

theorem **shiftFn_eq** (L : PeriodPair) (c z : ℂ) (hz : z ≠ 0) : shiftFn L c z = z ^ 2 * (L.weierstrassP z - c)

theorem **taylor0_prod_shiftFn** (L : PeriodPair) (S : Finset ℂ) (a : ℂ → ℂ) : AnalyticAt ℂ (fun z => ∏ P ∈ S, shiftFn L (a P) z) 0 ∧ taylor0 (fun z => ∏ P ∈ S, shiftFn L (a P) z) = ∏ P ∈ S, wpShift (pSeries L) (a P)

theorem **weierstrassP_nat_mul_product_series_identity** (L : PeriodPair) (n e : ℂ) (hn : n ≠ 0) (S : Finset ℂ) (a b : ℂ → ℂ) (h : ∀ z, n * z ∉ L.lattice → n ^ 2 * (L.weierstrassP (n * z) - e) * ∏ P ∈ S, (L.weierstrassP z - a P) = (L.weierstrassP z - e) * ∏ P ∈ S, (L.weierstrassP z - b P)) : rescale n (wpShift (pSeries L) e) * ∏ P ∈ S, wpShift (pSeries L) (a P) = wpShift (pSeries L) e * ∏ P ∈ S, wpShift (pSeries L) (b P)

theorem **map_wpShift** {K' : Type} [Field K'] (φ : K →+* K') (p : K⟦X⟧) (c : K) : PowerSeries.map φ (wpShift p c) = wpShift (PowerSeries.map φ p) (φ c)

theorem **map_rescale'** {K' : Type} [Field K'] (φ : K →+* K') (p : K⟦X⟧) (c : K) : PowerSeries.map φ (rescale c p) = rescale (φ c) (PowerSeries.map φ p)

theorem **series_u** {σ : Type} {p l linv A : K⟦X⟧} (hL : IsLogPair l linv) (hX : XIsWp p l A 0) (n e : K) (S : Finset σ) (a b : σ → K) (h : rescale n (wpShift p e) * ∏ P ∈ S, wpShift p (a P) = wpShift p e * ∏ P ∈ S, wpShift p (b P)) : (A.subst (mulBy l linv n) - C e * mulBy l linv n ^ 2) * (C (n ^ 2) * ∏ P ∈ S, (A - C (a P) * X ^ 2)) * X ^ 2 = mulBy l linv n ^ 2 * ((A - C e * X ^ 2) * ∏ P ∈ S, (A - C (b P) * X ^ 2))

theorem **homog_field** {F : Type} [Field F] (φ : K →+* F) (q : Polynomial K) (m : ℕ) (hq : q.natDegree ≤ m) (a y : F) (hy : y ≠ 0) : ∑ i ∈ Finset.range (m + 1), φ (q.coeff i) * a ^ i * y ^ (m - i) = y ^ m * q.eval₂ φ (a / y)

theorem **homog_prod** {σ : Type} (S : Finset σ) (d : K) (c : σ → K) (A : K⟦X⟧) (m : ℕ) (hm : S.card = m) : ∑ i ∈ Finset.range (m + 1), C ((Polynomial.C d * ∏ P ∈ S, (Polynomial.X - Polynomial.C (c P))).coeff i) * A ^ i * X ^ (2 * (m - i)) = C d * ∏ P ∈ S, (A - C (c P) * X ^ 2)

theorem **coeff_prod_le** {K : Type} [NormedField K] [IsUltrametricDist K] {σ : Type} (S : Finset σ) (c : σ → K) (hc : ∀ P ∈ S, ‖c P‖ ≤ 1) (i : ℕ) : ‖(∏ P ∈ S, (Polynomial.X - Polynomial.C (c P))).coeff i‖ ≤ 1

theorem **norm_two** (l : ℕ) [Fact l.Prime] (hl : 5 ≤ l) : ‖(2 : ℂ_[l])‖ = 1

theorem **mulBy_eq_zSer** {K : Type} [Field K] [CharZero K] (W : WeierstrassCurve K) (n : ℕ) : mulBy (Formalism.L2.Th1.A.logSer W) (Formalism.L2.Th1.A.expSer W) (n : K) = Formalism.L2.Th1.A.zSer W (n : ℤ)

theorem **odd_mul_sub_mem** (L : PeriodPair) {k : ℕ} (hk : Odd k) {s P : ℂ} (hs : 2 * s ∈ L.lattice) (hP : (k : ℂ) * P ∈ L.lattice) : (k : ℂ) * (s + P) - s ∈ L.lattice

theorem **wp_congr** (L : PeriodPair) {z w : ℂ} (h : z - w ∈ L.lattice) : L.weierstrassP z = L.weierstrassP w

theorem **nat_mul_mem** (L : PeriodPair) (k : ℕ) {z : ℂ} (h : z ∈ L.lattice) : (k : ℂ) * z ∈ L.lattice

theorem **height_le_two** (L : PeriodPair) (l : ℕ) [Fact l.Prime] (hl : 5 ≤ l) (ι : ℂ ≃+* ℂ_[l]) (h2 : ‖ι L.g₂‖ ≤ 1) (h3 : ‖ι L.g₃‖ ≤ 1) (hΔ : ‖ι L.g₂ ^ 3 - 27 * ι L.g₃ ^ 2‖ = 1) : ∃ n, n ≤ l ^ 2 ∧ ‖coeff n (Formalism.L2.Th1.C.mulBy (Formalism.L2.Th1.A.logSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (Formalism.L2.Th1.A.expSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (l : ℂ_[l]))‖ = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormalGroupHeightLeTwoGoodReductionStatement

theorem **_root_.GlobalTheory.T.formalGroup_height_le_two_good_reduction** : ∀ (L : PeriodPair) (l : ℕ) [Fact l.Prime], 5 ≤ l → ∀ (ι : ℂ ≃+* ℂ_[l]), ‖ι L.g₂‖ ≤ 1 → ‖ι L.g₃‖ ≤ 1 → ‖ι L.g₂ ^ 3 - 27 * ι L.g₃ ^ 2‖ = 1 → ∃ n, n ≤ l ^ 2 ∧ ‖coeff n (Formalism.L2.Th1.C.mulBy (Formalism.L2.Th1.A.logSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (Formalism.L2.Th1.A.expSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (l : ℂ_[l]))‖ = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormalGroupMulByIntegralOfKernelPrimeToL

theorem **mulBy_coefLe_of_kernel_prime_to** : ∀ (L : PeriodPair) (l : ℕ) [Fact l.Prime], 5 ≤ l → ∀ (ι : ℂ ≃+* ℂ_[l]), ‖ι L.g₂‖ ≤ 1 → ‖ι L.g₃‖ ≤ 1 → ‖ι L.g₂ ^ 3 - 27 * ι L.g₃ ^ 2‖ = 1 → ∀ (β : ℂ) (N : ℕ), Formalism.L2.Th1.F.MulStable L β → ‖ι β‖ = 1 → 0 < N → ¬ l ∣ N → (∀ z, β * z ∈ L.lattice → (N : ℂ) * z ∈ L.lattice) → CoefLe 1 (Formalism.L2.Th1.C.mulBy (Formalism.L2.Th1.A.logSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (Formalism.L2.Th1.A.expSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (ι β))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FormalGroupMulByIntegralOfKernelPrimeToLStatement

theorem **_root_.GlobalTheory.T.formalGroup_mulBy_integral_of_kernel_prime_to_l** : ∀ (L : PeriodPair) (l : ℕ) [Fact l.Prime], 5 ≤ l → ∀ (ι : ℂ ≃+* ℂ_[l]), ‖ι L.g₂‖ ≤ 1 → ‖ι L.g₃‖ ≤ 1 → ‖ι L.g₂ ^ 3 - 27 * ι L.g₃ ^ 2‖ = 1 → ∀ (β : ℂ) (N : ℕ), Formalism.L2.Th1.F.MulStable L β → ‖ι β‖ = 1 → 0 < N → ¬ l ∣ N → (∀ z, β * z ∈ L.lattice → (N : ℂ) * z ∈ L.lattice) → CoefLe 1 (Formalism.L2.Th1.C.mulBy (Formalism.L2.Th1.A.logSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (Formalism.L2.Th1.A.expSer (Formalism.L2.Th1.A.shortW (ι L.g₂) (ι L.g₃))) (ι β))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselBasePointIsUnfoldingBasePointSpec

theorem **ratV_eLast** (ρ : ℕ) : ratV (eLast D ρ) = ell D ρ

theorem **rat_zlQ** (B : BesLev D σ) : (rat (zlQ B) : Mx D B.L.rank (σ + 1)) = zl B

theorem **rat_gramQ** (W : HSp D) {ρ : ℕ} (x : Matrix (Fin W.rank) (Fin ρ) D.host.E) : (rat (gramQ W x) : Mx D ρ ρ) = adj (rat x) · (gA W · rat x)

theorem **gramQ_eq_of_gramSkew** {W W' : HSp D} {ρ : ℕ} (x : Matrix (Fin W.rank) (Fin ρ) D.host.E) (x' : Matrix (Fin W'.rank) (Fin ρ) D.host.E) (h : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew W (rat x : Mx D W.rank ρ) = GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.gramSkew W' (rat x')) : gramQ W x = gramQ W' x'

theorem **pairS_mul** (T : HSp D) {ρ : ℕ} (x : Mx D T.rank ρ) (c : Mx D ρ ρ) : pairS T x (x · c) = (adj c · (adj x · (gA T · x))).trace

theorem **gramQ_zlQ_mulVec_last** (B : BesLev D σ) (a : Fin (σ + 1) → D.host.E) : (gramQ B.L (zlQ B) *ᵥ a) (Fin.last σ) = (star B.ζ ·ᵥ B.L.gram *ᵥ B.ζ) * a (Fin.last σ)

theorem **basePoint_spec** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), ∃ x₀ : Matrix (Fin B.V.rank) (Fin (σ + 1)) D.host.E, (rat x₀ : Mx D B.V.rank (σ + 1)) = x0 B ∧ IsBase B.V x₀ ∧ Rgrp B.V (rat x₀ : Mx D B.V.rank (σ + 1)) = besGrp B.d B.o ∧ (∀ a : Fin (σ + 1) → D.host.E, gramQ B.V x₀ *ᵥ a = 0 → a (Fin.last σ) = 0) ∧ ∀ (S S₀ : HSp D) (d : HypDec S S₀ (σ + 1)) (c : skewA D (σ + 1)), d.qS B.V c (x0 B) = d.qS B.L c (zl B)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselBasePointIsUnfoldingBasePointSpecStatement

theorem **_root_.GlobalTheory.T.fourierBesselBasePoint_isUnfoldingBasePoint_spec** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), ∃ x₀ : Matrix (Fin B.V.rank) (Fin (σ + 1)) D.host.E, (rat x₀ : Mx D B.V.rank (σ + 1)) = x0 B ∧ IsBase B.V x₀ ∧ Rgrp B.V (rat x₀ : Mx D B.V.rank (σ + 1)) = besGrp B.d B.o ∧ (∀ a : Fin (σ + 1) → D.host.E, gramQ B.V x₀ *ᵥ a = 0 → a (Fin.last σ) = 0) ∧ ∀ (S S₀ : HSp D) (d : HypDec S S₀ (σ + 1)) (c : skewA D (σ + 1)), d.qS B.V c (x0 B) = d.qS B.L c (zl B)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselBasePointRationalWithRationalLeftIn

theorem **rat_x0Mat** (B : BesLev D σ) : rat (x0Mat B) = x0 B

theorem **besselBasePointLevelNorm_eq_gram_form** (B : BesLev D σ) : besselBasePointLevelNorm B = star B.ζ ·ᵥ B.L.gram *ᵥ B.ζ

theorem **besselBasePointLevelNorm_ne_zero** (B : BesLev D σ) : besselBasePointLevelNorm B ≠ 0

theorem **besselLevelDatum_qX_dot_e_eq_one** (B : BesLev D σ) : B.d.qX * B.d.e = 1

theorem **wittDecomposition_projectionX_mul_anisotropicInclusion_eq_zero** (B : BesLev D σ) : B.d.qX * B.d.i₀ = 0

theorem **wittDecomposition_projectionZero_mul_e_eq_zero** (B : BesLev D σ) : B.d.q0 * B.d.e = 0

theorem **wittDecomposition_projectionZero_mul_anisotropicInclusion_eq_one** (B : BesLev D σ) : B.d.q0 * B.d.i₀ = 1

theorem **besselBasePointLeftInverse_mul_snocMatrix** (B : BesLev D σ) : besselBasePointLeftInverse B * x0Mat B = 1

theorem **fourierBesselBasePoint_rational_with_rational_leftInverse** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∃ P ∈ ratMx D (σ + 1) B.V.rank, P · x0 B = 1) ∧ x0 B ∈ ratMx D B.V.rank (σ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselBasePointRationalWithRationalLeftInStatement

theorem **_root_.GlobalTheory.T.x0_mem_ratMx_and_exists_rat_leftInverse** : ∀ (D : D7) (σ : ℕ) (B : BesLev D σ), (∃ P ∈ ratMx D (σ + 1) B.V.rank, P · x0 B = 1) ∧ x0 B ∈ ratMx D B.V.rank (σ + 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselThetaCharIntegrandLaws

theorem **heisO_zero_apply** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (p : WO S T) (f : WO S T → ℂ) : heisO ψ S T p f 0 = f p

theorem **jacS_zero_apply** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x : Mx D T.rank ρ) (n : GA D S) (f : WO S₀ T → ℂ) : jacS ψ d T x n f 0 = ψ (cenS d T x n) * f (ptS d T x n)

theorem **jacS_nC** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x : Mx D T.rank ρ) (c : skewA D ρ) (f : WO S₀ T → ℂ) : jacS ψ d T x (d.nC c) f = ψ (d.qS T c x) • f

theorem **continuous_psiE** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) : Continuous (psiE ψ)

theorem **continuous_upSum** {X : Type} [TopologicalSpace X] {ρ : ℕ} {T : X → Mx D ρ ρ} (hT : Continuous T) : Continuous fun t => upSum (T t)

theorem **continuous_psiFlag** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : Continuous (psiFlag ψ d)

theorem **continuous_ptS** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x : Mx D T.rank ρ) : Continuous (ptS d T x)

theorem **continuous_cenS** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x : Mx D T.rank ρ) : Continuous (cenS d T x)

theorem **exists_bound_quasi** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) {S L : HSp D} {y : WO S L → ℂ} (hy : Quasi ψ S L y) : ∃ M : ℝ, ∀ w, ‖y w‖ ≤ M

theorem **zl_rat** {σ : ℕ} (B : BesLev D σ) : ∃ z0 : Matrix (Fin B.L.rank) (Fin (σ + 1)) D.host.E, zl B = rat z0

theorem **ptS_rat** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x0 : Matrix (Fin T.rank) (Fin ρ) D.host.E) {γ : GA D S} (hγ : γ ∈ GQ S) : ∃ g0, ptS d T (rat x0) γ = ratW T S₀ g0

theorem **psi_cenS_rat** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x0 : Matrix (Fin T.rank) (Fin ρ) D.host.E) {γ : GA D S} (hγ : γ ∈ GQ S) : ψ (cenS d T (rat x0) γ) = 1

theorem **upSum_rat** {ρ : ℕ} (M : Matrix (Fin ρ) (Fin ρ) D.host.E) : upSum (rat M : Mx D ρ ρ) = incA D (∑ i, ∑ j, if j.1 = i.1 + 1 then M i j else 0)

theorem **psiE_incA** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) (t : D.host.E) : psiE ψ (incA D t) = 1

theorem **psiFlag_rat** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsCharAF D.host.F ψ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {γ : GA D S} (hγ : γ ∈ GQ S) : psiFlag ψ d γ = 1

theorem **psiFlag_nC** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (c : skewA D ρ) : psiFlag ψ d (d.nC c) = 1

theorem **psiFlag_nB** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (b : Mx D ρ S₀.rank) : psiFlag ψ d (d.nB b) = 1

theorem **psiFlag_levi** (ψ : AddChar (AF D.host.F) ℂ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (u : GL (Fin ρ) (AE D)) : psiFlag ψ d (d.levi u) = psiUni ψ u

theorem **th_laws** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (y : WO J.W₀ B.L → ℂ), Quasi ψ J.W₀ B.L y → Continuous (th J ψ B y) ∧ (∃ M : ℝ, (∀ w, ‖y w‖ ≤ M) ∧ ∀ n, ‖th J ψ B y n‖ ≤ M) ∧ (∀ γ ∈ flagN J.d, γ ∈ GQ J.W → ∀ n ∈ flagN J.d, th J ψ B y (γ * n) = th J ψ B y n) ∧ (∀ c, ∀ n ∈ flagN J.d, th J ψ B y (J.d.nC c * n) = star (ψ (J.d.qS B.L c (zl B))) * th J ψ B y n) ∧ (∀ n ∈ flagN J.d, ∀ u ∈ UX D (σ + 1), th J ψ B y (n * J.d.levi u) = th J ψ B y n * star (psiUni ψ u)) ∧ ∀ b, ∀ u ∈ UX D (σ + 1), th J ψ B y (J.d.nB b * J.d.levi u) = star (psiUni ψ u) * star (y (zl B · b))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselThetaCharIntegrandLawsStatement

theorem **_root_.GlobalTheory.T.fourierBesselThetaCharIntegrand_laws** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (y : WO J.W₀ B.L → ℂ), Quasi ψ J.W₀ B.L y → Continuous (th J ψ B y) ∧ (∃ M : ℝ, (∀ w, ‖y w‖ ≤ M) ∧ ∀ n, ‖th J ψ B y n‖ ≤ M) ∧ (∀ γ ∈ flagN J.d, γ ∈ GQ J.W → ∀ n ∈ flagN J.d, th J ψ B y (γ * n) = th J ψ B y n) ∧ (∀ c, ∀ n ∈ flagN J.d, th J ψ B y (J.d.nC c * n) = star (ψ (J.d.qS B.L c (zl B))) * th J ψ B y n) ∧ (∀ n ∈ flagN J.d, ∀ u ∈ UX D (σ + 1), th J ψ B y (n * J.d.levi u) = th J ψ B y n * star (psiUni ψ u)) ∧ ∀ b, ∀ u ∈ UX D (σ + 1), th J ψ B y (J.d.nB b * J.d.levi u) = star (psiUni ψ u) * star (y (zl B · b))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselThetaIntegrandIsUnfoldingTheta

theorem **cA_cA'** (z : AE D) : cA D (cA D z) = z

theorem **form_mulVec** (S : HSp D) {ρ : ℕ} (J : Mx D S.rank ρ) (B : Fin ρ → AE D) (y : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.VA S) : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form S y (J *ᵥ B) = (fun k => cA D (B k)) ·ᵥ ((adj J · gA S) *ᵥ y)

theorem **tvN_apply** {ρ : ℕ} (A B : Fin ρ → AE D) (i k : Fin ρ) : tvN A B i k = A i * cA D (B k)

theorem **tvN_sq** {ρ : ℕ} {A B : Fin ρ → AE D} (h : (fun k => cA D (B k)) ·ᵥ A = 0) : tvN A B * tvN A B = 0

theorem **tvN_mulVec** {ρ : ℕ} (A B z : Fin ρ → AE D) : tvN A B *ᵥ z = ((fun k => cA D (B k)) ·ᵥ z) • A

theorem **adj_tvN** {ρ : ℕ} (A B : Fin ρ → AE D) : adj (tvN A B) = tvN B A

theorem **tvGL_val** {ρ : ℕ} (A B : Fin ρ → AE D) (h : (fun k => cA D (B k)) ·ᵥ A = 0) : (tvGL A B h).1 = 1 + tvN A B

theorem **tvGL_inv_val** {ρ : ℕ} (A B : Fin ρ → AE D) (h : (fun k => cA D (B k)) ·ᵥ A = 0) : (tvGL A B h)⁻¹.1 = 1 - tvN A B

theorem **mt_levi_tv** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (A B : Fin ρ → AE D) (h : (fun k => cA D (B k)) ·ᵥ A = 0) : mt S (d.levi (tvGL A B h)) = 1 + d.jX · tvN A B · d.pX - d.jY · tvN B A · d.pY

theorem **transv_null** {S : HSp D} {y v w : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.VA S} (hvv : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form S v v = 0) : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.transv S w v y = y + GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form S y v • w - GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.form S y w • v

theorem **levi_isTransv** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (A B : Fin ρ → AE D) (h : (fun k => cA D (B k)) ·ᵥ A = 0) : GlobalTheory.IndefiniteRowSideFullHecke.TransvectionAction.IsTransv S (d.jX *ᵥ A) (d.jY *ᵥ B) (d.levi (tvGL A B h))

theorem **trace_half_smul** (x : AF D.host.F) : Algebra.trace (AF D.host.F) (AE D) (x • incA D (algebraMap D.host.F D.host.E (1 / 2))) = x

theorem **exists_psiE_ne_one** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) : ∃ t : AE D, psiE ψ t ≠ 1

theorem **GlobalTheory.th_isTh** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (y : WO J.W₀ B.L → ℂ), Quasi ψ J.W₀ B.L y → ∀ x₀ : Matrix (Fin B.V.rank) (Fin (σ + 1)) D.host.E, (rat x₀ : Mx D B.V.rank (σ + 1)) = x0 B → IsTh ψ J.d B.V x₀ (th J ψ B y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierBesselThetaIntegrandIsUnfoldingThetaStatement

theorem **_root_.GlobalTheory.T.fourierBesselThetaIntegrand_isUnfoldingTheta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (y : WO J.W₀ B.L → ℂ), Quasi ψ J.W₀ B.L y → ∀ x₀ : Matrix (Fin B.V.rank) (Fin (σ + 1)) D.host.E, (rat x₀ : Mx D B.V.rank (σ + 1)) = x0 B → IsTh ψ J.d B.V x₀ (th J ψ B y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierCoefficientBessel

theorem **one_le_level** (X : HidaSetting D) : 1 ≤ X.B.g

theorem **one_mem_level** (X : HidaSetting D) : (1 : X.B.I.Gf) ∈ X.B.I.Kg X.B.g

theorem **psiInfU_of_mem_Hinf** (X : HidaSetting D) (u : D.what₂.Word) (η : X.B.I.Ginf) (hη : η ∈ Hinf X u) : X.psiInfU u η = 1

theorem **castGA_mem_rat** {I I' : Formalism.MainNotions.S.AutInst} (e : I = I') (x : I.GA) : x ∈ I.GFset ↔ castGA e x ∈ I'.GFset

theorem **iotaX_mem_rat** (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (g : GA D (D.sp.Y 0)) (hg : g ∈ GQ (D.sp.Y 0)) : iotaX X h g ∈ X.B.I.GFset

theorem **Ph_of_dec** (X : HidaSetting D) (u : D.what₂.Word) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK) (y : X.B.I.GA) (n : X.B.I.Gf) (hh : Hfin X u) (k : X.B.I.Gf) (hd : IsDec X u y.2 n hh k) : Ph X u hK T y = phVal X u hK T y.1 n hh k

theorem **continuous_toH** (B : SBes X h u ψ A) : Continuous B.toH

theorem **continuous_col** (hK : X.LevelSubgroupCharacter) (T : Cu X u hK) (ξ : X.Vpi u) : Continuous fun p : Hinf X u × Hfin X u => (T : X.Vpi u →ₗ[ℂ] (Hinf X u × Hfin X u → ℂ)) ξ p

theorem **ph_colForm** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK), (∀ ξ, IsContAut A (B.colForm T.1 ξ)) ∧ ∀ n ∈ flagN B.d, ∀ (a : GA D A) (y : X.B.I.Ginf), Ph X u hK T (iotaX X h (n ∗ embA B.d B.o a) ∗ (y, X.xu u)) = star (psiBes ψ B.d 1 (vecB B.o B.ζ) n) ∗ B.colForm T.1 (X.Wu u y) a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierCoefficientBesselStatement

theorem **_root_.GlobalTheory.T.fourier_coefficient_bessel** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (hK : X.LevelSubgroupCharacter) (T : Cu X u hK), (∀ ξ, IsContAut A (B.colForm T.1 ξ)) ∧ ∀ n ∈ flagN B.d, ∀ (a : GA D A) (y : X.B.I.Ginf), Ph X u hK T (iotaX X h (n ∗ embA B.d B.o a) ∗ (y, X.xu u)) = star (psiBes ψ B.d 1 (vecB B.o B.ζ) n) ∗ B.colForm T.1 (X.Wu u y) a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierInversionInverseRowTransformNegOne

theorem **finv_hat_neg_one** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (ψk : AddChar k Circle) (ν : Measure k), Continuous ψk → ψk ≠ 1 → ν.IsAddHaarMeasure → ∃ C : ℂ, ∀ c : Matrix n Unit k → ℂ, c ∈ SB (Matrix n Unit k) → ∀ Φ : Matrix Unit n k → ℂ, (∀ η, Φ η = GlobalTheory.StepData.hat ψk ν (-1) (fun y => c (Matrix.of fun r _ => y r)) (η ())) → GlobalTheory.KappaModel.finv ψk ν C Φ = c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierInversionInverseRowTransformNegOneStatement

theorem **_root_.GlobalTheory.T.fourierInversion_inverseRowTransform_negOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (ψk : AddChar k Circle) (ν : Measure k), Continuous ψk → ψk ≠ 1 → ν.IsAddHaarMeasure → ∃ C : ℂ, ∀ c : Matrix n Unit k → ℂ, c ∈ SB (Matrix n Unit k) → ∀ Φ : Matrix Unit n k → ℂ, (∀ η, Φ η = GlobalTheory.StepData.hat ψk ν (-1) (fun y => c (Matrix.of fun r _ => y r)) (η ())) → GlobalTheory.KappaModel.finv ψk ν C Φ = c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiBesselFlatMapFactorIsLocalModel

theorem **affPt_add_rowAt** (l : Fin c) (p : ℕ) (hl : l.val + 1 = p) (x x₀ : Matrix (n ⊕ Unit) n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) : affPt p x.toRows₁ e + rowAt n c p (x (Sum.inr ())) + embR l x₀ = affPt p (x + x₀).toRows₁ e + rowAt n c p ((x + x₀) (Sum.inr ()))

theorem **ch_affPt_add** (p : ℕ) (x x₀ : Matrix (n ⊕ Unit) n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) : ch (affPt p (x + x₀).toRows₁ e) = ch (affPt p x.toRows₁ e)

theorem **flRow_translate** (ψ : AddChar k Circle) (ν : Measure k) (l : Fin c) (p : ℕ) (hl : l.val + 1 = p) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (x₀ x : Matrix (n ⊕ Unit) n' k) : GlobalTheory.LocFlat.flRow ψ ν p (fun t => Φ (t + embR l x₀)) x = GlobalTheory.LocFlat.flRow ψ ν p Φ (x + x₀)

theorem **restr_affPt** (l : Fin c) (p : ℕ) (hl : l.val + 1 = p) (x : Matrix (n ⊕ Unit) n' k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free p rs} → k) : restr l (affPt p x.toRows₁ e + rowAt n c p (x (Sum.inr ()))) = x

theorem **flRow_modulate** (ψ : AddChar k Circle) (ν : Measure k) (l : Fin c) (p : ℕ) (hl : l.val + 1 = p) (Φ Φ' : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (g : Matrix (n ⊕ Unit) n' k → ℂ) (h : ∀ t, Φ' t = g (restr l t) ∗ Φ t) (x : Matrix (n ⊕ Unit) n' k) : GlobalTheory.LocFlat.flRow ψ ν p Φ' x = g x ∗ GlobalTheory.LocFlat.flRow ψ ν p Φ x

theorem **map_pr_mul** {l m m' : Type} [Fintype m] (M : Matrix l m D.host.E) (N : Matrix m m' D.host.E) : (M ∗ N).map p.pr = M.map p.pr ∗ N.map p.pr

theorem **pr_mulVec** {l m : Type} [Fintype m] (M : Matrix l m D.host.E) (v : m → D.host.E) (r : l) : p.pr ((M *ᵥ v) r) = ∑ a, p.pr (M r a) ∗ p.pr (v a)

theorem **wittDecomposition_coreProj_mul_core_eq_one** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.q0 ∗ d.i₀ = 1

theorem **projection_q_zero_i_zero_eq_one** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.q0.map p.pr ∗ d.i₀.map p.pr = 1

theorem **projection_q_zero_e_eq_zero** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.q0.map p.pr ∗ d.e.map p.pr = 0

theorem **projection_q_zero_e_prime_eq_zero** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.q0.map p.pr ∗ d.e'.map p.pr = 0

theorem **mul_col_eq** {R m N Y Z : Type} [CommRing R] [Fintype N] (Q : Matrix m N R) (F : Matrix N Y R) (y : Y) (C : R) (E : Matrix N Z R) (z : Z) (h : ∀ r, F r y = C ∗ E r z) (a : m) : (Q ∗ F) a y = C ∗ (Q ∗ E) a z

theorem **mul_col_eq'** {R m N Y Z : Type} [CommRing R] [Fintype N] (Q : Matrix m N R) (F : Matrix N Y R) (y : Y) (E : Matrix N Z R) (z : Z) (h : ∀ r, F r y = E r z) (a : m) : (Q ∗ F) a y = (Q ∗ E) a z

theorem **embR_eq** (l : Fin c) (x₀ : Matrix (n ⊕ Unit) n' k) : embR l x₀ = (selR k n l)ᵀ ∗ x₀ ∗ selC k n' (c + 1)

theorem **restr_eq** (l : Fin c) (t : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : restr l t = selR k n l ∗ t ∗ (selC k n' (c + 1))ᵀ

theorem **jac_inc** {ρ : ℕ} {J : JacLev D ρ} {s hh : k} {G₀ : Frame k J.W₀.rank n'} {F : Frame k J.W.rank (n' ⊕ Fin c)} (h : p.IsJacFrame J s hh G₀ F) : F.γ ∗ J.d.i₀.map p.pr = (selC k n' c)ᵀ ∗ G₀.γ

theorem **jac_proj** {ρ : ℕ} {J : JacLev D ρ} {s hh : k} {G₀ : Frame k J.W₀.rank n'} {F : Frame k J.W.rank (n' ⊕ Fin c)} (h : p.IsJacFrame J s hh G₀ F) : selC k n' c ∗ F.γ = G₀.γ ∗ J.d.q0.map p.pr

theorem **bes_inc** {ρ : ℕ} {B : BesLev D ρ} {s : k} {FA : Frame k B.A.rank n} {F₀ : Frame k B.V₀.rank (n ⊕ Unit)} {F : Frame k B.V.rank (n ⊕ Fin c)} (h₀ : p.IsCoreFrame B s FA F₀) (h : p.IsBesFrame B s FA F) (l : Fin c) (hl : l.val = ρ) : F.β ∗ (selR k n l)ᵀ = B.d.i₀.map p.pr ∗ F₀.β

theorem **bes_proj** {ρ : ℕ} {B : BesLev D ρ} {s : k} {FA : Frame k B.A.rank n} {F₀ : Frame k B.V₀.rank (n ⊕ Unit)} {F : Frame k B.V.rank (n ⊕ Fin c)} (h₀ : p.IsCoreFrame B s FA F₀) (h : p.IsBesFrame B s FA F) (l : Fin c) (hl : l.val = ρ) : B.d.q0.map p.pr ∗ F.β = F₀.β ∗ selR k n l

theorem **e_mul_ι** (x : k) : p.e ∗ p.ι x = p.ι x

theorem **incA_mul_ι** (a : D.host.E) (x : k) : (incA D a : AE D) ∗ p.ι x = p.ι (p.pr a ∗ x)

theorem **rat_mul_ι** {l m n : ℕ} (M : Matrix (Fin l) (Fin m) D.host.E) (N : Matrix (Fin m) (Fin n) k) : (rat M : Mx D l m) ∗ (N.map p.ι : Mx D m n) = (M.map p.pr ∗ N).map p.ι

theorem **ι_mul_rat** {l m n : ℕ} (N : Matrix (Fin l) (Fin m) k) (M : Matrix (Fin m) (Fin n) D.host.E) : (N.map p.ι : Mx D l m) ∗ (rat M : Mx D m n) = (N ∗ M.map p.pr).map p.ι

theorem **infPart_eq_zero** {e : AE D} (he : IsSplitIdem D e) {z : AE D} (hz : e ∗ z = z ∨ cA D e ∗ z = z) : infPart D z = 0

theorem **mxAt_of_half** {e : AE D} (he : IsSplitIdem D e) {m n : ℕ} {x : Mx D m n} (hx : IsHalf e x ∨ IsHalf (cA D e) x) : GlobalTheory.FinPl.MxAt (fun _ => True) x

theorem **isHalf_mul** {e : AE D} {l m n r : ℕ} (A : Mx D l m) {y : Mx D m n} (hy : IsHalf e y) (C : Mx D n r) : IsHalf e (A * y * C : Mx D l r)

theorem **hO_core** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (dS : HypDec S S₀ ρ) (dT : HypDec T T₀ σ) (x : WO S T) (y : WO S₀ T₀) : hO S T x (dS.ext0 T (dT.inc0 S₀ y)) = hO S₀ T₀ (dT.p0 * x * dS.j0 : Mx D _ _) y

theorem **continuous_hO_up** {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (y : WO S T) : Continuous fun t : Matrix Y Y' k => p.π (hO S T (p.up FS FT t) y)

theorem **up_embR** (hJ : p.IsJacFrame J s hh G₀ FW) (h₀ : p.IsCoreFrame B s' FA F₀) (hB : p.IsBesFrame B s' FA FV) (l : Fin c) (hl : l.val = σ) (x₀ : Matrix (n ⊕ Unit) n' k) : p.up FW FV (embR l x₀) = J.d.ext0 B.V (B.d.inc0 J.W₀ (p.up G₀ F₀ x₀))

theorem **core_up** (hJ : p.IsJacFrame J s hh G₀ FW) (h₀ : p.IsCoreFrame B s' FA F₀) (hB : p.IsBesFrame B s' FA FV) (l : Fin c) (hl : l.val = σ) (t : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : (B.d.p0 * p.up FW FV t * J.d.j0 : Mx D _ _) = p.up G₀ F₀ (restr l t)

theorem **factor_isModel** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (T₀ : SB (Matrix (n ⊕ Unit) n' k) →ₗ[ℂ] (WO J.W₀ B.V₀ → ℂ)), (∀ (Φ : SB (Matrix (n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiBesselFlatMapFactorIsLocalModelStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_bessel_flatMap_factor_isLocalModel** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (T₀ : SB (Matrix (n ⊕ Unit

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiBesselFlatMapRowMultiplication

theorem **infPart_half** {e : AE D} (he : IsSplitIdem D e) {z : AE D} (hz : cA D e * z = z) : infPart D z = 0

theorem **pr_ne** {a : D.host.E} (ha : a ≠ 0) : p.pr a ≠ 0

theorem **pr_dot** {N M : ℕ} (Q : Matrix (Fin M) (Fin N) D.host.E) (v : Fin N → D.host.E) (i : Fin M) (a : k) : ∑ r, p.pr (Q i r) * (a * p.pr (v r)) = a * p.pr ((Q *ᵥ v) i)

theorem **pr_zero** : p.pr 0 = 0

theorem **pr_one** : p.pr 1 = 1

theorem **pr_map_mul** {L M N : ℕ} (X : Matrix (Fin L) (Fin M) D.host.E) (Y : Matrix (Fin M) (Fin N) D.host.E) : (X * Y).map p.pr = X.map p.pr * Y.map p.pr

theorem **hO_iso** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (dS : HypDec S S₀ ρ) (dT : HypDec T T₀ σ) (x : WO S T) (N : Mx D S₀.rank σ) : hO S T x (dS.ext0 T (dT.incX S₀ (adj (gA S₀ · N)))) = (N · (dT.pY · (x · dS.j0))).trace

theorem **qY_frame** {σ : ℕ} (B : BesLev D σ) (s : k) {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} (FA : Frame k B.A.rank n) (FV : Frame k B.V.rank (n ⊕ Fin c)) (hB : p.IsBesFrame B s FA FV) (i : Fin σ) (l : Fin c) (hl : l.val + i.val = 2 * σ) (r : n ⊕ Fin c) : (B.d.qY.map p.pr * FV.β) i r = if r = Sum.inr l then s * (-p.hz B / p.xi) * (-1 / p.xi) ^ (σ - 1 - i.val) else 0

theorem **exists_row** (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : k) (hs : s ≠ 0) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (hh : k) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (FV : Frame k B.V.rank (n ⊕ Fin c)) (hJ : p.IsJacFrame J s hh G₀ FW) (hB : p.IsBesFrame B (s * p.xi) FA FV) (l : Fin c) (s' : n') (hl : σ < l.val) (b : k) : ∃ ξ : Mx D σ J.W₀.rank, (∀ i j, cA D p.e * ξ i j = ξ i j) ∧ ∀ t, p.π (hO J.W B.V (p.up FW FV t) (J.d.ext0 B.V (B.d.incX J.W₀ ξ))) = b * t (Sum.inr l) (Sum.inl s')

theorem **GlobalTheory.LocMod.flatFB_mulrow** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ l s', r = Sum.inr l ∧ σ < l.val ∧ q = Sum.inl s') → ∀ (b : k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiBesselFlatMapRowMultiplicationStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_bessel_flatMap_row_multiplication** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), (∃ l s', r = Sum.inr l ∧ σ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiBesselFlatMapUnipotentLaw

theorem **det_of_mem_N** {k : Type} [Field k] {n : Type} [Fintype n] [DecidableEq n] {c p : ℕ} {g : GL (n ⊕ Fin c) k} (hg : g ∈ N k n c p) : Matrix.GeneralLinearGroup.det g = 1

theorem **unip_law** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c +

1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ u : GL (Fin c) k, embA n u ∈ N k n c (σ + 1) → ∀ (Φ Φ' : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), (∀ t, Φ'.1 t = Φ.1 (((em

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiBesselFlatMapUnipotentLawStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_bessel_flatMap_unipotent_law** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ u : GL (Fin c) k, embA n u ∈ N k n c (σ + 1) → ∀ (Φ Φ' : SB (Matrix (n ⊕

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiFlatMapSupportFiniteCells

theorem **det_colE** {k : Type} [Field k] {n : Type} [Fintype n] [DecidableEq n] (c p : ℕ) (β : n → k) : Matrix.GeneralLinearGroup.det (colE c p β) = 1

theorem **om_inv_of_det_one** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g : GL b k) (hg : Matrix.GeneralLinearGroup.det g = 1) (Ψ : Matrix b a k → ℂ) : om μ (1 : GL a k) g⁻¹ Ψ = fun z => Ψ (g.val * z)

theorem **om_eq_mul_om_triv** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (h : GL b k) (Ψ : Matrix b a k → ℂ) : om μ (1 : GL a k) h Ψ = fun z => (((μ (Matrix.GeneralLinearGroup.det h))⁻¹ : ℂˣ) : ℂ) * om 1 (1 : GL a k) h Ψ z

theorem **opE_const_mul** [MeasurableSpace k] [BorelSpace k] {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (ψ : AddChar k Circle) (ν : Measure k) (p : ℕ) (s : ℂ) (Ψ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (x₀ : Matrix n n' k) : opE ψ ν p (fun z => s * Ψ z) x₀ = s * opE ψ ν p Ψ x₀

theorem **flCol_support_finite_cells** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (μ : kˣ →* ℂˣ) (J : Subgroup (GL (n ⊕ Fin c) k)), IsOpen (J : Set (GL (n ⊕ Fin c) k)) → ∀ (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (a : GL (n' ⊕ Fin (c + 1)) k), ∃ T : Finset (GL (n ⊕ Fin c) k), ∀ x : GL (n ⊕ Fin c) k, GlobalTheory.LocFlat.flCol ψ ν p (om μ a x Φ.1) ≠ 0 → ∃ β : n → k, ∃ r ∈ R k n c p, ∃ t ∈ T, ∃ j ∈ J, x = colE c p β * r * t * j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiFlatMapSupportFiniteCellsStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_flatMap_support_finite_cells** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (μ : kˣ →* ℂˣ) (J : Subgroup (GL (n ⊕ Fin c) k)), IsOpen (J : Set (GL (n ⊕ Fin c) k)) → ∀ (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (a : GL (n' ⊕ Fin (c + 1)) k), ∃ T : Finset (GL (n ⊕ Fin c) k), ∀ x : GL (n ⊕ Fin c) k, GlobalTheory.LocFlat.flCol ψ ν p (om μ a x Φ.1) ≠ 0 → ∃ β : n → k, ∃ r ∈ R k n c p, ∃ t ∈ T, ∃ j ∈ J, x = colE c p β * r * t * j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiGroupHaarIsMulRightInvariant

theorem **jacGrp_rightInv** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (μ : Measure (jacGrp d)), μ.IsHaarMeasure → μ.IsMulRightInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiGroupHaarIsMulRightInvariantStatement

theorem **_root_.GlobalTheory.T.fourierJacobiGroup_haar_isMulRightInvariant** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (μ : Measure (jacGrp d)), μ.IsHaarMeasure → μ.IsMulRightInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiGroupSplitRealFinite

theorem **tensNU_map** (φ : R' →ₐ[F] R) (j : R →ₗ[F] R') (hj : ∀ a b, j (a * b) = j a * j b) (h : ∀ a, j (φ a) = j 1 * a) (z : R' ⊗[F] E) : tensNU j hj (Algebra.TensorProduct.map φ (AlgHom.id F E) z) = tensNU (E

theorem **mat_extH_mapH** (φ : R' →ₐ[F] R) (j : R →ₗ[F] R') (hj : ∀ a b, j (a * b) = j a * j b) (h : ∀ a, j (φ a) = j 1 * a) (g : HA R' σ J) : ∃ c : R' ⊗[F] E, (HA.mat R' σ J (extH σ J j hj (mapH σ J φ g))).1 = (1 - c) • (1 : Matrix (Fin M) (Fin M) (R' ⊗[F] E)) + c • (HA.mat R' σ J g).1

theorem **conj_mix** {k n m : ℕ} (c : R) (P : Matrix (Fin k) (Fin n) R) (G : Matrix (Fin n) (Fin n) R) (Q : Matrix (Fin n) (Fin m) R) : P * (((1 - c) • (1 : Matrix (Fin n) (Fin n) R) + c • G) * Q) = (1 - c) • (P * Q) + c • (P * (G * Q))

theorem **mix_zero** {k m : ℕ} (c : R) {A B : Matrix (Fin k) (Fin m) R} (hA : A = 0) (hB : B = 0) : (1 - c) • A + c • B = 0

theorem **mix_uni** {k : ℕ} (c : R) {A B : Matrix (Fin k) (Fin k) R} (hA : A = 1) (hB : ∀ i j : Fin k, (j < i → B i j = 0) ∧ B i i = 1) (i j : Fin k) : (j < i → ((1 - c) • A + c • B) i j = 0) ∧ ((1 - c) • A + c • B) i i = 1

theorem **mt_ofInf_projInf** (V : HSp D) (g : GA D V) : ∃ c : AE D, mt V (ofInf D.sp.cm.sig V.gram (projInf D.sp.cm.sig V.gram g)) = (1 - c) • (1 : Mx D V.rank V.rank) + c • mt V g

theorem **mt_ofFin_projFin** (V : HSp D) (g : GA D V) : ∃ c : AE D, mt V (ofFin D.sp.cm.sig V.gram (projFin D.sp.cm.sig V.gram g)) = (1 - c) • (1 : Mx D V.rank V.rank) + c • mt V g

theorem **mem_jacGrp_of_mat** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (c : AE D) {g h : GA D S} (hg : g ∈ jacGrp d) (hh : mt S h = (1 - c) • (1 : Mx D S.rank S.rank) + c • mt S g) : h ∈ jacGrp d

theorem **jacGrp_split_inf_fin** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), ∀ a ∈ GinfC V, ∀ b ∈ GfC V, a * b ∈ jacGrp d → a ∈ jacGrp d ∧ b ∈ jacGrp d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiGroupSplitRealFiniteStatement

theorem **_root_.GlobalTheory.T.fourierJacobiGroup_split_real_finite** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ), ∀ a ∈ GinfC V, ∀ b ∈ GfC V, a * b ∈ jacGrp d → a ∈ jacGrp d ∧ b ∈ jacGrp d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiHandOffAutomorphic

theorem **handFBLaw** : ∀ D : D7, HandFBLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiHandOffAutomorphicStatement

theorem **_root_.GlobalTheory.T.fourierJacobiHandOff_automorphic** : ∀ D : D7, HandFBLaw D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiIntegralOverNUnfold

theorem **weilRepresentationOnFunctions_embed_mul** {D : D7} {V W : HSp D} (om : Om V W) (e g : GA D V) (h : GA D W) (Φ : WO V W → ℂ) : om (g, h) (om (e, 1) Φ) = om (g * e, h) Φ

theorem **fjN_unfold** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (G : GA D B.V → ℂ) (Φ : WO J.W B.V → ℂ), IsQuasiFam ψ J.W₀ B.L Y → RapidDec B.V G → IsThetaFunO ψ J.W B.V Φ → ∀ 𝓑 : Set (GA D B.V), IsFundamentalDomain (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)) 𝓑 s.vW.μ → ∀ h₀, ∫ n in J.vN.𝒩, fjSrcKerFun ψ J.d (J.ker ψ B.ζ Y) (s.lift G Φ) h₀ n ∂J.vN.ν = ∫ h in 𝓑, G h * tFlat s vL (Y h₀) (s.om (J.d.emb0 h₀, h) Φ) ∂s.vW.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiIntegralOverNUnfoldStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_integral_over_N_unfold** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (G : GA D B.V → ℂ) (Φ : WO J.W B.V → ℂ), IsQuasiFam ψ J.W₀ B.L Y → RapidDec B.V G → IsThetaFunO ψ J.W B.V Φ → ∀ 𝓑 : Set (GA D B.V), IsFundamentalDomain (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)) 𝓑 s.vW.μ → ∀ h₀, ∫ n in J.vN.𝒩, fjSrcKerFun ψ J.d (J.ker ψ B.ζ Y) (s.lift G Φ) h₀ n ∂J.vN.ν = ∫ h in 𝓑, G h * tFlat s vL (Y h₀) (s.om (J.d.emb0 h₀, h) Φ) ∂s.vW.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiInvariantFunctionIntegrable

theorem **vol_fund_lt_top** {D : D7} {V : HSp D} (hc : CoCpt V) (v : Vol V) : v.μ v.𝓕 < ⊤

theorem **jac_bound** (D : D7) (ρ : ℕ) (J : JacLev D ρ) (K : GA D J.W → ℂ) (hK : Continuous K) (hinv : ∀ γ ∈ GQ J.W ⊓ jacGrp J.d, ∀ h, K (γ * h) = K h) (h : GA D J.W) : ∃ C, ∀ (n : flagN J.d) (h' : GA D J.W₀), ‖K (n.1 * J.d.emb0 h' * h)‖ ≤ C

theorem **jac_absConv** : ∀ (D : D7) (ρ : ℕ) (J : JacLev D ρ) (K : GA D J.W → ℂ), Continuous K → (∀ γ ∈ GQ J.W ⊓ jacGrp J.d, ∀ h, K (γ * h) = K h) → ∀ h, AbsConv J.v0.μ J.v0.𝓕 J.vN.ν J.vN.𝒩 fun h' n => K (n.1 * J.d.emb0 h' * h)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiInvariantFunctionIntegrableStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_invariant_function_integrable** : ∀ (D : D7) (ρ : ℕ) (J : JacLev D ρ) (K : GA D J.W → ℂ), Continuous K → (∀ γ ∈ GQ J.W ⊓ jacGrp J.d, ∀ h, K (γ * h) = K h) → ∀ h, AbsConv J.v0.μ J.v0.𝓕 J.vN.ν J.vN.𝒩 fun h' n => K (n.1 * J.d.emb0 h' * h)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiKernelEqWeilRepresentationApply

theorem **fjKer_eq_om_apply** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (W W₀ V₀ A L : HSp D) (ρ : ℕ) (dW : HypDec W W₀ ρ) (o : OrthDec V₀ A L) (ζ : Fin L.rank → D.host.E) (χ χ' : ((AE D)ˣ →* Circle)) (om : Om V₀ W₀), IsWeilDatumO ψ V₀ W₀ χ χ' om → ∀ Ξ, IsThetaFunO ψ V₀ W₀ Ξ → ∀ h₀ n h, fjKer ψ dW o ζ om Ξ h₀ n h = ψ (cenT dW L (yZero L ρ ζ) n) * om (o.emb (h₀, 1), h) Ξ ((o.resS W₀).symm (0, ptT dW L (yZero L ρ ζ) n))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiKernelEqWeilRepresentationApplyStatement

theorem **_root_.GlobalTheory.T.fourierJacobiKernel_eq_weilRepresentation_apply** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (W W₀ V₀ A L : HSp D) (ρ : ℕ) (dW : HypDec W W₀ ρ) (o : OrthDec V₀ A L) (ζ : Fin L.rank → D.host.E) (χ χ' : ((AE D)ˣ →* Circle)) (om : Om V₀ W₀), IsWeilDatumO ψ V₀ W₀ χ χ' om → ∀ Ξ, IsThetaFunO ψ V₀ W₀ Ξ → ∀ h₀ n h, fjKer ψ dW o ζ om Ξ h₀ n h = ψ (cenT dW L (yZero L ρ ζ) n) * om (o.emb (h₀, 1), h) Ξ ((o.resS W₀).symm (0, ptT dW L (yZero L ρ ζ) n))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiLatticeSplitsAtNormalFramePlace

theorem **monoW_of_normalFrames** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊛ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m), GlobalTheory.FinPl.MonoAt ((y.jac j ρ h).lev

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiLatticeSplitsAtNormalFramePlaceStatement

theorem **_root_.GlobalTheory.T.fourierJacobiLattice_splits_at_normalFrame_place** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ {a : XGeo D} (y : YRat a) {βT : GL (Fin D.sp.X'.rank) (Lev.EAf D)} {PP : Pl D.host.E} (l : LocX a βT PP) (FY : FYFam l) (sc : ℕ → l.k), IsNormalFr l y FY sc → (∀ j, j ≤ D.R.m → (a.x j).Is) → ∀ (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) (M : ∀ j ρ (h : j + 2 * ρ + 2 = D.R.m), GlobalTheory.FinPl.Lat (y.bes j ρ h).V₀), (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (N j)) → (∀ j c (hc : j + c = l.s + 1), IntIn l.q (Yc D j) (FY j c hc) (N j)) → (∀ j ρ h, IsSelfDualC (y.bes j ρ h).V₀ (M j ρ h)) → (∀ j ρ h (hj : j ≤ l.s + 1) (F₀ : Frame l.k ((y.bes j ρ h).lev (a.x j).vol).V₀.rank (XIx (Fin D.R.k) (Fin D.R.t) j ⊛ Unit)), (ofSplit l.q).IsCoreFrame ((y.bes j ρ h).lev (a.x j).vol) (sc j) (l.FX j hj) F₀ → IntIn l.q _ F₀ (M j ρ h)) → ∀ j ρ (h : j + 2 * ρ + 2 = D.R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiMeasureUnitriangularCoordinates

theorem **lv_mem** (u : UX D ρ) : lv d u ∈ flagN d

theorem **radX_le** : radX d ≤ flagN d

theorem **lv_conj** (n : GA D S) (hn : n ∈ radX d) (u : UX D ρ) : lv d u * n * (lv d u)⁻¹ ∈ radX d

theorem **anisotropicEmbedding_conj_wittFlagUnipotentFixer** (n : GA D S) (hn : n ∈ radX d) (h : GA D S₀) : d.emb0 h * n * (d.emb0 h)⁻¹ ∈ radX d

theorem **radX_sup_lv** : radX d ⊔ (lv d).range = flagN d

theorem **wittFlagUnipotentFixer_sup_anisotropic_eq_stabilizer** : radX d ⊔ d.emb0.range = stabX d

theorem **blockX_mul_lv** {n : GA D S} (hn : n ∈ radX d) (u : UX D ρ) : blockX d (n * lv d u) = u.1.1

theorem **exists_rad_lv** {n : GA D S} (hn : n ∈ flagN d) : ∃ n₀ ∈ radX d, ∃ u, n = n₀ * lv d u

theorem **continuous_blockX** : Continuous (blockX d)

theorem **isClosed_stabX** : IsClosed (stabX d : Set (GA D S))

theorem **isClosed_flagN** : IsClosed (flagN d : Set (GA D S))

theorem **isClosed_radX** : IsClosed (radX d : Set (GA D S))

theorem **exists_proj_lv** : ∃ p : flagN d → UX D ρ, Continuous p ∧ ∀ (u : UX D ρ) (n : radX d) (hm : n.1 ∗ lv d u ∈ flagN d), p ⟨n.1 ∗ lv d u, hm⟩ = u

theorem **rat_lv** (n : GA D S) (hn : n ∈ radX d) (u : UX D ρ) : n ∗ lv d u ∈ GQ S ↔ n ∈ GQ S ∧ u ∈ UXQ D ρ

theorem **jac_measure** : ∀ (D : D7) (ρ : ℕ) (J : JacLev D ρ) (vU : VolU D ρ) (ν₀ : Measure (radX J.d)) (𝒩₀ : Set (radX J.d)), ν₀.IsHaarMeasure → IsFundamentalDomain ((GQ J.W).subgroupOf (radX J.d)) 𝒩₀ ν₀ → ν₀ 𝒩₀ = 1 → IsClosed ((radX J.d ⊔ J.d.emb0.range : Subgroup (GA D J.W)) : Set (GA D J.W)) ∧ (semiMeas (radX J.d) J.d.emb0 J.v0.μ ν₀).IsHaarMeasure ∧ (semiMeas (radX J.d) J.d.emb0 J.v0.μ ν₀).IsInvInvariant ∧ J.μR.IsInvInvariant ∧ ∀ F : GA D J.W → ENNReal, Measurable F → ∫⁻ r : jacGrp J.d, F r ∂J.μR = ∫⁻ u, ∫⁻ r : (radX J.d ⊔ J.d.emb0.range : Subgroup (GA D J.W)), F (r ∗ J.d.levi u.1) ∂(semiMeas (radX J.d) J.d.emb0 J.v0.μ ν₀) ∂vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiMeasureUnitriangularCoordinatesStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_measure_unitriangular_coordinates** : ∀ (D : D7) (ρ : ℕ) (J : JacLev D ρ) (vU : VolU D ρ) (ν₀ : Measure (radX J.d)) (𝒩₀ : Set (radX J.d)), ν₀.IsHaarMeasure → IsFundamentalDomain ((GQ J.W).subgroupOf (radX J.d)) 𝒩₀ ν₀ → ν₀ 𝒩₀ = 1 → IsClosed ((radX J.d ⊔ J.d.emb0.range : Subgroup (GA D J.W)) : Set (GA D J.W)) ∧ (semiMeas (radX J.d) J.d.emb0 J.v0.μ ν₀).IsHaarMeasure ∧ (semiMeas (radX J.d) J.d.emb0 J.v0.μ ν₀).IsInvInvariant ∧ J.μR.IsInvInvariant ∧ ∀ F : GA D J.W → ENNReal, Measurable F → ∫⁻ r : jacGrp J.d, F r ∂J.μR = ∫⁻ u, ∫⁻ r : (radX J.d ⊔ J.d.emb0.range : Subgroup (GA D J.W)), F (r ∗ J.d.levi u.1) ∂(semiMeas (radX J.d) J.d.emb0 J.v0.μ ν₀) ∂vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiNormalFrameIsotropicColumnsPairing

theorem **cols_pair** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ρ : ℕ) (J : JacLev D ρ) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (s hh : k) (F₀ : Frame k J.W₀.rank n) (F : Frame k J.W.rank (n ⊕ Fin c)), p.IsJacFrame J s hh F₀ F → ∀ i : Fin ρ, ∃ m₀ m₁ : n ⊕ Fin c, (∀ r, F.β r m₀ = s ∗ p.xi ^ (ρ − i.val) ∗ p.pr (J.d.e r i)) ∧ (∀ r, F.β r m₁ = s ∗ (hh / p.xi) ∗ (−1 / p.xi) ^ (ρ − 1 − i.val) ∗ p.pr (J.d.e' r i)) ∧ ∀ m, m ≠ m₁ → ∑ r, ∑ r', p.pr (star (J.d.e r i)) ∗ p.pr (J.W.gram r r') ∗ F.β r' m = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiNormalFrameIsotropicColumnsPairingStatement

theorem **_root_.GlobalTheory.T.fourierJacobiNormalFrame_isotropicColumns_pairing** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ρ : ℕ) (J : JacLev D ρ) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (s hh : k) (F₀ : Frame k J.W₀.rank n) (F : Frame k J.W.rank (n ⊕ Fin c)), p.IsJacFrame J s hh F₀ F → ∀ i : Fin ρ, ∃ m₀ m₁ : n ⊕ Fin c, (∀ r, F.β r m₀ = s ∗ p.xi ^ (ρ − i.val) ∗ p.pr (J.d.e r i)) ∧ (∀ r, F.β r m₁ = s ∗ (hh / p.xi) ∗ (−1 / p.xi) ^ (ρ − 1 − i.val) ∗ p.pr (J.d.e' r i)) ∧ ∀ m, m ≠ m₁ → ∑ r, ∑ r', p.pr (star (J.d.e r i)) ∗ p.pr (J.W.gram r r') ∗ F.β r' m = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiPeriodLiftEqIntegralUnfolded

theorem **per_lift_eq_integral_unfF** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (G : GA D B.V → ℂ) (Φ : WO J.W B.V → ℂ), IsQuasiFam ψ J.W₀ B.L Y → RapidDec B.V G → IsThetaFunO ψ J.W B.V Φ → ∀ 𝓑 : Set (GA D B.V), IsFundamentalDomain (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)) 𝓑 s.vW.μ → IntegrableOn (unfF s vL Y G Φ) 𝓑 s.vW.μ ∧ J.per ψ B.ζ Y (s.lift G Φ) = ∫ h in 𝓑, unfF s vL Y G Φ h ∂s.vW.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiPeriodLiftEqIntegralUnfoldedStatement

theorem **_root_.GlobalTheory.T.fourierJacobiPeriod_lift_eq_integral_unfolded** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (G : GA D B.V → ℂ) (Φ : WO J.W B.V → ℂ), IsQuasiFam ψ J.W₀ B.L Y → RapidDec B.V G → IsThetaFunO ψ J.W B.V Φ → ∀ 𝓑 : Set (GA D B.V), IsFundamentalDomain (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)) 𝓑 s.vW.μ → IntegrableOn (unfF s vL Y G Φ) 𝓑 s.vW.μ ∧ J.per ψ B.ζ Y (s.lift G Φ) = ∫ h in 𝓑, unfF s vL Y G Φ h ∂s.vW.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiPeriodUnfoldingOfWittDescent

theorem **GlobalTheory.Unf.unfFB_of_descent** : ∀ D : D7, Descent D → UnfFB D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiPeriodUnfoldingOfWittDescentStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_period_unfolding_of_witt_descent** : ∀ D : D7, Descent D → UnfFB D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiProductMapHaarFiniteOnCompacts

theorem **finite_on_compacts_prodRFJ** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ), 1 ≤ ρ → ρ < c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → IsFiniteMeasureOnCompacts (((m.prod (volN ν n c (ρ + 1))).prod (Measure.pi fun _ : n => ν)).map (prodRFJ ρ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiProductMapHaarFiniteOnCompactsStatement

theorem **_root_.GlobalTheory.T.fourierJacobiProductMap_haar_finiteOnCompacts** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ), 1 ≤ ρ → ρ < c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → IsFiniteMeasureOnCompacts (((m.prod (volN ν n c (ρ + 1))).prod (Measure.pi fun _ : n => ν)).map (prodRFJ ρ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiTargetBilinearFormIntegral

theorem **exists_isQuotInt_tInt** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c ρ : ℕ} (hρ : 1 ≤ ρ) (hρc : ρ ≤ c) {ψ : AddChar k Circle} (hψ : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (μ : kˣ →* ℂˣ) (hμ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] {A : Type} [AddCommGroup A] [Module ℂ A] {σ : Representation ℂ (GL (n ⊕ Fin c) k) A} (hσ : IsSmoothRep σ) {B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ} (hB : BilPos ψ μ ρ σ B) (F : A) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) : ∃ z, IsQuotInt ν μX μG ρ (tInt ψ ν μ ρ σ B F Φ.1) z

theorem **target_bilinear** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (_ : 1 ≤ ρ) (_ : c = 2 * ρ) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (fj : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] SB (n → k) →ₗ[ℂ] ℂ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : Represents ψ ν ρ σ fj B) (_ : BilPos ψ μ (ρ + 1) σ B) (F : A) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (z : ℂ), IsQuotInt ν μX μG (ρ + 1) (tInt ψ ν μ (ρ + 1) σ B F Φ.1) z ↔ FJInt ψ ν μX

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierJacobiTargetBilinearFormIntegralStatement

theorem **_root_.GlobalTheory.T.fourierJacobi_target_bilinear_form_integral** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (_ : 1 ≤ ρ) (_ : c = 2 * ρ) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (fj : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] SB (n → k) →ₗ[ℂ] ℂ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : Represents ψ ν ρ σ fj B) (_ : BilPos ψ μ (ρ + 1) σ B) (F : A) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (z : ℂ), IsQuotInt ν μX μG (ρ +

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingCharIndicatorDualLatticeCosets

theorem **inDual_add** {L : Submodule 𝒪[k] (n → k)} {η ξ : n → k} (h : InDual L η) (h' : InDual L ξ) : InDual L (η + ξ)

theorem **inDual_sub** {L : Submodule 𝒪[k] (n → k)} {η ξ : n → k} (h : InDual L η) (h' : InDual L ξ) : InDual L (η − ξ)

theorem **inDual_smul** {L : Submodule 𝒪[k] (n → k)} {η : n → k} (h : InDual L η) (s : 𝒪[k]) : InDual L (s • η)

theorem **isLocallyConstant_of_periodic** (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] {g : (n → k) → ℂ} (hg : ∀ η ξ, InDual L ξ → g (η + ξ) = g η) : IsLocallyConstant g

private theorem **secondCountableTopology_localField'** (K : Type) [Field K] [ValuativeRel K] [TopologicalSpace K] [IsNonarchimedeanLocalField K] : SecondCountableTopology K

theorem **integrable_of_periodic** (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] {g : (n → k) → ℂ} (hg : ∀ η ξ, InDual L ξ → g (η + ξ) = g η) (hs : ∀ η, g η ≠ 0 → InDual L (q.1 • η)) : Integrable g μ

theorem **dualInd_add** (L : Submodule 𝒪[k] (n → k)) (η ξ : n → k) (hξ : InDual L ξ) : dualInd L (η + ξ) = dualInd L η

theorem **dualInd_supp** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) (η : n → k) (h : dualInd L η ≠ 0) : InDual L (q.1 • η)

theorem **cosetInd_add** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) (f : Module.Dual 𝓀[k] (Red L)) (η ξ : n → k) (hξ : InDual L ξ) : cosetInd q L f (η + ξ) = cosetInd q L f η

theorem **inDual_of_lifts** {L : Submodule 𝒪[k] (n → k)} {F : n → k} {f : Module.Dual 𝓀[k] (Red L)} (h : Lifts L F f) : InDual L F

theorem **cosetInd_supp** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) (f : Module.Dual 𝓀[k] (Red L)) (η : n → k) (h : cosetInd q L f η ≠ 0) : InDual L (q.1 • η)

theorem **inner_add** (χ : AddChar k Circle) (hχ : HasConductor χ 1) {e : k} (he : e = 1 ∨ e = −1) [MeasurableSpace k] (μ : Measure (n → k)) (L : Submodule 𝒪[k] (n → k)) {a : (n → k) → ℂ} (ha : ∀ y, y ∉ L → a y = 0) (η ξ : n → k) (hξ : InDual L ξ) : ∫ y, a y * (χ (e * ((η + ξ) ·ᵥ y)) : ℂ) ∂μ = ∫ y, a y * (χ (e * (η ·ᵥ y)) : ℂ) ∂μ

theorem **fpair_charInd_dualInd_add_cosets** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k), (e = 1 ∨ e = −1) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (Q : Pair k n), fpair χ e μ (charInd (resChar χ hχ q) Q) (dualInd Q.lat + ∑ Q'' ∈ withLat Q.lat, cosetInd q Q''.lat Q''.red) = fpair χ e μ (vecInd Q.lat) (dualInd Q.lat)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingCharIndicatorDualLatticeCosetsStatement

theorem **_root_.GlobalTheory.T.fourierPairing_charIndicator_dualLattice_cosets** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k), (e = 1 ∨ e = −1) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (Q : Pair k n), fpair χ e μ (charInd (resChar χ hχ q) Q) (dualInd Q.lat + ∑ Q'' ∈ withLat Q.lat, cosetInd q Q''.lat Q''.red) = fpair χ e μ (vecInd Q.lat) (dualInd Q.lat)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingDualIndicatorAddNonzeroCosets

theorem **integral_vecInd_eq_zero** [MeasurableSpace k] [BorelSpace k] (χ : AddChar k Circle) (e : k) (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (L : Submodule 𝒪[k] (n → k)) (η : n → k) {y₀ : n → k} (hy₀ : y₀ ∈ L) (hne : χ (e * (η ·ᵥ y₀)) ≠ 1) : ∫ y, vecInd L y * (χ (e * (η ·ᵥ y)) : ℂ) ∂μ = 0

theorem **exists_lifts_value** (L : Submodule 𝒪[k] (n → k)) {f : Module.Dual 𝓀[k] (Red L)} (hf : f ≠ 0) {F : n → k} (hF : Lifts L F f) (b : 𝓀[k]) : ∃ y₀ ∈ L, ∃ s : 𝒪[k], (s : k) = F ·ᵥ y₀ ∧ residue 𝒪[k] s = b

theorem **integral_vecInd_coset** [MeasurableSpace k] [BorelSpace k] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k) (he : e = 1 ∨ e = −1) (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (L : Submodule 𝒪[k] (n → k)) {f : Module.Dual 𝓀[k] (Red L)} (hf : f ≠ 0) {F η : n → k} (hF : Lifts L F f) (hη : InDual L (η − q.inv • F)) : ∫ y, vecInd L y * (χ (e * (η ·ᵥ y)) : ℂ) ∂μ = 0

theorem **fpair_vecInd_dualInd_add_cosets** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle), HasConductor χ 1 → ∀ (e : k), (e = 1 ∨ e = −1) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k], fpair χ e μ (vecInd L) (dualInd L + ∑ Q ∈ withLat L, cosetInd q Q.lat Q.red) = fpair χ e μ (vecInd L) (dualInd L)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingDualIndicatorAddNonzeroCosetsStatement

theorem **_root_.GlobalTheory.T.fourierPairing_dualIndicator_add_nonzeroCosets** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle), HasConductor χ 1 → ∀ (e : k), (e = 1 ∨ e = −1) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k], fpair χ e μ (vecInd L) (dualInd L + ∑ Q ∈ withLat L, cosetInd q Q.lat Q.red) = fpair χ e μ (vecInd L) (dualInd L)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingPairCharIndicatorCosetIndicator

theorem **exists_lifts** [DecidableEq n] (P : Pair k n) : ∃ F, Lifts P.lat F P.red

theorem **inDual_of_lifts** (q : Unif k) {L : Submodule 𝒪[k] (n → k)} {f : Module.Dual 𝒦[k] (Red L)} {F F' : n → k} (hF : Lifts L F f) (hF' : Lifts L F' f) : InDual L (q.inv • F − q.inv • F')

theorem **cosetInd_add** (q : Unif k) {L : Submodule 𝒪[k] (n → k)} {f : Module.Dual 𝒦[k] (Red L)} {F : n → k} (hF : Lifts L F f) (η : n → k) : cosetInd q L f (η + q.inv • F) = dualInd L η

theorem **char_dual** (he : e = 1 ∨ e = −1) {L : Submodule 𝒪[k] (n → k)} {η y : n → k} (hη : InDual L η) (hy : y ∈ L) : χ (e ∗ (η ⬝ᵥ y)) = 1

theorem **char_coset** (q : Unif k) {ε : 𝒦[k]ˣ} (he : (e = 1 ∧ ε = 1) ∨ (e = −1 ∧ ε = −1)) {P : Pair k n} {F η y : n → k} (hF : Lifts P.lat F P.red) (hη : InDual P.lat η) (hy : y ∈ P.lat) : (χ (e ∗ ((η + q.inv • F) ⬝ᵥ y)) : ℂ) = resChar χ hχ q (ε ∗ P.ev y)

theorem **exists_char_ne** (q : Unif k) {ε : 𝒦[k]ˣ} (hε : ε = 1 ∨ ε = −1) {Q Q'' : Pair k n} (h : Q''.lat = Q.lat) (hne : Q'' ≠ ε • Q) : ∃ y ∈ Q.lat, resChar χ hχ q (−Q.ev y + ε ∗ Q''.ev y) ≠ 1

theorem **fpair_charInd_cosetInd** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k) (ε : 𝒦[k]ˣ) (he : (e = 1 ∧ ε = 1) ∨ (e = −1 ∧ ε = −1)) (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (Q Q'' : Pair k n) (h : Q''.lat = Q.lat) : (Q'' = ε • Q → fpair χ e μ (charInd (resChar χ hχ q) Q) (cosetInd q Q''.lat Q''.red) = fpair χ e μ (vecInd Q.lat) (dualInd Q.lat)) ∧ (Q'' ≠ ε • Q → fpair χ e μ (charInd (resChar χ hχ q) Q) (cosetInd q Q''.lat Q''.red) = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingPairCharIndicatorCosetIndicatorStatement

theorem **_root_.GlobalTheory.T.fourierPairing_pairCharIndicator_cosetIndicator** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k) (ε : 𝒦[k]ˣ), ((e = 1 ∧ ε = 1) ∨ (e = −1 ∧ ε = −1)) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (Q Q'' : Pair k n), Q''.lat = Q.lat → (Q'' = ε • Q → fpair χ e μ (charInd (resChar χ hχ q) Q) (cosetInd q Q''.lat Q''.red) = fpair χ e μ (vecInd Q.lat) (dualInd Q.lat)) ∧ (Q'' ≠ ε • Q → fpair χ e μ (charInd (resChar χ hχ q) Q) (cosetInd q Q''.lat Q''.red) = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingPairCharIndicatorDualLatticeIndica

theorem **fpair_charInd_dualInd** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k), (e = 1 ∨ e = −1) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (Q : Pair k n), fpair χ e μ (charInd (resChar χ hχ q) Q) (dualInd Q.lat) = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierPairingPairCharIndicatorDualLatticeIndicaStatement

theorem **_root_.GlobalTheory.T.fourierPairing_pairCharIndicator_dualLatticeIndicator_zero** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (χ : AddChar k Circle) (hχ : HasConductor χ 1) (e : k), (e = 1 ∨ e = −1) → ∀ (μ : Measure (n → k)) [μ.IsAddHaarMeasure] (Q : Pair k n), fpair χ e μ (charInd (resChar χ hχ q) Q) (dualInd Q.lat) = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierTransformCosetIndicatorEqFiniteColumnTran

theorem **hat_coset_box** [MeasurableSpace k] [BorelSpace k] {n : Type} [Fintype n] {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {ε : k} (hε : ε ∈ 𝒪[k]) (c : kˣ) (y₀ : n → k) {η : n → k} (hη : ∀ i, η i ∗ (c : k) ∈ 𝒪[k]) : hat ψ ν ε (cosetInd (box n c) (Submodule.Quotient.mk y₀)) η = ((nabs k c ^ Fintype.card n : ℝ) : ℂ) ∗ ψ (ε ∗ (η ⬝ᵥ y₀))

theorem **hat_coset_eq_ftY** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FourierTransformCosetIndicatorEqFiniteColumnTranStatement

theorem **_root_.GlobalTheory.T.fourierTransform_cosetIndicator_eq_finiteColumnTransform** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealCompletionLocallyPrincipal

theorem **exists_gen_semiLoc** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) {𝔟 : FractionalIdeal (𝒪 E)⁰ E} (h𝔟 : 𝔟 ≠ 0) : ∃ b : E, b ≠ 0 ∧ ∀ z, (∃ s ∉ v.asIdeal, algebraMap (𝒪 F) (𝒪 E) s • z ∈ (𝔟 : Submodule (𝒪 E) E)) ↔ b⁻¹ ∗ z ∈ semiLoc E v

theorem **exists_gen** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) {𝔟 : FractionalIdeal (𝒪 E)⁰ E} (h𝔟 : 𝔟 ≠ 0) : ∃ b : E, b ≠ 0 ∧ (∃ t : 𝒪 F, t ∉ v.asIdeal ∧ algebraMap F E (t : F) ∗ b ∈ 𝔟) ∧ ∀ c ∈ 𝔟, ∃ t : 𝒪 F, t ∉ v.asIdeal ∧ ∃ o : 𝒪 E, algebraMap F E (t : F) ∗ (b⁻¹ ∗ c) = algebraMap (𝒪 E) E o

theorem **inv_mem_integers** {v : Pl D} {t : 𝒪 D.host.F} (hv : t ∉ v.asIdeal) : algebraMap D.host.F (Fv D.host.F v) (t : D.host.F)⁻¹ ∈ v.adicCompletionIntegers D.host.F

theorem **smul_inv_mem** {v : Pl D} {t : 𝒪 D.host.F} (hv : t ∉ v.asIdeal) {α : Af D.host.F} (hα : α ∈ intAf D.host.F) (h0 : ∀ w, w ≠ v → evalV D.host.F w α = 0) : (t : D.host.F)⁻¹ • α ∈ intAf D.host.F ∧ ∀ w, w ≠ v → evalV D.host.F w ((t : D.host.F)⁻¹ • α) = 0

theorem **pairA_smul** {F : Type} [Field F] [NumberField F] (c : F) (α : Af F) : pairA F 0 (c • α) = c • pairA F 0 α

theorem **tmul_eq_of_ne** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (c : F) (hc : c ≠ 0) (α : Af F) (z : E) : pairA F 0 α ⊗ₜ[F] z = pairA F 0 (c⁻¹ • α) ⊗ₜ[F] (algebraMap F E c ∗ z)

theorem **tmul_eq** {v : Pl D} {t : 𝒪 D.host.F} (ht : t ∉ v.asIdeal) (α : Af D.host.F) (z : D.host.E) : pairA D.host.F 0 α ⊗ₜ[D.host.F] z = pairA D.host.F 0 ((t : D.host.F)⁻¹ • α) ⊗ₜ[D.host.F] (algebraMap D.host.F D.host.E (t : D.host.F) ∗ z)

theorem **gen_mem_idAt** {v : Pl D} {α : Af D.host.F} (h0 : ∀ w, w ≠ v → evalV D.host.F w α = 0) (z : D.host.E) : pairA D.host.F 0 α ⊗ₜ[D.host.F] z ∈ idAt D (· = v)

theorem **gen_mem_intAt** {v : Pl D} {α : Af D.host.F} (hα : α ∈ intAf D.host.F) (h0 : ∀ w, w ≠ v → evalV D.host.F w α = 0) (o : 𝒪 D.host.E) : pairA D.host.F 0 α ⊗ₜ[D.host.F] algebraMap (𝒪 D.host.E) D.host.E o ∈ intAt D (· = v)

theorem **intAt_zero'** (v : Pl D) : (0 : AE D) ∈ intAt D (· = v)

theorem **intAt_add'** {v : Pl D} {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a + b ∈ intAt D (· = v)

theorem **intAt_neg'** {v : Pl D} {a : AE D} (ha : a ∈ intAt D (· = v)) : -a ∈ intAt D (· = v)

theorem **incA_mul_tmul** (y : AF D.host.F) (e z : D.host.E) : incA D e * (y ⊗ₜ[D.host.F] z) = y ⊗ₜ[D.host.F] (e * z)

theorem **idV_gen** : ∀ (D : D7) (v : Pl D) (𝔞 : FI D), 𝔞 ≠ 0 → ∃ x : D.host.E, x ≠ 0 ∧ ∀ s, s ∈ idV 𝔞 v ↔ incA D x⁻¹ * s ∈ intAt D (· = v) ∧ s ∈ idAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealCompletionLocallyPrincipalStatement

theorem **_root_.GlobalTheory.T.fractionalIdealCompletion_locally_principal** : ∀ (D : D7) (v : Pl D) (𝔞 : FI D), 𝔞 ≠ 0 → ∃ x : D.host.E, x ≠ 0 ∧ ∀ s, s ∈ idV 𝔞 v ↔ incA D x⁻¹ * s ∈ intAt D (· = v) ∧ s ∈ idAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealCompletionPropertiesAtPlace

theorem **idV_props** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ v : Pl D, (∀ a, a ∈ idV (1 : FI D) v ↔ a ∈ intAt D (· = v)) ∧ (∀ 𝔞 𝔟 : FI D, 𝔞 ≠ 0 → 𝔟 ≠ 0 → ∀ s ∈ idAt D (· = v), (∀ a ∈ idV 𝔞 v, s * a ∈ idV 𝔟 v) ↔ s ∈ idV (𝔟 * 𝔞⁻¹) v) ∧ (∀ (𝔞 : FI D) (x : D.host.E), x ≠ 0 → ∀ s, s ∈ idV (FractionalIdeal.spanSingleton _ x * 𝔞) v ↔ incA D x⁻¹ * s ∈ idV 𝔞 v ∧ s ∈ idAt D (· = v)) ∧ (∀ n : ℤ, ∀ s, s ∈ idV (D.host.AA ^ n) v ↔ cA D s ∈ idV (D.host.AAbar ^ n) v) ∧ (∀ s ∈ idAt D (· = v), s ∈ idV (dI D) v ↔ ∀ o ∈ intAt D (· = v), psiE ψ (s * o) = 1) ∧ (∀ 𝔞 : FI D, 𝔞 ≠ 0 → ∃ g ∈ idV 𝔞 v, ∃ g' ∈ idAt D (· = v), (∀ s ∈ idAt D (· = v), g' * g * s = s) ∧ ∀ s, s ∈ idV 𝔞 v ↔ s ∈ idAt D (· = v) ∧ g' * s ∈ intAt D (· = v)) ∧ ∀ (S : HSp D) (L : Lat S) (y : Fin S.rank → AE D), y ∈ latA S L (· = v) ↔ (∀ i, y i ∈ idAt D (· = v)) ∧ ∀ s ∈ idV D.host.AA⁻¹ v, s • y ∈ latV S L (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealCompletionPropertiesAtPlaceStatement

theorem **_root_.GlobalTheory.T.fractionalIdealCompletion_properties_at_place** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ v : Pl D, (∀ a, a ∈ idV (1 : FI D) v ↔ a ∈ intAt D (· = v)) ∧ (∀ 𝔞 𝔟 : FI D, 𝔞 ≠ 0 → 𝔟 ≠ 0 → ∀ s ∈ idAt D (· = v), (∀ a ∈ idV 𝔞 v, s * a ∈ idV 𝔟 v) ↔ s ∈ idV (𝔟 * 𝔞⁻¹) v) ∧ (∀ (𝔞 : FI D) (x : D.host.E), x ≠ 0 → ∀ s, s ∈ idV (FractionalIdeal.spanSingleton _ x * 𝔞) v ↔ incA D x⁻¹ * s ∈ idV 𝔞 v ∧ s ∈ idAt D (· = v)) ∧ (∀ n : ℤ, ∀ s, s ∈ idV (D.host.AA ^ n) v ↔ cA D s ∈ idV (D.host.AAbar ^ n) v) ∧ (∀ s ∈ idAt D (· = v), s ∈ idV (dI D) v ↔ ∀ o ∈ intAt D (· = v), psiE ψ (s * o) = 1) ∧ (∀ 𝔞 : FI D, 𝔞 ≠ 0 → ∃ g ∈ idV 𝔞 v, ∃ g' ∈ idAt D (· = v), (∀ s ∈ idAt D (· = v), g' * g * s = s) ∧ ∀ s, s ∈ idV 𝔞 v ↔ s ∈ idAt D (· = v) ∧ g' * s ∈ intAt D (· = v)) ∧ ∀ (S : HSp D) (L : Lat S) (y : Fin S.rank → AE D), y ∈ latA S L (· = v) ↔ (∀ i, y i ∈ idAt D (· = v)) ∧ ∀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealGeneratedByOneAlmostEverywhere

theorem **exists_nat_mem_mul_integral** {E : Type} [Field E] [NumberField E] (I : FractionalIdeal (nonZeroDivisors (𝒪 E)) E) (hI : I ≠ 0) : ∃ N : ℕ, N ≠ 0 ∧ (N : E) ∈ I ∧ ∀ b ∈ I, ∃ o : 𝒪 E, (N : E) * b = (o : E)

theorem **aeGen** : ∀ (D : D7) (I : FractionalIdeal (nonZeroDivisors (𝒪 D.host.E)) D.host.E), I ≠ 0 → ∃ S : Finset (HeightOneSpectrum (𝒪 D.host.F)), ∀ v ∉ S, ∃ c : 𝒪 D.host.F, c ∉ v.asIdeal ∧ ((c : D.host.F) : D.host.E) ∈ I ∧ ∀ b ∈ I, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) * b = (o : D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealGeneratedByOneAlmostEverywhereStatement

theorem **_root_.GlobalTheory.T.fractionalIdeal_generated_by_one_almostEverywhere** : ∀ (D : D7) (I : FractionalIdeal (nonZeroDivisors (𝒪 D.host.E)) D.host.E), I ≠ 0 → ∃ S : Finset (HeightOneSpectrum (𝒪 D.host.F)), ∀ v ∉ S, ∃ c : 𝒪 D.host.F, c ∉ v.asIdeal ∧ ((c : D.host.F) : D.host.E) ∈ I ∧ ∀ b ∈ I, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) * b = (o : D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealHermitianSelfDualOfLocalGenerator

theorem **alg₁** {R : Type} [CommRing R] {u u' w w' e : R} (hU : u' * u = 1) (hC : w' * w = 1) (he : e * e = e) (c d h : R) : e * (c * d * h) = u * w * h * (u' * (e * d) * (w' * (e * c)))

theorem **alg₂** {R : Type} [CommRing R] {u u' : R} (hU : u' * u = 1) (w e d h : R) : u * w * h * (u' * (e * d)) = d * h * (w * e)

theorem **alg₃** {R : Type} [CommRing R] {u u' e : R} (hU : u' * u = 1) (he : e * e = e) (d y : R) : e * (d * y) = u' * (e * d) * (y * (u * e))

theorem **intAt_mul** {v : Pl D} {a b : AE D} (ha : a ∈ intAt D (· = v)) (hb : b ∈ intAt D (· = v)) : a * b ∈ intAt D (· = v)

theorem **cA_mem_intAt** {v : Pl D} {a : AE D} (ha : a ∈ intAt D (· = v)) : cA D a ∈ intAt D (· = v)

theorem **idem_mul_incA** (P : Pl D → Prop) (y : D.host.E) : idem D P * incA D y = pairA D.host.F 0 (ePl D P) ⊗ₜ[D.host.F] y

theorem **idem_mul_incA_mem** {𝔞 : FI D} {c : D.host.E} (hc : c ∈ 𝔞) (v : Pl D) : idem D (· = v) * incA D c ∈ idV 𝔞 v

theorem **mul_incA_mem** {v : Pl D} {𝔞 : FI D} {y : D.host.E} (hy : ∀ c ∈ 𝔞, ∃ b : 𝒪 D.host.E, (b : D.host.E) = y * c) {s : AE D} (hs : s ∈ idV 𝔞 v) : incA D y * s ∈ intAt D (· = v)

theorem **mul_cA_mem** {v : Pl D} {𝔞 : FI D} {y : D.host.E} (hy : ∀ c ∈ 𝔞, ∃ b : 𝒪 D.host.E, (b : D.host.E) = y * star c) {s : AE D} (hs : s ∈ idV 𝔞 v) : incA D y * cA D s ∈ intAt D (· = v)

theorem **exists_int_mul** (y : D.host.E) : ∃ t : 𝒪 D.host.F, t ≠ 0 ∧ ∃ b : 𝒪 D.host.E, (b : D.host.E) = algebraMap D.host.F D.host.E (t : D.host.F) * y

theorem **one_eq_sum** (T : Finset (Pl D)) : (1 : Af D.host.F) = ∑ v ∈ T, ePl D (· = v) + ePl D (· ∉ T)

theorem **exists_integer_of_forall** (y : D.host.E) (h : ∀ v : Pl D, idem D (· = v) * incA D y ∈ intAt D (· = v)) : ∃ b : 𝒪 D.host.E, (b : D.host.E) = y

theorem **idem_ne_zero** (v : Pl D) : idem D (· = v) ≠ 0

theorem **selfDual_of_local** : ∀ (D : D7) (h : D.host.E) (𝔞 : FI D) (u : (AE D)ˣ), (∀ x : Pl D, ∀ a ∈ idAt D (· = x), a ∈ idV 𝔞 x ↔ ((u⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) → (∀ x : Pl D, ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (u : AE D) * cA D (u : AE D) * incA D h * a ∈ intAt D (· = x)) → ∀ d : D.host.E, d ∈ 𝔞 ↔ ∀ c ∈ 𝔞, ∃ b : 𝒪 D.host.E, (b : D.host.E) = star c * d * h

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealHermitianSelfDualOfLocalGeneratorStatement

theorem **_root_.GlobalTheory.T.fractionalIdeal_hermitianSelfDual_of_local_generators** : ∀ (D : D7) (h : D.host.E) (𝔞 : FI D) (u : (AE D)ˣ), (∀ x : Pl D, ∀ a ∈ idAt D (· = x), a ∈ idV 𝔞 x ↔ ((u⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) → (∀ x : Pl D, ∀ a ∈ idAt D (· = x), a ∈ intAt D (· = x) ↔ (u : AE D) * cA D (u : AE D) * incA D h * a ∈ intAt D (· = x)) → ∀ d : D.host.E, d ∈ 𝔞 ↔ ∀ c ∈ 𝔞, ∃ b : 𝒪 D.host.E, (b : D.host.E) = star c * d * h

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealLocalGeneratorAtBasePrime

theorem **exists_gen_mul_of_isIntegral** {R S : Type*} [CommRing R] [CommRing S] [IsDedekindDomain S] [Algebra R S] [Algebra.IsIntegral R S] (hinj : Function.Injective (algebraMap R S)) (p : Ideal R) [p.IsPrime] (hp : p ≠ ⊥) (J : Ideal S) (hJ : J ≠ ⊥) : ∃ y ∈ J, y ≠ 0 ∧ ∃ c : R, c ∉ p ∧ ∀ j ∈ J, ∃ o : S, algebraMap R S c * j = y * o

theorem **fracIdeal_locGen** : ∀ (D : D7) (I : FractionalIdeal (nonZeroDivisors (𝒪 D.host.E)) D.host.E), I ≠ 0 → ∀ v : HeightOneSpectrum (𝒪 D.host.F), ∃ a : D.host.E, a ≠ 0 ∧ ∃ c : 𝒪 D.host.F, c ∉ v.asIdeal ∧ ((c : D.host.F) : D.host.E) * a ∈ I ∧ ∀ b ∈ I, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) * b = a * (o : D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealLocalGeneratorAtBasePrimeStatement

theorem **_root_.GlobalTheory.T.fractionalIdeal_localGenerator_at_basePrime** : ∀ (D : D7) (I : FractionalIdeal (nonZeroDivisors (𝒪 D.host.E)) D.host.E), I ≠ 0 → ∀ v : HeightOneSpectrum (𝒪 D.host.F), ∃ a : D.host.E, a ≠ 0 ∧ ∃ c : 𝒪 D.host.F, c ∉ v.asIdeal ∧ ((c : D.host.F) : D.host.E) * a ∈ I ∧ ∀ b ∈ I, ∃ o : 𝒪 D.host.E, ((c : D.host.F) : D.host.E) * b = a * (o : D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealLocallyPrincipalAdelic

theorem **GlobalTheory.FinPl.idV_principal_at** : ∀ (D : D7) (v : Pl D) (a : FI D), a ≠ 0 → ∃ g ∈ idV a v, ∃ g' ∈ idAt D (· = v), (∀ s ∈ idAt D (· = v), g' * g * s = s) ∧ ∀ s, s ∈ idV a v ↔ s ∈ idAt D (· = v) ∧ g' * s ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FractionalIdealLocallyPrincipalAdelicStatement

theorem **_root_.GlobalTheory.T.fractionalIdeal_locallyPrincipal_adelic** : ∀ (D : D7) (v : Pl D) (a : FI D), a ≠ 0 → ∃ g ∈ idV a v, ∃ g' ∈ idAt D (· = v), (∀ s ∈ idAt D (· = v), g' * g * s = s) ∧ ∀ s, s ∈ idV a v ↔ s ∈ idAt D (· = v) ∧ g' * s ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameColumnsUnitScaling

theorem **cA_sec_mul'** (p : SplitPt D k) (a : AE D) (t : k) : cA D (p.sec t) * a = cA D (p.sec (t * p.pr (cA D a)))

theorem **colMul_split** (q : SplitPt D k) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (i : Fin ρ) {a : AE D} (ha : a ∈ idAt D (· = q.v)) : colMul d.jX i a = (fun r => q.sec (q.pr a * q.pr (incA D (d.e r i)))) + (fun r => cA D (q.sec (q.pr (cA D a) * q.pr (cA D (incA D (d.e r i))))))

theorem **mul_unit_mem_iff** {b u : k} (hu : valuation k u = 1) : b * u ∈ 𝒪[k] ↔ b ∈ 𝒪[k]

theorem **colMul_mem_latV_iff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : GlobalTheory.FinPl.Lat S) (Y : Type) [Fintype Y] [DecidableEq Y] (F : Frame k S.rank Y) (i : Fin ρ) (y₁ y₂ : Y) (u₁ u₂ : k), IsSelfDualC S L → IntIn q S F L → valuation k u₁ = 1 → valuation k u₂ = 1 → (∀ r, F.β r y₁ = u₁ * q.pr (incA D (d.e r i))) → (∀ l, ∑ r, ∑ t, F.β r l * q.pr (cA D (incA D (HSpace.gram S r t))) * q.pr (cA D (incA D (d.e t i))) = if l = y₂ then u₂ else 0) → ∀ a : AE D, (GlobalTheory.FinPl.colMul d.jX i a ∈ GlobalTheory.FinPl.latV S L (· = q.v) ↔ a ∈ GlobalTheory.FinPl.intAt D (· = q.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameCoordsMemSelfDualLatticeCoordsIff

theorem **hT_frame** (D : D7) (S : HSp D) (u v : S.V) : S.hT u v = ∑ i, ∑ j, star (S.frame.equivFun u i) * S.gram i j * S.frame.equivFun v j

theorem **frameCoords_mem_selfDualLatticeCoords_iff** (D : D7) (S : HSp D) (u : S.V) : S.frame.equivFun u ∈ selfDualLatticeCoordsInHermitianFrame D S ↔ u ∈ S.L0

theorem **isSelfDualC_selfDualLatticeCoordsInFrame** (D : D7) (S : HSp D) : IsSelfDualC S (selfDualLatticeCoordsInHermitianFrame D S)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameFamilyCompactRepresentation

theorem **coeffSpan_finiteDimensional** {G W : Type} [Group G] [AddCommGroup W] [Module ℂ W] [FiniteDimensional ℂ W] (ρ : Representation ℂ G W) : FiniteDimensional ℂ (coeffSpan ρ)

theorem **coeffSpan_rtrans** {G W : Type} [Group G] [AddCommGroup W] [Module ℂ W] (ρ : Representation ℂ G W) (c : G) {h : G → ℂ} (hh : h ∈ coeffSpan ρ) : (fun k => h (k * c)) ∈ coeffSpan ρ

theorem **exists_space** {ι : Type} [Fintype ι] [DecidableEq ι] {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] {Γ : ι → Type} [∀ i, Group (Γ i)] (K : Subgroup G) (j : (∀ i, Γ i) →* K) (hjs : Function.Surjective j) (a : ∀ i, Γ i → G) (hj : ∀ i k, (j (Pi.mulSingle i k) : G) = a i k) (W : ι → Type) [∀ i, AddCommGroup (W i)] [∀ i, Module ℂ (W i)] [∀ i, FiniteDimensional ℂ (W i)] (ρ : ∀ i, Representation ℂ (Γ i) (W i)) : ∃ S : Submodule ℂ (K → ℂ), FiniteDimensional ℂ S ∧ (∀ f ∈ S, Continuous f) ∧ (∀ f ∈ S, ∀ k, (fun x => f (x * k)) ∈ S) ∧ ∀ φ : G → ℂ, Continuous φ → (∀ i x, (fun k => φ (x * a i k)) ∈ coeffSpan (ρ i)) → ∀ g, (fun k : K => φ (g * k)) ∈ S

theorem **GlobalTheory.typesFiniteSpace** : ∀ (D : D7) (θ : D.what₂.Tau → (Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ)), IsInfFam θ → ∀ (V : HSp D), ∀ (ε : D.what₂.Tau → Fin V.rank → Bool) (fr : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ), IsFrameFam V ε fr → ∀ K : Subgroup (GA D V), (∃ j : (∀ τ, Formalism.REAL.KG (ε τ)) →* K, Function.Surjective j ∧ ∀ τ k, (j (Pi.mulSingle τ k) : GA D V) = kAt θ V τ (ε τ) (fr τ) k) → ∀ (W : D.what₂.Tau → Type) [∀ τ, AddCommGroup (W τ)] [∀ τ, Module ℂ (W τ)] [∀ τ, FiniteDimensional ℂ (W τ)] (ρ : ∀ τ, Representation ℂ (Formalism.REAL.KG (ε τ)) (W τ)), ∃ S : Submodule ℂ (K → ℂ), FiniteDimensional ℂ S ∧ (∀ f ∈ S, Continuous f) ∧ (∀ f ∈ S, ∀ k, (fun x => f (x * k)) ∈ S) ∧ ∀ φ : GA D V → ℂ, Continuous φ → (∀ τ, HasTypeAt θ V τ (ε τ) (fr τ) (ρ τ) φ) → ∀ g, (fun k : K => φ (g * k)) ∈ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameGlInt

theorem **mat_int** {p : SplitPt D k} {S : HSp D} {L : Lat S} (hL : ∀ z, z ∈ L ↔ z ∈ intVec D S.rank) (β : Frame p S (Fin S.rank)) {Λ : Submodule 𝒪[k] (Fin S.rank → k)} (h : β.IsLat L Λ) (g : GL (Fin S.rank) k) (hg : g • Λ = Λ) (r s : Fin S.rank) : (β.b * g.val * β.b') r s ∈ 𝒪[k]

theorem **finM_mul** (V : HSp D) {n : ℕ} (hn : V.rank = n) (g h : GA D V) : GlobalTheory.AdForm.finM V hn (g * h) = GlobalTheory.AdForm.finM V hn g * GlobalTheory.AdForm.finM V hn h

theorem **prF_finM** (q : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (V : HSp D) {n : ℕ} (hn : V.rank = n) (g : GA D V) (r s : Fin n) : q.prF he (GlobalTheory.AdForm.finM V hn g r s) = (Formalism.L2.Theta.mt V g).map q.pr ((finCongr hn).symm r) ((finCongr hn).symm s)

theorem **finM_one** (V : HSp D) {n : ℕ} (hn : V.rank = n) : GlobalTheory.AdForm.finM V hn 1 = 1

theorem **exists_int** (p : SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (he : Formalism.L2.NTheta.IsIdemAt p.e PP) (S : HSp D) {n : ℕ} (hn : S.rank = n) (hgram : S.gram.reindex (finCongr hn) (finCongr hn) = 1) {L : Lat S} (hL : ∀ z, z ∈ L ↔ z ∈ intVec D S.rank) (β :

Frame p S (Fin S.rank)) {Λ : Submodule 𝒪[k] (Fin S.rank → k)} (hlat : β.IsLat L Λ) (g : GL (Fin S.rank) k) (hgl : g • Λ = Λ) : ∃ δ : GL (Fin n) (GlobalTheory.AdForm.EF D), δ.1 = GlobalTheory.AdForm.finM S hn (β.gl g) ∧ δ ∈ Formalism.UG.UPts (sigD D) (1 : Matrix (Fin n) (Fin n) D.host.E) (FiniteAdeleRing (𝒪 D.host.F) D.host.F) ∧ (∀ r r', δ.1 r r' ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) ∧ (∀ r r', (δ⁻¹).1 r r' ∈ Formalism.L2.Herm.intAf D.host.F D.host.E)

theorem **GlobalTheory.AdForm.frameGl_int** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k) (s : 𝓀[k]ˣ), g • Q j i = s • Q j i → ∃ δ : TC.GA B (B.R.N j), δ.1 = finM _ (rankN (I.rec7 l Z) j hj) ((β j i).gl g) ∧ δ ∈ Formalism.L2.Herm.UstdAf B.sp.cm (B.R.N j) ∧ (∀ r r', δ.1 r r' ∈ Formalism.L2.DIV.RInt B.host.F B.host.E) ∧ (∀ r r', (δ⁻¹).1 r r' ∈

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameImageTrivialOffLevelPrime

theorem **frame_trivial_off** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k) (δ : TC.GA B (B.R.N j)), δ.1 = finM _ (rankN (I.rec7 l Z) j hj) ((β j i).gl g) → ∀ w : HeightOneSpectrum (𝒪 B.fields.E), (w ∈ (B.primes X).vGs v w ∈ I.SQ v (w ∈ (B.primes X).vT u ∧ w ≠ v.1)) → ∀ r r', adAt B.host.F w ((δ⁻¹).1 r r') = if r = r' then 1 else 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameImageTrivialOffLevelPrimeStatement

theorem **_root_.GlobalTheory.T.frameImage_trivial_off_levelPrime** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k) (δ : TC.GA B (B.R.N j)), δ.1 = finM _ (rankN (I.rec7 l Z) j hj) ((β j i).gl g) → ∀ w : HeightOneSpectrum (𝒪 B.fields.E), (w ∈ (B.primes X).vGs v w ∈ I.SQ v (w ∈ (B.primes X).vT u ∧ w ≠ v.1)) → ∀ r r', adAt B.host.F w ((δ⁻¹).1 r r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameLatticeMembershipCriterion

theorem **sum_fin_two_mul_add_one** {M : Type*} [AddCommMonoid M] {ρ n : ℕ} (hn : n = 2 * ρ + 1) (f : Fin n → M) (a b : Fin ρ → M) (m : M) (ha : ∀ (i : Fin ρ) (j : Fin n), j.1 = i.1 → f j = a i) (hm : ∀ j : Fin n, j.1 = ρ → f j = m) (hb : ∀ (i : Fin ρ) (j : Fin n), j.1 + i.1 = 2 * ρ → f j = b i) : ∑ j, f j = ∑ i, a i + m + ∑ i, b i

theorem **mul_div_mem_integer_iff** {k : Type} [Field k] [ValuativeRel k] {c : k} (hc : valuation k c = 1) (w z : k) : w * (z / c) ∈ 𝒪[k] ↔ w * z ∈ 𝒪[k]

theorem **mul_mul_div_cancel_left'** {k : Type} [Field k] {c : k} (hc : c ≠ 0) (P X : k) : c * P * (X / c) = P * X

theorem **wShape_of_besFrame** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (ρ : ℕ) (h0 : 0 + 2 * ρ + 2 = D.R.m) (L : Lat (d.st 0).V), (∀ y : GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊛ Fin (G.s + 1) → G.k, (fun r => G.q.sec (((G.FY 0 (G.s + 1) (zero_add _)).β *ᵥ y) r)) ∈ latV (d.st 0).V L (· = G.q.v) ↔ (∀ a, y (Sum.inl a) ∈ 𝒪[G.k]) ∧ ∀ i : Fin (G.s + 1), ((((G.S.ϖ : G.kˣ) : G.k) ^ (i.1 + 1))⁻¹ * y (Sum.inr i)) ∈ 𝒪[G.k]) → WShape G.q (d.bes 0 ρ h0).d (d.bes 0 ρ h0).o (d.bes 0 ρ h0).ζ L (fun i => (((G.S.ϖ : G.kˣ) : G.k) ^ (i.1 + 1))⁻¹) (fun i => (((G.S.ϖ : G.kˣ) : G.k) ^ (2 * ρ - i.1 + 1))⁻¹) ((((G.S.ϖ : G.kˣ) : G.k) ^ (ρ + 1))⁻¹) 1 (d.x 0).L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameLatticeMembershipCriterionStatement

theorem **_root_.GlobalTheory.T.frame_lattice_membership_criterion** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (ρ : ℕ) (h0 : 0 + 2 * ρ + 2 = D.R.m) (L : Lat (d.st 0).V), (∀ y : GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊛ Fin (G.s + 1) → G.k, (fun r => G.q.sec (((G.FY 0 (G.s + 1) (zero_add _)).β *ᵥ y) r)) ∈ latV (d.st 0).V L (· = G.q.v) ↔ (∀ a, y (Sum.inl a) ∈ 𝒪[G.k]) ∧ ∀ i : Fin (G.s + 1), ((((G.S.ϖ : G.kˣ) : G.k) ^ (i.1 + 1))⁻¹ * y (Sum.inr i)) ∈ 𝒪[G.k]) → WShape G.q (d.bes 0 ρ h0).d (d.bes 0 ρ h0).o (d.bes 0 ρ h0).ζ L (fun i => (((G.S.ϖ : G.kˣ) : G.k) ^ (i.1 + 1))⁻¹) (fun i => (((G.S.ϖ : G.kˣ) : G.k) ^ (2 * ρ - i.1 + 1))⁻¹) ((((G.S.ϖ : G.kˣ) : G.k) ^ (ρ + 1))⁻¹) 1 (d.x 0).L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameMatrixDiagonalModLevelOfDiagonalAction

theorem **hatL_eq** (I : AutInst) : Formalism.L2.Herm.hatL (F

theorem **coord_zmod** (I : AutInst) (g : ℕ) {β : GL (Fin I.n) (I.E ⊗[I.F] Af I.F)} (hβ : IsFrame β I.L) {v : VA I} (hv : v ∈ mL I g) (i : Fin I.n) : ZMod' I g ((((β⁻¹ : GL (Fin I.n) (I.E ⊗[I.F] Af I.F)) : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Af I.F)) *ᵥ v) i)

theorem **diag_mod_in_frame** : ∀ (I : AutInst) (g : ℕ) (b : ℕ → VA I) (β : GL (Fin I.n) (I.E ⊗[I.F] Af I.F)) (Y : I.Gf) (d : Fin I.n → Formalism.L2.Herm.intAf I.F I.E), IsLiftFrame I g b β → (∀ y ∈ I.Lf, I.fmat Y *ᵥ y ∈ I.Lf) → (∀ y ∈ I.Lf, I.fmat Y⁻¹ *ᵥ y ∈ I.Lf) → (∀ j : Fin I.n, I.fmat Y *ᵥ b j.val - (d j).1 • b j.val ∈ mL I g) → ∃ x : GL (Fin I.n) (Formalism.L2.Herm.intAf I.F I.E), frGL β x = (Y : GL (Fin I.n) (I.E ⊗[I.F] Af I.F)) ∧ ∀ i j, ZMod' I g (((x : Matrix (Fin I.n) (Fin I.n) (Formalism.L2.Herm.intAf I.F I.E)) - Matrix.diagonal d) i j).1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameMatrixDiagonalModLevelOfDiagonalActionStatement

theorem **_root_.GlobalTheory.T.frame_matrix_diagonal_mod_level_of_diagonal_action** : ∀ (I : AutInst) (g : ℕ) (b : ℕ → VA I) (β : GL (Fin I.n) (I.E ⊗[I.F] Af I.F)) (Y : I.Gf) (d : Fin I.n → Formalism.L2.Herm.intAf I.F I.E), IsLiftFrame I g b β → (∀ y ∈ I.Lf, I.fmat Y *ᵥ y ∈ I.Lf) → (∀ y ∈ I.Lf, I.fmat Y⁻¹ *ᵥ y ∈ I.Lf) → (∀ j : Fin I.n, I.fmat Y *ᵥ b j.val - (d j).1 • b j.val ∈ mL I g) → ∃ x : GL (Fin I.n) (Formalism.L2.Herm.intAf I.F I.E), frGL β x = (Y : GL (Fin I.n) (I.E ⊗[I.F] Af I.F)) ∧ ∀ i j, ZMod' I g (((x : Matrix (Fin I.n) (Fin I.n) (Formalism.L2.Herm.intAf I.F I.E)) - Matrix.diagonal d) i j).1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameMatrixWeilPairActionLeftInv

theorem **mt_one'** {D : D7} (V : HSp D) : Formalism.L2.Theta.mt V 1 = 1

theorem **finE_inv_val'** {D : D7} (V : HSp D) (g : GA D V) : ((Lev.finE V g)⁻¹).1 = (Formalism.L2.Theta.mt V g⁻¹).map (toEAf D)

theorem **conj_inv_val'** {D : D7} {n : ℕ} (β y : GL (Fin n) (Lev.EAf D)) : ((β⁻¹ * y * β)⁻¹ : GL (Fin n) (Lev.EAf D)).1 = (β⁻¹).1 * (y⁻¹).1 * β.1

theorem **frMapO_actO_left** : ∀ (D : D7) (S T : HSp D) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (g : GA D S) (x : WO S T), frMapO S T βS βT (actO S T g 1 x) = HMul.hMul (frMapO S T βS βT x) ((βS⁻¹ * Lev.finE S g * βS)⁻¹ : GL (Fin S.rank) (Lev.EAf D)).val

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameMatrixWeilPairActionLeftInvStatement

theorem **_root_.GlobalTheory.T.frameMatrix_weilPairAction_left_inv** : ∀ (D : D7) (S T : HSp D) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (g : GA D S) (x : WO S T), frMapO S T βS βT (actO S T g 1 x) = HMul.hMul (frMapO S T βS βT x) ((βS⁻¹ * Lev.finE S g * βS)⁻¹ : GL (Fin S.rank) (Lev.EAf D)).val

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameValuesPolynomialDetStableHomogeneous

theorem **evalAt_eq** (y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : evalAt y F = MvPolynomial.eval (fun x : ι × Fin a × Fin a => y x.1 x.2.1 x.2.2) F

theorem **evalAt_frDet** (y : ι → Matrix (Fin a) (Fin a) ℂ) (τ : ι) : evalAt y (Formalism.L2.Ker.frDet ℂ a τ) = (y τ).det

theorem **evalAt_mul_prod** (y : ι → Matrix (Fin a) (Fin a) ℂ) (P : PY ι a) (n : ι → ℕ) : evalAt y (P * ∏ τ, (Formalism.L2.Ker.frDet ℂ a τ : PY ι a) ^ n τ) = evalAt y P * ∏ τ, (y τ).det ^ n τ

theorem **fnOf_one_apply** {G : ι → Matrix (Fin a) (Fin a) ℂ} (s : ι → ℤ) (P : PY ι a) (y : Frames G) : fnOf G s 1 P y = (∏ τ, (y.1 τ).det ^ s τ) * evalAt y.1 P

theorem **evalAt_lft** (g y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : evalAt y (lft g F) = evalAt (fun τ => g τ * y τ) F

theorem **shift_eq** {G : ι → Matrix (Fin a) (Fin a) ℂ} (lam : ι → DomWeight a) {n : ι → ℕ} {s : ι → ℤ} (hns : ∀ τ, (n τ : ℤ) = shiftOf lam τ + s τ) (P : PY ι a) (y : Frames G) : evalAt y.1 (P * ∏ τ, (Formalism.L2.Ker.frDet ℂ a τ : PY ι a) ^ n τ) * ∏ τ, (y.1 τ).det ^ (-s τ) = fnOf G (shiftOf lam) 1 P y

theorem **mem_valC_iff** (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a) (n : ι → ℕ) (s : ι → ℤ) (hns : ∀ τ, (n τ : ℤ) = shiftOf lam τ + s τ) (Φ : Frames G → ℂ) : Φ ∈ ValC G lam ↔ ∃ F ∈ Vn lam n, ∀ y, Φ y = evalAt y.1 F * ∏ τ, (y.1 τ).det ^ (-s τ)

theorem **nOf_eq** (lam : ι → DomWeight a) (τ : ι) : (nOf lam τ : ℤ) = shiftOf lam τ + sOf lam τ

theorem **valC_eq_map** (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a) : ValC G lam = (Vn lam (nOf lam)).map (fnOf G (fun τ => -sOf lam τ) 1)

theorem **finiteDimensional_valC** (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a) : FiniteDimensional ℂ (ValC G lam)

theorem **valC_stable** (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a) (Φ : Frames G → ℂ) (hΦ : Φ ∈ ValC G lam) (g : ι → Matrix (Fin a) (Fin a) ℂ) (Φ' : Frames G → ℂ) (hΦ' : ∀ y y', IsTr g y y' → Φ' y = Φ y') (hg : IsInv g) : Φ' ∈ ValC G lam

theorem **natW_anti'** (l : DomWeight a) : Antitone l.natW

theorem **isHomogeneous_of_mem_delSq** (l : DomWeight a) {f : MvPolynomial (Fin a × Fin a) ℂ} (hf : f ∈ delSq ℂ a l) : f.IsHomogeneous (∑ r, l.natW r)

theorem **eval_smul_of_isHomogeneous** {σ : Type} {f : MvPolynomial σ ℂ} {d : ℕ} (hf : f.IsHomogeneous d) (c : ℂ) (v : σ → ℂ) : MvPolynomial.eval (c • v) f = c ^ d * MvPolynomial.eval v f

theorem **sum_w_eq** (l : DomWeight a) : ∑ i, l.w i = ∑ i, ((l.natW i : ℕ) : ℤ) + (a : ℤ) * l.detShift

theorem **evalAt_scale_cSpan** (lam : ι → DomWeight a) (z : ι → ℂ) (y y' : ι → Matrix (Fin a) (Fin a) ℂ) (hy : ∀ τ, y' τ = z τ • y τ) (P : PY ι a) (hP : P ∈ CSpan lam) : evalAt y' P = (∏ τ, z τ ^ (∑ i, (lam τ).natW i)) * evalAt y P

theorem **valC_homog** (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a) (Φ : Frames G → ℂ) (hΦ : Φ ∈ ValC G lam) (z : ι → ℂ) (y y' : Frames G) (hy : ∀ τ, y'.1 τ = z τ • y.1 τ) : Φ y' = (∏ τ, z τ ^ (∑ i, (lam τ).w i)) * Φ y

theorem **GlobalTheory.KJInf.valC_poly** : ∀ (ι : Type) [Fintype ι] (a : ℕ) (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a), (∀ (n : ι → ℕ) (s : ι → ℤ), (∀ τ, (n τ : ℤ) = shiftOf lam τ + s τ) → ∀ Φ : Frames G → ℂ, Φ ∈ ValC G lam ↔ ∃ F ∈ Vn lam n, ∀ y, Φ y = evalAt y.1 F * ∏ τ, (y.1 τ).det ^ (-s τ)) ∧ FiniteDimensional ℂ (ValC G lam) ∧ (∀ Φ ∈ ValC G lam, ∀ g : ι → Matrix (Fin a) (Fin a) ℂ, ∀ Φ' : Frames G → ℂ, (∀ y y', IsTr g y y' → Φ' y = Φ y') → IsInv g → Φ' ∈ ValC G lam) ∧ ∀ Φ ∈ ValC G lam, ∀ (z : ι → ℂ) (y y' : Frames G), (∀ τ, y'.1 τ = z τ • y.1 τ) → Φ y' = (∏ τ, z τ ^ (∑ i, (lam τ).w i)) * Φ y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameValuesPolynomialDetStableHomogeneousStatement

theorem **_root_.GlobalTheory.T.frame_values_polynomial_det_stable_homogeneous** : ∀ (ι : Type) [Fintype ι] (a : ℕ) (G : ι → Matrix (Fin a) (Fin a) ℂ) (lam : ι → DomWeight a), (∀ (n : ι → ℕ) (s : ι → ℤ), (∀ τ, (n τ : ℤ) = shiftOf lam τ + s τ) → ∀ Φ : Frames G → ℂ, Φ ∈ ValC G lam ↔ ∃ F ∈ Vn lam n, ∀ y, Φ y = evalAt y.1 F * ∏ τ, (y.1 τ).det ^ (-s τ)) ∧ FiniteDimensional ℂ (ValC G lam) ∧ (∀ Φ ∈ ValC G lam, ∀ g : ι → Matrix (Fin a) (Fin a) ℂ, ∀ Φ' : Frames G → ℂ, (∀ y y', IsTr g y y' → Φ' y = Φ y') → IsInv g → Φ' ∈ ValC G lam) ∧ ∀ Φ ∈ ValC G lam, ∀ (z : ι → ℂ) (y y' : Frames G), (∀ τ, y'.1 τ = z τ • y.1 τ) → Φ y' = (∏ τ, z τ ^ (∑ i, (lam τ).w i)) * Φ y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameVectorLatticeCriterion

theorem **mulVec_sec_split** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) {n : ℕ} (g : Matrix (Fin n) (Fin n) (AE D)) (w : Fin n → k) : g *ᵥ (fun i => q.sec (w i)) = fun i => q.sec ((g.map q.pr *ᵥ w) i)

theorem **latV_normalFrame_zero** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (c : ℕ) (hc : 0 + c = G.s + 1) (L : Lat (d.st 0).V), Formalism.L2.DIV.IsFrame (d.st 0).β (d.st 0).Λ → Moves (d.st 0).V (d.t 0) (d.st 0).Λ L (· = G.q.v) → ∀ y : GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin c → G.k, (fun r => G.q.sec (((G.FY 0 c hc).β *ᵥ y) r)) ∈ latV (d.st 0).V L (· = G.q.v) ↔ (∀ a, y (Sum.inl a) ∈ 𝒪[G.k]) ∧ ∀ i : Fin c, ((((G.S.ϖ : G.kˣ) : G.k) ^ (i.1 + 1))⁻¹ * y (Sum.inr i)) ∈ 𝒪[G.k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrameVectorLatticeCriterionStatement

theorem **_root_.GlobalTheory.T.frame_vector_lattice_criterion** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (c : ℕ) (hc : 0 + c = G.s + 1) (L : Lat (d.st 0).V), Formalism.L2.DIV.IsFrame (d.st 0).β (d.st 0).Λ → Moves (d.st 0).V (d.t 0) (d.st 0).Λ L (· = G.q.v) → ∀ y : GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin c → G.k, (fun r => G.q.sec (((G.FY 0 c hc).β *ᵥ y) r)) ∈ latV (d.st 0).V L (· = G.q.v) ↔ (∀ a, y (Sum.inl a) ∈ 𝒪[G.k]) ∧ ∀ i : Fin c, ((((G.S.ϖ : G.kˣ) : G.k) ^ (i.1 + 1))⁻¹ * y (Sum.inr i)) ∈ 𝒪[G.k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FrsFBaseEqSelf

theorem **frsF_base_eq_self** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {u : B.Word} {a : I.ι} {Ch₀ : ℕ → Type} (A : TwDat I l Z u a Ch₀), A.Is → I.Ok → ∀ (P : BadPr I a), P.1 ∉ (B.primes X).vGs → ∀ (i : ℕ), i ≤ B.R.m → ∀ (β : Ch₀ i), (∀ v : HeightOneSpectrum (𝒪 B.host.F), ((B.T u : ℕ) : 𝒪 B.host.F) ∉ v.asIdeal → Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FubiniCompactQuotient

theorem **fubini_compact_quotient** : ∀ (H : Type) [Group H] [TopologicalSpace H] [IsTopologicalGroup H] [LocallyCompactSpace H] [SecondCountableTopology H] [T2Space H] [MeasurableSpace H] [BorelSpace H] (μ : Measure H) [μ.IsHaarMeasure] (Γ : Subgroup H), Countable Γ → DiscreteTopology Γ → (∃ K : Set H, IsCompact K ∧ ∀ g, ∃ γ ∈ Γ, γ * g ∈ K) → ∀ 𝓕 : Set H, IsFundamentalDomain Γ 𝓕 μ → ∀ (X : Type) [MeasurableSpace X] (m : Measure X) [SFinite m] (f : H → X → ℂ), Measurable (Function.uncurry f) → (∀ γ ∈ Γ, ∀ h x, f (γ * h) x = f h x) → (∀ K : Set H, IsCompact K → ∃ B : NNReal, ∀ h ∈ K, ∫⁻ x, ‖f h x‖ₑ ∂m ≤ B) → Integrable (Function.uncurry f) ((μ.restrict 𝓕).prod m)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FullOfSpanEqTop

theorem **exists_smul_mem** {E : Type} [Field E] [NumberField E] {V : Type} [AddCommGroup V] [Module E V] [Module (𝒪 E) V] [IsScalarTower (𝒪 E) E V] (N : Submodule (𝒪 E) V) {x : V} (hx : x ∈ Submodule.span E (N : Set V)) : ∃ d : 𝒪 E, d ≠ 0 ∧ d • x ∈ N

theorem **exists_nat_dvd** {E : Type} [Field E] [NumberField E] (d : 𝒪 E) (hd : d ≠ 0) : ∃ m : ℕ, 0 < m ∧ d ∣ (m : 𝒪 E)

theorem **full_of_span_eq_top** : ∀ {E : Type} [inst : Field E] [NumberField E] {n : ℕ} (N : Submodule (NumberField.RingOfIntegers E) (Formalism.L2.Col.Sp E n)), Submodule.span E (↑N : Set (Formalism.L2.Col.Sp E n)) = ⊤ → GlobalTheory.ThetaMove.Full N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FunctionalValuesIntegralVectorsOfIntegerVectors

theorem **mem_max_iff_val_lt_one** (x : 𝒪[k]) : x ∈ IsLocalRing.maximalIdeal 𝒪[k] ↔ valuation k (x : k) < 1

theorem **pr_incA_int_mem** (q : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) (x : 𝒪 D.host.E) : q.pr (incA D (x : D.host.E)) ∈ 𝒪[k]

theorem **exists_int_approx** (q : SplitPt D k) {PP : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt q.e PP) {u : k} (hu : u ∈ 𝒪[k]) : ∃ z : 𝒪 D.host.E, valuation k (u - q.pr (incA D (z : D.host.E))) < 1

theorem **hasVal_of_int_vectors** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (PP : HeightOneSpectrum (𝒪 D.host.E)), Formalism.L2.NTheta.IsIdemAt p.e PP → ∀ (m : ℕ) (b b' : Matrix (Fin m) (Fin m) k), b * b' = 1 → b' * b = 1 → ∀ P : Pair k (Fin m), (∀ x, x ∈ P.lat ↔ ∀ r, (b *ᵥ x) r ∈ 𝒪[k]) → ∀ c₀ : 𝒪 D.host.E, (∀ z : Fin m → D.host.E, (∀ r, IsIntegral ℤ (z r)) → ∀ r : Fin m, r.1 + 1 = m → ∃ s : 𝒪[k], (s : k) = p.pr (incA D ((c₀ : D.host.E) * z r)) ∧ P.HasVal (b' *ᵥ fun r' => p.pr (incA D (z r'))) (IsLocalRing.residue 𝒪[k] s)) → ∀ u : Fin m → k, (∀ r, u r ∈ 𝒪[k]) → ∀ r0 : Fin m, r0.1 + 1 = m → ∃ s : 𝒪[k], (s : k) = p.pr (incA D (c₀ : D.host.E)) * u r0 ∧ P.HasVal (b' *ᵥ u) (IsLocalRing.residue 𝒪[k] s)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.FunctionalValuesIntegralVectorsOfIntegerVectorsStatement

theorem **_root_.GlobalTheory.T.functional_values_integral_vectors_of_integer_vectors** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (PP : HeightOneSpectrum (𝒪 D.host.E)), Formalism.L2.NTheta.IsIdemAt p.e PP → ∀ (m : ℕ) (b b' : Matrix (Fin m) (Fin m) k), b * b' = 1 → b' * b = 1 → ∀ P : Pair k (Fin m), (∀ x, x ∈ P.lat ↔ ∀ r, (b *ᵥ x) r ∈ 𝒪[k]) → ∀ c₀ : 𝒪 D.host.E, (∀ z : Fin m → D.host.E, (∀ r, IsIntegral ℤ (z r)) → ∀ r : Fin m, r.1 + 1 = m → ∃ s : 𝒪[k], (s : k) = p.pr (incA D ((c₀ : D.host.E) * z r)) ∧ P.HasVal (b' *ᵥ fun r' => p.pr (incA D (z r'))) (IsLocalRing.residue 𝒪[k] s)) → ∀ u : Fin m → k, (∀ r, u r ∈ 𝒪[k]) → ∀ r0 : Fin m, r0.1 + 1 = m → ∃ s : 𝒪[k], (s : k) = p.pr (incA D (c₀ : D.host.E)) * u r0 ∧ P.HasVal (b' *ᵥ u) (IsLocalRing.residue 𝒪[k] s)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GaussianIsSchwartzFunction

theorem **norm_iteratedFDeriv_id_le_one_add** (j : ℕ) (x : E) : ‖iteratedFDeriv ℝ j (id : E → E) x‖ ≤ 1 + ‖x‖

theorem **norm_iteratedFDeriv_diag_le** (B : E →L[ℝ] E →L[ℝ] ℝ) (n : ℕ) (x : E) : ‖iteratedFDeriv ℝ n (fun y => B y y) x‖ ≤ ‖B‖ * (2 ^ n * (1 + ‖x‖) ^ 2)

theorem **norm_iteratedFDeriv_rexp** (i : ℕ) (y : ℝ) : ‖iteratedFDeriv ℝ i Real.exp y‖ = Real.exp y

theorem **exists_bound_poly_mul_exp** (c : ℝ) (hc : 0 < c) (N : ℕ) : ∃ C, ∀ r : ℝ, 0 ≤ r → (1 + r) ^ N * Real.exp (-(c * r ^ 2)) ≤ C

theorem **exists_coercive** [FiniteDimensional ℝ E] (B : E →L[ℝ] E →L[ℝ] ℝ) (hB : ∀ x, x ≠ 0 → B x x < 0) : ∃ c > 0, ∀ x, B x x ≤ -(c * ‖x‖ ^ 2)

theorem **exists_schwartz_exp_diag** [FiniteDimensional ℝ E] (B : E →L[ℝ] E →L[ℝ] ℝ) (hB : ∀ x, x ≠ 0 → B x x < 0) : ∃ f : SchwartzMap E ℂ, ∀ x, f x = ((Real.exp (B x x) : ℝ) : ℂ)

theorem **trFormC_apply** (w : ι → ℝ) (M : ι → Matrix (Fin m) (Fin m) ℂ) (x y : ι → Fin m → Fin n → ℂ) : trFormC w M x y = trForm w M x y

theorem **trace_conjTranspose_mul_mul** (M : Matrix (Fin m) (Fin m) ℂ) (Z : Matrix (Fin m) (Fin n) ℂ) : Matrix.trace (Zᴴ * M * Z) = ∑ j, star (fun a => Z a j) ⬝ᵥ (M *ᵥ fun a => Z a j)

theorem **re_trace_nonneg** {M : Matrix (Fin m) (Fin m) ℂ} (hM : M.PosDef) (Z : Matrix (Fin m) (Fin n) ℂ) : 0 ≤ (Matrix.trace (Zᴴ * M * Z)).re

theorem **re_trace_pos** {M : Matrix (Fin m) (Fin m) ℂ} (hM : M.PosDef) {Z : Matrix (Fin m) (Fin n) ℂ} (hZ : Z ≠ 0) : 0 < (Matrix.trace (Zᴴ * M * Z)).re

theorem **trForm_self_neg** (w : ι → ℝ) (hw : ∀ i, 0 < w i) (M : ι → Matrix (Fin m) (Fin m) ℂ) (hM : ∀ i, (M i).PosDef) (x : ι → Fin m → Fin n → ℂ) (hx : x ≠ 0) : trForm w M x x < 0

theorem **trace_conjTranspose_mul_mul_real** {M : Matrix (Fin m) (Fin m) ℂ} (hM : M.IsHermitian) (Z : Matrix (Fin m) (Fin n) ℂ) : (((Matrix.trace (Zᴴ * M * Z)).re : ℝ) : ℂ) = Matrix.trace (Zᴴ * M * Z)

theorem **exists_schwartz_gauss** : ∀ (D : D7) (n : ℕ) (V : HSp D) (l : D.what₂.Tau → ℝ), (∀ τ, l τ ≠ 0) → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∃ f : SchwartzMap (D.what₂.Tau → Fin V.rank → Fin n → ℂ) ℂ, ∀ z, f z = gauss V l s (fun τ => Matrix.of (z τ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GaussianIsSchwartzFunctionStatement

theorem **_root_.GlobalTheory.T.gaussian_is_schwartz_function** : ∀ (D : D7) (n : ℕ) (V : HSp D) (l : D.what₂.Tau → ℝ), (∀ τ, l τ ≠ 0) → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∃ f : SchwartzMap (D.what₂.Tau → Fin

V.rank → Fin n → ℂ) ℂ, ∀ z, f z = gauss V l s (fun τ => Matrix.of (z τ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GaussianReadingFiniteRationalHeisenbergTranslate

theorem **realPt_eq_iota** {θ : InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (hσ : GlobalTheory.FockCompare.Chart θ σ) (S T : HSp D) (Z : RW T S) : realPt θ (m

theorem **embH_toHom_apply** (S T : HSp D) (δ : ℂ) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (τ : D.what₂.Tau) (i : Fin T.rank) (j : Fin S.rank) : embH (tauE D) δ (toHom S T rfl rfl γ) τ i j = δ * tauE D τ (γ i j)

theorem **term_eq** (k sc : ℝ) (δ w g : ℂ) (hδ : δ ≠ 0) (hsc : sc * ‖δ‖ ^ 2 = 2 * k) : 2 * (Complex.I * (k : ℂ) * ((starRingEnd ℂ) g * -(δ⁻¹ * w))).re = ((sc : ℂ) * (w * (starRingEnd ℂ) (δ * g))).im

theorem **HV_conj** {ι : Type} [Fintype ι] {a b : ℕ} (sc : ℝ) (z w : VV ι a b) : HV sc w z = starRingEnd ℂ (HV sc z w)

theorem **omO_eq_im** (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (δ : ℂ) (sc : ℝ) (hP : StdPair θ S T δ sc ψ) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (z : VCx D.what₂.Tau S.rank T.rank) : omO S T (fun τ i j => -(δ⁻¹ * z τ i j)) (fun τ i j => tauE D τ (γ i j)) = (HV sc z (embH (tauE D) δ (toHom S T rfl rfl γ))).im

theorem **xiN_heisO_finOf** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ f, IsThetaFunO ψ S T f → ∀ γ v, GlobalTheory.AdTh.IsFinOf S T γ v → xiN θ S T hS hT δ sc (heisO ψ S T v f) = Formalism.L2.Th1.H.opU (HV sc) (embH (tauE D) δ (toHom S T hS hT γ)) (xiN θ S T hS hT δ sc f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GaussianReadingFiniteRationalHeisenbergTranslateStatement

theorem **_root_.GlobalTheory.T.gaussianReading_finiteRationalHeisenbergTranslate** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ f, IsThetaFunO ψ S T f → ∀ γ v, GlobalTheory.AdTh.IsFinOf S T γ v → xiN θ S T hS hT δ sc (heisO ψ S T v f) = Formalism.L2.Th1.H.opU (HV sc) (embH (tauE D) δ (toHom S T hS hT γ)) (xiN θ S T hS hT δ sc f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GaussianTwistedRealReadingEntireAndInjective

theorem **eq_of_comp** {θ : InfFam D} (hθ : IsInfFam θ) {m n : ℕ} (w w' : Mx D m n) (hw : w.map (finPart D) = 0) (hw' : w'.map (finPart D) = 0) (h : ∀ τ, compAt θ τ w = compAt θ τ w') : w = w'

theorem **realPt_eq** {θ : InfFam D} (hθ : IsInfFam θ) {m n : ℕ} (z : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ) (w : Mx D m n) (hw : w.map (finPart D) = 0) (hz : ∀ τ, compAt θ τ w = z τ) : realPt θ z = w

theorem **hAt_one** (S T : HSp D) (hS : S.gram = 1) (hT : T.gram = 1) (τ : D.what₂.Tau) (x y : Matrix (Fin T.rank) (Fin S.rank) ℂ) : GlobalTheory.VacUnique.hAt S T τ x y = ∑ i, ∑ j, x i j * starRingEnd ℂ (y i j)

theorem **xiN_eq_zero** (θ : InfFam D) (hθ : IsInfFam θ) (S T : HSp D) {a b : ℕ} (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ) (hδ : δ ≠ 0) (ψ : AddChar (AF D.host.F) ℂ) (f : WO S T → ℂ) (hf : IsThetaFunO ψ S T f) (h0 : xiN θ S T hS hT δ sc f = 0) : f = 0

theorem **key_sum** {ι : Type} [Fintype ι] (r0 k : ℝ) (s : ι → ℂ) : ((∑ τ, 2 * (Complex.I * (r0 : ℂ) * (k : ℂ) * s τ).re : ℝ) : ℂ) + Complex.I * ((∑ τ, 2 * (Complex.I * (r0 : ℂ) * (k : ℂ) * (-Complex.I * s τ)).re : ℝ) : ℂ) = 2 * Complex.I * ((r0 * k : ℝ) : ℂ) * ∑ τ, s τ

theorem **scale_eq** (x y sc : ℝ) (hy : y ≠ 0) (h : sc * y = 2 * x) : x * y⁻¹ = sc / 2

theorem **xiN_differentiable** (θ : InfFam D) (S T : HSp D) {a b : ℕ} (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ) (ψ : AddChar (AF D.host.F) ℂ) (hP : StdPair θ S T δ sc ψ) (f : WO S T → ℂ) (hf : IsHol θ S T ψ f) : Differentiable ℂ (xiN θ S T hS hT δ sc f)

theorem **xiN_entire_and_injective** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → (∀ f, IsHol θ S T ψ f → Differentiable ℂ (xiN θ S T hS hT δ sc f)) ∧ ∀ f, IsThetaFunO ψ S T f → xiN θ S T hS hT δ sc f = 0 → f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GaussianTwistedRealReadingEntireAndInjectiveStatement

theorem **_root_.GlobalTheory.T.gaussianTwistedRealReading_entire_and_injective** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → (∀ f, IsHol θ S T ψ f → Differentiable ℂ (xiN θ S T hS hT δ sc f)) ∧ ∀ f, IsThetaFunO ψ S T f → xiN θ S T hS hT δ sc f = 0 → f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GelfandTransposeEqOfInvariantWords

theorem **gelfA_conj** (G : Fin n → Fin n → A) (P Q : Matrix (Fin n) (Fin n) ℂ) (hQP : Q * P = 1) (k : ℕ) : gelfA (fun a b => ∑ c, ∑ d, (P a c * Q d b) • G c d) k = gelfA G k

theorem **gelfA_swap** (E : Fin n → Fin n → A) (k : ℕ) : gelfA (fun a b => E b a) k = ∑ idx : Fin (k + 1) → Fin n, (List.ofFn fun a : Fin (k + 1) => E (idx a) (idx (a + 1))).reverse.prod

theorem **prod_ofFn_neg** : ∀ (m : ℕ) (g : Fin m → A), (List.ofFn fun a => -g a).prod = (-1 : ℂ) ^ m • (List.ofFn g).prod | 0, _ => by simp | m + 1, g => by rw [List.ofFn_succ, List.ofFn_succ, List.prod_cons, List.prod_cons, prod_ofFn_neg m fun a => g a.succ, pow_succ] simp theorem sum_words (φ : Matrix (Fin n) (Fin n) ℂ →ₗ[ℂ] A) (M : Matrix (Fin n) (Fin n) ℂ) (hM : IsUnit M.det) (k : ℕ) : ∑ idx : Fin (k + 1) → Fin n, (List.ofFn fun a : Fin (k + 1) => φ (-(M * (Matrix.single (idx a) (idx (a + 1)) 1)ᵀ * M⁻¹))).prod = (-1 : ℂ) ^ (k + 1) • ∑ idx : Fin (k + 1) → Fin n, (List.ofFn fun a : Fin (k + 1) => φ (Matrix.single (idx a) (idx (a + 1)) 1)).reverse.prod

theorem **wordDer_eq** {D : D7} {θ : InfFam D} (hθ : IsInfFam θ) {V : HSp D} (τ : D.what₂.Tau) (w : List (Matrix (Fin V.rank) (Fin V.rank) ℂ)) (f : SmFun θ V) : wordDer θ V τ w f.1 = ((w.map (derEnd hθ τ)).prod f).1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GenValAddSegL

theorem **map_coe_segL** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ϖ : kˣ) (hϖ : valuation k (ϖ : k) = uniformizer k) (z : ℂˣ) (t : ℕ) : (segL ϖ z t).map (fun b : ℂˣ => (b : ℂ)) = (Finset.range t).val.map fun i : ℕ => (z : ℂ) * ((Real.sqrt (resCard k) : ℝ) : ℂ) ^ ((t : ℤ) - 1 - 2 * i)

theorem **genVal_add_segL** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (r N t n₀ : ℕ) (B : Multiset ℂˣ) (zS zT : ℂˣ), B.card = N → n₀ ≤ N + t → genVal (resCard k) r (N + t) (B.map (· * (zS * zT⁻¹)) + segL ϖ zS t) n₀ = (zS : ℂ) ^ n₀ * ((Real.sqrt (resCard k) : ℝ) : ℂ) ^ (n₀ * (r + N)) * ∑ n ∈ Finset.range (N + 1), (if n ≤ n₀ then ((resCard k ^ (n * (t - (n₀ - n))) * Formalism.L2.Col.gaussBin (resCard k) t (n₀ - n) : ℕ) : ℂ) else 0) * (((zT : ℂ) ^ n * ((Real.sqrt (resCard k) : ℝ) : ℂ) ^ (n * (r + N + t)))⁻¹ * genVal (resCard k) r N B n)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GeneratorMemSplitPrimeLevelPrimeMemPlace

theorem **mem_place_of_mem_prime** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) {w : HeightOneSpectrum (𝒪 D.host.E)} (hI : Formalism.L2.NTheta.IsIdemAt p.e w) (x : 𝒪 D.host.F) (y : 𝒪 D.host.E) (hxy : (y : D.host.E) = algebraMap D.host.F D.host.E (x : D.host.F)) (hy : y ∈ w.asIdeal) : x ∈ p.v.asIdeal

theorem **natCast_mem_place** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) {w : HeightOneSpectrum (𝒪 D.host.E)} (hI : Formalism.L2.NTheta.IsIdemAt p.e w) (n : ℕ) (hn : (n : 𝒪 D.host.E) ∈ w.asIdeal) : (n : 𝒪 D.host.F) ∈ p.v.asIdeal

theorem **GlobalTheory.AdForm.piT_mem_and_T_mem_place** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt (I.rec7 l Z) k) (P : (B.primes X).vT u), Formalism.L2.NTheta.IsIdemAt p.e P.1 → X.piT u ∈ P.1.asIdeal ∧ ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ p.v.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GeneratorMemSplitPrimeLevelPrimeMemPlaceStatement

theorem **_root_.GlobalTheory.T.generator_mem_split_prime_level_prime_mem_place** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt (I.rec7 l Z) k) (P : (B.primes X).vT u), Formalism.L2.NTheta.IsIdemAt p.e P.1 → X.piT u ∈ P.1.asIdeal ∧ ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ p.v.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GenericTermsPointsAgreeGroupSumsCorankOne

theorem **genericTerms_points_agree_groupSums_corankOne** : ∀ (D : D7) (X : HidaSetting D) (Y : Full X) (u : D.what₂.Word) (νL νB : X.B.I.Ginf → Measure X.B.I.Ginf) (ng : PN.Pts D.host.E D.R.rS → ℝ), (∀ γ, PN.IsPt (genT (scT (Y.tO u u))) (NOf1 Y u νL νB ng) γ ↔ PN.IsPt (genT (Y.tO u u)) (NOfG Y u νL νB ng) γ) ∧ (∀ γ, PN.IsPt (genT (scT (Y.tO u u))) (NOf1 Y u νL νB ng) γ ↔ HasPt Y u u (Y.tO u u).genSet γ) ∧ (∀ c γ ν, PN.grp (genT (scT (Y.tO u u))) (NOf1 Y u νL νB ng) c γ ν = grpF Y u u (Y.tO u u).genSet (cls1 Y u) c γ ν) ∧ ((∀ t ∈ (Y.tO u u).genSet, J2of Y u u νL (ptPN t) = 0 → (Y.tO u u).vec t = 0) → PN.AmpInput (genT (scT (Y.tO u u))) (NOf1 Y u νL νB ng))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GenusTypePlaceSetNotConjugatePlaces

theorem **pg_places_not_isConjPl** : ∀ {D : D7} (Pg : Finset (Pl D.host.E)), GlobalTheory.Core.Mi.IsTypePlaces Pg → (∃ e, IsIdemSum e Pg) → ∀ PP ∈ Pg, ∀ PP' ∈ Pg, ¬ IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GenusTypePlaceSetNotConjugatePlacesStatement

theorem **_root_.GlobalTheory.T.genusTypePlaceSet_not_conjugatePlaces** : ∀ {D : D7} (Pg : Finset (Pl D.host.E)), GlobalTheory.Core.Mi.IsTypePlaces Pg → (∃ e, IsIdemSum e Pg) → ∀ PP ∈ Pg, ∀ PP' ∈ Pg, ¬ IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GenusTypePlacesPairwiseNotConjugate

theorem **typePlaces_noConj** : ∀ (D : D7) (Pg : Finset (Pl D.host.E)), GlobalTheory.Core.Mi.IsTypePlaces Pg → ∀ PP ∈ Pg, ∀ PP' ∈ Pg, ¬ IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GenusTypePlacesPairwiseNotConjugateStatement

theorem **_root_.GlobalTheory.T.genus_type_places_pairwise_not_conjugate** : ∀ (D : D7) (Pg : Finset (Pl D.host.E)), GlobalTheory.Core.Mi.IsTypePlaces Pg → ∀ PP ∈ Pg, ∀ PP' ∈ Pg, ¬ IsConjPl (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlMemGfC

theorem **gl_mem_GfC** : ∀ {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {p : SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : Frame p S n) (g : GL n k), β.gl g ∈ GfC S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlobalDoublingZetaFormSelfDualLatticeTransfer

theorem **rmul_one'** {V : HSp D} (q : AutomorphicL2Unitary.Quot (GQ V)) : AutomorphicL2Unitary.rmul (GQ V) 1 q = q

theorem **descend_mul_right** {V : HSp D} {φ : GA D V → ℂ} (hφ : AutomorphicL2Unitary.LeftInv (GQ V) φ) (y : GA D V) : AutomorphicL2Unitary.descend (GQ V) (fun x => φ (x * y)) = AutomorphicL2Unitary.descend (GQ V) φ ∘ AutomorphicL2Unitary.rmul (GQ V) y

theorem **integral_descend_mul_right** {φ : GA D V → ℂ} (hφ : AutomorphicL2Unitary.LeftInv (GQ V) φ) (y : GA D V) : ∫ q, AutomorphicL2Unitary.descend (GQ V) (fun x => φ (x * y)) q ∂autMeasure V m = ∫ q, AutomorphicL2Unitary.descend (GQ V) φ q ∂autMeasure V m

theorem **integrable_descend_mul_right_iff** {φ : GA D V → ℂ} (hφ : AutomorphicL2Unitary.LeftInv (GQ V) φ) (y : GA D V) : Integrable (AutomorphicL2Unitary.descend (GQ V) (fun x => φ (x * y))) (autMeasure V m) ↔ Integrable (AutomorphicL2Unitary.descend (GQ V) φ) (autMeasure V m)

theorem **petersson_rTr** {φ ψ : GA D V → ℂ} (hφ : AutomorphicL2Unitary.LeftInv (GQ V) φ) (hψ : AutomorphicL2Unitary.LeftInv (GQ V) ψ) (y : GA D V) : petersson V m (rTr V y φ) (rTr V y ψ) = petersson V m φ ψ

theorem **integral_conj** (F : GA D V → ℂ) (k : GA D V) : ∫ y, F (k * y * k⁻¹) ∂m = ∫ y, F y ∂m

theorem **integrable_of_conj** {F : GA D V → ℂ} (k : GA D V) (h : Integrable (fun y => F (k * y * k⁻¹)) m) : Integrable F m

theorem **om_mul_one** (g g' : GA D A) (f : WO A B → ℂ) : om (g, 1) (om (g', 1) f) = om (g * g', 1) f

theorem **thetaKO_om** (g k : GA D A) (h : GA D B) (Φ : WO A B → ℂ) : thetaKO om g h (om (k, 1) Φ) = thetaKO om (g * k) h Φ

theorem **leftInv_thetaKO** (hW : IsWeilDatumO ψ A B χA χB om) {Φ : WO A B → ℂ} (hΦ : IsThetaFunO ψ A B Φ) (h : GA D B) : AutomorphicL2Unitary.LeftInv (GQ A) fun g => thetaKO om g h Φ

theorem **wcoef_om** (hW : IsWeilDatumO ψ A B χA χB om) {ν : Measure (WO A B)} {ℱ : Set (WO A B)} (hν : IsCovolOne (ratL B A) ν ℱ) {Φ₁ Φ₂ : WO A B → ℂ} (h₁ : IsThetaFunO ψ A B Φ₁) (h₂ : IsThetaFunO ψ A B Φ₂) (k y : GA D A) : wcoef om ν ℱ (om (k⁻¹, 1) Φ₁) (om (k⁻¹, 1) Φ₂) y = wcoef om ν ℱ Φ₁ Φ₂ (k * y * k⁻¹)

theorem **rcoef_rTr** {u₁ u₂ : GA D V → ℂ} (h₁ : AutomorphicL2Unitary.LeftInv (GQ V) u₁) (h₂ : AutomorphicL2Unitary.LeftInv (GQ V) u₂) (k y : GA D V) : rcoef m (rTr V k⁻¹ u₁) (rTr V k⁻¹ u₂) y = rcoef m u₁ u₂ (k * y * k⁻¹)

theorem **cuspFormsArchimedeanKFinite_left_invariant** {θ : InfFam D} {A : HSp D} {cA : CptF A} {u : GA D A → ℂ} (hu : u ∈ A0 θ A cA) : AutomorphicL2Unitary.LeftInv (GQ A) u

theorem **globalFormAt_move** : ∀ (D : D7) (A : HSp D) (L L' : Lat A), IsSelfDualC A L → IsSelfDualC A L' → GlobalFormAt A L → GlobalFormAt A L'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlobalDoublingZetaFormSelfDualLatticeTransferStatement

theorem **_root_.GlobalTheory.T.global_doubling_zeta_form_selfDual_lattice_transfer** : ∀ (D : D7) (A : HSp D) (L L' : Lat A), IsSelfDualC A L → IsSelfDualC A L' → GlobalFormAt A L → GlobalFormAt A L'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlobalFormFrameLattice

theorem **globalFormAt_frameLat** : GlobalTheory.Published.WeilRepresentationUnitaryPair → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.DoublingMethod → ∀ D : D7, SiegelWeilFock D → ∀ A : HSp D, GlobalFormAt A (frameLat A)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlobalFormFrameLatticeStatement

theorem **_root_.GlobalTheory.T.global_form_frame_lattice** : GlobalTheory.Published.WeilRepresentationUnitaryPair → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.DoublingMethod → ∀ D : D7, SiegelWeilFock D → ∀ A : HSp D, GlobalFormAt A (frameLat A)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlobalFormOfSiegelWeil

theorem **frameLat_isSelfDualC** {D : D7} (V : HSp D) : IsSelfDualC V (frameLat V)

theorem **globalForm_of_siegelWeil** : GlobalTheory.Published.WeilRepresentationUnitaryPair → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.WeilSiegelFormulaUnitary → GlobalTheory.Published.SiegelWeilUnitaryConvergent → GlobalTheory.Published.DoublingMethod → ∀ D : D7, GlobalForm D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GlobalFormOfSiegelWeilStatement

theorem **_root_.GlobalTheory.T.global_form_of_siegel_weil** : GlobalTheory.Published.WeilRepresentationUnitaryPair → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.WeilSiegelFormulaUnitary → GlobalTheory.Published.SiegelWeilUnitaryConvergent → GlobalTheory.Published.DoublingMethod → ∀ D : D7, GlobalForm D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedLatticeDatumMainPartMemSchwartzBruhat

theorem **mainDatum_mem_SB** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (ε : 𝓀[k]ˣ) (P : Pair k n') (Q : Pair k n) : mainDatum q χ ε P Q ∈ SB (Matrix n n' k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedLatticeDatumMainPartMemSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.gluedLatticeDatum_mainPart_mem_schwartzBruhat** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (ε : 𝓀[k]ˣ) (P : Pair k n') (Q : Pair k n), mainDatum q χ ε P Q ∈ SB (Matrix n n' k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedLatticeFiniteExpansionsLocalField

theorem **card_dual** (L : Submodule 𝒪[k] (n → k)) [Fintype n] [L.IsLattice k] : Nat.card (Module.Dual 𝓀[k] (Red L)) = resCard k ^ Fintype.card n

theorem **finsum_char** (χ : AddChar 𝓀[k] ℂ) (hχ : χ ≠ 1) (L : Submodule 𝒪[k] (n → k)) [Fintype n] [L.IsLattice k] (z : Red L) : ∑ᶠ f : Module.Dual 𝓀[k] (Red L), χ (-f z) = if z = 0 then ((resCard k : ℂ) ^ Fintype.card n) else 0

theorem **card_ne_zero** (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] : ((resCard k : ℂ) ^ Fintype.card n) ≠ 0

theorem **withCol_split** (x : Matrix n (n' ⊕ Fin 1) k) : withCol (x.submatrix id Sum.inl) (fun i => x i (Sum.inr 0)) = x

theorem **term_pos** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (P : Pair k n') {L : Submodule 𝒪[k] (n → k)} {x₀ : Matrix n n' k} (h0 : ∀ z ∈ P.ker, x₀ *ᵥ z ∈ L) (f : Module.Dual 𝓀[k] (Red L)) : term q χ 1 P L f x₀ = χ (-f (Red.mk L (imgE q P h0)))

theorem **term_neg** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (P : Pair k n') {L : Submodule 𝒪[k] (n → k)} {x₀ : Matrix n n' k} (h0 : ¬ ∀ z ∈ P.ker, x₀ *ᵥ z ∈ L) (f : Module.Dual 𝓀[k] (Red L)) : term q χ 1 P L f x₀ = 0

theorem **datum_expand** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (hχ : χ ≠ 1) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (P : Pair k n') (x₀ : Matrix n n' k) (y : n → k) : datum q P L x₀ y = ((resCard k : ℂ) ^ Fintype.card n)⁻¹ * ∑ᶠ f : Module.Dual 𝓀[k] (Red L), term q χ 1 P L f x₀ * (if h : y ∈ L then χ (-f (Red.mk L ⟨y, h⟩)) else 0)

theorem **glued_col** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (hχ : χ ≠ 1) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (P : Pair k n') : latInd (gluedS q 1 P.ker P.vec) L = ((resCard k : ℂ) ^ Fintype.card n)⁻¹ • (tensCol (latInd P.ker L) (vecInd L) + ∑ Q ∈ withLat L, tensCol (latInd P.ker Q.ker + mainDatum q χ 1 P Q) (charInd χ Q))

theorem **glued_sum** (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (hχ : χ ≠ 1) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (P : Pair k n') : latInd P.ker L + ∑ Q ∈ withLat L, (latInd P.ker Q.ker + mainDatum q χ 1 P Q) = ((resCard k : ℂ) ^ Fintype.card n) • latInd P.lat L

theorem **dualInd_eq** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) (η : n → k) : dualInd L η = if Lifts L (q.1 • η) 0 then 1 else 0

theorem **cosetInd_pair** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (f : {f : Module.Dual 𝓀[k] (Red L) // f ≠ 0}) (η : n → k) : cosetInd q (ofDual L f).lat (ofDual L f).red η = if Lifts L (q.1 • η) f.1 then 1 else 0

theorem **finsum_lifts** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) (η : n → k) (A : Module.Dual 𝓀[k] (Red L) → ℂ) : ∑ᶠ f : Module.Dual 𝓀[k] (Red L), A f * (if Lifts L (q.1 • η) f then 1 else 0) = if h : InDual L (q.1 • η) then A (exists_lifts h).choose else 0

theorem **coset_total** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (η : n → k) : (dualInd L + ∑ Q ∈ withLat L, cosetInd q Q.lat Q.red) η = if InDual L (q.1 • η) then 1 else 0

theorem **glued_row** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (R : Pair k n') : latInd L (gluedS q 1 R.ker R.vec) = tensRow (latInd L R.ker) (dualInd L) + ∑ Q ∈ withLat L, tensRow (phiOne Q R) (cosetInd q Q.lat Q.red)

theorem **last_ker** (L : Submodule 𝒪[k] (n → k)) (L' : Submodule 𝒪[k] (n' → k)) : latInd L (lastKer L') = tensRow (latInd L L') (dualInd L)

theorem **last_lat** (q : Unif k) (L : Submodule 𝒪[k] (n → k)) [L.IsLattice k] (L' : Submodule 𝒪[k] (n' → k)) [L'.IsLattice k] : latInd L (lastPair q L').lat = tensRow (latInd L L') (dualInd L + ∑ Q ∈ withLat L, cosetInd q Q.lat Q.red)

theorem **finite_expansions** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (χ : AddChar 𝓀[k] ℂ), χ ≠ 1 → ∀ (Λ : Submodule 𝒪[k] (n → k))

[Λ.IsLattice k], (∀ P : Pair k n', latInd (gluedS q 1 P.ker P.vec) Λ = ((resCard k : ℂ) ^ Fintype.card n)⁻¹ • (tensCol (latInd P.ker Λ) (vecInd Λ) + ∑ Q ∈ withLat Λ, tensCol (latInd P.ker Q.ker + mainDatum q χ 1 P Q) (charInd χ Q)) ∧ latInd P.ker Λ + ∑ Q ∈ withLat Λ, (latInd P.ker Q.ker + mainDatum q χ 1 P Q) = ((resCard k : ℂ) ^ Fintype.card n) • latInd P.lat Λ) ∧ (∀ R : Pair k n', latInd Λ (gluedS q 1 R.ker R.vec) = tensRow (latInd Λ R.ker) (dualInd Λ) + ∑ Q ∈ withLat Λ, tensRow (phiOne Q R) (cosetInd q Q.lat Q.red)) ∧ ∀ (L : Submodule 𝒪[k] (n' → k)) [L.IsLattice k], latInd Λ (lastKer L) = tensRow (latInd Λ L) (dualInd Λ) ∧

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedLatticeFiniteExpansionsLocalFieldStatement

theorem **_root_.GlobalTheory.T.gluedLattice_finite_expansions_localField** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (χ : AddChar 𝓀[k] ℂ), χ ≠ 1 → ∀ (Λ : Submodule 𝒪[k] (n → k)) [Λ.IsLattice k], (∀ P : Pair k n', latInd (gluedS q 1 P.ker P.vec) Λ = ((resCard k : ℂ) ^ Fintype.card n)⁻¹ • (tensCol (latInd P.ker Λ) (vecInd Λ) + ∑ Q ∈ withLat Λ, tensCol (latInd P.ker Q.ker + mainDatum q χ 1 P Q) (charInd χ Q)) ∧ latInd P.ker Λ + ∑ Q ∈ withLat Λ, (latInd P.ker Q.ker + mainDatum q χ 1 P Q) = ((resCard k : ℂ) ^ Fintype.card n) • latInd P.lat Λ) ∧ (∀ R : Pair k n', latInd Λ (gluedS q 1 R.ker R.vec) = tensRow (latInd Λ R.ker) (dualInd Λ) + ∑ Q ∈ withLat Λ, tensRow (phiOne Q R) (cosetInd q Q.lat Q.red)) ∧ ∀ (L : Submodule 𝒪[k] (n' → k)) [L.IsLattice k], latInd Λ (la

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedPlaceDoubleStepExactSignedConstant

theorem **card_withLat_smul** (g : GL n k) (L : Submodule 𝒪[k] (n → k)) : (withLat (g • L)).card = (withLat L).card

theorem **card_withLat_eq** (Q Q₀ : Pair k n) : (withLat Q.lat).card = (withLat Q₀.lat).card

theorem **card_withLat_pos** (Q : Pair k n) : 0 < (withLat Q.lat).card

theorem **pairCount_pos** : 0 < pairCount k n

theorem **card_withLat_eq_pairCount** (Q : Pair k n) : (withLat Q.lat).card = pairCount k n

theorem **vertexExact** (D : D7) : VertexExact D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedPlaceDoubleStepExactSignedConstantStatement

theorem **_root_.GlobalTheory.T.gluedPlace_doubleStep_exact_signed_constant** : ∀ D : D7, GlobalTheory.FinPl.VertexExact D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedPlaceDoubleStepVanishTargetCoreLattice

theorem **GlobalTheory.FinPl.vertexGlued_vanish_lastLat** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank), IsStdChar D ψ (-1) → UnrAt s₁ p.v → UnrAt s₂ p.v → ∀ (M₀ : Lat B.V₀) (M₁ : Lat J.W₀) (M : Lat B'.V₀) (α : Frame p B.A n') (α₀ : Frame p B.V₀ (n' ⊕ Fin 1)) (β : Frame p J.W₀ n) (γ : Frame p B'.A n'') (γ₀ : Frame p B'.V₀ (n'' ⊕ Fin 1)) (P : Pair k n') (Q : Pair k n) (Λ : Submodule 𝒪[k] (n'' → k)) [Λ.IsLattice k], IsSelfDualC B.V₀ M₀ → IsSelfDualC J.W₀ M₁ → IsSelfDualC B'.V₀ M → GluedCore B M₀ q α α₀ P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedPlaceDoubleStepVanishTargetCoreLatticeStatement

theorem **_root_.GlobalTheory.T.gluedPlace_doubleStep_vanish_targetCoreLattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank), IsStdChar D ψ (-1) → UnrAt s₁ p.v → UnrAt s₂ p.v → ∀ (M₀ : Lat B.V₀) (M₁ : Lat J.W₀) (M : Lat B'.V₀) (α : Frame p B.A n') (α₀ : Frame p B.V₀ (n' ⊕ Fin 1)) (β : Frame p J.W₀ n) (γ : Frame p B'.A n'') (γ₀ : Frame p B'.V₀ (n'' ⊕ Fin 1)) (P : Pair k n') (Q : Pair k n) (Λ : Submodule 𝒪[k] (n'' → k)) [Λ.IsLattice k], IsSelfDualC B.V₀ M₀ → IsSelfDualC J.W₀ M₁ → IsSelfDualC B'.V₀ M → G

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedPlaceSourceConstantProof

theorem **length_residue** : Module.length 𝒪[k] 𝓀[k] = 1

theorem **fn_surjective** (P : Pair k n') : Function.Surjective P.fn

theorem **valuation_det_mul_unif** (q : Unif k) (P : Pair k n') (g : GL n' k) (hg : g • P.ker = P.lat) : valuation k (((Matrix.GeneralLinearGroup.det g : kˣ) : k) * (q.unit : k)) = 1

theorem **Frame.rank_eq** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {p : SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : Frame p S n) : S.rank = Fintype.card n

theorem **chrPow_neg_one_mul** (χ : (AE D)ˣ →* Circle) (a : (AE D)ˣ) : chrPow χ a (-1) * chrPow χ a 1 = 1

theorem **exists_theta_exact** {D : D7} (hLin : LinExact D) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (hA : CoCpt A) (vA : Vol A) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (hW : IsWeilDatumO ψ S T χS χT om) (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)) (β : Frame p T n) {ζ : Fin L.rank → D.host.E} (hsum : IsSumFrame o ζ α α₀) : ∃ lS : kˣ →* ℂˣ, (CharUnramAt D χT p.v → IsUnram lS) ∧ (lS q.unit : ℂ) = (((resCard k : ℝ) ^ (-(T.rank : ℝ) / 2) : ℝ) : ℂ) * chrPow χT (p.idele q.unit) (-1) ∧ ∀ (LS : Lat S) (LT : Lat T) (f₀ : WO S T → ℂ), IsSelfDualC S LS → IsSelfDualC T LT → Vac ψ S T LS LT (· = p.v) f₀ → ∃ θ : CA A →ₗ[ℂ] (Matrix n (n' ⊕ Fin 1) k → ℂ) →ₗ[ℂ] (GA D T → WO L T → ℂ), (

theorem **GlobalTheory.FinPl.sourceExact** : ∀ D : D7, GlobalTheory.FinPl.SourceExact D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedPlaceSourceConstantProofStatement

theorem **_root_.GlobalTheory.T.gluedPlaceSourceConstant_proof** : ∀ D : D7, GlobalTheory.FinPl.SourceExact D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedSplitPlaceLineCoordinateIntegral

theorem **mt_sec'** (p : SplitPt D k) {S : HSp D} (h : GA D S) (u : Fin S.rank → k) : mt S h *ᵥ (fun i => p.sec (u i)) = fun i => p.sec (((mt S h).map p.pr *ᵥ u) i)

theorem **cAe_smul** (p : SplitPt D k) {r : ℕ} (z : Fin r → AE D) : cA D p.e • z = fun i => cA D (p.sec (p.pr (cA D (z i))))

theorem **form_eq** (p : SplitPt D k) (S : HSp D) (x : Fin S.rank → k) (c : Fin S.rank → D.host.E) : form p S x (fun j => p.pr (cA D (incA D (c j)))) = x ⬝ᵥ fun i => p.pr (incA D ((star (S.gram *ᵥ c)) i))

theorem **row_orth** {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) : star (S.gram *ᵥ vecB o ζ) ᵥ* o.iA = 0 ∧ star (S.gram *ᵥ vecB o ζ) ⬝ᵥ vecB o ζ = star ζ ⬝ᵥ L.gram *ᵥ ζ

theorem **form_frame** (p : SplitPt D k) {ρ : ℕ} (B : BesLev D ρ) {n : Type} [Fintype n] [DecidableEq n] (β : Frame p B.A n) (β₀ : Frame p B.V₀ (n ⊕ Fin 1)) (hs : IsSumFrame B.o B.ζ β β₀) (y : n ⊕ Fin 1 → k) : form p B.V₀ (β₀.b *ᵥ y) (fun j => p.pr (cA D (ratV (vecB B.o B.ζ) j))) = p.pr (incA D (star B.ζ ·ᵥ B.L.gram *ᵥ B.ζ)) ∗ y (Sum.inr 0)

theorem **GlobalTheory.FinPl.glued_last_integral** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (σ : ℕ) (B : BesLev D σ) (L₀ : Lat B.V₀) (β : Frame p B.A n) (β₀ : Frame p B.V₀ (n ⊕ Fin 1)) (P : Pair k n), IsSelfDualC B.V₀ L₀ → GluedCore B L₀ q β β₀ P → ∀ γ ∈ grpAt B.V₀ (· = p.v), (∀ a : AE D, a • ratV (vecB B.o B.ζ) ∈ latA B.V₀ L₀ (· = p.v) → mt B.V₀ γ⁻¹ *ᵥ (a • ratV (vecB B.o B.ζ)) ∈ latA B.V₀ L₀ (· = p.v)) → ∀ x : n ⊕ Fin 1 → k, mt B.V₀ γ⁻¹ *ᵥ (fun i => p.sec ((β₀.b *ᵥ x) i)) ∈ latV B.V₀ L₀ (· = p.v) → (q.1 : k) ∗ x (Sum.inr 0) ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GluedSplitPlaceLineCoordinateIntegralStatement

theorem **_root_.GlobalTheory.T.gluedSplitPlace_lineCoordinate_integral** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (q : Unif k) (σ : ℕ) (B : BesLev D σ) (L₀ : Lat B.V₀) (β : Frame p B.A n) (β₀ : Frame p B.V₀ (n ⊕ Fin 1)) (P : Pair k n), IsSelfDualC B.V₀ L₀ → GluedCore B L₀ q β β₀ P → ∀ γ ∈ grpAt B.V₀ (· = p.v), (∀ a : AE D, a • ratV (vecB B.o B.ζ) ∈ latA B.V₀ L₀ (· = p.v) → mt B.V₀ γ⁻¹ *ᵥ (a • ratV (vecB B.o B.ζ)) ∈ latA B.V₀ L₀ (· = p.v)) → ∀ x : n ⊕ Fin 1 → k, mt B.V₀ γ⁻¹ *ᵥ (fun i => p.sec ((β₀.b *ᵥ x) i)) ∈ latV B.V₀ L₀ (· = p.v) → (q.1 : k) ∗ x (Sum.inr 0) ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GodementSectionInvariantConvergent

theorem **grp_cover** {G : Type} [Group G] {c₀ b₀ B C L e c₁ c₂ y a y' c w w' : G} (h1 : b₀ ∗ B = e ∗ c₁) (h2 : c₀ ∗ e = e ∗ c₀) (h3 : c₀ ∗ (c₁ ∗ C) = c₂ ∗ (y' ∗ c)) (h4 : y ∗ a = e ∗ c₂) (h5 : y' ∗ a = a ∗ y') (h6 : L ∗ w = a ∗ L) (h7 : L ∗ w' = c ∗ L) : c₀ ∗ b₀ ∗ (B ∗ C ∗ L) = y ∗ y' ∗ L ∗ (w ∗ w')

theorem **grp_reduce** {G : Type} [Group G] {p u g B B' B'' C C' g' L κ a c : G} (e : p = B ∗ C ∗ g' ∗ L) (h1 : g ∗ B = B' ∗ g) (h2 : u ∗ B' = B'' ∗ u) (h3 : g ∗ C = C ∗ g) (h4 : u ∗ C = C' ∗ u) (h5 : g ∗ g' = κ) (h6 : κ ∗ L = L ∗ κ) (h7 : u ∗ L = a ∗ c) : u ∗ g ∗ p = B'' ∗ C' ∗ a ∗ (c ∗ κ)

theorem **add_rearr** {A : Type} [AddCommGroup A] {z z' c z'' : A} (e : z − z' = c + z'') : −c + z = z' + z''

theorem **row_ne_zero** {A : Type} [CommRing A] [Nontrivial A] {r n : ℕ} {M : Matrix (Fin r) (Fin n) A} {N : Matrix (Fin n) (Fin r) A} (h : M ∗ N = 1) (i : Fin r) : ¬∀ x, M i x = 0

theorem **nB_mem** (b : Mx D r T₀.rank) : d.nB b ∈ stab d

theorem **nC_mem** (c : skewA D r) : d.nC c ∈ stab d

theorem **wittCoreEmbedding_mem_isotropicStabilizer** (g : GA D T₀) : d.emb0 g ∈ stab d

theorem **cover_rad** : ∃ (W : Set (GA D T)) (C : ℝ), IsCompact W ∧ ∀ m : Fin r → ℕ, ∃ Y : Finset (GA D T), (Y.card : ℝ) ≤ C ∗ ∏ i, bs D ^ ((T₀.rank + r) ∗ m i) ∧ ∀ b c, ∃ γ ∈ stab d ⊓ GQ T, ∃ y ∈ Y, ∃ w ∈ W, γ ∗ (d.nB b ∗ d.nC c ∗ d.levi (dg (hf D) fun i => (m i : ℤ))) = y ∗ d.levi (dg (hf D) fun i => (m i : ℤ)) ∗ w

theorem **exists_decomp** (p : stab d) : ∃ b c g, (p : GA D T) = d.nB b ∗ d.nC c ∗ d.emb0 g ∗ d.levi (onX d p)

theorem **reduction_GL_diagonalUnitPowers_compact** {E A : Type} [Field E] [CommRing A] [TopologicalSpace A] [IsTopologicalRing A] [T2Space A] (j : E →+* A) {s : Aˣ} {ν : Aˣ →* ℝ≥0} (hν : Continuous ν) (hj : ∀ u : Eˣ, ν (Units.map j.toMonoidHom u) = 1) {b : ℝ} (hb : 0 < b) (hs : (ν s : ℝ) = b⁻¹) {r : ℕ} (hR : ∃ Ω : Set A, IsCompact Ω ∧ ∀ β : GL (Fin r) A, ∃ (δ : GL (Fin r) E) (l : Fin r → ℤ) (c : GL (Fin r) A), (∀ i k, c.1 i k ∈ Ω ∧ c⁻¹.1 i k ∈ Ω) ∧ δ.1.map j ∗ β.1 = diagonal (fun i => (s ^ l i).1) ∗ c.1) : ∃ (L : Set (GL (Fin r) A)) (C : ℝ), IsCompact L ∧ ∀ α : GL (Fin r) A, ∃ (δ : GL (Fin r) E) (l : Fin r → ℤ), ∃ c ∈ L, bc j δ ∗ α = dg s l ∗ c ∧ (ν (GeneralLinearGroup.det α) : ℝ) ≤ C ∗ ∏ i, (b ^ l i)⁻¹

theorem **reduce_gl** (D : D7) (r : ℕ) : ∃ (L : Set (GL (Fin r) (AE D))) (C : ℝ), IsCompact L ∧ ∀ α : GL (Fin r) (AE D), ∃ (δ : GL (Fin r) D.host.E) (l : Fin r → ℤ), ∃ c ∈ L, bc (incA D) δ ∗ α = dg (hf D) l ∗ c ∧ (absA D (GeneralLinearGroup.det α) : ℝ) ≤ C ∗ ∏ i, (bs D ^ l i)⁻¹

theorem **reduce** (h₀ : IsAniso T₀) : ∃ (W : Set (GA D T)) (C : ℝ), IsCompact W ∧ ∀ p : stab d, ∃ γ ∈ stab d ⊓ GQ T, ∃ b c, ∃ l : Fin r → ℤ, ∃ w ∈ W, γ ∗ p = d.nB b ∗ d.nC c ∗ d.levi (dg (hf D) l) ∗ w ∧ (absDet d p : ℝ) ≤ C ∗ ∏ i, (bs D ^ l i)⁻¹

theorem **pY_mul** (b : Mx D r T₀.rank) (c : skewA D r) (a : GL (Fin r) (AE D)) (w : GA D T) : d.pY · mt T (d.nB b ∗ d.nC c ∗ d.levi a ∗ w) = adj a⁻¹.1 · (d.pY · mt T w)

theorem **pY_row** (g : GA D T) (i : Fin r) : ¬∀ x, (d.pY · mt T g) i x = 0

theorem **bound_aux** {E A : Type} [Field E] [CommRing A] (j : E →+* A) {s : Aˣ} {Ω : Set A} {m₀ r n : ℕ} (hm₀ : ∀ m, m₀ ≤ m → ∀ ω ∈ Ω, ∀ x : E, j x = s.1 ^ m ∗ ω → x = 0) {l : Fin r → ℤ} {y : Matrix (Fin r) (Fin n) E} {w : Matrix (Fin r) (Fin n) A} (hw : ∀ i x, w i x ∈ Ω) (e : y.map j = (dg s l)⁻¹.1 ∗ w) (i : Fin r) (h : ¬∀ x, y.map j i x = 0) : −(m₀ : ℤ) ≤ l i

theorem **exists_bound** {W : Set (GA D T)} (hW : IsCompact W) : ∃ m₀ : ℕ, ∀ b c (l : Fin r → ℤ), ∀ w ∈ W, d.nB b ∗ d.nC c ∗ d.levi (dg (hf D) l) ∗ w ∈ GQ T → ∀ i, −(m₀ : ℤ) ≤ l i

theorem **dg_sub** (s : (AE D)ˣ) (m : Fin r → ℕ) (m₀ : ℤ) : (dg s fun i => (m i : ℤ) + m₀) = (dg s fun i => (m i : ℤ)) ∗ dg s fun _ => m₀

theorem **grp_count** {G : Type} [Group G] {γ' γ x p k n a z w y w' : G} (e : x = p ∗ k) (h1 : γ ∗ p = n ∗ (a ∗ z) ∗ w) (h2 : γ' ∗ (n ∗ a) = y ∗ a ∗ w') : γ' ∗ γ ∗ x = y ∗ a ∗ (w' ∗ z ∗ w ∗ k)

theorem **exists_card_le'** {Q : Type} {Z w : Q → GA D T} (hZ : ∀ q, Z q ∈ GQ T) (hi : Function.Injective Z) (hw : Bdd w) : ∃ N : ℕ, ∀ (z : GA D T) (S : Finset Q), (∀ q ∈ S, Z q = z ∗ w q) → S.card ≤ N

theorem **summable_absDet** (h₀ : IsAniso T₀) {Q : Type} {x k : Q → GA D T} {p : Q → stab d} (hx : ∀ q, x q ∈ GQ T) (e : ∀ q, x q = p q ∗ k q) (hk : Bdd k) (hi : ∀ q q', ∀ γ ∈ stab d, x q' = γ ∗ x q → q = q') {t : ℝ} (ht : ((T₀.rank + r : ℕ) : ℝ) < t) : Summable fun q => (absDet d (p q) : ℝ) ^ t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GodementSectionSmoothBounded

theorem **isRealLine_oneP** (hθ : IsInfFam θ) (V : HSp D) {τ : D.what₂.Tau} {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) : IsRealLine D.sp.cm.sig V.gram (oneP θ V τ X)

theorem **mem_smFun_of_isSm** (hθ : IsInfFam θ) {f : GA D V → ℂ} (h : GlobalTheory.UAut.IsSm D.sp.cm.sig V.gram f) : f ∈ SmFun θ V

theorem **derSpan_le** (hθ : IsInfFam θ) {f : GA D V → ℂ} (hf : f ∈ SmFun θ V) : derSpan V f ≤ SmFun θ V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GoodMajorantLevelConjUnitaryAdelicMaximalCompact

theorem **good_level_conj_stab** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (b : GL (Fin T.rank) (AE D)), IsMaj T b → HasIw d (Kmax T b) → ∀ l : Lvl T, l.Good → HasIw d l.K ∧ IsCompact (l.K : Set (GA D T)) ∧ ∃ q : stab d, ConjBy (q : GA D T) l.K (Kmax T b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GoodMajorantLevelConjUnitaryAdelicMaximalCompactStatement

theorem **_root_.GlobalTheory.T.goodMajorantLevel_conj_unitaryAdelicMaximalCompact** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (b : GL (Fin T.rank) (AE D)), IsMaj T b → HasIw d (Kmax T b) → ∀ l : Lvl T, l.Good → HasIw d l.K ∧ IsCompact (l.K : Set (GA D T)) ∧ ∃ q : stab d, ConjBy (q : GA D T) l.K (Kmax T b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GoodPositionSplitLinesReadCompletionsAtPlace

theorem **mul_mem_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {z : Fin S.rank → AE D} (hz : z ∈ latV S L P) : (fun i => ((1 : AF D.host.F) ⊗ₜ[D.host.F] a) * z i) ∈ latA S L P

theorem **latA_le_of_latV_le** {S : HSp D} {L L' : Lat S} {P : Pl D → Prop} (h : latV S L P ≤ latV S L' P) : latA S L P ≤ latA S L' P

theorem **latA_congr** {S : HSp D} {L L' : Lat S} {P : Pl D → Prop} (h : latV S L P = latV S L' P) : latA S L P = latA S L' P

theorem **latO_congr** {S T : HSp D} {LS LS' : Lat S} {LT LT' : Lat T} (hS : IsSelfDualC S LS) (hS' : IsSelfDualC S LS') (hT : IsSelfDualC T LT) (hT' : IsSelfDualC T LT') {P : Pl D → Prop} (eS : latV S LS P = latV S LS' P) (eT : latV T LT P = latV T LT' P) : latO S T (LS : Set _) (LT : Set _) P = latO S T (LS' : Set _) (LT' : Set _) P

theorem **monoAt_congr** {S S₀ : HSp D} {ρ : ℕ} {d : HypDec S S₀ ρ} {L L' : Lat S} {L₀ L₀' : Lat S₀} {v : Pl D} (e : latV S L (· = v) = latV S L' (· = v)) (e₀ : latV S₀ L₀ (· = v) = latV S₀ L₀' (· = v)) (H : MonoAt d L L₀ v) : MonoAt d L' L₀' v

theorem **orthAt_congr** {S A B : HSp D} {o : OrthDec S A B} {L L' : Lat S} {LA : Lat A} {LB : Lat B} {v : Pl D} (e : latV S L (· = v) = latV S L' (· = v)) (H : OrthAt o L LA LB v) : OrthAt o L' LA LB v

theorem **primAt_congr** {ψ : AddChar (AF D.host.F) ℂ} {S S₀ : HSp D} {ρ : ℕ} {d : HypDec S S₀ ρ} {L L' : Lat S} {v : Pl D} (e : latV S L (· = v) = latV S L' (· = v)) (H : PrimAt ψ d L v) : PrimAt ψ d L' v

theorem **colsAt_congr** {V V₀ W : HSp D} {ρ : ℕ} {d : HypDec V V₀ ρ} {x : Mx D W.rank ρ} {LV LV' : Lat V} {LW LW' : Lat W} {v : Pl D} (e : latV V LV (· = v) = latV V LV' (· = v)) (eW : latV W LW (· = v) = latV W LW' (· = v)) (H : ColsAt d x LV LW v) : ColsAt d x LV' LW' v

theorem **goodBF_congr** {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} {LV LV' : Lat B.V} {LV₀ LV₀' : Lat B.V₀} {LW LW' : Lat J.W} {LW₀ : Lat J.W₀} {v : Pl D} (sV : IsSelfDualC B.V LV') (sV₀ : IsSelfDualC B.V₀ LV₀') (sW : IsSelfDualC J.W LW') (eV : latV B.V LV (· = v) = latV B.V LV' (· = v)) (eV₀ : latV B.V₀ LV₀ (· = v) = latV B.V₀ LV₀' (· = v)) (eW : latV J.W LW (· = v) = latV J.W LW' (· = v)) (H : GoodBF ψ B J LV LV₀ LW LW₀ v) : GoodBF ψ B J LV' LV₀' LW' LW₀ v where

theorem **adaptFB_congr** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} {LV LV' : Lat J.W} {LV₀ : Lat J.W₀} {LW LW' : Lat B.V} {LW₀ LW₀' : Lat B.V₀} {v : Pl D} (sV : IsSelfDualC J.W LV') (sW : IsSelfDualC B.V LW') (sW₀ : IsSelfDualC B.V₀ LW₀') (eV : latV J.W LV (· = v) = latV J.W LV' (· = v)) (eW : latV B.V LW (· = v) = latV B.V LW' (· = v)) (eW₀ : latV B.V₀ LW₀ (· = v) = latV B.V₀ LW₀' (· = v)) (H : AdaptFB ψ J B LV LV₀ LW LW₀ v) : AdaptFB ψ J B LV' LV₀ LW' LW₀' v where

theorem **goodAt_splitAt_congr** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι) (N : ∀ j, Lat (c.V j)) (M : ∀ j ρ h, Lat (c.bes j ρ h).V₀) (x : Pl D), (∀ j, j < D.R.m → IsSelfDualC (c.V j) (v.N j) ∧ IsSelfDualC (c.V j) (N j) ∧ latV (c.V j) (N j) (· = x) = latV (c.V j) (v.N j) (· = x)) → (∀ j ρ h, IsSelfDualC _ (o.M j ρ h) ∧ IsSelfDualC _ (M j ρ h) ∧ latV _ (M j ρ h) (· = x) = latV _ (o.M j ρ h) (· = x)) → (GoodAt v o x → GoodAt { v with N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GoodPositionSplitLinesReadCompletionsAtPlaceStatement

theorem **_root_.GlobalTheory.T.goodPosition_splitLines_read_completions_at_place** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι) (N : ∀ j, Lat (c.V j)) (M : ∀ j ρ h, Lat (c.bes j ρ h).V₀) (x : Pl D), (∀ j, j < D.R.m → IsSelfDualC (c.V j) (v.N j) ∧ IsSelfDualC (c.V j) (N j) ∧ latV (c.V j) (N j) (· = x) = latV (c.V j) (v.N j) (· = x)) → (∀ j ρ h, IsSelfDualC _ (o.M j ρ h) ∧ IsSelfDualC _ (M j ρ h) ∧ latV _ (M j ρ h) (· = x) = latV _ (o.M j ρ h) (· = x)) → (GoodAt v o x → GoodAt { v with N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GradedSeriesMulNeZero

theorem **gradedSeries_mul_ne_zero** : ∀ (ι : Type) (deg : ι → ℕ) (a : ι → ℂ) (ρ δ : ℝ) (P Q : ℂ → ℂ), 0 < δ → Differentiable ℂ P → Differentiable ℂ Q → (∀ r : ℝ, 0 < r → r < ρ → Summable fun i => ‖a i‖ * r ^ deg i) → (∀ x : ℝ, 0 < x → x < δ → (∑' i, a i * (x : ℂ) ^ deg i) * P x = Q x) → ∀ X : ℂ, ‖X‖ < ρ → Q X ≠ 0 → P X ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GramEqOfInst

theorem **gram_eq_of_inst** {D : D7} (I : AutInst) (hI : I = Formalism.L2.SG.InstExistsPf.instOf D) (ιE : I.E ≃+* D.host.E) (hId : ∀ x : I.E, HEq (ιE x) x) (hn : (D.sp.Y 0).rank = I.n) (k l : Fin (D.sp.Y 0).rank) : ιE (I.J (Fin.cast hn k) (Fin.cast hn l)) = (D.sp.Y 0).gram k l

theorem **cY_isom** : ∀ (D : D7) (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D), IdE X → ∀ y y' : (VS D).V, star (GlobalTheory.Core.SBesEx.cY X h y') ·ᵥ ((D.sp.Y 0).gram *ᵥ GlobalTheory.Core.SBesEx.cY X h y) = (VS D).hT y' y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GramMatrixOfColumnsUnitriangularConjEq

theorem **gramQ_unitri** : ∀ (D : D7) (W : HSp D) (ρ : ℕ) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E), IsBase W x₀ → ∀ u : Matrix (Fin ρ) (Fin ρ) D.host.E, (∀ i, u i i = 1) → (∀ i k, k < i → u i k = 0) → uᴴ * gramQ W x₀ * u = gramQ W x₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GramMatrixOfColumnsUnitriangularConjEqStatement

theorem **_root_.GlobalTheory.T.gramMatrixOfColumns_unitriangular_conj_eq** : ∀ (D : D7) (W : HSp D) (ρ : ℕ) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E), IsBase W x₀ → ∀ u : Matrix (Fin ρ) (Fin ρ) D.host.E, (∀ i, u i i = 1) → (∀ i k, k < i → u i k = 0) → uᴴ * gramQ W x₀ * u = gramQ W x₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GramPoincareHilbertSchmidtDeviationLeSplit

theorem **hsNormSq_add_le_three_halves** {Wp Wq : Type*} [NormedAddCommGroup Wp] [InnerProductSpace ℂ Wp] [NormedAddCommGroup Wq] [InnerProductSpace ℂ Wq] [FiniteDimensional ℂ Wq] (A B : Wq →L[ℂ] Wp) : hsNormSq (A + B) ≤ 3 / 2 * hsNormSq A + 3 * hsNormSq B

theorem **hsDev_le_split_three_halves** : ∀ (D : D7) (u : D.what₂.Word), GlobalTheory.Core.Rk.hsDev D u ≤ 3 / 2 * hsNormSq (Kmat D.s u u − AdmissibleBoundsFromSource.Kpp D u u − ContinuousLinearMap.id ℂ (D.s.Cu u)) + 3 * hsNormSq (AdmissibleBoundsFromSource.Kpp D u u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GramPoincareHilbertSchmidtDeviationLeSplitStatement

theorem **_root_.GlobalTheory.T.gramPoincareHilbertSchmidtDeviation_le_split** : ∀ (D : D7) (u : D.what₂.Word), GlobalTheory.Core.Rk.hsDev D u ≤ 3 / 2 * hsNormSq (Kmat D.s u u − AdmissibleBoundsFromSource.Kpp D u u − ContinuousLinearMap.id ℂ (D.s.Cu u)) + 3 * hsNormSq (AdmissibleBoundsFromSource.Kpp D u u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.GramUnimodular

theorem **hform_frame** {R : Type} [CommRing R] {n : ℕ} (σ : R →+* R) (J B : Matrix (Fin n) (Fin n) R) (x y : Fin n → R) : Formalism.UG.hform σ J (B *ᵥ x) (B *ᵥ y) = ∑ i, σ (x i) * ((Formalism.UG.cT σ B * J * B) *ᵥ y) i

theorem **gram_int** {R : Type} [CommRing R] {n : ℕ} (σ : R →+* R) (O : Subring R) (hσ : ∀ z ∈ O, σ z ∈ O) (J B C H' : Matrix (Fin n) (Fin n) R) (L : (Fin n → R) → Prop) (hBC : B * C = 1) (hH : Formalism.UG.cT σ B * J * B * H' = 1) (hf : ∀ y, (∀ i, y i ∈ O) ↔ L (B *ᵥ y)) (hd : ∀ y, L y ↔ ∀ x, L x → Formalism.UG.hform σ J x y ∈ O) : (∀ i j, (Formalism.UG.cT σ B * J * B) i j ∈ O) ∧ ∀ i j, H' i j ∈ O

theorem **compFin_cA** {D : D7} (v : HeightOneSpectrum (𝒪 D.host.F)) (z : AE D) : compFin D.host.F D.host.E v (cA D z) = sigR _ D.sp.cm.sig (compFin D.host.F D.host.E v z)

theorem **map_gram** {D : D7} (A : HSp D) (v : HeightOneSpectrum (𝒪 D.host.F)) (b : Mx D A.rank A.rank) : (adj b * gA A * b).map (compFin D.host.F D.host.E v) = Formalism.UG.cT (sigR _ D.sp.cm.sig) (b.map (compFin D.host.F D.host.E v)) * JR (Fv D.host.F v) A.gram * b.map (compFin D.host.F D.host.E v)

theorem **gram_unimodular** : ∀ (D : D7) (A : HSp D) (b : GL (Fin A.rank) (AE D)), IsLatticeFrame (frameLat A) b → ∃ Hm : GL (Fin A.rank) (AE D), Hm.1 = adj b.1 * gA A * b.1 ∧ IntGL Hm

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarAdelicUpperUnitriangularEntryTranslateInvari

theorem **t2AE** (D : D7) : T2Space (AE D)

theorem **tagMx** (D : D7) (ρ : ℕ) : IsTopologicalAddGroup (Mx D ρ ρ)

theorem **lcMx** (D : D7) (ρ : ℕ) : LocallyCompactSpace (Mx D ρ ρ)

theorem **scMx** (D : D7) (ρ : ℕ) : SecondCountableTopology (Mx D ρ ρ)

theorem **t2Mx** (D : D7) (ρ : ℕ) : T2Space (Mx D ρ ρ)

theorem **borelMx** (D : D7) (ρ : ℕ) : BorelSpace (Mx D ρ ρ)

theorem **borelUX** (D : D7) (ρ : ℕ) : BorelSpace (UX D ρ)

theorem **isClosed_supMx** : IsClosed (supMx D ρ : Set (Mx D ρ ρ))

theorem **lcSup** (D : D7) (ρ : ℕ) : LocallyCompactSpace (supMx D ρ)

theorem **mul_mem_supMx** {g : Matrix (Fin ρ) (Fin ρ) (AE D)} (hg : BlockTriangular g id) {s : Mx D ρ ρ} (hs : s ∈ supMx D ρ) : g * s ∈ supMx D ρ

theorem **ccs** (D : D7) (ρ : ℕ) : ContinuousConstSMul (triGrp D ρ) (supMx D ρ)

theorem **modChar_eq_one_of_sq** {a d : triGrp D ρ} (h : d * a * d⁻¹ = a * a) : modChar D ρ a = 1

theorem **one_add_mul_one_sub** {x : Mx D ρ ρ} (hx : x * x = 0) : (1 + x) * (1 − x) = 1

theorem **one_sub_mul_one_add** {x : Mx D ρ ρ} (hx : x * x = 0) : (1 − x) * (1 + x) = 1

theorem **upper_one_add** {x : Mx D ρ ρ} (hx : x ∈ supMx D ρ) : BlockTriangular (1 + x) id

theorem **sq_col** {x : Mx D ρ ρ} (hx : x ∈ supMx D ρ) {j : Fin ρ} (hcol : ∀ i k, k ≠ j → x i k = 0) : x * x = 0

theorem **two_mul_twoInv** : (2 : AE D) * twoInv D = 1

theorem **modChar_col** {x : Mx D ρ ρ} (hx : x ∈ supMx D ρ) {j : Fin ρ} (hcol : ∀ i k, k ≠ j → x i k = 0) : modChar D ρ ⟨colU x (sq_col hx hcol), upper_one_add hx⟩ = 1

theorem **trunc_zero** (u : Mx D ρ ρ) : trunc u 0 = 1

theorem **trunc_top** (u : Mx D ρ ρ) : trunc u ρ = u

theorem **colPart_mem** (u : Mx D ρ ρ) (k : ℕ) : colPart u k ∈ supMx D ρ

theorem **colPart_col** (u : Mx D ρ ρ) {k : ℕ} (hk : k < ρ) : ∀ i l, l ≠ ⟨k, hk⟩ → colPart u k i l = 0

theorem **trunc_succ** {u : Mx D ρ ρ} (hu : u ∈ GlobalTheory.UnitU.uniSet D ρ) (k : ℕ) : trunc u (k + 1) = trunc u k + colPart u k

theorem **colPart_mul_trunc** {u : Mx D ρ ρ} (hu : u ∈ GlobalTheory.UnitU.uniSet D ρ) {k : ℕ} (hk : k < ρ) : colPart u k * trunc u k = colPart u k

theorem **modChar_of_trunc** {u : Mx D ρ ρ} (hu : u ∈ GlobalTheory.UnitU.uniSet D ρ) : ∀ k ≤ ρ, ∀ g : triGrp D ρ, (g : GL (Fin ρ) (AE D)).1 = trunc u k → modChar D ρ g = 1

theorem **modChar_ofUX** (u : UX D ρ) : modChar D ρ (ofUX u) = 1

theorem **map_smul_addHaar** (u : UX D ρ) : (Measure.addHaar : Measure (supMx D ρ)).map (fun s => ofUX u • s) = Measure.addHaar

theorem **sub_one_mem** (u : UX D ρ) : (u : GL (Fin ρ) (AE D)).1 − 1 ∈ supMx D ρ

theorem **one_add_mem** {s : Mx D ρ ρ} (hs : s ∈ supMx D ρ) : 1 + s ∈ GlobalTheory.UnitU.uniSet D ρ

theorem **coordU_mul** (g u : UX D ρ) : coordU D ρ (g * u) = ofUX g • coordU D ρ u + coordU D ρ g

theorem **haar_val_add_invariant** : ∀ (D : D7) (ρ : ℕ) (ν : Measure (UX D ρ)), ν.IsHaarMeasure → Measurable (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ∧ IsFiniteMeasureOnCompacts (Measure.map (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ν) ∧ ∀ t : Mx D ρ ρ, (∀ i j, ¬ i < j → t i j = 0) → Measure.map (fun x => x + t) (Measure.map (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ν) = Measure.map (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarAdelicUpperUnitriangularEntryTranslateInvariStatement

theorem **_root_.GlobalTheory.T.haar_adelicUpperUnitriangular_entry_translate_invariant** : ∀ (D : D7) (ρ : ℕ) (ν : Measure (UX D ρ)), ν.IsHaarMeasure → Measurable (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ∧ IsFiniteMeasureOnCompacts (Measure.map (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ν) ∧ ∀ t : Mx D ρ ρ, (∀ i j, ¬ i < j → t i j = 0) → Measure.map (fun x => x + t) (Measure.map (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ν) = Measure.map (fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)) ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarFiniteAdeleDisintegrationStabilizerCompact

theorem **exists_idem_of_subspace** {E : Type} [Field E] {n : ℕ} (W : Submodule E (Fin n → E)) : ∃ e : Matrix (Fin n) (Fin n) E, e * e = e ∧ ∀ x, x ∈ W ↔ e *ᵥ x = x

theorem **span_incl_eq** (I : AutInst) (T : Set (Fin I.n → I.E)) : Submodule.span (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) (incl I '' T) = Submodule.span (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) ((fun w i => GlobalTheory.UnipApprox.inc I (w i)) '' (Submodule.span I.E T : Set (Fin I.n → I.E)))

theorem **isClosed_cond** (I : AutInst) (v : Fin I.n → I.E) (T : Set (Fin I.n → I.E)) : IsClosed {x : I.Gf | (I.fmat x - 1) *ᵥ incl I v ∈ Submodule.span (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) (incl I '' T)}

theorem **isClosed_sfin** (I : AutInst) {r : ℕ} (e : Fin r → Fin I.n → I.E) (z : Fin I.n → I.E) : IsClosed (Sfin I e z)

theorem **disint_finPart** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word), ∃ μSf : Measure (Formalism.L2.SG.Pf.SfinSub X.B.I X.eR (X.zR u)), μSf.IsHaarMeasure ∧ SigmaFinite μSf ∧ ∃ c : ℝ, 0 < c ∧ ∀ φ : X.B.I.Gf → ENNReal, Measurable φ → (∀ x, φ x ≠ 0 → ∃ s ∈ SfinU X u, ∃ k ∈ X.K2, x = s * X.xu u * k) → ∫⁻ x, φ x ∂X.B.μf = ENNReal.ofReal c * ∫⁻ t, ∫⁻ k in X.K2, φ (t.1 * X.xu u * k) ∂X.B.μf ∂μSf

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarFiniteAdeleDisintegrationStabilizerCompactStatement

theorem **_root_.GlobalTheory.T.haar_finite_adele_disintegration_stabilizer_compact** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word), ∃ μSf : Measure (Formalism.L2.SG.Pf.SfinSub X.B.I X.eR (X.zR u)), μSf.IsHaarMeasure ∧ SigmaFinite μSf ∧ ∃ c : ℝ, 0 < c ∧ ∀ φ : X.B.I.Gf → ENNReal, Measurable φ → (∀ x, φ x ≠ 0 → ∃ s ∈ SfinU X u, ∃ k ∈ X.K2, x = s * X.xu u * k) → ∫⁻ x, φ x ∂X.B.μf = ENNReal.ofReal c * ∫⁻ t, ∫⁻ k in X.K2, φ (t.1 * X.xu u * k) ∂X.B.μf ∂μSf

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarMeasureDisintegrationBesselGroup

theorem **disint** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A), ∃ c : ℝ, 0 < c ∧ ∀ G : X.B.I.GA → ENNReal, Measurable G → (∀ x, G x ≠ 0 → ∃ s ∈ SfinU X u, ∃ k ∈ X.K2, x.2 = s * X.xu u * k) → ∫⁻ x, G x ∂(X.B.I.muA X.B.μf) = ENNReal.ofReal c * ∫⁻ y, ∫⁻ s, ∫⁻ k in X.K2, G (iotaX X h s.1 * (y, X.xu u) * (1, k)) ∂X.B.μf ∂B.lev.μR ∂(X.nuQ u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarMeasureDisintegrationBesselGroupStatement

theorem **_root_.GlobalTheory.T.haarMeasure_disintegration_besselGroup** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A), ∃ c : ℝ, 0 < c ∧ ∀ G : X.B.I.GA → ENNReal, Measurable G → (∀ x, G x ≠ 0 → ∃ s ∈ SfinU X u, ∃ k ∈ X.K2, x.2 = s * X.xu u * k) → ∫⁻ x, G x ∂(X.B.I.muA X.B.μf) = ENNReal.ofReal c * ∫⁻ y, ∫⁻ s, ∫⁻ k in X.K2, G (iotaX X h s.1 * (y, X.xu u) * (1, k)) ∂X.B.μf ∂B.lev.μR ∂(X.nuQ u)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarMeasureQuotientGroupProduct

theorem **quot_congr** {f f' : G₁ → ℂ} {z z' : ℂ} (h : IsQuotIntL1 ρ₁ μ₁ f z) (hf : ∀ x, f x = f' x) (hz : z = z') : IsQuotIntL1 ρ₁ μ₁ f' z'

theorem **iterated_integral_congr** {f f' : G₁ → G₂ → ℂ} {z z' : ℂ} (h : Iter2 ρ₁ μ₁ ρ₂ μ₂ f z) (hf : ∀ x y, f x y = f' x y) (hz : z = z') : Iter2 ρ₁ μ₁ ρ₂ μ₂ f' z'

theorem **iterated_integral_zero** : Iter2 ρ₁ μ₁ ρ₂ μ₂ 0 0

theorem **iterated_integral_add** {f f' : G₁ → G₂ → ℂ} {z z' : ℂ} (h : Iter2 ρ₁ μ₁ ρ₂ μ₂ f z) (h' : Iter2 ρ₁ μ₁ ρ₂ μ₂ f' z') : Iter2 ρ₁ μ₁ ρ₂ μ₂ (f + f') (z + z')

theorem **iterated_integral_mul** {u : G₁ → ℂ} {v : G₂ → ℂ} {c d : ℂ} (hu : IsQuotIntL1 ρ₁ μ₁ u c) (hv : IsQuotIntL1 ρ₂ μ₂ v d) : Iter2 ρ₁ μ₁ ρ₂ μ₂ (fun x y => u x * v y) (c * d)

theorem **iterated_integral_unique** {G₁ G₂ : Type} [Group G₁] [TopologicalSpace G₁] [IsTopologicalGroup G₁] [SecondCountableTopology G₁] [LocallyCompactSpace G₁] [T2Space G₁] [MeasurableSpace G₁] [BorelSpace G₁] [Group G₂] [TopologicalSpace G₂] [IsTopologicalGroup G₂] [SecondCountableTopology G₂] [LocallyCompactSpace G₂] [T2Space G₂] [MeasurableSpace G₂] [BorelSpace G₂] {R₁ : Subgroup G₁} {R₂ : Subgroup G₂} (hR₁ : IsClosed (R₁ : Set G₁)) (hR₂ : IsClosed (R₂ : Set G₂)) {ρ₁ : Measure R₁} {μ₁ : Measure G₁} {ρ₂ : Measure R₂} {μ₂ : Measure G₂} [μ₁.IsHaarMeasure] [μ₂.IsHaarMeasure] [ρ₁.IsHaarMeasure] [ρ₂.IsHaarMeasure] [ρ₁.IsInvInvariant] [ρ₂.IsInvInvariant] {f : G₁ → G₂ → ℂ} {z z' : ℂ} (h : Iter2 ρ₁ μ₁ ρ₂ μ₂ f z) (h' : Iter2 ρ₁ μ₁ ρ₂ μ₂ f z') : z = z'

theorem **quot_prod** {r : Measure R} {s : Measure S} {a : Measure A} {b : Measure B} [SFinite r] [SFinite s] [SFinite a] [SFinite b] (p : Measure (R.prod S)) (hp : p.map (R.prodEquiv S) = r.prod s) {u : A → ℂ} {v : B → ℂ} {c d : ℂ} (hu : IsQuotIntL1 r a u c) (hv : IsQuotIntL1 s b v d) : IsQuotIntL1 p (a.prod b) (fun X => u X.1 * v X.2) (c * d)

theorem **mul_mem_cell_left** {R K : Subgroup G} {t x s : G} (hs : s ∈ R) : s * x ∈ cell R K t ↔ x ∈ cell R K t

theorem **mul_mem_cell_right** {R K : Subgroup G} {t x k : G} (hk : k ∈ K) : x * k ∈ cell R K t ↔ x ∈ cell R K t

theorem **decomp** {M : Type} [AddCommMonoid M] (R K : Subgroup G) (T : Finset G) : ∀ φ : G → M, (∀ r ∈ R, ∀ x, φ (r * x) = φ x) → (∀ κ ∈ K, ∀ x, φ (x * κ) = φ x) → (∀ x, φ x ≠ 0 → ∃ t ∈ T, x ∈ cell R K t) → ∃ c : G → M, (∀ t, c t = 0 ∨ c t = φ t) ∧ ∀ x, φ x = ∑ t ∈ T, (cell R K t).indicator (fun _ => c t) x

theorem **indicator_fun_apply** {X Y : Type} (s : Set X) (c : Y → ℂ) (x : X) (y : Y) : (s.indicator (fun _ => c) x) y = s.indicator 1 x * c y

theorem **indicator_const_eq** {X : Type} (s : Set X) (m : ℂ) (x : X) : s.indicator (fun _ => m) x = s.indicator 1 x * m

theorem **mem_cell_prod** {A B : Type} [Group A] [Group B] (R K : Subgroup A) (S L : Subgroup B) (t X : A × B) : X ∈ cell (R.prod S) (K.prod L) t ↔ X.1 ∈ cell R K t.1 ∧ X.2 ∈ cell S L t.2

theorem **chi_prod** {A B : Type} [Group A] [Group B] (R K : Subgroup A) (S L : Subgroup B) (t X : A × B) : chi (R.prod S) (K.prod L) t X = chi R K t.1 X.1 * chi S L t.2 X.2

theorem **quot_chi** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [SecondCountableTopology G] [LocallyCompactSpace G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (R K : Subgroup G) (hR : IsClosed (R : Set G)) (hK : IsCO K) (ρ : Measure R) (μ : Measure G) [ρ.IsHaarMeasure] [ρ.IsInvInvariant] [μ.IsHaarMeasure] (t : G) : ∃ w, IsQuotIntL1 ρ μ (chi R K t) w

theorem **iterated_integral_product_groups** : ∀ (A₁ : Type) [Group A₁] [TopologicalSpace A₁] [IsTopologicalGroup A₁] [SecondCountableTopology A₁] [LocallyCompactSpace A₁] [T2Space A₁] [MeasurableSpace A₁] [BorelSpace A₁] (B₁ : Type) [Group B₁] [TopologicalSpace B₁] [IsTopologicalGroup B₁] [SecondCountableTopology B₁] [LocallyCompactSpace B₁] [T2Space B₁] [MeasurableSpace B₁] [BorelSpace B₁] (A₂ : Type) [Group A₂] [TopologicalSpace A₂] [IsTopologicalGroup A₂] [SecondCountableTopology A₂] [LocallyCompactSpace

A₂] [T2Space A₂] [MeasurableSpace A₂] [BorelSpace A₂] (B₂ : Type) [Group B₂] [TopologicalSpace B₂] [IsTopologicalGroup B₂] [SecondCountableTopology B₂] [LocallyCompactSpace B₂] [T2Space B₂] [MeasurableSpace B₂] [BorelSpace B₂] (R₁ : Subgroup A₁) (S₁ : Subgroup B₁) (R₂ : Subgroup A₂) (S₂ : Subgroup B₂), IsClosed (R₁ : Set A₁) → IsClosed (S₁ : Set B₁) → IsClosed (R₂ : Set A₂) → IsClosed (S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarMeasureQuotientGroupProductStatement

theorem **_root_.GlobalTheory.T.haar_measure_quotient_group_product** : ∀ (A₁ : Type) [Group A₁] [TopologicalSpace A₁] [IsTopologicalGroup A₁] [SecondCountableTopology A₁] [LocallyCompactSpace A₁] [T2Space A₁] [MeasurableSpace A₁] [BorelSpace A₁] (B₁ : Type) [Group B₁] [TopologicalSpace B₁] [IsTopologicalGroup B₁] [SecondCountableTopology B₁] [LocallyCompactSpace B₁] [T2Space B₁] [MeasurableSpace B₁] [BorelSpace B₁] (A₂ : Type) [Group A₂] [TopologicalSpace A₂] [IsTopologicalGroup A₂] [SecondCountableTopology A₂] [LocallyCompactSpace A₂] [T2Space A₂] [MeasurableSpace A₂] [BorelSpace A₂] (B₂ : Type) [Group B₂] [TopologicalSpace B₂] [IsTopologicalGroup B₂] [SecondCountableTopology B₂] [LocallyCompactSpace B₂] [T2Space B₂] [MeasurableSpace B₂] [BorelSpace B₂] (R₁ : Subgroup A₁) (S₁ : Subgroup B₁) (R₂ : Subgroup A₂) (S₂ : Subgroup B₂), IsClosed (R₁ : Set A₁) → IsClosed (S₁ : Set B₁) → IsClosed (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarOpenSubgroupVolumeInvEqRelIndex

theorem **measure_toReal_inv_eq_relIndex** : ∀ {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] (μ : MeasureTheory.Measure G) [μ.IsHaarMeasure] (H L : Subgroup G), H ≤ L → IsOpen (H : Set G) → IsCompact (L : Set G) → μ (L : Set G) = 1 → (μ (H : Set G)).toReal⁻¹ = (H.relIndex L : ℝ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarOpenSubgroupVolumeInvEqRelIndexStatement

theorem **_root_.GlobalTheory.T.haar_open_subgroup_volume_inv_eq_relIndex** : ∀ {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] (μ : MeasureTheory.Measure G) [μ.IsHaarMeasure] (H L : Subgroup G), H ≤ L → IsOpen (H : Set G) → IsCompact (L : Set G) → μ (L : Set G) = 1 → (μ (H : Set G)).toReal⁻¹ = (H.relIndex L : ℝ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarVolumeRatioCompactEqRelIndexRatio

theorem **relIndex_mul_measure_sub** {G : Type*} [Group G] [MeasurableSpace G] [MeasurableMul G] (H L : Subgroup G) (hHL : H ≤ L) [(H.subgroupOf L).FiniteIndex] (hH : MeasurableSet (H : Set G)) (μ : Measure G) [μ.IsMulLeftInvariant] : (H.relIndex L : ENNReal) * μ H = μ L

theorem **vol_ratio** : ∀ {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] (μ : MeasureTheory.Measure G) [μ.IsHaarMeasure] (A B : Subgroup G), IsOpen ((A ⊓ B : Subgroup G) : Set G) → IsCompact (A : Set G) → IsCompact (B : Set G) → (μ (A : Set G)).toReal / (μ (B : Set G)).toReal = (B.relIndex A : ℝ) / (A.relIndex B : ℝ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HaarVolumeRatioCompactEqRelIndexRatioStatement

theorem **_root_.GlobalTheory.T.haar_volume_ratio_compact_eq_relIndex_ratio** : ∀ {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] (μ : MeasureTheory.Measure G) [μ.IsHaarMeasure] (A B : Subgroup G), IsOpen ((A ⊓ B : Subgroup G) : Set G) → IsCompact (A : Set G) → IsCompact (B : Set G) → (μ (A : Set G)).toReal / (μ (B : Set G)).toReal = (B.relIndex A : ℝ) / (A.relIndex B : ℝ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfAComponentsMemCompletedLatticeOffPrimes

theorem **conjP_not_mem** {D : D7} {e : AE D} {SS : Finset (Formalism.L2.DIV.Pl D.host.E)} (he : IsIdemSum e SS) {P : HeightOneSpectrum (𝒪 D.host.E)} (hP : P ∈ SS) : Formalism.L2.Col.conjPrime₂ P ∉ SS

theorem **halfA_smul_mem_locP** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} (hc : 0 < c) {x : Formalism.L2.Col.HomE E a b} (hx : x ∈ halfAO SS M c) (Q : HeightOneSpectrum (𝒪 E)) : (c : E) • x ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M

theorem **halfA_mem_locP_conj** {D : D7} {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 D.host.E))} {M : Submodule (𝒪 D.host.E) (Formalism.L2.Col.HomE D.host.E a b)} {c : ℕ} (hc : 0 < c) {x : Formalism.L2.Col.HomE D.host.E a b} (hx : x ∈ halfAO SS M c) {P : HeightOneSpectrum (𝒪 D.host.E)} (hP : P ∈ SS) (hP' : Formalism.L2.Col.conjPrime₂ P ∉ SS) : x ∈ Formalism.HermitianLatticesTheory.localization₂ (Formalism.L2.Col.conjPrime₂ P).asIdeal M

theorem **halfA_mem_locP_off** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} {x : Formalism.L2.Col.HomE E a b} (hx : x ∈ halfAO SS M c) (Q : HeightOneSpectrum (𝒪 E)) (hQ : ¬ (Q ∈ SS ∨ Formalism.L2.Col.conjPrime₂ Q ∈ SS)) : x ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M

theorem **zmem** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, (GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) - ((LinearMap.toMatrix' x').map fun r => e * incA D r) ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfAComponentsMemCompletedLatticeOffPrimesStatement

theorem **_root_.GlobalTheory.T.halfA_components_mem_completedLattice_offPrimes** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, (GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) - ((LinearMap.toMatrix' x').map fun r => e * incA D r) ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfAIdempotentComponentCocycleTrivial

theorem **natCast_smul_mem_of_halfAO** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} (hc : 0 < c) {x : Formalism.L2.Col.HomE E a b} (hx : x ∈ halfAO SS M c) : (c : E) • x ∈ M

theorem **conjP_conjP** (Q : HeightOneSpectrum (𝒪 D.host.E)) : Formalism.L2.Col.conjPrime₂ (Formalism.L2.Col.conjPrime₂ Q) = Q

theorem **conjP_not_mem** {e : AE D} {SS : Finset (Formalism.L2.DIV.Pl D.host.E)} (h : IsIdemSum e SS) {PP : HeightOneSpectrum (𝒪 D.host.E)} (hPP : PP ∈ SS) : Formalism.L2.Col.conjPrime₂ PP ∉ SS

theorem **smul_tmul_fin** {a : AE D} (ha : infPart D a = 0) (r : D.host.E) : a * ((pairA D.host.F 0 1) ⊗ₜ[D.host.F] r) = a * incA D r

theorem **symplOH_smul** (a : AE D) (x y : WO S T) : symplOH S T (a • x) y = symplOH S T x (cA D a • y)

theorem **lhs_eq** {e : AE D} (hinf : infPart D (cA D e) = 0) (hec : e * cA D e = 0) (X : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : symplOH S T (X.map fun r => e * incA D r) (GlobalTheory.VacUnique.finRat S T X - X.map fun r => e * incA D r) = symplOH S T (X.map (incA D)) (cA D e • X.map (incA D))

theorem **smul_finRat_natCast** {e : AE D} (hinf : infPart D e = 0) (c : ℕ) (X : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : e • GlobalTheory.VacUnique.finRat S T ((c : D.host.E) • X) = (e * incA D (c : D.host.E)) • (X.map (incA D) : WO S T)

theorem **smul_map_inv** {e : AE D} (hcc : cA D e * cA D e = cA D e) (c : ℕ) (X : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : cA D e • (((c : D.host.E)⁻¹ • X).map fun r => cA D e * incA D r : WO S T) = (cA D e * incA D ((c : D.host.E)⁻¹)) • (X.map (incA D) : WO S T)

theorem **cA_incA_natCast** (c : ℕ) : cA D (incA D (c : D.host.E)) = incA D (c : D.host.E)

theorem **rhs_eq** {e : AE D} (hinf : infPart D e = 0) (hcc : cA D e * cA D e = cA D e) {c : ℕ} (hc : 0 < c) (X : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : symplOH S T (e • GlobalTheory.VacUnique.finRat S T ((c : D.host.E) • X)) (cA D e • (((c : D.host.E)⁻¹ • X).map fun r => cA D e * incA D r : WO S T)) = symplOH S T (X.map (incA D)) (cA D e • X.map (incA D))

theorem **finRat_mem** {N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (hcc : D.host.ccK = ⊤) (hN : IsSelfDualC S N) {SS : Finset (HeightOneSpectrum (𝒪 D.host.E))} {c : ℕ} (hc : 0 < c) {x' : Formalism.L2.Col.HomE D.host.E S.rank T.rank} (hx : x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c) : GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' ((c : D.host.E) • x')) ∈ latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **conjHalf_mem** {N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (hcc : D.host.ccK = ⊤) (hN : IsSelfDualC S N) {e : AE D} {SS : Finset (Formalism.L2.DIV.Pl D.host.E)} (hE : IsIdemSum e SS) {c : ℕ} {x' : Formalism.L2.Col.HomE D.host.E S.rank T.rank} (hx : x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c) : ((LinearMap.toMatrix' ((c : D.host.E)⁻¹ • x')).map fun r => cA D e * incA D r : WO S T) ∈ latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **cocycle_trivial** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ψ (symplOH S T ((LinearMap.toMatrix' x').map fun r => e * incA D r) ((GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) - ((LinearMap.toMatrix' x').map fun r => e * incA D r))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfAIdempotentComponentCocycleTrivialStatement

theorem **_root_.GlobalTheory.T.halfA_idempotent_component_cocycle_trivial** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ψ (symplOH S T ((LinearMap.toMatrix' x').map fun r => e * incA D r) ((GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) - ((LinearMap.toMatrix' x').map fun r => e * incA D r))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfASignTraceIsInteger

theorem **alV_embH_eq_exp_trace** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ) (hP : StdPair θ S T δ sc ψ) (y : Formalism.L2.Col.HomE D.host.E S.rank T.rank) : alV D.host.d sc (embH (tauE D) δ y) = Complex.exp (-(Real.pi : ℂ) * Complex.I * ((Algebra.trace ℚ D.host.E (D.host.xi0 * ((1 + D.host.sqrtMinusD) / 2) ^ 2 * GlobalTheory.LatVac.hRat S T (LinearMap.toMatrix' y) (LinearMap.toMatrix' y)) : ℚ) : ℂ))

theorem **trace_int** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ∃ k : ℤ, (k : ℚ) = Algebra.trace ℚ D.host.E (D.host.xi0 * ((1 + D.host.sqrtMinusD) / 2) ^ 2 * GlobalTheory.LatVac.hRat S T (LinearMap.toMatrix' x') (LinearMap.toMatrix' x'))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfASignTraceIsIntegerStatement

theorem **_root_.GlobalTheory.T.halfA_signTrace_isInteger** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, ∃ k : ℤ, (k : ℚ) = Algebra.trace ℚ D.host.E (D.host.xi0 * ((1 + D.host.sqrtMinusD) / 2) ^ 2 * GlobalTheory.LatVac.hRat S T (LinearMap.toMatrix' x') (LinearMap.toMatrix' x'))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfAvgLinear

theorem **halfAvg_linear** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ ℱ → ∀ e, IsSplitIdem D e → ∀ ℓ₀ : (WO S₀ T → ℂ) →ₗ[ℂ] ℂ, ∀ H : AddSubgroup (Mx D T.rank ρ), (∀ x, x ∈ H ↔ ∀ i j, cA D e * x i j = x i j) → ∀ [MeasurableSpace H] [BorelSpace H] (ν : Measure H) [ν.IsAddHaarMeasure], ∃ ℓ : (WO S T → ℂ) →ₗ[ℂ] ℂ, ∀ f, IsThetaFunO ψ S T f → ℓ f = ∫ x : H, ℓ₀ (d.mixS T ψ μ ℱ f x.1) ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfBComponentsMemCompletedLatticeOffConj

theorem **comp_mem** {D : D7} {S T : HSp D} {N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (hc : D.host.ccK = ⊤) (hN : IsSelfDualC S N) {e : AE D} {P : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt e P) (h0 : infPart D e = 0) {x : Formalism.L2.Col.HomE D.host.E S.rank T.rank} (hx : x ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N')) : ((LinearMap.toMatrix' x).map fun r => e * incA D r : WO S T) ∈ latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **halfB_mem_locP** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} {SS : Finset (HeightOneSpectrum (𝒪 E))} {M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)} {c : ℕ} (hc : 0 < c) {y : Formalism.L2.Col.HomE E a b} (hy : y ∈ halfBO SS M c) {P : HeightOneSpectrum (𝒪 E)} (hP : P ∈ SS) : y ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal M

theorem **ybmem** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ y' ∈ halfBO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, (GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' y')) - ((LinearMap.toMatrix' y').map fun r => cA D e * incA D r) ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfBComponentsMemCompletedLatticeOffConjStatement

theorem **_root_.GlobalTheory.T.halfB_components_mem_completedLattice_offConj** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ y' ∈ halfBO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, (GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' y')) - ((LinearMap.toMatrix' y').map fun r => cA D e * incA D r) ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfCocycleIsotropicIdempotentHalvesTrivial

theorem **half_eq** (D : D7) : incA D ((2 : D.host.E)⁻¹) = algebraMap (AF D.host.F) (AE D) (algebraMap D.host.F (AF D.host.F) (1 / 2))

theorem **half_mul_symplO** {D : D7} (S T : HSp D) (x y : WO S T) : algebraMap D.host.F (AF D.host.F) (1 / 2) * symplO S T x y = symplO S T (incA D ((2 : D.host.E)⁻¹) • x) y

theorem **half_idem_int** {D : D7} {e : AE D} {SS : Finset (Formalism.L2.DIV.Pl D.host.E)} (hI : IsIdemSum e SS) (h2 : ∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) : finPart D (incA D ((2 : D.host.E)⁻¹) * e) ∈ intAE D

theorem **GlobalTheory.HalfCocycle.char_symplOH_idem_conj** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ u ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True), ∀ v ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True), ψ (symplOH S T (e • u) (cA D e • v)) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfCocycleIsotropicIdempotentHalvesTrivialStatement

theorem **_root_.GlobalTheory.T.halfCocycle_isotropic_idempotentHalves_trivial** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ u ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True), ∀ v ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True), ψ (symplOH S T (e • u) (cA D e • v)) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfGeneratedPointActsScalarOnLatticeVacua

theorem **addChar_ne_zero** {A : Type} [AddCommGroup A] (ψ : AddChar A ℂ) (a : A) : ψ a ≠ 0

theorem **exists_scalar** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Λ : AddSubgroup (WO S T)) (z : WO S T), z ∈ AddSubgroup.closure {u | u ∈ Λ ∧ ∃ e, IsSplitIdem D e ∧ IsHalf e u} → ∃ c : ℂ, ∀ f, IsLatVac (heisO ψ S T) Λ f → heisO ψ S T z f = c • f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfGeneratedPointActsScalarOnLatticeVacuaStatement

theorem **_root_.GlobalTheory.T.halfGenerated_point_acts_scalar_on_latticeVacua** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Λ : AddSubgroup (WO S T)) (z : WO S T), z ∈ AddSubgroup.closure {u | u ∈ Λ ∧ ∃ e, IsSplitIdem D e ∧ IsHalf e u} → ∃ c : ℂ, ∀ f, IsLatVac (heisO ψ S T) Λ f → heisO ψ S T z f = c • f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfMatrixMoveDeepCongruenceBound

theorem **one_entry_le** {n : ℕ} (r s : Fin n) : v ((1 : Matrix (Fin n) (Fin n) R) r s) ≤ 1

theorem **inv_entry_le** {n : ℕ} (X Y : Matrix (Fin n) (Fin n) R) (hXY : X * Y = 1) {ε : Γ} (hε : ε < 1) (hX : ∀ r s, v ((X - 1) r s) ≤ ε) : ∀ r s, v ((Y - 1) r s) ≤ ε

theorem **moved_entry_le** {a b : ℕ} (X : Matrix (Fin a) (Fin a) R) (Y : Matrix (Fin b) (Fin b) R) (M : Matrix (Fin b) (Fin a) R) {ε : Γ} (hε : ε ≤ 1) (hX : ∀ r s, v ((X - 1) r s) ≤ ε) (hY : ∀ r s, v ((Y - 1) r s) ≤ ε) (hM : ∀ r s, v (M r s) ≤ 1) : ∀ r s, v ((Y * M * X - M) r s) ≤ ε

theorem **exists_exp_le_natCast** {E : Type} [Field E] [NumberField E] (L : ℕ) (hL : 0 < L) : ∃ N₀ : ℕ, 1 ≤ N₀ ∧ ∀ P : HeightOneSpectrum (𝒪 E), WithZero.exp (-(N₀ : ℤ)) ≤ Valued.v ((L : ℕ) : P.adicCompletion E)

theorem **map_sub_one_apply** {R R' : Type} [CommRing R] [CommRing R'] {n : ℕ} (f : R →+* R') (A : Matrix (Fin n) (Fin n) R) (r s : Fin n) : f ((A - 1) r s) = (A.map f - 1) r s

theorem **exists_den_mat** {D : D7} {m n : ℕ} (M : Matrix (Fin m) (Fin n) (AE D)) : ∃ d : ℕ, 0 < d ∧ ∀ r s, toEAf D ((d : AE D) * M r s) ∈ Formalism.L2.DIV.RInt D.host.F D.host.E

theorem **actO_deep** : ∀ (D : D7) (S T : HSp D) {a b : ℕ} (ha : S.rank = a) (hb : T.rank = b) (e : AE D) (M : WO S T), IsHalf e M → ∀ (k : ℕ), 0 < k → ∃ N₀ : ℕ, ∀ (w : GA D S) (w' : GA D T), w ∈ GfC S → w' ∈ GfC T → (∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (r s : Fin a), Valued.v (adAt D.host.F P (finM S ha w r s - (1 : Matrix (Fin a) (Fin a) (EF D)) r s)) ≤ WithZero.exp (-(N₀ : ℤ))) → (∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (r s : Fin b), Valued.v (adAt D.host.F P (finM T hb w' r s - (1 : Matrix (Fin b) (Fin b) (EF D)) r s)) ≤ WithZero.exp (-(N₀ : ℤ))) → ∃ t : WO S T, t ∈ stdHom S T ∧ IsHalf e t ∧ actO S T w⁻¹ w'⁻¹ M = M + k • t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HalfMatrixMoveDeepCongruenceBoundStatement

theorem **_root_.GlobalTheory.T.halfMatrix_move_deepCongruence_bound** : ∀ (D : D7) (S T : HSp D) {a b : ℕ} (ha : S.rank = a) (hb : T.rank = b) (e : AE D) (M : WO S T), IsHalf e M → ∀ (k : ℕ), 0 < k → ∃ N₀ : ℕ, ∀ (w : GA D S) (w' : GA D T), w ∈ GfC S → w' ∈ GfC T → (∀ (P :

HeightOneSpectrum (𝒪 D.host.E)) (r s : Fin a), Valued.v (adAt D.host.F P (finM S ha w r s − (1 : Matrix (Fin a) (Fin a) (EF D)) r s)) ≤ WithZero.exp (−(N₀ : ℤ))) → (∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (r s : Fin b), Valued.v (adAt D.host.F P (finM T hb w' r s − (1 : Matrix (Fin b) (Fin b) (EF D)) r s)) ≤ WithZero.exp (−(N₀ : ℤ))) → ∃ t : WO S T, t ∈ stdHom S T ∧ IsHalf e t ∧ actO S T w⁻¹ w'⁻¹ M = M + k • t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HandBFSmul

theorem **mixS_smul** {V V₀ : HSp D} {ρ : ℕ} (d : HypDec V V₀ ρ) (T : HSp D) (ψ : AddChar (AF D.host.F) ℂ) (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)) (a : ℂ) (f : WO V T → ℂ) (x : Mx D T.rank ρ) : d.mixS T ψ μ ℱ (a • f) x = a • d.mixS T ψ μ ℱ f x

theorem **mixT_smul** {W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (S : HSp D) (ψ : AddChar (AF D.host.F) ℂ) (μ : Measure (Mx D ρ S.rank)) (ℱ : Set (Mx D ρ S.rank)) (a : ℂ) (f : WO S W → ℂ) (y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (a • f) y = a • d.mixT S ψ μ ℱ f y

theorem **flatAt_smul** (ψ : AddChar (AF D.host.F) ℂ) {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (μS : Measure (Mx D W.rank ρ)) (ℱS : Set (Mx D W.rank ρ)) (μT : Measure (Mx D σ V₀.rank)) (ℱT : Set (Mx D σ V₀.rank)) (νU : Measure (UX D ρ)) (x : Mx D W.rank ρ) (y : Mx D V₀.rank σ) (a : ℂ) (Ψ : WO V W → ℂ) : GlobalTheory.MixPer.flatAt ψ dV dW μS ℱS μT ℱT νU x y (a • Ψ) = a • GlobalTheory.MixPer.flatAt ψ dV dW μS ℱS μT ℱT νU x y Ψ

theorem **flatBF_smul** {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) (a : ℂ) (Ψ : WO B.V J.W → ℂ) : flatBF s (a • Ψ) = a • flatBF s Ψ

theorem **flatFB_smul** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (a : ℂ) (Ψ : WO J.W B.V → ℂ) : flatFB s (a • Ψ) = a • flatFB s Ψ

theorem **handBF_smul** {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (φ : GA D B.A → ℂ) (a : ℂ) (Ξ : WO B.V₀ J.W₀ → ℂ) : handBF s l φ (a • Ξ) = star a • handBF s l φ Ξ

theorem **handBF_zero** {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {L' : HSp D} {ζ' : Fin L'.rank → D.host.E} (l : Link B L' ζ') (φ : GA D B.A → ℂ) : handBF s l φ 0 = 0

theorem **handFB_smul** {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) (vL : VolM D B.L.rank J.W₀.rank) (c b : ℂ) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (Ξ : WO J.W₀ B.V₀ → ℂ) : handFB s vL (c • Y) (b • Ξ) = (star c ∗ b) • handFB s vL Y Ξ

theorem **per_smul** {ρ : ℕ} (B : BesLev D ρ) (ψ : AddChar (AF D.host.F) ℂ) (a b : ℂ) (φ : GA D B.A → ℂ) (f : GA D B.V → ℂ) : B.per ψ (a • φ) (b • f) = a ∗ b ∗ B.per ψ φ f

theorem **hand_hom** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (B : BesLev D (σ + 1)) (J : JacLev D (σ + 1)) (B' : BesLev D σ) (s₁ : StepBF ψ B J) (s₂ : StepFB ψ J B') (l : Link B B'.L B'.ζ) (vL : VolM D B'.L.rank J.W₀.rank) (φ : GA D B.A → ℂ), (∀ (Ψ₁ : WO B.V J.W → ℂ) (Ψ₂ : WO J.W B'.V → ℂ) (a b : ℂ), handFB s₂ vL (handBF s₁ l φ (flatBF s₁ (a • Ψ₁))) (flatFB s₂ (b • Ψ₂)) = (a ∗ b) • handFB s₂ vL (handBF s₁ l φ (flatBF s₁ Ψ₁)) (flatFB s₂ Ψ₂)) ∧ (∀ Ξ₂, handFB s₂ vL (handBF s₁ l φ 0) Ξ₂ = 0) ∧ ∀ (a b : ℂ) (φ' : GA D B'.A → ℂ) (f : GA D B'.V → ℂ), B'.per ψ (a • φ') (b • f) = a ∗ b ∗ B'.per ψ φ' f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HarishChandraModulesIndefiniteUnitaryGroup

theorem **GlobalTheory.HcPort.kfinDense_UJC** {n : ℕ} {ε : Fin n → Bool} {H : Type} [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] {σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)} (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) : Dense {v : H | Formalism.MainNotions.S.IsKFin ε σ v}

theorem **exists_kfin_near_A** [CompleteSpace H] (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) {w : H} (_hw : w ∈ smD ε σ) {δ : ℝ} (hδ : 0 < δ) : ∃ a : H, Formalism.MainNotions.S.IsKFin ε σ a ∧ ‖a − w‖ < δ

theorem **eq_zero_of_perp_A** [CompleteSpace H] (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) {w : H} (hw : w ∈ smD ε σ) (h : ∀ a : H, Formalism.MainNotions.S.IsKFin ε σ a → ‹a, w›_ℂ = 0) : w = 0

theorem **casimir_symm_A** (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) (x y : smD ε σ) : ‹casimir (smE ε σ) x, y›_ℂ = ‹x, casimir (smE ε σ) y›_ℂ

theorem **gen_eq_A** (p : Fin n × Fin n) : Formalism.L2.SG.Pf.admitted.gen ε p = casGen ε p

theorem **coef_eq_A** (p : Fin n × Fin n) : (((Formalism.L2.SG.Pf.admitted.wt p ∗ Formalism.L2.SG.Pf.admitted.sg ε p : ℝ)) : ℂ) = casWt p ∗ casSg ε p

theorem **cas_eq_A** (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) (v : smD ε σ) : Formalism.L2.SG.Pf.admitted.Cas ε σ v = ((casimir (smE ε σ) v : smD ε σ) : H)

theorem **admissibleProjection_of_irreducible_indefUnitary** [CompleteSpace H] (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) (hirr : Formalism.MainNotions.S.IsIrredOn (Formalism.MainNotions.S.UJset ε) σ) : Formalism.MainNotions.BridgeCPX.AdmProjAt ε σ

theorem **casimirActsByScalar_indefUnitary** : CasScal

theorem **kFiniteWordBound_indefUnitary** : WordBd

theorem **kFiniteVectors_analytic_indefiniteUnitary** : GlobalTheory.Published.KFiniteAnalyticUJ

theorem **clause1_of** (hS : KFSm) (hW : IntWts) (hI : UIrr) (hA : UAdm) : Clause1

theorem **harishChandraThird_indefiniteUnitary** : Clause3

theorem **graphD** [CompleteSpace H₁] [CompleteSpace H₂] (h₁ : IsURepOn (UJset ε) σ₁) (h₂ : IsURepOn (UJset ε) σ₂) (hD₁ : IsDerivedOn ε σ₁ D₁ E₁) (hD₂ : IsDerivedOn ε σ₂ D₂ E₂) {N : Type} [AddCommGroup N] [Module ℂ N] (F : Fin n → Fin n → Module.End ℂ N) (κ₁ : N →ₗ[ℂ] D₁) (κ₂ : N →ₗ[ℂ] D₂) (hi : Function.Injective κ₁) (e₁ : ∀ a b w, E₁ a b (κ₁ w) = κ₁ (F a b w)) (e₂ : ∀ a b w, E₂ a b (κ₂ w) = κ₂ (F a b w)) {c : ℝ} (hc : ∀ v, casimir E₁ v = (c : ℂ) • v) (U : H₁ →L[ℂ] H₂) (hU : ∀ w, U (κ₁ w) = κ₂ w) {W : Submodule ℂ N} (hW : FiniteDimensional ℂ W) (hWK : IsKStableSub ε F W) {w : N} (hw : w ∈ W) {g : Matrix (Fin n) (Fin n) ℂ} (hg : g ∈ UJset ε) : U (σ₁ g (κ₁ w)) = σ₂ g (κ₂ w)

theorem **norm_detD** {g : Matrix (Fin n) (Fin n) ℂ} (hg : g ∈ Formalism.MainNotions.S.UJset ε) : ‖g.det‖ = 1

theorem **twD_rep** (m : ℕ) (h : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) σ) : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset ε) (twD m σ)

theorem **twD_irr** (m : ℕ) (h : Formalism.MainNotions.S.IsIrredOn (Formalism.MainNotions.S.UJset ε) σ) : Formalism.MainNotions.S.IsIrredOn (Formalism.MainNotions.S.UJset ε) (twD m σ)

theorem **twD_kfin** (m : ℕ) {v : H} (h : Formalism.MainNotions.S.IsKFin ε σ v) : Formalism.MainNotions.S.IsKFin ε (twD m σ) v

theorem **skewD** (h : IsHCMod πU M io) : SkewD M io

theorem **eigD** (h : IsHCMod πU M io) {X : Mat p q} (hX : X ∈ upq p q) {v : M.V} {c : ℂ} (hv : M.act X v = c • v) (t : ℝ) : πU.π (T.expt (t • X)) (io v) = Complex.exp (c * t) • io v

theorem **iE_memD** (a : Formalism.BasicNotions.TH.Ix p q) : Complex.I • Eab a a ∈ upq p q

theorem **exp_twoPiD** (a : Formalism.BasicNotions.TH.Ix p q) : NormedSpace.exp (((2 * Real.pi : ℝ) : ℂ) • rx (Complex.I • Eab a a)) = 1

theorem **signD** (h : IsHCMod πU M io) {j : ℕ} {σ : Matrix (Fin (p + q)) (Fin (p + q)) ℂ → (πU.H →L[ℂ] πU.H)} (h1 : σ 1 = 1) (hex : ∀ X ∈ upq p q, inG (epq p q) (rx X) ∧ ∀ t : ℝ, πU.π (T.expt (t • X)) = Complex.exp ((j : ℂ) * (t * X.trace / 2)) • σ (NormedSpace.exp ((t : ℂ) • rx X))) (a : Formalism.BasicNotions.TH.Ix p q) {v : M.V} (hv0 : v ≠ 0) {c : ℂ} (hv : M.act (Eab a a) v = c • v) : Complex.exp (Complex.I * c * (2 * Real.pi : ℝ)) = (-1) ^ j

theorem **dictD** (hS : KFSm) (h : IsHCMod πU M io) {j : ℕ} (m : ℕ) {σ : Matrix (Fin (p + q)) (Fin (p + q)) ℂ → (πU.H →L[ℂ] πU.H)} (hσ : Formalism.MainNotions.S.IsURepOn (Formalism.MainNotions.S.UJset (epq p q)) σ) (hσi : Formalism.MainNotions.S.IsIrredOn (Formalism.MainNotions.S.UJset (epq p q)) σ) (hkf : ∀ x, KFin T πU.π x ↔ Formalism.MainNotions.S.IsKFin (epq p q) σ x) (hex : ∀ X ∈ upq p q, inG (epq p q) (rx X) ∧ ∀ t : ℝ, πU.π (T.expt (t • X)) = Complex.exp ((j : ℂ) * (t * X.trace / 2)) • σ (NormedSpace.exp ((t : ℂ) • rx X))) : ∃ κ : M.V →ₗ[ℂ] smD (epq p q) (twD m σ), (∀ v, (κ v : πU.H) = io v) ∧ ∀ Z v, actM (smE (epq p q) (twD m σ)) (rx Z) (κ v) = κ (M.act Z v + (((m : ℂ) - j / 2) * Z.trace) • v)

theorem **clause2_of** (h1 : Clause1) (hS : KFSm) (hC : CasScal) (hDe : Descend) : Clause2

theorem **uIrr** : UIrr

theorem **uAdm** : UAdm

theorem **clause1** : Clause1

theorem **clause2** : Clause2

theorem **print_harishChandraModules** : GlobalTheory.Published.HarishChandraModules

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HarishChandraModulesIndefiniteUnitaryGroupStatement

theorem **_root_.GlobalTheory.T.harishChandraModules_indefiniteUnitaryGroup** : GlobalTheory.Published.HarishChandraModules

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HarishChandraNodesFarApartOfCubicNodeGap

theorem **harishChandraNodesFarApart_of_cubic_node_gap** : ∀ D : D7, D.NodeGap (4 * ((D.R.N D.R.m : ℕ) : ℝ) ^ 3 + 2 * D.R.t + 6) → GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.NodeGap D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HarmonicFockVectorsKilledByLoweringAdaptedBasis

theorem **exists_coef** {BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ} (hB : ∀ τ, (BS τ)ᴴ * gramAt D S τ * BS τ = 1) {p : ℕ} {B' : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hBT : IsONBasis (gramAt D T τ) (cart M τ) B' p) {X : Matrix (Fin T.rank) (Fin T.rank) ℂ} (lX : X ∈ lieAt T τ) (lY : Complex.I • (cart M τ * X) ∈ lieAt T τ) (ha : cart M τ * X = -(X * cart M τ)) : ∃ ν : Fin T.rank → Fin T.rank → ℂ, (∀ a b : Fin T.rank, ¬ (p ≤ (a : ℕ) ∧ (b : ℕ) < p) → ν a b = 0) ∧ ∀ a c b d, ∀ u, u = unitMap (BS τ) B' a c → ∀ u', u' = unitMap (BS τ) B' b d → ((omDir S T τ (X * u) τ u' : ℝ) + Complex.I * (omDir S T τ (cplxO 1 (cart M τ) (X * u)) τ u' : ℝ)) + (-Complex.I) * ((omDir S T τ (X * cplxO 1 (cart M τ) u) τ u' : ℝ) + Complex.I * (omDir S T τ (cplxO 1 (cart M τ) (X * cplxO 1 (cart M τ) u)) τ u' : ℝ)) = (if c = d then ν a b else 0) + (if d = c then ν b a else 0)

theorem **gen_kill** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ} (hB : ∀ τ, (BS τ)ᴴ * gramAt D S τ * BS τ = 1) {p : ℕ} {BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ} (hBT : IsONBasis (gramAt D T τ) (cart M τ) (BT τ) p) {X : Matrix (Fin T.rank) (Fin T.rank) ℂ} {GX GY : Module.End ℂ (WO S T → ℂ)} (hX : IsGen ψ om θ τ X GX) (hY : IsGen ψ om θ τ (Complex.I • (cart M τ * X)) GY) (lX : X ∈ lieAt T τ) (lY : Complex.I • (cart M τ * X) ∈ lieAt T τ) (ha : cart M τ * X = -(X * cart M τ)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hv : IsFockVacAt (heisO ψ S T) θ τ (cplxO 1 (cart M τ)) f) (h0 : (GX + Complex.I • GY) f = 0) {P : MvPolynomial (FVar D T.rank S.rank) ℂ} (hP : ∀ a b : Fin T.rank, p ≤ (a : ℕ) → (b : ℕ) < p → ∑ c : Fin S.rank, MvPolynomial.pderiv (τ, a, c) (MvPolynomial.pderiv (τ, b, c) P) = 0) : (GX + Complex.I • GY) (pureTens

theorem **killed_of_harmonic_adapted** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (Sp : D.what₂.Tau → SPar) (M : DSModel T Sp), S.SigOK → T.SigOK → ∀ (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ), (∀ τ, IsONBasis (gramAt D S τ) 1 (BS τ) S.rank) → ∀ p : D.what₂.Tau → ℕ, (∀ τ a, M.ε τ a = decide ((a : ℕ) < p τ)) → ∀ (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsONBasis (gramAt D T τ) (cart M τ) (BT τ) (p τ)) → ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO 1 (cart M τ)) f) → ∀ P : PolF D T.rank S.rank, (∀ τ, ∀ a b : Fin T.rank, p τ ≤ (a : ℕ) → (b : ℕ) < p τ → ∑ c : Fin S.rank, MvPolynomial.pderiv (τ, a, c) (MvPolynomial.pderiv (τ, b, c) P) = 0) → GlobalTheory.RawLift.FockCoef.K

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HarmonicFockVectorsKilledByLoweringAdaptedBasisStatement

theorem **_root_.GlobalTheory.T.harmonicFockVectors_killed_by_lowering_adaptedBasis** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (Sp : D.what₂.Tau → SPar) (M : DSModel T Sp), S.SigOK → T.SigOK → ∀ (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ), (∀ τ, IsONBasis (gramAt D S τ) 1 (BS τ) S.rank) → ∀ p : D.what₂.Tau → ℕ, (∀ τ a, M.ε τ a = decide ((a : ℕ) < p τ)) → ∀ (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ), (∀ τ, IsONBasis (gramAt D T τ) (cart M τ) (BT τ) (p τ)) → ∀ f : WO S T → ℂ, IsThetaFunO ψ S T f → (∀ τ, IsFockVacAt (heisO ψ S T) θ τ (cplxO 1 (cart M τ)) f) → ∀ P : PolF D T.rank S.rank, (∀ τ, ∀ a b : Fin T.rank, p τ ≤ (a : ℕ) → (b : ℕ) < p τ → ∑ c : Fin S.rank, MvPolynomial.pderiv (τ, a, c) (MvPolynomial.pderiv (τ,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtDOmT

theorem **conjAt_rel** {D : D7} (V : HSp D) (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : gramAt D V τ * Formalism.L2.Theta.conjAt V τ Z = -(Zᴴ * gramAt D V τ) ∧ (Formalism.L2.Theta.conjAt V τ Z)ᴴ * gramAt D V τ = -(gramAt D V τ * Z)

theorem **re_mem** {D : D7} (V : HSp D) (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : (1 / 2 : ℂ) • (Z + Formalism.L2.Theta.conjAt V τ Z) ∈ lieAt V τ

theorem **im_mem** {D : D7} (V : HSp D) (τ : D.what₂.Tau) (Z : Matrix (Fin V.rank) (Fin V.rank) ℂ) : (1 / (2 * Complex.I)) • (Z - Formalism.L2.Theta.conjAt V τ Z) ∈ lieAt V τ

theorem **hasDerivAt_dOmT** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : WRep S T} (hθ : IsInfFam θ) (hω : IsWeilDatumO ψ S T χS χT om) {τ : D.what₂.Tau} {X : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hX : X ∈ lieAt T τ) : (∀ f, IsThetaFunO ψ S T f → IsThetaFunO ψ S T (dOmT θ om τ X f)) ∧ (∀ f, IsThetaFunO ψ S T f → ∀ g h w, HasDerivAt (fun t : ℝ => om (g, h * oneP θ T τ X t) f w) (om (g, h) (dOmT θ om τ X f) w) 0) ∧ (∀ f, IsThetaFunO ψ S T f → ∀ w, HasDerivAt (fun t : ℝ => om (1, oneP θ T τ X t) f w) (dOmT θ om τ X f w) 0)

theorem **dOmT_add** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : WRep S T} (hθ : IsInfFam θ) (hω : IsWeilDatumO ψ S T χS χT om) {τ : D.what₂.Tau} {X : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hX : X ∈ lieAt T τ) {f f' : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hf' : IsThetaFunO ψ S T f') : dOmT θ om τ X (f + f') = dOmT θ om τ X f + dOmT θ om τ X f'

theorem **dOmT_smul** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : WRep S T} (hθ : IsInfFam θ) (hω : IsWeilDatumO ψ S T χS χT om) {τ : D.what₂.Tau} {X : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hX : X ∈ lieAt T τ) (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : dOmT θ om τ X (c • f) = c • dOmT θ om τ X f

theorem **GlobalTheory.Core.Rk.ZcDer.rderC_om** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : WRep S T), IsInfFam θ → IsWeilDatumO ψ S T χS χT om → ∀ (τ : D.what₂.Tau) (Z : Matrix (Fin T.rank) (Fin T.rank) ℂ), ∃ Dz : (WO S T → ℂ) → (WO S T → ℂ), (∀ f, IsThetaFunO ψ S T f → IsThetaFunO ψ S T (Dz f)) ∧ (∀ f f', IsThetaFunO ψ S T f → IsThetaFunO ψ S T f' → Dz (f + f') = Dz f + Dz f') ∧ (∀ (c : ℂ) f, IsThetaFunO ψ S T f → Dz (c • f) = c • Dz f) ∧ ∀ f, IsThetaFunO ψ S T f → ∀ g w, RderC θ T τ Z (fun h => om (g, h) f w) = fun h => om (g, h) (Dz f) w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtHeisenbergThetaSeries

theorem **hasDerivAt_term** {G : Type} [AddCommGroup G] (a : ℝ →+ G) (p : G →+ ℂ) (κ : ℂ) (hκ : ∀ t : ℝ, p (a t) = t * κ) (φ φ₁ : G → ℂ) (hφ : ∀ x, HasDerivAt (fun t : ℝ => φ (x + a t)) (φ₁ x) 0) (u : G) (s : ℝ) : HasDerivAt (fun t : ℝ => Complex.exp (t * p (a t) + t * p u + t * p u) * φ (u + a t)) (Complex.exp (s * p (a s) + s * p u + s * p u) * ((p + p) (u + a s) * φ (u + a s) + φ₁ (u + a s))) s

theorem **two_halves** (S T : HSp D) (x y : WO S T) : symplO S T x y = symplOH S T x y + symplOH S T x y

theorem **phase_identity** {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (x y a b ξ : Mx D T.rank ρ) : symplOH S T (d.extX T x + d.extY T y) (d.extX T a + d.extY T b) + (-(symplOH S T (d.extX T (x + a)) (d.extY T (y + b)))) + symplO S T (d.extX T (x + a + ξ)) (d.extY T (y + b)) = -(symplOH S T (d.extX T x) (d.extY T y)) + symplO S T (d.extX T (x + ξ)) (d.extY T y) + (symplOH S T (d.extX T a) (d.extY T b) + symplOH S T (d.extX T (x + ξ)) (d.extY T b) + symplOH S T (d.extX T (x + ξ)) (d.extY T b))

theorem **eq_ext** (A B : HSp D) (w : WO (dblSp A) B) : w = (dblDec A).extX B (HMul.hMul w (dblDec A).jX) + (dblDec A).extY B (HMul.hMul w (dblDec A).jY)

theorem **continuous_mulLine** {m n k : ℕ} {M : ℝ →+ Mx D m n} (hM : Continuous M) (j : Mx D n k) : Continuous (mulLine M j)

theorem **map_mul_zero** {m n k : ℕ} {R : Type} [CommRing R] (q : AE D →+* R) {u : Mx D m n} (hu : u.map q = 0) (j : Mx D n k) : (HMul.hMul u j : Mx D m k).map q = 0

theorem **exists_summable_bound** {θ : InfFam D} (hθ : IsInfFam θ) {m n : ℕ} {φ : Mx D m n → ℂ} (hφ : GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ) {Q : Set (Mx D m n)} (hQ : IsCompact Q) : ∃ b : Matrix (Fin m) (Fin n) D.host.E → ℝ, Summable b ∧ ∀ x ∈ Q, ∀ ξ, ‖φ (x + rat ξ)‖ ≤ b ξ

theorem **hasDerivAt_heisO_thetaSeries** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (φ : Mx D B.rank A.rank → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → ∀ f : WO (dblSp A) B → ℂ, (∀ w : WO (dblSp A) B, f w = ψ (-(symplOH (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY)))) * ∑' ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E, ψ (symplO (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX + rat ξ)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY))) * φ (HMul.hMul w (dblDec A).jX + rat ξ)) → ∀ M : ℝ →+ WO (dblSp A) B, Continuous M → (∀ t, (M t).map (finPart D) = 0) → ∃ φ' : Mx D B.rank A.rank → ℂ, GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ' ∧ ∀ f' : WO (dblSp A) B → ℂ, (∀ w : WO (dblSp A) B, f' w = ψ (-(sym

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtHeisenbergThetaSeriesStatement

theorem **_root_.GlobalTheory.T.hasDerivAt_heisenberg_thetaSeries** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (φ : Mx D B.rank A.rank → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → ∀ f : WO (dblSp A) B → ℂ, (∀ w : WO (dblSp A) B, f w = ψ (-(symplOH (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY)))) * ∑' ξ : Matrix (Fin B.rank) (Fin A.rank) D.host.E, ψ (symplO (dblSp A) B ((dblDec A).extX B (HMul.hMul w (dblDec A).jX + rat ξ)) ((dblDec A).extY B (HMul.hMul w (dblDec A).jY))) * φ (HMul.hMul w (dblDec A).jX + rat ξ)) → ∀ M : ℝ →+ WO (dblSp A) B, Continuous M → (∀ t, (M t).map (finPart D) = 0) → ∃ φ' : Mx D B.rank A.rank → ℂ, GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ' ∧ ∀ f' : WO (dblSp A) B → ℂ, (∀ w : WO (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtMixedModelSourceDualLineLinearForm

theorem **hasDerivAt_mixS_extY** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → ∀ L : ℝ →+ Mx D T.rank ρ, Continuous L → (∀ t, (L t).map (finPart D) = 0) → ∀ x w, HasDerivAt (fun t : ℝ => ψ (symplO S T (d.extX T x) (d.extY T (L t))) * d.mixS T ψ v.μ v.𝓕 f x w) (d.mixS T ψ v.μ v.𝓕 (fun p => deriv (fun t : ℝ => heisO ψ S T (d.extY T (L t)) f p) 0) x w) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtMixedModelSourceDualLineLinearFormStatement

theorem **_root_.GlobalTheory.T.hasDerivAt_mixedModelSource_dualLine_linearForm** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → ∀ L : ℝ →+ Mx D T.rank ρ, Continuous L → (∀ t, (L t).map (finPart D) = 0) → ∀ x w, HasDerivAt (fun t : ℝ => ψ (symplO S T (d.extX T x) (d.extY T (L t))) * d.mixS T ψ v.μ v.𝓕 f x w) (d.mixS T ψ v.μ v.𝓕 (fun p => deriv (fun t : ℝ => heisO ψ S T (d.extY T (L t)) f p) 0) x w) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtSobolevThetaFormHeisenbergAlongRealLin

theorem **hasDerivAt_form_heisO_line** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (ℓ : (WO S T → ℂ) → ℂ), IsSobForm ψ θ S T ℓ → ∀ f, IsThetaFunO ψ S T f → ∀ M : ℝ →+ WO S T, Continuous M → (∀ s, (M s).map (finPart D) = 0) → ∀ w : WO S T, HasDerivAt (fun s : ℝ => ℓ (heisO ψ S T (w + M s) f)) (ℓ (heisO ψ S T w (fun p => deriv (fun s : ℝ => heisO ψ S T (M s) f p) 0)) - deriv (fun s : ℝ => ψ (symplOH S T w (M s))) 0 * ℓ (heisO ψ S T w f)) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDerivAtSobolevThetaFormHeisenbergAlongRealLinStatement

theorem **_root_.GlobalTheory.T.hasDerivAt_sobolevThetaForm_heisenberg_along_real_line** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (ℓ : (WO S T → ℂ) → ℂ), IsSobForm ψ θ S T ℓ → ∀ f, IsThetaFunO ψ S T f → ∀ M : ℝ →+ WO S T, Continuous M → (∀ s, (M s).map (finPart D) = 0) → ∀ w : WO S T, HasDerivAt (fun s : ℝ => ℓ (heisO ψ S T (w + M s) f)) (ℓ (heisO ψ S T w (fun p => deriv (fun s : ℝ => heisO ψ S T (M s) f p) 0)) - deriv (fun s : ℝ => ψ (symplOH S T w (M s))) 0 * ℓ (heisO ψ S T w f)) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDiscreteSeriesKTypesCoverOfFactorsThroughCove

theorem **hasTypes_of_factors** : GlobalTheory.Dec.LowKMultOne → GlobalTheory.Dec.KTypesAbove → GlobalTheory.Dec.DSInfChar → ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (P : SignedPar n) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e P.sgn → Factors R.U e σ → IsDSPar P σ → HasTypes R (blockPar P e)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDiscreteSeriesKTypesCoverOfFactorsThroughCoveStatement

theorem **_root_.GlobalTheory.T.hasDiscreteSeriesKTypes_cover_of_factorsThroughCovering** : GlobalTheory.Dec.LowKMultOne → GlobalTheory.Dec.KTypesAbove → GlobalTheory.Dec.DSInfChar → ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (P : SignedPar n) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e P.sgn → Factors R.U e σ → IsDSPar P σ → HasTypes R (blockPar P e)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDiscreteSeriesKTypesOfHasParamCover

theorem **hasTypes_of_hasParam_of_group** : GlobalTheory.Dec.LowKMultOne → GlobalTheory.Dec.KTypesAbove → GlobalTheory.Dec.DSInfChar → ∀ (p q : ℕ), 1 ≤ p + q → ∀ (T : GrpData p q) (R : DSRep T) (lam : Ix p q → ℚ), IsParam lam → HasParam R lam → HasTypes R lam

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDiscreteSeriesKTypesOfHasParamCoverStatement

theorem **_root_.GlobalTheory.T.hasDiscreteSeriesKTypes_of_hasParam_cover** : GlobalTheory.Dec.LowKMultOne → GlobalTheory.Dec.KTypesAbove → GlobalTheory.Dec.DSInfChar → ∀ (p q : ℕ), 1 ≤ p + q → ∀ (T : GrpData p q) (R : DSRep T) (lam : Ix p q → ℚ), IsParam lam → HasParam R lam → HasTypes R lam

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDiscreteSeriesKTypesOfSqrtDetTwist

theorem **sum_rhoNC** (lam : Ix p q → ℚ) : ∑ a, rhoNC lam a = 0

theorem **normSq_add_half** (lam : Ix p q → ℚ) : normSq (fun a => lam a + 1 / 2) = normSq lam + ∑ a, (lam a + 1 / 2) - (Fintype.card (Ix p q) : ℚ) / 4

theorem **casimir_twAct** (act : Mat p q →ₗ[ℂ] Module.End ℂ V) (v : V) : casimir (Cover.twAct act) v = casimir act v - act 1 v + ((Fintype.card (Ix p q) : ℂ) / 4) • v

theorem **hwSub_twAct** (act : Mat p q →ₗ[ℂ] Module.End ℂ V) (μ : Ix p q → ℚ) : hwSub (Cover.twAct act) μ = hwSub act fun a => μ a + 1 / 2

theorem **sum_of_inCone** {lam μ : Ix p q → ℚ} (h : InCone lam μ) : ∑ a, μ a = ∑ a, lam a

theorem **inCone_add** {lam μ : Ix p q → ℚ} (c : ℚ) (h : InCone lam fun a => μ a - c) : InCone (fun a => lam a + c) μ

theorem **hasTypes_of_twist** (p q : ℕ) (T : GrpData p q) (R : DSRep T) (lam : Ix p q → ℚ) (h : HasTypes (Cover.twist R) lam) : HasTypes R fun a => lam a + 1 / 2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasDiscreteSeriesKTypesOfSqrtDetTwistStatement

theorem **_root_.GlobalTheory.T.hasDiscreteSeriesKTypes_of_sqrtDetTwist** : ∀ (p q : ℕ) (T : GrpData p q) (R : DSRep T) (lam : Ix p q → ℚ), HasTypes (Cover.twist R) lam → HasTypes R fun a => lam a + 1 / 2

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasFDerivAtOfContinuousCoordinatePartials

theorem **hasFDerivAt_of_continuous_coordinate_line_derivs** : ∀ (k : ℕ) (F : (Fin k → ℝ) → ℂ) (G : Fin k → (Fin k → ℝ) → ℂ), (∀ i, Continuous (G i)) → (∀ i t, HasDerivAt (fun s : ℝ => F (t + s • Pi.single i 1)) (G i t) 0) → ∀ t, HasFDerivAt F (∑ i, (ContinuousLinearMap.proj i : (Fin k → ℝ) →L[ℝ] ℝ).smulRight (G i t)) t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasFDerivAtOfContinuousCoordinatePartialsStatement

theorem **_root_.GlobalTheory.T.hasFDerivAt_of_continuous_coordinate_partials** : ∀ (k : ℕ) (F : (Fin k → ℝ) → ℂ) (G : Fin k → (Fin k → ℝ) → ℂ), (∀ i, Continuous (G i)) → (∀ i t, HasDerivAt (fun s : ℝ => F (t + s • Pi.single i 1)) (G i t) 0) → ∀ t, HasFDerivAt F (∑ i, (ContinuousLinearMap.proj i : (Fin k → ℝ) →L[ℝ] ℝ).smulRight (G i t)) t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasIwasawaDecompositionDoubledSplitPlace

theorem **shear_mulVec** (x y : m → A) : (shear e).1 *ᵥ pairVec e x y = pairVec e x (x + y)

theorem **bc_shear** {B : Type} [CommRing B] (f : A →+* B) : bc f (shear e) = shear e

theorem **pair_dblGram** (J : Matrix m m A) (x y x' y' : m → A) : pairVec e x y ⬝ᵥ dblGram e J *ᵥ pairVec e x' y' = x ⬝ᵥ J *ᵥ x' - y ⬝ᵥ J *ᵥ y'

theorem **keepsDiag_of_pairing** {J : Matrix m m A} (hJ : IsUnit J.det) {x : Matrix ι ι A} {y : GL ι A} (h : ∀ u' u, (x *ᵥ u') ⬝ᵥ dblGram e J *ᵥ (y.1 *ᵥ u) = u' ⬝ᵥ dblGram e J *ᵥ u) (hy : y ∈ stabGL (diagSub A e)) : KeepsDiag e x

theorem **iwasawa_diag** {V K : Type} [CommRing V] [IsDomain V] [ValuationRing V] [Field K] [Algebra V K] [IsFractionRing V K] {n : ℕ} (x : GL (Fin (n + n)) K) : ∃ k : GL (Fin (n + n)) V, x * bc (algebraMap V K) k ∈ stabGL (diagSub K (finSumFinEquiv (m

theorem **mem_siegelH_of_split** (h : IsCMExt F E σ) (hQ : Q = dblGram finSumFinEquiv J) (hJ : J.det ≠ 0) (τ : E →ₐ[F] Fv F v) {g : GLoc σ Q v} (hg : splitHom σ Q τ g ∈ stabGL (diagSub (Fv F v) finSumFinEquiv)) : g ∈ siegelH (Fv F v) σ Q

theorem **hasIwasawa_of_algHom** (h : IsCMExt F E σ) (hQ : Q = dblGram finSumFinEquiv J) (hJ : IsHermND σ J) (hQ' : IsHermND σ Q) (τ : E →ₐ[F] Fv F v) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (hL : IsSelfDualAt σ J L v) : HasIwasawa σ Q (KFin σ Q L v)

theorem **isSelfDualAt_of_isSelfDualC** {D : D7} {A : HSp D} {L : Lat A} (hL : IsSelfDualC A L) (v : FinPl D) : IsSelfDualAt D.sp.cm.sig A.gram L v

theorem **hasIwasawa_split** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ v : FinPl D, Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasIwasawaDecompositionDoubledSplitPlaceStatement

theorem **_root_.GlobalTheory.T.hasIwasawaDecomposition_doubled_split_place** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ v : FinPl D, Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasProdPartialLFunctionTwistedByCharLocalFactors

theorem **LSD_hasProd_LvTw** : ∀ (D : D7) (Sf : Finset (FinPl D)) (par : GlobalTheory.LFun.Fam D.host.E) (χ : (AE D)ˣ →* Circle) (r : ℕ) (κ : ℝ), GlobalTheory.LFun.Bounded (GlobalTheory.LFun.above D.host.E (Sf : Set (FinPl D))) par r κ → ∀ z : ℂ, κ + 1 < z.re → HasProd (fun

v : {v : FinPl D // v ∉ (Sf : Set (FinPl D))} => GlobalTheory.LFun.LvTw par (GlobalTheory.LFun.val χ) v.1 z) (GlobalTheory.LFun.LSD D Sf par χ z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasProdPartialLFunctionTwistedByCharLocalFactorsStatement

theorem **_root_.GlobalTheory.T.hasProd_partialLFunctionTwistedByChar_local_factors** : ∀ (D : D7) (Sf : Finset (FinPl D)) (par : GlobalTheory.LFun.Fam D.host.E) (χ : (AE D)ˣ →* Circle) (r : ℕ) (κ : ℝ), GlobalTheory.LFun.Bounded (GlobalTheory.LFun.above D.host.E (Sf : Set (FinPl D))) par r κ → ∀ z : ℂ, κ + 1 < z.re → HasProd (fun v : {v : FinPl D // v ∉ (Sf : Set (FinPl D))} => GlobalTheory.LFun.LvTw par (GlobalTheory.LFun.val χ) v.1 z) (GlobalTheory.LFun.LSD D Sf par χ z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasResOfValSubLt

theorem **fn_eq** {p : DecoF E n ST SG Q} {k : GL (Fin n) (AfE F E)} (h : IsPtOf hc k p) (hN : p.deco.1.N = Formalism.L2.Col.stdLat E n) (P : ST) (r : Fin n) (hr : r.1 + 1 = n) (c : 𝒪 E) (hrow : ∀ i, HasRes P.1 (adAt F P.1 ((k⁻¹ : GL (Fin n) (AfE F E)).1 r i)) (if i = r then Ideal.Quotient.mk P.1.asIdeal c else 0)) (x : Fin n → 𝒪 E) : p.deco.1.f P (fun i => (x i : E)) = Ideal.Quotient.mk P.1.asIdeal (c * x r)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumDoublingInert

theorem **hasSum_doubling_inert** : GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.UnramDoublingInert → ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : Lˣ) (σ : L →+* L) (q n : ℕ), IsUnramInv (ϖ : L) σ q → ∀ χ : Lˣ →* ℂˣ, GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsUnr χ → ∀ m : ℕ, ((χ ϖ : ℂˣ) : ℂ) = (-1) ^ m → ∀ (s : ℂ) (Φ : GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.GD σ n → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsSphSec σ q n χ s Φ → ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ (GU σ n) V) (x : V) (α : Fin (n / 2) → ℂˣ), IsSatVecU (ϖ : L) q ρ x α → ∀ ℓ : V →ₗ[ℂ] ℂ, (∀ k ∈ KU σ n, ∀ g, ℓ (ρ (k * g) x) = ℓ (ρ g x)) → (ℓ x ≠ 0 → GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.Domain (GlobalTheory.IndefiniteRowSideFullHe

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumHeckeSeriesBorelSphericalIntegralCosets

theorem **ord_det_of_mem_KO** {y : GL (Fin n) k} (hy : y ∈ KO k (Fin n)) : ord k (Matrix.GeneralLinearGroup.det y) = 0

private theorem **hermGL_coset_out_entries_integral** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : ℕ} {p : 𝒪[k]} (hp : Irreducible p) (a : Fin n → ℕ) (c : _root_.GlobalTheory.SatakeBound.HermIdx p a) : ∀ i j, (((_root_.GlobalTheory.SatakeBound.hermGL hp a c : GL (Fin n) k) ⧸ _root_.GlobalTheory.SplitWeil.KO k (Fin n)).out : GL (Fin n) k) : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]

private theorem **exists_hermGL_coset_eq_of_integral** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : ℕ} {p : 𝒪[k]} (hp : Irreducible p) (g : GL (Fin n) k) (hg : ∀ i j, (g : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]) : ∃ a c, ((_root_.GlobalTheory.SatakeBound.hermGL hp a c : GL (Fin n) k) : GL (Fin n) k ⧸ _root_.GlobalTheory.SplitWeil.KO k (Fin n)) = (g : GL (Fin n) k ⧸ _)

private theorem **ord_eq_of_nabs_eq_resCard_zpow_neg** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {x : kˣ} {m : ℤ} (hx : _root_.GlobalTheory.SplitWeil.nabs k x = (_root_.GlobalTheory.SplitWeil.resCard k : ℝ) ^ (-m)) : _root_.GlobalTheory.SmGL.ord k x = m

private theorem **ord_det_hermGL_mul_eq_sum** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : ℕ} {p : 𝒪[k]} (hp : Irreducible p) (a : Fin n → ℕ) (c : _root_.GlobalTheory.SatakeBound.HermIdx p a) {w : GL (Fin n) k} (hw : w ∈ _root_.GlobalTheory.SplitWeil.KO k (Fin n)) : _root_.GlobalTheory.SmGL.ord k (Matrix.GeneralLinearGroup.det (_root_.GlobalTheory.SatakeBound.hermGL hp a c * w)) = ∑ i, (a i : ℤ)

private theorem **hasSum_prod_pow_eq_prod_one_sub_inv** : ∀ (n : ℕ) (β : Fin n → ℂ), (∀ i, ‖β i‖ < 1) → Summable (fun a : Fin n → ℕ => ‖∏ i, β i ^ a i‖) ∧ HasSum (fun a : Fin n → ℕ => ∏ i, β i ^ a i) (∏ i, (1 - β i)⁻¹) | 0, β, _ => by refine (Summable.of_finite, ?_) simpa using (hasSum_fintype (fun a : Fin 0 → ℕ => (1 : ℂ))) | n + 1, β, hβ => by obtain ⟨hs, hh⟩

theorem **_root_.GlobalTheory.HeckeSeries.hasSum_integral_cosets** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), _root_.GlobalTheory.SatakeBound.IsBorelSph k γ φ → ∀ X : ℂ, (∀ i : Fin n, ‖(_root_.GlobalTheory.SplitWeil.resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X‖ < 1) → HasSum (fun c : {c : GL (Fin n) k ⧸ _root_.GlobalTheory.SplitWeil.KO k (Fin n) // (∀ i j, (c.out : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ (_root_.GlobalTheory.SmGL.ord k (Matrix.GeneralLinearGroup.det c.1.out)).toNat * φ c.1.out) (∏ i : Fin n, (1 - (_root_.GlobalTheory.SplitWeil.resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumHeckeSeriesBorelSphericalIntegralCosetsStatement

theorem **_root_.GlobalTheory.T.hasSum_heckeSeries_borelSpherical_integralCosets** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), IsBorelSph k γ φ → ∀ X : ℂ, (∀ i : Fin n, ‖(resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X‖ < 1) → HasSum (fun c : {c : GL (Fin n) k ⧸ KO k (Fin n) // (∀ i j, (c.out : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ (ord k (Matrix.GeneralLinearGroup.det c.1.out)).toNat * φ c.1.out) (∏ i : Fin n, (1 - (resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumHeckeSeriesSphericalEigenfunction

theorem **int_mul** {A B : Matrix (Fin n) (Fin n) k} (hA : ∀ i j, A i j ∈ 𝒪[k]) (hB : ∀ i j, B i j ∈ 𝒪[k]) (i j : Fin n) : (A * B) i j ∈ 𝒪[k]

theorem **ord_det_of_mem_KO** {κ : GL (Fin n) k} (hκ : κ ∈ KO k (Fin n)) : ord k (Matrix.GeneralLinearGroup.det κ) = 0

theorem **out_smul** (y : GL (Fin n) k) (c : GL (Fin n) k ⧸ KO k (Fin n)) : ∃ κ ∈ KO k (Fin n), ((y • c).out : GL (Fin n) k) = y * c.out * κ

theorem **int_out_smul** {y : GL (Fin n) k} (hy : y ∈ KO k (Fin n)) (c : IntCos k n) (i j : Fin n) : ((y • c.1).out : GL (Fin n) k).1 i j ∈ 𝒪[k]

theorem **diag_mul_upper** {A B : Matrix (Fin n) (Fin n) k} (hA : ∀ i j, j < i → A i j = 0) (hB : ∀ i j, j < i → B i j = 0) (i : Fin n) : (A * B) i i = A i i * B i i

theorem **diag_ne_zero** {b : GL (Fin n) k} (hb : b ∈ borel k (Fin n)) (i : Fin n) : (b : Matrix (Fin n) (Fin n) k) i i ≠ 0

theorem **exists_borel_mul** (h : GL (Fin n) k) : ∃ b ∈ borel k (Fin n), ∃ x ∈ KO k (Fin n), h = b * x

theorem **sph_borel_mul** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : IsBorelSph k γ φ) (hγ : ∀ i, γ i ≠ 0) {b : GL (Fin n) k} (hb : b ∈ borel k (Fin n)) (h : GL (Fin n) k) : φ (b * h) = (∏ i, γ i ^ ord k (diagU hb i)) * φ h

theorem **hasSum_hecke_shift** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), IsBorelSph k γ φ → (∀ i, γ i ≠ 0) → ∀ X : ℂ, (∀ i : Fin n, ‖(resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X‖ < 1) → ∀ g : GL (Fin n) k, HasSum (fun c : {c : GL (Fin n) k ⧸ KO k (Fin n) // (∀ i j, (c.out : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ (ord k (Matrix.GeneralLinearGroup.det c.1.out)).toNat * φ (g * c.1.out)) ((∏ i : Fin n, (1 - (resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹) * φ g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumHeckeSeriesSphericalEigenfunctionStatement

theorem **_root_.GlobalTheory.T.hasSum_hecke_series_spherical_eigenfunction** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), IsBorelSph k γ φ → (∀ i, γ i ≠ 0) → ∀ X : ℂ, (∀ i : Fin n, ‖(resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X‖ < 1) → ∀ g : GL (Fin n) k, HasSum (fun c : {c : GL (Fin n) k ⧸ KO k (Fin n) // (∀ i j, (c.out : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ (ord k (Matrix.GeneralLinearGroup.det c.1.out)).toNat * φ (g * c.1.out)) ((∏ i : Fin n, (1 - (resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹) * φ g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumHeckeSeriesSphericalIntegralInverse

private lemma **isBorelSph_comp_invRevT** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : _root_.GlobalTheory.SatakeBound.IsBorelSph k γ φ) : _root_.GlobalTheory.SatakeBound.IsBorelSph k (fun i => (γ i.rev)⁻¹) (fun g => φ (_root_.GlobalTheory.SatakeFlip.invRevT g))

private lemma **exists_cosetMap_out_eq_invRevT_mul** (c : GL (Fin n) k ⧸ _root_.GlobalTheory.SplitWeil.KO k (Fin n)) : ∃ x ∈ _root_.GlobalTheory.SplitWeil.KO k (Fin n), (_root_.GlobalTheory.SatakeFlip.cosetMap c).out = _root_.GlobalTheory.SatakeFlip.invRevT c.out * x

private lemma **mul_entries_integral** {g h : GL (Fin n) k} (hg : ∀ i j, (g : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]) (hh : ∀ i j, (h : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]) : ∀ i j, ((g * h : GL (Fin n) k) : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]

private lemma **invRevT_entries_integral_iff_inv** (g : GL (Fin n) k) : (∀ i j, ((_root_.GlobalTheory.SatakeFlip.invRevT g : GL (Fin n) k) : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]) ↔ ∀ i j, ((g⁻¹ : GL (Fin n) k) : Matrix (Fin n) (Fin n) k) i j ∈ 𝒪[k]

private lemma **det_invRevT'** (g : GL (Fin n) k) : Matrix.GeneralLinearGroup.det (_root_.GlobalTheory.SatakeFlip.invRevT g) = (Matrix.GeneralLinearGroup.det g)⁻¹

private lemma **ord_det_eq_zero_of_mem_KO** {x : GL (Fin n) k} (hx : x ∈ _root_.GlobalTheory.SplitWeil.KO k (Fin n)) : _root_.GlobalTheory.SmGL.ord k (Matrix.GeneralLinearGroup.det x) = 0

theorem **_root_.GlobalTheory.HeckeDual.hasSum_dual** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), _root_.GlobalTheory.SatakeBound.IsBorelSph k γ φ → (∀ i, γ i ≠ 0) → ∀ X : ℂ, (∀ i : Fin n, ‖(_root_.GlobalTheory.SplitWeil.resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X‖ < 1) → HasSum (fun c : {c : GL (Fin n) k ⧸ _root_.GlobalTheory.SplitWeil.KO k (Fin n) // (∀ i j, (c.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ (- _root_.GlobalTheory.SmGL.ord k (Matrix.GeneralLinearGroup.det c.1.out)).toNat * φ c.1.out) (∏ i : Fin n, (1 - (_root_.GlobalTheory.SplitWeil.resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X)⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumHeckeSeriesSphericalIntegralInverseStatement

theorem **_root_.GlobalTheory.T.hasSum_hecke_series_spherical_integral_inverse** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), IsBorelSph k γ φ → (∀ i, γ i ≠ 0) → ∀ X : ℂ, (∀ i : Fin n, ‖(resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X‖ < 1) → HasSum (fun c : {c : GL (Fin n) k ⧸ KO k (Fin n) // (∀ i j, (c.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ (-ord k (Matrix.GeneralLinearGroup.det c.1.out)).toNat * φ c.1.out) (∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X)⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumSphericalFunctionDoublingLength

theorem **int_mul_of_mem_KO** {g x : GL (Fin n) k} (hg : ∀ i j, g.1 i j ∈ 𝒪[k]) (hx : x ∈ KO k (Fin n)) : ∀ i j, (g * x).1 i j ∈ 𝒪[k]

theorem **ord_det_of_mem_KO** {x : GL (Fin n) k} (hx : x ∈ KO k (Fin n)) : ord k (Matrix.GeneralLinearGroup.det x) = 0

theorem **pairEquiv_apply** (p : (GL (Fin n) k ⧸ KO k (Fin n)) × (GL (Fin n) k ⧸ KO k (Fin n))) : pairEquiv k n p = (p.2, (QuotientGroup.mk (p.2.out⁻¹ * p.1.out) : GL (Fin n) k ⧸ KO k (Fin n)))

theorem **hasSum_of_subtype** {β : Type} {P : β → Prop} (f : β → ℂ) {g : β → ℂ} {a b : ℂ} (c : ℂ) (h : HasSum (fun x : {x // P x} => f x.1) a) (h1 : ∀ x, P x → g x = f x * c) (h0 : ∀ x, ¬ P x → g x = 0) (hb : b = a * c) : HasSum g b

theorem **hasSum_pairTerm_fiber** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : IsBorelSph k γ φ) (hγ : ∀ i, γ i ≠ 0) {X : ℂ} (hX : ∀ i : Fin n, ‖(resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X‖ < 1) (h : GL (Fin n) k ⧸ KO k (Fin n)) : HasSum (fun c => pairTerm X φ (h, c)) (if (∀ i j, (h.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k]) then X ^ (-ord k (Matrix.GeneralLinearGroup.det h.out)).toNat * φ h.out * ∏ i : Fin n, (1 - (resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹ else 0)

theorem **hasSum_pairTerm_total** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : IsBorelSph k γ φ) (hγ : ∀ i, γ i ≠ 0) {X : ℂ} (hX : ∀ i : Fin n, ‖(resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X‖ < 1) (a : ℂ) : HasSum (fun h : GL (Fin n) k ⧸ KO k (Fin n) => if (∀ i j, (h.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k]) then X ^ (-ord k (Matrix.GeneralLinearGroup.det h.out)).toNat * φ h.out * a else 0) (a * ∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X)⁻¹)

theorem **norm_pairTerm** (X : ℂ) (φ : GL (Fin n) k → ℂ) (p : (GL (Fin n) k ⧸ KO k (Fin n)) × (GL (Fin n) k ⧸ KO k (Fin n))) : (‖pairTerm X φ p‖ : ℝ) : ℂ) = pairTerm (‖X‖ : ℂ) (fun g => (‖φ g‖ : ℂ)) p

theorem **isBorelSph_norm** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (h : IsBorelSph k γ φ) : IsBorelSph k (fun i => (‖γ i‖ : ℂ)) (fun g => (‖φ g‖ : ℂ))

theorem **summable_pairTerm** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : IsBorelSph k γ φ) (hγ : ∀ i, γ i ≠ 0) {X : ℂ} (hX1 : ∀ i : Fin n, ‖(resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X‖ < 1) (hX2 : ∀ i : Fin n, ‖(resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X‖ < 1) : Summable (pairTerm X φ)

theorem **pairTerm_pairEquiv** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : IsBorelSph k γ φ) (X : ℂ) (g h : GL (Fin n) k ⧸ KO k (Fin n)) : pairTerm X φ (pairEquiv k n (g, h)) = if (∀ i j, (h.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k]) ∧ (∀ i j, (h.out⁻¹ * g.out : GL (Fin n) k).1 i j ∈ 𝒪[k]) then X ^ ((-ord k (Matrix.GeneralLinearGroup.det h.out)).toNat + (ord k (Matrix.GeneralLinearGroup.det (h.out⁻¹ * g.out))).toNat) * φ g.out else 0

theorem **hasSum_pairTerm_pairEquiv** {γ : Fin n → ℂ} {φ : GL (Fin n) k → ℂ} (hφ : IsBorelSph k γ φ) {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) {len : GL (Fin n) k → ℕ} (hlen : ∀ e : Fin n → ℤ, Antitone e → ∀ g ∈ cell k (Fin n) ϖ e, len g = (∑ i, |e i|).toNat) {X : ℂ} (hX : ∀ i : Fin n, ‖(resCard k : ℂ) ^ (i : ℕ) * X ^ 2‖ < 1) (g : GL (Fin n) k ⧸ KO k (Fin n)) : HasSum (fun h => pairTerm X φ (pairEquiv k n (g, h))) (X ^ len g.out * (∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * X ^ 2)⁻¹) * φ g.out)

theorem **exists_small** (q : ℝ) (hq0 : 0 ≤ q) (γ : Fin n → ℂ) : ∃ δ : ℝ, 0 < δ ∧ ∀ x : ℝ, 0 ≤ x → x < δ → ∀ i : Fin n, q ^ (n - 1 - (i : ℕ)) * ‖γ i‖ * x < 1 ∧ q ^ (i : ℕ) * ‖γ i‖⁻¹ * x < 1 ∧ q ^ (i : ℕ) * x ^ 2 < 1

theorem **hasSum_doublingLength** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), IsBorelSph k γ φ → (∀ i, γ i ≠ 0) → ∀ ϖ : kˣ, valuation k (ϖ : k) = uniformizer k → ∀ len : GL (Fin n) k → ℕ, (∀ e : Fin n → ℤ, Antitone e → ∀ g ∈ cell k (Fin n) ϖ e, len g = (∑ i, |e i|).toNat) → ∃ δ : ℝ, 0 < δ ∧ ∀ X : ℂ, ‖X‖ < δ → HasSum (fun c : GL (Fin n) k ⧸ KO k (Fin n) => X ^ len c.out * φ c.out) ((∏ i : Fin n, (1 - (resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹) * (∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X)⁻¹) * ∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * X ^ 2))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HasSumSphericalFunctionDoublingLengthStatement

theorem **_root_.GlobalTheory.T.hasSum_sphericalFunction_doublingLength** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (γ : Fin n → ℂ) (φ : GL (Fin n) k → ℂ), IsBorelSph k γ φ → (∀ i, γ i ≠ 0) → ∀ ϖ : kˣ, valuation k (ϖ : k) = uniformizer k → ∀ len : GL (Fin n) k → ℕ, (∀ e : Fin n → ℤ, Antitone e → ∀ g ∈ cell k (Fin n) ϖ e, len g = (∑ i, |e i|).toNat) → ∃ δ : ℝ, 0 < δ ∧ ∀ X : ℂ, ‖X‖ < δ → HasSum (fun c : GL (Fin n) k ⧸ KO k (Fin n) => X ^ len c.out * φ c.out) ((∏ i : Fin n, (1 - (resCard k : ℂ) ^ (n - 1 - (i : ℕ)) * γ i * X)⁻¹) * (∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * (γ i)⁻¹ * X)⁻¹) * ∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * X ^ 2))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HdNumbers

theorem **hd_numbers** : ∀ (D : D7) (u : D.what₂.Word) (L C' Cl Cg cb Cu cT CT : ℝ), HD D L C' Cl Cg cb Cu cT CT → 5 ≤ mfL D L ∧ 4 * D.R.f + 1 ≤ (D.mem u).A.T ∧ 0 < d0 D u ∧ 5 ≤ (D.R.f : ℝ) * D.R.k * L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharQuotientNeOneCompletelySplitPrime

theorem **val_quot_ne_one** : ∀ (D : D7) (c : Com D), c.Is → ∀ i j : ℕ, 1 ≤ i → i < j → j ≤ D.R.m → ∀ p : ℕ, GlobalTheory.IndefiniteRowSideFullHecke.IteratedLiftChains.SplitsCompletely D p → ∀ w : HeightOneSpectrum (𝒪 D.host.E), (p : 𝒪 D.host.E) ∈ w.asIdeal → w.under (𝒪 D.host.F) ∉ c.S → GlobalTheory.LFun.val (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharQuotientNeOneCompletelySplitPrimeStatement

theorem **_root_.GlobalTheory.T.heckeCharQuotient_ne_one_completelySplitPrime** : ∀ (D : D7) (c : Com D), c.Is → ∀ i j : ℕ, 1 ≤ i → i < j → j ≤ D.R.m → ∀ p : ℕ, GlobalTheory.IndefiniteRowSideFullHecke.IteratedLiftChains.SplitsCompletely D p → ∀ w : HeightOneSpectrum (𝒪 D.host.E), (p : 𝒪 D.host.E) ∈ w.asIdeal → w.under (𝒪 D.host.F) ∉ c.S → GlobalTheory.LFun.val (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharQuotientUnramifiedAtTwistPlace

theorem **loc_eq_of_isRes** {D : D7} {H H' : Formalism.L2.Host.HChar D.host.E} {PP : Pl D.host.E} {a : (kP PP)ˣ →* ℂˣ} (h : IsRes H PP a) (h' : IsRes H' PP a) (x : (PP.adicCompletion D.host.E)ˣ) (hx : Formalism.L2.Host.IsUnitAt PP x) : H.loc PP x = H'.loc PP x

theorem **GlobalTheory.ChainTw.quot_unrAt_twist** : ∀ (D : D7) (c : Com D) (w : Twist D), w.Is c → ∀ PP ∈ w.Q, ∀ i j : ℕ, 1 ≤ i → i ≤ D.R.m → 1 ≤ j → j ≤ D.R.m → GlobalTheory.LFun.UnrAt (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharQuotientUnramifiedAtTwistPlaceStatement

theorem **_root_.GlobalTheory.T.hecke_char_quotient_unramified_at_twist_place** : ∀ (D : D7) (c : Com D) (w : Twist D), w.Is c → ∀ PP ∈ w.Q, ∀ i j : ℕ, 1 ≤ i → i ≤ D.R.m → 1 ≤ j → j ≤ D.R.m → GlobalTheory.LFun.UnrAt (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharTameAtOfFactorsThroughResidue

theorem **isOpen_ltOne** (P : HeightOneSpectrum (𝒪 E)) : IsOpen {z : P.adicCompletion E | Valued.v z < 1}

theorem **val_alg** (P : HeightOneSpectrum (𝒪 E)) (k : E) : Valued.v (algebraMap E (P.adicCompletion E) k) = P.valuation E k

theorem **valuation_eq_one_of_not_mem** (P : HeightOneSpectrum (𝒪 E)) (s : 𝒪 E) (hs : s ∉ P.asIdeal) : P.valuation E (algebraMap (𝒪 E) E s) = 1

theorem **locK_eq_one** (D : D7) (H : Formalism.L2.Host.HChar D.host.E) (PP : Pl D.host.E) (c : (kP PP)ˣ →* ℂˣ) (hc : IsRes H PP c) (k : D.host.Eˣ) (hk : PP.valuation D.host.E ((k : D.host.E) - 1) < 1) : H.locK PP k = 1

theorem **tameAt_of_isRes** : ∀ (D : D7) (H : Formalism.L2.Host.HChar D.host.E) (PP : Pl D.host.E) (a : (kP PP)ˣ →* ℂˣ), IsRes H PP a → H.TameAt PP

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharTameAtOfFactorsThroughResidueStatement

theorem **_root_.GlobalTheory.T.heckeChar_tameAt_of_factors_through_residue** : ∀ (D : D7) (H : Formalism.L2.Host.HChar D.host.E) (PP : Pl D.host.E) (a : (kP PP)ˣ →* ℂˣ), IsRes H PP a → H.TameAt PP

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharUnramifiedAtQuotientSameResidual

theorem **exists_int_div** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) (k : E) (hk : P.valuation E k = 1) : ∃ a s : 𝒪 E, a ∉ P.asIdeal ∧ s ∉ P.asIdeal ∧ k * algebraMap (𝒪 E) E s = algebraMap (𝒪 E) E a

theorem **loc_eq_of_isRes** {D : D7} {H₁ H₂ : Formalism.L2.Host.HChar D.host.E} {PP : Pl D.host.E} {ξ : (kP PP)ˣ →* ℂˣ} (h₁ : IsRes H₁ PP ξ) (h₂ : IsRes H₂ PP ξ) (x : (PP.adicCompletion D.host.E)ˣ) (hx : IsUnitAt PP x) : H₁.loc PP x = H₂.loc PP x

theorem **unrAt_div_of_isRes** : ∀ (D : D7) (c₁ c₂ : (AE D)ˣ →* Circle) (H₁ H₂ : Formalism.L2.Host.HChar D.host.E) (PP : Pl D.host.E) (ξ : (kP PP)ˣ →* ℂˣ), ChrEq c₁ H₁ → ChrEq c₂ H₂ → IsRes H₁ PP ξ → IsRes H₂ PP ξ → GlobalTheory.LFun.UnrAt (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharUnramifiedAtQuotientSameResidualStatement

theorem **_root_.GlobalTheory.T.heckeCharUnramifiedAt_quotient_same_residual** : ∀ (D : D7) (c₁ c₂ : (AE D)ˣ →* Circle) (H₁ H₂ : Formalism.L2.Host.HChar D.host.E) (PP : Pl D.host.E) (ξ : (kP PP)ˣ →* ℂˣ), ChrEq c₁ H₁ → ChrEq c₂ H₂ → IsRes H₁ PP ξ → IsRes H₂ PP ξ → GlobalTheory.LFun.UnrAt (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharValueInertPlaceAndSegment

theorem **npow_eq_zpow** {K : Type} [Field K] [NumberField K] (w : HeightOneSpectrum (𝒪 K)) (N : ℕ) (hN : Ideal.absNorm w.asIdeal = N ^ 2) (k : ℤ) (z : ℂ) (hz : 2 * z = k) : GlobalTheory.LFun.npow w z = (N : ℂ) ^ k

theorem **range_reflect** (t : ℕ) : (Multiset.range t).map (fun i => t - 1 - i) = Multiset.range t

theorem **GlobalTheory.ChainOnto.val_seg_of_isInertAt** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle) (n : ℕ), IsHeckeChar D χ → Formalism.L2.Theta.ResEta D χ n → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)), w.under (𝒪 D.host.F) = v → IsInertAt D.host.F D.host.E v w → UnrAt χ w → val (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharValueInertPlaceAndSegmentStatement

theorem **_root_.GlobalTheory.T.heckeChar_value_inert_place_and_segment** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle) (n : ℕ), IsHeckeChar D χ → Formalism.L2.Theta.ResEta D χ n → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)), w.under (𝒪 D.host.F) = v → IsInertAt D.host.F D.host.E v w → UnrAt χ w → val (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharValueSplitPrimeNotRootOfUnity

theorem **not_isOfFinOrder_val** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle), IsHeckeChar D χ → (∀ τ : D.what₂.Tau, ∃ a : ℚ, a ≠ 0 ∧ HasInfExp D χ τ a) → ∀ p : ℕ, GlobalTheory.IndefiniteRowSideFullHecke.IteratedLiftChains.SplitsCompletely D p → ∀ w : HeightOneSpectrum (𝒪 D.host.E), (p : 𝒪 D.host.E) ∈ w.asIdeal → GlobalTheory.LFun.UnrAt χ w → ¬ IsOfFinOrder (GlobalTheory.LFun.val (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharValueSplitPrimeNotRootOfUnityStatement

theorem **_root_.GlobalTheory.T.heckeChar_value_split_prime_not_root_of_unity** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle), IsHeckeChar D χ → (∀ τ : D.what₂.Tau, ∃ a : ℚ, a ≠ 0 ∧ HasInfExp D χ τ a) → ∀ p : ℕ, GlobalTheory.IndefiniteRowSideFullHecke.IteratedLiftChains.SplitsCompletely D p → ∀ w : HeightOneSpectrum (𝒪 D.host.E), (p : 𝒪 D.host.E) ∈ w.asIdeal → GlobalTheory.LFun.UnrAt χ w → ¬ IsOfFinOrder (GlobalTheory.LFun.val (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharacterEqAtDetRootsOfCongruent

theorem **det_sub_one_le** {K Γ : Type} [Field K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) {n : ℕ} (M : Matrix (Fin n) (Fin n) K) (ε : Γ) (hε : ε ≤ 1) (h : ∀ r s, v (M r s - (1 : Matrix (Fin n) (Fin n) K) r s) ≤ ε) : v (M.det - 1) ≤ ε

theorem **exists_depth** (E : Type) [Field E] [NumberField E] (K : ℕ) (hK : 0 < K) : ∃ N₀ : ℕ, ∀ P : HeightOneSpectrum (𝒪 E), WithZero.exp (-(N₀ : ℤ)) < Valued.v ((K : P.adicCompletion E))

theorem **fin_mem_iff** (D : D7) (a : AE D) : finPart D a ∈ intAE D ↔ ∀ P : HeightOneSpectrum (𝒪 D.host.E), Valued.v (adAt D.host.F P (toEAf D a)) ≤ 1

theorem **isUnit_of_parts** (D : D7) (a : AE D) (h1 : infPart D a = 1) (h2 : ∀ P : HeightOneSpectrum (𝒪 D.host.E), Valued.v (adAt D.host.F P (toEAf D a)) = 1) : IsUnit a

theorem **natCast_le_one** {K Γ : Type} [Field K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) (n : ℕ) : v (n : K) ≤ 1

theorem **exists_root_same_value** : ∀ (D : D7) (c : (AE D)ˣ →* Circle), IsHeckeChar D c → ∃ N₀ : ℕ, ∀ (V : HSp D) {n : ℕ} (hn : V.rank = n) (g w : GA D V), w ∈ GfC V → (∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (r s : Fin n), Valued.v (adAt D.host.F P (finM V hn w r s - (1 : Matrix (Fin n) (Fin n) (EF D)) r s)) ≤ WithZero.exp (-(N₀ : ℤ))) → ∀ x, IsDetRoot V g x → ∃ x₁, IsDetRoot V (g * w) x₁ ∧ c x₁ = c x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharacterEqAtDetRootsOfCongruentStatement

theorem **_root_.GlobalTheory.T.heckeCharacter_eq_at_det_roots_of_congruent** : ∀ (D : D7) (c : (AE D)ˣ →* Circle), IsHeckeChar D c → ∃ N₀ : ℕ, ∀ (V : HSp D) {n : ℕ} (hn : V.rank = n) (g w : GA D V), w ∈ GfC V → (∀ (P : HeightOneSpectrum (𝒪 D.host.E)) (r s : Fin n), Valued.v (adAt D.host.F P (finM V hn w r s - (1 : Matrix (Fin n) (Fin n) (EF D)) r s)) ≤ WithZero.exp (-(N₀ : ℤ))) → ∀ x, IsDetRoot V g x → ∃ x₁, IsDetRoot V (g * w) x₁ ∧ c x₁ = c x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharacterEqOneOfTrivialSplitPlaces

theorem **heckeChar_eq_one_of_split** : ∀ (D : D7) (ν : (AE D)ˣ →* Circle), IsHeckeChar D ν → (∀ e, IsSplitIdem D e → ∀ x : (AE D)ˣ, (e + cA D e) * ((x : AE D) - 1) = (x : AE D) - 1 → ν x = 1) → ∀ x, ν x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCharacterEqOneOfTrivialSplitPlacesStatement

theorem **_root_.GlobalTheory.T.heckeCharacter_eq_one_of_trivial_splitPlaces** : ∀ (D : D7) (ν : (AE D)ˣ →* Circle), IsHeckeChar D ν → (∀ e, IsSplitIdem D e → ∀ x : (AE D)ˣ, (e + cA D e) * ((x : AE D) - 1) = (x : AE D) - 1 → ν x = 1) → ∀ x, ν x = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCombinationScalarFormKeepsAdelicLaws

theorem **trc_apply'** {D : D7} {V : HSp D} (c : GA D V →₀ ℂ) (f : GA D V → ℂ) (x : GA D V) : trc c f x = ∑ g ∈ c.support, c g * f (x * g)

theorem **list_prod_right** {G : Type} [Group G] (F : G → ℂ) (L : List G) (hL : ∀ a ∈ L, ∀ z, F (z * a) = F z) (z : G) : F (z * L.prod) = F z

theorem **list_prod_chi** {G : Type} [Group G] (U : Subgroup G) (χ : U →* ℂˣ) (L : List G) (hL : ∀ a ∈ L, ∃ h : a ∈ U, χ (a, h) = 1) : ∃ h : L.prod ∈ U, χ (_, h) = 1

theorem **rd_laws** {B : Base} {X : PreX B.toPre} (I : TC B X) (hok : I.Ok) (hpins : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch) (hA : A.Is) (θ : InfFam (I.rec7 l Z)) (hθ : IsInfFam θ) (j : ℕ) (hj : j ≤ B.R.m) (φ : ColT I l u a j) (hφ : φ ∈ (I.fm l u a).Fi j) (b : Ch j) (y : (I.chain l u a).Y j) : (∀ γ ∈ GQ ((I.rec7 l Z).sp.X j), ∀ g, A.rd θ j hj b φ y (γ * g) = A.rd θ j hj b φ y g) ∧ (∀ κ : A.U j b, ∀ g, A.rd θ j hj b φ y (g * κ.1) = (A.chi j b κ : ℂ) * A.rd θ j hj b φ y g) ∧ (∀ k ∈ GinfC ((I.rec7 l Z).sp.X j), ∀ y' : (I.chain l u a).Y j, IsTr (G

theorem **GlobalTheory.AdForm.trc_rd_laws** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (c : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X j) →₀ ℂ), (∀ g ∈ c.support, g ∈ grpAt _ fun x => ∃ i, x = (p i).v) → ∀ φ ∈ (I.fm l u a).Fi j, (∀ b y i g, g • Q j i = Q j i → rTrans ((β j i).gl g) (trc c (A.rd θ j hj b φ y)) = trc c (A.rd θ j hj b φ y)) → ∀ b y, ∀ γ ∈ GQ ((I.rec7 l Z).sp.X j), ∀ (e : {e // A.chOf

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCombinationScalarFormKeepsAdelicLawsStatement

theorem **_root_.GlobalTheory.T.heckeCombination_scalarForm_keeps_adelicLaws** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (c : GA (I.rec7 l Z) ((I.rec7 l

Z).sp.X j) →₀ ℂ), (∀ g ∈ c.support, g ∈ grpAt _ fun x => ∃ i, x = (p i).v) → ∀ φ ∈ (I.fm l u a).Fi j, (∀ b y i g, g • Q j i = Q j i → rTrans ((β j i).gl g) (trc c (A.rd θ j hj b φ y)) = trc c (A.rd θ j hj b φ y)) → ∀ b y, ∀ γ ∈ GQ ((I.rec

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCombinationScalarFormValuesMemWeylModule

theorem **trc_val_mem** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (c : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X j) →₀ ℂ), (∀ g ∈ c.support, g ∈ grpAt _ fun x => ∃ i, x = (p i).v) → ∀ φ ∈ (I.fm l u a).Fi j, (∀ b y i g, g • Q j i = Q j i → rTrans ((β j i).gl g) (trc c (A.rd θ j hj b φ y)) = trc c (A.rd θ j hj b φ y)) → ∀ e : (I.stp u a j).Cl, (fun (e : (I.stp u a j).Cl) (z : (I.chain l u a).Y j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCombinationScalarFormValuesMemWeylModuleStatement

theorem **_root_.GlobalTheory.T.heckeCombination_scalarForm_values_mem_weylModule** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (c : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X j) →₀ ℂ), (∀ g ∈ c.support, g ∈ grpAt _ fun x => ∃ i, x = (p i).v) → ∀ φ ∈ (I.fm l u a).Fi j, (∀ b y i g, g • Q j i = Q j i → rTrans ((β j i).gl g) (trc c (A.rd θ j hj b φ y)) = trc c (A.rd θ j hj b φ y)) → ∀ e : (I.stp u a j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCombinationScalarFormValuesStabiliserInvariStatement

theorem **_root_.GlobalTheory.T.heckeCombination_scalarForm_values_stabiliser_invariant** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (c : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X j) →₀ ℂ), (∀ g ∈ c.support, g ∈ grpAt _ fun x => ∃ i, x = (p i).v) → ∀ φ ∈ (I.fm l u a).Fi j, (∀ b y i g, g • Q j i = Q j i → rTrans ((β j i).gl g) (trc c (A.rd θ j hj b φ y)) = trc c (A.rd θ j hj b φ y)) → ∀ (e : (I.st

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeCombinationScalarFormValuesStabilizerInvariant

theorem **chi_stab** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (i : ℕ) (hi : i ≤ B.R.m) {bad : Set (HeightOneSpectrum (𝒪 B.host.F))} {Ch : Type} (chOf : (I.stp u a i).Cl → Ch) (K : ∀ β : Ch, LevelChart (I.rec7 l Z) (B.T u) bad i hi (fun c : {c // chOf c = β} => (I.chain l u a).stab i c.1) (fun c => latCrd (I.rec7 l Z) i hi (I.latAt u a i c.1))) (k : Ch → GA B (B.R.N i)) (hR : I.Reads u a i chOf (fun β => (K β).lev.Lb) (fun β => (K β).lev.U) (fun β => (K β).lev.chi) (fun c => ((K (chOf c)).y ⟨c, rfl⟩).1) (fun c s => ((K (chOf c)).stabI ⟨c, rfl⟩ s).1) k) (c : (I.stp u a i).Cl) (s : (I.stp u a i).Stab c) (κ : (K (chOf c)).lev.U) (hy : (fun g y x : GA B (B.R.N i) => y = g * y * x) (bc (inclE (F

theorem **finM_mul'** {D : Formalism.BasicNotions.D7} (V : Formalism.BasicNotions.TH.HSp D) {n : ℕ} (hn : V.rank = n) (g h : GA D V) : finM V hn (g * h) = finM V hn g * finM V hn h

theorem **rTrans_mul** {G : Type} [Monoid G] (a b : G) (F : G → ℂ) : rTrans (a * b) F = rTrans a (rTrans b F)

theorem **rTrans_one** {G : Type} [Monoid G] (F : G → ℂ) : rTrans (1 : G) F = F

theorem **rTrans_list_prod** {G α : Type} [Monoid G] (f : α → G) (F : G → ℂ) (h : ∀ i, rTrans (f i) F = F) : ∀ L : List α, rTrans (L.map f).prod F = F | [] => by simpa using rTrans_one F | i :: L => by rw [List.map_cons, List.prod_cons, rTrans_mul, rTrans_list_prod f F h L, h i] end GlobalTheory.AdForm.StabTrans set_option maxHeartbeats 4000000 in set_option synthInstance.maxHeartbeats 400000 in theorem GlobalTheory.AdForm.trans_stab : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeFormOfDoubling

theorem **localZetaOn_all_of_split_inert** (D : D7) (hs : LocalZetaOn D fun v => Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **paramBoundOn_all_of_split_inert** (D : D7) (hs : ParamBoundOn D fun v => Formalism.MainNotions.SchurIrreducible.IsSplit (E

theorem **heckeForm_of_doubling₂** : GlobalTheory.Published.WeilRepresentationUnitaryPair → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.WeilSiegelFormulaUnitary → GlobalTheory.Published.SiegelWeilUnitaryConvergent → GlobalTheory.Published.DoublingMethod → GlobalTheory.Published.UnramifiedDoublingGL → ∀ D : D7, HeckeForm D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeFormOfDoublingStatement

theorem **_root_.GlobalTheory.T.hecke_form_of_doubling** : GlobalTheory.Published.WeilRepresentationUnitaryPair → GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.WeilSiegelFormulaUnitary → GlobalTheory.Published.SiegelWeilUnitaryConvergent → GlobalTheory.Published.DoublingMethod → GlobalTheory.Published.UnramifiedDoublingGL → ∀ D : D7, HeckeForm D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeIdentityIntegral

theorem **GlobalTheory.ParHecke.heckeIdentity_integral** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (D D' : Type) [Fintype D] [DecidableEq D] [Fintype D'] [DecidableEq D'] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ t : ℕ, Fintype.card D' = Fintype.card D + t → ∀ (s : Finset D') RK, IsRepsK (torusElt k D' ϖ (indic s)) RK → ∀ (sn : ℕ → Finset D) (Rn : ℕ → Finset (GL D k)), (∀ n ≤ Fintype.card D, (sn n).card = n ∧ IsRepsK (torusElt k D ϖ (-indic (sn n))) (Rn n)) → ∀ y : Matrix D' D k, ∑ g ∈ RK, intInd ((g⁻¹).1 * y) = ∑ n ∈ Finset.range (Fintype.card D + 1), (if n ≤ s.card then ((resCard k ^ (n * (t - (s.card - n))) * Formalism.L2.Col.gaussBin (resCard k) t (s.card - n) : ℕ) : ℂ) else 0) * ∑ g ∈ Rn n, intInd (y * g.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeInnerProductFormulaOfDoublingEuler

theorem **heckeForm_of_global_euler** : ∀ D : D7, GlobalForm D → EulerRight D → HeckeForm D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeInnerProductFormulaOfDoublingEulerStatement

theorem **_root_.GlobalTheory.T.heckeInnerProductFormula_of_doublingEuler** : ∀ D : D7, GlobalForm D → EulerRight D → HeckeForm D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeLPartialContinuation

theorem **heckeLPartialContinuation** : GlobalTheory.Published.HeckeLPartialContinuation

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeOperatorInverseCosetConjEigenvalue

theorem **exists_dual_rep** (J : Set (GA D V)) (χ : GA D V → ℂ) (Sb : Finset (HeightOneSpectrum (𝒪 D.host.F))) (P : HeightOneSpectrum (𝒪 D.host.E)) (d : List ℤ) (R : Finset (GA D V)) (e f : Formalism.L2.Theta.AE D) (ϖ : 𝒪 D.host.E) (α : GA D V) (hS : OffS D Sb P) (hef : IsPrimeIdem P e f) (hϖ : ϖ ∈ P.asIdeal) (hϖ2 : ϖ ∉ P.asIdeal ^ 2) (hd : d.Pairwise (fun a b => b ≤ a)) (hJ : ∀ g, SuppF V f g → (g ∈ J ↔ g ∈ KsphC V ∅)) (hχ : ∀ g ∈ J, SuppF V f g → χ g = 1) (hα : SuppF V f α) (hdiv : DivAt V e (ϖ : D.host.E) d α) (hrep : IsCosetRep V {k | k ∈ J ∧ SuppF V f k} α R) : ∃ R' : Finset (GA D V), Hecke₂ V J χ Sb P (dualLab d) R' ∧ R'.card = R.card ∧ ∀ γ ∈ R', ∃ k₁ ∈ {k | k ∈ J ∧ SuppF V f k}, ∃ k₂ ∈ {k | k ∈ J ∧ SuppF V f k}, γ = k₁ ∗ α⁻¹ ∗ k₂

theorem **form_sum** (B : bddAut V →ₗ[ℂ] bddAut V →ₛₗ[starRingEnd ℂ] ℂ) (hinv : ∀ g v w, B (bddRep V g v) (bddRep V g w) = B v w) (K : Set (GA D V)) (F : bddAut V) (hF : ∀ k ∈ K, bddRep V k F = F) (α : GA D V) (R : Finset (GA D V)) (hR : ∀ γ ∈ R, ∃ k₁ ∈ K, ∃ k₂ ∈ K, γ = k₁ ∗ α ∗ k₂) : B (∑ γ ∈ R, bddRep V γ F) F = (R.card : ℂ) ∗ B (bddRep V α F) F

theorem **form_inv** (B : bddAut V →ₗ[ℂ] bddAut V →ₛₗ[starRingEnd ℂ] ℂ) (hsym : ∀ v w, B w v = starRingEnd ℂ (B v w)) (hinv : ∀ g v w, B (bddRep V g v) (bddRep V g w) = B v w) (F : bddAut V) (α : GA D V) : B (bddRep V α⁻¹ F) F = starRingEnd ℂ (B (bddRep V α F) F)

theorem **coe_sum_bddRep** (f : GA D V → ℂ) (hf : f ∈ bddAut V) (R : Finset (GA D V)) : ((∑ γ ∈ R, bddRep V γ (f, hf) : bddAut V) : GA D V → ℂ) = heckeLin V R f

theorem **heckeLin_eq_conj_of_dualLab** : ∀ (D : D7) (V : HSp D) (J : Set (GA D V)) (χ : GA D V → ℂ) (Sb : Finset (HeightOneSpectrum (𝒪 D.host.F))), SphOff J χ Sb → ∀ f ∈ GlobalTheory.FinZeta.bddAut V, (∀ k ∈ J, ∀ g, f (g ∗ k) = χ k ∗ f g) → ∀ e : HeightOneSpectrum (𝒪 D.host.E) → List ℤ → ℂ, (∀ P d R, Hecke₂ V J χ Sb P d R → heckeLin V R f = e P (dualLab d) • f) → ∀ P d R, Hecke₂ V J χ Sb P d R → heckeLin V R f = conj (e P d) • f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeOperatorInverseCosetConjEigenvalueStatement

theorem **_root_.GlobalTheory.T.heckeOperator_inverseCoset_conj_eigenvalue** : ∀ (D : D7) (V : HSp D) (J : Set (GA D V)) (χ : GA D V → ℂ) (Sb : Finset (HeightOneSpectrum (𝒪 D.host.F))), SphOff J χ Sb → ∀ f ∈ GlobalTheory.FinZeta.bddAut V, (∀ k ∈ J, ∀ g, f (g ∗ k) = χ k ∗ f g) → ∀ e : HeightOneSpectrum (𝒪 D.host.E) → List ℤ → ℂ, (∀ P d R, Hecke₂ V J χ Sb P d R → heckeLin V R f = e P (dualLab d) • f) → ∀ P d R, Hecke₂ V J χ Sb P d R → heckeLin V R f = conj (e P d) • f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeOperatorMoveAcrossInvariantBilinear

theorem **mem_doubleCoset_inv_iff** {a g : G} : g⁻¹ ∈ DoubleCoset.doubleCoset a (K : Set G) K ↔ g ∈ DoubleCoset.doubleCoset a⁻¹ (K : Set G) K

theorem **dosetInd_comp_inv** (a : G) : (fun g => dosetInd ℂ K a g⁻¹) = dosetInd ℂ K a⁻¹

theorem **out_mem_doubleCoset_iff** (a : G) (c : G ⧸ K) : c.out ∈ DoubleCoset.doubleCoset a (K : Set G) K ↔ ∃ k ∈ K, c = ((k ∗ a : G) : G ⧸ K)

private theorem **finite_left_cosets_of_isCommensurated** (hK : IsCommensurated K) (a : G) : {c : G ⧸ K | ∃ k ∈ K, c = ((k ∗ a : G) : G ⧸ K)}.Finite

theorem **finsum_dosetInd_out** (hK : IsCommensurated K) (a : G) : ∑ᶠ c : G ⧸ K, dosetInd ℂ K a c.out = (Nat.card {c : G ⧸ K // ∃ k ∈ K, c = ((k ∗ a : G) : G ⧸ K)} : ℂ)

private theorem **apply_eq_of_mem_fixed** {G V : Type} [Group G] {K : Subgroup G} [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ G V) {v : V} (hv : v ∈ fixed ρ K) {k : G} (hk : k ∈ K) : ρ k v = v

private theorem **invariant_pairing_doubleCoset_eq** {G U X X' : Type} [Group G] {K : Subgroup G} [AddCommGroup U] [Module ℂ U] [AddCommGroup X] [Module ℂ X] [AddCommGroup X'] [Module ℂ X'] (π : Representation ℂ G U) (ρ : Representation ℂ G X) (b : U →ₗ[ℂ] X →ₗ[ℂ] X') (hb : ∀ g u y, b (π g u) (ρ g y) = b u y) {u : U} (hu : u ∈ fixed π K) {y : X} (hy : y ∈ fixed ρ K) {a w : G} (hw : w ∈ DoubleCoset.doubleCoset a (K : Set G) K) : b (π w u) y = b (π a u) y

private theorem **pairing_heckeVec_dosetInd_eq_card_smul** {G U X X' : Type} [Group G] {K : Subgroup G} (hK : IsCommensurated K) [AddCommGroup U] [Module ℂ U] [AddCommGroup X] [Module ℂ X] [AddCommGroup X'] [Module ℂ X'] (π : Representation ℂ G U) (ρ : Representation ℂ G X) (b : U →ₗ[ℂ] X →ₗ[ℂ] X') (hb : ∀ g u y, b (π g u) (ρ g y) = b u y) {u : U} (hu : u ∈ fixed π K) {y : X} (hy : y ∈ fixed ρ K) (r : ℂ) (a : G) : b (heckeVec π K (r • dosetInd ℂ K a) u) y = (r ∗ (Nat.card {c : G ⧸ K // ∃ k ∈ K, c = ((k ∗ a : G) : G ⧸ K)} : ℂ)) • b (π a u) y

private theorem **comp_inv_mem_hecke** {G : Type} [Group G] {K : Subgroup G} {ψ : G → ℂ} (hψ : ψ ∈ Hecke ℂ K) : (fun g => ψ g⁻¹) ∈ Hecke ℂ K

theorem **heckeVec_move** : ∀ (G : Type) [Group G] (K : Subgroup G), IsCommensurated K → CosetSym K → ∀ (U X X' : Type) [AddCommGroup U] [Module ℂ U] [AddCommGroup X] [Module ℂ X] [AddCommGroup X'] [Module ℂ X'] (π : Representation ℂ G U) (ρ : Representation ℂ G X) (b : U →ₗ[ℂ] X →ₗ[ℂ] X'), (∀ g u y, b (π g u) (ρ g y) = b u y) → ∀ f ∈ fixed π K, ∀ x ∈ fixed ρ K, ∀ φ ∈ Hecke ℂ K, (fun g => φ g⁻¹) ∈ Hecke ℂ K ∧ b (heckeVec π K φ f) x = b f (heckeVec ρ K (fun g => φ g⁻¹) x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeOperatorMoveAcrossInvariantBilinearStatement

theorem **_root_.GlobalTheory.T.hecke_operator_move_across_invariant_bilinear** : ∀ (G : Type) [Group G] (K : Subgroup G), IsCommensurated K → CosetSym K → ∀ (U X X' : Type) [AddCommGroup U] [Module ℂ U] [AddCommGroup X] [Module ℂ X] [AddCommGroup X'] [Module ℂ X'] (π : Representation ℂ G U) (ρ : Representation ℂ G X) (b : U →ₗ[ℂ] X →ₗ[ℂ] X'), (∀ g u y, b (π g u) (ρ g y) = b u y) → ∀ f ∈ fixed π K, ∀ x ∈ fixed ρ K, ∀ φ ∈ Hecke ℂ K, (fun g => φ g⁻¹) ∈ Hecke ℂ K ∧ b (heckeVec π K φ f) x = b f (heckeVec ρ K (fun g => φ g⁻¹) x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeParametersOfElementFromOneNonzeroValue

theorem **GlobalTheory.ChainOnto.parOfVal_of_parts** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, MetEx D → LevFacts D → ParOfVal D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeParametersOfElementFromOneNonzeroValueStatement

theorem **_root_.GlobalTheory.T.hecke_parameters_of_element_from_one_nonzero_value** : GlobalTheory.Published.CuspRapidDecay → ∀ D : D7, MetEx D → LevFacts D → ParOfVal D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeSeriesCellHasSum

theorem **isInt_mul** {A B : GL (Fin n) k} (hA : IsInt A) (hB : IsInt B) : IsInt (A * B)

theorem **isInt_mul_KO_iff** {A κ : GL (Fin n) k} (hκ : κ ∈ KO k (Fin n)) : IsInt (A * κ) ↔ IsInt A

theorem **isInt_KO_mul_iff** {A κ : GL (Fin n) k} (hκ : κ ∈ KO k (Fin n)) : IsInt (κ * A) ↔ IsInt A

theorem **ord_det_of_mem_KO** {κ : GL (Fin n) k} (hκ : κ ∈ KO k (Fin n)) : ord k (Matrix.GeneralLinearGroup.det κ) = 0

theorem **ord_det_mul** (a b : GL (Fin n) k) : ord k (Matrix.GeneralLinearGroup.det (a * b)) = ord k (Matrix.GeneralLinearGroup.det a) + ord k (Matrix.GeneralLinearGroup.det b)

theorem **ord_det_inv** (a : GL (Fin n) k) : ord k (Matrix.GeneralLinearGroup.det a⁻¹) = -ord k (Matrix.GeneralLinearGroup.det a)

theorem **valuation_det_le_one** {A : GL (Fin n) k} (hA : IsInt A) : valuation k ((Matrix.GeneralLinearGroup.det A : kˣ) : k) ≤ 1

theorem **ord_nonneg_of_le_one** {x : kˣ} (hx : valuation k (x : k) ≤ 1) : 0 ≤ ord k x

theorem **ord_det_torusElt** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (e : Fin n → ℤ) : ord k (Matrix.GeneralLinearGroup.det (torusElt k (Fin n) ϖ e)) = ∑ i, e i

theorem **isInt_and_mul_torus_iff** {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (A : GL (Fin n) k) (e : Fin n → ℤ) : IsInt A ∧ IsInt (A * torusElt k (Fin n) ϖ e) ↔ IsInt (A * torusElt k (Fin n) ϖ fun i => min (e i) 0)

theorem **out_eq_mul** {r : GL (Fin n) k} {c : GL (Fin n) k ⧸ KO k (Fin n)} (h : (r : GL (Fin n) k ⧸ KO k (Fin n)) = c) : ∃ κ ∈ KO k (Fin n), c.out = r * κ

theorem **smul_out** (u : GL (Fin n) k) (c : GL (Fin n) k ⧸ KO k (Fin n)) : ∃ κ ∈ KO k (Fin n), (u • c).out = u * c.out * κ

theorem **hasSum_pair** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (ϖ : kˣ) (hϖ : valuation k (ϖ : k) = uniformizer k) (e : Fin n → ℤ) (g : GL (Fin n) k) (hg : g ∈ cell k (Fin n) ϖ e) (X : ℂ) (hX : ∀ i : Fin n, ‖(resCard k : ℂ) ^ (i : ℕ) * X ^ 2‖ < 1) : HasSum (fun h : {h : GL (Fin n) k ⧸ KO k (Fin n) // (∀ i j, (h.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k]) ∧ (∀ i j, (h.out⁻¹ * g : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ ((-ord k (Matrix.GeneralLinearGroup.det h.1.out)).toNat + (ord k (Matrix.GeneralLinearGroup.det (h.1.out⁻¹ * g))).toNat)) (X ^ (∑ i, |e i|).toNat * ∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * X ^ 2)⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeckeSeriesCellHasSumStatement

theorem **_root_.GlobalTheory.T.hecke_series_cell_has_sum** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : ℕ) (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (e : Fin n → ℤ), ∀ g ∈ cell k (Fin n) ϖ e, ∀ X : ℂ, (∀ i : Fin n, ‖(resCard k : ℂ) ^ (i : ℕ) * X ^ 2‖ < 1) → HasSum (fun h : {h : GL (Fin n) k ⧸ KO k (Fin n) // (∀ i j, (h.out⁻¹ : GL (Fin n) k).1 i j ∈ 𝒪[k]) ∧ (∀ i j, (h.out⁻¹ * g : GL (Fin n) k).1 i j ∈ 𝒪[k])} => X ^ ((-ord k (Matrix.GeneralLinearGroup.det h.1.out)).toNat + (ord k (Matrix.GeneralLinearGroup.det (h.1.out⁻¹ * g))).toNat)) (X ^ (∑ i, |e i|).toNat * ∏ i : Fin n, (1 - (resCard k : ℂ) ^ (i : ℕ) * X ^ 2)⁻¹)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeightIoLe

theorem **height_io_le** (D : D7) (A : HSp D) (g₁ g₂ : GA D A) : GlobalTheory.Grow.height (dblSp A) (io A (g₁, g₂)) ≤ GlobalTheory.Grow.height A g₁ * GlobalTheory.Grow.height A g₂

theorem **main** : ∀ (D : D7) (A : HSp D), ∃ (C : ℝ) (c : ℕ), ∀ g₁ g₂ : GA D A, GlobalTheory.Grow.height (dblSp A) (io A (g₁, g₂)) ≤ C * (GlobalTheory.Grow.height A g₁ * GlobalTheory.Grow.height A g₂) ^ c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeightTwoFormalGroupUnramifiedMultiplier

theorem **heightTwo_unitTwist** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] [CompleteSpace K] (l : ℕ) (σ : K →+* K), Unr K l σ → ResAlg K → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → σ (σ w) = w → ‖σ w - w‖ = 1 → (∃ n, n ≤ l ^ 2 ∧ ‖coeff n (Formalism.L2.Th1.C.mulBy f e (l : K))‖ = 1) → ∃ ε : K, ‖ε‖ = 1 ∧ CoefLe 1 (e.subst (C ε * ltLog K l))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeightTwoFormalGroupUnramifiedMultiplierStatement

theorem **_root_.GlobalTheory.T.heightTwo_formalGroup_unramified_multiplier** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] [CompleteSpace K] (l : ℕ) (σ : K →+* K), Unr K l σ → ResAlg K → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → σ (σ w) = w → ‖σ w - w‖ = 1 → (∃ n, n ≤ l ^ 2 ∧ ‖coeff n (Formalism.L2.Th1.C.mulBy f e (l : K))‖ = 1) → ∃ ε : K, ‖ε‖ = 1 ∧ CoefLe 1 (e.subst (C ε * ltLog K l))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeightTwoHondaTypeFirstCoeffNormEqOne

theorem **norm_natCast_coprime_eq_one** (l u : ℕ) (hp : l.Prime) (hl1 : ‖(l : K)‖ < 1) (hu : ¬ l ∣ u) : ‖(u : K)‖ = 1

theorem **pow_le_norm_natCast** (l : ℕ) (hp : l.Prime) (hl1 : ‖(l : K)‖ < 1) (k : ℕ) (hk : 1 ≤ k) : ‖(l : K)‖ ^ (k - 1) ≤ ‖(k : K)‖

theorem **coeff_mul_pow_le** (l : ℕ) (hp : l.Prime) (hl1 : ‖(l : K)‖ < 1) (f : K⟦X⟧) (hder : CoefLe 1 (derivative K f)) (k : ℕ) (hk : 1 ≤ k) : ‖coeff k f‖ * ‖(l : K)‖ ^ k ≤ ‖(l : K)‖

theorem **norm_coeff_pow_le** (g : K⟦X⟧) (r : ℝ) (m : ℕ) (hr : 0 ≤ r) (hg : ∀ j ≤ m, ‖coeff j g‖ ≤ r) : ∀ k j, j ≤ m → ‖coeff j (g ^ k)‖ ≤ r ^ k

theorem **norm_coeff_succ_pow_le** (g : K⟦X⟧) (hg0 : constantCoeff g = 0) (r : ℝ) (m : ℕ) (hr : 0 ≤ r) (hg : ∀ j ≤ m, ‖coeff j g‖ ≤ r) (k : ℕ) : ‖coeff (m + 1) (g ^ (k + 2))‖ ≤ r ^ (k + 2)

theorem **coeff_subst_eq_sum** (f g : K⟦X⟧) (hg0 : constantCoeff g = 0) (n : ℕ) : coeff n (subst g f) = ∑ d ∈ Finset.range (n + 1), coeff d f * coeff n (g ^ d)

theorem **norm_coeff_le_of_subst** (f g : K⟦X⟧) (ρ : ℝ) (hρ : 0 ≤ ρ) (hf0 : coeff 0 f = 0) (hf1 : coeff 1 f = 1) (hf : ∀ k, 1 ≤ k → ‖coeff k f‖ * ρ ^ k ≤ ρ) (hg0 : constantCoeff g = 0) (N : ℕ) (hy : ∀ n ≤ N, ‖coeff n (subst g f)‖ ≤ ρ) : ∀ n ≤ N, ‖coeff n g‖ ≤ ρ

theorem **norm_coeff_hondaSum_le** (τ : K →+* K) (q : ℕ) (hq : 2 ≤ q) (c : ℕ → K) (f : K⟦X⟧) (ρ : ℝ) (hρ : 0 ≤ ρ) (hc0 : ‖c 0‖ ≤ ρ) (hf1 : coeff 1 f = 1) (n : ℕ) (hn : n ≤ q) : ‖coeff n (hondaSum τ q c f)‖ ≤ ρ

theorem **norm_first_coeff_eq_one** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ c : ℕ → K, (∀ j, ‖c j‖ ≤ 1) → CoefLe ‖(l : K)‖ (C (l : K) * f - hondaSum (σ ^ 2) (l ^ 2) c f) → (∃ n, n ≤ l ^ 2 ∧ ‖coeff n (Formalism.L2.Th1.C.mulBy f e (l : K))‖ = 1) → ‖c 0‖ = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeightTwoHondaTypeFirstCoeffNormEqOneStatement

theorem **_root_.GlobalTheory.T.height_two_hondaType_first_coeff_norm_eq_one** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ c : ℕ → K, (∀ j, ‖c j‖ ≤ 1) → CoefLe ‖(l : K)‖ (C (l : K) * f - hondaSum (σ ^ 2) (l ^ 2) c f) → (∃ n, n ≤ l ^ 2 ∧ ‖coeff n (Formalism.L2.Th1.C.mulBy f e (l : K))‖ = 1) → ‖c 0‖ = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisOConstMul

theorem **idAt_mul_idOff** {D : D7} (P : Pl D → Prop) (a b : AE D) (ha : a ∈ idAt D P) (hb : b ∈ idOff D P) : a * b = 0

theorem **psiE_add** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (a b : AE D) : psiE ψ (a + b) = psiE ψ a * psiE ψ b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisOFinsetSum

theorem **heisO_finset_sum** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (x : WO S T) {ι : Type} (s : Finset ι) (a : ι → ℂ) (g : ι → WO S T → ℂ) : heisO ψ S T x (∑ i ∈ s, a i • g i) = ∑ i ∈ s, a i • heisO ψ S T x (g i)

theorem **comb_eq_sum** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T c f = c.sum fun u a => a • heisO ψ S T u f

theorem **comb_comb** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c c' : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T c (comb ψ S T c' f) = ∑ x ∈ c.support, ∑ y ∈ c'.support, (c x * c' y) • heisO ψ S T x (heisO ψ S T y f)

theorem **symplO_half** (S T : HSp D) (e e' : AE D) (x y : WO S T) (he : cA D e = e) (hee : e * e' = 0) (hx : IsHalf e x) (hy : IsHalf e' y) : symplO S T x y = 0

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D), (∀ c c' : WO S T →₀ ℂ, ∃ c'' : WO S T →₀ ℂ, (∀ w ∈ c''.support, ∃ x ∈ c.support, ∃ y ∈ c'.support, w = x + y) ∧ ∀ f, comb ψ S T c'' f = comb ψ S T c (comb ψ S T c' f)) ∧ (∀ c c' : WO S T →₀ ℂ, (∀ x ∈ c.support, ∀ y ∈ c'.support, symplO S T x y = 0) → ∀ f, comb ψ S T c (comb ψ S T c' f) = comb ψ S T c' (comb ψ S T c f)) ∧ (∀ (c : WO S T →₀ ℂ) f, IsThetaFunO ψ S T f → IsFinComb c → IsStdChar D ψ (-1) → IsThetaFunO ψ S T (comb ψ S T c f)) ∧ ∀ (e e' : AE D) (x y : WO S T), cA D e = e → e * e' = 0 → IsHalf e x → IsHalf e' y → symplO S T x y = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergAverageRealPlacesSmooth

theorem **contDiff_integral_kernel** {R : Type} [NormedAddCommGroup R] [NormedSpace ℝ R] [FiniteDimensional ℝ R] [MeasurableSpace R] [BorelSpace R] (μ : Measure R) [μ.IsAddHaarMeasure] (φ : R → ℝ) (hφ : ContDiff ℝ ∞ φ) (hφc : HasCompactSupport φ) (K : R → R → ℂ) (hK : ContDiff ℝ ∞ fun p : R × R => K p.1 p.2) (G : R → ℂ) (hG : Continuous G) : ContDiff ℝ ∞ fun y => ∫ x, (φ x : ℂ) * K y x * G (y + x) ∂μ

theorem **sm_heisO_average** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), GlobalTheory.FockCompare.Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T : HSp D) (φ : GlobalTheory.FockCompare.RW T A → ℝ), ContDiff ℝ ∞ φ → HasCompactSupport φ → ∀ (f : WO A T → ℂ), Continuous f → GlobalTheory.FockCompare.Sm (GlobalTheory.FockCompare.iotaW σ T A) (fun w => ∫ x, (φ x : ℂ) * heisO ψ A T (GlobalTheory.FockCompare.iotaW σ T A x) f w)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergAverageRealPlacesSmoothStatement

theorem **_root_.GlobalTheory.T.heisenbergAverage_realPlaces_smooth** : ∀ (D : D7) (θ : InfFam D) (σ : (D.what₂.Tau → ℂ) →+ AE D), GlobalTheory.FockCompare.Chart θ σ → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T : HSp D) (φ : GlobalTheory.FockCompare.RW T A → ℝ), ContDiff ℝ ∞ φ → HasCompactSupport φ → ∀ (f : WO A T → ℂ), Continuous f → GlobalTheory.FockCompare.Sm (GlobalTheory.FockCompare.iotaW σ T A) (fun w => ∫ x, (φ x : ℂ) * heisO ψ A T (GlobalTheory.FockCompare.iotaW σ T A x) f w)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergBasisDerivEqOfUniformLimits

theorem **tendstoUniformly_heisO** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {G : ℕ → WO S T → ℂ} {F : WO S T → ℂ} (h : TendstoUniformly G F Filter.atTop) (x : ℝ → WO S T) (w : WO S T) : TendstoUniformly (fun j t => heisO ψ S T (x t) (G j) w) (fun t => heisO ψ S T (x t) F w) Filter.atTop

theorem **heisDer_of_tendstoUniformly** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (G : ℕ → WO S T → ℂ) (F F' : WO S T → ℂ), (∀ j, IsThetaFunO ψ S T (G j)) → TendstoUniformly G F Filter.atTop → ∀ a : (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool), TendstoUniformly (fun j => GlobalTheory.Size.heisDer ψ θ a (G j)) F' Filter.atTop → GlobalTheory.Size.heisDer ψ θ a F = F'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergBasisDerivEqOfUniformLimitsStatement

theorem **_root_.GlobalTheory.T.heisenbergBasisDeriv_eq_of_uniform_limits** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (G : ℕ → WO S T → ℂ) (F F' : WO S T → ℂ), (∀ j, IsThetaFunO ψ S T (G j)) → TendstoUniformly G F Filter.atTop → ∀ a : (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool), TendstoUniformly (fun j => GlobalTheory.Size.heisDer ψ θ a (G j)) F' Filter.atTop → GlobalTheory.Size.heisDer ψ θ a F = F'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombAwayFromPlaceKeepsVacuum

theorem **comb_single_one** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (y : WO S T) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single y 1) f = heisO ψ S T y f

theorem **vac_comb** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q₀ : QStep S) (c₁ : WO S.B.V S.J.W →₀ ℂ) (c₂ : WO S.J.W S.B'.V →₀ ℂ), IsFinComb c₁ → IsFinComb c₂ → Q₀.Away c₁ → Q₀.Away c₂ → ∀ f₁ f₂, IsThetaFunO ψ _ _ f₁ → IsThetaFunO ψ _ _ f₂ → Q₀.Vac f₁ f₂ → Q₀.Vac (comb ψ _ _ c₁ f₁) (comb ψ _ _ c₂ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombAwayFromPlaceKeepsVacuumStatement

theorem **_root_.GlobalTheory.T.heisenbergComb_awayFromPlace_keeps_vacuum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q₀ : QStep S) (c₁ : WO S.B.V S.J.W →₀ ℂ) (c₂ : WO S.J.W S.B'.V →₀ ℂ), IsFinComb c₁ → IsFinComb c₂ → Q₀.Away c₁ → Q₀.Away c₂ → ∀ f₁ f₂, IsThetaFunO ψ _ _ f₁ → IsThetaFunO ψ _ _ f₂ → Q₀.Vac f₁ f₂ → Q₀.Vac (comb ψ _ _ c₁ f₁) (comb ψ _ _ c₂ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationCommFockPolynomialVectors

theorem **foldr_lin** {ι X : Type} (p : (X → ℂ) → Prop) (A : ι → (X → ℂ) → (X → ℂ)) (hp : ∀ i f, p f → p (A i f)) (hadd : ∀ i f g, p f → p g → A i (f + g) = A i f + A i g) (hsmul : ∀ i (c : ℂ) f, p f → A i (c • f) = c • A i f) (padd : ∀ f g, p f → p g → p (f + g)) (psmul : ∀ (c : ℂ) f, p f → p (c • f)) (c : ℂ) {f g : X → ℂ} (hf : p f) (hg : p g) (l : List ι) : l.foldr (fun i h => A i h) (c • f + g) = c • l.foldr (fun i h => A i h) f + l.foldr (fun i h => A i h) g

theorem **pureTensorO_lin** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (c : ℂ) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) (φ : MvPolynomial (FVar D T.rank S.rank) ℂ) : pureTensorO ψ θ S T sS sT BS BT (c • f + g) φ = c • pureTensorO ψ θ S T sS sT BS BT f φ + pureTensorO ψ θ S T sS sT BS BT g φ

theorem **pureTensorO_zero** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (φ : MvPolynomial (FVar D T.rank S.rank) ℂ) : pureTensorO ψ θ S T sS sT BS BT (0 : WO S T → ℂ) φ = 0

theorem **heis_comb_pureTensorO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D), IsStdChar D ψ (-1) → IsInfFam θ → ∀ sS sT BS BT (f : WO S T → ℂ), IsThetaFunO ψ S T f → ∀ (n : ℕ) (a : Fin n → ℂ) (y : Fin n → WO S T), (∀ i, (y i).map (infPart D) = 0) → ∀ φ, ∑ i, a i • heisO ψ S T (y i) (pureTensorO ψ θ S T sS sT BS BT f φ) = pureTensorO ψ θ S T sS sT BS BT (∑ i, a i • heisO ψ S T (y i) f) φ ∧ pureTensorO ψ θ S T sS sT BS BT (0 : WO S T → ℂ) φ = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationCommFockPolynomialVectorsStatement

theorem **_root_.GlobalTheory.T.heisenberg_combination_comm_fockPolynomialVectors** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D), IsStdChar D ψ (-1) → IsInfFam θ → ∀ sS sT BS BT (f : WO S T → ℂ), IsThetaFunO ψ S T f → ∀ (n : ℕ) (a : Fin n → ℂ) (y : Fin n → WO S T), (∀ i, (y i).map (infPart D) = 0) → ∀ φ, ∑ i, a i • heisO ψ S T (y i) (pureTensorO ψ θ S T sS sT BS BT f φ) = pureTensorO ψ θ S T sS sT BS BT (∑ i, a i • heisO ψ S T (y i) f) φ ∧ pureTensorO ψ θ S T sS sT BS BT (0 : WO S T → ℂ) φ = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationDisjointPlacesCommuteVacua

theorem **mul_eq_zero_of_disj** {P P' : Pl D → Prop} (hd : ∀ p, P p → ¬ P' p) {a b : AE D} (ha : a ∈ idAt D P) (hb : b ∈ idAt D P') : a * b = 0

theorem **mul_eq_zero_of_off** {P : Pl D → Prop} {a b : AE D} (ha : a ∈ idOff D P) (hb : b ∈ idAt D P) : a * b = 0

theorem **idem_mem** (P : Pl D → Prop) : idem P ∈ idAt D P

theorem **cA_idem** (P : Pl D → Prop) : cA D (idem P) = idem P

theorem **idem_mul** {P : Pl D → Prop} {a : AE D} (ha : a ∈ idAt D P) : idem P * a = a

theorem **symplO_disj** {S T : HSp D} {P P' : Pl D → Prop} (hd : ∀ p, P p → ¬ P' p) {x y : WO S T} (hx : MxAt P x) (hy : MxAt P' y) : symplO S T x y = 0

theorem **mxAt_mono** {P Q : Pl D → Prop} (h : ∀ p, P p → Q p) {m n : ℕ} {x : Mx D m n} (hx : MxAt P x) : MxAt Q x

theorem **comb_single** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w : WO S T) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single w 1) f = heisO ψ S T w f

theorem **mem_congGrp'** {V : HSp D} {I : Ideal (AE D)} {g : GA D V} (hg : g ∈ congGrp V I) : ∀ i j, mt V g i j - (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D)) i j ∈ I

theorem **actO_off** {S T : HSp D} {P : Pl D → Prop} {g : GA D S} {h : GA D T} (hg : g ∈ grpOff S P) (hh : h ∈ grpOff T P) {u : WO S T} (hu : MxAt P u) : actO S T g h u = u

theorem **compAt_idem** {P : Pl D → Prop} {w : Pl D} (hpw : P w) : compAt D w (idem P) = 1

theorem **sub_idem_mem** {P P' : Pl D → Prop} {a : AE D} (ha : a ∈ idAt D (fun p => P p ∨ P' p)) : a - idem P * a ∈ idAt D P'

theorem **split_latO** (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {LS : Lat S} {LT : Lat T} (hLS : IsSelfDualC S LS) (hLT : IsSelfDualC T LT) {P P' : Pl D → Prop} (hd : ∀ p, P p → ¬ P' p) {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) (fun p => P p ∨ P' p)) : MxAt P (idem P • u) ∧ MxAt P' (u - idem P • u) ∧ idem P • u ∈ latO S T (LS : Set _) (LT : Set _) P ∧ u - idem P • u ∈ latO S T (LS : Set _) (LT : Set _) P'

theorem **isHalf_parts** {S T : HSp D} {P : Pl D → Prop} {e : AE D} {u : WO S T} (h : IsHalf e u) : IsHalf e (idem P • u) ∧ IsHalf e (u - idem P • u)

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (P P' : Pl D → Prop), (∀ p, P p → ¬ P' p) → (∀ e e' : WO S T →₀ ℂ, CombAt P e → CombAt P' e' → ∀ f, comb ψ S T e (comb ψ S T e' f) = comb ψ S T e' (comb ψ S T e f)) ∧ (∀ e : WO S T →₀ ℂ, CombAt P e → IsFinComb e ∧ ∀ LS LT f, Vac ψ S T LS LT P' f → Vac ψ S T LS LT P' (comb ψ S T e f)) ∧ (∀ χS χT (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ e : WO S T →₀ ℂ, CombAt P e → ∀ g ∈ grpOff S P, ∀ h ∈ grpOff T P, ∀ f, IsThetaFunO ψ S T f → om (g, h) (comb ψ S T e f) = comb ψ S T e (om (g, h) f)) ∧ (∀ e e' : WO S T →₀ ℂ, CombAt (fun p => P p ∨ P' p) e → CombAt (fun p => P p ∨ P' p) e' → ∃ e'', CombAt (fun p => P p ∨ P' p) e'' ∧ ∀ f, comb ψ S T e'' f = comb ψ S T e (comb ψ S T e' f)) ∧ ∀ LS LT f, IsSelfDualC S LS → IsSelfDualC T LT → (Vac ψ S T LS LT (fun p => P p ∨ P' p) f ↔ Vac ψ S T LS LT P f ∧ Vac ψ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationDisjointPlacesCommuteVacuaStatement

theorem **_root_.GlobalTheory.T.heisenbergCombination_disjoint_places_commute_vacua** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (P P' : Pl D → Prop), (∀ p, P p → ¬ P' p) → (∀ e e' : WO S T →₀ ℂ, CombAt P e → CombAt P' e' → ∀ f, comb ψ S T e (comb ψ S T e' f) = comb ψ S T e' (comb ψ S T e f)) ∧ (∀ e : WO S T →₀ ℂ, CombAt P e → IsFinComb e ∧ ∀ LS LT f, Vac ψ S T LS LT P' f → Vac ψ S T LS LT P' (comb ψ S T e f)) ∧ (∀ χS χT (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ e : WO S T →₀ ℂ, CombAt P e → ∀ g ∈ grpOff S P, ∀ h ∈ grpOff T P, ∀ f, IsThetaFunO ψ S T f → om (g, h) (comb ψ S T e f) = comb ψ S T e (om (g, h) f)) ∧ (∀ e e' : WO S T →₀ ℂ, CombAt (fun p => P p ∨ P' p) e → CombAt (fun p => P p ∨ P' p) e' → ∃ e'', CombAt (fun p => P p ∨ P' p) e'' ∧ ∀ f, comb ψ S T e'' f = comb ψ S T e (comb ψ S T e' f)) ∧ ∀ LS LT f, IsSelfDualC S LS → IsSelfDualC T LT → (Va

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationEqOnVacuaOfEqOne

theorem **af_ext''** {D : D7} {x y : Af D.host.F} (h : ∀ w, evalV D.host.F w x = evalV D.host.F w y) : x = y

theorem **latO_signs** : ∀ (D : D7) (r : ℕ) (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (P : GlobalTheory.FinPl.Pl D → Prop), Function.Injective (fun i => (q i).v) → (∀ p, P p ↔ ∃ i, (q i).v = p) → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ x ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) P, ∃ ε : ℂ, ε * ε = 1 ∧ ∀ f, GlobalTheory.FinPl.Vac ψ S T LS LT P f → heisO ψ S T x f = ε • f

theorem **heis_comm** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (x γ : WO S T) (f : WO S T → ℂ) : heisO ψ S T γ (heisO ψ S T x f) = ψ (symplO S T γ x) • heisO ψ S T x (heisO ψ S T γ f)

theorem **comb_sub** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c₁ c₂ : WO S T →₀ ℂ) (f : WO S T → ℂ) : ∑ x ∈ (c₁ - c₂).support, (c₁ - c₂) x • heisLin ψ S T x f = comb ψ S T c₁ f - comb ψ S T c₂ f

theorem **comb_eq_on_vacua** : ∀ (D : D7) (r : ℕ) (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (P : GlobalTheory.FinPl.Pl D → Prop), Function.Injective (fun i => (q i).v) → (∀ p, P p ↔ ∃ i, (q i).v = p) → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ c₁ c₂ : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt P c₁ → GlobalTheory.FinPl.CombAt P c₂ → ∀ f₀, GlobalTheory.FinPl.Vac ψ S T LS LT P f₀ → f₀ ≠ 0 → comb ψ S T c₁ f₀ = comb ψ S T c₂ f₀ → ∀ f, GlobalTheory.FinPl.Vac ψ S T LS LT P f → comb ψ S T c₁ f = comb ψ S T c₂ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationEqOnVacuaOfEqOneStatement

theorem **_root_.GlobalTheory.T.heisenbergCombination_eq_on_vacua_of_eq_one** : ∀ (D : D7) (r : ℕ) (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (P : GlobalTheory.FinPl.Pl D → Prop), Function.Injective (fun i => (q i).v) → (∀ p, P p ↔ ∃ i, (q i).v = p) → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ c₁ c₂ : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt P c₁ → GlobalTheory.FinPl.CombAt P c₂ → ∀ f₀, GlobalTheory.FinPl.Vac ψ S T LS LT P f₀ → f₀ ≠ 0 → comb ψ S T c₁ f₀ = comb ψ S T c₂ f₀ → ∀ f, GlobalTheory.FinPl.Vac ψ S T LS LT P f → comb ψ S T c₁ f = comb ψ S T c₂ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationExtendByZeroSourceSlice

theorem **comb_extS_slice** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (c : WO A T →₀ ℂ) (f : WO S T → ℂ) (y : WO L T) (w : WO A T), comb ψ A T c (fun w' => f ((o.resS T).symm (w', y))) w = comb ψ S T (extS o T c) f ((o.resS T).symm (w, y))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationExtendByZeroSourceSliceStatement

theorem **_root_.GlobalTheory.T.heisenbergCombination_extendByZero_source_slice** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (c : WO A T →₀ ℂ) (f : WO S T → ℂ) (y : WO L T) (w : WO A T), comb ψ A T c (fun w' => f ((o.resS T).symm (w', y))) w = comb ψ S T (extS o T c) f ((o.resS T).symm (w, y))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationExtendByZeroTargetSlice

theorem **comb_extT_resT_symm** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (c : WO T A →₀ ℂ) (f : WO T S → ℂ) (y : WO T L) (w : WO T A), comb ψ T A c (fun w' => f ((o.resT T).symm (w', y))) w = comb ψ T S (extT o T c) f ((o.resT T).symm (w, y))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationExtendByZeroTargetSliceStatement

theorem **_root_.GlobalTheory.T.heisenbergCombination_extendByZero_target_slice** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (c : WO T A →₀ ℂ) (f : WO T S → ℂ) (y : WO T L) (w : WO T A), comb ψ T A c (fun w' => f ((o.resT T).symm (w', y))) w = comb ψ T S (extT o T c) f ((o.resT T).symm (w, y))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationTwoVariableModelAndLineExte

theorem **heisO_resS_slice** (ψ : AddChar (AF D.host.F) ℂ) {V A L S : HSp D} (o : OrthDec V A L) (F : WO V S → ℂ) (w u : WO A S) (v : WO L S) : heisO ψ V S ((o.resS S).symm (w, 0)) F ((o.resS S).symm (u, v)) = heisO ψ A S w (fun u' => F ((o.resS S).symm (u', v))) u

theorem **heisO_resT_slice** (ψ : AddChar (AF D.host.F) ℂ) {V A L S : HSp D} (o : OrthDec V A L) (F : WO S V → ℂ) (w u : WO S A) (v : WO S L) : heisO ψ S V ((o.resT S).symm (w, 0)) F ((o.resT S).symm (u, v)) = heisO ψ S A w (fun u' => F ((o.resT S).symm (u', v))) u

theorem **comb_resS_slice** (ψ : AddChar (AF D.host.F) ℂ) {V A L S : HSp D} (o : OrthDec V A L) (c : WO A S →₀ ℂ) (F : WO V S → ℂ) (u : WO A S) (v : WO L S) : comb ψ V S (c.mapDomain fun w => (o.resS S).symm (w, 0)) F ((o.resS S).symm (u, v)) = comb ψ A S c (fun u' => F ((o.resS S).symm (u', v))) u

theorem **comb_resT_slice** (ψ : AddChar (AF D.host.F) ℂ) {V A L S : HSp D} (o : OrthDec V A L) (c : WO S A →₀ ℂ) (F : WO S V → ℂ) (u : WO S A) (v : WO S L) : comb ψ S V (c.mapDomain fun w => (o.resT S).symm (w, 0)) F ((o.resT S).symm (u, v)) = comb ψ S A c (fun u' => F ((o.resT S).symm (u', v))) u

theorem **exists_isOne** (Y Y' : Type) [Fintype Y] [Fintype Y'] : ∃ Φ : SB (Matrix Y Y' k), IsOne Φ

theorem **source_slice_model** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∀ (a : (n → k) → ℂ), a ∈ SB _ → ∀ v : WO B.L S, ∃ T₁, p.IsModel ψ FA FS T₁ ∧ ∀ (Ξ : SB (Matrix n n' k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ * a fun r => x r (Sum.inr ())) → ∀ u, T₀ Ψ ((B.o.resS S).symm (u, v)) = T₁ Ξ u

theorem **target_slice_model** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ FS F₀ T₀ → ∀ (a : (n → k) → ℂ), a ∈ SB _ → ∀ v : WO S B.L, ∃ T₁, p.IsModel ψ FS FA T₁ ∧ ∀ (Ξ : SB (Matrix n' n k)) (Ψ : SB (Matrix (n' ⊕ Unit) n k)), (∀ x, Ψ.1 x = Ξ.1 x.toRows₁ * a (x (Sum.inr ()))) → ∀ u, T₀ Ψ ((B.o.resT S).symm (u, v)) = T₁ Ξ u

theorem **comb_bimodel_and_line** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''], (∀ (A S A' : HSp D) (FA : Frame k A.rank n') (FS : Frame k S.rank n) (FA' : Frame k A'.rank n'') Λ c c' (Ξ₁ O₁ : SB (Matrix n n' k)) (Ξ₂ O₂ : SB (Matrix n'' n k)) (Ξ O : SB (Matrix n n' k × Matrix n'' n k)), p.IsBiModel ψ FA FS FA' Λ → p.IsCombAt ψ FA FS c Ξ₁.1 → p.IsCombAt ψ FS FA' c' Ξ₂.1 → IsOne O₁ → IsOne O₂ → (∀ q, Ξ.1 q = Ξ₁.1 q.1 * Ξ₂.1 q.2) → (∀ q, O.1 q = O₁.1 q.1 * O₂.1 q.2) → Λ Ξ = biComb ψ A S A' c c' (Λ O)) ∧ ∀ (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), IsStdChar D ψ (-1) → p.IsCoreFrame B s FA F₀ → (∀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationTwoVariableModelAndLineExteStatement

theorem **_root_.GlobalTheory.T.heisenbergCombination_twoVariableModel_and_lineExtension** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''], (∀ (A S A' : HSp D) (FA : Frame k A.rank n') (FS : Frame k S.rank n) (FA' : Frame k A'.rank n'') Λ c c' (Ξ₁ O₁ : SB (Matrix n n' k)) (Ξ₂ O₂ : SB (Matrix n'' n k)) (Ξ O : SB (Matrix n n' k × Matrix n'' n k)), p.IsBiModel ψ FA FS FA' Λ → p.IsCombAt ψ FA FS c Ξ₁.1 → p.IsCombAt ψ FS FA' c' Ξ₂.1 → IsOne O₁ → IsOne O₂ → (∀ q, Ξ.1

q = Ξ₁.1 q.1 * Ξ₂.1 q.2) → (∀ q, O.1 q = O₁.1 q.1 * O₂.1 q.2) → Λ Ξ = biComb ψ A S A' c c' (Λ O)) ∧ ∀ (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.ra

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationsDistinctSplitPlacesCollect

theorem **foldr_comp_eq** {ι X : Type} (c o : ι → X → X) (hc : ∀ a b, a ≠ b → ∀ x, c a (o b x) = o b (c a x)) : ∀ (l : List ι), l.Nodup → ∀ F : X, l.foldr (fun i g => c i (o i g)) F = l.foldr (fun i g => c i g) (l.foldr (fun i g => o i g) F)

theorem **GlobalTheory.CollectComb.main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → GlobalTheory.SeeSaw.LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ F, (List.finRange n).foldr (fun i g => comb ψ S T (L i).e (comb ψ S T (L i).o g)) F = (List.finRange n).foldr (fun i g => comb ψ S T (L i).e g) ((List.finRange n).foldr (fun i g => comb ψ S T (L i).o g) F)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergCombinationsDistinctSplitPlacesCollectStatement

theorem **_root_.GlobalTheory.T.heisenbergCombinations_distinct_split_places_collect** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → GlobalTheory.SeeSaw.LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ F, (List.finRange n).foldr (fun i g => comb ψ S T (L i).e (comb ψ S T (L i).o g)) F = (List.finRange n).foldr (fun i g => comb ψ S T (L i).e g) ((List.finRange n).foldr (fun i g => comb ψ S T (L i).o g) F)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergDerivWeilTranslateContinuousCombinatio

theorem **actO_actO_inv** (g : GA D S) (h : GA D T) (x : WO S T) : actO S T g h (actO S T g⁻¹ h⁻¹ x) = x

theorem **dLine_apply** {θ : GlobalTheory.IndefiniteRowSideFullHecke.FockOrthogonalDecomposition.InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : Chart θ σ) {lam : AF D.host.F →+ ℂ} (hl : Continuous lam) (hlam : ∀ a, sndA D.host.F a = 0 → ψ a = Complex.exp (lam a)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S T) : dLine ψ S T L f w = phase lam S T (L 1) w * f w + fderiv ℝ (fun x => f (w + iotaW σ T S x)) 0 (piW θ T S (L 1))

theorem **dLine_eq_sum** (hψ : IsStdChar D ψ (-1)) {θ : GlobalTheory.IndefiniteRowSideFullHecke.FockOrthogonalDecomposition.InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : Chart θ σ) {ι : Type} [Fintype ι] (b : Module.Basis ι ℝ (RW T S)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) (N : ι → ℝ →+ WO S T) (hN : ∀ i, Continuous (N i)) (hN0 : ∀ i t, (N i t).map (finPart D) = 0) (hN1 : ∀ i, piW θ T S (N i 1) = b i) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : dLine ψ S T L f = ∑ i, ((b.coord i (piW θ T S (L 1)) : ℝ) : ℂ) • dLine ψ S T (N i) f

theorem **dLine_om_comb** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ M : ℝ →+ WO S T, Continuous M → (∀ s, (M s).map (finPart D) = 0) → ∃ (m : ℕ) (N : Fin m → ℝ →+ WO S T) (c : Fin m → GA D S × GA D T → ℂ), (∀ i, Continuous (N i)) ∧ (∀ i s, (N i s).map (finPart D) = 0) ∧ (∀ i, Continuous (c i)) ∧ ∀ p f, IsThetaFunO ψ S T f → GlobalTheory.WeilVacMove.dLine ψ S T M (om p f) = ∑ i, c i p • om p (GlobalTheory.WeilVacMove.dLine ψ S T (N i) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergDerivWeilTranslateContinuousCombinatioStatement

theorem **_root_.GlobalTheory.T.heisenbergDeriv_weilTranslate_continuous_combination** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ M : ℝ →+ WO S T, Continuous M → (∀ s, (M s).map (finPart D) = 0) → ∃ (m : ℕ) (N : Fin m → ℝ →+ WO S T) (c : Fin m → GA D S × GA D T → ℂ), (∀ i, Continuous (N i)) ∧ (∀ i s, (N i s).map (finPart D) = 0) ∧ (∀ i, Continuous (c i)) ∧ ∀ p f, IsThetaFunO ψ S T f → GlobalTheory.WeilVacMove.dLine ψ S T M (om p f) = ∑ i, c i p • om p (GlobalTheory.WeilVacMove.dLine ψ S T (N i) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergDerivativeDiagonalHalfEqSchrodingerDer

theorem **compAt_mul_rat** (θ : InfFam D) (hθ : IsInfFam θ) (τ : D.what₂.Tau) {m n k : ℕ} (w : Mx D m n) (Q : Matrix (Fin n) (Fin k) D.host.E) : compAt θ τ (w * rat Q) = compAt θ τ w * Q.map (tauE D τ)

theorem **isLineAt_extX** (θ : InfFam D) (hθ : IsInfFam θ) (A B : HSp D) (τ : D.what₂.Tau) (X : Matrix (Fin B.rank) (Fin A.rank) ℂ) (L : ℝ →+ Mx D B.rank A.rank) (hL : IsLineAt θ τ X L) : IsLineAt θ τ (X * (dblDec A).qX.map (tauE D τ)) (((dblDec A).extX B).comp L)

theorem **lineAt_isLineAt** (θ : InfFam D) (hθ : IsInfFam θ) {m n : ℕ} (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) : IsLineAt θ τ z (lineAt θ τ z)

theorem **lineAt_extX** (θ : InfFam D) (hθ : IsInfFam θ) (A B : HSp D) (τ : D.what₂.Tau) (X : Matrix (Fin B.rank) (Fin A.rank) ℂ) : (lineAt θ τ (X * (dblDec A).qX.map (tauE D τ)) : ℝ →+ WO (dblSp A) B) = ((dblDec A).extX B).comp (lineAt θ τ X)

theorem **hasDerivAt_mixS_extX** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {θ : InfFam D} (hθ : IsInfFam θ) {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {μ : Measure (Mx D T.rank ρ)} {𝓕 : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ 𝓕) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (M : ℝ →+ WO S T) (hM : IsRealDir T S M) (x : Mx D T.rank ρ) (w : WO S₀ T) : HasDerivAt (fun t : ℝ => d.mixS T ψ μ 𝓕 (heisO ψ S T (M t) f) x w) (d.mixS T ψ μ 𝓕 (GlobalTheory.OrientedHeis.dLine ψ S T M f) x w) 0

theorem **sch_dRho_diag** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → ∀ (τ : D.what₂.Tau) (X : Matrix (Fin B.rank) (Fin A.rank) ℂ) (x : Mx D B.rank A.rank), sch ψ A B ν 𝒴 (Formalism.IndefR.Datum.dRho (heisO ψ (dblSp A) B) θ τ (X * (dblDec A).qX.map (tauE D τ)) f) x = deriv (fun t : ℝ => sch ψ A B ν 𝒴 f (x + lineAt θ τ X t)) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergDerivativeDiagonalHalfEqSchrodingerDerStatement

theorem **_root_.GlobalTheory.T.heisenbergDerivative_diagonalHalf_eq_schrodingerDerivative** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ f, IsThetaFunO ψ (dblSp A) B f → ∀ (τ : D.what₂.Tau) (X : Matrix (Fin B.rank) (Fin A.rank) ℂ) (x : Mx D B.rank A.rank), sch ψ A B ν 𝒴 (Formalism.IndefR.Datum.dRho (heisO ψ (dblSp A) B) θ τ (X * (dblDec A).qX.map (tauE D τ)) f) x = deriv (fun t : ℝ => sch ψ A B ν 𝒴 f (x + lineAt θ τ X t)) 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergFixAndKillThetaOffSupportMove

theorem **map_infPart_eq_zero** {D : D7} {S T : HSp D} {b : AE D} (hb : infPart D b = 0) {y : WO S T} (hy : IsHalf b y) : y.map (infPart D) = 0

theorem **sep** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D), IsStdChar D ψ (-1) → ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ (Z : Finset (Pl D.host.E)) (e : AE D), IsIdemSum e Z → ∀ f₀ Φ f : WO S T → ℂ, IsSBHalf e (latO S T ΛS ΛT fun _ => True) Φ → IsThetaAt ψ (symplO S T) (heisO ψ S T) e

(latO S T ΛS ΛT fun _ => True) 1 f₀ Φ f → ∀ g : GA D S, (∀ x, Φ x ≠ 0 → Φ (actO S T g 1 x) = 0) → ∃ (n : ℕ) (a : Fin n → ℂ) (y : Fin n → WO S T), (∀ i, (y i).map (infPart D) = 0 ∧ (actO S T g⁻¹ 1 (-(y i))).map (infPart D) = 0) ∧ ∑ i, a i • heisO ψ S T (y i) f = f ∧ ∑ i, conj (a i) • heisO ψ S T (actO S T g⁻¹ 1 (-(y i))) f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergFixAndKillThetaOffSupportMoveStatement

theorem **_root_.GlobalTheory.T.heisenberg_fix_and_kill_theta_off_support_move** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D), IsStdChar D ψ (-1) → ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → ∀ (Z : Finset (Pl D.host.E)) (e : AE D), IsIdemSum e Z → ∀ f₀ Φ f : WO S T → ℂ, IsSBHalf e (latO S T ΛS ΛT fun _ => True) Φ → IsThetaAt ψ (symplO S T) (heisO ψ S T) e (latO S T ΛS ΛT fun _ => True) 1 f₀ Φ f → ∀ g : GA D S, (∀ x, Φ x ≠ 0 → Φ (actO S T g 1 x) = 0) → ∃ (n : ℕ) (a : Fin n → ℂ) (y : Fin n → WO S T), (∀ i, (y i).map (infPart D) = 0 ∧ (actO S T g⁻¹ 1 (-(y i))).map (infPart D) = 0) ∧ ∑ i, a i • heisO ψ S T (y i) f = f ∧ ∑ i, conj (a i) • heisO ψ S T (actO S T g⁻¹ 1 (-(y i))) f = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergFixesLatticeVacuumPeriodLattice

theorem **AA_eq** (hc : D.host.ccK = ⊤) : D.host.AA = D.host.AA0

theorem **hRat_one** (S T : HSp D) (hS : S.gram = 1) (hT : T.gram = 1) (x y : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : GlobalTheory.LatVac.hRat S T x y = ∑ c, ∑ a, star (y c a) * x c a

theorem **toMatrix_smul_eq** (S T : HSp D) (hS : S.gram = 1) {n : ℕ} (s s' : Fin n → Fin S.rank → D.host.E) (hd : ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * S.gram k j = if a = j then 1 else 0) (r : D.host.E) (x : (Fin S.rank → D.host.E) →ₗ[D.host.E] (Fin T.rank → D.host.E)) : LinearMap.toMatrix' (r • x) = ∑ i, GlobalTheory.LatVac.genMap S T r (s' i) (x (s i))

theorem **mem_latRat** (S T : HSp D) (hS : S.gram = 1) (hc : D.host.ccK = ⊤) (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (hN : IsSelfDualC S N) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (hγ : Matrix.toLin' γ ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') : γ ∈ GlobalTheory.LatVac.latRat S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E))

theorem **exists_trace_one_xi0_mul_star_eq_of_ccK_eq_top** (hc : D.host.ccK = ⊤) : ∃ w0 : 𝒪 D.host.E, (w0 : D.host.E) + star (w0 : D.host.E) = 1 ∧ D.host.xi0 * star (w0 : D.host.E) = (D.host.xi0 + 1) / 2

theorem **sum_conj_eq_trace** (D : D7) (e : D.host.E) : ∑ t, (tauE D t e + starRingEnd ℂ (tauE D t e)) = ((Algebra.trace ℚ D.host.E e : ℚ) : ℂ)

theorem **sum_two_re_eq_trace** (D : D7) (e : D.host.E) : ∑ t, (((2 * (tauE D t e).re : ℝ)) : ℂ) = ((Algebra.trace ℚ D.host.E e : ℚ) : ℂ)

theorem **tauE_star'** (τ : D.what₂.Tau) (x : D.host.E) : tauE D τ (star x) = starRingEnd ℂ (tauE D τ x)

theorem **sgnW_finTens** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} (w : D.host.E) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : sgnW ψ (incA D w) (finTens 1 γ : WO S T) = Complex.exp (Real.pi * Complex.I * ((Algebra.trace ℚ D.host.E (D.host.xi0 * hRat S T γ (w • γ)) : ℚ) : ℂ))

theorem **tauE_X** (S T : HSp D) (hS1 : S.gram = 1) (hT1 : T.gram = 1) (hxi : ∀ τ, tauE D τ D.host.xi0 = Complex.I * ((Real.sqrt D.host.d)⁻¹ : ℝ)) (w : D.host.E) (hw : D.host.xi0 * star w = (D.host.xi0 + 1) / 2) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (τ : D.what₂.Tau) : 2 * (tauE D τ (D.host.xi0 * hRat S T γ (w • γ))).re = ∑ c, ∑ a, Complex.normSq (tauE D τ (γ c a))

theorem **alV_eq** (S T : HSp D) (hS1 : S.gram = 1) (hT1 : T.gram = 1) (δ : ℂ) (sc : ℝ) (hsc : sc * ‖δ‖ ^ 2 = 2 * (Real.sqrt D.host.d)⁻¹) (hxi : ∀ τ, tauE D τ D.host.xi0 = Complex.I * ((Real.sqrt D.host.d)⁻¹ : ℝ)) (w : D.host.E) (hw : D.host.xi0 * star w = (D.host.xi0 + 1) / 2) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : alV D.host.d sc (embH (tauE D) δ (Matrix.toLin' γ)) = Complex.exp (Real.pi * Complex.I * ((Algebra.trace ℚ D.host.E (D.host.xi0 * hRat S T γ (w • γ)) : ℚ) : ℂ))

theorem **latSign** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)), Moves S hS g N → Moves T hT h N' → ∀ f, IsThetaFunO ψ S T f → IsLatVac (heisO ψ S T) (latO S T (latFr S hS N) (latFr T hT N') fun _ => True) f → ∀ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, toHom S T hS hT γ ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → ∀ v, GlobalTheory.AdTh.IsFinOf S T γ v → ∀ w, alV D.host.d sc (embH (tauE D) δ (toHom S T hS hT γ)) * heisO ψ S T v f w = f w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergFixesLatticeVacuumPeriodLatticeStatement

theorem **_root_.GlobalTheory.T.heisenberg_fixes_latticeVacuum_periodLattice** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)), Moves S hS g N → Moves T hT h N' → ∀ f, IsThetaFunO ψ S T f → IsLatVac (heisO ψ S T) (latO S T (latFr S hS N) (latFr T hT N') fun _ => True) f → ∀ γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E, toHom S T hS hT γ ∈ perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → ∀ v, GlobalTheory.AdTh.IsFinOf S T γ v → ∀ w, alV D.host.d sc (embH (tauE D) δ (toHom S T hS hT γ)) * heisO ψ S T v f w = f w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergFrameHalfInverse

theorem **conj_conj** (z : AE D) : cA D (cA D z) = z

theorem **e_mul_ι** (p : Place D k) (x : k) : p.e * p.ι x = p.ι x

theorem **conj_ι_mul_ι** (p : Place D k) (u v : k) : cA D (p.ι u) * p.ι v = 0

theorem **up_half** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x : Matrix Y Y' k) : IsHalf p.e (p.up FS FT x)

theorem **down_up** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x : Matrix Y Y' k) : p.down FS FT (p.up FS FT x) = x

theorem **up_down** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (z : WO S T) (hz : IsHalf p.e z) : p.up FS FT (p.down FS FT z) = z

theorem **hO_up_up** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x y : Matrix Y Y' k) : hO S T (p.up FS FT x) (p.up FS FT y) = 0

theorem **e_mul_hO_up** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (t : Matrix Y Y' k) (y : WO S T) : p.e * hO S T (p.up FS FT t) y = hO S T (p.up FS FT t) y

theorem **psi_sympl_up** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle) (hχ : p.IsChar ψ ψk) (t : Matrix Y Y' k) (y : WO S T) : ψ (symplO S T (p.up FS FT t) y) = ψk (p.π (hO S T (p.up FS FT t) y))

theorem **sum_four_eq_trace_mul** {m n : Type} [Fintype m] [Fintype n] (A : Matrix m m k) (N : Matrix m n k) (B : Matrix n n k) (X : Matrix n m k) : ∑ a, ∑ b, ∑ c, ∑ d, A a b * N b c * B c d * X d a = Matrix.trace (A * N * B * X)

theorem **pi_hO_up** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (t : Matrix Y Y' k) (N : Matrix (Fin S.rank) (Fin T.rank) k) : p.π (hO S T (p.up FS FT t) (N.transpose.map fun a => cA D (p.ι a))) = Matrix.trace ((S.gram⁻¹).map (p.π.comp (incA D)) * N * T.gram.map (p.π.comp (incA D)) * (FT.β * t * FS.γ))

theorem **exists_pairing** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (b : Matrix Y Y' k) : ∃ y : WO S T, IsHalf (cA D p.e) y ∧ ∀ t, p.π (hO S T (p.up FS FT t) y) = ∑ r, ∑ q, b r q * t r q

theorem **halves_at_split** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → (∀ x, IsHalf p.e (p.up FS FT x) ∧ p.down FS FT (p.up FS FT x) = x) ∧ (∀ z : WO S T, IsHalf p.e z → p.up FS FT (p.down FS FT z) = z) ∧ (∀ x y, symplO S T (p.up FS FT x) (p.up FS FT y) = 0) ∧ (∀ t (y : WO S T), IsHalf (cA D p.e) y → ψ (symplO S T (p.up FS FT t) y) = ψk (p.π (hO S T (p.up FS FT t) y))) ∧ ∀ b : Matrix Y Y' k, ∃ y : WO S T, IsHalf (cA D p.e) y ∧ ∀ t, p.π (hO S T (p.up FS FT t) y) = ∑ r, ∑ q, b r q * t r q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergFrameHalfInverseStatement

theorem **_root_.GlobalTheory.T.heisenberg_frame_half_inverse** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → (∀ x, IsHalf p.e (p.up FS FT x) ∧ p.down FS FT (p.up FS FT x) = x) ∧ (∀ z : WO S T, IsHalf p.e z → p.up FS FT (p.down FS FT z) = z) ∧ (∀ x y, symplO S T (p.up FS FT x) (p.up FS FT y) = 0) ∧ (∀ t (y : WO S T), IsHalf (cA D p.e) y → ψ (symplO S T (p.up FS FT t) y) = ψk (p.π (hO S T (p.up FS FT t) y))) ∧ ∀ b : Matrix Y Y' k, ∃ y : WO S T, IsHalf (cA D p.e) y ∧ ∀ t, p.π (hO S T (p.up FS FT t) y) = ∑ r, ∑ q, b r q * t r q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergIntegralFourierBesselBasePointEqPairin

theorem **ell_vecMul_last** {σ n : ℕ} (b : Mx D (σ + 1) n) : ell D (σ + 1) ᵥ* b = b (Fin.last σ)

theorem **zEll_mul_eq** {σ n : ℕ} (L : HSp D) (ζ : Fin L.rank → D.host.E) (b : Mx D (σ + 1) n) : zEll L (σ + 1) ζ · b = vecMulVec (ratV ζ) (b (Fin.last σ))

theorem **continuous_splitLine** {σ n : ℕ} (L : HSp D) (ζ : Fin L.rank → D.host.E) (i₀ : Fin L.rank) (h1 : ∀ i : Fin L.rank, i = i₀) (h0 : ζ i₀ ≠ 0) : Continuous (splitLine (σ

theorem **fourierBesselBasePoint_mul_eq** {σ : ℕ} (B : BesLev D σ) (S : HSp D) (b : Mx D (σ + 1) S.rank) : x0 B · b = B.d.incX S (Matrix.of fun i j => b (Fin.castSucc i) j) + B.d.inc0 S ((B.o.resT S).symm (0, zl B · b))

theorem **heisSymplecticHalf_fourierBesselBasePoint_mul** {σ : ℕ} (B : BesLev D σ) (S : HSp D) (b b' : Mx D (σ + 1) S.rank) : symplOH S B.V (x0 B · b) (x0 B · b') = symplOH S B.L (zl B · b) (zl B · b')

theorem **zl_mem_ratMx** {σ : ℕ} (B : BesLev D σ) : zl B ∈ ratMx D B.L.rank (σ + 1)

theorem **fourierBesselBasePoint_mul_mem_rationalAdelicMatrix** {σ : ℕ} (B : BesLev D σ) (S : HSp D) {γ : Mx D (σ + 1) S.rank} (hγ : γ ∈ ratMx D (σ + 1) S.rank) : x0 B · γ ∈ ratMx D B.V.rank S.rank

theorem **splitLine_symm_mem** {σ n : ℕ} (L : HSp D) (ζ : Fin L.rank → D.host.E) (i₀ : Fin L.rank) (h1 : ∀ i : Fin L.rank, i = i₀) (h0 : ζ i₀ ≠ 0) {p : Mx D σ n × Mx D L.rank n} (hp : p ∈ (ratMx D σ n).prod (ratMx D L.rank n)) : (splitLine L ζ i₀ h1 h0).symm p ∈ ratMx D (σ + 1) n

theorem **splitLine_mem** {σ n : ℕ} (L : HSp D) (ζ : Fin L.rank → D.host.E) (i₀ : Fin L.rank) (h1 : ∀ i : Fin L.rank, i = i₀) (h0 : ζ i₀ ≠ 0) {b : Mx D (σ + 1) n} (hb : b ∈ ratMx D (σ + 1) n) : splitLine L ζ i₀ h1 h0 b ∈ (ratMx D σ n).prod (ratMx D L.rank n)

theorem **heis_base** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : HSp D) (B : BesLev D σ) (vB : VolM D (σ + 1) S.rank) (vT : VolM D σ S.rank) (vL : VolM D B.L.rank S.rank) (f : WO S B.V → ℂ) (y : WO S B.L → ℂ), Quasi ψ S B.V f → Quasi ψ S B.L y → (∫ b in vB.𝓕, f (x0 B · b) * star (y (zl B · b)) ∂vB.μ = contractT B.o S vL.μ vL.𝓕 (B.d.mixT S ψ vT.μ vT.𝓕 f 0) y 0) ∧ ∫⁻ v in vL.𝓕, ‖B.d.mixT S ψ vT.μ vT.𝓕 f 0 ((B.o.resT S).symm (0, v))‖ₑ ∂vL.μ ≤ ∫⁻ b in vB.𝓕, ‖f (x0 B · b)‖ₑ ∂vB.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergIntegralFourierBesselBasePointEqPairinStatement

theorem **_root_.GlobalTheory.T.heisenbergIntegral_fourierBesselBasePoint_eq_pairing** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : HSp D) (B : BesLev D σ) (vB : VolM D (σ + 1) S.rank) (vT : VolM D σ S.rank) (vL : VolM D B.L.rank S.rank) (f : WO S B.V → ℂ) (y : WO S B.L → ℂ), Quasi ψ S B.V f → Quasi ψ S B.L y → (∫ b in vB.𝓕, f (x0 B · b) * star (y (zl B · b)) ∂vB.μ = contractT B.o S vL.μ vL.𝓕 (B.d.mixT S ψ vT.μ vT.𝓕 f 0) y 0) ∧ ∫⁻ v in vL.𝓕, ‖B.d.mixT S ψ vT.μ vT.𝓕 f 0 ((B.o.resT S).symm (0, v))‖ₑ ∂vL.μ ≤ ∫⁻ b in vB.𝓕, ‖f (x0 B · b)‖ₑ ∂vB.μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergInvariantFactorsContraction

theorem **coe_psi_mul_neg** {k : Type} [AddGroup k] (ψ : AddChar k Circle) (a : k) : (ψ a : ℂ) * ψ (-a) = 1

theorem **coe_psi_add** {k : Type} [AddGroup k] (ψ : AddChar k Circle) (a b : k) : (ψ (a + b) : ℂ) = ψ a * ψ b

theorem **contrFactor** : ContrFactor

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergInvariantFactorsContractionStatement

theorem **_root_.GlobalTheory.T.heisenbergInvariant_factors_contraction** : GlobalTheory.LocFlat.ContrFactor

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergIsotropicFixesVacuum

theorem **two_ne_zero_at** (w : Pl D) : (2 : Fv D.host.F w) ≠ 0

theorem **two_inv_int** (w : Pl D) (hw : (2 : 𝒪 D.host.F) ∉ w.asIdeal) : (2⁻¹ : Fv D.host.F w) ∈ w.adicCompletionIntegers D.host.F

theorem **finite_two** (P : Pl D → Prop) : {v : Pl D | P v ∧ (2 : 𝒪 D.host.F) ∈ v.asIdeal}.Finite

theorem **evalV_halfOff** (P : Pl D → Prop) (w : Pl D) : evalV D.host.F w (halfOff P) = if P w ∧ (2 : 𝒪 D.host.F) ∉ w.asIdeal then (2⁻¹ : Fv D.host.F w) else 0

theorem **halfOff_int** (P : Pl D → Prop) : halfOff P ∈ intAf D.host.F

theorem **pair_add** (a b : Af D.host.F) : pairA D.host.F 0 (a + b) = pairA D.host.F 0 a + pairA D.host.F 0 b

theorem **sum_pair** {ι : Type} (s : Finset ι) (α : ι → Af D.host.F) : ∑ i ∈ s, pairA D.host.F 0 (α i) = pairA D.host.F 0 (∑ i ∈ s, α i)

theorem **indicatorIdempotent_eq_add_sum** (P : Pl D → Prop) (Qs : Finset (Pl D)) (hQ : ∀ v, v ∈ Qs ↔ P v ∧ (2 : 𝒪 D.host.F) ∈ v.asIdeal) (ε : Pl D → AF D.host.F) (hε : ∀ v ∈ Qs, IsPlaceIdem D v (ε v)) : GlobalTheory.SelfDualAt.cP P = pairA D.host.F 0 (halfOff P + halfOff P) + ∑ v ∈ Qs, ε v

theorem **int_alg_pair** {β : Af D.host.F} (hβ : β ∈ intAf D.host.F) : finPart D (algebraMap (AF D.host.F) (AE D) (pairA D.host.F 0 β)) ∈ intAE D

theorem **fix_add** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {f : WO S T → ℂ} {z : WO S T} (hiso : ∀ c c' : AE D, symplOH S T (c • z) (c' • z) = 0) {a b : AE D} (ha : heisO ψ S T (a • z) f = f) (hb : heisO ψ S T (b • z) f = f) : heisO ψ S T ((a + b) • z) f = f

theorem **GlobalTheory.FinPl.FlatFix.fix_of_isotropic** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (LS : Lat S) (LT : Lat T) (P : Pl D → Prop), IsSelfDualC S LS → IsSelfDualC T LT → (∀ v, P v → Tame D v) → ∀ f : WO S T → ℂ, IsLatVac (heisO ψ S T) (latO S T (LS : Set _) (LT : Set _) P) f → ∀ z ∈ latO S T (LS : Set _) (LT : Set _) P, (∀ c c' : AE D, symplOH S T (c • z) (c' • z) = 0) → heisO ψ S T z f = f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOneParamDerivOrthDecJoinedEqSum

theorem **dLine_apply'** {S T : HSp D} {θ : GlobalTheory.IndefiniteRowSideFullHecke.FockOrthogonalDecomposition.InfFam D} {σ : (D.what₂.Tau → ℂ) →+ AE D} (h : Chart θ σ) {lam : AF D.host.F →+ ℂ} (hl : Continuous lam) (hlam : ∀ a, sndA D.host.F a = 0 → ψ a = Complex.exp (lam a)) {L : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (w : WO S T) : dLine ψ S T L f w = phase lam S T (L 1) w ∗ f w + fderiv ℝ (fun x => f (w + iotaW σ T S x)) 0 (piW θ T S (L 1))

theorem **dLine_add_line** (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {L L' : ℝ →+ WO S T} (hL : Continuous L) (hL0 : ∀ t, (L t).map (finPart D) = 0) (hL' : Continuous L') (hL0' : ∀ t, (L' t).map (finPart D) = 0) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : dLine ψ S T (L + L') f = dLine ψ S T L f + dLine ψ S T L' f

theorem **inl_fst_add_inr_snd** (x : WO W S) : inlH d S (fstH d S x) + inrH d S (sndH d S x) = x

theorem **dLine_inlH** (G : WO A S → WO S A' → ℂ) (M : ℝ →+ WO A S) : dLine ψ W S ((inlH d S).comp M) (join d S G) = join d S (fun u v => dLine ψ A S M (fun u' => G u' v) u)

theorem **dLine_inrH** (G : WO A S → WO S A' → ℂ) (M : ℝ →+ WO S A') : dLine ψ W S ((inrH d S).comp M) (join d S G) = join d S (fun u v => dLine ψ S A' M (G u) v)

theorem **dLine_join** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (W A S A' : HSp D) (d : OrthDec W A (GlobalTheory.ChiGrid.negSp A')) (G : WO A S → WO S A' → ℂ), IsBiTheta ψ A S A' G → ∀ M : ℝ →+ WO W S, Continuous M → (∀ s, (M s).map (finPart D) = 0) → (∀ s, (GlobalTheory.BiTransfer.fstH d S (M s)).map (finPart D) = 0) ∧ (∀ s, (GlobalTheory.BiTransfer.sndH d S (M s)).map (finPart D) = 0) ∧ GlobalTheory.WeilVacMove.dLine ψ W S M (GlobalTheory.BiTransfer.join d S G) = GlobalTheory.BiTransfer.join d S fun u v => GlobalTheory.WeilVacMove.dLine ψ A S ((GlobalTheory.BiTransfer.fstH d S).comp M) (fun u' => G u' v) u + GlobalTheory.WeilVacMove.dLine ψ S A' ((GlobalTheory.BiTransfer.sndH d S).comp M) (G u) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOneParamDerivOrthDecJoinedEqSumStatement

theorem **_root_.GlobalTheory.T.heisenbergOneParamDeriv_orthDecJoined_eq_sum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (W A S A' : HSp D) (d : OrthDec W A (GlobalTheory.ChiGrid.negSp A')) (G : WO A S → WO S A' → ℂ), IsBiTheta ψ A S A' G → ∀ M : ℝ →+ WO W S, Continuous M → (∀ s, (M s).map (finPart D) = 0) → (∀ s, (GlobalTheory.BiTransfer.fstH d S (M s)).map (finPart D) = 0) ∧ (∀ s, (GlobalTheory.BiTransfer.sndH d S (M s)).map (finPart D) = 0) ∧ GlobalTheory.WeilVacMove.dLine ψ W S M (GlobalTheory.BiTransfer.join d S G) = GlobalTheory.BiTransfer.join d S fun u v => GlobalTheory.WeilVacMove.dLine ψ A S ((GlobalTheory.BiTransfer.fstH d S).comp M) (fun u' => G u' v) u + GlobalTheory.WeilVacMove.dLine ψ S A' ((GlobalTheory.BiTransfer.sndH d S).comp M) (G u) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorConjugateHalfVacuumEqSemichara

theorem **mem_localization_of_mem_halfBO** {E : Type} [Field E] [NumberField E] [StarRing E] {a b : ℕ} (SS : Finset (HeightOneSpectrum (𝒪 E))) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)) (c : ℕ) (y : Formalism.L2.Col.HomE E a b) (hy : y ∈ halfBO SS M c) (Q : HeightOneSpectrum (𝒪 E)) (hQ : ¬ (Q ∈ SS ∨ Formalism.L2.Col.conjPrime₂ Q ∈ SS)) : y ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M

theorem **off_part_mem** (D : D7) (S T : HSp D) (Λ : AddSubgroup (WO S T)) (hstab : ∀ w : AE D, finPart D w ∈ intAE D → ∀ u ∈ Λ, w • u ∈ Λ) (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)) (he : IsIdemSum e SS) (c : ℕ) (hc : 0 < c) (d : 𝒪 D.host.E) (hd0 : d ≠ 0) (hdP : ∀ Q : HeightOneSpectrum (𝒪 D.host.E), (c : 𝒪 D.host.E) ∈ Q.asIdeal → d ∈ Q.asIdeal → Q ∈ SS ∨ Formalism.L2.Col.conjPrime₂ Q ∈ SS) (Y : WO S T) (hcY : incA D (c : D.host.E) • Y ∈ Λ) (hdY : incA D (d : D.host.E) • Y ∈ Λ) : (1 − e − cA D e) • Y ∈ Λ

theorem **heisO_conjHalf_eq** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS ∗ D.host.q ∗ D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N → (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N' → ∀ f₀, IsVacO ψ θ S T (fun _ => 1) (fun _ => 1) (latFr S hS N) (latFr T hT N') f₀ → ∀ y' ∈ halfBO

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorConjugateHalfVacuumEqSemicharaStatement

theorem **_root_.GlobalTheory.T.heisenbergOperator_conjugateHalf_vacuum_eq_semicharacter** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS ∗ D.host.q ∗ D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N → (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N' → ∀ f₀, IsVacO ψ θ S T (fun _ => 1) (f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorDisjointSupportSmulAdd

theorem **cA_smul** (c : AF D.host.F) (z : AE D) : cA D (c • z) = c • cA D z

theorem **hO_add_left** (S T : HSp D) (x x' y : WO S T) : hO S T (x + x') y = hO S T x y + hO S T x' y

theorem **hO_add_right** (S T : HSp D) (x y y' : WO S T) : hO S T x (y + y') = hO S T x y + hO S T x y'

theorem **hO_smul_smul** (S T : HSp D) (c c' : AF D.host.F) (x y : WO S T) : hO S T (c • x) (c' • y) = (c ∗ c') • hO S T x y

theorem **symplOH_add_left** (S T : HSp D) (x x' y : WO S T) : symplOH S T (x + x') y = symplOH S T x y + symplOH S T x' y

theorem **symplOH_add_right** (S T : HSp D) (x y y' : WO S T) : symplOH S T x (y + y') = symplOH S T x y + symplOH S T x y'

theorem **symplOH_smul_smul_eq_zero** (S T : HSp D) (c c' : AF D.host.F) (x y : WO S T) (h : c * c' = 0) : symplOH S T (c • x) (c' • y) = 0

theorem **heisO_smul_add_smul** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c c' : AF D.host.F) (x y : WO S T) (f : WO S T → ℂ), c * c' = 0 → heisO ψ S T (c • x + c' • y) f = heisO ψ S T (c • x) (heisO ψ S T (c' • y) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorDisjointSupportSmulAddStatement

theorem **_root_.GlobalTheory.T.heisenbergOperator_disjointSupport_smul_add** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c c' : AF D.host.F) (x y : WO S T) (f : WO S T → ℂ), c * c' = 0 → heisO ψ S T (c • x + c' • y) f = heisO ψ S T (c • x) (heisO ψ S T (c' • y) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorFinitePartRestActsBySemicharac

theorem **natsmul_mem_twO'** (L : Submodule (𝒪 E) W) (c : ℕ) (e e' : HeightOneSpectrum (𝒪 E) → ℤ) (he : ∀ Q, e' Q ≤ e Q + 1) {x : W} (hx : x ∈ Formalism.L2.Ker.twist₂ L c e) : (c : 𝒪 E) • x ∈ Formalism.L2.Ker.twist₂ L c e'

theorem **mem_twO_zero'** (M : Submodule (𝒪 E) W) (c : ℕ) (x : W) : x ∈ Formalism.L2.Ker.twist₂ M c (fun _ => 0) ↔ x ∈ M

theorem **heisO_rest_eq** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ f₀, IsVacO ψ θ S T (fun _ => 1) (fun _ => 1) (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) f₀ → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, heisO ψ S T ((GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) - ((LinearMap.toMatrix' x').map fun r => e * incA D r)) f₀ = alV D.host.d sc (embH (tauE D) δ x') • f₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorFinitePartRestActsBySemicharacStatement

theorem **_root_.GlobalTheory.T.heisenbergOperator_finitePartRest_acts_by_semicharacter** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ (c : ℕ), 0 < c → Odd c → ∀ f₀, IsVacO ψ θ S T (fun _ => 1) (fun _ => 1) (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) f₀ → ∀ x' ∈ halfAO SS (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N') c, heisO ψ S T ((GlobalTheory.VacUnique.finRat S T (LinearMap.toMatrix' x')) - ((LinearMap.toMatrix' x').map fun r => e * incA D r)) f₀ = alV D.host.d sc (embH (tauE D) δ x')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorHalfComponentOnVacuumTranslate

theorem **GlobalTheory.AdForm.HalfHeis.heisO_halfA_eq** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N → (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N' → ∀ f₀, IsVacO ψ θ S T (fun _ => 1) (fun _ => 1) (latFr S hS N) (latFr T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorHalfComponentOnVacuumTranslateStatement

theorem **_root_.GlobalTheory.T.heisenbergOperator_halfComponent_on_vacuumTranslates** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin a → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin b → D.host.E)) (c : ℕ), 0 < c → Odd c → (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → LaFacts (tauE D) δ D.host.d sc (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N' → ∀ (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S ΛS → IsSelfDualC T ΛT → (ΛS : Set (Fin S.rank → D.host.E)) = latFr S hS N → (ΛT : Set (Fin T.rank → D.host.E)) = latFr T hT N' → ∀ f₀, IsVacO ψ θ S T (fun _ => 1) (fun _

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorThetaFunctionalMixedSourceMode

theorem **heis_mixS** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → (∀ x a, d.mixS T ψ v.μ v.𝓕 (heisO ψ S T (d.extX T a) f) x = d.mixS T ψ v.μ v.𝓕 f (x + a)) ∧ (∀ x a, d.mixS T ψ v.μ v.𝓕 (heisO ψ S T (d.extY T a) f) x = ψ (symplO S T (d.extX T x) (d.extY T a)) • d.mixS T ψ v.μ v.𝓕 f x) ∧ (∀ x w, d.mixS T ψ v.μ v.𝓕 (heisO ψ S T (d.ext0 T w) f) x = heisO ψ S₀ T w (d.mixS T ψ v.μ v.𝓕 f x)) ∧ (Continuous fun p : Mx D T.rank ρ × WO S₀ T => d.mixS T ψ v.μ v.𝓕 f p.1 p.2) ∧ ∀ w, HasSum (fun x : Matrix (Fin T.rank) (Fin ρ) D.host.E => d.mixS T ψ v.μ v.𝓕 f (rat x) w) (f (d.ext0 T w))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorThetaFunctionalMixedSourceModeStatement

theorem **_root_.GlobalTheory.T.heisenbergOperator_thetaFunctional_mixedSourceModel** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (v : VolM D T.rank ρ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → (∀ x a, d.mixS T ψ v.μ v.𝓕 (heisO ψ S T (d.extX T a) f) x = d.mixS T ψ v.μ v.𝓕 f (x + a)) ∧ (∀ x a, d.mixS T ψ v.μ v.𝓕 (heisO ψ S T (d.extY T a) f) x = ψ (symplO S T (d.extX T x) (d.extY T a)) • d.mixS T ψ v.μ v.𝓕 f x) ∧ (∀ x w, d.mixS T ψ v.μ v.𝓕 (heisO ψ S T (d.ext0 T w) f) x = heisO ψ S₀ T w (d.mixS T ψ v.μ v.𝓕 f x)) ∧ (Continuous fun p : Mx D T.rank ρ × WO S₀ T => d.mixS T ψ v.μ v.𝓕 f p.1 p.2) ∧ ∀ w, HasSum (fun x : Matrix (Fin T.rank) (Fin ρ) D.host.E => d.mixS T ψ v.μ v.𝓕 f (rat x) w) (f (d.ext0 T w))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorThetaLawTendstoUniformNearZero

theorem **heisO_near_zero_unif** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T : HSp D) (q : WO A T → ℂ), Continuous q → (∀ (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) (u : WO A T), q (ratW T A γ + u) = ψ (-(symplOH A T (ratW T A γ) u)) * q u) → ∀ ε : ℝ, 0 < ε → ∃ N ∈ 𝒩 (0 : WO A T), ∀ a ∈ N, ∀ w, ‖heisO ψ A T a q w - q w‖ ≤ ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergOperatorThetaLawTendstoUniformNearZeroStatement

theorem **_root_.GlobalTheory.T.heisenbergOperator_thetaLaw_tendsto_uniform_near_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T : HSp D) (q : WO A T → ℂ), Continuous q → (∀ (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) (u : WO A T),

q (ratW T A γ + u) = ψ (-(symplOH A T (ratW T A γ) u)) * q u) → ∀ ε : ℝ, 0 < ε → ∃ N ∈ 𝒩 (0 : WO A T), ∀ a ∈ N, ∀ w, ‖heisO ψ A T a q w - q w‖ ≤ ε

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergSupSobolevSeminormThetaLaws

theorem **supSob_laws** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D), (∀ k (f : WO S T → ℂ), IsThetaFunO ψ S T f → supSob ψ θ k f ≠ ⊤) ∧ (∀ (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ), (∀ U : AddSubgroup (WO S T), ∀ f ∈ GlobalTheory.Size.thetaLevel ψ S T U, dRho (heisO ψ S T) θ τ z f ∈ GlobalTheory.Size.thetaLevel ψ S T U) ∧ ∃ C : ℝ≥0, ∀ k (f : WO S T → ℂ), IsThetaFunO ψ S T f → supSob ψ θ k (dRho (heisO ψ S T) θ τ z f) ≤ C * supSob ψ θ (k + 1) f) ∧ ∀ k (f g : WO S T → ℂ) (c : ℂ), IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → supSob ψ θ k (f + g) ≤ supSob ψ θ k f + supSob ψ θ k g ∧ supSob ψ θ k (c • f) = ‖c‖ₑ * supSob ψ θ k f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergSupSobolevSeminormThetaLawsStatement

theorem **_root_.GlobalTheory.T.heisenberg_sup_sobolev_seminorm_theta_laws** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D), (∀ k (f : WO S T → ℂ), IsThetaFunO ψ S T f → supSob ψ θ k f ≠ ⊤) ∧ (∀ (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ), (∀ U : AddSubgroup (WO S T), ∀ f ∈ GlobalTheory.Size.thetaLevel ψ S T U, dRho (heisO ψ S T) θ τ z f ∈ GlobalTheory.Size.thetaLevel ψ S T U) ∧ ∃ C : ℝ≥0, ∀ k (f : WO S T → ℂ), IsThetaFunO ψ S T f → supSob ψ θ k (dRho (heisO ψ S T) θ τ z f) ≤ C * supSob ψ θ (k + 1) f) ∧ ∀ k (f g : WO S T → ℂ) (c : ℂ), IsThetaFunO ψ S T f → IsThetaFunO ψ S T g → supSob ψ θ k (f + g) ≤ supSob ψ θ k f + supSob ψ θ k g ∧ supSob ψ θ k (c • f) = ‖c‖ₑ * supSob ψ θ k f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergThetaMixedModelTargetOfMixedModel

theorem **symplOH_skew'** (S T : HSp D) (x y : WO S T) : symplOH S T y x = -symplOH S T x y

theorem **symplO_eq_two** (S T : HSp D) (x y : WO S T) : symplO S T x y = symplOH S T x y + symplOH S T x y

theorem **mixT_heis_incX** {μ : Measure (Mx D ρ S.rank)} {ℱ : Set (Mx D ρ S.rank)} (hμ : IsCovolOne (ratMx D ρ S.rank) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (ξ : Mx D ρ S.rank) (y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (heisO ψ S T (d.incX S ξ) f) y = ψ (symplO S T (d.incAdj S y) (d.incX S ξ)) • d.mixT S ψ μ ℱ f y

theorem **mixedModelTarget_heisenberg_anisotropic_part** {μ : Measure (Mx D ρ S.rank)} {ℱ : Set (Mx D ρ S.rank)} (f : WO S T → ℂ) (v : WO S T₀) (y : Mx D S.rank ρ) : d.mixT S ψ μ ℱ (heisO ψ S T (d.inc0 S v) f) y = heisO ψ S T₀ v (d.mixT S ψ μ ℱ f y)

theorem **hasSum_mixT** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {μ : Measure (Mx D ρ S.rank)} {ℱ : Set (Mx D ρ S.rank)} (hμ : IsCovolOne (ratMx D ρ S.rank) μ ℱ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : HasSum (fun y : Matrix (Fin S.rank) (Fin ρ) D.host.E => d.mixT S ψ μ ℱ f (rat y) 0) (f 0)

theorem **heisenbergTarget** (D : D7) : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.HeisenbergTarget D

theorem **main** : ∀ D : D7, GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.HeisenbergTarget D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergThetaMixedModelTargetOfMixedModelStatement

theorem **_root_.GlobalTheory.T.heisenbergThetaMixedModelTarget_of_mixedModel** : ∀ D : D7, GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.HeisenbergTarget D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergTranslateThetaContinuousDifferentiable

theorem **enorm_le_supN** (φ : WO S T → ℂ) (w : WO S T) : ‖φ w‖ₑ ≤ supN φ

theorem **supN_add_le** (φ φ' : WO S T → ℂ) : supN (φ + φ') ≤ supN φ + supN φ'

theorem **supN_smul** (c : ℂ) (φ : WO S T → ℂ) : supN (c • φ) = ‖c‖ₑ * supN φ

theorem **supN_heisO** (hψ : IsStdChar D ψ (-1)) (u : WO S T) (φ : WO S T → ℂ) : supN (heisO ψ S T u φ) = supN φ

theorem **supN_ne_top** (hψ : IsStdChar D ψ (-1)) {φ : WO S T → ℂ} (h : IsThetaFunO ψ S T φ) : supN φ ≠ ⊤

theorem **heisO_add'** (u : WO S T) (φ φ' : WO S T → ℂ) : heisO ψ S T u (φ + φ') = heisO ψ S T u φ + heisO ψ S T u φ'

theorem **heisO_sub'** (u : WO S T) (φ φ' : WO S T → ℂ) : heisO ψ S T u (φ - φ') = heisO ψ S T u φ - heisO ψ S T u φ'

theorem **heisO_smul'** (u : WO S T) (c : ℂ) (φ : WO S T → ℂ) : heisO ψ S T u (c • φ) = c • heisO ψ S T u φ

theorem **supN_le_ofReal** {φ : WO S T → ℂ} {C : ℝ} (h : ∀ w, ‖φ w‖ ≤ C) : supN φ ≤ ENNReal.ofReal C

theorem **tendsto_supN_heisO** (hψ : IsStdChar D ψ (-1)) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Filter.Tendsto (fun u : WO S T => supN (heisO ψ S T u f - f)) (𝒩 0) (𝒩 0)

theorem **theta_rem** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (t : ℝ) : IsThetaFunO ψ S T (rem ψ L t f)

theorem **tendsto_supN_rem** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : Filter.Tendsto (fun t : ℝ => supN (rem ψ L t f)) (𝒩[≠] 0) (𝒩 0)

theorem **eq_zero_of_tendsto** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) {ι : Type} {l : Filter ι} [l.NeBot] {h : ι → WO S T → ℂ} (hh : ∀ i, IsThetaFunO ψ S T (h i)) {Λ : WO S T → ℂ} (h0 : Filter.Tendsto (fun i => supN (h i)) l (𝒩 0)) (h1 : Filter.Tendsto (fun i => supN (dLine ψ S T L (h i) - Λ)) l (𝒩 0)) : Λ = 0

theorem **continuous_cc** (hθ : IsInfFam θ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) : Continuous (cc θ τ z : WO S T → ℂ)

theorem **cc_add** (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (u v : WO S T) : cc θ τ z (u + v) = cc θ τ z u + cc θ τ z v

theorem **cc_zero** (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) : cc θ τ z (0 : WO S T) = 0

theorem **cc_line** (hθ : IsInfFam θ) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {L : ℝ →+ WO S T} (hL : Continuous L) (t : ℝ) : cc θ τ z (L t) = t * cc θ τ z (L 1)

theorem **word_smul** (l : List (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool)) (c : ℂ) (f : WO S T → ℂ) : wordAct (heisDer ψ θ) l (c • f) = c • wordAct (heisDer ψ θ) l f

theorem **word_add** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (l : List (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool)) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : wordAct (heisDer ψ θ) l (f + g) = wordAct (heisDer ψ θ) l f + wordAct (heisDer ψ θ) l g

theorem **der_sub** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (a : D.what₂.Tau × Fin T.rank × Fin S.rank × Bool) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : heisDer ψ θ a (f - g) = heisDer ψ θ a f - heisDer ψ θ a g

theorem **der_heisO** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (a : D.what₂.Tau × Fin T.rank × Fin S.rank × Bool) (u : WO S T) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : heisDer ψ θ a (heisO ψ S T u f) = heisO ψ S T u (heisDer ψ θ a f) + ccA θ a u • heisO ψ S T u f

theorem **tendsto_word_heisO** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) (l : List (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool)) : ∀ {f : WO S T → ℂ}, IsThetaFunO ψ S T f → Filter.Tendsto (fun u : WO S T => supN (wordAct (heisDer ψ θ) l (heisO ψ S T u f - f))) (𝒩 0) (𝒩 0)

theorem **der_rem** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) (a : D.what₂.Tau × Fin T.rank × Fin S.rank × Bool) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {t : ℝ} (ht : t ≠ 0) : heisDer ψ θ a (rem ψ L t f) - (ccA θ a (L 1) • f + dLine ψ S T L (heisDer ψ θ a f) - heisDer ψ θ a (dLine ψ S T L f)) = rem ψ L t (heisDer ψ θ a f) + ccA θ a (L 1) • (heisO ψ S T (L t) f - f)

theorem **tendsto_word_rem** (hψ : IsStdChar D ψ (-1)) (hθ : IsInfFam θ) {L : ℝ →+ WO S T} (hL : IsRealDir T S L) (l : List (D.what₂.Tau × Fin T.rank × Fin S.rank × Bool)) : ∀ {f : WO S T → ℂ}, IsThetaFunO ψ S T f → Filter.Tendsto (fun t : ℝ => supN (wordAct (heisDer ψ θ) l (rem ψ L t f))) (𝒩[≠] 0) (𝒩 0)

theorem **tendsto_supSob** {ι : Type} {l : Filter ι} {h : ι → WO S T → ℂ} (k : ℕ) (H : ∀ m, Filter.Tendsto (fun i => supN (wordAct (heisDer ψ θ) m (h i))) l (𝒩 0)) : Filter.Tendsto (fun i => supSob ψ θ k (h i)) l (𝒩 0)

theorem **heisO_taylor** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (k : ℕ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → Filter.Tendsto (fun u : WO S T => supSob ψ θ k (heisO ψ S T u f - f)) (nhds 0) (nhds 0) ∧ ∀ L : ℝ →+ WO S T, Continuous L → (∀ t, (L t).map (finPart D) = 0) → IsThetaFunO ψ S T (fun p => deriv (fun t : ℝ => heisO ψ S T (L t) f p) 0) ∧ Filter.Tendsto (fun t : ℝ => supSob ψ θ k ((t⁻¹ : ℂ) • (heisO ψ S T (L t) f - f) - fun p => deriv (fun t : ℝ => heisO ψ S T (L t) f p) 0)) (nhdsWithin 0 {0}ᶜ) (nhds 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HeisenbergTranslateThetaContinuousDifferentiableStatement

theorem **_root_.GlobalTheory.T.heisenbergTranslate_theta_continuous_differentiable_sobolev** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (S T : HSp D) (k : ℕ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → Filter.Tendsto (fun u : WO S T => supSob ψ θ k (heisO ψ S T u f - f)) (nhds 0) (nhds 0) ∧ ∀ L : ℝ →+ WO S T, Continuous L → (∀ t, (L t).map (finPart D) = 0) → IsThetaFunO ψ S T (fun p => deriv (fun t : ℝ => heisO ψ S T (L t) f p) 0) ∧ Filter.Tendsto (fun t : ℝ => supSob ψ θ k ((t⁻¹ : ℂ) • (heisO ψ S T (L t) f - f) - fun p => deriv (fun t : ℝ => heisO ψ S T (L t) f p) 0)) (nhdsWithin 0 {0}ᶜ) (nhds 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianAdjointExistsUniqueOfPosDef

theorem **hermitian_adjoint_exists_unique** : ∀ (A C : Type) [AddCommGroup A] [Module ℂ A] [AddCommGroup C] [Module ℂ C] [FiniteDimensional ℂ C] (ipA : A →ₗ[ℂ] A →ₗ*[ℂ] ℂ) (ip : C →ₗ[ℂ] C →ₗ*[ℂ] ℂ), (∀ x y, ip x y = conj (ip y x)) → (∀ x, x ≠ 0 → 0 < (ip x x).re) → ∀ (io : C →ₗ[ℂ] A) (u : A), ∃! x : C, ∀ y, ip y x = ipA (io y) u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianAdjointExistsUniqueOfPosDefStatement

theorem **_root_.GlobalTheory.T.hermitian_adjoint_exists_unique_of_posDef** : ∀ (A C : Type) [AddCommGroup A] [Module ℂ A] [AddCommGroup C] [Module ℂ C] [FiniteDimensional ℂ C] (ipA : A →ₗ[ℂ] A →ₗ*[ℂ] ℂ) (ip : C →ₗ[ℂ] C →ₗ*[ℂ] ℂ), (∀ x y, ip x y = conj (ip y x)) → (∀ x, x ≠ 0 → 0 < (ip x x).re) → ∀ (io : C →ₗ[ℂ] A) (u : A), ∃! x : C, ∀ y, ip y x = ipA (io y) u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianAdjointSymplecticActionThetaHeisenberg

theorem **dagger_add** (S T : HSp D) (x y : WO S T) : dagger S T (x + y) = dagger S T x + dagger S T y

theorem **continuous_dagger** (S T : HSp D) : Continuous (dagger S T)

theorem **dagger_rat** (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : dagger S T (rat γ) = rat (S.gram⁻¹ * (γᴴ * T.gram))

theorem **infPart_cA** (z : AE D) : infPart D (cA D z) = sigR (Ainf D.host.F) D.sp.cm.sig (infPart D z)

theorem **finPart_cA** (z : AE D) : finPart D (cA D z) = sigR (Af D.host.F) D.sp.cm.sig (finPart D z)

theorem **map_dagger_zero** {B : Type} [CommRing B] (p : AE D →+* B) (hp : ∀ z, p z = 0 → p (cA D z) = 0) (S T : HSp D) {x : WO S T} (hx : x.map p = 0) : (dagger S T x).map p = 0

theorem **infPart_zero** (z : AE D) (h : infPart D z = 0) : infPart D (cA D z) = 0

theorem **finPart_zero** (z : AE D) (h : finPart D z = 0) : finPart D (cA D z) = 0

theorem **hO_dagger** (S T : HSp D) (x y : WO S T) : hO T S (dagger S T x) (dagger S T y) = hO S T y x

theorem **symplO_dagger** (S T : HSp D) (x y : WO S T) : symplO T S (dagger S T x) (dagger S T y) = -symplO S T x y

theorem **symplOH_dagger** (S T : HSp D) (x y : WO S T) : symplOH T S (dagger S T x) (dagger S T y) = -symplOH S T x y

theorem **act_dagger** (S T : HSp D) (g : GA D S) (h : GA D T) (x : WO S T) : dagger S T (actO S T g h x) = actO T S h g (dagger S T x)

theorem **conj_psi** {ψ : AddChar (AF D.host.F) ℂ} (hn : ∀ a, ‖ψ a‖ = 1) (a : AF D.host.F) : starRingEnd ℂ (ψ a) = ψ (-a)

theorem **heis_dagger** (ψ : AddChar (AF D.host.F) ℂ) (hn : ∀ a, ‖ψ a‖ = 1) (S T : HSp D) (w : WO S T) (f : WO S T → ℂ) : heisO ψ T S (dagger S T w) (cj (dagger T S) f) = GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.cj (dagger T S) (heisO ψ S T w f)

theorem **theta_fwd** (ψ : AddChar (AF D.host.F) ℂ) (hn : ∀ a, ‖ψ a‖ = 1) (S T : HSp D) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ T S (cj (dagger T S) f) where

theorem **theta_iff** (ψ : AddChar (AF D.host.F) ℂ) (hn : ∀ a, ‖ψ a‖ = 1) (S T : HSp D) (f : WO S T → ℂ) : IsThetaFunO ψ T S (cj (dagger T S) f) ↔ IsThetaFunO ψ S T f

theorem **adjoint_clauses** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ S T : HSp D, (∀ x y : WO S T, symplO T S (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T x) (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T y) = -symplO S T x y) ∧ (∀ g h (x : WO S T), GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T (actO S T g h x) = actO T S h g (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T x)) ∧ (∀ f : WO S T → ℂ, IsThetaFunO ψ T S (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.cj (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger T S) f) ↔ IsThetaFunO ψ S T f) ∧ ∀ (w : WO S T) (f : WO S T → ℂ), heisO ψ T S (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T w) (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAd

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianAdjointSymplecticActionThetaHeisenbergStatement

theorem **_root_.GlobalTheory.T.hermitianAdjoint_symplectic_action_theta_heisenberg** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ S T : HSp D, (∀ x y : WO S T, symplO T S (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T x) (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T y) = -symplO S T x y) ∧ (∀ g h (x : WO S T), GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T (actO S T g h x) = actO T S h g (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S T x)) ∧ (∀ f : WO S T → ℂ, IsThetaFunO ψ T S (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.cj (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger T S) f) ↔ IsThetaFunO ψ S T f) ∧ ∀ (w : WO S T) (f : WO S T → ℂ), heisO ψ T S (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.dagger S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianCocycleIntegralOfLocal

theorem **hv_apply** (sc : ℝ) (z w : VV ι a b) : HV sc z w = (sc : ℂ) * ∑ τ, ∑ i, ∑ j, z τ i j * (starRingEnd ℂ) (w τ i j)

theorem **hv_swap** (sc : ℝ) (z w : VV ι a b) : HV sc w z = (starRingEnd ℂ) (HV sc z w)

theorem **embH_apply** (emb : ι → E →+* ℂ) (δ : ℂ) (x : HomE E a b) (τ : ι) (i : Fin b) (j : Fin a) : embH emb δ x τ i j = δ * emb τ (x (Pi.single j 1) i)

theorem **hv_smul** (emb : ι → E →+* ℂ) (δ : ℂ) (sc : ℝ) (hs : ∀ (τ : ι) (x : E), emb τ (star x) = starRingEnd ℂ (emb τ x)) (r : E) (x y : HomE E a b) : HV sc (embH emb δ (r • x)) (embH emb δ y) = HV sc (embH emb δ x) (embH emb δ (star r • y))

theorem **int_of_local** (M : Submodule (𝒪 E) (HomE E a b)) (f : HomE E a b → HomE E a b → ℝ) (hadd : ∀ x x' y, f (x + x') y = f x y + f x' y) (hsm : ∀ (r : E) x y, f (r • x) y = f x (star r • y)) (hM : ∀ x ∈ M, ∀ y ∈ M, ∃ n : ℤ, f x y = n) (x y : HomE E a b) (h : ∀ Q : HeightOneSpectrum (𝒪 E), ∃ k : E, k ≠ 0 ∧ k⁻¹ • x ∈ localization₂ Q.asIdeal M ∧ star k • y ∈ localization₂ (conjPrime₂ Q).asIdeal M) : ∃ n : ℤ, f x y = n

theorem **integral_of_local** : ∀ (E : Type) [Field E] [NumberField E] [StarRing E] (ι : Type) [Fintype ι] [Nonempty ι] (a b : ℕ) (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)), δ ≠ 0 → (∀ (τ : ι) (x : E), emb τ (star x) = starRingEnd ℂ (emb τ x)) → Formalism.L2.Th1.H.IsSelfDual (HV sc) (laV emb δ M) → Formalism.L2.Th1.H.IsSemiChar (HV sc) (laV emb δ M) (alV d sc) → ∀ x y : Formalism.L2.Col.HomE E a b, (∀ Q : HeightOneSpectrum (𝒪 E), ∃ k : E, k ≠ 0 ∧ k⁻¹ • x ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M ∧ star k • y ∈ Formalism.HermitianLatticesTheory.localization₂ (Formalism.L2.Col.conjPrime₂ Q).asIdeal M) → (∃ n : ℤ, Formalism.L2.Th1.H.altE (HV sc) (embH emb δ x) (embH emb δ y) = n) ∧ ∃ n : ℤ, (Real.sqrt d * (HV sc (embH emb δ x) (embH emb δ y)).re + Formalism.L2.Th1.H.altE (HV sc) (embH emb δ x) (embH emb δ y)) / 2 = n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianCocycleIntegralOfLocalStatement

theorem **_root_.GlobalTheory.T.hermitian_cocycle_integral_of_local** : ∀ (E : Type) [Field E] [NumberField E] [StarRing E] (ι : Type) [Fintype ι] [Nonempty ι] (a b : ℕ) (emb : ι → E →+* ℂ) (δ : ℂ) (d : ℕ) (sc : ℝ) (M : Submodule (𝒪 E) (Formalism.L2.Col.HomE E a b)), δ ≠ 0 → (∀ (τ : ι) (x : E), emb τ (star x) = starRingEnd ℂ (emb τ x)) → Formalism.L2.Th1.H.IsSelfDual (HV sc) (laV emb δ M) → Formalism.L2.Th1.H.IsSemiChar (HV sc) (laV emb δ M) (alV d sc) → ∀ x y : Formalism.L2.Col.HomE E a b, (∀ Q : HeightOneSpectrum (𝒪 E), ∃ k : E, k ≠ 0 ∧ k⁻¹ • x ∈ Formalism.HermitianLatticesTheory.localization₂ Q.asIdeal M ∧ star k • y ∈ Formalism.HermitianLatticesTheory.localization₂ (Formalism.L2.Col.conjPrime₂ Q).asIdeal M) → (∃ n : ℤ, Formalism.L2.Th1.H.altE (HV sc) (embH emb δ x) (embH emb δ y) = n) ∧ ∃ n : ℤ, (Real.sqrt d * (HV sc (embH emb δ x) (embH emb δ y)).re + Formalism.L2.Th1.H.altE (HV sc) (em

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianCoordinatesHomeomorph

theorem **sum_sum_split** {M : Type} [AddCommMonoid M] {n : ℕ} (f : Fin n → Fin n → M) : ∑ a, ∑ b, f a b = ∑ a, f a a + ∑ p : {p : Fin n × Fin n // p.1 < p.2}, (f p.1.1 p.1.2 + f p.1.2 p.1.1)

theorem **invMat_mem** (h : IsCMExt F E σ) {n : ℕ} (x : (Fin n → R) × ({p : Fin n × Fin n // p.1 < p.2} → R ⊗[F] E)) : invMat R σ x ∈ HermR R σ n

theorem **pair_eq** (h : IsCMExt F E σ) {n : ℕ} (S T : HermR R σ n) : pairHerm R σ n S T = coordPair R σ n (hermCoord R σ S) (hermCoord R σ T)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianLatticeCovolumeNonneg

theorem **hform_gramMatrix** {K : Type} [Field K] [StarRing K] {n j : ℕ} (G : Matrix (Fin n) (Fin n) K) (e : Fin j → Fin n → K) (x y : Fin j → K) : hform (Matrix.of fun a b => star (e a) ·ᵥ G *ᵥ e b) x y = hform G (∑ i, x i • e i) (∑ i, y i • e i)

theorem **lcv_nonneg_of_integral_submodule** (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (K0 : Subfield K) (hK0 : IsCMSub K0) (v0 : InfinitePlace K) (hv0 : IsConjAt v0) (n : ℕ) (G : Matrix (Fin n) (Fin n) K) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ) (hGX : IsRatHerm K0 G X) (N : Submodule (𝒪 K) (Fin n → K)) (hN : N.FG) (hint : Formalism.HermitianLatticesTheory.IsIntegralLat (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianLatticeCovolumeNonnegStatement

theorem **_root_.GlobalTheory.T.hermitian_lattice_covolume_nonneg** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (K0 : Subfield K), IsCMSub K0 → ∀ v0 : InfinitePlace K, IsConjAt v0 → ∀ (n : ℕ) (G : Matrix (Fin n) (Fin n) K) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ), IsRatHerm K0 G X → ∀ N : Submodule (𝒪 K) (Fin n → K), N.FG → Formalism.HermitianLatticesTheory.IsIntegralLat (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianLatticeMajorant

theorem **muMaxDual_nonpos_of_rat_coord** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (K0 : Subfield K), IsCMSub K0 → ∀ v0 : InfinitePlace K, IsConjAt v0 → ∀ (A : HLat K) (n : ℕ) (H : Matrix (Fin n) (Fin n) K) (e : (Fin n → K) ≃ₗ[K] A.V) (M : Submodule (𝒪 K) (Fin n → K)), A.Good → 0 < A.rank → (∀ i j, H i j ∈ K0) → (∀ (v : InfinitePlace K) (i j : Fin n), Formalism.HermitianLatticesTheory.majAt A.b A.X v (e (Pi.single i 1)) (e (Pi.single j 1)) = v.embedding (H i j)) → A.L = M.map (e.toLinearMap.restrictScalars (𝒪 K)) → Formalism.HermitianLatticesTheory.dual (hform H) M ≤ M → A.muMaxDual ≤ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianLatticeMajorantStatement

theorem **_root_.GlobalTheory.T.muMaxDual_nonpos_of_dual_le** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (K0 : Subfield K), IsCMSub K0 → ∀ v0 : InfinitePlace K, IsConjAt v0 → ∀ (A : HLat K) (n : ℕ) (H : Matrix (Fin n) (Fin n) K) (e : (Fin n → K) ≃ₗ[K] A.V) (M : Submodule (𝒪 K) (Fin n → K)), A.Good → 0 < A.rank → (∀ i j, H i j ∈ K0) → (∀ (v : InfinitePlace K) (i j : Fin n),

Formalism.HermitianLatticesTheory.majAt A.b A.X v (e (Pi.single i 1)) (e (Pi.single j 1)) = v.embedding (H i j)) → A.L = M.map (e.toLinearMap.restrictScalars (𝒪 K)) → Formalism.HermitianLatticesTheory.dual (hform H) M ≤ M → A.muMaxDual ≤ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianLinkMapAdditiveRationalEquivariant

theorem **infPart_conj** (z : AE D) : infPart D (cA D z) = sigR (Ainf D.host.F) D.sp.cm.sig (infPart D z)

theorem **finPart_conj** (z : AE D) : finPart D (cA D z) = sigR (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F) D.sp.cm.sig (finPart D z)

theorem **map_ofAdj_eq_zero** {B : Type} [CommRing B] (φ : AE D →+* B) (hφ : ∀ z, φ z = 0 → φ (cA D z) = 0) (L S : HSp D) {z : Mx D L.rank S.rank} (hz : z.map φ = 0) : (ofAdj L S z).map φ = 0

theorem **trace_mul_smul_middle_eq_smul_trace** {R : Type} [CommRing R] {m n : ℕ} (c : R) (a : Matrix (Fin m) (Fin m) R) (b : Matrix (Fin m) (Fin n) R) (g : Matrix (Fin n) (Fin n) R) (y : Matrix (Fin n) (Fin m) R) : (a * (b * ((c • g) * y))).trace = c * (a * (b * (g * y))).trace

theorem **hO_ofAdj** (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (c : D.host.E) (hτ : τᴴ * (L.gram * τ) = (c * (D.host.xi0 * star D.host.xi0)) • L'.gram) (x y : WO S L') : hO L S (ofAdj L S ((rat τ : Mx D L.rank L'.rank) * x)) (ofAdj L S ((rat τ : Mx D L.rank L'.rank) * y)) = incA D c * hO S L' y x

theorem **ofAdj_add** (L S : HSp D) (z z' : Mx D L.rank S.rank) : ofAdj L S (z + z') = ofAdj L S z + ofAdj L S z'

theorem **continuous_ofAdj** (L S : HSp D) : Continuous (ofAdj L S)

theorem **ofAdj_mem** (L S : HSp D) {z : Mx D L.rank S.rank} (hz : z ∈ ratMx D L.rank S.rank) : ofAdj L S z ∈ ratMx D S.rank L.rank

theorem **mt_inv_ofAdj** (L S : HSp D) (g : GA D S) (z : Mx D L.rank S.rank) : mt S g⁻¹ * ofAdj L S z = ofAdj L S (z * mt S g)

theorem **ofAdj_mul_mt_inv** (L S : HSp D) (g : GA D S) (z : Mx D L.rank S.rank) : ofAdj L S (z * mt S g⁻¹) = mt S g * ofAdj L S z

theorem **symplO_ofAdj** (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (ε : ℤ) (hτ : τᴴ * (L.gram * τ) = ((ε : D.host.E) * (D.host.xi0 * star D.host.xi0)) • L'.gram) (x y : WO S L') : symplO L S (ofAdj L S ((rat τ : Mx D L.rank L'.rank) * x)) (ofAdj L S ((rat τ : Mx D L.rank L'.rank) * y)) = -((ε : AF D.host.F) * symplO S L' x y)

theorem **kappa_props** : ∀ (D : D7) (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (ε : ℤ), τᴴ * (L.gram * τ) = ((ε : D.host.E) * (D.host.xi0 * star D.host.xi0)) • L'.gram → ∀ κ : WO S L' → WO L S, (∀ w, κ w = GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj L S ((rat τ : Mx D L.rank L'.rank) * w)) → (∀ w w', κ (w + w') = κ w + κ w') ∧ Continuous κ ∧ (∀ w ∈ ratMx D L'.rank S.rank, κ w ∈ ratMx D S.rank L.rank) ∧ (∀ (g : GA D S) (w : WO S L'), κ (w * mt S g⁻¹) = mt S g * κ w) ∧ (∀ w w', symplO L S (κ w) (κ w') = -((ε : AF D.host.F) * symplO S L' w w')) ∧ (∀ w, (w.map (infPart D) = 0 → (κ w).map (infPart D) = 0) ∧ (w.map (finPart D) = 0 → (κ w).map (finPart D) = 0))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianLinkMapAdditiveRationalEquivariantStatement

theorem **_root_.GlobalTheory.T.hermitianLinkMap_additive_rational_equivariant** : ∀ (D : D7) (L L' S : HSp D) (τ : Matrix (Fin L.rank) (Fin L'.rank) D.host.E) (ε : ℤ), τᴴ * (L.gram * τ) = ((ε : D.host.E) * (D.host.xi0 * star D.host.xi0)) • L'.gram → ∀ κ : WO S L' → WO L S, (∀ w, κ w = GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj L S ((rat τ : Mx D L.rank L'.rank) * w)) → (∀ w w', κ (w + w') = κ w + κ w') ∧ Continuous κ ∧ (∀ w ∈ ratMx D L'.rank S.rank, κ w ∈ ratMx D S.rank L.rank) ∧ (∀ (g : GA D S) (w : WO S L'), κ (w * mt S g⁻¹) = mt S g * κ w) ∧ (∀ w w', symplO L S (κ w) (κ w') = -((ε : AF D.host.F) * symplO S L' w w')) ∧ (∀ w, (w.map (infPart D) = 0 → (κ w).map (infPart D) = 0) ∧ (w.map (finPart D) = 0 → (κ w).map (finPart D) = 0))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianPairingOntoInverseIdealAtPlace

theorem **pz_add** (S : HSp D) (z x y : Fin S.rank → D.host.E) : pz S z (x + y) = pz S z x + pz S z y

theorem **pz_smul** (S : HSp D) (z x : Fin S.rank → D.host.E) (c : D.host.E) : pz S z (c • x) = star c * pz S z x

theorem **valSub_isFractional** (S : HSp D) (M : Lat S) (hM : IsSelfDualC S M) (z : Fin S.rank → D.host.E) : IsFractional (𝒪 D.host.E)⁰ (valSub S M z)

theorem **mem_valIdl** {S : HSp D} {M : Lat S} {hM : IsSelfDualC S M} {z : Fin S.rank → D.host.E} {a : D.host.E} : a ∈ valIdl S M hM z ↔ ∃ x ∈ M, pz S z x = a

theorem **hermAt_gen** (S : HSp D) (z x : Fin S.rank → D.host.E) (α : Af D.host.F) : hermAt S (ratV z) (fun i => pairA D.host.F 0 α ⊗ₜ[D.host.F] (1 * x i)) = pairA D.host.F 0 α ⊗ₜ[D.host.F] pz S z x

theorem **hermAt_add** (S : HSp D) (w x y : Fin S.rank → AE D) : hermAt S w (x + y) = hermAt S w x + hermAt S w y

theorem **hermAt_zero** (S : HSp D) (w : Fin S.rank → AE D) : hermAt S w 0 = 0

theorem **hermAt_neg** (S : HSp D) (w x : Fin S.rank → AE D) : hermAt S w (-x) = -hermAt S w x

theorem **cA_sum_eq** (S : HSp D) (Z y : Fin S.rank → AE D) : cA D (∑ l, (∑ m, cA D (Z m) * gA S m l) * y l) = hermAt S Z y

theorem **exists_not_intAt** (v : Pl D) : ∃ u ∈ idAt D (· = v), u ∉ intAt D (· = v)

theorem **exists_not_mem_idV** (v : Pl D) {c : FI D} (h : c ≠ 0) : ∃ t ∈ idAt D (· = v), t ∉ idV c v

theorem **hz_onto** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M : Lat S) (c : FI D) (v : Pl D), IsSelfDualC S M → LineId o ζ M c v → c ≠ 0 → ∀ g ∈ idV c⁻¹ v, ∃ y ∈ latV S M (· = v), cA D (∑ l, (∑ m, cA D (ratV (vecB o ζ) m) * gA S m l) * y l) = g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianPairingOntoInverseIdealAtPlaceStatement

theorem **_root_.GlobalTheory.T.hermitianPairing_onto_inverseIdeal_at_place** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M : Lat S) (c : FI D) (v : Pl D), IsSelfDualC S M → LineId o ζ M c v → c ≠ 0 → ∀ g ∈ idV c⁻¹ v, ∃ y ∈ latV S M (· = v), cA D (∑ l, (∑ m, cA D (ratV (vecB o ζ) m) * gA S m l) * y l) = g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianSelfDualMaxSlope

theorem **coords_dotProduct** {n : ℕ} (f : Module.Dual K (Fin n → K)) (x : Fin n → K) : (Pi.basisFun K (Fin n)).dualBasis.equivFun f ⬝ᵥ x = f x

theorem **dotProduct_mem_one** {n : ℕ} {M : Submodule (𝒪 K) (Fin n → K)} {y : Fin n → K} (hy : y ∈ coordDual M) : ∀ x ∈ M, y ⬝ᵥ x ∈ (1 : Submodule (𝒪 K) K)

theorem **coordDual_spec** [StarRing K] [IsTotallyComplex K] {n : ℕ} (hn : 0 < n) (M : Submodule (𝒪 K) (Fin n → K)) (hM : Submodule.IsLattice K M) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ) (hX : ∀ v, IsUnit (X v).det) : (coordDual M).FG ∧ coordDual M

≠ ⊥ ∧ Formalism.HN.muMaxDef (Pi.basisFun K (Fin n)) (Formalism.HN.dualFrame X) (coordDual M) = Formalism.HN.muMaxDef (Pi.basisFun K (Fin n)).dualBasis (Formalism.HN.dualFrame X) (Formalism.HN.dualLat M)

theorem **muMaxDual_nonpos** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (K0 : Subfield K), IsCMSub K0 → ∀ v0 : InfinitePlace K, IsConjAt v0 → ∀ (n : ℕ) (H : Matrix (Fin n) (Fin n) K) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ), IsRatHerm K0 H X → ∀ M : Submodule (𝒪 K) (Fin n → K), Submodule.IsLattice K M → Formalism.HermitianLatticesTheory.dual (hform H) M ≤ M → 0 < n → (Formalism.L2.DIV.toHLat X M).muMaxDual ≤ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HermitianSelfDualMaxSlopeStatement

theorem **_root_.GlobalTheory.T.hermitian_self_dual_max_slope** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (K0 : Subfield K), IsCMSub K0 → ∀ v0 : InfinitePlace K, IsConjAt v0 → ∀ (n : ℕ) (H : Matrix (Fin n) (Fin n) K) (X : InfinitePlace K → Matrix (Fin n) (Fin n) ℂ), IsRatHerm K0 H X → ∀ M : Submodule (𝒪 K) (Fin n → K), Submodule.IsLattice K M → Formalism.HermitianLatticesTheory.dual (hform H) M ≤ M → 0 < n → (Formalism.L2.DIV.toHLat X M).muMaxDual ≤ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HighestWeightLineOfKTypeMultiplicityOne

theorem **ht_lt** {a b : Ix p q} (h : posC a b) : ht b < ht a

theorem **ht_drop_cons** (r : PosRoot p q) (l : List (PosRoot p q)) : ∑ x, ht x * drop l x < ∑ x, ht x * drop (r :: l) x

theorem **ht_drop_nonneg** (l : List (PosRoot p q)) : 0 ≤ ∑ x, ht x * drop l x

theorem **eq_smul_of_mem_span** {V : Type*} [AddCommGroup V] [Module ℂ V] {act : Mat p q →ₗ[ℂ] Module.End ℂ V} (h : IsLieAct act) {μ : Ix p q → ℚ} {v : V} (hv : ∀ a, act (Eab a a) v = ((μ a : ℚ) : ℂ) • v) {w : V} (hw : w ∈ Submodule.span ℂ (Set.range (mono act v))) (hwt : ∀ a, act (Eab a a) w = ((μ a : ℚ) : ℂ) • w) : ∃ c : ℂ, c • v = w

theorem **isKTypeOf_span** (h : BlockOrd e ε) (hσ : Factors R.U e σ) (μ : Ix p q → ℚ) {v : R.HK.V} (hv : Formalism.BasicNotions.TH.IsHW R.HK.act μ v) : IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) ((Submodule.span ℂ (Set.range (mono R.HK.act v))).map R.ι)

theorem **finrank_hwSub_eq_one** : ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (ε : Fin n → Bool) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e ε → Factors R.U e σ → ∀ μ : Ix p q → ℚ, (∃ S, IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) S) → (∀ S S', IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) S → IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) S' → S = S') → Module.finrank ℂ ↥(hwSub R.HK.act μ) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HighestWeightLineOfKTypeMultiplicityOneStatement

theorem **_root_.GlobalTheory.T.highestWeightLine_of_kTypeMultiplicityOne** : ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (ε : Fin n → Bool) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e ε → Factors R.U e σ → ∀ μ : Ix p q → ℚ, (∃ S, IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) S) → (∀ S S', IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) S → IsKTypeOf ε σ (fun i => (μ (e.symm i) : ℝ)) S' → S = S') → Module.finrank ℂ ↥(hwSub R.HK.act μ) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HilbertClassFieldOfGlobalArtinReciprocity

theorem **art_diag** (ha : IsGlobalArtin art) (k : Eˣ) : art (diagUnits E k) = 1

theorem **chi_norm_local** (ha : IsGlobalArtin art) {B : Type} [CommGroup B] (χ : Gab E Ω →* B) (K : IntermediateField E Ω) [FiniteDimensional E K] [NumberField K] (hK : ∀ σ : G K Ω, χ (GlobalTheory.HeckeGal.toAb E Ω (σ.restrictScalars E)) = 1) (v : HeightOneSpectrum (𝒪 E)) (hv : ∀ x, IsUnitAt v x → χ (art (localIdele E v x)) = 1) (a : Kˣ) : χ (art (locUnit E v (Units.map (Algebra.norm E (S

theorem **infPart_mul_finPart** (a : Ideles E) : Units.map (infPart E) a * Units.map (finPart E) a = a

theorem **art_sign_mem** (ha : IsGlobalArtin art) (U : Subgroup (Gab E Ω)) (hU : IsOpen (U : Set (Gab E Ω))) (j : Ω →+* ℂ) (c : G E Ω) (hc : ∀ x, j (c x) = starRingEnd ℂ (j x)) (hcU : GlobalTheory.HeckeGal.toAb E Ω c ∈ U) : art (infIdele E (InfinitePlace.mk (j.comp (algebraMap E Ω))) (-1)) ∈ U

theorem **isReal_of_conj** (j : Ω →+* ℂ) (c : G E Ω) (hc : ∀ x, j (c x) = starRingEnd ℂ (j x)) : (InfinitePlace.mk (j.comp (algebraMap E Ω))).IsReal

theorem **art_sign_eq** (ha : IsGlobalArtin art) (j : Ω →+* ℂ) (c : G E Ω) (hc : ∀ x, j (c x) = starRingEnd ℂ (j x)) : art (infIdele E (InfinitePlace.mk (j.comp (algebraMap E Ω))) (-1)) = GlobalTheory.HeckeGal.toAb E Ω c

theorem **exists_conj** (w : InfinitePlace E) (hw : w.IsReal) : ∃ (j : Ω →+* ℂ) (c : G E Ω), (∀ x, j (c x) = starRingEnd ℂ (j x)) ∧ InfinitePlace.mk (j.comp (algebraMap E Ω)) = w

theorem **art_mem_of_class** (ha : IsGlobalArtin art) (U : Subgroup (Gab E Ω)) (hU : IsOpen (U : Set (Gab E Ω))) (hI : ∀ (A : ValuationSubring Ω), ∀ σ ∈ Ine (F

theorem **classHom_surjective** : Function.Surjective (GlobalTheory.Mem.UnitIndex.classHom E)

theorem **mem_ine_iff** (A : ValuationSubring Ω) (σ : G E Ω) : σ ∈ (A.inertiaSubgroup E).map (A.decompositionSubgroup E).subtype ↔ σ ∈ Ine (F

theorem **mem_hilbertSubgroup** (ha : IsGlobalArtin art) (σ : G E Ω) (x : Ideles E) (hσ : art x = GlobalTheory.HeckeGal.toAb E Ω σ) (hx : GlobalTheory.Mem.UnitIndex.classHom E x = 1) : σ ∈ hilbertSubgroup E Ω

theorem **GlobalTheory.Published.GlobalArtinReciprocity.hilbertClassField** : GlobalTheory.Published.GlobalArtinReciprocity → Neutral.HilbertClassField.Statement

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HilbertClassFieldOfGlobalArtinReciprocityStatement

theorem **_root_.GlobalTheory.T.hilbertClassField_of_globalArtinReciprocity** : GlobalTheory.Published.GlobalArtinReciprocity → Neutral.HilbertClassField.Statement

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HolomorphicContinuationValueEqDivMul

theorem **holValueAt_eq_div_mul** : ∀ (Z L d Y : ℂ → ℂ) (s₀ z l : ℂ) (s₁ : ℝ), 0 < s₀.re → s₁ < s₀.re → AnalyticOnNhd ℂ d (rightOf 0) → (∀ s : ℂ, 0 < s.re → d s ≠ 0) → DifferentiableOn ℂ Y (rightOf s₁) → EqRight Z (fun s => L s / d s * Y s) → HolValueAt Z s₀ z → HolValueAt L s₀ l → z = l / d s₀ * Y s₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HolomorphicContinuationValueEqDivMulStatement

theorem **_root_.GlobalTheory.T.holomorphicContinuationValue_eq_div_mul** : ∀ (Z L d Y : ℂ → ℂ) (s₀ z l : ℂ) (s₁ : ℝ), 0 < s₀.re → s₁ < s₀.re → AnalyticOnNhd ℂ d (rightOf 0) → (∀ s : ℂ, 0 < s.re → d s ≠ 0) → DifferentiableOn ℂ Y (rightOf s₁) → EqRight Z (fun s => L s / d

s * Y s) → HolValueAt Z s₀ z → HolValueAt L s₀ l → z = l / d s₀ * Y s₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HolomorphicContinuationValueNnrealPowFactor

theorem **holValueAt_npow_sub_mul** : ∀ (h h' : ℂ → ℂ) (s₀ c : ℂ) (r : ℝ≥0) (σ : ℝ), 0 < r → HolValueAt h s₀ c → (∀ s : ℂ, σ < s.re → h' s = npow r (s₀ - s) * h s) → HolValueAt h' s₀ c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HolomorphicContinuationValueNnrealPowFactorStatement

theorem **_root_.GlobalTheory.T.holomorphicContinuationValue_nnrealPow_factor** : ∀ (h h' : ℂ → ℂ) (s₀ c : ℂ) (r : ℝ≥0) (σ : ℝ), 0 < r → HolValueAt h s₀ c → (∀ s : ℂ, σ < s.re → h' s = npow r (s₀ - s) * h s) → HolValueAt h' s₀ c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomAtIffLocPPerO

theorem **exists_den_away** (D : D7) (P : HeightOneSpectrum (𝒪 D.host.E)) (hP : ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ P.asIdeal) : ∃ z : 𝒪 D.host.E, z ∉ P.asIdeal ∧ ∀ t ∈ D.host.AA0, ∃ w : 𝒪 D.host.E, (w : D.host.E) = t * z

theorem **homAt_iff_locP_perO** : ∀ (D : D7) (S T : HSp D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), (∀ Q ∈ SS, ((D.host.QS * D.host.q * D.host.d : ℕ) : 𝒪 D.host.E) ∉ Q.asIdeal) → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ x : Formalism.L2.Col.HomE D.host.E S.rank T.rank, x ∈ Formalism.L2.Col.homAt SS N N' ↔ ∀ P ∈ SS, x ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeAtPlacesCongrLocalLattices

theorem **latV_mono'** (V : HSp D) (L : Lat V) {P P' : Pl D → Prop} (h : ∀ w, P w → P' w) : latV V L P ≤ latV V L P'

theorem **latO_le_right** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (N' M' : Lat T) (P : Pl D → Prop) (h : latV T N' P ≤ latV T M' P) : latO S T LS (N' : Set _) P ≤ latO S T LS (M' : Set _) P

theorem **latO_le_left** (S T : HSp D) (N M : Lat S) (LT : Set (Fin T.rank → D.host.E)) (P : Pl D → Prop) (h : latV S N P ≤ latV S M P) : latO S T (N : Set _) LT P ≤ latO S T (M : Set _) LT P

theorem **latV_le_of_local** (S : HSp D) (N M : Lat S) (hN : Submodule.IsLattice D.host.E N) (hM : Submodule.IsLattice D.host.E M) (P : Pl D → Prop) (h : ∀ p, P p → latV S N (· = p) ≤ latV S M (· = p)) : latV S N P ≤ latV S M P

theorem **latO_congr** : ∀ (D : D7) (S T : HSp D) (N M : Lat S) (N' M' : Lat T), IsSelfDualC S N → IsSelfDualC S M → IsSelfDualC T N' → IsSelfDualC T M' → ∀ (P : Pl D → Prop), (∀ p, P p → latV S N (· = p) = latV S M (· = p) ∧ latV T N' (· = p) = latV T M' (· = p)) → latO S T (N : Set _) (N' : Set _) P = latO S T (M : Set _) (M' : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeAtPlacesCongrLocalLatticesStatement

theorem **_root_.GlobalTheory.T.homLatticeAtPlaces_congr_localLattices** : ∀ (D : D7) (S T : HSp D) (N M : Lat S) (N' M' : Lat T), IsSelfDualC S N → IsSelfDualC S M → IsSelfDualC T N' → IsSelfDualC T M' → ∀ (P : Pl D → Prop), (∀ p, P p → latV S N (· = p) = latV S M (· = p) ∧ latV T N' (· = p) = latV T M' (· = p)) → latO S T (N : Set _) (N' : Set _) P = latO S T (M : Set _) (M' : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionCutIntegralProperties

theorem **pairA_zero_mul_pairA_zero** (F : Type) [Field F] [NumberField F] (a b : Af F) : pairA F 0 a * pairA F 0 b = pairA F 0 (a * b)

theorem **pairA_zero_smul_tmul** (F E : Type) [Field F] [NumberField F] [Field E] [Algebra F E] (a b : Af F) (e : E) : pairA F 0 a • (pairA F 0 b ⊗ₜ[F] e : AF F ⊗[F] E) = pairA F 0 (a * b) ⊗ₜ[F] e

theorem **infPart_pairA_zero_tmul** (D : D7) (b : Af D.host.F) (e : D.host.E) : infPart D (pairA D.host.F 0 b ⊗ₜ[D.host.F] e) = 0

theorem **latO_algebra** : ∀ (D : D7) (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : Pl D → Prop) (u : WO S T), u ∈ latO S T LS LT P → (∀ γ ∈ intAf D.host.F, pairA D.host.F 0 γ • u ∈ latO S T LS LT (fun v => P v ∧ evalV D.host.F v γ ≠ 0)) ∧ pairA D.host.F 0 1 • u = u ∧ u.map (infPart D) = 0 ∧ ((∀ v, ¬ P v) → u = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionCutIntegralPropertiesStatement

theorem **_root_.GlobalTheory.T.homLatticeCompletion_cut_integral_properties** : ∀ (D : D7) (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : Pl D → Prop) (u : WO S T), u ∈ latO S T LS LT P → (∀ γ ∈ intAf D.host.F, pairA D.host.F 0 γ • u ∈ latO S T LS LT (fun v => P v ∧ evalV D.host.F v γ ≠ 0)) ∧ pairA D.host.F 0 1 • u = u ∧ u.map (infPart D) = 0 ∧ ((∀ v, ¬ P v) → u = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionLeClosureIsotropicHalves

theorem **list_sum_pair** {ι : Type} (l : List ι) (α : ι → Af D.host.F) : (l.map fun i => pairA D.host.F 0 (α i) ⊗ₜ[D.host.F] (1 : D.host.E)).sum = pairA D.host.F 0 (l.map α).sum ⊗ₜ[D.host.F] (1 : D.host.E)

theorem **sum_splitIdem_eq** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} {p : ι → SplitPt D k} {I : List ι} {P : Pl D → Prop} (h : Places p I P) : (I.map fun i => (p i).e + cA D (p i).e).sum = GlobalTheory.FinPlLoc.idem D P

theorem **isSplitIdem_cA** {e : AE D} (he : IsSplitIdem D e) : IsSplitIdem D (cA D e)

theorem **smul_mem_halves** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {P : Pl D → Prop} {e : AE D} (he : IsSplitIdem D e) {u : WO S T} (hu : u ∈ latO S T LS LT P) : e • u ∈ {x | x ∈ latO S T LS LT P ∧ ∃ e, IsSplitIdem D e ∧ IsHalf e x}

theorem **entry_mem_idAt** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {P : Pl D → Prop} {u : WO S T} (hu : u ∈ latO S T LS LT P) : ∀ i j, u i j ∈ idAt D P

theorem **latO_le_closure_halves** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)), ∀ u ∈ latO S T LS LT P, u ∈ AddSubgroup.closure {x | x ∈ latO S T LS LT P ∧ ∃ e, IsSplitIdem D e ∧ IsHalf e x}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionLeClosureIsotropicHalvesStatement

theorem **_root_.GlobalTheory.T.homLatticeCompletion_le_closure_isotropicHalves** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀

(S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)), ∀ u ∈ latO S T LS LT P, u ∈ AddSubgroup.closure {x | x ∈ latO S T LS LT P ∧ ∃ e, IsSplitIdem D e ∧ IsHalf e x}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionMovesByTranslatePair

theorem **finGL_mul** (V : HSp D) (g h : GA D V) : (finGL V (g * h)).val = (mt V g).map (toEAf D) * (finGL V h).val

theorem **latFr_rfl** (V : HSp D) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) : GlobalTheory.AdForm.latFr V rfl N = (N : Set (Fin V.rank → D.host.E))

theorem **eq_lat_of_moves** (V : HSp D) (g : GA D V) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hN : Moves V rfl g N) : N = GlobalTheory.NormSum.AdLattice.lat (finGL V g)

theorem **latO_move** : ∀ (D : D7) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (g s : GA D S) (h s' : GA D T) N N' N₁ N₁', Moves S hS g N → Moves T hT h N' → Moves S hS (s * g) N₁ → Moves T hT (s' * h) N₁' → ∀ u : WO S T, u ∈ (latO S T (latFr S hS N₁) (latFr T hT N₁') fun _ => True) ↔ ∃ u₀ ∈ (latO S T (latFr S hS N) (latFr T hT N') fun _ => True), u = actO S T s s' u₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionMovesByTranslatePairStatement

theorem **_root_.GlobalTheory.T.homLatticeCompletion_moves_by_translatePair** : ∀ (D : D7) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (g s : GA D S) (h s' : GA D T) N N' N₁ N₁', Moves S hS g N → Moves T hT h N' → Moves S hS (s * g) N₁ → Moves T hT (s' * h) N₁' → ∀ u : WO S T, u ∈ (latO S T (latFr S hS N₁) (latFr T hT N₁') fun _ => True) ↔ ∃ u₀ ∈ (latO S T (latFr S hS N) (latFr T hT N') fun _ => True), u = actO S T s s' u₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionSelfAnnihilatorAtPlaces

theorem **evalV_indP** (P : Pl D → Prop) (v : Pl D) : evalV D.host.F v (indP P) = if P v then 1 else 0

theorem **cA_alg** (a : AF D.host.F) : cA D (algebraMap (AF D.host.F) (AE D) a) = algebraMap (AF D.host.F) (AE D) a

theorem **hO_smul_swap** (S T : HSp D) (c : AF D.host.F) (x y : WO S T) : hO S T x (c • y) = hO S T (c • x) y

theorem **symplO_smul_swap** (S T : HSp D) (c : AF D.host.F) (x y : WO S T) : symplO S T x (c • y) = symplO S T (c • x) y

theorem **compFin_repr** (w : Pl D) (y : Af D.host.F ⊗[D.host.F] D.host.E) (k : Fin (Module.finrank D.host.F D.host.E)) : (Algebra.TensorProduct.basis (Fv D.host.F w) (Module.finBasis D.host.F D.host.E)).repr (GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w y) k = evalV D.host.F w ((Algebra.TensorProduct.basis (Af D.host.F) (Module.finBasis D.host.F D.host.E)).repr y k)

theorem **eq_zero_of_compFin** (y : Af D.host.F ⊗[D.host.F] D.host.E) (h : ∀ w, GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w y = 0) : y = 0

theorem **compFin_alg_cP** (P : Pl D → Prop) (w : Pl D) (hw : P w) : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w (finPart D (algebraMap (AF D.host.F) (AE D) (cP P))) = 1

theorem **algebraMap_placeIdempotent_mul** (P : Pl D → Prop) (z : AE D) (hz : z ∈ idAt D P) : algebraMap (AF D.host.F) (AE D) (cP P) * z = z

theorem **smul_mem_latO** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : Pl D → Prop) (y : WO S T) (hy : y ∈ latO S T LS LT fun _ => True) : cP P • y ∈ latO S T LS LT P

theorem **latO_ann_at** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (P : Pl D → Prop) (u : WO S T), MxAt P u → (u ∈ latO S T (LS : Set _) (LT : Set _) P ↔ ∀ u' ∈ latO S T (LS : Set _) (LT : Set _) P, ψ (symplO S T u u') = 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionSelfAnnihilatorAtPlacesStatement

theorem **_root_.GlobalTheory.T.homLatticeCompletion_self_annihilator_atPlaces** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (P : Pl D → Prop) (u : WO S T), MxAt P u → (u ∈ latO S T (LS : Set _) (LT : Set _) P ↔ ∀ u' ∈ latO S T (LS : Set _) (LT : Set _) P, ψ (symplO S T u u') = 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionSplitPlacesActsBySigns

theorem **af_ext'** {D : D7} {x y : Af D.host.F} (h : ∀ w, evalV D.host.F w x = evalV D.host.F w y) : x = y

theorem **latO_acts_by_signs** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ x ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) P, ∃ ε : ℂ, ε * ε = 1 ∧ ∀ f, Vac ψ S T LS LT P f → heisO ψ S T x f = ε • f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeCompletionSplitPlacesActsBySignsStatement

theorem **_root_.GlobalTheory.T.homLatticeCompletion_splitPlaces_acts_by_signs** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ x ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) P, ∃ ε : ℂ, ε * ε = 1 ∧ ∀ f, Vac ψ S T LS LT P f → heisO ψ S T x f = ε • f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeIdealIsUnitSplitPlace

theorem **unit_at_split_place** : ∀ (D : D7) (PP : Pl D.host.E), ((D.host.QS * D.host.q * Ideal.absNorm D.host.ccK : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal → ∀ (v : FinPl D) (φ : SplitProj D v), IsProjAt φ PP → ((∃ a : D.host.E, a ∈ D.host.AA ∧ Valued.v (rdQ φ a) = 1) ∧ ∀ a : D.host.E, a ∈ D.host.AA → Valued.v (rdQ φ a) ≤ 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeIdealIsUnitSplitPlaceStatement

theorem **_root_.GlobalTheory.T.homLatticeIdeal_isUnit_splitPlace** : ∀ (D : D7) (PP : Pl D.host.E), ((D.host.QS * D.host.q * Ideal.absNorm D.host.ccK : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal → ∀ (v : FinPl D) (φ : SplitProj D v), IsProjAt φ PP → ((∃ a : D.host.E, a ∈ D.host.AA ∧ Valued.v (rdQ φ a) = 1) ∧ ∀ a : D.host.E, a ∈ D.host.AA → Valued.v (rdQ φ a) ≤ 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeIndicatorOnSplitHalfEqHalfIndicator

theorem **sec_one** (p : SplitPt D k) : p.sec 1 = p.e

theorem **e_mul_sec** (p : SplitPt D k) (a : k) : p.e * p.sec a = p.sec a

theorem **sec_mul** (p : SplitPt D k) (a b : k) : p.sec (a * b) = p.sec a * p.sec b

theorem **e_mem_idAt** (p : SplitPt D k) : p.e ∈ idAt D (· = p.v)

theorem **mul_mem_latV** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {w : AE D} (hw : finPart D w ∈ intAE D) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (fun i => w * y i) ∈ latV S L P

theorem **mulVec_sec** (α : Frame p S n') (β : Frame p T n) {t : WO S T} (ht : IsHalf p.e t) (x : n' → k) : t *ᵥ (fun j => p.sec ((α.b *ᵥ x) j)) = fun i => p.sec ((β.b *ᵥ (rd α β t *ᵥ x)) i)

theorem **mem_latO_iff_rd** (α : Frame p S n') (β : Frame p T n) {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (hU : p.UnitAA) {ΛS : Submodule 𝒪[k] (n' → k)} {ΛT : Submodule 𝒪[k] (n → k)} (hα : α.IsLat LS ΛS) (hβ : β.IsLat LT ΛT) {t : WO S T} (ht : IsHalf p.e t) : t ∈ latO S T (LS : Set _) (LT : Set _) (· = p.v) ↔ ∀ x ∈ ΛS, rd α β t *ᵥ x ∈ ΛT

theorem **GlobalTheory.FinPl.LatIndHalf.extFun_latInd_eq_halfInd** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → p.UnitAA → ∀ (ΛS : Submodule 𝒪[k] (n' → k)) (ΛT : Submodule 𝒪[k] (n → k)), α.IsLat LS ΛS → β.IsLat LT ΛT → extFun α β (latInd ΛS ΛT) = halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticeIndicatorOnSplitHalfEqHalfIndicatorStatement

theorem **_root_.GlobalTheory.T.homLatticeIndicator_on_split_half_eq_half_indicator** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → p.UnitAA → ∀ (ΛS : Submodule 𝒪[k] (n' → k)) (ΛT : Submodule 𝒪[k] (n → k)), α.IsLat LS ΛS → β.IsLat LT ΛT → extFun α β (latInd ΛS ΛT) = halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticePlaceCutSmulMemOpenSubgroup

theorem **exists_finset_of_nhds_zero** {F : Type} [Field F] [NumberField F] (V : Set (Af F)) (hV : IsOpen V) (h0 : (0 : Af F) ∈ V) : ∃ s : Finset (HeightOneSpectrum (𝒪 F)), ∀ γ ∈ Formalism.MainNotions.StandardLattice.intAf F, (∀ v ∈ s, evalV F v γ = 0) → γ ∈ V

theorem **smul_mem_open_of_vanish** : ∀ (D : D7) (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : Pl D → Prop) (U : AddSubgroup (WO S T)), IsOpen (U : Set (WO S T)) → ∀ u ∈ latO S T LS LT P, ∃ s : Finset (Pl D), ∀ γ ∈ intAf D.host.F, (∀ v ∈ s, evalV D.host.F v γ = 0) → pairA D.host.F 0 γ • u ∈ U

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomLatticePlaceCutSmulMemOpenSubgroupStatement

theorem **_root_.GlobalTheory.T.homLattice_placeCut_smul_mem_openSubgroup** : ∀ (D : D7) (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (P : Pl D → Prop) (U : AddSubgroup (WO S T)), IsOpen (U : Set (WO S T)) → ∀ u ∈ latO S T LS LT P, ∃ s : Finset (Pl D), ∀ γ ∈ intAf D.host.F, (∀ v ∈ s, evalV D.host.F v γ = 0) → pairA D.host.F 0 γ • u ∈ U

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomotheticOfMonomialPrimitiveSelfDualCores

theorem **homothetic_of_monomial_primitive_selfDual_cores** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (z : Fin S₀.rank → D.host.E) (L L' : Lat S) (M M' : Lat S₀) (v : GlobalTheory.FinPl.Pl D) (c : AE D) (γ : GA D S₀), IsSelfDualC S L → IsSelfDualC S L' → IsSelfDualC S₀ M → z ≠ 0 → MonoAt d L M v → MonoAt d L' M' v → PrimAt ψ d L v → PrimAt ψ d L' v → LastAt ψ d z L M v → LastAt ψ d z L' M' v → NormOneAt v c → Formalism.L2.Theta.mt S₀ γ *ᵥ ratV z = ratV z → SMoves S₀ c γ M M' v → SMoves S c (d.emb0 γ) L L' v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HomotheticOfMonomialPrimitiveSelfDualCoresStatement

theorem **_root_.GlobalTheory.T.sMoves_emb0_of_sMoves_selfDual** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (z : Fin S₀.rank → D.host.E) (L L' : Lat S) (M M' : Lat S₀) (v : GlobalTheory.FinPl.Pl D) (c : AE D) (γ : GA D S₀), IsSelfDualC S L → IsSelfDualC S L' → IsSelfDualC S₀ M → z ≠ 0 → MonoAt d L M v → MonoAt d L' M' v → PrimAt ψ d L v → PrimAt ψ d L' v → LastAt ψ d z L M v → LastAt ψ d z L' M' v → NormOneAt v c → Formalism.L2.Theta.mt S₀ γ *ᵥ ratV z = ratV z → SMoves S₀ c γ M M' v → SMoves S c (d.emb0 γ) L L' v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaFrobeniusTwistLogAdditiveModPrime

theorem **twistLog_norm_iterate** {K : Type} [NormedField K] (σ : K →+* K) (hσ : ∀ x : K, ‖σ x‖ = ‖x‖) (i : ℕ) (x : K) : ‖(σ ^ i) x‖ = ‖x‖

theorem **twistLog_additive_mod** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ (i : ℕ), 0 < i → ∀ m, ‖MvPowerSeries.coeff m (Formalism.L2.Th1.C.be0 f e (frobOp σ l i f) (MvPowerSeries.X (0 : Fin 2)) (MvPowerSeries.X (1 : Fin 2)))‖ ≤ ‖(l : K)‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaFrobeniusTwistLogAdditiveModPrimeStatement

theorem **_root_.GlobalTheory.T.hondaFrobeniusTwistLog_additive_mod_prime** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ (i : ℕ), 0 < i → ∀ m, ‖MvPowerSeries.coeff m (Formalism.L2.Th1.C.be0 f e (frobOp σ l i f) (MvPowerSeries.X (0 : Fin 2)) (MvPowerSeries.X (1 : Fin 2)))‖ ≤ ‖(l : K)‖

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaFrobeniusTwistLogEigenModPrime

theorem **coefLe_mv_of_ps** {K : Type} [NormedField K] {r : ℝ} {A : K⟦X⟧} (h : ∀ n, ‖coeff n A‖ ≤ r) : ∀ m : Unit →₀ ℕ, ‖MvPowerSeries.coeff m A‖ ≤ r

theorem **norm_ringHom_pow_apply** {K : Type} [NormedField K] (σ : K →+* K) (hσ : ∀ x, ‖σ x‖ = ‖x‖) : ∀ (j : ℕ) (x : K), ‖(σ ^ j) x‖ = ‖x‖

private theorem **norm_coeff_pow_le_one** {K : Type} [NormedField K] [IsUltrametricDist K] {s : Type} (M : MvPowerSeries s K) (hM : ∀ m, ‖MvPowerSeries.coeff m M‖ ≤ 1) : ∀ (N : ℕ) (m : s →₀ ℕ), ‖MvPowerSeries.coeff m (M ^ N)‖ ≤ 1

theorem **frobLog_eigen** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → ∀ (i : ℕ), 0 < i → CoefLe ‖(l : K)‖ ((frobOp σ l i f).subst (Formalism.L2.Th1.C.mulBy f e w) - C ((σ ^ i) w) * frobOp σ l i f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaFrobeniusTwistLogEigenModPrimeStatement

theorem **_root_.GlobalTheory.T.hondaFrobeniusTwistLog_eigen_mod_prime** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), IntLog f e → ∀ w : K, CoefLe 1 (Formalism.L2.Th1.C.mulBy f e w) → ∀ (i : ℕ), 0 < i → CoefLe ‖(l : K)‖ ((frobOp σ l i f).subst (Formalism.L2.Th1.C.mulBy f e w) - C ((σ ^ i) w) * frobOp σ l i f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaFunctionalEquationOneTermType

theorem **sq_apply** (σ : K →+* K) (a : K) : (σ ^ 2) a = σ (σ a)

theorem **frobLift_sq** [IsUltrametricDist K] {l : ℕ} {σ : K →+* K} (h : Unr K l σ) : Formalism.G.IsFrobLift l (l ^ 2) (σ ^ 2)

theorem **setting** [IsUltrametricDist K] {l : ℕ} {σ : K →+* K} (h : Unr K l σ) : Formalism.G.FunctionalEquationSetting K l (l ^ 2) (σ ^ 2)

theorem **coeff_ltLog_pow** (l : ℕ) (hl : 2 ≤ l) (j : ℕ) : coeff ((l ^ 2) ^ j) (ltLog K l) = ((l : K) ^ j)⁻¹

theorem **coeff_ltLog_of_not** (l : ℕ) {n : ℕ} (h : ¬ ∃ j : ℕ, n = (l ^ 2) ^ j) : coeff n (ltLog K l) = 0

theorem **ltLog_funEq** [CharZero K] (l : ℕ) (hl : 2 ≤ l) (τ : K →+* K) : ltLog K l − (l : K)⁻¹ • (PowerSeries.map τ (ltLog K l)).subst ((X : K⟦X⟧) ^ (l ^ 2)) = X

theorem **funEq_scaled** [IsUltrametricDist K] {l : ℕ} {σ : K →+* K} {f : K⟦X⟧} {c ε : K} (hε : ‖ε‖ = 1) (hc : ε = c * (σ ^ 2) ε) (hH : CoefLe ‖(l : K)‖ (C (l : K) * f − C c * frobOp (σ ^ 2) (l ^ 2) 1 f)) (hl : (l : K) ≠ 0) : Formalism.G.IsFunEqType l (l ^ 2) (σ ^ 2) (ε⁻¹ • f)

theorem **funEqType_of_unit** {p q : ℕ} (hq : q ≠ 0) {σ : K →+* K} {g : K⟦X⟧} (hg : Formalism.G.IsFunEqTypeMv (ι

theorem **honda_funEq** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), Formalism.L2.Th1.C.IsLogPair f e → ∀ c ε : K, ‖ε‖ = 1 → ε = c * (σ ^ 2) ε → CoefLe ‖(l : K)‖ (C (l : K) * f − C c * frobOp (σ ^ 2) (l ^ 2) 1 f) → CoefLe 1 (e.subst (C ε * ltLog K l))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaFunctionalEquationOneTermTypeStatement

theorem **_root_.GlobalTheory.T.honda_functional_equation_one_term_type** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ) (σ : K →+* K), Unr K l σ → ∀ (f e : K⟦X⟧), Formalism.L2.Th1.C.IsLogPair f e → ∀ c ε : K, ‖ε‖ = 1 → ε = c * (σ ^ 2) ε → CoefLe ‖(l : K)‖ (C (l : K) * f − C c * frobOp (σ ^ 2) (l ^ 2) 1 f) → CoefLe 1 (e.subst (C ε * ltLog K l))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaTypeReductionToOneTerm

theorem **coeff_frobOp** (τ : K →+* K) {q : ℕ} (hq : q ≠ 0) (i : ℕ) (f : K⟦X⟧) (n : ℕ) : coeff n (frobOp τ q i f) = if q ^ i ∣ n then (τ ^ i) (coeff (n / q ^ i) f) else 0

theorem **pow_apply_pow** (τ : K →+* K) (i k : ℕ) (x : K) : (τ ^ i) ((τ ^ k) x) = (τ ^ (i + k)) x

theorem **norm_pow_apply** (τ : K →+* K) (hτ : ∀ x, ‖τ x‖ = ‖x‖) (i : ℕ) (x : K) : ‖(τ ^ i) x‖ = ‖x‖

theorem **coeff_frobOp_eq_zero** (τ : K →+* K) {q : ℕ} (hq : 2 ≤ q) {f : K⟦X⟧} (hf : constantCoeff f = 0) {k n : ℕ} (h : n ≤ k) : coeff n (frobOp τ q k f) = 0

theorem **pow_apply_coeff_frobOp** (τ : K →+* K) {q : ℕ} (hq : q ≠ 0) (f : K⟦X⟧) (i k : ℕ) {n : ℕ} (hd : q ^ i ∣ n) : (τ ^ i) (coeff (n / q ^ i) (frobOp τ q k f)) = coeff n (frobOp τ q (i + k) f)

theorem **coeff_frobOp_sub_hondaSum** (τ : K →+* K) (l : ℕ) {q : ℕ} (hq : 2 ≤ q) {f : K⟦X⟧} (hf : constantCoeff f = 0) (c : ℕ → K) (i n N : ℕ) (hN : n ≤ N) : coeff n (frobOp τ q i (C (l : K) * f − hondaSum τ q c f)) = (l : K) * coeff n (frobOp τ q i f) − ∑ j ∈ Finset.range N, (τ ^ i) (c j) * coeff n (frobOp τ q (i + (j + 1)) f)

theorem **sum_square_eq_sum_diag** (N : ℕ) (F : ℕ → ℕ → K) (hF : ∀ i j, N ≤ i + j → F i j = 0) : ∑ i ∈ Finset.range N, ∑ j ∈ Finset.range N, F i j = ∑ m ∈ Finset.range N, ∑ i ∈ Finset.range (m + 1), F i (m − i)

theorem **coeff_identity** (τ : K →+* K) (l : ℕ) {q : ℕ} (hq : 2 ≤ q) {f : K⟦X⟧} (hf : constantCoeff f = 0) (c t : ℕ → K) (ht0 : t 0 = 1) (ht : ∀ n, 2 ≤ n → (l : K) * t n = ∑ i ∈ Finset.range n, t i * (τ ^ i) (c (n − 1 − i))) (n : ℕ) : ∑ i ∈ Finset.range (n + 1), t i * coeff n (frobOp τ q i (C (l : K) * f − hondaSum τ q c f)) = (l : K) * coeff n f − (c 0 − l * t 1) * coeff n (frobOp τ q 1 f)

theorem **exists_one_term** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] [CompleteSpace K] (l q : ℕ) (τ : K →+* K), 2 ≤ q → ‖(l : K)‖ < 1 → (∀ x, ‖τ x‖ = ‖x‖) → ∀ (f : K⟦X⟧), constantCoeff f = 0 → ∀ c : ℕ → K, (∀ j, ‖c j‖ ≤ 1) → ‖c 0‖ = 1 → CoefLe ‖(l : K)‖ (C (l : K) * f − hondaSum τ q c f) → ∃ c' : K, ‖c'‖ = 1 ∧ CoefLe ‖(l : K)‖ (C (l : K) * f − C c' * frobOp τ q 1 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HondaTypeReductionToOneTermStatement

theorem **_root_.GlobalTheory.T.hondaType_reduction_to_one_term** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] [CompleteSpace K] (l q : ℕ) (τ : K →+* K), 2 ≤ q → ‖(l : K)‖ < 1 → (∀ x, ‖τ x‖ = ‖x‖) → ∀ (f : K⟦X⟧), constantCoeff f = 0 → ∀ c : ℕ → K, (∀ j, ‖c j‖ ≤ 1) → ‖c 0‖ = 1 → CoefLe ‖(l : K)‖ (C (l : K) * f − hondaSum τ q c f) → ∃ c' : K, ‖c'‖ = 1 ∧ CoefLe ‖(l : K)‖ (C (l : K) * f − C c' * frobOp τ q 1 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HostFieldsPrimesTies

theorem **host_fields_primes_ties** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (hX : B.fields.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)), (I.rec7 l Z).host.ccK = ⊤ ∧ GL3Theta.InducedCMType (I.rec7 l Z) ∧ GL3Theta.HostFieldConditions (I.rec7 l Z) ∧ B.fields.AbelianUnramifiedExtensionConditions ∧ (seamOf B (I.cols l) Z.W Z.V Z.S Z.N Z.G).Ok ∧ (∀ x : B.fields.K, B.embLL (star x) = starRingEnd ℂ (B.embLL x)) ∧ (B.primes X).Ok ∧ (B.primes X).PiGCong ∧ (B.primes X).cond = B.host.QS * B.host.q * B.host.d ∧ (∀ i, (B.primes X).g i = B.host.g i) ∧ B.wts.Ok

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HostTieTrueCols

theorem **host_tie_trueCols** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (h : I.Ok) (l : I.Laws) (Z : TC.Later B (I.cols l)) (L : SizeData₂ (I.rec7 l Z)), GlobalTheory.IndefiniteRowSideFullHecke.PlaceLogThetaBounds.HostTie (numDatumOfParameters (I.rec7 l Z) L) (seamOf B (I.cols l) Z.W Z.V Z.S Z.N Z.G) (B.primes X)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HoweDualityUnitarySubstitutionSpanUnit

theorem **ren_X** {a b : ℕ} (v : Unit × Fin b × Fin a) : ren a b (X v) = X v.2

theorem **ren_injective** (a b : ℕ) : Function.Injective (ren a b)

theorem **matRep_ren** {a b : ℕ} (g : GL (Fin a) ℂ) (h : GL (Fin b) ℂ) (P : FP Unit a b) : matRep a b (g, h) (ren a b P) = ren a b (mvZ (fun _ => (g : Matrix (Fin a) (Fin a) ℂ)ᵀ) (fun _ => (h : Matrix (Fin b) (Fin b) ℂ)) P)

theorem **subY_one_injective** {a b : ℕ} (hab : a ≤ b) : Function.Injective (subY (ι

theorem **vn_stable** {a : ℕ} (lam : Unit → DomWeight a) (n : Unit → ℕ) (g : Unit → Matrix (Fin a) (Fin a) ℂ) : ∀ F ∈ Vn lam n, lft g F ∈ Vn lam n

theorem **vnMap_apply** {a : ℕ} (lam : Unit → DomWeight a) (n : Unit → ℕ) (g : Unit → Matrix (Fin a) (Fin a) ℂ) (F : Vn lam n) : (vnMap lam n g F : PY Unit a) = lft g F

theorem **transpose_uni** {a : ℕ} (k : Matrix.unitaryGroup (Fin a) ℂ) : (k : Matrix (Fin a) (Fin a) ℂ)ᵀ ∈ Matrix.unitaryGroup (Fin a) ℂ

theorem **vnRep_apply** {a : ℕ} (lam : Unit → DomWeight a) (n : Unit → ℕ) (k : Matrix.unitaryGroup (Fin a) ℂ) (F : Vn lam n) : (vnRep lam n k F : PY Unit a) = lft (fun _ => (k : Matrix (Fin a) (Fin a) ℂ)ᵀ) F

theorem **vnRep_irreducible** {a : ℕ} (ha : 0 < a) (lam : Unit → DomWeight a) (n : Unit → ℕ) : (vnRep lam n).IsIrreducible

theorem **equiv_unitary_gl** {a b : ℕ} (ha : 1 ≤ a) (hb : 1 ≤ b) : Nonempty ((Representation.directSum fun j : Part (min a b) => Representation.tprod ((smallRep (unitaryIncl a) (pad a j)).comp (MonoidHom.fst (Matrix.unitaryGroup (Fin a) ℂ) (GL (Fin b) ℂ))) ((smallRep (MonoidHom.id (GL (Fin b) ℂ)) (pad b j)).comp (MonoidHom.snd (Matrix.unitaryGroup (Fin a) ℂ) (GL (Fin b) ℂ)))).Equiv (polyRep (unitaryIncl a) (MonoidHom.id (GL (Fin b) ℂ))))

theorem **subMap_apply** {a b : ℕ} (hab : a ≤ b) (lam : Unit → DomWeight a) (n : Unit → ℕ) (y' : Unit → Matrix (Fin b) (Fin b) ℂ) (F : Vn lam n) : subMap hab lam n y' F = ren a b (subY hab y' F)

theorem **polyRep_apply** {a b : ℕ} (k : Matrix.unitaryGroup (Fin a) ℂ) (h : GL (Fin b) ℂ) (P : FP Unit a b) : polyRep (unitaryIncl a) (MonoidHom.id (GL (Fin b) ℂ)) (k, h) (ren a b P) = ren a b (mvZ (fun _ => (k : Matrix (Fin a) (Fin a) ℂ)ᵀ) (fun _ => (h : Matrix (Fin b) (Fin b) ℂ)) P)

theorem **subMap_intertwines** {a b : ℕ} (hab : a ≤ b) (lam : Unit → DomWeight a) (n : Unit → ℕ) (y' : Unit → Matrix (Fin b) (Fin b) ℂ) (k : Matrix.unitaryGroup (Fin a) ℂ) (F : Vn lam n) : subMap hab lam n y' (vnRep lam n k F) = polyRep (unitaryIncl a) (MonoidHom.id (GL (Fin b) ℂ)) (k, 1) (subMap hab lam n y' F)

theorem **comp_subMap** {a b : ℕ} (hab : a ≤ b) (lam : Unit → DomWeight a) (n : Unit → ℕ) (y' : Unit → Matrix (Fin b) (Fin b) ℂ) (h : GL (Fin b) ℂ) : polyRep (unitaryIncl a) (MonoidHom.id (GL (Fin b) ℂ)) (1, h) ∘ₗ subMap hab lam n y' = subMap hab lam n (fun τ => (h : Matrix (Fin b) (Fin b) ℂ) * y' τ)

theorem **howeSpan_unit** : HoweSpan Unit

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HoweDualityUnitarySubstitutionSpanUnitStatement

theorem **_root_.GlobalTheory.T.howeDualityUnitarySubstitutionSpan_unit** : HoweSpan Unit

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HoweSpanFamiliesOfPlaces

theorem **lft_jY** (g : ι → Matrix (Fin a) (Fin a) ℂ) (τ : ι) (F : PY Unit a) : lft g (jY a τ F) = jY a τ (lft (fun _ => g τ) F)

theorem **mvZ_jZ** (g : ι → Matrix (Fin a) (Fin a) ℂ) (h : ι → Matrix (Fin b) (Fin b) ℂ) (τ : ι) (P : FP Unit a b) : mvZ g h (jZ a b τ P) = jZ a b τ (mvZ (fun _ => g τ) (fun _ => h τ) P)

theorem **subY_jY** (hab : a ≤ b) (y' : ι → Matrix (Fin b) (Fin b) ℂ) (τ : ι) (F : PY Unit a) : subY hab y' (jY a τ F) = jZ a b τ (subY hab (fun _ => y' τ) F)

theorem **tenU** (ι σ : Type) [Fintype ι] [DecidableEq ι] : ∃ e : PiTensorProduct ℂ (fun _ : ι => MvPolynomial (Unit × σ) ℂ) ≃ₗ[ℂ] MvPolynomial (ι × σ) ℂ, (∀ f, e (PiTensorProduct.tprod ℂ f) = ∏ τ, rename (fun x : Unit × σ => (τ, x.2)) (f τ)) ∧ ∀ U : ι → Submodule ℂ (MvPolynomial (Unit × σ) ℂ), Function.Injective (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (U τ).subtype) ∧ LinearMap.range (e.toLinearMap ∘ₗ PiTensorProduct.map fun τ => (U τ).subtype) = Submodule.span ℂ {p | ∃ f : ι → MvPolynomial (Unit × σ) ℂ, (∀ τ, f τ ∈ U τ) ∧ p = ∏ τ, rename (fun x : Unit × σ => (τ, x.2)) (f τ)}

theorem **mem_vn_unit** (l : DomWeight a) (n : ℕ) (F : PY Unit a) : F ∈ Vn (fun _ : Unit => l) (fun _ => n) ↔ ∃ f ∈ delSq ℂ a l, F = rename (Prod.mk ()) f * frDet ℂ a () ^ n

theorem **span_jY** [Fintype ι] (lam : ι → DomWeight a) (n : ι → ℕ) : Submodule.span ℂ {p | ∃ f : ι → PY Unit a, (∀ τ, f τ ∈ Vn (fun _ : Unit => lam τ) (fun _ => n τ)) ∧ p = ∏ τ, rename (fun x : Unit × Fin a × Fin a => (τ, x.2)) (f τ)} = Vn lam n

theorem **lftMap** [Fintype ι] [DecidableEq ι] (eY : PiTensorProduct ℂ (fun _ : ι => PY Unit a) ≃ₗ[ℂ] PY ι a) (hY1 : ∀ f, eY (PiTensorProduct.tprod ℂ f) = ∏ τ, jY a τ (f τ)) (V : ι → Submodule ℂ (PY Unit a)) (hV : ∀ τ k, ∀ F ∈ V τ, lft (fun _ : Unit => k) F ∈ V τ) (τ : ι) (k : Matrix (Fin a) (Fin a) ℂ) : eY.toLinearMap ∘ₗ PiTensorProduct.map (fun σ => (V σ).subtype) ∘ₗ PiTensorProduct.map (Function.update (fun σ => (LinearMap.id : V σ →ₗ[ℂ] V σ)) τ ((lft (fun _ : Unit => k)).toLinearMap.restrict (fun F hF => hV τ k F hF))) = (lft (Function.update 1 τ k)).toLinearMap ∘ₗ eY.toLinearMap ∘ₗ PiTensorProduct.map (fun σ => (V σ).subtype)

theorem **mvZMap** [Fintype ι] [DecidableEq ι] (hab : a ≤ b) (eZ : PiTensorProduct ℂ (fun _ : ι => FP Unit a b) ≃ₗ[ℂ] FP ι a b) (hZ1 : ∀ f, eZ (PiTensorProduct.tprod ℂ f) = ∏ τ, jZ a b τ (f τ)) (τ : ι) (k : Matrix (Fin a) (Fin a) ℂ) : eZ.toLinearMap ∘ₗ PiTensorProduct.map (Function.update (fun σ => (LinearMap.id : FP Unit a b →ₗ[ℂ] FP Unit a b)) τ (mvZ (fun _ : Unit => k) (fun _ => 1)).toLinearMap) = (mvZ (Function.update 1 τ k) (fun _ => 1)).toLinearMap ∘ₗ eZ.toLinearMap

theorem **pure_rep** [Fintype ι] [DecidableEq ι] (hab : a ≤ b) (ha : 0 < a) (lam : ι → DomWeight a) (n : ι → ℕ) (eY : PiTensorProduct ℂ (fun _ : ι => PY Unit a) ≃ₗ[ℂ] PY ι a) (hY1 : ∀ f, eY (PiTensorProduct.tprod ℂ f) = ∏ τ, jY a τ (f τ)) (eZ : PiTensorProduct ℂ (fun _ : ι => FP Unit a b) ≃ₗ[ℂ] FP ι a b) (hZ1 : ∀ f, eZ (PiTensorProduct.tprod ℂ f) = ∏ τ, jZ a b τ (f τ)) (hV : ∀ τ k, ∀ F ∈ Vn (fun _ : Unit => lam τ) (fun _ => n τ), lft (fun _ : Unit => k) F ∈ Vn (fun _ : Unit => lam τ) (fun _ => n τ)) (t : ∀ τ, Vn (fun _ : Unit => lam τ) (fun _ => n τ) →ₗ[ℂ] FP Unit a b) (ht : ∀ τ, ∀ k ∈ Matrix.unitaryGroup (Fin a) ℂ, t τ ∘ₗ (lft (fun _ : Unit => k)).toLinearMap.restrict (fun F hF => hV τ k F hF) = (mvZ (fun _ : Unit => k) (fun _ => 1)).toLinearMap ∘ₗ t τ) : PiTensorProduct.map t ∈ RepMaps hab (eY.toLinearMap ∘ₗ PiTensorProduct.map (fun τ => (Vn (fun _ : Unit => lam τ) (fun _ => n τ)).subtype)

theorem **GlobalTheory.KJInf.howeSpan_all** : ∀ (ι : Type) [Fintype ι], HoweSpan ι

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HoweSpanFamiliesOfPlacesStatement

theorem **_root_.GlobalTheory.T.howeSpan_families_of_places** : ∀ (ι : Type) [Fintype ι], HoweSpan ι

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HyperbolicZeroEmbeddingId

theorem **hyperbolic_zero_j0_one** (V : HSp D) : (hypZero V).j0 = 1

theorem **hyperbolic_zero_embedding_id** (V : HSp D) (g : GA D V) : (hypZero V).emb0 g = g

theorem **mem_flagN_hypZero** (V : HSp D) {n : GA D V} (hn : n ∈ flagN (hypZero V)) : n = 1

theorem **main** : ∀ (D : D7) (V : HSp D), (∀ g, (hypZero V).emb0 g = g) ∧ (∀ n ∈ flagN (hypZero V), n = 1) ∧ ∀ (A L : HSp D) (o : OrthDec V A L) h, embA (hypZero V) o h = o.emb (h, 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.HypsSatisfiable

theorem **weil_exists** : ∀ (D : D7) (n : ℕ) (V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ lam : D.host.F, lam ≠ 0 → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (Mx D V.rank n)) (ℱ : Set (Mx D V.rank n)), IsCovolOne (ratMx D V.rank n) μ ℱ → ∃ ω : G D n →* Module.End ℂ (schwartzA θ V.rank n), IsWeilSch (shift ψ lam) χ θ V μ ω

theorem **fock_gauss** (hg : ∀ (D : D7) (n : ℕ) (V : HSp D) (l : D.what₂.Tau → ℝ), (∀ τ, l τ ≠ 0) → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∃ f : SchwartzMap (D.what₂.Tau → Fin V.rank → Fin n → ℂ) ℂ, ∀ z, f z = gauss V l s (fun τ => Matrix.of (z τ))) : ∀ (D : D7) (n : ℕ) (V : HSp D) (θ : InfFam D) (l : D.what₂.Tau → ℝ), (∀ τ, l τ ≠ 0) → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ U : Set (Fin V.rank → Fin n → Af D.host.F ⊗[D.host.F] D.host.E), IsClopen U → IsCompact U → (0 : Fin V.rank → Fin n → Af D.host.F ⊗[D.host.F] D.host.E) ∈ U → ∃ Φ : schwartzA θ V.rank n, Φ.1 = (fun x => gauss V l s (fun τ => compAt θ τ x) * U.indicator 1 (fun i j => finPart D (x i j))) ∧ Φ.1 ∈ fockA θ V n l s ∧ Φ.1 0 ≠ 0

theorem **hyps_satisfiable** : (∀ (D : D7) (n : ℕ), ∃ H : G D n → ℝ≥0, IsHeight H) ∧ (∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsCharAF D.host.F ψ → ∀ θ : InfFam D, IsInfFam θ → ∃ l : D.what₂.Tau → ℝ, HasRealPar ψ θ l) ∧ (∀ (D : D7) (n : ℕ) (V : HSp D) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ lam : D.host.F, lam ≠ 0 → ∀ χ : (AE D)ˣ →* Circle, IsHeckeChar D χ → ResEta D χ V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (μ : Measure (Mx D V.rank n)) (ℱ : Set (Mx D V.rank n)), IsCovolOne (ratMx D V.rank n) μ ℱ → ∃ ω : G D n →* Module.End ℂ (schwartzA θ V.rank n), IsWeilSch (shift ψ lam) χ θ V μ ω) ∧ (∀ (D : D7) (n : ℕ) (V : HSp D) (θ : InfFam D) (l : D.what₂.Tau → ℝ), (∀ τ, l τ ≠ 0) → ∀ s : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ, (∀ τ, IsCartanInv (gramAt D V τ) (s τ)) → ∀ U : Set (Fin V.rank → Fin n → Af D.host.F ⊗[D.host.F] D.host.E), IsClopen U → IsCompact U → (0 : Fin V.rank

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdeleCharacterEqOneIntegralUnitsUnramifiedAdelic

theorem **ideleAt_eq_prod_locIdele** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) (x : (Fv F v ⊗[F] E)ˣ) : ideleAt F E v x = ∏ w ∈ (Formalism.L2.Host.finite_aboveF (L

theorem **charAt_eq_one_of_unrAt** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (μ : (AF F ⊗[F] E)ˣ →* Circle) (v : HeightOneSpectrum (𝒪 F)) (y : (Fv F v ⊗[F] E)ˣ) (hy : (y : Fv F v ⊗[F] E) ∈ intLoc F E v) (hy' : ((y⁻¹ : (Fv F v ⊗[F] E)ˣ) : Fv F v ⊗[F] E) ∈ intLoc F E v) (hμ : ∀ w : HeightOneSpectrum (𝒪 E), w.under (𝒪 F) = v → UnrAt μ w) : charAt μ v y = 1

theorem **GlobalTheory.LFun.charAt_eq_one_of_unrAt_datum** : ∀ (D : D7) (μ : (AE D)ˣ →* Circle) (v : FinPl D) (y : (Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ), (y : Fv D.host.F v ⊗[D.host.F] D.host.E) ∈ intLoc D.host.F D.host.E v → ((y⁻¹ : (Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ) : Fv D.host.F v ⊗[D.host.F] D.host.E) ∈ intLoc D.host.F D.host.E v → (∀ w : HeightOneSpectrum (𝒪 D.host.E), w.under (𝒪 D.host.F) = v → GlobalTheory.LFun.UnrAt μ w) → charAt μ v y = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdeleCharacterEqOneIntegralUnitsUnramifiedAdelicStatement

theorem **_root_.GlobalTheory.T.ideleCharacter_eq_one_integral_units_unramified_adelic** : ∀ (D : D7) (μ : (AE D)ˣ →* Circle) (v : FinPl D) (y : (Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ), (y : Fv D.host.F v ⊗[D.host.F] D.host.E) ∈ intLoc D.host.F D.host.E v → ((y⁻¹ : (Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ) : Fv D.host.F v ⊗[D.host.F] D.host.E) ∈ intLoc D.host.F D.host.E v → (∀ w : HeightOneSpectrum (𝒪 D.host.E), w.under (𝒪 D.host.F) = v → GlobalTheory.LFun.UnrAt μ w) → charAt μ v y = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdeleCharacterFinitePlaceComponentEqMulOfSplit

theorem **charAt_eq_locChar_mul_locChar** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (χ : (AF F ⊗[F] E)ˣ →* Circle), ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), ∀ (w' : HeightOneSpectrum (𝒪 E)) (h' : w'.under (𝒪 F) = v), w ≠ w' → (∀ u : HeightOneSpectrum (𝒪 E), u.under (𝒪 F) = v ↔ u = w ∨ u = w') → ∀ x : (Fv F v ⊗[F] E)ˣ, ((charAt χ v x : Circle) : ℂ) = (locChar χ w (Units.map (toCompl h).toMonoidHom x) : ℂˣ) * (locChar χ w' (Units.map (toCompl h').toMonoidHom x) : ℂˣ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdeleCharacterFinitePlaceComponentEqMulOfSplitStatement

theorem **_root_.GlobalTheory.T.ideleCharacterFinitePlaceComponent_eq_mul_of_split** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (χ : (AF F ⊗[F] E)ˣ →* Circle), ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), ∀ (w' : HeightOneSpectrum (𝒪 E)) (h' : w'.under (𝒪 F) = v), w ≠ w' → (∀ u : HeightOneSpectrum (𝒪 E), u.under (𝒪 F) = v ↔ u = w ∨ u = w') → ∀ x : (Fv F v ⊗[F] E)ˣ, ((charAt χ v x : Circle) : ℂ) = (locChar χ w (Units.map (toCompl h).toMonoidHom x) : ℂˣ) * (locChar χ w' (Units.map (toCompl h').toMonoidHom x) : ℂˣ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdeleCharacterFinitePlaceComponentEqOfInert

theorem **ideleAt_eq_locIdele_of_isInertAt** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (h : w.under (𝒪 F) = v) (hw : IsInertAt F E v w) (x : (Fv F v ⊗[F] E)ˣ) : ideleAt F E v x = locIdele F w (Units.map (toCompl h).toMonoidHom x)

theorem **charAt_eq_locChar_of_isInertAt** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (χ : (AF F ⊗[F] E)ˣ →* Circle), ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), IsInertAt F E v w → ∀ x : (Fv F v ⊗[F] E)ˣ, ((charAt χ v x : Circle) : ℂ) = (locChar χ w (Units.map (toCompl h).toMonoidHom x) : ℂˣ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdeleCharacterFinitePlaceComponentEqOfInertStatement

theorem **_root_.GlobalTheory.T.ideleCharacterFinitePlaceComponent_eq_of_inert** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (χ : (AF F ⊗[F] E)ˣ →* Circle), ∀ (v : HeightOneSpectrum (𝒪 F)) (w : HeightOneSpectrum (𝒪 E)) (h : w.under (𝒪 F) = v), IsInertAt F E v w → ∀ x : (Fv F v ⊗[F] E)ˣ, ((charAt χ v x : Circle) : ℂ) = (locChar χ w (Units.map (toCompl h).toMonoidHom x) : ℂˣ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdemAtUniqueOrth

theorem **idemAt_unique_orth** : ∀ (D : D7) (a b : AE D) (PP PP' : Formalism.L2.DIV.Pl D.host.E), Formalism.L2.NTheta.IsIdemAt a PP → Formalism.L2.NTheta.IsIdemAt b PP' → (PP = PP' → a = b) ∧ (PP ≠ PP' → a * b = 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdemEMulIdemE

theorem **evalV_idemF** (v w : Pl D) : evalV D.host.F w (idemF v) = if w = v then 1 else 0

theorem **af_ext** {x y : Af D.host.F} (h : ∀ w, evalV D.host.F w x = evalV D.host.F w y) : x = y

theorem **pairA_mul** (α β : Af D.host.F) : pairA D.host.F 0 α * pairA D.host.F 0 β = pairA D.host.F 0 (α * β)

theorem **idemE_mul_gen** (v : Pl D) (α : Af D.host.F) (x : D.host.E) : idemE v * (pairA D.host.F 0 α ⊗ₜ[D.host.F] x) = pairA D.host.F 0 (idemF v * α) ⊗ₜ[D.host.F] x

theorem **idemE_smul_gen** (v : Pl D) (α : Af D.host.F) (x : Fin T.rank → Fin S.rank → D.host.E) : idemE v • (Matrix.of fun i j => pairA D.host.F 0 α ⊗ₜ[D.host.F] x i j : WO S T) = Matrix.of fun i j => pairA D.host.F 0 (idemF v * α) ⊗ₜ[D.host.F] x i j

theorem **idemE_smul_mem** (v : Pl D) {P : Pl D → Prop} {u : WO S T} (hu : u ∈ latO S T LS LT P) : idemE v • u ∈ latO S T LS LT (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentHalfEqRationalAddNsmulStandardHom

theorem **half_eq_rat_add_nsmul** : ∀ {D : D7} (S T : HSp D) (e : AE D) (P : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e P → ∀ M : ℕ, 0 < M → ∀ w : WO S T, IsHalf e w → ∃ (z : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (v : WO S T), v ∈ stdHom S T ∧ IsHalf e v ∧ w = z.map (fun r => e * incA D r) + M • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentHalfEqRationalAddNsmulStandardHomStatement

theorem **_root_.GlobalTheory.T.idempotentHalf_eq_rational_add_nsmul_standardHom** : ∀ {D : D7} (S T : HSp D) (e : AE D) (P : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e P → ∀ M : ℕ, 0 < M → ∀ w : WO S T, IsHalf e w → ∃ (z : Matrix (Fin T.rank) (Fin S.rank) D.host.E) (v : WO S T), v ∈ stdHom S T ∧ IsHalf e v ∧ w = z.map (fun r => e * incA D r) + M • v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentHalfMemCompletedHomLatticeIffLocal

theorem **exists_den** {e : AE D} {P : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt e P) {r : D.host.E} (hr : finPart D (e * incA D r) ∈ intAE D) : ∃ s : 𝒪 D.host.E, s ∉ P.asIdeal ∧ ∃ o : 𝒪 D.host.E, (o : D.host.E) = s * r

theorem **form_apply** (s v : Fin S.rank → D.host.E) : form S s v = ∑ i, ∑ j, star (s i) * S.gram i j * v j

theorem **expand** {n : ℕ} {s s' : Fin n → Fin S.rank → D.host.E} (h : ∀ a j, ∑ i, s i a * ∑ k, star (s' i k) * S.gram k j = if a = j then 1 else 0) (v : Fin S.rank → D.host.E) : v = ∑ k, form S (s' k) v • s k

theorem **coefA_apply** (b : D.host.E) (s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) (u : WO S T) : coefA S T b s t u = ∑ i, ∑ j, ∑ m, incA D (b * (star (t i) * T.gram i j) * s m) * u j m

theorem **coefA_int** (hN : IsSelfDualC S N) (hN' : IsSelfDualC T N') {b : D.host.E} (hb : ∀ a ∈ D.host.AA, ∃ c : 𝒪 D.host.E, (c : D.host.E) = a * b) {s : Fin S.rank → D.host.E} (hs : s ∈ N) {t : Fin T.rank → D.host.E} (ht : t ∈ N') {u : WO S T} (hu : u ∈ latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True) : finPart D (coefA S T b s t u) ∈ intAE D

theorem **coefA_rat** (e : AE D) (b : D.host.E) (s : Fin S.rank → D.host.E) (t : Fin T.rank → D.host.E) (x : Formalism.L2.Col.HomE D.host.E S.rank T.rank) : coefA S T b s t ((LinearMap.toMatrix' x).map fun r => e * incA D r) = e * incA D (b * form T t (x s))

theorem **mem_localization_of_component** (hc : D.host.ccK = ⊤) (hN : IsSelfDualC S N) (hN' : IsSelfDualC T N') {e e' : AE D} {P : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt e' P) (hee : e' * e = e') {x : Formalism.L2.Col.HomE D.host.E S.rank T.rank} (hx : ((LinearMap.toMatrix' x).map fun r => e * incA D r : WO S T) ∈ latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True) : x ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N')

theorem **comp_mem** (hc : D.host.ccK = ⊤) (hN : IsSelfDualC S N) {e : AE D} {P : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt e P) {x : Formalism.L2.Col.HomE D.host.E S.rank T.rank} (hx : x ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N')) : ((LinearMap.toMatrix' x).map fun r => e * incA D r : WO S T) ∈ latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True

theorem **half_mem_latO_iff** : ∀ (D : D7) (S T : HSp D), D.host.ccK = ⊤ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ x : Formalism.L2.Col.HomE D.host.E S.rank T.rank, (((LinearMap.toMatrix' x).map fun r => e * incA D r : WO S T) ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True) ↔ ∀ P ∈ SS, x ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N'))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentHalfMemCompletedHomLatticeIffLocalStatement

theorem **_root_.GlobalTheory.T.idempotentHalf_mem_completedHomLattice_iff_local** : ∀ (D : D7) (S T : HSp D), D.host.ccK = ⊤ → ∀ (e : AE D) (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ (N : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (N' : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), IsSelfDualC S N → IsSelfDualC T N' → ∀ x : Formalism.L2.Col.HomE D.host.E S.rank T.rank, (((LinearMap.toMatrix' x).map fun r => e * incA D r : WO S T) ∈ (latO S T (N : Set (Fin S.rank → D.host.E)) (N' : Set (Fin T.rank → D.host.E)) fun _ => True) ↔ ∀ P ∈ SS, x ∈ Formalism.HermitianLatticesTheory.localization₂ P.asIdeal (perO (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E) N N'))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentMulEntryEqZeroOfCutInverse

theorem **cut_mul** {R : Type} [CommRing R] {n : ℕ} (e f : R) (hef : e * f = 0) (M : Matrix (Fin n) (Fin n) R) : (e • M + (1 - e) • (1 : Matrix (Fin n) (Fin n) R)) * (f • M + (1 - f) • 1) = (e + f) • M + (1 - (e + f)) • 1

theorem **mt_mul** (V : HSp D) (x y : GA D V) : mt V (x * y) = mt V x * mt V y

theorem **mt_one** (V : HSp D) : mt V 1 = 1

theorem **mt_injective** (V : HSp D) : Function.Injective (mt V)

theorem **cA_algebraMap** (a : AF D.host.F) : cA D (algebraMap (AF D.host.F) (AE D) a) = algebraMap (AF D.host.F) (AE D) a

theorem **cut_exists** (V : HSp D) (g : GA D V) (ε : AE D) (hεε : ε * ε = ε) (hε : cA D ε = ε) : ∃ k : GA D V, mt V k = ε • mt V g + (1 - ε) • (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D))

theorem **entry_mem** (V : HSp D) (P : Pl D → Prop) (g : GA D V) (hg : g ∈ grpOff V P) (i j : Fin V.rank) (u : Pl D) (hu : P u) : compAt D u ((mt V g - 1) i j) = 0

theorem **eFin_of_placeIdem** {v : Pl D} {ε : AF D.host.F} (h : IsPlaceIdem D v ε) : ε = eFin D.host.F v

theorem **eFin_idem** (v : Pl D) : eFin D.host.F v * eFin D.host.F v = eFin D.host.F v

theorem **compAt_eq** (u : Pl D) (y : AE D) : compAt D u y = Algebra.TensorProduct.map (evalV D.host.F u) (AlgHom.id D.host.F D.host.E) (Algebra.TensorProduct.map (sndA D.host.F) (AlgHom.id D.host.F D.host.E) y)

theorem **baseAdele_ext_infinite_finite** {a b : AF D.host.F} (h1 : fstA D.host.F a = fstA D.host.F b) (h2 : sndA D.host.F a = sndA D.host.F b) : a = b

theorem **c_mul** {W : Type} [Fintype W] (v : W → Pl D) (hv : Function.Injective v) (a : AF D.host.F) (ha : ∀ u, u ∉ Set.range v → evalV D.host.F u (sndA D.host.F a) = 0) : (1 - pairA D.host.F 1 0 - ∑ w, eFin D.host.F (v w)) * a = 0

theorem **repr_zero2** {F E A A' A'' ι : Type} [Field F] [CommRing E] [Algebra F E] [CommRing A] [Algebra F A] [CommRing A'] [Algebra F A'] [CommRing A''] [Algebra F A''] [Fintype ι] [DecidableEq ι] (b : Module.Basis ι F E) (f : A →ₐ[F] A') (g : A' →ₐ[F] A'') (y : A ⊗[F] E) (hy : Algebra.TensorProduct.map g (AlgHom.id F E) (Algebra.TensorProduct.map f (AlgHom.id F E) y) = 0) (k : ι) : g (f ((Algebra.TensorProduct.basis A b).repr y k)) = 0

theorem **key_gen** {F E A ι : Type} [Field F] [CommRing E] [Algebra F E] [CommRing A] [Algebra F A] [Fintype ι] [DecidableEq ι] (b : Module.Basis ι F E) (c : A) (y : A ⊗[F] E) (hy : ∀ k, c * (Algebra.TensorProduct.basis A b).repr y k = 0) : algebraMap A (A ⊗[F] E) c * y = 0

theorem **key** {W : Type} [Fintype W] (v : W → Pl D) (hv : Function.Injective v) (y : AE D) (hy : ∀ u, u ∉ Set.range v → compAt D u y = 0) : algebraMap (AF D.host.F) (AE D) (1 - pairA D.host.F 1 0 - ∑ w, eFin D.host.F (v w)) * y = 0

theorem **idempotent_mul_entry_eq_zero_of_cut_inverse** {R : Type} [CommRing R] {n : ℕ} (E : R) (hE : E * E = E) (M N : Matrix (Fin n) (Fin n) R) (h1 : (E • M + (1 - E) • (1 : Matrix (Fin n) (Fin n) R)) * N = 1) (i j : Fin n) : E * (M * N - 1) i j = 0

theorem **idempotent_mul_cut_sub_one_entry** {R : Type} [CommRing R] {n : ℕ} (E : R) (hE : E * E = E) (M : Matrix (Fin n) (Fin n) R) (i j : Fin n) : E * (E • M + (1 - E) • (1 : Matrix (Fin n) (Fin n) R) - 1) i j = (E • M + (1 - E) • (1 : Matrix (Fin n) (Fin n) R) - 1) i j

theorem **exists_widePullback_of_iteratedLine** {R : Type} [CommRing R] {n : ℕ} (t : R) (M : Matrix (Fin n) (Fin n) R) (hk : ∀ i j, (1 - t) * (M - 1) i j = 0) : M = t • M + (1 - t) • (1 : Matrix (Fin n) (Fin n) R)

theorem **mt_list_prod** (V : HSp D) {W : Type} (h : W → GA D V) (E : W → AE D) (M : Matrix (Fin V.rank) (Fin V.rank) (AE D)) (hh : ∀ w, mt V (h w) = E w • M + (1 - E w) • (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D))) (horth : ∀ w w', w ≠ w' → E w * E w' = 0) : ∀ l : List W, l.Nodup → mt V ((l.map h).prod) = (l.map E).sum • M + (1 - (l.map E).sum) • 1 | [], _ => by simp [mt_one] | w :: l, hl => by rw [List.nodup_cons] at hl rw [List.map_cons, List.prod_cons, mt_mul, mt_list_prod V h E M hh horth l hl.2, hh, List.map_cons, List.sum_cons] apply cut_mul rw [← List.sum_map_mul_left] apply List.sum_eq_zero intro x hx obtain ⟨y, hy, rfl⟩

theorem **prod_of_trivOff** : ∀ (D : D7) (V : HSp D) (W : Type) [Fintype W] [DecidableEq W] (v : W → Pl D) (e : W → AE D), Function.Injective v → (∀ w, IsSplitIdem D (e w) ∧ ∃ ε, IsPlaceIdem D (v w) ε ∧ e w + cA D (e w) = algebraMap (AF D.host.F) (AE D) ε) → ∀ g ∈ grpOff V (fun x => x ∉ Set.range v), ∃ a ∈ GinfC V, ∃ h : W → GA D V, (∀ w, SuppAt (e w) V (h w)) ∧ (∀ w w', w ≠ w' → Commute (h w) (h w')) ∧ (∀ w, Commute a (h w)) ∧ g = a * (Finset.univ.toList.map h).prod ∧ ∀ w, ∀ i j, (e w + cA D (e w)) * (mt V (g * (h w)⁻¹) - 1) i j = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentSubMulComplementEqComplement

theorem **up_down** (t : WO S T) : p.up FS FT (p.down FS FT t) = p.e • t

theorem **down_add_up** (t : WO S T) (x : Matrix Y Y' k) : p.down FS FT (t + p.up FS FT x) = p.down FS FT t + x

theorem **smul_up** (x : Matrix Y Y' k) : p.e • p.up FS FT x = p.up FS FT x

theorem **extend_apply** (e : AE D) (Ω : WO S T → ℂ) (Φ : SB (Matrix Y Y' k)) (x : WO S T) : extend p FS FT e Ω Φ x = if IsHalf e x then Φ.1 (p.down FS FT x) * Ω ((1 - p.e) • x) else 0

theorem **isModel_extend** {ψ : AddChar (AF D.host.F) ℂ} {e : AE D} (hq : p.e * e = p.e) {Λ : AddSubgroup (WO S T)} {f₀ : WO S T → ℂ} {Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)} (hTm : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e Λ 1 f₀ Tm) (hth : ∀ Φ, IsSBHalf e Λ Φ → IsThetaFunO ψ S T (Tm Φ)) (Ω : WO S T → ℂ) (hsb : ∀ Φ, IsSBHalf e Λ (extend p FS FT e Ω Φ)) : p.IsModel ψ FS FT (Tm ∘ₗ extend p FS FT e Ω)

theorem **β_eq_moved** (g : GA D S) : FS.β = (Formalism.L2.Theta.mt S g).map p.π * (moved p FS g⁻¹).β

theorem **map_eq_of_movedTo** {π' : EAf D →+* k} (hπ : ∀ a, π' (toEAf D a) = p.π a) {g : GA D S} {β : GL (Fin S.rank) (EAf D)} (B : Matrix Y' Y' k) (σ : Fin S.rank → Y') (h : MovedTo p S g β ((FS.β * B).submatrix id σ)) : β.val.map π' = ((moved p FS g⁻¹).β * B).submatrix id σ

theorem **coords_down** {π' : EAf D →+* k} (hπ : ∀ a, π' (toEAf D a) = p.π a) {ι ι' : Type} [Fintype ι] [DecidableEq ι] [Fintype ι'] [DecidableEq ι'] (LS : LatFrame k ι' Y') (LT : LatFrame k ι Y) (ea : ι' ≃ Fin S.rank) (eb : ι ≃ Fin T.rank) {βS : GL (Fin S.rank) (EAf D)} {βT : GL (Fin T.rank) (EAf D)} (hS : βS.val.map π' = (FS.β * LS.bas.val).submatrix id fun i => LS.ord (ea.symm i)) (hT : βT.val.map π' = (FT.β * LT.bas.val).submatrix id fun i => LT.ord (eb.symm i)) (x : WO S T) : LT.coords LS (p.down FS FT x) = ((frMapO S T βS βT x).map π').submatrix eb ea

theorem **omegaG_eq_prod** {a b r t j : ℕ} (hr : r ≤ a) (Pg Q : Finset (Pl D.host.E)) (P : LevelP Pg r ℂ) (χ : ℕ → ∀ w : Pl D.host.E, (kP w)ˣ →* ℂˣ) (ξ : ∀ w : Pl D.host.E, (kP w)ˣ →* ℂˣ) {X : Pg → Type} (Rd : ∀ w : Pg, X w → GL (Fin r) (kP w.1) → Prop) (lam : ∀ w : Pg, X w → ℂˣ) (hg : ∀ g : ∀ w : Pg, GL (Fin r) (kP w.1), g ∈ P.H ↔ ∀ w, ∃ x, Rd w x (g w)) (hc : ∀ (g : P.H) (x : ∀ w, X w), (∀ w, Rd w (x w) (g.1 w)) → P.chi g = ∏ w, lam w (x w)) (y : ∀ w : Pl D.host.E, Matrix (Fin b) (Fin a) (kP w)) : omegaG (Formalism.IndefR.Lev.datY a r t j hr Pg Q P χ ξ .B) (χ (j + 1)) y = ∏ w : Pg, locF t j hr (Rd w) (lam w) (fun c => χ c w.1) (y w.1)

theorem **idem_mul** {e a : AE D} {P : Finset (Pl D.host.E)} (he : IsIdemSum e P) {w : Pl D.host.E} (hw : w ∈ P) (ha : Formalism.L2.NTheta.IsIdemAt a w) : a * e = a

theorem **frMapO_smul** {S T : HSp D} (βS : GL (Fin S.rank) (EAf D)) (βT : GL (Fin T.rank) (EAf D)) (a : AE D) (x : WO S T) (i : Fin T.rank) (j : Fin S.rank) : frMapO S T βS βT (a • x) i j = toEAf D a * frMapO S T βS βT x i j

theorem **redM_eq** {a b : ℕ} {w : Pl D.host.E} {c : EAf D} {N N' : Matrix (Fin b) (Fin a) (EAf D)} (hN : IsIntM N) (hN' : IsIntM N') (h : ∀ i j, N' i j = c * N i j) (hc : ∀ (z : EAf D) (hz : z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E) (hz' : c * z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E), resS D.host.F w ⟨c * z, hz'⟩ = resS D.host.F w ⟨z, hz⟩) : redM w N = redM w N'

theorem **idempotent_sub_mul_complement_eq_complement** {R : Type} [CommRing R] {e a x : R} (hx : e * x = x) (ha : a * a = a) (h : a * e = a) : (e - a) * ((1 - a) * x) = (1 - a) * x

theorem **ite_and_eq_ite_mul_ite_of_implications** {A B B' : Prop} {u v w w' : ℂ} (h1 : B → B') (h2 : B → u = v * w) (h3 : B → w = w') (h4 : v ≠ 0 → B' → B) {A' : Prop} (h5 : A → A') {_ : Decidable (A ∧ B)} {_ : Decidable A} {_ : Decidable (A' ∧ B')} : (if A ∧ B then u else 0) = if A then v * (if A' ∧ B' then w' else 0) else 0

theorem **redM_eq_map** {a b : ℕ} (w : Pl D.host.E) {N : Matrix (Fin b) (Fin a) (EAf D)} (hN : IsIntM N) : redM w N = (Matrix.of fun i j => ((N i j, hN i j) : Formalism.L2.Herm.intAf D.host.F D.host.E)).map (resS D.host.F w)

theorem **locF_eq** {κ X : Type} [CommRing κ] {a b r t j : ℕ} {hr : r ≤ a} {Rd : X → GL (Fin r) κ → Prop} {lam : X → ℂˣ} {χ : ℕ → κˣ →* ℂˣ} {y : Matrix (Fin b) (Fin a) κ} {v : ℂ} (h : (v ≠ 0 → LowZero a y ∧ ∃ g : GL (Fin a) κ, g.val = topL a y ∧ g ∈ blockTri (blkOf r t) ∧ ∃ x, Rd x (upperLeftBlock hr g)) ∧ (LowZero a y → ∀ g : GL (Fin a) κ, g.val = topL a y → g ∈ blockTri (blkOf r t) → ∀ x, Rd x (upperLeftBlock hr g) → v = (lam x : ℂ) * (∏ c ∈ Finset.range j, (χ (c + 1) (dBlk (blkOf r t) (c + 1) g) : ℂ)) * (χ (j + 1) (Matrix.GeneralLinearGroup.det g) : ℂ)⁻¹)) : locF t j hr Rd lam χ y = v

theorem **firstOrd_val** {κ l r : ℕ} (h : κ + l = r) (a : Fin κ ⊕ Fin l) : (firstOrd h a).val = GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.framePos κ l a

theorem **isIx_of_isOrd** {κ l r t j a : ℕ} (h : κ + l = r) (ha : a = r + j * t) {e : Ix (Fin κ ⊕ Fin l) (Fin t) j ≃ Fin a} (he : GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.zPos e) : IsIx ((Equiv.sumCongr (firstOrd h).symm (Equiv.refl (Fin j × Fin t))).trans e)

theorem **rank_first** {w : Pl D.host.E} (G : GAt d τ w) : D.R.k + (G.s + 1) = D.R.r

theorem **red_iff** {w : Pl D.host.E} (G : GAt d τ w) {π' : EAf D →+* G.k} (hπ : ∀ a, π' (toEAf D a) = G.q.pr a) (hred : IsRed (resS D.host.F w) (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype)) (x : G.S.P.J) (g : GL (Fin D.R.r) (kP w)) (M : Matrix (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.k ⊕ Fin (G.s + 1)) (Formalism.L2.Herm.intAf D.host.F D.host.E)) (hM : g.val.submatrix (firstOrd (rank_first G)) (firstOrd (rank_first G)) = M.map (resS D.host.F w)) : G.Red x g ↔ MaxM ((x : GL _ G.k).val − M.map (π'.comp (Formalism.L2.Herm.intAf D.host.F D.host.E).subtype))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentSumDisjointUnion

theorem **idemSum_union** : ∀ (D : D7) (a b : AE D) (P Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (es : Formalism.L2.DIV.Pl D.host.E → AE D), Disjoint P Q → (∀ PP ∈ Q, Formalism.L2.NTheta.IsIdemAt (es PP) PP) → IsIdemSum a P → IsIdemSum b (P ∪ Q) → b = a + ∑ PP ∈ Q, es PP

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdempotentSumPartialProperties

theorem **mul_sum_of_isIdemAt** {D : D7} {x : AE D} {P : Formalism.L2.DIV.Pl D.host.E} (hx : Formalism.L2.NTheta.IsIdemAt x P) (S : Finset (Formalism.L2.DIV.Pl D.host.E)) (f : Formalism.L2.DIV.Pl D.host.E → AE D) (hf : ∀ P' ∈ S, Formalism.L2.NTheta.IsIdemAt (f P') P') : x * ∑ P' ∈ S, f P' = if P ∈ S then x else 0

theorem **sum_mul_sum_of_subset** {D : D7} (S S' : Finset (Formalism.L2.DIV.Pl D.host.E)) (f g : Formalism.L2.DIV.Pl D.host.E → AE D) (hf : ∀ P ∈ S, Formalism.L2.NTheta.IsIdemAt (f P) P) (hg : ∀ P ∈ S', Formalism.L2.NTheta.IsIdemAt (g P) P) (hS : S ⊆ S') : (∑ P ∈ S, f P) * ∑ P ∈ S', g P = ∑ P ∈ S, f P

theorem **sum_mul_of_not_mem** {D : D7} (S : Finset (Formalism.L2.DIV.Pl D.host.E)) (f : Formalism.L2.DIV.Pl D.host.E → AE D) (hf : ∀ P ∈ S, Formalism.L2.NTheta.IsIdemAt (f P) P) {x : AE D} {R : Formalism.L2.DIV.Pl D.host.E} (hx : Formalism.L2.NTheta.IsIdemAt x R) (hR : R ∉ S) : (∑ P ∈ S, f P) * x = 0

theorem **partial_sums** : ∀ (D : D7) (P₀ ι : Finset (Formalism.L2.DIV.Pl D.host.E)) (e e' : AE D) (es : ι → AE D), Disjoint P₀ ι → (∀ PP, Formalism.L2.NTheta.IsIdemAt (es PP) PP.1) → IsIdemSum e P₀ → IsIdemSum e' (P₀ ∪ ι) → e' * e' = e' ∧ ∀ Q : Finset ι, IsIdemSum (e + ∑ PP ∈ Q, es PP) (P₀ ∪ Q.map (Function.Embedding.subtype (· ∈ ι))) ∧ (e + ∑ PP ∈ Q, es PP) * (e + ∑ PP ∈ Q, es PP) = e + ∑ PP ∈ Q, es PP ∧ (e + ∑ PP ∈ Q, es PP) * e' = e + ∑ PP ∈ Q, es PP ∧ ∀ R, R ∉ Q → (e + ∑ PP ∈ Q, es PP) * es R = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdentityCosetEntryProportionalColumnDatum

theorem **om_one_one** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (Φ : Matrix b a k → ℂ) : om μ (1 : GL a k) (1 : GL b k) Φ = Φ

theorem **phi_eq** (j c : ℕ) (h : j + c = s) (x : Matrix (XIx (Fin κ) t (j + 1) ⊕ Fin c) (XIx (Fin κ) t j ⊕ Fin (c + 1)) k) : (S.T t).Phi j c h x = (S.P.iter t S.μ S.tame j).Phi t (S.μ (j + 1)) (((crd S x).submatrix (stdOrd (Fin κ) t (l

theorem **phi_supp** (j c : ℕ) (h : j + c = s) (x : Matrix (XIx (Fin κ) t (j + 1) ⊕ Fin c) (XIx (Fin κ) t j ⊕ Fin (c + 1)) k) (hx : (S.T t).Phi j c h x ≠ 0) : IntM (crd S x)

theorem **phi_per** (j c : ℕ) (h : j + c = s) (x y : Matrix (XIx (Fin κ) t (j + 1) ⊕ Fin c) (XIx (Fin κ) t j ⊕ Fin (c + 1)) k) (hy : MaxM (crd S y)) : (S.T t).Phi j c h (x + y) = (S.T t).Phi j c h x

theorem **reduction_one'** (P : HeightOneSpectrum (𝒪 E)) : redP P (1 : E) = 1

theorem **reduction_natCast_eq_zero** (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) : redP P (q : E) = 0

theorem **q_ne_zero** (hu : ((q : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) : (q : E) ≠ 0

theorem **pow_red** (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (k : ℕ) : IntAt P ((q : E) ^ (k + 1)) ∧ redP P ((q : E) ^ (k + 1)) = 0

theorem **valuation_one₂** (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (hu : ((q : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) : LocE.vP P q (1 : E) = 0

theorem **locChar_one** {K : Type} [Field K] (hq : ((q : ℕ) : 𝒪 E) ∈ P.asIdeal) (hu : ((q : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) (cq : K) (χ : (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) : LocE.locChar q cq χ (1 : E) = 1

theorem **relIndex_ne_of_map** {G : Type} [Group G] (H K : Subgroup G) {F : Type} [Finite F] (f : G → F) (h : ∀ x ∈ K, ∀ y ∈ K, f x = f y → x⁻¹ * y ∈ H) : H.relIndex K ≠ 0

theorem **relIndex_congruenceJacobiSubgroup_stabilizer_ne_zero** {E : Type} [Field E] [NumberField E] [StarRing E] {P : HeightOneSpectrum (𝒪 E)} (hU : ChainCounts.UnifAt P) (k₀ rS r t n c : ℕ) : (Loc.Cbe (LocE.JGsym P (qOf P) k₀ rS r t n c) 1 ⊓ Loc.N1 E n c).relIndex (Loc.N1 E n c ⊓ stabL E (n + c) (LocE.lamY P (qOf P) n c fun i => i + 1)) ≠ 0

theorem **volN_one_ne** (j : ℕ) {P : HeightOneSpectrum (𝒪 X.E)} (hP : P ∈ Pr.vGs) (hu : ((qOf P : ℕ) : 𝒪 X.E) ∉ P.asIdeal ^ 2) : Loc.volN X.K (Loc.Cbe (Jat (Pr

theorem **omPhi_one** {X : Fields} {Pr : Primes X} {R : Ranks} (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S₀ : Set ℕ} (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (i : ℕ) (P : rowSG Pr sel mu) (hq : ((qOf P.1 : ℕ) : 𝒪 X.E) ∈ P.1.asIdeal) (hu : ((qOf P.1 : ℕ) : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) : omPhi Ch sel mu cst i P 1 1 = PhiAt Ch mu cst.sharp i P.1

theorem **entry_agree** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (S : SymData k s R.k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → AtType Ch sel mu cst σ₀ P ι S → EntryAgree Ch sel mu cst σ₀ P ι S ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IdentityCosetEntryProportionalColumnDatumStatement

theorem **_root_.GlobalTheory.T.identityCoset_entry_proportional_columnDatum** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (S : SymData k s R.k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → AtType Ch sel mu cst σ₀ P ι S → EntryAgree Ch sel mu cst σ₀ P ι S ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IkedaMapSchrodingerPartialIntegral

theorem **trace_incA** (x : D.host.E) : Algebra.trace (AF D.host.F) (AE D) (incA D x) = algebraMap D.host.F (AF D.host.F) (Algebra.trace D.host.F D.host.E x)

theorem **trace_rat** {ρ : ℕ} (M : Matrix (Fin ρ) (Fin ρ) D.host.E) : (rat M : Mx D ρ ρ).trace = incA D M.trace

theorem **chi_add** {k ρ : ℕ} (M : Matrix (Fin k) (Fin ρ) D.host.E) (z z' : Mx D k ρ) : chi ψ M (z + z') = chi ψ M z * chi ψ M z'

theorem **chi_rat** (hψ : IsCharAF D.host.F ψ) {k ρ : ℕ} (M γ : Matrix (Fin k) (Fin ρ) D.host.E) : chi ψ M (rat γ) = 1

theorem **exists_chi_ne** (hψ : IsCharAF D.host.F ψ) {k ρ : ℕ} {M : Matrix (Fin k) (Fin ρ) D.host.E} (hM : M ≠ 0) : ∃ z, chi ψ M z ≠ 1

theorem **integral_chi** (hψ : IsCharAF D.host.F ψ) {k ρ : ℕ} (M : Matrix (Fin k) (Fin ρ) D.host.E) {μ : Measure (Mx D k ρ)} {𝓕 : Set (Mx D k ρ)} (hμ : IsCovolOne (ratMx D k ρ) μ 𝓕) : ∫ z in 𝓕, chi ψ M z ∂μ = if M = 0 then 1 else 0

theorem **continuous_chi** (hψ : IsCharAF D.host.F ψ) {k ρ : ℕ} (T : HSp D) {S : HSp D} (e : HypDec S (zeroSp D) ρ) (J : Matrix (Fin T.rank) (Fin k) D.host.E) (η : Matrix (Fin T.rank) (Fin ρ) D.host.E) : (∀ z : Mx D k ρ, ψ (symplO S T (e.extX T (rat η)) (e.extY T (rat J · z))) = chi ψ (Jᴴ ∗ T.gram ∗ η) z) ∧ Continuous (chi ψ (Jᴴ ∗ T.gram ∗ η))

theorem **countable_mat** (m n : ℕ) : Countable (Matrix (Fin m) (Fin n) D.host.E)

theorem **integral_series** (hψ : IsCharAF D.host.F ψ) {k ρ : ℕ} {T S : HSp D} (e : HypDec S (zeroSp D) ρ) (J : Matrix (Fin T.rank) (Fin k) D.host.E) {μ : Measure (Mx D k ρ)} {𝓕 : Set (Mx D k ρ)} (hμ : IsCovolOne (ratMx D k ρ) μ 𝓕) (c : Matrix (Fin T.rank) (Fin ρ) D.host.E → ℂ) (hc : Summable fun η => ‖c η‖) : ∫ (z : Mx D k ρ) in 𝓕, ∑' η, ψ (symplO S T (e.extX T (rat η)) (e.extY T (rat J · z))) ∗ c η ∂μ = ∑' η, if Jᴴ ∗ T.gram ∗ η = 0 then c η else 0

theorem **sFinite_of_prob** {X : Type} [MeasurableSpace X] (μ : Measure X) [IsProbabilityMeasure μ] : SFinite μ

theorem **continuous_mean** {X Y : Type} [TopologicalSpace X] [TopologicalSpace Y] [MeasurableSpace X] [BorelSpace X] [SecondCountableTopology X] [SecondCountableTopology Y] (μ : Measure X) [IsProbabilityMeasure μ] (F : X × Y → ℂ) (hF : Continuous F) (C : ℝ) (hC : ∀ ρ, ‖F ρ‖ ≤ C) : (Continuous fun y => ∫ x, F (x, y) ∂μ) ∧ ∀ y, ‖∫ x, F (x, y) ∂μ‖ ≤ C

theorem **integral_halves** {t ρ : ℕ} {S : HSp D} (e : HypDec S (zeroSp D) ρ) {μ₁ : Measure (Mx D t S.rank)} {𝓕₁ : Set (Mx D t S.rank)} (h₁ : IsCovolOne (ratMx D t S.rank) μ₁ 𝓕₁) {μ : Measure (Mx D t ρ)} {𝓕 : Set (Mx D t ρ)} (h₂ : IsCovolOne (ratMx D t ρ) μ 𝓕) (h : Mx D t S.rank → ℂ) (hc : Continuous h) (C : ℝ) (hC : ∀ x, ‖h x‖ ≤ C) (hp : ∀ γ ∈ ratMx D t S.rank, ∀ x, h (γ + x) = h x) : ∫ ξ in 𝓕₁, h ξ ∂μ₁ = ∫ u in 𝓕, ∫ v in 𝓕, h (u · e.pX + v · e.pY) ∂μ ∂μ

theorem **integral_unfold** {m n : ℕ} {μ : Measure (Mx D m n)} {𝓕 : Set (Mx D m n)} (hμ : IsCovolOne (ratMx D m n) μ 𝓕) (P : Mx D m n → ℂ) (hc : Continuous P) (hP : Integrable P μ) : ∫ u in 𝓕, ∑' γ : Matrix (Fin m) (Fin n) D.host.E, P (rat γ + u) ∂μ = ∫ x, P x ∂μ

theorem **integrable_slice** (hψ : IsStdChar D ψ (-1)) (v : GlobalTheory.Unf.VolM D T.rank ρ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {μ : Measure (Mx D t ρ)} {𝓕 : Set (Mx D t ρ)} (hμ : IsCovolOne (ratMx D t ρ) μ 𝓕) : Continuous (fun x : Mx D t ρ => e.mixS T ψ v.μ v.𝓕 f (d.jX · x) 0) ∧ Integrable (fun x : Mx D t ρ => e.mixS T ψ v.μ v.𝓕 f (d.jX · x) 0) μ

theorem **hasSum_recon** (hψ : IsStdChar D ψ (-1)) (v : GlobalTheory.Unf.VolM D T.rank ρ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (x y : Mx D T.rank ρ) : HasSum (fun η : Matrix (Fin T.rank) (Fin ρ) D.host.E => ψ (symplO S T (e.extX T (rat η + x)) (e.extY T y)) ∗ e.mixS T ψ v.μ v.𝓕 f (rat η + x) 0) (ψ (symplOH S T (e.extX T x) (e.extY T y)) ∗ f (e.extX T x + e.extY T y))

theorem **symplO_add_right** (x y y' : WO S T) : symplO S T x (y + y') = symplO S T x y + symplO S T x y'

theorem **iso_pair** (ξ ξ' : Mx D t S.rank) (w : WO S T') : symplOH S T (d.jX · ξ) (d.jX · ξ' + d.j0 · w) = 0

theorem **tsum_reindex** (G : Matrix (Fin T.rank) (Fin ρ) D.host.E → ℂ) : (∑' η, if d.eᴴ ∗ T.gram ∗ η = 0 then (if d.i₀ᴴ ∗ T.gram ∗ η = 0 then G η else 0) else 0) = ∑' γ : Matrix (Fin t) (Fin ρ) D.host.E, G (d.e ∗ γ)

theorem **inner_mean** (hψ : IsStdChar D ψ (-1)) (v : GlobalTheory.Unf.VolM D T.rank ρ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {μ : Measure (Mx D t ρ)} {𝓕 : Set (Mx D t ρ)} (hμ : IsCovolOne (ratMx D t ρ) μ 𝓕) {ν : Measure (Mx D T'.rank ρ)} {𝒴 : Set (Mx D T'.rank ρ)} (hν : IsCovolOne (ratMx D T'.rank ρ) ν 𝒴) (u : Mx D t ρ) : ∫ (z : Mx D t ρ) in 𝓕, ∫ (b : Mx D T'.rank ρ) in 𝒴, f (d.jX · (u · e.pX + z · e.pY) + d.j0 · (b · e.pY)) ∂ν ∂μ = ∑' γ : Matrix (Fin t) (Fin ρ) D.host.E, e.mixS T ψ v.μ v.𝓕 f (d.jX · (rat γ + u)) 0

theorem **at_zero** (hψ : IsStdChar D ψ (-1)) (v : GlobalTheory.Unf.VolM D T.rank ρ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {μ₁ : Measure (Mx D t S.rank)} {𝓕₁ : Set (Mx D t S.rank)} (h₁ : IsCovolOne (ratMx D t S.rank) μ₁ 𝓕₁) {μ : Measure (Mx D t ρ)} {𝓕 : Set (Mx D t ρ)} (hμ : IsCovolOne (ratMx D t ρ) μ 𝓕) {ν : Measure (Mx D T'.rank ρ)} {𝒴 : Set (Mx D T'.rank ρ)} (hν : IsCovolOne (ratMx D T'.rank ρ) ν 𝒴) : e.mixS T' ψ ν 𝒴 (ikeda ψ d μ₁ 𝓕₁ f) 0 0 = ∫ x : Mx D t ρ, e.mixS T ψ v.μ v.𝓕 f (d.jX · x) 0 ∂μ

theorem **sch_ikeda** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T T' : HSp D) (t : ℕ) (d : HypDec T T' t) (ν : Measure (WO A T)) (𝒴 : Set (WO A T)), IsCovolOne (ratL T A) ν 𝒴 → ∀ (ν' : Measure (WO A T')) (𝒴' : Set (WO A T')), IsCovolOne (ratL T' A) ν' 𝒴' → ∀ (μ₁ : Measure (Mx D t (dblSp A).rank)) (𝓕₁ : Set (Mx D t (dblSp A).rank)), IsCovolOne (ratMx D t (dblSp A).rank) μ₁ 𝓕₁ → ∀ (μ₂ : Measure (Mx D t A.rank)) (𝓕₂ : Set (Mx D t A.rank)), IsCovolOne (ratMx D t A.rank) μ₂ 𝓕₂ → ∀ φ, IsThetaFunO ψ (dblSp A) T φ → IsThetaFunO ψ (dblSp A) T' (ikeda ψ d μ₁ 𝓕₁ φ) ∧ sch ψ A T' ν' 𝒴' (ikeda ψ d μ₁ 𝓕₁ φ) = ikS d μ₂ (sch ψ A T ν 𝒴 φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IkedaMapSchrodingerPartialIntegralStatement

theorem **_root_.GlobalTheory.T.ikedaMap_schrodinger_partialIntegral** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A T T' : HSp D) (t : ℕ) (d : HypDec T T' t) (ν : Measure (WO A T)) (𝒴 : Set (WO A T)), IsCovolOne (ratL T A) ν 𝒴 → ∀ (ν' : Measure (WO A T')) (𝒴' : Set (WO A T')), IsCovolOne (ratL T' A) ν' 𝒴' → ∀ (μ₁ : Measure (Mx D t (dblSp A).rank)) (𝓕₁ : Set (Mx D t (dblSp A).rank)), IsCovolOne (ratMx D t (dblSp A).rank) μ₁ 𝓕₁ → ∀ (μ₂ : Measure (Mx D t A.rank)) (𝓕₂ : Set (Mx D t A.rank)), IsCovolOne (ratMx D t A.rank) μ₂ 𝓕₂ → ∀ φ, IsThetaFunO ψ (dblSp A) T φ → IsThetaFunO ψ (dblSp A) T' (ikeda ψ d μ₁ 𝓕₁ φ) ∧ sch ψ A T' ν' 𝒴' (ikeda ψ d μ₁ 𝓕₁ φ) = ikS d μ₂ (sch ψ A T ν 𝒴 φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IkedaPartialFourierIntegralMemAdelicSchwartz

theorem **exists_split** {D : D7} {θ : InfFam D} (hθ : IsInfFam θ) (m n : ℕ) : ∃ e : Mx D m n ≃ₜ+ (D.what₂.Tau → Fin m → Fin n → ℂ) × (Fin m → Fin n → AEf D), ∀ x, e x = (fun τ i j => compAt θ τ x i j, fun i j => finPart D (x i j))

theorem **continuous_lmul** {R : Type} [CommRing R] [TopologicalSpace R] [IsTopologicalRing R] {m t : ℕ} (n : ℕ) (J : Matrix (Fin m) (Fin t) R) : Continuous (lmul n J)

theorem **map_emb** (x : Mx D t n) (a : Mx D T'.rank n) : (fun i j => φ ((d.jX · x + d.j0 · a : Mx D T.rank n) i j)) = lmul n (d.jX.map φ) (fun i j => φ (x i j)) + lmul n (d.j0.map φ) (fun i j => φ (a i j))

theorem **pX_emb** (x : Fin t → Fin n → R) (a : Fin T'.rank → Fin n → R) : lmul n (d.pX.map φ) (lmul n (d.jX.map φ) x + lmul n (d.j0.map φ) a) = x

theorem **wittHyperbolic_projAnisotropic_leftMul_embedding** (x : Fin t → Fin n → R) (a : Fin T'.rank → Fin n → R) : lmul n (d.p0.map φ) (lmul n (d.jX.map φ) x + lmul n (d.j0.map φ) a) = a

theorem **map_emb_fin** (x : Mx D t n) (a : Mx D T'.rank n) : (fun i j => finPart D ((d.jX · x + d.j0 · a : Mx D T.rank n) i j)) = lmul n (d.jX.map (finPart D)) (fun i j => finPart D (x i j)) + lmul n (d.j0.map (finPart D)) (fun i j => finPart D (a i j))

theorem **pure** {θ : InfFam D} (hθ : IsInfFam θ) (μ : Measure (Mx D t n)) [μ.IsAddHaarMeasure] (f : SchwartzMap (D.what₂.Tau → Fin T.rank → Fin n → ℂ) ℂ) (g : (Fin T.rank → Fin n → AEf D) → ℂ) (hg : g ∈ WeilM.SB (Fin T.rank → Fin n → AEf D)) (Φ : Mx D T.rank n → ℂ) (hΦ : Φ = fun x => f (fun τ i j => compAt θ τ x i j) ∗ g (fun i j => finPart D (x i j))) : (∀ a : Mx D T'.rank n, Integrable (fun x : Mx D t n => Φ (d.jX · x + d.j0 · a)) μ) ∧ ikS d μ Φ ∈ schwartzA θ T'.rank n

theorem **ikS_mem_schwartzA** : ∀ (D : D7), ∀ θ : InfFam D, IsInfFam θ → ∀ (T T' : HSp D) (t n : ℕ) (d : HypDec T T' t) (μ : Measure (Mx D t n)) (ℱ : Set (Mx D t n)), IsCovolOne (ratMx D t n) μ ℱ → ∀ Φ ∈ schwartzA θ T.rank n, ikS d μ Φ ∈ schwartzA θ T'.rank n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IkedaPartialFourierIntegralMemAdelicSchwartzStatement

theorem **_root_.GlobalTheory.T.ikedaPartialFourierIntegral_mem_adelicSchwartz** : ∀ (D : D7), ∀ θ : InfFam D, IsInfFam θ → ∀ (T T' : HSp D) (t n : ℕ) (d : HypDec T T' t) (μ : Measure (Mx D t n)) (ℱ : Set (Mx D t n)), IsCovolOne (ratMx D t n) μ ℱ → ∀ Φ ∈ schwartzA θ T.rank n, ikS d μ Φ ∈ schwartzA θ T'.rank n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.Imports

theorem **exists_prime_mem** {A : Type} [inst : CommRing A] (P : Ideal A) (hP : Ideal.IsPrime P) : ∀ (n : ℕ), n ≠ (0 : ℕ) → (↑n : A) ∈ P → ∃ (ℓ : ℕ), Nat.Prime ℓ ∧ (↑ℓ : A) ∈ P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ImportsBis

theorem **weilRepresentationUnitaryPair** : GlobalTheory.Published.WeilRepresentationUnitaryPair

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ImportsTer

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (C : WO S T →₀ ℂ) (f : WO S T → ℂ), IsThetaFunO ψ S T f → IsThetaFunO ψ S T (comb ψ S T C f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InConeOfKTypesAbove

theorem **inCone_of_kTypesAbove** : GlobalTheory.Dec.KTypesAbove → ∀ (p q n : ℕ) (e : Ix p q ≃ Fin n) (P : SignedPar n) (T : GrpData p q) (R : DSRep T) (σ : Matrix (Fin n) (Fin n) ℂ → (R.U.H →L[ℂ] R.U.H)), BlockOrd e P.sgn → Factors R.U e σ → IsDSPar P σ → ∀ (μ : Ix p q → ℚ) v, IsHW R.HK.act μ v → InCone (blockPar P e) μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InLatticeStabilizerAtPlaceOfSupportedStabilizer

theorem **evalV_sv_same** (v : Pl D) : evalV D.host.F v (singleV D.host.F v 1) = 1

theorem **evalV_sv_ne** {v w : Pl D} (h : w ≠ v) : evalV D.host.F w (singleV D.host.F v 1) = 0

theorem **intAf_mul** {a b : Af D.host.F} (ha : a ∈ intAf D.host.F) (hb : b ∈ intAf D.host.F) : a * b ∈ intAf D.host.F

theorem **sv_int** (v : Pl D) : singleV D.host.F v 1 ∈ intAf D.host.F

theorem **one_sub_sv_int** (v : Pl D) : 1 - singleV D.host.F v 1 ∈ intAf D.host.F

theorem **compFin_e_same** (v : Pl D) : compFin D v (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) = 1

theorem **compFin_e_ne** {v w : Pl D} (h : w ≠ v) : compFin D w (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) = 0

theorem **infPart_pair** (x : Af D.host.F) (c : D.host.E) : infPart D (pairA D.host.F 0 x ⊗ₜ[D.host.F] c) = 0

theorem **finPart_pair** (x : Af D.host.F) (c : D.host.E) : finPart D (pairA D.host.F 0 x ⊗ₜ[D.host.F] c) = x ⊗ₜ[D.host.F] c

theorem **exists_fin_lift** (b : Af D.host.F ⊗[D.host.F] D.host.E) : ∃ a : AE D, infPart D a = 0 ∧ finPart D a = b

theorem **fin_mem_latHat** (S : HSp D) (L : Lat S) (P : Pl D → Prop) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (fun i => finPart D (y i)) ∈ latHat D.host.F D.host.E (L : Set (Fin S.rank → D.host.E))

theorem **latHat_smul** {n : ℕ} (Ls : Set (Fin n → D.host.E)) {t : Af D.host.F} (ht : t ∈ intAf D.host.F) {y : Fin n → Af D.host.F ⊗[D.host.F] D.host.E} (hy : y ∈ latHat D.host.F D.host.E Ls) : (t ⊗ₜ[D.host.F] (1 : D.host.E)) • y ∈ latHat D.host.F D.host.E Ls

theorem **exists_lift_latV** (S : HSp D) (L : Lat S) (v : Pl D) {z : Fin S.rank → Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ latHat D.host.F D.host.E (L : Set (Fin S.rank → D.host.E))) : ∃ y ∈ latV S L (· = v), (∀ i, infPart D (y i) = 0) ∧ ∀ i, finPart D (y i) = (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) * z i

theorem **e_mul_of_supp** (v : Pl D) (X : Af D.host.F ⊗[D.host.F] D.host.E) (h : ∀ w, w ≠ v → compFin D w X = 0) : (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) * X = X

theorem **e_idem** (v : Pl D) : (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) * (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) = singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)

theorem **one_sub_e** (v : Pl D) : 1 - singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E) = (1 - singleV D.host.F v 1) ⊗ₜ[D.host.F] (1 : D.host.E)

theorem **entries_mem** {S : HSp D} {P : Pl D → Prop} {κ : GA D S} (hg : κ ∈ grpAt S P) (i j : Fin S.rank) : mt S κ i j - (1 : Matrix (Fin S.rank) (Fin S.rank) (AE D)) i j ∈ idAt D P

theorem **keeps_of_core** (v : Pl D) {n : ℕ} (Ls : Set (Fin n → D.host.E)) (M : Matrix (Fin n) (Fin n) (Af D.host.F ⊗[D.host.F] D.host.E)) (hN : ∀ i j, (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) * (M i j - (1 : Matrix (Fin n) (Fin n) _) i j) = M i j - (1 : Matrix (Fin n) (Fin n) _) i j) (hcore : ∀ z, (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) • z = z → (z ∈ latHat D.host.F D.host.E Ls ↔ M *ᵥ z ∈ latHat D.host.F D.host.E Ls)) (y : Fin n → Af D.host.F ⊗[D.host.F] D.host.E) : y ∈ latHat D.host.F D.host.E Ls ↔ M *ᵥ y ∈ latHat D.host.F D.host.E Ls

theorem **mem_latHat_iff_lift** (S : HSp D) (L : Lat S) (v : Pl D) (z : Fin S.rank → Af D.host.F ⊗[D.host.F] D.host.E) (hz : (singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) • z = z) (y : Fin S.rank → AE D) (h1 : ∀ i, infPart D (y i) = 0) (h2 : ∀ i, finPart D (y i) = z i) : z ∈ latHat D.host.F D.host.E (L : Set (Fin S.rank → D.host.E)) ↔ y ∈ latV S L (· = v)

theorem **inStabAt_of_inK** (D : D7) (S : HSp D) (L : Lat S) (v : Pl D) (κ : GA D S) (_hL : IsSelfDualC S L) (hκ : InK S L (· = v) κ) : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.InStabAt S L v κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InLatticeStabilizerAtPlaceOfSupportedStabilizerStatement

theorem **_root_.GlobalTheory.T.inLatticeStabilizerAtPlace_of_supportedStabilizer** : ∀ (D : D7) (S : HSp D) (L : Lat S) (v : Pl D) (κ : GA D S), IsSelfDualC S L → InK S L (· = v) κ → GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.InStabAt S L v κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InLatticeStabilizerAtPlacesIffPlaceSetProd

theorem **idAt_mono** {P P' : Pl D → Prop} (h : ∀ w, P w → P' w) {a : AE D} (ha : a ∈ idAt D P) : a ∈ idAt D P'

theorem **latV_mono** (V : HSp D) (L : Lat V) {P P' : Pl D → Prop} (h : ∀ w, P w → P' w) : latV V L P ≤ latV V L P'

theorem **mul_mem_idAt** {P R : Pl D → Prop} {c : AE D} (hc : ∀ w, ¬ R w → compAt D w c = 0) {a : AE D} (ha : a ∈ idAt D P) : c * a ∈ idAt D (fun p => P p ∧ R p)

theorem **smul_mem_latV** (V : HSp D) (L : Lat V) {P R : Pl D → Prop} (β : Af D.host.F) (hβ : β ∈ intAf D.host.F) (h0 : ∀ v, ¬ R v → evalV D.host.F v β = 0) {y : Fin V.rank → AE D} (hy : y ∈ latV V L P) : algebraMap (AF D.host.F) (AE D) (pairA D.host.F 0 β) • y ∈ latV V L (fun p => P p ∧ R p)

theorem **indQ_int** (Q : Finset (Pl D)) : indQ D Q ∈ intAf D.host.F

theorem **indQc_int** (Q : Finset (Pl D)) : 1 - indQ D Q ∈ intAf D.host.F

theorem **eQ_off** (Q : Finset (Pl D)) (w : Pl D) (hw : ¬ w ∈ Q) : compAt D w (eQ Q) = 0

theorem **one_sub_eQ_off** (Q : Finset (Pl D)) (w : Pl D) (hw : ¬ w ∉ Q) : compAt D w (1 - eQ Q) = 0

theorem **eQ_mul_of_in** {P : Pl D → Prop} (Q : Finset (Pl D)) {a : AE D} (ha : a ∈ idAt D (fun p => P p ∧ p ∈ Q)) : eQ Q * a = a

theorem **eQ_mul_of_out** {P : Pl D → Prop} (Q : Finset (Pl D)) {a : AE D} (ha : a ∈ idAt D (fun p => P p ∧ p ∉ Q)) : eQ Q * a = 0

theorem **one_sub_mul_eq** {P : Pl D → Prop} (Q : Finset (Pl D)) {a : AE D} (ha : a ∈ idAt D P) : (1 - eQ Q) * a = algebraMap (AF D.host.F) (AE D) (idemQc D Q) * a

theorem **mem_latV_iff_parts** (V : HSp D) (L : Lat V) (P : Pl D → Prop) (Q : Finset (Pl D)) {y : Fin V.rank → AE D} (hy : ∀ i, y i ∈ idAt D P) : y ∈ latV V L P ↔ eQ Q • y ∈ latV V L (fun p => P p ∧ p ∈ Q) ∧ (1 - eQ Q) • y ∈ latV V L (fun p => P p ∧ p ∉ Q)

theorem **mt_mul** (V : HSp D) (a b : GA D V) : mt V (a * b) = mt V a * mt V b

theorem **mt_inj** (V : HSp D) {a b : GA D V} (h : mt V a = mt V b) : a = b

theorem **mulVec_mem** (V : HSp D) {I : Ideal (AE D)} (M : Matrix (Fin V.rank) (Fin V.rank) (AE D)) {y : Fin V.rank → AE D} (hy : ∀ i, y i ∈ I) (i : Fin V.rank) : (M *ᵥ y) i ∈ I

theorem **cut_sub_one** (c : A) (M : Matrix n n A) : c • M + (1 - c) • (1 : Matrix n n A) - 1 = c • (M - 1)

theorem **cut_mulVec** (c : A) (M : Matrix n n A) (y : n → A) (hy : c • y = y) : (c • M + (1 - c) • (1 : Matrix n n A)) *ᵥ y = M *ᵥ y

theorem **cut_mul_cut** (c : A) (hc : c * c = c) (M : Matrix n n A) : (c • M + (1 - c) • (1 : Matrix n n A)) * ((1 - c) • M + (1 - (1 - c)) • (1 : Matrix n n A)) = M

theorem **prod_parts** (c : A) (a b : Matrix n n A) (ha : c • (a - 1) = a - 1) (hb : c • (b - 1) = 0) (y : n → A) : c • ((a * b) *ᵥ y) = a *ᵥ (c • y) ∧ (1 - c) • ((a * b) *ᵥ y) = b *ᵥ ((1 - c) • y)

theorem **inK_cut** (V : HSp D) (L : Lat V) {P R : Pl D → Prop} (ε : AF D.host.F) (hε : ε * ε = ε) (hoff : ∀ w, ¬ R w → compAt D w (algebraMap (AF D.host.F) (AE D) ε) = 0) (hone : ∀ a ∈ idAt D (fun p => P p ∧ R p), algebraMap (AF D.host.F) (AE D) ε * a = a) (hlat : ∀ y : Fin V.rank → AE D, (∀ i, y i ∈ idAt D P) → y ∈ latV V L P → algebraMap (AF D.host.F) (AE D) ε • y ∈ latV V L (fun p => P p ∧ R p)) {κ : GA D V} (hκ : InK V L P κ) : InK V L (fun p => P p ∧ R p) (cutG V ε hε κ)

theorem **inK_iff_prod** : ∀ (D : D7) (V : HSp D) (L : Lat V) (P : Pl D → Prop) (Q : Finset (Pl D)) (κ : GA D V), InK V L P κ ↔ ∃ a b, InK V L (fun p => P p ∧ p ∈ Q) a ∧ InK V L (fun p => P p ∧ p ∉ Q) b ∧ κ = a * b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InLatticeStabilizerAtPlacesIffPlaceSetProdStatement

theorem **_root_.GlobalTheory.T.inLatticeStabilizerAtPlaces_iff_placeSet_prod** : ∀ (D : D7) (V : HSp D) (L : Lat V) (P : Pl D → Prop) (Q : Finset (Pl D)) (κ : GA D V), InK V L P κ ↔ ∃ a b, InK V L (fun p => P p ∧ p ∈ Q) a ∧ InK V L (fun p => P p ∧ p ∉ Q) b ∧ κ = a * b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InLatticeStabilizerAtPlacesMulOfMem

theorem **inK_mul** {D : D7} (S : HSp D) (L : Lat S) (P : Pl D → Prop) (a b : GA D S) : InK S L P a → InK S L P b → InK S L P (a * b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InLatticeStabilizerAtPlacesMulOfMemStatement

theorem **_root_.GlobalTheory.T.inLatticeStabilizerAtPlaces_mul_of_mem** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop) (a b : GA D S), InK S L P a → InK S L P b → InK S L P (a * b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InSuppMulLeft

theorem **inSupp_mul_left** {x : Matrix (A ⊕ D') (A ⊕ D) k} (hx : InSupp x) {j : GL (A ⊕ D') k} (hj : j ∈ parahoric k A D') : InSupp (j.val * x) ∧ valuation k ((j.val * x).toBlocks₁₁.det - j.val.toBlocks₁₁.det * x.toBlocks₁₁.det) < 1

theorem **omTwo_datum_inv** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D D' : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] [Fintype D'] [DecidableEq D'] (ξ μS μT : kˣ →* ℂˣ), IsTame ξ → IsUnram (μS * ξ⁻¹) → ∀ j ∈ parahoric k A D', ∀ d : kˣ, (d : k) = (j : Matrix (A ⊕ D') (A ⊕ D') k).toBlocks₂₂.det → omTwo μS μT (1 : GL (A ⊕ D) k) j (datum ξ⁻¹ : Matrix (A ⊕ D') (A ⊕ D) k → ℂ) = (ξ d : ℂ) • (datum ξ⁻¹ : Matrix (A ⊕ D') (A ⊕ D) k → ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IncAMulMemIntAt

theorem **incA_mul_mem_intAt** {v : Pl D} {c : D.host.E} {N : ℤ} (hN : (N : 𝒪 D.host.F) ∉ v.asIdeal) (hint : IsIntegral ℤ (N • c)) : ∀ a ∈ intAt D (· = v), incA D c * a ∈ intAt D (· = v)

theorem **latA_eq_latV** {v : Pl D} {N M : ℕ} (hvN : (N : 𝒪 D.host.F) ∉ v.asIdeal) (hvM : (M : 𝒪 D.host.F) ∉ v.asIdeal) (hNI : ∀ a ∈ D.host.AA, ∃ b : 𝒪 D.host.E, algebraMap _ _ b = a * N) (hMI : (M : D.host.E) ∈ D.host.AA) (S : HSp D) (L : Lat S) : latA S L (· = v) = latV S L (· = v)

theorem **notMem_of_coprime** {A : Type} [CommRing A] [IsDedekindDomain A] {v : HeightOneSpectrum A} {T N : ℕ} (hT : (T : A) ∈ v.asIdeal) (h : Nat.Coprime N T) : (N : A) ∉ v.asIdeal

theorem **unitAA_of_prime** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) {T : ℕ} (hT : T.Prime) (hv : (T : 𝒪 D.host.F) ∈ p.v.asIdeal) (h2 : T ∉ D.host.small) (hq : T ≠ D.host.q) (hd : T ≠ D.host.d) (hc : Nat.Coprime (Ideal.absNorm D.host.ccK) T) : p.UnitAA

theorem **unitAA_level** {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt (I.rec7 l Z) k) (hv : ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ p.v.asIdeal) : p.UnitAA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IncAdjYZeroMemLatOIff

theorem **latV_entry_mem'** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) (i : Fin S.rank) : y i ∈ idAt D P

theorem **incAdj_yZero_mulVec** : ∀ (D : D7) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (hρ : 0 < ρ) (a : AE D) (y : Fin B.V₀.rank → AE D), Matrix.mulVec (show Matrix (Fin J.W.rank) (Fin B.V₀.rank) (AE D) from J.d.incAdj B.V₀ (a • yZero B.V₀ ρ (vecB B.o B.ζ))) y = colMul J.d.jY ⟨ρ - 1, by omega⟩ (incA D D.host.xi0⁻¹ * cA D a * (∑ l, (∑ m, cA D (ratV (vecB B.o B.ζ) m) * gA B.V₀ m l) * y l))

theorem **cA_mem_idAt''** {v : Pl D} {y : AE D} (h : y ∈ idAt D (· = v)) : cA D y ∈ idAt D (· = v)

theorem **smul_colMul** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (i : Fin ρ) (s t : AE D) : s • colMul d.jY i t = colMul d.jY i (s * t)

theorem **colMul_jY_mem_latA_iff** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (ƀ : FI D) (v : Pl D) (i : Fin ρ), ƀ ≠ 0 → (∀ t ∈ idAt D (· = v), colMul d.jY i t ∈ latV S L (· = v) ↔ cA D t ∈ idV ƀ⁻¹ v) → ∀ t ∈ idAt D (· = v), colMul d.jY i t ∈ latA S L (· = v) ↔ cA D t ∈ idV ((D.host.AAbar : FI D) * ƀ⁻¹) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IncAdjYZeroMulVec

theorem **incAdj_yZero_mulVec** {V₀ W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (r : Fin ρ) (hr : r.1 + 1 = ρ) (z : Fin V₀.rank → D.host.E) (a : AE D) (y : Fin V₀.rank → AE D) : (show Matrix (Fin W.rank) (Fin V₀.rank) (AE D) from d.incAdj V₀ (a • yZero V₀ ρ z)) *ᵥ y = colMul d.jY r (incA D D.host.xi0⁻¹ * hermAt V₀ y (a • ratV z))

theorem **mxAt_mul** {P : Pl D → Prop} {l m n : ℕ} (M : Mx D l m) {N : Mx D m n} (h : MxAt P N) : MxAt P (M · N)

theorem **mxAt_mul'** {P : Pl D → Prop} {l m n : ℕ} {M : Mx D l m} (N : Mx D m n) (h : MxAt P M) : MxAt P (M · N)

theorem **mxAt_smul** {P : Pl D → Prop} {m n : ℕ} (a : AE D) {N : Mx D m n} (h : MxAt P N) : MxAt P (a • N)

theorem **mxAt_adj** {x : Pl D} {m n : ℕ} {N : Mx D m n} (h : MxAt (· = x) N) : MxAt (· = x) (adj N)

theorem **cA_xi** : cA D (incA D D.host.xi0) = -incA D D.host.xi0

theorem **cA_xi_inv** : cA D (incA D D.host.xi0⁻¹) = -incA D D.host.xi0⁻¹

theorem **last_of** {V₀ W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (z : Fin V₀.rank → D.host.E) {M : Lat V₀} {LW : Lat W} (hM : IsSelfDualC V₀ M) (hLW : IsSelfDualC W LW) {x : Pl D} {α m : (AE D)ˣ} {b : Fin ρ → (AE D)ˣ} (hb : ∀ r : Fin ρ, r.1 + 1 = ρ → b r = xiU D * cU α * m) (hα : ∀ y, y ∈ latA W LW (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV W LW (· = x)) (hW : ∀ i, ∀ a ∈ idAt D (· = x), colMul d.jY i a ∈ latV W LW (· = x) ↔ cA D (b i : AE D) * a ∈ intAt D (· = x)) (hz : ∀ a ∈ idAt D (· = x), a • ratV z ∈ latV V₀ M (· = x) ↔ ((m⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) (hρ : 0 < ρ) : ∀ a ∈ idAt D (· = x), d.incAdj V₀ (a • yZero V₀ ρ z) ∈ latO V₀ W (M : Set _) (LW : Set _) (· = x) ↔ a ∈ intAt D (· = x)

theorem **herm_kap** {L L'' S : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (w : WO S L'') (l : Fin L.rank → AE D) (s : Fin S.rank → AE D) : hermAt S ((show Matrix (Fin S.rank) (Fin L.rank) (AE D) from kap τ S w) *ᵥ l) s = incA D (-D.host.xi0⁻¹) * hermAt L l ((rat τ : Mx D _ _) *ᵥ ((show Matrix (Fin L''.rank) (Fin S.rank) (AE D) from w) *ᵥ s))

theorem **kap_of** {L L'' S : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) {ζ : Fin L.rank → D.host.E} {ζ₀ : Fin L''.rank → D.host.E} (hζ : IsLineVec L ζ) (hζ₀ : IsLineVec L'' ζ₀) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : τ *ᵥ ζ₀ = lam • ζ) {LL : Lat L} {LZ : Lat L''} {LS : Lat S} (hLL : IsSelfDualC L LL) (hLZ : IsSelfDualC L'' LZ) (hLS : IsSelfDualC S LS) {x : Pl D} {α μ : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hμ : ∀ a ∈ idAt D (· = x), a • ratV ζ₀ ∈ latV L'' LZ (· = x) ↔ ((μ⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) (hm : ∀ a ∈ idAt D (· = x), a • ratV ζ ∈ latV L LL (· = x) ↔ (((α * μ * inU lam hlam * (xiU D * cU α)⁻¹)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : ∀ w ∈ latO S L'' (LS : Set _) (LZ : Set _) (· = x), kap τ S w ∈ latO L S (LL : Set _) (LS : Set _) (· = x)

theorem **isU_xi** (h : p.UnitAA) : IsU p.v (xiU D : AE D)

theorem **isU_alpha** (h : p.UnitAA) {L : HSp D} {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = p.v) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = p.v)) : IsU p.v (α : AE D)

theorem **val_le_one** {u : AE D} (h : IsU p.v u) : valuation k (p.pr u) ≤ 1

theorem **val_eq_one** {u : (AE D)ˣ} (h : IsU p.v (u : AE D)) : valuation k (p.pr u) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexComapEqOfCosetsHit

theorem **index_comap_eq_of_cosets_hit** {A B : Type} [Group A] [Group B] (f : A →* B) (H : Subgroup B) (h : ∀ b : B, ∃ a : A, (f a)⁻¹ * b ∈ H) : (H.comap f).index = H.index

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexEquivalenceReduction

theorem **mem_iter_gen** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {R t : Type} [Fintype R] [DecidableEq R] [Fintype t] [DecidableEq t] (P : TypeDatum k R) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) (g : GL (Ix R t j) k) : g ∈ (P.iter t μ hμ j).J ↔ g ∈ KO k (Ix R t j) ∧ (∀ a b : Ix R t j, b.blk < a.blk → valuation k (g.val a b) < 1) ∧ ∃ x ∈ P.J, x.val = g.val.submatrix Sum.inl Sum.inl

theorem **isIxR_of_isIx** {e₀ : R ≃ Fin r} {ea : Ix R (Fin t) j ≃ Fin a} (h : GlobalTheory.FunObj.Comparison₂.IsIx ((Equiv.sumCongr e₀.symm (Equiv.refl (Fin j × Fin t))).trans ea)) : IsIxR e₀ ea

theorem **IsIxR.blkOf** {e₀ : R ≃ Fin r} {ea : Ix R (Fin t) j ≃ Fin a} (h : IsIxR e₀ ea) (i : Ix R (Fin t) j) : Formalism.L2.DIV.blkOf r t (ea i) = i.blk

theorem **val_succ_inl** (ha : IsIxR e₀ ea) (hb : IsIxR e₀ eb) (i : Ix R (Fin t) j) : (eb (ixSucc _ _ j (.inl i))).val = (ea i).val

theorem **topL_a** {κ : Type} [Zero κ] (ha : IsIxR e₀ ea) (hb : IsIxR e₀ eb) (y : Matrix (Fin b) (Fin a) κ) (i l : Ix R (Fin t) j) : topL a y (ea i) (ea l) = y (eb (ixSucc _ _ j (.inl i))) (ea l)

theorem **lowZero_a** (h : GlobalTheory.FunObj.Comparison₂.IsRed φ ψ) (ha : IsIxR e₀ ea) (hb : IsIxR e₀ eb) (N : Matrix (Fin b) (Fin a) S) : LowZero a (N.map φ) ↔ MaxM (((N.map ψ).submatrix eb ea).submatrix (ixSucc _ _ j) id).toRows₂

theorem **diagonalBlock_det_blockTriangular** (ha : IsIxR e₀ ea) {g : GL (Fin a) κ} (hg : g ∈ Formalism.L2.DIV.blockTri (Formalism.L2.DIV.blkOf r t)) (c : Fin j) : (Formalism.L2.DIV.dBlk (Formalism.L2.DIV.blkOf r t) (c.val + 1) g : κ) = (Ix.diagonalBlockOfMatrix (g.val.submatrix ea ea) c).det

theorem **firstBlock_topBlockMatrix_reduction_eq** (ha : IsIxR e₀ ea) (hb : IsIxR e₀ eb) (hr : r ≤ a) (N : Matrix (Fin b) (Fin a) S) {g : GL (Fin a) κ} (hg : g.val = topL a (N.map φ)) (hbt : g ∈ Formalism.L2.DIV.blockTri (Formalism.L2.DIV.blkOf r t)) : (Formalism.L2.DIV.upperLeftBlock hr g).val.submatrix e₀ e₀ = ((N.submatrix (fun i => eb (ixSucc _ _ j (.inl i))) ea).submatrix Sum.inl Sum.inl).map φ

theorem **iterPhi_read** : ∀ (S κ k : Type) [CommRing S] [CommRing κ] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (φ : S →+* κ) (ψ : S →+* k), GlobalTheory.FunObj.Comparison₂.IsRed φ ψ → ∀ (R : Type) [Fintype R] [DecidableEq R] (r t j a b : ℕ) (e₀ : R ≃ Fin r) (ea : Ix R (Fin t) j ≃ Fin a) (eb : Ix R (Fin t) (j + 1) ≃ Fin b), GlobalTheory.FunObj.Comparison₂.IsIx ((Equiv.sumCongr e₀.symm (Equiv.refl (Fin j × Fin t))).trans ea) → GlobalTheory.FunObj.Comparison₂.IsIx ((Equiv.sumCongr e₀.symm (Equiv.refl (Fin (j + 1) × Fin t))).trans eb) → ∀ (hr : r ≤ a) (P : TypeDatum k R) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ c, IsTame (μ c)) (χ : ℕ → κˣ →* ℂˣ), (∀ c, c ≤ j → GlobalTheory.FunObj.Comparison₂.Compat φ ψ

(χ (c + 1)) (μ (c + 1))) → ∀ Rd : P.J → GL (Fin r) κ → Prop, (∀ (x : P.J) (g : GL (Fin r) κ) (M : Matrix R R S), g.val.submatrix e₀ e₀ = M.map φ → (Rd x g ↔ MaxM ((x : GL

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexIntegralMatricesInSubstitutionImage

theorem **mulRight_apply** (G : Matrix Y' Y' K) (w : Matrix Y Y' K) : mulRight Y G w = w ∗ G

theorem **det_mulRight** (G : Matrix Y' Y' K) : LinearMap.det (mulRight Y G) = G.det ^ Fintype.card Y

theorem **exists_mul_map** (H : Matrix Y Y K) (G : Matrix Y' Y' K) : ∃ B : Matrix Y Y' K →ₗ[K] Matrix Y Y' K, (∀ w, B w = H ∗ w ∗ G) ∧ LinearMap.det B = G.det ^ Fintype.card Y ∗ H.det ^ Fintype.card Y'

theorem **index_integral_image** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (gk : GL Y' k) (hk : GL Y k), Enl gk hk → ∃ R₀ : Finset (Matrix Y Y' k), (R₀.card : ℝ) = nabs k (Matrix.GeneralLinearGroup.det hk) ^ (Fintype.card Y' : ℝ) ∗ nabs k (Matrix.GeneralLinearGroup.det gk) ^ (-(Fintype.card Y : ℝ)) ∧ (∀ v ∈ R₀, ∀ r q, ((↑(hk⁻¹) : Matrix Y Y k) ∗ v ∗ (↑gk : Matrix Y' Y' k)) r q ∈ 𝒪[k]) ∧ ∀ w : Matrix Y Y' k, (∀ r q, ((↑(hk⁻¹) : Matrix Y Y k) ∗ w ∗ (↑gk : Matrix Y' Y' k)) r q ∈ 𝒪[k]) → ∃! v, v ∈ R₀ ∧ ∀ r q, (w - v) r q ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexIntegralMatricesInSubstitutionImageStatement

theorem **_root_.GlobalTheory.T.index_integralMatrices_in_substitution_image** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (gk : GL Y' k) (hk : GL Y k), Enl gk hk → ∃ R₀ : Finset (Matrix Y Y' k), (R₀.card : ℝ) = nabs k (Matrix.GeneralLinearGroup.det hk) ^ (Fintype.card Y' : ℝ) ∗ nabs k (Matrix.GeneralLinearGroup.det gk) ^ (-(Fintype.card Y : ℝ)) ∧ (∀ v ∈ R₀, ∀ r q, ((↑(hk⁻¹) : Matrix Y Y k) ∗ v ∗ (↑gk : Matrix Y' Y' k)) r q ∈ 𝒪[k]) ∧ ∀ w : Matrix Y Y' k, (∀ r q, ((↑(hk⁻¹) : Matrix Y Y k) ∗ w ∗ (↑gk : Matrix Y' Y' k)) r q ∈ 𝒪[k]) → ∃! v, v ∈ R₀ ∧ ∀ r q, (w - v) r q ∈ 𝒪[k]

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexZeroLevelCharacterEqBesselCharacter

theorem **map_conj_local** {D : D7} {c : Comm D} {v : VCom c} (G : GLoc v) {β : GL (Fin (c.V 0).rank) (Lev.EAf D)} {b : GA D (c.V 0)} {π' : Lev.EAf D →+* G.k} (hπ : ∀ a, π' (toEAf D a) = G.q.pr a) {n : GA D (c.V 0)} {q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin (G.s + 1)) G.k} (hloc : (ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) (Nat.zero_add _)) n q) (σ : Fin (c.V 0).rank → XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin (G.s + 1)) (hσ : Function.Bijective σ) (hM : MovedTo (ofSplit G.q) (c.V 0) b β (((G.FY 0 (G.s + 1) (Nat.zero_add _)).β ∗ ((G.S.T (Fin D.R.t)).bas 0 (G.s + 1) (Nat.zero_add _)).val).submatrix id σ)) : ((β⁻¹ ∗ finE (c.V 0) (b⁻¹ ∗ n ∗ b) ∗ β).val).map π' = ((((G.S.T (Fin D.R.t)).bas 0 (G.s + 1) (Nat.zero_add _))⁻¹).val ∗ q.val ∗ ((G.S.T (Fin D.R.t)).bas 0 (G.s + 1) (Nat.zero_add _)).val).submatrix σ σ

theorem **psi_red_at** {D : D7} {c : Comm D} {v : VCom c} (G : GLoc v) (PP : Pl D.host.E) (b : GA D (c.V 0)) (β : GL (Fin (c.V 0).rank) (Lev.EAf D)) (hL : GlobalTheory.LevZero.LocAt G PP b β) (hr : D.R.r ≤ (c.V 0).rank) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin (G.s + 1)) G.k) (p : ℕ) (hq : q ∈ GlobalTheory.RootEx.N G.k _ (G.s + 1) p) (hloc : (ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) (Nat.zero_add _)) n q) (hS : finE (c.V 0) (b⁻¹ ∗ n ∗ b) ∈ St β) (hblk : redFn β PP (finE (c.V 0) (b⁻¹ ∗ n ∗ b)) ∈ blockTri (blk2 D.R.r)) (z : 𝒪 D.host.E) (hz : Ideal.Quotient.mk PP.asIdeal z = edgeSum (GlobalTheory.chainEdge D.R.k) (upperLeftBlock hr (redFn β PP (finE (c.V 0) (b⁻¹ ∗ n ∗ b))))) : (G.S.ψ (((G.S.ϖ⁻¹ : G.kˣ) : G.k) ∗ (ofSplit G.q).pr (z : D.host.E)) : ℂ) = (G.S.ψ (GlobalTheory.RootEx.simpleSum q) : ℂ)

theorem **levChi_eq_psiBes** : ∀ {D : D7} {c : Comm D} {v : VCom c} (cm : GlobalTheory.ChainOnto.Com D) (st : StAt (c.V 0)) (b : GA D (c.V 0)) (G : cm.Pg → GLoc v) (h0 : 0 + 2 ∗ D.R.rS + 2 = D.R.m), GlobalTheory.LevZero.Bot cm st b G → ∀ (PP : cm.Pg) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → b⁻¹ ∗ n ∗ b ∈ levS (c.V 0) st.β st.L → levChi (c.V 0) st.β st.L (b⁻¹ ∗ n ∗ b) = psiBes c.ψ (c.bes 0 D.R.rS h0).d 1 (vecB (c.bes 0 D.R.rS h0).o (c.bes 0 D.R.rS h0).ζ) n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexZeroLevelCharacterEqBesselCharacterStatement

theorem **_root_.GlobalTheory.T.indexZeroLevelCharacter_eq_besselCharacter** : ∀ {D : D7} {c : Comm D} {v : VCom c} (cm : GlobalTheory.ChainOnto.Com D) (st : StAt (c.V 0)) (b : GA D (c.V 0)) (G : cm.Pg → GLoc v) (h0 : 0 + 2 ∗ D.R.rS + 2 = D.R.m), GlobalTheory.LevZero.Bot cm st b G → ∀ (PP : cm.Pg) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → b⁻¹ ∗ n ∗ b ∈ levS (c.V 0) st.β st.L → levChi (c.V 0) st.β st.L (b⁻¹ ∗ n ∗ b) = psiBes c.ψ (c.bes 0 D.R.rS h0).d 1 (vecB (c.bes 0 D.R.rS h0).o (c.bes 0 D.R.rS h0).ζ) n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexZeroLevelGeneratedByLocalUnipotents

theorem **GlobalTheory.LevZero.gen** : ∀ {D : D7} {c : Comm D} {v : VCom c} (cm : GlobalTheory.ChainOnto.Com D) (st : StAt (c.V 0)) (b : GA D (c.V 0)) (G : cm.Pg → GLoc v) (h0 : 0 + 2 ∗ D.R.rS + 2 = D.R.m), GlobalTheory.LevZero.Bot cm st b G → ∀ Q : GA D (c.V 0) → Prop, (∀ k ∈ levS (c.V 0) st.β st.L, ∀ k' ∈ levS (c.V 0) st.β st.L, Q k → Q k' → Q (k ∗ k')) → (∀ (PP : cm.Pg) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (c.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → b⁻¹ ∗ n ∗ b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ ∗ n ∗ b)) → (∀ x ∈ GfC (c.X 0), b⁻¹ ∗ embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS h0).o x ∗ b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ ∗ embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS h0).o x ∗ b)) → (∀ k ∈ levS (c.V 0) st.β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndexZeroLevelGeneratedByLocalUnipotentsStatement

theorem **_root_.GlobalTheory.T.indexZeroLevel_generated_by_local_unipotents** : ∀ {D : D7} {c : Comm D} {v : VCom c} (cm : GlobalTheory.ChainOnto.Com D) (st : StAt (c.V 0)) (b : GA D (c.V 0)) (G : cm.Pg → GLoc v) (h0 : 0 + 2 ∗ D.R.rS + 2 = D.R.m), GlobalTheory.LevZero.Bot cm st b G → ∀ Q : GA D (c.V 0) → Prop, (∀ k ∈ levS (c.V 0) st.β st.L, ∀ k' ∈ levS (c.V 0) st.β st.L, Q k → Q k' → Q (k ∗ k')) → (∀ (PP : cm.Pg) (n : GA D (c.V 0)) (q : GL (XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (c.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → b⁻¹ ∗ n ∗ b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ ∗ n ∗ b)) → (∀ x ∈ GfC (c.X 0), b⁻¹ ∗ embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS h0).o x ∗ b ∈ levS (c.V 0) st.β st.L → Q (b⁻¹ ∗ embA (c.bes 0 D.R.rS h0).d (c.bes 0 D.R.rS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndicatorBoxEqCharSum

theorem **exists_box_superset** {K : Set (Matrix Y Y' k)} (hK : IsCompact K) : ∃ R : kˣ, ∀ x ∈ K, x ∈ box Y Y' R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndicatorHomSelfDualComponentsSchwartzBruhat

theorem **isLattice_of_isLat_selfDual** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {p : SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : Frame p S n) (L : Lat S) (hL : IsSelfDualC S L) (Λ :

Submodule 𝒪[k] (n → k)) (h : β.IsLat L Λ) : Λ.IsLattice k

theorem **latInd_mem_SB_of_selfDual** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (ΛS : Submodule 𝒪[k] (n' → k)) (ΛT : Submodule 𝒪[k] (n → k)), α.IsLat LS ΛS → β.IsLat LT ΛT → latInd ΛS ΛT ∈ SB (Matrix n n' k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndicatorHomSelfDualComponentsSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.indicator_hom_selfDual_components_schwartzBruhat** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (S T : HSp D) (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (ΛS : Submodule 𝒪[k] (n' → k)) (ΛT : Submodule 𝒪[k] (n → k)), α.IsLat LS ΛS → β.IsLat LT ΛT → latInd ΛS ΛT ∈ SB (Matrix n n' k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndicatorLocalHomLatticeMemSchwartzBruhat

theorem **indicator_latO_mem_SB** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M : Lat S) (M' : Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (P : LocPl ψ S T M M'), Set.indicator {x | (ofSplit P.q).up P.FS P.FT x ∈ latO S T (M : Set _) (M' : Set _) (· = P.q.v)} (fun _ => (1 : ℂ)) ∈ SB (Matrix P.Y P.Y' P.k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IndicatorLocalHomLatticeMemSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.indicator_localHomLattice_mem_schwartzBruhat** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M : Lat S) (M' : Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (P : LocPl ψ S T M M'), Set.indicator {x | (ofSplit P.q).up P.FS P.FT x ∈ latO S T (M : Set _) (M' : Set _) (· = P.q.v)} (fun _ => (1 : ℂ)) ∈ SB (Matrix P.Y P.Y' P.k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InducedKFiniteFiniteDimensional

theorem **transComb_apply** {G : Type} [Group G] (K : Subgroup G) (ψ : G → ℂ) (c : K →₀ ℂ) (g : G) : transComb K ψ c g = c.sum fun k z => z * ψ (g * k)

theorem **span_domSet_le** {G : Type} [Group G] {P : Subgroup G} (δ : P →* ℝ≥0) (K : Subgroup G) (χ : P →* Circle) (t : ℂ) (φ : G → ℂ) : Submodule.span ℂ (domSet δ K χ t φ) ≤ domSub δ K χ t φ

theorem **range_transComb_le** {G : Type} [Group G] (K : Subgroup G) (φ : G → ℂ) : LinearMap.range (transComb K φ) ≤ kSpan K φ

theorem **evalForm_injective** {G : Type} [Group G] {P : Subgroup G} (δ : P →* ℝ≥0) (K : Subgroup G) (χ : P →* Circle) (t : ℂ) (φ : G → ℂ) (hPK : ∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) : Function.Injective (evalForm δ K χ t φ)

theorem **finiteDimensional_span_domSet** : ∀ (G : Type) [Group G] (P K : Subgroup G) (δ : P →* ℝ≥0) (χ : P →* Circle) (t : ℂ) (φ : G → ℂ), (∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) → IsKFinite K φ → FiniteDimensional ℂ (Submodule.span ℂ (domSet δ K χ t φ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InducedSectionSmooth

theorem **isOpenEmbedding_val** (N : ℕ) : IsOpenEmbedding (Units.val : GL (Fin N) ℂ → Matrix (Fin N) (Fin N) ℂ)

theorem **MatSmooth.continuous** [LocallyCompactSpace G] (h : MatSmooth j ι φ) : Continuous φ

theorem **MatSmooth.der** (h : MatSmooth j ι φ) {x : ℝ → H} (hx : ContDiff ℝ ∞ fun t => ι (x t)) : MatSmooth j ι fun g => deriv (fun t => φ (g * j (x t))) 0

theorem **MatSmooth.contDiff** (h : MatSmooth j ι φ) {k : ℕ} {x : Fin k → ℝ → H} (hx : ∀ i, ContDiff ℝ ∞ fun t => ι (x i t)) (g : G) : ContDiff ℝ ∞ fun t : Fin k → ℝ => φ (g * (List.ofFn fun i => j (x i (t i))).prod)

theorem **isCompact_preimage** (hι : IsEmbedding ι) (h1 : ∀ y, ‖(ι y).det‖ = 1) {T : Set (Matrix (Fin N) (Fin N) ℂ)} (hT : IsCompact T) : IsCompact (ι ⁻¹' T)

theorem **IsBump.supp_local** (hf : IsBump f) (X₀ : Matrix (Fin N) (Fin N) ℂ) : ∃ U ∈ 𝒩 X₀, ∃ C : Set H, IsCompact C ∧ ∀ X ∈ U, ∀ y ∉ C, f (ι y⁻¹ * X) = 0

theorem **IsBump.weight** (hf : IsBump f) (X : Matrix (Fin N) (Fin N) ℂ) : (Continuous fun y => f (ι y⁻¹ * X)) ∧ HasCompactSupport fun y => f (ι y⁻¹ * X)

theorem **exists_bump** {V : Set (Matrix (Fin N) (Fin N) ℂ)} (hV : V ∈ 𝒩 1) : ∃ f, IsBump f ∧ (∀ X, f X ≠ 0 → X ∈ V) ∧ ∫ y, f (ι y⁻¹ * 1) ∂μ = 1

theorem **matSmooth_avg** (hf : IsBump f) (hψ : Continuous ψ) : MatSmooth j ι (avg j ι μ f ψ)

theorem **integrable_avg** (hf : IsBump f) (hψ : Continuous ψ) (g : G) : Integrable (fun y => f (ι y⁻¹ * 1) • ψ (g * j y)) μ

theorem **avg_mul** (hf : IsBump f) {k : G} {κ : H} (hκ : ι κ ∈ Matrix.unitaryGroup (Fin N) ℂ) (hk : ∀ y, k * j y = j (κ * y * κ⁻¹) * k) (g : G) : avg j ι μ f ψ (g * k) = avg j ι μ f (fun x => ψ (x * k)) g

theorem **tendsto_avg** {f : ℕ → Matrix (Fin N) (Fin N) ℂ → ℝ} (hf : ∀ i, IsBump (f i)) (hs : ∀ i X, f i X ≠ 0 → X ∈ Metric.ball 1 (1 / (i + 1 : ℝ))) (hm : ∀ i, ∫ y, f i (ι y⁻¹ * 1) ∂μ = 1) (hψ : Continuous ψ) (g : G) : Tendsto (fun i => avg j ι μ (f i) ψ g) atTop (𝒩 (ψ g))

theorem **matSmooth_of_mem** [LocallyCompactSpace G] {P K : Subgroup G} {δ : P →* ℝ≥0} {χ : P →* Circle} {t : ℂ} (hI : ∀ g : G, ∃ p : P, ∃ k ∈ K, g = p * k) (hK : ∀ k ∈ K, ∃ κ, ι κ ∈ Matrix.unitaryGroup (Fin N) ℂ ∧ ∀ y, k * j y = j (κ * y * κ⁻¹) * k) (hφ : φ ∈ indSpace δ K χ t) : MatSmooth j ι φ

theorem **norm_det_archMat** (h : IsCMExt F E σ) (hJ : IsHermND σ J) {w : InfinitePlace F} (τ : EmbOver E w) (x : GArch σ J w) : ‖(archMat σ J τ x).det‖ = 1

theorem **contDiff_exp_entry** (Z : Matrix (Fin n) (Fin n) ℂ) (a b : Fin n) : ContDiff ℝ ∞ fun t : ℝ => NormedSpace.exp (t • Z) a b

theorem **exists_mat** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (c : ∀ w : InfinitePlace F, GL (Fin n) (w.Completion ⊗[F] E)) : ∃ (N : ℕ) (ι : GInf σ J →* Matrix (Fin N) (Fin N) ℂ), IsEmbedding ι ∧ (∀ y, ‖(ι y).det‖ = 1) ∧ (∀ κ, (∀ w, projInfArch σ J w κ ∈ stabArch σ J w (c w)) → ι κ ∈ Matrix.unitaryGroup (Fin N) ℂ) ∧ ∀ γ, IsRealLine σ J γ → ∃ x : ℝ → GInf σ J, (∀ t, γ t = ofInf σ J (x t)) ∧ ContDiff ℝ ∞ fun t => ι (x t)

theorem **isSm_of_mem** (h : IsCMExt F E σ) (hJ : IsHermND σ J) {L : Submodule (𝒪 E) (Fin n → E)} {a : GL (Fin n) (AF F ⊗[F] E)} {P : Subgroup (GAd σ J)} {δ : P →* ℝ≥0} {χ : P →* Circle} {t : ℂ} (hI : ∀ g : GAd σ J, ∃ p : P, ∃ k ∈ levelAll σ J L a, g = p * k) {φ : GAd σ J → ℂ} (hφ : φ ∈ indSpace δ (levelAll σ J L a) χ t) : IsSm σ J φ ∧ ∀ l : List (ℝ → GAd σ J), (∀ γ ∈ l, IsRealLine σ J γ) → Continuous (l.foldr lineDer φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InducedSectionZFinite

theorem **isRealLine_conj** (h : IsCMExt F E σ) {γ : ℝ → GAd σ J} (hγ : IsRealLine σ J γ) (a : GAd σ J) : IsRealLine σ J fun t => a⁻¹ * γ t * a

theorem **isSm_mul_right** (h : IsCMExt F E σ) {f : GAd σ J → ℂ} (hf : IsSm σ J f) (a : GAd σ J) : IsSm σ J fun x => f (x * a)

theorem **foldr_mul_fin** {b : GAd σ J} (hb : b ∈ finPts σ J) (f : GAd σ J → ℂ) : ∀ l : List (ℝ → GAd σ J), (∀ γ ∈ l, IsRealLine σ J γ) → l.foldr lineDer (fun x => f (x * b)) = fun x => l.foldr lineDer f (x * b) | [], _ => rfl | γ :: l, hl => by rw [List.foldr_cons, List.foldr_cons, foldr_mul_fin hb f l fun γ' h => hl γ' (List.mem_cons_of_mem _ h)] obtain ⟨w, τ, Z, -, hγ⟩

theorem **dop_mul_right** (h : IsCMExt F E σ) {z : List (ℝ → GAd σ J) →₀ ℂ} (hz : IsCentral σ J z) {φ : GAd σ J → ℂ} (hφ : IsSm σ J φ) (k g : GAd σ J) : dop z φ (g * k) = dop z (fun x => φ (x * k)) g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertPlaceLocalCharacterConjugationInverse

theorem **secW_cA** (D : D7) {v : HeightOneSpectrum (𝒪 D.host.F)} {w : HeightOneSpectrum (𝒪 D.host.E)} (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (c : w.adicCompletion D.host.E →+* w.adicCompletion D.host.E) (hc : IsConjAt D.sp.cm.sig h c) (z : w.adicCompletion D.host.E) : secW hw h (c z) = cA D (secW hw h z)

theorem **thW_conj** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (c : w.adicCompletion D.host.E →+* w.adicCompletion D.host.E) (k : w.adicCompletion D.host.E), IsConjAt D.sp.cm.sig h c → (∀ z, c (c z) = z) → ∀ z, thW ψ hw h (k * c k) (c z) * thW ψ hw h (k * c k) z = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertPlaceLocalCharacterConjugationInverseStatement

theorem **_root_.GlobalTheory.T.inert_place_local_character_conjugation_inverse** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (c : w.adicCompletion D.host.E →+* w.adicCompletion D.host.E) (k : w.adicCompletion D.host.E), IsConjAt D.sp.cm.sig h c → (∀ z, c (c z) = z) → ∀ z, thW ψ hw h (k * c k) (c z) * thW ψ hw h (k * c k) z = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertPlaceLocalTraceCharacterConductorIntegers

theorem **ringHom_eq_id_of_int** (f : w.adicCompletion E →+* w.adicCompletion E) (h1 : ∀ x : E, f (algebraMap E _ x) = algebraMap E _ x) (h2 : ∀ z, Valued.v z ≤ 1 → Valued.v (f z) ≤ 1) : f = RingHom.id _

theorem **tensAt_mul_tensAt** (y y' : Fv D.host.F v ⊗[D.host.F] D.host.E) : tensAt D.host.E v y * tensAt D.host.E v y' = tensAt D.host.E v (y * y')

theorem **compFin_incA** (x : D.host.E) : compFin D.host.F D.host.E v (incA D x) = (1 : Fv D.host.F v) ⊗ₜ[D.host.F] x

theorem **tensAt_one_mul_incA** (x : D.host.E) : tensAt D.host.E v 1 * incA D x = tensAt D.host.E v ((1 : Fv D.host.F v) ⊗ₜ[D.host.F] x)

theorem **secW_mul** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (z z' : w.adicCompletion D.host.E) : secW hw h (z * z') = secW hw h z * secW hw h z'

theorem **complEquiv_symm_algebraMap** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (x : D.host.E) : (GlobalTheory.LFun.InertChart.complEquiv hw h).symm (algebraMap D.host.E _ x) = (1 : Fv D.host.F v) ⊗ₜ[D.host.F] x

theorem **isPrimeIdem_tensAt** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) : IsPrimeIdem w (tensAt D.host.E v 1) (tensAt D.host.E v 1)

theorem **thW_cond** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (c : w.adicCompletion D.host.E →+* w.adicCompletion D.host.E) (ϖ : w.adicCompletion D.host.E) (k : w.adicCompletion D.host.E), IsStdChar D ψ (-1) → IsConjAt D.sp.cm.sig h c → IsUnramInv ϖ c (qV D.host.F v) → GlobalTheory.InertLoc.IsGen D w k → ∀ z, (∀ u ∈ Formalism.UG.intL (w.adicCompletion D.host.E), thW ψ hw h (k * c k) (z * u) = 1) ↔ z ∈ Formalism.UG.intL (w.adicCompletion D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertPlaceLocalTraceCharacterConductorIntegersStatement

theorem **_root_.GlobalTheory.T.inertPlace_localTraceCharacter_conductor_integers** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (c : w.adicCompletion D.host.E →+* w.adicCompletion D.host.E) (ϖ : w.adicCompletion D.host.E) (k : w.adicCompletion D.host.E), IsStdChar D ψ (-1) → IsConjAt D.sp.cm.sig h c → IsUnramInv ϖ c (qV D.host.F v) → GlobalTheory.InertLoc.IsGen D w k → ∀ z, (∀ u ∈ Formalism.UG.intL (w.adicCompletion D.host.E), thW ψ hw h (k * c k) (z * u) = 1) ↔ z ∈ Formalism.UG.intL (w.adicCompletion D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertSupportedPointIntegralFramesMemLattice

theorem **secW_toCompl_tmul_mul** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (b : Fv D.host.F v) (o m : D.host.E) : secW hw h (toCompl h (b ⊗ₜ[D.host.F] o) * algebraMap D.host.E (w.adicCompletion D.host.E) m) = pairA D.host.F 0 (singleFin v b) ⊗ₜ[D.host.F] (o * m)

theorem **secW_rank_one_mem** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) {S T : HSp D} {a : D.host.E} (ha : a ∈ D.host.AA) {s : Fin S.rank → D.host.E} (hs : s ∈ frameLat S) {t : Fin T.rank → D.host.E} (ht : t ∈ frameLat T) {z : w.adicCompletion D.host.E} (hz : z ∈ Ov D.host.E w) : (Matrix.of fun i j => secW hw h (z * algebraMap D.host.E (w.adicCompletion D.host.E) (a * t i * ∑ k, star (s k) * S.gram k j)) : WO S T) ∈ latO S T (frameLat S) (frameLat T) (fun u => u = v)

theorem **secW_span_mem** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) {c : w.adicCompletion D.host.E →+* w.adicCompletion D.host.E} (hc : IsConjAt D.sp.cm.sig h c) {S T : HSp D} {a : D.host.E} (ha : a ∈ D.host.AA) {p : Fin T.rank → w.adicCompletion D.host.E} (hp : p ∈ latAt w (frameLat T)) {y : Fin S.rank → w.adicCompletion D.host.E} (hy : y ∈ latAt w (frameLat S)) : ∀ z ∈ Ov D.host.E w, (Matrix.of fun i j => secW hw h (z * algebraMap D.host.E (w.adicCompletion D.host.E) a * p i * ∑ k, c (y k) * algebraMap D.host.E (w.adicCompletion D.host.E) (S.gram k j)) : WO S T) ∈ latO S T (frameLat S) (frameLat T) (fun u => u = v)

theorem **uPt_mem_latO** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (S T : HSp D) (k : w.adicCompletion D.host.E) (aS : GL (Fin S.rank) (w.adicCompletion D.host.E)) (aT : GL (Fin T.rank) (w.adicCompletion D.host.E)), GlobalTheory.InertLoc.IsGen D w k → IsFrameAt w (frameLat S) aS → IsFrameAt w (frameLat T) aT → ∀ x : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E), (∀ i j, x i j ∈ Formalism.UG.intL (w.adicCompletion D.host.E)) → uPt hw h k aS aT x ∈ latO S T (frameLat S) (frameLat T) (fun u => u = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertSupportedPointIntegralFramesMemLatticeStatement

theorem **_root_.GlobalTheory.T.inert_supported_point_integral_frames_mem_lattice** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (S T : HSp D) (k : w.adicCompletion D.host.E) (aS : GL (Fin S.rank) (w.adicCompletion D.host.E)) (aT : GL (Fin T.rank) (w.adicCompletion D.host.E)), GlobalTheory.InertLoc.IsGen D w k → IsFrameAt w (frameLat S) aS → IsFrameAt w (frameLat T) aT → ∀ x : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E), (∀ i j, x i j ∈ Formalism.UG.intL (w.adicCompletion D.host.E)) → uPt hw h k aS aT x ∈ latO S T (frameLat S) (frameLat T) (fun u => u = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertUnitaryCosetSymmetryHeckeEven

theorem **cosets_eq** {G : Type} [Group G] (K : Subgroup G) {g g' : G} {a b : G} (ha : a ∈ K) (hb : b ∈ K) (e : g' = a * g * b) (c : G ⧸ K) : (∃ k ∈ K, c = ((k * g' : G) : G ⧸ K)) ↔ (∃ k ∈ K, c = ((k * g : G) : G ⧸ K))

theorem **cosetSym_of_inv** {G : Type} [Group G] (K : Subgroup G) (h : ∀ g : G, ∃ a ∈ K, ∃ b ∈ K, g⁻¹ = a * g * b) : CosetSym K

theorem **inv_mem_KU** {L : Type} [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] {ϖ : L} {σ : L →+* L} {q : ℕ} (hU : IsUnramInv ϖ σ q) (n : ℕ) (g : GU σ n) : ∃ a ∈ KU σ n, ∃ b ∈ KU σ n, g⁻¹ = a * g * b

theorem **inert_cosetSym_heckeEven** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : L) (σ : L →+* L) (q : ℕ), IsUnramInv ϖ σ q → ∀ n : ℕ, IsCommensurated (KU σ n) ∧ CosetSym (KU σ n) ∧ ∀ φ ∈ Hecke ℂ (KU σ n), ∀ g, φ g⁻¹ = φ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InertUnitaryCosetSymmetryHeckeEvenStatement

theorem **_root_.GlobalTheory.T.inert_unitary_coset_symmetry_hecke_even** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : L) (σ : L →+* L) (q : ℕ), IsUnramInv ϖ σ q → ∀ n : ℕ, IsCommensurated (KU σ n) ∧ CosetSym (KU σ n) ∧ ∀ φ ∈ Hecke ℂ (KU σ n), ∀ g, φ g⁻¹ = φ g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InfIdeleNegOneNotMemClosurePrincipalIdentityComp

theorem **pos_of_mem_conn** {w : InfinitePlace E} (hw : w.IsReal) {u : (InfiniteAdeleRing E)ˣ} (hu : u ∈ Subgroup.connectedComponentOfOne (InfiniteAdeleRing E)ˣ) : 0 < InfinitePlace.Completion.ringEquivRealOfIsReal hw ((u : InfiniteAdeleRing E) w)

theorem **norm_eq_one_of_sub_one_mem** {a : 𝒪 E} (ha : IsUnit a) (h3 : a - 1 ∈ Ideal.span {(3 : 𝒪 E)}) : Algebra.norm ℤ a = 1

theorem **div_norm_of_pos** {r : ℝ} (hr : 0 < r) : (r : ℂ) / ((‖(r : ℂ)‖ : ℝ) : ℂ) = 1

theorem **div_norm_of_neg** {r : ℝ} (hr : r < 0) : (r : ℂ) / ((‖(r : ℂ)‖ : ℝ) : ℂ) = -1

theorem **norm_ne_one_of_sign** {w : InfinitePlace E} (hw : w.IsReal) {k : E} (hk : k ≠ 0) (hneg : InfinitePlace.embedding_of_isReal hw k < 0) (hpos : ∀ (w' : InfinitePlace E) (hw' : w'.IsReal), w' ≠ w → 0 < InfinitePlace.embedding_of_isReal hw' k) : Algebra.norm ℚ k ≠ 1

theorem **infIdele_neg_one_not_mem** : ∀ (E : Type) [Field E] [NumberField E] (w : InfinitePlace E), w.IsReal → ReciprocityTheory.Neutral.Bridge.HCFR.infIdele E w (-1) ∉ ((diagUnits E).range ⊔ infConn E).topologicalClosure

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InfIdeleNegOneNotMemClosurePrincipalIdentityCompStatement

theorem **_root_.GlobalTheory.T.infIdele_neg_one_not_mem_closure_principal_identityComponent** : ∀ (E : Type) [Field E] [NumberField E] (w : InfinitePlace E), w.IsReal → ReciprocityTheory.Neutral.Bridge.HCFR.infIdele E w (-1) ∉ ((diagUnits E).range ⊔ infConn E).topologicalClosure

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InfPartEqZero

theorem **infPart_eq_zero** {e : AE D} (he : IsSplitIdem D e) {z : AE D} (hz : cA D e * z = z) : infPart D z = 0

theorem **map_infPart_mul** {l m n : ℕ} {x : Mx D l m} (hx : x.map (infPart D) = 0) (y : Mx D m n) : (x · y).map (infPart D) = 0

theorem **symplO_extX** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) (x : Mx D T.rank ρ) (w : WO S T) : symplO S T (d.extX T x) w = symplO S T (d.extX T x) (d.extY T (w · d.jY))

theorem **rank_qX** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.qX.rank = ρ

theorem **mixS_half** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ ℱ → ∀ e, IsSplitIdem D e → ∀ f, IsThetaFunO ψ S T f → (∃ U : AddSubgroup (Mx D T.rank ρ), IsOpen (U : Set (Mx D T.rank ρ)) ∧ ∀ u ∈ U, (∀ i j, cA D e * u i j = u i j) → ∀ x, d.mixS T ψ μ ℱ f (x + u) = d.mixS T ψ μ ℱ f x) ∧ ∃ K : Set (Mx D T.rank ρ), IsCompact K ∧ ∀ x : Mx D T.rank ρ, (∀ i j, cA D e * x i j = x i j) → d.mixS T ψ μ ℱ f x ≠ 0 → x ∈ K

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InfinitelyManySplitPlaces

theorem **infinite_splitIdem_places** (D : D7) : {v : Pl D | ∃ (e : AE D) (ε : AF D.host.F), IsSplitIdem D e ∧ IsPlaceIdem D v ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε}.Infinite

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InfinitelyManySplitPlacesStatement

theorem **_root_.GlobalTheory.T.infinitely_many_split_places** : ∀ D : D7, {v : Pl D | ∃ (e : AE D) (ε : AF D.host.F), IsSplitIdem D e ∧ IsPlaceIdem D v ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε}.Infinite

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InnerPoincareSeriesEqSetIntegralUnfolding

theorem **GlobalTheory.Core.Mi.inner_pser_eq_setIntegral** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (hK : X.LevelSubgroupCharacter) (Dt Fs : Set X.B.I.GA), IsReps X u Dt → InHH X u hK → IsFundamentalDomain (Srat X u) Fs (X.B.I.muA X.B.μf) → ∀ T : Cu X u hK, IntegrableOn (Ph X u hK T) Fs (X.B.I.muA X.B.μf) → ∀ (f : X.B.I.GA → ℂ) (φ : L2X X), RowFun X f → (φ : X.B.I.GA → ℂ) =ᵐ[muX X] f → ‹star φ, PuFun X u hK Dt T›_ℂ = ∫ x in Fs, f x * Ph X u hK T x ∂(X.B.I.muA X.B.μf)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InnerPoincareSeriesEqSetIntegralUnfoldingStatement

theorem **_root_.GlobalTheory.T.inner_poincareSeries_eq_setIntegral_unfolding** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (u : D.what₂.Word) (hK : X.LevelSubgroupCharacter) (Dt Fs : Set X.B.I.GA), IsReps X u Dt → InHH X u hK → IsFundamentalDomain (Srat X u) Fs (X.B.I.muA X.B.μf) → ∀ T : Cu X u hK, IntegrableOn (Ph X u hK T) Fs (X.B.I.muA X.B.μf) → ∀ (f : X.B.I.GA → ℂ) (φ : L2X X), RowFun X f → (φ : X.B.I.GA → ℂ) =ᵐ[muX X] f → ‹star φ, PuFun X u hK Dt T›_ℂ = ∫ x in Fs, f x * Ph X u hK T x ∂(X.B.I.muA X.B.μf)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntCoordsEv

theorem **alg_int_ev** (c : D.host.F) : ∀ᶠ w in Filter.cofinite, algebraMap D.host.F (Fv D.host.F w) c ∈ (w : Pl D).adicCompletionIntegers D.host.F

theorem **repr_tmul_loc** (w : Pl D) (x : Fv D.host.F w) (e : D.host.E) (i : Fin (Module.finrank D.host.F D.host.E)) : (Algebra.TensorProduct.basis (Fv D.host.F w) (Module.finBasis D.host.F D.host.E)).repr (x ⊗ₜ[D.host.F] e) i = x * algebraMap D.host.F (Fv D.host.F w) ((Module.finBasis D.host.F D.host.E).repr e i)

theorem **int_coords_ev** : ∀ᶠ w in Filter.cofinite, ∀ z ∈ intAE D, ∀ i, (Algebra.TensorProduct.basis (Fv D.host.F w) (Module.finBasis D.host.F D.host.E)).repr (GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.compFin D w z) i ∈ (w : Pl D).adicCompletionIntegers D.host.F

theorem **sum_repr_tmul** (w : Pl D) (y : Fv D.host.F w ⊗[D.host.F] D.host.E) : ∑ i, (Algebra.TensorProduct.basis (Fv D.host.F w) (Module.finBasis D.host.F D.host.E)).repr y i ⊗ₜ[D.host.F] (Module.finBasis D.host.F D.host.E) i = y

theorem **glue** : ∀ (D : D7) (a : Pl D → AE D), (∀ x, a x ∈ idAt D (· = x)) → (∃ T : Finset (Pl D), ∀ x ∉ T, a x ∈ intAt D (· = x)) → ∃ s : AE D, infPart D s = 0 ∧ ∀ x, s - a x ∈ idOff D (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntDenseAt

theorem **idem_at** (hW : AllTrue W) (a : GeoAll W u) (ha : a.Is) (n : Fin a.n) : Formalism.L2.NTheta.IsIdemAt (a.G n).q.e (a.pl n).1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntIffFrame

theorem **val_lt_one_iff** {E : Type} [Field E] [NumberField E] {P : HeightOneSpectrum (𝒪 E)} {ι : E →+* k} (hi : ∀ x, ι x ∈ 𝒪[k] ↔ IntAt P x) (a : E) : valuation k (ι a) < 1 ↔ P.valuation E a < 1

theorem **exists_unif** {E : Type} [Field E] [NumberField E] {P : HeightOneSpectrum (𝒪 E)} {ι : E →+* k} (hi : ∀ x, ι x ∈ 𝒪[k] ↔ IntAt P x) (hd : DenseRange ι) (h : ((qOf P : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) : ∃ ϖ : kˣ, (ϖ : k) = ι (qOf P : E) ∧ GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ

theorem **mul_sec** (q : GlobalTheory.FinPl.SplitPt D k) (x : AE D) (y : k) : x * q.sec y = q.sec (q.pr x * y)

theorem **type_prime** {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ (B.primes X).vGs) : ∃ i, qOf P = B.host.g i ∧ (B.host.g i).Prime ∧ ((B.host.g i : ℕ) : 𝒪 B.fields.E) ∈ P.asIdeal ∧ ((B.host.g i : ℕ) : 𝒪 B.fields.E) ∉ P.asIdeal ^ 2 ∧ B.host.g i ∉ B.host.small ∧ B.host.g i ≠ B.host.q ∧ B.host.g i ≠ B.host.d ∧ B.fields.conjO (X.piG i) ∈ P.asIdeal ∧ X.piG i ∉ P.asIdeal ∧ ∀ v, X.piT v - 1 ∈ P.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntIffUpMemStdHom

theorem **latO_mono** (S T : HSp D) (LS LT) {P Q : Pl D → Prop} (h : ∀ v, P v → Q v) : latO S T LS LT P ≤ latO S T LS LT Q

theorem **mem_latO_at** (q : SplitPt D k) {S T : HSp D} {LS LT} {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) (hh : IsHalf q.e u) : u ∈ latO S T LS LT (· = q.v)

theorem **int_iff_up_mem_stdHom** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ z, z ∈ LS ↔ z ∈ intVec D S.rank) → (∀ z, z ∈ LT ↔ z ∈ intVec D T.rank) → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → ∀ x : Matrix Y Y' k, (∀ r s, x r s ∈ 𝒪[k]) ↔ (GlobalTheory.LocMod.ofSplit q).up FS FT x ∈ stdHom S T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntIndFromCols

theorem **intInd_mem_SB** (a b : Type) [Fintype a] [Fintype b] : Set.indicator {x : Matrix a b k | ∀ r q, x r q ∈ 𝒪[k]} (fun _ => (1 : ℂ)) ∈ SB (Matrix a b k)

theorem **hat_const_mul** {n : Type} [Fintype n] (ψ : AddChar k Circle) (ν : Measure k) (ε : k) (c : ℂ) (f : (n → k) → ℂ) (η : n → k) : hat ψ ν ε (fun y => c * f y) η = c * hat ψ ν ε f η

theorem **hat_int** {n : Type} [Fintype n] [DecidableEq n] {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {ε : k} (hε : valuation k ε = 1) (η : n → k) : hat ψ ν ε (cosetInd (box n (1 : kˣ)) (Submodule.Quotient.mk 0)) η = if ∀ i, η i ∈ 𝒪[k] then 1 else 0

theorem **psi_mem_SB** (T : Chain k s X Z) (hT : T.Regular) (y : X (s + 1) → k) : (fun x₀ => T.Psi 1 x₀ y) ∈ SB (Matrix (X (s + 1)) (X s) k)

theorem **xiTop_mem_SB** (T : Chain k s X Z) (hT : T.Regular) {ψ : AddChar k Circle} (hψ : HasConductor ψ 1) {ν : Measure k} (hν : IsUnitVol ν) {ε : k} (hε : valuation k ε = 1) (yb : (X (s + 1) → k) ⧸ box (X (s + 1)) (1 : kˣ)) : T.xiTop (box _ (1 : kˣ)) ψ ν ε yb ∈ SB (Matrix (Z ⊕ Unit) (X (s + 1)) k)

theorem **cosetInd_zero** (y : a → k) : cosetInd (box a (1 : kˣ)) (Submodule.Quotient.mk 0) y = if ∀ i, y i ∈ 𝒪[k] then 1 else 0

theorem **intInd_fromCols** (q : Matrix a b k) (y : a → k) : Set.indicator {x : Matrix a (b ⊕ Unit) k | ∀ r c, x r c ∈ 𝒪[k]} (fun _ => (1 : ℂ)) (fromCols q (Matrix.of fun r _ => y r)) = Set.indicator {x : Matrix a b k | ∀ r c, x r c ∈ 𝒪[k]} (fun _ => (1 : ℂ)) q * cosetInd (box a (1 : kˣ)) (Submodule.Quotient.mk 0) y

theorem **intInd_rows** (q : Matrix (c ⊕ Unit) a k) : Set.indicator {x : Matrix (c ⊕ Unit) a k | ∀ r c, x r c ∈ 𝒪[k]} (fun _ => (1 : ℂ)) q = Set.indicator {x : Matrix c a k | ∀ r c, x r c ∈ 𝒪[k]} (fun _ => (1 : ℂ)) q.toRows₁ * if ∀ i, q (Sum.inr ()) i ∈ 𝒪[k] then 1 else 0

theorem **gLast** : ∀ (D : D7), LastLoc D → GlobalTheory.SymChain.AtMostOne → GLast D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntMOfMulRight

theorem **SmallM.int** {M : Matrix a b K} (h : SmallM v M) : IntM v M

theorem **IntM.add** {M N : Matrix a b K} (h : IntM v M) (h' : IntM v N) : IntM v (M + N)

theorem **IntM.sub** {M N : Matrix a b K} (h : IntM v M) (h' : IntM v N) : IntM v (M - N)

theorem **SmallM.add** {M N : Matrix a b K} (h : SmallM v M) (h' : SmallM v N) : SmallM v (M + N)

theorem **SmallM.sub** {M N : Matrix a b K} (h : SmallM v M) (h' : SmallM v N) : SmallM v (M - N)

theorem **IntM.mul** [Fintype b] {M : Matrix a b K} {N : Matrix b c K} (h : IntM v M) (h' : IntM v N) : IntM v (M * N)

theorem **SmallM.mul_int** [Fintype b] {M : Matrix a b K} {N : Matrix b c K} (h : SmallM v M) (h' : IntM v N) : SmallM v (M * N)

theorem **IntM.mul_small** [Fintype b] {M : Matrix a b K} {N : Matrix b c K} (h : IntM v M) (h' : SmallM v N) : SmallM v (M * N)

theorem **intM_one** [DecidableEq a] : IntM v (1 : Matrix a a K)

theorem **intM_of_near** [DecidableEq a] {A : Matrix a a K} (h : SmallM v (A - 1)) : IntM v A

theorem **intM_of_mul_left** [Fintype a] [DecidableEq a] {A : Matrix a a K} (h : SmallM v (A - 1)) {Y : Matrix a b K} (hY : IntM v (A * Y)) : IntM v Y

theorem **intM_of_mul_right** [Fintype a] [DecidableEq a] {A : Matrix a a K} (h : SmallM v (A - 1)) {Y : Matrix b a K} (hY : IntM v (Y * A)) : IntM v Y

theorem **two_sided** [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] {A : Matrix a a K} {C : Matrix b b K} (hA : SmallM v (A - 1)) (hC : SmallM v (C - 1)) {Y Z : Matrix a b K} (e : A * Y = Z * C) : (IntM v Y ↔ IntM v Z) ∧ (IntM v Y → SmallM v (Y - Z))

theorem **ratM_comp** (P : HeightOneSpectrum (𝒪 E)) {a b c : ℕ} (x : HomE E b c) (y : HomE E a b) : ratM P (x ∘ₗ y) = ratM P x * ratM P y

theorem **locM_mul** (P : HeightOneSpectrum (𝒪 E)) {n : ℕ} (g h : GL (Fin n) (AfE F E)) : locM F P (g * h) = locM F P g * locM F P h

theorem **locM_one** (P : HeightOneSpectrum (𝒪 E)) {n : ℕ} : locM F P (1 : GL (Fin n) (AfE F E)) = 1

theorem **ratM_apply** (P : HeightOneSpectrum (𝒪 E)) {a b : ℕ} (x : HomE E a b) (i : Fin b) (j : Fin a) : ratM P x i j = algebraMap E _ (x (Pi.single j 1) i)

theorem **intM_ratM** (P : HeightOneSpectrum (𝒪 E)) {a b : ℕ} (x : HomE E a b) : IntM Valued.v (ratM P x) ↔ ∀ j i, IntAt P (x (Pi.single j 1) i)

theorem **intAt_div_of_val** {P : HeightOneSpectrum (𝒪 E)} (hq : ((qOf P : ℕ) : 𝒪 E) ∉ P.asIdeal ^ 2) {a : E} (h : P.valuation E a < 1) : IntAt P (a / (qOf P : E))

theorem **single_std** {n : ℕ} (P : HeightOneSpectrum (𝒪 E)) (j : Fin n) : (Pi.single j 1 : Sp E n) ∈ stdAt E n P

theorem **frame_near** {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) (w : SG) : SmallM Valued.v (locM F w.1 g⁻¹ * ratM w.1 (p.deco.1.liftFr w).toLinearMap - 1)

theorem **locM_int** {g : GL (Fin n) (AfE F E)} {p : DecoF E n ST SG Q} (h : IsPtOf hc g p) (hN : p.deco.1.N = Formalism.L2.Col.stdLat E n) (w : SG) : IntM Valued.v (locM F w.1 g) ∧ IntM Valued.v (locM F w.1 g⁻¹)

theorem **read_cong** {a b : ℕ} {ST' : Finset (HeightOneSpectrum (𝒪 E))} {k y : GL (Fin a) (AfE F E)} {k' y' : GL (Fin b) (AfE F E)} {p q : DecoF E a ST SG Q} {p' q' : DecoF E b ST' SG Q} (hp : IsPtOf hc k p) (hN : p.deco.1.N = Formalism.L2.Col.stdLat E a) (hp' : IsPtOf hc k' p') (hN' : p'.deco.1.N = Formalism.L2.Col.stdLat E b) (hq : IsPtOf hc (y * k) q) (hq' : IsPtOf hc (y' * k') q') (w : SG) (x x₀ : HomE E a b) (hx : SmallM Valued.v (locM F w.1 y'⁻¹ * ratM w.1 x * locM F w.1 y - ratM w.1 x₀)) : (IntM Valued.v (ratM w.1 (inFr (p.deco.1.liftFr w) (p'.deco.1.liftFr w) x₀)) ↔ IntM Valued.v (ratM w.1 (inFr (q.deco.1.liftFr w) (q'.deco.1.liftFr w) x))) ∧ (IntM Valued.v (ratM w.1 (inFr (p.deco.1.liftFr w) (p'.deco.1.liftFr w) x₀)) → SmallM Valued.v (ratM w.1 (inFr (p.deco.1.liftFr w) (p'.deco.1.liftFr w) x₀) - ratM w.1 (inFr (q.deco.1.liftFr w) (q'.deco.1.liftFr w) x)))

theorem **locTop_shift** (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∈ Pr.vGs) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) {a b : Sp X.E (R.N (m2 + 1))} (ha : ∀ i, IntAt P.1 ((qOf P.1 : X.E) * a i)) (hb : ∀ i, IntAt P.1 (b i - a i)) : locTop Ch sel mu cst m2 P y b = locTop Ch sel mu cst m2 P y a

theorem **rep_sub_int** {u : Pr.Word} {j : ℕ} (h : j + 1 = R.m) (p q : PtAt R sel mu cst u j) (e : cosetIn q = cosetIn p) {P : rowSG Pr sel mu} (hP : P.1 ∈ Pr.vGs) (i : Fin (R.N j)) : IntAt P.1 ((frsF sel mu p.1.1 P).symm (repY p.1.2) i - (frsF sel mu q.1.1 P).symm (repY q.1.2) i)

theorem **locTop_of_cong** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∈ Pr.vGs) (hq : ((qOf P.1 : ℕ) : 𝒪 X.E) ∉ P.1.asIdeal ^ 2) {j : ℕ} (h : R.N j = R.N R.m) (y z : HomE X.E (R.N (m2 + 1)) (R.N j)) {a : Sp X.E (R.N (m2 + 1))} (ha : ∀ i, IntAt P.1 ((qOf P.1 : X.E) * a i)) (H : (IntM Valued.v (ratM P.1 y) ↔ IntM Valued.v (ratM P.1 z)) ∧ (IntM Valued.v (ratM P.1 y) → SmallM Valued.v (ratM P.1 y - ratM P.1 z))) : locTop Ch sel mu cst m2 P (castSp X.E h ∘ₗ y) a = locTop Ch sel mu cst m2 P (castSp X.E h ∘ₗ z) a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntMemInverseHalfDifferent

theorem **emb_injective** {R : Ranks} (H : Host R) : Function.Injective H.emb

theorem **embT_injective** {R : Ranks} (H : Host R) : Function.Injective (embT H)

theorem **bb_ne_bot** {R : Ranks} (H : Host R) : H.𝔟 ≠ ⊥

theorem **int_mem_inverse_half_different** {R : Ranks} (H : Host R) (t : 𝒪 H.E) : ((t : H.E)) ∈ (H.AA0 : Submodule (𝒪 H.E) H.E)

theorem **dgen_ne_zero** : ∀ {R : Ranks} (H : Host R) (cD : Tau R → ℂ), Th.IsDgen H (embT H) cD → ∀ τ, cD τ ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntOfSmul

theorem **herm_colY** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (i : Fin ρ) (a : AE D) (y : Fin S.rank → AE D) : hermAt S (colMul d.jY i a) y = a * cA D ((d.pX *ᵥ y) i)

theorem **dual_line** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} {x : Pl D} (hL : IsSelfDualC S L) (hm : MonoAt d L L₀ x) (i : Fin ρ) {b : (AE D)ˣ} (hX : ∀ a ∈ idAt D (· = x), colMul d.jX i a ∈ latV S L (· = x) ↔ ((b⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : ∀ a ∈ idAt D (· = x), colMul d.jY i a ∈ latV S L (· = x) ↔ cA D (b : AE D) * a ∈ intAt D (· = x)

theorem **last_iff** {V₀ W W₀ : HSp D} {ρ : ℕ} (d : HypDec W W₀ ρ) (z : Fin V₀.rank → D.host.E) {M : Lat V₀} {LW : Lat W} (hM : IsSelfDualC V₀ M) (hLW : IsSelfDualC W LW) {x : Pl D} {α m b : (AE D)ˣ} (r : Fin ρ) (hr : r.1 + 1 = ρ) (hb : b = xiU D * cU α * m) (hα : ∀ y, y ∈ latA W LW (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV W LW (· = x)) (hW : ∀ a ∈ idAt D (· = x), colMul d.jY r a ∈ latV W LW (· = x) ↔ cA D (b : AE D) * a ∈ intAt D (· = x)) : ∀ a ∈ idAt D (· = x), d.incAdj V₀ (a • yZero V₀ ρ z) ∈ latO V₀ W (M : Set _) (LW : Set _) (· = x) ↔ ((m : AE D) * a) • ratV z ∈ latV V₀ M (· = x)

theorem **int_of_smul** {S : HSp D} (hS : 0 < S.rank) {L : Lat S} (hL : IsSelfDualC S L) {x : Pl D} {u : AE D} (h : ∀ s ∈ latV S L (· = x), u • s ∈ latV S L (· = x)) : ∀ t ∈ intAt D (· = x), u * t ∈ intAt D (· = x)

theorem **isU_of_norm** {x : Pl D} {u : AE D} (h : ∀ t ∈ intAt D (· = x), u * t ∈ intAt D (· = x)) (hn : IsU x (u * cA D u)) : IsU x u

theorem **kap_test** {L L'' S : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) {ζ : Fin L.rank → D.host.E} {ζ₀ : Fin L''.rank → D.host.E} {lam : D.host.E} (hτ : τ *ᵥ ζ₀ = lam • ζ) {LL : Lat L} {LZ : Lat L''} {LS : Lat S} (hLL : IsSelfDualC L LL) (hLZ : IsSelfDualC L'' LZ) (hLS : IsSelfDualC S LS) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hμ : ∀ a ∈ idAt D (· = x), a • ratV ζ₀ ∈ latV L'' LZ (· = x) ↔ a ∈ intAt D (· = x)) (hk : ∀ w ∈ latO S L'' (LS : Set _) (LZ : Set _) (· = x), kap τ S w ∈ latO L S (LL : Set _) (LS : Set _) (· = x)) {a : AE D} (ha : a • ratV ζ ∈ latV L LL (· = x)) {s₀ s : Fin S.rank → AE D} (hs₀ : s₀ ∈ latV S LS (· = x)) (hs : s ∈ latV S LS (· = x)) : cA D α * (-incA D D.host.xi0⁻¹ * (α⁻¹ : (AE D)ˣ)) * incA D (star lam) * incA D (star ζ ⬝ᵥ L.gram *ᵥ ζ) * a * hermAt S s₀ s ∈ in

theorem **top_norm** {lam t t₀ : D.host.E} (hlam : lam ≠ 0) (hn : lam * star lam * t = -(D.host.xi0 * star D.host.xi0) * t₀) (α : (AE D)ˣ) {m : (AE D)ˣ} (hm' : α * (xiU D * cU α)⁻¹ * inU lam hlam = m) : cA D (m : AE D) = cA D α * (-incA D D.host.xi0⁻¹ * (α⁻¹ : (AE D)ˣ)) * incA D (star lam) ∧ (m : AE D) * cA D m * incA D t = -incA D t₀

theorem **top_line** {L L'' S : HSp D} (hS : 0 < S.rank) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) {ζ : Fin L.rank → D.host.E} {ζ₀ : Fin L''.rank → D.host.E} (hζ : IsLineVec L ζ) (hζ₀ : IsLineVec L'' ζ₀) {lam : D.host.E} (hlam : lam ≠ 0) (hn : lam * star lam * (star ζ ·ᵥ L.gram *ᵥ ζ) = -(D.host.xi0 * star D.host.xi0) * (star ζ₀ ·ᵥ L''.gram *ᵥ ζ₀)) (hτ : τ *ᵥ ζ₀ = lam • ζ) {LL : Lat L} {LZ : Lat L''} {LS : Lat S} (hLL : IsSelfDualC L LL) (hLZ : IsSelfDualC L'' LZ) (hLS : IsSelfDualC S LS) {x : Pl D} {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hμ : ∀ a ∈ idAt D (· = x), a • ratV ζ₀ ∈ latV L'' LZ (· = x) ↔ a ∈ intAt D (· = x)) (hk : ∀ w ∈ latO S L'' (LS : Set _) (LZ : Set _) (· = x), kap τ S w ∈ latO L S (LL : Set _) (LS : Set _) (· = x)) : ∀ a ∈ idAt D (· = x), a • ratV ζ ∈ latV L LL (· = x) ↔ (((α * (xiU D *

theorem **step_down** {x : Pl D} (hg : GoodAt v o x) {α m : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (j σ : ℕ) (h : j + 2 * (σ + 1) + 2 = D.R.m) (hz : ∀ a ∈ idAt D (· = x), a • ratV (vecB (c.bes (j + 1 + 1) σ (idx_step h)).o (c.bes (j + 1 + 1) σ (idx_step h)).ζ) ∈ latV _ (o.M (j + 1 + 1) σ (idx_step h)) (· = x) ↔ ((m⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : (∀ a ∈ idAt D (· = x), a • ratV (vecB (c.bes j (σ + 1) h).o (c.bes j (σ + 1) h).ζ) ∈ latV _ (o.M j (σ + 1) h) (· = x) ↔ (((α * (xiU D * cU α)⁻¹ * m)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) ∧ ∀ a ∈ idAt D (· = x), colMul (c.bes j (σ + 1) h).d.jX (Fin.last σ) a ∈ latV _ (v.N j) (· = x) ↔ (((α * α * m)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **last_ideal** (hc : c.Is) (sh : ShapeOff v o) (hp : GlobalTheory.MiTop.TopPrim v o) (hr : ∀ j, j < D.R.m → 0 < (c.X (j + 1)).rank) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : ∀ j hj, v.τ j hj *ᵥ o.ζ₀ = lam • (c.bes j 0 hj).ζ) {x : Pl D} (hx : x ∉ v.tP) {α : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (j ρ : ℕ) (h : j + 2 * ρ + 2 = D.R.m) (r : Fin ρ) (hρ : r.1 + 1 = ρ) : ∀ a ∈ idAt D (· = x), colMul (c.bes j ρ h).d.jX r a ∈ latV _ (v.N j) (· = x) ↔ (((α * α * ((α * (xiU D * cU α)⁻¹) ^ ρ * inU lam hlam))⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **prim_down** {ψ : AddChar (AF D.host.F) ℂ} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {x : Pl D} (hp : PrimAt ψ d L x) {cc b : (AE D)ˣ} (hψ : ∀ s ∈ idAt D (· = x), (∀ o ∈ intAt D (· = x), psiE ψ (s * o) = 1) ↔ ((cc⁻¹ : (AE D)ˣ) : AE D) * s ∈ intAt D (· = x)) (i j : Fin ρ) (hij : j.1 = i.1 + 1) (hL : ∀ a ∈ idAt D (· = x), colMul d.jX j a ∈ latV S L (· = x) ↔ ((b⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)) : ∀ a ∈ idAt D (· = x), colMul d.jX i a ∈ latV S L (· = x) ↔ (((cc * b)⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

theorem **all_ideal** (hc : c.Is) (sh : ShapeOff v o) (hp : GlobalTheory.MiTop.TopPrim v o) (hr : ∀ j, j < D.R.m → 0 < (c.X (j + 1)).rank) {lam : D.host.E} (hlam : lam ≠ 0) (hτ : ∀ j hj, v.τ j hj *ᵥ o.ζ₀ = lam • (c.bes j 0 hj).ζ) {x : Pl D} (hx : x ∉ v.tP) {α cc : (AE D)ˣ} (hα : ∀ (S : HSp D) (L : Lat S) y, y ∈ latA S L (· = x) ↔ ((α⁻¹ : (AE D)ˣ) : AE D) • y ∈ latV S L (· = x)) (hψ : ∀ s ∈ idAt D (· = x), (∀ o ∈ intAt D (· = x), psiE c.ψ (s * o) = 1) ↔ ((cc⁻¹ : (AE D)ˣ) : AE D) * s ∈ intAt D (· = x)) (j ρ : ℕ) (h : j + 2 * ρ + 2 = D.R.m) (n : ℕ) : ∀ r : Fin ρ, r.1 + n + 1 = ρ → ∀ a ∈ idAt D (· = x), colMul (c.bes j ρ h).d.jX r a ∈ latV _ (v.N j) (· = x) ↔ (((cc ^ n * (α * α * ((α * (xiU D * cU α)⁻¹) ^ ρ * inU lam hlam)))⁻¹ : (AE D)ˣ) : AE D) * a ∈ intAt D (· = x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntSubIsLattice

theorem **coe_intSub** (n : ℕ) : ((intSub D n : Submodule (𝒪 D.host.E) (Fin n → D.host.E)) : Set (Fin n → D.host.E)) = intVec D n

theorem **hat_intVec_iff** (D : D7) (n : ℕ) (z : Fin n → GlobalTheory.AdLat.RA D.host.F D.host.E) : z ∈ GlobalTheory.AdLat.hat D.host.F (intVec D n) ↔ ∀ i, Algebra.TensorProduct.comm D.host.F (Formalism.L2.Herm.Af D.host.F) D.host.E (z i) ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **dual_intVec** (D : D7) (n : ℕ) (v : Fin n → D.host.E) : (∀ y ∈ intVec D n, ∑ a, ∑ b, v a * (1 : Matrix (Fin n) (Fin n) D.host.E) a b * GlobalTheory.Member.cF D (y b) ∈ (1 : FractionalIdeal (nonZeroDivisors (𝒪 D.host.E)) D.host.E)) ↔ v ∈ intVec D n

theorem **intSub_fg** (n : ℕ) : (intSub D n).FG

theorem **intSub_span** (n : ℕ) : Submodule.span D.host.E (intSub D n : Set (Fin n → D.host.E)) = ⊤

theorem **sum_swap_one** (n : ℕ) (x y : Fin n → D.host.E) : ∑ i, ∑ j, star (x i) * (1 : Matrix (Fin n) (Fin n) D.host.E) i j * y j = ∑ a, ∑ b, y a * (1 : Matrix (Fin n) (Fin n) D.host.E) a b * GlobalTheory.Member.cF D (x b)

theorem **selfDual_int** (V : HSp D) (hV : V.gram = 1) : IsSelfDualC V (intSub D V.rank)

theorem **latOf_eq** (V : HSp D) (g : GA D V) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hN : Moves V rfl g N) : (N : Set (Fin V.rank → D.host.E)) = GlobalTheory.AdLat.latOf D.host.F (finHom V g).1 (intVec D V.rank)

theorem **exists_eq_latOf** (V : HSp D) (g : GA D V) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hN : Moves V rfl g N) : N.FG ∧ Submodule.span D.host.E (N : Set (Fin V.rank → D.host.E)) = ⊤ ∧ (GlobalTheory.AdLat.hat D.host.F (N : Set (Fin V.rank → D.host.E)) : Set (Fin V.rank → GlobalTheory.AdLat.RA D.host.F D.host.E)) = (finHom V g).1.mulVec '' (GlobalTheory.AdLat.hat D.host.F (intVec D V.rank))

theorem **hat_move** (V : HSp D) (g : GA D V) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hN : Moves V rfl g N) : (∀ z ∈ GlobalTheory.AdLat.hat D.host.F (intSub D V.rank : Set (Fin V.rank → D.host.E)), (finHom V g).1 *ᵥ z ∈ GlobalTheory.AdLat.hat D.host.F (N : Set (Fin V.rank → D.host.E))) ∧ (∀ z ∈ GlobalTheory.AdLat.hat D.host.F (N : Set (Fin V.rank → D.host.E)), (finHom V g⁻¹).1 *ᵥ z ∈ GlobalTheory.AdLat.hat D.host.F (intSub D V.rank : Set (Fin V.rank → D.host.E)))

theorem **selfDual_move** (V : HSp D) (hV : V.gram = 1) (g : GA D V) (N : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hN : Moves V rfl g N) : IsSelfDualC V N

theorem **isSBHalf_comp₂** {e : AE D} {Λ Λ' : AddSubgroup (WO S T)} {g : GA D S} {h : GA D T} (hΛ : ∀ u, u ∈ Λ ↔ actO S T g h u ∈ Λ') {Ψ : WO S T → ℂ} (hΨ : IsSBHalf e Λ' Ψ) : IsSBHalf e Λ fun t => Ψ (actO S T g h t)

theorem **halfInd_comp₂** {e : AE D} {Λ Λ' : AddSubgroup (WO S T)} {g : GA D S} {h : GA D T} (hΛ : ∀ u, u ∈ Λ ↔ actO S T g h u ∈ Λ') (t : WO S T) : halfInd e Λ' (actO S T g h t) = halfInd e Λ t

theorem **isPlaceModel_move₂** {ψ : AddChar (AF D.host.F) ℂ} {χS χT : (AE D)ˣ →* Circle} {om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)} (hW : IsWeilDatumO ψ S T χS χT om) {a : AE D} {Λ Λ' : AddSubgroup (WO S T)} {g : GA D S} {h : GA D T} (hΛ : ∀ u, u ∈ Λ ↔ actO S T g h u ∈ Λ') {f₀ : WO S T → ℂ} {Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)} (hTm : IsPlaceModel ψ (symplO S T) (heisO ψ S T) a Λ 1 f₀ Tm) (hth : ∀ Ψ, IsSBHalf a Λ Ψ → IsThetaFunO ψ S T (Tm Ψ)) : IsPlaceModel ψ (symplO S T) (heisO ψ S T) a Λ' 1 (om (g, h) f₀) (om (g, h) ∘ₗ Tm ∘ₗ LinearMap.funLeft ℂ ℂ (actO S T g h))

theorem **isThetaAt_move** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (a b : ℕ) (hS : S.rank = a) (hT : T.rank = b) (δ : ℂ) (sc : ℝ), StdPair θ S T δ sc ψ → ∀ (μ : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T μ μ om → ∀ e (SS : Finset (Formalism.L2.DIV.Pl D.host.E)), IsIdemSum e SS → ∀ f₀ Φ f, IsThetaFunO ψ S T f₀ → IsLatVac (heisO ψ S T) (stdHom S T) f₀ → IsSBHalf e (stdHom S T) Φ → IsThetaAt ψ (symplO S T) (heisO ψ S T) e (stdHom S T) 1 f₀ Φ f → ∀ g ∈ GfC S, ∀ h ∈ GfC T, ∀ N N', Moves S hS g N → Moves T hT h N' → IsSBHalf e (latO S T (latFr S hS N) (latFr T hT N') fun _ => True) (fun x => Φ (actO S T g⁻¹ h⁻¹ x)) ∧ IsThetaAt ψ (symplO S T) (heisO ψ S T) e (latO S T (latFr S hS N) (latFr T hT N') fun _ => True) 1 (om (g, h) f₀) (fun x => Φ (actO S T g⁻¹ h⁻¹ x)) (om (g, h) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableMajorantCoordinateCosetMeasure

theorem **finPart_incInf** (y : Ainf D.host.F ⊗[D.host.F] D.host.E) : finPart D (incInf D y) = 0

theorem **infPart_incInf** (y : Ainf D.host.F ⊗[D.host.F] D.host.E) : infPart D (incInf D y) = y

theorem **theta_realPt** {θ : InfFam D} (hθ : IsInfFam θ) (w : D.what₂.Tau → ℂ) (τ : D.what₂.Tau) : θ τ (infPart D (realPt D w)) = w τ

theorem **compAt_shift** {θ : InfFam D} (hθ : IsInfFam θ) {m n : ℕ} (P : Fin m → Fin n → Prop) (z : D.what₂.Tau × Fin m × Fin n → ℤ × ℤ) (τ : D.what₂.Tau) (i : Fin m) (j : Fin n) : compAt θ τ (shift P z) i j = if P i j then ((z (τ, i, j)).1 : ℂ) + ((z (τ, i, j)).2 : ℂ) * Complex.I else 0

theorem **finPart_shift** {m n : ℕ} (P : Fin m → Fin n → Prop) (z : D.what₂.Tau × Fin m × Fin n → ℤ × ℤ) (i : Fin m) (j : Fin n) : finPart D ((shift P z : Mx D m n) i j) = 0

theorem **shift_off** {m n : ℕ} (P : Fin m → Fin n → Prop) (z : D.what₂.Tau × Fin m × Fin n → ℤ × ℤ) (i : Fin m) (j : Fin n) (h : ¬ P i j) : (shift P z : Mx D m n) i j = 0

theorem **norm_compAt_le** (θ : InfFam D) {m n : ℕ} (x : Mx D m n) (τ : D.what₂.Tau) (i : Fin m) (j : Fin n) : ‖compAt θ τ x i j‖ ≤ realSize θ x

theorem **real_ineq** (d R : ℕ) : (((1 + (R : ℝ)) ^ (2 * d + 2))⁻¹) * ((((2 * (R + 1) + 1) ^ 2) ^ d : ℕ) : ℝ) ≤ 9 ^ d * ((((R : ℝ) + 1) ^ 2)⁻¹)

theorem **summable_nine_pow_mul_inv_sq_succ** (d : ℕ) : Summable fun R : ℕ => (9 : ℝ) ^ d * ((((R : ℝ) + 1) ^ 2)⁻¹)

theorem **maj_integrable** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ) (P : Fin m → Fin n → Prop) (a : Mx D m n) (μ : Measure (Mx D m n)), IsFiniteMeasureOnCompacts μ → (∀ᵐ x ∂μ, ∀ i j, ¬ P i j → x i j = a i j) → (∀ t : Mx D m n, (∀ i j, ¬ P i j → t i j = 0) → Measure.map (fun x => x + t) μ = μ) → ∃ N : ℕ, ∀ K : Set (Mx D m n), IsCompact K → Integrable (maj θ K N) μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableMajorantCoordinateCosetMeasureStatement

theorem **_root_.GlobalTheory.T.integrable_majorant_coordinate_coset_measure** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ) (P : Fin m → Fin n → Prop) (a : Mx D m n) (μ : Measure (Mx D m n)), IsFiniteMeasureOnCompacts μ → (∀ᵐ x ∂μ, ∀ i j, ¬ P i j → x i j = a i j) → (∀ t : Mx D m n, (∀ i j, ¬ P i j → t i j = 0) → Measure.map (fun x => x + t) μ = μ) → ∃ N : ℕ, ∀ K : Set (Mx D m n), IsCompact K → Integrable (maj θ K N) μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableMulHeightPowOfBoundedFactor

theorem **GlobalTheory.realZeta_integrable_mul_heightPow** : ∀ {G : Type} [MeasurableSpace G] [TopologicalSpace G] [OpensMeasurableSpace G] (μ : Measure G) (z m : G → ℂ) (H : G → ℝ) (C : ℝ) (b t : ℕ), Integrable z μ → Integrable m μ → Continuous H → (∀ g, 0 < H g ∧ H g ≤ 1) → t ≤ b → (∀ g, ‖z g‖ ≤ C * H g ^ ((b : ℝ) / 2) * ‖m g‖) → Integrable (fun g => z g * ((H g : ℝ) : ℂ) ^ (((0 : ℝ) : ℂ) - (t : ℂ) / 2)) μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableMulHeightPowOfBoundedFactorStatement

theorem **_root_.GlobalTheory.T.integrable_mul_heightPow_of_bounded_factor** : ∀ {G : Type} [MeasurableSpace G] [TopologicalSpace G] [OpensMeasurableSpace G] (μ : Measure G) (z m : G → ℂ) (H : G → ℝ) (C : ℝ) (b t : ℕ), Integrable z μ → Integrable m μ → Continuous H → (∀ g, 0 < H g ∧ H g ≤ 1) → t ≤ b → (∀ g, ‖z g‖ ≤ C * H g ^ ((b : ℝ) / 2) * ‖m g‖) → Integrable (fun g => z g * ((H g : ℝ) : ℂ) ^ (((0 : ℝ) : ℂ) - (t : ℂ) / 2)) μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableOfRapidDecay

theorem **continuous_toMixed** : Continuous (InfiniteAdeleRing.ringEquiv_mixedSpace F)

theorem **continuous_ofMixed** : Continuous (InfiniteAdeleRing.ringEquiv_mixedSpace F).symm

theorem **norm_toMixed_le** (y : InfiniteAdeleRing F) : ‖InfiniteAdeleRing.ringEquiv_mixedSpace F y‖ ≤ ∑ v, ‖y v‖

theorem **continuous_crdTup** : Continuous (crdTup F E b M n)

theorem **continuous_crdTup_symm** : Continuous (crdTup F E b M n).symm

theorem **continuous_mixTup** : Continuous (mixTup F M n (d

theorem **continuous_mixTup_symm** : Continuous (mixTup F M n (d

theorem **norm_tupEquiv_le** (X : Fin n → Fin M → Ainf F ⊗[F] E) : ‖tupEquiv F E b M n X‖ ≤ sizeTup b X

theorem **integrable_of_rapidDecay** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (d : ℕ) (b : Module.Basis (Fin d) F E) (M n : ℕ) (μi : Measure (Fin n → Fin M → Ainf F ⊗[F] E)), μi.IsAddHaarMeasure → ∀ Φi : (Fin n → Fin M → Ainf F ⊗[F] E) → ℂ, Continuous Φi → RapidDecay b Φi → Integrable Φi μi

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableSchwartzMajorantPullbackUnitriangular

theorem **pull_integrable** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m ρ : ℕ) (vU : VolU D ρ) (x : Matrix (Fin m) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∃ N : ℕ, ∀ K : Set (Mx D m ρ), IsCompact K → Integrable (fun u : UX D ρ => maj θ K N ((rat x : Mx D m ρ) · u.1.1)) vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableSchwartzMajorantPullbackUnitriangularStatement

theorem **_root_.GlobalTheory.T.integrable_schwartzMajorant_pullback_unitriangular** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m ρ : ℕ) (vU : VolU D ρ) (x : Matrix (Fin m) (Fin ρ) D.host.E), LinearIndependent D.host.E x.transpose → ∃ N : ℕ, ∀ K : Set (Mx D m ρ), IsCompact K → Integrable (fun u : UX D ρ => maj θ K N ((rat x : Mx D m ρ) · u.1.1)) vU.ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableStandardPairThetaWeilRange

theorem **integrable_theta** : GlobalTheory.Published.WeilThetaConvergence → ∀ (D : D7) (n : ℕ) (V : HSp D), 1 ≤ n → n + Formalism.MainNotions.SchurIrreducible.wittIdx V < V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (ω : G D n →* Module.End ℂ (schwartzA θ V.rank n)) (ν : Measure (GA D V)), ν.IsHaarMeasure → ∀ (Φ : schwartzA θ V.rank n) (g : G D n), Integrable (GlobalTheory.AutomorphicL2Unitary.descend (GQ V) (theta ω Φ g)) (autMeasure V ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableStandardPairThetaWeilRangeStatement

theorem **_root_.GlobalTheory.T.integrable_standardPair_theta_weilRange** : GlobalTheory.Published.WeilThetaConvergence → ∀ (D : D7) (n : ℕ) (V : HSp D), 1 ≤ n → n + Formalism.MainNotions.SchurIrreducible.wittIdx V < V.rank → ∀ θ : InfFam D, IsInfFam θ → ∀ (ω : G D n →*

Module.End ℂ (schwartzA θ V.rank n)) (ν : Measure (GA D V)), ν.IsHaarMeasure → ∀ (Φ : schwartzA θ V.rank n) (g : G D n), Integrable (GlobalTheory.AutomorphicL2Unitary.descend (GQ V) (theta ω Φ g)) (autMeasure V ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableUnitaryQuotientHeightPowMulRapidlyDecr

theorem **integrable_quotHeight_pow_mul_norm_of_rapidDec** : ∀ (D : D7) (A : HSp D) (m : Measure (GA D A)), m.IsHaarMeasure → ∀ (u : GA D A → ℂ), GlobalTheory.Grow.RapidDec A u → ∀ N : ℕ, Integrable (fun q : AutQ A => GlobalTheory.Grow.quotHeight A q.out ^ N * ‖u q.out‖) (autMeasure A m)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegrableUnitaryQuotientHeightPowMulRapidlyDecrStatement

theorem **_root_.GlobalTheory.T.integrable_unitaryQuotientHeight_pow_mul_rapidlyDecreasing** : ∀ (D : D7) (A : HSp D) (m : Measure (GA D A)), m.IsHaarMeasure → ∀ (u : GA D A → ℂ), GlobalTheory.Grow.RapidDec A u → ∀ N : ℕ, Integrable (fun q : AutQ A => GlobalTheory.Grow.quotHeight A q.out ^ N * ‖u q.out‖) (autMeasure A m)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralAdeleCongrZeroOfResiduesSquarefree

theorem **mem_span_of_forall_prime** {A : Type} [CommRing A] [IsDedekindDomain A] (x : A) (hx : x ≠ 0) (hsq : ∀ Q : HeightOneSpectrum A, x ∈ Q.asIdeal → x ∉ Q.asIdeal ^ 2) (a : A) (ha : ∀ Q : HeightOneSpectrum A, x ∈ Q.asIdeal → a ∈ Q.asIdeal) : a ∈ Ideal.span {x}

theorem **zmod_of_forall_resS_eq_zero** : ∀ (I : AutInst) (g : ℕ), g ≠ 0 → (∀ Q : Pl I.E, (g : 𝒪 I.E) ∈ Q.asIdeal → (g : 𝒪 I.E) ∉ Q.asIdeal ^ 2) → ∀ s : Formalism.L2.Herm.intAf I.F I.E, (∀ Q : Pl I.E, (g : 𝒪 I.E) ∈ Q.asIdeal → resS I.F Q s = 0) → ZMod' I g s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralAdeleCongrZeroOfResiduesSquarefreeStatement

theorem **_root_.GlobalTheory.T.integralAdele_congr_zero_of_residues_squarefree** : ∀ (I : AutInst) (g : ℕ), g ≠ 0 → (∀ Q : Pl I.E, (g : 𝒪 I.E) ∈ Q.asIdeal → (g : 𝒪 I.E) ∉ Q.asIdeal ^ 2) → ∀ s : Formalism.L2.Herm.intAf I.F I.E, (∀ Q : Pl I.E, (g : 𝒪 I.E) ∈ Q.asIdeal → resS I.F Q s = 0) → ZMod' I g s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralBaseTermUnfoldingCoordinates

theorem **integral_base_term_coords** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (χV χW : (AE D)ˣ →* Circle) (om : Om V W) (om' : Om V₀ W), IsWeilDatumO ψ V W χV χW om → IsWeilDatumO ψ V₀ W χV χW om' → ∀ (vS : VolM D W.rank ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (vW : Vol W) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (Th : GA D V → ℂ), IsTh ψ d W x₀ Th → ∀ (G : GA D W → ℂ) (Φ : WO V W → ℂ), RapidDec W G → IsThetaFunO ψ V W Φ → ∀ (𝒜 : Set (flagN d)) (ℬ : Set (GA D W)), NullMeasurableSet 𝒜 vN.ν → NullMeasurableSet ℬ vW.μ → (∀ f : GA D V → ENNReal, Measurable f → (∀ c, ∀ n ∈ flagN d, f (d.nC c * n) = f n) → ∫⁻ n in 𝒜, f n.1 ∂vN.ν = ∫⁻ u, ∫⁻ b in vb.ℱ, f (d.nB b * d.levi u.1) ∂vb.μ ∂vU.ν) → ∫⁻ p in 𝒜 ×ˢ ℬ, ‖Th p.1.1 * G p.2 * term ψ d W vS (om (p.1.1, p.2) Φ) x₀‖ₑ ∂(vN.ν.prod vW.μ) ≠ ⊤ → IntegrableOn (fun h => G h * fl ψ d W vS vb v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralBaseTermUnfoldingCoordinatesStatement

theorem **_root_.GlobalTheory.T.integral_baseTerm_unfolding_coordinates** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V V₀ W : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (χV χW : (AE D)ˣ →* Circle) (om : Om V W) (om' : Om V₀ W), IsWeilDatumO ψ V W χV χW om → IsWeilDatumO ψ V₀ W χV χW om' → ∀ (vS : VolM D W.rank ρ) (vN : VolN d) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (vW : Vol W) (x₀ : Matrix (Fin W.rank) (Fin ρ) D.host.E) (Th : GA D V → ℂ), IsTh ψ d W x₀ Th → ∀ (G : GA D W → ℂ) (Φ : WO V W → ℂ), RapidDec W G → IsThetaFunO ψ V W Φ → ∀ (𝒜 : Set (flagN d)) (ℬ : Set (GA D W)), NullMeasurableSet 𝒜 vN.ν → NullMeasurableSet ℬ vW.μ → (∀ f : GA D V → ENNReal, Measurable f → (∀ c, ∀ n ∈ flagN d, f (d.nC c * n) = f n) → ∫⁻ n in 𝒜, f n.1 ∂vN.ν = ∫⁻ u, ∫⁻ b in vb.ℱ, f (d.nB b * d.levi u.1) ∂vb.μ ∂vU.ν) → ∫⁻ p in 𝒜 ×ˢ ℬ, ‖Th p.1.1 * G p.2 * term ψ d W vS (om (p.1.1, p.2) Φ) x₀‖ₑ ∂(vN.ν.prod vW.μ) ≠ ⊤ → Integrabl

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralBlockIndicator

theorem **integral_block_indicator** (k : Type) [MeasurableSpace k] [Zero k] (ν : Measure k) [SigmaFinite ν] (I : Type) [Fintype I] (P Q R : I → Prop) [DecidablePred P] [DecidablePred Q] [DecidablePred R] (hR : ∀ i, R i ↔ P i ∧ ¬ Q i) (hQ : ∀ i, Q i → P i) (s : Set k) (hs : MeasurableSet s) (_hν : ν s ≠ ⊤) (G : (I → k) → ℂ) (hG : ∀ m : I → k, G m = (if ∀ i, Q i → m i ∈ s then (1 : ℂ) else 0) * G (fun i => if Q i then 0 else m i)) : ∫ e : {i // P i} → k, G (fun i => if h : P i then e ⟨i, h⟩ else 0) ∂(Measure.pi fun _ => ν) = (((ν s).toReal ^ Fintype.card {i // Q i} : ℝ) : ℂ) * ∫ e : {i // R i} → k, G (fun i => if h : R i then e ⟨i, h⟩ else 0) ∂(Measure.pi fun _ => ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralBlockIndicatorStatement

theorem **_root_.GlobalTheory.T.integral_pi_block_indicator_eq_pow_mul** : ∀ (k : Type) [MeasurableSpace k] [Zero k] (ν : Measure k) [SigmaFinite ν] (I : Type) [Fintype I] (P Q R : I → Prop) [DecidablePred P] [DecidablePred Q] [DecidablePred R], (∀ i, R i ↔ P i ∧ ¬ Q i) → (∀ i, Q i → P i) → ∀ (s : Set k), MeasurableSet s → ν s ≠ ⊤ → ∀ (G : (I → k) → ℂ), (∀ m : I → k, G m = (if ∀ i, Q i → m i ∈ s then (1 : ℂ) else 0) * G (fun i => if Q i then 0 else m i)) → ∫ e : {i // P i} → k, G (fun i => if h : P i then e ⟨i, h⟩ else 0) ∂(Measure.pi fun _ => ν) = (((ν s).toReal ^ Fintype.card {i // Q i} : ℝ) : ℂ) * ∫ e : {i // R i} → k, G (fun i => if h : R i then e ⟨i, h⟩ else 0) ∂(Measure.pi fun _ => ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralBoxInvariantEqMeasureRealMulSumReps

theorem **integral_eq_measureReal_box_mul_sum** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (n : Type) [Fintype n] (r R : kˣ) (S : Finset (n → k)), (∀ b ∈ S, b ∈ box n R) → (∀ a ∈ box n R, ∃! b, b ∈ S ∧ a - b ∈ box n r) → ∀ F : (n → k) → ℂ, (∀ t ∈ box n r, ∀ y, F (y + t) = F y) → (∀ y ∉ box n R, F y = 0) → ∫ y, F y ∂(Measure.pi fun _ => ν) = ((Measure.pi fun _ : n => ν).real (box n r) : ℂ) * ∑ b ∈ S, F b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralBoxInvariantEqMeasureRealMulSumRepsStatement

theorem **_root_.GlobalTheory.T.integral_box_invariant_eq_measureReal_mul_sum_reps** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (n : Type) [Fintype n] (r R : kˣ) (S : Finset (n → k)), (∀ b ∈ S, b ∈ box n R) → (∀ a ∈ box n R, ∃! b, b ∈ S ∧ a - b ∈ box n r) → ∀ F : (n → k) → ℂ, (∀ t ∈ box n r, ∀ y, F (y + t) = F y) → (∀ y ∉ box n R, F y = 0) → ∫ y, F y ∂(Measure.pi fun _ => ν) = ((Measure.pi fun _ : n => ν).real (box n r) : ℂ) * ∑ b ∈ S, F b

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralCompAddEquivEqSmul

theorem **integral_comp_addEquiv_eq_smul** {G : Type*} [AddGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [MeasurableSpace G] [BorelSpace G] (ν : Measure G) [ν.IsAddHaarMeasure] (T : G ≃+ G) (hc : Continuous T) (hc' : Continuous T.symm) : ∃ r : ℝ, 0 < r ∧ ∀ φ : G → ℂ, ∫ x, φ (T x) ∂ν = r * ∫ x, φ x ∂ν

theorem **isClosed_fix** (D : D7) (m n : ℕ) (c : AE D) : IsClosed {x : Mx D m n | ∀ i j, c * x i j = x i j}

theorem **integral_twist_fix_eq_smul** : ∀ (D : D7) (m n : ℕ) (c : AE D) (H : AddSubgroup (Mx D m n)), (∀ x, x ∈ H ↔ ∀ i j, c * x i j = x i j) → ∀ [MeasurableSpace H] [BorelSpace H] (ν : Measure H) [ν.IsAddHaarMeasure] (h : GL (Fin m) (AE D)) (a : GL (Fin n) (AE D)), ∃ r : ℝ, 0 < r ∧ ∀ φ : Mx D m n → ℂ, ∫ x : H, φ (h.1 · x.1 · a.1) ∂ν = r * ∫ x : H, φ x.1 ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralCoordinateShearHaarLocalField

theorem **measurePreserving_shear** {k : Type} [AddCommGroup k] [MeasurableSpace k] [MeasurableAdd₂ k] (ν : Measure k) [SigmaFinite ν] [ν.IsAddLeftInvariant] {J : Type} [Fintype J] [DecidableEq J] (S : Finset J) (g : (J → k) → (J → k)) (hg : Measurable g) (h1 : ∀ y y' : J → k, (∀ j, j ∉ S → y j = y' j) → g y = g y') (h2 : ∀ (y : J → k) j, j ∉ S → g y j = 0) : MeasurePreserving (fun y => y + g y) (Measure.pi fun _ : J => ν) (Measure.pi fun _ : J => ν)

theorem **integral_shear** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (J : Type) [Fintype J] [DecidableEq J] (S : Finset J) (g : (J → k) → (J → k)), Continuous g → (∀ y y' : J → k, (∀ j, j ∉ S → y j = y' j) → g y = g y') → (∀ (y : J → k) j, j ∉ S → g y j = 0) → ∀ F : (J → k) → ℂ, ∫ y, F (y + g y) ∂(Measure.pi fun _ => ν) = ∫ y, F y ∂(Measure.pi fun _ => ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralCoordinateShearHaarLocalFieldStatement

theorem **_root_.GlobalTheory.T.integral_coordinate_shear_haar_localField** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (J : Type) [Fintype J] [DecidableEq J] (S : Finset J) (g : (J → k) → (J → k)), Continuous g → (∀ y y' : J → k, (∀ j, j ∉ S → y j = y' j) → g y = g y') → (∀ (y : J → k) j, j ∉ S → g y j = 0) → ∀ F : (J → k) → ℂ, ∫ y, F (y + g y) ∂(Measure.pi fun _ => ν) = ∫ y, F y ∂(Measure.pi fun _ => ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralCoordinateVectorsChainSpaceSelfDual

theorem **intVec_isSelfDualC** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (j : ℕ), j ≤ B.R.m → ∀ L : Lat ((I.rec7 l Z).sp.X j), (∀ z, z ∈ L ↔ z ∈ intVec (I.rec7 l Z) _) → IsSelfDualC ((I.rec7 l Z).sp.X j) L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralCoordinateVectorsChainSpaceSelfDualStatement

theorem **_root_.GlobalTheory.T.integral_coordinate_vectors_chain_space_selfDual** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (j : ℕ), j ≤ B.R.m → ∀ L : Lat ((I.rec7 l Z).sp.X j), (∀ z, z ∈ L ↔ z ∈ intVec (I.rec7 l Z) _) → IsSelfDualC ((I.rec7 l Z).sp.X j) L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralDefiniteQuotientEqClassSum

theorem **mem_levG** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {J : Matrix (Fin n) (Fin n) D.host.E} {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) J T f bad) (g : GA D V) : g ∈ levG V hn S ↔ ∃ κ : S.U, κ.1.1 = finM V hn g

theorem **swapL_continuous** : letI

theorem **finM_inv_mul** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (a : GA D V) : finM V hn a⁻¹ * finM V hn a = 1

theorem **finM_mul_inv** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) (a : GA D V) : finM V hn a * finM V hn a⁻¹ = 1

theorem **finM_eq_one_of_mem_GinfC** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {k : GA D V} (hk : k ∈ GinfC V) : finM V hn k = 1

theorem **ncard_stab** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {J : Matrix (Fin n) (Fin n) D.host.E} {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) J T f bad) (a : GA D V) (y : UPts (sigD D) J (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (ha : finM V hn a = y.1.1) : {γ : GA D V | γ ∈ GQ V ∧ a⁻¹ * γ * a ∈ levG V hn S}.ncard = Nat.card {γ : GQ V // ∃ κ : S.U, finM V hn γ.1 * y.1.1 = y.1.1 * κ.1.1}

theorem **mem_levG_iff** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {J : Matrix (Fin n) (Fin n) D.host.E} {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) J T f bad) (g : GA D V) : g ∈ levG V hn S ↔ finGLHom V hn g ∈ S.U

theorem **finM_apply** {D : D7} (V : HSp D) (g : GA D V) (i j : Fin V.rank) : finM V rfl g i j = (swapL D.host.F D.host.E).symm ((finHom V g).1 i j)

theorem **continuous_finM** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) : letI

theorem **continuous_finGLHom** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) : letI

theorem **isOpen_levG** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) T f bad) : IsOpen ((levG V hn S : Subgroup (GA D V)) : Set (GA D V))

theorem **isCompact_levG** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) T f bad) (hV : ∀ τ, (gramAt D V τ).PosDef) : IsCompact ((levG V hn S : Subgroup (GA D V)) : Set (GA D V))

theorem **finGLHom_mem_ratPts** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {γ : GA D V} (hγ : γ ∈ GQ V) : finGLHom V hn γ ∈ ratPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)

theorem **exists_GQ_of_mem_ratPts** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {x : GL (Fin n) (D.host.E ⊗[D.host.F] FiniteAdeleRing (𝒪 D.host.F) D.host.F)} (hx : x ∈ ratPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) : ∃ γ ∈ GQ V, finGLHom V hn γ = x

theorem **existsUnique_class** {D : D7} (V : HSp D) {n : ℕ} (hn : V.rank = n) {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) T f bad) {cl : Type} (yc : cl → UPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (hy : Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S))) (h : cl → GA D V) (hh : ∀ c, finM V hn (h c) = (yc c).1.1) (x : GA D V) : ∃! c, ∃ γ ∈ GQ V, ∃ k ∈ levG V hn S, x = γ * h c * k

theorem **integral_eq_classSum** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (hn : V.rank = n) (J : Matrix (Fin n) (Fin n) D.host.E), J = V.gram.reindex (finCongr hn) (finCongr hn) → ∀ (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) J T f bad) (cl : Type) (yc : cl → UPts (sigD D) J (FiniteAdeleRing (𝒪 D.host.F) D.host.F)), Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S)) → V.SigOK → (∀ τ, (gramAt D V τ).PosDef) → ∀ [Fintype cl] (μ : Measure (GA D V)) (ℱ : Set (GA D V)), μ.IsHaarMeasure → IsFundamentalDomain (GQ V) ℱ μ → ∀ h : cl → GA D V, (∀ c, h c ∈ GfC V ∧ finM V hn (h c) = (yc c).1.1) → ∃ (vol : ℝ) (ν : Measure (GinfC V)), 0 < vol ∧ ν.IsHaarMeasure ∧ ν Set.univ = 1 ∧ ∀ Φ : GA D V → ℂ, Continuous Φ → (∀ γ ∈ GQ V, ∀ x, Φ (γ * x) = Φ x) → (∀ k ∈ GfC V, (∃ κ : S.U, κ.1.1 = finM V hn k) → ∀ x, Φ (x * k) = Φ x) → ∫

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralDefiniteQuotientEqClassSumStatement

theorem **_root_.GlobalTheory.T.integral_definiteQuotient_eq_classSum** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (V : HSp D) (n : ℕ) (hn : V.rank = n) (J : Matrix (Fin n) (Fin n) D.host.E), J = V.gram.reindex (finCongr hn) (finCongr hn) → ∀ (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 D.host.F))) (S : StepLevel D.host.F D.host.E (sigD D) J T f bad) (cl : Type) (yc : cl → UPts (sigD D) J (FiniteAdeleRing (𝒪 D.host.F) D.host.F)), Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S)) → V.SigOK → (∀ τ, (gramAt D V τ).PosDef) → ∀ [Fintype cl] (μ : Measure (GA D V)) (ℱ : Set (GA D V)), μ.IsHaarMeasure → IsFundamentalDomain (GQ V) ℱ μ → ∀ h : cl → GA D V, (∀ c, h c ∈ GfC V ∧ finM V hn (h c) = (yc c).1.1) → ∃ (vol : ℝ) (ν : Measure (GinfC V)), 0 < vol ∧ ν.IsHaarMeasure ∧ ν Set.univ = 1 ∧ ∀ Φ : GA D V → ℂ, Continuous Φ → (∀ γ ∈ GQ V, ∀ x, Φ (γ * x) = Φ x) → (∀ k ∈ GfC V, (∃ κ : S.U, κ.1.1 = f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralEntriesOfCarriesStandardLattice

theorem **carries_std_integral** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (k : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)), Formalism.L2.Herm.Carries k (Formalism.L2.Col.stdLat E n) (Formalism.L2.Col.stdLat E n) → (∀ i j, k.1 i j ∈ Formalism.L2.Herm.intAf F E) ∧ ∀ i j, (k⁻¹).1 i j ∈ Formalism.L2.Herm.intAf F E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralEntriesOfCarriesStandardLatticeStatement

theorem **_root_.GlobalTheory.T.integral_entries_of_carries_standardLattice** : ∀ (F E : Type) [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (n : ℕ) (k : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)), Formalism.L2.Herm.Carries k (Formalism.L2.Col.stdLat E n) (Formalism.L2.Col.stdLat E n) → (∀ i j, k.1 i j ∈ Formalism.L2.Herm.intAf F E) ∧ ∀ i j, (k⁻¹).1 i j ∈ Formalism.L2.Herm.intAf F E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralFiniteAdeleOfSplitPlaceComponents

theorem **GlobalTheory.IntPlaces.mem_of_idem_sum** {R : Type} [CommRing R] {n : ℕ} (S : Set R) (hS0 : (0 : R) ∈ S) (E : Fin n → R) (hEE : ∀ i, E i * E i = E i) (hE : ∀ i j, i ≠ j → E i * E j = 0) (G : Fin n → R → Prop) (hint : ∀ i z, z ∈ S ↔ (1 - E i) * z ∈ S ∧ G i z) (hG : ∀ i j z, i ≠ j → G i z → G i ((1 - E j) * z)) : ∀ z, (∑ i, E i) * z = z → (∀ i, G i z) → z ∈ S

theorem **GlobalTheory.IntPlaces.mem_intAf_of_split** : ∀ (D : D7) (n : ℕ) (k : Fin n → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (P : Fin n → Formalism.L2.DIV.Pl D.host.E), (∀ i, Formalism.L2.NTheta.IsIdemAt (q i).e (P i)) → Function.Injective P → ∀ (π' : ∀ i, Formalism.IndefR.Datum.EAf D →+* k i), (∀ i a, π' i (Formalism.IndefR.Datum.toEAf D a) = (q i).pr a) → (∀ i (z : Formalism.IndefR.Datum.EAf D), z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ (1 - Formalism.IndefR.Datum.toEAf D (q i).e) * z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ π' i z ∈ 𝒪[k i]) → ∀ z : Formalism.IndefR.Datum.EAf D, Formalism.IndefR.Datum.toEAf D (∑ i, (q i).e) * z = z → (∀ i, π' i z ∈ 𝒪[k i]) → z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralFiniteAdeleOfSplitPlaceComponentsStatement

theorem **_root_.GlobalTheory.T.integral_finiteAdele_of_split_place_components** : ∀ (D : D7) (n : ℕ) (k : Fin n → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (P : Fin n → Formalism.L2.DIV.Pl D.host.E), (∀ i, Formalism.L2.NTheta.IsIdemAt (q i).e (P i)) → Function.Injective P → ∀ (π' : ∀ i, Formalism.IndefR.Datum.EAf D →+* k i), (∀ i a, π' i (Formalism.IndefR.Datum.toEAf D a) = (q i).pr a) → (∀ i (z : Formalism.IndefR.Datum.EAf D), z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ↔ (1 - Formalism.IndefR.Datum.toEAf D (q i).e) * z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E ∧ π' i z ∈ 𝒪[k i]) → ∀ z : Formalism.IndefR.Datum.EAf D, Formalism.IndefR.Datum.toEAf D (∑ i, (q i).e) * z = z → (∀ i, π' i z ∈ 𝒪[k i]) → z ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralFoldViaLintegralFormula

theorem **integral_integral_eq_pos_sub_neg** {Y Z : Type} [MeasurableSpace Y] [MeasurableSpace Z] (ν : Measure Y) (m : Measure Z) [SFinite m] (h : Y × Z → ℝ) (hh : Measurable h) (hfin : ∫⁻ y, ∫⁻ s, ‖h (y, s)‖ₑ ∂m ∂ν < ⊤) : ∫ y, ∫ s, h (y, s) ∂m ∂ν = (∫⁻ y, ∫⁻ s, ENNReal.ofReal (h (y, s)) ∂m ∂ν).toReal − (∫⁻ y, ∫⁻ s, ENNReal.ofReal (−h (y, s)) ∂m ∂ν).toReal

theorem **clm_integral_integral** {Y Z : Type} [MeasurableSpace Y] [MeasurableSpace Z] (ν : Measure Y) (m : Measure Z) [SFinite m] (H : Y × Z → ℂ) (hH : Measurable H) (hfin : ∫⁻ y, ∫⁻ s, ‖H (y, s)‖ₑ ∂m ∂ν < ⊤) (L : ℂ →L[ℝ] ℝ) : L (∫ y, ∫ s, H (y, s) ∂m ∂ν) = ∫ y, ∫ s, L (H (y, s)) ∂m ∂ν

theorem **fold** : ∀ {α Y Z : Type} [MeasurableSpace α] [MeasurableSpace Y] [MeasurableSpace Z] (μ : Measure α) (Fs : Set α) (ν : Measure Y) (m : Measure Z) (S : Set Z), m S < ⊤ → ∀ (Φ : Y → Z → α), Measurable (Function.uncurry Φ) → ∀ (c : ℝ), 0 ≤ c → ∀ (P : (α → ENNReal) → Prop), (∀ F : α → ENNReal, Measurable F → P F → ∫⁻ x in Fs, F x ∂μ = ENNReal.ofReal c * ∫⁻ y, ∫⁻ s in S, F (Φ y s) ∂m ∂ν) → ∀ (G : α → ℂ), Measurable G → (∀ φ : ℂ → ENNReal, φ 0 = 0 → P (fun x => φ (G x))) → ∫⁻ y, ∫⁻ s in S, ‖G (Φ y s)‖ₑ ∂m ∂ν < ⊤ → IntegrableOn G Fs μ ∧ ∫ x in Fs, G x ∂μ = (c : ℂ) * ∫ y, ∫ s in S, G (Φ y s) ∂m ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralFoldViaLintegralFormulaStatement

theorem **_root_.GlobalTheory.T.integral_fold_via_lintegral_formula** : ∀ {α Y Z : Type} [MeasurableSpace α] [MeasurableSpace Y] [MeasurableSpace Z] (μ : Measure α) (Fs : Set α) (ν : Measure Y) (m : Measure Z) (S : Set Z), m S < ⊤ → ∀ (Φ : Y → Z → α), Measurable (Function.uncurry Φ) → ∀ (c : ℝ), 0 ≤ c → ∀ (P : (α → ENNReal) → Prop), (∀ F : α → ENNReal, Measurable F → P F → ∫⁻ x in Fs, F x ∂μ = ENNReal.ofReal c * ∫⁻ y, ∫⁻ s in S, F (Φ y s) ∂m ∂ν) → ∀ (G : α → ℂ), Measurable G → (∀ φ : ℂ → ENNReal, φ 0 = 0 → P (fun x => φ (G x))) → ∫⁻ y, ∫⁻ s in S, ‖G (Φ y s)‖ₑ ∂m ∂ν < ⊤ → IntegrableOn G Fs μ ∧ ∫ x in Fs, G x ∂μ = (c : ℂ) * ∫ y, ∫ s in S, G (Φ y s) ∂m ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralFrameMatrixIffMemSplitHalfHomLattice

theorem **down_eq** (hπ : ∀ a, π' (toEAf D a) = q.pr a) (x : WO S T) : (ofSplit q).down (frameAt π' βS) (frameAt π' βT) x = (frMapO S T βS βT x).map π'

theorem **up_down** {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (p : Place D k) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (t : WO S T) : p.up FS FT (p.down FS FT t) = p.e • t

theorem **down_up** {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (p : Place D k) (FS : GlobalTheory.LocMod.Frame k S.rank Y') (FT : GlobalTheory.LocMod.Frame k T.rank Y) (x : Matrix Y Y' k) : p.down FS FT (p.up FS FT x) = x

theorem **frMapO_smul** (a : AE D) (x : WO S T) (i : Fin T.rank) (j : Fin S.rank) : frMapO S T βS βT (a • x) i j = toEAf D a * frMapO S T βS βT x i j

theorem **smul_mem_at** {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {x : WO S T} (hx : x ∈ latO S T LS LT fun _ => True) : q.e • x ∈ latO S T LS LT (· = q.v)

theorem **latO_le_top** {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {P : HeightOneSpectrum (𝒪 D.host.F) → Prop} : latO S T LS LT P ≤ latO S T LS LT fun _ => True

theorem **eq_sum** {r : ℕ} {a : Fin r → AE D} {pl : Fin r → Formalism.L2.DIV.Pl D.host.E} {Q : Finset (Formalism.L2.DIV.Pl D.host.E)} {e : AE D} (ha : ∀ i, Formalism.L2.NTheta.IsIdemAt (a i) (pl i)) (hinj : Function.Injective pl) (hQ : ∀ PP, PP ∈ Q ↔ ∃ i, pl i = PP) (he : IsIdemSum e Q) : e = ∑ i, a i

theorem **isIntM_iff_mem_latO** : ∀ (D : D7) (r : ℕ) (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (pl : Fin r → Formalism.L2.DIV.Pl D.host.E) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (e : AE D), (∀ i, Formalism.L2.NTheta.IsIdemAt (q i).e (pl i)) → (∀ i, (q i).UnitAA) → Function.Injective pl → (∀ PP, PP ∈ Q ↔ ∃ i, pl i = PP) → IsIdemSum e Q → ∀ (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (βS : GL (Fin S.rank) (EAf D)) (βT : GL (Fin T.rank) (EAf D)), (∀ i (π' : EAf D →+* k i), (∀ a, π' (toEAf D a) = (q i).pr a) → (∀ y : Fin S.rank → k i, (∀ a, y a ∈ 𝒪[k i]) ↔ (fun a => (q i).sec ((βS.val.map π' *ᵥ y) a)) ∈ GlobalTheory.FinPl.latV S LS (· = (q i).v)) ∧ ∀ y : Fin T.rank → k i, (∀ a, y a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralFrameMatrixIffMemSplitHalfHomLatticeStatement

theorem **_root_.GlobalTheory.T.integralFrameMatrix_iff_mem_splitHalfHomLattice** : ∀ (D : D7) (r : ℕ) (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (pl : Fin r → Formalism.L2.DIV.Pl D.host.E) (Q : Finset (Formalism.L2.DIV.Pl D.host.E)) (e : AE D), (∀ i, Formalism.L2.NTheta.IsIdemAt (q i).e (pl i)) → (∀ i, (q i).UnitAA) → Function.Injective pl → (∀ PP, PP ∈ Q ↔ ∃ i, pl i = PP) → IsIdemSum e Q → ∀ (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (βS : GL (Fin S.rank) (EAf D)) (βT : GL (Fin T.rank) (EAf D)), (∀ i (π' : EAf D →+* k i), (∀ a, π' (toEAf D a) = (q i).pr a) → (∀ y : Fin S.rank → k i, (∀ a, y a ∈ 𝒪[k i]) ↔ (fun a => (q i).sec ((βS.val.map π' *ᵥ y) a)) ∈ GlobalTheory.FinPl.latV S LS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralGLCommensuratedDoubleCosetCardInvSymm

theorem **card_cos_eq_of_mem** {G : Type} [Group G] {K : Subgroup G} {g h a b : G} (ha : a ∈ K) (hb : b ∈ K) (e : h = a * g * b) : Nat.card (Cos K h) = Nat.card (Cos K g)

theorem **quotEquiv_mk** {G : Type} [Group G] (K : Subgroup G) (θ : G ≃* G) (hθ : ∀ x, θ x ∈ K ↔ x ∈ K) (x : G) : quotEquiv K θ hθ (x : G ⧸ K) = ((θ x : G) : G ⧸ K)

theorem **card_cos_aut** {G : Type} [Group G] (K : Subgroup G) (θ : G ≃* G) (hθ : ∀ x, θ x ∈ K ↔ x ∈ K) (g : G) : Nat.card (Cos K g) = Nat.card (Cos K (θ g))

theorem **cosetSym_of_aut** {G : Type} [Group G] (K : Subgroup G) (θ : G ≃* G) (hθ : ∀ x, θ x ∈ K ↔ x ∈ K) (h : ∀ g, ∃ a ∈ K, ∃ b ∈ K, θ g = a * g⁻¹ * b) : CosetSym K

theorem **invT_of_symm** {L : Type} [Field L] {n : ℕ} (d : GL (Fin n) L) (hd : (d : Matrix (Fin n) (Fin n) L)ᵀ = (d : Matrix (Fin n) (Fin n) L)) : invT (RingEquiv.refl L) n d = d⁻¹

theorem **invT_mem_iff** {L : Type} [Field L] {n : ℕ} (O : Subring L) (g : GL (Fin n) L) : invT (RingEquiv.refl L) n g ∈ Formalism.UG.GLint (Fin n) O ↔ g ∈ Formalism.UG.GLint (Fin n) O

theorem **invT_mem_doset** (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ) (g : GL (Fin n) (w.adicCompletion E)) : ∃ a ∈ KGL w n, ∃ b ∈ KGL w n, invT (RingEquiv.refl _) n g = a * g⁻¹ * b

theorem **GlobalTheory.ChainOnto.kGL_commensurated_cosetSym** : ∀ (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ), IsCommensurated (KGL w n) ∧ CosetSym (KGL w n)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralGLCommensuratedDoubleCosetCardInvSymmStatement

theorem **_root_.GlobalTheory.T.integral_GL_commensurated_doubleCosetCardInvSymm** : ∀ (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ), IsCommensurated (KGL w n) ∧ CosetSym (KGL w n)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralHeisenbergInvariantFormsProportional

theorem **heisInv_forms_proportional** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : L) (σ : L →+* L) (q : ℕ) (δ : L) (ψ : AddChar L ℂ) (n n' : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (ρ : Matrix (Fin n') (Fin n) L → Module.End ℂ M) (ω : GU σ n × GU σ n' →* Module.End ℂ M) (s : M), IsStd ϖ σ q δ ψ ρ ω s → ∀ l e : M →ₗ[ℂ] ℂ, (∀ u ∈ intMat L (Fin n') (Fin n), ∀ m, l (ρ u m) = l m) → (∀ u ∈ intMat L (Fin n') (Fin n), ∀ m, e (ρ u m) = e m) → ∀ m, e m * l s = l m * e s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralHeisenbergInvariantFormsProportionalStatement

theorem **_root_.GlobalTheory.T.integral_heisenberg_invariant_forms_proportional** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : L) (σ : L →+* L) (q : ℕ) (δ : L) (ψ : AddChar L ℂ) (n n' : ℕ) (M : Type) [AddCommGroup M] [Module ℂ M] (ρ : Matrix (Fin n') (Fin n) L → Module.End ℂ M) (ω : GU σ n × GU σ n' →* Module.End ℂ M) (s : M), IsStd ϖ σ q δ ψ ρ ω s → ∀ l e : M →ₗ[ℂ] ℂ, (∀ u ∈ intMat L (Fin n') (Fin n), ∀ m, l (ρ u m) = l m) → (∀ u ∈ intMat L (Fin n') (Fin n), ∀ m, e (ρ u m) = e m) → ∀ m, e m * l s = l m * e s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralIffCharacterTrivialSplit

theorem **triv_of_le_one** {ψk : AddChar k Circle} (hc : HasConductor ψk 1) {y : k} (hy : valuation k y ≤ 1) : ψk y = 1

theorem **trivial_iff_mem** {ψk : AddChar k Circle} (hc : HasConductor ψk 1) {ξ : k} (hξ : valuation k ξ = 1) (t : k) : (∀ c ∈ 𝒪[k], ψk (t * c / ξ) = 1) ↔ t ∈ 𝒪[k]

theorem **int_iff_psiE** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), valuation k ((ofSplit q).xi) = 1 → (ofSplit q).IsChar ψ ψk → HasConductor ψk 1 → ∀ s ∈ GlobalTheory.FinPl.idAt D (· = q.v), (s ∈ GlobalTheory.FinPl.intAt D (· = q.v) ↔ ∀ o ∈ GlobalTheory.FinPl.intAt D (· = q.v), psiE ψ (s * o) = 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralIndicatorGivesLocalVacuumSplitPlace

theorem **splitC** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) : IsSplitIdem D (cA D q.e)

theorem **e_eq** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) : q.sec 1 = q.e

theorem **mul_e** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (e' : Formalism.L2.Theta.AE D) (he : e' * e' = e') : q.e * e' = 0 ∨ q.e * e' = q.e

theorem **cases** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (e' : Formalism.L2.Theta.AE D) (he' : IsSplitIdem D e') {m n : ℕ} (u : Mx D m n) (hsupp : ∀ i j, (q.e + cA D q.e) * u i j = u i j) (hu : IsHalf e' u) : u = 0 ∨ IsHalf q.e u ∨ IsHalf (cA D q.e) u

theorem **GlobalTheory.LocMod.indicator_vac** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), IsStdChar D ψ (-1) → (ofSplit q).IsChar ψ ψk → HasConductor ψk 1 → ∀ (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → (∀ Tm O, (ofSplit q).IsModel ψ FS FT Tm → IsOne O → GlobalTheory.FinPl.Vac ψ S T LS LT (· = q.v) (Tm O)) ∧ ∀ f, GlobalT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralIndicatorGivesLocalVacuumSplitPlaceStatement

theorem **_root_.GlobalTheory.T.integralIndicator_gives_localVacuum_splitPlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k), q.UnitAA → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), IsStdChar D ψ (-1) → (ofSplit q).IsChar ψ ψk → HasConductor ψk 1 → ∀ (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T), IsSelfDualC S LS → IsSelfDualC T LT → (∀ y : Y' → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FS.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV S LS (· = q.v)) → (∀ y : Y → k, (∀ i, y i ∈ 𝒪[k]) ↔ (fun i => q.sec ((FT.β *ᵥ y) i)) ∈ GlobalTheory.FinPl.latV T LT (· = q.v)) → (∀ Tm O, (ofSplit q).IsModel ψ FS FT Tm → IsOne O → GlobalTheory.FinPl.Vac ψ S T LS

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralKernelLiftsFubiniSigmaFinite

lemma **lintegral_mul_le_sqrt** (ν : Measure Y) {a₁ a₂ : Y → ℝ≥0∞} (h₁ : Measurable a₁) (h₂ : Measurable a₂) : ∫⁻ y, a₁ y * a₂ y ∂ν ≤ (∫⁻ y, a₁ y ^ (2:ℝ) ∂ν) ^ (1 / (2:ℝ)) * (∫⁻ y, a₂ y ^ (2:ℝ) ∂ν) ^ (1 / (2:ℝ))

lemma **sqrt_le_of_sq_le** {I N : ℝ≥0∞} (h : I ≤ N ^ 2) : I ^ (1 / (2:ℝ)) ≤ N

lemma **lintegral_prod_split** {a₁ : X₁ → Y → ℝ≥0∞} {a₂ : X₂ → Y → ℝ≥0∞} (ha₁ : Measurable (Function.uncurry a₁)) (ha₂ : Measurable (Function.uncurry a₂)) (y : Y) : (∫⁻ x₁, a₁ x₁ y ∂μ₁) * (∫⁻ x₂, a₂ x₂ y ∂μ₂) = ∫⁻ p : X₁ × X₂, a₁ p.1 y * a₂ p.2 y ∂μ₁.prod μ₂

lemma **core** {a₁ : X₁ → Y → ℝ≥0∞} {a₂ : X₂ → Y → ℝ≥0∞} {b₁ : X₁ → ℝ≥0∞} {b₂ : X₂ → ℝ≥0∞} (ha₁ : Measurable (Function.uncurry a₁)) (ha₂ : Measurable (Function.uncurry a₂)) (hb₁ : Measurable b₁) (hb₂ : Measurable b₂) {N₁ : X₁ → ℝ≥0∞} {N₂ : X₂ → ℝ≥0∞} (hN₁ : ∀ x, ∫⁻ y, a₁ x y ^ 2 ∂ν ≤ N₁ x ^ 2) (hN₂ : ∀ x, ∫⁻ y, a₂ x y ^ 2 ∂ν ≤ N₂ x ^ 2) : ∫⁻ y, (∫⁻ x₁, a₁ x₁ y * b₁ x₁ ∂μ₁) * (∫⁻ x₂, a₂ x₂ y * b₂ x₂ ∂μ₂) ∂ν ≤ (∫⁻ x, N₁ x * b₁ x ∂μ₁) * (∫⁻ x, N₂ x * b₂ x ∂μ₂)

lemma **memLp_lift** {k : X₁ → Y → ℂ} {u : X₁ → ℂ} (hk : Measurable (Function.uncurry k)) (hu : Measurable u) {N : X₁ → ℝ≥0∞} (hN : ∀ x, ∫⁻ y, ‖k x y‖ₑ ^ 2 ∂ν ≤ N x ^ 2) (hfin : ∫⁻ x, N x * ‖u x‖ₑ ∂μ₁ ≠ ⊤) : MemLp (fun y => ∫ x, k x y * u x ∂μ₁) 2 ν

theorem **abstract_fubini_two_lifts** : ∀ {X₁ X₂ Y : Type*} [MeasurableSpace X₁] [MeasurableSpace X₂] [MeasurableSpace Y] (μ₁ : Measure X₁) (μ₂ : Measure X₂) (ν : Measure Y) [SigmaFinite μ₁] [SigmaFinite μ₂] [SigmaFinite ν] (k₁ : X₁ → Y → ℂ) (k₂ : X₂ → Y → ℂ) (u₁ : X₁ → ℂ) (u₂ : X₂ → ℂ), Measurable (Function.uncurry k₁) → Measurable (Function.uncurry k₂) → Measurable u₁ → Measurable u₂ → ∀ (N₁ : X₁ → ENNReal) (N₂ : X₂ → ENNReal), (∀ x, ∫⁻ y, ‖k₁ x y‖ₑ ^ 2 ∂ν ≤ N₁ x ^ 2) → (∀ x, ∫⁻ y, ‖k₂ x y‖ₑ ^ 2 ∂ν ≤ N₂ x ^ 2) → ∫⁻ x, N₁ x * ‖u₁ x‖ₑ ∂μ₁ ≠ ⊤ → ∫⁻ x, N₂ x * ‖u₂ x‖ₑ ∂μ₂ ≠ ⊤ → (∀ y, Integrable (fun x => k₁ x y * u₁ x) μ₁) → (∀ y, Integrable (fun x => k₂ x y * u₂ x) μ₂) → MemLp (fun y => ∫ x, k₁ x y * u₁ x ∂μ₁) 2 ν ∧ MemLp (fun y => ∫ x, k₂ x y * u₂ x ∂μ₂) 2 ν ∧ ∫ y, (∫ x, k₁ x y * u₁ x ∂μ₁) * conj (∫ x, k₂ x y * u₂ x ∂μ₂) ∂ν = ∫ p : X₁ × X₂, (∫ y, k₁ p.1 y * conj (k₂ p.2 y) ∂ν) * u₁ p.1 * conj

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralKernelLiftsFubiniSigmaFiniteStatement

theorem **_root_.GlobalTheory.T.integral_kernel_lifts_fubini_sigmaFinite** : ∀ {X₁ X₂ Y : Type*} [MeasurableSpace X₁] [MeasurableSpace X₂] [MeasurableSpace Y] (μ₁ : Measure X₁) (μ₂ : Measure X₂) (ν : Measure Y) [SigmaFinite μ₁] [SigmaFinite μ₂] [SigmaFinite ν] (k₁ : X₁ → Y → ℂ) (k₂ : X₂ → Y → ℂ) (u₁ : X₁ → ℂ) (u₂ : X₂ → ℂ), Measurable (Function.uncurry k₁) → Measurable (Function.uncurry k₂) → Measurable u₁ → Measurable u₂ → ∀ (N₁ : X₁ → ENNReal) (N₂ : X₂ → ENNReal), (∀ x, ∫⁻ y, ‖k₁ x y‖ₑ ^ 2 ∂ν ≤ N₁ x ^ 2) → (∀ x, ∫⁻ y, ‖k₂ x y‖ₑ ^ 2 ∂ν ≤ N₂ x ^ 2) → ∫⁻ x, N₁ x * ‖u₁ x‖ₑ ∂μ₁ ≠ ⊤ → ∫⁻ x, N₂ x * ‖u₂ x‖ₑ ∂μ₂ ≠ ⊤ → (∀ y, Integrable (fun x => k₁ x y * u₁ x) μ₁) → (∀ y, Integrable (fun x => k₂ x y * u₂ x) μ₂) → MemLp (fun y => ∫ x, k₁ x y * u₁ x ∂μ₁) 2 ν ∧ MemLp (fun y => ∫ x, k₂ x y * u₂ x ∂μ₂) 2 ν ∧ ∫ y, (∫ x, k₁ x y * u₁ x ∂μ₁) * conj (∫ x, k₂ x y * u₂ x ∂μ₂) ∂ν = ∫ p : X₁ × X₂, (∫ y, k₁ p.1 y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralLastFactorSliceMemSchwartzBruhat

theorem **integral_slice_mem_SB** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (I₀ I₁ I₂ : Type) [Fintype I₀] [Fintype I₁] [Fintype I₂] (Φ : SB ((I₀ → k) × (I₁ → k) × (I₂ → k))) : (fun x => ∫ a, Φ.1 (x, 0, a) ∂(Measure.pi fun _ => ν)) ∈ SB (I₀ → k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralLastFactorSliceMemSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.integral_last_factor_slice_mem_schwartzBruhat** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k) (I₀ I₁ I₂ : Type) [Fintype I₀] [Fintype I₁] [Fintype I₂] (Φ : SB ((I₀ → k) × (I₁ → k) × (I₂ → k))), (fun x => ∫ a, Φ.1 (x, 0, a) ∂(Measure.pi fun _ => ν)) ∈ SB (I₀ → k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralMatrixSplitPlaceMemStdHomLattice

theorem **sec_mul'** (q : GlobalTheory.FinPl.SplitPt D k) (a : k) (b : AE D) : q.sec a * b = q.sec (a * q.pr b)

theorem **int_up_mem_stdHom** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S T : HSp D) (a₀ : D.host.E), a₀ ∈ (D.host.AA : Set D.host.E) → q.pr (incA D a₀) ≠ 0 → ∀ (FS : Frame k S.rank (Fin S.rank)) (FT : Frame k T.rank (Fin T.rank)), FT.β = 1 → FT.γ = 1 → FS.γ = q.pr (incA D a₀) • (HSpace.gram S).map (fun a => q.pr (incA D a)) → ∀ t : Matrix (Fin T.rank) (Fin S.rank) k, (∀ r s, t r s ∈ 𝒪[k]) → (Place.ofSplitPt q).up FS FT t ∈ stdHom S T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralMatrixSplitPlaceMemStdHomLatticeStatement

theorem **_root_.GlobalTheory.T.integral_matrix_split_place_mem_stdHomLattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S T : HSp D) (a₀ : D.host.E), a₀ ∈ (D.host.AA : Set D.host.E) → q.pr (incA D a₀) ≠ 0 → ∀ (FS : Frame k S.rank (Fin S.rank)) (FT : Frame k T.rank (Fin T.rank)), FT.β = 1 → FT.γ = 1 →

FS.γ = q.pr (incA D a₀) • (HSpace.gram S).map (fun a => q.pr (incA D a)) → ∀ t : Matrix (Fin T.rank) (Fin S.rank) k, (∀ r s, t r s ∈ 𝒪[k]) → (Place.ofSplitPt q).up FS FT t ∈ stdHom S T

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralNormSqMatrixCoefficientNeZeroDefinite

theorem **integral_normSq_coef_ne_zero** : ∀ (D : D7) (V : HSp D), V.SigOK → V.IsPosDef → ∀ (c : CoefA V), c.Is → ∀ w : c.W, w ≠ 0 → ∀ mR : Measure (GinfC V), mR.IsHaarMeasure → ∫ b : GinfC V, ‖⟪w, c.ρ (b : GA D V) w⟫_ℂ‖ ^ 2 ∂mR ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralNormSqMatrixCoefficientNeZeroDefiniteStatement

theorem **_root_.GlobalTheory.T.integral_normSq_matrix_coefficient_ne_zero_definite** : ∀ (D : D7) (V : HSp D), V.SigOK → V.IsPosDef → ∀ (c : CoefA V), c.Is → ∀ w : c.W, w ≠ 0 → ∀ mR : Measure (GinfC V), mR.IsHaarMeasure → ∫ b : GinfC V, ‖⟪w, c.ρ (b : GA D V) w⟫_ℂ‖ ^ 2 ∂mR ≠ 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralOfAddCharTrivialOnHomLatticeLine

theorem **single_mul_mul_single** {ρ : ℕ} (M : Mx D ρ ρ) (i : Fin ρ) : (Matrix.single i i 1 : Mx D ρ ρ) · (M · (Matrix.single i i 1 : Mx D ρ ρ)) = Matrix.single i i (M i i)

theorem **pairS_single** {W W₀ : HSp D} {ρ : ℕ} (dW : HypDec W W₀ ρ) (i : Fin ρ) (c : AE D) (x : Mx D W.rank ρ) : pairS W (dW.jX · (Matrix.single i i c : Mx D ρ ρ)) x = cA D ((dW.pY · x) i i) ∗ c

theorem **int_of_psi_line** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ W W₀ : HSp D) (ρ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (LV : Lat V) (LV₀ : Lat V₀) (LW : Lat W) (LW₀ : Lat W₀) (v : Pl D), Tame D v → IsSelfDualC V LV → IsSelfDualC W LW → MonoAt dV LV LV₀ v → MonoAt dW LW LW₀ v → ColsAt dV dW.jX LV LW v → ∀ (i : Fin ρ) (c : AE D), c - 1 ∈ idAt D (· = v) → (∀ b ∈ idAt D (· = v), dV.extY W (dW.jY · Matrix.single i i b) ∈ latO V W (LV : Set _) (LW : Set _) (· = v) → ψ (symplO V W (dV.extX W (dW.jX · Matrix.single i i c)) (dV.extY W (dW.jY · Matrix.single i i b))) = 1) → c - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralOfAddCharTrivialOnHomLatticeLineStatement

theorem **_root_.GlobalTheory.T.integral_of_addChar_trivial_on_homLattice_line** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V V₀ W W₀ : HSp D) (ρ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (LV : Lat V) (LV₀ : Lat V₀) (LW : Lat W) (LW₀ : Lat W₀) (v : Pl D), Tame D v → IsSelfDualC V LV → IsSelfDualC W LW → MonoAt dV LV LV₀ v → MonoAt dW LW LW₀ v → ColsAt dV dW.jX LV LW v → ∀ (i : Fin ρ) (c : AE D), c - 1 ∈ idAt D (· = v) → (∀ b ∈ idAt D (· = v), dV.extY W (dW.jY · Matrix.single i i b) ∈ latO V W (LV : Set _) (LW : Set _) (· = v) → ψ (symplO V W (dV.extX W (dW.jX · Matrix.single i i c)) (dV.extY W (dW.jY · Matrix.single i i b))) = 1) → c - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralOfKeepsSelfDualLatticeLineColon

theorem **mem_int_of_pow** {E : Type} [Field E] [NumberField E] (Q : IsDedekindDomain.HeightOneSpectrum (𝒪 E)) (c y : Q.adicCompletion E) (hc : c ≠ 0) (h : ∀ n : ℕ, c ∗ y ^ n ∈ Q.adicCompletionIntegers E) : y ∈ Q.adicCompletionIntegers E

theorem **herm_cols** {S : HSp D} {ρ : ℕ} (A B : Mx D S.rank ρ) (κ a : AE D) (l l' : Fin ρ) : GlobalTheory.IwaPlace.hermAt S (fun j => a ∗ A j l) (fun i => κ ∗ B i l') = cA D κ ∗ a ∗ (adj B · (gA S · A)) l' l

theorem **mul_incA_mem** {v : Pl D} {a : AE D} (ha : a ∈ intAt D (· = v)) (o : 𝒪 D.host.E) : a ∗ incA D (o : D.host.E) ∈ intAt D (· = v)

theorem **exists_den** {ι κ : Type} [Fintype ι] [Fintype κ] (M : Matrix ι κ D.host.E) : ∃ t : 𝒪 D.host.E, t ≠ 0 ∧ ∀ k l, ∃ o : 𝒪 D.host.E, (t : D.host.E) ∗ M k l = o

theorem **colon_int** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSelfDualC S L → MonoAt d L L₀ v → ∀ (i j : Fin ρ) (x : AE D), x - 1 ∈ idAt D (· = v) → (∀ r ∈ idAt D (· = v), (∀ a : AE D, colMul d.jX j a ∈ latV S L (· = v) → colMul d.jX i (r ∗ a) ∈ latV S L (· = v)) → ∀ a : AE D, colMul d.jX j a ∈ latV S L (· = v) → colMul d.jX i (r ∗ x ∗ a) ∈ latV S L (· = v)) → x - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralOfKeepsSelfDualLatticeLineColonStatement

theorem **_root_.GlobalTheory.T.integral_of_keeps_selfDualLatticeLineColon** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSelfDualC S L → MonoAt d L L₀ v → ∀ (i j : Fin ρ) (x : AE D), x - 1 ∈ idAt D (· = v) → (∀ r ∈ idAt D (· = v), (∀ a : AE D, colMul d.jX j a ∈ latV S L (· = v) → colMul d.jX i (r ∗ a) ∈ latV S L (· = v)) → ∀ a : AE D, colMul d.jX j a ∈ latV S L (· = v) → colMul d.jX i (r ∗ x ∗ a) ∈ latV S L (· = v)) → x - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralPeriodicCompactSupportEqFiniteSum

theorem **exists_irreducible_of_isUnif** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ϖ : kˣ} (h : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) : ∃ p : 𝒪[k], Irreducible p ∧ (p : k) = ϖ

theorem **integral_eq_sum** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (ϖ : kˣ), GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → ∀ (A B : ℕ) (T : Finset k), (∀ t ∈ T, valuation k t ≤ valuation k (((ϖ ^ A)⁻¹ : kˣ) : k)) → (∀ x : k, valuation k x ≤ valuation k (((ϖ ^ A)⁻¹ : kˣ) : k) → ∃! t, t ∈ T ∧ valuation k (x - t) ≤ valuation k ((ϖ : k) ^ B)) → ∀ (I : Type) [Fintype I] [DecidableEq I] (F : (I → k) → ℂ), (∀ x, F x ≠ 0 → ∀ i, valuation k (x i) ≤ valuation k (((ϖ ^ A)⁻¹ : kˣ) : k)) → (∀ x y, (∀ i, valuation k (y i) ≤ valuation k ((ϖ : k) ^ B)) → F (x + y) = F x) → ∫ x, F x ∂(Measure.pi fun _ : I => ν) = (((resCard k : ℂ) ^ B)⁻¹) ^ Fintype.card I ∗ ∑ t ∈ Fintype.piFinset (fun _ : I => T), F t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralPeriodicCompactSupportEqFiniteSumStatement

theorem **_root_.GlobalTheory.T.integral_periodic_compactSupport_eq_finiteSum** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (ϖ : kˣ), GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ → ∀ (A B : ℕ) (T : Finset k), (∀ t ∈ T, valuation k t ≤ valuation k (((ϖ ^ A)⁻¹ : kˣ) : k)) → (∀ x : k, valuation k x ≤ valuation k (((ϖ ^ A)⁻¹ : kˣ) : k) → ∃! t, t ∈ T ∧ valuation k (x - t) ≤ valuation k ((ϖ : k) ^ B)) → ∀ (I : Type) [Fintype I] [DecidableEq I] (F : (I → k) → ℂ), (∀ x, F x ≠ 0 → ∀ i, valuation k (x i) ≤ valuation k (((ϖ ^ A)⁻¹ : kˣ) : k)) → (∀ x y, (∀ i, valuation k (y i) ≤ valuation k ((ϖ : k) ^ B)) → F (x + y) = F x) → ∫ x, F x ∂(Measure.pi fun _ : I => ν) = (((resCard k : ℂ) ^ B)⁻¹) ^ Fintype.card I ∗ ∑ t ∈ Fintype.piFinset (fun _ : I => T), F t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralQuotientEqIntegralFactorQuotient

theorem **mulProdEquiv_apply** (p : G₁ × G₂) : (mulProdEquiv hc e he p : H) = p.1 ∗ p.2

theorem **mulProdEquiv_symm_mul** (g : G) : (((mulProdEquiv hc e he).symm g).1 : H) ∗ ((mulProdEquiv hc e he).symm g).2 = g

theorem **eq_of_mul_eq** {p q : G₁ × G₂} (h : (p.1 : H) * p.2 = q.1 * q.2) : p = q

theorem **symm_mem** (g : R.subgroupOf G) : (((mulProdEquiv hc e he).symm g.1).1 : H) ∈ R ∧ (((mulProdEquiv hc e he).symm g.1).2 : H) ∈ R

theorem **mulProdEquivR_apply** (p : R.subgroupOf G₁ × R.subgroupOf G₂) : (mulProdEquivR hc e he hRm p : G) = mulProdEquiv hc e he (p.1.1, p.2.1)

theorem **f_mul_of_mem** (hc : ∀ a ∈ G₁, ∀ b ∈ G₂, a * b = b * a) (h1 : G₁ ≤ G) (h2 : G₂ ≤ G) (hf1 : ∀ r ∈ R, r ∈ G → ∀ g ∈ G, f (r * g) = f g) (hf2 : ∀ k ∈ K₂, ∀ g ∈ G, f (g * k) = f g) {x y b : H} (hx : x ∈ G₁) (hbR : b ∈ R) (hb : b ∈ G₂) (hby : b * y ∈ K₂) : f (x * y) = f x

theorem **f_mul_eq_zero** (hc : ∀ a ∈ G₁, ∀ b ∈ G₂, a * b = b * a) (e : (G₁ × G₂) ≃ₜ G) (he : ∀ p, (e p : H) = p.1 * p.2) (hRm : ∀ a ∈ G₁, ∀ b ∈ G₂, a * b ∈ R → a ∈ R ∧ b ∈ R) (h1 : G₁ ≤ G) (h2 : G₂ ≤ G) (hK₂ : K₂ ≤ G₂) (hf3 : ∀ g ∈ G, f g ≠ 0 → ∃ r ∈ R, r ∈ G ∧ ∃ x ∈ G₁, ∃ k ∈ K₂, g = r * x * k) {x y : H} (hx : x ∈ G₁) (hy : y ∈ G₂) (hno : ∀ b ∈ R, b ∈ G₂ → b * y ∉ K₂) : f (x * y) = 0

theorem **exists_eq_map_smul_prod** {A B P : Type} [Group A] [TopologicalSpace A] [IsTopologicalGroup A] [LocallyCompactSpace A] [SecondCountableTopology A] [MeasurableSpace A] [BorelSpace A] [Group B] [TopologicalSpace B] [IsTopologicalGroup B] [LocallyCompactSpace B] [SecondCountableTopology B] [MeasurableSpace B] [BorelSpace B] [Group P] [TopologicalSpace P] [IsTopologicalGroup P] [MeasurableSpace P] [BorelSpace P] (e : (A × B) ≃ₜ* P) (ν : Measure P) (νA : Measure A) (νB : Measure B) [ν.IsHaarMeasure] [νA.IsHaarMeasure] [νB.IsHaarMeasure] : ∃ c : ℝ≥0, 0 < c ∧ ν = (c • νA.prod νB).map e

theorem **isCompact_setOf_val** {X : Type} [TopologicalSpace X] {S K : Set X} (hK : IsCompact K) (hS : IsClosed S) : IsCompact {x : S | (x : X) ∈ K}

theorem **isInvInvariant_of_prod** {A B P : Type} [Group A] [TopologicalSpace A] [IsTopologicalGroup A] [SecondCountableTopology A] [MeasurableSpace A] [BorelSpace A] [Group B] [TopologicalSpace B] [IsTopologicalGroup B] [SecondCountableTopology B] [MeasurableSpace B] [BorelSpace B] [Group P] [TopologicalSpace P] [IsTopologicalGroup P] [MeasurableSpace P] [BorelSpace P] (e : (A × B) ≃ₜ* P) (ν : Measure P) [ν.IsInvInvariant] (νA : Measure A) (νB : Measure B) [SFinite νA] [SFinite νB] (c : ℝ≥0) (hc : 0 < c) (hν : ν = (c • νA.prod νB).map e) {U : Set A} (hUi : U⁻¹ = U) (hU1 : νA U = 1) : νB.IsInvInvariant

theorem **measure_setOf_mul_mem** {B : Type} [Group B] [MeasurableSpace B] [MeasurableMul B] [MeasurableInv B] (ν : Measure B) [ν.IsMulLeftInvariant] [ν.IsInvInvariant] {V : Set B} (hV : V⁻¹ = V) (b₀ : B) : ν {b | b * b₀ ∈ V} = ν V

theorem **collapse_factor** : ∀ (H : Type) [Group H] [TopologicalSpace H] [IsTopologicalGroup H] [SecondCountableTopology H] [LocallyCompactSpace H] [T2Space H] [MeasurableSpace H] [BorelSpace H] (G G₁ G₂ R K₁ K₂ : Subgroup H), IsClosed (G : Set H) → IsClosed (G₁ : Set H) → IsClosed (G₂ : Set H) → IsClosed (R : Set H) → G₁ ≤ G → G₂ ≤ G → (∀ a ∈ G₁, ∀ b ∈ G₂, a * b = b * a) → (∃ e : (G₁ × G₂) ≃ₜ G, ∀ p, (e p : H) = p.1 * p.2) → (∀ a ∈ G₁, ∀ b ∈ G₂, a * b ∈ R → a ∈ R ∧ b ∈ R) → K₁ ≤ G₁ → K₂ ≤ G₂ → IsCompact (K₁ : Set H) → IsCompact (K₂ : Set H) → IsOpen ((K₁.subgroupOf G₁ : Subgroup G₁) : Set G₁) → IsOpen ((K₂.subgroupOf G₂ : Subgroup G₂) : Set G₂) → ∀ (ρ : Measure (R.subgroupOf G)) (μ : Measure G) (ρ₁ : Measure (R.subgroupOf G₁)) (μ₁ : Measure G₁), ρ.IsHaarMeasure → μ.IsHaarMeasure → ρ₁.IsHaarMeasure → μ₁.IsHaarMeasure → ρ.IsInvInvariant → μ₁ {g | g.1 ∈ K₁} = 1 → ρ₁ {g | g.1.1 ∈ K₁} = 1 → ∀ (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralQuotientEqIntegralFactorQuotientStatement

theorem **_root_.GlobalTheory.T.integral_quotient_eq_integral_factor_quotient** : ∀ (H : Type) [Group H] [TopologicalSpace H] [IsTopologicalGroup H] [SecondCountableTopology H] [LocallyCompactSpace H] [T2Space H] [MeasurableSpace H] [BorelSpace H] (G G₁ G₂ R K₁ K₂ : Subgroup H), IsClosed (G : Set H) → IsClosed (G₁ : Set H) → IsClosed (G₂ : Set H) → IsClosed (R : Set H) → G₁ ≤ G → G₂ ≤ G → (∀ a ∈ G₁, ∀ b ∈ G₂, a * b = b * a) → (∃ e : (G₁ × G₂) ≃ₜ G, ∀ p, (e p : H) = p.1 * p.2) → (∀ a ∈ G₁, ∀ b ∈ G₂, a * b ∈ R → a ∈ R ∧ b ∈ R) → K₁ ≤ G₁ → K₂ ≤ G₂ → IsCompact (K₁ : Set H) → IsCompact (K₂ : Set H) → IsOpen ((K₁.subgroupOf G₁ : Subgroup G₁) : Set G₁) → IsOpen ((K₂.subgroupOf G₂ : Subgroup G₂) : Set G₂) → ∀ (ρ : Measure (R.subgroupOf G)) (μ : Measure G) (ρ₁ : Measure (R.subgroupOf G₁)) (μ₁ : Measure G₁), ρ.IsHaarMeasure → μ.IsHaarMeasure → ρ₁.IsHaarMeasure → μ₁.IsHaarMeasure → ρ.IsInvInvariant → μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralQuotientEqMeanIntermediateSubgroup

theorem **existsUnique_mul_mem** {G : Type} [Group G] {Γ R : Subgroup G} (hΓ : Γ ≤ R) {S : Set G} (hS : ∀ g : G, ∃! γ : Γ, (γ : G) * g ∈ S) (h : G) (r : R) : ∃! γ : Γ.subgroupOf R, γ • r ∈ {r : R | (r : G) * h ∈ S}

theorem **quot_fubini_mean** : ∀ (G : Type) [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (μ : Measure G) [μ.IsHaarMeasure] (R Γ : Subgroup G), Γ ≤ R → IsClosed (R : Set G) → Countable Γ → DiscreteTopology Γ → ∀ (μR : Measure R), μR.IsHaarMeasure → ∀ (𝓡 : Set R) (𝓑 : Set G), IsFundamentalDomain (Γ.subgroupOf R) 𝓡 μR → IsFundamentalDomain Γ 𝓑 μ → ∀ f : G → ℂ, Measurable f → (∀ γ ∈ Γ, ∀ g, f (γ * g) = f g) → IntegrableOn f 𝓑 μ → (∀ h, IntegrableOn (fun r : R => f (r * h)) 𝓡 μR) → IsQuotIntL1 μR μ (fun h => ∫ r in 𝓡, f ((r : G) * h) ∂μR) (∫ g in 𝓑, f g ∂μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralQuotientEqMeanIntermediateSubgroupStatement

theorem **_root_.GlobalTheory.T.integral_quotient_eq_mean_intermediate_subgroup** : ∀ (G : Type) [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (μ : Measure G) [μ.IsHaarMeasure] (R Γ : Subgroup G), Γ ≤ R → IsClosed (R : Set G) → Countable Γ → DiscreteTopology Γ → ∀ (μR : Measure R), μR.IsHaarMeasure → ∀ (𝓡 : Set R) (𝓑 : Set G), IsFundamentalDomain (Γ.subgroupOf R) 𝓡 μR → IsFundamentalDomain Γ 𝓑 μ → ∀ f : G → ℂ, Measurable f → (∀ γ ∈ Γ, ∀ g, f (γ * g) = f g) → IntegrableOn f 𝓑 μ → (∀ h, IntegrableOn (fun r : R => f (r * h)) 𝓡 μR) → IsQuotIntL1 μR μ (fun h => ∫ r in 𝓡, f ((r : G) * h) ∂μR) (∫ g in 𝓑, f g ∂μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralQuotientEqSetIntegralFundamentalDomain

theorem **integral_autMeasure_eq_setIntegral** : ∀ (D : D7) (V : HSp D) (ν : Measure (GA D V)) (X : Set (GA D V)), ν.IsHaarMeasure → IsFundamentalDomain (GQ V) X ν → ∀ F : GA D V → ℂ, (∀ γ ∈ GQ V, ∀ g, F (γ * g) = F g) → AEStronglyMeasurable F ν → ∫ q, GlobalTheory.AutomorphicL2Unitary.descend (GQ V) F q ∂(GlobalTheory.AutomorphicL2Unitary.autMeasure V ν) = ∫ g in X, F g ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralQuotientEqSetIntegralFundamentalDomainStatement

theorem **_root_.GlobalTheory.T.integral_quotient_eq_setIntegral_fundamentalDomain** : ∀ (D : D7) (V : HSp D) (ν : Measure (GA D V)) (X : Set (GA D V)), ν.IsHaarMeasure → IsFundamentalDomain (GQ V) X ν → ∀ F : GA D V → ℂ, (∀ γ ∈ GQ V, ∀ g, F (γ * g) = F g) → AEStronglyMeasurable F ν → ∫ q, GlobalTheory.AutomorphicL2Unitary.descend (GQ V) F q ∂(GlobalTheory.AutomorphicL2Unitary.autMeasure V ν) = ∫ g in X, F g ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralRealUnitaryHaarEqProdUpqHaar

theorem **continuous_psiT** (D : D7) (τ : D.what₂.Tau) : Continuous (psiT D τ)

theorem **fr_mul_inv** (τ : D.what₂.Tau) : M.fr τ * (M.fr τ)⁻¹ = 1

theorem **inv_mul_fr** (τ : D.what₂.Tau) : (M.fr τ)⁻¹ * M.fr τ = 1

theorem **ofFr_toJ** (τ : D.what₂.Tau) (g : Matrix (Fin V.rank) (Fin V.rank) ℂ) : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (toJ M τ g) = g

theorem **toJ_ofFr** (τ : D.what₂.Tau) (u : Matrix (Fin V.rank) (Fin V.rank) ℂ) : toJ M τ (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ u) = u

theorem **toJ_mul** (τ : D.what₂.Tau) (g h : Matrix (Fin V.rank) (Fin V.rank) ℂ) : toJ M τ (g * h) = toJ M τ g * toJ M τ h

theorem **toJ_one** (τ : D.what₂.Tau) : toJ M τ 1 = 1

theorem **inUJ_toJ** (τ : D.what₂.Tau) (g : Matrix (Fin V.rank) (Fin V.rank) ℂ) (hg : gᴴ * gramAt D V τ * g = gramAt D V τ) : Formalism.REAL.inUJ (M.ε τ) (toJ M τ g)

theorem **unit_ofFr** (τ : D.what₂.Tau) (u : Matrix (Fin V.rank) (Fin V.rank) ℂ) (hu : Formalism.REAL.inUJ (M.ε τ) u) : (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ u)ᴴ * gramAt D V τ * GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ u = gramAt D V τ

theorem **inUJ_umat** {n : ℕ} {ε : Fin n → Bool} (u : Formalism.REAL.UJ ε) : Formalism.REAL.inUJ ε (umat u)

theorem **infMat_mul** (τ : D.what₂.Tau) (g h : GA D V) : infMat D.sp.cm.sig V.gram (thetaC D τ) (g * h) = infMat D.sp.cm.sig V.gram (thetaC D τ) g * infMat D.sp.cm.sig V.gram (thetaC D τ) h

theorem **unitFam_ofFr** (us : ∀ τ, Formalism.REAL.UJ (M.ε τ)) : IsUnitFam D V fun τ => GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (umat (us τ))

theorem **continuous_matA** : Continuous (matA D V)

theorem **continuous_famM** : Continuous (famM D V)

theorem **continuous_fwd** : Continuous (fwd M)

theorem **continuous_bwd** : Continuous (bwd M)

theorem **ofFr_eqv** (g : GinfC V) (τ : D.what₂.Tau) : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (umat (eqv M g τ)) = infMat D.sp.cm.sig V.gram (thetaC D τ) (g : GA D V)

theorem **integral_prod_eq** : ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (θ : InfFam D), IsInfFam θ → ∀ (mR : Measure (GinfC V)) (μ : ∀ τ, Measure (Formalism.REAL.UJ (M.ε τ))), mR.IsHaarMeasure → (∀ τ, (μ τ).IsHaarMeasure) → ∃ c : ℝ, 0 < c ∧ ∀ F : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ → ℂ, (∀ τ, Integrable (fun u : Formalism.REAL.UJ (M.ε τ) => F τ (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (umat u))) (μ τ)) → Integrable (fun g : GinfC V => ∏ τ, F τ (Formalism.MainNotions.StandardLattice.infMat D.sp.cm.sig V.gram (θ τ) (g : GA D V))) mR ∧ ∫ g : GinfC V, ∏ τ, F τ (Formalism.MainNotions.StandardLattice.infMat D.sp.cm.sig V.gram (θ τ) (g : GA D V)) ∂mR = (c : ℂ) * ∏ τ, ∫ u : Formalism.REAL.UJ (M.ε τ), F τ (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (umat u)) ∂μ τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralRealUnitaryHaarEqProdUpqHaarStatement

theorem **_root_.GlobalTheory.T.integral_realUnitary_haar_eq_prod_Upq_haar** : ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (θ : InfFam D), IsInfFam θ → ∀ (mR : Measure (GinfC V)) (μ : ∀ τ, Measure (Formalism.REAL.UJ (M.ε τ))), mR.IsHaarMeasure → (∀ τ, (μ τ).IsHaarMeasure) → ∃ c : ℝ, 0 < c ∧ ∀ F : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ → ℂ, (∀ τ, Integrable (fun u : Formalism.REAL.UJ (M.ε τ) => F τ (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (umat u))) (μ τ)) → Integrable (fun g : GinfC V => ∏ τ, F τ (Formalism.MainNotions.StandardLattice.infMat D.sp.cm.sig V.gram (θ τ) (g : GA D V))) mR ∧ ∫ g : GinfC V, ∏ τ, F τ (Formalism.MainNotions.StandardLattice.infMat D.sp.cm.sig V.gram (θ τ) (g : GA D V)) ∂mR = (c : ℂ) * ∏ τ, ∫ u : Formalism.REAL.UJ (M.ε τ), F τ (GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.ofFr M τ (umat u)) ∂μ τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralSplitFinitePlaceFubini

theorem **integral_split_place** : ∀ (D : D7) (A : HSp D) (m : Measure (GA D A)), m.IsHaarMeasure → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (μ : Measure (GlobalTheory.FinZeta.Gv A v)), μ.IsHaarMeasure → ∀ J : Subgroup (GlobalTheory.FinZeta.Gv A v), IsOpen (J : Set (GlobalTheory.FinZeta.Gv A v)) → μ J = 1 → ∀ E : Set (GA D A), MeasurableSet E → (∀ x y, x * GlobalTheory.AdelicU.emb D.sp.cm.sig A.gram v y ∈ E ↔ x ∈ E) → ∀ (F : GA D A → ℂ) (ζ : GlobalTheory.FinZeta.Gv A v → ℂ), Integrable ζ μ → (∀ y ∈ J, ζ y = 1) → (∀ x ∈ E, GlobalTheory.AdelicU.proj D.sp.cm.sig A.gram v x = 1 → ∀ y, F (x * GlobalTheory.AdelicU.emb D.sp.cm.sig A.gram v y) = ζ y * F x) → IntegrableOn F (E ∩ {g | GlobalTheory.AdelicU.proj D.sp.cm.sig A.gram v g ∈ J}) m → IntegrableOn F E m ∧ ∫ g in E, F g ∂m = (∫ y, ζ y ∂μ) * ∫ g in E ∩ {g | GlobalTheory.AdelicU.proj D.sp.cm.sig A.gram v g ∈ J}, F g ∂m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralSplitFinitePlaceFubiniStatement

theorem **_root_.GlobalTheory.T.integral_split_finite_place_fubini** : ∀ (D : D7) (A : HSp D) (m : Measure (GA D A)), m.IsHaarMeasure → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (μ : Measure (GlobalTheory.FinZeta.Gv A v)), μ.IsHaarMeasure → ∀ J : Subgroup (GlobalTheory.FinZeta.Gv A v), IsOpen (J : Set (GlobalTheory.FinZeta.Gv A v)) → μ J = 1 → ∀ E : Set (GA D A), MeasurableSet E → (∀ x y, x * GlobalTheory.AdelicU.emb D.sp.cm.sig A.gram v y ∈ E ↔ x ∈ E) → ∀ (F : GA D A → ℂ) (ζ : GlobalTheory.FinZeta.Gv A v → ℂ), Integrable ζ μ → (∀ y ∈ J, ζ y = 1) → (∀ x ∈ E, GlobalTheory.AdelicU.proj D.sp.cm.sig A.gram v x = 1 → ∀ y, F (x * GlobalTheory.AdelicU.emb D.sp.cm.sig A.gram v y) = ζ y * F x) → IntegrableOn F (E ∩ {g | GlobalTheory.AdelicU.proj D.sp.cm.sig A.gram v g ∈ J}) m → IntegrableOn F E m ∧ ∫ g in E, F g ∂m = (∫ y, ζ y ∂μ) * ∫ g in E ∩ {g | GlobalTheory.AdelicU.proj D.sp.cm.sig A.gram v g ∈ J},

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralTranslate

theorem **integral_translate** {G : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [MeasurableSpace G] [BorelSpace G] (ν : Measure G) [ν.IsAddHaarMeasure] (F : G → ℂ) (a : G) : ∫ x, F (x + a) ∂ν = ∫ x, F x ∂ν

theorem **half_avg_heis_inv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (μ : Measure (Mx D T.rank ρ)) (𝓕 : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ 𝓕 → ∀ e, IsSplitIdem D e → ∀ ℓ₀ : (WO S₀ T → ℂ) →ₗ[ℂ] ℂ, (∀ u : WO S₀ T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S₀ T f → ℓ₀ (heisO ψ S₀ T u f) = ℓ₀ f) → ∀ H : AddSubgroup (Mx D T.rank ρ), (∀ x, x ∈ H ↔ ∀ i j, cA D e * x i j = x i j) → ∀ [MeasurableSpace H] [BorelSpace H] (ν : Measure H) [ν.IsAddHaarMeasure], ∀ u : WO S T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S T f → ∫ x : H, ℓ₀ (d.mixS T ψ μ 𝓕 (heisO ψ S T u f) x.1) ∂ν = ∫ x : H, ℓ₀ (d.mixS T ψ μ 𝓕 f x.1) ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralTsumNormalEigenfunctionsEqZero

theorem **measurable_tsum_of_summable** {X I : Type} [MeasurableSpace X] [Countable I] (F : I → X → ℂ) (hF : ∀ i, Measurable (F i)) (hs : ∀ x, Summable fun i => F i x) : Measurable fun x => ∑' i, F i x

theorem **summable_of_tsum_enorm_ne_top** {I : Type} (f : I → ℂ) (h : ∑' i, ‖f i‖ₑ ≠ ⊤) : Summable f

theorem **integrableOn_of_enorm_le** {X : Type} [MeasurableSpace X] {μ : Measure X} {s : Set X} (hs : μ s ≠ ⊤) {φ : X → ℂ} (hφ : Measurable φ) (B : NNReal) (hb : ∀ x, ‖φ x‖ₑ ≤ B) : IntegrableOn φ s μ

theorem **GlobalTheory.CutSum.integral_tsum_eq_zero** : ∀ (N : Type) [Group N] [TopologicalSpace N] [IsTopologicalGroup N] [LocallyCompactSpace N] [SecondCountableTopology N] [T2Space N] [MeasurableSpace N] [BorelSpace N] (ν : Measure N) [ν.IsHaarMeasure] (Γ M : Subgroup N), Countable Γ → DiscreteTopology Γ → M.Normal → IsClosed (M : Set N) → (∃ K : Set M, IsCompact K ∧ ∀ m, ∃ γ ∈ Γ.subgroupOf M, ∃ k ∈ K, m = γ * k) → ∀ 𝒩 : Set N, IsFundamentalDomain Γ 𝒩 ν → ν 𝒩 ≠ ⊤ → ∀ (I C : Type) [Countable I] (κ : I → C) (F : I → N → ℂ), (∀ i, Measurable (F i)) → (∃ B : NNReal, ∀ n, ∑' i, ‖F i n‖ₑ ≤ B) → (∀ γ ∈ Γ, ∀ n, ∑' i, F i (γ * n) = ∑' i, F i n) → (∀ c, ∀ γ ∈ Γ, γ ∈ M → ∀ n, ∑' i : {i // κ i = c}, F i (γ * n) = ∑' i : {i // κ i = c}, F i n) → (∀ c, ∃ u ∈ M, ∃ a : ℂ, a ≠ 1 ∧ ∀ i, κ i = c → ∀ m ∈ M, ∀ n, F i (m * u * n) = a * F i (m * n)) → ∫ n in 𝒩, ∑' i, F i n ∂ν = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralTsumNormalEigenfunctionsEqZeroStatement

theorem **_root_.GlobalTheory.T.integral_tsum_normal_eigenfunctions_eq_zero** : ∀ (N : Type) [Group N] [TopologicalSpace N] [IsTopologicalGroup N] [LocallyCompactSpace N] [SecondCountableTopology N] [T2Space N] [MeasurableSpace N] [BorelSpace N] (ν : Measure N) [ν.IsHaarMeasure] (Γ M : Subgroup N), Countable Γ → DiscreteTopology Γ → M.Normal → IsClosed (M : Set N) → (∃ K : Set M, IsCompact K ∧ ∀ m, ∃ γ ∈ Γ.subgroupOf M, ∃ k ∈ K, m = γ * k) → ∀ 𝒩 : Set N, IsFundamentalDomain Γ 𝒩 ν → ν 𝒩 ≠ ⊤ → ∀ (I C : Type) [Countable I] (κ : I → C) (F : I → N → ℂ), (∀ i, Measurable (F i)) → (∃ B : NNReal, ∀ n, ∑' i, ‖F i n‖ₑ ≤ B) → (∀ γ ∈ Γ, ∀ n, ∑' i, F i (γ * n) = ∑' i, F i n) → (∀ c, ∀ γ ∈ Γ, γ ∈ M → ∀ n, ∑' i : {i // κ i = c}, F i (γ * n) = ∑' i : {i // κ i = c}, F i n) → (∀ c, ∃ u ∈ M, ∃ a : ℂ, a ≠ 1 ∧ ∀ i, κ i = c → ∀ m ∈ M, ∀ n, F i (m * u * n) = a * F i (m * n)) → ∫ n in 𝒩, ∑' i, F i n ∂ν = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralUnitaryAdelicEqSumFiniteCosetsRealIntegr

theorem **integral_eq_of_lintegral_eq** {X Y : Type} [MeasurableSpace X] [MeasurableSpace Y] (μ : Measure X) (m : Measure Y) (f : X → ℂ) (g : Y → ℂ) (hf : Measurable f) (hg : Measurable g) (h : ∀ c : ℂ → ENNReal, Measurable c → ∫⁻ x, c (f x) ∂μ = ∫⁻ y, c (g y) ∂m) (hi : Integrable f μ) : Integrable g m ∧ ∫ x, f x ∂μ = ∫ y, g y ∂m

theorem **continuous_realPt** {θ : InfFam D} (hθ : IsInfFam θ) {W : HSp D} (hW : W.SigOK) {Fr : Frames W} (hFr : IsFrames W Fr) : Continuous fun b : (∀ ρ, Upq W ρ) => realElt θ W fun ρ => realMat W Fr ρ (b ρ)

theorem **finE_toFin** (W : HSp D) (h : GA D W) : finE W h = finE W (GlobalTheory.AutMult.toFin W h : GA D W)

theorem **integral_eq_cosetSum** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (W : HSp D), W.SigOK → ∀ (K : Subgroup (GL (Fin W.rank) (Lev.EAf D))) (ν : Measure (GA D W)), IsMeasLev W K ν → IsOpen {h : GA D W | finE W h ∈ K} → ∀ Fr : Frames W, IsFrames W Fr → ∀ mR : ∀ ρ, Measure (Upq W ρ), (∀ ρ, Lev.IsMinf (mR ρ)) → ∀ Z : GA D W → ℂ, Continuous Z → Integrable Z ν → (∀ g, ∀ k ∈ levA W K, Z (g * k) = Z g) → ∫⁻ b, ∑' x : FinCos W (levA W K), ‖Z (x.rep * realElt θ W fun ρ => realMat W Fr ρ (b ρ))‖ₑ ∂(Measure.pi mR) < ⊤ ∧ ∫ g, Z g ∂ν = ∫ b, ∑' x : FinCos W (levA W K), Z (x.rep * realElt θ W fun ρ => realMat W Fr ρ (b ρ)) ∂(Measure.pi mR)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralUnitaryAdelicEqSumFiniteCosetsRealIntegrStatement

theorem **_root_.GlobalTheory.T.integral_unitaryAdelic_eq_sum_finiteCosets_realIntegrals** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (W : HSp D), W.SigOK → ∀ (K : Subgroup (GL (Fin W.rank) (Lev.EAf D))) (ν : Measure (GA D W)), IsMeasLev W K ν → IsOpen {h : GA D W | finE W h ∈ K} → ∀ Fr : Frames W, IsFrames W Fr → ∀ mR : ∀ ρ, Measure (Upq W ρ), (∀ ρ, Lev.IsMinf (mR ρ)) → ∀ Z : GA D W → ℂ, Continuous Z → Integrable Z ν → (∀ g, ∀ k ∈ levA W K, Z (g * k) = Z g) → ∫⁻ b, ∑' x : FinCos W (levA W K), ‖Z (x.rep * realElt θ W fun ρ => realMat W Fr ρ (b ρ))‖ₑ ∂(Measure.pi mR) < ⊤ ∧ ∫ g, Z g ∂ν = ∫ b, ∑' x : FinCos W (levA W K), Z (x.rep * realElt θ W fun ρ => realMat W Fr ρ (b ρ)) ∂(Measure.pi mR)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralUnramifiedOutsideOpenSubsetInsertPlace

theorem **integral_openOff_insert** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ (m : Measure (GA D A)), m.IsHaarMeasure → ∀ (Sf : Finset (FinPl D)) (v : FinPl D), v ∉ Sf → ∀ μ : Measure (Gv A v), μ.IsHaarMeasure → μ (stabLoc D.sp.cm.sig A.gram L v) = 1 → ∀ (S' : Finset (FinPl D)), (∀ u, u ∈ S' ↔ u = v ∨ u ∈ Sf) → ∀ F : GA D A → ℂ, IntegrableOn F (openOff A L S') m → ∫ g in openOff A L S', F g ∂m = ∫ x in openOff A L Sf, (∫ y, F (x * (emb D.sp.cm.sig A.gram v (proj D.sp.cm.sig A.gram v x))⁻¹ * emb D.sp.cm.sig A.gram v y) ∂μ) ∂m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralUnramifiedOutsideOpenSubsetInsertPlaceStatement

theorem **_root_.GlobalTheory.T.integral_unramifiedOutsideOpenSubset_insert_place** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ (m : Measure (GA D A)), m.IsHaarMeasure → ∀ (Sf : Finset (FinPl D)) (v : FinPl D), v ∉ Sf → ∀ μ : Measure (Gv A v), μ.IsHaarMeasure → μ (stabLoc D.sp.cm.sig A.gram L v) = 1 → ∀ (S' : Finset (FinPl D)), (∀ u, u ∈ S' ↔ u = v ∨ u ∈ Sf) → ∀ F : GA D A → ℂ, IntegrableOn F (openOff A L S') m → ∫ g in openOff A L S', F g ∂m = ∫ x in openOff A L Sf, (∫ y, F (x * (emb D.sp.cm.sig A.gram v (proj D.sp.cm.sig A.gram v x))⁻¹ * emb D.sp.cm.sig A.gram v y) ∂μ) ∂m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralVectorBasesBoundedFiniteUnitaryClasses

theorem **trForm_denom** : ∀ (I : AutInst), ∃ c : ℕ, 0 < c ∧ ∀ x y : Fin I.n → I.E, IntVec I x → IntVec I y → ∃ m : ℤ, (c : ℚ) * trForm I x y = m

theorem **fix_one** (I : AutInst) {N : ℕ} (e : Fin N → Fin I.n → I.E) (A : Matrix (Fin I.n) (Fin I.n) I.E) (he : IsZBasis I e) (hA : ∀ j, A *ᵥ e j = e j) : A = 1

theorem **bases_finite_classes** : ∀ (I : AutInst) (N : ℕ) (R : ℝ), ∃ S : Set (Fin N → Fin I.n → I.E), S.Finite ∧ (∀ e₀ ∈ S, IsZBasis I e₀) ∧ ∀ e : Fin N → Fin I.n → I.E, IsZBasis I e → BddType I R e → ∃ e₀ ∈ S, ∃ γ : GL (Fin I.n) I.E, γ ∈ Formalism.UG.UJ (starRingEnd I.E) I.J ∧ (∀ i j, IsIntegral ℤ ((γ : Matrix (Fin I.n) (Fin I.n) I.E) i j)) ∧ (∀ i j, IsIntegral ℤ (((γ⁻¹ : GL (Fin I.n) I.E) : Matrix (Fin I.n) (Fin I.n) I.E) i j)) ∧ ∀ j, (γ : Matrix (Fin I.n) (Fin I.n) I.E) *ᵥ e j = e₀ j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntegralVectorBasesBoundedFiniteUnitaryClassesStatement

theorem **_root_.GlobalTheory.T.integralVectorBases_bounded_finite_unitaryClasses** : ∀ (I : AutInst) (N : ℕ) (R : ℝ), ∃ S : Set (Fin N → Fin I.n → I.E), S.Finite ∧ (∀ e₀ ∈ S, IsZBasis I e₀) ∧ ∀ e : Fin N → Fin I.n → I.E, IsZBasis I e → BddType I R e → ∃ e₀ ∈ S, ∃ γ : GL (Fin I.n) I.E, γ ∈ Formalism.UG.UJ (starRingEnd I.E) I.J ∧ (∀ i j, IsIntegral ℤ ((γ : Matrix (Fin I.n) (Fin I.n) I.E) i j)) ∧ (∀ i j, IsIntegral ℤ (((γ⁻¹ : GL (Fin I.n) I.E) : Matrix (Fin I.n) (Fin I.n) I.E) i j)) ∧ ∀ j, (γ : Matrix (Fin I.n) (Fin I.n) I.E) *ᵥ e j = e₀ j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InterEqSumReps

theorem **reps_bijective** {V : Type} [AddCommGroup V] (A₁ A₂ : AddSubgroup V) {ι : Type} (s : Finset ι) (z : ι → V) (hz : ∀ i ∈ s, z i ∈ A₂) (hinj : ∀ i ∈ s, ∀ j ∈ s, z i − z j ∈ A₁ → i = j) (hsurj : ∀ y ∈ A₂, ∃ i ∈ s, y − z i ∈ A₁) : Function.Bijective (fun i : s => (((z

i, hz i i.2) : A₂) : A₂ ⧸ A₁.addSubgroupOf A₂))

theorem **inter_eq_sum_reps** : ∀ {V : Type} [AddCommGroup V] [Module ℂ V] (H : Formalism.L2.Th1.H.Sesq V) (α : V → ℂ) (A₁ A₂ : AddSubgroup V), Formalism.L2.Th1.H.IsHerm H → Formalism.L2.Th1.H.IsSelfDual H A₂ → Formalism.L2.Th1.H.IsSemiChar H A₁ α → Formalism.L2.Th1.H.IsSemiChar H A₂ α → ∀ {ι : Type} (s : Finset ι) (z : ι → V), (∀ i ∈ s, z i ∈ A₂) → (∀ i ∈ s, ∀ j ∈ s, z i - z j ∈ A₁ → i = j) → (∀ y ∈ A₂, ∃ i ∈ s, y - z i ∈ A₁) → ∀ F ∈ Formalism.L2.Th1.H.fixedBy H α A₁, ∀ w, Formalism.L2.Th1.H.inter H α A₁ A₂ F w = ((Real.sqrt (s.card : ℝ) : ℝ) : ℂ)⁻¹ * ∑ i ∈ s, α (z i) * Formalism.L2.Th1.H.opU H (z i) F w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IntertwiningStabilizerTranslateScalar

theorem **exists_valuation_adapted_index** {T T' : HSp D} {t : ℕ} (d : HypDec T T' t) {p : GA D T} (hp : p ∈ stab d) : ∃ (β : Mx D t T'.rank) (g : GA D T'), mt T p · d.j0 = d.jX · β + d.j0 · mt T' g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvDiagTorusIntegralBesselJacobiUnramified

theorem **diag_inv** {ρ : ℕ} (t : GL (Fin ρ) (AE D)) (ht : ∀ i j, i ≠ j → t.1 i j = 0) : t.1 = diagonal (fun k => t.1 k k) ∧ (t⁻¹).1 = diagonal (fun k => (t⁻¹).1 k k) ∧ ∀ k, t.1 k k * (t⁻¹).1 k k = 1

theorem **mat_mem_idAt** {P : Pl D → Prop} {m n k l : ℕ} (A : Mx D m n) (M : Mx D n k) (B : Mx D k l) (hM : ∀ a b, M a b ∈ idAt D P) (i : Fin m) (j : Fin l) : (A · M · B) i j ∈ idAt D P

theorem **levi_sub_one_mem** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {P : Pl D → Prop} {t : GL (Fin ρ) (AE D)} (ht : d.levi t ∈ grpAt S P) (i j : Fin ρ) : t.1 i j - (1 : Mx D ρ ρ) i j ∈ idAt D P

theorem **symplO_extX_extY** {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (x y : Mx D T.rank ρ) : symplO S T (d.extX T x) (d.extY T y) = Algebra.trace (AF D.host.F) (AE D) (incA D D.host.xi0 * pairS T x y)

theorem **trace_single_mul** {ρ : ℕ} (i : Fin ρ) (β : AE D) (M : Mx D ρ ρ) : (Matrix.single i i β · M).trace = β * M i i

theorem **adj_single** {ρ : ℕ} (i : Fin ρ) (β : AE D) : adj (Matrix.single i i β) = Matrix.single i i (cA D β)

theorem **vac_fix_Y** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (LS : Lat S) (LT : Lat T) (v : Pl D), Tame D v → ∀ f, Vac ψ S T LS LT (· = v) f → ∀ y : Mx D T.rank ρ, d.extY T y ∈ latO S T (LS : Set _) (LT : Set _) (· = v) → heisO ψ S T (d.extY T y) f = f

theorem **inv_int** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → UnrAt s v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → ∀ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) → J.d.levi t ∈ grpAt J.W (· = v) → 𝔉 (s.om (1, J.d.levi t) f) ≠ 0 → ∀ i, (t⁻¹).1 i i - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvDiagTorusIntegralBesselJacobiUnramifiedStatement

theorem **_root_.GlobalTheory.T.inv_diag_torus_integral_besselJacobi_unramified** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → UnrAt s v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → ∀ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) → J.d.levi t ∈ grpAt J.W (· = v) → 𝔉 (s.om (1, J.d.levi t) f) ≠ 0 → ∀ i, (t⁻¹).1 i i - 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvHaarVolumeToRealEqRelIndexIntegral

theorem **inv_toReal_eq_relIndex** {Γ : Type*} [Group Γ] [MeasurableSpace Γ] (μ : Measure Γ) [μ.IsMulLeftInvariant] (K H : Subgroup Γ) (hHK : H ≤ K) (hH : MeasurableSet (H : Set Γ)) (hK : μ (K : Set Γ) = 1) (hn : H.relIndex K ≠ 0) : ((μ (H : Set Γ)).toReal)⁻¹ = (H.relIndex K : ℝ)

theorem **QTw.QExt.m_inv_toReal_eq_relIndex** : ∀ {D : D7} {c : Comm D} {v : VCom c} {B : GlobalTheory.Fam.Base c} {PP : Formalism.L2.DIV.Pl D.host.E} (G : GlobalTheory.QTw.QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle) (E : GlobalTheory.QTw.QExt G χ) (j : ℕ) (H : Subgroup (GL (XIx (Fin D.R.k) (Fin D.R.t) j) G.k)), H ≤ KO G.k _ → MeasurableSet (H : Set (GL (XIx (Fin D.R.k) (Fin D.R.t) j) G.k)) → H.relIndex (KO G.k _) ≠ 0 → ((((E.m j H).toReal)⁻¹ : ℝ) : ℂ) = (H.relIndex (KO G.k _) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvHaarVolumeToRealEqRelIndexIntegralStatement

theorem **_root_.GlobalTheory.T.inv_haarVolume_toReal_eq_relIndex_integral** : ∀ {D : D7} {c : Comm D} {v : VCom c} {B : GlobalTheory.Fam.Base c} {PP : Formalism.L2.DIV.Pl D.host.E} (G : GlobalTheory.QTw.QGeo v B PP) (χ : ℕ → (AE D)ˣ →* Circle) (E : GlobalTheory.QTw.QExt G χ) (j : ℕ) (H : Subgroup (GL (XIx (Fin D.R.k) (Fin D.R.t) j) G.k)), H ≤ KO G.k _ → MeasurableSet (H : Set (GL (XIx (Fin D.R.k) (Fin D.R.t) j) G.k)) → H.relIndex (KO G.k _) ≠ 0 → ((((E.m j H).toReal)⁻¹ : ℝ) : ℂ) = (H.relIndex (KO G.k _) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvMulVecStarMemDual

theorem **star_mem_one_O** {K : Type} [Field K] [NumberField K] [StarRing K] {a : K} (ha : a ∈ (1 : Submodule (𝒪 K) K)) : star a ∈ (1 : Submodule (𝒪 K) K)

theorem **hform_eq_dot** {K : Type} [Field K] [NumberField K] [StarRing K] {n : ℕ} (H : Matrix (Fin n) (Fin n) K) (a b : Fin n → K) : hform H a b = star a ⬝ᵥ H *ᵥ b

theorem **inv_mulVec_star_mem_dual** {K : Type} [Field K] [NumberField K] [StarRing K] {n : ℕ} (H : Matrix (Fin n) (Fin n) K) (hH : IsUnit H.det) (M : Submodule (𝒪 K) (Fin n → K)) (y : Fin n → K) (hy : ∀ x ∈ M, y ⬝ᵥ x ∈ (1 : Submodule (𝒪 K) K)) : H⁻¹ *ᵥ star y ∈ Formalism.HermitianLatticesTheory.dual (hform H) M

theorem **dual_form_integral** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] (n : ℕ) (H : Matrix (Fin n) (Fin n) K), Hᴴ = H → IsUnit H.det → ∀ M : Submodule (𝒪 K) (Fin n → K), Formalism.HermitianLatticesTheory.dual (hform H) M ≤ M → ∀ y y' : Fin n → K, (∀ x ∈ M, y ⬝ᵥ x ∈ (1 : Submodule (𝒪 K) K)) → (∀ x ∈ M, y' ⬝ᵥ x ∈ (1 : Submodule (𝒪 K) K)) → hform (H⁻¹)ᵀ y y' ∈ (1 : Submodule (𝒪 K) K)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantIntegralInStages

theorem **lintegral_eq_cutoff_inner** {T : Subgroup G} {μ : Measure G} {μT : Measure T} [SFinite μ] [μ.IsMulLeftInvariant] [SFinite μT] [μT.IsInvInvariant] {β : G → NNReal} (hβ : IsCosetCutoff T μT β) {H : G → ℝ≥0∞} (hH : Measurable H) : ∫⁻ x, H x ∂μ = ∫⁻ x, (β x : ℝ≥0∞) * ∫⁻ t : T, H (t * x) ∂μT ∂μ

theorem **map_val_eq_map_stages** {R₀ R : Subgroup G} (μ₀ : Measure R₀) (μR : Measure R) [SFinite μ₀] {U : Type} [MeasurableSpace U] (ν : Measure U) {j : U → G} (hj : Measurable j) (hst : ∀ F : G → ℝ≥0∞, Measurable F → ∫⁻ r : R, F r ∂μR = ∫⁻ u, ∫⁻ r : R₀, F (r * j u) ∂μ₀ ∂ν) : μR.map Subtype.val = (ν.prod μ₀).map (fun p : U × R₀ => (p.2 : G) * j p.1)

theorem **invariant_integral_in_stages** : ∀ (G : Type) [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (μ : Measure G) [μ.IsHaarMeasure] (R₀ R : Subgroup G),

IsClosed (R₀ : Set G) → IsClosed (R : Set G) → ∀ (μ₀ : Measure R₀) (μR : Measure R), μ₀.IsHaarMeasure → μ₀.IsInvInvariant → μR.IsHaarMeasure → μR.IsInvInvariant → ∀ (U : Type) [MeasurableSpace U] (ν : Measure U) [SFinite ν] (j : U → G), Measurable j → (∀ F : G → ENNReal, Measurable F → ∫⁻ r : R, F r ∂μR = ∫⁻ u, ∫⁻ r : R₀, F (r * j u) ∂μ₀ ∂ν) → ∀ f : G → ℂ, Measurable f → (∀ r ∈ R₀, ∀ g, f (r * g) = f g) → (∀ g, Integrable (fun u => f (j u * g)) ν) → (∃ z', IsQuotIntL1 μR μ (fun g => ((∫ u, ‖f (j u * g)‖ ∂ν : ℝ) : ℂ)) z') → ∀ z, IsQuotIntL1 μR μ (fun g => ∫ u, f (j u * g) ∂ν) z → IsQuotIntL1 μ₀ μ f z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantIntegralInStagesStatement

theorem **_root_.GlobalTheory.T.invariant_integral_in_stages_alt** : ∀ (G : Type) [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] (μ : Measure G) [μ.IsHaarMeasure] (R₀ R : Subgroup G), IsClosed (R₀ : Set G) → IsClosed (R : Set G) → ∀ (μ₀ : Measure R₀) (μR : Measure R), μ₀.IsHaarMeasure → μ₀.IsInvInvariant → μR.IsHaarMeasure → μR.IsInvInvariant → ∀ (U : Type) [MeasurableSpace U] (ν : Measure U) [SFinite ν] (j : U → G), Measurable j → (∀ F : G → ENNReal, Measurable F → ∫⁻ r : R, F r ∂μR = ∫⁻ u, ∫⁻ r : R₀, F (r * j u) ∂μ₀ ∂ν) → ∀ f : G → ℂ, Measurable f → (∀ r ∈ R₀, ∀ g, f (r * g) = f g) → (∀ g, Integrable (fun u => f (j u * g)) ν) → (∃ z', IsQuotIntL1 μR μ (fun g => ((∫ u, ‖f (j u * g)‖ ∂ν : ℝ) : ℂ)) z') → ∀ z, IsQuotIntL1 μR μ (fun g => ∫ u, f (j u * g) ∂ν) z → IsQuotIntL1 μ₀ μ f z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantIntegralSuccPositionEqExchange

theorem **rr_step** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (_ : 1 ≤ p) (_ : p < c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B B₁ : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ ρ σ B) (_ : ∀ Ψ v, B₁ Ψ v = exUp ν ρ σ (B Ψ) v) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ') (Rr : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] ℂ) (_ : ∀ Φ : SB (Matrix

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantIntegralSuccPositionEqExchangeStatement

theorem **_root_.GlobalTheory.T.invariantIntegral_succ_position_eq_exchange** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (_ : 1 ≤ p) (_ : p < c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B B₁ : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ ρ σ B) (_ : ∀ Ψ v, B₁ Ψ v = exUp ν ρ σ (B Ψ) v) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ') (Rr : SB (Matrix (n ⊕ Fi

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantOfTsupport

theorem **ind_mem** (h : IsBichar χ) (hN : ∀ y, ∀ Φ ∈ N, (fun t => χ t y * Φ t) ∈ N) (hsep : ∀ u, u ∉ K → ∃ y, (∀ κ ∈ K, χ κ y = 1) ∧ χ u y ≠ 1) (R : Finset G) : ∀ Φ ∈ N, (∀ u ∈ K, ∀ t, Φ (t + u) = Φ t) → Φ 0 ≠ 0 → (∀ t, Φ t ≠ 0 → t ∈ K ∨ ∃ r ∈ R, t − r ∈ K) → ind K ∈ N

theorem **mem_of_ind_mem** (hN : ∀ x, ∀ Φ ∈ N, (fun t => Φ (t + x)) ∈ N) (hK : ind K ∈ N) (R : Finset G) : ∀ Φ : G → ℂ, (∀ u ∈ K, ∀ t, Φ (t + u) = Φ t) → (∀ t, Φ t ≠ 0 → ∃ r ∈ R, t − r ∈ K) → Φ ∈ N

theorem **exists_map_wave₂** {V : Type} [AddCommGroup V] [Module ℂ V] (h : IsBichar χ) {L L' : AddSubgroup G} (hLL' : ∀ u ∈ L, ∀ y ∈ L', χ u y = 1) (hY : ∀ y, (∀ u ∈ L, χ u y = 1) → y ∈ L') (ρ σ : G → V →ₗ[ℂ] V) (hρ : ∀ x x' f, ρ x (ρ x' f) = ρ (x + x') f) (hσ : ∀ x x' f, σ x (σ x' f) = σ (x + x') f) (hc : ∀ x y f, ρ x (σ y f) = χ x y • σ y (ρ x f)) (f₀ : V) (hf : ∀ u ∈ L, ρ u f₀ = f₀) (hf' : ∀ u ∈ L', σ u f₀ = f₀) : ∃ T : (G → ℂ) →ₗ[ℂ] V, ∀ x y, T (wave χ L x y) = ρ x (σ y f₀)

theorem **eventually_invariant** {Φ : Matrix Y Y' k → ℂ} (hΦ : Φ ∈ SB (Matrix Y Y' k)) : ∀ᶠ u in 𝒩 (0 : Matrix Y Y' k), ∀ t ∈ tsupport Φ, Φ (t + u) = Φ t

theorem **invariant_of_tsupport** {Φ : Matrix Y Y' k → ℂ} {K : AddSubgroup (Matrix Y Y' k)} (h : ∀ u ∈ K, ∀ t ∈ tsupport Φ, Φ (t + u) = Φ t) (u : Matrix Y Y' k) (hu : u ∈ K) (t : Matrix Y Y' k) : Φ (t + u) = Φ t

theorem **exists_cosets** {Φ : Matrix Y Y' k → ℂ} (hΦ : Φ ∈ SB (Matrix Y Y' k)) (a : kˣ) : ∃ R : Finset (Matrix Y Y' k), ∀ t, Φ t ≠ 0 → ∃ r ∈ R, t − r ∈ box Y Y' a

theorem **sb_le** {ψ : AddChar k Circle} {d : kˣ} (hd : HasConductor ψ d) {N : Submodule ℂ (Matrix Y Y' k → ℂ)} (hN : ∀ x, ∀ Φ ∈ N, (fun t => Φ (t + x)) ∈ N) (hN' : ∀ y, ∀ Φ ∈ N, (fun t => pairing ψ t y * Φ t) ∈ N) {Φ : Matrix Y Y' k → ℂ} (hΦ : Φ ∈ SB (Matrix Y Y' k)) (h0 : Φ ≠ 0) (hΦN : Φ ∈ N) : SB (Matrix Y Y' k) ≤ N

theorem **ind_box_mem** (a : kˣ) : ind (box Y Y' a) ∈ SB (Matrix Y Y' k)

theorem **exists_embedding** {V : Type} [AddCommGroup V] [Module ℂ V] {ψ : AddChar k Circle} (hψ : Continuous ψ) (h1 : ψ ≠ 1) (ρ σ : Matrix Y Y' k → V →ₗ[ℂ] V) (hρ : ∀ x x' f, ρ x (ρ x' f) = ρ (x + x') f) (hσ : ∀ x x' f, σ x (σ x' f) = σ (x + x') f) (hc : ∀ x y f, ρ x (σ y f) = pairing ψ x y • σ y (ρ x f)) (P : Submodule ℂ V) (hP : ∀ x, ∀ f ∈ P, ρ x f ∈ P) (hP' : ∀ y, ∀ f ∈ P, σ y f ∈ P) {g : V} (hg : g ∈ P) (g0 : g ≠ 0) (hg1 : ∀ x ∈ box Y Y' (1 : kˣ), ρ x g = g) (hg2 : ∀ y, (∀ x ∈ box Y Y' (1 : kˣ), pairing ψ x y = 1) → σ y g = g) : ∃ T : (Matrix Y Y' k → ℂ) →ₗ[ℂ] V, ∀ Φ ∈ SB (Matrix Y Y' k), (T Φ = 0 → Φ = 0) ∧ T Φ ∈ P ∧ (∀ x, T (fun t => Φ (t + x)) = ρ x (T Φ)) ∧ ∀ y, T (fun t => pairing ψ t y * Φ t) = σ y (T Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantSphericalCoefficientEqZeroOrZonal

theorem **apply_heckeVec** {G V : Type} [Group G] [AddCommGroup V] [Module ℂ V] {K : Subgroup G} (hK : IsCommensurated K) {h : G → ℂ} (hh : h ∈ Formalism.UG.Hecke ℂ K) (ρ : Representation ℂ G V) (x : V) (l : V →ₗ[ℂ] ℂ) : l (GlobalTheory.LFun.heckeVec ρ K h x) = ∑ᶠ c : G ⧸ K, h c.out * l (ρ c.out x)

theorem **eq_zero_of_doset_sum** {G : Type} [Group G] {K : Subgroup G} (hK : IsCommensurated K) {w : G → ℂ} (hw : IsBiInv K w) (g : G) (e : ∑ᶠ c : G ⧸ K, dosetInd ℂ K g c.out * w c.out = 0) : w g = 0

theorem **coeff_zero_or_zonal** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : Lˣ) (σ : L →+* L) (q n : ℕ), IsUnramInv (ϖ : L) σ q → ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (ρ : Representation ℂ (GU σ n) V) (x : V) (α : Fin (n / 2) → ℂˣ), IsSatVecU (ϖ : L) q ρ x α → ∀ ℓ : V →ₗ[ℂ] ℂ, (∀ k ∈ KU σ n, ∀ g, ℓ (ρ (k * g) x) = ℓ (ρ g x)) → (ℓ x = 0 → ∀ g, ℓ (ρ g x) = 0) ∧ (ℓ x ≠ 0 → GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsZonal (ϖ : L) q α fun g => ℓ (ρ g x) / ℓ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InvariantSphericalCoefficientEqZeroOrZonalStatement

theorem **_root_.GlobalTheory.T.invariant_sphericalCoefficient_eq_zero_or_zonal** : ∀ (L : Type) [Field L] [Valued L (WithZero (Multiplicative ℤ))] [CompleteSpace L] (ϖ : Lˣ) (σ : L →+* L) (q n : ℕ), IsUnramInv (ϖ : L) σ q → ∀ (V : Type) [AddCommGroup V] [Module

ℂ V] (ρ : Representation ℂ (GU σ n) V) (x : V) (α : Fin (n / 2) → ℂˣ), IsSatVecU (ϖ : L) q ρ x α → ∀ ℓ : V →ₗ[ℂ] ℂ, (∀ k ∈ KU σ n, ∀ g, ℓ (ρ (k * g) x) = ℓ (ρ g x)) → (ℓ x = 0 → ∀ g, ℓ (ρ g x) = 0) ∧ (ℓ x ≠ 0 → GlobalTheory.IndefiniteRowSideFullHecke.SphericalSectionVariables.IsZonal (ϖ : L) q α fun g => ℓ (ρ g x) / ℓ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InverseDifferentCompletionIffCharacterDual

theorem **mem_dI_iff** (x : D.host.E) : x ∈ dI D ↔ TrInt D x

theorem **TrInt.mul** {x : D.host.E} (h : TrInt D x) (o : 𝒪 D.host.E) : TrInt D (x * o)

theorem **trInt_coe** (o : 𝒪 D.host.E) : TrInt D (o : D.host.E)

theorem **chi_tmul** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {β : Af D.host.F} (hβ : β ∈ intAf D.host.F) {x : D.host.E} (hx : TrInt D x) : Conductor.chi ψ (β ⊗ₜ[D.host.F] x) = 1

theorem **chi_mul_int** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {x : D.host.E} (hx : TrInt D x) {z : Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ intAE D) : Conductor.chi ψ (((1 : Af D.host.F) ⊗ₜ[D.host.F] x) * z) = 1

theorem **psiE_add** (ψ : AddChar (AF D.host.F) ℂ) (a b : AE D) : psiE ψ (a + b) = psiE ψ a * psiE ψ b

theorem **psiE_zero** (ψ : AddChar (AF D.host.F) ℂ) : psiE ψ (0 : AE D) = 1

theorem **psiE_neg** (ψ : AddChar (AF D.host.F) ℂ) (a : AE D) : psiE ψ (-a) = (psiE ψ a)⁻¹

theorem **forward** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (v : Pl D) {s : AE D} (hs : s ∈ idV (dI D) v) : ∀ o ∈ intAt D (· = v), psiE ψ (s * o) = 1

theorem **lift_tmul** (α : Af D.host.F) (x : D.host.E) : lift D (α ⊗ₜ[D.host.F] x) = pairA D.host.F 0 α ⊗ₜ[D.host.F] x

theorem **finPart_lift** (z : Af D.host.F ⊗[D.host.F] D.host.E) : finPart D (lift D z) = z

theorem **infPart_lift** (z : Af D.host.F ⊗[D.host.F] D.host.E) : infPart D (lift D z) = 0

theorem **eq_lift** {s : AE D} (h : infPart D s = 0) : s = lift D (finPart D s)

theorem **lift_int_mem** (v : Pl D) (hε : singleV D.host.F v 1 ∈ intAf D.host.F) {u : Af D.host.F ⊗[D.host.F] D.host.E} (hu : u ∈ intAE D) : lift D ((singleV D.host.F v 1 ⊗ₜ[D.host.F] (1 : D.host.E)) * u) ∈ idV (dI D) v

theorem **backward** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (v : Pl D) {s : AE D} (hs : s ∈ idAt D (· = v)) (h : ∀ o ∈ intAt D (· = v), psiE ψ (s * o) = 1) : s ∈ idV (dI D) v

theorem **idV_dI_iff** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (v : Pl D), ∀ s ∈ idAt D (· = v), s ∈ idV (dI D) v ↔ ∀ o ∈ intAt D (· = v), psiE ψ (s * o) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InverseDifferentCompletionIffCharacterDualStatement

theorem **_root_.GlobalTheory.T.inverseDifferentCompletion_iff_characterDual** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (v : Pl D), ∀ s ∈ idAt D (· = v), s ∈ idV (dI D) v ↔ ∀ o ∈ intAt D (· = v), psiE ψ (s * o) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InverseDifferentPowCompletionTrivialAtSplitPlace

theorem **eq_one_iff** {𝔞 : FI D} {v : GlobalTheory.FinPl.Pl D} : idV 𝔞 v = idV 1 v ↔ ∀ s, s ∈ idV 𝔞 v ↔ incA D (1 : D.host.E)⁻¹ * s ∈ intAt D (· = v) ∧ s ∈ idAt D (· = v)

theorem **mul_eq_one** {𝔞 𝔟 : FI D} {v : GlobalTheory.FinPl.Pl D} (h1 : idV 𝔞 v = idV 1 v) (h2 : idV 𝔟 v = idV 1 v) : idV (𝔞 * 𝔟) v = idV 1 v

theorem **pow_eq_one** {𝔞 : FI D} {v : GlobalTheory.FinPl.Pl D} (h : idV 𝔞 v = idV 1 v) (n : ℕ) : idV (𝔞 ^ n) v = idV 1 v

theorem **inv_eq_one** {𝔞 : FI D} {v : GlobalTheory.FinPl.Pl D} (h : idV 𝔞 v = idV 1 v) : idV 𝔞⁻¹ v = idV 1 v

theorem **AAbar_eq_one** {v : GlobalTheory.FinPl.Pl D} (hA : idV D.host.AA v = idV 1 v) : idV D.host.AAbar v = idV 1 v

theorem **fractionalIdealCompletion_xi0Span_eq_unit_at_splitPlace** (p : SplitPt D k) (hp : p.UnitAA) : idV (FractionalIdeal.spanSingleton _ D.host.xi0) p.v = idV 1 p.v

theorem **AA_eq_one** (p : SplitPt D k) (hp : p.UnitAA) : idV D.host.AA p.v = idV 1 p.v

theorem **GlobalTheory.Core.Mi.LatMatch.idV_dI_pow_unit** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k), p.UnitAA → ∀ (n : ℕ) (a : AE D), (a ∈ idV (dI D ^ n) p.v ↔ a ∈ intAt D (· = p.v)) ∧ (a ∈ idV (dI D ^ n)⁻¹ p.v ↔ a ∈ intAt D (· = p.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InverseDifferentPowCompletionTrivialAtSplitPlaceStatement

theorem **_root_.GlobalTheory.T.inverseDifferentPow_completion_trivial_at_splitPlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k), p.UnitAA → ∀ (n : ℕ) (a : AE D), (a ∈ idV (dI D ^ n) p.v ↔ a ∈ intAt D (· = p.v)) ∧ (a ∈ idV (dI D ^ n)⁻¹ p.v ↔ a ∈ intAt D (· = p.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InverseRowFourierTransformMemSchwartzBruhat

theorem **finv_mem_SB** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (ψk : AddChar k Circle) (ν : Measure k), Continuous ψk → ψk ≠ 1 → ν.IsAddHaarMeasure → ∀ (C : ℂ) (Φ : Matrix Unit n k → ℂ), Φ ∈ SB (Matrix Unit n k) → GlobalTheory.KappaModel.finv ψk ν C Φ ∈ SB (Matrix n Unit k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.InverseRowFourierTransformMemSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.inverseRowFourierTransform_mem_schwartzBruhat** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (ψk : AddChar k Circle) (ν : Measure k), Continuous ψk → ψk ≠ 1 → ν.IsAddHaarMeasure → ∀ (C : ℂ) (Φ : Matrix Unit n k → ℂ), Φ ∈ SB (Matrix Unit n k) → GlobalTheory.KappaModel.finv ψk ν C Φ ∈ SB (Matrix n Unit k)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IrreducibleKStableFiniteSubspace

theorem **admissible_and_kFin_smooth** : ∀ (n : ℕ) (ε : Fin n → Bool) (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)), IsURepOn (UJset ε) σ → IsIrredOn (UJset ε) σ → (∀ S : Submodule ℂ H, FiniteDimensional ℂ S → IsKStableOn ε σ S → ∀ v ∈ S, ∃ Y : Submodule ℂ H, FiniteDimensional ℂ Y ∧ v ∈ Y ∧ ∀ B : H →L[ℂ] H, (∀ k ∈ Formalism.REAL.Kset ε, ∀ u, B (σ k u) = σ k (B u)) → ∀ y ∈ Y, B y ∈ Y) ∧ ∀ v, IsKFin ε σ v → Formalism.QuantumDeterminants.SmoothOnH (Formalism.REAL.Jm ε) (fun g => σ g v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IrreducibleKStableFiniteSubspaceStatement

theorem **_root_.GlobalTheory.T.irreducible_K_stable_finite_subspace** : ∀ (n : ℕ) (ε : Fin n → Bool) (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (σ : Matrix (Fin n) (Fin n) ℂ → (H →L[ℂ] H)), IsURepOn (UJset ε) σ → IsIrredOn (UJset ε) σ → (∀ S : Submodule ℂ H, FiniteDimensional ℂ S → IsKStableOn ε σ S → ∀ v ∈ S, ∃ Y : Submodule ℂ H, FiniteDimensional ℂ Y ∧ v ∈ Y ∧ ∀ B : H →L[ℂ] H, (∀ k ∈ Formalism.REAL.Kset ε, ∀ u, B (σ k u) = σ k (B u)) → ∀ y ∈ Y, B y ∈ Y) ∧ ∀ v, IsKFin ε σ v → Formalism.QuantumDeterminants.SmoothOnH (Formalism.REAL.Jm ε) (fun g => σ g v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsAtTauChart

theorem **isUnit_det_map** {n : ℕ} {R : Type} [CommRing R] (f : D.host.E →+* R) {T : Matrix (Fin n) (Fin n) D.host.E} (hT : IsUnit T.det) : IsUnit (T.map f).det

theorem **isAtTau_iff** (θ : InfFam D) (V : HSp D) (τ : D.what₂.Tau) (x : Matrix (Fin V.rank) (Fin V.rank) ℂ) (h : GA D V) : IsAtTau θ V τ x h ↔ compAt θ τ (mt V h) = x ∧ (∀ τ', τ' ≠ τ → compAt θ τ' (mt V h) = 1) ∧ (mt V h).map (finPart D) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsBiThetaIffJoin

theorem **finPart_conj** (z : AE D) : finPart D (cA D z) = sigR (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F) D.sp.cm.sig (finPart D z)

theorem **infPart_conj** (z : AE D) : infPart D (cA D z) = sigR (Ainf D.host.F) D.sp.cm.sig (infPart D z)

theorem **fin_dagger** (S T : HSp D) {x : WO S T} (hx : x.map (finPart D) = 0) : (dagger S T x).map (finPart D) = 0

theorem **inf_dagger** (S T : HSp D) {x : WO S T} (hx : x.map (infPart D) = 0) : (dagger S T x).map (infPart D) = 0

theorem **dagger_add** (S T : HSp D) (x y : WO S T) : dagger S T (x + y) = dagger S T x + dagger S T y

theorem **dagger_zero** (S T : HSp D) : dagger S T 0 = 0

theorem **continuous_dagger** (S T : HSp D) : Continuous (dagger S T)

theorem **dagger_rat** (S T : HSp D) (γ : Matrix (Fin T.rank) (Fin S.rank) D.host.E) : dagger S T (rat γ) = rat (S.gram⁻¹ ∗ (γᴴ ∗ T.gram))

theorem **giA_negSp** (S : HSp D) : giA (negSp S) = −giA S

theorem **symplOH_dagger** (S T : HSp D) (x y : WO T S) : symplOH S T (dagger T S x) (dagger T S y) = symplOH (negSp T) S x y

theorem **fstH_apply** (x : WO W S) : fstH d S x = (d.resS S x).1

theorem **sndH_apply** (x : WO W S) : sndH d S x = dagger A' S (d.resS S x).2

theorem **inlH_apply** (u : WO A S) : inlH d S u = (d.resS S).symm (u, 0)

theorem **inrH_apply** (v : WO S A') : inrH d S v = (d.resS S).symm (0, dagger S A' v)

theorem **fst_inl** (u : WO A S) : fstH d S (inlH d S u) = u

theorem **fst_inr** (v : WO S A') : fstH d S (inrH d S v) = 0

theorem **snd_inl** (u : WO A S) : sndH d S (inlH d S u) = 0

theorem **snd_inr** (v : WO S A') : sndH d S (inrH d S v) = v

theorem **continuous_fstH** : Continuous (fstH d S)

theorem **continuous_sndH** : Continuous (sndH d S)

theorem **continuous_inlH** : Continuous (inlH d S)

theorem **continuous_inrH** : Continuous (inrH d S)

theorem **symplOH_split** (x y : WO W S) : symplOH W S x y = symplOH A S (fstH d S x) (fstH d S y) + symplOH S A' (sndH d S x) (sndH d S y)

theorem **map_resS_symm** {B : Type} [CommRing B] (φ : AE D →+* B) {u : WO A S} {v : WO (negSp A') S} (hu : u.map φ = 0) (hv : v.map φ = 0) : ((d.resS S).symm (u, v)).map φ = 0

theorem **map_inlH** {B : Type} [CommRing B] (φ : AE D →+* B) {u : WO A S} (hu : u.map φ = 0) : (inlH d S u).map φ = 0

theorem **fin_inrH** {v : WO S A'} (hv : v.map (finPart D) = 0) : (inrH d S v).map (finPart D) = 0

theorem **inf_inrH** {v : WO S A'} (hv : v.map (infPart D) = 0) : (inrH d S v).map (infPart D) = 0

theorem **join_apply** (C : WO A S → WO S A' → ℂ) (x : WO W S) : join d S C x = C (fstH d S x) (sndH d S x)

theorem **join_symm** (C : WO A S → WO S A' → ℂ) (u : WO A S) (y : WO (negSp A') S) : join d S C ((d.resS S).symm (u, y)) = C u (dagger A' S y)

theorem **join_inl_inr** (C : WO A S → WO S A' → ℂ) (u : WO A S) (v : WO S A') : join d S C (inlH d S u + inrH d S v) = C u v

theorem **isThetaFunO_join** (h : IsBiTheta ψ A S A' C) : IsThetaFunO ψ W S (join d S C) where

theorem **cont_of_join** (h : IsThetaFunO ψ W S (join d S C)) : Continuous fun p : WO A S × WO S A' => C p.1 p.2

theorem **smooth_of_join** (h : IsThetaFunO ψ W S (join d S C)) : IsBiSmooth C

theorem **quasiL_of_join** (h : IsThetaFunO ψ W S (join d S C)) (γ) (u : WO A S) (v : WO S A') : C (ratW S A γ + u) v = ψ (−(symplOH A S (ratW S A γ) u)) ∗ C u v

theorem **quasiR_of_join** (h : IsThetaFunO ψ W S (join d S C)) (γ) (u : WO A S) (v : WO S A') : C u (ratW A' S γ + v) = ψ (−(symplOH S A' (ratW A' S γ) v)) ∗ C u v

theorem **levL_of_join** (h : IsThetaFunO ψ W S (join d S C)) : ∃ U : AddSubgroup (WO A S), IsOpen (U : Set (WO A S)) ∧ ∀ a ∈ U, a.map (infPart D) = 0 → ∀ v, heisO ψ A S a (fun u => C u v) = fun u => C u v

theorem **levR_of_join** (h : IsThetaFunO ψ W S (join d S C)) : ∃ U : AddSubgroup (WO S A'), IsOpen (U : Set (WO S A')) ∧ ∀ a ∈ U, a.map (infPart D) = 0 → ∀ u, heisO ψ S A' a (C u) = C u

theorem **isBiTheta_of_join** (h : IsThetaFunO ψ W S (join d S C)) : IsBiTheta ψ A S A' C

theorem **isBiTheta_iff_join** : IsBiTheta ψ A S A' C ↔ IsThetaFunO ψ W S (join d S C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsBicharPairing

theorem **exists_fixed** {G V : Type} [AddCommGroup G] [AddCommGroup V] [Module ℂ V] {χ : G → G → ℂ} (h : IsBichar χ) (ρ σ : G → V →ₗ[ℂ] V) (hρ : ∀ x x' f, ρ x (ρ x' f) = ρ (x + x') f) (hσ : ∀ y f, σ y f = 0 → f = 0) (hc : ∀ x y f, ρ x (σ y f) = χ x y • σ y (ρ x f)) (L K : AddSubgroup G) [Finite (L ⧸ K.addSubgroupOf L)] (hsep : ∀ u ∈ L, u ∉ K → ∃ y, (∀ κ ∈ K, χ κ y = 1) ∧ χ u y ≠ 1) (P : Submodule ℂ V) (hP

: ∀ u ∈ L, ∀ f ∈ P, ρ u f ∈ P) (hP' : ∀ y, ∀ f ∈ P, σ y f ∈ P) {f : V} (hf : f ∈ P) (f0 : f ≠ 0) (hK : ∀ κ ∈ K, ρ κ f = f) : ∃ g ∈ P, g ≠ 0 ∧ ∀ u ∈ L, ρ u g = g

theorem **box_eq** (a : kˣ) : (box Y Y' a : Set (Matrix Y Y' k)) = Set.pi Set.univ fun _ : Y => Set.pi Set.univ fun _ : Y' => cball a

theorem **isCompact_box** (a : kˣ) : IsCompact (box Y Y' a : Set (Matrix Y Y' k))

theorem **isOpen_box** (a : kˣ) : IsOpen (box Y Y' a : Set (Matrix Y Y' k))

theorem **finite_box** (a b : kˣ) : Finite (box Y Y' a ⧸ (box Y Y' b).addSubgroupOf (box Y Y' a))

theorem **exists_box_subset** {U : Set (Matrix Y Y' k)} (hU : U ∈ 𝒩 0) : ∃ a : kˣ, ∀ x ∈ box Y Y' a, x ∈ U

theorem **pairing_comm** (ψ : AddChar k Circle) (t b : Matrix Y Y' k) : pairing ψ t b = pairing ψ b t

theorem **isBichar_pairing** (ψ : AddChar k Circle) : IsBichar (pairing (Y

theorem **pairing_single** (ψ : AddChar k Circle) (t : Matrix Y Y' k) (r : Y) (q : Y') (c : k) : pairing ψ t (Matrix.single r q c) = ψ (c * t r q)

theorem **pairing_eq_one** {ψ : AddChar k Circle} {d : kˣ} (hd : HasConductor ψ d) {t b : Matrix Y Y' k} (h : ∀ r q, valuation k (b r q * t r q) ≤ valuation k (d : k)) : pairing ψ t b = 1

theorem **exists_sep** {ψ : AddChar k Circle} {d : kˣ} (hd : HasConductor ψ d) (a : kˣ) {u : Matrix Y Y' k} (hu : u ∉ box Y Y' a) : ∃ y ∈ box Y Y' (d ⧸ a), pairing ψ u y ≠ 1

theorem **pairing_box** {ψ : AddChar k Circle} {d : kˣ} (hd : HasConductor ψ d) {a : kˣ} {u y : Matrix Y Y' k} (hu : u ∈ box Y Y' a) (hy : y ∈ box Y Y' (d ⧸ a)) : pairing ψ u y = 1

theorem **exists_vacuum** {V : Type} [AddCommGroup V] [Module ℂ V] {ψ : AddChar k Circle} (hψ : Continuous ψ) (h1 : ψ ≠ 1) (ρ σ : Matrix Y Y' k → V →ₗ[ℂ] V) (hρ : ∀ x x' f, ρ x (ρ x' f) = ρ (x + x') f) (hσ : ∀ x x' f, σ x (σ x' f) = σ (x + x') f) (ρ0 : ∀ f, ρ 0 f = f) (σ0 : ∀ f, σ 0 f = f) (hc : ∀ x y f, ρ x (σ y f) = pairing ψ x y • σ y (ρ x f)) (P : Submodule ℂ V) (hP : ∀ x, ∀ f ∈ P, ρ x f ∈ P) (hP' : ∀ y, ∀ f ∈ P, σ y f ∈ P) (sm : ∀ f ∈ P, ∀ᶠ x in 𝒩 0, ρ x f = f) (sm' : ∀ f ∈ P, ∀ᶠ x in 𝒩 0, σ x f = f) {f : V} (hf : f ∈ P) (f0 : f ≠ 0) : ∃ g ∈ P, g ≠ 0 ∧ (∀ x ∈ box Y Y' (1 : kˣ), ρ x g = g) ∧ ∀ y, (∀ x ∈ box Y Y' (1 : kˣ), pairing ψ x y = 1) → σ y g = g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCartanInvStandardInvolutionDoubledGram

theorem **tauE_star_eq** (τ : D.what₂.Tau) (x : D.host.E) : tauE D τ (star x) = star (tauE D τ x)

theorem **submatrix_fromBlocks_mul** {n : ℕ} (a b c d a' b' c' d' : Matrix (Fin n) (Fin n) ℂ) : blk4 a b c d * blk4 a' b' c' d' = blk4 (a * a' + b * c') (a * b' + b * d') (c * a' + d * c') (c * b' + d * d')

theorem **stdJ_mul_self** (n : ℕ) : (blk4 0 1 (-1) 0 : Matrix (Fin (n + n)) (Fin (n + n)) ℂ) * blk4 0 1 (-1) 0 = -1

theorem **stdJ_conjTranspose** (n : ℕ) : (blk4 0 1 (-1) 0 : Matrix (Fin (n + n)) (Fin (n + n)) ℂ)ᴴ = -blk4 0 1 (-1) 0

theorem **map_Jst** (τ : D.what₂.Tau) (n : ℕ) : (Jst D n).map (tauE D τ) = blk4 0 1 (-1) 0

theorem **isCartanInv_stdInv** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ s : RealMat (dblSp A), IsStdInv A μ s → ∀ τ, IsCartanInv (gramAt D (dblSp A) τ) (s τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCartanInvStandardInvolutionDoubledGramStatement

theorem **_root_.GlobalTheory.T.isCartanInv_standardInvolution_doubledGram** : ∀ (D : D7) (A : HSp D) (μ : D.host.E), IsSkew μ → ∀ s : RealMat (dblSp A), IsStdInv A μ s → ∀ τ, IsCartanInv (gramAt D (dblSp A) τ) (s τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCartanInvolutionOrthDecBlockSumRealMat

theorem **isCartanInv_blockSum** {k k₁ k₂ : ℕ} (G : Matrix (Fin k) (Fin k) ℂ) (GW : Matrix (Fin k₁) (Fin k₁) ℂ) (GH : Matrix (Fin k₂) (Fin k₂) ℂ) (iW : Matrix (Fin k) (Fin k₁) ℂ) (iH : Matrix (Fin k) (Fin k₂) ℂ) (pW : Matrix (Fin k₁) (Fin k) ℂ) (pH : Matrix (Fin k₂) (Fin k) ℂ) (sW : Matrix (Fin k₁) (Fin k₁) ℂ) (sH : Matrix (Fin k₂) (Fin k₂) ℂ) (hGh : Gᴴ = G) (hWW : iWᴴ * G * iW = GW) (hHH : iHᴴ * G * iH = GH) (hHW : iHᴴ * G * iW = 0) (h1 : pW * iW = 1) (h2 : pW * iH = 0) (h3 : pH * iH = 1) (h4 : pH * iW = 0) (hsum : iW * pW + iH * pH = 1) (hW : IsCartanInv GW sW) (hH : IsCartanInv GH sH) : IsCartanInv G (iW * sW * pW + iH * sH * pH)

theorem **isCartanInv_sumMat** : ∀ (D : D7) (B W H : HSp D) (o : OrthDec B W H) (sW : RealMat W) (sH : RealMat H), (∀ τ, IsCartanInv (gramAt D W τ) (sW τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → ∀ τ, IsCartanInv (gramAt D B τ) (sumMat o sW sH τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCartanInvolutionOrthDecBlockSumRealMatStatement

theorem **_root_.GlobalTheory.T.isCartanInvolution_orthDecBlockSumRealMat** : ∀ (D : D7) (B W H : HSp D) (o : OrthDec B W H) (sW : RealMat W) (sH : RealMat H), (∀ τ, IsCartanInv (gramAt D W τ) (sW τ)) → (∀ τ, IsCartanInv (gramAt D H τ) (sH τ)) → ∀ τ, IsCartanInv (gramAt D B τ) (sumMat o sW sH τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsClosedOfLocallyCompactSpace

theorem **left_and_zero** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [LocallyCompactSpace G] [SecondCountableTopology G] [T2Space G] [MeasurableSpace G] [BorelSpace G] {R : Subgroup G} (hR : IsClosed (R : Set G)) (μR : Measure R) [μR.IsHaarMeasure] [μR.IsInvInvariant] (μG : Measure G) [μG.IsHaarMeasure] {f : G → ℂ} {z : ℂ} (h : IsQuotIntL1 μR μG f z) : (∀ r ∈ R, ∀ x, f (r * x) = f x) ∧ (f = 0 → z = 0)

theorem **isClosed_of_locallyCompactSpace** {G : Type} [Group G] [TopologicalSpace G] [IsTopologicalGroup G] [T2Space G] (R : Subgroup G) [LocallyCompactSpace R] : IsClosed (R : Set G)

theorem **jac_haar_inv** {D : D7} {ρ : ℕ} (J : JacLev D ρ) : IsClosed (jacGrp J.d : Set (GA D J.W)) ∧ J.μR.IsHaarMeasure ∧ J.μR.IsInvInvariant

theorem **bes_haar_inv** {D : D7} {ρ : ℕ} (B : BesLev D ρ) : IsClosed (besGrp B.d B.o : Set (GA D B.V)) ∧ B.μR.IsHaarMeasure ∧ B.μR.IsInvInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCombAtReN

@[simp] theorem **reN_rfl** {N : ℕ} (F : Frame k N Y) : reN rfl F = F

theorem **exists_o** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) (q : GlobalTheory.FinPl.SplitPt D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (hS : IsSelfDualC S M) (hT : IsSelfDualC T M') : ∃ o : WO S T →₀ ℂ, (∀ x ∈ o.support, IsHalf (q.e + cA D q.e) x) ∧ ∀ Tm, (ofSplit q).IsModel ψ FS FT Tm → ∀ Φ₀ Φ₁ : SB (Matrix Y Y' k), IsOne Φ₁ → Φ₀.1 = Set.indicator {x | (ofSplit q).up FS FT x ∈ latO S T (M : Set _) (M' : Set _) (· = q.v)} (fun _ => 1) → Tm Φ₁ = comb ψ S T o (Tm Φ₀)

theorem **s_eq** (G : GLoc v) {j : ℕ} (h : j + 2 * 0 + 2 = D.R.m) : G.s = j

theorem **isCombAt_reN** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (q : GlobalTheory.FinPl.SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (FT : Frame k D.sp.X'.rank Y) (Ξ : Matrix Y Y' k → ℂ) : ∀ {s j : ℕ} (hs : s = j) (F : Frame k (Yc D (s + 1)).rank Y') (e : WO (Yc D (s + 1)) D.sp.X' →₀ ℂ) (e' : WO (Yc D (j + 1)) D.sp.X' →₀ ℂ), HEq e' e → (ofSplit q).IsCombAt ψ F FT e Ξ → (ofSplit q).IsCombAt ψ (reN (congrArg (fun s => (Yc D (s + 1)).rank) hs) F) FT e' Ξ

theorem **intIn_reN** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {Y' : Type} [Fintype Y'] [DecidableEq Y'] (q : GlobalTheory.FinPl.SplitPt D k) (N : ∀ j, GlobalTheory.FinPl.Lat (Yc D j)) : ∀ {s j : ℕ} (hs : s = j) (F : Frame k (Yc D (s + 1)).rank Y'), IntIn q (Yc D (s + 1)) F (N (s + 1)) → IntIn q (Yc D (j + 1)) (reN (congrArg (fun s => (Yc D (s + 1)).rank) hs) F) (N (j + 1))

theorem **eT_spec** {n : ℕ} {G : Fin n → GLoc v} (hGP : GPin G) {j : ℕ} (h : j + 2 * 0 + 2 = D.R.m) : (∀ i, HEq (eT hGP h i) (G i).egT) ∧ ∀ F, IsThetaFunO f.c.ψ _ _ F → comb f.c.ψ _ _ (v.eGT j h) F = (List.finRange n).foldr (fun i g => comb f.c.ψ _ _ (eT hGP h i) g) F

theorem **eT_is** {n : ℕ} {G : Fin n → GLoc v} (hGP : GPin G) {j : ℕ} (h : j + 2 * 0 + 2 = D.R.m) (i : Fin n) : (ofSplit (G i).q).IsCombAt f.c.ψ (frS (G i) h) (G i).FT (eT hGP h i) (phiT (G i)).1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCommensuratedKO

theorem **exists_cell** {N : ℕ} {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (g : GL (Fin N) k) : ∃ e : {e : Fin N → ℤ // Antitone e}, g ∈ cell k (Fin N) ϖ e

theorem **cellLen_eq** {N : ℕ} {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (e : Fin N → ℤ) (he : Antitone e) (g : GL (Fin N) k) (hg : g ∈ cell k (Fin N) ϖ e) : cellLen hϖ g = (∑ i, |e i|).toNat

theorem **sphSection_eq_pow** {N : ℕ} {ϖ : kˣ} (hϖ : valuation k (ϖ : k) = uniformizer k) (s : ℂ) (g : GL (Fin N) k) : sphSection k (Fin N) s (io k (Fin N) (g, 1)) = ((resCard k : ℂ) ^ (-((N : ℂ) / 2 + s))) ^ cellLen hϖ g

theorem **isCommensurated_KO** {N : ℕ} : Formalism.UG.IsCommensurated (KO k (Fin N))

theorem **factor_one** {q n : ℕ} (hq : 0 < q) (i : Fin n) (a s : ℂ) : (q : ℂ) ^ (n - 1 - (i : ℕ)) * (a * (((q : ℝ) ^ (((i : ℕ) : ℝ) - ((n : ℝ) - 1) / 2) : ℝ) : ℂ)) * (q : ℂ) ^ (-((n : ℂ) / 2 + s)) = a * (q : ℂ) ^ (-(s + 1 / 2))

theorem **qpow_mul_inv_mul_qpow_eq** {q n : ℕ} (hq : 0 < q) (i : Fin n) (a s : ℂ) : (q : ℂ) ^ (i : ℕ) * (a * (((q : ℝ) ^ (((i : ℕ) : ℝ) - ((n : ℝ) - 1) / 2) : ℝ) : ℂ))⁻¹ * (q : ℂ) ^ (-((n : ℂ) / 2 + s)) = a⁻¹ * (q : ℂ) ^ (-(s + 1 / 2))

theorem **prod_one_sub_eq_invNormFactor** {q n : ℕ} (hq : 0 < q) (s : ℂ) : ∏ i : Fin n, (1 - (q : ℂ) ^ (i : ℕ) * ((q : ℂ) ^ (-((n : ℂ) / 2 + s))) ^ 2) = invNormFactor q n s

theorem **unramifiedDoublingGL** : GlobalTheory.Published.UnramifiedDoublingGL

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactEntriesLeOne

theorem **continuous_lin_mvZ** {T : Type} [TopologicalSpace T] (g : T → ι → Matrix (Fin a) (Fin a) ℂ) (hg : Continuous g) (h : ι → Matrix (Fin b) (Fin b) ℂ) (P : FP ι a b) : ∀ Λ : FP ι a b →ₗ[ℂ] ℂ, Continuous fun x => Λ (mvZ (g x) h P)

theorem **continuous_evalAt_mul** (y : ι → Matrix (Fin a) (Fin a) ℂ) (F : PY ι a) : Continuous fun u : ι → Matrix (Fin a) (Fin a) ℂ => evalAt (fun τ => u τ * y τ) F

theorem **isCompact_entries_le_one** : IsCompact {u : ι → Matrix (Fin a) (Fin a) ℂ | ∀ τ i j, ‖u τ i j‖ ≤ 1}

theorem **subY_eq_sum_averages** : ∀ (ι : Type) [Fintype ι] (a b : ℕ) (hab : a ≤ b) (lam : ι → DomWeight a) (n : ι → ℕ) (K : Type) [Group K] [TopologicalSpace K] [IsTopologicalGroup K] [MeasurableSpace K] [BorelSpace K] (ν : Measure K) [IsProbabilityMeasure ν] (r : K → ι → Matrix (Fin a) (Fin a) ℂ), OntoUni r → ∀ y' : ι → Matrix (Fin b) (Fin b) ℂ, ∃ (m : ℕ) (y : Fin m → ι → Matrix (Fin a) (Fin a) ℂ) (P : Fin m → FP ι a b), (∀ j, IsInv (y j)) ∧ ∀ F ∈ Vn lam n, ∀ Λ : FP ι a b →ₗ[ℂ] ℂ, ∑ j, ∫ k, Λ (mvZ (fun τ => (r k τ)⁻¹) (fun _ => 1) (P j)) * evalAt (fun τ => r k τ * y j τ) F ∂ν = Λ (subY hab y' F)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactFiniteAdelicSubgroupIntegralOpen

theorem **isCompact_of_integral_open** (D : D7) (V : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)) (hΛ : IsSelfDualC V Λ) (K : Subgroup (GA D V)) (hKf : K ≤ GfC V) (hKi : ∀ k ∈ K, ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v k ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v) (hU : ∃ U : Subgroup (GA D V), IsOpen (U : Set (GA D V)) ∧ ∀ k ∈ U, k ∈ GfC V → k ∈ K) : IsCompact (K : Set (GA D V))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactFiniteAdelicSubgroupIntegralOpenStatement

theorem **_root_.GlobalTheory.T.isCompact_finite_adelic_subgroup_integral_open** : ∀ (D : D7) (V : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin V.rank → D.host.E)), IsSelfDualC V Λ → ∀ K : Subgroup (GA D V), K ≤ GfC V → (∀ k ∈ K, ∀ v, GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v k ∈ GlobalTheory.AdelicU.stabLoc D.sp.cm.sig V.gram Λ v) → (∃ U : Subgroup (GA D V), IsOpen (U : Set (GA D V)) ∧ ∀ k ∈ U, k ∈ GfC V → k ∈ K) → IsCompact (K : Set (GA D V))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactGinfCOfOntoUni

theorem **isCompact_ginfC_of_ontoUni** (D : D7) (θ : InfFam D) (V : HSp D) (B : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ) (hθ : IsInfFam θ) (hr : OntoUni (fun k : GinfC V => inB θ V B k.1)) : IsCompact (GinfC V : Set (GA D V))

theorem **integral_eq_classSum** : ∀ (D : D7) (θ : InfFam D) (V : HSp D) (B : D.what₂.Tau → Matrix (Fin V.rank) (Fin V.rank) ℂ), IsInfFam θ → OntoUni (fun k : GinfC V => inB θ V B k.1) → BorelSpace (GinfC V) ∧ CompactSpace (GinfC V) ∧ (∀ g ∈ GfC V, ∀ k ∈ GinfC V, g * k = k * g) ∧ ∃ ν : Measure (GinfC V), IsProbabilityMeasure ν ∧ ν.IsMulLeftInvariant ∧ ∀ (v : Vol V) (U : Subgroup (GA D V)) (cl : Type) [Fintype cl] (h : cl → GA D V), IsReps V U h → 0 < volLev v U ∧ (∀ a, 0 < stabOrd U (h a)) ∧ ∀ Φ : GA D V → ℂ, Continuous Φ → (∀ γ ∈ GQ V, ∀ g, Φ (γ * g) = Φ g) → (∀ κ ∈ U, ∀ g, Φ (g * κ) = Φ g) → ∫ g in v.𝒻, Φ g ∂v.μ = (volLev v U : ℂ) * ∑ a, ((stabOrd U (h a) : ℕ) : ℂ)⁻¹ * ∫ k, Φ (h a * k.1) ∂ν

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactOpenProductLevelsFamily

theorem **isCO_iSup_map** (V : HSp D) {r : ℕ} (k : Fin r → Type) [∀ i, Field (k i)] [∀ i, ValuativeRel (k i)] [∀ i, TopologicalSpace (k i)] [∀ i, IsNonarchimedeanLocalField (k i)] (q : ∀ i, GlobalTheory.FinPl.SplitPt D (k i)) (hq : Function.Injective fun i => (q i).v) (Y : Fin r → Type) [∀ i, Fintype (Y i)] [∀ i, DecidableEq (Y i)] (F : ∀ i, GlobalTheory.LocMod.Frame (k i) V.rank (Y i)) (gl : ∀ i, GL (Y i) (k i) →* GA D V) (hgl : ∀ i g, SuppAt (q i).e V (gl i g) ∧ (mt V (gl i g)).map (q i).pr = (F i).β * g.val * (F i).γ) (L : ∀ i, Subgroup (GL (Y i) (k i))) (hL : ∀ i, IsCO (L i)) : IsCO ((⨆ i, (L i).map (gl i)).subgroupOf (GlobalTheory.FinPl.grpAt V (fun p => ∃ i, (q i).v = p)))

theorem **isCO_levJ_levB** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin r → QStep S), Function.Injective (fun i => (Q i).q.v) → IsCO (levJ Q) ∧ IsCO (levB Q)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactOpenProductLevelsFamilyStatement

theorem **_root_.GlobalTheory.T.isCompactOpen_product_levels_family** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin r → QStep S), Function.Injective (fun i => (Q i).q.v) → IsCO (levJ Q) ∧ IsCO (levB Q)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactRealPointsOfDefiniteHermitianSpaceData

theorem **sig_eq_of_definite** {n : ℕ} {G : Matrix (Fin n) (Fin n) ℂ} (hG : G.PosDef ∨ (-G).PosDef) {p q : ℕ} (h : Formalism.UG.HasSignature G p q) : (p, q) = (n, 0) ∨ (p, q) = (0, n)

theorem **isDefinite_of_defSp** {D : D7} {A : HSp D} (x : DefSp A) (hx : x.Is) : A.IsDefinite

theorem **isCompact_ginfC_of_defSp** : ∀ (D : D7) (A : HSp D) (x : DefSp A), x.Is → IsCompact (GinfC A : Set (GA D A))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsCompactRealPointsOfDefiniteHermitianSpaceDataStatement

theorem **_root_.GlobalTheory.T.isCompact_realPoints_of_definiteHermitianSpaceData** : ∀ (D : D7) (A : HSp D) (x : DefSp A), x.Is → IsCompact (GinfC A : Set (GA D A))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsContAutRTrans

theorem **isContAut_zero** (V : HSp D) : IsContAut V (0 : GA D V → ℂ)

theorem **isContAut_add** {V : HSp D} {f g : GA D V → ℂ} (hf : IsContAut V f) (hg : IsContAut V g) : IsContAut V (f + g)

theorem **isContAut_smul** {V : HSp D} {f : GA D V → ℂ} (hf : IsContAut V f) (c : ℂ) : IsContAut V (c • f)

theorem **isContAut_rTrans** {V : HSp D} {f : GA D V → ℂ} (h : GA D V) (hf : IsContAut V f) : IsContAut V (rTrans h f)

theorem **isContAut_multT** {V : HSp D} {S : D.what₂.Tau → SPar} {θ : InfFam D} {M : DSModel V S} {J : Set (GA D V)} {χ : GA D V → ℂ} {G : SmT M →ₗ[ℂ] (GA D V → ℂ)} (hG : G ∈ MultT θ M J χ) (x : SmT M) : IsContAut V (G x)

theorem **per_congr** {ρ : ℕ} {ψ : AddChar (AF D.host.F) ℂ} {B₁ B₂ : BesLev D ρ} (h : B₁ = B₂) (φ₁ : GA D B₁.A → ℂ) (φ₂ : GA D B₂.A → ℂ) (hφ : HEq φ₁ φ₂) (f₁ : GA D B₁.V → ℂ) (f₂ : GA D B₂.V → ℂ) (hf : HEq f₁ f₂) : B₁.per ψ φ₁ f₁ = B₂.per ψ φ₂ f₂

theorem **GlobalTheory.Core.Mi.bot_of_pinBot** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ (D : D7) (u : D.what₂.Word) (F₀ : ∀ u, Formalism.L2.Forms.Forms0 D u) (S : STrue D F₀) (d : GlobalTheory.SeeSaw.Dat D u), GlobalTheory.Core.Mi.IdE S.X → GlobalTheory.Count.GapGe S.X.B (100 * (D.R.r : ℝ) ^ 4) → PinBot S d → d.Base → d.Bot

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFinCombOfIsThetaAt

theorem **isFinComb_zero** : IsFinComb ψ S T f₀ 0

theorem **isFinComb_add** {f g : WO S T → ℂ} (hf : IsFinComb ψ S T f₀ f) (hg : IsFinComb ψ S T f₀ g) : IsFinComb ψ S T f₀ (f + g)

theorem **isFinComb_smul** (a : ℂ) {f : WO S T → ℂ} (hf : IsFinComb ψ S T f₀ f) : IsFinComb ψ S T f₀ (a • f)

theorem **isFinComb_single** (a : ℂ) (x : WO S T) (hx : x.map (infPart D) = 0) : IsFinComb ψ S T f₀ (a • heisO ψ S T x f₀)

theorem **isFinComb_of_isThetaAt** : ∀ {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (e : AE D) (f₀ Φ f : WO S T → ℂ), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → e * e = e → infPart D e = 0 → IsSBHalf e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) Φ → IsThetaAt ψ (symplO S T) (heisO ψ S T) e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) 1 f₀ Φ f → IsFinComb ψ S T f₀ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentFiniteAdelesIffFrameUnipotentRadi

theorem **mem_span_image_iff** {R M ι : Type} [CommRing R] [AddCommGroup M] [Module R M] [Fintype ι] (v : ι → M) (s : Set ι) (w : M) : w ∈ Submodule.span R (v '' s) ↔ ∃ f : ι → R, (∀ i, i ∉ s → f i = 0) ∧ ∑ i, f i • v i = w

theorem **sigR_inclE** (e : I.E) : Formalism.UG.sigR I.cm.sig (Formalism.MainNotions.S.Af I.F) (Formalism.UG.inclE (F

theorem **hA_incl_incl** (p q : Fin I.n → I.E) : hA I (incl I p) (incl I q) = Formalism.UG.inclE (F

theorem **mulVec_flag** (v : Fin ρ → I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) : (Matrix.of fun k a => c a * x a k).map (Formalism.UG.inclE (F

theorem **mulVec_dual** (y : VA I) (a : Fin ρ) : ((Formalism.UG.cT (starRingEnd I.E) (Matrix.of fun k a => c a * x a k) * I.J).map (Formalism.UG.inclE (F

theorem **unip_flag_iff_Nfin** : ∀ (I : Formalism.MainNotions.S.AutInst) {ρ : ℕ} (x : Fin ρ → Fin I.n → I.E) (c : Fin ρ → I.E) (z : Fin I.n → I.E), (∀ a, c a ≠ 0) → (∀ a, Formalism.UG.hform (starRingEnd I.E) I.J (x a) z = 0) → ∀ (N : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)), GlobalTheory.BesselChar.Unip ((Matrix.of fun k a => c a * x a k).map (Formalism.UG.inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentFiniteAdelesIffFrameUnipotentRadiStatement

theorem **_root_.GlobalTheory.T.isFlagUnipotent_finiteAdeles_iff_frame_unipotentRadical** : ∀ (I : Formalism.MainNotions.S.AutInst) {ρ : ℕ} (x : Fin ρ → Fin I.n → I.E) (c : Fin ρ → I.E) (z : Fin I.n → I.E), (∀ a, c a ≠ 0) → (∀ a, Formalism.UG.hform (starRingEnd I.E) I.J (x a) z = 0) → ∀ (N : Matrix (Fin I.n) (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)), GlobalTheory.BesselChar.Unip ((Matrix.of fun k a => c a * x a k).map (Formalism.UG.inclE (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentMatrixIffJointlyBijectivePair

theorem **fwd** (f : A →+* R₁) (g : A →+* R₂) (hinj : ∀ y, f y = 0 → g y = 0 → y = 0) (hsurj : ∀ a b, ∃ y, f y = a ∧ g y = b) {n ρ : ℕ} (E : Matrix (Fin n) (Fin ρ) A) (K : Matrix (Fin ρ) (Fin n) A) (M : Matrix (Fin n) (Fin n) A) (h : GlobalTheory.BesselChar.Unip E K M) : GlobalTheory.BesselChar.Unip (E.map f) (K.map f) (M.map f)

theorem **bwd** (f : A →+* R₁) (g : A →+* R₂) (hinj : ∀ y, f y = 0 → g y = 0 → y = 0) (hsurj : ∀ a b, ∃ y, f y = a ∧ g y = b) {n ρ : ℕ} (E : Matrix (Fin n) (Fin ρ) A) (K : Matrix (Fin ρ) (Fin n) A) (M : Matrix (Fin n) (Fin n) A) (h1 : GlobalTheory.BesselChar.Unip (E.map f) (K.map f) (M.map f)) (h2 : GlobalTheory.BesselChar.Unip (E.map g) (K.map g) (M.map g)) : GlobalTheory.BesselChar.Unip E K M

theorem **unip_iff_of_joint** : ∀ {A R₁ R₂ : Type} [CommRing A] [CommRing R₁] [CommRing R₂] (f : A →+* R₁) (g : A →+* R₂), (∀ y, f y = 0 → g y = 0 → y = 0) → (∀ a b, ∃ y, f y = a ∧ g y = b) → ∀ {n ρ : ℕ} (E : Matrix (Fin n) (Fin ρ) A) (K : Matrix (Fin ρ) (Fin n) A) (M :

Matrix (Fin n) (Fin n) A), GlobalTheory.BesselChar.Unip E K M ↔ GlobalTheory.BesselChar.Unip (E.map f) (K.map f) (M.map f) ∧ GlobalTheory.BesselChar.Unip (E.map g) (K.map g) (M.map g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentMatrixIffJointlyBijectivePairStatement

theorem **_root_.GlobalTheory.T.isFlagUnipotentMatrix_iff_jointlyBijective_pair** : ∀ {A R₁ R₂ : Type} [CommRing A] [CommRing R₁] [CommRing R₂] (f : A →+* R₁) (g : A →+* R₂), (∀ y, f y = 0 → g y = 0 → y = 0) → (∀ a b, ∃ y, f y = a ∧ g y = b) → ∀ {n ρ : ℕ} (E : Matrix (Fin n) (Fin ρ) A) (K : Matrix (Fin ρ) (Fin n) A) (M : Matrix (Fin n) (Fin n) A), GlobalTheory.BesselChar.Unip E K M ↔ GlobalTheory.BesselChar.Unip (E.map f) (K.map f) (M.map f) ∧ GlobalTheory.BesselChar.Unip (E.map g) (K.map g) (M.map g)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentMatrixPiIffForallEval

theorem **unip_pi_iff** {ι S : Type} [Fintype ι] [DecidableEq ι] [CommRing S] {n ρ : ℕ} (E : Matrix (Fin n) (Fin ρ) (ι → S)) (K : Matrix (Fin ρ) (Fin n) (ι → S)) (M : Matrix (Fin n) (Fin n) (ι → S)) : GlobalTheory.BesselChar.Unip E K M ↔ ∀ i, GlobalTheory.BesselChar.Unip (E.map (Pi.evalRingHom (fun _ => S) i)) (K.map (Pi.evalRingHom (fun _ => S) i)) (M.map (Pi.evalRingHom (fun _ => S) i))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentMatrixPiIffForallEvalStatement

theorem **_root_.GlobalTheory.T.isFlagUnipotentMatrix_pi_iff_forall_eval** : ∀ {ι S : Type} [Fintype ι] [DecidableEq ι] [CommRing S] {n ρ : ℕ} (E : Matrix (Fin n) (Fin ρ) (ι → S)) (K : Matrix (Fin ρ) (Fin n) (ι → S)) (M : Matrix (Fin n) (Fin n) (ι → S)), GlobalTheory.BesselChar.Unip E K M ↔ ∀ i, GlobalTheory.BesselChar.Unip (E.map (Pi.evalRingHom (fun _ => S) i)) (K.map (Pi.evalRingHom (fun _ => S) i)) (M.map (Pi.evalRingHom (fun _ => S) i))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentRealPlaceIffFrameUnipotentRadical

theorem **mem_span_image_iff** (g : Fin ρ → Fin n → ℂ) (S : Set (Fin ρ)) (w : Fin n → ℂ) : w ∈ Submodule.span ℂ (g '' S) ↔ ∃ t : Fin ρ → ℂ, (∀ b, b ∉ S → t b = 0) ∧ ∑ b, t b • g b = w

theorem **mem_span_image_mulVec** (Q P : Matrix (Fin n) (Fin n) ℂ) (hPQ : P * Q = 1) (g : Fin ρ → Fin n → ℂ) (S : Set (Fin ρ)) (w : Fin n → ℂ) : Q *ᵥ w ∈ Submodule.span ℂ ((fun b => Q *ᵥ g b) '' S) ↔ w ∈ Submodule.span ℂ (g '' S)

theorem **cols_iff** (f : Fin ρ → Fin n → ℂ) (d : Fin ρ → ℂ) (hd : ∀ a, d a ≠ 0) (N : Matrix (Fin n) (Fin n) ℂ) : (∃ T : Matrix (Fin ρ) (Fin ρ) ℂ, GlobalTheory.BesselChar.StrictUp T ∧ N * Matrix.of (fun k a => d a * f a k) = Matrix.of (fun k a => d a * f a k) + Matrix.of (fun k a => d a * f a k) * T) ↔ ∀ i : Fin ρ, (N - 1) *ᵥ f i ∈ Submodule.span ℂ (f '' {b | b.1 < i.1})

theorem **colspan_iff** (f : Fin ρ → Fin n → ℂ) (d : Fin ρ → ℂ) (hd : ∀ a, d a ≠ 0) (N : Matrix (Fin n) (Fin n) ℂ) (y : Fin n → ℂ) : (∃ a, N *ᵥ y = y + Matrix.of (fun k a => d a * f a k) *ᵥ a) ↔ (N - 1) *ᵥ y ∈ Submodule.span ℂ (f '' {b | b.1 < ρ})

theorem **conj_sub_mulVec** (Q P N : Matrix (Fin n) (Fin n) ℂ) (hPQ : P * Q = 1) (w : Fin n → ℂ) : (Q * N * P - 1) *ᵥ (Q *ᵥ w) = Q *ᵥ ((N - 1) *ᵥ w)

theorem **hJ_frame** (ε : Fin n → Bool) (P Q Jc : Matrix (Fin n) (Fin n) ℂ) (hPQ : P * Q = 1) (hspec : Pᴴ * Jc * P = Formalism.REAL.Jm ε) (f y : Fin n → ℂ) : hJ ε (Q *ᵥ f) (Q *ᵥ y) = star f ⬝ᵥ (Jc *ᵥ y)

theorem **emb_hform** (I : Formalism.MainNotions.S.AutInst) (v : InfinitePlace I.F) (x z : Fin I.n → I.E) : I.φ v (Formalism.UG.hform (starRingEnd I.E) I.J x z) = star (fun i => I.φ v (x i)) ⬝ᵥ (I.J.map (I.φ v) *ᵥ fun i => I.φ v (z i))

theorem **perp_mulVec** (I : Formalism.MainNotions.S.AutInst) (v : InfinitePlace I.F) (x : Fin ρ → Fin I.n → I.E) (c : Fin ρ → I.E) (y : Fin I.n → ℂ) : (Formalism.UG.cT (starRingEnd I.E) (Matrix.of fun k a => c a * x a k) * I.J).map (I.φ v) *ᵥ y = fun a => starRingEnd ℂ (I.φ v (c a)) * (star (fun i => I.φ v (x a i)) ⬝ᵥ (I.J.map (I.φ v) *ᵥ y))

theorem **unip_iff_inN** : ∀ (I : Formalism.MainNotions.S.AutInst) (v : InfinitePlace I.F) {ρ : ℕ} (x : Fin ρ → Fin I.n → I.E) (c : Fin ρ → I.E) (z : Fin I.n → I.E), (∀ a, c a ≠ 0) → (∀ a, Formalism.UG.hform (starRingEnd I.E) I.J (x a) z = 0) → ∀ (N : Matrix (Fin I.n) (Fin I.n) ℂ), GlobalTheory.BesselChar.Unip ((Matrix.of fun k a => c a * x a k).map (I.φ v)) ((Formalism.UG.cT (starRingEnd I.E) (Matrix.of fun k a => c a * x a k) * I.J).map (I.φ v)) N ↔ Formalism.L2.SG.HidaSetting.InN (I.ε v) (fun a => Formalism.L2.SG.HidaSetting.rv I v (x a)) (Formalism.L2.SG.HidaSetting.rv I v z) ((I.P v)⁻¹ * N * I.P v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFlagUnipotentRealPlaceIffFrameUnipotentRadicalStatement

theorem **_root_.GlobalTheory.T.isFlagUnipotent_realPlace_iff_frame_unipotentRadical** : ∀ (I : Formalism.MainNotions.S.AutInst) (v : InfinitePlace I.F) {ρ : ℕ} (x : Fin ρ → Fin I.n → I.E) (c : Fin ρ → I.E) (z : Fin I.n → I.E), (∀ a, c a ≠ 0) → (∀ a, Formalism.UG.hform (starRingEnd I.E) I.J (x a) z = 0) → ∀ (N : Matrix (Fin I.n) (Fin I.n) ℂ), GlobalTheory.BesselChar.Unip ((Matrix.of fun k a => c a * x a k).map (I.φ v)) ((Formalism.UG.cT (starRingEnd I.E) (Matrix.of fun k a => c a * x a k) * I.J).map (I.φ v)) N ↔ Formalism.L2.SG.HidaSetting.InN (I.ε v) (fun a => Formalism.L2.SG.HidaSetting.rv I v (x a)) (Formalism.L2.SG.HidaSetting.rv I v z) ((I.P v)⁻¹ * N * I.P v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFockVectorOfSchrodingerMemFockSpace

theorem **killed_of_subfamily** {P Q V : Type} [AddCommGroup V] [Module ℂ V] (b : Q → V → V) (ι : P → Q) (T : V → Prop) (hb : ∀ q f, T f → T (b q f)) (hT : ∀ (c : ℂ) f, T f → T (c • f)) (hadd : ∀ q f g, T f → T g → b q (f + g) = b q f + b q g) (hsmul : ∀ q (c : ℂ) f, b q (c • f) = c • b q f) (hdec : ∀ q, ∃ (p p' : P) (c : ℂ), ∀ f, T f → b q f = b (ι p) f + c • b (ι p') f) (N : ℕ) : ∀ f, T f → (∀ l : List P, l.length = N → (l.map ι).foldr b f = 0) → ∀ l : List Q, l.length = N → l.foldr b f = 0

theorem **dRho_smul_fun** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (τ : D.what₂.Tau) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) (c : ℂ) (f : WO S T → ℂ) : Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z (c • f) = c • Formalism.IndefR.Datum.dRho (heisO ψ S T) θ τ z f

theorem **isFockVec_of_sch_mem_fockA** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ μ : D.host.E, IsSkew μ → ∀ (s' : RealMat (dblSp A)) (sB : RealMat B), IsStdInv A μ s' → (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) → ∀ l : D.what₂.Tau → ℝ, (∀ τ, |l τ| = ‖tauE D τ D.host.xi0‖ / ‖tauE D τ μ‖) → ∀ f, IsThetaFunO ψ (dblSp A) B f → sch ψ A B ν 𝒴 f ∈ fockA θ B A.rank l sB → IsFockVec ψ θ (dblSp A) B s' sB f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFockVectorOfSchrodingerMemFockSpaceStatement

theorem **_root_.GlobalTheory.T.isFockVector_of_schrodinger_mem_fockSpace** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ θ : InfFam D, IsInfFam θ → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒴 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒴 → ∀ μ : D.host.E, IsSkew μ → ∀ (s' : RealMat (dblSp A)) (sB : RealMat B), IsStdInv A μ s' → (∀ τ, IsCartanInv (gramAt D B τ) (sB τ)) → ∀ l : D.what₂.Tau → ℝ, (∀ τ, |l τ| = ‖tauE D τ D.host.xi0‖ / ‖tauE D τ μ‖) → ∀ f, IsThetaFunO ψ (dblSp A) B f → sch ψ A B ν 𝒴 f ∈ fockA θ B A.rank l sB → IsFockVec ψ θ (dblSp A) B s' sB f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFrameFinitePartMulOfCarriesLattice

theorem **finPart_finIn'** (z : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F ⊗[D.host.F] D.host.E) : finPart D (GlobalTheory.FinPlLoc.finIn D z) = z

theorem **finIn_mem_idAt** (z : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Af D.host.F ⊗[D.host.F] D.host.E) : GlobalTheory.FinPlLoc.finIn D z ∈ idAt D (fun _ => True)

theorem **mulVec_mem_ideal** {I : Ideal (AE D)} {m n : ℕ} (A : Matrix (Fin m) (Fin n) (AE D)) {y : Fin n → AE D} (hy : ∀ i, y i ∈ I) (k : Fin m) : (A *ᵥ y) k ∈ I

theorem **finE_mulVec_hat** (S : HSp D) (x : GA D S) (y : Fin S.rank → AE D) : (fun i => Formalism.IndefR.Lev.swapAE D (finPart D ((mt S x *ᵥ y) i))) = (Formalism.IndefR.Lev.finE S x).val *ᵥ (fun i => Formalism.IndefR.Lev.swapAE D (finPart D (y i)))

theorem **hat_iff_of_moves** (V : HSp D) (t : GA D V) (Λ M : Lat V) (hm : Moves V t Λ M (fun _ => True)) (z : Fin V.rank → Formalism.IndefR.Lev.EAf D) : z ∈ Formalism.L2.Herm.hatL (F

theorem **isFrame_of_moves** : ∀ (D : D7) (V : HSp D) (t : GA D V) (Λ M : Lat V) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)), Moves V t Λ M (fun _ => True) → Formalism.L2.DIV.IsFrame β Λ → Formalism.L2.DIV.IsFrame (finE V t * β) M

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFrameFinitePartMulOfCarriesLatticeStatement

theorem **_root_.GlobalTheory.T.isFrame_finitePart_mul_of_carries_lattice** : ∀ (D : D7) (V : HSp D) (t : GA D V) (Λ M : Lat V) (β : GL (Fin V.rank) (Formalism.IndefR.Lev.EAf D)), Moves V t Λ M (fun _ => True) → Formalism.L2.DIV.IsFrame β Λ → Formalism.L2.DIV.IsFrame (finE V t * β) M

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFundamentalDomainFlagUnipotentCoords

theorem **null_last** {N : Set U} (hN : ν N = 0) : (mA.prod (mB.prod ν)) {p | p.2.2 ∈ N} = 0

theorem **null_mid** {β : B → B} (hβ : Measure.QuasiMeasurePreserving β mB mB) {N : Set B} (hN : mB N = 0) : (mA.prod (mB.prod ν)) {p | β p.2.1 ∈ N} = 0

theorem **null_first** [mA.IsAddLeftInvariant] {α : A → A} (hα : Measure.QuasiMeasurePreserving α mA mA) {φ : B → A} (hφ : Measurable φ) {N : Set A} (hN : mA N = 0) : (mA.prod (mB.prod ν)) {p | α p.1 + φ p.2.1 ∈ N} = 0

theorem **isFundamentalDomain_three** [mA.IsAddLeftInvariant] (ΓA : AddSubgroup A) (ΓB : AddSubgroup B) (ΓU : Subgroup U) [Countable ΓB] [Countable ΓU] (α : U → A → A) (β : U → B → B) (hα : ∀ u, Measure.QuasiMeasurePreserving (α u) mA mA) (hβ : ∀ u, Measure.QuasiMeasurePreserving (β u) mB mB) (κ : B → B → A) (hκ : ∀ b', Measurable (κ b')) (e : A × (B × U) ≃ᵐ X) (hfree : ∀ (g g' : G) (x : X), g • x = g' • x → g = g') (hG : ∀ g : G, ∃ (c' : ΓA) (b' : ΓB) (u' : ΓU), ∀ p, g • e p = e (c'.1 + α u'.1 p.1 + κ b'.1 (β u'.1 p.2.1), b'.1 + β u'.1 p.2.1, u'.1 * p.2.2)) (hG' : ∀ (c' : ΓA) (b' : ΓB) (u' : ΓU), ∃ g : G, ∀ p, g • e p = e (c'.1 + α u'.1 p.1 + κ b'.1 (β u'.1 p.2.1), b'.1 + β u'.1 p.2.1, u'.1 * p.2.2)) {𝒞 : Set A} {ℱ : Set B} {𝒰 : Set U} (hC : IsAddFundamentalDomain ΓA 𝒞 mA) (hF : IsAddFundamentalDomain ΓB ℱ mB) (hU : IsFundamentalDomain ΓU 𝒰 ν) : IsFundamentalDomain G (e '' (𝒞 ×ˢ (ℱ ×ˢ 𝒰)))

theorem **act_coords** {D : D7} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (c' c : skewA D ρ) (b' b : Mx D ρ S₀.rank) (u' u : GL (Fin ρ) (AE D)) : d.nC c' * (d.nB b' * d.levi u') * (d.nC c * (d.nB b * d.levi u)) = d.nC (c' + skewConj u'.1 c + cComm S₀ b' (u'.1 · b)) * (d.nB (b' + u'.1 · b) * d.levi (u' * u))

theorem **fundDomain_coords** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d), (∀ p, (e p : GA D V) = d.nC p.1 * (d.nB p.2.1 * d.levi p.2.2.1)) → ∀ (lC : Measure (skewA D ρ)) (𝒞 : Set (skewA D ρ)) (𝒰 : Set (UX D ρ)), lC.IsAddHaarMeasure → SFinite lC → IsAddFundamentalDomain ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ)) 𝒞 lC → IsFundamentalDomain (UXQ D ρ) 𝒰 vU.ν → IsFundamentalDomain (NFlagQ V d.e) (e '' (𝒞 ×ˢ (vb.ℱ ×ˢ 𝒰))) ((lC.prod (vb.μ.prod vU.ν)).map e)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFundamentalDomainFlagUnipotentCoordsStatement

theorem **_root_.GlobalTheory.T.isFundamentalDomain_flagUnipotent_coords** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d), (∀ p, (e p : GA D V) = d.nC p.1 * (d.nB p.2.1 * d.levi p.2.2.1)) → ∀ (lC : Measure (skewA D ρ)) (𝒞 : Set (skewA D ρ)) (𝒰 : Set (UX D ρ)), lC.IsAddHaarMeasure → SFinite lC → IsAddFundamentalDomain ((ratMx D ρ ρ).addSubgroupOf (skewA D ρ)) 𝒞 lC → IsFundamentalDomain (UXQ D ρ) 𝒰 vU.ν → IsFundamentalDomain (NFlagQ V d.e) (e '' (𝒞 ×ˢ (vb.ℱ ×ˢ 𝒰))) ((lC.prod (vb.μ.prod vU.ν)).map e)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFundamentalDomainProdOfKernelImage

theorem **isFundamentalDomain_prod_of_ker_image** : ∀ (N G : Type) [Group N] [Group G] [MeasurableSpace N] [MeasurableSpace G] [MeasurableMul N] [MeasurableMul G] (ν : Measure N) (μ : Measure G) [SFinite ν] [SFinite μ] [ν.IsMulLeftInvariant] [μ.IsMulLeftInvariant] (Γ : Subgroup (N × G)) [Countable Γ] (𝒜 : Set N) (ℬ : Set G), IsFundamentalDomain (Γ.comap (MonoidHom.inl N G)) 𝒜 ν → IsFundamentalDomain (Γ.map (MonoidHom.snd N G)) ℬ μ → IsFundamentalDomain Γ (𝒜 ×ˢ ℬ) (ν.prod μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsFundamentalDomainProdOfKernelImageStatement

theorem **_root_.GlobalTheory.T.isFundamentalDomain_prod_of_kernel_image** : ∀ (N G : Type) [Group N] [Group G] [MeasurableSpace N] [MeasurableSpace G] [MeasurableMul N] [MeasurableMul G] (ν : Measure N) (μ : Measure G) [SFinite ν] [SFinite μ] [ν.IsMulLeftInvariant] [μ.IsMulLeftInvariant] (Γ : Subgroup (N × G)) [Countable Γ] (𝒜 : Set N) (ℬ : Set G), IsFundamentalDomain (Γ.comap (MonoidHom.inl N G)) 𝒜 ν → IsFundamentalDomain (Γ.map (MonoidHom.snd N G)) ℬ μ → IsFundamentalDomain Γ (𝒜 ×ˢ ℬ) (ν.prod μ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsHeckeCharRestrictionEtaInv

theorem **isHeckeChar_resEta_inv** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle) (m : ℕ), IsHeckeChar D χ → ResEta D χ m → IsHeckeChar D χ⁻¹ ∧ ResEta D χ⁻¹ m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsHeckeCharRestrictionEtaInvStatement

theorem **_root_.GlobalTheory.T.isHeckeChar_restrictionEta_inv** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle) (m : ℕ), IsHeckeChar D χ → ResEta D χ m → IsHeckeChar D χ⁻¹ ∧ ResEta D χ⁻¹ m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsIdentityOffFreePositionsRightBlockIff

theorem **st_of_kset** (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (β : GL (Fin (D.sp.Y 0).rank) (Lev.EAf D)) (hβ : Formalism.L2.DIV.IsFrame β (Formalism.L2.Herm.crdLat (D.sp.Y 0) (D.sp.Y 0).L0)) (κ : GA D (D.sp.Y 0)) (hκ : (iotaX X h κ).2 ∈ X.B.I.Kset) : Lev.finE (D.sp.Y 0) κ ∈ Formalism.L2.DIV.St β

theorem **exists_generic** (Pg : Finset (Pl D.host.E)) : ∃ ψ : ∀ PP : Pl D.host.E, AddChar (kP PP) ℂ, IsGenericPsi Pg ψ

theorem **isIdentityOffFreePositions_rightBlock_iff** {K : Type} [CommRing K] {r n : ℕ} (ρ k : ℕ) (hr : r ≤ n) (hn : n ≤ r) (g : GL (Fin n) K) : GlobalTheory.SymRes.IsPat ρ k (upperLeftBlock hr g) ↔ GlobalTheory.SymRes.IsPat ρ k g

theorem **reductionBlock_injective_of_eq** {K : Type} [CommRing K] {r n : ℕ} (hr : r ≤ n) (hn : n ≤ r) (g g' : GL (Fin n) K) (h : upperLeftBlock hr g = upperLeftBlock hr g') : g = g'

theorem **conj_mem_GfC** (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (s k : GA D (D.sp.Y 0)) (hk : k ∈ GfC (D.sp.Y 0)) : s * k * s⁻¹ ∈ GfC (D.sp.Y 0)

theorem **exists_read** (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (Pg : Finset (Pl D.host.E)) (hTP : IsTypePlaces Pg) (hsp : ∃ e, IsIdemSum e Pg) (hid : ∀ PP : Pg, ∃ e, Formalism.L2.NTheta.IsIdemAt (D

theorem **conj_mem_K_two** (u : D.what₂.Word) (F₀ : ∀ u, Formalism.L2.Forms.Forms0 D u) (S : STrue D F₀) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m) (hE : GlobalTheory.Core.Mi.IdE S.X) (hB : d.Base) (hTP : IsTypePlaces d.c.Pg) (hPG : GlobalTheory.LocMod.PinGBy G cG) (hLev : LevX (F₀ u) d h0) (B : SBes S.X S.inst u d.c.ψ (D.sp.X 0)) (hBl : B.lev = d.B 0 D.R.rS h0) (hAd : B.Adapted (S.tr u).wt) (a₀ : GA D (D.sp.X 0)) (hH0 : IsTransportAdelicCorrection (S.tr u) a₀) (a : GA D (D.sp.X 0)) (hac : a ∈ Subgroup.center (GA D (D.sp.X 0))) (t s : GA D (D.sp.Y 0)) (ht : t = d.t 0) (hs : s = ((iotaX S.X S.inst).symm (1, S.X.xu u))⁻¹ * embA B.d B.o (a₀ * a) * t) (hSt : ∀ κ : GA D (D.sp.Y 0), (κ ∈ GfC (D.sp.Y 0)

theorem **mem_K_two_iff** (u : D.what₂.Word) (F₀ : ∀ u, Formalism.L2.Forms.Forms0 D u) (S : STrue D F₀) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m) (hE : GlobalTheory.Core.Mi.IdE S.X) (hB : d.Base) (hTP : IsTypePlaces d.c.Pg) (hPG : GlobalTheory.LocMod.PinGBy G cG) (hLev : LevX (F₀ u) d h0) (B : SBes S.X S.inst u d.c.ψ (D.sp.X 0)) (hBl : B.lev = d.B 0 D.R.rS h0) (hAd : B.Adapted (S.tr u).wt) (a₀ : GA D (D.sp.X 0)) (hH0 : IsTransportAdelicCorrection (S.tr u) a₀) (a : GA D (D.sp.X 0)) (hac : a ∈ Subgroup.center (GA D (D.sp.X 0))) (t s : GA D (D.sp.Y 0)) (ht : t = d.t 0) (hs : s = ((iotaX S.X S.inst).symm (1, S.X.xu u))⁻¹ * embA B.d B.o (a₀ * a) * t) (hSt : ∀ κ : GA D (D.sp.Y 0), (κ ∈ GfC (D.sp.Y 0)

theorem **mem_K_two_iff_reduction_tuple** : ∀ (D : D7) (u : D.what₂.Word) (F₀ : ∀ u, Formalism.L2.Forms.Forms0 D u) (S : STrue D F₀) (d : GlobalTheory.SeeSaw.Dat D u) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : GlobalTheory.FinPl.OffDat d k ι) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d o.τ PP.1) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), GlobalTheory.Core.Mi.IdE S.X → d.Base → IsTypePlaces d.c.Pg → GlobalTheory.FinPl.PinOff o → GlobalTheory.LocMod.PinGBy G o.cG → (∀ v : GlobalTheory.FinPl.Pl D, o.atT v ↔ (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∈ v.asIdeal) → (∀ v : GlobalTheory.FinPl.Pl D, ¬ o.atT v → GlobalTheory.FinPl.latV (D.sp.X 0) (d.x 0).L (· = v) = GlobalTheory.FinPl.latV (D.sp.X 0) ((F₀ u).cls0.lev.Lb.map ((LinearEquiv.funCongrLeft D.host.E D.host.E (finCongr (Formalism.L2.Forms.rankN D 0 (Nat.zero_le _)))).toLinearMap.restrictScala

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsIntegralHondaLogarithmPairShortWeierstrass

theorem **intLog_shortW** (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (g2 g3 : K) (h2 : ‖(2 : K)‖ = 1) (hg2 : ‖g2‖ ≤ 1) (hg3 : ‖g3‖ ≤ 1) : IntLog (Formalism.L2.Th1.A.logSer (Formalism.L2.Th1.A.shortW g2 g3)) (Formalism.L2.Th1.A.expSer (Formalism.L2.Th1.A.shortW g2 g3))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsIntegralHondaLogarithmPairShortWeierstrassStatement

theorem **_root_.GlobalTheory.T.isIntegralHondaLogarithmPair_shortWeierstrass** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (g2 g3 : K), ‖(2 : K)‖ = 1 → ‖g2‖ ≤ 1 → ‖g3‖ ≤ 1 → IntLog (Formalism.L2.Th1.A.logSer (Formalism.L2.Th1.A.shortW g2 g3)) (Formalism.L2.Th1.A.expSer (Formalism.L2.Th1.A.shortW g2 g3))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsIntegralMulVec

theorem **isIntegral_mulVec** {E : Type} [Field E] {m n : ℕ} (M : Matrix (Fin m) (Fin n) E) (hM : ∀ i j, IsIntegral ℤ (M i j)) (x : Fin n → E) (hx : ∀ k, IsIntegral ℤ (x k)) (k : Fin m) : IsIntegral ℤ ((M *ᵥ x) k)

theorem **isIntegral_ite** {E : Type} [Field E] (p : Prop) [Decidable p] : IsIntegral ℤ (if p then (1 : E) else 0)

theorem **orthAt_oTop** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} (hI : I.Ok) (n : ℕ) (h : n = (I.rec7 l Z).R.m) (x : GlobalTheory.FinPl.Pl (I.rec7 l Z)), GlobalTheory.FinPl.OrthAt (GlobalTheory.XGeoStd.oTop l Z hI n h) (GlobalTheory.PlaceModel.latFrame ((I.rec7 l Z).sp.X n)) (GlobalTheory.PlaceModel.latFrame (I.rec7 l Z).sp.X') (GlobalTheory.PlaceModel.latFrame (GlobalTheory.XGeoStd.lineZ (I.rec7 l Z))) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsIteratedQuotientIntegralFinsetSum

theorem **isIteratedQuotientIntegral_finset_sum** {ι : Type} (s : Finset ι) (F : ι → G₁ → G₂ → ℂ) (z : ι → ℂ) (hF : ∀ i ∈ s, Iter2 ρ₁ μ₁ ρ₂ μ₂ (F i) (z i)) : Iter2 ρ₁ μ₁ ρ₂ μ₂ (fun x y => ∑ i ∈ s, F i x y) (∑ i ∈ s, z i)

theorem **isIteratedQuotientIntegral_const_mul** {F : G₁ → G₂ → ℂ} {z : ℂ} (a : ℂ) (hF : Iter2 ρ₁ μ₁ ρ₂ μ₂ F z) : Iter2 ρ₁ μ₁ ρ₂ μ₂ (fun x y => a * F x y) (a * z)

theorem **moves_iff_forall** {S : HSp D} {L : Lat S} (hL : IsSelfDualC S L) (P : Pl D → Prop) (g : GA D S) : Moves S g L L P ↔ ∀ v, P v → Moves S g L L (· = v)

theorem **intAt_iff_moves** {V : HSp D} {L : Lat V} {k Y : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [Fintype Y] [DecidableEq Y] (v : Pl D) (gl : GL Y k →* GA D V) (hr : gl.range = grpAt V (· = v)) (hx : ∀ x : GA D V, ∃! g, x * (gl g)⁻¹ ∈ grpOff V (· = v)) (hK : ∀ g, g ∈ KO k Y ↔ InK V L (· = v) (gl g)) (x : GA D V) : IntAt v gl x ↔ Moves V x L L (· = v)

theorem **vac_single** (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {P : Pl D → Prop} {v : Pl D} (hv : P v) {f : WO S T →₀ ℂ} (hf : Vac ψ S T LS LT P f) : Vac ψ S T LS LT (· = v) f

theorem **isFinComb_of_half** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) {S T : HSp D} {c : WO S T →₀ ℂ} (hc : ∀ u ∈ c.support, IsHalf (q.e + cA D q.e) u) : IsFinComb c

theorem **isFinComb_aconv** {S T : HSp D} (l : List (WO S T →₀ ℂ)) (hl : ∀ c ∈ l, IsFinComb c) : IsFinComb (aconv l)

theorem **aconv_single_cons** {G : Type} [AddMonoid G] (l : List (G →₀ ℂ)) : aconv (Finsupp.single 0 1 :: l) = aconv l

theorem **comb_single_zero** (S T : HSp D) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single 0 1) f = f

theorem **biComb_single_zero** (A S A' : HSp D) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' (Finsupp.single 0 1) (Finsupp.single 0 1) C = C

theorem **sum_support_smul** {G M : Type} [AddCommMonoid M] [Module ℂ M] (a : ℂ) (e : G →₀ ℂ) (F : G → M) : ∑ g ∈ (a • e).support, (a • e) g • F g = a • ∑ g ∈ e.support, e g • F g

theorem **comb_smul_left** (S T : HSp D) (a : ℂ) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (a • c) f = a • comb ψ S T c f

theorem **trc_smul_left** {V : HSp D} (a : ℂ) (e : GA D V →₀ ℂ) (f : GA D V → ℂ) : trc (a • e) f = a • trc e f

theorem **biComb_smul_left** (A S A' : HSp D) (a b : ℂ) (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' (a • c) (b • c') C = (a * b) • biComb ψ A S A' c c' C

theorem **isFinComb_extS** {S A L : HSp D} (o : OrthDec S A L) (T : HSp D) {c : WO A T →₀ ℂ} (hc : IsFinComb c) : IsFinComb (extS o T c)

theorem **isFinComb_extT** {T A L : HSp D} (o : OrthDec T A L) (S : HSp D) {c : WO S A →₀ ℂ} (hc : IsFinComb c) : IsFinComb (extT o S c)

theorem **per_sum** (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} (B : BesLev D ρ) {f : GA D B.V → ℂ} (hf : IsContAut B.V f) {I : Type} [Fintype I] (φ : I → GA D B.A → ℂ) (hφ : ∀ i, IsContAut B.A (φ i)) : B.per ψ (∑ i, φ i) f = ∑ i, B.per ψ (φ i) f

theorem **theta_flatBF** {f : WO S.B.V S.J.W → ℂ} (hf : IsThetaFunO ψ _ _ f) : IsThetaFunO ψ _ _ (flatBF S.s₁ f)

theorem **theta_flatFB** {f : WO S.J.W S.B'.V → ℂ} (hf : IsThetaFunO ψ _ _ f) : IsThetaFunO ψ _ _ (flatFB S.s₂ f)

theorem **val_eq_hand** (hS : HandIsLift S) {Φ₁ : WO S.B.V S.J.W → ℂ} {Φ₂ : WO S.J.W S.B'.V → ℂ} (h₁ : IsThetaFunO ψ _ _ Φ₁) (h₂ : IsThetaFunO ψ _ _ Φ₂) {φ : GA D S.B.A → ℂ} (hφ : IsContAut _ φ) (f : GA D S.B'.V → ℂ) (g x y) : S.val (Finsupp.single 0 1) (Finsupp.single 0 1) Φ₁ Φ₂ φ f g x y = S.B'.per ψ (Chain.hand S.s₁ S.s₂ S.l S.vL Φ₁ Φ₂ φ g x y) (rTrans y f)

theorem **val_one** (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (Φ₁ : WO S.B.V S.J.W → ℂ) (Φ₂ : WO S.J.W S.B'.V → ℂ) (φ : GA D S.B.A → ℂ) (f : GA D S.B'.V → ℂ) : S.val c₁ c₂ Φ₁ Φ₂ φ f 1 1 1 = S.B'.per ψ (biLift S.oma S.omb S.B.vA S.J.v0 φ (biComb ψ _ _ _ c₁ c₂ (GlobalTheory.Unf.contr S.l S.J.W₀ S.vL (flatBF S.s₁ Φ₁) (flatFB S.s₂ Φ₂)))) f

theorem **contAut_lift** (hψ : IsStdChar D ψ (-1)) (hS : HandIsLift S) {c₁ : WO S.B.A S.J.W₀ →₀ ℂ} {c₂ : WO S.J.W₀ S.B'.A →₀ ℂ} (hc₁ : IsFinComb c₁) (hc₂ : IsFinComb c₂) {Φ₁ : WO S.B.V S.J.W → ℂ} {Φ₂ : WO S.J.W S.B'.V → ℂ} (h₁ : IsThetaFunO ψ _ _ Φ₁) (h₂ : IsThetaFunO ψ _ _ Φ₂) {φ : GA D S.B.A → ℂ} (hφ : IsContAut _ φ) : IsContAut S.B'.A (biLift S.oma S.omb S.B.vA S.J.v0 φ (biComb ψ _ _ _ c₁ c₂ (GlobalTheory.Unf.contr S.l S.J.W₀ S.vL (flatBF S.s₁ Φ₁) (flatFB S.s₂ Φ₂))))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJacFramePairing

theorem **pair_col** (p : Place D k) {N ρ : ℕ} {κ : Type} [Fintype κ] (G : Matrix (Fin N) (Fin N) D.host.E) (X : Matrix (Fin N) κ D.host.E) (e : Matrix (Fin N) (Fin ρ) D.host.E) (j : κ) (i : Fin ρ) : ∑ r, p.pr (X r j) * p.pr (star ((G *ᵥ fun r => e r i) r)) = p.pr (star ((Xᴴ * G * e) j i))

theorem **pair_scaled** (p : Place D k) {N ρ : ℕ} {κ Z : Type} [Fintype κ] (G : Matrix (Fin N) (Fin N) D.host.E) (X : Matrix (Fin N) κ D.host.E) (e : Matrix (Fin N) (Fin ρ) D.host.E) (β : Matrix (Fin N) Z k) (m : Z) (j : κ) (t : k) (h : ∀ r, β r m = t * p.pr (X r j)) (i : Fin ρ) : ∑ r, β r m * p.pr (star ((G *ᵥ fun r => e r i) r)) = t * p.pr (star ((Xᴴ * G * e) j i))

theorem **pr_zero** (p : Place D k) : p.pr 0 = 0

theorem **pr_one** (p : Place D k) : p.pr 1 = 1

theorem **pr_star_one** (p : Place D k) {ρ : ℕ} (i' i : Fin ρ) : p.pr (star ((1 : Matrix (Fin ρ) (Fin ρ) D.host.E) i' i)) = if i' = i then 1 else 0

theorem **isJacFrame_pairing** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ρ : ℕ) (J : JacLev D ρ) (s hh : k) (n : Type) [Fintype n] [DecidableEq n] (c : ℕ) (F₀ : Frame k J.W₀.rank n) (F : Frame k J.W.rank (n ⊕ Fin c)), p.IsJacFrame J s hh F₀ F → valuation k s = 1 → valuation k hh = 1 → valuation k p.xi = 1 → ∀ i : Fin ρ, ∃ (b : k) (m₀ : n ⊕ Fin c), valuation k b = 1 ∧ (∀ r, b * p.pr (J.d.e r i) = F.β r m₀) ∧ (∀ m, ∑ r, F.β r m * p.pr (star ((J.W.gram *ᵥ fun r => J.d.e r i) r)) ∈ 𝒪[k]) ∧ ∃ m, valuation k (∑ r, F.β r m * p.pr (star ((J.W.gram *ᵥ fun r => J.d.e r i) r))) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaHeisenbergDeriv

theorem **dLine_inl** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (G : WO A S → WO S A' → ℂ) (M : ℝ →+ WO A S) : GlobalTheory.WeilVacMove.dLine ψ _ S ((inlH (sumDec A (GlobalTheory.ChiGrid.negSp A')) S).comp M) (join (sumDec A (GlobalTheory.ChiGrid.negSp A')) S G) = join (sumDec A (GlobalTheory.ChiGrid.negSp A')) S (fun u v => GlobalTheory.WeilVacMove.dLine ψ A S M (fun u' => G u' v) u)

theorem **dLine_inr** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (G : WO A S → WO S A' → ℂ) (M : ℝ →+ WO S A') : GlobalTheory.WeilVacMove.dLine ψ _ S ((inrH (sumDec A (GlobalTheory.ChiGrid.negSp A')) S).comp M) (join (sumDec A (GlobalTheory.ChiGrid.negSp A')) S G) = join (sumDec A (GlobalTheory.ChiGrid.negSp A')) S (fun u v => GlobalTheory.WeilVacMove.dLine ψ S A' M (G u) v)

theorem **isBiTheta_dLine** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A S A' : HSp D) (G : WO A S → WO S A' → ℂ), IsBiTheta ψ A S A' G → (∀ M : ℝ →+ WO A S, Continuous M → (∀ s, (M s).map (finPart D) = 0) → IsBiTheta ψ A S A' fun u v => GlobalTheory.WeilVacMove.dLine ψ A S M (fun u' => G u' v) u) ∧ ∀ M' : ℝ →+ WO S A', Continuous M' → (∀ s, (M' s).map (finPart D) = 0) → IsBiTheta ψ A S A' fun u v => GlobalTheory.WeilVacMove.dLine ψ S A' M' (G u) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaHeisenbergDerivStatement

theorem **_root_.GlobalTheory.T.isJointlySmoothBiTheta_heisenbergDeriv** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A S A' : HSp D) (G : WO A S → WO S A' → ℂ), IsBiTheta ψ A S A' G → (∀ M : ℝ →+ WO A S, Continuous M → (∀ s, (M s).map (finPart D) = 0) → IsBiTheta ψ A S A' fun u v => GlobalTheory.WeilVacMove.dLine ψ A S M (fun u' => G u' v) u) ∧ ∀ M' : ℝ →+ WO S A', Continuous M' → (∀ s, (M' s).map (finPart D) = 0) → IsBiTheta ψ A S A' fun u v => GlobalTheory.WeilVacMove.dLine ψ S A' M' (G u) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaTwoVarCombination

theorem **real_part** {S T : HSp D} (x : WO S T) : (GlobalTheory.FinTranslate.eInf D • x : WO S T).map (finPart D) = 0

theorem **fin_part** {S T : HSp D} (x : WO S T) : ((1 - GlobalTheory.FinTranslate.eInf D) • x : WO S T).map (infPart D) = 0

theorem **psi_fin_eq** {S T : HSp D} {w : WO S T} (hw : w.map (infPart D) = 0) (x : WO S T) : ψ (symplOH S T w x) = ψ (symplOH S T w ((1 - GlobalTheory.FinTranslate.eInf D) • x))

theorem **exists_orth_any** {S T : HSp D} (hψ : IsStdChar D ψ (-1)) (s : Finset (WO S T)) : ∃ U : AddSubgroup (WO S T), IsOpen (U : Set (WO S T)) ∧ ∀ w ∈ U, w.map (infPart D) = 0 → ∀ x ∈ s, ψ (symplOH S T w x) = 1

theorem **contDiff_psi** {S T : HSp D} (hψ : IsStdChar D ψ (-1)) {n : ℕ} {L : (Fin n → ℝ) →+ WO S T} (hL : Continuous L) (hLf : ∀ t, (L t).map (finPart D) = 0) (x : WO S T) : ContDiff ℝ ((⊤ : ℕ∞) : WithTop ℕ∞) fun t => ψ (symplOH S T (L t) x)

theorem **biTheta_combL** {A S A' : HSp D} (hψ : IsStdChar D ψ (-1)) {K : WO A S → WO S A' → ℂ} (hK : IsBiTheta ψ A S A' K) (c : WO A S →₀ ℂ) : IsBiTheta ψ A S A' (fun u v => comb ψ A S c (fun u' => K u' v) u)

theorem **biTheta_combR** {A S A' : HSp D} (hψ : IsStdChar D ψ (-1)) {K : WO A S → WO S A' → ℂ} (hK : IsBiTheta ψ A S A' K) (c : WO S A' →₀ ℂ) : IsBiTheta ψ A S A' (fun u v => comb ψ S A' c (K u) v)

theorem **biTheta_biComb** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A S A' : HSp D) (C : WO A S → WO S A' → ℂ), IsBiTheta ψ A S A' C → ∀ (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ), IsBiTheta ψ A S A' (biComb ψ A S A' c c' C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaTwoVarCombinationStatement

theorem **_root_.GlobalTheory.T.isJointlySmoothBiTheta_twoVarCombination** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A S A' : HSp D) (C : WO A S → WO S A' → ℂ), IsBiTheta ψ A S A' C → ∀ (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ), IsBiTheta ψ A S A' (biComb ψ A S A' c c' C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaWeilRepFirstVariable

theorem **isBiTheta_act_left** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (χA χS : (AE D)ˣ →* Circle) (oma : Om A S), IsWeilDatumO ψ A S χA χS oma → ∀ C, IsBiTheta ψ A S A' C → ∀ p, IsBiTheta ψ A S A' fun u v => oma p (fun u' => C u' v) u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaWeilRepFirstVariableStatement

theorem **_root_.GlobalTheory.T.isJointlySmoothBiTheta_weilRep_first_variable** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (χA χS : (AE D)ˣ →* Circle) (oma : Om A S), IsWeilDatumO ψ A S χA χS oma → ∀ C, IsBiTheta ψ A S A' C → ∀ p, IsBiTheta ψ A S A' fun u v => oma p (fun u' => C u' v) u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaWeilRepSecondVariable

theorem **conjSource_conjAdjoint_apply** {S A' : HSp D} (omb : Om S A') (g : GA D S) (h : GA D A') (f : WO S A' → ℂ) (y : WO A' S) : conjSource (conjAdjoint omb) (negGA A' h, g) (fun y' => f (dagger A' S y')) y = omb (g, h) f (dagger A' S y)

theorem **resS_symm_swap** {W A B : HSp D} (d : OrthDec W A B) (T : HSp D) (u : WO A T) (y : WO B T) : (d.symm.resS T).symm (y, u) = (d.resS T).symm (u, y)

theorem **isBiTheta_act_right** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (χS' χA' : (AE D)ˣ →* Circle) (omb : Om S A'), IsWeilDatumO ψ S A' χS' χA' omb → ∀ C, IsBiTheta ψ A S A' C → ∀ q, IsBiTheta ψ A S A' fun u v => omb q (C u) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsJointlySmoothBiThetaWeilRepSecondVariableStatement

theorem **_root_.GlobalTheory.T.isBiTheta_weilRep_second_variable** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (χS' χA' : (AE D)ˣ →* Circle) (omb : Om S A'), IsWeilDatumO ψ S A' χS' χA' omb → ∀ C, IsBiTheta ψ A S A' C → ∀ q, IsBiTheta ψ A S A' fun u v => omb q (C u) v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsKFinOfMem

theorem **isKFin_of_mem** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) {x : M.H τ} (hx : x ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ) : Formalism.MainNotions.S.IsKFin (M.ε τ) (M.σ τ) x

theorem **itSpans_of** (hDS : GlobalTheory.Published.DiscreteSeriesCover) (hHC : GlobalTheory.Published.HarishChandraModules) {D : D7} {V W : HSp D} {S S' : D.what₂.Tau → SPar} (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (aS aT : D.what₂.Tau → ℚ) (ω : GA D V × GA D W →* Module.End ℂ (WO V W → ℂ)) (f : WO V W → ℂ) (It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M) (hθ : IsInfFam θ) (hw : IsWeilExp ψ V W aS aT ω) (hv : IsVacVec ψ θ M M' f) (hi : IsItW θ M M' ω (ptOf ψ θ M M' f) It) : GlobalTheory.Core.Mi.ItSpans M M' It

theorem **GlobalTheory.Fam.commOf_is** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ g : GlobalTheory.Fam.Geo D, GeoMeas (pk W) g → (∀ j, j ≤ D.R.m → (g.x j).Is) → (commOf (pk W) u g).Is

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsKFinOfMemKfin

theorem **isKFin_of_mem_kfin** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) {x : M.H τ} (hx : x ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ) : IsKFin (M.ε τ) (M.σ τ) x

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsLatticeFrameAtPlaceDirectSumBlockDiagonal

theorem **latAt_sumLat_self** {E : Type} [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) : latAt w (sumLat finSumFinEquiv L L) = sumLat finSumFinEquiv (latAt w L) (latAt w L)

theorem **isFrameAt_sumLat_blkGL** : ∀ (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)) (a : GL (Fin n) (w.adicCompletion E)), IsFrameAt w L a → IsFrameAt w (sumLat finSumFinEquiv L L) (blkGL finSumFinEquiv (a, a))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsLatticeFrameAtPlaceDirectSumBlockDiagonalStatement

theorem **_root_.GlobalTheory.T.isLatticeFrameAtPlace_directSum_blockDiagonal** : ∀ (E : Type) [Field E] [NumberField E] (w : HeightOneSpectrum (𝒪 E)) (n : ℕ) (L : Submodule (𝒪 E) (Fin n → E)) (a : GL (Fin n) (w.adicCompletion E)), IsFrameAt w L a → IsFrameAt w (sumLat finSumFinEquiv L L) (blkGL finSumFinEquiv (a, a))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsLocalModelCreationWeilActionMovedFrames

theorem **half_iff** {m n : ℕ} (e : AE D) (u : Mx D m n) : IsHalf e u ↔ e • u = u

theorem **mtMul** (V : HSp D) (a b : GA D V) : Formalism.L2.Theta.mt V (a * b) = Formalism.L2.Theta.mt V a * Formalism.L2.Theta.mt V b

theorem **mtOne** (V : HSp D) : Formalism.L2.Theta.mt V 1 = 1

theorem **half_actO** {e : AE D} {S T : HSp D} (g : GA D S) (h : GA D T) {u : WO S T} (hu : IsHalf e u) : IsHalf e (actO S T g h u)

theorem **actO_actO_inv''** {S T : HSp D} (g : GA D S) (h : GA D T) (y : WO S T) : actO S T g h (actO S T g⁻¹ h⁻¹ y) = y

theorem **infPart_e** (p : Place D k) : infPart D p.e = 0

theorem **half_up** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x : Matrix Y Y' k) : IsHalf p.e (p.up FS FT x)

theorem **eq_of_half** (p : Place D k) {m n : ℕ} {z z' : Mx D m n} (hz : IsHalf p.e z) (hz' : IsHalf p.e z') (h : z.map p.π = z'.map p.π) : z = z'

theorem **up_move** (p : Place D k) {S T : HSp D} (FS GS : Frame k S.rank Y') (FT GT : Frame k T.rank Y) (g : GA D S) (h : GA D T) (hS : GS.β = (Formalism.L2.Theta.mt S g).map p.π ∗ FS.β) (hT : GT.β = (Formalism.L2.Theta.mt T h).map p.π ∗ FT.β) (x : Matrix Y Y' k) : p.up GS GT x = actO S T g h (p.up FS FT x)

theorem **isModel_move** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS GS : Frame k S.rank Y') (FT GT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (χS χT : (AE D)ˣ →∗ Circle) (ω : GA D S × GA D T →∗ Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT ω → ∀ (g : GA D S) (h : GA D T), GS.β = (Formalism.L2.Theta.mt S g).map p.π ∗ FS.β → GT.β = (Formalism.L2.Theta.mt T h).map p.π ∗ FT.β → ∀ sS sT BS BT P T₁, p.IsModel ψ FS FT T₁ → ∃ T₂, p.IsModel ψ GS GT T₂ ∧ ∀ Φ, T₂ Φ = ω (g, h) (pureTensorO ψ θ S T sS sT BS BT (T₁ Φ) P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsLocalModelCreationWeilActionMovedFramesStatement

theorem **_root_.GlobalTheory.T.isLocalModel_creation_weilAction_moved_frames** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS GS : Frame k S.rank Y') (FT GT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (χS χT : (AE D)ˣ →∗ Circle) (ω : GA D S × GA D T →∗ Module.End ℂ (WO S T → ℂ)), IsInfFam θ → IsWeilDatumO ψ S T χS χT ω → ∀ (g : GA D S) (h : GA D T), GS.β = (Formalism.L2.Theta.mt S g).map p.π ∗ FS.β → GT.β = (Formalism.L2.Theta.mt T h).map p.π ∗ FT.β → ∀ sS sT BS BT P T₁, p.IsModel ψ FS FT T₁ → ∃ T₂, p.IsModel ψ GS GT T₂ ∧ ∀ Φ, T₂ Φ = ω (g, h) (pureTensorO ψ θ S T sS sT BS BT (T₁ Φ) P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsLocalModelTensorWithHalfFunction

theorem **cA_invol** (z : AE D) : cA D (cA D z) = z

theorem **symplO_add_left'** (S T : HSp D) (x x' y : WO S T) : symplO S T (x + x') y = symplO S T x y + symplO S T x' y

theorem **half_sum** {m n : ℕ} {e e' : AE D} (he : e ∗ e' = 0) {u : Mx D m n} (hu : IsHalf e' u) : IsHalf (e + e') u

theorem **half_sum_cA** {m n : ℕ} {e e' : AE D} (he : e ∗ e' = 0) {u : Mx D m n} (hu : IsHalf (cA D e') u) : IsHalf (cA D (e + e')) u

theorem **symplO_half_zero** {S T : HSp D} {e e' : AE D} (he : e ∗ e' = 0) {s y : WO S T} (hs : IsHalf e s) (hy : IsHalf (cA D e') y) : symplO S T s y = 0

theorem **isModel_tensor** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (e : AE D), e ∗ p.e = 0 → ∀ Φ : WO S T → ℂ, (∀ t, Φ t ≠ 0 → IsHalf e t) → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) (e + p.e) (stdHom S T) 1 f₀ T' → (∀ Φ', IsSBHalf (e + p.e) (stdHom S T) Φ' → IsThetaFunO ψ S T (T' Φ')) → (∀ Ψ ∈ SB (Matrix Y Y' k), IsSBHalf (e + p.e) (stdHom S T) (p.tensorAt FS FT Φ Ψ)) → p.IsModel ψ FS FT (T' ∘ₗ p.tensorL FS FT Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsLocalModelTensorWithHalfFunctionStatement

theorem **_root_.GlobalTheory.T.isLocalModel_tensorWithHalfFunction** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (e : AE D), e ∗ p.e = 0 → ∀ Φ : WO S T → ℂ, (∀ t, Φ t ≠ 0 → IsHalf e t) → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) (e + p.e) (stdHom S T) 1 f₀ T' → (∀ Φ', IsSBHalf (e + p.e) (stdHom S T) Φ' → IsThetaFunO ψ S T (T' Φ')) → (∀ Ψ ∈ SB (Matrix Y Y' k), IsSBHalf (e + p.e) (stdHom S T) (p.tensorAt FS FT Φ Ψ)) → p.IsModel ψ FS FT (T' ∘ₗ p.tensorL FS FT Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsModelCombL

theorem **comb_apply** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) (w : WO S T) : comb ψ S T c f w = ∑ x ∈ c.support, c x ∗ (ψ (symplOH S T w x) ∗ f (w + x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsMulLeftInvariantMapLevyProductMapToFourierJaco

theorem **mul_prodRFJ_embX** (ρ : ℕ) (g' : GL n k) (q : (GL n k × N k n c (ρ + 1)) × (n → k)) : embX c g' ∗ (prodRFJ ρ q : GL (n ⊕ Fin c) k) = prodRFJ ρ ((g' ∗ q.1.1, q.1.2), q.2)

theorem **mul_prodRFJ_N** (ρ : ℕ) (u' : N k n c (ρ + 1)) (q : (GL n k × N k n c (ρ + 1)) × (n → k)) : (u' : GL (n ⊕ Fin c) k) ∗ (prodRFJ ρ q : GL (n ⊕ Fin c) k) = prodRFJ ρ ((q.1.1, conjN (ρ + 1) q.1.1 u' ∗ q.1.2), q.2)

theorem **mul_prodRFJ_colE** (ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (β' : n → k) (q : (GL n k × N k n c (ρ + 1)) × (n → k)) : colE c (ρ + 1) β' ∗ (prodRFJ ρ q : GL (n ⊕ Fin c) k) = prodRFJ ρ ((q.1.1, cjE ρ h1 h2 (((q.1.1)⁻¹ : GL n k).val ∗ᵥ β') q.1.2), ((q.1.1)⁻¹ : GL n k).val ∗ᵥ β' + q.2)

theorem **continuous_conjN** (ρ : ℕ) (u' : N k n c ρ) : Continuous fun g : GL n k => conjN ρ g u'

theorem **continuous_cjE** (ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) : Continuous fun x : (n → k) × N k n c (ρ + 1) => cjE ρ h1 h2 x.1 x.2

theorem **continuous_prodRFJ** (ρ : ℕ) : Continuous (prodRFJ (k

theorem **continuous_invMulVec** (β' : n → k) : Continuous fun g : GL n k => ((g⁻¹ : GL n k)).val ∗ᵥ β'

theorem **map_mul_left_of_conj** {Q G : Type} [MeasurableSpace Q] [Group G] [MeasurableSpace G] [MeasurableMul G] (μ : Measure Q) (f : Q → G) (hf : Measurable f) (h : G) (T : Q → Q) (hT : MeasurePreserving T μ μ) (he : ∀ q, h ∗ f q = f (T q)) : (μ.map f).map (h ∗ ·) = μ.map f

theorem **map_mul_left_mul** {G : Type} [Group G] [MeasurableSpace G] [MeasurableMul G] (κ : Measure G) (a b : G) (ha : κ.map (a ∗ ·) = κ) (hb : κ.map (b ∗ ·) = κ) : κ.map ((a ∗ b) ∗ ·) = κ

theorem **measurable_fibFJ** (ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (β' : n → k) : Measurable (Function.uncurry (fibFJ ρ h1 h2 β'))

theorem **map_fibFJ** (ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (ν : Measure k) [ν.IsAddHaarMeasure] [SFinite (volN ν n c (ρ + 1))] (β' : n → k) (g : GL n k) : ((volN ν n c (ρ + 1)).prod (Measure.pi fun _ : n => ν)).map (fibFJ ρ h1 h2 β' g) = (volN ν n c (ρ + 1)).prod (Measure.pi fun _ : n => ν)

theorem **mp_colE** (ρ : ℕ) (h1 : 1 ≤ ρ) (h2 : ρ < c) (ν : Measure k) [ν.IsAddHaarMeasure] [SFinite (volN ν n c (ρ + 1))] (m : Measure (GL n k)) [SFinite m] (β' : n → k) : MeasurePreserving (fun q : (GL n k × N k n c (ρ + 1)) × (n → k) => ((q.1.1, cjE ρ h1 h2 (((q.1.1)⁻¹ : GL n k).val ∗ᵥ β') q.1.2), ((q.1.1)⁻¹ : GL n k).val ∗ᵥ β' + q.2)) ((m.prod (volN ν n c (ρ + 1))).prod (Measure.pi fun _ : n => ν)) ((m.prod (volN ν n c (ρ + 1))).prod (Measure.pi fun _ : n => ν))

theorem **mp_N** (ρ : ℕ) (ν : Measure k) [ν.IsAddHaarMeasure] (V : Measure (N k n c (ρ + 1))) [V.IsMulLeftInvariant] [SFinite V] (m : Measure (GL n k)) [SFinite m] (u' : N k n c (ρ + 1)) : MeasurePreserving (fun q : (GL n k × N k n c (ρ + 1)) × (n → k) => ((q.1.1, conjN (ρ + 1) q.1.1 u' ∗ q.1.2), q.2)) ((m.prod V).prod (Measure.pi fun _ : n => ν)) ((m.prod V).prod (Measure.pi fun _ : n => ν))

theorem **mp_embX** (ρ : ℕ) (ν : Measure k) [ν.IsAddHaarMeasure] (V : Measure (N k n c (ρ + 1))) [SFinite V] (m : Measure (GL n k)) [m.IsMulLeftInvariant] [SFinite m] (g' : GL n k) : MeasurePreserving (fun q : (GL n k × N k n c (ρ + 1)) × (n → k) => ((g' ∗ q.1.1, q.1.2), q.2)) ((m.prod V).prod (Measure.pi fun _ : n => ν)) ((m.prod V).prod (Measure.pi fun _ : n => ν))

theorem **isMulLeftInvariant_map_prodRFJ** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ), 1 ≤ ρ → ρ < c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → (((m.prod (volN ν n c (ρ + 1))).prod (Measure.pi fun _ : n => ν)).map (prodRFJ ρ)).IsMulLeftInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsMulLeftInvariantMapLevyProductMapToFourierJacoStatement

theorem **_root_.GlobalTheory.T.isMulLeftInvariant_map_levyProductMapToFourierJacobi** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ), 1 ≤ ρ → ρ < c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → (((m.prod (volN ν n c (ρ + 1))).prod (Measure.pi fun _ : n => ν)).map (prodRFJ ρ)).IsMulLeftInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsMulRightInvariantMapWittFlagCoordinatesHaar

theorem **measurePreserving_skew_add** {A B : Type∗} [AddGroup A] [MeasurableSpace A] [MeasurableAdd₂ A] [MeasurableSpace B] (μ : Measure A) (ν : Measure B) [SFinite μ] [SFinite ν] [μ.IsAddLeftInvariant] {f : B → A} (hf : Measurable f) : MeasurePreserving (fun p : A × B => (f p.2 + p.1, p.2)) (μ.prod ν) (μ.prod ν)

theorem **isMulRightInvariant_map** {X G : Type∗} [MeasurableSpace X] [MeasurableSpace G] [Mul G] (P : Measure X) (e : X ≃ᵐ G) (h : ∀ q, ∃ T : X → X, MeasurePreserving T P P ∧ ∀ p, e (T p) = e p ∗ e q) : (P.map e).IsMulRightInvariant

theorem **coord_mul** {D : D7} {V V₀ : HSp D} {ρ : ℕ} (d : HypDec V V₀ ρ) (c c' : skewA D ρ) (b b' : Mx D ρ V₀.rank) (u u' : GL (Fin ρ) (AE D)) : d.nC ((skewConj u.1 c' + cComm V₀ b (u.1 · b')) + c) ∗ (d.nB (u.1 · b' + b) ∗ d.levi (u ∗ u')) = d.nC c ∗ (d.nB b ∗ d.levi u) ∗ (d.nC c' ∗ (d.nB b' ∗ d.levi u'))

theorem **measurableAdd₂_addSubgroup** {A : Type∗} [AddGroup A] [MeasurableSpace A] [MeasurableAdd₂ A] (H : AddSubgroup A) : MeasurableAdd₂ H

theorem **measurePreserving_triple** {A B U : Type∗} [AddGroup A] [MeasurableSpace A] [MeasurableAdd₂ A] [AddGroup B] [MeasurableSpace B] [MeasurableAdd₂ B] [Mul U] [MeasurableSpace U] [MeasurableMul U] (l : Measure A) (μ : Measure B) (ν : Measure U) [SFinite l] [SFinite μ] [SFinite ν] [l.IsAddLeftInvariant] [μ.IsAddLeftInvariant] [ν.IsMulRightInvariant] {f : B × U → A} (hf : Measurable f) {g : U → B} (hg : Measurable g) (u' : U) : MeasurePreserving (fun p : A × (B × U) => (f p.2 + p.1, (g p.2.2 + p.2.1, p.2.2 ∗ u'))) (l.prod (μ.prod ν)) (l.prod (μ.prod ν))

theorem **continuous_UX_val** (D : D7) (ρ : ℕ) : Continuous fun u : UX D ρ => (u.1.1 : Mx D ρ ρ)

theorem **continuous_UX_mul** {D : D7} {ρ n : ℕ} (b' : Mx D ρ n) : Continuous fun u : UX D ρ => (u.1.1 · b' : Mx D ρ n)

theorem **continuous_skewConj** {D : D7} {ρ : ℕ} (c' : skewA D ρ) : Continuous fun a : Mx D ρ ρ => skewConj a c'

theorem **continuous_cComm** {D : D7} (V₀ : HSp D) (ρ : ℕ) : Continuous fun x : Mx D ρ V₀.rank × Mx D ρ V₀.rank => cComm V₀ x.1 x.2

theorem **continuous_shift** {D : D7} (V₀ : HSp D) (ρ : ℕ) (c' : skewA D ρ) (b' : Mx D ρ V₀.rank) : Continuous fun y : Mx D ρ V₀.rank × UX D ρ => skewConj y.2.1.1 c' + cComm V₀ y.1 (y.2.1.1 · b')

theorem **measurable_shift** {D : D7} (V₀ : HSp D) (ρ : ℕ) (c' : skewA D ρ) (b' : Mx D ρ V₀.rank) : Measurable fun y : Mx D ρ V₀.rank × UX D ρ => skewConj y.2.1.1 c' + cComm V₀ y.1 (y.2.1.1 · b')

theorem **isMulRightInvariant_map_coord** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d), (∀ p, (e p : GA D V) = d.nC p.1 ∗ (d.nB p.2.1 ∗ d.levi p.2.2.1)) → ∀ (lC : Measure (skewA D ρ)), lC.IsAddHaarMeasure → SFinite lC → ((lC.prod (vb.μ.prod vU.ν)).map e).IsMulRightInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsMulRightInvariantMapWittFlagCoordinatesHaarStatement

theorem **_root_.GlobalTheory.T.isMulRightInvariant_map_wittFlagCoordinates_haar** : ∀ (D : D7) (V V₀ : HSp D) (ρ : ℕ) (d : HypDec V V₀ ρ) (vb : VolM D ρ V₀.rank) (vU : VolU D ρ) (e : skewA D ρ × (Mx D ρ V₀.rank × UX D ρ) ≃ᵐ flagN d), (∀ p, (e p : GA D V) = d.nC p.1 ∗ (d.nB p.2.1 ∗ d.levi p.2.2.1)) → ∀ (lC : Measure (skewA D ρ)), lC.IsAddHaarMeasure → SFinite lC → ((lC.prod (vb.μ.prod vU.ν)).map e).IsMulRightInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsOneCombSphLevel

theorem **isOneComb_sph_level** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c P : ℕ), 1 ≤ P → P ≤ c → ∀ (ψ : AddChar k Circle), Continuous ψ → HasConductor ψ 1 → ∀ ν : Measure k, IsUnitVol ν → ∀ (L : Level (GL (n ⊕ Fin c) k)), L.J = KO k _ → (∀ j, L.lam j = 1) → IsOneComb ψ ν P L 1 (Finsupp.single 1 1) ∧ prX (Cgrp L.J 1) = KO k n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsOpenMirabolicConjInfIntegralGLUnipotent

theorem **isOpen_inf_isCompact** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A : Type} [Fintype A] [DecidableEq A] {c : ℕ} (J : Subgroup (GL (A ⊕ Fin c) k)) (g : GL (A ⊕ Fin c) k), IsOpen (J : Set (GL (A ⊕ Fin c) k)) → IsCompact (J : Set (GL (A ⊕ Fin c) k)) → IsOpen ((((Cgrp J g).subgroupOf (N k A c 1)) ⊓ ((KO k (A ⊕ Fin c)).subgroupOf (N k A c 1)) : Subgroup (N k A c 1)) : Set (N k A c 1)) ∧ IsCompact (((Cgrp J g).subgroupOf (N k A c 1) : Subgroup (N k A c 1)) : Set (N k A c 1)) ∧ IsCompact (((KO k (A ⊕ Fin c)).subgroupOf (N k A c 1) : Subgroup (N k A c 1)) : Set (N k A c 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsOpenMirabolicConjInfIntegralGLUnipotentStatement

theorem **_root_.GlobalTheory.T.isOpen_mirabolicConj_inf_integralGL_unipotent** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A : Type} [Fintype A] [DecidableEq A] {c : ℕ} (J : Subgroup (GL (A ⊕ Fin c) k)) (g : GL (A ⊕ Fin c) k), IsOpen (J : Set (GL (A ⊕ Fin c) k)) → IsCompact (J : Set (GL (A ⊕ Fin c) k)) → IsOpen ((((Cgrp J g).subgroupOf (N k A c 1)) ⊓ ((KO k (A ⊕ Fin c)).subgroupOf (N k A c 1)) : Subgroup (N k A c 1)) : Set (N k A c 1)) ∧ IsCompact (((Cgrp J g).subgroupOf (N k A c 1) : Subgroup (N k A c 1)) : Set (N k A c 1)) ∧ IsCompact (((KO k (A ⊕ Fin c)).subgroupOf (N k A c 1) : Subgroup (N k A c 1)) : Set (N k A c 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsOpenMirabolicProjectionToGL

theorem **continuous_embX** {k : Type} [Field k] [TopologicalSpace k] [IsTopologicalRing k] {A : Type} [Fintype A] [DecidableEq A] (c : ℕ) : Continuous (embX c : GL A k → GL (A ⊕ Fin c) k)

theorem **isOpen_prX** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A : Type} [Fintype A] [DecidableEq A] {c : ℕ} (Ck : Subgroup (GL (A ⊕ Fin c) k)) (O : Set (GL (A ⊕ Fin c) k)), IsOpen O → (1 : GL (A ⊕ Fin c) k) ∈ O → (∀ z ∈ R k A c 1, z ∈ O → z ∈ Ck) → IsOpen (prX Ck : Set (GL A k))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsOpenMirabolicProjectionToGLStatement

theorem **_root_.GlobalTheory.T.isOpen_mirabolicProjectionToGL** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A : Type} [Fintype A] [DecidableEq A] {c : ℕ} (Ck : Subgroup (GL (A ⊕ Fin c) k)) (O : Set (GL (A ⊕ Fin c) k)), IsOpen O → (1 : GL (A ⊕ Fin c) k) ∈ O → (∀ z ∈ R k A c 1, z ∈ O → z ∈ Ck) → IsOpen (prX Ck : Set (GL A k))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPatOne

theorem **isPat_one** {K : Type} [CommRing K] {r : ℕ} (ρ κ : ℕ) : IsPat ρ κ (1 : GL (Fin r) K)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelCompIdempotentSplitTensor

theorem **smul_apply'** (b : AE D) (t : WO S T) (i j) : (b • t) i j = b * t i j

theorem **smul_half_zero** {a b : AE D} (hab : a * b = 0) {x : WO S T} (hx : IsHalf a x) : b • x = 0

theorem **half_smul** {b : AE D} (hb : b * b = b) (t : WO S T) : IsHalf b (b • t)

theorem **tens_trans** {a b : AE D} (hab : a * b = 0) (Φ Ψ : WO S T → ℂ) {x : WO S T} (hx : IsHalf a x) : tens b (fun t => Φ (t + x)) Ψ = fun t => tens b Φ Ψ (t + x)

theorem **tens_modul** (ψ : AddChar (AF D.host.F) ℂ) {a b : AE D} (hab : a * b = 0) (hb : b * b = b) (Φ Ψ : WO S T → ℂ) {y : WO S T} (hy : IsHalf (cA D a) y) : tens b (fun t => ψ (symplO S T t y) * Φ t) Ψ = fun t => ψ (symplO S T t y) * tens b Φ Ψ t

theorem **tens_vac** {a b : AE D} (hab : a * b = 0) (hb : b * b = b) (Λ : AddSubgroup (WO S T)) (hΛ : ∀ u ∈ Λ, b • u ∈ Λ) : tens b ((1 : ℂ) • halfInd a Λ) (halfInd b Λ) = (1 : ℂ) • halfInd (a + b) Λ

theorem **GlobalTheory.FinPl.PutIn.isPlaceModel_tensHalf** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (a b : AE D) (Λ : AddSubgroup (WO S T)) (f₀ : WO S T → ℂ) (Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), a * b = 0 → b * b = b → (∀ u ∈ Λ, b • u ∈ Λ) → IsPlaceModel ψ (symplO S T) (heisO ψ S T) (a + b) Λ 1 f₀ Tm → IsPlaceModel ψ (symplO S T) (heisO ψ S T) a Λ 1 f₀ (Tm ∘ₗ GlobalTheory.FinPl.PutIn.tensL b (halfInd b Λ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelCompIdempotentSplitTensorStatement

theorem **_root_.GlobalTheory.T.isPlaceModel_comp_idempotentSplitTensor** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (a b : AE D) (Λ : AddSubgroup (WO S T)) (f₀ : WO S T → ℂ) (Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), a * b = 0 → b * b = b → (∀ u ∈ Λ, b • u ∈ Λ) → IsPlaceModel ψ (symplO S T) (heisO ψ S T) (a + b) Λ 1 f₀ Tm → IsPlaceModel ψ (symplO S T) (heisO ψ S T) a Λ 1 f₀ (Tm ∘ₗ GlobalTheory.FinPl.PutIn.tensL b (halfInd b Λ))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelProdHalf

theorem **prodHalf_apply** {m n : ℕ} (a b : AE D) (Ψ Φ : Mx D m n → ℂ) (x : Mx D m n) : prodHalf a b Ψ Φ x = Φ (a • x) * Ψ (b • x) * (if IsHalf (a + b) x then 1 else 0)

theorem **combL_apply** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (d : WO S T →₀ ℂ) (f : WO S T → ℂ) : combL ψ S T d f = comb ψ S T d f

theorem **isSBHalf_prodHalf** {m n : ℕ} {a b : AE D} (ha : a * a = a) (hb : b * b = b) {Λ : AddSubgroup (Mx D m n)} (hΛ : ∀ e : AE D, e * e = e → ∀ u ∈ Λ, e • u ∈ Λ) {Ψ Φ : Mx D m n → ℂ} (hΨ : IsSBHalf b Λ Ψ) (hΦ : IsSBHalf a Λ Φ) : IsSBHalf (a + b) Λ (prodHalf a b Ψ Φ)

theorem **isPlaceModel_prodHalf** {S T : HSp D} {ψ : AddChar (AF D.host.F) ℂ} {Λ : AddSubgroup (WO S T)} {f₀ : WO S T → ℂ} {a b : AE D} {T₀ : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)} {Ψ : WO S T → ℂ} (ha : a * a = a) (hb : b * b = b) (hab : a * b = 0) (hΛ : ∀ e : AE D, e * e = e → ∀ u ∈ Λ, e • u ∈ Λ) (hΨ : IsSBHalf b Λ Ψ) (hT : IsPlaceModel ψ (symplO S T) (heisO ψ S T) (a + b) Λ 1 f₀ T₀) : IsPlaceModel ψ (symplO S T) (heisO ψ S T) a Λ 1 (T₀ (prodHalf a b Ψ (halfInd a Λ))) (T₀ ∘ₗ prodHalf a b Ψ)

theorem **placeModel_eq** {S T : HSp D} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) {e : AE D} (he : e * e = e) (hinf : infPart D e = 0) {f₁ : WO S T → ℂ} {T₁ T₂ : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)} (h₁ : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) 1 f₁ T₁) (h₂ : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) 1 f₁ T₂) {Φ : WO S T → ℂ} (hΦ : IsSBHalf e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) Φ) : T₁ Φ = T₂ Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelTensorIntegralIndicator

theorem **pi_e** : p.π p.e = 1

theorem **pi_of_mul** {e : AE D} (he : e * p.e = 0) : p.π e = 0

theorem **up_down_eq** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (t : WO S T) : p.up FS FT (p.down FS FT t) = t.map (fun a => p.e * a)

theorem **down_half_zero** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e * p.e = 0) {x : WO S T} (hx : IsHalf e x) : p.down FS FT x = 0

theorem **half_sum** {e : AE D} (he : e * p.e = 0) {x : WO S T} (hx : IsHalf e x) : IsHalf (e + p.e) x

theorem **half_sum_conj** {e : AE D} (he : e * p.e = 0) {y : WO S T} (hy : IsHalf (cA D e) y) : IsHalf (cA D (e + p.e)) y

theorem **hO_zero** {e : AE D} (he : e * p.e = 0) {w y : WO S T} (hw : IsHalf p.e w) (hy : IsHalf (cA D e) y) : hO S T w y = 0

theorem **symplO_sub_half** {e : AE D} (he : e * p.e = 0) (t : WO S T) {w y : WO S T} (hw : IsHalf p.e w) (hy : IsHalf (cA D e) y) : symplO S T (t - w) y = symplO S T t y

theorem **tensorOne_trans** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e * p.e = 0) (Φ : WO S T → ℂ) {x : WO S T} (hx : IsHalf e x) : p.tensorOne FS FT (fun t => Φ (t + x)) = fun t => p.tensorOne FS FT Φ (t + x)

theorem **tensorOne_modul** (ψ : AddChar (AF D.host.F) ℂ) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e * p.e = 0) (Φ : WO S T → ℂ) {y : WO S T} (hy : IsHalf (cA D e) y) : p.tensorOne FS FT (fun t => ψ (symplO S T t y) * Φ t) = fun t => ψ (symplO S T t

y) ∗ p.tensorOne FS FT Φ t

theorem **intInd_eq** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (hint : ∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ↔ p.up FS FT x ∈ stdHom S T) (x : Matrix Y Y' k) : Place.intInd Y Y' x = if p.up FS FT x ∈ stdHom S T then 1 else 0

theorem **tensorOne_vac** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e ∗ p.e = 0) (hint : ∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ↔ p.up FS FT x ∈ stdHom S T) (hstab : ∀ u ∈ stdHom S T, u.map (fun a => p.e ∗ a) ∈ stdHom S T) : p.tensorOne FS FT ((1 : ℂ) • halfInd e (stdHom S T)) = (1 : ℂ) • halfInd (e + p.e) (stdHom S T)

theorem **isPlaceModel_tensorOne** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (e : AE D), e ∗ p.e = 0 → (∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ↔ p.up FS FT x ∈ stdHom S T) → (∀ u ∈ stdHom S T, u.map (fun a => p.e ∗ a) ∈ stdHom S T) → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) (e + p.e) (stdHom S T) 1 f₀ T' → (∀ Φ, IsSBHalf e (stdHom S T) Φ → IsSBHalf (e + p.e) (stdHom S T) (p.tensorAt FS FT Φ (Place.intInd Y Y'))) → IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (stdHom S T) 1 f₀ (T' ∘ₗ p.tensorOne FS FT)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelTensorIntegralIndicatorStatement

theorem **_root_.GlobalTheory.T.isPlaceModel_tensorIntegralIndicator** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (e : AE D), e ∗ p.e = 0 → (∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ↔ p.up FS FT x ∈ stdHom S T) → (∀ u ∈ stdHom S T, u.map (fun a => p.e ∗ a) ∈ stdHom S T) → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) (e + p.e) (stdHom S T) 1 f₀ T' → (∀ Φ, IsSBHalf e (stdHom S T) Φ → IsSBHalf (e + p.e) (stdHom S T) (p.tensorAt FS FT Φ (Place.intInd Y Y'))) → IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (stdHom S T) 1 f₀ (T' ∘ₗ p.tensorOne FS FT)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelTensorSplitPlaceLatticeIndicator

theorem **latO_at_le** {S T : HSp D} (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (v : Pl D) : latO S T LS LT (· = v) ≤ latO S T LS LT fun _ => True

theorem **pairA_zero_mul** (a b : Af D.host.F) : pairA D.host.F 0 a ∗ pairA D.host.F 0 b = pairA D.host.F 0 (a ∗ b)

theorem **placeIdem_mul_mem** {S T : HSp D} (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) {v : Pl D} {ε : AF D.host.F} (hε : IsPlaceIdem D v ε) {u : WO S T} (hu : u ∈ latO S T LS LT fun _ => True) : (Matrix.of fun i j => algebraMap (AF D.host.F) (AE D) ε ∗ u i j : WO S T) ∈ latO S T LS LT (· = v)

theorem **splitIdem_mul_mem** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : SplitPt D k) {S T : HSp D} (LS : Lat S) (LT : Lat T) {u : WO S T} (hu : u ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) fun _ => True) : u.map (fun a => q.e ∗ a) ∈ latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = q.v)

theorem **tensorWith_apply** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (Ψ : Matrix Y Y' k → ℂ) (Φ : WO S T → ℂ) : tensorWith p FS FT Ψ Φ = p.tensorAt FS FT Φ Ψ

theorem **tensorAt_trans** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e ∗ p.e = 0) (Φ : WO S T → ℂ) (Ψ : Matrix Y Y' k → ℂ) {x : WO S T} (hx : IsHalf e x) : p.tensorAt FS FT (fun t => Φ (t + x)) Ψ = fun t => p.tensorAt FS FT Φ Ψ (t + x)

theorem **tensorAt_modul** (ψ : AddChar (AF D.host.F) ℂ) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e ∗ p.e = 0) (Φ : WO S T → ℂ) (Ψ : Matrix Y Y' k → ℂ) {y : WO S T} (hy : IsHalf (cA D e) y) : p.tensorAt FS FT (fun t => ψ (symplO S T t y) ∗ Φ t) Ψ = fun t => ψ (symplO S T t y) ∗ p.tensorAt FS FT Φ Ψ t

theorem **tensorAt_halfInd** (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) {e : AE D} (he : e ∗ p.e = 0) (Λ Λv : AddSubgroup (WO S T)) (hle : Λv ≤ Λ) (hstab : ∀ t ∈ Λ, t.map (fun a => p.e ∗ a) ∈ Λv) : p.tensorAt FS FT ((1 : ℂ) • halfInd e Λ) (Set.indicator {x | p.up FS FT x ∈ Λv} fun _ => 1) = (1 : ℂ) • halfInd (e + p.e) Λ

theorem **isPlaceModel_tensorLat** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M : Lat S) (M' : Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (P : LocPl ψ S T M M') (e : AE D), e ∗ P.q.e = 0 → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) (e + P.q.e) (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 f₀ T' → ∃ T'' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ), IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 f₀ T'' ∧ ∀ Φ, T'' Φ = T' ((ofSplit P.q).tensorAt P.FS P.FT Φ (Set.indicator {x | (ofSplit P.q).up P.FS P.FT x ∈ latO S T (M : Set _) (M' : Set _) (· = P.q.v)} fun _ => 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPlaceModelTensorSplitPlaceLatticeIndicatorStatement

theorem **_root_.GlobalTheory.T.isPlaceModel_tensor_splitPlaceLatticeIndicator** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M : Lat S) (M' : Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (P : LocPl ψ S T M M') (e : AE D), e ∗ P.q.e = 0 → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsPlaceModel ψ (symplO S T) (heisO ψ S T) (e + P.q.e) (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 f₀ T' → ∃ T'' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ), IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 f₀ T'' ∧ ∀ Φ, T'' Φ = T' ((ofSplit P.q).tensorAt P.FS P.FT Φ (Set.indicator {x | (ofSplit P.q).up P.FS P.FT x ∈ latO S T (M : Set _) (M' : Set _) (· = P.q.v)} fun _ => 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPointOfDecoratedLatticeUnique

theorem **mem_iff_hatL** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [StarRing E] {n : ℕ} {N : Submodule (𝒪 E) (Sp E n)} (hN : Submodule.IsLattice E N) (v : Sp E n) : v ∈ N ↔ (fun i => Formalism.UG.inclE (F

theorem **isPtOf_unique** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [StarRing E] (hc : Formalism.BasicNotions.CMData F E) {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} (g : GL (Fin n) (Formalism.L2.DIV.AfE F E)) (p p' : Formalism.L2.Col.DecoF E n ST SG Q), GlobalTheory.Mem.Col2.IsPtOf hc g p → GlobalTheory.Mem.Col2.IsPtOf hc g p' → p = p'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPointOfDecoratedLatticeUniqueStatement

theorem **_root_.GlobalTheory.T.isPointOfDecoratedLattice_unique** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] [StarRing E] (hc : Formalism.BasicNotions.CMData F E) {n : ℕ} {ST SG Q : Finset (HeightOneSpectrum (𝒪 E))} (g : GL (Fin n) (Formalism.L2.DIV.AfE F E)) (p p' : Formalism.L2.Col.DecoF E n ST SG Q), GlobalTheory.Mem.Col2.IsPtOf hc g p → GlobalTheory.Mem.Col2.IsPtOf hc g p' → p = p'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPointOfGenericCorankOne

theorem **occurs_iff_generic_corank_one** (νL νB : X.B.I.Ginf → Measure X.B.I.Ginf) (ng : PN.Pts D.host.E D.R.rS → ℝ) (c : Set (TauC Y u u)) (γ : PN.Pts D.host.E D.R.rS) (ν : NuC D) : PN.Occ (genT (scT (Y.tO u u))) (NOf1 Y u νL νB ng) c γ ν ↔ ∃ t, IsGenericCorankOneTerm Y u c γ t ∧ nmlOf Y u u t = ν

theorem **related_of_generic_corank_one** (hE : CorankOneIsEquivalenceOnRow) (hT : ConcTie X Y) {c : Set (TauC Y u u)} {γ : PN.Pts D.host.E D.R.rS} {t0 t : TauC Y u u} (h0 : IsGenericCorankOneTerm Y u c γ t0) (h : IsGenericCorankOneTerm Y u c γ t) : RelC Y u u t0 t

theorem **isPoint_of_generic_corank_one** {νL νB : X.B.I.Ginf → Measure X.B.I.Ginf} {ng : PN.Pts D.host.E D.R.rS → ℝ} {c : Set (TauC Y u u)} {γ : PN.Pts D.host.E D.R.rS} {t : TauC Y u u} (h : IsGenericCorankOneTerm Y u c γ t) : PN.IsPt (genT (Y.tO u u)) (NOfG Y u νL νB ng) γ

theorem **nmlInvOne_of** (hE : CorankOneIsEquivalenceOnRow) (hL : NmlLine) (hI : NmlInt) (hS : NmlSep) : NmlInvOne

theorem **stepNormals1_of** : CorankOneIsEquivalenceOnRow → Formalism.L2.SG.NGCut.NormalVectorPairings.NmlLine → Formalism.L2.SG.NGCut.NormalVectorPairings.NmlInt → Formalism.L2.SG.NGCut.NormalVectorPairings.NmlSep → Formalism.L2.SG.NGCut.NormalInvariants.NormCount → StepNormOneG → StepNormals1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPolynomialFockThetaVectorOrthogonalBoxProduct

theorem **killed_of_leibniz₂** {P₁ P₂ Q U₁ U₂ V : Type} [Zero U₁] [Zero U₂] [AddCommGroup V] (a₁ : P₁ → U₁ → U₁) (a₂ : P₂ → U₂ → U₂) (b : Q → V → V) (T₁ : U₁ → Prop) (T₂ : U₂ → Prop) (TV : V → Prop) (δ : U₁ → U₂ → V) (c₁ : Q → P₁) (c₂ : Q → P₂) (ha₁ : ∀ p f, T₁ f → T₁ (a₁ p f)) (ha₂ : ∀ p f, T₂ f → T₂ (a₂ p f)) (hb : ∀ q f, TV f → TV (b q f)) (hadd : ∀ q f g, TV f → TV g → b q (f + g) = b q f + b q g) (hb0 : ∀ q, b q 0 = 0) (hδ : ∀ f g, T₁ f → T₂ g → TV (δ f g)) (h1 : ∀ g, δ 0 g = 0) (h2 : ∀ f, δ f 0 = 0) (hL : ∀ q f g, T₁ f → T₂ g → b q (δ f g) = δ (a₁ (c₁ q) f) g + δ f (a₂ (c₂ q) g)) (N : ℕ) : ∀ (N₁ N₂ : ℕ) (f : U₁) (g : U₂), N₁ + N₂ ≤ N → T₁ f → T₂ g → (∀ l : List P₁, l.length = N₁ → l.foldr a₁ f = 0) → (∀ l : List P₂, l.length = N₂ → l.foldr a₂ g = 0) → ∀ l : List Q, l.length = N → l.foldr b (δ f g) = 0

theorem **lineAt_mul_left** (hθ : IsInfFam θ) {k m n : ℕ} (p : Mx D k m) (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) (t : ℝ) : lineAt θ τ (Formalism.IndefR.Datum.compAt θ τ p * z) t = (HMul.hMul p (lineAt θ τ z t) : Mx D k n)

theorem **compAt_pA** (hθ : IsInfFam θ) (τ : D.what₂.Tau) : Formalism.IndefR.Datum.compAt θ τ o.pA = o.qA.map (tauE D τ)

theorem **compAt_pB** (hθ : IsInfFam θ) (τ : D.what₂.Tau) : Formalism.IndefR.Datum.compAt θ τ o.pB = o.qB.map (tauE D τ)

theorem **dRho_boxT** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (z : Matrix (Fin B.rank) (Fin A.rank) ℂ) {f : WO A W → ℂ} {g : WO A H → ℂ} (hf : IsThetaFunO ψ A W f) (hg : IsThetaFunO ψ A H g) : Formalism.IndefR.Datum.dRho (heisO ψ A B) θ τ z (o.boxT A f g) = o.boxT A (Formalism.IndefR.Datum.dRho (heisO ψ A W) θ τ (o.qA.map (tauE D τ) * z) f) g + o.boxT A f (Formalism.IndefR.Datum.dRho (heisO ψ A H) θ τ (o.qB.map (tauE D τ) * z) g)

theorem **qA_sumMat** (sW : RealMat W) (sH : RealMat H) (τ : D.what₂.Tau) : o.qA.map (tauE D τ) * sumMat o sW sH τ = sW τ * o.qA.map (tauE D τ)

theorem **qB_sumMat** (sW : RealMat W) (sH : RealMat H) (τ : D.what₂.Tau) : o.qB.map (tauE D τ) * sumMat o sW sH τ = sH τ * o.qB.map (tauE D τ)

theorem **mul_cplxO** {k m n : ℕ} {q : Matrix (Fin k) (Fin m) ℂ} {s : Matrix (Fin m) (Fin m) ℂ} {s' : Matrix (Fin k) (Fin k) ℂ} (h : q * s = s' * q) (r : Matrix (Fin n) (Fin n) ℂ) (z : Matrix (Fin m) (Fin n) ℂ) : q * cplxO r s z = cplxO r s' (q * z)

theorem **annih_boxT** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (s : Matrix (Fin A.rank) (Fin A.rank) ℂ) (sW : RealMat W) (sH : RealMat H) (z : Matrix (Fin B.rank) (Fin A.rank) ℂ) {f : WO A W → ℂ} {g : WO A H → ℂ} (hf : IsThetaFunO ψ A W f) (hg : IsThetaFunO ψ A H g) : annih (heisO ψ A B) θ τ (cplxO s (sumMat o sW sH τ)) z (o.boxT A f g) = o.boxT A (annih (heisO ψ A W) θ τ (cplxO s (sW τ)) (o.qA.map (tauE D τ) * z) f) g + o.boxT A f (annih (heisO ψ A H) θ τ (cplxO s (sH τ)) (o.qB.map (tauE D τ) * z) g)

theorem **isFockVec_boxT** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B W H : HSp D) (o : OrthDec B W H) (sA : RealMat A) (sW : RealMat W) (sH : RealMat H), IsStdChar D ψ (-1) → IsInfFam θ → ∀ Φ Φ', IsFockVec ψ θ A W sA sW Φ → IsFockVec ψ θ A H sA sH Φ' → IsFockVec ψ θ A B sA (sumMat o sW sH) (o.boxT A Φ Φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPolynomialFockThetaVectorOrthogonalBoxProductStatement

theorem **_root_.GlobalTheory.T.isPolynomialFockThetaVector_orthogonalBoxProduct** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A B W H : HSp D) (o : OrthDec B W H) (sA : RealMat A) (sW : RealMat W) (sH : RealMat H), IsStdChar D ψ (-1) → IsInfFam θ → ∀ Φ Φ', IsFockVec ψ θ A W sA sW Φ → IsFockVec ψ θ A H sA sH Φ' → IsFockVec ψ θ A B sA (sumMat o sW sH) (o.boxT A Φ Φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPolynomialFockThetaVectorWeilOperatorCartan

theorem **theta_annih** (hθ : IsInfFam θ) (hψ : IsStdChar D ψ (-1)) (τ : D.what₂.Tau) (J : Matrix (Fin T.rank) (Fin S.rank) ℂ → Matrix (Fin T.rank) (Fin S.rank) ℂ) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (annih (heisO ψ S T) θ τ J z f)

theorem **om_annih** (hW : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) (τ : D.what₂.Tau) {sS sS' : Matrix (Fin S.rank) (Fin S.rank) ℂ} {sT sT' : Matrix (Fin T.rank) (Fin T.rank) ℂ} (hS : sS' * compAt θ τ (mt S g) = compAt θ τ (mt S g) * sS) (hT : sT' * compAt θ τ (mt T h) = compAt θ τ (mt T h) * sT) (z : Matrix (Fin T.rank) (Fin S.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : om (g, h) (annih (heisO ψ S T) θ τ (cplxO sS sT) z f) = annih (heisO ψ S T) θ τ (cplxO sS' sT') (compAt θ τ (mt T h) * z * compAt θ τ (mt S g⁻¹)) (om (g, h) f)

theorem **om_foldr** (hW : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (g : GA D S) (h : GA D T) {sS sS' : RealMat S} {sT sT' : RealMat T} (hS : ∀ τ, sS' τ * compAt θ τ (mt S g) = compAt θ τ (mt S g) * sS τ) (hT : ∀ τ, sT' τ * compAt θ τ (mt T h) = compAt θ τ (mt T h) * sT τ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (l : List (D.what₂.Tau × Matrix (Fin T.rank) (Fin S.rank) ℂ)) : IsThetaFunO ψ S T (l.foldr (fun p F => annih (heisO ψ S T) θ p.1 (cplxO (sS p.1) (sT p.1)) p.2 F) f) ∧ om (g, h) (l.foldr (fun p F => annih (heisO ψ S T) θ p.1 (cplxO (sS p.1) (sT p.1)) p.2 F) f) = (l.map fun p => (p.1, compAt θ p.1 (mt T h) * p.2 * compAt θ p.1 (mt S g⁻¹))).foldr (fun p F => annih (heisO ψ S T) θ p.1 (cplxO (sS' p.1) (sT' p.1)) p.2 F) (om (g, h) f)

theorem **isFockVec_om** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ θ : InfFam D, IsInfFam θ → ∀ (sS sS' : RealMat S) (sT : RealMat T) (g : GA D S), (∀ τ, sS' τ * compAt θ τ (mt S g) = compAt θ τ (mt S g) * sS τ) → ∀ f, IsFockVec ψ θ S T sS sT f → IsFockVec ψ θ S T sS' sT (om (g, 1) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsPolynomialFockThetaVectorWeilOperatorCartanStatement

theorem **_root_.GlobalTheory.T.isPolynomialFockThetaVector_weilOperator_cartan** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ θ : InfFam D, IsInfFam

θ → ∀ (sS sS' : RealMat S) (sT : RealMat T) (g : GA D S), (∀ τ, sS' τ * compAt θ τ (mt S g) = compAt θ τ (mt S g) * sS τ) → ∀ f, IsFockVec ψ θ S T sS sT f → IsFockVec ψ θ S T sS' sT (om (g, 1) f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsQuotIntCells

theorem **isQuotInt_cells** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ) (_ : 1 ≤ p) (_ : p ≤ c) (ν : Measure k) [ν.IsAddHaarMeasure] (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (J : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsCO J) (f : GL (n ⊕ Fin c) k → ℂ) (_ : ∀ h, ∀ j ∈ J, f (h * j) = f h) (_ : ∀ (x : GL n k), ∀ u ∈ N k n c p, ∀ h, f (embX c x * u * h) = ((nabs k (Matrix.GeneralLinearGroup.det x) ^ ((c : ℝ) + 1 - 2 * p) : ℝ) : ℂ) * f h) (T : Finset (GL (n ⊕ Fin c) k)) (_ : Covers p J T f), (∀ h, volCell ν μX p J h ≠ 0 ∧ volCell ν μX p J h ≠ ⊤) ∧ ∀ z, IsQuotInt ν μX μG p f z ↔ z = ∑ h ∈ T, (((μG J).toReal / (volCell ν μX p J h).toReal : ℝ) : ℂ) * f h

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsQuotIntCellsStatement

theorem **_root_.GlobalTheory.T.isQuotInt_cells_alt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ) (_ : 1 ≤ p) (_ : p ≤ c) (ν : Measure k) [ν.IsAddHaarMeasure] (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (J : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsCO J) (f : GL (n ⊕ Fin c) k → ℂ) (_ : ∀ h, ∀ j ∈ J, f (h * j) = f h) (_ : ∀ (x : GL n k), ∀ u ∈ N k n c p, ∀ h, f (embX c x * u * h) = ((nabs k (Matrix.GeneralLinearGroup.det x) ^ ((c : ℝ) + 1 - 2 * p) : ℝ) : ℂ) * f h) (T : Finset (GL (n ⊕ Fin c) k)) (_ : Covers p J T f), (∀ h, volCell ν μX p J h ≠ 0 ∧ volCell ν μX p J h ≠ ⊤) ∧ ∀ z, IsQuotInt ν μX μG p f z ↔ z = ∑ h ∈ T, (((μG J).toReal / (volCell ν μX p J h).toReal : ℝ) : ℂ) * f h

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsQuotientInvariantIntegralExchangeToPositionOne

@[simp] theorem **omRight_apply** (μ : kˣ →* ℂˣ) (g : GL n k) (Ψ : SB (Matrix n n' k)) : ((omRight (n'

theorem **bilPos_iff_isPos** (ψ : AddChar k Circle) (μ : kˣ →* ℂˣ) (p : ℕ) (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) : BilPos ψ μ p σ B ↔ IsPos ψ p (omRight μ) σ B

theorem **GlobalTheory.PosOneIntegral.positionOne_invariantIntegral_exchange_step** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (ψ : AddChar k Circle) (hψc : Continuous ψ) (hψ1 : ψ ≠ 1) (ν : Measure k) (hν : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (hμ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (hσ : IsSmoothRep σ) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (hρ' : IsSrc μ ρ') : ∀ q : ℕ, q + 1 ≤ c → ∀ (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ), BilPos ψ μ (q + 1) σ B → ∀ (Rr : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ

theorem **GlobalTheory.PosOneIntegral.isQuotInt_toOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ P σ B) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ') (Rr : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] ℂ) (_ : ∀ Φ : SB (Matrix (n ⊕ Fin

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsQuotientInvariantIntegralExchangeToPositionOneStatement

theorem **_root_.GlobalTheory.T.isQuotientInvariantIntegral_exchange_toPositionOne** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c P : ℕ) (_ : 1 ≤ P) (_ : P ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) (_ : IsSelfDual ψ ν) (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (μX : Measure (GL n k)) [μX.IsHaarMeasure] (μG : Measure (GL (n ⊕ Fin c) k)) [μG.IsHaarMeasure] (A : Type) [AddCommGroup A] [Module ℂ A] (σ : Representation ℂ (GL (n ⊕ Fin c) k) A) (_ : IsSmoothRep σ) (B : SB (Matrix n n' k) →ₗ[ℂ] A →ₗ[ℂ] ℂ) (_ : BilPos ψ μ P σ B) (F : A) (ρ' : Representation ℂ (GL (n' ⊕ Fin (c + 1)) k) (SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k))) (_ : IsSrc μ ρ') (Rr : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] ℂ) (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsRealSSmulTwoSmul

theorem **heisO_smul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w₀ : WO S T) (c : ℂ) (f : WO S T → ℂ) : heisO ψ S T w₀ (c • f) = c • heisO ψ S T w₀ f

theorem **comb_smul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (a : WO S T →₀ ℂ) (c : ℂ) (f : WO S T → ℂ) : comb ψ S T a (c • f) = c • comb ψ S T a f

theorem **biComb_smul** (ψ : AddChar (AF D.host.F) ℂ) (A S A' : HSp D) (a : WO A S →₀ ℂ) (a' : WO S A' →₀ ℂ) (c : ℂ) (C : WO A S → WO S A' → ℂ) : biComb ψ A S A' a a' (c • C) = c • biComb ψ A S A' a a' C

theorem **isThetaFunO_smul** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (c • f) where

theorem **isLatVac_smul** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : ℂ) {Λ : AddSubgroup (WO S T)} {f : WO S T → ℂ} (hf : IsLatVac (heisO ψ S T) Λ f) : IsLatVac (heisO ψ S T) Λ (c • f)

theorem **isBiVac_smul** {ψ : AddChar (AF D.host.F) ℂ} {A S A' : HSp D} {LA : Set (Fin A.rank → D.host.E)} {LS : Set (Fin S.rank → D.host.E)} {LA' : Set (Fin A'.rank → D.host.E)} (c : ℂ) {C : WO A S → WO S A' → ℂ} (h : IsBiVac ψ A S A' LA LS LA' C) : IsBiVac ψ A S A' LA LS LA' (c • C) where

theorem **isBiSmooth_smul** {A S A' : HSp D} (c : ℂ) {C : WO A S → WO S A' → ℂ} (h : IsBiSmooth C) : IsBiSmooth (c • C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsRootOfSUnitRatioUpToRootOfUnityClosureLaws

theorem **padicValRat_pow_mul_pow_ne_zero_imp** {r s : ℚ} (hr : r ≠ 0) (hs : s ≠ 0) (a b : ℕ) (p : ℕ) (hp : p.Prime) (h : padicValRat p (r ^ a * s ^ b) ≠ 0) : padicValRat p r ≠ 0 ∨ padicValRat p s ≠ 0

theorem **padicValRat_pow_div_pow_ne_zero_imp** {r s : ℚ} (hr : r ≠ 0) (hs : s ≠ 0) (a b : ℕ) (p : ℕ) (hp : p.Prime) (h : padicValRat p (r ^ a / s ^ b) ≠ 0) : padicValRat p r ≠ 0 ∨ padicValRat p s ≠ 0

theorem **unitRatio_ne_zero** {S : ℕ → Prop} {x y : ℂ} (h : UnitRatio S x y) : x ≠ 0

theorem **unitRatio_refl** (S : ℕ → Prop) {y : ℂ} (hy : y ≠ 0) : UnitRatio S y y

theorem **unitRatio_mul** {S : ℕ → Prop} {x y x' y' : ℂ} (h : UnitRatio S x y) (h' : UnitRatio S x' y') : UnitRatio S (x * x') (y * y')

theorem **unitRatio_div** {S : ℕ → Prop} {x y x' y' : ℂ} (h : UnitRatio S x y) (h' : UnitRatio S x' y') : UnitRatio S (x / x') (y / y')

theorem **unitRatio_of_pow_eq** (S : ℕ → Prop) {x y : ℂ} {n : ℕ} (hy : y ≠ 0) (hn : 0 < n) (h : x ^ n = y ^ n) : UnitRatio S x y

theorem **unitRatio_prod** (S : ℕ → Prop) {I : Type} (s : Finset I) (x y : I → ℂ) (h : ∀ i ∈ s, UnitRatio S (x i) (y i)) : UnitRatio S (∏ i ∈ s, x i) (∏ i ∈ s, y i)

theorem **unitRatio_laws** : ∀ S : ℕ → Prop, (∀ y : ℂ, y ≠ 0 → UnitRatio S y y) ∧ (∀ x y x' y' : ℂ, UnitRatio S x y → UnitRatio S x' y' → UnitRatio S (x * x') (y * y') ∧ UnitRatio S (x / x') (y / y') ∧ x ≠ 0) ∧ (∀ (x y : ℂ) (n : ℕ), y ≠ 0 → 0 < n → x ^ n = y ^ n → UnitRatio S x y) ∧ ∀ (I : Type) (s : Finset I) (x y : I → ℂ), (∀ i ∈ s, UnitRatio S (x i) (y i)) → UnitRatio S (∏ i ∈ s, x i) (∏ i ∈ s, y i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsRootOfSUnitRatioUpToRootOfUnityClosureLawsStatement

theorem **_root_.GlobalTheory.T.isRootOfSUnitRatioUpToRootOfUnity_closure_laws** : ∀ S : ℕ → Prop, (∀ y : ℂ, y ≠ 0 → UnitRatio S y y) ∧ (∀ x y x' y' : ℂ, UnitRatio S x y → UnitRatio S x' y' → UnitRatio S (x * x') (y * y') ∧ UnitRatio S (x / x') (y / y') ∧ x ≠ 0) ∧ (∀ (x y : ℂ) (n : ℕ), y ≠ 0 → 0 < n → x ^ n = y ^ n → UnitRatio S x y) ∧ ∀ (I : Type) (s : Finset I) (x y : I → ℂ), (∀ i ∈ s, UnitRatio S (x i) (y i)) → UnitRatio S (∏ i ∈ s, x i) (∏ i ∈ s, y i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsScalarMulIntegralOfConjIntegralGenerators

theorem **single_conj_apply** {K : Type} [Field K] {n : ℕ} (A B : Matrix (Fin n) (Fin n) K) (a b i j : Fin n) : (A * (Matrix.single a b (1 : K) : Matrix (Fin n) (Fin n) K) * B) i j = A i a * B b j

theorem **single_map_subtype** {K : Type} [Field K] (O : ValuationSubring K) {n : ℕ} (a b : Fin n) : (Matrix.single a b (1 : O)).map O.subtype = (Matrix.single a b (1 : K) : Matrix (Fin n) (Fin n) K)

theorem **entry_mul_inv_entry_mem** {K : Type} [Field K] (O : ValuationSubring K) {n : ℕ} (m : GL (Fin n) K) (T : Set (GL (Fin n) O)) (hT : Algebra.adjoin O ((fun x : GL (Fin n) O => (x : Matrix (Fin n) (Fin n) O)) '' T) = ⊤) (hc : ∀ x ∈ T, ∃ x' : GL (Fin n) O, bc O.subtype x' = m * bc O.subtype x * m⁻¹) (i a b j : Fin n) : (m : Matrix (Fin n) (Fin n) K) i a * ((m⁻¹ : GL (Fin n) K) : Matrix (Fin n) (Fin n) K) b j ∈ O

theorem **isScal_of_conj** (K : Type) [Field K] (O : ValuationSubring K) (n : ℕ) (m : GL (Fin n) K) (T : Set (GL (Fin n) O)) (hT : Algebra.adjoin O ((fun x : GL (Fin n) O => (x : Matrix (Fin n) (Fin n) O)) '' T) = ⊤) (hc : ∀ x ∈ T, ∃ x' : GL (Fin n) O, bc O.subtype x' = m * bc O.subtype x * m⁻¹) : IsScal O m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsScalarMulIntegralOfConjIntegralGeneratorsStatement

theorem **_root_.GlobalTheory.T.isScalarMulIntegral_of_conj_integral_generators** : ∀ (K : Type) [Field K] (O : ValuationSubring K) (n : ℕ) (m : GL (Fin n) K) (T : Set (GL (Fin n) O)), Algebra.adjoin O ((fun x : GL (Fin n) O => (x : Matrix (Fin n) (Fin n) O)) '' T) = ⊤ → (∀ x ∈ T, ∃ x' : GL (Fin n) O, bc O.subtype x' = m * bc O.subtype x * m⁻¹) → IsScal O m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsScalarMulIntegralOfNormalizesSymmetricLevel

theorem **exists_normalise** {K : Type} [Field K] (O : ValuationSubring K) {n : ℕ} (M : Matrix (Fin n) (Fin n) K) (hM : M ≠ 0) : ∃ c : Kˣ, (∀ i j, (c : K)⁻¹ * M i j ∈ O) ∧ ∃ i j, (c : K)⁻¹ * M i j = 1

theorem **one_add_unit** {K : Type} [Field K] (O : ValuationSubring K) {k : Type} [Field k] (r : O →+* k) (hr : Function.Surjective r) {n : ℕ} (Y : Matrix (Fin n) (Fin n) O) (hY : ∀ i j, r (Y i j) = 0) : ∃ x : GL (Fin n) O, (x : Matrix (Fin n) (Fin n) O) = 1 + Y

theorem **gl_of_int** {K : Type} [Field K] (O : ValuationSubring K) {n : ℕ} (m : GL (Fin n) K) (h1 : ∀ i j, (m : Matrix (Fin n) (Fin n) K) i j ∈ O) (h2 : ∀ i j, ((m⁻¹ : GL (Fin n) K) : Matrix (Fin n) (Fin n) K) i j ∈ O) : ∃ x : GL (Fin n) O, bc O.subtype x = m

theorem **mul_single_mul_apply** {n : ℕ} {R : Type} [CommRing R] (A B : Matrix (Fin n) (Fin n) R) (a b i j : Fin n) (c : R) : (A * Matrix.single a b c * B) i j = A i a * c * B b j

theorem **mem_unipSym_of** {k : Type} [CommRing k] {n ρ κ : ℕ} (g : GL (Fin n) k) (h1 : ∀ p q : Fin n, q.val < p.val → (g : Matrix (Fin n) (Fin n) k) p q = 0) (h2 : ∀ p, (g : Matrix (Fin n) (Fin n) k) p p = 1) (h3 : ∀ p q : Fin n, ρ ≤ p.val → p.val < ρ + κ → q.val = ρ + κ → (g : Matrix (Fin n) (Fin n) k) p q = 0) : g ∈ unipSym k n ρ κ

theorem **isScal_of_normalises** (K : Type) [Field K] (O : ValuationSubring K) (k : Type) [Field k] (r : O →+* k) (hr : Function.Surjective r) (n ρ κ : ℕ) (hn : n = 2 * ρ + κ + 1) (hρ : 1 ≤ ρ) (ψ : AddChar k ℂ) (hψ : ψ ≠ 1) (a : Fin n → Fin n → k) (ha : ∀ i j, symEdge ρ κ i j → a i j ≠ 0) (m : GL (Fin n) K) (H1 : ∀ x : GL (Fin n) O, bc r x ∈ unipSym k n ρ κ → ∃ x' : GL (Fin n) O, bc O.subtype x' = m * bc O.subtype x * m⁻¹ ∧ ψ (edgeVal ρ κ a (bc r x')) = ψ (edgeVal ρ κ a (bc r x))) (H2 : ∀ x : GL (Fin n) O, bc r x ∈ unipSym k n ρ κ → ∃ x' : GL (Fin n) O, bc O.subtype x' = m⁻¹ * bc O.subtype x * m) : IsScal O m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsScalarMulIntegralOfNormalizesSymmetricLevelStatement

theorem **_root_.GlobalTheory.T.isScalarMul_integral_of_normalizes_symmetricLevel** : ∀ (K : Type) [Field K] (O : ValuationSubring K) (k : Type) [Field k] (r : O →+* k), Function.Surjective r → ∀ (n ρ κ : ℕ), n = 2 * ρ + κ + 1 → 1 ≤ ρ → ∀ ψ : AddChar k ℂ, ψ ≠ 1 → ∀ a : Fin n → Fin n → k, (∀ i j, symEdge ρ κ i j → a i j ≠ 0) → ∀ m : GL (Fin n) K, (∀ x : GL (Fin n) O, bc r x ∈ unipSym k n ρ κ → ∃ x' : GL (Fin n) O, bc O.subtype x' = m * bc O.subtype x * m⁻¹ ∧ ψ (edgeVal ρ κ a (bc r x')) = ψ (edgeVal ρ κ a (bc r x))) → (∀ x : GL (Fin n) O, bc r x ∈ unipSym k n ρ κ → ∃ x' : GL (Fin n) O, bc O.subtype x' = m⁻¹ * bc O.subtype x * m) → IsScal O m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatAdelicMatrixClosedDerivLinearAdd

theorem **addHom_eq_zero** (F : Type) [DivisionRing F] [CharZero F] [Module F X] (h : ℝ →+ X) (hc : Continuous h) {b : ℝ} (hb : b ≠ 0) (h0 : h b = 0) (t : ℝ) : h t = 0

theorem **hasDerivAt_line** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (G : SchwartzMap E ℂ) (c s : E) : HasDerivAt (fun t : ℝ => G (s + t • c)) ((LineDeriv.lineDerivOp c G : SchwartzMap E ℂ) s) 0

theorem **exists_schwartz_deriv** (F : Type) [DivisionRing F] [CharZero F] [Module F X] (P : X → Prop) (hP : ∀ a b, P a → P b → P (a + b)) (φ : X → ℂ) (hφ : ∀ (k : ℕ) (L : (Fin k → ℝ) →+ X), (Continuous L ∧ ∀ t, P (L t)) → Function.Injective L → ∀ x, ∃ g : SchwartzMap (Fin k → ℝ) ℂ, ∀ t, g t = φ (x + L t)) (M : ℝ →+ X) (hM : Continuous M) (hMP : ∀ t, P (M t)) {k : ℕ} (L : (Fin k → ℝ) →+ X) (hL : Continuous L ∧ ∀ t, P (L t)) (hi : Function.Injective L) (x : X) : ∃ g : SchwartzMap (Fin k → ℝ) ℂ, ∀ s, HasDerivAt (fun t : ℝ => φ (x + L s + M t)) (g s) 0

theorem **isSchwartzBruhat_add** {φ φ' : Mx D m n → ℂ} (h : IsSchwartzBruhat φ) (h' : IsSchwartzBruhat φ') : IsSchwartzBruhat (φ + φ') where

theorem **isSchwartzBruhat_linear_mul** {φ : Mx D m n → ℂ} (h : IsSchwartzBruhat φ) (ℓ : Mx D m n →+ ℂ) (hc : Continuous ℓ) (h0 : ∀ u : Mx D m n, u.map (infPart D) = 0 → ℓ u = 0) : IsSchwartzBruhat fun x => ℓ x * φ x where

theorem **exists_isSchwartzBruhat_deriv** {φ : Mx D m n → ℂ} (h : IsSchwartzBruhat φ) (M : ℝ →+ Mx D m n) (hM : Continuous M) (hM0 : ∀ t, (M t).map (finPart D) = 0) : ∃ φ', IsSchwartzBruhat φ' ∧ ∀ x, HasDerivAt (fun t : ℝ => φ (x + M t)) (φ' x) 0

theorem **isSchwartzBruhat_closed** : ∀ (D : D7) (m n : ℕ) (φ : Mx D m n → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → (∀ M : ℝ →+ Mx D m n, Continuous M → (∀ t, (M t).map (finPart D) = 0) → ∃ φ', GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ' ∧ ∀ x, HasDerivAt (fun t : ℝ => φ (x + M t)) (φ' x) 0) ∧ (∀ ℓ : Mx D m n →+ ℂ, Continuous ℓ → (∀ u : Mx D m n, u.map (infPart D) = 0 → ℓ u = 0) → GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat fun x => ℓ x * φ x) ∧ ∀ φ', GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ' → GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat (φ + φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatAdelicMatrixClosedDerivLinearAddStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhatAdelicMatrix_closed_deriv_linear_add** : ∀ (D : D7) (m n : ℕ) (φ : Mx D m n → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ → (∀ M : ℝ →+ Mx D m n, Continuous M → (∀ t, (M t).map (finPart D) = 0) → ∃ φ', GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ' ∧ ∀ x, HasDerivAt (fun t : ℝ => φ (x + M t)) (φ' x) 0) ∧ (∀ ℓ : Mx D m n →+ ℂ, Continuous ℓ → (∀ u : Mx D m n, u.map (infPart D) = 0 → ℓ u = 0) → GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat fun x => ℓ x * φ x) ∧ ∀ φ', GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ' → GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat (φ + φ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatAdelicMatrixOfMemAdelicSchwartz

theorem **eventually_invariant** {M : Type} [AddCommGroup M] [TopologicalSpace M] [IsTopologicalAddGroup M] {Φ : M → ℂ} (h1 : IsLocallyConstant Φ) (h2 : HasCompactSupport Φ) : ∀ᶠ u in 𝒩 (0 : M), ∀ x, Φ (x + u) = Φ x

theorem **isOpen_periods** {M : Type} [AddCommGroup M] [TopologicalSpace M] [IsTopologicalAddGroup M] {Φ : M → ℂ} (h1 : IsLocallyConstant Φ) (h2 : HasCompactSupport Φ) : IsOpen (periods Φ : Set M)

theorem **zero** : IsSchwartzBruhat (0 : Mx D m n → ℂ) where

theorem **add** {φ ψ : Mx D m n → ℂ} (hφ : IsSchwartzBruhat φ) (hψ : IsSchwartzBruhat ψ) : IsSchwartzBruhat (φ + ψ) where

theorem **smul** (c : ℂ) {φ : Mx D m n → ℂ} (hφ : IsSchwartzBruhat φ) : IsSchwartzBruhat (c • φ) where

theorem **compAt_add** (θ : InfFam D) (τ : D.what₂.Tau) (x y : Mx D m n) : compAt θ τ (x + y) = compAt θ τ x + compAt θ τ y

theorem **norm_compAt_le** (θ : InfFam D) (x : Mx D m n) (τ : D.what₂.Tau) (i : Fin m) (j : Fin n) : ‖compAt θ τ x i j‖ ≤ GlobalTheory.Unf.realSize θ x

theorem **realSize_nonneg** (θ : InfFam D) (x : Mx D m n) : 0 ≤ GlobalTheory.Unf.realSize θ x

theorem **norm_vec_le** (θ : InfFam D) (x : Mx D m n) : ‖(fun τ i j => compAt θ τ x i j : D.what₂.Tau → Fin m → Fin n → ℂ)‖ ≤ GlobalTheory.Unf.realSize θ x

theorem **realSize_le** (θ : InfFam D) (x : Mx D m n) : GlobalTheory.Unf.realSize θ x ≤ (Fintype.card D.what₂.Tau * (m * n) : ℝ) * ‖(fun τ i j => compAt θ τ x i j : D.what₂.Tau → Fin m → Fin n → ℂ)‖

theorem **realSize_zero** (θ : InfFam D) : GlobalTheory.Unf.realSize θ (0 : Mx D m n) = 0

theorem **continuous_finVec** : Continuous (finVec (D

theorem **pure** (θ : InfFam D) (hθ : IsInfFam θ) (f : SchwartzMap (D.what₂.Tau → Fin m → Fin n → ℂ) ℂ) (g : (Fin m → Fin n → Af D.host.F ⊗[D.host.F] D.host.E) → ℂ) (hg : g ∈ GlobalTheory.WeilM.SB (Fin m → Fin n → Af D.host.F ⊗[D.host.F] D.host.E)) : IsSchwartzBruhat fun x : Mx D m n => f (fun τ i j => compAt θ τ x i j) * g (fun i j => finPart D (x i j)) where

theorem **isSchwartzBruhat_of_mem_schwartzA** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ), ∀ φ ∈ GlobalTheory.IndefiniteRowSideFullHecke.ModerateGrowthSchwartz.schwartzA θ m n, GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatAdelicMatrixOfMemAdelicSchwartzStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhatAdelicMatrix_of_mem_adelicSchwartz** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ), ∀ φ ∈ GlobalTheory.IndefiniteRowSideFullHecke.ModerateGrowthSchwartz.schwartzA θ m n, GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatDoubledPairSchrodingerFunction

theorem **continuous_finPart'** : Continuous (finPart D)

theorem **sch_level** : ∃ U : AddSubgroup (Mx D B.rank A.rank), IsOpen (U : Set (Mx D B.rank A.rank)) ∧ ∀ u ∈ U, u.map (infPart D) = 0 → ∀ x, sch ψ A B ν 𝒿 f (x + u) = sch ψ A B ν 𝒿 f x

theorem **sch_supp** : ∃ K : Set (Matrix (Fin B.rank) (Fin A.rank) (Af D.host.F ⊗[D.host.F] D.host.E)), IsCompact K ∧ ∀ x, sch ψ A B ν 𝒿 f x ≠ 0 → x.map (finPart D) ∈ K

theorem **sch_schwartz** (k : ℕ) (L : (Fin k → ℝ) →+ Mx D B.rank A.rank) (hL : GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsRealParam L) (hinj : Function.Injective L) (x : Mx D B.rank A.rank) : ∃ g : SchwartzMap (Fin k → ℝ) ℂ, ∀ t, g t = sch ψ A B ν 𝒿 f (x + L t)

theorem **isSchwartzBruhat_sch** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒿 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒿 → ∀ f, IsThetaFunO ψ (dblSp A) B f → GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat (sch ψ A B ν 𝒿 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatDoubledPairSchrodingerFunctionStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhat_doubledPairSchrodingerFunction** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (A B : HSp D) (ν : Measure (WO A B)) (𝒿 : Set (WO A B)), IsCovolOne (ratL B A) ν 𝒿 → ∀ f, IsThetaFunO ψ (dblSp A) B f → GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat (sch ψ A B ν 𝒿 f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfSourceBoxLattice

theorem **half_add** {m n : ℕ} {e : AE D} {u t : Mx D m n} (hu : IsHalf e u) (ht : IsHalf e t) : IsHalf e (u + t)

theorem **half_sub** {m n : ℕ} {e : AE D} {u t : Mx D m n} (hu : IsHalf e u) (ht : IsHalf e t) : IsHalf e (u - t)

theorem **half_nsmul** {m n : ℕ} {e : AE D} {u : Mx D m n} (hu : IsHalf e u) (N : ℕ) : IsHalf e (N • u)

theorem **half_mul** {m n l : ℕ} {e : AE D} {u : Mx D m n} (hu : IsHalf e u) (x : Mx D n l) : IsHalf e (u * x : Mx D m l)

theorem **half_of_res** {S A L T : HSp D} (o : OrthDec S A L) {e : AE D} {t : WO S T} (h1 : IsHalf e (o.resS T t).1) (h2 : IsHalf e (o.resS T t).2) : IsHalf e t

theorem **zcol_add** (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (y y' : WO L T) : zcol β ζ (y + y') = zcol β ζ y + zcol β ζ y'

theorem **zcol_zero** (β : Frame p T n) (ζ : Fin L.rank → D.host.E) : zcol β ζ 0 = 0

theorem **zcol_nsmul** (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (y : WO L T) (N : ℕ) : zcol β ζ (N • y) = N • zcol β ζ y

theorem **colFun_ne_zero** (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (Λ : Submodule 𝒪[k] (n → k)) (y : WO L T) : colFun β ζ (vecInd Λ) y ≠ 0 ↔ IsHalf p.e y ∧ zcol β ζ y ∈ Λ

theorem **colFun_add** (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (Λ : Submodule 𝒪[k] (n → k)) (y u : WO L T) (hu : IsHalf p.e u) (hu' : zcol β ζ u ∈ Λ) : colFun β ζ (vecInd Λ) (y + u) = colFun β ζ (vecInd Λ) y

theorem **halfInd_ne_zero** {m l : ℕ} (e : AE D) (Λ : AddSubgroup (Mx D m l)) (t : Mx D m l) : halfInd e Λ t ≠ 0 ↔ t ∈ Λ ∧ IsHalf e t

theorem **isSBHalf_boxS_gen** {S A : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (ΛA : AddSubgroup (WO A T)) (ΛS : AddSubgroup (WO S T)) (ΛT : Submodule 𝒪[k] (n → k)) (hf : halfInd p.e ΛS = o.boxS T (halfInd p.e ΛA) (colFun β ζ (vecInd ΛT))) (Φ : WO A T → ℂ) (hΦ : IsSBHalf p.e ΛA Φ) : IsSBHalf p.e ΛS (o.boxS T Φ (colFun β ζ (vecInd ΛT)))

theorem **isSBHalf_boxS** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v)) = o.boxS T (halfInd p.e (latO A T (LA : Set _) (LT : Set _) (· = p.v))) (GlobalTheory.FinPl.ExtZero.colFun β ζ (vecInd ΛT)) → ∀ Φ, IsSBHalf p.e (latO A T (LA : Set _) (LT : Set _) (· = p.v)) Φ → IsSBHalf p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v)) (o.boxS T Φ (GlobalTheory.FinPl.ExtZero.colFun β ζ (vecInd ΛT)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfSourceBoxLatticeStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhatHalf_source_box_lattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), halfInd p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v)) = o.boxS T (halfInd p.e (latO A T (LA : Set _) (LT : Set _) (· = p.v))) (GlobalTheory.FinPl.ExtZero.colFun β ζ (vecInd ΛT)) → ∀ Φ, IsSBHalf p.e (latO A T (LA : Set _) (LT : Set _) (· = p.v)) Φ → IsSBHalf p.e (latO S T (LS : Set _) (LT : Set _) (· = p.v)) (o.boxS T Φ (GlobalTheory.FinPl.ExtZero.colFun β ζ (vecInd ΛT)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfSplitDatumCompletedLattice

theorem **isSBHalf_extFun_latO** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S T : HSp D) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ Ψ ∈ SB (Matrix n n' k), IsSBHalf p.e (latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = p.v)) (extFun α β Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfSplitDatumCompletedLatticeStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhatHalf_split_datum_completed_lattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S T : HSp D) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ Ψ ∈ SB (Matrix n n' k), IsSBHalf p.e (latO S T (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (· = p.v)) (extFun α β Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfSplitPlaceProduct

theorem **eventually_pow_smul_mem** {K M : Type} [Semiring K] [TopologicalSpace K] [AddCommMonoid M] [Module K M] [TopologicalSpace M] [ContinuousSMul K M] {c : K} (hc : Tendsto (fun n : ℕ => c ^ n) atTop (𝓝 0)) {C U : Set M} (hC : IsCompact C) (hU : U ∈ 𝓝 (0 : M)) : ∀ᶠ n in atTop, ∀ x ∈ C, c ^ n • x ∈ U

theorem **exists_nat_pow_tendsto_zero** : ∃ p : ℕ, 0 < p ∧ Tendsto (fun n : ℕ => (p : k) ^ n) atTop (𝓝 0)

theorem **down_eq** (q : GlobalTheory.FinPl.SplitPt D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x : WO S T) : (ofSplit q).down FS FT x = FT.γ * (x : Matrix (Fin T.rank) (Fin S.rank) (AE D)).map q.pr * FS.β

theorem **down_add** (q : GlobalTheory.FinPl.SplitPt D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x y : WO S T) : (ofSplit q).down FS FT (x + y) = (ofSplit q).down FS FT x + (ofSplit q).down FS FT y

theorem **down_nsmul** (q : GlobalTheory.FinPl.SplitPt D k) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (n : ℕ) (x : WO S T) : (ofSplit q).down FS FT (n • x) = n • (ofSplit q).down FS FT x

theorem **exists_den_pr** (q : GlobalTheory.FinPl.SplitPt D k) {ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)} {ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)} (hS : ΛS.FG) (hT : ΛT.FG) : ∃ M : ℕ, 0 < M ∧ ∀ u ∈ latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) (fun _ => True), ∀ i j, (M : k) * q.pr (u i j) ∈ 𝒪[k]

theorem **toEAf_mem_of_pr** (q : GlobalTheory.FinPl.SplitPt D k) {x : AE D} (hx : q.pr x ∈ 𝒪[k]) : toEAf D (q.e * x) ∈ Formalism.L2.Herm.intAf D.host.F D.host.E

theorem **isSBHalf_splitProd** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), Submodule.IsLattice D.host.E ΛS → Submodule.IsLattice D.host.E ΛT → ∀ e : AE D, e * e = e → q.e * e = q.e → ∀ Ω : WO S T → ℂ, IsSBHalf (e − q.e) (latO S T ΛS ΛT fun _ => True) Ω → ∀ Φ : SB (Matrix Y Y' k), IsSBHalf e (latO S T ΛS ΛT fun _ => True) (fun x : WO S T => if IsHalf e x then Φ.1 ((ofSplit q).down FS FT x) * Ω ((1 − q.e) • x) else 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfSplitPlaceProductStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhatHalf_splitPlace_product** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)), Submodule.IsLattice D.host.E ΛS → Submodule.IsLattice D.host.E ΛT → ∀ e : AE D, e * e = e → q.e * e = q.e → ∀ Ω : WO S T → ℂ, IsSBHalf (e − q.e) (latO S T ΛS ΛT fun _ => True) Ω → ∀ Φ : SB (Matrix Y Y' k), IsSBHalf e (latO S T ΛS ΛT fun _ => True) (fun x : WO S T => if IsHalf e x then Φ.1 ((ofSplit q).down FS FT x) * Ω ((1 − q.e) • x) else 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfTargetBoxDualIndicator

theorem **isHalf_add_of** {e : AE D} {a b : ℕ} {x y : Mx D a b} (hx : IsHalf e x) (hy : IsHalf e y) : IsHalf e (x + y)

theorem **isHalf_nsmul_of** {e : AE D} {a b : ℕ} {x : Mx D a b} (hx : IsHalf e x) (m : ℕ) : IsHalf e (m • x)

theorem **isHalf_of_add** {e : AE D} {a b : ℕ} {x y : Mx D a b} (hxy : IsHalf e (x + y)) (hy : IsHalf e y) : IsHalf e x

theorem **isHalf_resT_symm_pair** {S A L : HSp D} (o : OrthDec S A L) (T : HSp D) {e : AE D} {a : WO T A} {b : WO T L} (ha : IsHalf e a) (hb : IsHalf e b) : IsHalf e ((o.resT T).symm (a, b))

theorem **zrow_add'** {T L : HSp D} (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (x y : WO T L) : zrow β ζ (x + y) = zrow β ζ x + zrow β ζ y

theorem **zrow_nsmul'** {T L : HSp D} (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (m : ℕ) (x : WO T L) : zrow β ζ (m • x) = m • zrow β ζ x

theorem **inDual_add'** {Λ : Submodule 𝒪[k] (n → k)} {η ξ : n → k} (h : InDual Λ η) (h' : InDual Λ ξ) : InDual Λ (η + ξ)

theorem **inDual_sub'** {Λ : Submodule 𝒪[k] (n → k)} {η ξ : n → k} (h : InDual Λ η) (h' : InDual Λ ξ) : InDual Λ (η − ξ)

theorem **inDual_nsmul'** {Λ : Submodule 𝒪[k] (n → k)} {η : n → k} (h : InDual Λ η) (m : ℕ) : InDual Λ (m • η)

theorem **rowFun_dualInd_ne_zero** {T L : HSp D} (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (ΛT : Submodule 𝒪[k] (n → k)) {y : WO T L} (h : rowFun β ζ (dualInd ΛT) y ≠ 0) : IsHalf p.e y ∧ InDual ΛT (zrow β ζ y)

theorem **rowFun_dualInd_eq_one** {T L : HSp D} (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (ΛT : Submodule 𝒪[k] (n → k)) {y : WO T L} (hy : IsHalf p.e y) (hd : InDual ΛT (zrow β ζ y)) : rowFun β ζ (dualInd ΛT) y = 1

theorem **rowFun_dualInd_add_nsmul** {T L : HSp D} (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (ΛT : Submodule 𝒪[k] (n → k)) {y : WO T L} (hy : IsHalf p.e y) (hd : InDual ΛT (zrow β ζ y)) (m : ℕ) (x : WO T L) : rowFun β ζ (dualInd ΛT) (x + m • y) = rowFun β ζ (dualInd ΛT) x

theorem **halfInd_eq_one_iff'** {e : AE D} {a b : ℕ} (Λ : AddSubgroup (Mx D a b)) (t : Mx D a b) : halfInd e Λ t = 1 ↔ t ∈ Λ ∧ IsHalf e t

theorem **halfInd_ne_zero_iff'** {e : AE D} {a b : ℕ} (Λ : AddSubgroup (Mx D a b)) (t : Mx D a b) : halfInd e Λ t ≠ 0 ↔ t ∈ Λ ∧ IsHalf e t

theorem **isSBHalf_boxT_rowFun_dualInd** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), halfInd p.e (latO T S (LT : Set _) (LS : Set _) (· = p.v)) = o.boxT T (halfInd p.e (latO T A (LT : Set _) (LA : Set _) (· = p.v))) (GlobalTheory.FinPl.ExtZero.rowFun β ζ (dualInd ΛT)) → ∀ Φ, IsSBHalf p.e (latO T A (LT : Set _) (LA : Set _) (· = p.v)) Φ → IsSBHalf p.e (latO T S (LT : Set _) (LS : Set _) (· = p.v)) (o.boxT T Φ (GlobalTheory.FinPl.ExtZero.rowFun β ζ (dualInd ΛT)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatHalfTargetBoxDualIndicatorStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhatHalf_target_box_dualIndicator** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S A L T : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (β : Frame p T n) (LA : Lat A) (LS : Lat S) (LT : Lat T) (ΛT : Submodule 𝒪[k] (n → k)), halfInd p.e (latO T S (LT : Set _) (LS : Set _) (· = p.v)) = o.boxT T (halfInd p.e (latO T A (LT : Set _) (LA : Set _) (· = p.v))) (GlobalTheory.FinPl.ExtZero.rowFun β ζ (dualInd ΛT)) → ∀ Φ, IsSBHalf p.e (latO T A (LT : Set _) (LA : Set _) (· = p.v)) Φ → IsSBHalf p.e (latO T S (LT : Set _) (LS : Set _) (· = p.v)) (o.boxT T Φ (GlobalTheory.FinPl.ExtZero.rowFun β ζ (dualInd ΛT)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatIffMemAdelicMatrixSchwartzSpace

theorem **GlobalTheory.SchwartzIff.isSchwartzBruhat_iff_mem_schwartzA** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ) (φ : Mx D m n → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ ↔ φ ∈ GlobalTheory.IndefiniteRowSideFullHecke.ModerateGrowthSchwartz.schwartzA θ m n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatIffMemAdelicMatrixSchwartzSpaceStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhat_iff_mem_adelicMatrixSchwartzSpace** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (m n : ℕ) (φ : Mx D m n → ℂ), GlobalTheory.IndefiniteRowSideFullHecke.SchwartzBruhatMixedFamilies.IsSchwartzBruhat φ ↔ φ ∈ GlobalTheory.IndefiniteRowSideFullHecke.ModerateGrowthSchwartz.schwartzA θ m n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatTensorHalfAddPlaceIdempotent

theorem **isSBHalf_tensorAt** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (e : AE D), e * p.e = 0 → (∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ↔ p.up FS FT x ∈ stdHom S T) → (∀ u ∈ stdHom S T, u.map (fun a => p.e * a) ∈ stdHom S T) → ∀ Φ : WO S T → ℂ, IsSBHalf e (stdHom S T) Φ → ∀ Ψ ∈ SB (Matrix Y Y' k), IsSBHalf (e + p.e) (stdHom S T) (p.tensorAt FS FT Φ Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSchwartzBruhatTensorHalfAddPlaceIdempotentStatement

theorem **_root_.GlobalTheory.T.isSchwartzBruhat_tensor_half_add_placeIdempotent** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (e : AE D), e * p.e = 0 → (∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ↔ p.up FS FT x ∈ stdHom S T) → (∀ u ∈ stdHom S T, u.map (fun a => p.e * a) ∈ stdHom S T) → ∀ Φ : WO S T → ℂ, IsSBHalf e (stdHom S T) Φ → ∀ Ψ ∈ SB (Matrix Y Y' k), IsSBHalf (e + p.e) (stdHom S T) (p.tensorAt FS FT Φ Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualAtSumLattice

theorem **cT_fromCols** {n₁ n₂ : Type} (x : Matrix m n₁ A) (y : Matrix m n₂ A) : cT σ (fromCols x y) = fromRows (cT σ x) (cT σ y)

theorem **cT_fromRows** {n₁ n₂ : Type} (x : Matrix n₁ m A) (y : Matrix n₂ m A) : cT σ (fromRows x y) = fromCols (cT σ x) (cT σ y)

theorem **cT_smul_add** {s : A} (h : σ s = s) (x y : Matrix n k A) : cT σ (s • (x + y)) = s • (cT σ x + cT σ y)

theorem **cT_smul_sub** {s : A} (h : σ s = s) (x y : Matrix n k A) : cT σ (s • (x − y)) = s • (cT σ x − cT σ y)

theorem **cols_mul_rows** {s u : A} (h : 2 * (s * u) = 1) {x y : Matrix n k A} {z : Matrix n m A} {p q : Matrix k n A} {r : Matrix m n A} (h1 : x · p + z · r + y · q = 1) (e : n ≃ (k ⊕ m) ⊕ k) : (frameCols s x y z).submatrix id e · (frameRows u p q r).submatrix e id = 1

theorem **rows_form** (φ : A →+* B) {u : A} (h : σ u = u) (hu : φ (2 * (u * u)) = 1) (p q : Matrix k n A) (r : Matrix m n A) (e : n ≃ (k ⊕ m) ⊕ k) : (cT σ ((frameRows u p q r).submatrix e id) · (frameRows u p q r).submatrix e id).map φ = (cT σ p · p + cT σ r · r + cT σ q ·

q).map φ

theorem **cols_gram** (φ : A →+* B) {s : A} (h : σ s = s) (hs : φ (2 * (s * s)) = 1) {x y : Matrix n k A} {z : Matrix n m A} {G : Matrix n n A} (xx : cT σ x · (G · x) = 0) (xy : cT σ x · (G · y) = 1) (yx : cT σ y · (G · x) = 1) (yy : cT σ y · (G · y) = 0) (xz : cT σ x · (G · z) = 0) (yz : cT σ y · (G · z) = 0) (zx : cT σ z · (G · x) = 0) (zy : cT σ z · (G · y) = 0) {ε : m → Bool} (zz : (cT σ z · (G · z)).map φ = diagonal fun i => if ε i = true then 1 else -1) (e : n ≃ (k ⊕ m) ⊕ k) : ∃ ε' : n → Bool, (cT σ ((frameCols s x y z).submatrix id e) · G · (frameCols s x y z).submatrix id e).map φ = diagonal fun i => if ε' i = true then 1 else -1

theorem **compInf_cA** (w : InfinitePlace D.host.F) (x : AE D) : compInf D.host.F D.host.E w (cA D x) = sigR w.Completion D.sp.cm.sig (compInf D.host.F D.host.E w x)

theorem **compInf_incA** (w : InfinitePlace D.host.F) (x : D.host.E) : compInf D.host.F D.host.E w (incA D x) = inclR w.Completion x

theorem **maj_eq** (V : HSp D) (a : GL (Fin V.rank) (AE D)) (w : InfinitePlace D.host.F) : Formalism.UG.cT (sigR w.Completion D.sp.cm.sig) (Formalism.UG.bc (compInf D.host.F D.host.E w) a).1 * JR w.Completion V.gram * (Formalism.UG.bc (compInf D.host.F D.host.E w) a).1 = (adj a.1 * gA V * a.1).map (compInf D.host.F D.host.E w)

theorem **exists_scalars** (D : D7) : ∃ s u : AE D, cA D s = s ∧ cA D u = u ∧ 2 * (s * u) = 1 ∧ (∀ w, compInf D.host.F D.host.E w (2 * (s * s)) = 1) ∧ ∀ w, compInf D.host.F D.host.E w (2 * (u * u)) = 1

theorem **mem_sumLattice** (z : Fin T.rank → D.host.E) : z ∈ sumLattice d L' ↔ (d.qX *ᵥ z ∈ stdLattice D.host.E t ∧ d.q0 *ᵥ z ∈ L' ∧ d.qY *ᵥ z ∈ stdLattice D.host.E t)

theorem **eq_one_of_rat** {k : ℕ} {a : Matrix (Fin k) (Fin T.rank) D.host.E} {b : Matrix (Fin T.rank) (Fin k) D.host.E} (h : rat a * rat b = 1) (x : Fin k → D.host.E) : a *ᵥ (b *ᵥ x) = x

theorem **eq_zero_of_rat** {k l : ℕ} {a : Matrix (Fin k) (Fin T.rank) D.host.E} {b : Matrix (Fin T.rank) (Fin l) D.host.E} (h : rat a * rat b = 0) (x : Fin l → D.host.E) : a *ᵥ (b *ᵥ x) = 0

theorem **e_mem** {a : Fin t → D.host.E} (ha : a ∈ stdLattice D.host.E t) : d.e *ᵥ a ∈ sumLattice d L'

theorem **e'_mem** {a : Fin t → D.host.E} (ha : a ∈ stdLattice D.host.E t) : d.e' *ᵥ a ∈ sumLattice d L'

theorem **i₀_mem** {a : Fin T'.rank → D.host.E} (ha : a ∈ L') : d.i₀ *ᵥ a ∈ sumLattice d L'

theorem **decomp** (z : Fin T.rank → D.host.E) : z = d.e *ᵥ (d.qX *ᵥ z) + d.i₀ *ᵥ (d.q0 *ᵥ z) + d.e' *ᵥ (d.qY *ᵥ z)

theorem **mulVec_mem_span** {k n : ℕ} (A : Matrix (Fin n) (Fin k) D.host.E) {N : Submodule (𝒪 D.host.E) (Fin k → D.host.E)} (hN : Submodule.IsLattice D.host.E N) {L : Submodule (𝒪 D.host.E) (Fin n → D.host.E)} (h : ∀ a ∈ N, A *ᵥ a ∈ L) (z : Fin k → D.host.E) : A *ᵥ z ∈ Submodule.span D.host.E (L : Set (Fin n → D.host.E))

theorem **isLattice_sumLattice** (hL : Submodule.IsLattice D.host.E L') : Submodule.IsLattice D.host.E (sumLattice d L')

theorem **hE_eq** {n : ℕ} (J : Matrix (Fin n) (Fin n) D.host.E) (x y : Fin n → D.host.E) : hE D.sp.cm.sig J x y = star x ⬝ᵥ (J *ᵥ y)

theorem **hE_mulVec** {n k : ℕ} (J : Matrix (Fin n) (Fin n) D.host.E) (A : Matrix (Fin n) (Fin k) D.host.E) (x : Fin k → D.host.E) (y : Fin n → D.host.E) : hE D.sp.cm.sig J (A *ᵥ x) y = star x ⬝ᵥ ((Aᴴ * J) *ᵥ y)

theorem **hE_i₀** (x : Fin T'.rank → D.host.E) (y : Fin T.rank → D.host.E) : hE D.sp.cm.sig T.gram (d.i₀ *ᵥ x) y = hE D.sp.cm.sig T'.gram x (d.q0 *ᵥ y)

theorem **mem_std_iff** (y : Fin t → D.host.E) : y ∈ stdLattice D.host.E t ↔ ∀ x ∈ stdLattice D.host.E t, star x ⬝ᵥ y ∈ (algebraMap (𝒪 D.host.E) D.host.E).range

theorem **isSelfDualAt_sumLattice** (hL : Submodule.IsLattice D.host.E L') (hd : ∀ v, IsSelfDualAt D.sp.cm.sig T'.gram L' v) (v : HeightOneSpectrum (𝒪 D.host.F)) : IsSelfDualAt D.sp.cm.sig T.gram (sumLattice d L') v

theorem **isMaj_iff** (V : HSp D) (a : GL (Fin V.rank) (AE D)) : IsMaj V a ↔ ∀ w, ∃ ε : Fin V.rank → Bool, (adj a.1 · gA V · a.1).map (compInf D.host.F D.host.E w) = Matrix.diagonal fun i => if ε i = true then 1 else -1

theorem **sum_frame** (c : GL (Fin T'.rank) (AE D)) : d.jX · d.pX + (d.j0 · c.1) · ((c⁻¹).1 · d.p0) + d.jY · d.pY = 1

theorem **gram_Xc** : adj d.jX · (gA T · (d.j0 · c)) = 0

theorem **gram_Yc** : adj d.jY · (gA T · (d.j0 · c)) = 0

theorem **gram_cX** : adj (d.j0 · c) · (gA T · d.jX) = 0

theorem **gram_cY** : adj (d.j0 · c) · (gA T · d.jY) = 0

theorem **gram_cc** : adj c · gA T' · c = adj (d.j0 · c) · (gA T · (d.j0 · c))

theorem **adjoint_form_p_zero** : adj d.p0 · ((adj c · c) · d.p0) = adj (c · d.p0) · (c · d.p0)

theorem **exists_frame** (l' : Lvl T') (hm : IsMaj T' l'.c) : ∃ c : GL (Fin T.rank) (AE D), IsMaj T c ∧ ∀ w, (adj (c⁻¹).1 · (c⁻¹).1 : Mx D T.rank T.rank).map (compInf D.host.F D.host.E w) = (adj d.pX · d.pX + adj d.p0 · (l'.form · d.p0) + adj d.pY · d.pY : Mx D T.rank T.rank).map (compInf D.host.F D.host.E w)

theorem **exists_good** (V : HSp D) : ∃ l : Lvl V, l.Good

theorem **exists_good_compat** {T T' : HSp D} {t : ℕ} (d : HypDec T T' t) (l' : Lvl T') (h : l'.Good) : ∃ l : Lvl T, l.Good ∧ Compat d l l'

theorem **good_iff_isLevel** {V : HSp D} (l : Lvl V) : l.Good ↔ GlobalTheory.EisCont.IsLevel V l.L l.c

theorem **K_eq_lev** {V : HSp D} (l : Lvl V) : l.K = GlobalTheory.EisCont.lev V l.L l.c

theorem **levels_exist** (D : D7) : (∀ V : HSp D, ∃ l : Lvl V, l.Good) ∧ (∀ (T T' : HSp D) (t : ℕ) (d : HypDec T T' t) (l' : Lvl T'), l'.Good → ∃ l : Lvl T, l.Good ∧ Compat d l l') ∧ ∀ (V : HSp D) (l : Lvl V), (l.Good ↔ GlobalTheory.EisCont.IsLevel V l.L l.c) ∧ l.K = GlobalTheory.EisCont.lev V l.L l.c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualCFrameLat

theorem **isSelfDualC_frameLat** {D : D7} (V : HSp D) : IsSelfDualC V (frameLat V)

theorem **enorm_dblIntegrand** {D : D7} {V : HSp D} (μ : (AE D)ˣ →* Circle) (E : GA D (dblSp V) → ℂ) (f₁ f₂ : GA D V → ℂ) (x : GA D V × GA D V) : ‖dblIntegrand V μ E f₁ f₂ x‖ₑ = ‖E (io V x)‖ₑ * (‖f₁ x.1‖ₑ * ‖f₂ x.2‖ₑ)

theorem **dblConv_of_growth** {D : D7} {V : HSp D} {μ : (AE D)ˣ →* Circle} (hμ : IsHeckeChar D μ) {m : Measure (GA D V)} (hm : m.IsHaarMeasure) {f₁ f₂ : GA D V → ℂ} (h₁ : GlobalTheory.Grow.RapidDec V f₁) (h₂ : GlobalTheory.Grow.RapidDec V f₂) {E : GA D (dblSp V) → ℂ} (hE : LeftInv (GQ (dblSp V)) E) (hc : Continuous E) {C : ℝ} {N : ℕ} (hb : ∀ g, ‖E g‖ ≤ C * GlobalTheory.Grow.height (dblSp V) g ^ N) : DblConv V μ m E f₁ f₂

theorem **absConv_unfolded** : ∀ (D : D7) (V : HSp D) (a : GL (Fin V.rank) (AE D)), IsMaj V a → ∀ (μ : (AE D)ˣ →* Circle), IsHeckeChar D μ → ∀ s : ℂ, (V.rank : ℝ) / 2 < s.re → ∀ φ ∈ degPS V a μ s, GlobalTheory.Grow.ModGrowth (dblSp V) (GlobalTheory.Eis.eis V

(GlobalTheory.Eis.sph V a (s.re : ℂ))) → ∀ m : Measure (GA D V), m.IsHaarMeasure → ∀ f₁ f₂ : GA D V → ℂ, GlobalTheory.Grow.RapidDec V f₁ → GlobalTheory.Grow.RapidDec V f₂ → (∀ 𝓕 : Set (GA D V), IsFundamentalDomain (GQ V) 𝓕 m → ∑' q : GlobalTheory.Eis.SiegelCosets V, ∫⁻ x in 𝓕 ×ˢ 𝓕, ‖GlobalTheory.Eis.dblIntegrand V μ (fun x => φ (((q.out : GQ (dblSp V)) : GA D (dblSp V)) * x)) f₁ f₂ x‖ₑ ∂m.prod m ≠ ⊤) ∧ GlobalTheory.Eis.DblConv V μ m (GlobalTheory.Eis.eis V φ) f₁ f₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualCOfGramOne

theorem **isSelfDualC_of_gram_one** {S : HSp D} (hG : S.gram = 1) (L : Lat S) (hL : ∀ z, z ∈ L ↔ z ∈ intVec D S.rank) : IsSelfDualC S L

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualLatticeAtPlaceComapUnitarySupported

theorem **exists_nat_mul_isIntegral'** (e : D.host.E) : ∃ n : ℕ, 0 < n ∧ IsIntegral ℤ ((n : D.host.E) * e)

theorem **mul_mem_intAE'** {x y : Af D.host.F ⊗[D.host.F] D.host.E} (hx : x ∈ intAE D) (hy : y ∈ intAE D) : x * y ∈ intAE D

theorem **natCast_mem_intAE** (n : ℕ) : (n : Af D.host.F ⊗[D.host.F] D.host.E) ∈ intAE D

theorem **exists_natCast_mul_mem_intAE** (z : Af D.host.F ⊗[D.host.F] D.host.E) : ∃ n : ℕ, 0 < n ∧ (n : Af D.host.F ⊗[D.host.F] D.host.E) * z ∈ intAE D

theorem **exists_natCast_mul_mem_intAt** {P : Pl D → Prop} {a : AE D} (ha : a ∈ idAt D P) : ∃ n : ℕ, 0 < n ∧ (n : AE D) * a ∈ intAt D P

theorem **intAt_add** {P : Pl D → Prop} {a b : AE D} (ha : a ∈ intAt D P) (hb : b ∈ intAt D P) : a + b ∈ intAt D P

theorem **intAt_mul** {P : Pl D → Prop} {a b : AE D} (ha : a ∈ intAt D P) (hb : b ∈ intAt D P) : a * b ∈ intAt D P

theorem **intAt_natCast_mul** {P : Pl D → Prop} {a : AE D} (n : ℕ) (ha : a ∈ intAt D P) : (n : AE D) * a ∈ intAt D P

theorem **intAt_sum** {P : Pl D → Prop} {ι : Type} (s : Finset ι) (f : ι → AE D) (h : ∀ i ∈ s, f i ∈ intAt D P) : ∑ i ∈ s, f i ∈ intAt D P

theorem **cA_mulVec** {n : ℕ} (M : Mx D n n) (y : Fin n → AE D) : (fun i => cA D ((M *ᵥ y) i)) = Matrix.vecMul (fun i => cA D (y i)) (adj M)

theorem **hermAt_eq** (S : HSp D) (x y : Fin S.rank → AE D) : hermAt S x y = Matrix.vecMul (fun i => cA D (y i)) (gA S) ⬝ᵥ x

theorem **hermAt_mt** (S : HSp D) (g : GA D S) (x y : Fin S.rank → AE D) : hermAt S (mt S g *ᵥ x) (mt S g *ᵥ y) = hermAt S x y

theorem **mulVec_eq_add** {R : Type} [CommRing R] {n : ℕ} (M : Matrix (Fin n) (Fin n) R) (y : Fin n → R) (i : Fin n) : (M *ᵥ y) i = y i + ∑ j, (M i j - (1 : Matrix (Fin n) (Fin n) R) i j) * y j

theorem **eq_mulVec_sub** {R : Type} [CommRing R] {n : ℕ} (M : Matrix (Fin n) (Fin n) R) (y : Fin n → R) (i : Fin n) : y i = (M *ᵥ y) i - ∑ j, (M i j - (1 : Matrix (Fin n) (Fin n) R) i j) * y j

theorem **smul_mulVec_apply** {R : Type} [CommRing R] {n : ℕ} (M : Matrix (Fin n) (Fin n) R) (y : Fin n → R) (c : R) (i : Fin n) : (c • (M *ᵥ y)) i = c * y i + ∑ j, (c * (M i j - (1 : Matrix (Fin n) (Fin n) R) i j)) * y j

theorem **mulVec_natCast_smul** {R : Type} [CommRing R] {n : ℕ} (M : Matrix (Fin n) (Fin n) R) (y : Fin n → R) (m K : ℕ) : M *ᵥ (((m * K : ℕ) : R) • y) = (m : R) • ((K : R) • (M *ᵥ y))

theorem **mulVec_mulVec_of_mul_eq_one** {R : Type} [CommRing R] {n : ℕ} {M N : Matrix (Fin n) (Fin n) R} (h : M * N = 1) (x : Fin n → R) : M *ᵥ (N *ᵥ x) = x

theorem **exists_common_natCast_mul_mem_intAt** {P : Pl D → Prop} {ι : Type} [Fintype ι] (a : ι → AE D) (ha : ∀ i, a i ∈ idAt D P) : ∃ K : ℕ, 0 < K ∧ ∀ i, (K : AE D) * a i ∈ intAt D P

theorem **isSDAt_comap_of** (S : HSp D) (v : Pl D) (M N : Mx D S.rank S.rank) (hMN : M * N = 1) (hM1 : ∀ i j, M i j - (1 : Mx D S.rank S.rank) i j ∈ idAt D (· = v)) (hherm : ∀ x y, hermAt S (M *ᵥ x) (M *ᵥ y) = hermAt S x y) (Λ : AddSubgroup (Fin S.rank → AE D)) (hΛ : IsSDAt S v Λ) : IsSDAt S v (Λ.comap (Matrix.mulVecLin M).toAddMonoidHom)

theorem **isSDAt_comap** (D : D7) (S : HSp D) (v : Pl D) (g : GA D S) (hg : g ∈ grpAt S (· = v)) (Λ : AddSubgroup (Fin S.rank → AE D)) (hΛ : IsSDAt S v Λ) : IsSDAt S v (Λ.comap (Matrix.mulVecLin (mt S g)).toAddMonoidHom)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualLatticeAtPlaceComapUnitarySupportedStatement

theorem **_root_.GlobalTheory.T.isSelfDualLatticeAtPlace_comap_unitarySupported** : ∀ (D : D7) (S : HSp D) (v : Pl D) (g : GA D S), g ∈ grpAt S (· = v) → ∀ Λ : AddSubgroup (Fin S.rank → AE D), IsSDAt S v Λ → IsSDAt S v (Λ.comap (Matrix.mulVecLin (mt S g)).toAddMonoidHom)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualLatticeAtPlaceLatticeCompletion

theorem **exists_idem** (v : Pl D) : ∃ ε : AE D, cA D ε = ε ∧ ε ∈ intAt D (· = v) ∧ ∀ a, a ∈ idAt D (· = v) → ε * a = a

theorem **hermAt_mul_inc** (y : Fin S.rank → AE D) (a : AE D) (s : Fin S.rank → D.host.E) : hermAt S y (fun i => a * incA D (s i)) = cA D a * pairing S s y

theorem **hermAt_sum_right** {m : ℕ} (y : Fin S.rank → AE D) (x : Fin m → Fin S.rank → AE D) : hermAt S y (fun i => ∑ l, x l i) = ∑ l, hermAt S y (x l)

theorem **hermAt_mem_idAt** {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : ∀ i, y i ∈ idAt D P) (x : Fin S.rank → AE D) : hermAt S y x ∈ idAt D P

theorem **exists_mul_mem** (hL : IsSelfDualC S L) (v : Pl D) : ∃ m : ℕ, 0 < m ∧ ∀ y : Fin S.rank → AE D, (∀ i, y i ∈ intAt D (· = v)) → (m : AE D) • y ∈ latV S L (· = v)

theorem **isSDAt_latV** : ∀ (D : D7) (S : HSp D) (L : Lat S) (v : Pl D), IsSelfDualC S L → IsSDAt S v (latV S L (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsSelfDualLatticeAtPlaceLatticeCompletionStatement

theorem **_root_.GlobalTheory.T.isSelfDualLatticeAtPlace_latticeCompletion** : ∀ (D : D7) (S : HSp D) (L : Lat S) (v : Pl D), IsSelfDualC S L → IsSDAt S v (latV S L (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsStepVecOfIff

theorem **isMoved_move** {p : SplitPt D k} {S : HSp D} (β : Frame p S n) (g : GA D S) : IsMoved g β (move β g)

theorem **isMoved_back** {p : SplitPt D k} {S : HSp D} (β : Frame p S n) (g : GA D S) : IsMoved g (move β g⁻¹) β

theorem **besCols_toFin** {p : SplitPt D k} {ρ : ℕ} (B : BesLev D ρ) (s : k) (α : GlobalTheory.LocMod.Frame k B.A.rank n) (β : GlobalTheory.LocMod.Frame k B.V.rank (n ⊕ Fin c)) : BesCols B (ofSplit p).xi s (toFin p _ α) (toFin p _ β) ↔ (ofSplit p).IsBesFrame B s α β

theorem **jacCols_toFin** {p : SplitPt D k} {ρ : ℕ} (J : JacLev D ρ) (s h : k) (α : GlobalTheory.LocMod.Frame k J.W₀.rank n) (β : GlobalTheory.LocMod.Frame k J.W.rank (n ⊕ Fin c)) : JacCols J (ofSplit p).xi s (h / (ofSplit p).xi) (toFin p _ α) (toFin p _ β) ↔

(ofSplit p).IsJacFrame J s h α β

theorem **exists_jacCols** {p : SplitPt D k} {ρ : ℕ} (J : JacLev D ρ) {θ s a : k} (hθ : θ ≠ 0) (hs : s ≠ 0) (ha : a ≠ 0) (α : Frame p J.W₀ n) (hc : c = 2 * ρ) : ∃ β : Frame p J.W (n ⊕ Fin c), JacCols J θ s a α β

theorem **exists_besCols** {p : SplitPt D k} {ρ : ℕ} (B : BesLev D ρ) {θ s : k} (hθ : θ ≠ 0) (hs : s ≠ 0) (α : Frame p B.A n) (hc : c = 2 * ρ + 1) : ∃ β : Frame p B.V (n ⊕ Fin c), BesCols B θ s α β

theorem **isStepVecOf_iff** (c : GlobalTheory.ChainOnto.Com D) (s s' : GlobalTheory.ChainOnto.PiLocalization₂ D) (j : ℕ) (ξ f₀ f) : IsStepVecOf c.ψ c.θ s.V s'.V (cart s.M) (cart s'.M) s.Λ s'.Λ s.β s'.β s.L (c.chi (j + 1)) ∅ ξ f₀ f ↔ IsVacO c.ψ c.θ s.V s'.V (cart s.M) (cart s'.M) s.Λ s'.Λ f₀ ∧ ∃ e : AE D, IsIdemSum e (s.L.Pg ∪ ∅) ∧ IsThetaAt c.ψ (symplO s.V s'.V) (heisO c.ψ s.V s'.V) e (latO s.V s'.V s.Λ s'.Λ fun _ => True) 1 f₀ (stepDat c s s' j ξ e) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsTameFinitePlaceHermitianAll

theorem **extUnramAt** (D : D7) (v : Pl D) : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.ExtUnramAt D v

theorem **split_over_two** (D : D7) (v : Pl D) (h2 : (2 : 𝒪 D.host.F) ∈ v.asIdeal) : ∃ e ε, IsSplitIdem D e ∧ IsPlaceIdem D v ε ∧ e + cA D e = algebraMap (AF D.host.F) (AE D) ε

theorem **tame** : ∀ (D : D7) (v : Pl D), Tame D v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsTameFinitePlaceHermitianAllStatement

theorem **_root_.GlobalTheory.T.isTameFinitePlaceHermitian_all** : ∀ (D : D7) (v : Pl D), Tame D v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaAtMoveLat

theorem **mt_mul₀** (V : HSp D) (g h : GA D V) : mt V (g * h) = mt V g * mt V h

theorem **mt_one₀** (V : HSp D) : mt V 1 = 1

theorem **mt_mul_inv** (V : HSp D) (g : GA D V) (y : Fin V.rank → AE D) : mt V g *ᵥ (mt V g⁻¹ *ᵥ y) = y

theorem **mulVec_mem_idAt** {n : ℕ} (A : Matrix (Fin n) (Fin n) (AE D)) {P : Pl D → Prop} {y : Fin n → AE D} (hy : ∀ i, y i ∈ idAt D P) (i : Fin n) : (A *ᵥ y) i ∈ idAt D P

theorem **moves_inv** {S : HSp D} {g : GA D S} {L M : Lat S} {P : Pl D → Prop} (h : Moves S g L M P) : Moves S g⁻¹ M L P

theorem **mulVec_mem_latA** {S : HSp D} {g : GA D S} {L M : Lat S} {P : Pl D → Prop} (h : Moves S g L M P) {z : Fin S.rank → AE D} (hz : z ∈ latA S L P) : mt S g *ᵥ z ∈ latA S M P

theorem **actO_eq** (S T : HSp D) (g : GA D S) (h : GA D T) (u : WO S T) : actO S T g h u = HMul.hMul (HMul.hMul (mt T h) (u : Matrix (Fin T.rank) (Fin S.rank) (AE D))) (mt S g⁻¹)

theorem **actO_mem_latO** {S T : HSp D} {Λ M : Lat S} {Λ' M' : Lat T} (hΛ : IsSelfDualC S Λ) (hM : IsSelfDualC S M) (hΛ' : IsSelfDualC T Λ') (hM' : IsSelfDualC T M') {P : Pl D → Prop} {g : GA D S} {h : GA D T} (hg : Moves S g Λ M P) (hh : Moves T h Λ' M' P) {u : WO S T} (hu : u ∈ latO S T (Λ : Set _) (Λ' : Set _) P) : actO S T g h u ∈ latO S T (M : Set _) (M' : Set _) P

theorem **mem_latO_iff_actO** {S T : HSp D} {Λ M : Lat S} {Λ' M' : Lat T} (hΛ : IsSelfDualC S Λ) (hM : IsSelfDualC S M) (hΛ' : IsSelfDualC T Λ') (hM' : IsSelfDualC T M') {P : Pl D → Prop} {g : GA D S} {h : GA D T} (hg : Moves S g Λ M P) (hh : Moves T h Λ' M' P) (u : WO S T) : u ∈ latO S T (Λ : Set _) (Λ' : Set _) P ↔ actO S T g h u ∈ latO S T (M : Set _) (M' : Set _) P

theorem **isThetaAt_moveLat** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ (Λ M : Lat S) (Λ' M' : Lat T), IsSelfDualC S Λ → IsSelfDualC S M → IsSelfDualC T Λ' → IsSelfDualC T M' → ∀ b ∈ GfC S, ∀ b' ∈ GfC T, Moves S b Λ M (fun _ => True) → Moves T b' Λ' M' (fun _ => True) → ∀ e Φ f₀ f, e * e = e → infPart D e = 0 → IsThetaFunO ψ S T f₀ → IsThetaFunO ψ S T f → IsThetaAt ψ (symplO S T) (heisO ψ S T) e (latO S T (Λ : Set _) (Λ' : Set _) fun _ => True) 1 f₀ Φ f → IsThetaAt ψ (symplO S T) (heisO ψ S T) e (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 (om (b, b') f₀) (fun x => Φ (actO S T b⁻¹ b'⁻¹ x)) (om (b, b') f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOConstMul

theorem **isThetaFunO_const_mul** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (fun w => c * f w) where

theorem **isThetaFunO_kap** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ · (L.gram · τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (S : HSp D) {ψ : AddChar (AF D.host.F) ℂ} {f : WO L S → ℂ} (hf : IsThetaFunO ψ L S f) : IsThetaFunO ψ S L'' (fun w => f (kap τ S w))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOSourceSlice

theorem **isThetaFunO_slice** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (f : WO S T → ℂ) (y : WO L T), IsThetaFunO ψ S T f → IsThetaFunO ψ A T (fun w => f ((o.resS T).symm (w, y)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOSourceSliceStatement

theorem **_root_.GlobalTheory.T.isThetaFunO_source_slice** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (f : WO S T → ℂ) (y : WO L T), IsThetaFunO ψ S T f → IsThetaFunO ψ A T (fun w => f ((o.resS T).symm (w, y)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOSplitPlacesSchrodingerModelSum

theorem **isThetaFunO_sum_placeModel** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : Lat S) (M' : Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), Vac ψ S T M M' (fun _ => True) f₀ → IsPlaceModel ψ (symplO S T) (heisO ψ S T) (∑ i, (L i).q.e) (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 f₀ T' → ∀ Φ, IsSBHalf (∑ i, (L i).q.e) (latO S T (M : Set _) (M' : Set _) fun _ => True) Φ → IsThetaFunO ψ S T (T' Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOSplitPlacesSchrodingerModelSumStatement

theorem **_root_.GlobalTheory.T.isThetaFunO_splitPlaces_schrodingerModel_sum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : Lat S) (M' : Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ (f₀ : WO S T → ℂ) (T' : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), Vac ψ S T M M' (fun _ => True) f₀ → IsPlaceModel ψ (symplO S T) (heisO ψ S T) (∑ i, (L i).q.e) (latO S T (M : Set _) (M' : Set _) fun _ => True) 1 f₀ T' → ∀ Φ, IsSBHalf (∑ i, (L i).q.e) (latO S T (M : Set _) (M' : Set _) fun _ => True) Φ → IsThetaFunO ψ S T (T' Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOTargetSliceOrthogonalSum

theorem **isThetaFunO_sliceT** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (f : WO T S → ℂ) (y : WO T L), IsThetaFunO ψ T S f → IsThetaFunO ψ T A (fun w => f ((o.resT T).symm (w, y)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunOTargetSliceOrthogonalSumStatement

theorem **_root_.GlobalTheory.T.isThetaFunO_target_slice_orthogonal_sum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S A L T : HSp D) (o : OrthDec S A L) (f : WO T S → ℂ) (y : WO T L), IsThetaFunO ψ T S f → IsThetaFunO ψ T A (fun w => f ((o.resT T).symm (w, y)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunSplitPlaceSchroedingerModel

theorem **infPart_of_isSplitIdem** {D : D7} {e : AE D} (he : IsSplitIdem D e) : infPart D e = 0

theorem **isThetaFunO_placeModel** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ΛS : Lat S) (ΛT : Lat T) (e : AE D) (f₀ : WO S T → ℂ) (Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)) (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) (he : IsSplitIdem D e) (hf : IsThetaFunO ψ S T f₀) (hTm : IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) 1 f₀ Tm) (Φ : WO S T → ℂ) (hΦ : IsSBHalf e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) Φ) : IsThetaFunO ψ S T (Tm Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunSplitPlaceSchroedingerModelStatement

theorem **_root_.GlobalTheory.T.isThetaFun_splitPlaceSchroedingerModel** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ΛS : Lat S) (ΛT : Lat T) (e : AE D) (f₀ : WO S T → ℂ) (Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)), IsStdChar D ψ (-1) → IsSelfDualC S ΛS → IsSelfDualC T ΛT → IsSplitIdem D e → IsThetaFunO ψ S T f₀ → IsPlaceModel ψ (symplO S T) (heisO ψ S T) e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) 1 f₀ Tm → ∀ Φ : WO S T → ℂ, IsSBHalf e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) Φ → IsThetaFunO ψ S T (Tm Φ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunSuperpositionTargetOrthSummand

theorem **symplOH_zero_left** {D : D7} {S T : HSp D} (y : WO S T) : symplOH S T 0 y = 0

theorem **symplOH_zero_right** {D : D7} {S T : HSp D} (x : WO S T) : symplOH S T x 0 = 0

theorem **psi_neg_mul** {D : D7} (ψ : AddChar (AF D.host.F) ℂ) (s : AF D.host.F) : ψ (-s) * ψ s = 1

theorem **symplOH_resT_symm** {D : D7} {T A B : HSp D} (o : OrthDec T A B) (S : HSp D) (u u' : WO S A) (v v' : WO S B) : symplOH S T ((o.resT S).symm (u, v)) ((o.resT S).symm (u', v')) = symplOH S A u u' + symplOH S B v v'

theorem **superpos_target** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (T A B S : HSp D) (o : OrthDec T A B) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ (Ω : Type) [MeasurableSpace Ω] (μ : Measure Ω) (k : Ω → WO S B) (a : Ω → ℂ), Measurable k → Integrable a μ → IsThetaFunO ψ S A fun u => ∫ w, a w * F ((o.resT S).symm (u, k w)) ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunSuperpositionTargetOrthSummandStatement

theorem **_root_.GlobalTheory.T.isThetaFun_superposition_target_orthSummand** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (T A B S : HSp D) (o : OrthDec T A B) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ (Ω : Type) [MeasurableSpace Ω] (μ : Measure Ω) (k : Ω → WO S B) (a : Ω → ℂ), Measurable k → Integrable a μ → IsThetaFunO ψ S A fun u => ∫ w, a w * F ((o.resT S).symm (u, k w)) ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunctionCompSkewAdjointSimilitude

theorem **isThetaFunO_comp_kappa** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (L S L'' : HSp D) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∀ f, IsThetaFunO ψ L S f → IsThetaFunO ψ S L'' (fun w => f (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj L S ((rat τ : Mx D L.rank L''.rank) * w)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunctionCompSkewAdjointSimilitudeStatement

theorem **_root_.GlobalTheory.T.isThetaFunction_comp_skewAdjointSimilitude** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (L S L'' : HSp D) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∀ f, IsThetaFunO ψ L S f → IsThetaFunO ψ S L'' (fun w => f (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj L S ((rat τ : Mx D L.rank L''.rank) * w)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunctionDoublingSeesawOfStandardChar

theorem **continuous_fstH** (A B : HSp D) : Continuous (fstH A B)

theorem **continuous_sndH** (A B : HSp D) : Continuous (sndH A B)

theorem **map_fstH** {R : Type} [CommRing R] (φ : AE D →+* R) (A B : HSp D) {u : WO (dblSp A) B} (h : u.map φ = 0) : (fstH A B u).map φ = 0

theorem **map_sndH** {R : Type} [CommRing R] (φ : AE D →+* R) (A B : HSp D) {u : WO (dblSp A) B} (h : u.map φ = 0) : (sndH A B u).map φ = 0

theorem **hO_dbl** (A B : HSp D) (x y : WO (dblSp A) B) : hO (dblSp A) B x y = hO A B (fstH A B x) (fstH A B y) - hO A B (sndH A B x) (sndH A B y)

theorem **symplOH_dbl** (A B : HSp D) (x y : WO (dblSp A) B) : symplOH (dblSp A) B x y = symplOH A B (fstH A B x) (fstH A B y) - symplOH A B (sndH A B x) (sndH A B y)

theorem **conj_char** {G : Type} [AddCommGroup G] (ψ : AddChar G ℂ) (h : ∀ x, ‖ψ x‖ = 1) (x : G) : conj (ψ x) = ψ (-x)

theorem **isThetaFunO_delta** {ψ : AddChar (AF D.host.F) ℂ} (hn : ∀ x, ‖ψ x‖ = 1) {A B : HSp D} {f₁ f₂ : WO A B → ℂ} (h₁ : IsThetaFunO ψ A B f₁) (h₂ : IsThetaFunO ψ A B f₂) : IsThetaFunO ψ (dblSp A) B (delta A B f₁ f₂)

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D), IsStdChar D ψ (-1) → ∀ f₁ f₂, IsThetaFunO ψ A B f₁ → IsThetaFunO ψ A B f₂ → IsThetaFunO ψ (dblSp A) B (delta A B f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunctionDoublingSeesawOfStandardCharStatement

theorem **_root_.GlobalTheory.T.isThetaFunction_doublingSeesaw_of_standard_char** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (A B : HSp D), IsStdChar D ψ (-1) → ∀ f₁ f₂, IsThetaFunO ψ A B f₁ → IsThetaFunO ψ A B f₂ → IsThetaFunO ψ (dblSp A) B (delta A B f₁ f₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunctionSuperpositionOrthogonalSlices

theorem **symplOH_zero_right** {S T : HSp D} (x : WO S T) : symplOH S T x 0 = 0

theorem **symplOH_zero_left** {S T : HSp D} (x : WO S T) : symplOH S T 0 x = 0

theorem **psi_neg_mul** (ψ : AddChar (AF D.host.F) ℂ) (s : AF D.host.F) : ψ (-s) * ψ s = 1

theorem **symm_add** (r u : WO A T) (y : WO B T) : (o.resS T).symm (u + r, y) = (o.resS T).symm (u, y) + (o.resS T).symm (r, 0)

theorem **sympl_ext_right** (r u : WO A T) (y : WO B T) : symplOH S T ((o.resS T).symm (u, y)) ((o.resS T).symm (r, 0)) = symplOH A T u r

theorem **sympl_ext_left** (r u : WO A T) (y : WO B T) : symplOH S T ((o.resS T).symm (r, 0)) ((o.resS T).symm (u, y)) = symplOH A T r u

theorem **matrixExtendByZero_apply** (r : WO A T) : ext0 o T r = (o.resS T).symm (r, 0)

theorem **continuous_wittExtendByZero** : Continuous (ext0 o T)

theorem **matrixExtendByZero_map** {R : Type} [Semiring R] (f : AE D →+* R) (r : WO A T) (h : r.map f = 0) : (ext0 o T r).map f = 0

theorem **homExtensionByZeroOrthDecomp_rational** (γ : Matrix (Fin T.rank) (Fin A.rank) D.host.E) : ext0 o T (ratW T A γ) = ratW T S (γ ∗ o.qA)

theorem **superpos_isTheta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S A B T : HSp D) (o : OrthDec S A B) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ (Ω : Type) [MeasurableSpace Ω] (μ : Measure Ω) (k : Ω → WO B T) (a : Ω → ℂ), Measurable k → Integrable a μ → IsThetaFunO ψ A T fun u => ∫ w, a w * F ((o.resS T).symm (u, k w)) ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsThetaFunctionSuperpositionOrthogonalSlicesStatement

theorem **_root_.GlobalTheory.T.isThetaFunction_superposition_orthogonalSlices** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S A B T : HSp D) (o : OrthDec S A B) (F : WO S T → ℂ), IsThetaFunO ψ S T F → ∀ (Ω : Type) [MeasurableSpace Ω] (μ : Measure Ω) (k : Ω → WO B T) (a : Ω → ℂ), Measurable k → Integrable a μ → IsThetaFunO ψ A T fun u => ∫ w, a w * F ((o.resS T).symm (u, k w)) ∂μ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsTypePlacesOfLevCh

theorem **isTypePlaces_of_levCh** {L : LevCh D Unit} (hL : L.Is ()) : GlobalTheory.Core.Mi.IsTypePlaces L.Pg

theorem **com_is_of_levCh** {L : LevCh D Unit} (hL : L.Is ()) (c : GlobalTheory.ChainOnto.Com D) (hθ : IsInfFam c.θ) (hψ : IsStdChar D c.ψ (-1)) (hP : c.Pg = L.Pg) (hμ : ∀ j : Fin D.R.m, ChrEq (c.μ ((j : ℕ) + 1)) (L.C.mu0 j)) (hη : ChrEq c.η L.C.eta) (hχ : ∀ j : Fin D.R.m, ∀ PP ∈ c.Pg, IsRes (L.C.mu0 j) PP (c.chi ((j : ℕ) + 1) PP)⁻¹) : c.Is

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsUnramifiedHeightFunctionLowerBlock

theorem **isHgt_lower** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ) (H : GL (A ⊕ D) k → ℝ), IsHgt ϖ H → IsHgt ϖ (fun m : GL D k => H (emb₂ k (Equiv.refl (A ⊕ D)) (1, m)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsUnramifiedHeightFunctionLowerBlockStatement

theorem **_root_.GlobalTheory.T.isUnramifiedHeightFunction_lowerBlock** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ) (H : GL (A ⊕ D) k → ℝ), IsHgt ϖ H → IsHgt ϖ (fun m : GL D k => H (emb₂ k (Equiv.refl (A ⊕ D)) (1, m)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsWeilDatumConjNegHermitianSpaces

theorem **cj_id_cj_id** {X : Type} (f : X → ℂ) : cj id (cj id f) = f

theorem **cj_id_apply** {X : Type} (f : X → ℂ) (x : X) : cj id f x = starRingEnd ℂ (f x)

theorem **hecke_inv** {χ : (AE D)ˣ →* Circle} (h : IsHeckeChar D χ) : IsHeckeChar D χ⁻¹

theorem **enorm_conj** (z : ℂ) : ‖starRingEnd ℂ z‖ₑ = ‖z‖ₑ

theorem **mt_negGA_symm** (V : HSp D) (g : GA D (negSp V)) : mt V ((negGA V).symm g) = mt (negSp V) g

theorem **continuous_negGA_symm** (V : HSp D) : Continuous (negGA V).symm

theorem **UJ_neg** {A : Type} [CommRing A] {n : Type} [Fintype n] [DecidableEq n] (σ : A →+* A) (J : Matrix n n A) : Formalism.UG.UJ σ (-J) = Formalism.UG.UJ σ J

theorem **negGA_symm_mem_GQ** (V : HSp D) (γ : GA D (negSp V)) (h : γ ∈ GQ (negSp V)) : (negGA V).symm γ ∈ GQ V

theorem **splitFacO_inv** (χS χT : (AE D)ˣ →* Circle) (x y : (AE D)ˣ) : ((splitFacO χS⁻¹ χT⁻¹ x y : Circle) : ℂ) = starRingEnd ℂ (splitFacO χS χT x y)

theorem **conjTarget_apply** {S T : HSp D} (om : Om S T) (p : GA D S × GA D (negSp T)) (f : WO S T → ℂ) : conjTarget om p f = cj id (om (p.1, (negGA T).symm p.2) (cj id f))

theorem **conjSource_apply** {S T : HSp D} (om : Om S T) (p : GA D (negSp S) × GA D T) (f : WO S T → ℂ) : conjSource om p f = cj id (om ((negGA S).symm p.1, p.2) (cj id f))

theorem **actO_negT** (S T : HSp D) (g : GA D S) (h : GA D (negSp T)) (w : WO S T) : actO S (negSp T) g h w = actO S T g ((negGA T).symm h) w

theorem **actO_negS** (S T : HSp D) (g : GA D (negSp S)) (h : GA D T) (w : WO S T) : actO (negSp S) T g h w = actO S T ((negGA S).symm g) h w

theorem **weil_conjTarget** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (hw : IsWeilDatumO ψ S T χS χT om) : IsWeilDatumO ψ S (negSp T) χS⁻¹ χT⁻¹ (conjTarget om)

theorem **weil_conjSource** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (hw : IsWeilDatumO ψ S T χS χT om) : IsWeilDatumO ψ (negSp S) T χS⁻¹ χT⁻¹ (conjSource om)

theorem **conj_isWeilDatumO** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → IsWeilDatumO ψ S (GlobalTheory.ChiGrid.negSp T) χS⁻¹ χT⁻¹ (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.conjTarget om) ∧ IsWeilDatumO ψ (GlobalTheory.ChiGrid.negSp S) T χS⁻¹ χT⁻¹ (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.conjSource om)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsWeilDatumConjNegHermitianSpacesStatement

theorem **_root_.GlobalTheory.T.isWeilDatum_conj_negHermitianSpaces** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → IsWeilDatumO ψ S (GlobalTheory.ChiGrid.negSp T) χS⁻¹ χT⁻¹ (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.conjTarget om) ∧ IsWeilDatumO ψ (GlobalTheory.ChiGrid.negSp S) T χS⁻¹ χT⁻¹ (GlobalTheory.IndefiniteRowSideFullHecke.ConjugateAdjointDagger.conjSource om)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsolateAbstract

theorem **isolate_abstract** : ∀ (X : Type) [AddCommGroup X] (Γ G : AddSubgroup X) (V : Type) [AddCommGroup V] [Module ℂ V] (T : X → V →ₗ[ℂ] V) (W : Set V) (β : X → X → ℂ) (a : X → ℂ) (χ : X → X → ℂ), Γ ≤ G → (∀ v, T 0 v = v) → (∀ x y v, T x (T y v) = β x y • T (x + y) v) → (∀ x y, β x y ≠ 0) → (∀ γ ∈ Γ, a γ ≠ 0 ∧ ∀ v ∈ W, T γ v = a γ • v) → (∀ γ ∈ Γ, ∀ x v, T γ (T x v) = χ x γ • T x (T γ v)) → (∀ x, ∀ γ ∈ Γ, ∀ γ' ∈ Γ, χ (x + γ') γ = χ x γ) → (∀ x ∈ G, ∀ y ∈ G, x - y ∉ Γ → ∃ γ ∈ Γ, χ x γ ≠ χ y γ) → ∀ (J : Finset X) (c : X → ℂ), (∀ x ∈ J, x ∈ G) → (∀ v ∈ W, ∑ x ∈ J, c x • T x v = 0) ∨ ∃ κ : X →₀ ℂ, (∀ y ∈ κ.support, y ∈ G) ∧ ∀ v ∈ W, ∑ y ∈ κ.support, κ y • T y (∑ x ∈ J, c x • T x v) = v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IsolateAbstractStatement

theorem **_root_.GlobalTheory.T.eq_zero_or_exists_inverse_combination_of_separating** : ∀ (X : Type) [AddCommGroup X] (Γ G : AddSubgroup X) (V : Type) [AddCommGroup V] [Module ℂ V] (T : X → V →ₗ[ℂ] V) (W : Set V) (β : X → X → ℂ) (a : X → ℂ) (χ : X → X → ℂ), Γ ≤ G → (∀ v, T 0 v = v) → (∀ x y v, T x (T y v) = β x y • T (x + y) v) → (∀ x y, β x y ≠ 0) → (∀ γ ∈ Γ, a γ ≠ 0 ∧ ∀ v ∈ W, T γ v = a γ • v) → (∀ γ ∈ Γ, ∀ x v, T γ (T x v) = χ x γ • T x (T γ v)) → (∀ x, ∀ γ ∈ Γ, ∀ γ' ∈ Γ, χ (x + γ') γ = χ x γ) → (∀ x ∈ G, ∀ y ∈ G, x - y ∉ Γ → ∃ γ ∈ Γ, χ x γ ≠ χ y γ) → ∀ (J : Finset X) (c : X → ℂ), (∀ x ∈ J, x ∈ G) → (∀ v ∈ W, ∑ x ∈ J, c x • T x v = 0) ∨ ∃ κ : X →₀ ℂ, (∀ y ∈ κ.support, y ∈ G) ∧ ∀ v ∈ W, ∑ y ∈ κ.support, κ y • T y (∑ x ∈ J, c x • T x v) = v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedDepthOneLevelCharOfOneRowFirstBlock

theorem **det_eq_one_of_one_row** {k n : Type*} [Field k] [Fintype n] [DecidableEq n] (M : Matrix n n k) (i : n) (h : ∀ a b, a ≠ i → M a b = (1 : Matrix n n k) a b) (hii : M i i = 1) : M.det = 1

theorem **iter_lam_one_row** (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (r t : Type) [Fintype r] [DecidableEq r] [Fintype t] [DecidableEq t] (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) (r₀ : r) : ∀ (h : (P.iter t μ hμ j).J), (∀ a b, a ≠ Sum.inl r₀ → (h : GL (Ix r t j) k).val a b = (1 : Matrix (Ix r t j) (Ix r t j) k) a b) → (h : GL (Ix r t j) k).val (Sum.inl r₀) (Sum.inl r₀) = 1 → ∃ g : P.J, (g : GL r k).val = (h : GL (Ix r t j) k).val.submatrix Sum.inl Sum.inl ∧ (P.iter t μ hμ j).lam h = P.lam g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedDepthOneLevelCharOfOneRowFirstBlockStatement

theorem **_root_.GlobalTheory.T.iteratedDepthOneLevel_char_of_one_row_first_block** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (r t : Type) [Fintype r] [DecidableEq r] [Fintype t] [DecidableEq t] (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) (r₀ : r) (h : (P.iter t μ hμ j).J), (∀ a b, a ≠ Sum.inl r₀ → (h : GL (Ix r t j) k).val a b = (1 : Matrix (Ix r t j) (Ix r t j) k) a b) → (h : GL (Ix r t j) k).val (Sum.inl r₀) (Sum.inl r₀) = 1 → ∃ g : P.J, (g : GL r k).val = (h : GL (Ix r t j) k).val.submatrix Sum.inl Sum.inl ∧ (P.iter t μ hμ j).lam h = P.lam g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedDepthOneLevelCharacterClosedForm

theorem **iter_lam_closed** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (r t : Type) [Fintype r] [DecidableEq r] [Fintype t] [DecidableEq t] (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) (g : (P.iter t μ hμ j).J), ∃ (g₀ : P.J) (D : Fin j → GL t k), (g₀ : GL r k).val = (g : GL (Ix r t j) k).val.submatrix Sum.inl Sum.inl ∧ (∀ i, D i ∈ KO k t ∧ (D i).val = Ix.diagonalBlockOfMatrix (g : GL (Ix r t j) k).val i) ∧ (P.iter t μ hμ j).lam g = P.lam g₀ * ∏ i : Fin j, μ (i.val + 1) (Matrix.GeneralLinearGroup.det (D i))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedDepthOneLevelCharacterClosedFormStatement

theorem **_root_.GlobalTheory.T.iteratedDepthOneLevel_character_closed_form** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (r t : Type) [Fintype r] [DecidableEq r] [Fintype t] [DecidableEq t] (P : TypeDatum k r) (μ : ℕ → kˣ →* ℂˣ) (hμ : ∀ j, IsTame (μ j)) (j : ℕ) (g : (P.iter t μ hμ j).J), ∃ (g₀ : P.J) (D : Fin j → GL t k), (g₀ : GL r k).val = (g : GL (Ix r t j) k).val.submatrix Sum.inl Sum.inl ∧ (∀ i, D i ∈ KO k t ∧ (D i).val = Ix.diagonalBlockOfMatrix (g : GL (Ix r t j) k).val i) ∧ (P.iter t μ hμ j).lam g = P.lam g₀ * ∏ i : Fin j, μ (i.val + 1) (Matrix.GeneralLinearGroup.det (D i))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedGainRapidDecay

theorem **nil_mem** {ι : Type} [DecidableEq ι] (B : Finset ι) (k : ℕ) : [] ∈ words B k

theorem **append_mem** {ι : Type} [DecidableEq ι] {B : Finset ι} {k : ℕ} {l : List ι} {i : ι} (hl : l ∈ words B k) (hi : i ∈ B) : l ++ [i] ∈ words B (k + 1)

theorem **mem_letters** {ι : Type} [DecidableEq ι] (B : Finset ι) : ∀ (k : ℕ), ∀ l ∈ words B k, ∀ i ∈ l, i ∈ B

theorem **root_le** {x y : ℝ} {m : ℕ} (hm : m ≠ 0) (hx : 0 ≤ x) (hy : 0 ≤ y) (h : x ^ m ≤ y ^ m) : x ≤ y

theorem **step** {G ι : Type} [DecidableEq ι] (B : Finset ι) (q : G → ℝ) (𝒞 : Set (G → ℂ)) (Dr : ι → (G → ℂ) → G → ℂ) (m : ℕ) (C δ₀ K : ℝ) (Q : G → ℝ) (A : ℕ) (hm : m ≠ 0) (hδ₀ : 0 < δ₀) (hC : 1 ≤ C) (hK : 1 ≤ K) (hKm : K ^ m = C) (hq : ∀ g, 1 ≤ q g) (hQ1 : ∀ g, 1 ≤ Q g) (hQm : ∀ g, Q g ^ m = q g) (hcl : ∀ ψ ∈ 𝒞, ∀ i ∈ B, Dr i ψ ∈ 𝒞) (hgain : ∀ g, ∃ δ : ℝ, 0 < δ ∧ δ ≤ 1 ∧ q g * δ ^ m ≤ C ∧ (δ₀ ≤ δ ∨ ∃ Ω : Set G, (∀ w ∈ Ω, q w ≤ C * q g ∧ q g ≤ C * q w) ∧ ∀ ψ ∈ 𝒞, ∀ c : ι → ℝ, (∀ i ∈ B, 0 ≤ c i ∧ ∀ w ∈ Ω, ‖Dr i ψ w‖ ≤ c i) → ‖ψ g‖ ≤ C * δ * ∑ i ∈ B, c i)) (k : ℕ) : ∃ Ck : ℝ, 0 ≤ Ck ∧ ∀ ψ ∈ 𝒞, ∀ c : List ι → ℝ, (∀ l ∈ words B k, 0 ≤ c l) → (∀ l ∈ words B k, ∀ g, ‖l.foldr Dr ψ g‖ ≤ c l * q g ^ A) → ∀ g, ‖ψ g‖ * Q g ^ k ≤ Ck * q g ^ A * ∑ l ∈ words B k, c l

theorem **iterated_gain_rapid_decay** : ∀ (G ι : Type) (B : Finset ι) (q : G → ℝ) (𝒞 : Set (G → ℂ)) (Dr : ι → (G → ℂ) → G → ℂ) (b : ℕ) (C δ₀ : ℝ), 0 < δ₀ → (∀ g, 1 ≤ q g) → (∀ ψ ∈ 𝒞, ∀ i ∈ B, Dr i ψ ∈ 𝒞) → (∀ g, ∃ δ : ℝ, 0 < δ ∧ δ ≤ 1 ∧ q g * δ ^ b ≤ C ∧ (δ₀ ≤ δ ∨ ∃ Ω : Set G, (∀ w ∈ Ω, q w ≤ C * q g ∧ q g ≤ C * q w) ∧ ∀ ψ ∈ 𝒞, ∀ c : ι → ℝ, (∀ i ∈ B, 0 ≤ c i ∧ ∀ w ∈ Ω, ‖Dr i ψ w‖ ≤ c i) → ‖ψ g‖ ≤ C * δ * ∑ i ∈ B, c i)) → ∀ A N : ℕ, ∃ (L : Finset (List ι)) (C' : ℝ), (∀ l ∈ L, ∀ i ∈ l, i ∈ B) ∧ ∀ ψ ∈ 𝒞, ∀ c : List ι → ℝ, (∀ l ∈ L, ∀ g, ‖l.foldr Dr ψ g‖ ≤ c l * q g ^ A) → ∀ g, ‖ψ g‖ * q g ^ N ≤ C' * ∑ l ∈ L, c l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedGainRapidDecayStatement

theorem **_root_.GlobalTheory.T.exists_rapid_decay_of_iterated_gain** : ∀ (G ι : Type) (B : Finset ι) (q : G → ℝ) (𝒞 : Set (G → ℂ)) (Dr : ι → (G → ℂ) → G → ℂ) (b : ℕ) (C δ₀ : ℝ), 0 < δ₀ → (∀ g, 1 ≤ q g) → (∀ ψ ∈ 𝒞, ∀ i ∈ B, Dr i ψ ∈ 𝒞) → (∀ g, ∃ δ : ℝ, 0 < δ ∧ δ ≤ 1 ∧ q g * δ ^ b ≤ C ∧ (δ₀ ≤ δ ∨ ∃ Ω : Set G, (∀ w ∈ Ω, q w ≤ C * q g ∧ q g ≤ C * q w) ∧ ∀ ψ ∈ 𝒞, ∀ c : ι → ℝ, (∀ i ∈ B, 0 ≤ c i ∧ ∀ w ∈ Ω, ‖Dr i ψ w‖ ≤ c i) → ‖ψ g‖ ≤ C * δ * ∑ i ∈ B, c i)) → ∀ A N : ℕ, ∃ (L : Finset (List ι)) (C' : ℝ), (∀ l ∈ L, ∀ i ∈ l, i ∈ B) ∧ ∀ ψ ∈ 𝒞, ∀ c : List ι → ℝ, (∀ l ∈ L, ∀ g, ‖l.foldr Dr ψ g‖ ≤ c l * q g ^ A) → ∀ g, ‖ψ g‖ * q g ^ N ≤ C' * ∑ l ∈ L, c l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedInnerIntegralAtFirstOfSplitPlaces

theorem **pcomb_eq_sum** {G : Type} (e : G →₀ ℂ) (F : G → ℂ) : GlobalTheory.PosOne.comb e F = ∑ a ∈ e.support, e a * F a

theorem **pcomb_congr** {G : Type} (e : G →₀ ℂ) {F F' : G → ℂ} (h : ∀ a ∈ e.support, F a = F' a) : GlobalTheory.PosOne.comb e F = GlobalTheory.PosOne.comb e F'

theorem **pcomb_mapDomain** {G H : Type} (e : G →₀ ℂ) (φ : G → H) (F : H → ℂ) : GlobalTheory.PosOne.comb (e.mapDomain φ) F = GlobalTheory.PosOne.comb e (fun a => F (φ a))

theorem **pcomb_fsum** {G ι : Type} (f : ι →₀ ℂ) (k : ι → ℂ → (G →₀ ℂ)) (F : G → ℂ) : GlobalTheory.PosOne.comb (f.sum k) F = f.sum fun a b => GlobalTheory.PosOne.comb (k a b) F

theorem **pcomb_single** {G : Type} (x : G) (y : ℂ) (F : G → ℂ) : GlobalTheory.PosOne.comb (Finsupp.single x y) F = y * F x

theorem **pcomb_conv_cons** {G : Type} [Monoid G] (e : G →₀ ℂ) (l : List (G →₀ ℂ)) (F : G → ℂ) : GlobalTheory.PosOne.comb (conv (e :: l)) F = GlobalTheory.PosOne.comb (conv l) (fun b => GlobalTheory.PosOne.comb e (fun a => F (a * b)))

theorem **conv_support_mem** {G : Type} [Monoid G] (M : Submonoid G) (l : List (G →₀ ℂ)) (h : ∀ e ∈ l, ∀ g ∈ e.support, g ∈ M) : ∀ g ∈ (conv l).support, g ∈ M

theorem **comb_eq_fsum** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T c f = c.sum fun u a => a • heisO ψ S T u f

theorem **comb_aconv_cons** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) (l : List (WO S T →₀ ℂ)) (h : ∀ x ∈ c.support, ∀ y ∈ (aconv l).support, symplO S T x y = 0) (f : WO S T → ℂ) : comb ψ S T (aconv (c :: l)) f = comb ψ S T c (comb ψ S T (aconv l) f)

theorem **congGrp_le** (V : HSp D) {I J : Ideal (AE D)} (hIJ : I ≤ J) : congGrp V I ≤ congGrp V J

theorem **grpOff_le_at'** (V : HSp D) {P : Pl D → Prop} {v : Pl D} (hv : P v) : grpOff V P ≤ grpOff V (· = v)

theorem **grpAt_le_grpOff'** (V : HSp D) {P P' : Pl D → Prop} (h : ∀ p, P p → ¬ P' p) : grpAt V P ≤ grpOff V P'

theorem **quotient_integral_fsum** {G : Type} [Group G] [MeasurableSpace G] {R : Subgroup G} {ρ : Measure R} {μ : Measure G} {ι : Type} (s : Finset ι) (g : ι → G → ℂ) (z : ι → ℂ) (hs : ∀ i ∈ s, IsQuotIntL1 ρ μ (g i) (z i)) : IsQuotIntL1 ρ μ (fun h => ∑ i ∈ s, g i h) (∑ i ∈ s, z i)

theorem **rest_grp** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ a ∈ (conv (List.ofFn fun i : Fin r => (Q i.succ).e₀.mapDomain (Q i.succ).gl₀)).support, a ∈ grpOff S.B.V (· = (Q 0).q.v)

theorem **innerVal_iter2** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ m₁ s₁ m₂ s₂, LocNorm (Q 0) m₁ s₁ m₂ s₂ → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → ∀ x ∈ grpAt S.J.W (places fun i => Q i.succ), ∀ y ∈ grpAt S.B'.V (places fun i => Q i.succ), Iter2 s₁ m₁ s₂ m₂ (fun x₀ y₀ => integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (x * (Q 0).gl₁ x₀) (y * (Q 0).gl₂ y₀)) (innerVal Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedInnerIntegralAtFirstOfSplitPlacesStatement

theorem **_root_.GlobalTheory.T.iteratedInnerIntegral_at_first_of_splitPlaces** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (r : ℕ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ m₁ s₁ m₂ s₂, LocNorm (Q 0) m₁ s₁ m₂ s₂ → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → ∀ x ∈ grpAt S.J.W (places fun i => Q i.succ), ∀ y ∈ grpAt S.B'.V (places fun i => Q i.succ), Iter2 s₁ m₁ s₂ m₂ (fun x₀ y₀ => integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ (x * (Q 0).gl₁ x₀) (y * (Q 0).gl₂ y₀)) (innerVal Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x y)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedIntegralValueAtRPlacesZero

theorem **quotient_integral_of_subsingleton** {G : Type} [Group G] [MeasurableSpace G] [Subsingleton G] {R : Subgroup G} (μR : Measure R) (μG : Measure G) (hR : μR Set.univ = 1) (hG : μG Set.univ = 1) (f : G → ℂ) : IsQuotIntL1 μR μG f (f 1)

theorem **conv_nil'** {G : Type} [Monoid G] : conv ([] : List (G →₀ ℂ)) = Finsupp.single 1 1

theorem **aconv_singleton'** {G : Type} [AddMonoid G] (c : G →₀ ℂ) : aconv [c] = c

theorem **aconv_nil'** {G : Type} [AddMonoid G] : aconv ([] : List (G →₀ ℂ)) = Finsupp.single 0 1

theorem **comb_single_zero'** (S' T : HSp D) (u : WO S' T → ℂ) : comb ψ S' T (Finsupp.single 0 1) u = u

theorem **trc_single_one'** {V : HSp D} (u : GA D V → ℂ) : trc (Finsupp.single 1 1) u = u

theorem **value_zero_eq** (Q : Fin 0 → QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (f₁ : WO S.B.V S.J.W → ℂ) (f₂ : WO S.J.W S.B'.V → ℂ) (φ : GA D S.B.A → ℂ) (f : GA D S.B'.V → ℂ) (g : GA D S.B.V) (h₁ : GA D S.J.W) (h₂ : GA D S.B'.V) : value Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ = integrand Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ 1 1

theorem **valQr_zero** : ∀ D : D7, ValQr D 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedIntegralValueAtRPlacesZeroStatement

theorem **_root_.GlobalTheory.T.iteratedIntegralValueAtRPlaces_zero** : ∀ D : D7, ValQr D 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedIntegralValueSuccPlacesOuter

theorem **gl₂_comm** (Q₀ Q' : QStep S) (hne : Q₀.q.v ≠ Q'.q.v) (a : GL (Q₀.n₂ ⊕ Fin (Q₀.c + 1)) Q₀.k) (b : GL (Q'.n₂ ⊕ Fin (Q'.c + 1)) Q'.k) : Q₀.gl₂ a * Q'.gl₂ b = Q'.gl₂ b * Q₀.gl₂ a

theorem **typed_trc** (Q₀ Q' : QStep S) (hne : Q₀.q.v ≠ Q'.q.v) (e : GL (Q₀.n₂ ⊕ Fin (Q₀.c + 1)) Q₀.k →₀ ℂ) {f : GA D S.B'.V → ℂ} (hf : Q'.Typed f) : Q'.Typed (trc (e.mapDomain Q₀.gl₂) f)

theorem **away_pair** (Q₀ Q' : QStep S) (hne : Q₀.q.v ≠ Q'.q.v) {A T : HSp D} (c d : WO A T →₀ ℂ) (hc : Q'.Away c) (hd : ∀ u ∈ d.support, IsHalf (Q₀.q.e + cA D Q₀.q.e) u) : Q'.Away (aconv [c, d])

theorem **hyps_succ** {r : ℕ} (Q : Fin (r + 1) → QStep S) (hinj : Function.Injective fun i => (Q i).q.v) {c₁ : WO S.B.A S.J.W₀ →₀ ℂ} {c₂ : WO S.J.W₀ S.B'.A →₀ ℂ} {f₁ : WO S.B.V S.J.W → ℂ} {f₂ : WO S.J.W S.B'.V → ℂ} {φ : GA D S.B.A → ℂ} {f : GA D S.B'.V → ℂ} {g : GA D S.B.V} {h₁ : GA D S.J.W} {h₂ : GA D S.B'.V} (hH : Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂) (γ : (Q 0).A₁) (β : (Q 0).A₂) : Hyps (fun i => Q i.succ) (aconv [c₁, (Q 0).d₁ γ]) (aconv [c₂, (Q 0).d₂ γ β]) f₁ f₂ φ (trc (((Q 0).e₂ β).mapDomain (Q 0).gl₂) f) g h₁ h₂

theorem **trc_conv_cons** {V : HSp D} (E : GA D V →₀ ℂ) (l : List (GA D V →₀ ℂ)) (h : ∀ x ∈ E.support, ∀ e ∈ l, ∀ y ∈ e.support, x * y = y * x) (f : GA D V → ℂ) : trc (conv (E :: l)) f = trc (conv l) (trc E f)

theorem **sum_pi_succ** {r : ℕ} {α : Fin (r + 1) → Type} [∀ i, Fintype (α i)] (F : (∀ i, α i) → ℂ) : ∑ γ, F γ = ∑ a : α 0, ∑ γ' : ∀ i : Fin r, α i.succ, F (Fin.cons a γ')

theorem **value_succ** {r : ℕ} (Q : Fin (r + 1) → QStep S) (hcomm : ∀ i : Fin r, ∀ a b, (Q 0).gl₂ a * (Q i.succ).gl₂ b = (Q i.succ).gl₂ b * (Q 0).gl₂ a) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ) (f₁ : WO S.B.V S.J.W → ℂ) (f₂ : WO S.J.W S.B'.V → ℂ) (φ : GA D

S.B.A → ℂ) (f : GA D S.B'.V → ℂ) (g : GA D S.B.V) (h₁ : GA D S.J.W) (h₂ : GA D S.B'.V) : value Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ = ∑ p : (Q 0).A₁ × (Q 0).A₂, (Q 0).volJ * ((Q 0).w₁ p.1 * (Q 0).w₂ p.2) * value (fun i => Q i.succ) (aconv [c₁, (Q 0).d₁ p.1]) (aconv [c₂, (Q 0).d₂ p.1 p.2]) f₁ f₂ φ (trc (((Q 0).e₂ p.2).mapDomain (Q 0).gl₂) f) g h₁ h₂

theorem **outer** : ∀ (D : D7) (r : ℕ), ValQr D r → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ μ₁ ρ₁ μ₂ ρ₂, Norm (fun i => Q i.succ) μ₁ ρ₁ μ₂ ρ₂ → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → Iter2 ρ₁ μ₁ ρ₂ μ₂ (fun x y => innerVal Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x.1 y.1) (value Q c₁ c₂ f₁ f₂ φ f g h₁ h₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedIntegralValueSuccPlacesOuterStatement

theorem **_root_.GlobalTheory.T.iteratedIntegralValue_succ_places_outer** : ∀ (D : D7) (r : ℕ), ValQr D r → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : Fin (r + 1) → QStep S), Function.Injective (fun i => (Q i).q.v) → ∀ μ₁ ρ₁ μ₂ ρ₂, Norm (fun i => Q i.succ) μ₁ ρ₁ μ₂ ρ₂ → ∀ c₁ c₂ f₁ f₂ φ f g h₁ h₂, Hyps Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ → Iter2 ρ₁ μ₁ ρ₂ μ₂ (fun x y => innerVal Q c₁ c₂ f₁ f₂ φ f g h₁ h₂ x.1 y.1) (value Q c₁ c₂ f₁ f₂ φ f g h₁ h₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedLevelPatternOfFramePlaces

theorem **patOfPlaces** : ∀ (F : Type) [Zero F] [One F] (κ ρ s τ j : ℕ) (R : ℕ → ℕ → F), ((∀ p', p' < κ + s + 1 → R p' p' = 1 ∧ ∀ p, p < κ + s + 1 + j * τ → p ≠ p' → (κ + s + 1 ≤ p ∨ symIdx κ ρ p' < symIdx κ ρ p ∨ (p' = κ + ρ ∧ p < κ)) → R p p' = 0) ∧ (∀ p p', κ + s + 1 ≤ p' → p' < κ + s + 1 + j * τ → p < κ + s + 1 + j * τ → (p' - (κ + s + 1)) / τ < (p - (κ + s + 1)) / τ → R p p' = 0)) → (∀ (a b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) j), ¬ FreeIx ρ a b → R (zPos a) (zPos b) = if a = b then 1 else 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedLevelPatternOfFramePlacesStatement

theorem **_root_.GlobalTheory.T.iteratedLevelPattern_of_framePlaces** : ∀ (F : Type) [Zero F] [One F] (κ ρ s τ j : ℕ) (R : ℕ → ℕ → F), ((∀ p', p' < κ + s + 1 → R p' p' = 1 ∧ ∀ p, p < κ + s + 1 + j * τ → p ≠ p' → (κ + s + 1 ≤ p ∨ symIdx κ ρ p' < symIdx κ ρ p ∨ (p' = κ + ρ ∧ p < κ)) → R p p' = 0) ∧ (∀ p p', κ + s + 1 ≤ p' → p' < κ + s + 1 + j * τ → p < κ + s + 1 + j * τ → (p' - (κ + s + 1)) / τ < (p - (κ + s + 1)) / τ → R p p' = 0)) → (∀ (a b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) j), ¬ FreeIx ρ a b → R (zPos a) (zPos b) = if a = b then 1 else 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedLevelPatternToFramePlaces

theorem **zPos_surj** {κ τ l j p : ℕ} (hp : p < κ + l + j * τ) : ∃ a : Ix (Fin κ ⊕ Fin l) (Fin τ) j, zPos a = p

theorem **symIdx_cases** (κ ρ i : ℕ) : (i < κ ∧ symIdx κ ρ i = i + ρ) ∨ (κ ≤ i ∧ i < κ + ρ ∧ symIdx κ ρ i = i - κ) ∨ (κ + ρ ≤ i ∧ symIdx κ ρ i = i)

theorem **block_div** {r τ i u : ℕ} (hu : u < τ) : (r + i * τ + u - r) / τ = i

theorem **pattern_to_places** (F : Type) [Zero F] [One F] (κ ρ s τ j : ℕ) (R : ℕ → ℕ → F) : (∀ (a b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) j), ¬ FreeIx ρ a b → R (zPos a) (zPos b) = if a = b then 1 else 0) → ((∀ p', p' < κ + s + 1 → R p' p' = 1 ∧ ∀ p, p < κ + s + 1 + j * τ → p ≠ p' → (κ + s + 1 ≤ p ∨ symIdx κ ρ p' < symIdx κ ρ p ∨ (p' = κ + ρ ∧ p < κ)) → R p p' = 0) ∧ (∀ p p', κ + s + 1 ≤ p' → p' < κ + s + 1 + j * τ → p < κ + s + 1 + j * τ → (p' - (κ + s + 1)) / τ < (p - (κ + s + 1)) / τ → R p p' = 0))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedLevelPatternToFramePlacesStatement

theorem **_root_.GlobalTheory.T.iteratedLevelPattern_to_framePlaces** : ∀ (F : Type) [Zero F] [One F] (κ ρ s τ j : ℕ) (R : ℕ → ℕ → F), (∀ (a b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) j), ¬ FreeIx ρ a b → R (zPos a) (zPos b) = if a = b then 1 else 0) → ((∀ p', p' < κ + s + 1 → R p' p' = 1 ∧ ∀ p, p < κ + s + 1 + j * τ → p ≠ p' → (κ + s + 1 ≤ p ∨ symIdx κ ρ p' < symIdx κ ρ p ∨ (p' = κ + ρ ∧ p < κ)) → R p p' = 0) ∧ (∀ p p', κ + s + 1 ≤ p' → p' < κ + s + 1 + j * τ → p < κ + s + 1 + j * τ → (p' - (κ + s + 1)) / τ < (p - (κ + s + 1)) / τ → R p p' = 0))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedThetaUnfoldingOfDoubleStepLastTwoSteps

theorem **rTrans_lift** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (s : Step ψ V V₀ W W₀ ρ σ) (G : GA D W → ℂ) (Φ : WO V W → ℂ) (g : GA D V) : rTrans g (s.lift G Φ) = s.lift G (s.om (g, 1) Φ)

theorem **GlobalTheory.Unf.iter_of_doubleStep_lastLaw** : ∀ D : D7, DoubleStep D → LastLaw D → IT D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedThetaUnfoldingOfDoubleStepLastTwoStepsStatement

theorem **_root_.GlobalTheory.T.iteratedThetaUnfolding_of_doubleStep_lastTwoSteps** : ∀ D : D7, DoubleStep D → LastLaw D → IT D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedThetaUnfoldingOfWeilDatumDescent

theorem **GlobalTheory.Unf.iteratedUnfolding_of_descent** : ∀ D : D7, Descent D → IT D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratedThetaUnfoldingOfWeilDatumDescentStatement

theorem **_root_.GlobalTheory.T.iteratedThetaUnfolding_of_weilDatumDescent** : ∀ D : D7, Descent D → IT D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratesSpanConsecutiveSpaces

theorem **itSpans_pk** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j + 1 < D.R.m → GlobalTheory.Core.Mi.ItSpans ((pk W).MY j) ((pk W).MY (j + 1)) ((pk W).ItY j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratesSpanConsecutiveSpacesStatement

theorem **_root_.GlobalTheory.T.iterates_span_consecutive_spaces** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ j, j + 1 < D.R.m → GlobalTheory.Core.Mi.ItSpans ((pk W).MY j) ((pk W).MY (j + 1)) ((pk W).ItY j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratesSpanLastStep

theorem **itSpans_pk** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc :

Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → GlobalTheory.Core.Mi.ItSpans ((pk W).MY (D.R.m − 1)) (pk W).MX (pk W).ItX

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IteratesSpanLastStepStatement

theorem **_root_.GlobalTheory.T.iterates_span_last_step** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → GlobalTheory.Core.Mi.ItSpans ((pk W).MY (D.R.m − 1)) (pk W).MX (pk W).ItX

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDecompositionIsotropicStabilizerArchimedeStatement

theorem **_root_.GlobalTheory.T.iwasawa_decomposition_isotropicStabilizer_archimedean** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (b : GL (Fin T.rank) (AE D)), IsMaj T b → ∀ y ∈ GinfC T, ∃ q ∈ stab d, ∃ k ∈ Kmax T b, q ∈ GinfC T ∧ y = q ∗ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDecompositionIsotropicStabilizerArchimedean

theorem **local_decomp** {A : Type} [CommRing A] {n r : ℕ} (σ : A →+∗ A) (f : A →+∗ ℂ) (hf : Function.Bijective f) (hσ : ∀ x, f (σ x) = star (f x)) {J : Matrix (Fin n) (Fin n) A} (b : GL (Fin n) A) (ε : Fin n → Bool) (hb : Formalism.UG.cT σ b.1 ∗ J ∗ b.1 = Matrix.diagonal fun i => if ε i then 1 else -1) (X : Matrix (Fin n) (Fin r) A) (hX : Formalism.UG.cT σ X ∗ J ∗ X = 0) (g : GL (Fin n) A) (hg : g ∈ UJ σ J) : ∃ k ∈ UJ σ J, b⁻¹ ∗ k ∗ b ∈ UJ σ (1 : Matrix (Fin n) (Fin n) A) ∧ ∃ β : Matrix (Fin r) (Fin r) A, (g ∗ k⁻¹).1 ∗ X = X ∗ β

theorem **iso_arch** {D : D7} {T T₀ : HSp D} {r : ℕ} (d : HypDec T T₀ r) (w : InfinitePlace D.host.F) : (Formalism.UG.cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR w.Completion D.sp.cm.sig) (Matrix.map d.jX (compInf D.host.F D.host.E w)) ∗ Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR w.Completion T.gram : Matrix (Fin r) (Fin T.rank) _) ∗ Matrix.map d.jX (compInf D.host.F D.host.E w) = 0

theorem **mem_stab_of_arch** {D : D7} {T T₀ : HSp D} {r : ℕ} (d : HypDec T T₀ r) (q : GA D T) (hfin : q ∈ GinfC T) (harch : ∀ w : InfinitePlace D.host.F, ∃ β : Matrix (Fin r) (Fin r) (TensorProduct D.host.F w.Completion D.host.E), (matHom _ D.sp.cm.sig T.gram (projArch D.sp.cm.sig T.gram w q)).1 ∗ Matrix.map d.jX (compInf D.host.F D.host.E w) = Matrix.map d.jX (compInf D.host.F D.host.E w) ∗ β) : q ∈ stab d

theorem **glue** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (k : ∀ w : InfinitePlace F, GArch σ J w) : ∃ g : GAd σ J, (∀ w, projArch σ J w g = k w) ∧ projFin σ J g = 1

theorem **local_at_arch** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} (hc : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.IsCMExt F E σ) {n r : ℕ} (J : Matrix (Fin n) (Fin n) E) (w : InfinitePlace F) (b : GL (Fin n) (TensorProduct F w.Completion E)) (ε : Fin n → Bool) (hb : Formalism.UG.cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR w.Completion σ) b.1 ∗ Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR w.Completion J ∗ b.1 = Matrix.diagonal fun i => if ε i then 1 else -1) (X : Matrix (Fin n) (Fin r) (TensorProduct F w.Completion E)) (hX : (Formalism.UG.cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR w.Completion σ) X ∗ Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR w.Completion J : Matrix (Fin r) (Fin n) _) ∗ X = 0) (g : GL (Fin n) (TensorProduct F w.Completion E))

theorem **mem_stabArch_of** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (w : InfinitePlace F) (b : GL (Fin n) (TensorProduct F w.Completion E)) (g : GArch σ J w) (h : b⁻¹ ∗ matHom _ σ J g ∗ b ∈ UJ (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR w.Completion σ) 1) : g ∈ stabArch σ J w b

theorem **iwasawa_stab_arch** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (b : GL (Fin T.rank) (AE D)), IsMaj T b → ∀ y ∈ GinfC T, ∃ q ∈ stab d, ∃ k ∈ Kmax T b, q ∈ GinfC T ∧ y = q ∗ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDecompositionSelfDualWittSplitAtPlace

theorem **flag_times_stab** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSelfDualC S L → IsSelfDualC S₀ L₀ → MonoAt d L L₀ v → ∀ h ∈ grpAt S (· = v), ∃ p ∈ grpAt S (· = v), (∃ a : Mx D ρ ρ, (∀ i j, j < i → a i j = 0) ∧ mt S p · d.jX = d.jX · a) ∧ Moves S (p⁻¹ ∗ h) L L (· = v)

theorem **iwasawa_at_place** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSelfDualC S L → IsSelfDualC S₀ L₀ → MonoAt d L L₀ v → ∀ h ∈ grpAt S (· = v), ∃ n ∈ flagN d, n ∈ grpAt S (· = v) ∧ ∃ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) ∧ d.levi t ∈ grpAt S (· = v) ∧ ∃ γ ∈ grpAt S₀ (· = v), ∃ κ, InK S L (· = v) κ ∧ h = n ∗ d.levi t ∗ d.emb0 γ ∗ κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDecompositionSelfDualWittSplitAtPlaceStatement

theorem **_root_.GlobalTheory.T.iwasawa_decomposition_selfDual_wittSplit_at_place** : ∀ (D : D7) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (L : Lat S) (L₀ : Lat S₀) (v : Pl D), Tame D v → IsSelfDualC S L → IsSelfDualC S₀ L₀ → MonoAt d L L₀ v → ∀ h ∈ grpAt S (· = v), ∃ n ∈ flagN d, n ∈ grpAt S (· = v) ∧ ∃ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) ∧ d.levi t ∈ grpAt S (· = v) ∧ ∃ γ ∈ grpAt S₀ (· = v), ∃ κ, InK S L (· = v) κ ∧ h = n ∗ d.levi t ∗ d.emb0 γ ∗ κ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDecompositionUnitary

theorem **exists_mul_of_rank** (D : D7) (A : HSp D) (g : GlobalTheory.StdUnn.G D A.rank) : ∃ p ∈ GlobalTheory.StdUnn.P D A.rank, ∃ k ∈ GlobalTheory.StdUnn.K D A.rank, g = p ∗ k

theorem **exists_mul** : ∀ (D : D7) (n : ℕ) (g : GlobalTheory.StdUnn.G D n), ∃ p ∈ GlobalTheory.StdUnn.P D n, ∃ k ∈ GlobalTheory.StdUnn.K D n, g = p ∗ k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDoubledHermitianSpaceNonsplitPlace

theorem **cT_cT** (hs : ∀ x, s (s x) = x) (M : Matrix n ι 𝕆) : cT s (cT s M) = M

theorem **exists_dual_isotropic** (hs : ∀ x, s (s x) = x) {τ : 𝕆} (hτ : τ + s τ = 1) {Q : Matrix ι ι 𝕆} (hQ : cT s Q = Q) (hu : IsUnit Q) {Y : Matrix ι n 𝕆} {U : Matrix n ι 𝕆} (hU : U ∗ Y = 1) (hY : cT s Y ∗ Q ∗ Y = 0) : ∃ Z : Matrix ι n 𝕆, cT s Y ∗ Q ∗ Z = 1 ∧ cT s Z ∗ Q ∗ Y = 1 ∧ cT s Z ∗ Q ∗ Z = 0

theorem **exists_hyperbolic** (e : n ⊕ n ≃ ι) (hs : ∀ x, s (s x) = x) {τ : 𝕆} (hτ : τ + s τ = 1) {Q : Matrix ι ι 𝕆} (hQ : cT s Q = Q) (hu : IsUnit Q) {Y : Matrix ι n 𝕆} {U : Matrix n ι 𝕆} (hU : U ∗ Y = 1) (hY : cT s Y ∗ Q ∗ Y = 0) : ∃ (T : Matrix ι (n ⊕ n) 𝕆) (S : Matrix (n ⊕ n) ι 𝕆), S ∗ T = 1 ∧ T ∗ S = 1 ∧ cT s T ∗ Q ∗ T = fromBlocks 0 1 1 0 ∧ T ∗ (fromRows 1 0 : Matrix (n ⊕ n) n 𝕆) = Y

theorem **exists_isometry** (e : n ⊕ n ≃ ι) (hs : ∀ x, s (s x) = x) {τ : 𝕆} (hτ : τ + s τ = 1) {Q : Matrix ι ι 𝕆} (hQ : cT s Q = Q) (hu : IsUnit Q) {Y Y' : Matrix ι n 𝕆} {U U' : Matrix n ι 𝕆} (hU : U ∗ Y = 1) (hY : cT s Y ∗ Q ∗ Y = 0) (hU' : U' ∗ Y' = 1) (hY' : cT s Y' ∗ Q ∗ Y' = 0) : ∃ k ∈ UJ s Q, k.1 ∗ Y' = Y

theorem **dCol_mulVec** (x : n → A) : dCol A e *ᵥ x = pairVec e x x

theorem **fCol_mulVec** (x : n → A) : fCol A e *ᵥ x = pairVec e x 0

theorem **transpose_fCol_mul** (M : Matrix (n ⊕ n) n A) : (fCol A e)ᵀ ∗ M.submatrix e.symm id = M.toRows₁

theorem **dCol_map** {B : Type} [CommRing B] (f : A →+* B) : (dCol A e).map f = dCol B e

theorem **fCol_map** {B : Type} [CommRing B] (f : A →+* B) : (fCol A e).map f = fCol B e

theorem **dCol_isotropic** (σ : A →+* A) (J : Matrix n n A) : cT σ (dCol A e) ∗ dblGram e J ∗ dCol A e = 0

theorem **keepsDiag_of_dCol** {M : Matrix ι ι A} {c : Matrix n n A} (h : M ∗ dCol A e = dCol A e ∗ c) : KeepsDiag e M

theorem **exists_std** (O : Subring A) (hO : HasFrames e O) {σ : A →+* A} {s : O →+* O} (hσ : ∀ x : O, (s x : A) = σ x) (hs : ∀ x, s (s x) = x) {τ : O} (hτ : τ + s τ = 1) {J : Matrix n n O} (hJ : cT s J = J) (hu : IsUnit J) {h : GL ι A} (hh : h ∈ UJ σ (dblGram e (J.map O.subtype))) : ∃ k ∈ UJ s (dblGram e J), ∃ c : Matrix n n A, h.1 ∗ dCol A e = k.1.map O.subtype ∗ dCol A e ∗ c

theorem **mulVec_mem_iff** (O : Subring A) (k : GL ι O) (y : ι → A) : (∀ i, y i ∈ O) ↔ ∀ i, (k.1.map O.subtype *ᵥ y) i ∈ O

theorem **hform_mulVec** (σ : A →+* A) (M a : Matrix n n A) (x y : n → A) : hform σ M (a *ᵥ x) (a *ᵥ y) = (σ ∘ x) ⬝ᵥ ((cT σ a ∗ M ∗ a) *ᵥ y)

theorem **exists_gram** (O : Subring A) {σ : A →+* A} (hσ : ∀ x ∈ O, σ x ∈ O) {J : Matrix n n A} (hu : IsUnit J) {Λ : Submodule O (n → A)} (hΛ : ∀ y, y ∈ Λ ↔ ∀ x ∈ Λ, hform σ J x y ∈ O) {a : GL n A} (ha : ∀ y, (∀ i, y i ∈ O) ↔ a.1 *ᵥ y ∈ Λ) : ∃ B : Matrix n n O, IsUnit B ∧ B.map O.subtype = cT σ a.1 ∗ J ∗ a.1

theorem **exists_stab** (O : Subring A) (hO : HasFrames e O) {σ : A →+* A} (hσ : ∀ x ∈ O, σ x ∈ O) (hs : ∀ x, σ (σ x) = x) {τ : A} (hτO : τ ∈ O) (hτ : τ + σ τ = 1) {J : Matrix n n A} (hJ : cT σ J = J) (hu : IsUnit J) {Λ : Submodule O (n → A)} (hΛ : ∀ y, y ∈ Λ ↔ ∀ x ∈ Λ, hform σ J x y ∈ O) {a : GL n A} (ha : ∀ y, (∀ i, y i ∈ O) ↔ a.1 *ᵥ y ∈ Λ) {g : GL ι A} (hg : g ∈ UJ σ (dblGram e J)) : ∃ k ∈ UJ σ (dblGram e J), k ∈ stabGL (sumLat e Λ Λ) ∧ g ∗ k ∈ stabGL (diagSub A e)

theorem **hasFrames_of_equiv** {m : ℕ} (O : Subring A) {K : Type} [Field K] (V : ValuationSubring K) (φ : A ≃+* K) (hφ : ∀ y, y ∈ O ↔ φ y ∈ V) : HasFrames (finSumFinEquiv : Fin m ⊕ Fin m ≃ _) O

theorem **hasIwasawa_of_inert** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} (hσ : ∀ x, σ (σ x) = x) {c : 𝒪 E} (hc : (c : E) + σ c = 1) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) {Q : Matrix (Fin (n + n)) (Fin (n + n)) E} (hQ : Q = dblGram finSumFinEquiv J) {L : Submodule (𝒪 E) (Fin n → E)} [Submodule.IsLattice E L] {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (hw : Formalism.UG.IsInertAt F E v w) (hL : IsSelfDualAt σ J L v) : HasIwasawa σ Q (KFin σ Q L v)

theorem **hasIwasawa_inert** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ v : FinPl D, ¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDoubledHermitianSpaceNonsplitPlaceStatement

theorem **_root_.GlobalTheory.T.iwasawa_doubled_hermitian_space_nonsplit_place** : ∀ (D : D7) (A : HSp D) (L : Lat A), IsSelfDualC A L → ∀ v : FinPl D, ¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDoubledHermitianSpaceRealPlace

theorem **cT_add** (σ : A →+* A) (M N : Matrix n n A) : cT σ (M + N) = cT σ M + cT σ N

theorem **cT_sub** (σ : A →+* A) (M N : Matrix n n A) : cT σ (M − N) = cT σ M − cT σ N

theorem **map_cT** (σ : A →+* A) (f : A →+* ℂ) (hσ : ∀ x, f (σ x) = star (f x)) (M : Matrix n n A) : (cT σ M).map f = (M.map f)ᴴ

theorem **isUnit_of_isotropic** (x y x' y' : Matrix n n ℂ) (h1 : xᴴ ∗ x = yᴴ ∗ y) (h2 : x'ᴴ ∗ x − y'ᴴ ∗ y = 1 + 1) : IsUnit x

theorem **exists_mul_inv_mem** (σ : A →+* A) (f : A →+* ℂ) (hf : Function.Bijective f) (hσ : ∀ x, f (σ x) = star (f x)) (e : n ⊕ n ≃ ι) (G : GL ι A) (hG : G ∈ UJ σ (dblGram e (1 : Matrix n n A))) : ∃ K ∈ UJ σ (dblGram e (1 : Matrix n n A)), K ∈ UJ σ (1 : Matrix ι ι A) ∧ G ∗ K⁻¹ ∈ stabGL (diagSub A e)

theorem **conj_mem_UJ** (σ : A →+* A) {Q : Matrix ι ι A} (C : GL ι A) {M : GL ι A} (hM : M ∈ UJ σ Q) : C⁻¹ ∗ M ∗ C ∈ UJ σ (cT σ C.1 ∗ Q ∗ C.1)

theorem **cT_inv_cancel** (σ : A →+* A) (Q : Matrix ι ι A) (C : GL ι A) : cT σ (C⁻¹).1 ∗ (cT σ C.1 ∗ Q ∗ C.1) ∗ (C⁻¹).1 = Q

theorem **exists_of_conj** (σ : A →+* A) (e : n ⊕ n ≃ ι) {Q : Matrix ι ι A} (C : GL ι A) (hC : cT σ C.1 ∗ Q ∗ C.1 = dblGram e (1 : Matrix n n A)) (hCd : C ∈ stabGL (diagSub A e)) (hcore : ∀ G ∈ UJ σ (dblGram e (1 : Matrix n n A)), ∃ K ∈ UJ σ (dblGram e (1 : Matrix n n A)), K ∈ UJ σ (1 : Matrix ι ι A) ∧ G ∗ K⁻¹ ∈ stabGL (diagSub A e)) (g : GL ι A) (hg : g ∈ UJ σ Q) : ∃ k ∈ UJ σ Q, C⁻¹ ∗ k ∗ C ∈ UJ σ (1 : Matrix ι ι A) ∧ g ∗ k⁻¹ ∈ stabGL (diagSub A e)

theorem **swapMat_mul_self** (ε : n → Bool) : swapMat (A

theorem **cT_swapMat** (σ : A →+* A) (ε : n → Bool) : cT σ (swapMat (A

theorem **swapMat_dblGram** (ε : n → Bool) : swapMat (A

theorem **keepsDiag_swapMat** (ε : n → Bool) : KeepsDiag e (swapMat (A

theorem **swapGL_mem_UJ_one** (σ : A →+* A) (ε : n → Bool) : swapGL (A

theorem **swapGL_mem_stabGL** (ε : n → Bool) : swapGL (A

theorem **exists_decomp_frame** (σ : A →+* A) (f : A →+* ℂ) (hf : Function.Bijective f) (hσ : ∀ x, f (σ x) = star (f x)) (e : n ⊕ n ≃ ι) {J : Matrix n n A} (b : GL n A) (ε : n → Bool) (hb : cT σ b.1 ∗ J ∗ b.1 = sgnMat ε) (g : GL ι A) (hg : g ∈ UJ σ (dblGram e J)) : ∃ k ∈ UJ σ (dblGram e J), (blkGL e (b, b))⁻¹ ∗ k ∗ blkGL e (b, b) ∈ UJ σ (1 : Matrix ι ι A) ∧ g ∗ k⁻¹ ∈ stabGL (diagSub A e)

theorem **hasIwasawa_KArch** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} (hc : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.IsCMExt F E σ) {m : ℕ} {J : Matrix (Fin m) (Fin m) E} {Q : Matrix (Fin (m + m)) (Fin (m + m)) E} (hQ : Q = dblGram finSumFinEquiv J) (w : InfinitePlace F) (b : GL (Fin m) (w.Completion ⊗[F] E)) (ε : Fin m → Bool) (hb : cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR w.Completion σ) b.1 ∗ Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.JR w.Completion J ∗ b.1 = diagonal fun i => if ε i then 1 else −1) : HasIwasawa σ Q (KArch σ Q w b)

theorem **hasIwasawa_dblSp_arch** : ∀ (D : D7) (A : HSp D) (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ w : InfinitePlace D.host.F, HasIwasawa D.sp.cm.sig (dblSp A).gram (KArch D.sp.cm.sig (dblSp A).gram w (Formalism.UG.bc (compInf D.host.F D.host.E w) a))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaDoubledHermitianSpaceRealPlaceStatement

theorem **_root_.GlobalTheory.T.iwasawa_doubled_hermitian_space_real_place** : ∀ (D : D7) (A : HSp D) (a : GL (Fin A.rank) (AE D)), IsMaj A a → ∀ w : InfinitePlace D.host.F, HasIwasawa D.sp.cm.sig (dblSp A).gram (KArch D.sp.cm.sig (dblSp A).gram w (Formalism.UG.bc (compInf D.host.F D.host.E w) a))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicColumnsRingIsoComplex

theorem **map_cT_rect** (σ : A →+* A) (f : A →+* ℂ) (hσ : ∀ x, f (σ x) = star (f x)) {m p : Type} (M : Matrix m p A) : (cT σ M).map f = (M.map f)ᴴ

theorem **exists_compact_keep_cols** {n r : ℕ} (σ : A →+* A) (f : A →+* ℂ) (hf : Function.Bijective f) (hσ : ∀ x, f (σ x) = star (f x)) {J : Matrix (Fin n) (Fin n) A} (b : GL (Fin n) A) (ε : Fin n → Bool) (hb : cT σ b.1 * J * b.1 = (Matrix.diagonal fun i => if ε i then 1 else -1)) (X : Matrix (Fin n) (Fin r) A) (hX : cT σ X * J * X = 0) (g : GL (Fin n) A) (hg : g ∈ UJ σ J) : ∃ k ∈ UJ σ J, b⁻¹ * k * b ∈ UJ σ (1 : Matrix (Fin n) (Fin n) A) ∧ ∃ β : Matrix (Fin r) (Fin r) A, (g * k⁻¹).1 * X = X * β

theorem **iw_iso_cols** : ∀ {A : Type} [CommRing A] {n r : ℕ} (σ : A →+* A) (f : A →+* ℂ), Function.Bijective f → (∀ x, f (σ x) = star (f x)) → ∀ {J : Matrix (Fin n) (Fin n) A} (b : GL (Fin n) A) (ε : Fin n → Bool), Formalism.UG.cT σ b.1 * J * b.1 = (Matrix.diagonal fun i => if ε i then 1 else -1) → ∀ (X : Matrix (Fin n) (Fin r) A), Formalism.UG.cT σ X * J * X = 0 → ∀ g : GL (Fin n) A, g ∈ UJ σ J → ∃ k ∈ UJ σ J, b⁻¹ * k * b ∈ UJ σ (1 : Matrix (Fin n) (Fin n) A) ∧ ∃ β : Matrix (Fin r) (Fin r) A, (g * k⁻¹).1 * X = X * β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicColumnsRingIsoComplexStatement

theorem **_root_.GlobalTheory.T.iwasawa_isotropic_columns_ring_iso_complex** : ∀ {A : Type} [CommRing A] {n r : ℕ} (σ : A →+* A) (f : A →+* ℂ), Function.Bijective f → (∀ x, f (σ x) = star (f x)) → ∀ {J : Matrix (Fin n) (Fin n) A} (b : GL (Fin n) A) (ε : Fin n → Bool), Formalism.UG.cT σ b.1 * J * b.1 = (Matrix.diagonal fun i => if ε i then 1 else -1) → ∀ (X : Matrix (Fin n) (Fin r) A), Formalism.UG.cT σ X * J * X = 0 → ∀ g : GL (Fin n) A, g ∈ UJ σ J → ∃ k ∈ UJ σ J, b⁻¹ * k * b ∈ UJ σ (1 : Matrix (Fin n) (Fin n) A) ∧ ∃ β : Matrix (Fin r) (Fin r) A, (g * k⁻¹).1 * X = X * β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicStabiliserFinitePlace

theorem **iwasawa_fin** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r), (∀ v : FinPl D, (2 : 𝒪 D.host.F) ∈ v.asIdeal → Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicStabiliserFinitePlaceStatement

theorem **_root_.GlobalTheory.T.iwasawa_isotropic_stabiliser_finite_place** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T T₀ r), (∀ v : FinPl D, (2 : 𝒪 D.host.F) ∈ v.asIdeal → Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicStabiliserSplitPlace

theorem **map_keeps** (f : A →+* R) {M : Matrix n n A} {j : Matrix n ρ A} {p : Matrix ρ n A} (hp : p * j = 1) {a : Matrix ρ ρ R} (h : M.map f * j.map f = j.map f * a) : (M * j).map f = (j * (p * (M * j))).map f

theorem **keeps_cols** (ε : n ≃ (ρ ⊕ s) ⊕ ρ) {e e' : Matrix n ρ A} {i : Matrix n s A} {B b : GL n A} (hB : B.1 = (Matrix.fromCols (Matrix.fromCols e i) e').submatrix id ε) (hb : b.1.BlockTriangular (lev ∘ ε)) : ∃ a : Matrix ρ ρ A, (B * b * B⁻¹).1 * e = e * a

theorem **keeps_rows** (ε : n ≃ (ρ ⊕ s) ⊕ ρ) {p p' : Matrix ρ n A} {q : Matrix s n A} {B b : GL n A} (hB : (B⁻¹).1 = (Matrix.fromRows (Matrix.fromRows p q) p').submatrix ε id) (hb : b.1.BlockTriangular (lev ∘ ε)) : ∃ c : Matrix ρ ρ A, p' * (B * b * B⁻¹).1 = c * p'

theorem **keeps_dual** {G x x' : Matrix n n A} {e f : Matrix n ρ A} {c : Matrix ρ ρ A} {xi Gi : Matrix n n A} (hx : x * xi = 1) (hG : G * Gi = 1) (h : x'.transpose * G * x = G) (hf : e.transpose * G * f = 1) (hc : e.transpose * G * x = c * (e.transpose * G)) : ∃ a : Matrix ρ ρ A, x' * e = e * a

theorem **emb_keeps** (h : IsCMExt F E σ) (τ : E →ₐ[F] Fv F v) (q : GLoc σ J v) {j : Matrix (Fin n) ρ (AF F ⊗[F] E)} {p : Matrix ρ (Fin n) (AF F ⊗[F] E)} (hp : p * j = 1) {e : Matrix (Fin n) ρ E} (hj : j = e.map (inclR (AF F))) {a a' : Matrix ρ ρ (Fv F v)} (h1 : (splitHom σ J τ q).1 * e.map τ = e.map τ * a) (h2 : (splitHom σ J (τ.comp σ) q).1 * e.map (τ.comp σ) = e.map (τ.comp σ) * a') : (matHom _ σ J (emb σ J v q)).1 * j = j * (p * ((matHom _ σ J (emb σ J v q)).1 * j))

theorem **exists_flag_basis** (τ : E →ₐ[F] Fv F v) (hJ : IsHermND σ J) {e e' : Matrix (Fin n) ρ E} {i : Matrix (Fin n) s E} {p : Matrix ρ (Fin n) E} {q : Matrix s (Fin n) E} (ε : Fin n ≃ (ρ ⊕ s) ⊕ ρ) (hsum : e * p + i * q + e' * ((e.map σ)ᵀ * J) = 1) (hf : (e.map σ)ᵀ * J * e' = 1) : ∃ (B : GL (Fin n) (Fv F v)) (lv : Fin n → ℕ), ∀ g : GLoc σ J v, (B⁻¹ * splitHom σ J τ g * B).1.BlockTriangular lv → (∃ a : Matrix ρ ρ (Fv F v), (splitHom σ J τ g).1 * e.map τ = e.map τ * a) ∧ ∃ a : Matrix ρ ρ (Fv F v), (splitHom σ J (τ.comp σ) g).1 * e.map (τ.comp σ) = e.map (τ.comp σ) * a

theorem **iwasawa_of_triangular** (h : IsCMExt F E σ) (hJ : IsHermND σ J) (τ : E →ₐ[F] Fv F v) (L : Submodule (𝒪 E) (Fin n → E)) [Submodule.IsLattice E L] (hL : IsSelfDualAt σ J L v) (Q : Subgroup (GLoc σ J v)) (B : GL (Fin n) (Fv F v)) (lv : Fin n → ℕ) (hQ : ∀ g : GLoc σ J v, (B⁻¹ * splitHom σ J τ g * B).1.BlockTriangular lv → g ∈ Q) (y : GLoc σ J v) : ∃ q k : GLoc σ J v, q ∈ Q ∧ k ∈ stabLoc σ J L v ∧ y = q * k

theorem **iwasawa_stab_split** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (v : FinPl D), Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicStabiliserSplitPlaceStatement

theorem **_root_.GlobalTheory.T.iwasawa_isotropic_stabiliser_split_place** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (v : FinPl D), Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicStabilizerInertPlace

theorem **exists_minor_ne_zero** [Nontrivial A] {X : Matrix (Fin n) (Fin r) A} {P : Matrix (Fin r) (Fin n) A} (h : P * X = 1) : ∃ f : Fin r → Fin n, (X.submatrix f id).det ≠ 0

theorem **exists_frame** (O : Subring A) {K : Type} [Field K] (V : ValuationSubring K) (φ : A ≃+* K) (hφ : ∀ y, y ∈ O ↔ φ y ∈ V) {X : Matrix (Fin n) (Fin r) A} {P : Matrix (Fin r) (Fin n) A} (h : P * X = 1) : ∃ (C : Matrix (Fin n) (Fin r) O) (c : GL (Fin r) A), (∃ L : Matrix (Fin r) (Fin n) O, L * C = 1) ∧ X = C.map O.subtype * c.1

theorem **isLocalRing_of_equiv** (O : Subring A) {K : Type} [Field K] (V : ValuationSubring K) (φ : A ≃+* K) (hφ : ∀ y, y ∈ O ↔ φ y ∈ V) : IsLocalRing O

theorem **exists_stab** (O : Subring A) {K : Type} [Field K] (V : ValuationSubring K) (φ : A ≃+* K) (hφ : ∀ y, y ∈ O ↔ φ y ∈ V) {σ : A →+* A} (hσ : ∀ x ∈ O, σ x ∈ O) (hs : ∀ x, σ (σ x) = x) {τ : A} (hτO : τ ∈ O) (hτ : τ + σ τ = 1) (hN : ∀ u ∈ O, σ u = u → (∃ b ∈ O, u * b =

1) → ∃ x ∈ O, σ x ∗ x = u) {J : Matrix (Fin n) (Fin n) A} (hJ : cT σ J = J) (hu : IsUnit J) {Λ : Submodule O (Fin n → A)} (hΛ : ∀ y, y ∈ Λ ↔ ∀ x ∈ Λ, hform σ J x y ∈ O) {a : GL (Fin n) A} (ha : ∀ y, (∀ i, y i ∈ O) ↔ a.1 ∗ᵥ y ∈ Λ) {X : Matrix (Fin n) (Fin r) A} {P : Matrix (Fin r) (Fin n) A} (hP : P ∗ X = 1) (hX : cT σ X ∗ J ∗ X = 0) {g : GL (Fin n) A} (hg : g ∈ UJ σ J) : ∃ k ∈ UJ σ J, k ∈ stabGL Λ ∧ ∃ β : GL (Fin r) A, (g ∗ k⁻¹).1 ∗ X = X ∗ β.1

theorem **tensNU_mul_inclR** (j : R →ₗ[F] R') (hj : ∀ a b, j (a ∗ b) = j a ∗ j b) (y : R ⊗[F] E) (x : E) : tensNU j hj (y ∗ inclR R x) = tensNU j hj y ∗ inclR R' x

theorem **glue_mul_rect** {A B : Type} [CommRing A] [CommRing B] (j : B →ₙ+* A) {M : Matrix (Fin n) (Fin n) B} {β : Matrix (Fin r) (Fin r) B} {X : Matrix (Fin n) (Fin r) B} {X' : Matrix (Fin n) (Fin r) A} (hj : ∀ y a b, j (y ∗ X a b) = j y ∗ X' a b) (h : M ∗ X = X ∗ β) : glue j 1 M ∗ X' = X' ∗ glue j 1 β

theorem **extH_stab** (j : R →ₗ[F] R') (hj : ∀ a b, j (a ∗ b) = j a ∗ j b) (e : Matrix (Fin n) (Fin r) E) (g : HA R σ J) (h : ∃ β : GL (Fin r) (R ⊗[F] E), (HA.mat R σ J g).1 ∗ e.map (inclR R) = e.map (inclR R) ∗ β.1) : ∃ β : GL (Fin r) (R' ⊗[F] E), (HA.mat R' σ J (extH σ J j hj g)).1 ∗ e.map (inclR R') = e.map (inclR R') ∗ β.1

theorem **exists_of_inert** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {σ : E →ₐ[F] E} (hσ : IsCMExt F E σ) {c : 𝒪 E} (hc : (c : E) + σ c = 1) {n r : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : IsHermND σ J) {L : Submodule (𝒪 E) (Fin n → E)} [Submodule.IsLattice E L] {v : HeightOneSpectrum (𝒪 F)} {w : HeightOneSpectrum (𝒪 E)} (hw : Formalism.UG.IsInertAt F E v w) (hv : GlobalTheory.IndefiniteRowSideFullHecke.UnitaryLatticeOrthogonal.UnramAt F E v) (hL : IsSelfDualAt σ J L v) {e : Matrix (Fin n) (Fin r) E} {p : Matrix (Fin r) (Fin n) E} (hp : p ∗ e = 1) (he : Formalism.UG.cT (σ : E →+* E) e ∗ J ∗ e = 0) (y : GLoc σ J v) : ∃ q k : GLoc σ J v, (∃ β : GL (Fin r) (AF F ⊗[F] E), (HA.mat (AF F) σ J (emb σ J v q)).1 ∗ e.map (inclR (AF F)) = e.map (inclR (AF F)) ∗ β.1) ∧ k ∈ stabLoc σ J L v ∧ y = q ∗ k

theorem **iwasawa_inert** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (v : FinPl D), ¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.IwasawaIsotropicStabilizerInertPlaceStatement

theorem **_root_.GlobalTheory.T.iwasawa_isotropicStabilizer_inertPlace** : ∀ (D : D7) (T T₀ : HSp D) (r : ℕ) (d : HypDec T T₀ r) (v : FinPl D), ¬ Formalism.MainNotions.SchurIrreducible.IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JZeroMulMemLatticeIff

theorem **supp_of_colMul** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (L : Lat S) (v : Pl D) (i : Fin ρ) (a : AE D) (h : colMul d.jX i a ∈ latV S L (· = v)) : a ∈ idAt D (· = v)

theorem **idV_supp** (a : FI D) (v : Pl D) {a : AE D} (h : a ∈ idV a v) : a ∈ idAt D (· = v)

theorem **j_zero_mul_mem_lattice_iff** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (L : Lat S) (M : Lat S₀) (v : Pl D) (hL : MonoAt d L M v) (w : Fin S₀.rank → AE D) : d.j0 ∗ᵥ w ∈ latV S L (· = v) ↔ w ∈ latV S₀ M (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiActionsMatchThroughLinkMap

theorem **gram_zEll** (T : HSp D) (ρ : ℕ) (z : Fin T.rank → D.host.E) : adj (zEll T ρ z) · (gA T · zEll T ρ z) = incA D (star z ·ᵥ T.gram ∗ᵥ z) • vecMulVec (fun i => cA D (ell D ρ i)) (ell D ρ)

theorem **gA_jY** : gA W · d.jY = adj d.pX

theorem **ptT_eq** (L : HSp D) (y : Mx D L.rank ρ) (n : GA D W) : ptT d L y n = GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj L W₀ (y · (d.pX · (mt W n · d.j0)))

theorem **trace_inverseGram_conj_smul_eq** {R : Type} [CommRing R] {m k : ℕ} (s t : R) (G G' : Matrix (Fin m) (Fin m) R) (hG : G' ∗ G = 1) (y : Matrix (Fin m) (Fin k) R) (y' : Matrix (Fin k) (Fin m) R) (M : Matrix (Fin k) (Fin k) R) : (G' ∗ ((t • (G ∗ y)) ∗ (M ∗ (s • (y' ∗ G))))).trace = (((t ∗ s) • (y' ∗ (G ∗ y))) ∗ M).trace

theorem **hO_jY** (L : HSp D) (n : GA D W) (η : Mx D ρ L.rank) : hO L W (d.jY · η) (mt W n⁻¹ · (d.jY · η)) = (giA L · (adj η · ((d.pX · (mt W n · d.jY)) · η))).trace

theorem **cenT_eq** (L : HSp D) (y : Mx D L.rank ρ) (n : GA D W) : cenT d L y n = trHalf D ((incA D (D.host.xi0 ∗ star D.host.xi0)⁻¹ • (adj y · (gA L · y)) · (d.pX · (mt W n · d.jY))).trace)

theorem **neg_cenS_eq** (T : HSp D) (x : Mx D T.rank ρ) (n : GA D W) : −cenS d T x n = trHalf D (((adj x · (gA T · x)) · (d.pX · (mt W n · d.jY))).trace)

theorem **link_point** (D : D7) (ρ σ : ℕ) (B : BesLev D σ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E) (_ : IsLineVec L' ζ') (l : Link B L' ζ') (W W₀ : HSp D) (d : HypDec W W₀ ρ) (n : GA D W) (_ : n ∈ flagN d) : l.κ W₀ (ptS d L' (zEll L' ρ ζ') n) = ptT d B.L (yZero B.L ρ B.ζ) n ∧ cenT d B.L (yZero B.L ρ B.ζ) n = −cenS d L' (zEll L' ρ ζ') n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiActionsMatchThroughLinkMapStatement

theorem **_root_.GlobalTheory.T.jacobiActions_match_through_linkMap** : ∀ (D : D7) (ρ σ : ℕ) (B : BesLev D σ) (L' : HSp D) (ζ' : Fin L'.rank → D.host.E), IsLineVec L' ζ' → ∀ (l : Link B L' ζ') (W W₀ : HSp D) (d : HypDec W W₀ ρ), ∀ n ∈ flagN d, l.κ W₀ (ptS d L' (zEll L' ρ ζ') n) = ptT d B.L (yZero B.L ρ B.ζ) n ∧ cenT d B.L (yZero B.L ρ B.ζ) n = −cenS d L' (zEll L' ρ ζ') n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselFlatIntertwinesWeilAction

theorem **besselAnisotropicEmbedding_fixes_basePoint** {σ : ℕ} (B : BesLev D σ) (h : GA D B.A) : mt B.V (embA B.d B.o h) · x0 B = x0 B

theorem **continuous_psiUni** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) (ρ : ℕ) : Continuous fun u : UX D ρ => psiUni ψ u.1

theorem **sFinite_of_haar** {G : Type} [MeasurableSpace G] [Group G] [TopologicalSpace G] (hl : LocallyCompactSpace G) (hs : SecondCountableTopology G) (μ : Measure G) (hμ : μ.IsHaarMeasure) : SFinite μ

theorem **flatFB_om** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (Ψ : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V Ψ → ∀ (g : GA D J.W₀) (h : GA D B.A), flatFB s (s.om (J.d.emb0 g, embA B.d B.o h) Ψ) = s.om₀ (g, B.o.emb (h, 1)) (flatFB s Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselFlatIntertwinesWeilActionStatement

theorem **_root_.GlobalTheory.T.jacobiToBesselFlat_intertwines_weilAction** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (Ψ : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V Ψ → ∀ (g : GA D J.W₀) (h : GA D B.A), flatFB s (s.om (J.d.emb0 g, embA B.d B.o h) Ψ) = s.om₀ (g, B.o.emb (h, 1)) (flatFB s Ψ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselFlatMapSupportProof

theorem **mulVec_latA** {S T : HSp D} {L : Lat S} {L' : Lat T} {P : Pl D → Prop} (M : Mx D T.rank S.rank) (hM : ∀ y ∈ latV S L P, M *ᵥ y ∈ latV T L' P) {z : Fin S.rank → AE D} (hz : z ∈ latA S L P) : M *ᵥ z ∈ latA T L' P

theorem **latA_supp** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latA S L P) (i : Fin S.rank) : y i ∈ idAt D P

theorem **smul_latV** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {w : AE D} (hw : finPart D w ∈ intAE D) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : w • y ∈ latV S L P

theorem **smul_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {w : AE D} (hw : finPart D w ∈ intAE D) {y : Fin S.rank → AE D} (hy : y ∈ latA S L P) : w • y ∈ latA S L P

theorem **smul_latO** {S T : HSp D} {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {P : Pl D → Prop} {w : AE D} (hw : finPart D w ∈ intAE D) {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) P) : w • u ∈ latO S T (LS : Set _) (LT : Set _) P

theorem **mxAt_mul_left** {P : Pl D → Prop} {l m n : ℕ} (A : Mx D l m) {x : Mx D m n} (hx : MxAt P x) : MxAt P (A · x)

theorem **mxAt_mul_right** {P : Pl D → Prop} {l m n : ℕ} {x : Mx D l m} (hx : MxAt P x) (A : Mx D m n) : MxAt P (x · A)

theorem **mxAt_smul** {P : Pl D → Prop} {m n : ℕ} (c : AE D) {x : Mx D m n} (hx : MxAt P x) : MxAt P (c • x)

theorem **sandwich_latO** {S T : HSp D} {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {P : Pl D → Prop} (A : Mx D T.rank T.rank) (hA : ∀ y ∈ latV T LT P, A *ᵥ y ∈ latV T LT P) (B : Mx D S.rank S.rank) (hB : ∀ y ∈ latV S LS P, B *ᵥ y ∈ latV S LS P) {u : WO S T} (hu : u ∈ latO S T (LS : Set _) (LT : Set _) P) : A · (u · B) ∈ latO S T (LS : Set _) (LT : Set _) P

theorem **eps_mul** {v : Pl D} {a : AE D} (ha : a ∈ idAt D (· = v)) : eps v * a = a

theorem **eps_idAt** (v : Pl D) : eps v ∈ idAt D (· = v)

theorem **int_of_idem** {e : AE D} (he : e * e = e) : finPart D e ∈ intAE D

theorem **eps_int** (v : Pl D) : finPart D (eps v) ∈ intAE D

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **exists_parts** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {v : Pl D} (hv : Tame D v) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT (· = v) f) : ∃ a b : AE D, a + b = 1 ∧ ∀ u ∈ latO S T (LS : Set _) (LT : Set _) (· = v), heisO ψ S T (a • u) f = f ∧ heisO ψ S T (b • u) f = f

theorem **exists_parts_om** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {LS : Lat S} {LT : Lat T} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) {v : Pl D} (hv : Tame D v) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT (· = v) f) (g : GA D T) : ∃ a b : AE D, a + b = 1 ∧ ∀ u ∈ latO S T (LS : Set _) (LT : Set _) (· = v), heisO ψ S T (a • (mt T g · u)) (om (1, g) f) = om (1, g) f ∧ heisO ψ S T (b • (mt T g · u)) (om (1, g) f) = om (1, g) f

theorem **colMul_mul** {m n : ℕ} (x : Mx D m n) (i : Fin n) (c a : AE D) : colMul x i (c * a) = c • colMul x i a

theorem **single_mulVec'** {m n l : ℕ} (x : Mx D m n) (i : Fin n) (j : Fin l) (s : AE D) (y : Fin l → AE D) : (x · (Matrix.single i j s : Mx D n l)) *ᵥ y = colMul x i (s * y j)

theorem **mulVec_colMul** {l m n : ℕ} (A : Mx D l m) (x : Mx D m n) (i : Fin n) (a : AE D) : A *ᵥ colMul x i a = colMul (A · x) i a

theorem **pX_colMul** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (i : Fin ρ) (a : AE D) : (d.pX *ᵥ colMul d.jX i a) i = a

theorem **idAt_of_colMul** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {P : Pl D → Prop} {i : Fin ρ} {a : AE D} (h : colMul d.jX i a ∈ latV S L P) : a ∈ idAt D P

theorem **psi_parts** (ψ : AddChar (AF D.host.F) ℂ) {a b : AE D} (hab : a + b = 1) {s : AE D} (ha : psiE ψ (a * s) = 1) (hb : psiE ψ (b * s) = 1) : psiE ψ s = 1

theorem **hO_extX** {V V₀ W : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (x : Mx D W.rank ρ) (u : WO V W) : hO V W (dV.extX W x) u = pairS W x (u · dV.jY)

theorem **smul_eq_self** {α : Type} {F : α → ℂ} (hF : F ≠ 0) {c : ℂ} (h : F = c • F) : c = 1

theorem **dual_col** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {V V₀ W : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) {LV : Lat V} {LV₀ : Lat V₀} {LW : Lat W} (hV : IsSelfDualC V LV) (hW : IsSelfDualC W LW) {v : Pl D} (hm : MonoAt dV LV LV₀ v) (x : Mx D W.rank ρ) {β : Type} (𝔉 : (WO V W → ℂ) → β → ℂ) (G : WO V W → ℂ) (hne : 𝔉 G ≠ 0) (hdual : ∀ a : Mx D W.rank ρ, MxAt (· = v) a → UInv W x a → 𝔉 (heisO ψ V W (dV.extY W a) G) = ψ (symplO V W (dV.extX W x) (dV.extY W a)) • 𝔉 G) (g : GA D W) {a b : AE D} (hab : a + b = 1) (hfix : ∀ u ∈ latO V W (LV : Set _) (LW : Set _) (· = v), heisO ψ V W (a • (mt W g · u)) G = G ∧ heisO ψ V W (b • (mt W g · u)) G = G) (p p' : Mx D W.rank W.rank) (hp : adj p' · gA W = gA W · p) (hp' : ∀ y ∈ latV W LW (· = v), p' *ᵥ y ∈ latV W LW (· = v)) (i : Fin ρ) (hi : ∀ j < i, ∀ k, (p · (mt W g⁻¹ · x)) k j = 0) (c : AE D) (hc : colMul dV.jX i c ∈ latV V LV (· = v)) : c

theorem **shift_one** {ψ : AddChar (AF D.host.F) ℂ} {V V₀ W : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) {Λ : AddSubgroup (WO V W)} {v : Pl D} (x : Mx D W.rank ρ) {β : Type} (𝔉 : (WO V W → ℂ) → β → ℂ) (G : WO V W → ℂ) (hne : 𝔉 G ≠ 0) (hsh : ∀ N : Mx D ρ ρ, MxAt (· = v) N → (∀ i j, j ≤ i → N i j = 0) → 𝔉 (heisO ψ V W (dV.extX W (x · N)) G) = psiE ψ (upSum N) • 𝔉 G) (g : GA D W) {a b : AE D} (hab : a + b = 1) (hfix : ∀ u ∈ Λ, heisO ψ V W (a • (mt W g · u)) G = G ∧ heisO ψ V W (b • (mt W g · u)) G = G) (i j : Fin ρ) (hij : j.1 = i.1 + 1) (s : AE D) (hs : s ∈ idAt D (· = v)) (hu : dV.extX W (mt W g⁻¹ · x · (Matrix.single i j s : Mx D ρ ρ)) ∈ Λ) : psiE ψ s = 1

theorem **eps_mem_of_products** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} (hL : IsSelfDualC S L) {v : Pl D} (hm : MonoAt d L L₀ v) (i : Fin ρ) (H : AddSubgroup (AE D)) (hH : ∀ α β : AE D, colMul d.jX i α ∈ latV S L (· = v) → (∀ a, colMul d.jX i a ∈ latV S L (· = v) → β * a ∈ intAt D (· = v)) → α * β ∈ H) : eps v ∈ H

theorem **int_of_stable** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} (hL : IsSelfDualC S L) {v : Pl D} (hm : MonoAt d L L₀ v) (i : Fin ρ) {c : AE D} (hc1 : c - 1 ∈ idAt D (· = v)) (hc : ∀ a, colMul d.jX i a ∈ latV S L (· = v) → colMul d.jX i (c * a) ∈ latV S L (· = v)) : finPart D c ∈ intAE D

theorem **stable_of_prim** {ψ : AddChar (AF D.host.F) ℂ} {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} (hL : IsSelfDualC S L) {v : Pl D} (hm : MonoAt d L L₀ v) (hp : PrimAt ψ d L v) (i j : Fin ρ) (hij : j.1 = i.1 + 1) {c : AE D} (hc : ∀ s ∈ idAt D (· = v), (∀ a, colMul d.jX j a ∈ latV S L (· = v) → colMul d.jX i (s * a) ∈ latV S L (· = v)) → psiE ψ (c * s) = 1) (a : AE D) (ha : colMul d.jX i a ∈ latV S L (· = v)) : colMul d.jX i (c * a) ∈ latV S L (· = v)

theorem **diag_mulVec** {ρ : ℕ} {t : Mx D ρ ρ} (ht : ∀ i j, i ≠ j → t i j = 0) (y : Fin ρ → AE D) (i : Fin ρ) : (t *ᵥ y) i = t i i * y i

theorem **mul_diag** {m ρ : ℕ} (x : Mx D m ρ) {t : Mx D ρ ρ} (ht : ∀ i j, i ≠ j → t i j = 0) (k : Fin m) (i : Fin ρ) : (x · t) k i = x k i * t i i

theorem **inv_diag_mul** {ρ : ℕ} {t : GL (Fin ρ) (AE D)} (ht : ∀ i j, i ≠ j → t.1 i j = 0) (i : Fin ρ) : (t⁻¹).1 i i * t.1 i i = 1

theorem **inv_diag** {ρ : ℕ} {t : GL (Fin ρ) (AE D)} (ht : ∀ i j, i ≠ j → t.1 i j = 0) (i j : Fin ρ) (h : i ≠ j) : (t⁻¹).1 i j = 0

theorem **levi_entry** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {P : Pl D → Prop} {t : GL (Fin ρ) (AE D)} (ht : d.levi t ∈ grpAt S P) (i : Fin ρ) : t.1 i i - 1 ∈ idAt D P

theorem **levi_mulVec_mem** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} {v : Pl D} (hm : MonoAt d L L₀ v) {t : GL (Fin ρ) (AE D)} (ht : ∀ i j, i ≠ j → t.1 i j = 0) (h1 : ∀ i, finPart D (t.1 i i) ∈ intAE D) (h2 : ∀ i, finPart D ((t⁻¹).1 i i) ∈ intAE D) {y : Fin S.rank → AE D} (hy : y ∈ latV S L (· = v)) : mt S (d.levi t) *ᵥ y ∈ latV S L (· = v)

theorem **inK_levi** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {L : Lat S} {L₀ : Lat S₀} {v : Pl D} (hm : MonoAt d L L₀ v) {t : GL (Fin ρ) (AE D)} (ht : ∀ i j, i ≠ j → t.1 i j = 0) (hg : d.levi t ∈ grpAt S (· = v)) (h1 : ∀ i, finPart D (t.1 i i) ∈ intAE D) (h2 : ∀ i, finPart D ((t⁻¹).1 i i) ∈ intAE D) : InK S L (· = v) (d.levi t)

theorem **inK_mul** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a b : GA D S} (ha : InK S L P a) (hb : InK S L P b) : InK S L P (a * b)

theorem **mul_xLast_castSucc** {T T₀ : HSp D} {σ m : ℕ} (d : HypDec T T₀ σ) (z : Fin T₀.rank → D.host.E) (M : Mx D m T.rank) (k : Fin m) (i : Fin σ) : (M · xLast d z) k i.castSucc = (M · d.jX) k i

theorem **mul_xLast_last** {T T₀ : HSp D} {σ m : ℕ} (d : HypDec T T₀ σ) (z : Fin T₀.rank → D.host.E) (M : Mx D m T.rank) (k : Fin m) : (M · xLast d z) k (Fin.last σ) = ((M · d.j0) *ᵥ ratV z) k

theorem **xLast_castSucc** {T T₀ : HSp D} {σ : ℕ} (d : HypDec T T₀ σ) (z : Fin T₀.rank → D.host.E) (k : Fin T.rank) (i : Fin σ) : xLast d z k i.castSucc = d.jX k i

theorem **xLast_last** {T T₀ : HSp D} {σ : ℕ} (d : HypDec T T₀ σ) (z : Fin T₀.rank → D.host.E) (k : Fin T.rank) : xLast d z k (Fin.last σ) = (d.j0 *ᵥ ratV z) k

theorem **support_parts** {ψ : AddChar (AF D.host.F) ℂ} {σ : ℕ} {J : JacLev D (σ + 1)} {B : BesLev D σ} (s : StepFB ψ J B) {LV : Lat J.W} {LV₀ : Lat J.W₀} {LW : Lat B.V} {LW₀ : Lat B.V₀} {v : Pl D} (hA : AdaptFB ψ J B LV LV₀ LW LW₀ v) {𝔉 : (WO J.W B.V → ℂ) → (WO J.W₀ B.V₀ → ℂ)} (h𝔉 : IsFlatFB s (· = v) 𝔉) (hsh : ShiftLaw ψ J.d (xLast B.d (vecB B.o B.ζ)) (· = v) 𝔉) {f : WO J.W B.V → ℂ} (hf : Vac ψ J.W B.V LV LW (· = v) f) {t : GL (Fin σ) (AE D)} (ht : ∀ i j, i ≠ j → t.1 i j = 0) (htG : B.d.levi t ∈ grpAt B.V (· = v)) (γ : GA D B.V₀) (hne : 𝔉 (s.om (1, B.d.levi t * B.d.emb0 γ) f) ≠ 0) : InK B.V LW (· = v) (B.d.levi t) ∧ ∀ a : AE D, a • ratV (vecB B.o B.ζ) ∈ latA B.V₀ LW₀ (· = v) → mt B.V₀ γ⁻¹ *ᵥ (a • ratV (vecB B.o B.ζ)) ∈ latA B.V₀ LW₀ (· = v)

theorem **supportShift** : ∀ D : D7, GlobalTheory.FinPl.SupportShift D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselFlatMapSupportProofStatement

theorem **_root_.GlobalTheory.T.jacobiToBesselFlatMapSupport_proof** : ∀ D : D7, GlobalTheory.FinPl.SupportShift D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselFlatSchwartzAddSmul

theorem **mixS_add** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) {μ : Measure (Mx D T.rank ρ)} {𝓕 : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ 𝓕) {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) (x : Mx D T.rank ρ) : d.mixS T ψ μ 𝓕 (f + g) x = d.mixS T ψ μ 𝓕 f x + d.mixS T ψ μ 𝓕 g x

theorem **continuous_mixS** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (T : HSp D) {μ : Measure (Mx D T.rank ρ)} {𝓕 : Set (Mx D T.rank ρ)} (hμ : IsCovolOne (ratMx D T.rank ρ) μ 𝓕) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (x : Mx D T.rank ρ) : Continuous (d.mixS T ψ μ 𝓕 f x)

theorem **integrableOn_mixT_of_quasi** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) {μ : Measure (Mx D ρ S.rank)} {𝓕 : Set (Mx D ρ S.rank)} (hμ : IsCovolOne (ratMx D ρ S.rank) μ 𝓕) {f : WO S T → ℂ} (hc : Continuous f) (hq : ∀ γ w, f (ratW T S γ + w) = ψ (-(symplOH S T (ratW T S γ) w)) * f w) (y : Mx D S.rank ρ) (w₀ : WO S T₀) : IntegrableOn (fun ξ => heisO ψ S T (d.incX S ξ) (heisO ψ S T (d.incAdj S y) (heisO ψ S T (d.inc0 S w₀) f)) 0) 𝓕 μ

theorem **evalAt_add** {ψ : AddChar (AF D.host.F) ℂ} (hψ : Continuous ψ) {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (vS : VolM D W.rank ρ) (vT : VolM D σ V₀.rank) {f g : WO V W → ℂ} (hf : IsThetaFunO ψ V W f) (hg : IsThetaFunO ψ V W g) (x : Mx D W.rank ρ) (y : Mx D V₀.rank σ) : evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 (f + g) x y = evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 f x y + evalAt ψ dV dW vS.μ vS.𝓕 vT.μ vT.𝓕 g x y

theorem **flatFB_add_smul** : ∀ (D : D7), FlatLaw D → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (f g : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V f → IsThetaFunO ψ J.W B.V g → flatFB st (f + g) = flatFB st f + flatFB st g ∧ ∀ a : ℂ, flatFB st (a • f) = a • flatFB st f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselFlatSchwartzAddSmulStatement

theorem **_root_.GlobalTheory.T.jacobiToBesselFlatSchwartz_add_smul** : ∀ (D : D7), FlatLaw D → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (f g : WO J.W B.V → ℂ), IsThetaFunO ψ J.W B.V f → IsThetaFunO ψ J.W B.V g → flatFB st (f + g) = flatFB st f + flatFB st g ∧ ∀ a : ℂ, flatFB st (a • f) = a • flatFB st f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselSecondStepFlatSupportRamified

theorem **suppSecond_holds** : ∀ D : D7, SuppSecond D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselSecondStepFlatSupportRamifiedStatement

theorem **_root_.GlobalTheory.T.jacobiToBessel_second_step_flat_support_ramified** : ∀ D : D7, GlobalTheory.FinPl.SuppSecond D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselUnfoldedIntegrandIntegrable

theorem **quasiFam_bdd_on_compact** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S L : HSp D} (v : VolM D L.rank S.rank) {Y : GA D S → WO S L → ℂ} (hY : IsQuasiFam ψ S L Y) {K : Set (GA D S)} (hK : IsCompact K) : ∃ M : ℝ, ∀ h ∈ K, ∀ w, ‖Y h w‖ ≤ M

theorem **unfolded_integrable** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (G : GA D B.V → ℂ) (Φ : WO J.W B.V → ℂ), IsQuasiFam ψ J.W₀ B.L Y → RapidDec B.V G → IsThetaFunO ψ J.W B.V Φ → ∀ 𝓑 : Set (GA D B.V), IsFundamentalDomain (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)) 𝓑 s.vW.μ → Integrable (Function.uncurry fun h₀ h => G h * tFlat s vL (Y h₀) (s.om (J.d.emb0 h₀, h) Φ)) ((J.v0.μ.restrict J.v0.𝓕).prod (s.vW.μ.restrict 𝓑))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JacobiToBesselUnfoldedIntegrandIntegrableStatement

theorem **_root_.GlobalTheory.T.jacobiToBessel_unfoldedIntegrand_integrable** : ∀ (D : D7), Descent D → ∀ (ψ : AddChar (AF D.host.F) ℂ) (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (s : StepFB ψ J B), ∀ (vL : VolM D B.L.rank J.W₀.rank) (Y : GA D J.W₀ → WO J.W₀ B.L → ℂ) (G : GA D B.V → ℂ) (Φ : WO J.W B.V → ℂ), IsQuasiFam ψ J.W₀ B.L Y → RapidDec B.V G → IsThetaFunO ψ J.W B.V Φ → ∀ 𝓑 : Set (GA D B.V),

IsFundamentalDomain (GQ B.V ⊓ besGrp B.d B.o : Subgroup (GA D B.V)) 𝓑 s.vW.μ → Integrable (Function.uncurry fun h₀ h => G h ∗ tFlat s vL (Y h₀) (s.om (J.d.emb0 h₀, h) Φ)) ((J.v0.μ.restrict J.v0.𝓕).prod (s.vW.μ.restrict 𝓑))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JoinSubspaceRealDataExists

theorem **exists_read_of_latTie** : ∀ (D : D7) (G : Type) [Group G] [Fintype G] (Pl0 I0 T : Type) (A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m) (Ks : Type) [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat3 D A3 Ks) (F : Face3 D A3) (n : ℕ), A3.fam3.Ok3 n → ∀ (S0 : IndefData (A3.side3 F.P0 F.ψ0 (R.toCx3 F).zs)) (p0 : BaseLevelRowSide D) (Sp : SPer (F.Dc (R.toCx3 F).zs)) (C : ColData (F.Dc (R.toCx3 F).zs) Sp ((R.toCx3 F).data F.cr true)) (NuC : ∀ u, Submodule (RingOfIntegers D.what₂.LL) (Fin ((F.Dc (R.toCx3 F).zs).a u) → D.what₂.LL)) (I : DivisionCompatibilityDataThreeSided D A3 F.P0 F.ψ0 F.P1 F.ψ1 (R.toCx3 F).zs S0 ((R.toCx3 F).data F.cr true) p0 (F.Dc (R.toCx3 F).zs)), DivDef3 D A3 F.P0 F.ψ0 F.P1 F.ψ1 (R.toCx3 F).zs S0 ((R.toCx3 F).data F.cr true) p0 C NuC I → ∃ Rd : Read D (R.toCx3 F) F.fc (twAt D.what₂.embLL A3.A.v0), HEq Rd.wU I.wU ∧ HEq Rd.eNu I.eNu ∧ HEq Rd.eRR I.eR

theorem **kernelJoin_thetaSurjective_goodPart_latticeCondition** : ∀ (D : D7) (G : Type) [Group G] [Fintype G] (Pl0 I0 T : Type) (A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m) (Ks : Type) [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat3 D A3 Ks) (F : Face3 D A3) (n : ℕ), A3.fam3.Ok3 n → A3.fam3.TypesSep3 → R.WeakCharacterisation F → PrimRows3 R F → SchurAll D → (D.sp.Y 0).SigOK → ∀ (S0 : IndefData (A3.side3 F.P0 F.ψ0 (R.toCx3 F).zs)) (p0 : BaseLevelRowSide D) (C : ColData (F.Dc (R.toCx3 F).zs) (F.Sp (R.toCx3 F).zs) ((R.toCx3 F).data F.cr true)) (NuC : ∀ u, Submodule (RingOfIntegers D.what₂.LL) (Fin ((F.Dc (R.toCx3 F).zs).a u) → D.what₂.LL)) (I : DivisionCompatibilityDataThreeSided D A3 F.P0 F.ψ0 F.P1 F.ψ1 (R.toCx3 F).zs S0 ((R.toCx3 F).data F.cr true) p0 (F.Dc (R.toCx3 F).zs)), DivDef3 D A3 F.P0 F.ψ0 F.P1 F.ψ1 (R.toCx3 F).zs S0 ((R.toCx3 F).data F.cr true) p0 C NuC I → H

theorem **core_latTie** {G : Type} [Group G] [Fintype G] {Pl0 I0 T : Type} {A3 : FamIn3 D.what₂.LL G Pl0 I0 (List ℕ) T D.R.m} (Rl : RealDat3 D A3 Ks) (F : Face3 D A3) (n : ℕ) (hOk : A3.fam3.Ok3 n) (hSep : A3.fam3.TypesSep3) (hCh : Rl.WeakCharacterisation F) (hPrim : PrimRows3 Rl F) (hSchur : SchurAll D) (hSig : (D.sp.Y 0).SigOK) (Zs : DefSub A3.fam3.amb3) (hzs : (Rl.toCx3 F).zs = Zs) (S0 : IndefData (A3.side3 F.P0 F.ψ0 Zs)) (S1 : IndefData (A3.side3 F.P1 F.ψ1 Zs)) (hs1 : HEq ((Rl.toCx3 F).data F.cr true) S1) (Dc : DefCols (A3.side3 F.P1 F.ψ1 Zs)) (hdc : F.Dc Zs = Dc) (Sp : SPer Dc) (hsp : HEq (F.Sp Zs) Sp) (p0 : BaseLevelRowSide D) (C : ColData Dc Sp S1) (NuC : ∀ u, Submodule (RingOfIntegers D.what₂.LL) (Fin (Dc.a u) → D.what₂.LL)) (I : DivisionCompatibilityDataThreeSided D A3 F.P0 F.ψ0 F.P1 F.ψ1 Zs S0 S1 p0 Dc) (hDef : DivDef3 D A3 F.P0 F.ψ0 F.P1 F.ψ1 Zs S0 S1 p0 C NuC I) (hLat : HLatTie (F

theorem **join_subspace_lattice_condition** (Q : RealT R P Dcf Spf Zscf S Ks) (hPin : RealPins Q) (hSee : SeeEx Q) (n : ℕ) (hOk : P.A3.fam3.Ok3 n) (hSep : P.A3.fam3.TypesSep3) (hPrim : PrimRows3 Q.Rl Q.F) (hLat : HLatTie (Formalism.BasicNotions.RowSide.sharp D).RR (P.A3.indefDown P.S1).RR (P.A3.indefDown P.S1).XE P.I.eRR) (hDc : RowsHoldDcPt3 P) (hSchur : SchurAll D) (hSig : (D.sp.Y 0).SigOK) (hJ : GlobalTheory.Mem.JTop.JoinPin Q) : Nonempty (Formalism.L2.SG.JoinSI.JoinSI R S Q) ∧ (D.what₂.V0 D.what₂.mu0).map (D.what₂.iota D.what₂.mu0) = ⊤

theorem **join_subspace_real_data_lattice** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (K : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl K Yd cX) (x : TwIn ex) (Rc : Formalism.L2.RecT ex) (P : SPack D fun u => (Rc.T.cols.forms u).to0) (S : SRest P) (y : RealIn x Rc (S.toSTiesR Rc)) (c : ℝ), ∀ n : ℕ, SchurAll D → RealPins (realTOf y) → Holds x.R x.F → x.A3.fam3.Ok3 n → x.A3.fam3.TypesSep3 → HLatTie (Formalism.BasicNotions.RowSide.sharp D).RR (x.A3.indefDown (divisionPairOfAdelicFormsFaceColumnsData x).S1).RR (x.A3.indefDown (divisionPairOfAdelicFormsFaceColumnsData x).S1).XE (divisionPairOfAdelicFormsFaceColumnsData x).I.eRR → RowsHoldDcPt3 (divisionPairOfAdelicFormsFaceColumnsData x) → SeeAt x.R.toRealDat x.F.lv2 P.X P.inst y.tS → EvalTieT (MP x.R.toRealDat) x.R.toRea

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JoinSubspaceRealDataExistsStatement

theorem **_root_.GlobalTheory.T.join_subspace_real_data_exists** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (K : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl K Yd cX) (x : TwIn ex) (Rc : Formalism.L2.RecT ex) (P : SPack D fun u => (Rc.T.cols.forms u).to0) (S : SRest P) (y : RealIn x Rc (S.toSTiesR Rc)) (c : ℝ), ∀ n : ℕ, SchurAll D → RealPins (realTOf y) → Holds x.R x.F → x.A3.fam3.Ok3 n → x.A3.fam3.TypesSep3 → HLatTie (Formalism.BasicNotions.RowSide.sharp D).RR (x.A3.indefDown (divisionPairOfAdelicFormsFaceColumnsData x).S1).RR (x.A3.indefDown (divisionPairOfAdelicFormsFaceColumnsData x).S1).XE (divisionPairOfAdelicFormsFaceColumnsData x).I.eRR → RowsHoldDcPt3 (divisionPairOfAdelicFormsFaceColumnsData x) → SeeAt x.R.toRealDat x.F.lv2 P.X P.inst y.tS → EvalTieT (MP x.R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JointHarmonicsLowestKTypeCoefficientReal

theorem **l2_self_eq_setIntegral_norm_sq** {X : Type} [MeasurableSpace X] (μ : Measure X) (𝒳 : Set X) (f : X → ℂ) : l2 μ 𝒳 f f = ((∫ w in 𝒳, ‖f w‖ ^ 2 ∂μ : ℝ) : ℂ)

theorem **fockIp_eq_fock** {ι : Type} (P Q : MvPolynomial ι ℂ) : GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.fockIp P Q = Formalism.E.fock P Q

theorem **fock_coef_real** : GlobalTheory.IndefiniteRowSideFullHecke.MatrixMinorProducts.HarmonicModule → ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.PlaceModels D → ∀ i t σ s P cμ cη m ω BW f₀ f c io, IsLiftDataC R F i t σ s P cμ cη m ω BW f₀ f c io → ∀ (μ : Measure (WO (Vc D i) (Wc D i))) (𝒳 : Set (WO (Vc D i) (Wc D i))), IsCovolOne (ratL (Wc D i) (Vc D i)) μ 𝒳 → ∀ g : D.what₂.Tau → Matrix (Fin (Wc D i).rank) (Fin (Wc D i).rank) ℂ, (∀ ρ, g ρ ∈ Formalism.QuantumDeterminants.UH (gramAt D (Wc D i) ρ)) → ∀ (e e' : ∀ ρ, lowT (R.MW i σ) ρ) (w w' : c.W), l2 μ 𝒳 (ω (1, realElt R.θ (Wc D i) g) (ptAt R i σ BW f (io (PiTensorProduct.tprod ℂ e) w))) (ptAt R i σ BW f (io (PiTensorProduct

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JointHarmonicsLowestKTypeCoefficientRealStatement

theorem **_root_.GlobalTheory.T.jointHarmonics_lowest_KType_coefficient_real** : GlobalTheory.IndefiniteRowSideFullHecke.MatrixMinorProducts.HarmonicModule → ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.PlaceModels D → ∀ i t σ s P cμ cη m ω BW f₀ f c io, IsLiftDataC R F i t σ s P cμ cη m ω BW f₀ f c io → ∀ (μ : Measure (WO (Vc D i) (Wc D i))) (𝒳 : Set (WO (Vc D i) (Wc D i))), IsCovolOne (ratL (Wc D i) (Vc D i)) μ 𝒳 → ∀ g : D.what₂.Tau → Matrix (Fin (Wc D i).rank) (Fin (Wc D i).rank) ℂ, (∀ ρ, g ρ ∈ Formalism.QuantumDeterminants.UH (gramAt D (Wc D i) ρ)) → ∀ (e e' : ∀ ρ, lowT (R.MW i σ) ρ) (w w' : c.W), l2 μ 𝒳 (ω (1, realElt R.θ (Wc D i) g) (ptAt R i σ BW f (io (PiTensorProduct.t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JointOfFamilyAndService

theorem **joint_of_famAllQ** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), FamAllQ W → ServedQ W → Joint W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JointOfFamilyAndServiceStatement

theorem **_root_.GlobalTheory.T.joint_of_family_and_service** : GlobalTheory.Published.CuspRapidDecay → GlobalTheory.Published.HarishChandraModules → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), FamAllQ W → ServedQ W → Joint W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JointStarProductNonvanishing

theorem **cols_heq** {Df Df' : DefSide Bs} (hDf : Df = Df') {Dc : DefCols Df} {Dc' : DefCols Df'} (hD : HEq Dc Dc') {m : Dc.Mem} {m' : Dc'.Mem} (hm : HEq m m') : Dc.a m = Dc'.a m' ∧ (∀ v, HEq (Dc.Xc m v) (Dc'.Xc m' v)) ∧ ∀ t (x : Fin (Dc.a m) → K) (x' : Fin (Dc'.a m') → K) (F : Fin (Df.nV t) → K) (F' : Fin (Df'.nV t) → K), HEq x x' → HEq F F' → Dc.Pt m t x F = Dc'.Pt m' t x' F'

theorem **cols_mem** {Df Df' : DefSide Bs} (hDf : Df = Df') {Dc : DefCols Df} {Dc' : DefCols Df'} (hD : HEq Dc Dc') (m : Dc.Mem) : ∃ m' : Dc'.Mem, HEq m m'

theorem **heq_fin** {α : Type} {n n' : ℕ} (h : n = n') (x : Fin n → α) : ∃ x' : Fin n' → α, HEq x x'

theorem **heq_fin'** {α : Type} {n n' : ℕ} (h : n = n') (x' : Fin n' → α) : ∃ x : Fin n → α, HEq x x'

theorem **heq_mat** {n n' : ℕ} (h : n = n') (M' : Matrix (Fin n') (Fin n') ℂ) : ∃ M : Matrix (Fin n) (Fin n) ℂ, HEq M M'

theorem **heq_mulVec** {n n' : ℕ} (h : n = n') {M : Matrix (Fin n) (Fin n) ℂ} {M' : Matrix (Fin n') (Fin n') ℂ} (hM : HEq M M') {x : Fin n → ℂ} {x' : Fin n' → ℂ} (hx : HEq x x') : HEq (M.mulVec x) (M'.mulVec x')

theorem **heq_fnorm** {n n' : ℕ} (h : n = n') {Y : Matrix (Fin n) (Fin n) ℂ} {Y' : Matrix (Fin n') (Fin n') ℂ} (hY : HEq Y Y') {x : Fin n → ℂ} {x' : Fin n' → ℂ} (hx : HEq x x') : fnorm Y x = fnorm Y' x'

theorem **heq_cpt** {K : Type} [Field K] (σ : K →+* ℂ) {n n' : ℕ} (h : n = n') {x : Fin n → K} {x' : Fin n' → K} (hx : HEq x x') : HEq (cpt σ x) (cpt σ x')

theorem **heq_cjv** (s : Bool) {n n' : ℕ} (h : n = n') {x : Fin n → ℂ} {x' : Fin n' → ℂ} (hx : HEq x x') : HEq (Formalism.L2.SG.JoinSI.cjv s x) (Formalism.L2.SG.JoinSI.cjv s x')

theorem **dc_heq** (W : RealPack ex) : HEq (W.Q.F.fc (cxW W).zs).Dc W.P.Dc

theorem **df_eq** (W : RealPack ex) : (W.Q.F.fc (cxW W).zs).Df1 = W.P.A3.side3 W.P.Ps W.P.ψs W.P.Zs

theorem **mem_word** (W : RealPack ex) (m : (W.Q.F.fc (cxW W).zs).Dc.Mem) : ∃ u, HEq m (W.P.I.wU u)

theorem **anchor_word** (W : RealPack ex) (z : GA D (D.sp.Y 0)) (u : D.what₂.Word) (d : BotDat W u) (v : (W.Q.Rl.MY 0).Vec) (U : D.what₂.Mu u →ₗ[ℂ] D.what₂.Mu u) (r : ℝ) (c : W.P.T → ℂ) (hR : BotRel W z d) (hA : (∀ φ, ∃ C, IsCol W u C φ) ∧ (∀ T ∈ ThetaLiftSpaceZero W, ∀ C, ∃ b, IsPair W u (z ∗ (show GA D (D.sp.Y 0) from W.Q.tS)) (T (PiTensorProduct.tprod ℂ v)) C b) ∧ ∃ (φ : D.what₂.Mu u) (C : _) (b : ℂ), ∃ T ∈ ThetaLiftSpaceZero W, IsCol W u C φ ∧ IsPair W u (z ∗ (show GA D (D.sp.Y 0) from W.Q.tS)) (T (PiTensorProduct.tprod ℂ v)) C b ∧ b ≠ 0) (hr : 0 < r) (hU : ∀ φ, ‖U φ‖ = r ∗ ‖φ‖) (hRel : d.Rel U c) (hrow : ∀ t (F : Fin ((W.Q.F.fc (cxW W).zs).Df1.nV t) → D.what₂.LL), rowAt W t F ∈ ThetaLiftSpaceZero W) : ∃ (R : D.what₂.Mu u ≃ₗᵢ[ℂ] D.what₂.Mu u) (p b₀ : ℂ), p ≠ 0 ∧ b₀ ≠ 0 ∧ ∀ t (φ : D.what₂.Mu u) (F : Fin ((W.P.A3.side3 W.P.Ps W.P.ψs W.P.Zs).nV t) → D.what₂.LL) (F' : Fin ((W.Q.F.fc (cxW W).

theorem **miStar2_at_realisation** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → (∀ t F, (cxW W).θ t ((cxW W).jR t (W.Q.F.eV (cxW W).zs t F)) ∈ (cxW W).EEl true) → Joint W → PuNZ D → (cxW W).MIstar2 (W.Q.F.eV (cxW W).zs) W.Q.sEE

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.JointStarProductNonvanishingStatement

theorem **_root_.GlobalTheory.T.joint_star_product_nonvanishing** : GlobalTheory.Published.DiscreteSeriesClassification → GlobalTheory.Published.CuspRapidDecay → ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), BotHyp W → (∀ t F, (cxW W).θ t ((cxW W).jR t (W.Q.F.eV (cxW W).zs t F)) ∈ (cxW W).EEl true) → Joint W → PuNZ D → (cxW W).MIstar2 (W.Q.F.eV (cxW W).zs) W.Q.sEE

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KFiniteSpanLeSpanRealIntertwinerValues

theorem **kfin_eq** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) : GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ = kFin M τ

theorem **span_le_range** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (h : KSm M) : Submodule.span ℂ {w : SmT M | ∃ v : M.Vec, (∀ τ, (v τ : M.H τ) ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ) ∧ w = PiTensorProduct.tprod ℂ v} ≤ LinearMap.range (inclK h)

theorem **range_le_span** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (h : KSm M) (y : KT M) : inclK h y ∈ Submodule.span ℂ {w : SmT M | ∃ v : M.Vec, (∀ τ, (v τ : M.H τ) ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ) ∧ w = PiTensorProduct.tprod ℂ v}

theorem **kfinSpan_le_span_it** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (aS aT : D.what₂.Tau → ℚ) (ω : (GA D V × GA D W →* Module.End ℂ (WO V W → ℂ))) (f : WO V W → ℂ) (It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M), IsInfFam θ → IsWeilExp ψ V W aS aT ω → IsVacVec ψ θ M M' f → IsItW θ M M' ω (ptOf ψ θ M M' f) It → (Submodule.span ℂ {w : SmT M | ∃ v : (M).Vec, (∀ τ, (v τ : (M).H τ) ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ) ∧ w = PiTensorProduct.tprod ℂ v} ≤ Submodule.span ℂ {w | ∃ P, ∃ v ∈ (Submodule.span ℂ {w : SmT M' | ∃ v : (M').Vec, (∀ τ, (v τ : (M').H τ) ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M' τ) ∧ w = PiTensorProduct.tprod ℂ v}), w = It P v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KFiniteSpanLeSpanRealIntertwinerValuesStatement

theorem **_root_.GlobalTheory.T.kFiniteSpan_le_span_realIntertwiner_values** : GlobalTheory.Published.DiscreteSeriesCover → GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (θ : InfFam D) (ψ : AddChar (AF D.host.F) ℂ) (aS aT : D.what₂.Tau → ℚ) (ω : (GA D V × GA D W →* Module.End ℂ (WO V W → ℂ))) (f : WO V W → ℂ) (It : PolF D W.rank V.rank →ₗ[ℂ] SmT M' →ₗ[ℂ] SmT M), IsInfFam θ → IsWeilExp ψ V W aS aT ω → IsVacVec ψ θ M M' f → IsItW θ M M' ω (ptOf ψ θ M M' f) It → (Submodule.span ℂ {w : SmT M | ∃ v : (M).Vec, (∀ τ, (v τ : (M).H τ) ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M τ) ∧ w = PiTensorProduct.tprod ℂ v} ≤ Submodule.span ℂ {w | ∃ P, ∃ v ∈ (Submodule.span ℂ {w : SmT M' | ∃ v : (M').Vec, (∀ τ, (v τ : (M').H τ) ∈ GlobalTheory.IndefiniteRowSideFullHecke.ThetaSourceAction.kfin M' τ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KFiniteTranslatesFiniteDimensional

theorem **frameInv_eq_one** {θ : InfFam D} {A : HSp D} {cA : CptF A} {a : GL (Fin A.rank) (AE D)} (hD : Defin θ A cA a) (τ : D.what₂.Tau) : frameInv θ A a τ = 1

theorem **isOpen_Kmax** {θ : InfFam D} {A : HSp D} {cA : CptF A} {a : GL (Fin A.rank) (AE D)} (hD : Defin θ A cA a) : IsOpen (Kmax A a : Set (GA D A))

theorem **finiteDimensional_span_io** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A : HSp D) (cA : CptF A) (a : GL (Fin A.rank) (AE D)), Defin θ A cA a → ∀ (W : HSp D) (χ : (AE D)ˣ →* Circle) (omh : WRep (dblSp A) W), IsWeilDatumO ψ (dblSp A) W 1 χ omh → ∀ φ, IsKFinVec omh (Khat A a) φ → ∀ K : Set (GA D A), IsCompact K → FiniteDimensional ℂ (Submodule.span ℂ {f : WO (dblSp A) W → ℂ | ∃ g₁ ∈ K, ∃ g₂ ∈ K, f = omh (io A (g₁, g₂), 1) φ})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KFiniteTranslatesFiniteDimensionalStatement

theorem **_root_.GlobalTheory.T.K_finite_translates_finite_dimensional** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A : HSp D) (cA : CptF A) (a : GL (Fin A.rank) (AE D)), Defin θ A cA a → ∀ (W : HSp D) (χ : (AE D)ˣ →* Circle) (omh : WRep (dblSp A) W), IsWeilDatumO ψ (dblSp A) W 1 χ omh → ∀ φ, IsKFinVec omh (Khat A a) φ → ∀ K : Set (GA D A), IsCompact K → FiniteDimensional ℂ (Submodule.span ℂ {f : WO (dblSp A) W → ℂ | ∃ g₁ ∈ K, ∃ g₂ ∈ K, f = omh (io A (g₁, g₂), 1) φ})

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KFiniteVectorsDiscreteSeriesSmoothAbsolutelyIrre

theorem **kFin_rank_zero** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) (h0 : V.rank = 0) : kFin M τ ≤ M.sm τ ∧ ∃ a : Matrix (Fin V.rank) (Fin V.rank) ℂ → Module.End ℂ (kFin M τ), GlobalTheory.Chain.IoLift.IsKAct M τ a ∧ GlobalTheory.Chain.IoLift.AbsIrr (fun X : lieAt V τ => a X)

theorem **kFin_smooth_absIrr** (hH : GlobalTheory.Published.HarishChandraModules) (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (τ : D.what₂.Tau) : kFin M τ ≤ M.sm τ ∧ ∃ a : Matrix (Fin V.rank) (Fin V.rank) ℂ → Module.End ℂ (kFin M τ), GlobalTheory.Chain.IoLift.IsKAct M τ a ∧ GlobalTheory.Chain.IoLift.AbsIrr (fun X : lieAt V τ => a X)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KFiniteVectorsDiscreteSeriesSmoothAbsolutelyIrreStatement

theorem **_root_.GlobalTheory.T.kFinite_vectors_discrete_series_smooth_absolutely_irreducible** : GlobalTheory.Published.HarishChandraModules → ∀ (D : D7) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (τ : D.what₂.Tau), GlobalTheory.IndefiniteRowSideFullHecke.KTypeExtension.kFin M τ ≤ M.sm τ ∧ ∃ a : Matrix (Fin V.rank) (Fin V.rank) ℂ → Module.End ℂ (GlobalTheory.IndefiniteRowSideFullHecke.KTypeExtension.kFin M τ), GlobalTheory.Chain.IoLift.IsKAct M τ a ∧ GlobalTheory.Chain.IoLift.AbsIrr (fun X : lieAt V τ => a X)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KStableSubmoduleEqTopOfHighestWeightLe

theorem **eq_top_of_hwSub_le** : ∀ (p q : ℕ) (M : GKMod p q) (N : Submodule ℂ M.V), KStable M.act N → (∀ μ, hwSub M.act μ ≤ N) → N = ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KStableSubmoduleEqTopOfHighestWeightLeStatement

theorem **_root_.GlobalTheory.T.kStable_submodule_eq_top_of_highestWeight_le** : ∀ (p q : ℕ) (M : GKMod p q) (N : Submodule ℂ M.V), KStable M.act N → (∀ μ, hwSub M.act μ ≤ N) → N = ⊤

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaConjugateHalfPointTranslatesLastColumn

theorem **adelicConj_adelicConj_eq_self** (x : AE D) : cA D (cA D x) = x

theorem **e_mul_sec** (p : SplitPt D k) (x : k) : p.e ∗ p.sec x = p.sec x

theorem **map_pr_rat** (p : SplitPt D k) {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) : (rat M : Mx D m n).map p.pr = M.map (iota p)

theorem **matrix_mul_replicateCol_eq_replicateCol_mulVec** {R : Type} [CommRing R] {m l : Type} [Fintype l] (A : Matrix m l R) (x : l → R) : A ∗ replicateCol (Fin 1) x = replicateCol (Fin 1) (A *ᵥ x)

theorem **adj_wCol** (p : SplitPt D k) {W₀ L' : HSp D} (ζ' : Fin L'.rank → D.host.E) (β : Frame p W₀ n) (t : n → k) : adj (wCol p ζ' β t) = Matrix.of fun i j => p.sec (dualPick ((L'.gram.map (iota p)) *ᵥ fun i => iota p (ζ' i)) j ∗ ((iota p D.host.xi0)⁻¹ ∗ ((W₀.gram.map (iota p)) *ᵥ (β.b *ᵥ t)) i))

theorem **kappa_wCol_translate** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (L' W₀ : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ'), IsLineVec L' ζ' → ∀ (α : Frame p B.A n') (α₀ : Frame p B.V₀ (n' ⊕ Fin 1)) (β : Frame p W₀ n), IsSumFrame B.o B.ζ α α₀ → ∀ t : n → k, IsHalf p.e ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wCol p ζ' β t))) ∧ ∀ i j, rd α₀ β ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wCol p ζ' β t))) i j = Sum.elim (fun _ => 0) (fun _ => -t i) j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaConjugateHalfPointTranslatesLastColumnStatement

theorem **_root_.GlobalTheory.T.kappa_conjugateHalfPoint_translates_lastColumn** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (L' W₀ : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ'), IsLineVec L' ζ' → ∀ (α : Frame p B.A n') (α₀ : Frame p B.V₀ (n' ⊕ Fin 1)) (β : Frame p W₀ n), IsSumFrame B.o B.ζ α α₀ → ∀ t : n → k, IsHalf p.e ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wCol p ζ' β t))) ∧ ∀ i j, rd α₀ β ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wCol p ζ' β t))) i j = Sum.elim (fun _ => 0) (fun _ => -t i) j

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaMapsBesselLineLatticeIffSplit

theorem **star_self_pair** : ∀ (D : D7) (L : HSp D) (ζ : Fin L.rank → D.host.E), star (star ζ ·ᵥ HSpace.gram L *ᵥ ζ) = star ζ ·ᵥ HSpace.gram L *ᵥ ζ

theorem **kappa_line_lat_iff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n' n'' : Type) [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (α : ∀ i, Frame (p i) B.A n') (α₀ : ∀ i, Frame (p i) B.V₀ (n' ⊕ Fin 1)) (γ : ∀ i, Frame (p i) B'.A n'') (γ₀ : ∀ i, Frame (p i) B'.V₀ (n'' ⊕ Fin 1)) (M : Lat B.V₀) (LS : Lat S) (M' : Lat B'.V₀) (Λ : ι → Submodule 𝒪[k] (n' → k)) (Λ' : ι → Submodule 𝒪[k] (n'' → k)), IsSelfDualC B.V₀ M → IsSelfDualC S LS → IsSelfDualC B'.V₀ M' → (∀ i, IsSumFrame B.o B.ζ (α i) (α₀ i) ∧ IsSumFrame B'.o B'.ζ (γ i) (γ₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i)) ∧ (γ₀ i).IsLat M' (lastKer (Λ' i))) → ∀ x : WO S B'.L, MxAt P x → ((B'.o.resT S).symm (0, x) ∈ l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaMapsBesselLineLatticeIffSplitStatement

theorem **_root_.GlobalTheory.T.kappa_maps_besselLineLattice_iff_split** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n' n'' : Type) [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (α : ∀ i, Frame (p i) B.A n') (α₀ : ∀ i, Frame (p i) B.V₀ (n' ⊕ Fin 1)) (γ : ∀ i, Frame (p i) B'.A n'') (γ₀ : ∀ i, Frame (p i) B'.V₀ (n'' ⊕ Fin 1)) (M : Lat B.V₀) (LS : Lat S) (M' : Lat B'.V₀) (Λ : ι → Submodule 𝒪[k] (n' → k)) (Λ' : ι → Submodule 𝒪[k] (n'' → k)), IsSelfDualC B.V₀ M → IsSelfDualC S LS → IsSelfDualC B'.V₀ M' → (∀ i, IsSumFrame B.o B.ζ (α i) (α₀ i) ∧ IsSumFrame B'.o B'.ζ (γ i) (γ₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i)) ∧ (γ₀ i).IsLat M' (lastKer (Λ' i))) → ∀ x : WO S B'.L,

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaMemLineLatticeCompletionIffScalar

theorem **latV_mem_idAt** {D : D7} {S : HSp D} {L : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : ∀ r, y r ∈ idAt D P

theorem **GlobalTheory.Corev.kappa_mem_latO_iff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n' : Type) [Fintype n'] [DecidableEq n'] (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (α : ∀ i, Frame (p i) B.A n') (α₀ : ∀ i, Frame (p i) B.V₀ (n' ⊕ Fin 1)) (M : Lat B.V₀) (LS : Lat S) (Λ : ι → Submodule 𝒪[k] (n' → k)), IsSelfDualC B.V₀ M → IsSelfDualC S LS → (∀ i, IsSumFrame B.o B.ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ x : WO S B'.L, MxAt P x → ((B.o.resS S).symm (0, l.κ S x) ∈ latO B.V₀ S (M : Set _) (LS : Set _) P ↔ ∀ y ∈ latV S LS P, ∃ d : AE D, d ∈ idAt D P ∧ (x : Matrix _ _ (AE D)) *ᵥ y = d • ratV B'.ζ ∧ ∀ c : AE D, c ∈ idAt D P → (∀ i, (p i).pr c ∈ 𝒪[k] ∧ (p i).pr (cA D c) ∗ (p i).pr (incA D (star B.ζ ·ᵥ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaMemLineLatticeCompletionIffScalarStatement

theorem **_root_.GlobalTheory.T.kappa_mem_lineLatticeCompletion_iff_scalar** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n' : Type) [Fintype n'] [DecidableEq n'] (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (α : ∀ i, Frame (p i) B.A n') (α₀ : ∀ i, Frame (p i) B.V₀ (n' ⊕ Fin 1)) (M : Lat B.V₀) (LS : Lat S) (Λ : ι → Submodule 𝒪[k] (n' → k)), IsSelfDualC B.V₀ M → IsSelfDualC S LS → (∀ i, IsSumFrame B.o B.ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ x : WO S B'.L, MxAt P x → ((B.o.resS S).symm (0, l.κ S x) ∈ latO B.V₀ S (M : Set _) (LS : Set _) P ↔ ∀ y ∈ latV S LS P, ∃ d : AE D, d ∈ idAt D P ∧ (x : Matrix _ _ (AE D)) *ᵥ y = d • ratV B'.ζ ∧ ∀ c : AE D, c ∈ idAt D P → (∀ i, (p i).pr c ∈ 𝒪[k] ∧ (p i).pr (cA D c) ∗ (p

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaSplitHalfPointMultipliesByCharacter

theorem **half_mul_right** {m n r : ℕ} {f : AE D} {M : Mx D m n} (h : IsHalf f M) (N : Mx D n r) : IsHalf f (M · N)

theorem **half_mul_left** {m n r : ℕ} {f : AE D} {M : Mx D n r} (N : Mx D m n) (h : IsHalf f M) : IsHalf f (N · M)

theorem **half_smul** {m n : ℕ} {f : AE D} {M : Mx D m n} (c : AE D) (h : IsHalf f M) : IsHalf f (c • M)

theorem **half_adj** {m n : ℕ} {f : AE D} {M : Mx D m n} (h : IsHalf f M) : IsHalf (cA D f) (adj M)

theorem **trace_half** {n : ℕ} {f : AE D} {M : Mx D n n} (h : IsHalf f M) : f ∗ M.trace = M.trace

theorem **e_mul_sec** (p : GlobalTheory.FinPl.SplitPt D k) (a : k) : p.e ∗ p.sec a = p.sec a

theorem **wRow_half** {n : Type} [Fintype n] [DecidableEq n] (p : GlobalTheory.FinPl.SplitPt D k) {W₀ L' : HSp D} (ζ' : Fin L'.rank → D.host.E) (β : GlobalTheory.FinPl.Frame p W₀ n) (t : n → k) : IsHalf p.e (GlobalTheory.FinPl.LineAct.wRow p ζ' β t)

theorem **trace_vecMulVec_mul** {R : Type} [CommRing R] {a b c : Type} [Fintype a] [Fintype b] [Fintype c] (u : a → R) (v : b → R) (X : Matrix b c R) (J : Matrix c a R) : (vecMulVec u v ∗ X ∗ J).trace = v ·ᵥ (X *ᵥ (J *ᵥ u))

theorem **pr_trace** {n : Type} [Fintype n] [DecidableEq n] (p : GlobalTheory.FinPl.SplitPt D k) {W₀ L' : HSp D} {a b : ℕ} (τ : Matrix (Fin a) (Fin L'.rank) D.host.E) (ζ' : Fin L'.rank → D.host.E) (ζ : Fin a → D.host.E) (hτ : τ *ᵥ ζ' = −ζ) (β : GlobalTheory.FinPl.Frame p W₀ n) (t : n → k) (x : Mx D W₀.rank b) (iB : Matrix (Fin b) (Fin a) D.host.E) : p.pr (((rat τ : Mx D a L'.rank) · GlobalTheory.FinPl.LineAct.wRow p ζ' β t) · (x · (rat iB : Mx D b a))).trace = −(t ·ᵥ (β.b' *ᵥ (x.map p.pr *ᵥ (iB.map (iota p) *ᵥ fun i => iota p (ζ i))))))

theorem **kappa_wRow_char** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (χ : AddChar k Circle), p.IsChar ψ χ → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (L' W₀ : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (α : Frame p B.A n') (α₀ : Frame p B.V₀ (n' ⊕ Fin 1)) (β : Frame p W₀ n), IsSumFrame B.o B.ζ α α₀ → ∀ t : n → k, IsHalf (cA D p.e) ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wRow p ζ' β t))) ∧ ∀ x : WO B.V₀ W₀, IsHalf p.e x → ψ (symplO B.V₀ W₀ x ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wRow p ζ' β t)))) = (χ (−(t ·ᵥ fun i => rd α₀ β x i (Sum.inr 0))) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KappaSplitHalfPointMultipliesByCharacterStatement

theorem **_root_.GlobalTheory.T.kappa_splitHalfPoint_multiplies_by_character** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (χ : AddChar k Circle), p.IsChar ψ χ → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (L' W₀ : HSp D) (ζ' : Fin L'.rank → D.host.E) (l : Link B L' ζ') (α : Frame p B.A n') (α₀ : Frame p B.V₀ (n' ⊕ Fin 1)) (β : Frame p W₀ n), IsSumFrame B.o B.ζ α α₀ → ∀ t : n → k, IsHalf (cA D p.e) ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wRow p ζ' β t))) ∧ ∀ x : WO B.V₀ W₀, IsHalf p.e x → ψ (symplO B.V₀ W₀ x ((B.o.resS W₀).symm (0, l.κ W₀ (GlobalTheory.FinPl.LineAct.wRow p ζ' β t)))) = (χ (−(t ·ᵥ fun i => rd α₀ β x i (Sum.inr 0))) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KeepLatticeHyperbolicEmbedding

theorem **cT_map** (hφ : ∀ x, φ (σ x) = τ (φ x)) {m p : Type} (M : Matrix m p A) : (cT σ M).map φ = cT τ (M.map φ)

theorem **conj_mem_iff** {n : ℕ} (b g : GL (Fin n) B) : b⁻¹ ∗ g ∗ b ∈ UJ τ (1 : Matrix (Fin n) (Fin n) B) ↔ cT τ g.1 ∗ (cT τ (b⁻¹).1 ∗ (b⁻¹).1) ∗ g.1 = cT τ (b⁻¹).1 ∗ (b⁻¹).1

theorem **conj_bc_mem_iff** (hφ : ∀ x, φ (σ x) = τ (φ x)) {n : ℕ} (c g : GL (Fin n) A) : (bc φ c)⁻¹ ∗ bc φ g ∗ bc φ c ∈ UJ τ (1 : Matrix (Fin n) (Fin n) B) ↔ (cT σ g.1 ∗ (cT σ (c⁻¹).1 ∗ (c⁻¹).1) ∗ g.1).map φ = (cT σ (c⁻¹).1 ∗ (c⁻¹).1).map φ

theorem **compInf_sigR** (σ : E →ₐ[F] E) (w : InfinitePlace F) (x : AF F ⊗[F] E) : compInf F E w (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ x) = Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ (compInf F E w x)

theorem **mem_levelAll_iff** (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (L : Submodule (𝒪 E) (Fin n → E)) (c : GL (Fin n) (AF F ⊗[F] E)) (g : GAd σ J) : g ∈ levelAll σ J L c ↔ (∀ v y, y ∈ latLoc F L v ↔ (GlobalTheory.AdelicU.matHom _ σ J g).1.map (compFin F E v)

*ᵥ y ∈ latLoc F L v) ∧ ∀ w, (cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ) (GlobalTheory.AdelicU.matHom _ σ J g).1 * (cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ) (c⁻¹).1 * (c⁻¹).1) * (GlobalTheory.AdelicU.matHom _ σ J g).1).map (compInf F E w) = (cT (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.sigR _ σ) (c⁻¹).1 * (c⁻¹).1).map (compInf F E w)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KeepsDiagOfMulStkRowsOne

theorem **keepsDiag_of_mul_stkRows_one** {R : Type} [CommRing R] {n : ℕ} {M : Matrix (Fin (n + n)) (Fin (n + n)) R} {β : Matrix (Fin n) (Fin n) R} (h : M * stkRows 1 1 = stkRows β β) : KeepsDiag finSumFinEquiv M

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KerReductionLeRangeReductionOfMulEqSmul

theorem **ker_map_subset_range_map_of_mul_eq_smul** : ∀ {O : Type} [CommRing O] [IsDomain O] {k : Type} [Field k] (r : O →+* k), Function.Surjective r → ∀ {n : ℕ} (M N : Matrix (Fin n) (Fin n) O) (π : O), π ≠ 0 → (∀ t : O, r t = 0 → ∃ s, t = π * s) → M * N = π • (1 : Matrix (Fin n) (Fin n) O) → ∀ (v : Fin n → k), N.map r *ᵥ v = 0 → ∃ u, v = M.map r *ᵥ u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KerReductionLeRangeReductionOfMulEqSmulStatement

theorem **_root_.GlobalTheory.T.ker_reduction_le_range_reduction_of_mul_eq_smul** : ∀ {O : Type} [CommRing O] [IsDomain O] {k : Type} [Field k] (r : O →+* k), Function.Surjective r → ∀ {n : ℕ} (M N : Matrix (Fin n) (Fin n) O) (π : O), π ≠ 0 → (∀ t : O, r t = 0 → ∃ s, t = π * s) → M * N = π • (1 : Matrix (Fin n) (Fin n) O) → ∀ (v : Fin n → k), N.map r *ᵥ v = 0 → ∃ u, v = M.map r *ᵥ u

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelEntryRationalEquivarianceTransports

theorem **ent_smul** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws), I.Ok → ∀ (u : B.Word) (a : I.ι) (i : ℕ) {b : (I.stp u a i).β} {b' : (I.stp u a (i + 1)).β} (q : (I.stp u a i).D b) (q' : (I.stp u a (i + 1)).D b') (g : (I.stp u a i).G b) (g' : (I.stp u a (i + 1)).G b') (x : GlobalTheory.Mem.Col2.Fr B.fields.emb B.embLL (B.R.N i) → ℂ), (I.chain l u a).sharp i x ∈ LinearMap.range (fnOf (Gr B.fields.emb B.embLL (B.R.N i)) 0 1) → ent I l u a i ((I.stp u a i).act b g q) ((I.stp u a (i + 1)).act b' g' q') ((I.stp u a i).tr B.fields.emb B.embLL b g x) = (I.stp u a (i + 1)).tr B.fields.emb B.embLL b' g' (ent I l u a i q q' x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelEntryRationalEquivarianceTransportsStatement

theorem **_root_.GlobalTheory.T.kernelEntry_rationalEquivariance_transports** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws), I.Ok → ∀ (u : B.Word) (a : I.ι) (i : ℕ) {b : (I.stp u a i).β} {b' : (I.stp u a (i + 1)).β} (q : (I.stp u a i).D b) (q' : (I.stp u a (i + 1)).D b') (g : (I.stp u a i).G b) (g' : (I.stp u a (i + 1)).G b') (x : Fr B.fields.emb B.embLL (B.R.N i) → ℂ), (I.chain l u a).sharp i x ∈ LinearMap.range (fnOf (Gr B.fields.emb B.embLL (B.R.N i)) 0 1) → ent I l u a i ((I.stp u a i).act b g q) ((I.stp u a (i + 1)).act b' g' q') ((I.stp u a i).tr B.fields.emb B.embLL b g x) = (I.stp u a (i + 1)).tr B.fields.emb B.embLL b' g' (ent I l u a i q q' x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelExpansionAbsoluteConvergenceEqualWords

theorem **kExp_absConv_tied** : Formalism.L2.SG.ThmSOwn.KExpTied ∧ AbsConvTied

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelExpansionAbsoluteConvergenceEqualWordsStatement

theorem **_root_.GlobalTheory.T.kernel_expansion_absolute_convergence_equal_words** : Formalism.L2.SG.ThmSOwn.KExpTied ∧ AbsConvTied

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelPartLawsOfColumnPins

theorem **clsKL_is** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → JetS I l Z R.ψ R.θ s → (∀ j, j ≤ B.R.m → (g.x j).Is) → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelPartLawsOfColumnPinsStatement

theorem **_root_.GlobalTheory.T.kernelPart_laws_of_columnPins** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → JetS I l Z R.ψ R.θ s → (∀ j, j ≤ B.R.m → (g.x j).Is) → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelReductionExtendEvaluatedIntegrated

theorem **mul_sum_law** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (S : Finset (m × m')), (∀ e ∈ S, ∀ (b : k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = (ψ (b * (t e.1 e.2 - β e.1 e.2)) : ℂ) * 0.1 t) → l 0' = l 0) → ∀ (b : m × m' → k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = (ψ (∑ e ∈ S, b e * (t e.1 e.2 - β e.1 e.2)) : ℂ) * 0.1 t) → l 0' = l 0

theorem **trans_sum_law** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (S : Finset (m × m')), (∀ e ∈ S, ∀ (a : k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + Matrix.single e.1 e.2 a)) → l 0' = l 0) → ∀ (a : m × m' → k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + ∑ e ∈ S, Matrix.single e.1 e.2 (a e))) → l 0' = l 0

theorem **kerRed_extend** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (ev it ev' it' : Finset (m × m')), ev ⊆ ev' → it ⊆ it' → Disjoint ev' it' → GlobalTheory.FlatFactor.KerRed ν ev it β l → (∀ e ∈ ev', e ∉ ev → ∀ (b : k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = (ψ (b * (t e.1 e.2 - β e.1 e.2)) : ℂ) * 0.1 t) → l 0' = l 0) → (∀ e ∈ it', e ∉ it → ∀ (a : k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + Matrix.single e.1 e.2 a)) → l 0' = l 0) → GlobalTheory.FlatFactor.KerRed ν ev' it' β l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelReductionExtendEvaluatedIntegratedStatement

theorem **_root_.GlobalTheory.T.kernelReduction_extend_evaluated_integrated** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (ev it ev' it' : Finset (m × m')), ev ⊆ ev' → it ⊆ it' → Disjoint ev' it' → GlobalTheory.FlatFactor.KerRed ν ev it β l → (∀ e ∈ ev', e ∉ ev → ∀ (b : k) (θ θ' : SB (Matrix m m' k)), (∀ t, θ'.1 t = (ψ (b * (t e.1 e.2 - β e.1 e.2)) : ℂ) * θ.1 t) → l θ' = l θ) → (∀ e ∈ it', e ∉ it → ∀ (a : k) (θ θ' : SB (Matrix m m' k)), (∀ t, θ'.1 t = θ.1 (t + Matrix.single e.1 e.2 a)) → l θ' = l θ) → GlobalTheory.FlatFactor.KerRed ν ev' it' β l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelsCommuteExplicitConst

theorem **kernelsCommuteExplicitConst** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (ψ : AddChar k Circle) (hψ : HasConductor ψ 1) (l' l : kˣ →* ℂˣ), IsUnram l' → IsUnram l → ∀ (P₀ : Pair k n') (Q₀ : Pair k n), ∀ (V' V : Type) [AddCommGroup V'] [Module ℂ V'] [AddCommGroup V] [Module ℂ V] (π' : Representation ℂ (GL n' k) V') (π : Representation ℂ (GL n k) V) (θ : V' →ₗ[ℂ] SB (Matrix n n' k) →ₗ[ℂ] V), (∀ (g : GL n' k) (h : GL n k) (v : V') (Ψ Ψ' : SB (Matrix n n' k)), (∀ x, Ψ'.1 x = (l' (Matrix.GeneralLinearGroup.det g) : ℂ) * (l (Matrix.GeneralLinearGroup.det h) : ℂ) * Ψ.1 ((h⁻¹ : GL n k).val * x * g.val)) → θ (π' g v) Ψ' = π h (θ v Ψ)) → ∃ K : EqForm π' →ₗ[ℂ] EqForm π, (∀ φ, (K φ).1 Q₀ = θ (φ.1 P₀) (phiOneSB P₀ Q₀)) ∧ (∀ (s : 𝓀[k]ˣ) φ, (K (dia s φ.1, dia_mem_

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelsCommuteExplicitConstStatement

theorem **_root_.GlobalTheory.T.kernelsCommuteExplicitConstAlt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (ψ : AddChar k Circle) (hψ : HasConductor ψ 1) (l' l : kˣ →* ℂˣ), IsUnram l' → IsUnram l → ∀ (P₀ : Pair k n') (Q₀ : Pair k n), ∀ (V' V : Type) [AddCommGroup V'] [Module ℂ V'] [AddCommGroup V] [Module ℂ V] (π' : Representation ℂ (GL n' k) V') (π : Representation ℂ (GL n k) V) (θ : V' →ₗ[ℂ] SB (Matrix n n' k) →ₗ[ℂ] V), (∀ (g : GL n' k) (h : GL n k) (v : V') (Ψ Ψ' : SB (Matrix n n' k)), (∀ x, Ψ'.1 x = (l' (Matrix.GeneralLinearGroup.det g) : ℂ) * (l (Matrix.GeneralLinearGroup.det h) : ℂ) * Ψ.1 ((h⁻¹ : GL n k).val * x * g.val)) → θ (π' g v) Ψ' = π h (θ v Ψ)) → ∃ K : EqForm π' →ₗ[ℂ] EqForm π, (∀ φ, (K φ).1 Q₀ = θ (φ.1 P₀) (phiOneSB P₀ Q₀)) ∧ (∀ (s : 𝓀[k]ˣ) φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelsCommuteTwisted

theorem **unram_val** {χ : kˣ →* ℂˣ} (hχ : IsUnram χ) (u z : kˣ) (hz : valuation k ((z * u : kˣ) : k) = 1) : (χ z : ℂ) = (χ u : ℂ)⁻¹

theorem **theta_mem** (hl : IsUnram l) (hθ : IsTwEquiv l' l π' π θ) (P₀ : Pair k n') (Q₀ : Pair k n) (v : V') : θ v (phiOneSB P₀ Q₀) ∈ levelVec π Q₀

theorem **KK_mem** (φ : Form k n' V') : KK hl hθ P₀ Q₀ φ ∈ EqForm π

theorem **KK_smul** (φ : Form k n' V') (h : GL n k) : KK hl hθ P₀ Q₀ φ (h • Q₀) = π h (θ (φ P₀) (phiOneSB P₀ Q₀))

theorem **KK_move** (hl' : IsUnram l') {φ : Form k n' V'} (hφ : φ ∈ EqForm π') (u : kˣ) (g : GL n' k) (h : GL n k) (hg : valuation k ((Matrix.GeneralLinearGroup.det g * u : kˣ) : k) = 1) (hh : valuation k ((Matrix.GeneralLinearGroup.det h * u : kˣ) : k) = 1) : KK hl hθ P₀ Q₀ φ (h • Q₀) = ((l' u : ℂ) * (l u : ℂ))⁻¹ • θ (φ (g • P₀)) (phiOneSB (g • P₀) (h • Q₀))

theorem **KK_same** (hl' : IsUnram l') {φ : Form k n' V'} (hφ : φ ∈ EqForm π') {P : Pair k n'} {Q : Pair k n} (hP : P.lat = P₀.lat) (hQ : Q.lat = Q₀.lat) : KK hl hθ P₀ Q₀ φ Q = θ (φ P) (phiOneSB P Q)

theorem **KK_up** (hl' : IsUnram l') (q : Unif k) {φ : Form k n' V'} (hφ : φ ∈ EqForm π') {P : Pair k n'} {Q : Pair k n} (hP : P.ker = P₀.lat) (hQ : Q.ker = Q₀.lat) : KK hl hθ P₀ Q₀ φ Q = ((l' q.unit : ℂ) * (l q.unit : ℂ))⁻¹ • θ (φ P) (phiOneSB P Q)

theorem **U_KK** (hl' : IsUnram l') (q : Unif k) {φ : Form k n' V'} (hφ : φ ∈ EqForm π') : U q (KK hl hθ P₀ Q₀ φ) = ((l' q.unit : ℂ) * (l q.unit : ℂ))⁻¹ • KK hl hθ P₀ Q₀ (U q φ)

theorem **KK_dia** (hl' : IsUnram l') (s : 𝓀[k]ˣ) {φ : Form k n' V'} (hφ : φ ∈ EqForm π') : KK hl hθ P₀ Q₀ (dia s φ) = dia s (KK hl hθ P₀ Q₀ φ)

theorem **KK_ram** (hl' : IsUnram l') {φ : Form k n' V'} (hφ : φ ∈ EqForm π') (h : φ ∈ Mram k n' V') : KK hl hθ P₀ Q₀ φ ∈ Mram k n V

theorem **FFStar_KK** (hl' : IsUnram l') (q : Unif k) (χ : AddChar 𝓀[k] ℂ) {φ : Form k n' V'} (hφ : φ ∈ EqForm π') (h : φ ∈ Mram k n' V') : FFStar q χ (KK hl hθ P₀ Q₀ φ) = ((l' q.unit : ℂ) * (l q.unit : ℂ))⁻¹ • KK hl hθ P₀ Q₀ (FFStar q χ φ)

theorem **KK_FFStar** (hl' : IsUnram l') (q : Unif k) (χ : AddChar 𝓀[k] ℂ) {φ : Form k n' V'} (hφ : φ ∈ EqForm π') (h : φ ∈ Mram k n' V') : KK hl hθ P₀ Q₀ (FFStar q χ φ) = ((l' q.unit : ℂ) * (l q.unit : ℂ)) • FFStar q χ (KK hl hθ P₀ Q₀ φ)

theorem **kernelsCommuteTwisted** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (ψ : AddChar k Circle) (hψ : HasConductor ψ 1) (l' l : kˣ →* ℂˣ), IsUnram l' → IsUnram l → ∀ (P₀ : Pair k n') (Q₀ : Pair k n), ∃ c : ℂ, c ≠ 0 ∧ ∀ (V' V : Type) [AddCommGroup V'] [Module ℂ V'] [AddCommGroup V] [Module ℂ V] (π' : Representation ℂ (GL n' k) V') (π : Representation ℂ (GL n k) V) (θ : V' →ₗ[ℂ] SB (Matrix n n' k) →ₗ[ℂ] V), (∀ (g : GL n' k) (h : GL n k) (v : V') (Ψ Ψ' : SB (Matrix n n' k)), (∀ x, Ψ'.1 x = (l' (Matrix.GeneralLinearGroup.det g) : ℂ) * (l (Matrix.GeneralLinearGroup.det h) : ℂ) * Ψ.1 ((h⁻¹ : GL n k).val * x * g.val)) → θ (π' g v) Ψ' = π h (θ v Ψ)) → ∃ K : EqForm π' →ₗ[ℂ] EqForm π, (∀ φ, (K φ).1 Q₀ = θ (φ.1 P₀) (phiOneSB P₀ Q₀)) ∧ (∀ (s : 𝓀[k]ˣ) φ, (K (dia s φ.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KernelsCommuteTwistedStatement

theorem **_root_.GlobalTheory.T.kernelsCommuteTwistedAlt** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (q : Unif k) (ψ : AddChar k Circle) (hψ : HasConductor ψ 1) (l' l : kˣ →* ℂˣ), IsUnram l' → IsUnram l → ∀ (P₀ : Pair k n') (Q₀ : Pair k n), ∃ c : ℂ, c ≠ 0 ∧ ∀ (V' V : Type) [AddCommGroup V'] [Module ℂ V'] [AddCommGroup V] [Module ℂ V] (π' : Representation ℂ (GL n' k) V') (π : Representation ℂ (GL n k) V) (θ : V' →ₗ[ℂ] SB (Matrix n n' k) →ₗ[ℂ] V), (∀ (g : GL n' k) (h : GL n k) (v : V') (Ψ Ψ' : SB (Matrix n n' k)), (∀ x, Ψ'.1 x = (l' (Matrix.GeneralLinearGroup.det g) : ℂ) * (l (Matrix.GeneralLinearGroup.det h) : ℂ) * Ψ.1 ((h⁻¹ : GL n k).val * x * g.val)) → θ (π' g v) Ψ' = π h (θ v Ψ)) → ∃ K : EqForm π' →ₗ[ℂ] EqForm π, (∀ φ, (K φ).1 Q₀ = θ (φ.1 P₀) (phiOneSB P₀ Q₀)) ∧ (∀ (s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KleinFormTorsionUnit

lemma **pt_add** (L : PeriodPair) (m : ℕ) (c d c' d' : ℤ) : pt L m (c + c') (d + d') = pt L m c d + pt L m c' d'

lemma **pt_sub** (L : PeriodPair) (m : ℕ) (c d c' d' : ℤ) : pt L m c d - pt L m c' d' = pt L m (c - c') (d - d')

lemma **pt_mem_iff** (L : PeriodPair) {m : ℕ} (hm : 0 < m) (c d : ℤ) : pt L m c d ∈ L.lattice ↔ (m : ℤ) ∣ c ∧ (m : ℤ) ∣ d

lemma **rep_sub_pt_mem** (L : PeriodPair) {m : ℕ} (hm : 0 < m) (x : ZMod m × ZMod m) (c d : ℤ) (h1 : x.1 = (c : ZMod m)) (h2 : x.2 = (d : ZMod m)) : rep L m x - pt L m c d ∈ L.lattice

lemma **rep_add** (L : PeriodPair) {m : ℕ} (hm : 0 < m) (x y : ZMod m × ZMod m) : rep L m (x + y) - (rep L m x + rep L m y) ∈ L.lattice

lemma **rep_sub** (L : PeriodPair) {m : ℕ} (hm : 0 < m) (x y : ZMod m × ZMod m) : rep L m (x - y) - (rep L m x - rep L m y) ∈ L.lattice

lemma **rep_zero** (L : PeriodPair) (m : ℕ) : rep L m 0 = 0

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KmapEq

theorem **Kmap_eq** (I : TC B X) (l : I.Laws) (u : B.Word) (a : I.ι) (i : ℕ) (c : (I.stp u a i).Cl) (c' : (I.stp u a (i + 1)).Cl) : (I.chain l u a).Kmap i c c' = ent I l u a i ((I.stp u a i).rep c) ((I.stp u a (i + 1)).rep c')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.KmapMemValCTrue

theorem **fnOf_one_apply** {ι : Type} [Fintype ι] {n : ℕ} (G : ι → Matrix (Fin n) (Fin n) ℂ) (s : ι → ℤ) (P : MvPolynomial (ι × Fin n × Fin n) ℂ) (y : Formalism.L2.Forms.Frames G) : fnOf G s 1 P y = (∏ τ, (y.1 τ).det ^ s τ) * MvPolynomial.eval (fun x => y.1 x.1 x.2.1 x.2.2) P

theorem **jetMap_mem_VV** (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (J : (DVar X.Tau (R.N i) (R.N (i + 1)) →₀ ℕ) → ℂ) (P : Pol ℂ X.Tau (R.N i)) (hP : P ∈ VV ℂ (trueWt hc i)) : jetMap J (nuEnd hc i) P ∈ VV ℂ (trueWt hc (i + 1))

theorem **jetMap_zero_mul** {ι : Type} [Fintype ι] {a b : ℕ} (J : (DVar ι a b →₀ ℕ) → ℂ) (n : ι → ℕ) (P : Pol ℂ ι a) : jetMap J 0 (P * ∏ τ, frDet ℂ a τ ^ n τ) = jetMap J n P

theorem **kmap_mem_valC** (I : ChainIn X.E X.K X.Tau R) (σ : X.K →+* ℂ) (h : I.Ok) (hc : X.Tau → HCParam R) {i : ℕ} (hi : i < R.m) (c : (I.S i).Cl) (c' : (I.S (i + 1)).Cl) (J : (DVar X.Tau (R.N i) (R.N (i + 1)) →₀ ℕ) → ℂ) (hJ : I.jet i c c' = jetMap J 0) (f : Fr X.emb σ (R.N i) → ℂ) (hf : f ∈ ValC (Gr X.emb σ (R.N i)) (trueWt hc i)) : (I.chain X.emb σ h hc).Kmap i c c' f ∈ ValC (Gr X.emb σ (R.N (i + 1))) (trueWt hc (i + 1))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LabelAtRealPlaceEqEmbeddingOverReflexField

theorem **sigmaOf_sqrt** (D : D7) (τ : D.what₂.Tau) : D.host.sigmaOf ((D.eTau τ).1 : D.R.GG) D.host.sqrtMinusD = D.host.sqrtMinusD

theorem **labE_sqrt** (D : D7) (σ : D.what₂.LL →+* ℂ) (s : ℂ ≃+* ℂ) (h : IsEmbOverK D σ s) (τ ρ : D.what₂.Tau) : Formalism.IndefR.Links.labE σ τ D.host.sqrtMinusD = tauE D ρ D.host.sqrtMinusD

theorem **labE_lab** (D : D7) (σ : D.what₂.LL →+* ℂ) (s : ℂ ≃+* ℂ) (h : IsEmbOverK D σ s) (ρ : D.what₂.Tau) : Formalism.IndefR.Links.labE σ (Formalism.IndefR.Links.lab σ ρ) = tauE D ρ

theorem **main** : ∀ (D : D7) (σ : D.what₂.LL →+* ℂ) (s : ℂ ≃+* ℂ), IsEmbOverK D σ s → ∀ ρ, Formalism.IndefR.Links.labE σ (Formalism.IndefR.Links.lab σ ρ) = tauE D ρ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LabelAtRealPlaceEqEmbeddingOverReflexFieldStatement

theorem **_root_.GlobalTheory.T.labelAtRealPlace_eq_embedding_over_reflexField** : ∀ (D : D7) (σ : D.what₂.LL →+* ℂ) (s : ℂ ≃+* ℂ), IsEmbOverK D σ s → ∀ ρ, Formalism.IndefR.Links.labE σ (Formalism.IndefR.Links.lab σ ρ) = tauE D ρ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LamEqOfRed

theorem **lam_eq_of_red** : ∀ {D : D7} {c : Comm D} {v : VCom c} (G : GLoc v) (PP : Formalism.L2.DIV.Pl D.host.E), Formalism.L2.NTheta.IsIdemAt G.q.e PP → ∀ (x : G.S.P.J) (g : GL (Fin D.R.r) (kP PP)), GlobalTheory.LevZero.Red G PP x g → ∀ z : 𝒪 D.host.E, Ideal.Quotient.mk PP.asIdeal z = edgeSum (GlobalTheory.chainEdge D.R.k) g → ((G.S.P.lam x : ℂˣ) : ℂ) = (G.S.ψ (((G.S.ϖ⁻¹ : G.kˣ) : G.k) * (ofSplit G.q).pr (z : D.host.E)) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LamTEqLamTHcW

theorem **half_mul** (R : Ranks) (j : ℕ) : ((j * (R.t + 1) / 2 : ℕ) : ℤ) = (j : ℤ) * (((R.t + 1) / 2 : ℕ) : ℤ)

theorem **domWeight_ext** {N : ℕ} {a b : DomWeight N} (h : a.w = b.w) : a = b

theorem **lamT_eq_lamT_hcW** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (j : ℕ) : Formalism.L2.Forms.lamT (D

theorem **evz_mem_valC** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (ψ : ColT I l u a B.R.m), ψ ∈ (I.fm l u a).Fi B.R.m → ∀ c, (fun y : FrN (I.rec7 l Z) (B.R.N B.R.m) => (I.per l u a).evz ψ c (y.1, y.2)) ∈ ValC _ (Formalism.L2.Forms.lamT (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LambdaValueEdgeSum

theorem **char_of_red** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {O : Type} [CommRing O] (I : Ideal O) (π : O →+ k) {s κ r : ℕ} (hr : r = κ + (s + 1)) (S : GlobalTheory.SymChain.SymData k s κ) (x : S.P.J) (g : GL (Fin r) (O ⧸ I)) (hx : ∀ (a b : Fin κ ⊕ Fin (s + 1)) (a' b' : Fin r), a'.val = GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.framePos _ _ a → b'.val = GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.framePos _ _ b → ∀ z : O, Ideal.Quotient.mk I z = g.val a' b' → valuation k ((x : GL (Fin κ ⊕ Fin (s + 1)) k).val a b - π z) < 1) (z : O) (hz : Ideal.Quotient.mk I z = edgeSum (GlobalTheory.chainEdge κ) g) : ((S.P.lam x : ℂˣ) : ℂ) = (S.ψ (((S.ϖ⁻¹ : kˣ) : k) * π z) : ℂ)

theorem **lam_eq_of_red** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (x : G.S.P.J) (g : GL (Fin D.R.r) (kP PP)), G.Red x g → ∀ z : 𝒪 D.host.E, Ideal.Quotient.mk PP.asIdeal z = edgeSum (GlobalTheory.chainEdge D.R.k) g → ((G.S.P.lam x : ℂˣ) : ℂ) = (G.S.ψ (((G.S.ϖ⁻¹ : G.kˣ) : G.k) * (GlobalTheory.LocMod.ofSplit G.q).pr (z : D.host.E)) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LambdaValueEdgeSumStatement

theorem **_root_.GlobalTheory.T.lambda_value_edge_sum** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (x : G.S.P.J) (g : GL (Fin D.R.r) (kP PP)), G.Red x g → ∀ z : 𝒪 D.host.E, Ideal.Quotient.mk PP.asIdeal z = edgeSum (GlobalTheory.chainEdge D.R.k) g → ((G.S.P.lam x : ℂˣ) : ℂ) = (G.S.ψ (((G.S.ϖ⁻¹ : G.kˣ) : G.k) * (GlobalTheory.LocMod.ofSplit G.q).pr (z : D.host.E)) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LambdaZeroEqProduct

theorem **rho_apply** (z : 𝒪 D.host.E) : rho G z = (ofSplit G.q).pr (z : D.host.E)

theorem **rho_int** (z : 𝒪 D.host.E) : rho G z ∈ 𝒪[G.k]

theorem **rho_lt** {z : 𝒪 D.host.E} (hz : z ∈ PP.asIdeal) : valuation G.k (rho G z) < 1

theorem **mk_out** (y : Formalism.L2.DIV.kP PP) : Ideal.Quotient.mk PP.asIdeal (out y) = y

theorem **rho_sub_lt** {y : Formalism.L2.DIV.kP PP} {z : 𝒪 D.host.E} (hz : Ideal.Quotient.mk PP.asIdeal z = y) : valuation G.k (rho G (out y) - rho G z) < 1

theorem **pos_val** (a : Fin D.R.k ⊕ Fin (G.s + 1)) : (pos G a).val = GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.framePos _ _ a

theorem **lift_int** (g : Matrix (Fin D.R.r) (Fin D.R.r) (Formalism.L2.DIV.kP PP)) : IntM (lift G g)

theorem **lift_mul** (g h : Matrix (Fin D.R.r) (Fin D.R.r) (Formalism.L2.DIV.kP PP)) : MaxM (lift G g * lift G h - lift G (g * h))

theorem **lift_one** : MaxM (lift G 1 - 1)

theorem **red_iff** (x : G.S.P.J) (g : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP)) : G.Red x g ↔ MaxM ((x : GL (Fin D.R.k ⊕ Fin (G.s + 1)) G.k).val - lift G g.val)

theorem **intM_J** (x : G.S.P.J) : IntM (x : GL (Fin D.R.k ⊕ Fin (G.s + 1)) G.k).val ∧ IntM ((x : GL (Fin D.R.k ⊕ Fin (G.s + 1)) G.k)⁻¹).val

theorem **red_one** : G.Red 1 1

theorem **red_mul** {x y : G.S.P.J} {g h : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP)} (hx : G.Red x g) (hy : G.Red y h) : G.Red (x * y) (g * h)

theorem **red_inv** {x : G.S.P.J} {g : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP)} (hx : G.Red x g) : G.Red x⁻¹ g⁻¹

theorem **lam_congr** {x x' : G.S.P.J} {g : GL (Fin D.R.r) (Formalism.L2.DIV.kP PP)} (hx : G.Red x g) (hx' : G.Red x' g) : G.S.P.lam x = G.S.P.lam x'

theorem **lambda_zero_eq_product** (g : grp G) (x : ∀ PP, (G PP).S.P.J) (hx : ∀ PP, (G PP).Red (x PP) (g.1 PP)) : lam0 G g = ∏ PP, (G PP).S.P.lam (x PP)

theorem **comb_single_zero** (f : WO S T →₀ ℂ) : comb ψ S T (Finsupp.single 0 1) f = f

theorem **exists_prod** {ι : Type} (c : ι → (WO S T →₀ ℂ)) (P : WO S T → Prop) (h0 : P 0) (hadd : ∀ x y, P x → P y → P (x + y)) (hc : ∀ i, ∀ w ∈ (c i).support, P w) (l : List ι) : ∃ c' : WO S T →₀ ℂ, (∀ w ∈ c'.support, P w) ∧ ∀ f, comb ψ S T c' f = l.foldr (fun i f' => comb ψ S T (c i) f') f

theorem **v_mem_S** {PP : d.c.Pg} (G : GAt d τ PP.1) : G.q.v ∈ d.c.S

theorem **cg_supp** {PP : d.c.Pg} (G : GAt d τ PP.1) {j : ℕ} (hj : j < D.R.m) : ∀ w ∈ (G.cg j).support, MxAt (· ∈ d.c.S) w

theorem **main** : ∀ (D : D7) (u : D.what₂.Word) (d : Dat D u) τ (G : ∀ PP : d.c.Pg, GAt d τ PP.1), d.c.Is → (∃ P : Formalism.L2.DIV.LevelP d.c.Pg D.R.r ℂ, (∀ g, g ∈ P.H ↔ ∀ PP, ∃ x, (G PP).Red x (g PP)) ∧ ∀ (g : P.H) (x : ∀ PP, (G PP).S.P.J), (∀ PP, (G PP).Red (x PP) (g.1 PP)) → P.chi g = ∏ PP, (G PP).S.P.lam (x PP)) ∧ ∃ cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ, (∀ j, ∀ w ∈ (cG j).support, GlobalTheory.FinPl.MxAt (· ∈ d.c.S) w) ∧ ∀ j, j < D.R.m → ∀ f, IsThetaFunO d.c.ψ _ _ f → comb d.c.ψ _ _ (cG j) f = d.c.Pg.attach.toList.foldr (fun PP f' => comb d.c.ψ _ _ ((G PP).cg j) f') f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LargeGapAllMembers

theorem **poly_bound** {M k m f r L T n t N : ℝ} (hM : 1 ≤ M) (hk1 : 1 ≤ k) (hk : k ≤ M) (hm1 : 1 ≤ m) (hm : m ≤ M) (hf0 : 0 ≤ f) (hf : f ≤ M) (hr0 : 0 ≤ r) (hr : r ≤ M) (hL0 : 0 ≤ L) (hL : L ≤ M) (hT : T ≤ M) (hn0 : 0 ≤ n) (hn : n ≤ 10 * r * (f + f ^ 2 * (k - 1) ^ 3 * L ^ 2 / m) + 8) (ht0 : 0 ≤ t) (ht : t ≤ 8 * f * r + 4 + T * n) (hN : N = k + m * (t + 1)) : 4 * N ^ 3 + 2 * t + 6 ≤ 300000 * M ^ 30

theorem **rank_bound** (s : Scale) (R : Ranks) (h : RankVal s R) : 4 * ((R.N R.m : ℕ) : ℝ) ^ 3 + 2 * R.t + 6 ≤ 300000 * (4 * s.x ^ (377 : ℝ)) ^ 30

theorem **nodeGap_of_val** {C : FC} {cD ch : ℝ} {D : D7} {L : SizeData₂ D} (h : Val C cD ch L) : D.NodeGap (h0v L.sc ch : ℝ)

theorem **bigGap_of_val** {C : FC} {cD ch : ℝ} {D : D7} {L : SizeData₂ D} (h : Val C cD ch L) (hch : 11311 ≤ ch) (hx : 300000 * 4 ^ 30 ≤ L.sc.x) : BigGap D

theorem **rowE_bigGap** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => GlobalTheory.Core.Rk.BigGap D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LargeGapAllMembersStatement

theorem **_root_.GlobalTheory.T.large_gap_all_members** : ∀ (hL : GlobalTheory.Published.LocalLanglandsGL) (p : Bl.Prm hL), GlobalTheory.Fc.RowE (Bl.IsGoodE p.s) (Bl.IsLinked p) (fun _ D _ _ _ => GlobalTheory.Core.Rk.BigGap D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastColumnDatumTensorMemSchwartzBruhat

theorem **isLocallyConstant_integral** {Q Y : Type} [TopologicalSpace Q] [TopologicalSpace Y] [MeasurableSpace Y] (μ : Measure Y) {H : Q × Y → ℂ} (hH : IsLocallyConstant H) {K : Set Y} (hK : IsCompact K) (hz : ∀ q, ∀ y ∉ K, H (q, y) = 0) : IsLocallyConstant fun q => ∫ y, H (q, y) ∂μ

theorem **hatCol_mem_SB** (ψ : AddChar k Circle) (hψ : Continuous ψ) (hne : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] {n n' : Type} [Fintype n] [DecidableEq n] {Ψ : Matrix n (n' ⊕ Unit) k → ℂ} (hΨ : Ψ ∈ SB (Matrix n (n' ⊕ Unit) k)) : (fun p : Matrix n n' k × (n → k) => GlobalTheory.StepData.hat ψ ν (-1) (fun y => Ψ (fromCols p.1 (Matrix.of fun r _ => y r))) p.2) ∈ SB (Matrix n n' k × (n → k))

theorem **lastColumn_datum_mem_SB** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψk : AddChar k Circle), Continuous ψk → ψk ≠ 1 → ∀ (n n' Z : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype Z] [DecidableEq Z], (∀ (Ψ₁ : SB (Matrix n (n' ⊕ Unit) k)) (Ψ₂ : SB (Matrix Z n k)), ∃ Ξ : SB (Matrix n n' k × Matrix (Z ⊕ Unit) n k), ∀ q, Ξ.1 q = Ψ₂.1 q.2.toRows₁ * GlobalTheory.StepData.hat ψk ν (-1) (fun y => Ψ₁.1 (fromCols q.1 (Matrix.of fun r _ => y r))) (q.2 (Sum.inr ()))) ∧ ∀ (Ξ₁ : SB (Matrix n n' k)) (c : SB (Matrix n Unit k)), ∃ Ψ₁ : SB (Matrix n (n' ⊕ Unit) k), ∀ x, Ψ₁.1 x = Ξ₁.1 x.toCols₁ * c.1 (Matrix.of fun r _ => x r (Sum.inr ()))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastColumnDatumTensorMemSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.last_column_datum_tensor_mem_schwartzBruhat** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψk : AddChar k Circle), Continuous ψk → ψk ≠ 1 → ∀ (n n' Z : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype Z] [DecidableEq Z], (∀ (Ψ₁ : SB (Matrix n (n' ⊕ Unit) k)) (Ψ₂ : SB (Matrix Z n k)), ∃ Ξ : SB (Matrix n n' k × Matrix (Z ⊕ Unit) n k), ∀ q, Ξ.1 q = Ψ₂.1 q.2.toRows₁ * GlobalTheory.StepData.hat ψk ν (-1) (fun y => Ψ₁.1 (fromCols q.1 (Matrix.of fun r _ => y r))) (q.2 (Sum.inr ()))) ∧ ∀ (Ξ₁ : SB (Matrix n n' k)) (c : SB (Matrix n Unit k)), ∃ Ψ₁ : SB (Matrix n (n' ⊕ Unit) k), ∀ x, Ψ₁.1 x = Ξ₁.1 x.toCols₁ * c.1 (Matrix.of fun r _ => x r (Sum.inr ()))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastDiagonalEntryTorusIntegralOfFlatMapNeZero

theorem **smul_mem_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {a : D.host.E} (ha : a ∈ D.host.AA) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (fun i => ((1 : AF D.host.F) ⊗ₜ[D.host.F] a) ∗ y i) ∈ latA S L P

theorem **mxAt_mul_left** {P : Pl D → Prop} {a b c : ℕ} (y : Mx D a b) {x : Mx D b c} (hx : MxAt P x) : MxAt P (y · x)

theorem **mxAt_mul_right** {P : Pl D → Prop} {a b c : ℕ} {x : Mx D a b} (hx : MxAt P x) (y : Mx D b c) : MxAt P (x · y)

theorem **jX_pX_mem_latV** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {L : Lat T} {L₀ : Lat T₀} {v : Pl D} (hm : MonoAt d L L₀ v) {u : Fin T.rank → AE D} (hu : u ∈ latV T L (· = v)) : d.jX ∗ᵥ (d.pX ∗ᵥ u) ∈ latV T L (· = v)

theorem **jX_pX_mem_latA** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) {L : Lat T} {L₀ : Lat T₀} {v : Pl D} (hm : MonoAt d L L₀ v) {z : Fin T.rank → AE D} (hz : z ∈ latA T L (· = v)) : d.jX ∗ᵥ (d.pX ∗ᵥ z) ∈ latA T L (· = v)

theorem **wittExtZero_inclusionX_mem_homLatticeCompletion** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (dS : HypDec S S₀ ρ) (dT : HypDec T T₀ σ) {LS : Lat S} {LS₀ : Lat S₀} {LT : Lat T} {LT₀ : Lat T₀} {v : Pl D} (hS : IsSelfDualC S LS) (hT : IsSelfDualC T LT) (hS₀ : IsSelfDualC S₀ LS₀) (hmS : MonoAt dS LS LS₀ v) (hmT : MonoAt dT LT LT₀ v) {w : WO S₀ T} (hw : w ∈ latO S₀ T (LS₀ : Set _) (LT : Set _) (· = v)) : dS.ext0 T (dT.incX S₀ (dT.pX · w)) ∈ latO S T (LS : Set _) (LT : Set _) (· = v)

theorem **wittExtZero_inclusionX_smul** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (dS : HypDec S S₀ ρ) (dT : HypDec T T₀ σ) (c : AE D) (ξ : Mx D σ S₀.rank) : dS.ext0 T (dT.incX S₀ (c • ξ)) = c • dS.ext0 T (dT.incX S₀ ξ)

theorem **incAdj_smul** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (c : AE D) (y : Mx D S.rank ρ) : d.incAdj S (c • y) = cA D c • d.incAdj S y

theorem **symplO_incAdj_proj** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (z : Mx D S.rank ρ) (w : WO S T) : symplO S T (d.incAdj S z) w = symplO S T (d.incAdj S z) (d.incX S (d.pX · w))

theorem **symplO_add_right'** (S T : HSp D) (x y y' : WO S T) : symplO S T x (y + y') = symplO S T x y + symplO S T x y'

theorem **half_smul_mem** {S T : HSp D} {LS : Set (Fin S.rank → D.host.E)} {LT : Set (Fin T.rank → D.host.E)} {v : Pl D} (h2 : (2 : 𝒪 D.host.F) ∉ v.asIdeal) {u : WO S T} (hu : u ∈ latO S T LS LT (· = v)) : (pairA D.host.F 0 (GlobalTheory.FinPl.LatMem.indP D.host.F (· = v) ∗ algebraMap D.host.F (Af D.host.F) (2 : D.host.F)⁻¹) ⊗ₜ[D.host.F] (1 : D.host.E)) • u ∈ latO S T LS LT (· = v)

theorem **exists_idemAt** (D : D7) (v : Pl D) : ∃ e : AE D, e ∈ intAt D (· = v) ∧ (∀ a ∈ idAt D (· = v), a ∗ e = a) ∧ cA D e = e ∧ ((2 : 𝒪 D.host.F) ∉ v.asIdeal → ∃ c : AE D, c + c = e ∧ ∀ (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) (u : WO S T), u ∈ latO S T LS LT (· = v) → c • u ∈ latO S T LS LT (· = v))

theorem **fix_char** {ψ : AddChar (AF D.host.F) ℂ} {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} (s : StepBF ψ B J) {P : Pl D → Prop} {𝔉 : (WO B.V J.W → ℂ) → (WO B.V₀ J.W₀ → ℂ)} (h𝔉 : IsFlatBF s P 𝔉) {f : WO B.V J.W → ℂ} (hθ : IsThetaFunO ψ B.V J.W f) (t : GL (Fin ρ) (AE D)) (hne : 𝔉 (s.om (1, J.d.levi t) f) ≠ 0) (ξ : Mx D ρ B.V₀.rank) (hξ : MxAt P ξ) (hfix : heisO ψ B.V J.W (B.d.ext0 J.W (J.d.incX B.V₀ ξ)) f = f) : ψ (symplO B.V₀ J.W (J.d.incAdj B.V₀ (yZero B.V₀ ρ (vecB B.o B.ζ))) (J.d.incX B.V₀ (t.1 · ξ))) = 1

theorem **yZero_mul_diag** (S : HSp D) {ρ : ℕ} (z : Fin S.rank → D.host.E) (t : Matrix (Fin ρ) (Fin ρ) (AE D)) (ht : ∀ i j, i ≠ j → t i j = 0) (i : Fin ρ) (hi : i.1 + 1 = ρ) : yZero S ρ z · t = t i i • yZero S ρ z

theorem **symplO_incAdj_eq** {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (z : Mx D S.rank ρ) (w : WO S T) : symplO S T (d.incAdj S z) w = −Algebra.trace (AF D.host.F) (AE D) (z · (d.pX · w)).trace

theorem **mxAt_smul** {P : Pl D → Prop} {m n : ℕ} (c : AE D) {x : Mx D m n} (hx : MxAt P x) : MxAt P (c • x)

theorem **smul_isHalf'** {e : AE D} (he : e ∗ e = e) {m n : ℕ} (u : Mx D m n) : IsHalf e (e • u)

theorem **smul_eq_self** {P : Pl D → Prop} {e : AE D} (he : ∀ a ∈ idAt D P, e ∗ a = a) {m n : ℕ} {x : Mx D m n} (hx : MxAt P x) : e • x = x

theorem **add_smul_mx** {m n : ℕ} (a b : AE D) (x : Mx D m n) : a • x + b • x = (a + b) • x

theorem **chi_add** (ψ : AddChar (AF D.host.F) ℂ) {T T₀ : HSp D} {ρ : ℕ} (d : HypDec T T₀ ρ) (S : HSp D) (y : Mx D S.rank ρ) (t : Matrix (Fin ρ) (Fin ρ) (AE D)) (ξ₁ ξ₂ : Mx D ρ S.rank) : ψ (symplO S T (d.incAdj S y) (d.incX S (t · (ξ₁ + ξ₂)))) = ψ (symplO S T (d.incAdj S y) (d.incX S (t · ξ₁))) ∗ ψ (symplO S T (d.incAdj S y) (d.incX S (t · ξ₂)))

theorem **last_diag_int** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → UnrAt s v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → ∀ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) → J.d.levi t ∈ grpAt J.W (· = v) → 𝔉 (s.om (1, J.d.levi t) f) ≠ 0 → ∀ i : Fin ρ, i.1 + 1 = ρ → t.1 i i − 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastDiagonalEntryTorusIntegralOfFlatMapNeZeroStatement

theorem **_root_.GlobalTheory.T.lastDiagonalEntry_torus_integral_of_flatMap_ne_zero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (s : StepBF ψ B J) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → UnrAt s v → ∀ 𝔉, IsFlatBF s (· = v) 𝔉 → ∀ f, Vac ψ B.V J.W LV LW (· = v) f → ∀ t : GL (Fin ρ) (AE D), (∀ i j, i ≠ j → t.1 i j = 0) → J.d.levi t ∈ grpAt J.W (· = v) → 𝔉 (s.om (1, J.d.levi t) f) ≠ 0 → ∀ i : Fin ρ, i.1 + 1 = ρ → t.1 i i − 1 ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastDoubleThetaLiftIntegrableLinear

theorem **ker_smul** {A S X : HSp D} (oma : Om A S) (omX : Om S X) (a : ℂ) (C : WO A S → WO S X → ℂ) (h₀ : GA D A) (h₁ : GA D S) (h₂ : GA D X) : GlobalTheory.SeeSmallPf.ker oma omX (a • C) h₀ h₁ h₂ = a ∗ GlobalTheory.SeeSmallPf.ker oma omX C h₀ h₁ h₂

theorem **biLift_smul** {A S X : HSp D} (oma : Om A S) (omX : Om S X) (vA : Vol A) (vS : Vol S) (φ : GA D A → ℂ) (a : ℂ) (C : WO A S → WO S X → ℂ) (h₂ : GA D X) : biLift oma omX vA vS φ (a • C) h₂ = a ∗ biLift oma omX vA vS φ C h₂

theorem **theta_smul** {S T : HSp D} {ψ : AddChar (AF D.host.F) ℂ} (a : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (a • f) where

theorem **sum_fin** {G M : Type} [AddCommMonoid M] (s : Finset G) (F : G → M) : ∑ g ∈ s, F g = ∑ i : Fin s.card, F (s.equivFin.symm i)

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c), c.Is → t.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastDoubleThetaLiftIntegrableLinearStatement

theorem **_root_.GlobalTheory.T.lastDoubleThetaLift_integrable_linear** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c), c.Is → t.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastGroupEmbeddingRealComponentsBlockForm

theorem **iXm_entry** {D : D7} (hi : ∀ (a : Fin D.sp.X'.rank) (b : Fin (D.sp.X D.R.m).rank), (a : ℕ) = b → D.sp.iX' (D.sp.X'.frame a) = (D.sp.X D.R.m).frame b) (i : Fin (D.R.N D.R.m)) (b : Fin (D.R.N D.R.m − 1)) : iXm D i b = if (i : ℕ) = b then 1 else 0

theorem **piT_coe_ne_zero** {B : GlobalTheory.Mem.Base} (X : PreX B.toPre) (u : B.Word) : ((X.piT u : 𝒪 B.host.E) : B.host.E) ≠ 0

theorem **isBlock_infM_emb** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (θ : InfFam (I.rec7 l Z)), IsInfFam θ → ∀ (emb : GA (I.rec7 l Z) (I.rec7 l Z).sp.X' →* GA (I.rec7 l Z) ((I.rec7 l Z).sp.X B.R.m)) (hN : B.R.N B.R.m = (B.R.N B.R.m − 1) + 1), IsEmbTop (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastGroupEmbeddingRealComponentsBlockFormStatement

theorem **_root_.GlobalTheory.T.lastGroup_embedding_real_components_block_form** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (θ : InfFam (I.rec7 l Z)), IsInfFam θ → ∀ (emb : GA (I.rec7 l Z) (I.rec7 l Z).sp.X' →* GA (I.rec7 l Z) ((I.rec7 l Z).sp.X B.R.m)) (hN : B.R.N B.R.m = (B.R.N B.R.m − 1) + 1), IsEmbTop (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastIsotropicLineHomothetyMatchesLattices

theorem **cancel** (v : GlobalTheory.FinPl.Pl D) (x : D.host.E) (hx : x ≠ 0) (A : AE D → Prop) (hA : ∀ a ∈ idAt D (· = v), A a ↔ incA D x⁻¹ ∗ a ∈ intAt D (· = v)) (P Q : AE D → Prop) (hP : ∀ o ∈ intAt D (· = v), ∀ t, P t → P (o ∗ t)) (hQ : ∀ o ∈ intAt D (· = v), ∀ t, Q t → Q (o ∗ t)) (h : ∀ s ∈ idAt D (· = v), (∀ a ∈ idAt D (· = v), A a → P (s ∗ a)) ↔ (∀ a ∈ idAt D (· = v), A a → Q (s ∗ a))) : ∀ t ∈ idAt D (· = v), (P t ↔ Q t)

theorem **colMul_mul** {m n : ℕ} (X : Mx D m n) (i : Fin n) (o t : AE D) : colMul X i (o ∗ t) = o • colMul X i t

theorem **lastLine_match** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (z : Fin S₀.rank → D.host.E) (L L' : Lat S) (M M' : Lat S₀) (v : GlobalTheory.FinPl.Pl D) (c : AE D) (γ : GA D S₀), IsSelfDualC S₀ M → z ≠ 0 → LastAt ψ d z L M v → LastAt ψ d z L' M' v → NormOneAt v c → Formalism.L2.Theta.mt S₀ γ ∗ᵥ ratV z = ratV z → SMoves S₀ c γ M M' v → ∀ i : Fin ρ, i.1 + 1 = ρ → ∀ a ∈ idAt D (· = v), (colMul d.jX i a ∈ latV S L (· = v) ↔ colMul d.jX i (c ∗ a) ∈ latV S L' (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastIsotropicLineHomothetyMatchesLatticesStatement

theorem **_root_.GlobalTheory.T.lastIsotropicLine_homothety_matches_lattices** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (z : Fin S₀.rank → D.host.E) (L L' : Lat S) (M M' : Lat S₀) (v : GlobalTheory.FinPl.Pl D) (c : AE D) (γ : GA D S₀), IsSelfDualC S₀ M → z ≠ 0 → LastAt ψ d z L M v → LastAt ψ d z L' M' v → NormOneAt v c → Formalism.L2.Theta.mt S₀ γ ∗ᵥ ratV z = ratV z → SMoves S₀ c γ M M' v → ∀ i : Fin ρ, i.1 + 1 = ρ → ∀ a ∈ idAt D (· = v), (colMul d.jX i a ∈ latV S L (· = v) ↔ colMul d.jX i (c ∗ a) ∈ latV S L' (· = v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastKernelDatumInLastLineFrameAtLevelPrime

theorem **hasVal_transport** {Q : Pair k n} {R : Pair k n₂} (G : Matrix n₂ n k) (G' : Matrix n n₂ k) (hGG : G ∗ G' = 1) {e : n₂ → k} (he : R.HasVal e 1) (hker : ∀ y, y ∈ Q.ker ↔ G ∗ᵥ y ∈ R.ker) {s : 𝒦[k]ˣ} (hs : Q.HasVal (G' ∗ᵥ e) s) (y : n → k) (a : 𝒦[k]) : Q.HasVal y a ↔ R.HasVal (G ∗ᵥ y) (((s⁻¹ : 𝒦[k]ˣ) : 𝒦[k]) ∗ a)

theorem **maps_transport** {Q : Pair k n₂} {R : Pair k n} {n₁ : Type} [Fintype n₁] [DecidableEq n₁] (P : Pair k n₁) (G : Matrix n n₂ k) (s : 𝒦[k]ˣ) (hT : ∀ y a, Q.HasVal y a ↔ R.HasVal (G ∗ᵥ y) (((s⁻¹ : 𝒦[k]ˣ) : 𝒦[k]) ∗ a)) (x : Matrix n₂ n₁ k) : Maps (s⁻¹ • P) R (G ∗ x) ↔ Maps P Q x

theorem **phiOne_transport** {Q : Pair k n₂} {R : Pair k n} {n₁ : Type} [Fintype n₁] [DecidableEq n₁] (P : Pair k n₁) (G : Matrix n n₂ k) (s : 𝒦[k]ˣ) (hT : ∀ y a, Q.HasVal y a ↔ R.HasVal (G ∗ᵥ y) (((s⁻¹ : 𝒦[k]ˣ) : 𝒦[k]) ∗ a)) (x : Matrix n₂ n₁ k) : phiOne (s⁻¹ • P) R (G ∗ x) = phiOne P Q x

theorem **extFun_eq_of** (α : Frame p S n') (β : Frame p T n₂) (β' : Frame p T n) (Ψ : Matrix n₂ n' k → ℂ) (Ψ' : Matrix n n' k → ℂ) (hΨ : ∀ x, Ψ' ((β'.b' ∗ β.b) ∗ x) = Ψ x) : extFun α β Ψ = extFun α β' Ψ'

theorem **overAll_of_extFun_eq** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (α : ∀ i, Frame (p i) S n') (β : ∀ i, Frame (p i) T n₂) (β' : ∀ i, Frame (p i) T n) (LS : Lat S) (LT : Lat T) (Ψ : ι → Matrix n₂ n' k → ℂ) (Ψ' : ι → Matrix n n' k → ℂ) (h : ∀ i, extFun (α i) (β i) (Ψ i) = extFun (α i) (β' i) (Ψ' i)) (hSB : ∀ i, Ψ' i ∈ SB (Matrix n n' k)) : ∀ (l : List ι) (f f' : WO S T → ℂ), OverAll ψ α β LS LT Ψ l f f' → OverAll ψ α β' LS LT Ψ' l f f' | [], _, _, hf => hf | i :: l, f, f', hf => by obtain (g, hg, hov) : ∃ g, OverAll ψ α β LS LT Ψ l f g ∧ Over ψ (α i) (β i) LS LT g (Ψ i) f'

theorem **isLattice_of_lastKer** {m : Type} [Fintype m] [DecidableEq m] {Q : Pair k n₂} (G : Matrix (m ⊕ Fin 1) n₂ k) (G' : Matrix n₂ (m ⊕ Fin 1) k) (hGG : G ∗ G' = 1) (L : Submodule 𝒪[k] (m → k)) (hker : ∀ y, y ∈ Q.ker ↔ G ∗ᵥ y ∈ GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.lastKer L) : L.IsLattice k

theorem **GlobalTheory.SeeSaw.lastKernel_reps** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι), c.Is → v.Is → ShapeOff v o → ∀ j (h : j + 2 ∗ 0 + 2 = D.R.m), ∃ (Λ : ι → Submodule 𝒪[k] (Fin c.T.rank → k)) (hΛ : ∀ i, (Λ i).IsLattice k) (u : ι → 𝒦[k]ˣ), (∀ i, (o.γ i).IsLat v.NT (Λ i)) ∧ Reps c.ψ (o.β (j + 1)) (o.γ₀ (j + 1 + 1) h) (c.x (j + 1)).L (c.x (j + 1 + 1)).L (fun i => haveI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastKernelDatumInLastLineFrameAtLevelPrimeStatement

theorem **_root_.GlobalTheory.T.lastKernelDatum_in_lastLineFrame_at_levelPrime** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι), c.Is → v.Is → ShapeOff v o → ∀ j (h : j + 2 ∗ 0 + 2 = D.R.m), ∃ (Λ : ι → Submodule 𝒪[k] (Fin c.T.rank → k)) (hΛ : ∀ i, (Λ i).IsLattice k) (u : ι → 𝒦[k]ˣ), (∀ i, (o.γ i).IsLat v.NT (Λ i)) ∧ Reps c.ψ (o.β (j + 1)) (o.γ₀ (j + 1 + 1) h) (c.x (j + 1)).L (c.x (j + 1 + 1)).L (fun i => haveI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastKernelFiniteDatumBoundedLevelSize

theorem **smallDatum_xiCpl2_sharp** : ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu : Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ (p : ChPt (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastKernelFiniteDatumBoundedLevelSizeStatement

theorem **_root_.GlobalTheory.T.lastKernelFiniteDatum_bounded_level_size** : ∀ (CY : ℝ) (D : D7) (N : NumDatum) (X : Fields) (s : Seam D X) (Pr : Primes X) (W : Wts X Pr.Word N.R) (Ch : Chars X Pr N.R.m) (G : ChGlob X N.R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (Sg : Set ℕ) (mu : Tup X.F X.E X.K Sg) (cst : LocCst X (rowSG Pr sel mu)) (I : ChIdx X Pr N.R (QE sel (mu :

Chr X.F X.E X.K))) (dl : ℂ) (χ : TaylorLogMetric.IdE D → ℂ) (cD : D.what₂.Tau → ℂ) (pl : RPl N.R ≃ X.Tau) (u : Pr.Word) (δ : Code N), HypS CY N s W Ch G sel mu cst I dl χ cD pl u δ → ∀ (p : ChPt (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastPeriodClassesGiveTailReading

theorem **tailRd_of_clsT** : ∀ (D : D7) (c : Comm D) (t : Tup c) (ρ : RealD c) (A : ClsT ρ t), A.Is → t.cP ≠ 0 → t.TailRd

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastPeriodClassesGiveTailReadingStatement

theorem **_root_.GlobalTheory.T.lastPeriodClasses_give_tailReading** : ∀ (D : D7) (c : Comm D) (t : Tup c) (ρ : RealD c) (A : ClsT ρ t), A.Is → t.cP ≠ 0 → t.TailRd

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastPeriodLawMovedLevel

theorem **multT_right** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (θ : D.what₂.Tau → Ainf D.host.F ⊗[D.host.F] D.host.E →+* ℂ) (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) (x : (⊗[ℂ] (τ : D.what₂.Tau), ɪ(M.sm τ)) →ₗ[ℂ] GA D V → ℂ) (hx : x ∈ MultT θ M J χ) (v : ⊗[ℂ] (τ : D.what₂.Tau), ɪ(M.sm τ)) : ∀ k ∈ J, ∀ g, x v (g ∗ k) = χ k ∗ x v g

theorem **rd_right** {B : Base} {X : PreX B.toPre} (I : TC B X) (hOk : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch : ℕ → Type) (A : TwDat I l Z u a Ch) (hA : A.Is) (θ : InfFam (I.rec7 l Z)) (hθ : IsInfFam θ) (j : ℕ) (hj : j ≤ B.R.m) (φ : ColT I l u a j) (hφ : φ ∈ (I.fm l u a).Fi j) (b : Ch j) (y : (I.chain l u a).Y j) (x : GA (I.rec7 l Z) ((I.rec7 l Z).sp.X j)) (κ : A.U j b) : A.rd θ j hj b φ y (x ∗ κ.1) = (A.chi j b κ : ℂ) ∗ A.rd θ j hj b φ y x

theorem **GlobalTheory.AdForm.TopInv.law_top** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastPeriodLawMovedLevelStatement

theorem **_root_.GlobalTheory.T.lastPeriod_law_movedLevel** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastPeriodTailEqLoweredEvaluationPeriod

theorem **tailTop_eq_of_pin** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastPeriodTailEqLoweredEvaluationPeriodStatement

theorem **_root_.GlobalTheory.T.lastPeriod_tail_eq_lowered_evaluation_period** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastRowInvFrame

theorem **row_of_smul_eq** {n : ℕ} (Q : GlobalTheory.LatPair.Pair k (Fin n)) (b b' : Matrix (Fin n) (Fin n) k) (hbb' : b ∗ b' = 1) (c : k) (r₀ : Fin n) (hQ : ∀ (x : Fin n → k) (a : 𝒦[k]), Q.HasVal x a ↔ ∃ y : Fin n → 𝒪[k], (∀ r, (y r : k) = (b *ᵥ x) r) ∧ ∃ s : 𝒪[k], (s : k) = c ∧ a = residue 𝒪[k] (s ∗ y r₀)) (g : GL (Fin n) k) (s : 𝒦[k]ˣ) (hg : g • Q = s • Q) (m : Fin n) : ∃ t : 𝒪[k], (t : k) = (b ∗ (g⁻¹).val ∗ b') r₀ m ∧ residue 𝒪[k] t = if m = r₀ then (s : 𝒦[k]) else 0

theorem **resS_eq_rho** {D : D7} (q : GlobalTheory.FinPl.SplitPt D k) {P : HeightOneSpectrum (𝒪 D.host.E)} (he : Formalism.L2.NTheta.IsIdemAt q.e P) (ρ : 𝒦[k] ≃+* (𝒪 D.host.E ⧸ P.asIdeal)) (hp : ∀ (x : 𝒪 D.host.E) (t : 𝒪[k]), (t : k) = q.pr (incA D (x : D.host.E)) → ρ (residue 𝒪[k] t) = Ideal.Quotient.mk P.asIdeal x) (a : Formalism.L2.Herm.intAf D.host.F D.host.E) (t : 𝒪[k]) (ht : (t : k) = q.prF he a.1) : Formalism.L2.DIV.resS D.host.F P a = ρ (residue 𝒪[k] t)

theorem **GlobalTheory.AdForm.lastRow_inv_frame** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, Frame (p i) ((I.rec7 l Z).sp.X j)) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ j (hj : j ≤ B.R.m) (i : ι) (v : (I.atT l u a).vT), Formalism.L2.NTheta.IsIdemAt (p i).e v.1 → ∀ (ρ : 𝒦[k] ≃+* (𝒪 B.fields.E ⧸ v.1.asIdeal)), (∀ (x : 𝒪 B.fields.E) (t : 𝒪[k]), (t : k) = (p i).pr (incA (I.rec7 l Z) (x : B.fields.E)) → ρ (residue 𝒪[k] t) = Ideal.Quotient.mk v.1.asIdeal x) → ∀ (g : GL (Fin ((I.rec7 l Z).sp.X j).rank) k) (s : 𝒦[k]ˣ), g • Q j i = s • Q j i → ∀ δ : TC.GA B (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawCombinationsOperatorDatum

theorem **cA_mem_idAt** {P : Pl D → Prop} {a : AE D} (h : a ∈ idAt D P) : cA D a ∈ idAt D P

theorem **mxAt_adj** {P : Pl D → Prop} {m n : ℕ} {x : Mx D m n} (hx : MxAt P x) : MxAt P (adj x)

theorem **mxAt_mul_l** {P : Pl D → Prop} {l m n : ℕ} (A : Mx D l m) {x : Mx D m n} (hx : MxAt P x) : MxAt P (A ∗ x)

theorem **mxAt_mul_r** {P : Pl D → Prop} {l m n : ℕ} {x : Mx D l m} (hx : MxAt P x) (A : Mx D m n) : MxAt P (x ∗ A)

theorem **mxAt_smul** {P : Pl D → Prop} {m n : ℕ} (c : AE D) {x : Mx D m n} (hx : MxAt P x) : MxAt P (c • x)

theorem **mxAt_add** {P : Pl D → Prop} {m n : ℕ} {x y : Mx D m n} (hx : MxAt P x) (hy : MxAt P y) : MxAt P (x + y)

theorem **mxAt_zero** {P : Pl D → Prop} {m n : ℕ} : MxAt P (0 : Mx D m n)

theorem **combAt_single** {P : Pl D → Prop} {S T : HSp D} {z : WO S T} (hz : MxAt P z) (c : ℂ) : CombAt P (Finsupp.single z c)

theorem **comb_apply** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) (w : WO S T) : comb ψ S T c f w = ∑ z ∈ c.support, c z * heisO ψ S T z f w

theorem **comb_single** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (z : WO S T) (c : ℂ) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single z c) f = c • heisO ψ S T z f

theorem **symplOH_zero_right** (S T : HSp D) (w : WO S T) : symplOH S T w 0 = 0

theorem **ofAdj_starAdj** (S T : HSp D) (x : WO S T) : ofAdj S T (starAdj S T x) = x

theorem **mul_invLine** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hL : L.rank = 1) (hL'' : L''.rank = 1) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) : τ * invLine τ = 1

theorem **kap_kapInv** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hL : L.rank = 1) (hL'' : L''.rank = 1) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (S : HSp D) (b : WO L S) : kap τ S (kapInv τ S b) = b

theorem **mxAt_kapInv** {P : Pl D → Prop} {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (S : HSp D) {b : WO L S} (hb : MxAt P b) : MxAt P (kapInv τ S b)

theorem **mxAt_kap** {P : Pl D → Prop} {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (S : HSp D) {w : WO S L''} (hw : MxAt P w) : MxAt P (kap τ S w)

theorem **kap_add** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (S : HSp D) (w w' : WO S L'') : kap τ S (w + w') = kap τ S w + kap τ S w'

theorem **symplOH_kap** {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (S : HSp D) (w w' : WO S L'') : symplOH L S (kap τ S w) (kap τ S w') = symplOH S L'' w w'

theorem **lastBi_apply** {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ) (u : WO A S) (v : WO S X) : lastBi o o'' τ Ξ Φ₂ u v = Φ₂ (o''.resT S v).1 * Ξ ((o.resS S).symm (u, kap τ S (o''.resT S v).2))

theorem **heisO_split** {V₀ A L S L'' : HSp D} (o : OrthDec V₀ A L) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (ψ : AddChar (AF D.host.F) ℂ) (Ξ : WO V₀ S → ℂ) (x u : WO A S) (y w : WO S L'') : heisO ψ V₀ S ((o.resS S).symm (x, kap τ S y)) Ξ ((o.resS S).symm (u, kap τ S w)) = ψ (symplOH A S u x) * ψ (symplOH S L'' w y) * Ξ ((o.resS S).symm (u + x, kap τ S (w + y)))

theorem **biComb_lastBi** {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (ψ : AddChar (AF D.host.F) ℂ) (d : WO A S →₀ ℂ) (d' : WO S X →₀ ℂ) (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ) : biComb ψ A S X d d' (lastBi o o'' τ Ξ Φ₂) = ∑ p ∈ d.support ×ˢ d'.support, lastBi o o'' τ (comb ψ V₀ S (Finsupp.single ((o.resS S).symm (p.1, kap τ S (o''.resT S p.2).2)) (d p.1 * d' p.2)) Ξ) (comb ψ S T (Finsupp.single (o''.resT S p.2).1 1) Φ₂)

theorem **lastBi_comb** {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hL : L.rank = 1) (hL'' : L''.rank = 1) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (ψ : AddChar (AF D.host.F) ℂ) (e : WO V₀ S →₀ ℂ) (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ) : lastBi o o'' τ (comb ψ V₀ S e Ξ) Φ₂ = ∑ z ∈ e.support, biComb ψ A S X (Finsupp.single (o.resS S z).1 (e z)) (Finsupp.single ((o''.resT S).symm (0, kapInv τ S (o.resS S z).2)) 1) (lastBi o o'' τ Ξ Φ₂)

theorem **sum_equivFin** {α M : Type} [AddCommMonoid M] (s : Finset α) (f : α → M) : ∑ p ∈ s, f p = ∑ i : Fin s.card, f (s.equivFin.symm i : α)

theorem **lastOp** : ∀ D : D7, GlobalTheory.FinPl.LastOp D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawCombinationsOperatorDatumStatement

theorem **_root_.GlobalTheory.T.lastSeeSaw_combinations_operatorDatum** : ∀ D : D7, GlobalTheory.FinPl.LastOp D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumAtSplitPlacesProof

theorem **vac_sum** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P : Pl D → Prop} {κ : Type*} (s : Finset κ) (a : κ → ℂ) (F : κ → WO S T → ℂ) (hF : ∀ i ∈ s, Vac ψ S T LS LT P (F i)) : Vac ψ S T LS LT P (fun w => ∑ i ∈ s, a i * F i w)

theorem **vac_zero** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P : Pl D → Prop} : Vac ψ S T LS LT P 0

theorem **vac_mono** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P Q : Pl D → Prop} (h : ∀ x, Q x → P x) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) : Vac ψ S T LS LT Q f

theorem **biVac_biComb_off** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {A S A' : HSp D} {LA : Lat A} {LS : Lat S} {LA' : Lat A'} {Q P' : Pl D → Prop} (hd : ∀ x, Q x → ¬ P' x) {d : WO A S →₀ ℂ} {d' : WO S A' →₀ ℂ} (hd1 : CombAt Q d) (hd2 : CombAt Q d') {Y : WO A S → WO S A' → ℂ} (hY : BiVacAt ψ LA LS LA' P' Y) : BiVacAt ψ LA LS LA' P' (biComb ψ A S A' d d' Y)

theorem **vac_of_list** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {LS : Lat S} {LT : Lat T} (hLS : IsSelfDualC S LS) (hLT : IsSelfDualC T LT) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : ∀ (l : List (Pl D)), l.Nodup → ∀ (P : Pl D → Prop), (∀ x, P x ↔ x ∈ l) → (∀ x ∈ l, Vac ψ S T LS LT (· = x) f) → Vac ψ S T LS LT P f

theorem **biComb_eq_foldr_vac** {ψ : AddChar (AF D.host.F) ℂ} {A S A' : HSp D} {LA : Lat A} {LS : Lat S} {LA' : Lat A'} {P : Pl D → Prop} (c : WO A S →₀ ℂ) (c' : WO S A' →₀ ℂ) (l : List ((WO A S →₀ ℂ) × (WO S A' →₀ ℂ))) (hc : ∀ f, Vac ψ A S LA LS P f → comb ψ A S c f = l.foldr (fun p f' => comb ψ A S p.1 f') f) (hc' : ∀ f, Vac ψ S A' LS LA' P f → comb ψ S A' c' f = l.foldr (fun p f' => comb ψ S A' p.2 f') f) (Y : WO A S → WO S A' → ℂ) (hY : BiVacAt ψ LA LS LA' P Y) : biComb ψ A S A' c c' Y = l.foldr (fun p C => biComb ψ A S A' p.1 p.2 C) Y

theorem **overAll_cons'** {i : ι} {J : List ι} {f f' : WO S T → ℂ} : OverAll ψ α β LS LT Ψ (i :: J) f f' ↔ ∃ g, OverAll ψ α β LS LT Ψ J f g ∧ Over ψ (α i) (β i) LS LT g (Ψ i) f'

theorem **overAll_steps** : ∀ (J : List ι) (f f' : WO S T → ℂ), OverAll ψ α β LS LT Ψ J f f' → ∀ j ∈ J, ∃ g g', Over ψ (α j) (β j) LS LT g (Ψ j) g'

theorem **vac_foldr_off** (hψ : IsStdChar D ψ (-1)) {x : Pl D} {κ : Type} (r : κ → WO S T →₀ ℂ) (P' : κ → Pl D → Prop) (l : List κ) (hr : ∀ j ∈ l, CombAt (P' j) (r j)) (hd : ∀ j ∈ l, ¬ P' j x) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT (· = x) f) : Vac ψ S T LS LT (· = x) (l.foldr (fun j f' => comb ψ S T (r j) f') f)

theorem **overAll_foldr** (hψ : IsStdChar D ψ (-1)) (hinj : Function.Injective fun i => (p i).v) (r : ι → WO S T →₀ ℂ) (hr1 : ∀ i, CombAt (· = (p i).v) (r i)) (hr2 : ∀ i f₀, Vac ψ S T LS LT (· = (p i).v) f₀ → Over ψ (α i) (β i) LS LT f₀ (Ψ i) (comb ψ S T (r i) f₀)) : ∀ (l

: List ι), l.Nodup → ∀ f, (∀ i ∈ l, Vac ψ S T LS LT (· = (p i).v) f) → OverAll ψ α β LS LT Ψ l f (l.foldr (fun i f' => comb ψ S T (r i) f') f)

theorem **comb_eq_foldr** (hP : Places p I P) (hψ : IsStdChar D ψ (-1)) (hLS : IsSelfDualC S LS) (hLT : IsSelfDualC T LT) {c : WO S T →₀ ℂ} (hc : Reps ψ α β LS LT Ψ I P c) (r : ι → WO S T →₀ ℂ) (hr1 : ∀ i, CombAt (· = (p i).v) (r i)) (hr2 : ∀ i f₀, Vac ψ S T LS LT (· = (p i).v) f₀ → Over ψ (α i) (β i) LS LT f₀ (Ψ i) (comb ψ S T (r i) f₀)) : ∀ f, Vac ψ S T LS LT P f → comb ψ S T c f = I.foldr (fun i f' => comb ψ S T (r i) f') f

theorem **over_of_overAll** (hP : Places p I P) (hψ : IsStdChar D ψ (-1)) {f Ξ : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) (hΞ : OverAll ψ α β LS LT Ψ I f Ξ) (i : ι) : ∃ g, Over ψ (α i) (β i) LS LT g (Ψ i) Ξ

theorem **main** : ∀ D : D7, GlobalTheory.FinPl.LastTAll D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumAtSplitPlacesProofStatement

theorem **_root_.GlobalTheory.T.lastSeeSawDatumAtSplitPlaces_proof** : ∀ D : D7, GlobalTheory.FinPl.LastTAll D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumExistsAtLevelPrimePlace

theorem **comb_apply'** {S T : HSp D} (c : WO S T →₀ ℂ) (g : WO S T → ℂ) (u : WO S T) : comb ψ S T c g u = ∑ x ∈ c.support, c x * (ψ (symplOH S T u x) * g (u + x))

theorem **isThetaFunO_add'** {S T : HSp D} {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : IsThetaFunO ψ S T (f + g) where

theorem **heisO_smul'** {S T : HSp D} (u : WO S T) (r : ℂ) (f : WO S T → ℂ) : heisO ψ S T u (r • f) = r • heisO ψ S T u f

theorem **heisO_add'** {S T : HSp D} (u : WO S T) (f g : WO S T → ℂ) : heisO ψ S T u (f + g) = heisO ψ S T u f + heisO ψ S T u g

theorem **heisO_zero'** {S T : HSp D} (u : WO S T) : heisO ψ S T u (0 : WO S T → ℂ) = 0

theorem **isThetaFunO_zero'** (S T : HSp D) : IsThetaFunO ψ S T (0 : WO S T → ℂ) where

theorem **isThetaFunO_smul'** {S T : HSp D} (r : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (r • f) where

theorem **vac_zero** : Vac ψ S T LS LT P 0

theorem **vac_add** {f g : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) (hg : Vac ψ S T LS LT P g) : Vac ψ S T LS LT P (f + g)

theorem **vac_smul** (r : ℂ) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) : Vac ψ S T LS LT P (r • f)

theorem **vac_sum** {ι : Type} (s : Finset ι) (r : ι → ℂ) (f : ι → WO S T → ℂ) (h : ∀ i ∈ s, Vac ψ S T LS LT P (f i)) : Vac ψ S T LS LT P (∑ i ∈ s, r i • f i)

theorem **comb_smul_fun** (c : WO S T →₀ ℂ) (r : ℂ) (f : WO S T → ℂ) : comb ψ S T c (r • f) = r • comb ψ S T c f

theorem **comb_sum_fun** {ι : Type} (c : WO S T →₀ ℂ) (s : Finset ι) (r : ι → ℂ) (f : ι → WO S T → ℂ) : comb ψ S T c (∑ i ∈ s, r i • f i) = ∑ i ∈ s, r i • comb ψ S T c (f i)

theorem **comb_swap** {A S X : HSp D} (c : WO A S →₀ ℂ) (c' : WO S X →₀ ℂ) (G : WO A S → WO S X → ℂ) (u : WO A S) (x : WO S X) : comb ψ S X c' (fun x' => comb ψ A S c (fun u' => G u' x') u) x = comb ψ A S c (fun u' => comb ψ S X c' (G u') x) u

theorem **lastDatum** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (χ : AddChar k Circle) (hχ : HasConductor χ 1), p.IsChar ψ χ → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (ζ : Fin L.rank → D.host.E) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L ζ → IsLineVec L'' ζ₀ → ∀ (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (lam : D.host.E), Matrix.mulVec τ ζ₀ = lam • ζ → valuation k (p.pr (incA D lam)) = 1 → τ.conjTranspose * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∃ ε : 𝓀[k]ˣ, ∀ (α : Frame p A n') (α₀ : Frame p V₀ (n' ⊕ Fin 1)) (β : Frame p S n) (γ : Frame p T n'') (γ₀ : Frame p X (n''

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumExistsAtLevelPrimePlaceStatement

theorem **_root_.GlobalTheory.T.lastSeeSawDatum_exists_at_levelPrimePlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (χ : AddChar k Circle) (hχ : HasConductor χ 1), p.IsChar ψ χ → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (ζ : Fin L.rank → D.host.E) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L ζ → IsLineVec L'' ζ₀ → ∀ (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (lam : D.host.E), Matrix.mulVec τ ζ₀ = lam • ζ → valuation k (p.pr (incA D lam)) = 1 → τ.conjTranspose * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∃ ε : 𝓀[k]ˣ, ∀ (α : Frame p A n') (α₀ : Frame p V₀ (n' ⊕ Fin 1)) (β :

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumIsTwoVariableThetaFunction

theorem **sOH_zero_left** (S T : HSp D) (y : WO S T) : symplOH S T 0 y = 0

theorem **sOH_zero_right** (S T : HSp D) (x : WO S T) : symplOH S T x 0 = 0

theorem **sOH_resS_symm** {V₀ A L : HSp D} (o : OrthDec V₀ A L) (S : HSp D) (a a' : WO A S) (b b' : WO L S) : symplOH V₀ S ((o.resS S).symm (a, b)) ((o.resS S).symm (a', b')) = symplOH A S a a' + symplOH L S b b'

theorem **quasi_of_mem** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {f : WO S T → ℂ} (h : IsThetaFunO ψ S T f) {x : WO S T} (hx : x ∈ ratMx D T.rank S.rank) (w : WO S T) : f (x + w) = ψ (-(symplOH S T x w)) * f w

theorem **resS_symm_mem** {V₀ A L : HSp D} (o : OrthDec V₀ A L) (S : HSp D) {a : WO A S} {b : WO L S} (ha : a ∈ ratMx D S.rank A.rank) (hb : b ∈ ratMx D S.rank L.rank) : (o.resS S).symm (a, b) ∈ ratMx D S.rank V₀.rank

theorem **resS_symm_map** {B : Type} [CommRing B] (φ : AE D →+* B) {V₀ A L : HSp D} (o : OrthDec V₀ A L) (S : HSp D) {a : WO A S} {b : WO L S} (ha : a.map φ = 0) (hb : b.map φ = 0) : ((o.resS S).symm (a, b)).map φ = 0

theorem **isBiTheta_lastBi** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∀ (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ), IsThetaFunO ψ V₀ S Ξ → IsThetaFunO ψ S T Φ₂ → IsBiTheta ψ A S X (lastBi o o'' τ Ξ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumIsTwoVariableThetaFunctionStatement

theorem **_root_.GlobalTheory.T.lastSeeSaw_datum_isTwoVariableThetaFunction** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∀ (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ), IsThetaFunO ψ V₀ S Ξ → IsThetaFunO ψ S T Φ₂ → IsBiTheta ψ A S X (lastBi o o'' τ Ξ Φ₂)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumOfVacuaIsVacuumBoth

theorem **lastBi_apply** {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ) (u : WO A S) (v : WO S X) : lastBi o o'' τ Ξ Φ₂ u v = Φ₂ (o''.resT S v).1 * Ξ ((o.resS S).symm (u, kap τ S (o''.resT S v).2))

theorem **heis_left** (ψ : AddChar (AF D.host.F) ℂ) {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ) (a : WO A S) (v : WO S X) : heisO ψ A S a (fun u => lastBi o o'' τ Ξ Φ₂ u v) = fun u => Φ₂ (o''.resT S v).1 * heisO ψ V₀ S ((o.resS S).symm (a, 0)) Ξ ((o.resS S).symm (u, kap τ S (o''.resT S v).2))

theorem **heis_right** (ψ : AddChar (AF D.host.F) ℂ) {V₀ A L S X T L'' : HSp D} (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (hτ : τᴴ * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram) (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ) (b : WO S X) (u : WO A S) : heisO ψ S X b (lastBi o o'' τ Ξ Φ₂ u) = o''.boxT S (heisO ψ S T (o''.resT S b).1 Φ₂) (fun w => heisO ψ V₀ S ((o.resS S).symm (0, kap τ S (o''.resT S b).2)) Ξ ((o.resS S).symm (u, kap τ S w)))

theorem **slice_left** {ψ : AddChar (AF D.host.F) ℂ} {A S A' : HSp D} {C : WO A S → WO S A' → ℂ} (h : IsBiTheta ψ A S A' C) (v : WO S A') : IsThetaFunO ψ A S fun u => C u v where

theorem **slice_right** {ψ : AddChar (AF D.host.F) ℂ} {A S A' : HSp D} {C : WO A S → WO S A' → ℂ} (h : IsBiTheta ψ A S A' C) (u : WO A S) : IsThetaFunO ψ S A' (C u) where

theorem **isHalf_mul_left** {e : AE D} {k m n : ℕ} (P : Mx D k m) {y : Mx D m n} (h : IsHalf e y) : IsHalf e (P * y)

theorem **isHalf_mul_right** {e : AE D} {k m n : ℕ} {y : Mx D k m} (P : Mx D m n) (h : IsHalf e y) : IsHalf e (y * P)

theorem **isHalf_add** {e : AE D} {m n : ℕ} {y z : Mx D m n} (hy : IsHalf e y) (hz : IsHalf e z) : IsHalf e (y + z)

theorem **isHalf_zero** (e : AE D) (m n : ℕ) : IsHalf e (0 : Mx D m n)

theorem **isHalf_smul** {e : AE D} {m n : ℕ} (c : AE D) {y : Mx D m n} (h : IsHalf e y) : IsHalf e (c • y)

theorem **isHalf_adj** {e : AE D} {m n : ℕ} {y : Mx D m n} (h : IsHalf e y) : IsHalf (cA D e) (adj y)

theorem **isHalf_kap** {e : AE D} {L L'' : HSp D} (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (S : HSp D) {w : WO S L''} (h : IsHalf e w) : IsHalf (cA D e) (kap τ S w)

theorem **cA_cA** (x : AE D) : cA D (cA D x) = x

theorem **isSplitIdem_cA** {e : AE D} (he : IsSplitIdem D e) : IsSplitIdem D (cA D e)

theorem **isHalf_resS_symm** {e : AE D} {V₀ A L : HSp D} (o : OrthDec V₀ A L) (S : HSp D) {a : WO A S} {b : WO L S} (ha : IsHalf e a) (hb : IsHalf e b) : IsHalf e ((o.resS S).symm (a, b))

theorem **isHalf_resT** {e : AE D} {X T L'' : HSp D} (o'' : OrthDec X T L'') (S : HSp D) {w : WO S X} (h : IsHalf e w) : IsHalf e (o''.resT S w).1 ∧ IsHalf e (o''.resT S w).2

theorem **smul_latV_mem_latA** {S : HSp D} (L : Lat S) (P : Pl D → Prop) {c : D.host.E} (hc : c ∈ D.host.AA) {w : Fin S.rank → AE D} (hw : w ∈ latV S L P) : ((1 : AF D.host.F) ⊗ₖ[D.host.F] c : AE D) • w ∈ latA S L P

theorem **latA_orth** {S A B : HSp D} (o : OrthDec S A B) (L : Lat S) (LA : Lat A) (LB : Lat B) (x : Pl D) (hO : OrthAt o L LA LB x) {z : Fin S.rank → AE D} (hz : z ∈ latA S L (· = x)) : o.pA *ᵥ z ∈ latA A LA (· = x) ∧ o.pB *ᵥ z ∈ latA B LB (· = x)

theorem **extLat** : ∀ (D : D7) (V₀ A L S : HSp D) (o : OrthDec V₀ A L) (M : Lat V₀) (LA : Lat A) (LL : Lat L) (LS : Lat S) (x : Pl D), IsSelfDualC V₀ M → IsSelfDualC A LA → IsSelfDualC L LL → IsSelfDualC S LS → OrthAt o M LA LL x → ∀ (a : WO A S) (b : WO L S), a ∈ latO A S (LA : Set _) (LS : Set _) (· = x) → b ∈ latO L S (LL : Set _) (LS : Set _) (· = x) → (o.resS S).symm (a, b) ∈ latO V₀ S (M : Set _) (LS : Set _) (· = x)

theorem **resLat** : ∀ (D : D7) (X T L'' S : HSp D) (o'' : OrthDec X T L'') (LX : Lat X) (LT : Lat T) (LZ : Lat L'') (LS : Lat S) (x : Pl D), IsSelfDualC X LX → IsSelfDualC T LT → IsSelfDualC L'' LZ → IsSelfDualC S LS → OrthAt o'' LX LT LZ x → ∀ w : WO S X, w ∈ latO S X (LS : Set _) (LX : Set _) (· = x) → (o''.resT S w).1 ∈ latO S T (LS : Set _) (LT : Set _) (· = x) ∧ (o''.resT S w).2 ∈ latO S L'' (LS : Set _) (LZ : Set _) (· = x)

theorem **lastVac** : ∀ D : D7, GlobalTheory.FinPl.LastVac D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDatumOfVacuaIsVacuumBothStatement

theorem **_root_.GlobalTheory.T.lastSeeSawDatum_of_vacua_is_vacuum_both** : ∀ D : D7, GlobalTheory.FinPl.LastVac D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDoubleLiftAdditive

theorem **biLift_eq** {A S X : HSp D} (oma : Om A S) (omX : Om S X) (vA : Vol A) (vS : Vol S) (φ : GA D A → ℂ) (C : WO A S → WO S X → ℂ) (h₂ : GA D X) : biLift oma omX vA vS φ C h₂ = ∫ h₁ in vS.𝓕, (∫ h₀ in vA.𝓕, φ h₀ * ker oma omX C h₀ h₁ h₂ ∂vA.μ) ∂vS.μ

theorem **ker_sum** {n : ℕ} (C : Fin n → WO A S → WO S X → ℂ) (h₀ : GA D A) (h₁ : GA D S) (h₂ : GA D X) : ker oma omX (∑ i, C i) h₀ h₁ h₂ = ∑ i, ker oma omX (C i) h₀ h₁ h₂

theorem **ker_cont** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) : Continuous fun x : GA D A × GA D S × GA D X => ker oma omX C x.1 x.2.1 x.2.2

theorem **ker_left** (hwa : IsWeilDatumO ψ A S χA χS oma) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) {γ : GA D A} (hγ : γ ∈ GQ A) (h₀ : GA D A) (h₁ : GA D S) (h₂ : GA D X) : ker oma omX C (γ * h₀) h₁ h₂ = ker oma omX C h₀ h₁ h₂

theorem **ker_mid** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) {γ : GA D S} (hγ : γ ∈ GQ S) (h₀ : GA D A) (h₁ : GA D S) (h₂ : GA D X) : ker oma omX C h₀ (γ * h₁) h₂ = ker oma omX C h₀ h₁ h₂

theorem **ker_right** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) {δ : GA D X} (hδ : δ ∈ GQ X) (h₀ : GA D A) (h₁ : GA D S) (h₂ : GA D X) : ker oma omX C h₀ h₁ (δ * h₂) = ker oma omX C h₀ h₁ h₂

theorem **inner** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) (hA : CoCpt A) (vA : Vol A) {φ : GA D A → ℂ} (hφ : IsContAut A φ) : (∀ h₁ h₂, IntegrableOn (fun h₀ => φ h₀ * ker oma omX C h₀ h₁ h₂) vA.𝓕 vA.μ) ∧ Continuous fun p : GA D S × GA D X => ∫ h₀ in vA.𝓕, φ h₀ * ker oma omX C h₀ p.1 p.2 ∂vA.μ

theorem **mid** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) (hA : CoCpt A) (hS : CoCpt S) (vA : Vol A) (vS : Vol S) {φ : GA D A → ℂ} (hφ : IsContAut A φ) : (∀ h₂, IntegrableOn (fun h₁ => ∫ h₀ in vA.𝓕, φ h₀ * ker oma omX C h₀ h₁ h₂ ∂vA.μ) vS.𝓕 vS.μ) ∧ Continuous fun h₂ => biLift oma omX vA vS φ C h₂

theorem **outer** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) {C : WO A S → WO S X → ℂ} (hC : IsBiTheta ψ A S X C) (hA : CoCpt A) (hS : CoCpt S) {T L'' : HSp D} (hT : CoCpt T) (vA : Vol A) (vS : Vol S) (vT : Vol T) (o'' : OrthDec X T L'') {φ : GA D A → ℂ} (hφ : IsContAut A φ) {F : GA D T → ℂ} (hF : IsContAut T F) : IntegrableOn (fun h' => F h' * biLift oma omX vA vS φ C (o''.emb (h', 1))) vT.𝓕 vT.μ

theorem **biLift_sum** (hB : BiAct D) (hwa : IsWeilDatumO ψ A S χA χS oma) (hwX : IsWeilDatumO ψ S X χS' χX omX) (hA : CoCpt A) (hS : CoCpt S) (vA : Vol A) (vS : Vol S) {φ : GA D A → ℂ} (hφ : IsContAut A φ) {n : ℕ} (C : Fin n → WO A S → WO S X → ℂ) (hC : ∀ i, IsBiTheta ψ A S X (C i)) (h₂ : GA D X) : biLift oma omX vA vS φ (∑ i, C i) h₂ = ∑ i, biLift oma omX vA vS φ (C i) h₂

theorem **seeSmall** : ∀ D : D7, SeeSmall D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawDoubleLiftAdditiveStatement

theorem **_root_.GlobalTheory.T.lastSeeSaw_doubleLift_additive** : ∀ D : D7, GlobalTheory.FinPl.SeeSmall D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawThetaIdentity

theorem **ofAdj_zero** (D : D7) (L S : HSp D) : GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj L S 0 = 0

theorem **seeLast** : ∀ D : D7, SeeLast D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeeSawThetaIdentityStatement

theorem **_root_.GlobalTheory.T.lastSeeSaw_thetaIdentity** : ∀ D : D7, SeeLast D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeesawDatumEqLatticePairOverVacuum

theorem **tensRow_eq_phiOne** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n n'' : Type} [Fintype n] [DecidableEq n] [Fintype n''] [DecidableEq n''] (q : Unif k) (Q : Pair k n) (Λ : Submodule 𝒪[k] (n'' → k)) [Λ.IsLattice k] (c : 𝓀[k]ˣ) : tensRow (latInd Q.lat Λ) (cosetInd q Q.lat (c • Q).red) = phiOne (c • Q) (lastPair q Λ)

theorem **charInd_ne_zero** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : Type} [Fintype n] (χ : AddChar 𝓀[k] ℂ) (Q : Pair k n) : charInd χ Q ≠ 0

theorem **lastTarget** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (χ : AddChar k Circle) (hχ : HasConductor χ 1), p.IsChar ψ χ → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (ζ : Fin L.rank → D.host.E) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L ζ → IsLineVec L'' ζ₀ → ∀ (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (lam : D.host.E), Matrix.mulVec τ ζ₀ = lam • ζ → valuation k (p.pr (incA D lam)) = 1 → τ.conjTranspose * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∃ ε : 𝓀[k]ˣ, ∀ (α : Frame p A n') (α₀ : Frame p V₀ (n' ⊕ Fin 1)) (β : Frame p S n) (γ : Frame p T n'') (γ₀ : Frame p X (n''

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSeesawDatumEqLatticePairOverVacuumStatement

theorem **_root_.GlobalTheory.T.lastSeesaw_datum_eq_lattice_pair_over_vacuum** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (χ : AddChar k Circle) (hχ : HasConductor χ 1), p.IsChar ψ χ → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (ζ : Fin L.rank → D.host.E) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L ζ → IsLineVec L'' ζ₀ → ∀ (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (lam : D.host.E), Matrix.mulVec τ ζ₀ = lam • ζ → valuation k (p.pr (incA D lam)) = 1 → τ.conjTranspose * (L.gram * τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∃ ε : 𝓀[k]ˣ, ∀ (α : Frame p A n') (α₀ : Frame p V₀ (n' ⊕ Fin 1)) (β

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSpaceMovedLevelClassPointsRepresentatives

theorem **hasMat_iff** (h : GA D D.sp.X') (y) : GlobalTheory.Core.Rk.Top.HasMat h y ↔ h ∈ GfC D.sp.X' ∧ finM D.sp.X' (Formalism.L2.Forms.rankP D) h = y.1

theorem **nontrivial_finAdeles** : Nontrivial (FiniteAdeleRing (𝒪 D.host.F) D.host.F)

theorem **inclE_inj** : Function.Injective (Formalism.UG.inclE (F

theorem **bc_inclE_inj** {n : ℕ} : Function.Injective (Formalism.UG.bc (Formalism.UG.inclE (F

theorem **eq_one_of_GQ** {V : HSp D} {γ : GA D V} (hγ : γ ∈ GQ V) (h1 : finM V rfl γ = 1) : γ = 1

theorem **eq_one_of_GQ'** {V : HSp D} {n : ℕ} (hn : V.rank = n) {γ : GA D V} (hγ : γ ∈ GQ V) (h1 : finM V hn γ = 1) : γ = 1

theorem **finGLHom_injOn** {V : HSp D} {n : ℕ} (hn : V.rank = n) {γ₁ γ₂ : GA D V} (h1 : γ₁ ∈ GQ V) (h2 : γ₂ ∈ GQ V) (e : GlobalTheory.ClassSumLevel.finGLHom V hn γ₁ = GlobalTheory.ClassSumLevel.finGLHom V hn γ₂) : γ₁ = γ₂

theorem **card_eq_of_lift** {G A : Type} [Group G] (H : Subgroup G) (p : G → Prop) (q : A → Prop) (ρ : A → G) (hp : ∀ a, ρ a ∈ H) (hpq : ∀ a, p (ρ a) ↔ q a) (hi : ∀ a b, ρ a = ρ b → a = b) (hs : ∀ γ ∈ H, ∃ a, ρ a = γ) : Nat.card {γ : H // p γ.1} = Nat.card {a // q a}

theorem **isReps_of_level** (V : HSp D) {n : ℕ} (hn : V.rank = n) (hV : ∀ τ, (gramAt D V τ).PosDef) {T f : ℕ} {bad : Set (HeightOneSpectrum (𝒪 D.host.F))} (S : StepLevel D.host.F D.host.E (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) T f bad) {cl : Type} (yc : cl → Formalism.UG.UPts (sigD D) (V.gram.reindex (finCongr hn) (finCongr hn)) (FiniteAdeleRing (𝒪 D.host.F) D.host.F)) (hy : Function.Bijective (fun c => (Quotient.mk'' (yc c) : Cls S))) (h : cl → GA D V) (hf : ∀ c, h c ∈ GfC V) (hh : ∀ c, finM V hn (h c) = (yc c).1.1) (U : Subgroup (GA D V)) (mem : ∀ x, x ∈ U ↔ x ∈ GfC V ∧ x ∈ GlobalTheory.ClassSumLevel.levG V hn S) : IsReps V U h ∧ ∀ c, stabOrd U (h c) = Nat.card {δ : Formalism.UG.UJ (sigD D : D.host.E →+* D.host.E) (V.gram.reindex (finCongr hn) (finCongr hn)) // Formalism.UG.bc (Formalism.UG.inclE (F

theorem **isReps_stabOrd_top** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastSpaceMovedLevelClassPointsRepresentativesStatement

theorem **_root_.GlobalTheory.T.last_space_moved_level_class_points_representatives** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l

Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → ∀ j : SmT R.MX →ₗ[ℂ] (FrP (I.rec7 l Z) → ℂ), GlobalTheory.Core.Rk.Top.IsJ u R.MX j → JOf R j J → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepChartBasePointCarriesStdPoint

theorem **sel_under_of_mem_QE** {B : Base} {X : PreX B.toPre} (I : TC B X) (hI : I.Ok) {b : I.ι} {P : HeightOneSpectrum (𝒪 B.fields.E)} (hP : P ∈ I.Q b) : I.sel (P.under (𝒪 B.fields.F)) = P

theorem **topBase_ok** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) {C : Type} (chOf : (I.stp u a B.R.m).Cl → C) (Lb : C → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)) (U : C → Subgroup (TC.GA B (B.R.N B.R.m))) (chi : ∀ β, U β →* ℂˣ) (y : (I.stp u a B.R.m).Cl → TC.GA B (B.R.N B.R.m)) (st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E) (k : C → TC.GA B (B.R.N B.R.m)), I.Reads u a B.R.m chOf Lb U chi y st k → (∀ β, I.TopPt l Z u a (k β)) → ∀ β, (∃ q : Formalism.L2.Col.ChPt (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepChartBasePointCarriesStdPointStatement

theorem **_root_.GlobalTheory.T.lastStepChartBasePoint_carries_stdPoint** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) {C : Type} (chOf : (I.stp u a B.R.m).Cl → C) (Lb : C → Submodule (𝒪 B.host.E) (Fin (B.R.N B.R.m) → B.host.E)) (U : C → Subgroup (TC.GA B (B.R.N B.R.m))) (chi : ∀ β, U β →* ℂˣ) (y : (I.stp u a B.R.m).Cl → TC.GA B (B.R.N B.R.m)) (st : ∀ c, (I.stp u a B.R.m).Stab c → GL (Fin (B.R.N B.R.m)) B.host.E) (k : C → TC.GA B (B.R.N B.R.m)), I.Reads u a B.R.m chOf Lb U chi y st k → (∀ β, I.TopPt l Z u a (k β)) → ∀ β, (∃ q : Formalism.L2.Col.ChPt (Pr

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepDatumTransportBadPrime

theorem **locTop_shiftQ** (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∉ Pr.vGs) (hD : 2 * cst.M ≤ cst.D P) (y : HomE X.E (R.N (m2 + 1)) (R.N R.m)) {a b : Sp X.E (R.N (m2 + 1))} (ha : ∀ i, IntAt P.1 ((qOf P.1 : X.E) ^ cst.M * a i)) (hb : ∀ i, IntAt P.1 (((qOf P.1 : X.E) ^ cst.M)⁻¹ * (b i - a i))) : locTop Ch sel mu cst m2 P y b = locTop Ch sel mu cst m2 P y a

theorem **locTop_congQ** (hm : R.m = m2 + 2) (hc : cst.Ok) {P : rowSG Pr sel mu} (hP : P.1 ∉ Pr.vGs) (hD : 2 * cst.M ≤ cst.D P) (y z : HomE X.E (R.N (m2 + 1)) (R.N R.m)) (hyz : ∀ s l, IntAt P.1 (((qOf P.1 : X.E) ^ (cst.M + 1))⁻¹ * (y (Pi.single s 1) l - z (Pi.single s 1) l))) {a : Sp X.E (R.N (m2 + 1))} (ha : ∀ i, IntAt P.1 ((qOf P.1 : X.E) ^ cst.M * a i)) : locTop Ch sel mu cst m2 P y a = locTop Ch sel mu cst m2 P z a

theorem **rep_sub_intQ** {u : Pr.Word} {j : ℕ} (h : j + 1 = R.m) (p q : PtAt R sel mu cst u j) (e : cosetIn q = cosetIn p) {P : rowSG Pr sel mu} (hP : P.1 ∉ Pr.vGs) (i : Fin (R.N j)) : IntAt P.1 (((qOf P.1 : X.E) ^ cst.M)⁻¹ * ((frsF sel mu p.1.1 P).symm (repY p.1.2) i - (frsF sel mu q.1.1 P).symm (repY q.1.2) i))

theorem **locTop_transport_tw** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (ok : I.Ok) (u : B.Word) (a : I.ι) (n : ℕ) (h : n + 1 + 1 = B.R.m) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup a)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u (n + 1)) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (n + 1 + 1)) (k y : TC.GA B (B.R.N (n + 1))) (k' y' : TC.GA B (B.R.N (n + 1 + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (n + 1)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (n + 1 + 1)) → TC.PtOf (y * k) q.1.1 → TC.PtOf (y' * k') q'.1.1 → cosetIn q = cosetIn p → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N (n + 1)) (B.R.N (n + 1 + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → GlobalTheory.TopStep.locTop I.Ch I.sel (I.tup a) (I.cst a) n P (castSp B.fields.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepDatumTransportBadPrimeStatement

theorem **_root_.GlobalTheory.T.lastStep_datum_transport_badPrime** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (ok : I.Ok) (u : B.Word) (a : I.ι) (n : ℕ) (h : n + 1 + 1 = B.R.m) (P : Formalism.L2.Col.rowSG (B.primes X) I.sel (I.tup a)) (p q : PtAt B.R I.sel (I.tup a) (I.cst a) u (n + 1)) (p' q' : PtAt B.R I.sel (I.tup a) (I.cst a) u (n + 1 + 1)) (k y : TC.GA B (B.R.N (n + 1))) (k' y' : TC.GA B (B.R.N (n + 1 + 1))), TC.PtOf k p.1.1 → p.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (n + 1)) → TC.PtOf k' p'.1.1 → p'.1.1.deco.1.N = Formalism.L2.Col.stdLat B.fields.E (B.R.N (n + 1 + 1)) → TC.PtOf (y * k) q.1.1 → TC.PtOf (y' * k') q'.1.1 → cosetIn q = cosetIn p → ∃ M : ℕ, 0 < M ∧ ∀ M₀ : ℕ, M ∣ M₀ → ∀ x x₀ : Formalism.L2.Col.HomE B.fields.E (B.R.N (n + 1)) (B.R.N (n + 1 + 1)), GlobalTheory.Transport.CongAt B P.1 M₀ y y' x x₀ → GlobalTheory.TopStep.locTop I.Ch I.sel (I.tu

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepTestFunctionEqColumnsTestFunction

theorem **topPhi** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepTestFunctionEqColumnsTestFunctionStatement

theorem **_root_.GlobalTheory.T.lastStepTestFunction_eq_columnsTestFunction** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepTestFunctionsAgreeRationalPoints

theorem **stdOrd_top** {κ τ s : ℕ} (b : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) (s + 1)) : ∃ b', stdOrd (Fin κ) (Fin τ) (s + 1) 0 rfl b = Sum.inl b' ∧ xPos b' = zPos b

theorem **top_entries** (q : ℕ) (ι : E →+* k) {s κ : ℕ} (S : SymData k s κ) (τ n' n : ℕ) (e' : XIx (Fin κ) (Fin τ) (s + 1) ⊕ Fin 0 ≃ Fin (n' + 0)) (e : Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) (s + 2) ≃ Fin (n + 0)) (he' : IsOrd (yPos xPos n') e') (he : IsOrd zPos e) (x : (Fin (n' + 0) → E) →ₗ[E] (Fin (n + 0) → E)) : (S.T (Fin τ)).PhiTop (toLoc ι e e' x) = S.P.iterPhi (Fin τ) S.μ S.tame (s + 1) (Matrix.of fun a b => ι (LocE.entB q κ 0 n' 0 n (fun i => i + 1) (fun i => i + 1) x (zPos a) (zPos b)))

theorem **bottom_iff** {P : HeightOneSpectrum (𝒪 E)} {q : ℕ} {ι : E →+* k} {ϖ : kˣ} (hC : IsCompl P q ι ϖ) (hU : GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ) (κ l τ j : ℕ) (M : ℕ → ℕ → E) : MaxM ((Matrix.of fun a b => ι (M (zPos a) (zPos b)) : Matrix (Ix (Fin κ ⊕ Fin l) (Fin τ) (j + 1)) (Ix (Fin κ ⊕ Fin l) (Fin τ) j) k).submatrix (ixSucc (Fin κ ⊕ Fin l) (Fin τ) j) id).toRows₂ ↔ ∀ p p', κ + l + j * τ ≤ p → p < κ + l + (j + 1) * τ → p' < κ + l + j * τ → IntAt P (M p p' / (q : E))

theorem **unit_char** {P : HeightOneSpectrum (𝒪 E)} {q : ℕ} {ι : E →+* k} {σ₀ : K →+* ℂ} {ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K} {χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ} {s κ : ℕ} {S : SymData k s κ} (hT : Comparison₂ P q ι σ₀ ψ₀ χ S) (i : ℕ) (d : E) (u : kˣ) (hd : IntAt P d) (hu : valuation k (u : k) = 1) (e : (u : k) = ι d) : (S.μ (i + 1) u : ℂ) = σ₀ (LocE.unitV (χ i) d)

theorem **intAt_div** {P : HeightOneSpectrum (𝒪 E)} {q : ℕ} {a : E} (h : IntAt P (a / (q : E))) (hq : (q : E) ≠ 0) : IntAt P a

theorem **core** (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K) (χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) {s κ : ℕ} (S : SymData k s κ) (hT : Comparison₂ P q ι σ₀ ψ₀ χ S) (τ j N : ℕ) (hN : N = κ + s + 1 + (j + 1) * τ) (M : ℕ → ℕ → E) : S.P.iterPhi (Fin τ) S.μ S.tame j (Matrix.of fun a b => ι (M (zPos a) (zPos b))) = σ₀ (phiM q ψ₀ χ κ S.ρ (κ + s + 1) τ j N M)

theorem **GlobalTheory.SymBridge.phiTopAgree_of_tie** : ∀ (E K k : Type) [Field E] [NumberField E] [StarRing E] [Field K] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K) (χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (s κ : ℕ) (S : SymData k s κ), Comparison₂ P q ι σ₀ ψ₀ χ S → PhiTopAgree P q ι σ₀ ψ₀ χ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepTestFunctionsAgreeRationalPointsStatement

theorem **_root_.GlobalTheory.T.lastStep_testFunctions_agree_rationalPoints** : ∀ (E K k : Type) [Field E] [NumberField E] [StarRing E] [Field K] [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (P : HeightOneSpectrum (𝒪 E)) (q : ℕ) (ι : E →+* k) (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 E ⧸ P.asIdeal) K) (χ : ℕ → (𝒪 E ⧸ P.asIdeal)ˣ →* Kˣ) (s κ : ℕ) (S : SymData k s κ), Comparison₂ P q ι σ₀ ψ₀ χ S → PhiTopAgree P q ι σ₀ ψ₀ χ S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepsKernelEqSumCouplingCosets

theorem **kernel_eq_sum** : ∀ (D : D7) (c : Comm D) (v : VCom c) χ (Q : QAt v χ), c.Is → v.Is → ∀ (h : Q.s + 2 * 0 + 2 = D.R.m), c.Lz.rank = 1 → (v.τ Q.s h)ᴴ * ((c.bes Q.s 0 h).L.gram * v.τ Q.s h) = (-(D.host.xi0 * star D.host.xi0)) • c.Lz.gram → ∀ s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h), s₁.χW = χ Q.s → ∀ Ψ₁ Ψ₂, GlobalTheory.FinPl.Vac c.ψ _ _ (v.N Q.s) (v.N (Q.s + 1)) (· = Q.q.v) Ψ₁ → GlobalTheory.FinPl.Vac c.ψ _ _ (v.N (Q.s + 1)) v.NT (· = Q.q.v) Ψ₂ → ∀ α (S : Finset ((Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂)), Q.T.coupIdx Q.A (box _ Q.c₁) (box _ Q.c₂) ⊆ ↑S → ∀ (d₁ : (Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂ → WO (c.X Q.s) (c.X (Q.s + 1)) →₀ ℂ) (d₂ : (Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂ → WO (c.X (Q.s + 1)) (c.X (Q.s + 1 + 1)) →₀ ℂ), (∀ yb, (ofSplit Q.q).IsCombAt c.ψ (Q.FX Q.s (Nat.le_succ _)) (Q.FX (Q.s + 1) le_rfl) (d₁ yb) (Q.T.xiPen (box _ Q.c₂) (Q.rep Q.s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastStepsKernelEqSumCouplingCosetsStatement

theorem **_root_.GlobalTheory.T.lastStepsKernel_eq_sum_couplingCosets** : ∀ (D : D7) (c : Comm D) (v : VCom c) χ (Q : QAt v χ), c.Is → v.Is → ∀ (h : Q.s + 2 * 0 + 2 = D.R.m), c.Lz.rank = 1 → (v.τ Q.s h)ᴴ * ((c.bes Q.s 0 h).L.gram * v.τ Q.s h) = (-(D.host.xi0 * star D.host.xi0)) • c.Lz.gram → ∀ s₁ : StepBF c.ψ (c.B Q.s 0 h) (c.J Q.s 0 h), s₁.χW = χ Q.s → ∀ Ψ₁ Ψ₂, GlobalTheory.FinPl.Vac c.ψ _ _ (v.N Q.s) (v.N (Q.s + 1)) (· = Q.q.v) Ψ₁ → GlobalTheory.FinPl.Vac c.ψ _ _ (v.N (Q.s + 1)) v.NT (· = Q.q.v) Ψ₂ → ∀ α (S : Finset ((Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂)), Q.T.coupIdx Q.A (box _ Q.c₁) (box _ Q.c₂) ⊆ ↑S → ∀ (d₁ : (Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂ → WO (c.X Q.s) (c.X (Q.s + 1)) →₀ ℂ) (d₂ : (Q.X (Q.s + 1) → Q.k) ⧸ box (Q.X (Q.s + 1)) Q.c₂ → WO (c.X (Q.s + 1)) (c.X (Q.s + 1 + 1)) →₀ ℂ), (∀ yb, (ofSplit Q.q).IsCombAt c.ψ (Q.FX Q.s (Nat.le_succ _)) (Q.FX (Q.s + 1) le_r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastTwoStepsDoubleLiftPeriodEqIntegral

theorem **sympl_zero_l** (S T : HSp D) (x : WO S T) : symplOH S T 0 x = 0

theorem **sympl_zero_r** (S T : HSp D) (x : WO S T) : symplOH S T x 0 = 0

theorem **biTheta_of_maps** {ψ : AddChar (AF D.host.F) ℂ} {V₀ A S X T : HSp D} (a : WO A S →+ WO V₀ S) (b : WO S X →+ WO V₀ S) (p : WO S X →+ WO S T) (ca : Continuous a) (cb : Continuous b) (cp : Continuous p) (ra : ∀ γ, ∃ γ', a (ratW S A γ) = ratW S V₀ γ') (rb : ∀ γ, ∃ γ', b (ratW X S γ) = ratW S V₀ γ') (rp : ∀ γ, ∃ γ', p (ratW X S γ) = ratW T S γ') (fa : ∀ x : WO A S, x.map (finPart D) = 0 → (a x).map (finPart D) = 0) (fb : ∀ x : WO S X, x.map (finPart D) = 0 → (b x).map (finPart D) = 0) (fp : ∀ x : WO S X, x.map (finPart D) = 0 → (p x).map (finPart D) = 0) (ia : ∀ x : WO A S, x.map (infPart D) = 0 → (a x).map (infPart D) = 0) (ib : ∀ x : WO S X, x.map (infPart D) = 0 → (b x).map (infPart D) = 0) (ip : ∀ x : WO S X, x.map (infPart D) = 0 → (p x).map (infPart D) = 0) (hs : ∀ u v u' v', symplOH S T (p v) (p v') + symplOH V₀ S (a u + b v) (a u' + b v') = symplOH A S u u' + symplOH S X v v') {

theorem **lastBi_biTheta** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (V₀ A L S X T L'' : HSp D) (o : OrthDec V₀ A L) (o'' : OrthDec X T L'') (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), τᴴ · (L.gram · τ) = (-(D.host.xi0 * star D.host.xi0)) • L''.gram → ∀ (Ξ : WO V₀ S → ℂ) (Φ₂ : WO S T → ℂ), IsThetaFunO ψ V₀ S Ξ → IsThetaFunO ψ S T Φ₂ → IsBiTheta ψ A S X (lastBi o o'' τ Ξ Φ₂)

theorem **integral_automorphic_rankZeroWittEmbedding** : ∀ (D : D7) (S S₀ : HSp D) (d : HypDec S S₀ 0) (v : Vol S) (v₀ : Vol S₀), v.μ = Measure.map d.emb0 v₀.μ → ∀ f : GA D S → ℂ, (∀ γ ∈ GQ S, ∀ g, f (γ * g) = f g) → ∫ g in v.ℱ, f g ∂v.μ = ∫ g in v₀.ℱ, f (d.emb0 g) ∂v₀.μ

theorem **flatBF_rankZero** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (B : BesLev D 0) (J : JacLev D 0) (s : StepBF ψ B J) (Ψ : WO B.V J.W → ℂ), flatBF s Ψ = fun w => Ψ (B.d.ext0 J.W (J.d.inc0 B.V₀ w))

theorem **ker_inv** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {γ : GA D S} (hγ : γ ∈ GQ S) {γ' : GA D T} (hγ' : γ' ∈ GQ T) (g : GA D S) (h : GA D T) : om (γ * g, γ' * h) f 0 = om (g, h) f 0

theorem **ker_inv_right** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {γ' : GA D T} (hγ' : γ' ∈ GQ T) (g : GA D S) (h : GA D T) : om (g, γ' * h) f 0 = om (g, h) f 0

theorem **ker_inv_left** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {γ : GA D S} (hγ : γ ∈ GQ S) (g : GA D S) (h : GA D T) : om (γ * g, h) f 0 = om (g, h) f 0

theorem **liftS_contAut** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hT : CoCpt T) (v : Vol T) {G : GA D T → ℂ} (hG : IsContAut T G) : IsContAut S (liftS om v.μ v.ℱ G f)

theorem **liftS_trans** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (v : Vol T) {G : GA D T → ℂ} (hG : ∀ γ ∈ GQ T, ∀ h, G (γ * h) = G h) (g : GA D S) (b : GA D T) : liftS om v.μ v.ℱ (rTrans b G) (om (g, b) f) = rTrans g (liftS om v.μ v.ℱ G f)

theorem **contAut_rTrans** {V : HSp D} {F : GA D V → ℂ} (hF : IsContAut V F) (b : GA D V) : IsContAut V (rTrans b F)

theorem **cocpt_zero** {S S₀ : HSp D} (d : HypDec S S₀ 0) (h : CoCpt S₀) : CoCpt S

theorem **val_zero** {ψ : AddChar (AF D.host.F) ℂ} {B : BesLev D 0} {J : JacLev D 0} (s : StepBF ψ B J) {Ψ : WO B.V J.W → ℂ} (hΨ : IsThetaFunO ψ B.V J.W Ψ) : IsThetaFunO ψ B.V₀ J.W₀ (flatBF s Ψ) ∧ ∀ g h, s.om (B.d.emb0 g, J.d.emb0 h) Ψ 0 = s.om₀ (g, h) (flatBF s Ψ) 0

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c), c.Is → c.VolLast → t.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastTwoStepsDoubleLiftPeriodEqIntegralStatement

theorem **_root_.GlobalTheory.T.lastTwoSteps_doubleLiftPeriod_eq_integral** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c), c.Is → c.VolLast → t.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastWittLineOfBesselJacobiLatticesGood

theorem **mem_idAt_of_idV** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {v : GlobalTheory.FinPl.Pl D} {a : FI D} {s : AE D} (hs : s ∈ idV a v) : s ∈ idAt D (· = v)

theorem **sameAt_div** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {v : GlobalTheory.FinPl.Pl D} {a a' b b' : FI D} (ha : a ≠ 0) (ha' : a' ≠ 0) (hb : b ≠ 0) (hb' : b' ≠ 0) (ha : SameAt v a a') (hb : SameAt v b b') : SameAt v (b * a⁻¹) (b' * a'⁻¹)

theorem **sameAt_trans** {v : GlobalTheory.FinPl.Pl D} {a b c : FI D} (h1 : SameAt v a b) (h2 : SameAt v b c) : SameAt v a c

theorem **sameAt_refl** (v : GlobalTheory.FinPl.Pl D) (a : FI D) : SameAt v a a

theorem **sameAt_mul_of_one** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {v : GlobalTheory.FinPl.Pl D} {c n : FI D} (hc : c ≠ 0) (hn : n ≠ 0) (h : SameAt v 1 n) : SameAt v c (c * n)

theorem **colMul_jX** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (i : Fin ρ) (a : AE D) : colMul d.jX i a = a • ratV (col d i)

theorem **col_ne_zero** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (i : Fin ρ) : col d i ≠ 0

theorem **vecB_ne_zero** {S₀ A L : HSp D} (o : OrthDec S₀ A L) {ζ : Fin L.rank → D.host.E} (hζ : IsLineVec L ζ) : vecB o ζ ≠ 0

theorem **smul_colMul'** {m n : ℕ} (x : Mx D m n) (i : Fin n) (s t : AE D) : s • colMul x i t = colMul x i (s * t)

theorem **cols_sameAt** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} {LV : Lat B.V} {LV₀ : Lat B.V₀} {LW : Lat J.W} {LW₀ : Lat J.W₀} {v : GlobalTheory.FinPl.Pl D} (hG : GoodBF ψ B J LV LV₀ LW LW₀ v) (i : Fin ρ) {e b : FI D} (hb0 : b ≠ 0) (he : ∀ a ∈ idAt D (· = v), a • ratV (col B.d i) ∈ latV B.V LV (· = v) ↔ a ∈ idV e v) (hb : ∀ a ∈ idAt D (· = v), a • ratV (col J.d i) ∈ latV J.W LW (· = v) ↔ a ∈ idV b v) : SameAt v e (b * (D.host.AA : FI D))

theorem **last_sameAt** {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {ρ : ℕ} {B : BesLev D ρ} {J : JacLev D ρ} {LV : Lat B.V} {LV₀ : Lat B.V₀} {LW : Lat J.W} {LW₀ : Lat J.W₀} {v : GlobalTheory.FinPl.Pl D} (hG : GoodBF ψ B J LV LV₀ LW LW₀ v) (i : Fin ρ) (hi : i.1 + 1 = ρ) {ξ : FI D} {b : ℕ → FI D} (hξ0 : ξ ≠ 0) (hξ : GlobalTheory.SeeSaw.LineId B.o B.ζ LV₀ ξ v) (hb0 : b i ≠ 0) (hmono : GlobalTheory.SeeSaw.MonoId J.d LW LW₀ b v) : SameAt v 1 (FractionalIdeal.spanSingleton _ D.host.xi0 * ((D.host.AAbar : FI D) * (b i)⁻¹) * ξ)

theorem **ideal_eq** (b ξ : FI D) (hb0 : b ≠ 0) (hξ0 : ξ ≠ 0) : b * (D.host.AA : FI D) * ξ⁻¹ * (FractionalIdeal.spanSingleton _ D.host.xi0 * ((D.host.AAbar : FI D) * b⁻¹) * ξ) = GlobalTheory.SeeSaw.dI D

theorem **GlobalTheory.Core.Mi.LatMatch.lastAt_of_goodBF** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : GlobalTheory.FinPl.Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → LastAt ψ B.d (vecB B.o B.ζ) LV LV₀ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LastWittLineOfBesselJacobiLatticesGoodStatement

theorem **_root_.GlobalTheory.T.lastWittLine_of_besselJacobiLatticesGood** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (LV : Lat B.V) (LV₀ : Lat B.V₀) (LW : Lat J.W) (LW₀ : Lat J.W₀) (v : GlobalTheory.FinPl.Pl D), GoodBF ψ B J LV LV₀ LW LW₀ v → LastAt ψ B.d (vecB B.o B.ζ) LV LV₀ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatEqOfCarries

theorem **lat_eq_of_carries** {g : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} {N₁ N₂ : Submodule (𝒪 E) (Fin n → E)} (h₁ : Submodule.IsLattice E N₁) (h₂ : Submodule.IsLattice E N₂) (c₁ : Formalism.L2.Herm.Carries g (Formalism.L2.Col.stdLat E n) N₁) (c₂ : Formalism.L2.Herm.Carries g (Formalism.L2.Col.stdLat E n) N₂) : N₁ = N₂

theorem **frsF_symm_mem** {u : Pr.Word} {j : ℕ} (p : ChPt (R

theorem **frsF_mem** {u : Pr.Word} {j : ℕ} (p : ChPt (R

theorem **cosetIn_eq_of** {u : Pr.Word} {j : ℕ} (r : PtAt R sel mu cst u j) (P : rowSG Pr sel mu) (hP : P.1 ∈ Pr.vGs) (y : Sp X.E (R.N j)) (e : rowSG Pr sel mu → ℤ) (he : e P = 0) (hY : r.1.2 = {z | z - y ∈ locLat (rowSG Pr sel mu) r.1.1.deco.1.N e}) : cosetIn r P = {z | z - (frsF sel mu r.1.1 P).symm y ∈ stdAt X.E (R.N j) P.1}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatIndLastKerRow

theorem **restrModel_apply** {m n m' n' m'' n'' : ℕ} (r : Mx D m n ≃+ Mx D m' n' × Mx D m'' n'') (h : Mx D m'' n'' → ℂ) (T' : (Mx D m n → ℂ) →ₗ[ℂ] (Mx D m n → ℂ)) (y : Mx D m'' n'') (Φ : Mx D m' n' → ℂ) (w : Mx D m' n') : restrModel r h T' y Φ w = T' (fun t => Φ (r t).1 * h (r t).2) (r.symm (w, y))

theorem **heisO_resS_symm** (ψ : AddChar (AF D.host.F) ℂ) {S A L T : HSp D} (o : OrthDec S A L) (a : WO A T) (F : WO S T → ℂ) (w : WO A T) (y : WO L T) : heisO ψ S T ((o.resS T).symm (a, 0)) F ((o.resS T).symm (w, y)) = heisO ψ A T a (fun w' => F ((o.resS T).symm (w', y))) w

theorem **heisO_resT_symm** (ψ : AddChar (AF D.host.F) ℂ) {S A L T : HSp D} (o : OrthDec S A L) (a : WO T A) (F : WO T S → ℂ) (w : WO T A) (y : WO T L) : heisO ψ T S ((o.resT T).symm (a, 0)) F ((o.resT T).symm (w, y)) = heisO ψ T A a (fun w' => F ((o.resT T).symm (w', y))) w

theorem **symplO_resS_symm** {S A L T : HSp D} (o : OrthDec S A L) (t : WO S T) (a : WO A T) : symplO S T t ((o.resS T).symm (a, 0)) = symplO A T ((o.resS T) t).1 a

theorem **symplO_resT_symm** {S A L T : HSp D} (o : OrthDec S A L) (t : WO T S) (a : WO T A) : symplO T S t ((o.resT T).symm (a, 0)) = symplO T A ((o.resT T) t).1 a

theorem **half_resS_symm** {S A L T : HSp D} (o : OrthDec S A L) {e : AE D} {a : WO A T} (ha : IsHalf e a) : IsHalf e ((o.resS T).symm (a, 0))

theorem **half_resT_symm** {S A L T : HSp D} (o : OrthDec S A L) {e : AE D} {a : WO T A} (ha : IsHalf e a) : IsHalf e ((o.resT T).symm (a, 0))

theorem **latInd_lastKer** (ΛA : Submodule 𝒪[k] (n' → k)) (ΛT : Submodule 𝒪[k] (n → k)) : latInd (lastKer ΛA) ΛT = tensCol (latInd ΛA ΛT) (vecInd ΛT)

theorem **latInd_lastKer_row** (ΛA : Submodule 𝒪[k] (n' → k)) (ΛT : Submodule 𝒪[k] (n → k)) : latInd ΛT (lastKer ΛA) = tensRow (latInd ΛT ΛA) (dualInd ΛT)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeChainStandardPair

theorem **aaOf_top** (D : D7) : (Formalism.L2.PairingIndexBounds.F.aaOf D ⊤) = (D.host.AA0 : Submodule (𝒪 D.host.E) D.host.E)

theorem **thetaSpace_std_eq_line** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z) (i : ℕ) (hi : i + 1 ≤ B.R.m), ∀ F ∈ Formalism.L2.Th1.H.Th (HV I.A.sc) (alV (I.rec7 l Z).host.d I.A.sc) (LamC (tauE (I.rec7 l Z)) I.A.δ ((I.rec7 l Z).host.AA0 : Submodule (𝒪 (I.rec7 l Z).host.E) (I.rec7 l Z).host.E) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N i)) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N (i + 1)))), ∃ c : ℂ, F = c • I.J.V.thMoved (B.R.N i) (B.R.N (i + 1)) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N i)) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N (i + 1)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeChainStandardPairStatement

theorem **_root_.GlobalTheory.T.lattice_chain_standard_pair** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (w : World I l Z) (i : ℕ) (hi : i + 1 ≤ B.R.m), ∀ F ∈ Formalism.L2.Th1.H.Th (HV I.A.sc) (alV (I.rec7 l Z).host.d I.A.sc) (LamC (tauE (I.rec7 l Z)) I.A.δ ((I.rec7 l Z).host.AA0 : Submodule (𝒪 (I.rec7 l Z).host.E) (I.rec7 l Z).host.E) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N i)) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N (i + 1)))), ∃ c : ℂ, F = c • I.J.V.thMoved (B.R.N i) (B.R.N (i + 1)) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N i)) (Formalism.L2.Col.stdLat (I.rec7 l Z).host.E (B.R.N (i + 1)))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionAAScaledEqAtUnitSplitPlaces

theorem **pairA_sum** {κ : Type} (s : Finset κ) (f : κ → Af D.host.F) : pairA D.host.F 0 (∑ i ∈ s, f i) = ∑ i ∈ s, pairA D.host.F 0 (f i)

theorem **latMul_mono** (S : HSp D) (α : Set D.host.E) (L : Lat S) {P P' : Pl D → Prop} (h : ∀ w, P w → P' w) : latMul S α L P ≤ latMul S α L P'

theorem **latMul_le_of_parts** (S : HSp D) (α : Set D.host.E) (L : Lat S) (M : AddSubgroup (Fin S.rank → AE D)) {P : Pl D → Prop} (s : Finset (Pl D)) (hs : ∀ w, P w → w ∈ s) (h : ∀ w ∈ s, latMul S α L (· = w) ≤ M) : latMul S α L P ≤ M

theorem **latA_eq_latV** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (S : HSp D) (L : Lat S), latA S L P = latV S L P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionAAScaledEqAtUnitSplitPlacesStatement

theorem **_root_.GlobalTheory.T.latticeCompletionAAScaled_eq_at_unitSplitPlaces** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (S : HSp D) (L : Lat S), latA S L P = latV S L P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionAtPlacesMemCompletedHermitianLa

theorem **gen_mem** (D : D7) (S : HSp D) (L : Lat S) (α : Af D.host.F) (hα : α ∈ intAf D.host.F) (x : Fin S.rank → D.host.E) (hx : x ∈ L) : (fun i => Formalism.IndefR.Lev.swapAE D (finPart D ((pairA D.host.F 0 α) ⊗ₜ[D.host.F] ((1 : D.host.E) ∗ x i)))) ∈ Formalism.L2.Herm.hatL (F

theorem **latV_mem_hatL** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop) (y : Fin S.rank → AE D), y ∈ latV S L P → (fun i => Formalism.IndefR.Lev.swapAE D (finPart D (y i))) ∈ Formalism.L2.Herm.hatL (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionAtPlacesMemCompletedHermitianLaStatement

theorem **_root_.GlobalTheory.T.latticeCompletionAtPlaces_mem_completedHermitianLattice** : ∀ (D : D7) (S : HSp D) (L : Lat S) (P : Pl D → Prop) (y : Fin S.rank → AE D), y ∈ latV S L P → (fun i => Formalism.IndefR.Lev.swapAE D (finPart D (y i))) ∈ Formalism.L2.Herm.hatL (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionAtPlacesSelfDualOfSelfDual

theorem **cA_tmul** (a : AF D.host.F) (e : D.host.E) : cA D (a ⊗ₜ[D.host.F] e) = a ⊗ₜ[D.host.F] star e

theorem **incA_eq** (e : D.host.E) : incA D e = (1 : AF D.host.F) ⊗ₜ[D.host.F] e

theorem **intAt_zero** (P : Pl D → Prop) : (0 : AE D) ∈ intAt D P

theorem **intAt_add** {P : Pl D → Prop} {a b : AE D} (ha : a ∈ intAt D P) (hb : b ∈ intAt D P) : a + b ∈ intAt D P

theorem **intAt_neg** {P : Pl D → Prop} {a : AE D} (ha : a ∈ intAt D P) : -a ∈ intAt D P

theorem **hA_zero_left** (S : HSp D) (s : Fin S.rank → AE D) : hA S 0 s = 0

theorem **hA_add_left** (S : HSp D) (y y' s : Fin S.rank → AE D) : hA S (y + y') s = hA S y s + hA S y' s

theorem **hA_neg_left** (S : HSp D) (y s : Fin S.rank → AE D) : hA S (-y) s = -hA S y s

theorem **hA_zero_right** (S : HSp D) (y : Fin S.rank → AE D) : hA S y 0 = 0

theorem **hA_add_right** (S : HSp D) (y s s' : Fin S.rank → AE D) : hA S y (s + s') = hA S y s + hA S y s'

theorem **hA_neg_right** (S : HSp D) (y s : Fin S.rank → AE D) : hA S y (-s) = -hA S y s

theorem **hA_gen** (S : HSp D) (α β : Af D.host.F) (x x' : Fin S.rank → D.host.E) : hA S (fun j => pairA D.host.F 0 α ⊗ₜ[D.host.F] (1 ∗ x j)) (fun i => pairA D.host.F 0 β ⊗ₜ[D.host.F] (1 ∗ x' i)) = pairA D.host.F 0 (β * α) ⊗ₜ[D.host.F] (∑ i, ∑ j, star (x' i) ∗ HSpace.gram S i j ∗ x j)

theorem **hA_mem** {S : HSp D} {LS : Lat S} (hS : IsSelfDualC S LS) {P : Pl D → Prop} {y s : Fin S.rank → AE D} (hy : y ∈ latV S LS P) (hs : s ∈ latV S LS P) : hA S y s ∈ intAt D P

theorem **wachExpansion_smul_compatible** {S : HSp D} {LS : Lat S} (P : Pl D → Prop) {s : Fin S.rank → D.host.E} (hs : s ∈ LS) {z : Af D.host.F ⊗[D.host.F] D.host.E} (hz : z ∈ intAE D) : (idem D P ∗ finIn D z) • (fun l => incA D (s l)) ∈ latV S LS P

theorem **smul_mem** {S : HSp D} {LS : Lat S} {P : Pl D → Prop} {c : AE D} (hc : c ∈ intAt D P) {s : Fin S.rank → D.host.E} (hs : s ∈ LS) : c • (fun l => incA D (s l)) ∈ latV S LS P

theorem **cA_test** (P : Pl D → Prop) (e : D.host.E) : cA D (pairA D.host.F 0 (ePl D P) ⊗ₜ[D.host.F] (1 ∗ e)) = idem D P ∗ incA D (star e)

theorem **selfDual_latV** : ∀ (D : D7) (S : HSp D) (LS : Lat S), IsSelfDualC S LS → ∀ (P : Pl D → Prop) (y : Fin S.rank → AE D), (∀ r, y r ∈ idAt D P) → (y ∈ latV S LS P ↔ ∀ s ∈ latV S LS P, (∑ i, ∑ j, cA D (s i) ∗ incA D (HSpace.gram S i j) ∗ y j) ∈ intAt D P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionAtPlacesSelfDualOfSelfDualStatement

theorem **_root_.GlobalTheory.T.latticeCompletionAtPlaces_selfDual_of_selfDual** : ∀ (D : D7) (S : HSp D) (LS : Lat S), IsSelfDualC S LS → ∀ (P : Pl D → Prop) (y : Fin S.rank → AE D), (∀ r, y r ∈ idAt D P) → (y ∈ latV S LS P ↔ ∀ s ∈ latV S LS P, (∑ i, ∑ j, cA D (s i) ∗ incA D (HSpace.gram S i j) ∗ y j) ∈ intAt D P)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionEqOfLocalAgreement

theorem **latV_congr** (S : HSp D) (L : Lat S) {P P' : Pl D → Prop} (h : ∀ w, P w ↔ P' w) : latV S L P = latV S L P'

theorem **latV_le_of_int_smul_on** (S : HSp D) (L M : Lat S) (d : ℤ) (hd : d ≠ 0) (hLM : ∀ x ∈ L, d • x ∈ M) (R : Pl D → Prop) (hR : ∀ v, R v → ((d : 𝒪 D.host.F)) ∉ v.asIdeal) : latV S L R ≤ latV S M R

theorem **latV_le_finite** (S : HSp D) (L N : Lat S) (d : ℤ) (hd : d ≠ 0) (hLN : ∀ x ∈ L, d • x ∈ N) (P : Pl D → Prop) (hP : ∀ w, P w → latV S L (· = w) = latV S N (· = w)) (T : Finset (Pl D)) : latV S L (fun p => P p ∧ p ∈ T) ≤ latV S N (fun p => P p ∧ p ∈ T)

theorem **latV_le_of_local** (S : HSp D) (L N : Lat S) (d : ℤ) (hd : d ≠ 0) (hLN : ∀ x ∈ L, d • x ∈ N) (P : Pl D → Prop) (hP : ∀ w, P w → latV S L (· = w) = latV S N (· = w)) : latV S L P ≤ latV S N P

theorem **latV_eq_of_local** : ∀ {D : D7} (S : HSp D) (L N : GlobalTheory.FinPl.Lat S) (P : GlobalTheory.FinPl.Pl D → Prop), Submodule.IsLattice D.host.E L → Submodule.IsLattice D.host.E N → (∀ w, P w → GlobalTheory.FinPl.latV S L (· = w) = GlobalTheory.FinPl.latV S N (· = w)) → GlobalTheory.FinPl.latV S L P = GlobalTheory.FinPl.latV S N P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeCompletionEqOfLocalAgreementStatement

theorem **_root_.GlobalTheory.T.latticeCompletion_eq_of_local_agreement** : ∀ {D : D7} (S : HSp D) (L N : GlobalTheory.FinPl.Lat S) (P : GlobalTheory.FinPl.Pl D → Prop), Submodule.IsLattice D.host.E L → Submodule.IsLattice D.host.E N → (∀ w, P w → GlobalTheory.FinPl.latV S L (· = w) = GlobalTheory.FinPl.latV S N (· = w)) → GlobalTheory.FinPl.latV S L P = GlobalTheory.FinPl.latV S N P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeEqBaseOffTwist

theorem **lattice_eq_base_off_twist** : ∀ {B : GlobalTheory.Mem.Base} {X : GlobalTheory.Mem.PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → a.True → ∀ row col, (a.fam row col).c.Is → ∀ p, p ∉ a.v.tP → GlobalTheory.FinPl.latV ((I.rec7 l Z).sp.X 0) (a.g.x 0).L (· = p) = GlobalTheory.FinPl.latV ((I.rec7 l Z).sp.X 0) ((F0 W u).cls0.lev.Lb.map ((LinearEquiv.funCongrLeft (I.rec7 l Z).host.E (I.rec7 l Z).host.E (finCongr (Formalism.L2.Forms.rankN (I.rec7 l Z) 0 (Nat.zero_le _)))).toLinearMap.restrictScalars (𝒪 (I.rec7 l Z).host.E))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeLineRatioTransportOfColumnsCarryLines

theorem **incA_mul_tmul** (b : D.host.E) (p : AF D.host.F) (z : D.host.E) : incA D b * (p ⊗ₜ[D.host.F] z) = p ⊗ₜ[D.host.F] (b * z)

theorem **colMul_add** {m n : ℕ} (x : Mx D m n) (l : Fin n) (c c' : AE D) : colMul x l (c + c') = colMul x l c + colMul x l c'

theorem **colMul_mul** {m n : ℕ} (x : Mx D m n) (l : Fin n) (b c : AE D) : (fun k => b * colMul x l c k) = colMul x l (b * c)

theorem **mul_mem_latMul** {S : HSp D} {𝔞 𝔟 : Set D.host.E} {L L' : Lat S} {P : Pl D → Prop} (b : D.host.E) (h : ∀ a ∈ 𝔞, ∀ x ∈ L, ∃ a' ∈ 𝔟, ∃ x' ∈ L', ∀ i, b * (a * x i) = a' * x' i) {y : Fin S.rank → AE D} (hy : y ∈ latMul S 𝔞 L P) : (fun i => incA D b * y i) ∈ latMul S 𝔟 L' P

theorem **latV_to_latA** {S : HSp D} {L : Lat S} {P : Pl D → Prop} {α : D.host.E} (hα : α ∈ D.host.AA) {y : Fin S.rank → AE D} (hy : y ∈ latV S L P) : (fun i => incA D α * y i) ∈ latA S L P

theorem **latA_to_latV** {S : HSp D} {L : Lat S} {P : Pl D → Prop} (hAA : D.host.AA ≠ 0) {β : D.host.E} (hβ : β ∈ D.host.AA⁻¹) {y : Fin S.rank → AE D} (hy : y ∈ latA S L P) : (fun i => incA D β * y i) ∈ latV S L P

theorem **latA_eq_zero** (hAA : D.host.AA = 0) {S : HSp D} {L : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latA S L P) : y = 0

theorem **idAt_eq_zero_of_AA** (hAA : D.host.AA = 0) {V V₀ W W₀ : HSp D} {ρ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (LV : Lat V) (LW : Lat W) (v : Pl D) (hsdV : IsSelfDualC V LV) (hC : ColsAt dV dW.jX LV LW v) (i : Fin ρ) (a : AE D) (ha : a ∈ idAt D (· = v)) : a = 0

theorem **GlobalTheory.FinPl.ratio_of_colsAt** : ∀ (D : D7) (V V₀ W W₀ : HSp D) (ρ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (LV : Lat V) (LV₀ : Lat V₀) (LW : Lat W) (LW₀ : Lat W₀) (v : Pl D), Tame D v → IsSelfDualC V LV → IsSelfDualC W LW → MonoAt dV LV LV₀ v → MonoAt dW LW LW₀ v → ColsAt dV dW.jX LV LW v → ∀ (i j : Fin ρ) (r : AE D), r ∈ idAt D (· = v) → (∀ a : AE D, colMul dV.jX j a ∈ latV V LV (· = v) → colMul dV.jX i (r * a) ∈ latV V LV (· = v)) → ∀ a ∈ idAt D (· = v), colMul dW.jX j a ∈ latV W LW (· = v) → colMul dW.jX i (r * a) ∈ latV W LW (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeLineRatioTransportOfColumnsCarryLinesStatement

theorem **_root_.GlobalTheory.T.latticeLineRatio_transport_of_columnsCarryLines** : ∀ (D : D7) (V V₀ W W₀ : HSp D) (ρ : ℕ) (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ ρ) (LV : Lat V) (LV₀ : Lat V₀) (LW : Lat W) (LW₀ : Lat W₀) (v : Pl D), Tame D v → IsSelfDualC V LV → IsSelfDualC W LW → MonoAt dV LV LV₀ v → MonoAt dW LW LW₀ v → ColsAt dV dW.jX LV LW v → ∀ (i j : Fin ρ) (r : AE D), r ∈ idAt D (· = v) → (∀ a : AE D, colMul dV.jX j a ∈ latV V LV (· = v) → colMul dV.jX i (r * a) ∈ latV V LV (· = v)) → ∀ a ∈ idAt D (· = v), colMul dW.jX j a ∈ latV W LW (· = v) → colMul dW.jX i (r * a) ∈ latV W LW (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeLocalEqIntegralAlmostAll

theorem **exists_zsmul_mem_of_span** {L : Submodule (𝒪 E) V} {x : V} (hx : x ∈ Submodule.span E (L : Set V)) : ∃ N : ℤ, N ≠ 0 ∧ N • x ∈ L

theorem **exists_zsmul_mem_of_fg** {L M : Submodule (𝒪 E) V} (hL : ∀ x, ∃ N : ℤ, N ≠ 0 ∧ N • x ∈ L) (hM : M.FG) : ∃ N : ℤ, N ≠ 0 ∧ ∀ x ∈ M, N • x ∈ L

theorem **exists_commensurable** {n : ℕ} (L : Submodule (𝒪 E) (Fin n → E)) (hL : Submodule.IsLattice E L) : ∃ N : ℤ, N ≠ 0 ∧ (∀ x ∈ L, ∀ i, ∃ o : 𝒪 E, (o : E) = N • x i) ∧ ∀ (i : Fin n) (o : 𝒪 E), (Pi.single i (N • (o : E)) : Fin n → E) ∈ L

theorem **repr_map** (f : A →ₐ[F] B) (a : A ⊗[F] E) (k : ι) : (Algebra.TensorProduct.basis B b).repr (Algebra.TensorProduct.map f (AlgHom.id F E) a) k = f ((Algebra.TensorProduct.basis A b).repr a k)

theorem **evalV_single_same** (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v) : evalV F v (singleV F v y) = y

theorem **evalV_single_ne** (v : HeightOneSpectrum (𝒪 F)) (y : Fv F v) {w : HeightOneSpectrum (𝒪 F)} (hw : w ≠ v) : evalV F w (singleV F v y) = 0

theorem **af_ext** {a b : Af F} (h : ∀ w, evalV F w a = evalV F w b) : a = b

theorem **intSupp_single_mul** (v : HeightOneSpectrum (𝒪 F)) {α : Af F} (hα : α ∈ intAf F) : IntSupp v (singleV F v 1 * α)

theorem **IntSupp.inv_smul** {v : HeightOneSpectrum (𝒪 F)} {α : Af F} (hα : IntSupp v α) {N : ℤ} (hN : (N : 𝒪 F) ∉ v.asIdeal) : IntSupp v ((N : F)⁻¹ • α)

theorem **single_mul_eq** (v : HeightOneSpectrum (𝒪 F)) (c : AF F) (h1 : fstA F c = 0) (h2 : ∀ w, ¬ w = v → evalV F w (sndA F c) = 0) : pairA F 0 (singleV F v 1) ∗ c = c

theorem **pairA_smul** (c : F) (α : Af F) : pairA F 0 (c • α) = c • pairA F 0 α

theorem **tmul_eq** {E : Type} [Field E] [Algebra F E] (α : Af F) (x : E) {N : ℤ} (hN : N ≠ 0) : pairA F 0 α ⊗ₜ[F] x = pairA F 0 ((N : F)⁻¹ • α) ⊗ₜ[F] (N • x)

theorem **eventually_notMem** {N : ℤ} (hN : N ≠ 0) : ∀ᶠ v : HeightOneSpectrum (𝒪 F) in cofinite, (N : 𝒪 F) ∉ v.asIdeal

theorem **mem_idAt** {P : Pl D → Prop} {a : AE D} : a ∈ idAt D P ↔ infPart D a = 0 ∧ ∀ w, ¬ P w → compAt D w a = 0

theorem **smul_eq_of_mem_idAt** {v : Pl D} {a : AE D} (ha : a ∈ idAt D (· = v)) : pairA D.host.F 0 (singleV D.host.F v 1) • a = a

theorem **tmul_mem_intAt** {v : Pl D} {α : Af D.host.F} (hα : IntSupp v α) (o : 𝒪 D.host.E) : (pairA D.host.F 0 α ⊗ₜ[D.host.F] (o : D.host.E) : AE D) ∈ intAt D (· = v)

theorem **plc_tmul** (v : Pl D) (α : Af D.host.F) (x : D.host.E) : plc v (α ⊗ₜ[D.host.F] x) = pairA D.host.F 0 (singleV D.host.F v 1 ∗ α) ⊗ₜ[D.host.F] x

theorem **plc_finPart** (v : Pl D) (a : AE D) : plc v (finPart D a) = pairA D.host.F 0 (singleV D.host.F v 1) • a

theorem **eventually_latV_iff** : ∀ (D : D7) (S : HSp D) (L : Lat S), Submodule.IsLattice D.host.E L → ∀ᶠ v : Pl D in cofinite, ∀ y : Fin S.rank → AE D, y ∈ latV S L (· = v) ↔ ∀ i, y i ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeLocalEqIntegralAlmostAllStatement

theorem **_root_.GlobalTheory.T.lattice_local_eq_integral_almost_all** : ∀ (D : D7) (S : HSp D) (L : Lat S), Submodule.IsLattice D.host.E L → ∀ᶠ v : Pl D in cofinite, ∀ y : Fin S.rank → AE D, y ∈ latV S L (· = v) ↔ ∀ i, y i ∈ intAt D (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeOrthogonalSumLineAtSplitPlace

theorem **mulVec_sec** {m n : ℕ} (M : Matrix (Fin m) (Fin n) (AE D)) (u : Fin n → k) : M *ᵥ (fun i => p.sec (u i)) = fun i => p.sec ((M.map p.pr *ᵥ u) i)

theorem **mulVec_csec** {m n : ℕ} (M : Matrix (Fin m) (Fin n) (AE D)) (z : Fin n → k) : M *ᵥ (fun i => cA D (p.sec (z i))) = fun i => cA D (p.sec ((M.map (fun a => p.pr (cA D a)) *ᵥ z) i))

theorem **eq_sec_add_csec** {n : ℕ} {y : Fin n → AE D} (hy : ∀ i, y i ∈ idAt D (· = p.v)) : y = (fun i => p.sec (p.pr (y i))) + (fun i => cA D (p.sec (p.pr (cA D (y i)))))

theorem **mulVec_mem_of_parts** {S T : HSp D} {L : Lat S} {L' : Lat T} (hL : IsSelfDualC S L) (M : Matrix (Fin T.rank) (Fin S.rank) (AE D)) (hw : ∀ u, (fun i => p.sec (u i)) ∈ latV S L (· = p.v) → (fun i => p.sec ((M.map p.pr *ᵥ u) i)) ∈ latV T L' (· = p.v)) (hc : ∀ z, (fun i => cA D (p.sec (z i))) ∈ latV S L (· = p.v) → (fun i => cA D (p.sec ((M.map (fun a => p.pr (cA D a)) *ᵥ z) i))) ∈ latV T L' (· = p.v)) {y : Fin S.rank → AE D} (hy : y ∈ latV S L (· = p.v)) : M *ᵥ y ∈ latV T L' (· = p.v)

theorem **pair_eq** (S : HSp D) (x z : Fin S.rank → k) : ∑ i, ∑ j, x i ∗ p.pr (cA D (incA D (S.gram i j))) ∗ z j = x ⬝ᵥ (S.gram.map (p.pr.comp ((cA D).comp (incA D))) *ᵥ z)

theorem **conjT_map** {m n : Type} (M : Matrix m n D.host.E) : Mᴴ.map (p.pr.comp ((cA D).comp (incA D))) = (M.map (p.pr.comp (incA D)))ᵀ

theorem **orthAt_of_frames** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n' : Type) [Fintype n'] [DecidableEq n'] (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)), IsSumFrame o ζ α α₀ → p.UnitAA → ∀ (LA : Lat A) (LS : Lat S) (ΛA : Submodule 𝒪[k] (n' → k)), IsSelfDualC A LA → IsSelfDualC S LS → α.IsLat LA ΛA → α₀.IsLat LS (lastKer ΛA) → ∃ LL : Lat L, OrthAt o LS LA LL p.v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeOrthogonalSumLineAtSplitPlaceStatement

theorem **_root_.GlobalTheory.T.lattice_orthogonal_sum_line_at_split_place** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n' : Type) [Fintype n'] [DecidableEq n'] (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : Frame p A n') (α₀ : Frame p S (n' ⊕ Fin 1)), IsSumFrame o ζ α α₀ → p.UnitAA → ∀ (LA : Lat A) (LS : Lat S) (ΛA : Submodule 𝒪[k] (n' → k)), IsSelfDualC A LA → IsSelfDualC S LS → α.IsLat LA ΛA → α₀.IsLat LS (lastKer ΛA) → ∃ LL : Lat L, OrthAt o LS LA LL p.v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticePairCharacterIndicatorTransformEqCosetInd

theorem **colExt** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (L T : HSp D) (n : Type) [Fintype n] [DecidableEq n] (β : Frame p T n) (ζ : Fin L.rank → D.host.E) (s : k), s ≠ 0 → ∀ (FL : GlobalTheory.LocMod.Frame k L.rank Unit), (∀ r, FL.β r () = s ∗ (GlobalTheory.LocMod.Place.ofSplitPt p).pr (ζ r)) → ∀ (a : (n → k) → ℂ) (c : SB (Matrix n Unit k)), (∀ M, c.1 M = a (s⁻¹ • fun r => M r ())) → GlobalTheory.LocMod.Witness.ext (GlobalTheory.LocMod.Place.ofSplitPt p) FL (GlobalTheory.LocMod.Frame.ofFinPl β) c = colFun β ζ a

theorem **rowExt** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (T L : HSp D) (n : Type) [Fintype n] [DecidableEq n] (β : Frame p T n) (ζ₀ : Fin L.rank → D.host.E), IsLineVec L ζ₀ → ∀ (FL : GlobalTheory.LocMod.Frame k L.rank Unit), (∀ r, FL.β r () = (GlobalTheory.LocMod.Place.ofSplitPt p).pr (ζ₀ r)) → ∀ (b : (n → k) → ℂ) (Φ : SB (Matrix Unit n k)), (∀ M, Φ.1 M = b (M ())) → GlobalTheory.LocMod.Witness.ext (GlobalTheory.LocMod.Place.ofSplitPt p) (GlobalTheory.LocMod.Frame.ofFinPl β) FL Φ = rowFun β ζ₀ b

theorem **fourier** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : MeasureTheory.Measure k), ν.IsAddHaarMeasure → ∀ (n : Type) [Fintype n] [DecidableEq n] (χ : AddChar k Circle) (hχ : HasConductor χ 1) (q : Unif k) (ψk : AddChar k Circle) (xi s : k), (∀ x, ψk x = χ (xi ∗ x)) → s ≠ 0 → ∀ w : 𝒪[k]ˣ, −(xi ∗ s) = ((w : 𝒪[k]) : k) → ∀ Q : Pair k n, ∃ (C : ℂ) (c : SB (Matrix n Unit k)) (Ψ : SB (Matrix Unit n k)), (∀ M, c.1 M = charInd (resChar χ hχ q) Q (s⁻¹ • fun r => M r ())) ∧ (∀ M, Ψ.1 M = cosetInd q Q.lat ((GlobalTheory.IndefiniteRowSideFullHecke.AppendedVectorData.resUnit w)⁻¹ • Q).red (M ())) ∧ ∀ η, C ∗ Ψ.1 η = GlobalTheory.StepData.hat ψk ν (−1) (fun y => c.1 (Matrix.of fun r _ => y r)) (η ())

theorem **kapStep** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k) (ψ : AddChar (AF D.host.F) 𝕊), IsStdChar D ψ (−1) → p.UnitAA → ∀ (χ : AddChar k Circle) (hχ : HasConductor χ 1), p.IsChar ψ χ → ∀ (n : Type) [Fintype n] [DecidableEq n] (L S L'' : HSp D) (ζ : Fin L.rank → D.host.E) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L ζ → IsLineVec L'' ζ₀ → ∀ (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (lam : D.host.E), Matrix.mulVec τ ζ₀ = lam • ζ → valuation k (p.pr (incA D lam)) = 1 → τ.conjTranspose ∗ (L.gram ∗ τ) = (−(D.host.xi0 ∗ star D.host.xi0)) • L''.gram → ∃ ε : 𝒦[k]ˣ, ∀ (β : Frame p S n) (LS : Lat S), IsSelfDualC S LS → ∀ (Q : Pair k n) (g₁ : WO L S → ℂ) (T₁ : (WO L S → ℂ) →ₗ[ℂ] (WO L S → ℂ)), IsThetaFunO ψ L S g₁ → IsPlaceModel ψ (symplO L S) (heisO ψ L S) p.e (latO L S ((Submodule.span (NumberField

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticePairCharacterIndicatorTransformEqCosetIndStatement

theorem **_root_.GlobalTheory.T.latticePairCharacterIndicator_transform_eq_cosetIndicator** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : Unif k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → p.UnitAA → ∀ (χ : AddChar k Circle) (hχ : HasConductor χ 1), p.IsChar ψ χ → ∀ (n : Type) [Fintype n] [DecidableEq n] (L S L'' : HSp D) (ζ : Fin L.rank → D.host.E) (ζ₀ : Fin L''.rank → D.host.E), IsLineVec L ζ → IsLineVec L'' ζ₀ → ∀ (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (lam : D.host.E), Matrix.mulVec τ ζ₀ = lam • ζ → valuation k (p.pr (incA D lam)) = 1 → τ.conjTranspose ∗ (L.gram ∗ τ) = (-(D.host.xi0 ∗ star D.host.xi0)) • L''.gram → ∃ ε : 𝒦[k]ˣ, ∀ (β : Frame p S n) (LS : Lat S), IsSelfDualC S LS → ∀ (Q : Pair k n) (g₁ : WO L S → ℂ) (T₁ : (WO L S → ℂ) →ₗ[ℂ] (WO L S → ℂ)), IsThetaFunO ψ L S g₁ → IsPlaceModel

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeReductionEqOneOfLocalComponentOne

theorem **redFn_eq_one** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (g : GA D V) (PP : Pl D.host.E), Lev.finE V g ∈ Formalism.L2.DIV.St β → GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram (PP.under (𝒪 D.host.F)) g = 1 → Formalism.L2.DIV.redFn β PP (Lev.finE V g) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeReductionEqOneOfLocalComponentOneStatement

theorem **_root_.GlobalTheory.T.latticeReduction_eq_one_of_local_component_one** : ∀ (D : D7) (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (g : GA D V) (PP : Pl D.host.E), Lev.finE V g ∈ Formalism.L2.DIV.St β → GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram (PP.under (𝒪 D.host.F)) g = 1 → Formalism.L2.DIV.redFn β PP (Lev.finE V g) = 1

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeResiduePairEqScalarOfSameStabilizer

theorem **scal_inv_smul** (s : k) (hs : s ≠ 0) (x : n → k) : (scal s hs : GL n k)⁻¹ • x = s⁻¹ • x

theorem **scal_comm** (s : k) (hs : s ≠ 0) (h : GL n k) : h ∗ scal s hs = scal s hs ∗ h

theorem **transvection_mulVec'** (b d : n) (t : k) (x : n → k) : Matrix.transvection b d t *ᵥ x = x + Pi.single b (t ∗ x d)

theorem **trv_inv_smul** (b d : n) (t : k) (hbd : b ≠ d) (x : n → k) : (trv b d t hbd)⁻¹ • x = x + Pi.single b (-t ∗ x d)

theorem **transvection_smul_stdPair** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (i b d : n) (t : 𝒪[k]) (g : GL n k), b ≠ d → (b ≠ i ∨ t ∈ IsLocalRing.maximalIdeal 𝒪[k]) → g.val = Matrix.transvection b d (t : k) → g • GlobalTheory.LatPair.stdPair k i = GlobalTheory.LatPair.stdPair k i

theorem **eq_scalar_smul_of_stab** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (P P' : GlobalTheory.LatPair.Pair k n), 3 ≤ Fintype.card n → (∀ g : GL n k, g • P = P ↔ g • P' = P') → ∃ (g : GL n k) (s : k), s ≠ 0 ∧ g.val = s • (1 : Matrix n n k) ∧ P' = g • P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeResiduePairEqScalarOfSameStabilizerStatement

theorem **_root_.GlobalTheory.T.latticeResiduePair_eq_scalar_of_same_stabilizer** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (P P' : GlobalTheory.LatPair.Pair k n), 3 ≤ Fintype.card n → (∀ g : GL n k, g • P = P ↔ g • P' = P') → ∃ (g : GL n k) (s : k), s ≠ 0 ∧ g.val = s • (1 : Matrix n n k) ∧ P' = g • P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeVacuumCombinationsDifferByScalar

theorem **psi_ne** {A : Type} [AddGroup A] (ψ : AddChar A ℂ) (a : A) : ψ a ≠ 0

theorem **psi_neg** {A : Type} [AddGroup A] (ψ : AddChar A ℂ) (a : A) : ψ (-a) = (ψ a)⁻¹

theorem **iso_mem** {V ι J : Type} [AddCommGroup V] [Module ℂ V] (A : ι → V →ₗ[ℂ] V) (U : Submodule ℂ V) (hU : ∀ i, ∀ x ∈ U, A i x ∈ U) (l : J → ι → ℂ) (s : Finset J) : ∀ z : J → V, (∀ j ∈ s, ∀ i, A i (z j) = l j i • z j) → (∀ j ∈ s, ∀ j' ∈ s, l j = l j' → j = j') → ∑ j ∈ s, z j ∈ U → ∀ j ∈ s, z j ∈ U

theorem **heisO_zero_fun** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (u : WO S T) : heisO ψ S T u 0 = 0

theorem **heisO_inv** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (m : WO S T) (f : WO S T → ℂ) : heisO ψ S T (-m) (heisO ψ S T m f) = ψ (symplOH S T (-m) m) • f

theorem **eig_ne** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) {m : WO S T} {f : WO S T → ℂ} {c : ℂ} (h : heisO ψ S T m f = c • f) (hf : f ≠ 0) : c ≠ 0

theorem **symplOH_eq** (S T : HSp D) (m y : WO S T) : symplOH S T m y = symplO S T m y + symplOH S T y m

theorem **heisO_eig** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) {m y : WO S T} {f : WO S T → ℂ} {c : ℂ} (h : heisO ψ S T m f = c • f) : heisO ψ S T m (heisO ψ S T y f) = (ψ (symplO S T m y) ∗ c) • heisO ψ S T y f

theorem **vac_eigen** (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (-1)) (S T : HSp D) (LS : GlobalTheory.FinPl.Lat S) (LT : GlobalTheory.FinPl.Lat T) (hLS : IsSelfDualC S LS) (hLT : IsSelfDualC T LT) (Q : Finset (GlobalTheory.FinPl.Pl D)) : ∃ q : WO S T → ℂ, ∀ m ∈ latO S T (LS : Set _) (LT : Set _) (· ∈ Q), ∀ f, GlobalTheory.FinPl.Vac ψ S T LS LT (· ∈ Q) f → heisO ψ S T m f = q m • f

theorem **symplO_split** (S T : HSp D) (m x x' : WO S T) : symplO S T m x = symplO S T m x' + symplO S T m (x - x')

theorem **class_mem** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Λ : AddSubgroup (WO S T)) (P : WO S T → Prop) (hself : ∀ m ∈ Λ, ∀ m' ∈ Λ, ψ (symplO S T m m') = 1) (hann : ∀ x, P x → (∀ m ∈ Λ, ψ (symplO S T m x) = 1) → x ∈ Λ) (hPΛ : ∀ m ∈ Λ, P m) (hPsub : ∀ x y, P x → P y → P (x - y)) (f q : WO S T → ℂ) (hq : ∀ m ∈ Λ, heisO ψ S T m f = q m • f) (hq0 : ∀ m ∈ Λ, q m ≠ 0) (U : Submodule ℂ (WO S T → ℂ)) (hU : ∀ m ∈ Λ, ∀ x ∈ U, heisO ψ S T m x ∈ U) {ι : Type} (s : Finset ι) (y : ι → WO S T) (c : ι → ℂ) (hy : ∀ i ∈ s, P (y i)) (hsum : ∑ i ∈ s, c i • heisO ψ S T (y i) f ∈ U) : ∀ i ∈ s, ∑ j ∈ s with (QuotientAddGroup.mk (y j) : WO S T ⧸ Λ) = QuotientAddGroup.mk (y i), c j • heisO ψ S T (y j) f ∈ U

theorem **class_line** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (Λ : AddSubgroup (WO S T)) (f q : WO S T → ℂ) (hq : ∀ m ∈ Λ, heisO ψ S T m f = q m • f) {ι : Type} (s : Finset ι) (y : ι → WO S T) (c : ι → ℂ) (i : ι) : ∑ j ∈ s with (QuotientAddGroup.mk (y j) : WO S T ⧸ Λ) = QuotientAddGroup.mk (y i), c j • heisO ψ S T (y j) f ∈ ℂ • heisO ψ S T (y i) f

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M N : GlobalTheory.FinPl.Lat S) (M' N' : GlobalTheory.FinPl.Lat T) (Q : Finset (GlobalTheory.FinPl.Pl D)) (o o' : WO S T →ₗ[ℂ] ℂ) (g₀ : WO S T → ℂ), IsStdChar D ψ (-1) → IsSelfDualC S M → IsSelfDualC T M' → IsSelfDualC S N → IsSelfDualC T N' → GlobalTheory.FinPl.CombAt (· ∈ Q) o → GlobalTheory.FinPl.CombAt (· ∈ Q) o' → GlobalTheory.FinPl.Vac ψ S T M M' (· ∈ Q) g₀ → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ Q) (comb ψ S T o g₀) → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ Q) (comb ψ S T o' g₀) → comb ψ S T o' g₀ ≠ 0 → ∃ c : ℂ, comb ψ S T o g₀ = c • comb ψ S T o' g₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeVacuumCombinationsDifferByScalarStatement

theorem **_root_.GlobalTheory.T.latticeVacuumCombinations_differ_by_scalar** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (M N : GlobalTheory.FinPl.Lat S) (M' N' : GlobalTheory.FinPl.Lat T) (Q : Finset (GlobalTheory.FinPl.Pl D)) (o o' : WO S T →₀ ℂ) (g₀ : WO S T → ℂ), IsStdChar D ψ (-1) → IsSelfDualC S M → IsSelfDualC T M' → IsSelfDualC S N → IsSelfDualC T N' → GlobalTheory.FinPl.CombAt (· ∈ Q) o → GlobalTheory.FinPl.CombAt (· ∈ Q) o' → GlobalTheory.FinPl.Vac ψ S T M M' (· ∈ Q) g₀ → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ Q) (comb ψ S T o g₀) → GlobalTheory.FinPl.Vac ψ S T N N' (· ∈ Q) (comb ψ S T o' g₀) → comb ψ S T o' g₀ ≠ 0 → ∃ c : ℂ, comb ψ S T o g₀ = c • comb ψ S T o' g₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeVacuumHeckeIdentitySplitPlaceCharts

lemma **opOf_dualC_splitC** {D : D7} {V : HSp D} {M : Type} [AddCommGroup M] [Module ℂ M] (π : GA D V →* Module.End ℂ M) (P : Formalism.L2.DIV.Pl D.host.E) (n : ℕ) (z : ℕ → ℂ) : GlobalTheory.Hk3.opOf π P (GlobalTheory.Hk3.dualC (GlobalTheory.Hk3.splitC n z)) = ∏ j ∈ Finset.range (n + 1), z j • GlobalTheory.Hk3.opOf π P (Finsupp.single (dualLab (splitLab n j)) 1)

theorem **hecke_vac_split_chart** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ t : ℕ, T.rank = S.rank + t → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeVacuumHeckeIdentitySplitPlaceChartsStatement

theorem **_root_.GlobalTheory.T.latticeVacuum_heckeIdentity_splitPlace_charts** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : GA D S × GA D T →* Module.End ℂ (WO S T → ℂ)), IsWeilDatumO ψ S T χS χT om → ∀ t : ℕ, T.rank = S.rank + t → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (h : w.under (𝒪 D.host.F) = v), IsSplit (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeVacuumIffPlacewise

theorem **latO_mono** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) {P Q : Pl D → Prop} (h : ∀ v, Q v → P v) : latO S T LS LT Q ≤ latO S T LS LT P

theorem **heisO_zero** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (f : WO S T → ℂ) : heisO ψ S T 0 f = f

theorem **pairA_add** (a b : Af D.host.F) : pairA D.host.F 0 (a + b) = pairA D.host.F 0 a + pairA D.host.F 0 b

theorem **pairA_mul** (a b : Af D.host.F) : pairA D.host.F 0 a * pairA D.host.F 0 b = pairA D.host.F 0 (a * b)

theorem **pairA_zero** : pairA D.host.F 0 (0 : Af D.host.F) = 0

theorem **latVac_of_places** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (LS : Lat S) (LT : Lat T) (P : Pl D → Prop) (f : WO S T → ℂ) (hf : IsThetaFunO ψ S T f) (hv : ∀ v, P v → Vac ψ S T LS LT (· = v) f) : IsLatVac (heisO ψ S T) (latO S T (LS : Set _) (LT : Set _) P) f

theorem **vac_loc** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (P : Pl D → Prop), (∀ v, P v → Tame D v) → ∀ f, (Vac ψ S T LS LT P f ↔ IsThetaFunO ψ S T f ∧ ∀ v, P v → Vac ψ S T LS LT (· = v) f) ∧ (Vac ψ S T LS LT P f → ∀ g ∈ grpOff S P, ∀ h ∈ grpOff T P, Vac ψ S T LS LT P (om (g, h) f)) ∧ (Vac ψ S T LS LT P f → ∀ (g : GA D S) (h : GA D T) (MS : Lat S) (MT : Lat T), IsSelfDualC S MS → IsSelfDualC T MT → Moves S g LS MS P → Moves T h LT MT P → Vac ψ S T MS MT P (om (g, h) f))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticeVacuumIffPlacewiseStatement

theorem **_root_.GlobalTheory.T.latticeVacuum_iff_placewise** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ (LS : Lat S) (LT : Lat T), IsSelfDualC S LS → IsSelfDualC T LT → ∀ (P : Pl D → Prop), (∀ v, P v → Tame D v) → ∀ f, (Vac ψ S T LS LT P f ↔ IsThetaFunO ψ S T f ∧ ∀ v, P v → Vac ψ S T LS LT (· = v) f) ∧ (Vac ψ S T LS LT P f → ∀ g ∈ grpOff S P, ∀ h ∈ grpOff T P, Vac ψ S T LS LT P (om (g, h) f)) ∧ (Vac ψ S T LS LT P f → ∀ (g : GA D S) (h : GA D T) (MS : Lat S) (MT : Lat T), IsSelfDualC S MS → IsSelfDualC T MT → Moves S g LS MS P → Moves T h LT MT P → Vac ψ S T MS MT P (om (g, h) f))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticesAgreeAlmostEverywhere

theorem **exists_int_smul_mem_of_mem_span** (M : Submodule (𝒪 E) V) {y : V} (hy : y ∈ Submodule.span E (M : Set V)) : ∃ c : ℤ, c ≠ 0 ∧ c • y ∈ M

theorem **exists_int_smul_le** (L M : Submodule (𝒪 E) V) (hL : L.FG) (hM : Submodule.span E (M : Set V) = ⊤) : ∃ d : ℤ, d ≠ 0 ∧ ∀ x ∈ L, d • x ∈ M

theorem **finite_places_int** (d : ℤ) (hd : d ≠ 0) : {v : Pl D | ((d : 𝒪 D.host.F)) ∈ v.asIdeal}.Finite

theorem **pairA_tmul_int_gen** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (d : ℤ) (hd : d ≠ 0) (α : Af F) (e : E) : pairA F 0 α ⊗ₜ[F] (1 * e) = pairA F 0 ((d : F)⁻¹ • α) ⊗ₜ[F] (1 * (d • e))

theorem **latV_le_of_int_smul** (S : HSp D) (L M : Lat S) (d : ℤ) (hd : d ≠ 0) (hLM : ∀ x ∈ L, d • x ∈ M) (v : Pl D) (hv : ((d : 𝒪 D.host.F)) ∉ v.asIdeal) : latV S L (· = v) ≤ latV S M (· = v)

theorem **latV_eq_off_finite** : ∀ (D : D7) (S : HSp D) (L M : Lat S), Submodule.IsLattice D.host.E L → Submodule.IsLattice D.host.E M → ∃ T : Finset (Pl D), ∀ v, v ∉ T → latV S L (· = v) = latV S M (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticesAgreeAlmostEverywhereStatement

theorem **_root_.GlobalTheory.T.lattices_agree_almost_everywhere** : ∀ (D : D7) (S : HSp D) (L M : Lat S), Submodule.IsLattice D.host.E L → Submodule.IsLattice D.host.E M → ∃ T : Finset (Pl D), ∀ v, v ∉ T → latV S L (· = v) = latV S M (· = v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticesSameSplitShapeCarriedByCentral

theorem **sMoves_of_wShape** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S S₀ A Ln : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (o : OrthDec S₀ A Ln) (ζ : Fin Ln.rank → D.host.E) (L L' : Lat S) (N N' : Lat A) (α α' β β' : Fin ρ → k) (γ δ c₀ : k) (a₀ : GA D A), IsLineVec Ln ζ → IsSelfDualC S L → IsSelfDualC S L' → WShape p d o ζ L α α' γ 1 N → WShape p d o ζ L' β β' δ a₀ N' → latV A N (· = p.v) = latV A N' (· = p.v) → c₀ ≠ 0 → (∀ i, valuation k (β i * c₀) = valuation k (α i)) → (∀ i, valuation k (β' i * c₀) = valuation k (α' i)) → valuation k (δ * c₀) = valuation k γ → ∃ (a : GA D A) (c : AE D), CenAt A p.v a ∧ NormOneAt p.v c ∧ SMoves S c (embA d o (a₀ * a)) L L' p.v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LatticesSameSplitShapeCarriedByCentralStatement

theorem **_root_.GlobalTheory.T.lattices_same_splitShape_carried_by_central** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (S S₀ A Ln : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (o : OrthDec S₀ A Ln) (ζ : Fin Ln.rank → D.host.E) (L L' : Lat S) (N N' : Lat A) (α α' β β' : Fin ρ → k) (γ δ c₀ : k) (a₀ : GA D A), IsLineVec Ln ζ →

IsSelfDualC S L → IsSelfDualC S L' → WShape p d o ζ L α α' γ 1 N → WShape p d o ζ L' β β' δ a₀ N' → latV A N (· = p.v) = latV A N' (· = p.v) → c₀ ≠ 0 → (∀ i, valuation k (β i * c₀) = valuation k (α i)) → (∀ i, valuation k (β' i * c₀) = valuation k (α' i)) → valuation k (δ * c₀) = valuation k γ → ∃ (a : GA D A) (c : AE D), CenAt A p.v a ∧ NormOneAt p.v c ∧ SMoves S c (embA d o (a₀ * a)) L L' p.v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LawsRescaleWithS

theorem **heisO_smul** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (w : WO S T) (s : ℂ) (f : WO S T → ℂ) : heisO ψ S T w (s • f) = s • heisO ψ S T w f

theorem **dRho_smul** (hs : Mx D m n → (Mx D m n → ℂ) → (Mx D m n → ℂ)) (hh : ∀ w (s : ℂ) f, hs w (s • f) = s • hs w f) (θ : InfFam D) (τ : D.what₂.Tau) (z : Matrix (Fin m) (Fin n) ℂ) (s : ℂ) (f : Mx D m n → ℂ) : Datum.dRho hs θ τ z (s • f) = s • Datum.dRho hs θ τ z f

theorem **aPlusAll_smul** (hs : Mx D m n → (Mx D m n → ℂ) → (Mx D m n → ℂ)) (hh : ∀ w (s : ℂ) f, hs w (s • f) = s • hs w f) (θ : InfFam D) (J : D.what₂.Tau → Matrix (Fin m) (Fin n) ℂ → Matrix (Fin m) (Fin n) ℂ) (B : D.what₂.Tau → Matrix (Fin n) (Fin n) ℂ) (B' : D.what₂.Tau → Matrix (Fin m) (Fin m) ℂ) (k) (s : ℂ) (f : Mx D m n → ℂ) : aPlusAll hs θ J B B' k (s • f) = s • aPlusAll hs θ J B B' k f

theorem **opPow_smul** {ι X : Type} (A : ι → (X → ℂ) → (X → ℂ)) (hA : ∀ i (s : ℂ) f, A i (s • f) = s • A i f) (k : ι →₀ ℕ) (s : ℂ) (f : X → ℂ) : opPow A k (s • f) = s • opPow A k f

theorem **pureTensorL_smul** {ι X : Type} (A : ι → (X → ℂ) → (X → ℂ)) (hA : ∀ i (s : ℂ) f, A i (s • f) = s • A i f) (s : ℂ) (f : X → ℂ) : pureTensorL A (s • f) = s • pureTensorL A f

theorem **pureTensorO_smul** (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (s : ℂ) (f : WO S T → ℂ) : pureTensorO ψ θ S T sS sT BS BT (s • f) = s • pureTensorO ψ θ S T sS sT BS BT f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LazardBinomialNormUltrametric

theorem **norm_natCast_eq_one_of_not_dvd** {K : Type} [NormedField K] [IsUltrametricDist K] {l m : ℕ} (hp : l.Prime) (hl : ‖(l : K)‖ < 1) (hm : ¬ l ∣ m) : ‖(m : K)‖ = 1

theorem **not_dvd_choose_pow_mul** {l : ℕ} (hp : l.Prime) (m : ℕ) (hm : ¬ l ∣ m) : ∀ v : ℕ, ¬ l ∣ (l ^ v * m).choose (l ^ v)

theorem **choose_prime_pow_pred** {l : ℕ} (hp : l.Prime) (i : ℕ) (hi : 0 < i) : ∃ m : ℕ, ¬ l ∣ m ∧ (l ^ i).choose (l ^ (i − 1)) = l * m

theorem **lazard_binomial** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ), l.Prime → ‖(l : K)‖ < 1 → ∀ (a : K) (n : ℕ), 2 ≤ n → (∀ k, 0 < k → k < n → ‖a * (n.choose k : K)‖ ≤ ‖(l : K)‖) → ‖a‖ ≤ ‖(l : K)‖ ∨ (‖a‖ ≤ 1 ∧ ∃ i : ℕ, 0 < i ∧ n = l ^ i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LazardBinomialNormUltrametricStatement

theorem **_root_.GlobalTheory.T.lazard_binomial_norm_ultrametric** : ∀ (K : Type) [NormedField K] [IsUltrametricDist K] [CharZero K] (l : ℕ), l.Prime → ‖(l : K)‖ < 1 → ∀ (a : K) (n : ℕ), 2 ≤ n → (∀ k, 0 < k → k < n → ‖a * (n.choose k : K)‖ ≤ ‖(l : K)‖) → ‖a‖ ≤ ‖(l : K)‖ ∨ (‖a‖ ≤ 1 ∧ ∃ i : ℕ, 0 < i ∧ n = l ^ i)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LcvNonnegOfIntegralFull

theorem **lcv_nonneg_of_integral_full** : ∀ (K : Type) [Field K] [NumberField K] [StarRing K] [IsTotallyComplex K] (j : ℕ) (Γ : Matrix (Fin j) (Fin j) K) (Y : InfinitePlace K → Matrix (Fin j) (Fin j) ℂ) (L : Submodule (𝒪 K) (Fin j → K)), (∀ v, IsUnit (Y v).det) → Submodule.IsLattice K L → Γ.det ≠ 0 → Formalism.HermitianLatticesTheory.IsIntegralLat (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevChiLift

theorem **pt_sum** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} (hψ : IsStdChar D ψ (−1)) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) {ι : Type} (s : Finset ι) (a : ι → ℂ) (g : ι → WO S T → ℂ) (hg : ∀ i, IsThetaFunO ψ S T (g i)) (φ : PolF D T.rank S.rank) : IsThetaFunO ψ S T (∑ i ∈ s, a i • g i) ∧ pureTensorO ψ θ S T sS sT BS BT (∑ i ∈ s, a i • g i) φ = ∑ i ∈ s, a i • pureTensorO ψ θ S T sS sT BS BT (g i) φ

theorem **pt_om** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {k : GA D S} (hk : k ∈ GfC S) (φ : PolF D T.rank S.rank) : pureTensorO ψ θ S T sS sT BS BT (om (k, 1) f) φ = om (k, 1) (pureTensorO ψ θ S T sS sT BS BT f φ)

theorem **lift_sum** {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) (hθ : IsInfFam θ) (sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) {f f' : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) {ι : Type} [Fintype ι] (κ : ι → GA D S) (hκ : ∀ i, κ i ∈ GfC S) (a : ι → ℂ) (ha : ∀ i, a i ≠ 0) (hf' : f' = ∑ i, (a i)⁻¹ • om (κ i, 1) f) (ν : Measure (GA D T)) (ℱ : Set (GA D T)) (F : GA D T → ℂ) (φ : PolF D T.rank S.rank) (hi : ∀ g, IntegrableOn (fun h => thetaKO om g h (pureTensorO ψ θ S T sS sT BS BT f φ) * F h) ℱ ν) (U : GA D S → ℂ) (hU : ∀ g, Lift.liftInt ν ℱ (thetaKO om) (pureTensorO ψ θ S T sS sT BS BT f φ) F g = U g) (hl : ∀ i g, U (g * κ i) = a i * U g) : (∀ g, IntegrableOn (fun h => thetaKO om g h (pureTensorO ψ θ S T sS sT BS BT f' φ) * F h) ℱ ν

theorem **levChi_lift** {V : HSp D} {β : GL (Fin V.rank) (Lev.EAf D)} {L : LevDat D.host.E ℂ V.rank} (P : LevelP L.Pg L.r ℂ) (p : P.H) {κ : GA D V} (hκ : IsLevelLift V β L p.1 κ) : κ ∈ levS V β (withP L P)

theorem **step_eigen** {S T : HSp D} {Sp : D.what₂.Tau → SPar} {ψ : AddChar (AF D.host.F) ℂ} {θ : InfFam D} (hθ : IsInfFam θ) (hCR : GlobalTheory.Published.CuspRapidDecay) {χS χT : (AE D)ˣ →* Circle} {om : Om S T} (hom : IsWeilDatumO ψ S T χS χT om) {sS BS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ} {sT BT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ} {f : WO S T → ℂ} (hpt : ∀ P, IsThetaFunO ψ S T (pureTensorO ψ θ S T sS sT BS BT f P)) {M : DSModel S Sp} {β : GL (Fin S.rank) (Lev.EAf D)} {L : LevDat D.host.E ℂ S.rank} (hL : flag1L L = L) {μ : Measure (GA D S)} {ℰ : Set (GA D S)} (hμ : μ.IsHaarMeasure ∧ IsFundamentalDomain (GQ S) ℰ μ) {Ds : Type} [AddCommGroup Ds] [Module ℂ Ds] {It : PolF D T.rank S.rank →ₗ[ℂ] Ds →ₗ[ℂ] Ds →ₗ[ℂ] GA D T → ℂ)} {SmT M} (hIt : It ≠ 0) {CC : Submodule ℂ (Ds →ₗ[ℂ] (GA D T → ℂ))} (hCC : ∀ F : CC, ∀ v, ∀ γ ∈ GQ T, ∀ g, F.1 v (γ * g) = F.1 v g) {β' : GL (Fin T.rank) (Lev.EAf D)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevChiWithPOfMem

theorem **levelCharacter_withLevel_of_mem** {P : LevelP L.Pg L.r ℂ} {κ : GA D V} (hp : rTup β L (Lev.finE V κ) ∈ P.H) : Lev.levChi V β (Lev.withP L P) κ = ((P.chi (_, hp) : ℂˣ) : ℂ) * Lev.levChi V β (Lev.flagL L) κ

theorem **hasType_withCharacter_iff** (P : LevelP L.Pg L.r ℂ) (F : GA D V → ℂ) : Lev.HasType V β (Lev.withP L P) F ↔ Lev.HasType V β (Lev.flag1L L) F ∧ ∀ (p : P.H) κ, Lev.IsLevelLift V β L p.1 κ → ∀ g, (Lev.levChi V β (Lev.flagL L) κ)⁻¹ * F (g * κ) = ((P.chi p : ℂˣ) : ℂ) * F g

theorem **GlobalTheory.ChainOnto.LevelWithP.mem_multT_withP_iff** : ∀ (D : D7) (θ : InfFam D) (V : HSp D) (S : D.what₂.Tau → SPar) (M : DSModel V S) (β : GL (Fin V.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ V.rank) (P : LevelP L.Pg L.r ℂ) (T : SmT M →ₗ[ℂ] (GA D V → ℂ)), T ∈ MultT θ M (levS V β (Lev.withP L P)) (Lev.levChi V β (Lev.withP L P)) ↔ T ∈ MultT θ M (levS V β (Lev.flag1L L)) (Lev.levChi V β (Lev.flag1L L)) ∧ ∀ (p : P.H) κ, Lev.IsLevelLift V β L p.1 κ → ∀ w g, (Lev.levChi V β (Lev.flagL L) κ)⁻¹ * T w (g * κ) = ((P.chi p : ℂˣ) : ℂ) * T w g

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevDatYZero

theorem **lev_datY_zero** {D : D7} (V : HSp D) (β : GL (Fin V.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ V.rank) (r t : ℕ) (hr : r ≤ V.rank) (hn : V.rank = r) (Pg : Finset (Pl D.host.E)) (P : LevelP Pg r ℂ) (χ ξ) (hL : L = datY V.rank r t 0 hr Pg ⌀ P χ ξ Frm.B) (g : GL (Fin V.rank) (Lev.EAf D)) (hU : g ∈ UA D.sp.cm V.gram) : (g ∈ Formalism.L2.DIV.lev D.sp.cm V.gram β L ↔ g ∈ St β ∧ (fun PP : Pg => upperLeftBlock hr (redFn β PP.1 g)) ∈ P.H) ∧ ∀ h : (fun PP : Pg => upperLeftBlock hr (redFn β PP.1 g)) ∈ P.H, levFn β L g = P.chi ⟨_, h⟩

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevOfMem

theorem **lev_of_mem** {D : D7} (θ : InfFam D) {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (J : Set (GA D V)) (χ : GA D V → ℂ) (T : MMap M) (hT : T ∈ Mult θ M J χ) : ∀ v, ∀ k ∈ J, ∀ g, T v (g * k) = χ k * T v g

theorem **valMove** : ∀ D : D7, ValMove D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevTruePackOfPart

theorem **isTypePlaces_vGs** {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (h : I.Ok) : GlobalTheory.Core.Mi.IsTypePlaces (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterEqOneOfDeepAtSplitBadPlace

theorem **adAt_conjP_eq_one** (hc : CMData F E) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : J = 1) {u : GL (Fin n) (AfE F E)} (hu : u ∈ UPts hc.sig J (FiniteAdeleRing (𝒪 F) F)) (P : HeightOneSpectrum (𝒪 E)) (hP : ∀ r s, adAt F P (u.1 r s) = (1 : Matrix (Fin n) (Fin n) (P.adicCompletion E)) r s) (r s : Fin n) : adAt F (conjPrime₂ P) (u.1 r s) = (1 : Matrix (Fin n) (Fin n) ((conjPrime₂ P).adicCompletion E)) r s

theorem **bc_prV_eq_one_of_split** (hc : CMData F E) {n : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : J = 1) {u : GL (Fin n) (AfE F E)} (hu : u ∈ UPts hc.sig J (FiniteAdeleRing (𝒪 F) F)) (P : HeightOneSpectrum (𝒪 E)) (v : HeightOneSpectrum (𝒪 F)) (hover : ∀ w : HeightOneSpectrum (𝒪 E), w.under (𝒪 F) = v → w = P ∨ w = conjPrime₂ P) (hP : ∀ r s, adAt F P (u.1 r s) = (1 : Matrix (Fin n) (Fin n) (P.adicCompletion E)) r s) : bc (prV (E

theorem **exists_root_GL** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) (m n : ℕ) (hn : 0 < n) (c : P.adicCompletion E) (hc : Valued.v c < 1) (A : GL (Fin m) (P.adicCompletion E)) (hA : ∀ r s, Valued.v (A.1 r s - (1 : Matrix (Fin m) (Fin m) (P.adicCompletion E)) r s) ≤ Valued.v ((n : P.adicCompletion E) ^ 2 * c)) : ∃ M : GL (Fin m) (P.adicCompletion E), M ^ n = A ∧ ∀ r s, Valued.v (M.1 r s - (1 : Matrix (Fin m) (Fin m) (P.adicCompletion E)) r s) ≤ Valued.v ((n : P.adicCompletion E) * c)

theorem **chi_eq_one_of_pow** {G : Type} [Group G] {U : Subgroup G} (χ : U →* ℂˣ) (x z : U) (n : ℕ) (hz : χ z ^ n = 1) (hy : χ ((z ^ n)⁻¹ * x) = 1) : χ x = 1

theorem **map_inv_pow_mul_eq_one** {G H : Type} [Group G] [Group H] (f : G →* H) {x z : G} (n : ℕ) (hz : f z = 1) (hx : f x = 1) : f ((z ^ n)⁻¹ * x) = 1

theorem **chi_eq_one_of_deep** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) (T i : ℕ) (hi : i ≤ B.R.m) (bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))) (S : StepLevel (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) (gramN (I.rec7 l Z) i hi) T (I.rec7 l Z).R.f bad) (n : ℕ), 0 < n → (∀ κ : S.U, S.chi κ ^ n = 1) → ∀ (P : SBof I a) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)), Formalism.L2.Col.IsOver v (GlobalTheory.TwU.seamP (sm I l Z) P.1) → ∃ N : ℕ, ∀ x : S.U, (∀ v', v' ≠ v → bc (prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterEqOneOfDeepAtSplitBadPlaceStatement

theorem **_root_.GlobalTheory.T.levelCharacter_eq_one_of_deep_at_splitBadPlace** : ∀ {B : Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) (T i : ℕ) (hi : i ≤ B.R.m) (bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))) (S : StepLevel (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) (gramN (I.rec7 l Z) i hi) T (I.rec7 l Z).R.f bad) (n : ℕ), 0 < n → (∀ κ : S.U, S.chi κ ^ n = 1) → ∀ (P : SBof I a) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)), Formalism.L2.Col.IsOver v (GlobalTheory.TwU.seamP (sm I l Z) P.1) → ∃ N : ℕ, ∀ x : S.U, (∀ v', v' ≠ v → bc (prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterMatchesWitness

theorem **twistedAdelicData_characters_eq_frame** (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Fr B.toPre) (T : TwDat B) (w : Bl.ColW B f T) (Z : w.Lt) (Wt : Set B.places.Pl) (L : SizeData₂ (w.st Z).toD7) (hT : GatTies (w.st Z).toD7) (Gi : G.Inst (w.st Z).toD7 L) (Gl : G.LatInst (w.st Z).toD7) (Kc : G.Consts) (Yd : DivX (w.st Z).toD7) (cX : ExtraC) (ex : Extra (w.st Z).toD7 L hT Gi Gl Kc Yd cX) (t : Bl.RowsT K w Z Wt ex.p0) (a : R3.TwArgs ex t) (j : Fin B.R.m) : Formalism.IndefR.Links.ChrEq (D

theorem **level_character_matches_witness** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → ∀ j : Fin q.B.R.m, Formalism.IndefR.Links.ChrEq ((pk (packB b)).mu0 ((j : ℕ) + 1)) (q.w.I.C.mu0 j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterMatchesWitnessStatement

theorem **_root_.GlobalTheory.T.chrEq_mu0_of_bWit** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → ∀ j : Fin q.B.R.m, Formalism.IndefR.Links.ChrEq ((pk (packB b)).mu0 ((j : ℕ) + 1)) (q.w.I.C.mu0 j)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterOnParahoricChartImage

theorem **levChi_chart** : ∀ (D : D7) (c : Com D) (w : Twist D), w.Is c → ∀ j ≤ D.R.m, ∀ s : PiLocalization₂ D, IsStage c w j s → ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ ι, IsChart s j φ ι → ∀ ξl, LiftsChar φ PP ξl (w.ξ PP) → ∀ y ∈ parahoric (Fv D.host.F v) (Fin D.R.r) (Blocks j D.R.t), ∀ d : (Fv D.host.F v)ˣ, (d : Fv D.host.F v) = y.1.toBlocks₂₂.det → Lev.levChi s.V s.β s.L (ι y) = (((ξl d)⁻¹ : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterOnParahoricChartImageStatement

theorem **_root_.GlobalTheory.T.level_character_on_parahoric_chart_image** : ∀ (D : D7) (c : Com D) (w : Twist D), w.Is c → ∀ j ≤ D.R.m, ∀ s : PiLocalization₂ D, IsStage c w j s → ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ ι, IsChart s j φ ι → ∀ ξl, LiftsChar φ PP ξl (w.ξ PP) → ∀ y ∈ parahoric (Fv D.host.F v) (Fin D.R.r) (Blocks j D.R.t), ∀ d : (Fv D.host.F v)ˣ, (d : Fv D.host.F v) = y.1.toBlocks₂₂.det → Lev.levChi s.V s.β s.L (ι y) = (((ξl d)⁻¹ : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterTrivialNearOneOfLocalBadPlaces

theorem **chi_eq_one_of_split** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 F))) (S : StepLevel F E σ J T f bad) (u : GL (Fin n) (E ⊗[F] FiniteAdeleRing (𝒪 F) F)) (s : Finset (HeightOneSpectrum (𝒪 F))) (H : ∀ v ∈ s, ∀ x : S.U, (∀ v', v' ≠ v → bc (prV (E

theorem **main** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 F))) (S : StepLevel F E σ J T f bad) (Vs : Set (HeightOneSpectrum (𝒪 F))), Vs.Finite → (∀ v ∈ bad, v ∈ Vs) → (letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterTrivialNearOneOfLocalBadPlacesStatement

theorem **_root_.GlobalTheory.T.levelCharacter_trivial_near_one_of_local_bad_places** : ∀ {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (σ : E →ₐ[F] E) {n : ℕ} (J : Matrix (Fin n) (Fin n) E) (T f : ℕ) (bad : Set (HeightOneSpectrum (𝒪 F))) (S : StepLevel F E σ J T f bad) (Vs : Set (HeightOneSpectrum (𝒪 F))), Vs.Finite → (∀ v ∈ bad, v ∈ Vs) → (letI

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterTwistedClass

theorem **chi_deep** (hc : CMData F E) {n T f : ℕ} {J : Matrix (Fin n) (Fin n) E} (hJ : J = 1) {bad : Set (HeightOneSpectrum (𝒪 F))} (S : StepLevel F E hc.sig J T f bad) (m : ℕ) (hm : 0 < m) (hchi : ∀ κ : S.U, S.chi κ ^ m = 1) (P : HeightOneSpectrum (𝒪 E)) (hne : conjPrime₂ P ≠ P) (hcj : (conjPrime₂ P).under (𝒪 F) = P.under (𝒪 F)) (hq : Valued.v ((qOf P : ℕ) : P.adicCompletion E) = WithZero.exp (-1 : ℤ)) (v : HeightOneSpectrum (𝒪 F)) (hv : IsOver v P) : ∃ N : ℕ, ∀ x : S.U, (∀ v', v' ≠ v → bc (prV (E

theorem **chi_near_one** (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) (a : I.ι) {bad : Set (HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F))} (S : StepLevel (I.rec7 l Z).host.F (I.rec7 l Z).host.E (sigD (I.rec7 l Z)) (gramP (I.rec7 l Z)) 1 (I.rec7 l Z).R.f bad) (hbad : ∀ v ∈ bad, v ∈ badG (I.rec7 l Z) v I.UnderQ a v) (hfin : ∃ n : ℕ, 0 < n ∧ ∀ κ, S.chi κ ^ n = 1) : letI

theorem **underQ_of_ram** {I : TC B X} (hI : I.Ok) (a : I.ι) (v : HeightOneSpectrum (𝒪 B.host.F)) (h : ∃ w ∈ Formalism.L2.Host.aboveF (L

theorem **exists_rep** {E : Type} [Field E] [NumberField E] (P : HeightOneSpectrum (𝒪 E)) (x : (𝒪 E ⧸ P.asIdeal)ˣ) : ∃ (b : 𝒪 E) (k : Eˣ), (k : E) = b ∧ (x : 𝒪 E ⧸ P.asIdeal) = Ideal.Quotient.mk P.asIdeal b

theorem **levRead_of** (hI : I.Ok) (hP : TC.Pins B X) (l : I.Laws) (Z : TC.Later B (I.cols l)) {T : Type} (L : LevCh (I.rec7 l Z) T) (μ0 : T) (hL : L.Is μ0) (hC : ∀ j, L.C.mu0 j = I.C.mu0 j) (tup : T → I.ι) (hxi : ∀ t, L.xiH t = I.xi (tup t)) (hPg : L.Pg = (B.primes X).vGs) (hQ : ∀ t, L.Q t = I.Q (tup t)) (hβ : L.βX = 1) (eG : R3.Gc B X ≃* Formalism.L2.DIV.G0 L.Pg (I.rec7 l Z).R.r) (hG : ∀ (g : R3.Gc B X) (P : Pl (I.rec7 l Z).host.E) (h : P ∈ (B.primes X).vGs) (h' : P ∈ L.Pg), eG g ⟨P, h'⟩ = g ⟨P, h⟩) : LevRead I l Z L (levPOf eG B.embLL (R3.lev1 I) (R3.chr1 I)) tup

theorem **levRead_rows** (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Top.Fr B.toPre) (T : TwDat B) (w : Bl.ColW B f T) (Z : w.Lt) (Wt : Set B.places.Pl) (p0 : BaseLevelRowSide (w.D Z)) (t : Bl.RowsT K w Z Wt p0) : LevRead w.I w.l Z t.r.z.R.toLevCh (t.r.z.R.toRealDat.levR t.r.z.F.lv2 true) t.r.Y.tup

theorem **levRead_twArgs** (K : Ctx) (B : GlobalTheory.Mem.Base) (f : Top.Fr B.toPre) (T : TwDat B) (w : Bl.ColW B f T) (Z : w.Lt) (Wt : Set B.places.Pl) (L : SizeData₂ (w.st Z).toD7) (hT : GatTies (w.st Z).toD7) (Gi : G.Inst (w.st Z).toD7 L) (Gl : G.LatInst (w.st Z).toD7) (Kc : G.Consts) (Yd : DivX (w.st Z).toD7) (cX : ExtraC) (ex : Extra (w.st Z).toD7 L hT Gi Gl Kc Yd cX) (t : Bl.RowsT K w Z Wt ex.p0) (a : R3.TwArgs ex t) : LevRead w.I w.l Z a.x.R.toLevCh (a.x.R.toRealDat.levR a.x.F.lv2 true) a.x.Y.tup

theorem **levRead_of_bWit** {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c) (hw : Bl.BWit q b) : LevRead q.w.I q.w.l q.Z (pk (packB b)).toLevCh (lvP (packB b)) (packB b).P.I.tup

theorem **levIs_of_bWit** {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c) (hw : Bl.BWit q b) (u : (q.w.D q.Z).what₂.Word) (t : (packB b).P.T) (h : (packB b).P.I.tup t ≠ (q.w.D q.Z).what₂.mu0) : LevIs ((packB b).R.T.tw.pt.cls u ((packB b).P.I.tup t) h).lev (levS _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t)) (levChi _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t))

theorem **levChi_of_bWit** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → ∀ (u : (q.w.D q.Z).what₂.Word) (t : (packB b).P.T) (h : (packB b).P.I.tup t ≠ (q.w.D q.Z).what₂.mu0) (g : GA (q.w.D q.Z) (q.w.D q.Z).sp.X') (κ : ((packB b).R.T.tw.pt.cls u ((packB b).P.I.tup t) h).lev.U), GlobalTheory.Core.Rk.Top.HasMat g κ.1 → (levChi _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t)) g = star ((((packB b).R.T.tw.pt.cls u ((packB b).P.I.tup t) h).lev.chi κ : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCharacterTwistedClassStatement

theorem **_root_.GlobalTheory.T.level_character_twisted_class** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → ∀ (u : (q.w.D q.Z).what₂.Word) (t : (packB b).P.T) (h : (packB b).P.I.tup t ≠ (q.w.D q.Z).what₂.mu0) (g : GA (q.w.D q.Z) (q.w.D q.Z).sp.X') (κ : ((packB b).R.T.tw.pt.cls u ((packB b).P.I.tup t) h).lev.U), GlobalTheory.Core.Rk.Top.HasMat g κ.1 → (levChi _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t)) g = star ((((packB b).R.T.tw.pt.cls u ((packB b).P.I.tup t) h).lev.chi κ : ℂˣ) : ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCongruenceIffFlagConditions

theorem **inv_mulVec_col** (B C : Matrix (Fin n) (Fin n) A) (hCB : C * B = 1) (u : A) (j i : Fin n) : (C *ᵥ (u • fun l => B l j)) i = if i = j then u else 0

theorem **col_iff** (B C m X : Matrix (Fin n) (Fin n) A) (hCB : C * B = 1) (hX : X * B = B * m) (hBC : B * C = 1) (hm : ∀ i j, m i j ∈ S) (M : Submodule S (Fin n → A)) (hM : ∀ y, y ∈ M ↔ ∀ i, ∃ t ∈ S, (C *ᵥ y) i = g * t) (T : Set (Fin n → A)) (N : ℕ) (ε : A) (j : Fin n) (v : Fin n → A) (hv : (ε • fun l => B l j) - v ∈ M) (h1 : ∀ s ∈ T, ∃ (u : A) (i : Fin n), i.val < N ∧ (u • fun l => B l i) - s ∈ M) (h2 : ∀ i : Fin n, i.val < N → (ε • fun l => B l i) ∈ Submodule.span S T ⊔ M) : X *ᵥ v ∈ Submodule.span S T ⊔ M ↔ ∀ i : Fin n, N ≤ i.val → ∃ t ∈ S, ε * m i j = g * t

theorem **hatL_eq** (I : AutInst) : Formalism.L2.Herm.hatL (F

theorem **mem_mL_iff** (I : AutInst) (g : ℕ) (β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) (hβ : IsFrame β I.L) (y : VA I) : y ∈ mL I g ↔ ∀ i, ∃ t ∈ IntAf I.F I.E, (((β⁻¹ : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) : Matrix (Fin I.n) (Fin

I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) *ᵥ y) i = (g : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) * t

theorem **matIn_int** (I : AutInst) (β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) (x : I.Gf) (hx : (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β) (i j : Fin I.n) : (matIn I β x - 1) i j ∈ IntAf I.F I.E

theorem **k2cong_iff_flagCond** (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F) (hR : IsRefFrame I g e0 f0 z0 bw bwd ew) (hn : I.n = 2 * ρ + k + 1) (β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) (hβ : IsLiftFrame I g (flagVec e0 f0 z0 bw bwd) β) (x : I.Gf) (hx : (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β) : K2cong I g e0 z0 bw x ↔ FlagCond (ZMod' I g) ρ k ew (matIn I β x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelCongruenceIffFlagConditionsStatement

theorem **_root_.GlobalTheory.T.levelCongruence_iff_flag_conditions** : ∀ (I : AutInst) (g : ℕ) {ρ k : ℕ} (e0 f0 : Fin ρ → VA I) (z0 : VA I) (bw bwd : Fin k → VA I) (ew : I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsRefFrame I g e0 f0 z0 bw bwd ew → I.n = 2 * ρ + k + 1 → ∀ β : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F), IsLiftFrame I g (flagVec e0 f0 z0 bw bwd) β → ∀ x : I.Gf, (x : GL (Fin I.n) (I.E ⊗[I.F] Formalism.MainNotions.S.Af I.F)) ∈ St β → (K2cong I g e0 z0 bw x ↔ FlagCond (ZMod' I g) ρ k ew (matIn I β x))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelGroupPropertyClosedMul

theorem **redFn_mul** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) (PP : Pl E) {x y : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} (hx : x ∈ St β) (hy : y ∈ St β) : redFn β PP (x * y) = redFn β PP x * redFn β PP y

theorem **redFn_inv** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (β : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)) (PP : Pl E) {x : GL (Fin n) (E ⊗[F] Formalism.L2.Herm.Af F)} (hx : x ∈ St β) : redFn β PP x⁻¹ = (redFn β PP x)⁻¹

theorem **reductionBlock_injective_of_eq** {κ : Type} [CommRing κ] {n r : ℕ} (hr : r ≤ n) (hn : n = r) : Function.Injective (upperLeftBlock (κ

theorem **red_unique** {D : D7} {u : D.what₂.Word} {d : GlobalTheory.SeeSaw.Dat D u} {τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E} {PP : Pl D.host.E} (G : GlobalTheory.LocMod.GAt d τ PP) {x : G.S.P.J} {g g' : GL (Fin D.R.r) (kP PP)} (h : G.Red x g) (h' : G.Red x g') : g = g'

theorem **GlobalTheory.LevZeroInd.gen** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), d.Base → IsTypePlaces d.c.Pg → GlobalTheory.LocMod.PinGBy G cG → ∀ Q : GA D (d.st 0).V → Prop, (∀ k ∈ d.lev 0, ∀ k' ∈ d.lev 0, Q k → Q k' → Q (k * k')) → (∀ (PP : d.c.Pg) (n : GA D (d.st 0).V) (q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (d.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (GlobalTheory.LocMod.ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → (d.t 0)⁻¹ * n * d.t 0 ∈ d.lev 0 → Q ((d.t 0)⁻¹ * n * d.t 0)) → (∀ b ∈ GfC (D.sp.X 0), (d.t 0)⁻¹ * embA (

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelGroupPropertyClosedMulAltBis

theorem **_root_.GlobalTheory.T.level_group_property_closed_mul_alt** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), d.Base → IsTypePlaces d.c.Pg → GlobalTheory.LocMod.PinGBy G cG → ∀ Q : GA D (d.st 0).V → Prop, (∀ k ∈ d.lev 0, ∀ k' ∈ d.lev 0, Q k → Q k' → Q (k * k')) → (∀ (PP : d.c.Pg) (n : GA D (d.st 0).V) (q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (d.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (GlobalTheory.LocMod.ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → (d.t 0)⁻¹ * n * d.t 0 ∈ d.lev 0 → Q ((d.t 0)⁻¹ * n * d.t 0)) → (∀ b ∈ GfC

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelGroupPropertyClosedMulStatement

theorem **_root_.GlobalTheory.T.level_group_property_closed_mul** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), d.Base → IsTypePlaces d.c.Pg → GlobalTheory.LocMod.PinGBy G cG → ∀ Q : GA D (d.st 0).V → Prop, (∀ k ∈ d.lev 0, ∀ k' ∈ d.lev 0, Q k → Q k' → Q (k * k')) → (∀ (PP : d.c.Pg) (n : GA D (d.st 0).V) (q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), n ∈ flagN (d.bes 0 D.R.rS h0).d → q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (GlobalTheory.LocMod.ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → (d.t 0)⁻¹ * n * d.t 0 ∈ d.lev 0 → Q ((d.t 0)⁻¹ * n * d.t 0)) → (∀ b ∈ GfC (D.s

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelGroupReferenceTuple

theorem **levFn_congr** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n r t s : ℕ} (h : r ≤ n) (β) {SG Q : Finset (Pl E)} (P : LevelP SG r ℂ) (cg cg' : ℕ → ∀ PP : Pl E, (kP PP)ˣ →* ℂˣ) (xq : ∀ PP : Pl E, (kP PP)ˣ →* ℂˣ) (e : ∀ c, c < s → ∀ PP ∈ SG, cg (c + 1) PP = cg' (c + 1) PP) (g) : levFn (F

theorem **isRes_unique** {D : D7} {χ : Formalism.L2.Host.HChar D.host.E} {P : Pl D.host.E} {c c' : (kP P)ˣ →* ℂˣ} (h : IsRes χ P c) (h' : IsRes χ P c') : c = c'

theorem **isRes_chiC** (h : I.Ok) (j : Fin B.R.m) (PP : Pl (I.rec7 l Z).host.E) (hw : PP ∈ (B.primes X).vGs) : IsRes (D

theorem **topLev_datY** (ok : I.Ok) (pins : TC.Pins B X) (v : B.Word) {Ch : ℕ → Type} {chOf : ∀ i, ((I.rec7 l Z).mem v).C.cls i → Ch i} (F : FormsTiedSym (I.rec7 l Z) v Ch chOf) (hF : I.StdF l Z v F) (hr : (I.rec7 l Z).R.r ≤ (I.rec7 l Z).sp.X'.rank) (P' : LevelP (B.primes X).vGs B.R.r ℂ) (cg : ℕ → ∀ PP : Pl B.fields.E, (kP PP)ˣ →* ℂˣ) (ξ : ∀ PP : Pl B.fields.E, (kP PP)ˣ →* ℂˣ) (hH : ∀ g, g ∈ P'.H ↔ ∀ w, g w ∈ (I.Ps w).H) (hchi : ∀ (g : P'.H) (hg : ∀ w, g.1 w ∈ (I.Ps w).H), ((P'.chi g : ℂˣ) : ℂ) = ∏ w, B.embLL (((I.Ps w).lam (g.1 w, hg w) : B.fields.Kˣ) : B.fields.K)) (hcg : ∀ (j : Fin B.R.m), ∀ PP ∈ (B.primes X).vGs, IsRes (D

theorem **topLev_levPOf** (ok : I.Ok) (pins : TC.Pins B X) (v : B.Word) {Ch : ℕ → Type} {chOf : ∀ i, ((I.rec7 l Z).mem v).C.cls i → Ch i} (F : FormsTiedSym (I.rec7 l Z) v Ch chOf) (hF : I.StdF l Z v F) (hr : (I.rec7 l Z).R.r ≤ (I.rec7 l Z).sp.X'.rank) (Pg Q : Finset (Pl (I.rec7 l Z).host.E)) (β : GL (Fin (I.rec7 l Z).sp.X'.rank) (EAf (I.rec7 l Z))) (hPg : Pg = (B.primes X).vGs) (hQ : Q = ∅) (hβ : β = 1) (eG : R3.Gc B X ≃* G0 Pg (I.rec7 l Z).R.r) (hG : ∀ (g : R3.Gc B X) (P : Pl (I.rec7 l Z).host.E) (h : P ∈ (B.primes X).vGs) (h' : P ∈ Pg), eG g ⟨P, h'⟩ = g ⟨P, h⟩) (cg : ℕ → ∀ PP : Pl (I.rec7 l Z).host.E, (kP PP)ˣ →* ℂˣ) (ξ : ∀ PP : Pl (I.rec7 l Z).host.E, (kP PP)ˣ →* ℂˣ) (hcg : ∀ (j : Fin B.R.m), ∀ PP ∈ Pg, IsRes (D

theorem **topLev_of_btw** {D : D7} {O : Obj D} {c : ℝ} (b : GlobalTheory.Fc.Bundle D O c) (x : R3.TwIn O.ex) (y : Bl.BTw b x) {u : D.what₂.Word} (K₀ : ClsPTied u) (h : PairDataBaseTupleLevelEqualsClassLevel x K₀) (t : (packB b).P.T) (ht : (packB b).P.I.tup t =

D.what₂.mu0) : GlobalTheory.Core.Rk.Top.TopLev K₀ (levS _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t)) (levChi _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t))

theorem **Top_of_args** {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) (a : R3.TwArgs q.O.ex q.x.what₂) (v : q.B.Word) {Ch : ℕ → Type} {chOf : ∀ i, ((q.w.D q.Z).mem v).C.cls i → Ch i} (F : FormsTiedSym (q.w.D q.Z) v Ch chOf) (hF : TC.StdF q.w.I q.w.l q.Z v F) : PairDataBaseTupleLevelEqualsClassLevel (O

theorem **levG_base_of_bWit** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → ∀ (u : (q.w.D q.Z).what₂.Word) (t : (packB b).P.T), (packB b).P.I.tup t = (q.w.D q.Z).what₂.mu0 → LevG ((packB b).R.T.cols.forms u).clsP.toG 1 (levS _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t)) (levChi _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelGroupReferenceTupleStatement

theorem **_root_.GlobalTheory.T.level_group_reference_tuple** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → ∀ (u : (q.w.D q.Z).what₂.Word) (t : (packB b).P.T), (packB b).P.I.tup t = (q.w.D q.Z).what₂.mu0 → LevG ((packB b).R.T.cols.forms u).clsP.toG 1 (levS _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t)) (levChi _ (pk (packB b)).βX (datXP (pk (packB b)) (lvP (packB b)) t))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelMatchOfWitness

theorem **idE_levelMatch_of_bwit** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → GlobalTheory.Core.Mi.IdE b.S.X ∧ ∀ u, Formalism.L2.SG.ColDict.LevelMatch b.S.X (b.S.tr u) b.S.Y.hK

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelMatchOfWitnessStatement

theorem **_root_.GlobalTheory.T.level_match_of_witness** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → GlobalTheory.Core.Mi.IdE b.S.X ∧ ∀ u, Formalism.L2.SG.ColDict.LevelMatch b.S.X (b.S.tr u) b.S.Y.hK

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelMeasurePushforwardEqProductRealMeasures

theorem **exists_realPart** (θ : InfFam D) (hθ : IsInfFam θ) (W : HSp D) (hW : W.SigOK) (Fr : Frames W) (hFr : IsFrames W Fr) : ∃ φ : GA D W → ∀ ρ, Upq W ρ, Continuous φ ∧ ∀ h ρ, UpqT.mat (φ h ρ) ∗ Fr ρ = Fr ρ ∗ infMat D.sp.cm.sig W.gram (θ ρ) h

theorem **mat_cancel** (W : HSp D) (hW : W.SigOK) (Fr : Frames W) (hFr : IsFrames W Fr) (ρ : D.what₂.Tau) {b b' : Upq W ρ} (h : UpqT.mat b ∗ Fr ρ = UpqT.mat b' ∗ Fr ρ) : b = b'

theorem **realMat_of** (W : HSp D) (Fr : Frames W) (hFr : IsFrames W Fr) (ρ : D.what₂.Tau) (b : Upq W ρ) (M : Matrix (Fin W.rank) (Fin W.rank) ℂ) (h : UpqT.mat b ∗ Fr ρ = Fr ρ ∗ M) : realMat W Fr ρ b = M

theorem **realMat_mem** (W : HSp D) (hW : W.SigOK) (Fr : Frames W) (hFr : IsFrames W Fr) (ρ : D.what₂.Tau) (b : Upq W ρ) : realMat W Fr ρ b ∈ Formalism.QuantumDeterminants.UH (gramAt D W ρ)

theorem **infMat_realElt** (θ : InfFam D) (hθ : IsInfFam θ) (W : HSp D) (u : D.what₂.Tau → Matrix (Fin W.rank) (Fin W.rank) ℂ) (hu : ∀ τ, u τ ∈ Formalism.QuantumDeterminants.UH (gramAt D W τ)) (ρ : D.what₂.Tau) : infMat D.sp.cm.sig W.gram (θ ρ) (realElt θ W u) = u ρ

theorem **realElt_mem** (θ : InfFam D) (W : HSp D) (u : D.what₂.Tau → Matrix (Fin W.rank) (Fin W.rank) ℂ) : realElt θ W u ∈ GinfC W

theorem **finE_realElt** (θ : InfFam D) (W : HSp D) (u : D.what₂.Tau → Matrix (Fin W.rank) (Fin W.rank) ℂ) : finE W (realElt θ W u) = 1

theorem **continuous_realElt** (θ : InfFam D) (hθ : IsInfFam θ) (W : HSp D) (hW : W.SigOK) (Fr : Frames W) (hFr : IsFrames W Fr) : Continuous fun b : (∀ ρ, Upq W ρ) => realElt θ W fun ρ => realMat W Fr ρ (b ρ)

theorem **lintegral_levelSet_eq_pi** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (W : HSp D), W.SigOK → ∀ (K : Subgroup (GL (Fin W.rank) (Lev.EAf D))) (ν : Measure (GA D W)), IsMeasLev W K ν → ∀ Fr : Frames W, IsFrames W Fr → ∀ mR : ∀ ρ, Measure (Upq W ρ), (∀ ρ, Lev.IsMinf (mR ρ)) → ∀ Z : GA D W → ENNReal, Measurable Z → (∀ g, ∀ k ∈ levA W K, Z (g ∗ k) = Z g) → MeasurableSet {h : GA D W | finE W h ∈ K} → ∫⁻ g in {h : GA D W | finE W h ∈ K}, Z g ∂ν = ∫⁻ b, Z (realElt θ W fun ρ => realMat W Fr ρ (b ρ)) ∂(Measure.pi mR)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelMeasurePushforwardEqProductRealMeasuresStatement

theorem **_root_.GlobalTheory.T.level_measure_pushforward_eq_product_real_measures** : ∀ (D : D7) (θ : InfFam D), IsInfFam θ → ∀ (W : HSp D), W.SigOK → ∀ (K : Subgroup (GL (Fin W.rank) (Lev.EAf D))) (ν : Measure (GA D W)), IsMeasLev W K ν → ∀ Fr : Frames W, IsFrames W Fr → ∀ mR : ∀ ρ, Measure (Upq W ρ), (∀ ρ, Lev.IsMinf (mR ρ)) → ∀ Z : GA D W → ENNReal, Measurable Z → (∀ g, ∀ k ∈ levA W K, Z (g ∗ k) = Z g) → MeasurableSet {h : GA D W | finE W h ∈ K} → ∫⁻ g in {h : GA D W | finE W h ∈ K}, Z g ∂ν = ∫⁻ b, Z (realElt θ W fun ρ => realMat W Fr ρ (b ρ)) ∂(Measure.pi mR)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlaceFrameExists

theorem **tOf_fix_first** {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (hW : PkHyp W) (hr : 1 ≤ D.R.rS) (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (a₀ a : GA D (D.sp.X 0)) (c : AE D) : Matrix.mulVec (Formalism.L2.Theta.mt (D.sp.Y 0) (tOf f B a₀ a)⁻¹) (fun i => c ∗ Formalism.L2.Theta.incA D (W.S.X.ιE (W.S.X.eR (0, hr) (Fin.cast (GlobalTheory.Core.Mi.rank_eq W.S.X W.Q.instEq) i)))) = Matrix.mulVec (Formalism.L2.Theta.mt (D.sp.Y 0) (show GA D (D.sp.Y 0) from f.g.b 0)⁻¹) (fun i => c ∗ Formalism.L2.Theta.incA D (W.S.X.ιE (W.S.X.eR (0, hr) (Fin.cast (GlobalTheory.Core.Mi.rank_eq W.S.X W.Q.instEq) i))))

theorem **mem_latLoc_line_iff** {D : D7} (V : Formalism.BasicNotions.TH.HSp D) (L : GlobalTheory.FinPl.Lat V) (hL : L.FG) (v : GlobalTheory.FinPl.Pl D) (t : GA D V) (ε : Fin V.rank → D.host.E) (y : AE D) (hy : y ∈ GlobalTheory.FinPl.idAt D (· = v)) : Matrix.mulVec ((GlobalTheory.AdelicU.matHom (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) D.sp.cm.sig V.gram (GlobalTheory.AdelicU.proj D.sp.cm.sig V.gram v t))⁻¹).val (GlobalTheory.AdelicU.vecLoc v (finPart D ∘ fun _ : Fin 1 => y) 0 • fun i => (1 : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v) ⊗ₜ[D.host.F] ε i) ∈ GlobalTheory.AdelicU.latLoc D.host.F L v ↔ Matrix.mulVec (Formalism.L2.Theta.mt V t⁻¹) (fun i => y ∗ Formalism.L2.Theta.incA D (ε i)) ∈ GlobalTheory.FinPl.latV V L (· = v)

theorem **line_atG** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → 1 ≤ D.R.rS → ∀ v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F), (∃ PP ∈ (pk W).Pg, PP.under (𝒪 D.host.F) = v) → ∃ (e : Fin (D.sp.Y 0).rank → Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E) (α : (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ), ∀ u (f : FamCh W u) (B

: SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ a, IsAt f B a₀ a → ∀ c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, Matrix.mulVec ((GlobalTheory.AdelicU.matHom (Formalism.MainNotions.GlobalT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlaceFrameExistsStatement

theorem **_root_.GlobalTheory.T.level_place_frame_exists** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → 1 ≤ D.R.rS → ∀ v : IsDedekindDomain.HeightOneSpectrum (𝒪 D.host.F), (∃ PP ∈ (pk W).Pg, PP.under (𝒪 D.host.F) = v) → ∃ (e : Fin (D.sp.Y 0).rank → Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E) (α : (Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E)ˣ), ∀ u (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), Placed f B → ∀ a₀ a, IsAt f B a₀ a → ∀ c : Formalism.MainNotions.GlobalTheoryTheory.GenusMassVolumes.Fv D.host.F v ⊗[D.host.F] D.host.E, Matrix.mulVec ((GlobalTheory.AdelicU

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesDatumInvLeftMulTargetLevel

theorem **IsTopB.move** (hab : a ≤ b) (ea : a = r + j * t) (eb : b = r + (j + 1) * t) {y : Matrix (Fin b) (Fin a) κ} {g : GL (Fin a) κ} {H : GL (Fin b) κ} (h : IsTopB r t y g) (hT : H ∈ blockTri (blkOf r t)) : IsTopB r t ((H⁻¹).val * y) ((upperLeftBlock hab H)⁻¹ * g)

theorem **restrictBlock_inv_mul_block_triangular_move** (hab : a ≤ b) (ha : r ≤ a) (hb : r ≤ b) (ea : a = r + j * t) (eb : b = r + (j + 1) * t) {g : GL (Fin a) κ} {H : GL (Fin b) κ} (hg : g ∈ blockTri (blkOf r t)) (hT : H ∈ blockTri (blkOf r t)) : upperLeftBlock ha ((upperLeftBlock hab H)⁻¹ * g) = (upperLeftBlock hb H)⁻¹ * upperLeftBlock ha g

theorem **char_move** (hab : a ≤ b) (ea : a = r + j * t) (eb : b = r + (j + 1) * t) {A : Type} [CommGroup A] (χ : ℕ → κˣ →* A) {g : GL (Fin a) κ} {H : GL (Fin b) κ} (hg : g ∈ blockTri (blkOf r t)) (hT : H ∈ blockTri (blkOf r t)) : (∏ c ∈ Finset.range j, χ (c + 1) (dBlk (blkOf r t) (c + 1) ((upperLeftBlock hab H)⁻¹ * g))) * (χ (j + 1) (Matrix.GeneralLinearGroup.det ((upperLeftBlock hab H)⁻¹ * g)))⁻¹ = (∏ c ∈ Finset.range (j + 1), χ (c + 1) (dBlk (blkOf r t) (c + 1) H))⁻¹ * χ (j + 1) (Matrix.GeneralLinearGroup.det H) * ((∏ c ∈ Finset.range j, χ (c + 1) (dBlk (blkOf r t) (c + 1) g)) * (χ (j + 1) (Matrix.GeneralLinearGroup.det g))⁻¹)

theorem **omegaG_of_top** (g : ∀ PP : ɪPg, GL (Fin a) (kP PP.1)) (hg : ∀ PP : ɪPg, IsTopB r t (y PP.1) (g PP)) (hm : (fun PP : ɪPg => upperLeftBlock ha (g PP)) ∈ P.H) : omegaG (Lev.datY a r t k ha Pg ⌀ P χ ξ Lev.Frm.B) χj y = ((P.chi ⟨_, hm⟩ * ∏ PP : ɪPg, ((∏ c ∈ Finset.range k, χ (c + 1) PP.1 (dBlk (blkOf r t) (c + 1) (g PP))) * (χj PP.1 (Matrix.GeneralLinearGroup.det (g PP)))⁻¹) : ℂˣ) : ℂ)

theorem **omegaG_of_notTop** (hg : ¬ ∃ g : ∀ PP : ɪPg, GL (Fin a) (kP PP.1), (∀ PP : ɪPg, IsTopB r t (y PP.1) (g PP)) ∧ (fun PP : ɪPg => upperLeftBlock ha (g PP)) ∈ P.H) : omegaG (Lev.datY a r t k ha Pg ⌀ P χ ξ Lev.Frm.B) χj y = 0

theorem **omegaG_levelP_move** : ∀ (D : D7) (a b r t k : ℕ) (ha : r ≤ a) (hb : r ≤ b), a = r + k * t → b = r + (k + 1) * t → ∀ (Pg : Finset (Pl D.host.E)) (P : LevelP Pg r ℂ) (χ : ℕ → ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (H : ∀ PP : Pl D.host.E, GL (Fin b) (kP PP)), (∀ PP ∈ Pg, H PP ∈ blockTri (blkOf r t)) → ∀ hP : (fun PP : ɪPg => upperLeftBlock hb (H PP.1)) ∈ P.H, ∀ y : ∀ PP : Pl D.host.E, Matrix (Fin b) (Fin a) (kP PP), omegaG (Lev.datY a r t k ha Pg ⌀ P χ ξ Lev.Frm.B) (χ (k + 1)) (fun PP => ((H PP)⁻¹).val * y PP) = (((P.chi ⟨_, hP⟩ * ∏ PP ∈ Pg, ∏ i ∈ Finset.range (k + 1), χ (i + 1) PP (dBlk (blkOf r t) (i + 1) (H PP)))⁻¹ * ∏ PP ∈ Pg, χ (k + 1) PP (Matrix.GeneralLinearGroup.det (H PP)) : ℂˣ) : ℂ) * omegaG (Lev.datY a r t k ha Pg ⌀ P χ ξ Lev.Frm.B) (χ (k + 1)) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesDatumInvLeftMulTargetLevelStatement

theorem **_root_.GlobalTheory.T.levelPlacesDatum_inv_left_mul_target_level** : ∀ (D : D7) (a b r t k : ℕ) (ha : r ≤ a) (hb : r ≤ b), a = r + k * t → b = r + (k + 1) * t → ∀ (Pg : Finset (Pl D.host.E)) (P : LevelP Pg r ℂ) (χ : ℕ → ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (H : ∀ PP : Pl D.host.E, GL (Fin b) (kP PP)), (∀ PP ∈ Pg, H PP ∈ blockTri (blkOf r t)) → ∀ hP : (fun PP : ɪPg => upperLeftBlock hb (H PP.1)) ∈ P.H, ∀ y : ∀ PP : Pl D.host.E, Matrix (Fin b) (Fin a) (kP PP), omegaG (Lev.datY a r t k ha Pg ⌀ P χ ξ Lev.Frm.B) (χ (k + 1)) (fun PP => ((H PP)⁻¹).val * y PP) = (((P.chi ⟨_, hP⟩ * ∏ PP ∈ Pg, ∏ i ∈ Finset.range (k + 1), χ (i + 1) PP (dBlk (blkOf r t) (i + 1) (H PP)))⁻¹ * ∏ PP ∈ Pg, χ (k + 1) PP (Matrix.GeneralLinearGroup.det (H PP)) : ℂˣ) : ℂ) * omegaG (Lev.datY a r t k ha Pg ⌀ P χ ξ Lev.Frm.B) (χ (k + 1)) y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesDatumMulRightSourceLevel

theorem **topL_mul** {κ : Type} [CommRing κ] {a b : ℕ} (y : Matrix (Fin b) (Fin a) κ) (γ : Matrix (Fin a) (Fin a) κ) : topL a (y * γ) = topL a y * γ

theorem **lowZero_mul** {κ : Type} [CommRing κ] {a b : ℕ} {y : Matrix (Fin b) (Fin a) κ} (hy : LowZero a y) (γ : Matrix (Fin a) (Fin a) κ) : LowZero a (y * γ)

theorem **cond_mul** {a b : ℕ} (L : LevDat D.host.E ℂ a) (γ : ∀ PP : Pl D.host.E, GL (Fin a) (kP PP)) (hγ : ∀ PP ∈ L.Pg, γ PP ∈ blockTri (blkOf L.r L.t)) (hP : (fun PP : ɪL.Pg => upperLeftBlock L.hr (γ PP.1)) ∈ L.P.H) (y z : ∀ PP : Pl D.host.E, Matrix (Fin b) (Fin a) (kP PP)) (hz : ∀ PP, z PP = y PP * (γ PP).val) (h : Cond L y) : Cond L z

theorem **omegaG_mul_right** : ∀ (D : D7) (a b : ℕ) (L : LevDat D.host.E ℂ a) (χ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (G : ∀ PP : Pl D.host.E, GL (Fin a) (kP PP)), (∀ PP ∈ L.Pg, G PP ∈ blockTri (blkOf L.r L.t)) → ∀ hP : (fun PP : ɪL.Pg => upperLeftBlock L.hr (G PP.1)) ∈ L.P.H, ∀ y : ∀ PP : Pl D.host.E, Matrix (Fin b) (Fin a) (kP PP), omegaG L χ (fun PP => y PP * (G PP).val) = ((L.P.chi ⟨_, hP⟩ * (∏ PP ∈ L.Pg, ∏ i ∈ Finset.range L.s, L.cg (i + 1) PP (dBlk (blkOf L.r L.t) (i + 1) (G PP))) * ∏ PP ∈ L.Pg, (χ PP (Matrix.GeneralLinearGroup.det (G PP)))⁻¹ : ℂˣ) : ℂ) * omegaG L χ y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesDatumMulRightSourceLevelStatement

theorem **_root_.GlobalTheory.T.levelPlacesDatum_mul_right_source_level** : ∀ (D : D7) (a b : ℕ) (L : LevDat D.host.E ℂ a) (χ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (G : ∀ PP : Pl D.host.E, GL (Fin a) (kP PP)), (∀ PP ∈ L.Pg, G PP ∈ blockTri (blkOf L.r L.t)) → ∀ hP : (fun PP : ɪL.Pg => upperLeftBlock L.hr (G PP.1)) ∈ L.P.H, ∀ y : ∀ PP : Pl D.host.E, Matrix (Fin b) (Fin a) (kP PP), omegaG L χ (fun PP => y PP * (G PP).val) = ((L.P.chi ⟨_, hP⟩ * (∏ PP ∈ L.Pg, ∏ i ∈ Finset.range L.s, L.cg (i + 1) PP (dBlk (blkOf L.r L.t) (i + 1) (G PP))) * ∏ PP ∈ L.Pg, (χ PP (Matrix.GeneralLinearGroup.det (G PP)))⁻¹ : ℂˣ) : ℂ) * omegaG L χ y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesDisjointIdempotents

theorem **pk_places** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → (∃ e : AE D, IsIdemSum e ((pk W).Pg ∪ ⌀)) ∧ ∀ t, Disjoint (pk W).Pg ((pk W).Q t) ∧ (∀ PP ∈ (pk W).Pg, ∀ PP' ∈ (pk W).Q t, PP.under (𝒪 D.host.F) ≠ PP'.under (𝒪 D.host.F)) ∧ (∃ e : AE D, IsIdemSum e ((pk W).Pg ∪ (pk W).Q t)) ∧ ∃ e : AE D, IsIdemSum e (⌀ ∪ (pk W).Q t)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesDisjointIdempotentsStatement

theorem **_root_.GlobalTheory.T.level_places_disjoint_idempotents** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → (∃ e : AE D, IsIdemSum e ((pk W).Pg ∪ ∅)) ∧ ∀ t, Disjoint (pk W).Pg ((pk W).Q t) ∧ (∀ PP ∈ (pk W).Pg, ∀ PP' ∈ (pk W).Q t, PP.under (𝒪 D.host.F) ≠ PP'.under (𝒪 D.host.F)) ∧ (∃ e : AE D, IsIdemSum e ((pk W).Pg ∪ (pk W).Q t)) ∧ ∃ e : AE D, IsIdemSum e (∅ ∪ (pk W).Q t)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesFinite

theorem **finite_levelPlaces_or_tuplePlaces** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → Set.Finite {PP : Pl D.host.E | PP ∈ (pk W).Pg ∨ QAll W PP}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesFiniteStatement

theorem **_root_.GlobalTheory.T.level_places_finite** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → Set.Finite {PP : Pl D.host.E | PP ∈ (pk W).Pg ∨ QAll W PP}

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesIntegralNonzeroConstant

theorem **continuous_wcoef** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {χS χT : (AE D)ˣ →* Circle} {om : WRep S T} (hom : IsWeilDatumO ψ S T χS χT om) {ν : Measure (WO S T)} {ℱ : Set (WO S T)} (hν : IsCovolOne (ratL T S) ν ℱ) {Φ₁ Φ₂ : WO S T → ℂ} (h₁ : IsThetaFunO ψ S T Φ₁) (h₂ : IsThetaFunO ψ S T Φ₂) : Continuous (wcoef om ν ℱ Φ₁ Φ₂)

theorem **continuous_rcoef** {θ : InfFam D} {A : HSp D} {cA : CptF A} (hA : HSpace.SigOK A) (hP : HSpace.IsPosDef A) {m : Measure (GA D A)} (hm : m.IsHaarMeasure) {u : GA D A → ℂ} (hu : u ∈ A0 θ A cA) : Continuous (rcoef m u u)

theorem **wcoef_mul_fix** {S T : HSp D} (om : WRep S T) (ν : Measure (WO S T)) (ℱ : Set (WO S T)) (Φ₁ Φ₂ : WO S T → ℂ) (x κ : GA D S) (h : om (κ, 1) Φ₁ = Φ₁) : wcoef om ν ℱ Φ₁ Φ₂ (x * κ) = wcoef om ν ℱ Φ₁ Φ₂ x

theorem **rcoef_mul_fix** {A : HSp D} (m : Measure (GA D A)) (u : GA D A → ℂ) (x κ : GA D A) (h : ∀ g, u (g * κ) = u g) : rcoef m u u (x * κ) = rcoef m u u x

theorem **integral_const** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A : HSp D) (cA : CptF A) (a : GL (Fin A.rank) (AE D)), Defin θ A cA a → ∀ (B W H : HSp D) (o : OrthDec B W H) (χA χW : (AE D)ˣ →* Circle) (omB : WRep A B) (omW : WRep A W) (omH : WRep A H), IsWeilDatumO ψ A B χA χW omB → IsWeilDatumO ψ A W χA χW omW → IsWeilDatumO ψ A H χA 1 omH → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∀ m : Measure (GA D A), m.IsHaarMeasure → ∀ u ∈ A0 θ A cA, (∀ κ ∈ prin A β Z, ∀ g, u (g * κ) = u g) → ∀ (sW : RealMat W) (Φ : WO A W → ℂ), Occ ψ θ sW omW m β Z u Φ → ∀ f, IsThetaFunO ψ A H f → f ≠ 0 → (∀ κ ∈ prin A β Z, omH (κ, 1) f = f) → (∀ b ∈ GinfC A, omH (b, 1) f = f) → (∀ ν ℱ, IsCovolOne (ratL H A) ν ℱ → ∀ g ∈ openOff A (frameLat A) (under Z), g ∉ prinInf A β Z → wcoef omH ν ℱ f f g = 0) → ∀ νB ℱB, IsCovolOne (rat

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesIntegralNonzeroConstantStatement

theorem **_root_.GlobalTheory.T.levelPlaces_integral_nonzero_constant** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (A : HSp D) (cA : CptF A) (a : GL (Fin A.rank) (AE D)), Defin θ A cA a → ∀ (B W H : HSp D) (o : OrthDec B W H) (χA χW : (AE D)ˣ →* Circle) (omB : WRep A B) (omW : WRep A W) (omH : WRep A H), IsWeilDatumO ψ A B χA χW omB → IsWeilDatumO ψ A W χA χW omW → IsWeilDatumO ψ A H χA 1 omH → ∀ (β : GL (Fin A.rank) (Formalism.IndefR.Lev.EAf D)) (Z : Finset (Formalism.L2.DIV.Pl D.host.E)), Places A β Z → ∀ m : Measure (GA D A), m.IsHaarMeasure → ∀ u ∈ A0 θ A cA, (∀ κ ∈ prin A β Z, ∀ g, u (g * κ) = u g) → ∀ (sW : RealMat W) (Φ : WO A W → ℂ), Occ ψ θ sW omW m β Z u Φ → ∀ f, IsThetaFunO ψ A H f → f ≠ 0 → (∀ κ ∈ prin A β Z, omH (κ, 1) f = f) → (∀ b ∈ GinfC A, omH (b, 1) f = f) → (∀ ν ℱ, IsCovolOne (ratL H A) ν ℱ → ∀ g ∈ openOff A (frameLat A) (under Z), g ∉ prinInf A β Z → wcoef

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesUnderEq

theorem **natCast_mem_of_under_eq** {PP PP' : Pl D.host.E} (h : PP.under (𝒪 D.host.F) = PP'.under (𝒪 D.host.F)) {n : ℕ} (hn : ((n : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) : ((n : ℕ) : 𝒪 D.host.E) ∈ PP'.asIdeal

theorem **not_two_primes_mem** (PP : Pl D.host.E) {p q : ℕ} (hp : p.Prime) (hq : q.Prime) (hne : p ≠ q) (h1 : ((p : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) (h2 : ((q : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal) : False

theorem **twist_prime** {gp : Fin 2 → ℕ} {Q : Finset (Pl D.host.E)} (hQ : IsTwistPlaces gp Q) {PP : Pl D.host.E} (hP : PP ∈ Q) : ∃ q : ℕ, q.Prime ∧ ((q : ℕ) : 𝒪 D.host.E) ∈ PP.asIdeal ∧ ∀ i, q ≠ gp i

theorem **g_mem_of_div** (P : Formalism.L2.Host.PiBar D.host) {k : Fin 2} {w : Pl D.host.E} (hw : P.Div k w) : ((D.host.g k : ℕ) : 𝒪 D.host.E) ∈ w.asIdeal

theorem **GlobalTheory.Fam.pg_qAll_under_inj** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ PP PP' : Pl D.host.E, (PP ∈ (pk W).Pg ∨ QAll W PP) → (PP' ∈ (pk W).Pg ∨ QAll W PP') → PP.under (𝒪 D.host.F) = PP'.under (𝒪 D.host.F) → PP = PP'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesUnderEqStatement

theorem **_root_.GlobalTheory.T.level_places_under_eq** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ PP PP' : Pl D.host.E, (PP ∈ (pk W).Pg ∨ QAll W PP) → (PP' ∈ (pk W).Pg ∨ QAll W PP') → PP.under (𝒪 D.host.F) = PP'.under (𝒪 D.host.F) → PP = PP'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesUnderInjective

theorem **under_injective** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → Function.Injective (fun i : {PP // PP ∈ (pk W).Pg ∨ QAll W PP} => HeightOneSpectrum.under (𝒪 D.host.F) i.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPlacesUnderInjectiveStatement

theorem **_root_.GlobalTheory.T.level_places_under_injective** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX :

ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → Function.Injective (fun i : {PP // PP ∈ (pk W).Pg ∨ QAll W PP} => HeightOneSpectrum.under (𝒪 D.host.F) i.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeLawLastTwoStepsAllTuples

theorem **combAt_fin** {D : D7} {P : Pl D → Prop} {S T : HSp D} {c : WO S T →₀ ℂ} (h : CombAt P c) : IsFinComb c

theorem **main** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι), c.Is → v.Is → ShapeOff v o → ∀ j (h : j + 2 * 0 + 2 = D.R.m), ∃ (ε : ι → 𝓀[k]ˣ) (op : BiOp (c.X j) (c.X (j + 1)) (c.X (j + 1 + 1))) (r : ι → ℝ) (e : ι → ℤ), (∀ i, 0 < r i) ∧ OpAt (· ∈ v.tP) op ∧ ∀ t : Tup c, t.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeLawLastTwoStepsAllTuplesStatement

theorem **_root_.GlobalTheory.T.levelPrime_law_last_two_steps_all_tuples** : ∀ (D : D7) (c : Comm D) (v : VCom c) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι), c.Is → v.Is → ShapeOff v o → ∀ j (h : j + 2 * 0 + 2 = D.R.m), ∃ (ε : ι → 𝓀[k]ˣ) (op : BiOp (c.X j) (c.X (j + 1)) (c.X (j + 1 + 1))) (r : ι → ℝ) (e : ι → ℤ), (∀ i, 0 < r i) ∧ OpAt (· ∈ v.tP) op ∧ ∀ t : Tup c, t.Is → WeilOK (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeNotMemPlaces

theorem **levelPrime_not_mem_places** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → (∀ PP ∈ (pk W).Pg, (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∉ (PP.under (𝒪 D.host.F)).asIdeal) → ∀ PP : Pl D.host.E, PP ∈ (pk W).Pg ∨ QAll W PP → (((D.mem u).A.T : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeOperatorsActOnRamifiedColumns

theorem **ramAt_fixed** {p₀ : SplitPt D k} {β : Frame p₀ S n} {P : Pair k n} {φ₀ : GA D S → ℂ} (h : GlobalTheory.FinPl.RamAt β P φ₀) {g : GL n k} (hg : g • P = P) : rTrans (β.gl g) φ₀ = φ₀

theorem **comb_foldr** (β : ∀ i, Frame (p i) S n) (Q : ι → Pair k n) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (η : ι → 𝓀[k]ˣ) (J : List ι) : ∃ c : GA D S →₀ ℂ, (∀ g ∈ c.support, g ∈ grpAt S fun x => ∃ i, x = (p i).v) ∧ ∀ φ, J.foldr (fun i f => sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) (η i) f) φ = trc c φ

theorem **sOp_keeps** (hinj : Function.Injective fun i => (p i).v) (β : ∀ i, Frame (p i) S n) (Q : ι → Pair k n) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (ε : 𝓀[k]ˣ) (i : ι) {φ : GA D S → ℂ} (hφ : IsContAut S φ ∧ ∀ i', GlobalTheory.FinPl.RamAt (β i') (Q i') φ) : IsContAut S (sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) ε φ) ∧ ∀ i', GlobalTheory.FinPl.RamAt (β i') (Q i') (sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) ε φ)

theorem **foldr_keeps** (hinj : Function.Injective fun i => (p i).v) (β : ∀ i, Frame (p i) S n) (Q : ι → Pair k n) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (ε : ι → 𝓀[k]ˣ) (l : List ι) {φ : GA D S → ℂ} (hφ : IsContAut S φ ∧ ∀ i', GlobalTheory.FinPl.RamAt (β i') (Q i') φ) : IsContAut S (l.foldr (fun i φ' => sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) (ε i) φ') φ) ∧ ∀ i', GlobalTheory.FinPl.RamAt (β i') (Q i') (l.foldr (fun i φ' => sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) (ε i) φ') φ)

theorem **twS** : TwS'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeOperatorsActOnRamifiedColumnsStatement

theorem **_root_.GlobalTheory.T.levelPrime_operators_act_on_ramified_columns** : GlobalTheory.AdForm.TwS'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeOperatorsActRamifiedColumnsVanishing

theorem **sOp_zero_fun** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {p : SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : Frame p S n) (P : Pair k n) (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (ε : 𝓀[k]ˣ) : sOp β P q χ ε 0 = 0

theorem **foldr_sOp_zero_fun** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} {p : ι → SplitPt D k} {S : HSp D} {n : Type} [Fintype n] [DecidableEq n] (β : ∀ i, Frame (p i) S n) (P : ι → Pair k n) (q : Unif k) (χ : ι → AddChar 𝓀[k] ℂ) (η : ι → 𝓀[k]ˣ) (J : List ι) : J.foldr (fun i f => sOp (β i) (P i) q (χ i) (η i) f) 0 = 0

theorem **infM_one_eq** {D : D7} (θ : InfFam D) (V : HSp D) {n : ℕ} (hn : V.rank = n) (τ : D.what₂.Tau) : infM θ V hn 1 τ = 1

theorem **TwDat.rd_eq_zero_of_vanish** (A : TwDat I l Z u a Ch) (θ : InfFam (I.rec7 l Z)) (j : ℕ) (hj : j ≤ B.R.m) (b : Ch j) (φ : ColT I l u a j) (hφ : ∀ c, A.chOf j c = b → φ c = 0) (y : (I.chain l u a).Y j) : A.rd θ j hj b φ y = 0

theorem **twS_ramL** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch₀ Ch : ℕ → Type) (A : TwDat I l Z u a Ch₀) (ιc : ∀ j, Ch j → Ch₀ j), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, GlobalTheory.FinPl.Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ (J : List ι) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1), ∃ S : ∀ j, (ι → 𝓀[k]ˣ) → ColT I l u a j → ColT I l u a j, ∀ j (hj : j ≤ B.R.m) φ, A.RamL ιc j φ → (∀ b y, IsContAut _ (A.rd θ j hj b φ y) ∧ ∀ i, GlobalTheory.FinPl.RamAt (β j i) (Q j i) (A.rd θ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeOperatorsActRamifiedColumnsVanishingStatement

theorem **_root_.GlobalTheory.T.levelPrimeOperators_act_ramifiedColumns_vanishing** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)) (u : B.Word) (a : I.ι) (Ch₀ Ch : ℕ → Type) (A : TwDat I l Z u a Ch₀) (ιc : ∀ j, Ch j → Ch₀ j), A.Is → ∀ θ, IsInfFam θ → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt (I.rec7 l Z) k) (β : ∀ j i, GlobalTheory.FinPl.Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)) (Q : ∀ j, ι → Pair k (Fin ((I.rec7 l Z).sp.X j).rank)) (L : ∀ j, Lat ((I.rec7 l Z).sp.X j)), StdT I l Z u p β Q L → ∀ (J : List ι) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1), ∃ S : ∀ j, (ι → 𝓀[k]ˣ) → ColT I l u a j → ColT I l u a j, ∀ j (hj : j ≤ B.R.m) φ, A.RamL ιc j φ → (∀ b y, IsContAut _ (A.rd θ j hj

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeOperatorsStepZeroScaleNorms

theorem **isLinearMap_of_inner** {E : Type*} [NormedAddCommGroup E] [InnerProductSpace ℂ E] (f : E → E) (Q : ℝ) (h : ∀ x y, inner ℂ (f x) (f y) = (Q : ℂ) * inner ℂ x y) : IsLinearMap ℂ f

theorem **norm_of_inner** {E : Type*} [NormedAddCommGroup E] [InnerProductSpace ℂ E] (f : E → E) (Q : ℝ) (hQ : 0 ≤ Q) (h : ∀ x y, inner ℂ (f x) (f y) = (Q : ℂ) * inner ℂ x y) (x : E) : ‖f x‖ = Real.sqrt Q * ‖x‖

theorem **comm_of_sub_one_mul** {R : Type*} [Ring R] (M N : R) (h1 : (M - 1) * (N - 1) = 0) (h2 : (N - 1) * (M - 1) = 0) : M * N = N * M

theorem **placeIdem_mul** {D : D7} {v v' : HeightOneSpectrum (𝒪 D.host.F)} {ε ε' : AF D.host.F} (h : IsPlaceIdem D v ε) (h' : IsPlaceIdem D v' ε') (hv : v ≠ v') : ε * ε' = 0

theorem **gl_comm_other** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p p' : SplitPt D k) (S : HSp D) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (β : Frame p S n) (β' : Frame p' S n') (g : GL n k) (h : GL n' k), p.v ≠ p'.v → β.gl g * β'.gl h = β'.gl h * β.gl g

theorem **rep_comm** (β : Frame p S n) (β' : Frame p' S n') (hv : p.v ≠ p'.v) (g : GL n k) (h : GL n' k) (f : GA D S → ℂ) : β.rep g (β'.rep h f) = β'.rep h (β.rep g f)

theorem **sOp_ramAt_other'** (β : Frame p S n) (β' : Frame p' S n') (P : Pair k n) (P' : Pair k n') (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (ε : 𝓀[k]ˣ) (φ₀ : GA D S → ℂ) (hv : p.v ≠ p'.v) (h : RamAt β' P' φ₀) : RamAt β' P' (sOp β P q χ ε φ₀)

theorem **normStepZero** : ∀ (D : D7) (u : D.what₂.Word) (F : Formalism.L2.Forms.Forms0 D u) (θ : InfFam D), IsInfFam θ → CoCpt (D.sp.X 0) → ∀ (vS : Vol (D.sp.X 0)) (y : (D.mem u).C.Y 0), ∃ c : ℝ, 0 < c ∧ ∀ φ ψ : D.what₂.Mu u, ∫ g in vS.𝓕, (GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u φ).1 y g * star ((GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u ψ).1 y g) ∂vS.μ = (c : ℂ) * inner ℂ ψ φ

theorem **sOp_ramAt_other** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p p' : SplitPt D k) (S : HSp D) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (β : Frame p S n) (β' : Frame p' S n') (P : Pair k n) (P' : Pair k n') (q : Unif k) (χ : AddChar 𝓀[k] ℂ) (ε : 𝓀[k]ˣ) (φ₀ : GA D S → ℂ), p.v ≠ p'.v → RamAt β' P' φ₀ → RamAt β' P' (sOp β P q χ ε φ₀)

theorem **ramAt_foldr** (i : ι) (f : GA D S → ℂ) : ∀ I : List ι, (∀ j ∈ I, (p j).v ≠ (p i).v) → RamAt (β i) (Q i) f → RamAt (β i) (Q i) (I.foldr (fun j g => sOp (β j) (Q j) q (χ' j) (η j) g) f) | [], _, h => h | j :: I, hI, h => by simp only [List.foldr_cons] exact sOp_ramAt_other D k (p j) (p i) S n n (β j) (β i) (Q j) (Q i) q (χ' j) (η j) _ (hI j (by simp)) (ramAt_foldr i f I (fun j' hj' => hI j' (by simp [hj'])) h) theorem inner_foldr (hRU : RamUnitary D) (hS : CoCpt S) (vS : Vol S) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (hinj : Function.Injective fun i => (p i).v) : ∀ I : List ι, I.Nodup → ∀ f f' : GA D S → ℂ, (∀ i ∈ I, RamAt (β i) (Q i) f) → (∀ i ∈ I, RamAt (β i) (Q i) f') → ∫ g in vS.𝓕, I.foldr (fun j h => sOp (β j) (Q j) q (resChar (χ j) (hχ j) q) (η j) h) f g * star (I.foldr (fun j h => sOp (β j) (Q j) q (resChar (χ j) (hχ j) q) (η j) h) f' g) ∂vS.μ = (resCard k

theorem **opsStepZero_norm** : ∀ (D : D7) (u : D.what₂.Word) (F : Formalism.L2.Forms.Forms0 D u) (θ : InfFam D), IsInfFam θ → RamUnitary D → CoCpt (D.sp.X 0) → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (β : ∀ i, Frame (p i) (D.sp.X 0) (Fin (D.sp.X 0).rank)) (Q : ι → Pair k (Fin (D.sp.X 0).rank)), I.Nodup → (Function.Injective fun i => (p i).v) → (∀ (φ : D.what₂.Mu u) y i, RamAt (β i) (Q i) ((GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u φ).1 y)) → ∀ η : ι → 𝓀[k]ˣ, (∀ φ : D.what₂.Mu u, ∃ φ' : D.what₂.Mu u, ∀ y, (GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u φ').1 y = I.foldr (fun i f => sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) (η i) f) ((GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u φ).1 y)) → ∃ (S : D.what₂.Mu u ≃ₗ[ℂ] D.what₂.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeOperatorsStepZeroScaleNormsStatement

theorem **_root_.GlobalTheory.T.levelPrime_operators_stepZero_scale_norms** : ∀ (D : D7) (u : D.what₂.Word) (F : Formalism.L2.Forms.Forms0 D u) (θ : InfFam D), IsInfFam θ → RamUnitary D → CoCpt (D.sp.X 0) → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (q : Unif k) (χ : ι → AddChar k Circle) (hχ : ∀ i, HasConductor (χ i) 1) (β : ∀ i, Frame (p i) (D.sp.X 0) (Fin (D.sp.X 0).rank)) (Q : ι → Pair k (Fin (D.sp.X 0).rank)), I.Nodup → (Function.Injective fun i => (p i).v) → (∀ (φ : D.what₂.Mu u) y i, RamAt (β i) (Q i) ((GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u φ).1 y)) → ∀ η : ι → 𝓀[k]ˣ, (∀ φ : D.what₂.Mu u, ∃ φ' : D.what₂.Mu u, ∀ y, (GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu u φ').1 y = I.foldr (fun i f => sOp (β i) (Q i) q (resChar (χ i) (hχ i) q) (η i) f) ((GlobalTheory.Core.Mi.chart0 F).rd θ (D.eMu

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelPrimeWindow

theorem **levelPrime_window** : ∀ (D : D7) (u : D.what₂.Word), (RankMu D u → (d0 D u : ℝ) ≤ Real.exp 1 / (D.mem u).A.GamP ^ 2 * D.what₂.Dstar) ∧ (Real.log (D.mem u).A.Xu - 2 * Real.log (D.mem u).A.GamP / ((D.R.f : ℝ) * D.R.k) ≤ Real.log ((D.mem u).A.T : ℝ)) ∧ (∀ L C : ℝ, RankMu D u → 1 + 2 * C * mfL D L ≤ 2 * Real.log (D.mem u).A.GamP → Small D u L C)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelStepVectorEqHeisenbergCombinationVacuum

theorem **idem_mem_idAt** {es : AE D} {PP : Pl D.host.E} (h : Formalism.L2.NTheta.IsIdemAt es PP) : es ∈ GlobalTheory.FinPl.idAt D (· = PP.under (𝒪 D.host.F))

theorem **idemSum_mem_idAt** {e : AE D} {Z : Finset (Pl D.host.E)} (h : IsIdemSum e Z) : e ∈ GlobalTheory.FinPl.idAt D (fun p => ∃ PP ∈ Z, p = PP.under (𝒪 D.host.F))

theorem **mxAt_of_mem_half** {S T : HSp D} {a : AE D} {P : GlobalTheory.FinPl.Pl D → Prop} (ha : a ∈ GlobalTheory.FinPl.idAt D P) {u : WO S T} (hu : u ∈ GlobalTheory.PlaceModel.half S T a) : GlobalTheory.FinPl.MxAt P u

theorem **isSBHalf_stepDatumQ** (S T : HSp D) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (L : LevDat D.host.E ℂ S.rank) (χ ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (Q : Finset (Pl D.host.E)) (e : AE D) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) (he : IsIdemSum e (L.Pg ∪ Q)) : IsSBHalf e (latO S T ΛS ΛT fun _ => True) (fun x : WO S T => if IsHalf e x ∧ IsIntM (frMapO S T βS βT x) then omegaG L χ (fun PP => redM PP (frMapO S T βS βT x)) * omegaQ Q L.r ξ 1 (fun PP => redM PP (frMapO S T βS βT x)) else 0)

theorem **comb_add'** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (c c' : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (c + c') f = comb ψ S T c f + comb ψ S T c' f

theorem **comb_smul'** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (a : ℂ) (c : WO S T →₀ ℂ) (f : WO S T → ℂ) : comb ψ S T (a • c) f = a • comb ψ S T c f

theorem **comb_zero'** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (f : WO S T → ℂ) : comb ψ S T 0 f = 0

theorem **comb_single'** (ψ : AddChar (AF D.host.F) ℂ) {S T : HSp D} (u : WO S T) (a : ℂ) (f : WO S T → ℂ) : comb ψ S T (Finsupp.single u a) f = a • heisO ψ S T u f

theorem **mem_of_mem_support_add** {S T : HSp D} {H : Set (WO S T)} {c c' : WO S T →₀ ℂ} (hc : ∀ u ∈ c.support, u ∈ H) (hc' : ∀ u ∈ c'.support, u ∈ H) : ∀ u ∈ (c + c').support, u ∈ H

theorem **mem_of_mem_support_smul** {S T : HSp D} {H : Set (WO S T)} (a : ℂ) {c : WO S T →₀ ℂ} (hc : ∀ u ∈ c.support, u ∈ H) : ∀ u ∈ (a • c).support, u ∈ H

theorem **exists_comb_of_isThetaAt** {S T : HSp D} (ψ : AddChar (AF D.host.F) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (e : AE D) (f₀ Φ f : WO S T → ℂ) (hψ : IsStdChar D ψ (-1)) (hS : IsSelfDualC S ΛS) (hT : IsSelfDualC T ΛT) (he : e * e = e) (hec : e * cA D e = 0) (hinf : infPart D e = 0) (hΦ : IsSBHalf e (latO S T (ΛS : Set (Fin

S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) Φ) (hθ : IsThetaAt ψ (symplO S T) (heisO ψ S T) e (latO S T (ΛS : Set (Fin S.rank → D.host.E)) (ΛT : Set (Fin T.rank → D.host.E)) fun _ => True) 1 f₀ Φ f) : ∃ d : WO S T →₀ ℂ, (∀ u ∈ d.support, u ∈ half S T (e + cA D e)) ∧ f = comb ψ S T d f₀

theorem **exists_comb_of_isStepVecOf** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (LS : LevDat D.host.E ℂ S.rank) (χj : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (Q : Finset (Pl D.host.E)) (ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (f₀ f : WO S T → ℂ), IsSelfDualC S ΛS → IsSelfDualC T ΛT → Formalism.L2.DIV.IsFrame βS ΛS → Formalism.L2.DIV.IsFrame βT ΛT → IsStepVecOf ψ θ S T sS sT ΛS ΛT βS βT LS χj Q ξ f₀ f → ∃ d : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt (fun p => ∃ PP ∈ LS.Pg ∪ Q, p = PP.under (𝒪 D.host.F)) d ∧ f = comb ψ S T d f₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelStepVectorEqHeisenbergCombinationVacuumStatement

theorem **_root_.GlobalTheory.T.levelStepVector_eq_heisenbergCombination_vacuum** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (βS : GL (Fin S.rank) (Lev.EAf D)) (βT : GL (Fin T.rank) (Lev.EAf D)) (LS : LevDat D.host.E ℂ S.rank) (χj : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (Q : Finset (Pl D.host.E)) (ξ : ∀ PP : Pl D.host.E, (kP PP)ˣ →* ℂˣ) (f₀ f : WO S T → ℂ), IsSelfDualC S ΛS → IsSelfDualC T ΛT → Formalism.L2.DIV.IsFrame βS ΛS → Formalism.L2.DIV.IsFrame βT ΛT → IsStepVecOf ψ θ S T sS sT ΛS ΛT βS βT LS χj Q ξ f₀ f → ∃ d : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt (fun p => ∃ PP ∈ LS.Pg ∪ Q, p = PP.under (𝒪 D.host.F)) d ∧ f = comb ψ S T d f₀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelTraceAdaptedTranslate

theorem **col_level_match_fam** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), LevTrace f B → B.Adapted (W.S.tr u).wt → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∀ x ∈ GfC (D.sp.X 0), ((show GA D (D.sp.Y 0) from f.g.b 0)⁻¹ * embA B.d B.o x * (show GA D (D.sp.Y 0) from f.g.b 0) ∈ rJ W ↔ ((iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u))⁻¹ * embA B.d B.o (a₀ * x * a₀⁻¹) * ((iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u)) ∈ rowJ W.S.X W.Q.instEq) ∧ ((show GA D (D.sp.Y 0) from f.g.b 0)⁻¹ * embA B.d B.o x * (show GA D (D.sp.Y 0) from f.g.b 0) ∈ rJ W → rowChi W.S.X W.Q.instEq (((iot

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelTraceAdaptedTranslateStatement

theorem **_root_.GlobalTheory.T.level_trace_adapted_translate** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), PkHyp W → ∀ (u : D.what₂.Word) (f : FamCh W u) (B : SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)), LevTrace f B → B.Adapted (W.S.tr u).wt → ∀ a₀ : GA D (D.sp.X 0), IsTransportAdelicCorrection (W.S.tr u) a₀ → ∀ x ∈ GfC (D.sp.X 0), ((show GA D (D.sp.Y 0) from f.g.b 0)⁻¹ * embA B.d B.o x * (show GA D (D.sp.Y 0) from f.g.b 0) ∈ rJ W ↔ ((iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u))⁻¹ * embA B.d B.o (a₀ * x * a₀⁻¹) * ((iotaX W.S.X W.Q.instEq).symm (1, W.S.X.xu u)) ∈ rowJ W.S.X W.Q.instEq) ∧ ((show GA D (D.sp.Y 0) from f.g.b 0)⁻¹ * embA B.d B.o x * (show GA D (D.sp.Y 0) from f.g.b 0) ∈ rJ W

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelTrueOfWitness

theorem **lev_of_btw** {D : D7} {O : Obj D} {c : ℝ} (b : GlobalTheory.Fc.Bundle D O c) (x : R3.TwIn O.ex) (y : Bl.BTw b x) : (pk (packB b)).Pg = x.R.Pg ∧ ∀ t, (pk (packB b)).xiH t = x.R.toLevCh.xiH (x.Y.tupT ((packB b).P.I.tup t))

theorem **lev_of_args** {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) (x : R3.TwIn q.O.ex) (a : R3.TwArgs q.O.ex q.x.what₂) (ax : HEq a.x x) : x.R.Pg = (q.B.primes q.w.px).vGs ∧ ∀ μ, Formalism.IndefR.RecTie.Used0 (q.w.D q.Z) μ → x.R.toLevCh.xiH (x.Y.tupT μ) = q.w.I.xi μ

theorem **levTrue_packB** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → LevTrue (packB b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelTrueOfWitnessStatement

theorem **_root_.GlobalTheory.T.level_true_of_witness** : ∀ {hL : GlobalTheory.Published.LocalLanglandsGL} {p : Bl.Prm hL} {K : Ctx} (q : Bl.WPt p.s K) {c : ℝ} (b : Bundle (q.w.D q.Z) q.O c), Bl.BWit q b → LevTrue (packB b)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelZeroElementReduction

theorem **map_conj_local** {D : D7} {u : D.what₂.Word} {d : GlobalTheory.SeeSaw.Dat D u} {τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E} {PP : Pl D.host.E} (G : GlobalTheory.LocMod.GAt d τ PP) {π' : Lev.EAf D →+* G.k} (hπ : ∀ a, π' (toEAf D a) = G.q.pr a) {n : GA D (d.st 0).V} {q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin (G.s + 1)) G.k} (hloc : (GlobalTheory.LocMod.ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) (Nat.zero_add _)) n q) (σ : Fin (d.st 0).V.rank → GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin (G.s + 1)) (hσ : Function.Bijective σ) (hM : GlobalTheory.LocMod.MovedTo (GlobalTheory.LocMod.ofSplit G.q) (d.st 0).V (d.t 0) (d.st 0).β (((G.FY 0 (G.s + 1) (Nat.zero_add _)).β * ((G.S.T (Fin D.R.t)).bas 0 (G.s + 1) (Nat.zero_add _)).val).submatrix id σ)) : (((d.st 0).β⁻¹ * finE (d.st 0).V ((d.t 0)⁻¹ * n * d.t 0) * (d.st 0).β).val).map π' = ((

theorem **edge_sum_typeFrame** {K : Type} [Field K] (κ τ s N r p : ℕ) (hr : r ≤ N) (hrk : r = κ + (s + 1)) (e : GlobalTheory.StepData.Ix (Fin κ ⊕ Fin (s + 1)) (Fin τ) 0 ≃ Fin N) (he : GlobalTheory.SymBridge.IsOrd GlobalTheory.SymBridge.zPos e) (ϖ : Kˣ) (q : GL (GlobalTheory.StepData.XIx (Fin κ) (Fin τ) 0 ⊕ Fin (s + 1)) K) (hq : GlobalTheory.RootEx.InN p q) : (∑ i : Fin r, ∑ j : Fin r, if GlobalTheory.chainEdge κ i j then (((GlobalTheory.StepData.typeFrame ϖ (GlobalTheory.StepData.stdOrd (Fin κ) (Fin τ) 0 (s + 1) (Nat.zero_add _))).bas⁻¹).val * q.val * (GlobalTheory.StepData.typeFrame ϖ (GlobalTheory.StepData.stdOrd (Fin κ) (Fin τ) 0 (s + 1) (Nat.zero_add _))).bas.val) (GlobalTheory.StepData.stdOrd (Fin κ) (Fin τ) 0 (s + 1) (Nat.zero_add _)) (e.symm (Fin.castLE hr i))) (GlobalTheory.StepData.stdOrd (Fin κ) (Fin τ) 0 (s + 1) (Nat.zero_add _) (e.symm (Fin.castLE hr j))) else 0) = (ϖ : K) * Globa

theorem **psi_red_at** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (hr : D.R.r ≤ (d.st 0).V.rank) (n : GA D (d.st 0).V) (q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin (G.s + 1)) G.k) (p : ℕ), q ∈ GlobalTheory.RootEx.N G.k _ (G.s + 1) p → (GlobalTheory.LocMod.ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) (Nat.zero_add _)) n q → finE (d.st 0).V ((d.t 0)⁻¹ * n * d.t 0) ∈ St (d.st 0).β → redFn (d.st 0).β PP (finE (d.st 0).V ((d.t 0)⁻¹ * n * d.t 0)) ∈ blockTri (blk2 D.R.r) → ∀ z : 𝒪 D.host.E, Ideal.Quotient.mk PP.asIdeal z = edgeSum (GlobalTheory.chainEdge D.R.k) (upperLeftBlock hr (redFn (d.st 0).β PP (finE (d.st 0).V ((d.t 0)⁻¹ * n * d.t 0)))) → (G.S.ψ (((G.S.ϖ⁻¹ : G.kˣ) : G.k) * (GlobalTheory.LocMod.ofSplit G.q).pr (z : D.host.E)) : ℂ)

theorem **redFn_conj_eq_one** {D : D7} {V : HSp D} {β : GL (Fin V.rank) (Lev.EAf D)} {a e : AE D} {P : Finset (Pl D.host.E)} {PP Q : Pl D.host.E} (ha : IsIdemSum a P) (hPP : PP ∈ P) (he : Formalism.L2.NTheta.IsIdemAt e PP) {n : GA D V} (hn : SuppAt e V n) (t : GA D V) (hS : finE V (t⁻¹ * n * t) ∈ St β) (hQ : Q ∈ P) (hne : PP ≠ Q) : redFn β Q (finE V (t⁻¹ * n * t)) = 1

theorem **levChi_eq_psiBes** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), d.Base → IsTypePlaces d.c.Pg → GlobalTheory.LocMod.PinGBy G cG → ∀ (PP : d.c.Pg) (n : GA D (d.st 0).V) (q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (GlobalTheory.LocMod.ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → (d.t 0)⁻¹ * n * d.t 0 ∈ d.lev 0 → levChi (d.st 0).V (d.st 0).β (d.st 0).L ((d.t 0)⁻¹ * n * d.t 0) = psiBes d.c.ψ (d.bes 0 D.R.rS h0).d 1 (vecB (d.bes 0 D.R.rS h0).o (d.bes 0 D.R.rS h0).ζ) n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelZeroElementReductionStatement

theorem **_root_.GlobalTheory.T.level_zero_element_reduction** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (G : ∀ PP : d.c.Pg, GlobalTheory.LocMod.GAt d τ PP.1) (cG : ∀ j, WO (D.sp.X j) (D.sp.X (j + 1)) →₀ ℂ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), d.Base → IsTypePlaces d.c.Pg → GlobalTheory.LocMod.PinGBy G cG → ∀ (PP : d.c.Pg) (n : GA D (d.st 0).V) (q : GL (GlobalTheory.StepData.XIx (Fin D.R.k) (Fin D.R.t) 0 ⊕ Fin ((G PP).s + 1)) (G PP).k), q ∈ GlobalTheory.RootEx.N (G PP).k _ ((G PP).s + 1) (D.R.rS + 1) → (GlobalTheory.LocMod.ofSplit (G PP).q).IsLoc ((G PP).FY 0 ((G PP).s + 1) (Nat.zero_add _)) n q → (d.t 0)⁻¹ * n * d.t 0 ∈ d.lev 0 → levChi (d.st 0).V (d.st 0).β (d.st 0).L ((d.t 0)⁻¹ * n * d.t 0) = psiBes d.c.ψ (d.bes 0 D.R.rS h0).d 1 (vecB (d.bes 0 D.R.rS h0).o (d.bes 0 D.R.rS h0).ζ) n

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelZeroTypeFrameLocality

theorem **mat_read** {k : Type} [CommRing k] {N Y : Type} [Fintype N] [DecidableEq N] [Fintype Y] [DecidableEq Y] (σ : N ≃ Y) (β' βi' W Mi M0 : Matrix N N k) (Fb : Matrix N Y k) (Fg : Matrix Y N k) (bs bi X : Matrix Y Y k) {Yy G : Matrix _ _ k} (h1 : β' * W * βi' = Mi * Yy * M0) (hY : Yy = Fb * G * Fg) (hG : G = bs * X * bi) (h2 : βi' * β' = 1) (h3 : M0 * Mi = 1) (h4 : Fg * Fb = 1) (h5 : bi * bs = 1) (h6 : β' = (Mi * Fb * bs).submatrix id σ) : W = X.submatrix σ σ

theorem **map_mt_conj** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.LocMod.Place D k) (V : HSp D) (t y : GA D V) : (Formalism.L2.Theta.mt V (t⁻¹ * y * t)).map p.π = (Formalism.L2.Theta.mt V t⁻¹).map p.π * (Formalism.L2.Theta.mt V y).map p.π * (Formalism.L2.Theta.mt V t).map p.π

theorem **map_mt_inv** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : GlobalTheory.LocMod.Place D k) (V : HSp D) (t : GA D V) : (Formalism.L2.Theta.mt V t).map p.π * (Formalism.L2.Theta.mt V t⁻¹).map p.π = 1

theorem **map_inv_mul** {R k : Type} [CommRing R] [CommRing k] {n : ℕ} (f : R →+* k) (β : GL (Fin n) R) : (β⁻¹).val.map f * β.val.map f = 1

theorem **bc_apply** {A B : Type} [CommRing A] [CommRing B] {n : ℕ} (f : A →+* B) (g : GL (Fin n) A) (i j : Fin n) : (Formalism.UG.bc f g).val i j = f (g.val i j)

theorem **intGL_apply** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] {n : ℕ} (w : GL (Fin n) (Formalism.L2.Herm.intAf F E)) (i j : Fin n) : (intGL F E n w).val i j = (w.val i j : E ⊗[F] Formalism.L2.Herm.Af F)

theorem **GlobalTheory.ReadAtBase.red_of_isLoc** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (hr : D.R.r ≤ (d.st 0).V.rank) (hc : 0 + (G.s + 1) = G.s + 1) (x : G.S.P.J) (y : GA D (d.st 0).V), (GlobalTheory.LocMod.ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) hc) y ((GlobalTheory.StepData.typeFrame G.S.ϖ (GlobalTheory.StepData.stdOrd (Fin D.R.k) (Fin D.R.t) 0 (G.s + 1) hc)).conj (GlobalTheory.StepData.reGL (GlobalTheory.StepData.ixZero (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t)) (x : GL (Fin D.R.k ⊕ Fin (G.s + 1)) G.k))) → Formalism.IndefR.Lev.finE (d.st 0).V ((d.t 0)⁻¹ * y * d.t 0) ∈ Formalism.L2.DIV.St (d.st 0).β → G.Red x (Formalism.L2.DIV.upperLeftBlock hr (Formalism.L2.DIV.redFn (d.st 0).β PP (Formalism.IndefR.Lev.finE (d.st 0).V ((d.t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelZeroTypeFrameLocalityStatement

theorem **_root_.GlobalTheory.T.level_zero_type_frame_locality** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Formalism.L2.DIV.Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP) (hr : D.R.r ≤ (d.st 0).V.rank) (hc : 0 + (G.s + 1) = G.s + 1) (x : G.S.P.J) (y : GA D (d.st 0).V), (GlobalTheory.LocMod.ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) hc) y ((GlobalTheory.StepData.typeFrame G.S.ϖ (GlobalTheory.StepData.stdOrd (Fin D.R.k) (Fin D.R.t) 0 (G.s + 1) hc)).conj (GlobalTheory.StepData.reGL (GlobalTheory.StepData.ixZero (Fin D.R.k ⊕ Fin (G.s + 1)) (Fin D.R.t)) (x : GL (Fin D.R.k ⊕ Fin (G.s + 1)) G.k))) → Formalism.IndefR.Lev.finE (d.st 0).V ((d.t 0)⁻¹ * y * d.t 0) ∈ Formalism.L2.DIV.St (d.st 0).β → G.Red x (Formalism.L2.DIV.upperLeftBlock hr (Formalism.L2.DIV.redFn (d.st 0).β PP (Formalism.IndefR.Lev.finE

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelZeroUniformizerFrameUnipotentAndEmbedGL

theorem **back_eps** (i : Fin κ) : back (s

theorem **levelZeroOldIndexOfNew_inr** (i : Fin (s + 1)) : back (κ

theorem **scal_ne_zero** (ϖ : kˣ) (r : XIx (Fin κ) t 0 ⊕ Fin (s + 1)) : scal t ϖ r ≠ 0

theorem **bas_val** : ((S.T t).bas 0 (s + 1) (Nat.zero_add _)).val = diagonal (scal t S.ϖ)

theorem **bas_inv_val** : (((S.T t).bas 0 (s + 1) (Nat.zero_add _))⁻¹).val = diagonal fun r => (scal t S.ϖ r)⁻¹

theorem **upZero_entry** (x : GL (Fin κ ⊕ Fin (s + 1)) k) (r c : XIx (Fin κ) t 0 ⊕ Fin (s + 1)) : (upZero S t x).val r c = scal t S.ϖ r * x.val (back t r) (back t c) * (scal t S.ϖ c)⁻¹

theorem **xZero_symm** (i : Fin κ) : (xZero (Fin κ) t).symm (.inl (.inl i)) = i

theorem **upZero_mem_N** {n : GL (Fin κ ⊕ Fin (s + 1)) k} (hn : n ∈ N k (Fin κ) (s + 1) (S.ρ + 1)) : upZero S t n ∈ N k (XIx (Fin κ) t 0) (s + 1) (S.ρ + 1)

theorem **simpleSum_upZero** (n : GL (Fin κ ⊕ Fin (s + 1)) k) : simpleSum (upZero S t n) = ((S.ϖ⁻¹ : kˣ) : k) * simpleSum n

theorem **superSum_one** : superSum Fin.val (1 : Matrix (Fin κ) (Fin κ) k) = 0

theorem **lam_eq** (n : S.P.J) (hn : n.1 ∈ N k (Fin κ) (s + 1) (S.ρ + 1)) : ((S.P.lam n : ℂˣ) : ℂ) = (S.ψ (simpleSum (upZero S t n.1)) : ℂ)

theorem **upZero_embX** (γ : GL (Fin κ) k) : upZero S t (embX (s + 1) γ) = embX (s + 1) (reGL (xZero (Fin κ) t) γ)

theorem **upZero_N_embX** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (s κ : ℕ) (S : GlobalTheory.SymChain.SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], (∀ n : S.P.J, n.1 ∈ GlobalTheory.RootEx.N k (Fin κ) (s + 1) (S.ρ + 1) → GlobalTheory.LevZeroGen.upZero S t n.1 ∈ GlobalTheory.RootEx.N k (GlobalTheory.StepData.XIx (Fin κ) t 0) (s + 1) (S.ρ +

1) ∧ ((S.P.lam n : ℂˣ) : ℂ) = (S.ψ (GlobalTheory.RootEx.simpleSum (GlobalTheory.LevZeroGen.upZero S t n.1)) : ℂ)) ∧ ∀ γ : GL (Fin κ) k, GlobalTheory.LevZeroGen.upZero S t (GlobalTheory.RootEx.embX (s + 1) γ) = GlobalTheory.RootEx.embX (s + 1) (GlobalTheory.StepData.reGL (GlobalTheory.LevZeroGen.xZero (Fin κ) t) γ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevelZeroUniformizerFrameUnipotentAndEmbedGLStatement

theorem **_root_.GlobalTheory.T.levelZeroUniformizerFrame_unipotent_and_embedGL** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (s κ : ℕ) (S : GlobalTheory.SymChain.SymData k s κ) (t : Type) [Fintype t] [DecidableEq t], (∀ n : S.P.J, n.1 ∈ GlobalTheory.RootEx.N k (Fin κ) (s + 1) (S.ρ + 1) → GlobalTheory.LevZeroGen.upZero S t n.1 ∈ GlobalTheory.RootEx.N k (GlobalTheory.StepData.XIx (Fin κ) t 0) (s + 1) (S.ρ + 1) ∧ ((S.P.lam n : ℂˣ) : ℂ) = (S.ψ (GlobalTheory.RootEx.simpleSum (GlobalTheory.LevZeroGen.upZero S t n.1)) : ℂ)) ∧ ∀ γ : GL (Fin κ) k, GlobalTheory.LevZeroGen.upZero S t (GlobalTheory.RootEx.embX (s + 1) γ) = GlobalTheory.RootEx.embX (s + 1) (GlobalTheory.StepData.reGL (GlobalTheory.LevZeroGen.xZero (Fin κ) t) γ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LeviFactorCharactersSplitPlace

theorem **circle_eq_one_of_pos** {u : Circle} {t : ℝ} (ht : 0 < t) (h : (u : ℂ) = t) : u = 1

theorem **suppAt_one** (e : AE D) (V : HSp D) : SuppAt e V 1

theorem **isDetAt_one** (e : AE D) (V : HSp D) : IsDetAt e V 1 1

theorem **exists_form** (ψ : AddChar (AF D.host.F) ℂ) (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (om₀ : Om S₀ T) (hω : IsWeilDatumO ψ S T χS χT om) (hω₀ : IsWeilDatumO ψ S₀ T χS χT om₀) (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)) (hμ : IsCovolOne (ratMx D T.rank ρ) μ ℱ) (α : GL (Fin ρ) (AE D) →* Circle) (ξ : GA D S₀ →* Circle) (H : ∀ a g f, IsThetaFunO ψ S T f → d.mixS T ψ μ ℱ (om (d.levi a * d.emb0 g, 1) f) 0 = ((α a : ℂ) * (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.halfMod T.rank a : ℂ) * (ξ g : ℂ)) • om₀ (g, 1) (d.mixS T ψ μ ℱ (f) 0)) (e : AE D) (he : IsSplitIdem D e) : ∃ ℓ : (WO S T → ℂ) →ₗ[ℂ] ℂ, (∀ u : WO S T, (∀ i j, cA D e * u i j = u i j) → ∀ f, IsThetaFunO ψ S T f → ℓ (heisO ψ S T u f) = ℓ f) ∧ (∃ f, IsThetaFunO ψ S T f ∧ ℓ f ≠ 0) ∧ ∀ a g, SuppAt e S₀ g → ∀ x₀, IsDetAt e S₀ g x₀ → ∃ R : ℝ, 0 < R ∧ ∀ f, IsThetaF

theorem **levi_chars_split** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (om₀ : Om S₀ T), IsWeilDatumO ψ S T χS χT om → IsWeilDatumO ψ S₀ T χS χT om₀ → ∀ (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ ℱ → ∀ (α : GL (Fin ρ) (AE D) →* Circle) (ξ : GA D S₀ →* Circle), (∀ a g f, IsThetaFunO ψ S T f → d.mixS T ψ μ ℱ (om (d.levi a * d.emb0 g, 1) f) 0 = ((α a : ℂ) * (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.halfMod T.rank a : ℂ) * (ξ g : ℂ)) • om₀ (g, 1) (d.mixS T ψ μ ℱ (f) 0)) → ∀ e, IsSplitIdem D e → (∀ g, SuppAt e S₀ g → ξ g = 1) ∧ ∀ a : GL (Fin ρ) (AE D), (∀ i j, (e + cA D e) * (a.1 - 1) i j = (a.1 - 1) i j) → α a = (χT (Matrix.GeneralLinearGroup.det a))⁻¹

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LeviFactorCharactersSplitPlaceStatement

theorem **_root_.GlobalTheory.T.leviFactor_characters_splitPlace** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S S₀ T : HSp D) (ρ : ℕ) (d : HypDec S S₀ ρ) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (om₀ : Om S₀ T), IsWeilDatumO ψ S T χS χT om → IsWeilDatumO ψ S₀ T χS χT om₀ → ∀ (μ : Measure (Mx D T.rank ρ)) (ℱ : Set (Mx D T.rank ρ)), IsCovolOne (ratMx D T.rank ρ) μ ℱ → ∀ (α : GL (Fin ρ) (AE D) →* Circle) (ξ : GA D S₀ →* Circle), (∀ a g f, IsThetaFunO ψ S T f → d.mixS T ψ μ ℱ (om (d.levi a * d.emb0 g, 1) f) 0 = ((α a : ℂ) * (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.halfMod T.rank a : ℂ) * (ξ g : ℂ)) • om₀ (g, 1) (d.mixS T ψ μ ℱ (f) 0)) → ∀ e, IsSplitIdem D e → (∀ g, SuppAt e S₀ g → ξ g = 1) ∧ ∀ a : GL (Fin ρ) (AE D), (∀ i j, (e + cA D e) * (a.1 - 1) i j = (a.1 - 1) i j) → α a = (χT (Matrix.GeneralLinearGroup.det a))⁻¹

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapHaarFiniteOnCompacts

theorem **blocks_of_mem_R** {r : GL (n ⊕ Fin c) k} (hr : r ∈ R k n c p) : ∃ x : GL n k, r.val.toBlocks₁₁ = x.val ∧ (r⁻¹).val.toBlocks₁₁ = (x⁻¹).val

theorem **gpart_prodR** (q : GL n k × N k n c p) : gpart (prodR p q) = q.1

theorem **mem_N_of_mem_R** (r : R k n c p) : (embX c (gpart r))⁻¹ * (r : GL (n ⊕ Fin c) k) ∈ N k n c p

theorem **continuous_gpart** (p : ℕ) : Continuous (gpart (k

theorem **finite_r** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → IsFiniteMeasureOnCompacts ((m.prod (volN ν n c p)).map (prodR p))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapHaarFiniteOnCompactsStatement

theorem **_root_.GlobalTheory.T.levyProductMap_haar_finiteOnCompacts** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → IsFiniteMeasureOnCompacts ((m.prod (volN ν n c p)).map (prodR p))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapHaarIsMulLeftInvariant

theorem **GlobalTheory.HaarR.map_prodR_isMulLeftInvariant** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → ((m.prod (volN ν n c p)).map (prodR p)).IsMulLeftInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapHaarIsMulLeftInvariantStatement

theorem **_root_.GlobalTheory.T.levyProductMap_haar_isMulLeftInvariant** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), 1 ≤ p → p ≤ c → ∀ (ν : Measure k), ν.IsAddHaarMeasure → ∀ m : Measure (GL n k), m.IsHaarMeasure → ((m.prod (volN ν n c p)).map (prodR p)).IsMulLeftInvariant

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapPreimageIntegralGL

theorem **rootExchangeEmbedGL_mem_integralGL** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {n : Type} [Fintype n] [DecidableEq n] (c : ℕ) {g : GL n k} (hg : g ∈ KO k n) : embX c g ∈ KO k (n ⊕ Fin c)

theorem **prodR_preimage_KO** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), prodR (k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapPreimageIntegralGLStatement

theorem **_root_.GlobalTheory.T.levyProductMap_preimage_integralGL** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (c p : ℕ), prodR (k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapToFourierJacobiPreimageIntegralGL

theorem **uni_of_mem_N** {ρ : ℕ} {u : GL (n ⊕ Fin c) k} (hu : u ∈ N k n c (ρ + 1)) : Uni (midLev c ρ) u.val

theorem **uni_colE** (ρ : ℕ) (β : n → k) : Uni (midLev c ρ) (colE c (ρ + 1) β).val

theorem **uni_inl_inl** {ρ : ℕ} {M : Matrix (n ⊕ Fin c) (n ⊕ Fin c) k} (h : Uni (midLev c ρ) M) (i j : n) : M (Sum.inl i) (Sum.inl j) = (1 : Matrix n n k) i j

theorem **embX_mem_KO** {g : GL n k} (hg : g ∈ KO k n) : embX c g ∈ KO k (n ⊕ Fin c)

theorem **colE_mem_KO** (ρ : ℕ) {β : n → k} (hβ : ∀ i, β i ∈ 𝒪[k]) : colE c ρ β ∈ KO k (n ⊕ Fin c)

theorem **embX_mul_inl** {ρ : ℕ} (g : GL n k) (w : GL (n ⊕ Fin c) k) (hw : Uni (midLev c ρ) w.val) (i j : n) : (embX c g * w).val (Sum.inl i) (Sum.inl j) = g.val i j

theorem **mul_embX_inl** {ρ : ℕ} (g : GL n k) (w : GL (n ⊕ Fin c) k) (hw : Uni (midLev c ρ) w.val) (i j : n) : (w * embX c g).val (Sum.inl i) (Sum.inl j) = g.val i j

theorem **col_eq** {ρ : ℕ} (hρ : ρ < c) (u : GL (n ⊕ Fin c) k) (hu : u ∈ N k n c (ρ + 1)) (β : n → k) (t : n) : (u * colE c (ρ + 1) β).val (Sum.inl t) (Sum.inr ⟨ρ, hρ⟩) = β t

theorem **mem_KO_iff_prod** {ρ : ℕ} (hρ : ρ < c) (g : GL n k) (u : GL (n ⊕ Fin c) k) (hu : u ∈ N k n c (ρ + 1)) (β : n → k) : embX c g * u * colE c (ρ + 1) β ∈ KO k (n ⊕ Fin c) ↔ g ∈ KO k n ∧ u ∈ KO k (n ⊕ Fin c) ∧ ∀ i, β i ∈ 𝒪[k]

theorem **GlobalTheory.HaarFJInt.prodRFJ_preimage_KO** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ), ρ < c → prodRFJ (k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LevyProductMapToFourierJacobiPreimageIntegralGLStatement

theorem **_root_.GlobalTheory.T.levyProductMapToFourierJacobi_preimage_integralGL** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ), ρ < c → prodRFJ (k

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftConvContGrowth

theorem **lift_conv_cont_growth** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T), IsWeilDatumO ψ S T χS χT om → ∀ (v : Vol T) (G : GA D T → ℂ) (f : WO S T → ℂ), RapidDec T G → IsThetaFunO ψ S T f → (∀ g, IntegrableOn (fun h => om (g, h) f 0 * G h) v.𝓕 v.μ) ∧ IsAutFun S (liftS om v.μ v.𝓕 G f) ∧ IsContAut S (liftS om v.μ v.𝓕 G f) ∧ ModGrowth S (liftS om v.μ v.𝓕 G f) ∧ ∀ g₀ h₀, liftS om v.μ v.𝓕 (rTrans h₀ G) (om (g₀, h₀) f) = rTrans g₀ (liftS om v.μ v.𝓕 G f)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftLaws

theorem **std_target** : ∀ (D : D7) (c : Comm D) (v : VCom c) (G : GLoc v), IsStdChar D c.ψ (-1) → v.Is → ∀ (j : ℕ) (χS χT : (AE D)ˣ →* Circle) (om : Om (c.V j) (c.V (j + 1))), IsWeilDatumO c.ψ _ _ χS χT om → ∀ (μ : ℕ → G.kˣ →* ℂˣ), (∀ i, IsTame (μ i)) → (∀ i (x : G.kˣ), valuation G.k (x : G.k) = 1 → μ i x = G.S.μ i x) → (ofSplit G.q).IsComp χS⁻¹ (μ (j + 1)) → (∃ μ', (ofSplit G.q).IsComp χT μ' ∧ ∀ x : G.kˣ, valuation G.k (x : G.k) = 1 → μ' x * μ (j + 1) x = 1) → ∀ (b : ℕ) (hb : j + 1 + b = G.s + 1) (x : ((G.S.T (Fin D.R.t)).lev (j + 1) b hb).J) (h : GA D (c.V (j + 1))), (ofSplit G.q).IsLoc (G.FY (j + 1) b hb) h x.1 → ∀ f : WO (c.V j) (c.V (j + 1)) → ℂ, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· = G.q.v) f → om (1, h) (comb c.ψ _ _ (G.eg j) f) = ((((G.S.T (Fin D.R.t)).lev (j + 1) b hb).lam x : ℂ))⁻¹ • comb c.ψ _ _ (G.eg j) f

theorem **std_source** : ∀ (D : D7) (c : Comm D) (v : VCom c) (G : GLoc v), IsStdChar D c.ψ (-1) → v.Is → ∀ (j : ℕ) (χS χT : (AE D)ˣ →* Circle) (om : Om (c.V j) (c.V (j + 1))), IsWeilDatumO c.ψ _ _ χS χT om → ∀ (μ : ℕ → G.kˣ →* ℂˣ), (∀ i, IsTame (μ i)) → (∀ i (x : G.kˣ), valuation G.k (x : G.k) = 1 → μ i x = G.S.μ i x) → (ofSplit G.q).IsComp χS⁻¹ (μ (j + 1)) → (∃ μ', (ofSplit G.q).IsComp χT μ' ∧ ∀ x : G.kˣ, valuation G.k (x : G.k) = 1 → μ' x * μ (j + 1) x = 1) → j ≤ G.s → ∀ (b : ℕ) (hb : j + b = G.s + 1) (x : ((G.S.T (Fin D.R.t)).lev j b hb).J) (h : GA D (c.V j)), (ofSplit G.q).IsLoc (G.FY j b hb) h x.1 → ∀ f : WO (c.V j) (c.V (j + 1)) → ℂ, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· = G.q.v) f → om (h, 1) (comb c.ψ _ _ (G.eg j) f) = (((G.S.T (Fin D.R.t)).lev j b hb).lam x : ℂ) • comb c.ψ _ _ (G.eg j) f

theorem **std_isolate** : ∀ (D : D7) (c : Comm D) (v : VCom c) (n : ℕ) (G : Fin n → GLoc v), IsStdChar D c.ψ (-1) → v.Is → GPin G → ∀ (j : ℕ), j + 1 < D.R.m → ∀ (f : WO (c.V j) (c.V (j + 1)) → ℂ), Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· ∈ v.gP) f → ∀ (i : Fin n), ∃ g, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· = (G i).q.v) g ∧ comb c.ψ _ _ (v.eG j) f = comb c.ψ _ _ ((G i).eg j) g

theorem **loc_tgt** {D : D7} {c : Comm D} {v : VCom c} (G : GLoc v) (hc : IsStdChar D c.ψ (-1)) (hv : v.Is) {j : ℕ} {χS χT : (AE D)ˣ →* Circle} {om : Om (c.V j) (c.V (j + 1))} (hom : IsWeilDatumO c.ψ _ _ χS χT om) (μ : ℕ → G.kˣ →* ℂˣ) (hμ : ∀ i, IsTame (μ i)) (hu : ∀ i (x : G.kˣ), valuation G.k (x : G.k) = 1 → μ i x = G.S.μ i x) (hS : (ofSplit G.q).IsComp χS⁻¹ (μ (j + 1))) (hT : ∃ μ', (ofSplit G.q).IsComp χT μ' ∧ ∀ x : G.kˣ, valuation G.k (x : G.k) = 1 → μ' x * μ (j + 1) x = 1) (b : ℕ) (hb : j + 1 + b = G.s + 1) (x : ((G.S.T (Fin D.R.t)).lev (j + 1) b hb).J) (h : GA D (c.V (j + 1))) (hx : (ofSplit G.q).IsLoc (G.FY (j + 1) b hb) h x.1) (f : WO (c.V j) (c.V (j + 1)) → ℂ) (hf : Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· = G.q.v) f) : om (1, h) (comb c.ψ _ _ (G.eg j) f) = ((((G.S.T (Fin D.R.t)).lev (j + 1) b hb).lam x : ℂ))⁻¹ • comb c.ψ _ _ (G.eg j) f

theorem **seeSaw_local_sourceLevel_character** {D : D7} {c : Comm D} {v : VCom c} (G : GLoc v) (hc : IsStdChar D c.ψ (-1)) (hv : v.Is) {j : ℕ} {χS χT : (AE D)ˣ →* Circle} {om : Om (c.V j) (c.V (j + 1))} (hom : IsWeilDatumO c.ψ _ _ χS χT om) (μ : ℕ → G.kˣ →* ℂˣ) (hμ : ∀ i, IsTame (μ i)) (hu : ∀ i (x : G.kˣ), valuation G.k (x : G.k) = 1 → μ i x = G.S.μ i x) (hS : (ofSplit G.q).IsComp χS⁻¹ (μ (j + 1))) (hT : ∃ μ', (ofSplit G.q).IsComp χT μ' ∧ ∀ x : G.kˣ, valuation G.k (x : G.k) = 1 → μ' x * μ (j + 1) x = 1) (hj : j ≤ G.s) (b : ℕ) (hb : j + b = G.s + 1) (x : ((G.S.T (Fin D.R.t)).lev j b hb).J) (h : GA D (c.V j)) (hx : (ofSplit G.q).IsLoc (G.FY j b hb) h x.1) (f : WO (c.V j) (c.V (j + 1)) → ℂ) (hf : Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· = G.q.v) f) : om (h, 1) (comb c.ψ _ _ (G.eg j) f) = (((G.S.T (Fin D.R.t)).lev j b hb).lam x : ℂ) • comb c.ψ _ _ (G.eg j) f

theorem **decomp** {D : D7} {c : Comm D} {v : VCom c} {n : ℕ} (G : Fin n → GLoc v) (hG : GPin G) (j : ℕ) (k : GA D (c.V j)) (hk : k ∈ grpAt (c.V j) (· ∈ v.gP)) : ∃ h : Fin n → GA D (c.V j), (∀ i b hb, ∃ x, (G i).Mat j b hb k x ∧ (ofSplit (G i).q).IsLoc ((G i).FY j b hb) (h i) x) ∧ k = ((List.finRange n).map h).prod

theorem **isolate** {D : D7} {c : Comm D} {v : VCom c} {n : ℕ} (G : Fin n → GLoc v) (hc : IsStdChar D c.ψ (-1)) (hv : v.Is) (hG : GPin G) (j : ℕ) (hj : j + 1 < D.R.m) (f : WO (c.V j) (c.V (j + 1)) → ℂ) (hf : Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· ∈ v.gP) f) (i : Fin n) : ∃ g, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· = (G i).q.v) g ∧ comb c.ψ _ _ (v.eG j) f = comb c.ψ _ _ ((G i).eg j) g

theorem **tgt_law** {D : D7} {c : Comm D} {v : VCom c} {n : ℕ} (G : Fin n → GLoc v) (hc : IsStdChar D c.ψ (-1)) (hv : v.Is) (hG : GPin G) {j : ℕ} (hj : j + 1 < D.R.m) {χS χT : (AE D)ˣ →* Circle} {om : Om (c.V j) (c.V (j + 1))} (hom : IsWeilDatumO c.ψ _ _ χS χT om) (hch : ∀ i, ∃ μ : ℕ → (G i).kˣ →* ℂˣ, (∀ l, IsTame (μ l)) ∧ (∀ l (x : (G i).kˣ), valuation (G i).k (x : (G i).k) = 1 → μ l x = (G i).S.μ l x) ∧ (ofSplit (G i).q).IsComp χS⁻¹ (μ (j + 1)) ∧ ∃ μ', (ofSplit (G i).q).IsComp χT μ' ∧ ∀ x : (G i).kˣ, valuation (G i).k (x : (G i).k) = 1 → μ' x * μ (j + 1) x = 1) : ∀ k ∈ v.JG (j + 1), ∃ a, a ≠ 0 ∧ GLam G (j + 1) k a ∧ ∀ f, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· ∈ v.gP) f → om (1, k) (comb c.ψ _ _ (v.eG j) f) = a⁻¹ • comb c.ψ _ _ (v.eG j) f

theorem **src_law** {D : D7} {c : Comm D} {v : VCom c} {n : ℕ} (G : Fin n → GLoc v) (hc : IsStdChar D c.ψ (-1)) (hv : v.Is) (hG : GPin G) {j : ℕ} (hj : j + 1 < D.R.m) {χS χT : (AE D)ˣ →* Circle} {om : Om (c.V j) (c.V (j + 1))} (hom : IsWeilDatumO c.ψ _ _ χS χT om) (hch : ∀ i, ∃ μ : ℕ → (G i).kˣ →* ℂˣ, (∀ l, IsTame (μ l)) ∧ (∀ l (x : (G i).kˣ), valuation (G i).k (x : (G i).k) = 1 → μ l x = (G i).S.μ l x) ∧ (ofSplit (G i).q).IsComp χS⁻¹ (μ (j + 1)) ∧ ∃ μ', (ofSplit (G i).q).IsComp χT μ' ∧ ∀ x : (G i).kˣ, valuation (G i).k (x : (G i).k) = 1 → μ' x * μ (j + 1) x = 1) : ∀ k ∈ v.JG j, ∃ a, a ≠ 0 ∧ GLam G j k a ∧ ∀ f, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· ∈ v.gP) f → om (k, 1) (comb c.ψ _ _ (v.eG j) f) = a • comb c.ψ _ _ (v.eG j) f

theorem **GlobalTheory.RowLevel.Std.lift_laws** : ∀ (D : D7) (c : Comm D) (v : VCom c) (n : ℕ) (G : Fin n → GLoc v), IsStdChar D c.ψ (-1) → v.Is → GPin G → ∀ t : Tup c, t.Is → (∀ i, (G i).Chars t) → ∀ j σ (_ : j + 2 * σ + 2 = D.R.m), (∀ k ∈ v.JG (j + 1), ∃ a, a ≠ 0 ∧ GLam G (j + 1) k a ∧ ∀ f, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· ∈ v.gP) f → t.om j (1, k) (comb c.ψ _ _ (v.eG j) f) = a⁻¹ • comb c.ψ _ _ (v.eG j) f) ∧ (∀ k ∈ v.JG j, ∃ a, a ≠ 0 ∧ GLam G j k a ∧ ∀ f, Vac c.ψ _ _ (v.N j) (v.N (j + 1)) (· ∈ v.gP) f → t.om j (k, 1) (comb c.ψ _ _ (v.eG j) f) = a • comb c.ψ _ _ (v.eG j) f) ∧ (0 < σ → (∀ k ∈ v.JG (j + 1 + 1), ∃ a, a ≠ 0 ∧ GLam G (j + 1 + 1) k a ∧ ∀ f, Vac c.ψ _ _ (v.N (j + 1)) (v.N (j + 1 + 1)) (· ∈ v.gP) f → t.om (j + 1) (1, k) (comb c.ψ _ _ (v.eG (j + 1)) f) = a⁻¹ • comb c.ψ _ _ (v.eG (j + 1)) f) ∧ ∀ k ∈ v.JG (j + 1), ∃ a, a ≠ 0 ∧ GLam G (j + 1) k a ∧ ∀ f, Vac c.ψ _ _ (v.N (j + 1)) (v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftNearOne

theorem **near_left** {K Γ : Type*} [Field K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) {n : ℕ} {C M : Matrix (Fin n) (Fin n) K} (hC : ∀ i j, v (C i j) ≤ 1) (hM : ∀ i j, v (M i j - (1 : Matrix (Fin n) (Fin n) K) i j) < 1) (h : ∀ i j, v ((C * M) i j - (1 : Matrix (Fin n) (Fin n) K) i j) < 1) (i j : Fin n) : v (C i j - (1 : Matrix (Fin n) (Fin n) K) i j) < 1

theorem **near_right** {K Γ : Type*} [Field K] [LinearOrderedCommGroupWithZero Γ] (v : Valuation K Γ) {n : ℕ} {C M : Matrix (Fin n) (Fin n) K} (hM : ∀ i j, v (M i j) ≤ 1) (hC : ∀ i j, v (C i j - (1 : Matrix (Fin n) (Fin n) K) i j) < 1) (h : ∀ i j, v ((C * M) i j - (1 : Matrix (Fin n) (Fin n) K) i j) < 1) (i j : Fin n) : v (M i j - (1 : Matrix (Fin n) (Fin n) K) i j) < 1

theorem **kMat_mul** (P : HeightOneSpectrum (𝒪 E)) (g k : GL (Fin n) (AfE F E)) (f : Sp E n ≃ₗ[E] Sp E n) : kMat P (g * k) f = kMat P k 1 * kMat P g f

theorem **kMat_one** (P : HeightOneSpectrum (𝒪 E)) (f : Sp E n ≃ₗ[E] Sp E n) (i j : Fin n) : kMat P (1 : GL (Fin n) (AfE F E)) f i j = algebraMap E _ (f (Pi.single j 1) i)

theorem **lift_near_one** {y k k' : GL (Fin n) (AfE F E)} {p p₀ : DecoF E n ST SG Q} (h : IsPtOf hc (y * k) p) (h' : IsPtOf hc (y * k') p) (h₀ : IsPtOf hc k' p₀) (hN : p₀.deco.1.N = stdLat E n) (w : SG) (hk : bc (prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftPairPositiveDefiniteWittIndexZeroCartan

theorem **main** : ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), GlobalTheory.Hk3.Core R F → (∃ L : SizeData₂ D, Nonempty (ParameterChoice' D L)) → ∀ (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), GlobalTheory.Cusp.wittIndex (Vc D i) = 0 ∧ (Wc D i).rank < D.R.t ∧ ∃ cA : GlobalTheory.Mem.Th.CptF (Wc D i), GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.HidaSetting R.θ (Wc D i) (Vc D i) (cart (R.MW i σ)) (fun _ => 1) D.R.t cA ∧ ∀ U ∈ R.Wall i σ, ∀ v : ∀ ρ, lowT (R.MW i σ) ρ, U (fun ρ => v ρ) ∈ GlobalTheory.Cusp.A0 R.θ (Wc D i) cA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftPairPositiveDefiniteWittIndexZeroCartanStatement

theorem **_root_.GlobalTheory.T.liftPair_positiveDefinite_wittIndexZero_cartan** : ∀ {D : D7} {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {A : FamIn D.what₂.LL G Pl0 T D.R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks D.what₂.LL] (R : RealDat D A Ks) (F : Face D A), GlobalTheory.Hk3.Core R F → (∃ L : SizeData₂ D, Nonempty (ParameterChoice' D L)) → ∀ (i : Fin D.R.m) (σ : D.what₂.LL →+* ℂ), GlobalTheory.Cusp.wittIndex (Vc D i) = 0 ∧ (Wc D i).rank < D.R.t ∧ ∃ cA : GlobalTheory.Mem.Th.CptF (Wc D i), GlobalTheory.IndefiniteRowSideFullHecke.SphericalVolumeLift.HidaSetting R.θ (Wc D i) (Vc D i) (cart (R.MW i σ)) (fun _ => 1) D.R.t cA ∧ ∀ U ∈ R.Wall i σ, ∀ v : ∀ ρ, lowT (R.MW i σ) ρ, U (fun ρ => v ρ) ∈ GlobalTheory.Cusp.A0 R.θ (Wc D i) cA

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftsCharOfResidueCharAtSplitPlace

theorem **swapProj_one_tmul** (φ : SplitProj D v) (a : Fv D.host.F v) : swapProj φ ((1 : D.host.E) ⊗ₜ[D.host.F] a) = a

theorem **swapProj_prV** (φ : SplitProj D v) (z : EAf D) : swapProj φ (Formalism.UG.prV v z) = GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.rdE φ z

theorem **swapProj_prV_oToS** (φ : SplitProj D v) (x : 𝒪 D.host.E) : swapProj φ (Formalism.UG.prV v (Formalism.L2.DIV.oToS D.host.F D.host.E x).1) = GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.rdQ φ (x : D.host.E)

theorem **isIdemOf_idem** (φ : SplitProj D v) : IsIdemOf φ (GlobalTheory.EmbChart.idem (GlobalTheory.SplitProjChart.emb φ))

theorem **eq_splitPrime** {φ : SplitProj D v} {PP : Pl D.host.E} (hφ : IsProjAt φ PP) : PP = splitPrime (GlobalTheory.SplitProjChart.emb φ)

theorem **exists_mem_only** (τ : D.host.E →ₐ[D.host.F] Fv D.host.F v) : ∃ p ∈ (splitPrime τ).asIdeal, ∀ P : HeightOneSpectrum (𝒪 D.host.E), P.under (𝒪 D.host.F) = v → p ∈ P.asIdeal → P = splitPrime τ

theorem **resS_eq** {φ : SplitProj D v} {PP : Pl D.host.E} (hφ : IsProjAt φ PP) (s : Formalism.L2.Herm.intAf D.host.F D.host.E) (x : 𝒪 D.host.E) (h : Valued.v (GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.rdE φ s.1 - GlobalTheory.IndefiniteRowSideFullHecke.RestrictedTensorSplitting.rdQ φ (x : D.host.E)) < 1) : resS D.host.F PP s = Ideal.Quotient.mk PP.asIdeal x

theorem **exists_unit_lift** {A K : Type} [CommRing A] [CommRing K] (π : A →+* K) {e : A} (hee : e * e = e) (hs : Function.Surjective π) (hk : ∀ a, π a = 0 ↔ e * a = 0) (u : Kˣ) : ∃ x : Aˣ, (1 - e) * (x : A) = 1 - e ∧ e * ((x : A) - 1) = (x : A) - 1 ∧ π x = u

theorem **liftsChar_of_resAt** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle) (PP : Pl D.host.E) (a : (kP PP)ˣ →* ℂˣ), GlobalTheory.StepLaw.ResAt χ PP a → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ μ : (Fv D.host.F v)ˣ →* ℂˣ, (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μ y = Circle.toUnits (χ x)) → LiftsChar φ PP μ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiftsCharOfResidueCharAtSplitPlaceStatement

theorem **_root_.GlobalTheory.T.liftsChar_of_residueChar_at_splitPlace** : ∀ (D : D7) (χ : (AE D)ˣ →* Circle) (PP : Pl D.host.E) (a : (kP PP)ˣ →* ℂˣ), GlobalTheory.StepLaw.ResAt χ PP a → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ μ : (Fv D.host.F v)ˣ →* ℂˣ, (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 − e) ∗ (x : AE D) = 1 − e ∧ rdA φ (x : AE D) = y) → μ y = Circle.toUnits (χ x)) → LiftsChar φ PP μ a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineContractionSplitPlaceLocalModel

theorem **contrLoc** : ∀ (D : D7), GlobalTheory.LocFlat.ContrFactor → ContrLoc D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineContractionSplitPlaceLocalModelStatement

theorem **_root_.GlobalTheory.T.lineContraction_splitPlace_localModel** : ∀ (D : D7), GlobalTheory.LocFlat.ContrFactor → ContrLoc D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineHomAdjointTwistMemCompletedHomLattice

theorem **incA_eq** (e : D.host.E) : incA D e = (1 : AF D.host.F) ⊗ₜ[D.host.F] e

theorem **tmul_eq** (γ : Af D.host.F) (e : D.host.E) : pairA D.host.F 0 γ ⊗ₜ[D.host.F] (1 ∗ e) = (pairA D.host.F 0 γ ⊗ₜ[D.host.F] (1 : D.host.E)) ∗ incA D e

theorem **cA_tmul_one** (a : AF D.host.F) : cA D (a ⊗ₜ[D.host.F] (1 : D.host.E)) = a ⊗ₜ[D.host.F] (1 : D.host.E)

theorem **pairH_add_left** (L : HSp D) (z z' y : Fin L.rank → AE D) : pairH L (z + z') y = pairH L z y + pairH L z' y

theorem **pairH_neg_left** (L : HSp D) (z y : Fin L.rank → AE D) : pairH L (−z) y = −pairH L z y

theorem **pairH_zero_left** (L : HSp D) (y : Fin L.rank → AE D) : pairH L 0 y = 0

theorem **pairH_add_right** (L : HSp D) (z y y' : Fin L.rank → AE D) : pairH L z (y + y') = pairH L z y + pairH L z y'

theorem **pairH_neg_right** (L : HSp D) (z y : Fin L.rank → AE D) : pairH L z (−y) = −pairH L z y

theorem **pairH_zero_right** (L : HSp D) (z : Fin L.rank → AE D) : pairH L z 0 = 0

theorem **pairH_gen** (L : HSp D) (α β : Af D.host.F) (x x' : Fin L.rank → D.host.E) : pairH L (fun i => pairA D.host.F 0 α ⊗ₜ[D.host.F] (1 ∗ x i)) (fun i => pairA D.host.F 0 β ⊗ₜ[D.host.F] (1 ∗ x' i)) = pairA D.host.F 0 (α ∗ β) ⊗ₜ[D.host.F] (1 ∗ ∑ i, ∑ j, star (x i) ∗ L.gram i j ∗ x' j)

theorem **pairH_int** {L : HSp D} {LL : Lat L} (hL : IsSelfDualC L LL) {P : Pl D → Prop} {z y : Fin L.rank → AE D} (hz : z ∈ latV L LL P) (hy : y ∈ latV L LL P) : finPart D (pairH L z y) ∈ intAE D

theorem **pairH_eq** (S : HSp D) (t : Fin S.rank → D.host.E) (x : Fin S.rank → AE D) : pairH S (fun l => incA D (t l)) x = ∑ b, (∑ k, incA D (star (t k)) ∗ gA S k b) ∗ x b

theorem **smul_mulVec'** {m n : ℕ} (c : AE D) (M : Mx D m n) (v : Fin n → AE D) : (c • M) *ᵥ v = c • (M *ᵥ v)

theorem **mem_latV_of_pairH** {S : HSp D} {MS : Lat S} (hS : IsSelfDualC S MS) {P : Pl D → Prop} {x : Fin S.rank → AE D} (hx : ∀ b, x b ∈ idAt D P) (hint : ∀ t ∈ MS, finPart D (pairH S (fun a => incA D (t a)) x) ∈ intAE D) : x ∈ latV S MS P

theorem **pairH_kap** (S L L'' : HSp D) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (w : WO S L'') (t : Fin S.rank → D.host.E) (y : Fin L.rank → AE D) : pairH S (fun a => incA D (t a)) (GlobalTheory.FinPl.kap τ S w *ᵥ y) = incA D (−D.host.xi0⁻¹) ∗ pairH L ((rat τ : Mx D L.rank L''.rank) *ᵥ (w *ᵥ fun a => incA D (t a))) y

theorem **kap_mem_latO** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S L L'' : HSp D) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (MS : GlobalTheory.FinPl.Lat S) (LL : GlobalTheory.FinPl.Lat L) (LZ : GlobalTheory.FinPl.Lat L''), q.UnitAA → IsSelfDualC S MS → IsSelfDualC L LL → IsSelfDualC L'' LZ → (∀ y ∈ GlobalTheory.FinPl.latV L'' LZ (· = q.v), (rat τ : Mx D L.rank L''.rank) *ᵥ y ∈ GlobalTheory.FinPl.latV L LL (· = q.v)) → ∀ w ∈ latO S L'' (MS : Set _) (LZ : Set _) (· = q.v), GlobalTheory.FinPl.kap τ S w ∈ latO L S (LL : Set _) (MS : Set _) (· = q.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineHomAdjointTwistMemCompletedHomLatticeStatement

theorem **_root_.GlobalTheory.T.lineHomAdjointTwist_mem_completedHomLattice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S L L'' : HSp D) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E) (MS : GlobalTheory.FinPl.Lat S) (LL : GlobalTheory.FinPl.Lat L) (LZ : GlobalTheory.FinPl.Lat L''), q.UnitAA → IsSelfDualC S MS → IsSelfDualC L LL → IsSelfDualC L'' LZ → (∀ y ∈ GlobalTheory.FinPl.latV L'' LZ (· = q.v), (rat τ : Mx D L.rank L''.rank) *ᵥ y ∈ GlobalTheory.FinPl.latV L LL (· = q.v)) → ∀ w ∈ latO S L'' (MS : Set _) (LZ : Set _) (· = q.v), GlobalTheory.FinPl.kap τ S w ∈ latO L S (LL : Set _) (MS : Set _) (· = q.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineLatticeSelfDual

theorem **isSelfDualC_lineLat** : ∀ (D : D7) (S : HSp D) (ζ : Fin S.rank → D.host.E), IsLineVec S ζ → ∀ (𝔄 : FI D), (∀ d : D.host.E, d ∈ 𝔄 ↔ ∀ c ∈ 𝔄, ∃ b : 𝒪 D.host.E, (b : D.host.E) = star c ∗ d ∗ (star ζ ·ᵥ S.gram *ᵥ ζ)) → IsSelfDualC S (GlobalTheory.SeeSaw.LiftModel.lineLat S ζ 𝔄)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineMemLatVIff

theorem **pr_ofSplit** (q : SplitPt D k) (x : D.host.E) : (ofSplit q).pr x = q.pr (incA D x)

theorem **eq_sec_add_cA** (q : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = q.v)) : a = q.sec (q.pr a) + cA D (q.sec (q.pr (cA D a)))

theorem **pr_mem_of_int** (q : SplitPt D k) {a : AE D} (ha : finPart D a ∈ intAE D) : q.pr a ∈ 𝒪[k]

theorem **mem_intAt_iff_pr** (q : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = q.v)) : a ∈ intAt D (· = q.v) ↔ q.pr a ∈ 𝒪[k] ∧ q.pr (cA D a) ∈ 𝒪[k]

theorem **single_int** {Y' : Type} [DecidableEq Y'] (m₁ : Y') (i : Y') : (Pi.single m₁ (1 : k) : Y' → k) i ∈ 𝒪[k]

theorem **wpart** {q : SplitPt D k} {S : HSp D} {L : Lat S} {F : Frame k S.rank Y} {u : Fin S.rank → D.host.E} {c : k} {m₀ : Y} (hI : IntIn q S F L) (hc : valuation k c = 1) (hu : ∀ r, c ∗ (ofSplit q).pr (u r) = F.β r m₀) (x : k) : (fun r => q.sec (x ∗ (ofSplit q).pr (u r))) ∈ latV S L (· = q.v) ↔ x ∈ 𝒪[k]

theorem **wbar** {q : SplitPt D k} {S : HSp D} {L : Lat S} {F : Frame k S.rank Y} {u : Fin S.rank → D.host.E} (hL : IsSelfDualC S L) (hI : IntIn q S F L) (hb : ∀ m, ∑ r, F.β r m ∗ (ofSplit q).pr (star ((S.gram *ᵥ u) r)) ∈ 𝒪[k]) (hb1 : ∃ m, valuation k (∑ r, F.β r m ∗ (ofSplit q).pr (star ((S.gram *ᵥ u) r))) = 1) (A : k) : (fun r => cA D (q.sec (A ∗ (ofSplit q).pr (star (u r))))) ∈ latV S L (· = q.v) ↔ A ∈ 𝒪[k]

theorem **line_mem_latV_iff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (S : HSp D) (L : GlobalTheory.FinPl.Lat S) (Y : Type) [Fintype Y] [DecidableEq Y] (F : Frame k S.rank Y) (u : Fin S.rank → D.host.E) (c : k) (m₀ : Y), IsSelfDualC S L → IntIn q S F L → valuation k c = 1 → (∀ r, c ∗ (ofSplit q).pr (u r) = F.β r m₀) → (∀ m, ∑ r, F.β r m ∗ (ofSplit q).pr (star ((S.gram ∗ᵥ u) r)) ∈ 𝒪[k]) → (∃ m, valuation k (∑ r, F.β r m ∗ (ofSplit q).pr (star ((S.gram ∗ᵥ u) r))) = 1) → ∀ a ∈ GlobalTheory.FinPl.idAt D (· = q.v), ((fun r => a ∗ incA D (u r)) ∈ GlobalTheory.FinPl.latV S L (· = q.v) ↔ a ∈ GlobalTheory.FinPl.intAt D (· = q.v))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineModelKappaTransformIsLocalModel

theorem **kappa_model** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (L S L'' : HSp D) (ζ : Fin L.rank → D.host.E) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), L''.rank = 1 → τᴴ ∗ (L.gram ∗ τ) = (-(D.host.xi0 ∗ star D.host.xi0)) • L''.gram → ∀ (s : k) (z₀ : Fin L''.rank → k), s ≠ 0 → τ.map p.pr ∗ᵥ z₀ = (-(p.xi ∗ s)) • (fun i => p.pr (ζ i)) → ∀ (n : Type) [Fintype n] [DecidableEq n] (FS : Frame k S.rank n) (FL : Frame k L.rank Unit), (∀ r, FL.β r () = s ∗ p.pr (ζ r)) → ∀ TLS, p.IsModel ψ FL FS TLS → ∃ FL'' : Frame k L''.rank Unit, (∀ r, FL''.β r () = z₀ r) ∧ ∃ TL, p.IsModel ψ FS FL'' TL ∧ ∀ (c : SB (Matrix n Unit k)) (Φ : SB (Matrix Unit n k)),

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineModelKappaTransformIsLocalModelStatement

theorem **_root_.GlobalTheory.T.lineModel_kappa_transform_isLocalModel** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (L S L'' : HSp D) (ζ : Fin L.rank → D.host.E) (τ : Matrix (Fin L.rank) (Fin L''.rank) D.host.E), L''.rank = 1 → τᴴ ∗ (L.gram ∗ τ) = (-(D.host.xi0 ∗ star D.host.xi0)) • L''.gram → ∀ (s : k) (z₀ : Fin L''.rank → k), s ≠ 0 → τ.map p.pr ∗ᵥ z₀ = (-(p.xi ∗ s)) • (fun i => p.pr (ζ i)) → ∀ (n : Type) [Fintype n] [DecidableEq n] (FS : Frame k S.rank n) (FL : Frame k L.rank Unit), (∀ r, FL.β r () = s ∗ p.pr (ζ r)) → ∀ TLS, p.IsModel ψ FL FS TLS → ∃ FL'' : Frame k L''.rank Unit, (∀ r, FL''.β r () = z₀ r) ∧ ∃ TL, p.IsModel ψ FS FL'' TL ∧ ∀ (c : S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineMultipleMemSplitCoreLatticeCompletionIff

theorem **e_half** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) (α : Frame p A n) (α₀ : Frame p S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : Submodule 𝒪[k] (n → k)), IsSumFrame o ζ α α₀ → α₀.IsLat M (lastKer Λ) → ∀ c : AE D, ((fun r => p.e ∗ (c ∗ incA D (vecB o ζ r))) ∈ latV S₀ M (· = p.v) ↔ p.pr c ∈ 𝒪[k])

theorem **mem_latV_iff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : ∀ i, Frame (p i) A n) (α₀ : ∀ i, Frame (p i) S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : ι → Submodule 𝒪[k] (n → k)), IsSelfDualC S₀ M → (∀ i, IsSumFrame o ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ c : AE D, c ∈ idAt D P → ((fun r => c ∗ incA D (vecB o ζ r)) ∈ latV S₀ M P ↔ (∀ i, (p i).pr c ∈ 𝒪[k] ∧ (p i).pr (cA D c) ∗ (p i).pr (incA D (star ζ ·ᵥ HSpace.gram L ∗ᵥ ζ)) ∈ 𝒪[k]))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineMultipleMemSplitCoreLatticeCompletionIffStatement

theorem **_root_.GlobalTheory.T.lineMultiple_mem_splitCoreLattice_completion_iff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : ∀ i, Frame (p i) A n) (α₀ : ∀ i, Frame (p i) S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : ι → Submodule 𝒪[k] (n → k)), IsSelfDualC S₀ M → (∀ i, IsSumFrame o ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ c : AE D, c ∈ idAt D P → ((fun r => c ∗ incA D (vecB o ζ r)) ∈ latV S₀ M P ↔ (∀ i, (p i).pr c ∈ 𝒪[k] ∧ (p i).pr (cA D c) ∗ (p i).pr (incA D (star ζ ·ᵥ HSpace.gram L ∗ᵥ ζ)) ∈ 𝒪[k]))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineOfGlued

theorem **pr_mem_of_intAt** (p : SplitPt D k) {a : AE D} (ha : finPart D a ∈ intAE D) : p.pr a ∈ 𝒪[k]

theorem **eq_sec_add** (p : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = p.v)) : a = p.sec (p.pr a) + cA D (p.sec (p.pr (cA D a)))

theorem **mem_intAt_iff** (p : SplitPt D k) {a : AE D} (ha : a ∈ idAt D (· = p.v)) : a ∈ intAt D (· = p.v) ↔ p.pr a ∈ 𝒪[k] ∧ p.pr (cA D a) ∈ 𝒪[k]

theorem **valuation_xi_zero_eq_one** (p : SplitPt D k) (hp : p.UnitAA) : valuation k (p.pr (incA D D.host.xi0)) = 1

theorem **single_mem_glued_iff** {n : Type} [DecidableEq n] (q : Unif k) (P : Pair k n) (t : k) : (Pi.single (Sum.inr 0) t : n ⊕ Fin 1 → k) ∈ gluedS q 1 P.ker P.vec ↔ t ∈ 𝒪[k]

theorem **pair_sumFrame** {p : SplitPt D k} {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) {n : Type} [Fintype n] [DecidableEq n] {β : Frame p A n} {β₀ : Frame p S (n ⊕ Fin 1)} (hs : IsSumFrame o ζ β β₀) (x : n ⊕ Fin 1 → k) (t : k) : ∑ i, ∑ j, (β₀.b ∗ᵥ x) i ∗ p.pr (cA D (incA D (S.gram i j))) ∗ (t ∗ p.pr (incA D (star (vecB o ζ j)))) = t ∗ (p.pr (incA D (star ζ ·ᵥ L.gram ∗ᵥ ζ)) ∗ x (Sum.inr 0))

theorem **line_of_glued** {p : SplitPt D k} {ρ : ℕ} {B : BesLev D ρ} {M : Lat B.V₀} {q : Unif k} {n : Type} [Fintype n] [DecidableEq n] {β : Frame p B.A n} {β₀ : Frame p B.V₀ (n ⊕ Fin 1)} {P : Pair k n} (hM : IsSelfDualC B.V₀ M) (hg : GluedCore B M q β β₀ P) {a : AE D} (ha : a ∈ idAt D (· = p.v)) : a • ratV (vecB B.o B.ζ) ∈ latV B.V₀ M (· = p.v) ↔ a ∈ intAt D (· = p.v)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineProjectionMemHomLatticeCompletion

theorem **hermAt_eq** (S : HSp D) (x y : Fin S.rank → AE D) : hermAt S x y = (fun i => cA D (y i)) ·ᵥ (gA S ∗ᵥ x)

theorem **hermAt_mulVec** (S : HSp D) {π : Mx D S.rank S.rank} (h : adj π · gA S = gA S · π) (x y : Fin S.rank → AE D) : hermAt S (π ∗ᵥ x) y = hermAt S x (π ∗ᵥ y)

theorem **hermAt_smul** (S : HSp D) (a : AE D) (x y : Fin S.rank → AE D) : hermAt S (a • x) y = hermAt S x (cA D a • y)

theorem **hermAt_add_left** (S : HSp D) (x x' y : Fin S.rank → AE D) : hermAt S (x + x') y = hermAt S x y + hermAt S x' y

theorem **add_mem_intAt** {P : Pl D → Prop} {a b : AE D} (ha : a ∈ intAt D P) (hb : b ∈ intAt D P) : a + b ∈ intAt D P

theorem **proj_selfAdj** {S A L : HSp D} (o : OrthDec S A L) : adj (o.jB · o.pB) · gA S = gA S · (o.jB · o.pB)

theorem **mulVec_mem_of_half** (p : SplitPt D k) {S : HSp D} {M : Lat S} (hM : IsSelfDualC S M) {π : Mx D S.rank S.rank} (h : adj π · gA S = gA S · π) (hw : ∀ z ∈ latV S M (· = p.v), π ∗ᵥ (p.e • z) ∈ latV S M (· = p.v)) {y : Fin S.rank → AE D} (hy : y ∈ latV S M (· = p.v)) : π ∗ᵥ y ∈ latV S M (· = p.v)

theorem **proj_frame** {p : SplitPt D k} {n : Type} [Fintype n] [DecidableEq n] {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) {α : Frame p A n} {α₀ : Frame p S (n ⊕ Fin 1)} (h : IsSumFrame o ζ α α₀) (x : n ⊕ Fin 1 → k) : (o.jB · o.pB).map p.pr *ᵥ (α₀.b *ᵥ x) = α₀.b *ᵥ Sum.elim 0 (x ∘ Sum.inr)

theorem **proj_half** {p : SplitPt D k} {n : Type} [Fintype n] [DecidableEq n] {S A L : HSp D} (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) {α : Frame p A n} {α₀ : Frame p S (n ⊕ Fin 1)} (h : IsSumFrame o ζ α α₀) {M : Lat S} {Λ : Submodule 𝒪[k] (n → k)} (hΛ : α₀.IsLat M (lastKer Λ)) {z : Fin S.rank → AE D} (hz : z ∈ latV S M (· = p.v)) : (o.jB · o.pB) *ᵥ (p.e • z) ∈ latV S M (· = p.v)

theorem **lineProj_mem_latO** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n' : Type) [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (α : ∀ i, Frame (p i) B.A n') (α₀ : ∀ i, Frame (p i) B.V₀ (n' ⊕ Fin 1)) (M : Lat B.V₀) (LS : Lat S) (Λ : ι → Submodule 𝒪[k] (n' → k)), IsSelfDualC B.V₀ M → IsSelfDualC S LS → (∀ i, IsSumFrame B.o B.ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ x ∈ latO B.V₀ S (M : Set _) (LS : Set _) P, (B.o.resS S).symm (0, (B.o.resS S x).2) ∈ latO B.V₀ S (M : Set _) (LS : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineProjectionMemHomLatticeCompletionStatement

theorem **_root_.GlobalTheory.T.lineProjection_mem_homLatticeCompletion** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n' : Type) [Fintype n'] [DecidableEq n'] (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (α : ∀ i, Frame (p i) B.A n') (α₀ : ∀ i, Frame (p i) B.V₀ (n' ⊕ Fin 1)) (M : Lat B.V₀) (LS : Lat S) (Λ : ι → Submodule 𝒪[k] (n' → k)), IsSelfDualC B.V₀ M → IsSelfDualC S LS → (∀ i, IsSumFrame B.o B.ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ x ∈ latO B.V₀ S (M : Set _) (LS : Set _) P, (B.o.resS S).symm (0, (B.o.resS S x).2) ∈ latO B.V₀ S (M : Set _) (LS : Set _) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineProjectionMemSplitCoreCompletion

theorem **latV_le_of** {S : HSp D} (L : Lat S) {P P' : Pl D → Prop} (h : ∀ w, P w → P' w) : GlobalTheory.FinPl.latV S L P ≤ GlobalTheory.FinPl.latV S L P'

theorem **row_iA** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) : star (vecB o ζ) ᵥ* HSpace.gram S₀ ᵥ* o.iA = 0

theorem **row_vecB** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) : (star (vecB o ζ) ᵥ* HSpace.gram S₀) ·ᵥ vecB o ζ = star ζ ·ᵥ HSpace.gram L *ᵥ ζ

theorem **star_self** {L : HSp D} (ζ : Fin L.rank → D.host.E) : star (star ζ ·ᵥ HSpace.gram L *ᵥ ζ) = star ζ ·ᵥ HSpace.gram L *ᵥ ζ

theorem **sum_incA_mulVec** {s t : ℕ} (ρ : Fin s → D.host.E) (N : Matrix (Fin s) (Fin t) D.host.E) (u : Fin t → AE D) : ∑ j, incA D (ρ j) * (rat N *ᵥ u) j = ∑ a, incA D ((ρ ᵥ* N) a) * u a

theorem **e_vecB_mem** (hsum : IsSumFrame o ζ α α₀) (hlat : α₀.IsLat M (lastKer Λ)) : (fun r => q.e * incA D (vecB o ζ r)) ∈ GlobalTheory.FinPl.latV S₀ M (· = q.v)

theorem **pr_coeff_mem** (hsum : IsSumFrame o ζ α α₀) (hlat : α₀.IsLat M (lastKer Λ)) {m : Fin S₀.rank → AE D} {c : AE D} (hmdec : ∀ r, m r = (o.jA *ᵥ (o.pA *ᵥ m)) r + c * incA D (vecB o ζ r)) (hem : (fun r => q.e * m r) ∈ GlobalTheory.FinPl.latV S₀ M (· = q.v)) : q.pr c ∈ 𝒪[k]

theorem **pr_cA_mem** {P : Pl D → Prop} {a : AE D} (ha : a ∈ intAt D P) : q.pr (cA D a) ∈ 𝒪[k]

theorem **e_mul_mem** (hM : IsSelfDualC S₀ M) {P : Pl D → Prop} {m : Fin S₀.rank → AE D} (hm : m ∈ GlobalTheory.FinPl.latV S₀ M P) : (fun r => q.e * m r) ∈ GlobalTheory.FinPl.latV S₀ M (· = q.v)

theorem **pairing_eq** {S₀ A L : HSp D} (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E) (e c : AE D) {m : Fin S₀.rank → AE D} (hmdec : ∀ r, m r = (o.jA *ᵥ (o.pA *ᵥ m)) r + c * incA D (vecB o ζ r)) : (∑ a, ∑ b, cA D (e * incA D (vecB o ζ a)) * incA D (HSpace.gram S₀ a b) * m b) = cA D e * (c * incA D ((star (vecB o ζ) ᵥ* HSpace.gram S₀) ·ᵥ vecB o ζ))

theorem **lineProj_mem** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : ∀ i, Frame (p i) A n) (α₀ : ∀ i, Frame (p i) S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : ι → Submodule 𝒪[k] (n → k)), IsSelfDualC S₀ M → (∀ i, IsSumFrame o ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ m ∈ latV S₀ M P, o.jB *ᵥ (o.pB *ᵥ m) ∈ latV S₀ M P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineProjectionMemSplitCoreCompletionStatement

theorem **_root_.GlobalTheory.T.lineProjection_mem_splitCoreCompletion** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (p : ι → SplitPt D k) (I : List ι) (P : Pl D → Prop), Places p I P → ∀ (n : Type) [Fintype n] [DecidableEq n] (S₀ A L : HSp D) (o : OrthDec S₀ A L) (ζ : Fin L.rank → D.host.E), IsLineVec L ζ → ∀ (α : ∀ i, Frame (p i) A n) (α₀ : ∀ i, Frame (p i) S₀ (n ⊕ Fin 1)) (M : Lat S₀) (Λ : ι → Submodule 𝒪[k] (n → k)), IsSelfDualC S₀ M → (∀ i, IsSumFrame o ζ (α i) (α₀ i) ∧ (α₀ i).IsLat M (lastKer (Λ i))) → ∀ m ∈ latV S₀ M P, o.jB *ᵥ (o.pB *ᵥ m) ∈ latV S₀ M P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineRowTranslationHeisenbergSplitPlace

theorem **half_mul** {e : AE D} {l m n : ℕ} {u : Mx D l m} (hu : IsHalf e u) (v : Mx D m n) : IsHalf e (u · v)

theorem **mul_half** {e : AE D} {l m n : ℕ} (u : Mx D l m) {v : Mx D m n} (hv : IsHalf e v) : IsHalf e (u · v)

theorem **adj_half** {e : AE D} {m n : ℕ} {u : Mx D m n} (hu : IsHalf e u) : IsHalf (cA D e) (adj u)

theorem **ofAdj_half** {e : AE D} (S T : HSp D) {z : Mx D S.rank T.rank} (hz : IsHalf e z) : IsHalf (cA D e) (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj S T z)

theorem **xi_mul_hO** (S T : HSp D) (x : WO S T) (z : Mx D S.rank T.rank) : incA D D.host.xi0 * hO S T x (GlobalTheory.IndefiniteRowSideFullHecke.SkewGramHalfModule.ofAdj S T z) = (z · x).trace

theorem **incA_mul_ι** (a : D.host.E) (x : k) : incA D a * p.ι x = p.ι (p.pr a * x)

theorem **pr_mulVec** {m n : ℕ} (M : Matrix (Fin m) (Fin n) D.host.E) (v : Fin n → D.host.E) (r : Fin m) : p.pr ((M *ᵥ v) r) = ∑ a, p.pr (M r a) * p.pr (v a)

theorem **line_row** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (s : k), s ≠ 0 → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n) (FA' : Frame k B'.A.rank n'') (F₀' : Frame k B'.V₀.rank (n'' ⊕ Unit)), p.IsCoreFrame B s FA F₀ → p.IsCoreFrame B' (s * p.xi) FA' F₀' → ∀ u₀ : n → k, ∃ x : WO S B'.L, p.up FS F₀' (fromRows 0 (Matrix.of fun _ => u₀)) = (B'.o.resT S).symm (0, x) ∧ IsHalf (cA D p.e) ((B.o.resS S).symm (0, l.κ S x)) ∧ ∀ t : Matrix n (n' ⊕ Unit) k, ψ (symplO B.V₀ S (p.up F₀ FS t) ((B.o.resS S

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineRowTranslationHeisenbergSplitPlaceStatement

theorem **_root_.GlobalTheory.T.lineRow_translation_heisenberg_splitPlace** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ σ : ℕ) (B : BesLev D ρ) (B' : BesLev D σ) (l : Link B B'.L B'.ζ) (S : HSp D) (s : k), s ≠ 0 → ∀ (n n' n'' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] [Fintype n''] [DecidableEq n''] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n) (FA' : Frame k B'.A.rank n'') (F₀' : Frame k B'.V₀.rank (n'' ⊕ Unit)), p.IsCoreFrame B s FA F₀ → p.IsCoreFrame B' (s * p.xi) FA' F₀' → ∀ u₀ : n → k, ∃ x : WO S B'.L, p.up FS F₀' (fromRows 0 (Matrix.of fun _ => u₀)) = (B'.o.resT S).symm (0, x) ∧ IsHalf (cA D p.e) ((B.o.resS S).symm (0, l.κ S x)) ∧ ∀ t : Matrix n (n'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineVectorBesselExistence

theorem **quad_mulVec** {m n : ℕ} (A : Matrix (Fin m) (Fin n) D.host.E) (G : Matrix (Fin m) (Fin m) D.host.E) (v : Fin n → D.host.E) : star (A *ᵥ v) ⬝ᵥ G *ᵥ (A *ᵥ v) = star v ⬝ᵥ (Aᴴ * G * A) *ᵥ v

theorem **conj_mul_mul** {S S' : HSp D} {a b : ℕ} (P : Matrix (Fin S'.rank) (Fin S.rank) D.host.E) (A : Matrix (Fin S.rank) (Fin a) D.host.E) (B : Matrix (Fin S.rank) (Fin b) D.host.E) : (P * A)ᴴ * S'.gram * (P * B) = Aᴴ * (Pᴴ * S'.gram * P) * B

theorem **mat_of_isom** (W₁ W₂ : HSp D) (φ : W₁.V ≃ₗ[D.host.E] W₂.V) (hφ : ∀ x y, W₂.hT (φ x) (φ y) = W₁.hT x y) : ∃ P : Matrix (Fin W₂.rank) (Fin W₁.rank) D.host.E, Pᴴ * W₂.gram * P = W₁.gram

theorem **yc_facts** (j : ℕ) (hj : j ≤ D.R.m) : (Yc D j).SigOK ∧ (Yc D j).rank = D.R.rankY j ∧ ∀ τ, (Yc D j).sig τ = sigYD D.R j

theorem **bes_to_Yc** (j ρ : ℕ) (hj : j ≤ D.R.m) {V : HSp D} (b : BesAt V (D.sp.X j) ρ) (hr : V.rank = D.R.rankY j) (hs : V.SigOK) (hsig : ∀ τ, V.sig τ = sigYD D.R j) : ∃ b' : BesAt (Yc D j) (D.sp.X j) ρ, star b'.ζ ⬝ᵥ b'.L.gram *ᵥ b'.ζ = star b.ζ ⬝ᵥ b.L.gram *ᵥ b.ζ

theorem **jac_to_Yc** (j ρ : ℕ) (hj : j ≤ D.R.m) {V : HSp D} (b : JacAt V (D.sp.X j) ρ) (hr : V.rank = D.R.rankY j) (hs : V.SigOK) (hsig : ∀ τ, V.sig τ = sigYD D.R j) : Nonempty (JacAt (Yc D j) (D.sp.X j) ρ)

theorem **besF_zero** (b0 : BesAt (D.sp.Y 0) (D.sp.X 0) D.R.rS) (b' : ∀ j ρ, j + 2 * ρ + 2 = D.R.m → BesAt (Yc D j) (D.sp.X j) ρ) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m) : besF b0 b' 0 D.R.rS h0 = b0

theorem **besF_norm** (b0 : BesAt (D.sp.Y 0) (D.sp.X 0) D.R.rS) (b' : ∀ j ρ, j + 2 * ρ + 2 = D.R.m → BesAt (Yc D j) (D.sp.X j) ρ) (c N : D.host.E) (h0 : star b0.ζ ⬝ᵥ b0.L.gram *ᵥ b0.ζ = c) (h' : ∀ j ρ h, star (b' j ρ h).ζ ⬝ᵥ (b' j ρ h).L.gram *ᵥ (b' j ρ h).ζ * N ^ (j / 2) = c) : ∀ j ρ h, star (besF b0 b' j ρ h).ζ ⬝ᵥ (besF b0 b' j ρ h).L.gram *ᵥ (besF b0 b' j ρ h).ζ * N ^ (j / 2) = c | 0, ρ, h => by obtain rfl : D.R.rS = ρ

theorem **sbes_norm** {X : Formalism.L2.SG.HidaSetting.HidaSetting D} {h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D} {u : D.what₂.Word} {ψ : AddChar (AF D.host.F) ℂ} (B : GlobalTheory.Core.Mi.SBes X h u ψ (D.sp.X 0)) (hE : GlobalTheory.Core.Mi.IdE X) (hz : ∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = X.ιE (X.zR u (Fin.cast (GlobalTheory.Core.Mi.rank_eq X h) k))) : star B.ζ ⬝ᵥ B.L.gram *ᵥ B.ζ = -(((D.mem u).A.T : ℕ) : D.host.E)

theorem **tau_line** {n p : ℕ} (hn : n = 1) (hp : p = 1) (G : Matrix (Fin n) (Fin n) D.host.E) (G₀ : Matrix (Fin p) (Fin p) D.host.E) (ζ : Fin n → D.host.E) (ζ₀ : Fin p → D.host.E) (lam c : D.host.E) (h₀ : star ζ₀ ⬝ᵥ G₀ *ᵥ ζ₀ ≠ 0) (hrel : star lam * lam * (star ζ ⬝ᵥ G *ᵥ ζ) = c * (star ζ₀ ⬝ᵥ G₀ *ᵥ ζ₀)) : (Matrix.of fun i k => lam * ζ i / ζ₀ k)ᴴ * (G * Matrix.of fun i k => lam * ζ i / ζ₀ k) = c • G₀ ∧ (Matrix.of fun i k => lam * ζ i / ζ₀ k) *ᵥ ζ₀ = lam • ζ

theorem **exists_yRat** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ a : XGeo D, IsLineVec a.Lz a.ζ₀ → star a.ζ₀ ⬝ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → ∃ (y : YRat a) (B : GlobalTheory.Core.Mi.SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), y.Tau (-D.host.xi0 ^ (D.R.m / 2)) ∧ B.lev = (y.bes 0 D.R.rS h0).lev (a.x 0).vol ∧ B.Adapted (W.S.tr u).wt ∧ GlobalTheory.MiTop.Framed B ∧ star B.ζ ⬝ᵥ B.L.gram *ᵥ B.ζ = -(((D.mem u).A.T : ℕ) : D.host.E) ∧ ∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = W.S.X.ιE (W.S.X.zR u (Fin.cast (GlobalTheory.Core.Mi.rank_eq W.S.X W.Q.instEq) k))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineVectorBesselExistenceStatement

theorem **_root_.GlobalTheory.T.line_vector_bessel_existence** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ a : XGeo D, IsLineVec a.Lz a.ζ₀ → star a.ζ₀ ⬝ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → ∃ (y : YRat a) (B : GlobalTheory.Core.Mi.SBes W.S.X W.Q.instEq u W.Q.Rl.ψ (D.sp.X 0)) (h0 : 0 + 2 * D.R.rS + 2 = D.R.m), y.Tau (-D.host.xi0 ^ (D.R.m / 2)) ∧ B.lev = (y.bes 0 D.R.rS h0).lev (a.x 0).vol ∧ B.Adapted (W.S.tr u).wt ∧ GlobalTheory.MiTop.Framed B ∧ star B.ζ ⬝ᵥ B.L.gram *ᵥ B.ζ = -(((D.mem u).A.T : ℕ) : D.host.E) ∧ ∀ k, (B.d.i₀ *ᵥ vecB B.o B.ζ) k = W.S.X.ιE (W.S.X.zR u (Fin.cast (GlobalTheory.Core.Mi.rank_eq W.S.X W.Q.instEq) k))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineVectorPairingMemInverseIdealCompletion

theorem **intAt_zero'** (P : Pl D → Prop) : (0 : AE D) ∈ intAt D P

theorem **intAt_add'** {P : Pl D → Prop} {a b : AE D} (ha : a ∈ intAt D P) (hb : b ∈ intAt D P) : a + b ∈ intAt D P

theorem **intAt_neg'** {P : Pl D → Prop} {a : AE D} (ha : a ∈ intAt D P) : -a ∈ intAt D P

theorem **latV_entry_mem_idAt** {S : HSp D} {M : Lat S} {P : Pl D → Prop} {y : Fin S.rank → AE D} (hy : y ∈ latV S M P) (i : Fin S.rank) : y i ∈ idAt D P

theorem **idV_mem_idAt** {α : FI D} {v : Pl D} {t : AE D} (ht : t ∈ idV α v) : t ∈ idAt D (· = v)

theorem **cA_mem_idAt** {v : Pl D} {a : AE D} (h : a ∈ idAt D (· = v)) : cA D a ∈ idAt D (· = v)

theorem **pr_zero_left** (S : HSp D) (y : Fin S.rank → AE D) : pr S 0 y = 0

theorem **pr_zero_right** (S : HSp D) (w : Fin S.rank → AE D) : pr S w 0 = 0

theorem **pr_add_left** (S : HSp D) (w w' y : Fin S.rank → AE D) : pr S (w + w') y = pr S w y + pr S w' y

theorem **pr_add_right** (S : HSp D) (w y y' : Fin S.rank → AE D) : pr S w (y + y') = pr S w y + pr S w y'

theorem **pr_neg_left** (S : HSp D) (w y : Fin S.rank → AE D) : pr S (-w) y = -pr S w y

theorem **pr_neg_right** (S : HSp D) (w y : Fin S.rank → AE D) : pr S w (-y) = -pr S w y

theorem **pr_gen** (S : HSp D) (α β : Af D.host.F) (x x' : Fin S.rank → D.host.E) : pr S (fun i => pairA D.host.F 0 α ⊗ₜ[D.host.F] (1 * x i)) (fun i => pairA D.host.F 0 β ⊗ₜ[D.host.F] (1 * x' i)) = pairA D.host.F 0 (α * β) ⊗ₜ[D.host.F] (∑ m, ∑ l, star (x m) * S.gram m l * x' l)

theorem **pr_mem_intAt** {S : HSp D} {M : Lat S} (hM : IsSelfDualC S M) {P : Pl D → Prop} {w y : Fin S.rank → AE D} (hw : w ∈ latV S M P) (hy : y ∈ latV S M P) : pr S w y ∈ intAt D P

theorem **line_pair_mem_idV** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M : Lat S) (c : FI D) (v : Pl D), IsSelfDualC S M → LineId o ζ M c v → c ≠ 0 → ∀ y ∈ latV S M (· = v), cA D (∑ l, (∑ m, cA D (ratV (vecB o ζ) m) * gA S m l) * y l) ∈ idV c⁻¹ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LineVectorPairingMemInverseIdealCompletionStatement

theorem **_root_.GlobalTheory.T.lineVector_pairing_mem_inverseIdealCompletion** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S A L : HSp D) (o : OrthDec S A L) (ζ : Fin L.rank → D.host.E) (M : Lat S) (c : FI D) (v : Pl D), IsSelfDualC S M → LineId o ζ M c v → c ≠ 0 → ∀ y ∈ latV S M (· = v), cA D (∑ l, (∑ m, cA D (ratV (vecB o ζ) m) * gA S m l) * y l) ∈ idV c⁻¹ v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearIndependentOfIsParamFunV

theorem **linearIndependent_of_isParamFunV** : ∀ (D : D7) (V : HSp D) (Sb : Finset (HeightOneSpectrum (𝒪 D.host.F))) (ι : Type) (f : ι → (GA D V → ℂ)) (B : ι → GlobalTheory.LFun.Fam D.host.E), (∀ i, f i ≠ 0) → (∀ i, GlobalTheory.LFun.IsParamFunV V Sb (f i) (B i)) → (∀ i i', i ≠ i' → ∃ w ∉ GlobalTheory.LFun.above D.host.E (Sb : Set (HeightOneSpectrum (𝒪 D.host.F))), B i w ≠ B i' w) → LinearIndependent ℂ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearIndependentOfIsParamFunVStatement

theorem **_root_.GlobalTheory.T.linearIndependent_of_ne_zero_of_isParamFunV_of_separated** : ∀ (D : D7) (V : HSp D) (Sb : Finset (HeightOneSpectrum (𝒪 D.host.F))) (ι : Type) (f : ι → (GA D V → ℂ)) (B : ι → GlobalTheory.LFun.Fam D.host.E), (∀ i, f i ≠ 0) → (∀ i, GlobalTheory.LFun.IsParamFunV V Sb (f i) (B i)) → (∀ i i', i ≠ i' → ∃ w ∉ GlobalTheory.LFun.above D.host.E (Sb : Set (HeightOneSpectrum (𝒪 D.host.F))), B i w ≠ B i' w) → LinearIndependent ℂ f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearIndependentScaledBasisColumnsMatrixTranspo

theorem **linearIndependent_eX_of_dual** {D : D7} (X : HidaSetting D) : LinearIndependent D.host.E X.eX

theorem **eM_transpose_linearIndependent** : ∀ {D : D7} (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (t lam : D.host.E), t ≠ 0 → lam ≠ 0 → LinearIndependent D.host.E (GlobalTheory.Core.SBesEx.eM X h t lam).transpose

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearIndependentScaledBasisColumnsMatrixTranspoStatement

theorem **_root_.GlobalTheory.T.linearIndependent_scaledBasisColumnsMatrix_transpose** : ∀ {D : D7} (X : HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (t lam : D.host.E), t ≠ 0 → lam ≠ 0 → LinearIndependent D.host.E (GlobalTheory.Core.SBesEx.eM X h t lam).transpose

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearLawLocalModelInv

theorem **nabs_inv_of_unit** (x : kˣ) : nabs k x⁻¹ = (nabs k x)⁻¹

theorem **nabs_pos_of_unit** (x : kˣ) : 0 < nabs k x

theorem **linearFormula_inv_cancel** {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (μS μT : kˣ →* ℂˣ) (gk : GL Y' k) (hk : GL Y k) (Φ : Matrix Y Y' k → ℂ) : (((μS * μT⁻¹) (Matrix.GeneralLinearGroup.det gk) : ℂˣ) : ℂ) • om μS⁻¹ gk hk ((((μS * μT⁻¹) (Matrix.GeneralLinearGroup.det gk⁻¹) : ℂˣ) : ℂ) • om μS⁻¹ gk⁻¹ hk⁻¹ Φ) = Φ

theorem **lawLoc_inv** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (S T : HSp D) (ω : Om S T) (μS μT : kˣ →* ℂˣ) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)) (g : GA D S) (gk : GL Y' k) (h : GA D T) (hk : GL Y k), LawLoc ω μS μT Tm g gk h hk → LawLoc ω μS μT Tm g⁻¹ gk⁻¹ h⁻¹ hk⁻¹

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearLawLocalModelInvStatement

theorem **_root_.GlobalTheory.T.linearLaw_localModel_inv** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (S T : HSp D) (ω : Om S T) (μS μT : kˣ →* ℂˣ) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)) (g : GA D S) (gk : GL Y' k) (h : GA D T) (hk : GL Y k), LawLoc ω μS μT Tm g gk h hk → LawLoc ω μS μT Tm g⁻¹ gk⁻¹ h⁻¹ hk⁻¹

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearLawLocalModelMulClosed

theorem **om_smul** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ : kˣ →* ℂˣ) (g : GL a k) (h : GL b k) (c : ℂ) (Φ : Matrix b a k → ℂ) : om μ g h (c • Φ) = c • om μ g h Φ

theorem **lawLoc_mul** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (S T : HSp D) (ω : Om S T) (μS μT : kˣ →* ℂˣ) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)) (g g' : GA D S) (gk gk' : GL Y' k) (h h' : GA D T) (hk hk' : GL Y k), LawLoc ω μS μT Tm g gk h hk → LawLoc ω μS μT Tm g' gk' h' hk' → LawLoc ω μS μT Tm (g * g') (gk * gk') (h * h') (hk * hk')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearLawLocalModelMulClosedStatement

theorem **_root_.GlobalTheory.T.linearLaw_localModel_mul_closed** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (S T : HSp D) (ω : Om S T) (μS μT : kˣ →* ℂˣ) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO S T → ℂ)) (g g' : GA D S) (gk gk' : GL Y' k) (h h' : GA D T) (hk hk' : GL Y k), LawLoc ω μS μT Tm g gk h hk → LawLoc ω μS μT Tm g' gk' h' hk' → LawLoc ω μS μT Tm (g * g') (gk * gk') (h * h') (hk * hk')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearMapKillingPartialEvalIntegralKernelTransla

theorem **fill_add_single** {k : Type} [Field k] {m m' : Type} [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (ev it : Finset (m × m')) (β t : Matrix m m' k) (x : {e : m × m' // e ∈ it} → k) (e : m × m') (he : e ∈ it) (a : k) : fill ev it β t x + Matrix.single e.1 e.2 a = fill ev it β t (x + Pi.single ⟨e, he⟩ a)

theorem **continuous_fill** {k : Type} [Field k] [TopologicalSpace k] {m m' : Type} [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (ev it : Finset (m × m')) (β t : Matrix m m' k) : Continuous (fill ev it β t)

theorem **hasCompactSupport_comp_fill** {k : Type} [Field k] [TopologicalSpace k] [T2Space k] {m m' : Type} [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (ev it : Finset (m × m')) (β t : Matrix m m' k) (θ : Matrix m m' k → ℂ) (hθ : HasCompactSupport θ) : HasCompactSupport (fun x => θ (fill ev it β t x))

theorem **kerRed_translate_invariant** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (ev it : Finset (m × m')), Disjoint ev it → GlobalTheory.FlatFactor.KerRed ν ev it β l → (∀ e ∈ it, ∀ (a : k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + Matrix.single e.1 e.2 a)) → l 0' = l 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearMapKillingPartialEvalIntegralKernelTranslaStatement

theorem **_root_.GlobalTheory.T.linearMap_killing_partialEvalIntegralKernel_translation_invariant** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (ev it : Finset (m × m')), Disjoint ev it → GlobalTheory.FlatFactor.KerRed ν ev it β l → (∀ e ∈ it, ∀ (a : k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + Matrix.single e.1 e.2 a)) → l 0' = l 0)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearMapKillsPartialEvalIntegralKernelOfLaws

theorem **pullSB_apply** {X Y : Type} [TopologicalSpace X] [TopologicalSpace Y] (e : X ≃ₜ Y) (Φ : SB Y) (x : X) : (pullSB e Φ).1 x = Φ.1 (e x)

theorem **ofTrip_toTrip** (ev it : Finset (m × m')) (β t : Matrix m m' k) : ofTrip ev it β (toTrip ev it β t) = t

theorem **toTrip_ofTrip** (ev it : Finset (m × m')) (hd : Disjoint ev it) (β : Matrix m m' k) (x : (Rest ev it → k) × ({e // e ∈ ev} → k) × ({e // e ∈ it} → k)) : toTrip ev it β (ofTrip ev it β x) = x

theorem **trip_apply** (ev it : Finset (m × m')) (hd : Disjoint ev it) (β t : Matrix m m' k) : trip ev it hd β t = toTrip ev it β t

theorem **sum_single_apply** (it : Finset (m × m')) (a : m × m' → k) (r : m) (q : m') : (∑ e ∈ it, Matrix.single e.1 e.2 (a e)) r q = if (r, q) ∈ it then a (r, q) else 0

theorem **toTrip_add** (ev it : Finset (m × m')) (hd : Disjoint ev it) (β t : Matrix m m' k) (a : {e // e ∈ it} → k) : toTrip ev it β (t + ∑ e ∈ it, Matrix.single e.1 e.2 (if h : e ∈ it then a ⟨e, h⟩ else 0)) = toTrip ev it β t + (0, 0, a)

theorem **ofTrip_zero** (ev it : Finset (m × m')) (β : Matrix m m' k) (x₀ : Rest ev it → k) (a : {e // e ∈ it} → k) : ofTrip ev it β (x₀, 0, a) = fill ev it β (ofTrip ev it β (x₀, 0, 0)) a

theorem **kerRed_of_laws** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (ev it : Finset (m × m')), Disjoint ev it → (∀ (b : m × m' → k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = (ψ (∑ e ∈ ev, b e * (t e.1 e.2 - β e.1 e.2)) : ℂ) * 0.1 t) → l 0' = l 0) → (∀ (a : m × m' → k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + ∑ e ∈ it, Matrix.single e.1 e.2 (a e))) → l 0' = l 0) → GlobalTheory.FlatFactor.KerRed ν ev it β l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearMapKillsPartialEvalIntegralKernelOfLawsStatement

theorem **_root_.GlobalTheory.T.linearMap_kills_partialEvalIntegralKernel_of_laws** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k), Continuous ψ → ψ ≠ 1 → ν.IsAddHaarMeasure → ∀ (m m' : Type) [Fintype m] [DecidableEq m] [Fintype m'] [DecidableEq m'] (M : Type) [AddCommGroup M] [Module ℂ M] (l : SB (Matrix m m' k) →ₗ[ℂ] M) (β : Matrix m m' k) (ev it : Finset (m × m')), Disjoint ev it → (∀ (b : m × m' → k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = (ψ (∑ e ∈ ev, b e * (t e.1 e.2 - β e.1 e.2)) : ℂ) * 0.1 t) → l 0' = l 0) → (∀ (a : m × m' → k) (0 0' : SB (Matrix m m' k)), (∀ t, 0'.1 t = 0.1 (t + ∑ e ∈ it, Matrix.single e.1 e.2 (a e))) → l 0' = l 0) → GlobalTheory.FlatFactor.KerRed ν ev it β l

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearModelSplitPlaceCharacterEquivariance

theorem **linFirst** (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (χS χT : (AE D)ˣ →* Circle) (om : Om S T) (hW : IsWeilDatumO ψ S T χS χT om) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (α : Frame p S n') (β : Frame p T n) (LS : Lat S) (LT : Lat T) (f₀ : WO S T → ℂ) (Ψ : Matrix n n' k → ℂ) (f : WO S T → ℂ) (hO : Over ψ α β LS LT f₀ Ψ f) (g : GL n' k) (h : GL n k) : Over ψ α β LS LT f₀ (fun x => ⟨lam p χT⁻¹ ((T.rank : ℝ) / 2) (Matrix.GeneralLinearGroup.det g) : ℂ) * ⟨lam p χS (-(S.rank : ℝ) / 2) (Matrix.GeneralLinearGroup.det h) : ℂ) * Ψ ((h⁻¹ : GL n k).val * x * g.val)⟩ (om ⟨α.gl g, β.gl h⟩) f)

theorem **idl_eq** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (a : kˣ) : idl p a = p.idele a

theorem **algebraMap_inj** (D : D7) : Function.Injective (algebraMap (AF D.host.F) (AE D))

theorem **idem_calc** {R : Type} [CommRing R] (E C s t : R) (he : E * E = E) (hcc : C * C = C) (hc : E * C = 0) (hs : E * s = s) (hcs : C * t = t) : (1 - E + s) * (1 - C + t) * (1 - (E + C)) = 1 - (E + C)

theorem **norm_idele_mul** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (a : kˣ) (ε : AF D.host.F) (hε : p.e + cA D p.e = algebraMap (AF D.host.F) (AE D) ε) : Algebra.norm (AF D.host.F) (p.idele a : AE D) * (1 - ε) = 1 - ε

theorem **normOff** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (a : kˣ) (w : Pl D), w ≠ p.v → GlobalTheory.IndefiniteRowSideFullHecke.SplitLinearModelNorm.normV D w (p.idele a) = 1

theorem **norm_idele_chart** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (π : AE D →+* Fv D.host.F p.v) (halg : ∀ y : AF D.host.F, π (algebraMap (AF D.host.F) (AE D) y) = evalV D.host.F p.v (sndA D.host.F y)) (πe : π p.e = 1) (πce : π (cA D p.e) = 0) (a : kˣ) : ((GlobalTheory.IndefiniteRowSideFullHecke.SplitLinearModelNorm.normV D p.v (p.idele a) : (Fv D.host.F p.v)ˣ) : Fv D.host.F p.v) = π (p.sec a)

theorem **chartAt** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k), ∃ φ : k ≃+* Fv D.host.F p.v, (∀ a, Valued.v (φ a) ≤ 1 ↔ a ∈ 𝒪[k]) ∧ ∀ a : kˣ, ((GlobalTheory.IndefiniteRowSideFullHecke.SplitLinearModelNorm.normV D p.v (p.idele a) : (Fv D.host.F p.v)ˣ) : Fv D.host.F p.v) = φ a

theorem **absAt** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : GlobalTheory.LatPair.Unif k), GlobalTheory.IndefiniteRowSideFullHecke.SplitLinearModelNorm.absV D.host.F p.v (GlobalTheory.IndefiniteRowSideFullHecke.SplitLinearModelNorm.normV D p.v (p.idele q.unit)) = ((resCard k : ℕ) : ℝ)⁻¹

theorem **absFin_unif** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : GlobalTheory.LatPair.Unif k) : GlobalTheory.IndefiniteRowSideFullHecke.SplitLinearModelNorm.absFin D (idl p q.unit) = ((resCard

k : ℕ) : ℝ)⁻¹

theorem **lam_unif** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : SplitPt D k) (q : GlobalTheory.LatPair.Unif k) (χ : (AE D)ˣ →* Circle) (r : ℝ) : ((lam p χ r q.unit : ℂˣ) : ℂ) = (((resCard k : ℝ) ^ (-r) : ℝ) : ℂ) * (χ (p.idele q.unit) : ℂ)

theorem **linExact** : ∀ D : D7, GlobalTheory.FinPl.LinExact D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearModelSplitPlaceCharacterEquivarianceStatement

theorem **_root_.GlobalTheory.T.linearModel_splitPlace_characterEquivariance** : ∀ D : D7, GlobalTheory.FinPl.LinExact D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearModelSplitPlaceTwistSchwartzBruhat

theorem **triv_eq_ofLocal** (Pg : Finset (Pl D.host.E)) (r : ℕ) : LevelP.triv Pg r ℂ = LevelP.ofLocal (fun _ => ⊥) (fun _ => 1)

theorem **detAt_exists** {v : FinPl D} (φ : SplitProj D v) {e : AE D} (he : IsIdemOf φ e) (V : HSp D) (B : GL (Fin V.rank) (Fv D.host.F v)) (g : GA D V) : ∃ a : (AE D)ˣ, IsDetAt e V g a ∧ (1 - e) * (a : AE D) = 1 - e ∧ rdA φ (a : AE D) = ((Matrix.GeneralLinearGroup.det (grpCo φ V B g) : (Fv D.host.F v)ˣ) : Fv D.host.F v)

theorem **mem_GfC_of_gAt** {v : FinPl D} (φ : SplitProj D v) {e : AE D} (he : IsIdemOf φ e) (V : HSp D) (g : GA D V) (hg : gAt V (.inl v) g = g) : g ∈ GfC V

theorem **splitFac_coe** (χS χT : (AE D)ˣ →* Circle) (a b : (AE D)ˣ) : ((splitFacO χS χT a b : Circle) : ℂ) = (((Circle.toUnits (χT a))⁻¹ * Circle.toUnits (χS b) : ℂˣ) : ℂ)

theorem **placeModel_isometry_borel** (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (θ : InfFam D) (S T : HSp D) (sS : D.what₂.Tau → Matrix (Fin S.rank) (Fin S.rank) ℂ) (sT : D.what₂.Tau → Matrix (Fin T.rank) (Fin T.rank) ℂ) (ΛS : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) (ΛT : Submodule (𝒪 D.host.E) (Fin T.rank → D.host.E)) (f₀ : WO S T → ℂ) (μ : Measure (WO S T)) (ℱ : Set (WO S T)) (hb : IsBase ψ θ S T sS sT ΛS ΛT f₀) (hc : IsCovolOne (ratL T S) μ ℱ) (v : FinPl D) (a : AE D) (ha : IsSplitIdem D a) (m n : ℕ) (κ : WO S T → Matrix (Fin m) (Fin n) (Fv D.host.F v)) (hκ : IsCoord a (latO S T ΛS ΛT fun _ => True) κ) (Tm : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ)) (hT : IsPlaceModel ψ (symplO S T) (heisO ψ S T) a (latO S T ΛS ΛT fun _ => True) 1 f₀ Tm) [inst : MeasurableSpace (Fv D.host.F v)] [hbor : BorelSpace (Fv D.host.F v)] (ν : Measure (Fv D.host.F v)) (hν : ν.IsAddHaarMeasure) : GlobalTheory.SplitWeil

theorem **GlobalTheory.TwistLinModel.linearModel_at_twist** : ∀ D : D7, ForStepQ D fun c w k s s' p => ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ ι ι', IsChart s k φ ι → IsChart s' (k + 1) φ ι' → ∀ μS μT ξl : (Fv D.host.F v)ˣ →* ℂˣ, (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μS y = Circle.toUnits (c.chS (k + 1) x)) → (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μT y = Circle.toUnits (c.chT (k + 1) x)) → LiftsChar φ PP ξl (w.ξ PP) → ∀ P m X, IsCovolOne (ratL s'.V s.V) m X → ∃ (hW : ∀ g h, ∀ u ∈ GlobalTheory.SplitWeil.SB (Matrix (Fin D.R.r ⊕ Blocks (k + 1) D.R.t) (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v)), GlobalTheory.ParHecke.omTwo μS μT g h u ∈ GlobalTheory.SplitWeil.SB (Matrix (Fin D.R.r ⊕ Blocks

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearModelSplitPlaceTwistSchwartzBruhatStatement

theorem **_root_.GlobalTheory.T.linearModel_splitPlace_twist_schwartzBruhat** : ∀ D : D7, ForStepQ D fun c w k s s' p => ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ ι ι', IsChart s k φ ι → IsChart s' (k + 1) φ ι' → ∀ μS μT ξl : (Fv D.host.F v)ˣ →* ℂˣ, (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μS y = Circle.toUnits (c.chS (k + 1) x)) → (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μT y = Circle.toUnits (c.chT (k + 1) x)) → LiftsChar φ PP ξl (w.ξ PP) → ∀ P m X, IsCovolOne (ratL s'.V s.V) m X → ∃ (hW : ∀ g h, ∀ u ∈ GlobalTheory.SplitWeil.SB (Matrix (Fin D.R.r ⊕ Blocks (k + 1) D.R.t) (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v)), GlobalTheory.ParHecke.omTwo μS μT g h u ∈ GlobalTheory.SplitWeil.SB (Matrix (F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearModelTwistPlaceInThetaFunctions

theorem **GlobalTheory.ChainTw.linModel_theta** : ∀ D : D7, ForStepQ D fun c w k s s' p => ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ ι ι', IsChart s k φ ι → IsChart s' (k + 1) φ ι' → ∀ μS μT ξl : (Fv D.host.F v)ˣ →* ℂˣ, (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μS y = Circle.toUnits (c.chS (k + 1) x)) → (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μT y = Circle.toUnits (c.chT (k + 1) x)) → LiftsChar φ PP ξl (w.ξ PP) → ∀ P, ∃ (W : Submodule ℂ (Matrix (Fin D.R.r ⊕ Blocks (k + 1) D.R.t) (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) → ℂ)) (hW : ∀ g h, ∀ u ∈ W, GlobalTheory.ParHecke.omTwo μS μT g h u ∈ W) (hΦ : datum ξl⁻¹ ∈ W) (T : W →ₗ[ℂ] (WO s.V s'.V → ℂ)), T ⟨_, hΦ⟩ = p.pt P ∧ (∀ u, IsThetaFunO c.ψ s.V

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearModelTwistPlaceInThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.linearModel_twistPlace_in_thetaFunctions** : ∀ D : D7, ForStepQ D fun c w k s s' p => ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ ι ι', IsChart s k φ ι → IsChart s' (k + 1) φ ι' → ∀ μS μT ξl : (Fv D.host.F v)ˣ →* ℂˣ, (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μS y = Circle.toUnits (c.chS (k + 1) x)) → (∀ (x : (AE D)ˣ) (y : (Fv D.host.F v)ˣ), (∃ e, IsIdemOf φ e ∧ (1 - e) * (x : AE D) = 1 - e ∧ rdA φ (x : AE D) = y) → μT y = Circle.toUnits (c.chT (k + 1) x)) → LiftsChar φ PP ξl (w.ξ PP) → ∀ P, ∃ (W : Submodule ℂ (Matrix (Fin D.R.r ⊕ Blocks (k + 1) D.R.t) (Fin D.R.r ⊕ Blocks k D.R.t) (Fv D.host.F v) → ℂ)) (hW : ∀ g h, ∀ u ∈ W, GlobalTheory.ParHecke.omTwo μS μT g h u ∈ W) (hΦ : datum ξl⁻¹ ∈ W) (T : W →ₗ[ℂ] (WO s.V s'.V → ℂ)), T ⟨_, hΦ⟩ = p.pt P ∧

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearWeilActionSupportAndPeriodIntegral

theorem **om_support_period** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A B : Type} [Fintype A] [DecidableEq A] [Fintype B] [DecidableEq B] (ϖ : kˣ), valuation k (ϖ : k) < 1 → ∀ (i : ℕ) (μ : kˣ →* ℂˣ) (Φ : Matrix B A k → ℂ), (∀ x, Φ x ≠ 0 → IntM x) → (∀ x y, MaxM y → Φ (x + y) = Φ x) → ∀ (g : GL A k) (g' : GL B k), IntM (g⁻¹ : GL A k).val → IntM ((ϖ : k) ^ i • g.val) → IntM (g'⁻¹ : GL B k).val → IntM ((ϖ : k) ^ (i + 1) • g'.val) → (∀ x, om μ g g' Φ x ≠ 0 → ∀ r s, valuation k (x r s) ≤ valuation k (((ϖ ^ (i + 1))⁻¹ : kˣ) : k)) ∧ (∀ x y, (∀ r s, valuation k (y r s) ≤ valuation k ((ϖ : k) ^ (i + 1))) → om μ g g' Φ (x + y) = om μ g g' Φ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinearWeilActionSupportAndPeriodIntegralStatement

theorem **_root_.GlobalTheory.T.linearWeilAction_support_and_period_integral** : ∀ {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {A B : Type} [Fintype A] [DecidableEq A] [Fintype B] [DecidableEq B] (ϖ : kˣ), valuation k (ϖ : k) < 1 → ∀ (i : ℕ) (μ : kˣ →* ℂˣ) (Φ : Matrix B A k → ℂ), (∀ x, Φ x ≠ 0 → IntM x) → (∀ x y, MaxM y → Φ (x + y) = Φ x) → ∀ (g : GL A k) (g' : GL B k), IntM (g⁻¹ : GL A k).val → IntM ((ϖ : k) ^ i • g.val) → IntM (g'⁻¹ : GL B k).val → IntM ((ϖ : k) ^ (i + 1) • g'.val) → (∀ x, om

μ g g' Φ x ≠ 0 → ∀ r s, valuation k (x r s) ≤ valuation k (((ϖ ^ (i + 1))⁻¹ : kˣ) : k)) ∧ (∀ x y, (∀ r s, valuation k (y r s) ≤ valuation k ((ϖ : k) ^ (i + 1))) → om μ g g' Φ (x + y) = om μ g g' Φ x)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinkLawIffPeriodLinkQuasi

theorem **heisO_heisO** (S T : HSp D) (x y : WO S T) (F : WO S T → ℂ) : heisO ψ S T x (heisO ψ S T y F) = ψ (symplOH S T x y) • heisO ψ S T (x + y) F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinkZeroOfChartZero

theorem **besLev_per_congr** {D : D7} {ρ : ℕ} {B B' : BesLev D ρ} (h : B = B') (ψ) {φ : GA D B.A → ℂ} {φ' : GA D B'.A → ℂ} {g : GA D B.V → ℂ} {g' : GA D B'.V → ℂ} (h1 : HEq φ φ') (h2 : HEq g g') : B.per ψ φ g = B'.per ψ φ' g'

theorem **linkZero_of_chartZero** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), ∀ u (f : FamCh W u) (d : BotDat W u), ChartZero f → (∀ h₀ : 0 + 2 * D.R.rS + 2 = D.R.m, d.B.lev = f.c.B 0 D.R.rS h₀) → d.b = (show GA D (D.sp.Y 0) from f.g.b 0) → LinkZero f d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LinkZeroOfChartZeroStatement

theorem **_root_.GlobalTheory.T.link_zero_of_chart_zero** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex), ∀ u (f : FamCh W u) (d : BotDat W u), ChartZero f → (∀ h₀ : 0 + 2 * D.R.rS + 2 = D.R.m, d.B.lev = f.c.B 0 D.R.rS h₀) → d.b = (show GA D (D.sp.Y 0) from f.g.b 0) → LinkZero f d

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LintegralBesselGroupHaarEqConstMulProd

theorem **iota_prodEquiv** {D : D7} (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) {u : D.what₂.Word} {ψ : AddChar (AF D.host.F) ℂ} {A : HSp D} (B : SBes X h u ψ A) (p : Formalism.L2.SG.Pf.SinfSub X.B.I X.eR (X.zR u) × Formalism.L2.SG.Pf.SfinSub X.B.I X.eR (X.zR u)) : iotaX X h (prodEquiv X h B p).1 = ((p.1 : X.B.I.Ginf), (p.2 : X.B.I.Gf))

theorem **lintegral_besGrp_eq** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (μSf : Measure (Formalism.L2.SG.Pf.SfinSub X.B.I X.eR (X.zR u))), μSf.IsHaarMeasure → SigmaFinite μSf → IsHaarOnSet X.B.I (X.SinfU u) (((X.muN u).prod (X.muH u)).map (fun p => p.1 * p.2)) → ∃ c : ℝ, 0 < c ∧ ∀ Φ : X.B.I.GA → ENNReal, Measurable Φ → ∫⁻ s, Φ (iotaX X h s.1) ∂B.lev.μR = ENNReal.ofReal c * ∫⁻ a, ∫⁻ t, Φ (a, t.1) ∂μSf ∂(((X.muN u).prod (X.muH u)).map (fun p => p.1 * p.2))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LintegralBesselGroupHaarEqConstMulProdStatement

theorem **_root_.GlobalTheory.T.lintegral_besselGroup_haar_eq_const_mul_prod** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (μSf : Measure (Formalism.L2.SG.Pf.SfinSub X.B.I X.eR (X.zR u))), μSf.IsHaarMeasure → SigmaFinite μSf → IsHaarOnSet X.B.I (X.SinfU u) (((X.muN u).prod (X.muH u)).map (fun p => p.1 * p.2)) → ∃ c : ℝ, 0 < c ∧ ∀ Φ : X.B.I.GA → ENNReal, Measurable Φ → ∫⁻ s, Φ (iotaX X h s.1) ∂B.lev.μR = ENNReal.ofReal c * ∫⁻ a, ∫⁻ t, Φ (a, t.1) ∂μSf ∂(((X.muN u).prod (X.muH u)).map (fun p => p.1 * p.2))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LintegralBesselGroupUnfoldFundamentalDomain

theorem **lintegral_unfold** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (𝒮 : Set (besGrp B.d B.o)), IsFundamentalDomain ((GQ (D.sp.Y 0)).subgroupOf (besGrp B.d B.o)) 𝒮 B.lev.μR → ∀ f : X.B.I.GA → ENNReal, Measurable f → ∫⁻ s, f (iotaX X h s.1) ∂B.lev.μR = ∫⁻ s in 𝒮, ∑' σ : Srat X u, f ((σ : X.B.I.GA) * iotaX X h s.1) ∂B.lev.μR

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LintegralBesselGroupUnfoldFundamentalDomainStatement

theorem **_root_.GlobalTheory.T.lintegral_besselGroup_unfold_fundamental_domain** : ∀ (D : D7) (X : Formalism.L2.SG.HidaSetting.HidaSetting D) (h : X.B.I = Formalism.L2.SG.InstExistsPf.instOf D) (u : D.what₂.Word) (ψ : AddChar (AF D.host.F) ℂ) (A : HSp D) (B : SBes X h u ψ A) (𝒮 : Set (besGrp B.d B.o)), IsFundamentalDomain ((GQ (D.sp.Y 0)).subgroupOf (besGrp B.d B.o)) 𝒮 B.lev.μR → ∀ f : X.B.I.GA → ENNReal, Measurable f → ∫⁻ s, f (iotaX X h s.1) ∂B.lev.μR = ∫⁻ s in 𝒮, ∑' σ : Srat X u, f ((σ : X.B.I.GA) * iotaX X h s.1) ∂B.lev.μR

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LintegralUnfoldSumTransitiveOrbit

theorem **lintegral_integral_unfold_transitive** : ∀ (H : Type) [Group H] [MeasurableSpace H] [MeasurableMul H] (ν : Measure H) [ν.IsMulLeftInvariant] (Γ : Subgroup H) [Countable Γ] (Y : Type) [Countable Y] [MulAction Γ Y] (y₀ : Y), (∀ y, ∃ γ : Γ, γ • y₀ = y) → ∀ (𝓕 𝓖 : Set H), IsFundamentalDomain Γ 𝓕 ν → IsFundamentalDomain ((MulAction.stabilizer Γ y₀).map Γ.subtype) 𝓖 ν → ∀ Φ : Y → H → ℂ, (∀ y, Measurable (Φ y)) → (∀ (γ : Γ) y h, Φ (γ • y) ((γ : H) * h) = Φ y h) → (∫⁻ h in 𝓖, ‖Φ y₀ h‖ₑ ∂ν = ∫⁻ h in 𝓕, ∑' y, ‖Φ y h‖ₑ ∂ν) ∧ (∫⁻ h in 𝓕, ∑' y, ‖Φ y h‖ₑ ∂ν ≠ ⊤ → ∫ h in 𝓖, Φ y₀ h ∂ν = ∫ h in 𝓕, ∑' y, Φ y h ∂ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LintegralUnfoldSumTransitiveOrbitStatement

theorem **_root_.GlobalTheory.T.lintegral_unfold_sum_transitive_orbit** : ∀ (H : Type) [Group H] [MeasurableSpace H] [MeasurableMul H] (ν : Measure H) [ν.IsMulLeftInvariant] (Γ : Subgroup H) [Countable Γ] (Y : Type) [Countable Y] [MulAction Γ Y] (y₀ : Y), (∀ y, ∃ γ : Γ, γ • y₀ = y) → ∀ (𝓕 𝓖 : Set H), IsFundamentalDomain Γ 𝓕 ν → IsFundamentalDomain ((MulAction.stabilizer Γ y₀).map Γ.subtype) 𝓖 ν → ∀ Φ : Y → H → ℂ, (∀ y, Measurable (Φ y)) → (∀ (γ : Γ) y h, Φ (γ • y) ((γ : H) * h) = Φ y h) → (∫⁻ h in 𝓖, ‖Φ y₀ h‖ₑ ∂ν = ∫⁻ h in 𝓕, ∑' y, ‖Φ y h‖ₑ ∂ν) ∧ (∫⁻ h in 𝓕, ∑' y, ‖Φ y h‖ₑ ∂ν ≠ ⊤ → ∫ h in 𝓖, Φ y₀ h ∂ν = ∫ h in 𝓕, ∑' y, Φ y h ∂ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ListConvolutionDistinctPlacesSupportedAndComposi

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ι : Type) (v : ι → GlobalTheory.FinPl.Pl D) (c : ι → WO S T →₀ ℂ), Function.Injective v → (∀ i, GlobalTheory.FinPl.CombAt (· = v i) (c i)) → ∀ L : List ι, L.Nodup → GlobalTheory.FinPl.CombAt (fun p => ∃ i ∈ L, v i = p) (aconv (L.map c)) ∧ ∀ f, comb ψ S T (aconv (L.map c)) f = L.foldr (fun i g => comb ψ S T (c i) g) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ListConvolutionDistinctPlacesSupportedAndComposiStatement

theorem **_root_.GlobalTheory.T.listConvolution_distinctPlaces_supported_and_composition** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) (ι : Type) (v : ι → GlobalTheory.FinPl.Pl D) (c : ι → WO S T →₀ ℂ), Function.Injective v → (∀ i, GlobalTheory.FinPl.CombAt (· = v i) (c i)) → ∀ L : List ι, L.Nodup → GlobalTheory.FinPl.CombAt (fun p => ∃ i ∈ L, v i = p) (aconv (L.map c)) ∧ ∀ f, comb ψ S T (aconv (L.map c)) f = L.foldr (fun i g => comb ψ S T (c i) g) f

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.ListProdMapNeZero

private theorem **list_prod_map_pow_eq** {α : Type} (l : List α) (f g : α → ℂ) (n : ℕ) (h : ∀ a, f a ^ n = g a ^ n) : (l.map f).prod ^ n = (l.map g).prod ^ n

private theorem **list_prod_map_ne_zero** {α : Type} (l : List α) (f : α → ℂ) (h : ∀ a, f a ≠ 0) : (l.map f).prod ≠ 0

theorem **mono_pow_eq** {D : D7} {c : Comm D} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} (o : TDat c k ι) (r : ι → ℝ) (e : ι → ℤ) (χ χ' : (AE D)ˣ →* Circle) (n : ℕ) (hn : χ ^ n = χ' ^ n) : o.mono r e χ ^ n = o.mono r e χ' ^ n

theorem **mono_ne_zero** {D : D7} {c : Comm D} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} (o : TDat c k ι) (r : ι → ℝ) (e : ι → ℤ) (χ : (AE D)ˣ →* Circle) (hr : ∀ i, 0 < r i) : o.mono r e χ ≠ 0

theorem **mono_unitRatio** : ∀ (D : D7) (c : Comm D) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (o : TDat c k ι) (r : ι → ℝ) (e : ι → ℤ) (χ χ' : (AE D)ˣ →* Circle) (S : ℕ → Prop), (∀ i, 0 < r i) → (∃ n : ℕ, 0 < n ∧ χ ^ n = χ' ^ n) → UnitRatio S (o.mono r e χ) (o.mono r e χ')

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiveChartsKernelEqClassSumJets

theorem **kval** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → JetS I l Z R.ψ R.θ s → (∀ j, j ≤ B.R.m → (g.x j).Is) → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LiveChartsKernelEqClassSumJetsStatement

theorem **_root_.GlobalTheory.T.liveCharts_kernel_eq_classSum_jets** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → JetS I l Z R.ψ R.θ s → (∀ j, j ≤ B.R.m → (g.x j).Is) → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LivechartKernelVectorsRealPointsDetPowers

theorem **castP_mvZ** {D : D7} {p q a b : ℕ} (hp : p = a) (hq : q = b) (M : D.what₂.Tau → Matrix (Fin p) (Fin p) ℂ) (M' : D.what₂.Tau → Matrix (Fin q) (Fin q) ℂ) (P : PolF D q p) : castP (D

theorem **comb_smul** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : WO S T →₀ ℂ) (x : ℂ) (f : WO S T → ℂ) : comb ψ S T c (x • f) = x • comb ψ S T c f

theorem **GlobalTheory.KJRealLive.clsKL_real** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → JetS I l Z R.ψ R.θ s → (∀ j, j ≤ B.R.m → (g.x j).Is) → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LivechartKernelVectorsRealPointsDetPowersStatement

theorem **_root_.GlobalTheory.T.livechart_kernelVectors_realPoints_detPowers** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} (I : TC B X), I.Ok → TC.Pins B X → ∀ (l : I.Laws) (Z : TC.Later B (I.cols l)), ∀ (u : B.Word) {G : Type} [Group G] [Fintype G] {Pl0 T : Type} {F : FamIn (I.rec7 l Z).what₂.LL G Pl0 T (I.rec7 l Z).R.m} {Ks : Type} [Field Ks] [NumberField Ks] [Algebra Ks (I.rec7 l Z).what₂.LL] (R : RealDat (I.rec7 l Z) F Ks) (g : GlobalTheory.Fam.Geo (I.rec7 l Z)) (c₀ s : ℕ → ℂ) J, PinX I l Z R.ψ R.θ g.x g.vac c₀ → JetS I l Z R.ψ R.θ s → (∀ j, j ≤ B.R.m → (g.x j).Is) → ∀ (P : LevelP R.Pg (I.rec7 l Z).R.r ℂ) (t : T) (r : RowCh R P u g t) (a : I.ι) (k : ColCh (D

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocXiEqMulMidXi

theorem **iter_le** {r t : Type} [Fintype r] [DecidableEq r] [Fintype t] [DecidableEq t] (P : TypeDatum k r) {μ μ' : ℕ → kˣ →* ℂˣ} (h : ∀ j, IsTame (μ j)) (h' : ∀ j, IsTame (μ' j)) (n : ℕ) (e : ∀ j, 1 ≤ j → j ≤ n → μ j = μ' j) (j : ℕ) (hj : j ≤ n) : P.iter t μ h j = P.iter t μ' h' j

theorem **withMu_le** {s κ : ℕ} (S : SymData k s κ) (μ : ℕ → kˣ →* ℂˣ) (h : ∀ j, IsTame (μ j)) (e : ∀ j, 1 ≤ j → j ≤ s + 2 → μ j = S.μ j) (t : Type) [Fintype t] [DecidableEq t] : ((withMu S μ h).T t).Phi = (S.T t).Phi ∧ ((withMu S μ h).T t).PhiTop = (S.T t).PhiTop ∧ ((withMu S μ h).T t).lev = (S.T t).lev

theorem **exists_normal** {s κ : ℕ} (S : SymData k s κ) (ϖ' : kˣ) (h : valuation k ((ϖ' : k) − (S.ϖ : k)) < valuation k (S.ϖ : k)) (τ : ℕ) (ν : Measure k) : ∃ S' : SymData k s κ, S'.ϖ = ϖ' ∧ S'.ψ = S.ψ ∧ (∀ j, j ≤ s + 2 → S'.μ j = S.μ j) ∧ (∀ j, s + 2 < j → S'.μ j = 1) ∧ SameData S' S τ ν

theorem **crd_inl** {s κ : ℕ} (S : SymData k s κ) {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c c' : ℕ} (x : Matrix (n ⊕ Fin c) (n' ⊕ Fin c') k) (r s') : ∃ d : k, GlobalTheory.EntryBridge.crd S x r s' = d * x r s' ∧ ((∃ a b, r = .inl a ∧ s' = .inl b) → d = 1)

theorem **entry_mul_KOne** {s κ : ℕ} (S : SymData k s κ) (t : Type) [Fintype t] [DecidableEq t] (ν : Measure k) (j c : ℕ) (H : j + 1 + c = s) {a : GL (XIx (Fin κ) t (j + 1)) k} (ha : a ∈ KOne k _) {b : GL (XIx (Fin κ) t j) k} (hb : b ∈ KOne k _) (x) : (S.T t).entry S.ψ ν j c H 1 1 (a.val * x * b.val) = (S.T t).entry S.ψ ν j c H 1 1 x

theorem **psi_tie** (q : GlobalTheory.FinPl.SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (hv : q.v = PP.under (𝒪 D.host.F)) (hi : ∀ x, (ofSplit q).pr x ∈ 𝒪[k] ↔ IntAt PP x) (hd : DenseRange (ofSplit q).pr) (hp : (qOf PP).Prime) (hsq : ((qOf PP : ℕ) : 𝒪 D.host.E) ∉ PP.asIdeal ^ 2) (hxi : IntAt PP D.host.xi0) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (−1)) {K : Type} [Field K] (σ₀ : K →+* ℂ) (ψ₀ : AddChar (𝒪 D.host.E ⧸ PP.asIdeal) K) (hW : ∀ (z : 𝒪 D.host.E ⧸ PP.asIdeal) (b : ℤ), redP PP D.host.xi0 * z = b → σ₀ (ψ₀ z) = Complex.exp (2 * Real.pi * Complex.I * b / (qOf PP : ℂ))) (ψk : AddChar k Circle) (hch : (ofSplit q).IsChar ψ ψk) (a : D.host.E) (ha : IntAt PP a) : (ψk ((q.pr.comp (incA D)) (a / (qOf PP : D.host.E))) : ℂ) = σ₀ (ψ₀ (redP PP a))

theorem **chi_tie** (q : GlobalTheory.FinPl.SplitPt D k) {PP : Formalism.L2.DIV.Pl D.host.E} (hid : Formalism.L2.NTheta.IsIdemAt q.e PP) (hi : ∀ x, (ofSplit q).pr x ∈ 𝒪[k] ↔ IntAt PP x) (hd : DenseRange (ofSplit q).pr) (μ : ℕ → (AE D)ˣ →* Circle) (η : (AE D)ˣ →* Circle) (j : ℕ) {H Hη : Formalism.L2.Host.HChar D.host.E} {c : (Formalism.L2.DIV.kP PP)ˣ →* ℂˣ} (h1 : ChrEq (μ j) H) (h2 : IsRes H PP c⁻¹) (h3 : ChrEq η Hη) (h4 : IsRes Hη PP 1) (m : kˣ →* ℂˣ) (hm : (ofSplit q).IsComp (Lift.chrStep μ η j).1⁻¹ m) (a : D.host.E) (u : kˣ) (r : (Formalism.L2.DIV.kP PP)ˣ) (ha : IntAt PP a) (hr : (r : Formalism.L2.DIV.kP PP) = redP PP a) (hu : (u : k) = (ofSplit q).pr a) : m u = c r

theorem **exists_KOne** (ι : E →+* k) {Y : Type} [Fintype Y] [DecidableEq Y] {a n : ℕ} (h : n = a) (F : Frame k n Y) (e : Y ≃ Fin n) (hF : F.β = (1 : Matrix (Fin n) (Fin n) k).submatrix id e) (f : (Fin a → E) ≃ₗ[E] (Fin a → E)) (hf : ∀ r s, valuation k (ι (f (Pi.single s 1) r − if r = s then 1 else 0)) < 1) : ∃ g ∈ KOne k Y, g.val = F.γ ∗ Matrix.of fun r c => ι (f (Pi.single (e.trans (finCongr h) c) 1) (finCongr h r))

theorem **toLoc_inFr** (ι : E →+* k) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {a b nS nT : ℕ} (hS : nS = a) (hT : nT = b) (FS : Frame k nS Y') (FT : Frame k nT Y) (e' : Y' ≃ Fin nS) (e : Y ≃ Fin nT) (hFS : FS.β = (1 : Matrix (Fin nS) (Fin nS) k).submatrix id e') (hFT : FT.β = (1 : Matrix (Fin nT) (Fin nT) k).submatrix id e) (f : (Fin a → E) ≃ₗ[E] (Fin a → E)) (f' : (Fin b → E) ≃ₗ[E] (Fin b → E)) (hf : ∀ r s, valuation k (ι (f (Pi.single s 1) r − if r = s then 1 else 0)) < 1) (hf' : ∀ r s, valuation k (ι (f' (Pi.single s 1) r − if r = s then 1 else 0)) < 1) : ∃ g ∈ KOne k Y, ∃ g' ∈ KOne k Y', ∀ z : Matrix (Fin nT) (Fin nS) E, toLoc ι (e.trans (finCongr hT)) (e'.trans (finCongr hS)) (Formalism.L2.Col.inFr f f' (Matrix.toLin' (z.reindex (finCongr hT) (finCongr hS)))) = g.val ∗ (FT.γ ∗ z.map ι ∗ FS.β) ∗ g'.val

theorem **xiAt_tw_eq** {X : Fields} {Pr : Primes X} {R : Ranks} (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) {S : Set ℕ} (mu mu' : Formalism.L2.DIV.Tup X.F X.E X.K S) (G : ChGlob X R.m) (P : rowSG Pr sel mu) (P' : rowSG Pr sel mu') (h : P.1 = P'.1) (hP : P.1 ∈ Pr.vGs) (i : ℕ) : XiAt Ch sel mu (twCst Ch sel mu G) i P 1 1 = XiAt Ch sel mu' (twCst Ch sel mu' G) i P' 1 1

theorem **exists_atType** (hW : AllTrue W) (a : GeoAll W u) {row col} (hR : GRead (a.fam row col) a.G a.pl) (n : Fin a.n) (hϖ : valuation (a.G n).k ((((a.G n).S.ϖ : (a.G n).kˣ) : (a.G n).k) − ((qOf (a.pl n).1 : ℕ) : (a.G n).k)) < valuation (a.G n).k (((a.G n).S.ϖ : (a.G n).kˣ) : (a.G n).k)) : ∃ S' : SymData (a.G n).k (a.G n).s B.R.k, SameData S' (a.G n).S B.R.t (a.G n).ν ∧ ∀ (b : I.ι) (P : BadPr I b), P.1 = (a.pl n).1 → AtType I.Ch I.sel (I.tup b) (I.cst b) B.embLL P ((a.G n).q.pr.comp (incA (I.rec7 l Z))) S'

theorem **frame_cong** (hI : I.Ok) {b : I.ι} {j : ℕ} (p : Formalism.L2.Col.ChPt (Pr

theorem **midXi_isLocFac** {a : GeoAll W u} (ha : a.Is) (hat : a.True) {row col} (hR : GRead (a.fam row col) a.G a.pl) (n : Fin a.n) (j c : ℕ) (H : j + 1 + c = (a.G n).s) : IsLocFac (I.rec7 l Z) (pk W).ψ _ _ (a.pl n).1 ((a.G n).cg j) (midXi a n j c H)

theorem **locXi_eq_mul_midXi** (hW : AllTrue W) {a : GeoAll W u} (ha : a.Is) {row col} (hR : GRead (a.fam row col) a.G a.pl) (n : Fin a.n) (hϖ : valuation (a.G n).k ((((a.G n).S.ϖ : (a.G n).kˣ) : (a.G n).k) − ((qOf (a.pl n).1 : ℕ) : (a.G n).k)) < valuation (a.G n).k (((a.G n).S.ϖ : (a.G n).kˣ) : (a.G n).k)) (j c : ℕ) (H : j + 1 + c = (a.G n).s) (hj : j + 1 ≤ B.R.m) : ∃ C : ℂ, C ≠ 0 ∧ ∀ t (J : QJ a t), J.Reads → ∀ (α : J.Ch j) (γ : J.Ch (j + 1)) (P : BadPr I (W.P.I.tup t)), P.1 = (a.pl n).1 → ∀ z, B.embLL (locXi J.A j (J.ι j α) (J.ι (j + 1) γ) P (homOf I l Z j hj z)) = C ∗ midXi a n j c H z

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalBimodelTranslation

theorem **GlobalTheory.TriStep.valK_gl₂** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (g : GL Q.n₂ Q.k) (Ξ₁ : SB (Matrix Q.n₁ Q.n₀ Q.k)) (Ξ₂ Ξ₂' : SB (Matrix Q.n₂ Q.n₁ Q.k)), Ξ₂'.1 = om Q.μ₂ 1 g Ξ₂.1 → IsBiTheta ψ S.B.A S.J.W₀ S.B'.A (biComb ψ S.B.A S.J.W₀ S.B'.A c₁ c₂ (Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂))) → ∀ f ∈ GlobalTheory.TriStep.space Q, GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂') (rTrans (Q.gl₂ (embX (Q.c + 1) g)) f) = ((nabs Q.k (Matrix.GeneralLinearGroup.det g) ^ ((((Q.c + 1 : ℕ) : ℝ) + 1) / 2 − ((σ + 1 : ℕ) : ℝ)) : ℝ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalBimodelTranslationStatement

theorem **_root_.GlobalTheory.T.local_bimodel_translation** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (σ : ℕ) (S : Dbl2 ψ σ) (Q : QStep S) (c₁ : WO S.B.A S.J.W₀ →₀ ℂ) (c₂ : WO S.J.W₀ S.B'.A →₀ ℂ), Q.Away c₁ → Q.Away c₂ → ∀ φ, IsContAut S.B.A φ → ∀ (Λ : SB (Matrix Q.n₁ Q.n₀ Q.k × Matrix Q.n₂ Q.n₁ Q.k) →ₗ[ℂ] (WO S.B.A S.J.W₀ → WO S.J.W₀ S.B'.A → ℂ)), (ofSplit Q.q).IsBiModel ψ Q.F.F₀ Q.F.F₁ Q.F.F₂ Λ → ∀ (g : GL Q.n₂ Q.k) (Ξ₁ : SB (Matrix Q.n₁ Q.n₀ Q.k)) (Ξ₂ Ξ₂' : SB (Matrix Q.n₂ Q.n₁ Q.k)), Ξ₂'.1 = om Q.μ₂ 1 g Ξ₂.1 → IsBiTheta ψ S.B.A S.J.W₀ S.B'.A (biComb ψ S.B.A S.J.W₀ S.B'.A c₁ c₂ (Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂))) → ∀ f ∈ GlobalTheory.TriStep.space Q, GlobalTheory.TriStep.valK S Q c₁ c₂ φ Λ (GlobalTheory.LocFlat.CFac.prodSB Ξ₁ Ξ₂') (rTrans (Q.gl₂ (embX (Q.c + 1) g)) f) = ((nabs Q.k (Matrix.GeneralLinearGroup.det g) ^ ((((Q.c + 1 : ℕ) : ℝ) + 1) / 2 − ((σ +

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalCharacterConductorOneExists

theorem **splitPlace_polarisationDatum_inv_integral** (p : SplitPt D k) (hU : p.UnitAA) : p.pr (incA D D.host.xi0⁻¹) ∈ 𝒪[k]

theorem **hasConductor_of_mul** {χ ψk : AddChar k Circle} {ξ : k} (hv : valuation k ξ = 1) (he : ∀ x, ψk x = χ (ξ ∗ x)) (h : HasConductor χ 1) : HasConductor ψk 1

theorem **hasConductor_weil** (q : SplitPt D k) (hU : q.UnitAA) (ψ : AddChar (AF D.host.F) ℂ) (hψ : IsStdChar D ψ (−1)) (ψk : AddChar k Circle) (h : (ofSplit q).IsChar ψ ψk) (hx : (ofSplit q).xi ≠ 0) : HasConductor ψk 1

theorem **int_transfer** {k' : Type} [Field k'] [ValuativeRel k'] [TopologicalSpace k'] [IsNonarchimedeanLocalField k'] (q : SplitPt D k) (q' : SplitPt D k') (he : q.e = q'.e) (z : AE D) : q.pr z ∈ 𝒪[k] ↔ q'.pr z ∈ 𝒪[k']

theorem **dense_transfer** {k' : Type} [Field k'] [ValuativeRel k'] [TopologicalSpace k'] [IsNonarchimedeanLocalField k'] (q : SplitPt D k) (q' : SplitPt D k') (he : q.e = q'.e) (hd : DenseRange (q'.pr.comp (incA D))) : DenseRange (q.pr.comp (incA D))

theorem **lt_one_le_exp** {x : WithZero (Multiplicative ℤ)} (h : x < 1) : x ≤ WithZero.exp (−1 : ℤ)

theorem **isUnif_of** (P : HeightOneSpectrum (𝒪 K)) (ι : K →+* k) (hint : ∀ a, ι a ∈ 𝒪[k] ↔ Formalism.L2.Col.IntAt P a) (hd : DenseRange ι) (hq : ((Formalism.L2.Col.qOf P : ℕ) : 𝒪 K) ∉ P.asIdeal ^ 2) : ∃ ϖ : kˣ, GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ ∧ (ϖ : k) = ι ((Formalism.L2.Col.qOf P : ℕ) : K)

theorem **cond_unif** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (PP : Pl D.host.E), PP ∈ (pk W).Pg ∨ QAll W PP → ∀ l : LocX a (pk W).βX PP, (∃ ψk : AddChar l.k Circle, (ofSplit l.q).IsChar (pk W).ψ ψk ∧ HasConductor ψk 1) ∧ (∃ ν : Measure l.k, IsUnitVol ν) ∧ (QAll W PP → ∃ ϖ : l.kˣ, GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ ∧ GlobalTheory.SymBridge.IsCompl PP (Formalism.L2.Col.qOf PP) (l.q.pr.comp (incA D)) ϖ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalCharacterConductorOneExistsStatement

theorem **_root_.GlobalTheory.T.local_character_conductor_one_exists** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (PP : Pl D.host.E), PP ∈ (pk W).Pg ∨ QAll W PP → ∀ l : LocX a (pk W).βX PP, (∃ ψk : AddChar l.k Circle, (ofSplit l.q).IsChar (pk W).ψ ψk ∧ HasConductor ψk 1) ∧ (∃ ν : Measure l.k, IsUnitVol ν) ∧ (QAll W PP → ∃ ϖ : l.kˣ, GlobalTheory.IndefiniteRowSideFullHecke.FramePositiveSum.IsUnif ϖ ∧ GlobalTheory.SymBridge.IsCompl PP (Formalism.L2.Col.qOf PP) (l.q.pr.comp (incA D)) ϖ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalCombinationsComposeFinitelyManyPlaces

theorem **combAt_o** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {M : GlobalTheory.FinPl.Lat S} {M' : GlobalTheory.FinPl.Lat T} (l : GlobalTheory.SeeSaw.LocPl ψ S T M M') {P : GlobalTheory.FinPl.Pl D → Prop} (hP : P l.q.v) : GlobalTheory.FinPl.CombAt P l.o

theorem **combAt_comp** {D : D7} {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S T : HSp D} {P : GlobalTheory.FinPl.Pl D → Prop} {e e' : WO S T →₀ ℂ} (he : GlobalTheory.FinPl.CombAt P e) (he' : GlobalTheory.FinPl.CombAt P e') : ∃ e'', GlobalTheory.FinPl.CombAt P e'' ∧ ∀ f, comb ψ S T e'' f = comb ψ S T e (comb ψ S T e' f)

theorem **main** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → GlobalTheory.SeeSaw.LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ P : GlobalTheory.FinPl.Pl D → Prop, (∀ i, P (L i).q.v) → ∃ o' : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt P o' ∧ ∀ F, comb ψ S T o' F = (List.finRange n).foldr (fun i g => comb ψ S T (L i).o g) F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalCombinationsComposeFinitelyManyPlacesStatement

theorem **_root_.GlobalTheory.T.localCombinations_compose_finitely_many_places** : ∀ (D : D7) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (M : GlobalTheory.FinPl.Lat S) (M' : GlobalTheory.FinPl.Lat T), IsSelfDualC S M → IsSelfDualC T M' → ∀ (n : ℕ) (L : Fin n → GlobalTheory.SeeSaw.LocPl ψ S T M M'), (Function.Injective fun i => (L i).q.v) → ∀ P : GlobalTheory.FinPl.Pl D → Prop, (∀ i, P (L i).q.v) → ∃ o' : WO S T →₀ ℂ, GlobalTheory.FinPl.CombAt P o' ∧ ∀ F, comb ψ S T o' F = (List.finRange n).foldr (fun i g => comb ψ S T (L i).o g) F

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentAtSplitPlace

theorem **glOf_loc** (p : Place D k) (S : HSp D) {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k S.rank Y) (g : GL Y k) : p.IsLoc F (glOf p S F g) g

theorem **GlobalTheory.QGeoAt.geo_of_frames** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b : ∀ j, GA D (Yc D j)) (v : VCom (commOf (pk W) u (mkGeo m b))), (commOf (pk W) u (mkGeo m b)).Is → v.τ = y.τ → v.NT = a.LT → ∀ (PP : Pl D.host.E), QAll W PP → ∀ (l : LocX a (pk W).βX PP), LocY l y v.N (pk W).βY b (fun _ _ _ => 1) → (∀ j hj, IsStdFr GlobalTheory.SymBridge.xPos (l.FX j hj)) → IsStdFr GlobalTheory.SymBridge.zPos l.FT → ∃ G : GlobalTheory.QTw.QGeo v (baseOf (pk W) (lvP W) u (mkGeo m b)) PP, (∀ j hj, IsStdFr GlobalTheory.SymBridge.xPos (G.FX j hj)) ∧ IsStdFr GlobalTheory.SymBridge.zPos G.FT ∧ G.z₀ = fun r => (ofSplit

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentAtSplitPlaceStatement

theorem **_root_.GlobalTheory.T.local_component_at_split_place** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b : ∀ j, GA D (Yc D j)) (v : VCom (commOf (pk W) u (mkGeo m b))), (commOf (pk W) u (mkGeo m b)).Is → v.τ = y.τ → v.NT = a.LT → ∀ (PP : Pl D.host.E), QAll W PP → ∀ (l : LocX a (pk W).βX PP), LocY l y v.N (pk W).βY b (fun _ _ _ => 1) → (∀ j hj, IsStdFr GlobalTheory.SymBridge.xPos (l.FX j hj)) → IsStdFr GlobalTheory.SymBridge.zPos l.FT → ∃ G : GlobalTheory.QTw.QGeo v (baseOf (pk W) (lvP W) u (mkGeo m b)) PP, (∀ j hj, IsStdFr GlobalTheory.SymBridge.xPos (G.FX j hj)) ∧ IsStdFr GlobalTheory.SymBridge.zPos G.FT ∧ G.z₀ =

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentIntegralOfFiniteMatrixIntegral

theorem **comm_symm_mem** {F E : Type} [Field F] [NumberField F] [Field E] [NumberField E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) {a : E ⊗[F] v.adicCompletion F} (ha : a ∈ Formalism.UG.tensInt (F

theorem **prV_gen** {F E : Type} [Field F] [NumberField F] [Field E] [Algebra F E] (v : HeightOneSpectrum (𝒪 F)) (y : AF F ⊗[F] E) : (Algebra.TensorProduct.comm F (v.adicCompletion F) E).symm (Formalism.UG.prV (E

theorem **mem_intLoc_of_finM** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (κ : GA D V) (x : GL (Fin n) (GlobalTheory.AdForm.EF D)), GlobalTheory.AdForm.finM V hn κ = x.1 → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)), (∀ i j, (Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentIntegralOfFiniteMatrixIntegralStatement

theorem **_root_.GlobalTheory.T.localComponent_integral_of_finiteMatrix_integral** : ∀ (D : D7) (V : HSp D) (n : ℕ) (hn : V.rank = n) (κ : GA D V) (x : GL (Fin n) (GlobalTheory.AdForm.EF D)), GlobalTheory.AdForm.finM V hn κ = x.1 → ∀ (v : HeightOneSpectrum (𝒪 D.host.F)), (∀ i j, (Formalism.UG.bc (Formalism.UG.prV (E

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureBound

theorem **GlobalTheory.Fam.goodBF_of_normalFrames** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.NT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureBoundStatement

theorem **_root_.GlobalTheory.T.local_component_measure_bound** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureEquation

theorem **splitAt_places** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat

(commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.NT = a.LT → o.ζ₀ = a.ζ₀ → I

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureEquationStatement

theorem **_root_.GlobalTheory.T.local_component_measure_equation** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureIdentity

theorem **coreFrame_beta_eq** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {n : Type} [Fintype n] [DecidableEq n] {ρ : ℕ} (B : BesLev D ρ) (s : k) (FA : Frame k B.A.rank n) {F F' : Frame k B.V₀.rank (n ⊕ Unit)} (h : p.IsCoreFrame B s FA F) (h' : p.IsCoreFrame B s FA F') : F.β = F'.β

theorem **exists_shape_integral_in_normal_frames** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.NT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureIdentityAltBis

theorem **_root_.GlobalTheory.T.local_component_measure_identity_alt** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureIdentityStatement

theorem **_root_.GlobalTheory.T.local_component_measure_identity** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureRelation

theorem **form_eq** {ι : Type} [Fintype ι] (G : Matrix ι ι D.host.E) (x y : ι → D.host.E) : ∑ i, ∑ j, star (x i) * G i j * y j = star x ·ᵥ (G *ᵥ y)

theorem **isSelfDualC_iff** (S : HSp D) (Λ : Submodule (𝒪 D.host.E) (Fin S.rank → D.host.E)) : IsSelfDualC S Λ ↔ SelfDualFor S.gram Λ

theorem **integral_sign** {s b : D.host.E} (hs : s = 1 ∨ s = -1) : (∃ o : 𝒪 D.host.E, (o : D.host.E) = s * b) ↔ ∃ o : 𝒪 D.host.E, (o : D.host.E) = b

theorem **selfDualFor_map** {ι κ : Type} [Fintype ι] [Fintype κ] [DecidableEq ι] [DecidableEq κ] {G' : Matrix ι ι D.host.E} {G : Matrix κ κ D.host.E} (Q : Matrix ι κ D.host.E) {s : D.host.E} (hs : s = 1 ∨ s = -1) (hG : G = s • (Qᴴ * G' * Q)) (hdet : IsUnit G.det) (hcard : Fintype.card κ = Fintype.card ι) {Λ : Submodule (𝒪 D.host.E) (κ → D.host.E)} (hΛ : SelfDualFor G Λ) : SelfDualFor G' (Λ.map ((Matrix.mulVecLin Q).restrictScalars (𝒪 D.host.E)))

@[simp] theorem **extL_apply** {ι κ : Type} (v : ι → D.host.E) : extL (D

@[simp] theorem **extR_apply** {ι κ : Type} (v : κ → D.host.E) : extR (D

theorem **mem_sumLat** {ι κ : Type} {ΛA : Submodule (𝒪 D.host.E) (ι → D.host.E)} {ΛB : Submodule (𝒪 D.host.E) (κ → D.host.E)} (y : ι ⊕ κ → D.host.E) : y ∈ sumLat ΛA ΛB ↔ y ∘ Sum.inl ∈ ΛA ∧ y ∘ Sum.inr ∈ ΛB

theorem **split_vec** {ι κ : Type} (y : ι ⊕ κ → D.host.E) : y = extL (D

theorem **form_sum** {ι κ : Type} [Fintype ι] [Fintype κ] (GA : Matrix ι ι D.host.E) (GB : Matrix κ κ D.host.E) (x y : ι ⊕ κ → D.host.E) : star x ·ᵥ (fromBlocks GA 0 0 GB *ᵥ y) = star (x ∘ Sum.inl) ·ᵥ (GA *ᵥ (y ∘ Sum.inl)) + star (x ∘ Sum.inr) ·ᵥ (GB *ᵥ (y ∘ Sum.inr))

theorem **selfDualFor_sum** {ι κ : Type} [Fintype ι] [Fintype κ] {GA : Matrix ι ι D.host.E} {GB : Matrix κ κ D.host.E} {ΛA : Submodule (𝒪 D.host.E) (ι → D.host.E)} {ΛB : Submodule (𝒪 D.host.E) (κ → D.host.E)} (hA : SelfDualFor GA ΛA) (hB : SelfDualFor GB ΛB) : SelfDualFor (fromBlocks GA 0 0 GB) (sumLat ΛA ΛB)

theorem **line_mat_eq** {κ : Type} [Fintype κ] (hk : Fintype.card κ = 1) {M N : Matrix κ κ D.host.E} {ζ : κ → D.host.E} (hζ : star ζ ·ᵥ (M *ᵥ ζ) ≠ 0) (h : star ζ ·ᵥ (M *ᵥ ζ) = star ζ ·ᵥ (N *ᵥ ζ)) : M = N

theorem **exists_prime_not_mem** (K : Type) [Field K] [NumberField K] {n : ℕ} (hn : n ≠ 0) : ∃ p : IsDedekindDomain.HeightOneSpectrum (𝒪 K), ((n : 𝒪 K)) ∉ p.asIdeal

theorem **quad_mul** {ι κ : Type} [Fintype ι] [Fintype κ] (A : Matrix ι κ D.host.E) (G : Matrix ι ι D.host.E) (x : κ → D.host.E) : star x ·ᵥ ((Aᴴ * (G * A)) *ᵥ x) = star (A *ᵥ x) ·ᵥ (G *ᵥ (A *ᵥ x))

theorem **quad_smul** {κ : Type} [Fintype κ] (c : D.host.E) (G : Matrix κ κ D.host.E) (x : κ → D.host.E) : star (c • x) ·ᵥ (G *ᵥ (c • x)) = star c * c * (star x ·ᵥ (G *ᵥ x))

theorem **quad_smul_mat** {κ : Type} [Fintype κ] (c : D.host.E) (G : Matrix κ κ D.host.E) (x : κ → D.host.E) : star x ·ᵥ ((c • G) *ᵥ x) = c * (star x ·ᵥ (G *ᵥ x))

theorem **exists_selfDual_core** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg v QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.NT = a.LT → o.ζ₀ = a.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureRelationStatement

theorem **_root_.GlobalTheory.T.local_component_measure_relation** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg v QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureScalar

theorem **monoAt_of_besFrame** (hU : q.UnitAA) {σ : ℕ} (B : BesLev D σ) {n : Type} [Fintype n] [DecidableEq n] {s : k} (hs : valuation k s = 1) (hz : valuation k ((ofSplit q).hz B) = 1) {FA : Frame k B.A.rank n} {F : Frame k B.V.rank (n ⊕ Fin (2 * σ + 1))} {F₀ : Frame k B.V₀.rank (n ⊕ Unit)} (hF : (ofSplit q).IsBesFrame B s FA F) (hcore : (ofSplit q).IsCoreFrame B s FA F₀) {L : Lat B.V} {L₀ : Lat B.V₀} (hL : IsSelfDualC B.V L) (hL₀ : IsSelfDualC B.V₀ L₀) (h : IntIn q B.V F L) (h₀ : IntIn q B.V₀ F₀ L₀) : MonoAt B.d L L₀ q.v

theorem **line_jac** (hU : q.UnitAA) {ρ : ℕ} (J : JacLev D ρ) {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {s hh : k} (hs : valuation k s = 1) (hz : valuation k hh = 1) {F₀ : Frame k J.W₀.rank n} {F : Frame k J.W.rank (n ⊕ Fin c)} (hF : (ofSplit q).IsJacFrame J s hh F₀ F) {L : Lat J.W} (hL : IsSelfDualC J.W L) (h : IntIn q J.W F L) (i : Fin ρ) (a : AE D) (ha : a ∈ idAt D (· = q.v)) : colMul J.d.jX i a ∈ latV J.W L (· = q.v) ↔ a ∈ intAt D (· = q.v)

theorem **line_bes** (hU : q.UnitAA) {σ : ℕ} (B : BesLev D σ) {n : Type} [Fintype n] [DecidableEq n] {c : ℕ} {s : k} (hs : valuation k s = 1) (hz : valuation k ((ofSplit q).hz B) = 1) {FA : Frame k B.A.rank n} {F : Frame k B.V.rank (n ⊕ Fin c)} (hF : (ofSplit q).IsBesFrame B s FA F) {L : Lat B.V} (hL : IsSelfDualC B.V L) (h : IntIn q B.V F L) (i : Fin (σ + 1)) (a : AE D) (ha : a ∈ idAt D (· = q.v)) : colMul (xLast B.d (vecB B.o B.ζ)) i a ∈ latV B.V L (· = q.v) ↔ a ∈ intAt D (· = q.v)

theorem **psiE_triv_iff** (hU : q.UnitAA) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {s : AE D} (hs : s ∈ idAt D (· = q.v)) : (∀ o ∈ intAt D (· = q.v), psiE ψ (s * o) = 1) ↔ s ∈ intAt D (· = q.v)

theorem **primAt_of_lines** (hU : q.UnitAA) {ψ : AddChar (AF D.host.F) ℂ} (hψ : IsStdChar D ψ (-1)) {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (L : Lat S) (h : ∀ (i : Fin ρ), ∀ a ∈ idAt D (· = q.v), colMul d.jX i a ∈ latV S L (· = q.v) ↔ a ∈ intAt D (· = q.v)) : PrimAt ψ d L q.v

theorem **GlobalTheory.Fam.adaptFB_of_normalFrames** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg v QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.N

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureScalarStatement

theorem **_root_.GlobalTheory.T.local_component_measure_scalar** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg v QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureValue

theorem **tuplePlace_under_not_mem_tP** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg v QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.NT = a.LT → o.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentMeasureValueStatement

theorem **_root_.GlobalTheory.T.local_component_measure_value** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg v QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentPlacesCharacterized

theorem **latV_eq_of_moves** {D : D7} {S : HSp D} {g : GA D S} {Λ M N : GlobalTheory.FinPl.Lat S} {P : GlobalTheory.FinPl.Pl D → Prop} (hM : GlobalTheory.FinPl.Moves S g Λ M P) (hN : GlobalTheory.FinPl.Moves S g Λ N P) : GlobalTheory.FinPl.latV S M P = GlobalTheory.FinPl.latV S N P

theorem **exists_common_vacua** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b : ∀ j, GA D (Yc D j)) (v : VCom (commOf (pk W) u (mkGeo m b))), (commOf (pk W) u (mkGeo m b)).Is → v.τ = y.τ → v.NT = a.LT → (∀ p, p ∈ v.gP ↔ ∃ PP ∈ (pk W).Pg, p = PP.under (𝒪 D.host.F)) → GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg → (∀ j, b j ∈ GfC (Yc D j)) → a.bT ∈ GfC D.sp.X' → IsSelfDualC D.sp.X' a.LT → (∀ P, GlobalTheory.FinPl.Moves D.sp.X' a.bT (pk W).ΛX a.LT P) → (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (v.N j) ∧ ∀ P : GlobalTheory.FinPl.Pl D → Prop, (∀ p, P p → p ∉ v.gP) → GlobalTheory.FinPl.Moves (Yc D j) (b j) ((pk W).ΛY j) (v.N j) P

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentPlacesCharacterizedStatement

theorem **_root_.GlobalTheory.T.local_component_places_characterized** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b : ∀ j, GA D (Yc D j)) (v : VCom (commOf (pk W) u (mkGeo m b))), (commOf (pk W) u (mkGeo m b)).Is → v.τ = y.τ → v.NT = a.LT → (∀ p, p ∈ v.gP ↔ ∃ PP ∈ (pk W).Pg, p = PP.under (𝒪 D.host.F)) → GlobalTheory.Core.Mi.IsTypePlaces (pk W).Pg → (∀ j, b j ∈ GfC (Yc D j)) → a.bT ∈ GfC D.sp.X' → IsSelfDualC D.sp.X' a.LT → (∀ P, GlobalTheory.FinPl.Moves D.sp.X' a.bT (pk W).ΛX a.LT P) → (∀ j, j < D.R.m → IsSelfDualC (Yc D j) (v.N j) ∧ ∀ P : GlobalTheory.FinPl.Pl D → Prop, (∀ p, P p → p ∉ v.gP) → GlobalTheory.FinPl.Moves

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentVectorRelation

theorem **level_prime_not_mem_under_Pg** {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word) : ∀ PP ∈ (pk W).Pg, (((D.mem u).A.T : ℕ) : 𝒪 D.host.F) ∉ (PP.under (𝒪 D.host.F)).asIdeal

theorem **lattices_frames_basePoints_exist** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) (o : TDat (commOf (pk W) u (mkGeo m b₀)) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.τ = y.τ → v.NT = a.LT

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalComponentVectorRelationStatement

theorem **_root_.GlobalTheory.T.local_component_vector_relation** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (a : XGeo D) (y : YRat a) (m : YMeas y) (b₀ : ∀ j, GA D (Yc D j)) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : Type) (v : VCom (commOf (pk W) u (mkGeo m b₀))) k ι) (l : ∀ PP, PP ∈ (pk W).Pg ∨ QAll W PP → LocX a (pk W).βX PP) (bas : ∀ PP h j c, j + c = (l PP h).s + 1 → GL (XIx (Fin D.R.k) (Fin D.R.t) j ⊕ Fin c) (l PP h).k), (commOf (pk W) u (mkGeo m b₀)).Is → y.Tau (-D.host.xi0 ^ (D.R.m / 2)) → star a.ζ₀ ·ᵥ a.Lz.gram *ᵥ a.ζ₀ = (((D.mem u).A.T : ℕ) : D.host.E) → v.

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalDoublingIntegralOddCase

theorem **om_colE_indep** {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (μ : kˣ →* ℂˣ) (β : n → k) (Φ : Matrix (n ⊕ Fin c) n' k → ℂ) : om μ 1 (colE c ρ β) Φ = om 1 1 (colE c ρ β) Φ

theorem **isClosedEmbedding_negCol** (n n' : Type) : Topology.IsClosedEmbedding fun q : Matrix n n' k × (n → k) => (fromCols q.1 (Matrix.of fun r (_ : Unit) => (-q.2) r) : Matrix n (n' ⊕ Unit) k)

theorem **isClosedEmbedding_row** (n n' : Type) : Topology.IsClosedEmbedding fun q : Matrix n n' k × (n' → k) => (fromRows q.1 (Matrix.of fun (_ : Unit) => q.2) : Matrix (n ⊕ Unit) n' k)

theorem **exists_sum_prod** (n n' m' : Type) [Fintype n] [Fintype n'] [Fintype m'] (F : SB (Matrix n n' k × (m' → k))) : ∃ (m : ℕ) (Ψ : Fin m → SB (Matrix n n' k)) (Φ₀ : Fin m → SB (m' → k)), ∀ x β, F.1 (x, β) = ∑ i, (Ψ i).1 x * (Φ₀ i).1 β

theorem **fl_eq_flCol** (ψ : AddChar k Circle) (hψ : Continuous ψ) (hne : ψ ≠ 1) (ν : Measure k) (hν : IsSelfDual ψ ν) {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c ρ : ℕ) (h1 : 1 ≤ ρ) (hc : c = 2 * ρ) (μ : kˣ →* ℂˣ) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)) (x : Matrix n n' k) (β : n → k) : fl ψ ν μ ρ Φ.1 x β = flCol ψ ν (ρ + 1) Φ.1 (fromCols x (Matrix.of fun r (_ : Unit) => (-β) r))

theorem **contr_flat_eq_sum** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ ν : Measure k, IsSelfDual ψ ν → ∀ (n₀ : Type) [Fintype n₀] [DecidableEq n₀] (n₁ : Type) [Fintype n₁] [DecidableEq n₁] (n₂ : Type) [Fintype n₂] [DecidableEq n₂] (c ρ : ℕ), c + 1 = 2 * ρ → ∀ (μ : kˣ →* ℂˣ) (Φ₁ : SB (Matrix (n₁ ⊕ Fin (c + 1)) (n₀ ⊕ Fin (c + 1 + 1)) k)) (Φ₂ : SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k)), GlobalTheory.LocFlat.flCol ψ ν (ρ + 1) Φ₁.1 ∈ SB (Matrix n₁ (n₀ ⊕ Unit) k) ∧ GlobalTheory.LocFlat.flRow ψ ν ρ Φ₂.1 ∈ SB (Matrix (n₂ ⊕ Unit) n₁ k) ∧ (∀ Φ₀ : SB (n₁ → k), opEFJ ψ ν ρ Φ₀.1 Φ₂.1 ∈ SB (Matrix n₂ n₁ k)) ∧ (∃ (m : ℕ) (Ψ : Fin m → SB (Matrix n₁ n₀ k)) (Φ₀ : Fin m → SB (n₁ → k)), ∀ x β, fl ψ ν μ ρ Φ₁.1 x β = ∑ i, (Ψ i).1 x * (Φ₀ i).1 β) ∧ ∀ (m : ℕ) (Ψ : Fin m → SB

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapContractionSupportFinitelyManyCells

theorem **opE_const_mul** {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (ψ : AddChar k Circle) (ν : Measure k) (ρ : ℕ) (C : ℂ) (F : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (x₀ : Matrix n n' k) : opE ψ ν ρ (fun z => C * F z) x₀ = C * opE ψ ν ρ F x₀

theorem **opEFJ_const_mul** {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (ψ : AddChar k Circle) (ν : Measure k) (ρ : ℕ) (Φ₀ : (n' → k) → ℂ) (C : ℂ) (F : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (x₀ : Matrix n n' k) : GlobalTheory.IndefiniteRowSideFullHecke.FourierJacobiAverage.opEFJ ψ ν ρ Φ₀ (fun z => C * F z) x₀ = C * GlobalTheory.IndefiniteRowSideFullHecke.FourierJacobiAverage.opEFJ ψ ν ρ Φ₀ F x₀

theorem **om_eq_const_mul** {a b : Type} [Fintype a] [DecidableEq a] [Fintype b] [DecidableEq b] (μ' : kˣ →* ℂˣ) (x : GL a k) (y : GL b k) (Φ : Matrix b a k → ℂ) : om μ' x y Φ = fun z => ((μ' (Matrix.GeneralLinearGroup.det x) : ℂ) * ((μ' (Matrix.GeneralLinearGroup.det y))⁻¹

: ℂˣ)) ∗ om 1 1 y (om 1 x 1 Φ) z

theorem **GlobalTheory.LocFlat.contr_support_finite_cells** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ (ν : Measure k), IsSelfDual ψ ν → ∀ (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] (c ρ : ℕ), c + 1 = 2 ∗ ρ → ∀ (μ μ' : kˣ →∗ ℂˣ) (J : Subgroup (GL (n₂ ⊕ Fin c) k)), IsOpen (J : Set (GL (n₂ ⊕ Fin c) k)) → ∀ (Φ₁ : SB (Matrix (n₁ ⊕ Fin (c + 1)) (n₀ ⊕ Fin (c + 1 + 1)) k)) (Φ₂ : SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k)) (a : GL (n₀ ⊕ Fin (c + 1 + 1)) k) (x : GL (n₁ ⊕ Fin (c + 1)) k), ∃ T : Finset (GL (n₂ ⊕ Fin c) k), ∀ y : GL (n₂ ⊕ Fin c) k, GlobalTheory.LocFlat.contr ψ ν (GlobalTheory.LocFlat.flCol ψ ν (ρ + 1) (om μ a x Φ₁.1)) (GlobalTheory.LocFlat.flRow ψ ν ρ (om μ' x y Φ₂.1)) ≠ 0 → ∃ r ∈ R k n₂

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapContractionSupportFinitelyManyCellsStatement

theorem **_root_.GlobalTheory.T.localFlatMapContraction_support_finitelyManyCells** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle), Continuous ψ → ψ ≠ 1 → ∀ (ν : Measure k), IsSelfDual ψ ν → ∀ (n₀ n₁ n₂ : Type) [Fintype n₀] [DecidableEq n₀] [Fintype n₁] [DecidableEq n₁] [Fintype n₂] [DecidableEq n₂] (c ρ : ℕ), c + 1 = 2 ∗ ρ → ∀ (μ μ' : kˣ →∗ ℂˣ) (J : Subgroup (GL (n₂ ⊕ Fin c) k)), IsOpen (J : Set (GL (n₂ ⊕ Fin c) k)) → ∀ (Φ₁ : SB (Matrix (n₁ ⊕ Fin (c + 1)) (n₀ ⊕ Fin (c + 1 + 1)) k)) (Φ₂ : SB (Matrix (n₂ ⊕ Fin c) (n₁ ⊕ Fin (c + 1)) k)) (a : GL (n₀ ⊕ Fin (c + 1 + 1)) k) (x : GL (n₁ ⊕ Fin (c + 1)) k), ∃ T : Finset (GL (n₂ ⊕ Fin c) k), ∀ y : GL (n₂ ⊕ Fin c) k, GlobalTheory.LocFlat.contr ψ ν (GlobalTheory.LocFlat.flCol ψ ν (ρ + 1) (om μ a x Φ₁.1)) (GlobalTheory.LocFlat.flRow ψ ν ρ (om μ' x y Φ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapDualColumnMultiplicationLaws

theorem **e_iota** (p : Place D k) (x : k) : p.e ∗ p.ι x = p.ι x

theorem **iota_mul** (p : Place D k) (x : k) (m : AE D) : p.ι x ∗ m = p.ι (x ∗ p.π m)

theorem **xi_ne** (p : Place D k) : p.xi ≠ 0

theorem **map_iota_mul** (p : Place D k) {a b c : ℕ} (zk : Matrix (Fin a) (Fin b) k) (x : Mx D b c) : (zk.map p.ι : Mx D a b) · x = (zk ∗ x.map p.π).map p.ι

theorem **trace_map_iota** (p : Place D k) {a : ℕ} (x : Matrix (Fin a) (Fin a) k) : (x.map p.ι : Mx D a a).trace = p.ι x.trace

theorem **map_iota_pi** (p : Place D k) {a b : ℕ} (x : Matrix (Fin a) (Fin b) k) : (x.map p.ι).map p.π = x

theorem **hO_extY** {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (w : WO S T) (a : Mx D T.rank ρ) : hO S T w (d.extY T a) = pairS T (w · d.jX) a

theorem **pairS_giA_adj** (T : HSp D) {ρ : ℕ} (x : Mx D T.rank ρ) (z : Mx D ρ T.rank) : pairS T x (giA T · adj z) = (z · x).trace

theorem **psi_sympl_extY** (p : Place D k) {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (hχ : p.IsChar ψ ψk) {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) (zk : Matrix (Fin ρ) (Fin T.rank) k) (w : WO S T) : ψ (symplO S T w (d.extY T (giA T · adj (zk.map p.ι : Mx D ρ T.rank)))) = (ψk ((zk ∗ (w · d.jX : Mx D T.rank ρ).map p.π).trace) : ℂ)

theorem **isHalf_mul_left** {e : AE D} {a b c : ℕ} (x : Mx D a b) {y : Mx D b c} (h : IsHalf e y) : IsHalf e (x · y)

theorem **isHalf_mul_right** {e : AE D} {a b c : ℕ} {x : Mx D a b} (y : Mx D b c) (h : IsHalf e x) : IsHalf e (x · y)

theorem **isHalf_adj** (p : Place D k) {a b : ℕ} (zk : Matrix (Fin a) (Fin b) k) : IsHalf (cA D p.e) (adj (zk.map p.ι : Mx D a b))

theorem **exists_mxAt** (p : Place D k) : ∃ v : GlobalTheory.FinPl.Pl D, ∀ (a b : ℕ) (x : Mx D a b), IsHalf (cA D p.e) x → GlobalTheory.FinPl.MxAt (· = v) x

theorem **trace_mul_uni** {ρ : ℕ} (M : Mx D ρ ρ) (hM : ∀ i j, j < i → M i j = 0) (u : UX D ρ) : (M · u.1.1).trace = M.trace

theorem **col_mul** {K m l o o' : Type} [Field K] [Fintype l] (M : Matrix m l K) (E : Matrix l o K) (F : Matrix l o' K) (j : o) (j' : o') {κ : K} (hκ : κ ≠ 0) (h : ∀ x, F x j' = κ ∗ E x j) (r : m) : (M ∗ E) r j = κ⁻¹ ∗ (M ∗ F) r j'

theorem **row_mul** {K m l o : Type} [Field K] [Fintype l] {ρ : ℕ} (i₀ : Fin ρ) (a : K) (G : Matrix m l K) (r : m) (X : Matrix l o K) (i : Fin ρ) (j : o) : ((Matrix.of fun i' β => if i' = i₀ then a ∗ G r β else 0 : Matrix (Fin ρ) l K) ∗ X) i j = if i = i₀ then a ∗ (G ∗ X) r j else 0

theorem **trace_row_mul** {K m l : Type} [Field K] [Fintype l] {ρ : ℕ} (i₀ : Fin ρ) (a : K) (G : Matrix m l K) (r : m) (X : Matrix l (Fin ρ) K) : ((Matrix.of fun i' β => if i' = i₀ then a ∗ G r β else 0 : Matrix (Fin ρ) l K) ∗ X).trace = a ∗ (G ∗ X) r i₀

theorem **law_dual_col** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c +

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapDualColumnMultiplicationLawsStatement

theorem **_root_.GlobalTheory.T.localFlatMap_dualColumn_multiplicationLaws** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsMode

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFactorIntertwinesMultiplication

theorem **symplectic_pairing_ext_zero_inc_zero** {V V₀ W W₀ : HSp D} {ρ σ : ℕ} (dV : HypDec V V₀ ρ) (dW : HypDec W W₀ σ) (t : WO V W) (y : WO V₀ W₀) : symplO V W t (dV.ext0 W (dW.inc0 V₀ y)) = symplO V₀ W₀ (dW.p0 · t · dV.j0) y

theorem **isHalf_mul** {a b c d : ℕ} {e : AE D} (A : Mx D a b) {y : Mx D b c} (C : Mx D c d) (hy : IsHalf e y) : IsHalf e (A · y · C)

theorem **mul_iota** (b : AE D) (a : k) : b ∗ p.ι a = p.ι (p.π b ∗ a)

theorem **rat_mul_iota** {a b c : ℕ} (R : Matrix (Fin a) (Fin b) D.host.E) (A : Matrix (Fin b) (Fin c) k) : rat R · (A.map p.ι : Mx D b c) = ((R.map p.pr ∗ A).map p.ι : Mx D a c)

theorem **iota_mul_rat** {a b c : ℕ} (A : Matrix (Fin a) (Fin b) k) (R : Matrix (Fin b) (Fin c) D.host.E) : (A.map p.ι : Mx D a b) · rat R = ((A ∗ R.map p.pr).map p.ι : Mx D a c)

theorem **map_pr_mul** {a b c : ℕ} (R : Matrix (Fin a) (Fin b) D.host.E) (R' : Matrix (Fin b) (Fin c) D.host.E) : (R ∗ R').map p.pr = R.map p.pr ∗ R'.map p.pr

theorem **q_zero_projection_identities** {W W₀ : HSp D} {σ : ℕ} (d : HypDec W W₀ σ) : d.q0.map p.pr ∗ d.i₀.map p.pr = 1 ∧ d.q0.map p.pr ∗ d.e.map p.pr = 0 ∧ d.q0.map p.pr ∗ d.e'.map p.pr = 0

theorem **q_zero_jacobi_frame_projection** {ρ : ℕ} (J : JacLev D ρ) (s hh : k) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)) (h : p.IsJacFrame J s hh G₀ FW) : J.d.q0.map p.pr ∗ FW.β = fromCols G₀.β 0

theorem **bessel_normal_frame_core_frame_part** {ρ : ℕ} (B : BesLev D ρ) (s : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin c)) (h₀ : p.IsCoreFrame B s FA F₀) (h : p.IsBesFrame B s FA FV) (j : Fin c) (hj : j.val = ρ) : B.d.i₀.map p.pr ∗ F₀.β = FV.β.submatrix id (Sum.map id fun _ => j)

theorem **core_up** {ρ : ℕ} (B : BesLev D ρ) (J : JacLev D ρ) (s hh : k) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)) (h₀ : p.IsCoreFrame B s FA F₀) (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s hh G₀ FW) (j : Fin (c + 1)) (hj : j.val = ρ) (t : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) : J.d.p0 · (p.up FV FW t : WO B.V J.W) · B.d.j0 = p.up F₀ G₀ (t.submatrix Sum.inl (Sum.map id fun _ => j))

theorem **submatrix_affPt** (ρ : ℕ) (j : Fin (c + 1)) (hj : j.val = ρ) (x : Matrix n (n' ⊕ Unit) k) (e : {rs : (n ⊕ Fin c) × (n' ⊕ Fin (c + 1)) // Free (ρ + 1) rs} → k) : (affPt (ρ + 1) x.toCols₁ e + GlobalTheory.LocFlat.colAt n' c (ρ + 1) fun r => x r (Sum.inr ())).submatrix Sum.inl (Sum.map id fun _ => j) = x

theorem **continuous_hO_up** {S T : HSp D} (FS : Frame k S.rank n') (FT : Frame k T.rank n) (y : WO S T) : Continuous fun t => p.π (hO S T (p.up FS FT t) y)

theorem **model_modul** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ (L₀ : SB (Matrix n (n' ⊕ Unit) k) →ₗ

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFactorIntertwinesMultiplicationStatement

theorem **_root_.GlobalTheory.T.localFlatMap_factor_intertwines_multiplication** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.Is

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFactorIntertwinesTranslations

theorem **ι_sandwich** (a c : AE D) (m : k) : p.ι (p.π a ∗ m ∗ p.π c) = a ∗ p.ι m ∗ c

theorem **mul_map_ι_mul** {a b l m : Type} [Fintype a] [Fintype b] (A : Matrix m a (AE D)) (M : Matrix a b k) (C : Matrix b l (AE D)) : A ∗ M.map p.ι ∗ C = (A.map p.π ∗ M ∗ C.map p.π).map p.ι

theorem **map_incA_map** {a b : ℕ} (M : Matrix (Fin a) (Fin b) D.host.E) : (M.map (incA D)).map p.π = M.map p.pr

theorem **ι_mem_idAt** : ∃ v, ∀ y : k, p.ι y ∈ GlobalTheory.FinPl.idAt D (· = v)

theorem **sb_translate** {X : Type} [TopologicalSpace X] [AddGroup X] [IsTopologicalAddGroup X] {Φ : X → ℂ} (hΦ : Φ ∈ SB X) (t : X) : (fun z => Φ (z + t)) ∈ SB X

theorem **flCol_translate** [MeasurableSpace k] [BorelSpace k] (ψ : AddChar k Circle) (ν : Measure k) (c ρ : ℕ) (Φ : Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k → ℂ) (x t : Matrix n (n' ⊕ Unit) k) : GlobalTheory.LocFlat.flCol ψ ν (ρ + 1) (fun z => Φ (z + bigPt c ρ x)) t = GlobalTheory.LocFlat.flCol ψ ν (ρ + 1) Φ (t + x)

theorem **mul_bigPt_mul** {a b : Type} {c ρ : ℕ} (j₀ : Fin (c + 1)) (hj : j₀.val = ρ) (β : Matrix a (n ⊕ Fin c) k) (β₀ : Matrix a n k) (γ : Matrix (n' ⊕ Fin (c + 1)) b k) (γ₀ : Matrix (n' ⊕ Unit) b k) (hβ : ∀ r i, β r (Sum.inl i) = β₀ r i) (hγ : ∀ i s, γ (Sum.inl i) s = γ₀ (Sum.inl i) s) (hγ' : ∀ s, γ (Sum.inr j₀) s = γ₀ (Sum.inr ()) s) (x : Matrix n (n' ⊕ Unit) k) : β ∗ bigPt c ρ x ∗ γ = β₀ ∗ x ∗ γ₀

theorem **rows_of_inverse** {N N₀ c : ℕ} (j₀ : Fin (c + 1)) (F : Frame k N (n' ⊕ Fin (c + 1))) (F₀ : Frame k N₀ (n' ⊕ Unit)) (q : Matrix (Fin N₀) (Fin N) k) (h1 : ∀ r i, (q ∗ F.β) r (Sum.inl i) = F₀.β r (Sum.inl i)) (h2 : ∀ r, (q ∗ F.β) r (Sum.inr j₀) = F₀.β r (Sum.inr ())) (h3 : ∀ r j, j ≠ j₀ → (q ∗ F.β) r (Sum.inr j) = 0) : (∀ i s, F.γ (Sum.inl i) s = (F₀.γ ∗ q) (Sum.inl i) s) ∧ ∀ s, F.γ (Sum.inr j₀) s = (F₀.γ ∗ q) (Sum.inr ()) s

theorem **proj_laws** {S S₀ : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) : d.p0.map p.π ∗ d.i₀.map p.pr = 1 ∧ d.p0.map p.π ∗ d.e.map p.pr = 0 ∧ d.p0.map p.π ∗ d.e'.map p.pr = 0

theorem **pr_eq** : p.pr = prHom p

theorem **up_bigPt** {ρ : ℕ} (B : BesLev D ρ) (J : JacLev D ρ) (s hh : k) {n n' : Type} [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] {c : ℕ} (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)) (hC : p.IsCoreFrame B s FA F₀) (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s hh G₀ FW) (x : Matrix n (n' ⊕ Unit) k) : p.up FV FW (bigPt c ρ x) = B.d.ext0 J.W (J.d.inc0 B.V₀ (p.up F₀ G₀ x))

theorem **factor_trans** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ (L₀ : SB (Matrix n (n' ⊕ Unit) k) →

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFactorIntertwinesTranslationsStatement

theorem **_root_.GlobalTheory.T.localFlatMap_factor_intertwines_translations** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsMo

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFactorValuesThetaFunctions

theorem **flat_model_theta** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ (L₀ : SB (Matrix n (n' ⊕ Unit)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFactorValuesThetaFunctionsStatement

theorem **_root_.GlobalTheory.T.localFlatMap_factor_values_thetaFunctions** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFreeEntryTranslationLawsBesselToJaco

theorem **e_mul_ι** (p : Place D k) (x : k) : p.e * p.ι x = p.ι x

theorem **ι_mul** (p : Place D k) (x : k) (z : AE D) : p.ι x * z = p.ι (x * p.π z)

theorem **infPart_e** (p : Place D k) : infPart D p.e = 0

theorem **ι_mem** (p : Place D k) (x : k) : p.ι x ∈ GlobalTheory.FinPl.idAt D (fun _ => True)

theorem **sand** (p : Place D k) {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (r : Y) (q : Y') (a : k) {m m' : ℕ} (N : Mx D m T.rank) (N' : Mx D S.rank m') (i : Fin m) (j : Fin m') : (N · p.up FS FT (Matrix.single r q a) · N') i j = p.ι ((N.map p.π * FT.β) i r * a * (FS.γ * N'.map p.π) q j)

theorem **col_coord** {N : ℕ} (F : Frame k N Y) (j : Y) (c : k) (v : Fin N → k) (h : ∀ r, F.β r j = c * v r) (q : Y) : c * ∑ t, F.γ q t * v t = if q = j then 1 else 0

theorem **col_coord_ne** {N : ℕ} (F : Frame k N Y) (j : Y) (c : k) (v : Fin N → k) (h : ∀ r, F.β r j = c * v r) (q : Y) (hq : q ≠ j) : ∑ t, F.γ q t * v t = 0

theorem **row_coord** (p : Place D k) {N m m' : ℕ} (F : Frame k N Y) (j : Y) (c : k) (M : Mx D m N) (M' : Mx D N m') (i : Fin m') (h : ∀ r, F.β r j = c * p.π (M' r i)) (l : Fin m) : (M.map p.π * F.β) l j = c * p.π ((M · M') l i)

theorem **symplO_skew** (S T : HSp D) (x y : WO S T) : symplO S T y x = −symplO S T x y

theorem **ch_single_succ** (l : Fin c) (a : k) : ch (Matrix.single (Sum.inr l : n ⊕ Fin c) (Sum.inr l.succ : n' ⊕ Fin (c + 1)) a) = a

theorem **ch_single_ne** (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)) (a : k) (h : ∀ l, r = Sum.inr l → q ≠ Sum.inr l.succ) : ch (Matrix.single r q a) = 0

theorem **bes_X** (hB : p.IsBesFrame B s FA FV) (q : n' ⊕ Fin (c + 1)) (hq : ∀ j, q = Sum.inr j → ρ ≤ j.val) (i : Fin ρ) : (FV.γ * B.d.jX.map p.π) q i = 0

theorem **bes_Y_ne** (hB : p.IsBesFrame B s FA FV) (q : n' ⊕ Fin (c + 1)) (i : Fin ρ) (hq : ∀ j, q = Sum.inr j → j.val + i.val ≠ 2 * ρ) : (FV.γ * B.d.jY.map p.π) q i = 0

theorem **bes_Y** (hB : p.IsBesFrame B s FA FV) (j : Fin (c + 1)) (i : Fin ρ) (h : j.val + i.val = 2 * ρ) : s * (−p.hz B / p.xi) * (−1 / p.xi) ^ (ρ − 1 − i.val) * (FV.γ * B.d.jY.map p.π) (Sum.inr j) i = 1

theorem **besselNormalFrame_lineVectorColumn** (hB : p.IsBesFrame B s FA FV) (j : Fin (c + 1)) (h : j.val = ρ) (q : n' ⊕ Fin (c + 1)) : s * ∑ t, FV.γ q t * p.pr ((B.d.i₀ *ᵥ vecB B.o B.ζ) t) = if q = Sum.inr j then 1 else 0

theorem **bes_zero** (hC : p.IsCoreFrame B s FA F₀) (hB : p.IsBesFrame B s FA FV) (j : Fin (c + 1)) (hj : ρ < j.val) (t : Fin B.V₀.rank) : (FV.γ * B.d.j0.map p.π) (Sum.inr j) t = 0

theorem **jac_low** (hJ : p.IsJacFrame J s hh G₀ FW) (l : Fin c) (hl : l.val < ρ) {m : ℕ} (M : Mx D m J.W.rank) (i : Fin m) : (M.map p.π * FW.β) i (Sum.inr l) = s * p.xi ^ (ρ − l.val) * p.π ((M · J.d.jX) i (l.val, hl))

theorem **jac_Y** (hJ : p.IsJacFrame J s hh G₀ FW) (l : Fin c) (i i' : Fin ρ) (h : l.val + i'.val + 1 = 2 * ρ) : (J.d.pY.map p.π * FW.β) i (Sum.inr l) = s * (hh / p.xi) * (−1 / p.xi) ^ (ρ − 1 − i'.val) * if i = i' then 1 else 0

theorem **jac_Y_ne** (hJ : p.IsJacFrame J s hh G₀ FW) (r : n ⊕ Fin c) (i : Fin ρ) (h : ∀ l, r = Sum.inr l → l.val + i.val + 1 ≠ 2 * ρ) : (J.d.pY.map p.π * FW.β) i r = 0

theorem **psi_xi** (p : Place D k) {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (hψ : p.IsChar ψ ψk) (x : k) : ψ (Algebra.trace (AF D.host.F) (AE D) (incA D D.host.xi0 * cA D (p.ι (−x)))) = ψk x

theorem **up_mul_eq_zero** (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)) (a : k) {m : ℕ} (N : Mx D B.V.rank m) (h : ∀ j, (FV.γ * N.map p.π) q j = 0) : p.up FV FW (Matrix.single r q a) · N = 0

theorem **mul_up_eq_zero** (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)) (a : k) {m : ℕ} (N : Mx D m J.W.rank) (h : ∀ i, (N.map p.π * FW.β) i r = 0) : N · p.up FV FW (Matrix.single r q a) = 0

theorem **mxAt_sand** (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)) (a : k) {m m' : ℕ} (N : Mx D m J.W.rank) (N' : Mx D B.V.rank m') : GlobalTheory.FinPl.MxAt (fun _ => True) (N · p.up FV FW (Matrix.single r q a) · N')

theorem **up_eq_extY** (hC : p.IsCoreFrame B s FA F₀) (hB : p.IsBesFrame B s FA FV) (r : n ⊕ Fin c) (j : Fin (c + 1)) (hj : ρ < j.val) (a : k) : p.up FV FW (Matrix.single r (Sum.inr j) a) = B.d.extY J.W (p.up FV FW (Matrix.single r (Sum.inr j) a) · B.d.jY)

theorem **pairS_jX** {W W₀ : HSp D} (d : HypDec W W₀ ρ) (u : Mx D ρ ρ) (y : Mx D W.rank ρ) : pairS W (d.jX · u) y = (adj (d.pY · y) · u).trace

theorem **uInv_up** {hh : k} (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s hh G₀ FW) (r : n ⊕ Fin c) (j : Fin (c + 1)) (hr : ∀ l, r = Sum.inr l → l.val < j.val) (a : k) : GlobalTheory.FinPl.UInv J.W (xZero J.d) (p.up FV FW (Matrix.single r (Sum.inr j) a) · B.d.jY)

theorem **trace_dual** (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s (p.hz B) G₀ FW) (r : n ⊕ Fin c) (j : Fin (c + 1)) (hj : ρ < j.val) (a : k) : (J.d.pY · (p.up FV FW (Matrix.single r (Sum.inr j) a) · B.d.jY)).trace = p.ι (-ch (Matrix.single r (Sum.inr j : n' ⊕ Fin (c + 1)) a))

theorem **value_dual** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (hψ : p.IsChar ψ ψk) (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s (p.hz B) G₀ FW) (r : n ⊕ Fin c) (j : Fin (c + 1)) (hj : ρ < j.val) (a : k) : ψ (symplO B.V J.W (B.d.extX J.W (xZero J.d)) (B.d.extY J.W (p.up FV FW (Matrix.single r (Sum.inr j) a) · B.d.jY))) = ψk (ch (Matrix.single r (Sum.inr j : n' ⊕ Fin (c + 1)) a))

theorem **up_eq_incX** {hh : k} (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s hh G₀ FW) (l : Fin c) (hl : l.val < ρ) (q : n' ⊕ Fin (c + 1)) (hq : ∀ j, q = Sum.inr j → j.val = ρ) (a : k) : p.up FV FW (Matrix.single (Sum.inr l) q a) = B.d.ext0 J.W (J.d.incX B.V₀ (J.d.pX · p.up FV FW (Matrix.single (Sum.inr l) q a) · B.d.j0))

theorem **splitComponent_wittZero_besselYZeroPoint** (t : Fin B.V.rank) (i : Fin ρ) : p.π ((B.d.j0 · yZero B.V₀ ρ (vecB B.o B.ζ)) t i) = -(p.pr ((B.d.i₀ *ᵥ vecB B.o B.ζ) t) * if i.val + 1 = ρ then 1 else 0)

theorem **bes_yZero** (hB : p.IsBesFrame B s FA FV) (j : Fin (c + 1)) (hj : j.val = ρ) (q : n' ⊕ Fin (c + 1)) (i : Fin ρ) : s * (FV.γ * (B.d.j0 · yZero B.V₀ ρ (vecB B.o B.ζ)).map p.π) q i = -((if q = Sum.inr j then 1 else 0) * if i.val + 1 = ρ then 1 else 0)

theorem **trace_iso** {hh : k} (hB : p.IsBesFrame B s FA FV) (hJ : p.IsJacFrame J s hh G₀ FW) (l : Fin c) (hl : l.val < ρ) (q : n' ⊕ Fin (c + 1)) (hq : ∀ j, q = Sum.inr j → j.val = ρ) (a : k) : (yZero B.V₀ ρ (vecB B.o B.ζ) · (J.d.pX · p.up FV FW (Matrix.single (Sum.inr l) q a) · B.d.j0)).trace = -(incA D D.host.xi0 * p.ι (ch (Matrix.single (Sum.inr l : n ⊕ Fin c) q a)))

theorem **flatBF_trans** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)), p.IsModel ψ FV FW Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c +

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapFreeEntryTranslationLawsBesselToJacoStatement

theorem **_root_.GlobalTheory.T.localFlatMap_freeEntry_translationLaws_besselToJacobi** : ∀ (D : D7), FlatLaw D → GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ν : Measure k), ν.IsAddHaarMeasure → ∀ (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (ρ : ℕ) (B : BesLev D ρ) (J : JacLev D ρ) (st : StepBF ψ B J) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FV : Frame k B.V.rank (n' ⊕ Fin (c + 1))) (G₀ : Frame k J.W₀.rank n) (FW : Frame k J.W.rank (n ⊕ Fin c)), p.IsCoreFrame B s FA F₀ → p.IsBesFrame B s FA FV → p.IsJacFrame J s (p.hz B) G₀ FW → ∀ (Tm : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k) →ₗ[ℂ] (WO B.V J.W → ℂ)) (WO B.V J.W → ℂ)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapTranslationLawsJacobiToBesselFree

theorem **xLast_eq_rat** {T T₀ : HSp D} {σ : ℕ} (d : HypDec T T₀ σ) (z : Fin T₀.rank → D.host.E) : xLast d z = rat (lastMat d z)

theorem **e_mul_ι** (x : k) : p.e * p.ι x = p.ι x

theorem **ι_mul_incA** (x : k) (b : D.host.E) : p.ι x * incA D b = p.ι (x * p.pr b)

theorem **incA_mul_ι** (x : k) (b : D.host.E) : incA D b * p.ι x = p.ι (p.pr b * x)

theorem **map_ι_mul_rat** {a b c : Type} [Fintype b] (M : Matrix a b k) (N : Matrix b c D.host.E) : M.map p.ι * N.map (incA D) = (M * N.map p.pr).map p.ι

theorem **rat_mul_map_ι** {a b c : Type} [Fintype b] (N : Matrix a b D.host.E) (M : Matrix b c k) : N.map (incA D) * M.map p.ι = (N.map p.pr * M).map p.ι

theorem **infPart_e** : infPart D p.e = 0

theorem **infPart_ι** (x : k) : infPart D (p.ι x) = 0

theorem **mxAt_of_ι** {m' n' : ℕ} (M : Mx D m' n') (hM : ∀ i j, ∃ x, M i j = p.ι x) : GlobalTheory.FinPl.MxAt (fun _ => True) M

theorem **mxAt_colMx** {T : HSp D} {ρ : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (FT : Frame k T.rank Y) (r : Y) (i : Fin ρ) (b : k) : GlobalTheory.FinPl.MxAt (fun _ => True) (colMx p FT r i b)

theorem **mxAt_rowMx** {ρ m : ℕ} (l : Fin ρ) (v : Fin m → k) : GlobalTheory.FinPl.MxAt (fun _ => True) (rowMx p l v)

theorem **pr_ne_zero** {a : D.host.E} (ha : a ≠ 0) : p.pr a ≠ 0

theorem **up_single** {S T : HSp D} {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (r : Y) (q : Y') (a : k) (row : Fin T.rank) (col : Fin S.rank) : p.up FS FT (Matrix.single r q a) row col = p.ι (FT.β row r * a * FS.γ q col)

theorem **extY_colMx** {S S₀ T : HSp D} {ρ : ℕ} (d : HypDec S S₀ ρ) {Y : Type} [Fintype Y] [DecidableEq Y] (FT : Frame k T.rank Y) (r : Y) (i : Fin ρ) (b : k) (row : Fin T.rank) (col : Fin S.rank) : d.extY T (colMx p FT r i b) row col = p.ι (b * FT.β row r * p.pr (d.qY i col))

theorem **wittExtZero_inclusionX_adelicSingleRowMatrix** {S S₀ T T₀ : HSp D} {ρ σ : ℕ} (d : HypDec S S₀ ρ) (d' : HypDec T T₀ σ) (l : Fin σ) (v : Fin S₀.rank → k) (row : Fin T.rank) (col : Fin S.rank) : d.ext0 T (d'.incX S₀ (rowMx p l v)) row col = p.ι (∑ m, p.pr (d'.e row l) * v m * p.pr (d.q0 m col))

theorem **frame_row** {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k N Y) (q : Y) (v : Fin N → k) (h : ∀ q', ∑ col, v col * F.β col q' = if q' = q then 1 else 0) (col : Fin N) : F.γ q col = v col

theorem **pr_mul** {l m n : ℕ} (M : Matrix (Fin l) (Fin m) D.host.E) (N : Matrix (Fin m) (Fin n) D.host.E) : (M * N).map p.pr = M.map p.pr * N.map p.pr

theorem **pr_zero** : p.pr 0 = 0

theorem **pr_sum_zero** {l m n : ℕ} {M : Matrix (Fin l) (Fin m) D.host.E} {N : Matrix (Fin m) (Fin n) D.host.E} (h : M * N = 0) (i : Fin l) (j : Fin n) : ∑ c, p.pr (M i c) * p.pr (N c j) = 0

theorem **pr_sum_one** {l m : ℕ} {M : Matrix (Fin l) (Fin m) D.host.E} {N : Matrix (Fin m) (Fin l) D.host.E} (h : M * N = 1) (i j : Fin l) : ∑ c, p.pr (M i c) * p.pr (N c j) = if i = j then 1 else 0

theorem **gram_swap'** (S : HSp D) {l m : ℕ} {M : Matrix (Fin S.rank) (Fin l) D.host.E} {N : Matrix (Fin S.rank) (Fin m) D.host.E} {Z : Matrix (Fin l) (Fin m) D.host.E} (h : Mᴴ * S.gram * N = Z) : Nᴴ * S.gram * M = Zᴴ

theorem **qY_i₀** : d.qY * d.i₀ = 0

theorem **qY_e'** : d.qY * d.e' = 1

theorem **wittDecomposition_coreProj_mul_e_eq_zero** : d.q0 * d.e = 0

theorem **wittZeroProjection_mul_dualBasis_eq_zero** : d.q0 * d.e' = 0

theorem **wittDecomposition_coreProj_mul_core_eq_one** : d.q0.map p.pr * d.i₀.map p.pr = 1

theorem **jac_cases** (h : c = 2 * ρ) (j : Fin c) : (∃ i : Fin ρ, j.val = i.val) ∨ ∃ i : Fin ρ, j.val + i.val + 1 = 2 * ρ

theorem **jac_γ_inr** (h : p.IsJacFrame J s hh G₀ FW) (hs : s ≠ 0) (hhh : hh ≠ 0) (hξ : p.xi ≠ 0) (i : Fin ρ) (j : Fin c) (hij : j.val + i.val + 1 = 2 * ρ) (col : Fin J.W.rank) : FW.γ (Sum.inr j) col = (s * (hh / p.xi) * (-1 / p.xi) ^ (ρ - 1 - i.val))⁻¹ * p.pr (J.d.qY i col)

theorem **jac_γ_inl** (h : p.IsJacFrame J s hh G₀ FW) (j : n') (col : Fin J.W.rank) : FW.γ (Sum.inl j) col = ∑ m, G₀.γ j m * p.pr (J.d.q0 m col)

theorem **gram_colMx** (m m' : Fin ρ) : (adj (colMx p FT r i b) · (gA T · rat X)) m m' = if m = i then cA D (p.ι (b * pairVal p FT X r m')) else 0

theorem **pairS_colMx** : pairS T (rat X) (colMx p FT r i b) = cA D (p.ι (b * pairVal p FT X r i))

theorem **uInv_colMx** (h : ∀ m : Fin ρ, m < i → pairVal p FT X r m = 0) : GlobalTheory.FinPl.UInv T (rat X) (colMx p FT r i b)

theorem **psi_symplO_colMx** {ψ : AddChar (AF D.host.F) ℂ} {ψk : AddChar k Circle} (hχ : p.IsChar ψ ψk) {S S₀ : HSp D} (d : HypDec S S₀ ρ) : ψ (symplO S T (d.extX T (rat X)) (d.extY T (colMx p FT r i b))) = (ψk (-(b * pairVal p FT X r i)) : ℂ)

theorem **zCol_apply** (r : Fin B.V.rank) : zCol B r 0 = (B.d.i₀ *ᵥ vecB B.o B.ζ) r

theorem **zCol_gram** {m : ℕ} (N : Matrix (Fin B.V.rank) (Fin m) D.host.E) : (zCol B)ᴴ * B.V.gram * N = (Matrix.of fun r (_ : Fin 1) => B.ζ r)ᴴ * (B.o.iBᴴ * (B.d.i₀ᴴ * B.V.gram * N))

theorem **zCol_e** : (zCol B)ᴴ * B.V.gram * B.d.e = 0

theorem **zCol_e'** : (zCol B)ᴴ * B.V.gram * B.d.e' = 0

theorem **zCol_A** : (zCol B)ᴴ * B.V.gram * (B.d.i₀ * B.o.iA) = 0

theorem **zCol_zCol** : ((zCol B)ᴴ * B.V.gram * zCol B) 0 0 = star B.ζ ·ᵥ B.L.gram *ᵥ B.ζ

theorem **qY_zCol** : B.d.qY * zCol B = 0

theorem **qY_A** : B.d.qY * (B.d.i₀ * B.o.iA) = 0

theorem **pairVal_castSucc** (r : n ⊕ Fin c) (i : Fin σ) : pairVal p FV (lastMat B.d (vecB B.o B.ζ)) r i.castSucc = ∑ x, p.pr (B.d.qY i x) * FV.β x r

theorem **pairVal_last** (r : n ⊕ Fin c) : pairVal p FV (lastMat B.d (vecB B.o B.ζ)) r (Fin.last σ) = ∑ x, p.pr (((zCol B)ᴴ * B.V.gram) 0 x) * FV.β x r

theorem **sum_core_zero** {m : ℕ} {Q : Matrix (Fin m) (Fin B.V.rank) D.host.E} (hQ : Q * (B.d.i₀ * B.o.iA) = 0) (i : Fin m) (a : n) : ∑ x, p.pr (Q i x) * ((B.d.i₀ * B.o.iA).map p.pr * FA.β) x a = 0

theorem **pairVal_mid** (h : p.IsBesFrame B s FA FV) (l : Fin c) (hl : l.val = σ) : pairVal p FV (lastMat B.d (vecB B.o B.ζ)) (Sum.inr l) (Fin.last σ) = s * p.hz B

theorem **pairVal_dual** (h : p.IsBesFrame B s FA FV) (l : Fin c) (i : Fin (σ + 1)) (hl : σ < l.val) (hli : l.val + i.val = 2 * σ) : pairVal p FV (lastMat B.d (vecB B.o B.ζ)) (Sum.inr l) i = s * (-p.hz B / p.xi) * (-1 / p.xi) ^ (σ - 1 - i.val)

theorem **pairVal_zero** (h : p.IsBesFrame B s FA FV) (r : n ⊕ Fin c) (i : Fin (σ + 1)) (hr : ∀ l, r = Sum.inr l → l.val + i.val ≠ 2 * σ) : pairVal p FV (lastMat B.d (vecB B.o B.ζ)) r i = 0

theorem **scale_ratio** {k : Type} [Field k] (a s ξ z t : k) (hs : s ≠ 0) (hξ : ξ ≠ 0) (hz : z ≠ 0) (ht : t ≠ 0) : -(a * (s * (-ξ ^ 2 * z / ξ) * (t * (-1 / ξ)))⁻¹ * (s * ξ * (-z / ξ) * t)) = a

theorem **GlobalTheory.LocMod.flatFB_trans_free** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), Free (σ + 1) (r, q) → (∀ i, q = Sum.inr i → σ < i.val) → ∀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFlatMapTranslationLawsJacobiToBesselFreeStatement

theorem **_root_.GlobalTheory.T.localFlatMap_translationLaws_jacobiToBessel_free** : ∀ (D : D7), GlobalTheory.FinPl.FlatLaws D → OmLoc D → ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ψk : AddChar k Circle), p.IsChar ψ ψk → ∀ (σ : ℕ) (J : JacLev D (σ + 1)) (B : BesLev D σ) (st : StepFB ψ J B) (s : k), s ≠ 0 → ∀ (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c : ℕ) (G₀ : Frame k J.W₀.rank n') (FW : Frame k J.W.rank (n' ⊕ Fin (c + 1))) (FA : Frame k B.A.rank n) (F₀ : Frame k B.V₀.rank (n ⊕ Unit)) (FV : Frame k B.V.rank (n ⊕ Fin c)), p.IsJacFrame J s (-p.xi ^ 2 * p.hz B) G₀ FW → p.IsCoreFrame B (s * p.xi) FA F₀ → p.IsBesFrame B (s * p.xi) FA FV → ∀ Tm, p.IsModel ψ FW FV Tm → ∀ (r : n ⊕ Fin c) (q : n' ⊕ Fin (c + 1)), Free (σ + 1) (r, q) → (∀ i

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFockIntertwinerAdjointExchangedPair

theorem **weak_deriv** : ∀ (H : Type) [NormedAddCommGroup H] [InnerProductSpace ℂ H] [CompleteSpace H] (f g : ℝ → H), Continuous g → (∀ (y : H) (t : ℝ), HasDerivAt (fun s => ⟪y, f s⟫_ℂ) ⟪y, g t⟫_ℂ t) → ∀ t, HasDerivAt f (g t) t

theorem **continuous_of_adj** {E F : Type} [NormedAddCommGroup E] [InnerProductSpace ℂ E] [CompleteSpace E] [NormedAddCommGroup F] [InnerProductSpace ℂ F] [CompleteSpace F] (T : E →ₗ[ℂ] F) (h : ∀ y : F, ∃ y' : E, ∀ x, ⟪y, T x⟫_ℂ = ⟪y', x⟫_ℂ) : Continuous T

theorem **conj_trace_of_lie** {n : ℕ} {G X : Matrix (Fin n) (Fin n) ℂ} (hG : G.det ≠ 0) (hX : Xᴴ * G + G * X = 0) : starRingEnd ℂ X.trace = -X.trace

theorem **gram_det_ne** {D : D7} {V : HSp D} {S : D.what₂.Tau → SPar} (M : DSModel V S) (τ : D.what₂.Tau) : (gramAt D V τ).det ≠ 0

theorem **exp_cont** {n : ℕ} (X : Matrix (Fin n) (Fin n) ℂ) : Continuous fun t : ℝ => NormedSpace.exp (t • X)

theorem **cont_rep_exp** (M : DSModel V S) (τ : D.what₂.Tau) {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) (x : M.H τ) : Continuous fun t : ℝ => M.rep τ (NormedSpace.exp (t • X)) x

theorem **inner_rep_exp** (M : DSModel V S) (τ : D.what₂.Tau) {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) (t : ℝ) (y x : M.H τ) : ⟪y, M.rep τ (NormedSpace.exp (t • X)) x⟫_ℂ = ⟪M.rep τ (NormedSpace.exp ((-t) • X)) y, x⟫_ℂ

theorem **neg_mem_lie** {τ : D.what₂.Tau} {X : Matrix (Fin V.rank) (Fin V.rank) ℂ} (hX : X ∈ lieAt V τ) : -X ∈ lieAt V τ

theorem **locS_add** (X Y : Matrix (Fin a) (Fin a) ℂ) : locS (b

theorem **locS_smul** (c : ℂ) (X : Matrix (Fin a) (Fin a) ℂ) : locS (b

theorem **locS_neg** (X : Matrix (Fin a) (Fin a) ℂ) : locS (b

theorem **locT_neg** (Y : Matrix (Fin b) (Fin b) ℂ) : locT (a

theorem **exists_adj** {L L' E F : Type} [AddCommGroup L] [Module ℂ L] [AddCommGroup L'] [Module ℂ L'] [NormedAddCommGroup E] [InnerProductSpace ℂ E] [NormedAddCommGroup F] [InnerProductSpace ℂ F] (A : L →ₗ[ℂ] F →ₗ[ℂ] E) (j : L ≃+ L') (hj : ∀ (c : ℂ) P, j (c • P) = conj c • j P) (h : ∀ P w, ∃ w', ∀ y, ⟪w, A P y⟫_ℂ = ⟪w', y⟫_ℂ) : ∃ A' : L' →ₗ[ℂ] E →ₗ[ℂ] F, ∀ Q w y, ⟪A' Q w, y⟫_ℂ = ⟪w, A (j.symm Q) y⟫_ℂ

theorem **adj_locIt** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (τ : D.what₂.Tau) (aS aT : ℚ) (A : LPol W.rank V.rank →ₗ[ℂ] M'.H τ →ₗ[ℂ] M.H τ), LocIt M M' τ aS aT A → ∃ A' : LPol V.rank W.rank →ₗ[ℂ] M.H τ →ₗ[ℂ] M'.H τ, LocIt M' M τ aT aS A'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalFockIntertwinerAdjointExchangedPairStatement

theorem **_root_.GlobalTheory.T.localFockIntertwiner_adjoint_exchangedPair** : ∀ (D : D7) (V W : HSp D) (S S' : D.what₂.Tau → SPar) (M : DSModel V S) (M' : DSModel W S') (τ : D.what₂.Tau) (aS aT : ℚ) (A : LPol W.rank V.rank →ₗ[ℂ] M'.H τ →ₗ[ℂ] M.H τ), LocIt M M' τ aS aT A → ∃ A' : LPol V.rank W.rank →ₗ[ℂ] M.H τ →ₗ[ℂ] M'.H τ, LocIt M' M τ aT aS A'

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalGroupActionInertSupportedPointsFrames

theorem **compW_eq_toCompl** (h : w.under (𝒪 D.host.F) = v) (a : AE D) : GlobalTheory.InertHomIdx.compW D w a = toCompl h (GlobalTheory.AdelicU.compFin D.host.F D.host.E v a)

theorem **secW_mul** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (z : w.adicCompletion D.host.E) (a : AE D) : secW hw h z * a = secW hw h (z * GlobalTheory.InertHomIdx.compW D w a)

theorem **mul_secW** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (z : w.adicCompletion D.host.E) (a : AE D) : a * secW hw h z = secW hw h (GlobalTheory.InertHomIdx.compW D w a * z)

theorem **mul_map_secW** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) {m n p : ℕ} (A : Matrix (Fin m) (Fin n) (AE D)) (M : Matrix (Fin n) (Fin p) (w.adicCompletion D.host.E)) : A * M.map (secW hw h) = (A.map (GlobalTheory.InertHomIdx.compW D w) * M).map (secW hw h)

theorem **map_secW_mul** (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) {m n p : ℕ} (M : Matrix (Fin m) (Fin n) (w.adicCompletion D.host.E)) (B : Matrix (Fin n) (Fin p) (AE D)) : M.map (secW hw h) * B = (M * B.map (GlobalTheory.InertHomIdx.compW D w)).map (secW hw h)

theorem **GlobalTheory.InertFrameAct.actO_uPt** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (S T : HSp D) (k : w.adicCompletion D.host.E) (aS : GL (Fin S.rank) (w.adicCompletion D.host.E)) (aT : GL (Fin T.rank) (w.adicCompletion D.host.E)) (gS : Gv S v) (gT : Gv T v) (x : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E)), actO S T (embV S v gS) (embV T v gT) (uPt hw h k aS aT x) = uPt hw h k aS aT ((aT⁻¹ * locMat D.sp.cm.sig T.gram h gT * aT).1 * x * (aS⁻¹ * locMat D.sp.cm.sig S.gram h gS⁻¹ * aS).1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalGroupActionInertSupportedPointsFramesStatement

theorem **_root_.GlobalTheory.T.local_group_action_inert_supported_points_frames** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (w : HeightOneSpectrum (𝒪 D.host.E)) (hw : IsInertAt D.host.F D.host.E v w) (h : w.under (𝒪 D.host.F) = v) (S T : HSp D) (k : w.adicCompletion D.host.E) (aS : GL (Fin S.rank) (w.adicCompletion D.host.E)) (aT : GL (Fin T.rank) (w.adicCompletion D.host.E)) (gS : Gv S v) (gT : Gv T v) (x : Matrix (Fin T.rank) (Fin S.rank) (w.adicCompletion D.host.E)), actO S T (embV S v gS) (embV T v gT) (uPt hw h k aS aT x) = uPt hw h k aS aT ((aT⁻¹ * locMat D.sp.cm.sig T.gram h gT * aT).1 * x * (aS⁻¹ * locMat D.sp.cm.sig S.gram h gS⁻¹ * aS).1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalGroupsCharactersFamily

theorem **isComp_cmp** (q : GlobalTheory.FinPl.SplitPt D k) (χ : (AE D)ˣ →* Circle) : (ofSplit q).IsComp χ (cmp q χ)

theorem **cmp_apply** (q : GlobalTheory.FinPl.SplitPt D k) (χ : (AE D)ˣ →* Circle) (x : kˣ) : ((cmp q χ x : ℂˣ) : ℂ) = (χ (GlobalTheory.LocChar.unitsHom (ofSplit q) x) : ℂ)

theorem **unitsHom_int** (q : GlobalTheory.FinPl.SplitPt D k) (y : kˣ) (hy : valuation k (y : k) = 1) : finPart D ((GlobalTheory.LocChar.unitsHom (ofSplit q) y : (AE D)ˣ) : AE D) ∈ intAE D

theorem **unit_trivial** (q : GlobalTheory.FinPl.SplitPt D k) {χ : (AE D)ˣ →* Circle} (h : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D χ q.v) (x : kˣ) (hx : valuation k (x : k) = 1) : χ (GlobalTheory.LocChar.unitsHom (ofSplit q) x) = 1

theorem **val_eq_one_of_sub** (x : kˣ) (hx : valuation k ((x : k) - 1) < 1) : valuation k (x : k) = 1

theorem **under_ne** (hP : PkHyp W) (t : W.P.T) {PP P' : Formalism.L2.DIV.Pl D.host.E} (hPg : PP ∈ (pk W).Pg) (hQ : P' ∈ (pk W).Q t) : P'.under (𝒪 D.host.F) ≠ PP.under (𝒪 D.host.F)

theorem **xi_charUnramAt** (hP : PkHyp W) (t : W.P.T) {PP : Formalism.L2.DIV.Pl D.host.E} (hPg : PP ∈ (pk W).Pg) : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D ((pk W).toLevCh.xi t) (PP.under (𝒪 D.host.F))

theorem **eta_charUnramAt** (hP : PkHyp W) (v : HeightOneSpectrum (𝒪 D.host.F)) : GlobalTheory.IndefiniteRowSideFullHecke.UnramifiedCharacter.CharUnramAt D (pk W).eta v

theorem **gLoc_chars** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX}

(W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (f : FamCh W u) (v : VCom f.c), f.RowOK → VTie f v → ∀ t (n : ℕ) (G : Fin n → GLoc v) pl, GRead f G pl → ∀ i, (G i).Chars (f.t t)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalGroupsCharactersFamilyStatement

theorem **_root_.GlobalTheory.T.local_groups_characters_family** : ∀ {D : D7} {L : SizeData₂ D} {hT : GatTies D} {Gi : Formalism.MainNotions.G.Inst D L} {Gl : Formalism.MainNotions.G.LatInst D} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX D} {cX : ExtraC} {ex : Extra D L hT Gi Gl Kc Yd cX} (W : RealPack ex) (u : D.what₂.Word), PkHyp W → ∀ (f : FamCh W u) (v : VCom f.c), f.RowOK → VTie f v → ∀ t (n : ℕ) (G : Fin n → GLoc v) pl, GRead f G pl → ∀ i, (G i).Chars (f.t t)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalGroupsMatrixReading

theorem **locF_prod_level** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u) (v : VCom f.c) (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GRead f G pl → ∀ (j : ℕ) (y : ∀ PP : Formalism.L2.DIV.Pl D.host.E, Matrix (Fin (Yc D (j + 1)).rank) (Fin (Yc D j).rank) (Formalism.L2.DIV.kP PP)), omegaG (datY _ D.R.r D.R.t j (r_le_Y j) (pk W).Pg ∅ (lvP W) (pk W).chi (fun _ => 1) Frm.B) ((pk W).chi (j + 1)) y = ∏ i, GlobalTheory.LocMod.StepVec.locF D.R.t j (r_le_Y j) (GlobalTheory.MovedDat.RdAt (G i) (pl i).1) (fun x => (G i).S.P.lam x) (fun c => (pk W).chi c (pl i).1) (y (pl i).1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalGroupsMatrixReadingStatement

theorem **_root_.GlobalTheory.T.local_groups_matrix_reading** : ∀ (D : D7) (L : SizeData₂ D) (hT : GatTies D) (Gi : Formalism.MainNotions.G.Inst D L) (Gl : Formalism.MainNotions.G.LatInst D) (Kc : Formalism.MainNotions.G.Consts) (Yd : DivX D) (cX : ExtraC) (ex : Extra D L hT Gi Gl Kc Yd cX) (W : RealPack ex) (u : D.what₂.Word) (f : FamCh W u) (v : VCom f.c) (n : ℕ) (G : Fin n → GLoc v) (pl : Fin n ≃ (pk W).Pg), GRead f G pl → ∀ (j : ℕ) (y : ∀ PP : Formalism.L2.DIV.Pl D.host.E, Matrix (Fin (Yc D (j + 1)).rank) (Fin (Yc D j).rank) (Formalism.L2.DIV.kP PP)), omegaG (datY _ D.R.r D.R.t j (r_le_Y j) (pk W).Pg ∅ (lvP W) (pk W).chi (fun _ => 1) Frm.B) ((pk W).chi (j + 1)) y = ∏ i, GlobalTheory.LocMod.StepVec.locF D.R.t j (r_le_Y j) (GlobalTheory.MovedDat.RdAt (G i) (pl i).1) (fun x => (G i).S.P.lam x) (fun c => (pk W).chi c (pl i).1) (y (pl i).1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalIntegersTensorCoordsPrimeToDiscriminant

theorem **isIntegral_of_trace_norm** (w : E) (t n : 𝒪 F) (ht : w + σ w = algebraMap F E t) (hn : w * σ w = algebraMap F E n) : IsIntegral ℤ w

theorem **exists_coords** (h : IsCMExt F E σ) (w : E) (t n : 𝒪 F) (ht : w + σ w = algebraMap F E t) (hn : w * σ w = algebraMap F E n) (x : 𝒪 E) : ∃ a b : 𝒪 F, (x : E) * algebraMap F E ((t ^ 2 - 4 * n : 𝒪 F) : F) = algebraMap F E a + algebraMap F E b * w

theorem **tmul_move** {K : Type} [Field K] [Algebra F K] (m : K) (r : F) (e : E) : (algebraMap F K r * m) ⊗ₜ[F] e = m ⊗ₜ[F] (algebraMap F E r * e)

theorem **coords_add** {K : Type} [Field K] [Algebra F K] (a b a' b' : K) (w : E) : (a ⊗ₜ[F] (1 : E) + b ⊗ₜ[F] w) + (a' ⊗ₜ[F] (1 : E) + b' ⊗ₜ[F] w) = (a + a') ⊗ₜ[F] (1 : E) + (b + b') ⊗ₜ[F] w

theorem **coords_neg** {K : Type} [Field K] [Algebra F K] (a b : K) (w : E) : -(a ⊗ₜ[F] (1 : E) + b ⊗ₜ[F] w) = (-a) ⊗ₜ[F] (1 : E) + (-b) ⊗ₜ[F] w

theorem **mem_intV_coords** : ∀ (D : D7) (v : Pl D) (w : D.host.E) (t n : 𝒪 D.host.F), w + D.sp.cm.sig w = algebraMap D.host.F D.host.E t → w * D.sp.cm.sig w = algebraMap D.host.F D.host.E n → t ^ 2 - 4 * n ∉ v.asIdeal → ∀ z ∈ intV D v, ∃ a ∈ v.adicCompletionIntegers D.host.F, ∃ b ∈ v.adicCompletionIntegers D.host.F, z = a ⊗ₜ[D.host.F] (1 : D.host.E) + b ⊗ₜ[D.host.F] w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalIntegersTensorCoordsPrimeToDiscriminantStatement

theorem **_root_.GlobalTheory.T.localIntegers_tensor_coords_prime_to_discriminant** : ∀ (D : D7) (v : Pl D) (w : D.host.E) (t n : 𝒪 D.host.F), w + D.sp.cm.sig w = algebraMap D.host.F D.host.E t → w * D.sp.cm.sig w = algebraMap D.host.F D.host.E n → t ^ 2 - 4 * n ∉ v.asIdeal → ∀ z ∈ intV D v, ∃ a ∈ v.adicCompletionIntegers D.host.F, ∃ b ∈ v.adicCompletionIntegers D.host.F, z = a ⊗ₜ[D.host.F] (1 : D.host.E) + b ⊗ₜ[D.host.F] w

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalInvariantIntegralsSymmetricHaarSubgroups

theorem **eq_of_mass_one** {G : Type} [TopologicalSpace G] [Group G] [IsTopologicalGroup G] [MeasurableSpace G] [BorelSpace G] [LocallyCompactSpace G] [SecondCountableTopology G] (κ μ : Measure G) [μ.IsHaarMeasure] [IsFiniteMeasureOnCompacts κ] [κ.IsMulLeftInvariant] (S : Set G) (hS : μ S = 1) (hκ : κ S = 1) : κ = μ

theorem **quot_int_haar** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ) (m : Measure (GL n k)) (G : Measure (GL (n ⊕ Fin c) k)), m.IsHaarMeasure → G.IsHaarMeasure → m (KO k n) = 1 → (c + 1 = 2 * ρ → ∀ μR : Measure (R k n c ρ), μR.IsHaarMeasure → μR {x | x.1 ∈ KO k (n ⊕ Fin c)} = 1 → μR.IsInvInvariant ∧ ∀ f z, IsQuotInt ν m G ρ f z → IsQuotIntL1 μR G f z) ∧ (1 ≤ ρ → c = 2 * ρ → ∀ μR : Measure (RFJ k n c ρ), μR.IsHaarMeasure → μR {x | x.1 ∈ KO k (n ⊕ Fin c)} = 1 → μR.IsInvInvariant ∧ ∀ f z, IsQuotIntFJ ν m G ρ f z → IsQuotIntL1 μR G f z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalInvariantIntegralsSymmetricHaarSubgroupsStatement

theorem **_root_.GlobalTheory.T.localInvariantIntegrals_symmetric_haarSubgroups** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → ∀ (n : Type) [Fintype n] [DecidableEq n] (c ρ : ℕ) (m : Measure (GL n k)) (G : Measure (GL (n ⊕ Fin c) k)), m.IsHaarMeasure → G.IsHaarMeasure → m (KO k n) = 1 → (c + 1 = 2 * ρ → ∀ μR : Measure (R k n c ρ), μR.IsHaarMeasure → μR {x | x.1 ∈ KO k (n ⊕ Fin c)} = 1 → μR.IsInvInvariant ∧ ∀ f z, IsQuotInt ν m G ρ f z → IsQuotIntL1 μR G f z) ∧ (1 ≤ ρ → c = 2 * ρ → ∀ μR : Measure (RFJ k n c ρ), μR.IsHaarMeasure → μR {x | x.1 ∈ KO k (n ⊕ Fin c)} = 1 → μR.IsInvInvariant ∧ ∀ f z, IsQuotIntFJ ν m G ρ f z → IsQuotIntL1 μR G f z)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalLatticeExistsScalarBoundAttained

theorem **lattice_exists_scalar_bound_attained** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] (L : Submodule 𝒪[k] (n → k)), L.IsLattice k → Nonempty n → ∃ c : k, c ≠ 0 ∧ (∀ x ∈ L, ∀ j, c⁻¹ * x j ∈ 𝒪[k]) ∧ ∃ x ∈ L, ∃ j, x j = c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalLatticeExistsScalarBoundAttainedStatement

theorem **_root_.GlobalTheory.T.localLattice_exists_scalar_bound_attained** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (n : Type) [Fintype n] (L : Submodule 𝒪[k] (n → k)), L.IsLattice k → Nonempty n → ∃ c : k, c ≠ 0 ∧ (∀ x ∈ L, ∀ j, c⁻¹ * x j ∈ 𝒪[k]) ∧ ∃ x ∈ L, ∃ j, x j = c

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalLevelComparison

theorem **prV_comm** (v : GlobalTheory.FinPl.Pl D) (z : Af ɪD.host.F ⊗[ɪD.host.F] D.host.E) : Formalism.UG.prV (E

theorem **prV_toEAf_congr** (v : GlobalTheory.FinPl.Pl D) {y y' : AE D} (h : GlobalTheory.FinPl.compAt D v y = GlobalTheory.FinPl.compAt D v y') : Formalism.UG.prV (E

theorem **bc_eq_one_of_off** (V : HSp D) {n : ℕ} (hn : V.rank = n) {P : GlobalTheory.FinPl.Pl D → Prop} {v : GlobalTheory.FinPl.Pl D} (hv : P v) {g : GA D V} (hg : g ∈ GlobalTheory.FinPl.grpOff V P) {x : GL (Fin n) (EF D)} (hx : x.1 = finM V hn g) : Formalism.UG.bc (Formalism.UG.prV (E

theorem **exists_splitPt** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} {p : ι → GlobalTheory.FinPl.SplitPt (I.rec7 l Z) k} {β : ∀ j i, GlobalTheory.FinPl.Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)} {Q : ∀ j, ι → GlobalTheory.LatPair.Pair k (Fin ((I.rec7 l Z).sp.X j).rank)} {L : ∀ j, GlobalTheory.FinPl.Lat ((I.rec7 l Z).sp.X j)} (hS : StdT I l Z u p β Q L) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)) (hv : ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ v.asIdeal) : ∃ i, v = (p i).v

theorem **uloc_le_of_std** (hOk : I.Ok) (hP : TC.Pins B X) {a a' : I.ι} {Ch Ch' : ℕ → Type} {A : TwDat I l Z u a Ch} {A' : TwDat I l Z u a' Ch'} (hA : A.Is) (hA' : A'.Is) {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {ι : Type} {p : ι → GlobalTheory.FinPl.SplitPt (I.rec7 l Z) k} {β : ∀ j i, GlobalTheory.FinPl.Frame (p i) ((I.rec7 l Z).sp.X j) (Fin ((I.rec7 l Z).sp.X j).rank)} {Q : ∀ j, ι → GlobalTheory.LatPair.Pair k (Fin ((I.rec7 l Z).sp.X j).rank)} {L : ∀ j, GlobalTheory.FinPl.Lat ((I.rec7 l Z).sp.X j)} (hS : StdT I l Z u p β Q L) (J : List ι) (hn : J.Nodup) (hall : ∀ i, i ∈ J) (j : ℕ) (hj : j ≤ B.R.m) (b : Ch j) (b' : Ch' j) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)) (hv : ∃ i, v = (p i).v) : (A.K j hj b).lev.Uloc v ≤ (A'.K j hj b').lev.Uloc v

theorem **uloc_le_levelPrime** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → a.True → ∀ j (hj : j + 1 = B.R.m), ∀ t (J : QJ a t), J.Reads → ∀ (α : J.Ch j) t' (J' : QJ a t'), J'.Reads → ∀ (α' : J'.Ch j) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)), ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ v.asIdeal → (J.A.K j (show j ≤ B.R.m by omega) (J.ι j α)).lev.Uloc v ≤ (J'.A.K j (show j ≤ B.R.m by omega) (J'.ι j α')).lev.Uloc v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalLevelComparisonStatement

theorem **_root_.GlobalTheory.T.true_geometry_coset_value** : ∀ {B : GlobalTheory.Mem.Base} {X : PreX B.toPre} {I : TC B X} {l : I.Laws} {Z : TC.Later B (I.cols l)} {L : SizeData₂ (I.rec7 l Z)} {hT : GatTies (I.rec7 l Z)} {Gi : Formalism.MainNotions.G.Inst (I.rec7 l Z) L} {Gl : Formalism.MainNotions.G.LatInst (I.rec7 l Z)} {Kc : Formalism.MainNotions.G.Consts} {Yd : DivX (I.rec7 l Z)} {cX : ExtraC} {ex : Extra (I.rec7 l Z) L hT Gi Gl Kc Yd cX} (W : RealPack ex), AllTrue W → ∀ (u : B.Word) (a : GeoAll W u), a.Is → a.True → ∀ j (hj : j + 1 = B.R.m), ∀ t (J : QJ a t), J.Reads → ∀ (α : J.Ch j) t' (J' : QJ a t'), J'.Reads → ∀ (α' : J'.Ch j) (v : HeightOneSpectrum (𝒪 (I.rec7 l Z).host.F)), ((B.T u : ℕ) : 𝒪 (I.rec7 l Z).host.F) ∈ v.asIdeal → (J.A.K j (show j ≤ B.R.m by omega) (J.ι j α)).lev.Uloc v ≤ (J'.A.K j (show j ≤ B.R.m by omega) (J'.ι j α')).lev.Uloc v

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalLevelZeroStabilizerMem

theorem **matrix_in_frame** {k n Y A : Type} [CommRing k] [Fintype n] [DecidableEq n] [Fintype Y] [DecidableEq Y] [Fintype A] [DecidableEq A] (β : Matrix n Y k) (γ : Matrix Y n k) (h : γ * β = 1) (b : GL Y k) (σ : A ≃ Y) (e : A ≃ n) (X : Matrix A A k) (m : Matrix n n k) (hm : ((β * b.val).submatrix id fun i => σ (e.symm i)) * m = β * (b.val * X.submatrix σ.symm σ.symm * (b⁻¹).val) * γ * (β * b.val).submatrix id fun i => σ (e.symm i)) : m = X.submatrix e.symm e.symm

theorem **toEAf_onto** (D : D7) : Function.Surjective (toEAf D)

theorem **finE_val** {D : D7} (V : HSp D) (g : GA D V) : (Lev.finE V g).val = (Formalism.L2.Theta.mt V g).map (toEAf D)

theorem **upperLeftBlock_apply_of_eq** {κ : Type} [CommRing κ] {r n : ℕ} (hr : r ≤ n) (hn : n ≤ r) (g : GL (Fin n) κ) (i j : Fin r) : (upperLeftBlock hr g).val i j = g.val (Fin.castLE hr i) (Fin.castLE hr j)

theorem **red_of_isLoc** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP), d.Base → ∀ (hr : D.R.r ≤ (d.st 0).V.rank) (g : GA D (d.st 0).V) (x : G.S.P.J), (GlobalTheory.LocMod.ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) (Nat.zero_add _)) g (GlobalTheory.LevZeroGen.upZero G.S (Fin D.R.t) x.1) → finE (d.st 0).V ((d.t 0)⁻¹ * g * d.t 0) ∈ St (d.st 0).β → G.Red x (upperLeftBlock hr (redFn (d.st 0).β PP (finE (d.st 0).V ((d.t 0)⁻¹ * g * d.t 0))))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalLevelZeroStabilizerMemStatement

theorem **_root_.GlobalTheory.T.local_level_zero_stabilizer_mem** : ∀ (D : D7) (u : D.what₂.Word) (d : GlobalTheory.SeeSaw.Dat D u) (τ : ∀ j h, Matrix (Fin (d.bes j 0 h).L.rank) (Fin d.Lz.rank) D.host.E) (PP : Pl D.host.E) (G : GlobalTheory.LocMod.GAt d τ PP), d.Base → ∀ (hr : D.R.r ≤ (d.st 0).V.rank) (g : GA D (d.st 0).V) (x : G.S.P.J), (GlobalTheory.LocMod.ofSplit G.q).IsLoc (G.FY 0 (G.s + 1) (Nat.zero_add _)) g (GlobalTheory.LevZeroGen.upZero G.S (Fin D.R.t) x.1) → finE (d.st 0).V ((d.t 0)⁻¹ * g * d.t 0) ∈ St (d.st 0).β → G.Red x (upperLeftBlock hr (redFn (d.st 0).β PP (finE (d.st 0).V ((d.t 0)⁻¹ * g * d.t 0))))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelBoxProductOrthogonalSumTarget

theorem **map_pr_mul** (p : Place D k) {l m m' : Type} [Fintype m] (M : Matrix l m D.host.E) (N : Matrix m m' D.host.E) : (M * N).map p.pr = M.map p.pr * N.map p.pr

theorem **incA_mul_ι** (p : Place D k) (a : D.host.E) (x : k) : (incA D a : AE D) * p.ι x = p.ι (p.pr a * x)

theorem **rat_mul_ι** (p : Place D k) {l m : ℕ} {Y : Type} [Fintype Y] (M : Matrix (Fin l) (Fin m) D.host.E) (N : Matrix (Fin m) Y k) : (rat M : Matrix (Fin l) (Fin m) (AE D)) * (N.map p.ι) = (M.map p.pr * N).map p.ι

theorem **resT_up** (p : Place D k) {S X T L : HSp D} (o : OrthDec X T L) (FS : Frame k S.rank n) (FT : Frame k T.rank Z) (FL : Frame k L.rank Unit) (FX : Frame k X.rank (Z ⊕ Unit)) (hβ : FX.β = fromCols (o.iA.map p.pr * FT.β) (o.iB.map p.pr * FL.β)) (x : Matrix (Z ⊕ Unit) n k) : o.resT S (p.up FS FX x) = (p.up FS FT x.toRows₁, p.up FS FL x.toRows₂)

theorem **hO_add_left'** (S T : HSp D) (x x' y : WO S T) : hO S T (x + x') y = hO S T x y + hO S T x' y

theorem **symplO_add_left'** (S T : HSp D) (x x' y : WO S T) : symplO S T (x + x') y = symplO S T x y + symplO S T x' y

theorem **symplO_zero_left'** (S T : HSp D) (y : WO S T) : symplO S T 0 y = 0

theorem **isThetaFunO_zero'** (ψ : AddChar (AF D.host.F) ℂ) (S T : HSp D) : IsThetaFunO ψ S T 0 where

theorem **isThetaFunO_add'** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {f g : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) (hg : IsThetaFunO ψ S T g) : IsThetaFunO ψ S T (f + g) where

theorem **isThetaFunO_smul'** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (c • f) where

theorem **isLocallyConstant_of_near** {G : Type} [TopologicalSpace G] [AddCommGroup G] [IsTopologicalAddGroup G] (χ : G → ℂ) (hχ : ∀ a b, χ (a + b) = χ a * χ b) (t₀ : G) (h0 : χ t₀ ≠ 0) (U : Set G) (hU : U ∈ nhds t₀) (hc : ∀ t ∈ U, χ t = χ t₀) : IsLocallyConstant χ

theorem **isModel_boxT** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S X T L'' : HSp D) (o'' : OrthDec X T L'') (n Z : Type) [Fintype n] [DecidableEq n] [Fintype Z] [DecidableEq Z] (FS : Frame k S.rank n) (FT : Frame k T.rank Z) (FL : Frame k L''.rank Unit) (FX : Frame k X.rank (Z ⊕ Unit)), FX.β = fromCols (o''.iA.map p.pr * FT.β) (o''.iB.map p.pr * FL.β) → ∀ T₂ TL, p.IsModel ψ FS FT T₂ → p.IsModel ψ FS FL TL → ∀ TX : SB (Matrix (Z ⊕ Unit) n k) →ₗ[ℂ] (WO S X → ℂ), (∀ (a : SB (Matrix Z n k)) (b : SB (Matrix Unit n k)) (Φ : SB (Matrix (Z ⊕ Unit) n k)), (∀ x, Φ.1 x = a.1 x.toRows₁ * b.1 x.toRows₂) → TX Φ = o''.boxT S (T₂ a) (TL b)) → p.IsModel ψ FS FX TX

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelBoxProductOrthogonalSumTargetStatement

theorem **_root_.GlobalTheory.T.local_model_box_product_orthogonal_sum_target** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (S X T L'' : HSp D) (o'' : OrthDec X T L'') (n Z : Type) [Fintype n] [DecidableEq n] [Fintype Z] [DecidableEq Z] (FS : Frame k S.rank n) (FT : Frame k T.rank Z) (FL : Frame k L''.rank Unit) (FX : Frame k X.rank (Z ⊕ Unit)), FX.β = fromCols (o''.iA.map p.pr * FT.β) (o''.iB.map p.pr * FL.β) → ∀ T₂ TL, p.IsModel ψ FS FT T₂ → p.IsModel ψ FS FL TL → ∀ TX : SB (Matrix (Z ⊕ Unit) n k) →ₗ[ℂ] (WO S X → ℂ), (∀ (a : SB (Matrix Z n k)) (b : SB (Matrix Unit n k)) (Φ : SB (Matrix (Z ⊕ Unit) n k)), (∀ x, Φ.1 x = a.1 x.toRows₁ * b.1 x.toRows₂) → TX Φ = o''.boxT S (T₂ a) (TL b)) → p.IsModel ψ FS FX TX

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelRestrictedToLineSlice

theorem **ι_mul** (x : k) (b : AE D) : p.ι x * b = p.ι (x * p.π b)

theorem **map_ι_mul** {a b c : Type} [Fintype b] (M : Matrix a b k) (N : Matrix b c (AE D)) : M.map p.ι * N = (M * N.map p.π).map p.ι

theorem **frame_card** {N : ℕ} {Y : Type} [Fintype Y] [DecidableEq Y] (F : Frame k N Y) : Fintype.card Y = N

theorem **box_apply** (Ξ : SB (Matrix n n' k)) (c : SB (Matrix n Unit k)) (x : Matrix n (n' ⊕ Unit) k) : (box Ξ c).1 x = Ξ.1 x.toCols₁ * c.1 x.toCols₂

theorem **heis_slice** {S A L T : HSp D} (d : OrthDec S A L) (ψ : AddChar (AF D.host.F) ℂ) (f : WO S T → ℂ) (u : WO A T) (w v : WO L T) : heisO ψ S T ((d.resS T).symm (0, v)) f ((d.resS T).symm (u, w)) = heisO ψ L T v (fun w => f ((d.resS T).symm (u, w))) w

theorem **symplO_zero_right'** (S T : HSp D) (x : WO S T) : symplO S T x 0 = 0

theorem **isHalf_slice** {S A L T : HSp D} (d : OrthDec S A L) {a : AE D} {y : WO L T} (hy : IsHalf a y) : IsHalf a ((d.resS T).symm (0, y))

theorem **beta_eq** (h : p.IsCoreFrame B s FA F₀) (hL : ∀ r, FL.β r () = s * p.pr (B.ζ r)) : F₀.β = fromCols (B.o.iA.map (p.π.comp (incA D)) * FA.β) (B.o.iB.map (p.π.comp (incA D)) * FL.β)

theorem **rat_map_π** {a b : ℕ} (M : Matrix (Fin a) (Fin b) D.host.E) : (rat M).map p.π = M.map (p.π.comp (incA D))

theorem **up_fromCols** (h : p.IsCoreFrame B s FA F₀) (hL : ∀ r, FL.β r () = s * p.pr (B.ζ r)) {S : HSp D} (FS : Frame k S.rank n) (x : Matrix n Unit k) : p.up F₀ FS (fromCols 0 x) = (B.o.resS S).symm (0, p.up FL FS x)

theorem **resS_up_snd** (h : p.IsCoreFrame B s FA F₀) (hL : ∀ r, FL.β r () = s * p.pr (B.ζ r)) {S : HSp D} (FS : Frame k S.rank n) (t : Matrix n (n' ⊕ Unit) k) : (B.o.resS S (p.up F₀ FS t)).2 = p.up FL FS t.toCols₂

theorem **toCols_add** {k n n' : Type} [Field k] (t : Matrix n (n' ⊕ Unit) k) (x : Matrix n Unit k) : (t + fromCols (0 : Matrix n n' k) x).toCols₁ = t.toCols₁ ∧ (t + fromCols (0 : Matrix n n' k) x).toCols₂ = t.toCols₂ + x

theorem **slice_line** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∃ FL : Frame k B.L.rank Unit, (∀ r, FL.β r () = s * p.pr (B.ζ r)) ∧ ∀ (Ξ : SB (Matrix n n' k)) (u₀ : WO B.A S), ∃ T', p.IsModel ψ FL FS T' ∧ ∀ (c : SB (Matrix n Unit k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ * c.1 (Matrix.of fun r _ => x r (Sum.inr ()))) → ∀ w, T' c w = T₀ Ψ ((B.o.resS S).symm (u₀, w))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelRestrictedToLineSliceStatement

theorem **_root_.GlobalTheory.T.local_model_restricted_to_line_slice** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∃ FL : Frame k B.L.rank Unit, (∀ r, FL.β r () = s * p.pr (B.ζ r)) ∧ ∀ (Ξ : SB (Matrix n n' k)) (u₀ : WO B.A S), ∃ T', p.IsModel ψ FL FS T' ∧ ∀ (c : SB (Matrix n Unit k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ * c.1 (Matrix.of fun r _ => x r (Sum.inr ()))) → ∀ w, T' c w = T₀ Ψ ((B.o.resS S).symm (u₀, w))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelSliceFirstSummand

theorem **heis_slice_left** {S A L T : HSp D} (d : OrthDec S A L) (ψ : AddChar (AF D.host.F) ℂ) (f : WO S T → ℂ) (u u' : WO A T) (w : WO L T) : heisO ψ S T ((d.resS T).symm (u', 0)) f ((d.resS T).symm (u, w)) = heisO ψ A T u' (fun u => f ((d.resS T).symm (u, w))) u

theorem **isHalf_slice_left** {S A L T : HSp D} (d : OrthDec S A L) {a : AE D} {y : WO A T} (hy : IsHalf a y) : IsHalf a ((d.resS T).symm (y, 0))

theorem **up_fromCols_left** (h : p.IsCoreFrame B s FA F₀) (hL : ∀ r, FL.β r () = s * p.pr (B.ζ r)) {S : HSp D} (FS : Frame k S.rank n) (x : Matrix n n' k) : p.up F₀ FS (fromCols x 0) = (B.o.resS S).symm (p.up FA FS x, 0)

theorem **resS_up_fst** (h : p.IsCoreFrame B s FA F₀) (hL : ∀ r, FL.β r () = s * p.pr (B.ζ r)) {S : HSp D} (FS : Frame k S.rank n) (t : Matrix n (n' ⊕ Unit) k) : (B.o.resS S (p.up F₀ FS t)).1 = p.up FA FS t.toCols₁

theorem **toCols_add_left** {k n n' : Type} [Field k] (t : Matrix n (n' ⊕ Unit) k) (x : Matrix n n' k) : (t + fromCols x (0 : Matrix n Unit k)).toCols₁ = t.toCols₁ + x ∧ (t + fromCols x (0 : Matrix n Unit k)).toCols₂ = t.toCols₂

theorem **slice_left** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∀ (c : SB (Matrix n Unit k)) (w₀ : WO B.L S), ∃ T', p.IsModel ψ FA FS T' ∧ ∀ (Ξ : SB (Matrix n n' k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ ∗ c.1 (Matrix.of fun r _ => x r (Sum.inr ()))) → ∀ u, T' Ξ u = T₀ Ψ ((B.o.resS S).symm (u, w₀))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelSliceFirstSummandStatement

theorem **_root_.GlobalTheory.T.localModel_slice_firstSummand** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ) (ρ : ℕ) (B : BesLev D ρ) (S : HSp D) (s : k) (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (FA : Frame k B.A.rank n') (F₀ : Frame k B.V₀.rank (n' ⊕ Unit)) (FS : Frame k S.rank n), p.IsCoreFrame B s FA F₀ → ∀ T₀, p.IsModel ψ F₀ FS T₀ → ∀ (c : SB (Matrix n Unit k)) (w₀ : WO B.L S), ∃ T', p.IsModel ψ FA FS T' ∧ ∀ (Ξ : SB (Matrix n n' k)) (Ψ : SB (Matrix n (n' ⊕ Unit) k)), (∀ x, Ψ.1 x = Ξ.1 x.toCols₁ ∗ c.1 (Matrix.of fun r _ => x r (Sum.inr ()))) → ∀ u, T' Ξ u = T₀ Ψ ((B.o.resS S).symm (u, w₀))

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelWeilOperatorTrivialAtPlaceIsModel

theorem **placeIdem_mul_eq_zero** {D : D7} (v : GlobalTheory.FinPl.Pl D) (ε : AF D.host.F) (hε : IsPlaceIdem D v ε) (a : AE D) (ha : a ∈ GlobalTheory.FinPl.idOff D (· = v)) : algebraMap (AF D.host.F) (AE D) ε ∗ a = 0

theorem **ent_of_mem_grpOff** {D : D7} {V : HSp D} {P : GlobalTheory.FinPl.Pl D → Prop} {g : GA D V} (hg : g ∈ GlobalTheory.FinPl.grpOff V P) : ∀ i j, mt V g i j − (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D)) i j ∈ GlobalTheory.FinPl.idOff D P

theorem **mul_mul_eq_self** {D : D7} {m n : ℕ} (v : GlobalTheory.FinPl.Pl D) (ε : AF D.host.F) (hε : IsPlaceIdem D v ε) (A : Matrix (Fin m) (Fin m) (AE D)) (B : Matrix (Fin n) (Fin n) (AE D)) (y : Matrix (Fin m) (Fin n) (AE D)) (hA : ∀ i j, A i j − (1 : Matrix (Fin m) (Fin m) (AE D)) i j ∈ GlobalTheory.FinPl.idOff D (· = v)) (hB : ∀ i j, B i j − (1 : Matrix (Fin n) (Fin n) (AE D)) i j ∈ GlobalTheory.FinPl.idOff D (· = v)) (hy : ∀ i j, algebraMap (AF D.host.F) (AE D) ε ∗ y i j = y i j) : HMul.hMul (HMul.hMul A y) B = y

theorem **actO_eq_self** {D : D7} {V W : HSp D} (v : GlobalTheory.FinPl.Pl D) (ε : AF D.host.F) (hε : IsPlaceIdem D v ε) {g : GA D V} (hg : g ∈ GlobalTheory.FinPl.grpOff V (· = v)) {h : GA D W} (hh : h ∈ GlobalTheory.FinPl.grpOff W (· = v)) (y : WO V W) (hy : IsHalf (algebraMap (AF D.host.F) (AE D) ε) y) : actO V W g h y = y

theorem **isModel_weil_off** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (V W : HSp D) (χV χW : (AE D)ˣ →∗ Circle) (ω : Om V W), IsWeilDatumO ψ V W χV χW ω → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FV : Frame k V.rank Y') (FW : Frame k W.rank Y) (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO V W → ℂ)), (ofSplit q).IsModel ψ FV FW Tm → ∀ g ∈ GlobalTheory.FinPl.grpOff V (· = q.v), ∀ h ∈ GlobalTheory.FinPl.grpOff W (· = q.v), (ofSplit q).IsModel ψ FV FW ((ω (g, h)) ∘ₗ Tm)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalModelWeilOperatorTrivialAtPlaceIsModelStatement

theorem **_root_.GlobalTheory.T.local_model_weil_operator_trivialAtPlace_isModel** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (q : GlobalTheory.FinPl.SplitPt D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → ∀ (V W : HSp D) (χV χW : (AE D)ˣ →∗ Circle) (ω : Om V W), IsWeilDatumO ψ V W χV χW ω → ∀ (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FV : Frame k V.rank Y') (FW : Frame k W.rank Y) (Tm : SB (Matrix Y Y' k) →ₗ[ℂ] (WO V W → ℂ)), (ofSplit q).IsModel ψ FV FW Tm → ∀ g ∈ GlobalTheory.FinPl.grpOff V (· = q.v), ∀ h ∈ GlobalTheory.FinPl.grpOff W (· = q.v), (ofSplit q).IsModel ψ FV FW ((ω (g, h)) ∘ₗ Tm)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalSchrodingerModelUniqTransportExists

theorem **upAdd** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (x x' : Matrix Y Y' k) : p.up FS FT (x + x') = p.up FS FT x + p.up FS FT x'

theorem **upZero** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) {Y Y' : Type} [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] {S T : HSp D} (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) : p.up FS FT 0 = 0

theorem **charZero_of_place** {D : D7} {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) : CharZero k

theorem **localModel_uniq_transport_exist** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → (∀ T₁ T₂, p.IsModel ψ FS FT T₁ → p.IsModel ψ FS FT T₂ → ∀ Φ₀, Φ₀ ≠ 0 → T₁ Φ₀ = T₂ Φ₀ → T₁ = T₂) ∧ (∀ T₁, p.IsModel ψ FS FT T₁ → ∀ A : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ), (∀ f, IsThetaFunO ψ S T f → IsThetaFunO ψ S T (A f)) → (∀ f, IsThetaFunO ψ S T f → ∀ y : WO S T, IsHalf (p.e + cA D p.e) y → A (heisO ψ S T y f) = heisO ψ S T y (A f)) → p.IsModel ψ FS FT (A ∘ₗ T₁)) ∧ ∀ f₀, IsThetaFunO ψ S T f₀ → (∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) → heisO ψ S T (p.up FS FT x) f₀ = f₀) → (∀ y : WO S T, IsHalf (cA D p.e) y → (∀ x : Matrix Y Y' k, (∀ r q, x r q

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalSchrodingerModelUniqTransportExistsStatement

theorem **_root_.GlobalTheory.T.localSchrodingerModel_uniq_transport_exists** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (S T : HSp D) (FS : Frame k S.rank Y') (FT : Frame k T.rank Y) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (−1) → (∀ T₁ T₂, p.IsModel ψ FS FT T₁ → p.IsModel ψ FS FT T₂ → ∀ Φ₀, Φ₀ ≠ 0 → T₁ Φ₀ = T₂ Φ₀ → T₁ = T₂) ∧ (∀ T₁, p.IsModel ψ FS FT T₁ → ∀ A : (WO S T → ℂ) →ₗ[ℂ] (WO S T → ℂ), (∀ f, IsThetaFunO ψ S T f → IsThetaFunO ψ S T (A f)) → (∀ f, IsThetaFunO ψ S T f → ∀ y : WO S T, IsHalf (p.e + cA D p.e) y → A (heisO ψ S T y f) = heisO ψ S T y (A f)) → p.IsModel ψ FS FT (A ∘ₗ T₁)) ∧ ∀ f₀, IsThetaFunO ψ S T f₀ → (∀ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) → heisO ψ S T (p.up FS FT x) f₀ = f₀) → (∀ y : WO S T, IsHalf (cA D p.e) y → (∀

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalSelfDualLatticeSplitPlaceDecomposition

theorem **isHalf_neg** {S T : HSp D} {e : AE D} {y : WO S T} (h : IsHalf e y) : IsHalf e (−y)

theorem **symplO_neg_right** (S T : HSp D) (x y : WO S T) : symplO S T x (−y) = −symplO S T x y

theorem **psi_neg_eq_one** {ψ : AddChar (AF D.host.F) ℂ} {a : AF D.host.F} (h : ψ a = 1) : ψ (−a) = 1

theorem **eq_zero_of_halves** {e : AE D} (he : IsSplitIdem D e) {S T : HSp D} {y : WO S T} (h1 : IsHalf e y) (h2 : IsHalf (cA D e) y) : y = 0

theorem **mem_lat** {u : WO S T} : u ∈ lat p ψ S T FS FT ↔ ∃ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ∧ ∃ y ∈ dualHalf p ψ S T FS FT, p.up FS FT x + y = u

theorem **up_mem_lat** {x : Matrix Y Y' k} (hx : ∀ r q, x r q ∈ 𝒪[k]) : p.up FS FT x ∈ lat p ψ S T FS FT

theorem **dualHalf_le_lat** : dualHalf p ψ S T FS FT ≤ lat p ψ S T FS FT

theorem **half_e_of_mem** {u : WO S T} (hu : u ∈ lat p ψ S T FS FT) (he : IsHalf p.e u) : ∃ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ∧ u = p.up FS FT x

theorem **half_c_of_mem** {u : WO S T} (hu : u ∈ lat p ψ S T FS FT) (hc : IsHalf (cA D p.e) u) : u ∈ dualHalf p ψ S T FS FT

theorem **selfDual_lattice_split** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y), ∃ Λ : AddSubgroup (WO S T), (∀ u ∈ Λ, ∀ u' ∈ Λ, ψ (symplO S T u u') = 1) ∧ (∀ y, IsHalf p.e y → y ∉ Λ → ∃ l ∈ Λ, IsHalf (cA D p.e) l ∧ ψ (symplO S T l y) ≠ 1) ∧ (∀ y, IsHalf (cA D p.e) y → y ∉ Λ → ∃ l ∈ Λ, IsHalf p.e l ∧ ψ (symplO S T l y) ≠ 1) ∧ (∀ x : Matrix Y Y' k, p.up FS FT x ∈ Λ ↔ ∀ r q, x r q ∈ 𝒪[k]) ∧ (∀ u ∈ Λ, IsHalf p.e u → ∃ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ∧ u = p.up FS FT x) ∧ (∀ u ∈ Λ, IsHalf (cA D p.e) u → ∀ t : Matrix Y Y' k, (∀ r q, t r q ∈ 𝒪[k]) → ψ (symplO S T (p.up FS FT t) u) = 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalSelfDualLatticeSplitPlaceDecompositionStatement

theorem **_root_.GlobalTheory.T.localSelfDualLattice_split_place_decomposition** : ∀ (D : D7) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (p : Place D k) (ψ : AddChar (AF D.host.F) ℂ), IsStdChar D ψ (-1) → ∀ (S T : HSp D) (Y Y' : Type) [Fintype Y] [DecidableEq Y] [Fintype Y'] [DecidableEq Y'] (FS : Frame k S.rank Y') (FT : Frame k T.rank Y), ∃ Λ : AddSubgroup (WO S T), (∀ u ∈ Λ, ∀ u' ∈ Λ, ψ (symplO S T u u') = 1) ∧ (∀ y, IsHalf p.e y → y ∉ Λ → ∃ l ∈ Λ, IsHalf (cA D p.e) l ∧ ψ (symplO S T l y) ≠ 1) ∧ (∀ y, IsHalf (cA D p.e) y → y ∉ Λ → ∃ l ∈ Λ, IsHalf p.e l ∧ ψ (symplO S T l y) ≠ 1) ∧ (∀ x : Matrix Y Y' k, p.up FS FT x ∈ Λ ↔ ∀ r q, x r q ∈ 𝒪[k]) ∧ (∀ u ∈ Λ, IsHalf p.e u → ∃ x : Matrix Y Y' k, (∀ r q, x r q ∈ 𝒪[k]) ∧ u = p.up FS FT x) ∧ (∀ u ∈ Λ, IsHalf (cA D p.e) u → ∀ t : Matrix Y Y' k, (∀ r q, t r q ∈ 𝒪[k]) → ψ (symplO S T (p.up FS FT t) u) = 1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalTrilinearFormUnique

@[simp] theorem **rt_apply** (g : GL n k) (Ψ : SB (Matrix m n k)) (x : Matrix m n k) : (rt g Ψ).1 x = Ψ.1 (x ∗ g.val)

theorem **rt_one** (Ψ : SB (Matrix m n k)) : rt (1 : GL n k) Ψ = Ψ

theorem **rt_rt** (a b : GL n k) (Ψ : SB (Matrix m n k)) : rt a (rt b Ψ) = rt (a ∗ b) Ψ

theorem **rt_mem_Mid** (l'' : kˣ →∗ ℂˣ) (π : Representation ℂ (GL m' k) V) (g : GL n k) (f : SB (Matrix m' n k) →ₗ[ℂ] V) (hf : f ∈ Mid l'' π) : f ∘ₗ rt g ∈ Mid l'' π

@[simp] theorem **piM_apply** (l'' : kˣ →∗ ℂˣ) (π : Representation ℂ (GL m' k) V) (g : GL n k) (f : Mid (n

@[simp] theorem **kernel_trilinear_apply** {V' A B V : Type} [AddCommGroup V'] [Module ℂ V'] [AddCommGroup A] [Module ℂ A] [AddCommGroup B] [Module ℂ B] [AddCommGroup V] [Module ℂ V] (θ : V' →ₗ[ℂ] A →ₗ[ℂ] B →ₗ[ℂ] V) (p : Submodule ℂ (B →ₗ[ℂ] V)) (hmem : ∀ v a, θ v a ∈ p) (v : V') (a : A) : (th1 θ p hmem v a).1 = θ v a

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalTwistDatumRenumberRowsColumns

theorem **val_le_iff** (x y : p.adicCompletion F) : valuation (p.adicCompletion F) x ≤ valuation (p.adicCompletion F) y ↔ Valued.v x ≤ Valued.v y

theorem **mem_O_iff** (x : p.adicCompletion F) : x ∈ 𝒪[p.adicCompletion F] ↔ Valued.v x ≤ 1

theorem **val_lt_one_iff** (x : p.adicCompletion F) : valuation (p.adicCompletion F) x < 1 ↔ Valued.v x < 1

theorem **val_eq_one_iff** (x : p.adicCompletion F) : valuation (p.adicCompletion F) x = 1 ↔ Valued.v x = 1

theorem **mem_O_iff'** (x : p.adicCompletion F) : x ∈ 𝒪[p.adicCompletion F] ↔ x ∈ p.adicCompletionIntegers F

theorem **isOpen_lt_one** : IsOpen {x : p.adicCompletion F | Valued.v x < 1}

theorem **isCompact_le_one** : IsCompact {x : p.adicCompletion F | Valued.v x ≤ 1}

theorem **residue_eq_zero_iff** (x : 𝒪[k]) : IsLocalRing.residue 𝒪[k] x = 0 ↔ valuation k (x : k) < 1

theorem **det_sub_lt** {n : Type} [Fintype n] [DecidableEq n] (M N : Matrix n n k) (hM : ∀ i j, M i j ∈ 𝒪[k]) (hN : ∀ i j, N i j ∈ 𝒪[k]) (h : ∀ i j, valuation k (M i j - N i j) < 1) : valuation k (M.det - N.det) < 1

theorem **inSupp_of_near** {x y : Matrix (A ⊕ D) (A ⊕ D') k} (hx : InSupp x) (h : ∀ i j, valuation k (y i j - x i j) < 1) : InSupp y ∧ valuation k (y.toBlocks₁₁.det - x.toBlocks₁₁.det) < 1

theorem **datum_eq_of_near** (ξ : kˣ →∗ ℂˣ) (hξ : GlobalTheory.SplitWeil.IsTame ξ) {x y : Matrix (A ⊕ D) (A ⊕ D') k} (hx : InSupp x) (h : ∀ i j, valuation k (y i j - x i j) < 1) : datum ξ y = datum ξ x

theorem **datum_isLocallyConstant** (ξ : (p.adicCompletion F)ˣ →∗ ℂˣ) (hξ : GlobalTheory.SplitWeil.IsTame ξ) : IsLocallyConstant (datum (k

theorem **datum_hasCompactSupport** (ξ : (p.adicCompletion F)ˣ →∗ ℂˣ) : HasCompactSupport (datum (k

theorem **le_inr** (j : A) : r ≤ (ε (.inr j) : ℕ)

theorem **eq_inl** (j : Fin a) (hj : j.val < r) : ε (.inl ⟨j, hj⟩) = j

theorem **exists_inr** (j : Fin a) (hj : r ≤ j.val) : ∃ j', ε (.inr j') = j

theorem **topL_eq** (z : Matrix (Fin b) (Fin a) κ) : Formalism.IndefR.Datum.topL r z = (z.submatrix ε' ε).toBlocks₁₁

theorem **phiQ_eq** (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) {r a b : ℕ} {A B : Type} [Fintype A] [DecidableEq A] [Fintype B] [DecidableEq B] (ε : Fin r ⊕ A ≃ Fin a) (ε' : Fin r ⊕ B ≃ Fin b) (hε : ∀ i, (ε (.inl i) : ℕ) = i) (hε' : ∀ i, (ε' (.inl i) : ℕ) = i) (ξ : (Fv D.host.F v)ˣ →∗ ℂˣ) : phiQ v r ξ = fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => datum ξ (z.submatrix ε' ε)

theorem **mem_SB_comp** (e : X ≃ₜ Y) {u : Y → ℂ} (hu : u ∈ GlobalTheory.SplitWeil.SB Y) : (u ∘ e) ∈ GlobalTheory.SplitWeil.SB X

theorem **mem_SB_comp_iff** (e : X ≃ₜ Y) (u : Y → ℂ) : (u ∘ e) ∈ GlobalTheory.SplitWeil.SB X ↔ u ∈ GlobalTheory.SplitWeil.SB Y

theorem **omTwo_mem_SB** (μS μT : (Fv D.host.F v)ˣ →∗ ℂˣ) (g : GL (Fin r ⊕ A) (Fv D.host.F v)) (h : GL (Fin r ⊕ B) (Fv D.host.F v)) (u : Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v) → ℂ) (hu : u ∈ GlobalTheory.SplitWeil.SB _) : GlobalTheory.ParHecke.omTwo μS μT g h u ∈ GlobalTheory.SplitWeil.SB _

theorem **omTwo_reidx** (hε : ∀ i, (ε (.inl i) : ℕ) = i) (hε' : ∀ i, (ε' (.inl i) : ℕ) = i) (μS μT : (Fv D.host.F v)ˣ →* ℂˣ) (g : GL (Fin r ⊕ A) (Fv D.host.F v)) (h : GL (Fin r ⊕ B) (Fv D.host.F v)) (u : Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v) → ℂ) (G : GL (Fin a) (Fv D.host.F v)) (H : GL (Fin b) (Fv D.host.F v)) (hG : G.1.submatrix ε ε = g.1) (hH : H.1.submatrix ε' ε' = h.1) : (fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => GlobalTheory.ParHecke.omTwo μS μT g h u (z.submatrix ε' ε)) = (((μT (GeneralLinearGroup.det G))⁻¹ * μS (GeneralLinearGroup.det H) : ℂˣ) : ℂ) • linOm v G H fun z => u (z.submatrix ε' ε)

theorem **ip_reidx** [MeasurableSpace (Fv D.host.F v)] [BorelSpace (Fv D.host.F v)] (ν : Measure (Fv D.host.F v)) (hν : ν.IsAddHaarMeasure) (u u' : Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v) → ℂ) : GlobalTheory.SplitWeil.ip ν (fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => u (z.submatrix ε' ε)) (fun z => u' (z.submatrix ε' ε)) = GlobalTheory.SplitWeil.ip ν u u'

theorem **GlobalTheory.Reidx.renumber** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (r a b : ℕ) (A B : Type) [Fintype A] [DecidableEq A] [Fintype B] [DecidableEq B] (ε : Fin r ⊕ A ≃ Fin a) (ε' : Fin r ⊕ B ≃ Fin b), (∀ i, (ε (.inl i) : ℕ) = i) → (∀ i, (ε' (.inl i) : ℕ) = i) → (∀ ξ : (Fv D.host.F v)ˣ →* ℂˣ, (phiQ v r ξ = fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => datum ξ (z.submatrix ε' ε)) ∧ (GlobalTheory.SplitWeil.IsTame ξ → datum ξ ∈ GlobalTheory.SplitWeil.SB (Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v)))) ∧ (∀ u : Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v) → ℂ, u ∈ GlobalTheory.SplitWeil.SB _ ↔ (fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => u (z.submatrix ε' ε)) ∈ GlobalTheory.SplitWeil.SB _) ∧ (∀ (μS μT : (Fv D.host.F v)ˣ →* ℂˣ) (g : GL (Fin r ⊕ A) (Fv D.host.F v)) (h : GL (Fin r ⊕ B) (Fv D.host.F v)) (u : Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v) → ℂ), (u ∈ G

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalTwistDatumRenumberRowsColumnsStatement

theorem **_root_.GlobalTheory.T.local_twist_datum_renumber_rows_columns** : ∀ (D : D7) (v : HeightOneSpectrum (𝒪 D.host.F)) (r a b : ℕ) (A B : Type) [Fintype A] [DecidableEq A] [Fintype B] [DecidableEq B] (ε : Fin r ⊕ A ≃ Fin a) (ε' : Fin r ⊕ B ≃ Fin b), (∀ i, (ε (.inl i) : ℕ) = i) → (∀ i, (ε' (.inl i) : ℕ) = i) → (∀ ξ : (Fv D.host.F v)ˣ →* ℂˣ, (phiQ v r ξ = fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => datum ξ (z.submatrix ε' ε)) ∧ (GlobalTheory.SplitWeil.IsTame ξ → datum ξ ∈ GlobalTheory.SplitWeil.SB (Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v)))) ∧ (∀ u : Matrix (Fin r ⊕ B) (Fin r ⊕ A) (Fv D.host.F v) → ℂ, u ∈ GlobalTheory.SplitWeil.SB _ ↔ (fun z : Matrix (Fin b) (Fin a) (Fv D.host.F v) => u (z.submatrix ε' ε)) ∈ GlobalTheory.SplitWeil.SB _) ∧ (∀ (μS μT : (Fv D.host.F v)ˣ →* ℂˣ) (g : GL (Fin r ⊕ A) (Fv D.host.F v)) (h : GL (Fin r ⊕ B) (Fv D.host.F v)) (u : Matrix (Fin r ⊕ B) (Fin r

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalVolumeAgreesRowSubgroupUnitary

theorem **index_comap_of_dense_image** {A B : Type} [Group A] [Group B] (f : A →* B) (H : Subgroup B) (hf : ∀ b, ∃ a, (f a)⁻¹ * b ∈ H) : (H.comap f).index = H.index

theorem **relIndex_comap_of_dense_image** {Γ G : Type} [Group Γ] [Group G] (φ : Γ →* G) (K U : Subgroup G) (h : ∀ g ∈ U, ∃ γ, (φ γ)⁻¹ * g ∈ K ⊓ U) : (K.comap φ).relIndex (U.comap φ) = K.relIndex U

theorem **vol_agree** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalVolumeAgreesRowSubgroupUnitaryStatement

theorem **_root_.GlobalTheory.T.local_volume_agrees_row_subgroup_unitary** : ∀ (X : Fields) (Pr : Primes X) (R : Ranks) (Ch : Chars X Pr R.m) (sel : HeightOneSpectrum (𝒪 X.F) → HeightOneSpectrum (𝒪 X.E)) (S₀ : Set ℕ) (mu : Tup X.F X.E X.K S₀) (cst : LocCst X (rowSG Pr sel mu)) (σ₀ : X.K →+* ℂ) (P : rowSG Pr sel mu) (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (ι : X.E →+* k) (s : ℕ) (W : TwData k) [MeasurableSpace k] [BorelSpace k] (ν : Measure k), IsUnitVol ν → cst.Ok → AtTwist (R

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocalZetaIntegralOpenSubgroup

theorem **covers_opE_finite** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] [MeasurableSpace k] [BorelSpace k] (n n' : Type) [Fintype n] [DecidableEq n] [Fintype n'] [DecidableEq n'] (c p : ℕ) (_ : 1 ≤ p) (_ : p ≤ c) (ψ : AddChar k Circle) (_ : Continuous ψ) (_ : ψ ≠ 1) (ν : Measure k) [ν.IsAddHaarMeasure] (μ : kˣ →* ℂˣ) (_ : IsOpen (μ.ker : Set kˣ)) (J : Subgroup (GL (n ⊕ Fin c) k)) (_ : IsOpen (J : Set (GL (n ⊕ Fin c) k))) (Φ : SB (Matrix (n ⊕ Fin c) (n' ⊕ Fin (c + 1)) k)), ∃ T, Covers p J T fun h => opE ψ ν p (om μ 1 h Φ.1)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocallyConstantPartial

theorem **exists_majorant** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (F : 𝒮(X × A, E)) (k : ℕ) : ∃ b : X → ℝ, Integrable b μ ∧ ∀ x a, ‖a‖ ^ k * ‖F (x, a)‖ ≤ b x

theorem **continuous_slice** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (F : 𝒮(X × A, E)) (a : A) : Continuous fun x => F (x, a)

theorem **integrable_slice** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (F : 𝒮(X × A, E)) (a : A) : Integrable (fun x => F (x, a)) μ

theorem **hasFDerivAt_slice** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (F : 𝒮(X × A, E)) (x : X) (a : A) : HasFDerivAt (fun a => F (x, a)) (dSnd F (x, a)) a

theorem **hasFDerivAt_integral** {E : Type} [NormedAddCommGroup E] [NormedSpace ℝ E] (F : 𝒮(X × A, E)) (a : A) : HasFDerivAt (fun a => ∫ x, F (x, a) ∂μ) (∫ x, dSnd F (x, a) ∂μ) a

theorem **smooth_decay** (n : ℕ) : ∀ (E : Type) [NormedAddCommGroup E] [NormedSpace ℝ E] (F : 𝒮(X × A, E)), ContDiff ℝ n (fun a => ∫ x, F (x, a) ∂μ) ∧ ∀ k : ℕ, ∃ C : ℝ, ∀ a, ‖a‖ ^ k * ‖iteratedFDeriv ℝ n (fun a => ∫ x, F (x, a) ∂μ) a‖ ≤ C

theorem **schwartz_partial** [FiniteDimensional ℝ A] {V : Type} [NormedAddCommGroup V] [NormedSpace ℝ V] (f : 𝒮(V, ℂ)) (L₁ : X →ₗ[ℝ] V) (L₂ : A →ₗ[ℝ] V) (hL : ∀ x a, L₁ x + L₂ a = 0 → x = 0 ∧ a = 0) : (∀ a, Integrable (fun x => f (L₁ x + L₂ a)) μ) ∧ ∃ h : 𝒮(A, ℂ), ∀ a, h a = ∫ x, f (L₁ x + L₂ a) ∂μ

theorem **eventually_invariant** {g : W → ℂ} (h1 : IsLocallyConstant g) (h2 : HasCompactSupport g) : ∀ᶠ u in 𝒩 (0 : W), ∀ x, g (x + u) = g x

theorem **locallyConstant_partial** {g : W → ℂ} (h1 : IsLocallyConstant g) (h2 : HasCompactSupport g) {M₁ : Y → W} {M₂ : B → W} {Q₁ : W → Y} {Q₂ : W → B} (c₁ : Continuous M₁) (c₂ : Continuous M₂) (d₁ : Continuous Q₁) (d₂ : Continuous Q₂) (e₁ : ∀ y b, Q₁ (M₁ y + M₂ b) = y) (e₂ : ∀ y b, Q₂ (M₁ y + M₂ b) = b) : (∀ b, Integrable (fun y => g (M₁ y + M₂ b)) ν) ∧ IsLocallyConstant (fun b => ∫ y, g (M₁ y + M₂ b) ∂ν) ∧ HasCompactSupport fun b => ∫ y, g (M₁ y + M₂ b) ∂ν

theorem **exists_integral_split** {G X Y : Type} [AddCommGroup G] [TopologicalSpace G] [IsTopologicalAddGroup G] [MeasurableSpace G] [BorelSpace G] [AddCommGroup X] [TopologicalSpace X] [IsTopologicalAddGroup X] [LocallyCompactSpace X] [SecondCountableTopology X] [MeasurableSpace X] [BorelSpace X] [AddCommGroup Y] [TopologicalSpace Y] [IsTopologicalAddGroup Y] [LocallyCompactSpace Y] [SecondCountableTopology Y] [MeasurableSpace Y] [BorelSpace Y] (e : G ≃ₜ+ X × Y) (m : Measure G) [m.IsAddHaarMeasure] (μ : Measure X)

[μ.IsAddHaarMeasure] (ν : Measure Y) [ν.IsAddHaarMeasure] : ∃ c : ℂ, ∀ (u : X → ℂ) (v : Y → ℂ), Integrable u μ → Integrable v ν → Integrable (fun z => u (e z).1 * v (e z).2) m ∧ ∫ z, u (e z).1 * v (e z).2 ∂m = c * ((∫ x, u x ∂μ) * ∫ y, v y ∂ν)

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocallyConstantSupportInvarianceOfDenseRational

theorem **isOpen_box_set** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A : Type) [Fintype A] (c : kˣ) : IsOpen ((GlobalTheory.StepData.box A c : Submodule 𝒪[k] (A → k)) : Set (A → k))

theorem **support_invariance_of_dense** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (E : Type) [Field E] (ι : E →+* k), DenseRange ι → ∀ (A B : Type) [Fintype A] [Fintype B] (f : Matrix A B k → (A → k) → ℂ), IsLocallyConstant (fun p : Matrix A B k × (A → k) => f p.1 p.2) → ∀ (c : kˣ), (∀ (x : Matrix A B E) (y : A → E), f (x.map ι) (fun a => ι (y a)) ≠ 0 → (fun a => ι (y a)) ∈ box A c⁻¹) → (∀ (x : Matrix A B E) (y z : A → E), (fun a => ι (z a)) ∈ box A c → f (x.map ι) (fun a => ι (y a + z a)) = f (x.map ι) (fun a => ι (y a))) → ∀ x y, (f x y ≠ 0 → y ∈ box A c⁻¹) ∧ ∀ z ∈ box A c, f x (y + z) = f x y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocallyConstantSupportInvarianceOfDenseRationalStatement

theorem **_root_.GlobalTheory.T.locallyConstant_support_invariance_of_dense_rational** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (E : Type) [Field E] (ι : E →+* k), DenseRange ι → ∀ (A B : Type) [Fintype A] [Fintype B] (f : Matrix A B k → (A → k) → ℂ), IsLocallyConstant (fun p : Matrix A B k × (A → k) => f p.1 p.2) → ∀ (c : kˣ), (∀ (x : Matrix A B E) (y : A → E), f (x.map ι) (fun a => ι (y a)) ≠ 0 → (fun a => ι (y a)) ∈ box A c⁻¹) → (∀ (x : Matrix A B E) (y z : A → E), (fun a => ι (z a)) ∈ box A c → f (x.map ι) (fun a => ι (y a + z a)) = f (x.map ι) (fun a => ι (y a))) → ∀ x y, (f x y ≠ 0 → y ∈ box A c⁻¹) ∧ ∀ z ∈ box A c, f x (y + z) = f x y

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocallyFiniteOfTupleConditions

theorem **latO_mono'** (S T : HSp D) (LS : Set (Fin S.rank → D.host.E)) (LT : Set (Fin T.rank → D.host.E)) {P Q : Pl D → Prop} (h : ∀ v, Q v → P v) : latO S T LS LT Q ≤ latO S T LS LT P

theorem **vac_mono** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P Q : Pl D → Prop} {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) (h : ∀ v, Q v → P v) : Vac ψ S T LS LT Q f

theorem **theta_smul'** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : ℂ) {f : WO S T → ℂ} (hf : IsThetaFunO ψ S T f) : IsThetaFunO ψ S T (c • f) where

theorem **vac_smul** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} {LS : Lat S} {LT : Lat T} {P : Pl D → Prop} (c : ℂ) {f : WO S T → ℂ} (hf : Vac ψ S T LS LT P f) : Vac ψ S T LS LT P (c • f)

theorem **comb_smul'** {ψ : AddChar (AF D.host.F) ℂ} {S T : HSp D} (c : WO S T →₀ ℂ) (a : ℂ) (f : WO S T → ℂ) : comb ψ S T c (a • f) = a • comb ψ S T c f

theorem **mem_cong** {V : HSp D} {I : Ideal (AE D)} {g : GA D V} : g ∈ congGrp V I ↔ ∀ i j, Formalism.L2.Theta.mt V g i j - (1 : Matrix (Fin V.rank) (Fin V.rank) (AE D)) i j ∈ I

theorem **real_le_off** (V : HSp D) (P : Pl D → Prop) : GinfC V ≤ grpOff V P

theorem **grpOff_inf_le** (V : HSp D) (P P' : Pl D → Prop) : grpOff V P ⊓ grpOff V P' ≤ grpOff V (fun p => P p ∨ P' p)

theorem **exists_level** (D : D7) (V : HSp D) (N : Lat V) (A B : Finset (Pl D)) (JA JB : Subgroup (GA D V)) (b : GA D V) (hAB : Disjoint B A) (hN : IsSelfDualC V N) (hb : b ∈ GfC V) (hA : JA ≤ grpAt V (· ∈ A)) (hAc : IsCompact (JA : Set (GA D V))) (hAo : IsOpen ((JA.subgroupOf (grpAt V (· ∈ A)) : Subgroup _) : Set (grpAt V (· ∈ A)))) (hB : JB ≤ grpAt V (· ∈ B)) (hBc : IsCompact (JB : Set (GA D V))) (hBo : IsOpen ((JB.subgroupOf (grpAt V (· ∈ B)) : Subgroup _) : Set (grpAt V (· ∈ B)))) : ∃ K : Subgroup (GA D V), K ≤ GfC V ∧ IsCompact (K : Set (GA D V)) ∧ IsOpen ((K.subgroupOf (GfC V) : Subgroup (GfC V)) : Set (GfC V)) ∧ ∀ k, k ∈ K ↔ ∃ κ x y, InK V N (fun p => p ∉ A ∧ p ∉ B) κ ∧ x ∈ JA ∧ y ∈ JB ∧ k = b⁻¹ * (κ * x * y) * b

theorem **split_off** (D : D7) (V : HSp D) (A B : Finset (Pl D)) (hAB : Disjoint B A) (z : GA D V) (hz : z ∈ grpOff V (fun p => p ∉ A ∧ p ∉ B)) : ∃ a ∈ GinfC V, ∃ x ∈ grpAt V (· ∈ A), ∃ y ∈ grpAt V (· ∈ B), z = a * y * x

theorem **comm_at** (D : D7) (V : HSp D) (A B : Finset (Pl D)) (hAB : Disjoint B A) {x y : GA D V} (hx : x ∈ grpAt V (· ∈ A)) (hy : y ∈ grpAt V (· ∈ B)) : y * x = x * y

theorem **locFin** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → OffLaw v w → GSupp (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LocallyFiniteOfTupleConditionsStatement

theorem **_root_.GlobalTheory.T.locally_finite_of_tuple_conditions** : ∀ (D : D7) (c : Comm D) (v : VCom c) (t : Tup c) (w : VTup t), c.Is → v.Is → t.Is → VForm v w → OffLaw v w → GSupp (t

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LowerBlockCoefficientIsSphericalHeckeEigen

theorem **esymm_map_mul_right'** (t : ℂ) (M : Multiset ℂ) (j : ℕ) : (M.map fun b => b * t).esymm j = t ^ j * M.esymm j

theorem **det_torus_indic** {k : Type} [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] {D : Type} [Fintype D] [DecidableEq D] (ϖ : kˣ) (s : Finset D) : GeneralLinearGroup.det (torusElt k D ϖ (indic s)) = ϖ ^ s.card

theorem **genVal_shift** (q a N : ℕ) (B : Multiset ℂˣ) (u : ℂˣ) (n : ℕ) : genVal q a N B n = genVal q 0 N (B.map (· * u⁻¹)) n * ((u : ℂ) ^ n * ((Real.sqrt q : ℝ) : ℂ) ^ (a * n))

theorem **realKey** (a n : ℕ) (c q Hm Hmg : ℝ) (hc : 0 < c) (hq0 : 0 < q) (hq : c ^ 2 * Hmg = q ^ n * Hm) (h : a = 0 ∨ 0 < Hm) : (c * Real.sqrt Hmg) ^ a = (Real.sqrt q ^ n * Real.sqrt Hm) ^ a ∧ Real.sqrt Hmg ^ a ≠ 0

theorem **realNeg** (n : ℕ) (c q Hm Hmg : ℝ) (hc : 0 < c) (hq0 : 0 ≤ q) (hq : c ^ 2 * Hmg = q ^ n * Hm) (h : Hm ≤ 0) : Hmg ≤ 0

theorem **alg1** (fmg P r ξm ξκ ξn ξκ' C : ℂ) (hr : r ≠ 0) (h1 : ξm ≠ 0) (h2 : ξκ ≠ 0) (h3 : ξn ≠ 0) (h4 : ξκ' ≠ 0) (hf : fmg = (ξm * (ξκ * (ξn * ξκ')))⁻¹ * P / r) : fmg * (ξn * ξm * (C * r)) = ξκ⁻¹ * (C * (ξκ'⁻¹ * P))

theorem **alg2** (S P lam0 ξn ξm Q Sg : ℂ) (h1 : ξn ≠ 0) (h2 : ξm ≠ 0) (h3 : Q ≠ 0) (h4 : Sg ≠ 0) (h : S * (ξn * ξm * (Q * Sg)) = lam0 * (ξn * Q) * P) : S = lam0 * (ξm⁻¹ * P / Sg)

theorem **coeff_isEigen** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (ξ : kˣ →* ℂˣ), IsTame ξ → ¬ IsUnram ξ → ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (A ⊕ D) k) V) (B : Multiset ℂˣ) (x : V) (ℓ : V →ₗ[ℂ] ℂ), IsGen π ξ ϖ B x → IsCoType π ξ ℓ → ∀ H : GL D k → ℝ, IsHgt ϖ H → ∀ f : GL D k → ℂ, (∀ m, f m = (((ξ (GeneralLinearGroup.det m))⁻¹ : ℂˣ) : ℂ) * ℓ (π (emb₂ k (Equiv.refl (A ⊕ D)) (1, m)) x) / ((Real.sqrt (H m) : ℝ) : ℂ) ^ Fintype.card A) → (∀ κ ∈ KO k D, ∀ m, f (κ * m) = f m ∧ f (m * κ) = f m) ∧ ∀ (s : Finset D) R, IsRepsK (torusElt k D ϖ (indic s)) R → ∀ m, ∑ g ∈ R, f (m * g) = genVal (resCard k) 0 (Fintype.card D) (B.map (· * (ξ ϖ)⁻¹)) s.card * f m

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LowerBlockCoefficientIsSphericalHeckeEigenStatement

theorem **_root_.GlobalTheory.T.lowerBlockCoefficient_isSphericalHeckeEigen** : ∀ (k : Type) [Field k] [ValuativeRel k] [TopologicalSpace k] [IsNonarchimedeanLocalField k] (A D : Type) [Fintype A] [DecidableEq A] [Fintype D] [DecidableEq D] (ϖ : kˣ), valuation k (ϖ : k) = uniformizer k → ∀ (ξ : kˣ →* ℂˣ), IsTame ξ → ¬ IsUnram ξ → ∀ (V : Type) [AddCommGroup V] [Module ℂ V] (π : Representation ℂ (GL (A ⊕ D) k) V) (B : Multiset ℂˣ) (x : V) (ℓ : V →ₗ[ℂ] ℂ), IsGen π ξ ϖ B x → IsCoType π ξ ℓ → ∀ H : GL D k → ℝ, IsHgt ϖ H → ∀ f : GL D k → ℂ, (∀ m, f m = (((ξ (GeneralLinearGroup.det m))⁻¹ : ℂˣ) : ℂ) * ℓ (π (emb₂ k (Equiv.refl (A ⊕ D)) (1, m)) x) / ((Real.sqrt (H m) : ℝ) : ℂ) ^ Fintype.card A) → (∀ κ ∈ KO k D, ∀ m, f (κ * m) = f m ∧ f (m * κ) = f m) ∧ ∀ (s : Finset D) R, IsRepsK (torusElt k D ϖ (indic s)) R → ∀ m, ∑ g ∈ R, f (m * g) = genVal (resCard k) 0 (Fintype.card D) (B.map (· * (ξ ϖ)⁻¹)) s.card

Development.AbelianVarietyTheory.Proofs.DecoratedLatticeProofs.LowerBlockDoubleCosetOperatorKeepsLevelCharacter

theorem **sum_translate** {G : Type} [Group G] (J : Subgroup G) (g₀ : G) (R : Finset G) (hRJ : ∀ y ∈ R, y ∈ J) (hR : ∀ j ∈ J, ∃! j', j' ∈ R ∧ (j' * g₀)⁻¹ * (j * g₀) ∈ J) (χ : G → ℂ) (hmul : ∀ a ∈ J, ∀ b ∈ J, χ (a * b) = χ a * χ b) (hne : ∀ a ∈ J, χ a ≠ 0) (hconj : ∀ m ∈ J, g₀⁻¹ * m * g₀ ∈ J → χ (g₀⁻¹ * m * g₀) = χ m) (Φ : G → ℂ) (hΦ : ∀ x, ∀ k ∈ J, Φ (x * k) = χ k * Φ x) {y₀ : G} (hy₀ : y₀ ∈ J) : ∑ y ∈ R, (χ y)⁻¹ * Φ (y₀ * y * g₀) = χ y₀ * ∑ y ∈ R, (χ y)⁻¹ * Φ (y * g₀)

theorem **det_lower_ne** {j : GL (A ⊕ D) k} (hj : j ∈ parahoric k A D) : j.1.toBlocks₂₂.det ≠ 0

theorem **det_lower_conj_mtor** (ϖ : kˣ) (e : D → ℤ) (m : GL (A ⊕ D) k) : ((mtor ϖ e)⁻¹ * m * mtor ϖ e : GL (A ⊕ D) k).1.toBlocks₂₂.det = m.1.toBlocks₂₂.det

theorem **hecke_level** : ∀ D : D7, ForStageQ D fun c w j s => ∀ PP ∈ w.Q, ∀ (v : HeightOneSpectrum (𝒪 D.host.F)) (φ : SplitProj D v), IsProjAt φ PP → ∀ (ξl : (Fv D.host.F v)ˣ →* ℂˣ), LiftsChar φ PP ξl (w.ξ PP) → ∀ ι, IsChart s j φ ι → ∀ (ϖ : (Fv D.host.F v)ˣ) (e : Blocks j D.R.t → ℤ) (R : Finset (GL (Fin D.R.r ⊕ Blocks j D.R.t) (Fv D.host.F v))) (d : GL (Fin D.R.r ⊕ Blocks j D.R.t) (Fv D.host.F v) → (Fv D.host.F v)ˣ), GlobalTheory.ParHecke.IsReps (GlobalTheory.ZonalCellDecomposition.mtor ϖ e) R d → ∀ F : GA D s.V → ℂ, (∀ κ ∈ Lev.levG s.V s.β s.L, ∀ g, F (g * κ) = Lev.levChi s.V s.β s.L κ * F g) → ∀ κ ∈ Lev.levG s.V s.β s.L, ∀ g, ∑ y ∈ R, ((ξl (d y) : ℂˣ) : ℂ) * F (g * κ * ι (y * GlobalTheory.ZonalCellDecomposition.mtor ϖ e)) = Lev.levChi s.V s.β s.L κ * ∑ y ∈ R, ((ξl (d y) : ℂˣ) : ℂ) * F (g * ι (y * GlobalTheory.ZonalCellDecomposition.mtor ϖ e))